Microcontroller Based LED/Audio Beacon

Final Report

Team Disco:

Kyle McClary

Josh Land

Kevin Ufferman

Advisors:

Dr John Wu

Bruce Rahn

Stephen Mascarella

Steve Overmeyer

Dr. Joseph Slater

Table of Contents:

Project Title: Project Disco!

Group Leader: Kyle McClary.

Group Members: Kevin Ufferman, Joshua Land

Advisor: Dr. John Wu

Abstract

We have created a visible/audible beacon for use on the WSU High Altitude Balloon (HAB) project. It has been a problem in the past where the team experienced difficulty in locating the payload either in a night-time setting, or in areas with dense vegetation. This new capability will provide the recovery team with enhanced visual tracking of the balloon as it falls from the sky, as well as provide valuable audio feedback while searching for the device on the ground. This will reduce the search time required to locate and retrieve the payload.

Our design was implemented through the use of an Arduino Microcontroller, 1W LEDs, a 100dB piezo-electric speaker, and various driving circuitry. The circuit board was designed using CadSoft Eagle 5.11.0. After thorough testing our device was successfully utilized on Wright State University High Altitude Balloon Team's Launch 20.

INTRODUCTION

Purpose

The purpose of this document is to review our beacon design as a part of the High Altitude Balloon.

Project Scope

Our project is to design a combined LED/Audio beacon for payload retrieval as a part of the High Altitude Balloon team. The given design specifications were to have the payload blink at night to satisfy FAA regulations, and to have an audio beacon mounted in the payload to increase the chances that the package is seen or heard by the public, or by the team while foxhunting. The beacon is meant to serve as a standalone last resort for retrieval of the package since if all other systems fail this standalone system will still function and will attract the attention of anyone in the area where the payload lands. Our design aims to be as versatile, light, and efficient as possible.

DISCUSSION

Design Specifications

- **Blinking High Intensity LEDs (x4)**
  - Satisfy FAA regulations in case of a failed package cut down, and package descent at night.
    - 1 Hz or lower frequency, white strobe
  - Low power in relation to other sources
    - 1 watt (3 V at 350 mA)
  - Lightweight
    - ~1g each
  - Reliable
    - Burns for ~10,000 Hours
    - Solid State
  - High Visibility
    - Visible up to 1 mile

- **Piezo-electric Buzzer**
    - Highly Audible
        - 100dBa @ 3500 Hz resonant frequency
    - Low Power
        - Beeping sequence – not constantly activated
        - 27 mW
    - Lightweight
        - 10g

Design Implementation

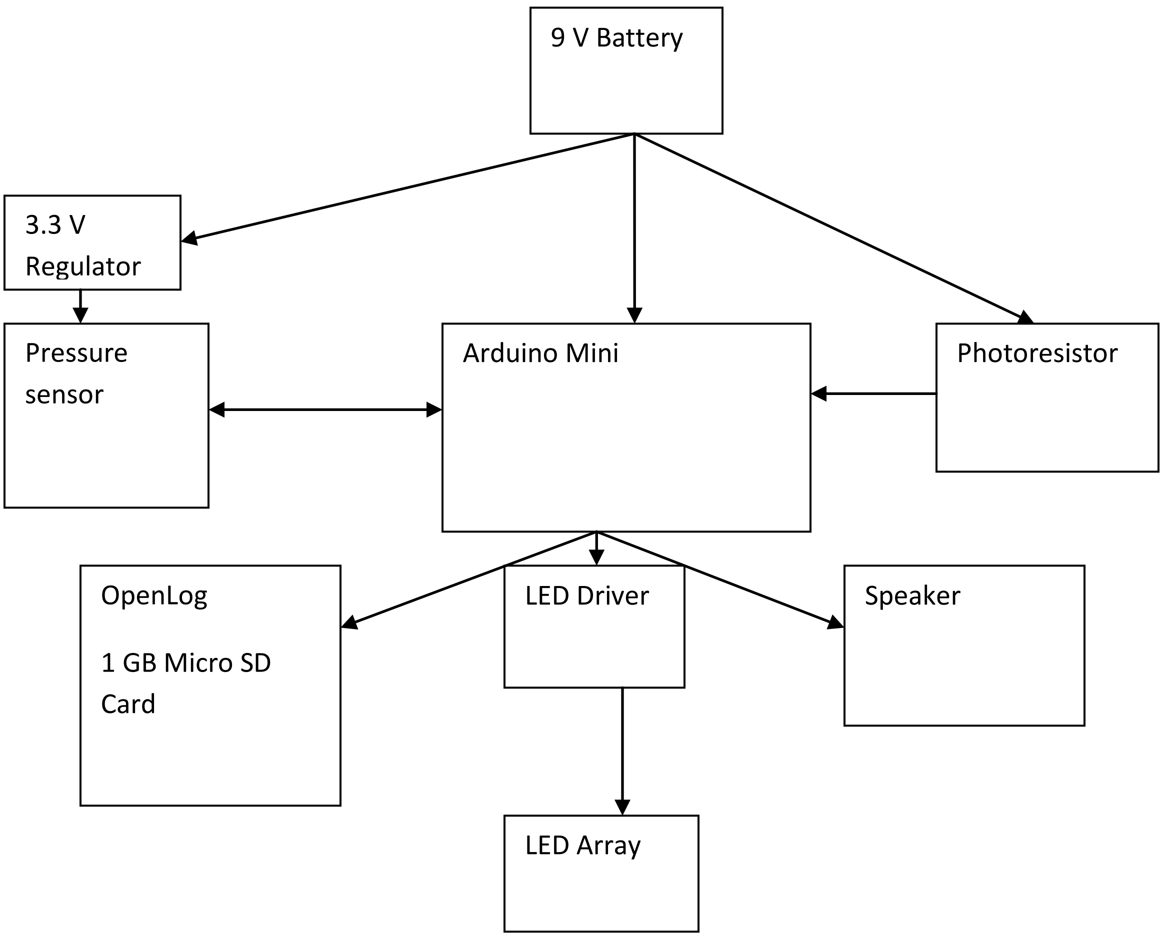


Figure 1. Block Diagram of System

For our design we have chosen to drive the LEDs and buzzer utilizing an altitude triggered microcontroller. We have chosen the Arduino Mini Light microcontroller for its easy programming language, open source development, and large support community. The Arduino Mini Light is based on the ATMega 168 microcontroller chip with pin-outs to a 1.2x0.8in circuit board. We chose the light edition of this microcontroller since it does not have pins soldered to the board to save on a little weight and so that we could make connections where we need them and are not forced to connect every pin on the board. In order to upload code, or sketches, to the board we need to purchase an FTDI USB to serial RS232 adapter specific to the Arduino Mini. To run the circuit we will use a primary lithium caseless 9V battery. We chose this battery for its weight reduction in being caseless and for its proven usage onboard the High Altitude Balloon. We will also use a Texas Instruments LED driver and the accompanying circuit layout to step up the 9V from the battery to ~14V and provide the 350mA to the LEDs that we need to drive all 4 LEDs in series. We plan on incorporating a high power by-pass resistor on each LED so that in case of a malfunction the power to the other LEDs will not be disabled. Our design will also incorporate a lightweight barometric pressure sensor in order to estimate the altitude and maintain a fully standalone system. Altitude data could have been transmitted from the onboard GPS in the balloon payload but if the GPS system were to fail the beacon system would fail. We will also incorporate a photo-resistor in our system to meet the requirement of blinking only at night to save power. After speaking to our associate Mechanical Engineering team we plan on adding a ni-chrome wire cut down system for parachute release on their ballute.

The design will function by having a code uploaded to the microcontroller which reads from the photo-resistor and pressure sensor. The LEDs will begin to blink at 1 Hz when the photo-resistor has a resistance of approximately 10kΩ, which will occur when it is dark outside, and when the pressure sensor reads that the balloon is inside FAA airspace, under 65,000 feet. Between 65,000 feet and 100,000 feet the beacon will go into standby mode and monitor the altitude less often. Once the balloon reaches its peak altitude, the package will enter freefall and the system will go fully active again. The parachute release system will cut through a nylon string when the package has gone through freefall for a pre-determined distance. Once the parachute is deployed and the package re-enters FAA airspace the system will go into low power mode and have no output unless the conditions for blinking are met. Once the package gets below 1,000 feet the package will start beeping, on an interval that is to be determined, to increase its chances of being seen by the public. Once the package has landed the beacon will keep beeping and if the package has landed in a dark area the lights will begin to flash. If a member of the public comes upon the package while it is active there will be a well marked power switch to stop the beeping, and the person will hopefully contact the retrieval team.

Testing

Following prototype completion, the microcontroller was programmed. This led to the first phase of testing through program development. The microcontroller was programmed to output a divided digital clock signal to the pressure sensor chip as per the chip requirements of a clock within 30-35kHz. This was accomplished using the ATMega 168 clock register and dividing the 16MHz clock by 64, yielding a clock of 31.125kHz. This clock was believed, at the time, to be sufficient but has now been proven to have too much variance and therefore cannot be utilized.

The microcontroller was then programmed to utilize SPI communication to exchange information with the pressure sensor. The SPI clock was initially left to the default setting, and communication was attempted utilizing a serial monitor through the programming utility. When given the proper codes, the pressure sensor would only ever output strings of 0xFF or 0x00. Through experimentation and investigation the default SPI clock signal was assumed to be too high and was changed to the lowest speed possible: 125kHz. Communication was then attempted again resulting in the same chip output.

After a few nights of testing and checking code, the attempt was made at adding an extra 0x00 at the end of data transmission to the sensor. This solved the 0xFF and 0x00 output problem and allowed the proper calibration constant calls to be made. This change also resulted in pressure and temperature readings being output. The output pressure and temperature readings were found to be in error and the code was checked and re-checked for accuracy and variable length. After resetting all utilized variables to short signed integers (16bit), the pressure and temperature readings were closer to what was expected with a +50C error in the temperature reading. To test and see if the temperature offset was constant, the device was programmed to log data every 30 seconds to the built in data logger, and thrown in the freezer. The device was left in the freezer at a temperature of -20C for 15 minutes. The resulting readings were not as expected. After a few minutes in the freezer (time varied between 2 and 3 minutes) the pressure and temperature dropped to -500 millibars and -170 celcius, and would change drastically. Errors were also noted in the reading of the sensor calibration constants.

After consultation with Bruce Rahn, the clock is thought to have changed speed as the temperature decreased to something outside of the range allowable for proper sensor readings. The clock generation was then changed to utilize the Tone library for Arduino, and the clock was set to 35kHz. At this clock speed the chip functionality was as expected until the chip returned to being cold where the same errors occurred. Utilization of the sensor chip without an external clock crystal was deemed unlikely and was not pursued farther. Errors due to clock skew can be found in Appendix I.

In an attempt to meet the altitude triggering requirement, a Garmin 15L GPS unit was hooked up to the serial pins on the microcontroller and reading was attempted. After reading some odd ASCII characters more research was done into the form of serial communication used by the GPS. It turns out that the Garmin 15L utilizes RS-232 whereas the Arduino uses TTL. Since the two forms of communication are incompatible without a hardware converter, a hardware converter was ordered. Testing data can be found in Appendix II.

Before the RS-232 to TTL converter could arrive, a launch was proposed and the team decided to put up our device. The device was re-wired to have wire lengths meeting the requirements for team Chutes and Giggles ballute, and was installed in the ballute. The launch took place on May 31, 2011. Since utilization of altitude data was impossible on the flight, the device was set to poll the photoresistor every 30 seconds and blink if dark. The audio beacon was set to start after 10,000 seconds (~3hrs). This amount of time was chosen as it would be sitting on the ground turned on prior to launch for at least half an hour, in flight for an assumed hour or more, and after descending it may have been an hour before rescuers got within range. After the successful launch and descent, the ballute was found blinking just off Trebein Road in Beavercreek Ohio. The LED beacon was fully functional since the ballute landed with the photosensor in the grass. Had enough time passed, the speaker would have started screaming but since the ballute had a premature descent not enough time had elapsed.

CONCLUSION

Summary

The LED/Audio beacon is not only a convenience factor on the payload of the High Altitude Balloon, it is a necessary component as a fail-safe for package retrieval, parachute deployment, and for meeting FAA regulations. The system is composed of an Arduino Mini Microcontroller, a barometric pressure sensor, 4 high intensity LEDs, and a piezo-electric buzzer along with lower level circuitry to drive the system. After thorough testing, reading the altitude utilizing the microcontroller at this time was deemed unfeasible. The team believes that it would be better visited as a summer project and will be considered after graduation. Implementation of a crystal for clock management of the pressure sensor will be attempted first, followed by implementation of the RS-232 to TTL converter and GPS library development.

Contact Info

Kyle McClary – kyle.mcclary@gmail.com

Joshua Land – landjo77@gmail.com

Kevin Ufferman – ufferman.2@wright.edu

Appendix I. Pressure Sensor Testing

Normal Operation at Room Temperature - 31.125 kHz Clock:

Time(s): 2
Avg Pressure Temp
11112
698
Light (%): 1023
Dark, Flashing Lights

Time(s): 3
Avg Pressure Temp
11113
698
Light (%): 1023
Dark, Flashing Lights

Time(s): 3
Avg Pressure Temp
11113
699

Operation at -20C - 31.125 kHz clock

Time(s): 105
Avg Pressure Temp
4257
-1170
Light (%): 1023
Dark, Flashing Lights

Time(s): 105
Avg Pressure Temp
7073
152
Light (%): 1023
Dark, Flashing Lights

Time(s): 106
Avg Pressure Temp
-1080
-1358
Light (%): 1023
Dark, Flashing Lights

Time(s): 107
Avg Pressure Temp
5614
-110
Light (%): 1023

Dark, Flashing Lights

Time(s): 107
Avg Pressure Temp
3142
-288
Light (%): 1023
Dark, Flashing Lights

Time(s): 108
Avg Pressure Temp
6898
-310
Light (%): 1023
Dark, Flashing Lights

Time(s): 109
Avg Pressure Temp
3519
-270

Normal Operation at Room Temperature - 35 kHz Clock

Time(s): 0
Avg Pressure Temp
11185
654
Light (%): 1023

Time(s): 1
Avg Pressure Temp
11184
656
Light (%): 1023

Time(s): 1
Avg Pressure Temp
11183
657
Light (%): 1023

Time(s): 2
Avg Pressure Temp
11185
656
Light (%): 1023

Time(s): 3
Avg Pressure Temp
11184
657

Operation at -20C - 31.125 kHz clock

Time(s): 25
Avg Pressure Temp
8901
666
Light (%): 1023

Time(s): 26
Avg Pressure Temp
25688
-32731
Light (%): 1023

Time(s): 27
Avg Pressure Temp
-15619
-591
Light (%): 1023

Time(s): 28
Avg Pressure Temp
-20362
-591
Light (%): 1023

Time(s): 28
Avg Pressure Temp
-20281
-591
Light (%): 1023

Time(s): 29
Avg Pressure Temp
-18379
-789
Light (%): 1023

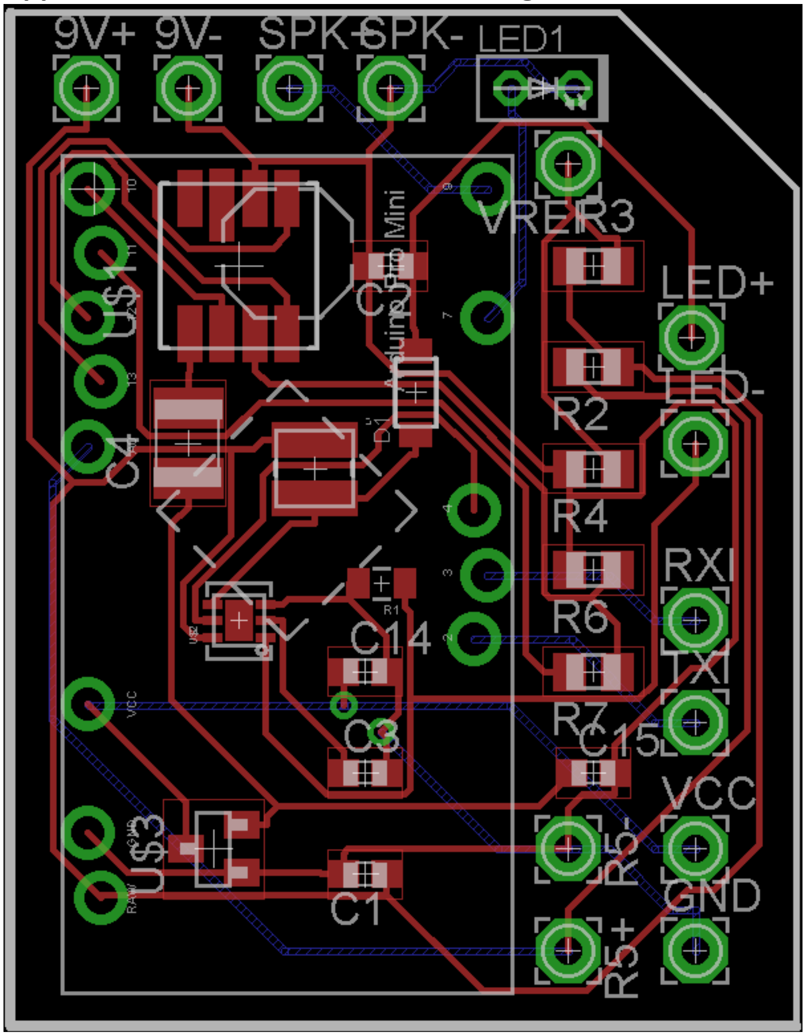
Time(s): 30
Avg Pressure Temp
12956
-1103
Light (%): 1023

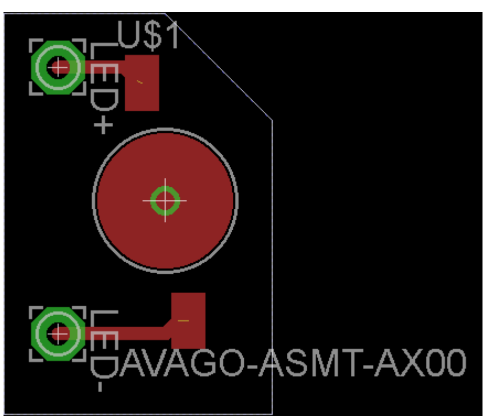
Time(s): 30
Avg Pressure Temp
-20110
-749
Light (%): 1023

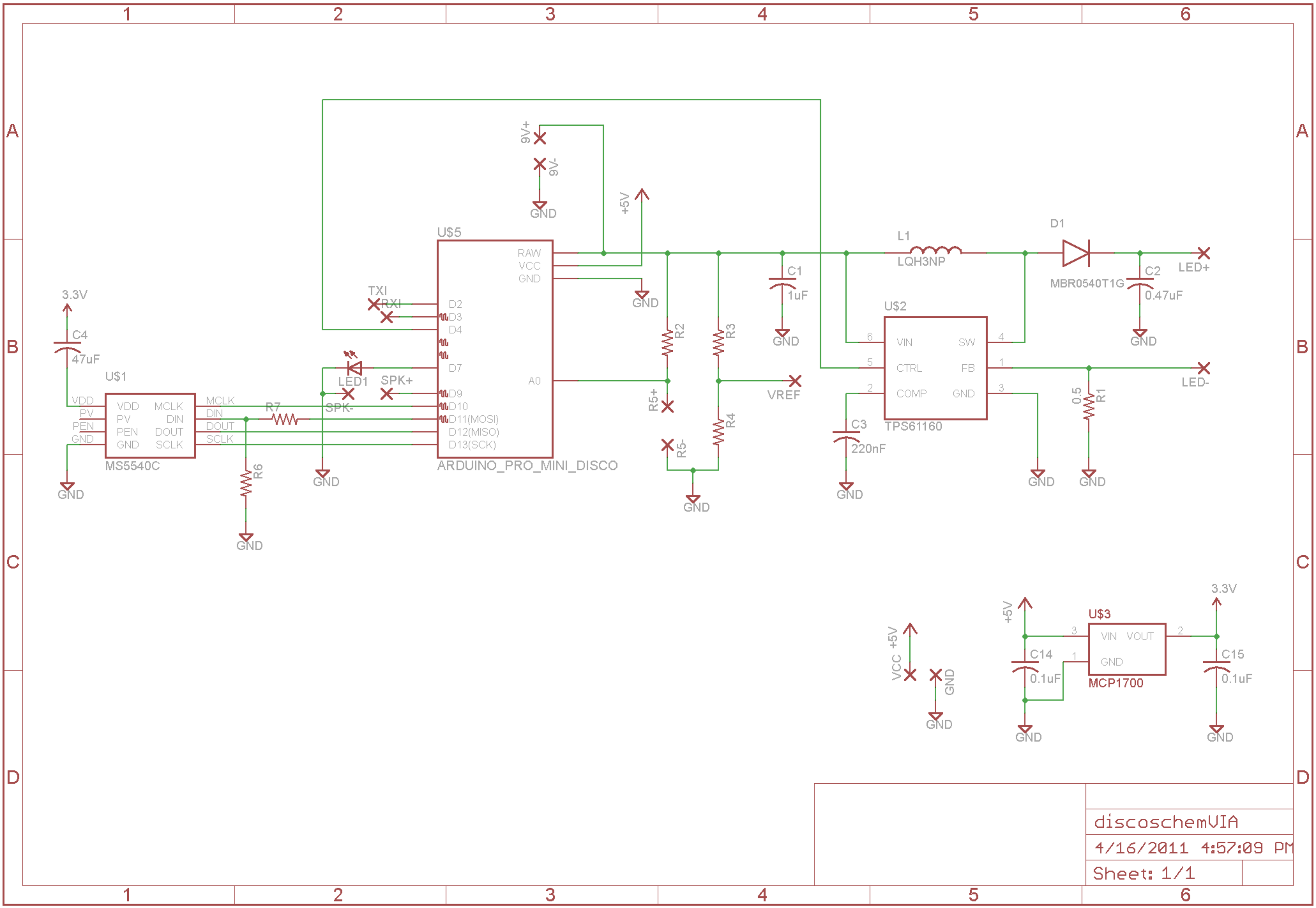
Appendix II. GPS Testing without RS-232 to TTL Conversion

RS-232
+ÿÿÿÔ
TTL
Ô
RS-232
}m•åŸ•™'§S§2•›£›—
§c§Ÿ™•&£››•§Q§§§›'Ÿ•:§ŸŸ"£§Q«"{åëÿÿ,'jb`bbjfnX¬XÍrjd\bdhrX
œX`pfjÙ\bdbjX®XXXdn`jbÅX``l\bX®Tl„
TTL
,'jb`bbjfnX¬XÍrjd\bdhrXœX`pfjÙ\bdbjX®XXXdn`jbÅX``l\bX®Tl„
RS-232
·q¾qq}§Ÿ•™'§™•›F›—§c§Ÿ™•"£›:•§Q§Ÿ§ŸŸ§§§e§§e§§«—›åë
TTL
HŽAŽŽ,X`bbjfnXfrjd¹bdhrXœX`pfjl\bdÅjX®X`X``XXXšXXšXXThdÿ
RS-232
+ÿÿÿÔ
TTL
Ô
RS-232
}m•åŸ•™§S§d•›£›—§c§Ÿ™•&£››•§Q§§§›'Ÿ•:§ŸŸ"£§Q«"—
åëÿÿ,'jb`bbjfpX¬X›rjd\bdhrXœX`pfjÙ\bdbjX®XXXdn`jbÅX``l\bX®Tlh
TTL
,'jb`bbjfpX¬X›rjd\bdhrXœX`pfjÙ\bdbjX®XXXdn`jbÅX``l\bX®Tlh
RS-232
·q¾qq}§Ÿ•™§™•›F›—§c§Ÿ™•"£›:•§Q§Ÿ§ŸŸ§§§e§§e§§«—wåë
TTL
HŽAŽŽ,X`bbjfpXfrjd¹bdhrXœX`pfjl\bdÅjX®X`X``XXXšXXšXXThˆÿ
RS-232
+ÿÿÿÔ
TTL
Ô
RS-232
}m•åŸ•™§S§2•›£›—
§c§Ÿ™•&£››•§Q§§§›'Ÿ•:§ŸŸ"£§Q«"•åëÿÿ,'jb`bbjfrX¬XÍrjd\bdhrX
œX`pfjÙ\bdbjX®XXXdn`jbÅX``l\bX®Tlj

Appendix III. Printed Circuit Board Design







U$5
RAW
VCC
GND
D2
D3
D4
D7
A0
D9
D10
D11(MOSI)
D12(MISO)
D13(SCK)
ARDUINO_PRO_MINI_DISCO
9V+
9V-
GND
+5V
TXI
RXI
LED1
SPK+
SPK-
U$1
VDD
PV
PEN
MCLK
DIN
DOUT
SCLK
MS5540C
3.3V
C4
47uF
R7
R6
R2
R3
R4
R5+
R5-
VREF
C1
1uF
L1
LQH3NP
D1
MBR0540T1G
C2
0.47uF
LED+
LED-
U$2
VIN
CTRL
COMP
SW
FB
TPS61160
C3
220nF
0.5
R1
VCC
U$3
VOUT
MCP1700
C14
0.1uF
C15
0.1uF
discoschemVIA
4/16/2011 4:57:09 PM
Sheet: 1/1