Auto Anti-Collision and Parking

A Major Qualifying Project submitted to the faculty of the

Worcester Polytechnic Institute

Electrical and Computer Engineering Department

in partial fulfillment of the requirements for the

Degree of Bachelor of Science

30 May, 2007

Michael J. Leferman
Leferman@WPI.edu

James A. Yasuhara
Yasu4303@WPI.edu

Chris M. Ceccarini
CCecc85@WPI.edu

Project Advisor: Professor David Cyganski

Project: DC-0601

Abstract

This project developed an automobile object detection system with after-market installation features. A sensor array installed on the front and back bumpers of a vehicle will wirelessly transmit readings to a display inside of the vehicle. The project was made with cars and SUVs in mind, but can be used for other applications such as remote robotics and boating. The display inside of the vehicle was designed to warn the driver without being a distraction. The device includes ultrasonic ranging, Zigbee wireless, rechargeable switching supply, embedded controller and display subsystems.

Executive Summary

As vehicles get larger and cities become more crowded, knowing what objects are around your vehicle is more important than ever, and more difficult to determine. Some vehicles come equipped with object detection systems built in, but many vehicles, new and old, do not. Current after-market systems require running wires through your vehicle, a long complicated process. In this project, a device was developed that would ultrasonically detect objects, wirelessly transmit this information to a display unit inside the vehicle and alert the user to the surrounding objects. When available, off-the-shelf modules were used to reduce the cost and time necessary to develop the prototype, such as the Zigbee communications modules. Ultrasonic impulse-response sensors were used in a custom designed circuit to detect objects as this met our design criteria for all weather operation and low power consumption in the remote, battery powered subsystems. A console mounted, car powered embedded processor and display system received the data from the remote sensors and presented a graphical depiction of the automobile's environment. The prototype successfully transmitted data wirelessly, displayed that data on an LED display and recharged Nickel-Metal Hydride batteries. Unfortunately, the final ultrasonic sensor subsystem, implemented on a custom PCB failed to send a pulse to be reflected by object, though it had previously worked during breadboard testing. The current revision of the prototype could be commercialized using off the shelf ultrasonic subsystems, or the low power sensors developed for this project could be finished with further troubleshooting.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Right now there is a lot of research being done on fully autonomous vehicles. The idea for our project spawned from the desire to make a semi-autonomous vehicle one which was primarily controlled by a person, but in certain situations the vehicle would be able to take over and avoid hazards. The primary function of this system would be to avoid collisions. To accomplish this we must design sensor modules and an interface module.

For the sensor module we must choose an object detection method. Our most immediate and well investigated low cost options are either optical or ultrasonic. The primary differentiating factor between the sensors is that if we deal with an optical solution we must be able to deal with different ambient light situations. With the ultrasonic detection system we would be dealing with very specific audio frequencies at specific times, making it a more desirable option from the perspective of interference from outside sources. In order to control the ultrasonic sensors we will need a controller that sends out pulses, and interprets the received reflected signal. Because this sensing module would be on the outside of the vehicle, it will need to wirelessly communicate with the in-car module. Aside from a wireless communication device the sensing module will also need voltage controllers that provide the appropriate voltages required by the wireless device, the sensors, and the sensor controllers.

The in-car module will be the user interface. We had originally planned to make the user interface automated so that our device would automatically take over the vehicle's control. However, we decided that a driver might not be comfortable with a car that took over control with zero warning which it would have to do if it were avoiding an object. So, we decided to just make it a warning system. With just the warning system this object detection system could be made to be an aftermarket device, and could be attached to any vehicle.

With this new goal in mind we would have to make a display system that shows the location of objects being sensed by the sensor system. This display would ideally be low power, low cost, and easy to see. We believe an LED system would suit this purpose well. This display system would require a controller, a multiplexer, and a series of LEDs. This module will also require a wireless communication unit and voltage regulators.

This system is now supposed to be aftermarket, so in order for it to be easily reasonably installed by a user it should not require tapping into the main power of the vehicle. The in car unit can be powered by a cigarette lighter jack, but the external sensor modules will require battery power. To make it more convenient we would also like the battery powered units to have built in battery chargers.

Figure 1.1 depicts the layout of the sensors on the vehicle and the display inside of the car. The figure on the left shows a child running in front of the front right sensor on the vehicle. This figure also shows how the eight sensors would be laid out on the car. The right figure shows what the driver would see inside of the vehicle. The corresponding LEDs for the front right sensor are dimmed to represent the object.

The task before us is now to create an aftermarket sensor system that will detect objects, and wirelessly transmit that information to a display unit inside the car. The tasks are essentially broken into wireless communication, object detection, and power. Each of the

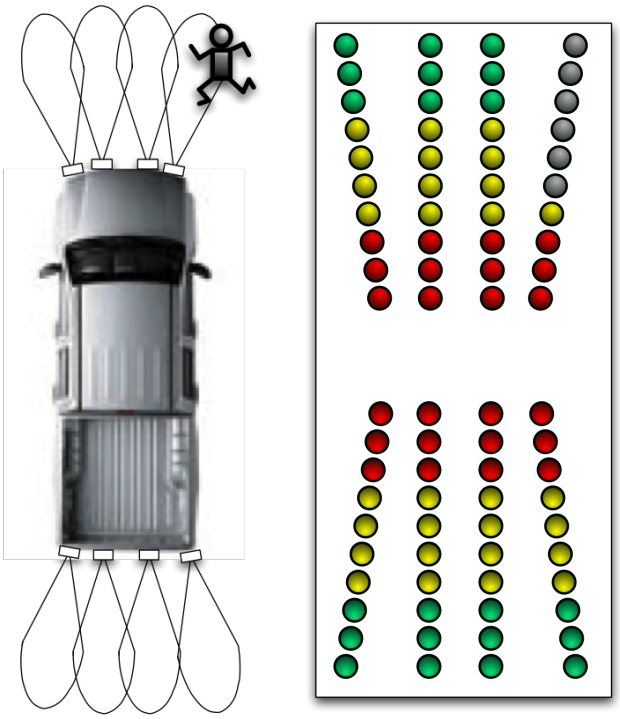


Figure 1.1: Sensor Layout On Car and Corresponding Display

members in our group will be assigned to be the primary designer of one of these tasks. Because this is a group effort there will of course be much collaboration in order to ensure proper cohesion of all of the parts to form a single working product.

Chapter 2

Design

The design processes we undertook led to decisions further defining the device specifications. These decisions were then used in the implementation process. The overall system, as seen in Figure 2.1, consists of three parts, one display module central to the system, two communications modules wirelessly connected to the central module and an array of sensors connected to each communications module.

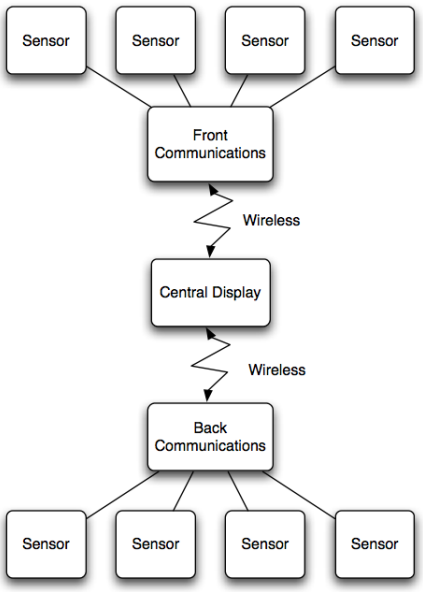


Figure 2.1: System Diagram

2.1 Object Detection

The nature of the application ruled out physically touching the object to be detected. Some sort of wave would have to be reflected off the object and used to measure the distance to the object. Three types of waves were investigated to measure distance, acoustic, light, and radio waves. Small targets would require a very high radio frequency to be detected, resulting in complex, perhaps power hungry circuitry. This ruled out the use of radio waves for object detection in our project. A light based system was a strong possibility, with low power requirements, low complexity, and a large enough range for the application, but this option was ruled out because of poor reliability in inclement weather. Range finding using audio waves would be inexpensive, had the range we desired, inexpensive and would still function in poor conditions.

Echo time sonar range finding works by transmitting an acoustic pulse while starting a timer. The signal bounces off an object in range and returns to the receiver. When the receiver detects the signal, the timer is stopped. The measured time (t) is put into Equation 2.1, using the speed of sound(S), the distance (D) can be solved for. Dividing by two takes into account the time taken both two the object and returning. The units of the distance will match the units the speed of sound is in, both the time and the speed of sound must be in the same units (seconds).

$$D = \frac{S \times t}{2} \tag{2.1}$$

Echo time was chosen over a Doppler method because it can detect stationary objects. Doppler methods measure a change in distance, requiring either the object or the sensor to be moving and only measuring the change in distance. The method used will detect both moving and stationary objects and will report an absolute distance.

Acoustic range finding uses a transducer to convert electric signals into an acoustic pulse and the transmitted acoustic pulse back into electric signals. Sound waves are considered ultrasonic when they have a higher frequency than than the human ear can detect, that is, above 20 kHz. Two types of ultrasonic transducers were investigated, piezoelectric and electrostatic. Electrostatic transducers are slightly more complex to employ than piezoelectric transducers. Electrostatic transducers send and receive on the same transducer, but need to be held at high bias voltage to receive the signal, they also operate at a much higher voltage than piezoelectric transducers. Piezoelectric systems need two transducers, one to send, one to receive, but are often much less than half as expensive, making them the cheaper overall option. Both types need control circuitry to generate the proper frequency signal and will need some signal processing on the receive side. Some sort of filter will also be needed to remove noise from the signal and prevent false readings. The transducers are being attached to the outside of a vehicle, so they will also need to be weather proofed.

2.2 Wireless Communication

Four methods of wireless communication were investigated for this project, Bluetooth, 801.11 (WiFi), ZigBee and Infrared. Cost, Range and feasibility were the driving factors in the decision. The links would have to originate on the outside bumper of a vehicle and be received at the unit inside the vehicle. This requirement ruled out infrared communications due to its line of sight limitation. The wireless system needs to handle a three node network at a range of about ten feet, possibly more if the system were to be used on trucks. The amount of data transmitted will be fairly low and due to its benign nature does not need to be encrypted. WiFi networks require a lot of processing power due to their complexity. These networks offer high throughput with little concern for power efficiency, making them a poor method of communication for this application. Bluetooth, designed to connect peripherals to a computer, can be very complex, requiring advanced processing. The development kits

for Bluetooth are very expensive, partially due to the closed nature of its standards body. ZigBee communications were designed for home automation using numerous small remote nodes around your home. Simplicity and low power consumption are both priorities for ZigBee systems and due to its Do-It-Yourself nature was made much more accessible to designers.

2.3 System Control

The vast majority of microcontroller manufactures today offer a line of microcontrollers that are a system on chip in which memory, digital I/O, DAC and UART interfaces are all built into the chip. The decision regarding which chip line to use was based on in-house experience. The WPI ECE department just developed a course centered around Texas Instrument's MSP430 line of microcontrollers, which one of the members of this group had taken. The department's expertise and facilities would be available if we chose this line of chips. The features of the chips allowed this advantage to be taken advantage of without compromise. As seen in Table 2.1, the chips have plenty of digital I/O, UART interface for communication to the wireless module and low power consumption.

Microcontroller	Power Consumption μA	I/O	Timers	UART
MSP430F169	250	48	10	2

Table 2.1: Microcontroller Comparison Chart

The microcontrollers will be used at each wireless node; one for each bumper and one for the display circuitry. They are capable of generating clock and control signals for the sensor circuits while using a timer for the distance measurements. The development environment, IAR Embedded Workbench, uses a high level language(C/C++) for easy programming.

2.4 User Interface

The user interface needed to display detected objects to the driver of the vehicle. This unit would be inside of the vehicle with the driver, allowing the unit to be plugged into the vehicle's power system and loosening the power restrictions. The display needs to alert the driver of surrounding objects in a way that does not distract him or her from driving, but is sure to get their attention when an object is too close. The information being displayed is also static in format since a maximum of eight sensors are reporting back to this unit to be displayed. LEDs are bright enough to alert the driver, and their physical and power constraints are not a concern in this application. LEDs would be cheaper than an LCD of comparable size. The large size LEDs will allow the driver to easily check the device while not interfering with his or her driving.

2.5 Voltage Control

There are essentially three different modules that will need power. The units on the front and back bumper are identical and will each require a 3.3V rail with a minimal current requirement. Each of the sensor units will require a 5V rail with a max current rating of 200mA. The inside the car unit will require a 3.3V rail, and probably no more than 200mA depending on the display setup. Efficiency will also play a major role in the decision making process because we do not know how the batteries will be recharged, and if we are to have any hope of using an alternative energy source we will have to keep the efficiency as high as possible. Cost is also a fairly large factor because we will need one power supply for each of the sensor units, which could be as many as five in the front plus five in the back, plus supplies for the two processing units and the display unit. The three types of voltage regulator that we researched were the charge pump, the LDO, and switching converters.

2.5.1 Charge Pump

The charge pump is a non-inductive switching converter that typically has very low high frequency noise, and reasonable efficiency. The charge pump typically works by switching between three phases. In the first phase charge is transferred from the input voltage to the flying capacitor. This phase always takes place for half of the internal oscillator period. Once the flying capacitor is charged the device goes into an idle phase where all switches are open, and the output capacitor is discharging to the load. It stays in this mode so long as the feedback voltage is greater than the internal voltage reference. Phase three begins when the feedback voltage drops too low, and charge is transferred from the flying capacitor to the output capacitor to maintain the desired output voltage. If this phase takes less than half the oscillator period, then the device returns to the idle phase, if it exceeds half the oscillator period, then it returns to phase one so the flying capacitor can be recharged. Figure 2.2 shows the schematic of a very basic charge pump.

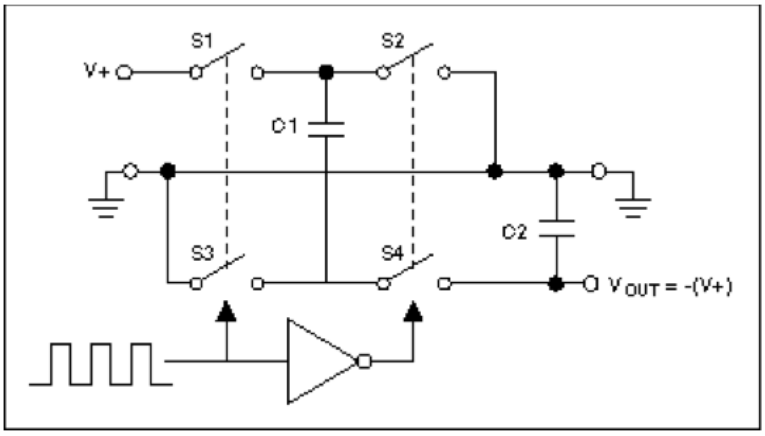


Figure 2.2: Charge Pump Operation [8]

One noted drawback is that the each of the switches has its own internal resistance, so the large number of switches is a source of reduced efficiency.

2.5.2 LDO

A low-dropout voltage regulator (LDO) can be used alone or used in conjunction with voltage converters to provide a constant dc voltage. These circuits are desirable when there is a potentially large variation in input battery voltage. Either LDOs or switching converters can be used alone to provide this constant dc voltage. Switching converters typically have higher efficiency, but also have more high frequency noise from the starting and stopping of the clock switch.

The dropout voltage of the LDO is defined as the difference between the minimum input voltage and the dc output voltage required before the device stops operating. The most important factor in LDO efficiency is the quiescent operating current. When the output current is low, as is commonly the case, the quiescent current becomes the biggest concern as the efficiency of the LDO is given by Equation 2.2. [16]

$$\text{Efficiency} = \frac{I_{Load}}{I_{Load} + I_q} \tag{2.2}$$

This equation shows that as quiescent current goes down or load current goes up, efficiency goes up. Also, the closer Vout is to Vin, or in other words, the smaller the dropout voltage, the lower the quiescent current, and the better the efficiency of the LDO. Because of this, a dc-dc switching converter is preferred in situations where there is a large difference between the input and the output voltages, especially if the voltage must be stepped up which an LDO cannot do. However, in situations where there is a large input/output voltage differential and a low noise output is required, it is common to cascade an LDO after the dc-dc converter to produce a low noise constant dc voltage. The basic operation of an LDO is shown in Figure 2.3.

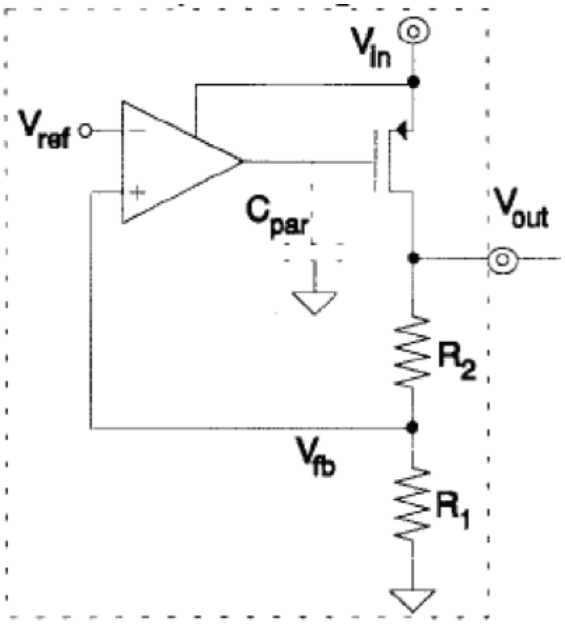


Figure 2.3: LDO operation [16]

2.5.3 Switching Converters

The operation of the most basic switching converters can be described by the use of an inductor, a capacitor, a diode, and a switch. A buck converter can step a dc voltage down to a lower dc voltage, a boost converter steps up a dc voltage, and a buck-boost converter can do both. There are many other different kinds of switching converters, but these are the most basic. All of the basic switch converters work in more or less the same way. When the switch is on the inductor gets charged up, and when the switch turns off the inductor discharges to the capacitor and load. The capacitor acts to remove the output voltage ripple so all the load sees is the dc component. The transfer function depends on the configuration of the diode, switch, and inductor. The transfer function can be determined by analyzing the current through the inductor in the on state, and the off state, and giving them a weighting which is determined by the duty cycle. The resulting transfer functions and basic operation are shown in Figure 2.4.

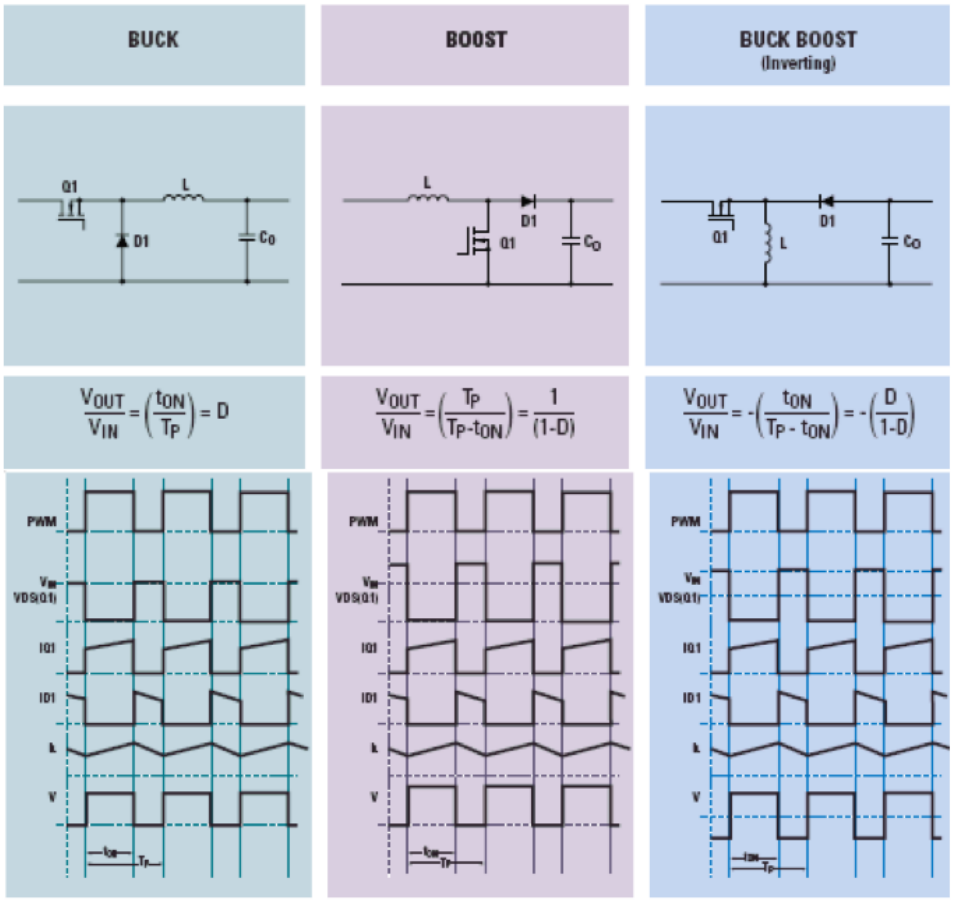


Figure 2.4: Operation of Basic Switching Converters [19]

2.5.4 On the Market

From researching the operations of each power supply type, we found that under the right conditions any one of the power supply topologies could be used. Once we started shopping for parts I found that most of the switching converters had built in LDOs which reduced the high frequency noise problem. The only general trend in price was that the LDOs tended to be much less expensive at around 50 cents. The charge pumps had a huge price range at anywhere from around two dollars to up to around seven dollars. The efficiency of the switching converters were typically above 90% while charge pump efficiency ranged from 50-90%. LDO efficiency was typically not specified, which makes sense as the efficiency depends entirely on how close Vin is to Vout, and how much larger the load current is than

the quiescent current.

An LDO cannot do boost operations, so it is out of the question for the 5V rail, and because the battery voltage will have a large range the efficiency will have a large range as well. This makes the LDO a poor choice for either rail. However we did find one LDO that may be of use if we need to drive the 5V rails from the 3.3V rail. The MCP1726 can source as much as 1500mA and only costs $1.03. This may be the only cost effective solution to driving large currents at 3.3V. To compare results we made the table shown in Figure 2.5 which outlines the performance of each of the chips we thought were worth looking at.

2.5.5 Selection

The most promising options we found at lower currents were the MCP1252 and the LTC3525. Both of these chips have 3.3V and 5V rail options. The change in output voltage as load current changes is around 0.03V for the LTC3525 and about 0.1V for the MCP1252. The added precision is impressive, but not very necessary as they both exceed the minimum accuracy required by the devices they will be supplying. The input voltage ranges are 2.0 to 5.5V and 1.2 to 5V respectively. So, both chips can handle the battery voltage ranges no matter what battery setup we decide to use. Both chips have a maximum continuous current rating of 150mA. Both have very few external parts which include input and output capacitors for both, a flying capacitor for the MCP1252, and an inductor for the LTC3525. The costs of the two chips are $1.85 for the MCP1252, and $2.12 for the LTC3525. Thats less than a 30 cent difference, which gives the MCP1252 a slight advantage. However, we believe the deciding factor should be the efficiency. The LTC3525 has between a 90 and 95% efficiency when the load current exceeds 0.5mA, which it will. The efficiency also goes up as supply voltage goes up, so if we use a lithium battery the efficiency can get even higher. The MCP1252 however has a maximum efficiency of just below 90% at very low supply voltages, and decreases as input voltage increases. If we were to use a 5V Li-ion battery we would

Chip	MAX682	MCP1252	LTC3525	TC1107	MCP1700	TC120	MCP1726
Type	Charge pump	Charge pump	Burst Mode Synchronous Boost	CMOS LDO	CMOS LDO	PWM Buck	LDO
Vout	5V	3.3 or 5V (same pkg)	3, 3.3, or 5V (dif. Pkgs)	1.8-5V (same pkg)	2.5-5V (same pkg)	3.0,3.3,5.0 (same pkg)	3.0,3.3,5.0 (same pkg)
Vin	2.5-5.5V	2.0-5.5V	1.2-5V	2.7-6.0V	2.3-6.0V	1.8-10V	2.3-6V
Iout	250mA	150mA	150mA	300mA	250mA	600mA	800-1500mA
Iin (no load)	7.5mA			50uA	1.6u		
eff	70-85 (from 3.6 to 3.0Vin)	50-90%	85-95 (from 1.2 to 3Vin)			95	
unit cost	6.51	1.85	2.12				
bulk cost	3.47			0.35	0.29	call for pricing	1.03
External Parts							
Cin	1uF	10uF	1-10uF	1uF	1uF ceramic	47uF tantalum	4.7uF
Cout	2uF	10uF	10-22uF	1uF	1uF ceramic	47uF tantalum	1.0uF
other parts	0.47uF Cx	1uF ceramic	4.7-15uH L	470pF Cbypass		22uH L, and schottky diode, optional switchin MOSFET	optional 1000pF, voltage divider
Package Type	8-pin SO	8-lead MSOP	6-pin SC70	8-pin MSOP or SOIC	3pin SOT23-A, SOT-89, TO-92	8-pin SOP	8-pin SOIC
Other:	can use 2 in parallel to provide up to 500mA					incase we need more current	

Figure 2.5: Comparison of Voltage Control Chips

be looking at an efficiency in the range of 50 to 70%. Because we did not know what our battery charging situation is going to look like at the beginning of the project, we think it is extremely important that efficiency be as high as possible. So, for slightly more money we have chosen to use the LTC3525.

The LTC3525 must be used for the 3.3 volt rail, but the 5 volt rail can either be driven from the 3.3 volt regulator, or be driven directly from the battery. LDO's are more efficient when the input voltage is closer to the output voltage, so there may be an advantage to daisy chain the two regulators. This would require a 3.3 volt regulator capable of sourcing more current, such as the MCP1726. The added costs associated with this marginal improvement in efficiency was not worth is, so the 5 volt supplies were driven directly from the battery.

2.6 Power Storage

Even before knowing the power demands of the sensor circuit we can begin looking at battery requirements. For the front and back bumper sensor modules we require that the batteries be rechargeable and have reasonably simple charging circuits. Other criteria we will judge batteries on will be cost, memory effect, maximum number of recharges, power storage (Amp-hours), nominal voltage, and maximum current. Because we do not yet know the maximum current or voltage requirements, and because there are generally several different models for each battery type that can accommodate a large range of currents and voltages, we will not consider nominal voltage or maximum current. Based on these criteria we will critique each of the types of batteries being considered: Lithium-Poly, Lithium-Ion, Rechargeable Alkaline-Manganese (RAM), Nickel Cadmium (NiCd), and Nickel Metal-Hydride (NiMH).

2.6.1 Cost

After initial investigation, it appears the sensors and wireless devices may be very expensive. After accounting for the other small costs and possible unforeseen costs, we find that

our project is falling under strict budget constraints. For this reason we want to cut costs wherever possible, so the cost of the battery will have a weight of 4.

2.6.2 Memory Effect

Memory effect is defined by the situation in which a battery begins to lose its maximum storage capacity due to partial charging many times in a row, or due to being charged after only being partially discharged. This occurs because the tiny crystals in the active materials at the surface of the batterys electrodes change from thousands of microscopic crystals to fewer larger crystals, which effectively reduces the surface area. Because our circuit will ideally be experiencing very shallow discharge/charge cycles, it is important that the batteries not experience memory effect. Though, if the charging works well enough, the batteries may not need a large storage capacity, but we will not know if this is the case until very late in the project. All things considered, we are giving Memory Effect a weight of 3.

2.6.3 Maximum Number of Recharges

The way a battery charges and discharges is chemistry dependent. The maximum number of recharges refers to full charge/discharge cycle. The maximum number of recharges is a somewhat nebulous term because of the rate at which a batterys energy storage capacity decays. If the battery suffers from the memory effect, then the amount of charge stored per cycle will be dramatically decreased. However, if the battery capacity can be assumed to be decaying at an ordinary rate, then the maximum number of recharges is a good measure of how long the battery will last. Some batteries also have a maximum shelf life, which is time dependent rather than being dependent on the number of recharging cycles. However, the battery shelf life is never noted by the manufacturer, and varies between battery types as well as between batteries of the same type. For this reason our only somewhat reliable measure of battery life is the maximum number of recharges, but because this is a rather vague figure we are giving it a weight of 3.

2.6.4 Energy Storage

The storage of a battery is rated in Amp-hours, meaning the number of hours it would take to discharge with a constant output current of one Amp. The values given for this category are the nominal values, and because these batteries are rechargeable, this value decreases with every charge. Energy storage can also depend on how the batteries react to different charging cycle depths. Other factors aside, the amp-hour rating of a battery is a good measure of the batterys total capacity. The amp-hour rating is in the scale of hundreds and thousands, and typically the marked value is less than the actual value. For these reasons we give Energy Storage a weighting of 4.

2.6.5 Battery Types

RAM

RAM batteries are a newer generation of Rechargeable Alkaline batteries. They can be charged by either a DC current at voltages below 1.65V, or pulse charged very quickly. These batteries perform better with shallow charge cycles, which may be advantageous for our purposes since we plan to charge these batteries as we use them. RAM batteries also cost less than NiCd batteries, around $2 each. They also have a much slower self-discharge rate than NiCd or NiMH (about 10% per year). We have not found any notes on the maximum number of charges, or the maximum power stored, only that the initial power density is around 75Wh/kg, which is comparable to NiMH. This sounded like a very promising option if we could actually find a seller. The information reported is on batteries from a Canadian company Battery Technologies, but they did not have order forms for commercial batteries on their webpage so this approach was abandoned.

Lithium-Poly & Lithium-Ion

These two battery types both represent very new technologies, their only real difference being that the Poly batteries do not require a rigid metal container, making it possible for them to fit into oddly shaped tight spaces. These batteries have extremely high power densities (>100Wh/kg), but are very difficult to recharge. These batteries cannot be charged as quickly as NiCds or NiMH, and cannot be trickle or float charged. These batteries have no memory effect, can be charged more than 600 times, and have a relatively low self-discharge rate. The reason both of these batteries are being clumped into one category is they both share the same major drawback. They are extremely expensive! Some of these batteries range up to five times higher than the next most expensive battery. So, between the charging difficulties and the cost, these batteries are out of the question.

NiCd

NiCds do not perform well under a floating charge, and prefer to be very deeply cycled between charged and nearly fully discharged. These batteries also last longer if they are charged quickly, preferably pulse charged, which again is unlikely for this project. The most notable drawback of the NiCd batteries are that they experience a memory effect, and must be conditioned, meaning nearly fully discharged, and fully charged a couple of times to maintain its capacity. These have a relatively low energy density and have a high self-discharge rate. Aside from being very widely available, and inexpensive, these batteries have little to offer our project. NiMH

NiMH batteries are similar to the NiCd batteries, and do experience some memory effect if not discharged almost completely at least once every 30 cycles. If treated properly these batteries can be recharged over 1000 times, but by all accounts have a shorter working life (about 50%) relative to time rather than number of charges. They also have a higher capacity (about 30%) and higher power density (about 50%) relative to NiCd. NiMH must

be charged more slowly than NiCD, and also dissipate heat while they are being charged, so most NiMH chargers involve temperature sensors. This fact may not be that detrimental, as NiMH batteries are commonly used in electric vehicles which experience huge temperature ranges. These vehicles do have large air cooling systems for the batteries, that will probably not be desirable for our project. These batteries do cost a little bit more than NiCd and have many tradeoffs that make them probably only a little more desirable than the NiCd batteries.

2.6.6 Conclusion

It is safe to say that the Lithium based batteries and the NiCd batteries are out of the question. The remaining options are the NiMH and RAM batteries. After checking the availability of these batteries we found the RAM batteries to be extremely hard to find. We were only able to find one seller of RAM batteries, and there was absolutely no mention of how to make a RAM battery charger. Therefore, we would be unable to charge the batteries, and it would be extremely inconvenient for consumers to purchase new batteries. So, we have chosen to use the NiMH batteries for their availability, the simplicity of the charging circuit, and the low cost.

2.7 Recharging

2.7.1 Energy Source

Our external units will be run off of battery power. If we do not want the user to have to purchase and replace batteries regularly, then we will have to find some way to recharge the batteries. For this reason the battery powered units will come equipped with battery chargers. It would also be an added benefit if we could somehow draw energy from a source other than the vehicle power supply or a wall outlet. Some possible options include solar energy, wirelessly transmitted energy, thermal energy, or mechanical energy. However, the

majority of these options can be eliminated without much research.

Solar energy is not a very good option, because the amount of available sunlight varies by location, and also from day to day. This would mean that it might be possible that the device would work some days, but not others, which is not an option for our device. There are several potential hot spots on a vehicle that could be used for thermal energy capture, but the location of these points varies from car to car. Also, a thermal energy capturing device would have to be designed to be easily installed, hard to dislodge, and not interfere with the vehicles operation. This task alone could be its own MQP. Wirelessly transmitted energy could be sent from the battery in the front of the vehicle to both of the bumpers, but the state of wireless power technology is nowhere near available enough to make this a viable option. Available wireless energy gathering devices are too inefficient, and devices that could possibly be viable solutions are still very experimental or are too cutting edge to be described in any sort of usable detail. Cost is also a huge deterrent in the cases of thermal and wireless energy options.

The only remaining option is mechanical energy. Mechanically gathered energy has already been used in automotive applications such as the process of gathering energy from magnetic breaks. The vehicle constantly vibrates from the movement of the engine, and the motion of the vehicle itself could potentially be used to harness energy. For these reasons we investigated the possibilities of using mechanical energy to recharge our batteries.

Mechanical Energy

Harvesting kinetic energy is a fairly old technology, and as such there are a number of different designs available. The difficult part is finding a method that can be used on a vehicle, and can provide enough power to recharge the batteries faster than they are drained. One potential model for such a device is shown below in Figure 2.6.

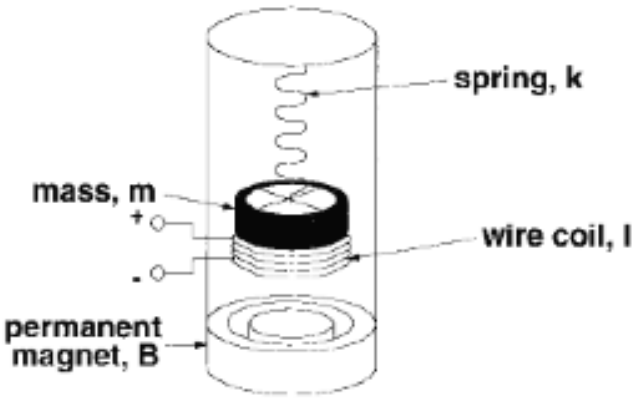


Figure 2.6: Mechanical Schematic for a Magnetic Generator[1]

A wire coil is wrapped around a mass which is connected to a moving rigid frame by a spring. When the outside frame moves, the mass will have a delay before it can react to the movement, at which time it will move through the magnetic field, forcing a voltage through the wire coil. This principle is similar to that used in a moving coil speaker. The experiment using this model used a small 0.5g mass and was able to get out a 180mV peak. This voltage is too small to rectify using a diode, so a transformer must be used to increase the voltage to a usable amount. The voltage can also be manipulated by changing the strength of the magnet or the number of turns of the coil on the mass.

The inertial solenoid generator in an LED flashlight shown in Figure 2.7 generates 200mW and weighs 150g. This design might be usable if we were using the motion of the vehicle rather than the vibrational energy. However, the abrupt starting and stopping, or sharp vertical bumps that would be required to produce energy are too infrequent to be relied on for battery recharging purposes.

Another design of interest for kinetic energy harvesting is that used in watches. The design shown in Figure 2.8 can generate up to 1mW when forcibly shaken, and is extremely compact. The problem with such a design is that it relies on Micro-Electronic Mechanical Systems (MEMS) technology, which we do not have access to, and falls well outside of the

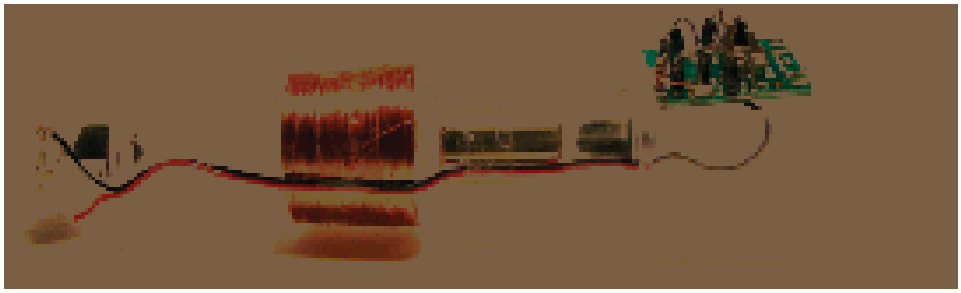


Figure 2.7: Inertial solenoid generator in an LED flashlight [13]

scope of this project.

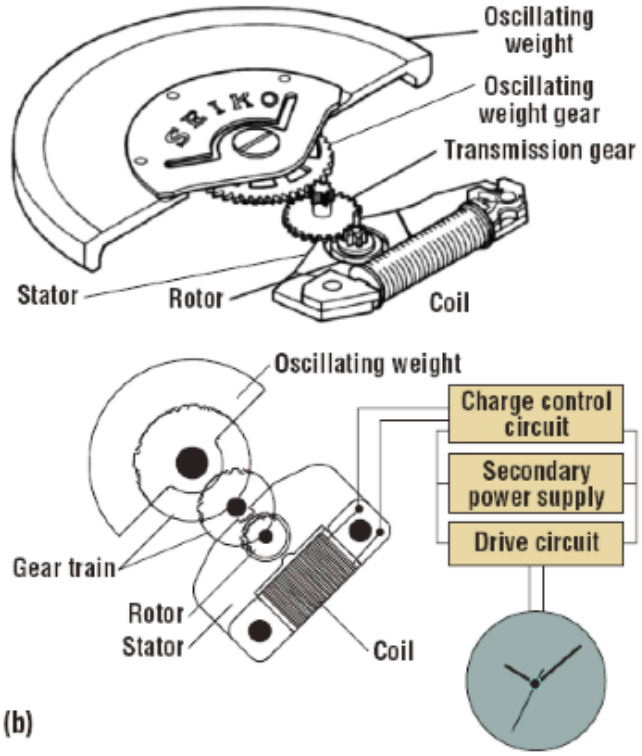


Figure 2.8: Seiko Automatic Generating System [13]

Another potential way to harvest energy is by the use of piezoelectric materials which deform in the presence of an applied voltage, and conversely produce a surface voltage when deformed. These materials are commonly used in audio transducers as well as in highly accurate clocks. There exists a patent for a piezoelectric energy harvesting system

that can be mounted on the inside of a tire, but there are no test results given with the patent.[21] However, this design is not applicable to our design as installing a piezoelectric generator inside of a tire would not be a very practical installation for an aftermarket device. Piezoelectric materials also have dimension dependent resonant frequencies that must be used in order to efficiently capture energy. Theoretically a 1.5 x 2.5 x .0075 sheet could theoretically produce 9 watts of electrical power from 100 watts of mechanical energy input under resonant conditions. More practical numbers for the same conditions would be around 3.6mW with max voltage peaks around 15V, and an RMS Voltage of about 5.3Vrms.[14] Under ideal charging conditions for two NiMH batteries we would like to be charging at 2.8V and 450mA, or 1.26W. This is enormously more power than we can get out of a single sheet of piezoelectric material. Furthermore, a single 2.85 square sheet 0.005 thick costs $100![15] This makes using a piezoelectric sheet extremely out of budget.

Conclusion

Mechanical energy harvesting would have been the only viable source of energy. However, devices that take advantage of the motion of the vehicle would only produce energy when the car accelerated/decelerated, or when the car hit sharp bumps. Neither of these methods would produce a consistent enough in flux of energy. Therefore the constant vibrational energy of the vehicle would be the only viable option. However, neither MEMS devices nor piezoelectric technologies produce enough energy on a practical scale, and both are phenomenally expensive. For these reasons we have decided to use a wall outlet as our source of energy.

2.8 Charging NiMH Batteries

Any number of different sources will give different recommendations on how to charge a NiMH battery. How to charge the battery also depends on the manufacturer and type as well.

Some NiMH batteries such as the Energizer rechargeables have a nominal storage capacitance of 2500mAH, and can be charged in as little as 15 minutes.[4] Other older NiMH batteries store as little as 1500mAH, and cannot be fast charged.[6] Specific charging information is generally not provided by the manufacturers because they always have a charger that they are also trying to sell. The charging information I found is primarily based on the NiMH battery charging page from www.powerstream.com.[7] This web page provides thorough information on battery chemistry and battery chargers.

Traditionally, older battery chargers work on either a timer or on peak voltage bump detection. Battery chargers that work on a timer have a high risk of over charging. The timer can only work if the battery is fully discharged, and if they are only partially discharged, then providing enough amp-hours to fully charge the battery can damage the battery. The peak voltage bump detection charge method is based on the fact that when batteries are fully charged the voltage drops very slightly. However, in NiMH batteries this peak voltage bump is very slight and hard to detect, and there are also occasional bumps before the peak voltage.

For charging NiMH batteries there are three recommended charging methods. Fast charging can be achieved with a temperature sensing system, and can charge batteries at a rate of up to 1C. C is the nominal amp-hour rating of the battery divided by hours. That is, if a battery has a 1500mAH rating, then 1C equals 1500mA. However, not only does this method require an extra thermistor, but many NiMH batteries cannot be fast charged. The second method is a slow overnight charge at a rate of C/10. Not only is this method extremely slow, but according to Panasonics NiMH handbook[12] a slow charging method can reduce the lifetime of the battery. The third recommended method is a medium speed charge at a rate of C/3.33. This method should only be used if there is some shutdown method that senses when the battery is fully charged. This can be done with either a timer or preferably with a voltage sensor. Ideally we would like to find a charger that can use this method.

2.8.1 Battery Charger Controllers

We would like our device to have a built in battery charger. We also had intentions of using an alternative energy source, in which case it would become important to be able to charge a battery with low voltage, and possibly even varying voltage. The ability to sense the battery's charge level is very important. Overcharging is a problem for all battery types, so the charger must be able to shut off once the battery is charged. Maximum charge current does not have to be very high, because we are planning on using a medium speed charge method for NiMH batteries. NiMH batteries range from 1500-2500mAH, so at a rate of C/3.33 that is a charging current range of 450-750mA. Because the charger should work safely for all NiMH we should use the minimum charging current of 450mA. This low charging current requirement should make it easier to find a battery charging controller. Input voltage range, minimum input voltage, battery charge level sensing, and cost will be the primary concerns for selecting a charger.

On the recommendation of Power Design professor, Prof. Thompson, and having had good success with Linear Technologies in the past, Linear Tech was the first vendor we checked. The battery charger solution manual provided by Linear Tech provided a comprehensive comparison of all the battery chargers the company has to offer.[5] After reading through the document it appeared that the market tended to be slanting toward lithium ion batteries. Also, the majority of chargers that worked for NiMH batteries had options that worked for both NiMH and Li-ion batteries. For this reason we decided it may be a good idea to keep the option of using Li-ion batteries as an option by selecting a charger that can charge either type of battery. Charger types that can charge either type include certain switchmode monolithic chargers, switchmode controller chargers, and switchmode smart battery chargers.

All of the available charging circuits that met specifications were switchmode controllers, meaning their operation depends on the duty cycle delivered to a pass transistor. This

presents another advantageous option that the pass transistor may be integrated into the controller itself. If this feature is added in to the selection criterion, then there are only five available controllers to choose from. Comparing these controllers, the controllers with the lowest minimum input voltage are the LT1512 and LT1513. The LT1513 is a larger, more expensive version of the LT1512, and can support up to 2A of charging current, whereas the LT1512 can only support up to 0.8A. However, this 0.8A is well above our required charge current of 0.45A, making the LT1513 unnecessary. The LT1512 has an input voltage range of 2.4V to 29V, which is much larger than required. When comparing the price of the LT1512 to other suitable chargers that had a less suitable input voltage range, the LT1512 was also the cheapest. The LT1512 also has the second fewest number of external parts required, making it the clear choice of the battery charging controllers offered by Linear Tech.

2.8.2 SEPIC Converter Operation

The LT1512 is a Single Ended Primary Inductor Converter (SEPIC) which is a type of non-inverting DC/DC switch mode converter that can either buck or boost. This type of converter requires a switch, a pair of inductors which can be coupled in a transformer, and at least three capacitors. The typical SEPIC configuration is shown in Figure 2.9. The

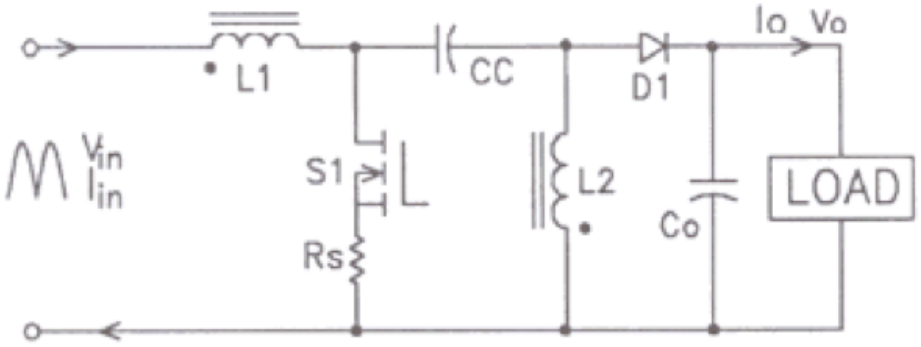


Figure 2.9: SEPIC Converter (input capacitor not shown)

coupling capacitor marked as Cc clearly blocks all current from the input to the output,

so the operation of such a converter is not very intuitive. The best description we could find of how a SEPIC converter works, as well as the above schematic came from a research paper High Power Factor Preregulator Using the SEPIC Converter[3] by Lloyd Dixon. The first step to understanding the SEPIC converter is to know that the DC current through a capacitor is zero, and the dc voltage across an inductor is also zero. Using KVL loops we can show that whether the switch is on or off, the DC voltage across Cc is equal to Vin. So, if Vcc = Vin, then voltages across the inductors can be determined in both states.

When the switch is on the voltage across L1 is equal to Vin, and if you take the inductor voltage orientation with the dot as the positive end, then the voltage across L2 is also Vin, going from ground to Vin. This is also consistent with our left to right, positive to negative orientation of Cc so that the voltage across Cc also equals Vin. Because Vout is of the same polarity as Vin we know that when Vin is positive, Vout must be positive, forcing the diode to be off when the switch is in the on state. For inductors V = L di/dt, and because we know that they have a positive voltage, we know that the inductor current must be ramping up.

When the switch is off the current from the first inductor charges up the voltage of the capacitor, however no dc current can make it through the capacitor to the diode. Since the current in L2 has now ramped up and nothing is forcing the voltage down at the anode of the diode, that means the charge now wants to escape through the diode. If the diode is now on, then the voltage across the load, Vout, is now applied in the reverse polarity to L2, and the voltage on the other side of Cc must be Vout + Vin. The voltage on the other terminal of L1 is still Vin, so the voltage across L1 is also Vout. This means that the currents through the inductors go up and down at the same rate throughout the cycle. This also means all the current that is delivered to the load comes from L2 and is applied when the switch is off. The output capacitor serves to filter this current so that a DC current is applied to the load.

The LT1512 takes advantage of the fact that the DC current through L2 is the same as the DC current applied to the load. By placing a small sense resistor between L2 and ground the current into the battery can be measured. The value of the sense resistor determines the maximum current that can be delivered to the battery. The duty cycle of the pass transistor will be adjusted based on the voltage at the Ifb pin. Also, because a battery has a low internal resistance, a voltage divider using large resistors can be used to detect the battery voltage without a significant amount of leakage current. This voltage divider is set such that the divided voltage is compared to an internal voltage reference of 1.245V. In this way the nominal voltage can be chosen with the selection of the resistors in the divider, and the LT1512 will stop delivering current when this voltage is reached.

2.8.3 Conclusion

The LT1512 can limit the current, the charging shuts off when the nominal battery voltage is reached, there are relatively few external parts required, the minimum input voltage is low, and this is one of the cheapest battery charger controllers Linear Tech offers. Since we have not found a suitable alternative energy source we are presently planning on plugging the charger into a wall outlet through a 5V AC/DC adapter.

Chapter 3

Implementation

3.1 Sensor Circuit

3.1.1 Overview

An impulse-echo rangefinder requires some specific control circuitry. Piezoelectric transducers commonly resonate at 40 kHz, requiring some signal to be generated at that frequency. The transducer produces more audio power as the voltage increases as seen in Figure 3.1, so a transformer will be used to up convert the low voltages used in powering this system. Each sensor will use its own power regulator, so they can be shut off when the sensor is not taking a reading; this will reduce the power consumption of the system. By regulating the power locally, the power in the circuit will also have fewer sources of noise. To take a proper reading, the transmit signal needs to be shut off so interference on the board does not cause false positives. On the receive side, the small signal generated by the receive transducer has to be amplified and extra noise needs to be removed to enable accurate readings. This circuitry then needs a connection to the communications module. These design requirements lead to a system design seen in Figure 3.2.

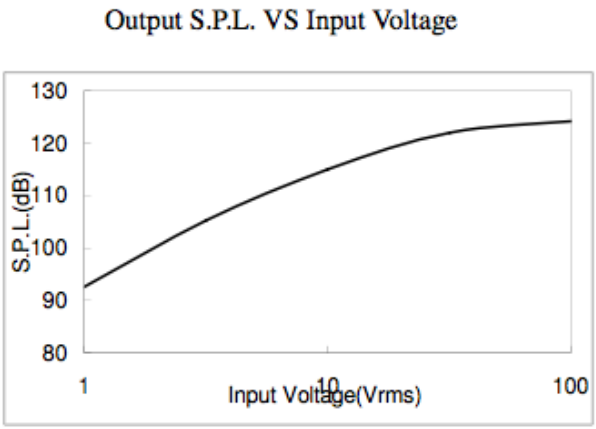


Figure 3.1: Audio Power vs Voltage [2]

3.1.2 Transmit

To meet the requirements of the system, the TL851 Sonar Ranging Control chip was used, made by Texas Instruments. This chip takes a 340 kHz clock signal from the MSP430 and divides it by 8.5 to generate the 40 kHz signal. When the Init signal is set high, the control chip generates 16 pulses of the 40 kHz signal on the Xmit line. The transmit signal then goes to a MET-23 Audio Transformer. This transformer was chosen because of its high frequency capabilities. Relative to the power input at 1 kHz, there is 3 dB of attenuation at 100 kHz. The impedance ratio of the primary to the secondary is $1.6k\Omega$: 3.2Ω, using Equation 3.1 we know the turns ratio to be 22.36:1, or that for every volt put on the secondary, 22.36 volts will be seen across the primary.

$$\sqrt{\frac{Z_P}{Z_S}} = \frac{N_P}{N_S} \tag{3.1}$$

To increase the current drive capability of the 5 volt logic output by the control chip, the transmit signal was be connected to a transistor drive circuit. The transistor was be connected from the transformer's secondary through a resistor to ground, as seen in Figure 3.3. This results in over 100 volts across the transducer, which is off the graph from the transducer data sheet as seen in Figure 3.1 and well into the region where the audio output power begins to level off. By maximizing the output power of the transducer, the reflected

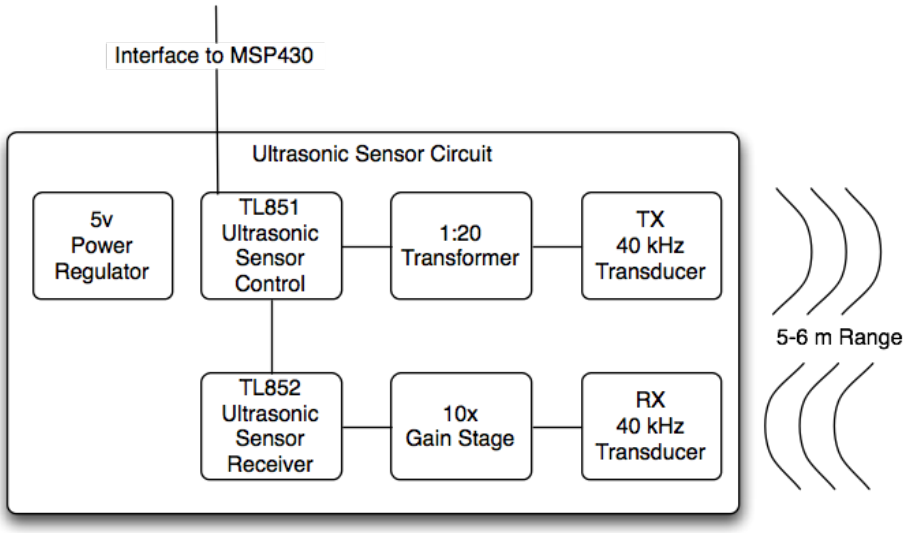


Figure 3.2: Overall Sensor Circuit

echo will be easier to detect and more distinguishable from noise.

3.1.3 Receive

Piezoelectric transducers do not need to be held at a bias voltage to receive a signal as do electrostatic transducers as discussed earlier, so the receive transducer is connected directly into the first gain stage. There are two -10x gain stages in the inverting configuration, as seen in Figure 3.4. In Figure 3.4 both R5 and R7 are the input resistances and R6 and R8 are the feedback resistances. Using Equation 3.2 the gain of each stage can be solved for. For both stages, R_{in}= 1$k\Omega$ and $R_{feedback}$=10 $k\Omega$ and the result is a -10x gain.

$$\frac{V_{out}}{V_{in}} = -\frac{R_{feedback}}{R_{in}} \tag{3.2}$$

When the Gain stages were first designed, LM741 dual sided op amps by National Semiconductor were used, and the inverting configuration did not matter. While designing the printed circuit board a single sided op-amp was needed, so the op amp was changed to Texas Instrument's TLV263X line of low power wide bandwidth single supply op-amps. This line

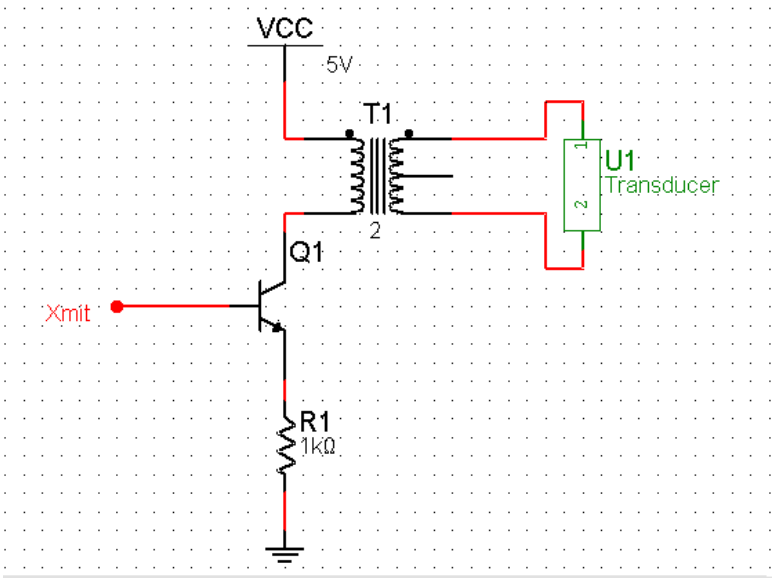


Figure 3.3: Transformer Connection

has the choice of 1, 2 or 4 op-amps per chip and optional power on/off control. Due to supply limitations, we used two separate single op-amp chips and we did not need the power control, resulting in the TLV2631 as desired. This chip has a 9 MHz gain bandwidth product. The two gain stages together would require 100 (both stages together) $\times$ 40 kHz or a 4 MHz gain bandwidth product, which the TLV2631 is capable of. The inverting configuration became a problem with no negative rail, so on the PCB the second gain stage was disconnected. It was not replaced with a non-inverting configuration because at the time this problem was discovered there was also a problem with false positive responses which decreased with the reduction in gain. The output of this stage then went into the receive chip.

The companion chip to the TL851 is the TL852 Sonar Ranging Receiver. This analog circuit includes gain stages and filters to clean the incoming signal and output a signal for the TL852 to determine the presence of an object. The first gain stage is an inverting op-amp following Equation 3.2, but currently this is set for unity gain with $R_{feedback} = R_{in}$. This gain stage will be able to compensate for the removal of the second external gain stage. This stage also introduces a 0.7 volt offset.

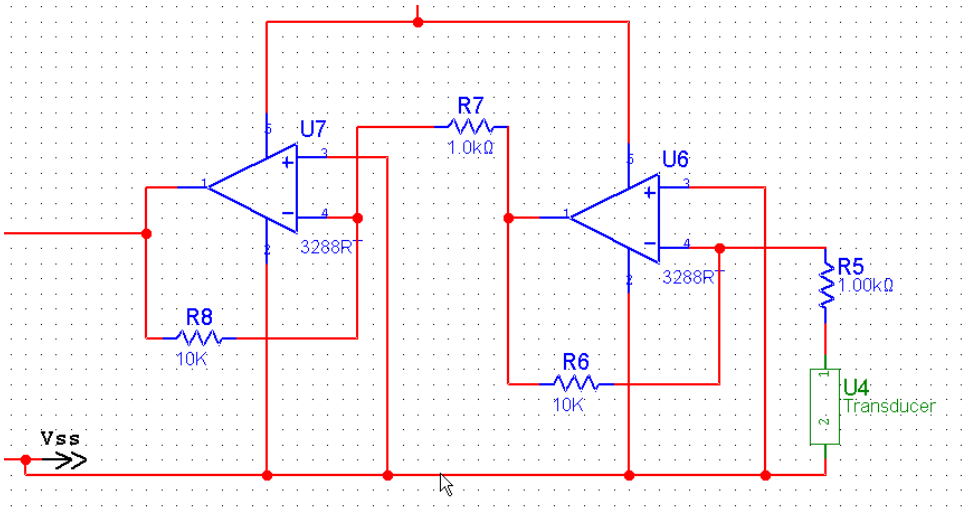


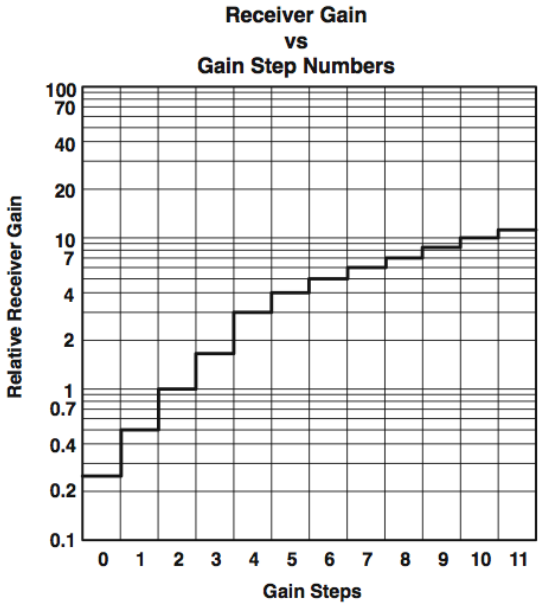
Figure 3.4: Inverting Gain Stages

The further away the object is, the more attenuation will reduce the power of the returning impulse and the longer it will take for the echo to return. The TL851 and TL852 work together to account for this by using a timing controlled gain stage. The TL851 uses the clock to keep track of how long the pulse has been traveling. It then sends logic over four gain control lines to let the TL852 know what gain stage it should be using as seen in Figure 3.5. The timing is dependent on the input clock; the exact times in the table are for a 420 kHz input clock, slightly faster than the 340 kHz clock we are using.

In addition to the frequency response of the transducer, as seen in Figure 3.6, an LC circuit is used by the receive chip to further filter out noise. This LC circuit has a center frequency found using equation 3.3.

$$f = \frac{1}{2\pi\sqrt{LC}} \tag{3.3}$$

In-order to get the desired 40 kHz, LC needed to be solved for, the inductance (L) and capacitance (C) used also needed to be values that could be ordered off the self. The board had already been ordered, so the parts also had to fit a specific footprint. Using a $47\mu H$ inductor and a $0.33\mu F$ capacitor resulted in a 40.412 kHz center frequency.



GAIN CONTROL OUTPUT TABLE

STEP NUMBER	GCD	GCC	GCB	GCA	TIME (ms) FROM INITIATE↑†
0	L	L	L	L	2.38 ms
1	L	L	L	H	5.12 ms
2	L	L	L	L	7.87 ms
3	L	L	H	H	10.61 ms
4	L	H	L	L	13.35 ms
5	L	H	L	H	16.09 ms
6	L	H	H	L	18.84 ms
7	L	H	H	H	21.58 ms
8	H	L	L	L	27.07 ms
9	H	L	L	H	32.55 ms
10	H	L	H	L	38.04 ms
11	H	L	H	H	INIT ↓

† This is the time to the end of the indicated step and assumes a nominal 420-kHz ceramic resonator.

Figure 3.5: Receiver Gain Over Time [18, 17]

This amplified and filtered signal was then returned to the TL851 control chip and put through a comparator. This comparator is set internally to the TL851 at 1.2 volts. When the Receive signal passes this threshold, the Echo line is set high and held high until the Init line goes low. These chips can be used to detect multiple objects in a single reading, but that functionality was not used in this project.

3.1.4 Voltage Regulator

The high efficiency boost requirements of this system lends itself to Linear Technology's line of switching voltage regulators. The LTC3525-5 will give the 5 volt rail from the 2.4 volt battery and supply the needed current requirements. The circuit needed very little support circuitry, as seen in Figure 3.7. The data sheet suggests ceramic capacitors for both C1 and C2 and an additional tantalum capacitor, C3, if the battery is more than a "few" inches away from the regulator.

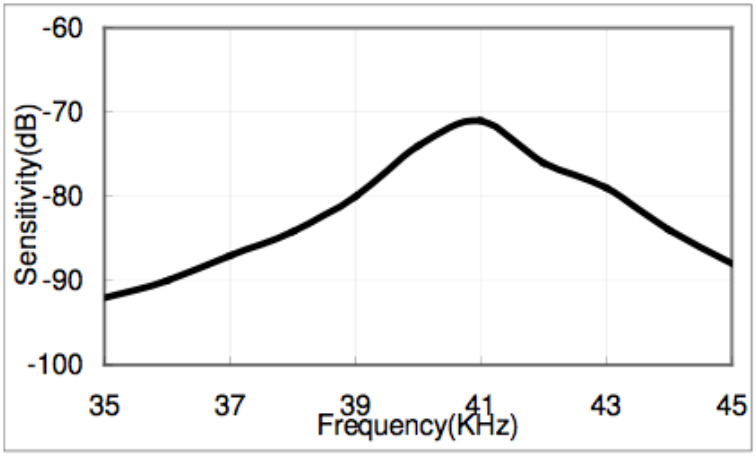


Figure 3.6: Sensitivity vs Frequency [2]

3.1.5 Interface to MSP430

The cable going from the Communications module to the sensor modules consists of 6 lines. The first two lines carry power and ground. The MSP first sets the Power Control line high to turn on the voltage regulator. A 5 ms pause is then needed for the control circuitry to power up. The 340 kHz clock is sent to all four sensors continuously. The acoustic pulses are sent when the Init line is taken high and the MSP chip starts a timer. The timer is stopped when the Echo line goes high, driven by the sensor circuit. The cable used in the prototype is standard Cat 5 cable with homemade connectors. Three of the eight wires were used for ground, twisted with the Power, Clock and Echo lines to reduce interference caused by the length of cable.

3.1.6 Printed Circuit Board

To ensure each of these systems were working, they were prototyped on breadboards before a Printed Circuit Board was ordered. The sensor circuit construction and evaluation started poorly with a bad breadboard, a difficult bug to detect. With control and clock signals generated by a Spartan 3 board the sensor circuit was tested before the MSP430 code was finished. The Spartan 3 board was able to generate any clock desired, but generated a 340

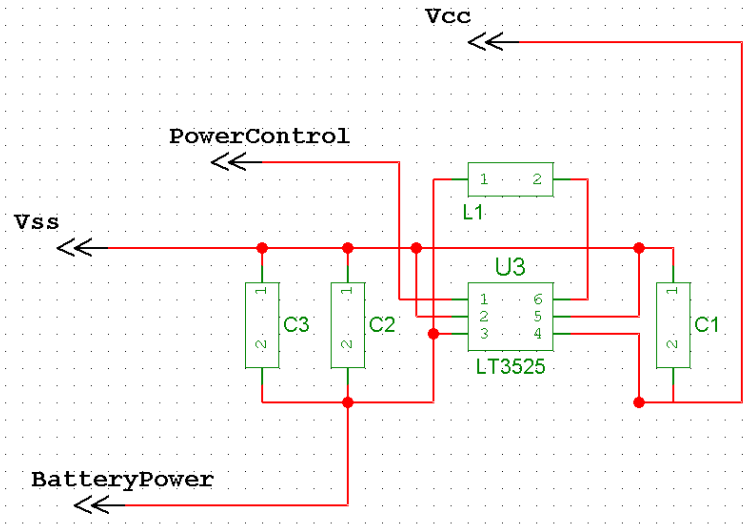


Figure 3.7: Power Regulator

kHz signal for the clock and a 100 ms pulse for the Init line. Power was held high, so the switching and associated timing was not tested. After some problems with the sending pulses were addressed, the breadboard worked with approximately a 0.5 meter range. The voltage regulator was also prototyped on a breadboard and tested. The LTC3525, needing a breakout board, worked intermitently. The problem was found to be the breakout board lifting itself out of the breadboard and losing connections. The resulting sensor circuit can be seen in Figure 3.8 and its power regulator in Figure 3.7.

A printed circuit board was then ordered and populated. The resulting board can be seen in Figure 3.9. Four copies were made for the board that was sent out. Issues discussed in the Communication Module section forced these copies to be generated in the schematic capture and laid out four times. After populating the board, issues with the 16 driving pulses were discovered. The board was then changed to represent the circuit in Figure 3.3.

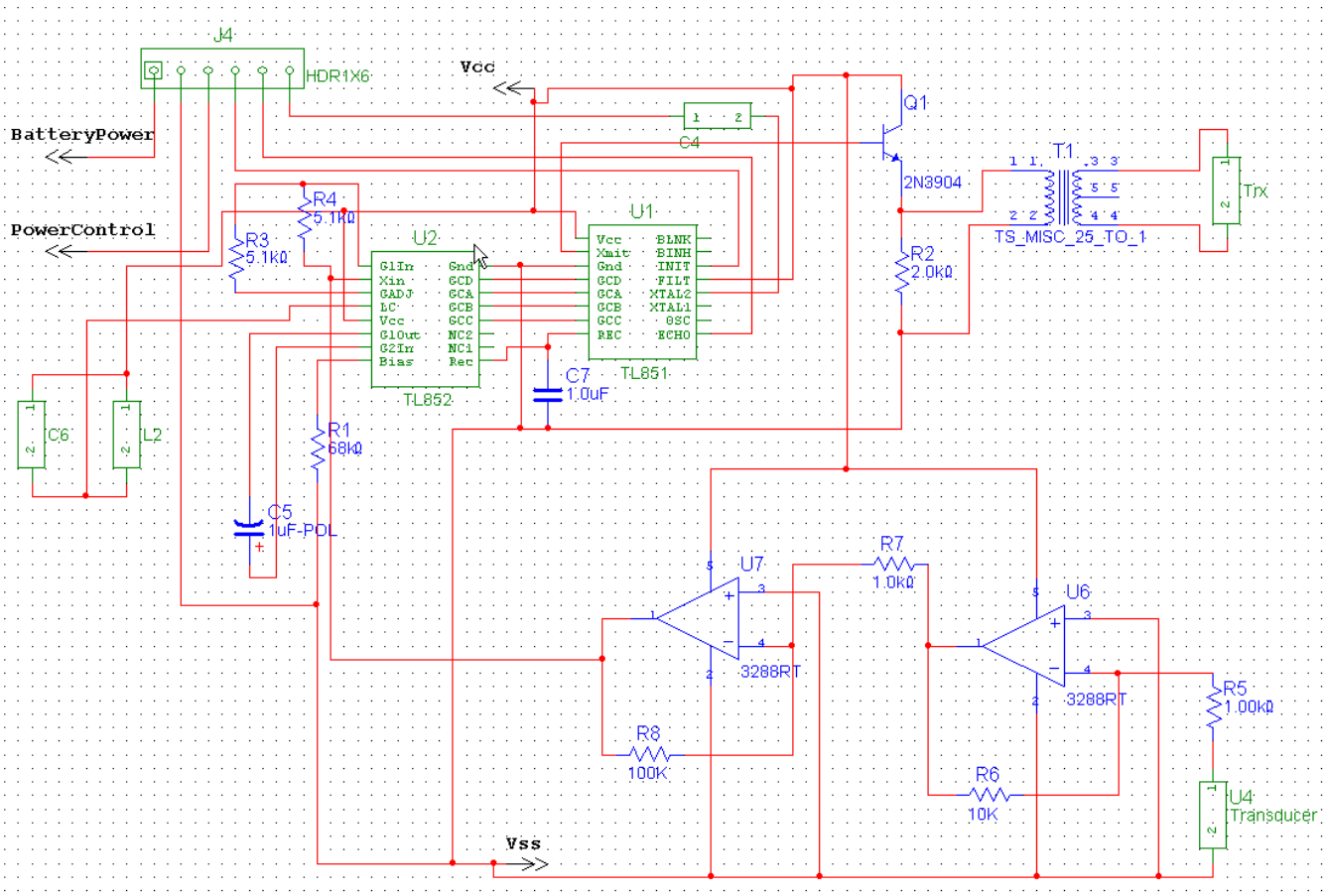


Figure 3.8: Sensor Schematic

3.2 Communications Module

3.2.1 Overview

The communications module sits on the bumper of the car and is connected to the four sensors. It drives the sensors and transmits the results wirelessly to the display module. This is also the module with the battery recharger. An overview of the Module can be seen in Figure 3.10

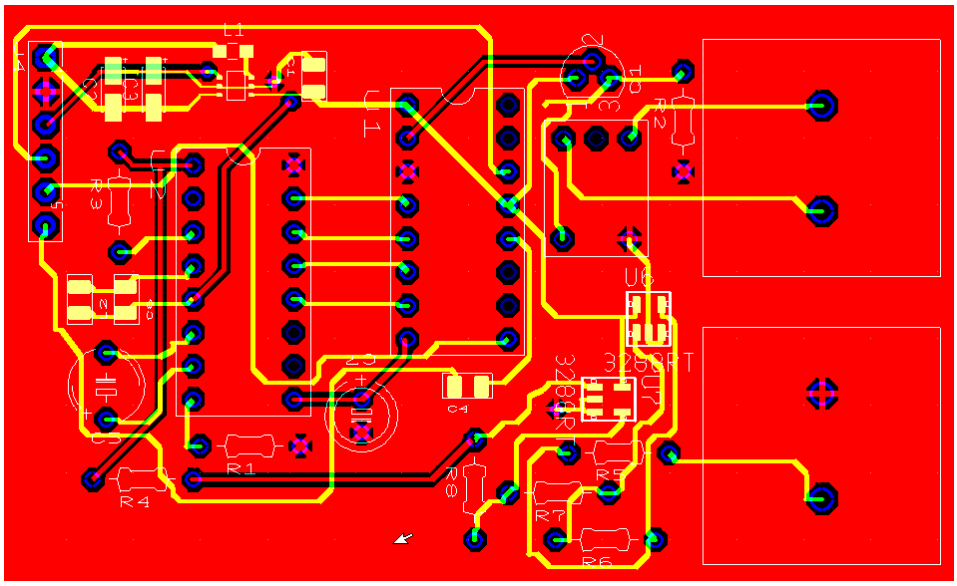


Figure 3.9: Sensor Printed Circuit Board

3.2.2 Wireless Communications

Originally a wireless module was to be developed by this MQP team. Design started centered around Chipcon's CC1000. Due to stringent requirements in RF design, a printed circuit board would have needed to be ordered early in the design process. The team also did not have experience in antenna design, an important aspect for this 900 MHz chipset. During the layout process a complete module was found, MaxStream's XBee module with onboard chip antenna. This solution provided communications in the unlicensed 2.4 GHz ISM (Industrial, Scientific and Medical) radio band, without complications involved with a homemade antenna. The board also has header pins, making prototyping easier. The pins were at a 2mm pitch, not the usual 1/16", so it could not be directly placed on a breadboard. The module only needed power and ground and a two pin UART interface to the microcontroller. More advanced features of the chip could be accessed through "AT" commands.

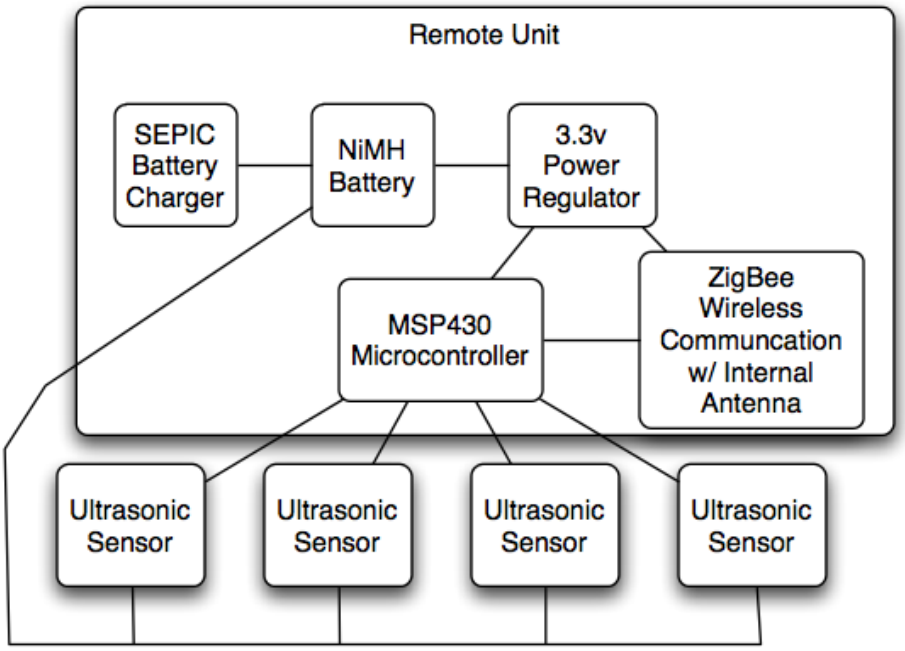


Figure 3.10: Communications Module Overview

3.2.3 Microcontroller

Texas Instrument's MSP430 line of microcontrollers provides a plethora of options. The design process in the previous chapter resulted in the choice of the MSP430F149. In the inital stages, a breadboard of the communications module was to be built, but only surface mount packages are available for the MSP430 line of chips. A breakout board was ordered from Sparkfun.com to enable the use of a breadboard. Availability of breakout boards on the Sparkfun site was limited to few MSP430 and forced us to upgrade to the MSP430F169. This board included JTAG pins and an onboard 32.768 kHz oscillator crystal. The board can even be powered over the JTAG connection, though we found not reliably, but helped with initial testing of code. The connection between the microcontroller, ZigBee module and sensors can be seen in Figure 3.11. The sensors are plugged into each of the headers seen in the schematic and the off page connections show where the connections to the voltage regulators are made.

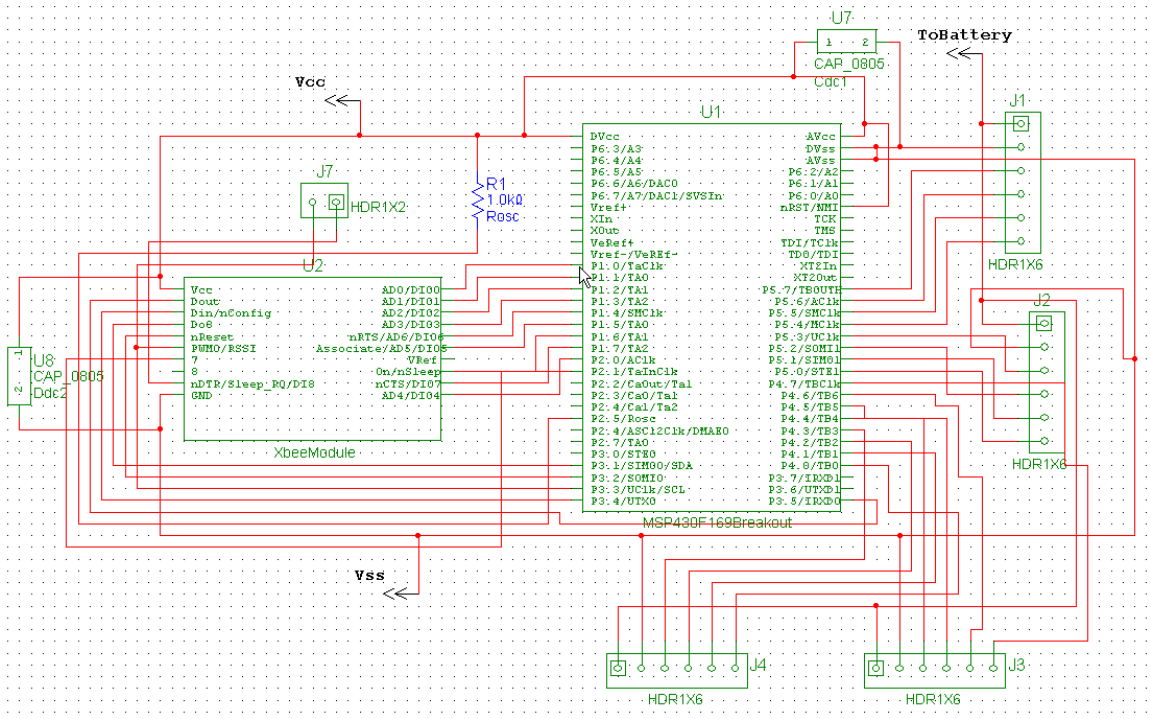


Figure 3.11: Microcontroller Schematic

3.2.4 Program

The programs running on each bumper module are largely the same. The differences are centered around addressing of the two modules. The program goes through a short initialization routine and then infinitely loops in the main function. These programs can be found in Appendix A and are written in C in the IAR Embedded Workbench. The program utilizes JTAG programming and debugging communications. The overall program flow described can be seen in Figure 3.12.

Initialization

The initialization of the bumper circuits is very similar to the initialization of the display module with few differences. The watchdog timer is disabled to enable JTAG debugging and the GIE flag in the status register is set. Three I/O ports for each sensor along with a clock signal with a frequency of 340 kHz are used for each sensor. The clock signal is

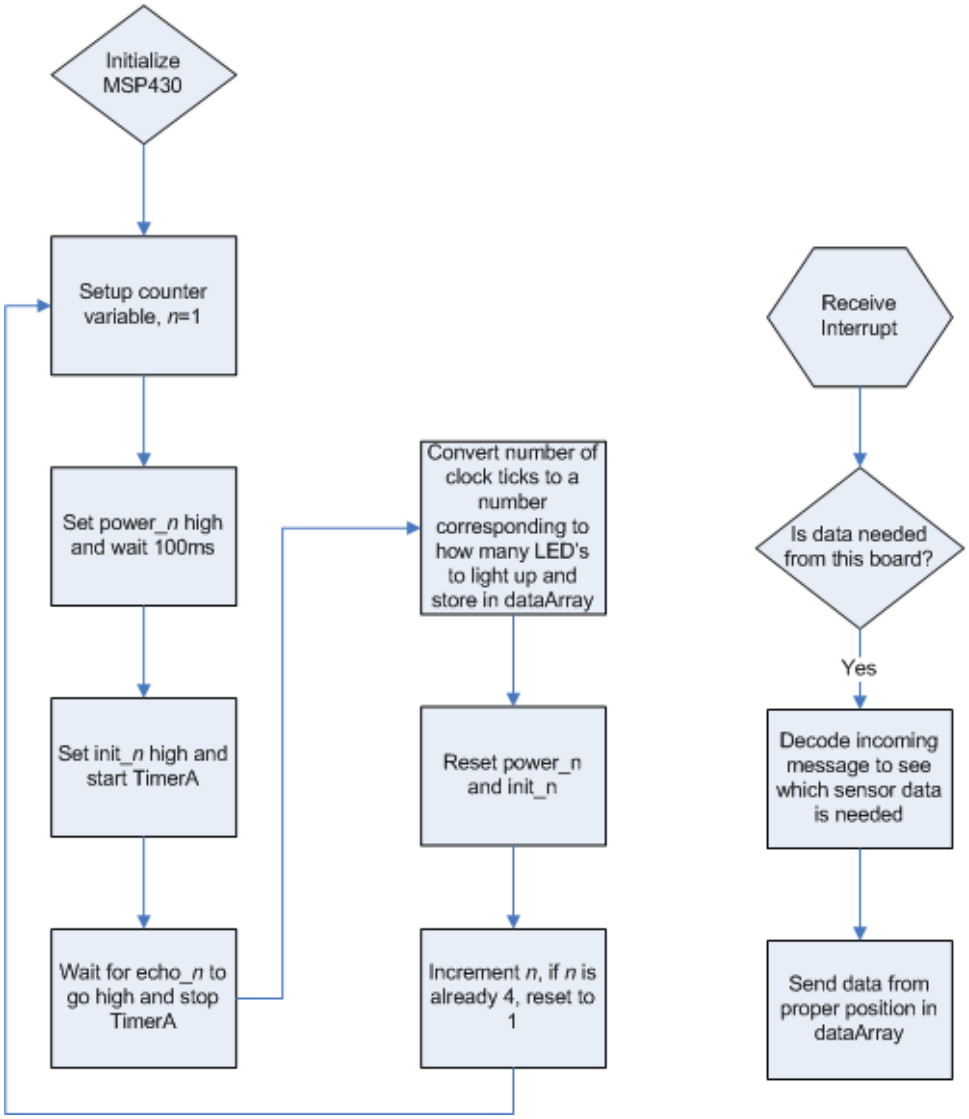


Figure 3.12: Communications Module Code Flow Chart

created by the internal Digitally Controlled Oscillator (DCO). The DCO can be adjusted using one of seven internal frequencies, selected in the DCOCTL register. This frequency can be more finely tuned using one of seven internal resistances, set in the BCSCTL1 register. The chart in Figure 3.13 displays the frequencies available (DCO0-DCO7) for each of the internal resistances (RSEL0-RSEL7). In order to select a frequency, you can either go along the bottom axis to choose a frequency and then choose a resistor value to get the frequency desired (left axis). We found it easier to first choose a resistor value and fine tune the frequency using the DCOCTL register. A divider can also be used for the clocks to achieve

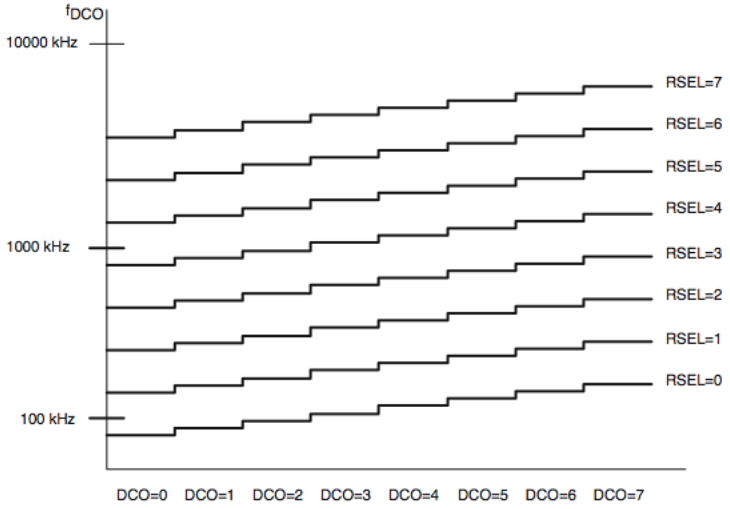


Figure 3.13: DCO Frequencies by Internal Resistance [20]

frequencies that can not be directly set. The DCO can be output on one of two output pins, or can be used internally. One of those output pins is then connected to each of the four sensors. The other three pins for each sensor are set as inputs/outputs. These signals are: power (output), init (output), and echo (input).

Main Function

The infinite loop in the main function is responsible for getting data from the sensor circuits. To take a sensor reading, the power signal is set high, powering up the sensor

circuit. After a 5 ms delay, the init pin is set high, sending the audio impulse. The timer on the MSP430 starts counting at this point and stops counting when a high signal on the echo pin is received. The timer data is then sent into a function determining which of the 10 LEDs should be illuminated on the display circuit. This data is stored to an array for use by the receive interrupt of the bumper circuit.

Receive Interrupt

The receive interrupt is used to signal when to send the data for a given sensor back to the display circuitry. Upon receiving a signal to send back data, the receive interrupt retrieves data from the data vector created by the sensor functions and sends it over the UART. Addressing data is added to the display data to let the Display Module know which LED bank to display the data on. Each character sent uses the most significant nibble to designate which sensor the data is for, and the least significant nibble contains the data corresponding to which LED to light up. In addition to simply receiving data, the receive interrupt is also responsible for putting the bumper circuit into low-power mode if the inside display circuit is turned off. By using Timer A, we can keep track of how often the MSP430 is being polled. If three seconds go by without any wireless activity, the device is placed into Low Power Mode 3 (LPM3), which disables the CPU, Master Clock, Submaster Clock, Digitally Controlled Oscillator, and the DC generator. One clock remains active to allow the device to be brought out of LPM3 using the receive interrupt.

3.2.5 Voltage Regulation

The voltage regulator used is in the same line as the voltage regulator on the sensor circuit. The preset voltage is different, so the particular chip selected this time is the LTC3525-3.3, for the 3.3 volts needed for the microcontroller and wireless module. See Section 3.1.4 for further details. The only difference is that the Power control pin is tied to the battery, holding the regulator on all the time.

3.2.6 Battery Charging

Being unable to find an effective onboard way to charge the module (such as solar or vibration energy sources), a battery charger that could be plugged into the wall was developed. Figure 3.14 shows the schematic for the charger based around Linear Technology's LT1512. It takes advantage of the fact that the DC current through L2 is the same as the DC current

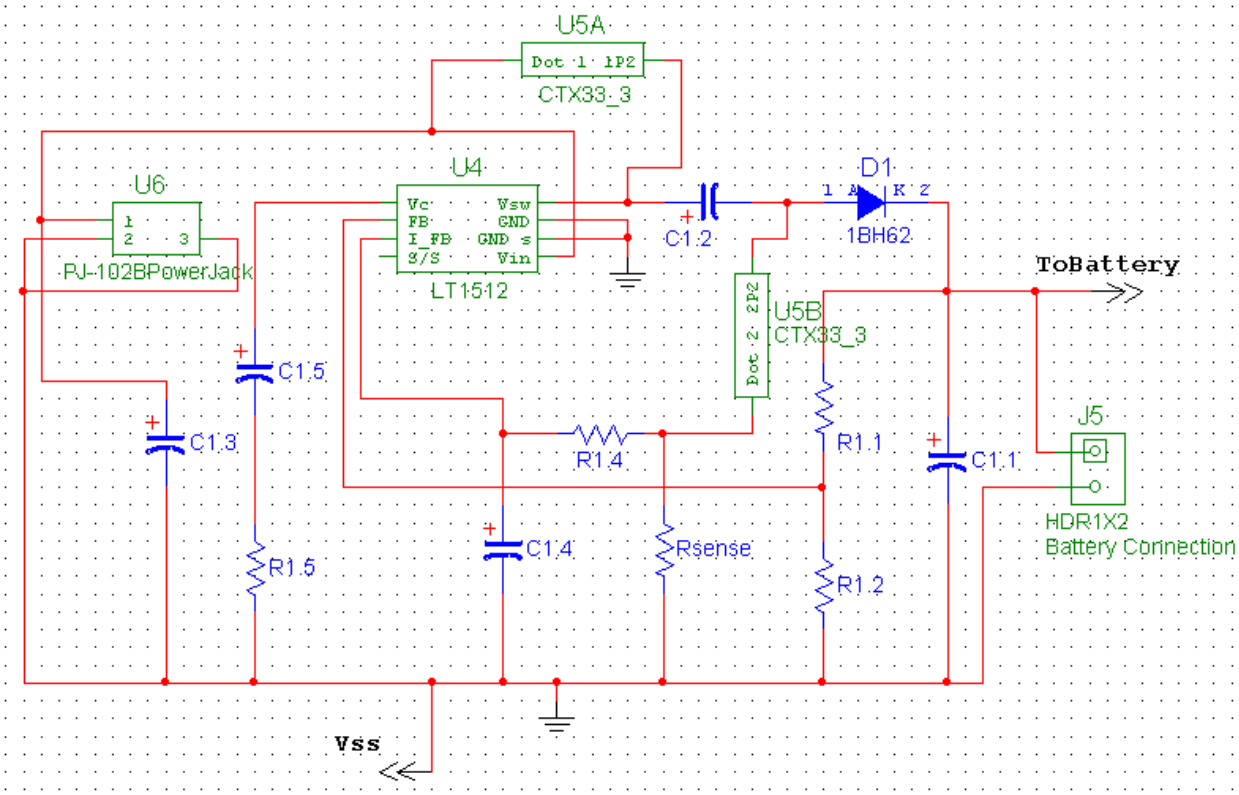


Figure 3.14: Battery Charger

applied to the load. By placing a small sense resistor between L2 and ground the current into the battery can be measured. The value of the sense resistor determines the maximum current that can be delivered to the battery. The voltage across the resistor is compared to 100mV, so if the desired current is 450mA, then the resistor value would have to be 0.222Ω. The closest value available is a 0.2Ω resistor which will give a charging current of 500mA which is acceptably high.

The duty cycle of the pass transistor will be adjusted based on the voltage at the Ifb pin. Because a battery has a low internal resistance, a voltage divider using large resistors can be used to detect the battery voltage without a significant amount of leakage current. This voltage divider is set such that the divided voltage is compared to an internal voltage reference of 1.245V. For NiMH batteries the nominal charging voltage is actually 1.41V rather than the 1.2V. This means that the target battery voltage for the two battery cells is 2.8V. Using Equation 3.4 shown below we can select R2 and calculate R1.

$$R_1 = \frac{R_2 \times (V_{BAT} - 1.245)}{1.245 + (R_2 \times 0.3\mu A)} \tag{3.4}$$

Using R1 = 15.4kΩ R1 ideally equals 19.4kΩ. The closest available value that wont make the charging voltage too high was 19.2kΩ, which will give a shutoff voltage of 2.8V. Once this voltage is reached the charger will stop charging.

3.2.7 Printed Circuit Board

After developing a strong revision of the schematics, a Printed Circuit Board was ordered. Instead of ordering new microcontrollers to be soldered directly onto the board, we re-used the breakout boards used in prototyping. A socket also needed to be created for the XBee module. The final layout can be seen in Figure 3.15. When the boards were ordered three copies were to be made. The copies were simply cut and pasted in Ultiboard, saving the trouble of laying out the circuit three time, but preventing Ultiboard from running connectivity checks on the two copies of the original. Ultiboard correctly connected the original circuit's connections to ground to the ground plane, but the copy paste process did not copy the ground plane connections. This problem was not discovered until the boards had been made. Only two copies of the board were needed, so testing began with the properly ground connected board while each ground connection was manually made on one of the copies.

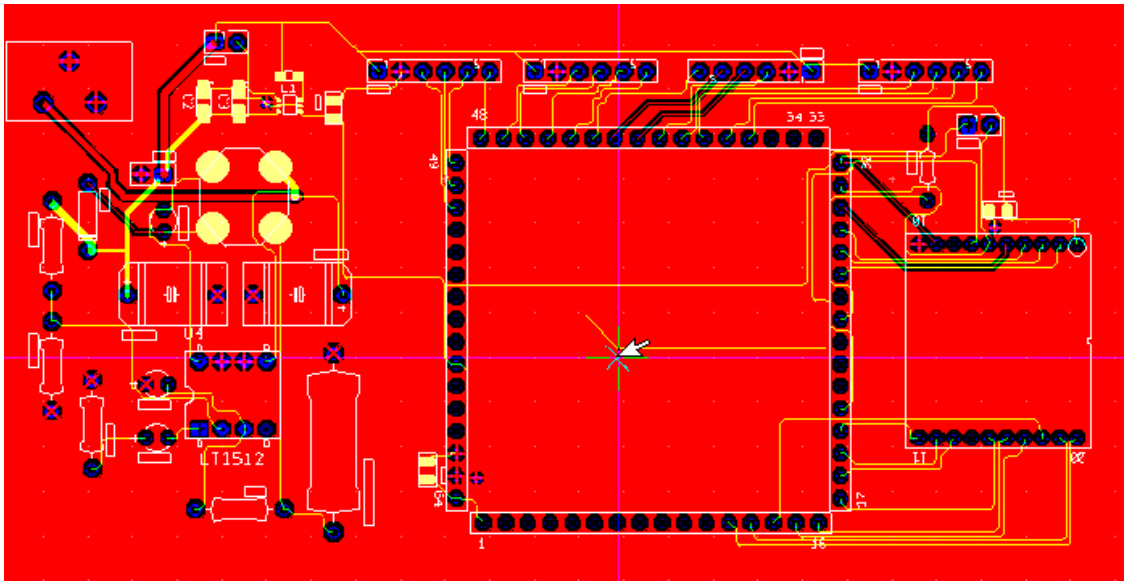


Figure 3.15: Communications Module PCB

While the voltage regulator was tested and used elsewhere in the system, one of the voltage regulators was not functioning properly on the PCB. After extensive connectivity testing on the board and replacing assorted parts, it was discovered the broken connection was the ground pin of the battery header and the board. After reflowing this connection the regulator resumed normal operation.

We used the suggested layout given in the data sheet for the battery charging circuitry, but there was apparently a reversal in the direction of the transformer terminals used for the second inductor. After making this correction the charging current was about half of what the data sheet specified, so in order to fix that we could change the sense resistor to a 0.1Ω resistor. However, the 0.1Ω sense resistor we have has connecting wires that are too thick to fit through our the through holes on the PCB. This correction could be made in future revisions, but for now the charger will still work with the charging current of 250mA.

3.3 Display Module

3.3.1 Overview

The unit inside the car, which plugs directly into the car's power system, displays the sensor readings in a way that does not distract the driver, but alerts him or her to surrounding objects. Power requirements are no longer a concern because of the large amount of power made available by the car. The circuit needs to receive messages from the Communications modules and display the results for each of the eight sensors. This resulted in a module depicted in Figure 3.16.

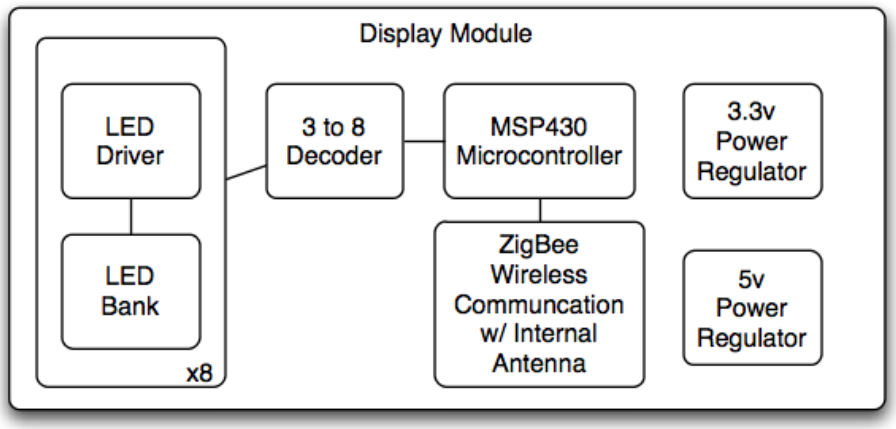


Figure 3.16: Display Module Overview

3.3.2 LEDs

The LEDs are driven using the LM3914 LED Driver by National Semiconductor. The driver's job is to take a signal from the MSP430 and determine how many of the ten-LEDs to illuminate. By using low and high reference pins on the LM3914, the range of voltages the input signal will be within is specified. By setting the low reference to be the ground of the entire system and the high reference to be the highest voltage the MSP430 will output, the driver will interpret the signal being output by the DAC.

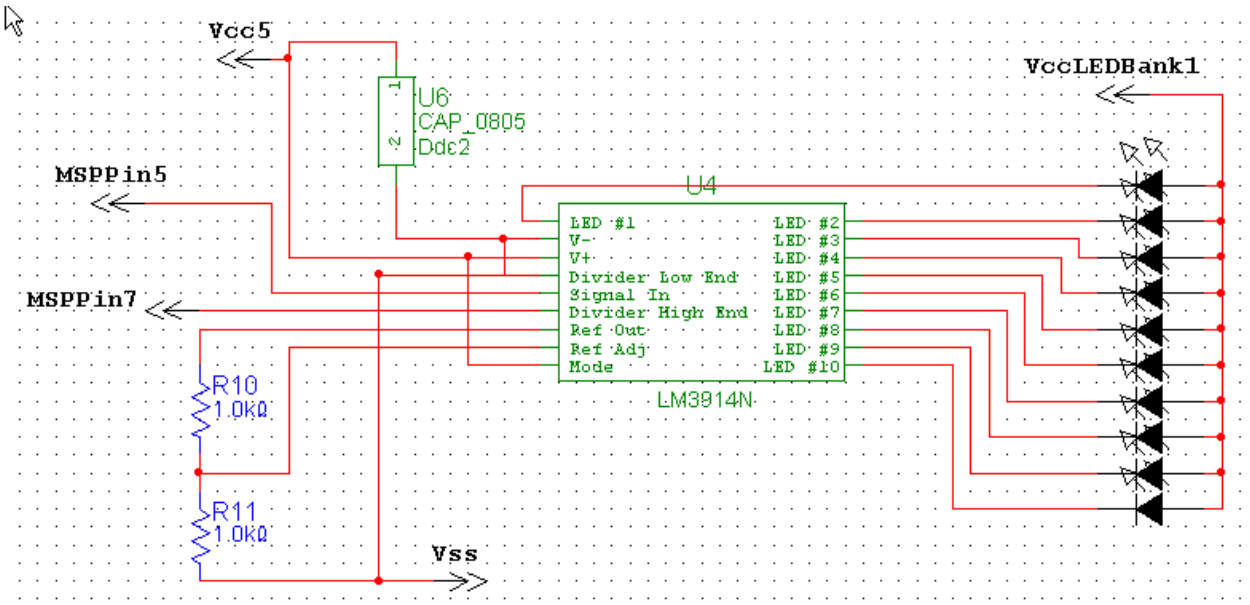


Figure 3.17: LED Bank Circuitry

The signal from the DAC is always sent to all eight LED drivers, the bank of LEDs to illuminate is selected though multiplexing circuitry. The multiplexing circuitry will control which of the banks of LEDs are connected to V_{cc}. By using software delays and placing all of this into a software loop, the active banks cycle fast enough so that it appears that all banks are illuminated at the same time. Figure 3.17 shows one of the banks of LEDs. The drivers use open collector inputs with the current set using R_{10} and calculated using Equation 3.5.[11] Using a $1k\Omega$ resistor for R_{10} results in a 12.5 mA current draw for each LED lit.

$$I_{LED} = \frac{12.5}{R_{10}} \tag{3.5}$$

3.3.3 Multiplexing

Multiplexing refers to the process of sending multiple signals on one line. It is normally done in a way such that it can easily be demultiplexed and used by a specific component. Multiplexing in this circuit is controlled by the MSP430 by sending signals to a 74LS138 3-to-8 decoder. The truth table shows that this device is active low, meaning that all output

INPUTS						OUTPUTS							
$\overline{E_1}$	$\overline{E_2}$	E_3	A_0	A_1	A_2	$\overline{O_0}$	$\overline{O_1}$	$\overline{O_2}$	$\overline{O_3}$	$\overline{O_4}$	$\overline{O_5}$	$\overline{O_6}$	$\overline{O_7}$
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	L	X	X	X	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	L	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	L	H	H
L	L	H	L	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	L

H = HIGH Voltage Level
L = LOW Voltage Level
X = Don't Care

Figure 3.18: Decoder Truth Table [10]

pins will be high except for the active pin, which will be low. By using this device, we can distinguish between 8 different signals controlled by the inputs $A_0, A_1, and A_2$, connected to I/O pins of the MSP430. The MSP is configured through code to output values 000 to 111 depending on which signal is being output through the DAC.

Figure 3.19 shows how the multiplexing circuitry controls the positive 5 volt rail to each LED bank. The outputs of the multiplexor are connected to the bases of PNP Bipolar Junction Transistors (BJTs). With the collector of the BJT tied to the 5 volt rail and the collector connected to the cathodes of the LEDs, current will be allowed to flow from collector to emitter when the base is low, illuminating the LEDs. When the base is high, there will be no current flow, preventing that bank from being driven by the current signal generated by the DAC.

3.3.4 Microcontroller

The microcontroller is the same MSP430F169 as discussed in Section 3.2.3. For this module the DAC is also utilized. An internally set reference voltage is output to each of the LED drivers, a value is then set in the DAC which outputs that percentage of the reference voltage, up to 100%.

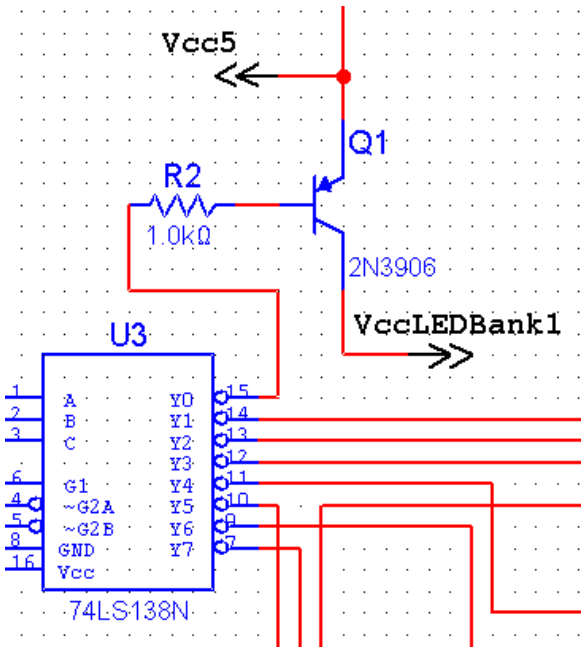


Figure 3.19: V_{cc} connection to LED Drivers

3.3.5 Wireless Module

The wireless communications on the display module are the same as the Communications module, as discussed in Section 3.2.2.

3.3.6 Program

The main processing unit of the system is the MSP430 on the interior display. Although the sensor circuits also contain MSP430 microprocessors, they are controlled wirelessly by the interior unit. This section will go over the startup and main functions of the MSP430 on the interior display. The Flow of the overall program can be seen in Figure 3.20, the code can be found in Appendix B.

Initialization

The initialization of the MSP430 is handled by the multiple functions called in the init_sys() function. The first action is the watchdog timer is disabled. This is done be-

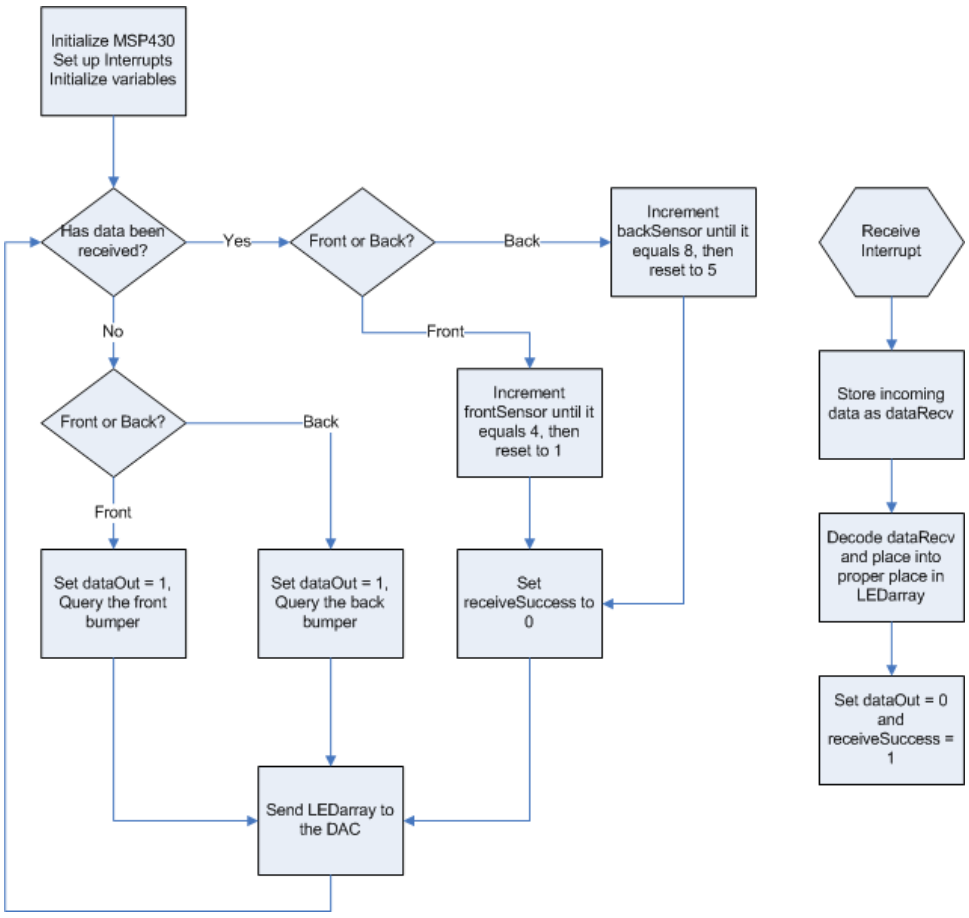


Figure 3.20: Display Module Program Flow

cause when connected to the JTAG interface, the watchdog timer can interfere with the debugging feature in the IAR Workbench. The MSP430 is also able to handle certain interrupts. In this system, the only three interrupts used are the Timer A interrupt, the Timer B interrupt, and the UART receive interrupt. In order for these to be able to function, we must set certain bits in the status register (the Global Interrupt Enable bits). This is done by the _BIS_SR(GIE) command. The Universal Asynchronous Receiver/Transmitter (UART), responsible for sending and receiving data to the wireless unit, needs to be configured. This configuration is handled in the configUART() function. This function sets up the proper ports to be used as UART transmit and receive rather than their peripheral functions. It also sets the character size to be 8 bits and the baud rate to 2400. The last thing it does is enables the interrupt for the receiver, which stores the input of the receive buffer when it fills. Most of the different input/output (I/O) ports have two different functions, either as I/O, or some peripheral function. The only I/O ports used on the interior display are the four pins that control the inputs to the 3 to 8 decoder. The function configExtPins() sets up Ports 4.3-0 to be used as digital outputs, three used to encode which LED bank to light up, and the fourth as an enable signal for the 3 to 8 decoder. The next initialization function is configDAC. This function sets up the Digital to Analog Converter, which is used as the signal input for our bar graph drivers. The DAC uses the internal Analog to Digital Converter (ADC) reference voltage of 2.5 volts, and uses three times the ADC voltage as the max range to compensate for the effects of multiplexing the signal to the bar graph drivers. That is, the drivers were not getting a full 2.5 volts because it was taking the average DC value on the signal line. By having the maximum voltage reference for the drivers be 2.5 and the output of the DAC be 7.5, we can easily attain a value whose multiplexed DC output would be the 2.5 volts needed to light up all 10 LEDs. The final initialization function, configZigBee(), is used to send the initialization commands to our X-Bee module. It uses AT commands to first put the module into command mode. While in command mode, it simply configures the device to be used as a Coordinator. This means that the module on

the Display Circuit will be in charge of the the two bumper modules (end-devices). After doing this, the MSP430 simply sends the command to the X-Bee module to exit command mode. Figure 3.21 shows an example of how to send a command to the X-Bee module.

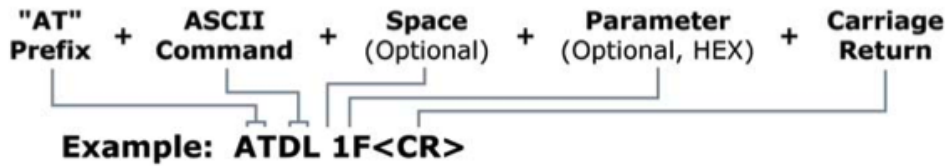


Figure 3.21: ZigBee Command Example [9]

Main Function

Rather than go through every step of the main function, this section will provide an overview of how the system works. After initialization of the MSP430, the wireless communication to the sensor modules begins. Wireless communication is implemented with a simple method of flow control and addressing mode. We can assume that all data sent out of the UART is wirelessly transmitted to the other two modules. The first thing to happen is that the MSP430 sends a character of data to ask for information from one of the two sensor boards. The most significant nibble represents the module that should respond. The least significant nibble indicates which sensor on that module the data should come from. For example, if 0x12 is sent out over the UART, sensor module 2 on the front bumper should respond. If 0x24 is sent, the back bumper should respond with data for sensor 4 on that board.

After sending the data out, the MSP430 keeps track of whether or not it has received a reply. If by the next time the loop occurs the MSP430 has not received a reply, the data is retransmitted. Only when a reply has been received, the variable keeping track of which sensor to query will change. This is a very simple implementation of stop-and-wait flow control. The data received is set up in such a way that it is known which sensor the data is from. The most significant nibble represents the position in a stored LED array that the

data belongs. The least significant nibble represents which LEDs should be illuminated.

Regardless of receiving a reply, our display needs to be updated. The previous values of the data received are kept in an array. Every time the loop occurs, this data is sent to the display circuit by first disabling the 74LS138, sending a hex code to the 138, sending a signal to the LED driver, and enabling the 138. There are also some short delays in this function to make sure the LEDs remain lit long enough to be visible.

3.3.7 Voltage Regulation

Being powered off the car's alternator and battery, the display circuitry is not limited by power effiency. The microcontroller and wireless modules both need 3.3 volt supply, buthe the multiplexing circuitry and LED drivers need a 5 volt supply. The 5 volt supply needs to be able to handle the current requirements of the LEDs. Linear Technology's LT1962-5 can handle the input voltage and can output as much as 300 mA. It needs few external capacitors and a variety of capacitor types can be used. Figure 3.22 shows the suport circuitry needed for both the 5 volt and 3.3 volt regulators. The 3.3 volt regualtor needs to only power the wireless module and microcontroller, copied from the low power deisgns needs a very small output current. Linear Technolgoy's LT1761-33 was used to supply the 3.3 volt rail.

3.3.8 Printed Circuit Board

Generating a PCB for this module was relatively straight forward. When using Ultiboard to design the PCB, there are many different parts to be used in the library. However, when trying to find the corect model for one of the voltage regulators, a generic model needed to be used. This model had the wrong footprint and was too large for our actual part and was modified using copper tape. The footprint for the LEDs were much larger than the LEDs used for this project, resulting in a much larger PCB than necessary. Figure 3.23 shows four of the LED banks, the wireless module, XBee and voltage regulators.

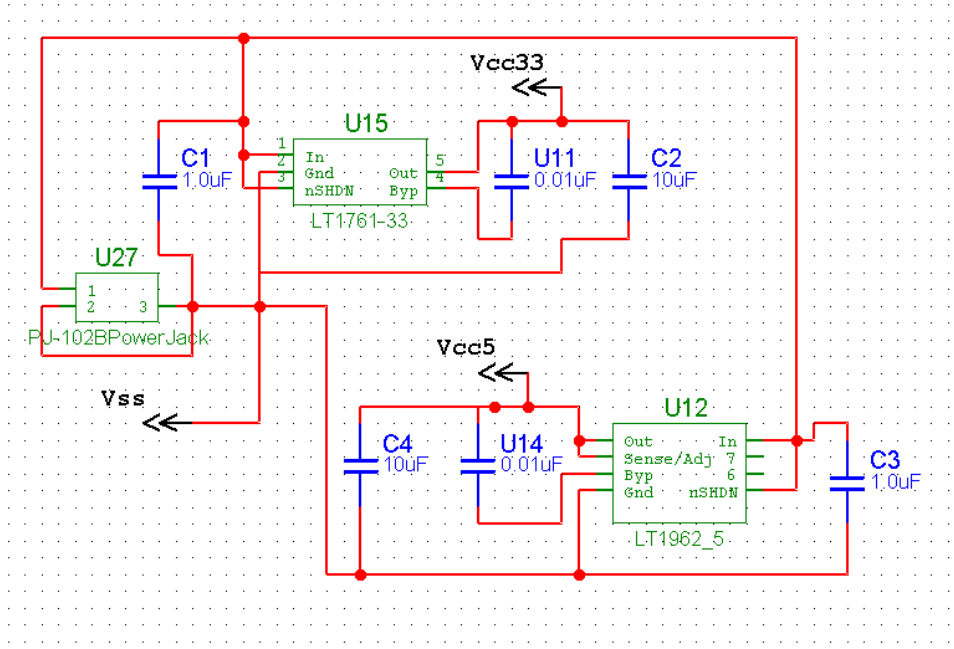


Figure 3.22: Display Module Voltage Regulation

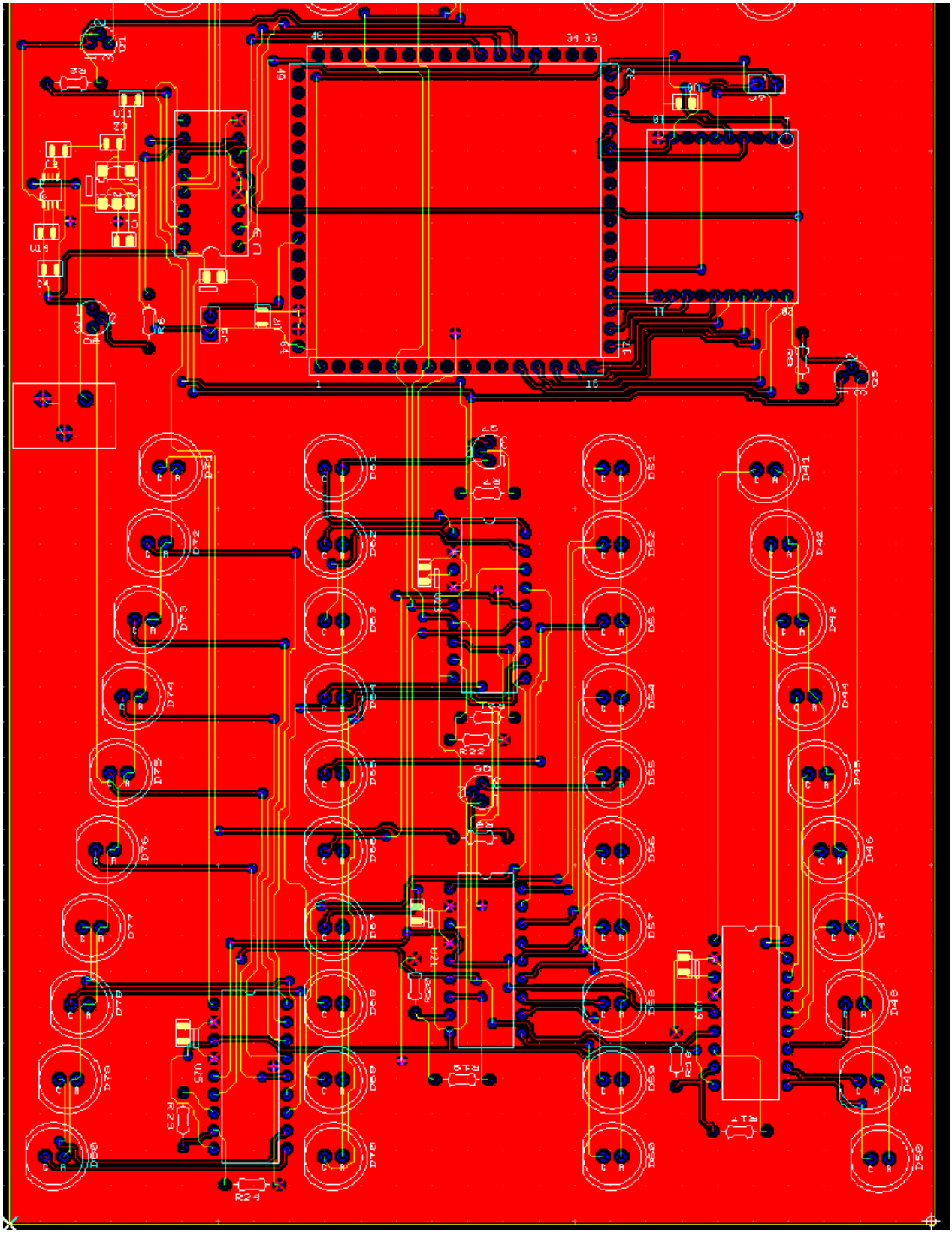


Figure 3.23: Display Module PCB

Chapter 4

Results

4.1 Sensor

The populated sensor board can be seen in Figure 4.1. After populating the Sensor board, power and ground were connected. The power control line was then set high and some magic smoke escaped from the board. Originally, the power regulator was suspected as the source, but some testing eliminated that component. It was later found that the second Op-Amp stage was the source of the smoke. The entire stage was removed to simplify the circuit during the testing process. The transmit portion of the circuit to the transformer, but the transformer output was not as was expected. The issues with the transformer remain unresolved. With an inability to send a signal, the receive circuitry was difficult to test. Basic tests indicate the receive circuitry is operating as expected. While we successfully demonstrated a sensor circuit as a breadboarded prototype, a working PCB version was never completed.

4.2 Battery Charger

The two battery charing circuits were populated on the board. The second board needed all connections to the ground plane to be jumped and re-connected due to the PCB issues

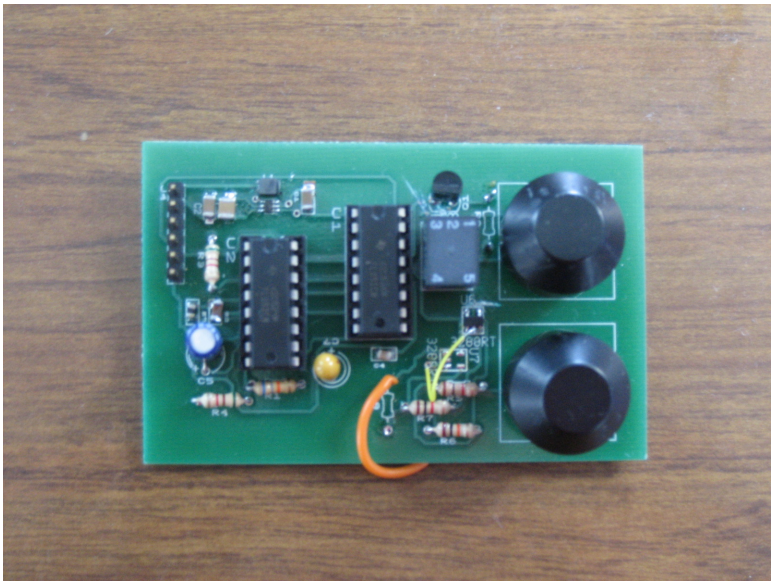


Figure 4.1: Sensor Circuit On PCB

dicussed earlier. The circuit was then tested and the charging current was about half of what the data sheet specified, so in order to fix that we could change the sense resistor to a 0.1Ω resistor. However, the 0.1Ω sense resistor we have has connecting wires that are too thick to fit through our the through holes on the PCB. This correction could be made in future revisions, but for now the charger will still work with the charging current of 250mA.

4.3 Wireless Communications

As parts were received and PCBs were populated, individual portions of the project were tested for basic functionality. The first thing that was tested was the MSP430/XBee Module board. The results of this test showed that both the MSP430 and the XBee module were properly being powered by the battery pack. Simple test code was run to make sure the MSP430 was properly communicating with the XBee module. With this connection functioning properly, the next step was to test that wireless communication was working. This was tested by sending a message wireless from one board to the other. Upon receiving a wireless message, the other board would reply back with the same message with the

characters OK added to the end. This was also a successful test. Then integration was tested. To do this, we used the sensor boards to generate dummy data to send to the display unit. The display board properly displays the data received from the sensor boards for a few minutes. After this, one of the sensor boards stops responding, causing the display circuit to loop on that sensor. The likely problem is that one of them loses power or it is falling through the code and reaching an end.

Possible ways to test this would be to get three JTAG programmers on three separate computers so that the debuggers would be able to run and set breakpoints to see exactly where the programs are getting hung up. Another improvement that could be made is to make the display board notify in a way (maybe by flashing LEDs) when a specific sensor or entire bumper is not responding.

The Battery Charger and Wireless Communications were both combined to form the Communications Module, as seen in Figure 4.2. The batteries are attached by jumper pins, as the sensors would be as well. Two of these boards were populated and are working.

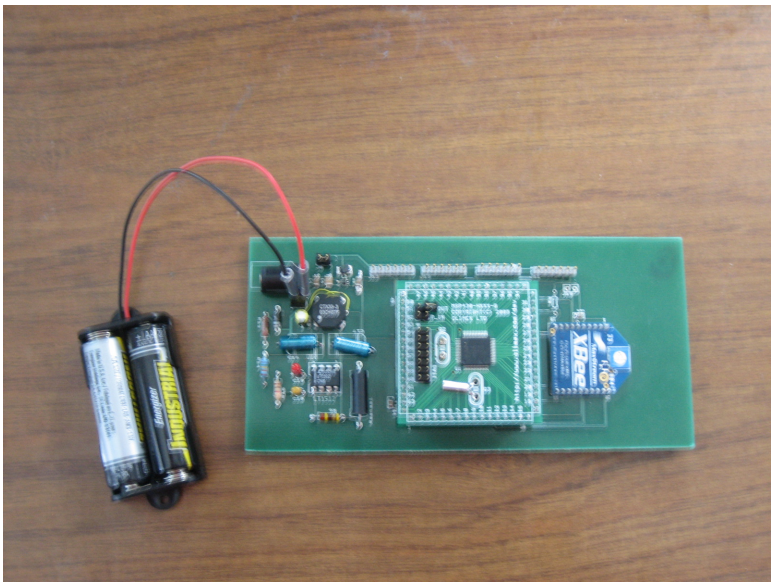


Figure 4.2: Communications Module On PCB

4.4 Display Module

The Display Module was tested in stages. First, one LED bank was tested. This test revealed problems in the code with using the 3 to 8 decoder for multiplexing. This was also the test were the limits of the DAC was discovered. To light the tenth LED, the input voltage has to be higher than the reference voltage. The MSP430's DAC outputs a percentage of the reference voltage, with 100% being the maximum. This prevented the current setup of the board from lighting the tenth LED. Changing the way the reference voltage is set up in the MSP or putting a voltage divider on the reference voltage output are possible solutions, but were not implemented. After these issues were addressed, each LED bank was tested individually to check the addressing and for any faulty soldering. The last test incorporated wireless communications. Basic wireless communications were tested with the Communications Module. For the wireless test, random data was generated on the Communications Modules and then transmitted to the Display Module to be displayed. This data was correctly transmitted and displayed as the corresponding sensor. Figure 4.3 shows the resulting PCB.

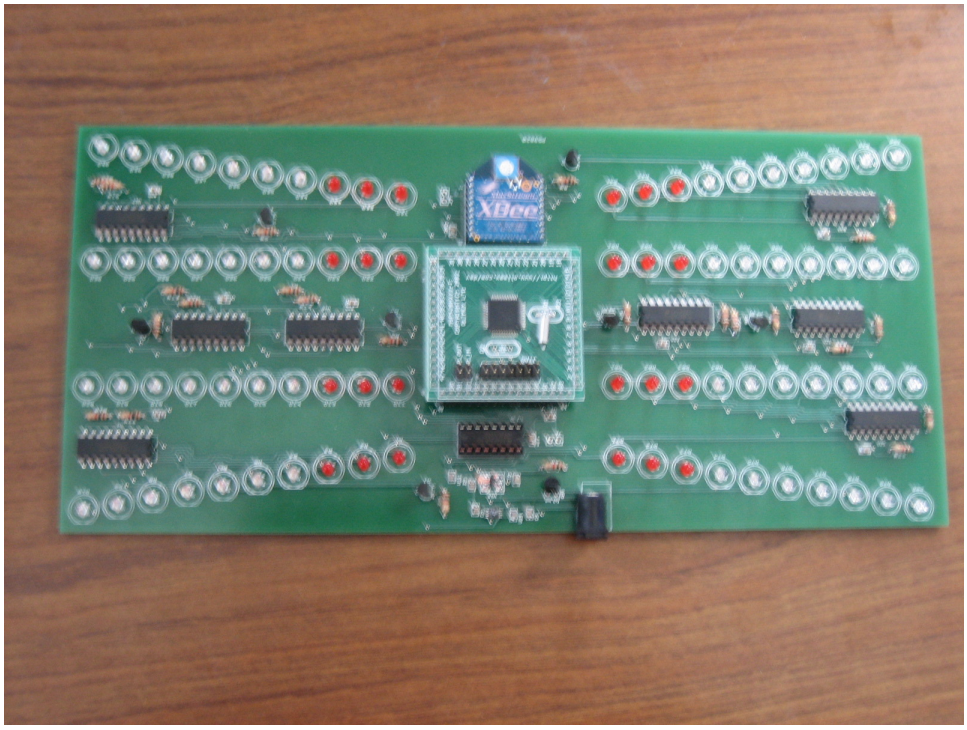


Figure 4.3: Display Module On PCB

Chapter 5

Conclusion

As is intended, the MQP is an extensive learning process. When this project began, PCBs were planned to only have one revision. Due to problems that are often difficult, sometimes impossible to foresee, multiple revisions of a board should be planned from the onset of the project. Stricter time tables needed to be enforced to ensure the project would be completed on-time.

The effort of commercializing this project would include development of an enclosure and a way to attach the bumper units to the car without doing damage. This system was designed with prototyping in mind. Almost all of the through hole components can be replaced with surface mount analogs. An in-house wireless communications system could also be developed further reducing the physical size of the device. The JTAG interface would also not be needed for the commercial version of the project.

Future revisions of this project should include a more dynamic display, making the system more versatile. More dynamic code and the ability to plug multiple sensors into one set of headers would allow more than four sensors to be connected to each communications module. Further research should be done on a self-contained source of energy for the communications modules. As existing systems improve and new ones are developed it may become possible

to eliminate plugging the module into the wall to charge it. The wireless communications can also be used for car to car communication. Other ultrasonic impulse-echo sensors can be used instead of the one developed in this project.

The communications and display modules are currently working as described. The communications modules can drive the ultrasonic sensors and send the data back to the display module wirlessly. The display module polls the communications modules for data and correctly displays this information on the LEDs. If off the self sensors are plugged into the communications module, the system will work as desired, but the sensor developed for this project does not work on the PCB. The sensor circuit did work in the breadboarded prototype. These off the self sensors would require additional voltage regulators on the communications module. This project will bring driver awareness to a much larger group of people; preventing parking accidents and reversing over children or their toys.

Chapter 6

Bibliography

Bibliography

[1] R. Amirtharajah and A. Chandrakasan. Self-Powered Signal Processing Using Vibration-Based Power Generation. *IEEE Journal of Solid-State Circuits*, Volume 33, 1998.

[2] APC International, Ltd. 40CA-18ST/40CA-18SC/40CA-18SR Ultrasonic Transducer. Retreived from `http://www.americanpiezo.com/products_services/air_transducers.html`.

[3] L. Dixon. High Power Factor Preregulator Using the SEPIC Converter, 1993.

[4] Energizer. Rechargebles-Sizes. Retreived from `http://www.energizer.com/products/rechargeables/sizes.aspx`, 2006.

[5] Linear Technology. Battery Charger Solutions. Retreived from `http://www.linear.com/pc/downloadDocument.do?id=10777`, 2007.

[6] M Lund. Battery Technology Comparison – Rechargeable. Retreived from `http://www.powerstream.com/Compare.htm` from Powerstream Technologies, 2007.

[7] M Lund. NiMH Battery Charging Basics. Retreived from `http://www.powerstream.com/NiMH.htm` from Powerstream Technologies, 2007.

[8] Maxim/Dallas. DC/DC Conversion without Inductors. Retreived from `http://www.maxim-ic.com/appnotes.cfm/appnote_number/725`, December 29 2000.

[9] MaxStream, 355 South S20 West Suite 180, Lindon, UT 84042. *XBee/XBee-PRO OEM RF Modules*, 2006.10.13 edition, 2006.

[10] Motorola. 1-Of-8 Decoder/Demultiplexer. Retreived from `http://www.chipdocs.com/datasheets/datasheet-pdf/Motorola/SN54LS138.html`, 2006.

[11] National Semiconductor. LM3914 Dot/Bar Display Driver. Retreived from `http://www.national.com/pf/LM/LM3914.html`, 2003.

[12] Panasonic. Nickel Metal Hydride Handbook: Precautions for Designing Devices with NiMH Batteries. Retreived from `http://www.panasonic.com/industrial/battery/oem/images/pdf/Panasonic_NIMH_Precautions.pdf`, 2005.

[13] J. Paradiso and T. Starner. Energy Scavenging for Mobile and Wireless Electronics. *IEEE*, 2005.

[14] Piezo Systems Inc. Frequently Asked Questions. Retreived from `http://www.piezo.com/tech3faq.html#app4`.

[15] Piezo Systems Inc. Pricing Information - Piezo Sheets. Retreived from `http://www.piezo.com/prodsheet2sq5H.html#priceinfo`, 2007.

[16] Rincon-Mora. "A Low-Voltage, Low Quiescent Current, Low Drop-Out Regulator". *IEEE*, Volume 30, 1998.

[17] Texas Instruments. TL851 Sonar Ranging Control. Retreived from `http://focus.ti.com/docs/prod/folders/print/tl851.html`, March 1988.

[18] Texas Instruments. TL852 Sonar Ranging Receiver. Retreived from `http://focus.ti.com/docs/prod/folders/print/tl852.html`, March 1988.

[19] Texas Instruments. Power Supply Topologies. Retreived from `http://focus.ti.com/lit/ml/sluw001a/sluw001a.pdf`, 2004.

[20] Texas Instruments. MSP430x1xx Family User's Guide. Retreived from http://focus.ti.com/mcu/docs/mcuprodtechdoc.tsp?sectionId=95&tabId=1201&familyId=342&techDoc=6&documentCategoryId=6, February 2006.

[21] Charles G. Triplett. Vehicular Mounted Piezoelectric Generator - US patent 4,504,761. Patent and Trademark Office, 1985.

Appendix A

Front Communications Module Code

```
#include "msp430x16x.h"
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <in430.h>

/*************************
 *  Placeholder Functions  *
 *************************/
void clearArray(char[]);
// The following are configuration functions
//used to set up the MSP
void init_sys(void);
void configUART(void);
void configExtPins(void);
void configDAC(void);
```

```
// The following function sends information over the UART
void toUART(char);

// Functions to start and stop the timers
void runTimerB(void);
void stopTimerB(int);
void runTimerA(void);
void stopTimerA(int);

// Various delays
void delay(void);
void wait1sec(void);
void wait(void);
void LED_wait(void);

// Sets the init and/or power pins high or low
void set_init(char);
void set_power(char);

// Functions to get information from the sensors
void getPin(void);
void waitForInput(void);
void startDCOClk(void);
void stopDCOClk(int);

// Functions to interact with the DAC (the LED driver circuit)
```

```
void toDAC(int);
void set_HEX(int);
void LED_enable(int);
void toDisp(int,int);

void configZigBee(void);
void zigEnterCommand(void);
void zigSetCoord(void);
void zigExitCommand(void);

// Global Variable Declarations
unsigned short int timer=0; //count interrupts from TimerB0
unsigned short int min; // elapsed minutes
unsigned short int sec; // elapsed seconds
unsigned short int dsec; // elapsed 1/10 seconds
unsigned short int csec; // elapsed 1/100 seconds
unsigned short int timer2=0; //count interrupts from TimerB0
unsigned short int min2; // elapsed minutes
unsigned short int sec2; // elapsed seconds
unsigned short int dsec2; // elapsed 1/10 seconds
unsigned short int csec2; // elapsed 1/100 seconds
unsigned short int tdir; // Direction of timer
                         // 1=up, 0=down
unsigned short int commandMode = 0;
unsigned int hitPin = 0x00;
int clockStart = 0;
int clockStop = 0;
```

```
static char dataRecv;
static char stringSend[3];
static int dummyTimes[8];
static char dummyFrontData[4];

static char commandSend[3];
static char commandRecv[3];

int testVar;

int a = 0;
char i = 0;
char j = 0;

// Sensor pin definitions
#define init  BIT1
#define power BIT2
#define input BIT3

/*************************************
 * The pins are defined as follows:  *
 *                                   *
 * ~420kHz Clock Signal:  Pin 16     *
 * Input:                 Pin 15     *
 * Power:                 Pin 14     *
 * Init:                  Pin 13     *
```

```
 ****************************************/

// DAC definitions for the 74LS138 demultiplexing circuit
#define hex0  BIT0
#define hex1  BIT1
#define hex2  BIT2
#define LEDen BIT3

#define CTS BIT7
#define RTS BIT4

#define CARRIAGE_RET 0x0D
#define SPACE 0x20
/****************************************
 * The pins are defined as follows:     *
 *                                      *
 * CTS:   Pin 19        (P1.7)          *
 * RTS:   Pin 16        (P1.4)          *
 * Din:   Pin 35                        *
 * Dout:  Pin 34                        *
 ****************************************/

/**************************
 *       Interrupts       *
 **************************/

// This is the interrupt that handles Timer A and its variables
```

```
#pragma vector = TIMERA0_VECTOR
__interrupt void Timer_A0(void)
{
  timer2++;
  if(timer2>9)
  {
    dsec2++;
    timer2=0;
  }
  if(dsec2>9)
  {
    sec2++;
    dsec2=0;
  }
}

// This is the interrupt that handles Timer B and its variables
#pragma vector = TIMERB0_VECTOR
__interrupt void Timer_B0(void)
{
  timer++;
  if(timer>9)
  {
    dsec++;
    timer=0;
  }
  if(dsec>9)
```

```
  {
    sec++;
    dsec=0;
  }

  if (clockStop == 1)
  {
    stopTimerB(1);
  }

}

// This interrupt handles received data in
// the UART1 receive buffer
// The data will be coming in from the ZigBee Module
#pragma vector=UART1RX_VECTOR
__interrupt void usart1_rx (void)
{
  sec2= 0;
  _BIC_SR_IRQ(LPM3_bits);                    // Exit LPM3

  if(commandMode){
    commandRecv[j++] = RXBUF1;
    if (j > sizeof commandRecv-1)
    {
      i = 0;
      j = 0;
```

```
    }
  }
  else{
    dataRecv = RXBUF1;
    if(dataRecv == 0x11)
    {
      //while (!(IFG2 & UTXIFG1));
      TXBUF1 = dummyFrontData[0];
    }
    if(dataRecv == 0x12)
    {
      //while (!(IFG2 & UTXIFG1));
      TXBUF1 = dummyFrontData[1];
    }
    if(dataRecv == 0x13)
    {
      //while (!(IFG2 & UTXIFG1));
      TXBUF1 = dummyFrontData[2];
    }
    if(dataRecv == 0x14)
    {
      //while (!(IFG2 & UTXIFG1));
      TXBUF1 = dummyFrontData[3];
    }
  }
}
```

```
void main(){
  init_sys();
  dummyFrontData[0] = 0x10;
  dummyFrontData[1] = 0x2A;
  dummyFrontData[2] = 0x30;
  dummyFrontData[3] = 0x4A;
  int i = 0;
  int j = 0;

  while(1){
    for(i=0; i<10; i++){
      wait1sec();
      dummyFrontData[0]++;
      dummyFrontData[2]++;
      dummyFrontData[1]--;
      dummyFrontData[3]--;
    }
    for(i=0; i<10; i++){
      wait1sec();
      dummyFrontData[0]--;
      dummyFrontData[2]--;
      dummyFrontData[1]++;
      dummyFrontData[3]++;
    }
  }

  runTimerB();
```

```
  if(sec2== 3)
    _BIS_SR(LPM3_bits + GIE);// Enter LPM3 with interrupts enabled
}

// Runs all necessary configuration functions
void init_sys(void)
{
  WDTCTL = WDTPW + WDTHOLD;              // Stop watchdog timer
  _BIS_SR(GIE);                          // Global Interrupt enable
  configUART();                          // Setup the UART Interface
  //configExtPins();
  configDAC();
  /*
  int del = 0;
  while(del < 4){
    wait1sec();
    del++;
  }
  */
}

// Sets up the UART1 to be ready to send and receive data in the
// proper format and at the correct data rate.
void configUART(void)
{
  P1DIR |= RTS;                          // P1.4 Direction is output
  P1DIR &= ~CTS;                         // P1.7 Direction is input
```

```
  P1SEL &= ~(RTS|CTS);   // P1.3-1 I/O Function
  P3SEL |= 0xC0;                    // P3.6,7 = USART1 TXD/RXD
  ME2 |= UTXE1 + URXE1; // Enable USART1 TXD/RXD
  UCTL1 |= CHAR;                    // 8-bit character
  UTCTL1 |= SSEL0;                  // UCLK = ACLK
  UBR11 = 0x00;                     // 32k/2400 - 13.65
  UBR01 = 0x03;                     //
  UMCTL1 = 0x4A;                    // Modulation
  UCTL1 &= ~SWRST;                  // Initialize USART state machine
  IE2 |= URXIE1;                    // Enable USART1 RX/TX interrupt
}

void toUART(char data)
{
  TXBUF1 = data;
  wait();
}
/*
// Setup the ports for transducers
//(connectivity with the sensor circuits)
void configExtPins(void)
{
  P1DIR |= (BIT4|BIT2|BIT1);      // P1.4,2,1 Direction is output
  P1SEL &= ~(BIT3|BIT2|BIT1);      // P1.3-1 I/O Function
  P1SEL |= BIT4;                // P1.4 Peripheral Function (SMCLK)
  P1DIR &= ~BIT3;               // P1.3 is input
  P1OUT = 0;
```

```
  P3DIR |= (BIT3|BIT2|BIT1|BIT0);      // P3.3-0 Direction is output
  P3SEL &= ~(BIT3|BIT2|BIT1|BIT0);     // P3.3-0 I/O Function
  P3OUT = 0;
}
*/

// This code is used to initialize the DAC
//to use the internal 1.5V reference
// with a gain of 1 (no gain)
void configDAC(void)
{
 ADC12CTL0 = REF2_5V + REFON;              // Internal 1.5V ref on
 DAC12_0CTL = DAC12IR + DAC12AMP_5 + DAC12ENC;//Internal ref gain 1
}

/****************** runTimerB **************/
void runTimerB(void)
{
  TBCTL=TBSSEL_1+CNTL_0+MC_1+ID_0;//ACLK,16 Bit,up mode,div=1
  TBCCR0 = 0x127; // 327 ACLK tics = ~1/100 seconds
  TBCCTL0 = CCIE; // TBCCR0 interrupt enabled
}

/****************** stopTimerB **************/
void stopTimerB(int reset)
{
```

```
  TBCTL = MC_0; // stop timer
  TBCCTL0 &= ~CCIE; // TBCCR0 interrupt disabled
  if(reset)
  {
    timer=0;
    min=0;
    sec=0;
    dsec=0;
    csec=0;
  }
}

/****************** runTimerA ***************/
void runTimerA(void)
{
  TACTL=TASSEL_2+CNTL_0+MC_1+ID_0; //ACLK, 16 Bit, up mode, div=1
  TACCR0 = 0x147 // 327 ACLK tics = ~1/100 seconds
  TACCTL0 = CCIE; // TBCCR0 interrupt enabled
}

/****************** stopTimerA ***************/
void stopTimerA(int reset)
{
  TACTL = MC_0; // stop timer
  TACCTL0 &= ~CCIE; // TACCR0 interrupt disabled
  if(reset)
  {
```

```
      timer2=0;
      min2=0;
      sec2=0;
      dsec2=0;
      csec2=0;
    }
}

// Simple delay function using a while loop
/*
void delay()
{
   int i;
   for(i=0; i<100; i++);
}
*/

// Delay using Timer B
void wait(void)
{
   timer = 0;
   runTimerB();
   while(timer2 != 1);
   stopTimerB(1);
}

// Delay of about 1 second using Timer B
```

```
void wait1sec(void)
{
  sec = 0;
  runTimerB();
  while(sec != 1);
  stopTimerB(1);
}

// Delay using Timer A for the LED circuit
void LED_wait(void)
{
  timer = 0;
  runTimerA();
  while(timer2 != 1);
  stopTimerA(1);
}



/***************** set_init() ***********************
 * This function sets the init pin depending on hilo *
 ****************************************************/
void set_init(char hilo)
{
  if (hilo == 'h')
    P1OUT |= init;
  else if (hilo == 'l')
    P1OUT &= !init;
```

```
}


/****************** set_power() ***********************
 * This function sets the power pin depending on hilo *
 ******************************************************/
void set_power(char hilo)
{
  if (hilo == 'h')
    P1OUT |= power;
  else if (hilo == 'l')
    P1OUT &= !power;
}

// This code determines which of the pins were pressed and
//activates/resets the timer accordingly
void  getPin()
{
   //char   inReg;
   hitPin = P2IN & 0x0F;
   /* Remember: Buttons are implemented "active low". That is, the
   input pin will equal logic 0 when the button is pressed */

   /* Read the input register and shift bits 7-4 to be bits 3-0
   then mask out low nibble */
   //inReg = P2IN & 0x0F;
   if (hitPin == 0x01)
```

```
    {
      clockStop = 0;
      clockStart = 1;
    }
    else clockStart = 0;
    if (hitPin == 0x02)
      clockStop = 1;
}

/****************** wait() ************************
 *  This function pauses until an input is recieved *
 ***************************************************/
void waitForInput()
{
  int inputRecieved = 0;
  while (!inputRecieved)
    if ((P1IN & input) || timer == 100)
      inputRecieved = 1;
}

void startDCOClk(void)
{
  /*asm{
    mov   BCSCTL2, 0x00;
    mov   BCSCTL1, 0x42;
    mov   DCOCTL, 0x00;
  }*/
```

```
  DCOCTL = 0;
  BCSCTL1 = RSEL1;
  BCSCTL2 = DCOR + DIVS0;

  TBCTL=TBSSEL_1+CNTL_0+MC_1+ID_0;// SMCLK, 16 Bit, up mode, div=1
  TBCCR0 = 0x247; // 327 ACLK tics = ~1/100 seconds
  TBCCTL0 = CCIE; // TBCCR0 interrupt enabled
}

/****************** stopDCOClk **************/
void stopDCOClk(int reset)
{
  TBCTL = MC_0; // stop timer
  TBCCTL0 &= ~CCIE; // TBCCR0 interrupt disabled
  if(reset)
  {
    timer=0;
    min=0;
    sec=0;
    dsec=0;
    csec=0;
  }
}

// This code takes in the modified value of
//the distance and puts it into the DAC.
```

```
//The DAC signal will then be used to
//drive the LM3914 signal.

void toDAC(int dist)
{
  /*
  int DACdist;

  if (dist >= 15)
  {
    DACdist = 0;
  }
  else
  {
    DACdist = 4096 - (4096/15) * dist;
  }
  */
  DAC12_0DAT = dist;
}

void set_HEX(int decimal)
{
  if (decimal == 1)
  {
    P4OUT = 0x00;
  }
  if (decimal == 2)
```

```
{
  P4OUT = 0x01;
}
if (decimal == 3)
{
  P4OUT = 0x02;
}
if (decimal == 4)
{
  P4OUT = 0x03;
}
if (decimal == 5)
{
  P4OUT = 0x04;
}
if (decimal == 6)
{
  P4OUT = 0x05;
}
if (decimal == 7)
{
  P4OUT = 0x06;
}
if (decimal == 8)
{
  P4OUT = 0x07;
}
```

```
}

void LED_enable(int onOff){
  if (onOff == 1)
    P4OUT |= LEDen;
  else if (onOff == 0)
    P4OUT &= !LEDen;
}

void toDisp(int LEDbank, int data)
{
  int LEDdelayVar = 3;

  LED_enable(0);       // Disable the 3-to-8 decoder
  toDAC(data);         //Send data to the signal
                       //input of bar graph driver
  set_HEX(LEDbank);    // Select the first bank of LEDs
  LED_enable(1);       // Enable the 3-to-8 decoder

  for(int i=0; i < LEDdelayVar; i++)
    LED_wait();          // Wait 1/100 seconds
}

void commandDelay(void)
{
  wait1sec();
```

```
    wait1sec();
    wait1sec();
}

void zigEnterCommand(void)
{
    commandMode = 1;
    while(!(commandRecv[0] == 'O' && commandRecv[1] == 'K'))
    {
        commandDelay();
        toUART('+');
        toUART('+');
        toUART('+');
    }
    clearArray(commandRecv);
}

void zigExitCommand(void)
{
    while(!(commandRecv[0] == 'O' && commandRecv[1] == 'K')){
        commandDelay();
        toUART('A');
        toUART('T');
        toUART('C');
        toUART('N');
        toUART(CARRIAGE_RET);
    }
```

```
    clearArray(commandRecv);
}

void zigSetCoord(void)
{
    while(!(commandRecv[0] == 'O' && commandRecv[1] == 'K'))
    {
        commandDelay();
        toUART('A');
        toUART('T');
        toUART('C');
        toUART('E');
        toUART(SPACE);
        toUART('1');
        toUART(CARRIAGE_RET);
        commandDelay();
    }
    clearArray(commandRecv);
}

void configZigBee(void)
{
    commandMode = 1;
    zigEnterCommand();
    zigSetCoord();
    zigExitCommand();
```

```
  commandMode = 0;
}

void clearArray(char array[])
{
  for(int i=0; i <= sizeof array; i++)
    array[i] = 0;
}
```

Appendix B

Display Module Code

```
#include "msp430x16x.h"
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <in430.h>
#include "C:\ccecc85\MSPfunctions.h"
```

```
/*************************
 *         Interrupts        *
 *************************/

// This is the interrupt that handles Timer A and its variables
```

```
#pragma vector = TIMERA0_VECTOR
__interrupt void Timer_A0(void)
{
 timer2++;        //Every time the interrupt occurs, increment
                  //timer variable, this happens every 1/100 seconds
  if(timer2>9)    //If timer2 has incremented 10 times(1/10 seconds)
  {
    dsec2++;      //Increment dsec2 (1/10 second variable)
    timer2=0;     //Reset timer2 variable
  }
  if(dsec2>9)     //If dsec2 increments 10 times (1 second)
  {
    sec2++;       //Increment sec2 (1 second variable)
    dsec2=0;      //Reset dsec variable
  }

  if (clockStop == 1)
  {
    stopTimerB(1);
  }

}


// This is the interrupt that handles Timer B and its
//variables and works the way as the TimerA interrupt
#pragma vector = TIMERB0_VECTOR
```

```
__interrupt void Timer_B0(void)
{
  timer++;
  if(timer>9)
  {
    dsec++;
    timer=0;
  }
  if(dsec>9)
  {
    sec++;
    dsec=0;
  }

  if (clockStop == 1)
  {
    stopTimerB(1);
  }

}

// This interrupt handles received data in the
    //UART1 receive buffer
// The data will be coming in from the ZigBee Module
#pragma vector=UART1RX_VECTOR
__interrupt void usart1_rx (void)
{
```

```
  if(commandMode){                    //If we are in command mode
    commandRecv[j++] = RXBUF1;        //Use the command array to
                                       //store data
    if (j > sizeof commandRecv-1) //If the command array is full
    {
      i = 0;                          //Reset counter variables
      j = 0;
    }
  }
  else{                               //If not in command mode
    dataRecv = RXBUF1;                //Store single character

    convertArray(dataRecv);//Put the received data
                                  //into the LED array
    dataOut = 0;          //Tell system that there is no data out
    receiveSuccess = 1; //Tell system there was a successful
                                                  //data reception
  }
}


void main(){

  init_sys();             // Begin MSP initialization
  configZigBee();         // Setup the ZigBee to be a coordinator
  char holdData;          // Variable to hold received wireless data
```

```
int frontBack = 0;      //Variable to determine which bumper
                        //is currently queried

while(1){

  if(receiveSuccess == 0){     // If there was no data received
    if(frontBack == 0){  //If we are working w/ the front bumper
      dataOut = 1;             //Tell system we have sent a request
      queryFront(frontSensor);//Send the request to front sensor
    }
    if(frontBack){          //Are we working with the back bumper?
        dataOut = 1;           //Tell system we have sent a request
        queryBack(backSensor);//Send the request to the back bumper
    }
  }

  if(receiveSuccess == 1){//If there is data received
    if(frontBack==0){ //If we are working w/ the front bumper
      frontSensor++;     //Increment the sensor that will
                         //be checked next time
      if(frontSensor==4)//If we have reached a sensor that does
                         // not exist
        frontSensor = 0;// Reset the variable
      frontBack = 1;     // Check the back next time
    }

    else if(frontBack){  //If we are working with the back bumper
```

```
        backSensor++;       // Increment the sensor that will
                            //be checked next time
        if(backSensor == 8)//If we have reached a sensor that does
                            //not exist
          backSensor = 4;   // Reset the variable
        frontBack = 0;      // Check the front next time
      }
      receiveSuccess = 0;   //Reset to let system know we are ready
                            //to start questioning the sensor
    }

    holdData = dataRecv;    // Once we have data received, save it

    toDisp(1,LEDarray[0]);          // Send data to LED bank 0
    toDisp(2,LEDarray[1]);          // Send data to LED bank 1
    toDisp(3,LEDarray[2]);          // Send data to LED bank 2
    toDisp(4,LEDarray[3]);          // Send data to LED bank 3
    toDisp(5,LEDarray[4]);          // Send data to LED bank 4
    toDisp(6,LEDarray[5]);          // Send data to LED bank 5
    toDisp(7,LEDarray[6]);          // Send data to LED bank 6
    toDisp(8,LEDarray[7]);          // Send data to LED bank 7
  }
}
```