GPS Series

By Michael Simpson

Build a GPS Smart Logger

As seen in November 2008 of Servo Magazine
Pick up an issue at
www.servomagazine.com

I recently did a GPS series covering various GPS modules and their interfaces. A project that I have had in mind for a while was a small GPS Smart Logger. I call it a smart logger because in addition to the GPS data, you can log various other telemetry. You can also set the conditions and type of GPS data logged. When I started this project, I came up with the following requirements:

- Self contained operation
- Able to operate on battery for 24 hours
- Able to detect low battery levels
- Log data must be written to SD memory card
- WAAS support
- Single button operation
- LED indicators
- 3.3v operation
- Ability to log additional data
- Must Cost under $150 to build.

I call this a smart logger because you have the ability to choose what does and does not get written to the SD memory card. You also can use one of the many AtoD lines on the DiosPro to log additional data. I chose to operate the smart logger at 3.3v for a couple of reasons. First, both the SD card and the GPS module operate at 3.3v. Second, when operating at the lower voltage, more run time is obtained on a set of batteries.

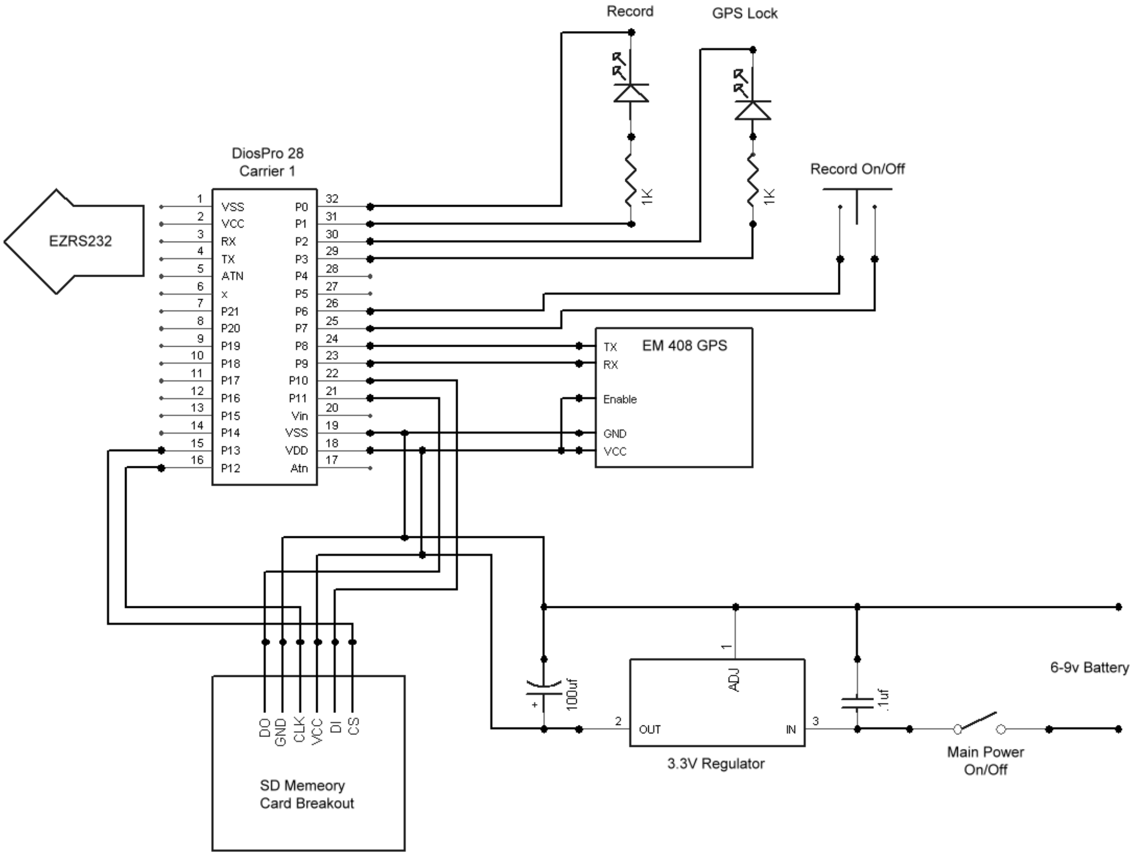
The DiosPro microcontroller chip that I have chosen to use in this project will operate at 3.3v just as easy as it does at 5v. The DiosPro also supports a LVD library that is capable of detecting a low voltage drop. This is important in order to keep from corrupting the log file when the battery dies.

The wiring diagram for the project is shown in Schematic 1. I have used a couple of techniques to make hookup easier. Notice that the two LEDs are connected to 2 ports each. Ports 0 and 2 are held low, so this provides a GND for the LEDs. While this uses a port on the DiosPro, it does make the connection of the LEDs much simpler as you will see later.

Another shortcut I have taken is to tie both leads on the record button to Ports 6 and 7. The DiosPro has the ability to hold Ports 0-7 high with a weak internal resistor. Port 6 is held low so that when the record button is pressed, this forces Port 7 low as well.

The EM 408 GPS connection is straight forward. I have simply connected it to the onboard UART. Notice how the Enable lead is also tied to VCC. This needs to be done in order for the GPS module to operate.

The SD Memory card interface utilizes the Dios MMC library and is forced to use IO ports 10, 11, and 12. The CS (Card Select) is configurable and can be set to any port. In this case, I have used Port 13.



Schematic 1

Construction

You will need the following components in order to complete the project. Please refer to the source listing at the end of this article for part numbers.

Major Functional Components

- DiosPro 28 chip
- Dios Carrier 1
- EM 408 GPS
- SD Breakout Board
- EZRS232 (Needed for Programming the DiosPro)
- SD Memory Card

Additional Components

- Push Button
- 2, Red LEDs
- 2, Green LEDs
- 100uf Capacitor
- On/Off Switch
- Jumpers
- Battery Holder
- Battery Connector Clip
- 2, 1K Resistors
- 36-Pin Female Header
- 40-Pin Male Header
- 2, 4.5” x 3.5” Pieces of Plastic
- 3.3v Regulator
- 4, #4 1” Standoffs (F-F)
- Female Crimp Pins
- 8, 1/2" #4 machine screws
- 8, #4 hex nuts
- Double Stick Foam Tape

You will also need a copy of the Dios compiler. This compiler is free and can be found on the KronosRobotics web site at www.kronosrobotics.com

Step 1

The first thing you need to do is to build the Dios Carrier 1. Build it according to the included instructions but hold off attaching the two included 16-pin headers as shown in Figure 2. The carrier comes with a 10uf capacitor and this is plenty for the circuit, but you may want to replace it with a 100uf capacitor. This will keep the insertion of the memory card from resetting the DiosPro chip. A 100uf capacitor won't fit on the board, so you will have to bend the pins and insert it at an angle.

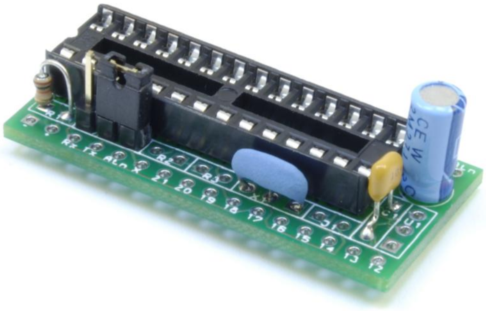


Figure 2

Step 2

The project requires a 3.3v regulator to power the circuit. The most compact way to do this is to attach it directly to the Dios Carrier 1 board. Take the regulator and insert pins 1 and 2 into the board as shown in Figure 3. From the underside of the board, bend pin-1 on the regulator and solder in place as shown in Figure 4. Be sure to solder both at the entry pad and the pad located at the tip of the pin.

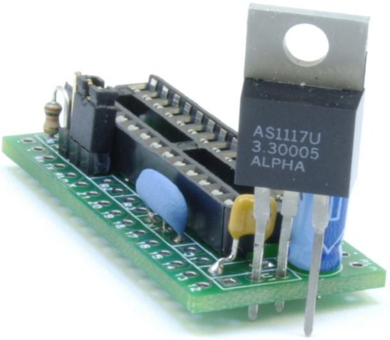


Figure 3

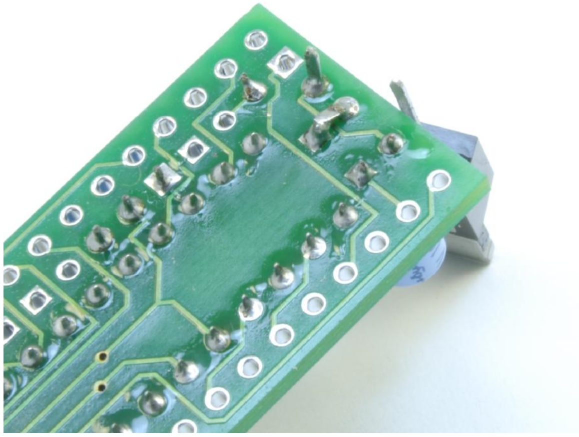


Figure 4

Now gently twist the regulator and insert the remaining pin into the hole shown in Figure 5. With the regulator attached, you can now attach the two 16-pin headers on the top of the board as shown in Figure 6. Take an additional 2-pin header and attach it to the position marked J2.

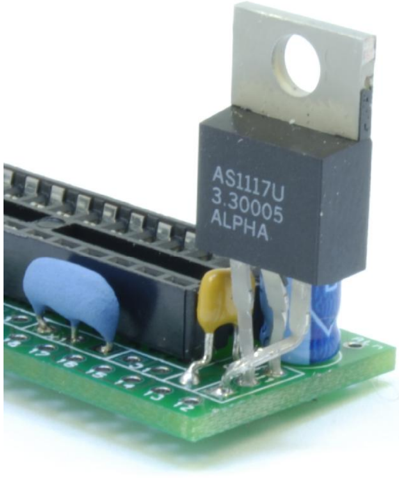


Figure 5

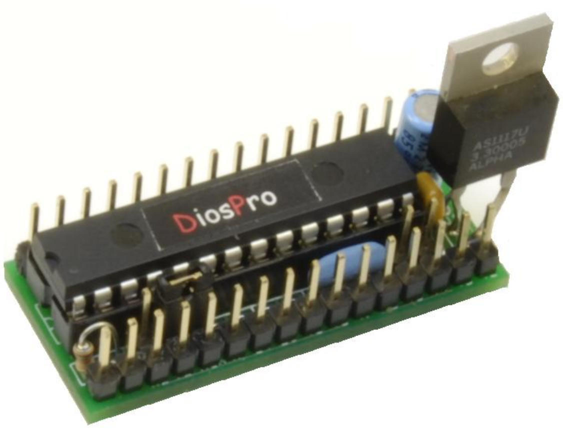


Figure 6

Step 3

You will need to cut two pieces of plastic like the ones shown in Figure 4. I used compressed PVC cut to 4.5” x 3.5”. The bulk of the size of the project is devoted to the batteries used to power the device. I am using a 6 AA cell pack to power my unit. If you want to use a small 9v-like battery this would allow you to create a much smaller enclosure. The trade off, however, would be less run time. Place 4 small 1/8” holes in each corner about 1/8” from both edges. A trick I use to place the holes all in the same place in each corner is to drill one hole in one piece. Then use that hole to mark each of the corners in the other piece. You do this by flipping the piece with the hole as needed. Once all 4 holes are marked, drill them out and use this piece to make the original.

Take the completed carrier module and attach some double stick foam tape to the bottom, then attach the carrier to the base as shown in Figure 8. Notice how I placed standoffs on the board so the I can judge the distances better.

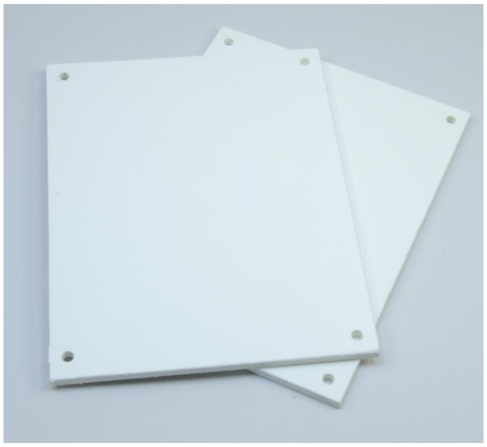


Figure 7

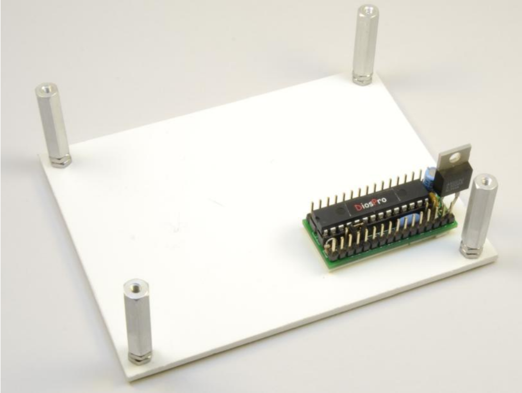


Figure 8

Step 4

The EM408 comes with a small cable that needs to be modified for this project. You can even purchase an extra cable from SparkFun if need be. Cut one end and attach the leads to small headers as shown in Figure 9.

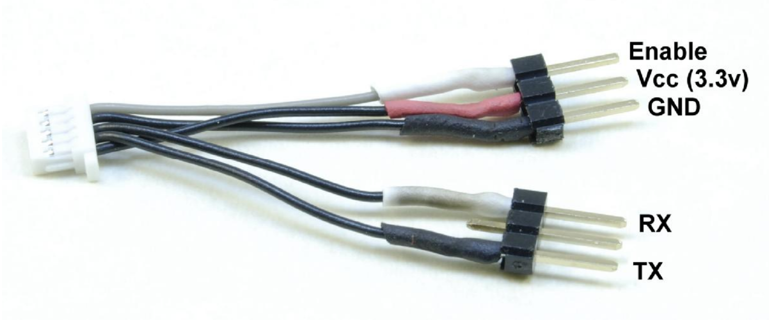


Figure 9

Step 5

Take the SD memory breakout board and attach a 5-pin header into the first five pads shown in Figure 10. Take a few pieces of double stick foam tape and attach them to the memory board as shown in Figure 11. Now take the EM 408 and attach it to the memory board as shown in Figure 11. Make sure the antenna connector is off to one side of the card slot. This way, if you decide to use an external antenna it won't be in the way when you insert the memory card into its socket. Place some double stick tape on the bottom of the memory breakout board and attach it to the base as shown in Figure 12.

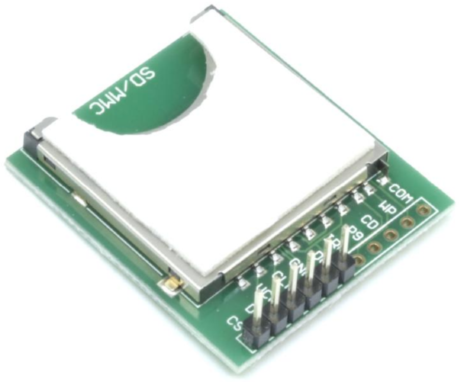


Figure 10

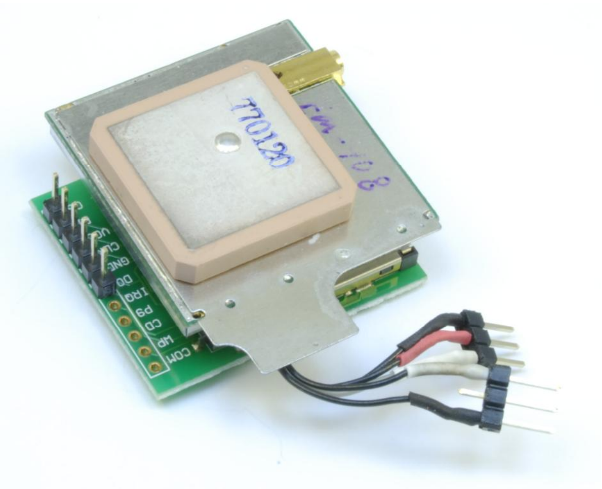


Figure 11

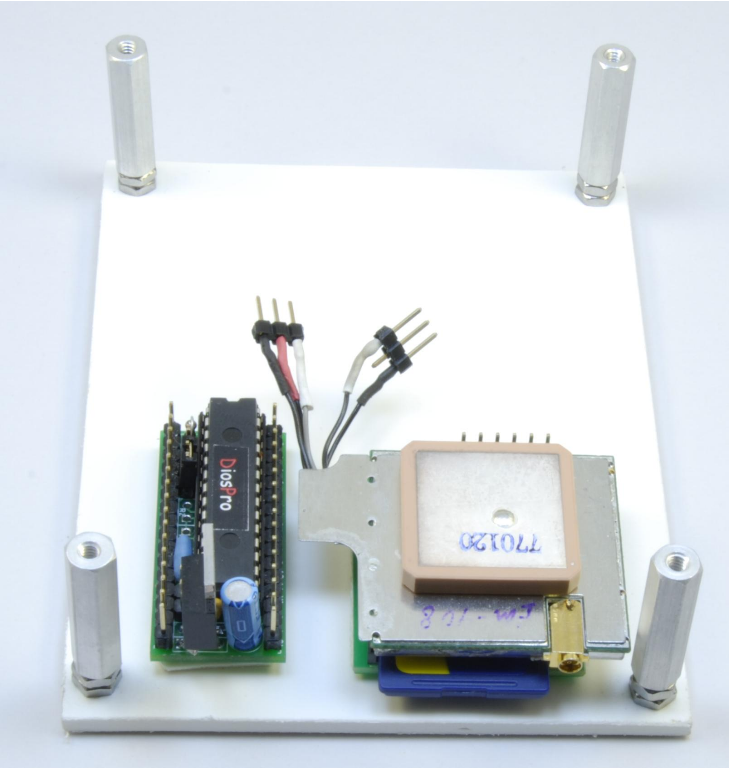


Figure 12

Step 6

Take a red LED and attach a 1K resistor as shown in Figure 13. Then cut the leads so that they can slip into the crimp pins as shown in figure 14. You can also solder the leads to the crimp pins. As an option, you may connect a small bit of heat shrink to the leads as shown in Figure 15. Repeat the step with the green LED. If you don't want to use the crimp pins you may also use a 2-pin female header instead.

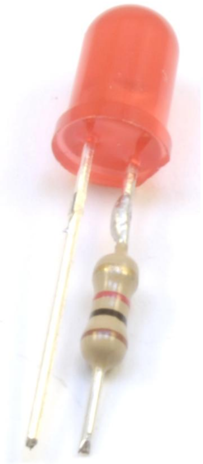


Figure 13

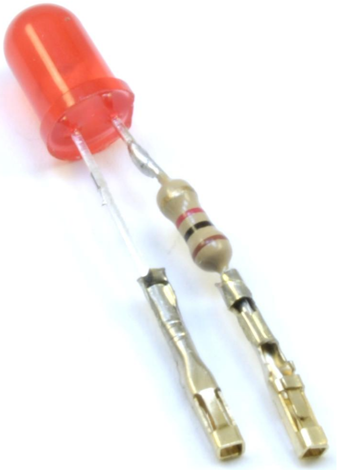


Figure 14

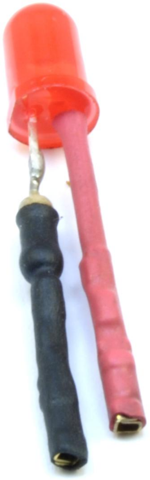


Figure 15

Step 7

Attach 2 crimp pins to a push button as shown in figure 16. You can also use a female header. The actual connection technique depends upon the type button that you use. As before, you can cover the leads with heat shrink.

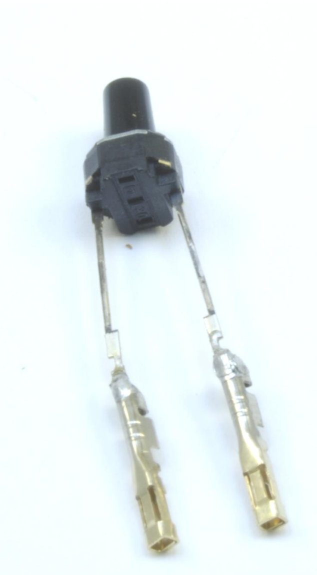


Figure 16

Step 8

Attach the red LED to Ports 0 and 1 as shown in Figure 17. Make sure the flat side of the LED is connected to Port 0. Attach the green LED to Ports 2 and 3. Make sure the flat side of the LED is attached to Port 2. Attach the button Ports 6 and 7 as shown in Figure 17.

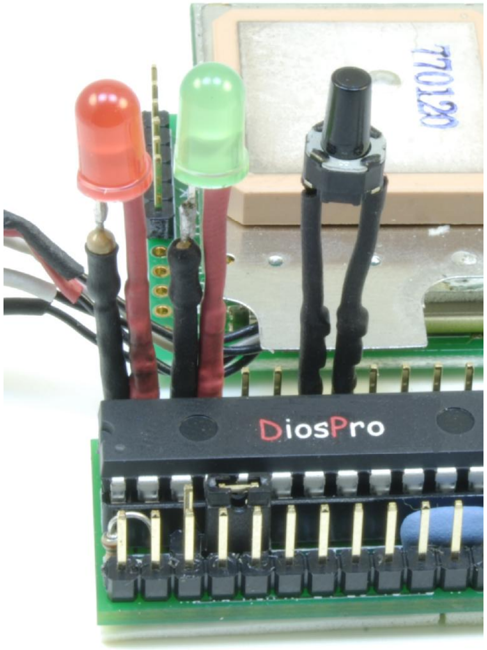


Figure 17

Step 9

You need to build a battery connector assembly. This connector will work with the 6-cell battery holder as well as a 9 volt battery. Take the battery clip and attach the ends to a small 2-pin female header as shown in Figure 18. Cut the red lead at about 3.5 inches from the clip. Strip and solder each red lead to a small toggle switch as shown in Figure 18. Attach the cable assembly to the carrier board as shown in Figure 19. Make sure the black lead faces the capacitor.

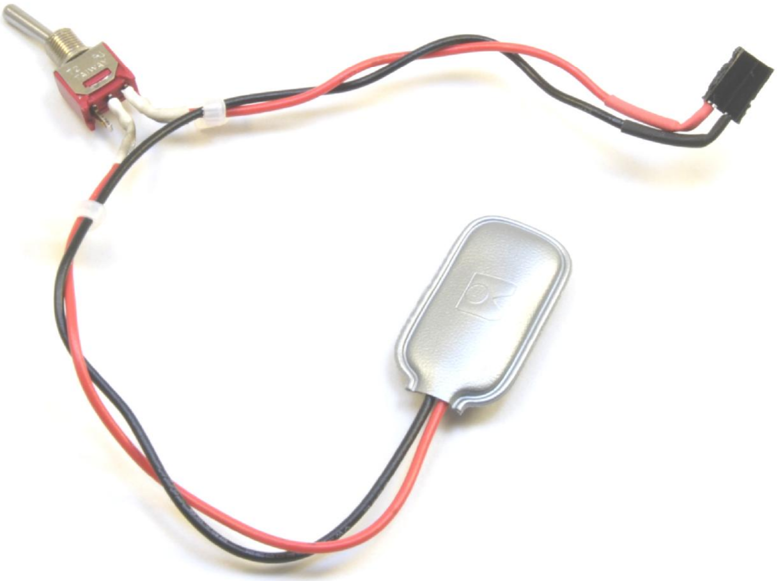


Figure 18

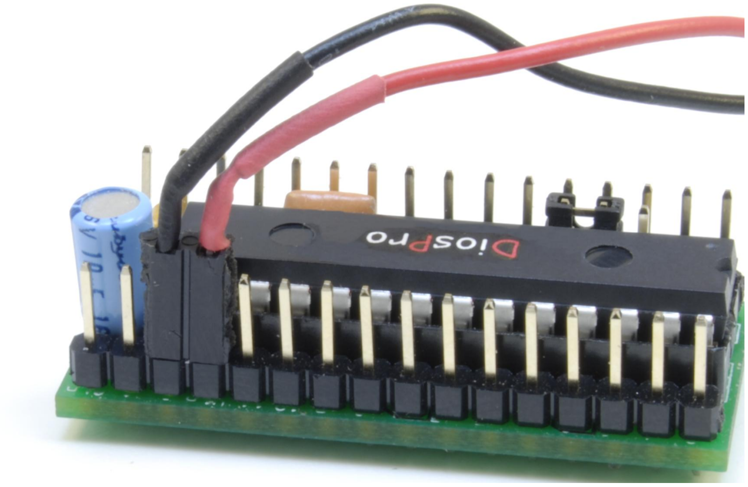


Figure 19

Step 10

At this point you can perform an LED test. But before you can do that, you need to build the EZRS232 driver. The EZRS232 driver comes with a male header so you have two choices. The first is to install a 5-pin female header in place of the included male header as shown in Figure 20. Another option is to create a female-to-female connector by attaching two 5-pin female headers together and using this as an adapter.

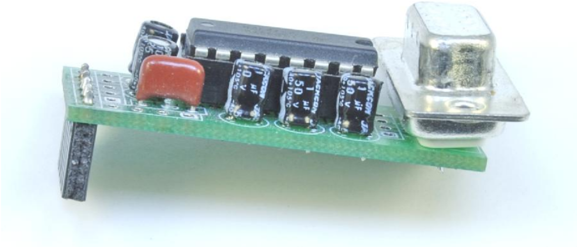


Figure 20

Step 11

Attach the battery to the battery clip and the LEDs should begin to light. This is because each DiosPro chip comes preprogrammed with a test program. Attach the EZRS232 to the Dios Carrier as shown in Figure 21, then connect the 9-pin cable to your PC and start up the Dios compiler. Load and program the included file called **DiosLEDtest.txt** into the Dios. The two LEDs should blink and then stop when the button is pressed.

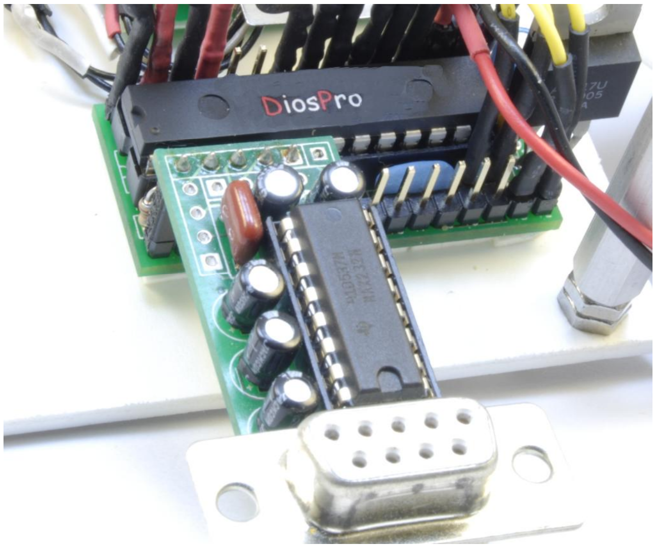


Figure 21

Step 12

I will be using some small female-to-female jumpers to connect the memory card and GPS module to the Dios carrier. These can be purchased from Shmartboard at:

http://www.schmartboard.com/index.asp?a=11&id=42

We need to create a couple of split power connectors. In order to do this, take 3 of the jumpers and cut them in half. Then connect three of the pieces to create the split jumper shown in Figure 22. You will need to make two of these jumpers.

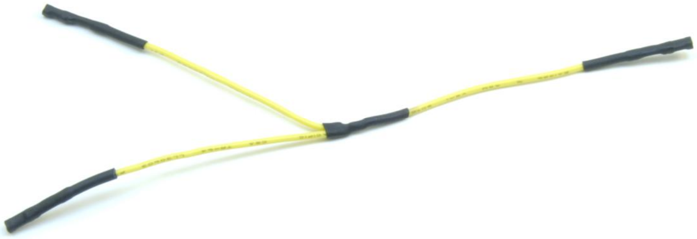


Figure 22

Step 13

Take one of the split jumpers and connect one end to the J1 Pin-2 as shown in Figure 23. Then connect one of the other ends to the GPS GND lead (black). Connect the last end to the memory board pin-5 (GND).

With the second split jumper connect one end to the J1 Pin-1 as shown in Figure 23. Then connect the other ends to the GPS VCC and Enable leads (White and Red).

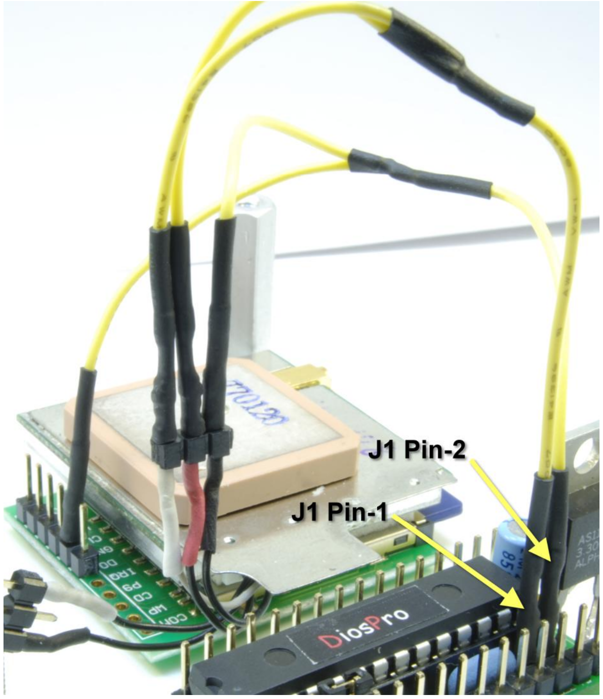


Figure 23

Step 14

Take a single 5” jumper and connect it to the VCC pin (Marked +) on the carrier and the VCC lead on the memory board (Pin 3) as shown in Figure 24.

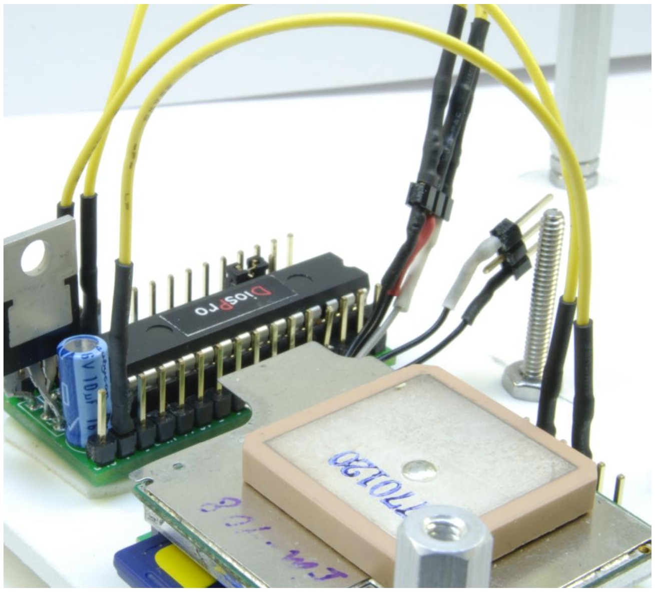


Figure 24

Step 15

For the following connections you need to refer to Schematic 1 and the Dios Carrier 1 manual to help you locate the pin location.

Connect a jumper between the GPS TX lead (black) to the carrier Port 8. Next, connect the GPS RX lead (white) to the carrier Port 9.

At this point you can test the GPS. Apply power to the Dios and attach the RS232 driver. Load and program the Dios with the file called DiosEM408test.txt that I have included. The GPS Dios should start to display NMEA information to the debug terminal.

Step 16

Make the following connections to complete the hookup:

- Connect the memory card pin-1 (CS) to the carrier Port 13.
- Connect the memory card pin-2 (DI) to the carrier Port 10.
- Connect the memory card pin-4 (CLK) to the carrier Port 12.
- Connect the memory card pin-6 (DO) to the carrier Port 11.

Now insert an SD memory card into the memory card breakout board and apply power. Then plug the EZRS232 driver into the carrier and load the program file called **DiosLogEM408.txt.**

This is the main logger program.

If the memory card is working properly you should se a message that looks something like this:

```
Memory Card Initialized
Volume Type=FAT16
Partition starts at sector 0.101
Reserved sectors 7
Each cluster has 8 sectors
Number of fat tables 2
Each FAT has 244 sectors
Root has 512 entries or 16384 bytes. 32 sectors
Fat1=0.109
Fat2=0.353
Root=0.597
Data=0.629
memory Card Initilized
```

Step 17

In order to attach the standoffs, insert a 1/2” machine screw through a hole in the bottom base, then attach two nuts to the screw and finally the 1” standoff as shown in Figures 8 and 12. This configuration will make the 6-cell battery holder fit tight inside the enclosure. If you use another kind of battery you will need to adjust accordingly.

Step 18

In order to attach the top base you will need to line up the LEDs and button to create some holes. There are a couple of ways to do this. One way is to cut out a piece of tracing paper that is the same size as the base and place it over the LEDs, then mark the LED and button positions. Before drilling the holes, add an extra 1/8” between the two LEDs.

For the LEDs I drilled 5/32” holes. The button will depend on the type used. If you use a button that mounts to the top base then you can attach wires to the leads and route them to ports 6 and 7. In this case, the button can be located in several locations.

You will also need to drill a hole for the power switch. The actual diameter will depend on the switch you use.

Attach the base to the standoffs with 4 1/2" machine screws as shown in Figure 25. You will probably have to bend the leads on the LEDs in order for them to fit properly. This is normal, and if you used heat shrink as insulation, should not present a problem.

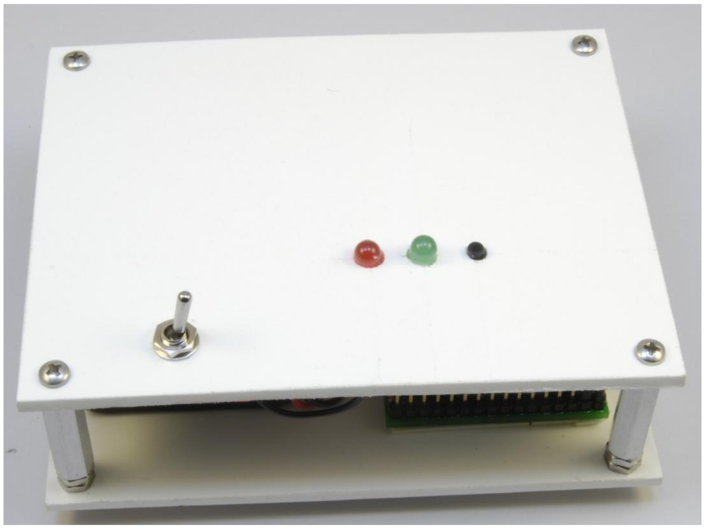


Figure 25

Smart Logger Operation

When power is applied to the smart logger it will test the memory card for a file called LOGDATA2.txt. If it is not found a 20 meg file will be created. This is just a place holder file and will be truncated as necessary when collecting data.

Startup Indicators

Red LED blinks 4 times and stops	Smart Logger is ready and in run mode.
Red and Green LEDs alternate a 1 second blink	Memory card failure.
Red LED does nothing	Battery dead
Red LED blinks 4 times then green LED flashes fast	Battery low

Fix Indicator

Green LED alternates on and off every second or so	Smart Logger is ready and in run mode with no Fix.
Green LED flickers on with one quick blink	GPS has a SPS mode 1 Fix
Green LED flickers on with two quick blinks	GPS has a DGPS or WAAS mode 2 Fix
Green LED flickers very fast constantly	Indicates battery level has dropped to low for operation
Green LED fails to light at all	Battery dead

Record Indicator

Solid Red LED	When record button is pressed the LED will light solid red until button is released. Press a second time to exit record mode.
Red LED flashed quickly	Each time a NMEA GGA or RMC messages is saved the red LED will flash quickly
Red LED not flashing	Not in Record mode.

Each line of data is actually written to a buffer and when that buffer reaches a value of 512 bytes, it is saved to the SD memory card. When recording is stopped or the battery level drops too low, the buffer is also saved to the memory card.

It is not recommended that you remove the memory card while in record mode; you could corrupt the log file. Take the logger out of record mode first. You can stop and start the record mode at will. When restarted, it will append data to the end of the log file. Note that if power is removed, the logger will start back at the beginning of the file when power is restored.

How Well Does it Work?

On a freshly charged set of 2000mah rechargeable batteries I get about 20 hours of record time. This could probably be extended if the LEDs were turned off completely. By only recording the GGA and RMC messages, we can store about 39 hours of data in a single 20 meg file. Because of the way I setup the **addtofile** function, only 65000 sectors can be written to the file on the SD memory card. This is about 33 meg. If you need to store more data then you need to add more code to flush the current file and open up a second. Another option would be to modify the **addtofile** and **setfilesize** functions to handle a larger sector counter.

I have included a sample log file that I created using the Smart Logger and I have to say it worked perfectly. The logger is small enough to fit into most robots that would be using GPS and can be made even smaller if you remove the battery and power it from your robots logic power source.

Going Further

If you look closely at the code in the DiosLogEM408.txt program, you can see that I have added my own NMEA messages. When a low battery is detected a $GPMSG,1,LowBat*7C is sent to the log. The DiosPro has several unused digital and AtoD ports so it would be possible to add additional data to the logfile. This would enable you to actually record robot control telemetry as well as sensor data.

The DiosPro supports a sleep mode so it is possible to log intermittent data. This would allow you to log days worth of data on a single charge.

While built the Smart Logger for logging GPS and telemetry data for robotic applications, there is no reason you cant use it to log positional and speed data in your automobile.

SparkFun DataLogger

I would be remiss if I didn’t mention the SparkFun data logger module. For those of you who want a pre-built data logger module, this may be what you are looking for. The GPS Logger v2.4 is a self contained module. All you need to add is 4-7 volts DC. The module has a EM406 GPS module installed on the top as shown in Figure 26. It supports up to a 1 Gig SD memory card which is inserted on the bottom as shown in Figure 27.

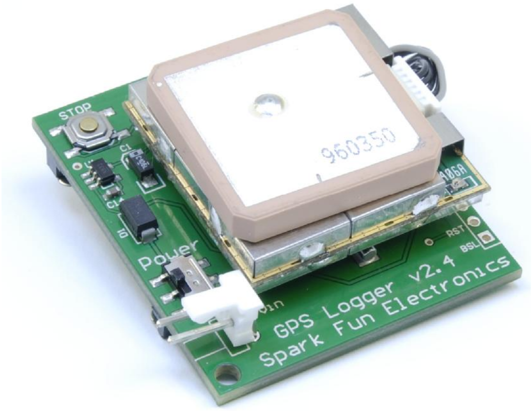


Figure 26

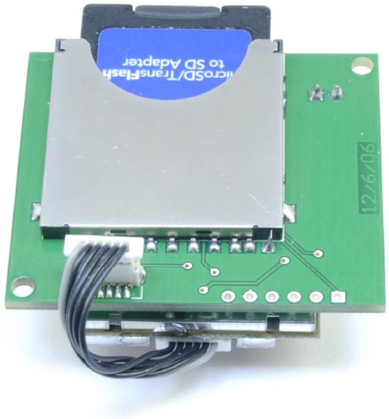


Figure 27

To power the module, I recommend a set of 2000mha rechargeable batteries and a battery holder like the one shown in Figure 28. SparkFun also sells a battery holder with a 2-pin connector that mates directly with this module. With this power source you will get just under 12 hours of operation with a steady stream of data writing to the memory card. You will get a couple days of run time if you opt for an intermittent write of once every few minutes.

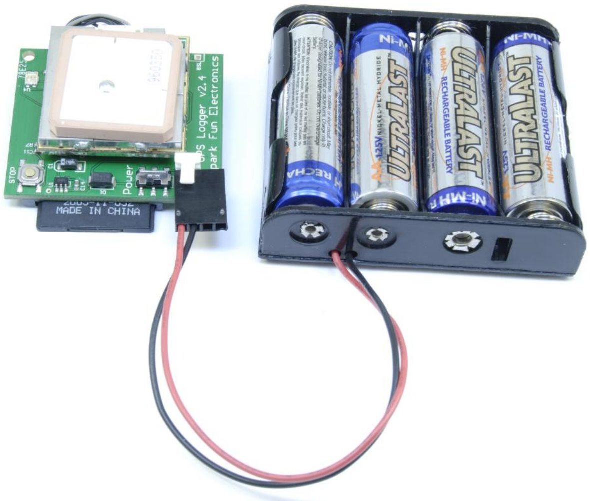


Figure 28

Once you insert a memory card and apply power, the module will create a default config file called GLOGCON.TXT as shown in Table 1. By default, the config file will be set to mode 1, which will log all data. By changing this to mode 1, you can set the time between logs, thus putting the module to sleep when it is not writing. Another setting you will want to change if you are in the US is the WAAS setting. By setting this to 1, the modules accuracy will be greatly increased.

```
Mode = 0
Log What = RMC;GGA;GSA
Time Between Logs = 00:10:00
Holdoff = 5
WAAS = 0
Max Time to Lock = 300
```

Table 1

If you decide to use the SparkFun module here are a few tips:

- The module does not monitor the battery so it's up to you to make sure you don't drain the batteries completely. If you do, you will probably corrupt the card and it will require reformatting. For long term logging you will want to use the intermittent write method mentioned previously.
- You need to hit the small button on the module to stop the logging operation. This can be a problem if you mount the module in any kind of enclosure. You can connect an external button to the two outside leads on the button. This way you can rout them to any location on your enclosure.
- To mount the module to an enclosure, place some double stick tape on the top of the GPS antenna and stick the module to the underside of the top of your enclosure.
- If the power connecter also gets in your way, try removing the connector completely and soldering a couple of wires to the bottom of the board.

Final Thoughts

All in all the SparkFun Data logger works pretty well and presents itself in a very small package. The output on the SD memory card is NMEA 0183 so you won't have any compatibility issues. If, however, you want total control over the output and the ability to log some of your own messages such as AtoD, you will want to stick with the GPS Smart Logger. The cost is about the same for both systems.

Be sure to check for updates and downloads for this article at:

http://www.kronosrobotics.com/Projects/GPSLOG.shtml

Parts

The following is a breakdown of the source for all the components referenced in this project.

Spark Fun Electronics

EM-408 GPS Module
http://www.sparkfun.com/commerce/product_info.php?products_id=8234

DF-MMC Breakout Board
http://www.sparkfun.com/commerce/product_info.php?products_id=204

9-Pin Serial Cable
http://www.sparkfun.com/commerce/product_info.php?products_id=65

External Antenna with SMA connector (Optional)
http://www.sparkfun.com/commerce/product_info.php?products_id=464

SMA to MMCX adapter cable (Optional)
http://www.sparkfun.com/commerce/product_info.php?products_id=285

Optional

GPS Logger v2.4
http://www.sparkfun.com/commerce/product_info.php?products_id=8237

ShmartBOARD

5” Yellow Jumper
http://www.schmartboard.com/index.asp?a=11&id=42

KronosRobotics

EZRS232
http://www.kronosrobotics.com/xcart/product.php?productid=16167

DiosPro 28 Chip
http://www.kronosrobotics.com/xcart/product.php?productid=16429

Dios Carrier 1
http://www.kronosrobotics.com/xcart/product.php?productid=16170

800ma 3.3V Regulator
http://www.kronosrobotics.com/xcart/product.php?productid=16565

6 Cell Battery Pack
http://www.kronosrobotics.com/xcart/product.php?productid=16321

36 Pin Female Header
http://www.kronosrobotics.com/xcart/product.php?productid=16291

Female Crimp Pins
http://www.kronosrobotics.com/xcart/product.php?productid=16261