

January 2013

# Design of an Advanced Lighting Measurement System for Roadway Safety

Mathew Johnson

University of South Florida, mpjohns3@mail.usf.edu

Follow this and additional works at: <http://scholarcommons.usf.edu/etd>

 Part of the [Computer Engineering Commons](#)

## Scholar Commons Citation

Johnson, Mathew, "Design of an Advanced Lighting Measurement System for Roadway Safety" (2013). *Graduate Theses and Dissertations*.

<http://scholarcommons.usf.edu/etd/4906>

This Thesis is brought to you for free and open access by the Graduate School at Scholar Commons. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact [scholarcommons@usf.edu](mailto:scholarcommons@usf.edu).

# Design of an Advanced Lighting Measurement System for Roadway Safety

by

Mathew P. Johnson

A thesis submitted in partial fulfillment  
of the requirements for the degree of  
Master of Science in Computer Engineering  
Department of Computer Science and Engineering  
College of Engineering  
University of South Florida

Co-Major Professor: Srinivas Katkoori, Ph.D.  
Co-Major Professor: Aldo Fabregas, Ph.D.  
Pei-Sung Lin, Ph.D.  
Hao Zheng, Ph.D.

Date of Approval:  
November 7, 2013

Keywords: Arduino, Transportation, Illumination, Embedded System,  
Systems Engineering

Copyright © 2013, Mathew P. Johnson

## **DEDICATION**

This is dedicated to my family, who have always given me their full support. I would not be where I am today without their continuous love and encouragement through the difficult times.

## **ACKNOWLEDGMENTS**

This project has been supported and funded by the Center for Urban Transportation and Research (CUTR) at the University of South Florida. I am thankful to CUTR and particularly thankful to Dr. Aldo Fabregas and Dr. Pei-Sung Lin for guiding me on this project. Dr. Srinivas Katkoori has my sincere gratitude for his advice, guidance, and encouragement not only through my thesis development, but through my entire studies at USF. I would also like to thank Dr. Hao Zheng for serving on my thesis committee. Michael Bato has my extreme gratitude for his participation in the data validation and collection process and his tireless work on the tremendous effort of post-processing. Thank you all.

## TABLE OF CONTENTS

LIST OF TABLES.....	iii
LIST OF FIGURES.....	iv
ABSTRACT .....	vi
CHAPTER 1: INTRODUCTION .....	1
1.1 Current Approach .....	2
1.2 Proposed Solution – An Overview .....	3
1.2.1 Arduino Based System .....	3
1.2.2 System Specifications.....	5
1.2.3 Validation Plans .....	5
1.3 Summary .....	6
CHAPTER 2: RELATED WORK.....	7
2.1 Study 1 – Iowa State University and VTTI .....	7
2.1.1 Phase I.....	7
2.1.2 Phase II .....	8
2.2 Study 2 – University of South Florida .....	9
2.3 Study 3 – Helsinki University of Technology.....	10
2.4 Summary .....	11
CHAPTER 3: PROPOSED EMBEDDED SYSTEM.....	12
3.1 Conceptual Diagram .....	12
3.2 Arduino Based Design .....	13
3.2.1 Arduino Microcontroller Board .....	13
3.2.2 GPS Module .....	16
3.2.3 Event Logger .....	18
3.2.4 USB Host Module.....	20
3.2.5 MicroSD .....	21
3.2.6 Distance Measurement Instrument .....	23
3.2.7 Light Meters.....	25
3.3 Software .....	28
3.3.1 Program Flow Chart .....	28
3.3.2 Program Description .....	30
3.4 System Cost .....	30
3.5 Summary .....	31

CHAPTER 4: EXPERIMENTAL RESULTS .....	32
4.1 System Validation.....	32
4.1.1 Static Test .....	32
4.1.2 Dynamic Test.....	34
4.2 Location Data.....	35
4.2.1 GPS Location.....	35
4.2.2 DMI Location.....	36
4.3 Output File Cleaning .....	36
4.4 GIS Representation .....	39
4.4.1 Route Tracing .....	39
4.4.2 Map Overlay .....	41
4.5 Summary .....	42
CHAPTER 5: CONCLUSION AND FUTURE WORK .....	43
REFERENCES .....	45

## LIST OF TABLES

Table 1 UP501 pin description.....	17
Table 2 Event logger code descriptions .....	19
Table 3 MicroSD card pin mapping .....	22
Table 4 Manual and automatic fc range.....	28
Table 5 Cost breakdown by part.....	30

## LIST OF FIGURES

Figure 1 Poorly lit roadway .....	1
Figure 2 Properly lit roadway .....	2
Figure 3 Block level diagram of ALMS .....	4
Figure 4 Measurement system developed by CTRE and VTTI.....	9
Figure 5 CUTR illumination measurement system .....	9
Figure 6 Results of CCD camera illumination measurements .....	10
Figure 7 Conceptual diagram of ALMS.....	12
Figure 8 Arduino Uno development board.....	13
Figure 9 ATmega328 block diagram.....	14
Figure 10 Pin mapping between ATmega328 and Arduino Uno board .....	15
Figure 11 GPS shield for Arduino Uno .....	18
Figure 12 Examples of event codes .....	20
Figure 13 Pin mapping of USB Host Shield.....	20
Figure 14 RAC Plus I distance measurement instrument.....	23
Figure 15 Installation instructions for MDS .....	24
Figure 16 T-10A relative spectral response .....	25
Figure 17 Light meters mounted on top of vehicle .....	26
Figure 18 Analog voltage to lux conversion factors .....	27
Figure 19 ALMS program flow chart.....	29
Figure 20 Static results at 30 inches.....	33



Figure 21 Static results at 60 inches.....	33
Figure 22 Dynamic validation test .....	34
Figure 23 Location data for Fletcher Ave .....	36
Figure 24 Flowchart for data cleaning program .....	38
Figure 25 Data cleaning program .....	39
Figure 26 Lane routes for Fletcher Ave created in Google Earth.....	40
Figure 27 Light meter routes on Fletcher Ave .....	40
Figure 28 GIS representation of illumination data on Fletcher Ave .....	41
Figure 29 Heat map of illumination on Fletcher Ave .....	42

## **ABSTRACT**

Roadway illumination is a vital component of safety while driving during the night. There are regulations in place to ensure all publicly maintained roads are properly lit, but the validation process is too time consuming, costly, and dangerous for adequate data collection studies. The work in this thesis is aimed toward remedying this problem by creating an Advanced Lighting Measurement System (ALMS) capable of recording illumination readings while traveling at normal driving speeds. This solution is based on the Arduino Uno development board, a cost effective yet powerful embedded platform. This thesis involves collecting data along 100 centerline miles of Florida roadways and converting the resulting illumination readings into GIS format, allowing them to be included in the roadway database of the Florida Department of Transportation (FDOT). By including this data FDOT will be able to repair poorly lit corridors and will be aware of possible safety concerns. The illumination values recorded by the ALMS have been validated and verified as an accurate replacement for conventional lighting measurement system.

## CHAPTER 1: INTRODUCTION

In 2012 there were over 280,000 traffic accidents in the state of Florida, with over half occurring during night time [1]. The difference between day and night crashes is made even more drastic when the number of drivers on the roadways is taken into account. It is estimated that drivers are 2.5 times more likely to be involved in an accident during night hours than they are during the day [2]. The increase in accident rate cannot be blamed solely on roadway lighting, since other issues such as fatigue and alcohol usage increase at night. However, there is strong evidence that an increase in illumination can drastically reduce accidents.



Figure 1: Poorly lit roadway [3]

A poorly lit roadway does not necessarily imply lack of lighting. It is also hazardous to have illumination that is inconsistent throughout the road, thus leaving a high contrast between bright and dark areas. In Figure 1, not only is the lighting inconsistent, but the orientation of the luminaires creates horizontal light spillage, resulting in a glare for oncoming vehicles. These factors greatly increase the risk of traffic accidents. By updating the lighting on a street, as shown in Figure 2, accidents can be reduced by as much as 30% [2].

The ability to detect poorly lit roadways is crucial to proper infrastructure maintenance and adequate illumination. This process can be very difficult and time consuming and is therefore often neglected. With enhanced measurement techniques, illumination standards can be met and drivers will be safer at night.

### **1.1 Current Approach**

The conventional method for measuring roadway illumination involves spot checking along a grid set up in the selected road. Measurements are taken using a light meter placed six inches above the ground. The operator must set up the light



Figure 2: Properly lit roadway [3]

meter, stand back, and trigger a measurement. The light meter is then moved to the next point on the grid and the process is repeated.

This method is both time consuming and dangerous. It is a lengthy process for the operator to set up the collection device in the roadway, step back, and record a reading. This leaves the operator and the testing equipment vulnerable to the dangers of the road for prolonged periods of time. This danger can be avoided with lane closures and police presence, but doing so adds additional cost to the measurement process. Due to these factors, extended roadway illumination studies are rarely implemented, leaving questions on whether the roads meet the standards for sufficient lighting.

## **1.2 Proposed Solution – An Overview**

In order to properly collect large amounts of illumination data in a safe and time effective manner we must be able to take measurements from a moving vehicle. This would eliminate the dangers to the measurement personnel in the roadway without road closures and expensive police presence. It is necessary that the proposed system be accurate and automated, allowing simple data collection in a moving vehicle.

### **1.2.1 Arduino Based System**

The proposed system is based on an Arduino microcontroller, interfaced with six subsystems to provide the necessary functionality (Figure 3).

- *Arduino Uno Microcontroller:* The Uno is based on the ATmega328 microcontroller, has 14 digital input/output pins, 6 analog input pins, and an analog to digital converter. This system uses 2 Unos which interface with the other systems and process all data received.

- *Distance Measurement Instrument (DMI):* The DMI is connected to the vehicle's speed sensor and provides pulses at a specified interval, triggering light meter measurements.
- *USB Module:* The USB module interfaces with the Arduino using the serial peripheral interface (SPI), allowing the usage of a USB number pad.
- *Event Logger:* The event logger is a wireless number pad used to type filenames and log various events during data collection.
- *GPS Module:* This system uses a Fastrax UP-501 GPS receiver. This module is capable of acquiring location, date, time, and velocity data.
- *Light Meters:* This system makes use of two Konica Minolta T-10A light meters. These light meters are accurate to one one-thousandth of a foot candle and

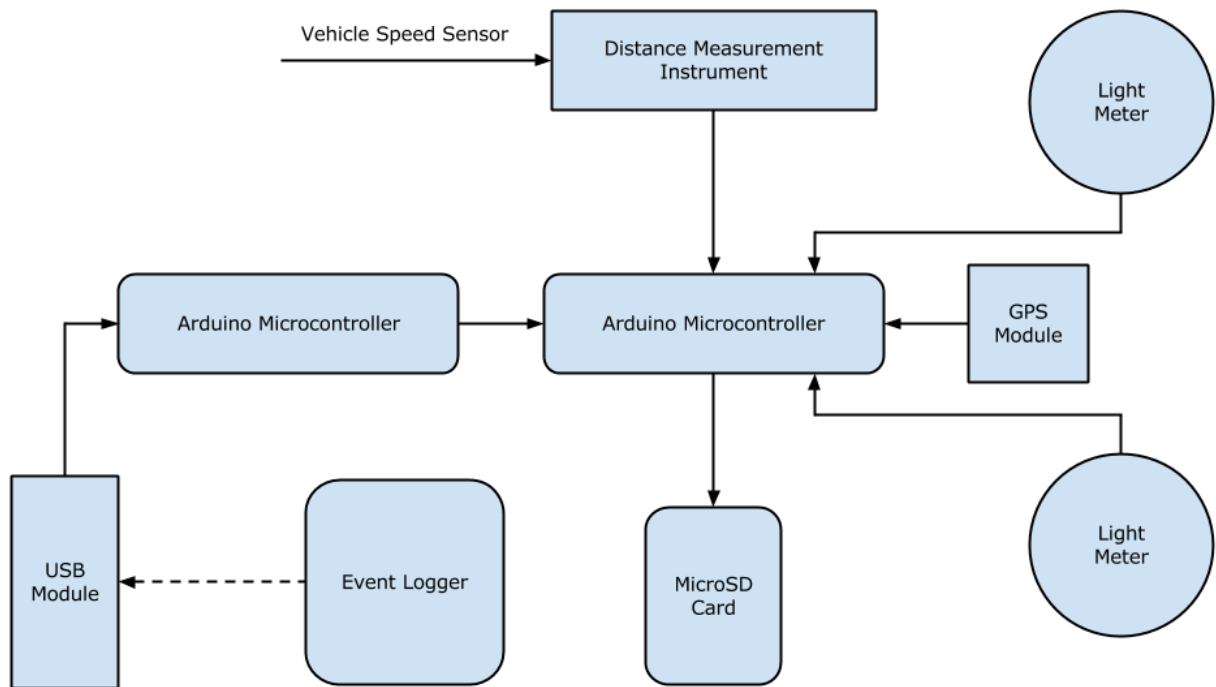


Figure 3: Block level diagram of ALMS

provide an analog output. The sensor of the T-10A can also be removed from the body of the unit.

- *MicroSD Card*: The MicroSD module communicates with the Arduino through serial protocol, allowing data to be saved.

### **1.2.2 System Specifications**

The following system specifications must be met:

1. Be capable of collecting data at speeds up to 30 miles per hour
2. Input filename using a number pad
3. Acquire date, time, speed, and exact location via GPS coordinates
4. Take light meter readings every 10 feet
5. Record important events (such as a cross-section, light pole, etc.)
6. Save all the data in a microSD card

### **1.2.3 Validation Plans**

To judge the success of this system the results must be within an acceptable range of error from results obtained through the conventional method of illumination measurement. There are multiple tests the system will undergo to validate the results.

1. Static Test – This test is used to validate the analog output and system algorithm used to determine the illumination level. The system will be tested at 30” and 60” using both the built in output of the light meter and the automatic logging provided through the proposed Arduino system. These two sets of results will be compared to determine any necessary modifications.
2. Dynamic Test – This test will validate the data collected while the vehicle is in motion. Stationary measurements will be taken at mounting height to provide

a baseline. The vehicle will then be driven at 30 mph along the same path, taking measurements in the same locations. These results will be compared to the stationary measurements to verify the measurements made while the vehicle is in motion.

### **1.3 Summary**

This chapter provides background information on the problem of efficient illumination measurement. The current methodology is outlined and reasons for developing an improved solution are provided. The proposed embedded system based on an Arduino microcontroller is introduced and the method for validating the system is described. Chapter 2 will survey related work in developing mobile illumination systems. Chapter 3 will describe the proposed system in detail, and Chapter 4 will present the experimental results. Finally, Chapter 5 will provide directions for future work and summarize the findings of this project.



## **CHAPTER 2: RELATED WORK**

The contemporary method for measuring roadway illumination, as described in Chapter 1, has been identified as a poor approach. Therefore, there have been multiple solutions proposed to overcome many shortcomings. In this chapter, we will briefly discuss three such systems. The first one was developed by Iowa State University in conjunction with Virginia Tech Transportation Institute. The second system was developed by University of South Florida. The third system was proposed by Helsinki University of Technology.

### **2.1 Study 1 – Iowa State University and VTTI**

In 2011, the Center for Transportation Research and Education (CTRE) at Iowa State University partnered with the Virginia Tech Transportation Institute (VTTI) to complete Phase II of their lighting research. In Phase I crash statistics were researched using binary representation for lighting at intersections (lighted or unlighted). In Phase II a system capable of measuring the exact illumination levels was developed [4].

#### **2.1.1 Phase I**

Phase I was carried out between 2003 and 2005 and involved recording information on 274 rural intersections throughout the state of Iowa. This data included lighting information such as location, type of luminaire, location of lighting fixture in relation to the intersection, and type of light pole. Intersections were then

eliminated from the study if they were deemed unusual. An intersection could be deemed unusual due to presence of an external light source (e.g., gas station) or if the intersection did not meet requirements for a rural location. This reduced the number of intersections to 223.

The collected information was combined with a binary representation of lighting, either lighted or unlighted. Traffic and crash data were collected at these intersections. The results showed that the ratio of crashes from night to day was lower at lighted intersections than at unlighted intersections. Total night crashes were also lower at lighted intersections.

### **2.1.2 Phase II**

Phase II of the research was carried out by CTRE to research the quality of illumination at the previously monitored intersections as opposed to simple presence of light. CTRE partnered with VTTI to implement a measurement system for this research. They developed a system that contained multiple light meters collecting measurements from behind the windshield (Figure 4). The data collected was paired with GPS coordinates to map lighting data on the approach of intersections.

While this system was successful in measuring lighting data as the vehicle approached an intersection, the configuration of the device does not align with the methodology of collecting lighting data. This set up was designed to measure illumination from a driver's perspective, as the light meters were mounted behind the windshield. In order to collect more general information on the illumination of roadways the system must be arranged so that data collection occurs parallel with the roadway, since illumination standards are defined as the values 6" above the roadway.



Figure 4: Measurement system developed by CTRE and VTTI [4]

## 2.2 Study 2 – University of South Florida

In 2008, the Center for Urban Transportation and Research (CUTR) at the University of South Florida implemented a mobile system for measuring roadway illumination. This system was tested along major Florida roadways with the intent of integrating the results into the Florida Department of Transportation's roadway database [5].

The system created by CUTR used light meters mounted on the top of a vehicle, a DMI, and a laptop computer (Figure 5). This system relied on a laptop and software to integrate data between the light meters and the DMI. This design did not include GPS location data. By relying solely on the DMI for location, the system was

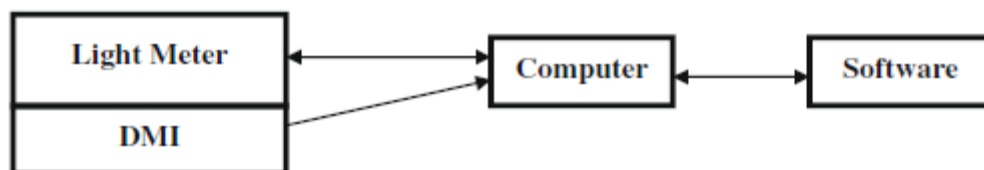


Figure 5: CUTR illumination measurement system [5]

vulnerable to location accuracy issues, as there is no way to verify where the measurements took place. The usage of a laptop instead of an embedded system reduces the degree of automation, requiring more user input.

### 2.3 Study 3 – Helsinki University of Technology

The Helsinki University of Technology performed a study on the effects of weather on roadway illumination [12]. This study was completed using non-conventional measurement systems, relying on a CCD camera as opposed to using standard light meters. The study was performed in five test locations throughout Finland and was conducted in various seasons and weather conditions. The research hypothesis was that certain weather conditions increase the reflectivity of the road, making an excess of illumination dangerous.

The measurement system uses a CCD camera mounted at 1.5 meters in an attempt to mimic how a driver would perceive the road [12]. The data is collected by the camera and is later analyzed using computer models. The results can be shown as a heat map of the roadway (Figure 6).

By mounting the measurement system at eye level parallel to the roadway, the Helsinki University of Technology research is not consistent with the US standards

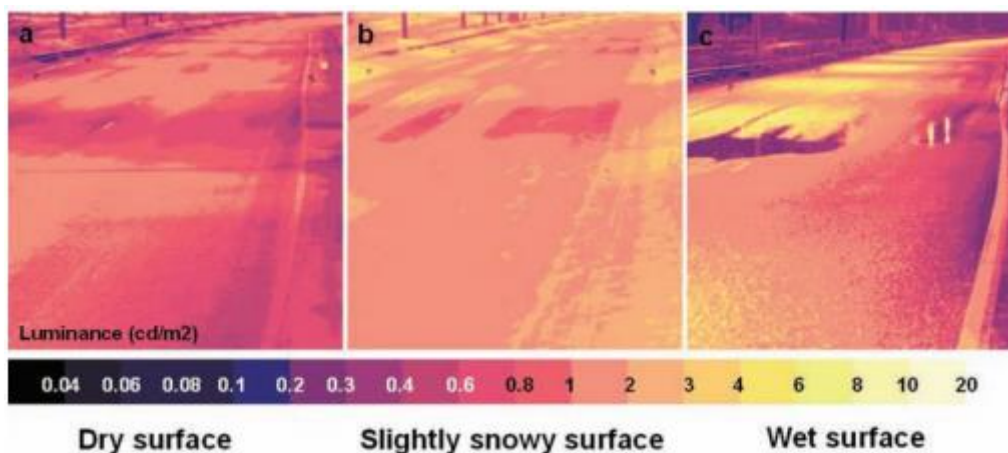


Figure 6: Results of CCD camera illumination measurements [12]

of illumination measurement. The research focuses on the driver's perspective of roadway lighting as opposed to the overall illumination present in the road segment. This research is also incompatible with conventional measurement systems by using a CCD camera to measure illumination. This setup does not allow for standard illumination measurements, as it relies on picking up illumination data from hard surfaces.

## **2.4 Summary**

This chapter outlined multiple systems that have been developed to measure roadway illumination. While the solutions presented were able to measure lighting data, they each had issues preventing them from being viable solutions to large scale data collection. The system proposed by Iowa State University did not meet the requirements for illumination measurement because the light sensors were not mounted parallel to the road. University of South Florida's implementation lacks a GPS, making it unable to verify location. This solution also requires more user input, as the system is running on a laptop requiring user action. Helsinki University of Technology focuses on the driver's perspective, taking measurements facing forward as opposed to the direct illumination on the roadway. It also uses a non-standard measurement system, opting for a CCD camera instead of a conventional light meter.

## CHAPTER 3: PROPOSED EMBEDDED SYSTEM

The proposed embedded system was designed to be small, reliable, and to read and store data automatically while traveling up to 30 mph. The only user interaction is in terms of providing a filename and logging important events. Adding higher performance to the system to enable travel faster than 30 mph was not a consideration for this phase of the research.

### 3.1 Conceptual Diagram

The embedded system is composed of seven subsystems, shown in Figure 7, integrated to provide the necessary functionality. The sections that follow will describe each subsystem in detail.

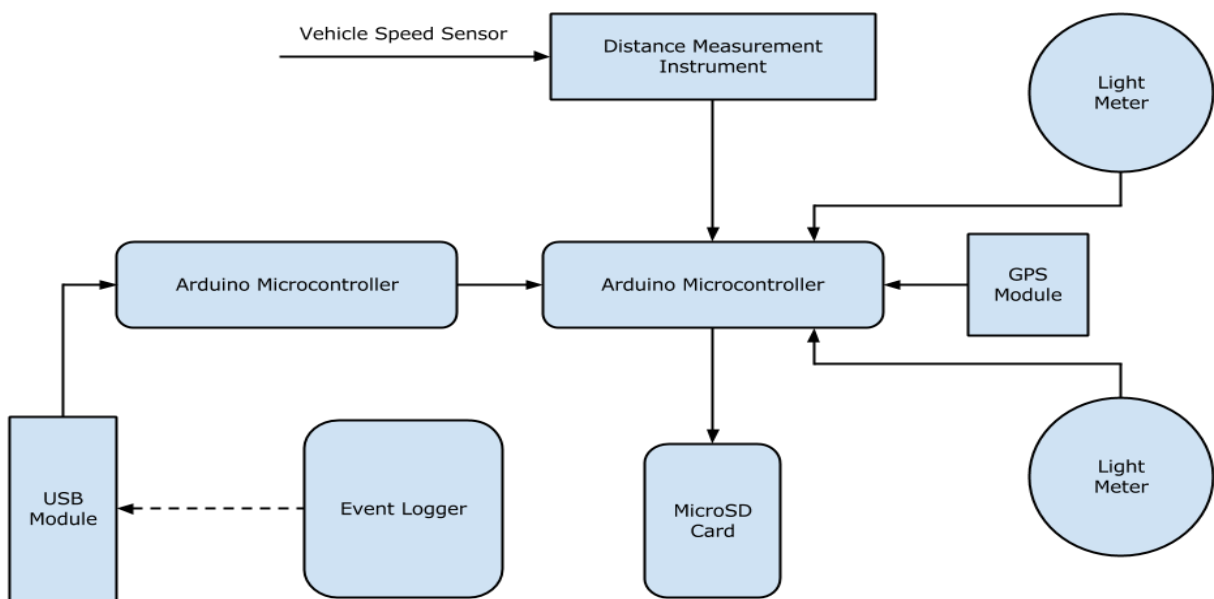


Figure 7: Conceptual diagram of ALMS

### 3.2 Arduino Based Design

The Advanced Lighting Measurement System (ALMS) is based on the Arduino Uno development board (Figure 8). This is a widely used platform for prototyping as there is a large community of developers who have created open source libraries for additional functionality. Along with libraries, there are a significant number of Arduino shields, boards designed to integrate various components with the Uno by stacking on top of the development board.

#### 3.2.1 Arduino Microcontroller Board

The microcontroller is the most important component in any embedded system. The Arduino Uno development board comes equipped with an ATmega328 microcontroller. The ATmega328 is an 8-bit microcontroller with 32 data registers and 32 KB of flash memory for program storage [7]. This gives plenty of space to store the necessary libraries along with the code necessary to implement the

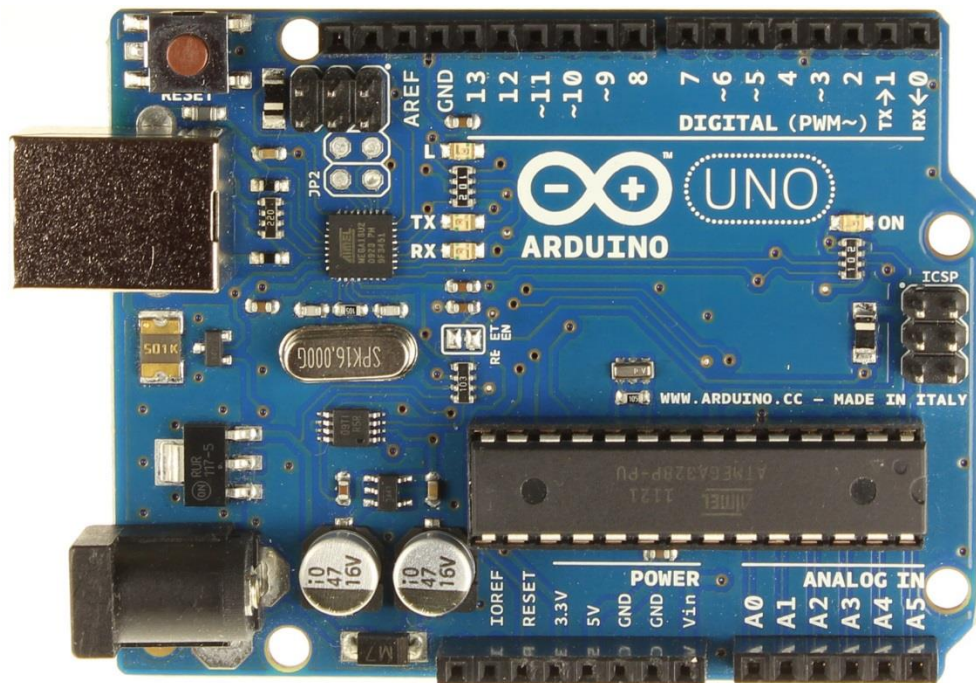


Figure 8: Arduino Uno development board [6]

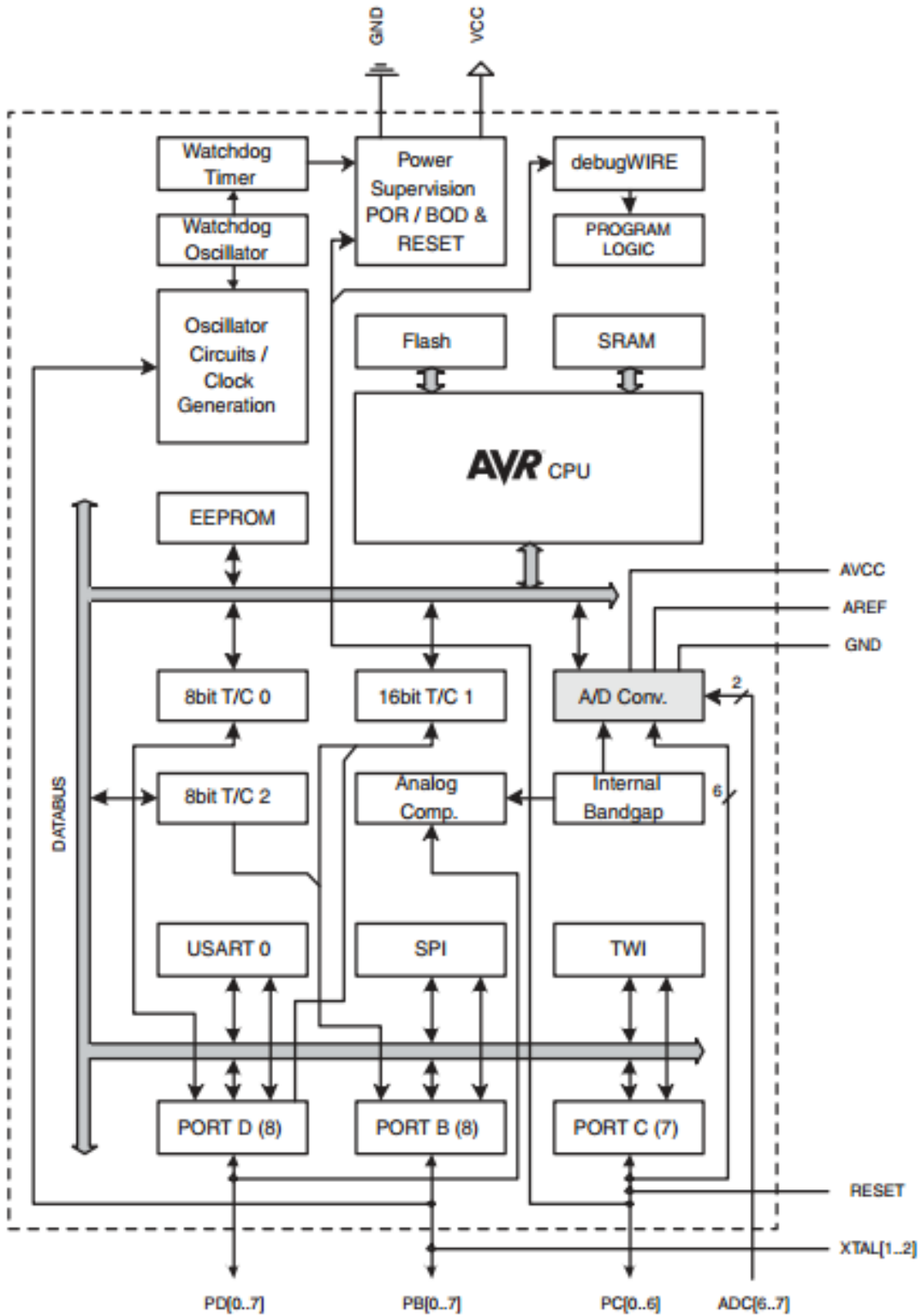


Figure 9: ATmega328 block diagram [7]



desired functionality. Another major benefit of using the ATmega328 is the built in Analog to Digital Converter (ADC). This is a necessary component for the ALMS design, as it relies on analog output from the light sensors and an internal algorithm to convert to foot candles. The Atmega328 has the following pins available to the system (Figure 10):

- Power pins – These pins are used to provide 5V to additional components. There is also an analog reference pin used to provide the upper limit for the ADCs. In the case of the ALMS design this pin is provided 3.3 V to maximize resolution of the light meter output.
- Serial Pins – The ATmega328 has one set of serial pins (TX and RX), used to transmit and receive data serially. These pins are used when programming the microcontroller and are also used to communicate with the GPS in the ALMS design.
- SPI – These pins are able to communicate with components using the Serial Peripheral Interface. SPI requires the usage of four pins:

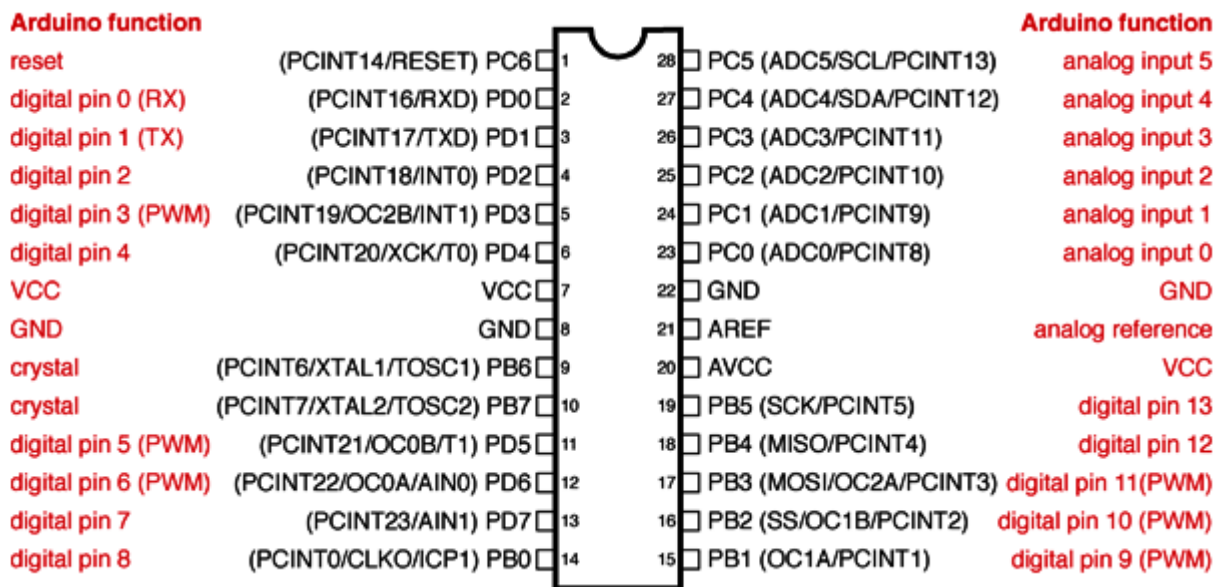


Figure 10: Pin mapping between ATmega328 and Arduino Uno board

- MISO (Master In Slave Out) – The slave line used to send data to the master from peripherals. This is pin 12 on the Arduino board.
- MOSI (Master Out Slave In) – The line used for the master device to communicate with peripheral devices. This is pin 11 on the Arduino board.
- SCK (Serial Clock) – This pin is used to pulse and synchronize data transmission. It is mapped to pin 13.
- SS (Slave Select) – This line is necessary for each slave device, only when a peripheral has a “low” value on this line can it communicate with the master. This is pin 10 on the Arduino.
- Digital Pins – The ATmega328 has a total of 14 digital input/output pins, including the serial and SPI pins mentioned above.
- Analog Pins – There are 6 analog input pins, each capable of using the built in ADCs of the ATmega328. These pins can also be configured as digital input/output pins.

The ATmega328 has more than enough analog and digital pins to interface with all of the ALMS subsystems. However, due to timing constraints involved with collecting data, the system required an additional Arduino Uno board used to interface with the USB module described later in the chapter.

### **3.2.2 GPS Module**

The ALMS design incorporates a Fastrax UP501 GPS device. This is a small, inexpensive, and accurate GPS module with 66 channels to improve locking time and satellite reception. The UP501 transmits data serially using the National Marine Electronics Association (NMEA) standard for data. The NMEA strings sent from the

GPS device include date, time, location coordinates, speed, signal quality, and data age. It is also possible to send NMEA codes to the GPS module to change various default settings. The following are settings changed during the execution of the ALMS program:

- Return RMC sentence only – This setting makes the GPS return only the recommended minimum location data. This is the only information needed for the ALMS and receiving only this data increases the possible throughput of the GPS.
- Set frequency – The GPS tracking frequency is changed from 1 Hz to 10 Hz to increase the rate at which location is updated. This is a necessary change in order to have accurate data for each reading while traveling at the desired speed of 30 mph.
- Enable WAAS – Wide Area Augmentation System (WAAS) is used to refine the GPS location by using known reference locations to adjust for any latitude or longitude errors. It can improve accuracy to within 3 meters on 95% of all readings.

Table 1: UP501 pin description [8]

Contact	Signal Name	I/O	Signal Description
1	RX	I	UART input port
2	TX	O	UART output port
3	GND	I	Ground
4	VDD	I	Main Power Supply
5	VDD_B	I	Backup Power Supply
6	PPS	O	Pulse Per Second Output

- Set BAUD rate – The default BAUD rate for the UP501 is 9600. However, by increasing the refresh rate to 10 Hz, the BAUD rate must be increased to 38400 in order to handle the increased throughput.

In order to use the UP501 with the Arduino Uno it was necessary to incorporate a GPS shield. This shield handles the power requirements of the GPS, regulating the voltage from 5V to 3.3V. It also routes the VCC, GND, RX, and TX pins from the GPS module to the appropriate pins on the Uno. The connections for the GPS are visible in the center of the shield shown in Figure 11.

### 3.2.3 Event Logger

One of the required feature of the ALMS is the ability to log important events while collecting data. This is accomplished through the event logger system which employs a wireless USB device. The event logger is used to provide location data for illumination values we anticipate to be unusual. It is also used as an aid for post-processing, by providing landmarks for aligning individual lanes along the roadway.



Figure 11: GPS shield for Arduino Uno [9]

This allows for segments to be forced into a straight path through the lane to adjust for GPS location error. By having start codes and crosswalk codes the data can be aligned with a satellite map overlay.

The event logger is composed of a standard wireless number pad. This allows for ease of use while the data is being collected by eliminating excess wires and having a familiar layout. The wireless receiver for the number pad has an LED which flashes when data is received, providing a failsafe mechanism to determine whether an event was properly logged.

The event logger number pad is also used for naming data files prior to collection. The filename is a five digit number with digits one and two being the day of data collection, digits three and four being the segment ID, and the fifth digit being the lane number in the segment. This approach provides well organized files once the data is transferred from the device to a computer, easing the map alignment and post-processing.

Table 2: Event logger code descriptions

Event Code	Type	Description
1	Intersection	Beginning of intersection
2	Intersection	End of intersection
3	Light Pole	Location of luminaire
4	Blockage	Blockage of luminaire (tree, etc.)
5	Crosswalk	Location of crosswalk
6	Special Source	Any light source other than luminaire (car dealer, advertising signage, etc.)
7	Vehicle	Oncoming vehicle light, either in front of behind

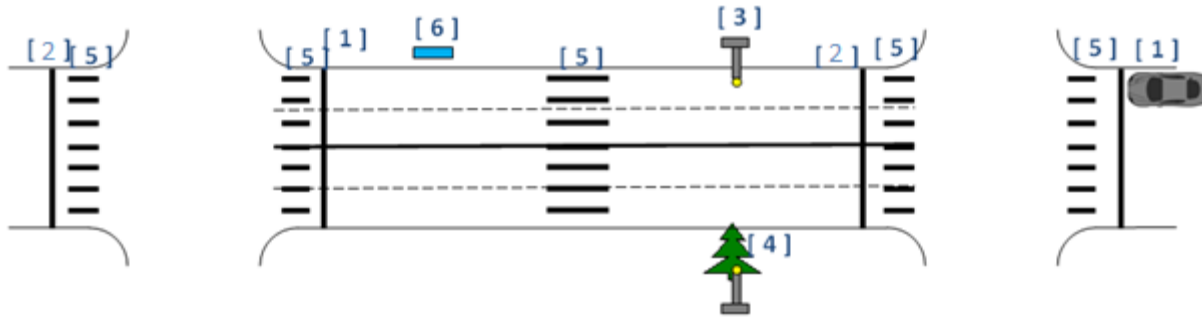


Figure 12: Examples of event codes

### 3.2.4 USB Host Module

In order to integrate the event logger into the Arduino based system, a USB host shield is necessary. This component allows a USB device to be connected to the Arduino Uno with the Uno as the host device. The USB host shield interfaces with the Arduino using SPI protocol, providing fast and accurate transmission of data. The main function of the shield uses a Maxim MAX3421E USB peripheral/host controller

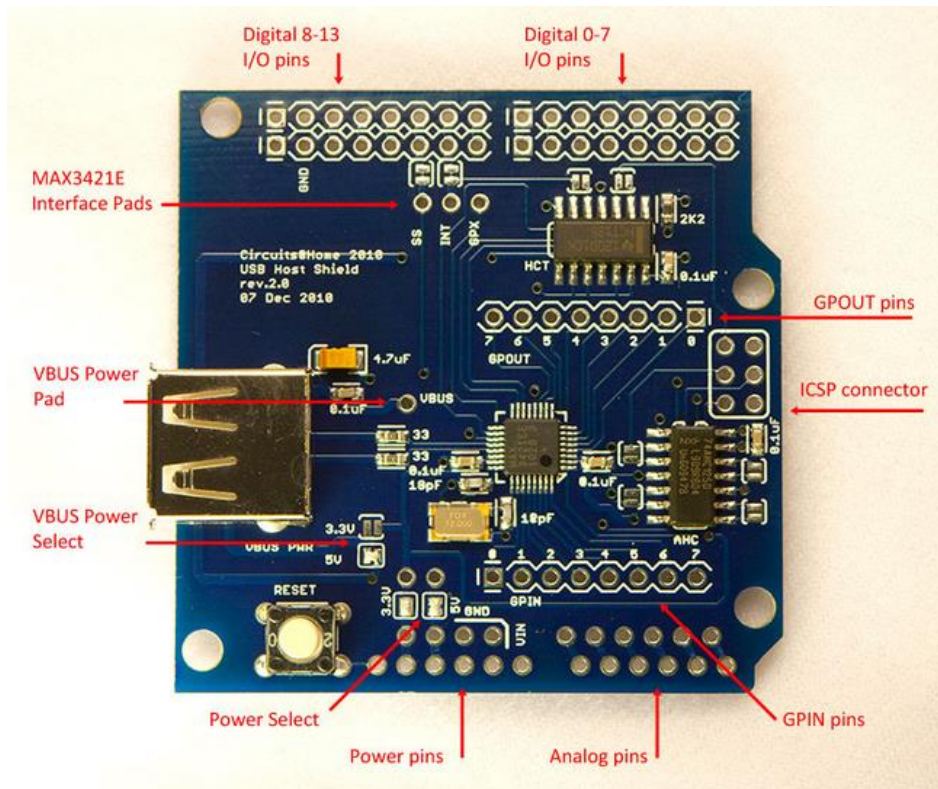


Figure 13: Pin mapping of USB Host Shield [9]

with SPI interface [9]. This chip provides the interpretation of USB protocol and converts the signals into SPI format.

This shield is connected to the secondary Arduino device, which is used only for communication with the event logger. With the USB keyboard library running on an Arduino, there is a busy wait function. Because of this, the event logger cannot be running on the primary Arduino as it would interfere with receiving the necessary signals to collect data as designed.

The supplied USB library required modifications to properly send the event inputs between the two Arduino microcontrollers. The library is designed to provide troubleshooting information via the serial lines of the Uno. However, the communication between secondary and primary microcontrollers occurs over the serial lines. In order for the correct information to be read by the primary device, changes had to be made to eliminate troubleshooting information being sent serially. It was also necessary to process the data received from the USB number pad to disable certain outputs and only transmit the relevant ASCII numerical character. The library had to be modified because it was built for a full keyboard, which changes some of the key mapping when compared to only using a number pad.

### **3.2.5 MicroSD**

Due to the limited flash memory included on the Arduino Uno it is necessary to add storage by utilizing a microSD card to save all information produced during data collection. The Uno does not have support built in for saving to microSD cards so a host shield was necessary. This shield communicates between the Arduino and the microSD socket using SPI protocol.

Table 3: MicroSD card pin mapping

Pin Number	Pin Name	Signal Function
1	NC	No Connect
2	CS	Chip Select
3	DI	Master Out/Slave In (MOSI)
4	Vdd	Supply Voltage 2.7v / 3.6v
5	CLK	Clock
6	GND	Ground
7	DO	Master In/Slave Out (MISO)
8	RSV	Reserved

The ALMS software initializes by requesting a filename for the current segment. This is entered through the event logger number pad and written to the microSD card as a csv (comma separated values) file. This allows simple implementation of an Excel readable file. If a filename is entered in error, it can be erased by simply resetting the program. As long as the data collection process has not begun the file will be empty and will not be stored on the microSD card.

As data is collected, the current segment file is opened and the data is appended to the file through SPI. After the data is written the file is closed. By opening and closing the file with each write, the file is protected from corruption that could occur with sudden loss of power to the device or with a reset of the software before file closure. Once the current segment is complete, the program is reset and will then await a new filename input to begin data collection on the next segment.



### 3.2.6 Distance Measurement Instrument

To maintain the most accurate distance traveled, which is required for taking readings at distance intervals, a distance measurement instrument (DMI) is necessary. The RAC Plus I by Jamar Technologies was chosen based on the requirements for this system. This DMI is highly accurate, maintaining an error rate of less than one foot per mile.

There are multiple components required to integrate the RAC Plus I with an Arduino based system. The first component is a modular distance sensor (MDS). This device is provided with the RAC Plus I and must be connected to the vehicle's speed sensor in order to give the DMI accurate distance readings. The MDS receives speed signals from the vehicle, filters them, converts them to distance readings, and signals the RAC Plus I to increment the counter [10].

The RAC Plus I requires calibration before it can be used to measure distances. It has an automatic calibration mode requiring the operator to drive along a path of known distance, which is entered into the DMI at the beginning of calibration. Calibration is started, the vehicle is driven along the known path, and then calibration is ended. The RAC Plus I automatically uses this data to generate a calibration factor to adjust its functionality to the individual vehicle.



Figure 14: RAC Plus I distance measurement instrument [10]

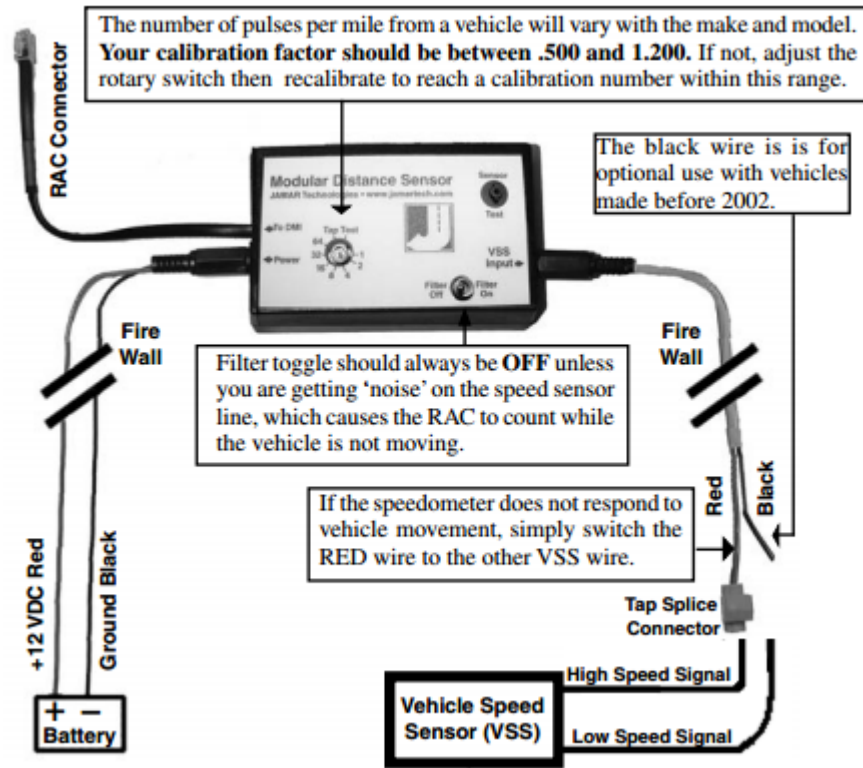


Figure 15: Installation instructions for MDS [10]

The DMI interfaces with the Arduino microcontroller via one of the Arduino's digital input pins. The DMI is configured to raise this pin high every ten feet the vehicle travels. This change on the pin will be observed by the microcontroller and the program will execute the appropriate functions. The steps necessary to configure the DMI to perform the correct operations are as follows:

1. Power on the DMI
2. Enter the menu of the DMI and press '5' to enter the Distance Pulse Output (DPO) configuration
3. Enter desired distance interval (for the ALMS this is 10 feet)
4. Enter signal length ('2' for ALMS which implies length of 20 ms)
5. Set audible DPO trigger (set to '0' to turn off sound at DPO triggering)

The above process instructs the DMI to monitor the distance as it accumulates and send a signal back to the MDS once the target interval is reached. This signal is passed through the MDS and onto the Arduino pin, triggering execution of the data collection code. Using this method, it automatically ensures that lighting and GPS data will be collected at intervals of ten feet without any user interaction required.

### 3.2.7 Light Meters

The most important components in the system are the light meters used to measure illumination values. For the ALMS the Konica Minolta T-10A light meter was selected. This is one of the most accurate light meters on the market, being able to measure up to 0.001 fc with accuracy within two percent. The relative spectral responsivity of the T-10A is within six percent of the spectral luminous efficiency [11]. This is a measure of how the human eye perceives light at different wavelengths (Figure 16), meaning the closer a light meter can approximate this curve the more accurately it represents what the eye can see.

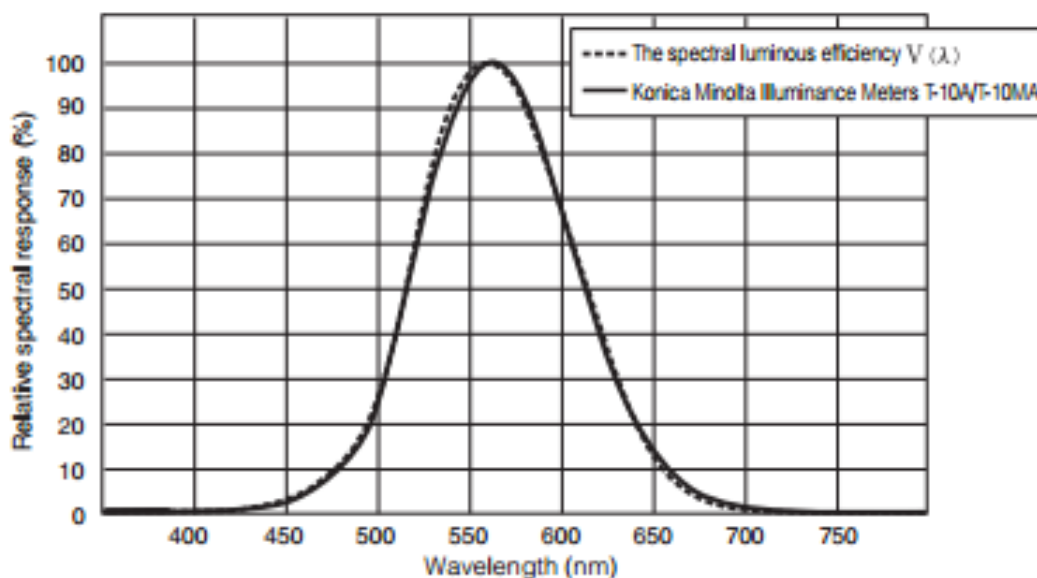


Figure 16: T-10A relative spectral response [11]

Other design factors exist that make the T-10A an attractive choice for use in the ALMS system. The ability to detach the receptor head from the main body of the meter allows the majority of the device to be inside the car, minimizing risk to the electrical components. The receptor head is the only part of the device located on the outside of the vehicle, connecting to the body through an Ethernet cable. For the ALMS design, there are two light sensors mounted on the top of the vehicle using a custom mounting solution (Figure 17). This configuration allows for more accurate light readings and closer mimics the conventional method of illumination data collection.

Another capability of the T-10A is the option to output the illumination values through an analog connector. This is a crucial ability in the ALMS design, enabling the Arduino to read illumination values. If analog output were not possible, a computer would be required to interpret the light meter readings. The analog connector plugs into the receptor head and provides a continuous voltage output in



Figure 17: Light meters mounted on top of vehicle

the range of 0-3V. This output connects to one of the analog input pins of the Arduino where the ADC is capable of converting the voltage level to a foot candle reading using the conversion factors provided in Figure 18.

When the T-10A is connected with the analog plug it requires a static manual operating range (Table 4), as changing the range changes the multiplier required to convert the voltage to a foot candle reading. The following steps must be taken to connect the light meter to the ALMS and configure it in analog output mode with the proper range selected:

1. Connect T-10A receptor head to body using Ethernet cable
2. With the lens cover on, power on the T-10A and wait for automatic calibration to finish
3. Connect the analog output plug to the receptor head, thus locking in the desired range
4. Remove lens cover

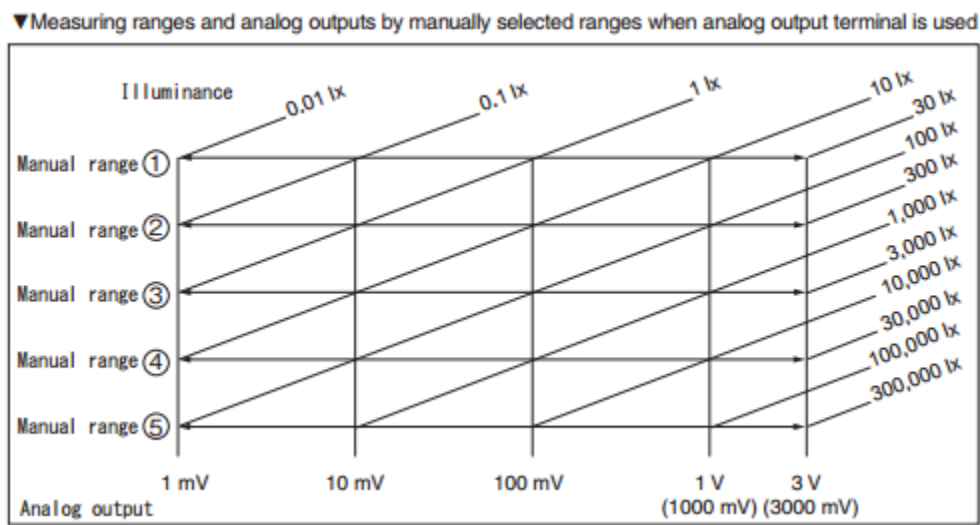


Figure 18: Analog voltage to lux conversion factors [11]

Table 4: Manual and automatic fc range

Range	Manual	Auto
1	0.000 – 2.999	0.000 – 2.999
2	0.00 – 29.99	2.50 – 29.99
3	0.0 – 299.9	25.0 – 299.9
4	0 – 2999	250 – 2999
5	0 – 29990	2500 – 29990

### 3.3 Software

Throughout this chapter elements of the ALMS program have been discussed as they related to the corresponding hardware components. In this section the software will be looked at in further detail, providing a program flow chart and briefly reviewing the important functions in the ALMS software design. The program includes software libraries to incorporate the various hardware components and ties them together to meet the system requirements.

#### 3.3.1 Program Flow Chart

The ALMS program only requires user input to initialize the segment filename and to log events. There is only one I/O operation and it only executes when necessary to increase performance. The main code loop has minimal code in order to respond quickly to DMI pulses and event logger inputs. The program flow chart is shown in Figure 19.

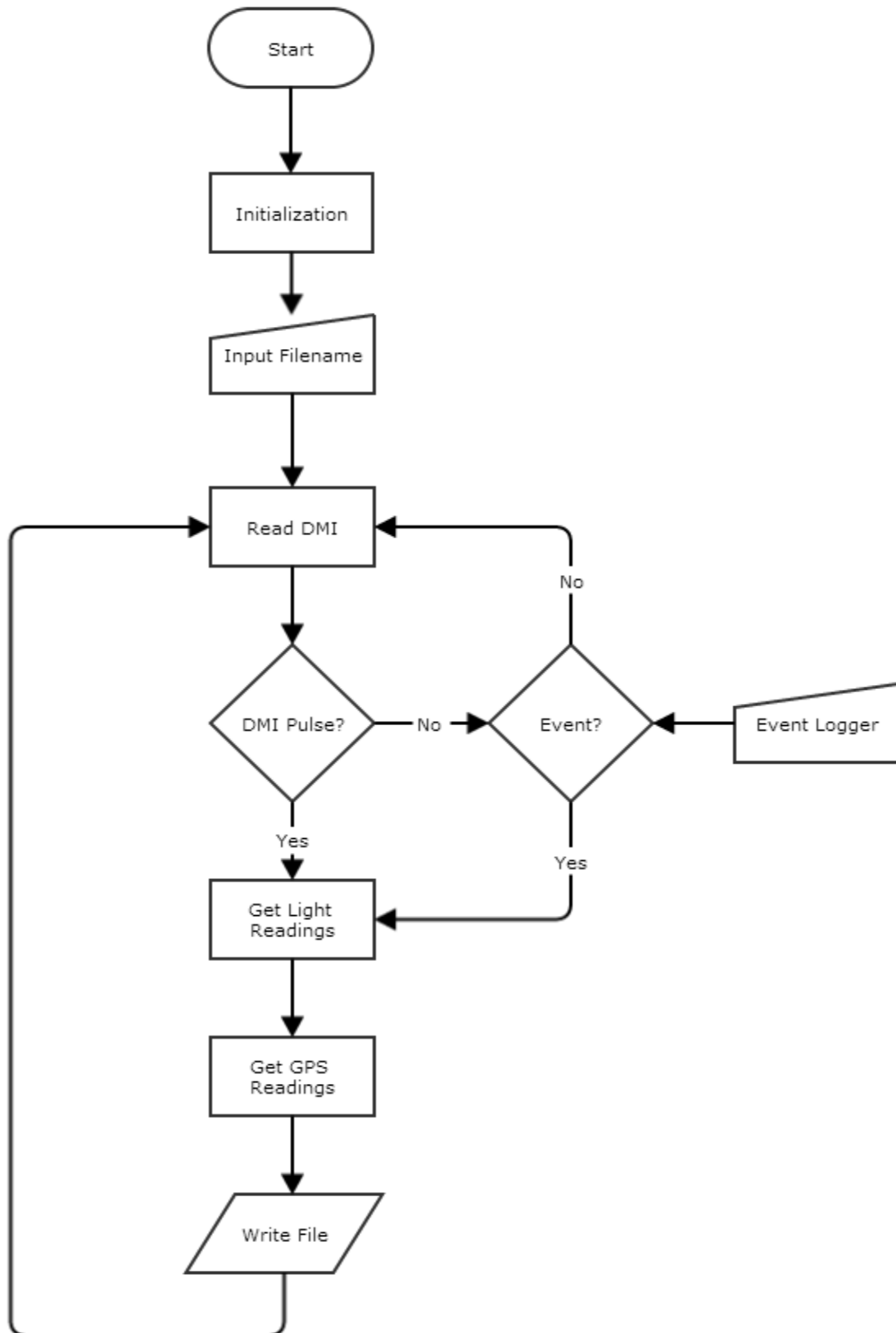


Figure 19: ALMS program flow chart

### 3.3.2 Program Description

The ALMS program incorporates the following Arduino libraries:

- TinyGPS – This library supplies functions to communicate serially with the GPS module. It was modified to correctly map the pins between the GPS shield and the Arduino.
- SoftwareSerial – This library creates serial communication by initializing digital pins as RX and TX. This library did not need modification.
- SD – This library provides function calls to make use of the microSD card. It allows reading/writing of files. It was modified to correctly map the pins between the microSD shield and the Arduino.
- USB – This library is used by the secondary Arduino to communicate with the wireless number pad (event logger). It was modified to eliminate troubleshooting information being sent over the serial lines.

### 3.4 System Cost

The cost of the ALMS system is broken down by part as shown in Table 5. If this were to be expanded into a widespread study with multiple vehicles the cost would be reduced by bulk purchases.

Table 5: Cost breakdown by part

Part	Unit Price	Quantity	Total Price
Arduino Uno	\$29.95	2	\$59.90
UP501 GPS	\$54.87	1	\$54.87
GPS Shield	\$14.95	1	\$14.95
MicroSD Shield	\$14.95	1	\$14.95
MicroSD Card	\$6.99	1	\$6.99
USB Host Shield	\$25.00	1	\$25.00
Wireless Number Pad	\$31.99	1	\$31.99
RAC Plus I DMI	\$595.00	1	\$595.00
T-10A Light Meter	\$1,440.10	2	\$2,880.20
Total			\$3,683.85



### **3.5 Summary**

This chapter describes the proposed system in detail, explaining purpose and defining characteristics of each component. The most critical components in the system are the T-10A light meters, the RAC Plus I DMI, and the Arduino microcontrollers. The software on the secondary Arduino is responsible for handling the USB requirements of the event logger. This allows for faster processing on the primary system. The primary Arduino software integrates all subsystems and takes illumination readings when the DMI sends a pulse signaling 10 feet have been traveled or when an event is received from the secondary Arduino.

## **CHAPTER 4: EXPERIMENTAL RESULTS**

For the first phase of this research, ALMS was prototyped and validation tests were performed to ensure that it is a viable solution to measuring roadway illumination while traveling at 30 mph. Also during this phase, 100 centerline miles were driven to further test and refine the device while collecting illumination data for the Florida Department of Transportation. Data was then verified, cleaned, and processed to comply with GIS standards for roadway information.

### **4.1 System Validation**

The ALMS was put through two separate validation tests to insure the system provides accurate illumination data in both a static and from a dynamic environment. The first validation test was performed in order to verify consistency between standard light meter readings and ALMS light meter readings. The second test was created to verify illumination data integrity while the system is in a real world situation, traveling at 30 mph.

#### **4.1.1 Static Test**

The purpose of a static validation test is to compare illumination readings between the light meters operating in automatic range mode while displaying the digital readings and the readings acquired by the ALMS via the light meters outputting analog voltages in the manual range mode. This test was conducted on the Tampa campus of University of South Florida near the Research Park.

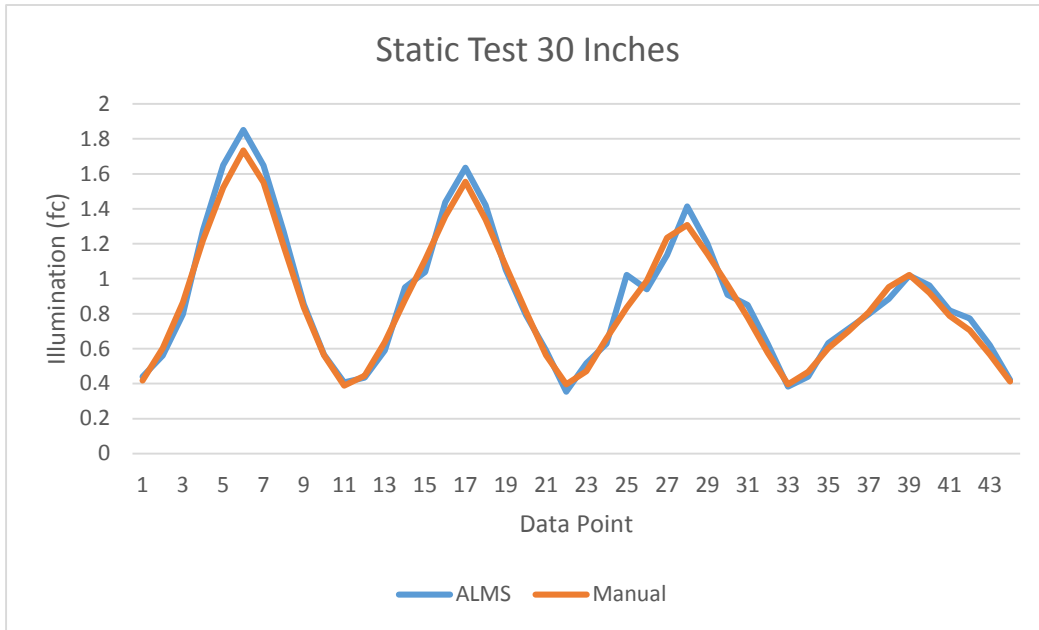


Figure 20: Static results at 30 inches

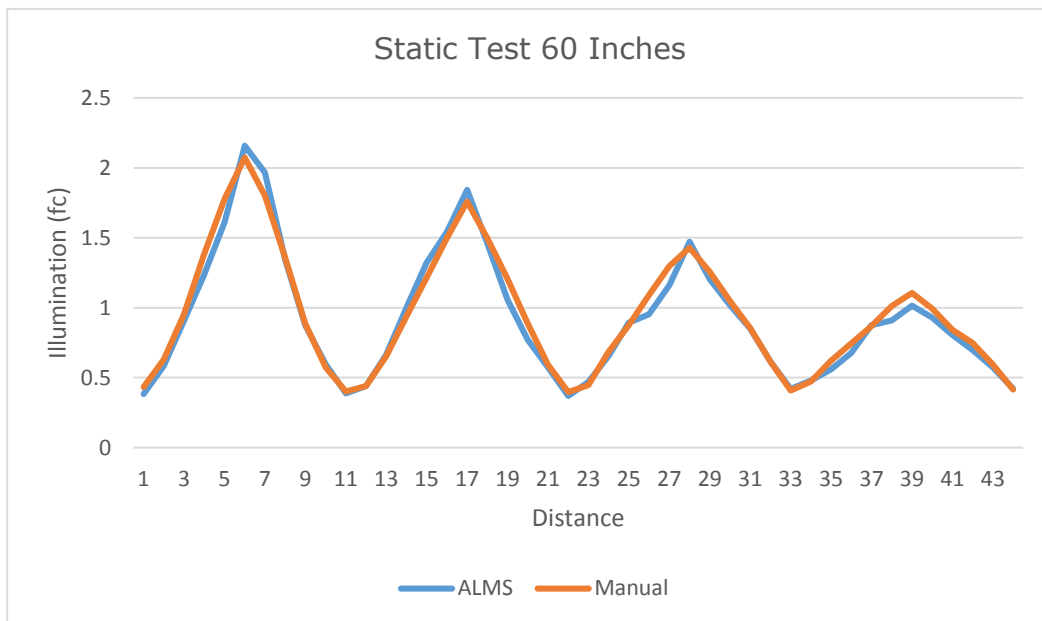


Figure 21: Static results at 60 inches

The results of the static test were positive, showing that there were only slight differences between manual and ALMS data collection. The range of error between the two can likely be attributed to natural fluctuation in illumination levels, as repeating exact measurements in the same location was difficult even when using

the same measurement method repeatedly. With the illumination values on the low range of the sensor and the light meter being a sensitive piece of equipment, some variation between subsequent readings is to be expected.

#### 4.1.2 Dynamic Test

The second validation test performed was intended to verify equality of data between stationary and moving readings. This test was performed using only the ALMS. The first set of data was collected with the vehicle at rest at each distance while the second set of data was measured with the vehicle traveling at 30 mph. Data was collected over 120 feet. For the stationary readings measurements were taken every 10 feet with additional measurements 2 feet to either side. These three values were averaged to bracket the desired location under the assumption that the moving illumination reading would not be exactly in phase with the stationary readings. Moving measurements were taken over 20 separate runs and the illumination values were averaged over the 20 readings for each distance. This was done to minimize error from being out of phase with the stationary readings.

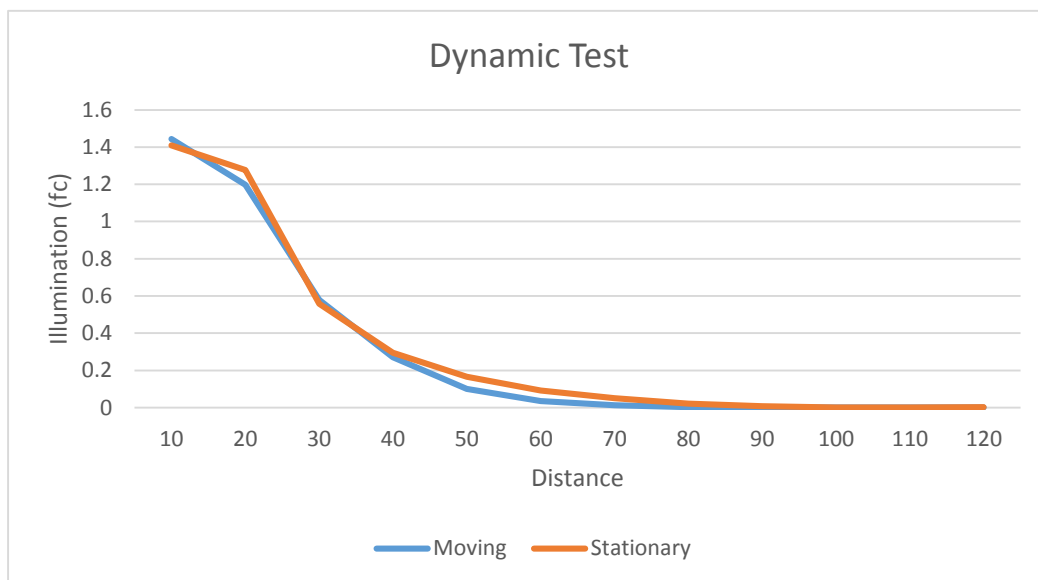


Figure 22: Dynamic validation test

The results for the dynamic validation test were also very promising, as there were minimal differences between the moving and stationary illumination readings. These minor discrepancies can likely be attributed to the problem of aligning the starting point of the moving test with the starting point of the stationary readings. This problem was anticipated and actions were taken to adjust for this as much as possible but it cannot be known how much error still remained. The results of this test were positive enough that large scale data collection over 100 centerline miles was commenced.

## **4.2 Location Data**

Location data was recorded by two different methods at each illumination measurement: GPS location and DMI distance traveled. These separate data points were necessary to ensure maximum location accuracy in data reports. The GPS coordinates were used to determine alignment of start and end points while the DMI distances were used to force data along the actual path of the roadway. This combination resulted in logged events (intersections, crosswalks, etc.) aligning with those same locations on satellite imagery.

### **4.2.1 GPS Location**

Once the segment data was collected it became necessary to realign the locations supplied by GPS along paths created through satellite imaging on Google Earth. While the GPS was accurate in most cases, there were times when the location data was significantly displaced from the lane in which the vehicle was traveling (Figure 23). This problem was mostly evident in areas with an abundance of trees, buildings, and other objects that can block GPS satellite reception.



Figure 23: Location data for Fletcher Ave

#### **4.2.2 DMI Location**

While the DMI does not provide exact location data, it generates very precise distance measurements from the starting location. The most difficult aspect of the DMI is that it cannot directly interface with the rest of the system. The distances are measured when the DMI determines a 10 foot interval and sends a pulse to the Arduino. This becomes a problem when events are logged, as illumination values are recorded but the system cannot query the DMI for the current distance traveled. This aspect is another instance where it is a benefit to have two location services, as the distance can be calculated by a combination of GPS coordinates and time elapsed between DMI pulses.

#### **4.3 Output File Cleaning**

Before the output data could be converted to GIS format it was necessary to clean the files generated by the ALMS. This process was intended to fix the following issues:

- Extra illumination readings – These can occur before/after designated segment endpoints while the DMI is still running. Segment beginning and end are marked with event logger codes
- Missing distance values – This issue occurs when events are logged. It is caused by the lack of DMI query ability. It can be fixed using elapsed time between distance pulses in combination with vehicle speed. It can also be fixed by calculating the orthodromic distance between the surrounding GPS coordinates.
- Measurement offset – This is caused by an event being logged while the system is taking a reading initiated by a DMI pulse. It causes a data logging measurement (which should have a distance of 0) to register the distance generated by the DMI, resulting in one row of data where there should be two. This creates an offset of 10 feet, which compounds on every occurrence. It can be fixed by splitting the row of data into two distinct measurements, one for the event and one for the DMI pulse.

To fix these issues the data must be run through a program created for this research by Dr. Zhenyu Wang from the Center for Urban Transportation Research at the University of South Florida (Figures 24 & 25). Once the data has gone through the program it can be aligned with lane routes and converted to GIS format.

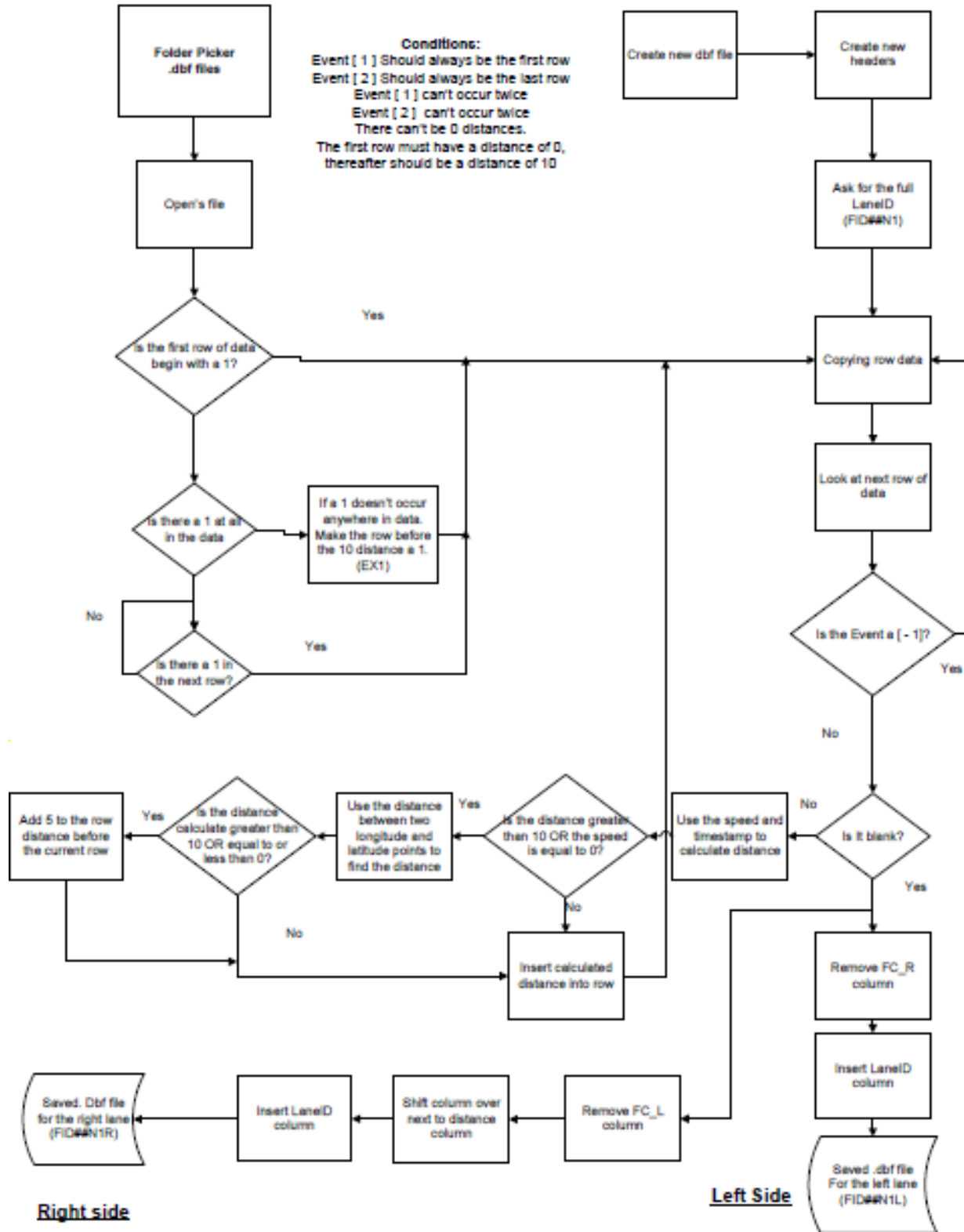


Figure 24: Flowchart for data cleaning program



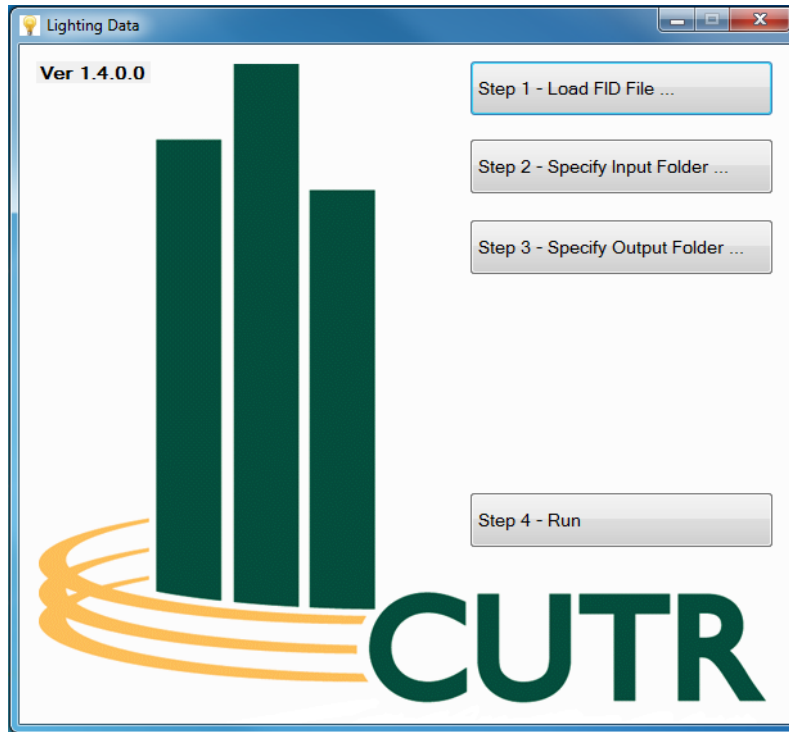


Figure 25: Data cleaning program. Created by Dr. Zhenyu Wang

#### 4.4 GIS Representation

Once cleaned, the collected illumination data must be converted into GIS format. This is a multi-step process involving tracing the routes of each lane on Google Earth and using this data to align the illumination measurements from each sensor to the corresponding route.

##### 4.4.1 Route Tracing

Using the add route functionality on Google Earth, routes can be traced through the center of each segment lane (Figure 26). This data will provide start/stop points and a guideline through the segment for each data file. Lane routes are split into routes for the individual light meters to provide more accurate illumination placement (Figure 27). Once this is completed, the collected data can be aligned properly within the segments.



Figure 26: Lane routes for Fletcher Ave created in Google Earth



Figure 27: Light meter routes on Fletcher Ave

#### 4.4.2 Map Overlay

Once the paths are created for the illumination data of each light meter, the data can be placed along the routes. This process uses the distances created by the DMI and by the data cleaning program to space the measurements along the light meter paths. After aligning the data, the paths are calibrated by ensuring the start/stop points and crosswalks events match with the satellite imagery available in GIS. This results in evenly spaced placement of illumination data throughout the roadway (Figure 28). With the data in GIS format it is compliant with the standards of the DOT, allowing illumination levels to be incorporated into the existing roadway data. It can also be used to create a heatmap of illumination, showing exactly where the lighting is sufficient and where improvement is needed (Figure 29).



Figure 28: GIS representation of illumination data on Fletcher Ave

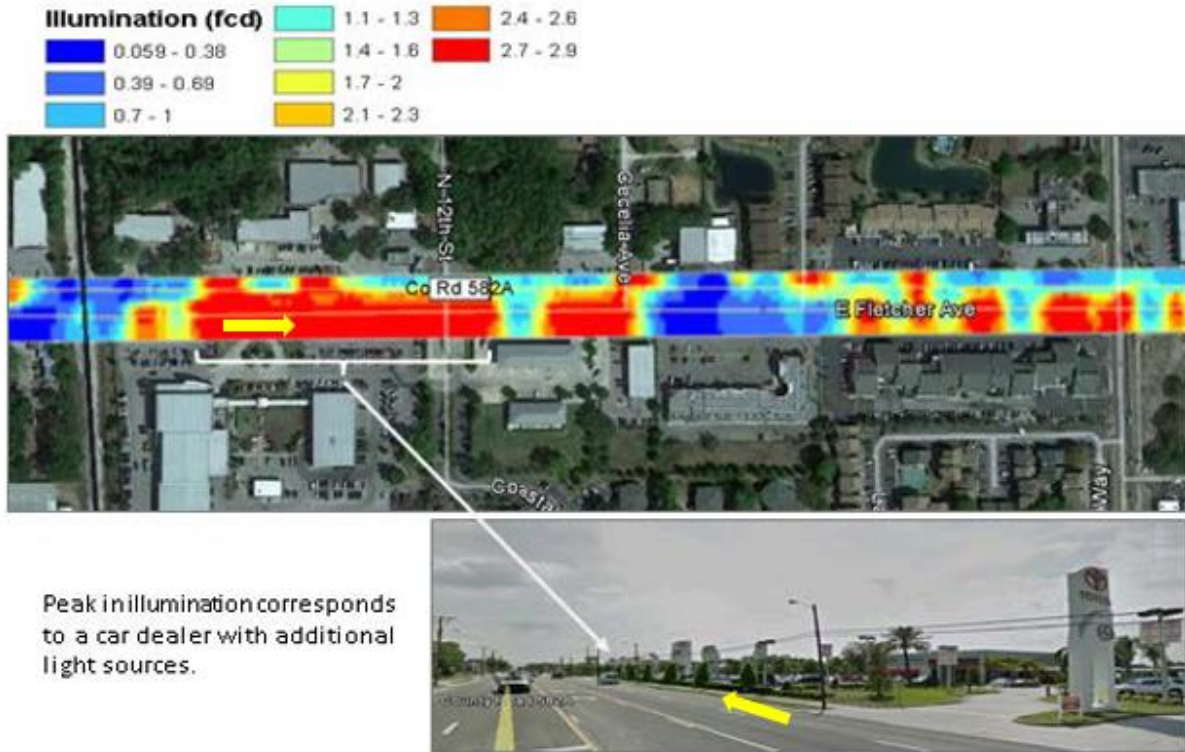


Figure 29: Heat map of illumination on Fletcher Ave. Created by Dr. Aldo Fabregas

#### 4.5 Summary

This chapter explains the system validation procedure, demonstrating that the results produced by the ALMS meet the expected results of conventional illumination measurements. The results and fine tuning process of a large scale (100 centerline miles) data collection undertaking are presented and explained in detail. This chapter also explains the data cleaning process and how illumination values are transformed into the GIS standard. The benefits of GIS format are outlined and an example of a GIS based illumination heatmap is given.

## CHAPTER 5: CONCLUSION AND FUTURE WORK

The ALMS system described in this thesis has been demonstrated to be an accurate and efficient method for measuring roadway illumination. There were several challenges faced during data collection and post-processing, especially false data being produced by the ALMS programming. This issue was not discovered until after the data collection period ended because it was slight enough to not be immediately noticed, yet drastic enough to cause significant problems with the final data set. These issues were all addressed during post-processing, but most could be fixed by changes made to the ALMS program. This would save large amounts of time on further data collection.

Future work will commence with Phase II of the ALMS data collection. This will involve making the noted changes to the software to accelerate post-processing. Further changes can be made to the system to enhance usability, such as an LCD display to make naming files and logging events easier for the user to verify. The system can also be implemented on a PCB to reduce the device footprint improve connections between components.

Another area that can be improved in future work is the portability of the system. Currently, the ALMS requires the MDS which ties the device to a single vehicle. In a future implementation alternatives to using the vehicles speed sensor to obtain the distance traveled can be explored. This would include using an OBD-II

adapter to measure vehicle speed. This approach would allow vehicle switching, since all vehicles are equipped with an OBD-II port. A final aspect of future work on the ALMS would be modifying it so data can be collected at a speed greater than 30 mph. This could be done by upgrading the microcontroller, allowing the distance pulses from the DMI to be shortened to 10 ms without the device missing any signals. This change could allow the ALMS to collect data at a speed of 40 mph.

## REFERENCES

- [1] Florida. Department of Highway Safety and Motor Vehicles. *Traffic Crash Facts Annual Report 2012*. Appriss, Inc., 2013. Web.
- [2] R. A. Hargroves, "Road lighting," *Physical Science, Measurement and Instrumentation, Management and Education - Reviews, IEE Proceedings A*, vol.130, no.8, pp.420-441. November 1983.
- [3] P. Morante, "Mesopic Street Lighting Demonstration and Evaluation Final Report." *Groton Utilities*. (2008). Web.
- [4] O. Smadi, N. R. Hawkins, and B. Aldemir-Bektas. *Roadway Lighting and Safety: Monitoring Quality, Durability, and Efficiency*. Ames, Iowa: Institute for Transportation, Iowa State University, 2011.
- [5] H. Zhou, F. Pirinccioglu, and P. Hsu. "A New Roadway Lighting Measurement System," *Transportation Research Part C: Emerging Technologies*, Volume 17, Issue 3, June 2009, Pages 274-284.
- [6] Arduino Technical Staff, *Arduino Uno R3*, Arduino, 2011.
- [7] Atmel Corporation, "Atmel 8-bit Microcontroller," ATmega328 datasheet, Feb. 2013
- [8] Fastrax, "Fastrax GPS patch antenna module," UP501 datasheet, May 2013.
- [9] Circuits@Home Technical Staff, *USB Host Shield for Arduino*, Circuits@Home, 2011.
- [10] JAMAR Technologies Technical Staff, *RAC Plus I Distance Measurement Instrument*, JAMAR Technologies, 2012
- [11] Konica Minolta Technical Staff, *Illuminance Meter T-10A/T-10MA*, Konica Minolta, 2012.
- [12] E. Aleksanteri, M. Eloholma, and L. Halonen. "Analysis of road lighting quantity and quality in varying weather conditions," *LEUKOS, The Journal of the Illuminating Engineering Society of North America*, volume 4, number 2, pages 89-98.