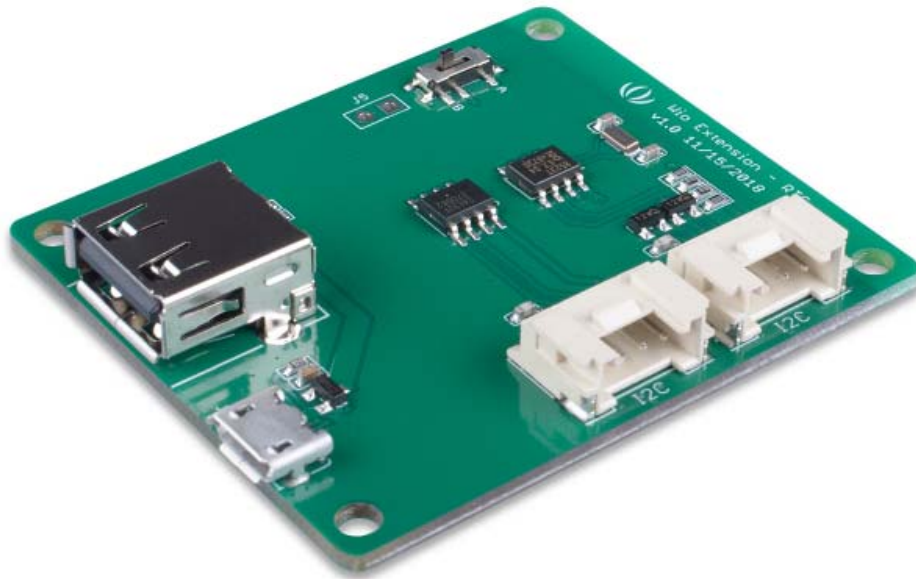# Wio Extension RTC

The Wio Extension - RTC is an extension board for Wio LTE, it can provide the Real-Time Clock function via the I2C port. This board is based on NXP PCF8523 chip, which can provide year, month, day, weekday, hours, minutes, and seconds information.

This board is powered by Micro-USB port, communicate with the Wio LTE via I2C port, and we also provide a USB power output which can be turned off/on by a on-board switch, so that you can use the Wio Extension - RTC board to power the Wio LTE. When the power supply to Wio boards (Like the following picture), the standby current of whole system is less than 1 uA.
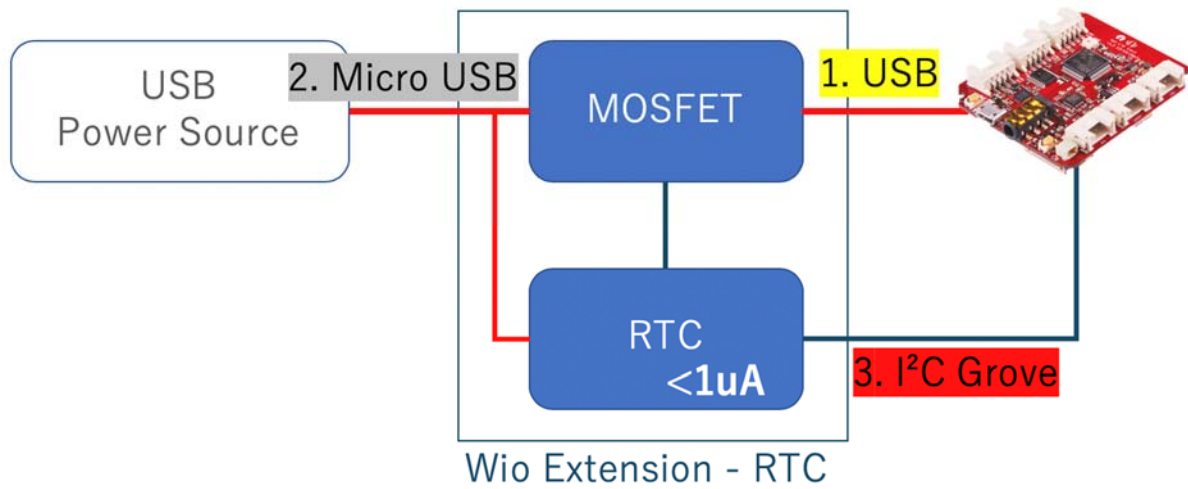
Feature

- Extensibility
- Able to supply Wio boards with 3.3 voltage.

Hardware Overview
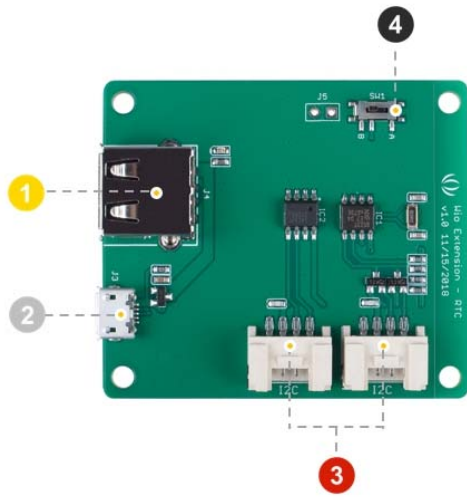
4 **Switch**: manually turn on/off the external output power
3 **I2C Grove Port**: access to RTC and EEPROM via I2C
2 **Micro USB**: Powering the module
1 **USB**: Provide 5V power for external output

USB Power Source — 2. Micro USB → MOSFET → 1. USB

RTC <1uA — 3. I²C Grove
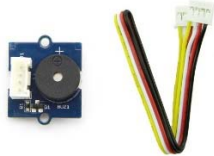
Wio Extension - RTC

Platforms Supported

Arduino	Raspberry Pi	BeagleBone	Wio	LinkIt ONE
	TO DO	TO DO		TO DO

Getting Started

Play With Arduino

Materials required

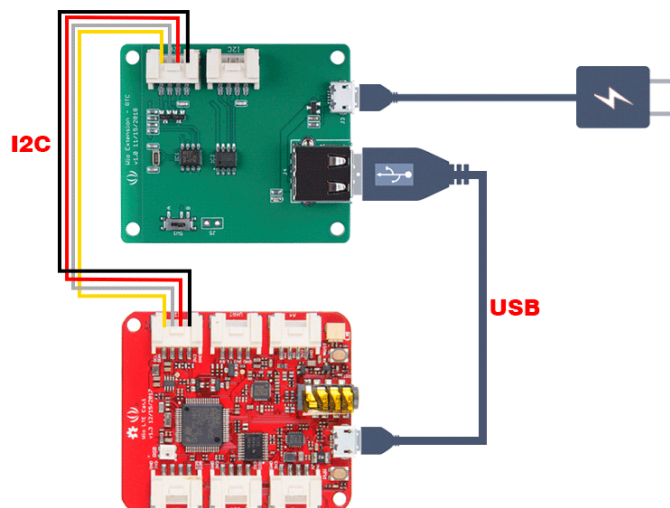| Wio LTE Boards	Wio-Extension-RTC	Grove - Buzzer	Grove - Red LED

Since Wio Extension - RTC just controlling USB power supply set from I2C, you can use this board to manage the power supply almost for every MCU boards powering from USB.

> **Note**
> **1** Please plug the USB cable gently, otherwise you may damage the port. Please use the USB cable with 4 wires inside, the 2 wires cable can't transfer data. If you are not sure about the wire you have, you can click here to buy
> **2** Each Grove module comes with a Grove cable when you buy. In case you lose the Grove cable, you can click here to buy.

Hardware

- **Step 1.** Connect the Wio-Extension-RTC to the **I2C** port of the Wio LTE Boards.
- **Step 2.** Connect Wio LTE Boards.to PC via a USB cable.
- **Step 3.** Connect Grove - Buzzer or Grove - Red LED to D38 of Wio LTE.

Attention

If this is the first time you work with Arduino, we strongly recommend you to see Getting Started with Arduinobefore the start.The driver of this board is rely on the head file of `Seeed STM32F4 Board(JP mirror) by Seeed K.K.` , so whether you have installed your wio board with the tutorial of Getting Started with Arduino, you need to do the following step.

Step 1 Install library Open your Arudino IDE, click on File > Preferences, and copy below url to Additional Boards Manager URLs.
`http://www.seeed.co.jp/package_SeeedJP_index.json`
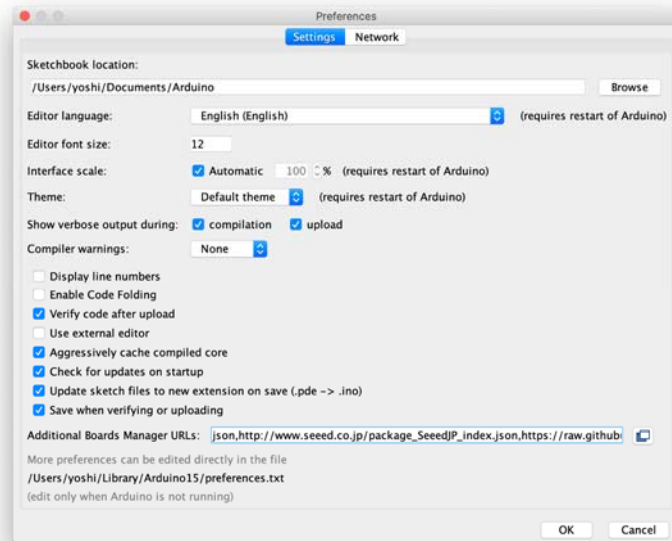
Click on Toos > Board > Board Manager, and enter `Wio` to the text box.
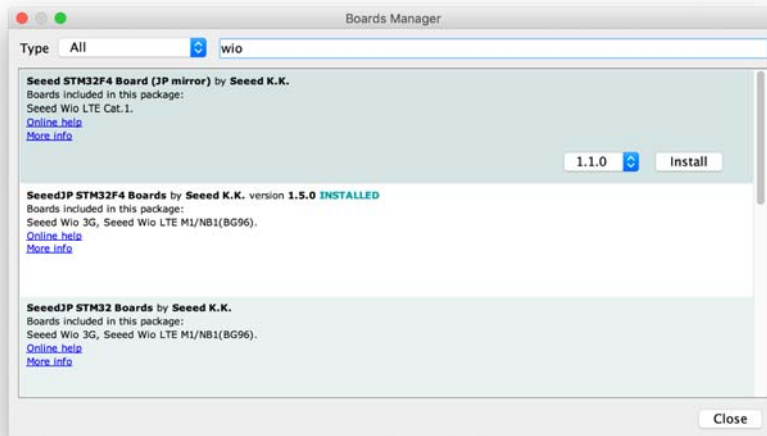
Click `Seeed STM32F4 Board(JP mirror) by Seeed K.K.` then an Install button appear, click on it to finish the step, this process takes about 5 minutes to half an hour, which depend on the speed of your network. Click on Tools > Manage Libraries, and enter `Wio` to the text box.
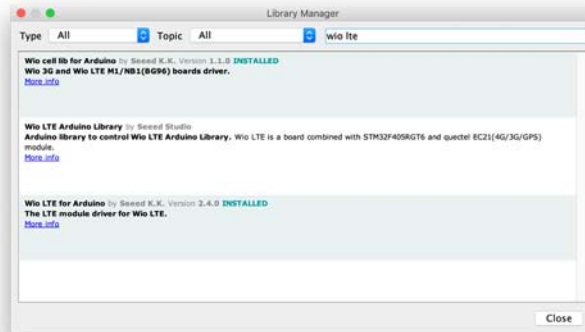
Click `Wio LTE for Arduino by Seeed K.K.` then an Install button appear, click on it to finish the step.

Unzip the sample sketch, and open `wiortc-sample.ino` with Arduino IDE.

Step2 Download the code

1. Press and hold BOOT button at back side of the Wio LTE and plug the USB to PC.
2. We will see STM BOOTLARDER in device manager.
3. Select Tools→Boards→Wio_Tracker_LTE.

4. Select Sketch→Upload to upload the code to Wio_LTE.

5. Press RST button to enable the COM port. **Tips**
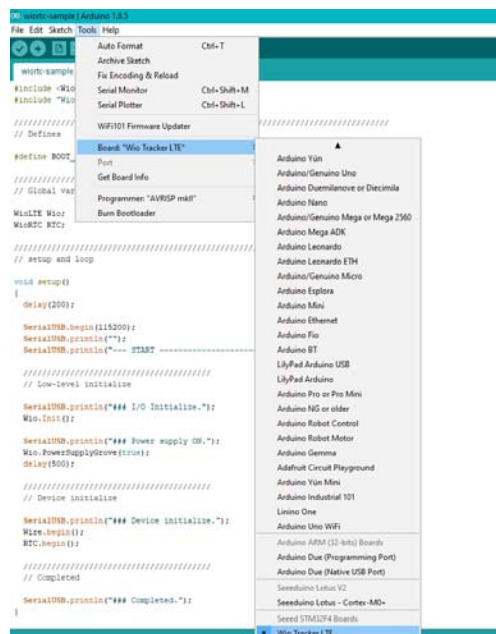
   When you download most Arduino bords, you need to choose a right COM port, but for this board, you must keep the COM configuration to be blank.
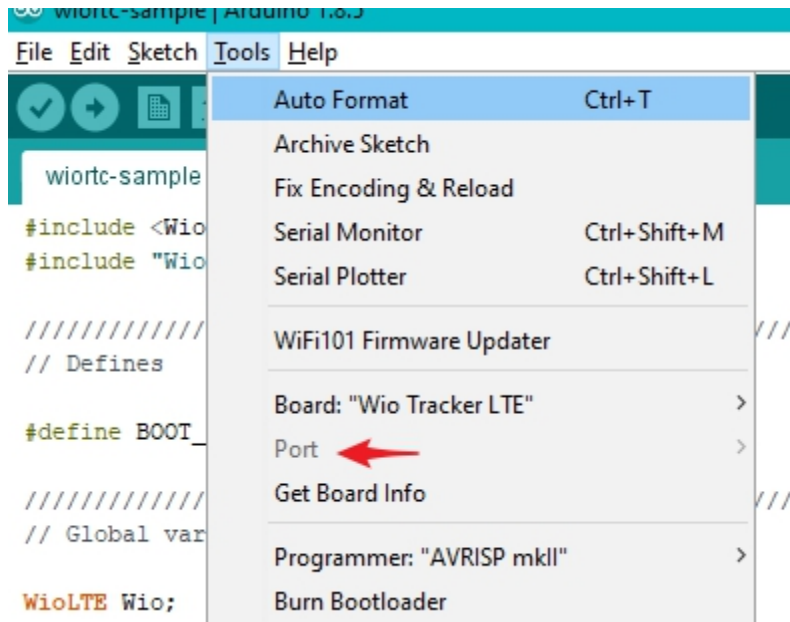
1. Use Serial monitor to print the serial message.

```
 #include <WioLTEforArduino.h>
1#include "WioRTC.h"

2/////////////////////////////////////////////////////////////////////////////
 ////
3// Defines

4#define BOOT_INTERVAL   (30)  // [sec.]

5/////////////////////////////////////////////////////////////////////////////
 ////
6// Global variables

7WioLTE Wio;
 WioRTC RTC;
8
 /////////////////////////////////////////////////////////////////////////////
9////
1// setup and loop
0
1void setup()
1{
1   delay(200);
2
    SerialUSB.begin(115200);
```

```cpp
  SerialUSB.println("");
  SerialUSB.println("--- START -----------------------------------------------");

  //////////////////////////////////////
  // Low-level initialize

  SerialUSB.println("### I/O Initialize.");
  Wio.Init();

  SerialUSB.println("### Power supply ON.");
  Wio.PowerSupplyGrove(true);
  delay(500);

  //////////////////////////////////////
  // Device initialize

  SerialUSB.println("### Device initialize.");
  Wire.begin();
  RTC.begin();

  //////////////////////////////////////
  // Completed

  SerialUSB.println("### Completed.");
}

void loop()
{
  uint8_t val;
  RTC.EepromRead(0, &val, sizeof(val));
  SerialUSB.print("EEPROM value is ");
  SerialUSB.println(val);

  val++;
  RTC.EepromWrite(0, &val, sizeof(val));

  SerialUSB.println("Beep.");
  pinMode(WIO_D38, OUTPUT);
  digitalWrite(WIO_D38, HIGH);
  delay(200);
  digitalWrite(WIO_D38, LOW);

  SerialUSB.println("Shutdown.");
  RTC.SetWakeupPeriod(BOOT_INTERVAL);
  RTC.Shutdown();
  while (1) {}
}
////////////////////////////////////////////////////////////////////////////
////
```

4
1
4
2
4
3
4
4
4
5
4
6
4
7
4
8
4
9
5
0
5
1
5
2
5
3
5
4
5
5
5
6
5
7
5
8
5
9
6
0
6
1
6
2
6
3
6
4
6
5
6
6
6
7
6
8

```
6
9
7
0
7
1
```

After you download the sample sketch, Wio Extension - RTC set to shut down the system for 30 sec with

```
1RTC.SetWakeupPeriod(BOOT_INTERVAL);
```

and then set to shut down the whole system on

```
1RTC.Shutdown();
```