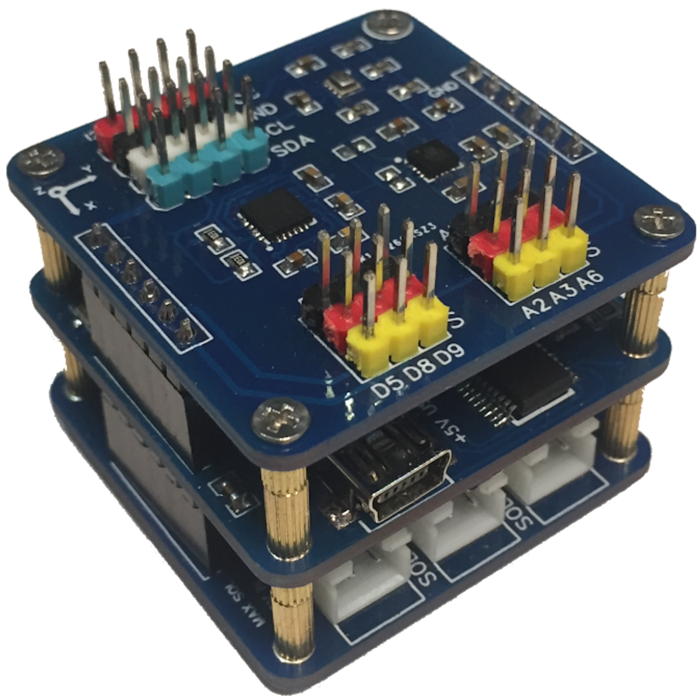
FOSSACUBE INSTRUCTIONAL MANUAL



FOSSA

Revision: V1

Index

1. Kit Contents

The kit should include the following items inside labelled bags as follows:

a) On-Board Processing and Communication Unit.
b) Power Distribution and Regulation Board.
c) Sensor and Breakout Board
d) 800mAh 3.7V Li-ion Battery
e) Micro USB Li-Ion Battery Charger
f) Mounting Hardware Kit 1
g) Mounting Hardware Kit 2
h) Solar Panel Set 1
i) Antenna Combo
j) Solar Wiring Harness

3Dicey Order Specifics:

- SDR and Corresponding Antenna
- Additional Hardware

2. OBC and Communication Board

The OBC is powered by an Atmega328P-AU 8 Bit AVR microprocessor running at an 8MHz clock speed on 3.3V. The board includes:

- An onboard FT232R FTDI USB to Serial converter to simplify communication using a mini USB cable. The incoming 5V from the USB is stepped down to 3.3v for the system onboard. 2 LED lights indicate TX/RX communication.

- A Micro-SD card reader, ideal for gathering large amounts of data, using as a data buffer for transmissions (store data until a transmission is requested) or simply avoiding the use of radio telecommunications.

- 5 Voltage Sensors based on a 6.8 kOhm & 22 kOhm voltage division circuit, sensor breakout pins include this. (Sketch explained in coding section).

- An SX1278 100mW LoRa SPI transceiver, capable of transmitting in AFSK, FSK, LoRa and FM. Maximum technical datarates are achievable in excess of 30kbps but it is recommended to use LoRa transmission within the following specified ranges: SF7-SF11, 125kHz, Coding Rate 4. The antenna connects to a U.FL port.

LEGISLATION WARNING:

All transmission must be carried out following ISM spectrum allocated frequencies under specified conditions. In the case of UK based transmissions, these **must** be carried out under IR2030/1/10 in reference to EN 300 300 2013/752/EU band no. 44b and 45b which states the following settings:
- Max bandwidth of 1.74MHz
- Frequency Range of 433.05 - 434.79MHz
- A maximum Duty cycle of 10% at 10mW (10dBm) (ie: 0.5s transmitting per 5s)

Distributing entities are not held responsible for the misuse of equipment.

Link to Ofcom regulations:

https://www.etsi.org/deliver/etsi_en/300200_300299/30022002/03.01.01_60/en_30022002v030101p.pdf
https://www.ofcom.org.uk/__data/assets/pdf_file/0022/103297/fat-ism-frequencies.pdf

These specified radio settings can be simply programmed into the LoRa transmission sketches explained in the coding section.

Connections

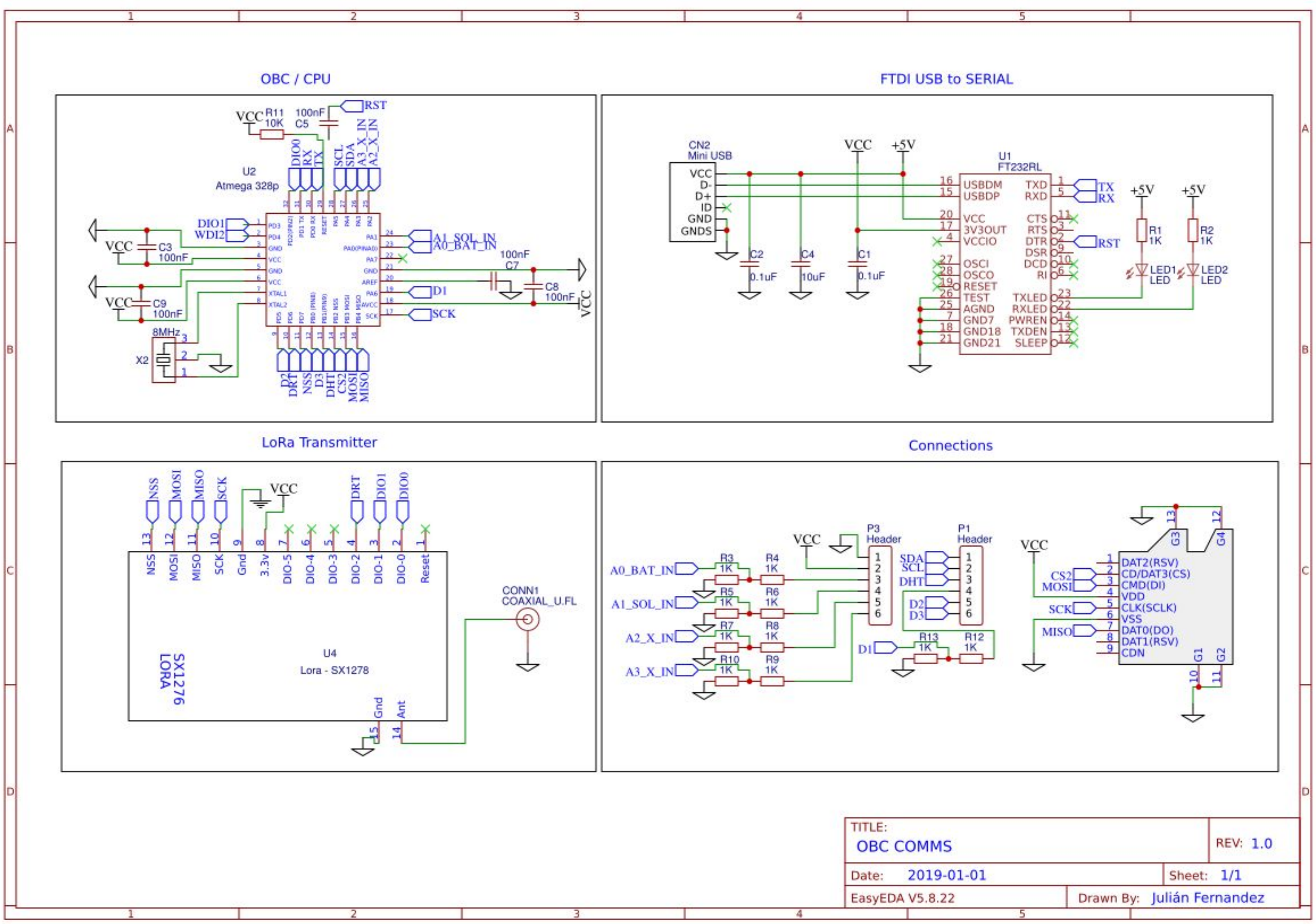
More detailed schematics can be found in the “Schematics” folder. The following connections must be known for programming:

- LoRa: NSS → D7, MOSI → D11, MISO → D12, DIO-1 → D3, DIO-0 → 2, DIO-2 → 6, SCK → 13

- Voltage Sensors: Battery Voltage → A0, Solar Voltage → A1

- SD Card: SD CS → 10, MOSI → D11, MISO → D12, SCK → 13

- Sensor breakout pins are labelled.



3. Power Distribution Board

The main purpose of this board is the regulation of incoming solar power in order to charge the battery and the regulation of a 3.3V common bus powered by the battery. The board uses standard 2.54mm JST connectors for a modular design.

Solar Regulation:

This is carried out using an SPV1040 Maximum Power Point Tracker that maximizes the efficiency of the solar cells. This circuit can boost any voltages from 0.8V to 5V to the rated charging voltage of 3.98V at a maximum power of 1.8A. We recommend charging at under 200mA using the provided solar cells and wiring harnesses in exclusively parallel setups (Max 2V).

Common Bus Regulation:

The common bus is powered by a TC1262 LDO Linear Voltage Regulator, this device can take a voltage of 2.7v to 6V from a battery and lower it to 3.3V with a maximum current draw of 500mA.

Warnings:

DO NOT PLUG BATTERY INTO ANY SOLAR INPUTS! This will burn the SPV1040 modules and disable solar charging.

Do NOT LET BATTERY SINK UNDER 3.3V! This will permanently damage it and is dangerous. Periodically charge the battery using the solar or USB method.

Maximum Ratings:

Characteristic	Units	Max	Minimum	Recommended
Solar Incoming Power	V	3.0	1.0	2.0
Battery Voltage	V	4.2	3.3	3.7
Charge Current (Depends on solar intensity)	mA	1800	-	100

4. Sensor Breakout

The sensor breakout board serves as a multifunction expansion board with several inputs and outputs for sensors as well as counting with the following I2C sensors:

- BMP280 Barometer (Altimeter), Temperature Sensor and Humidity Sensor
- MPU6050 Triple Axis Gyroscope and Accelerometer
- QM5883L Magnetometer (Not to be confused with HMC5883L)

All these sensors have example sketches in the primary folder with the corresponding libraries.

Breakout Pins

The board counts with a variety of input and output pins with the following settings:

Analog In: A2, A3 and A6 serve as voltage DAC inputs for thermistors, current readings, voltage readings etc. These each are preceded by a voltage divider circuit which allows the connections of devices up to 13.9V for voltage readings (Sketches included in the primary folder)

Digital In/Out: Can be used to connect GPS, OneWire Sensors, LED’s, Buttons etc.

I2C: SDA and SCL with already included pull up resistors.

Important: All logic levels and VCC pins are 3.3v besides the Analog inputs.

5. Initial Setup

5.1 Mechanical Assembly

You will need to open the A, B, C and F bags to complete the centre stack.

a) Place the OBC's male headers into the power distribution board's female headers, make sure the alignment mark is on the correct side to avoid incorrect orientation.
b) Screw the standoffs into the OBC module
c) Insert the Sensor Board's male headers into the OBC's female headers and secure using screws. Check the alignment mark before inserting them.

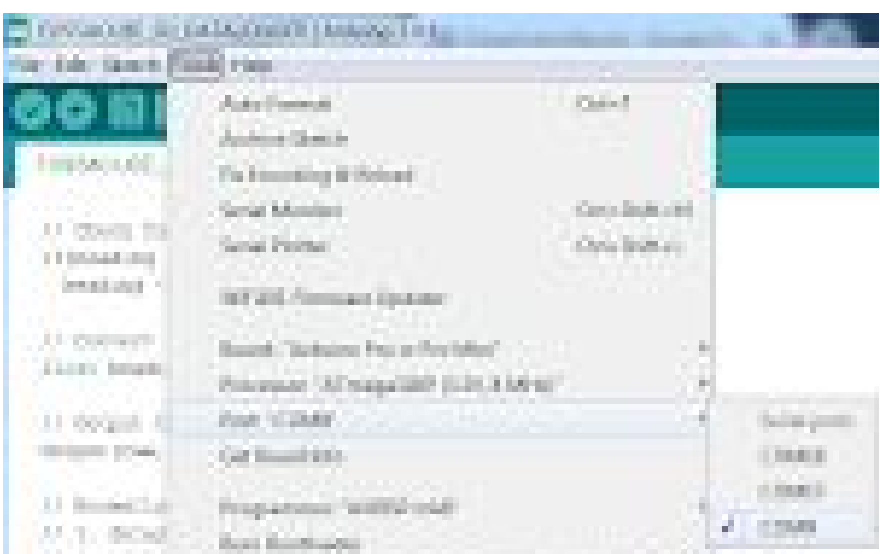
Additional M2 hardware is included, this can be used to secure the stack to the 3D printed model.

5.2 Software Configuration

a) Please install the latest Arduino IDE to ensure compatibility.

b) Once up and running it is necessary to install the included ZIP libraries in the Arduino IDE, this can be done by going to Sketch> Include Library> Add .ZIP library and individually selecting the libraries included in the primary folder.

c) Go to Tools > Board > Select Arduino Pro Mini and then Tools > Board > Processor > Select 3.3V & 8MHz. Use the same configuration as the image below. (Note that port will be different)



d) Plug in the USB mini cable to the OBC and select the corresponding serial port in Tools > Ports.

e) We recommend uploading the Arduino Basic Example "AnalogReadSerial", if your configuration is successful then you should see a sensor value is printed in the serial monitor.

6. RF Setup & Reception

6.1 Antenna Setup

A series of monopole, helical and dipole antennas are included in the kit. These can be experimented with by analysing the maximum received LoRa power depending on the situation. Use a flat head screwdriver to install the U.FL connector before assembling the centre stack.

6.2 LoRa use

A basic example sketch transmits a series of sensor values in a packet, this obviously is only a demonstration and can be expanded to include functions like command reception, data storage, remote configuration etc. One kit can be used as a receiver and one kit can be used as a transmitter using the RX and TX sketches. The link budget for the transmission can be calculated using the LoRa calculator available at: https://www.semtech.com/uploads/documents/SX1272LoRaCalculatorSetup1_1.zip or using datasheet sensitivity information depending on the spreading factor, bandwidth and coding rate. A higher data rate will result in a smaller link budget. LoRa is always recommended over RTTY due to its ease of use, longer link budget and higher data rate.

6.3 RTTY use

A basic example sketch transmits a series of sensor values in RTTY 45 BAUD ITA2, this obviously is only a demonstration and can be expanded to include functions like command reception, data storage, remote configuration etc. RTTY cannot be received using SX1278 transceivers but can be received using a typical SDR such as the one included in the kit, it is a slower modulation but is very common in the HAM amateur world. To decode it, a series of programs must be installed:

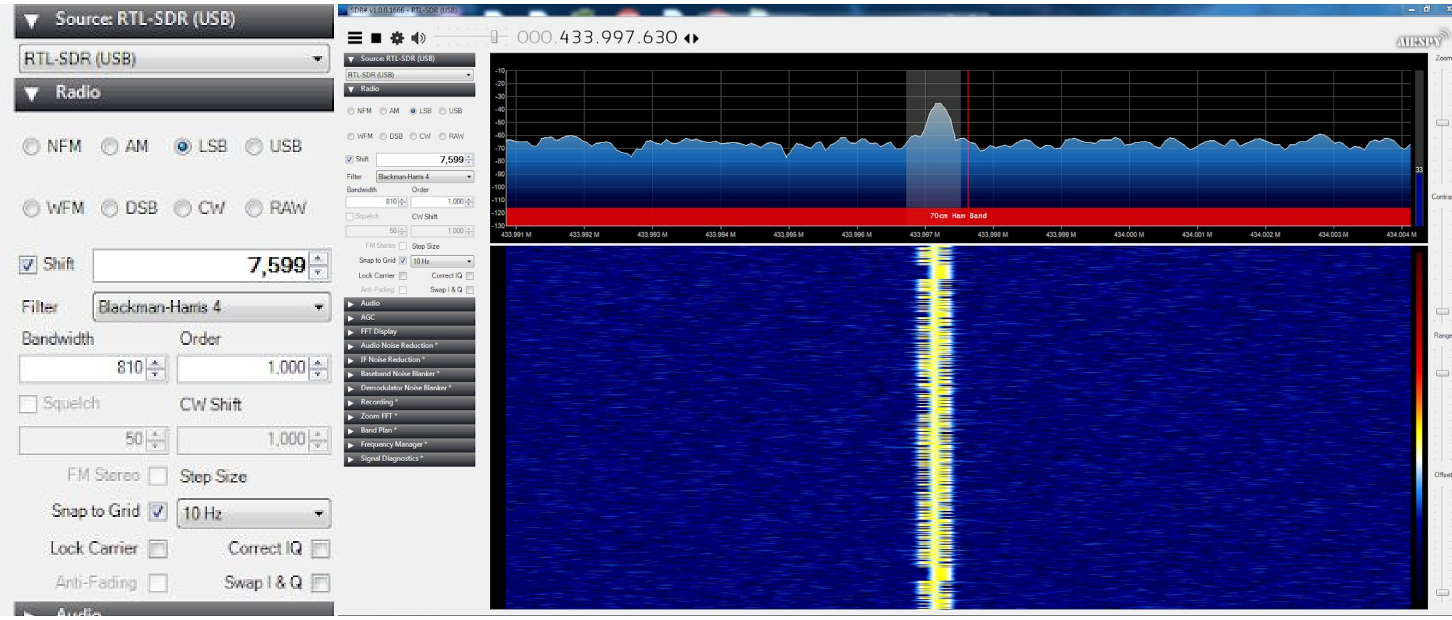
- SDR Reciever: This software will connect to the SDR and give you the tuning and “listening” abilities with the SDR radio waterfall. We recommend the use of SDRSharp for Windows or GQRX for OSX.

- RTTY Decoder: The RTTY itself is an audible modulation similar in functionality to fast morse, however, software is needed to decode this sound. We recommend MMTTY (https://hamsoft.ca/media/Dload/mmtty/MMTTY170K.exe) for windows or FLDIGI (https://sourceforge.net/projects/fldigi/files/) for both OSX and Windows.

- Sound Board: the radio signals picked up by the SDR are then played through your computers sound card, the RTTY decoder will need a sound input which is by default your microphone. We need this to be the computer audio so it can directly be sent to the decoder. Windows by default cannot do this and it necessary to use software such as VoiceMeeter.

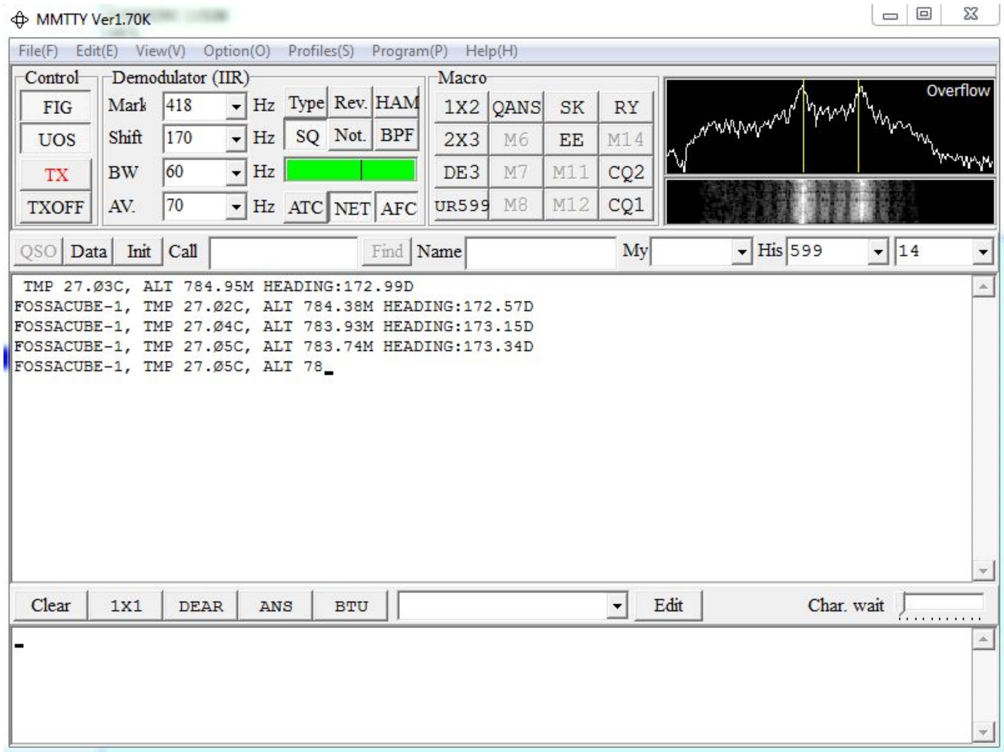
<u>6.3 RTTY Setup</u>

<u>SDRSharp Setup:</u>



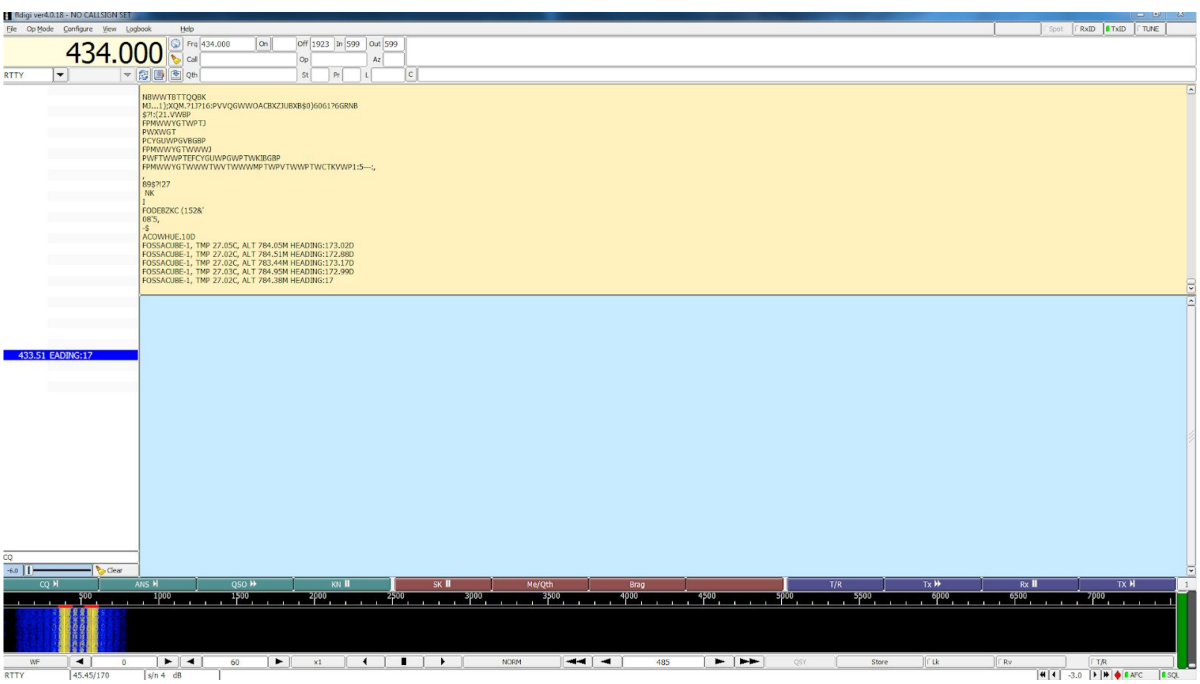
Tune to 434MHz (Zoom out using the right-hand slider if you cannot differentiate a strong signal) Use Lower Side Band, Shift is irrelevant (Dependant on the SDR) and 810hz +/- bandwidth. As you can see in the waterfall, the white tuner should occupy the whole signal and the sound should be clear.

MTTY:

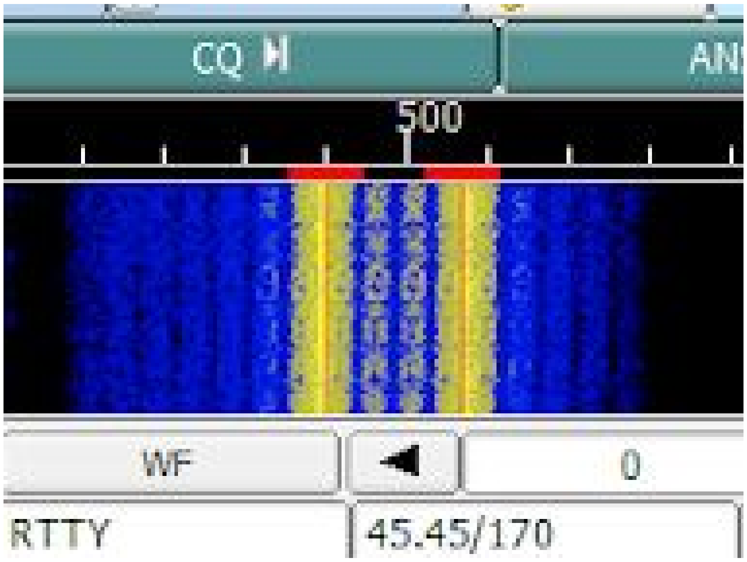


Go to Profiles > Standard RTTY, then click on either side of the right-hand waterfall to adjust the Mark until the 2 RTTY signals are found. You may need to wait until the transmission starts again in order to start decoding messages.

FIDIGI:



Go to OP Mode > RTTY > RTTY-45. Adjust the bottom mark slider at the bottom of the page (Image below) until the signal matches the 2 tuning sliders.



7. Extra transmission styles

Besides the previously mentioned RTTY and LoRa transmission methods, the SX1278 is able to transmit in standard GFSK which features address filtering and in AFSK. Using AFSK it is possible to enter direct mode using DIO-2 and transmit direct sounds from the ATMEGA 328P-AU or tones. This can simulate an FM music signal. Morse can simply be implemented by using an implementation of the .tone FSK function which allows transmission of a specific tone for a set amount of time.