VEHICLE MONITORING SYSTEM

By

Christopher Blount

Michael Jermann

Final Report for ECE 445, Senior Design, Spring 2012

TA: Justine Fortier

2 May 2012

Project No. 35

Abstract

This paper describes the design and implementation of a next-generation car alarm system. The product provides a vehicle owner with increased situational awareness by transmitting images of the vehicle to the user's smartphone through use of a GSM data connection. The user can then control the operation of the alarm by replying with SMS messages.

Several modular subcomponents, such as a camera, GSM/GPRS module, and control circuitry are interfaced to a microcontroller. The system is powered from 12 VDC, provided by the car battery.

This product is compatible with existing car alarm systems and has the potential to decrease noise pollution while improving effectiveness. Our design was fully functional and exceeded the majority of our performance requirements.

Contents

1. Introduction

Our project, “Vehicle Monitoring System,” is an augmentation to traditional car alarm systems that increases the user’s situational awareness of vehicle security threats via the real-time transmission of on-site images to the user’s smartphone. We believe that many consumers would prefer to visually judge their vehicle’s safety during times of crisis rather than rely upon the car alarm siren, which is frequently ignored by passerby. This product is designed to meet the needs of these consumers and improve upon current vehicle alarm systems by taking advantage of high-speed cellular networks and the proliferation of smartphones.

1.1 Purpose

Our primary objective was to improve user situational awareness during vehicle security threats. A cabin-mounted camera captures an image when a threat is detected (i.e. when the existing car alarm is activated). This image is sent to the user’s smartphone via MMS messaging. The user is able to visually judge the situation and control the operation of the car alarm via an SMS response. Using GSM connectivity as the wireless standard allows this system to achieve fast, reliable communication at any distance from the vehicle. This product has the potential to reduce noise pollution and improve the effectiveness of car alarms. Additionally, it is designed to be compatible with existing car alarm systems so that adding this product to existing vehicles will be a low-cost endeavor.

1.2 Specifications

The basic layout of the modular components of this system is displayed in Figure B.1.

In order to provide sufficiently high-resolution images of the vehicle, it is necessary that this system is capable of transmitting images of at least 2 kB to the user’s smartphone. This quantity was adjusted from our original requirement of 200 kB due to the addition of JPEG compression. Additionally, experimentation led us to realize that 2 kB images were of sufficient resolution for a smartphone screen. Images are transmitted to the user’s smartphone through a MMS message, which is formatted and sent by the GPRS module. An example image of this size is displayed in Figure E.1. We found the quality and resolution of this image to be acceptable for the purpose of determining if there is a security threat in the near proximity.

It is also necessary that the user be able to receive an alert within 5 minutes of a security incursion. This latency results from factors such as the image size, software run-time, and cellular signal strength.

In order to facilitate easy installation into existing car alarm systems, we require that this system have no more than 10 inputs and outputs. Additionally, we desire that this system be low-cost; therefore, we set $300 as the upper cost threshold.

1.3 Subprojects

The modular components of this design will be described in the sections following. It should be noted that the names of two of the subprojects have been changed to more appropriate descriptions since our design review (“Car Alarm” has been changed to “Car Alarm Control Circuitry” and “Wireless

Transceiver" has been changed to "GPRS Module"). However, the content and function of these components has remained the same.

1.3.1 Microcontroller

The microcontroller is interfaced to the camera, GPRS module, and car alarm control circuitry. It checks whether or not the car alarm has been activated, commands that a picture be taken, sends the picture to the GPRS shield, and commands the GPRS shield to send an MMS. It also parses the SMS messages received by the GPRS shield and, based on the content, inputs a signal to the car alarm control circuitry. This signal controls whether or not the car alarm should be played.

1.3.2 Car Alarm Control Circuitry

The car alarm control circuitry interfaces the microcontroller to the existing car alarm system. It outputs a signal to the microcontroller that indicates whether or not the car alarm has been activated. Additionally, it controls whether or not the car alarm siren is played, based on input from the microcontroller.

1.3.3 Camera

A cabin-mounted CMOS camera is used for image capture. It provides 128x160 resolution and implements JPEG compression, allowing us to acquire RGB images. Its operation is controlled by the microcontroller. Additionally, it transfers the image data to the microcontroller after image acquisition and compression is complete.

1.3.4 GPRS Module

This component provides our system with the ability to send MMS images to the user and respond to SMS commands. This module allows us to connect to a GSM network and access GPRS capabilities. Connecting to a GSM network provides us with nearly universal cellular coverage at with high data-transfer speeds. A prepaid SIM card is required for this module as well.

1.3.5 Power Supply

The power supply converts the 12 V DC provided from the car battery into the voltages needed to power the camera and GPRS module. The camera requires a peak of 60 mA at 3.3 V whereas the GPRS module requires a peak of 2 A at 5 V. The power supply has one output at 3.3 V and one output at 5 V, both of which are capable of supplying the necessary amount of current. Due to potential fluctuations of input power, the power supply is capable of outputting these same voltage levels over an input range of 9.6 to 13.2 V.

2 Design

2.1 Design Procedure

2.1.1 Microcontroller

In our system the microcontroller acts as the central link between the camera, GPRS module, and car alarm control circuitry. Therefore, it was vital to choose a microcontroller that could be easily interfaced to all three of these components. Our first design decision was to use an Arduino rather than a PIC due to its dearth of compatible shields and premade libraries. Our other requirement was that it had at least two serial ports along with at least 2 kB of writeable non-volatile memory. The two serial ports were necessary for the microcontroller to simultaneously interface with both the camera and GPRS module. The 2 kB image size requirement was determined experimentally by viewing images of different resolutions and data sizes on a smartphone. Both of these requirements led us to choose the Arduino Mega 2560 over other variants. The Arduino Mega is the only Arduino microcontroller with more than one UART port (the Mega has 4) and more than 1 kB of writeable EEPROM (the Mega has 4 kB).

After choosing the Arduino Mega 2560 microcontroller, we walked through an Arduino tutorial [1] in order to familiarize ourselves with the development environment and coding syntax. This basic tutorial primarily dealt with digital input and output, and allowed us to accomplish tasks such as blink LEDs and read input voltages. The basic digital input/output knowledge learned here guided the design of the car alarm control circuitry.

Our next task was to explore the premade EEPROM library. Fortunately, there is excellent documentation regarding this library [2]. We used EEPROM library functions in order to write and read data from several memory locations. Familiarizing ourselves with this library facilitated us determining how to save JPEG data to the EEPROM memory rather than saving it to external memory (and incurring the cost of a component such as an SD card) or pushing it directly to the GPRS (and risking overflow or data corruption).

2.1.2 Car Alarm Control Circuitry

The purpose of the car alarm control circuitry is to interface the system with existing car alarms. Therefore, the first step in designing this component was to research existing car alarms to determine how this component could interface with them. Online research [3] indicated that the majority of car alarms output a 12 V DC signal to the siren when a security incursion is detected. Therefore, we determined that we needed to design a component to allow the microcontroller to detect when this signal is output (and a security incursion is present) along with control the propagation of this 12 V DC signal to the siren.

In order to detect the presence of the 12 V DC signal (and accompanying security incursion) the most effective and simplistic option was implementing a voltage divider in order to output a low-voltage signal to the microcontroller. All other options were prohibitive due to their increased price and complexity. However, we did consider two implementation options for the relay that controls the propagation of the 12 V DC signal to the siren. The first option we considered was to drive a relay directly from a digital output of the Arduino. This option requires current close to the upper limit of what the Arduino can provide (40 mA) but is easily implementable. The second option was to connect the output of the Arduino to the base of a BJT transistor in order to indirectly drive the relay. This allows the relay to be driven by low amounts of current but increases the complexity of the relay circuit. We chose to implement the first option with a component we had found (the KHAU-17D11-12 by Tyco Electronics) that was a 12 V DC, 3 A relay. This relay could be driven by 33 mA at 5 V DC [4] , which was within the output range of the Arduino's digital pins.

While this component worked correctly initially, we found that its operation was intermittent when the entire system was hooked up. This was due to the current requirement being close to the upper limit of what the Arduino could provide. Therefore, we purchased new separate relay module called the SainSmart 5 V Relay. This relay is specifically designed to be driven by the output of an Arduino. Additionally, this component's relay contacts are rated to 10 A at 30 VDC, which is more than sufficient for our 12 V DC signal. Replacing the relay with this module increased the reliability of our car alarm control circuitry.

2.1.3 Camera

The camera we chose for this project was the C328R. This camera has UART serial capabilities that make it very simple to communicate with the Arduino. It also has onboard JPEG compression which allowed us to send higher quality pictures in a smaller package size. We did need to step down the signal to the Rx pin of the camera to 3.3V from the Arduino's logic level of 5V. Functionality-wise the manual [5] outlined all of the commands we would need to implement to get a working camera. We also found an Arduino library [6] for the C328R which helped out a lot because of our limited microcontroller experience.

The first thing we did to this code was removed it from the library shell adding all of the functions we needed to one code block. We wanted to understand what was going on behind the scenes of the library to cater it to our own needs. Next we removed the soft serial aspect of the library changing it to communicate with the Rx1 and Tx1 serial pins of the Arduino Mega (pins Rx0 and Tx0 are reserved for the USB communication with the computer).

With the camera code all visible we ran through the part of the code to get the camera synced. This took a few tries the camera was a little troublesome at first but once we got it synced the first time it started to sync consistently after that.

Next we let the code run through its entirety. All of the initializations were set and the get picture command was sent. This resulted in the data being sent back to the serial monitor on the computer as symbols. This was a good sign but we needed to verify this seemly garbage to be an actual photo. We found someone who had created some processing code for the Arduino to do take the packages sent and turn them into a JPEG photo [7]. After some minor tweaking to the settings we were using this sent us back pictures from the camera. We tried different resolutions and found the largest we could store on the Arduino's EEPROM.

We changed the original code to write to the EEPROM and the computer's serial port and then told it to write it to the serial monitor from the EEPROM. This resulted in two copies of the picture data being displayed and so we knew that the process had worked correctly. Everything on the camera end was taken care of and we just simply needed to load the picture from the EEPROM to the GPRS.

2.1.4 GPRS Module

The GPRS module needed to contain a chip that would allow us to access GSM/GPRS networks and open data connections. There are two primary chips that provide these GSM capabilities; the SIM900 series by SIMCOM and the SM5100B by Spreadtrum. While both of these chips enable data access through GSM networks (when connected to an unlocked, data-enabled SIM card), research revealed that a SIM900 firmware update supports a set of MMS-specific AT-commands [8]. While the SM5100B would allow MMS by opening up a TCP connection, the specialized MMS commands simplified the software, causing us to choose the SIM900 chip.

The second step was to choose a module that uses this SIM900 chip. We researched various break-out boards and realized that the most effective solution would be to use a GPRS shield that was designed for use with an Arduino microcontroller. We chose to use the GPRS Shield produced by Seeed Studio due to its comprehensive wiki containing implementation guidelines [9]. We did not "stack" the shield on top of the Arduino (as shields are typically used) because that arrangement connects the UART of the GPRS module to digital pins on the Arduino. That necessitates using a simulated serial port (a "SoftwareSerial" object) which increases the error rate. Therefore, we chose to place the GPRS shield adjacently and connect its serial port to "Serial2" of the Arduino.

In order to use the GPRS shield, we needed to obtain a SIM card with a GSM-based carrier and a data connection. The three major U.S. carriers with SIM cards on GSM networks are AT&T, T-Mobile, and Verizon. We first attempted to use a SIM card on the T-Mobile network, but experienced errors with the MMSC address ("http://t-mobile.com/mms/wapenc") not ending in ".com" and thus not being recognized by the MMS AT commands. We then purchased an AT&T SIM card, whose MMSC address ended in ".com" ("http://mmsc.cingular.com "), and were able to successfully connect to the MMS server.

An issue we ran into when first attempting to use the MMS AT commands was that the GPRS shield was not recognizing them. For instance, when we would send the command to initiate MMS connection ("AT+CMMSINIT"), we would receive the response "ERROR" rather than "OK". Due to poor documentation we had issues resolving this problem until we located a firmware update that allowed us to successfully perform these commands [10].

2.1.5 Power Supply

The power supply went through multiple revisions. First we had design a supply using the TI TPS54383 dual output buck converter. This was designed to power both the C328R camera and the SIM900 chip. The design had one output at 4.1V and 2A and a second output at 3.3V and 60mA for the C328R camera. This initial design schematic can be seen in Figure C.3. We followed the design examples from the datasheet [11] for our design. We ordered the parts for this design and proceeded with the rest of our project in the meantime. We knew the Arduino had a 3.3V output on it with close to the correct amount of current for the camera. We tried the camera powered off of the Arduino and it worked just fine if we unplugged the USB communication for the Arduino it worked as well. This made our output for the 3.3V of the buck converter not needed. We got in the components for the power supply and had issues getting it to function properly. We then found that our breakout board preferred a 5V input. We needed to start over with our power supply.

We now went with the TI LM2576, a 5V 3A voltage regulator. This is a very simple buck converter design that takes in a voltage range from 7-40V and outputs 5V with a max current draw of 2A. This was a perfect solution for our project at the time and is Figure C.4. We needed something simple at this point since we were worried about time as well as it still would have relatively good efficiency still being a buck converter. We followed the datasheet [12] for the design and picked out the parts according to their specifications. At this point we would have been hard pressed to churn out a PCB for our project but we had more power supply issues and this forced us to stick with a perforated board.

Our final power supply issue came back to the 3.3 voltage level. Our camera, though found earlier to be functional off the Arduino's 3.3V output, when we did further reliability testing we decided we did need an external 3.3V supply for the camera. When the USB communication was removed from the camera it did work off the Arduino but only about four out of every five times. This was not acceptable for us so we took the 5V we already had and stepped it down with the TI LM317T linear regulator to 3.3V. This design took a single equation from the datasheet [3] to calculate a resistor value to control the output shown in Figure C.5. We were under a tight time constraint and this solution, though not as elegant and efficient as we would have liked, could be bought at a local electronics store and was simple to implement. This was the last piece of the project we completed and was easily added to the perforated board a few days before the demo.

2.2 Design Details

2.2.1 Microcontroller

The Arduino Mega 2560, as seen in Figure C.1, is powered by the 12 V DC supply. It is connected to the car alarm control circuitry through two of its digital pins. Digital pin 10 is configured to be an input and is connected to the voltage divider of the car alarm circuitry. The status of this pin is continually checked by the microcontroller. When a high voltage is seen at this input, the image acquisition process begins.

The other digital pin that is used is digital pin 11. This pin is configured to be an output and is set to "HIGH" if an SMS is received by the GPRS shield following the sending of the MMS. This output is used to drive the relay contained in the car alarm control circuitry.

Additionally, the microcontroller is connected to the camera and GPRS module through serial connections (Serial1 and Serial2, respectively). This allows it to communicate with both of them at the appropriate times (the camera immediately after digital pin 10 is read to be "HIGH," and the GPRS shield after the JPEG data has been saved to EEPROM).

2.2.2 Car Alarm Control Circuitry

The car alarm control circuitry consists of a voltage divider circuit and a 12 VDC relay, as seen in Figure C.1. Given that the digital inputs of the Arduino will be read as "HIGH" as long as the input voltage is between 3 and 5.5 V [14], our goal was to design this voltage divider (as shown in Figure 2) such that 4 V is seen at this input when the 12 VDC car alarm signal is activated.

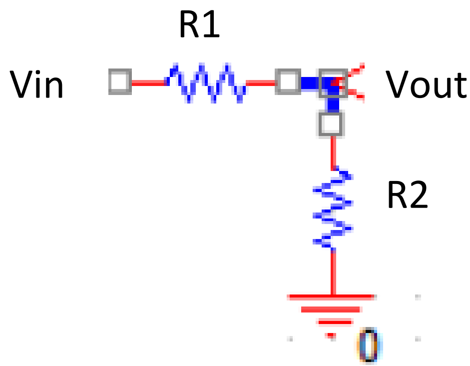


Figure 1: Voltage Divider Circuit Diagram

The calculation of the ratio of R1 to R2 was accomplished is

$$Vout = Vin\left(\frac{R2}{R1+R2}\right) \tag{1}$$

$$Vout * R1 + Vout * R2 = Vin * R2 \tag{2}$$

$$\frac{R2}{R1} = \frac{Vout}{Vin-Vout} = \frac{4}{12-4} = .5 \tag{3}$$

where Vout = 4 V and Vin = 12 V. Given that any digital pin configured as an input as a high impedance (approximately 1 MΩ) the magnitude of the resistors is insignificant; therefore, we chose R1 = 10 kΩ and R2 = 5 kΩ, which satisfies Equation 3.

The relay (the SainSmart 5 V Relay Module) was simply connected such that the output of the Arduino (Digital Pin 11) was connected to its input while the 12 VDC signal and car alarm siren input were connected across its contacts (i.e. across the ports that are shorted when the input is high) [15] as seen in Figure C.1.

2.2.3 Camera

A diagram of the camera can be found in Figure C.2 and the connections for this are shown in Figure C.1. Equation 2.2.3.1 shows the calculation for the voltage divider.

$$\frac{5V\ x\ 10k\Omega}{(10k\Omega + 5k\Omega)} = 3.33\ V \tag{4}$$

The flow chart for the camera's section of the Arduino code is Figure B.4. This shows the steps at which the camera goes through from the beginning of a detection to the picture stored on the EEPROM. We set the baud rate to 9600 bps which we found best for communication with the Arduino. The parameters for the picture were set to a 128 by 160 resolution with JPEG compression, resulting in a picture size less than the 4 kB of space on the EEPROM of the Arduino.

2.2.4 GPRS Module

Given that the GPRS module is connected to the microcontroller via the serial port and makes use of 5 V logic, the hardware connection is straightforward. The major design challenges of using the GPRS shield involved implementing the software for sending MMS messages and parsing received SMS messages.

The flowchart for the software for sending MMS messages is shown in Figure B.2. The Arduino sends a series of AT commands (as per the documentation in the MMS Command Manual [8]) to initiate an MMS connection, configure network parameters (based on which provider's GSM network is accessed), and open an MMS connection. This is a series of 10 distinct commands (each of which sets a different network parameter) which are repeated until an acknowledgement (in the form of an "OK") is received from the GPRS module. After the configuration is complete, the Arduino begins pushing the JPEG image data stored in its EEPROM to the GPRS shield. It precedes the data by sending the number of image packets to expect. Once all packets are processed, the Arduino begins sending the MMS – setting the receiving phone number, setting the title of the MMS (in our system it is "ALARM"), and finally sending the "AT+CMMSSEND" command which completes the process.

The flowchart for the software for parsing received SMS messages is shown in Figure B.3. After the MMS has been sent, the Arduino enters a while loop that consists of sending an AT command to the GPRS that checks how many new SMS messages have been received. If the response to this is non-zero, an AT-command is send to the GPRS shield that asks it to return the text of the most recent SMS message. If not, the Arduino delays for five seconds before repeating the while loop. If a text message has been received, it parsed the text to see if it contains the characters "ON". Based on the contents, the Arduino will drive the relay appropriately.

2.2.5 Power Supply

The calculations for the first design of the power supply for the TI TPS 54383 chip that was unused is located in Calculations F.1.

The TI LM2576 had a straight forward design that came directly from the datasheet [12] and this design is shown in Figure C.4. The chip was designed when hooked up to the given circuit to take in a wide range of voltages from 7-40V and output 5V with a max current of 3A. We needed 2A during peak transmission draw from the SIM900 and this supply could handle that with a comfortable amount of room. The datasheet recommended an electrolytic capacitor for Cin and Cout so we chose the Panasonic EEU-HD1E101B 100µF 25V aluminum electrolytic capacitor for Cin and the Panasonic EEU-FR1E102B 1000µF 25V aluminum electrolytic capacitor for Cout . They also suggested a power inductor for L that could handle the correct amount of current for the application. We chose the Murata 15104C 100µH 2.62A power inductor. Lastly we needed a diode for the design with a current of at least 3A and the Fairchild Semiconductor EGP30A rectifier was chosen.

The TI LM317T linear regulator design is shown in Figure C.5 and we used the datasheet [13] for this design. The design was not stringent on the components and we acquired all of them from the senior design lab. The one component we needed to pay particular attention to was R2. This component value needed to be calculated by using Equation 2.2.5.1 from the datasheet [13] R2 is set to 394Ω.

$$R2 = \left(\frac{V_{\ out}}{1.25} - 1\right) x\ R1 = \left(\frac{3.3}{1.25} - 1\right) x\ 240 = 394\Omega \tag{5}$$

3. Design Verification

3.1 Testing

Our design consists of five relatively modular components – the power supply, microprocessor, GPRS module, CMOS camera, and car-alarm relay. At the highest level, we verified the working status of all five components by setting the 12 V car alarm input, receiving an MMS containing a picture of the vehicle, and subsequently sending an SMS message to the system to initiate car alarm operation. By measuring the voltage of the relay output, along with the received MMS message, we confirmed that the system is accomplishing its basic function.

While building the system, we tested and integrated each modular component individually. The status of each of these tests can be found in Table A.1.

3.1.1 Microcontroller

The microcontroller tested through #2 and #3 in Table A.1. Fortunately, both of these requirements were satisfied as we achieved full system-functionality. Additional testing was conducted by forwarding all serial inputs/outputs seen by the Arduino to a connected PC in order to check that all communications occurred in the desired manner. The microcontroller satisfied all performance requirements.

3.1.2 Car Alarm Control Circuitry

The car alarm control circuitry was required to output between 11-13 V to the car alarm siren when its relay was activated (as per requirement #1 of Table A.1). Additionally, it was required to input a signal between 3-5 V to Arduino digital pin 10 when the car alarm signal was activated (as per requirement #4 of Table A.1). As viewed in this table, both requirements were met, with 11.69 V being seen at the car alarm siren output and 4.11 V being seen at the Arduino digital pin 10 under the appropriate conditions.

3.1.3 Camera

The part of the design that concerned the camera was under a single requirement from the Requirements and Verifications Table A.1. Requirement #2 outline the camera needed to take a picture and store it in the Arduino's memory. We had the camera with the ability to successfully sync, take the picture, and store it in the EEPROM, just as the Flow Chart B.4 shows.

3.1.4 GPRS Module

The functioning of the GPRS module was primarily verified through full-system testing; however, the timing requirements were especially interesting given that we exceeded expectations. The MMS needs to be received by the user within 5 minutes of the command being send to the GPRS module (as per requirement #3 of Table A.1) and the relay output needs to be switched within 5 minutes of the SMS text containing "ON" being sent to the GPRS module (as per requirement #1 of Table A.1). The GPRS module exceeded both requirements by sending the MMS within two minutes of security incursion and forwarding the SMS such that the relay is switched within 20 seconds of the message being sent.

3.1.5 Power Supply

The power supply was tested to the fifth and sixth requirements in the Requirements and Verifications Table A.1. We also tested the efficiency of the power supply as well.

The sixth requirement was met easily. The input range of the TM2576 was from 7-40V and this requirement specified a range of 9.6-13.2V. In Figures D.1 and D.2 the 5V output is shown at a 5.12V output for both the input of 9.56V and 13.25V. In Figures D.3 and D.4 the 3.3V output is shown at a 3.34V output for both the input of 9.56V and 13.25V.

The fifth requirement was to have a standby current draw of less than 100mA. We had a current draw of 169.8 mA as shown from Table D.1. This was close to our goal of 100 mA and was still reasonable for our application. We wanted to figure out where this extra current was coming from so we separated all of the components and found the standby current draw and power of each. The relay at 137.8 mA was found to be the source of extra current. To change our design in the future we would first use a relay with only one output since that is all we need. Secondly we would find a medium between the simple relay that was unreliable and this highly reliable but highly complicated and therefore power hungry relay we ended up with.

Lastly we found the efficiency of our power supply. From Table D.2 we had a input power of 192 mW and an output power of 76.96 mW. We took the ratio of these two numbers and calculated our efficiency to be 40%.

3.2 Testing Results

Our system achieved full-system functionality and worked according to all of our requirements and specifications, minus the static power consumption requirement. However, we still believe that the power consumption is low enough as to not unnecessarily drain the car battery and as such, are completely satisfied with the results of our verification testing. Each of our components functioned as desired when tested on both an individualized and full-system level.

4. Costs

4.1 Parts

As shown in Table 1 the total cost of non-power supply components was $274.49. Table 2 displays the cost of power supply parts, which amounted to $15.11. Therefore, the total cost of parts was $289.60, which fulfills our requirement that the system parts cost less than $300. However, this ended up being $52.22 more than the $237.38 we anticipated that we would spend in our design review [16]. The biggest discrepancy came from the SIM card being much more expensive than expected ($89.97 rather than $36.00).

Table 1 Parts Costs

Part	Manufacturer	Retail Cost ($)	Bulk Purchase Cost ($)	Actual Cost ($)
C328R	Electronics123.com	$59.95		$59.95
Arduino Mega 2560	Arduino	$41.50		$41.50
30 VDC Relay (5V)	SainSmart	$8.99		$8.99
GPRS Shield	Seeed Studio	$59.90		$59.90
SIM Card + Data Plan	Red Pocket Mobile	$89.97		$89.97
Enclosure	Parts Express	$6.28		$6.28
Perforated Board	Parts Express	$7.90		$7.90
Total		**$274.49**		**$274.49**

Table 2 Power Supply Parts Costs

Part	Manufacturer	Retail Cost ($)	Bulk Purchase Cost ($)	Actual Cost ($)
LM2576	Texas Instruments	$6.55		$6.55
EEU-HD1E101B 1000 µF Capacitor	Panasonic	$.91		$.91
EEU-FR1E102B 100 µF Capacitor	Panasonic	$.42		$.42
15104C 100 µH Inductor	Murata	$2.00		$2.00
EGP30A rectifier	Fairchild Semiconductor	$.64		$.64
LM317T	Texas Instruments	$2.99		$2.99
Capacitors	-	$1.00		$1.00
Resistors	-	$.60		$.60
Total		**$15.11**		**$15.11**

4.2 Labor

Ideal Salary = $40/hour

Actual Hours Spent = 20 hours/week * 13 weeks/Semester = 260 hours/Semester

Labor Cost = 260 Hours * $40/Hour * 2.5* 2 = $52,000

Labor costs were roughly equivalent to what we anticipated (given that we estimate that we spent on average 20 hours per week on this project).

4.3 Total Cost

Total Cost = Labor Cost + Parts Cost = $52000 + $289.60 = $52,289.60

This is roughly $52.22 over our anticipated total cost. Therefore, we only exceeded our budget by .1%.

5. Conclusion

We had a very successful project. We exceeded all of our design requirements except for one for which we found a quick fix for a future solution. It was very rewarding taking a project from the brainstorming stage all the way to a working proof of concept as shown in Pictures E.2 and E.3. We both learned a lot about microcontrollers, cellular networks, power supplies, and serial communication. This project has prepared us for the real engineering process, along with how to properly document and verify our design. We will surely use the skills learned in this class every day in our future careers.

5.1 Accomplishments

Our system was fully functional. We created a protection device that can be installed in most any car with a very simple procedure. We met our goals for reliability and for quickness of alert. A user would be notified within just 2 minutes of an incursion no matter where they are in the vast cellular network with a picture directly from their vehicle. We generally exceeded our performance requirements which was a huge success. We became fluent in with the Arduino, which in having zero prior experience with microcontrollers was also an accomplishment.

5.2 Uncertainties

Our only requirement not met was the 100mA standby current. This flaw as talked about in the testing section came from our desire for reliability over power consumption. In the end a working project was much more important to us and with the time constrains of some of the issues with the camera and relay arose we did not have the time to assess every viable option and instead went with choices that we knew would work. With more time we could design a more efficient power supply and find a relay that fits both our microcontrollers and our power consumption needs.

5.3 Ethical considerations

The first ethical consideration that we need to address is the chance of falsely incriminating a potential intruder. Our system does not deem whether or not an intrusion has occurred but merely uses the existing alarm that the manufacturer has deemed appropriate. It responds to this in real-time and therefore there is little chance of false incrimination.

The second ethical consideration we needed to address is the chance of privacy invasion. Since this is a security system it could in theory be used for purposes other than monitoring intruders. Our system requires professional installation because of the imbedded nature of the existing car alarm and therefore there is no chance of it being used to invade someone's privacy.

The third ethical consideration we needed to address is the chance of capturing a police officer on our system and whether or not that act is illegal. This act has been deemed by the courts to be protected by our first amendment right to freedom of the press. As long as we are not taking video or audio of an officer there is no legal ramifications of catching a photograph of them.

5.4 Future work

As far as our project goes in the future it is mainly improvements in the power supply. We can use a more efficient buck converter to increase our efficiency as well as use a buck converter for the 3.3V section. We need a more efficient relay to meet our last requirement not met too.

Improvements upon the design that would enhance the users experience would be to add a SD card slot onto the Arduino to have larger storage space that would also be permanent. The system would be able to take higher resolution pictures and store them in case the user’s phone data was deleted. This would also allow us to add a higher resolution camera as well so the pictures could be taken up to the maximum size of the MMS capabilities.

References

[1] "Arduino Tutorial." Web. 13 Mar. 2012. <http://www.ladyada.net/learn/arduino/>.

[2] "Arduino - Libraries." *Arduino*. Web. 02 May 2012. <http://arduino.cc/it/Reference/Libraries>.

[3] "Car Security and Convenience / Basic Alarm Output Connections." *The12volt.com*. Web. 02 May 2012. <http://www.the12volt.com/carsecurity/page3.asp>.

[4] KHA Series Datasheet, Tyco Electronics, Winston-Salem, NC

[5] C328R User Manual, Electronics123.com, 2007.

[6] Sean Voisen, Arduino C328R Library, http://letsmakerobots.com/node/13354, unpublished.

[7] Project2501, Processing Code, http://letsmakerobots.com/node/13354, unpublished.

[8] *SIM900 MMS AT Command Manual V1.01*. 25 Oct. 2011.

[9] "GPRS Shield." Seeed Studio. Web. 02 May 2012. <http://www.seeedstudio.com/wiki/GPRS_Shield>.

[10] "GPRS Shield ˆSIM900 Firmware Update." *Geek on Fire*. Web. 02 May 2012.

<http://www.geekonfire.com/wiki/index.php?title=GPRS_Shield（SIM900）_Firmware_Update>.

[11] TPS 54383 Datasheet, Texas Instruments, Dallas, TX, 2007.

[12] LM2576 Datasheet, Texas Instruments, Dallas, TX, 2004.

[13] LM317T Datasheet, Texas Instruments, Dallas, TX, 2011.

[14] 8-bit Atmel Microcontroller with 64K/128K/256K Bytes In-System Programmable Flash Datasheet, Atmel, San Jose, CA, 2011

[15] SRD-05VDC-SL-C Datasheet, Songle Relay, Zhejiang ,China

[16] Jermann, Michael, and Christopher Blount. *Design Review*. 23 Feb. 2012.

Appendix A Requirement and Verification Table

This section contains our verification requirements, along with the status of each sub-requirement.

Table A.1 System Requirements and Verifications

Requirement	Verification	Status
1. Relay is controlled by SMS messages sent to GSM/GPRS module a. The relay is in the “OFF” state by default b. SMS messages are received and properly processed by microcontroller c. Relay is controlled by PWM output from microcontroller d. Output from relay is able to act as the input to the car-alarm siren	1. The 12V car alarm input will be switched on manually. An SMS message containing the text “ON” will then be sent to the phone number assigned to the system SIM card. If successful, the load connected to the output of the relay will be at 12 V shortly thereafter. Additionally, if the system passes this test the following verification steps (a-d) may be skipped. If not, they must be undertaken sequentially. a. Prior to sending the SMS message the voltage across the relay output load will be measured. If this voltage is zero, the relay is in the “OFF” state and the system is operating correctly. b. If within 5 minutes of the SMS message being sent (based on sample texts sent and received) there is current at the output of the relay, the microcontroller has responded to the SMS input. We will also send an SMS message with an input other than “ON.” If there is current at the output of the relay within 5 minutes of this message being sent, this functionality is not working correctly. c. We will measure the current at the PWM output. If at least 13.3 mA of current is output following the SMS message, this is working correctly. If not, this output is not providing sufficient current to switch “ON” the relay.	1. Verified (as per full-system functionality test) a. Relay output measured at 0.01 V before SMS message sent b. Relay output was activated within 20 seconds of SMS message being sent c. Arduino digital pin current output sufficient to activate relay d. Relay measured at 11.69 V after relay activated

	d. Upon the relay being switched on, the voltage drop across the load 12 V ± 10%. If it is outside this range, the output signal is not capable of powering the alarm siren.	
2. Microcontroller is able to control and receive JPEG images from the CMOS camera a. Microcontroller is able to sync with camera b. Camera takes picture when commanded to by microcontroller c. Image data is transferred from camera to microcontroller flash memory	2. We will connect the microcontroller to a computer via a serial-to-USB connection and undertake the procedure to command the camera to take a picture. After approximately 1 minute (necessary for data transfer and JPEG compression) we will transfer the data from the flash memory to the computer. If this data contains a valid JPEG image, the test has succeeded. a. After sending the SYNC message for 62 seconds the microcontroller should have received an ACK message from the camera. If it has not, the test has failed. b. After sending the GETPICTURE command to the camera, the microcontroller should receive a message containing an image size less than 200 kB within 30 seconds as specified. If it does not receive this message, no picture was taken and the test has failed. Additionally, if the image size is more than 200 kB our microprocessor may not be able to process the image and the test has failed. c. We will transfer the image data from the flash memory of the microcontroller to the computer. The data must be in JPEG format or else the test has failed. The image must also be less than 200 kB or else the test has failed. Finally, the image must be in color or else the test has failed.	2. Verified as per full-system functionality test a. Sync was successful within 10 seconds b. Image of approximately 4 kB written to microprocessor EEPROM within 60 seconds c. Picture was sent in valid JPEG format and successfully reconstructed
3. System is able to send	3. We will connect the microcontroller	3. Verified as per full-system

MMS messages to the user within 5 minutes. a. Microcontroller is able to sync the GSP/GPRS module b. MMS is successfully received by user	to a computer via a serial-to-USB connection and undertake the procedure (as listed in the flow chart) to send an MMS message. We will manually load the microcontroller flash memory with a JPEG of approximately 200 kB. We will then undergo the procedure to send the MMS message to a user. If the user receives the MMS message containing the correct image within 5 minutes, the test was successful. If the MMS is not received, the transmission takes more than 5 minutes, or the image data is incorrect, then the test has failed. a. After sending the SYNC message the microcontroller should receive a "CONNECT" message within 60 seconds. If it does not, the test has failed. b. Within 5 minutes of sending the MMS ("AT+CMMSDOWN" command) the pre-programmed phone number should have received the MMS message. If the user has an active data connection and has not received the message in this time frame, try again at a different time (to account for carrier difficulties). If it still does not work, then test has failed.	functionality test a. Sync was successful within 10 seconds b. MMS was message received by user within 120 seconds
4. System is able to detect when the car alarm goes off	4. We will monitor the voltage of the microprocessor analog input connected to the car alarm signal through resistors. We will measure at this node using a voltmeter first with the car alarm signal off. This voltage should measure zero. We will then turn the 12V car alarm signal on. The voltage at the digital input should now be between 3-5 V. If not, the test has failed.	4. Verified as per full-system functionality test When car alarm signal was not activated, the input of the corresponding pin of the microcontroller was measured at 0 V. When activated, the voltage of the pin was measured to be 4.11 V.
5. Standby current draw of the system is less than 100 mA in order to not drain the car battery.	5. The car alarm input will be set to zero. We will then measure the current entering the circuit using a multimeter. If the current is more than 100 mA the	5. Not verified – See Table D.1

a. Test the power consumption of the microcontroller. b. Test the power consumption of the wireless card. c. Test the power consumption of the camera.	test has failed; if not, the test was successful. a. We will test the current in the microcontroller's power supply section of the circuit. If it is less than ~30 mA then the test passes. b. We will test the current in the wireless card's power supply section of the circuit. If it is less than ~30 mA then the test passes. c. We will test the current in the camera's power supply section of the circuit. If it is less than ~30 mA then the test passes.	
6. Power supply functions for input range of 13.2 to 9.6 V	6. The DC signal will be swept slowly from 13.2 to 9.6 V. If the power supply is still outputting the correct voltages on the outputs then the test is successful if not the test has failed and the components need to be checked for accurate placement and connection.	6. Verified – See Figures D.1 – D.4

Appendix B Block Diagrams and Flow Charts

The figures below contain diagrams that describe the high-level interfaces and software for the various subcomponents of the system.

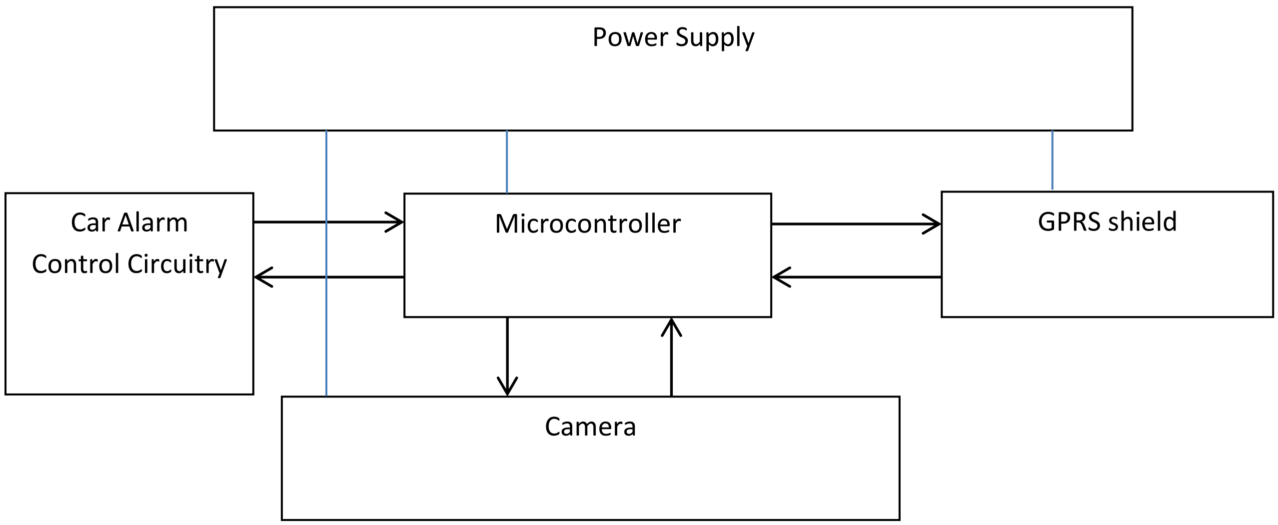


Figure B.1: High-level system block diagram

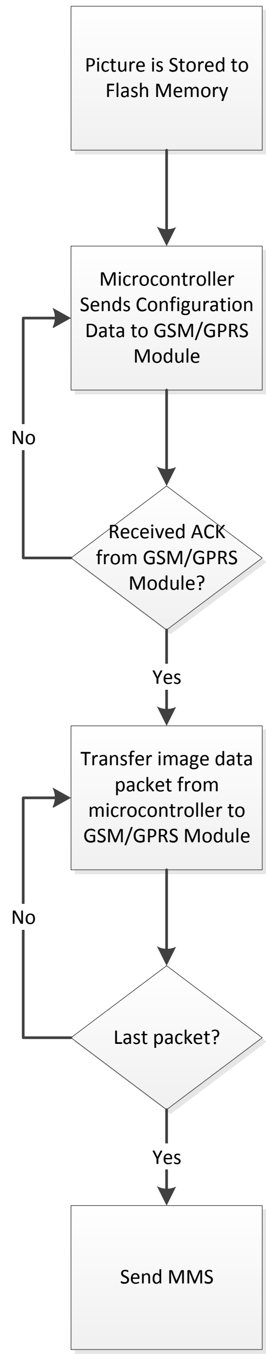


Figure B.2: MMS Software Flowchart

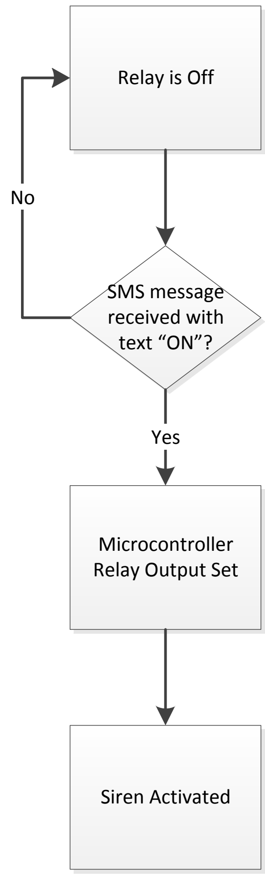


Figure B.3: SMS Software Flowchart

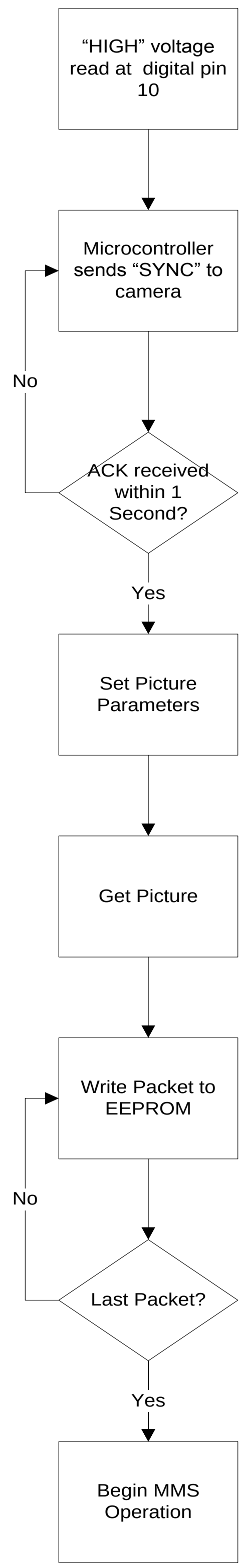


Figure B.4: Camera Software Flowchart

Appendix C Schematics and Pin-Outs

This section contains the schematics and pin-outs for various system components.

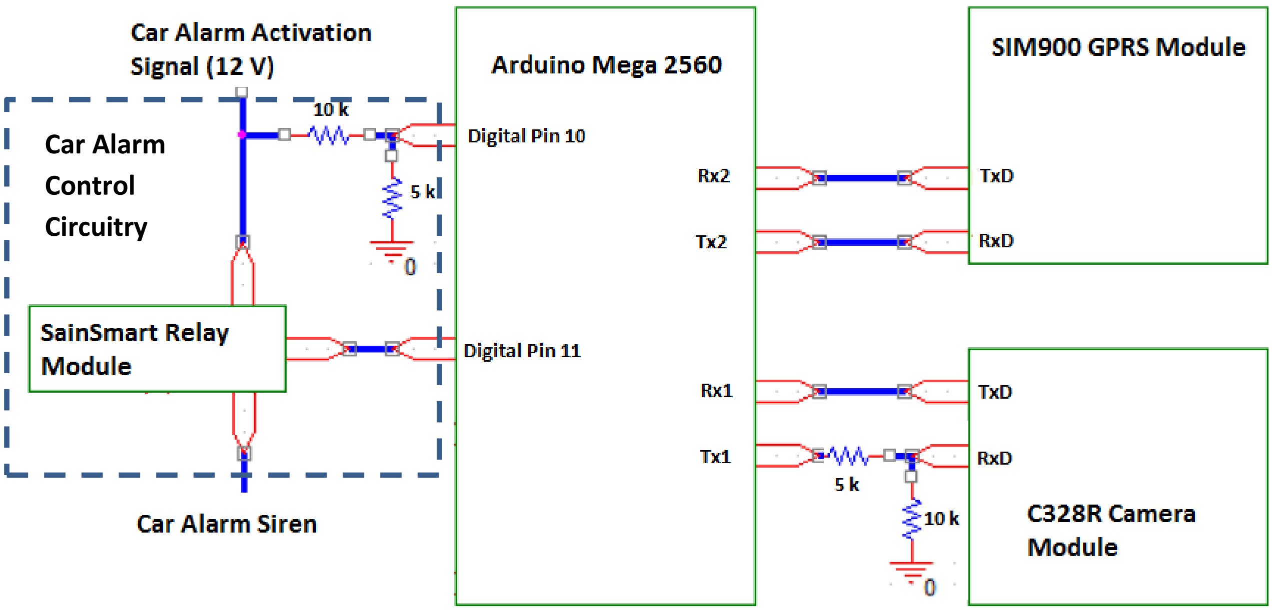


Figure C.1: System Schematic

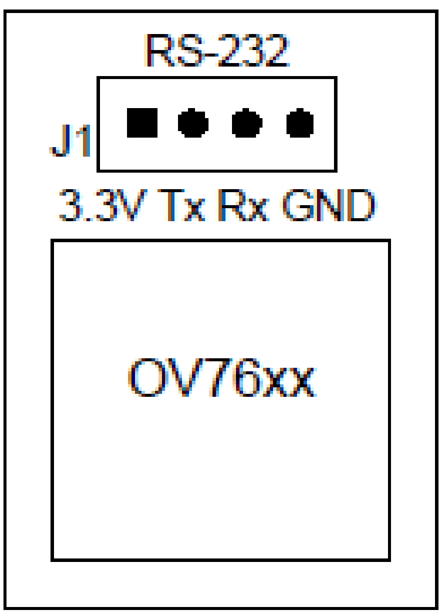


Figure C.2: Camera Schematic

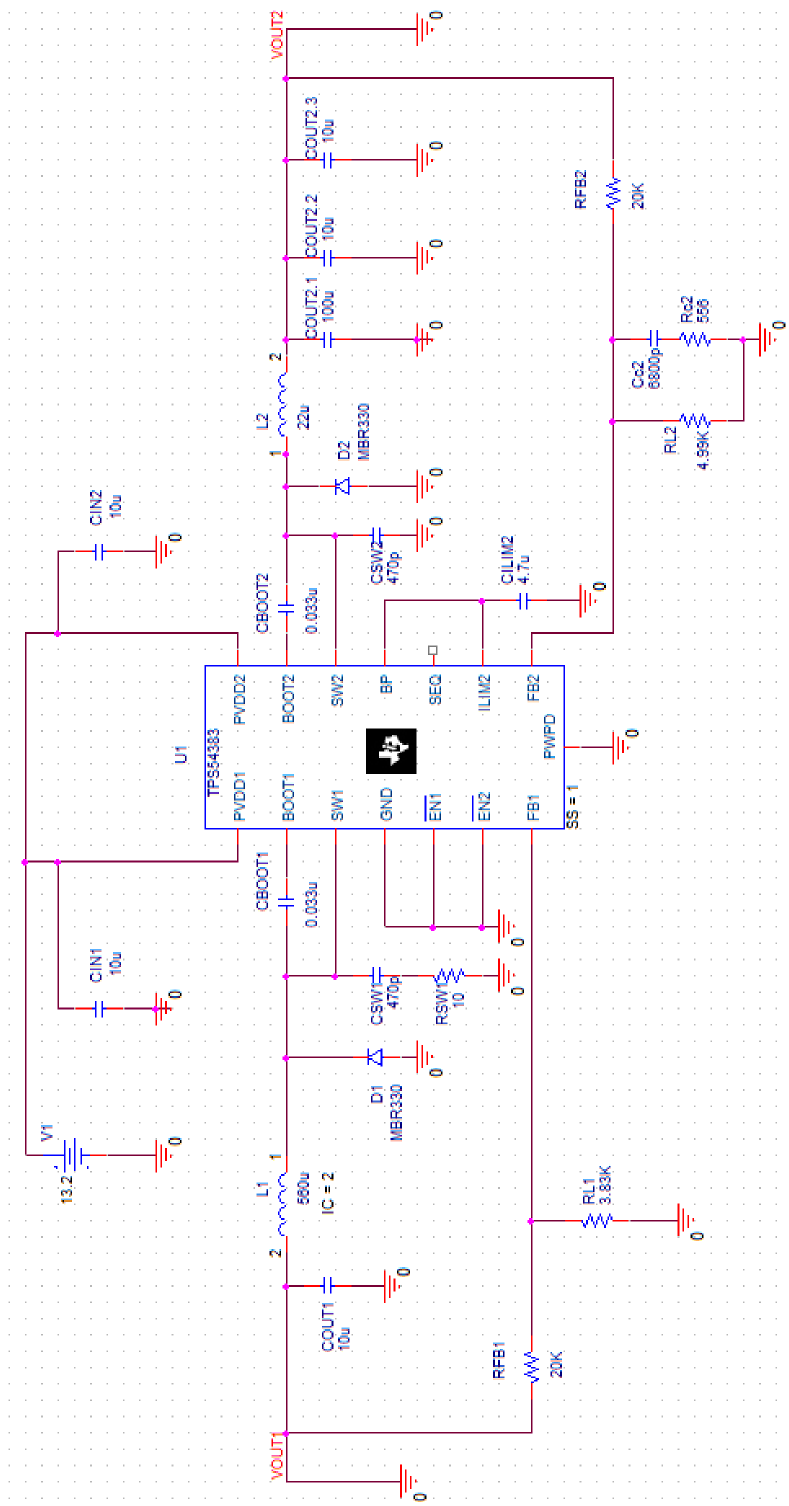


Figure C.3: Initial Power supply schematic

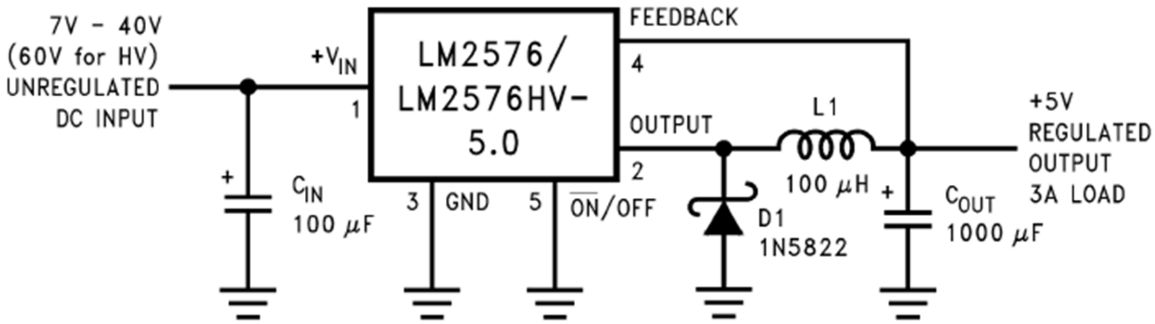


Figure C.4: TI LM2576 5V Buck Converter Schematic[2]

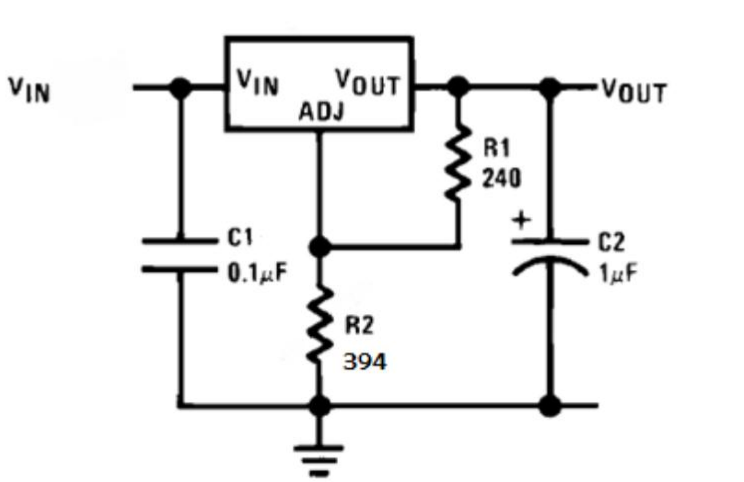


Figure C.5: TI LM317T 3.3V Linear Regulator Schematic [3]

Appendix D Test Data

This section contains graphs, screenshots, and tables resulting from verification testing.

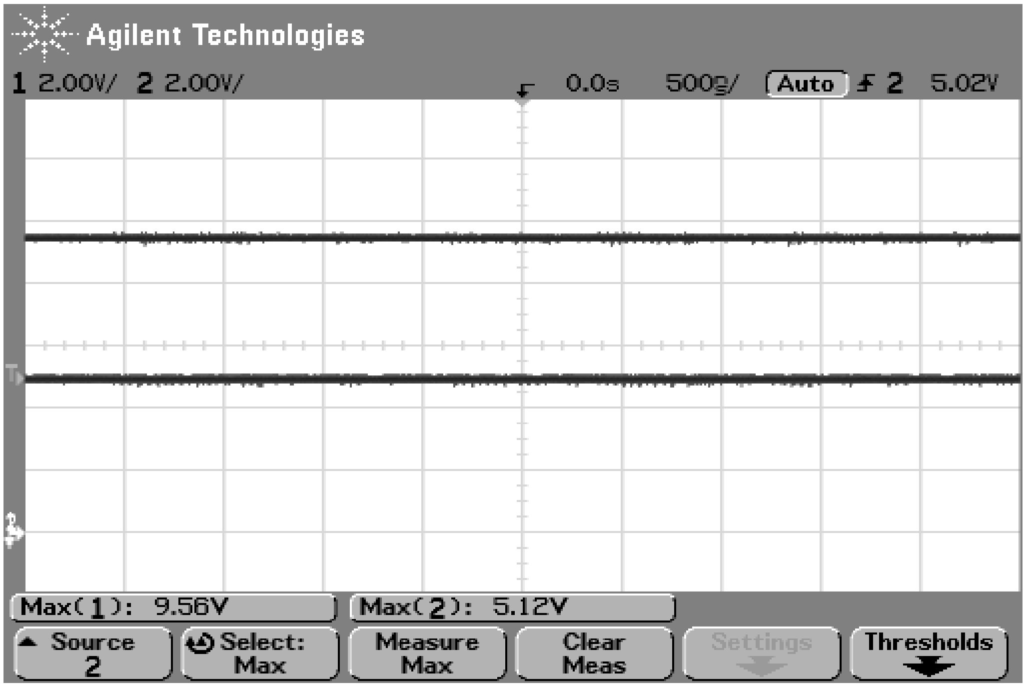


Figure D.1 Channel 1 9.6V input, Channel 2 5V expected output

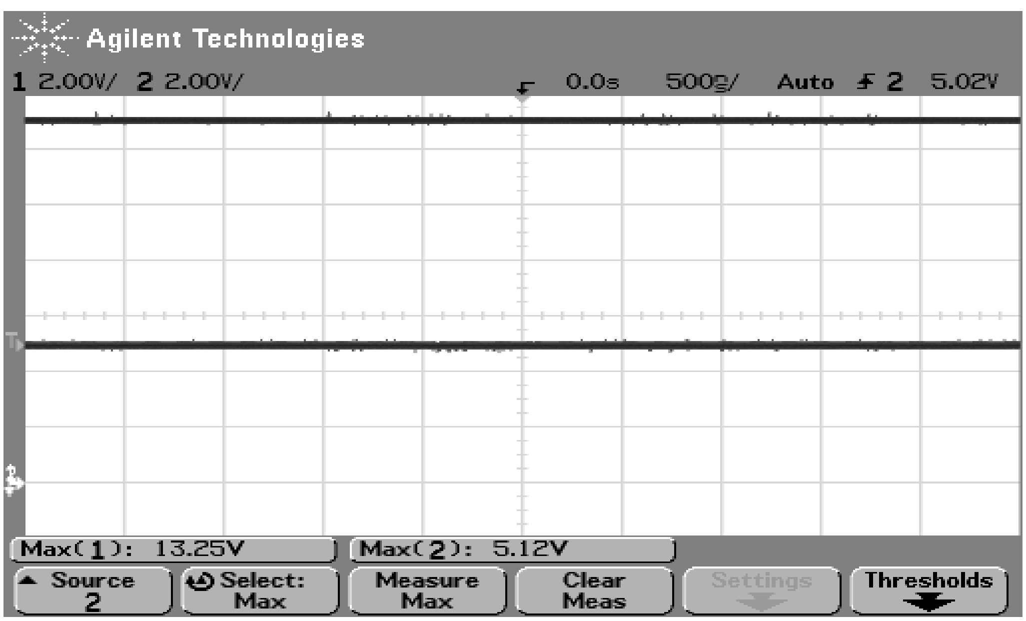


Figure D.2 Channel 1 13.2V input, Channel 2 5V expected output

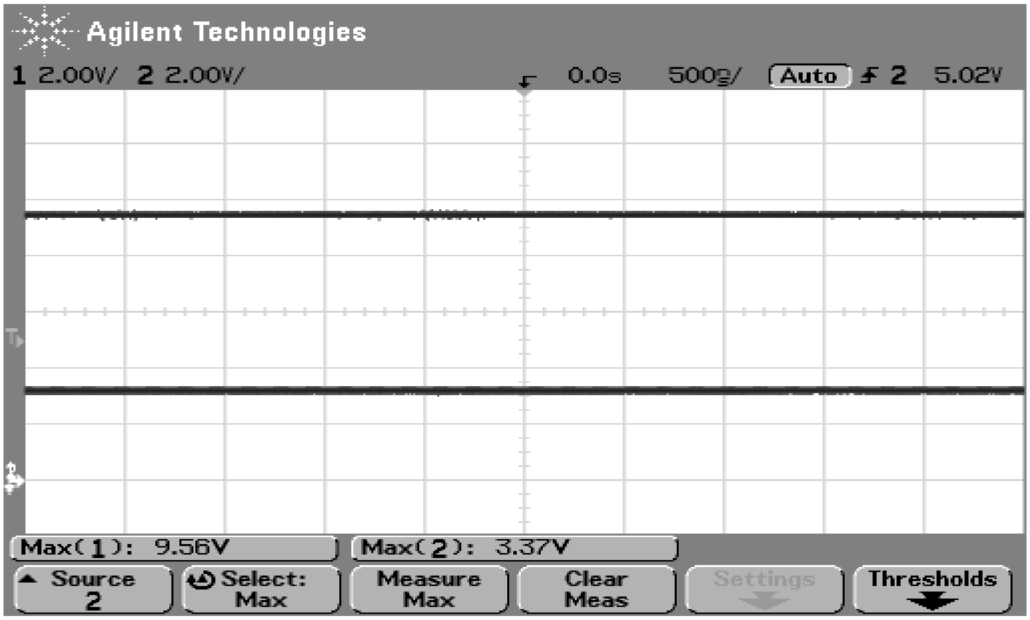


Figure D.3 Channel 1 9.6V input, Channel 2 3.3V expected output

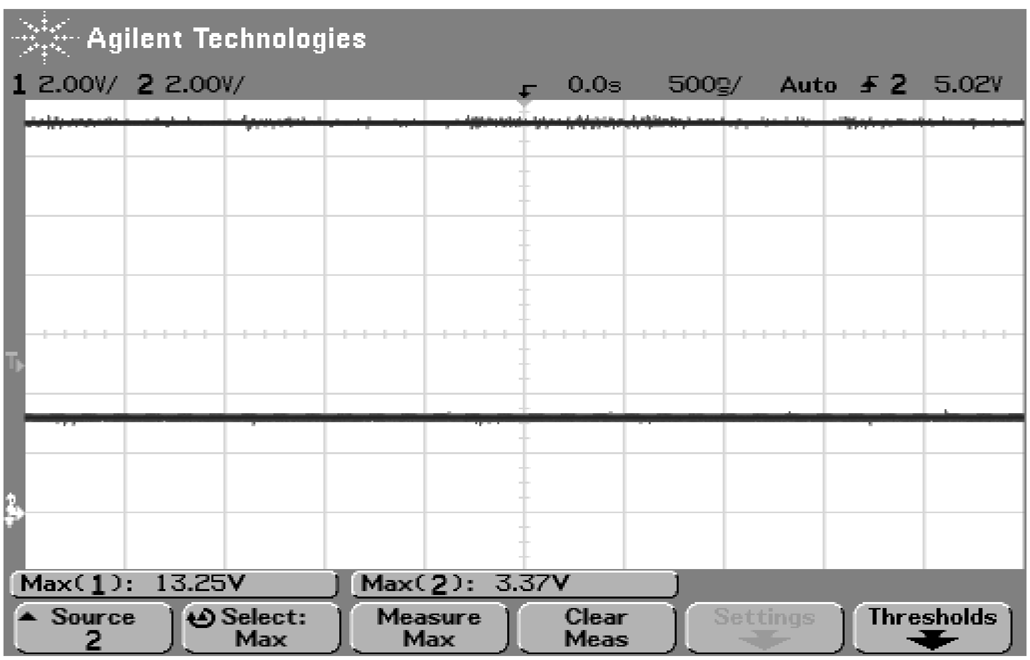


Figure D.4 Channel 1 13.2V input, Channel 2 3.3V expected output

Table D.1 Current and Power draw of components

Components	Current Draw	Power
Full System w/o GPRS on	169 mA	2.03 W
Full System w/ GPRS on	184 mA	2.21 W
Arduino	77.7 mA	932 mW
Camera	.67 mA	2.2 mW
GPRS	17.9 mA	89.5 mW
Relay	137.8 mA	689 mW

Table D.2 Efficiency of Power supply

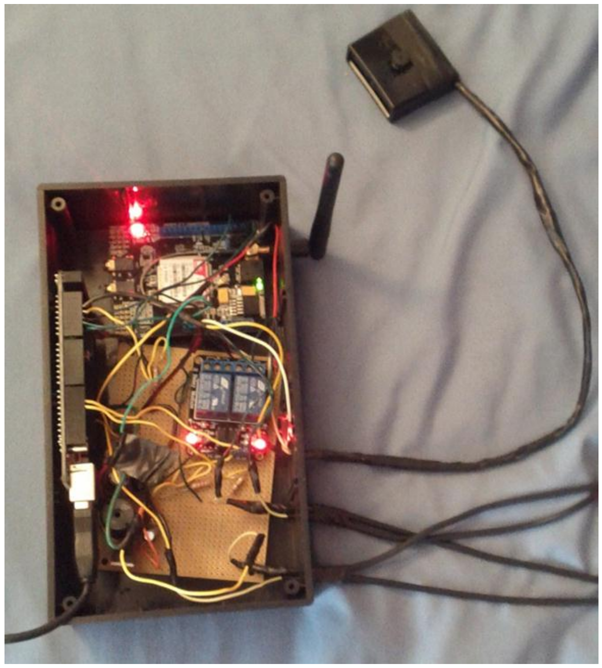
Input Power	192 mW
Output Power	76.96 mW
Efficiency	40%

Appendix E Pictures

This section contains sample pictures and pictures of the project.

Figure E.1: Actual Image Taken By System

Figure E.2: Picture of Final Product



Picture E.3: Picture of Internal Circuitry

Appendix F Reference Calculations

This section contains lengthy calculations that we performed when designing system components.

Calculation F.1 TI TPS54383 Dual Output Buck Converter

We followed the datasheet [11] specifications for the design and below are the calculations performed for the external circuitry.

(Subscript 1 is for the camera side and subscript 2 is for the wireless card side)

Predetermined constants for this chip:

- Switching frequency: $f_{SW} = 310kHz$
- Internal voltage reference: $V_{FB} = .8V$

Our Circuit parameters:

- $V_{IN} = 9.6 - 13.2V$
- $V_{OUT1} = 3.3V$
- $V_{OUT1(MIN)} = 3.2V$
- $V_{OUT1(MAX)} = 3.4V$
- $V_{OUT2} = 4.0V$
- $V_{OUT2(MIN)} = 3.1V$
- $V_{OUT2(MAX)} = 4.8V$
- $I_{OUT1} = .06\ A$
- $I_{OUT2} = 2.5\ A$

Duty Cycle:

To estimate the duty cycle the datasheet[1] gave us equations F.1.1 and F.1.2:

$$D_{MAX} \approx \frac{V_{OUT} + V_{FD}}{V_{IN(\mathrm{MIN})} + V_{FD}} \tag{F.1.1}$$

$$D_{MIN} \approx \frac{V_{OUT} + V_{FD}}{V_{IN(\mathrm{MAX})} + V_{FD}} \tag{F.1.2}$$

(The value V_{FD} is equal to the estimated forward drop of a Schottky rectifier diode of .5V)

$$D_{MAX1} \approx \frac{3.3 + .5}{9.6 + .5} = 37.6\% \qquad D_{MAX2} \approx \frac{4.0 + .5}{9.6 + .5} = 44.6\% \tag{F.1.1}$$

$$D_{MIN1} \approx \frac{3.3 + .5}{13.2 + .5} = 27.7\% \qquad D_{MIN2} \approx \frac{4.0 + .5}{13.2 + .5} = 32.8\% \tag{F.1.2}$$

Inductor:

To select an inductor first we need to calculate $I_{LRIP(MAX)}$ the peak to peak ripple current which is 30% of the maximum output current Equation F.1.3.

$$I_{IRIP1(MAX)} = .06\ x\ .3 = 18\ mA \qquad I_{IRIP2(MAX)} = 2.5\ x\ .3 = 750\ mA \tag{F.1.3}$$

These were put into the given Equation F.1.4 to find the minimum value of the inductor:

$$L_{MIN} = \frac{V_{IN(\mathrm{MAX})} - V_{OUT}}{I_{IRIP(MAX)}}\ x\ D_{MIN}\ x\ \frac{1}{f_{SW}} \tag{F.1.4}$$

$$L_{MIN1} = \frac{13.2 - 3.3}{.018}\ x\ .277\ x\ \frac{1}{310x10^3} = 491x10^{-6}H \tag{F.1.4}$$

$$L_{MIN2} = \frac{13.2 - 4.0}{.7}\ x\ .328\ x\ \frac{1}{310x10^3} = 13x10^{-6}H \tag{F.1.4}$$

The next higher standard inductor value of 560 µH is best for the first circuit. We chose the coilcraft MSS1278-564KLB as our inductor.

The next higher standard inductor value of 22 µH is best for the second circuit. We chose the coilcraft MSS1278-153ML as our inductor as recommended by the manufacturer.

Rectifier Diode:

First we need to calculate the minimum breakdown voltage of the Schottky diode in Equation F.1.5:

$$V_{BR(MIN)} \geq 1.2\ x\ V_{IN(MAX)} = 1.2\ x\ 13.2 = 15.84\ V \tag{F.1.5}$$

The diode chosen for the circuit is the ON SEMI MBRS330T3 because of its reverse breakdown voltage characteristics as recommended.

Now we must estimate the average current in the diode given by the Equation F.1.6:

$$I_{D(AVG)} \approx I_{OUT(MAX)}\ x\ (1 - D_{MIN}) \tag{F.1.6}$$

$$I_{D(AVG)1} \approx .06\ x\ (1 - .277) = .043\ A \tag{F.1.6}$$

$$I_{D(AVG)2} \approx 2.5\ x\ (1 - .328) = 1.68\ A \tag{F.1.6}$$

Now we estimate max power through the diode with Equation F.1.7:

$$P_{D(MAX)} = I_{D(AVG)}\ x\ V_{FM} \tag{F.1.7}$$

The forward voltage drop at the selected current is V_{FM1} = .03V and V_{FM2} = .04V.

$$P_{D(MAX)1} = .043\ x\ .03 = 1.29\ mW \tag{F.1.7}$$

$$P_{D(MAX)2} = 1.68\ x.04 = 670\ mW \tag{F.1.7}$$

Output Capacitor:

The converter's internal compensation creates a f_{res} at 3kHz. The Equation F.1.8 given for output capacitance is as follows:

$$C_{OUT} = \frac{1}{4\ x\ \pi^2 x\ (f_{res})^2\ x\ L} \tag{F.1.8}$$

$$C_{OUT1} = \frac{1}{4\ x\ \pi^2 x\ (3x10^3)^2\ x\ 560x10^{-6}} = 5.03\ \mu F \tag{F.1.8}$$

$$C_{OUT2} = \frac{1}{4\ x\ \pi^2 x\ (3x10^3)^2\ x\ 22x10^{-6}} = 128\ \mu F \tag{F.1.8}$$

To pick the correct capacitor we need calculate the maximum ESR they can have with Equation F.1.9:

$$ESR_{MAX} = \frac{V_{RIP}}{I_{RIP}} - \frac{D}{f_{SW}\ x\ C_{OUT}} \tag{F.1.9}$$

The manufacturer rounds the duty cycle to 50% and gives the ripple voltage as 50 mV[1].

$$ESR_{MAX1} = \frac{.05}{.018} - \frac{.5}{310x10^3\ x\ 5.03x10^{-6}} = 2.46\ \Omega \tag{F.1.9}$$

$$ESR_{MAX} = \frac{.05}{.75} - \frac{.5}{310x10^3\ x\ 128x10^{-6}} = 54\ m\Omega \tag{F.1.9}$$

For the first circuit we chose the next highest capacitance of 10 µF and a ceramic TDK C2012X5R0J106M is chosen. This easily fits the ESR requirements with only an impedance of 2.5 mΩ.

For the second circuit we chose an electrolytic 100 µF Panasonic EEE FC1A101P with 400 mΩ ESR and two ceramic 10 µF TDK C2012X5R0J106M capacitors with 2.5 mΩ ESR. These capacitors are put in parallel to provide a combined ESR of 28 mΩ at 300 kHz.

Voltage setting resistors:

The primary feedback resistor between the VOUT and FB pins is recommended to be set at 20 kΩ. To calculate the lower resistors this Equation F.1.10 is used:

$$R_L = \frac{V_{FB} x\ R_{FB}}{V_{OUT} +\ V_{FB}} \tag{F.1.10}$$

$$R_{L1} = \frac{.8\ x\ 20x10^3}{3.3\ -\ .8} = 6.4\ k\Omega \tag{F.1.10}$$

$$R_{L2} = \frac{.8\ x\ 20x10^3}{4.0\ -\ .8} = 5\ k\Omega \tag{F.1.10}$$

Standard 0603 1/16 watt resistors are chosen at the values of 6.34 and 4.99 kΩ respectively.

Compensation Capacitors:

We need to check the ESR zero of the main output capacitor and if it is less than 20 kHz then and R-C filter is needed in parallel with R_L.

$$f_{ESR(ZERO)} = \frac{1}{2\ x\ \pi\ x\ ESR\ x\ C} \tag{F.1.10}$$

$$f_{ESR(ZERO)1} = \frac{1}{2\ x\ \pi\ x\ .0025\ x\ 10x10^{-6}} = 6.3x10^6\ Hz \tag{F.1.10}$$

$$f_{ESR(ZERO)2} = \frac{1}{2\ x\ \pi\ x\ .4\ x\ 100x10^{-6}} = 3980\ Hz \tag{F.1.10}$$

For the second circuit we need compensation in the form of an R-C circuit shown in Equations F.1.11-13:

$$R_C = \frac{R_{L2}}{\left(\frac{f_{ESR(DESIRED)}}{f_{ESR(ZERO)2}} - 1\right)} = \frac{5x10^3}{\left(\frac{40x10^3}{3980} - 1\right)} = 556\ \Omega \tag{F.1.11}$$

A standard value of 590 Ω is selected.

$$R_{EQ} = R_C + (R_{FB}||R_{L2}) = 590 + 4000 = 4.59\ k\Omega \qquad \text{(F.1.12)}$$

$$C_C = \frac{1}{2\ x\ \pi\ x\ R_{EQ}\ x\ f_{ESR(ZERO)}} = \frac{1}{2\ x\ \pi\ x\ 4590\ x\ 3980} = 8.71\ \text{nF} \qquad \text{(F.1.13)}$$

The TDK C1005X7R1E682MT is chosen at 6800 pF for the closest equivalent as suggested by the datasheet[1].

Input Capacitor:

A minimum 10 μF ceramic input capacitor on each PVDD pin so these are added as C_{IN}.

Boot Strap Capacitor:

The manufacturer requires a 33 nF capacitor across the BOOT pin and the SW pin. Also required off of these pins is a 470 pF capacitor in series with a 10 Ω resistor tied to ground[1].