Design Strategies for Multiplayer Gaming on Embedded Platforms Using Wireless Networks

THESIS

Submitted in partial fulfillment

of the requirements for the degree of

DOCTOR OF PHILOSOPHY

by

MANOJ KUMAR

Under the Supervision of

Dr. Kaushik Saha

BIRLA INSTITUE OF TECHNOLOGY AND SCIENCE

PILANI (RAJASTHAN) INDIA

2014

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

PILANI (RAJASTHAN)

CERTIFICATE

This thesis is submitted under Regulation 8.20 (a) of the Academic Regulations for Doctoral Programmes which allows a faculty member of the Institute/Professional to do Ph.D. research without the benefit of a supervisor.

This is to certify that the thesis entitled **Design Strategies for Multiplayer Gaming on Embedded Platforms Using Wireless Networks,** which is submitted for award of Ph.D. Degree of the Institute embodies my original work.

Signature in full	: Manoj Kumar
Name in block letters	: **MANOJ KUMAR**
ID No	: **2005PHXF433P**
Designation	: **Senior Group Manager, STMicroelectronics Pvt Ltd, INDIA**
Date	: 23rd June 2014

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

PILANI (RAJASTHAN)

CERTIFICATE

This is to certify that the thesis entitled **Design Strategies for Multiplayer Gaming on Embedded Platforms Using Wireless Networks** by **MANOJ KUMAR** ID No. **2005PHXF433P which is submitted** for award of Ph. D. Degree of the Institute, embodies original work done by him under my supervision.

Signature in full of the Supervisor :

Name in capital block letters : **Dr. KAUSHIK SAHA**

Designation : **Director, Samsung R&D Institute India, Delhi, INDIA**

Date : 23rd June 2014

ACKNOWLEDGEMENTS

I am immensely thankful to Prof. B. N. Jain, Vice-Chancellor, BITS, Pilani for providing me this opportunity to pursue the off-campus Ph.D. of the Institute. I express my gratitude to Prof. Ashis Kumar Das, Dean, Research and Consultancy Division (RCD) and Prof. Sanjay Kumar Verma, Dean, Academic Research Division (ARD), BITS, Pilani for their official support and guidance.

I thank my Doctoral Advisory Committee (DAC) members, Dr. V.K. Chaubey and Dr. Rahul Banerjee, who spared their valuable time to go through my draft thesis and were audience to my pre-submission seminar in order to provide several valuable suggestions that immensely helped in improving the quality of my thesis report. Thanks to Dr. Anu Gupta and Dr. K.K. Gupta for their valuable suggestions during and after the pre-submission seminar.

I thank Dr. Dinesh Kumar, Dr. Sharad Shrivastava and other nucleus members of RCD, BITS, Pilani, for providing guidance and support during past few semesters. I also express my gratitude to the office staff of RCD whose secretarial assistance helped me in submitting the various evaluation documents in time and making pre-submission seminar smoothly. I thank Dr. Hemant R. Jadhav, Associate Dean, Academic Research Division (ARD) for his guidance.

I am specially thankful and indebted to my research supervisor, Dr. Kaushik Saha who has been a great source of direction for this work and a constant motivator.

My special thanks to my System LAB India colleagues who have been involved in projects referred to in this work. They have been associated with various projects in their normal work and their contributions have been indirectly relevant to this work. All this was possible thanks to the conducive learning environment created by Matteo Lo PRESTI in System LAB.

Special mention to my wife Shalini Sharma and son Pinaq Sharma who sacrificed several of their weekends and always helped to keep my motivation high during this period.

ABSTRACT

Gaming over home area networks usually involves two or more players. These players may either use handheld gadgets with displays or share a common display like a TV screen or a computer monitor. In the case of shared display. players connect to the display through a common gaming console which provides two or more ports for the players to connect their handheld units to. The players have the same level of access for the game actions and manipulate the game through their handheld devices.

The number of players in this case is limited to the number of ports available and more importantly the separation between the players and the video screen is limited by the length of the cords between the console and the handheld units. In general. there is not much consideration for the power consumption in this case as the unit is normally wall-socket powered.

The wireless connection to these handheld gadgets helps to solve the problems of distance and the number of ports. In other words the players have the flexibility to move within a certain area and more number of players could be added easily. If one of the player moves too far. he may be considered not participating in the game with or without his knowledge. Similarly. it is important that this player is allowed to join as soon as he reenters the coverage area.

Our studies are motivated by the fact that today. in a networked home many devices may have the possibility of being connected to each other through wireless with addressing capability. This situation is similar to multiplayer gaming where devices may be in the range or outside, switched on or off etc. Most devices may be switched on or off for long periods of time and may not be necessarily communicating any data for long periods.

We have studied the scenarios in wireless gaming where a limited number of players are participating and a fair chance to each of the participants is given. The players may choose to join or leave the game with no penalties for their actions or behaviors.

Several possibilities exist today to network the players or intelligent devices in home and industrial environments. The wireless network itself can be established using a variety of protocols available. Bluetooth has been widely adapted in short range due to its widespread availability in several mobile phones and gaming devices.

ZigBee is one such technology that primarily has been used for sensor nodes up to now and the present work attempts to use ZigBee for data exchanges that are not limited to control only. The ZigBee protocol is suitable for such scenarios and power consumption is less than the other available choices.

The present work compares Bluetooth and Wi-Fi. the current favorites for gaming with ZigBee which is a protocol primarily used for low data rate sensor networks. In our studies we have found that Bluetooth is not the ideal choice for gaming networks due to its high battery drain and limited range. We find that the ZigBee consumes significantly less power than Bluetooth or wireless LAN and it is better suited for gaming applications that are control oriented.

While studying the software and the processor architectures. we have found that Java is a good environment for gaming applications. The games based on Java have the advantage of being able to run on several other compatible environments without much rework for each individual target. This allows the use of desktop computer for easy development of games.

In case of the handheld gaming units. the Java games can be designed and ported to these platforms without great deal of reworking. This assumes the support availability for Java on the target. Since handheld games may usually use microcontroller based platform. the necessary support for Java on the chosen microcontroller platform becomes the main criterion for choosing the target.

The above limitation of the Java based platform can be overcome by porting a Java compatible core on an ASIC block. We have found Java Optimized Processor (JOP) to be an attractive open source implementation of a JAVA processor in hardware.

Energy efficiency is very important. particularly in battery powered devices. This becomes even more important when the regular sources of energy may not be available. ZigBee devices installed as sensor networks in unmanned or remote areas can be powered through solar energy and if the associated electronics is made more efficient, a longer useful life of the devices may be obtained. Optimizing the middleware or protocols offers additional battery life.

The gaming devices have several playing keys to produce the desired gaming effects. Microelectromechanical Systems (MEMS) are being increasingly proliferated for implementing tilt. acceleration. vibration and rotation actions in such devices. Actions of game players very closely resemble MEMS accelerometer and gyroscope characteristics. Several types of sensors can be combined to create an inertial measurement unit that offers ease of integration to the host platforms. One key finding of our work is that ZigBee and MEMS combined with Java environment on a platform present a very viable alternative to the existing choices.

It is possible to optimize the computing requirements even further by using a processor that is optimized for Java. is asynchronous and is capable of working with ZigBee protocol or the protocol that is even simpler than ZigBee to achieve the longer battery life and easier game environment specific to the wireless gaming needs. This is suggested as an area of future work.

A Java optimized processor block, ZigBee and MEMS combination is proposed to make the gaming environment effective in this work. There are limitations in terms of the bandwidth available and the ZigBee based platforms do not compete favorably with the other standards offering larger bandwidths and data rates. The applications of ZigBee and the way the network is formed have their own advantages in several situations. Its applications can be extended to other similar situations outside the gaming environment.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

List of Abbreviations, Symbols, Definitions

ASPIDA	: Asynchronous, open source, Processor IP of the DLX Architecture
BLE	: Bluetooth Low Energy
BT	: Bluetooth
CC	: Cloud Computing
CSMA/CA	: Carrier Sense Multiple Access with Collision Avoidance
CSS	: Cascading Style Sheets
DCF	: Distributed co-ordination function
DLX	: RISC processor architecture by Hennessey and Patterson
FFD	: Full Function Device
FPGA	: Field Programmable Gate Array
GAP	: Generic Access Profile
GUI	: Graphical User Interface
HART	: Highway Addressable Remote Transducer
IEEE	: Institute of Electrical and Electronics Engineers
IETF	: Internet Engineering Task Force
IMU	: Inertial Measurement Unit
IoT	: Internet of Things
ISM	: Industrial Scientific and Medical license free frequency band
Java™	: Java is an object oriented programming language, henceforth referred to as Java
JOP	: Java Optimized Processor

JSR	: Java Specification Request
JVM	: Java Virtual Machine
MAC	: Media Access Control
MCU	: Microcontroller Unit
MEMS	: Microelectromechanical Systems
OS	: Operating System
PAN	: Personal Area Network
PDA	: Personal Digital Assistant
PWM	: Pulse width modulation
QoS	: Quality of Service
RFD	: Reduced Function Device
RISC	: Reduced Instruction Set Computer
RTOS	: Real Time OS
RTSJ	: Real Time Specifications for Java
SIG	: Special Interest Group
WCET	: Worst Case Execution Time
WG	: Working Group
WMN	: Wireless mesh network
Wi-Fi™	: Wi-Fi alliance trademark to certify products complying to IEEE 802.11 standard, henceforth referred to as Wi-Fi
ZigBee®	: Wireless protocol based on IEEE 802.15.4 standard, henceforth referred to as ZigBee

1 INTRODUCTION AND BACKGROUND

Multiplayer gaming is an interesting area of study from the perspective of connected devices. The gaming patterns and behaviors are governed by the technology used for interaction, platforms, display mechanisms, and the feedback mechanisms on the gaming devices. Protocols, ease of joining and leaving, power consumption, cost, age and orientation of the players involved further influence the gaming behaviours.

Web based Multiplayer games can be designed using the availability of the internet backbone. The players in this case can be situated far away from each other. Each player has his own display, usually a computer. These games require availability of good internet bandwidth. Wired networks can also be formed without internet being used in situations where distance is not large and in some cases it is possible for the players to share the same display. The power consumption, even if important, is not a primary concern in such scenarios as gaming devices are usually powered from the direct wall supply.

On the other hand, gadgets for wireless games are usually powered by batteries. In case of wireless gaming too, there is a possibility of sharing a common display or if powerful processors are used, each user can have his own display. The physical distance between the players starts to play an important role in this case. The wireless gaming devices usually would operate with batteries and time between recharges is important.

Sensor oriented wireless protocols are meant to send infrequent and less data and are designed to conserve the battery power. Ad-hoc wireless networks have been studied and used for the wireless sensor networks. Typically these networks have many nodes that participate in the formation of the network. A software stack is often needed to perform all network related functions. The computational and power consumption requirements are driven by the network functions and the frequency of usage of these nodes.

In the instances when a great networking capability in terms of dynamic addressing is not needed and there is not a great loss if a few data packets are dropped, the node architecture can be simplified further. A user often chooses to use one of the available networking topologies that support extensive networking protocols even when the requirements are not so demanding. The *ready availability of the devices supporting the* *standard protocols often turns out to be the deciding factor* for using more complex devices even for simpler applications.

Several wireless standards offer limited or great ease of networking suitable for ad-hoc networks. ZigBee. Bluetooth and Wi-Fi are well known technologies that can be used to form such networks. Bluetooth and Wireless LANs are widely being used in the hand held mobile terminals and computing environments. ZigBee is not in very widespread usage as of date for in consumer applications, its applications being limited to sensor networks or industrial applications.

Wireless gaming industry has been using the dedicated or proprietary protocols. Ad-hoc networks have not been applied widely in the gaming applications. Ad-hoc networking based on standard technologies is the scope of present work and our studies have revealed that ZigBee is an excellent choice for these applications.

ZigBee provides good ad-hoc capability and this feature has been used in sensor networks. ZigBee nodes can very easily join and leave the network. The protocol provides the capability of various low power modes and switching to the active mode from the low power modes and vice-versa is fast. The effective data rate in ZigBee is low, 250 kbps being the maximum on-air data rate [1].

While there has been a lot of discussion about the protocols, less attention has been paid to the platforms. This may be in view of the fact that implementation details of embedded computing platforms are usually proprietary and not available in the public domain. In this work, we study possibilities of using different platforms and highlight the usage of each choice.

There are several games available from internet sources that work with computer keyboards. Typically these are single user or two player games. These games use up, down, left and right arrow keys of the computer keyboard for controlling game actions. In case of two player games, another set of four computer keys is used to capture the actions by the second player.

MEMS can be used to improve the user experience of gaming. Tilt, vibration, movement control actions in the available games, which are usually implemented using a set of computer keyboard keys, may be captured with greater ease using MEMS sensors. In the present work the computer software drivers have been modified in a manner to capture the MEMS action that is the equivalent of the keys for which these games were originally designed. This saved a lot of time to prove the concept as several readily available games could be quickly adapted to the MEMS hardware designed.

Battery consumption is an important parameter for battery operated devices. The usual approach for making such devices is to start building the platform using one of the readily available computing platforms. The power consumption figures are thus dependent on the platform chosen. Other factors in determining battery power consumption are the wireless protocols and the radios chosen.

This work proposes and demonstrates the use of low data rate, control oriented wireless communication protocols (like ZigBee) for gaming applications in preference over the established protocols like Bluetooth and Wi-Fi for ad-hoc networking. The platform selection, usage of MEMS for better gaming experience and more efficient electronics for lower power consumption of handheld units are discussed this work.

In order to prove the proposed concepts, different environments have been created and tested. These include several platforms that are based on ZigBee, MEMS, microcontrollers and Java based computer games played by two or more players with ZigBee connectivity.

The ZigBee based wireless gaming platforms are primarily used for the purpose of this thesis. Wi-Fi and Bluetooth are used for comparison as it is relatively easy to find modules supporting these technologies and these are in widespread usage currently.

1.1 Current Implementations

Current wireless gaming environments are primarily based on Bluetooth and Wi-Fi. Besides these, there are now a host of online gaming applications based on other high data rate protocols as well.

Wireless ad-hoc networks consist of a collection of peer wireless mobile or stationary nodes that are capable of communicating with each other without any help from fixed infrastructures [2]. The interconnections among nodes often change continually and arbitrarily. Nodes within each other's radio range (one-hop) communicate directly via wireless links, while those that are far apart use other nodes as relays in multi-hop routing.

These networks will play an increasingly important role in many environments, such as ad-hoc networking for collaborative and distributed computing, disaster recovery, crowd control and search-and-rescue operations etc.

In wireless ad-hoc networks, nodes communicate with far off destinations using intermediate nodes as relays. Since wireless nodes are energy constrained, it may not be in the best interest of a node to always accept relay requests. On the other hand, if all nodes decide not to expend energy in relaying, then network throughput will drop dramatically. Both these extreme scenarios (complete cooperation and complete noncooperation) are not in the best interest of a user. Cooperation in wireless ad-hoc networks has been described in [3]. A distributed and scalable acceptance algorithm called generous tit-for-tat (GTFT) has been proposed in this paper. The acceptance algorithm is used by the nodes to decide whether to accept or reject a relay request.

Co-operation strategies in ad-hoc networks using co-operative and non-co-operative game theory have been dealt in [4]. Every node locally rates its neighbors through a monitoring mechanism. The observations collected by the monitoring mechanism are processed to evaluate a reputation value associated to each neighbor. The reputation value is used in a step-like cooperation policy. Only those nodes that have reputations that satisfy the requirement of being greater than a defined threshold are served (i.e. data and routing packets are forwarded), while nodes with low reputation values are gradually isolated from the network.

The non-cooperative / cooperative games can be classified into two categories: complete and incomplete information games. In the complete information game, each player has all the knowledge about others' characteristics, strategy spaces, payoff functions, and so forth, but all these information are not necessarily available in incomplete information game [5].

Incompletely Co-operative game theory for the wireless mesh networks (WMN) has been proposed [6] for IEEE 802.11x distributed co-ordination function (DCF). In this proposal, each node estimates the current game state (e.g., the number of competing nodes). The node then adjusts its equilibrium strategy by tuning its local contention parameters (e.g., the minimum contention window) to the estimated game state. Finally, the game is repeated several times to get the optimal performance. To use the game effectively in WMNs, a hybrid

CSMA/CA protocol is presented by integrating a proposed virtual CSMA/CA and the standard CSMA/CA protocol. When a node has no packet to send, it contends for the channel in virtual CSMA/CA mode. In this way the node can estimate the game state and obtain the optimal strategy.

In the commercial space also, there are several implementations for the gaming platforms based on Bluetooth or Wi-Fi or both. The Wii console was the first major gaming platform in the market to make use of Bluetooth and Wi-Fi in a single platform [7]. Enhancements were made to ensure that the wireless controller system delivers the same reliable, low-latency responsiveness that gamers are accustomed to have in wired controllers. The Wii console also includes 54g® Wi-Fi solution, which provides a high-performance wireless link for playing online or multi-player games. Both Bluetooth and 802.11g Wi-Fi products operate in the 2.4 GHz radio frequency band, introducing the possibility of radio interference between the two radio signals. The other famous gadgets in the market today are Sony Playstation and Microsoft Xbox. These are based on the Wi-Fi or Bluetooth or a combination of these.

Similarly most mobile phones that are equipped with wireless gaming capability use primarily the Bluetooth connectivity. The smart phones from the leading mobile companies are again in this category where Wi-Fi or a combination with Bluetooth is used. For the online wireless gaming, Wi-Fi is the preferred technology as of today.

All mobile devices are equipped with some kind of battery for their operation. The nickel-cadmium and lithium-ion batteries generally used in laptops, cellular phones, PDAs, and other portable computing and communications devices have increased their energy capacity by 10 to 15 percent per year. However these technologies may be capable of providing only another 15 to 25 percent in the best case [8].

Some companies are looking for ways to supplement battery power with supercapacitors, which offer very high capacitance in a small package, generating energy via a static charge rather than an electrochemical process. Supercapacitors transfer energy and recharge faster than batteries. A number of companies are working to bring supercapacitors to laptops and other mobile devices [8].

Energy conservation techniques for highly interactive games have been studied in [9]. The three baseline approaches taken in this approach were (1) a "black box" approach where clients drop packet with no support from the application: (2) a "white box" approach where a game's Dead Reckoning threshold is varied. and (3) a "black box" approach where the game server varies its sending rate. By using various approaches an efficiency improvement of about 20% is reported [9].

A comprehensive summary of software based energy saving methodologies for handheld wireless communication devices is presented in [**10**]. The paper divides the energy efficiency into various topics including smart batteries. energy-efficient graphical user interface design. the concept of a sleep state for an operating system. power efficient communication. proxy assisted energy saving. source-level power control. transport control protocol based energy saving. upper-level power management. virtual memory on the network. programming and compilation techniques. integrated power management. energy estimation models. impacts of dynamic power management (DPM) strategies. and future directions in energy saving.

A vibration driven energy harvesting platform based on piezoelectric material is proposed in [**11**]. A new maximum power point tracking (MPPT) method for this platform is also presented. This platform harvests ambient vibration energy as its power source. and is capable of self starting. and self-powered operation without the need of a battery.

A survey of various technical papers on energy harvesting is presented in [**12**]. Various sources of energy available for harvesting are identified and described in the paper including the work carried out by other researchers. Energy Scavenging for Mobile and Wireless Electronics from human activity and ways of deriving limited energy from ambient heat. light. radio. or vibrations are presented in [**13**]. It is shown that with that heel-strike could harvest energy in the order of mill watts levels.

A very extensive review of the principles and state-of-art in motion-driven miniature energy harvesters. trends. applications. and possible future developments is presented in [**14**]. It is concluded that successful exploitation of motion energy harvesting for many applications is likely to require integrated design of the complete wireless system. including power-aware operation of the powered device.

The wireless gaming involves transceivers. The transceiver electronics energy cost is not negligible and often has a major impact on the battery life. An approach to enable the joint optimization of system and circuit parameters for energy limited wireless transceivers with arbitrary communication protocols is presented in [15].

An on-demand power management framework for ad-hoc networks that adapts to traffic load is presented in [16]. To achieve reduced energy consumption while maintaining effective communication, this framework integrates routing information from on-demand ad-hoc routing protocols and power management capabilities from the MAC layer. Energy conservation is achieved by judiciously turning on and off the radios of specific nodes in the network.

Gaming experience can be enhanced by use of MEMS and other sensors. Smart phones and other computing gadgets use MEMS sensors to enrich the gaming experience. Accelerometers types of MEMS are primarily used for gaming for their tilt and vibration detection capability [17].

The action of MEMS can be further extended to gesture recognition. To ensure a fast adoption rate of gesture recognition as a ubiquitous input mechanism, technologies already available in mobile phones can be utilized. Features like accelerometer sensing and vibro-tactile feedback are readily available in high-end mobile phones, and this should filter through to most mobile phones in the future [18].

A command interface for games based on hand gestures defined by postures, movement and location is described in explained in [19]. The large varieties of gestures are thus possible and this increases usability by allowing a better match between gestures and actions. The system uses computer vision requiring no sensors or markers on the user or background. The analysis of requirements for games, the architecture and implementation are discussed, as well as the results of several tests to evaluate how well each requirement is met.

Cloud Computing (CC) and Internet-of-Things (IoT) have seen significant progress in recent times. Cloud based peer-to-peer games have appeared since then. Cloud platform providers such as Amazon Web Services and Microsoft Azure offer video game companies the option to host virtual game servers in the cloud. These services now allow gamers to build custom game servers in the cloud [20]. Here, the server is responsible for maintaining game state information between clients, synchronization, and communications.

With the explosive growth of the mobile applications and emerging of Cloud Computing concept, Mobile Cloud Computing (MCC) has been introduced to be a potential technology for mobile services [21]. MCC refers to an infrastructure where both the data storage and the data processing happen outside of the mobile device. One of the technologies that is expected to advance MCC in this paper is the latest version of the Web's markup language, HTML5.

1.2 Ad-hoc networks and Multiplayer Wireless Games

The research efforts in the area of ad-hoc networks have evolved rapidly since the establishment of the Internet Engineering Task Force (IETF) Mobile Ad-hoc Network (MANET) workgroup. The major focal points have been the routing protocols and the ongoing enhancements to the hardware of personal devices. Solutions that lead to increase in effective bandwidth associated with relative decrease in bandwidth variations and end-to-end delays have been described in literature [22], [23]. Adaptive and self-organizing wireless networks are gaining in popularity [24], [25]. Several media access and routing protocols have been proposed for such networks and performance of such protocols was evaluated based on simulations. Researchers have evaluated the practicality of realizing an ad-hoc wireless network and investigated the performance issues related to various parameters like media access, routing and packet loss etc [24], [25].

Game theory also leads to self coordinating behavior. The desirable behavior of various participants in the game can be obtained from autonomous selfish agents in power control and random access protocols [26].

Several mobile computers were enhanced with ad-hoc routing capability and were deployed in an outdoor environment and communication performances associated with ad-hoc communication were evaluated [24], [25]. These computers periodically sent beacons to their neighbors to declare their presence. The impact of varying packet size, beaconing interval, route hop count, communication throughput, end-to-end delay and packet loss was examined. Mobility experiments were performed and route reconstruction time incurred was evaluated. File transfer times associated with sending information reliably over multi-hop wireless links were also presented. The experimental results obtained revealed that it was feasible to augment existing wireless computers with ad-hoc networking capability [24].

End-to-end performance in ad-hoc routes is less affected by beaconing intervals than packet size or route length. Similarly, communication throughput is more dependent on packet size and route length with the exception at very high beaconing frequencies. It was concluded that the route discovery time in ad-hoc wireless networks were more dependent on channel conditions and route length than variations in beaconing intervals.

Initial work in ad-hoc routing has considered only the problem of providing efficient mechanisms for finding paths in very dynamic networks, without considering security. Because of this, there are a number of attacks that can be used to manipulate the routing in an ad-hoc network. In this paper [27], these threats are described specifically showing their effects on ad-hoc on-demand distance vector and dynamic source routing. The protocol, named authenticated routing for ad-hoc networks (ARAN), uses public-key cryptographic mechanisms to defeat all identified attacks. ARAN is characterized and evaluated and shown to be able to effectively and efficiently discover secure routes within an ad-hoc network. A simplification of synchronization requirements is necessary to meet the low power goals of wireless sensor networks (WSNs). Simply scaling existing synchronization systems to WSN data rates consumes more power than the entire allotted node budget [27].

A cooperative game (also called coalitional) is a game in which the players can make binding commitments, as is not the case in the non cooperative game [2]. Analysis in cooperative game theory is centered on coalition formation and distribution of wealth gained through cooperation. Within these two areas, finding procedures leading to outcomes that are most likely to occur under reasonable rationality assumptions in various game situations and devising solution concepts showing attractive stability features are primary concerns in most research endeavors. Cooperative game theory is most naturally applied to situations arising in political science or international relations, where concepts like power are most important.

The definition draws the usual distinction between the two theories of games, but the real difference lies in the modeling approach. The main focus of the non cooperative game is individual rationality and individual optimal strategy, but the cooperative game emphasizes collective rationality, fairness, effectiveness etc.

The handheld mobile devices are now sometimes used in multiplayer games. The trend for the multiplayer Games for handheld devices is now to download the games over the air to the handheld device. Some carriers offer users the option to purchase their games on their website and then transfer the game from the computer to the handset via a data cable, Infrared or Bluetooth connection [**28**].

1.3 Gaming Platforms

A significant body of work exists in the domain of protocol optimization and energy consumption optimization. *Only a limited number of implementations are reported with platform details.* A wearable FPGA based sensor platform is reported in [**29**]. The hardware, the firmware and applications of the platform are described in this paper. Bluetooth has been chosen for the wireless connectivity in this platform and it is concluded that it may not be the optimal protocol for small devices powered by a battery. Wibree technology is expected to be a better choice for the next version in the future work proposed for the sensor platforms [**29**].

There are several gaming platforms available and several games have been implemented, particularly on mobile phones. Most of these platforms are proprietary and some of them allow creating games or other applications on these platforms by providing opensource operating systems or virtual platforms like Java.

1.4 Recent developments

Recently several new publications of similar interest have appeared. Dynamic integration of new ZigBee nodes into the existing home network has been reported in [**30**]. The main idea of the paper is that an ad-hoc ZigBee home network device is represented by a device proxy service in the proposed architecture so that it can be dynamically registered and discovered.

A cooperative differential game model to find a rate control solution based on the tradeoff between network throughput and energy efficiency is proposed in [**31**]. This model takes into account the network throughput increase and energy consumption reduction at the same time. It is shown in this paper that cost obtained by any play in grand coalition is lower than that of non-cooperative one.

In wireless packet-forwarding networks with selfish nodes, application of a repeated game can induce the nodes to forward each others' packets, so that the network performance can be improved. However, the nodes on the boundary of such networks cannot benefit from this strategy, as the other nodes do not depend on them. This problem is sometimes known as the curse of the boundary nodes. To overcome this problem, an approach based on coalition games is proposed [32], in which the boundary nodes can use cooperative transmission to help the backbone nodes in the middle of the network. In return, the backbone nodes are willing to forward the boundary nodes' packets.

A thesis on mathematical aspects of the cooperative interval games has been presented [33] with detailed explanation of applicability to the situations with uncertain payoffs. The situations where the games agents cannot await the realizations of their coalition payoffs cannot be modeled according to classical game theory.

A motion capture system with multi-sensing-node based on inertial MEMS sensors and ZigBee network has been reported [34]. It defines the multi-node motion capture system, which can track human motions and postures simultaneously and wirelessly. Also it reports designing multi micro inertial measurement units by integrating the inertial MEMS sensors, micro controller unit and ZigBee module.

In another instance of integrating more aspects of systems design, MEMS and J2ME have been used to implement a portable online monitoring system [35].

1.5 Limitations in the available literature

The current implementations of the wireless gaming are primarily based on the Bluetooth and Wi-Fi. ZigBee has been primarily used for sensors and ad-hoc networks. Several commercially available products are making use of Bluetooth and Wi-Fi whereas ZigBee usage remains limited to sensors and industrial controls. Bluetooth and Wi-Fi have been primarily used in mobile phones, tablet PCs, laptops or hotspots. In these applications typically more processing power is available in the host.

The aim of this work is to combine the techniques available for various aspects of wireless gaming and propose a complete system platform for gaming incorporating wireless networks, low cost platforms, energy saving mechanisms and MEMS usage for player interaction. An attempt has been made to reuse available computer games software in wireless multiplayer games. *Advantage has been taken of the fact that, in general, such* *games require only low data rate communication to exchange data concerning players' actions and the internal states of the state machine of the game.*

The gaming devices need to use chargers for charging the on-board batteries. These chargers are not very efficient and are generally considered to be good since the consumer gaming applications do not attach great significance to power saving. In the present work, efficient electronic devices are proposed that make the converters for battery chargers more efficient and added benefits are obtained when these are used in the context of solar power.

Most of the research available in literature is focused on specific aspects of multiplayer wireless gaming. There are good number of references available in each of the area of wireless networking, power saving, designing games with Java also. The aim of the current work is to treat the major multiplayer wireless gaming system design issues simultaneously and provide the possibility of extending them to many available games. Thereby, we demonstrate the utility of such platforms for other applications where the requirements on platforms, networking and information exchange may be similar.

Bluetooth and Wi-Fi are the established methods for wireless gaming today. ZigBee is mainly used for the sensor networks. In the present work the ZigBee is proposed to be a viable alternative for wireless gaming, utilizing the energy saving of the ZigBee and gaming experience of the Bluetooth and Wi-Fi. The energy consumption can further be reduced by using low power electronics.

The further extensions to this work would be in the fields of multimedia, industrial, smart energy, medical, robotics and also military applications for the co-coordinated action.

1.6 Research Methodology

In this work wireless technologies are studied and compared for the ad-hoc gaming network formation. Bluetooth, Wireless LAN and ZigBee are primarily compared for their suitability for gaming.

Primary and secondary sources, including internet resources, have been used to describe the theory of operation. Datasheets of the components used in the design of various platforms have been referred. The results of the various platforms have been produced and referred to in this work.

For the gaming environment. various operating systems are studied. Java platform is studied for implementation of games. ZigBee based games are introduced to replace the keyboard shortcuts that were originally used in the development of these games. Several platforms that illustrate the concepts utilized in this work are made and their results are presented to prove utility of wireless networking for the gaming environment. A platform based on microcontrollers. MEMS and ZigBee and integrated with the computer using the Java environment for gaming is then used.

Treatment of the platforms is presented with advantages of each. The user can decide to choose any platform and integrate a control oriented protocol available from open sources. The concepts of energy saving or harvesting are applicable to all platforms. Similarly the concept of Java using computers for gaming and MEMS for gaming actions are common across all the platforms.

1.7 Thesis Structure

This work is based on several concepts of communication. gaming experience and environment and their implementations in parts and finally a full system. There are a number of choices for each one of these areas.

The first chapter of this report is about the background of the subject. The research gaps and the limitations in the available literature are highlighted in this chapter. The gaming scenarios are introduced in *chapter two* to explain the various scenarios based on the experience today.

Chapter three is the comprehensive background for the available protocols and their comparison. It discusses the platforms for demonstrating the concepts. A microcontroller based platform and a Java optimized processor are then discussed. Operating systems for gaming and Java for gaming are discussed thereafter. The chapter concludes with energy efficiency and microelectromechanical systems discussions.

Sub-systems to prove the concept of ZigBee and MEMS are presented in *chapter four*. Porting of DLX processor and creation of Java based games for windows environments are illustrated in this chapter.

The above concepts are then utilized to demonstrate the wireless multiplayer gaming by using ZigBee and MEMS for gaming in *chapter five*. The Java games available are modified suitably for integrating the MEMS and ZigBee into the gaming environment.

The work is concluded in *chapter six* with a proposal to make an even more efficient platform that integrates DLX processor and JOP on an FPGA with solar platform and low loss electronics as a future scope of the work. The applicability of the work in different environment is highlighted and the limitations of the work are discussed.

Some aspects of this work have been presented in conferences and workshops. Low power microcontrollers and ZigBee capability and the electronics for low power appeared in international magazines and an IEEE journal.

2 GAMING SCENARIOS

In this chapter various possible scenarios for wireless gaming are presented where two or more players may be playing with / against each other. The following are the possible scenarios:

- TV monitor as console
- Computer monitor / Laptop as the console
- Games with their own display

These scenarios can have different wireless connectivity depending on the specific implementations.

2.1 TV monitor as a console

In this case, a TV monitor is connected to a gaming console. The gaming console has the wireless connectivity in built that may be Bluetooth, Wi-Fi, infrared or any other wireless implementation. In fact the console only as the receiver only in most of these games and may not be sending any data to the wireless remotes. This is particularly true when the transmitters are infrared ones. This is the simplest case and can be thought as extension of the present wired consoles with the gaming handheld devices converted to suitable wireless part.

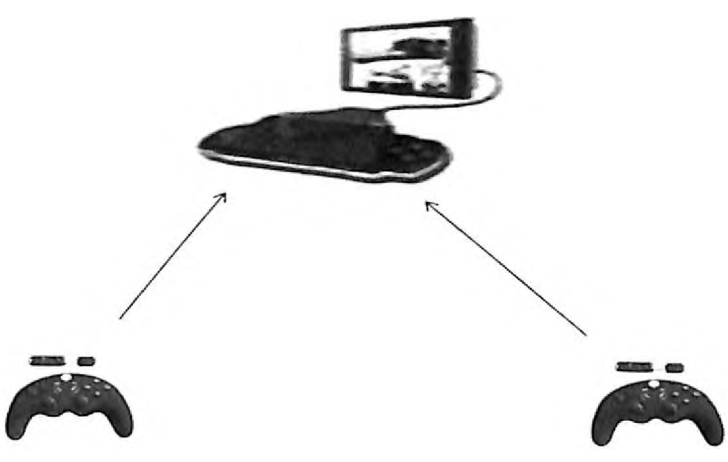


Figure 2-1: A multiplayer TV Game

The battery consumption starts to become important here as the handheld devices are now powered by the batteries. In the multiplayer TV games, the control is normally one way

2 GAMING SCENARIOS

In this chapter various possible scenarios for wireless gaming are presented where two or more players may be playing with / against each other. The following are the possible scenarios:

- TV monitor as console
- Computer monitor / Laptop as the console
- Games with their own display

These scenarios can have different wireless connectivity depending on the specific implementations.

2.1 TV monitor as a console

In this case, a TV monitor is connected to a gaming console. The gaming console has the wireless connectivity in built that may be Bluetooth, Wi-Fi, infrared or any other wireless implementation. In fact the console only as the receiver only in most of these games and may not be sending any data to the wireless remotes. This is particularly true when the transmitters are infrared ones. This is the simplest case and can be thought as extension of the present wired consoles with the gaming handheld devices converted to suitable wireless part.

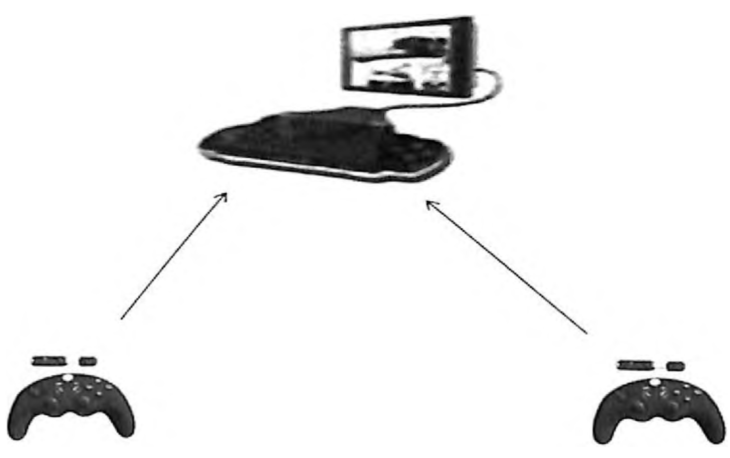


Figure 2-1: A multiplayer TV Game

The battery consumption starts to become important here as the handheld devices are now powered by the batteries. In the multiplayer TV games, the control is normally one way i.e. from the handheld unit to the console. The collisions are mostly overcome by the brute force as each player gets to execute the issued command by trial and wait. It is possible to implement schemes like time division duplex (TDD) to make the games more fair.

The idea of integrating the physical world of RC toys with the virtual world of multiplayer computer games has been experimented be researchers. The concept of game gadgets and their ability to detect RC toys in their surroundings is explained in this approach [36]. This paper presents a very interesting thought as to how the techniques of wireless sensor networks can be applied to gaming also.

2.2 Computer monitor as the console

Computer games played on consoles or PCs have become one of the most successful type of commercial applications, and gaming is today one of the biggest branches in the entertainment industry. As mobile devices have become more and more powerful with improved colour screens and graphics, the game industry has started to focus on game development for mobile and wireless devices. There are challenges for peer-to-peer games development in J2ME using the available Bluetooth API. The games can be grouped games according to the user interaction patterns or according to how data is updated [37].

In this case the computer or laptop monitor acts as the display screen for the participants. The Wireless connectivity can be established through numerous means here. Given the availability of several wireless devices that fit on to the USB port of the computer or the serial interface, the applications are numerous and more importantly can be developed on such platforms. A large database of these games is available that have been implemented for Java medium or any other environment in a computer and are played by typically two players by using the keyboard keys. The same games can be used with suitable modifications in the drivers so that the actions of the keys on the keyboard are replicated using the keys on the remote or handheld terminals. This increases the convenience of staying apart from other players during the play.

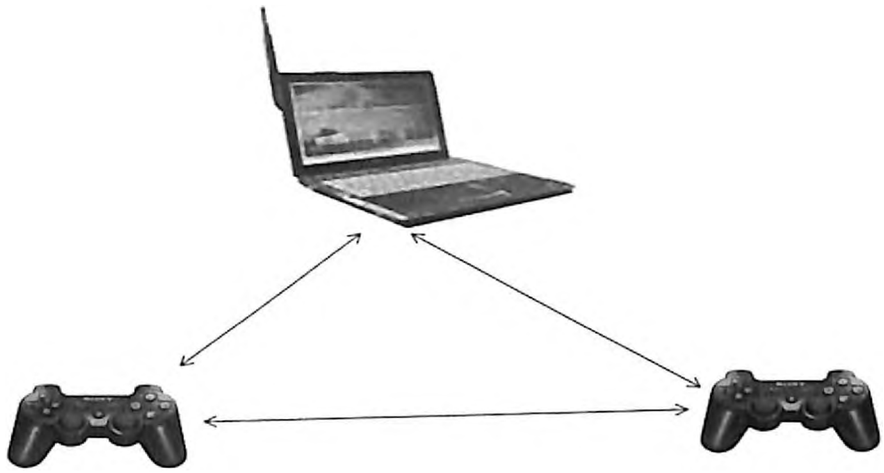


Figure 2-2: Computer as console for gaming

These platforms are most versatile as there is lot of programming possibility to modify and rebuild the games. This environment is mainly used in the present work.

2.3 Games with their own display

In a network intensive environment. it is possible to build games that have their own displays and two or more players can play the same game by concentrating on their own displays. It is the responsibility of the network transmissions to ensure synchronization of the games on all the connected terminals. The complexity of the protocol and the other associated overheads go on increasing with increasing number of devices. Typically these games need more bandwidth and higher transmission rates. Mobile phones and PDAs are the gadgets that have their own display and may be configured for multiplayer gaming.

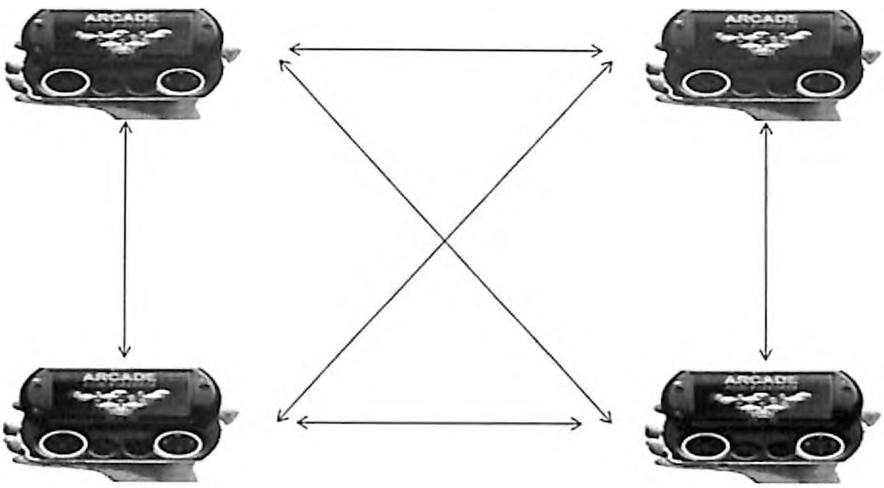


Figure 2-3: Handheld multiplayer games

In recent times, mobile games have gained popularity for providing personal entertainment on the go. This popularity has made mobile gaming play a pivotal role in revenue generation for the cellular carriers, game publishers and handset makers, while generating numerous opportunities for game developers and associated professionals. With the number of mobile gamers increasing around the world, the mobile gaming business is projected to expand to higher levels and constitute a bigger portion of the profit pie for the cellular carriers and handset makers [38].

Mobile games can be classified into three broad categories:

- **Embedded games:** Games that are hardcoded into the mobile handset's system and shipped with it. Soon to be outdated/obsolete. Example: Snake, shipped with all Nokia phones.

- **SMS games:** Games played by sending text messages—for example, SMS to game server—that process them and sends back the result through SMS. Often in the form of live contests and polls. Not very popular because the cost of gaming increases with each SMS sent to the game server.

- **Browser games:** These games are played using mobile phone's built-in microbrowser (net browser for mobile devices), either in online or offline mode. Players can play such games online through their cellular carrier's or a third-party game provider's game Web site, or download them for offline gaming. This category includes a wide range of games, such as solo or multiplayer games, network games, offline games, arcade games, and so forth

Among these three categories, browser games are today's most popular type of mobile games for their innovative and multimedia-rich content, appealing presentation, and lower cost of gaming compared to SMS games.

2.4 Classification of peer-to-peer games

The usual way of classifying games is to divide games into genres that reflect the game experience or gameplay. An example of such a classification is to divide games into the following categories: Action, adventure, driving, puzzle, role-playing, simulation, sports, and strategy [37]. A framework to classify games that can group games according to how network users interact and how the peer-to-peer network is proposed in reference [37] and explained in next sections.

2.4.1 User Interaction Based

Controlled: In this category, the user interactions of the gamers are controlled through a well-defined protocol, where one of the peers in the network must be a master controlling the user interaction. For games in this category, it is not up to the players themselves when to interact, since the master peer controls the interaction. This category suits well games that are turn-based or games where the users must interact in specific patterns or sequences. This category also covers games where all the users must follow a specific pattern or route (e.g. driving games).

User initiated: In this category, the users have to explicitly trigger actions that will cause interaction (exchange of data) with other gamers. This category suits well games that include trading of e.g. weapons, equipment and other items between gamers. In addition, this category covers games where it is up to the gamer how he wants to interact with the other gamers.

Automatically triggered: In this category, the mobile device of a gamer searches for other gamers nearby, and if one is found it can trigger an action to get the gamers attention (sound or vibration).

Automatic: In this category, the devices of the gamers interact without the user interacting with the game itself. The gamers typically configure and build autonomous characters that can interact with other user characters on behalf of their "owners". Such games can e.g. be role-playing games, where the player equips and trains his characters, and the fighting between network gamers goes on without the gamers interacting.

2.4.2 Synchronisation and data update based

The second dimension of the classification framework focuses on synchronisation and update of data between the peers. This dimension is divided into three categories:

Asynchronous: Asynchronous update is for network games where update between the peers is not time critical, but can be updated whenever possible. This category fits best games that are "slow-paced" and do need fresh data to proceed in the game.

Synchronous: In this category, the peers participating are depending on frequent update of data to be able to play the game. Further, for this category only small amount of information is necessary to be sent between the peers to keep the gamers in a consistent state. Such information can typically be game scores, lap-times, ranking, simple player states etc.

Real-time: In this category, the games rely on heavy data exchange between peers in order to give users a game experience. The data exchange between peers can typically be position of character or a vehicle in a common world or track, state of character or vehicle, movements of character or vehicle etc.

Even if the above framework is in the context of Bluetooth **[37]**, it applies equally to other kinds of transmissions like Wi-Fi, ZigBee etc. Interaction in such networks is dependent of the users to be geographically collocated (e.g. within a 10 meter radius) and the interaction between the users is highly dynamic.

Proximity-based ad-hoc interaction, enabled by personal area networks, is referred to as impromptu collaboration **[39]**, **[40]**. Impromptu collaboration is recognizable as being *opportunistic*, where the technology enables people to take advantage of opportunities that present themselves: *spontaneous*, where the collaboration effort is not planned in any way in advance; *proximity based*, where the peers have to be physically collocated; and *transient*, when the interaction between peers can be very short, e.g. a few minutes or seconds. Game sessions for games using ad-hoc peer-to-peer networks are generally shorter than console game sessions, but may last at least a couple of minutes. There are several reasons the mobile peer-to-peer game sessions are generally shorter than standard gaming sessions. First, the gamers are generally on the move and typically play games while waiting for something else to happen. Second, the energy consumption of playing network games on mobile devices will limit the length of the session. Finally, the smaller screens and limited input devices will cause the gamer to get more easily tired **[39]**.

Impromptu collaboration can involve different degrees of user interaction from situations where the device will do all the interaction on behalf of the users, to situations where the user actively interacts with nearby users. Further, the update frequency of the data sent between users is an important aspect of mobile peer-to-peer gaming. Based on these two facts a peer-to-peer game classification framework consisting of two dimensions has been presented in [**41**].

In the present work, the scenarios with a computer monitor are primarily discussed. This is done in order to take benefit of the large number of computer games that are already available and can be played by computer keyboards. This aspect has been widely used to demonstrate the usefulness of tilt action in wireless gaming.

3 TECHNOLOGY SURVEY

This chapter presents a brief theoretical overview about various technologies and techniques used for the purpose of this work. It includes wireless networking technologies and their comparison, discussion on available choices for processors, Java, energy efficiency for the gaming devices and introduction to working of MEMS. The chapter contains a detailed section on ZigBee for protocol, STM32 for Microcontroller platform, J2ME for operating environment and MEMS for gaming action.

ZigBee, Bluetooth and Wi-Fi have been compared in this chapter. An additional section on HTML5 has been included to compare it with Java for its usability in gaming applications. WirelessHART and Bluetooth Low Energy have been studied to complete the comparison for low power and low bandwidth.

Increasing number of sensors in various computing platforms requires sensor data to be presented in standard formats. The standardization efforts by various companies are presented in a section dedicated to this.

Display of a handheld device consumes significant amount of power. An attempt is made to show power consumption of some devices with and without display.

3.1 Competing Wireless standards for gaming environment

While choosing an environment, the first questions to be answered are the data rate and range requirements. In an indoor gaming scenario, often the star network becomes the preferred choice. The packet forwarding can normally be excluded as the response time is required to be within the acceptable limit. Speed has tradeoffs with range and latency. The latency measures the time required to transmit a data payload between devices.

Power consumption is particularly important in battery-powered devices such as wireless remotes or gaming consoles. The design must determine the run time required by the application, the battery type used (e.g. rechargeable or disposable), total amount of power consumed and voltage levels available to the device. The amount of processing required at the nodes is another parameter that influences the choice of the technology.

The main aspects for indoor wireless gaming are the ease of networking and the ability to join and leave seamlessly. Several technologies can be used for this purpose. Bluetooth, Wi-Fi and ZigBee are used for the purpose of comparison in this section.

3.1.1 Bluetooth

Bluetooth is an industrial specification for wireless personal area networks (PANs). Bluetooth provides a way to connect and exchange information between devices such as mobile phones. laptops. personal computers. printers. global positioning system (GPS) receivers. digital cameras. and video game consoles over a secure. globally unlicensed short-range radio frequency. The Bluetooth specifications are developed and licensed by the Bluetooth Special Interest Group. It is being used for sensor networks [**42**]

Bluetooth is a standard and communications protocol primarily designed for low power consumption. with a short range based on low-cost transceiver microchips in each device.

Bluetooth enables these devices to communicate with each other when they are in range. The devices use a radio communications system. so they do not have to be in line of sight of each other. and can even be in other rooms. as long as the received signal is above the threshold.

The Bluetooth standard defines different classes of devices. these are summarized in Table 3-1, [**43**]:

Table 3-1: Bluetooth Device Classes [43]

Class	Maximum Power (milliwatt / dBm)	Approximate Range (Meters)
1	100 / 20	100
2	2.5 / 4	10
3	1 / 0	1

The Bluetooth radio is built into a small microchip and operates in the 2.4Ghz band, a globally available frequency band ensuring communication compatibility worldwide. It uses frequency hopping spread spectrum, which changes its signal 1600 times per second which helps to avoid interception by unauthorized parties. In addition software controls and identity coding built into each microchip ensure that only those units that are authorized by their owners can communicate.

It supports both point-to-point and point-to-multipoint connections and provides up to 720 kbps data transfer within a range of 10 meters for Class 2 devices, Table 3-1. The technology uses omnidirectional radio waves that can transmit through walls and other non-metal barriers. If there is interference from other devices, the transmission speed decreases but does not stop.

Up to seven slave devices can be set to communicate with a master radio in one device. This connection of devices (slaves and master) is called a piconet. Several piconets can be linked together to form scatternets which allow communication between other device configurations.

Bluetooth operates in the 2.4 GHz industrial scientific and medical (ISM) band. In the US and Europe, a band of 83.5 MHz width is available. In this band, 79 RF channels spaced 1 MHz apart are defined. In France, a smaller band is available where 23 RF channels spaced 1 MHz apart are defined.

The channel is represented by a pseudo-random hopping sequence hopping through the 79 or 23 RF channels. Two or more Bluetooth devices using the same channel form a piconet. There is one master and one or more slave(s) in each piconet. The hopping sequence is unique for the piconet and is determined by the Bluetooth device address of the master. The phase in the hopping sequence is determined by the Bluetooth clock of the master. The channel is divided into time slots where each slot corresponds to an RF hop frequency. Consecutive hops correspond to different RF hop frequencies.

The channel is divided into time slots, each 625 μs in length. The time slots are numbered according to the Bluetooth clock of the piconet master.

A time division duplex (TDD) scheme is used where master and slave alternatively transmit. The master starts its transmission in even-numbered time slots only, and the slave starts its transmission in odd-numbered time slots only. The packet is aligned with the slot start.

3.1.2 Bluetooth Low Energy

Bluetooth Low Energy (BLE) is an emerging wireless technology developed by the Bluetooth Special Interest Group (SIG) for short-range communication. In contrast with previous Bluetooth flavors, BLE has been designed as a low-power solution for control and monitoring applications. BLE is the distinctive feature of the Bluetooth 4.0 specification [44].

The advent of BLE has occurred while other low-power wireless solutions, such as ZigBee, 6LoWPAN or Z-Wave, have been steadily gaining momentum in application domains that require multihop networking. However, BLE constitutes a single-hop solution applicable to a different space of use cases in areas such as healthcare, consumer electronics, smart energy and security.

The widespread use of Bluetooth technology (e.g., in mobile phones, laptops, automobiles, etc.) may fuel adoption of BLE, since implementation of the latter can leverage similarities with classic Bluetooth. According to published forecasts, BLE is expected to be used in billions of devices in the near future. In fact, the IETF 6LoWPAN Working Group (WG) has already recognized the importance of BLE for the Internet of Things. As of the writing of this article, the 6LoWPAN WG is developing a specification for the transmission of IPv6 packets over BLE.Bluetooth low energy technology offers ultra-low power, state-of the- art communication capabilities for consumer medical, mobile accessories, sports and wellness applications. Compared to classic Bluetooth capabilities, Bluetooth low energy is a connectionless protocol, which significantly reduces the amount of time the radio must be on. It requires only a fraction of the power consumption of traditional Bluetooth technology and can enable target applications to operate on a coin cell for more than a year.

Although BLE Controllers inherit some features from the classic Bluetooth Controllers, both types of Controllers are not compatible [45]. The Host implements the functionalities of the upper layers. These functionalities include : L2CAP, GAP, ATT, GATT and SM. The logical link control and adaptation protocol (L2CAP) defines the procedures for higher level multiplexing, packet segmentation and transfer of quality of service (QoS) information.

Table 3-2 BT and BLE Comparison [44]

	Bluetooth Low Energy	Classic Bluetooth
RF band (MHz)	2400	2400
Bit rate (kbps)	1000	≤721(v1.2), 3000 (v2+EDR), ≤24,000 (v3+HS)
Modulation	GFSK	GFSK (v1.2), GFSK/π/4-DQPSK/8DPSK (v2+EDR), 802.11 (v3+HS)
Spreading technique	FHSS (2 MHz channel width)	FHSS(1 MHz channel width)
Receiver sensitivity (dBm)	≤−70(required) −87 to −93 (typical)	−90(typical)
Transmit power (dBm)	−20 to 10	20/4/0(Class 1/2/3)
Latency (ms)	<3	<100
Hop limit	1	Outside scope of Bluetooth specifications

The generic access profile (GAP) specifies generic procedures related to the discovery of devices, link establishment and termination management and procedures related to the use of the different security levels; it also includes the common format requirements of the parameters accessible on the user interface level. The security manager (SM) handles the management of pairing. authentication. bounding and encryption for BLE communication. The attribute protocol (ATT) specifies the mechanisms for discovering. reading and writing attributes on a peer device, and the generic attribute profile (GATT) provides the framework for discovering services and for reading and writing characteristic values.

Like IEEE 802.15.4. BLE operates in the 2.4 GHz ISM band. In order to reduce the transceivers' costs and the amount of energy consumed. BLE prescribes binary frequency modulation with a 1 Mbit/s over-the-air data rate. Unlike the classical Bluetooth. which uses 79 1-MHz-wide channels. BLE uses 40 2-MHz wide channels. Three of these channels. which are located between commonly used wireless local area network channels, are used for advertising and service discovery and are called advertising channels. The remaining 37 data channels are used to transfer the data. The transmission of data between BLE devices is bound to time units known as advertising and connection events.

3.1.3 Wireless LAN

Wi-Fi is based on the IEEE 802.11 family of standards and is primarily a local area networking (LAN) technology designed to provide in-building broadband coverage.

Current Wi-Fi systems based on IEEE 802.11a/g support a peak physical-layer data rate of 54Mbps and typically provide indoor coverage over a distance of 100 feet.

Wi-Fi has become the de facto standard for last feet broadband connectivity in homes. offices, and public hotspot locations. Wi-Fi offers remarkably higher peak data rates than the 3G systems, primarily since it operates over a larger 20 MHz or 25 MHz bandwidth but Wi-Fi systems are not designed to support high-speed mobility.

One significant advantage of Wi-Fi over WiMAX and 3G is the wide availability of terminal devices. A vast majority of laptops shipped today have a built-in Wi-Fi interface. Wi-Fi interfaces are now also being built into a variety of devices. including personal data assistants (PDAs). cordless and cellular phones. cameras. and media players.

All Wi-Fi networks are contention-based TDD systems where the access point and the mobile stations all compete to use the same channel. Because of the shared media operation, all Wi-Fi networks are half duplex.

Table 3-3: Summary of 802.11 standards [46]

Feature	**Wi-Fi (802.11b)**	**Wi-Fi (802.11a/g)**
Primary Application	Wireless LAN	Wireless LAN
Frequency Band	2.4 GHz ISM	2.4 GHz ISM (g). 5 GHz U-NII (a)
Channel Bandwidth	25 MHz	20 MHz
Half/Full Duplex	Half	Half
Radio Technology	Direct Sequence Spread Spectrum	OFDM (64-channels)
Bandwidth Efficiency	<=0.44 bps/Hz	<=2.7 bps/Hz
Modulation	QPSK	BPSK, QPSK, 16-, 64-QAM
FEC	None	Convolutional Code
Encryption	Optional-RC4 (AES in 802.11i)	Optional-RC4 (AES in 802.11i)
Mesh	Vendor Proprietary	Vendor Proprietary
Access Protocol	CSMA/CA	CSMA/CA

There are equipment vendors who are selling products capable of Wi-Fi mesh configurations, but those implementations incorporate technologies that are not defined in the standards. As a result of this, there are some products available in the consumer market that are not strictly Wi-Fi compliant.

The Wi-Fi standards define a fixed channel bandwidth of 25 MHz for 802.11b and 20 MHz for either 802.11a or g networks [46].

Attempts have been made to investigate the mechanism of saving energy by modifying the standard techniques available in the original standard based on cooperation among mobile devices operating within a cellular communication system [47]. The performance of cooperative system has been investigated in ideal case and has been compared with the cooperative system performance using 802.11 WLAN network. The comparison shows that cooperation using omnipresent technology does not perform as well as it is expected. The existing MAC layer protocol worsens the performance because it is not designed for this particular scenario. A new and improved scheme is needed for this scenario, and yet it should also be easy to develop on top of existing system. This issue is addressed by an energy saving scheme proposed in [47].

The proposed scheme in [47] is then simulated and compared with ideal condition (where no collision occurs) and 802.11 WLAN network. The result shows that the proposed scheme gives significant improvement from 802.11 WLAN network. In the subsequent investigations, it is also found that the system works better under higher number of mobile devices and higher cluster period. The later even approaches proposed scheme's performance closer to ideal case. The simulation also proves that the average system energy saving for a pure environment of wise strategy is better than selfish strategy.

The 802.11 WLANs can be designed to meet varying needs of throughput and energy. A non-cooperative game-theoretical study of the power and rate control problem in IEEE 802.11 WLANs where network participants choose appropriate transmission power and data rate to achieve maximum throughput with minimum energy consumption has been presented in [48].

3.1.4 ZigBee

There are several protocols which use 802.15.4 as its MAC layer. ZigBee is one of the most popular among these as of now. There is sometimes confusion between 802.15.4 and ZigBee and sometimes these are used interchangeably [49]. The ZigBee alliance is a consortium driven by industry and research institutions. It finalized the ZigBee specification in December 2004 and describes the higher layer protocols (networking, application) that operate on top of 802.15.4.

ZigBee, in addition 802.15.4, defines the nature of its data communication and routing within networks, as well as the way nodes discover one another [**50**].

The IEEE 802.15.4 standard for low data rate wireless personal area networks (PANs) is widely considered as one of the technology candidates for wireless sensor networks [**51**], [**52**].

Two network topologies are allowed by the standard, both of which rely on the presence of a central controller device known as the PAN coordinator. In a peer-to-peer topology, the devices can communicate with each other directly, as long as they are within the physical range. In a star shaped topology, the devices must communicate through a PAN coordinator. The network uses two types of channel access mechanisms, one based on the slotted Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) in which the slots are aligned with the beacon frames sent periodically by the PAN coordinator, and another based on unslotted CSMA-CA.

In the beacon enabled networks, channel time is divided into super-frames that are bounded by the beacon transmissions from the coordinator. The basic time units of the MAC protocol are the backoff periods to which all transmissions must be synchronized. In the uplink direction individual nodes access the channel using CSMA-CA algorithm, and the channel must be idle for two successive backoff periods before transmission can start. If the channel is found to be busy, the random backoff countdown is repeated, possibly with a larger starting value.

The WPAN Low Rate Task Group (TG4) was chartered to investigate a low data rate solution with multi-month to multi-year battery life and very low complexity [**53**]. This standard specifies two physical layers: an 868 MHz/915 MHz direct sequence spread spectrum PHY and a 2.4 GHz direct sequence spread spectrum PHY. The 2.4 GHz PHY supports an over air data rate of 250 kb/s and the 868 MHz/915 MHz PHY supports over the air data rates of 20 kb/s and 40 kb/s. The physical layer chosen depends on local regulations and user preference. Potential applications are sensors, interactive toys, smart badges, remote controls, and home automation.

The ZigBee standard is a superset of the 802.15.4 standard and specifies the network security layer, application framework and application profiles so that ZigBee-certified equipment from different vendors will interoperate seamlessly. A ZigBee-certified application must conform to both the ZigBee standard and the 802.15.4 standard.

The coexistence between IEEE 802.11 and IEEE 802.15.4 and measurement of the impact these two wireless technologies have on each other when operating concurrently and in range have been studied [54].

3.1.4.1 THE 802.15.4 and ZigBee

In this section, the underlying standard for ZigBee i.e. 802.15.4 is discussed and its relationship to ZigBee is described.

3.1.4.2 The 802.15.4 PHY

The standard offers two PHY options based on the frequency band. Both are based on direct sequence spread spectrum (DSSS). The data rate is 250 kbps at 2.4GHz, 40 kbps at 915 MHz and 20 kbps at 868 MHz.

The higher data rate at 2.4 GHz is attributed to a higher-order modulation scheme.

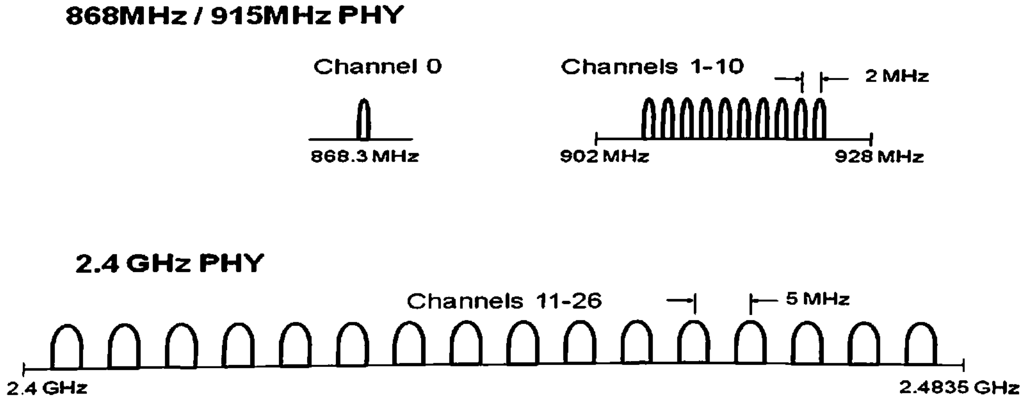


Figure 3-1 : The 802.15.4 Channels [1]

Lower frequency provides longer range due to lower propagation losses. Low rate can be translated into better sensitivity and larger coverage area. Higher rate means higher throughput, lower latency or lower duty cycle. The channel spacing and frequency bands are shown in Figure 3-1 [1].

The PHY provides two services: the PHY data service and PHY management service interfacing to the physical layer management entity (PLME). The PHY data service enables the transmission and reception of PHY protocol data units (PPDU) across the physical radio channel.

The features of the PHY are activation and deactivation of the radio transceiver, energy detection (ED), link quality indication (LQI), channel selection, clear channel assessment (CCA) and transmitting as well as receiving packets across the physical medium.

The frequency bands and modulation parameters are presented in Table 3-4 [43].

Table 3-4: 802.15.4 Frequency Bands and Modulation Parameters [43]

PHY (MHz)	Frequency Band (MHz)	Spreading Parameters		Data Parameters		No. of Channels
		Chip rate (kchips/s)	Modulation	Bit rate (kbps)	Symbol Rate (ksymbols/s)	
868/915	868-868.6	300	BPSK	20	20	1
	902-928	600	BPSK	40	40	10
2450	2400-2483.5	2000	O-QPSK	250	62.5	16

3.1.4.3 The 802.15.4 and ZigBee

The relationship of ZigBee with 802.15.4 is depicted in Figure 3-2. The PHY and MAC functions are indicated next to the layers depicted in the figure. The ZigBee is a layer that is on top of the 802.15.4 and is responsible for network routing, address translation, packet segmentation and profiles. The user applications are written on top of ZigBee profiles.

The IEEE 802.15.4-2003 standard defines the two lower layers: the physical (PHY) layer and the medium access control (MAC) sub-layer. The ZigBee Alliance builds on this foundation by providing the network (NWK) layer and the framework for the application layer. The application layer framework consists of the application support sub-layer (APS) and the ZigBee device objects (ZDO). Manufacturer-defined application objects use the framework and share APS and security services with the ZDO [55].

The ZigBee stack is defined by ZigBee alliance [56]. Its architecture is shown in Figure 3-2

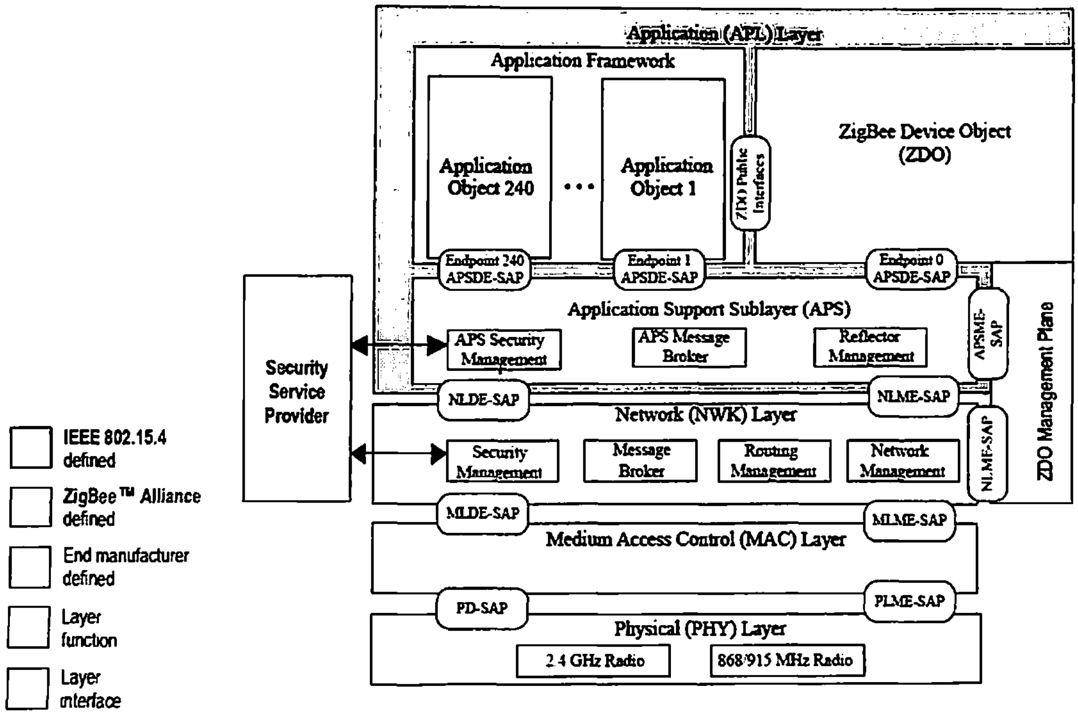


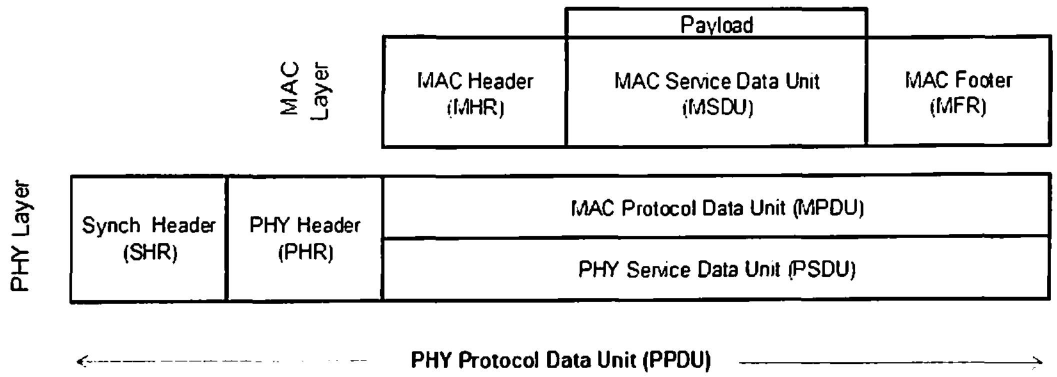
Figure 3-2: The ZigBee stack architecture [56]

IEEE 802.15.4-2003 has two PHY layers that operate in two separate frequency ranges: 868/915 MHz and 2.4 GHz. The lower frequency PHY layer covers both the 868 MHz European band and the 915 MHz band, used in countries such as the United States and Australia. The higher frequency PHY layer is used virtually worldwide.

3.1.4.4 The 802.15.4 Frame Structure

The 802.15.4 has four frame structures, each designated as PHY Protocol Data Unit (PPDU) in the standard for data transmissions – a beacon frame, a data frame, an acknowledgement frame and a MAC command frame.

All frames are structured in the similar fashion. The main difference is in the primary purpose or the payload. Each PPDU is constructed with a Synchronization header (SHR), a PHY header (PHR), and a PHY service data unit, composed of a MAC payload data unit (MPDU) as a data structure that services the MAC protocol layer. The frame structure format is shown in Figure 3-3 [1].



Types of MAC Frames : Beacon Frame, Data Frame, Acknowledgement Frame, MAC Command Frame

Figure 3-3: The 802.15.4 MAC Frames [1]

3.1.4.5 Key 802.15.4 Terms

- ***PAN coordinator.*** The PAN coordinator is the node (strictly speaking, the coordinator node) that initiates the network and is the primary controller of the network. The PAN coordinator may transmit beacons and can communicate directly with any device in range. Depending on the network design, it may have memory sufficient to store information on all devices in the network, and must have memory sufficient to store routing information as required by the algorithm employed by the network.

- ***Coordinator.*** The coordinator may transmit beacons and can communicate directly with any device in range. A coordinator may become a PAN coordinator, should it start a new network.

- ***Device.*** A network device does not beacon and can directly communicate only with a coordinator or PAN coordinator.

- ***Full function device*** (FFD). An FFD can operate in any of the three network roles (PAN coordinator, coordinator, or device). It must have memory sufficient to store routing information as required by the algorithm employed by the network. The complete protocol set is implemented in an FFD.

- ***Reduced function device*** (RFD). An RFD is a very low cost device, with minimal memory requirements. It can only function as a network device. Its role is limited to star topology or an end device in peer-to-peer network. It cannot become a PAN coordinator.

3.1.4.6 TOPOLOGIES

The IEEE 802.15.4 standard supports multiple network topologies. In the standard, two general types are discussed — star networks and peer-to-peer networks. In the star network, the master device is the PAN coordinator (an FFD), and the other network nodes may either be FFDs or RFDs. In the peer-to-peer network, FFDs are used, one of which is the PAN coordinator. RFDs may be used in a peer-to-peer network, but they can only communicate with a single FFD belonging to the network, and so do not save true "peer-to-peer" communication.

A ZigBee system consists of several components. The most basic one is the device. A device can be a full-function device (FFD) or reduced-function device (RFD). A network shall include at least one FFD, operating as the PAN coordinator. The FFD can operate in three modes: a personal area network (PAN) coordinator, a router, or a device. A RFD is intended for simple applications that do not need to send large amounts of data. A FFD can talk to RFDs or FFDs while a RFD can only talk to a FFD.

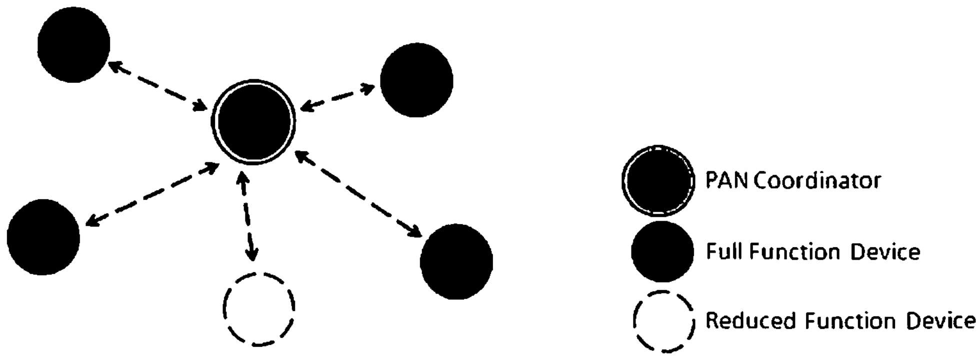


Figure 3-4: The Star Topology [51]

In the star topology, Figure 3-4 [51], the communication is established between devices and a single central controller, called the PAN coordinator. The PAN coordinator may be AC powered while other devices will most likely be battery powered. Applications for this topology include home automation, personal computer (PC) peripherals, toys and games. After a FFD is activated for the first time, it may establish its own network and become the PAN coordinator. Each star-topology network chooses a PAN identifier, which is not currently used by any other network within the communication range. This allows each star network to operate independently. Beacon is used to synchronize every node with PAN coordinator.

In peer-to-peer (mesh) topology, Figure 3-5 [51], there is also one PAN coordinator. In

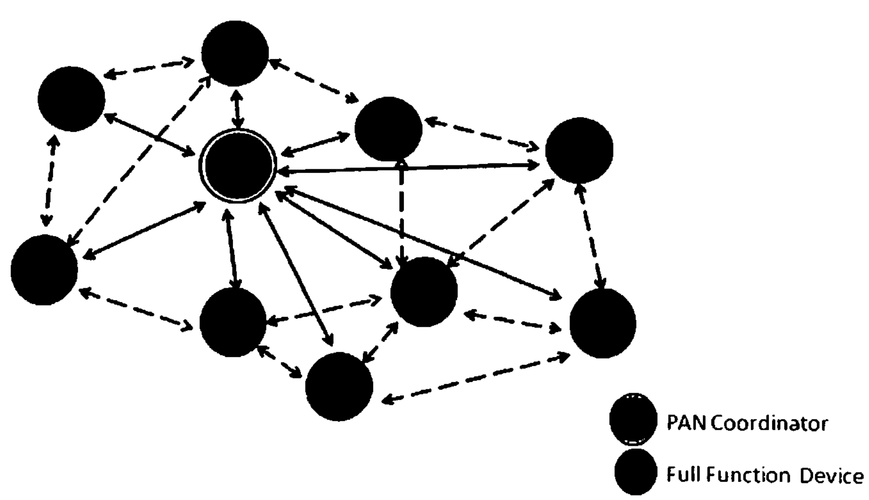


Figure 3-5: Mesh Topology [51]

contrast to star topology, any device can communicate with any other device as long as they are in range of one another. A peer-to-peer network can be ad-hoc, self-organizing and selfhealing. Applications such as industrial control and monitoring, wireless sensor networks, asset and inventory tracking benefit from such topology. It also allows multiple hops to route messages from any device to any other device in the network. It can provide reliability by multipath routing. Beacon is not used for peer-to-peer topology. This reduces control and increases collisions as compared to the beacon enabled network.

Cluster-tree network, Figure 3-6 [51], is a special case of a peer-to-peer network in which most devices are FFDs and a RFD may connect to a cluster-tree network as a leaf node at the end of a branch. Any of the FFD can act as a router and provide synchronization services to other devices and routers. Only one of these routers is the PAN coordinator. The PAN coordinator forms the first cluster by establishing itself as the cluster head (CLH) with a

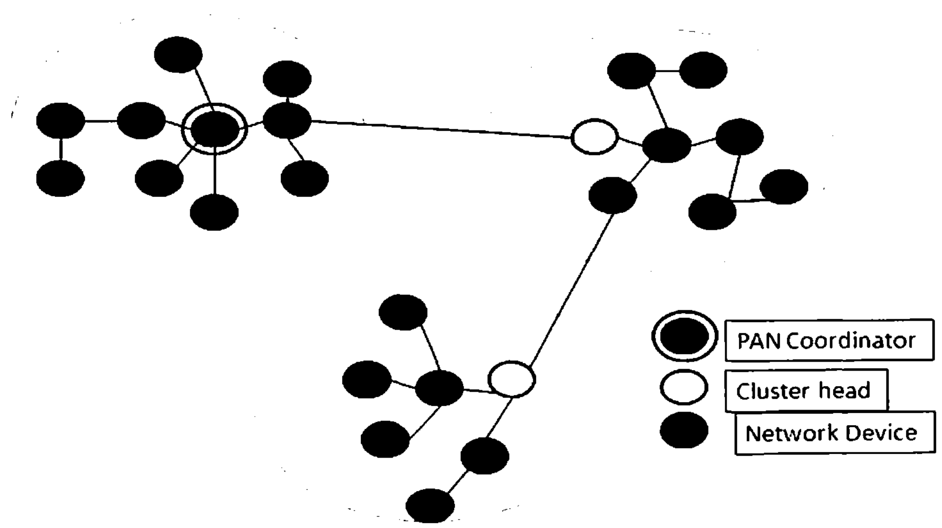


Figure 3-6: Cluster tree topology [51]

cluster identifier (CID) of zero, choosing an unused PAN identifier, and broadcasting beacon frames to neighboring devices. A candidate device receiving a beacon frame may request the CLH to join the network. If the PAN coordinator permits the device to join, it will add this new device as a child device in its neighbor list. The newly joined device will add the CLH as its parent in its neighbor list and begins transmitting periodic beacons such that other candidate devices may then join the network at that device. Once application or network requirements are met, the PAN coordinator may instruct a device to become the CLH of a new cluster adjacent to the first one. The advantage of this multi cluster, hierarchical structure is the increased coverage area at the cost of increased message latency.

The ZigBee network starts its formation as soon as the devices become active. The first FFD device that starts communication can establish itself as a ZigBee coordinator. All other devices that join later can request the ZigBee coordinator to join the network. Since no outside intervention is needed in this case, the ZigBee networks are considered to be self-forming networks.

Another feature of the mesh networking is that the network re-organizes itself in such a manner that the communication between the nodes continues even if some of the nodes stop to respond due to their movement of battery getting depleted.

Similarly ZigBee offers the ad-hoc networking, meaning thereby that the devices act as data pipes for the data that is not necessarily meant for them i.e. these devices may neither be the source or the destination of the data being forwarded or received.

3.1.4.7 ZigBee and ZigBee PRO

ZigBee PRO enhances the capabilities of ZigBee further. By adopting ZigBee PRO as an enhanced version, it is possible to provide for the additional capabilities of some applications, while retaining a simpler, low cost stack and retaining the lower power consumption for those applications that do not require the additional capabilities.The main differences between the ZigBee-2006v and the 2007 new release called ZigBee-PRO are [51].

- ***Stochastic addressing***: In the first ZigBee implementation the address was chosen by the Coordinator regarding the node position in the network tree. Now the 16b network address is chosen randomly. If the nodes choose the same address it is solved using the 64b IEEE 802.15.4 MAC address.

- ***Mesh data management***: In the previous implementation each node had to keep a table of any of the routes from and to the gateway to any device (if it was in the routing path), now the nodes just save the way to get to the gateway (reducing the memory space needed). The gateway (a node supposed to have higher RAM resources) stores the path (with all the hops) to any of the nodes. When the gateway has to send a packet to a specific node it appends the information about the hops which has to be taken in the same packet. This method is called "many to one".

- ***Fragmentation***: Large data packets can be easily fragmented.
- ***Dynamic best channel choice***: The nodes will move to another channel if the current one has interferences or noise (energy over a specific threshold).
- ***Asymmetric connections***: The links among the nodes are not always symmetric and the quality of the connection is different from node A to B than from node B to A. This is due to several reasons like the load of the node the interferences and the noise etc. For this reason the PRO version tries to take this issue into account and to make the best possible paths.
- ***Security***: The 2006 ZigBee implementation used up to AES 128 bit and a global network key to create secure communications. The new version has a more complex system which lets each couple of nodes to have their own key so that p2p encryption can be performed. A peer to peer link encryption layer is added.

3.1.4.8 BATTERY LIFE

The ZigBee compliant radios switch automatically to sleep mode and remain in this mode until there is a need for these to communicate. The ZigBee alliance does not specify the RF power. The IEEE 802.15.4 does specify a minimum power of 1mW. There are techniques that allow the RF power to be increased with sacrifice of the battery life. In some industrial applications, it is desirable to have high power ZigBee radios for better industrial controls. The sleep mode time and the lowest supply voltage that guarantees the correct operation are the two major parameters which affect the life of the node or the battery.

3.1.5 Bluetooth and Wireless LAN Comparison

In the preceding sections, various short range wireless technologies have been described. Wi-Fi, Bluetooth and ZigBee are now compared in tabular forms. Where it is not possible to compare equally, two of these have been compared separately. These are compared in Table 3-5 [55], [57].

Table 3-5: Bluetooth and Wireless LAN Comparison [58]

Standard	Bluetooth	Wireless LAN
Frequency band	2.4 GHz	2.4 GHz, 5 GHz
Coexistence mechanism	Adaptive frequency hopping	Transmit power control
Multiplexing	FHSS	DSSS, CCK, OFDM
Future multiplexing	UWB	MIMO
Noise adaptation	link layer	physical layer
Typical output power	1-10 mW (1-10 dBm)	30-100 mW (15-20 dBm)
Nominal range	10 m	100 m
Max one-way data rate	732 kb/s	31.4 Mb/s
Basic cell	Piconet	BSS
Extension of the basic cell	Scatternet	ESS
Max number of devices in the basic cell	8 active devices; 255 in park mode	Unlimited in ad-hoc networks (IBSS); up to 2007 devices in infrastructure networks (structured BSS).
Maximum signal rate	1 Mb/s	54 Mb/s
Channel access method	Centralised: Polling	Distributed: CSMA/CA
Channel efficiency	Constant	Decreasing with offered traffic
Spatial capacity	From 0.1 to 400 kb/s·m2	About 15 kb/s·m2

Data protection	16-bit CRC (ACL links only)	32-bit CRC
Procedures used for the network setup	Inquiry, Page	Ad-hoc networks: Scan, Authentication; Infrastructured: Scan, Authentication, Association
Average speed in network setup without external interferences	5 s + n·1.28 s, where n is the number of slaves in the Piconet, ranging from 1 to 7	n·c·1.35 ms for an unsaturated network, c probed channels ($1 \leq c \leq 13$), n stations (excluding the AP), active scan, Infrastructured topology
Authentication	Shared secret	Pairing Shared secret, Challenge-response
Encryption	E0 stream cipher	RC4 stream cipher
QoS mechanism	Link types	Coordination functions
Typical current absorbed	1 - 35 mA	100 - 350 mA
Power save modes	Sniff, Hold, Park; Standby	Doze

3.1.6 Battery Drain comparison of ZigBee and Bluetooth

Packet length can affect battery drain. Typically the shorter the packet, quicker the device can go to sleep. Bluetooth is a slotted protocol. Communication can occur in either: 625 µS, 1875 µS, or 3125 µS slots.

The peaks for the Bluetooth rate are a result of the three slot sizes. when a packet becomes too big for one slot it must increment to the next slot even though it doesn't fill the whole slot allocation.

IEEE 802.15.4 was designed for small packets so it is no surprise it is more efficient at those small packets resulting in a higher effective rate despite its lower raw data rate.

When the transmission predominantly involves larger packet sizes. the effective data rate approaches the raw data rate. For packets less than 75 bytes ZigBee has a higher effective data rate than Bluetooth. Having a lower rate for small packets means that BT needs longer transmit and receive times and therefore current drain is higher for small data packets.

3.1.7 Comparison Table for Wireless technologies

Various wireless standards for low range have been studied in detail. The NFC is good for mobile banking applications where user is assured that his data does not go beyond few centimeters. Bluetooth is good for applications where battery charging is a feature like mobile phones. Wi-Fi is good for applications where wall power is available and ZigBee is the choice where long battery life is required and low data rates are good enough.

The Table 3-6 compares the feature of various technologies in respect to the requirements of the ad-hoc wireless gaming scenario [55], [58]:

Table 3-6: Wireless technologies Comparison for ad-hoc networking [58]

Feature	Bluetooth	Wi-Fi	ZigBee
Range (m)	Up to 10 m	~ 100m	Up to 70 m
Speed	721 kbps	> = 11 Mbps	<= 250 kbps
System Resources	250 KB	1 MB	4KB-32 KB
Current Requirements	1-35 mA	100-350 mA	< 1 mA
Battery Life	1 – 7 days	0.5 – 5 days	100 – 1000 days
Number of nodes	8 (One master and 7	2007 (Structured	65,536

	slaves)	BSS)	
Ease of integration and availability	Easy	Lot of over head	One time cost / set up
Power Requirement	100 mW	750 mW	30 mW
Success Parameters	Cost, Convenience	Speed, Flexibility	Reliability, Power, Cost
Application Focus	Cable replacement	Web, Email, Video	Monitoring and control
Typical network join time	>3 sec	variable, 1 sec typically	30 ms typically
Interference avoidance method	FHSS (frequency-hopping spread spectrum)	FHSS, DSSS (802.11), DSSS (802.11b), OFDM (802.11a,g)	DSSS (direct-sequence spread spectrum)
Current consumption	60 mA (Tx mode)	400 mA (Tx mode) 20 mA (Standby mode)	25-35 mA (Tx mode) 3 μA (Standby mode)
Minimum quiet bandwidth required	15 MHz (dynamic)	22 MHz (static)	3 MHz (static)

Wi-Fi has all the virtues required for the gaming but it consumes more power hence results in maximum battery drain.

The number of nodes in each of the case needs elaboration. The maximum number of nodes in a Bluetooth network can be up to 255 in park mode and 7 devices can be active at a given time. The maximum number of devices in a structured 802.11 network is 2007. The ZigBee Alliance website [59] and some technical magazines [60], [61] contain a comparison between ZigBee and other technologies where network size for 802.11b is mentioned as 32 nodes. It may be mentioned here that correct number of nodes for 802.11b as per the IEEE standards is as given in Table 3-5 and Table 3-6. The interference avoidance methods mentioned in the Table 3-6 are not exclusive to the technologies mentioned. These are generic communication techniques which find usage is multiple standards.

Table 3-7 ZigBee and BLE Comparison [44]

	ZigBee	Bluetooth Low Energy (BLE)
Bit rate (kbps)	250	1000
Spreading Technique	DSSS	FHSS
Transmit Power (dBm)	-32 to 0	-20 to 10
Latency (ms)	<5	<3
Multi-hop Capability	Yes	No

Following the discussion in section 3.1.2, it is important to note that Bluetooth Low Energy (BLE) is proposed as an alternative to ad-hoc sensor networks [44]. It has much less power consumption than Classic Bluetooth and its latency is also comparable to that of ZigBee as shown in Table 3-7.

Although some of the BLE Controller features are inherited from the classic Bluetooth Controller, both types of Controller are currently incompatible. Hence, a device that only implements BLE (which is referred to as a single-mode device) cannot communicate with a device that only implements classic Bluetooth. It is expected that many devices will implement both the classic Bluetooth and the BLE protocol stacks. These devices are called dual-mode devices [44].

It is to be noted that BLE is a recent development and is starting to be a candidate for sensor networks only recently. Since the present work started much earlier the focus in this work stays on ZigBee. The BLE devices cannot be connected to Bluetooth enabled phones unless there is a dual mode support.

Very recently there are other wireless technologies that are making news. WirelessHART [62] and ANT [63] have been introduced. It has been shown that WirelessHART is more robust and consumes even lesser power than ZigBee [64].

WirelessHART is based on the existing wired HART (Highway Addressable Remote Transducer) protocol used in industrial automation. HART supports a data rate of 1.2 kbps using FSK modulation. It defines a time multiplexed protocol for accessing multiple nodes of sensors and actuators. WirelessHART is backward compatible with existing HART devices, commands and tools. Hence if an industrial installation is already using wired HART, addition of WirelessHART may be a good choice.

The usage of WirelessHART is not as widespread yet as ZigBee, Bluetooth and Wi-Fi technologies. The low data rate limits its usage to sensor controlling environments. Due to specificity of WirelessHART and considering it to be a natural extension to wired HART, its usage outside the controlled industrial environments is limited.

The usage of ANT, similarly is starting to grow in recent times. There are now devices available for fitness and sports that are having ANT capability.

Mobile battery power and similarly battery of any handheld device depends also on the medium access protocol perspectives in wireless networks. Energy conservation has typically been considered at physical layer issues, and to a certain extent at the access protocol level. Various energy conservation techniques proposed in different access protocols including IEEE 802.11 are described, analyzed and compared with other protocols by several researchers. The analysis shows that protocols that aim to reduce the number of contentions perform better from energy consumption perspective. The receiver usage time, however, tends to be higher for protocols that require the mobile to sense the medium before attempting transmission. [65]

For short-range RF transceivers operating in the Giga-Hertz carrier frequency range, the circuit electronics dominate the power consumption. Therefore it is important to model

It is to be noted that BLE is a recent development and is starting to be a candidate for sensor networks only recently. Since the present work started much earlier the focus in this work stays on ZigBee. The BLE devices cannot be connected to Bluetooth enabled phones unless there is a dual mode support.

Very recently there are other wireless technologies that are making news. WirelessHART [62] and ANT [63] have been introduced. It has been shown that WirelessHART is more robust and consumes even lesser power than ZigBee [64].

WirelessHART is based on the existing wired HART (Highway Addressable Remote Transducer) protocol used in industrial automation. HART supports a data rate of 1.2 kbps using FSK modulation. It defines a time multiplexed protocol for accessing multiple nodes of sensors and actuators. WirelessHART is backward compatible with existing HART devices, commands and tools. Hence if an industrial installation is already using wired HART, addition of WirelessHART may be a good choice.

The usage of WirelessHART is not as widespread yet as ZigBee, Bluetooth and Wi-Fi technologies. The low data rate limits its usage to sensor controlling environments. Due to specificity of WirelessHART and considering it to be a natural extension to wired HART, its usage outside the controlled industrial environments is limited.

The usage of ANT, similarly is starting to grow in recent times. There are now devices available for fitness and sports that are having ANT capability.

Mobile battery power and similarly battery of any handheld device depends also on the medium access protocol perspectives in wireless networks. Energy conservation has typically been considered at physical layer issues, and to a certain extent at the access protocol level. Various energy conservation techniques proposed in different access protocols including IEEE 802.11 are described, analyzed and compared with other protocols by several researchers. The analysis shows that protocols that aim to reduce the number of contentions perform better from energy consumption perspective. The receiver usage time, however, tends to be higher for protocols that require the mobile to sense the medium before attempting transmission. [65]

For short-range RF transceivers operating in the Giga-Hertz carrier frequency range, the circuit electronics dominate the power consumption. Therefore it is important to model the circuit power in battery life analysis. It is equally important to model the communication protocols and parameters such as modulation. data rate. and channel condition, as they determine the RF transmit power. the transceiver operation time, and the transceiver architecture. which in turn affect the circuit electronics power consumption. By working on the joint optimization, it is found that the battery life can be improved by nearly an order of magnitude [15].

Bluetooth and ZigBee emerge as the two technologies that are suitable for wireless gaming. ZigBee has all the merits required but is not so readily available for experimentation. Bluetooth though consumes more power than the ZigBee but has the advantage of being available easily on many mobile phones. Thus the choice for the wireless gaming is Bluetooth and ZigBee. *It is clear that ZigBee offers very good battery life, particularly for those cases when data transmission is not too heavy. The set-up times are very low for ZigBee compared to Bluetooth and Wi-Fi and ZigBee stack is several orders smaller than the other available choices. This makes ZigBee an ideal candidate for industrial control applications and wireless gaming where the data transmission requirements are of the order of few kilobits per second.*

3.2 Processors for Gaming

While choosing the right platform for any application, there are several parameters that a system designer has to take care:

- Performance
- Cost
- Energy and power
- Predictability
- Security

In this section. different possibilities for choosing a processor are discussed. In particular, representative cases are taken for clock-less processor. Java optimized processor and a microcontroller based platform.

3.2.1 The DLX Processor

Clock-less processors, also called asynchronous or self-timed, don't use the oscillating crystal that serves as the regularly "ticking" clock that paces the work done by traditional synchronous processors. Rather than waiting for a clock tick, clock-less chip elements hand off the results of their work as soon as they are finished [66]. Low Power requirements, increased performance and design for reuse are the key drivers for the clock-less approach in VLSI systems.

In the original MIPS design one of the methods used to gain performance was to force all instructions to complete in one cycle, forcing the compiler to insert "noops" in cases where the instruction would definitely take longer, as in memory access for instance. In the DLX design a more modern approach to long instructions has been used, using a data-forwarding system and reordering instructions. In this case the longer instructions are "stalled" in their functional units, and then re-inserted into the instruction stream when they do complete. Externally it appears execution occurred linearly.

DLX instructions can be broken down into three types, R-type, I-type and J-type. R-type instructions are pure register instructions, with an operand and three registers contained in the 32-bit word. I-type instructions are similar, but use include a single register only, and use the 16-bits used to indicate the other two registers in the R-type to hold an immediate value. Finally J-type instructions are jumps, containing an operand and a 26-bit address.

Instructions are 6-bits in length, for a total of 64 possible basic instructions. 4-bits are needed to select one of sixteen registers. In the case of R-type instructions this means that only 18-bits of the 32-bit word are used, which allows the lower 6-bits to be used as "extended instructions". This allows the DLX to support more than 64 instructions, as long as those instructions work purely on registers. This is useful for things like FPU (floating point unit) support.

The DLX, like the MIPS design, bases its performance on the use of an instruction pipeline. In the DLX design, the pipeline contains five stages [67]:

- *IF* - Instruction Fetch unit, typically referred to as "the load unit" in modern terminology

- ***ID*** - Instruction Decode unit, this unit gets instruction from IF, and extracts opcode and operand from that instruction. It also retrieves register values if requested by the operation.

- ***EX*** - Execution unit, runs the instructions, typically referred to as the ALU in modern terminology

- ***MEM*** - Memory access unit, the MEM unit fetches data from main memory, under the control of the instructions from ID and EX.

- ***WB*** - WriteBack unit, typically referred to as "the store unit" in modern terminology.

The HDL source for the DLX processor has been ported to EP3C25F324 available from Altera Corporation. The HDL description available from internet sources **[68]** has been used as the starting point.

3.2.2 The Java Optimized Processor

The Java is easily used on the desktop computers. However when the Java games are required to be run on the handheld devices, the rules of the game change. There are several constrains in the handheld devices, particularly related to the memory available and computing power. The J2ME or other Java environments are normally good when there are no limitations of the resources for computing.

For the handheld devices, the processor has to support Java. The Java Optimized Processor is a possible choice for the handheld computing platforms. It attempts to carry the same philosophy of Java of *write once, execute everywhere*. The common design practice for embedded systems is using an off-the-shelf microcontroller and programming in C with or without an RTOS. The JOP approach differs in the following ways:

- The processor is a soft core for an FPGA. FPGAs are expensive compared to microcontrollers. However it is possible to fit the processor core and other peripherals into a single FPGA itself. In this manner the FPGAs offer even more flexibility than the microcontroller based designs.

- The Java based systems need JVM as interpreter. The JOP implements the instruction set of the JVM in hardware, thus minimizing the gap between C and Java.

In this study, the JOP platform is considered as one possible platform for handheld multiplayer wireless gaming applications.

3.2.2.1 Salient Features

JOP is designed from the ground up with time-predictable execution of Java bytecode as a major design goal. All functional units, and especially the interactions between them, are carefully designed to avoid any time dependency between bytecodes.

JOP is a stack computer with its own instruction set, called microcode [69]. Java bytecodes are translated into microcode instructions or sequences of microcode. The main difference between JOP and JVM is that the JVM is CISC stack architecture, whereas JOP is RISC stack architecture. The architectural features and highlights of JOP are:

- Dynamic translation of the CISC Java bytecodes to a RISC, stack based instruction set (the microcode) that can be executed in a 3 stage pipeline.

- The translation takes exactly one cycle per bytecode and is therefore pipelined. Compared to other forms of dynamic code translation the proposed translation does not add any variable latency to the execution time and is therefore time predictable.

- Interrupts are inserted in the translation stage as special bytecodes and are transparent to the microcode pipeline.

- The short pipeline results in short conditional branch delays and a hard to analyze branch prediction logic or branch target buffer can be avoided.

- Simple execution stage with the two topmost stack elements as discrete registers. No write back stage or forwarding logic is needed.

- The Java based systems need JVM as interpreter. The JOP implements the instruction set of the JVM in hardware. thus minimizing the gap between C and Java.

In this study. the JOP platform is considered as one possible platform for handheld multiplayer wireless gaming applications.

3.2.2.1 Salient Features

JOP is designed from the ground up with time-predictable execution of Java bytecode as a major design goal. All functional units. and especially the interactions between them. are carefully designed to avoid any time dependency between bytecodes.

JOP is a stack computer with its own instruction set, called microcode **[69]**. Java bytecodes are translated into microcode instructions or sequences of microcode. The main difference between JOP and JVM is that the JVM is CISC stack architecture, whereas JOP is RISC stack architecture. The architectural features and highlights of JOP are:

- Dynamic translation of the CISC Java bytecodes to a RISC. stack based instruction set (the microcode) that can be executed in a 3 stage pipeline.

- The translation takes exactly one cycle per bytecode and is therefore pipelined. Compared to other forms of dynamic code translation the proposed translation does not add any variable latency to the execution time and is therefore time predictable.

- Interrupts are inserted in the translation stage as special bytecodes and are transparent to the microcode pipeline.

- The short pipeline results in short conditional branch delays and a hard to analyze branch prediction logic or branch target buffer can be avoided.

- Simple execution stage with the two topmost stack elements as discrete registers. No write back stage or forwarding logic is needed.

- Constant execution time (one cycle) for all microcode instructions. The microcode pipeline never stalls. Loads and stores of object fields are handled explicitly.

- No time dependencies between bytecodes result in a simple processor model for the low-level WCET (worst-case execution time) analysis.

- Time predictable instruction cache that caches whole methods. Only invoke and return instructions can result in a cache miss. All other instructions are guaranteed cache hits.

- Time predictable data cache for local variables and the operand stack. Access to local variables is a guaranteed hit and no pipeline stall can happen. Stack cache fill and spill is under microcode control and analyzable.

- No prefetch buffers or store buffers that can introduce unbound time dependencies of instructions. Even simple processors can contain an instruction prefetch buffer that prohibits exact WCET values. The design of the method cache and the translation unit avoids the variable latency of a prefetch buffer.

- Good average case performance compared to other non real-time Java processors.

- Avoidance of hard to analyze architectural features results in a very small design. Therefore an available real estate can be used for a chip multi-processor solution.

- JOP is the smallest hardware implementation of the JVM available to date. This fact enables usage of low-cost FPGAs in embedded systems. The resource usage of JOP can be configured to trade size against performance for different application domains.

- JOP is actually in use in several real-world applications showing that a Java based embedded system implemented in an FPGA is a viable option.

JOP is implemented as a soft-core in a field programmable gate array (FPGA) giving a lot of flexibility for the overall hardware design. The processor can easily be extended by peripheral components inside the same chip. Therefore, it is possible to customize the solution exactly to the needs of the system.

3.2.2.2 JOP Architecture

This section gives an overview of JOP architecture. Figure 3-7 shows JOP's major function units [70]. A typical configuration of JOP contains the processor core, a memory interface and a number of IO devices. The module extension provides the link between the processor core, and the memory and IO modules.

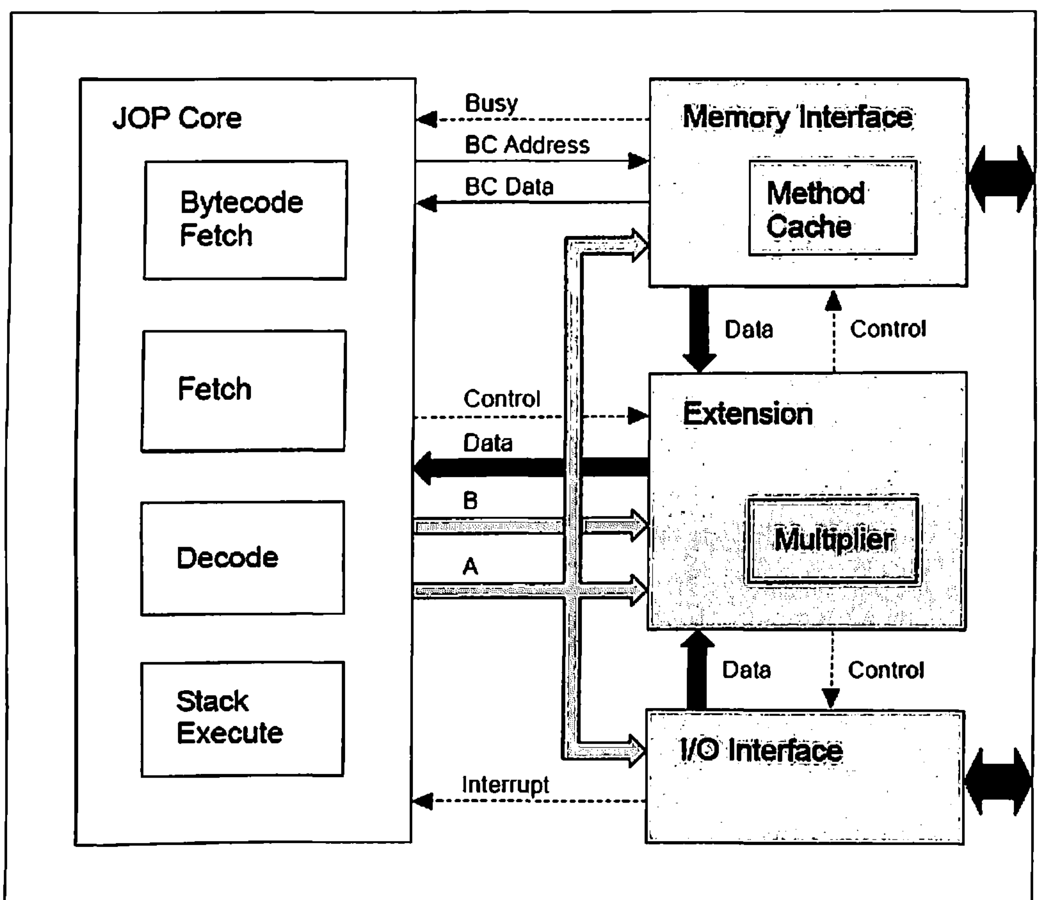


Figure 3-7: Java Processor Block Diagram [70]

The processor core contains the four pipeline stages: bytecode fetch, microcode fetch, decode and execute. The ports to the other modules are the address and data bus for the bytecode instructions, the two top elements of the stack (A and B), input to the top-of-stack

(Data) and a number of control signals. There is no direct connection between the processor core and the external world.

The memory interface provides a connection between the main memory and the processor core. It also contains the bytecode cache. The extension module controls data read and write. The busy signal is used by the microcode instruction wait to synchronize the processor core with the memory unit. The core reads bytecode instructions through dedicated buses (BC address and BC data) from the memory subsystem.

The request for a method to be placed in the cache is performed through the extension module, but the cache hit detection and load is performed by the memory interface independently of the processor core (and therefore concurrently).

The I/O interface contains peripheral devices, such as the system time and timer interrupt, a serial interface and application-specific devices. Read and write to and from this module are controlled by the extension module. All external devices are connected to the I/O interface.

The extension module performs three functions:

a. it contains hardware accelerators (such as the multiplier unit in this example),

b. the control for the memory and the I/O module, and

c. the multiplexer for the read data that is loaded in the top-of stack register.

The write data from the top-of-stack (A), Figure 3-7 is connected directly to all modules.

The division of the processor into those four modules greatly simplifies the adaptation of JOP for different application domains or hardware platforms. Porting JOP to a new FPGA board usually results in changes in the memory module alone. Using the same board for different applications only involves making changes to the I/O module. JOP has been ported to several different FPGAs and prototyping boards and has been used in different applications, but it never proved necessary to change the processor core.

3.2.2.3 The processor pipeline

JOP is a fully pipelined architecture with single cycle execution of microcode instructions and a novel approach to mapping Java bytecode to these instructions. Figure 3-8 shows the datapath for JOP [69].

Three stages form the JOP core pipeline, executing microcode instructions. An additional stage in the front of the core pipeline fetches Java bytecodes – the instructions of the JVM – and translates these bytecodes into addresses in microcode. Bytecode branches are also decoded and executed in this stage. The second pipeline stage fetches JOP instructions from the internal microcode memory and executes microcode branches. Besides the usual decode function, the third pipeline stage also generates addresses for the stack RAM. As every stack machine instruction has either pop or push characteristics, it is possible to generate fill or spill addresses for the following instruction at this stage. The last pipeline stage performs ALU operations, load, store and stack spill or fill. At the execution stage, operations are performed with the two topmost elements of the stack.

The stack architecture allows for a short pipeline, which results in short branch delays. Two branch delay slots are available after a conditional microcode branch

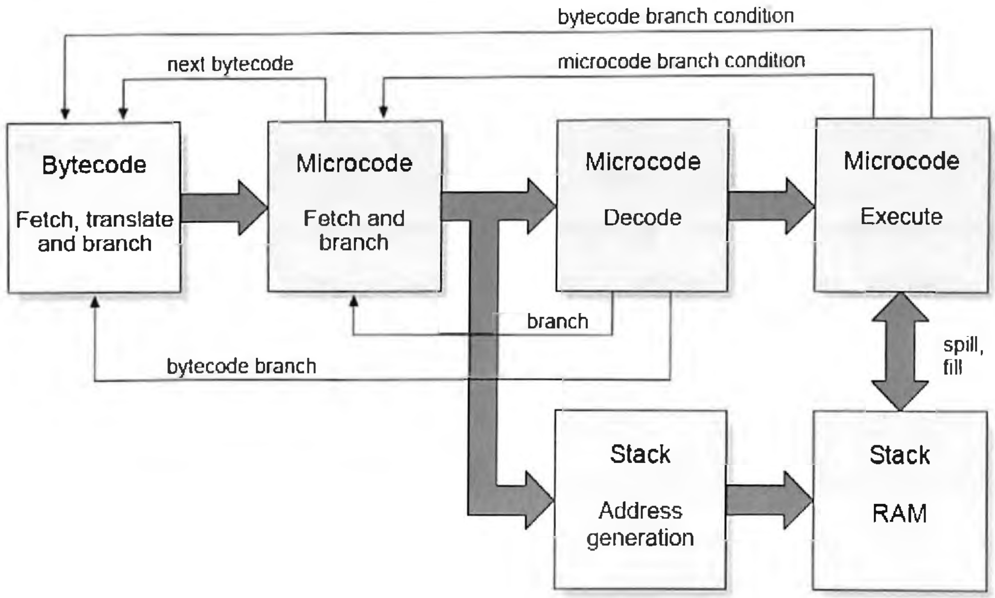


Figure 3-8: Datapath for JOP [69]

The method cache (Bytecode RAM), microcode ROM, and stack RAM are implemented with single cycle access in the FPGA's internal memories.

3.2.2.4 Cache

In order to fill the gap between processor speed and the memory access time, caches are mandatory, even in embedded systems. However, standard cache organizations improve the average execution time but are difficult to predict for WCET analysis. Two time-predictable caches are proposed for JOP: a stack cache as a substitution for the data cache and a method cache to cache the instructions.

As the stack is a heavily accessed memory region, the stack – or part of it – is placed in on-chip memory. This part of the stack is referred to as the stack cache. Fill and spill of the stack cache is subjected to microcode control and therefore time-predictable.

In JOP a novel way to organize an instruction cache, as method cache, is implemented. The cache stores complete methods, and cache misses can only occur on method invocation and return. Cache block replacement depends on the call tree, instead of instruction addresses. This method cache is easy to analyze with respect to worst-case behavior and still provides substantial performance gain when compared against a solution without an instruction cache.

3.2.2.5 Performance

Although JOP is intended as a processor for embedded real-time systems, whereas accurate worst case execution time analysis is more important than average case performance, its general performance is still important.

To provide a realistic workload for embedded systems, a real-time application (the Kippfahrleitung) was adapted to create the benchmark. Figure 3-9 shows the performance of several embedded Java systems [70]. The results are in iterations per second - a higher value means higher performance. Note that the vertical axis is logarithmic, in order to obtain useful figures to show the great variation in performance.

When comparing JOP against TINI, and muvium (a Java compiler for PIC micro-controller) we can see that a Java processor is up to 250 times faster than an interpreting or precompiled JVM on a standard processor for an embedded system. The average performance of JOP is even better than a JIT-compiler solution (+68%) on an embedded system, as represented by the EJC system (ARM720T at 74MHz). JOP is about 7 times faster than the aJ80 Java processor and about 17% faster than the aJ100 [**70**].

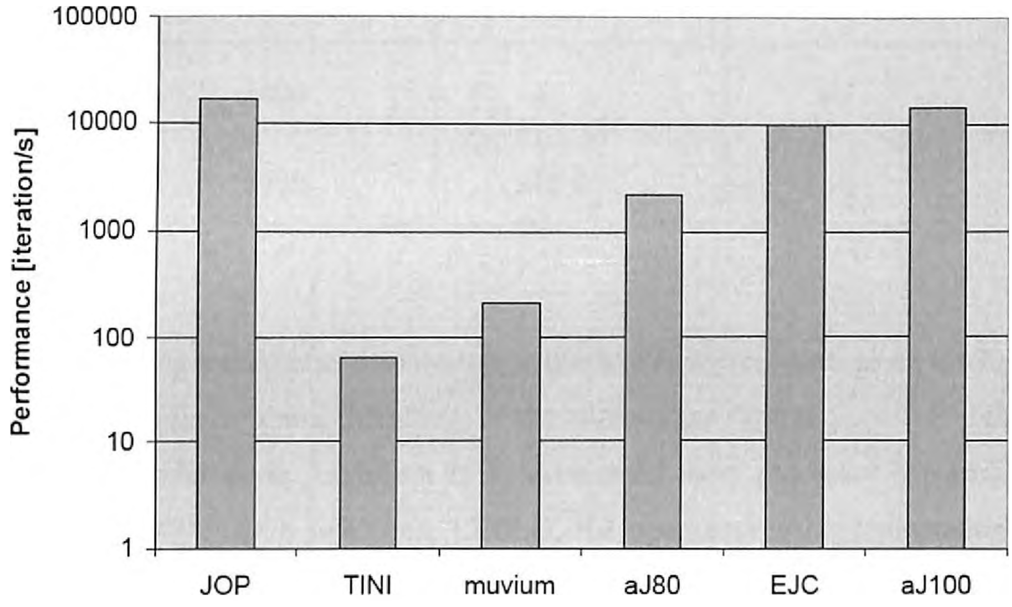


Figure 3-9: Performance Comparison of various Java Processors [70]

3.2.2.6 Size

One major design objective in the development of JOP was to create a small system that could be implemented in a lowcost FPGA. Table 3-8 shows the resource usage for different configurations of JOP and different soft-core processors implemented in an Altera Cyclone low-cost FPGA [**70**]. The size is given by the two basic structures in an FPGA: Logic Cells (LC) and embedded memory blocks.

Table 3-8: FPGA soft core processors [70]

Processor	Resources (LC)	Memory (KB)	fmax (MHz)
JOP Minimal	1077	3.25	98
JOP Typical	1831	3.25	101
Lightfoot	3400	4	40
LEON3	7978	10.9	35

The typical configuration also contains some useful I/O devices such as an UART and a timer with interrupt logic for multi-threading. In the minimal configuration shift and multiply are implemented in microcode. Lightfoot is a commercial Java processor available to be implemented in a FPGA. As a reference, LEON3, the open-source implementation if the SPARC V8 architecture, is given in the last row. Table 3-9 provides gate count estimates for JOP, picoJava (a Java processor by Sun), and the aJile Java processor. Equivalent gate count for an LC varies between 5.5 and 7.4 – we chose a factor of 6 gates per LC and 1.5 gates per memory bit for the estimated gate count [70].

Table 3-9: Gate count estimates for different processors [70]

Processor	Core (gate)	Memory (gate)	Total (gate)
JOP	11K	40K	51K
picoJava	128K	314K	442K
aJ80/aj100	25K	590K	615K

3.2.2.7 JOP Conclusions

The Java Optimized Processor (JOP) has been created by Martin Schoeberl and is continuously being developed as an open source project. JOP is a soft-core Java-processor written in VHDL which makes it possible to execute Java code on an FPGA. JOP is designed to be time predictable. and the number of Java byte codes (JVM instructions set) can be predicted in clock cycles. It is thus possible to build the real time Java applications.

The introduction of Java in embedded systems enables multithreading and object-oriented programming (OOP). It is easier to create more readable and safer coding with Java which also is platform independent. The use of JOP will facilitate the acceptance of Java for embedded systems.

Since JOP operates on an FPGA. it is easy to implement new peripheral devices to the processor. It provides the possibility of creating custom designed hardware for specifically required specifications. It also provides flexible boundaries between hardware and software.

For real time systems, the critical parts of Java code can be optimized in terms of creating customized Java-byte code which calls a set of implemented JOP-microcode instructions. The alternative would be to design new JOP-microcode instructions implemented in VHDL and executed directly in hardware.

Java combined with JOP is an interesting platform applicable to teal-time embedded systems. JOP is able to handle even hard real-time systems. JOP is deterministic platform which is predictable when using a simple architecture.

Chip multiprocessing design is an emerging trend for embedded systems. A Java multiprocessor system-on-chip has been described in [71]. It is a symmetric shared-memory multiprocessor and consists of up to 8 Java Optimized Processor (JOP) cores. an arbitration control device, and a global shared memory. All components are interconnected with a system-on-chip bus. It has been shown that even in a medium sized low-cost FPGA it is possible to run 8 cores of a Java processor in parallel.

Profiling of embedded Java applications is a tedious and time-consuming task. requiring either a simulator of the target platform or deployment of the embedded application. Simulators can be extremely slow: hence. embedded Java applications are rarely profiled in an early phase of development. Moreover. there is little support for simulating the effects of new hardware optimizations or for predicting the performance of large-scale applications. To overcome these limitations. a technique for platform-independent cross-profiling has been proposed in [72]. The cross-profiler is run in any standard Java environment (independently of hardware, operating system. and virtual machine). but generates profiles using the execution time metric of the embedded system. We call the environment running the cross-profiler the host, and we refer to the embedded system for which the profiles are collected as the target. The cross-profiler is not connected in any way to the target system.

Recent technologies such as the Real-Time Specification for Java (RTSJ) promise to bring Java's advantages to real-time systems. While these technologies have made Java more predictable, they lack a crucial element: support for determining the worst-case execution time (WCET). Without knowledge of WCET, the correct temporal behavior of a Java program cannot be guaranteed. Although considerable research has been applied to the theory of WCET analysis, implementations are much less common, particularly for Java. Recognizing this deficiency, a tool that supports WCET analysis of Java programs has been proposed in [73]. Designed for flexibility, it is built around a plugin model that allows features to be incorporated as needed. Users can plug in various processor models, loop bound detectors, and WCET analysis algorithms without having to understand or alter the tool's internals.

3.2.3 The Microcontroller based Platform

Often a microcontroller is the easiest way to start the development of a new system. This is mainly due to the fact that there are a large number of microcontrollers available at a reasonable cost and the development of small applications may be done with the tools that may be readily or freely available from the respective companies.

For the present work, two microcontrollers have been used. One is STM8 that is the 8-bit microcontroller and the other one is STM32, a 32-bit microcontroller. Both of these are from STMicroelectronics [74].

With the narrowing of the price gap between 8-bit and 32-bit controllers, designers find new flexibility on MCUs while choosing platforms. While 8-bit Micro controller units (MCUs) are preferred for cost sensitive reasons and less to medium complex applications, 32-bit MCUs are getting larger popularity on new applications. With the lowering of costs and increasing performance of 32-bit MCUs, it is further accelerating their demand.

3.2.3.1 8 bit Microcontrollers: STM8

STMicroelectronics has a new 8-bit microcontroller platform designed to offer outstanding levels of performance and cost-effectiveness in a wide range of applications. Implemented around a high-performance 8-bit core and a state-of-the-art set of peripherals. the STM8 platform is manufactured using an ST-proprietary 130nm embedded non-volatile memory technology.

STMicroelectronics' STM8S family of general-purpose 8-bit Flash microcontrollers, Figure 3-10. offers ideal solutions for industrial and appliance market requirements. The true embedded EEPROM and the calibrated RC oscillator bring significant cost effectiveness to the majority of applications. An easy-to-use and intuitive development environment contributes improving time to market [75].

The STM8 family uses a 3-stage pipeline to increase the speed of the flow of instructions sent to the processor. Pipelined execution allows several operations to be performed simultaneously. rather than serially:

- Fetch
- Decode and address
- Execute

The Program Counter (PC) points always to the instruction in decode stage as shown in Figure 3-10.

The CPU has an 8-bit architecture. Six internal registers allow efficient data manipulations. The CPU is able to execute 80 basic instructions. It features 20 addressing modes and can address six internal registers. The in-circuit programming (ICP) method is used to update the entire content of the memory, using the SWIM interface to load the user application into the microcontroller. ICP offers quick and efficient design iterations and eliminates unnecessary package handling or socketing of devices. The SWIM interface (single wire interface module) uses the SWIM pin to the programming tool [75].

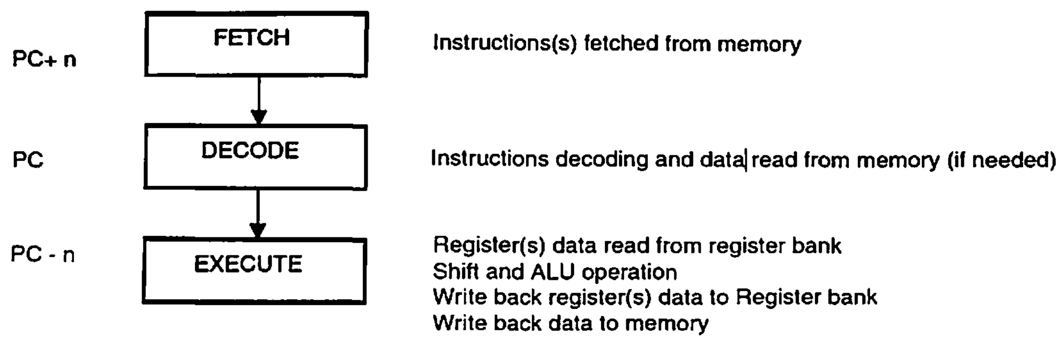


Figure 3-10: STM8S pipeline [75]

The STM8 offers various possibilities of saving power. By default, after a system or power reset, the microcontroller is in Run mode. In this mode the CPU is clocked by f_{CPU} and executes the program code, the system clocks are distributed to the active peripherals and the microcontroller is drawing full power. While in Run mode, still keeping the CPU running and executing code, the application has several ways to reduce power consumption, such as:

- Slowing down the system clocks
- Gating the clocks to individual peripherals when they are unused
- Switching off any unused analog functions

However, when the CPU does not need to be kept running, three dedicated low power modes can be used:

- Wait
- Active-halt (configurable for slow or fast wakeup)
- Halt (configurable for slow or fast wakeup)

One of these three modes can be selected and configured to obtain the best compromise between lowest power consumption, fastest start-up time and available wakeup sources.

STMicroelectronics has introduced its first ultra-low-power family based on the 8-bit STM8 core, called STM8L. Paving the way for a future complete ultra-low-power platform, the STM8L family combines high performance and ultra-low power consumption thanks to a new proprietary ultra-low-leakage process and optimized architecture.

The versatile STM32 based platform has been used extensively in this work and is treated separately in the next section of this chapter.

3.2.3.2 STM32: A versatile 32 bit Microcontroller

With a wide range of MCUs available today, it becomes easy for developers to choose a platform based on their application requirements. The investments on the knowhow and tools are minimized and newer designs can come much faster in the market.

The STM32 MCUs are based on popular Cortex-M3 [76] core and there are a range of products to offer based on this core.

3.2.3.2.1 ARM Cortex-M3

The ARM Cortex-M3 processor is a low-power processor that features low gate count, low interrupt latency, and low-cost debug. It is intended for deeply embedded applications that require fast interrupt response features. The processor implements the ARM architecture v7-M.

The processor incorporates, Figure 3-11, [76]:

- Processor core. A low gate count core, with low latency interrupt processing that features:
    - o ARMv7-M. A Thumb®-2 Instruction Set Architecture (ISA) subset, consisting of all base Thumb-2 instructions, 16-bit and 32-bit, and excluding blocks for media, Single Instruction Multiple Data (SIMD), enhanced Digital Signal Processor (DSP) instructions (E variants), and ARM system access.
    - o Banked Stack Pointer (SP) only.

- Hardware divide instructions, SDIV and UDIV (Thumb-2 instructions).
- Handler and Thread modes.
- Thumb and Debug states.
- Interruptible-continued LDM/STM, PUSH/POP for low interrupt latency.
- Automatic processor state saving and restoration for low latency Interrupt Service Routine (ISR) entry and exit.
- ARM architecture v6 style BE8/LE support.
- ARMv6 unaligned accesses.

- Nested Vectored Interrupt Controller (NVIC) closely integrated with the processor core to achieve low latency interrupt processing. Features include:
  - External interrupts of 1 to 240 configurable size.
  - Bits of priority of 3 to 8 configurable size.
  - Dynamic reprioritization of interrupts.
  - Priority grouping. This enables selection of pre-empting interrupt levels and non pre-empting interrupt levels.
  - Support for tail-chaining and late arrival of interrupts. This enables back-to-back interrupt processing without the overhead of state saving and restoration between interrupts.
  - Processor state automatically saved on interrupt entry, and restored on interrupt exit, with no instruction overhead.
- Memory Protection Unit (MPU). An optional MPU for memory protection:
  - Eight memory regions.

- Sub Region Disable (SRD), enabling efficient use of memory regions.
- A background region can be enabled that implements the default memory map attributes.

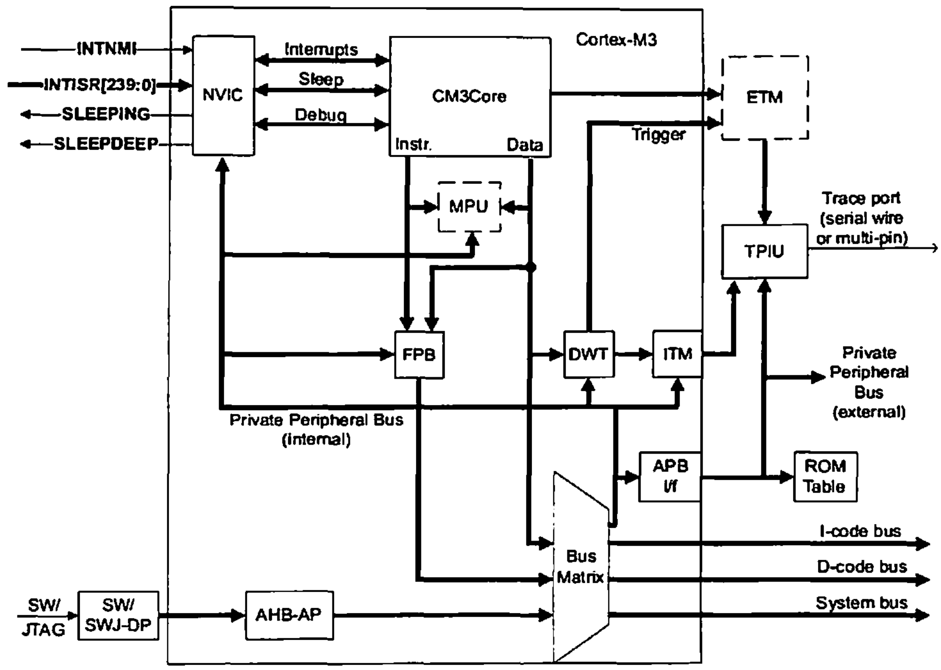


Figure 3-11 Cortex M3 Block Diagram [76]

- Bus interfaces:
  - Advanced High-performance Bus-Lite (AHB-Lite) ICode, DCode and System bus interfaces.
  - Advanced Peripheral Bus (APB) and Private Peripheral Bus (PPB) Interface.
  - Bit band support that includes atomic bit band write and read operations.
  - Memory access alignment.
  - Write buffer for buffering of write data.
- Low-cost debug solution that features:

- Debug access to all memory and registers in the system, including Cortex-M3 register bank when the core is running, halted, or held in reset.
- Serial Wire Debug Port (SW-DP) or Serial Wire JTAG Debug Port (SWJ-DP) debug access, or both.
- Flash Patch and Breakpoint (FPB) unit for implementing breakpoints and code patches.
- Data Watchpoint and Trace (DWT) unit for implementing watchpoints, data tracing, and system profiling.
- Instrumentation Trace Macrocell (ITM) to support debugging.
- Trace Port Interface Unit (TPIU) for bridging to a Trace Port Analyzer (TPA).
- Optional Embedded Trace Macrocell (ETM) for instruction trace.

Figure 3-11 shows that the processor components exist in two levels of hierarchy. This represents the RTL hierarchy of the design. Four components, ETM, TPIU, SW/SWJ-DP, and ROM table, exist outside the Cortex-M3 level because these components are either optional, or there is flexibility in their implementation and use.

3.2.3.2.2 STM32: Cortex-M3 core based Microcontroller

STM32 is a 32 bit microcontroller from STMicroelectronics that is based on Coretx M3 core. STM32 family consists of several variants with options of varying RAM, flash, pin count and other specific interfaces for applications like camera interface, motor control and so on. Architecture of one of its sub-family, called high-density value line is shown in Figure 3-12. In high-density value line devices, the main system consists of:

- Four masters:
  - Cortex™-M3 core DCode bus (D-bus) and System bus (S-bus)
  - GP-DMA1 & 2 (general-purpose DMA)

- Four slaves:
  - Internal SRAM
  - Internal Flash memory
  - FSMC
  - AHB to APB bridges (AHB to APBx), which connect all the APB peripherals

The ICode bus connects the instruction bus of the Cortex™-M3 core to the Flash memory instruction interface. Instruction fetches are performed on this bus. The DCode bus connects the Cortex™-M3 core to the Flash memory data interface. The System bus connects the system bus of the Cortex™-M3 core (peripherals bus) to a bus matrix which manages the arbitration between the core and the DMA. The DMA bus connects the AHB master interface of the DMA to the bus matrix which manages the access of CPU DCode and DMA to the SRAM, Flash memory and peripherals. The bus matrix manages the access arbitration between the core system bus and the DMA master bus. The arbitration uses a round robin algorithm. The bus matrix is composed of four masters (CPU DCode, System bus, DMA1 bus and DMA2 bus) and four slaves (FLITF, SRAM, FSMC and AHB to APB bridges). AHB peripherals are connected to the system bus through the bus matrix to allow DMA access. The two AHB/APB bridges provide full synchronous connections between the AHB and the two APB buses. APB buses operate at full speed (up to 24 MHz). After each device reset, all peripheral clocks are disabled.

Program memory, data memory, registers and I/O ports are organized within the same linear 4-Gbyte address space. The bytes are coded in memory in little endian format. The lowest numbered byte in a word is considered the word's least significant byte and the highest numbered byte, the most significant. The addressable memory space is divided into 8 main blocks, each of 512 MB.

The device requires a 2.0-to-3.6 V operating voltage supply (VDD). An embedded regulator is used to supply the internal 1.8 V digital power.

Direct memory access (DMA) is used in order to provide high-speed data transfer between peripherals and memory as well as memory to memory. Data can be quickly moved by DMA without any CPU actions. This keeps CPU resources free for other operations. The two DMA controllers have 12 channels in total (7 for DMA1 and 5 for DMA2), each dedicated to managing memory access requests from one or more peripherals. It has an arbiter for handling the priority between DMA requests.

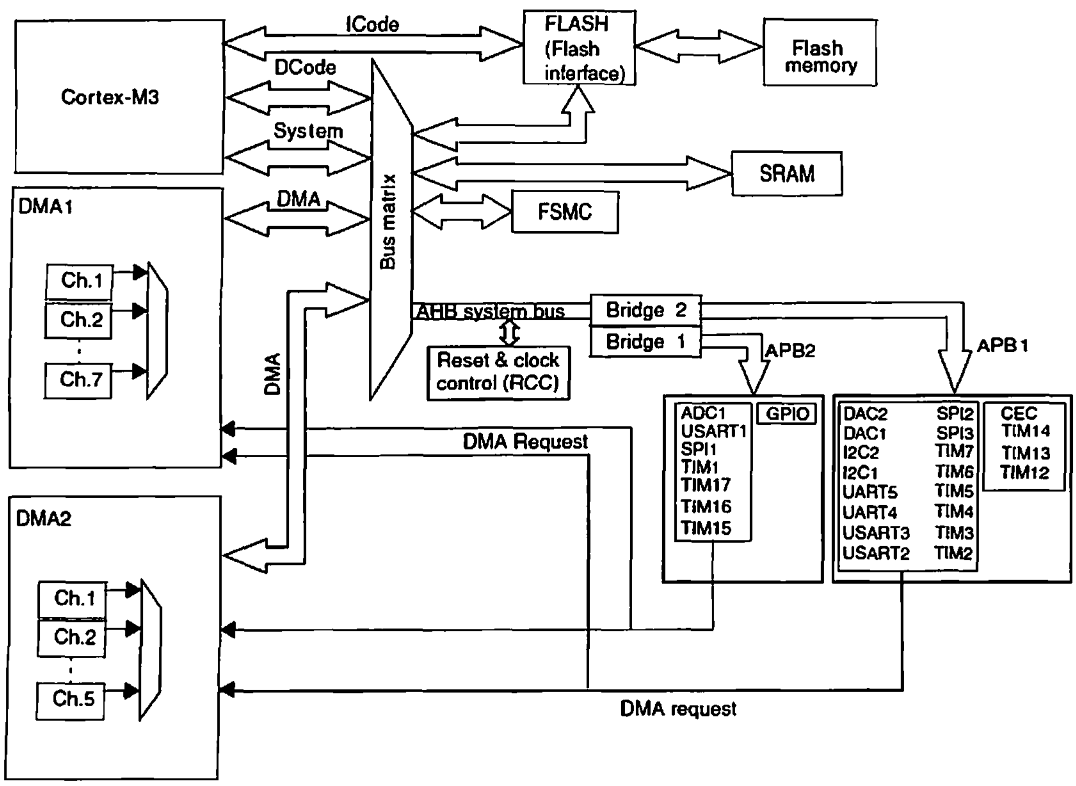


Figure 3-12 STM32 System Architecture

The 12-bit ADC is a successive approximation analog-to-digital converter. It has up to 18 multiplexed channels allowing it to measure signals from 16 external and two internal sources. A/D conversion of the various channels can be performed in single, continuous, scan or discontinuous mode. The result of the ADC is stored in a left-aligned or right-aligned 16-bit data register. The analog watchdog feature allows the application to detect if the input voltage goes outside the user-defined high or low thresholds.

The DAC module is a 12-bit, voltage output digital-to-analog converter. The DAC can be configured in 8- or 12-bit mode and may be used in conjunction with the DMA controller. In 12-bit mode, the data could be left- or right-aligned. The DAC has two output channels, each with its own converter. In dual DAC channel mode, conversions could be done independently or simultaneously when both channels are grouped together for synchronous update operations.

The STM32F100xx are built around a Cortex™-M3 core which contains hardware extensions for advanced debugging features. The debug extensions allow the core to be stopped either on a given instruction fetch (breakpoint) or data access (watchpoint). When stopped, the core's internal state and the system's external state may be examined. Once examination is complete, the core and the system may be restored and program execution resumed. The debug features are used by the debugger host when connecting to and debugging the STM32F100xx MCUs. Two interfaces for debug are available:

- Serial wire
- JTAG debug port

The STM32 family benefits from the Cortex-M3 architectural enhancements including the Thumb-2 instruction set to deliver improved performance with better code density, significantly faster response to interrupts, all combined with industry leading power consumption. The STM32 family is built to offer new degrees of freedom to MCU users. It offers a complete 32-bit product range that combines high-performance, real-time, low-power and low-voltage operation, while maintaining full integration and ease of development. Compatibility of pin-assignments, peripherals and software across all STM32 devices is a core technical feature throughout this family of microcontrollers [77].

3.2.3.2.3 STM32W: STM32 with ZigBee

The STM32 family is further divided into three categories: STM32F, STM32L and STM32W. The STM32F is the foundation of the STM32 family. It provides perfectly balanced products for applications that do not require extreme low power. They combine high performance with first-class peripherals and low-power, low-voltage operation. The STM32F products target a wide range of applications in the industrial, medical and consumer markets.

The STM32L MCU family, extends the ultra-low-power portfolio in performance, features, memory size and package pin count. The STM32L family combines very high performance and ultra-low power consumption, as a result of ST's proprietary ultra-low leakage process and optimized architecture.

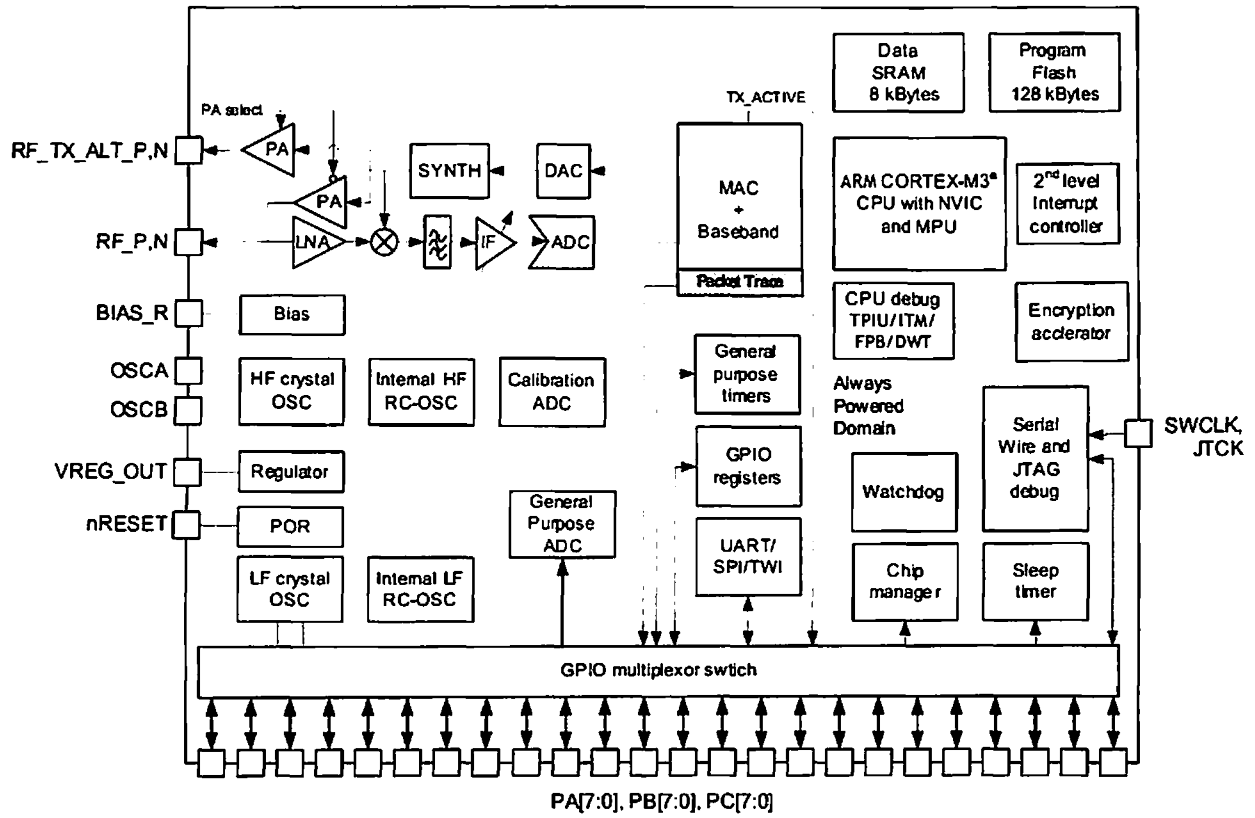


Figure 3-13: STM32W Block Diagram [78]

The STM32W, Figure 3-13, is the wireless member of STM32 family and brings outstanding radio and low-power microcontroller performances. With a configurable total link budget up to 109 dB and the efficiency of the ARM Cortex-M3 core, the STM32W is a perfect fit for the wireless sensor network market. Compliant with the IEEE 802.15.4 radio standard, this open and flexible platform supports the most popular protocol stacks such as RF4CE, ZigBee-PRO, 6LoWPAN and more.

The STM32W, a member of STM32 family includes an on-chip 802.15.4 compliant radio. Its key features are:

- 32-bit ARM Cortex-M3 core running @ 24 MHz
- 128-Kbyte Flash, 8-Kbyte RAM
- Fully IEEE 802.15.4 compliant radio @ 2.4 GHz
- Power management
- Deep sleep mode <1 μA with RAM retention

- On-chip debug support ARM JTAG/SWD
- Packet trace interface enables remote monitoring of radio messages
- ARM memory protection unit to detect erroneous software accesses
- Sleep timer, watchdog timer and GP timers
- AES 128 encryption acceleration
- Serial communication (UART/SPI/I^2C)
- ADC (6 channels, first order sigma delta)

3.3 Java for Gaming

The operating environments for gaming can be divided into two main categories. When computer monitor is used for display, we have the liberty of making use of all the resources of the computer and using the handheld devices for control.

On the other hand when the gadgets interact with each other in standalone mode or per-to-peer mode, usually these are accompanied by a display also. In such cases the operating systems have more stringent requirements and less versatility compared to operating environment available on the computer.

The Java has become popular for mobile gaming, owning to its availability and ease of development. The original idea was to develop a language in which you would write your code once, and then it would run on any platform supporting a Java Virtual Machine.

Java has extended its reach far beyond desktop machines. Two years after the introduction of Java, a new edition was released, Java 2 Enterprise Edition, providing support for large-scale, enterprise-wide applications. The Micro Edition targeted "information appliances," ranging from Internet-enabled TV set-top boxes to cellular phones [79].

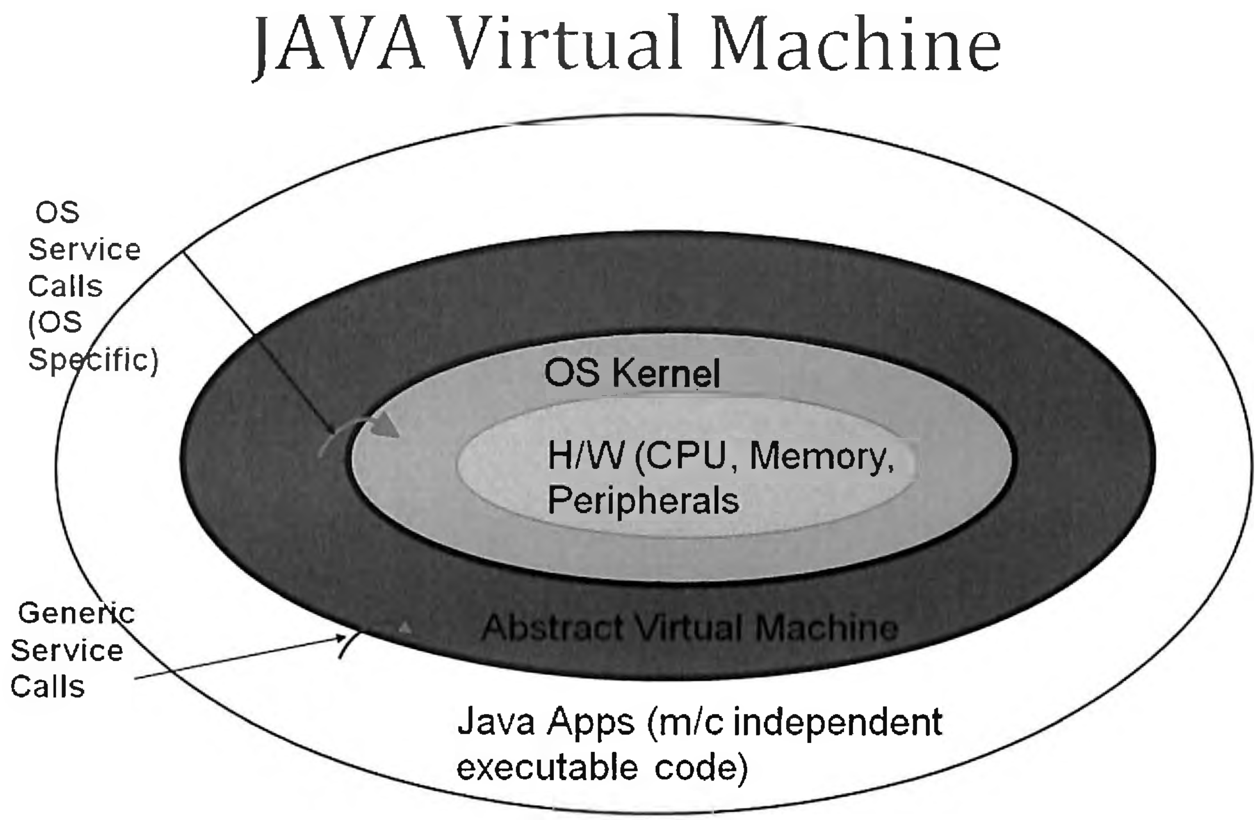


Figure 3-14: The Java Virtual Machine and its relation with OS and hardware

For the embedded systems, Java Virtual Machine (JVM) is of significance. A Java Virtual Machine (JVM) enables a set of computer software programs and data structures to use a virtual machine model for the execution of other computer programs and scripts. The model used by a JVM accepts a form of computer intermediate language commonly referred to as Java bytecode. Figure 3-14 shows the JVM and its relation to OS and other hardware.

Real-Time JVM guarantees deterministic run-time behaviour such as a real-time garbage collector. This JVM would only be implementable atop suitable target platforms, such as a real-time operating system (RTOS).

3.3.1 J2ME for Consumer Devices

J2ME is aimed at consumer devices with limited computing capabilities. Many such devices (e.g., a mobile phone or pager) have no option to download and install software beyond what was configured during the manufacturing process. With the introduction of J2ME, "micro" devices no longer need to be "static" in nature. Not unlike a web browser downloading Java applets, an implementation of J2ME on a device affords the option to browse, download and install Java applications and content.

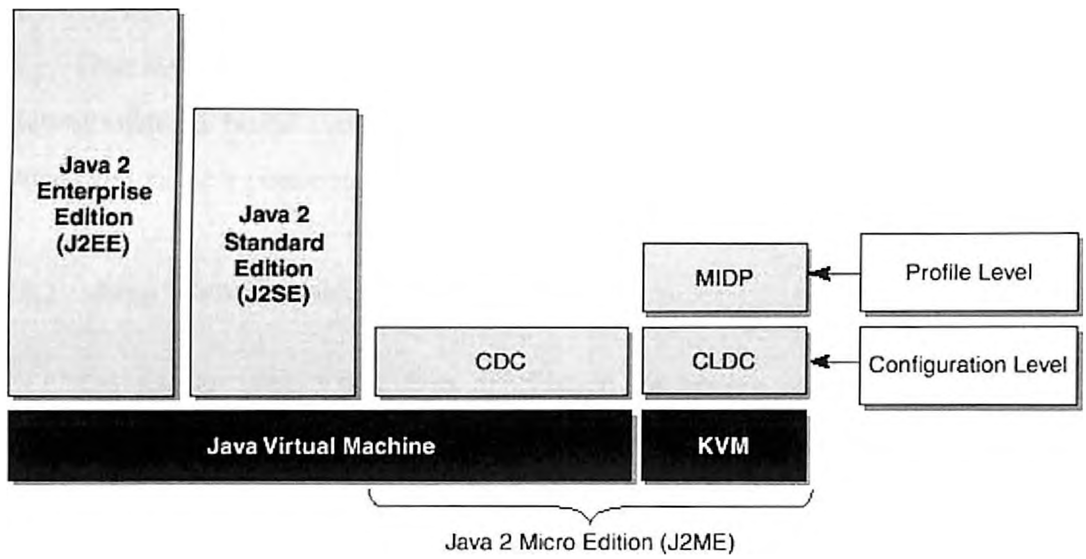


Figure 3-15 Various Java Editions and J2ME [80]

Small consumer electronics have a way of changing our lives. Mobile phones let us communicate when away from our home or office. Personal digital assistants (PDAs) let us access email, browse the internet and run applications of all shapes and forms. With the introduction of Java for such devices, we now have access to the features inherent to the Java language and platform. That is, a programming language that is easy to master, a runtime environment that provides a secure and portable platform and access to dynamic content, not to mention an estimated developer community of over 2 million people.

Although it would be nice to have the entire J2SE Application Programming Interface (API) available on a micro device, it's not realistic. For example, a mobile phone with its limited display cannot provide all the functionality available in the Abstract Window Toolkit, the first graphical user interface released with Java. The "Micro Edition" was introduced to address the special needs of consumer devices that are outside the scope of J2SE and J2EE.

The capabilities of devices within the "Micro Edition" may vary greatly. An Internet Screenphone (a hardware device designed to provide access to email. news. online banking. etc.) may have a much larger display than a pager. However. even devices that seem similar in size may vary greatly in their capabilities. A cell phone and PDA are both limited in physical size. yet a typical cell phone may have a display with a total resolution of 12.288 pixels (96 x 128). whereas a PDA resolution may start at 20.000 pixels and go up from there.

One Java platform will most definitely not fit all. To better understand how J2ME will accommodate a broad range of consumer electronics and embedded devices. we need to introduce two new concepts. configurations and profiles.

3.3.2 Java Virtual Machines

The engine behind any Java application (or applet, servlet, etc.) is the JVM, Figure 3-14. Once Java source code is compiled into a class file. and optionally included them in a Java Archive (JAR) file. the JVM translates the class files into machine code for the platform running the JVM. The JVM is also responsible for providing security. allocating and freeing memory and managing threads of execution. It's what makes your Java programs go. so to speak.

For CDC, the virtual machine has the same specification as J2SE. For CLDC. Sun has developed what is referred to as a reference implementation of a virtual machine. known as the K Virtual Machine. or simply KVM. This virtual machine was designed to handle the special considerations of resource-constrained devices. It's clear the KVM is not the "traditional" Java virtual machine [79]:

- The virtual machine itself requires only 40 and 80 kilobytes of memory
- Only 20 - 40 kilobytes of dynamic memory (heap) are required
- Can run on 16-bit processors clocked at only 25 MHz

3.3.3 The J2ME Mobile Information Device Profile 2.0

Here we will speak about Sun Microsystems's J2ME platform. the Mobile Information Device Profile (MIDP) version 2.0. The new MIDP 2.0 includes a large set of new features;

some important for business-class applications, while the majority of these new features are mainly targeted at the consumer space [81].

The previous version of MIDP (version 1.0) already provided the foundation needed for business-type applications including HTTP-based network connectivity, local persistence, the ability to operate in disconnected mode, and the ability to write smart applications using Java. Lacking was support for end-to-end security, wide adoption of MIDP by wireless network providers, and attractive MIDP-based devices — this resulted in a lack of useful business applications.

With MIDP 2.0, end-to-end security is now supported. MIDP 2.0 introduces the security concepts of application signing and privileged domains. With application signing applications can be trusted or not based on the ability to corroborate, via the use of a X.509 digital certificate, the application's origin and integrity. And through privileged domains, defined via a policy definition, device vendors and wireless provides can define which APIs are considered restricted. These new features protect the device against unauthorized applications accessing data and functions. Also standard security protocols such as HTTPS, TLS/SSL and WTLS allow for secure transmission by encrypting the data. It is important to understand that MIDP in general is targeted at cellphones and low-end PDAs; the kind of devices that are personal in nature and that are increasingly becoming business tools. Thus IT managers must ensure their employees are accessing corporate data in a safe manner, which is why the new security features found in MIDP 2.0 are welcome.

The new MIDP 2.0 also includes support for an enhanced user interface API that allows developers to create more functional and attractive business or consumer applications with less effort. It also includes new gaming and media APIs that can be used for business-applications but that are more targeted at the consumer space.

Applicable to both business and consumer applications, MIDP 2.0 has expanded network connectivity support beyond HTTP, with support for UDP datagrams, TCP sockets, and serial port communication, and as mentioned before secure connections. Also new is a powerful feature called the push registry that activates dormant applications when new information is available. For example, with push support, server software can activate an application on the device to notify employees when a new calendar event has been scheduled or a new sales lead generated, even if the application is not currently running.

The new Over-The-Air (OTA) application provisioning standardizes support for client-initiated download of applications over the wireless network. It allows users to find and install authorized applications wirelessly. It also allows application providers to track the installation and removal of applications for billing (or other) purposes. With OTA, the IT department can enforce or ensure that the right application or application version are easily available and in use.

3.3.4 Developing Java-Based Mobile Games

In recent times, mobile games have gained popularity for providing personal entertainment on the go. This popularity has mobile gaming playing a pivotal role in revenue generation for the cellular carriers, game publishers, and handset makers, while generating numerous opportunities for game developers and associated professionals. With the number of mobile gamers around the world expected to reach 220 million by 2009, the mobile gaming business is projected to expand to higher levels and constitute a bigger portion of the profit pie for the cellular carriers and handset makers.

Mobile games can be classified into three broad categories [**38**]:

- Embedded games: Games that are hardcoded into the mobile handset's system and shipped with it. These will likely become outdated/obsolete in near future. Example: Snake, shipped with some basic feature phones.

- SMS games: Games played by sending text messages—for example, SMS to game server—that process them and sends back the result through SMS. Often in the form of live contests and polls. Not very popular because the cost of gaming increases with each SMS sent to the game server.

- Browser games: These games are played using mobile phone's built-in micro browser (net browser for mobile devices), either in online or offline mode. Players can play such games online through their cellular carrier's or a third-party game provider's game Web site, or download them for offline gaming. This category includes a wide range of games, such as solo or multiplayer games, network games, offline games, arcade games, and so forth

Mobile games can be developed using C++. Java (Java 2 Micro Edition. to be precise). and the Binary Runtime Environment for Wireless (BREW) platform from Qualcomm.

3.3.5 Advantages of Java for Mobile Gaming

Although C++ has the advantage of being compiled into native code with direct access to system resources. and with BREW the platform provides end-to-end solutions to mobile game developers while allowing them to work with any desired language (including C++. Java. XML. and Flash). Java is the most popular choice for game development. In general. any J2ME-enabled device. regardless of its operating system. can run Java code. including that used in games. Another advantage of using J2ME to build games is that game companies have access to the many experienced Java developers **[82]**. Java. or the Java 2 Micro Edition (J2ME) platform to be precise, is identified as the most convenient for developing mobile games. The driving forces behind J2ME's popularity are **[83]**:

- J2ME enjoys the status of an industry standard backed by all major handset makers, with most of the present day mobile phones being Java-enabled.
- J2ME is a free and open platform. This helps keep the development costs low and provides the necessary flexibility with ample support freely available for developers using it.
- Its highly portable nature ("Write once run anywhere") ensures that a game application written for one brand/type of handset will work with all other brands/types of Java-enabled handsets.
- It is especially optimized for small devices, is lightweight, and is highly secure because applications written on it cannot access or affect other applications running on the phone/device.

J2ME consists of the Mobile Information Device Profile (MIDP) API that is designed specifically for developing applications for mobile devices including mobile phones. keeping in mind their limitations and constraints. Furthermore. the latest MIDP version 2.0 itself dedicates a whole API to game development. making game development simpler and quicker.

3.3.6 Real Time Specifications for Java

It should be mentioned here that Java had initial limitations to address the real time systems appropriately. These limitations resulted in the slow adoption of Java usage for real time systems. These limitations were [84], [85] :

- The garbage collected dynamic memory management could interrupt the execution of applications for unpredictable intervals of time. This originates from the non-deterministic behaviour of the garbage collector, because it chooses when to de-allocate objects from memory heap and in no particular order.

- Thread scheduling. Since the JVM (Java Virtual Machine) relies on the host operating systems scheduler, then the operating system must be capable of real-time scheduling. This is not true for all hosts where JVM is available.

It is vital for critical mission real-time control systems software as in a Nuclear Power Plant, Aircraft Control, Submarine Control, Factory Automation, Airport Aviation Flight Control, Energy and Power Systems Supply, Telecommunication Satellite and so forth to have a deterministic scheduling of events not in an indeterminate state. The risks associated with writing such critical mission real-time software in Java far outweighs the other benefits of Java.

A requirements group was created to address these limitations in 1998. It was coordinated by the National Institute of Standards and Technology (NIST) to draft requirements for real-time Java [86]. The members were representatives from 50 different companies, government institutes and researchers from academic institutions. The main aim of the group was to develop a cross-disciplinary specification for real-time functionality that was expected to be needed by real-time applications written in Java programming language and being executed on various platforms. The group workshops at NIST produced nine core requirements for a Real-Time Java specification, together with number of derived sub-core requirements. This is the basis of the Real-Time Specification for Java (RTSJ). The weaknesses of the Java language specification for writing real-time applications are addressed in the core requirements.

The guiding principles for RTSJ are [86]:

- No restriction to use a particular Java environment
- Backward compatibility
- Write once run everywhere
- Address real time issues
- Predictable execution
- No new keywords
- Allow flexibility of implementation

3.3.7 The Role of MIDP 2.0 in Game Development

MIDP 2.0 API is a set of feature-loaded APIs used for developing secure, rich-content multimedia applications, including games, for mobile devices. MIDP 2.0 builds upon its predecessor MIDP 1.0 to provide a better development platform for building efficient and fast mobile applications. For more information on MIDP 2.0, see the Resources section at the end of this article **[87]**.

MIDP 2.0 further refines the features and functionalities provided by MIDP 1.0. Through the Game API, MIDP 2.0 provides game developers with the readymade building blocks that were to be developed from scratch in the case of MIDP 1.0. These building blocks are classes for creating and controlling various game elements such as game canvas, sprites, layers, and so forth (these are explained in the next section). Thus, MIDP 2.0 significantly reduces the time involved in game development.

From the above discussions it is clear that J2ME offers a very promising environment for the games. The games are developed on the computer and downloaded to the mobile or the handheld unit that has the Java compatible processor. Several Microcontrollers are having built-in support for Java. Even in case of FPGAs, a Java processor like JOP can be created where the Java games developed on the computer can be downloaded on to the target.

3.4 Other Gaming Environments : HTML5

In section 3.3 Java has been presented as an environment for gaming. There are other options for the gaming environments. HTML5 is one of options and is described briefly in this section. Flash Player by Adobe is another possibility that is not included in the discussions.

Traditionally the role of HTML has been used primarily to define and layout the text on web pages. Over the years, however, its role has slowly evolved to be more interactive. The new HTML5 spec furthers this evolution by including definitions for several additional elements aimed at interactivity, multimedia, and graphics [**88**].

Combining these new HTML5 elements with the faster and enhanced JavaScript engines available in modern browsers results into a set of tools well suited for online game development.

Not only are the HTML5 technologies well suited for creating browser based games, they offer several advantages over proprietary options. HTML5 is an open technology. Compatibility is another advantage HTML5 has. HTML is the lowest common denominator for all web based devices. By using HTML5, a wider array of devices and gadgets can be targeted with no additional porting each time a platform is changed. Another advantage is ease of use [**88**].

HTML has been designed as a markup language to create web pages so that it can render in a web browser. HTML5 is the next revision of HTML. The generalized HTML5, includes new and enhanced HTML, CSS3, JavaScript API and a technical portfolio of events.

The main features of HTML5 in addition to standard HTML are [**89**]:

- Improve optimization page elements
- Form
- Using the <canvas> element
- WebSocket
- Local Storage
- Information transfer between page
- Video and Audio (timered media player)

- Location (Communication APIs)
- Micro Data

The realization of the online real-time acquisition system is benefits from the Canvas and WebSocket HTML5 API functions.

Canvas was initially introduced by Apple for use inside their own Mac OS X WebKit component in 2004. Canvas consists of a drawable region defined in HTML code with height and width attributes. JavaScript code may access the area through a full set of drawing functions similar to those of other common 2D APIs, thus allowing for dynamically generated graphics. Some anticipated uses of canvas include building graphs, animations, games, and image composition.

WebSocket is designed to be implemented in web browsers and web servers, but it can be used by any client or server application. The WebSocket Protocol is an independent TCP-based protocol. Its only relationship to HTTP is that its handshake is interpreted by HTTP servers as an Upgrade request.

3.4.1 HTML5 Develpoment Environments

With the growing usage of HTML5, existing development environments (SDKs) have been modified to provide support to it. On the handheld environments also there is an effort to provide support to HTML5 by the device manufacturers.

As an example Intel XDK (Cross Platform Development Kit) is a HTML5 cross-platform solution enables developers to write web and hybrid apps once, and deploy across many app stores and form factor devices.

The XDK offers a comprehensive simulation and debugging capability like any other mobile development system. Apps can be rapidly tested during development using the XDK's built-in emulator, or can be tested on the actual device using WiFi and AppLab testing apps [90].

The XDK by Intel is the world's first cloud-powered mobile application development environment. It has advantages of a single and always up-to-date development system. Using industry-standard HTML5 and CSS, animation, effects, styling and video for enhanced interactivity can be included.

The key features are:

•Easy-to-use: Streamlined workflow from design to app store

•Develop faster: Integrated design, test, and build tools

•Deploy simply: Across more app stores, and form factors

Quite like Java *Write-Once-Run-Everywhere*, XDK has the philosophy of using the same development to many target platforms.

HTML5 standards are under definition and last version by W3C was updated in June 2014 **[91]**.

3.5 Energy Efficiency for Gaming

In the preceding sections, wireless protocols, different options for processors, operating environments were discussed. A system needs efficient auxiliary circuitry as well in order to be energy efficient. The energy efficiency itself can result from efficient usage of the energy available, choice of right protocols, processors and devices.

The embedded systems, including gaming devices can greatly benefit from energy harvesting. Some of the energy harvesting methods available are solar cells, piezoelectric, vibration and thermoelectric. The general properties to be considered to characterize a portable energy supplier are electrical properties such as power density, maximum voltage and current; physical properties such as the size, shape and weight; environmental properties such as water resistance and operating temperature range; as well as operational and maintenance properties. Sufficient care should be taken while using the energy harvesters in the embedded systems to improve the performance and lifetime of the system **[12]**.

When a device is subjected to vibration, an inertial mass can be used to create movement. This movement can be converted to electrical energy using three mechanisms: piezoelectric, electrostatic and electromagnetic. The form of energy utilized here is the mechanical energy.

Thermoelectric generators follow the principle of thermoelectricity to produce the required electrical energy. The phenomena of creating electric potential with a temperature difference and vice-versa can be termed as thermoelectricity. Here. the thermal energy is scavenged to obtain electrical energy to power the electronic devices. Thermoelectric devices are primarily used in space and terrestrial applications.

One of the ways to optimize energy efficiency is to optimize the middleware for the wireless technology used. Where it is possible. the handheld devices may be supplied with solar power or energy harvesting from other sources like movement could be used.

3.5.1 Power Aware Middleware

With the improved processing power. graphic quality and high-speed wireless connection in recent generations of mobile phone. it looks more attractive than ever to introduce networked games on these devices. While device features and application resource requirements are rapidly growing. the battery technologies are not growing at the same pace. Networked Mobile games are a class of application. which consume higher levels of energy. as they are naturally more computationally intensive and use hardware components including audio. display and network to their fullest capacities. Therefore. the main concern is the limitation of the battery power of such portable devices to support the potentially long-hour of game play [92].

ARIVU [92] is presented as environment to focus on reducing the energy consumption of the wireless network interfaces. The key mechanism to reduce the power consumption of wireless interfaces is to put the interface into sleep mode whenever possible. The challenge is to do this without affecting the game play.

The amount of power saved ranges from 40 to 35%. The percentage of power saved is shown to be 20-30% even in case of number of players going up to 15.

Power Aware Game transport (PGPT) has been presented in [93]. Applications on the smartphones are able to capitalize on the increasingly advanced hardware to provide a user experience reasonably impressive. However. the advancement of these applications is hindered by the battery lifetime of the smartphones. The battery technologies have a relatively low growth rate. Applications like mobile multiplayer games are especially power hungry as they maximize the use of the network. display and CPU resources. The PGTP is aware of both the transport requirement of these multiplayer mobile games and the limitation posed by battery resource. PGTP dynamically controls the transport based on the criticality of game state and the network state to save energy at the wireless network interface level with almost no degradation to the quality of the game play.

Using PGPT it has been shown that with 8 users of different levels of experience in games. the quality drop is almost not noticeable. The PGPT shows a saving of up to 40% of the power consumed by wireless network interface. Power saved is slightly sensitive to player density (number of players in a given game map).

The niche application area which PGTP addresses is networked real-time interactive applications in battery operated mobile devices that have small payload sizes for transmission and do not require reliable transmission for most of their packets [**93**].

3.5.2 Photovoltaic

Photovoltaic is the direct conversion of light into electricity at the atomic level. Some materials exhibit a property known as the photoelectric effect that causes them to absorb photons of light and release electrons. When these free electrons are captured, an electric current is produced that can be used as electricity [**94**].

The photoelectric effect was first noted by a French physicist, Edmund Bequerel, in 1839, who found that certain materials would produce small amounts of electric current when exposed to light. In 1905. Albert Einstein described the nature of light and the photoelectric effect on which photovoltaic technology is based, for which he later won a Nobel Prize in physics. The first photovoltaic module was built by Bell Laboratories in 1954. It was billed as a solar battery and was mostly just a curiosity as it was too expensive to gain widespread use. In the 1960s, the space industry began to make the first serious use of the technology to provide power aboard spacecraft. Through the space programs, the technology advanced, its reliability was established. and the cost began to decline. During the energy crisis in the 1970s, photovoltaic technology gained recognition as a source of power for non-space applications.

Solar cells are made of the same kinds of semiconductor materials, such as silicon, used in the microelectronics industry. For solar cells. a thin semiconductor wafer is specially treated to form an electric field, positive on one side and negative on the other. When light energy strikes the solar cell, electrons are knocked loose from the atoms in the semiconductor material. If electrical conductors are attached to the positive and negative sides, forming an electrical circuit, the electrons can be captured in the form of an electric current, that is, electricity. This electricity can then be used to power a load, such as a light or a tool.

A number of solar cells electrically connected to each other and mounted in a support structure or frame is called a photovoltaic module [95]. Modules are designed to supply electricity at a certain voltage, such as a common 12 volts system. The current produced is directly dependent on how much light strikes the module.

Multiple modules can be wired together to form an array. In general, the larger the area of a module or array, the more electricity will be produced. Photovoltaic modules and arrays produce direct-current (dc) electricity. They can be connected in both series and parallel electrical arrangements to produce any required voltage and current combination.

The solar energy is representing a viable alternative or an additional supplement to the existing sources of energy. The solar power is available to the most parts of the world. Regardless of whether it is used, this power is always incident upon the earth. This calls in for the engineering solutions that could harness the available power, helping the mankind in its survival fight for the energy.

Electromagnetic energy transmitted from the sun is referred to as solar energy. Note that technically speaking power is energy per unit time, although casually both terms are used interchangeably. Only a very small portion of total power radiated from the sun reaches the Earth. Numerically, this amount is characterized by solar constant- the average amount of sun radiation that reaches the earth's upper atmosphere on a surface perpendicular to the sun's rays.

Part of sun's radiation is absorbed and scattered by the earth's atmosphere. By the time sunlight reaches Earth's surface, its peak power drops to about 1,000 watts in average per square meter at noon on a bright day. The commercially available panels for homes deliver efficiencies between 12 to 20%. This means that peak output of a solar power system at noon can be up to 120-200 watt per square meter, or 11 to 18 watt per square foot. Additional reflective devices, such as troughs or mirror panels can be used to concentrate more sun's energy, but they are normally used only in commercial applications. Solar constant is approximately 1360-1367 watts per square meter. Sun's radiation can be used to produce heat and electricity.

Solar power can be converted directly to electricity in photovoltaic (PV) cells, which are casually called solar cells. A PV cell is usually made of specially treated semiconductor with p-n junctions. When photon of light strikes the top of the wafer, it can penetrate through a p-n junction and free an electron, which can cross the junction into the n-type region. Free electrons are then held in the n-region, being unable to return into the p-region. This creates a voltage of about 0.50-0.65 volts under open circuit (no-load) condition. When connected to a load, a typical commercial PV cell can produce about from 0.5 to 5.0 watt of electrical power. In typical small residential PV electric systems 32 to 40 cells are connected in series and mounted together on a support structure to form modules (or panels).

Such photovoltaic panels can produce voltages of up to 20 V at open circuit, and about 16-17 volt at peak power. These voltages are suitable for charging 12V batteries via charge regulators. There are also commercial panels with nominal voltages from 24V to 48V. The characteristics of the panels are basically the same as those of the cells, only scaled up in voltage and/or current based on the number of the cells used and their connection. The multiple panels in turn can be interconnected in an array to produce higher voltages, currents and power levels. Parallel connections of the modules increase the maximum available load current, while series connection increases the output voltage. Large-scale installations are configured for voltages up to 600V.

Photovoltaic arrays for residential solar electric generators are usually attached to the roof of the house. Normally they are mounted on racks, which have to be bolted into the roof joists. For the longest length of time of direct sunlight on the panels, ideally they should face true south if you are in the northern hemisphere, and true north if you in the southern hemisphere. The tilt should be close to your latitude [96].

PV cells do not store energy- at night or on a rainy day they would not produce electricity. To provide emergency backup power for your house during outages or at night in off-grid installations you need to store the excess of the "PV electricity" generated from your panels in a large battery bank. The battery bank is normally connected to the PV array via a charger (charge controller). A DC voltage from the battery bank is then converted to AC voltage by a DC-AC inverter, which incorporates an SMPS converter that regulates output voltage by pulse width modulation.

The efficiency of the solar panels and the efficiency of the solar inverter depend on several factors. For the best efficiency, diodes with minimum forward resistance and forward voltage drop are most suitable. SiC switches and diodes show an exciting possibility in terms of the band gap voltage and provide lower junction parasitic capacitances. This eventually leads to faster switching in the inverters and for the blocking and bypass diodes in the solar systems as well. This concept has been used to demonstrate the suitability of SiC diodes for detection of the ultra-violet (UV) rays index [97].

Solar power is the practical alternative to the existing sources of energy. It effectively supplements the current sources of energy. With intelligent designs, the solar cells can be integrated into the final consumer electronic or portable devices. This reduces the dependence of recharging these with the electricity, thus making this ideal choice for the locations where electricity may not be actually available. The solar panels are costly today but their costs are coming down to affordable levels. Also many a times the government aids are available especially for the remote areas.

With the solar panels being available in various form factors, a range of consumer electronic devices making use of the solar panels are making entry into the market place and it is expected that this trend would continue in the future. As of now there are speakers, earphones, calculators and other small power consuming devices already available. The trend is to integrate the solar power in the small form factor that enhances the battery life and acts as an additional source of energy to supplement the battery power from other sources. As the efficiency of the solar panels and conversion would become better, a lesser area would be able to generate more energy and eventually the solar charging may become the sole energy source in many consumer devices.

For the handheld and the gaming applications, this can be very useful. The gadgets are charged by the available solar energy and thus these obviate the need of the normal battery or the rechargeable batteries as the present case.

In this work, the solar energy and its efficient usage is discussed. A special type of diode based on Silicon Carbide (Si-C) material is discussed. These diodes have been found to be extremely useful in the solar and other rectification circuits [97].

3.5.3 Energy Conservation and Low Power Systems

The solar power is an excellent candidate for the consumer applications. Bluetooth speakers. mobile phones. PDAs can be good candidates. Many toys need the rechargeable batteries quite often. Having a solar panel on boards can help to reduce the need of the batteries and also have long life for the devices.

There are many devices based on the solar power that have already appeared in the marketplace in recent times. Renewable energy sources are a viable alternative to the traditional sources [95]. This is applicable to all kind of applications and products where energy consumption is a concern.

Figure 3-16: Solar powered Speakers

The solar energy is becoming popular even in consumer devices. Figure 3-16 shows a solar powered speaker. photographs from website http://www.envirogadget.com.

The solar energy presents very promising increase in the efficiency of the inverters and converters. As pointed out in [95]. the efficiency of the solar conversion depends on the bypass and blocking diodes. New diodes and transistors that are based on Si-C are resulting in better efficiency due to low switching times and ability to work at higher temperatures. A UV index monitoring system based on the Si-C diodes has been demonstrated [97].

In future the transistors based on the SiC would be a reality and a widespread usage of these diodes and transistors will result in considerable improvements in the conversion efficiencies.

3.5.4 Energy Harvesting and Low Power Systems

Another area where considerable interest is growing is the energy harvesting. One way to generate energy is to make use of vibrations. A battery-less vibration based energy harvesting system has been presented in [11]. It was implemented using 0.35μm CMOS process.

3.6 Microelectromechanical Systems

MEMS is an acronym for Microelectromechanical Systems (MEMS). it defines mechanical structures fabricated with IC processing on (most often) silicon wafers. In Europe. MEMS is labeled Microsystems and in Japan it is labeled Micromachines. MEMS defines the technology: not specific products. This technology encompasses a collection of a variety of processes enabling three-dimensional shaping of wafers or stacks of wafers. While most of the applications use silicon wafers. many other materials have been used. including glass and quartz wafers [98].

MEMS is the integration of mechanical elements and electronics on a common substrate through specialized fabrication techniques. The capability of these devices to capture mechanical movements and provide a usable electronic output generates a whole new industry segment. The MEMS sensors family ranges from 2- and 3-axis linear accelerometers to single- and multi-axis gyroscopes. sensor modules. and microphones [17].

Micro-Electro-Mechanical Systems (MEMS) is the integration of mechanical elements. sensors. actuators. and electronics on a common silicon substrate through micro-fabrication technology. The electronics are fabricated using integrated circuit (IC) process sequences and the micromechanical components are fabricated using compatible micromachining processes that selectively etch away parts of the silicon wafer or add new structural layers to form the mechanical and electromechanical devices.

These techniques of integration are resulting into several types of MEMS – accelerometers, gyroscopes etc. MEMS is an enabling technology allowing the development of smart products, augmenting the computational ability of microelectronics with the perception and control capabilities of micro-sensors and micro-actuators and expanding the space of possible designs and applications.

Today, communication, consumer and industrial markets are considered emerging markets for MEMS devices and will be the key drivers for their success. In a mobile phone or PDA, for example, these sensors can add a new, intuitive motion-based approach to navigation within and between pages. In addition, game pads can use these sensors to enable the user to play by just moving the pad to allow the sensor to determine the inclination rather than pushing buttons. Another application uses a MEMS based accelerometer as a vibration detector in home appliances, i.e. for washers or dryers.

MEMS defines the technology; not specific products. This technology encompasses a collection of a variety of processes enabling three-dimensional shaping of wafers or stacks of wafers. While most of the applications use silicon wafers, many other materials have been used, including glass and quartz wafers [**99**].

3.6.1 MEMS Sensors

The MEMS defines the technology and not specific products [**100**]. MEMS have become popular because of certain characteristics that these possess as given in Table 3-10 [**101**].

The silicon is almost as strong but lighter than steel, has large critical stress and no elasticity limit at room temperature as it is a perfect crystal ensuring that it will recover from large strain. Unfortunately it is brittle and this may pose problem in handling wafer, but it is rarely a source of failure for MEMS components. It is clear from the Table 3-10 that silicon is very good choice for sensors, thanks to no fatigue and its weight is also lesser compared to hard material like steel.

Table 3-10: Silicon and Steel Propoerties Comparison [101]

Material	Steel	Silicon
Sensitivity to Stress	Low	High
Fabrication resolution (μm)	10	0.1
Fatigue failures	Yes	No
Mechanical Hysteresis	Yes	No

Several types of variations of MEMS are possible, depending on how the sensed motions is captured. The accelerometers and gyroscopes will be discussed in detail in the following sections.

3.6.2 Accelerometers

An accelerometer is an electromechanical device that will measure acceleration forces. These forces may be static, like the constant force of gravity pulling at your feet, or they could be dynamic - caused by moving or vibrating the accelerometer.

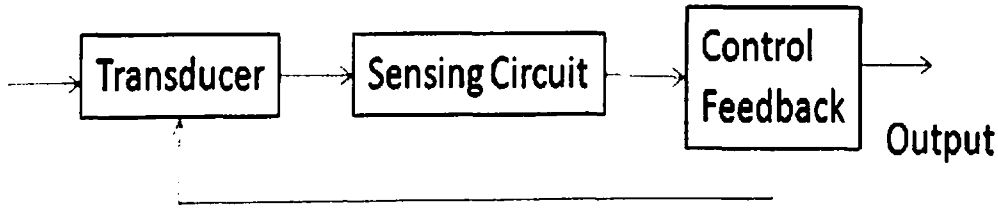


Figure 3-17: The accelerometer sensing Principle [17]

There are many different ways to make an accelerometer work. Some accelerometers use the piezoelectric effect - they contain microscopic crystal structures that get stressed by accelerative forces, which cause a voltage to be generated. Another way to do it is by sensing changes in capacitance. Two microstructures placed next to each other, have a certain capacitance between them. If an accelerative force moves one of the structures, then the capacitance will change. This change in capacitance can be converted to voltage variations that in turn results in an accelerometer. The sensitivity of the sense elements (the ratio of deflection to acceleration) is determined by the mass of the sense element, the distance from the center of mass to the torsion bar axis, and the torsion bar stiffness. Each complete sense element chip contains two wings for a total of four sensing capacitors.

Figure 3-17 depicts the general principle of accelerometer [17]. The acceleration measurement is achieved by means of a proof- mass, spring and damper and damper system made up of micro-fabricated structures. The equivalent mechanical parameter schematic is shown in Figure 3-18. The substrate acceleration generates an inertial force on the proof mass. In the useful operating range of accelerometer, the differential equation in the sensing axis can be written as

$$F_{inx}(t) = m\frac{d^2x}{dt^2} + b_x\frac{dx}{dt} + k_x(t) = ma_{inx}(t) \tag{3-1}$$

where 'F_{inx}' is the feedback force, 'm' is the mass of the proof mass, 'x' is the deflection of the proof mass, 'b' is the damping coefficient and 'k' is the spring constant.

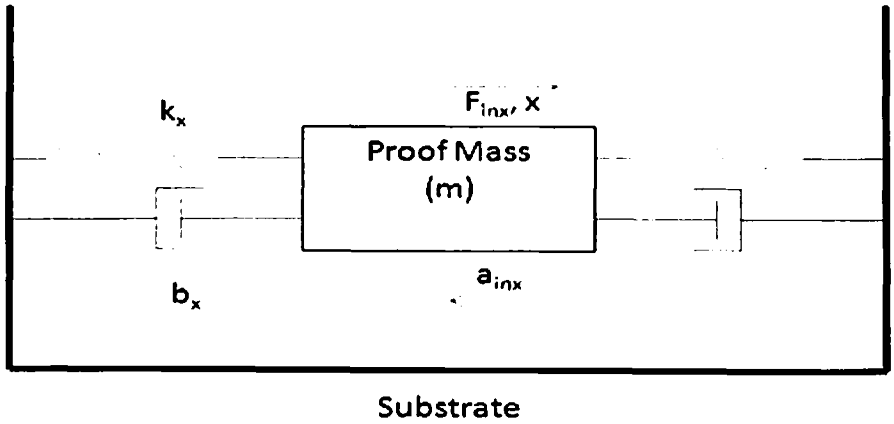


Figure 3-18: Lumped parameter model of a single axis accelerometer [17]

The device has suspended silicon structures along each direction which are attached to the substrate in anchor points and are free to move in the direction of the sensed acceleration. When acceleration is applied to the sensor, the associated moving elements displace from the nominal position, causing an imbalance in the capacitive half-bridge, which is measured using charge integration in response to a voltage pulse applied to the sense capacitor. Thus the motion measured by the sensor is translated into an analog or digital signal.

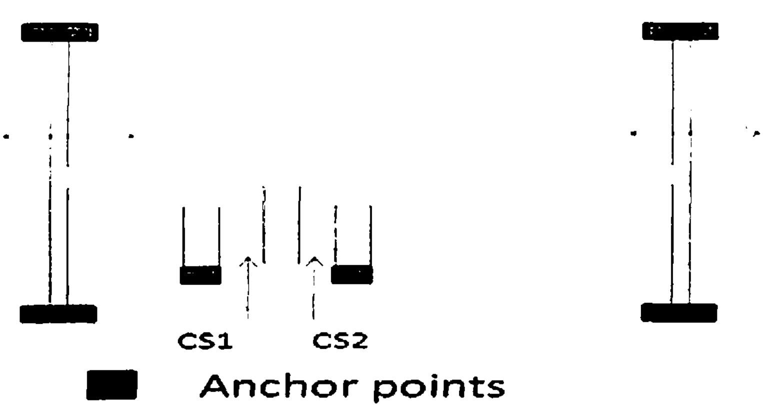


Figure 3-19: Capacitive sensing of acceleration [17]

Figure 3-19 shows the principle of capacitive sensing of the element [17] . The vertical frames have spring action and the springs are tied. In response to the movement, the middle element moves in the direction opposite to the force applied due to inertia. This action causes the imbalance between the two capacitances as shown in Figure 3-20 [17].

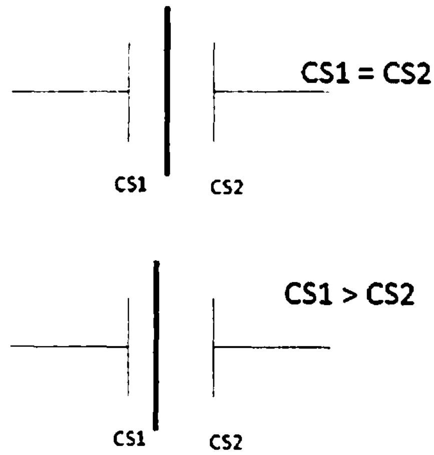


Figure 3-20: Capacitance change with acceleration [17]

3.6.3 MEMS Gyroscopes

MEMS sensors are designed to measure angular rate using the Coriolis force. In the 1830's G.G. de Coriolis discovered that an object moving in a rotating frame would cause an observer on the rotating frame to see an apparent acceleration of the object. In other words, if an object is moving in a straight line, and it is subject to a rotation, you will see a deviation from the original straight line.

The Coriolis Effect is an apparent deflection of moving objects when they are viewed from a rotating reference frame. The Coriolis force acts in a direction perpendicular to the rotation axis and to the velocity of the body in the rotating frame and is proportional to the object's speed in the rotating frame.

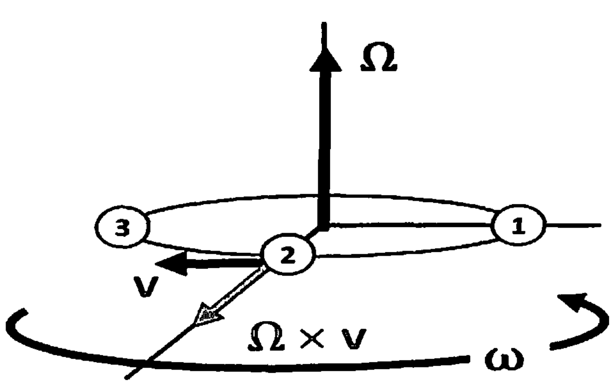


Figure 3-21: Coriolis Force

At a given rate of rotation of the observer, the magnitude of the Coriolis acceleration of the object is proportional to the velocity of the object and also to the *sine* of the angle between the direction of movement of the object and the axis of rotation. The equation for the Coriolis force is

$$\overrightarrow{F_C} = -2m\vec{\Omega}\times\vec{v}$$

(3-2)

where F_C is the Coriolis Force, Ω is the angular velocity and v is the velocity.

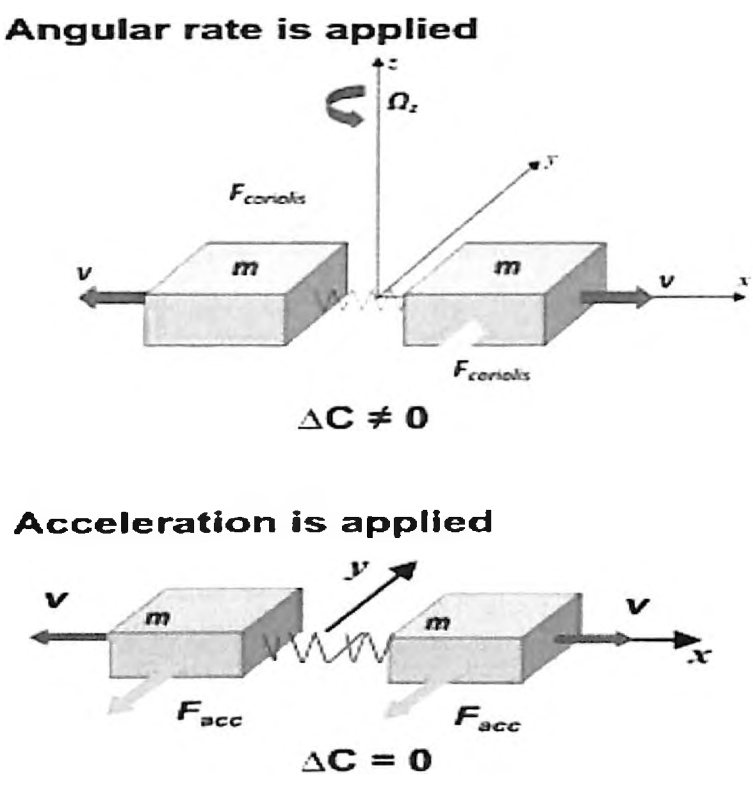


Figure 3-22: Principle of Tuning Fork Gyroscope

The tuning fork gyroscope principle is shown in Figure 3-22 (REF : www.st.com). The two blocks of the mass are shown. When there is a rotation around the z- axis. Coriolis force acts as shown. Similarly when the motion is linear, there is no Coriolis force produced.

3.7 Inertial Measurement Units (IMU)

The continuous innovation in new technological processes permits the Inertial Measurement Units (IMUs) to become a fundamental part in a broad range of applications, from the most typical ones such as dead reckoning and game controllers to the last technological breakthrough sectors as patients' rehabilitation in medical segment or the Electronic Stability Control (ESC) in the automotive segment **[102]**.

The reason of this great success is mainly attributed to two factors: first because of the MEMS-based technology development that has significantly improved inertial sensors' performances and strongly reduced package sizes, making another step forward in the field of the system miniaturization; then, because of the use of more reliable embedded algorithms and calibration procedures, designed to enable the convergence of several sensors in the same platform and to make the system more robust.

Based on these assumptions, IMUs' capability of characterizing processes or environments has become a fundamental feature for the understanding and the development of system solutions.

In the gaming industry, the market trend is toward active gaming platforms, where a smart wearable motion capture suite could expand the capability of this type of game and open total virtual reality scenarios, at the edge of the augmented reality.

The microcontrollers described in section 3.2.3 and MEMS described in section 3.6 can be combined to create Inertial Measurement Units.

IMUs represent a complete hardware solution for a variety of applications including human machine interfaces, robotics, platform stabilization, and virtual and augmented reality. Today's motion sensing technology, mixed with untraditional algorithms, is enabling new levels of innovation in all electronics markets.

For example, since its first appearance on the market, multisensorial platforms have changed the way of playing with the game consoles in a new dynamic mode. This has been possible thanks to the data fusion among the different sensors of the IMU used to implement the game controllers.

Data fusion among several sensors is also important for navigation system solutions either in industrial, automotive or gaming applications used as handheld devices. The Inertial Measurement Units can be also useful as Human Machine Interface in industrial processes, to increase workers' safety avoiding any physical risk in objects manipulation and environment interaction. IMUs are used to assign cognition capability to industrial manipulators, small smart arms, and exoskeleton parts, in order to help people to better manage assembly processes.

Distributed sensors architecture for motion capture would be hosted on different structure of robots, like manipulators and rovers.

3.7.1 IMU Hardware Structure

In an Inertial Measurement Unit the inertial sensors and the Microcontroller Unit (MCU) represent the core of the platform. After data capturing, the MCU executes the Extended Kalman Filter (EKF), a set of mathematical equations that provides an efficient computational means able to minimize the mean of the squared error.

The design of an inertial platform must follow several requirements and constrains in order to have the best trade-off between performances, cost, and system's flexibility to cover a wide range of applications. iNEMO platform has been designed following these guidelines in order to have a modular solution based on the principles of miniaturization, low power consumption, and cost-effectiveness.

The starting point for designing an inertial platform is the definition of the main components.

The iNEMO platform is provided with a 10-Degree-of-Freedom (10-DOF) sensors system, so the products selection is fundamental to mark out the system performances. The key components of ST iNEMO are:

3.7.2 Geomagnetic Module

The 3-axis accelerometer and 3-axis magnetometer are included in the LSM303DLH geomagnetic module in 5 × 5 × 1.5 mm package. The accelerometer part has a dynamically selectable full-scale range of ±2 g/±4 g/±8 g, the data output data rate is from 0.5 Hz to 1 kHz.

Data fusion among several sensors is also important for navigation system solutions either in industrial, automotive or gaming applications used as handheld devices. The Inertial Measurement Units can be also useful as Human Machine Interface in industrial processes, to increase workers' safety avoiding any physical risk in objects manipulation and environment interaction. IMUs are used to assign cognition capability to industrial manipulators, small smart arms, and exoskeleton parts, in order to help people to better manage assembly processes.

Distributed sensors architecture for motion capture would be hosted on different structure of robots, like manipulators and rovers.

3.7.1 IMU Hardware Structure

In an Inertial Measurement Unit the inertial sensors and the Microcontroller Unit (MCU) represent the core of the platform. After data capturing, the MCU executes the Extended Kalman Filter (EKF), a set of mathematical equations that provides an efficient computational means able to minimize the mean of the squared error.

The design of an inertial platform must follow several requirements and constrains in order to have the best trade-off between performances, cost, and system's flexibility to cover a wide range of applications. iNEMO platform has been designed following these guidelines in order to have a modular solution based on the principles of miniaturization, low power consumption, and cost-effectiveness.

The starting point for designing an inertial platform is the definition of the main components.

The iNEMO platform is provided with a 10-Degree-of-Freedom (10-DOF) sensors system, so the products selection is fundamental to mark out the system performances. The key components of ST iNEMO are:

3.7.2 Geomagnetic Module

The 3-axis accelerometer and 3-axis magnetometer are included in the LSM303DLH geomagnetic module in 5 × 5 × 1.5 mm package. The accelerometer part has a dynamically selectable full-scale range of ±2 g/±4 g/±8 g, the data output data rate is from 0.5 Hz to 1 kHz, in very small sizes (3 × 3 × 1 mm). In the accelerometer, the sensing element, capable of detecting the acceleration, is manufactured using a dedicated process developed by ST to produce inertial sensors and actuators in silicon.

The magnetometer range is from ±1.3 to ±8.1 (gauss) and the bandwidth is about 20 Hz. The magnetometer is based on a thin film trigate fluxgate for detecting a component of a magnetic field in directions of three axes.

3.7.2.1 Gyroscopes

The iNEMO platform includes one 1-axis Yaw gyro LY330ALH and the biaxial Roll Pitch gyro LPR5430AL.

The gyros have a miniaturized 3 × 5 × 1 mm and 4 × 5 × 1 mm package, respectively, a full-scale range of ±300 Deg/s with a bandwidth of 140 Hz, and sensitivity of 3.3 mV/Deg/s. Particularly output of LY330ALH has a full scale of ±300°/s and is capable of measuring rates with a −3 dB bandwidth up to 88 Hz.

The combination of these sensors allows a compact design with all the 3-axial gyro system in one planar layer.

The LPR430AL has a similar structure for each axis.

3.7.3 Pressure and Temperature Sensors

The LPS001DL pressure sensor is the 300–1100 mbar absolute full scale with *I2C* digital output and barometer. The STLM75 is the temperature sensor with −55 to +125°C range and *I2C* digital interface.

3.7.4 Microcontroller Unit

The STM32F103 MCU collects the data from the sensor and performs the EKF algorithm. The MCU is a high-performance ARM Cortex-M3 with 32-bit RISC core working at 72 MHz, high-speed embedded memories (flash memory up to 128 Kbytes and SRAM up to 20 Kbytes), and an extensive range of enhanced I/Os and peripherals connected to two APB buses.

3.7.5 Algorithms for IMU : Sensor / Data Fusion

In the IMU platform, a data fusion algorithm calculates the orientation data, starting from the measurements of several sensors. A set of mathematical equations. called Kalman filter. combines measurements coming from different sensors [**102**].

The Kalman filter provides an efficient computational recursive means to estimate the state of a process. minimizing the mean of the squared error. As very powerful tool. it supports estimations of past, present. and even future states. and it can do so even when the precise nature of the modeled system is unknown.

When either the system state dynamics or the observation dynamics is nonlinear. the Extended Kalman Filter (EKF) is adopted.

In this scenario, using several kinds of sensors, the characteristics of each one overcome the limitation of the others. So, while gyroscopes measure orientation by integrating angular velocities. and the accelerometer and magnetometer provide a noisy and disturbed but drift-free measurement of orientation, the EKF weights the three sources of information in an appropriate way.

In this section. after a brief formulation of the discrete-time EKF algorithm. it will be described the structure of the quaternion-based EKF.

3.7.5.1 Kalman Filter Formulation

In general, the Kalman Filter algorithm estimates the state of a discrete-time process starting from the equations below:

$$x_{k+1} = A_k x_k + B_k u_k + w_k \tag{3-3}$$

$$z_k = H_k x_k + v_k \tag{3-4}$$

Where,

- x_k is the state vector at the k time step. while z_k is the output
- A. B. and H are. respectively. state. input. and output matrices and
- W, v are state and measured noise. They are random. Gaussian. and white noise source with covariance matrix Q and R. respectively.

Equation (3-3) is the state equation while (3-4) is the output equation. The vector x contains all of the information about the present state of the system. but we cannot measure x directly.

For every time step. the algorithm provides estimation both for the state x_k and for the error covariance P_k. This one provides an indication of the uncertainty associated with the current state estimate.

The updated measured equations (corrector equations) provide a feedback by incorporating a new measured value into the *a priori* estimate to get an improved *a posteriori* estimate.

The Kalman Gain K derives from the minimizing of the a posteriori covariance error and could be considered as a measure of the confidence level of the predicted state. In fact, if R approaches zero. the actual measured z is more reliable than the predicted measured $\tilde{H}x$. while if P approaches zero, the predicted measure is more dependable.

3.8 Chapter Conclusions and findings

In this chapter, exhaustive study has been done to arrive at the final choices made for the selection of various technologies, hardware and operating environments.

ZigBee is found to be the choice for the ad-hoc networking. It needs small footprint for memory. forms ad-hoc network in the quickest time and has reasonable costs compared to Bluetooth and Wireless LAN.

DLX processor is studied and porting of JOP to an FPGA is discussed. This can be the most optimized platform for the mobile gaming with even lower footprint. The initial implementation can be proven on the FPGA. This is suggested for the future work. For this work. microcontroller environments have been used for the sake of simplicity and availability as several FPGA platforms are needed to prove the gaming concept. The porting of DLX, ASPIDA and JOP has been done on an Altera Cyclone III FPGA.

For the handheld devices there are several operating systems available. From the point of view of the support required for large setups, commercial operating systems like Windows CE are useful. For the development oriented environment and proof of concept, the Linux based operating systems for the embedded devices are preferred as these are free and allow the designers full creativity to be exercised.

The J2ME is concluded to be excellent choice for the handheld devices in the instances where the target processor has support for Java. When a computer monitor is used as the common platform by the players, the Java offers equally good environment for development of multi-player games in the host. This subject is illustrated in the methodologies chapter.

In the proposed target i.e. Java Optimized Processor, the J2ME based games can be ported as part of future work.

The gaming experience today is based on the acceleration, tilt, vibration and rotation. These actions can be captured by the accelerometers and gyroscopes. The semiconductor accelerometers and gyroscopes have the advantage of being able to connect to microcontrollers through easy interfaces. This allows making very versatile platforms based on the MEMS.

As the sizes of the silicon and subsystem shrink, it is possible to integrate many units into smaller sizes. IMU is a unit that has several MEMS and other sensors connected to a microcontroller. Sensor fusion or data fusion algorithms based on Kalman filtering provide the sensor data to the microcontroller is pre-defined manner. This data can be used to control the gaming actions and sent to the gaming consoles through connecting mechanism – wired or wireless. This presents a very powerful mechanism for gaming and this concept can be used in several applications including the accurate motion detection of body parts [**102**].

To conclude then, choice is made to use ZigBee, Microcontrollers, MEMS, J2ME and Java and low power electronics for conversion. These platforms are discussed in the Subsystem Modules in next chapter. The IMU presents an excellent next step towards further integration with ZigBee for enhanced gaming experience.

4 SUBSYSTEM MODULES

As discussed in preceding chapters, there are a number of wireless standards, operating environments and processors to choose from. In this chapter some of these implementations are discussed and optimum choices for this work are based on the various results and the suitability.

This chapter focuses on the development, design and verification of the individual sub-systems that prove the concepts / technologies shortlisted in the last chapter. Their integration to create a gaming environment is discussed in a separate chapter.

The conclusions based on the study as presented in the previous chapters have resulted in using ZigBee for communication, MEMS for gaming experience, Microcontrollers for platforms and Java as gaming environment. The various sub-systems developed and tested for this work based on the selected areas and their details are given below.

It must be mentioned here that the boards developed and used for this work are primarily based on STMicroelectronics devices. The boards are developed in Systems LAB for the ongoing activities. The detailed datasheets or references are intentionally not given. All the datasheets and some of the boards used in this work are accessible through STMicroelectronics internet site, *www.st.com*.

4.1 ZigBee Communication Platforms

ZigBee communication for ad-hoc networks has been demonstrated using a number of different platforms. A ZigBee based home automation network platform has been implemented that [51] establishes itself seamlessly and the nodes continue to send their data to the central node periodically.

Another implementation of the ZigBee based networks is the camera system that captures the images and sends out to a display that is connected to the camera through ZigBee and displays the images captured periodically [51], Figure 4-1.

4.1.1 A ZigBee camera receiver and transmitter

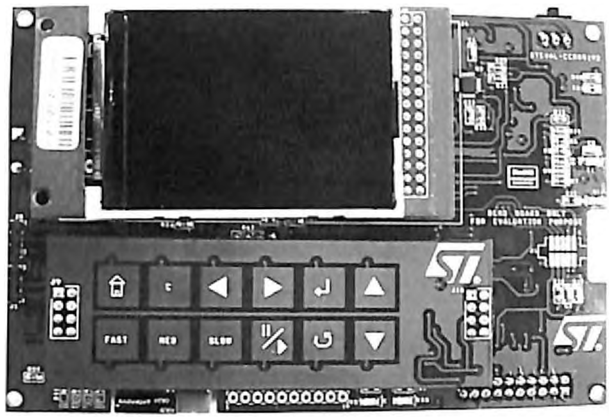


Figure 4-1 : A ZigBee receiver for displaying images received [51]

The main components of the platform shown in Figure 4-1 are [**17**]

- Microcontroller (STM32)
- TFT (AM-240320L8TNQW00H)
- Storage media (Micro SD card)
- MEMS (LIS331DLH)
- ZigBee Module (SPZB260)

STM32-E is a 32 bit MCU based on popular ARM 32-bit Cortex™-M3 CPU running at 72 MHz with performance of 90 DMIPS with 1.25 DMIPS/MHz. The memories embedded in it are up to 64Kbytes of SRAM and 512Kbytes of flash memory. The microcontroller has Single-cycle multiplication and hardware division. It has up to 112 fast general purpose IOs to enhance the overall performance. The IOs are 5V tolerant.

The microcontroller has up to 13 communication interfaces which includes two I2C (400 KHz), five USARTs (4.5Mbps), three SPIs (18MHz), CAN 2.0B Active interface, USB 2.0 (12Mbps) full speed interface, SDIO interface and FSMC interface. Among these this application is using the two SPIs, two I2C, SDIO, FSMC and USB interfaces for this application.

The microcontroller works on single voltage from 2V to 3.6V unlike several other microcontrollers that may require dual voltage.

The STM32 is a generic name that represents 32 bit microcontroller family from STMicroelectronics. Various variations of packages, pin-outs and internal memory result into large number of members in this and the size continues to grow.

Many features of microcontroller help in reducing the overall cost of the system. For example inbuilt RTC can be used for implementation of calendar or any other reference clock.

The TFT is interfaced with the microcontroller through FSMC. FSMC is parallel mode of interface using which, image data dumping in the TFT RAM is faster.

LIS331DLH is a motion sensor device. MEMS is used in this system for add-on feature like auto-orientation of image as per the frame alignments like landscape or portrait. MEMS is using SPI to be interfaced with STM32.

4.1.2 A ZigBee based home automation system

A ZigBee based platform for home automation has been developed. It is a handheld device which works as the hub for low data rate wireless network. This system measures the temperature, humidity and light intensity and gathers information of various sensors (temperature, humidity, light intensity and MEMS) from nodes placed at different locations. It can also configure the alarm thresholds for the parameters of all the connected nodes, as well as for itself. Lights (or any other AC load) connected to nodes may be controlled through the Smart Monitoring Station with addition of a few components and its status is available on the Smart Monitoring Station display.

The ZigBee network is set up based on the previous history of connection. A new node can be connected to the system based on the session initiated by the master node. Once the node has been authenticated, it joins and leaves the network in an ad-hoc manner.

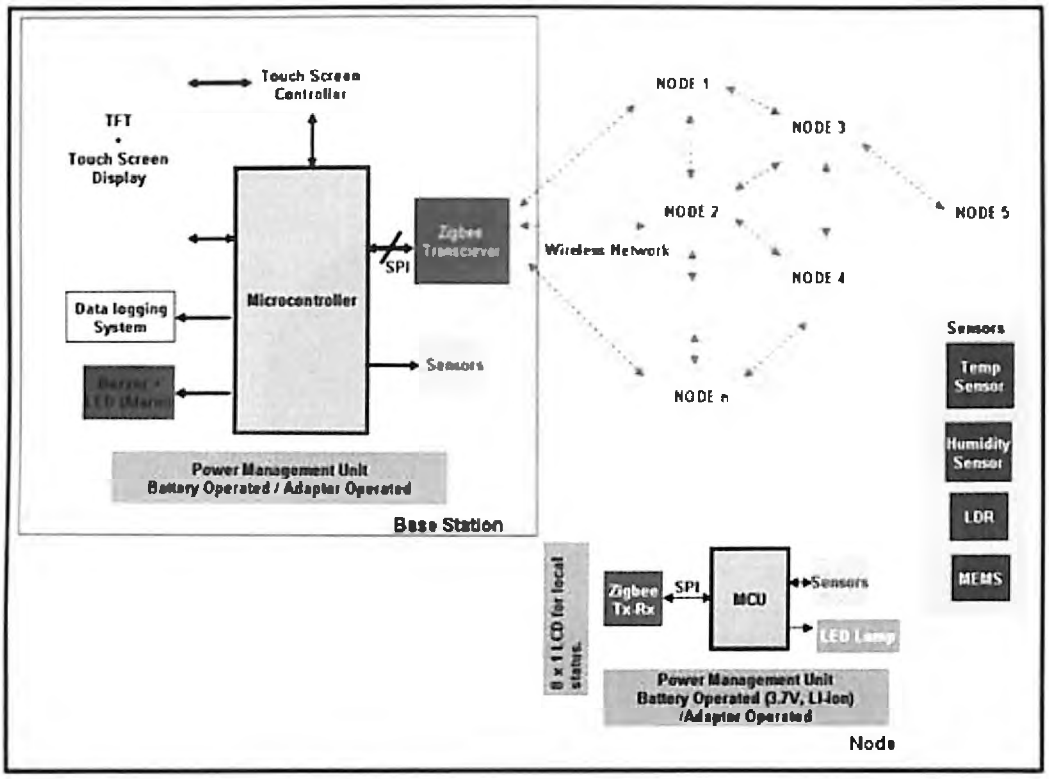


Figure 4-2: A ZigBee based home networking platform [103]

The system has a user interface with the graphic menu, color TFT display (240 X 320) and Touch Screen. Data logging and analysis of logged data can be done using the GUI developed for this system. The board can be powered up using 3.7V Li-ion Battery. Alternatively it can work with USB power without battery. When battery is mounted, the same USB charges battery and operates the system.

4.2 MEMS for gaming

The MEMS are becoming increasingly popular for the handheld gaming. The actions required for the gaming purpose include vibration, inclination, shock information that can be captured in real time and displayed.

4.2.1 Sensing Inclination using MEMS

The MEMS action as explained above can be captured in a handheld device / gadget and transferred over wireless to a console where the gaming application based on Java may

Figure 4-3 : The MEMS Accelerometer for tilt detection [17]

be running on the computer. With a wireless protocol that supports multi-inputs (like ZigBee or Bluetooth), more than one player could be playing this game. If the transmission times are small enough, the users get the feeling of the real time behavior.

The sensing of the tilt is demonstrated using a MEMS accelerometer based gadget shown in Figure 4-3. The vibration detection can also be similarly detected and used to produce meaning results [17].

The main features of this demo are:

- Detection of Pitch and Roll motion of the board and display of this information using the LEDs.
- Detection of circular motion of the board.
- Detection of free fall of the board.
- Six different modes of operation of the system.
- Music/rhythm patterns as per the mode and motion of the board.
- Low power consumption along with standby of the system on regular intervals of time.
- Battery monitoring and display of the battery status using the LEDs

- Powered up by 3 AAA batteries.

The system. called as Inclination Analysis Demonstration Board. is a handheld device which detects the pitch (angular motion about Y axis) and roll (angular motion about X axis) and this information is displayed using the different color LEDs placed in two circles.

The system can detect the free fall of the board and this information is displayed using the Bi-color led placed at the center of the board. The board can also be configured to detect the circular motion of the board and LED patterns are generated depending on the motion of the board. The system is also provided with the demonstration mode, in which different LED patterns get displayed irrespective of the position/motion of the board making the system suitable for exhibitions and seminars.

The system is put on the standby whenever there is no motion of the board for more than 10 seconds. reducing the power consumption of the system. The system wakes up from standby whenever any vibration/motion is detected. The system also monitors the battery and user can check the battery status at any time. Whenever the low battery is detected, the system alerts the user to change the battery and enters into halt mode to avoid any misbehavior of the system.

4.2.2 Sensing Vibration using MEMS

An STM8 based platform has been implemented for vibration detection. It is shown in Figure 4-4. Its features are [17]:

- Detection of vibration of the board and display of this information using the LEDs.
- Detection of circular motion of the board.
- Detection of tilt of the board.
- Six different modes of operation of the system.
- Music/rhythm patterns as per the mode and motion of the board.
- Low power consumption along with standby of the system on regular intervals of time.

- Battery monitoring and display of the battery status using the LEDs
- Powered up by 3 AAA batteries.

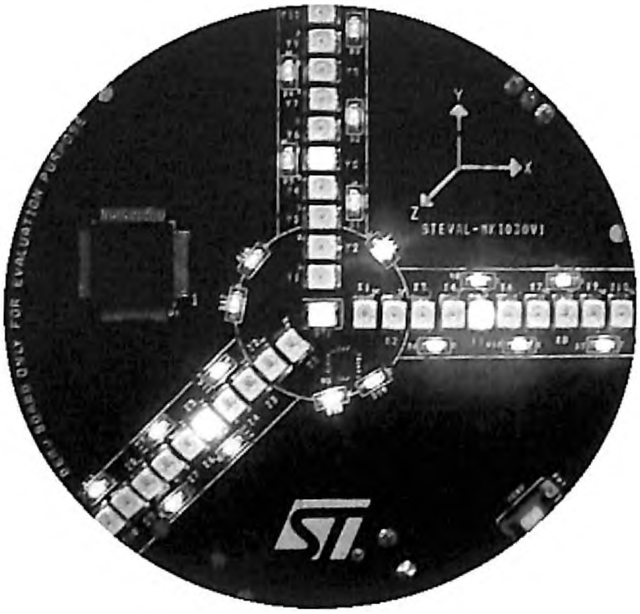


Figure 4-4: The MEMS accelerometer for vibration detection [17]

The system, called as Vibration Analysis Board, is a demonstration board which detects the vibration of the board and measures the acceleration in all three axes(X, Y and Z) because of this motion. This information is displayed using the different color LEDs placed in three axes.

The system can detect the tilt of the board and this information is displayed using the Bi-color LED placed at the center of the board. The board can also be configured to detect the circular motion of the board and LED patterns are generated depending on the motion of the board.

The system is also provided with the demonstration mode, in which different LED patterns get displayed irrespective of the position/motion of the board. There are three musical modes in which the music/rhythm is played as per the mode/motion of the board and there are three silent modes making total of six different modes of operation of the system. The system can be switched from one mode to other by pressing an onboard push button designated MODE.

The system is put on the standby whenever there is no motion of the board for more than 10 seconds, reducing the power consumption of the system. The system wakes up from standby whenever any vibration/motion is detected. The system also monitors the battery and user can check the battery status at any time. Whenever the low battery is detected the system alerts the user to change the battery and enters into halt mode to avoid any misbehavior of the system.

The board is of circular shape with total dimensions of 84mm diameter and 26mm height.

4.2.3 MEMS Controlled Wireless Vehicle

A toy vehicle that operates wirelessly with the action of joysticks or MEMS has been developed, Figure 4-5. The vehicle accepts the wireless commands sent from the handheld remote without any security. A proprietary protocol based on the PWM modulation has been developed and successfully tested on this system.

The system supports the wireless control of the vehicle motion through the remote using the joysticks and gestures. The system supports the complete motion of control of the vehicle (FORWARD, Backward, Left turn, Right turn). It is configured to support 5 levels of different speeds and 5 different level of angle in each direction.

Depending on the mode of the remote, the remote gets the user inputs for the vehicle motion. In joystick mode, the user needs to use two joysticks and in gesture mode the user is required to press one joystick and tilt the hand to give different gestures. These user inputs are encoded to the system commands and the control signal is provided to the RF transmitter module. These commands are modulated to RF frequency of 433 MHz and are transmitted in open space by the RF transmitter.

The vehicle which is battery powered and is having the various electronic components receives these signals with the help of the RF receiver module mounted on the vehicle. The module demodulates the commands sent by the remote and gives the PWM signal to the MCU (STM32F103VET6). The MCU of the vehicle decodes the command and takes the appropriate action.

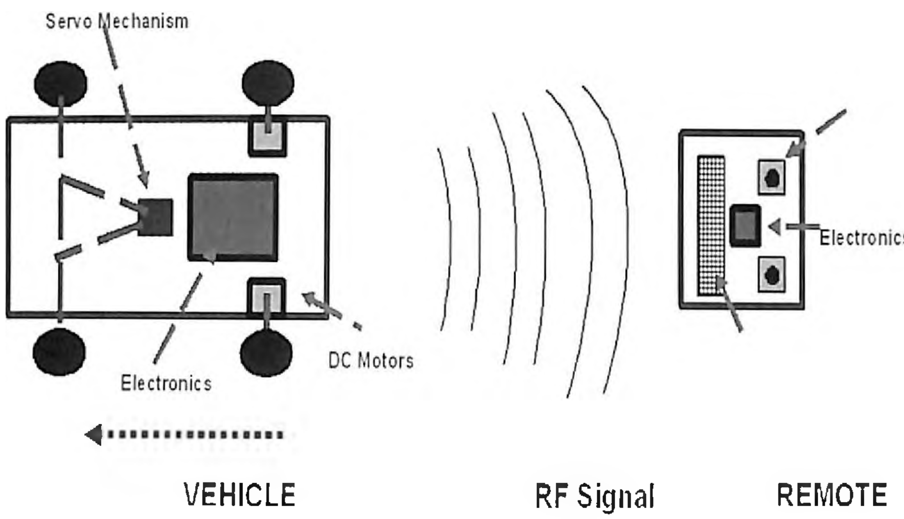


Figure 4-5: Wireless MEMS toy vehicle

The speed and the forward/backward motion of the vehicle is controlled using the DC motors which itself is controlled by the PWM signals issued from the MCU to the motor driver. More the duty cycle of the PWM signal, more is speed. The angular motion of the vehicle is controlled using the servo mechanism which in turn is controlled through the PWM signal issued by the MCU. There is one ultrasonic transducer which detects the obstacles after every 50 milliseconds. If an obstacle is detected by the system, the user is warned by buzzer sound and the vehicle stops its forward motion. If the obstacle is removed then the vehicle will continue the forward motion if its speed is changed. Also during every angular motion, the indicator LEDs blink with the indicator sound to give the effect of actual vehicle.

Toy vehicle is developed using the two DC motors, servo mechanism and the required electronics. The vehicle is powered with the rechargeable 12.0 V battery.

The two DC motors provide the torque to the rear wheels of the vehicle. One servo mechanism is developed to provide the angular control to the front wheels of the vehicle. As shown in the block diagram, Figure 4-6, the electronics of the vehicle consists of many components (STM32 Microcontroller, LIS331DLH Accelerometer, LY510ALH Gyroscope etc...). There is one RF module receiver which receives the modulated RF signals at 433 MHz and gives the demodulated signal to MCU

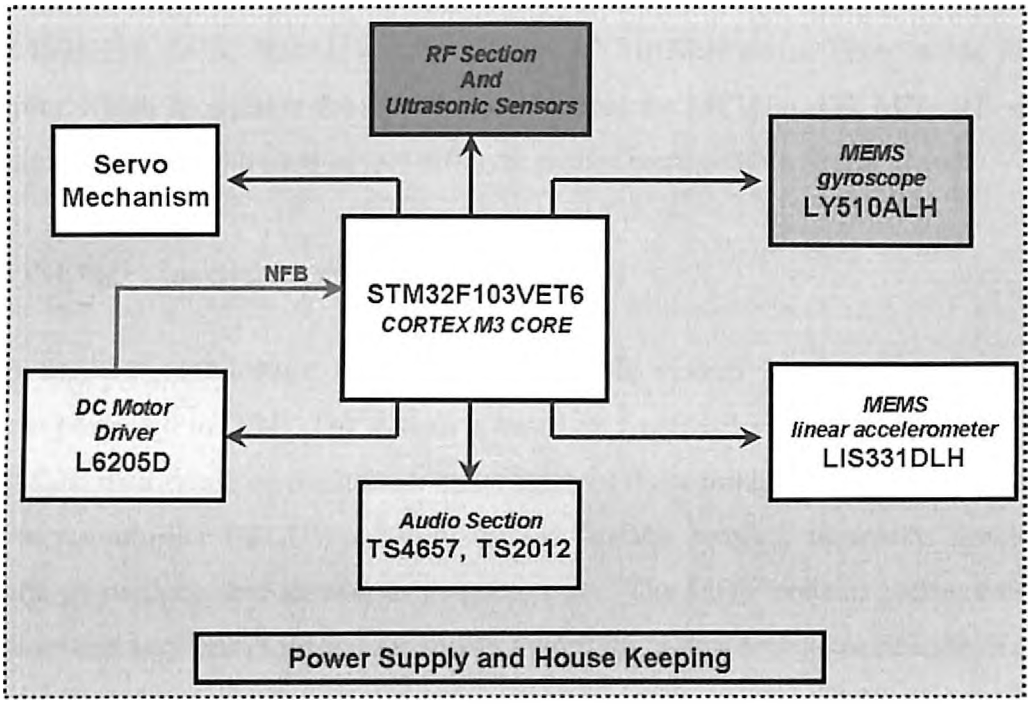


Figure 4-6: Block Diagram of the Toy Vehicle

Ultrasonic transducer is available at the front of the vehicle which works on the principle of SONAR and detects the obstacle. On detection of the obstacle, the user is intimated with the buzzer sound and the vehicle stops moving in the forward direction.

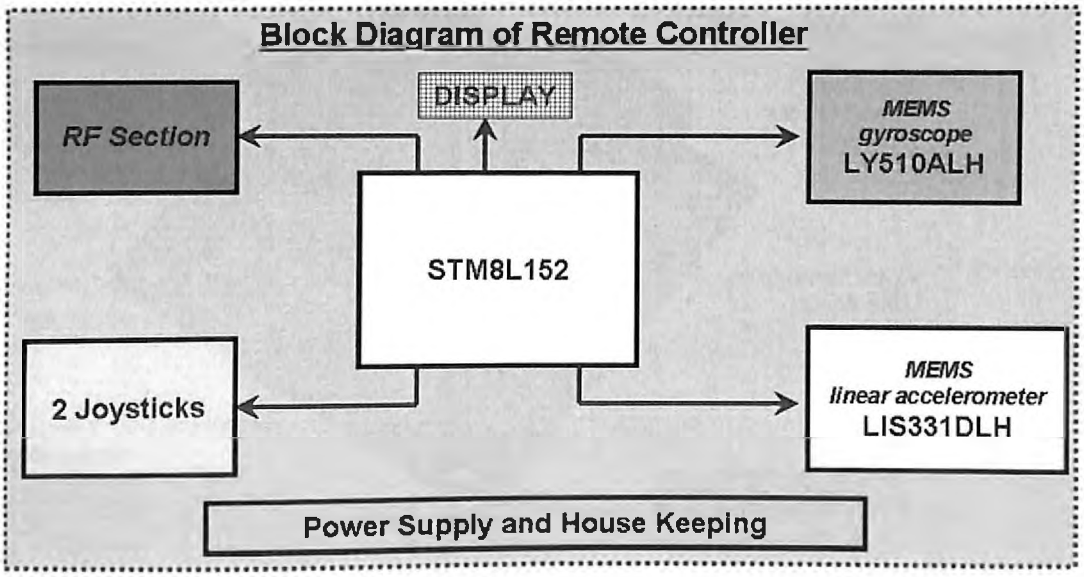


Figure 4-7: The remote control for wireless toy vehicle [74]

Remote controller consists of two joysticks and one LCD display for the user interface. As shown in the block diagram. Figure 4-7. the electronics of the remote consists of STM8L152C6T6. LDS3985M33R. LIS331DLH. LY510ALH etc.... There is one RF module transmitter which modulates the PWM signal issued by MCU to 433 MHz RF signal and transmits. Remote can be used in two different modes configurable by the switch.

4.2.4 iNEMO : Inertial Measurement Unit (IMU)

A modular architecture to develop a wearable system for real-time human motion capture is presented in [**104**]. The system is based on a network of smart inertial measurement units (IMUs) distributed on the human body. Each of these modules is provided with a 32-bit RISC microcontroller (MCU) and miniaturized MEMS sensors: three-axis accelerometer. three-axis gyroscopes. and three-axis magnetometer. The MCU collects measurements from the sensors and implement the sensor fusion algorithm. a quaternion-based extended Kalman filter to estimate the attitude and the gyroscope biases. The design of the proposed IMU. in order to overcome the problems of the commercial solution. aims to improve performance and to reduce size and weight. In this way. it can be easily embedded in a tracksuit for total body motion reconstruction with considerable enhancement of the wearability and comfort. Furthermore. the main achievements will be presented with a performance comparison between the proposed IMU and some commercial platforms.

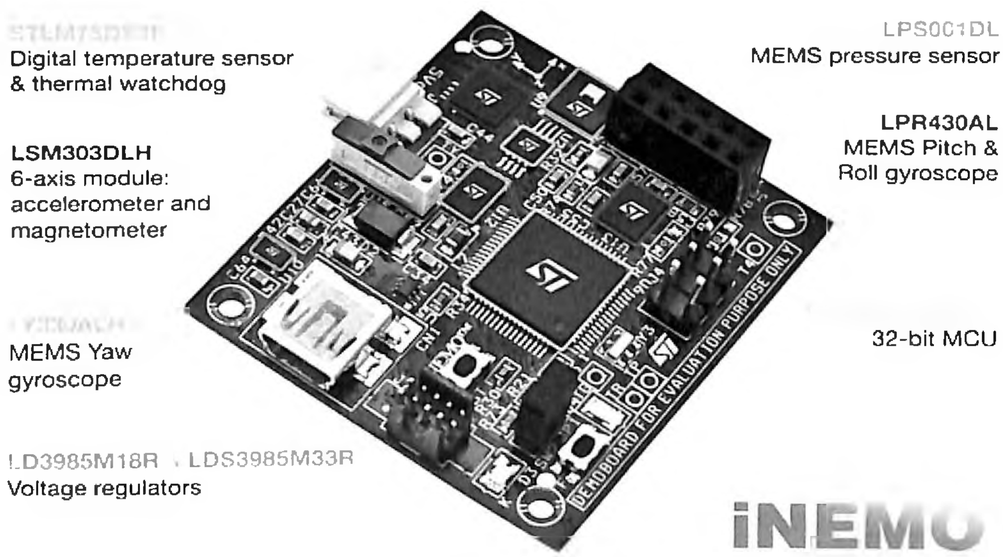


Figure 4-8The iNEMO Platform [104]

In the gaming industry, the market trend is toward active gaming platforms, where a smart wearable motion capture suite could expand the capability of this type of game and open total virtual reality scenarios, at the edge of the augmented reality.

Wireless connectivity would represent a significant improvement on wearability, but power consumption issues and consequently power storage problems (i.e., battery weight and size) become relevant. However, the ZigBee protocol provides a reliable low-power connection with a great deal of flexibility in the network connection.

The small packages housing MEMS sensors allow the miniaturization of the whole system, favoring wearability and portability. Due to its characteristics, this method represents the best tradeoff between performance, comfort, and cost to cover a wide range of applications.

iNEMO is equipped with various MEMS sensors to estimate the orientation of human body segments. In particular, we use a three-axis accelerometer, a three-axis magnetometer, and three axis gyroscope [**104**].

4.3 Porting of ASPIDA DLX processor to Altera Cyclone III

The HDL description of the ASPIDA DLX processor has been ported on to the Cyclone-III FPGA from Altera. The target used for this purpose is EP3C25F324.

The HDL description of the ASPIDA by Technical University of Darmstadt is used as a reference for the purpose of porting on to the FPGA. The design files include DlxPackage.vhd, Dlx.vhd, Environment.vhd, TestBench.vhd.

The Cyclone III FPGA Starter Development Kit has the following features:

1. Cyclone III starter board (Figure 4-9)
2. Cyclone III EP3C25F324 FPGA
3. Embedded USB-Blaster™ circuitry (includes an Altera EPM3128A CPLD) allowing download of FPGA configuration files via the user's USB port
4. 256-Mbit of DDR SDRAM

5. 1-Mbyte of synchronous SRAM

6. 16-Mbytes of Intel P30/P33 flash

7. 50-MHz on-board oscillator

The development environment used is Quartus II Web Edition. Version 7.1 or later. The sources as above are compiled after proper pin assignment and other steps of programming are completed.

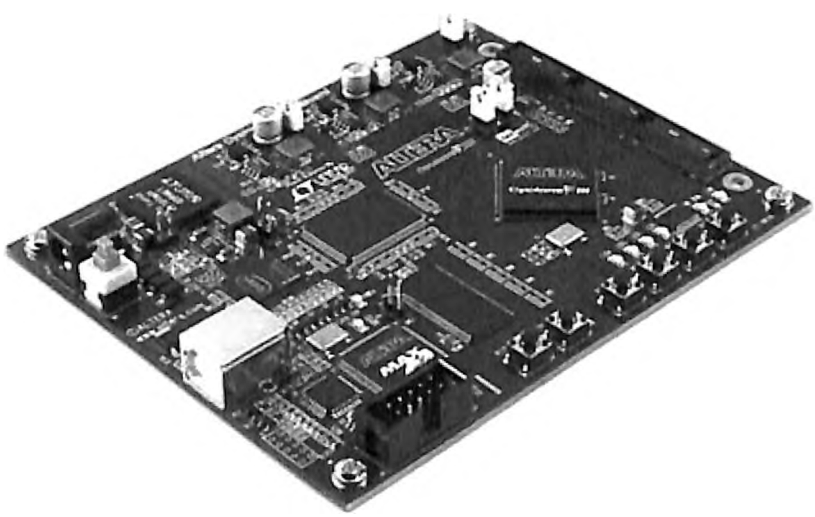


Figure 4-9: Altera Cyclone III development platform

The total power consumed is calculated using the Power Play Tool that is the part of the tool chain. The total Thermal Power Dissipated is calculated to be 73 mW using this tool.

4.4 Java Optimized Processor

The common design practice for embedded systems is using an off-the-shelf microcontroller and programming in C, with or without an RTOS. The JOP approach differs in the following ways:

- The processor is a soft core for an FPGA. FPGAs are expensive compared to microcontrollers. However it is possible to fit the processor core and other peripherals into a single FPGA itself. In this manner the FPGAs offer even more flexibility than the microcontroller based designs.

- The Java based systems need JVM as interpreter. The JOP implements the instruction set of the JVM in hardware, thus minimizing the gap between C and Java.

The hardware implementation of the Java virtual machine (JVM), targeting small embedded systems has been done in this work. JOP is designed from the ground up with time-predictable execution of Java bytecode as a major design goal. All functional units, and especially the interactions between them, are carefully designed to avoid any time dependency between bytecodes.

The table below is about the gate count estimates.

Table 4-1: Gate count estimate [69]

Processor	**Core (gate)**	**Memory (gate)**	**Total (gate)**
JOP	11K	40K	51K
picoJava	128K	314K	442K
aJ80/aj100	25K	590K	615K

4.5 Java for Gaming

The Java Wireless Kit for the Gaming environments presents very useful environment for game development. The output of this directly runs on the target mobile device.

From the Project Menu, we can package and have an executable Java file, ready to run on the target. The above has been tested on NOKIA N73 mobile phone successfully.

peripherals into a single FPGA itself. In this manner the FPGAs offer even more flexibility than the microcontroller based designs.

- The Java based systems need JVM as interpreter. The JOP implements the instruction set of the JVM in hardware, thus minimizing the gap between C and Java.

The hardware implementation of the Java virtual machine (JVM), targeting small embedded systems has been done in this work. JOP is designed from the ground up with time-predictable execution of Java bytecode as a major design goal. All functional units, and especially the interactions between them, are carefully designed to avoid any time dependency between bytecodes.

The table below is about the gate count estimates.

Table 4-1: Gate count estimate [69]

Processor	**Core (gate)**	**Memory (gate)**	**Total (gate)**
JOP	11K	40K	51K
picoJava	128K	314K	442K
aJ80/aj100	25K	590K	615K

4.5 Java for Gaming

The Java Wireless Kit for the Gaming environments presents very useful environment for game development. The output of this directly runs on the target mobile device.

From the Project Menu, we can package and have an executable Java file, ready to run on the target. The above has been tested on NOKIA N73 mobile phone successfully.

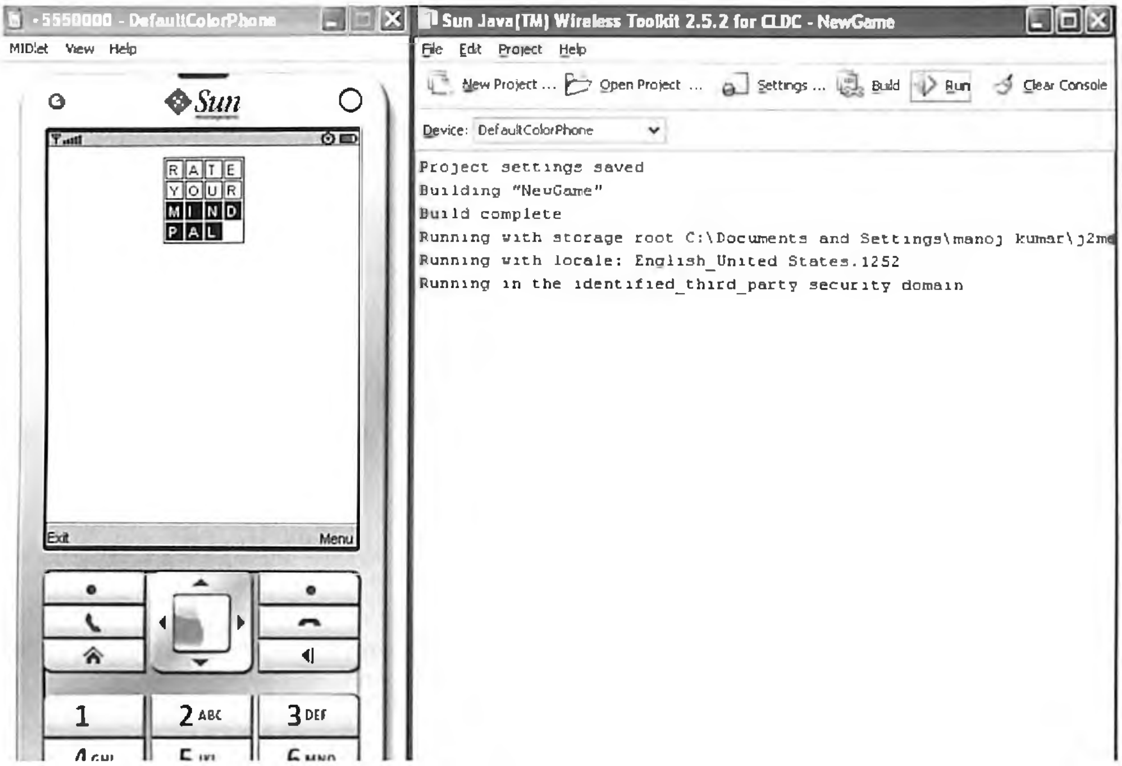


Figure 4-10: Creating a Java Game for mobile phones

4.6 Chapter Conclusions and findings

In this chapter ZigBee based platform has been used for demonstrating the point-to-point communication between two ZigBee transceivers. The platform continues to work in a reliable manner and setting up of the ZigBee communication is easy.

A solar USB charger is implemented as a handy system that helps to make use of the solar energy directly to the gadget or through a reservoir battery.

Two MEMS based boards demonstrate the usefulness of MEMS action in gaming. Porting of ASPIDA is illustrated as it forms part of the future scope of work. Simple Java based games have been developed and demonstrated on common mobile phones available to demonstrate the usefulness of Java in gaming. The next chapter contains complete demonstrations.

5 IMPLEMENTED PLATFORMS

In this chapter the individual units studied separately in the preceding chapters have been combined to create an environment that creates a good scenario for the multiplayer gaming using ZigBee and MEMS. First environment is concept of ad-hoc networking for building automation and limited data transfer, followed by two different environments are created to demonstrate the usefulness and applicability of the ZigBee for gaming.

5.1 Platform 1: ZigBee based Smart Monitoring System

This platform, called "Smart Monitoring System" was introduced in section 4.1.2. "Smart Monitoring System" works on wireless technology – ZigBee and measures temperature, humidity and light intensity of remote and different locations within ZigBee range at a time. The system also demonstrates the wireless control of home appliance and can serve as a basis of many new systems [**103**].

The objective of this evaluation board is to design a complete system which demonstrates the effectiveness of low data rate wireless solutions in automating the various measurements for weather parameters. This system can be further interfaced to an automation system take an appropriate action based on the parameters being monitored. The system can very easily be modified to control various household appliances by suitable additional hardware. This system uses various devices, the important ones are STM32x (Microcontroller), SN260 (ZigBee device), LIS331DLH (MEMS), STMPE811 (Touch screen controller), STLM20 (Temperature sensor) and various power supply components along with battery charger. The system can be powered up using an external power supply or a 3.7V Li-ion rechargeable battery.

The key features of this system are:

- Wireless network
- Weather parameter monitoring (Temperature, Humidity, Light Intensity)
- Wireless light control application
- Individual node movement and simultaneous node movement detection

- Various weather warning configurations and alarming messages
- TFT display with touch screen
- Real time clock along with alarm configurations
- Data logging for more than a year
- GUI for data reading, saving and analysis using graphs.
- Power management system
- Battery charging capability

This system has the possibility of detecting the movement of all the nodes. An interesting real time situation is when all the nodes move at the same time. When used in a remote unmanned site, it can be an indication of a possible earthquake.

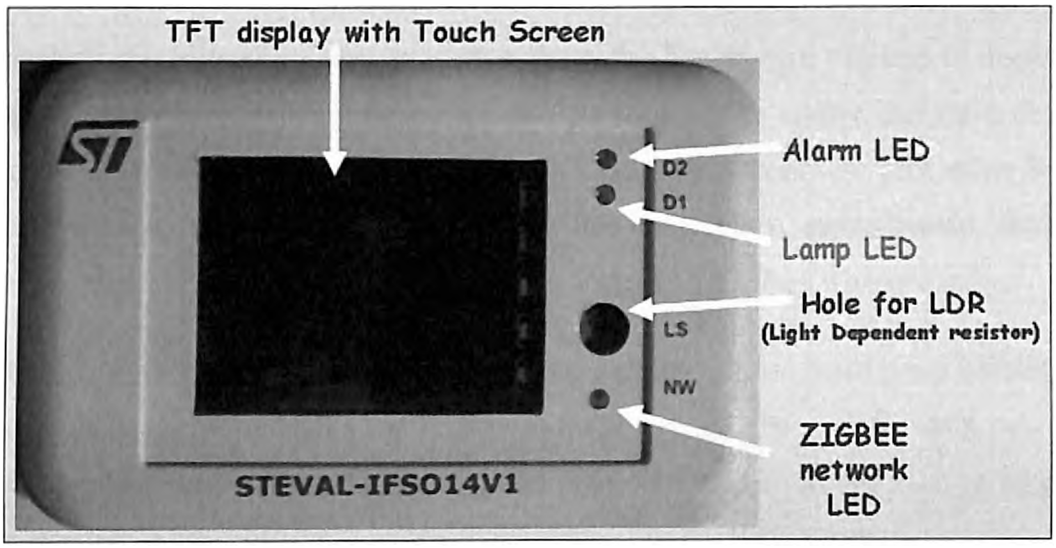


Figure 5-1 : Smart Monitoring System Controller [103]

This has an on-board memory that is used for data logging application. The parameters to be logged like, humidity, temperature and light intensity can be measured. Their threshold values can be defined as also the logging interval. A graphical user interface that imports all this data for the offline analysis has also been developed as part of this.

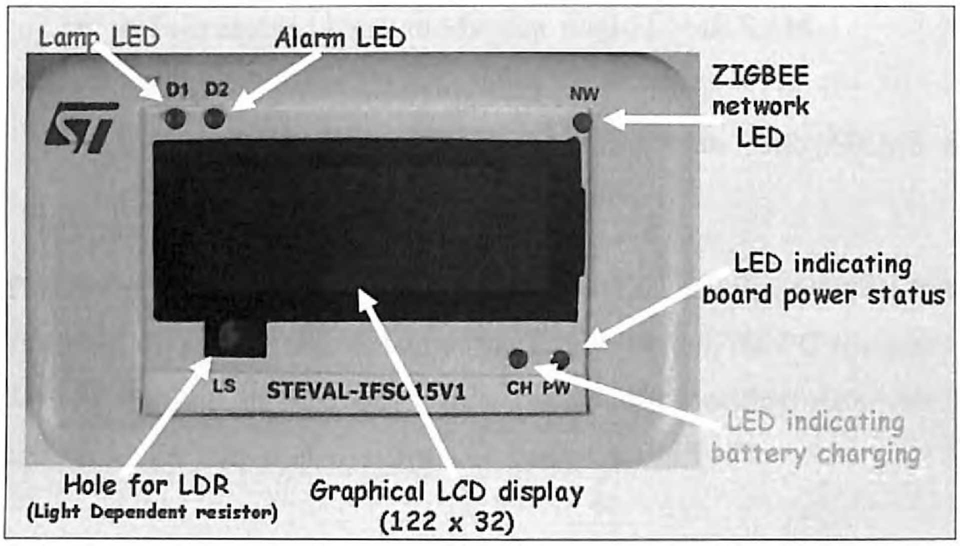


Figure 5-2 : Smart Monitoring System Node [103]

Each of the units i.e. master as well as the nodes has its own displays to display their local temperature, humidity and light intensity. As soon as the master and slave detect the presence of each other through ZigBee transmissions, they authenticate each other based on the authentication mechanism implemented in this case. Once authenticated, the master requests the weather parameters from each of the nodes and displays it accordingly.

This configuration is implemented using star topology. Since there is a possibility of up to 8 nodes being communicating with the master at a given time, all the data may not be displayed on the master screen for all the nodes. Thus node selection has been provided. The logging function works for all the nodes all the time.

With this platform, a secure ZigBee communication is demonstrated. The nodes once authenticated can leave and rejoin the network once these are back in the vicinity. An indoor range of about 25 meters has been found after the units are sealed in the casing as shown in Figure 5-1 and Figure 5-2.

This solution has been designed in the Systems LAB of STMicroelectronics in India. A somewhat similar system has been reported in [**105**].

5.2 Platform 2: Wireless Gaming and Plane Demo

This platform consists of

- A Java enabled Pentium Machine with 512 MB RAM
- An STM32 based ZigBee dongle
- Two STM32 based ZigBee handheld units with MEMS accelerometers integrated

This setup uses the MEMS tilt action [17] and the ZigBee wireless transmission [51] for the purpose of demonstration. To run the demonstration, the PC requires the installation of the Java SE Runtime Environment (JRE). The demo successfully runs with JRE 6.5, which is available from the Sun Microsystems web site.

5.2.1 Wireless Handheld Module

This board is based on the ZigBee module SPZB250 which performs both the RF radio and controlling functions. The controller inside SPZB250 module monitors the acceleration sensor and following the programmed strategy sends the relevant data by means of ZigBee. The ZigBee manages at the same time also all the networking functions.

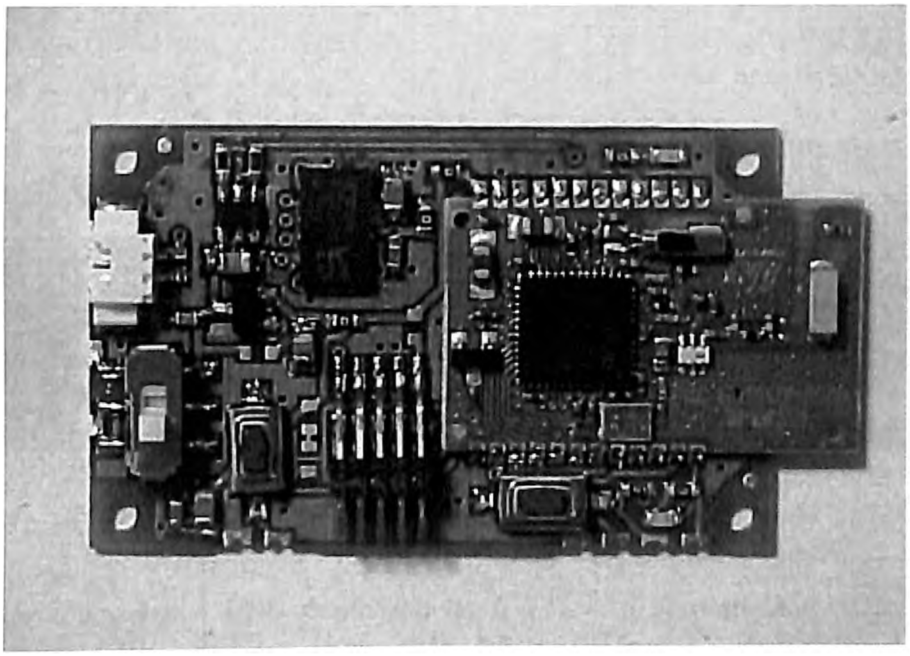


Figure 5-3: ZigBee and MEMS based gaming Platform

Networking capability is made possible by means of the integration of the powerful ZigBee stack inside the same controller. The firmware to the board can be downloaded by means of the SIF connectors provided on the boards. A power supply source needs to be connected, the connection is done by a standard two poles connector and a switch is present in series to the supply connection to power on / off the board. The power supply is by means of 4 AAA batteries. Two push buttons are available on the board, the Reset and the Commissioning button to be used to start the network joining. Two LEDs, one indicating the connection and the other for general purpose are available.

The accelerometer is a 3-axis digital-output linear MEMS sensor that includes a sensing element and an IC interface capable of taking information from the sensing element and providing the measured acceleration signals to external applications through an I2C/SPI serial interface. The LIS3LV02DL has a user-selectable full scale of ±2 g, ±6 g and is capable of measuring acceleration data over a bandwidth of 640 Hz for all axes. The device bandwidth may be selected according to application requirements.

5.2.2 USB-ZigBee Dongle

The dongle is powered from the USB port directly. It can be used to transfer data to the connected operator for data analysis, storage and computation. The dongle uses a USB controller from FTDI Ltd. to communicate with the host computer.

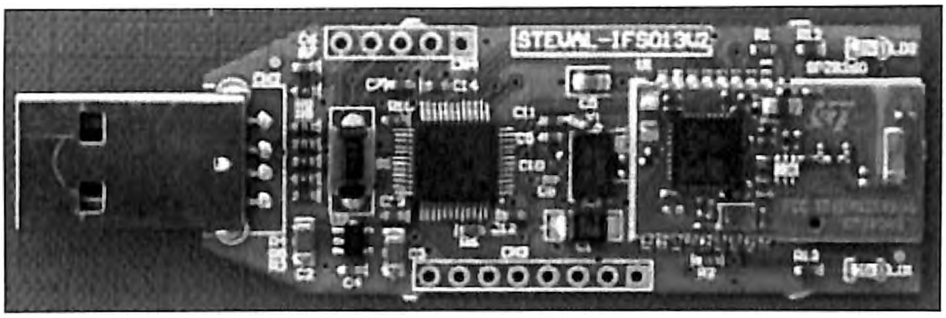


Figure 5-4: The USB-ZigBee Dongle

The dongle integrates a USB controller from FTDI to communicate with the host computer. To communicate with the dongle, the FTDI drivers must be installed on the host. FTDI provides drivers for Windows, Linux, BSD, Macintosh, and Windows CE. Windows users require the virtual COM port (VCP) drivers. The necessary drivers can be downloaded from the driver section of the FTDI Chip web site. The USB dongle is first plugged into an available USB port on the PC. The operating system detects the new hardware and prompts for the location where the driver files can be found.

After installing the driver, the dongle appears as a COM port in the Windows device manager as shown in Figure 5-5. Multiple dongles may be connected to the USB ports of the same computer simultaneously. In this case each dongle will be assigned a different COM port identifier.

5.2.3 The Java Application Framework and driver installation

In order to run the demo, the FTDI driver and the Java based game are required to be installed. The procedure for the same is given below.

After installing the driver, the dongle appears as a COM port in the Windows device manager Installation of the plane demo

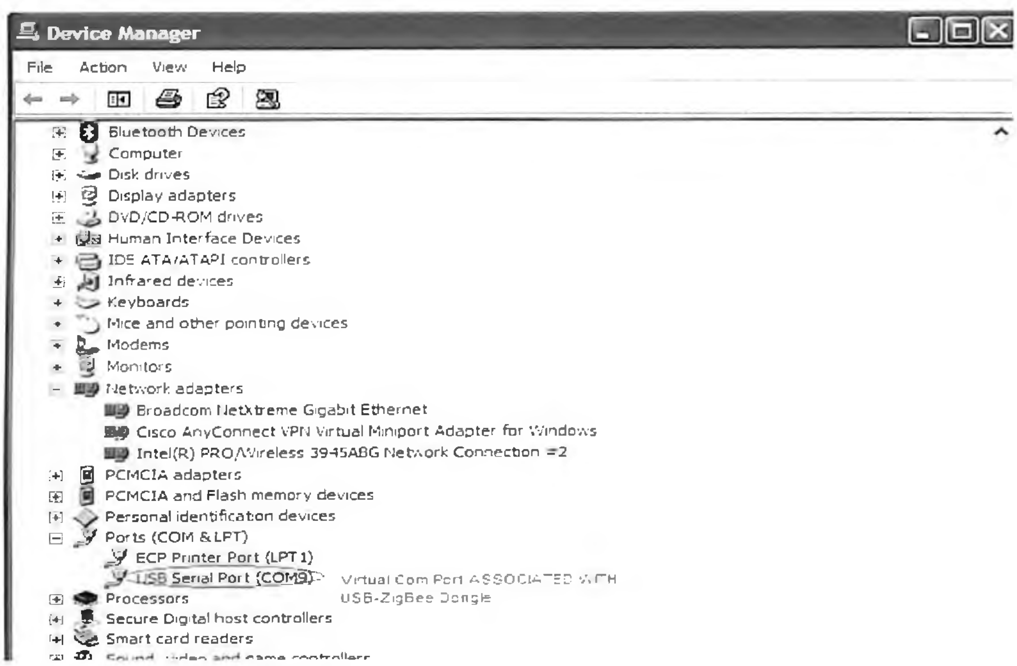


Figure 5-5: USB-ZigBee Dongle associated with a COM port

In order to properly run the Plane demo, the following steps must be followed:

- Install the Java Runtime Environment (JRE) - the Plane demo has been developed in Java and requires the JRE installation on the PC. The demo has been successfully tested with JRE 6.5.
- Install the Java3D package - Java 3D is an additional Java package that enables the creation of three-dimensional graphics applications and Internet-based applets.

5.2.3.1 The plane Demo

The demo includes a program which provides an example of a possible use of the data on the PC to which the dongle is connected. The program demonstrates the tilt measurement capabilities of accelerometers using a simulated aircraft.

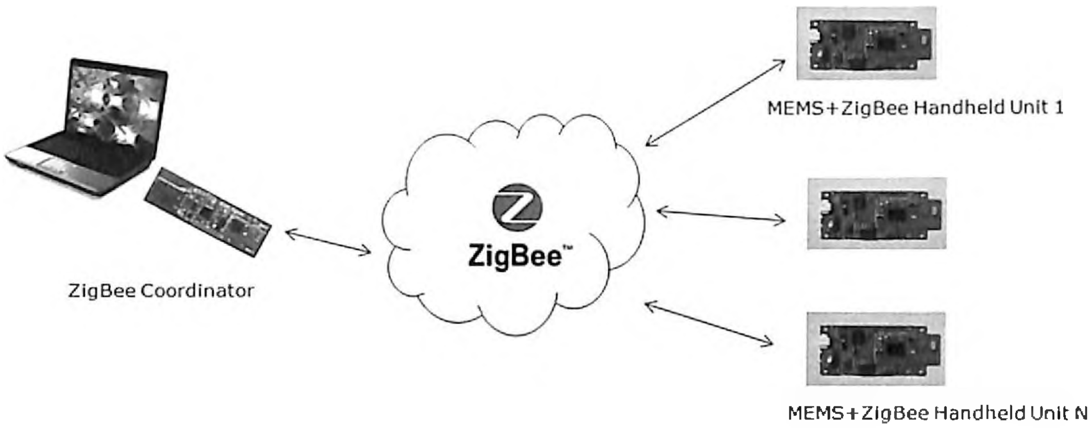


Figure 5-6: The Gaming Environment

By creating a connection with the COM port, several kinds of applications running on the host computer can be used or developed to store, visualize or elaborate sensor data. The sender of each packet can be recognized thanks to the NodeID field that is assigned to each ZigBee handheld unit, Figure 5-3, during the process of joining the network.

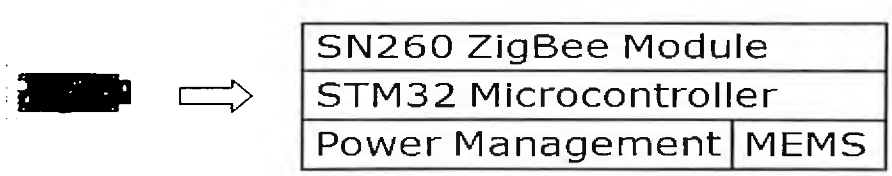


Figure 5-7: Architecture of handheld unit

Figure 5-6 shows the gaming environment. In this case two or more nodes are participating in the game and USB-ZigBee dongle, Figure 5-4, acts as the PAN co-ordinator. The joining, removal and rejoining are controlled by this. The ZigBee module SN260 has been used.

The application shows how accelerometers can be used to measure the tilt of an object. where the tilt is a static measurement and gravity is the acceleration being measured. The force of gravity is used as an input to determine the orientation of an object. calculating its degree of tilt. The x. y. z acceleration values are elaborated on the PC application to evaluate the rotation angles of a graphical object (Plane). The behavior of the demo is intuitive. as the graphical object on the PC monitor moves according to the movements impressed on the handheld ZigBee units.

When the plane demo is successfully installed and application run. the application starts and initial screen is shown in Figure 5-8

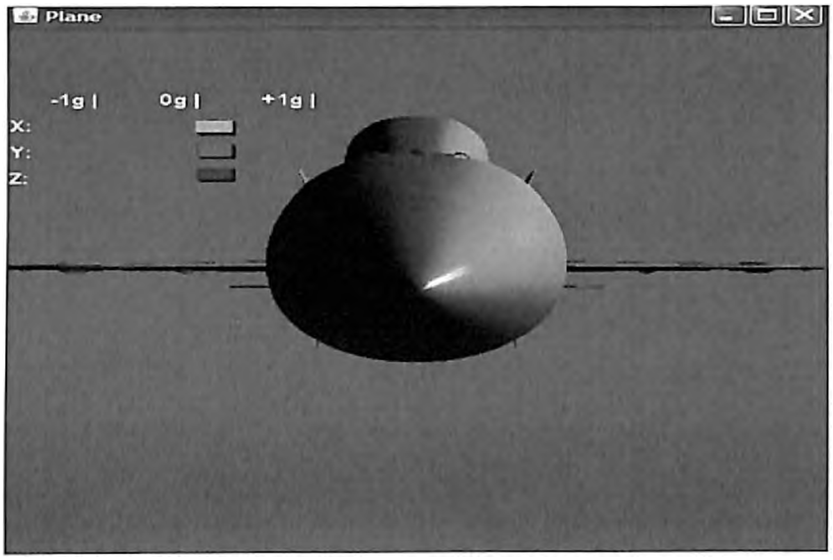


Figure 5-8: The game demo

The ZigBee nodes with MEMS on them can join the USB-ZigBee dongle. Once a node joins successfully an indicator indicates the same. The motion of these nodes is continuously transferred to the dongle and it is reflected in two ways. One that the orientation of the plane changes in the way that user moves the node. Another indication is that tilt experienced in each of the axis is displayed graphically and on the DOS window where one can see the corresponding numerical values.

If another player / node joins the network. the ZigBee sequentially displays the data corresponding to each of the nodes. When the data from the two nodes is similar to one another. the plane is seen to be not flickering.

This game is representative game. This concept has been tried on other free Java games also and the key-mapping concept works almost universally.

The goal of this application is to show how a ZigBee network can be formed and the ease that ZigBee offers for other ad-hoc nodes to join or leave the network. This is elaborated on the PC application that is used to evaluate the rotation angles of a graphical object (Plane). The behavior of the demo is intuitive, as the graphical object on the PC monitor moves according to the movements impressed on the wireless modules. Figure 5-3.

Figure 5-9: Object orientation controlled by ZigBee using MEMS

Up to four wireless modules can send data to the same dongle, Figure 5-4. The firmware is programmed to sample data once in 33 ms. The dongle collects data received from all the available modules and sends to the PC graphical user interface via the COM port that is made available on the USB port after FTDI driver installation.

This experiment also demonstrates the capability of MEMS accelerometers for the wireless handheld gaming. In fact the forward, backward and two sideway keys of a computer keyboard are extended to a handheld gadget and the response time is quite satisfactory and instantaneous. It is faster than what can be observed by human reactions.

5.3 Platform 3: Wireless Gaming, Weather and Presentation Gadget

The concepts presented and demonstrated in previous section are extended further. A network motion sensing and control platform, called ***nMoSeCo*** is created. It uses the same USB-ZigBee dongle as used previously. The key features of this systems are:

- MEMS based Gaming Control
- On-Board MEMS Sensors : Accelerometer, Magnetometer
- Barometric Pressure & Altitude Measurements
- Temperature Sensor Measurements
- Easy USB-ZigBee interface

This system has boards, graphical user interface, presentation pointer and the gaming implementations. These components are briefly described below:

5.3.1 Gaming Controller

The gaming controller (also called Gadget), Figure 5-10, is the battery powered handled portable unit with all the sensors. The board is capable of wireless ZigBee communication with another USB-ZigBee DONGLE. This gaming controller has onboard display to show the sensor data/information and a MENU for control and setting facilities.

The gadget board is 3.3V battery powered. Slide switch SW1 is used to switch on the board. In case, the board indicates "Low battery", connect the board to PC for charging using micro-B USB cable insertion in the on-board USB slot. If the battery is not present, the Gadget board must be powered with a micro USB cable. The gadget displays the welcome message on screen with welcome tone play indicating successful boot up of board.

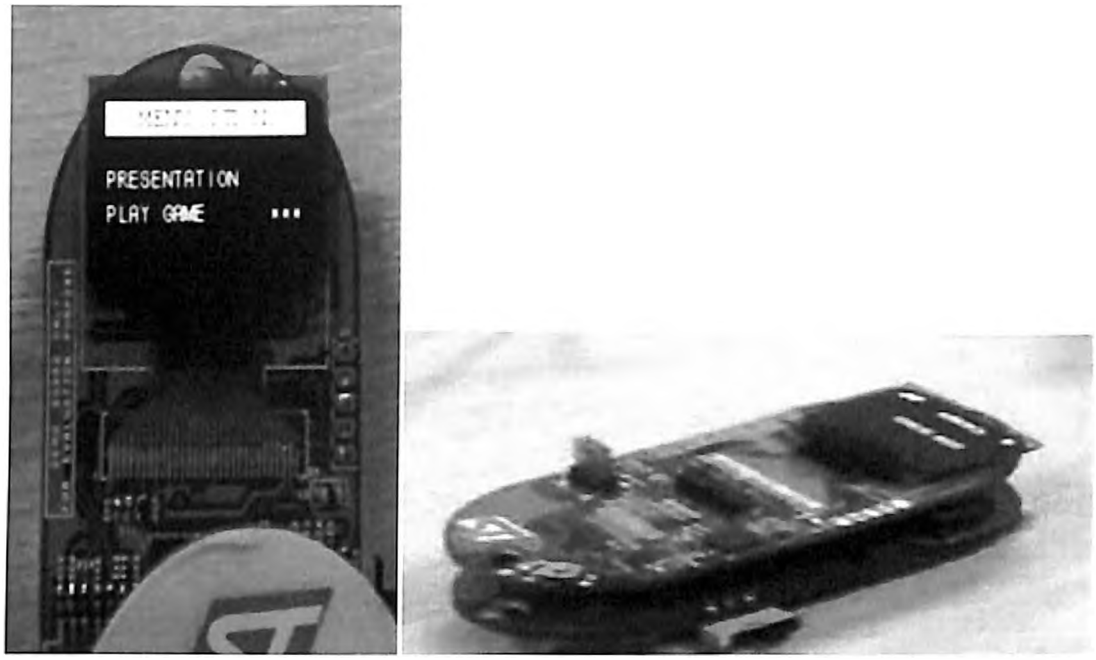


Figure 5-10 : Gaming Controller

5.3.2 Dongle

The dongle is USB powered USB-ZigBee bridge between Computer & Gaming Controller and transfers data/command from the gadget to computer and vice versa.

The dongle board is USB powered. Connect the board to Computer using USB A connector. Removable Disk drive is shown in the "My Computer" indicating successful board boot-up No driver installation is required. The board makes use of Microsoft Windows in-built HID drivers. For USB-ZigBee communication, before powering on the dongle board, make sure that the gadget has completed the boot up sequence.

5.3.3 Graphical User interface

A dedicated graphical user interface (GUI) has been designed for *n-MoSeCo*. The systems requirements for installing the GUI are

Operating System

- Windows XP/Windows 2003 Server /Windows 2008 Server/Windows Vista/Windows 7
- The application is not supported on Windows 98/ME/Windows NT/Windows 2000

Software

- .NET framework 3.5 SP1 must be installed on the system.
- Visual C++ 2008 Runtime SP1 should be installed on the system.
- All critical updates from windows update should be installed.

Hardware

- Minimum 1GB RAM . 500MB HDD space. 2GHz Processor
- Recommended 2GB RAM. 500MB HDD Space. 2Ghz multi-core processor. DX10 Graphics Card

Once the graphical user interface is installed. the following screen appears for the *n-MoSeCo* control centre

Figure 5-11 The n-MosECo Graphical User Interface

The n-MoSeCo control center is very intuitive software to demonstrate the capabilities of the n-MoSeCo device. There are two buttons on the GUI, Figure 5-11, as explained below:

1. **Connect Button**: The connect button establishes the USB connection with the device. Plug-in the dongle device and click connect button before any other operation can be performed.
2. **Disconnect Button**: This button should be clicked before removing the device. This button is also used if there is some device malfunction that warrants a reconnection.

5.3.3.1 Software Operation Modes:

1. **Presentation Mode**: This is the default mode of the operation. It is used to control a power point presentation. This mode should be selected when user is not using the Game Mode or the Sensor Mode. The mode selection is performed on the Gaming Controller.
2. **Game Mode**: Game mode is used to play games. Game Mode is selected on the n-MoSeCo Gaming Controller to switch to the game mode. The gaming action is captured by tilting the Gaming Controller in required directions. It's recommended to go into the game mode only after starting the game and come out of it as soon as game is ended by selecting the Presentation Mode. This is because of the fact the MEMS continue to send the mapped keys data unless the gaming mode is existed.
3. **Sensor Mode**: This mode is used to monitor the sensor data. The sensor mode is selected on the Gaming Controller to enter into this mode. Select the Presentation Mode to come out of the Sensor Mode.

5.3.4 Complete System and Working

The block diagram of the complete system is shown in Figure 5-12.

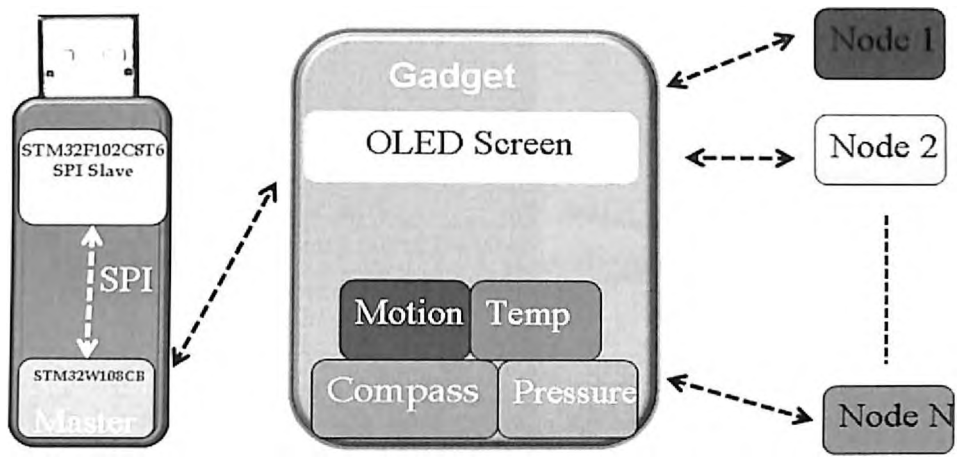


Figure 5-12 The n-MoSeCo System Diagram

The System is designed in a manner that the Gaming Controller or Gadget acts as a master. For the sake of gaming, the data from the gadget is captured based on its movement. The principle of gaming is illustrated by means of readily available games that are typically single player and up to two player keyboard games.

The MEMS motion is captured and converted into equivalent keystrokes in the PC driver. This concept allows the hardware to be used with any available game designed to take keystrokes as input.

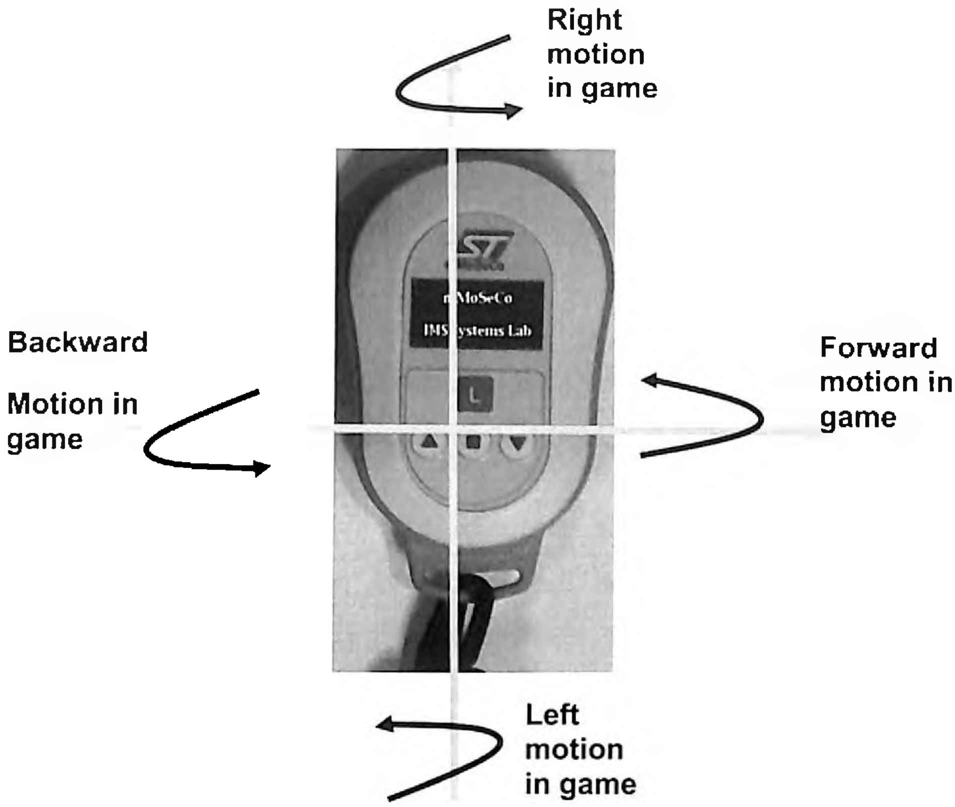


Figure 5-13 : Gaming Controller or Gadget

In this platform, the concept of controlling java based games by ZigBee and MEMS is extended to all available Java based games using computer keyboards that are available in good numbers due to historic evolution of computer based games industry.

5.4 Comparison of platforms for power consumption

In the previous sections, three different platforms have been presented to validate different features of ZigBee and MEMS.

The first platform, section 5.1, demonstrates the ad-hoc nature of ZigBee and nodes join and leave the network depending on distance and other network conditions. The nodes are sending their data to the master in a controlled manner. The master can initiate the data request to the nodes. The display is localized in this case and the batteries need to be charged using an USB charger.

The second platform, section 5.2, is without the local displays. It uses the display of a computer. Once the nodes are authenticated and join the network it is assumed that these are always present in the network range. The nodes continue to send the data regardless of the fact if the data sent be these has been received. In this setup retransmissions make up for any lost packets. This simplification proves to be useful in the gaming scenario as the newer positions must be transmitted as soon as possible. In this case the absence of local displays helps in the longer battery life. The batteries used in this case are AAA size and non-rechargeable.

The last platform, section 5.3 has an OLED display that consumes significantly less power than the LCD display of section 5.1. Besides this it has more number of MEMS sensors that the first case.

The display used in case of section 5.3 consumes 20 mA at 3V. The power consumption by ZigBee device is 25 mA for receive and 29 mA for transmit at 3V. These values are for full screen display at maximum brightness and continuous transmission and reception by the ZigBee device STM32W. Assuming the worst case consumption for ZigBee (29 mA) and full screen brightness for the display, it can be deduced that display accounts for close to 40% of the power. This ratio increases when ZigBee transmission is not continuous. In the system designed, it is possible to send the display in sleep mode when not in operation. Hence it can be deduced that display is quite expensive from battery consumption point of view. Multiplayer games where a common display can be wall-powered will have more battery life for the hand-held units on this account.

Using these three different platforms, it has been observed that the platform without the display required least number of battery changes. The other platforms that were designed were with the rechargeable batteries and different battery charging components and also different kinds of displays.

Owing to several differences between the platforms as mentioned above quantitative analysis and comparison of batteries between the platforms was not carried out. It must be mentioned here that it was possible to use platform 5.3 without OLED display also with some changes in hardware or without mounting of the display component. In this case it was observed that battery used to last longer than case with OLED display being mounted, however these results are not presented as this becomes another subject of investigation on the batteries themselves. Precise computation of this behavior requires consistency between batteries, their charge cycles besides the remaining capacity of each of the different battery.

5.5 Discussions on the results

In the preceding sections, the ZigBee modules interfaced with MEMS accelerometers and STM32 microcontrollers have been used as gaming actors. A Java game developed on the computer that captures the MEMS action and reflects on the plane object is used to prove the gaming concept.

The following are the observations:

- *The network formation of the ZigBee is very easy and simple*
- *Addition of new nodes or players to the existing set-up takes place seamlessly*
- *The response of the plane demo to the MEMS tilting is instantaneous. The measurements of this time were not done but from user experience there is no feeling of wires being not present. This may be partly due to the fact that the distances at which the player nodes were less than 10 meters.*
- *The leaving and rejoining of the nodes is seamless.*
- *The microcontrollers provide excellent programmability for the USB on the computer side and ZigBee on the other.*
- *The Java based platforms offer very easy interface.*
- *The MEMS accelerometers and gyroscopes provide very intuitive and easy to use interface for gaming applications.*
- *The ZigBee combined with the MEMS action and simplicity of microcontrollers provides a very viable alternative to the gaming applications, currently dominated by Bluetooth and Wi-Fi.*

The above observations are in line with the initial hypotheses for the research work. It may be mentioned here that recently there have been other gaming areas where ZigBee and MEMS have been used together. A Study of the Integrated Automated Emotion Music with the Motion Gesture Synthesis via ZigBee Wireless Communication has been carried out using these advantages as reported in **[106]**. The musical psychology literatures are suggested to be used to perform the automated composition, and the system uses emotion as an input parameter to allow users to compose music easily without having to learn the complicated music theory, with the emotion-mapped EM data input instead.

Another development worth mentioning here is interest to minimise the energy requirements of latency sensitive applications, especially Multiplayer Mobile Games without affecting the user experience adversely. An environment for power aware middleware for multiplayer games, called ARIVU has been presented in [92]. It claims to reduce energy consumption by up to 60% of the total energy consumed by the 802.11 g wireless interfaces for First Person Shooting (FPS) games and up to 35% of the total energy for Massively Multiplayer Online Role Playing Games (MMORPG). ZigBee is shown to be the best interface to be used.

5.6 Other Observations

The multiplayer Java games available from various sources and internet that work with keyboard of the computers have been used in this work. Typically the keyboard based games are restricted to two players with Left, Right, Forward and Backward controls. MEMS accelerometers provide these features in a very intuitive manner.

Further it is concluded that the actions of the keys of the computer can be mapped to most of the games. This the USB-ZigBee dongle designed and the MEMS handheld units used in this work interface easily to most of the games available.

This setup was tried using a game based on C++ also and the same results were obtained.

5.7 Proposal for the Efficient Games Platform

In this work the platforms based on the STM32 microcontroller, MEMS and ZigBee have been highlighted. For the gaming scenario, these offer a good choice. In order to optimize the protocols even further, the following platform is proposed.

This platform makes use of Java optimized processor that can run the JVM. For the wireless protocol, ZigBee is quite efficient for the power consumption and the speed of network formation. If the number of players is restricted to 8, which usually is the case, an even more efficient protocol can be used. This protocol that is based on ON-OFF KEYING (OOK) has been already used and tested in the Wireless Toy Vehicle Platform, Figure 4-5. This protocol is very simple, has no encryption and has limited functionality good for ad-hoc cooperative gaming environments. The response time and the translation of MEMS accelerometer and gyroscope are quite acceptable in this platform already tested. This protocol can be suitably modified for simplicity and further ease of formation of the network.

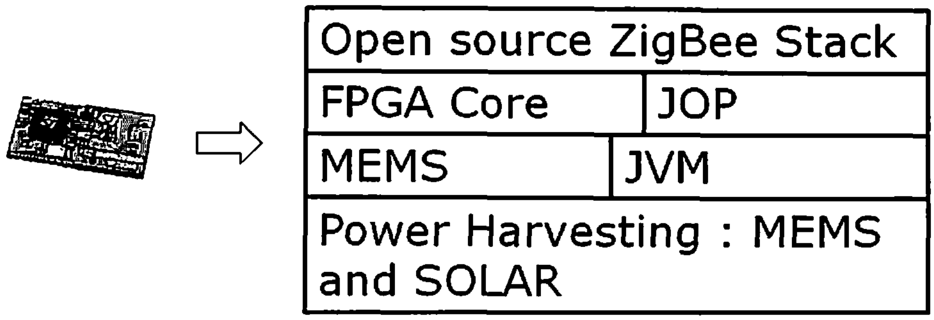


Figure 5-14: Future Platform for Gaming

As a further extension of this work, a platform based on low power and low gate-count FPGA can be used to implement the prototype of the Java Optimized Processor. This implementation can be thereafter used to implement an ASIC to minimize cost. The Java Optimized Processor choice allows JVM implementation in hardware, thereby minimizing memory requirements and maximizing throughput. This gives us a platform where readily available implementations of Java games can be easily ported. The JVM environment can be used to design suitable Java based games that can be ported to this platform. Open source ZigBee stack can be used or else a proprietary and simple wireless protocol can be used to reduce the computing power even further. To increase the efficiency and the battery life of the handheld devices used in outdoor environments, small solar panel can be attached to recharge the batteries. MEMS in future may be used to generate energy or harvest energy from the surroundings. The MEMS action is based on movements that are integral part of gaming actions. Thus the user may be able to add some energy to the platform while playing the game.

This type of platform could be useful in remote monitoring in bad weather conditions. The use of the ad-hoc networks would be that the nodes can stay with the network when the energy is available and are not part of the network when the battery is totally drained.

An application in foreseeable future for the wireless multiplayer gaming is in the vehicle to vehicle communication. Combined with the power of internet and networking sites, the vehicle drivers may be able to get information if their contacts on a common networking site are also in the nearby areas based on the location based services integration to ad-hoc networks.

The body area network in the limited range that has capability to monitor and report various parameters like heart rate, blood pressure monitor and other sensors that can be in the vicinity and can also be implemented effectively using this platform and various concepts presented in this work.

6 CONCLUSIONS AND SCOPE FOR FUTURE WORK

6.1 Conclusions

This work includes system design aspects of multiplayer gaming using wireless networks. In order to arrive at the final choice in each of the aspects. possible aspects have been studied and most relevant choice made or suggested.

For choosing suitable protocol. a comparative analysis of the available wireless protocols suitable for wireless multiplayer gaming has been done. Bluetooth. Wireless LAN and ZigBee have been compared for multiplayer gaming. Comparative analysis of these protocols for data rates, range, networking nodes, ease of forming the network and battery life has been presented. ZigBee has been found to be the most suitable protocol for multiplayer wireless control-oriented gaming environments where the data rate requirements are low. power consumption is less and a node is able to enter into and exit from a network quickly.

The platform choices for multiplayer gaming applications have been compared next. Advantages of asynchronous clock-less processors over the synchronous ones for the battery powered applications have been highlighted. The Java optimized processor, called JOP has been studied next. The small core for JOP, availability of opensource code of its core and more importantly its Java capability have been found to be the major features relevant to our approach. JOP source code has been ported to an FPGA platform and its thermal power dissipated using the Altera tool has been calculated. The STM32, a 32 bit microcontroller based on ARM Cortex-M3. has been studied and chosen as the platform to prove the concepts. This microcontroller does not offer support for Java in its present core. In the scenarios presented Java has been used in the common display resident on computer.

. Material properties of Silicon Carbide diodes, a new type of emerging devices. have been found to be promising for future use in the efficient chargers and converters.

For multiplayer wireless games with common display. several possibilities of operating environments have been studied. Only the Java part has been reported in this work as this was the environment chosen. Possibilities of using other operating systems were investigated but have been excluded from this work as reasonable ready material exists in literature and web sources for these environments. The Java platforms have been focused in this work of multiplayer gaming as there is a large pool of such games software already developed for Java compatible platforms, that has been considered as pool available for Java based gaming.

MEMS have been studied and used in this work to extend the user experience in multiplayer gaming environments. Accelerometers and gyroscope have been used in various sub-systems and final platforms. An integrated platform that uses Java environment on the computer, ZigBee connectivity and MEMS accelerometers on the handheld units has been used to validate the multiplayer wireless gaming concept.

It must be mentioned here that the concepts validated in this work are applicable to other wireless technologies. It is necessary to consider constraints related to range and power consumption of the respective protocols. The unique advantage of ZigBee has been the seamless networking and low power consumption.

An interesting by-product of this work has been discovery of a vast body of the Java games software already available for the computers which could be modified to accept the inputs from MEMS on the remote wireless devices through minor modifications. This greatly reduced the effort in proving the concept.

We thus present techniques for complete system design focusing on several design aspects relevant to multiplayer gaming.

6.2 Scope for future work

The future extension of this work can be towards lesser power consumption by using a more efficient core. An FPGA platform with Java Optimized Processor and JVM ported can result into further power savings. In order to save the cost, ZigBee from open sources can be used or a proprietary and simpler protocol designed. The gate count of the system designed in this manner is expected to be low as the published results for JOP indicate. An ASIC implementation of this platform may provide a low cost and dedicated platform for wireless multiplayer gaming applications.

The additional benefit of JOP and JVM would be that a game designed on a computer can be downloaded to this embedded target. With suitable display interfaced to this target, a peer to peer network can be established resulting into multiplayer environment without needing a computer or a TV monitor. Vehicle to vehicle communication is another area that can be further investigated by use of ad-hoc networks for various use cases.

The user experience for gaming has greatly changed in recent years with the introduction of MEMS. Haptic is an emerging touch sensitive technology that is proposed to change the user experience in the coming years. A haptic touch screen can be visualized in the future gaming platforms.

The Inertial Measurement Units (IMU) are a great step forward. These are already available as system-on-board now and the trend is to have these on a single system-in-package. Several sensors then provide relevant data to the computing units through Sensor Fusion algorithms. IMUs and Sensor Fusion / Data Fusion algorithms will enhance the gaming experience further and find applications in several related areas.

The new fields of energy harvesting are emerging and one of these is energy harvesting from the motion. The Motion sensor based energy harvesting combined with solar energy usage in the handheld devices will present a good combination in the gaming devices.

One could then visualize wireless gaming gadgets that have simple and versatile platforms, capable of making use of already available techniques in gaming and other areas, rich in user experience through motion sensors for representing actions and energy harvesting, energy efficient by using new low loss materials for their energy needs and using renewable energy when feasible to supplement the regular energy sources.

7 BIBLIOGRAPHY

[1] IEEE Standard. IEEE Standard for Information Technology- Local and Metropolitan Area Networks- Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs) , 2006. Revision of IEEE Std 802.15.4-2003.

[2] Y. XIAO. X. SHAN. and Y. REN. "Game Theory Models for IEEE 802.11 DCF in Wireless Ad Hoc Networks." *IEEE Radio Communications*. pp. 22-26. March 2005.

[3] V. Srinivasan, P. Nuggehalli, C.F. Chiasserini. and R.R. Rao. "Cooperation in wireless ad hoc networks ." in *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications*. 2003, pp. 808-817.

[4] P. Michiardi and R. Molva. "Analysis of Cooperation Strategies in Mobile Ad hoc Networks with Imperfect Monitoring." Institut Eurécom. Sophia-Antipolis, N° RR-04-099, 2004.

[5] S. Mehta and K.S. Kwak, "An energy-efficient MAC protocol in wireless sensor networks: a game theoretic approach," *EURASIP Journal on Wireless Communications and Networking - Special issue on theoretical and algorithmic foundations of wireless ad hoc and sensor networks* , pp. 1-10. April 2010.

[6] L. Zhao and J. Zhang. "Using Incompletely Cooperative Game Theory in Wireless Mesh Networks." *IEEE Network*. pp. 39 - 44, Jan 2008.

[7] Broadcom Advances the Wireless Gaming Experience With Bluetooth® and Wi-Fi® Technologies Integrated Into Nintendo's Wii™ Game Console. [Online], http://www.broadcom.com/press/release.php?id=854744. last accessed Nov 2011.

[8] L.D. Paulson, "Will fuel cells replace batteries in mobile devices? ," *IEEE Computer*,

pp. 10 - 12, Nov 2003.

[9] A. Bhojan, A. L. Ananda, M. C. Chan, L.T. Long, and R.K. Balan, "Game Action Based Power Management for Multiplayer Online Game," in *First ACM SIGCOMM Workshop on Networking, Systems, Applications on Mobile Handhelds*, 2009, pp. 55-60.

[10] K. Naik, "A Survey of Software Based Energy Saving Methodologies for Handheld Wireless Communication Devices," Dept. of ECE, University of Waterloo, Tech. Report No. 2010-13, 2010.

[11] L. Chao, C. Y. Tsui, and W. H. Ki, "A Batteryless Vibration-based Energy Harvesting System for Ultra Low Power Ubiquitous Applications," in *IEEE International Symposium of Circuits and Systems*, May 2007, pp. 1349-1352.

[12] S. Chalasani and J.M. Conrad, "A survey of energy harvesting sources for embedded systems ," in *IEEE Southeastcon, 2008.*, 2008, pp. 442 - 447.

[13] J.A. Paradiso and T. Starner, "Energy scavenging for mobile and wireless electronics," *IEEE Pervasive Computing*, pp. 18-27, March 2005.

[14] P.D. Mitcheson, E.M. Yeatman, G.K. Rao, A.S. Holmes, and T.C. Green, "Energy Harvesting From Human and Machine Motion for Wireless Electronic Devices ," *Proceedings of the IEEE* , pp. 1457 - 1486 , Sept 2008.

[15] A.Y. Wang and C.G. Sodini, "On the Energy Efficiency of Wireless Transceivers," in *IEEE International Conference on Communications*, 2006, pp. 3783 - 3788.

[16] R. Zheng and R. Kravets, "On-demand power management for ad hoc networks," in *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications*, 2003, pp. 481-491.

[17] M. Kumar, "MEMS : Theory and Usage in Industrial and Consumer Applications," in *National Conference on Computational Instrumentation, Technical Session : Embedded and Intelligent Instrumentation*, 2010, pp. 1-6.

[18] G. Niezen and G.P. Hancke, "Gesture recognition as ubiquitous input for mobile phones," in *Devices that Alter Perception*, 2008, pp. 16-20.

[19] J. Bernardes, R. Nakamura, and R. Tori, "Design and Implementation of a Flexible Hand Gesture Command Interface for Games Based on Computer Vision," in *VIII Brazilian Symposium on Games and Digital Entertainment*, 2010, pp. 64 - 73.

[20] V. Clincy and B. Wilgor, "Performance of Cloud-based P2P Game Architecture," in *ICNS 2013 : The Ninth International Conference on Networking and Services*, 2013, pp. 44-46.

[21] S. Xinogalos, K.E. Psannis, and A. Sifaleras, "Recent Advances Delivered by HTML 5 in Mobile Cloud Computing Applications: A Survey," in *Proceedings of the Fifth Balkan Conference in Informatics* , 2012, pp. 199-204.

[22] Y. L. Morgan and T. Kunz, "A proposal for an ad hoc network QoS gateway," in *Proceedings of the IEEE International Conference on Wireless Mobile Computing, Networking and Communications*, vol. 3, 2005, pp. 221-228.

[23] S. Wu and Y. Tseng, *Wireless AD HOC Networking : Personal-Area, Local-Area, and the Sensory-Area Networks.*: Auerbach Publications, 2007.

[24] C. -K. Toh, M. Delwar, and D. Allen, "Evaluating the Communication Performance of an Ad Hoc Wireless Network," *IEEE Transactions on Wireless Communications*, pp. 402-414, July 2002.

[25] C. -K. Toh, *Ad Hoc Mobile Wireless Networks : Protocols and Systems.*: Pearson Education, 2007.

[26] A. B. MacKenzie and S. B. Wicker, "Game Theory and the Design of Self-configuring, Adaptive Wireless Networks," *IEEE Communications Magazine*, pp. 126-131, Nov 2001.

[27] J. Ammer and J. Rabaey, "Low Power Synchronization for Wireless Sensor Network

Modems," in *IEEE Wireless Communications and Networking Conference*, 2005, pp. 670-675.

[28] N. Mohamudally, "A Massive Multiplayer Game Framework for Mobile Learning," in *Fourth IEEE International Workshop on Wireless, Mobile and Ubiquitous Technology in Education*, 2006, pp. 23-25.

[29] T. Ahola et al., "Wearable FPGA Based Wireless Sensor Platform," in *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2007, pp. 2288 - 2291.

[30] Y. Ha, "Dynamic Integration of Zigbee Home Networks into Home Gateways Using OSGi Service Registry," *IEEE Transactions on Consumer Electronics*, pp. 470-476, May 2009.

[31] X. Miao, X. Zhou, and H. Wu, "A cooperative differential game model based on transmission rate in wireless networks," *Operations Research Letters*, pp. 292-295, July 2010.

[32] Z. Han and H.V. Poor, "Coalition games with cooperative transmission: a cure for the curse of boundary nodes in selfish packet-forwarding wireless networks," *IEEE Transactions on Communications*, pp. 203-213, Jan 2009.

[33] S.Z.A. Gok, Cooperative Interval Games, A Ph.D. Thesis submitted to the Graduate School of Applied Mathematics of Middle East Technical University, 2009.

[34] G. Shi et al., "Towards an ubiquitous motion capture system using inertial MEMS sensors and ZigBee network," in *IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems*, 2011, pp. 230 - 234.

[35] J. Hua et al., "MEMS and J2ME based acceleration real-time measurement and monitoring system for fuel cell city bus," in *IEEE International Conference on Mechatronics*, 2009, pp. 1 - 6.

[36] G. Pablo, C. Mariano, and B. Alejandro, Relying on Wireless Sensor Networks to

Enhance the RC-Gaming Experience. [Online]. www.dvs.tu-darmstadt.de/publications/pdf/wsntoys.pdf. last accessed Nov 2011.

[37] A. I. Wang. M.S. Norum. and C.W. Lund. "Issues related to Development of Wireless Peer-to-Peer Games in J2ME," in *Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services*. 2006. pp. 115-115.

[38] M. Chauhan. Developing Java-Based Mobile Games. [Online]. http://www.developer.com/java/j2me/article.php/10934_3502741_1. last accessed Nov 2011.

[39] K. Gerd et al., "When Peer-to-Peer comes Face-to-Face: Collaborative Peer-to-Peer Computing in Mobile Ad hoc Networks," in *First International Conference on Peer-to-Peer Computing*. 2001, pp. 75-91.

[40] A. I. Wang. "Mobile Peer-to-peer Collaborative Framework and Applications," in *Mobile Peer-to-peer Computing for Next Generation Distributed Environments.*: IGI Global. 2009. ch. 19, pp. 416-436.

[41] A. I. Wang and C. Sørensen. "Mobile Peer-to-Peer Technology used to Promote Spontaneous Collaboration," in *IEEE : Proceedings of the 2005 International Symposium on Collaborative Technologies and Systems*. 2005, pp. 125-136.

[42] A. Muneeb et al., "Medium Access Control Issues in Sensor Networks," *ACM SIGCOMM Computer Communication Review*. pp. 33-36, April 2006.

[43] IEEE Standard, IEEE Standard for Information Technology - Local and Metropolitan Area Networks - Specific Requirements. - Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specification for Wireless Personal Area Networks (WPANs) , 2005. Revision of IEEE Std 802.15.1-2002.

[44] C. Gomez, J. Oller. and J. Paradells. "Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology." *Sensors*. pp. 11734 - 11754. August 2012.

[45] K. Mikhaylov, N. Plevritakis, and J. Tervonen, "Performance Analysis and Comparison of Bluetooth Low Energy with IEEE 802.15.4 and SimpliciTI," *Journal of Sensor and Actuator Networks*, pp. 589-613, August 2013.

[46] IEEE Standard, IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2007.

[47] P. N. Anggraeni, S. A. Wardana, R. H. P. Fitzek, and M. Katz, "Energy Efficient Medium Access Control Protocol for Cooperative Wireless Networks," in *IEEE Annual Conference*, 2008, pp. 1-5.

[48] L. Chen and J. Leneutre, "On the Power and Rate Control in IEEE 802.11 WLANs - A Game Theoretical Approach ," in *Proceedings of 16th International Conference on Computer Communications and Networks*, 2007, pp. 450 - 456.

[49] A. Willig, K. Matheus, and A. Wolisz, "Wireless Technology in Industrial Networks," in *Proceedings of the IEEE*, June 2005, pp. 1130-1151.

[50] D. Geer, "Users Make a Beeline for ZigBee Sensor Technology," *IEEE Computer*, pp. 16-19, Dec 2005.

[51] M. Kumar, "ZigBee : The Low Data Rate wireless technology for ad-hoc and sensor networks," in *National Conference on Computational Instrumentation, Technical Session : Sensor , Wireless Networking for Agro and Health Sector*, March 2010, pp. 138-144.

[52] G. Roberts, IEEE 802.15 Overview of WG and Task Groups, 2003, [Online], http://ieee802.org/15/pub/Tutorials.html, last accessed Nov 2011.

[53] J. Lee and Y. Huang, "ITRI ZBnode: A ZigBee/IEEE 802.15.4 Platform for Wireless Sensor Networks," in *IEEE International Conference on Systems, Man and Cybernetics*, 2006, pp. 1462-1467.

[54] M. Petrova. J. Riihijarvi. P. Mahonen. and S. Labella. "Performance Study of IEEE 802.15.4 Using Measurements and Simulations," in *Wireless Communications and Networking Conference*. 2006, pp. 487-492.

[55] B. A. Abdelhakim. B. Mohammed. and B. Abderrahim. "Quality of Service and Communication Technologies for Wireless Multimedia Sensor Networks." in *17th Telecommunications forum TELFOR*, 2009. pp. 254-257.

[56] ZigBee Alliance. ZigBee Specifications. 2005. ZigBee Document 053474r06. Version 1.0.

[57] F. Potorti and E. Ferro, "Bluetooth and IEEE 802.11 Wireless Protocols: A Survey and a Comparison," *IEEE Wireless Communications*. pp. 12-26, February 2005.

[58] J. Lee. Y. Su, and C. Shen. "A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," in *33rd Annual Conference of the IEEE Industrial Electronics Society*. 2007. pp. 46 - 51.

[59] ZigBee Alliance Website. [Online], http://zigbee.org/About/FAQ.aspx. last accessed June 2014.

[60] G. Fyke, Unlocking the Potential of "Big Data" in Low-Power Wireless Sensor Networks, 2013. [Online], http://www.rtcmagazine.com/articles/view/102975. last accessed June 2014.

[61] A. Lundgren. Software Development Tools Optimize ZigBee Performance. 2012, [Online], http://www.wirelessdesignmag.com/blogs/2012/09/software-development-tools-optimize-zigbee-performance, last accessed June 2014.

[62] J. Song et al., "WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control." in *IEEE Real-Time and Embedded Technology and Applications Symposium*. April 2008, pp. 377-386.

[63] S. Selvakennedy, S. Sinnappan, and Y. Shang, "T-ANT: A Nature-Inspired Data Gathering Protocol for Wireless Sensor Networks." *JOURNAL OF*

COMMUNICATIONS (JCM), pp. 22-29, May 2006.

[64] T. Lennvall, S. Svensson, and F. Hekland, "A Comparison of WirelessHART and ZigBee for Industrial Applications," *Proceedings of the IEEE International Workshop on Factory Communication Systems*, pp. 85-88, May 2008.

[65] C. Jyh-Cheng, M. S. Krishna, A. Prathima, and K. Shalinee, "A Comparison of MAC Protocols forWireless Local Networks Based on Battery Power Consumption," in *Proceedings of IEEE INFOCOM, 1998*, 1998, pp. 150-157.

[66] D. Geer, "Is it time for clockless chips? [Asynchronous processor chips]," *IEEE Computer*, pp. 18-21, March 2005.

[67] H.K. Kapoor, "Formal Modelling and Verification of an Asynchronous DLX Pipeline," in *Proceedings of the Fourth IEEE International Conference on Software Engineering and Formal Methods*, 2006, pp. 118-127.

[68] D. A. Patterson and J. L. Hennessy, *Computer Architecture : A Quantitative Approach.*: Morgan Kaufmann, 1996.

[69] M. Schoeberl, *JOP reference handbook.*: Create Space, 2009.

[70] JOP - Java Optimized Processor. [Online], http://www.jopdesign.com/, last accessed Nov 2011.

[71] C. Pitter and M. Schoeberl, "Performance evaluation of a java chip-multiprocessor," in *International Symposium on Industrial Embedded Systems*, 2008, pp. 34 - 42.

[72] W. Binder, M. Schoeberl, P. Moret, and A. Villazon, "Cross-Profiling for Embedded Java Processors," in *Fifth International Conference on Quantitative Evaluation of Systems*, 2008, pp. 287 - 296.

[73] T. Harmon, R. Kirner, M. Schoeberl, and R. Klefstad, "A Modular Worst-case Execution Time Analysis Tool for Java Processors," in *IEEE Real-Time and Embedded Technology and Applications Symposium*, 2008, pp. 47 - 57.

[74] R. Pila, M. Kumar, and V. Thaplial, Next-gen flexible MCU platforms enable faster time-to-market, 2010. [Online]. http://www.ednindia.com/article-26741-nextgenflexiblemcuplatformsenablefastertimetomarket-Asia.html, last accessed Nov 2011.

[75] STM8S and STM8A microcontroller families, STM8 Microcontroller reference manual RM0016, [Online]www.st.com, last accessed Nov 2011.

[76] Cortex M3 Technical Reference Manual, [Online] , Revision r1p1, from www.arm.com, last accessed Nov 2011.

[77] STM32F100xx advanced ARM-based 32-bit MCUs, STM32 Reference Manual RM0041, [Online] www.st.com, last accessed Nov 2011.

[78] STM32W : High-performance, IEEE 802.15.4 wireless system-on-chip, [Online], http://www.st.com/internet/mcu/subclass/1377.jsp, last accessed Nov 2011.

[79] J.W. Muchow, *Core J2ME Technology and MIDP.*: Prentice Hall, 2001.

[80] The basics of Java mobile development, [Online]. http://www.morkalork.com/mork/article/79/How_to_create_a_basic_mobile_application.htm, last accessed Nov 2011.

[81] The J2ME Mobile Information Device Profile 2.0, 2003, [Online], http://www.developer.com/ws/j2me/article.php/1574791, last accessed Nov 2011.

[82] N. Leavitt, "Will wireless gaming be a winner? ," *IEEE Computer* , pp. 24-27, Jan 2003.

[83] Why Choose Java For Mobile Game Development?, [Online], http://www.developer.nokia.com/Community/Wiki/Why_Choose_Java_For_Mobile_Game_Development%3F, last accessed Nov 2011.

[84] Sun Java Real-Time System, [Online].

http://java.sun.com/javase/technologies/realtime/index.jsp, last accessed Nov 2011.

[85] P. Sione, Real-time Specification for Java (RTSJ), [Online], http://www.developer.com/java/other/article.php/10936_1367671_2/Real-time-Specification-for-Java-RTSJ.htm, last accessed Nov 2011.

[86] B. Brosgol et al., *The real time specifications for Java.*: Addison-Wesley, 2000.

[87] Game Development - J2ME, [Online], http://www.profmax.com/pbs/web_tutorials/game_development_j2me/game%20development%20j2me.aspx, last accessed Nov 2011.

[88] J. Draney, HTML5 and the Future of Online Games, 2012, [Online], http://www.sitepoint.com/html5-and-the-future-of-online-games/, last accessed 20th June 2014.

[89] J. Allen, Build your own online real-time acquisition system – HTML5 application in embedded systems, 2014, [Online], http://wizwiki.net/build-your-own-online-real-time-acquisition-system/, last accessed 20th June 2014.

[90] Create World Class HTML5 Apps & Web Apps with the XDK, [Online], http://dev.html5m.com/index.php?q=content/create-world-class-html5-apps-web-apps-xdk, last accessed 20th June 2014.

[91] W3C, HTML Working Group, 2014, [Online], http://www.w3.org/html/wg/, last accessed 20th June 2014.

[92] B. Anand, A.L. Ananda, M.C. Chan, and R.K. Balan, "ARIVU: Making Networked Mobile Games Green: A Scalable Power-Aware Middleware," *Mobile Networks and Applications*, pp. 21-28, 2012.

[93] B. Anand et al., "PGTP: Power Aware Game Transport Protocol for Multi-Player Mobile Games," in *Communications and Signal Processing (ICCSP)*, 2011, pp. 399-404.

[94] How do Photovoltaics Work?, [Online]. http://science.nasa.gov/science-news/science-at-nasa/2002/solarcells/, last accessed Nov 2011.

[95] M. Kumar and K. Saha, ST and Solar Photovoltaic, Indo-Canada Workshop hosted by the Indian Institute of Technology Delhi on Electricity Generation using Renewable Energy,2009. [Online]. http://www.wise.uwaterloo.ca/pdf/ISTP-India-Oct09/Kumar_Oct09_Delhi.pdf, last accessed Nov 2011.

[96] A guide to solar power your home. [Online], http://www.generators.smps.us/solarpower.html, last accessed Nov 2011.

[97] M. Mazzillo et al., "4H-SiC Schottky Photodiode Based Demonstrator Board for UV-Index Monitoring," *IEEE SENSORS JOURNAL*, pp. 377-381, Feb 2011.

[98] J. Bryzek, *Principles of MEMS.*: John Wiley & Sons, 2005.

[99] C. Fabrizio, Mechanical characterization and simulation of fracture processes in polysilicon Micro Electro Mechanical Systems (MEMS), Ph. D. Thesis, Politecnico di Milano, 2007.

[100] A. Yurgens, Micromachining and MEMS Devices, [Online]. http://fy.chalmers.se/~yurgens/FKA196/lectures/Micromachining.pdf, last accessed Nov 2011.

[101] S. Beeby, G. Ensell, M. Kraft, and N. White, *MEMS Mechanical Sensors.*: Artech House, 2004.

[102] N. Abbate, A. Basile, C. Brigante, A. Faulisi, and F. La Rosa, "Modern Breakthrough Technologies Enable New Applications Based on IMU Systems," *Journal of Sensors*, 2011.

[103] M. Mangal and N. Omar, Picking the right technologies for your home network design, 2009. [Online]. http://www.eetimes.com/design/embedded/4008833/Picking-the-right-technologies-for-your-home-network-design?pageNumber=0, last accessed

Nov 2011.

[104] C. Brigante. N. Abbate. A. Basile, A.C. Faulisi, and S. Sessa, "Towards Miniaturization of a MEMS-Based Wearable Motion Capture System." *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*. vol. 58, no. 8. pp. 3234-3241, August 2011.

[105] K. Gill, S. Yang, F. Yao. and Xin Lu. "A ZigBee-Based Home Automation System." in *IEEE Transactions on Consumer Electronics*. 2009. pp. 422 - 430.

[106] C. Huang and W. Nien. "A Study of the Integrated Automated Emotion Music with the Motion Gesture Synthesis via ZigBee Wireless Communication," *International Journal of Distributed Sensor Networks*, 2013.

APPENDICES

APPENDIX A

LIST OF PUBLICATIONS

[1] M. Kumar, "ZigBee : The Low Data Rate wireless technology for ad-hoc and sensor networks," in National Conference on Computational Instrumentation, Technical Session : Sensor ,Wireless Networking for Agro and Health Sector, March 2010, pp. 138-144.

[2] R. Pila, M. Kumar, and V. Thaplial, Next-gen flexible MCU platforms enable faster time-to-market, 2010, [Online], http://www.ednindia.com/article-26741-nextgenflexiblemcuplatformsenablefastertimetomarket-Asia.html, last accessed Nov 2011.

[3] M. Kumar, "MEMS : Theory and Usage in Industrial and Consumer Applications," in National Conference on Computational Instrumentation, Technical Session : Embedded and Intelligent Instrumentation, 2010, pp. 1-6.

[4] M. Kumar and K. Saha, ST and Solar Photovoltaic, Indo-Canada Workshop hosted by the Indian Institute of Technology Delhi on Electricity Generation using Renewable Energy,2009. [Online], http://www.wise.uwaterloo.ca/pdf/ISTP-India-Oct09/Kumar_Oct09_Delhi.pdf, last accessed Nov 2011.

[5] M. Mazzillo, P.Shukla, R. Mallik, M. Kumar, R. Previti, and G. Di Marco, "4H-SiC Schottky Photodiode Based Demonstrator Board for UV-Index Monitoring," *IEEE SENSORS JOURNAL*, pp. 377-381, Feb 2011.

[6] R. M. Imayavaramban, R. Mallik, and M. Kumar, "Solar Mobile Charging Solutions". [Online]. http://electronicsmaker.com/em/admin/pdfs/free/Solar%20Mobile.pdf, last accessed June 2014

[7] M. Kumar and P. Pandey. "Wireless Gaming platform based on Motion MEMS and ZigBee". [Online], http://www.st.com, last accessed June 2014.

APPENDIX B

AUTHOR BIOGRAPHY

Manoj Kumar is working with STMicroelectronics India at Greater Noida since 1999. He is currently leading the System LAB of STMicroelectronics in India. The System LAB designs reference designs and demos based on ST products in the areas of MEMS, Microcontrollers, Power Systems, Low Data Rate Wireless, Industrial Automation, Lighting and Motor Control. He has been associated with Wireless LAN and Bluetooth activities of STMicroelectronics in the past. Previously he worked with Central Research Laboratory (CRL) of Bharat Electronics Limited (BEL), Ghaziabad. He completed his B.E. in Electronics Engineering from Visvesvaraya National Institute of Technology (VNIT, previously VRCE) Nagpur in 1992 and M.Tech. in Electronics Engineering from Indian Institute of Technology, Banaras Hindu University (IIT-BHU, previously IT-BHU) in 1995.

APPENDIX C

SUPERVISOR BIOGRAPHY

Dr. Kaushik Saha is Director, Samsung R&D Institute India, Delhi, INDIA since 2013. Prior to this he was working with STMicroelectronics Ltd. since 1996, after having obtained his B.Tech, M.Tech. and Ph.D. degrees from Indian Institute of Technology, Delhi. He had joined STMicroelectronics as a designer of semiconductor memories in the Memory Products Group. Subsequently, he worked for the Applications Lab of the company and was involved in the design of consumer electronics systems around the devices designed and fabricated by the company. His last position in STMicroelectronics was Principal Member Technical Staff in the Advanced Systems Technologies group which is involved in research and development of future generations of devices and systems planned by the company for the markets of tomorrow. His research interests are in the areas of Advanced Processor Architectures and Parallel Algorithms & Architectures for Digital Signal Processing Applications in which he has 19 patents and 14 publications in various international conferences. He is also associated with the Indian Institute of Technology, Delhi in the capacity of Adjunct Faculty.