A GPS 24 h Clock Project

The GPS Clock project describes a highly accurate 24 h clock using a UBLOX-NEO6M GPS module, a JHD162A 2x16 LCD display and an Arduino Nano micro-controller board. The parts are readily available from internet stores at reasonable cost. The clock shows the local time on the first line of the LCD display in a 24 h format and shows the date on the second line of the display. The project is easy to build and the executable file (Intel hex format) is provided for uploading to the micro-controller's flash memory. The project source code files are included.

Date: 2016 July 24

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

Table of Contents

1 Introduction.....3
2 Parts list.....5
3 Construction details.....7
4 Putting the project together.....8
4.1 Assembly steps.....8
4.2 Burning the m328-GPS-clock.hex program.....11
4.3 Powering-up the GPS Clock project.....13
5 Configuring a new Time Zone setting.....15
5.1 PuTTY Configuration.....15
5.2 Time Zone setup program.....18
5.3 Adjusting for Daylight Saving Time.....20
6 Todo list and Observations.....21
6.1 Observations.....21

Copyright © 2016 - Francis Lyn

1 Introduction

This project describes a simple GPS based clock with time and date shown on a 2 line by 16 character LCD display. Time is shown in a 24 h format, corrected for local time based on the entered time zone offset for the clock's location. The date is shown in dd-mm-yyyy format. The date is not corrected for local time zone and is based on the UTC (Coordinated Universal Time) date. The clock runs off an externally supplied DC power source (+7 to +12 V at VIN pin recommended). The GPS clock project is relatively easy to build and is completely stand-alone in operation. The clock has a default time zone of -5 h, a value chosen because the clock was tested in this zone. The default time zone can easily be changed and stored in EEPROM by the time zone setting sub-program which is included with the main program.

The GPS clock is based on the UBlox NEO6M GPS module, an Arduino Nano micro-controller board, and a JHD162A LCD display module (2x16) with an attached Backpack TWI interface board to support serial communications with the Nano controller board. All the hardware components used for this project are readily available from numerous sellers on eBay and other on-line retailers.

The software for the GPS clock project is provided as an executable file (m328-GPS-clock.hex), ready for uploading to the ATmega328P chip on the controller board. Program source code is included. A method for programming the ATmega328P chip using a USB ISP programmer and the AVRDude software tool is included.

The NEO-6M GPS module is a complete GPS receiver module with 3.3 V Rx and Tx input/output signals. The Tx output pin on the GPS module transmits several NMEA “sentences” once a second. This output is connected to the Nano board's Rx input pin. The RX input on the GPS module is not used in this application and is left unconnected.

The Arduino Nano board is used for this project because of its small footprint, low cost, on-board 5 V supply, ISP header for programming, and USB serial interface. All of these extra features are used in the GPS clock project.

The clock program scans the incoming burst of NMEA “sentences” for a specific $GPRMC sentence. An example of this sentence is shown here:

"$GPRMC,213226.00,V,,,,,,,250616,,,N*7D"

The $GPRMC sentence is the "Recommended minimum specific GPS/Transit data", and is identified by the $GPRMC header. This sentence contains several fields including the time (213226.00) field and the date (250716) field. All NMEA sentences consists of ASCII printable characters. The clock program extracts the time and date fields on the fly and saves the ASCII encoded data in internal RAM buffers.

The received time and date ASCII encoded string data in the RAM buffers is sent to the local LCD for display. The same time and date data is transmitted to the controller's USB serial port for display on a terminal emulator program running on a host computer. This secondary terminal screen display is optional and is not required during stand-alone operation of the clock.

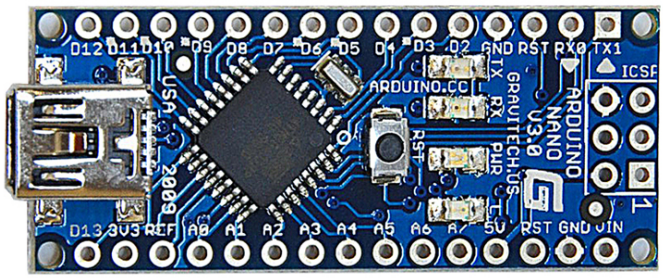
The finished GPS clock in a Hammond plastic utility case.

2 Parts list

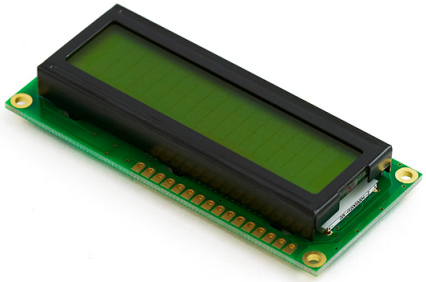
You need the following parts for the clock-timer project:

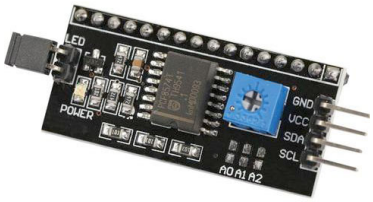
1. Arduino Nano controller board, 16 MHz clock, with ATmega328P microcontroller chip.
2. uBLOX-6M GPS module, type GY-GPS6MV2. This module runs on 5 V supply. Only the TX signal is connected to the Nano controller board in this application.
3. 5 V power supply (a cell phone charger connected via the USB socket will work). You can also use +7 V to +12 V (recommended) power supplied to the VIN pad to power the clock.
4. JHD 162A 2x16 LCD display module with a Backpack TWI interface board.
5. Miscellaneous hardware:
    1. Small piece of perforated prototyping board (main circuit board).
    2. Two 4 pin headers, two 4 conductor ribbon cables with connectors on both ends to mate with pin headers.
    3. One 2 pin header and a shorting jumper.
    4. Two 10 k ohm pull-up resistors.
    5. One 47 uF/16 V tantalum or electrolytic capacitor.
    6. Wire-wrap wire, plastic project case, screws & nuts, spacers etc.

The uBLOX-NEO-6M GPS module



The Arduino Nano controller board





LCD Display and Backpack TWI module

3 Construction details

The clock consists of a main board circuit on which is mounted the Nano controller board, two 4 pin strip headers, one two pin strip header, two 10 k ohm pull-up resistors and a 47 uF/16 V tantalum bypass capacitor. Use a small piece of perforated prototyping board, preferable one with solder pads arranged on a 0.1 x 0.1 inch grid pattern.

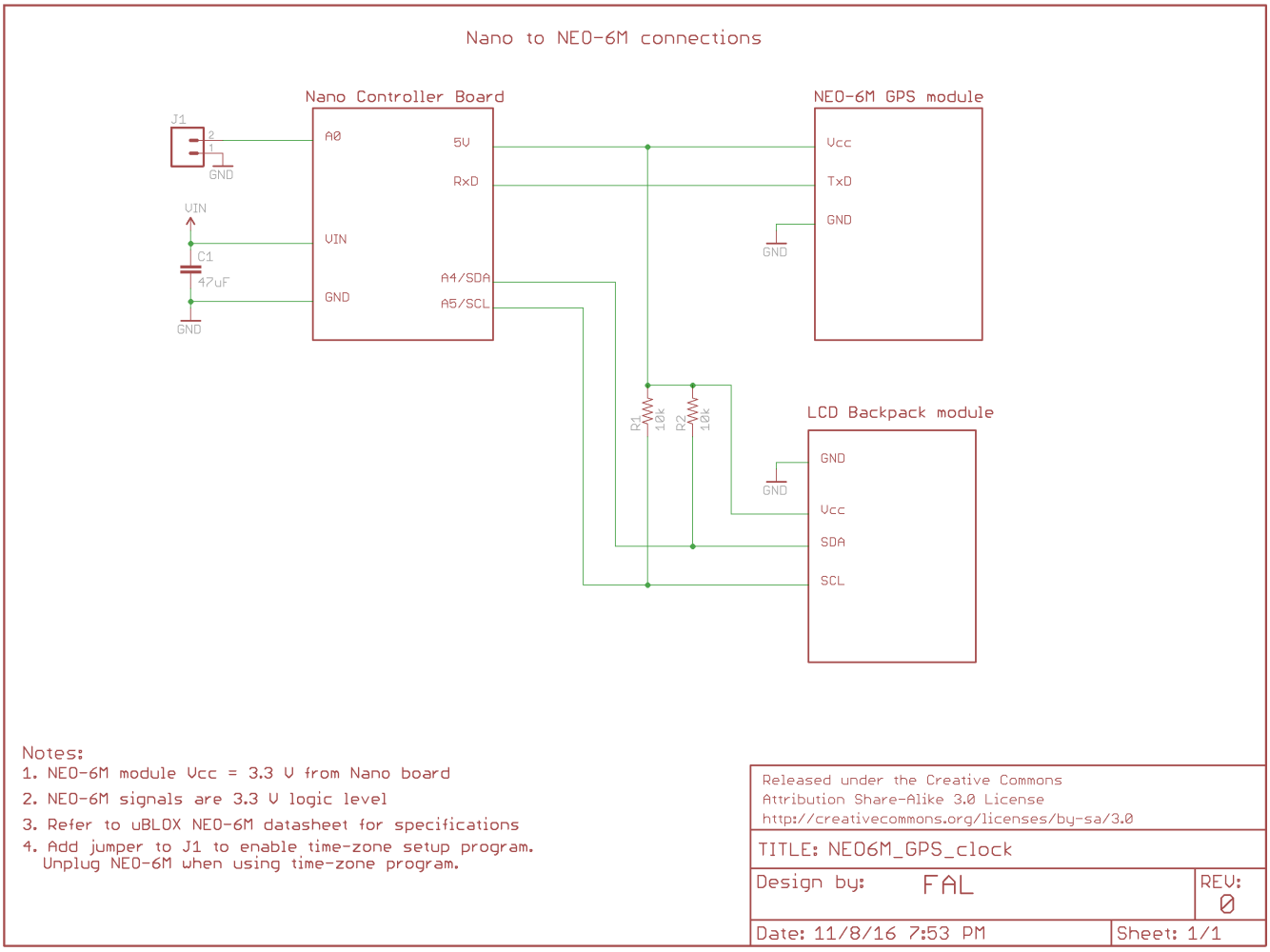
The NEO-6M is a 3.3 V device and the Tx and Rx signal pins operate at a 3.3 V level. The NEO-6M module operates with a 5 V Vcc power supplied from the Nano's on-board 5 V pin.

The Rx and Tx I/O on the Nano are 5 V compatible. The Tx output from the NEO-6M can drive the Rx input on the Nano since this input can accept the 3.3 V level output from the NEO-6M.

Refer to the GPS-clock-diagram for connection details between NEO-6M and the Nano controller board. Only three connections are required for Vcc, GND and TX. These connections should be made using a ribbon cable with connectors on both ends that mate to matching pin headers on the NEO-6M and the prototyping board on which the Nano controller board is mounted. You need to be able to disconnect the NEO-6M from the Nano controller when you wish to run the time zone setup sub-program routine.

4 Putting the project together

The connection diagram below shows the connections between the Nano controller board, the NEO-6M module and the JHD 162A LCD Backpack TWI module.



4.1 Assembly steps

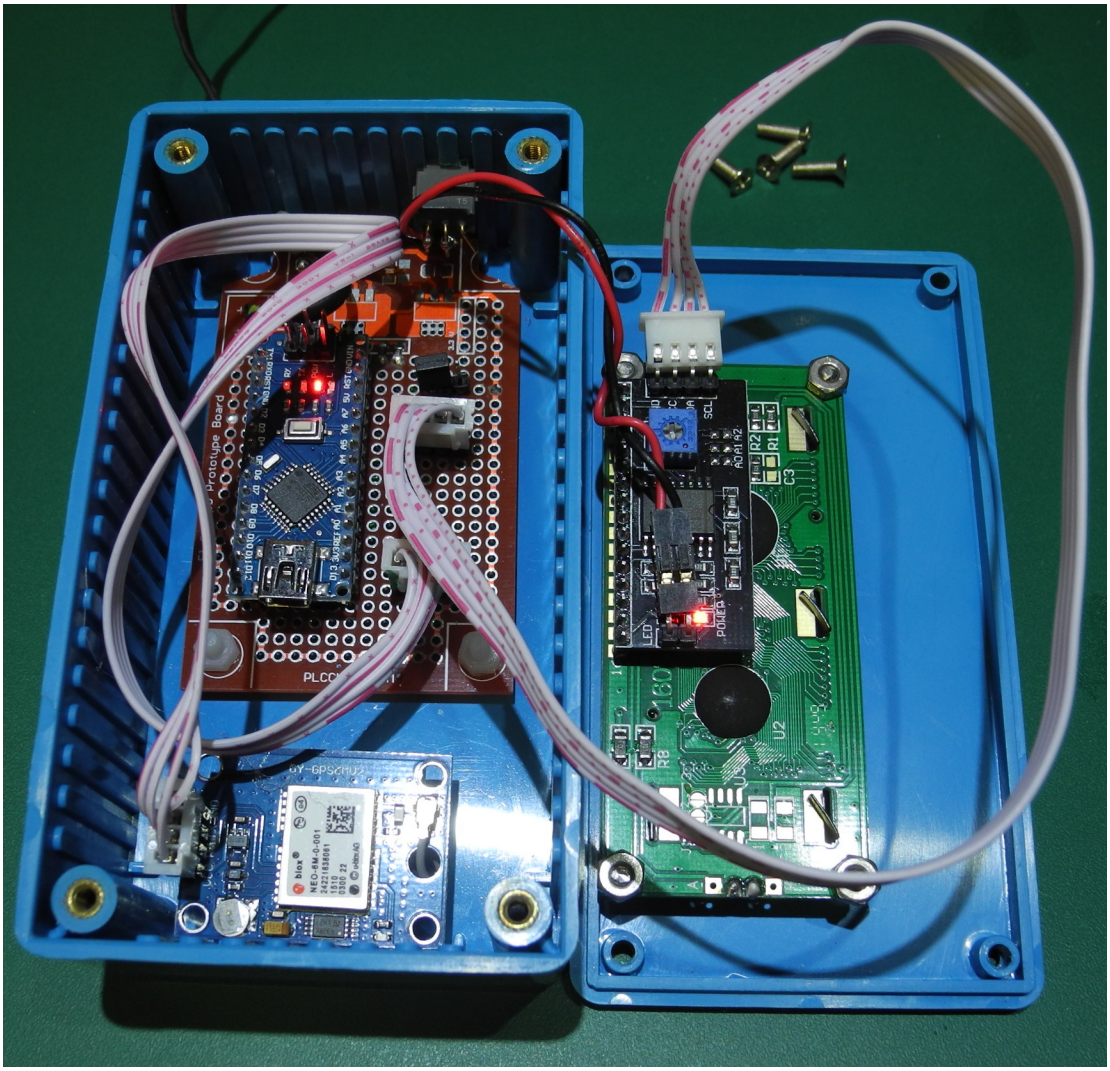
In the prototype project the Nano board is mounted on a small prototyping circuit board with a grid pattern of solder pads. You can see the arrangement in the pictures. 30 AWG kynar insulated wire-wrap

wire is used for the connections on the circuit board. Strip headers are mounted on the circuit board to facilitate connections to the NEO-6M and LCD Backpack modules.

When you have completed the wiring of the components and modules, carefully check the wiring against the connection diagram again and make sure there are no wiring errors. Look especially at the +5 V power and ground wiring as this type of check-out is one of the easient ways to avoid experiencing a “smoke” event.

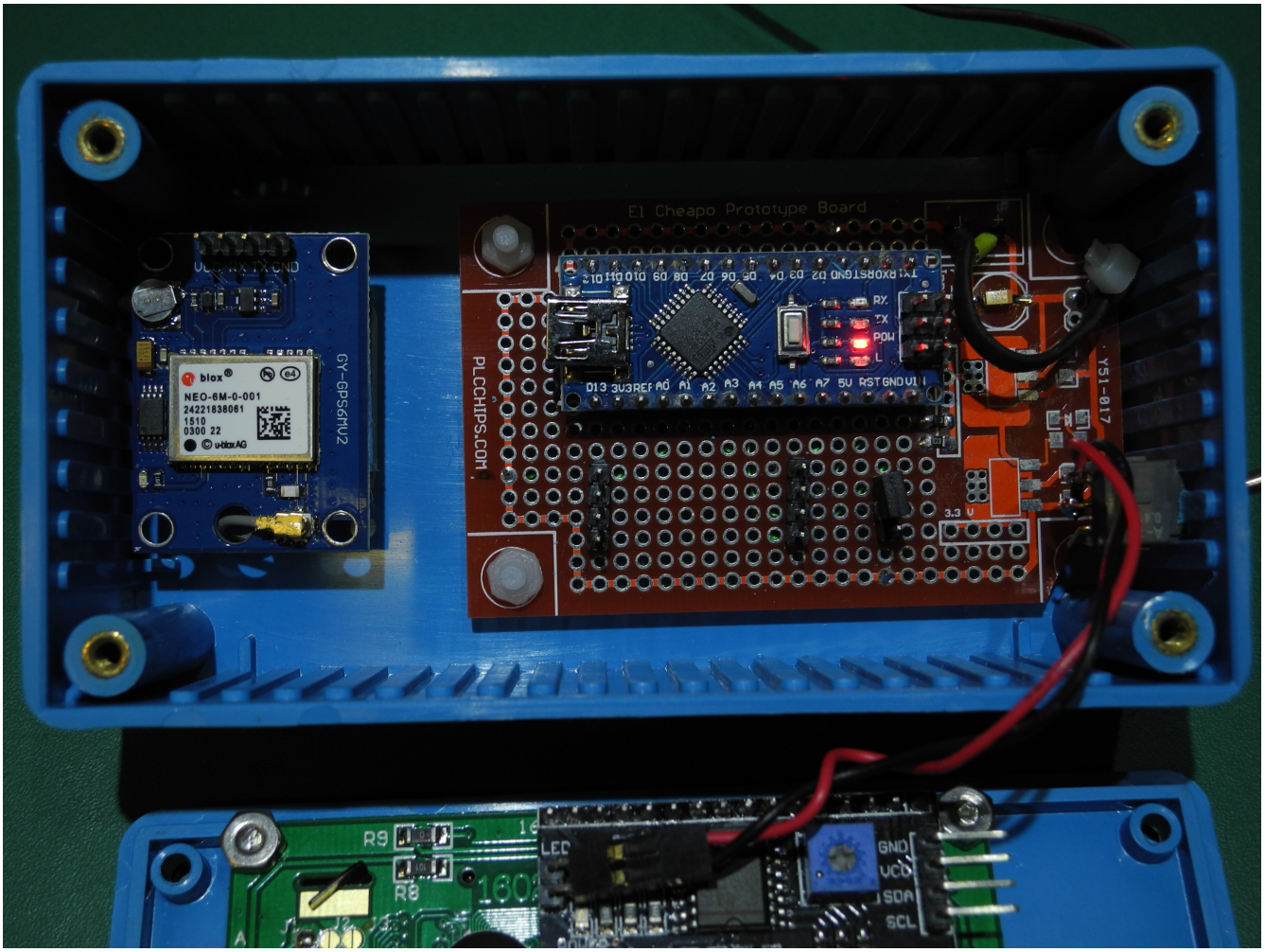
Do not apply power to the Nano board as yet. At this stage you are now ready to upload the m328-GPS-clock.hex file you downloaded from the PLCCHIPS.com web site, to the Nano's flash memory. The programming method is described in the next section. You can leave the NEO-6M and LCD Backpack modules connected during the programing procedure.

The photograph of the internal view of the clock with the two 4 conductor ribbon cable/connector assemblies connecting the NEO-6 module and LCD/Backpack modules to the main board is shown below. The red and black wires from the Backpack module goes to a small toggle switch that is used to manually switch the LED backlight on and off. You can see above the LCD/Backpack 4 pin header the black 2-pin shorting jumper. The jumper is plugged in only on one pin, i.e. <u>not</u> in the shorting position, for normal clock operation.



Internal view of GPS clock with connectors in place.

The photograph below shows a close-up view the clock modules, without the ribbon cables installed. You can see the black and red wires to the SPDT toggle switch that replaces the shorting jumper on the LCD/Backpack for the backlight LED control. Note this switch is not shown in the GPS Clock schematic diagram as it is optional. You can install or remove the shorting jumper to enable/disable the LCD backlight LED and omit the toggle switch.



4.2 Burning the m328-GPS-clock.hex program.

The programming method used to upload the executable hex file to the Nano's flash memory uses the AVRDude programming software and a USB-ISP-programmer avaialable from many on-line retailers. The Arduino IDE is not used in this method of programming, nor is the boot-loader supplied with the Arduino Nano board.

First, download and install on your PC the AVRDude software. There are numerous excellent articles on the web describing how to install and configure AVRdude so we'll skip repeating the same thing here. If you have trouble finding the software, take a look at this site:

http://winavr.sourceforge.net/download.html

The tricky part to programming the Nano board with AVRDude is to invoke the proper command line instruction. We'll get to this in a moment.

First, you need a USB-ASP-Programmer, a typical one is shown in the picture below. You should also get a 10-pin socket to 6-pin plug adapter to make it easier to connect the 10-pin ribbon cable from the programmer to the 6 pin ISP header on the Nano board. The USB programmer is recognized by AVRDude as programmer type usbasp. See the following site for further details of this programmer.

http://www.fischl.de/usbasp/

We'll assume you are running in a Windows environment and that you placed the m328-GPS-clock.hex program image file in a sub-directory such as:

```
c:\Users\yourname\My Documents\avr\m328-GPS-clock.hex
```

When you invoke AVRDude be sure to specify the full path to the directory where avrdude.exe resides. We'll asume for this discussion that the avrdude files are in:

```
c:\opt\avr\avrdude.exe
```

The avrdude.conf and avrdude.rc files are also in this sub-directory.

When you are ready to program the Nano's flash with m328-GPS-clock.hex executable image, plug the programmer into a USB port on the PC and plug the ribbon cable with the 10-to-6 pin adapter onto

the 6-pin ISP programming header on the Nano board. Make sure you connect the adapter the right way round (adapter socket 1 connected to ISP header pin 1). The programmer will supply 5 V power to the Nano board and the power indicator LED will light up.

Open up a command line interface in a Windows terminal and go to the subdirectory c:\Users\yourname\My Documents\avr\ where the m328-GPS-clock.hex file is located. Type in the following command:

"c:\opt\avr\avrdude.exe" -C"c:\opt\avr\avrdude.conf" -p m328 -c usbasp -u -U flash:w:m328-GPS-clock.hex:i

AVRDude will perform a number of programming steps and print some messages on the console screen while it uploads the m328-GPS-clock.hex file to the Nano's flash memory. If the programming is successful, you'll see the program verification step completed.

As a side note to AVRDude use, a very useful companion program to have in your toolbox is the AVRdudess, a GUI for AVRDude. Check out this link if you are interested:

http://blog.zakkemble.co.uk/avrdudess-a-gui-for-avrdude/

Unplug the programmer from the Nano. Now go to the final step of powering up the GPS clock project in the following section.

4.3 Powering-up the GPS Clock project

Once the Nano controller is programmed with the m328-GPS-clock.hex executable file and the NEO-6m and LCD Backpack modules are connected to the prototype board with the Nano controller, you can apply power to the Nano controller board via the mini USB cable and plug the other end of the cable into a host computer, or to a cell phone charger. Later on for stand-alone operation, you can power the clock from an external 7 to 12 V power supply connected to the Nano's VIN pin.

When the GPS clock is first powered up it will take a minute or more for the GPS module to acquire the overhead satelites before it starts displaying the GPS derived time and date information on the LCD

display. During this satelite acquisition phase, you may see some unrecognizable characters displayed. When the satellites are acquired, the display will begin to show the local time using the default -5 h time zone location.

If you are in a different time zone to the default time zone, you will need to invoke the time zone setup part of the GPS clock program to enter a new time zone setting so that your local time display is correct. This procedure is described in the next section below.

5 Configuring a new Time Zone setting.

The time zone setup sub-program is invoked whenever you need to change the time zone offset. You can set the offset within the range of -14 to +12 h. If you use the clock in a country that observes daylight saving time, you can change the offset to effect daylight saving time display.

The time zone setup sub-program requires a terminal emulator program running on a host PC (Windows based) to act as a terminal interface for the setup program. A good program to use is PuTTY, an open source SSH and Telnet client available from:

http://www.putty.org/

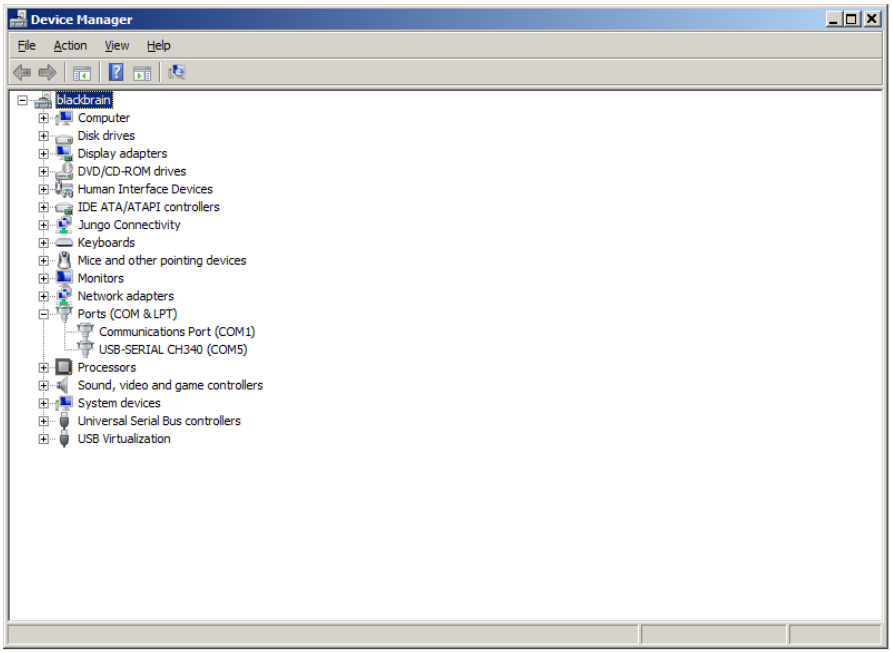
Download and install PuTTY on your PC and go through the configuration procedure to allow PuTTY to be used with the clock setup program.

5.1 PuTTY Configuration

Connect the Arduino Nano board to a PC USB port for the set-up procedure.

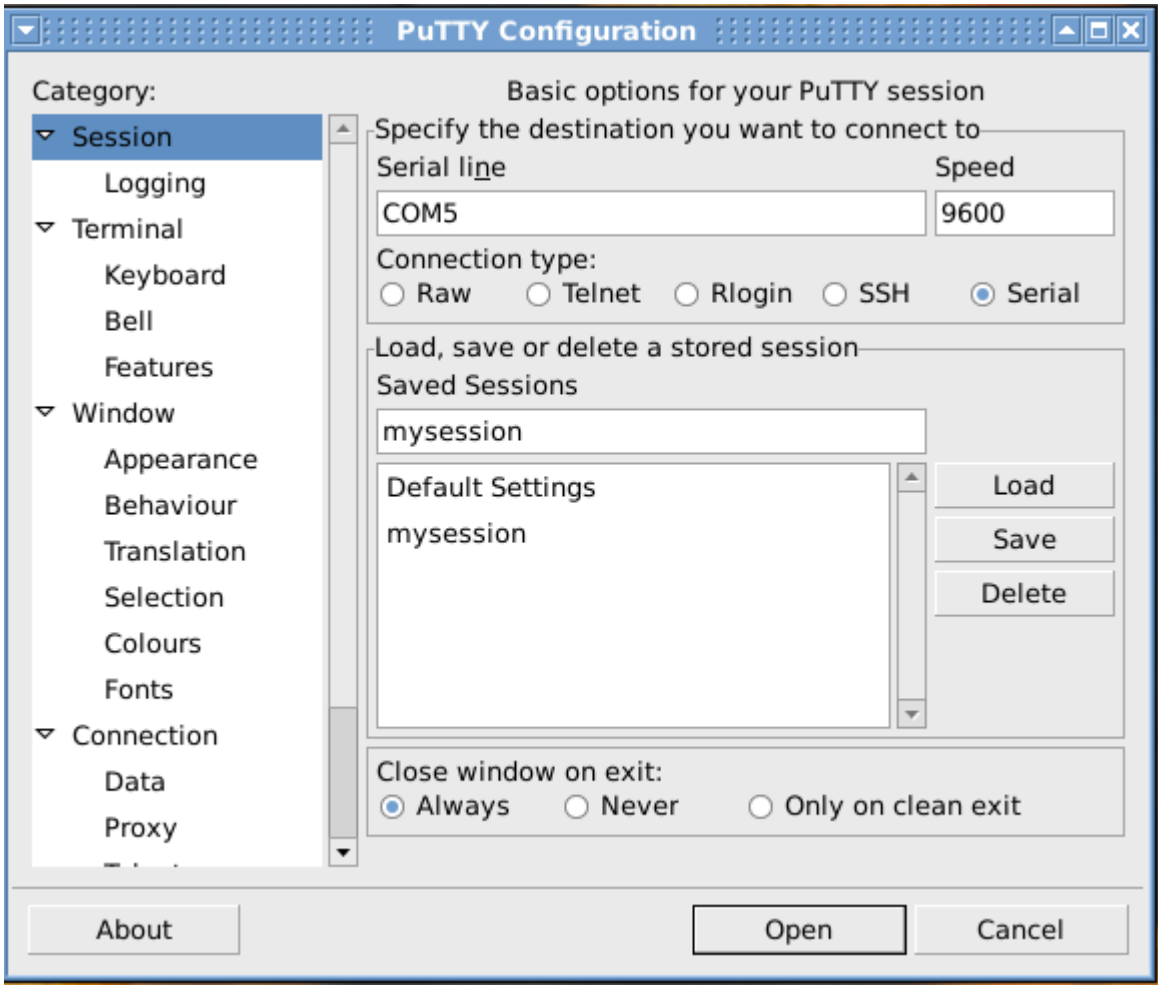
The first thing to do is to install the required USB device driver for the Nano controller board. You need to get and install the correct driver from the Arduino web site. Check other popular Arduino web sites like Lady Ada, Sparkfun, etc. for advice on setting up the Arduino board serial port driver.

You now need to determine which serial COM port the OS assigned to the plugged-in controller board. To find out which COM port is assigned, open the Windows Control Panel, select Hardware and Sound, and select Device Manager. You should see a screen like the following:

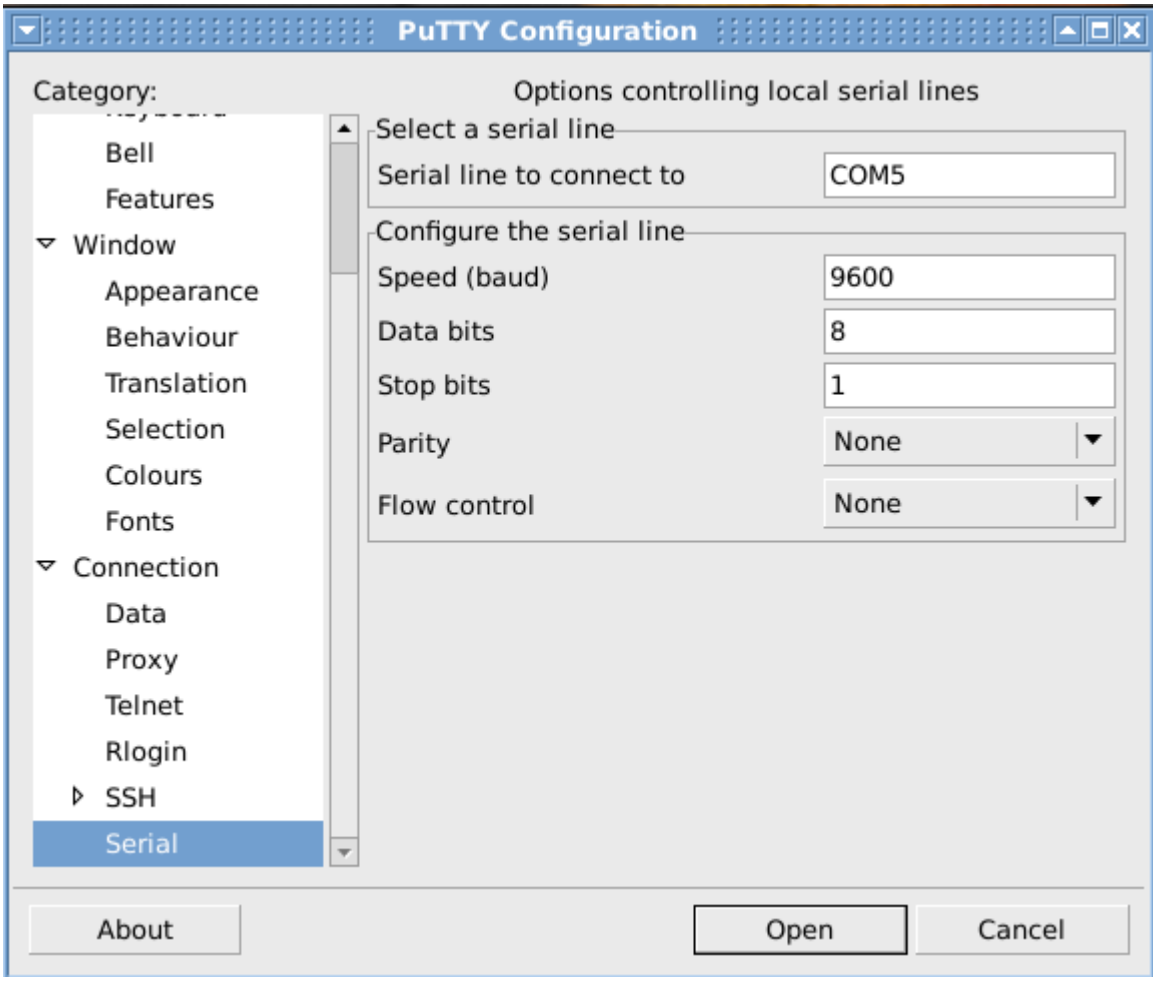


Under Ports (COM & LPT) folder you see USB-SERIAL CH340 (COM5) listed, which identifies COM5 as the assigned port for the Arduino Nano board. Close the Device Manager program and start PuTTY.

The PuTTY configuration screen appears when you start PuTTY. Modify the Session settings as shown in the screen shot below by selecting the Serial button under Connection type. Enter COM5 in the Serial line box and 9600 under the Speed box.



Under the Category selection Serial on the left pane, Connection, SSH, open the screen that shows the serial settings, as shown in the next screen shot, and for the Flow control box select none. Go back to the Session screen and save the configuration used by providing a session name to use (mysession used in this example) and clicking on the Save button.



This configuration is the bare minimum required to get PuTTY to work with the GPS clock. You can customize PuTTY further to suite your taste, like changing the terminal font size, screen size, etc.

You can use PuTTY to view the clock time and date output on the terminal interface as well.

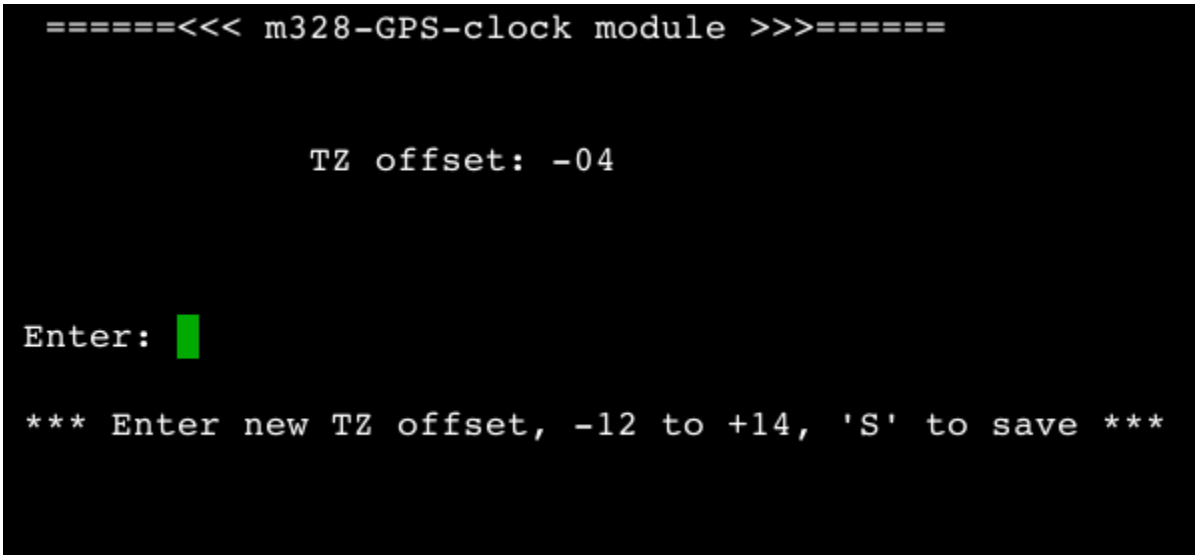
5.2 Time Zone setup program

To enter the time zone setup sub-program, power down the clock and follow the steps below:

- Disconnect the NEO-6 module connections to the main board by unplugging one end of the interconnecting ribbon cable. Don't forget to do this step.

- If you are using an external power connection to the Nano's VIN pin, disconnect the power supply from this pin.
- Place the shorting jumper on the two pin strip header on the main board. This will invoke the setup sub-program when the Nano is reset.
- Using a mini USB cable, connect the Nano's USB connector to a host PC USB port. Start PuTTY on the PC and configure it as described above. The setup screen should appear. You may need to press the reset button on the Nano to get the sign-on screen to appear. The USB cable will provide power to the clock while the cable is connected to the host PC.
- You should see the default time zone setting of -5 displayed. To change the time zone simply enter a new offset value, including a + or – sign. The newly entered setting is displayed. Type an 's' to save the new setting to EEPROM. When the GPS clock is started up in clock mode again, the new time zone setting will be used.
- Power down the clock by removing the USB cable connection. Remove the shorting jumper and reconnect the NEO-6 module back to the Nano board to place the clock in the normal clock mode again. Reconnect the external power connection to the Nano's VIN pin if you are using this external supply.
- Power up the clock and it should run using the new time zone setting.

The screen shot below shows what the time zone setting display screen looks like.



5.3 Adjusting for Daylight Saving Time

If you live in a country that observes daylight saving time (DST) you need to adjust the clock's local time forward by one hour at the beginning of Summer and backward again at the end of Summer.
For example, in Ontario, Canada you advance the time by 1 h when DST begins so the default -5 h is changed by adding +1 h for a new time zone offset of -4 h. At the end of DST you change the time zone offset back to -5 h.

6 Todo list and Observations

The GPS clock program currently displays the UTC date information without any correction for time zone offset, unlike the time display which is corrected for local time display. As a result, the clock's date display rolls over to the next day when the UTC time changes to 00:00:00 and not when the local time rolls over to 00:00:00.

This results in the clock's date changing slightly ahead or after the UTC midnight event, depending on the time zone offset setting of the local clock. For example, with a time zone offset of -5 h (the default setting), the date rolls over when the local time is showing 00:00:00 – 5 = 19:00:00. The date display rolls over 5 h before midnight local time. This minor inconvenience can be fixed by adding code to compensate the date display for time zone offset and may be done at another code revision if time permits.

6.1 Observations

The GPS clock project can be assembled by anyone with average construction skills in about two evenings work. A large part of the building time is taken up in the preparation of the case and in arranging all the component modules to fitting together without undue intereference.

Plan your component layout carefully, work slowly, and double check the wiring on the main board. The photographs of the prototype GPS clock shows one possible case layout, you can choose almost any type of case. Mounting the clock in an existing clock case, or in a custom made enclosure are other alternatives.

When your clock is set up and running, you will enjoy having a super accurate clock providing accurate time for years to come. We hope you enjoy using your GPS clock as much as we do.