Department of Electrical Engineering and Computer Science

EEI-4914 Senior Design 1

Motorcycle Tracking Security System

Group #8

Andres F. Suarez

Brian Maldonado

Rigel Jimenez

Table of Contents

List of Figures

List of Tables

Section 1: Introduction

1.1 Executive Summary

Electrical Engineers have always been recognized as main contributors in the technology sector. All of our group members wanted to use this opportunity to build a technological Senior Design Project that will improve one of the many activities that a person can perform. There are quite a number of motorcycle riders around the world; so we decided to focus on building a prototype that we named Motorcycle Tracking Security System (MTS), and its main duty is to significantly enhance the motorcycle riding experience. Our plan is to make a modified version of the typical motorcycle helmet, and in addition to that we are going to place a unit on the motorcycle itself; with this specific design the motorcyclist will be able to perform a whole bunch of features that will make his or her riding experience more fun, secure and sophisticated.

Motorcycle Tracking Security System (MTS) is going to solve many of the problems that a motorcycle rider encounters on the daily basis. The Senior Design Project will be composed of many different subsystems that will response according to the user needs. One of them will be the alarm subsystem that will increase the security features of the bike, since many stock models do not come with one incorporated. The system will also some different modules that will allow the biker to interact with his or her personal smart phone, in order to have access to make and receive phone calls while riding, as well as listening to music, or some other audio recording that the cell phone can provide. Finally, Motorcycle Tracking Security System (MTS) will also have the ability to track the position of the bike by having a GPS incorporated, and will be able to send information to the user and an emergency contact, in case of and accident.

Choosing a Senior Design project was not such an easy task; in fact, it required a lot of brainstorming because we wanted to come out with something that would show our Electrical Engineering skills, but at the same time was useful for the community. Motorcycle Tracking Security System (MTS) is basically an integration project that will give the opportunity to all the three member of the group to learn about different technologies that provide security and communication features which are quite popular in the market right now. It will also give a better insight on how to connect all the systems and make them work as one; by doing a proper integration, coding, and manipulation of a microcontroller or unit to process data. Throughout this document, the reader will see step by step how the Senior Design Project is going to be done. It will start with some motivational aspects of the project, as well as goals and main function of the project; then it will show an extensive research of the different modules and subsystems required to integrate the main system; and finally, it will show the design of the different parts of the project, as well as the schematic of the full integration of all the parts.

1.2 Motivation

Our team wanted to come with a solution to a problem. The problematic situation that we are taking into consideration is the simplicity of the common used motorcycle helmet. According to the National Highway Traffic Safety Administration (NHTSA), there were over 7 million motorcycles registered in 2007 in the United Stated, and the number just keeps increasing. As we can see there is a huge demand for motorcycles, therefore for safety regulations the government has established laws around the different states requiring the use of this protective headgear.

Rigel Jimenez is one of our group members, and he is also a motorcycle owner, he really enjoys using his ride, but at the same time he knows that besides riding, there is not much he can do with his helmet. Well that is about to change because we are going to transform the regular helmet into a more technological device that will allow the user to not only be safer, but to be able to enjoy more features that some other people are currently enjoying in some other transportation vehicles.

All of us decided to pick Electrical Engineering as our major because we want to be active contributors in this society, and our main motivation is to make this world a better. We might not be able to solve all the issues that the humanity currently have, but at least we can use our imagination and abilities to come out with ideas, products or projects that can make life somehow better.

1.3 Goals and Objectives

Our group wanted to learn about many different aspect of technology. One reason we picked this design is to incorporate many different technologies that are available and learn as much as we can about each. The MTS system allows us to use many different sensors to provide different functions. This way our group will receive great experience with the different type of technologies. Below we list the goals and objectives:

- Adding the electrical components to the helmet without increasing the weight of the helmet very much.
- The parts and tools to be cost efficient to make it a worthwhile investment for companies to create and consumers to buy.

- Maintaining a simple user interface and the interface should not require a lot of physical effort to maintain the riding experience as safe and comfortable as possible.
- Making the riding experience safer and more multitasking, the motorcycle owner should be able to use his bike as usual, and on top of that, the rider should be able to enjoy the comfort and applications that a Bluetooth device can provide to his or her user.
- Making the riding experience more fun and productive by giving the opportunity to the motorcyclist to do stuff that he or she could not do before.
- Quick emergency response, the system will contact 911 immediately in case an accident occurs.
- Using a GPS system to acquire location.
- Being able to sync the helmet to your cell phone or some other Bluetooth device to make phone call and listen to music, and also being able to play music or make phone calls without having a Bluetooth headset or headphones.
- The system should be able to send text messages.
- Providing security features to the helmet and the bike.
- Implementing an accelerometer to the bike and helmet.
- Keeping the helmet as spacious as it was originally, even though it will now have all the electronic components.
- Incorporating a security alarm system for the motorcycle and the Helmet.
- Picking a helmet that has already been manufactured, and that has enough safety characteristics. We are planning on limiting our total spending amount, so we might pick a used one that is in really good condition, or a new one that has a reasonable price.

1.4 Requirements and Specifications

Our design will have several features on the motorcycle and helmet that require different types of sensors. Global Positioning Systems will be used to monitor the location of the motorcycle. This will provide the location of the bike if it is stolen. To know if the helmet is being taken or not a Radio Frequency module will provide the link between the helmet and the bike. The link will always be established when the alarm system is in its ON state. A sensor in the helmet will detect unauthorized movement. Once the threshold is breached it will send the command to the main module, which sets off the alarm. As the alarm is triggered the rider will be noticed via text. Aside from security features it will also have some convenient features for the rider. For that a Bluetooth chip will be placed to allow communication using the rider’s cell phone. For this system to be complete it will also have speaker to provide audio output and a microphone for audio input.

The motorcycle will also have its own features. As mentioned before it will have a GPS chip to be able to track its location. An accelerometer will be placed to monitor for sudden changes in g-force that would signify an accident. This will trigger the GPS to send out the latitude and longitude location to an emergency response contact. In the case of possible unauthorized movement of the motorcycle, a trigger will be put to sound of an alarm to scare off the violators.

As far as the power source, the motorcycle itself will be used to power the microcontroller and its components. A voltage regulator will be used to make sure each component gets the right amount of input voltage. The helmet will have a battery pack to power up its own components. The batteries are going to be rechargeable and have a circuit to charge up the batteries so they don't have to be taken out each time.

1.5 Block Diagram

The Motorcycle Tracking Security System is composed by two main sub-systems. The first sub-system will have different modules and it will be located on the bike; the second one will work together with the first one, and it will be placed on the motorcyclist's helmet. Figure 1 below depicts the block diagram for the motorcycle side of the design. This design is driven by the motorcycle voltage supply of the battery and centered about a main control module. This could be a microcontroller or an RF module of some sort depending on the degree of complexity of the processes being implemented. This diagram shows two external communications one that communicates to external emergency contacts in case of an accident for notification and safety purposes, the other external routes to the helmet through an RF medium of communication. The RF establishes the communication link to the helmet to send and receive data.

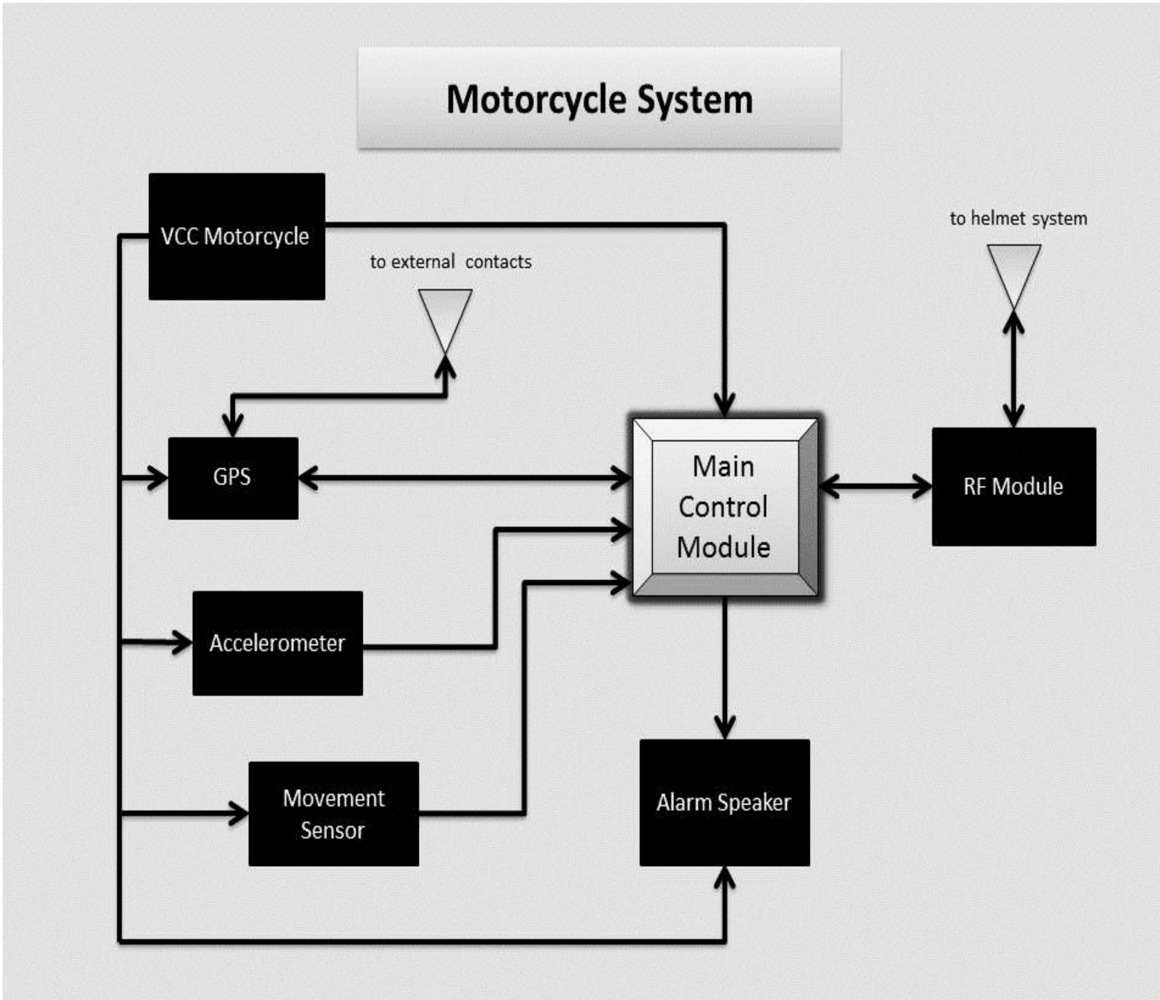


Figure 1: Motorcycle System Block Diagram

Figure 2 below depicts the block diagram from the helmet design's perspective. This is the system that will be hopefully implemented around and within the helmet. Similarly to the motorcycle's block diagram this block diagram contains familiar elements including a secondary control module and further communication systems between the helmet and externals. The three communication peripherals include communication between the helmet and the user's cellular device via Bluetooth. The second peripheral connects to the motorcycle's control unit, which is the medium of data transfer between them.

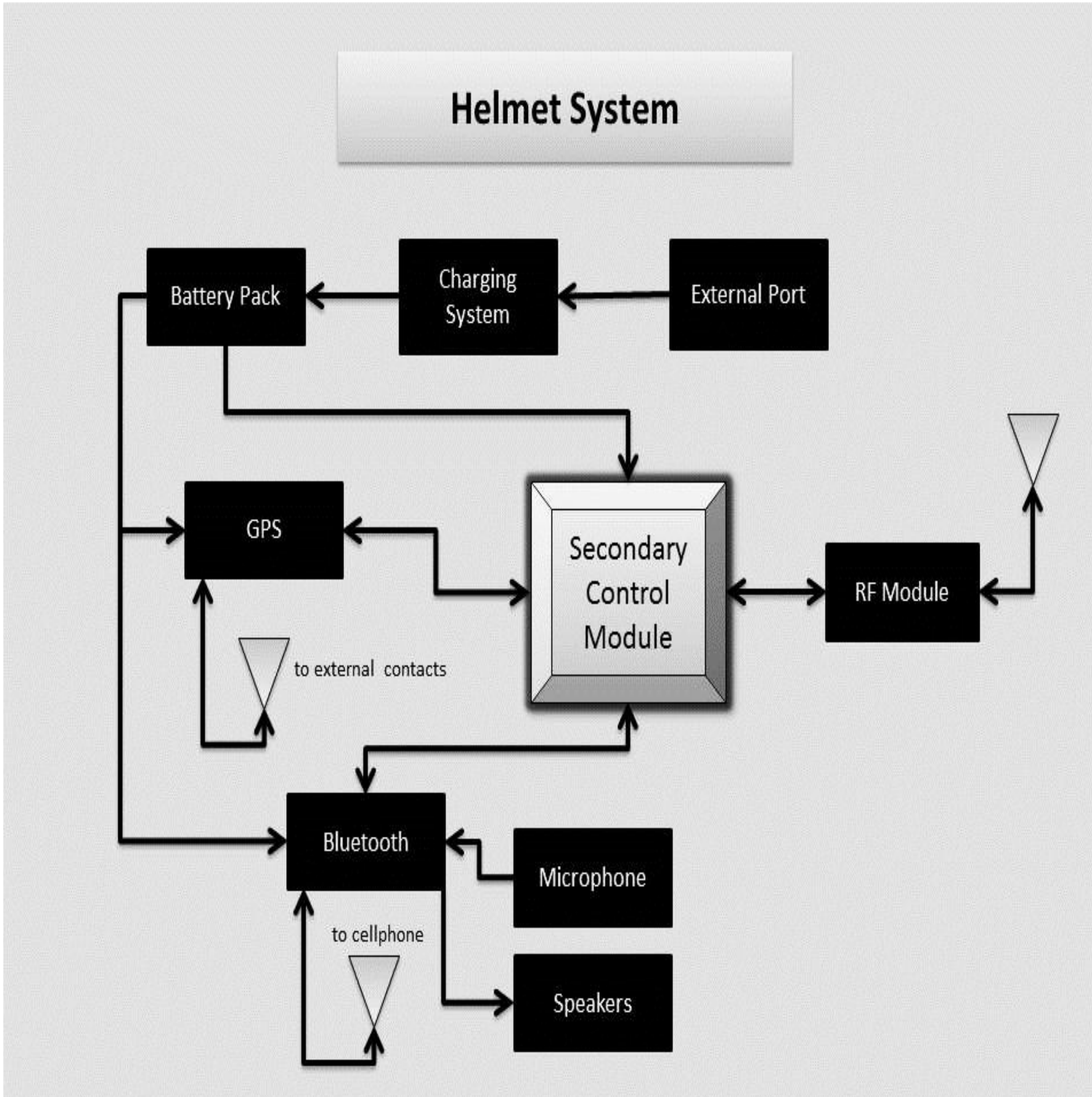


Figure 2: Helmet System Block Diagram

1.5.1 Distribution of Block Workload

MTS is a project based on teamwork. A proper and well-balanced workload distribution is essential to obtain optimal results. Table 1 below shows the topics selected by each member of the group.

	Rigel	Brian	Andres
Driving Power Supply	X		
Control Module	X		
RF peripherals and communications	X		
Accelerometer		X	
Motion Sensor		X	
Battery and Recharge Circuit		X	
Bluetooth			X
GPS			X
GSM			X
Audio Input and Output			X

Table 1. Workload Distribution

1.6 Project Function

The Motorcycle Safety and Security System incorporate a more technological version of the original bike helmet. Our design will allow the user to have access to Bluetooth technology without the use of a Bluetooth headset. As we mention before one of our group members is a motorcycle owner, and he tried many time to use his headphones or headset while riding, and they always fall off, and the reception is not that good. The Bluetooth incorporated in the helmet will give access to the user to a new amount of features that originally did not have while riding. The motorcyclist will be able to sync the helmet to his or her personal cell phone or electronic device with Bluetooth capability in general, depending on the device used, the person will be able to make phone calls, listen to music, or perform some other applications that uses voice recognition.

We are also going to implement GPS technology to obtain our current location and to get information of the places we want to go to. This will help us improve our riding experience. A microphone and a good audio system will be added to the helmet too, so we can efficiently interact with our cell phone. Our design will use voice recognition to communicate with the external devices that we are interacting with.

Most of our circuitry will be located underneath the motorcycle seat to keep the helmet as specious as possible; and to make it look nice we are going to put everything inside on a box that will protect the circuitry, and at the same time the it will be as light as possible to keep the weight of the bike about the same.

We are also planning on adding a basic security alarm system to the project, it is important to mention that the speakers that the alarm will activate are not the ones we are planning on placing on the helmet, they are going to be located in the bike, and the main reason we are doing this is to keep the helmet with as much room as it comes originally from the manufacturer.

Section 2: Research

2.1 Existing / Related Projects

When we decided on this project to design we knew that there are many different variations out there and had to make ours more unique then what is out in the market already. We made it unique by either adding extra features that are not commonly found in these types of products or by combining features you find in different products into one. Making our design stand out and be different from anything that is available on the market. Security system products are readily available in many different forms. It is an area of market that has been targeted for many years since keeping things safe has always been a concern for belongings. That includes motorcycles, because we don't live in a perfect world there will be people out there that are going to try and steal it and people want security to keep that from happening. There are a lot of systems out there so one of our goals is to make it one of the most advanced and most effective.

There is only one company that has a product very similar to what we are aiming for which makes it that much easier to make ours more unique. The company Phantom Tracking offers a similar product to our design. It is a GPS motorcycle tracking and security system that offers protection to the motorcycle and safety to the rider. If the motorcycle is stolen even with the alarm on, it can track the location of it through GPS making only a matter of time before the thieves are caught. It comes with crash detection, motion sensing, GSM/GPRS messaging. All this features are also featured in our design but where ours stand out that it's the complete package where in not only includes the motorcycle and rider, but it includes the helmet as well. And by the complete package we mean all components needed for riding are protected with some kind of technology that we are implementing. The Phantom Tracking covers overall the motorcycle and rider but not the riders' helmet. Our project offers communication to the helmet for safety and security.

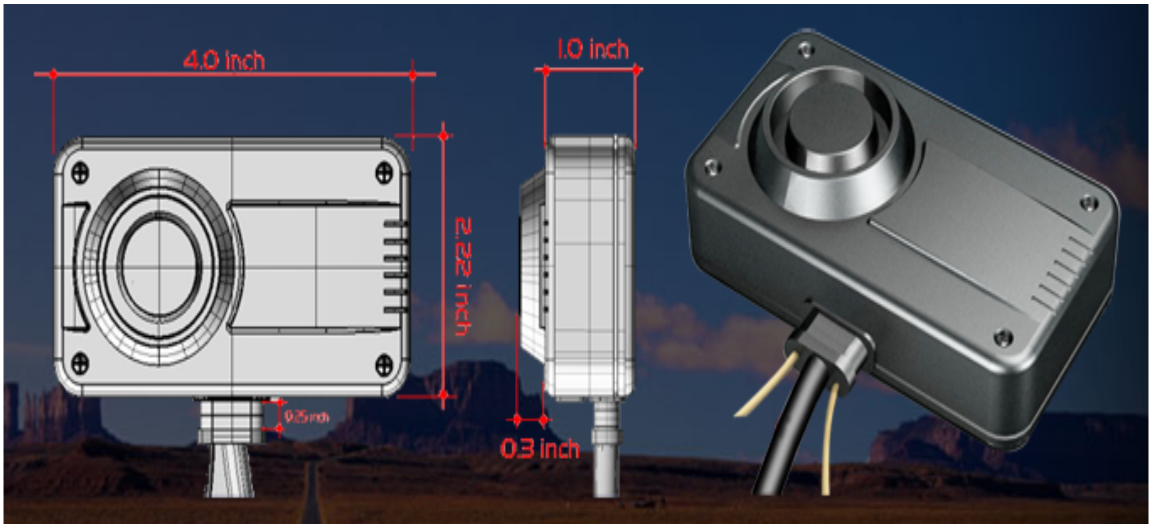


Figure 3: Phantom Tracking System

(Getting permission in progress)

How the Phantom Tracking focuses on the motorcycle, there are many other products that focus only on the helmet. Implementing technology in a helmet has been a very popular demand especially since technology is becoming more and more advanced every year. The biggest implementation to be put in a helmet is Bluetooth. It allows the rider to be able to communicate using the Bluetooth from their phone. Another feature that Bluetooth gives is the ability to listen to music while riding.

One company that has been leading the competition in this is ChatterBox. They have several different type of product that center around Bluetooth technology. There product is simple and small enough that it attaches to the side of the helmet. Then once its connects with the riders phone it can make phone calls, listen to music, and get GPS directions all wirelessly. Another cool feature they have is that it can communicate with other riders that have the same product with the same capabilities. Now where our design stands out is with extra features. The Bluetooth features are pretty much the same since our device has the same technology. GPS will be an extra feature for the helmet that will be embedded in it. This is to protect your helmet from theft. With GPS the helmet can be tracked down to its location. It is also going to have GSM to be able to transmit its location when needed. With these extra features it makes our design different and unique from what is out in the market today.

Figure 4 Chatter Box

(Getting Permission in progress)

2.2 Wireless Communication

In our current era wireless communication devices are pretty much everywhere; in fact, to successfully develop this Senior Design project, the use of this technology is essential. But realistically, even though the term sounds familiar, and as it was mention before, almost every single person interacts with one of these devices in a daily basis, there is a lack of knowledge from our part on how to build a communication system that implements this technology, and on the different options that are in the market right now that can help build this system and added to the motorcycle helmet. First of all, it is important to define this term; wireless communication is the technology exchanging of information between two or more points that are not connected by an electrical conductor. This kind of technology is a must have for the project because one of the main features of the helmet is the ability to connect to a cell phone, in order to make phone calls and listen to music. The research that follows looks into some of the most popular methods available for this project. Some of the wireless communication technologies that will be looked into are Wi-Fi, Infrared, Bluetooth, and RF communication. Also, some key factors that will take into consideration are the data rate, the range, and connectivity to the cell phone brand and model that the project will be using.

2.2.1 Wi-Fi

Wi-Fi is the first protocol taken into consideration for this project, so let's go ahead and define it. In general, Wi-Fi is a very popular technology that allows transferring data wirelessly using radio waves over a computer network, including high-speed Internet connections. Wi-Fi stands for Wireless Fidelity, but it is also known as wireless local area network (WLAN), and it is based on the Institute of Electrical and Electronics Engineers (IEEE) 802.11 standards. Wi-Fi operates in the 2.4 GHz or 5 GHz radio bands, and it also provides a well-established connection and this is very helpful because it compensates network congestion and it helps minimize the errors on the connections. This technology is very popular among computer networks, as well as in video game consoles, cell phones, personal computers and tablets.

The main goal of Motorcycle Tracking Security System (MTS) that is being design is to make the riding experience safer and at the same time more enjoyable. Being realistic, who would not like to be able to ride a bike and at the same time listen to music, make phone calls and have security features. Being able to make the system wireless is a key factor in this design, because a person would not be comfortable driving a bike, while having a whole bunch of wires hanging, this will produce a hazardous situation. Wi-Fi is pretty much everywhere, so that is why it comes to mine as soon as you talk about wireless communication, but what makes it appealing to this project. A fast transferring of data is a factor that makes the performing of the design better, and among all the wireless technologies, Wi-Fi has the highest data rate transfer; in fact, its performance is very similar to a wired connection. Not only that, this protocol is extremely secure, and this is a feature that you want, if you are sending personal information such as the location of the motorcycle rider, in case of an accident.

This project is trying to keep a low consumption of power, so let's see how is the power consumption for this technology. The average power consumption for this technology is the highest compared to other wireless methods when transmitting over long distances. This is something to really take into consideration because what is the point of creating a motorcycle and helmet security system that is very secure, but consumes a lot of power. If a Wi-Fi enabled transmitter is implemented in the senior design project, the battery life of the device will be highly affected, and that is something that a user would not like. The customers or users of any electronic device always want a product that is reliable and has a good performance.

<u>Advantages:</u>

- Wi-Fi technology provides a long-range of coverage.
- It has the highest data range. Its performance is very similar to a wired connection.

- Its signal is not easily blocked.
- It has a low cost compare to other technologies that offer a similar performance.
- It works on the unlicensed 2.4 and/or 5 GHz range.
- It is very easy to integrate with a Smartphone. Even though we are limiting to IPhone and maybe Android cell phones.
- This technology is quite popular in the market because it is very secure.

Disadvantages:

- It requires high power consumption, which is something we cannot afford
- Slight signal interference may occur.
- The configuration of a network based on it can be complex.

After understanding this wireless communication method better, it became clear to our group members that the implementation of this technology would not be such a great idea, because we are trying to minimized the power consumption as much as possible. But, this protocol will not be completely rejected, it also has some good features such as the high security levels, and it might be consider as a back up plan later on when we are actually building the prototype. Our knowledge about wireless communication has increase, but we need to keep researching about some other methods that we might be able to use in the project.

2.2.2 Infrared (IR)

Another wireless communication method available in the market is Infrared (IR). This protocol has been out there for quite a while, and we daily interact with products that use it. This technology allows the transmission of data within a short range using infrared beams. Its current applications in the Electrical Engineering field show us that it could also be an option when it comes to building the motorcycle and helmet security system. Infrared is widely employed among computer peripherals and personal digital assistants. It is important to know that when you are limited to only two devices at the time when you are using infrared; also, they need to be in a direct line of sight so that the devices can detect each other and communicate.

Advantages:

- It has low rates of power consumption.
- Decent speed, it has a transmission speed up to 16 Mbit/s.
- It offers a safe transmission.

- The technology has been used for a long time, so it has been studied a lot, and there is a lot of information about it.

Disadvantages:

- The connection is limited to a short range, but in our case, we can deal with this.
- It is limited to only connect two devices at a time.
- It requires a direct line of sight to operate properly.
- It does not go through objects, which is something that we need to implement.
- It can produce a bad signal or the connection can be interrupted due to many factors such as: the distance between the two objects that are being connected, a wrong angle, noise, heat, or light waves.

Infrared has been found to be successful in many electrical engineering projects, and it is a well-respected technology because it has been in the market for quite a while. Unfortunately, according to its characteristics, it has a lot of disadvantages that will make our project very inefficient when it comes to performance. Even though is able to transfer data in a short range, which is what we need, it has a lot of weak points. We need to have a reliable wireless communication module that will be able to transmit the signal in an efficient way, so the motorcycle rider feels confident and satisfied with his or her helmet.

2.2.3 Radio Frequency Transmission

According to webopedia.com RF or Radio Frequency is referred to as “any frequency within the electromagnetic spectrum associated with radio wave propagation.” Usually RF transmission follows standard electrical signal propagation format. An initial RF current is supplied to an antenna, which in turn creates an electromagnetic field that then begins to propagate through space. It is interesting to know that this simple idea is the most basic grounding foundation for many new modern wireless technologies out in the field today.

It should be noted that there are also different classifications of radio frequency oscillations. These range from ELF or extremely low frequency to THF with frequencies of 3-30 Hz and 300-3000 GHz respectively. As noted there are just too many different forms of frequencies propagating through the air and in turn this is where the use of a radio tuner comes in. The radio tuner will enable us to tune into a particular frequency of preference. The tuner is usually a pretty simple resonating circuit (e.g. an LC circuit, which forms a filter and in turn helps with forming a pass-band and a rejection band to filter out unwanted frequencies and keep those that are of interest. Frequencies within our pass-band are accepted while all others are attenuated.

Perhaps one of the most important communication systems being used in our design would be the RF transmission module. We are faced with a variety of communication systems in our design. The use of Bluetooth will be primarily useful for voice commands, calls, music and synchronization with the user's cellular device while the GSM chip would be primarily useful for text messages and notifications. Aside from those two systems we have one of the most integral communication system involved in our design, the RF frequency transmission module. In essence, we want to dedicate the RF module to be used as a means of communication between the motorcycle and the helmet.

If we go further into specifications we have a few parameters to consider and tailor to cater to our specifications. We do not want to have too much power in our hands or too little in which case we need to take weight of what we need and exactly just how much of it we need. For our wireless module we are mainly concerned with range of transmission, data rate and network acquisition time. We desire to run our project in real time; therefore network acquisition serves an important role here.

The rider will ideally have a remote to control the alarm and certain parameters of our system. The range does not need be very long, as ideally key-less systems are not designed for long-range applications. The idea here is that if we provide the user with a remote to control the system, we would intend it to be utilized in a very similar fashion to that of a key-less FOB remote of modern cars today.

Following that, we run into data rate, small packets will be sent back and forth between our nodes therefore data rate does not need to be very fast for effective use of our module. Low data rate is sufficient for the remote aspect of our project however for our Bluetooth module, data rate might be more of a priority as opposed to range due to the fact that we will be dealing with commands, phone calls, music streaming and general status notifications.

Different wireless technologies were considered when it came to choosing which fit the project best. There are just so many out there which many could be adapted to suit our project specifications however the group discussed a few options some more complex to implement than others but nevertheless feasible options. These few were Bluetooth, wireless USB or a ZigBee/Xbee module. Table 2 below shows a comparison chart between these modules.

Specification	Wi-Fi	Bluetooth	ZigBee
Data Rate	54 Mb/s	3 Mb/s	240 Kb/s
Range	100m	100m (class 1)	100m
Networking Topology	Point to Hub	Ad-Hoc	Ad-Hoc/PTP/mesh
Operating Frequency	2.4 GHz	2.4 GHz	2.4 GHz
Power Consumption	High	Medium	Very Low
Network Acquisition Time	3 – 5 s	< 10s	30ms

Table 2. RF Transmission Technology Options

2.2.3.1 ZigBee Module

On the same footnote, I would like to write a bit about a very important and crucial communication system that we want to implement in our system design. ZigBee is a protocol that can be directly embedded into various applications at the general standard 2.4 GHz operating frequency. ZigBee is similar to Bluetooth as it can use radio frequency and has the ability to link to multiple nodes. It is of course low power consumption, which makes it appealing due to the limited space in our design. ZigBee as well as XBee were considered for this project. ZigBee's parameters are reasonable to us as it is low power, low network acquisition time and low data rate. We will most likely be using an Arduino board, which does not require high data rate, making ZigBee a pretty good fit for our project.

ZigBee features Carrier Sense Multiple Access/Collision avoidance mechanisms, which allows communication to be very efficient and free of collisions for the most part. The interface would only send out single signals to the board only when user input is detected. Both nodes would be in sleep mode for most of the time until mutual acknowledgement is received. After that, nodes will communicate and return to a dormant state. ZigBee protocols tend to reduce the time devices are in active mode in order to manage power as efficiently as it does. Its low network acquisition of 30ms makes it a perfect candidate for real-time applications.

Further scrutinizing revealed that ZigBee provides numerous options and features. Perhaps we could consider not only using one, but 2 of these modules to implement additional features or further optimize our project to increase efficiency and processing. Parameters considered here were frequency, RF line of sight range, transmission power, receiver sensitivity, data rate and cost. When these were taken into consideration we were able to narrow down some more possibilities for our RF modules. It should be noted that all of these also require a line of sight, which means that communication might be jeopardized by intervening objects. In the table below are the three different candidates found suitable for our project.

Specification	XBee 802.15.4	XBee ZB	XBee Digi Mesh 24
Topology	Point-to-Multipoint	Mesh	Mesh
Frequency	2.4 GHz	2.4 GHz	2.4 GHz
RF Line of Sight Range	90m	120m	90m
Transmission Power	0 dBm	3 dBm	0 dBm
Receiver Sensitivity	-92 dBm	-96 dBm	-92 dBm
RF Data Rate	250 Kb/s	250 Kb/s	250 Kb/s
Cost	$19.00	$17.00	$19.00

Table 3 ZigBee Wireless Module Specifications

2.2.4 Bluetooth

Now let's get some information about Bluetooth, which is one of the most popular wireless communication protocols in the market. One of our goals for this senior design project is to be able to transfer data wirelessly within a short distance, and this is basically the primary goal of the Bluetooth technology. This technology exchanges data using short-wavelength radio transmissions in the industrial, scientific and medical (ISM) band from 2400-2480 MHz from fixed and mobile devices; and it also creates personal area networks (PANs) with high levels of security, which is something that we will really like to implement in our project. We want the user to feel as secure as possible when synching his or her devices with the helmet.

This is a wireless technology that enables communication between Bluetooth compatible devices, and do not think of this as a limitation because currently almost every cell phone or electronic device has a Bluetooth module on it due to its high efficiency and security features. The IPhone 5, which is one of the products that we are planning on synching with the helmet, it has Bluetooth built-in and it offers an easy synching process, which is something that the motorcycle rider will really like. Bluetooth is used in cell phones to send files between two or more phones, or from the cell phone unit to a computer and vice versa. It is also used to connect to headsets or wireless earpieces, which is something that needs to be incorporated in the project. The helmet should have an earpiece and a microphone to communicate to the external devices that are being synced. One thing to take into consideration is the fact very comment to use the same headset to connect with different Bluetooth devices, and this is definitively a feature that we want for our project. We might be limiting the types of cell phones that can be synched to the helmet, but we still want it to sync to more than one device. In this case, multipoint pairing is essential. In order to achieve multipairing an industry standard color code is needed, this allow the device to distinguish one pair from the others.

As it was mentioned previously, analyzing the range of coverage is important to choose the right wireless protocol for the project. Bluetooth is a protocol that is mainly designed for low power consumption, and the range changes according to its power class type. The different Bluetooth classes and its corresponding ranges are shown in the table below (Table 4).

Class	Maximum Power	Operating Range
Class 1	100mW (20dBm)	100 meters
Class 2	2.5mW (4dBm)	10 meters
Class 3	1mW (0dBm)	1 meter

Table 4: Bluetooth Classes

Table 4 clearly shows that a Bluetooth device of class 3 has a very small operating range, it only covers 1 meter; this limitation of range makes the class 3 very unpopular and rare in the market, so class 2 and 1 are the most often purchased by designers that decide to transfer data via Bluetooth.

After selecting a range for a specific project, it is indispensable to use Bluetooth devices that have the same range. For example, and IPhone is a device that has a class 2 Bluetooth unit, so in order to successfully connect it to an external Bluetooth module, a class 2 unit needs to be chosen. They both provide the same range (10 meters) and maximum power (2.5 mW), and this will generate the desired output. When the motorcycle rider is on the road, most likely will carry the cell phone use to sync with the helmet on his or her backpack or pockets, so the data will only need to be transferred across a short distance; in fact, in case a class 2 Bluetooth module is chosen the range would not be an issue because 10 meters are enough to complete the tasks that are being implemented in this design. So far this protocol seems appealing to the project because most likely the project will be limited to Apple IPhone, and Android cellular phones. Both of these cell phone brands that are being used to sync with the project already have a Bluetooth module incorporated, and this will totally facilitate the syncing process.

Advantages:

- It has a very low power consumption compared to other wireless protocols.
- Does not require straight line of sight to transfer data.
- Little radio wave interference.
- Spread spectrum frequency hopping.
- Already incorporated in a wide range of devices.
- Many robust profiles.
- It has the ability to penetrate surfaces.
- Already incorporated on the cell phone used in the project.

- Bluetooth uses a standard frequency, which allows all the Bluetooth devices to be compatible with each other.
- “Piconets” are easily configurable. A Piconet is a network that is formed when to or more Bluetooth devices are connected to each other, one Piconet can have up to 8 devices.

Disadvantages:

- It covers a very low range, but this is not a issues in our case.
- It has relative slow data transmission speeds. The transmission speed is fixed based on the Bluetooth device that is being used. Almost all Android cell phones and IPhone’s have at least Bluetooth 2.0 + Enhance data rate (EDR). The nominal rate of EDR is about 3 Mbit/sec.
- It has low penetration qualities.
- On cluttered 2.4 GHz ISM band.
- The price of a module is not very cost effective compared to other protocols.

2.2.4.1 PAN1321i Bluetooth Module

Selecting the right module for Motorcycle Tracking Security System (MTS) will facilitate our integration and will make the performance more effective. The PAN1321i made by Panasonic is the first module that is going to be analyzed. This is a Bluetooth RF module compatible with Apple devices such as the IPod, IPhone, and IPad. This is possible because the module interfaces with the Apple authentication coprocessor and supports IPod Accessory Protocol (IAP) to enable Bluetooth Serial Port data communication with Bluetooth enabled Apple devices. As we mentioned before, we are mainly planning to sync our project with and IPhone 5, so this will be perfect. At the same time, it is very important to notice that the module is also compatible with Android, and most Bluetooth enabled devices currently available.

The PAN1321i is used on Smart phones, for proximity, in Heart Rate applications, Generic I/O, HVAC, battery monitor, and in most wireless applications. It also has a reasonable price or around $27.00, which is acceptable for our budget. Figure 5 illustrates this module.

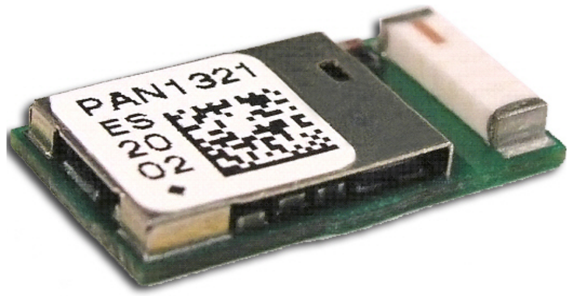


Figure 5: PAN1321i

Printed with permission of Open Source

The figure above displays how the module looks like, but it does not really show its actual size. The unit is small, to be more precise its dimensions are 15.6 x 8.7 x 2.8 mm^3, which is a good size for our project. We now know that is compatible with devices that we want to connect to our senior design project, and we also know that its size is acceptable; but we need to know some more about the Bluetooth module specifications. This information is summarized on the table 5, shown below.

Parameter	**Value**	**Condition / Notes**
Receiver Sensitivity (BER = 10^-3)	-86 dBm	Ideal signal
Output Power	+ 3 dBm	@ 50 Ohm antenna pin
Power Supply	2.7 - 3.6 V	Single operation Voltage
Ultra Low Power Scan	80 uA	T=25 degrees C
ACL (Transmit 3-DH1)	40 mA	Enhanced Data Rate, 531.2 kb/s
ACL (Receive 3-DH1)	37 mA	Enhanced Data Rate, 531.2 kb/s
Operating Temperature Range	-40 to 85 degrees C	

Table 5: PAN1321i specifications

2.2.4.2 WT32 Bluetooth Audio Module

The main purpose of our Bluetooth module will be to transfer Audio signal between the user's cell phone and the helmet, and vice versa. WT32 is a *Bluetooth* 2.1 + EDR module targeted for *Bluetooth* audio applications. In addition to *Bluetooth* radio, antenna and iWRAP *Bluetooth* stack, WT32 contains a DSP processor, a stereo audio codec and a battery charger making it ideal for portable battery *Bluetooth* stereo or hands-free audio applications. The figure 6 shows the outside of the module.

Figure 6: Bluegiga WT32 Printed with permission of Open Source

This is a class 2 module, so according to our previous research it has the basic requirements that we need to accomplish our goals, as far as, wireless communications is concerned. Here is a list of its main features:

- Plug and play Bluetooth solutions for mono and audio stereo solutions.
- Integrated Digital Signal Processor (DSP), stereo codec and battery charger.
- Integrated antenna.
- Class 2 module.
- Industrial temperature range from -40 C to 85 C.
- Low Power Consumption 1.8 V operation, 1.8 V to 3.6 V I/O.
- IWrap Firmware for controlling connections and configuring settings.
- Ten software compatible IO pins.
- Dimensions: 15.9 x 23.9 x 2.5 mm.

2.2.4.3 RN–42 Bluetooth Module

The previous two modules seem to be good options for our motorcycle and helmet system. But, we also want to take a look at the RN-42 module because it has some good features for a very reasonable price; it can be purchased for as low as $15.95, and it also a class 2 module that is compatible with the IPhone and the Android devices. We also like the fact that it Supports multiple Bluetooth profiles such as SPP and HID and simple UART hardware interface, and it is simple to integrate into an embedded system or simply connect to an existing device, which is our case, we. The RN-42, illustrated in Figure 7, is FCC and Bluetooth SIG certified making it a complete embedded Bluetooth solution.

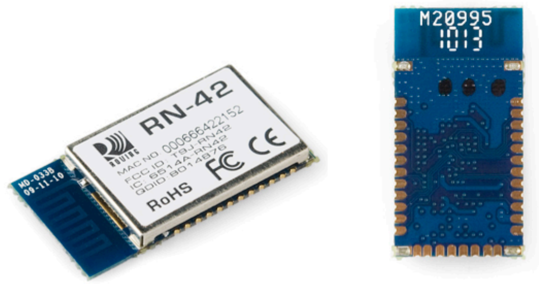


Figure 7: RN-42 Module - Printed with permission of Open Source

Here is a list of the main features that make this device a good option for our project:

- It is a fully certified Class 2 Bluetooth 2.1 + EDR module.
- Onboard embedded Bluetooth stack (no host processor required).
- Supports Bluetooth data link to IPhone, IPad, and IPod touch.
- It offers a secure communication, 128-bit encryption.
- Castellated SMT pads for easy and reliable PCB mounting.
- Low power sleep mode.

2.2.4.4 Helmet Wireless Communication Module

Now that our group has a better understanding of what wireless communication is all about; it is time for us to make a list of the requirements that we need to meet for our project. Let's start with the range of coverage; most of the time the driver of the motorcycle will be riding while using the different features of the helmet, but we also want the user to be able to connect with the electronic devices and have a good reception, even when they are a little far. Usually the cell phone is located inside of the rider's pocket or in a backpack so the distance from the cell phones to the Bluetooth module is not that much. We are not sure yet, if the module will be located underneath the driver's seat, or it will be inside of the helmet; but one thing is for sure, we need to select a component that will be able to cover a range between 0 to 10 meters, the distance between the cell phone and the unit will be less than a meter when the person is riding, so this will allow reception if the biker is walking on an area close to the bike, or in case the user falls off the bike, the module will still have signal, and it will be able to send a text message to an emergency contact to report an accident.

The next aspect to analyze will be the data rate. We want to make a product that will be able to transfer data at least a rate of 2 Mbps, but if we can get one that has a better rate it would not be a problem. To be sincere, we will try to get the module that has the highest data rate, because this will allow us to consume less power. The power consumption can be reduced if we transfer our data really fast because this will require less time to send the data for the communication. The only thing that will limit the data rate of the module that we will select, will be the compatibility with the cell phone and components that we are going to use with Motorcycle Tracking Security System (MTS), as well as the cost; we are trying to keep our budget kind of low, so we will not get a unit that is very expensive, even if its performance is great.

Finally, the last key factor that we want to take into consideration is the power consumption. We are trying to maintain a low power usage because this will allow the battery unit to last a longer time, while being as effective as it was when it was first charged. Looking back at the wireless protocols that we analyzed, we can totally tell that Wi-Fi will not be an option for our project because of its high consumption rate, while Infrared and Bluetooth do agree with our goal. We are trying to limit our power consumption rate to a maximum of 5 mW (8 dBm), and we will try to select the wireless communication module that requires the less power while being efficient. As well as the data rate, there is only one thing that will limit our range of selections, and that will be the price, we want to choose something effective and efficient, but it has to have a reasonable price too.

After doing the research about some of the different wireless communication technologies available in the market right now, as well as establishing some basic requirements to choose a module; our group decided to select Bluetooth as the protocol that is going to use in the helmet and motorcycle system to transfer data wirelessly. It is important to remember though, that we will not completely forget about the other technologies that were previously taken into consideration because they might be used as a second option.

Let go back and analyzed Wi-Fi a little bit, we found out that it is a technology that offers a very fast transfer rate; actually, it is almost like using a wired connection, which is something that really caught our attention; unfortunately, we had one big issue with it, and that is the fact that it requires a high power consumption, and that is something that we cannot afford to have in this project; therefore, we had to let it go. After that we took a look at Infrared, and we were really impressed with the amount of information that there is about this technology, it is very solid and well known. We really liked the fact that is really good as far as power consumption, but we were really disappointed when we found out that the signal transmitted needed a straight line of sight to operate properly; and not only that, the signal can be interrupted or lose very easy. Even if the two devices that are connected are misplaced, the signal will be lost because they need to be in a specific angle to have a clear and successful transmission. This will be a big issue for our project because the motorcycle will be in motion and there will be a lot of interference. Also, if we chose this technology the helmet with be limited to only syncing two devices, and we want it to be able to sync to a few devices. As we previously stated, we decided to select Bluetooth because is the wireless communication technology that seems to be more well-rounded; it is not perfect, but according to the information obtained, it does match with the requirements that we are trying to meet, and it is also very popular, so we will be able to choose from a variety of products and classes, and we will have a lot of good information online about the building and integration of the unit in our own system.

To conclude with this part we want to reiterate that the decision was made after taking different things into consideration such us: range of coverage, security levels, data rate transfer, reliability, efficiency, power consumption, availability, compatibility with electronics devices being used to synch with the module, and the ability to send the signal through objects. The research really helped us because it incremented our knowledge about this topic and it helped us pick the wireless communication technology that seemed to be more appropriate for our senior design project. We still need to look at the different Bluetooth Modules available in the market, to find out which one will be the more adequate to use in our prototype; so far we do not know much about the Bluetooth products available, but we do know that we most likely have to get a class 2 Bluetooth module according to the characteristics analyzed before in the Bluetooth section. In the next section of the project, we will carefully examine a few modules that we think are a good option for our design.

2.3 Global Positioning System (GPS)

Being able to locate a specific place, or to provide a current location has become a very popular feature among the newest and highest technological apparatuses; and it is one of the features that we want to have in Motorcycle Tracking Security System (MTS). For our project, we need to know the location of the motorcycle rider because this will be a very important piece of information for the emergency contacts, in case of an accident. If the user is injured and it is not capable of making a phone call, we are planning on texting an emergency contact with the biker’s location. But what technology can help us do this? Well, this answer to this question can be summarized in three letters GPS. The technology known as GPS stands for Global Positioning System, and is a space-based satellite navigational system that allows land, sea, and airborne users to determine their exact location, velocity and time 24 hours a day, and it works under any weather conditions, anywhere on or near the Earth where there is an unobstructed line of site to four or more GPS satellites. In order to acquire the motorcycle location, we will need to add a GPS receiver to our project. This receiver will be integrated to Motorcycle Tracking Security System (MTS), and it will be next to the user at all the time, the primary duty of this receiver is to receive the data from the satellites, and then use it to solve the navigation equations shown below.

$$p_i = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} - bc, \; i = 1, 2, ..., n$$

In the equation x, y, and z are the component of the satellite position and the time sent are designated by the i values in all the three directions. In our case, the result of these equations is the location of the receiver module that we are going to place on Motorcycle Tracking Security System (MTS). We are not going to fully analyze the Navigation Equations, but reading about them, and seeing some mathematical exercises online helped us better understand how a GPS works. We are Engineering Students, and we based our knowledge in Math. Now that we know, how a GPS receiver works, and how it obtains the data to provide the location that we are going to use for our project; we want to analyze the different type of receivers that are available. The first type is the multiplexing receiver; this receiver uses time division multiplexing to gathers data by switching at a high frequency, and it can track multiple satellites at the same time; another characteristic of the multiplexing receiver is that it uses a single, dual or even multiple receiver units, which provides more options when it comes to the hardware design. The second type is the parallel receiver; it has dedicated hardware to receive satellite information that is needed to determine a specific location. The last type is the sequential receiver; which is not very popular anymore because of its not so good performance compared to the previous two types. The sequential receiver switches and obtains information from multiple satellites, but it does not work in the same way that the multiplexing receiver does. The sequential receiver gathers all its data from one satellite and then it moves to the next one to continue to collect data to provide location. Among the three types of GPS receivers; the parallel and the multiplexing seem to be the more reliable ones, we definitively do not one to implement sequential because we want to have an efficient project. To find the location of the rider, the GPS module will need to have access to at least 4 satellites at the same time, so the unit that has access to more satellites signals will be able to provide the more accurate data; this is a little bit easier for the multiplexing receiver because the parallel requires more hardware.

To select the right GPS module some other features need to be analyzed. These features are: size, update rate, power requirements, and number of channels, antennas, and accuracy. As far as the size, we need to pick a module that is relative small, but we do not want to get something extremely small because it will require a tiny antenna, which will affect lock time and accuracy. Our project requires at least an average update rate, because our motorcycle and helmet system is design to be on the road most of the time; the standard for most devices is 1 Hz (only one per second), and this is relative enough for our project, there are modules out there that provide a faster rate around 5 or 10 Hz, which is also good, the only problem is that they will require our microprocessor to process more data, the good thing about these faster modules is the fact that they give more accurate data and they can usually be configured to run at an easier pace. Let's take a look at the power consumption; on average a GPS module uses around 30 mA at 3.3 V, so we have to get something around that range, or if we can find something with a lower power consumption it would be phenomenal, but if the power consumption is very low, we have to make sure that the unit has an antenna attached to it. There are 35 systems of navigation satellites in space, and 24 of them are active at any given time; as we can see there are not that many, so the number of channels that the module runs will affect the time to first fix. The GPS receiver does not know which satellites are in view, so the more frequencies that you can check at once, the faster you will find a fix. On average, 12 to 14 channels will work fine for tracking, but if we want a faster lock, we will need a module with more channels. As far as the antennas, most of the modules come a chunk of ceramic on top, which is the antenna. Each antenna is finely trimmed to pick up the GPS L1 frequency of 1.57542 GHz; we will try to get a module that already has one incorporated to facilitate the integration. Finally, let talk about the accuracy, according to our research online, in average you can easily find the position of the module, any where in the world, within 30 seconds, down to + / - 10 m. But most module can get it down to + / - 3 m, which is a perfect for our project. The plus or minus sign states that it can more or less vary according to the module, time of the day, clarity of reception, location of the module, etc.

Now that we have a better knowledge about the Global Positioning System technology, and the different types of receiver available in the market; we can confidently say that we can incorporate this technology into our project. As a matter of fact, we can implement a module that will be extremely fast and accurate; but our only limitation is the amount of money that we are willing to spend. That does not mean we are going to get a bad unit, we will do our best to select a unit that is reliable, effective, accurate, has a small size, a good update rate, good power requirements, and that of course is within our budget. In the following sections of our research we will look at few models of GPS receiver modules that seem to be appealing to our senior design project.

2.3.1 D2523T Helical GPS Receiver

Figure 8 shows the D2523T dimension. This is a compact GPS smart-antenna engine board, which comes equipped with a Sarantel GeoHelix high-gain active antenna and GPS receiver circuits. The module is based around the high performance 50-channel u-blox 5 platform, and its omni-directional antenna provides great sensitivity, even when you do not a have a clear view of the sky. This module will be really good for our project because it will provide a very high and accurate performance, while keeping low power consumption. The only bad thing about is its price. It cost about $79.99, and even though it is not the most expensive module, there are quite a few that can do the task for a lower price. The helical antenna, which is not the basic model for antennas, is one of the reasons that make this receiver a little bit more expensive.

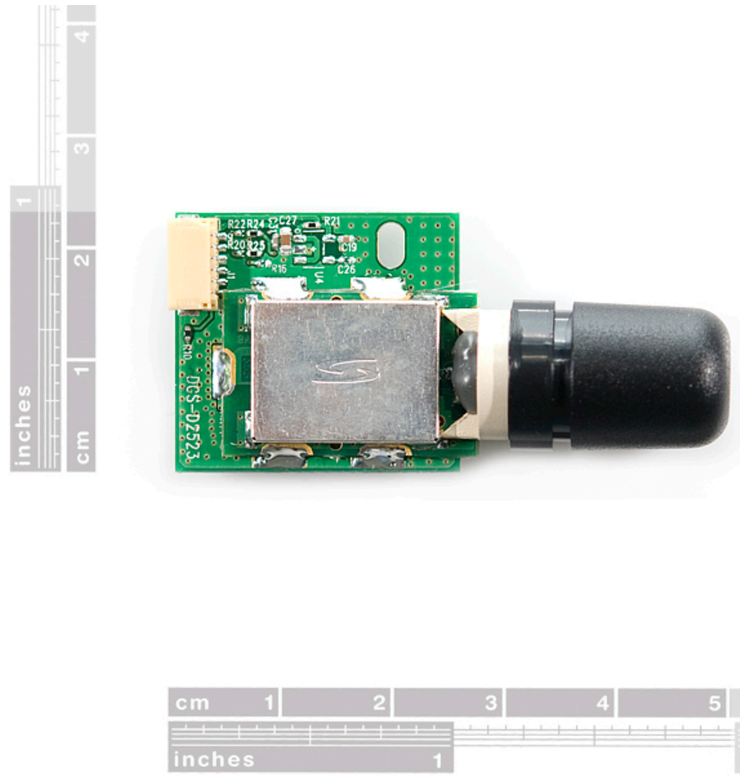


Figure 8: D2523T dimensions - Printed with permission of Open Source

The advantages of this module are:

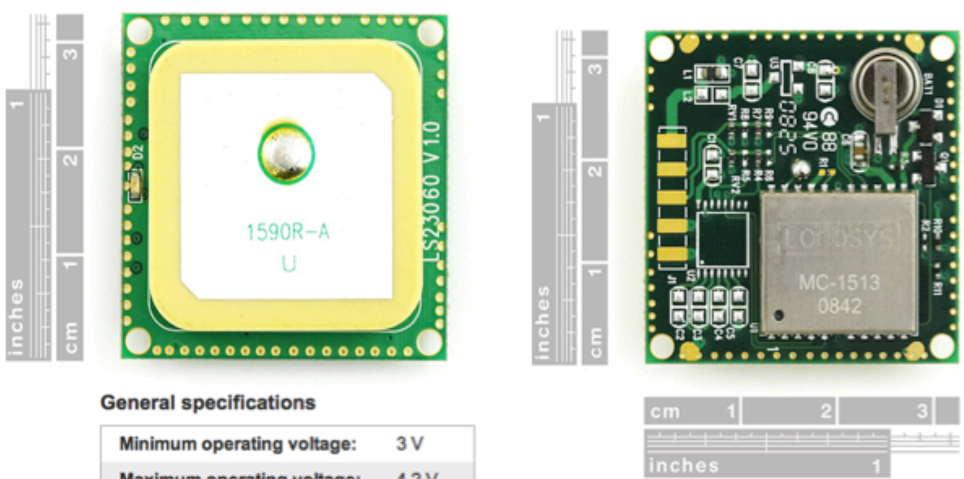
- 3.3 V supplied voltage for low power consumption.
- 50-channel u-box engine, which will provide a faster and more accurate position, for the biker.
- It has a high performance antenna already integrated in the module.
- It has a high immunity to jamming, which is something that will come handy in our project because the bike will be on the road.
- It offers an accelerated startup at weak signals.
- It usually offers a 1Hz update rate, but it can go up to 4 Hz, which is more than average.

Some of the disadvantages that we can encounter, if we pick this module for the motorcycle and helmet unit will be the following:

- The cost of the unit is more expensive than average GPS modules.
- It has an update rate that is faster than average, so it will require the microprocessor to process more information.
- It does not have a LED.

2.3.2 LS20031 GPS 5Hz Receiver

Our second option is the Locosys LS20031 GPS receiver; this is a complete GPS smart antenna receiver that includes an embedded antenna and GPS receiver circuits. This low-cost unit shown in Figure 9, outputs an astounding amount of position information 5 times a second. The receiver is based on the proven technology found in LOCOSYS 66 channel GPS SMD type receivers that use MediaTek chip solution.



General specifications

Minimum operating voltage:	3 V
Maximum operating voltage:	4.2 V
Supply current:	29 mA[1]

Figure 9. LS20031 with its dimensions and its main features

Printed with permission of Open Source

Advantages that made us consider this module:

- It has a LED indicator for fix or no fix.
- Its pads are easy to solder compared to other models.
- Average power consumption 3.3 V.
- Supports 66 channels for faster performance.
- It can go up to 10 Hz update rate, but it can also be configured to lower its update rate.
- Built in micro battery to preserve system data for rapid satellite acquisition.
- Reasonable price for its features $49.99.

On the other hand, some disadvantages are:

- It requires voltage regulation if the module is implemented in a system that run on a different voltage.

- It can transfer up to 57600 bps, which is a lot of data for the microprocessor to handle in a short period of time, but this can be modified.

2.3.3 Venus638FLPx-L GPS Receiver

The Venus638FLPx is a model that really caught or attention due to its size. It is not as powerful as the receivers previously reviewed, but it still is a high performance, low cost, single chip GPS receiver. It targets mobile consumers and cellular handset applications due to its reduced size, and it also offers very low power consumption, high sensitivity, and best in class signal acquisition and time-to-first-fix performance. Venus638FLPx contains all the necessary components of a complete GPS receiver, includes 1.2dB cascaded system NF RF front-end, GPS baseband signal processor, 0.5ppm TCXO, 32.768kHz RTC crystal, RTC LDO regulator, and passive components. It requires very low external component count and takes up only 100mm2 PCB footprint.

Figure 10, which is located below has a complete list of all the features that the receiver has to offer, and here are a few advantages that we might think can make this module a good option for us:

- It is a very small object, its dimensions are 10 x 10 x 1.3 mm, and it only takes up to 100 mm^2. Figure 5 can easily show how small it is compared to a coin.
- It offers jamming detection.
- It is compatible with active or passive antennas.
- Low price ($39.95)

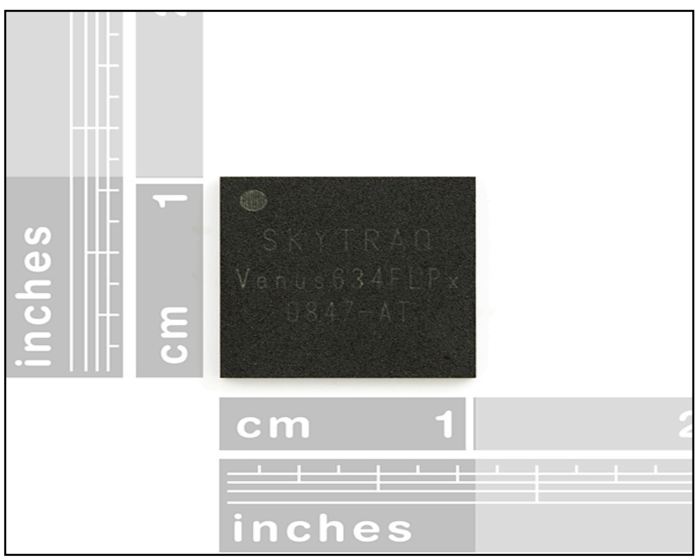


Figure 10. Venus638FLPx dimensions

Printed with permission of Open Source

As all the previous devices, the Venus638FLPx also has a few characteristics that can make it not so ideal to use in our senior design project. The disadvantages that we found are:

- A slower update rate compared to previous receivers.
- It does not have an antenna incorporated.
- It does not have an LED indicator for fix or no fix.

2.3.4 GP-2106 SiRF IV GPS Receiver

There are quite a few GPS receiver modules available in the market right now, but the GP-2106 is the last one that we are going to take into consideration. This module comes with a built-in antenna and is also relative small compared to other designs; its dimensions are 21 x 6 x 6.2 mm. The SiRF Star IV powers the receiver, and the module can acquire satellites as low as -163dBm. It also has a hibernate mode which can go as low as 30uA while maintaining a hot start. It is important to know that the module runs at 1.8V VCC, however, you can use 3.3V TTL on the TX and RX pins. Also, the ON/OFF pin needs to be toggled high (to 1.8V) or low in order to turn on the module.

This module is not the most powerful one, but it does get the job done, as far as, locating the motorcycle rider while he or she is on the road. Let's check out the main advantages that we found:

- It has a good number of channels.
- It offers an average update rate of 1 HZ, which will not transfer a lot of data at the same time.
- It has a built-in antenna.
- Cost $49.95, which is not that expensive.
- It's smaller than quite a number of devices that offer similar characteristics. Figure 11 shows how this specific receiver looks, and some of its other features:

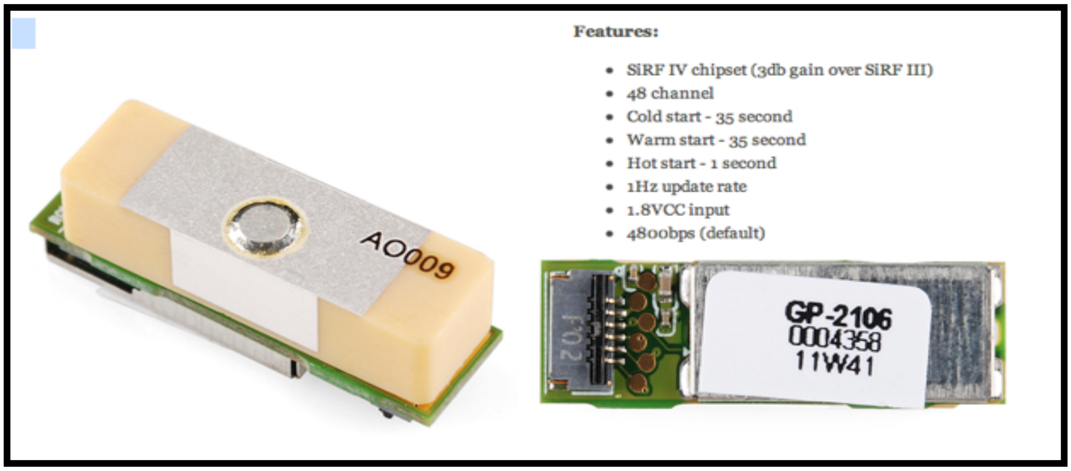


Figure 11: GP-2106 with its main features

Printed with permission of Open Source

This is a well-rounded receiver for its price, and it offers a good coverage, unfortunately, it also has some disadvantages such as:

- It does not have a LED.
- It has a limited update rate, so it will limit the speed of our design.
- It requires voltage regulation if the module is implemented in a system that run on a different voltage.

2.4 Accelerometer

An accelerometer is a special sensor that can detect accelerations and decelerations in any axis, depending of course on what that certain accelerometer is capable of. An accelerometer in this project will be given two main functions, first to act as a motion sensor, and second to detect possible accidents. In detecting accidents two accelerometers will be used to detect difference in accelerations. In detecting the difference between the two, the module will be able to know if the rider has been flown off the bike or not. Both accelerometers will be implemented for both crash and motion detection.

Since an accelerometer picks up movement in different directions it would be a good choice to use as a motion sensor or a tilt sensor. Its requirements for a tilt sensor is to pick up minimal movement to the motorcycle or helmet while in the locked position, meaning with the motorcycle off and alarm system on. The direction does not really matter for this function only movement detection in any axis. It is a safety feature that is going to be incorporated to the motorcycle alarm system. If anyone attempts to steal the motorcycle the moment it is moved the alarm system will be triggered and the owner will be notified via SMS and it can be tracked down with GPS. The same thing will happen if the helmet is moved.

The second function of the accelerometer that is going to be applied is crash detection. Since one of the abilities of an accelerometer is to pick up decelerations, that can be used to detect any sudden high decelerations that are indicative of an accident. This will be part of the safety system that will help the motorcyclist get a quicker response for help during an emergency, since motorcycle accidents can be deadly. Most of the time motorcyclists are by themselves and if by some chance they get in an accident with no one around how will they call for help? With this feature they can feel more at ease about that possibility since the safety system can call for them. The accelerometer will have a certain threshold that once the amount of deceleration or g-force passes it, it will trigger the safety system, which will then send a SMS with coordinates of the crash location. The accelerometer will also be able to detect if the motorcycle is on its side and if the rider was flown from the bike. With all this information will help to understand how grave of an accident was experienced. That way the motorcyclist will get immediate help. The parameters will need to be chosen carefully for the system to work properly and accurately.

To be as accurate as possible using a three-axis accelerometer would be the best choice. Since there are many factors to be accounted using a two axis accelerometer would be a little inaccurate. For starters the way the module would be position in the motorcycle will not be exactly on two axis, it might already have some tilt to it. Because of that we are not going to be able to know exactly in which axis the movement will be on. The motorcycle itself might be modified different then when how it came from the factory. So to cover all different possibilities and still work like it’s suppose the accelerometer would have to be a three axis sensor. The same applies to the crash detection function. Having a three axis will help be more accurate in calculating the g-forces in an accident, which is very important for the safety of the rider.

When the accelerometer is acting as the motion sensor of the motorcycle system, sensitivity is very important and it needs to be high. It needs to be able to pick up the slightest movements for the security system to work properly. If a motorcycle were to be stolen the most likely thing a thief would do is load it up in a truck and carry it away. To have the alarm system respond to this as soon as possible the accelerometer needs to be sensitive enough to pick up any small change of movement of the motorcycle no matter on which axis. Now as far as when the accelerometer is being used as a detection device, sensitivity is not very important and it would need to be low. When dealing with an accident we are dealing with high forces of deceleration, g-forces, thus the input received by the accelerometer will be high. To be able to attain an appropriate output within the range desired, an accelerometer with low sensitivity would be best. Sensitivity is the measure of how much an input signal is amplified, so depending how much we want to amplify the received input signal to the accelerometer that will determine the sensitivity. This then makes sense why we need high sensitivity sensor for motion detection and low sensitivity sensor for crash detection.

Another important aspect of the accelerometer that will also be crucial in its accuracy is its bandwidth. Bandwidth gives you the range of the inputs that it can accept. Depending on what application it is going to be used knowing the type of inputs it will get will help in choosing the bandwidth requirement for the accelerometer. In our project we will be dealing with two different types of applications. The motorcycle will have motion sensing to be able to detect any unauthorized tampering and be able to trigger a response. If someone were to try to steal the motorcycle they would need to move just slightly. Because the bike itself will be slightly tilted then the accelerometer would need to pick up very small movements. Having the right Bandwidth range will help in picking up such movements. Generally a small range of forty to fifty hertz will be good enough to pick up those small movements. As far as using the accelerometer for crash detection it is quite a different story. In crash detection the type of inputs that the accelerometer will be receiving will be much higher than in motion. An accelerometer of several hundred hertz will be required to properly read the inputs and produce accurate outputs.

For the system to perform well and not cause problems for the motorcycle we have to make sure the module we design does not pull out enough current from the battery that will prevent the rider from starting the motorcycle. To be able to ensure that the module will not put too much strain on the battery we have to choose components of low power usage. This is not hard to find in accelerometers nowadays. On a motorcycle battery there are already components that are drawing current from it at all times. The draw of current has to be small enough that it could take weeks of no usage for it to have any effect on the battery. A typical motorcycle can have as much as 10 mA of draw when it’s off. The highest tolerable readings can be anywhere between 20 to 30 mA before it starts putting too much strain on the battery. Accelerometers are low power devices and the current needed for it when the motorcycle is off is much less than the typical draw. This makes power consumption the easiest parameter to meet.

The communication of the accelerometer to the rest of the components will depend on whether it is digital or analog. Those are the two forms that the information received can be transmitted. Analogs are constant readings. They can measure changes at a faster rate than digital can handle. It would excel over digital in cases where it is reading vibrations, or seismic activities. Basically in conditions that the inputs received by the accelerometer are dynamic at all times. It is basically sending a signal at all times of the input that it is receiving where as digital will send pulse width signals upon receiving an input. Depending on the acceleration will dictate how long the pulse width will be. For the task the accelerometer will be doing, digital would be the best choice. There will be no continuous input signals to be interpreted. If it’s either for motion sensing and crash detecting the signals will be short length in time making analog outputs not very efficient for this task.

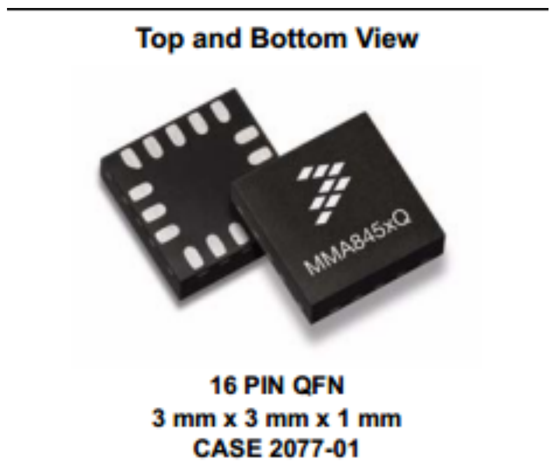


Figure 12: Digital Accelerometer

(Permission in progress)

The accelerometer that we pick is going to be integrated in the circuit and for it to be able to function it must be able to communicate properly with the microcontroller. In other words the microcontroller and accelerometer need to talk the same language so there can be communication. In communication of sensors there are two main protocols that are used, Serial Peripheral Interface (SPI) and Inter Integrated Circuit (I2C).

SPI protocol is a simpler form of communication than I2C. It uses three lines for data communication, which are Master In Slave Out (MISO), Master Out Slave In (MOSI), and a clock line. They also have a fourth line needed to be able to commence communication from the device. A Chip Select (CS) signal is needed to enable transfer of data. The clock line carries the clock pulses that provide synchronization of the data. The other two lines are the actual lines that transfer data out and in. The SPI 4-wire protocol support full duplex communication with faster data transfers than the I2C. In a full duplex communication system, the device must have data coming in for it to send data out. One method to get around that is to send dummy bytes. Using SPI 3-wire protocol gets around that problem as well. The 3-wire version uses three lines by combining the MISO and MOSI into one line for data transfer. Instead of two it uses one called Single Input Single Output (SISO). SISO is a single bidirectional data line that carries data in both directions. To be able to properly use the 3-wire version of SPI the device it is communicating to must support that version. If it supports only 4-wire version a method called bit banging can be used to be able to still use the 3-wire version.

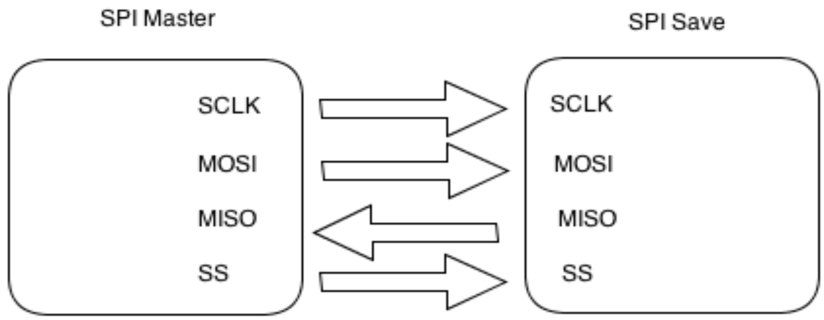


Figure 13: SPI communication system

The I2C protocol consists of two bidirectional data lines that are pulled up to the voltage supply line usually called Vdd. The two lines are the Serial Data line (SDA) and the Serial Clock line (SCL). One of the big benefits over SPI it can communicate with many more devices. To start the communication between devices the master sends pulses of data with a START bit at the beginning which indicates the beginning of the data. Then it is followed by either a WRITE bit or READ bit to know which action in needs to take. After that it has the actual data it wants to transmit, and the transfer stops after a STOP bit is read. That START and STOP commands are just a transition from high to low on the SDA line with SCL high and low to high on the SDA line with SCL high.

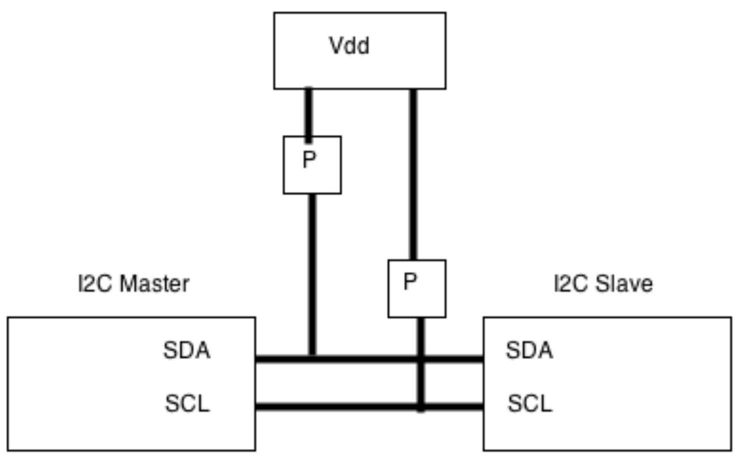


Figure 14: I2C communication system

By researching all the different specifications needed for our design to work as intended it would seem that having two different accelerometers over one could be more efficient and accurate. To have just one accelerometer would be great since it is less components to deal with but since the parameters for the design are so wide it would be very hard to find one accelerometer that can do all the tasks accurately. To better focus we can split the tasks into two, one just for motion sensing and the other just for crash detection. We will be better able to meet the parameters and to keep accuracy and performance as high as possible meeting the requirements of the design.

Another possibility to picking an accelerometer could be picking one that comes with adjustable sensitivity and bandwidth. From the research taken to find the right component, we discovered that nowadays we can purchase an accelerometer with different settings of adjustability while keeping cost still close enough to that of a regular one. This would be the best solution, it keeps the components being used to a minimum thus also making it easier to make our module as small as possible. It would greatly help in achieving the goals of the project design.

We have come up with a possible three different accelerometers that could work with our project. The table below shows the different specs of each of the accelerometer compared to each other. The first one is the ADXL345 by Analog Devices, then the ADXL362 also by Analog Devices, and MMA8452Q by Free scale Semiconductors. Each of them has the potential to meet the requirement we have for our design.

Name	**ADXL345**	**ADXL362**	**MMA8452Q**
G-Scale and sensitivity	±2g, 3.9 mg/LSB ±4g, 7.8 mg/LSB ±8g, 15.6 mg/LSB ±16g, 31.2 mg/LSB	±2g, 1 mg/LSB ±4g, 2 mg/LSB ±8g, 4 mg/LSB	±2g, 1 mg/LSB ±4g, 2 mg/LSB ±8g, 3.9 mg/LSB
Output Data Rate	6.25-3200 Hz	12.5-400 Hz	1.56-800 Hz
Resolution	10-13 bits	12 bits	8-12 bits
Interface	SPI or I2C	SPI	I2C
Interrupt pins	2	2	2
Price	$27.95	$14.95	$9.95

Table 6: Accelerometer Specification Comparison

2.5 Global System of Mobile Communications (GSM)

As the project implements Global Positioning System technology, the realization of a Global System of Mobile Communications (GSM) could be a good idea because it will allow us to have access to a mobile communications network. GSM is a digital cellular phone technology based on time division multiple access (TDMA); GSM defines the entire cellular system, not just the TDMA air interface. Currently GSM utilizes a variation of TDMA, and is the most popular among the three digital wireless telephony technologies, which are: TDMA, GPS and Code division multiple access (CDMA). The basic concepts of a GSM are the followings: At first, the GSM unit digitalizes and compresses data; then it sends this data down a channel with two other streams of user data, and each one has its own time slot. Finally, it is important to mention that the GSM operates in a number of different carrier frequency ranges (separated into GSM frequency ranges for 2G and UMTS frequency bands for 3G). Most of 2G GSM networks operate in the 900 MHz or 1800 MHz bands, and most of the 3G networks operate around the 2100 MHz frequency band.

Motorcycle Tracking Security System (MTS) will need to have a GSM module, in order to have access to a network that will allow it to send text messages. Pretty much, our project will send two types of messages: the first one will be a text to the owner of the bike in case the motorcycle is stolen; and the second one will be a text to an emergency contact, informing that an accident occurred, and it will provide the biker’s location. There are quite a few networks available, from Edge to 4G LTE; but we do not actually need a super fast network such as 4G LTE, because we are only sending messages. As we mention before 3G operates around the 2100 MHz frequency band, and it is also well-received around the market, so our group has decided to get a module that implements this network. In order to send text messages the GSM module will need to be connected to a SMS module or have one already integrated. SMS stands for Short Message System; and it is a text messaging service component of a phone, web, or mobile communications systems. Originally the SMS technology was only used on cell phones that had GSM technology incorporated, but they have become so popular, that now they are supported by all the major cell phone systems.

Let us talk a little bit, about the services that the SMS module will allows us to have. This technology will allow us to send messages that can be up to 160 characters in length, but some use a 5 bit mode that allows sending up to 224 characters in one message; we are only planning on texting the motorcycle location (longitude and latitude), and a small note that says, “your bike has been stolen” or another one that says “an accident occurred”, plus the location. This technology will be more than enough because we can totally do this in 160 characters. Now that we selected GSM and SMS as the technologies that we are going to use, and we also know more about how they work and the basic specifications that we need. We are going to analyze a few modules that seem to be appropriate for our Senior Design project.

2.5.1 GSM/GPRS Module - SM5100B

The SM5100B is the first module that caught our attention. It is a miniature, quad-band (frequency bands) GSM850 / EGSM 900 / DCS 1800 / PCS 1900 module that due its size and basic features can be compatible with our project. Our goal is to place this module on the PCB that is going to be located in the motorcycle system that will be located underneath the bike's seat; therefore, we are trying to save as much space as possible. Figure 15 shows the size of the SM5100 B module.

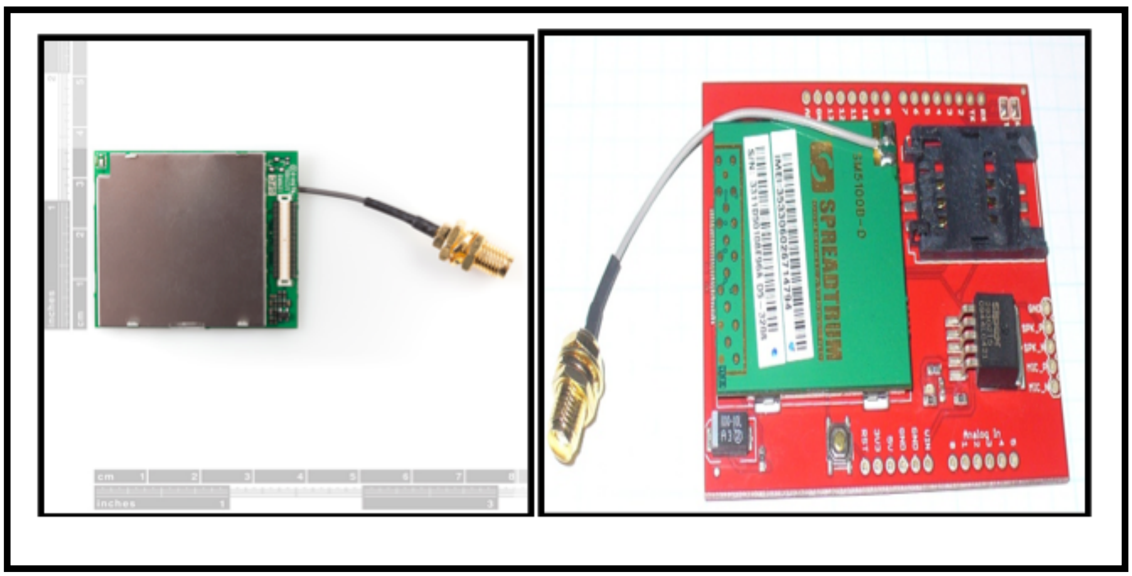


Figure 15: SM5100B dimensions & Arduino Cellular Shield with the module incorporated.

Printed with permission of Open Source

As it can be seen in the figure above the dimension of the module are quite small, so it fits our requirements for the PCB integration. Let us take a look at some other advantages that make this product appealing to us:

- It is a miniature module; its dimensions are: 35.0 x 39.0 x 2.9 mm. Also shown in Figure 7.
- No need of additional module to send SMS, this module is capable to provide almost all the basic features that a regular cell phone offers.
- As far as SMS technology. The module supports MO and MT for SMS. It also supports point-to-point short message broadcast. And it Supports TEXT and PDU modes as well.

- As far as the protocol, it supports GSM/GPRS 2/2+.
- It has an antenna already attached to the module, which facilitates the integrations.

Some disadvantages are:

- Its power voltage goes from 3.3 V to 4.2 V (3.6 V is recommended), which is higher than the other components selected so far. Voltage regulator will be required.
- The shape of the antenna might require more space than what we want to use.
- It does not have a SIM card socket, so it would require a SIM card breakout board, and it cost around $14.95.
- Its base price is $59.99 making this module more expensive than the one that we are going to check out next

In case, we decide to go with this module, it would be a wise idea to use a Cellular Shield that includes all the parts needed to create an interface between the microprocessor and the cellular module. An example of this can be found on the previous Figure 7 that shows the dimension of the SM5200B, as well as an Arduino Cellular Shield with the Quad Band Module incorporated. The actual cellular shield illustrated in this figure has a price of $ 99.00 and it includes the GSM module already.

2.5.2 ADH8066 GSM Module

The second GSM module that we are going to analyze is the AMH8066. It offers similar characteristics to the SM5100B; it is also a miniature, with quad-band GSM 850 / EGSM 900 / DCS 1800 / PCS 1900 module, and due to its dimension it can be easily and effectively integrated to our project, refer to Figure 16 below for actual dimensions.

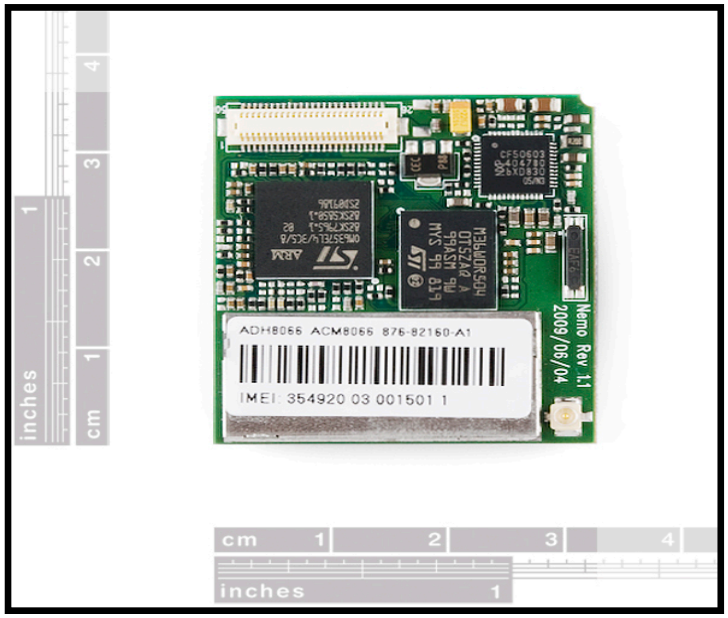


Figure 16: ADH8066 dimensions - Printed with permission of Open Source

Even though this module is not the most expensive or the most powerful in the market, it has more than enough power to send text messages, which is the main reason why we are incorporating this type of module into our Helmet and Motorcycle system. Here are some of the advantages that the ADH8066 GSM module has to offer:

- Provides a good number of features for a reasonable price of $ 49.95.
- A little smaller than the previous GSM module. Its dimensions are 33.0 mm x 36.0 mm x 5.4 mm with the SIM card holder.
- Includes a SIM card holder, which facilitates the integration process.
- Standard SIM interface (3V / 1.8V).
- Protocol supports GSM/GPRS Phase2/2+.

The two things that we don't like about this module are the following:

- Its power voltage goes from 3.4 V to 4.5 V (4.0 V is recommended), which is higher than the other components selected so far. Voltage regulator will be required.
- It does not have an antenna incorporated.

2.6 Battery

What type of battery we use for our design is going to be a crucial part for both the motorcycle and helmet subsystem. There are many different types of batteries in today's market to choose from and we have to pick one that their capability will meet the parameters we have set for our design. Since in this generation we are exposed to so much technology we are already familiar more or less with what type are used to power up small devices. The main two types of batteries that are going to be considered for our design because of their many advantages and capabilities are lithium polymer and lithium ion. These two are the most commonly used for its size, and ability to recharge well. They are used in many application that we use nowadays, the most popular one are the batteries for hand held devices. Size, cost, and performance are some of the main points that we will focus on for our design. The battery we use needs to be light in weight and as small as possible still meeting all the other requirements, and to be able to hold its charge for the time required.

For this design we are going to implement two batteries, one for the helmet to be able to operate and for the motorcycle as an emergency backup. In the helmet subsystem the battery is going to provide power to the Bluetooth module and the radio frequency module during operation. Its task is while the rider is using the helmet he needs to be able to use the Bluetooth for outgoing and ingoing calls, GPS, or music purposes. The amount of charge the battery can provide has to be able to last longer than at least one entire days use. Which it is highly unlike someone will ride a motorcycle for one whole day but that gives a good over calculation taken into account if they forget to charge it or other unforeseen possibilities. It is very important that the rider can be able to use its Bluetooth assistance in case of an accident. If an accident were to occur an emergency response team can call ahead to check for consciousness of the rider. The rider being conscious can answer the call without moving to let help know of the severity of the accident, and if not then they know they are dealing with a severe situation. Then for proper usage the next day it would need to be charged during night time or times of inactivity. For the motorcycle subsystem its purpose will be to provide a backup power source just in case that the main power source is cut off or disconnected. This would most likely happen if the motorcycle is stolen and the main battery could be taken out or just disconnect, having a backup battery will give police more time to track it down if needed.

First we are going to be looking at the lithium polymer batteries. They have many good advantages. One of them being that they offer a very low profile which is small enough to resemble credit cards or even smaller if the demand is high enough. Manufacturers are not bound by any standard cell format. The polymers can be design to almost any shape and size. One advantage they have over lithium ion is that they use a gelled electrolyte over liquid. By having these designs, manufacturers do not need to put a hard metal enclosure and instead use heat-sealed foil. This allows a space to be utilized a lot more efficiently making them much smaller than the lithium ion. Another benefit to its improved design is that it is less likely to leak electrolyte, which makes it a much safer alternative. Li-ion has about the same benefits that Li-Po offer, except that Li-ion has higher energy density over Li-Po. They have a lower cost mainly because of higher supply and demand for Li-ion.

So after looking at its design benefits we can focus on its actual capabilities. A single cell is rated at 3.7 volts, which is the average rating for the battery. It is not going to operate at 3.7 volts its entire time of operation. When the battery is fully charged it will operate at around 4.23 volts and when it's discharged it will be around 2.7 volts. That means that whether the battery is at 4.23V or 2.7V all of our components that are running off of the battery still need to operate properly and accurately. Their voltage capacities can vary depending on manufacturers' specs. If the input voltage needed to operate is greater than 2.7V then maybe a double cell battery will be more adequate. Lithium cells are set up in series and like stated before each cell rated at 3.7V and a Lithium pack can come with multiple cells depending of course on the parameters needed.

Li-Ion can provide with a voltage of 3.7V, knowing how much current our design need will help us pick the right capacity. Current is what dictates how much power is leaving the battery to supply the circuit. Our components are going to be low power for this main reason, that we want our battery to last as long as possible. Capacity is measured in milliampere hours (mAh). This rating lets us know how long the battery will last at a certain load. For example a battery rated at 1000 mAh with a load of 250 mA will only last four hours. So knowing that, the capacity for our design needs to be high enough to handle the load of our circuit for a full day. The discharge rate is related to the capacity but for our purposes it is not very important. All it tells us is the maximum current load the battery can handle, and since our circuitry deals with few low power components that won't be an issue.

The recharging process of Li-Ion battery can be more intricate than others. To start, the charger going to be used must stop charging at 4.2V and not going anything beyond. The Li-Ion cannot handle being charged at a voltage that is more than its rating. If the charge goes beyond that voltage it could catch on fire or worse explode. Constant Current/Constant Voltage, CC/CV, method must be implemented to the charger or circuit connected to the battery to properly and safely charge. The way the method works, at first the battery is charged with a constant current until it reaches a voltage of 4.2V. Then it must switch from constant current to constant voltage to keep at a constant 4.2V. The charger must gradually reduce the current until it gets to 2%-3% of the original constant current that it was providing. Charging a multi-cell Li-Ion requires an extra procedure of monitoring each cell individually. No two batteries are ever the same meaning that they will have a different charge rate that must be monitored to not pass their charging capacity. Li-Po packs come with balancer plugs for the purpose of monitoring each cell to make sure that their voltage level be kept the same. The figure below shows the graph of the charging time of the Li-Ion battery.

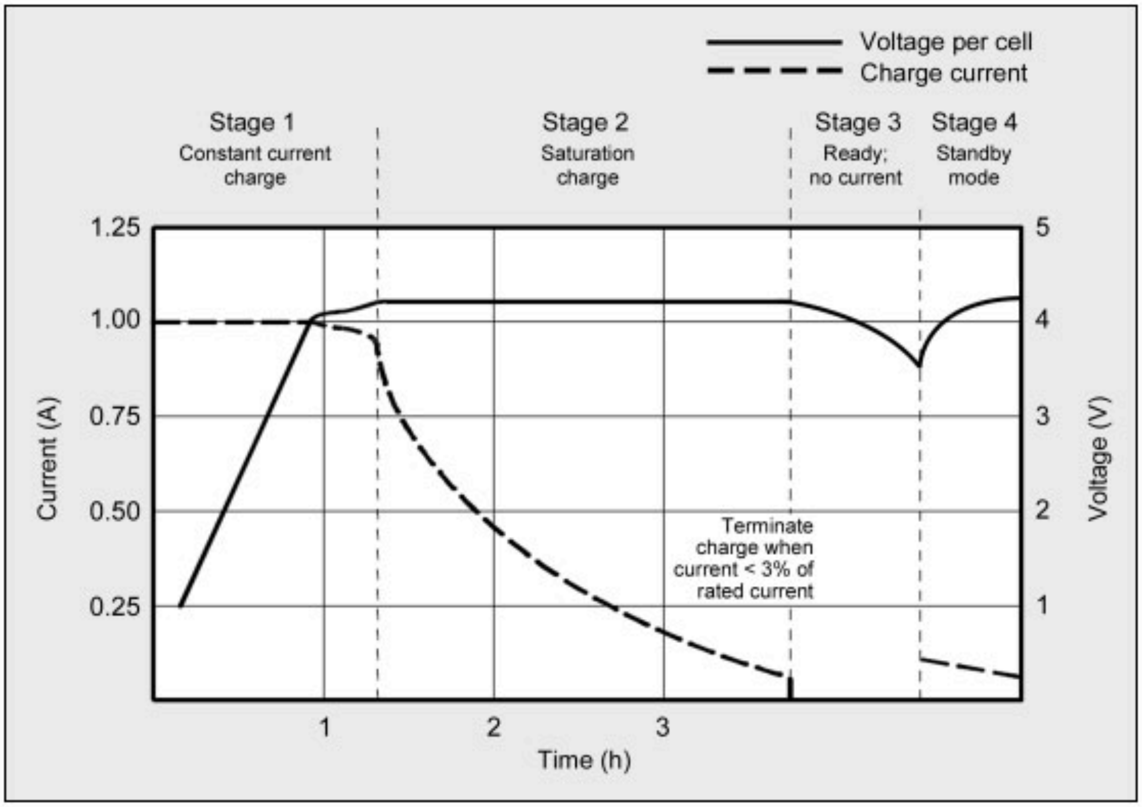


Figure 17: Charge Conditions

(Waiting for permission)

Nowadays integrated circuit chips are manufactured just for the purpose of properly charging any kind of Li-Ion and Li-Po batteries. Most chips take either a DC voltage power supply or USB power supply and regulate it according to the CC/CV method to charge the lithium battery. This way any kind of overcharging and potential hazardous accidents are prevented. The specs of the chip have to be looked at carefully to make sure it will work well with the battery chosen. The rate of current at which it charges has to be below the max rating of the battery and same goes for the voltage.

Looking back at both the advantages and disadvantages of the Li-Ion and Li-Po we see that they are very evenly matched as far as what we need them for. The Parameters that need to be met in our design can easily be met by either of the battery. But in one area that excels just a bit more than the other is in size. Size is a big issue in our design since this component is going to be put in a spot that has almost no space. The Li-Po excels more on the size department because of their unique design. That way we can get the same performance of a Li-Ion but smaller in size. The task the battery has to perform are minimal and don't place a heavy burden on it. The main points are to provide a simple power source that can last for several uses before recharging and keeping its size as small as possible.

2.7 Microcontroller

Though we've put copious amounts of research and conscious efforts in picking our parts to meet our design specifications and requirements, all would simply be futile without the direct control of a smart mind behind it all. Of course we cannot control these components with our own brains while we wear the helmet, not yet at least. However adding intelligence, control and seamless integration to our design is nothing out of this world. This pivotal device to our design integration is of course a microcontroller.

The function of this microcontroller is truly the heart of our design as it will be what will be controlling, sending and receiving signals between our various peripherals being used in this project. For our project we do not need massive amounts of processing or computational power therefore a low cost low power microcontroller will suffice for the specific tasks we are implementing here. Some basic things we are looking for when choosing our microcontroller are:

- Have enough input and output pins to support our peripherals.
- Be compatible with our communication modules, accelerometer alarm and FOB control.
- Have sufficient memory and processing power to support our various triggering interrupts.
- Be able to efficiently and seamlessly handle interrupts and exception handling.
- Low power and somewhat resistant to temperature and certain weather conditions.

The microcontroller will be responsible for handling every signal sent from our peripherals. For one, it will be directly connected to our accelerometer and will be handling the signals sent and will then be deciding whether or not conditions are normal or a special exception worthy of a software interrupt in which it will decide which action to take afterwards. For example, if our bike is parked and at a steady state and suddenly our accelerometer triggers a tilt in orientation, it will then proceed to triggering a theft exception in which the microcontroller will then trigger our loudspeaker alarm mechanism and or proceed to send an SMS to the owner notifying of the current status the bike is found to be in.

We also took into account that none of us were computer engineers nor experienced programmers therefore our microcontroller would have to be a programming friendly and intuitive one. We would prefer to program in C rather than assembly and we would rather use an intuitive language as opposed to a cryptic one that would just end up giving us a hard time when we could've accomplished the same task using the simpler microcontroller from the start.

For our project we will not need a super powerful microcontroller but rather a solid, reliable and practical one. Basically any major microcontroller our there could suit our needs as we do not need a massive amount of I/O pins, memory or processing power therefore we will focus our search in just finding the best fit microcontroller that will allow us to intuitively program it in an IDE environment and integrate it into a PCB board along with all our other peripherals.

Another design consideration would be which architecture to use to sufficiently fit our needs. More specifically we looked at how many bits would suffice to successfully implement our design. In the microcontroller world we are presented with very different varieties of controllers among these are the 8-bit, 16-bit 32-bit or even 64-bit architectures. For our design It is logical to pick an 8 or 16-bit architecture as it is sufficient yet not a complete waste of processing power. This will be explained with more detail further through the report.

Once the chip is programmed, we will solder it into the integrated circuit board as carefully as possible. Carefully being a key point as the pins and connections are sensitive and by doing this we ensure that our connections are free of shorts, disturbances or inhibitions of any sort in turn, our design will be ensured to work as best as possible after we solder and connect all of our components.

Before we begin to even consider implementing much less choosing a specific microcontroller into our project, we must first touch some bases and have a strong foundation on what exactly a microcontroller is. Simply put, a microcontroller conceptually is a controlling brain that does what it is told to do by previously specified orders. It does nothing more and nothing less than what it is told to do. These specified orders would be analogous to code, which can be as high leveled as assembly or machine code up to C, C++ or even java. It is designed to have connections or input and output ports through which it will receive or transmit signals and base its decisions with respect to the coded interrupt handling provided. I mention interrupts because the nature of our project is a fusion of sensors, which will transmit signals which will then trigger certain interrupts on our microcontroller.

2.7.1 Researched Microcontrollers

Having taken the newest Embedded Systems course which was based around the ever popular Texas Instruments MSP430 my first logical step was to weigh our options and do some research on said powerful yet versatile device. The MSP430 was a top potential candidate due to it being so accessible, cheap and effective with even complex projects. With it being attainable at $4.30, serious consideration was not given a second thought. It is capable of ADC, has a respectable amount of flash memory and SRAM along with a timer. The MSP430 ships straight from TI at $4.30 brand new in the box with the following:

- MSP-EXP430G2 Development board
- M430G2553 MPU
- M430G2452 MPU
- 32.768 KHz oscillator crystal which you can solder onto the board for further timing accuracy
- Two sets of extra header pins
- USB cable used for power supply as well as programming
- Some neat TI Launch pad stickers

Right off the bat we can see that TI is pushing its MSP430 Launchpad into hobbyist's hands at a dirt-cheap price. This is very hard to pass off on in comparison to an Arduino board that will run you up a couple more and not include nearly as much for the price asked. In the long run, the MSP430 is a great bang for the buck.

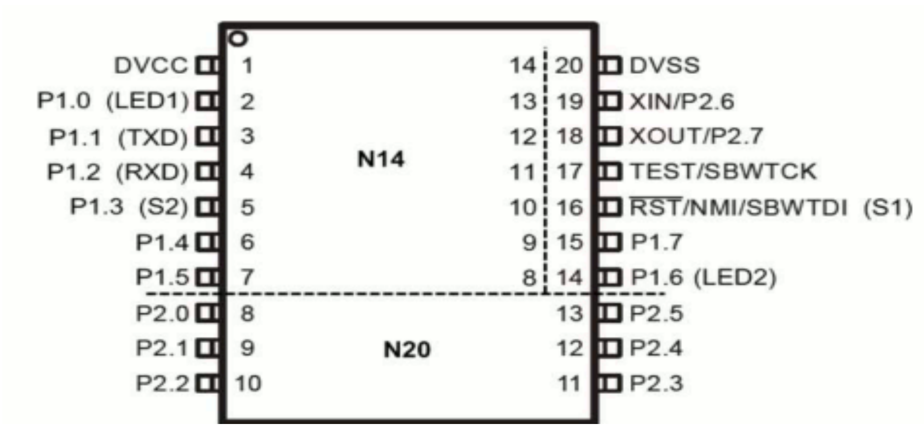


Figure 18: MSP430 device pin out

Printed with permission of Open Source

The layout of the board is split up into two logical sides. The USB interface is on the left hand side of the board and comes already pre-jumpered with the USB interface connected. There is of course a power LED on the USB side and power can come via USB or the header pins on the other side of the board.

On the other side of the board we have two push buttons, two LED's, header breakouts for each of the 20 pins on the IC socket, 3 pins for power input header (VCC, GND, GND), and two jumpers to connect and disconnect the two integrated LED's. The board itself is a very simple circuit excluding the USB converter on the side of the board.

Out of the two buttons, one of those buttons is a reset button hardwired to pin 16 and of course there's no real way to change the function of that button. The other button switch goes to pin 5 of the IC socket and that is of course an all-purpose programmable switch. There are also two programmable LED's, LED 1 (red) and LED 2 (green) and you have the option to easily disconnect them via jumpers.

Overall the board is cheap, well-designed and sturdy, small footprint and perhaps its most prominent figure of merit; power consumption. The MSP430 is extremely low power by nature and when it comes to integrating it in our design, it would make for a ridiculously low power consumption feature that I am sure none would protest to.

The next slate to touch would be to cross into the Arduino realm. The Atmel AVR controllers have tons of features to offer. The first upside being that they are the basis or the super popular and ever growing group of Arduino boards and as engineers, we can definitely agree that there is strength in the numbers. The Arduino community is unparalleled, which such a large crowd there is just a plethora of information on just about every product and task implementation ever thought of which makes Arduino automatically attractive to amateur programmers like us.

Our search began with the ATMega328, a widely used microcontroller. It has an I2C port that allows it to be used for any I2C peripheral needing the port. It has a decent amount of SRAM and flash memory meaning that it can hold modules of applications. As with the MSP430 the pin counts are unfortunately low but of course this does not limit its use as a secondary microcontroller in our system should we find the need to have one, we do have two separate systems after all. The other varieties were the Atmel Xmega line of microcontrollers, which offer much more power and processing capabilities. With their superior memory capacity, speed and greater pin counts they are easily an opportunity to abuse power. They provide multiple I2Cs, UARTs and pin counts that exceeded our estimated pin counts. While tempting, the Xmega series just seemed like too much power for our intended purpose of use when it came down to a choosing a microcontroller.

While we were in at that high of a level we also swooned over to the Stellaris LM4F. This is a seriously powerful surface mount microcontroller designed by none other but TI. It provides multiple I2Cs and UART ports. It has more memory than the MSP and Xmega as well as a much faster clock speed. The evaluation board runs up to around $150 and that of course was a red flag in our search as we do not need that kind of power nor that fancy of a price tag.

Having both too much power at a high price and not enough for a really nice price tag left us wanting to finally touch some neutral ground with a good combination of the two. Looking further through the Arduino website we found the trademark ATMEGA 2560 and while we loved what it had to offer we saw a very similar compromise and this time we couldn't really nitpick it, as it had just what we needed. This microcontroller is referred to as the Arduino Mega.

The Arduino Mega is a microcontroller based on the Atmega1280. It has 54 digital input/output pins with 15 having the capability of being pulse width modulated outputs. 16 analog inputs, 4 UARTs, a 16MHz crystal oscillator, a USB connection, a power jack, an ICSP header and a reset button. At a cursory glance, this is a seriously potential final candidate for us.

While not being as small nor power efficient as the MSP430, it gives us a lot more flexibility when it comes to pin counts and memory capacity which is always a good thing due to the fact that there will surely be a lot of features and maybe even parts that we will switch around, but the fact stays that it is a capable microcontroller with a large community support base, compatibility and ease of use and programming, which all are a plus when it comes to our limited programming experience. Below is a chart of the basic features the 1280 has to offer.

Microcontroller	ATmega1280
Operating voltage	5V
Recommended Vin	7-12V
Input Voltage limit	6-20V
Digital I/O pins	54(15 being PWM)
Analog Input pins	16
DC current per I/O pin	40mA
DC current for 3.3V pin	50mA
Flash Memory	128KB, 4KB being used by boot loader
SRAM	8KB
EEPROM	4KB
Clock Speed	16MHz

Table 7: ATmega1280

In summary, while the MSP430 offered low cost, accessibility and functionality if fell short in pin counts and communication interfaces. The Xmega series we just too much as well as the Stellaris, as beautiful of a microcontroller as it. And finally the 1280 was found to be a more reasonable and definitely suitable neutral ground between the MSP and the Stellaris.

2.8 HELMET

For security purposes, the government has created some rules and regulations for the motorcyclists around the United States; these regulations can change according to the state. The use of a protective headgear is one of these rules; in fact, this is a key factor that needs to be taken into consideration because the safety of the rider should always be a priority. For the Motorcycle Tracking Security System (MTS), we are planning on buying a helmet already built, but we are going to modify it, in order to properly add the required modules, buttons, speakers, microphone, and other components that make part of the helmet system. In order to keep our project as safe as possible, we are going to buy a brand new helmet unit that is certified with recognized safety standards and regulations. We could get a used headgear, but this is not recommended because a damaged helmet might not show any visible signs, and still not be on optimal conditions. For us, it is not acceptable to risk the security of the user just because we want to save a few bucks getting a used helmet.

There are different sizes of helmets, but we are thinking about selecting full-face medium helmet, according to Rigel Jimenez, one of our group members, and motorcycle rider. This model offers a comfortable design, it's stylish, and it should have enough empty space to locate the parts that need to be incorporated. We are contemplating the idea on having a small PCB inside of the helmet with a microcontroller, a Bluetooth module, a RF device, buttons, as well as the microphone and speakers for interaction. Table 7 shows some helmet models that we are taking in consideration, and that are affordable. Motorcycle helmet could be very expensive; for example, Shoei is one of the best brands, but their prices go above $400.00. The brands listed below are also very good and offer a good amount of security features.

Brand	**Model**	**Size**	**Price**
Scorpion	EXO 750	Medium	119.95
Scorpion	EXO 400	Medium	99
HJC	CS-R2	Medium	79.5
HJC	CL-16	Medium	125.99
Bell Arrow	Solid	Medium	89.95

Icon	Airmada	Medium	131.99
Nolan	N-43e	Medium	150

Table 8: Helmets and Prices

A full-face motorcycle helmet has a protective outer shell built stand impacts of large magnitude without cracking, followed by a layer of insulating material usually made out of cloth of soft material; within the two holes that allow air through the helmet. Perhaps the most key features of this type of helmet is that it covers your full face, specially by a full visor that can be flipped, some come with tinted shields to protect your vision from the Sun while riding. Safety wise, these are the most protective helmets money can buy. Figure 19 shows an example of a Full-face Medium helmet; it can better demonstrate the features previously said about this type of helmet.

Figure 19: Full-face Medium Helmet (Scorpion EXO-750)

(Permission in progress)

<u>Section 3: Design Details</u>

3.1 GPS Selection

After doing some research about GPS technology; and also comparing four different GPS modules, our group decided to select the Locosys LS20031 GPS receiver for our project. This module was chosen because its offers some really good features for its price. This receiver is well balanced for our project, and its specifications meet the requirements that we need to successfully built our motorcycle and helmet system. Figure 20, displayed below, shows that the module has 5 pins that will need connection, in order to be properly integrated into an electronic system. Figure 20 by itself does not describe each pin, but Table number 9 gives this information. The process of soldering the unit should be easy due to the shape and number of pins of the module. To solder down the receiver we are going to use an L-shaped wire. Referring back to our research about the Locosys LS2003, the unit requires an input voltage of 3.3 V; we will be able to achieve this exact voltage by using a voltage regulator.

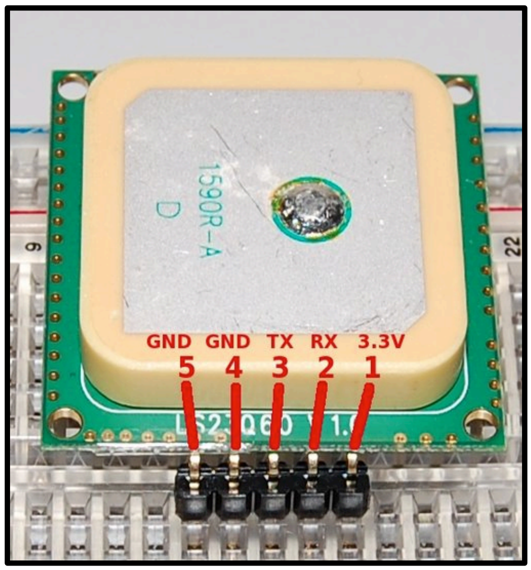


Figure 20: GPS LS20031Receiver Pin Layout

(Printed with permission of Open Source)

Pin #	Name	Description
1	VCC	Power Input
2	RX	Data Input (TTL level)
3	TX	Data Output (TTL level)
4	GND	Ground
5	GND	Ground

Table 9: GPS LS20031Receiver Pin Description

Another key factor to take into consideration is kind of information that the GPS receiver can give. The LS20031 module provides NMEA (National Marine Electronic Association) massages as output. The possible outputs that you can get from this receiver are shown below in Table 10.

NMEA record	Description
GGA	Global Positioning System Fix data
GLL	Geographic position (latitude / longitude)
GSA	GNSS DOP and active satellites
GSV	GNSS satellites in view
RMC	Recommended minimum specific GNSS data
VTG	Course over ground and ground speed

Table 10: GPS LS20031 Operation Modes

For our motorcycle and helmet system, we are only going to use the first two NMEA record outputs, which are GGA and GLL. GGA stands for Global positioning system fixed data. Table 11 shows a more detail description of the output that the receiver provides with GGA is used

**$GPGGA,053740.000,2503.6319,N,12136.0099,E,1,081.1,63.8,M,15.2,M,,00
00*64**

Name	Example	Units	Description
Message ID	$GPGGA		GGA protocol header
UTC Time	53740		hhmmss.sss
Latitude	2503.6319		ddmm.mmmm
N/S indicator	N		N = North or S = South
Longitude	12136.0099		ddmm.mmmm
E/W indicator	E		E = East or W = West
Position Fix Indicator	1		See Table Below
Satellites Used	8		Range 0 to 12
HDOP	1.1		Horizontal Dilution of Precision
MSL Altitude	63.8	meters	
Units	M	meters	
Geoid Separation	15.2	meters	
Units	M	meters	
Age of Diff. Corr.		second	Null fields when DGPS is not used
Diff. Ref. Station ID	0		
Checksum	*64		

Table 11: Global Positioning System Fixed Data (GGA) Output Description

To better understand the output provider by the module, we also need to understand the specific values that the GPS positioning fix indicator provides. Table 12 shows all the possible values.

Value	Description
0	Fix not available or invalid
1	GPS SPS Mode, fix valid
2	Differential GPS, SPS Mode, fix valid
3	Not Supported
4	Not Supported
5	Not Supported
6	Dead Reckoning Mode, fix valid

Table 12: Positioning Fix Indicators

The second NMEA that we need to understand is the GLL (Geographic Position-Latitude / Longitude); this is some of the data that we need to create the SMS message that we will send to the motorcycle owner, and the emergency contacts; in case of an accident or if the bike gets stolen, of course this data will need to be process to provide and output in English. Table 13 shows the code, in which, the GLL output is provided by the GPS receiver.

$GPGLL.2503.6319.N.12136.0099.E.053740.000.A.A*52

Name	Example	Description
Message ID	$GPGLL	GLL protocol header
Latitude	2503.6319	ddmm.mmm
N/S indicator	N	N = North or S = South
Longitude	12146.0099	ddmm.mmmm
E/W indicator	E	E = East or W = West
UTC Time	53740	hhmmss.sss
Status	A	A = Data valid or V = data not valid
Mode	A	A = autonomous, D = DGPS, E = DR
Checksum	*52	
<CR><LF>		End of message termination

Table 13: Geographic Position (latitude / longitude) Description

By obtaining the GLL and GGA output data; out project will be able to get accurate data about how many satellites are currently being seen by the GPS receiver, as well as the current geographic position, which the main purpose that made us integrate this module to our senior design project.

As we can see on the Figure 21 below, the first pin that we are going to be using is the VCC pin; we are going to use it to supply the 3.3 V that the module requires. The Pin 2 and 3, which are TX0 and RX0 respectably, will be connected to the microcontroller to transfer data. Finally the pins 4 and 5 will be grounded.

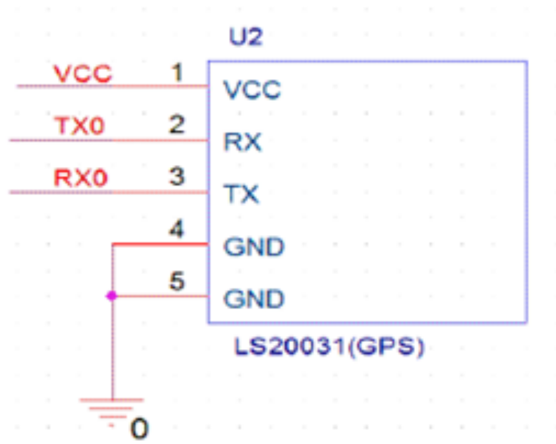


Figure 21: LS20031 GPS Receiver Schematic
Printed with permission of Open Source

3.2 Accelerometer Design

After the research was done and the parameters were narrowed down we were ready to pick the right accelerometer for our design. We looked at both options of using either two accelerometers to implement one function each and also one accelerometer to implement both functions. We were able to find one accelerometer with enough adjustability that can be used accurately to implement both options, and it will also be used for the helmet subsystem. Having two of the same product will make integration more compatible. The ADXL345 accelerometer was chosen to function as the motion and crash detector. Their cost is $27.95 from the vendor Sparkfun, and is manufactured by Analog Devices. Since this is a very small chip we decided to get it with a breakout board. This makes the soldering process much simpler. The breakout board will be soldered using headers to the circuit board. This part was also chosen because of the wide research and information available for it.

The ADXL345 meets all the requirements that are needed for our design. It also has extra features that if needed can also be implemented. By having the extra features it gives more room for its implementation and possible modifications. It has three different protocols interface that can be used with a microcontroller, SPI 3-wire, SPI 4-wire, and I2C. Since our accelerometer will only need to speak with the microcontroller we will be using the SPI 3-wire version. It is has of the simpler ones and requires less connections. The sensitivity and g-force mode has adjustability of ±2g, ±4g, ±8g, and ±16g selectable by the user, which is shown on the table below. It has 10 to 13 bits of resolution.

Sensitivity Modes of Operation			
FS[1:0]	**Range**	**Sensitivity**	**Application**
Set to 00	±2g	3.9 mg/LSB	Motion/Tilt
Set to 01	±4g	7.8 mg/LSB	Crash Detection
Set to 10	±8g	15.6 mg/LSB	Not Used
Set to 11	±16g	31.2 mg/LSB	Not Used

Table 14: Accelerometer Sensitivity

As far as the power consumption, the parameters for the ADXL345 are way below our max threshold. It puts very little strain on the battery that will be used as the power source. Its supply voltage ranges from 2.0 V to 3.6 V and the current consumption ranges from 40 µA to 145 µA. It comes with different modes of operation that conserve energy.

The accelerometer has a total of 14 pins, with two programmable internal interrupt pins that can directly communicate with a microcontroller when a certain threshold has been exceeded in the accelerometer. Also comes with 12 different modes that can have a certain threshold and once it is met it will be registered to its corresponding address in the register. The two modes that apply to our need are the tap ready and Activity mode. In Single tap mode a bit is set when a single acceleration event is greater than the value of the threshold. The Activity mode triggers when acceleration is greater the value stored in the threshold register.

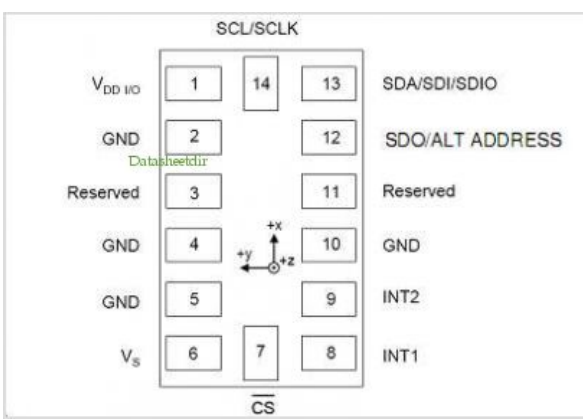


Figure 22: ADXL345 Pin Configuration

Printed with permission of Open Source

The accelerometer is going to be surface mounted to the PCB board. Pins 7, 13, and 14 are the I2C interface pins that communicate with the microcontroller. Pin 7 is the CS Chip Select that allows the communication for the accelerometer to start. This pin must be set low for communication to pass. Pins 13 and 14 can also be set to communicate by the SPI format as well. Applying the appropriate bit in the SPI register will select either the 3-wire or 4-wire communication. Any kind of information needed to be sent into the accelerometer will be done by either of these interfaces. It can also be programmed through this interface to sent interrupts if certain thresholds are met within the accelerometer. Pins 8 and 9, which are the interrupt pins, are going to be connected to the microcontroller. When a certain threshold has been reached it will send either of these two as a confirmation signal to the microcontroller.

Pin #	Pin Name	Description	Pin Status
1	VDDIO	Digital Interface Supply Voltage	Input
2	GND	Must be connected to Ground	Input
3	Reserved	Reserved.	Open
4	GND	Must be connected to Ground	Input
5	GND	Must be connected to Ground	Input
6	Vs	Supply Voltage	Input
7	CS	Chip Select	Input
8	INT1	Internal Interrupt 1	Output
9	INT2	Internal Interrupt 2	Output
10	NC	Not internally connected	
11	Reserved	Reserved.	Open
12	SDO/ALT	Serial Data Output/Alternate I2C	Output
13	SDA/SDI/SDIO	(I2C)/ (SPI 4-wire)/ (SPI 3-wire)	Input/Output
14	SCL/SCLK	Serial Communications Clock	Input

Table 15: ADXL345 pin description

The accelerometer comes with many different functions of interrupts to can be applied. Each of the interrupt functions can be used simultaneously, with the only exception being that a certain function may need to share the interrupt pins to work properly. To enable interrupts the appropriate bit has to be set in the INT_ENABLE register at the address 0x2E and are mapped either interrupts by the INT_MAP register at address 0x2F. Analog Devices does recommend having an interrupt disable just to make sure there are no faulty interrupts set. They can be monitored by reading the data in the INT_SOURCE register at address 0x30. All the functions that provide interrupts are listed below with an explanation of what each function does.

Interrupt Functions

- **DATA_READY**: Bit is set in DATA_READY register when new data is available, and it is cleared when there is no new data available.
- **SINGLE_TAP:** Set when a single acceleration event is greater than the value set in the THRESH_TAP register at address 0x1D. It also must occur for less time than the specified value in register DUR.
- **DOUBLE_TAP:** Set when two acceleration events are greater than the value set in the THRESH_TAP register occurring for less time than the specified value in register DUR.
- **ACTIVITY:** Set when acceleration value is greater than the specified value in register THRESH_ACT in address 0x24.
- **INACTIVITY:** Set when acceleration less than the value stored in register THRESH_INACT at address 0x26.
- **FREE_FALL:** Set when acceleration less than the specified value stored in the register THRESH_FF at address 0x28 is experienced longer than the duration value stored in register TIME_FF.

To be able to use all the different functions of this accelerometer we have to have the right bits in the right address. The microcontroller through the SPI 3-wire interface must give the data with the right address. Below is a Register Map that has the important registers that are needed for our project. In each bit of the register has a function that it implements unless noted otherwise. Through the registers below we can adjust the sensitivity, pick a certain interrupt source, a certain power mode, and activate the axis that is needed.

REGISTER MAP

D7	D6	D5	D4	D3	D2	D1	D0
1	1	1	0	0	1	0	1

Register 0x00-DEVID, Device ID

D7	D6	D5	D4	D3	D2	D1	D0
ACT ac/dc	ACT_X enable	ACT_Y enable	ACT_Z enable	INACT ac/dc	INACT_X enable	INACT_Y enable	INACT_Z enable

Register 0x27-ACT_INACT_CTL, Axis enable control for activity and inactivity detection.

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	Suppress	TAP_X enable	TAP_Y enable	TAP_Z enable

Register 0x2A-TAP_AXES, Axis control for tap/double tap.

D7	D6	D5	D4	D3	D2	D1	D0
0	ACT_X source	ACT_Y source	ACT_Z source	Asleep	TAP_X source	TAP_Y source	TAP_Z source

Register 0x2B-ACT_TAP_STATUS, Source of tap/double tap.

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	LOW_POWER	RATE			

Register 0x2C-BW_RATE, Data Rate and power mode control.

D7	D6	D5	D4	D3	D2	D1	D0
0	0	Link	AUTO_SLEEP	Measure	Sleep	Wakeup	

Register 0x2D-POWER_CTL, Power-saving features control.

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	Suppress	TAP_X enable	TAP_Y enable	TAP_Z enable

Register 0x2E-INT_ENABLE, Interrupt enable control.

D7	D6	D5	D4	D3	D2	D1	D0
DATA_ READY	SINGLE_ TAP	DOUBLE_ TAP	Activity	Inactivity	FREE_ FALL	Watermark	Overrun

Register 0x2F-INT_MAP, Interrupt mapping control.

D7	D6	D5	D4	D3	D2	D1	D0
DATA_ READY	SINGLE_ TAP	DOUBLE_ TAP	Activity	Inactivity	FREE_ FALL	Watermark	Overrun

Register 0x30-INT_SOURCE, Source of Interrupts.

D7	D6	D5	D4	D3	D2	D1	D0
SELF_TEST	SPI	INT_INVERT	0	FULL_RES	Justify	Range	

Register 0x31-DATA_FORMAT, Data format control.

Now knowing all the information on the accelerometer we need to know how to put it on a circuit board properly, meaning making we apply the right power source. The ADXL345 is quite simple to put together and apply the power it needs. As stated above the SPI interface pins and the Interrupt control go to the microcontroller. As far as the power the only precaution that needs to be taken is to place two capacitor one for the Vs and the other for Vdd I/O. The Capacitor Cs must be rated for 1 μF and C_{IO} at 0.1 μF.

These two input pins also have modes that can be activated with the right signal. The modes are Power Off, Bus Disabled, Bus Enabled, and Standby or Measurement. The two Bus modes are of no concern for our project, only off and Standby or Measurement. Both of these inputs are going to be sharing the same power source meaning it can only be on those two modes mentioned. Once the voltage is applied the accelerometer goes first into standby mode. It is recommended to configure the device in standby mode than by applying the right bit on the POWER_CTL register it can be switched to Measurement. In Measurement the device is fully on, receiving the inputs of the 3-axis.

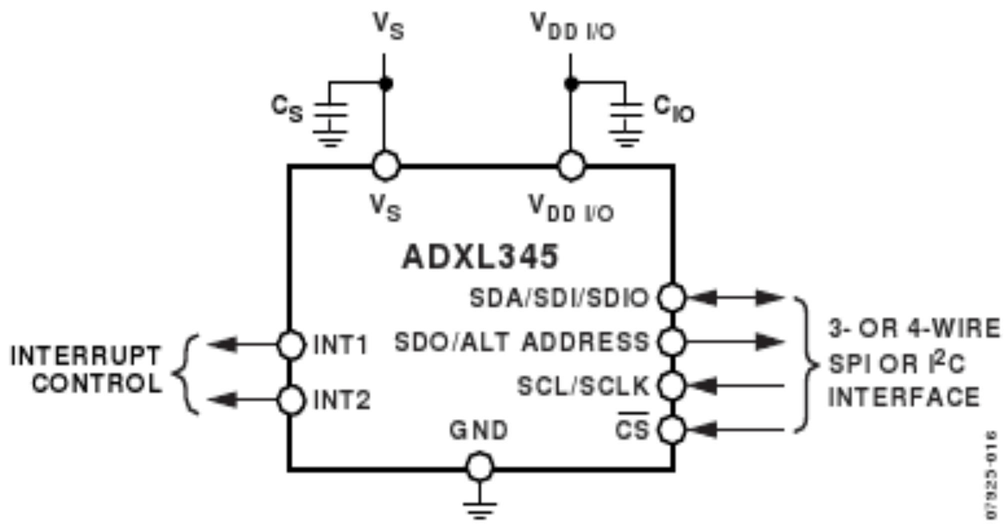


Figure 23: ADXL345 block schematic

Printed with permission from an open source Sparkfun.com

3.3 GSM Design

In the research part of the project, we analyzed the GSM and SMS technologies, and this improved our knowledge about the topic. We checked out two modules that met the basic requirements that we set up for the project, and we selected the SM5100B. Some of the things that made us choose this module were its specifications, its price, and compatibility with the selected microcontroller. As we mentioned before we liked the fact that the module had an incorporated antenna for better reception.

In order to properly integrate this part into the main system we need to connect it properly to our PCB design located in the Motorcycle Module. Figure 24 shows the Eagle Cad schematics for the SM5100B, showing the name of all its pins.

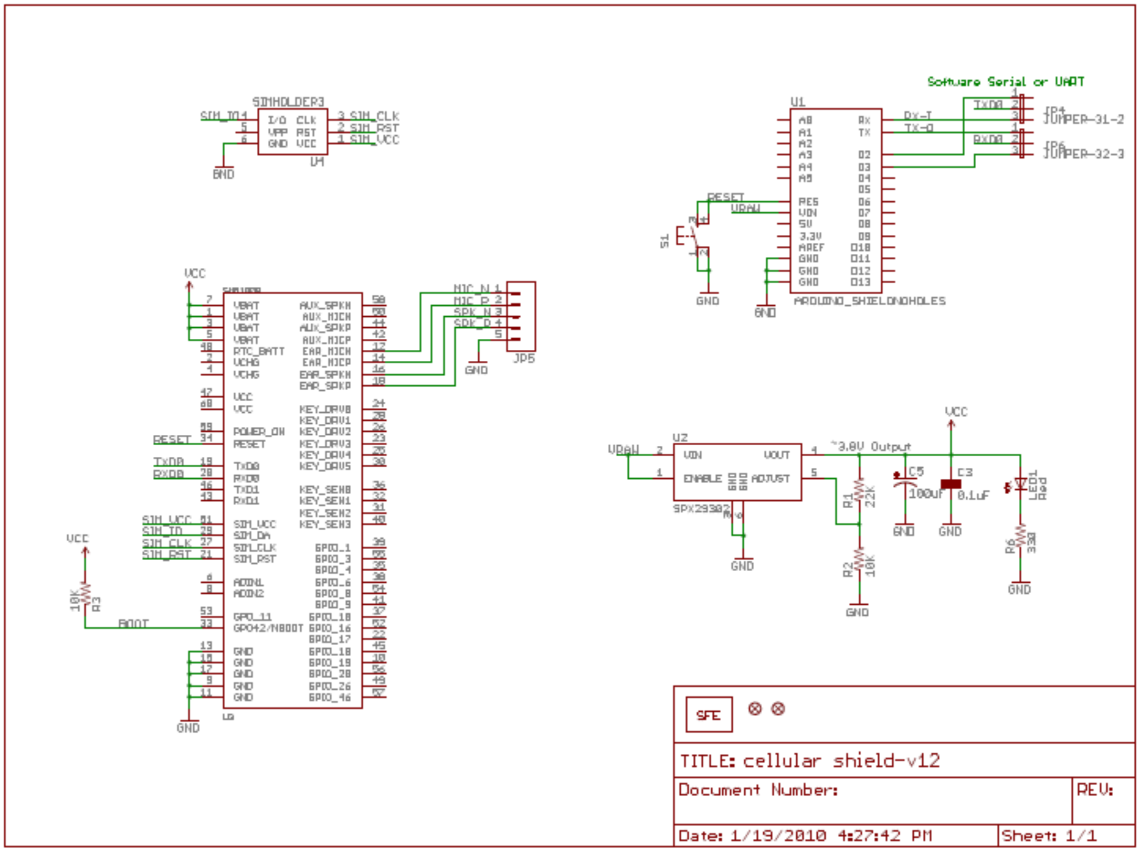


Figure 24: SM5100B Schematics
Printed with permission from an open source Sparkfun.com

The main purpose of using the module is to be able to send messages to the user or an emergency contact, in case of an accident or theft. In order to make this happen, we have to use the pins of the module that allows connecting to a cellular network by using a SIM card. Table 16 below shows the description of the pins that are used for this.

Pin number	Pin name	Type	Description
29	SIM_DA	I/O	SIM Serial Data
27	SIM_CLK	O	SIM Clock
21	SIM_RST	O	SIM Reset
51	SIM_VCC	P	SIM Power Supply

Table 16: SIM pin description

The module will also be connected to the power pins and to the 6 pins for the UART0 interface to transfer and receive data. These pins will be connect to the PCB and will go to the microcontroller to process the data. Table 17 shows the UART0 pin description.

Pin number	Pin name	Type	Signal	Description
19	TXD0	O	U0TXDN	Transmit Data
20	RSD0	I	U0RXDN	Receive Data
10	GPIO19/ U0_CTS/JTAG_TMS	I	U0CTSN	Clear to Send
22	GPIO17/U0_RTS/JTAG_DI	O	U0RTSN	Request to Send

Table 17: UART0 pin description

3.4 Bluetooth Design

The Bluetooth module WT32 was selected after reviewing a few modules in the research part of the project. The WT32 met our requirements as far as Bluetooth class, transfer rate, compatibility with cell phones being synch to the system, protocol compatibility, audio transfer capabilities, and range of coverage. Also, its input voltage can be stabilized with a voltage regulator to have a proper operation. One of the main reasons that made us picked this part was the fact that it’s popular in the market, and it has a reasonable part. The Figure 26 shows the schematics of the project with the names of all its pins. This design was taken from the website Sparkfun.com, the site offers good info about the part, including its datasheet, and it also sells the unit. This product has a few pins, but we still be able to perform the soldering so it can be integrated into the PCB located in the Helmet. This module will be located on the Helmet system, and it will be connected to the microcontroller, so it can be connected to a set of Speaker and a microphone that will allow the interaction with the Cell phone that will be synch to the project.

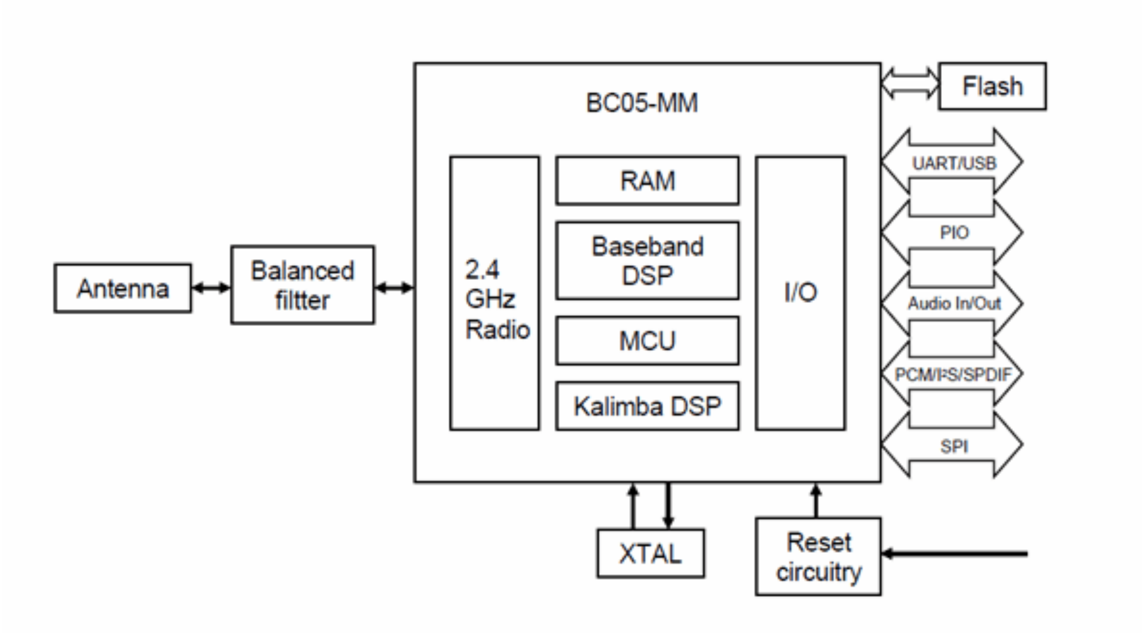


Figure 25: WT32 Block Diagram
Printed with permission from an open source Sparkfun.com

Figure 25 above shows a simplified block diagram of the module WT32. The module has 50 pins, and its descriptions are below in Figure 26. These are the uses of the main pins that we are connecting: The pins DGND and AGND will be use to ground the needed digital and audio pins; all the pins that start with VDD will be use as input for internal power; VReg will be used for the power regulator; RES is the integrated reset pin; and all the pins that start with the letters UART will be connected properly to transfer data through the UART interface; the use that we are not going to use will not be connected, or they just might end up being grounded.

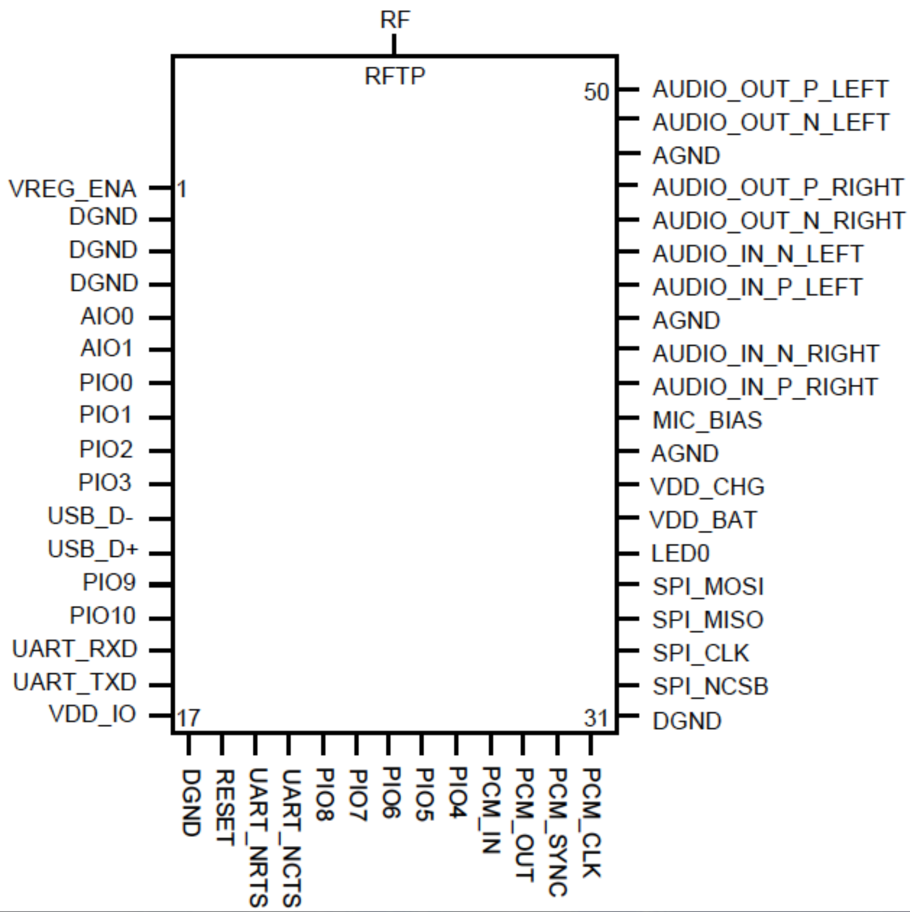


Figure 26: WT32 Pin descriptions
Printed with permission from an open source Sparkfun.com

3.5 RF Device Selection

Aside from Zigbee, there wasn't a large deviation on research between brands of different RF cards due to the fact that we trusted Zigbee and were just impressed by their compatibility and integrability in just about any project that involved RF communication between devices. This card is perfect for communication between the motorcycle and the helmet. As a solution to that, our team chose to use the Xbee 2.4GHz RF module by MaxStream shown in figure X. The module input power is of 3.3 V, which will be supplied by the actual motorcycle battery with the aid of some linear voltage regulators. And of course the communication peripheral is achieved through a simple wired antenna mounted on top of the Xbee chip.

Figure 27: XBee 2.4GHz RF Module

Printed with permission from an open source Sparkfun.com

The XBee has a “change detect” function in which a digital line passing is used to transmit a signal. This is basically a virtual wire created between an output pin on the transmitting RF card to a corresponding input pin on the receiving card so the RF cards serve functionality of transmitting and routing signals back and forth which one of our core requirements for our project.

Table 18 lists the assignments for the complete 20 pins on the XBee chip. First and foremost, power is supplied at pin 1 while pin 10 is utilized for grounding purposes. Our pins of primary interest are pins 18, 19 and 20. Not to mention that pin 18 will be used for our accelerometer for crash detection purposes. These will be the primary pins that will be used for transmission and reception.

Pin #	Name	Direction	Description
1	VCC	-	Power Supply
2	DOUT	Output	UART Data Out
3	DIN/CONFIG	Input	UART Data In
4	DO8	Output	Digital Output 8
5	RESET		Module Reset
6	PWM0/RSSI	Output	PWM Output 0/RX Signal Strength
7	PWM1	Output	PWM Output 1
8	Reserved	-	Do not connect
9	DTR/SLEEP_RQ/DI8	Input	Pin Sleep Control Line or Digital Input 8
10	GND	-	Ground
11	AD4/D1O4	Either	Analog Input 4 or Digital I/O 4
12	CTS/DIO7	Either	Clear-to-Send Flow Control - Digital I/O 7
13	ON/SLEEP	Output	Module Status Indicator
14	VREF	Input	Voltage Reference for A/D inputs
15	Associate/ADS/DIO5	Either	Analog input 5 or Digital I/O 5
16	RTS/AD6/DIO6	Either	Request-to-Send, Analog input 6 or Digital I/O 6
17	AD3/DIO3	Either	Analog input 3 or Digital I/O 3
18	AD2/DIO2	Either	Analog input 2 or Digital I/O 2
19	AD1/D1O1	Either	Analog input 1 or Digital I/O 1
20	AD0/DIO0	Either	Analog input 0 or Digital I/O 0

Table 18: XBee Pin Assignments

The inputs and outputs of the chip will of course be configured as digital I/O with a specific sampling rate depending on the connections. We will of course check for the latest firmware to ensure that the chip is working as efficiently as possible before integrating it into our project, this firmware possibly being the A04 revision.

As far as the physical connections being implemented, we will be adding some capacitors to the input of the RF card for noise filtering purposes. We will be doing our best to keep the pins clear of noise and interference at all costs. A good RF design will compensate for noise and loss and that's the exact same path we want to take with the Xbee and after further reading, it is important that we leave all unused pins disconnected to ensure proper device functionality.

3.6 Battery Selection

After looking at several different types of batteries we decided to go with a Li-Po battery manufactured by Union fortune and sold by the vendor Sparkfun at a price of $11.95. The battery is a single cell Li-Po rated for 1000 mAh of duration. It meets all of the requirements needed for our design. It offers a very slim design profile. Its width is only 5.9mm which helps us make the place where we are going to put at as compact as possible. The figure 28 shows the Li-Po battery we picked and the dimension of its height and length.

Figure 28: Dimensions of Li-Po Battery

Printed with permission from an open source Sparkfun.com

The table below shows the specifications that are important toward our design. With a capacity rating of 1000 mAh the battery will be able to last for 10 plus hours of operation. The components will only pull a small current discharge from the battery when it is in operation mode, also keeping well below the max discharge threshold. It can supply what the components need at a very safe margin.

Specifications	
Nominal Capacity	1000 mAh
Nominal Voltage	3.7V
Charge Current	Standard 0.2CA, Max 1CA
Charge Cut-off Voltage	4.2V
Standard Discharge Current	0.2CA
Max Discharge Current	2.0CA
Discharge Cut-off Voltage	2.75V

Table 18: Li-Po battery specs

3.7 Recharge Circuitry

To be able to properly charge the Li-Po battery chosen for our project we had to get an IC chip programmed with the CC/CV method to charge the battery. This will make sure to prolong the life of the battery to its maximum use. We decided to go with the MAX1555 SOT23 Dual-Input USB/AC Adapter 1-Cell Li+ Battery Charger manufactured by Maxim Integrated Products for $1.95. It meets the requirements to properly charge our Li-Po battery. The Figure below displays the MAX1555 with its pins labeled.

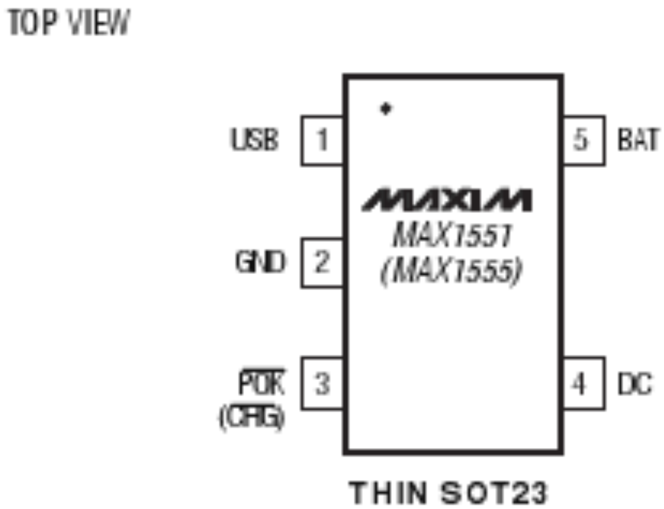


Figure 29: MAX1555 IC with pins

Printed with permission from an open source Sparkfun.com

The MAX1555 can take two different types of input sources, either a USB source, or a DC input from a wall plug. Depending on which source is used will dictate the rate of which the battery is charged. The USB input charges at a current of 100 mA and DC input at a max current of 280 mA. We will choose to charge with the DC input source because standard charge current is 200 mA and USB will not be efficient enough to charge it. The DC source will be connected to a USB mini-B since it is one of the most common connections. This helps keep the device very universal to be able to charge it, not needing any kind of special charger.

Figure 30 shows the circuit suggested by MAXIM to properly install the MAX1555, and also the inner workings of the chip itself. The figure is followed by Table 20 of the functions of the pins. Where USB and DC are the input source, BAT connects to the battery being charged, ground, and CHG which is the status indicator. CHG is the active-low Open drain status pin. Once the battery has been fully recharge CHG goes high to provide indication.

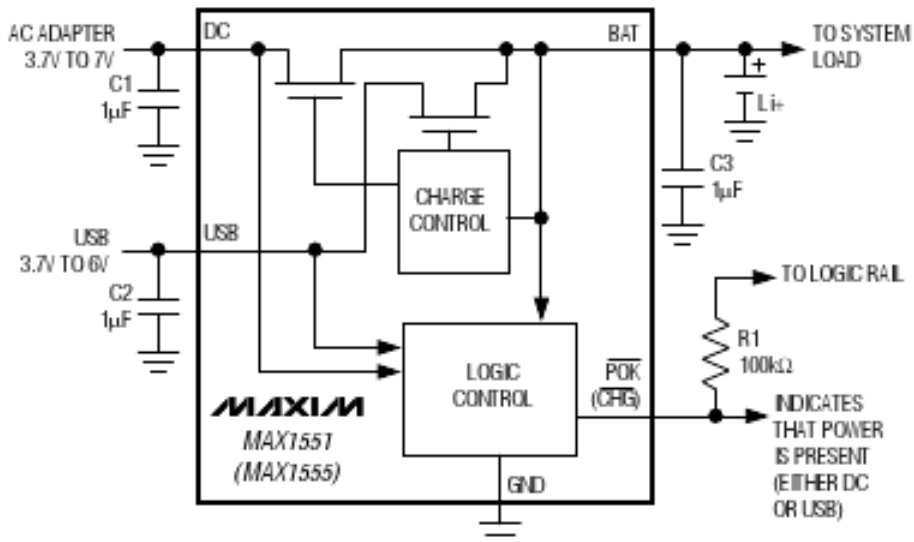


Figure 30: Suggested Circuit with description

Printed with permission from an open source Sparkfun.com

Pin	Name	Function
1	USB	USB Port Charger Supply Input
2	GND	Ground
3	CHG	Active-Low Open-Drain Charger Status Indicator pulls low when the battery is charging. CHG goes to a high-impedance state indicating the battery is fully charged, when the charger is in voltage mode and change current falls below 50 mA. CHG is high impedance when both input sources are low.
4	DC	DC Charger Supply Input for an AC Adapter.
5	BAT	Battery Connection.

Table 20: MAX1555 pin function

3.8 Microcontroller Selection

As noted in our research, lots of time was put into researching our options when it came to microcontroller. While some had too much power and or pins, others just simply did not. As far as finding a neutral balance, it was not too difficult to run into. We spoke to previous senior design students and discussed our project idea, function and goals and while we got some mixed opinions, some even a bit biased the majority sent us to the Atmel realm. This makes perfect sense as none of us are programmers nor do we have any extensive programming experience. Atmel features a familiar C language with some easy to use functions, which fit perfectly with what we're up against for the remainder of our senior design curriculum.

On figure 31 below is our narrowed down finalized choice for a microcontroller. We chose to go with the ATmega1280. A high performance, low power Atmel 8 bit AVR RISC-based microcontroller. At first we were going to go for the 2560 model but through some further research we realized that we did not need as much memory as the 2560 offered but we still needed the same basic features which is exactly the difference between the 1280 and the 2560. This microcontroller has more than enough pins, decent memory, clock speed and fits right in overall with our design specifications. With multiple UART and SPI interfaces and one I2C, it is compatible with ever device we have listed so far in this design.

One of the reasons we went with Atmel was the fact that we are looking to have as much support as possible with the microcontroller community and using this microcontroller automatically puts us at an advantage when it comes to finding previously utilized coded, not to mention all the forums that are available for us to ask questions regarding the compatibility and quirks any of our modules might have once we put the two together.

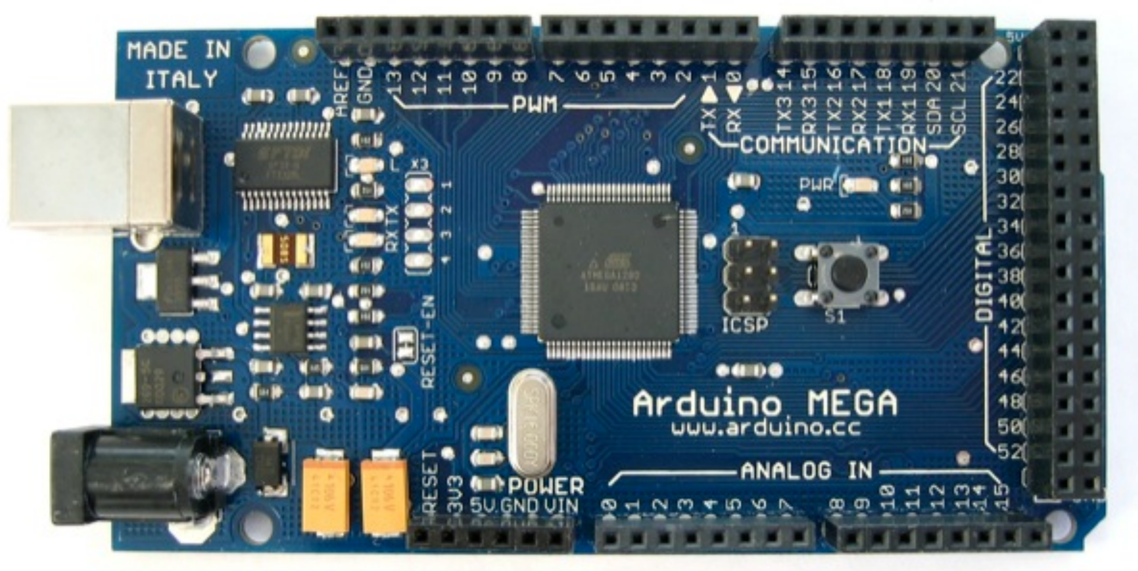


Figure 31- Arduino MEGA 1280

Printed with permission from an open source Sparkfun.com

Section 4: Design Summary

The design of the motorcycle safety system was a collaborative effort as to help improve rider's safety and functionality in their everyday gear usage. We are a team of three Electrical Engineering students with different interests in the many subfields of the discipline therefore we chose a project that could further develop us as engineers and challenge our knowledge while learning about current topics in our respective technology subfields. The main requirements for the project were to incorporate technology and intelligence into an everyday riding gear. Nobody would assume that something so common could provide so much more functionality and further build on its original safety purpose.

As with many other designs, our system is of course a conglomeration of different subsystems but as usual, that is the basis of any engineering system. The main subsystems of our design include GPS, Bluetooth, RF communication module, alarm software/hardware, Accelerometer/Motion sensor, power system and a main brain to control all of the signals being sent back and forth; the microcontroller.

Our design current design is not as user friendly due to the fact that the user does not have too much control over customizing the system. The system works as is and sends signals to the microcontroller with respect to the instructions pre-programmed into the board. As primitive as the design is, if it were ever to make it into the market for regular consumer use, an additional peripheral would have to be implemented, this being a sort of user interface whether it be a touch screen, voice recognition or more buttons to control some of the basic functions of the helmet-motorcycle subsystem.

Perhaps the most basic feature we have on this design is the alarm software, which incorporates the accelerometer, the microcontroller, RF system and external handheld controller (FOB). The FOB will have a minimal amount of buttons to just trigger the alarm on and off. The mechanism is simple, the accelerometer detects a tilt or sudden significant movement and then it signals an interrupt to the microcontroller in which then the microcontroller can trigger the loudspeaker and sound off the alarm, very basic system yet effective as it is a feature normally found in cars whereas motorcycles all lack this feature in stock form. To further add to this feature, an accelerometer will also be implemented in the helmet itself to protect it from any kind of theft.

The GPS will serve a basic function of just keeping track of where the bike and helmet are at all times whether It be normal state or an altered emergency state. Of course the GPS module can give us much more functionality to exploit than what we're currently choosing but the design specification is just to keep track of the bike in case of theft, monitoring purposes or accidents in which any case the user and his desired contacts will have the current latitude and longitude readings which can be plugged into any mapping software, that way the resultant location can be calculated to a respectable accuracy.

Besides providing safety and security it has its features of convenience. For instance, Bluetooth will be implemented in the helmet to provide a means of hands free communication for the rider in case of emergencies or leisure calls. The Bluetooth will be able to pair to any of the smart phones out in the market that have the capabilities. Once the rider's device is paired it can transmit the audio from the phone, meaning that it can also listen to music or any kind of audio file. A more important function it gives is that during a case of an accident, the emergency response receives the message; it can call at the riders designated cell phone to check for the seriousness of the matter. If rider is not in a serious accident it can pick up the phone call and explain the situation, but if the accident is severe most likely the rider will not be able to answer the call. When this happens the worst case scenario will have to be assume and send for help as quickly as possible to the coordinates given.

The different functions of our design can be described as three different states; Alarm off with bike off, alarm on with bike on, and bike on with all components on. With the alarm off and the bike off it's pretty simple to know the meaning of that state. All components are powered down or to very low current consumption and are not providing any function, in other words it completely shuts down the device. Using our frequency oscillating button (FOB) device, we will turn on or off the alarm system wirelessly. Once we use the FOB to turn the alarm system on and begin to monitor. The main device that will provide monitoring function will be the accelerometer. At this state the accelerometer will be set as a motion sensor. There will be one sensor on the motorcycle and one on the helmet to protect both objects from theft. If the accelerometers threshold is passed, then it will wake up the rest of the system and take action. The accelerometer sends an interrupt signal to the microcontroller, it then wakes up to take immediate action. By sending signal to the GPS and retrieving the coordinates, it then transfers it to the GSM module. The GSM module is then responsible for transmitting the message with its coordinates through the network to the appropriate recipient. The table below shows in a more organized manner the task of each device at this state of the alarm

Device	Description
FOB	To transmit radio frequencies that turn on and off the alarm system.
Accelerometer	Monitors for motion, any motion detected on the helmet or on the bike that passes its threshold will set a trigger for the alarm. Powers up Microcontroller from sleep mode.
GPS module	Retrieves current coordinates and sends it to microcontroller; when the microcontroller receives the interrupt signal from accelerometer.
GSM module	Establishes connection with network. Once it receives signal from accelerometer it sends coordinates from the GPS through the network to its proper recipient.
XBEE	Transmits signal from FOB to the microcontroller. Also transmits data from the helmet accelerometer to microcontroller.
Microcontroller	Transmits data with all modules of the system. Receives interrupt signal and sends out certain commands.

Table 21: On state with bike off component tasks

The last state the motorcycle can be in is the Bike On all components On. In this state all the components are powered on. The accelerometer switches from detection motion to detecting accidents. The microcontroller gives the program to the accelerometer that changes the sensitivity and the mode the accelerometer is in. Once again the trigger starts with the accelerometer passing its threshold. If the threshold is passed, that means that the rider experienced a g-force typical to that of an accident. Once the threshold is passed, it will send an interrupt signal to the microcontroller. Now the microcontroller will send out its given instructions. It will retrieve the GPS coordinate from the module and transmit it to the GSM to send through the network. Having both accelerometers we can know if the bike is on its side and if the rider was flown out of his bike. By having this information it can be relayed to the emergency response contact. The message relayed will have information of rider, emergency response can than choose to call the rider if need be to assess further the situation. The Bluetooth integration will make it possible for the rider to answer phone calls. The table below will give the task of each of the devices.

Device	Description
Accelerometer	It is set detect high levels of g-force. Once threshold is passed it sends interrupt signal to the microcontroller. Also sense for bike orientation after accident. The helmet accelerometer senses if rider is flown off the bike.
GPS module	Retrieves current coordinates and sends it to microcontroller; when the microcontroller receives the interrupt signal from accelerometer.
GSM module	Establishes connection with network. Once it receives signal from accelerometer it sends coordinates from the GPS through the network to its proper recipient.
XBEE	Transmits data from the helmet accelerometer to microcontroller.
Microcontroller	Transmits data with all modules of the system. Receives interrupt signal and sends out certain commands.
Bluetooth	To be able to receive calls hands free.

Table 22: Bike on component tasks

Section 5: Testing Procedure

5.1 GPS Subsystem Test

The GPS subsystem is one of the various subsystems that incorporate Motorcycle Tracking Security System (MTS). Tracking the geographic location of the bike is the main purpose of it; in fact, it should be able to find the receiver anywhere in the world. The first thing that we are going to check is the ability of the receiver to effectively fix, which is pretty much making sure that the module is properly connecting to the satellites. The second aspect that we are going to check is how long the receiver takes to lock on a satellite and starts transmitting coordinates; this is a cold start test. To do the cold start test we are going to unplug the unit to make sure it does not get any power, and we are going to leave it like that for a day. After that we are going to check the connecting time, we are going to repeat this process for three days to get a little more accurate values. The third thing to take into consideration is the location, the rider will be utilizing our project in different places; therefore, we decided to test the GPS in different locations around town to make sure that we are getting the correct output at all the time, we would like to test the product on different cities or states, but it will require us to spend more money to pay for the travelling expenses. Some of the places that we are planning to use to test the unit are:

- The first floor of the UCF library. Testing in this place will show that the unit is working on an indoor location.
- The second floor of the UCF parking garage C. This is a mixture of an indoor and outdoor location, which is one of the environments that we want to test. Also the motorcycles are often park in these places, so having a good reception on a parking garage is important.
- One of the streets of downtown Orlando. Making sure that the receiver is getting signal outdoors.

We are going to check the geographic location outputs that we get in all these places, and we are going to compare them to the values that the free cell phone app Google maps provides, to make sure our subsystem is accurate. We are expecting to not have a difference greater than 3 meters when we check latitude and longitude.

5.2 Motion system testing

Before integrating the accelerometer to the project we first have to do the proper tests to make sure the sensor is working like it is supposed to. Each of the interrupt functions also have to be tested since it is a crucial part of the task the accelerometer has to provide. Luckily the accelerometer comes with its own Self-Test feature that we can use to confirm the operation of the device. The Self-Test feature effectively tests its mechanical and electronic systems. The register DATA_FORMAT in address 0x31 holds the bit that turns on the Self-Test bit. After the bit is set to 1 then the device puts an electrostatic force on the mechanical sensor. This force applied to the accelerometer behaves in the same manner acceleration would. If the device is already undergoing some kind of acceleration, the force applied by the Self-Test will be added on to the present acceleration. In other words the current acceleration will be offset by the value on the Self-Test. However these Self-Test values all have a certain threshold limit for each range of g-force that it is in. The Table below shows the upper and lower bounds of the Self-Test function. For the proper function of the Self-Test mode, the output data rate should be set to 100 Hz or higher.

G-Force Range	Axis	Min (LSB)	Max (LSB)
±2g	X	50	540
±2g	Y	-540	-50
±2g	Z	75	875
±4g	X	25	270
±4g	Y	-270	-25
±4g	Z	38	438
±8g	X	12	135
±8g	Y	-135	-12
±8g	Z	19	219
±16g	X	6	67
±16g	Y	-67	-6
±16g	Z	10	110

Table 23: Upper and Lower bounds of Self-Test mode

Before we can even try the Self-Test function we need to initialize the accelerometer with the proper coding to be able to store and see the data that the device is producing. The right file libraries need to be uploaded and the right registers need to be set. All of this is done through the code set in the microcontroller. We will set up a very basic code to make sure that the accelerometer is working properly.

First we start with the header, where all the parameters are initialized. Below will be a section of the code that shows the initialization process. The SPI.h library is added to the sketch.SPI, this is the communication protocol that the microcontroller uses to be able send data and retrieve it from the accelerometer. Next thing we have to initialize is the CS pin, which it needs to be set for data to come out of the accelerometer. Now that we have established the communication, we have to create variables for the registers that are going to be used. Variables are also created to store temporary data and use it for certain functions.

This second section will be used to configure the other different aspects that are needed for proper communication with the microcontroller. There are two functions that are going to be enabled to be able to get data from the accelerometer, the g-force range and measurement mode. With these two functions on we now implement a loop that is going to be reading the data being produced by the x, y, and z-axis. The first 8 bits will be put in its proper register and it will take the remaining two, because this accelerometer reads with 10 bits of resolution, and move them to a different register. Following the loop are print functions that are going to display the data it is receiving from the accelerometer. By giving it a 10-millisecond delay it ensures that the loop is running at 100 Hz.

Now that we are able to read data from the axis and put them in the accelerometer, we need to be able to extract that data into the microcontroller. The command that does this is readRegister function, which needs three parameters to go with it. It needs the register address, the number of registers that should be read, and where the values should be stored. Then to be able to start the communication between the two devices the CS pin must be set low. Once it is low the following commands will take action and take data from the accelerometer. When the communication is finished the CS pin is set to high again.

Finally it will display the data on the terminal window of the microcontroller. If we are getting readings it means that the code is working and so is the accelerometer. But we still have to make sure they are appropriate readings. Now that the data is continually displayed, we can start by moving the accelerometer and see the changes taking place. Once that confirms we can then set the Self-Test bit on to check if the values that are being displayed are the proper ones. If everything confirms then we now that the accelerometer is working as it is design too and the code can be changed to the one needed to perform the tasks for the design.

5.3 GSM Subsystem Test

In order to test the GSM module SM5100B, we are going to use the Cellular Shield (SM5100B) with Arduino, this is pretty much a subsystem already build that has all the capabilities that a regular phone will have; and it comparable with our microcontroller, Figure 32 shows how this Shield looks like. The main components of the Cellular Shield are a 60-pin SM5100B connector, a SIM card socket, and an SPX29302 voltage regulator configured to regulate the Arduino's raw voltage to 3.8V. The board's red LED indicates power. The Arduino's reset button is also brought out on the shield. The Cellular Shield already has a socket for the SIM card for easy integration, and we are just going to use one of our already activated SIM cards to test if it does receive a signal and operates properly.

Figure 32: the Cellular Shield (SM5100B) with Arduino
Printed with permission from an open source Sparkfun.com

5.4 Bluetooth Subsystem Test

To test the Bluetooth module, we are going to do a Loopback testing. First of all we are going to wire up the module properly to the Arduino board, the Arduino board is going to be getting the supply power from its power source, then we are going to short circuit the rx and tx pin. By doing this you get exactly what you send. Figure 33 below show the standard proper wiring between and Arduino and a Bluetooth module.

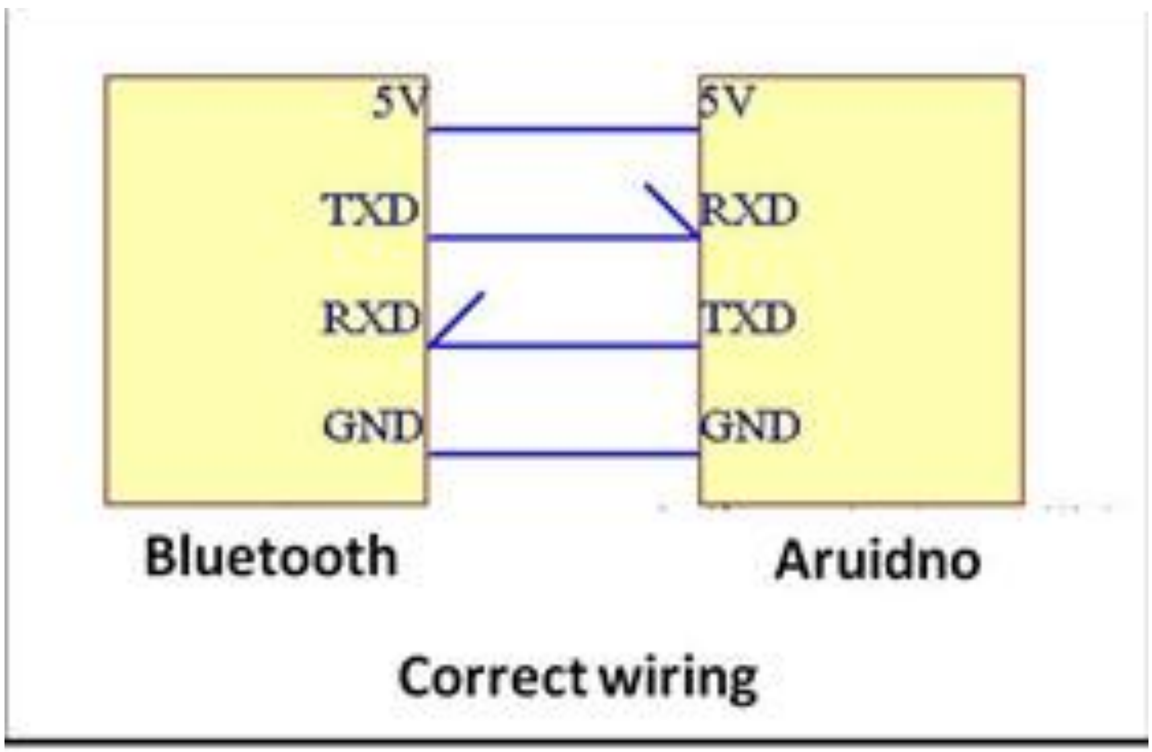


Figure 34: Correct Wiring

Printed with permission from an open source

The second part to the test is having the Bluetooth connected to the ATmega1280 we need to give it commands and see how the module response to it. Bluetooth module comes with Self-Test coding that needs to be implemented to the ATmega1280 microcontroller.

The code will be implemented using the Arduino environment, which is the one used by the ATmega1280 microcontroller unit. The testing code starts by configuring the ANSEL and ANSELH pins as digital pints; then the UART module is initialized at 9600 bps with a delay of 100 ms. Finally, a while loop will be implemented and it will display the word TEST every 2000 ms. The figure number 35 is an example of the output If device is working properly, and the testing went accordingly the desired output shown in the figure below will be displayed.

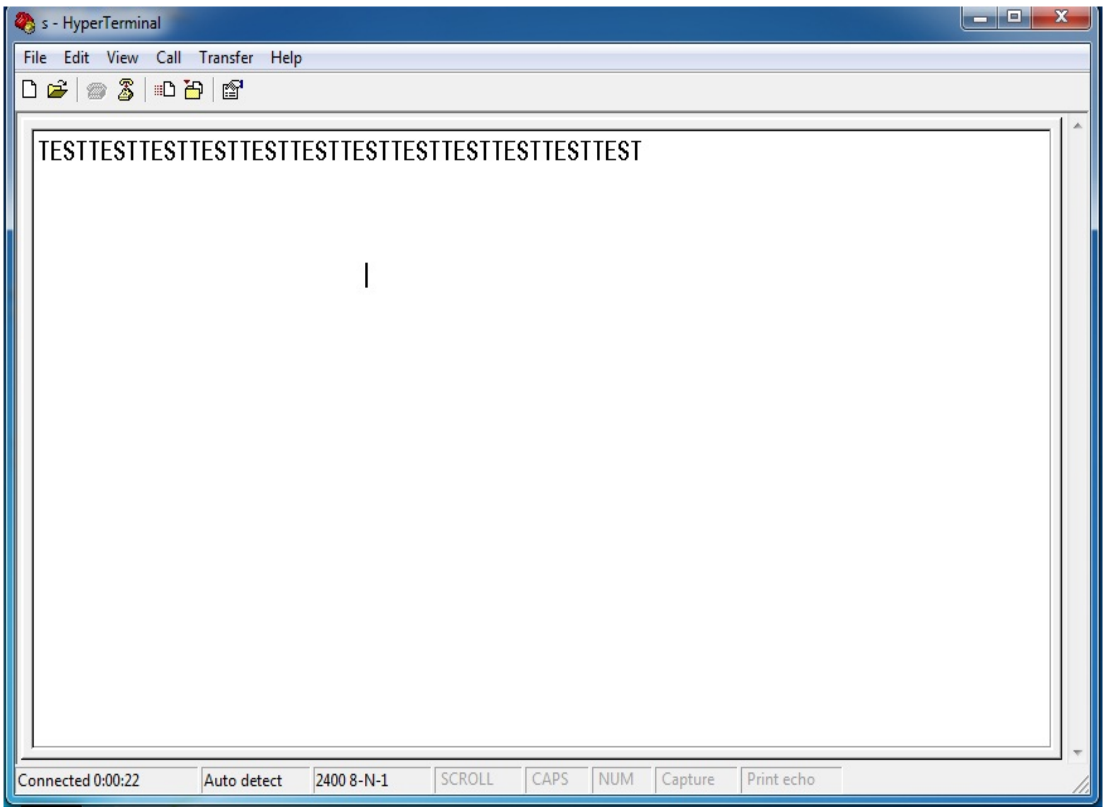


Figure 35: Bluetooth output result

Printed with permission from an open source

Finally, the WT32 is a Class 2 Bluetooth module with Audio transfer capabilities, so once we getting working, we are going to test with a few devices such as an IPhone, and Android smart phone, and a pair of Wireless Bluetooth headsets to make sure that it does pair and works properly, before it is fully integrated in our main system

5.5 Microcontroller Testing

There are many different ways to test the microcontroller unit, one of course just run some code in the IDE and check that it compiles correctly and ensure that the microcontroller in itself is working properly but in reality a more effective way would be to test every single peripheral along with the microcontroller to check and make sure that module is interacting correctly with the microcontroller. I have chosen to prove this by testing the XBee module and the microcontroller together.

First and foremost we want to test the connections between the microcontroller and the XBee module all of which are very simple and almost self-explanatory 4 wire connection. The +5V on the Arduino is connected to the +5V on the XBee, the two grounds are connected together, the Digital pin 2 is connected to TxD on the XBee and finally digital pin 3 is connected to the RxD on the XBee. From here we can find out how the Arduino can talk to the XBee via a serial port. From here the rest is easy as we have a code we can run to ensure proper functionality of not only the microcontroller but also the Xbee-Microcontroller subsystem.

The testing code that we are using to test the microcontrollers and the Xbee units requires the use of the external Arduino library called <SoftwareSerial.h>; this library will need to be added to our Arduino MCU software. Then we are going to set up the Xbee module, and we wil assign the RX and TX to the pins 2 and 3. The purpose of this code is to transfer data; and every time a character is transfer, and LED that will be connected to the Arduino MCU will light up, as you can see on Figure 36.

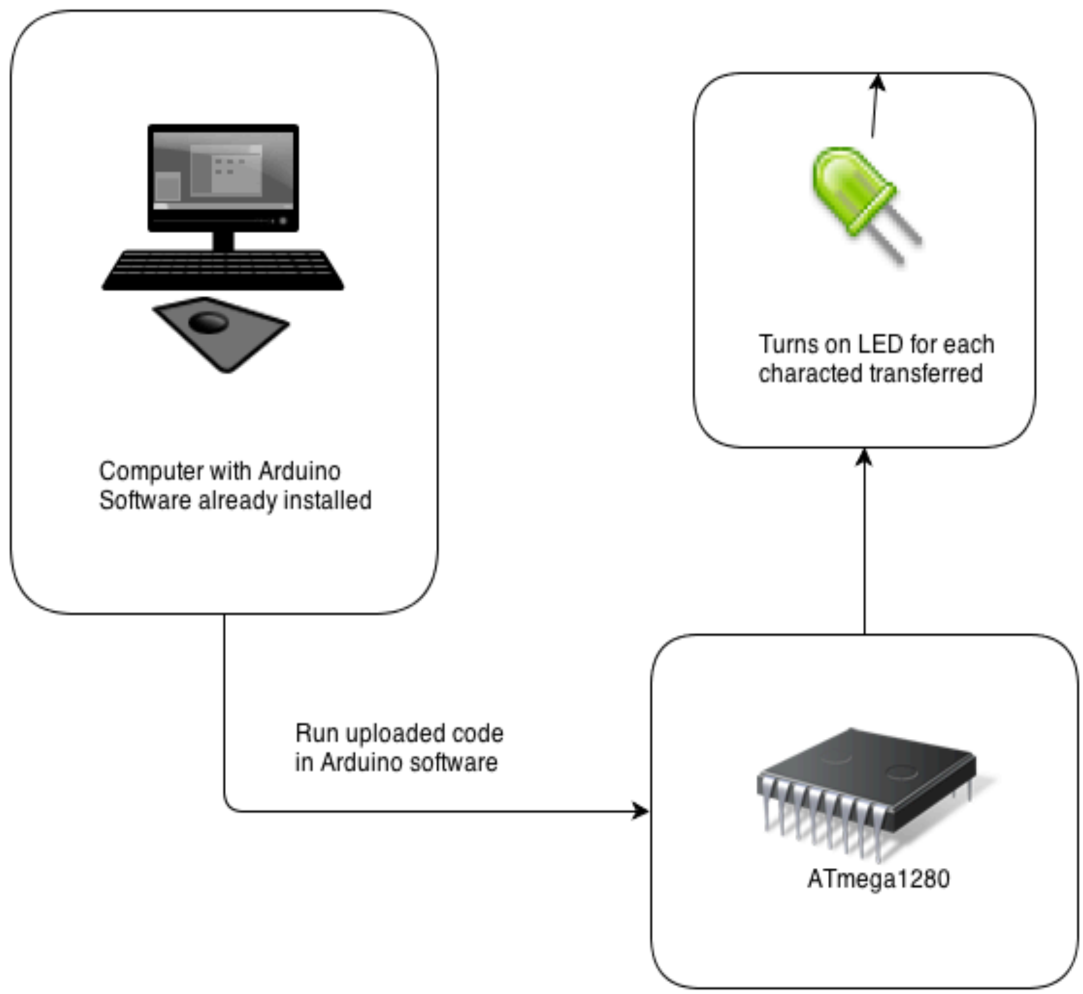


Figure 36 Block Diagram of Arduino Testing Code for MCU

After we compile this program and upload it to the Arduino. On the PC, we can use X-CTU software and click on the terminal tab to monitor data received wirelessly from the Arduino. On the Arduino side, we can go to the IDE and use the tools menu to open the console window to see the Arduino print the characters as they are sent. Overall, the code that we are implementing will make alphabet characters appear once a second in both windows. One is the console, attached to the transmitting Arduino; the other one is the terminal window of the X-CTU software on the PC side, receiving the characters sent wirelessly.

There are also some simple Examples that come with the Arduino software that can me used to test the microcontroller unit without the need to connect other external components to the MCU. One of them is the popular Blink code that turns on one of the little lights that are integrated in the ATmega1280 board, when you connect the unit to the computer via USB, and of course you have to upload and run the code in Arduino.

5.6 RF Module Testing

As stated before, our group decided to use the XBee 2.4GHz RF module. While these things work straight out of the box with little to no setting up it is crucial that we perform some tests and know how to modify some parameters before we can say that we can implement it into our group's overall design. The XBee has a main tool that we can use for testing means as well as changing some crucial parameters around. This tool is of course the X-CTU tool that can be installed into a PC and from there we could play around with a couple of parameters the XBee has to offer. Assuming we already have X-CTU installed we can begin to tinker with some parameters as soon as we plug in our XBee explorer. X-CTU has a couple of tabs up top of which the most important ones are PC settings, modem configuration range test and terminal.

Range test is of course made to test our XBee's communication range which in turn becomes handy to check the range I different locations and situations and of course we know that none of these parameters will be ideal as they are describe. There is always loss and noise involved with any RF communication system. The range will vary from different materials, just because we have a certain distance in a clear path does not mean that that will be the case for another material such as walls or any obstacle. We can check the range for any of these conditions by just using this tool. This will give us a good clear idea of what to expect once this project leaves testing and is actually implemented with real life conditions.

The terminal is used as another means for serial communication and for our purposes; we'll just use it to double check out our serial communications within XCTU.

The figure below shows the XCTU tool, we can see the tabs and the different parameters that we are able to adjust in order to tweak our XBee's settings. First and foremost our PC settings will be an important feature in case something should go wrong with our XBee module. As we know, electronic devices are sensitive and they can be damaged if we are careless either physically or internally and the XBee is no exception. We must take special care to never overload our XBee nor mess with the internals through software, should we somehow manage to do that, our XBee will end up fried or bricked, neither of which are favorable in any form. This will put us back not only in time I our design but also in budget. This is where PC settings come in handy as it can be used as a tester to see what's really wrong with a malfunctioning XBee chip. This is also a nice tool to check our baud setting to ensure they are what we desire, or in case we need to check for any random XBee chip.

The standard settings that should be set for the chip are 9600 baud, no flow control, 8 data bits, no parity and one stop bit. These should be our standard settings and they're of course displayed at the right of the PC settings. Another quick test we can do is click the Test/Query button, the test window should pop up saying some information about our XBee if this is the case, everything should check off for a fully operational chip. If this is not the case, we have a defect on our hands or we have bricked our XBee, note that there are some methods out there to help us unbrick our chip should it ever come to that.

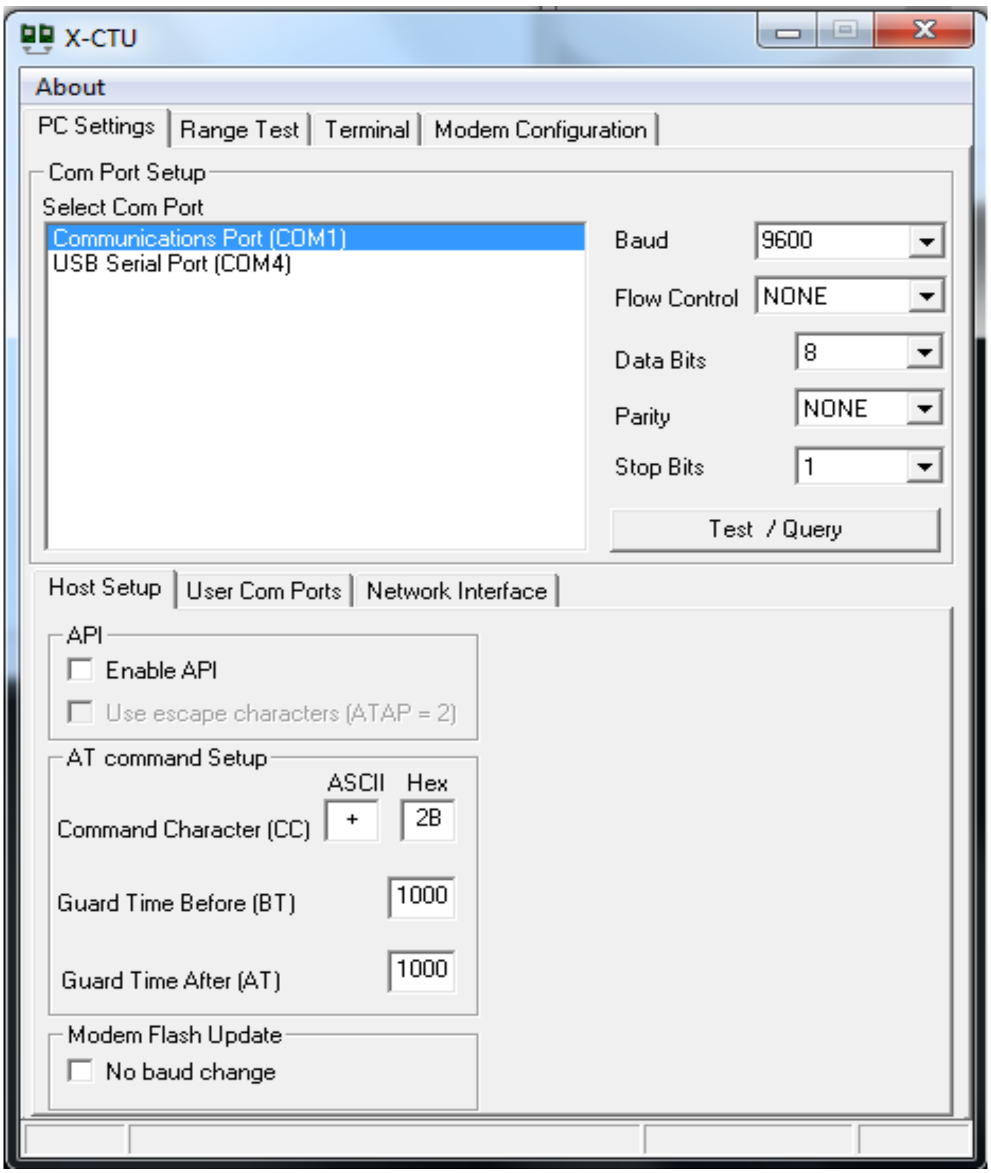


Figure 37: C X-CTU Tool

Screenshot from personal PC

We can also test and change our channel should we need to have different devices working separately that way interference and noise are all avoided all together. This is accomplished by looking in the same list that baud was in and from here we find the (FILL IN THE BLANK), click on the ID and then type in our channel of choice. Once we make these adjustments we click the write button next to the read button. It will take a few seconds but afterwards we should get some feedback that should say the following:

Getting modem type .... OK
Modem's firmware not updated
Setting AT parameters .. OK
Write parameters.... Complete.

This should confirm that we are good to go for our channel modification. And these are just some basic tests we can do to ensure that we have a good working XBee chip that way we can have some peace of mind in the fact that should anything go wrong with our design we can at least lay down the option of our XBee chip being the root of that problem.

Section 6: Administrative Content

6.1 Budget and Financing

The Motorcycle Tracking Security System is a Senior Design project that is being self funded; therefore, we have tried to keep the spending within a reasonable amount. Some of the modules required to properly build this prototype are expensive, so we are going to be as careful as possible when we deal with them, so we do not have to replace them. The table below shows the prices of the parts that we selected for the MTS system.

Parts	Description	Quantity	Cost
Microcontroller	ATmega1280	1	$20.00
Microcontroller	ATmega128	1	$15.00
GSM Module	SM5100B	1	$100.00
GPS Module	Locosys LS20031	1	$100.00
Accelerometer	ADXL345	2	$60.00
RF Module	Xbee RF Module	2	$62.00
FOB		1	$25.00
Bluetooth Module	Bluegiga WT32	1	$100.00
Battery Pack	1000 mAh Li-Po	1	$12.00
USB Mini-B	Sparkfun	1	$1.50
Recharge Chip	MAX1555 SOT23	1	$12.00
Microphone	Generic	1	$3.00
Speaker	Generic	2	$5.00
PCB board	4PCB.com	2	$60.00
Miscellaneous Parts		~	~
Bike Helmet		1	$30.00
Total			$539.00

Table 24: Budget and Finance

6.2 Timeline

In order to have a successful Senior Design Project, the way we manage our time will be extremely important. The tables 25 and 26 show how we are planning to organize our schedule for Senior Design 1 and 2.

Senior Design 1

Date	Task Description
February 11th	Research on which microcontroller would be best for the project.
February 18th	Research on GPS/GSM module for both motorcycle and helmet and its circuit requirement
February 25th	Research on which type of module to be used for the helmet to motorcycle communication.
March 4th	Research Bluetooth module to implement in the helmet and the components that are compatible with it.
March 11th	Implementation of battery pack with a recharging circuit design.
March 18th	Accelerometer implementation and programming into microcontroller.
March 25th	Alarm System Programming.
April 1st	PCB layout Design to implement all components together properly.
April 15th	Put all materials together and touch up on the last details.
April 22nd	Turn in Final copy of Research Paper

Table 25: Senior Design 1 Timeline

Senior Design 2

Date	Task Description
May 13th	Start Ordering and getting parts together.
May 20th	Start Programming each of the components and implementing them in the board.
June 3rd	Program the second board that goes into the helmet, and tune them to communicate with each other.
June 10th	Program all the components to work with the specified requirements.
June 24th	Place the board for the motorcycle in special modeled box, and place board for the helmet with components.
July 8th	Put it all together to have a working prototype
July 15th	Fix all the bugs to make sure it works to specifications
July 29th	Present working project.

Table 26: Senior Design 2 Timeline

Section 7: Appendix

7.1 Work Cited

Getting the proper information was a key factor to come out with the design of the Motorcycle Tracking Security System. These are the websites that we used to do our research.

- http://phantomtracking.com/product.aspx
- http://chatterboxusa.com/solo-wired/
- http://www.diagnosticnews.com/featured/parasitic-battery-drains/
- http://www.dimensionengineering.com/info/accelerometers
- http://www.analog.com/en/content/td_accelerometer_specifications_definitions/fca.html
- http://dev.emcelettronica.com/i2c-or-spi-serial-communication-which-one-to-go
- http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/
- http://batteryuniversity.com/learn/article/understanding_lithium_ion
- http://code-42.blogspot.com/search/label/Electronics
- http://www.sparkfun.com

7.2 Permissions

All the copyrighted images and content included in this report are pending on confirmation permission.

Andres F. Suarez (andresfelipess@hotmail.com)

1:31 PM

To: edwjim@hotmail.com, Andres F. Suarez

Andres F. Suarez

Edit profile details

From: **Andres F. Suarez** (andresfelipess@hotmail.com)
Sent: Sat 4/20/13 1:31 PM
To: edwjim@hotmail.com (edwjim@hotmail.com); Andres F. Suarez (andresfelipess@hotmail.com)

Hi, My name is Andres Suarez and I am a Senior Electrical Engineering Student at UCF. My team is building a smart motorcycle and helmet system; and we found some of your research from project SERU very interesting. Can we use your information that you posted in your SD website.

https://sites.google.com/site/ucfeecsspring2012g08/home

Thank you

Andres F. Suarez

Andres F. Suarez (andresfelipess@hotmail.com)

1:31 PM

To: edwjim@hotmail.com, Andres F. Suarez

-

Andres F. Suarez
Edit profile details

From: **Andres F. Suarez** (andresfelipess@hotmail.com)
Sent: Sat 4/20/13 1:31 PM
To: edwjim@hotmail.com (edwjim@hotmail.com); Andres F. Suarez (andresfelipess@hotmail.com)

Hi, My name is Andres Suarez and I am a Senior Electrical Engineering Student at UCF. My team is building a smart motorcycle and helmet system; and we found some of your research from project SERU very interesting. Can we use your information that you posted in your SD website.

https://sites.google.com/site/ucfeecsspring2012g08/home

Thank you

Andres F. Suarez

Wikipedia

Terms of Use

This is a human-readable **summary** of the Terms of Use.

Disclaimer: This summary is not a part of the Terms of Use and is not a legal document. It is simply a handy reference for understanding the full terms. Think of it as the user-friendly interface to the legal language of our Terms of Use.

Part of our mission is to:

- **Empower and Engage** people around the world to collect and develop educational content and either publish it under a free license or dedicate it to the public domain.
- **Disseminate** this content effectively and globally, free of charge.

You are free to:

- **Read and Print** our articles and other media free of charge.
- **Share and Reuse** our articles and other media under free and open licenses.
- **Contribute To and Edit** our various sites or Projects.

Under the following conditions:

- **Responsibility** – You take responsibility for your edits (since we only *host* your content).
- **Civility** – You support a civil environment and do not harass other users.
- **Lawful Behavior** – You do not violate copyright or other laws.
- **No Harm** – You do not harm our technology infrastructure.
- **Terms of Use and Policies** – You adhere to the below Terms of Use and to the applicable community policies when you visit our sites or participate in our communities.

With the understanding that:

- **You License Freely Your Contributions** – you generally must license your contributions and edits to our sites or Projects under a free and open license (unless your contribution is in the public domain).

- **No Professional Advice** – the content of articles and other projects is for informational purposes only and does not constitute professional advice.

Andres F. Suarez (andresfelipess@hotmail.com)

1:50 PM

To: marketing@sparkfun.com, Andres F. Suarez

From: **Andres F. Suarez** (andresfelipess@hotmail.com)
Sent: Sat 4/20/13 1:50 PM
To: marketing@sparkfun.com (marketing@sparkfun.com); Andres F. Suarez (andresfelipess@hotmail.com)

My name is Andres F. Suarez. I am a Senior Electrical Engineering Student at the University of Central Florida (UCF). I am currently working on a Senior Desing project that requires the use of Bluetooth Technology, GPS, GSM Modules, and some other electronic devices. I am writing this letter to get permission to use some of the information from your website in my paper. All my work will be cited.

Thank you for your consideration.

Sincerely,

Andres F. Suarez

Sparkfun.com