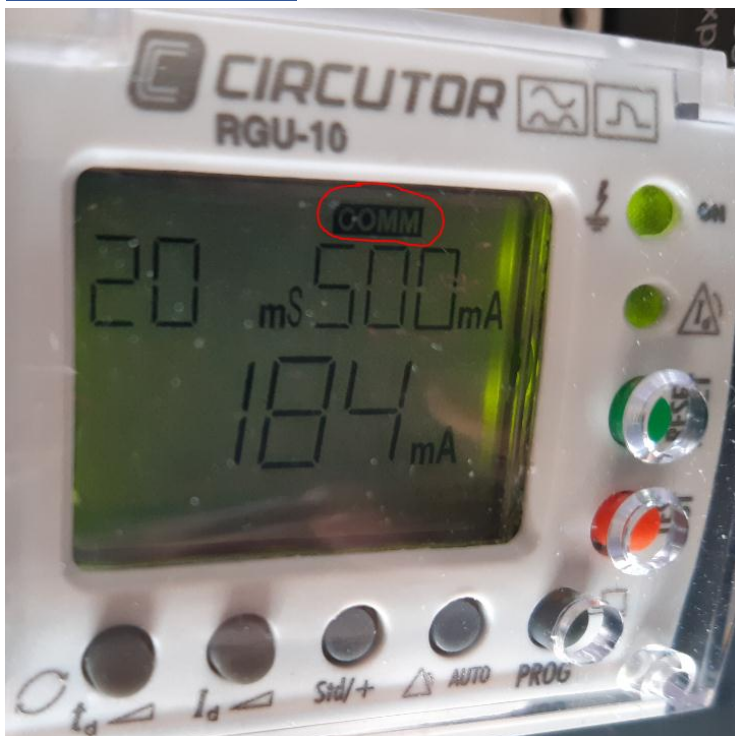


# Dragino RS485 to LoRa test

Let's test with this Modbus device

By Xavier Florensa Berenguer From Noria GRUPO DE COMPRAS

[RGU10C Circutor](#)



## 6.3.- COMANDOS MODBUS

Todas las direcciones del mapa Modbus están en Hexadecimal.

Tabla 12: Mapa de memoria Modbus.

| Parámetro                          | Símbolo | Dirección | Lectura / Escritura | Valores  | Unidades |
|------------------------------------|---------|-----------|---------------------|--|----------|
| Nº de periférico                   | PERI    | 0000      | R/W                 | 1 - 99   | -        |
| Velocidad de comunicación          | bd      | 0001      | R/W                 | 2400-4800-9600-19200-38400-57600-115200                          | baudios  |
| Tipo de paridad                    | PARI    | 0002      | R/W                 | None - Odd - Even  | -        |
| Frecuencia de trabajo              | FREC    | 0003      | R/W                 | 50 - 60  | Hz       |
| Corriente disparo Relé principal   | Id      | 0004      | R/W                 | 0.03 - 0.1 - 0.3 - 0.5 - 1 - 3 - 5 - 10                          | A        |
| Tiempo de retardo Relé principal   | td      | 0005      | R/W                 | INS - SEL - 0.02 - 0.1 - 0.3 - 0.4 - 0.5 - 0.75 - 1 - 3 - 5 - 10 | s        |
| Polaridad contactos Relé principal | Std/+   | 0006      | R/W                 | Estándar - Positiva  | -        |
| Corriente disparo Prealarma        | Id'     | 0007      | R/W                 | OFF - 50 - 60 - 70 - 80  | %        |
| Tiempo de retardo Prealarma        | td'     | 0008      | R/W                 | 0.02 - 0.1 - 0.2 - 0.3 - 0.4 - 0.5 - 0.75 - 1 - 3 - 5 - 10       | s        |
| Polaridad contactos Prealarma      | Std/+   | 0009      | R/W                 | Estándar - Positiva  | -        |
| Reconexión prealarma               | REC     | 000A      | R/W                 | Manual - REC   | -        |

Let's read frequency on register 3

04 means read input registers

0003 is the address we read: Frequency

0002 is the byte number (2)

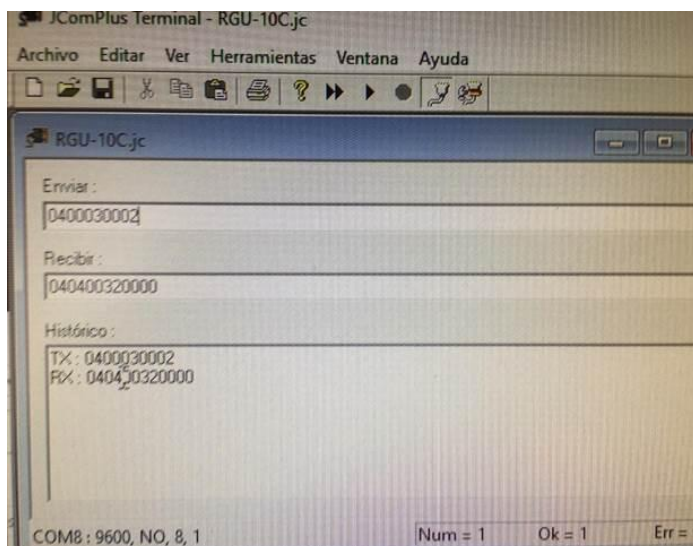
And this is the answer

04 means read input registers

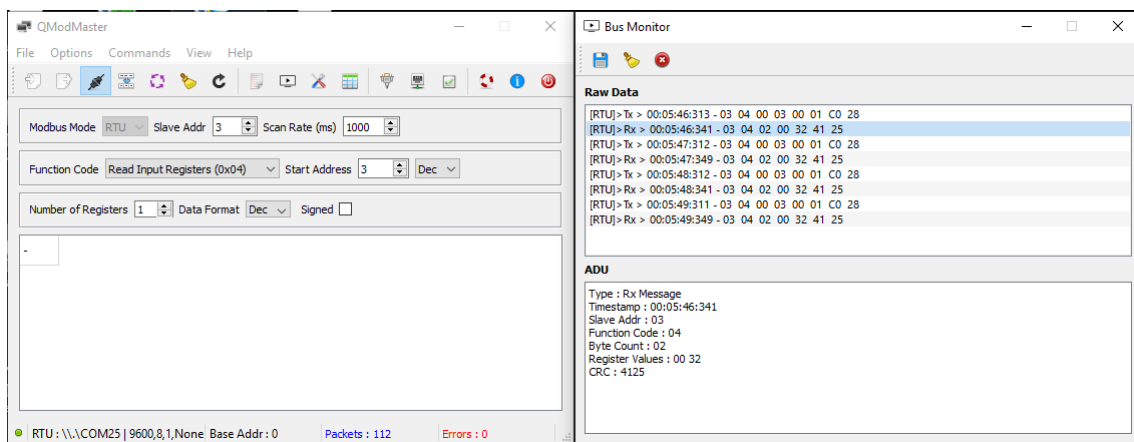
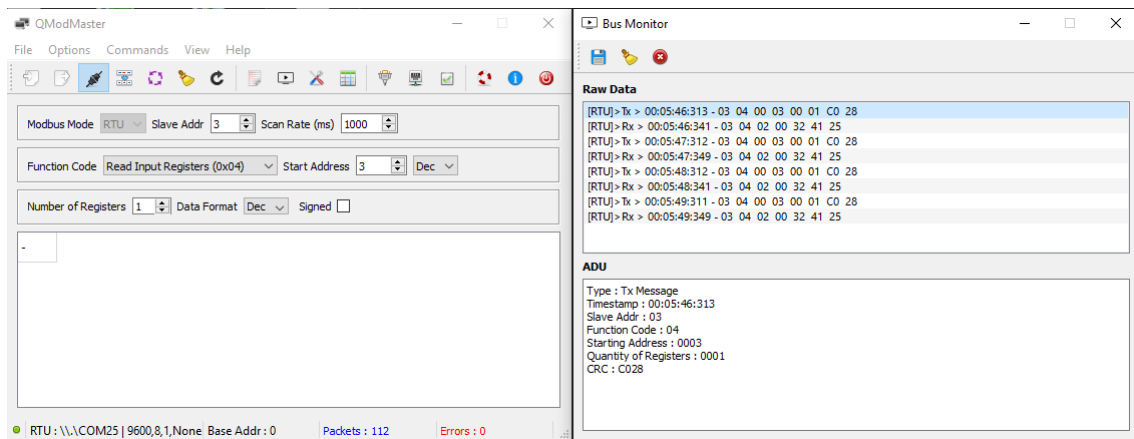
04 is the returned bytes amount, yes 00 32 00 00 (4 bytes)

0032 is 50 in decimal, so the working frequency

With Jcom



And in more Depth including CRC



Sys > 22:39:57:938 - Connecting to Serial Port [\\.\COM25]...OK

[RTU]>Tx > 22:40:05:301 - 03 04 00 03 00 01 C0 28

[RTU]>Rx > 22:40:05:326 - 03 04 02 00 32 41 25

[RTU]>Tx > 22:40:06:303 - 03 04 00 03 00 01 C0 28

[RTU]>Rx > 22:40:06:334 - 03 04 02 00 32 41 25

[RTU]>Tx > 22:40:07:304 - 03 04 00 03 00 01 C0 28

[RTU]>Rx > 22:40:07:342 - 03 04 02 00 32 41 25

[RTU]>Tx > 22:40:08:302 - 03 04 00 03 00 01 C0 28

[RTU]>Rx > 22:40:08:333 - 03 04 02 00 32 41 25

Now we try to enter this telegram on the Draguno RS485 to send the answer to TTN

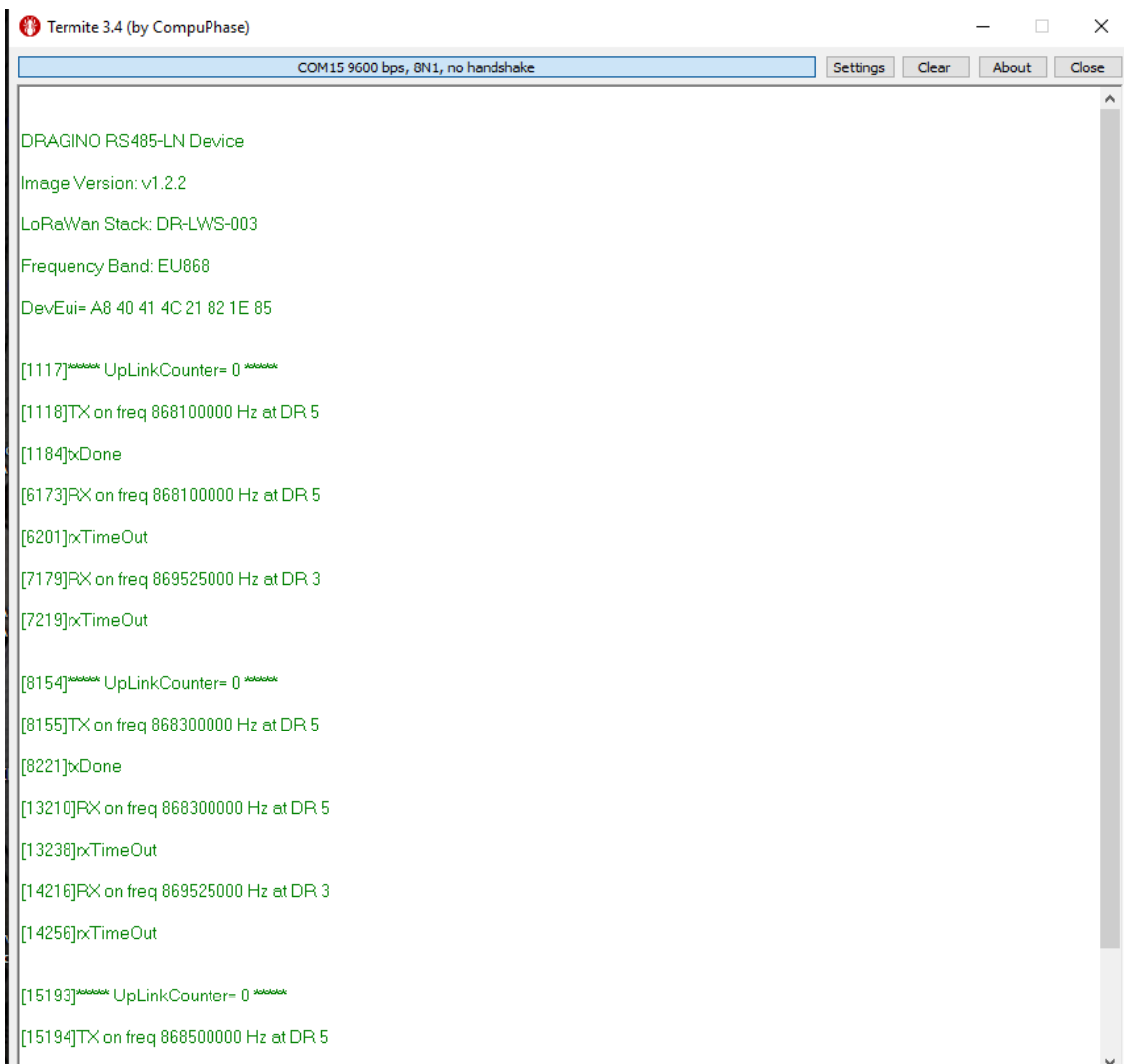
Let's see the response from the Dragino device

Let's connect the programming USB to TTL. This i sport 15

Let's open Terminate terminal

### Serial port settings

|   |  |  |  |   |  |
|---|--|--|--|---|--|
| <b>Port configuration</b><br>Port: <input type="text" value="COM15"/> |  | <b>Transmitted text</b><br><input type="radio"/> Append nothing<br><input checked="" type="radio"/> Append CR<br><input type="radio"/> Append LF<br><input type="radio"/> Append CR-LF<br><input checked="" type="checkbox"/> Local echo |  | <b>Options</b><br><input type="checkbox"/> Stay on top<br><input checked="" type="checkbox"/> Quit on Escape<br><input checked="" type="checkbox"/> Autocomplete edit line<br><input checked="" type="checkbox"/> Keep history<br><input type="checkbox"/> Close port when inactive |  |
| Baud rate: <input type="text" value="9600"/>                          |  | <b>Received text</b><br>Polling: <input type="text" value="100"/> ms<br>Max. lines: <input type="text"/>   |  | <b>Plug-ins</b><br><input type="text"/>   |  |
| Data bits: <input type="text" value="8"/>                             |  | Font: <input type="text" value="default"/>   |  |   |  |
| Stop bits: <input type="text" value="1"/>                             |  | <input type="checkbox"/> Word wrap   |  |   |  |
| Parity: <input type="text" value="none"/>                             |  |  |  |   |  |
| Flow control: <input type="text" value="none"/>                       |  |  |  |   |  |
| Forward: <input type="text" value="none"/>                            |  |  |  |   |  |
| User interface language: <input type="text" value="English (en)"/>    |  | <input type="button" value="Cancel"/>  |  | <input type="button" value="OK"/>   |  |



Now let's try to send some AT commands

```
AT+BAUDR=9600
```

```
OK
```

```
AT+PARITY=0
```

```
OK
```

Yeah, it is alive!

Let's connect our RGU10C to the RS485-LN

The AT+BAUDR command can set the baud rate;

The AT+PARITY command can set the data verification method;

The example given by your email, the corresponding configuration of the device:

```
AT+COMMAND1=03 04 00 03 00 01,1
```

```
AT+DATACUT1=7,2,4~5
```

```
Sys > 22:39:57:938 - Connecting to Serial Port [\\.\COM25]...OK
```

```
[RTU]>Tx > 22:40:05:301 - 03 04 00 03 00 01 C0 28
```

```
[RTU]>Rx > 22:40:05:326 - 03 04 02 00 32 41 25
```

```
[RTU]>Tx > 22:40:06:303 - 03 04 00 03 00 01 C0 28
```

```
[RTU]>Rx > 22:40:06:334 - 03 04 02 00 32 41 25
```

```
[RTU]>Tx > 22:40:07:304 - 03 04 00 03 00 01 C0 28
```

```
[RTU]>Rx > 22:40:07:342 - 03 04 02 00 32 41 25
```

```
[RTU]>Tx > 22:40:08:302 - 03 04 00 03 00 01 C0 28
```

```
[RTU]>Rx > 22:40:08:333 - 03 04 02 00 32 41 25
```

```
AT+COMMAND1=03 04 00 03 00 01,1
```

```
AT+DATACUT1=7,2,4~5
```

Give the commands to program the transactions

The example given by your email, the corresponding configuration of the device:

```
AT+COMMAND1=03 04 00 03 00 01,1
```

## AT+COMMAND1

AT+COMMANDx=XX XX XX XX XX XX XX XX XX XX XX,m

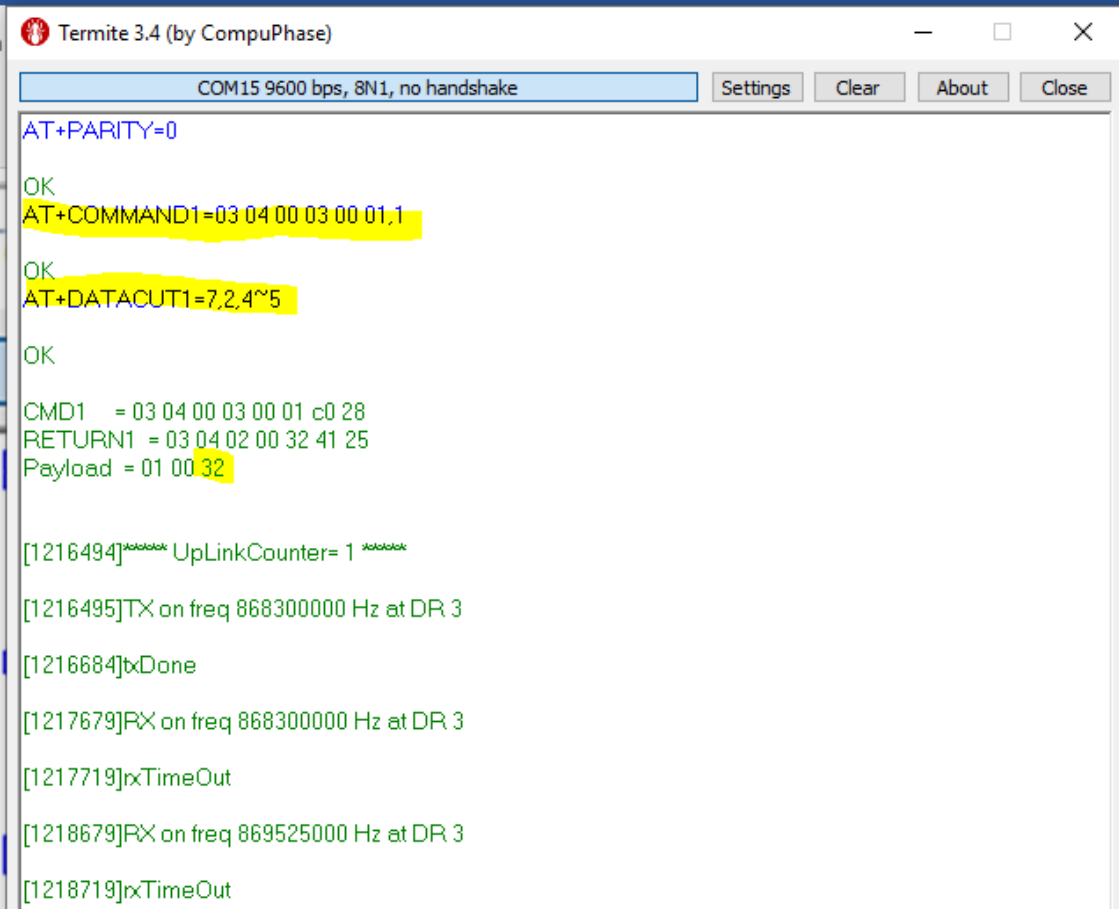
XX XX XX XX XX XX XX XX XX XX XX XX: The RS485 command to be sent

m: 0: no CRC, 1: add CRC-16/MODBUS in the end of this command

## AT+DATA CUT1=7,2,4~5

AT+DATA CUTx=a,b,c

- ✧ a: length for the return of AT+COMMAND
- ✧ b:1: grab valid value by byte, max 6 bytes. 2: grab valid value by bytes section, max 3 sections.
- ✧ c: define the position for valid value.



```
Termit 3.4 (by CompuPhase)
COM15 9600 bps, 8N1, no handshake  Settings  Clear  About  Close
AT+PARITY=0
OK
AT+COMMAND1=03 04 00 03 00 01,1
OK
AT+DATA CUT1=7,2,4~5
OK
CMD1  = 03 04 00 03 00 01 c0 28
RETURN1 = 03 04 02 00 32 41 25
Payload = 01 00 32

[1216494]***** UpLinkCounter= 1 *****
[1216495]TX on freq 868300000 Hz at DR 3
[1216684]txDone
[1217679]RX on freq 868300000 Hz at DR 3
[1217719]rxTimeOut
[1218679]RX on freq 869525000 Hz at DR 3
[1218719]rxTimeOut
```

It Works!

### APPLICATION DATA

Filters: uplink downlink activation ack error

|   | time     | counter | port |   |
|---|----------|---------|------|---|
| ▲ | 24:15:30 | 1       | 2    | payload: 01 00 32   |
| ▼ | 24:05:23 |         | 0    |   |
| ▲ | 24:05:22 | 0       | 2    | retry payload: 01   |
| ⚡ | 24:05:13 |         |      | dev addr: 26 01 58 AC app eui: A8 40 41 D1 71 82 1E 85 dev eui: A8 40 41 4C 21 82 1E 85 |
| ⚡ | 23:59:13 |         |      | dev addr: 26 01 48 98 app eui: A8 40 41 D1 71 82 1E 85 dev eui: A8 40 41 4C 21 82 1E 85 |

And even gets periodically each 10 minutes

### APPLICATION DATA

Filters: uplink downlink activation ack error

|   | time     | counter | port |                   |
|---|----------|---------|------|-------------------|
| ▲ | 24:25:21 | 2       | 2    | payload: 01 00 32 |
| ▲ | 24:15:30 | 1       | 2    | payload: 01 00 32 |

## APPLICATION DATA


Filters uplink downlink activation ack error

time counter port

▲ 24:25:21 2 2 payload: 01 00 32

### Uplink

#### Payload

01 00 32 

#### Fields

*no fields*

#### Metadata

```
{
  "time": "2020-09-18T22:25:21.206797562Z",
  "frequency": 868.3,
  "modulation": "LORA",
  "data_rate": "SF9BW125",
  "coding_rate": "4/5",
  "gateways": [
    {
      "gtw_id": "eui-b827ebffffe09b11d",
      "timestamp": 1327912652,
      "time": "",
      "channel": 1,
      "rssi": -71,
      "snr": 10.2
    }
  ]
}
```

#### Estimated Airtime

103.424 ms



### APPLICATION DATA

Filters: uplink downlink activation ack error

| time       | counter | port |                   |
|------------|---------|------|-------------------|
| ▲ 24:25:21 | 2       | 2    | payload: 01 00 32 |
| ▲ 24:15:30 | 1       | 2    | payload: 01 00 32 |

Now let's try to change the period from 60 seconds (default) to 10 seconds  
From the program terminal

```
AT+TDC=10000
OK
```

Nothing changes  
We have to reset or power off and on the Dragino device  
Now we have a periodo f 10 seconds

### APPLICATION DATA

Filters: uplink downlink activation ack error

| time       | counter | port |   |
|------------|---------|------|---|
| ▲ 01:08:27 | 3       | 2    | dev id: <a href="#">87654321</a> payload: 01 00 32  |
| ▲ 01:08:17 | 2       | 2    | dev id: <a href="#">87654321</a> payload: 01 00 32  |
| ▲ 01:08:07 | 1       | 2    | dev id: <a href="#">87654321</a> payload: 01 00 32  |
| ▼ 01:08:01 | 0       |      | dev id: <a href="#">87654321</a>  |
| ▲ 01:07:59 | 0       | 2    | dev id: <a href="#">87654321</a> payload: 01 00 32  |
| ⚡ 01:07:50 |         |      | dev id: <a href="#">87654321</a> dev addr: 26 01 26 F4 appeui: A8 40 41 D1 71 82 1E 85 dev eui: A8 40 41 4C 21 82 |
| ▲ 01:05:08 | 1       | 2    | dev id: <a href="#">87654321</a> payload: 01 00 32  |
| ▼ 24:55:12 | 0       |      | dev id: <a href="#">87654321</a>  |

Let's see how to change the programmed transmissi3n  
Now we will try read on register 4 (This is the preset trip current)

From serial Terminal  
AT+COMMAND1=03 04 00 04 00 01,1

Yes

```
AT+COMMAND1=03 04 00 04 00 01,1
```

```
OK
```

```
CMD1 = 03 04 00 04 00 01 71 e9
```

```
RETURN1 = 03 04 02 00 05 00 f3
```

```
Payload = 01 00 05
```

```
[1169427]***** UpLinkCounter= 117 *****
```

```
[1169428]TX on freq 867100000 Hz at DR 5
```

```
[1169484]txDone
```

```
[1170473]RX on freq 867100000 Hz at DR 5
```

```
[1170501]rxTimeOut
```

```
[1171479]RX on freq 869525000 Hz at DR 3
```

```
[1171519]rxTimeOut
```

Yes now we get 5 corresponding to 3.0 A  
Since the possible values are

|                                  |    |      |     |  |   |
|----------------------------------|----|------|-----|--|---|
| Corriente disparo Relé principal | Id | 0004 | R/W | 0.03 - 0.1 - 0.3 - 0.5 - 1 -<br>3 - 5 - 10 | A |
|----------------------------------|----|------|-----|--|---|

Now we change the preset on the leak circuit breaker relay to 0.03 (30mA)  
Yes, 0 is 30mA!

## APPLICATION DATA

Filters

|   | time     | counter | port | dev id:                  | payload: |
|---|----------|---------|------|--------------------------|----------|
| ▲ | 01:31:17 | 142     | 2    | <a href="#">87654321</a> | 01 00 00 |
| ▲ | 01:31:17 | 141     | 2    | <a href="#">87654321</a> | 01 00 05 |
| ▲ | 01:31:01 | 140     | 2    | <a href="#">87654321</a> | 01 00 05 |

Finally we disconnect the RS485 cable from The leakage realy RGU10 and we get the following result

## APPLICATION DATA

Filters

|   | time     | counter | port |  |
|---|----------|---------|------|--|
| ▲ | 01:41:17 | 203     | 2    | dev id: <a href="#">87654321</a> payload: 01 00 00 |
| ▼ | 01:41:10 |         | 0    | dev id: <a href="#">87654321</a>                   |
| ▲ | 01:41:08 | 201     | 2    | dev id: <a href="#">87654321</a> payload: 01 00 00 |
| ▼ | 01:41:09 |         | 0    | dev id: <a href="#">87654321</a>                   |
| ▲ | 01:40:57 | 200     | 2    | dev id: <a href="#">87654321</a> payload: 01 00 00 |
| ▼ | 01:40:55 |         | 0    | dev id: <a href="#">87654321</a>                   |
| ▲ | 01:40:53 | 199     | 2    | dev id: <a href="#">87654321</a> payload: 01 00 00 |
| ▲ | 01:40:37 | 198     | 2    | dev id: <a href="#">87654321</a> payload: 01 00 00 |

Obvious since there is no one listening at address 3  
But the payload is made of bytes 4 and 5

```
CMD1 = 03 04 00 04 00 01 71 e9  
RETURN1 = 00 00 00 00 00 00 00  
Payload = 01 00 00
```

```
[1909002]***** UpLinkCounter= 192 *****
```

```
[1909003]TX on freq 868300000 Hz at DR 5
```

```
[1909059]txDone
```

```
[1910048]FX on freq 868300000 Hz at DR 5
```

```
[1910077]rxTimeOut
```

```
[1911054]FX on freq 869525000 Hz at DR 3
```

```
[1911094]rxTimeOut
```

```
CMD1 = 03 04 00 04 00 01 71 e9  
RETURN1 = 00 00 00 00 00 00 00  
Payload = 01 00 00
```

## Variable Speed drive and Dragino RS485 to Lora for Monitoring

### Salicru CV30 VFD

Now we want to read the speed of a VFD

Our VFD i son address 1

9600 bauds

No Parity, 8 bits, 1 stop bit: N,8,1

Reading the speed on register:

|                        |       |                    |   |
|------------------------|-------|--------------------|---|
| Velocidad de operación | 3005H | Rango: 0~65535 RPM | R |
|------------------------|-------|--------------------|---|

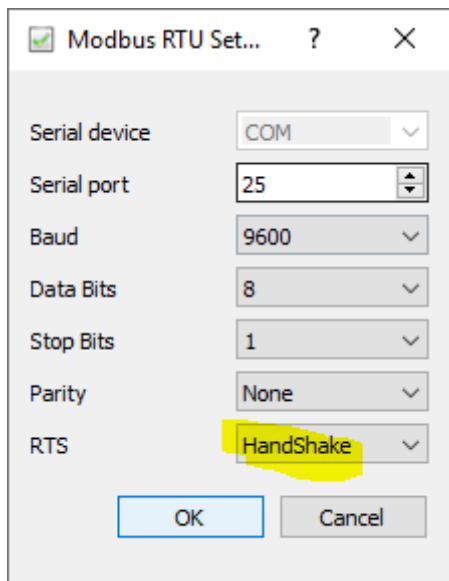
Parameter 12293 in decimal is 3005 in Hex

First we try with qModMaster terminal, in order to find out the right parameters

Connecting the VFD to the computer with a RS-485 to USB converter

Yes, we have the VFD stopped, with 0 speed as we see here 00 00 RPM

If yo do not set the RTS to Handshake you will get an error



### Raw Data

```
[RTU]> Tx > 11:20:40:475 - 01 03 30 05 00 01 9B 0B
[RTU]> Rx > 11:20:40:511 - 01 03 02 00 00 B8 44
[RTU]> Tx > 11:20:43:479 - 01 03 30 05 00 01 9B 0B
[RTU]> Rx > 11:20:43:502 - 01 03 02 00 00 B8 44
```

The we start the motor manually with a digital input on the drive (speed is controlled by Modbus)

We see

595 RPM (0253 Hex)

It Works!

The image shows two windows from the QModMaster software. The top window is the main configuration interface, and the bottom window is the Bus Monitor.

**QModMaster Configuration:**

- Modbus Mode: RTU
- Slave Addr: 1
- Scan Rate (ms): 3000
- Function Code: Read Holding Registers (0x03)
- Start Address: 12293 (Dec)
- Number of Registers: 1
- Data Format: Hex

The main display area shows the value **0253**.

**Bus Monitor Raw Data:**

```
[RTU]>Tx > 11:20:40:475 - 01 03 30 05 00 01 9B 0B
[RTU]>Rx > 11:20:40:511 - 01 03 02 00 00 B8 44
[RTU]>Tx > 11:20:43:479 - 01 03 30 05 00 01 9B 0B
[RTU]>Rx > 11:20:43:502 - 01 03 02 00 00 B8 44
[RTU]>Tx > 11:20:46:474 - 01 03 30 05 00 01 9B 0B
[RTU]>Rx > 11:20:46:509 - 01 03 02 02 54 B8 DB
[RTU]>Tx > 11:20:49:478 - 01 03 30 05 00 01 9B 0B
[RTU]>Rx > 11:20:49:501 - 01 03 02 02 52 38 D9
[RTU]>Tx > 11:20:52:489 - 01 03 30 05 00 01 9B 0B
[RTU]>Rx > 11:20:52:524 - 01 03 02 02 53 F9 19
```

ADU

So now we know the right command to programm to The Dragino RS-485 transactions

And this is:

AT+COMMAND1=01 03 30 05 00 01,1

AT+DATA CUT1=7,2,4~5 (But this was already programmed on the Dragino so we do not need to resend this AT command)

Yes, we have already programmed the Dragino, but the response is still 00000000 since we have not connected the VFD to the Dragino

```
AT+COMMAND1=01 03 30 05 00 01,1
OK
CMD1 = 01 03 30 05 00 01 9b 0b
RETURN1 = 00 00 00 00 00 00
Payload = 01 00 00

[209427]***** UpLinkCounter= 20 *****
[209428]TX on freq 867900000 Hz at DR 3
[209597]txDone
[210592]RX on freq 867900000 Hz at DR 3
[210632]rxTimeOut
```

Let's connect the VFD to Dragino RS-485-LN

Voilà,

First drive stopped, and the drive started at speed 02 54 in Hex (595 RPM in decimal)

| APPLICATION DATA  |          |         |      |          |          |
|---|----------|---------|------|----------|----------|
| Filters   |          |         |      |          |          |
| <input type="checkbox"/> uplink <input type="checkbox"/> downlink <input type="checkbox"/> activation <input type="checkbox"/> ack <input type="checkbox"/> error |          |         |      |          |          |
|   | time     | counter | port | dev id:  | payload: |
| ▲   | 11:41:32 | 57      | 2    | 87654321 | 01 02 54 |
| ▲   | 11:41:22 | 56      | 2    | 87654321 | 01 02 54 |
| ▲   | 11:41:12 | 55      | 2    | 87654321 | 01 02 55 |
| ▲   | 11:41:02 | 54      | 2    | 87654321 | 01 00 00 |
| ▲   | 11:40:52 | 53      | 2    | 87654321 | 01 00 00 |

CMD1 = 01 03 30 05 00 01 9b 0b  
RETURN1 = 01 03 02 00 00 b8 44  
Payload = 01 00 00

[429427]\*\*\*\*\* UpLinkCounter= 42 \*\*\*\*\*

[429428]TX on freq 868300000 Hz at DR 3

[429597]txDone

[430592]RX on freq 868300000 Hz at DR 3

[430632]rxTimeOut

[431592]RX on freq 869525000 Hz at DR 3

[431632]rxTimeOut

CMD1 = 01 03 30 05 00 01 9b 0b  
RETURN1 = 01 03 02 02 54 b8 db  
Payload = 01 02 54

## APPLICATION DATA

Filters

|   | time     | counter | port |  |
|---|----------|---------|------|--|
| ▲ | 11:41:32 | 57      | 2    | dev id: <a href="#">87654321</a> payload: 01 02 54 |
| ▲ | 11:41:22 | 56      | 2    | dev id: <a href="#">87654321</a> payload: 01 02 54 |
| ▲ | 11:41:12 | 55      | 2    | dev id: <a href="#">87654321</a> payload: 01 02 55 |
| ▲ | 11:41:02 | 54      | 2    | dev id: <a href="#">87654321</a> payload: 01 00 00 |
| ▲ | 11:40:52 | 53      | 2    | dev id: <a href="#">87654321</a> payload: 01 00 00 |



But let's decode the payload since it is in Hex



## APPLICATION DATA

Filters

|   | time     | counter | port |                                  |                   |          |
|---|----------|---------|------|----------------------------------|-------------------|----------|
| ▲ | 12:39:22 | 271     | 2    | dev id: <a href="#">87654321</a> | payload: 01 00 00 | rpm: 0   |
| ▲ | 12:39:12 | 270     | 2    | dev id: <a href="#">87654321</a> | payload: 01 02 54 | rpm: 596 |
| ▲ | 12:39:02 | 269     | 2    | dev id: <a href="#">87654321</a> | payload: 01 00 00 | rpm: 0   |

This is the payload decoder

## PAYLOAD FORMATS

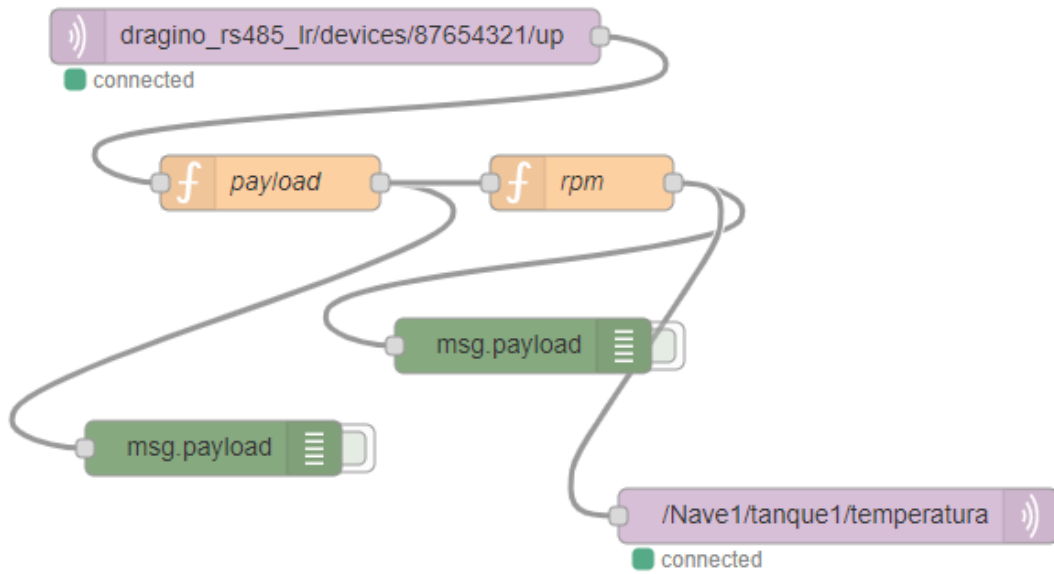
### Payload Format

The payload format sent by your devices

Custom

```
1 function Decoder(bytes, port) {  
2   // Decode an uplink message from a buffer  
3   // (array) of bytes to an object of fields.  
4   var decoded = {};  
5  
6   if (port === 2) decoded.rpm = bytes[1]*256+bytes[2];  
7  
8   return decoded;  
9 }
```

How to send the speed values to a mobile phone with “IoT On Off  
“ App



### Edit mqtt in node

Delete Cancel Done

**Properties** [Settings] [List] [Preview]

Server: eu.thethings.network [Edit]

Topic: dragino\_rs485\_lr/devices/87654321/up



QoS: 2 [Dropdown]

Output: auto-detect (string or buffer) [Dropdown]

Name: Name

Edit mqtt in node > **Edit mqtt-broker node**

Delete Cancel Update

**Properties**  

**Name**

**Connection** Security Messages

**Server**  Port

Enable secure (SSL/TLS) connection



**Client ID**

Keep alive time (s)   Use clean session

Use legacy MQTT 3.1 support

Edit mqtt in node > **Edit mqtt-broker node**

Delete Cancel Update

**Properties**  

**Name**




Connection **Security** Messages


**Username**


**Password**

**Edit function node**

Delete Cancel Done

**Properties**   




**Name**  


**Function** 


```
1 var msg1 = { payload: msg.payload.length };
2 msg1.payload = JSON.parse(msg.payload);
3 msg1.payload = msg1.payload.payload_fields;
4
5 return msg1;
```

**Edit function node**

Delete Cancel Done

**Properties**   

**Name**  

**Function** 

```
1 var a = msg.payload;
2 msg.payload=a.rpm;
3 return msg;
```

Delete
Cancel Done

⚙️ **Properties**

⚙️
📄
📏

🌐 Server

broker.hivemq.com:1883

✎

📄 Topic

/Nave1/tanque1/temperatura

⊕ QoS

▼

↺ Retain

▼

📌 Name

Name

Tip: Leave topic, qos or retain blank if you want to set them via msg properties.

You can see the video here

[https://www.youtube.com/watch?v=TAFZ5eaf-MY&t=6s&ab\\_channel=XavierFlorensaBerenguer](https://www.youtube.com/watch?v=TAFZ5eaf-MY&t=6s&ab_channel=XavierFlorensaBerenguer)

You can find the Node-RED code here:

<https://github.com/xavierflorensa/Salicru-VFD-Dracino-RS485-to-LoRaWAN-to-IOT-OnOff-App-Node-RED-flow>

How to change the speed from TTN downlink message injection

|   |    |         |
|---|----|---------|
| <a href="#">Configure AT+COMMANDx or AT+DATA CUTx</a>   | AF | Dynamic |
| <div style="display: flex; gap: 5px;"> <span style="border: 1px solid #ccc; padding: 2px 5px;">AF</span> <span style="border: 1px solid #ccc; padding: 2px 5px;">MM</span> <span style="border: 1px solid #ccc; padding: 2px 5px;">NN</span> <span style="border: 1px solid #ccc; padding: 2px 5px;">LL</span> <span style="border: 1px solid #ccc; padding: 2px 5px;">XX</span> <span style="border: 1px solid #ccc; padding: 2px 5px;">XX</span> <span style="border: 1px solid #ccc; padding: 2px 5px;">XX</span> <span style="border: 1px solid #ccc; padding: 2px 5px;">XX</span> <span style="border: 1px solid #ccc; padding: 2px 5px;">YY</span> </div> |    |         |

### Type Code 0xAF

0xAF downlink command can be used to set AT+COMMANDx or AT+DATACUTx.

Note: if user use AT+COMMANDx to add a new command, he also need to send AT+DATACUTx downlink.

Format: 

|    |    |    |    |             |    |
|----|----|----|----|-------------|----|
| AF | MM | NN | LL | XX XX XX XX | YY |
|----|----|----|----|-------------|----|

Where:

- ✧ MM: the ATCOMMAND or AT+DATACUT to be set. Value from 01 ~ 0F,
- ✧ NN: 0: no CRC; 1: add CRC-16/MODBUS ; 2: set the AT+DATACUT value.
- ✧ LL: The length of AT+COMMAND or AT+DATACUT command
- ✧ XX XX XX XX: AT+COMMAND or AT+DATACUT command
- ✧ YY: If YY=0, RS485-LN will execute the downlink command without uplink; if YY=1, RS485-LN will execute an uplink after got this command.

Example:

|    |    |    |    |                   |    |
|----|----|----|----|-------------------|----|
| AF | 03 | 01 | 06 | 0A 05 00 04 00 01 | 00 |
|----|----|----|----|-------------------|----|

: Same as AT+COMMAND3=0A 05 00 04 00 01,1

But first we have to simulate with qModbusMaster to check and see the right datagram

If we take a look at the VFD preset value which is located on parameter:

P17.00=19,95 Hz which corresponds to 596 Hz

## Grupo P17 – Función de Monitorización

|        |                        |   |
|--------|------------------------|---|
| P17.00 | Consigna de frecuencia | Muestra la consigna de frecuencia (frecuencia ajustada) actual del variador<br>Rango: 0.00Hz~P00.03 |
|--------|------------------------|---|

Since last written value on register 8193 was this speed.

8193 (=2001H)

|       |   |     |
|-------|---|-----|
| 2001H | Consigna de frecuencia por comunicación<br>(0~Fmax(unidad: 0.01Hz)) | W/D |
|-------|---|-----|

So we can write from qModbusMaster a new speed and then look at the change on parameter P17.00

Speed must be written in Hz so from 0 to 50

Then note the datagram needed

Unit is 0.01Hz

So we need to give the numer 4000 if we want to write 40 Hz (x100)

QModMaster

File Options **Commands** View Help

Modbus Mode **RTU** Slave Addr **1** Scan Rate (ms) **3000**

Function Code **Write Multiple Registers (0x10)** Start Address **8193** **Dec**

Number of Registers **2** Data Format **Dec** Signed

|   |   |   |      |   |   |   |   |   |   |
|---|---|---|------|---|---|---|---|---|---|
| x | x | x | 4000 | 0 | x | x | x | x | x |
|---|---|---|------|---|---|---|---|---|---|

RTU : \\.\COM25 | 9600,8,1,None Base Addr : 0 Packets : 7 Errors : 0

Bus Monitor

**Raw Data**

```
[RTU]>Tx > 00:05:33:772 - 01 10 20 01 00 01 02 0F A0 83 CB
[RTU]>Rx > 00:05:33:802 - 01 10 20 01 00 01 5B C9
Sys > 00:05:33:802 - values written correctly.
[RTU]>Tx > 00:05:35:349 - 01 03 20 01 00 02 9E 0B
[RTU]>Rx > 00:05:35:385 - 01 03 04 0F A0 00 00 F9 05
[RTU]>Tx > 00:05:35:403 - 01 03 20 01 00 02 9E 0B
[RTU]>Rx > 00:05:35:433 - 01 03 04 0F A0 00 00 F9 05
```

**ADU**

```
Type : Tx Message
Timestamp : 00:05:33:772
Slave Addr : 01
Function Code : 10
Starting Address : 2001
Quantity of Registers : 0001
Byte Count : 02
Output Values : 0F A0
CRC : 83CB
```

QModMaster

File Options Commands View Help

Modbus Mode RTU Slave Addr 1 Scan Rate (ms) 3000

Function Code Write Multiple Registers (0x10) Start Address 8193 Dec

Number of Registers 2 Data Format Dec Signed

|   |   |   |      |   |   |   |   |   |   |
|---|---|---|------|---|---|---|---|---|---|
| x | x | x | 4000 | 0 | x | x | x | x | x |
|---|---|---|------|---|---|---|---|---|---|

RTU : \\.\COM25 | 9600,8,1,None Base Addr : 0 Packets : 7 Errors : 0

Bus Monitor

Raw Data

```
[RTU]>Tx > 00:05:33:772 - 01 10 20 01 00 01 02 0F A0 83 CB
[RTU]>Rx > 00:05:33:802 - 01 10 20 01 00 01 5B C9
Sys > 00:05:33:802 - values written correctly.
[RTU]>Tx > 00:05:35:349 - 01 03 20 01 00 02 9E 0B
[RTU]>Rx > 00:05:35:385 - 01 03 04 0F A0 00 00 F9 05
[RTU]>Tx > 00:05:35:403 - 01 03 20 01 00 02 9E 0B
[RTU]>Rx > 00:05:35:433 - 01 03 04 0F A0 00 00 F9 05
```

ADU

```
Type : Rx Message
Timestamp : 00:05:33:802
Slave Addr : 01
Function Code : 10
Starting Address : 2001
Quantity of Registers : 0001
CRC : 5BC9
```

What do we have now on parameter P17.00?

39.96

Voilà



So these are the datagrams

01 10 20 01 00 01 02 0F A0 83 CB

01 is slave address

10 is Modbus write type in hex (16 in Decimal)

2001 is the speed preset register address

0001 is only one byte to be written

02 is

0FA0 is 4000 in decimal

Rest is CRC

So

01 10 20 01 00 01 02 0F A0

But we will try to write 30 Hz (3000 or BB8 in Hex)

So we will try with

01 10 20 01 00 01 02 0B B8

So Payload for downlink must be

Example:

**AF 03 01 06 0A 05 00 04 00 01 00**: Same as AT+COMMAND3=0A 05 00 04 00 01,1

In our case

AF 01 01 09 01 10 20 01 00 01 02 0B B8 00

We try manually from TTN

**DOWNLINK**

**Scheduling** replace first last **FPort**   **Confirmed**

**Payload** bytes fields  14 bytes

| APPLICATION DATA |         |          |            |     |  |
|------------------|---------|----------|------------|-----|--|
| Filters          |         |          |            |     |  |
|                  | uplink  | downlink | activation | ack | error  |
| time             | counter | port     |            |     |  |
| ▲ 24:24:56       | 811     | 2        |            |     | payload: 01 01 00 rpm: 256                         |
| ▼ 24:24:48       |         | 1        |            |     | payload: AF 01 01 09 01 10 20 01 00 01 02 0B B8 00 |
| ▲ 24:24:46       | 810     | 2        |            |     | payload: 01 00 00 rpm: 0                           |
| ▼ 24:24:45       |         | 1        | scheduled  |     | payload: AF 01 01 09 01 10 20 01 00 01 02 0B B8 00 |
| ▲ 24:24:36       | 809     | 2        |            |     | payload: 01 00 00 rpm: 0                           |
| ▲ 24:24:26       | 808     | 2        |            |     | payload: 01 00 00 rpm: 0                           |

Let's see what do we have on register P17.00

Voilà, 29,95

And if we start the drive

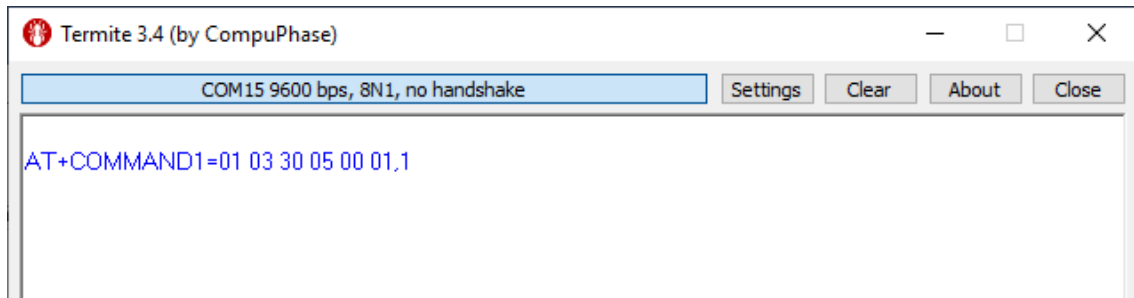
But RPM should be 0 Now ¿

|            |     |   |           |  |  |
|------------|-----|---|-----------|--|--|
| ▲ 24:25:26 | 814 | 2 |           |  | payload: 01 01 00 rpm: 256                         |
| ▲ 24:25:16 | 813 | 2 |           |  | payload: 01 01 00 rpm: 256                         |
| ▲ 24:25:06 | 812 | 2 |           |  | payload: 01 01 00 rpm: 256                         |
| ▲ 24:24:56 | 811 | 2 |           |  | payload: 01 01 00 rpm: 256                         |
| ▼ 24:24:48 |     | 1 |           |  | payload: AF 01 01 09 01 10 20 01 00 01 02 0B B8 00 |
| ▲ 24:24:46 | 810 | 2 |           |  | payload: 01 00 00 rpm: 0                           |
| ▼ 24:24:45 |     | 1 | scheduled |  | payload: AF 01 01 09 01 10 20 01 00 01 02 0B B8 00 |
| ▲ 24:24:36 | 809 | 2 |           |  | payload: 01 00 00 rpm: 0                           |
| ▲ 24:24:26 | 808 | 2 |           |  | payload: 01 00 00 rpm: 0                           |

And if we start the drive, the speed is not updated

**This means we have to write again the original read command after a downlink message**

To recover let's give the command on the USB to TTL programmer



Even though we do not get the OK response, the Dragino is now configured to send the speed.

Then you have to push the reset button

And disconnect the Dragino from power

And speed is 10Hz since we have changed with a downlink message

We are ready to change the speed from the downlink

Next step is to do this from Node-RED

How to change the speed from a mobile phone