University of Cincinnati

Date: 4/23/2013

I, Dhanashree Ambekar, hereby submit this original work as part of the requirements for the degree of Master of Science in Electrical Engineering.

It is entitled:
Development of a microcontroller-based head impact detection system for contact sports

Student’s name: Dhanashree Ambekar

This work and its defense approved by:

Committee chair: Fred Beyette, Ph.D.

Committee member: Wen Ben Jone, Ph.D.

Committee member: Carla Purdy, Ph.D.
Development of a Microcontroller-based Head Impact Detection System For Contact Sports

A thesis submitted to the Graduate School of the University of Cincinnati in partial fulfillments of the requirements for the degree of Master of Science in School of Electronics and Computing Systems of the College of Engineering and Applied Science

April 2013

By

Dhanashree Ambekar Bachelor of Engineering (Electronics & Communication Engineering) Birla Institute of Technology, Mesra

Thesis Advisor and Committee Chair: Dr. Fred R. Beyette Jr.
Abstract

A concussion or a mild traumatic brain injury occurs due to blow on the head. During a concussion there is a jarring movement of the brain that causes a person to lose alertness and sometimes consciousness. Each year there are several concussions that go undiagnosed. Contact sports lead to repeated head impacts and blows to the head that can lead to concussions. Some studies prove that repeated concussions can have an irreversible impact on the brain leading to significant issues with attention, concentration, depression and even early death. Early detection of concussion and/or head impacts is critical for successful treatment. A lot of research is being done to develop devices that can detect head impacts. Most of these devices are bulky making them difficult to be fit in helmets let alone making them convenient for sporting activities that do not utilize a helmet (ex. Soccer, swimming, diving, cheerleading, etc.). Some of these devices also require proprietary software to interact with a computer or a tablet which limits their portability across multiple computing platforms. The goal of this research is to develop a prototype that can detect head impacts and provide information about the head impacts suffered by a player. The device should be point-of-care (i.e. easily used during play/practice activities) and useful across any platform.

A compact point-of-care head impact detection system is developed in this system using PIC microcontrollers and accelerometers. The key features of the system are its ability to measure acceleration due to impacts in any direction and any orientation. The system can tell us the exact time and duration of impact. The system transmits data wirelessly to a laptop using ZigBee RF communication modules. This ensures that a large amount of data can be transmitted and stored on a laptop. The system is battery powered with a projected battery life of ~50 hours assuming the system is powered with a 9 volt battery. The system is also quite inexpensive and easy to use.
The head impact detection system uses a PIC microcontroller PIC18F8722 at its core. The sensor module is implemented using Freescale accelerometers and a wireless communication module is implemented using Xbee wireless kits. The system is used to measure the acceleration during the impact and the time of its occurrence. An experimental set up was built to simulate head impacts and test the functionality of the system. Preliminary tests show that the system is able to accurately measure head impact accelerations.
Acknowledgements

I would like to express gratitude towards my advisor Dr. Fred R. Beyette Jr. for giving me this opportunity to work on this exciting project and guiding me throughout this enriching journey. Thank you Dr. Beyette for your immense encouragement, support and guidance throughout my graduate studies which helped me achieve my goals.

I am grateful to my committee members, Dr. Carla Purdy and Dr. Wen-Ben Jone, who took time out to serve on my committee. The computer labs and equipment provided by University of Cincinnati helped me greatly in my work.

I thank Geethanga de Silva and Jeff Simkins for their technical support and guidance which has helped me overcome some crucial hurdles during my project. I would also like to thank Zachariah Al-Deneh, Tritt Dao and Alex Dziech for their assistance towards my work. I greatly appreciate Tritt’s and Alex’s design that helped me mount my device and conduct successful tests. I would also like to thank Vignesh Subbian and all other members of POCSDL for their valuable suggestions. I will always cherish the time that I spent with them both as friends and lab mates.

My time in Cincinnati would not have been complete without my friends who became an integral part of my life here. I would like to thank Omkar, Pradeepa, Sumit, Shalini and Nirjhar for all the fun times we had together.

I would also like to thank my family for their constant support and encouragement. The constant motivation and moral support from my family has helped me greatly in the completion of my program.
# Table of Contents

1. Introduction ................................................................................................................................. 1
   1.1 Concussion in contact sports .................................................................................................. 1
   1.2 Objective ............................................................................................................................... 2
   1.3 Review of previous work ......................................................................................................... 4
   1.4 Thesis outline .......................................................................................................................... 5

2. Design of the Impact Detection System ..................................................................................... 6
   2.1 Working of the system ........................................................................................................... 6
   2.2 Design Considerations .......................................................................................................... 9

3. Implementation Details of the System ....................................................................................... 11
   3.1 PIC18F8722 microcontroller ............................................................................................... 11
      3.1.1 Oscillator configuration ............................................................................................... 14
      3.1.2 Data memory ................................................................................................................ 16
      3.1.3 I/O Ports ...................................................................................................................... 16
      3.1.4 Timers .......................................................................................................................... 17
      3.1.5 Interrupts ...................................................................................................................... 18
      3.1.6 Analog to Digital Converter Module ......................................................................... 19
3.1.7 Enhanced Universal Synchronous Asynchronous Receiver Transmitter ........... 23

3.1.8 Programming the PIC microcontroller ................................................................ 27

3.2 Accelerometer sensors ......................................................................................... 28

3.2.1 Operating characteristics .................................................................................. 30

3.2.2 Pin diagram and connection details .................................................................... 32

3.3 ZigBee module ....................................................................................................... 33

3.3.1 Introduction to ZigBee ...................................................................................... 33

3.3.2 Xbee-ZigBee device features ............................................................................. 37

3.3.3 Xbee module operation ..................................................................................... 39

3.4 Final design and schematic .................................................................................. 44

4. Experimental Verification ...................................................................................... 47

4.1 Preliminary testing ............................................................................................... 47

4.2 Experimental testing and validation ...................................................................... 49

4.2.1 Functional testing ............................................................................................. 49

4.2.2 Validation .......................................................................................................... 55

5. Conclusion and Future Work .................................................................................. 60

5.1 Summary ............................................................................................................... 60

vii
5.2 Future Work .......................................................................................................................... 61
List of Figures

Figure 2.1 Flowchart for operation of CONLISUS device .............................................. 8

Figure 2.2 Block diagram of impact detection system ...................................................... 9

Figure 3.1 Pin diagram of PIC18F8722 ........................................................................... 12

Figure 3.2 Block diagram of clock sources of PIC18F8722 ............................................. 14

Figure 3.3 Oscillator configuration for PIC18F8722 in HS mode ................................... 15

Figure 3.4 T0CON: Timer0 control register ...................................................................... 17

Figure 3.5 ADCON0 register .......................................................................................... 20

Figure 3.6 ADCON1 register .......................................................................................... 21

Figure 3.7 ADCON2 register .......................................................................................... 22

Figure 3.8 TXSTAx register ............................................................................................ 24

Figure 3.9 RCSTAx register ............................................................................................ 25

Figure 3.10 BAUDCONx register ..................................................................................... 26

Figure 3.11 Connection between PIC18F8722 and PICkit3 [22] ...................................... 28

Figure 3.12 Simplified transducer model and equivalent electrical model ......................... 29

Figure 3.13 Dynamic acceleration in the sensing direction for MMA3204 ......................... 30
Figure 3.14 Dynamic acceleration in the sensing direction for MMA1200 _______________ 30

Figure 3.15 Pin out for MMA1200 ____________________________________________ 33

Figure 3.16 Pin out for MMA3204KEG ________________________________________ 33

Figure 3.17 ZigBee Topologies ______________________________________________ 34

Figure 3.18 Zigbee network model ___________________________________________ 36

Figure 3.19 Xbee pin diagram ______________________________________________ 37

Figure 3.20 Suggested connection between microcontroller and Xbee modules __________ 39

Figure 3.21 X-CTU software screenshot __________________________________________ 41

Figure 3.22 Pin diagram of MCP1801 __________________________________________ 43

Figure 3.23 Functional block diagram __________________________________________ 43

Figure 3.24 Typical application circuit for MCP1801 _____________________________ 44

Figure 3.25 Schematic of impact detection system __________________________________ 44

Figure 3.26 PCB layout of impact detection system ________________________________ 45

Figure 3.27 Top view image of the impact detection device 1a: MMA1200 1b: MMA3204 2: PIC18F8722 3: Xbee series 2 _____________________________________________ 45

Figure 4.1 MB Dynamics CAL50 Exciter __________________________________________ 47

Figure 4.2 Acceleration of impact detection system from CAL50 Exciter _____________ 48
Figure 4.3 Experimental setup for validation of impact detection system

Figure 4.4 Data being logged at the laptop

Figure 4.5 Acceleration versus time graph for 5 cm

Figure 4.6 Acceleration versus time graph for 7 cm

Figure 4.7 Acceleration versus time graph for 11 cm

Figure 4.8 Acceleration versus time graph for 9 cm

Figure 4.9 Acceleration versus time graph for 15 cm

Figure 4.10 Acceleration versus time graph for 13 cm

Figure 4.11 Acceleration versus time graph for 19 cm

Figure 4.12 Acceleration versus time graph for 17 cm

Figure 4.13 Mount set up for CONLISUS device

Figure 4.14 Comparison of peak acceleration values from CONLISUS device and the experiment set-up for different stretching distances

Figure 4.15 Acceleration vector plot showing orientation and elevation of the system (starting from left to right: (φ, θ) values are (60°, 135°), (135°, 270°), (15°, 180°), (20°, 45°), (10°, 90°), (45°, 45°), (30°, 45°) respectively; (r_c, r_e) = (45.1, 44.5), (36.8, 34.9), (30.4, 29.8), (13.2, 11.9), (27.3, 26.5), (4.2, 3.7), (19.4, 18.2) respectively)
List of Tables

Table 3-1 Pin description of pins used in PIC18F8722 .......................................................... 13

Table 3-2 Baud rate formulas .................................................................................................. 26

Table 3-3 Operating characteristics of MMA3204................................................................. 31

Table 3-4 Operating characteristics of MMA1200................................................................. 32

Table 3-5 Pinout of Xbee ........................................................................................................ 38

Table 3-6 List of AT Commands used to configure Xbee ...................................................... 41

Table 3-7 Pin description of MCP1801 .................................................................................. 43
1. Introduction

1.1 Concussion in contact sports

A concussion is often defined as an alteration in mental status as a consequence of trauma to the head that may or may not involve loss of consciousness [1]. A bump, blow, or jolt to the head can cause a concussion, a type of mild traumatic brain injury [2]. Concussions can also occur due to excessive force or due to a blow to the body that causes the head and brain to move rapidly back and forth—literally causing the brain to bounce around or twist within the skull. This sudden movement of the brain causes stretching/compression within the brain, damaging the cells and creating chemical changes in the neural tissue. Once these changes occur, the brain is more vulnerable to further injury and sensitive to any increased stress until it fully recovers.

Concussions occur very frequently in football, soccer, boxing and other contact sports. Although less frequent, concussion can also be a problem in non-contact sports such as swimming, diving and cheerleading. It is true that not all head impacts lead to concussion but a head impact is nonetheless harmful to the player's health. The incidence of sports related concussion is estimated to be between 1.6 and 3.8 million concussions annually in the United States. This has led the Centers for Disease Control (C.D.C.) to conclude that sports concussions have reached an "epidemic level" [3]. Concussion is a major concern in American Football where it is estimated that at least one player sustains a mild concussion in each game. A study in the year 2000, focusing on football players and concussions surveyed 1,090 former N.F.L. players and found that more than 60 percent of the players had suffered at least one concussion in their careers and 26 percent had had three or more [4].
Extensive research is being conducted to assess the cumulative effect of head impacts and injuries on youth, college, amateur and professional athletes of contact sports. Every year, a large number of high school students in the United States is diagnosed with concussions [5]. Sports related head injuries are the second leading cause of traumatic brain injury behind motor vehicle accidents among young people between the ages of 15 and 24 [6]. All these studies show that teenagers and young adults are more susceptible to negative outcomes from sports brain injuries and concussions because their brain has not yet fully developed.

Multiple studies have shown that repetitive head injuries can have a deeper impact on a person’s brain. Repetitive brain injuries can result in reduction of brain injury threshold, depression, chronic traumatic encephalopathy and slowed recovery of neurological function [7-9]. Several researchers are now beginning to study the long-term effects of sports brain injury. As has been documented over the last few years, many professional athletes who have donated their brain for study post mortem have been diagnosed with chronic traumatic encephalopathy (CTE) (a non-reversible neurodegenerative disease characterized by memory loss, confusion, dementia, aggression and depression). All of this data highlights the importance of diagnosing sports related head injuries. Specifically, there is a critical need for point-of-care technologies that can measure the head impacts/acceleration experienced by athletes in contact sports and non-contact sports.

1.2 Objective

The goal of this thesis is to develop a device that will help in the detection of concussion or a mild head injury by measuring the impact occurring on the head. During a game of football an athlete can suffer from mild to severe head impacts. There have been several studies on the
symptoms and treatment of concussion [10]. However, there has been significantly less work on understanding the mechanisms of concussion [7, 11-12] and its relation to the impact exposure and concussion tolerance level at different levels of play [13]. Thus, to understand the mechanism of head injuries and what causes them it is important to study the relationship between head impacts and the corresponding force levels experienced by the player. This can be possible if the study is done in vivo as it will give more accurate results. This calls for a smart device that can be easily used and will not prove a hindrance during the study.

A point-of-care approach is needed for the design and implementation of this device. This is to ensure that the device is small enough to be attached as a headband or fixed into a football helmet. It has to be a point-of-care device that can be used by anyone and in any sports. It should be easily attached to the head and can give out instant results on any receiver. It should give a measure of the frequency, duration and the level of impact experienced by an athlete. This in turn will help in providing a detailed analysis of sports brain injuries experienced by an athlete over the course of a game or an entire season. This can prove helpful in the diagnosis of sports brain injury, its severity and the effect it might have on the athlete. The long term objective of the device is to provide an analysis of the player’s brain injuries and to determine their well-being. This work can also provide an insight into the bio-mechanics of repetitive sports brain injury and its long term effects.

Since a concussion is caused due to a heavy blow on the head, it is important to know the amount of force experienced during this jarring contact. Measuring of acceleration can provide a quantitative measure of this force. Further, by accurately measuring acceleration during a head impact, it should be possible to determine whether the impact has contributed towards the occurrence of a concussion. In addition to measuring the acceleration events, it is also important
that the sensor technology provide data logging to insure that sensor data is available for further analysis and diagnostic purposes.

The device presented here is called CONLISUS (meaning “collision” or “clash” in Latin). The COLISUS device accommodates the required features, with circuitry that includes two accelerometer sensor chips from Freescale, a PIC microcontroller and a ZigBee wireless communication module. The Freescale accelerometers sense acceleration in 3 orthogonal directions X, Y and Z. The PIC microcontroller is the brain of the device where all of the data is recorded, stored and transmitted using the ZigBee module. The ZigBee module, implemented using Xbee kits from Digi International, Inc. includes both a transmitter and a receiver. The transmitter is connected to the microcontroller and receiver is connected to a laptop. The device is called CONLISUS (meaning “collision” or “clash” in Latin).

1.3 Review of previous work

Several head impact monitoring devices have been developed to detect sports related brain injuries. A lot of research is being done to develop prototypes that can measure acceleration and help quantify the external forces leading to brain injury. Some of these devices are helmet-based devices [14-15], chinstrap-based devices [16], devices fitted in mouth guards [17] and many others [18].

Most of these devices are prototypes that measure head acceleration. Some of them are fitted in helmets which limit their use to sports that require the use of helmets; others are fitted in chinstraps which again require a strap on the player’s chin. Frequently, these devices are bulky and cumbersome to use. Moreover, the addition of accelerometers and other systems to helmets increases their cost making them less attractive for high school, middle school and little league
players. Some of these systems also require specific software and equipment in order to gather impact data. All of these drawbacks provide incentive for the development of a simple, low cost, comprehensive device that can be used easily by anyone for impact detection in sports.

1.4 Thesis outline

Chapter 1 gives a brief introduction and background of the thesis topic along with the objective of this thesis. A review of previous work is also given as a measure of contemporary technology and development.

Chapter 2 deals with the design of the system and all of its features.

Chapter 3 deals with the development of the system and its associated components including, explanations of the accelerometer sensors, ZigBee network and methods for interfacing all peripherals to the microcontroller.

Chapter 4 discusses the experimental set up for the validation of the system and experimental results from CONLISUS device testing.

Chapter 5 concludes the thesis and discusses future work.
2. Design of the Impact Detection System

This chapter deals with the overview of the entire system and provides details on the overall working of the device. The head impact detection system, also named as CONLISUS, is aimed at providing an accurate measure of head impacts endured by an athlete during practice and/or game conditions. This impact data may be useful in the early detection of a concussion or a mild brain injury [19]. The goal of the system is to provide a detailed log of the impact accelerations experienced by a player over a period of time. The system has following features:

1. The CONLISUS device is capable of measuring acceleration in all three orthogonal directions when an impact occurs.
2. It should also give the time of the impact and its duration.
3. The system should be small enough to be attached to the forehead or fixed in a helmet.
4. The system should be battery powered so that it can last for the duration of a single game.
5. The system should be able to store and transmit data to facilitate further study and archival storage of head impact data.

2.1 Working of the system

The CONLISUS device is designed to measure and record acceleration experienced by a player. The accelerometers measure linear acceleration producing an analog voltage that is proportional to the acceleration. The output voltage from the accelerometers is then sampled by the PIC microcontroller. The PIC microcontroller reads one axis data at a time, converts it to a digital value and compares it to a threshold. The threshold for the device is set to 5g, where g is acceleration due to gravity. While the exact relationship between acceleration threshold and mild traumatic brain injury is unknown, there has been some effort to determine a threshold for head
injury [20]. In this previous work, a translational acceleration of 85g was suggested as the threshold for onset of mild traumatic brain injury. For this thesis work, a measurement threshold of 5g was chosen to record even the slightest blow to the head. A low threshold value will result in every single blow being stored and recorded so that the player’s profile over an entire game can be quantified. This approach will also help in giving a detailed account of impact data; this ensures that even a small push to the head is recorded along with the major blows. Since all of the acceleration data is transmitted to a laptop for analysis and archival storage, a low threshold can be set for detailed analysis.

After the threshold is reached, the data is stored in the data memory buffer of the microcontroller. This is to prevent data loss when the device is out of range from the ZigBee receiver. Along with the acceleration values, the time of the occurrence of the impact is also stored. Each axis has its own dedicated buffer in the data memory. After the data is stored, it is transmitted to the laptop via the ZigBee module. This is done via data logging terminal program where all of the data is stored in an Excel sheet or a similar application. This data can be used for further analysis or just to find out peak accelerations. If the impact is high enough, the player can be removed from practice/play to facilitate evaluations using standard concussion tests and/or provide appropriate medical care. Figure 2.1 shows a flow chart for operation of the CONLISUS device.

As shown, the device first acquires the accelerometer data from each of the orthogonal axis. Then the magnitude of the acceleration is calculated and compared to the 5g threshold. If the acceleration is less than the 5g threshold, the device checks to determine if data is waiting in the data buffer and if the ZigBee receiver is in range. If either of these conditions is No, then the measure process is repeated. If both of these conditions are yes then the device transmits the
buffered data via the ZigBee module. If the result of the acceleration calculation is larger than the 5 g threshold, the accelerometer values and timestamp are stored in the data buffer before the measurement process is repeated. It should be noted that during normal sampling operation the device is configured for sampling rate of 20,000 samples per second (i.e. every 50 µs). Since most sporting collisions have a duration measured in tens of milliseconds, the CONLISUS device can accurately capture several hundred data points during each collision.

![Flowchart](image)

**Figure 2.1 Flowchart for operation of CONLISUS device**
2.2 Design Considerations

Figure 2.2 shows the block diagram of the system incorporating the features presented above. The core of the system is the PIC18F8722 microcontroller from Microchip. Two Freescale accelerometer sensors are used to measure acceleration in all three directions; a ZigBee module with a transmitter and receiver for communication purposes. System power is provided through a battery which is regulated using the low dropout voltage regulators from Microchip.

The choice of battery technology depends on the specific application. For sports that do not require a helmet, the CONLISUS device can be powered from lightweight button cells. For sports that allow for insertion into a helmet, a 9V battery may be used to extend the operating life. Regardless of application, the power consumed by each individual component is a major design consideration. For the work presented in this thesis, the system is powered using a 9V battery. Two MCP1801 low dropout voltage regulators are used to convert the 9V battery potential to 5V and 3.3V. These regulators available from Microchip were chosen because of their low dropout current feature and their extremely small size. Moreover, they work for an input power supply range between 5V-9V allowing for easy conversion between button cell and conventional 9V battery.
PIC18F8722 is an 8-bit microcontroller from Microchip. It is a low power microcontroller with run mode current as low as 25 μA. This makes it ideal for battery powered use. It also has 10-bit 16 channel ADC module; multiple channels can be used to connect multiple sensors. It has 4 timers which can be used simultaneously, priority driven interrupts and 2 EUSART (Enhanced Universal Synchronous Asynchronous Receiver Transmitter). It can be programmed using the PICkit 3 programmer in C with its dedicated C18 compiler.

The Freescale sensors used in the system (MMA3204KEG and MMA1200KEG) are both surface micromachined integrated-circuit accelerometers. MMA3204KEG is a dual axis accelerometer and MMA1200KEG is a single axis accelerometer. These sensors are easy to prototype because of their surface mount packaging. The sensing range of the sensors makes them ideal for use in impact monitoring. They provide an analog output that is proportional to the acceleration and their 5V dynamic range makes them ideal for integration with PIC microcontrollers.

The ZigBee wireless communication technology chosen for this project runs over the unlicensed 2.4GHz spectrum and consumes very little power compared to other wireless protocols such as Bluetooth. It has an efficient idle mode that enables significant power savings by allowing the communication model to move easily between sleep and transmit operation. It also has the advantage of operating under star, mesh and point-to-point network topologies. The ZigBee module is implemented using Xbee kits from Digi International. These kits consist of a transmitter connected to the PIC microcontroller via UART (Universal Asynchronous Receiver Transmitter) and a receiver connected to a laptop via USB. Xbee kits are chosen for their small size, ease of availability and implementation.
3. Implementation Details of the System

3.1 PIC18F8722 microcontroller

The microcontroller is central to the functioning of the CONLISUS device. As a control device it is well suited for integrating multiple sensors, communication devices and other support hardware. All PIC microcontrollers are programmable using the MPLAB IDE development environment that supports program downloaded into the microcontrollers program memory using the PICkit3 programmer. The development environment supports programing in either assembly language of C using the C18 compiler. For this project, microcontroller programing was developed in C.

The PIC18F8722 is an 80-pin 1Mbit enhanced flash microcontroller with 10-bit A/D [21]. There are 70 Input/output pins divided into 9 ports namely A, B, C, D, E, F, G, H and J. Each of these ports is bi-directional and 8 bits wide except for port G which is 6 bits wide. Many of the chip pins are mapped to specific functionality necessary to support peripheral operations (ex. comparators, serial interface protocols including SPI and I²C and parallel communication via UART). If a peripheral operation is being utilized by the PIC then the peripheral operation is enabled and the pin may not be used as a general purpose I/O.

The PIC18F8722 microcontroller has several features which contributed to its selection as a part of the design:

- Low power consumption, run mode currents down to 25 μA typical
- A 16 –channel, 10-bit Analog to Digital Converter module that supports conversion during sleep mode
• 4 Timer modules, Two 8-bit timers and three 16-bit timers (Timer1 can be operated as both 8-bit or 16-bit)

• Enhanced addressable USART module that supports RS-232

• 4Kb of Data Memory and 128Kb of programmable flash memory

• Single-Supply 5V In-Circuit Serial Programming (ICSP) via two pins

• Priority levels for interrupts

• C compiler optimized architecture

![Figure 3.1 Pin diagram of PIC18F8722](image-url)
Figure 3.1 shows the pin diagram of the PIC18F8722 microcontroller. A brief description of the pins used in the device is also given. The A/D module, Timers, Enhanced USART, data memory, ICSP and oscillator modes are described in detail later.

Table 3.1 describes the pins used in this device.

**Table 3-1 Pin description of pins used in PIC18F8722**

<table>
<thead>
<tr>
<th>Pin name</th>
<th>Function</th>
<th>I/O</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSC1/CLKI/RA7</td>
<td>OSC1</td>
<td>I</td>
<td>Oscillator crystal or external clock input.</td>
</tr>
<tr>
<td>OSC2/CLKO/RA6</td>
<td>OSC2</td>
<td>I</td>
<td>Oscillator crystal or clock output.</td>
</tr>
<tr>
<td>RB6/KBI2/PGC</td>
<td>PGC</td>
<td>I/O</td>
<td>In-Circuit Debugger and ICSP programming clock pin</td>
</tr>
<tr>
<td>RB7/KBI3/PGD</td>
<td>PGD</td>
<td>I/O</td>
<td>In-Circuit Debugger and ICSP programming data pin</td>
</tr>
<tr>
<td>RC6/TX1/CK1</td>
<td>TX1</td>
<td>-</td>
<td>EUSART1 asynchronous transmit.</td>
</tr>
<tr>
<td>RC7/RX1/DT1</td>
<td>RX1</td>
<td>-</td>
<td>EUSART1 asynchronous receive</td>
</tr>
<tr>
<td>RD7/PSP7/SS2</td>
<td>RD7</td>
<td>O</td>
<td>Digital O</td>
</tr>
<tr>
<td>RF0/AN5</td>
<td>AN5</td>
<td>I</td>
<td>Analog input 5</td>
</tr>
<tr>
<td>RF1/AN6/C2OUT</td>
<td>AN6</td>
<td>I</td>
<td>Analog input 6</td>
</tr>
<tr>
<td>RF2/AN7/C1OUT</td>
<td>AN7</td>
<td>I</td>
<td>Analog input 7</td>
</tr>
<tr>
<td>RG5/MCLR/VPP</td>
<td>MCLR/VPP</td>
<td>I/P</td>
<td>MCLR- Master Clear (Reset) input. This pin is an active-low Reset to the device. VPP- Programming voltage input.</td>
</tr>
<tr>
<td>VSS</td>
<td>Gnd</td>
<td>P</td>
<td>Ground reference for logic and I/O pins</td>
</tr>
<tr>
<td>VDD</td>
<td>Power</td>
<td>P</td>
<td>Positive supply for logic and I/O pins</td>
</tr>
<tr>
<td>AVSS</td>
<td>Power</td>
<td>P</td>
<td>Ground reference for analog modules.</td>
</tr>
<tr>
<td>AVDD</td>
<td>Power</td>
<td>P</td>
<td>Positive supply for analog modules.</td>
</tr>
</tbody>
</table>
3.1.1 Oscillator configuration

PIC18F8722 has three clock sources: the Primary Oscillator, the Secondary Oscillator and the Internal Oscillator. The Primary Oscillator includes the external crystal/resonator mode, the external RC mode, the external clock mode and the internal oscillator block. The secondary oscillators are those oscillators that are not connected to the OSC pins. Finally, the internal oscillator is a primary oscillator that is integrated into the chip circuitry. In addition eliminating the need for external clock circuitry, the internal oscillator is a power managed clock source which can be utilized to optimize the trade-off between power consumption and system performance.

Figure 3.2 shows a block diagram of the clock management circuitry including the different clock sources of PIC18F8722

![Block diagram of clock sources of PIC18F8722](image-url)

**Figure 3.2 Block diagram of clock sources of PIC18F8722**
The PIC18F8722 can be configured in ten different oscillation modes:

- LP Low-Power Crystal
- XT Crystal/Resonator
- HS High-Speed Crystal/Resonator
- HSPLL High-Speed Crystal/Resonator with PLL enabled
- RC External Resistor/Capacitor with Fosc/4 output on RA6
- RCIO External Resistor/Capacitor with I/O on RA6
- INTIO1 Internal Oscillator with Fosc/4 output on RA6 and I/O on RA7
- INTIO2 Internal Oscillator with I/O on RA6 and RA7
- EC External Clock with Fosc/4 output
- ECIO External Clock with I/O on RA6

The XT, LP, HS or HSPLL oscillator modes require a ceramic or a crystal resonator/oscillator. In the CONLISUS device, HS i.e. High Speed oscillation mode is used. A crystal oscillator/resonator is connected to the OSC1 and OSC2 pins to establish oscillation. The oscillator design requires the use of a parallel crystal cut. The device uses 10.0 MHz 20pF crystal. Figure 3.3 shows the pin connections.

Capacitors C1 and C2 are 22pF ceramic capacitors, Rs is an optional series resistance that may

![Oscillator configuration for PIC18F8722 in HS mode](image)

Figure 3.3 Oscillator configuration for PIC18F8722 in HS mode
be required for some AT strip cut crystals and $R_F$ is an internal resistance that varied depending on the oscillator mode selected.

### 3.1.2 Data memory

The PIC18F8722 includes a data memory that is implemented using static RAM. The size of the data memory is 4096 bytes with each register having a 12-bit address. The memory is divided into 16 banks of 256 bytes each. The data memory consists of General Purpose Registers (GPRs) and Special Function Registers (SFRs). The SFRs are used for control and status of the controller and peripheral functions; these include the control registers for all the different modules. GPRs are used for data storage and scratchpad operations in the user’s application. Any read of an unimplemented location will read as ‘0’s. As the SFRs occupy a part of the static RAM, the actual data memory bytes available for user are 3936 bytes.

For the CONLISUS device, two data banks are combined to form one memory buffer. This creates 512 bytes of memory buffer to store values of acceleration of each axis. The three memory buffers are consecutively arranged so that each axis gets its own devoted buffer to ensure data clarity. A fourth memory buffer of 256 bytes is created to store the time stamp values.

Banks 4-9 are used to store acceleration data and bank 10 is used to store the timer values.

### 3.1.3 I/O Ports

PIC18F8722 has nine I/O ports where each bidirectional pin can be designated as input or output, or the pins can perform other peripheral functions. If a pin is used for a peripheral function, it cannot be used for general I/O purpose. Each port has three registers for its operation:
- **TRIS Register** (Data Direction Register) – This register is used to configure the direction of the port-input or output. When set as ‘0’ the port is configured as output.

- **LAT Register** (Data Latch) – This register is used to write/read/modify the values of the I/O pins of the port. Any value written to the port is written through this register.

- **Port register** – This register is used to read the levels on the pins of the device.

### 3.1.4 Timers

Timer0 can be configured as a timer or counter in 8-bit or 16-bit modes. It has two readable and writeable 8-bit registers, TMR0L and TMR0H. Timer0 has a dedicated 8-bit software programmable prescaler. The prescaler values can be configured from 1:2 through 1:256 in

![Figure 3.4 T0CON: Timer0 control register](image)
power of 2 increments. Timer0 can have two clock sources, either the internal instruction clock or a clock connected to the T0CKI pin. Timer0 can be configured using the T0CON Timer0 Control Register as shown in the Figure 3.4.

In the CONLISUS device, the timer is used as an 8-bit timer with the prescaler disabled so that the Timer0 increments with each instruction clock.

### 3.1.5 Interrupts

The PIC18F8722 has multiple interrupt sources and an interrupt priority feature. Interrupts can be high priority or low priority. The high priority interrupt vector is located at program memory address 0008H and the low priority interrupt vector is at program memory address 0018H. There are ten registers that are used to control interrupt operation. These are:

- RCON
- INTCON, INTCON2, INTCON3
- PIR1, PIR2, PIR3
- PIE1, PIE2, PIE3
- IPR1, IPR2, IPR3

Three bits are used to control the interrupt source operations. These are:

- Flag bit to determine if an interrupt has occurred or not
- Enable bit which sends the program to the interrupt service routine location whenever the bit is set
- Priority bit which determines the priority level of the interrupt, either high or low
The interrupt priority can be implemented by using the IPEN bit in the RCON register. In order to set high and low interrupt priority, GIEH and GIEL bits in the INTCON register need to be set. When an interrupt occurs, the program branches to the interrupt vector location of the respective priority interrupt. When several interrupts of similar priority are enabled, the interrupt flags are polled to check for the occurrence of an interrupt. When an interrupt occurs, the global interrupt is cleared to prevent the occurrence of any further interrupts while the current one is being serviced. The return address is pushed onto the stack and the program counter (PC) is loaded with the interrupt vector address. After the Interrupt Service Routine is executed, all the interrupt flags are cleared, the global interrupt is enabled and the program control is transferred to the main program.

A Timer0 interrupt occurs when the timer overflows from FFh to 00h. The Timer0 interrupt flag is set when an interrupt occurs. This flag needs to be cleared in software during the Interrupt Service Routine.

The CONLISUS device uses the Timer0 set for a low priority interrupt. In the interrupt service routine, a value of 84H is added to the TMR0L to adjust the overflow interval to exactly 50 µs. This interval is used to initiate ADC readings from the accelerometers. A separate variable is used to count up to 100 msec and then update the real time clock.

3.1.6 Analog to Digital Converter Module

The A/D converter module is a 10-bit converter with 16 input channels. The module converts an analog voltage to a 10-bit digital reading from 0 to 1023. The following registers are used for the A/D module:
- A/D Result High Register (ADRESH) – This register stored the high byte of the conversion result
- A/D Result Low Register (ADRESL) – This register stores the low byte of the conversion result
- A/D Control Register 0 (ADCON0) – This register controls the operation of the A/D module
- A/D Control Register 1 (ADCON1) – This configures the functions of the port pins and sets the reference voltage
- A/D Control Register 2 (ADCON2) – This register configures the A/D clock, justification and programmed acquisition time.

Figure 3.5 shows the ADCON0 register. This register is used to select the channels to be used for analog to digital conversion. This project uses channel 5(AN5), 6(AN6) and 7(AN7) for the three

<table>
<thead>
<tr>
<th>U-0</th>
<th>U-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
<th>RW-0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend:
- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as '0'
- **.n** = Value at POR
- ‘0’ = Bit is cleared
- ‘X’ = Bit is unknown

Figure 3.5 ADCON0 register
different axes. ADCON0 register is also used to turn ON and OFF the ADC module and to initiate the conversion.

Figure 3.6 shows the ADCON1 register. This register is used to set the reference voltage for the ADC module. It can be done using the AVDD and AVSS. It is also used to configure the port pins as analog or digital.

![Figure 3.6 ADCON1 register](image-url)
Figure 3.7 ACON2 register

Figure 3.7 shows the ACON2 register. It is used to set the justification of the ADC module. When right justification is used, the lower 8-bits (including LSB) are stored in ADRESL and upper 2 bits are stored in ADRESH. When left justification is used, upper 8 bits (including MSB) are stored in ADRESH and lower 2 bits (including LSB) are stored in ADRESL. ACON2 is also used to set the A/D acquisition time and A/D conversion clock frequency.

The A/D module is configured as follows in this device:

- A/D module used Channels 5, 6 and 7 for X, Y and Z axes of the accelerometer sensors respectively. The output of these pins is connected to RF0, RF1 and RF2 respectively.
- A/D module uses 0 to 5V as reference voltages. ACON1 is set so that pins 0(AN0) to pin 7 (AN7) are analog. The result is left justified so ADRESH stores upper 8-bits and ADRESL stores lower 2 bits.
• For accurate operation, the conversion clock should be as small as possible but greater than the minimum period TAD specified. The conversion clock is set to \( \frac{F_{osc}}{16} \) i.e. it is 0.625 MHz. The acquisition time is set to be 12 TAD.

3.1.7 Enhanced Universal Synchronous Asynchronous Receiver Transmitter

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) is one of two serial modules of PIC18F8722. It can be used for full-duplex asynchronous communication to connect to peripherals devices such as computers, CRT terminals, etc. The EUSART can be configured in the following modes:

• Asynchronous (full duplex) with:
  - Auto-Wake-up on Character Reception
  - Auto-Baud Calibration
  - 12-bit Break Character Transmission

• Synchronous – Master (half duplex) with selectable Clock Polarity

• Synchronous – Slave (half duplex) with selectable Clock Polarity

There are two EUSART modules EUSART1 implemented using PORT C and EUSART2 using PORT G pins. RC6/TRISC 6 acts as TX and RC7/TRISC 7 acts as RX pin for PORT C and RG1/TRISG 1 acts as TX and RG2/TRISG 2 acts as RX pin for PORT G. Each EUSART module is controlled through the following registers:

• TXSTA\textsubscript{x} – This is the transmit status and control register. It is used to select the USART mode either asynchronous or synchronous, enable transmission and select high or low baud rate.

23
- RCSTAx – This is the receive status and control register. It is used to enable the serial port and the receiver.
- BAUDCONx – This is the baud rate control register. It is used to set the baud rate, auto-detect it and select the baud rate generator.

In this device, EUSART1 is used to connect to the ZigBee module. The following pins must be used to set up EUSART1:

- bit SPEN (RCSTA1<7>) is set to 1
- bit TRISC<7> is set to 1
- bit TRISC<6> is cleared for Asynchronous and Synchronous Master modes
- bit TRISC<6> is set for Synchronous Slave mode

**Figure 3.8 TXSTAx register**

<table>
<thead>
<tr>
<th>RW/G</th>
<th>RW/M</th>
<th>RW/6</th>
<th>RW/4</th>
<th>RW/6</th>
<th>RW/4</th>
<th>RW/6</th>
<th>RW/6</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSRC</td>
<td>TXG</td>
<td>TXEN</td>
<td>SYNC</td>
<td>SEGBB</td>
<td>BRGH</td>
<td>TRMT</td>
<td>TXGO</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 7</th>
<th>bit 6</th>
<th>bit 5</th>
<th>bit 4</th>
<th>bit 3</th>
<th>bit 2</th>
<th>bit 1</th>
<th>bit 0</th>
</tr>
</thead>
</table>
| CSRC  | Clock Source Select bit
|       | Asynchronous mode:
|       | Don't care.
|       | Synchronous mode:
|       | 1 = Master mode (clock generated internally from BR32)
|       | 0 = Slave mode (clock from external source)
| TXG   | Select Transmit Enable bit
|       | 1 = Selects 8-bit transmission
|       | 0 = Selects 1-bit transmission
| TXEN  | Transmit Enable bit
|       | 1 = Transmit enabled
|       | 0 = Transmit disabled
| TRMT  | Transmit Shift Register Status bit
|       | 1 = TSRx empty
|       | 0 = TSRx full
| TXGO  | Ninth bit of Transmit Data
|       | Can be address data bit or a parity bit.

Legend:
- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as 0'
- **1** = value at POR
- **'Y'** = bit is set
- **'O'** = bit is cleared
- **x** = bit is unknown
<table>
<thead>
<tr>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R-0</th>
<th>R-0</th>
<th>RXD</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPEN</td>
<td>RX9</td>
<td>SREN</td>
<td>CREN</td>
<td>ADDEN</td>
<td>FERR</td>
<td>OERR</td>
<td>RXD</td>
</tr>
</tbody>
</table>

**Legend:**

- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as '0'
- **n** = Value at POR
- **'1'** = Bit is set
- **'0'** = Bit is cleared
- **x** = Bit is unknown

**bit 7**

- **SPEN**: Serial Port Enable bit
  
  - 1 = Serial port enabled (configures RXxD/Tx and TXxD/Rx pins as serial port pins)
  
  - 0 = Serial port disabled (held in Reset)

**bit 8**

- **RX9**: 9-bit Receive Enable bit
  
  - 1 = Selects 9-bit reception
  
  - 0 = Selects 8-bit reception

**bit 5**

- **SREN**: Single Receive Enable bit
  
  **Asynchronous mode:**
  
  - Don't care
  
  **Synchronous mode** – **Master**:
  
  - 1 = Enables single receive
  
  - 0 = Disables single receive
  
  This bit is cleared after reception is complete.

  **Synchronous mode** – **Slave**:
  
  - Don't care

**bit 4**

- **CRN**: Continuous Receive Enable bit
  
  **Asynchronous mode**:
  
  - 1 = Enables receiver
  
  - 0 = Disables receiver

  **Synchronous mode**:
  
  - 1 = Enables continuous receive until enable bit CRN is cleared (CRN overrides SREN)
  
  - 0 = Disables continuous receive

**bit 3**

- **ADDEN**: Address Detect Enable bit
  
  **Asynchronous mode** (bit RXS = 1):
  
  - 1 = Enables address detection, enables interrupt and loads the receive buffer when RSRx<>Rx is set
  
  - 0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit.

  **Asynchronous mode** (bit RXS = 0):
  
  - Don't care.

**bit 2**

- **FERR**: Framing Error bit
  
  - 1 = Framing error (can be updated by reading RCREGx register and receiving next valid byte)
  
  - 0 = No framing error

**bit 1**

- **OERR**: Overrun Error bit
  
  - 1 = Overrun error (can be cleared by clearing bit CRN)
  
  - 0 = No overrun error

**bit 0**

- **RXD**: 9th bit of Received Data
  
  This can be added as a data bit or a parity bit and must be calculated by user firmware.

---

**Figure 3.9 RCSTAx register**

Figure 3.8, Figure 3.9, and Figure 3.10 show the TXSTAx, RCSTAx, and BAUDCONx respectively.

The BRG is a dedicated 8-bit or 16-bit baud rate generator that is used to drive the asynchronous or synchronous modes of EUSART. The SPBRGx:SPR register pair is used to set the baud rate by controlling the period of a free running timer. The BRG runs in 8-bit mode by default; setting the BRG16 bit changes it to 16-bit generator. The value of the SPBRGHx:SPR register...
Figure 3.10 BAUDCONx register

The BAUDCONx register pair can be calculated to the nearest integer if the desired baud rate and frequency of oscillation (Fosc) values are known. The formulae are listed in Table 3.2.

Table 3-2 Baud rate formulas

<table>
<thead>
<tr>
<th>Configuration Bits</th>
<th>BRG/EUSART Mode</th>
<th>Baud Rate Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BRG16</td>
<td>BRGH</td>
</tr>
<tr>
<td>00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>00</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>x</td>
</tr>
</tbody>
</table>

Legend: x = Don't care, n = value of SPBRGHx:SPBRGx register pair
The EUSART 1 module is configured as follows for the CONLISUS device:

- The EUSART 1 module is set up in the asynchronous mode by clearing the SYNC bit. A standard Non-Return-to Zero (NRZ) format is used and data is 8 bits long.
- A continuous transmission and reception mode is selected where data is constantly loaded and then transmitted from the TXREG register.
- The baud rate generator is set to 8-bit and high speed baud rate is selected. The SPBRGHx:SPBRG register pair has a value of 10 resulting in a baud rate of 57600 bps.

3.1.8 Programming the PIC microcontroller

The PIC18f8722 can be programmed in either C or assembly language using the MPLAB IDE development environment provided by Microchip Inc. The following are the requirements to program the PIC microcontroller:

- Source Code – The source code can be in C or in assembly language. For this project, the source code was written in C. It contains the necessary instructions for the microcontroller to perform its functions, interface with peripherals and implement various operations. MPLAB IDE tool is used to write the program source code. MPLAB IDE also contains a debugger tool kit to interact with the Microchip hardware tools.
- Assembler/Compiler – The source code is compiled using C18 compiler to generate a HEX file which contains the opcodes for the microcontroller. MPLAB IDE contains the MPLAB C18 compiler which is a C compiler for PIC microcontrollers.
- Programmer – The programmer, as its name suggests, is used to program the microcontroller. MPLAB In-Circuit Debugger 2 (ICD2) or PICKit 2 or PICKit 3 is used as a programmer that uses the HEX file generated to program into the microcontroller. The
Figure 3.11 Connection between PIC18F8722 and PICkit3 [22]

PICkit3 programmer is used in this system to program the microcontroller. A 6-pin connector is used between the PICkit 3 programmer and PIC18F8722. Figure 3.11 shows the connections between PICkit3 and the target microcontroller [22].

3.2 Accelerometer sensors

The sensors used in the CONLISUS device are Freescale MEMS based high-g accelerometers. There are two different sensors that are used, one is MMA3204 dual axis X-Y accelerometer and the other is the MMA1200 single axis Z accelerometer. This ensures that acceleration in each orthogonal direction is measured. The following features of the Freescale sensors make them ideal for use in the point-of-care device:

- The range for MMA3204 is 100g in X – axis and 30g in Y – axis. The range for MMA1200 is 250g in Z – axis.
- Internal signal conditioning and ratiometricity ensure that it can be easily interfaced to the A/D input of a microcontroller.
- The sensor has a calibrated self-test to allow the verification of the mechanical and electrical integrity of the accelerometer at any time before or after installation.
- The operating temperature for sensor is wide enough from -40 C to 125 C.
- The output voltage of the sensor varies from 0 V to 5 V where 2.5 V denotes 0g acceleration or no output acceleration.
- The sensor design is robust and has a high shock survivability enabling it to be used for impact detection.

These Freescale accelerometers are surface-micromachined integrated-circuit accelerometers [23-24]. Each accelerometer consists of a surface-micromachined capacitive sensing g-cell and a CMOS signal conditioning ASIC contained in a single integrated circuit package. The sensing element is sealed hermetically at the wafer level using a bulk micromachined “cap” wafer.

The g-cell is a polysilicon based mechanical structure that can be modeled as a moveable plate placed between two stationary plates. Acceleration to the system causes the center plate to deviate from its zero or central position. When the center plate deviates, the distance from it to one of the plates increases by the same amount that the distance to the other plate decreases which gives us a measure of acceleration. This arrangement forms two back-to-back capacitors. As the center plate moves due to acceleration, the distance between the plates changes and each capacitor’s value changes, \( C = N A \varepsilon / d \); where \( A \) is the area of the plate, \( \varepsilon \) is the dielectric constant, \( N \) is the number of beams and \( d \) is the distance between the plates. Figure 3.12 shows an equivalent model of the capacitive arrangement.

![Figure 3.12 Simplified transducer model and equivalent electrical model](image-url)
The CMOS ASIC uses switched capacitor techniques to measure the g-cell capacitors and extract the acceleration data from the difference between the two capacitors. Signal conditioning and filtering is done using the ASIC, providing a high level output voltage that is ratiometric and proportional to acceleration.

Both sensors, MMA3204 and MMA1200 have the same working principle. However, the X-Y device (MMA3204) contains two structures of plates that are orthogonal to each other. Thus, two sets of outputs are displayed by it, namely, $X_{\text{out}}$ and $Y_{\text{out}}$. The Z axis sensor has only one structure that gives the $Z_{\text{out}}$ output.

The sensors give a ratiometric output. This means that the output offset voltage and sensitivity will scale linearly with applied voltage; that is, as the supply voltage increases the sensitivity and offset increase linearly; as supply voltage decreases, offset and sensitivity decrease linearly.

Figure 3.13 and figure 3.14 shows the acceleration direction for each sensor.

### 3.2.1 Operating characteristics

The operating characteristics of any sensor specify parameters such as response time, warm up time, supply voltage, offset accuracy and sensitivity.
The operating characteristics of MMA3204 are listed in Table 3.3 and the operating characteristics of MMA1200 are listed in Table 3.4.

The transfer function of the sensor is given by:

\[ V_{\text{out}} = \text{accl} \times S + V_0 \]

Where \( V_{\text{out}} \) – output voltage in Volts
Accl- acceleration value in multiple of g (where g is acceleration due to gravity)
S- sensitivity of the sensor in V/g
\( V_0 \) – output voltage when the acceleration is zero in Volts

Table 3-3 Operating characteristics of MMA3204

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Range (^{[2]})</td>
<td>( V_{\text{DD}} )</td>
<td>4.75</td>
<td>5.00</td>
<td>5.25</td>
<td>V</td>
</tr>
<tr>
<td>Supply Voltage (^{[3]})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>mA</td>
</tr>
<tr>
<td>Supply Current (^{[4]})</td>
<td>( I_{\text{DD}} )</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Operating Temperature Range</td>
<td>( T_A )</td>
<td>-40</td>
<td></td>
<td>+125</td>
<td>°C</td>
</tr>
<tr>
<td>Acceleration Range X-axis</td>
<td>( F_{\text{RX}} )</td>
<td>112.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Acceleration Range Y-axis</td>
<td>( F_{\text{RY}} )</td>
<td>33.75</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output Signal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zero g X-axis ((T_A = 25^\circ \text{C}, V_{\text{DD}} = 5.0 \text{ V}))</td>
<td>( V_{\text{OFF}} )</td>
<td>2.35</td>
<td>2.5</td>
<td>2.6</td>
<td>V</td>
</tr>
<tr>
<td>Zero g Y-axis ((T_A = 25^\circ \text{C}, V_{\text{DD}} = 5.0 \text{ V}))</td>
<td>( V_{\text{OFF}} )</td>
<td>2.25</td>
<td>2.5</td>
<td>2.75</td>
<td>V</td>
</tr>
<tr>
<td>Zero g Z-axis ((T_A = 25^\circ \text{C}, V_{\text{DD}} = 5.0 \text{ V}))</td>
<td>( V_{\text{OFF}} )</td>
<td>0.45</td>
<td>0.60</td>
<td>0.64</td>
<td>V</td>
</tr>
<tr>
<td>Sensitivity X-axis ((T_A = 25^\circ \text{C}, V_{\text{DD}} = 5.0 \text{ V}))</td>
<td>( S_X )</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>mV/g</td>
</tr>
<tr>
<td>Sensitivity Y-axis ((T_A = 25^\circ \text{C}, V_{\text{DD}} = 5.0 \text{ V}))</td>
<td>( S_Y )</td>
<td>68.83</td>
<td>68.67</td>
<td>70.0</td>
<td>mV/g</td>
</tr>
<tr>
<td>Sensitivity Z-axis</td>
<td>( S_Z )</td>
<td>3.72</td>
<td>4</td>
<td>4.26</td>
<td>mV/g</td>
</tr>
<tr>
<td>Bandwidth Response</td>
<td>( f_{\text{MB}} )</td>
<td>300</td>
<td>400</td>
<td>440</td>
<td>Hz</td>
</tr>
<tr>
<td>Nonlinearity</td>
<td>( N_{\text{L}}\text{OUT} )</td>
<td>-1.0</td>
<td></td>
<td>+1.0</td>
<td>%FSO</td>
</tr>
<tr>
<td>Noise</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMS (50 Hz – 1 kHz)</td>
<td>( n_{\text{RMS}} )</td>
<td></td>
<td></td>
<td>2.5</td>
<td>mVrms</td>
</tr>
<tr>
<td>Power Spectral Density</td>
<td>( n_{\text{PSD}} )</td>
<td></td>
<td></td>
<td>110</td>
<td>mV/√Hz</td>
</tr>
<tr>
<td>Clock Noise (without RC load on output) (^{[6]})</td>
<td>( n_{\text{CLK}} )</td>
<td></td>
<td>2.0</td>
<td></td>
<td>mVpk</td>
</tr>
<tr>
<td>Self-Test</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output Response</td>
<td>( q_{\text{GT}} )</td>
<td>9.6</td>
<td>12</td>
<td>14.4</td>
<td>( \mu \text{A} )</td>
</tr>
<tr>
<td>Input Low</td>
<td>( q_{\text{IL}} )</td>
<td>( V_{\text{IL}} )</td>
<td></td>
<td>0.3 \times ( V_{\text{DD}} )</td>
<td>( \mu \text{A} )</td>
</tr>
<tr>
<td>Input High</td>
<td>( q_{\text{IH}} )</td>
<td>( V_{\text{IH}} )</td>
<td>0.7 \times ( V_{\text{DD}} )</td>
<td>( V_{\text{DD}} )</td>
<td>( \mu \text{A} )</td>
</tr>
<tr>
<td>Input Loading (^{[7]})</td>
<td>( q_{\text{IN}} )</td>
<td>-10</td>
<td>-100</td>
<td>-300</td>
<td>( \mu \text{A} )</td>
</tr>
<tr>
<td>Response Time (^{[8]})</td>
<td>( q_{\text{GT}} )</td>
<td></td>
<td>2.0</td>
<td></td>
<td>ms</td>
</tr>
<tr>
<td>Status (^{[12]}) (^{[13]})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output Low ((I_{\text{LOAD}} = 100 \mu\text{A}))</td>
<td>( q_{\text{OL}} )</td>
<td></td>
<td></td>
<td>0.4</td>
<td>V</td>
</tr>
<tr>
<td>Output High ((I_{\text{LOAD}} = 100 \mu\text{A}))</td>
<td>( q_{\text{OH}} )</td>
<td>( V_{\text{DD}}-0.6 )</td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>Minimum Supply Voltage (LVD Trips)</td>
<td>( V_{\text{LVD}} )</td>
<td>2.7</td>
<td>3.25</td>
<td>4.0</td>
<td>V</td>
</tr>
<tr>
<td>Clock Monitor Fail Detection Frequency</td>
<td>( f_{\text{MIN}} )</td>
<td>60</td>
<td></td>
<td>280</td>
<td>Hz</td>
</tr>
<tr>
<td>Output Stage Performance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Electrical Saturation Recovery Time (^{[9]})</td>
<td>( t_{\text{DELAY}} )</td>
<td></td>
<td></td>
<td>0.2</td>
<td>ms</td>
</tr>
<tr>
<td>Full Scale Output Range ((I_{\text{OUT}} = 200 \mu\text{A}))</td>
<td>( V_{\text{FGO}} )</td>
<td>0.25</td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>Capacitive Load Drift (^{[10]})</td>
<td>( C_{\text{O}} )</td>
<td>300</td>
<td></td>
<td></td>
<td>( \Omega )</td>
</tr>
<tr>
<td>Output Impedance</td>
<td>( Z_{\text{O}} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mechanical Characteristics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transverse Sensitivity (^{[11]})</td>
<td>( V_{\text{HL}YZ} )</td>
<td></td>
<td></td>
<td>10</td>
<td>%FSO</td>
</tr>
<tr>
<td>Package Resonance</td>
<td>( f_{\text{PRG}} )</td>
<td></td>
<td></td>
<td></td>
<td>kHz</td>
</tr>
</tbody>
</table>
### Table 3-4 Operating characteristics of MMA1200

(Unless otherwise noted: -40°C ≤ TA ≤ +105°C, 4.75 ≤ VDD ≤ 5.25, Acceleration = 0g, Loaded output(1))

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Range(2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supply Voltage(3)</td>
<td>VDD</td>
<td>4.75</td>
<td>5.00</td>
<td>5.25</td>
<td>V</td>
</tr>
<tr>
<td>Supply Current</td>
<td>ID</td>
<td>3.0</td>
<td>—</td>
<td>6.0</td>
<td>mA</td>
</tr>
<tr>
<td>Operating Temperature Range</td>
<td>TA</td>
<td>-40</td>
<td>—</td>
<td>-125</td>
<td>°C</td>
</tr>
<tr>
<td>Acceleration Range</td>
<td>qFE</td>
<td>—</td>
<td>201</td>
<td>—</td>
<td>g</td>
</tr>
<tr>
<td>Output Signal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zero g (TA = 25°C, VDD = 5.0 V)(4)</td>
<td>VOFF</td>
<td>2.35</td>
<td>2.5</td>
<td>2.65</td>
<td>V</td>
</tr>
<tr>
<td>Zero g</td>
<td>VOFF,V</td>
<td>0.47 VDD</td>
<td>0.50 VDD</td>
<td>0.53 VDD</td>
<td>V</td>
</tr>
<tr>
<td>Sensitivity (TA = 25°C, VDD = 5.0 V)(3)</td>
<td>S</td>
<td>7.6</td>
<td>8.0</td>
<td>8.4</td>
<td>mV/g</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>SV</td>
<td>1.49</td>
<td>1.6</td>
<td>1.71</td>
<td>mV/gV</td>
</tr>
<tr>
<td>Bandwidth Response</td>
<td>f3dB</td>
<td>300</td>
<td>400</td>
<td>440</td>
<td>Hz</td>
</tr>
<tr>
<td>Nonlinearity</td>
<td>NLOUT</td>
<td>-2.0</td>
<td>—</td>
<td>2.0</td>
<td>% FS</td>
</tr>
<tr>
<td>Noise</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMS (0.1-1 kHz)</td>
<td>nRMS</td>
<td>—</td>
<td>—</td>
<td>2.8</td>
<td>mVrms</td>
</tr>
<tr>
<td>Power Spectral Density</td>
<td>nPSD</td>
<td>110</td>
<td>—</td>
<td>—</td>
<td>μW/(Hz)^(1/2)</td>
</tr>
<tr>
<td>Clock Noise (without RC load on output(4))</td>
<td>nCLK</td>
<td>2.0</td>
<td>—</td>
<td>—</td>
<td>mVpk</td>
</tr>
<tr>
<td>Self-Test</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output Response(7)</td>
<td>qST</td>
<td>55</td>
<td>75</td>
<td>95</td>
<td>g</td>
</tr>
<tr>
<td>Input Low</td>
<td>VIL</td>
<td>—</td>
<td>—</td>
<td>0.3 × VDD</td>
<td>V</td>
</tr>
<tr>
<td>Input High</td>
<td>VIH</td>
<td>VDD - 0.7</td>
<td>VDD</td>
<td>VDD</td>
<td>V</td>
</tr>
<tr>
<td>Input Loading(5)</td>
<td>VIN</td>
<td>-30</td>
<td>-100</td>
<td>-200</td>
<td>μA</td>
</tr>
<tr>
<td>Response Time(9)</td>
<td>tST</td>
<td>—</td>
<td>2.0</td>
<td>10</td>
<td>ms</td>
</tr>
<tr>
<td>Voltage (6)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output Low (ILOAD = 100 μA)</td>
<td>VOL</td>
<td>—</td>
<td>—</td>
<td>0.4</td>
<td>V</td>
</tr>
<tr>
<td>Output High (ILOAD = 100 μA)</td>
<td>VOH</td>
<td>VDD - 0.8</td>
<td>—</td>
<td>—</td>
<td>V</td>
</tr>
<tr>
<td>Minimum Supply Voltage (LVO Trig)</td>
<td>VLV0</td>
<td>2.7</td>
<td>3.25</td>
<td>4.0</td>
<td>V</td>
</tr>
<tr>
<td>Clock Monitor Fail Detection Frequency</td>
<td>tFM</td>
<td>50</td>
<td>—</td>
<td>160</td>
<td>kHz</td>
</tr>
<tr>
<td>Output Stage Performance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Electrical Saturation Recovery Time(12)</td>
<td>tDELAY</td>
<td>—</td>
<td>0.2</td>
<td>—</td>
<td>ms</td>
</tr>
<tr>
<td>Full Scale Output Range (IOP = 200 μA)</td>
<td>VFSO</td>
<td>0.25</td>
<td>—</td>
<td>VDD - 0.26</td>
<td>V</td>
</tr>
<tr>
<td>Capacitive Load Drive(13)</td>
<td>C2</td>
<td>—</td>
<td>—</td>
<td>100</td>
<td>pF</td>
</tr>
<tr>
<td>Output Impedance</td>
<td>ZO</td>
<td>300</td>
<td>—</td>
<td>—</td>
<td>Ω</td>
</tr>
<tr>
<td>Mechanical Characteristics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transverse Sensitivity(4)</td>
<td>VTXYZ</td>
<td>—</td>
<td>—</td>
<td>5.0</td>
<td>% FS</td>
</tr>
<tr>
<td>Package Resonance</td>
<td>τyz</td>
<td>10</td>
<td>—</td>
<td>—</td>
<td>kHz</td>
</tr>
</tbody>
</table>

#### 3.2.2 Pin diagram and connection details

The supply voltage for each sensor is provided at 5 V and is connected to the A/D input channel of the microcontroller. Each sensor can measure acceleration in positive and negative axis direction. The output acceleration is zero and the corresponding voltage is 2.5 V or VDD/2. With increase in acceleration in the positive direction, the voltage increases proportionally from 2.5 V to 5 V (for maximum output acceleration) and for acceleration in the negative direction, the output decreases below 2.5 V or VDD/2.
The $X_{out}$, $Y_{out}$ and $Z_{out}$ are connected to the PIC18F8722 via an RC filter network of 1kΩ and 0.01 μF to RF0, RF1 and RF2 pins. The RC filter network is used to minimize clock noise from the switched capacitor filter circuit. The pinout for MMA3204 and MMA1200 is shown in Figure 3.15 and Figure 3.16 respectively.

### 3.3 ZigBee module

#### 3.3.1 Introduction to ZigBee

Due to the compactness of the CONLISUS device, ZigBee standard protocol was chosen for wireless communication between the CONLISUS device and a standard laptop computer. ZigBee is based on the IEEE 802.15 standard [25] with a set of specifications defined by the IEEE 802.15.4 Wireless networking standards. It is used for establishing Wireless Personal Area Networks (WPAN) between computers and other peripheral devices. More specifically, ZigBee
can act as a replacement for physical data buses such as Ethernet, USB or RS-232. ZigBee is characterized by low data rates, low power consumption and long battery life. Finally, ZigBee can be used for relatively short distances ranges from of 10-100 meters.

The ZigBee network operates over the Industrial, Scientific and Medical (ISM) radio bands operating at 2.4GHz and 868/915MHz. The maximum data rates allowed for each of these frequency bands are 250Kbps at 2.4GHz, 40Kbps at 915MHz and 20Kbps at 868MHz. The addressing system uses from 16-bits up to 64-bit addresses so that there are 65,536 addresses (minimum). This means that at least 65,000 devices can be set up on a ZigBee network. There are two main modes of operations in ZigBee devices: Tx/Rx and Sleep mode. Sleep mode ensures low power consumption and longer battery life. In the Tx/Rx mode, continuous transmission and reception of data takes place. The device is always in use and leads to slightly more power consumption than when it is in sleep mode. As the name suggests, the sleep mode occurs when there is no communication. The device can be woken up from its sleep by external triggers such as when a signal is detected; or it can be programmed to go to sleep and wake up for specific intervals. ZigBee is also characterized by low cost and simple implementation. ZigBee networks can have star, peer-to-peer or mesh topologies as shown in Figure 3.17.

![ZigBee Topologies](image)

Figure 3.17 ZigBee Topologies
The ZigBee architecture can be divided into three different layers:

1. Network and Application Support Layer – The network layers permits the establishment of multiple devices over the network so that it can expand without the use of high power transmitters. The Application Support layer ensures the discovery of compatible devices and maintenance of tables that enable matching between two devices and communication among them.

2. Physical Layer (PHY) – It is for the physical and hardware layer. The ZigBee protocol enables the use of simple hardware and easy implementation thereby enabling high levels of integration. The physical layer enables communication between two devices.

3. Medium Access Control (MAC) layer – The MAC layer permits the use of different topologies in a simple implementation and enables the use of multiple devices over the network.

There are three different types of nodes in any ZigBee network. These are:

1. Coordinator node – Each ZigBee network has only one coordinator node that acts as the parent or root of the network. It contains the information relevant to the network such as the PAN ID, device ID, etc.

2. Router – A ZigBee router can be an intermediary transferring data from other devices. It is also known as a Full Function Device; it can not only communicate with the coordinator but among other routers as well.

3. End device – The end device can talk to the parent node but cannot communicate with other devices. So, it requires less memory and is low cost.
A ZigBee network is established by the coordinator, router and end device. The coordinator stores the information related to the network such as node information, message routing between nodes and manages the storage of node information. The router establishes communication between the coordinator and other nodes. Routers can transfer data between two devices. The end devices are the main source of information. They contain the data which can be sent to the router or to the coordinator/parent node. Figure 3.18 shows a general ZigBee network model. ZigBee is thus a low cost, easy to implement and efficient wireless protocol that can be used in place of a physical connection. The network has its own PAN ID so one network does not interfere with the other; moreover each device has its own address which facilitates network formation and data security. ZigBee network is thus a smart choice for the CONLISUS device as it serves all of device communication purposes.

![Zigbee network model](image)

**Figure 3.18** Zigbee network model
3.3.2 Xbee-ZigBee device features

ZigBee was implemented using Xbee radios from Digi International [26]. These are OEM RF modules that implement the ZigBee protocol. They are low cost devices with low power consumption. These modules operate on 2.4GHz frequency band. The Xbee series 2 low power devices are used to implement point-to-point or peer-to-peer network.

The indoor range of Xbee modules is 40 meters and the outdoor line of sight range is 120 meters. The maximum data rate is 250,000 bps and a full-duplex communication can be established. These Xbee devices support point-to-point, point-to-multi-point and peer-to-peer topologies. Further, they are low power devices with transmit and receive current as low as 40 mA. Finally, in sleep mode, operating currents are as low as 1μA. The power supply for these devices is 3.3V. Figure 3.19 shows the pin diagram.

The Xbee devices have the following modes of operations:

- Idle mode – When the device is not receiving or transmitting, Xbee is in idle mode.
  During the idle mode, Xbee is also checking for wireless data from other devices.
- Transmit mode – During the transmit mode the Xbee module transmits data to another

![Figure 3.19 Xbee pin diagram](image)
Table 3-5 Pinout of Xbee

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Name</th>
<th>Direction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VCC</td>
<td>-</td>
<td>Power supply</td>
</tr>
<tr>
<td>2</td>
<td>DOUT</td>
<td>Output</td>
<td>UART Data Out</td>
</tr>
<tr>
<td>3</td>
<td>DIN / CONFIG</td>
<td>Input</td>
<td>UART Data In</td>
</tr>
<tr>
<td>4</td>
<td>DIO8</td>
<td>Either</td>
<td>Digital I/O 8</td>
</tr>
<tr>
<td>5</td>
<td>RESET</td>
<td>Input</td>
<td>Module Reset (reset pulse must be at least 200 ns)</td>
</tr>
<tr>
<td>6</td>
<td>PWM0 / RSSI / DIO10</td>
<td>Output</td>
<td>PWM Output 0 / RX Signal Strength Indicator / Digital I/O</td>
</tr>
<tr>
<td>7</td>
<td>PWM / DIO11</td>
<td>Either</td>
<td>Digital I/O 11</td>
</tr>
<tr>
<td>8</td>
<td>[reserved]</td>
<td>-</td>
<td>Do not connect</td>
</tr>
<tr>
<td>9</td>
<td>DTR / SLEEP / RQ / D18</td>
<td>Input</td>
<td>Pin Sleep Control Line or Digital Input 8</td>
</tr>
<tr>
<td>10</td>
<td>GND</td>
<td>-</td>
<td>Ground</td>
</tr>
<tr>
<td>11</td>
<td>DIO4</td>
<td>Either</td>
<td>Digital I/O 4</td>
</tr>
<tr>
<td>12</td>
<td>GTS / DIO7</td>
<td>Either</td>
<td>Clear-to-Send Flow Control or Digital I/O 7</td>
</tr>
<tr>
<td>13</td>
<td>ON / SLEEP</td>
<td>Output</td>
<td>Module Status Indicator</td>
</tr>
<tr>
<td>14</td>
<td>[reserved]</td>
<td>-</td>
<td>Do not connect</td>
</tr>
<tr>
<td>15</td>
<td>Associate / DIO5</td>
<td>Either</td>
<td>Associated Indicator, Digital I/O 5</td>
</tr>
<tr>
<td>16</td>
<td>RTS / DIO6</td>
<td>Either</td>
<td>Request-to-Send Flow Control, Digital I/O 6</td>
</tr>
<tr>
<td>17</td>
<td>A03 / DIO3</td>
<td>Either</td>
<td>Analog Input 3 or Digital I/O 3</td>
</tr>
<tr>
<td>18</td>
<td>A02 / DIO2</td>
<td>Either</td>
<td>Analog Input 2 or Digital I/O 2</td>
</tr>
<tr>
<td>19</td>
<td>A01 / DIO1</td>
<td>Either</td>
<td>Analog Input 1 or Digital I/O 1</td>
</tr>
<tr>
<td>20</td>
<td>A00 / DIO0</td>
<td>Either</td>
<td>Analog Input 0 or Digital I/O 0</td>
</tr>
</tbody>
</table>

module. The destination address determines which node will receive the data. The Xbee module exits idle mode, determines the 16-bit network address of the receiving node and a routing path to the receiving node in order to transmit data.

- **Receive mode** – As a data packet is received by a node, it is verified for the correct destination address and then data is processed.

- **Command mode** – In this mode, the Xbee module is configured to participate in the network. In this mode, all incoming serial characters are interpreted as commands.

- **Sleep mode** – The sleep mode is what makes ZigBee so power efficient. End devices exclusively support sleep mode; the router and coordinator are intended to always check for incoming data packets and hence are always in active mode. The end device has to inform the parent node when it wakes up from sleep mode so that parent can transmit data.
### 3.3.3 Xbee module operation

The Xbee devices in the CONLISUS device are used for point-to-point communication. The Xbee devices can be connected to a host device through a logic-level asynchronous serial port. The Xbee device can be connected to a microcontroller via voltage compatible UART. The Xbee modules can thus communicate with each other via UART and send sensor data. Figure 3.20 shows the suggested connection between the Xbee module and microcontroller.

The DIN pin of Xbee module sends data to the UART as an asynchronous signal. The DOUT pin is used to send out data to other devices. Each data byte consists of a start bit (logic low), 8 data bits with the least significant bit first and finally a stop bit (logic high). If no data is being transmitted, Xbee has an idle high. For communication between two microcontrollers via Xbee, the two UARTs should have compatible settings, that is, same baud rate, parity bits, start and stop bits, etc. The arrangement in Figure 3.20 shows the use of Xbee as a replacement for serial UART connection between two devices. This is also a point-to-point communication between two devices; it is also known as the transparent or AT mode.

The Xbee device has to be configured for use in the AT/transparent mode. Each Xbee device has

---

**Figure 3.20 Suggested connection between microcontroller and Xbee modules**
its own 16-bit address assigned to it; the address is in the form of high address byte and low address byte. The transmitter sends data to the Xbee device whose destination address is configured into it. In order to set up a point-to-point ZigBee network using Xbee devices, there has to be one coordinator and one router or end device. The coordinator acts as the receiver while the router/end device acts as a transmitter. The coordinator is connected to a laptop via USB and the router is connected to a microcontroller via UART as discussed above. An exclusive PAN ID has to be defined for the network to prevent any interference with other ZigBee networks. The PAN ID should be the same for all the devices in the network; the PAN ID serves as a unique identifier for the network. Once the PAN ID is set up, the destination address is to be set up for the router and a source address for the coordinator. The destination address for the router is the address of the coordinator and the source address for the coordinator is the address of the router. Each node of the network can be given an identifier name such as router1, router2, etc. so that the user can identify them. The Xbee devices also have to be configured for other network parameters such as baud rate, parity bits, start and stop bits, etc. in order for a successful ZigBee network formation.

Xbee devices can be configured using the X-CTU software provided by Digi International. The device is connected to a laptop via USB and then configured using AT commands. In order for the device to be configured, it has to enter into command mode and then the parameters are written to Xbee. To enter into command mode, send three plus characters (“+++”) within one second. The device sends an “OK” signal if it is ready to be in command mode. There are several AT commands that are used to configure the Xbee. To exit AT command mode, we need to send ATCN command followed by a carriage return.
Table 3-6 List of AT Commands used to configure Xbee

<table>
<thead>
<tr>
<th>AT Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>+++</td>
<td>Enter command mode</td>
</tr>
<tr>
<td>ATBD#</td>
<td>Set up the baud rate of the module in AT mode. Each number corresponds to a specific baud rate</td>
</tr>
<tr>
<td>ATID</td>
<td>Check and set up the PAN ID of the network</td>
</tr>
<tr>
<td>ATV R</td>
<td>Check the firmware version of Xbee</td>
</tr>
<tr>
<td>ATNI</td>
<td>Check the node identifier (node identifier is the name given to the router or end device by the user)</td>
</tr>
<tr>
<td>ATCN</td>
<td>Exit command mode</td>
</tr>
</tbody>
</table>

There are several AT commands that can be used to configure the Xbee or read its parameters. Table 3.6 lists a few of them that are used to configure Xbee. X-CTU provides a direct method of configuration using the “Modem Configuration” tab where all parameters can be set at once and written to the Xbee module. Figure 3.21 shows a screenshot of X-CTU software being used to program a coordinator.

![X-CTU software screenshot](image)

**Figure 3.21 X-CTU software screenshot**
For PIC18F8722 and Xbee series 2, RC6/TX is connected to DIN and RC7/RX is connected to DOUT. Data from the microcontroller is sent to the Xbee module through the DIN pin as an asynchronous signal. A receiver connected to the laptop receives data from this microcontroller. To avoid any loss of data, the devices are always kept in transmit and receive modes respectively.

**Power supply and voltage regulators**

The CONLISUS device is powered using simple 9V batteries or a series of 1.5 V AA batteries. It can also be powered using coin or button cells. In order to use the same battery as a single power source, voltage regulators are used to produce the 5V supply for the sensors and microcontroller and a 3.3V supply for the Xbee modules. The MCP1801-3303 provides 3.3V and the MCP1801-5000 provides 5V.

The following are some of the features of the MCP1801 low dropout (LDO) voltage regulators:

- The MCP1801 is a family of CMOS low dropout voltage regulators
- The maximum output current can be 150 mA while consuming only 25μA of quiescent current
- The input voltage can be from 2.0V to 10.0V
- Output voltage can be 0.9V to 6.0V
- LDO is stable with a 1μF of output capacitance provided by ceramic, tantalum or aluminum electrolytic capacitors

The MCP1801 CMOS low dropout linear voltage regulator is used for applications where low current consumption is required. The source of a P-Channel PMOS pass transistor is connected to the input of MCP1801. The LDO stability is obtained by using low source impedance. Input
capacitance is provided by ceramic capacitors and can vary from 0.1 μF to 4.7 mF. A part of the LDO output voltage is fed back to the internal error amplifier and compared with the precision internal bandgap reference. The output of the error amplifier adjusts the current flow to the P-channel pass transistor. This regulates the voltage to the desired value. The error amplifier responds to any changes made to the input voltage or output current and thus it adjusts the output voltage to the desired voltage. The current limiter is used to limit the current level to less than 300mA so that desired output voltage is maintained while keeping low current. The SHDN input is used to turn the LDO output voltage on and off. To enable the LDO output voltage, SHDN is at logic high. An output capacitance of 1μF is used to maintain output voltage stability. Ceramic capacitors are recommended due to their size, cost and availability. Figure 3.23 shows the functional block diagram of MCP1801. MCP1801 has SOT-23 package with pin configuration as shown in Figure 3.22 with pin descriptions given in Table 3.7.

Table 3-7 Pin description of MCP1801

<table>
<thead>
<tr>
<th>Pin No. SOT-23-5</th>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>V&lt;sub&gt;IN&lt;/sub&gt;</td>
<td>Unregulated Supply Voltage</td>
</tr>
<tr>
<td>2</td>
<td>GND</td>
<td>Ground Terminal</td>
</tr>
<tr>
<td>3</td>
<td>SHDN</td>
<td>Shutdown Input</td>
</tr>
<tr>
<td>4</td>
<td>NC</td>
<td>No Connection</td>
</tr>
<tr>
<td>5</td>
<td>V&lt;sub&gt;OUT&lt;/sub&gt;</td>
<td>Regulated Voltage Output</td>
</tr>
</tbody>
</table>
Figure 3.24 Typical application circuit for MCP1801

Figure 3.24 shows the standard application circuit for MCP1801 with the input and output capacitance. SHDN is kept logic high for getting output voltage from the LDO. The absolute maximum input voltage that can be sustained by MCP1801 is 12V. The continuous power dissipation is 250mW.

3.4 Final design and schematic

The final design consists of incorporating all the components mentioned above on a single printed circuit board (PCB). The device was designed and developed using Eagle CAD PCB
software. Figure 3.25 shows the schematic of the device. Figure 3.26 shows the final PCB design. Figure 3.27 shows the final assembled board for the CONLISUS device.

The CONLISUS device is compact and small enough to fit into a helmet or it can be attached to a headband. The printed circuit board is a double-sided board. Its size is 2.2” x 2.2” and weighs about 21.21 grams. The cost of components for this prototype is about $70. The device also has an LED in its top right corner to indicate that it is working correctly. As mentioned earlier, the device is powered using 9V battery (not shown in the figure).

A power analysis was done for the CONLISUS device by summing the maximum currents of all components used in the design. The analysis was done for each of operations modes, namely:

- Run Mode – In this mode all the components are being used. This is the transmit mode of the system where continuous data transmission is done.
- Xbee sleep mode – In this mode, Xbee is asleep and no data transmission takes place. Data collected while the device is in this mode is stored in the microcontroller. This mode occurs when the Xbee receiver is out of range.
Table 3-8 Power analysis of CONLISUS device

<table>
<thead>
<tr>
<th>Mode</th>
<th>Current</th>
<th>Power</th>
<th>9V Battery Life</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run Mode</td>
<td>56.05 mA</td>
<td>212.25 mW</td>
<td>~50 hours</td>
</tr>
<tr>
<td>Xbee sleep mode</td>
<td>16.052 mA</td>
<td>80.255 mW</td>
<td>~96 hours</td>
</tr>
<tr>
<td>Idle mode</td>
<td>1.93 μA</td>
<td>6.59 μW</td>
<td>~10 days</td>
</tr>
</tbody>
</table>

- Idle mode – In this mode, all the components are in their sleep mode. The device is not collecting or transmitting any data.

Table 3.8 gives us brief description of the power consumption of the CONLISUS device in various modes. The table lists running current, power consumption and battery life assuming a 9V battery and continuous operation in the operating mode. From the battery life column we see that the battery life ranges from 2.5 days in run mode to 10 days in idle mode.
4. Experimental Verification

4.1 Preliminary testing

Preliminary tests of the CONLISUS device were done by using a) drop-test methods, and b) MB Dynamics CAL50 Exciter. The drop tests involved simply dropping the device and recording the measured acceleration. These drop tests were carried out for each orthogonal direction X, Y and Z. As anticipated for the acceleration due to gravity equal to g, the CONLISUS device measured 1g during each drop test orientation. The drop tests served as preliminary confirmation for the successful operation of the CONLISUS device.

A second set of tests was performed using the MB Dynamics CAL50 Exciter. The Exciter is a shake bed with an amplifier as shown in Figure 4.1. It has a moving element within that vibrates upon applying an input signal. The input signal is provided by a function generator. The moving element vibrates in a to and fro motion in one orthogonal direction. The input signal can be a sine wave, square wave or any other wave. The frequency of the signal can be varied from 0.5 Hz to 10,000 Hz. The CONLISUS device was placed on the moving element and a sine wave

![Figure 4.1 MB Dynamics CAL50 Exciter](image)
with a frequency of 1 Hz was applied through the function generator. Since the moving element vibrates only in one direction, only the Z axis was tested using this set up. The threshold for the impact detection device was set to be zero in this case. The graph of the Z-axis acceleration is shown in figure 4.2. The vibrating up-down motion of the moving element can be inferred from the graph due to the positive and negative accelerations.

The test in Figure 4.2 concludes that the CONLISUS device works and gives us acceleration in terms of g, acceleration due to gravity along with the time of the impact. However, for accurate measurements, a validation of the system is needed. A validation set up and associated experiments are discussed in Section 4.2.

![Acceleration from CAL50 Exciter](image)

**Figure 4.2 Acceleration of impact detection system from CAL50 Exciter**
4.2 Experimental testing and validation

An experimental set up was built to simulate head impacts and test the validity of the CONLISUS device. This experimental set up is used to roughly mimic head impacts and the force generated by them. Figure 4.3 shows the design and components involved in the experimental setup.

The experimental set up consists of a wooden track with a derby car on it. There is a bungee cord against which the derby car is pulled and collided against a spring. Steel screws (not shown in Figure 4.3) are fastened to the colliding face of the car and the spring. The heads of the screws face each other so as to create an impact during collision. This collision is used to mimic a head impact or any kind of blow to the head.

4.2.1 Functional testing

The CONLISUS device is placed on the derby car and then collided against the spring. The central position of the taut bungee cord is marked as zero and stretching distance is calculated further away from the central position. The cord is pulled to different stretching distances.

![Figure 4.3 Experimental setup for validation of impact detection system](image)
starting from 5 cm to 21 cm and impacts are measured for all these distances. The output is viewed on a screen using any standard terminal window program such as CoolTerm or RealTerm. The format of the data received is as follows: x acceleration, y acceleration, z acceleration and time in hh:mm:ss.msec. Each new data entry occurs on a new line. The data can be stored in a comma separated variable text file that easily allows additional analysis using common data analysis software.

Figure 4.4 shows a screenshot of the serial terminal program, CoolTerm that is used to view the data obtained from the CONLISUS device.

![Figure 4.4 Data being logged at the laptop](image-url)
Figure 4.5-4.12 show the impact measurements obtained for different stretching distances. The graphs show the entire duration of collision and the respective acceleration measurements. It also shows the sudden change in acceleration at the time of impact, and right after the impact. The graphs also give a detailed analysis of the acceleration before, during, and after the impact. It can be seen that as the distance increases, the intensity of the impact also increases.

Figure 4.5 Acceleration versus time graph for 5 cm

Figure 4.6 Acceleration versus time graph for 7 cm
Figure 4.7 Acceleration versus time graph for 11 cm

Figure 4.8 Acceleration versus time graph for 9 cm
Figure 4.10 Acceleration versus time graph for 13 cm

Figure 4.9 Acceleration versus time graph for 15 cm
The CONLISUS device can thus measure and transmit acceleration data. It can also be observed that the overall duration of the impact is about 8 ms. It takes about 2 seconds for the entire derby run to complete.
The acceleration data obtained through this device is collected in real time. It is recorded, sent and then stored to a laptop continuously. An impact can be immediately detected as soon as it occurs. Thus, this impact data could be used to check if the player has a significant head impact or not. The graphs in Fig 4.5-4.12 give all the acceleration data throughout the experiment. Since the acceleration data is received and stored on a laptop it can be processed later for further clinical analysis. It can also be used to keep a history of the athlete’s head impacts and injuries.

Several tests were conducted with the derby car to test its functionality and repeatability. It was found that the CONLISUS device yields consistent acceleration values for each impact.

4.2.2 Validation

The above tests prove that the CONLISUS device is able to measure acceleration. However, validation of the device is necessary in order to verify acceleration values. Validation tests were done using a Microchip PIC24 microcontroller demonstration board [25] and a displacement sensor connected to the derby car crash test system.

The validation method consists of attaching a spring with a spring constant of 11.751KN/m positioned around the shaft of a bolt on one side of the wall. A Panasonic sliding potentiometer is attached to the other end of the bolt. The sliding potentiometer is used to measure the compression, X, of the spring. When the derby car hits the spring-bolt arrangement, it compresses the spring. This in turn slides the linear potentiometer; the distance that the potentiometer slides gives us the value of X.

The compression of the spring can be used to calculate the acceleration. The compression of the spring along with the spring constant gives us a measure of the force by Hook’s law:
\[ F = Kx \]

The acceleration of the derby car can be found using Newton’s second law of motion:

\[ F = ma \]

Where \( F \) is the force found using Hook’s law and \( m \) is a mass of the derby car.

The CONLISUS device prototype is a double sided board. It needs a holding arrangement for proper measurements to be taken. It also requires a mount so that the orientation of the system could be changed. Accelerations can then be measured for all directions at different angles and elevation. The mount was designed in SolidWorks and printed using a desktop 3-D printer. The mount can be oriented with the X-Y plane normal to the collision wall. The X-Y plane can also be rotated from 0 degrees to 360 degrees on the mount. Finally, one edge of the mount can be elevated in the Z direction while the opposite edge stays connected to the X-Y plane. Thus, by elevating one edge in the Z-direction the CONLISUS device can be mounted in a plane that rises above the X-Y plane and an angle from 0 degrees to 60 degrees. A close-up image of the mounting assembly is shown in figure 4.13 [28].

The total mass of the derby car is the sum of the mass of the derby car itself, the CONLISUS device, the battery and the mounting assembly. The total mass was measured to be 558.5 grams. This mass was then used to calculate the acceleration of the car.

![Figure 4.13 Mount set up for CONLISUS device](image-url)
Experiments were then performed in order to validate the CONLISUS device. The initial orientation of the device was 0 degrees and its elevation was 0 degrees. The orientation was then changed in steps of 30 degree from 0 to 180 degrees and its elevation was varied from 0 to 60 degrees. The magnitude of the acceleration was calculated using the sum of the squares of the individual acceleration in each direction, as shown in the equation below:

\[ a = \sqrt{x^2 + y^2 + z^2} \]

Where \( a \) is the magnitude of the final acceleration, 
\( x \) is the acceleration measured in \( x \) direction; 
\( y \) is the acceleration measured in \( y \) direction; 
\( z \) is the acceleration measured in \( z \) direction.

The acceleration values obtained from this validation set up were then compared to the values obtained from the CONLISUS device. Figure 4.14 shows the peak acceleration values from both the CONLISUS device and the validation set up.

![Validation of acceleration](image)

**Figure 4.14** Comparison of peak acceleration values from CONLISUS device and the experiment set-up for different stretching distances
As seen in figure 4.14, the peak acceleration measured with the CONLISUS device tracks very closely with the acceleration determined by the validation system. The average relative error between the CONLISUS device and the validation set up is 4.08%.

Figure 4.15 shows the acceleration obtained at various elevations and orientations. Measurements from both the CONLISUS device and the validation set up are plotted. This shows that the system can accurately measure acceleration from any arbitrary orientation. Thus, the CONLISUS device could be placed in a sports helmet or on an athlete’s head and would accurately quantify head impacts for any direction.

Figure 4.15 Acceleration vector plot showing orientation and elevation of the system (starting from left to right: (φ, 0) values are (60°, 135°), (135°, 270°), (15°, 180°), (20°, 45°), (10°, 90°), (45°, 45°), (30°, 45°) respectively; (ρ, r) = (45.1, 44.5), (36.8, 34.9), (30.4, 29.8), (13.2, 11.9), (27.3, 26.5), (4.2, 3.7), (19.4, 18.2) respectively)
When the CONLISUS device is not transmitting any data, only the Xbee (ZigBee) module goes into sleep mode but the sensor and the microcontroller are in run mode. Tests were conducted to check the battery life of the device and it was found that the device has a battery life of about 2.5 days when running and transmitting continuously. Accelerations are measured at all angles, orientations and for different stretching distances. Therefore, the system can be used to reliably measure acceleration caused due to an impact. All of the above tests demonstrate proper functioning of the system.
5. Conclusion and Future Work

5.1 Summary

A prototype for head impact detection is designed and developed. The CONLISUS device consists of two accelerometers, a PIC microcontroller and Xbee transmitter/receiver connected to a laptop. The Xbee sends data over to the laptop which can be stored for further use and analysis. The CONLISUS device can be used in contact sports or areas where the head is prone to mild traumatic brain injury. The system can be used at any orientation and elevation and can determine the magnitude of the acceleration. It can be easily attached on to a helmet or worn as a head band. Thus, it is a point-of-care device that can aid in the detection of a heavy impact or blow to the head that might contribute towards concussion.

The CONLISUS device is a low cost wireless device that can be used to reliably measure acceleration upon head impacts. The initial manufacturing cost of the prototype is $70 but large scale production could bring down the cost significantly. In addition to measuring acceleration the CONLISUS device gives the time of an impact and its duration. Since the system uses ZigBee RF communication protocol, each player on a team can wear a CONLISUS device that connects to a single laptop receiver enabling an entire football or soccer team to monitor for head impact detection during a game or practice. The wireless receiver requires a terminal program like CoolTerm or RealTerm to receive data. This makes the CONLISUS device platform independent; it can be connected to any computer with any operating system that has a USB port and terminal program.

The CONLISUS device functions like standard plug-and-play devices. It can be used by anyone to detect sudden impacts or blows. It is a robust, compact system that can aid in the early
detection of mild traumatic brain injury. It transmits data to the laptop immediately; a large amount of impact data is collected. This data can be stored in the laptop memory and can be used for further analysis by researchers, physicians or coaches. It can also give a detailed report of the athlete’s performance during a game or practice. A large collection of such data over a long period of time can be useful to study the long-term effects of head impacts.

5.2 Future Work

The present CONLISUS device can measure linear acceleration using the MMA3204 and MMA1200 Freescale accelerometers. These sensors are orthogonal linear accelerometers that can measure acceleration at any orientation. The resultant acceleration from all three axes gives us the magnitude of the acceleration and its angular orientation. However, this set of sensors is not appropriate for measuring angular acceleration. A device that included both linear accelerometers and gyroscopes could be used to measure both linear and angular acceleration and could provide more detailed information about the head impact and its effect on the brain.

In the present system, the battery is placed outside the system. A new system may be developed that has a built-in battery; this might have to be done without compromising the size of the system. The present system uses 9V battery as input for the voltage regulator but the low dropout voltage regulator MCP1801 can have a variable input from 5V to 9V. So small button cells or watch batteries can be used as input for the voltage regulator.

The system sends data to a laptop via a receiver. This data can then be used to analyze the impact effects. An application can be developed to collect and analyze this data on a tablet or a smart phone rather than a laptop. The tablet could be connected to the receiver and instant results could
be obtained. The system can become a truly platform independent system that can be used over a wide range of devices.

The system has demonstrated that it can be used to measure acceleration caused due to impact during contact sports. Therefore, it could be actually used to study impacts during a game. A pilot study can be planned and executed to observe and analyze impact data during a practice session of a football game. A preliminary use of the system in the field can help in further improvements and enhancements of the system.
References


2. Heads Up, Concussion in football, USA Football, CDC


4. On-field football concussions, Seegerweiss LLP


21. Datasheet of PIC18F8722 microcontroller

22. PICkit3 user guide, Microchip Technology Inc., 2009

23. Datasheet of MMA3204

24. Datasheet of MMA1200


26. Datasheet of Xbee series 2