School of Electronic and Computing Systems (SECS)

University of Cincinnati

GPS Data Logger

Andrew Sweeney
Eric Fuhrman

6/4/12

Submitted in partial fulfillment of the degree of Bachelor of Science in Electrical Engineering Technology

Advisor: Xuefu Zhou
# Table of Contents

**Introduction** ....................................................................................................................................3

Abstract ....................................................................................................................................... 4

Problem ....................................................................................................................................... 4

Solution ....................................................................................................................................... 4

**PCB Design** .....................................................................................................................................6

Circuit Schematic ......................................................................................................................... 5

PCB Design ............................................................................................................................................ 6

**Testing and Calibration** ...............................................................................................................11

LCD Display .................................................................................................................................. 8

Single Point Location .................................................................................................................... 13

**Gant Chart** ......................................................................................................................................8

**Budget** .............................................................................................................................................7

**Embellishments** ............................................................................................................................14

**Conclusion** ....................................................................................................................................14

**Source Code** ..................................................................................................................................15

**Data Sheets** ...................................................................................................................................29

**References** .....................................................................................................................................32

**Table of Figures** ............................................................................................................................33
A. Introduction
   a. The purpose of this report is to explain the need for the GPS data logger and the need for an Introduction, discussion, conclusion, and resources.
   b. The project is an inexpensive GPS data logger for taking on backpacking trips. Most GPS’s on the market are entirely too expensive and not versatile, and durable.
   c. Developing a GPS in order to be able to track movement throughout the forest in all weather conditions. The project is to be accurate within 2 meters, and be able to track a full day’s hike, or last an entire weekend.
   d. I am an Electrical Engineer with experience in microcontrollers and their functionality
   e. The problem is that most of the GPS’s on the market today are too expensive and not expandable. The waterproof ones are even more expensive. This project is relatively inexpensive, tracks movements accurately, and allows for a more fluid movement, while still remaining waterproof.
   
B. Discussion
   a. Problem
      i. The problem is that the GPS’s are too expensive and inaccurate for the everyday hiker. Hiking throughout any weekend you must be able to cope with the harshness of nature. Most GPS’s are not waterproof and the ones which are, are entirely too expensive, over $200. This is a problem because most of the gear for hiking is expensive and in order to be able to accurately track where you go is a very useful tool in that arsenal. I cannot spend the amount that it would cost in order to get a GPS in which is waterproofed and useful for the duration of the time hiking.
   b. Solution/Methodology
      i. Design an inexpensive GPS data logger in which is waterproof and resilient to what nature and you can throw at it.
      ii. This is a solution because the Data Logger can be easily made and even manufactured later on as an inexpensive useful tool for hikers as opposed to the expensive GPS devices on the market.
      iii. The way in which I will implement this design is to have a suitable micro in order to talk with the GPS device and output the results to an LCD screen and an external storage device.
      iv. I believe that I will be able to fix this problem because I am a hiker and I know what the market needs for the whole project.
   c. Implementation
      i. Senior Design, and Gantt chart
      ii. Budget is under $100, which this project clearly meets those specifications
   d. Rationale/Benefits
      i. Chart
      ii. Chart should explain how we plan to use this device over the period of time and how it will be able to log all of the data for the project.
         1. The benefits of the project are the accuracy, durability, and usability
2. People who benefit are daytime hikers, climbers, mountaineers, anyone who is outdoors and wants to be able to track their movement throughout the woods.

3. Nature – storms and creeks do not bother it, and weather is a huge concern, Cost and accuracy

C. Conclusion
   a. The need for this project is a reliable GPS unit, in which are inexpensive and completely reusable many times. The GPS project has to be cheap, durable, and reliable for a period of time. This project completes all of those objectives.

D. Resources
   a. http://www.avrfreaks.net/
   b. http://www.avrprojects.net/

Abstract

What is a GPS data logger? The data logger created is an inexpensive GPS tracking unit that is waterproof and durable for backpacking, hiking, and other various outdoor activities. An individual can log where he or she has been on various journeys and then have the capabilities of retracing one’s steps back to the specific location using the great circle distance calculation. The GPS logger displays the individuals GPS coordinates on an LCD screen and saves those coordinates to an SD card to be tracked at a later time.

Problem/Solution

While hiking a person may forget specific locations they visited; this project is designed to display GPS coordinates for tracking. Also most modern day tracking devices are expensive. This device is designed to be simplistic and inexpensive. Most trackers are not durable and can be bulky. This device is designed to be compact and weather proof. The overall purpose of this project is to identify and trace a person’s GPS coordinates and be able to calculate distance between saved locations.

Conclusion

In conclusion, the purpose of this product is to better assist outdoor enthusiasts by tracking various locations to share those experiences with others or simply themselves. This product is a simplistic yet beneficial hand held electronic device that can withstand the test of Mother Nature.
Figure 1.1: This diagram represents the layout and connections throughout the finished GPS data logger.
PCB Design

The printed circuit board for the GPS Data Logger is designed to be compact and small. It is centered on an ATMEGA88 for a compact design. Figure 1.1 above is the schematic of the PCB design. Figure 1.2 below shows the physical layout of the PCB.

Figure 1.2: Created using CadSoft EAGLE PCB Design Software
**Budget**

The components required for this project are estimated in Figure 1.3.

<table>
<thead>
<tr>
<th>Product</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Misc</td>
<td>$11.29</td>
</tr>
<tr>
<td>LCD</td>
<td>$13.95</td>
</tr>
<tr>
<td>GPS Module</td>
<td>$17.06</td>
</tr>
<tr>
<td>40pin Header</td>
<td>$2.88</td>
</tr>
<tr>
<td>Battery Holder</td>
<td>$0.89</td>
</tr>
<tr>
<td>Resistors</td>
<td>$3.00</td>
</tr>
<tr>
<td>LED</td>
<td>$0.20</td>
</tr>
<tr>
<td>IC Sockets</td>
<td>$1.42</td>
</tr>
<tr>
<td>Capacitors</td>
<td>$1.60</td>
</tr>
<tr>
<td>Voltage Regulator</td>
<td>$2.02</td>
</tr>
<tr>
<td>Microcontroller</td>
<td>$2.85</td>
</tr>
<tr>
<td>SD Reader</td>
<td>$16.99</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>$74.15</strong></td>
</tr>
</tbody>
</table>

Figure 1.3: Budget
Gant Chart

Figure 1.4 below shows the time line and development process used to create the GPS Data Logger of excellent quality.

![Gantt Chart](image)

Figure 1.4: Gantt chart for the development and construction of GPS Data Logger.

**LCD Display**

The Liquid crystal display is the output module of the GPS Data Logger. The LCD used for the project is 16x2 character display (see datasheet for Xiamen Amotec Display). Figure 2.1, 2.2 shows the LCD output module for the GPS Data Logger.
Figure 2.1
Figure 2.2
Testing and Calibration

Below figure 3.1 show the testing of the GPS module in the Red River Gorge National Forest. I decided to take a trip to Eagles Point Buttress, and the GPS coordinates showed me that the total mileage for the trip was .828 miles. The sign on the roadmap states that it is a 1 mile trip. I am not certain how accurate that the road maps are in this area how precise the signs are. The fact that the sign stated 1 mile and I found the trip to be .828 miles leads me to believe that I am using a much more accurate device, or the rangers in the area were merely using a measure of rounding when posting signs.

Figure 3.1
Use search result as a distance marker? Yes No

Figure 3.2
Single Point Location

I was also curious on how accurate the GPS would be on one single point. I decided to pick a spot that would be noticeable from a satellite vantage point. The spot in which I chose is a large parking lot in the middle of the woods. The results of this test were quite pleasing as it showed me that the single point locating was extraordinarily accurate. As you can see in figure 4.1 below the point is taken from a single reading of the GPS LCD display. I know exactly where I had parked in the parking lot which is quite distinguishable from the rest of the parking lot. I parked right next to the dumpster which is located on the northern part of the parking lot directly next to the road, as figure 4.2 shows that GPS location plotted on a map.

Figure 4.1


**Embellishments**

The addition of a locator would be a nice change to the way that the GPS already operates. An input format would need to be coded into the software and the distance and direction calculations are already performed within the software. This would not be an issue to add on to the project. Data logging and the calculation are a drain on the battery so some form of power saving would be useful. The addition of a Photoreceptor to the LCD backlight would reduce the need for the backlight to be on during the day saving the battery further. A sleep mode would also be very useful to conserve battery life.

**Conclusion**

The GPS Data Logger is quite an efficient way to track yourself anywhere you may need something that is more resilient than a cell phone. The Data Logger also is much cheaper than most of the GPS’s on the market, which are $150 and up for the waterproof ones. While providing a low cost solution for Data Logging, and reading GPS coordinates, serves as a reasonably priced educational DIY project for future students with room for expansion.
Source Code

'Andrew Sweeney & Eric Fuhrman

'GPS Data Logger v1

'Senior Design

'Advisor: Professor Zhou

'Fall 2012

'6/7/12

,'

'Compiler version: bascom-avr 1.11.9

'Microcontroller ATMEGA88 PU

,'

,'

'-------- Compiler directives statement -------------------------------

$regfile = "m88def.dat"

$crystal = 8000000

$baud = 4800                                                 'by the GPS module output baud rate

'--------- LCD connection, key definitions -------------------------------

Config Lcdpin = Pin , Db4 = Portd.4 , Db5 = Portd.5 , Db6 = Portd.6 , Db7 = Portd.7 , E = Portd.3 , Rs = Portd.2

Config Lcd = 16 * 2

Config Pinc.3 = Input                                       'defined PortB.1 as input control terminal P

Portc.3 = 1                                                 'Portc .1 pull-up resistor and effective for the "1"

'-------------------SD card configureation and key definitions-------------------

' Chip Select Pin  => Pin 1 of MMC/SD
Config Pinb.2 = Output
Mmc_cs1 Alias Portb.2
Set Mmc_cs1

' MOSI - Pin => Pin 2 of MMC/SD DI
Config Pinb.3 = Output
Set Pinb.3
Mmc_portmosi Alias Portb
Bmmc_mosi Alias 1

' MISO - Pin => Pin 7 of MMC/SD D0
Config Pinb.4 = Input
Mmc_portmiso Alias Pinb
Bmmc_miso Alias 3

' SCK - Pin => Pin 5 of MMC/SD
Config Pinb.5 = Output
Set Pinb.5
Mmc_portsck Alias Portb
Bmmc_sck Alias 2

'-------- Variables and constants are defined in ------------------------------------
Dim Ews As String * 10 , Nss As String * 10 , Km As String * 5       'definition of longitude,
latitude, speed,
Dim Hh As String * 2, Mm As String * 2, Ss As String * 2 'definition of hours, minutes, seconds
Dim Ns As String * 1, Ew As String * 1, Dw As String * 1 'definition of latitude north and south signs, things longitude signs, whether the location
Dim Ye As String * 2, Mo As String * 2, Da As String * 2 'Definition of the year month and date
Dim Londeg As String * 6, Latdeg As String * 9
Dim Dblon As Single, Dblat As Single, Temp1 As Single, Temp2 As Single
Dim Speed As Single
Dim Pval As Integer, Pval2 As Integer, Ho As Integer
Dim Ldlat(100) As Single, Ldlon(100) As Single
Dim A As Single, B As Single, C As Single, D As Single, E As Single, F As Single, G As Single, H As Single, I As Single, J As Single, K As Single
Dim Distance As Single
Dim R As Integer, Qrntr As Byte

Config Pinc.5 = Input
Entersw Alias Pinc.5

Entersw = 1
Qrntr = 0

Dim Przecinek As Byte, P As Byte
Dim Run As Byte, Gprmc As Byte, Cntr As Byte, Tuder As Byte, Ok As Bit, X As Integer
Run = 1 : Reset Ok
P = 1
Pval = 2
R = 3963
Config Xram = Enabled

'-------------- Defined and open the serial port interrupt subroutine interrupt ---------------------------
------------
On Urxc Uart_rx
Enable Urxc
Enable Interrupts
Initial:

------
Cls
Cursor Off
Locate 1, 4
Lcd "Hello World!"
Lowerline
Lcd "Adventure Time!"

Wait 1

'----------------- The main program display interface -----------------

Main:
Do
While Ok = 0 : Wend
Reset Ok

If Pinc.3 = 0 Then

'Pinb.1 grounding also open up
'Switch to display the page
Do
Loop Until Pinc.3 = 1
Incr P
If P > 3 Then
P = 1
End If
End If

If Dw = "V" Then
Goto Lock
End If

Select Case P
' - P1
Case 1

Pval = Val(latdeg)
Pval2 = Val(londeg)
Dblon = Val(nss)
Dblat = Val(ews)
Temp1 = Dblon / 60
Temp2 = Dblat / 60
Dblon = 0
Dblat = 0
Dblat = Temp1 + Pval
Dblon = Temp2 + Pval2
If Ew = "W" Then
Dblon = -dblon
End If

If Ns = "S" Then
Dblat = -dblat
End If

Cls
Cursor Off
Locate 1, 1
Lcd Dblat
Locate 2, 1
Lcd Dblon
Locate 2, 15
Lcd Dw

If X >= 600 Then
If Entersw = 0 Then
P = 1

Goto Calculate
X = 0
End If
End If

X = X + 1 'X can go up to 32768 before overflow

Case 2

Ho = Val(hh)
Ho = Ho - 4
X = X + 1
If X >= 600 Then
If Entersw = 0 Then
P = 2
Goto Calculate
End If
End If
End If

Cls
Cursor Off

Locate 1, 1
Lcd "UTC" : Lcd Hh : Lcd ":" : Lcd Mm : Lcd ":" : Lcd Ss ' first line of the display UTC time

Locate 2, 1
Lcd "20" : Lcd Ye : Lcd "-" : Lcd Mo : Lcd "-" : Lcd Da ' second line shows the date, but there are problems
Case 3

Speed = Val(km)
Speed = Speed * 1.5077
Cls
Cursor Off
Locate 1, 1
Lcd Distance
Locate 1, 5
Lcd "miles"
Locate 2, 1
Lcd Speed
Locate 2, 5
Lcd "speed mi/h"
Locate 2, 15
Lcd Dw
X = X + 1
If X >= 600 Then
If Entersw = 0 Then
P = 3
Goto Calculate
End If
End If
End Select
If X > 3000 Then
X = 0
End If

Loop
End

Calculate:                                                   'This is in order to compute the Latitude and Longitude in
Decimal form
Disable Interrupts

Ldlat(qrntr) = Dblat
Ldlon(qrntr) = Dblon

If Qrntr > 2 Then
If Ew = "W" Then
Dblon = -dblom
End If

If Ns = "S" Then
Dblat = -dblat
End If

A = Ldlat(qrntr)
B = Ldlon(qrntr)
C = Ldlat(qrntr - 1)
D = Ldlon(qrntr - 1)
A = Deg2rad(a)
B = Deg2rad(b)
C = Deg2rad(c)
D = Deg2rad(d)

If A = C And B = D Then
   Distance = 0
Else
   E = Sin(a)
   F = Sin(c)
   G = E * F  'sin(A)*sin(C)
   H = Cos(a)
   I = Cos(c)
   J = H * I  'cos(A)*cos(C)
   If D > B Then
      K = D - B
   Else
      K = B - D
   End If
   K = Cos(k)  'cos(b-d)
   K = J * K  'cos(A)*cos(C)*cos(b-d)
   F = G + K  'SIN(A)SIN(C)+COS(A)COS(C)COS(B-D)
   Distance = R * Acos(f)
End If
End If
Cntr = 0
Entersw = 1
X = 0
Set Ok
Run = 1
Incr Cntr
Enable Interrupts

Return

Lock:
Cls
Locate 1, 1
Lcd "Cannot find signal"
Wait 2
Cntr = 0
Entersw = 1
Set Ok
Run = 1
P = 1
Incr Cntr
Return

'Big thanks to AVRfreaks.com for help with the serial port subroutine

'---------------------- Serial port interrupt reception subroutine ---------------------- -----

Uart_rx:
If Run <> 0 Then
Run = Udr Xor 36
If Run = 0 Then
    Cntr = 0
    Nss = ""
    Ews = ""
    Km = ""
    Ns = ""
    Ew = ""
    Dw = ""
    Londeg = ""
    Latdeg = ""
    Hh = ""
    Mm = ""
    Ss = ""
    Ye = ""
    Mo = ""
    Da = ""
End If
Else
Select Case Cntr
Case 0 To 4
    Tudr = Lookup(cntr, Rmc) "to determine whether the received characters
        include "GPRMC"
    Run = Udr Xor Tudr
Case 6 To 7
Hh = Hh + Chr(udr)  'UTC Time: "time"
Case 8 To 9
Mm = Mm + Chr(udr)  'UTC Time: "Separation"
Case 10 To 11
Ss = Ss + Chr(udr)  'UTC Time: "seconds"
Case 17
Dw = Dw + Chr(udr)  'whether the location mark
Case 19 To 20
Latdeg = Latdeg + Chr(udr)
Case 21 To 27
Nss = Nss + Chr(udr)  'latitude
Case 29
Ns = Ns + Chr(udr)  'North latitude flag
Case 31 To 33
Londeg = Londeg + Chr(udr)
Case 34 To 40
Ews = Ews + Chr(udr)  'longitude
Case 42
Ew = Ew + Chr(udr)  'stuff longitude flag
Case 44 To 47
Km = Km + Chr(udr)  'speed, unit section, a = 1852M / H, is not converted into kilometers
Case 56 To 57
Da = Da + Chr(udr)
Case 58 To 59
Mo = Mo + Chr(udr)
Case 60 To 61
Ye = Ye + Chr(udr)
Case 64
Set Ok : Run = 1

Case Else
Tudr = Udr
End Select

Incr Cntr
End If
Return

'GPS statement, the head of ASCII code

Gga:
Data 71, 80, 71, 71, 65 'GPGGA

Vtg:
Data 71, 80, 86, 84, 71 'GPVTG

Rmc:
Data 71, 80, 82, 77, 67 'GPRMC
4. Absolute maximum ratings

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Standard</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power voltage</td>
<td>(V_{DD} - V_{SS})</td>
<td>0</td>
<td>7.0</td>
</tr>
<tr>
<td>Input voltage</td>
<td>(V_{IN})</td>
<td>(V_{SS})</td>
<td>(V_{D}D)</td>
</tr>
<tr>
<td>Operating temperature range</td>
<td>(V_{OP})</td>
<td>0</td>
<td>+50</td>
</tr>
<tr>
<td>Storage temperature range</td>
<td>(V_{ST})</td>
<td>-10</td>
<td>+60</td>
</tr>
</tbody>
</table>

5. Block diagram

6. Interface pin description

<table>
<thead>
<tr>
<th>Pin no.</th>
<th>Symbol</th>
<th>External connection</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Vss</td>
<td>Power supply</td>
<td>Signal ground for LCM</td>
</tr>
<tr>
<td>2</td>
<td>(V_{DD})</td>
<td>Power supply</td>
<td>Power supply for logic for LCM</td>
</tr>
<tr>
<td>3</td>
<td>(V_{G})</td>
<td>Contrast adjust</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>RS</td>
<td>MPU</td>
<td>Register select signal</td>
</tr>
<tr>
<td>5</td>
<td>R/W</td>
<td>MPU</td>
<td>Read/write select signal</td>
</tr>
<tr>
<td>6</td>
<td>E</td>
<td>MPU</td>
<td>Operation (data read/write) enable signal</td>
</tr>
<tr>
<td>7~10</td>
<td>DB0~DB3</td>
<td>MPU</td>
<td>Four low order bi-directional three-state data bus lines. Used for data transfer between the MPU and the LCM. These four are not used during 4-bit operation.</td>
</tr>
<tr>
<td>11~14</td>
<td>DB4~DB7</td>
<td>MPU</td>
<td>Four high order bi-directional three-state data bus lines. Used for data transfer between the MPU</td>
</tr>
<tr>
<td>15</td>
<td>LED+</td>
<td>LED BKL power supply</td>
<td>Power supply for BKL</td>
</tr>
<tr>
<td>16</td>
<td>LED-</td>
<td>LED BKL power supply</td>
<td>Power supply for BKL</td>
</tr>
</tbody>
</table>
Features
- High performance, low power Atmel® AVR® 8-bit microcontroller
- Advanced RISC architecture
  - 131 powerful instructions – most single clock cycle execution
  - 32 x 8 general purpose working registers
  - Fully static operation
  - Up to 26 MIPS throughput at 20MHz
  - On-chip 2-cycle multiplier
- High endurance non-volatile memory segments
  - 4/8/16 Kbytes of in-system self-programmable flash program memory
  - 256/512/512 bytes EEPROM
  - 8/2/1K bytes internal SRAM
  - Write/erase cycles: 10,000 flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C
  - Optional boot code section with independent lock bits
  - In-system programming by on-chip boot program
  - True read while write operation
  - Programming lock for software security
- QTouch™ library support
  - Capacitive touch buttons, sliders and wheels
  - QTouch and QMatrix acquisition
  - Up to 64 sense channels
- Peripheral features
  - Two 8-bit timer/counters with separate prescaler and compare mode
  - One 16-bit timer/counter with separate prescaler, compare mode, and capture mode
  - Real time counter with separate oscillator
  - Six PWM channels
  - 8-channel 10-bit ADC in TQFP and QFN/MLF package
  - 6-channel 10-bit ADC in PDIP Package
  - Programmable serial USART
  - Master/Slave SPI serial interface
  - Byte-oriented 2-wire serial interface (Philips IC compatible)
  - Programmable watchdog timer with separate on-chip oscillator
  - On-chip analog comparator
  - Interrupt and wake-up on pin change
- Special microcontroller features
  - DebugWIRE on-chip debug system
  - Power-on reset and programmable brown-out detection
  - Internal calibrated oscillator
  - External and internal interrupt sources
  - Five sleep modes: Idle, ADC noise reduction, power-save, power-down, and standby
- I/O and packages
  - 23 programmable I/O lines
  - 28-pin DIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating voltage:
  - 1.8V - 5.5V for Atmel ATmega48/88/168V
  - 2.7V - 5.5V for Atmel ATmega48/88/168
- Temperature range:
  - -40°C to 85°C
- Speed grade:
  - ATmega48/88/168V: 0 - 4MHz @ 1.8V - 5.5V, 0 - 10MHz @ 2.7V - 5.5V
  - ATmega48/88/168: 0 - 10MHz @ 2.7V - 5.5V, 0 - 20MHz @ 4.5V - 5.5V
- Low power consumption
  - Active mode:
    - 250μA at 1MHz, 1.8V
    - 15μA at 32kHz, 1.8V (including oscillator)
  - Power-down mode:
    - 0.1μA at 1.8V

Note: 1. See "Data Retention" on page 7 for details.
PMB-648 GPS module

PMB-648 FEATURES

- Built-in SiRFstarIII chipsets receivers give unparalleled GPS performance and precision. 20 parallel satellite-tracking channels for fast acquisition and reacquisition.
- Built-in WAAS /EGNOS Demodulator.
- Low power consumption and ultra mini size only 32x32mm.
- Built-in rechargeable battery for backup memory and RTC backup.
- Support NMEA0183 v2.2 data protocol.
- Enhanced algorithms providing superior navigation performance in urban, canyon and foliage environments.

Top View

Side View

32mm

9.0 mm

Connector Pitch 1.25mm

Pin 1 2 3 4 5 6

RS-232 RX
RS-232 TX
Gnd
Vcc
TTLRX
TTL TX
References


Table of Figures

Figure 1.1 ...................... Schematic
Figure 1.2 ...................... PCB
Figure 1.3 ...................... Budget
Figure 1.4 ...................... Gantt Chart
Figure 2.1 ...................... LCD Display
Figure 2.2 ...................... LCD Display 2
Figure 3.1 ...................... GPS Data from Red River Gorge
Figure 3.2 ...................... GPS Plot of Data
Figure 4.1 ...................... GPS Single Point Location
Figure 4.2 ...................... GPS Single Point Location Map