

# The WeatherDuinoPi tutorial

Reveal the full power of your WeatherDuino with a Raspberry Pi and your favorite weather software

by engolling

30.12.2020

## Basics

1. Download your desired Raspbian distribution. I recommend Lite or only with desktop.  
<https://www.raspberrypi.org/>  
Extract the image to a SD card for example with Win32Disk imager
2. If you have a screen and a keyboard close to you I recommend starting up the raspberry and configure the network connection and ssh via raspi-config, else follow with step 3.
3. After copying is finished a partition with the name "Boot" will appear. For allowing a headless start (without keyboard and display) the following steps have to be carried out:

- a. In the "Boot" partition, create a new utf8 file with UNIX line feed named **wpa\_supplicant.conf**.  
Be aware that your operating system is maybe hiding the correct file name ending.
- b. Add the following lines to the file and adjust them to your needs:

```
country=US
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
    ssid="YourNetworkSSID"
    psk="Your Network's Passphrase"
    key_mgmt=WPA-PSK
}
```

- c. Enable ssh access by creating an empty file named **ssh** without ending.

Source: <https://medium.com/@danidudas/install-raspbian-jessie-lite-and-setup-wi-fi-without-access-to-command-line-or-using-the-network-97f065af722e>

4. After that, your raspberry is ready for the first start. Put in the SD card and connect it to the power supply. After a few minutes of waiting, the raspberry should be ready to connect via SSH using Putty or Kitty.  
The IP address can be found with an IP scanner or in the config page of your wireless router.  
**Tip: If you cannot connect to your raspberry it is advisable to plug in a display and a keyboard. Be aware that you might need a micro USB to USB A adaptor and a mini HDMI to standard HDMI adaptor, depending which raspberry you are using.**
5. After the LAN connection is established there should also be an internet connection available. Execute the following commands to get your raspberry up-to-date:

```
sudo apt-get update
sudo apt-get upgrade
```

6. Change the hostname of your WeatherDuinoPi. Depending on your OS version you can / have to do this in different ways.
  - a) The Raspberry PI Configuration menu in the GUI.
  - b) Manually Change all appearances of the old hostname (raspberrypi) in the following files to the new one:

```
sudo nano /etc/hosts  
sudo nano /etc/hostname
```

Then reboot.

- c) In the raspi-config menu of the console

```
sudo raspi-config
```

Navigate to "2 Network options" and then "N1 Hostname".

There change the hostname to "WeatherDuinoPi" or whatever you want.

Source: <https://www.tomshardware.com/how-to/raspberry-pi-change-hostname>

## Install CumulusMX

1. If you want to use CumulusMX register at the [Cumulus forum](#) for downloading cumulus. After downloading, you can upload it to your WeatherDuinoPi and then follow the instructions of the [official tutorial](#).  
The WeatherDuino serial connection is available under the device "/dev/ttyACM0"
2. Moreover, it is advisable to use the cumulus start-stop script. For installation, follow the instructions in the [forum thread](#).

## Install WeeWx

1. If you have chosen WeeWx as your favorite weather software, please refer to the installation manual of [WeeWx](#). Another very good installation tutorial can be found [here](#).
2. In the weewx.conf file, make sure that your stations settings contain the following:

```
[Vantage]
# This section is for the Davis Vantage series of weather stations.
# Connection type: serial or ethernet
type = serial
# If the connection type is serial, a port must be specified:
# Debian, Ubuntu, Redhat, Fedora, and SuSE:
port = /dev/ttyACM0
[...]
# Vantage model Type: 1 = Vantage Pro; 2 = Vantage Pro2
model_type = 2
```

In the `[[FTP]]` section you might fill in the ftp user for your webserver.

In the section `[StdArchive]` I recommend using an archive interval of 60 seconds and the setting `record_generation = software`

Using hardware consumes more CPU time of your WeatherDuino since it has to browse the flash memory each archive interval.

**If you use a WeatherDuino model (Pro 2 Compact) without a hardware logger you should add the following line:**

```
no_catchup = True
```

3. The other settings have to be customized according to your needs. Please refer to the WeeWx manual for further instructions.
4. I like the [NeoWx](#) skin with WeeWx.
5. Moreover there are useful tools like you can generate WeeWx database backups and send all the raw data to an external server allowing to plot everything you want in a web browser.

**In any case: Set the switch of the WeatherDuino in "RUN" mode after programming. Otherwise a never ending boot loop might occur.**

## Samba network share

1. First install samba

```
sudo apt-get install samba
```

2. Create a folder named WeatherDuino and setup samba to use it as network share in the config file.

```
sudo mkdir WeatherDuino
sudo chmod -R 777 WeatherDuino
sudo nano /etc/samba/smb.conf
```

Walk until the end of the file and add the following lines:

```
[WeatherDuino]
Comment = WeatherDuino folder
Path = /home/pi/WeatherDuino
Browseable = yes
Writeable = yes
create mask = 0777
directory mask = 0777
Public = yes
Guest ok = yes
```

3. After a reboot you can access your WeatherDuinoPi via network and upload files or some other stuff

Source: <https://www.raspberrypi.org/magpi/samba-file-server/>

4. When using WeeWx as weather software I recommend sharing also the following folders:

```
[WeeWX Config]
Comment = WeeWx config folder
Path = /etc/weewx/
Browseable = yes
Writeable = yes
create mask = 0777
directory mask = 0777
Public = yes
Guest ok = yes
```

```
[WeeWX Files]
Comment = WeeWx files folder
Path = /usr/share/weewx/
Browseable = yes
Writeable = yes
create mask = 0777
directory mask = 0777
Public = yes
Guest ok = yes
```

*[WeeWX Database]*  
*Comment = WeeWx Database folder*  
*Path = /var/lib/weewx*  
*Browseable = yes*  
*Writable = yes*  
*create mask = 0777*  
*directory mask = 0777*  
*Public = yes*  
*Guest ok = yes*

*[WeeWX HTML]*  
*Comment = WeeWx HTML folder*  
*Path = /var/www/html/weewx*  
*Browseable = yes*  
*Writable = yes*  
*create mask = 0777*  
*directory mask = 0777*  
*Public = yes*  
*Guest ok = yes*

*[Logfiles]*  
*Comment = WeeWx Logfile folder*  
*Path = /var/log*  
*Browseable = yes*  
*Writable = yes*  
*create mask = 0777*  
*directory mask = 0777*  
*Public = yes*  
*Guest ok = yes*

5. I also suggest to make each of the folders read and writable for everyone since it is easier to configure the system.

**Attention:** Only give full permissions if your system is in a closed network.

```
sudo chmod -R 777 path_of_folder
```

## Install GUI, Remotedesktop and the Arduino IDE

If you want to flash your WeatherDuino or to modify the software with the IDE you can install the GUI stuff which you can access via remote desktop.

1. Install all the GUI stuff (only if you have installed the light version of Raspbian)

```
sudo apt-get install --no-install-recommends xserver-xorg
sudo apt-get install raspberrypi-ui-mods
```

2. In raspi-config set dektop to auto start and auto login.
3. Install XRDP

```
sudo apt-get install xrdp
```

When using the light version it might be necessary:

```
sudo apt-get purge xserver-xorg-legacy
```

4. After the first remote desktop connection is established, it is useful to disable the background image and the screensaver.
5. Finally, you can [download the latest Arduino IDE](#) for ARM and copy it to your network share. Then you can directly execute the install.sh from your remote desktop. This allows you to start the IDE and flash the latest WeatherDuino software to your devices.

Be aware that you have to stop your weather software and put the WeatherDuino in the programming mode if you want to flash your WeatherDuino.

## Automatic programming switch

There are relay modules for the Raspberry PI available that allow you to switch an potential free relay with an IO pin of your Raspberry. They are called like "5v Relay Board for Arduino Relay Module 1 Channel Opto-isolated High Or Low Level Trigger". The module should include an opto coupler and detect 3.3V as high level.

You can supply it via 5V and GND of your Raspberries pin header and attach the trigger for example to GPIO 23. The RUN switch should be connected to COM and NC of the relay output.

Executing the following bash script with elevated rights (sudo) you can enter the programming mode without manual switching.

```
#!/bin/bash
echo "Switch GPIO 23 to high"
chmod 222 /sys/class/gpio/export /sys/class/gpio/unexport
echo "23" > /sys/class/gpio/export
echo "out" > /sys/class/gpio/gpio23/direction
echo "1" > /sys/class/gpio/gpio23/value
read -p "Set GPIO23 low with Enter"
echo "0" > /sys/class/gpio/gpio23/value
echo "GPIO 23 is low"
```

## Optional: Real Time Clock

Optionally you can attach a real time clock (rtc) to your WeatherDuinoPi so it makes correct even when it reboots without an internet connection.

First activate the I<sup>2</sup>C interface of the raspberry using

```
sudo raspi-config
```

Navigate to "3 Interface Options" and then "P5 I2C".

Install I2C Tools:

```
sudo apt-get install i2c-tools
```

And check the address of your device (probably 0x68)

```
sudo i2cdetect -y 1
```

Next you can configure the device and see the actual set time using:

```
echo ds3231 0x68 | sudo tee /sys/class/i2c-adapter/i2c-1/new_device  
sudo hwclock
```

The time can be set using:

```
sudo hwclock -w
```

Finally add the command

```
echo ds3231 0x68 > /sys/class/i2c-adapter/i2c-1/new_device
```

in the line before before 'exit 0' in

```
sudo nano /etc/rc.local
```

and disable the software clock

```
sudo update-rc.d fake-hwclock disable
```