RAK13101 WisBlock GSM/GPRS Module Datasheet

Overview

Description

WisBlock RAK13101 GSM/GPRS Module is a part of the WisBlock Wireless series that provides GSM/GPRS capability on the WisBlock platform by using Quectel MC20CE cellular module.

Features

- Quectel MC20CE module
- Supports GSM/GPRS/GNSS: 850/900/1800/1900 MHz
- Supports BeiDou/GPS/GLONASS/QZSS
- Built-in LNA
- IPEX connectors for the GSM and GNSS antenna
- Micro-USB debug and log output connector
- Nano SIM and ESIM options
- 3.3 V Power supply
- Small size: 25 mm x 45 mm

Specifications

Overview

The RAK13101 module can be mounted on the IO slot of the WisBlock Base board. Figure 1 shows the mounting mechanism of the RAK13101 on a WisBlock Base module, such as a RAK5005-O.

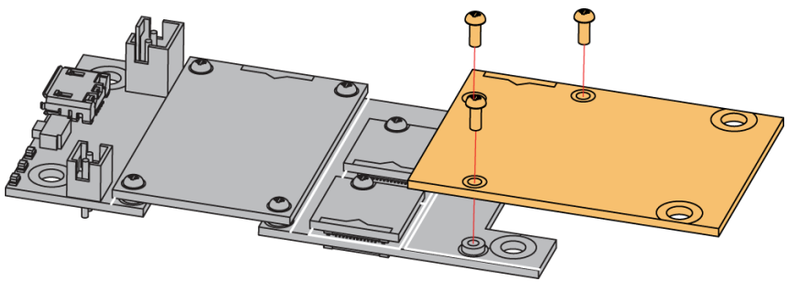


Figure 1: RAK13101 mounting mechanism on a WisBlock Base module

Hardware

The hardware specification is categorized into five parts. It shows the chipset of the module and discusses the pinouts and its corresponding functions and diagrams. It also covers the electrical and mechanical parameters that include the tabular data of the functionalities and standard ratings of the RAK13101 WisBlock GSM/GPRS Module.

Chipset

Vendor	Part number
Quectel	MC20CE

Pin Definition

The RAK13101 WisBlock GSM/GPRS module comprises a standard WisBlock IO slot connector. The WisBlock connector allows the RAK13101 module to be mounted on a WisBlock baseboard with IO slot, such as RAK5005-O. The pin order of the connector and the pinout definition is shown in **Figure 2**.

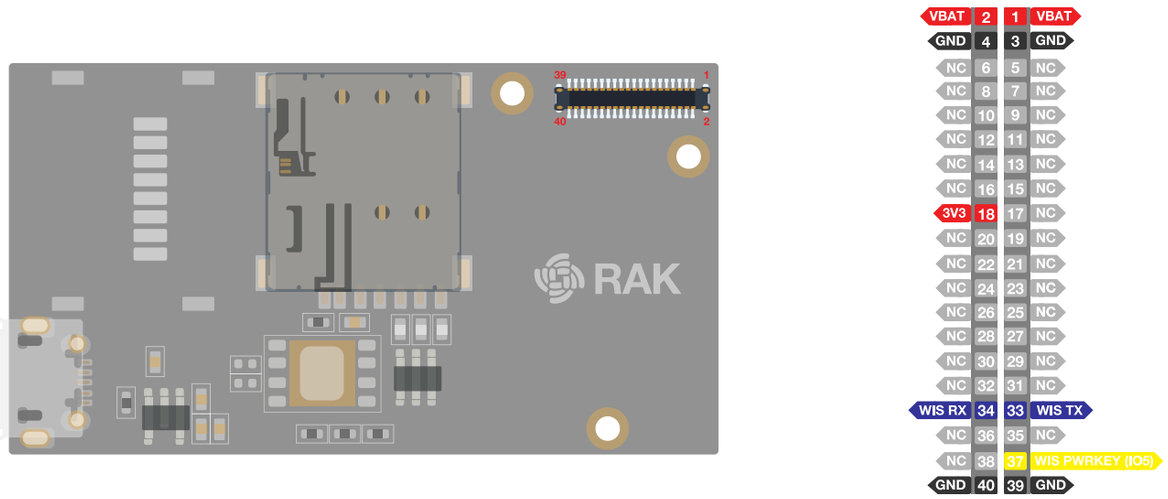


Figure 2: RAK13101 Connector Pin Definition

> 📝 **NOTE**
>
> - RAK13101 WisBlock IO slot connector utilizes the **UART** related pins, **PWRKEY** via WB_IO5, **VBAT**, **3V3**, and **GND**.
> - **VBAT** is the battery voltage input with max voltage 4.2 V. During GPRS data and GSM dial mode, the peak current is 1170 mA and 769 mA, respectively, which exceeds the USB port supply current. That is why you must use a dedicated battery.

Electrical Characteristics

Absolute Maximum Ratings

Parameter	Minimum	Maximum	Unit
VBAT	-0.3	+4.2	V
GNSS_VCC	-0.3	+4.2	V
Power supply peak current0	0	2	A

Power Supply Ratings

Symbol	Description	Condition	Min.	Nom.	Max.	Unit
VBAT	Supply Voltage	Input voltage must within this range	3.3	4.0	4.2	V
GNSS_VCC	GNSS Part Supply voltage	Input voltage must within this range	2.8	3.3	4.2	V
Ipeak	Peak Current @ VBAT = 3.8 V	Idle	-	-	110	mA

Symbol	Description	Condition	Min.	Nom.	Max.	Unit
	Peak Current @ VBAT = 3.8 V	GSM Dial	-	-	769	mA
	Peak Current @ VBAT = 3.8 V	GPRS Data	-	-	1170	mA
Ivbat	Average Supply Current	Power down mode	-	20	-	mA
	Average Supply Current	Sleep mode @DRX = 5	-	1.2	-	mA

Mechanical Characteristics

Board Dimensions

Figure 3 shows the dimensions and the mechanical drawing of the RAK13101 module.

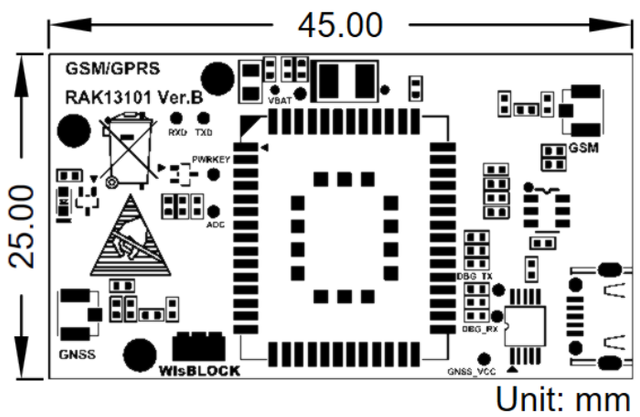
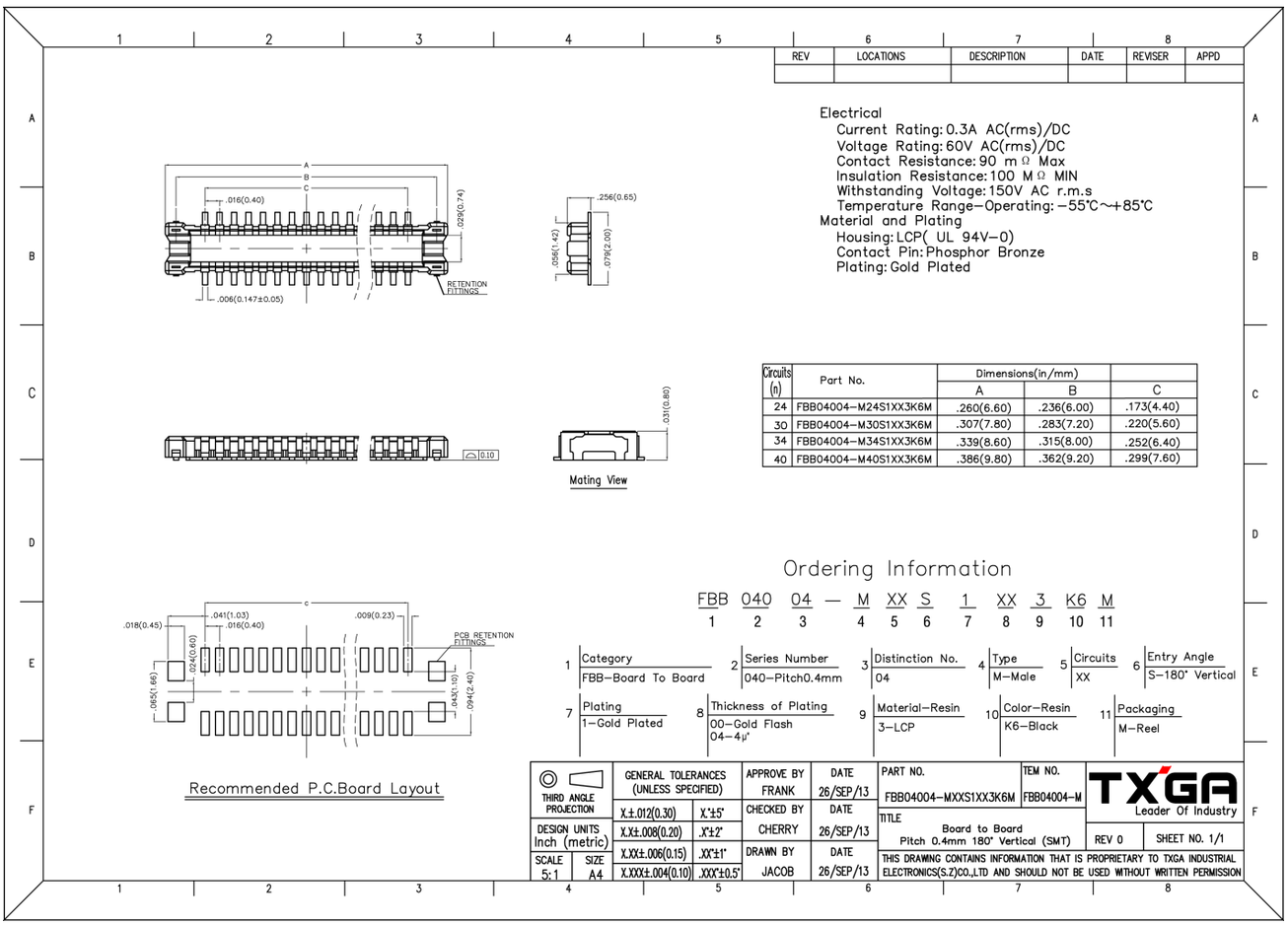


Figure 3: RAK13101 Mechanical Drawing

WisConnector PCB Layout



Circuits (n)	Part No.	Dimensions(in/mm) A	B	C
24	FBB04004-M24S1XX3K6M	.260(6.60)	.236(6.00)	.173(4.40)
30	FBB04004-M30S1XX3K6M	.307(7.80)	.283(7.20)	.220(5.60)
34	FBB04004-M34S1XX3K6M	.339(8.60)	.315(8.00)	.252(6.40)
40	FBB04004-M40S1XX3K6M	.386(9.80)	.362(9.20)	.299(7.60)

Figure 4: WisConnector PCB Footprint and Recommendations

Schematic Diagram

Power Supply

The Quectel MC20CE module's main power supply comes from **VBAT**, which is a battery voltage connected to the WisBlock Base board. **GNSS_VCC** is the GNSS section supply voltage and is turned on or off via **GNSS_VCC_EN**, which is connected to MC20CE. The power supply of the GNSS part is controlled via the AT command `AT+QGNSSC`.

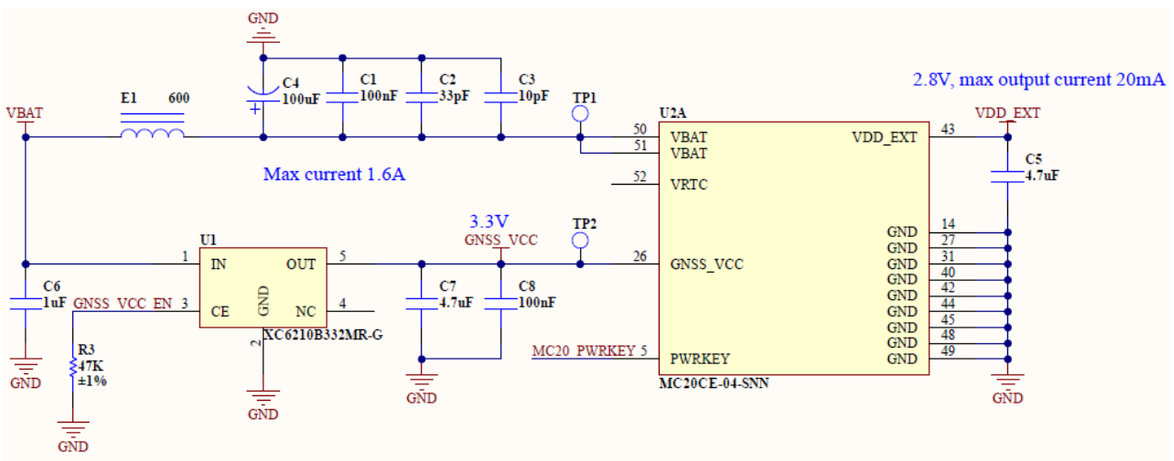


Figure 5: RAK13101 Power Supply

GSM and GNSS

J3 is GSM antenna connector and **J2** is GNSS antenna connector.

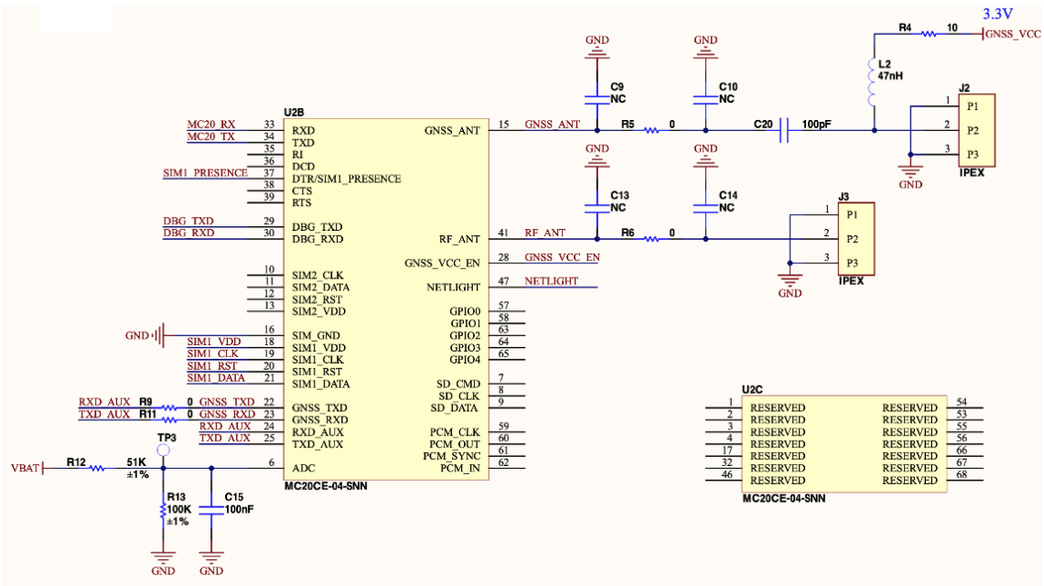


Figure 6: RAK13101 GSM and GNSS Antenna Circuit

Voltage Level Transfer

The GSM/GPRS module operates at 2.8 V, while the WisBlock Core is at 3.3 V. That is why a voltage level shifter is needed.

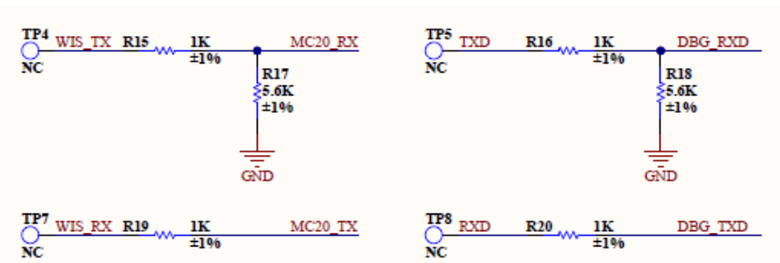


Figure 7: RAK13101 Voltage Level Shifter Circuit

SIM Circuit

Figure 8 shows the schematic of RAK13101 module as a slot for standard SIM card and an optional **ESIM** pcb footprint.

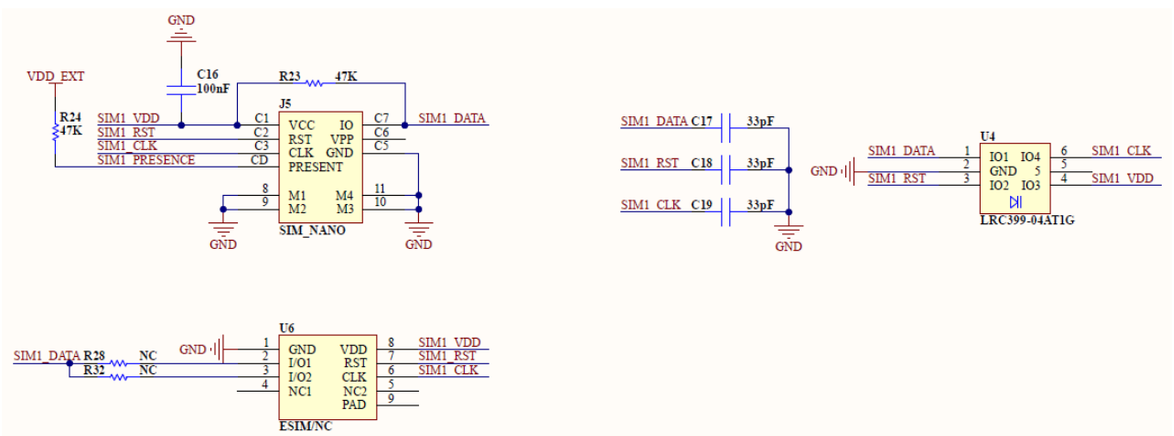


Figure 8: RAK13101 GSM/GPRS Module SIM Circuit

WisConnector

RAK13101 WisBlock IO slot connector utilizes **UART** related pins, **PWRKEY** via WB_IO5, **VBAT**, **3V3**, and **GND**.

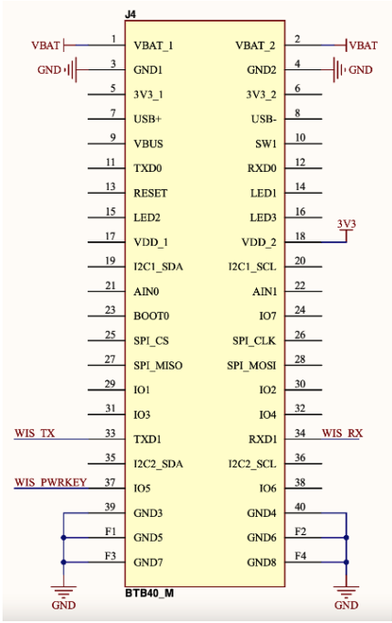


Figure 9: RAK13101 IO Slot Connector

Debug Connector

Figure 10 shows the RAK13101 USB debugging circuit.

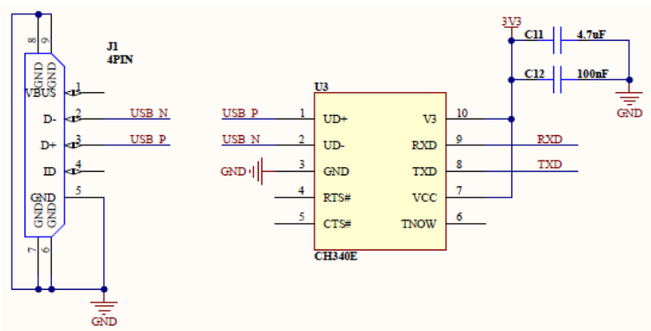


Figure 10: RAK13101 USB Debugging

LED/PWRKER Control Circuit

Figure 11 shows the RAK13101 module LED and PWRKEY controlled circuit.

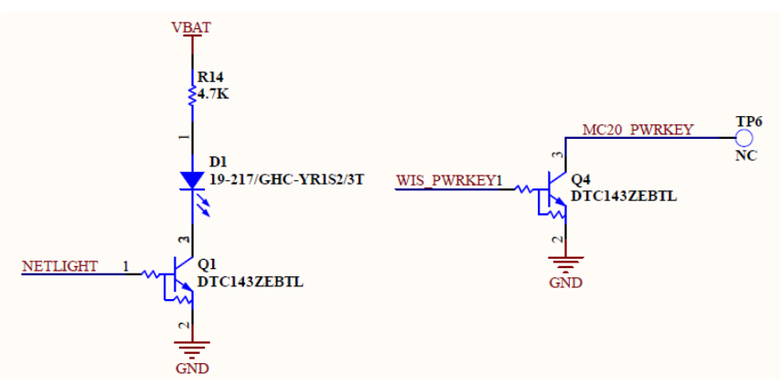


Figure 11: RAK13101 Module LED PWRKEY Control Circuit

Last Updated: 1/10/2022, 2:07:11 AM

RAK13101 Quick Start Guide

Prerequisite

What Do You Need?

Before going through each and every step on using the RAK13101 WisBlock module, make sure to prepare the necessary items listed below:

Hardware

- RAK13101
- Your choice of WisBlock Base
- Your choice of WisBlock Core
- USB Cable
- GNSS and GSM Antennas
- Li-Ion/LiPo battery
- Solar charger (optional)

Software

- Download and install Arduino IDE .
- To add the RAKwireless Core boards on your Arduino board, install the RAKwireless Arduino BSP. Follow the steps in the GitHub repo .

Product Configuration

Hardware Setup

You can integrate the RAK13101 module on your WisBlock project to extend its functionality and have GSM/GPRS/GNSS capability. This is ideal for tracking applications with GSM/GPSR cellular connectivity in the area. For more information about RAK13101, refer to its Datasheet.

RAK13101 module can be mounted to the IO slot of the WisBlock Base and communicates to the WisBlock Core via UART. The module is activated via `WB_IO5` pin of the WisBlock Core. Two antennas must be connected to the module as well, one for the GNSS antenna port and one for the GSM antenna port. An external battery (Li-Ion/LiPo 3.7-4.2V) is also required to power up the module properly.

> ⚠️ **WARNING**
>
> - Batteries can cause harm if not handled properly.
> - Only 3.7-4.2 V Rechargeable LiPo batteries are supported. It is highly recommended not to use other types of batteries with the system unless you know what you are doing.
> - If a non-rechargeable battery is used, it has to be unplugged first before connecting the USB cable to the USB port of the board to configure the device. Not doing so might damage the battery or cause a fire.
> - Only 5 V solar panels are supported. Do not use 12 V solar panels. It will destroy the charging unit and eventually other electronic parts.
> - Make sure the battery wires match the polarity on the WisBlock Base board. Not all batteries have the same wiring.

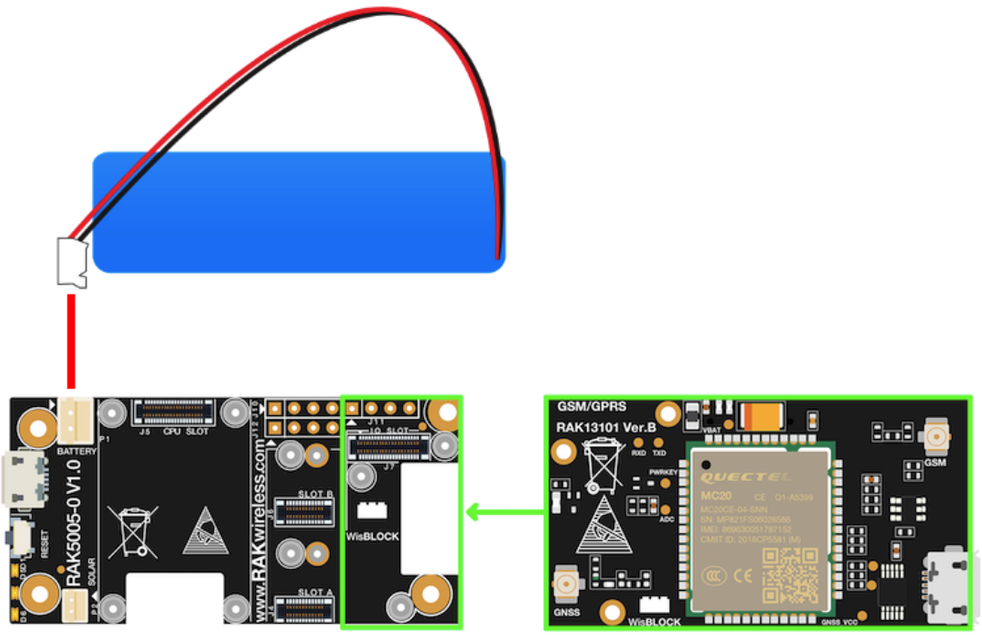


Figure 1: RAK13101 connection to WisBlock Base

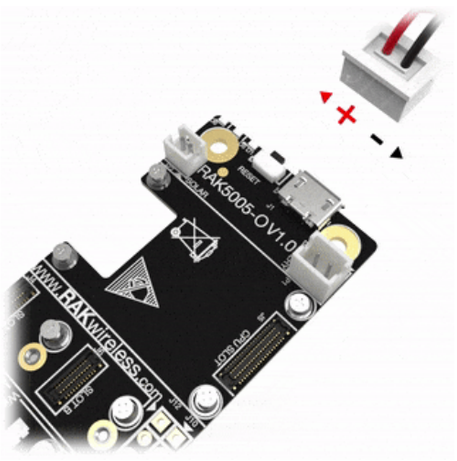


Figure 2: WisBlock Base RAK5005-O battery polarity and connection

Assembling and Disassembling of WisBlock Modules

Assembling

As shown in **Figure 3**, the location for Slot A, B, C, and D are properly marked by silkscreen. Follow carefully the procedure defined in RAK5005-O module assembly/disassembly instructions to attach a WisBlock module. Once attached, carefully fix the module with one or more pieces of M1.2 x 3 mm screws depending on the module.

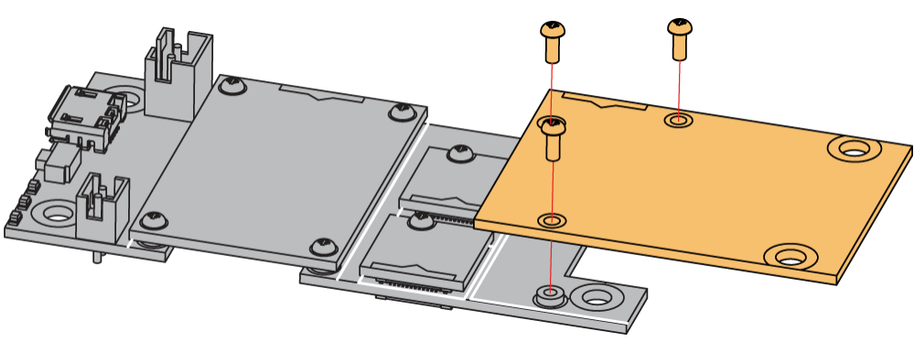


Figure 3: RAK13101 connection to WisBlock Base

Disassembling

The procedure in disassembling any type of WisBlock modules is the same.

1. First, remove the screws.

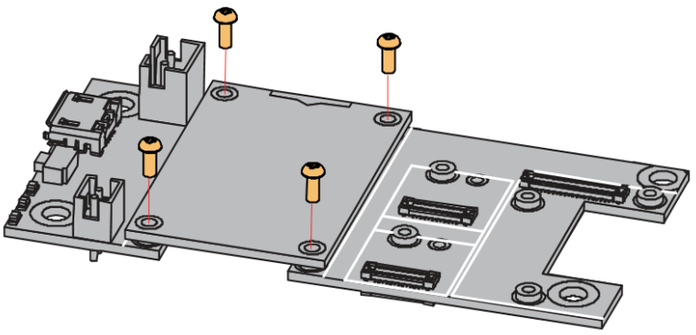


Figure 4: Removing screws from the WisBlock module

2. Once the screws are removed, check the silkscreen of the module to find the correct location where force can be applied.

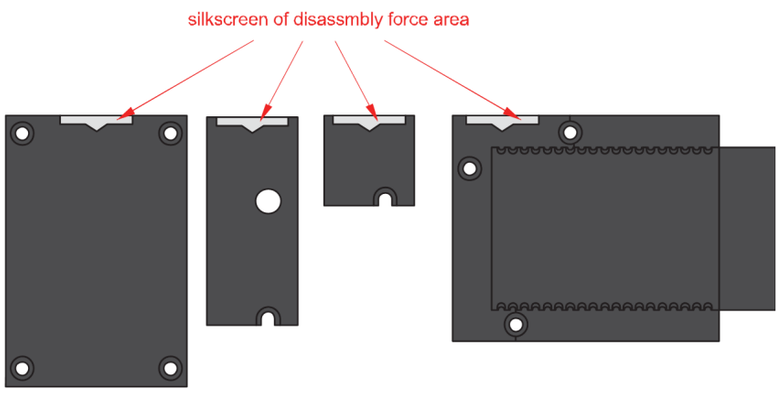


Figure 5: Detaching silkscreen on the WisBlock module

3. Apply force to the module at the position of the connector, as shown in **Figure 6**, to detach the module from the baseboard.

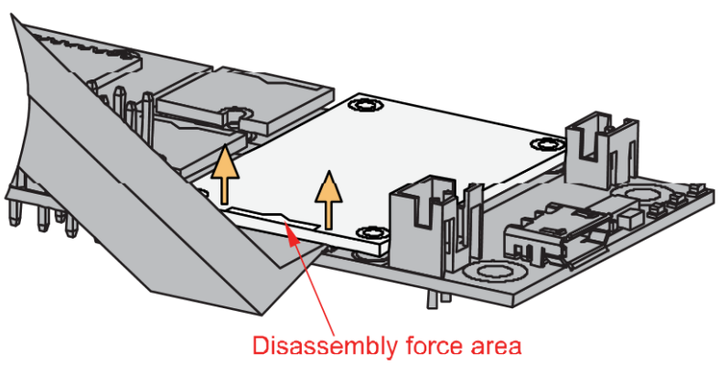


Figure 6: Applying even forces on the proper location of a WisBlock module

> 📝 **NOTE**
>
> If you will connect other modules to the remaining WisBlock Base slots, check on the WisBlock Pin Mapper tool for possible conflicts. RAK13101 uses UART communication lines, and it can cause possible conflict especially on some modules that also use UART.

Software Configuration and Example

The RAK13101 WisBlock GSM/GPRS Module uses UART serial communication lines. In this example code, you will be able to send AT commands to the RAK13101 module. This will ensure that your RAK13101 is functional and ready for your IoT project.

Initial Test of the RAK13101 WisBlock Module

If you already installed the RAKwireless Arduino BSP , the WisBlock Core and example code should now be available on the Arduino IDE.

1. You need to select first the WisBlock Core you have, as shown in **Figure 7** to **Figure 9**.

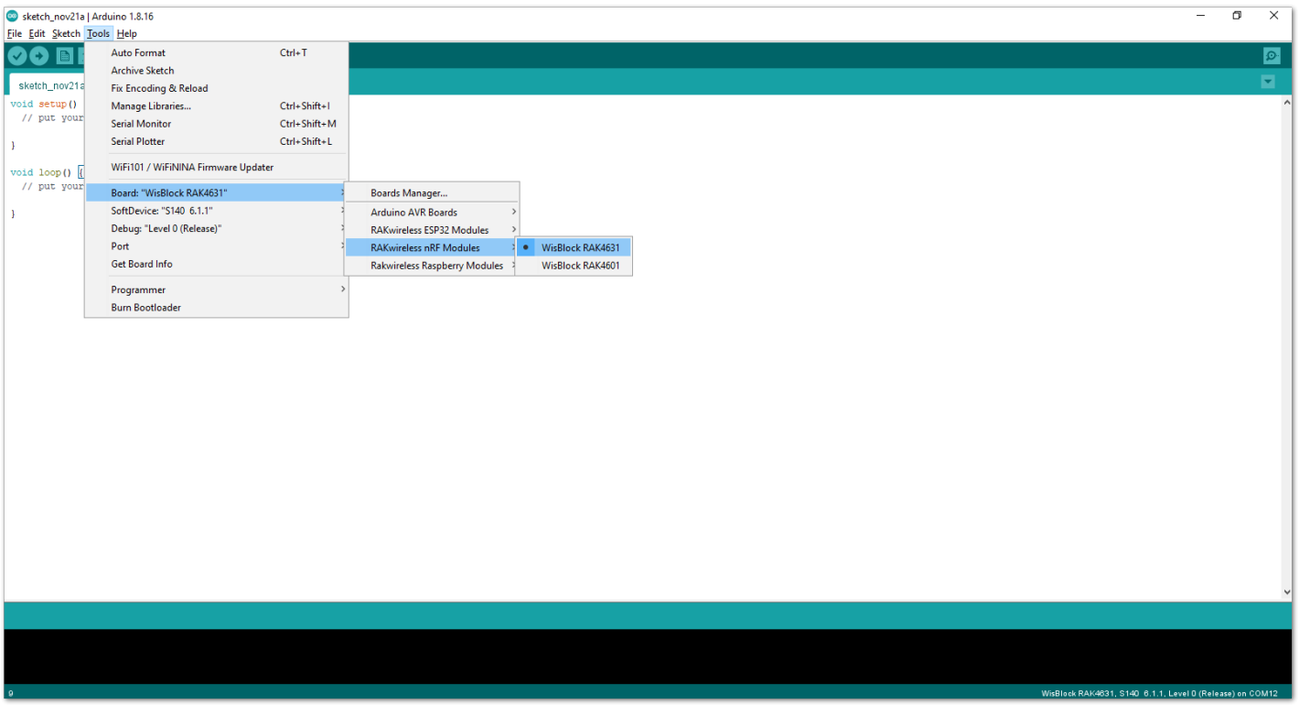


Figure 7: Selecting RAK4631 as WisBlock Core

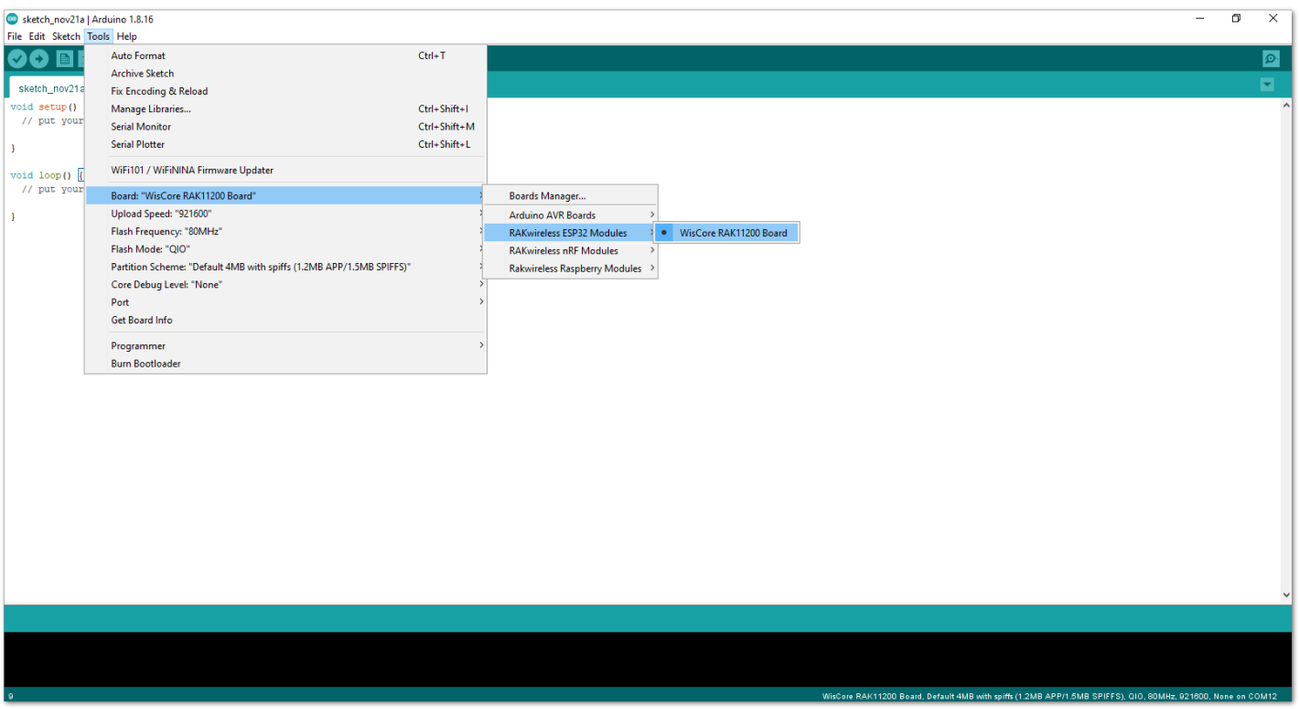


Figure 8: Selecting RAK11200 as WisBlock Core

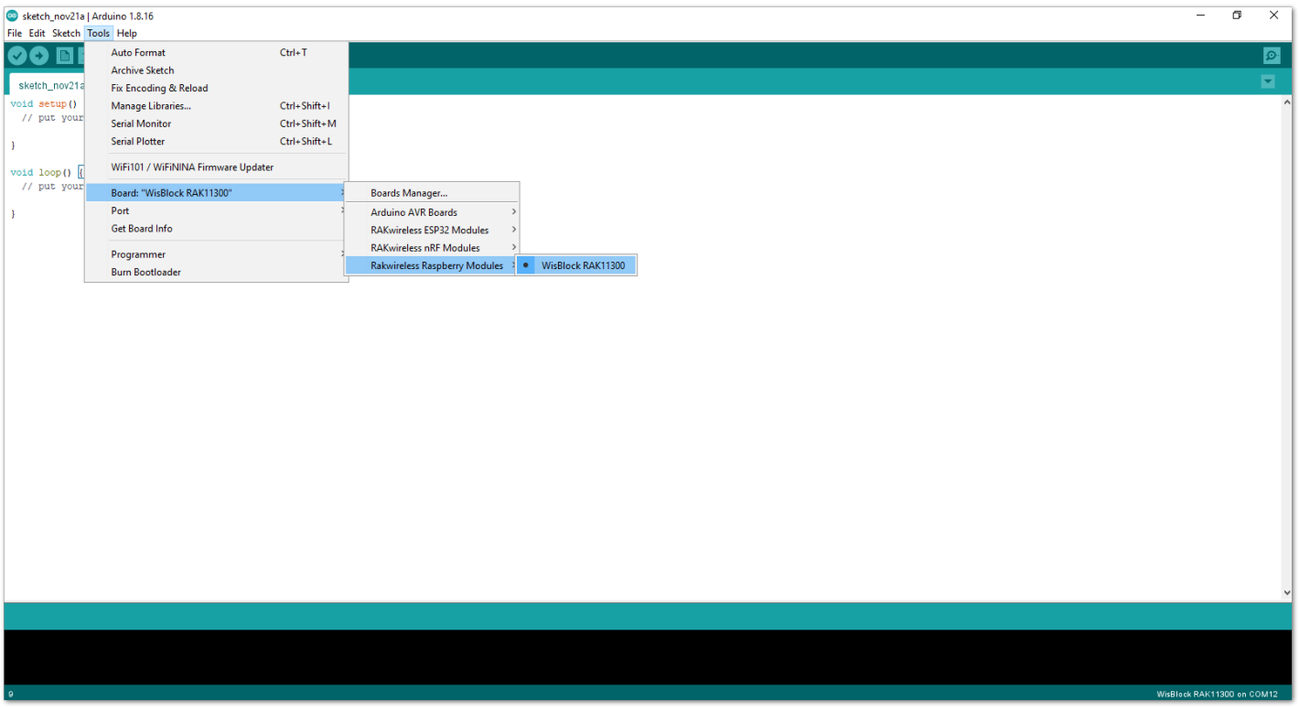


Figure 9: Selecting RAK11300 as WisBlock Core

2. Next, copy the following sample code into your Arduino IDE:

```c
/**
   @file Unvarnished_Transmission.ino
   @author rakwireless.com
   @brief unvarnished transmission via USB
   @version 0.1
   @date 2021-6-28
   @copyright Copyright (c) 2020
**/

#include <Wire.h>

#define POWER_KEY WB_IO5

void setup()
{
    time_t serial_timeout = millis();
    Serial.begin(115200);

    while (!Serial)
    {
        if ((millis() - serial_timeout) < 5000)
        {
            delay(100);
        }
        else
        {
            break;
        }
    }

    Serial.println("AT CMD TEST!");
    // Check if the modem is already awake
    time_t timeout = millis();
    bool moduleSleeps = true;
    Serial1.begin(9600);
    delay(1000);
    Serial1.println("ATI");
    //MC20 init
    while ((millis() - timeout) < 6000)
    {
        if (Serial1.available())
        {
            String result = Serial1.readString();
            Serial.println("Modem response after start:");
            Serial.println(result);
            moduleSleeps = false;
        }
    }
    if (moduleSleeps)
    {
        // Module slept, wake it up
    pinMode(POWER_KEY, OUTPUT);
        digitalWrite(POWER_KEY, 0);
        delay(200);
        digitalWrite(POWER_KEY, 1);
        delay(2000);
        digitalWrite(POWER_KEY, 0);
        delay(1000);
    }
    Serial.println("MC20 power up!");
}
```

```
void loop()
{
    int timeout = 100;
    String resp = "";
    String snd = "";
    char cArr[128] = {0};
    while (timeout--)
    {
        if (Serial1.available() > 0)
        {
            resp += char(Serial1.read());
        }
        if (Serial.available() > 0)
        {
            snd += char(Serial.read());
        }
        delay(1);
    }
    if (resp.length() > 0)
    {
        Serial.print(resp);
    }
    if (snd.length() > 0)
    {
        memset(cArr, 0, 128);
        snd.toCharArray(cArr, snd.length() + 1);
        Serial1.write(cArr);
        delay(10);
    }
    resp = "";
    snd = "";
}
```

3. You can now select the right serial port and upload the code, as shown in **Figure 10** and **Figure 11**.

> 📝 **NOTE:**
>
> If you are using RAK11200 as WisBlock Core, you need to configure the BOOT0 pin before uploading. You need to short it to the ground then press the reset button of the WisBlock Base before releasing the BOOT0 pin. If not done properly, uploading the source code to RAK11200 will fail. Check the full details on the RAK11200 Quick Start Guide.

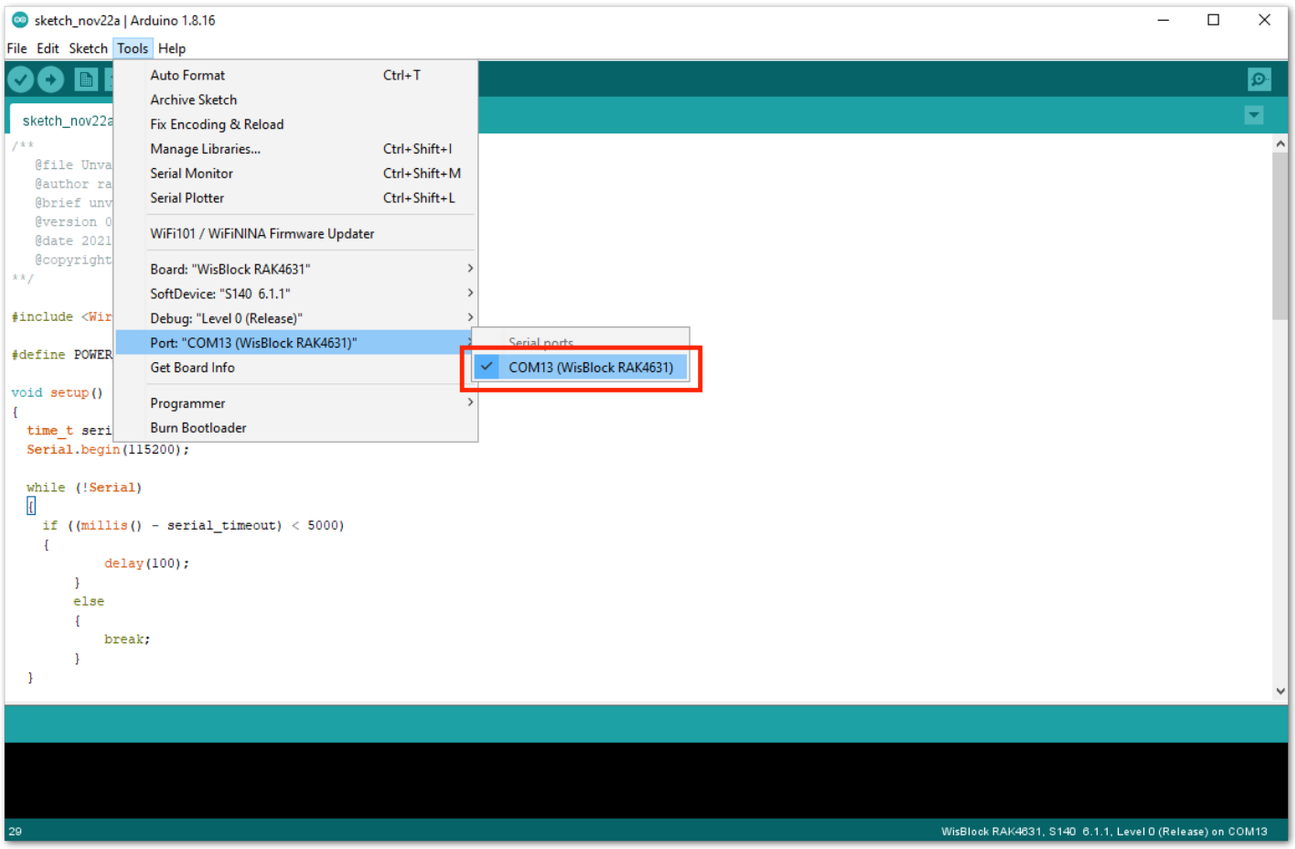


Figure 10: Selecting the correct Serial Port

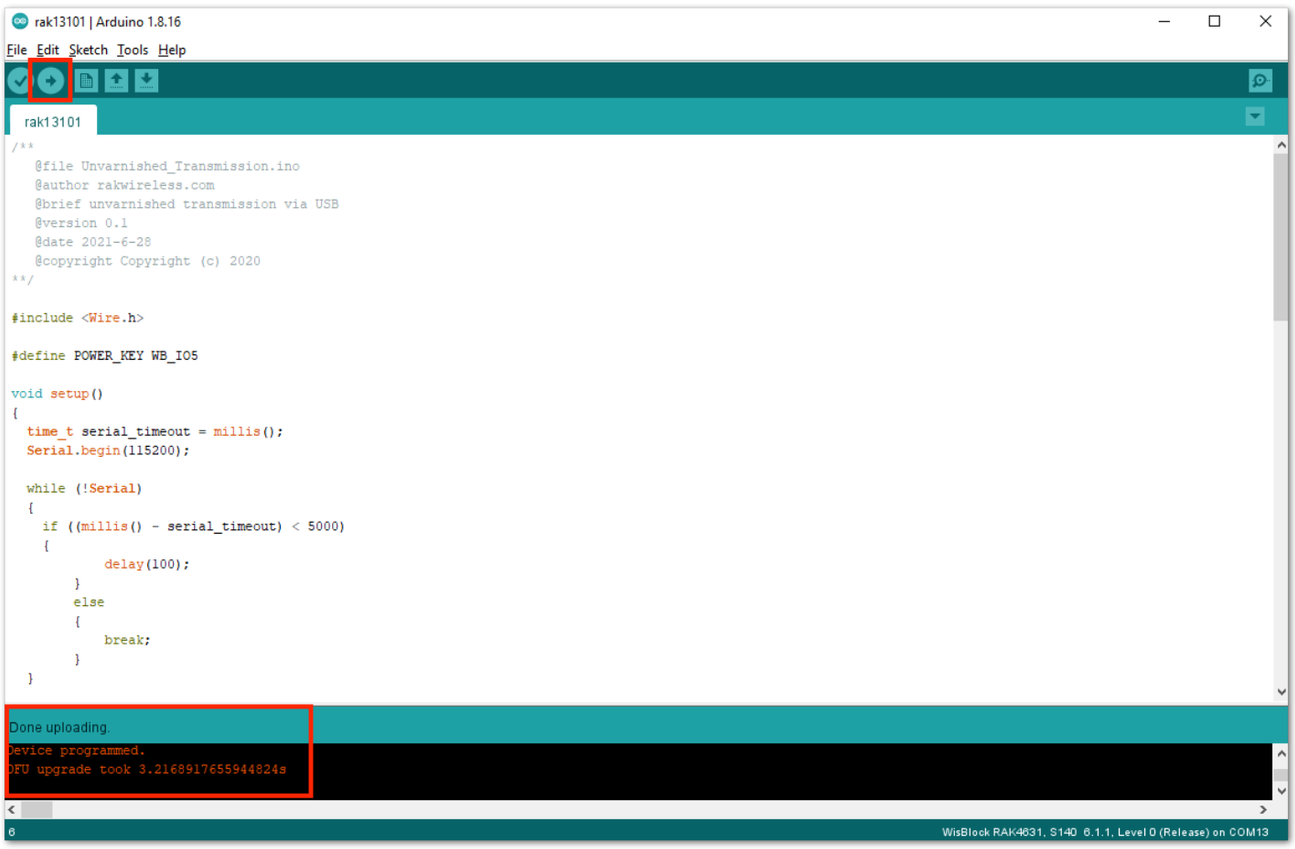


Figure 11: Uploading the RAK13101 example code

> 📝 **NOTE**
>
> If you experience any error in compiling the example sketch, check the updated code for the RAK13101 Module that can be found on the WisBlock Example Code Repository .

4. When you successfully uploaded the example sketch, open the Serial Monitor of the Arduino IDE to see the initial logs, as shown in **Figure 12**. If you do not see any logs, you can try to disconnect the USB cable and

battery, then reconnect again with the battery first. If you see that the LED of the RAK13101 is blinking after a few seconds, the module is now initialized properly.

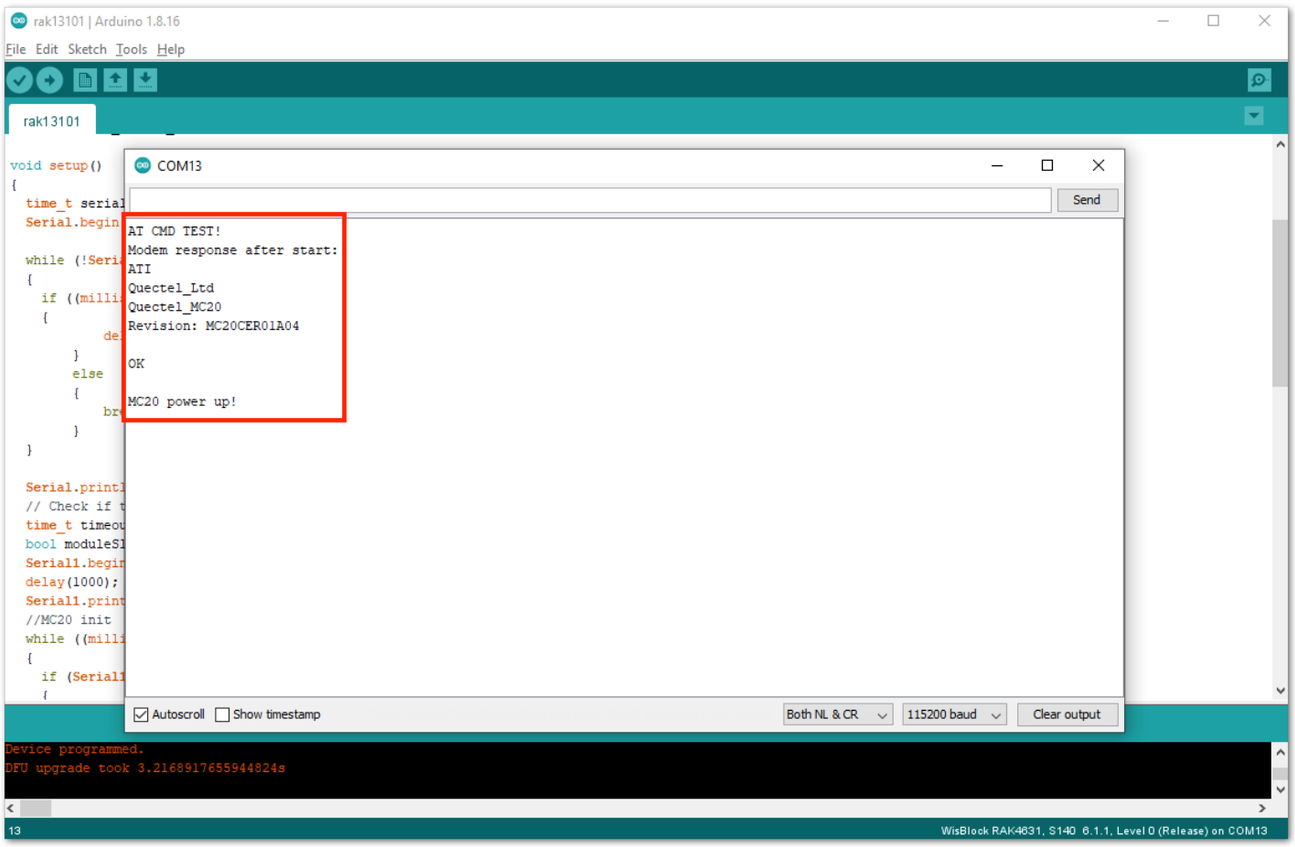


Figure 12: RAK13101 initial logs

5. There are times that you might not see the initial logs if you open the Serial Monitor. The best way to test if the module is working is by sending actual AT commands, as shown in **Figure 13**. You can try to send the basic commands, `AT` and `ATI`.

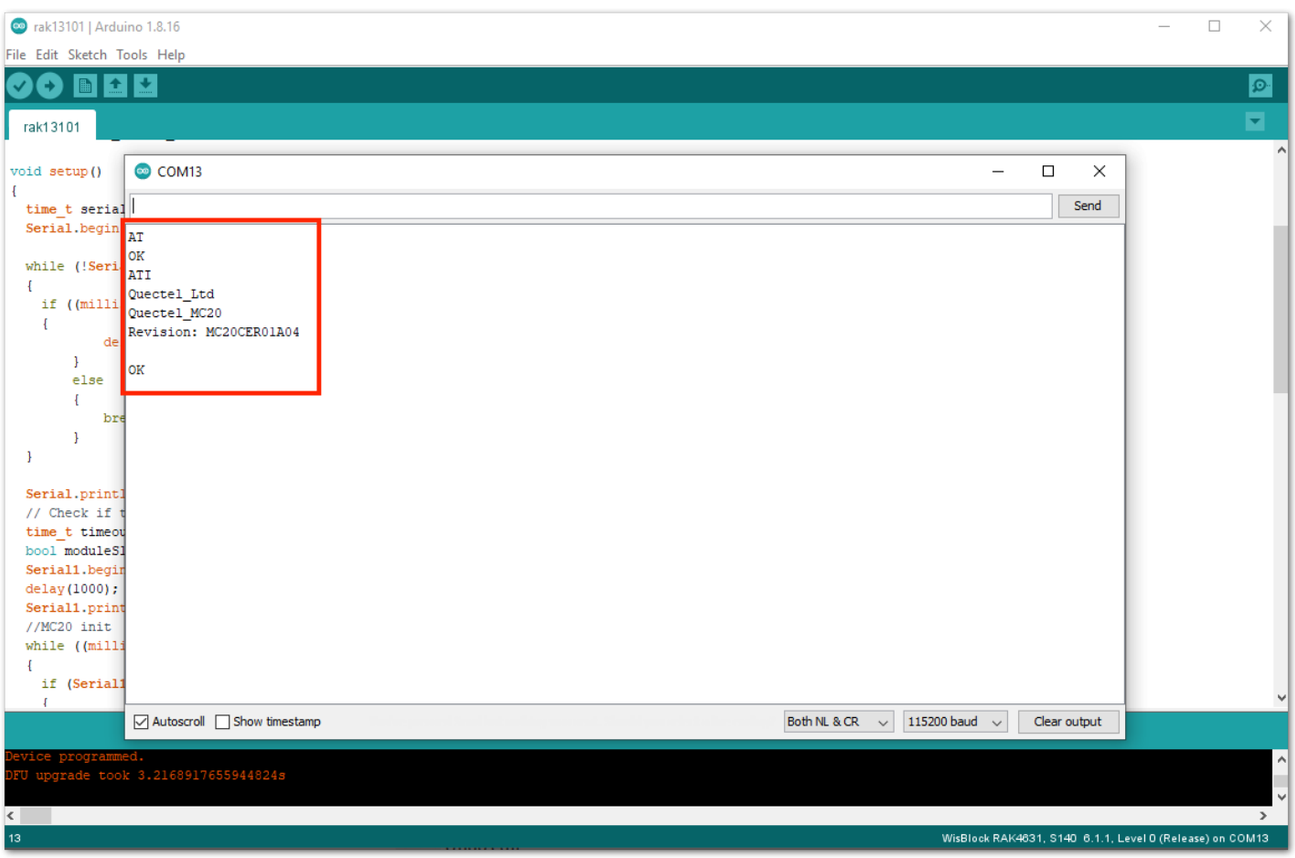


Figure 13: RAK13101 AT and ATI command