Portable Netduino GPS with LCD Display

R.Grokett Kinetic Designs IoT

2016.03.08

Overview

I found an unused Netduino V1 card in my parts box, so wanted to find something I could dedicate it to. It was already connected to a Nokia 5110 LCD display for an earlier uncompleted project. Powering it up, I found both were working fine. Just a few days earlier, I had seen a low-cost GPS module on Amazon, so decided to build a portable GPS unit.

Parts List

- Netduino V1 or higher
- Andoer NEO-6M GPS Module or similar
  http://amzn.to/1TIUOyx
- Nokia 5110 LCD display with 8 pin interface
  http://amzn.to/1Lc2qqe
- Visual Studio 201X and Netduino SDK software
  http://www.netduino.com/downloads/
- GPS software from GITHUB:
  https://github.com/rgrokett/Netduino-GPS
- USB Cable
- 5V battery with USB connector for portable operation.

Software

I based some of my inspiration from:

"GPS using the Netduino"
http://www.codeproject.com/Articles/795474/GPS-using-the-Netduino

I also had run across the project :

"Graphic LCD 84×48 and Netduino Plus"
https://atoussaint.wordpress.com/2012/06/20/graphic-lcd-84x48-and-netduino-plus/

This gave me a big boost on coding a GPS reader with LCD display in Microsoft C#. The software for my project is based on that code.

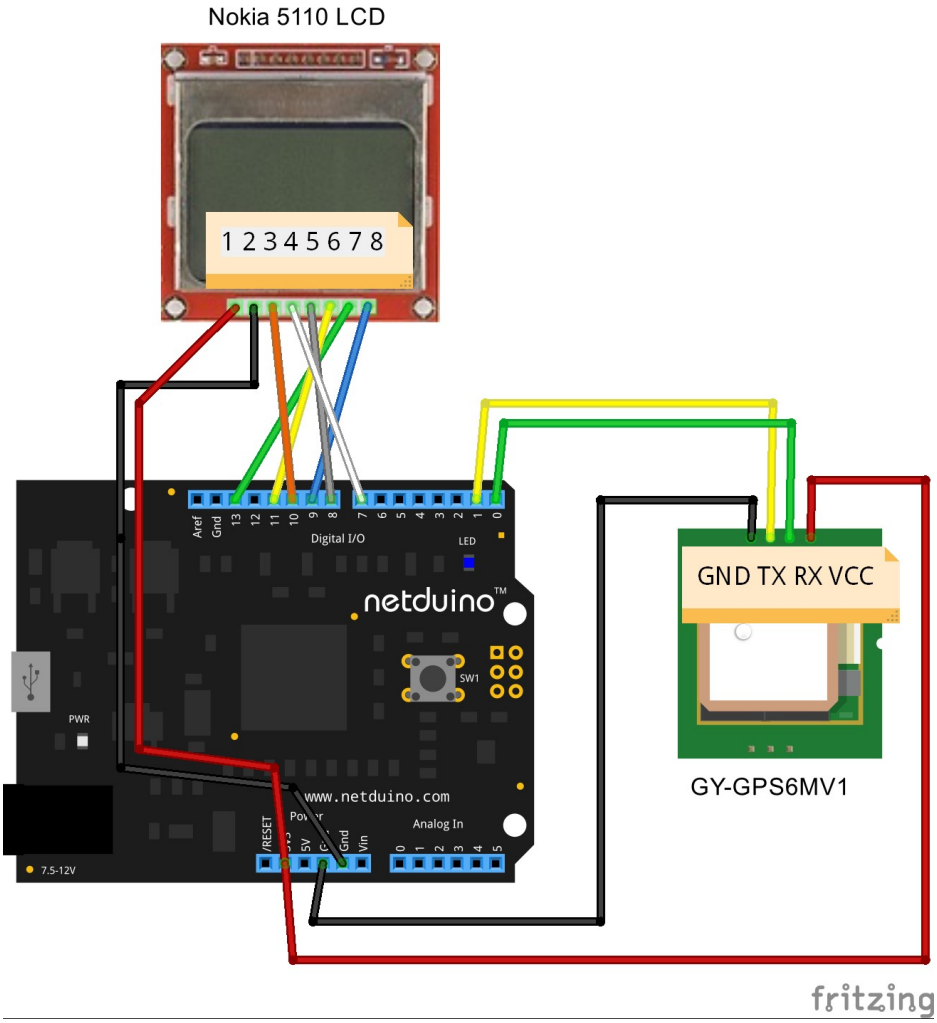
I have only used Visual Studio a couple times, so had to work my way through it. I suggest downloading the versions of software from www.netduino.com for your particular Netduino board. I think the C# code for this project will work with most any version.

NOTE: Be sure you can code and upload a simple program, such as an LED blinker, before attempting this project. I also suggest testing your Nokia LCD using the "Graphic LCD 84×48 and Netduino Plus" link above.

Hardware

Connect the Netduino to the Nokia 5110 LCD and the GPS Module using below. Note that different versions of each of these hardware elements may have DIFFERENT PINOUTs than below, so I strongly suggest verifying your pinouts with data sheets before connecting!

Also, if you try adapting this to an Arduino, remember that it's Digital I/O pins are 5v and not 3.3v so ZAP, if you forget to translate the power! (See Adafruit's Learning site for GPS projects for Arduino!)

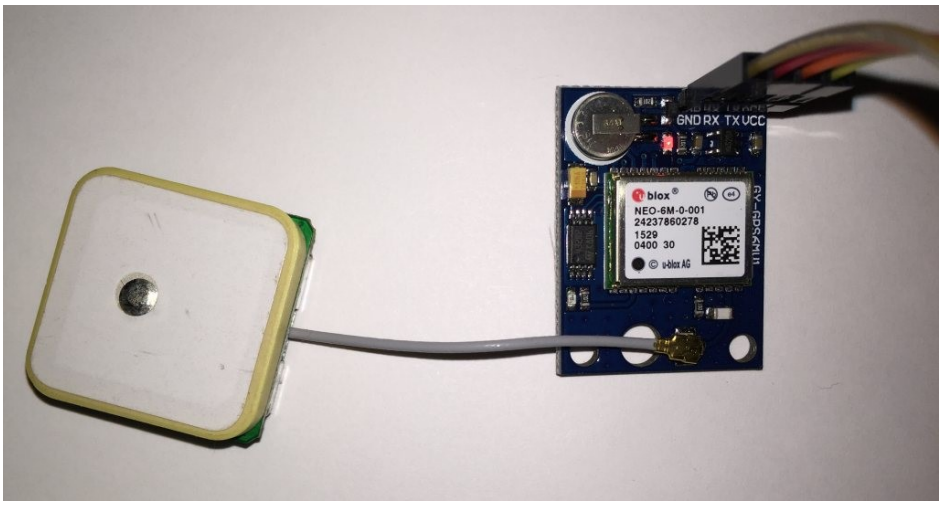


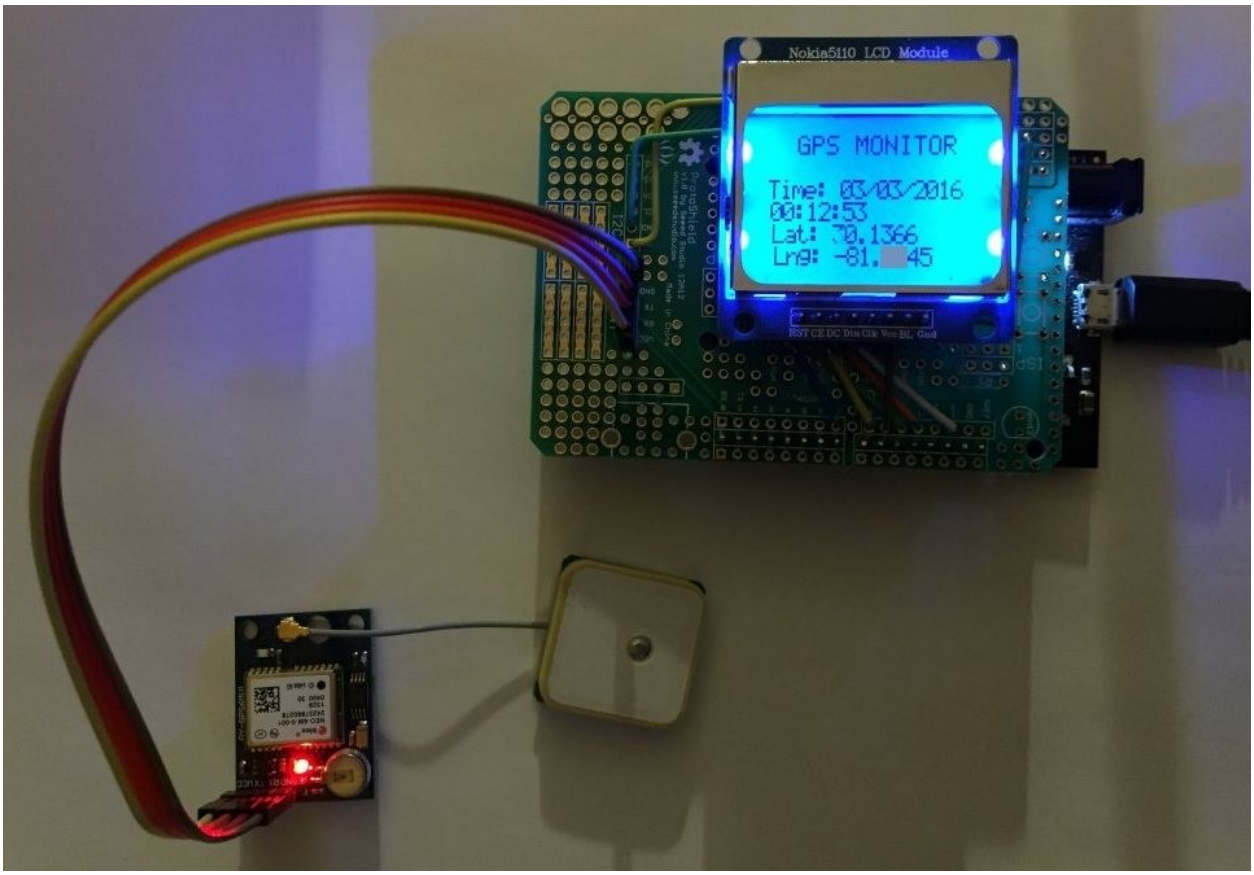
GPS Module

GPS Pin	Description	Netduino
1- GND	Ground	GND
2- RX	Receive	Digital I/O Pin 0
3- TX	Transmit	Digital I/O Pin 1
4- VCC	3.3v Power	3.3V

Nokia 5110

LCD Pin	Description	Netduino
1 – VCC	Power	3.3V
2 – GND	Ground	Gnd
3 – CE	Chipselect; any GPIO pin of the Netduino	Digital 10
4 – RST	Reset; any GPIO pin of the Netduino	Digital 7
5 – D/C	Data/Command switch; any GPIO pin of the Netduino	Digital 8
6 – DN(MOSI)		Digital pin 11
7 – CLK		Digital pin 13
8 – LED	LED: Backlight; any PWM pin of the Netduino	Digital 9





My Completed Unit in operation

Code

GITHUB: https://github.com/rgrokett/Netduino-GPS

The code (excitingly called "Program.cs") requires the "Nokia.cs" file added to it. I suspect you could use a library that supports Nokia 5110, but you would need to refactor the Lcd commands to fit it.

The folder called "GPS" from GITHUB contains the C# files needed for this project.

After loading into Visual Studio and compiling in **Debug** mode, and running, you should see a display on your Nokia LCD as well as lots of DEBUG messages from your GPS unit.

***NOTE**: If you do not see GPS messages, but only the first two lines (bold below), then you may have the TX/RX leads reversed on your GPS module.*

***NOTE**: GPS may not be able to acquire a satellite fix from inside your house. You'll need to go outside. The LCD will NOT display anything until the GPS can get a fix.*

If you get no display at all on your Nokia, but see the Serial debug messages below, take a look at project:
https://atoussaint.wordpress.com/2012/06/20/graphic-lcd-84x48-and-netduino-plus/

You are looking for GPRMC lines of data from your GPS. This line includes coordinate location data.

Sample GPS Serial output - GPRMC line

```
GPS thread started...
Main...
GOT $GPRMC LINE
GpsPoint Parse
$GPRMC,010916.00,A,3108.19303,N,08245.86442,W,0.339,,260216,,,A*62
GOT $GPRMC LINE
GpsPoint Parse
$GPRMC,010917.00,A,3108.19311,N,08245.86430,W,0.821,,260216,,,A*67
GOT $GPRMC LINE
GpsPoint Parse
$GPRMC,010918.00,A,3108.19291,N,08245.86433,W,0.444,,260216,,,A*6D
GOT $GPRMC LINE
```

The **GprmcParser** class Parse() handles parsing the comma separated GPRMC string from the GPS. If your GPS module outputs a different format than above, then you will need to refactor the parts[] array for your module.

The **GeoDistanceCalculator** class is called but isn't used in my modified version. I had to refactor it a bit in order to work with the version of Netduino SDK I had to use. It's too cool to remove this code as it could be useful for future projects, but if necessary, you could remove it (and its related method calls).

In this version of the program the minDistanceInMilesBetweenPoints is set to zero, which causes the GPS code to update constantly. Changing this value from 0.0 to 1.0 would cause the display to update only once every mile(!) of change, as based on that GeoDistanceCalculator class.

```
Reader gpsShield = new Reader(serialPort, 100, 0.0);
```

Please feel free to modify this code for any features you would like, as I was just interested in integrating the GPS module with the Nokia LCD and running on a battery for a portable GPS hack!

Thanks go to Bob Cravens on his excellent efforts for the original Netduino/GPS code!