RAK5801 WisBlock 4-20mA Interface Module Datasheet

Overview

Description

The RAK5801 WisBlock Interface module was designed to be part of a production-ready IoT solution in a modular way and must be combined with a WisBlock Core and a Base module.

The RAK5801 is a 4-20 mA current loop extension module that allows the users to make an IoT solution for analog sensors with a 4-20 mA interface. This module converts the 4-20 mA current signal into voltage range supported by the WisBlock Core module (MCU) for further digitalization and data transmission.

The RAK5801 module features two input channels of 4-20 mA. Inside, a high-precision operational amplifier is used for signal amplification and conversion and supports a wide range of operating temperatures.

This module integrates a 12 V power supply, which can be used to power external sensors. The RAK5801 can be connected to 2-wire, 3-wire, or 4-wire types of 4-20 mA sensor. The module external interface is reached by a fast crimping terminal that allows connection for the 4-20 mA sensors (including power) and to the I2C bus. The fast crimping terminals can be used without the need of special tools, which simplifies the installation process on the field.

Features

- Two 4-20 mA analog inputs
- Compatible with multiple WisBlock Core modules, such as RAK4631
- 0.005 mA conversion accuracy
- Supports low power consumption mode. The module can be powered off by the WisBlock Core module for saving energy during idle periods.
- 12 V output to power external sensors
- Reserved I2C expansion interface
- Fast crimping terminals
- Designed with a 2 kV ESD protection level
- Small dimensions of 35 mm x 25 mm

Specifications

Overview

The overview discusses the block diagram of the board. It also shows the installation mechanism on how to mount the board into the baseboard.

Block Diagram

The RAK5801 module was designed to convert 4-20 mA current signals into voltage signals by applying a sampling resistor. As shown in **Figure 1**, the input current signal from the sensor is conditioned by an operational amplifier to match the level supported by the ADC input of an MCU where the signal is digitized.

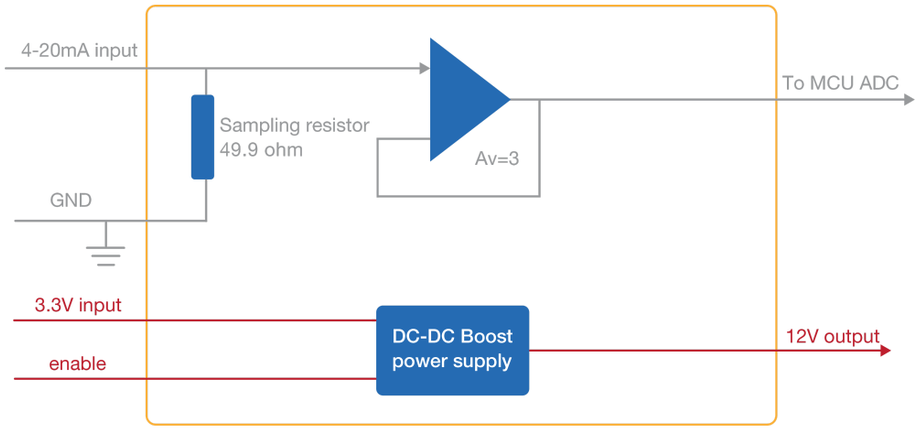


Figure 1: RAK5801 Block Diagram

Once the signal is digitalized, you can recover the original current value by applying the following formula:

```
I = U/149
```

Where **U** is the ADC reading and **I** the sensor current.

As shown in **Figure 1**, the module provides an output of 12 V for powering passive 4-20 mA sensors. This 12 V output is boosted by an internal DC-DC booster. The enable pin allows to control the power conversion module and set the RAK5801 module into a low power consumption mode.

Hardware

The hardware specification is categorized into four parts. It discusses the pinouts of the board and its functionalities and diagrams.

Device Specification

The following table shows the parameters and the description of the RAK5801 WisBlock 4-20mA Interface Module:

Parameter	Description
Analog Input Interface	2 channels of 4-20 mA
Analog Sampling Resolution	0.005 mA
Analog Sampling Accuracy	1%
Analog Maximum Input Current	25 mA (There is a risk to burn the circuit surpassing this limit.)
Analog Port ESD Protection Level	2 kV HBM
Current Sampling Resistor	49.9 Ω
Operational Amplifier Gain	3.0
Input Voltage	3.0-3.6 V
Output Voltage	12 V
Output Current	Maximum 30 mA
Operating Temperature	-30 °C ~ 65 °C
Storage Temperature	-40 °C ~ 85 °C
Module Dimensions	35x25 mm

Pin Definition

This section covers the pin number of the sensor connector, the definition, and the functionalities of each pin shown in a tabular representation.

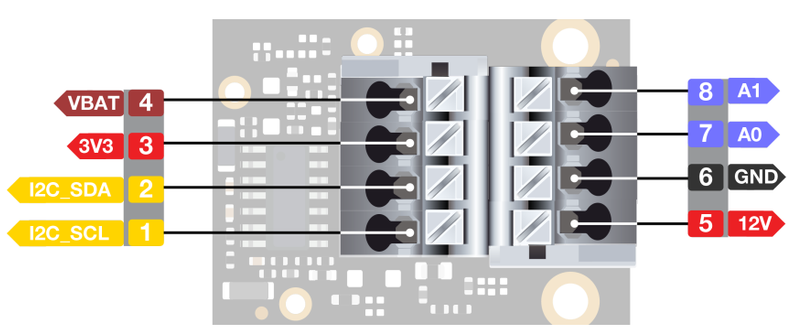


Figure 2: RAK5801 Sensor Connector

> 📝 **NOTE:**
>
> **A0** cannot be used as an analog input channel by default because it is used for measuring battery voltage. But if you need to use **A0**, there are few hardware modifications needed to configure. Please see instructions below.
>
> To enable **A0** as an additional channel:
>
> 1. Remove **R7** on the WisBlock Base such as the RAK5005-O to disconnect Vbat sensing.
> 2. On RAK5801, remove the 0 Ohms resistor in **R94** and put it to **R95**. Please see **Figure 3**.

A0 Hardware Modifications

Figure 3: A0 Hardware Modifications

Pin Number	Function Description
1	SCL of the I2C interface
2	SDA of the I2C interface
3	3V3 output
4	VBAT, Battery output
5	12 V output for external sensors
6	GND
7	Analog input 0
8	Analog input 1

Figure 4 shows the pin order for the IO connector of the module. Through this connector, the RAK5801 module is attached to the WisBoard baseboard.

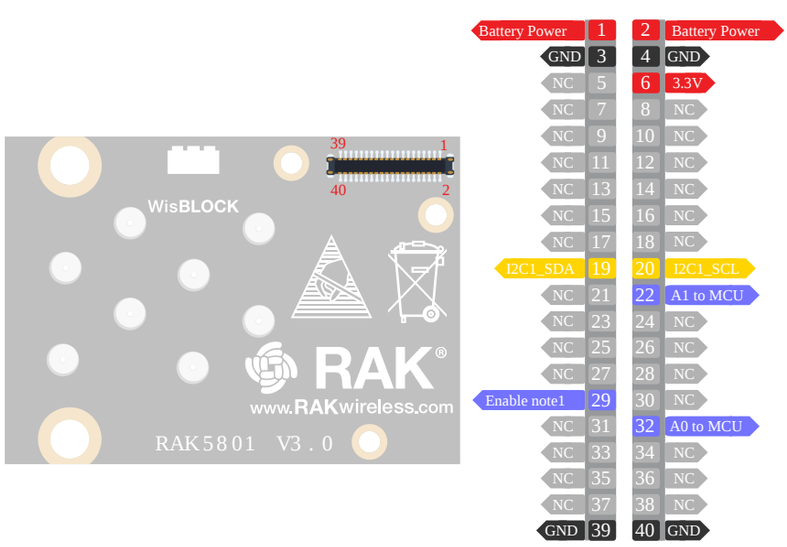


Figure 4: RAK5801 Internal WisIO Connector

The functionalities of each pins of the WisIO connector are tabulated below.

Pin Number	Description	Pin Number	Description
1	Battery Power	2	Battery Power
3	GND	4	GND
5	NC, reserved for 3V3	6	3.3V Power
7	NC	8	NC
9	NC	10	NC
11	NC	12	NC
13	NC	14	NC
15	NC	16	NC
17	NC	18	NC
19	SDA for I2C1	20	SCL for I2C1
21	NC	22	Analog to MCU
23	NC	24	NC
25	NC	26	NC
27	NC	28	NC
29	Enable note1	30	NC
31	NC	32	Analog0 to MCU
33	NC	34	NC
35	NC	36	NC
37	NC	38	NC
39	GND	40	GND

📝 **NOTE:**

This signal controls the dc-dc power supply on RAK5801, before capturing analog signal, set this pin to high to enable power for RAK5801.

Mechanical Characteristics

Board Dimensions

Refer to **Figure 5** below for the mechanical dimensions of the RAK5801 module.

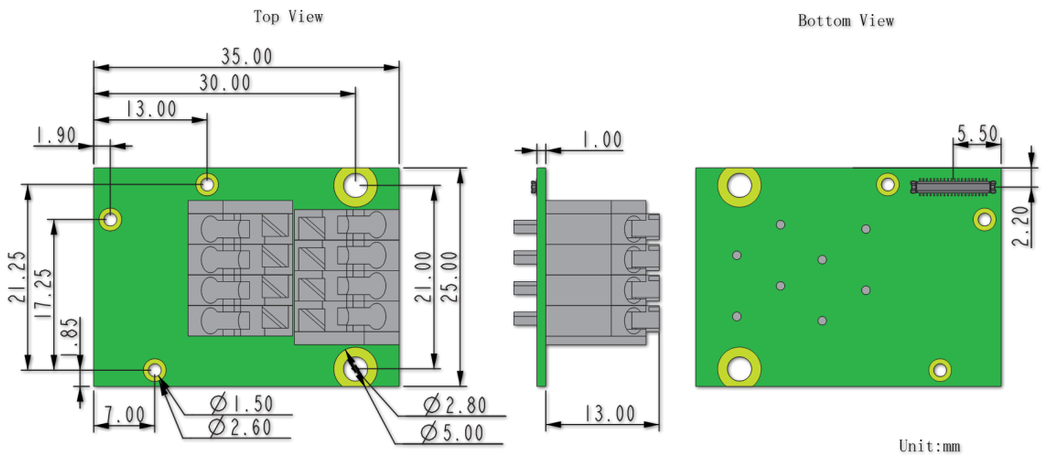
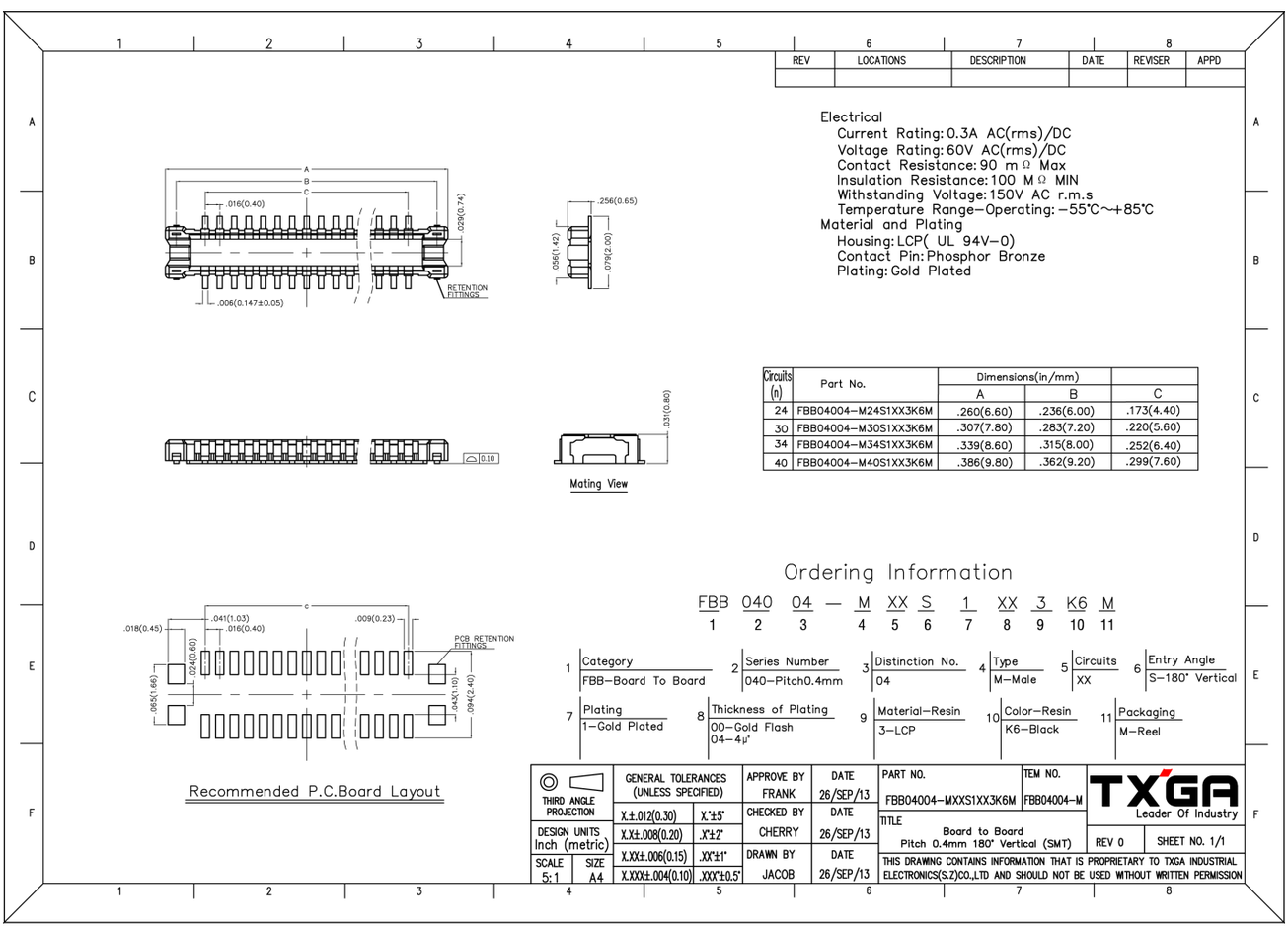


Figure 5: RAK5801 Mechanical Dimensions

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 6: WisConnector PCB footprint and recommendations

Schematic Diagram

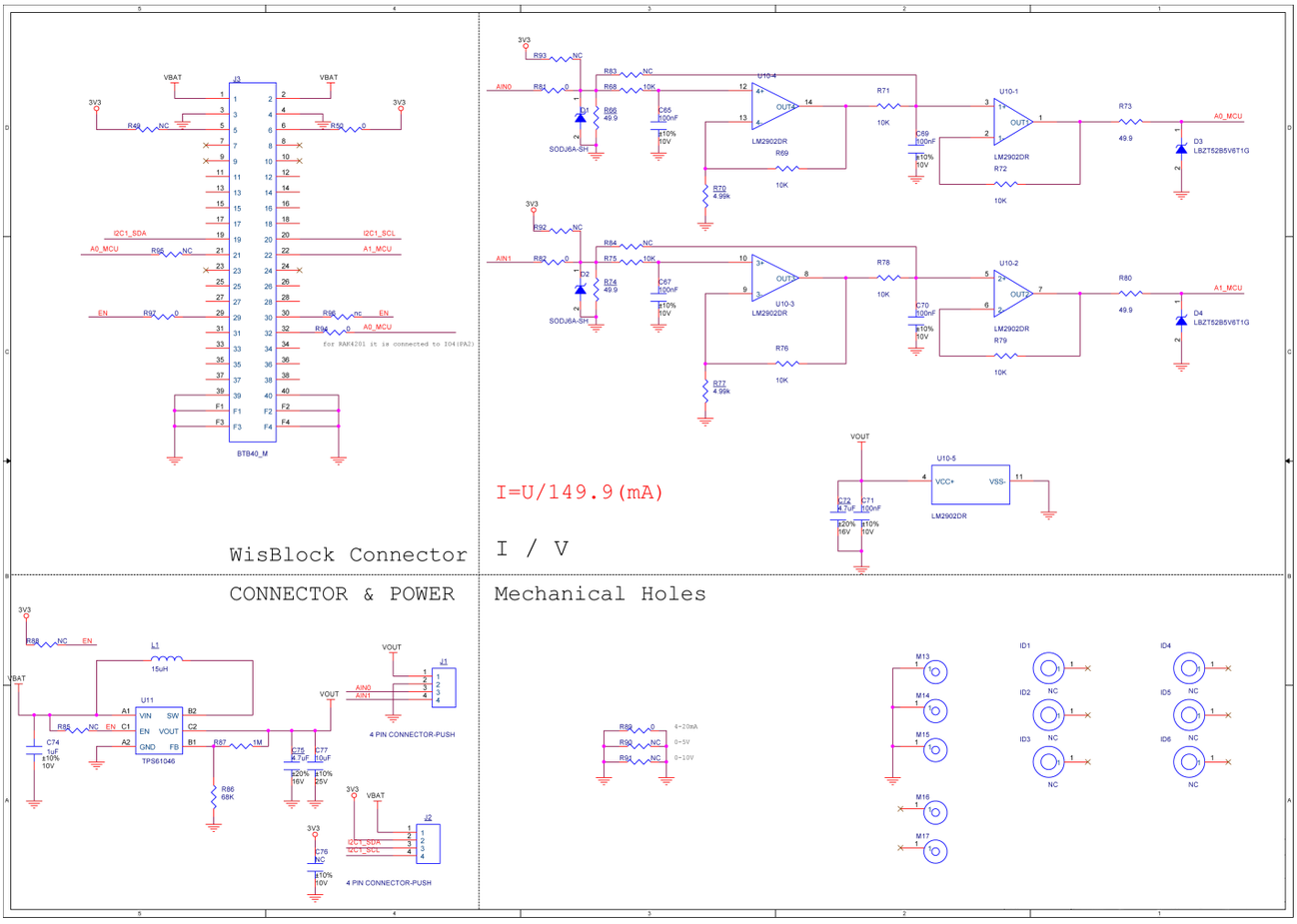


Figure 7: RAK5801 Schematic Diagram

WisBlock Compatibility

Since a WisBlock module can be combined with a variety of different functional modules, the pin functions of the MCU are multiplexed, so the interface expansion module for each specific function may need to be properly adapted for the WisBlock. The compatibility details of the RAK5801 module are as shown in the Table below:

WisBlock Module	Adaptable Module	Description
WisBase Base board	RAK5005/RAK5005-O	RAK5801 is designed to be assembled in the IO slot of RAK5005-O baseboard.
WisBlock Core Module	RAK4631	RAK5801 is compatible with RAK4631.
	RAK4201	Please select RAK4201L-ADC for the low band or RAK4201H-ADC for the high band.
	RAK4202	Please refer to Note 2 for hardware adaptions to the RAK5005-O and RAK5801.
	RAK4261	Please refer to Note 3 for hardware adaptions to the RAK5005-O and RAK5801.

📝 **NOTE**

1. The RAK5801+RAK4601

The RAK5801 is not compatible with RAK4601. The main reason is because RAK4601 doesn't expose any ADC pin through the RAK5005-O baseboard.

📝 **NOTE**

2. RAK5801+RAK4202+RAK5005-O

In order to combine a RAK5801 module, a RAK4202 (WisBlock Core module), and the RAK5005-O, the following modification must be introduced:

- In RAK5005-O, remove the R7 resistor as shown in **Figure 8.**
- In RAK5801, remove R94 to R95 resistors, and use PA0 of STM32L151 to read the analog data of the channel “analog0”, and use PA2 of STM32L151 to read the analog data of Channel analog1. **Figure 9** shows the resistors R94 and R95 on the RAK5801 module.

This combination has the following restriction:

- The adapted RAK5005-O module will not able to sense the battery voltage anymore.

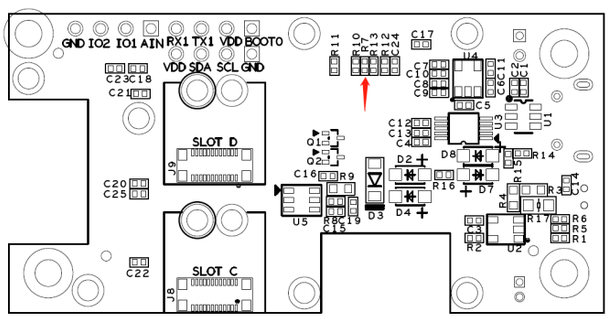


Figure 8: R7 on RAK5005-O

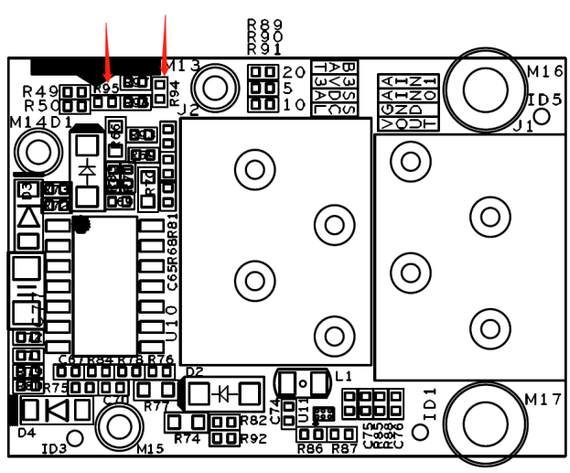


Figure 9: R94 and R95 on RAK5801.

📝 **NOTE**

3. RAK5801+RAK4261+RAK5005-O

In order to combine a RAK5801 module, a RAK4261(WisBlock Core module), and the RAK5005-O, the following modification must be introduced:

- In RAK5005-O, remove the R7 resistor. See **Figure 8**.
- In RAK5801, remove R94 to R95 resistors (see **Figure 9**), and use PA08 of ATSAMR34 to read the analog data of the channel “analog0”, and use PA09 of ATSAMR34 to read the analog data of Channel analog1.

This combination has the following restriction:

- The adapted RAK5005-O module will not able to sense the battery voltage anymore.

RAK5801 Quick Start Guide

Prerequisite

What Do You Need?

Before going through each and every step on using the RAK5801 WisBlock module, make sure to prepare the necessary items listed below:

Hardware

- RAK5801 WisBlock 4-20mA Interface Board
- WisBlock Base
- Your choice of WisBlock Core
- USB Cable
- Li-Ion/LiPo battery (optional)
- Solar charger (optional)

Software

Arduino

- You need to download and install Arduino IDE .
- To add the RAKwireless Core boards on your Arduino Boards Manager, install the RAKwireless Arduino BSP .

Product Configuration

Block Diagram

The RAK5801 module was designed to convert 4-20 mA current signals into voltage signals by applying a sampling resistor. As shown in **Figure 1**, the input current signal from the sensor is conditioned by an operational amplifier to match the level supported by the ADC input of an MCU where the signal is digitized.

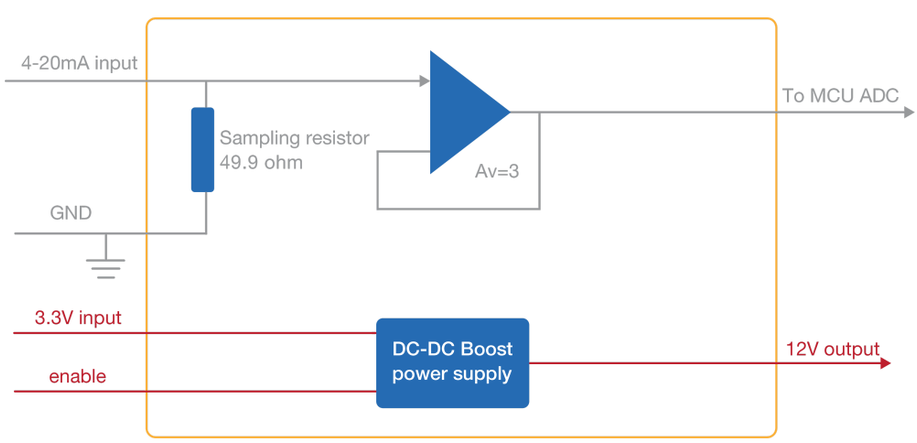


Figure 1: RAK5801 Block Diagram

Once the signal is digitalized, the user can recover the original current value by applying the following formula:

```
I = U/149
```

Where **U** is the ADC reading and **I** is the sensor current.

As shown in **Figure 1**, the module provides an output of 12 V for powering passive 4-20 mA sensors. This 12 V output is boosted by an internal DC-DC booster. The enable pin allows to control the power conversion module and set the RAK5801 module into a low-power consumption mode.

Hardware Setup

The RAK5801 is a 4-20 mA current loop extension module that allows you to make an IoT solution for analog sensors with 4-20 mA interface. This module converts the 4-20 mA current signal into voltage range supported by the WisBlock Core module for further digitalization and data transmission. This module integrates a 12 V power supply, which can be used to power external sensors. The RAK5801 can be connected to 2-wire, 3-wire, or 4-wire types of 4-20 mA sensor. For more information about RAK5801, refer to the Datasheet.

Installation

Mounting Mechanism

The RAK5801 module is part of the WisBlock Interface category, which connects to the baseboard through the IO slot. The installation method is shown in **Figure 2** and **Figure 3**.

1. Keep the RAK5801 module parallel to the baseboard, and gently place and plug WisConnector into the IO slot receptacle of the baseboard. The IO slot has an outer silkscreen on it to assist with the alignment. At this point, apply force evenly along with the module and press again. There will be a sound to confirm the successful completion of the attachment process.

> 📝 **NOTE:**
>
> For detailed instructions, refer to the WisBlock Installation Guide.

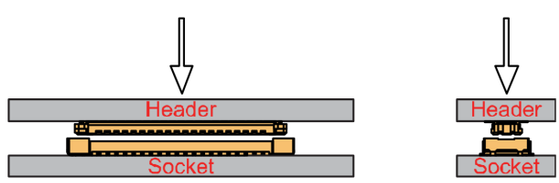


Figure 2: WisConnector

2. Always secure the RAK5801 module with **3 x M1.2 x3 pan head screws**, as shown in **Figure 3**.

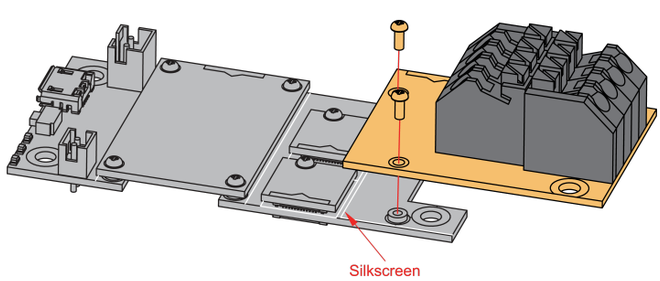


Figure 3: RAK5801 mounting mechanism on a WisBlock Base module

RAK5801 Fast Crimping Terminal Mechanism

The RAK5801 features a fast-crimping terminal connector to simplify and ensure the wiring process on the fields. The fast-crimping terminal can support cable with a width between 20 AWG to 24 AWG. The usual stripping length is around 6 to 7 mm.

As shown in **Figure 4**, during the crimping process, you should first press down and maintain the spring head of the crimping terminal firmly, then insert the stripped cable head into the corresponding connector's hole. Once

inserted correctly, then release the spring head, and the crimping process is completed.

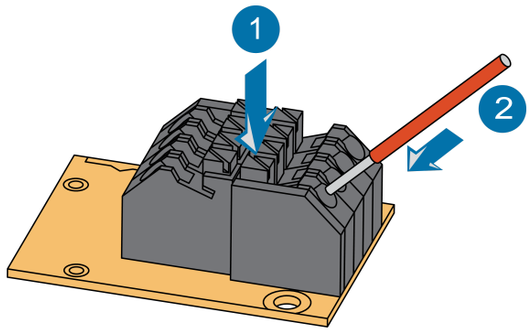


Figure 4: RAK5801 Sensor Connector

Disassembling Procedure

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

1. First, remove the screws.

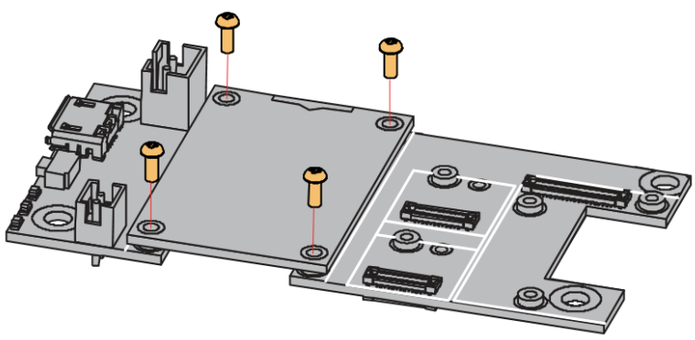


Figure 5: Removing screws from the WisBlock module

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

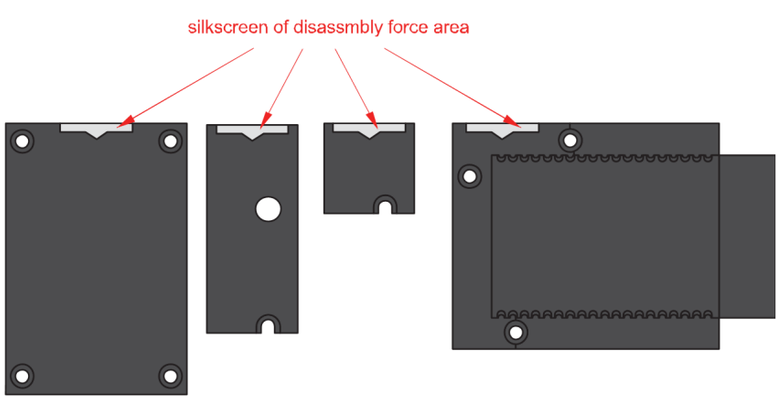


Figure 6: Detaching silkscreen on the WisBlock module

3. Apply force to the module at the position of the connector, as shown in **Figure 7**, to detach the module from the baseboard.

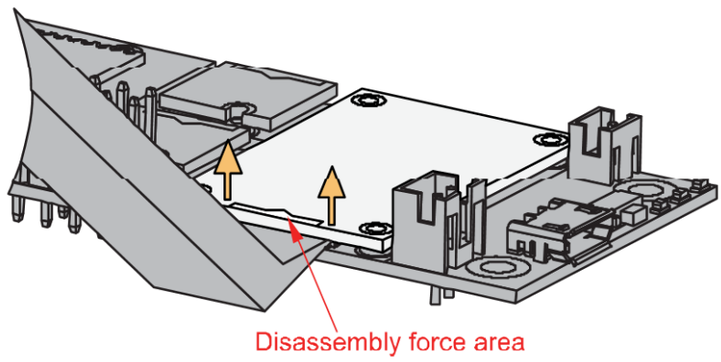


Figure 7: Applying even forces on the proper location of a WisBlock module

> 📝 **NOTE**
>
> If you will connect other modules to the remaining WisBlock Base slots, check on the WisBlock Pin Mapper tool for possible conflicts.

Pin Definition

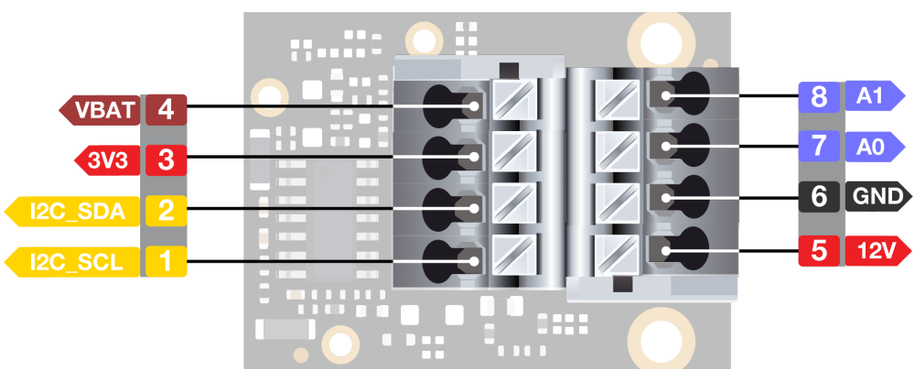


Figure 8: RAK5801 Pin Definition

Typical Application

Two-Wire sensor

Two-wire transmitters are energized by the current loop, where the supply voltage is included in the receptor. The transmitter is floating and the ground is in the receptor.

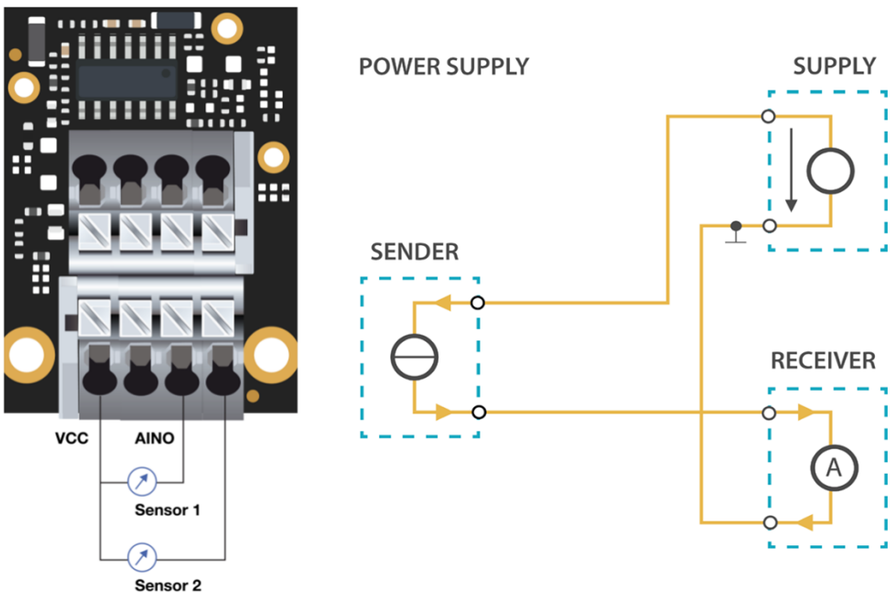


Figure 9: RAK5801 with 2-wire/4-20mA sensor.

Three-Wire Sensor

Three-wire transmitters have three wires powered by the source voltage in them. In this case, the transmitter is the power source for the current loop. The transmitter common is connected to the common of the receptor.

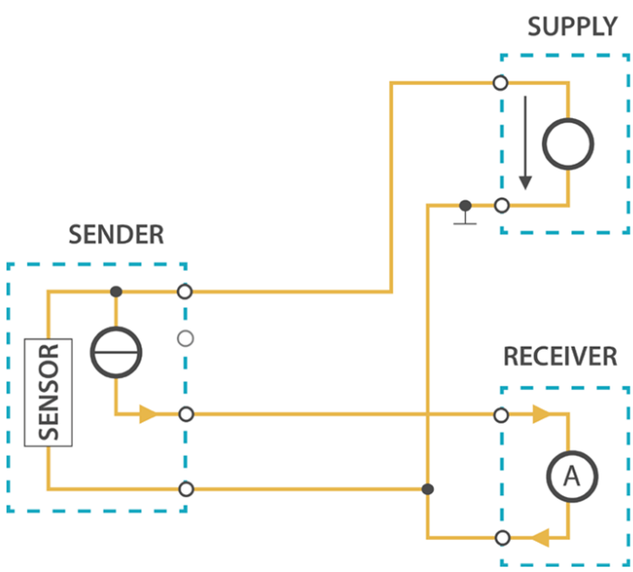


Figure 10: RAK5801 with 3-wire/4-20mA sensor.

Four-Wire Sensor

Four-wire transmitters have four wires powered by the source voltage in them. The transmitter powers the current loop and the receptor acts as a floating load.

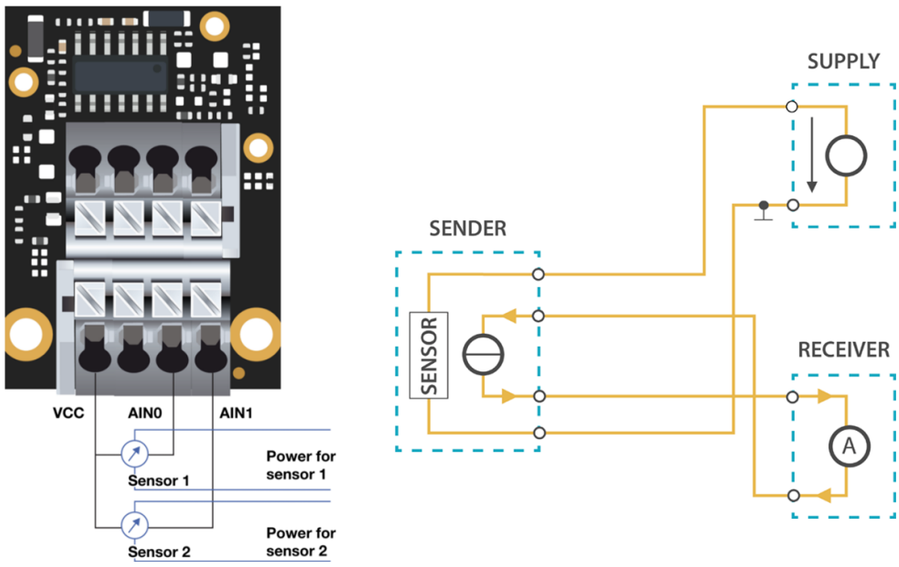


Figure 11: RAK5801 with 4-wire/4-20mA sensor.

Now, you can connect the battery (optional) and USB cable to start programming your WisBlock Core.

> ⚠️ **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 are matching the polarity on the WisBlock Base board. Not all batteries have the same wiring.

Software Configuration and Example

The RAK5801 module includes a 12 V voltage source which is controlled by the WisBlock Core module via IO1 (WB_IO1) of the WisBlock Base. This GPIO must be set to **HIGH** before sampling. The 12 V voltage source is designed to provide power supply to 4-20 mA sensors. The majority of 4-20 mA sensor works in the 9-24 V range. Before connecting a sensor to the RAK5801 module, you must be sure that the sensor can safely operate at 12 V.

For RAK5801, the accessible ADC pin assignments are defined as follows in the Arduino IDE:

- `WB_IO4` for AIN0, ADC Input pin
- `WB_A1` for AIN1, ADC Input pin

> 📝 **NOTE**
>
> - **WB_IO4** for AIN0 only works for the RAK4631 core module.
> - If you will be using the RAK11200 core module, there are few hardware modifications needed to configure. Check the Pin Definition section in the Datasheet.

These are the quick links that go directly to the software guide for the specific WisBlock Core module you use:

- RAK5801 in RAK4631 WisBlock Core Guide
- RAK5801 in RAK11200 WisBlock Core Guide

- RAK5801 in RAK11310 WisBlock Core Guide
- LoRaWAN Hydraulic Pressure Monitoring with RAK5801

RAK5801 in RAK4631 WisBlock Core Guide

Sensor Connection on RAK5801

This is just an example and illustration on how to use the RAK5801 for external analog sensors. There are two analog input (ADC) pins available on the RAK5801, you can use any of the ADC pins as long as your sensors operate at 3.3 V or 12 V with 4-20 mA operating current.

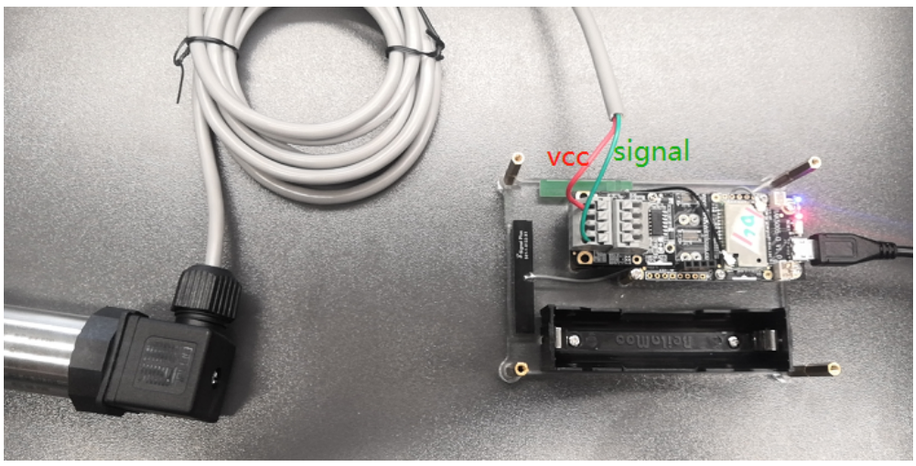


Figure 12: Connecting the RAK5801 to the SUP-P300 Hydraulic Pressure Sensor

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

1. First, you need to select the RAK4631 WisBlock Core.

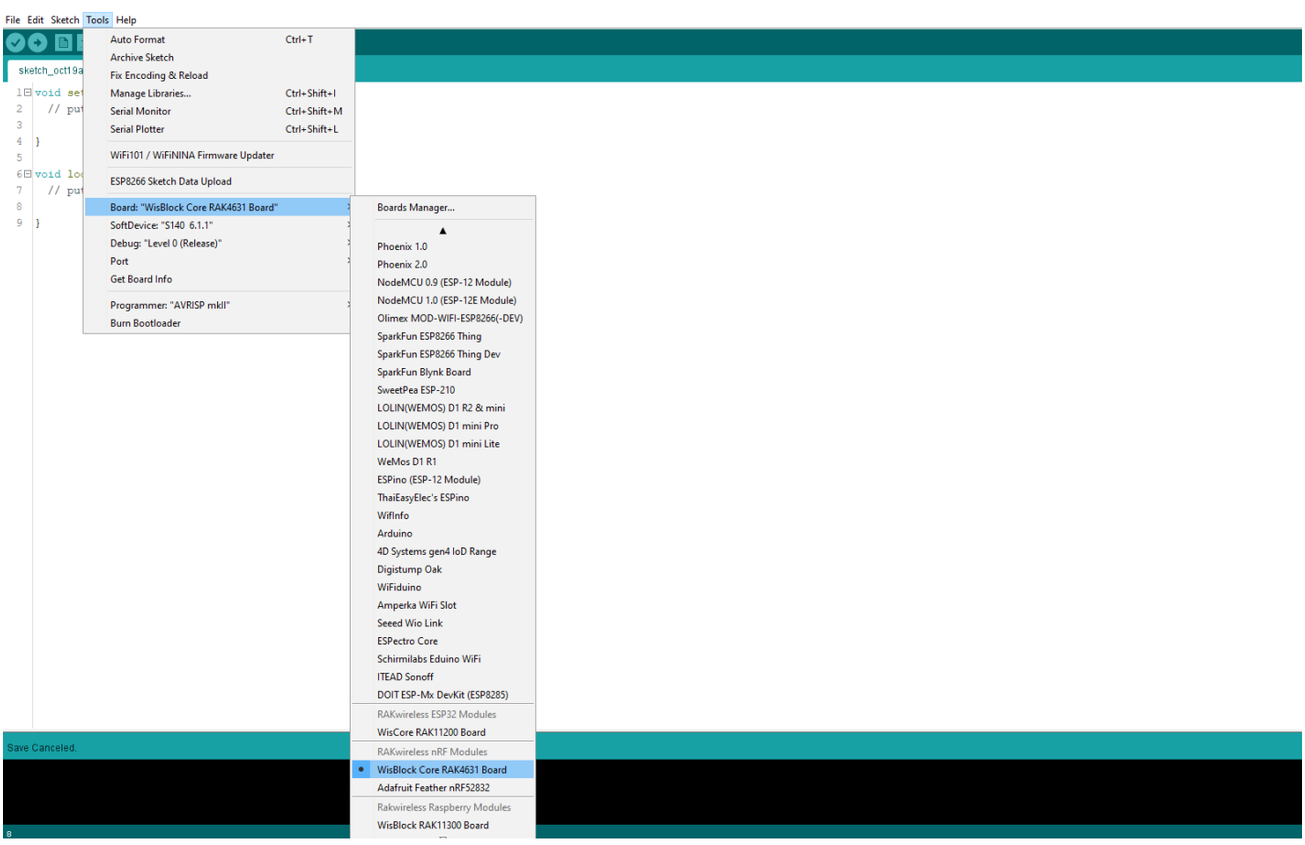


Figure 13: Selecting RAK4631 as WisBlock Core

2. The Basic Sample Code for RAK5801 in GitHub will work on RAK4631 WisBlock Core. You can also open the example codes depending on your WisBlock Core, as shown in **Figure 14**.

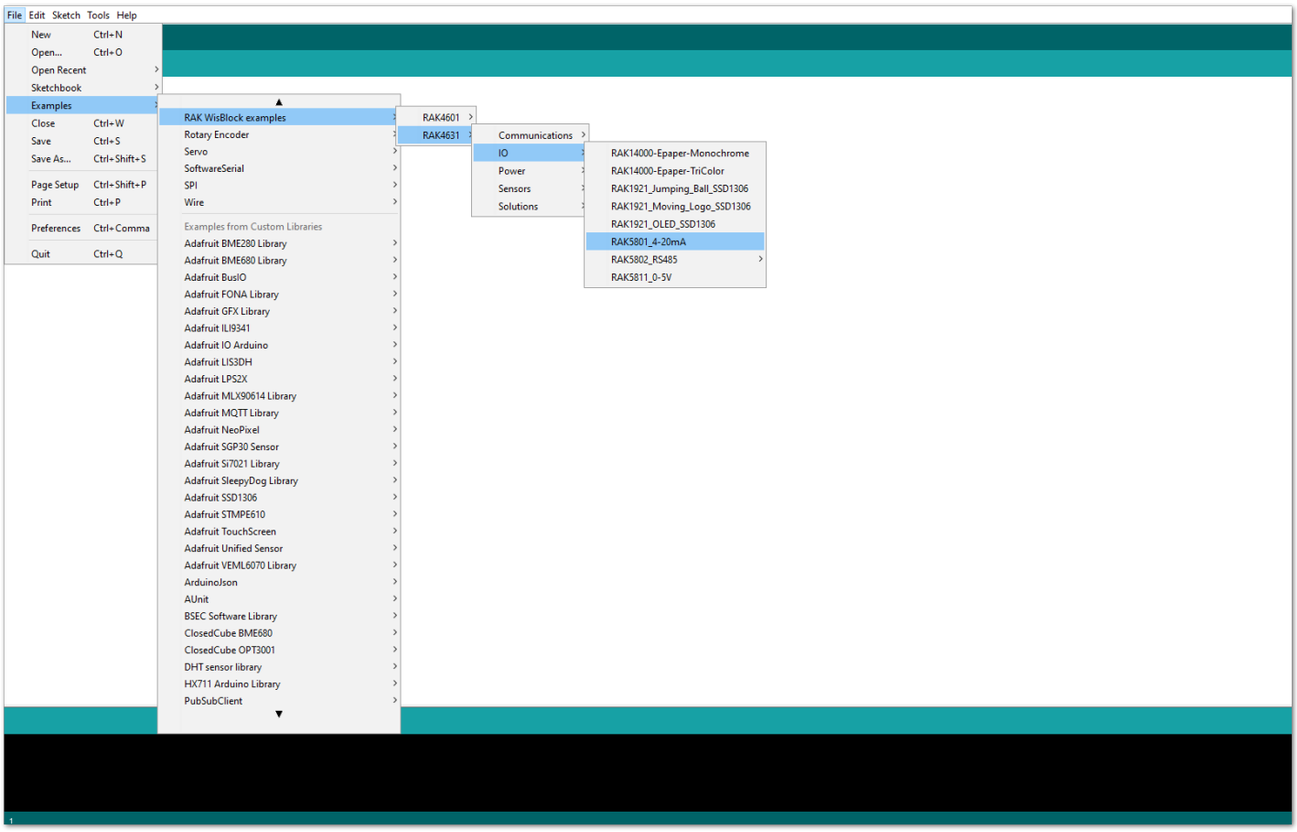


Figure 14: Opening RAK5801 example for RAK4631 WisBlock Core

Code Explanation

1.1 Initializes WisBlock IO 4-20mA Board

```c
{
  /* WisBLOCK 5801 Power On*/

  pinMode(WB_IO1, OUTPUT);
  digitalWrite(WB_IO1, HIGH);

  /* WisBLOCK 5801 Power On*/
}
```

1.2 Get ADC Results

```c
{
    for (i = 0; i < NO_OF_SAMPLES; i++)
  {
    mcu_ain_raw += analogRead(WB_A1);       // select the input pin A1 for the analog sensor
  }
  average_raw = mcu_ain_raw / i;
}
```

1.3 Calculate the Current Value

```c
{
  voltage_ain = average_raw * 3.6 / 1024;     //raef 3.6v / 10bit ADC
  current_sensor = voltage_ain / 149.9 * 1000;  //WisBlock RAK5801 (0 ~ 20mA) I=U/149.9*1000 (mA)
}
```

3. After opening the example code, you can now select the right port and upload the code.

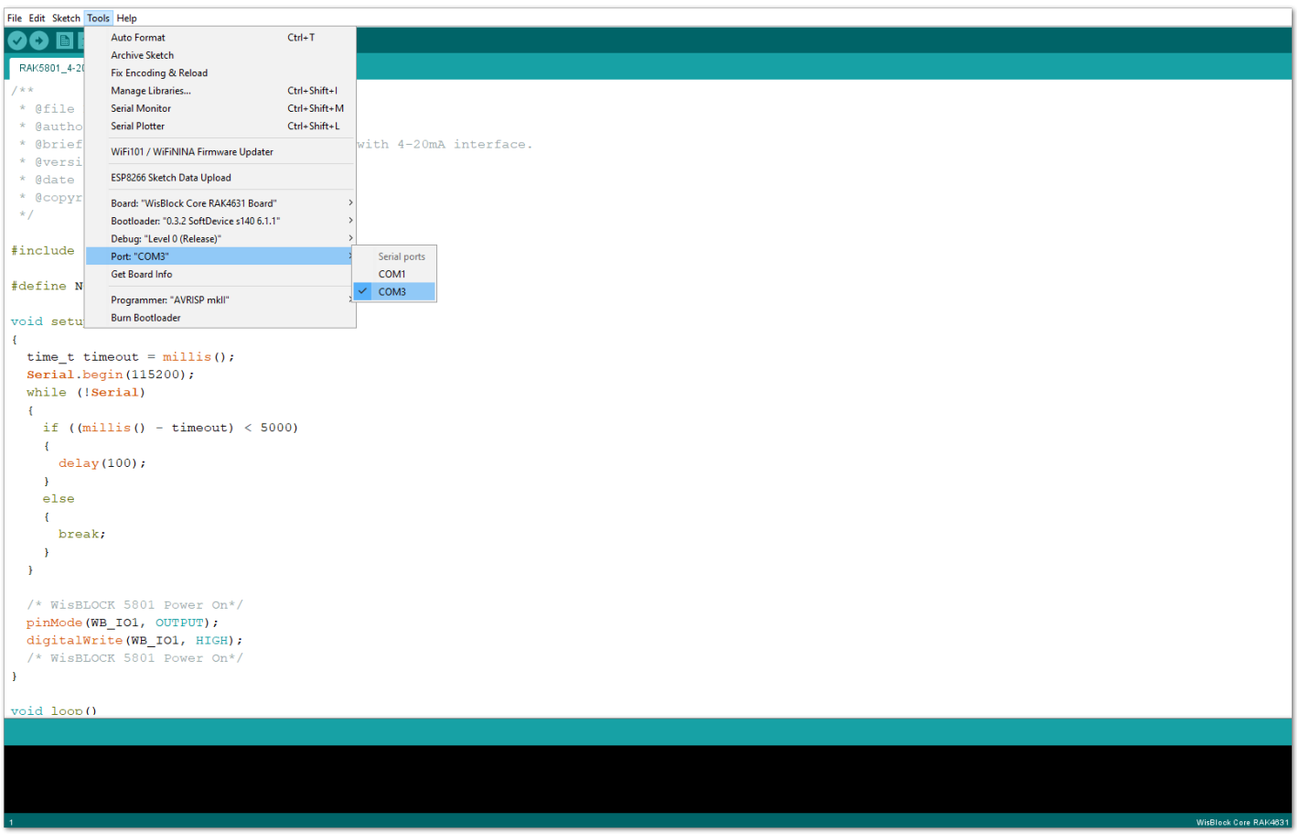


Figure 15: Selecting the correct Serial Port

4. When you successfully uploaded the example sketch, open the Serial Monitor of the Arduino IDE to see the sensor's reading logs, as shown in **Figure 16**.

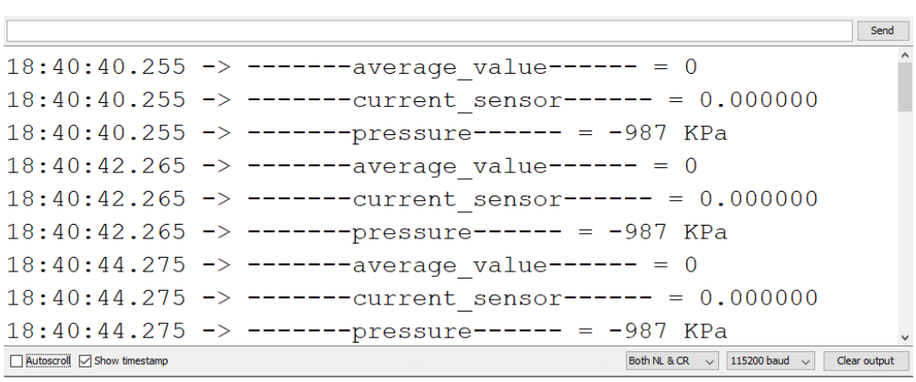


Figure 16: SUP-P300 Hydraulic Pressure sensor data logs

RAK5801 in RAK11200 WisBlock Core Guide

Sensor Connection on RAK5801

This is just an example and illustration on how to use the RAK5801 for external analog sensors. There are two analog input (ADC) pins available on the RAK5801. You can use any of the ADC pins as long as your sensors operate at 3.3 V or 12 V with 4-20 mA operating current.

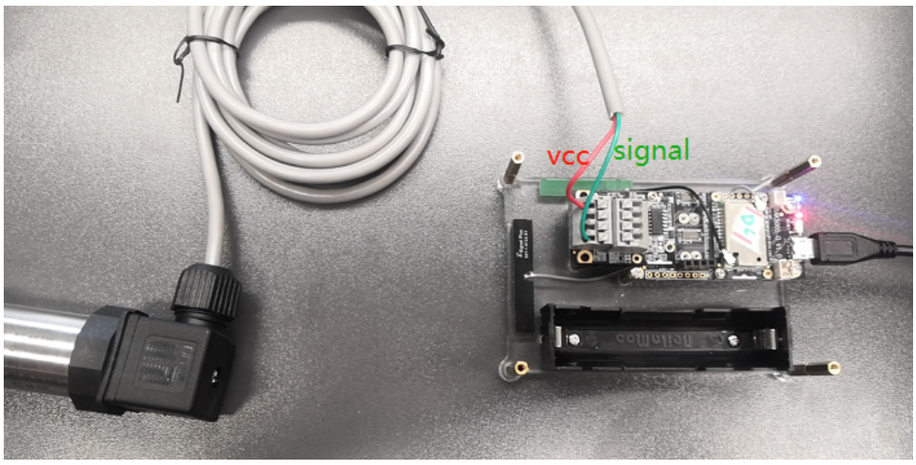


Figure 17: Connecting the RAK5801 to the SUP-P300 Hydraulic Pressure Sensor

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

1. First, you need to select the RAK11200 WisBlock Core.

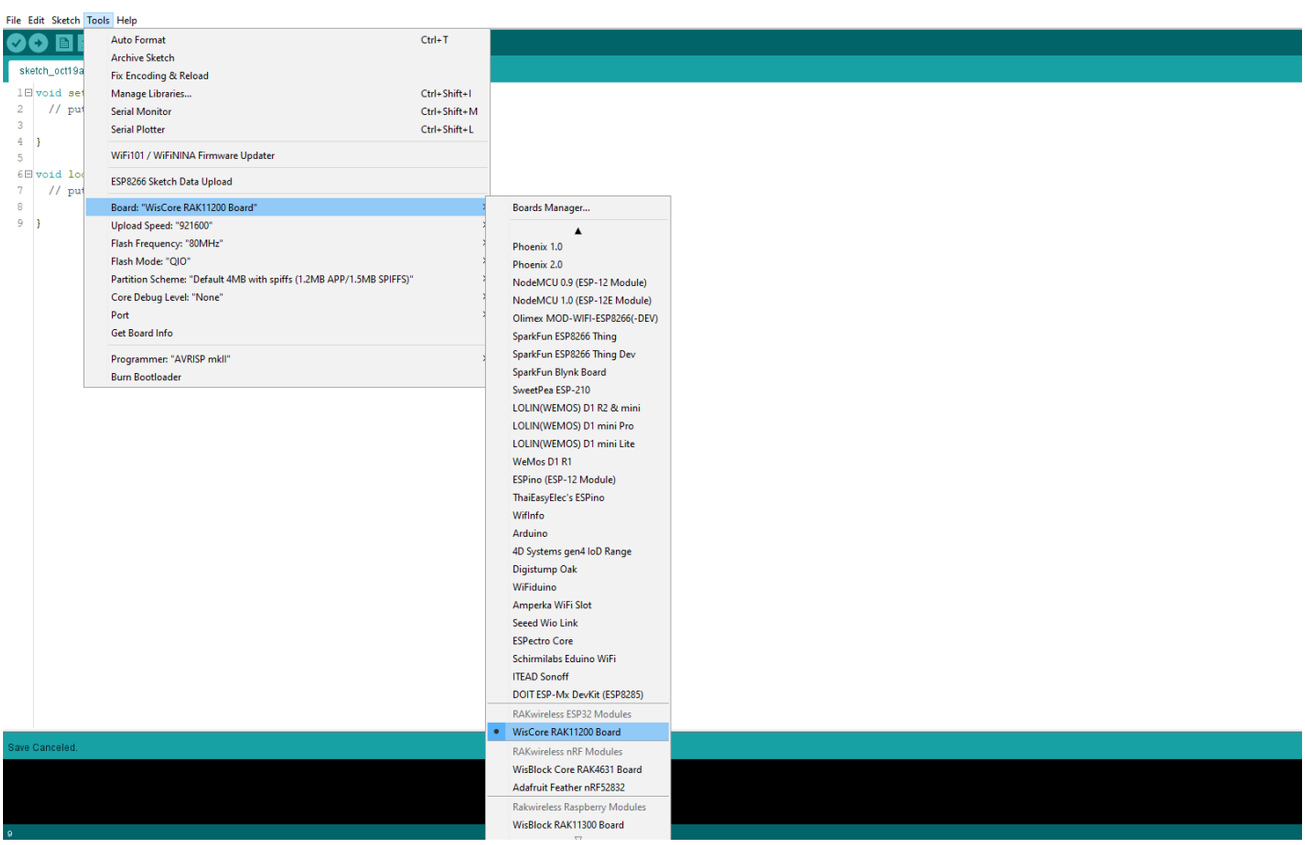


Figure 18: Selecting RAK11200 as WisBlock Core

2. The Basic Sample Code for RAK5801 in GitHub will work on RAK11200 WisBlock Core. You can also open the example codes depending on your WisBlock Core, as shown in **Figure 19**.

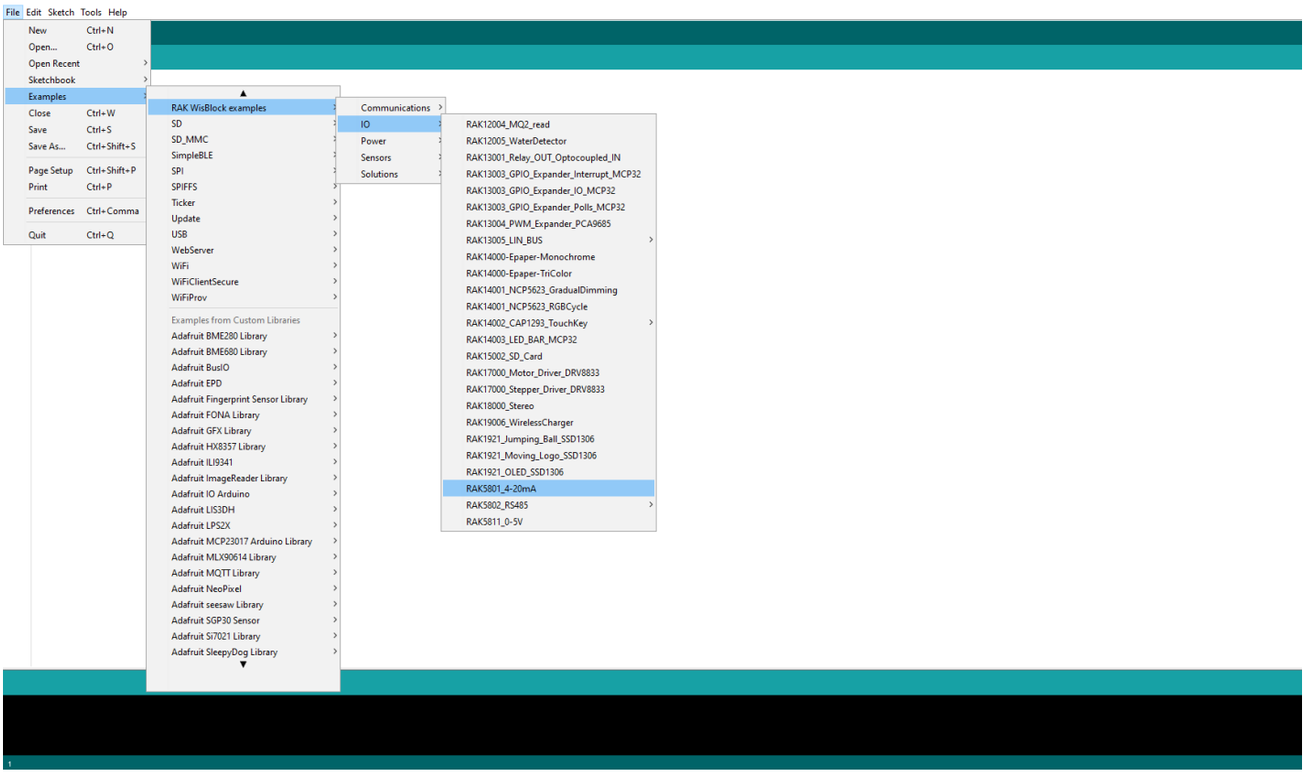


Figure 19: Opening RAK5801 example for RAK11200 WisBlock Core

Code Explanation

1.1 Initializes WisBlock IO 4-20mA Board

```c
{
  /* WisBLOCK 5801 Power On*/

  pinMode(WB_IO1, OUTPUT);
  digitalWrite(WB_IO1, HIGH);

  /* WisBLOCK 5801 Power On*/
}
```

1.2 Get ADC Results

```c
{
    for (i = 0; i < NO_OF_SAMPLES; i++)
  {
    mcu_ain_raw += analogRead(WB_A1);       // select the input pin A1 for the analog sensor
  }
  average_raw = mcu_ain_raw / i;
}
```

1.3 Calculate the Current Value

```c
{
  voltage_ain = average_raw * 3.6 / 1024;     //raef 3.6v / 10bit ADC
  current_sensor = voltage_ain / 149.9 * 1000;  //WisBlock RAK5801 (0 ~ 20mA) I=U/149.9*1000 (mA)
}
```

3. After opening the example code, you can now select the right port and upload the code.

> 📝 **NOTE**
>
> RAK11200 requires the **Boot0** pin to be configured properly first before uploading. If not done properly, uploading the source code to RAK11200 will fail. Check the full details on the RAK11200 Quick Start Guide .

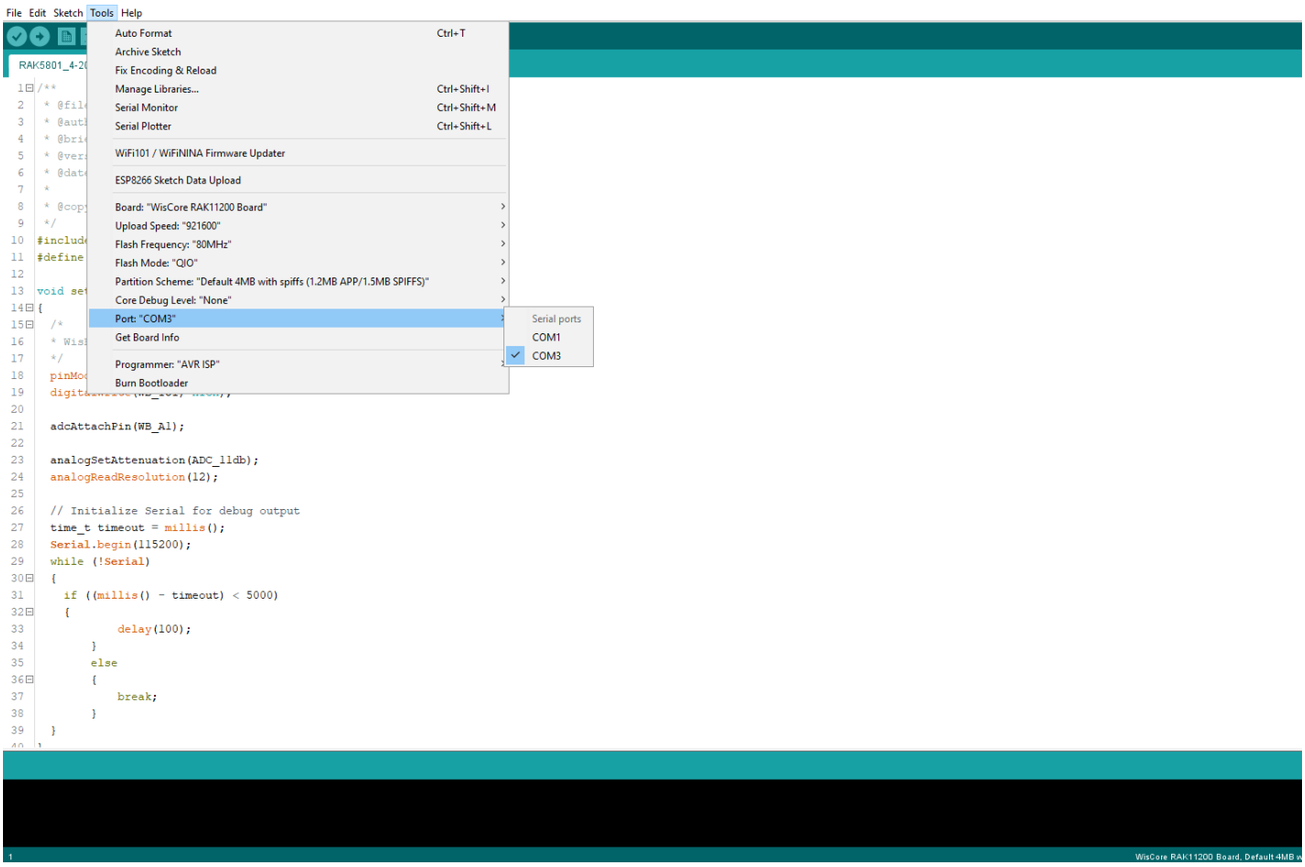


Figure 20: Selecting the correct Serial Port

4. When you successfully uploaded the example sketch, open the Serial Monitor of the Arduino IDE to see the sensor's reading logs, as shown in **Figure 21**.

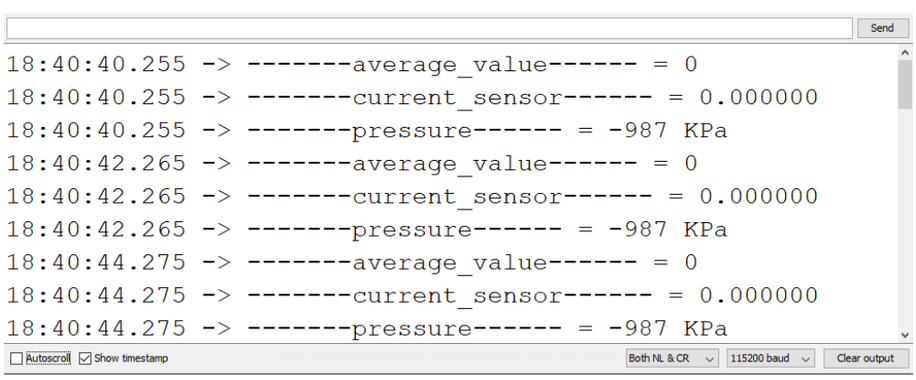


Figure 21: SUP-P300 Hydraulic Pressure sensor data logs

RAK5801 in RAK11310 WisBlock Core Guide

Sensor Connection on RAK5801

This is just an example and illustration on how to use the RAK5801 for external analog sensors. There are two analog input (ADC) pins available on the RAK5801. You can use any of the ADC pins as long as your sensors operate at 3.3 V or 12 V with 4-20 mA operating current.

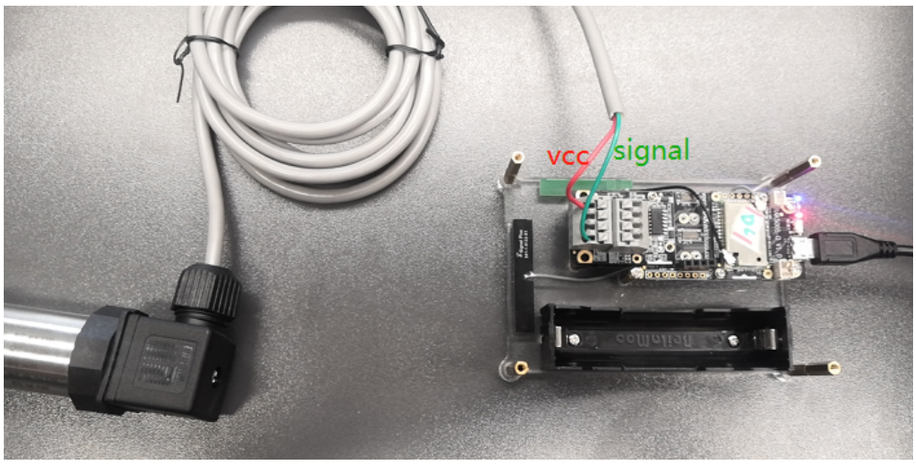


Figure 22: Connecting the RAK5801 to the SUP-P300 Hydraulic Pressure Sensor

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

1. First, you need to select the RAK11310 WisBlock Core.

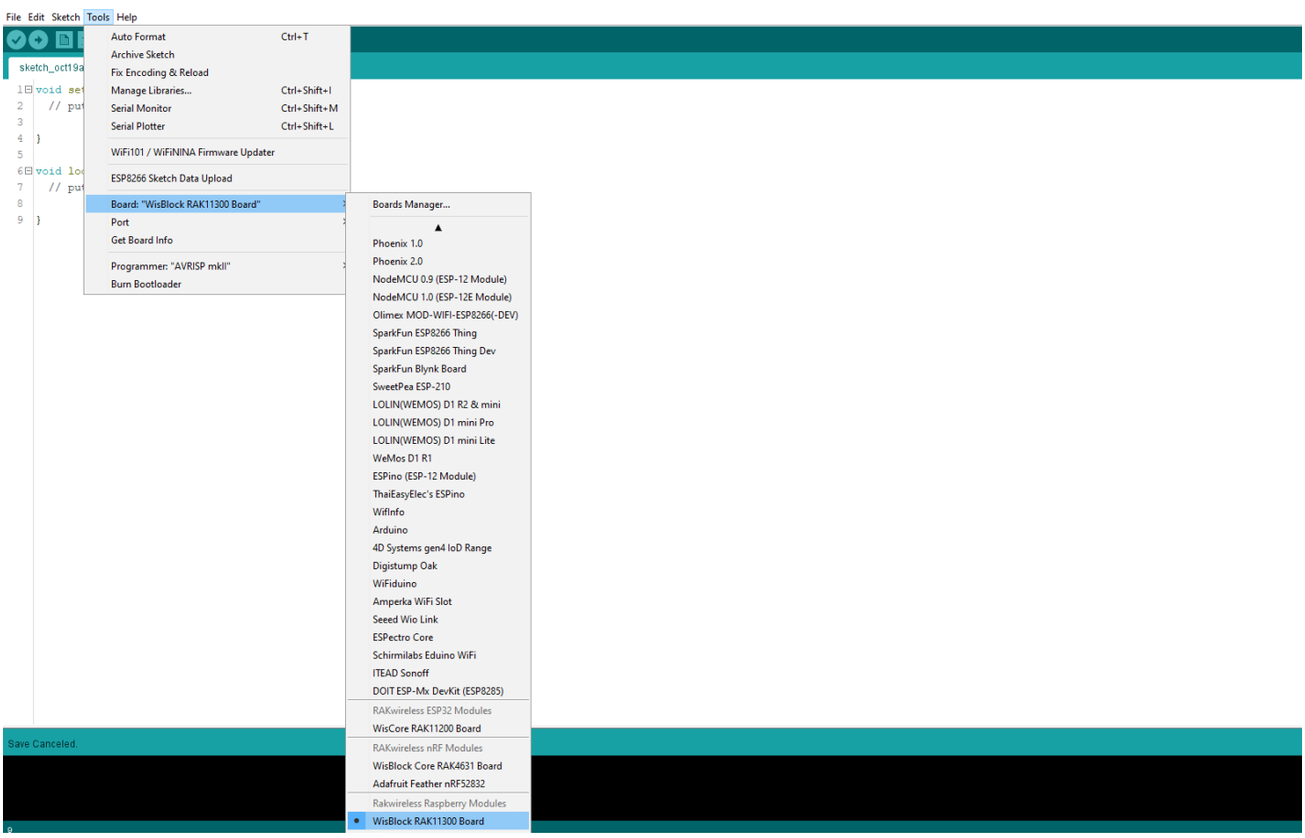


Figure 23: Selecting RAK11310 as WisBlock Core

2. The Basic Sample Code for RAK5801 in GitHub will work on RAK11310 WisBlock Core. You can also open the example codes depending on your WisBlock Core, as shown in **Figure 24**.

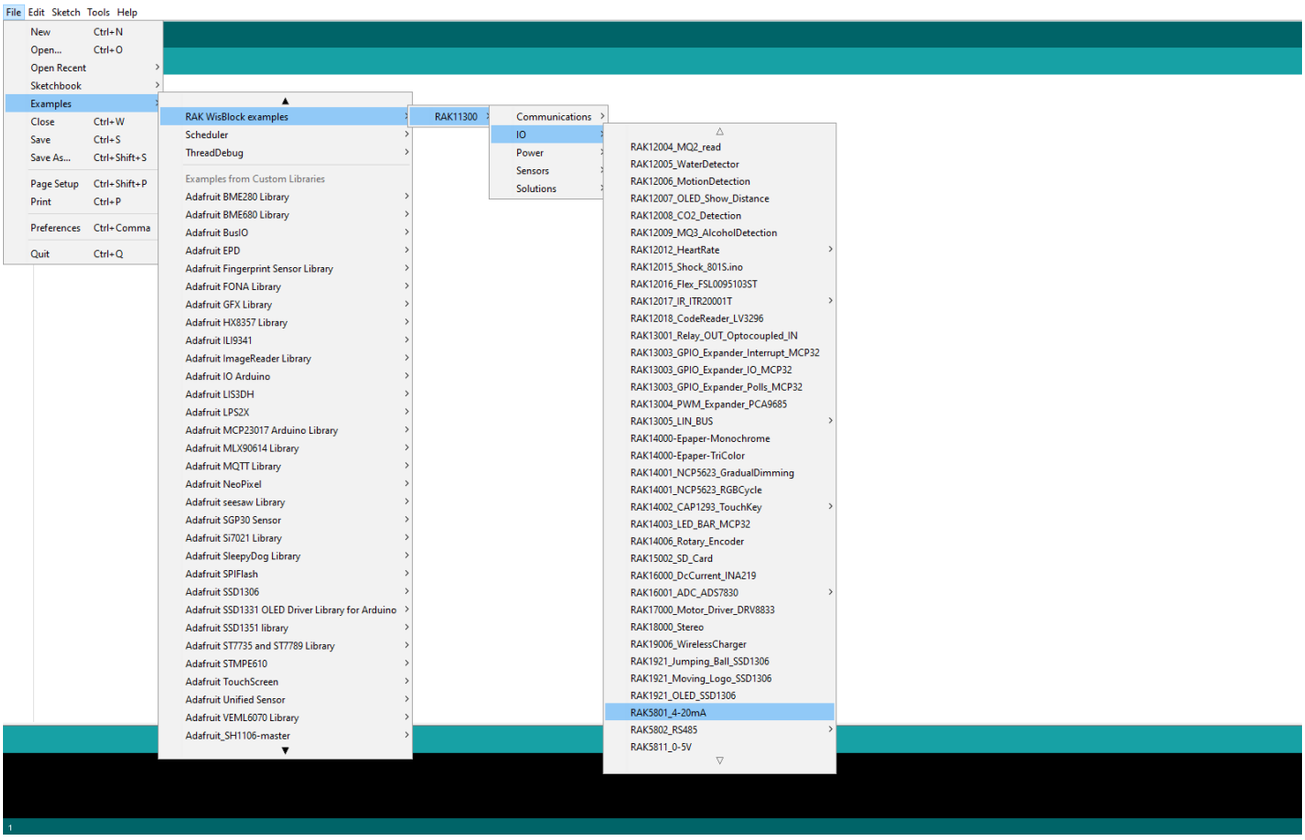


Figure 24: Opening RAK5801 example for RAK11310 WisBlock Core

Code Explanation

1.1 Initializes WisBlock IO 4-20mA Board

```c
{
  /* WisBLOCK 5801 Power On*/

  pinMode(WB_IO1, OUTPUT);
  digitalWrite(WB_IO1, HIGH);

  /* WisBLOCK 5801 Power On*/
}
```

1.2 Get ADC Results

```c
{
    for (i = 0; i < NO_OF_SAMPLES; i++)
  {
    mcu_ain_raw += analogRead(WB_A1);       // select the input pin A1 for the analog sensor
  }
  average_raw = mcu_ain_raw / i;
}
```

1.3 Calculate the Current Value

```c
{
  voltage_ain = average_raw * 3.6 / 1024;     //raef 3.6v / 10bit ADC
  current_sensor = voltage_ain / 149.9 * 1000;  //WisBlock RAK5801 (0 ~ 20mA) I=U/149.9*1000 (mA)
}
```

3. After opening the example code, you can now select the right port and upload the code.

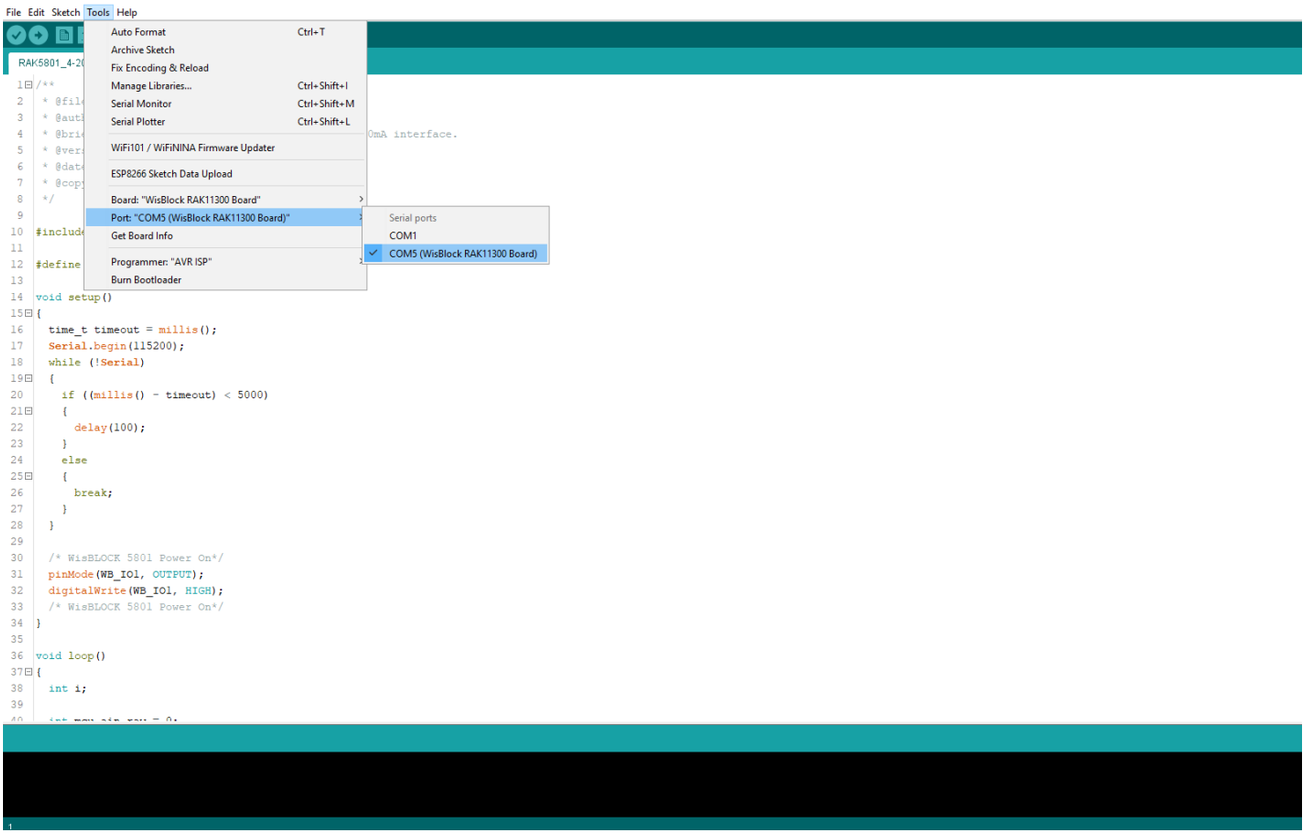


Figure 25: Selecting the correct Serial Port

4. When you successfully uploaded the example sketch, open the Serial Monitor of the Arduino IDE to see the sensor's reading logs, as shown in **Figure 26**.

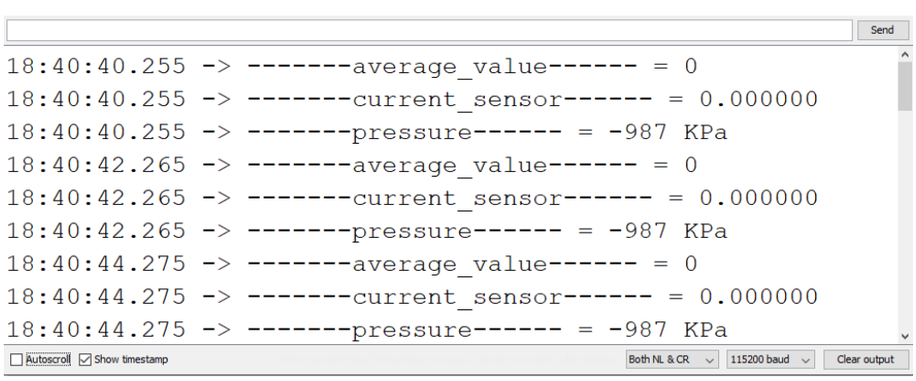


Figure 26: SUP-P300 Hydraulic Pressure sensor data logs

LoRaWAN Hydraulic Pressure Monitoring with RAK5801

For WisBlock Core RAK4630, check the example for LoRaWAN Hydraulic Pressure Monitoring with RAK5801 Module.

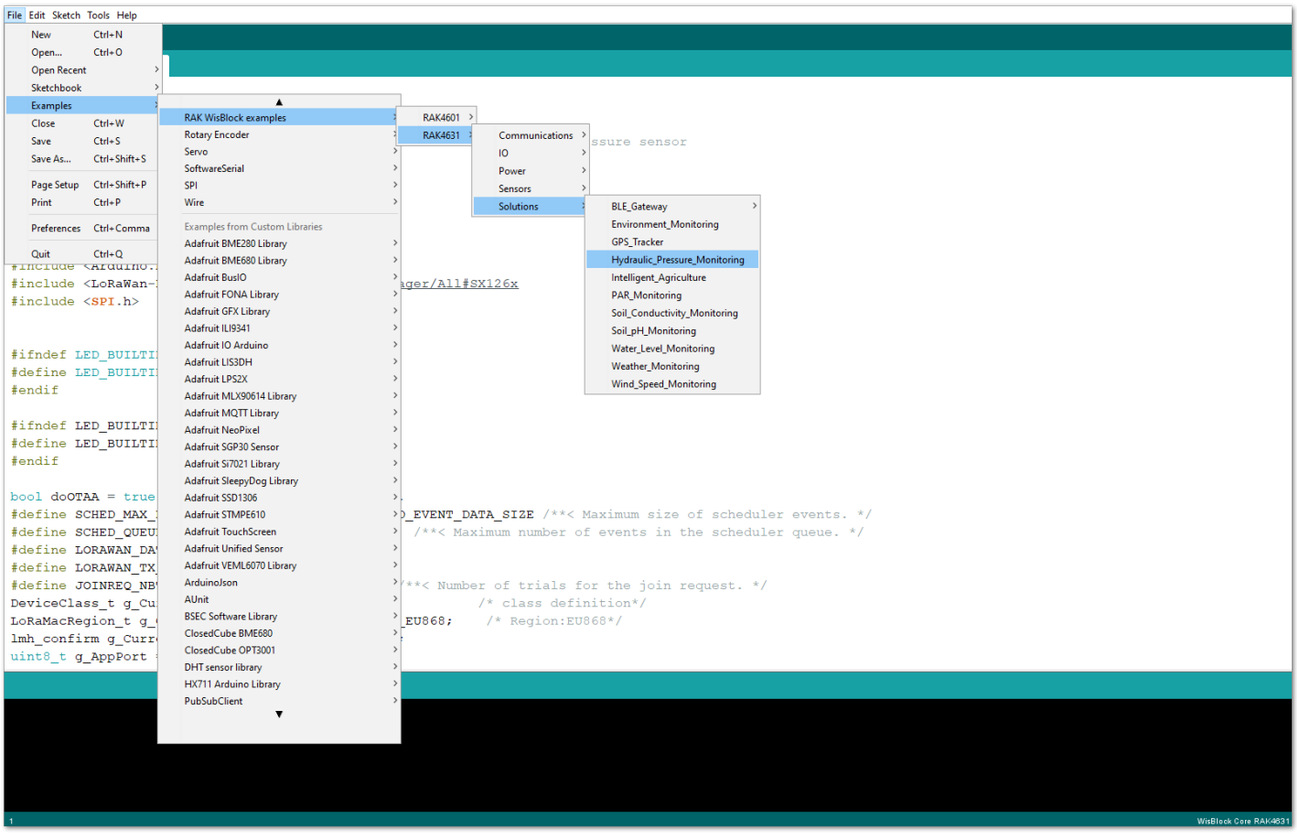


Figure 27: LoRaWAN Hydraulic Pressure Monitoring example code

Last Updated: 1/14/2022, 8:59:51 AM