

Remote Antenna Management

Setting up pizero-w

The first thing you need to do is to load an operating system on to the sd card.

<https://www.raspberrypi.org/documentation/installation/installing-images/>

I recommend using an sd card that is suitable for HD video as I have experienced failure due to corruption of files with early sd cards.

When you have the pi up and operating the next thing is to configure the pi to connect to your local wifi network.

<https://www.raspberrypi.org/forums/viewtopic.php?t=184651>

To test if this is working use the web browser on the pi screen to see if you can connect to your favourite web site.

Next is to install “node.js”. This application is the one which allows us to serve a web page and allows us to control the various devices. A very good tutorial on doing just this is available:-

https://www.w3schools.com/nodejs/nodejs_raspberrypi.asp

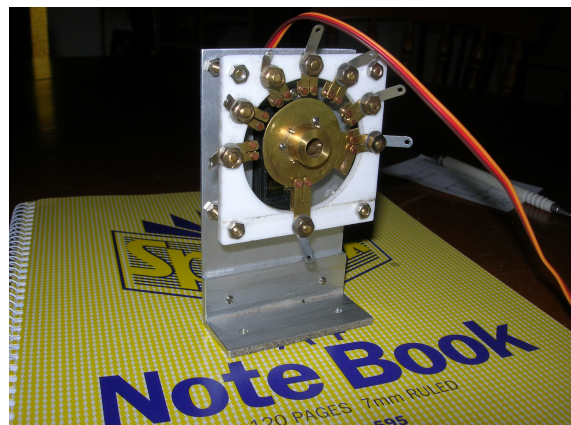
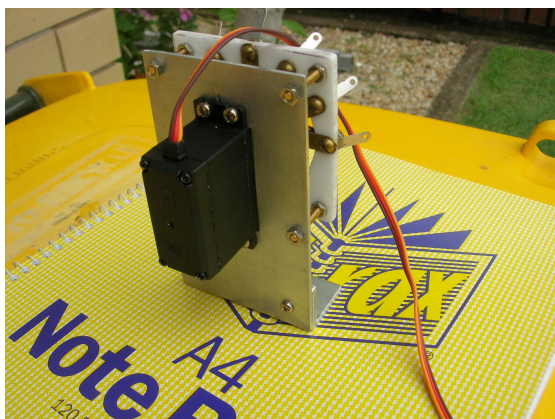
The above site is a great learning site for all types of programming hints in HTML, JavaScript and the like. This site also covers the first two steps as well as covering the hardware of the pi’s GPIO connector.

Because of the battery and solar power supply a battery management package is included. This was implemented using the RRDtools package and cron jobs which via the ADS1115 shield obtain and record the battery voltage every minute. Another cron job via a command line script produce a graph of these voltages for the last 24 hours. The cron package is already installed with the creation as above, of the SD card. RRDtools is installed from the command line as follows:-

```
sudo apt-get install rrdtools
```

As a result of using commercial switches in the original prototype, at very high or low impedances and 100W, these components caught fire. As a result I copied designs from WW2 ATU’s.

The inductor servo switch constructed to handle high voltages and current in this coupler.



The constructed serial parallel switch to handle the high voltages and current.

