RAK13003 Quick Start Guide

Prerequisite

What Do You Need?

Before going through each and every step on using RAK13003 WisBlock module, make sure to prepare the necessary items listed below:

Hardware

- RAK13003 IO Expansion Module
- Your choice of WisBlock Base
- Your choice of WisBlock Core
- Light-emitting diode or LEDs
- USB Cable
- Li-Ion/LiPo battery (optional)
- Solar charger (optional)

Software

Arduino

- Download and install Arduino IDE .
- To add the RAKwireless Core boards on your Arduino Boards Manager, install the RAKwireless Arduino BSP .

Product Configuration

Hardware Setup

The RAK13003 is an IO expansion module that can be mounted to the IO slot of the WisBlock Baseboard. It offers 16 bidirectional I/O ports by using MCP23017 IC from Microchip. The configuration of this module is via the I2C interface, and it supports both standard and fast I2C modes.

Row/Column	Column 1	Column 2	Column 3	Column 4
Row 1	PA0	PA1	PB6	PB7
Row 2	PA2	PA3	PB4	PB5
Row 3	PA4	PA5	PB2	PB3
Row 4	PA6	PA7	PB0	PB1
Row 5	GND	VCC	GND	VCC

For more information about RAK13003, refer to the Datasheet.

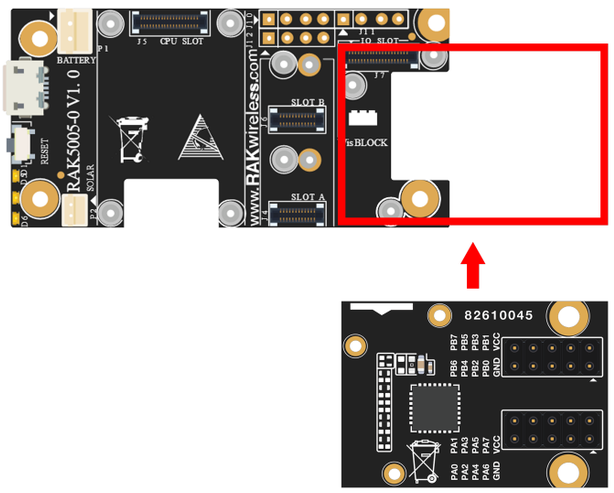

Figure 1: RAK13003 Connection to WisBlock Base module

Assembling and Disassembling of WisBlock Modules

Assembling Procedure

The RAK13003 module can be mounted on the IO slot of the WisBlock Base board, as shown in **Figure 2**. Also, always secure the connection of the WisBlock module by using the compatible screws.

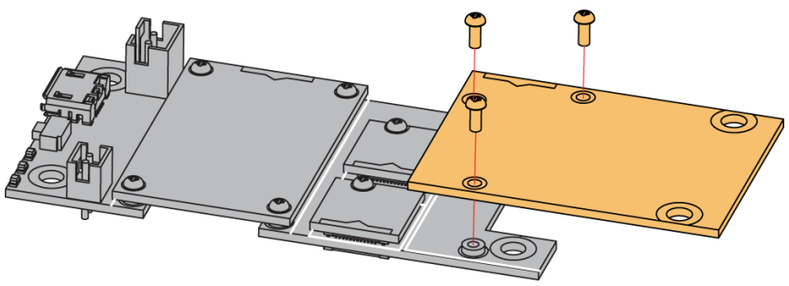

Figure 2: RAK13003 mounting connection to WisBlock Base module

Disassembling Procedure

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

1. First, remove the screws.

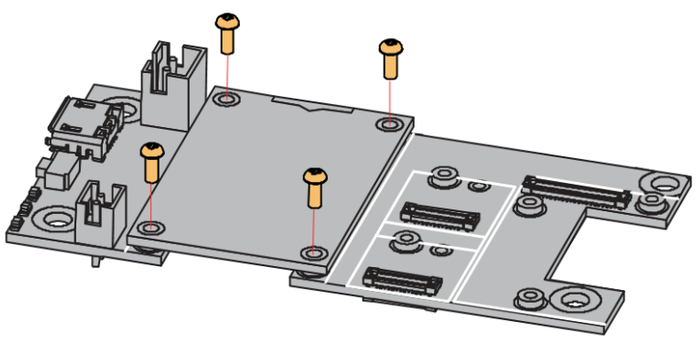

Figure 3: Removing screws from the WisBlock module

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

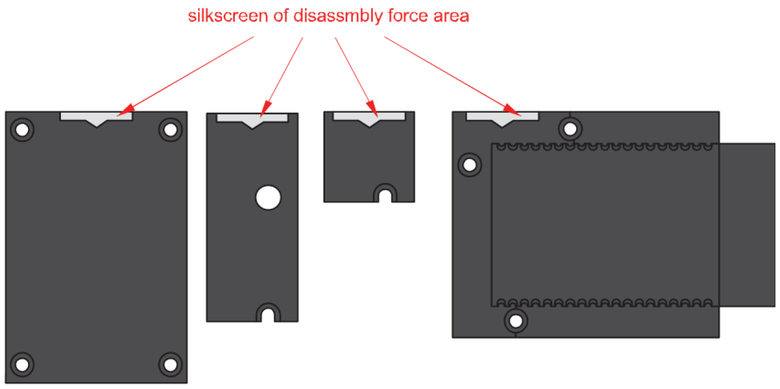

Figure 4: Detaching silkscreen on the WisBlock module

3. Apply force to the module at the position of the connector, as shown in **Figure 5**, to detach the module from the baseboard.

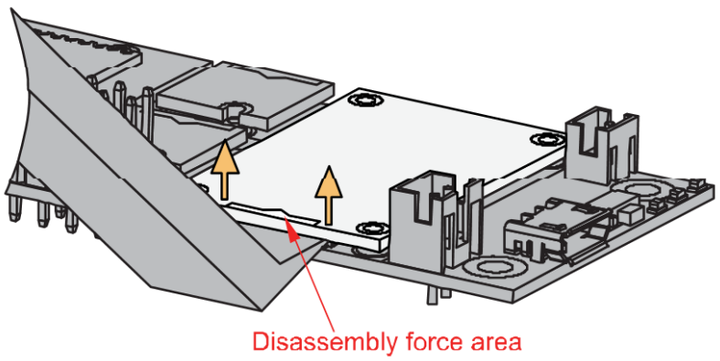

Figure 5: Applying even forces on the proper location of a WisBlock module

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

Software Configuration and Example

In the following example, you will be using the RAK13003 WisBlock IO Expansion Module to power LEDs.

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

- RAK13003 in RAK4631 WisBlock Core Guide
- RAK13003 in RAK11200 WisBlock Core Guide
- RAK13003 in RAK11300 WisBlock Core Guide

RAK13003 in RAK4631 WisBlock Core Guide

Arduino Setup

Shown in **Figure 6** is the illustration on how to use the RAK13003 IO Expansion Module to power ON LEDs using digitalWrite function.

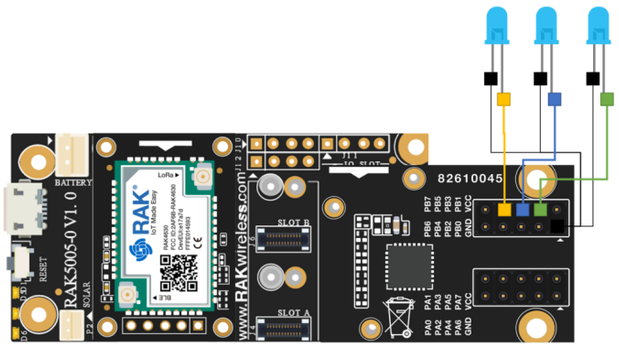

Figure 6: RAK13003 as Output to LEDs

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

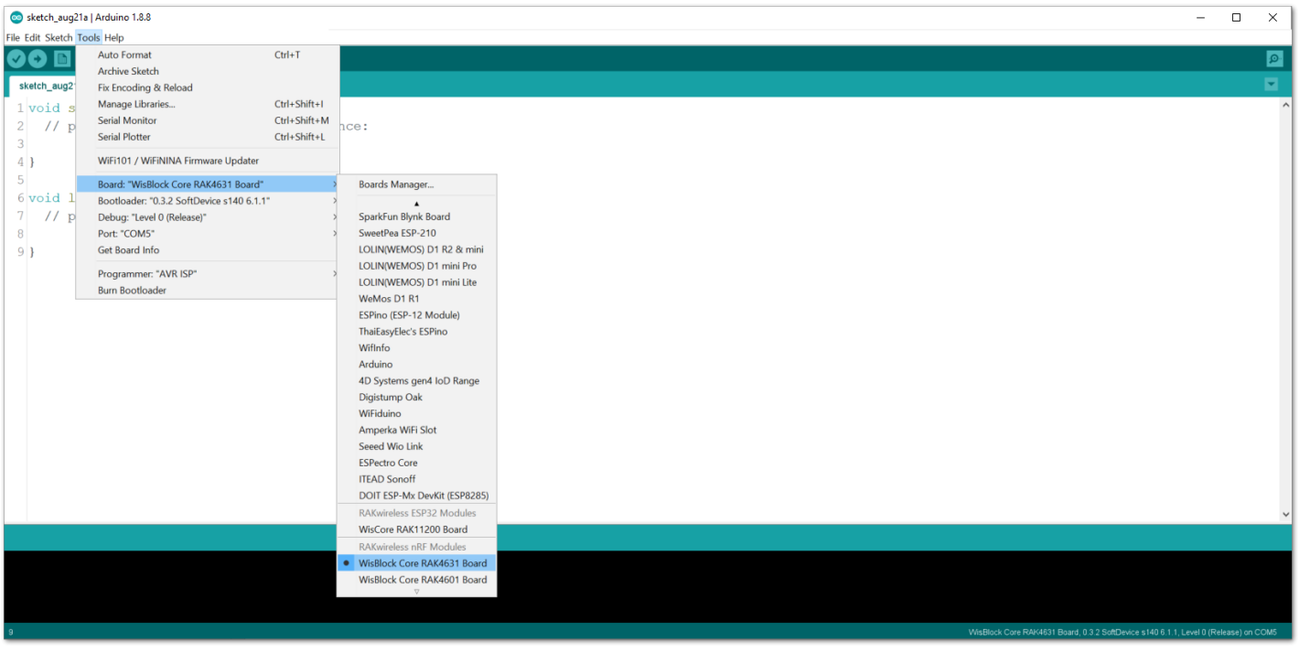

Figure 7: Selecting RAK4631 as WisBlock Core

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

```c
/**
   @file RAK13003_GPIO_Expander_IO_MCP32.ino
   @author rakwireless.com
   @brief Use IIC to expand 16 GPIO.
          Configure PA input PB output, or PA output PB input.Serial port print GPIO status.
   @version 0.1
   @date 2021-2-24
   @copyright Copyright (c) 2021
**/
#include <Wire.h>
#include <Adafruit_MCP23017.h>  //click here to get the library: http://librarymanager/All#Adafru

#define PAIN_PBOUT  //PB is set as output here and PA as input.
//#define PAOUT_PBIN

Adafruit_MCP23017 mcp;

void setup()
{
  pinMode(WB_IO2, OUTPUT);
  digitalWrite(WB_IO2, 1);

  // Reset device
  pinMode(WB_IO4, OUTPUT);
  digitalWrite(WB_IO4, 1);
  delay(10);
  digitalWrite(WB_IO4, 0);
  delay(10);
  digitalWrite(WB_IO4, 1);
  delay(10);

  // Initialize Serial for debug output
  time_t timeout = millis();
  Serial.begin(115200);
  while (!Serial)
  {
    if ((millis() - timeout) < 5000)
    {
      delay(100);
    }
    else
    {
      break;
    }
  }

  Serial.println("MCP23017 GPIO Input Output Test.");

  mcp.begin(); // use default address 0.

#ifdef PAIN_PBOUT
  for(int i=0 ;i < 8 ;i++)
  {
    mcp.pinMode(i, INPUT);  // PA input.
  }
  for(int j=8 ;j < 16 ;j++)
  {
    mcp.pinMode(j, OUTPUT); // PB output.
  }
  mcp.digitalWrite(8, LOW); // The output state of the PB port can be changed to high or low leve
  mcp.digitalWrite(9, HIGH); //PIN PB1
  mcp.digitalWrite(10, LOW); //PIN PB2
```

```
  mcp.digitalWrite(11, LOW); //PIN PB3

  mcp.digitalWrite(12, LOW); //PIN PB4
  mcp.digitalWrite(13, LOW); //PIN PB5
  mcp.digitalWrite(14, LOW); //PIN PB6
  mcp.digitalWrite(15, HIGH);//PIN PB7

  Serial.println();
  for(int i=0; i < 8; i++ )
  {
    if(mcp.digitalRead(i) == 1)
      Serial.printf("GPIO A %d Read High\r\n",i);
    else
      Serial.printf("GPIO A %d Read Low\r\n",i);
  }
#endif

#ifdef PAOUT_PBIN
  for(int i=0 ;i < 8 ;i++)
  {
    mcp.pinMode(i, OUTPUT); // PA output.
  }
  for(int j=8 ;j < 16 ;j++)
  {
    mcp.pinMode(j, INPUT);  // PB input.
  }
  mcp.digitalWrite(0, LOW); // The output state of the PA port can be changed to high or low leve
  mcp.digitalWrite(1, HIGH);
  mcp.digitalWrite(2, LOW);
  mcp.digitalWrite(3, HIGH);

  mcp.digitalWrite(4, LOW);
  mcp.digitalWrite(5, HIGH);
  mcp.digitalWrite(6, LOW);
  mcp.digitalWrite(7, HIGH);
  Serial.println();
  for(int i=8; i < 16; i++ )
  {
    if(mcp.digitalRead(i) == 1)
      Serial.printf("GPIO B %d Read High\r\n",i-8);
    else
      Serial.printf("GPIO B %d Read Low\r\n",i-8);
  }
#endif
}
void loop()
{

}
```

> 📝 **NOTE**
>
> If you experience any error in compiling the example sketch, check the updated code for the RAK4631 WisBlock Core Module that can be found on the RAK13003 WisBlock Example Code Repository .

3. Install the required library, as shown in **Figure 8**.

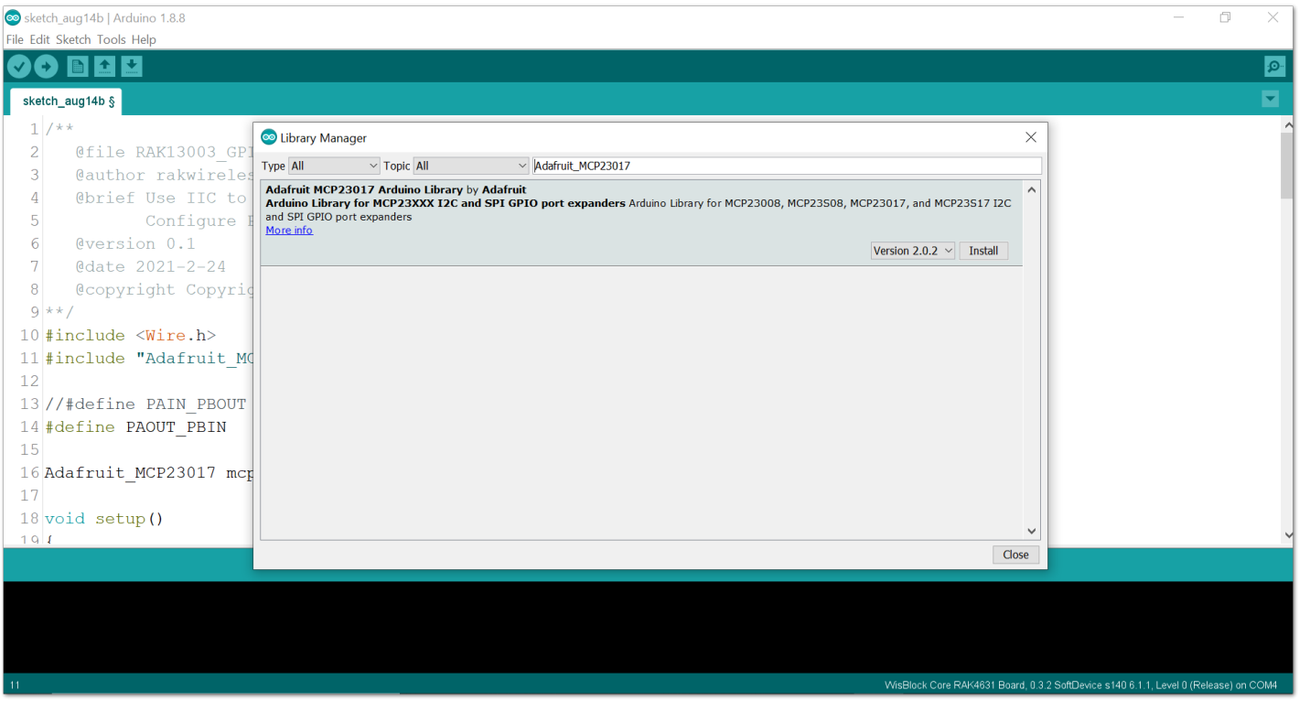

Figure 8: Installing the Library

4. Choose Version 1.3.0 of the library, as shown in **Figure 9**

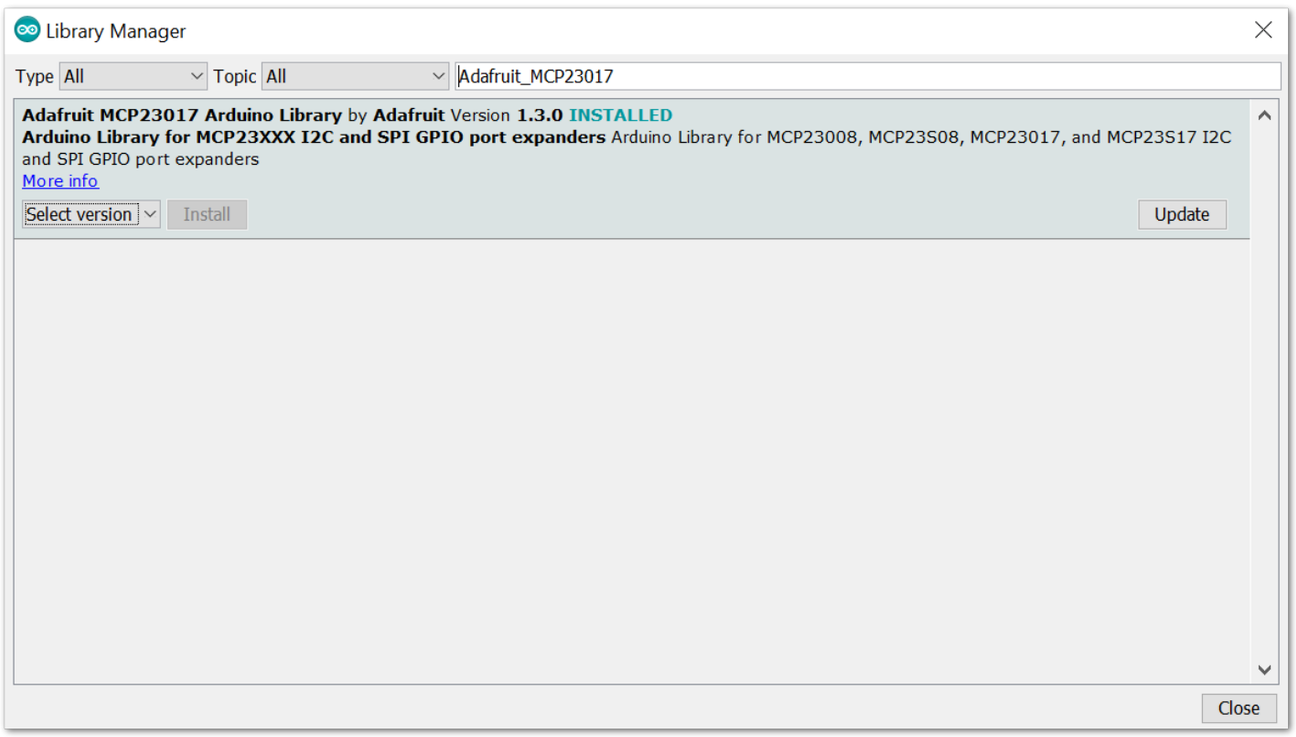

Figure 9: Selecting Version 1.3.0

5. Select the correct port and upload your code, as shown in **Figure 10** and **Figure 11**.

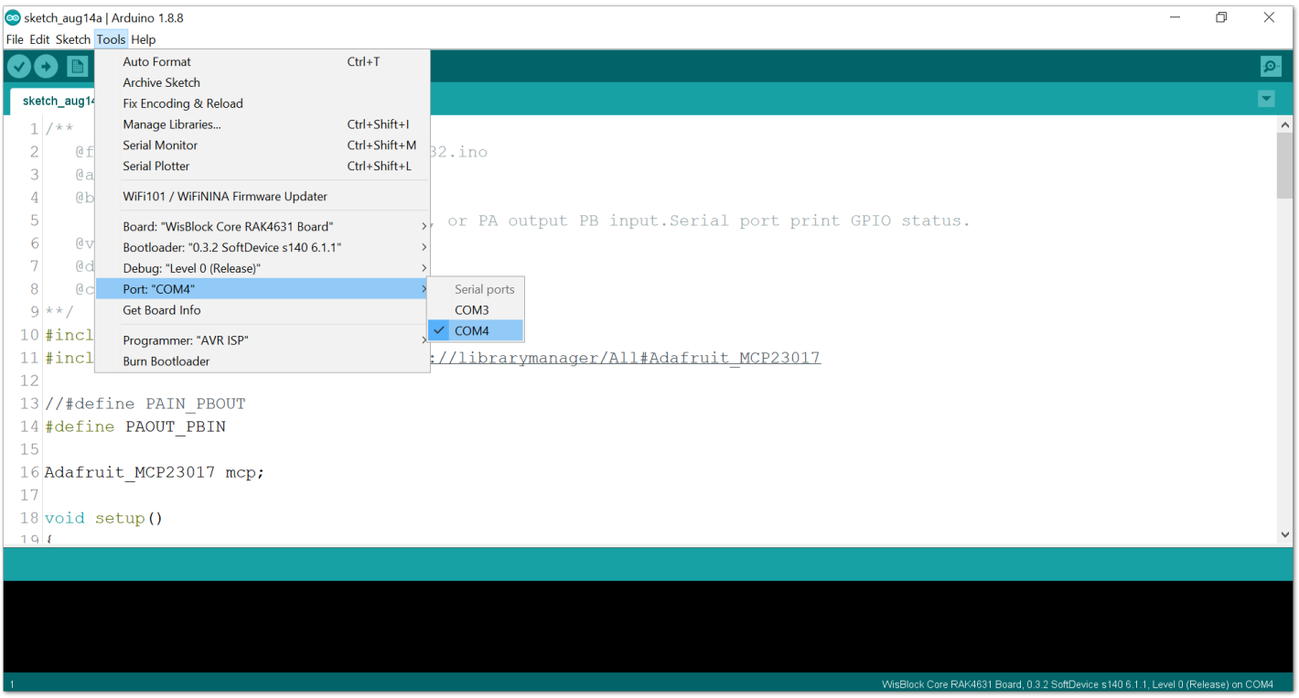

Figure 10: Selecting the correct Serial Port

Figure 11: Uploading code

6. When you have successfully uploaded the example sketch, you can see that the LEDs are powered ON. You can also switch PB as INPUT and PA as OUTPUT by changing this line of code shown in **Figure 12**.

RAK13003 §

```
   @author rakwireless.com
   @brief Use IIC to expand 16 GPIO.
          Configure PA input PB output, or PA output PB input.Serial port print GPIO status.
   @version 0.1
   @date 2021-2-24
   @copyright Copyright (c) 2021
**/
#include <Wire.h>
#include <Adafruit_MCP23017.h>  //http://librarymanager/All#Adafruit_MCP23017

//#define PAIN_PBOUT  //PB is set as output here and PA as input.
#define PAOUT_PBIN  //PA is set as output here and PB as input.

Adafruit_MCP23017 mcp;

void setup()
```

Figure 12: Switching between PA and PB

> 📝 **NOTE**
>
> You can use `mcp.digitalWrite(pin_no,state)` and `mcp.digitalRead(pin_no)` to send or read states.

RAK13003 in RAK11200 WisBlock Core Guide

Arduino Setup

Shown in **Figure 13** is the illustration on how to use the RAK13003 IO Expansion Module to power ON LEDs using digitalWrite function.

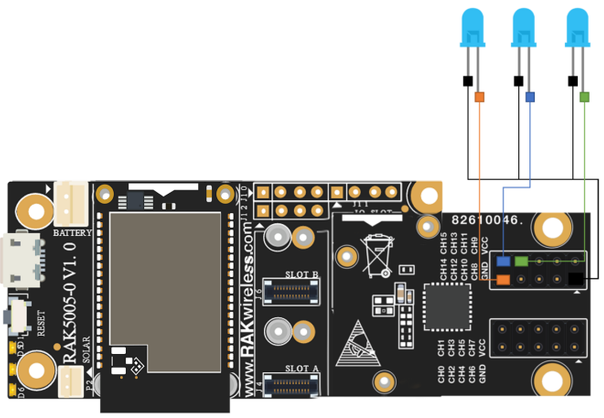

Figure 13: RAK13003 as Output to LEDs

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

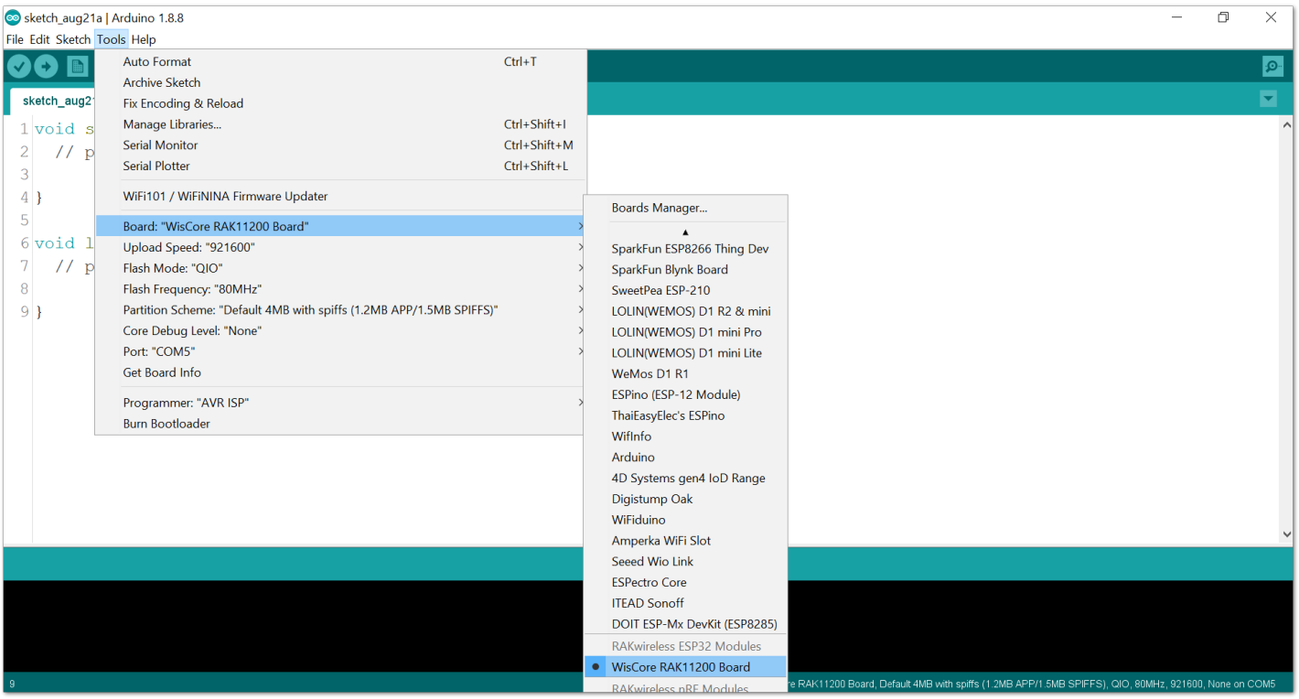

Figure 14: Selecting RAK11200 as WisBlock Core

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

```c
/**
   @file RAK13003_GPIO_Expander_IO_MCP32.ino
   @author rakwireless.com
   @brief Use IIC to expand 16 GPIO.
          Configure PA input PB output, or PA output PB input.Serial port print GPIO status.
   @version 0.1
   @date 2021-2-24
   @copyright Copyright (c) 2021
**/
#include <Wire.h>
#include <Adafruit_MCP23017.h>  //click here to get the library: http://librarymanager/All#Adafru

#define PAIN_PBOUT  //PB is set as output here and PA as input.
//#define PAOUT_PBIN

Adafruit_MCP23017 mcp;

void setup()
{
  pinMode(WB_IO2, OUTPUT);
  digitalWrite(WB_IO2, 1);

  // Reset device
  pinMode(WB_IO4, OUTPUT);
  digitalWrite(WB_IO4, 1);
  delay(10);
  digitalWrite(WB_IO4, 0);
  delay(10);
  digitalWrite(WB_IO4, 1);
  delay(10);

  // Initialize Serial for debug output
  time_t timeout = millis();
  Serial.begin(115200);
  while (!Serial)
  {
    if ((millis() - timeout) < 5000)
    {
      delay(100);
    }
    else
    {
      break;
    }
  }

  Serial.println("MCP23017 GPIO Input Output Test.");

  mcp.begin(); // use default address 0.

#ifdef PAIN_PBOUT
  for(int i=0 ;i < 8 ;i++)
  {
    mcp.pinMode(i, INPUT);  // PA input.
  }
  for(int j=8 ;j < 16 ;j++)
  {
    mcp.pinMode(j, OUTPUT); // PB output.
  }
  mcp.digitalWrite(8, LOW); // The output state of the PB port can be changed to high or low leve
  mcp.digitalWrite(9, HIGH); //PIN PB1
  mcp.digitalWrite(10, LOW); //PIN PB2
```

```
  mcp.digitalWrite(11, LOW); //PIN PB3

  mcp.digitalWrite(12, LOW); //PIN PB4
  mcp.digitalWrite(13, LOW); //PIN PB5
  mcp.digitalWrite(14, LOW); //PIN PB6
  mcp.digitalWrite(15, HIGH);//PIN PB7

  Serial.println();
  for(int i=0; i < 8; i++ )
  {
    if(mcp.digitalRead(i) == 1)
      Serial.printf("GPIO A %d Read High\r\n",i);
    else
      Serial.printf("GPIO A %d Read Low\r\n",i);
  }
#endif

#ifdef PAOUT_PBIN
  for(int i=0 ;i < 8 ;i++)
  {
    mcp.pinMode(i, OUTPUT); // PA output.
  }
  for(int j=8 ;j < 16 ;j++)
  {
    mcp.pinMode(j, INPUT);  // PB input.
  }
  mcp.digitalWrite(0, LOW); // The output state of the PA port can be changed to high or low leve
  mcp.digitalWrite(1, HIGH);
  mcp.digitalWrite(2, LOW);
  mcp.digitalWrite(3, HIGH);

  mcp.digitalWrite(4, LOW);
  mcp.digitalWrite(5, HIGH);
  mcp.digitalWrite(6, LOW);
  mcp.digitalWrite(7, HIGH);
  Serial.println();
  for(int i=8; i < 16; i++ )
  {
    if(mcp.digitalRead(i) == 1)
      Serial.printf("GPIO B %d Read High\r\n",i-8);
    else
      Serial.printf("GPIO B %d Read Low\r\n",i-8);
  }
#endif
}
void loop()
{

}
```

> 📝 **NOTE**
>
> If you experience any error in compiling the example sketch, check the updated code for the RAK11200 WisBlock Core Module that can be found on the RAK13003 WisBlock Example Code Repository .

3. Install the required library, as shown in **Figure 15**.

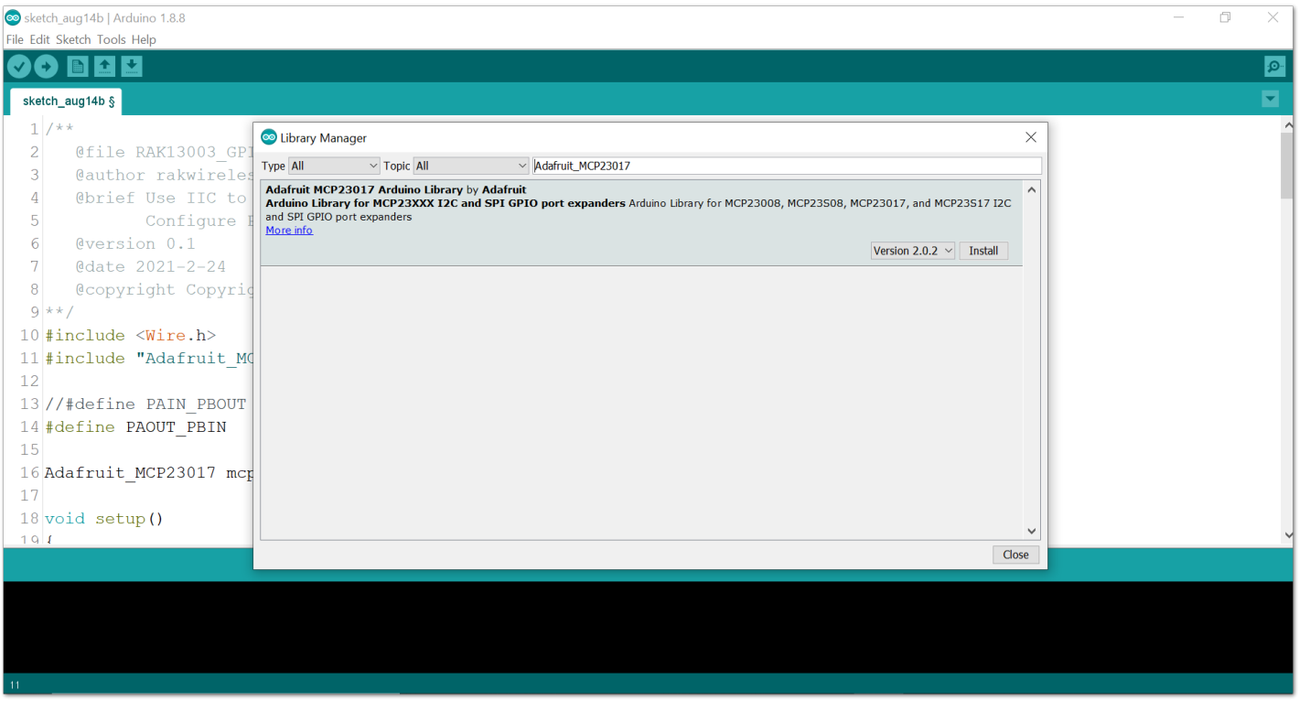

Figure 15: Installing the Library

4. Choose Version 1.3.0 of the library, as shown in **Figure 16**

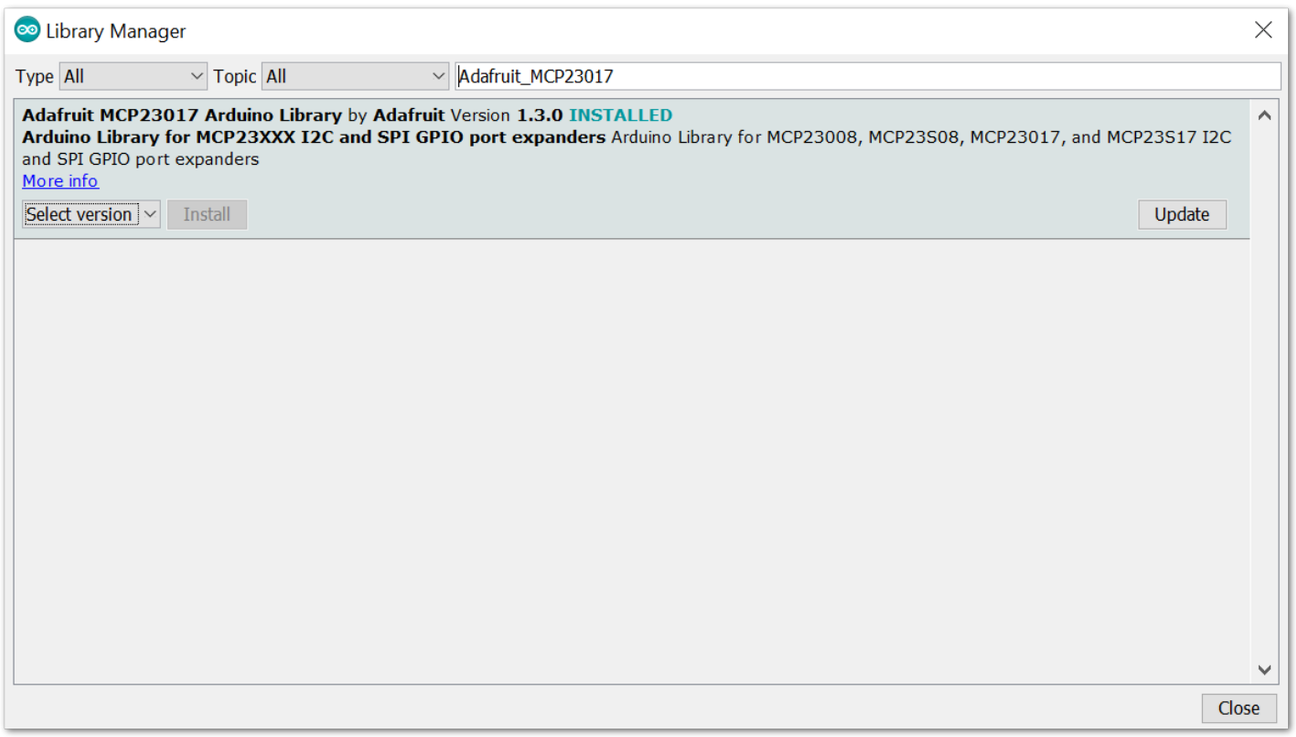

Figure 16: Selecting Version 1.3.0

5. Select the correct port and upload your code, as shown in **Figure 17** and **Figure 18**.

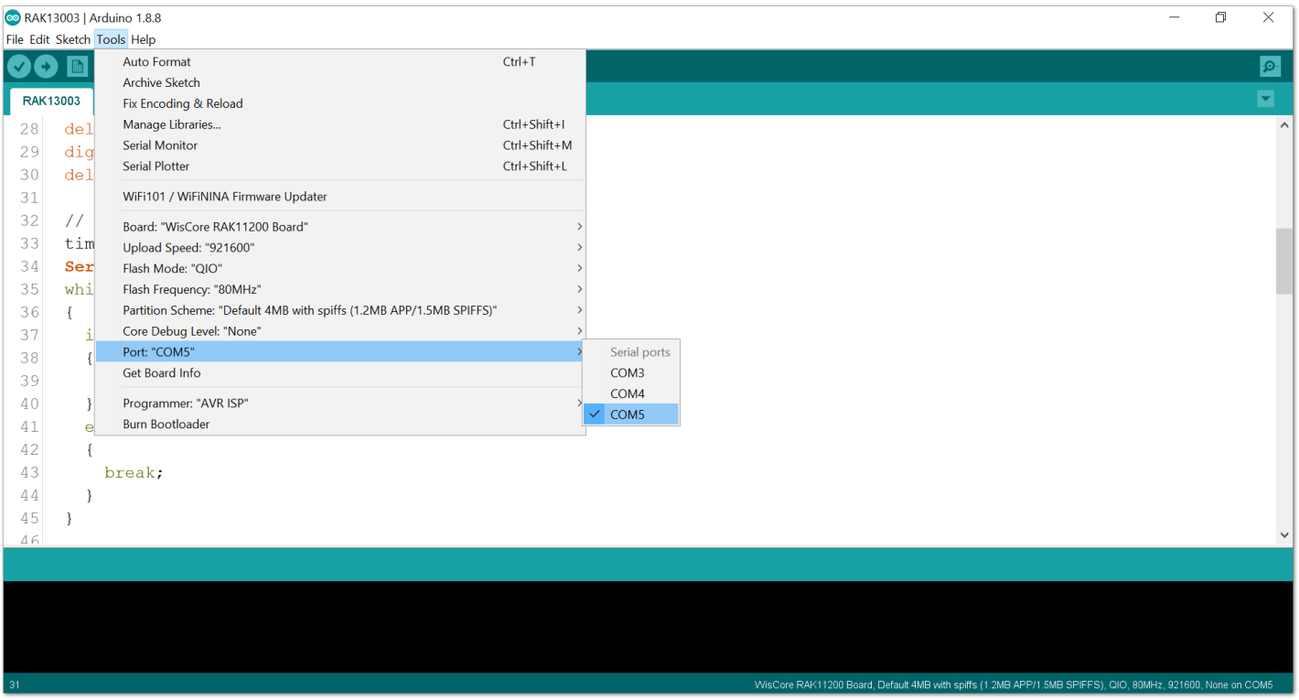

Figure 17: Selecting the correct Serial Port

Figure 18: Uploading code

> 📝 **NOTE:**
>
> RAK11200 requires the BOOT0 pin to be configured properly before uploading. If not done properly, uploading the source code to RAK11200 will fail. Check the full details on the RAK11200 Quick Start Guide.

6. When you have successfully uploaded the example sketch, you can see that the LEDs are powered ON. You can also switch PB as INPUT and PA as OUTPUT by changing this line of code shown in **Figure 19**.

```
RAK13003 §
 3    @author rakwireless.com
 4    @brief Use IIC to expand 16 GPIO.
 5           Configure PA input PB output, or PA output PB input.Serial port print GPIO status.
 6    @version 0.1
 7    @date 2021-2-24
 8    @copyright Copyright (c) 2021
 9  **/
10  #include <Wire.h>
11  #include <Adafruit_MCP23017.h>  //http://librarymanager/All#Adafruit_MCP23017
12
13  //#define PAIN_PBOUT  //PB is set as output here and PA as input.
14  #define PAOUT_PBIN  //PA is set as output here and PB as input.
15
16  Adafruit_MCP23017 mcp;
17
18  void setup()
```

Figure 19: Switching between PA and PB

> 📝 **NOTE**
>
> You can use `mcp.digitalWrite(pin_no,state)` and `mcp.digitalRead(pin_no)` to send or read states.

RAK13003 in RAK11300 WisBlock Core Guide

Arduino Setup

1. First, you need to select the RAK11300 WisBlock Core as shown in **Figure 20**.

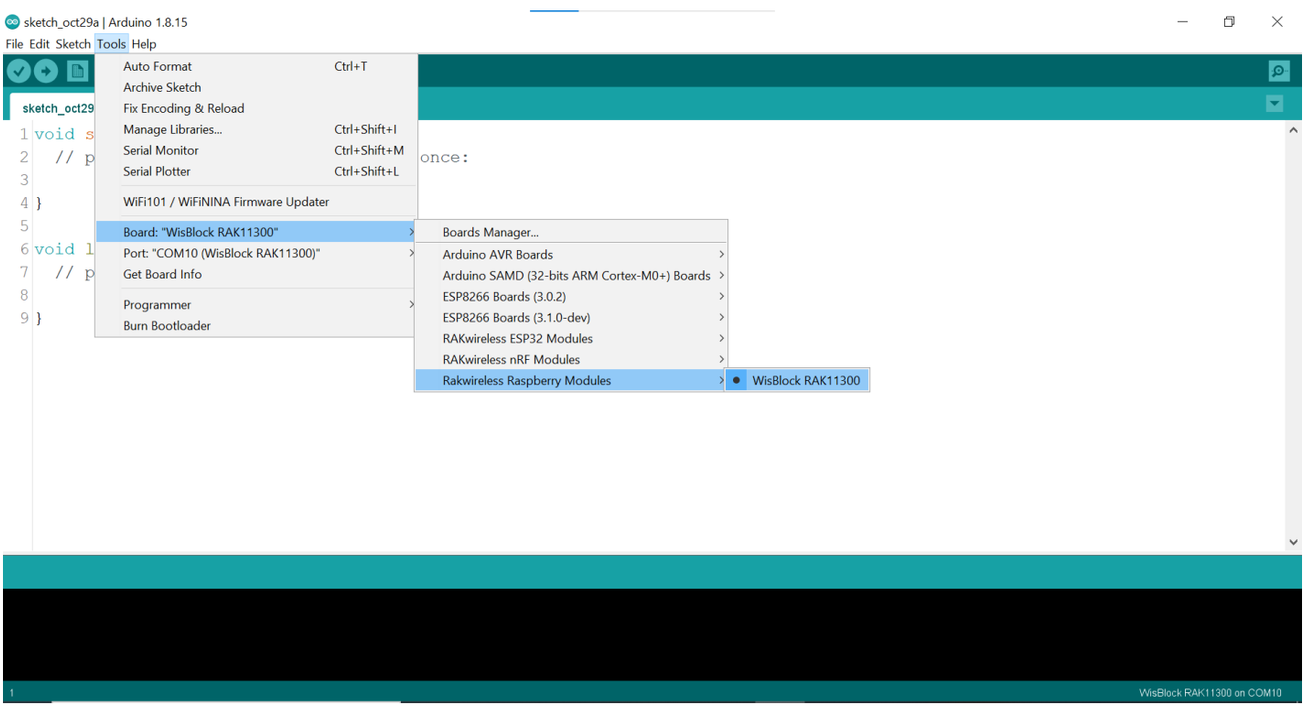

Figure 20: Selecting RAK11300 as WisBlock Core

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

```c
/**
   @file RAK13003_GPIO_Expander_IO_MCP32.ino
   @author rakwireless.com
   @brief Use IIC to expand 16 GPIO.
          Configure PA input PB output, or PA output PB input.Serial port print GPIO status.
   @version 0.1
   @date 2021-2-24
   @copyright Copyright (c) 2021
**/
#include <Wire.h>
#include <Adafruit_MCP23017.h>  //click here to get the library: http://librarymanager/All#Adafru

#define PAIN_PBOUT  //PB is set as output here and PA as input.
//#define PAOUT_PBIN

Adafruit_MCP23017 mcp;

void setup()
{
  pinMode(WB_IO2, OUTPUT);
  digitalWrite(WB_IO2, 1);

  // Reset device
  pinMode(WB_IO4, OUTPUT);
  digitalWrite(WB_IO4, 1);
  delay(10);
  digitalWrite(WB_IO4, 0);
  delay(10);
  digitalWrite(WB_IO4, 1);
  delay(10);

  // Initialize Serial for debug output
  time_t timeout = millis();
  Serial.begin(115200);
  while (!Serial)
  {
    if ((millis() - timeout) < 5000)
    {
      delay(100);
    }
    else
    {
      break;
    }
  }

  Serial.println("MCP23017 GPIO Input Output Test.");

  mcp.begin(); // use default address 0.

#ifdef PAIN_PBOUT
  for(int i=0 ;i < 8 ;i++)
  {
    mcp.pinMode(i, INPUT);  // PA input.
  }
  for(int j=8 ;j < 16 ;j++)
  {
    mcp.pinMode(j, OUTPUT); // PB output.
  }
  mcp.digitalWrite(8, LOW); // The output state of the PB port can be changed to high or low leve
  mcp.digitalWrite(9, HIGH); //PIN PB1
  mcp.digitalWrite(10, LOW); //PIN PB2
```

```
  mcp.digitalWrite(11, LOW); //PIN PB3

  mcp.digitalWrite(12, LOW); //PIN PB4
  mcp.digitalWrite(13, LOW); //PIN PB5
  mcp.digitalWrite(14, LOW); //PIN PB6
  mcp.digitalWrite(15, HIGH);//PIN PB7

  Serial.println();
  for(int i=0; i < 8; i++ )
  {
    if(mcp.digitalRead(i) == 1)
      Serial.printf("GPIO A %d Read High\r\n",i);
    else
      Serial.printf("GPIO A %d Read Low\r\n",i);
  }
#endif

#ifdef PAOUT_PBIN
  for(int i=0 ;i < 8 ;i++)
  {
    mcp.pinMode(i, OUTPUT); // PA output.
  }
  for(int j=8 ;j < 16 ;j++)
  {
    mcp.pinMode(j, INPUT);  // PB input.
  }
  mcp.digitalWrite(0, LOW); // The output state of the PA port can be changed to high or low leve
  mcp.digitalWrite(1, HIGH);
  mcp.digitalWrite(2, LOW);
  mcp.digitalWrite(3, HIGH);

  mcp.digitalWrite(4, LOW);
  mcp.digitalWrite(5, HIGH);
  mcp.digitalWrite(6, LOW);
  mcp.digitalWrite(7, HIGH);
  Serial.println();
  for(int i=8; i < 16; i++ )
  {
    if(mcp.digitalRead(i) == 1)
      Serial.printf("GPIO B %d Read High\r\n",i-8);
    else
      Serial.printf("GPIO B %d Read Low\r\n",i-8);
  }
#endif
}
void loop()
{

}
```

> 📝 **NOTE**
>
> If you experience any error in compiling the example sketch, check the updated code for the RAK11300 WisBlock Core Module that can be found on the RAK13003 WisBlock Example Code Repository .

3. Install the required library, as shown in **Figure 21**.

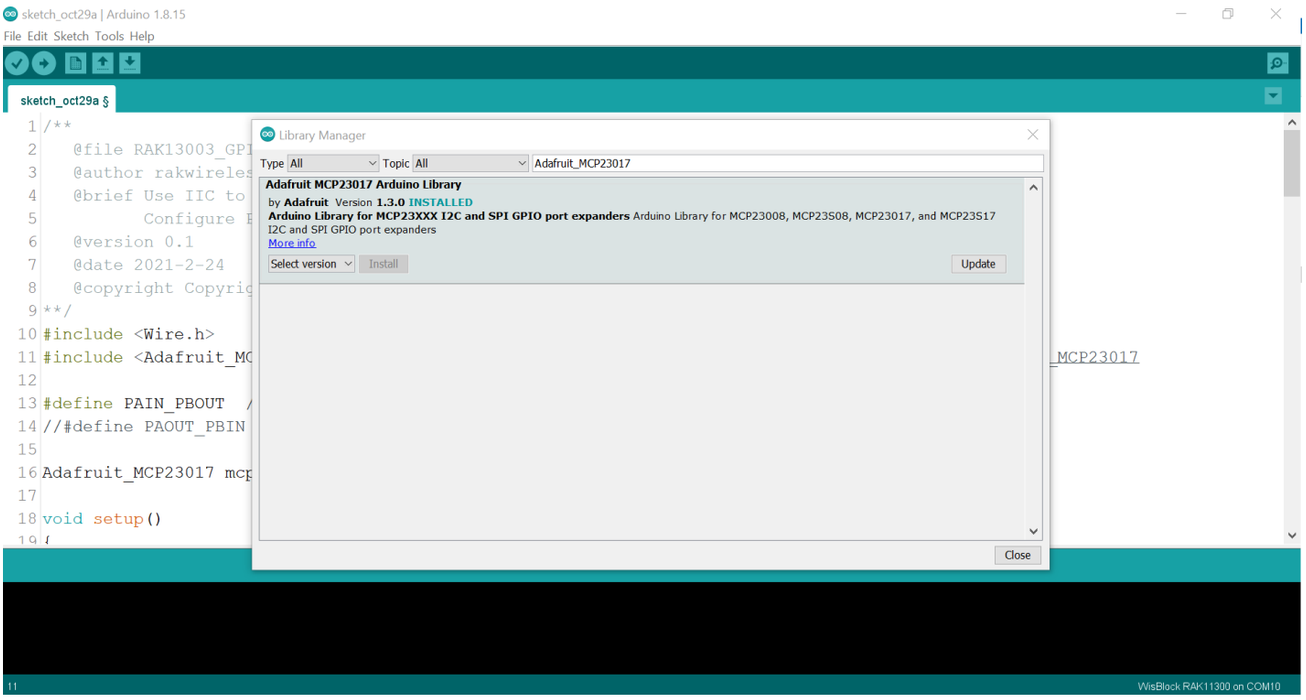

Figure 21: Installing the Library

4. Choose Version 1.3.0 of the library, as shown in **Figure 22**

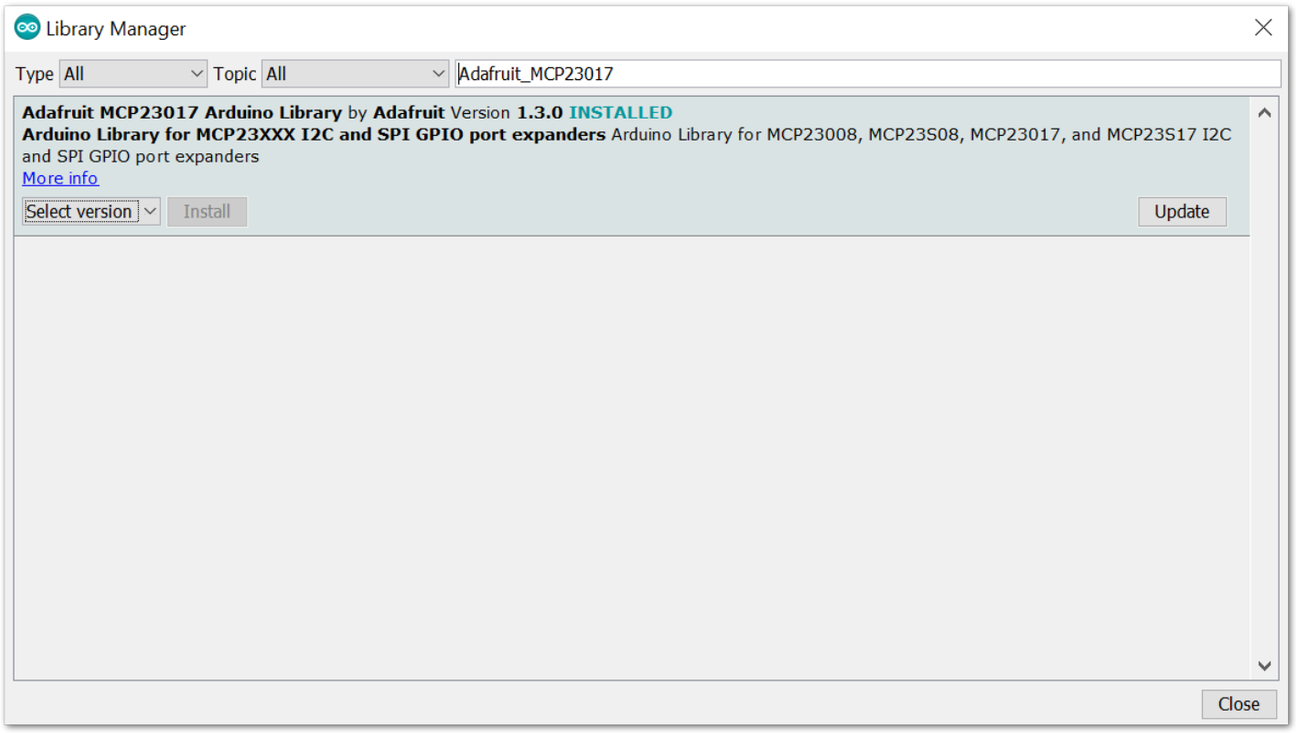

Figure 22: Selecting Version 1.3.0

5. Select the correct port and upload your code, as shown in **Figure 23** and **Figure 24**.

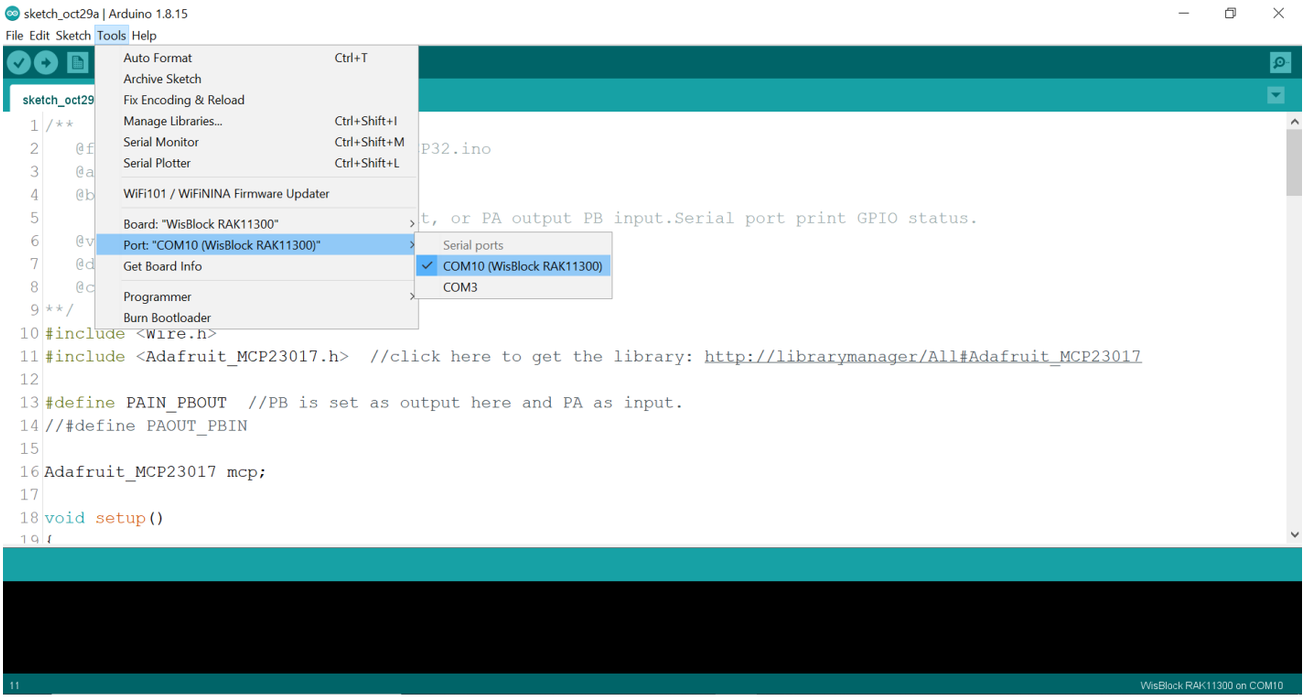

Figure 23: Selecting the correct Serial Port

Figure 24: Uploading code

6. When you have successfully uploaded the example sketch, you can see that the LEDs are powered ON. You can also switch PB as INPUT and PA as OUTPUT by changing this line of code shown in **Figure 25**.

RAK13003 §

```
   @author rakwireless.com
   @brief Use IIC to expand 16 GPIO.
          Configure PA input PB output, or PA output PB input.Serial port print GPIO status.
   @version 0.1
   @date 2021-2-24
   @copyright Copyright (c) 2021
**/
#include <Wire.h>
#include <Adafruit_MCP23017.h>  //http://librarymanager/All#Adafruit_MCP23017

//#define PAIN_PBOUT  //PB is set as output here and PA as input.
#define PAOUT_PBIN  //PA is set as output here and PB as input.

Adafruit_MCP23017 mcp;

void setup()
```

Figure 25: Switching between PA and PB

> 📝 **NOTE**
>
> You can use `mcp.digitalWrite(pin_no,state)` and `mcp.digitalRead(pin_no)` to send or read states.

Last Updated: 11/4/2021, 5:44:19 AM