Emergency Signal Generator

By

Jee Hyoung Lee,
Shahzad Hameedi,
Young Hun Lee

ECE 445 Senior Design Project

Fall 2006
Professor Jonathan Makela
TA: Purvesh Thakker
Group # 5

Due Date: December 5th, 2006

ABSTRACT

In this paper we will discuss the many design features of our project, the design issues that we encountered, the cost of our emergency signal generator, schematic illustrations of the various modules, and many more issues that are given in the table of contents for your reference.

My lab partners and I came up with the idea of designing an emergency signal generator similar to “On-Star” which can be found in most vehicles now-a-days. However, our project utilizes the GSM networks via a cellular module and provides the services to customers such as bicycle riders, motorcycle riders, hikers, and other such consumers that do not require a vehicle to travel and do their work at a recreational level.

Our project is divided into three main portions. The first part is where the accelerometer is used to detect an impact, next there is a timer circuit that will allow us to reset our system before it sends an SOS signal to the emergency facilities. The final portion is the cellular module and this is where the module sends an SOS message through the GSM network to the emergency facilities.

TABLE OF CONTENTS

INTRODUCTION

Our group designed and developed an emergency signal generator. This system allows the consumer to contact the emergency facilities when there is a need for one. Similar to the "On Star" device that can be found in most recent car now-a-days, this device will allow the consumer with a push of a button or an impact to have there three respective coordinates with an SOS message sent to the emergency facilities. The three coordinates here are their respective latitude, longitude, and altitude. This data along with the current UTC time can easily be obtained from the GPS system incorporated in our design. Now that data has been filtered from the GPS we will be constantly updating it in the cellular module. Moreover, as soon as a hit or an impact is detected and the reset time is up, the cellular will take the last know latitude, longitude, and altitude along with an SOS message and sent it to any programmed phone number.

1.1 PURPOSE

The main purpose of our project was to gather our diverse knowledge and ideas from our previously taken courses and apply them towards our recent project. For example, we all utilized and applied our knowledge from the electronic circuits' course. Additionally, Jee Hyoung Lee and Young Hun Lee helped with their management skills and I helped with the programming portions of the project. Towards the end of the course we all learned an important lesson that time management, practicality, and the scope of the project are very important factors to consider when designing a device. Our project was a bit more ambitious than we expected and required much more time than we had anticipated. Therefore, we were not able to complete the whole project and last minute changes were made to the design to obtain some achievements.

Getting back to the design purpose, after accumulating the management skills and having a pretty decent programming background we were able to divide the project into three parts. The accelerometer and comparator circuit, the timer circuit, and finally the PIC and cellular module circuit. Initially, we came up the PIC design but after extensive research on the cellular module, we determined that if we use the cellular module we would not require the PIC because the module has its own memory RAM and ROM which is Read Only Memory and Random Access Memory, therefore eliminating the use of the PIC.

Upon designing this device all three of us spent a very large amount of the time trying to figure out what this product will ultimately do and how to improve in innovating engineering design in general. After a great deal of effort we all came up with the design and reasoned with the fact that this devices is not yet available for everyday consumers like we have mentioned in the introduction.

1.2 SPECIFICATIONS

Our project was mainly split into three main modules and five sub-modules. For your reference you may find the diagrams of these modules and sub-modules in the appendices at the end of the report. More about these modules is stated in section 1.3.

The emergency signal generator as a whole is required to send an SOS message via a cellular phone which uses a GSM SIM card to unitize the cellular network. Among the few networks out there we decided to use T-Mobile only because I had an extra SIM card available for this network. The Telit GM862-Quad-PY cellular module, a picture of which is illustrated in the appendix D, is programmed in Python. Python is a scripting language, similar to perl that is used for communication tools. We were under the impression that the module can be programmed in C instead of Python. Learning the new language was not anticipated as part of the schedule hence delayed the over all design of the project. As a result we were not able to complete the project on time. As we mentioned earlier that this project can be completed given that we had more time to complete it.

The DE-ACCM3D accelerometer that we used reads a three axis coordinates. This means that hits can be analyzed from three axes not only from a straight hit. After the hit is read it is sent to the comparator circuit which with a reference voltage compares the hit and sends the signal to the timer circuit. The timer circuit is triggered on pin two and we can vary the time of the circuit by using the following equation:

$$T = 1.1 \times R \times C \tag{1.1}$$

where T is the time that we can set, R is the resistance, and C is the capacitance. If we want to delay to be one minute we can set T equal to one minute and solve for a particular combinations of resistance and capacitance. If the delay time is up and the reset switch is not toggled then the signal is sent to the PIC telling it to store the last location and send the information to the LCD display and to the cellular module.

1.3 SUBPROJECTS

Our design, as mentioned before, was split into three main modules however there were other various sub-modules, these main module and sub-modules are explained in more details below:

1.3.1 POWER SUPPLY

For our project we used a 9-volt DC battery. This main battery supplied a constant +5V, +3.8V, and +3.0V with respect to ground. Additionally, we had to make sure that the correct voltages are applied accordingly. For example, the PIC Module requires a +5V DC power supply, whereas, the Cellular Module only takes a max of +3.8V DC. Therefore we used voltage dividers circuits in our design to obtain this goal.

1.3.2 ACCELEROMETER MODULE

The accelerometer module of the circuit is the main part where the hit or impact is detected. This module can detect a three axis hit from the xyz coordinates. This is a great feature because the hit that is detected is not required to be a single axis hit or a linear hit.

The accelerometer is a very sensitive device and needs to be handled with care. Just to give you an idea, its sensitivity is registered as 360 milli volts per one G, which is a small voltage hence making this a very sensitive accelerometer. You can find the schematic diagram of this module in the appendix B.

1.3.3 COMPARATOR MODULE

The comparator module is where the hit is compared to a reference voltage. These reference voltages are labeled as V_{refX}, V_{refY}, and V_{refZ}. Now the reference voltages have a range from low to high. For example, V_{refX} and V_{refY} range 2.6 Volts, 2.8 Volts, and 3.0 Volts refereeing to a low, medium, and high hit respectively. Moreover, the z reference voltage differ such that the range is as follows; 2.67 Volts, 2.87 Volts, and 3.07 Volts. You can find the schematic diagram of this module in the appendix B.

1.3.4 LED UNIT

Once the signal is compared then the output is sent through the OR-Gate logic and a particular set of LED's are lit. For example, if the hit is low then the Green LED's are lit, if the hit is a medium one then the Green and Yellow LED's are lit, and if the impact is high then all three set of LED's are lit (Green, Yellow, and Red).

1.3.5 TIMER DELAY MODULE

Now that an impact is determined then the signal triggers a timer circuit. This module is used to establish the delay in the circuit. Let us suppose if there is no delay in the circuit then as soon as a hit is established the SOS message will be sent. This is not a good design because what if the hit is low we do not want to send an SOS message unless it is really an emergency. Therefore if there is a real emergency a person will not be able to reset the switch and ultimately letting the SOS message to pass through and sent to the emergency facilities. The triple five timer circuit uses pin two as its trigger and once this pin is triggered then depending on the resistance and capacitance the time can be varied. Pin four is the resetting pin of the integrated circuit therefore, the toggle switch is attached to pin four and ground. More about the schematic of this module can be found in the appendix B portion of the report.

1.3.6 PIC AND GPS MODULE

We used a forty pin Microchip PIC16F877A enhanced flash microcontroller. This microcontroller is a suffocated device. It is basically a mini computer that can hold data in through various ports and can serially display or sent data through other ports. The input/output ports are ports A through E. This chip has thirty five instruction set, two analog comparators, eight input channels for analog to digital conversion, parallel communication is also available, and much more. You can find a schematic diagram of this PIC in the appendix B, along with the functional port illustrations. We used a Garmin

GPS V unit. This unit can be linked through the serial port to a computer and we can read the incoming data via a program called Hyper Terminal.

1.3.7 LCD DISPLAY MODULE

In our project we used a BPI-216 Serial LCD Module. The great thing about this LCD display is that it communicates serially and can be interfaced with the PIC microcontroller via its serial communication ports. Programming the LCD is also quite nice since it can be programmed in C, which I am very familiar with. In general, the LCD display reads whatever is sent to it one by one. For example, if you send the word "HELLO" the LCD will display "HELLO", hence the serial communication. This is a two by sixteen LCD module, which means it has two rows and sixteen columns that we can utilize for the display.

1.3.8 CELLULAR MODULE

For our project we used a Telit GM862-Quad-PY cellular module. This too is a very sophisticated machine. It is basically a cellular phone. One only has to know how to program it. The programming is the hard part because it is something a group, team, or individual has to learn on their own. The cellular module is only programmed in Python, which is a scripting type language similar to Perl. Like all cellular phones this module has its own RAM and ROM so we can eliminate the use of a PIC in our design. Additionally, all the headphone input/outputs, mic output, LED status light, on/off button, communication ports either UART level and RS232 level, speaker, general purpose input/outputs, and much more is available with this module. However, it is very crucial that the proper kit is used because any mistake in integration can ruin the project.

DESIGN PROCEDURE

2.1 POWER SUPPLY DESIGN

For our project we used a nine volt Industrial Energizer Battery, the reason for this design as such is because our project need to be mobile therefore we cannot attach it to any kind of stationary power supply. Moreover, since many parts in our project required different levels of power we decided to use a more stable nine volt battery instead of a conventional for 1.5 volt AA batteries.

Next we needed to supply a constant regulated +5V, +3.8V, and +3.0V with respect to ground to various portion of the circuit. For example, most of the circuit required five volt for operational, however, the cellular module is a sensitive device and we were required to have some pins powered to +3.8V and other pins to +3.0V with respect to ground. To establish a good and constant power supply to the circuitry we used the most commonly known LM317 regulator. This allowed us to supply a constant five volts to the circuit and where we required a lower voltage we used a voltage divider circuit to accomplish this task.

2.2 COMPARATOR MODULE DESIGN

As we mentioned before that our project is divided into three major module and five sub-modules. Here we are going to discuss the major modules and their design procedures. Designing the major modules was a crucial task because each module was dependant on the other module to work properly.

We began our design by initially understanding that the signals from the accelerometer were to somehow be compared and since we have three comparisons (low, medium, and high) we required a sophisticated technique to achieve this goal. We knew that we will have a signal coming in from the accelerometer, which is a three-dimensional hit, once the hit is registered it is sent through a LM324 operational amplifier where the signal is compared. What happens is that if the signal from the accelerometer is greater than the reference voltage the LM324 IC outputs a V_{CC} voltage otherwise it will output a zero. The high and low or logically put one and zero are compared in the logic gate unit which is the output of the amplifiers. In our design the amplifiers are not used to amplify a signal but rather as a comparison tool.

One of the design challenges that we had encountered was with the accelerometer design. The accelerometer calculates impact and tilts which means that if we tilt the accelerometer at a certain angle we would generate a hit, which is a challenge because it has to be mounted correctly otherwise the initial voltages change and hence we would have to reconsider the whole design. We mounted the accelerometer such that the unit would experience an initial one gravitational pull per unit force. We had our initial voltage as X_{Out}=Y_{Out}=1.66 Volts and Z_{Out}=1.99 Volts. This is the exact particular reason why the V_{refX} and V_{refY} ranges differ from the V_{refZ} range.

The output of the high reference hit is sent to a D flip-flop because this signal is the one that will trigger the timer module. The reason we decided to use a D flip-flop for only the high hit is because we only wanted to trigger the LM-555 timer only when a high hit was received, otherwise the LM-555 will not be triggered and the signal will not be sent for the low and medium hits. Initially we had not thought about designing our circuit like this but as we got more involved into the design we decided that this would be a better design since it eliminates the lower two impacts from sending the signal to the emergency facilities. Now if the impact is high the timer will be triggered and if the user feels that they do not want to contact the emergency facilities they have an option to reset the system via the LM-555 timer.

2.3 TIMER MODULE DESIGN

The benefit of this part of the design was that it would allow the customer the opportunity to reset the device in case the hit was high but an emergency facility was not required. We had to consider many scenarios before we incorporated this module into our design. For example, if the driver of a motorcycle was driving at seventy miles per hour and a truck ahead of the motorcyclist is at a velocity of sixty five miles per hour. Now right before the motorcyclist is about to over-take the truck a rock propels from the truck and hits the accelerometer on the motorcyclist's helmet. Well, it is obvious that the impact will be high with a magnitude of 135 miles per hour. However, such debris is not really harmful for an average motorcyclist unless it hits them in the eye or somewhere sensitive. Keeping this fact in mind that an average person is going to be using this device it was imperative to have a reset switch as part of our design.

The LM-555 IC is a very commonly used timer IC and is very easy to incorporate into many designs; therefore, we decided to go with this IC instead of other similar designs out there in the market. A detailed design of this circuit is given in the appendix B.

During the design of this part we encountered challenges such as alternating the output so that the reset can take place. The LM-555 IC is high at its initial triggering state so when the resetting takes place we have to bring the next state to a low for that to occur. This was a challenge and what we did was used a 74F157A Mux and inverters to obtain this goal. Once this was part was operational we then went on to the programming of the cellular module.

2.4 CELLULAR MODULE DESIGN

The most difficult portion of the project was the cellular module. For our project we used a Telit GM862-Quad-PY cellular module, which is basically a cellular phone. This fifty pin kit has everything from have the off hand headpiece connection to the speaker connection and video camera connection. I mean really, it is a very nice piece of equipment if it can be programmed properly and if the popper complete kit is available. The module has its own Read Only Memory and Random Access Memory, have its own memories allows the user to eliminate the use of an additionally microcontroller as we had in our initial design.

The module operates in layers, commands are initially programmed into the ROM which is a flash type memory but once the ROM is programmed the commands are internally stored and commands are directed to its proper locations when the device is turned on. It has its own status LED that would turn on when the machine is operational or performing a task and an on/off LED when it is on or off. When we were testing this device the status LED was blinking indicating that the device was working but was not able to communicate to the computer.

Moreover, the module has built in libraries which are upgradeable, there are features like built in print commands, MDM libraries, SER libraries, general purpose input and output libraries, MOD libraries, the GPRS modem engine that connects to the GPS satellite, and much more. A pictorial of the internal modules along with the coding that we used to program the kit is given in the appendix D.

The good thing about Python is that it is on open source programming language similar to Linux that is also open source so we do not need to purchase a license for it. Python is dynamic, object orientated, can easily be integrated with various tools, has its own standard libraries, and much more; However, the kit that we ordered was not a complete one and the one we really need was out of stock. Therefore, initially we decided that we would purchase this lower end kit and engineer the other components onto it, and eventually turning it into the complete kit. We did extensive testing with this tool and were not able to engineer it as it would have been with the complete kit.

We were however able to get the lower end kit to turn on and off, get the GPS incorporated into the design with the cellular module instead of connecting it to the PIC, get the popper power to work, and the switching to work completely. About the only thing the module was not able to do was to communicate with the computer this we think was so because the serial RS232 port from the cellular module was sending the proper signal out but were not being received at the computer end. Now this could have been the programming fault, cable fault, or some sort of computer setting fault, but in short we were not able to get it operational in time for the demo.

The kit that we really needed for this project was the one that came with an internal GPS system in it, a universal serial port, a power cable and supply that properly powers this sensitive equipment, an antenna, and the cellular module itself. So off the many things we have mentioned just now we were only able to obtain the cellular module, a proto-board, and an antenna for it.

The Telit GM862-Quad-PY cellular module is only programmed Python and cannot use other language as a programming tool. This was difficult because learning a new language along with taking on other challenges was a difficult task. Towards the end we had no choice but to give up on the cellular module and have the PIC and GPS system incorporated together along with the LCD displaying the data that we would receive from the GPS.

DESIGN DETAILS

3.1 POWER SUPPLY

We used a nine volt power supply because it was the most optimal way to supply our circuit with a constant regulated +5V, +3.8V, and +3.0V with respect to ground to various portion of the circuit. The main reason for having a battery source instead of using an AC to DC converter was that our project has to be mobile and therefore requires a mobile source of energy. That is why we decided to use the battery instead of the AC to DC converter.

3.2 ACCELEROMETER MODULE

The Pspice simulations from the accelerometer are given in the appendix C. At first, we were not sure which accelerometer module we had to use because there are a variety of them in the market. We just wanted to use some module which can detect impacts. As we research for the accelerometer we found there was an accelerometer that can react with our required type of impacts. We needed to choose between an accelerometer with analog outputs or digital outputs, number of axis, maximum swing, and their sensitivity levels.

We had to use mathematical equations to decide which accelerometer would work with our project. After many discussions we figured out that gravity for the reference voltage can be solved with the regulated 3.3V supply and an X_{REF} = 3.0V, which indicates the emergency situation. Using equation 3.1 below:

$$\text{The gravity acceleration for x-axis} = 3.0V - 1.66V = 1.34V \quad (3.1)$$

We can solve for the sensitivity at 3.3V which is 333 milli volts per one gravitational pull. 1.34V / 0.333V/g = 4.024g, which is the gravitational pull of our accelerometer's x-axis.

Moreover we obtained the Y_{REF} = 3.0V, which indicates the emergency situation for the y-axis. Using equation 3.2 below:

$$\text{The gravity acceleration for y-axis} = 3.0V - 1.66V = 1.34V \quad (3.2)$$

We can solve for the sensitivity at 3.3V which is 333 milli volts per one gravitational pull. 1.34V / 0.333V/g = 4.024g, which is the gravitational pull of our accelerometer's y-axis.

For the z-axis we obtained a different value due to the initial voltage outputs of the z-axis being different from the x and y axis's. Our Z_{REF} = 3.07V, which indicates the emergency situation for the z-axis. Using equation 3.3 below:

$$\text{The gravity acceleration z-axis} = 3.07V - 1.99V = 1.08V \quad (3.3)$$

We can solve for the sensitivity at 3.3V which is 333 milli volts per one gravitational pull. 1.08V / 0.333V/g = 3.243g, which is the gravitational pull of our accelerometer's z-axis.

If we got an accelerometer which can detect higher gravity acceleration, we could have tested real emergency situation, however, in our case, we used the accelerometer as a demonstration, so we got an accelerometer which can detect small gravity accelerations, however it was more sensitive.

The way we chose reference voltage was we mounted an accelerometer on a toy car and put it on an incline plane. Then we let it slide and measured the output voltage from each axis depending on different angle. When the angle was about 20 degree, we set it as a reference voltage for the emergency.

The measured voltages are given in the appendix C. Unfortunately, the z-axis did not wok when we tried to measure the output voltage using oscilloscope. We had burned it out while testing for soldering the module on the mount that we used.

3.3 COMPARATOR MODULE

The output of the comparator is shown in the appendix C. After we measured the reference voltage, we used the voltage divider rule to generate a reference voltage.

For the xyz-axis's we used the equation below:

$$Vin \times (R2/(R1+R2)) = Vout \tag{3.4}$$

Case X:
$5V \times (R2/(R1+R2)) = 3V$
$5R2 = 3R2 + 3R1$
$R2 = 1.5R1$

Case Y:
$5V \times (R2/(R1+R2)) = 3V$
$5R2 = 3R2 + 3R1$
$R2 = 1.5R1$

Case Z:
$5V \times (R2/(R1+R2)) = 3.07V$
$5R2 = 3.07R2 + 3.07R1$
$R2 = 1.59067R1$

Following table is for the matched resistor values,

Axis		Red	Yellow	Green
X	R1(KΩ)	1.0	1.62	1.5
	R2(KΩ)	1.5	2.0	1.62
	Vref(V)	3.0	2.8	2.6
Y	R1(KΩ)	1.0	1.62	1.5
	R2(KΩ)	1.5	2.0	1.62
	Vref(V)	3.0	2.8	2.6
Z	R1(KΩ)	1.5	1.5	1.3
	R2(KΩ)	1.62	1.62	1.5
	Vref(V)	3.07	2.87	2.67

In our design, we assumed that there is an accident if one of the axes from the accelerometer detects an impact. Using OR gate, each output of the comparator is connected to an input of an OR gate so that OR gate output high pulse when one of the axes from the accelerometer detects an impact.

3.4 LED UNIT

The LED's are just to show the intensity of the impact. If the intensity of the impact is low, medium, or high, the intensity is represented by green, yellow, and red LED, respectively.

3.5 TIMER DELAY MODULE

The output of comparator is connected to the clock of D flip-flop. Once the clock is triggered the D flip-flop outputs high pulse to the 555 timer since the input of D flip-flop is connected to Vcc at all time the inverted pulse triggers the 555 timer and the pulse stays there for a certain time that we can adjust using equation

$$T = 1.1 \text{ X } R \text{ X } C \qquad (3.5)$$

In our design, we set it to 7.5 sec delay using a ten mega ohm resistor and one 0.68 micro Farad capacitor. The output timing pulse can be adjusted from approximately one millisecond to as high as on hundreds seconds. There is actually no theoretical upper limit on the time, only practical ones. The lower limit is 10us, upper limit can be infinity.

When we tested the exact time for the 555 timer, the time was very close to 7.5 sec. While the pulse stays in the 555 timer for 7.5 sec, if a user reset the circuit, the entire circuit goes back to its initial state. If not, the 555 timer output a high pulse. Using an AND gate, we can output high pulse when reset button is not switched on.

In our 555 timer circuit, the trigger input is initially high. In the data sheet, it says that the 555 timer will generate its single-duration output pulse when negative going trigger pulse is applied to the trigger input. However, in our test of the 555 timer, the trigger input should be applied by positive going trigger pulse after negative going trigger pulse is applied. So, we manipulated to generate positive going trigger pulse using one MUX and time delay of output of 555 timer using a few inverters. We were able to generate one clock cycle for the trigger input of the 555 timer. The measured cycle is shown in the appendix C.

3.6 PIC AND GPS MODULE

We did not have much choice in selecting either one of these modules because it was what was available to us from the laboratory and the ECE store. The standard forty pin Microchip PIC16F877A enhanced flash microcontroller was available at the ECE store. Moreover, it was the most convenient microcontroller since most of its tutorials were given to use on the ECE 445 website.

The microcontroller was programmed in C with the use of MPLab IDE software and a PICC Complier along with the Smart-Start Emulator that hard codes the written software onto the IC. The code that was used for programming the IC is given in the appendix C. Please refer to the appendix C for further details about the programming. The programming has comments that are self-explanatory about the operational use of the program.

The PIC can also be programmed in assembly however; from my knowledge of ECE 190 it was evident that an optimal choice for programming the PIC would be C instead of assembly, this information is just so the reader understands that the PIC can be programmed in a variety of method. As engineers one of the many lessons learned are that abstraction is important when programming devices, what level are we required to work at? Do we really need the unnecessary information or not? From this we can make a decision on which program is better to use.

We programmed the PIC such that it was able to obtain data from the GPS system and when an impact was determined the PIC would store the last received GPS data and display it via output port to the LCD display. More on the LCD display is explained in section 3.7.

The GPS that we used for our project was the Garmin GPS V unit and this unit can be found in the ECE 445 laboratory for any student taking this course. The GPS unit has many great features like about ninety-five percent location accuracy, interfaces easily with the computer via the RS232 serial port, uses 4 AA batteries for twenty-five hours (great consumption time), and much more.

The GPs receiver data can be extracted from the unit in the form of NMEA, which is an acronym for National Marine Electronics Association standard through the data out PM. The NMEA protocol is based on ASCII character being sent along the wire at 4800 bits per second or about 600 characters per second. The $ symbol indicates the beginning of each new data received and is followed by a GP with indicates that the signal is coming from a Garmin GPS unit. The next three letters describe the type of data is being sent. For our case we are more concerned with the ($GPGLL), which represents the geographical positions latitude and longitudes.

We used the Hyper-Terminal of the computer to read what the GPS unit was reading and then once we got all the data on the Hyper-Terminal we were able to use the GPS unit data-sheet and determine which data from the set was really required from the system. Once this was established, we programmed the PIC such that only the required data from the GPS was read and stored in the PIC, and later displayed on the LCD display. This was a great challenging in the beginning because I thought that the LCD could only be programmed in Basic Stamp however, after studying the data sheet for the BPI-216 Serial LCD Modules it was evident that C and be used to program this LCD, hence the use of abstraction skill and engineering decisions.

3.7 LCD DISPLAY MODULE

The LCD display is a two by sixteen matrix, which means it has two row and sixteen columns that the user can utilize for the display of information. The LCD was easily incorporated with the PIC because the PIC can send serial data through its output ports and the LCD can display serial data very easily. For example, if data is sent, one by one the data will be displayed on the unit, just like first come first serve method. However this happens so quickly that the display seems to have everything in a quick moment at a blink of an eye.

The LCD display has a potentiometer on its reverse side that lowers the contrast of the display. It is very important to have this in the middle range initially so one can see if the display is operational. There is a simple test that can be performed to achieve this goal. Once we have established that the LCD display is functional we connected it serially to the PIC and began programming and testing the PIC. After many hours of hard we were finally able to obtain an output on the LCD that would be considered satisfactory for the demo.

3.8 CELLULAR MODULE

The cellular module that we used for our project was a Telit GM862-Quad-PY cellular module. The decision to use this particular device was recommended by the professor and by our TA. I must admit that the idea is very elegant however, it was not something that we anticipated to be so difficult to program. Originally we were using a transmitter and receiver circuit to perform our task. However, after our design changes we had to order the cellular module and incorporate that into our design.

The cellular module is powered by a 3.8 voltage source which is usually the voltage supply to most cellular phone. Since these cellular phones are made in large quantities and with sophisticated machinery we were not able to achieve the same goal as accepted. The module utilities the GSM networks via the use of a SIM card that is inserted into the given slot. There are fifty pins on the module proto-board; some of these pins are output pins while others are input pins. For example, there is a pin for the headpiece output, there is a mic input pin, there is keyboard input and output pins, camera pins, and much more that can be attached to the module.

DESIGN VERIFICATION

Testing is a very important part of designing any sort of engineering project. If we design something we must first have an idea how we would test this device or gadget, otherwise, we will not really have engineered something that is practical or useful. In a nut shell, for our project we wanted the circuit to able to read a impacts, light up the LED’s and send the SOS message over the wireless network. Not everything mentioned here was accomplished; however, there were alternatives that were taking when other routes were failed.

4.1 TESTING

Our main purpose from testing was that we wanted to make sure all our modules were operational separately and work together when integrated into a system. Many tests were performed to accomplish this task, the details of which are given below.

4.1.1 ACCELEROMETER MODULE

We were able to test the accelerometer by hitting the module with a stick and obtaining particular waveforms as a result of the intensity of the hit. These waveforms can be found in the appendix B of the report. Please refer to the appendix C for more details on the waveforms. For example, a typical waveform would have a peak of about four volts and a period of five hundred micro seconds.

4.1.2 COMPARATOR MODULE

The comparator circuit was tested by the logic analyzer, we had incorporated the circuitry together and once a hit was performed we observed the outputs of the logic gates of each set of intensities (low, medium, and high).

4.1.3 LED UNIT

We tested the LED by first hitting the accelerometer and observing whether the LED's lit up. Once this was determined we then tested the LED's for precision meaning that if a small hit was perform we observed if only the green LED's were lit or not and if a great hit was performed then the green and yellow LED's should be lit. Finally, if we performed a hard hit all three set of LED's were to be hit. We were not surprised to see all the LED's performing proper operations. Occasionally we had an LED not lit up but this was due to either over load of current which ultimately burned the LED. As a result, we decided to regulate the supply voltage and thus eliminating the problem of burning the LED's.

4.1.4 TIMER DELAY MODULE

The timer circuit was tested in various ways, theoretically the circuit is suppose to send a signal to the PIC after a particular time which is determined by the equation 3.5. So we can set the capacitance and resistance according to this formula. Moreover, the charge time of the capacitor is sixty-three percent to the total determined time. This too is a theoretical value. So, we tested the circuit by timing it with a stop-watch and figured out that our theoretical value that we had set coincided with the measure stop-watch time, which was nine seconds for our project. We could have made the delay for a longer or shorter period as desired but this was just a trial project and if the device was to go into mass production a precise time can be set.

4.1.5 PIC AND GPS MODULE

The PIC was tested with the use of the LCD display, since our LCD display was outputting the correct locations and time on the display this concluded that our PIC was programmed properly and was able to communicate with the LCD serially via pin number twenty-one on the PIC and the serial input of the LCD display. Another test that we had done to see if the GPS module was working with the PIC microcontroller was we used the output ports as mentioned earlier reports which are a group of eight pins. In our case these pins were pins twenty-eight to pins thirty-six. We then attached these pins to eight LED's and observed that when the PIC and the GPS unit were powered the LED's would blink synchronically with the data that the GPS was outputting on the Hyper-

Terminal. Since both were synchronically working together we can conclude that the PIC was programmed properly and the GPS was send data to the right pins.

Next we had to basically re-program the microcontroller such that the test pins would not send the signals and only pin twenty-one was sending the data serially to the LCD. Initially we used a LED to test this part and as we slowly improved our code by adding new code and omitting other parts of the code we got the LED on pin twenty-one to blink. Lastly we had to filter the data that the GPS was sending and only display the required information on the LCD display. The Hyper-Terminal was a great tool to observe the output of the GPS and we were able to achieve the goal of eliminating unnecessary information from the incoming data and only have the required data display on the LCD.

4.2 CONCLUSIONS

Even though our project did not work completely we were satisfied with the amount of work that we had accomplished. We had gotten the PIC to communicate with the GPS unit and have the proper data from the GPS displayed on the LCD. This was an achievement given the scope and the complexity of our project.

Moreover, even though that the cellular module was not able to work properly and communicate with the Hyper-Terminal we were able to turn the module on and off indicating that the module was programmed properly but was not able to communicate with the computer for testing.

COST

For our project we have had various estimates as our design got more sophisticated and elegant we had to increase some units and eliminate others. For example, the PIC was initially part of the design, later it was eliminated when the cellular module was carefully studied, and finally when the proper kit was not provided the PIC had to be incorporated into the design. This varied the cost of project from time to time but our final cost of parts and labor is stated below.

5.1 PARTS

Our project constituted of various parts that we had to either order from websites or purchase the items from the ECE store in the basement of Everitt Laboratory. A detailed list of these parts is given in the appendix E. Please refer to Table 1 in the appendix E for details. The cost for the whole project excluding the labor cost and the enclosures is $364.97.

5.2 LABOR

After many arguments we concluded that our salary on average would be approximately fifty dollars per hour. This was the exact amount that we proposed at the beginning of the semester during our proposal. Additionally we believe that an estimated time of about a hundred hours would be spent on this project. Since we are three engineers working together we multiplied that by three. Therefore our formula to calculate the labor is give by equation 5.1 below:

$$\text{Labor} = \$(50/\text{hr})\times 120\text{hours} \times 3\text{Engineers} \times 2.5 = \$45{,}000 \qquad (5.1)$$

Therefore the total cost for the project is give by equation 5.2 below as the sum of the Labor and the parts:

$$\text{Total Cost} = \text{Parts} + \text{Labor} = 364.97 + 45{,}000 = \$\ 45{,}364.97 \qquad (5.2)$$

This is a very expensive piece of equipment. However, we have estimated that if we were to mass produce this device the cost of an individual device would be lowered significantly. Keep in mind that by mass productions we are referring to production of the gadget in hundreds of thousands. Given this condition we have estimated that the individual cost of a device would be approximately $223.67. The more we mass produce the product the lesser the individual cost, it is all part of the economic theory from ECON 102 that we applied here.

CONCLUSIONS

In conclusion we were a little happy that the major part of our project was working even though all of it was not operational. This was mainly due to the fact that we did not receive the proper kit on time. Whatever we did get operational we had working properly, that is why we think that given enough time we would have been able to complete the whole project. The accelerometer was reading everything properly, the timing circuit was delaying the switching as we calculated and the PIC that we programmed was also working properly. Just the cellular module was not working correctly and the reasons for that were explained earlier.