

Getting Started with the micro:bit LoRa Node

Getting Started with the micro:bit LoRa Node

Our IoT micro:bit LoRa Node allows you to create an inexpensive LoRa node, compatible with The Things Network, in conjunction with a BBC micro:bit or other single board computers. This board allows quicker prototyping as it has the LoRa stack on the chip. Add sensors, buttons and more to complete your LoRa network.

Follow the links below to access the various sections more quickly

- [Features](#)
- [What you need](#)
- [Setting up TTN device](#)
 - [Step 1 - Login to TTN](#)
 - [Step 2 - Create an application](#)
 - [Step 3 - Register new device](#)
 - [Step 4 - Setup Microsoft Makecode](#)
- [IoT LoRa Code Blocks](#)
 - [Initialisation Block](#)
 - [Add digital value Block](#)
 - [Add analogue value Block](#)
 - [Add light value Block](#)
 - [Add temperature value Block](#)
- [For all value data Blocks](#)
 - [Transmit LoRa data Block](#)
- [Create a Simple Program](#)

Features

- Uses RAK811 LoRa Radio with full LoRaWAN Stack embedded
- Supports LoRaWAN Connections and LoRaP2P Modes.
- u.FL connector and onboard antenna allowing different antennas to be used or integrated into boxes with external antennas.
- Also adds 8 Extra GPIO Pins controllable via the UART including 2 pins which can read analogue sensors!

- Low power - uses less than 50mA During transmission

What you need

- BBC micro:bit board
- micro:bit LoRa Node
- Micro USB cable
- AA Battery pack (optional)

Setting up a TTN Device

Step 1 - Login to TTN

First go to <https://console.thethingsnetwork.org>

- If not logged in login using your TTN Username and password.
- If you don't have an account, click create an account and follow the steps to setup a TTN Account

- Click on applications
- Then click "**add application**"
- For application ID use "**my-microbit-application-**", in my case it is "**my-microbit-application-chris**" or use something unique to you.
- Description: use something like "**PiSupply IoT nodes**"
- Handler registration use "**ttn-handler-eu**"
- Click "**add Application**"

Step 3 - Register new device

- Now click "**Devices**" in the toolbar
 - Then "**register device**"

- Here create a unique ID, in my case I've done "**chris-microbit**"
- For device EUI click the two arrows to the left to generate one
- Then click "**Register**"

Pi Supply

The Maker Emporium
https://learn.pi-supply.com

In this device, next we want to click "**Settings**"

○ Change the activation method to ABP by clicking the ABP Button

- For development we want to disable frame counter checks, do this by unticking this field at the bottom

- Now TTN Is ready for our microbit! Keep this page open in a tab for later on

Step 4 - Setup Microsoft Makecode

- Go to <https://makecode.microbit.org/>
- Once the editor has loaded, next click the cog icon to the left of the microsoft logo and then extensions

Pi Supply

The Maker Emporium

<https://learn.pi-supply.com>

In here at the top box type in "IoT LoRa Node"

• Press enter to search and click the **iot-lora-node** package

- You should now see a tab for **IoTLoRaNode** in the middle of the editor

IoT LoRa Code Blocks

Our Makecode package has a selection of blocks used to communicate and build payloads to send over LoRa. Data is built in the Cayenne LPP Format allowing quick decoding by The Things Network and easy integration into the Cayenne console.

Initialisation Block

This block is used to configure the LoRa Node with the device address, network session key and application session key from the things network console. It also allows the spreading factor to be changed.

Note: Currently the Makecode package only supports ABP activation.

This block is used to add a boolean / digital value (True or False).

You can transmit a manually set true or false value or you can drag in boolean value blocks to transmit if a button is pressed or the result of logic operations.

[Add Analogue Value Block](#)

This block allows you to transmit a value between 0 and 254, as an analog value it's compatible with analog values produced by the microbit itself.

[Add Light Value](#)

Similar to the add analogue value, this block tells the things network it is a light sensor and will format appropriately. Ideal with use of the built in light level function or external sensors via the I2C Breakout.

[Add Temperature Value](#)

Once again similar to the analogue and light values, this block is designed to transmit temperature values and will automatically format it with the Cayenne LPP to decode in degrees. Ideal for use with the temperature function.

For all value blocks

As you can see there is the option to set which channel the block transmits on. This is so that you can set each value to its own unique channel so after rebooting it'll always stay in the same order. While it will still transmit if all values are on the same channel this may cause some issues. Currently the software supports building a payload of up to 9 channels.

Transmit LoRa data block

This block when ran will then transmit all of the data added to the payload string using the Add value blocks. Once transmitted it will then reset the payload string to blank ready for it to be repopulated.

Creating a Simple program

- To begin we need to setup our microbit lora node with the correct keys to login to The Things Network
- First drag the **Initialise LoRa Radio** block into the **on start** block
- Next we need to fill out the contents of this block from the device data from the Things Network Tab

- Copy over the following into the Microbit program - To copy click the clipboard icon next to the key. The keys will be hidden as standard.
 - **Device address** into the Device Address Field
 - **Network Session Key** into the network session key
 - **App session Key** into the app session key field
 - Leave SF at Seven.
- Your program should now look like the following

Pi Supply

The Maker Emporium

<https://learn.pi-supply.com>

- Now we're going to add the code to transmit a simple boolean value when we press the A button on the micro:bit:

- Click the input category tab and then drag the “On button A Pressed” block into the workspace
- Next from the IoT LoRa Node category drag the “Add Digital Value false on channel: One” block
- And then from the IoT LoRa Node category drag the "Transmit LoRa Data" block underneath the Add Digital Value Block.
- Finally click on the false and change this to **true**. This way when the button is pressed it'll build the payload with a message saying it detected a true value and adds it to channel One of our message.
- Your code should look like the following

- And we're ready to test our program!
 - Click on **Download**
 - If the automatic downloading setup correctly it'll automatically program the microbit and you should see the micro:bit after a few seconds say LoRa ready.
 - If not it'll download the file. Drag this over to your Microbit's virtual USB Drive.
 - Once the board has displayed “LoRa Ready” press the A button to transmit your message!
 - You should then be able to go to your TTN Console, click on the Data tab and see the data which has been transmitted.

time	counter	port	
▲ 10:34:05	0	1	payload: 01 00 01 digital_in_1: 1

And if that worked correctly you have successfully transmitted a basic packet using your microbit!

As you can see the payload automatically decoded to detect that it was a digital input, on channel 1 of a value of 1 (True).

- Next we're going to add a new add Analogue Value block in-between the add digital value and Transmit LoRa Data blocks. To do this simply drag from the IOT LoRa Node tab the Add Analog value block.
 - For this we then need to change the channel to Two, and you can put an analogue value of your choice in.
- Next click download and either let the web browser flash the micro:bit or copy the file over. And then test by pressing the A button.
- You should now see a new payload with a digital in and an analogue value!



A screenshot of a serial monitor window. At the top, there are buttons for back, forward, and refresh. Below that, the text "20:49:31" is displayed. To the right of the timestamp are three numerical fields: "0", "1", and "retry". Further to the right is the text "payload: 01 00 01 02 02 10 68". After a short gap, the text "analog_in_2: 42" appears, followed by "digital_in_1: 1".

- Finally we're going to add in the code to transmit the Microbit's temperature sensor. As our library is designed around the Cayenne Low Power Protocol we can use the Add Temperature Value block instead. While the temperature is an analogue value by using the temperature block it will automatically format it to Degrees C.
 - To do this from the IOT LoRa Node library drag in the Add Temperature Value block.
 - Change it to channel 3
 - Then from the Input library drag the temperature circle into the area where you can put the value.
 - Your code should now look like this

- Once again click download, run the code on the Microbit and press the A Button to transmit. If successful you should see a line like the following on your TTN Console!

```
▲ 20:48:39      0      1      retry      payload: 01 00 01 02 02 10 68 03 67 00 F0      analog_in_2: 42      digital_in_1: 1      temperature_3: 24
```

Summary

Hopefully after following this tutorial you should have been able to setup your micro:bit node and send packets of data to your Gateway using The Things Network. If for whatever reason that your node is not working or you need help with something then please email us at **sales@pisupply.com** or visit our Discord channel - <https://pisupp.ly/chat>