Design of a Highly Portable Data Logging Embedded System for Naturalistic Motorcycle Study

Noureddine Elmehraz

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

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

Part of the Computer Engineering Commons

Scholar Commons Citation

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.
Design of a Highly Portable Data Logging Embedded System for Naturalistic Motorcycle Study

by

Noureddine Elmehraz

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: Achilleas Kourtellis, Ph.D.
Swaroop Ghosh, Ph.D.

Date of Approval:
June 28, 2013

Keywords: Transportation, Arduino, DAS, SPI, I2C

Copyright © 2013, Noureddine Elmehraz
DEDICATION

To my loving family for their great patience and support.
ACKNOWLEDGMENTS

This work has been supported and funded by the Center for Urban Transportation and Research (CUTR) at USF. I am very thankful to CUTR and specifically to Dr Achilleas Kourtellis for his tremendous help. My sincere gratitude to Dr. Srinivas Katkoori for his advice and guidance. I would like also to thank Dr. Swaroop Ghosh for serving on my thesis committee. Thanks to you all.
# TABLE OF CONTENTS

LIST OF TABLES .................................................................................................................. iii

LIST OF FIGURES ............................................................................................................... iv

ABSTRACT ............................................................................................................................. vi

CHAPTER 1: INTRODUCTION .............................................................................................. 1
  1.1 System Approach .......................................................................................................... 2
  1.2 Methods .......................................................................................................................... 3
    1.2.1 Naturalistic Study .................................................................................................... 3
    1.2.2 Data Acquisition .................................................................................................... 3
  1.3 Other Studies .................................................................................................................. 4
  1.4 Proposed Solution – An Overview ............................................................................... 5
    1.4.1 Arduino Based System ........................................................................................... 5
    1.4.2 System Specifications ............................................................................................ 7
    1.4.3 Timing Constraint .................................................................................................. 7
    1.4.4 Test Plan .................................................................................................................. 7
    1.4.5 System Cost ........................................................................................................... 8
  1.5 Summary ........................................................................................................................ 8

CHAPTER 2: RELATED WORK ............................................................................................. 9
  2.1 Study 1 – Monash University [12, 21] ....................................................................... 9
    2.1.1 Stage 1 .................................................................................................................. 9
    2.1.2 Stage 2 ................................................................................................................ 10
    2.1.3 Stage 3 ................................................................................................................ 10
  2.2 Study 2 – VTTI and Monash University [23] ............................................................... 11
  2.3 Study 3 – NHTSA [24] ............................................................................................... 11
  2.4 Summary ...................................................................................................................... 12

CHAPTER 3: PROPOSED EMBEDDED SYSTEM .............................................................. 13
  3.1 DAS Design for Naturalistic Study ............................................................................. 13
    3.1.1 Low Power Consumption .................................................................................... 14
    3.1.2 Form Factor ......................................................................................................... 14
    3.1.3 Data Logging Duration ....................................................................................... 15
  3.2 Conceptual Diagram ................................................................................................. 15
  3.3 Arduino Based System ............................................................................................... 16
LIST OF TABLES

Table 1 SD card pin description ............................................................................................................................................... 26
Table 2 System cost breakdown ............................................................................................................................................... 32
Table 3 Power consumption experiment .................................................................................................................................. 32
Table 4 Power consumption using datasheet ................................................................................................................................ 33
Table 5 Data size measurement ............................................................................................................................................... 44
LIST OF FIGURES

Figure 1 Motorcycle crash .............................................................................................................1
Figure 2 Motorcycle registrations and motorcyclist fatalities, 1976-2010 ..........................................2
Figure 3 E-bike used in VTTI’s naturalistic study...........................................................................4
Figure 4 Schematic of DAS used in VTTI bicycle naturalistic study .............................................5
Figure 5 High level block diagram of the proposed system ...........................................................6
Figure 6 Helmet mounted camera ................................................................................................10
Figure 7 DAS - high level block diagram .....................................................................................15
Figure 8 Block diagram AVR CPU core architecture ......................................................................17
Figure 9 Arduino Mega 2560 ..........................................................................................................19
Figure 10 Clock distribution ..........................................................................................................20
Figure 11 PCB layouts and pin description .....................................................................................23
Figure 12 OV6620 CMOS image sensor block diagram .................................................................24
Figure 13 Data bus YUV output .....................................................................................................25
Figure 14 SD card interfacing with breakout board .........................................................................27
Figure 15 SPI bus ..........................................................................................................................27
Figure 16 SD card schematic .........................................................................................................27
Figure 17_1 GPS module schematic .............................................................................................28
Figure 17_2 GPS module ...............................................................................................................29
ABSTRACT

According to Motorcycle Industrial Council (MIC), in USA the number of owned motorcycle increased during last few years and most likely will keep increasing. However, the number of the deadly crash accidents associated with motorcycles is on the rise. Although MIC doesn’t explain why the accident rate has increased, the unprotected motorcyclist gear can be one of the reasons. The most recent National Highway Traffic Safety Administration (NHTSA) annual report stated that its data analyses are based on their experiences and the best judgment is not based on solid scientific experiment [3]. Thus, building a framework for the data acquisition about the motorcyclist environment is a first step towards decreasing motorcyclist crashes. There are a few naturalistic motorcycle studies reported in the literature. The naturalistic motorcycle study also identifies the behaviors and environmental crash hazards. The primary objective of this thesis work is to design a highly portable data logging embedded system for naturalistic motorcycle study with capability of collecting many types of data such as images, speed, acceleration, time, location, distance approximation, etc. This thesis work is the first phase (of three phases) of a naturalistic motorcycle study project. The second phase is to optimize system area, form factor, and power consumption. The third phase will be concerned with aggressive low power design and energy harvesting. The proposed embedded system design is based on an Arduino microcontroller. A whole suite of Arduino based prototype boards, sensor boards, support software, and user forum is available. The system is high portable with
capability to store up to eight (8) hours of text/image data during a one month study period. We have successfully designed and implemented the system and performed three trial runs. The data acquired has been validated and found to be accurate.
CHAPTER 1: INTRODUCTION

In 2009, Florida ranked second after California in traffic fatalities, and kept the same rank in 2011 after Texas, according to Motorcyclist Traffic Fatalities’ [2] preliminary study report. These high ranks imply serious issues related to economy and society. In addition, motorcyclists are 35 times more likely to be involved in a deadly crash than car drivers [1]. What makes the analysis of motorcycle fatality traffic data challenging is the complexity of the contributed factors (such as proximity to the adjacent vehicles, speed, acceleration, braking patterns, etc.) and lack of adequate data about them. Thus, there is a need for a data acquisition system that can aid in collecting such data that can be used to prevent these crashes. The analytic method based on the data collected should identify all relevant risk factors affecting motorcycle crashes.

Figure 1: Motorcycle crash [1]. (Public domain image)
1.1 System Approach

Undoubtedly, motorcyclist behavior is a significant factor in a motorcycle crash. However, motorcycle crash analyses based only on the motorcyclist’s behavior will not be comprehensive. The best approach is to get a full spectrum of all behaviors and environmental risk factors. The behavior constitutes all decisions made by the motorcyclist during riding course, and environmental risk factors include all the relevant factors while driving, which include for example road condition, timing, and location. The main goal is to conduct a naturalistic study of motorcyclists to identify relevant risk crash factors. Such a study will greatly help in enhancing motorcycle safety.
1.2 Methods

1.2.1 Naturalistic Study

As mentioned before, the method to be employed is the naturalistic study of motorcyclist behavior. In the proposed study, fifty participants will ride for a period of one month. Each participant’s motorcycle will be equipped with a data acquisition system designed and prototyped as part of this thesis work. For each participant, data for an average of eight (8) hours will be collected. Statistical and other analyses will then be carried out.

1.2.2 Data Acquisition

Designing a highly portable data logging embedded system for naturalistic motorcycle study will facilitate immediate and accurate identification of the most relevant risk factors. Different types of data will be collected such as motorcycle speed, acceleration, distance approximation, location, 3D inclination, and images. Each type can be categorized based on behavior or environmental factor(s). The pre-event data will be collected; it will be significantly useful in identifying risk factors. There are other factors such as gas prices, social ranking, and national economic status that can be indirectly inferred by analyzing data that will be collected. The four main requirements for data logging embedded system are:

- High Portability: Small portable device that can be easily mounted on any motorcycle.
- Large Storage Capability: It should be able to record images and texts for an average of eight (8) hours during one month period.
- Low Power Consumption: As it will be battery operated, it should be energy efficient and consume the least amount of power possible.
- Affordability: The final prototype’s cost cannot exceed $300.
1.3 Other Studies

Researchers have conducted naturalistic studies in the past [12]. However, the objectives and depth of each were different across the studies. Some studies tried to reinvestigate safety processing [13], others targeted specific cases [14], and the rest tried to identify the risk factors [12, 14, 15]. The naturalistic bicycling study by VTTI (Virginia Tech Transportation Institute) is one of the best examples. They chose HPT (Human Powered Transportation) as their subject of study as bicyclists are most of the times subject to aggressive actions by other engine powered commuters. The eco-friendly nature of this type of transportation (bicycle) adds environmental protection perspective besides traffic safety. In addition, data logging device design for this type of transportation required certain degree of optimization to achieve the portability and energy independency needed.

![E-bike used in VTTI's bicycle naturalistic study](image)

Figure 3: E-bike used in VTTI’s bicycle naturalistic study
1.4 Proposed Solution – An Overview

In order to accomplish a naturalistic study, unobtrusive intervention is quintessential during the course of data collection. To carry out the study, the auto functionality and high portability are the key design requirements.

1.4.1 Arduino Based System

We propose a system based on Arduino embedded microcontroller [18, 19]. The proposed system is composed of eight major sub-systems (Figure 5).

- **Battery**: Most motorcycles carry 12V battery, the actual supply voltage is between 12V to 14V, depending on whether the motorcycle engine is on or off.
- **DC-to-DC converter**: The 9V is the recommended voltage for the Arduino microcontroller. The DC-to-DC converter generates 9V from a source voltage of 12-14V.
- **SD Card**: The expansion board operating at 3.3V is used to interface the microcontroller with SD card. All data is stored on the SD card.
• GPS Module: The GPS unit is used to log time, date, location, and speed. The GPS unit follows the NMEA standard protocol, with an external antenna for better connection with the satellite.

• Gyroscope: The gyroscope unit supports SPI and \( \text{I}^2\text{C} \) communication protocols and is used to get 3D inclination.

• Camera: Two cameras with quart inch digital color output are used to record the surrounding environment for later analysis.

• Arduino Mega 2560: A microcontroller based on the ATmega 2560 with 16 MHz clock and 54 digital pins is used to process data and feed it to SD card memory.

• Sonar (EZ1): Is a range finder device with 7 digital pins; it can be interfaced using pulse width, analog voltage, or serial digital output format.

Figure 5: High level block diagram of the proposed system (data logging embedded system for Naturalistic Motorist Study)
1.4.2 System Specifications

The following are the system specifications that should be met:

1. Calculate the motorcycle speed
2. Get the time and the date
3. Calculate X, Y, and Z degrees of inclination
4. Get the exact location (GPS co-ordinates)
5. Get the acceleration
6. Get the proximity with vehicles in the front, back, and on the sides
7. Get the image and video information in the front and back of the motorcycle

1.4.3 Timing Constraint

There are no real-time processing requirements as the proposed system is a data logging system. However, there are internal timing constraints arising due to simultaneous acquisition of data from multiple sensors which need to be recorded in the memory. Such timing requirements are based on a 16 MHz Microcontroller clock.

1.4.4 Test Plan

- Accessibility: All files use the standard format “file.txt” or “file.csv,” which enables easy import and export of files.
- Accuracy and reliability: Several test runs will be conducted and compared with known results to confirm reliability and accuracy.
1.4.5 System Cost

The cost breakdown of the system must be provided. The cost needs to be optimized with various sub-system alternatives and their cost/performance trade-offs.

1.5 Summary

This chapter motivates the need for designing data logging embedded system for naturalistic motorcycle study. A brief overview of related work is presented to get insight on what has been done so far. An Arduino embedded micro-controller based solution has been outlined. The rest of the thesis is organized as follows: Chapter 2 presents three major related data logging embedded systems. Chapter 3 presents in detail the proposed solution. Chapter 4 reports the experimental results. Finally, Chapter 5 draws conclusions and outlines future work.
CHAPTER 2: RELATED WORK

There are several kinds of naturalistic studies; some focus on identifying patterns that can lead to a crash, others to get the broad picture of human behavior to reason why certain behaviors had been triggered. Moreover a number of studies use the cognition methods approach to study the correlation between a participant’s decision and the data collected. All of these studies concur that to get unbiased results, we must rely on naturalistic accurate data acquisition system. In this chapter, we will review three major studies reported in the literature.

2.1 Study 1 – Monash University [12, 21]

This study was conducted in 2009 by Monash University in Melbourne. The study was based on video recording only. Due to increased number of bicycle users in Australia, bicycling is considered the fourth popular sport activity with significant increase in the bicycle crashes [12]. However, this correlation doesn’t justify why the bicycle crashes occur, but it shows that there are deep problems that need to be identified and solved. Therefore, different strategies have been applied to solve these problems. The research study was conducted in three stages.

2.1.1 Stage 1

In this stage, sixteen sites were carefully chosen. A couple of cameras were installed in each intersection site, to record the activities on bicycle lines. The preliminary data that had been
collected during this stage were: gender, bicycle type, clothes type, bicyclist interaction, and direction of travel. It was an observation set up stage for the next one.

2.1.2 Stage 2

Before the first stage there were many questions raised, and some of them were answered in the end. In the second stage, one of the most important questions was: *did the cyclist use awareness such as head checks?* The film footage could not give a definite answer. As a consequence, the participants were asked to wear a helmet mounted camera (see Figure 6). This approach allowed recording of the actual cycling ride experience; that gave clear answers to direction of travel, head checks, and cyclist behavior.

![Helmet mounted camera](image)

Figure 6: Helmet mounted camera

2.1.3 Stage 3

As always there were some questions that remained unanswered. This could be due to the question itself or the limitations of the preview methods that had been used. Thus, an online survey was conducted to answer some of these questions. The survey allowed two things to be achieved. First, a large segment of the cyclists had been interviewed. Second, questions such as
cyclist attitude, behavior, age, experience, and knowledge were asked. These types of questions can’t be answered using the preview methods.

2.2 Study 2 – VTTI and Monash University [23]

Monash University used the first study as a framework for more deep and complete study by co-operating with Virginia Tech Transportation Institute (VTTI). As result of combined efforts, in 2012 the two Universities developed a data acquisition system (DAS) for naturalistic bicycling study. The system allowed collecting different types of data such as, location, distance approximation, speed, acceleration, and video recording. The scope of the study was naturalistic-based. The data that had been collected was used to determine risk factors of the bicycle crashes. The size of DAS and power consumption were key factors for this type of HPT (Human Powered Transportation) machines.

2.3 Study 3 – NHTSA [24]

The pilot study, which was presented by NHTSA, was a well-rounded study. The study was meant for motorcycle naturalistic study. The DAS had the ability to collect different types of data. Several cameras were used, which allowed capturing different view angles. The most interesting is the software tool that has been used. It was based on MATLAB and it could synchronize multiple data events together. Thus, all critical events were easy identified. Image processing was used to get eye and line tracking. However, the system had a large form factor.
2.4 Summary

The three studies showed that DAS plays a key role in any naturalistic study. However, the researchers did not discuss the drawbacks of their systems. Furthermore, details of their DAS were not provided.
CHAPTER 3: PROPOSED EMBEDDED SYSTEM

The idea behind the proposed embedded system was to build a small and reliable data logging system. Other factors, such as low power and high performance are not primary factors during the first phase.

3.1 DAS Design for Naturalistic Study

Any hardware design has tradeoffs between functionality, cost, performance, and power. Therefore, to get the best tradeoff possible that meets the goals of the study, and flexible to adopt new ideas, we have decided to employ an evolutionary design approach. Thus, the DAS design will be performed in three phases. This thesis reports the first phase results.

1. The first phase was to rapidly prototype a data logging embedded system that fulfilled all functionality requirements. In this phase, the functionality is briefly summarized as follows: all types of data must be collected, data logging duration must be fulfilled, and data error range must be low. In addition to the functionality requirements, power saving technique will be attempted. Even though, power consumption is not an issue during this phase (most of the motorcycles are equipped with a 12V battery which can be used as power source) it is better to tackle this issue as early as possible since the same system is expected to be employed in future e-bike naturalistic studies. The first step to save area
and power is with the appropriate selection of system components. Thus, the component size and power consumption were the main selection criteria.

2. The second phase is to optimize system area and power consumption. The first phase findings will drive the optimization decisions.

3. The third phase will be concerned with aggressive low power design and energy harvesting (for example, solar energy based sourcing) techniques. During this phase, we will focus on power source efficiency and duration. However, the solution must be practical and cost efficient.

3.1.1 Low Power Consumption

Power savings are obtained based on the following three approaches. First, by using power consumption as the selection criteria during the component selection. Second, by reducing the number of pin connections between the microcontroller and devices as it used I²C (Inter-Integrated Circuit) protocol rather than SPI (Serial Peripheral Interface). Third, by putting a device into sleep mode (low power state) as needed i.e., during system idle states.

3.1.2 Form Factor

The system devices were carefully chosen and form factor was one of the main criteria. In addition using PCB board on top of Arduino leads to compact rigid system devices significantly reducing the system form factor.
3.1.3 Data Logging Duration

The 32 GB SD card was used to get preliminary estimates of the data logging duration. The goal is to get eight (8) hours of image and text record data over a period of one month. In the worst case scenario 128 GB SD will be used.

The low power and reduced form factor will allow easy extension to e-bike using combined series of 1.5V DC batteries or one 9V battery.

3.2 Conceptual Diagram

The system is composed of eight sub-systems. The combined characteristics determine the overall system performance. Therefore, the following sections will describe each device characteristics and performance.

![Figure 7: DAS - high level block diagram](image_url)
3.3 Arduino Based System

The main device which controls and synchronizes tasks between the components is the ATmega2560 microcontroller. This project is an Arduino based design. All components are Arduino IDE compatible. The following is a detailed description of the Arduino ATmega 2560 board components.

3.3.1 Arduino Processor

The microcontroller is designed for the following two highly desirable properties: the simplicity and the long time reliability. To achieve that, ATmega2560 is based on Harvard Architecture. On contrary to von Neumann architecture, where program instructions and data lie in the same memory block; in Harvard architecture they are completely separated. There is of course a tradeoff: most of the microcontrollers perform simple task and do not require a lot of data exchange but require good reliability; this results in a tradeoff of simplicity and reliability versus flexibility. When it comes to power consumption and performance, ATmega2560 has a good balance. It relies heavily on parallelism with pipelining. It has 32 x 8 bit general purpose register linked directly to Arithmetic Logic Unit; two independent registers can be accessed in one clock instruction. Thus, performance and power consumption are significantly improved (Figure 8).
Figure 8: Block diagram AVR CPU core architecture
The Arduino mega2560 board was used as the base platform for prototyping the DAS system. All microcontroller pins can be accessed through board pins. In the following section, some of the Arduino mega2560 board key features will be explained.

The Arduino mega2560 board has more digital pins than any other Arduino board. It has 54 digital I/O digital pins (Figure 9). These pins are classified as following:

- **Power pins**

  There are 5 pins that can be used to supply 3.3V or 5V to peripherals. There is also a VIN pin that can be used to power the Arduino board. We can reset the board by simply connecting the reset pin to the ground, of course, after we power it up.

- **Communication pins**

  - Serial communication: there are 4 serial lines RX and TX of communication, and correspond to 0-1 and 14-19 pins on the board.
  
  - Serial peripheral interface (SPI) communication pins are: MISO, MOSI, SCK, and SS (50-53 pins).
  
  - Two wire interface (TWI) communication are: SDA (pin 20) and SCL (pin 21).
  
  - Pulse width modulation outputs: it generates 8-bit output PWM; the PWM pins are 2-13 and 44-46.
  
- **The rest are digital pins that can be programmed either as input or output pins.**

- **Memory**

  The Arduino Mega 2560 has three types of memory, namely, the Flash, the EEPROM, and the SRAM. The Flash memory is 256KB in size and is divided into two sections one for boot loader and another for application program. The SRAM with 8 KB and EEPROM with 4KB can be used as external storages.
Clock systems

There is a timing constraint and synchronization requirements between Arduino’s tasks. The Arduino mega board is equipped with clock systems that deal with these challenges. The ATmega2560 has five clock systems (Figure 10). The CPU clock is used to access the general purpose registers.

The I/O clock is generally used for I/O modules such as SPI, USART, and external interrupt. The Flash clock is used to manage the Flash interface activities. It is often triggered concurrently with the CPU clock. The asynchronous timer clock is for time counter to be clocked externally. Finally, the ADC clock is used to reduce circuit noise by stopping both CPU and I/O clocks.
Figure 10: Clock distribution
3.3.2 Camera

The camera C3088 is used to capture video and images to the system. There are many reasons (functionality and feature related) behind choosing this type of camera. This section elaborates on these reasons.

3.3.2.1 Camera Functionality

The C3088 camera uses an OV6620 CMOS image sensor (Figure 12). The C3088 is color image camera with 352 x 292 resolution. It has digital video port that can be configured at 8/16 bit data stream. The camera is I²C prototype; it uses two wire interface (TWI) to communicate with peripherals. All camera settings such resolution, windowing, frame rate, gain, exposure, and others can be done though I²C protocol. Even though, camera uses I²C protocol to communicate with the microcontroller; the camera data sheet defines serial camera control bus (SCCB) as the main communication protocol. If we dig deep, the SCCB protocol is very similar to I²C. There are a few differences, such as SCCB protocol uses the third line SCCB_E to initiate the start and the stop bit; moreover, SCCB protocol requires one master and at least one slave. Some devices are shipped with SCCB_E set low by default, which makes I²C the main protocol of communication. Back to the camera setting, the C3088 uses Id address 0xC0 for writing and 0xC1 for reading. The I²C protocol uses a seven bit address format, the next bit after the address bit indicating 0 for writing and 1 for reading. More details on complete writing and reading protocol are as follows.
3.3.2.2 Writing

- Start communication
- Call device used 0xC0 address
- Call register address you like to write to it
- Write data
- End communication

3.3.2.3 Reading

- Start communication
- Call device used 0xC1 address
- Call register address you like to read from
- Read data
- End communication

All camera settings are done using the writing protocol above. Each setting requires writing certain value to a specific register. The special registers and their values will be found in the camera datasheet. The camera data bus line can be configured to read 8 or 16 data bits through Y and UV bus. To get valid data there are certain timing restrictions that must be followed, for example, enable 8 bit data line through Y port and the PCLK must be twice faster (Figure 13). The data line VSYNC, HREF, PCLK, and data line Y/UV are inputs from the camera to the microcontroller. The C3088 camera has 32 pins their description is provided in Figure 11.
<table>
<thead>
<tr>
<th>Pin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1~8</td>
<td>Y0-Y7 Digital output Y Bus.</td>
</tr>
<tr>
<td>9</td>
<td>PWDN Power down mode</td>
</tr>
<tr>
<td>10</td>
<td>RST Reset</td>
</tr>
<tr>
<td>11</td>
<td>SDA I'C Serial data</td>
</tr>
<tr>
<td>12</td>
<td>FODD Odd Field flag</td>
</tr>
<tr>
<td>13</td>
<td>SCL I'C Serial clock input</td>
</tr>
<tr>
<td>14</td>
<td>HREF Horizontal window reference output</td>
</tr>
<tr>
<td>15</td>
<td>AGND Analog Ground</td>
</tr>
<tr>
<td>16</td>
<td>VSYN Vertical Sync output</td>
</tr>
<tr>
<td>17</td>
<td>AGND Analog Ground</td>
</tr>
<tr>
<td>18</td>
<td>PCLK Pixel clock output</td>
</tr>
<tr>
<td>19</td>
<td>EXCLK External Clock input (remove crystal)</td>
</tr>
<tr>
<td>20</td>
<td>VCC Power Supply 5VDC</td>
</tr>
<tr>
<td>21</td>
<td>AGND Analog Ground</td>
</tr>
<tr>
<td>22</td>
<td>VCC Power Supply 5VDC</td>
</tr>
<tr>
<td>23-30</td>
<td>UV0-UV7 Digital output UV bus.</td>
</tr>
<tr>
<td>31</td>
<td>GND Common ground</td>
</tr>
<tr>
<td>32</td>
<td>VTO Video Analog Output (75Ω monochrome)</td>
</tr>
</tbody>
</table>

Pins that not been used are:
- 12
- 23-30
- 19
- 32

Figure 11: PCB layouts and pin description
Figure 12: OV6620 CMOS image sensor block diagram
A 32 GB SD card was used as storage memory. The SD card needs a slot for interfacing, and software to run it. The Arduino IDE provides an SDK library that can be readily included in our software. The SD card supports only SPI protocol (Figures 14, 15, and 16) and operates at
3.3V. A significant care must be used, when connecting SD card to Arduino board. Most of Arduino pins are 5V which can burn SD card. The SD card bus has nine pins (Figure 16). Detailed description of each pin is presented in Table 1. The 32 GB is the highest capacity storage we were able to use. Even though, we are looking for the high storage possible, we are limited by the file format of Fat 16/32. The 64 GB and above are supported by EXFAT format.

Table 1: SD card pin description

<table>
<thead>
<tr>
<th>Pin Number</th>
<th>Pin Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CS</td>
<td>Chip Select</td>
</tr>
<tr>
<td>2</td>
<td>DI</td>
<td>Data In</td>
</tr>
<tr>
<td>3</td>
<td>VSS</td>
<td>Ground</td>
</tr>
<tr>
<td>4</td>
<td>VDD</td>
<td>Voltage</td>
</tr>
<tr>
<td>5</td>
<td>SCLK</td>
<td>Clock</td>
</tr>
<tr>
<td>6</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>7</td>
<td>DO</td>
<td>Data Out</td>
</tr>
<tr>
<td>8</td>
<td>REV</td>
<td>Reserved</td>
</tr>
<tr>
<td>9</td>
<td>REV</td>
<td>Reserved</td>
</tr>
</tbody>
</table>
Figure 14: SD card interfacing with breakout board

Figure 15: SPI bus

Figure 16: SD card schematic
3.3.4 GPS Module

It is a small device that can provide longitude, latitude, speed, and UTC time data. It is compatible with the Arduino environment. To get it working, we download TinyGPS folder to the Arduino Library. The GPS needs satellite connection to get required data; this takes approximately 10 to 15 minutes. Using GPS antenna and covering it with some closure can significantly reduce the initialization time. It uses NMEA 0183 standard communication protocol. The NMEA is an acronym for National Marine Electronics Association. For more information on how NMEA communication works, the reader is referred to [25]. The GPS device has twelve pins, eight for communication and four for power supplies. These eight pins fit nicely in Arduino mega board communication pins 14-21. The interface is straightforward, however, the Arduino’s last two pins 20 and 21 which correspond to SDA and SCL must be carefully mapped to the corresponding GPS pins TX and RX (Figures 17 and 18). The GPS device requires 5V DC supply.

Figure 17_1: GPS module schematic
Figure 17_2: GPS module

Figure 18: GPS module and Arduino interfacing
3.3.5 Gyroscope

Compared to other gyroscope devices, the L3G4200D Gyro has many advantages. It can operate using \( \text{I}^2\text{C} \) or SPI protocol. In addition, it can also output temperature. Even though, L3G4200D can be configured using SPI protocol, the default communication setting is \( \text{I}^2\text{C} \) protocol (Figure 19). The L3G4200D has 32 FIFO buffers that can be set in different mode; this gives the microcontroller timing flexibility. The advantage from using buffers is that the microcontroller doesn’t need to pull data from Gyro. It can be busy doing other stuff or simply be idle, and pull data when needed. The L3G4200D Gyro can be set to five different FIFO modes.

![Figure 19: \( \text{I}^2\text{C} \) interfacing](image)

3.3.6 Sonar

One of the project requirements is to approximate the distance between the motorcycle and other vehicles in the vicinity of the device. The LV-Max Sonar EZ1 is the device that has been used to accomplish this task. The sonar range of detection is 0 to 6.45 meter; it can be
interfaced using pulse width, analog voltage, or serial digital output format (Figure 20). The Sonar EZ1 was chosen due to its:

- Small size
- Low power consumption
- Output format flexibility
- Direct reading

This type of sonar is very sensitive to motion. Therefore, distance accuracy must be considered during the measurements. To get the best accuracy possible:

- The filter algorithm has been implemented during the measurements.
- The output format with the good accuracy been chosen.

The test result shows that the pulse width format output is the most accurate one.
3.4 System Cost

Table 2: System cost breakdown

<table>
<thead>
<tr>
<th>Part</th>
<th>Price($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arduino Mega 2560</td>
<td>17.69</td>
</tr>
<tr>
<td>GPS</td>
<td>117.00</td>
</tr>
<tr>
<td>Camera C3380</td>
<td>47.88</td>
</tr>
<tr>
<td>SD Card</td>
<td>10.12</td>
</tr>
<tr>
<td>SD card Slot</td>
<td>23.00</td>
</tr>
<tr>
<td>Gyroscope</td>
<td>29.00</td>
</tr>
<tr>
<td>Sonar EZ1</td>
<td>24.00</td>
</tr>
<tr>
<td>Regulator</td>
<td>10.00</td>
</tr>
<tr>
<td><strong>Total Price</strong></td>
<td><strong>278.69</strong></td>
</tr>
</tbody>
</table>

3.5 Power Analysis and Tradeoff

In Table 3, the power consumption results for active and idle states are reported. Two experiments were conducted to measure the power consumption. In the first experiment, we used a multimeter to measure the currents. They were seven measurements for each state. The system is powered by 9V battery. In Table 4, we used datasheet to calculate power consumption for each component. As discussed earlier, although the motorcycle has sufficient power source, our future plans to re-use the system for e-bike requires us to be power-aware.

Table 3: Power consumption experiment

<table>
<thead>
<tr>
<th>Active State</th>
<th>Idle State</th>
</tr>
</thead>
<tbody>
<tr>
<td>592mA</td>
<td>120mA</td>
</tr>
<tr>
<td>640mA</td>
<td>136mA</td>
</tr>
<tr>
<td>720mA</td>
<td>146mA</td>
</tr>
<tr>
<td>680mA</td>
<td>147mA</td>
</tr>
<tr>
<td>685mA</td>
<td>140mA</td>
</tr>
</tbody>
</table>
Nowadays, the trend is to build fast and low cost embedded device. The fact is, Moore law supports these goals, and is not hard to achieve them. However, a device with high performance and low power is challenging. The power analysis tradeoff techniques, such as loop unrolling and code optimization are very well known solutions and can be applied in our case.

### 3.6 Summary

In this chapter, a detailed description of the major sub-systems of the proposed embedded system was provided. Most of the device functionality fits well in the system. However, when it
comes to the cost and power, there is still room/need for optimization, which must be made in the succeeding phase.
CHAPTER 4: EXPERIMENTAL RESULTS

After assembling all the system components together and checking them using computer serial monitor, several experiments were performed to validate the system. For debugging purposes, the system was assembled on a breadboard (Figure 21). This provides two advantages:

- Flexibility in adding/removing the components
- Simplicity in accessing and debugging each component

However, the drawback is that it is difficult to conduct field experiments due to the large form factor and weather elements. Therefore, the system was encased in a rugged plastic box. The breadboard was used for system validation purposes. For the deployment during the naturalistic study, PCB implementation will be used. In the following sections, the experiments and the data collected are described.

Figure 21: Breadboard implementation of the data logging embedded system
4.1 System Validation Experiments

Before we started collecting the data, we need to make sure that all devices work properly, both individually and together. Therefore, extensive testing was conducted. There were two testing phases before the actual data collection:

- Test phase one - We had many small runs for each type of data that had to be collected. These first runs were just for small distance and short period. They provided essential checks of the devices’ response and code correctness. They also aided in code debugging and device settings (Figure 25). During this phase, the data sheets of the devices were heavily consulted.

- Test phase two - These runs were performed in medium distance, to check all system device functionality and accuracy. Here some tests were used repetitively to check the accuracy. Some hardware and software adjustment was needed to get accurate results.

After initial testing, the experiments that involved longer rides were performed. For each type of data, three long runs were conducted. All data presented below was collected during these long runs.

4.2 GPS Data

We used GPS Google map annotated path info to get GPS. There were three trial runs for data collection.

- During the first run GPS data accuracy was tested. A well-known and relatively small area was chosen to conduct this experiment. The results were very accurate (Figure 22).
During second run GPS signal stability was tested. The experiment was conducted using interstate I-75 from Sarasota in South to Ellenton in North. Both cities are in Florida State. During the ride the maximum speed reached was approximately 70 miles per hour. The goal was to check GPS stability boundary. Due to safety concerns, the experiment was conducted using car (Figure 23).

During the third run data stability was tested. Situations where weather conditions and traffic were not favorable were chosen on purpose (Figure 24).
On all GPS data collection runs, the results were completely accurate. There is a waiting time to get GPS and satellite connection, which can be from 2 to 15 minutes.

4.3 Image Data

The biggest hurdle during data collection process was getting clear images. The images were being streamed directly from the camera data to the SD card, which caused a lot of data corruption. To overcome this problem, first the camera clock was reduced to the lowest speed of 69 KHz. Second, an interrupt handler was introduced when images were taken. Third, before taking each image, setting validation occurred. After many trials (Figure 25) and tuning, clear images were obtained.
Figure 25: Detecting camera’s signals using oscilloscope.

The BMP file format was used to convert camera pixels to color. There were two sets of image data collected. For the first set, the speed was very low (20 mph) (Figure 26).

(A)                                                             (B)

Figure 26: Images (A) and (B) taken at low speed

For the second set, the speed was 50 miles per hour and we chose low traffic conditions for safety reasons (Figure 27).
In the second set, the images were blurred due to vibration. The image collection design will be revisited in the second phase of the project. Direct data image streaming is not a good approach to get reliable image data. Arduino board extra memory, as storage buffer, has to be considered.

### 4.4 Distance Approximation Data

The distance data from Sonar was collected and reported in the graphs (Figures 28 and 29). The furthest distance approximation that was detected was 6.5 meters (~22 feet). There is 1.5 cm accuracy on most distance data that has been collected. Two sets of distance data have been presented - one for high traffic area (Figure 28) and other for low traffic area (Figure 29).
4.5 Acceleration and Speed Data

The GPS device allowed us to obtain both speed and acceleration simultaneously. Speed and acceleration data were collected, using different speed regimes (Figures 30 and 31). Driving at very low speed has been recorded as well (Figures 32 and 33).
Figure 31: Acceleration corresponds to different speed regimes

Figure 32: Driving at low speed

Figure 33: Acceleration corresponds to low speed
4.6 Gyroscope Data

The device was very responsive during data collection. Despite using the I²C protocols for interfacing; the reading was fast due to Gyroscope buffering capability. The three directions X, Y, and Z of the angular rotation were recorded and planted on graph (Figure 34). More information about the Gyroscope device and its buffering modes are presented in Chapter 3.

![Figure 34: Gyroscope data](image)

4.7 Data Size

Although twenty data collection trials were conducted, only nine were successful. Most of data failures were due to the camera setting; they were couple of failures specific to the first experiments, which were related to wiring and data formatting. To fix data failures the camera clock was reduced, the interrupt was disabled during picture taking, a small delay before and after camera setting was added, and the wires were hardened. All successful data trials are presented in Table 5.
Table 5: Data size measurement

<table>
<thead>
<tr>
<th>Trails</th>
<th>Type of Data</th>
<th>Success</th>
<th>Fail</th>
<th>Data Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Text</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>Text</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>Text</td>
<td>X</td>
<td>-</td>
<td>61.5KB</td>
</tr>
<tr>
<td>4</td>
<td>Text</td>
<td>X</td>
<td>-</td>
<td>154KB</td>
</tr>
<tr>
<td>5</td>
<td>Text</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>Text</td>
<td>X</td>
<td>-</td>
<td>162.8KB</td>
</tr>
<tr>
<td>7</td>
<td>Image</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>Image</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>9</td>
<td>Image</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>Image</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>11</td>
<td>Image</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>12</td>
<td>Image</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>13</td>
<td>Image</td>
<td>X</td>
<td>-</td>
<td>162KB</td>
</tr>
<tr>
<td>14</td>
<td>Text/Image</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>15</td>
<td>Text/Image</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>16</td>
<td>Text/Image</td>
<td>X</td>
<td>-</td>
<td>525KB</td>
</tr>
<tr>
<td>17</td>
<td>Text/Image</td>
<td>X</td>
<td>-</td>
<td>625KB</td>
</tr>
<tr>
<td>18</td>
<td>Text/Image</td>
<td>X</td>
<td>-</td>
<td>800KB</td>
</tr>
<tr>
<td>19</td>
<td>Text/Image</td>
<td>X</td>
<td>-</td>
<td>1105KB</td>
</tr>
<tr>
<td>20</td>
<td>Text/Image</td>
<td>X</td>
<td>-</td>
<td>900KB</td>
</tr>
<tr>
<td>Total</td>
<td>20</td>
<td>9</td>
<td>11</td>
<td>4495.3KB</td>
</tr>
</tbody>
</table>
It can be concluded that a prototype system that meets the initial project requirements was successfully designed and tested. They were many hurdles during data collection. However, the majority of these hurdles were related to direct data image streaming. Many steps were adopted to overcome these hurdles, such as camera clock speed reduction, delay insertion, and interrupt disabling. The down side was that the system performance was negatively affected. The best approach is using Arduino external memory during data image collection as buffering zone and transfers it later to the SD card. This approach will smoothen data transfer and will increase system performance as well. The form factor and power analysis results were very encouraging; therefore porting from motorcycle to bicycle will be feasible.

The future work includes development of a PCB based interconnect that compactly interfaces the sensors and the Arduino board. Further, solar cell based energy harvesting will be attempted to further facilitate porting of the logging system to e-bikes.
REFERENCES


