

# Setting up an RPI with FLDIGI

This is a very rudimentary explanation of getting FLDIGI onto the raspberry PI once you have already configured the PI with Raspbian. I actually used a pre-built RPI image from SDRPLAY which has built in drivers for my SDR-PLAY but it was based on the standard Raspbian image most people use for RPI's. Once you can get on your PI do the following:

Open a Terminal Session and type:

```
sudo apt-get update
```

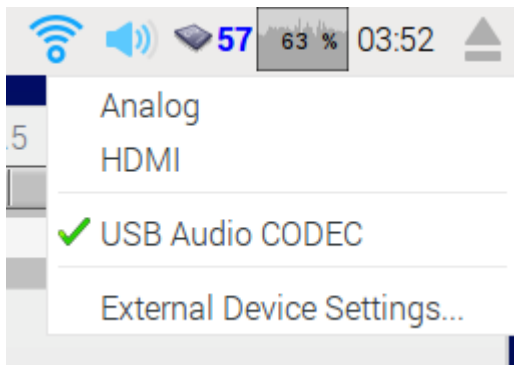
Next:

```
sudo apt-get install fldigi
```

Install drivers to allow control of USB signalink:

```
sudo apt-get install pavucontrol
```

Select the USB Audio CODEC



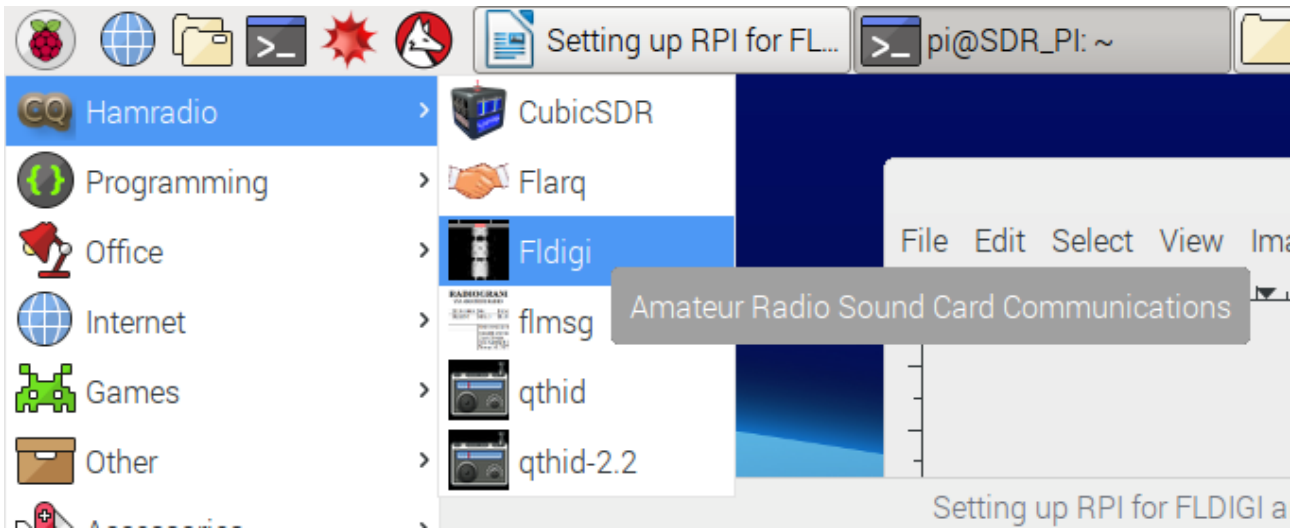
Add user to dial out (use “pi” if you are using default rpi user)

```
sudo adduser pi dialout
```

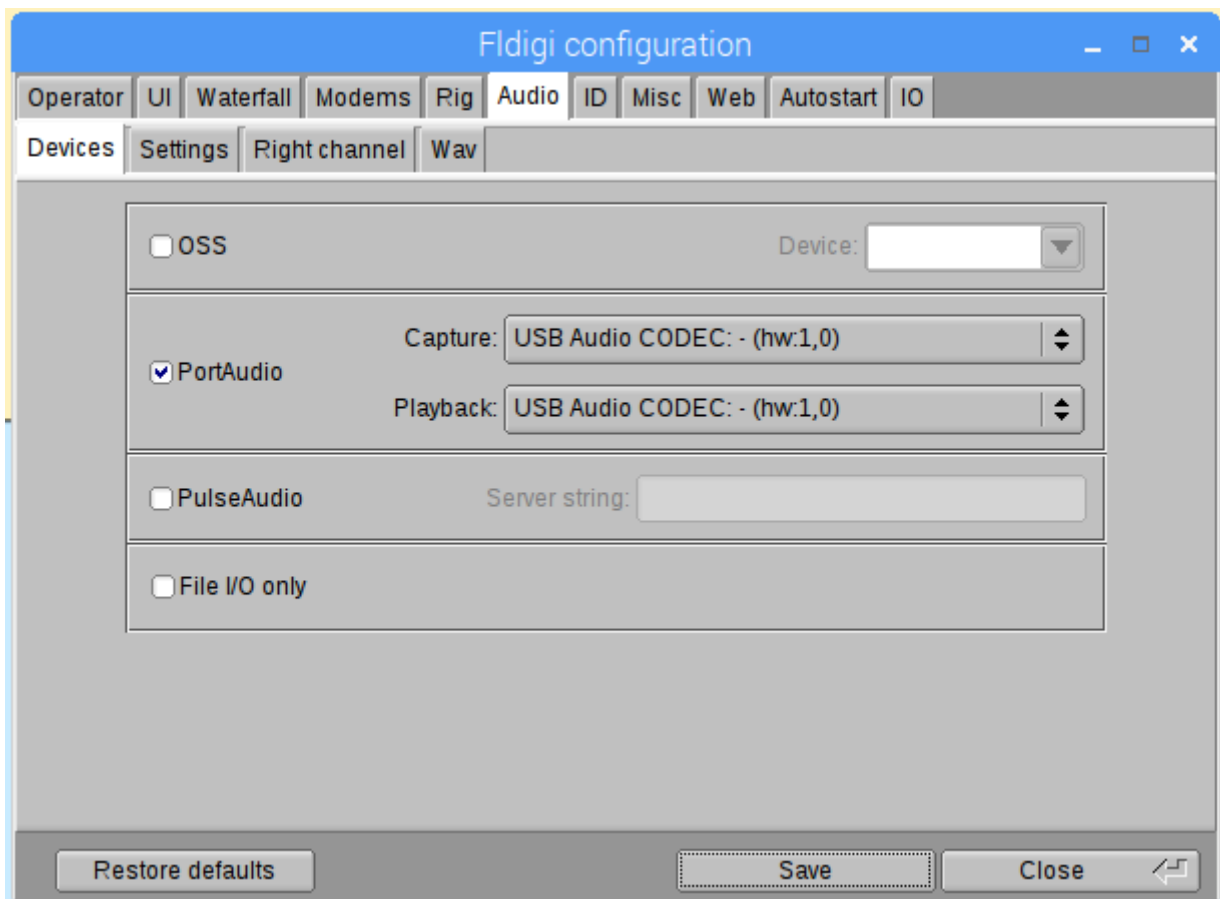
Restart the pi

```
sudo reboot
```

After PI reboots Open FLDIGI from Ham Radio > FLDIGI under the raspberry “start button”

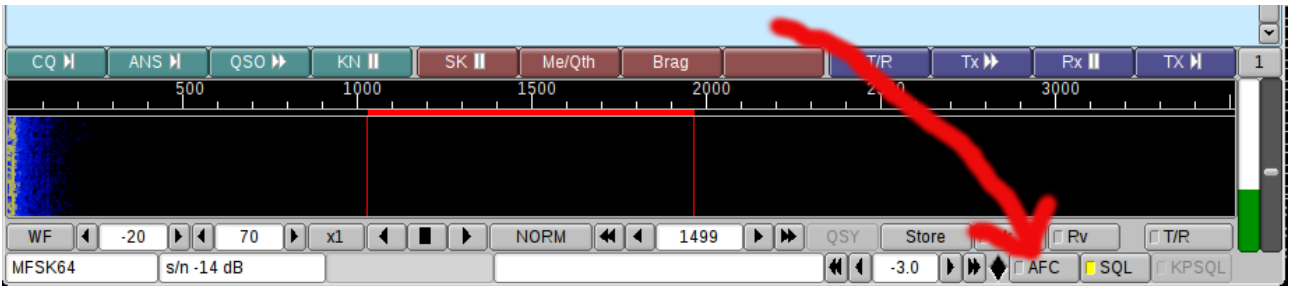


You will be walked through configuration of FLDIGI, make sure you select the following values for the USB soundcard:

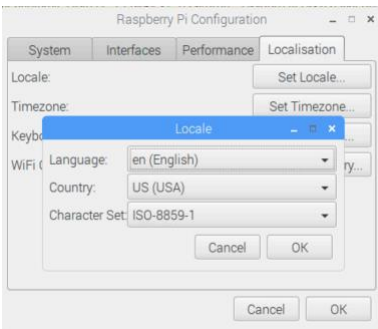
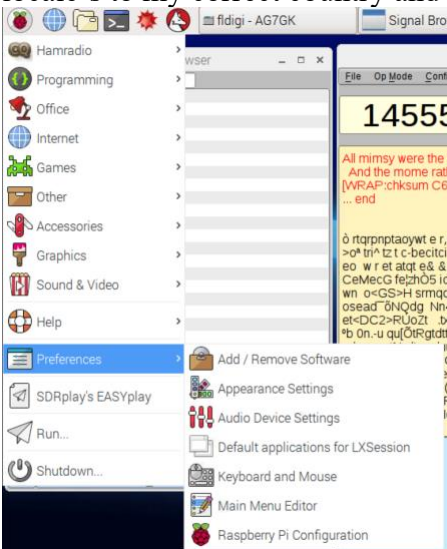


Once in FLDIGI set up like any FLDIGI configuration

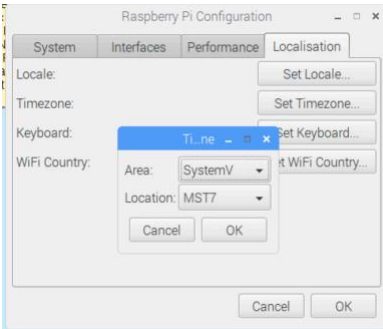
Turn off Automatic Frequency control



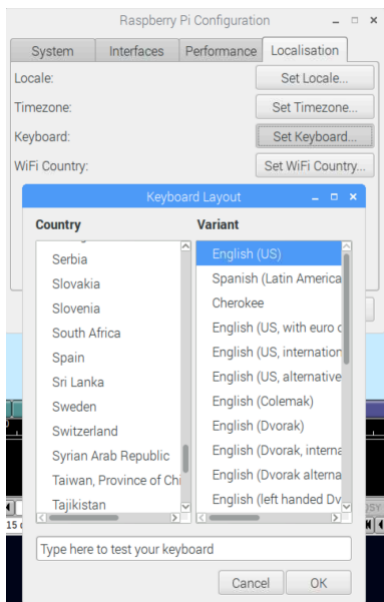
NOTE – I have found some odd situations since loading it, one of the things I did was set the locale's to my correct country and keyboard using the RPI configuration tool under:



Timezone:



Keyboard: (test by typing Shift+2 and you should get @ otherwise you might have the wrong keyboard)



Here are some additional standard FLDIGI settings:

#### FLDIGI Settings:

- RSID - Read Solomon ID should be turned on for the modes we are using, see the FLDIGI users manual here: [http://www.w1hkj.com/FldigiHelp/id\\_configuration\\_page.html](http://www.w1hkj.com/FldigiHelp/id_configuration_page.html)
- TxID and RxID buttons: Turned on:
- RxID button - toggles the detection of [Reed Solomon Identification](#) codes.
- TxID button - toggles the transmission of the RSID signal.
- VideoID: This is fun but not required, you can send your call sign using tones which will write the callsign into the waterfall of the receiving stations. A good tutorial of setting up a macro and your system for this is at: <http://orcadigitalnet.com/macros/>
- 1500hz Offset - we will try to maintain this convention of transmitting at 1500hz, so in the waterfall you select 1500 with your mouse click than push the LK button

Here's a screenshot of how my FLDIGI Looks with all settings selected:

**Using FLRIG if you have CAT control, otherwise set your frequency on your Radio**  
**145550.000**

**Note Frequency at Offset**

**Set rig to 145550.000 Digital**

**VIDEID Macro**

**RxID & TxID On**

**Frequency offset selected in Waterfall**

**TX Locked**

**Squelch ON**

**FLMSG Running**

**Squelch Slider to avoid garbage decodes**

Go to Configure > ID's > RSID:

**Fldigi configuration**

Operator UI Waterfall Modems Rig Audio ID Misc Web Autostart IO

RSID Video CW

**Reed-Solomon ID (Rx)**

Receive modes

Notifies only

Searches passband  Disable alert dialog

Mark prev freq/mode  Retain tx freq lock

Disables detector  Disable freq change

Medium Allow errors 5 Squelch open (sec)

**Pre-Signal Tone**

0.0 Seconds

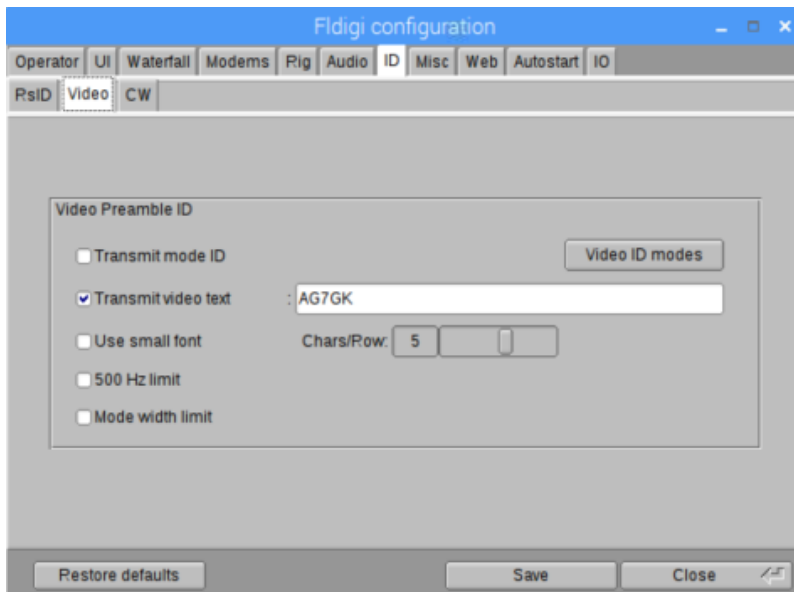
**Reed-Solomon ID (Tx)**

Transmit modes

End of xmt ID

Restore defaults Save Close

Change “Video” tab to the following, filling in for your callsign and characters in your callsign(4,5, or 6)



## Hardware

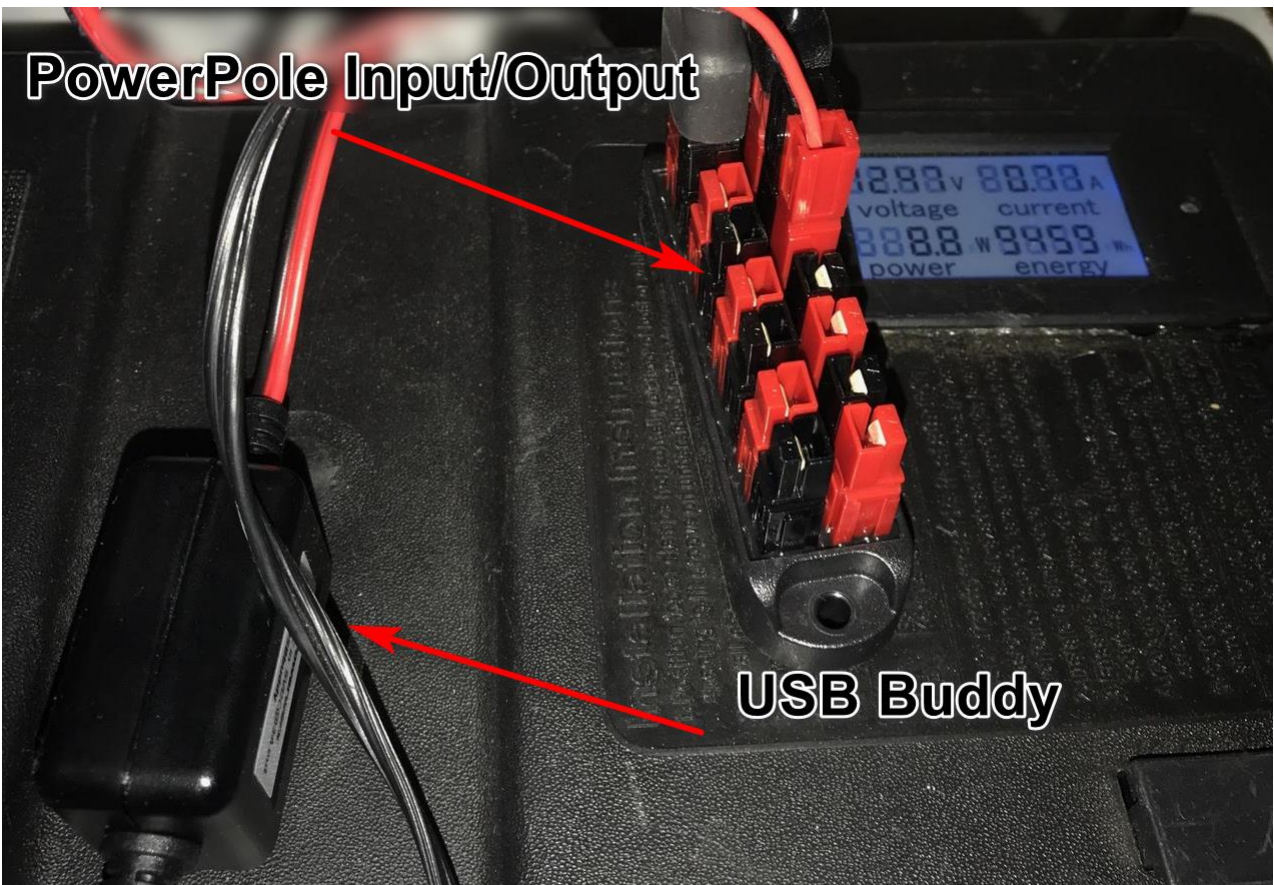
- Raspberry PI
- Signalink USB Soundcard
- FT2D HT with Signalink Interface Cable
- Powerwerx USB Buddy – 5v @3amps, perfect to power the RPI 3 power needs including connected signalink
- Any 12v Battery, I had it hooked to a 12v/20ah battery and then to my deep cycle so I can run it all week or longer in receive mode
- Once the PI is initially configured I connect over VNC and do not run a screen

## Pictures of my setup

PowerWerx USB Buddy



FT2D HT, Rpi (on top of the Signalink), Signalink USB

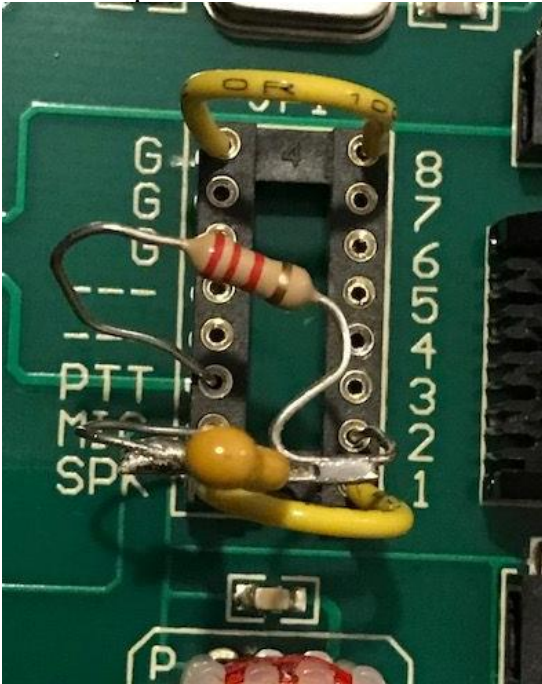


## SIGNALINK Setup for YAESU FT2D

NOTE – when connecting an external soundcard interface to an HT radio some protection for voltages is required, look to your manual and the interface provider for specifics for your radio. In the case of the Yaesu FT2D radio the MIC and PTT share a circuit so you put a resistor and capacitor in place when PTT occurs you don't overload your radio input.

### Screenshot of the Jumpers

2.2 k resistor and a 10uF capacitor connecting the signalink jumpers with wires and the resistor/capacitor.



### Laptop Test with FT2D and Signalink

