Raspberry Pi NTP Clock Setup Guide

Several steps are involved in getting your Raspberry Pi to operate as a NTP Clock. To begin with, you must obtain a LCD Plate (www.adafruit.com) and build it. You must acquire a power supply, network cable, video cable, and case, then assemble. Finally you need to install OS and configure it to run the NTP Clock software. The following steps must be carried out in order:

1. Prepare your SD card with and NOOBS
2. Setup Raspberian (command line, language, location, time zone)
3. Setup Raspberian to support the I2C Bus
4. Verify that your LCD Plate is working
5. Install the WB4SON Clock Code
6. Modify the OS to run the WB4SON Clock Code on power-up

Essentially, the first two steps are getting the Raspberry Pi setup WITH A WORKING INTERNET CONNECTION and the Raspberian OS installed. The first two steps are covered in DETAIL by both Adafruit and www.raspberrypi.org and are beyond the scope of this document, although they are covered in summary below.

Prepae your SD card with NOOBS

The Raspberry Pi board requires a operating system installed on a SD card to function. There are many choices, but the default suggested operating system is Raspberian. The easiest way to install Raspberian is to create a SD card with the NOOBS (New Out Of the Box Software) installation software.

These instructions will guide you through installing a program on your SD card that will allow you to easily install different OS's and to recover your card if you break it.

NOTE: Because NOOBS doesn't begin with a full driver install, you MUST start this with at least a standard USB mouse (Wireless keyboards or mice will NOT work). Also be careful to select US English and Keyboard (choices at the bottom of the screen)

1. Insert an SD card that is 4GB or greater in size into your computer

2. Format the SD card so that the Pi can read it

i. Download the SD Association's Formatting Tool from https://www.sdcard.org/downloads/formatter_4/eula_windows/

ii. Install and run the Formatting Tool on your machine

iii. Set "FORMAT SIZE ADJUSTMENT" option to "ON" in the "Options" menu

iv. Check that the SD card you inserted matches the one selected by the Tool

v. Click the “Format” button

3. Download the New Out Of Box Software (NOOBS) from:

downloads.raspberrypi.org/noobs

4. Unzip the downloaded file

5. Copy the extracted files onto the SD card that you just formatted

6. Insert the SD card into your Pi and connect the power supply

Your Pi will now boot into NOOBS and should display a list of operating systems that you can choose to install. To use the NTP Clock, please install the Raspberian OS. Select that choice and follow the prompts on the screen.

Setup Rasberian (command line, language, location, time zone)

Once Rasberian boots for the first time, it will start a program called "raspi-config" that allows the user to setup the OS. This involves setting up a language, setting your location and time zone, and setting the OS to continue to use the command line (as opposed to the GUI).

1 Expand Filesystem -- NOOBS which does this automatically
2 Change User Password -- Up to you
3 Enable Boot do Desktop -- Don't want to do this, keep it at command line
4 Internationalisations Options -- Want to do this for language, location, zone

I1 - Change Local: Using the spacebar (toggles selections), and arrow keys, eliminate any checked option other than "en-US.UTF-8 UTF-8". Use the <Tab> key to highlight the <OK> and press <Enter>. On the following page, use the down arrow key to highlight "en.US.UTF-8" and press <Enter>. This will set the language choice.

I2 - Change Timezone: Press <Enter>, then pick "US" on the folling

page and press <Enter>. Pick your timezone name,
Then press <Enter>

Once the above steps have been done, use the <Tab> key to locate <Finish> and press <Enter>. Allow the system to reboot, by selecting <Yes>

Setup Rasberian to support the I2C Bus

Raspberian users must edit /etc/modules to add support for the i2c bus. Raspberian comes with a nice command line editor called "nano". But because the /etc/modules file is a protected system file, only a "Superuser" can do this so type the following: "sudo nano /etc/modules"

Once the editor starts use the down arrow to move to the bottom of the file and add the following two lines:

i2c-bcm2708
i2c-dev

Press <Ctrl><X> to Exit the editor and "Y" to save the changes. Support for the i2c bus is now ready to be loaded when the RPI boots.

Force a boot by typing "sudo reboot".

After the system reboots, enter the following two lines to install the python support for the SMBus, and the I2C Tools (note they will take a bit of time to complete, and python-smbus will automatically load the I2C Tools):

sudo apt-get install python-smbus

With your Adafruit LCD Plate installed, type the following command:

sudo i2cdetect -y 1 (use a 0 instead if you have an older Pi)

This will pop up a list of all I2C devices. You should see a "20" on the left-most colum (the I2C Bus address of the LCD).

Verify that your LCD Plate is working (optional)

Obtaining the standard Adafruit LCD code is entirely optional, but it does allow you to run their test examples to be sure your Adafruit LCD Plate is working.

The easiest way to get the standard LCD code onto your Pi is to hook up an Ethernet cable, and clone it directly using 'git'. Simply run the following commands from an appropriate location (ex. "cd /home/pi"):

```
cd /home/pi
sudo apt-get install git
git clone https://github.com/adafruit/Adafruit-Raspberry-Pi-Python-Code.git
cd Adafruit-Raspberry-Pi-Python-Code
cd Adafruit_CharLCDPlate
```

You will also need to install RPi.GPIO, the python library for Pi that allows easy GPIO access. Just run:

```
sudo apt-get install python-dev
sudo apt-get install python-rpi.gpio
```

Force a boot by typing "sudo reboot".

Change to the directory where you installed the Adafruit-Raspberry-Pi-Python-Code, and type the following:

```
cd /home/pi/Adafruit-Raspberry-Pi-Python-Code/Adafruit_CharLCDPlate
sudo python Adafruit_CharLCDPlate.py
```

This will start the Adafruit demo program. The LCD should illuminate and you should be able to press buttons on the front of the LCD plate and see them identify themselves on the LCD. When you are finished, press <Ctrl><C>.

Note: You might have to adjust the contrast control (lower right corner) to see anything on the screen. Also if you don't have an RGB display you will see the backlight turn off as various buttons are pressed.

Install the WB4SON Clock Code

(Note: There is a "_" character used below as in "wb4son_ntp_clock.git")

```
cd /home/pi
git clone https://BBeatty@bitbucket.org/BBeatty/wb4son_ntp_clock.git
```

Test the program by typing

```
sudo python /home/pi/wb4son_ntp_clock/Clock.py
```

Execute the program on power-up

Edit the /etc/rc.local file ("sudo nano /etc/rc.local") and add the following to the end

```
sudo python /home/pi/wb4son_ntp_clock/Clock.py &
```

The "&" at the end allows the program to run in the background and returns you to the shell. The Clock.py command will consume about 0.2% of the RPi's CPU resources.

Issue a "sudo reboot" command to reboot and auto start the program.

Note: If you don't have a Wireless LAN configured yet, you will see a message at the end of the reboot process that "device WLAN0 does not exist". Your NTP clock will show the ETH0 (wired) Ethernet address instead.

Setting up WiFi Connections

The easiest way to setup the WiFi environment is to start the GUI. Type "startx" at the command line. It will take a minute or two for X-Windows to start. Once it does, select "WiFi Config" from the left side of the screen. You will need to identify your WiFi network and enter a password for the connection. Describing this process in detail is beyond the scope of this document. You might find the following Adafruit documentation helpful:

http://learn.adafruit.com/adafruits-raspberry-pi-lesson-3-network-setup/setting-up-wifi-with-raspbian

Once you have a valid WiFi setting, you can operate the NTP Clock from the WiFi connection.

Safely shutting down your RPi

You might be tempted just to pull the power plug, and many do, but you do run the risk of corrupting the SD card if the RPi happened to be writing to the card when the power is removed. The safe way is to issue a

```
sudo shutdown -h now
```

command. This will shut down everything safely, and will halt the CPU rather than rebooting. Once it is halted, you can safely remove the power.

Enhanced Clock Operation

With the October 6, 2013 version of Clock.py, the clock will now wait for the WiFi interface to start working on a power-up for up to 1 minute after the OS boots (which, in itself takes about a minute). This ensures that the clock will be accurate. Once the time is on the display, the user can discover the active IP address (eth0 or wlan0) by pressing and holding the SELECT button. Pressing and holding the UP button will turn the LCD to full white (if it is a color display). Pressing and holding the DOWN button will turn the LCD to red (if it is a color display). Pressing and holding both the RIGHT and LEFT buttons will cause the Pi to shutdown -- wait 15 seconds then remove power. (I put that in the program out of concern that the SD card gets corrupted sometimes when the user simply kills power to the Pi while a program is running.)

73, Bob, WB4SON

October 6, 2013