

The RFQuack firmware is broken in the latest version. While the problem is fixed you can use RFQuack in this way:

1.- RFQuack installation

Run the following commands:

- git clone <https://github.com/rfquack/RFQuack.git>
- cd RFQuack

```
joel@joel:~$ sudo rm -r RFQuack/
joel@joel:~$ git clone https://github.com/rfquack/RFQuack.git
Clonando en 'RFQuack'...
remote: Enumerating objects: 1206, done.
remote: Counting objects: 100% (1206/1206), done.
remote: Compressing objects: 100% (583/583), done.
remote: Total 1206 (delta 678), reused 1099 (delta 576), pack-reused 0
Recibiendo objetos: 100% (1206/1206), 24.59 MiB | 16.66 MiB/s, listo.
Resolviendo deltas: 100% (678/678), listo.
joel@joel:~$ cd RFQuack/
joel@joel:~/RFQuack$
```

Run the following command:

- git checkout 982455583e29ec67302149e62ab3b2806a3c49d3

```
joel@joel:~/RFQuack$ git checkout 982455583e29ec67302149e62ab3b2806a3c49d3
Nota: actualizando el árbol de trabajo '982455583e29ec67302149e62ab3b2806a3c49d3'.

Te encuentras en estado 'detached HEAD'. Puedes revisar por aquí, hacer
cambios experimentales y confirmarlos, y puedes descartar cualquier
commit que hayas hecho en este estado sin impactar a tu rama realizando
otro checkout.

Si quieres crear una nueva rama para mantener los commits que has creado,
puedes hacerlo (ahora o después) usando -b con el comando checkout. Ejemplo:

git checkout -b <nombre-de-nueva-rama>

HEAD está ahora en 9824555 Missing `_mod->`
```

Edit the build.env file with this:

```
BOARD=ESP32
RADIOA=CC1101
RADIOA_CS=27
RADIOA_IRQ=25
RADIOB=CC1101
RADIOB_CS=5
RADIOB_IRQ=2
```

Save the changes and run the following command:

- make docker-build-nc && make build

```
Compiling .pio/build/ESP32/FrameworkArduino/stdlib_noniso.c.o
Compiling .pio/build/ESP32/FrameworkArduino/wiring_pulse.c.o
Compiling .pio/build/ESP32/FrameworkArduino/wiring_shift.c.o
Archiving .pio/build/ESP32/libFrameworkArduino.a
Linking .pio/build/ESP32/firmware.elf
Retrieving maximum program size .pio/build/ESP32/firmware.elf
Checking size .pio/build/ESP32/firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [== ] 16.9% (used 55432 bytes from 327680 bytes)
Flash: [== ] 23.0% (used 300870 bytes from 1310720 bytes)
Building .pio/build/ESP32/firmware.bin
esptool.py v3.1
Merged 1 ELF section
===== [SUCCESS] Took 26.84 seconds =====

Environment      Status      Duration
-----
ESP32            SUCCESS    00:00:26.836
===== 1 succeeded in 00:00:26.836 =====
joel@joel:~/RFQuack$
```

Run the following command:

- make proto

```
joel@joel:~/RFQuack$ make proto
cd "/home/joel/.platformio/lib/Nanopb/generator/proto" ; make
make[1]: se entra en el directorio '/home/joel/.platformio/lib/Nanopb/generator/proto'
make[1]: No se hace nada para 'all'.
make[1]: se sale del directorio '/home/joel/.platformio/lib/Nanopb/generator/proto'
cd "src" ; \
protoc --plugin=protoc-gen-nanopb=/home/joel/.platformio/lib/Nanopb/generator/protoc-gen-nanopb \
--nanopb_out=./ \
rfquack.proto \
--python_out=client/
joel@joel:~/RFQuack$
```

Connect Evil Crow RF and flash the firmware with the following command:

- PORT=/dev/ttyUSB0 make flash

```
Compressed 300688 bytes to 153720...
Writing at 0x00010000... (10 %)
Writing at 0x0001dfb6... (20 %)
Writing at 0x000286b2... (30 %)
Writing at 0x0002e011... (40 %)
Writing at 0x00033f43... (50 %)
Writing at 0x0003b338... (60 %)
Writing at 0x000408c0... (70 %)
Writing at 0x00047c52... (80 %)
Writing at 0x000518d5... (90 %)
Writing at 0x00057014... (100 %)
Wrote 300688 bytes (153720 compressed) at 0x00010000 in 13.8 seconds (effective 174.9 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
===== [SUCCESS] Took 49.05 seconds =====

Environment      Status      Duration
-----
ESP32            SUCCESS    00:00:49.049
===== 1 succeeded in 00:00:49.049 =====
joel@joel:~/RFQuack$
```

Exit RFQuack directory:

- `cd ..`

```
joel@joel:~/RFQuack$ cd ..
joel@joel:~$
```

2.- RFQuack-cli installation

Run the following commands:

- `git clone https://github.com/rfquack/RFQuack-cli.git`
- `cd RFQuack-cli`

```
joel@joel:~$ git clone https://github.com/rfquack/RFQuack-cli.git
Clonando en 'RFQuack-cli'...
remote: Enumerating objects: 167, done.
remote: Counting objects: 100% (167/167), done.
remote: Compressing objects: 100% (87/87), done.
remote: Total 167 (delta 101), reused 143 (delta 79), pack-reused 0
Recibiendo objetos: 100% (167/167), 61.62 KiB | 1.99 MiB/s, listo.
Resolviendo deltas: 100% (101/101), listo.
joel@joel:~$ cd RFQuack-cli
joel@joel:~/RFQuack-cli$
```

Run the following command:

- `make docker-build-nc`

```
WARNING: Running pip as the 'root' user can result i
.io/warnings/venv
Removing intermediate container ac6e7b3e5b9e
--> 23667ad0cdd1
Step 5/6 : ENTRYPOINT ["rfquack"]
--> Running in a59474b7f3c0
Removing intermediate container a59474b7f3c0
--> b800a0ca1c2b
Step 6/6 : CMD [ "--help" ]
--> Running in 8eeb0d3d6b25
Removing intermediate container 8eeb0d3d6b25
--> 00a772e2e54d
Successfully built 00a772e2e54d
Successfully tagged rfquack/cli:latest
joel@joel:~/RFQuack-cli$
```

Connect Evil Crow RF and run the following command:

- `docker run --device /dev/ttyUSB0 --rm -it rfquack/cli:latest tty -P /dev/ttyUSB0`

```
Select a dongle typing: q.dongle(id)
- Dongle 0: RFQUACK
> You have selected dongle 0: RFQUACK
```

3.- Example:

Run the following command:

- `q.radioA.set_modem_config(modulation="OOK", carrierFreq=433.920, syncWords=b"", useCRC=False, bitRate=1.7*2, rxBandwidth=58)`
- `q.radioA.set_packet_len(isFixedPacketLen=True, packetLen=100)`
- `q.radioA.rx()`

```
RFQuack(/dev/ttyUSB0)> q.radioA.set_modem_config(modulation="OOK", carrierFreq=433.920, syncWords=b"", useCRC=False, bitRate=1.7*2, rxBandwidth=58)

result = 0
message = 6 changes applied and 0 failed.

RFQuack(/dev/ttyUSB0)> q.radioA.set_packet_len(isFixedPacketLen=True, packetLen=100)

result = 0
message =

RFQuack(/dev/ttyUSB0)> q.radioA.rx()

result = 0
message =

RFQuack(/dev/ttyUSB0)> □
```

Transmit with a remote control:

```
result = 0
message =

data = b'\x03\x00P`b(\x00\x00\x00\x00\x00\x00 \x00\x00\x04\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x01\x08T\x08\x00\x00\x00\x0c\x
rxRadio = 0
millis = 387056
repeat = 0
bitRate = 3.4000000953674316
carrierFreq = 433.9200134277344
syncWords = b'\xd3\x91'
modulation = OOK
frequencyDeviation = 0.0
RSSI = -71.0
model = CC1101
hex data = 030050606228000000000000200000040000000000030800000000000000000
0000100
```