Separated GPS Unit Design Document

Al Niessner

April 7, 2011

Contents

List of Figures

List of Tables

1 Overview

The GPS design described in this document is satisfies two criteria: One, the display and receiver should be separated as to make the receiver as light and simple to carry as possible making the unit simpler to carry and use while hiking. Two, the display unit should use open protocols and open source freeing it from all of proprietary nonsense that currently surrounds the one that I own. To these ends, I designed, with help, yet another GPS unit and made it an open design and open software.

The main design feature is quite simple. Build a GPS receiver and data recorder with a USB interface and they use a laptop or netbook or tablet or smart phone or some device with its own power supply and nice display. The receiver should operate independently of being connected to the display, but the display should power the unit when connected. This breaks the design into three essential parts: One, the hardware for the GPS receiver and data recorder. Two, the firmware used on the hardware to receive and record the data. Three, the software that resides on the commercially available display device.

2 Hardware

Figure 1 on the following page shows the overall layout of the hardware design. The idea is to use lower power parts that can accumulate the GPS data until a much more sophisticated and power hungry device reads it and does something useful with it. Hence, the center piece of the design is the uC. It is responsible for giving access to the data and GPS from the USB interface as well as shuffling the GPS data to Flash memory. A small number of switches and LEDs are the input and output to the user during normal operation.

The Microchip PIC18LF14K50 uC contains several modules necessary to accomplish the tasks assigned to it. It has a built in USB module that handles the USB protocol and data transfer. It has a UART for communicating with the GPS unit. It has a memory module for Flash memory. Finally, it has built in IO for the switches and LEDs.

The power source is the other key element in the block diagram. The unit should have its own power when not plugged into the USB, but it should use USB power when connected. The power source shown in the block diagram is responsible for switching between a battery pack and USB without loss of power to the rest of the circuit. It must also regulate this voltage to 3.3 volts. While the uC will continue to operate with voltages as low as 2.7 volts, the GPS has a higher threshold of 3.0 volts.

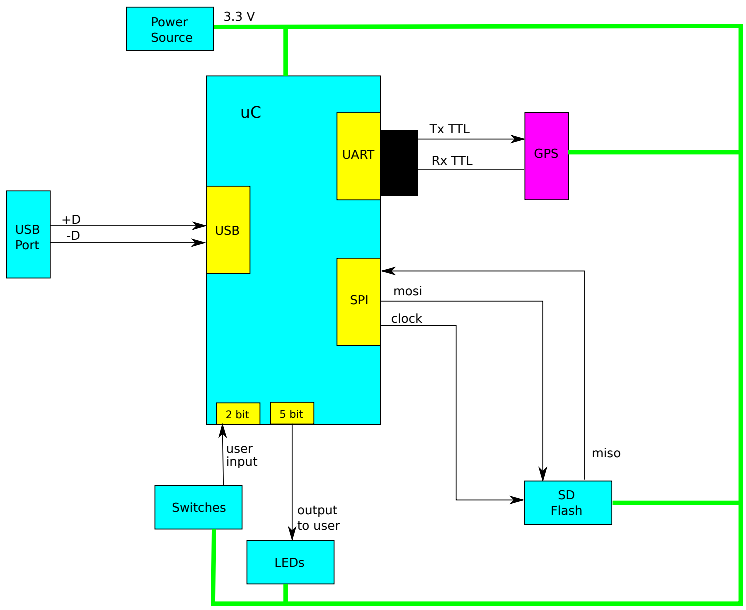

Figure 1: Hardware Block Diagram

2.1 Power Source

The power source element of the block diagram in figure 1 has to switch between the USB and battery source as the USB is willy nilly plugged in and out without loss of power to the rest of system. Figure 2 on the following page shows the complete circuit for the power source module. The diodes protect the circuits from reverse bias but there is always some current flow in the reverse direction. The resister R between the battery and diode allow the reverse current flow to be shunt to ground allowing the battery to safely discharge at all times. The capacitor after the voltage regulator gives the diodes time to switch between the USB and battery. Given the circuit will draw less than 100 mA under normal operation, the capacitor C was chosen to allow about 10 μs for the circuit to switch between the two diodes. Both diodes are Schottky diodes, chosen for their low forward voltage drop, and have nanosecond response times.

The parts list for segment of the power supply circuit is in Table 1 on the next page.

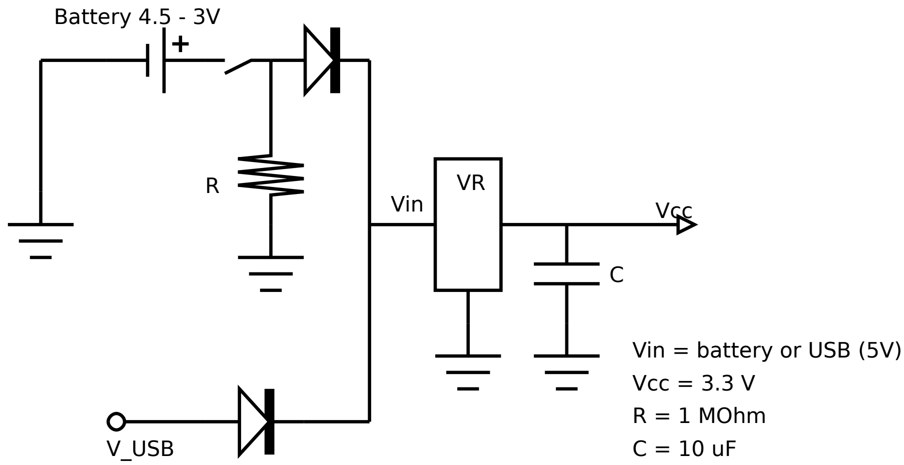

Figure 2: Power Source Circuit

VR is a voltage regulator.

Nomenclature	Part Number
Battery	3x AAA alkaline
Diodes	497-2493-1-ND
Voltage Regulator	
R	
C	

Table 1: Power Source Parts List

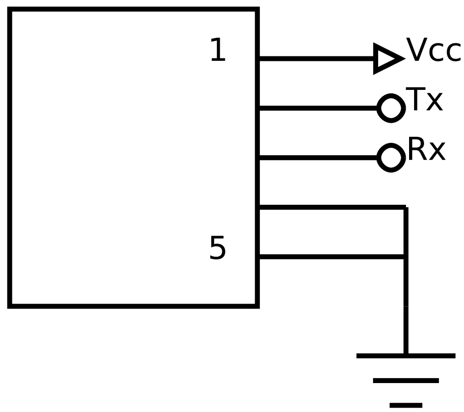

Figure 3: GPS Connection

In this diagram the signal known as Tx is connected to the input of the GPS unit, and the Rx signal is connected to the output of the GPS.

2.2 GPS

The GPS is an off-the-shelf component and requires no work other than to interface to it. The part number is LS2003-1 and it is connected to the uC and power as shown in figure 3.

2.3 micro SD Flash

The SD Flash is an off-the-shelf component and requires no work other than to interface to it. Figure 4 on the next page shows a simple non-hot-swap connection for the memory. It allows the user of the device to change the memory and size it according to their desire.

2.4 Switches and LEDs

There are four switches and five LEDs. Two of the switches feed the uC as user input and the other two have influence over the circuit. One of the switches is used in the power source (see section 2.1). The discussion in this section will cover the five LEDs and three switches shown in figure 5 on page 8 whose parts are listed in table 2 on the next page.

LEDs 0 through 3 inclusive are used to display the state of the FSM discussed in section 3.1. Switch A is a push button switch that allows the state to be viewed. This is a power saving mechanism because those LEDs are not monitored most of the time.

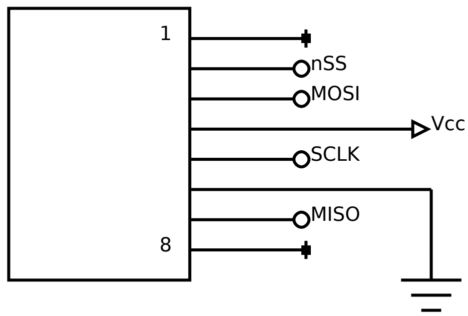

Figure 4: micro SD Flash Circuit

This circuit is NOT hot swap capable. Need to follow the manufactures power up circuit for hot swap.

Nomenclature	Part Number
LED 0-3	LN1471SYTRP
LED 4	LNJ312G8LRA
SW A	D6R30 F1 LFS
SW B	D6R40 F1 LFS
SW C	D6R10 F1 LFS
R_1	
R_2	

Table 2: User Interface Parts List

LED 4 is the indicator to show that the uC has detected switch B or C or both have been pressed for at least 1 second.

2.5 USB

A USB-B connector is used to signal to the user that the device will be using power from the connection. The circuit is shown in figure 6 on the following page. The 5V from the USB powers three different circuits. The first is the uC to allow it to match its internal voltage with that of the USB. The second is the power supply to supply entire device. The third is to drive an interrupt pin to let the controller know that the USB has been connected or removed. The voltage divider to reduce the USB 5V to 3.4V at the uC interrupt pin was taken directly off the XuLA board[1].

[1]For more details on the XuLA board see http://www.xess.com

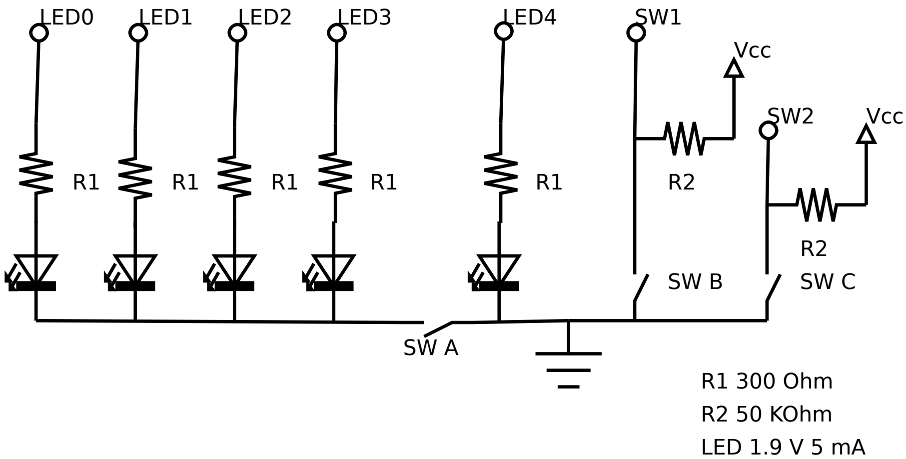

Figure 5: User Interface

LED 0 through 4 inclusive as well as SW 1 and 2 map to the micro controller. R_1 and R_2 are current limiting resistors to protect the micro controller IO pins.

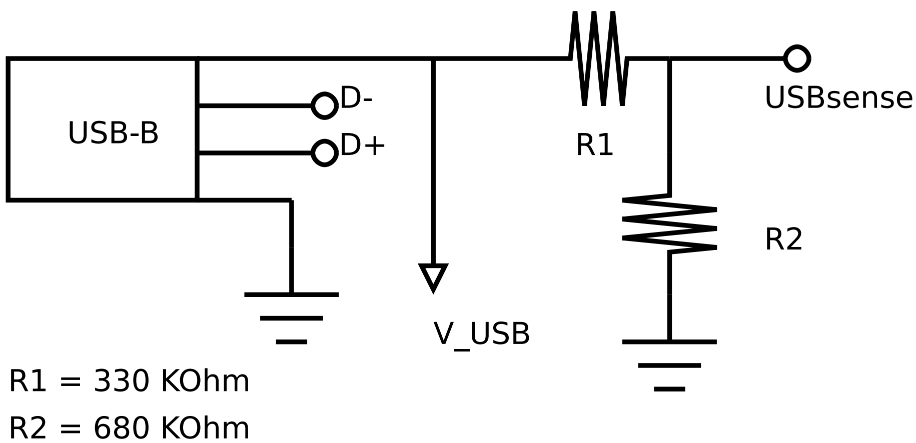

Figure 6: USB Connection

The V_{USB} supply goes to the uC and the power supply both.

Nomenclature	Part Number
USB-B	
R_1	
R_2	

Table 3: USB Part List

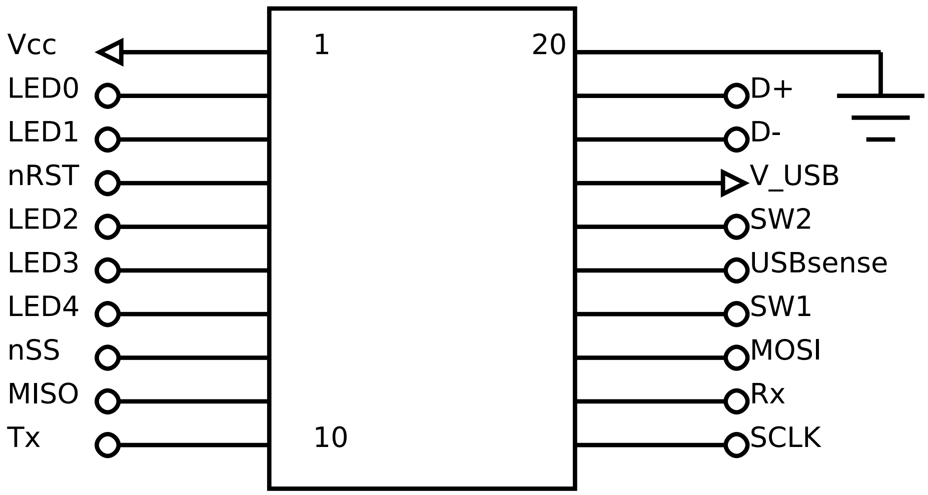

Figure 7: uC Connections

All of these connections can be found in all of the previous subsections.

2.6 uC

The complexity of the uC is not in the circuit wiring show in figure 7. Instead it hides its complexity in its firmware described in detail in section 3.

2.7 Power

One of the most important parts of this device is that it is low power when in normal operation. Table 4 on the following page gives a listing of each module in figure 1 on page 4. These are estimated values from data sheets.

3 Firmware

3.1 Finite State Machine (FSM)

As shown in Figure 8 on page 12 a Finite State Machine (FSM) is used to describe how the micro-controller interacts with the stimuli to record the GPS data and allow commands from the USB. The states are:

S0 Power Up State

1. Turn all LEDs off

Module	I_{Typ}	I_{max}	Power (typ)	Power (max)
Power Source		-200 mA		-1 W
GPS	30 mA	30+ mA	99 mW	99+ mW
SD Flash	300 μA	100 mA	990 μA	330 mW
LEDs	5 mA	25 mA	16.5 mA	82.5 mW
Switches	66 μA	66 μA	218 μW	218 μW
USB	0	0	0	0
uC		245 mA		800 mW

Table 4: Power Consumption

The power consumption from the switches and LEDs are not a continuous load but user demanded loads.The negative current and power represents that the power source module is sourcing the current and power rather than consuming it.

2. Set global record boolean to False
3. Wait for event:
   (a) 1 Hz
   (b) USB sending data
   (c) clear button pushed
   (d) track button pushed

S1 Idle

1. Set global record boolean to True
2. Clear push buffer
3. Wait for event:
   (a) 1 Hz
   (b) UART receiving data
   (c) USB sending command
   (d) track button pushed
   (e) waypoint button pushed

S2 Track button

1. Change the state of the button LED
2. When light is on, put $T<CR><LF> into push buffer

3. Generate Always event

S3 Track button

1. Change the state of the button LED
2. When light is on, put $W<CR><LF> into push buffer
3. Generate Always event

S4 UART Get character

1. Put it into push buffer
2. Generate Always event

S5 USB Process command

1. Jump to specified next state

S6 Clear button

1. Set the button LED to on Clear memory
2. Wait 1 second set the button LED to off
3. Generate Always event

S7 Push message

1. Set last to the last character in the message
2. Stream the message into the FLASH
3. Generate Always event

S8 Write to GPS

1. Send the message string to GPS and append <CR><LF>
2. Generate Always event

S9 Pop message

1. Stream data from FLASH to local memory until and including <LF>
2. Send message back along USB
3. Generate Always event

For S8 and S9 they return to S0 or S1 depending on global record boolean.

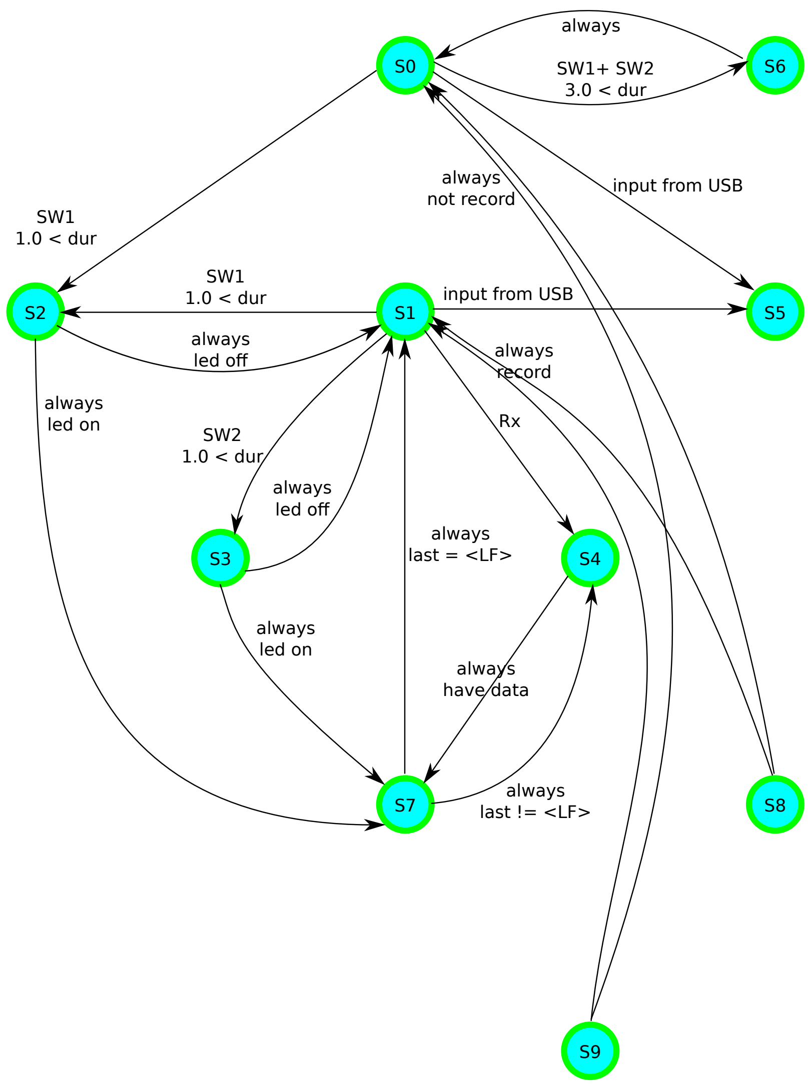

Figure 8: Firmware FSM
On transitions, the event is always listed. In multi-line transitions, the first or upper line is the event and the subsequent information are the guards to the transition.

4 Software