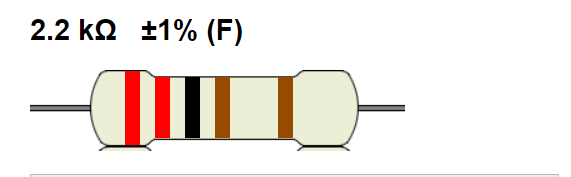
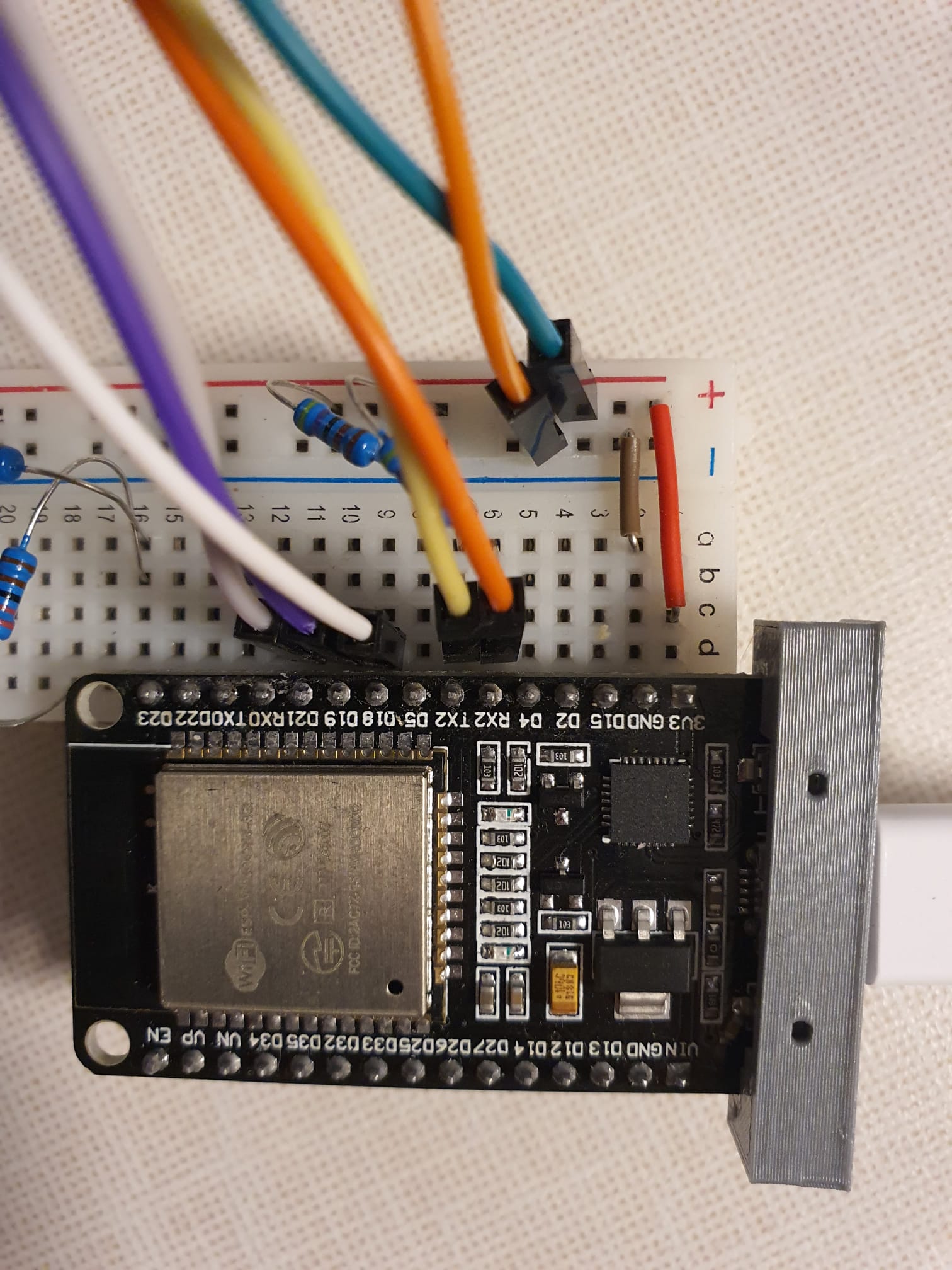
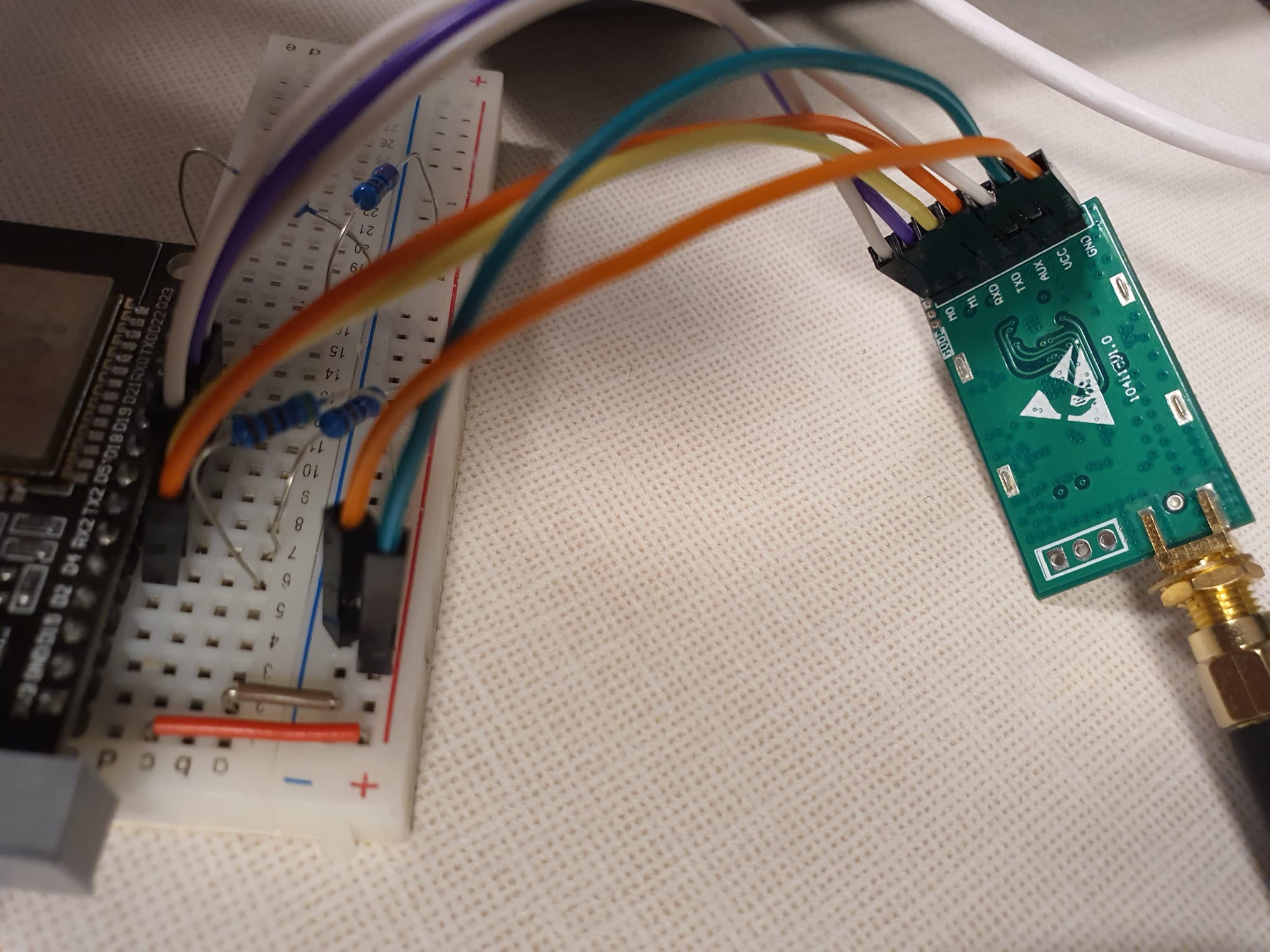
Set and Get Configuration - EBYTE:  
  
While working with the EBYTE Lora module we relied on this website and the library in it (LoRa\_E220) :   
<https://www.mischianti.org/2022/03/11/ebyte-lora-e220-llcc68-device-for-arduino-esp32-or-esp8266-specs-and-basic-use-1/#Normal_mode_connection>

PULL UP Resistors:  
4.7K ohm and two 2.2k ohm  


The board connections:

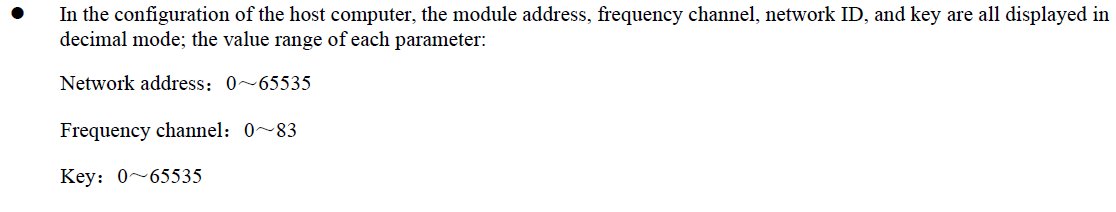
  
  
  
  
  
**Stand alone EBYTE:**  
  
- getConfiguration():

While running this Lora\_e220 library example code we used this Lora\_E220 constructor:

// ---------- esp32 pins --------------

LoRa\_E220 e220ttl(&Serial2, 18, 21, 19); // RX AUX M0 M1

When uploading the code , we kept getting the following message in the serial monitor:

“No response from device! (Check wiring)”  
   
As well as getting results that don’t fit the frequency channel  
The channel should be between 0 to 83 and the frequency is calculated as   
Frequency = 850.125 + channel \* 1Mhz  
(the following is taken from the ebyte manual page 17):  
  
  
  
  
  
Previous Serial Monitor results:

HEAD : F0 1E 9C // the head includes: command (2bytes) - starting address (2bytes) - Length (2 bytes). This is a command that is sent to the hardware.  
 read info. See below more reference from manual.

AddH : 52

AddL : 3A

Chan : 240 -> 650MHz // channel is over 83 and freq below 850.125

SpeedParityBit : 10 -> 8E1

SpeedUARTDatte : 1 -> 2400bps

SpeedAirDataRate : 110 -> 38.4kbps

OptionSubPacketSett: 10 -> 64bytes

OptionTranPower : 10 -> 13dBm

OptionRSSIAmbientNo: 0 -> Disabled (default)

TransModeWORPeriod : 101 -> 3000ms

TransModeEnableLBT : 0 -> Disabled (default)

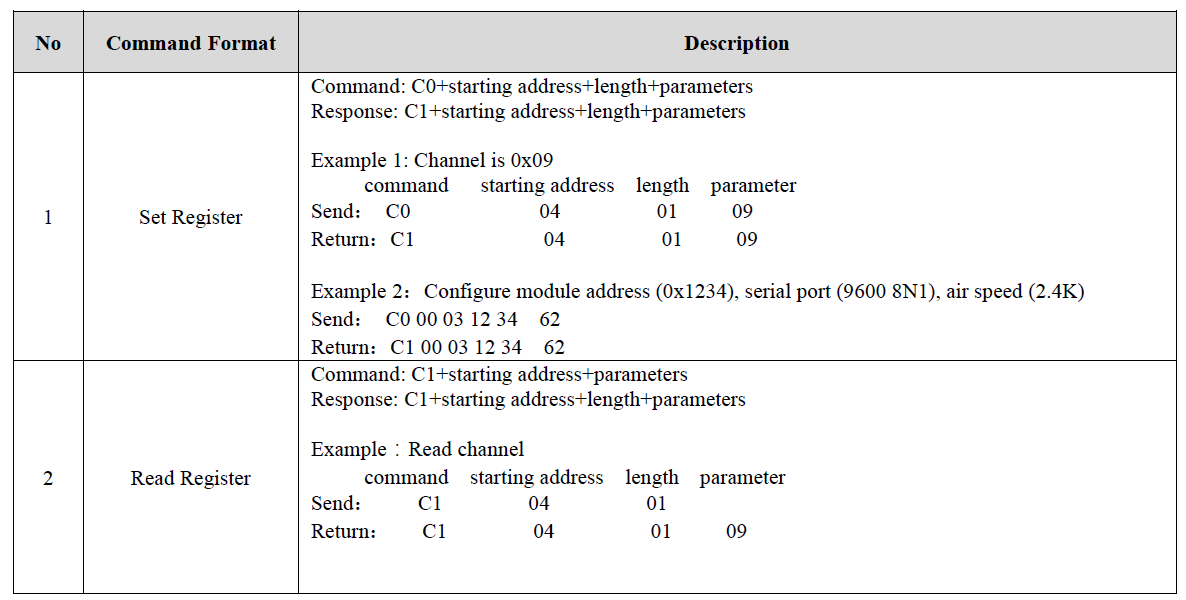
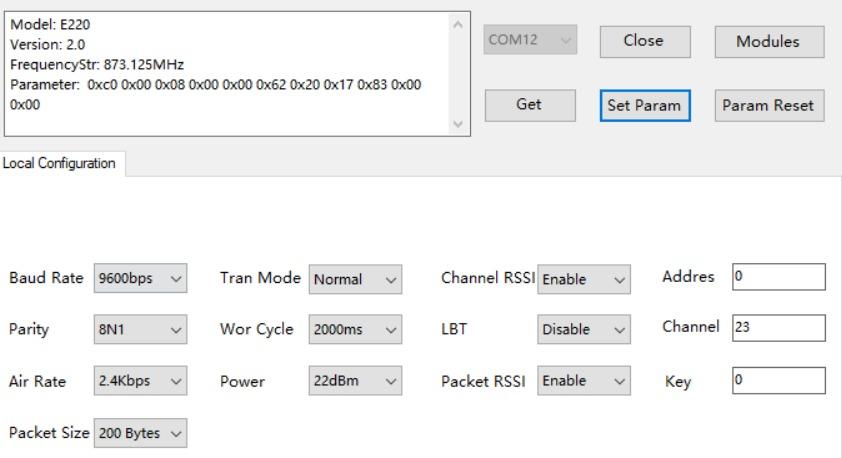
TransModeEnableRSSI: 1 -> Enabled

TransModeFixedTrans: 1 -> Fixed transmission (first three bytes can be used as high/low address and channel)

----------------------------------------

No response from device! (Check wiring) //!

12

Manual Commands to the device, specifically reading the config registers (in this example the starting address is 0x04, which contains the channel address, more info on where the information of each parameter resigns is in the manual page 14 “7.2 Register Description” ) :  
  
  
  
  
We configured the RF settings with the software provided from the manufacturer like this:  
  
   
  
  
While debugging, compiling and running the code a few times we managed to get the correct information from the serial monitor (we did not get the “No response from device! (Check wiring)” error, even though we haven’t changed the set up or the code):

----------------------------------------

Success

1

----------------------------------------

HEAD : C1 0 8

AddH : 0

AddL : 0

Chan : 23 -> 433MHz //the OPERATING\_FREQUENCY in the library is set to 410 Mhz instead of 850Mhz, defined in include/statesNaming.h

SpeedParityBit : 0 -> 8N1 (Default)

SpeedUARTDatte : 11 -> 9600bps (default)

SpeedAirDataRate : 10 -> 2.4kbps (default)

OptionSubPacketSett: 0 -> 200bytes (default)

OptionTranPower : 0 -> 22dBm (Default)

OptionRSSIAmbientNo: 1 -> Enabled

TransModeWORPeriod : 11 -> 2000ms (default)

TransModeEnableLBT : 0 -> Disabled (default)

TransModeEnableRSSI: 1 -> Enabled

TransModeFixedTrans: 0 -> Transparent transmission (default)

----------------------------------------

Success

1

----------------------------------------

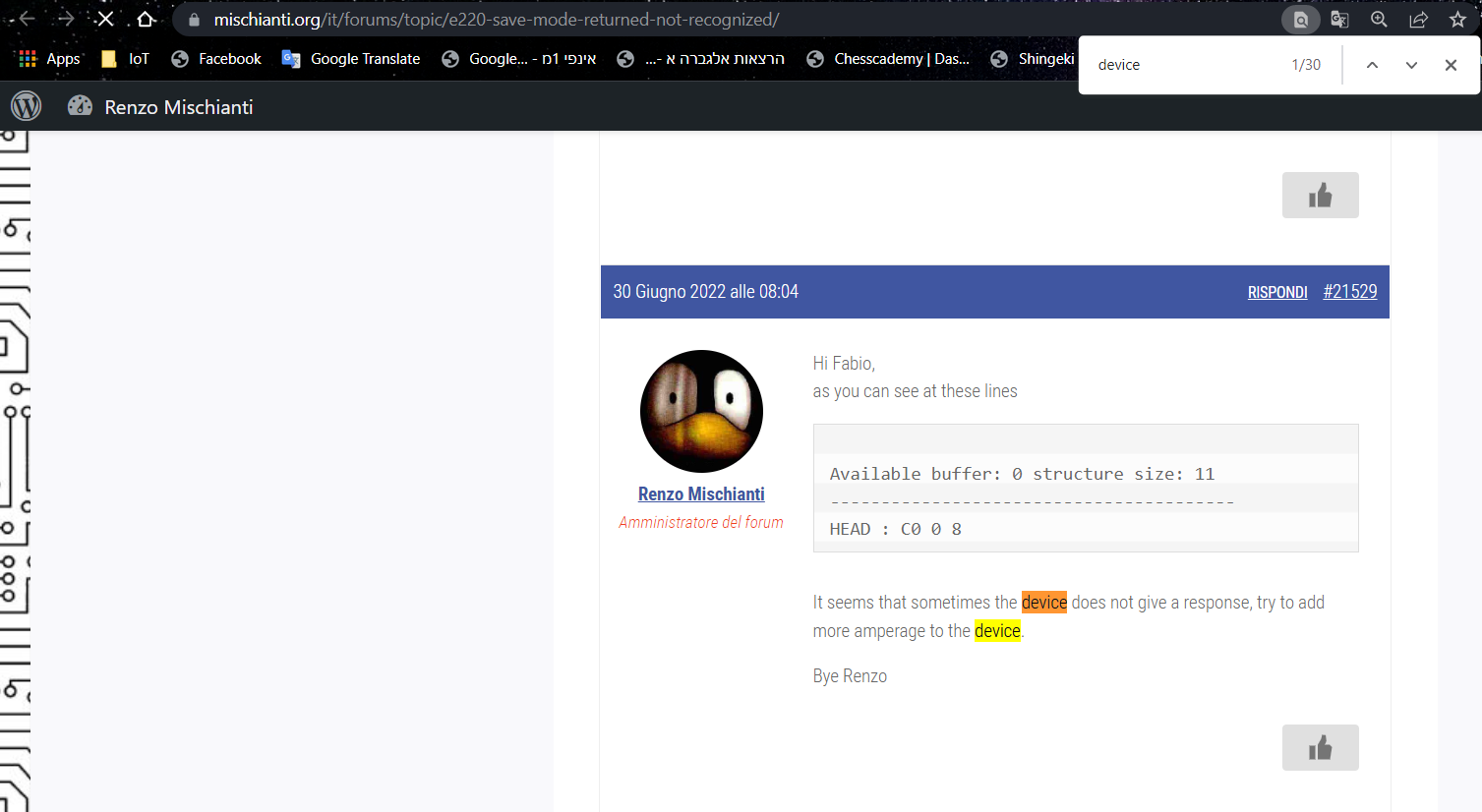
HEAD: C1 8 3

Model no.: 20

Version : B

Features : 16

----------------------------------------

These results correspond to the RF settings perfectly.  
While troubleshooting we saw a similar problem in the forum on the Lora library website, which suggested the problem could be caused from the power voltage not being high enough:  
<https://www.mischianti.org/it/forums/topic/e220-save-mode-returned-not-recognized/>  
  


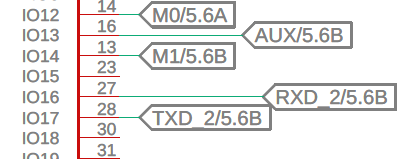
**NODE EBYTE:**

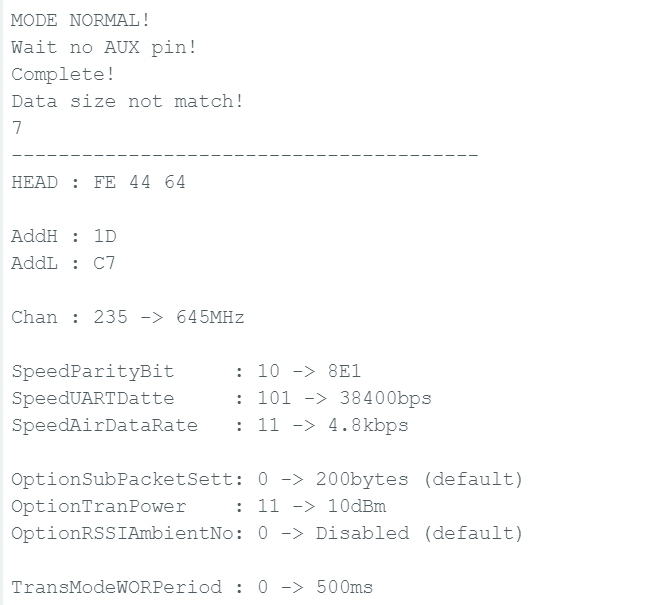
-getConfiguration():  
While running this Lora\_e220 library example code we used this Lora\_E220 constructor:

// ---------- esp32 pins --------------

LoRa\_E220 e220ttl(&Serial2, 13, 12, 14); // RX AUX M0 M1

Once it’s compiled and running we get this message:

“Guru Meditation Error: Core 1 panic'ed (LoadProhibited). Exception was unhandled.”  
  
We started debugging (using the #define LoRa\_E220\_DEBUG in the header file of the lib) and received “timeout!!” error before the core panic.  
We pinpointed the problem in a lib function ( waitCompleteResponse() ), according to the connections schematic, the AUX should be pulled up,  
In the node however the AUX there is no pull up resistor, hence the timeout error and.  
  
  
  
  
  
We decided to ignore the aux input and leave it empty by using this constructor  
// ---------- esp32 pins --------------

LoRa\_E220 e220ttl(&Serial2, -1, 12, 14); // RX AUX M0 M1  
  
  
While running setconfiguration we’d receive:  
  


Recommended experiments -

2 main working modes -

1. test ebyte on node with Arduino library (currently not working)
   1. AUX pin (only in software) - currently tries and no success.
   2. power issue
2. use Danielle’s code. making sure it is transmitting at fixed intervals.
   1. make small modifications to make sure the code is transmitting at 1 second intervals.
      1. where in the code is the LORA target address?
      2. where in the code is the target LORA channel?

Make sure Ebyte on node is working:

1. higher voltage

| **1** | **node** | **gateway (ESP32 devkit)** |
| --- | --- | --- |
| **2** |  |  |
| **3** |  |  |