

Manual LoRaWAN DIN-rail module

Date: 24-04-2017
Version: 1.0
Title: Manual LoRaWAN DIN-rail module

1 Revision history

Version	Date	Changes
1.0	24-04-2017	First release

1.1 Contact

IdeeTron B.V.
Dropsstraat 81
3941 JL Doorn
Nederland

Tel: +31 (0)343 769094
E-mail: info@ideetron.nl

2 Table of contents

1	REVISION HISTORY	2
1.1	Contact	2
2	TABLE OF CONTENTS	2
3	INTRODUCTION	4
4	CONNECTIONS	4
5	SPECIFICATIONS	5
6	DEFAULT SETTINGS	5
7	LORAWAN INTRODUCTION	6
8	OPERATION	6
9	COMMANDS	8
9.1	mac data [data]	8
9.2	mac join	9
9.3	mac set/get devaddr	9

9.4	mac set/get nwkskey	9
9.5	mac set/get appskey	10
9.6	mac set/get drrx	10
9.7	mac set/get drtx	10
9.8	mac set/get chrX	11
9.9	mac set/get chtx	11
9.10	mac set/get pwrIdx	11
9.11	mac set/get cnf	12
9.12	mac set/get chop	12
9.13	mac set/get appkey	12
9.14	mac set/get deveui	13
9.15	mac set/get appeui	13
9.16	Input get/set mask	13
9.17	Input get/set dataX	14
10	EXAMPLES	15
10.1	Activation By Personalisation (ABP)	15
10.2	Over The Air Activation with Semtech (OTAA)	19
10.3	Over The Air Activation with The Things Network (OTAA)	21

3 Introduction

This documentation describes the installation and use of the LoRaWAN DIN-rail module. The module is fitted with a serial communication port and two digital inputs.

4 Connections

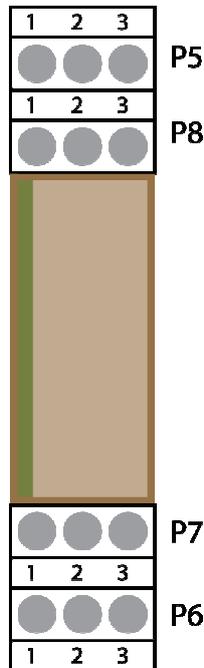


Figure 1: Connections of module

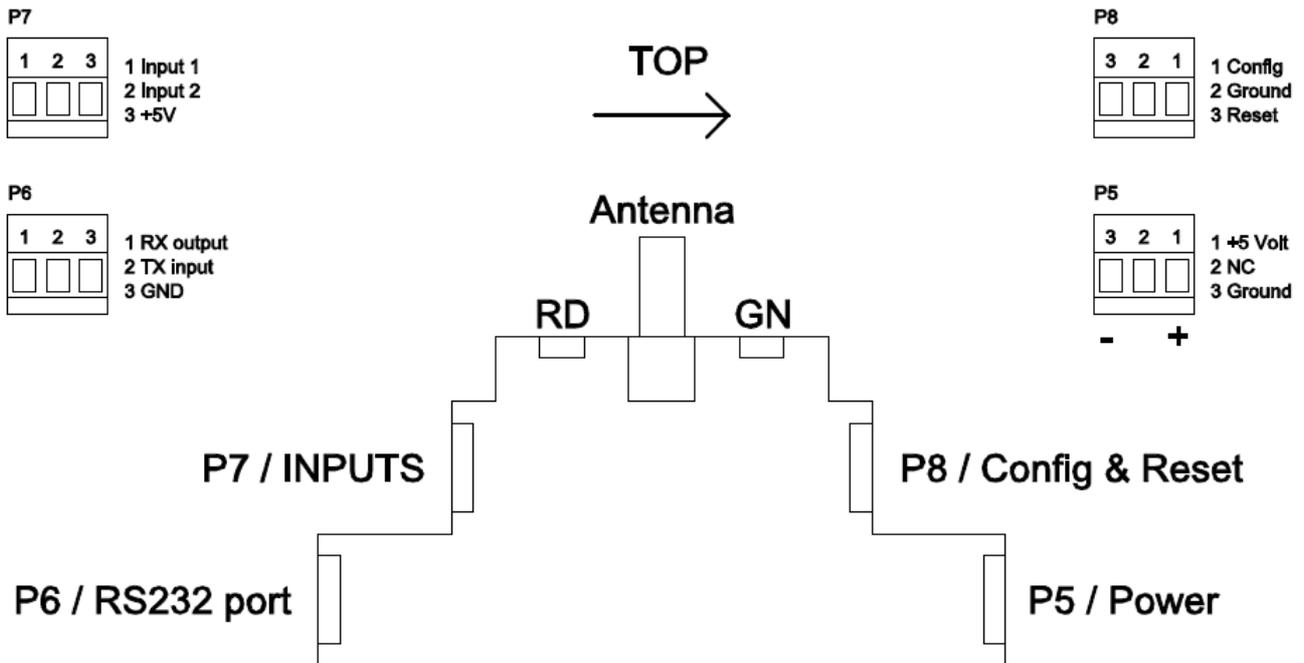


Figure 2: Connections of module

5 Specifications

Power supply	Value
Supply voltage	5 – 15 Vdc
Voltage inputs	5 – 15 Vdc
Serial connection	Value
Datarate	19200 bps
Number of data bits	8
Parity	None
Number of stop bits	1

Table 1: Specifications

6 Default settings

The following settings are the default setting on start-up of the module.

Radio connection	
Modulation	LoRa
Channel Tx	868.100 MHz
Datarate Tx	SF12 BW 125 kHz
Channel Rx	869.525 MHz
Datarate Rx	SF9 BW 125 kHz
Device Address	4 last bytes of DS2401 unique number
Network session key	2B7E151628AED2A6ABF7158809CF4F3C
Application session key	2B7E151628AED2A6ABF7158809CF4F3C
Confirm	No
Channel Hopping	Yes
DevEUI	1D EE and 6 bytes of DS2401
AppEUI	00 00 00 00 00 00 00 00
Application key	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Input mask	03
Data0	00
Data1	01
Data2	02
Data3	03

Table 2: Default settings

See chapter 9 if you want to change these settings.

7 LoRaWAN introduction

The LoRaWAN communication takes place in two directions, up (from the module to gateway) and down (from gateway to the module).

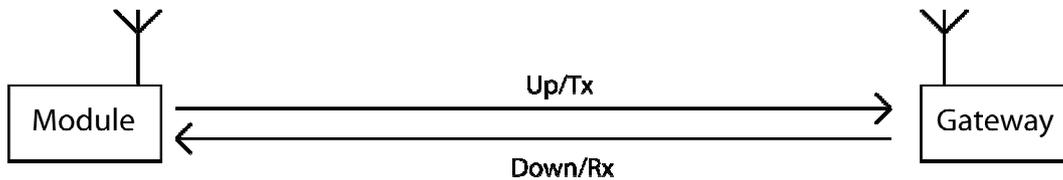


Figure 3: LoRaWAN communication

The channel and datarate for the both directions can be set.

The module is a type C mote can receive a message from the gateway any time, except when the module is transmitting a message.

8 Operation

The module can send and receive messages using the LoRaWAN network. Using the serial interface described in chapter 9, it is possible to make different settings and send data.

Depending on the settings of the input mask the module will also send a message when one of the inputs changes. The data send in the message is adjustable with the input commands.

When the module receives a message it will check if the CRC and MIC is ok and if it is a message for this device. Then it will be broken down in the different fields. The module will send the following information using to serial communication:

Message	Value	Description
CRC	OK/NOK	If the CRC of the received radio message is ok.
MIC	OK/NOK	If the MIC calculation of the LoRaWAN package is ok.
Address	OK/NOK	If the package contains the address for this mote
MAC Header	See Table 5	LoRaWAN MAC header
Sensor	4 bytes in Hex	Device address
Frame counter	2 bytes in Hex	Number of frame
Frame control	00 or 20	LoRaWAN Fctrl field
Data	Bytes in Hex	Data send from mote or gateway

Table 3: Message information

The MAC Header specifies the message type and the version of the LoRaWAN protocol used.

Bits	7 – 5	4 – 2	1 – 0
Field	Message type	Reserved	LoRaWAN protocol

Table 4: MAC Header fields

The LoRaWAN protocol field will always be 00.
The reserved field bits will always be 0.

The message type field can have several values, in Table 5 the different values for the MAC header are listed.

MAC Header value Hex	Message type
60	Unconfirmed data down
A0	Confirmed data down

Table 5: Message types

The frame control field consists of different bits, the most important is bit 5 containing the ACK. Most of the times the other bits will be 0, so the frame control field can have the following values:

00 – No ACK
20 – ACK

For a full description of the frame control field see the LoRaWAN specification.

When the module receives a Join Accept message after sending a Join request the following data will be shown:

- Device address
- Network session key
- Application session key

9 Commands

This chapter describes the commands that can be used to communicate with the module using a terminal program. All commands and data is sent as ASCII characters. All data is one or more bytes of hexadecimal numbers represented by 2 ASCII characters. So all data exist of 2 ASCII characters that can be 0 – F. All other characters will be set to 0.

The following command structure is used

Command type	Parameter 1	Parameter 2	Value	Description	
mac	data		Maximum of 51 bytes	Send LoRaWAN message with this data	
	join		-	Send a Join Request message	
	get/set	devaddr		4 bytes	The the device address
		nwkskey		16 bytes	The network session key
		appskey		16 bytes	The application session key
		drxr		1 byte	Transmit datarate
		drtx		1 byte	Receive datarate
		chrx		1 byte	Transmit channel
		chtx		1 byte	Receive channel
		pwridx		1 byte	Power index
		cnf		1 byte	Confirmation
		chhop		1 byte	Channel hopping
		appkey		16 bytes	Application key
		deveui		8 bytes	Device EUI
appeui		8 bytes	Application EUI		
input	get/set	mask	1 byte	Input mask	
		data0	1 byte	Data input value 00	
		data1	1 byte	Data input value 01	
		data2	1 byte	Data input value 10	
		data3	1 byte	Data input value 11	

Table 6: Command structure

The command type, parameters and values are separated by a space.

9.1 mac data [data]

This command is used to send a LoRaWAN data message containing the data that is included in this command. The maximum number of bytes that a LoRaWAN message can contain is 51 bytes.

Example: mac data 00FF

The answer from the module on this command will be:

Data: [data]

9.2 mac join

This command is used to send a LoRaWAN join request message. The join request message will use the following parameters:

- appkey
- deveui
- appeui

The join request message will use the default Datarate and Channel for transmit. And it will listen on the set receive Datarate and Channel for a Join accept message.

The join accept message will be received about 6 sec after the join request message is sent. When a join accept message is received it will show the following

DevAddr: [device address]
NwkSKey: [network session key]
AppSKey: [application session key]

9.3 mac set/get devaddr

This command is used to read or write the device address.

To read the device address sent: mac get devaddr
To write the device address sent: mac set devaddr [device address]

[device address] is 4 bytes long: 00112233

The answer from the module is:

DevAddr: [device address]

9.4 mac set/get nwkskey

This command is used to read or write the network session key.

To read the key sent: mac get nwkskey
To write the key sent: mac set nwkskey [key]

[key] is 16 bytes long: 00112233445566778899AABBCCDDEEFF

The answer from the module is:

NwkSKey: [key]

9.5 mac set/get appskey

This command is used to read or write the application session key.

To read the key sent: mac get appskey

To write the key sent: mac set appskey [key]

[key] is 16 bytes long: 00112233445566778899AABBCCDDEEFF

The answer from the module is:

AppSKey: [key]

9.6 mac set/get drrx

This command is used to read or write the receive datarate.

To read the datarate sent: mac get drrx

To write the datarate sent: mac set drrx [datarate]

[datarate] can have to values shown in Table 7.

[datarate]	Description
00	SF 12 BW 125 kHz
01	SF 11 BW 125 kHz
02	SF10 BW 125 kHz
03	SF9 BW 125 kHz
04	SF8 BW 125 kHz
05	SF7 BW 125 kHz
06	SF7 BW 250 kHz

Table 7: Datarate values

The answer from the module is:

Datarate Rx: [datarate]

9.7 mac set/get drtx

This command is used to read or write the transmit datarate.

To read the datarate sent: mac get drtx

To write the datarate sent: mac set drtx [datarate]

[datarate] can have to values shown in Table 7.

The answer from the module is:

Datarate Tx: [datarate]

9.8 mac set/get chrx

This command is used to read or write the receive channel.

To read the channel sent: mac get chrx

To write the channel sent: mac set chrx [channel]

[channel] can have to values shown in Table 8.

The answer form the module is:

Channel Rx: [channel]

Value	Channel (MHz)
00	868.100
01	868.300
02	868.500
03	867.100
04	867.300
05	867.500
06	867.700
07	867.900
10	869.525

Table 8: Channel values

9.9 mac set/get chtx

This command is used to read or write the transmit channel.

To read the channel sent: mac get chtx

To write the channel sent: mac set chtx [channel]

[channel] can have to values shown in Table 8.

The answer form the module is:

Channel Tx: [channel]

9.10 mac set/get pwridx

This command is used to set the transmit power. A higher transmit power will use more energy but you can communicate over greater distance.

To read the transmit power sent: mac get pwridx

To write the transmit power sent: mac set pwridx [power]

[power] can have a value from 00 to 0F, where 00 is the lowest power and 0F the highest power.

The answer form the module is:

Power: [power]

9.11 mac set/get cnf

With this command you can set the message type of the LoRaWAN message. It can be set to a confirmed or unconfirmed message. This determines if you request an acknowledge from the LoRaWAN service.

To read confirmation sent: mac get cnf

To write confirmation sent: mac set cnf [Confirm]

[Confirm] can have the following values:

00 Unconfirmed message type

01 Confirmed message type

The answer from the module is:

Confirm: [Confirm]

9.12 mac set/get chop

With this command you can chose if channel hopping is activated. When channel hopping is activated every data message will be send on the next channel. If the last message was sent on channel 07 it will go back to channel 00. See Table 8 for the different channels

To read channel hopping sent: mac get chhop

To write channel hopping sent: mac set chhop [Channel hopping]

[Channel hopping] can have the following values:

00 No channel hopping

01 Channel hopping active

The answer from the module is:

Channel Hopping: [Channel hopping]

9.13 mac set/get appkey

This command is used to read or write the application key.

To read the key sent: mac get appkey

To write the key sent: mac set appkey [key]

[key] is 16 bytes long: 00112233445566778899AABBCCDDEEFF

The answer from the module is:

AppKey: [key]

9.14 mac set/get deveui

This command is used to read or write the device EUI.

To read the EUI sent: mac get deveui
To write the EUI sent: mac set deveui [EUI]

[EUI] is 8 bytes long: 0011223344556677

The answer from the module is:

DevEUI: [EUI]

9.15 mac set/get appeui

This command is used to read or write the application EUI.

To read the EUI sent: mac get appeui
To write the EUI sent: mac set appeui [EUI]

[EUI] is 8 bytes long: 0011223344556677

The answer from the module is:

AppEUI: [EUI]

9.16 Input get/set mask

This command is used to read or write the input mask. The mask determines which inputs are active. When the corresponding bit for an input is set, the input is active. To turn off the inputs set the mask to 00.

Bit 7-2	Bit 1	Bit 0
Not used	Input 2	Input 1

Table 9: Input mask bits

To read the input mask sent: input get mask
To write the input mask sent: input set mask [mask]

[mask] is 1 byte long: 00 to 03

The answer from the module is:

Input masks: [mask]

9.17 Input get/set dataX

When the inputs from the module changes, the module will sent a LoRaWAN messages containing one byte. Which byte is send depends on the status of the inputs and the value of the input mask. The value of the different bytes can be set with this command.

There a two inputs so the input can have 4 different states.

Input 2	Input 1	State	Data
0	0	0	Data0
0	1	1	Data1
1	0	2	Data2
1	1	3	Data3

Table 10: Input states

When the state of the input changes from one state to the other the module will send a message containing the corresponding data of that state.

When an input changes, the value of the mask will be checked if there will be any action.

Example

Start: Both inputs are low and the mask is set to 02.

Now input 2 change to high. The module will now send a message containing the value of Data2.

Now input 1 change to high. The module will take no action because the mask for input 1 is not set.

To read the data sent: input get dataX

To write the data sent: input set dataX [data]

The answer from the module is:

Input X data: [data]

Example

To change Data1 to 0E, sent: input set data1 0E

10 Examples

The examples written in this chapter are using the Semtech development site: iot.semtech.com and the site of The Thing Network: <https://account.thethingsnetwork.org/users/login>. This site is free and can be fully used after registration. The terminal program used is RealTerm, that can be downloaded here:

<https://sourceforge.net/projects/realterm/files/Realterm/2.0.0.70/>

10.1 Activation By Personalisation (ABP)

Activation by personalisation means that the following parameters are set in the mote:

- DevAddr
- NwkSKey
- AppSKey

To add a mote whit ABP to semtech follow these steps:

1. Sign in or create an account on iot.semtech.com
2. On the left side go to applications
3. Then click the number behind the defaultApp.

Home \ Applications

- ▶ Home
- ▶ First Steps
- ▶ Applications
- ▶ Motes
- ▶ Gateways
- ▶ Network Activity
- ▶ Network Map

Applications

Below is a list of LoRa applications on the network. Use the fields at the top to set up a new one on the server.

Name ▲	Owner	EUI (AppEUI)	Configured Motes
New: <input type="text"/> <input type="text"/> <input type="text"/> <input type="button" value="Add"/>			
defaultApp	[Unknown]	00-00-00-00-00-00-00	54 <input type="button" value="📄"/>
PublicApp_2	Semtech	00-16-C0-00-00-00-01	0 <input type="button" value="📄"/>
Test1234	Hiddink	1D-EE-00-00-00-00-01	<input type="button" value="Delete"/> 0 <input type="button" value="📄"/>

Figure 4: Selecting default app

4. To add the device you need to know the NwkSKey, AppSKey and DevAddr. For this example we use the default key's
 NwkSkey: 2b7e151628aed2a6abf7158809cf4f3c
 AppSkey: 2b7e151628aed2a6abf7158809cf4f3c
5. To get the Device address from the module you need to connected the module and start realterm.
6. Then goto the tab port and make the following settings:
 Set baudrate to 9600
 Select the right port
 Toggle the open button

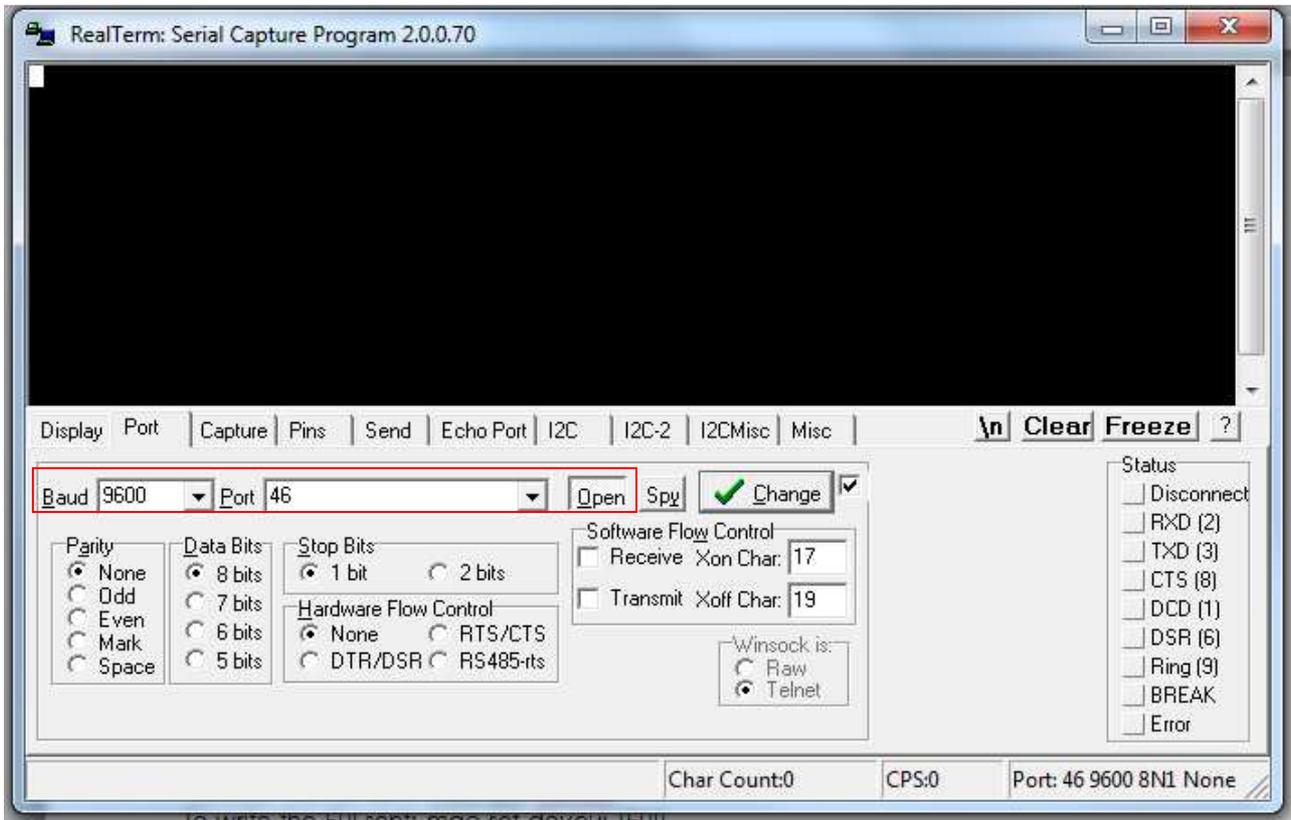


Figure 5: Opening a port with realterm

7. Then goto the tab Send.
8. Type the command: "mac get devaddr" in the field
9. Hit the Send ASCII button

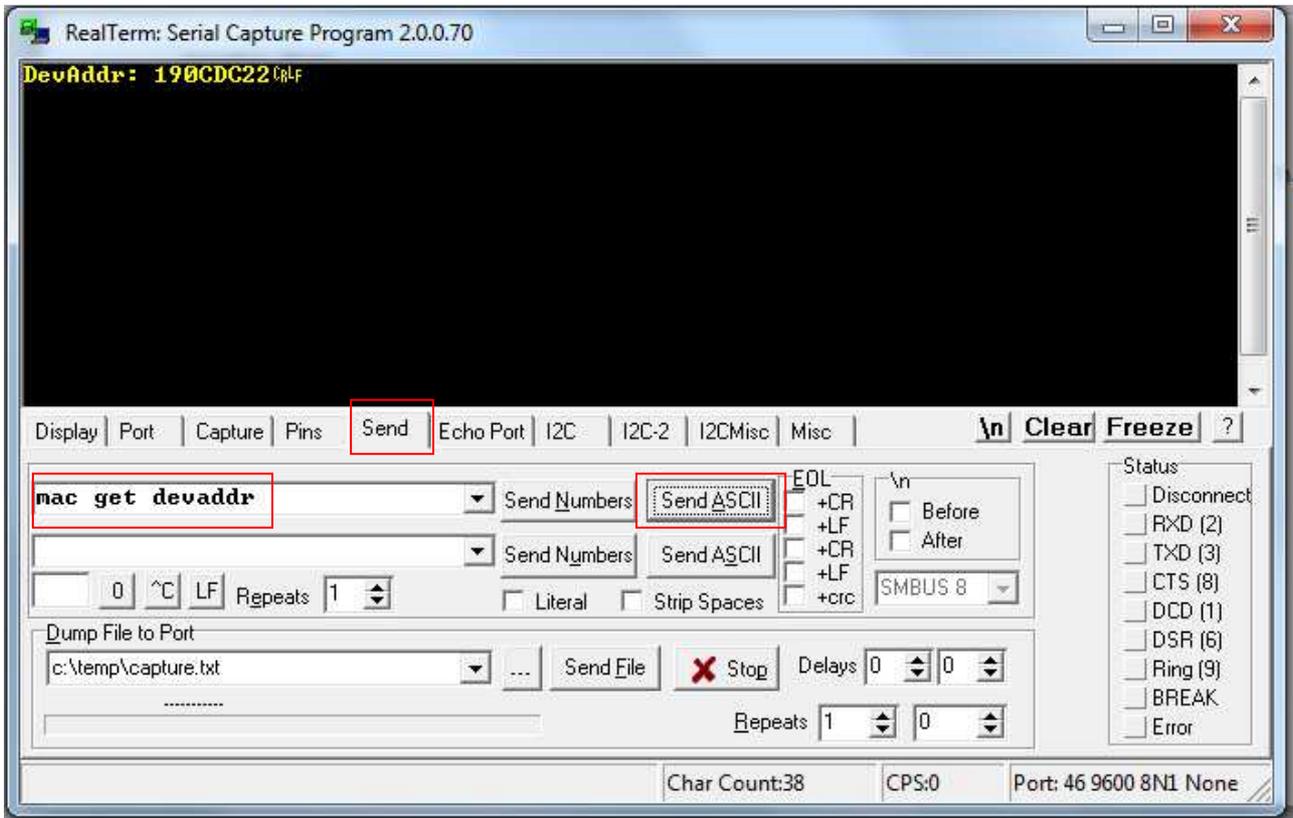


Figure 6: Sending command

10. Now you have all the information you need
11. Go back to the Semtech site and fill in the information into the correct fields
12. Then hit the add button

Network Address (DevAddr)	Application Session Key (AppSKey)	Network Session Key (NwkSKey)	Class
190CDC22	2b7e151628aed2a6abf7158809cf4f3c	2b7e151628aed2a6abf7158809cf4f3c	<input checked="" type="radio"/> AB <input type="radio"/> C

Figure 7: Add to semtech

13. Your mote is now added to the list below, find it and click on the number
14. Now go back to Realterm and send a message by using the mac data command

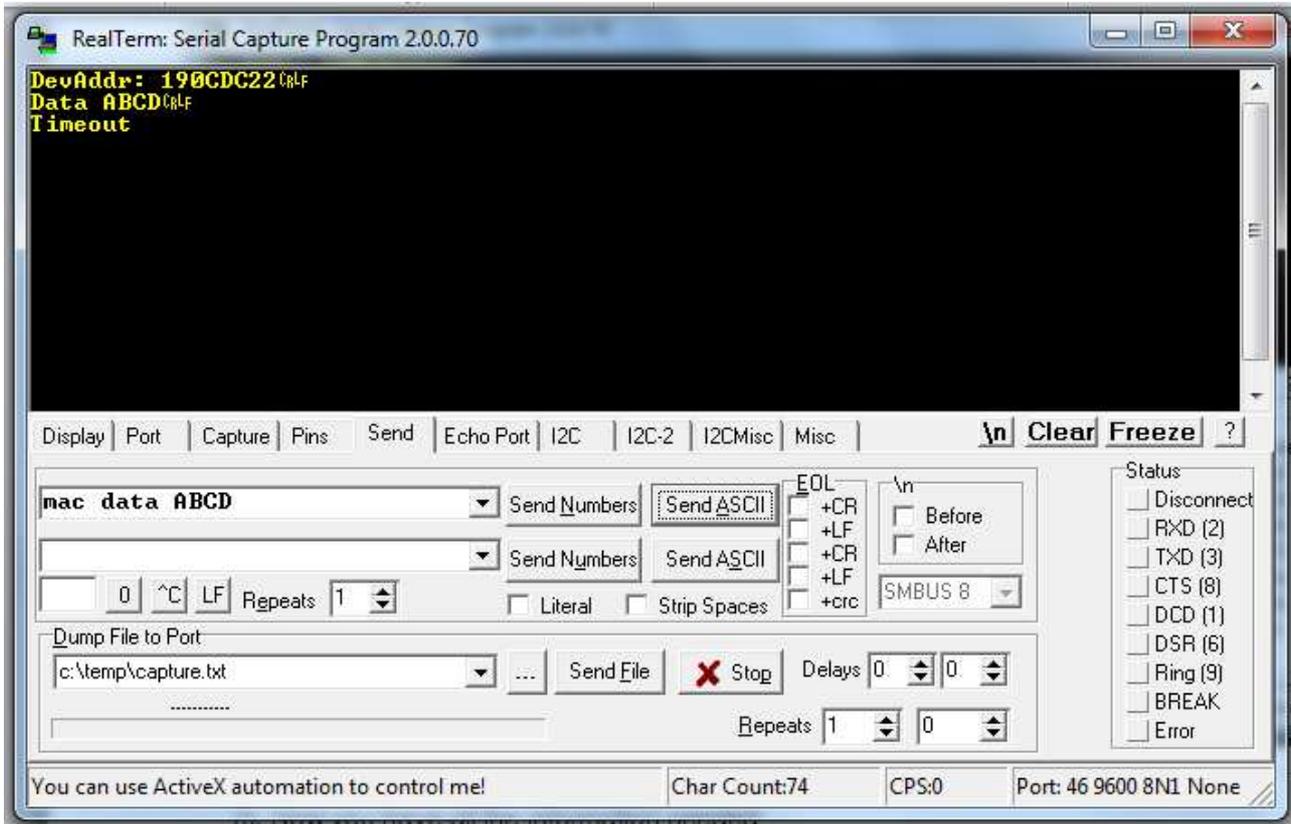


Figure 8: Send data command

15. Now go back to the semtech site and check if the message has arrived. This may take some time +/- 1 min
16. To see the data you can click "View data from application defaultApp"
17. To see the radio settings for a message click "View transmission performance"

LoRaMote 00-00-00-00-19-0C-DC-22

Network address: 19:0C:DC:22

Class: A/B

Application: defaultApp (00-00-00-00-00-00-00-00)

Owner: [Unknown]

[View transmission performance](#)

[View data from application defaultApp](#)

Figure 9: View message

Port	Time	Sequence #	Application Data
1	2017-02-16 10:57:12	0	ab cd

Figure 10: Received data

Sequence #	Freq (MHz)	Modulation	BW (Hz)	SF	Coding Rate	ADR	Gateway	Time	Precise	Chan	RSSI (dBm)	SNR (dB)
0	868.1	LoRa	125000	SF12	4/5	off	1D-EE-12-A4-A2-DF-A6-A6:0	2017-02-16 10:57:12	N/A	0	-76	9

Figure 11: Transmission performance

10.2 Over The Air Activation with Semtech (OTAA)

Over the Air Activation means that the following parameters are set in the mote:

- AppKey
- DevEUI
- AppEUI

During the activation the mote will receive this session data:

- DevAddr
- NwkSkey
- AppSkey

Use the following settings to join

Description	Setting	Value
Transmit channel	chtx	00
Transmit datarate	drtx	00 – 02
Receive channel	chrx	10
Receive datarate	drx	03

Table 11: Channel and datarate settings Semtech

To add a mote whit OTAA to semtech follow these steps:

1. Sign in or create an account on iot.semtech.com
2. On the left side go to applications
3. Now add a new application to fill in the fields and press add

Name	Owner	EUI (AppEUI)	Configured Motes
New: <input type="text" value="Ideetron"/>	<input type="text" value="Ideetron"/>	<input type="text" value="1DEE000000000002"/>	<input type="button" value="Add"/>

Figure 12: Create new application

4. Now click the number behind the new application
5. Now add an over the air mote. The DevEUI and AppEUI are generated with the website www.random.org

Mote (DevEUI)	Application Key (AppKey)
New: <input type="text" value="9b684af9c677f687"/>	<input type="text" value="aaf464807ab3904e3b18119533f377f2"/>

Figure 13: Add a mote

6. Now start up realterm term and set the parameters

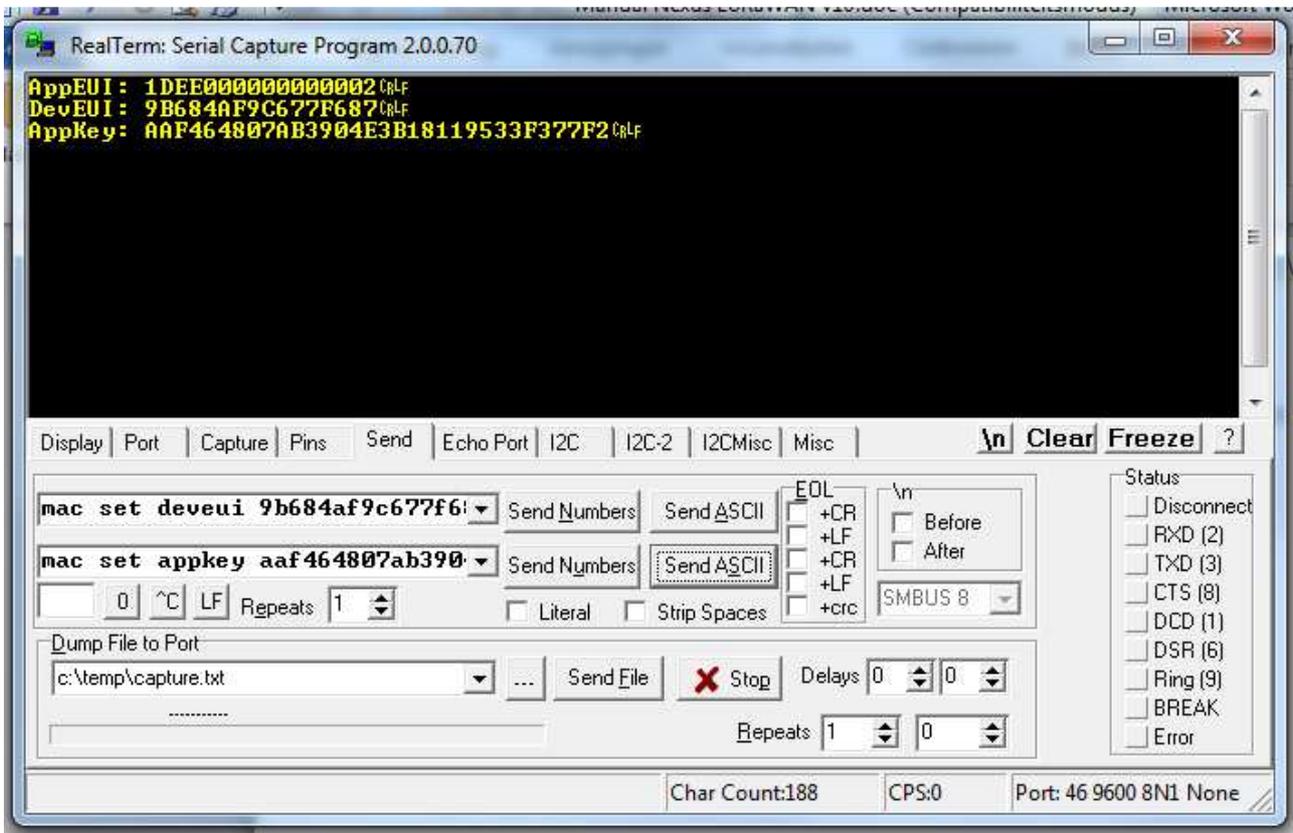


Figure 14: Set parameters

7. Send the join command and wait for a reply

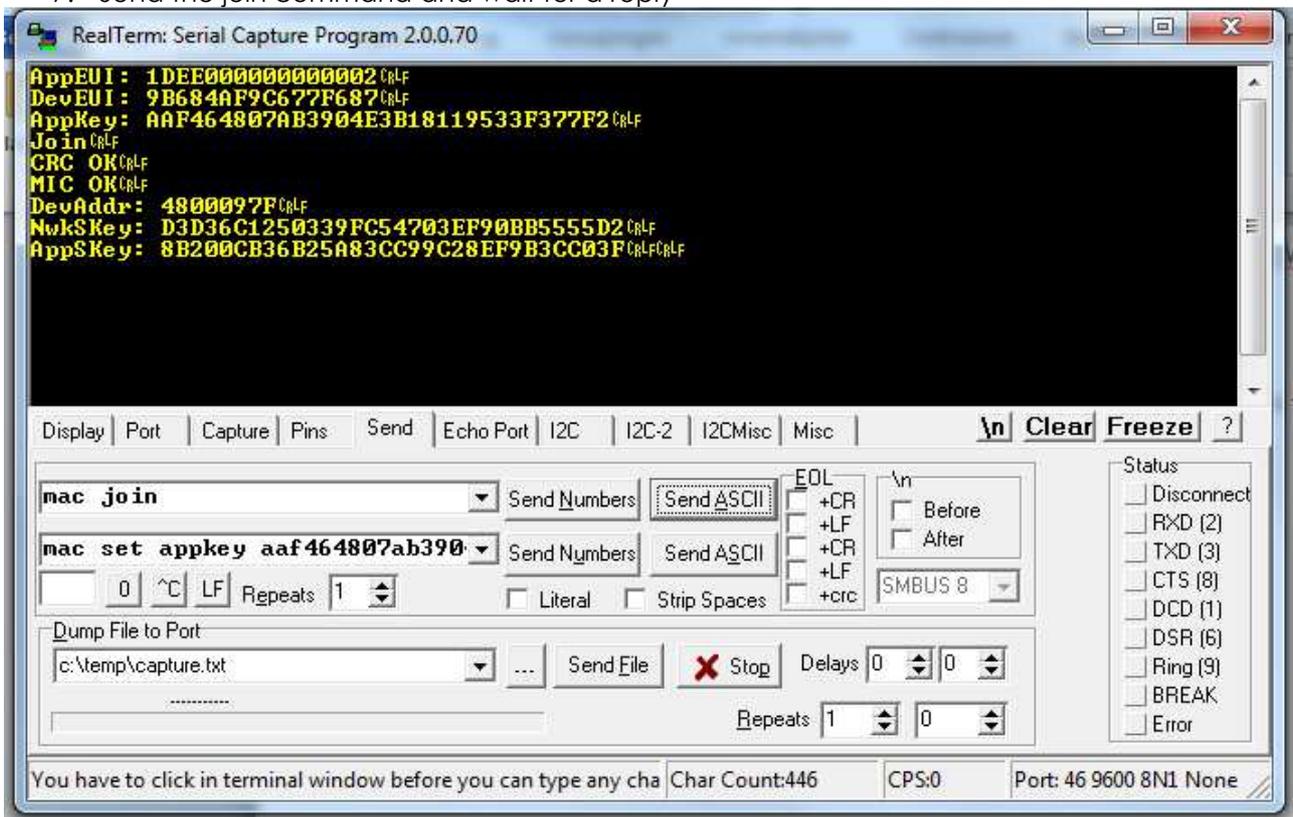


Figure 12: OTAA response

8. Now the mote is joined, try to send some data with the data command
9. Check the received data in semtech

10.3 Over The Air Activation with The Things Network (OTAA)

Over the Air Activation means that the following parameters are set in the mote:

- AppKey
- DevEUI
- AppEUI

During the activation the mote will receive this session data:

- DevAddr
- NwkSkey
- AppSkey

Use the following settings to join:

Description	Setting	Value
Transmit channel	chtx	00
Transmit datarate	drtx	00 – 02
Receive channel	chrx	10
Receive datarate	drx	00

Table 12: Channel and datarate settings TTN

To add a mote whit OTAA to TTN follow these steps:

1. Sing in or create an account on the TTN site.
2. Goto Applications.
3. Click “add application”.

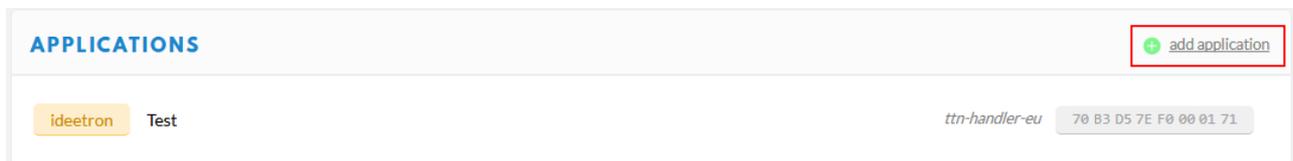


Figure 15: Create new application

4. Fill in the fields and press “Add application”.
5. After registering the application you will see a screen with all the information regarding the application. Now click “register device”.

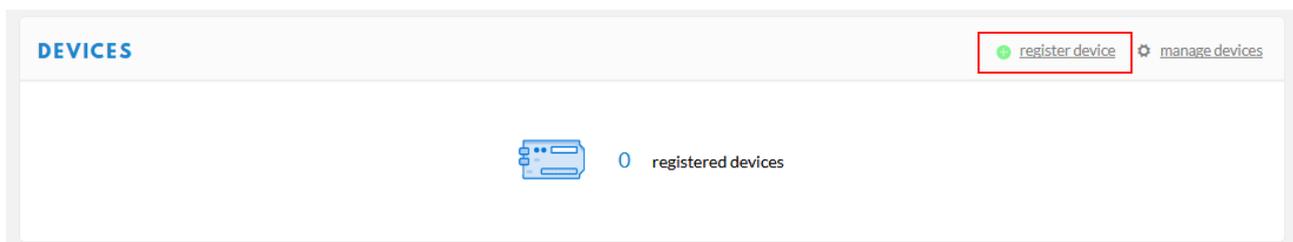


Figure 16: Register a device

6. Fill in the Device ID. Set the Device EUI to be generated.

REGISTER DEVICE

[bulk import devices](#)

Device ID
This is the unique identifier for the device in this app. The device ID will be immutable.

Device EUI
The device EUI is the unique identifier for this device on the network. You can change the EUI later.

App Key
The App Key will be used to secure the communication between you device and the network.

App EUI

Cancel

Figure 17: Device form

7. After registering the device, you will see an overview of the device settings. You will see the Device EUI, Application EUI and Application Key.

DEVICE OVERVIEW

Application ID **ideetron_demonstration**

Device ID device1

Activation Method **OTAA**

Device EUI <> 00 B8 5E 2A 7C 79 5A 8A hex

Application EUI <> 70 B3 D5 7E F0 00 45 9E hex

App Key <> 5C 3D 0F 2A 21 3E 12 17 02 39 01 B4 DA 7C 4A FE hex

Status ● never seen

Frames up 0 [reset frame counters](#)

Frames down 0

Figure 18: Device overview

8. Start up realterm and set the parameters.

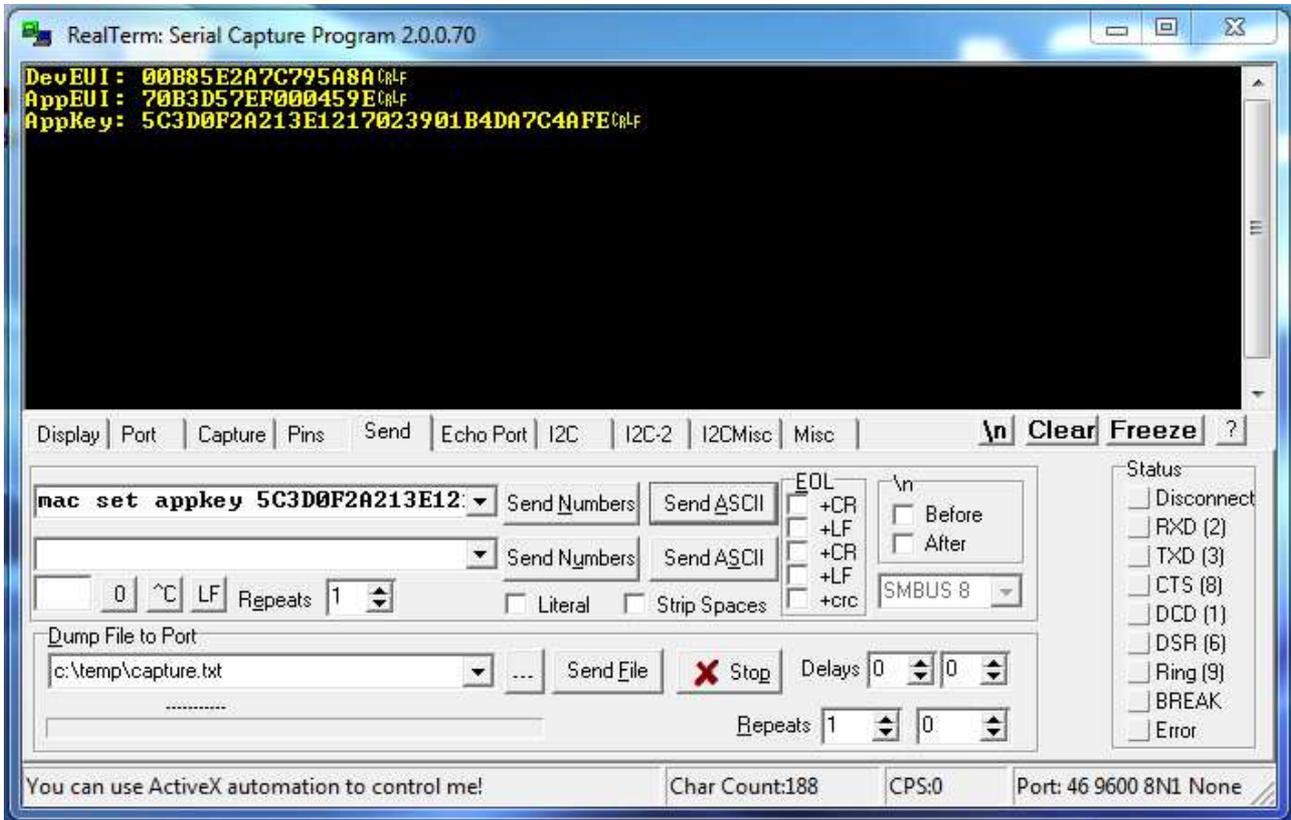


Figure 19: Set parameters

9. Send the join command and wait for reply.

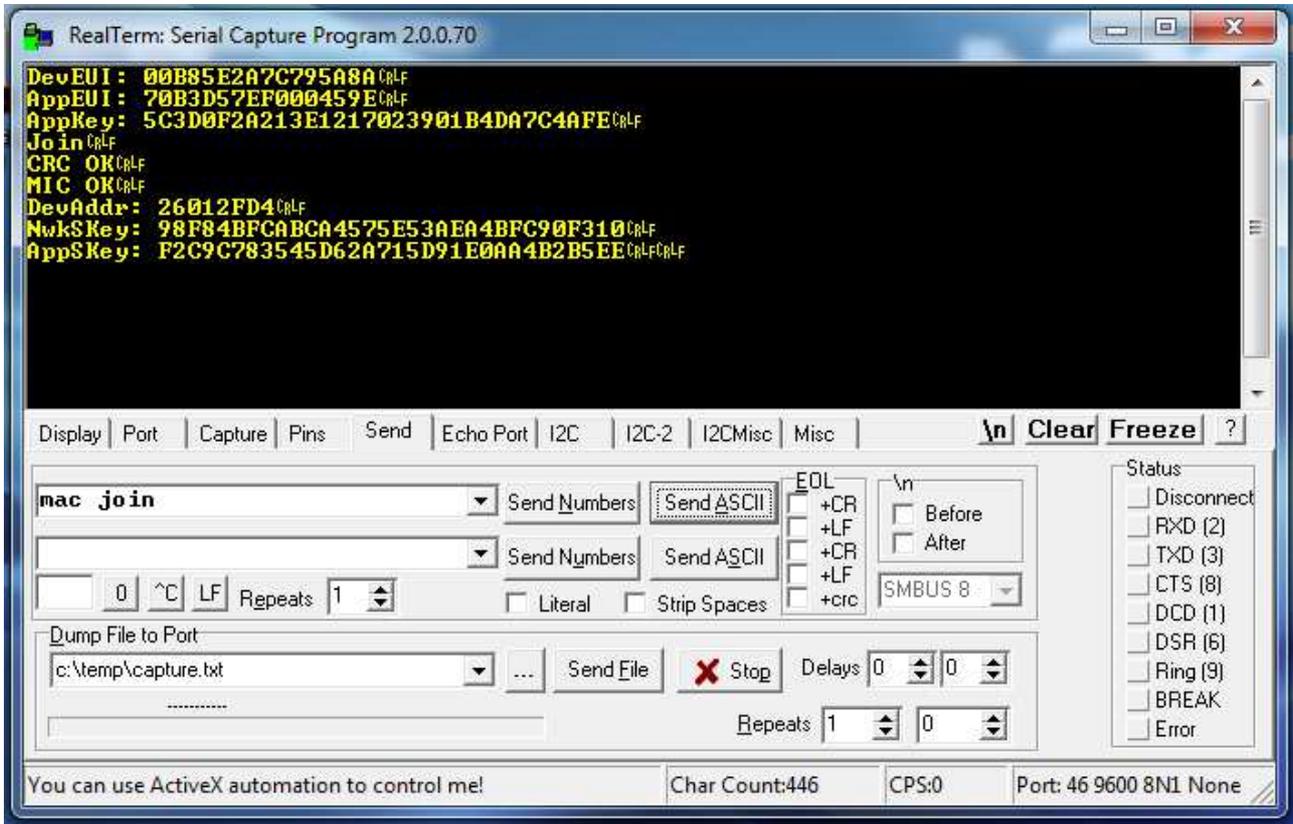


Figure 20: OTAA response