

## Contents

---

Purpose .....	1
Requirements.....	1
Objectives .....	2
Procedure.....	2
Step 1. Download and Install the Arduino IDE v1.8.5 .....	2
Step 2. Configure the Sketchbook Location, Board Manager URL & Other Preferences.....	3
Step 3. Install “Arduino SAMD” and “SODAQ” Boards Packages .....	4
Step 4. Verify Installation of Required Application Libraries .....	5
Step 5. Connect the ExpLoRer Starter Kit.....	6
Arduino IDE Basics .....	9
Step 6. Explore a Basic Sketch and Blink an LED .....	10
Usage of the Hardware Serial Ports on the ExpLoRer Starter Kit.....	11
Using the Arduino IDE Serial Monitor Window.....	12
Conclusion.....	12

## Purpose

---

To get familiarized with the Arduino Software (IDE).

To be ready to use the ExpLoRer Starter Kit by installing Board support and the library.

## Requirements

---

<b>Development Environment:</b>	Windows 7 PC with 1 free USB port and Internet connectivity
<b>IDE/C Compiler:</b>	N/A
<b>Hardware Tools:</b>	SODAQ ExpLoRer Board (“Rev. 6c”)
<b>Software:</b>	N/A
<b>Lab Sketch Location:</b>	C:\Masters\22080\Sketches
<b>Additional Requirements:</b>	Fresh class files installed in “C:\Masters\22080”

## Objectives

---

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the ExpLoRer Starter Kit. It runs Windows, Mac OS X and Linux. The environment is written in java and based on Processing and another open-source software. Here, we will install and configure the environment to be ready for the Microchip ExpLoRer Starter Kit.

## Procedure

---

**Note:** If you currently have a version of the Arduino IDE installed and want a “clean” installation:

- Un-install the Arduino IDE using the standard Windows program un-installation procedure.
  - This uninstalls all files and deletes “C:\Program Files (x86)\Arduino”
- Delete folder “C:\Users\<UserName>\Documents\Arduino”
  - This could delete any existing sketches and installed user libraries, so be sure to save them first!
- Delete (hidden) folder “C:\Users\<UserName>\AppData\Local\Arduino15”
  - You will have to re-install any board packages and external libraries, so please note which packages and libraries you had installed before deleting this folder.
- Un-install any Arduino USB devices in Device Manager

### Step 1. Download and Install the Arduino IDE v1.8.5

---

- a. Download/Install Arduino IDE v1.8.5 using the Windows Installer from the following URL:  
[https://www.arduino.cc/download\\_handler.php?f=/arduino-1.8.5-windows.exe](https://www.arduino.cc/download_handler.php?f=/arduino-1.8.5-windows.exe)
- b. Launch the IDE.

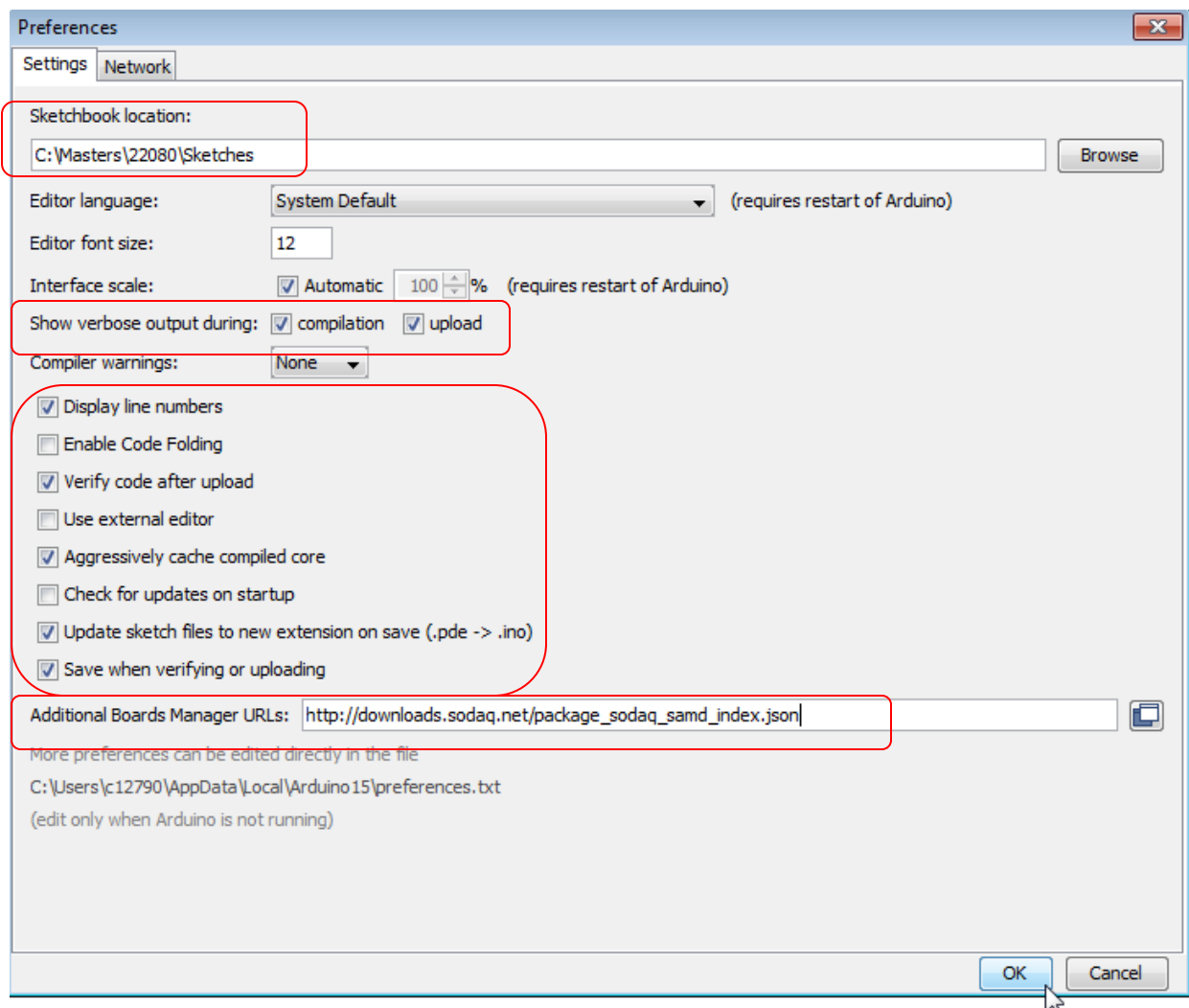
Note that the folder “C:\Users\<UserName>\Documents\Arduino” is created on first start up. This is where all your sketches will be saved, unless you change the default “Sketchbook location” as described in Step 2.

Step 2. Configure the Sketchbook Location, Board Manager URL & Other Preferences

a. Open the menu: **File** → **Preferences**, and set the preferences as indicated below. Note:

The “**Sketchbook location**” is changed to point to where the course-specific sketches are located (“C:\Masters\22080\Sketches”).

The “**Additional Boards Manager URLs**” setting (“http://downloads.sodaq.net/package\_sodaq\_samd\_index.json”) will enable you to install the **SODAQ Board Manager** in a subsequent step. This board manager will install the ExpLoRer board into the Arduino environment.



Select OK. **Close & Restart the IDE for these changes to take effect.**

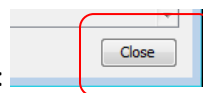
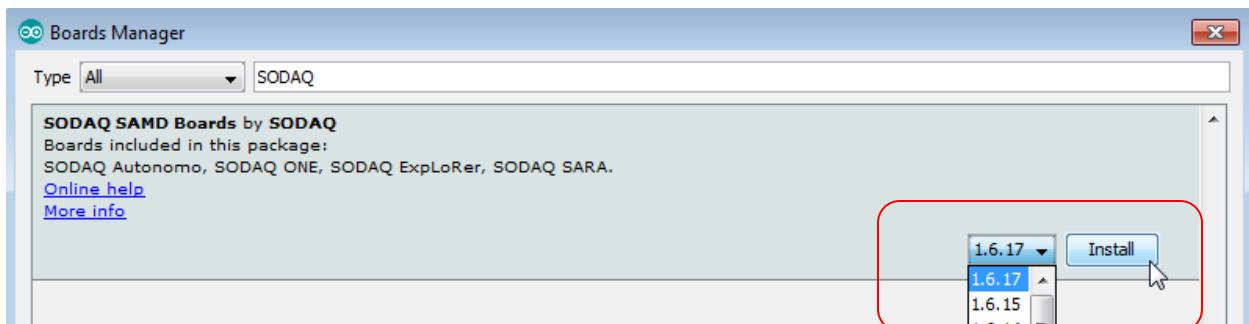
Step 3. Install “Arduino SAMD” and “SODAQ” Boards Packages

The SODAQ ExpLoRer board is based on the Genuino Zero board (Based on the SAM D21 Microcontroller). This requires the installation of Arduino Core libraries & tools for this MCU:

- GCC g++ compiler for the SAM 32-bit MCU core used on the ExpLoRer board
  - USB Drivers for generic SAM-based Arduino and SODAQ ExpLoRer boards
  - Board-specific configuration file for SODAQ ExpLoRer board.
- a. Open the menu **Tools** → **Board: “Arduino/Genuino UNO”** → **Boards Manager**. The “Arduino SAMD Boards” package should appear on the list. Click the “More info” link. From the pull-down, select latest package version. (here this is version **1.6.17**)



- b. Next, find/install the SODAQ SAMD Boards package version **1.6.17**:



When complete. Close the Boards Manager:

## Appendix B: Get Started with Arduino IDE & ExpLoRer

- c. Now, the ExpLoRer board should be visible from the Boards menu. Open the menu **Tools** → **Board** and select the “SODAQ ExpLoRer” as shown:



### Step 4. Verify Installation of Required Application Libraries

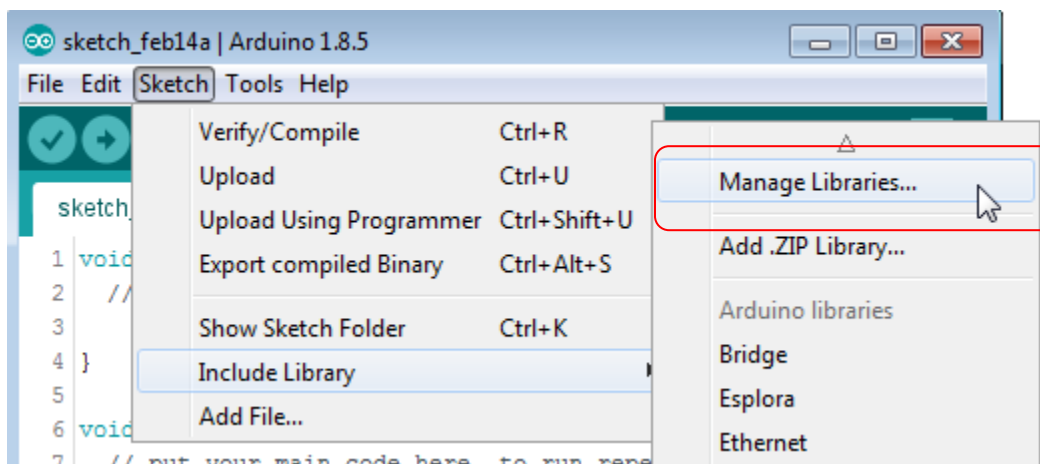
---

To use the features of the RN2xx3 LoRa radio for LoRaWAN connectivity, we need to have The Things Network device library installed and available in our Arduino Environment.

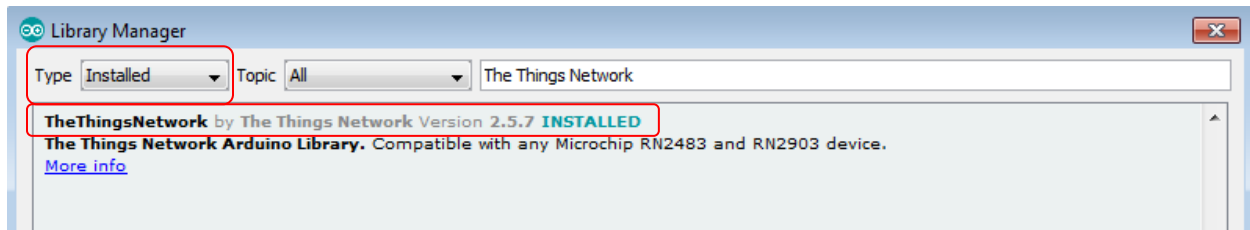
The default Sketchbook folder (“C:\Masters\22080\Sketches”) contains a sub-folder (“Libraries”) with the following library **pre-installed**:

**arduino-device-lib v2.5.7** (APIs for Sending/Receiving Class A payloads on RN2xx3)

- a. Verify the presence of this library in the Arduino Environment by opening the **Library Manager**:  
**Sketch** → **Include Library** → **Manage Libraries...**



Now, search/verify that the library is **installed** as shown here:



If you do not see this library as “installed” on the Library Manager dialog, re-check your Arduino Preferences (Step 2.) and verify that the default sketchbook location matches where you installed the class files, and then restart Arduino IDE.

#### To Update This Library:

The latest .zip file can be downloaded from:

<https://github.com/TheThingsNetwork/arduino-device-lib>

Simply delete the existing “\Sketches\Libraries\arduino-device-lib-master” folder, then extract the .zip file into “\Sketches\Libraries”

When finished, close the Library Manager.

### Step 5. Connect the ExpLoRer Starter Kit

---

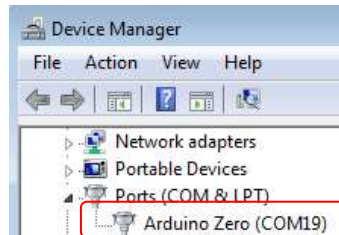
- a. **Connect** the Starter Kit to your computer using the micro USB cable and wait for the driver installation and the COM port mounting. The USB port powers the board and enables the user to communicate with the kit.



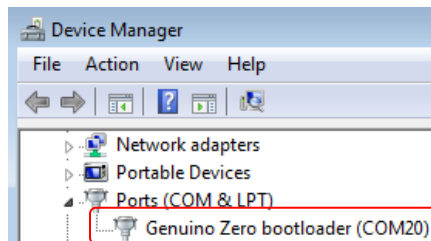
## Appendix B: Get Started with Arduino IDE & ExpLoRer

Note that by default the board is running in **application mode**, and will enumerate a virtual COM port on your PC, for use in communicating with a terminal application.

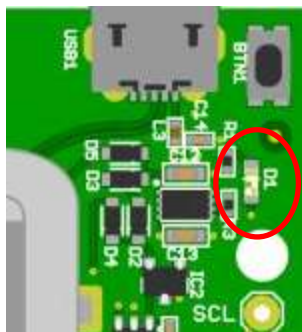
You can always force the board to reset/restart its application by pressing the RESET button (near the USB connector) once:



To program new sketches into the board, it needs to be in **bootloader mode**. To manually enter bootloader mode, press the RESET button twice within 1 second to enter this mode. You should see a second (different) COM port enumerated:



You can confirm bootloader operating mode by the slow fade-in /fade-out of the on-board blue LED:



After uploading a sketch in **bootloader mode**, the board will automatically switch to **application mode** and re-enumerate the application mode's COM port number.

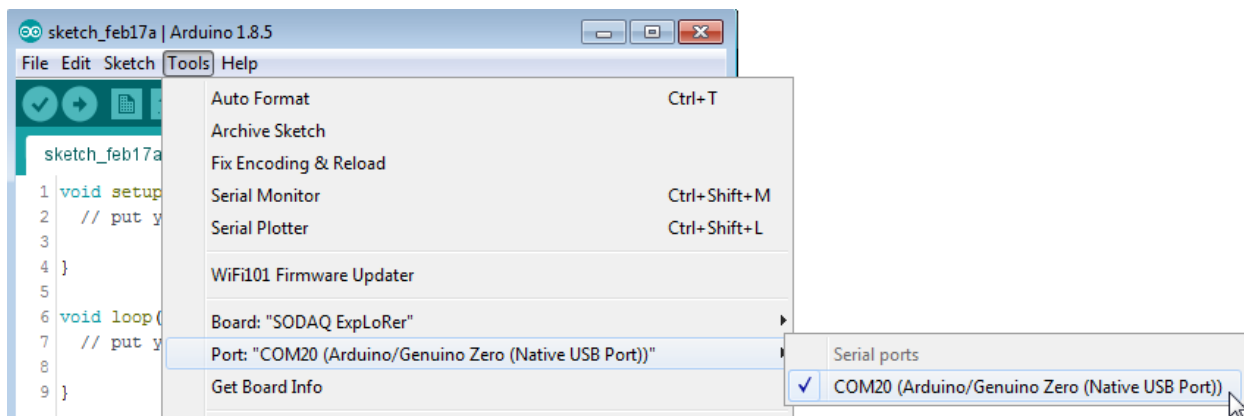
**Note on Uploading Sketches:**

The board firmware is designed to automatically switch into/out of bootloader mode for uploading sketches.

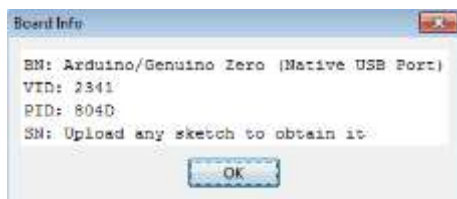
When you initiate an upload procedure, the board should automatically re-enumerate the bootloader mode COM port number, program the sketch, then re-enumerate the application mode COM port number.

If this doesn't work on your PC, you can always force the board into bootloader mode before programming, using the procedure described above.

- b. Place the board in **Bootloader Mode** by pressing the Reset button twice within 1 second. Then note the COM port number displayed in Device Manager (see (ii) above) and set the corresponding COM port within the Arduino IDE: **Tools** → **Port** → **COM**



- c. Check the communication by selecting **Tools** → **Get Board Info**





## Arduino IDE Basics

---

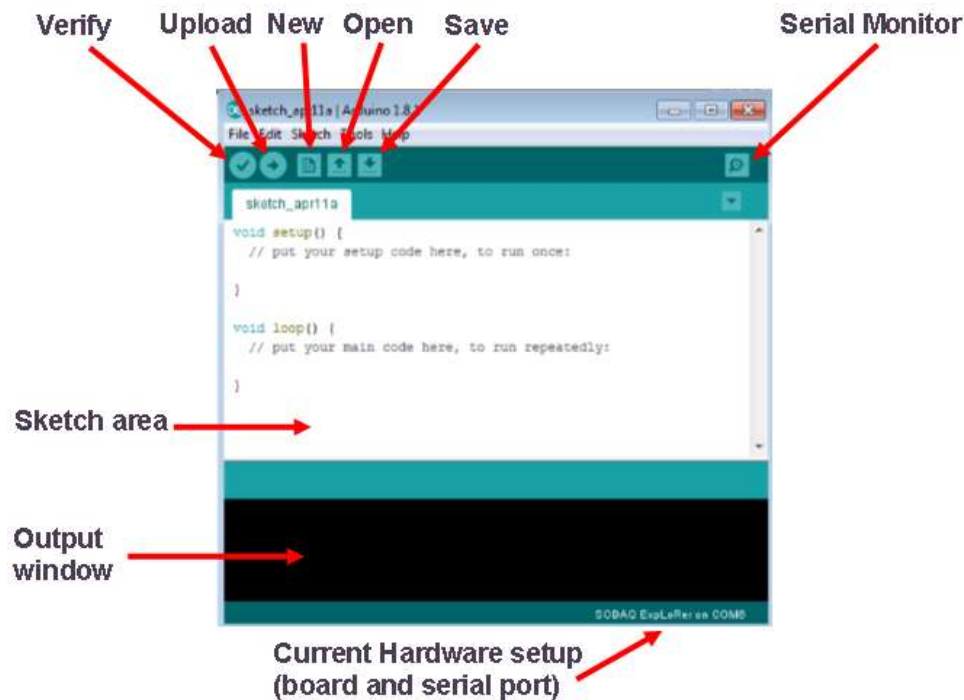
A “**sketch**” is the name that Arduino uses for a program. It is the unit of code that is uploaded to and run on an Arduino board. The programming language used in the Arduino IDE system is C++. Familiarity with C or C++ will, therefore, be very helpful in learning to write sketches using Arduino IDE.

There are two basic functions required by all sketches: `setup()` and `loop()`.

`setup()` is called once at the beginning of execution of the sketch. This function performs any initialization that is needed to set up the execution environment for the sketch. This could include setting pin direction on I/O pins, initializing libraries and so on.

`loop()` is called repeatedly during the execution of the sketch. This is where the body of the sketch is located.

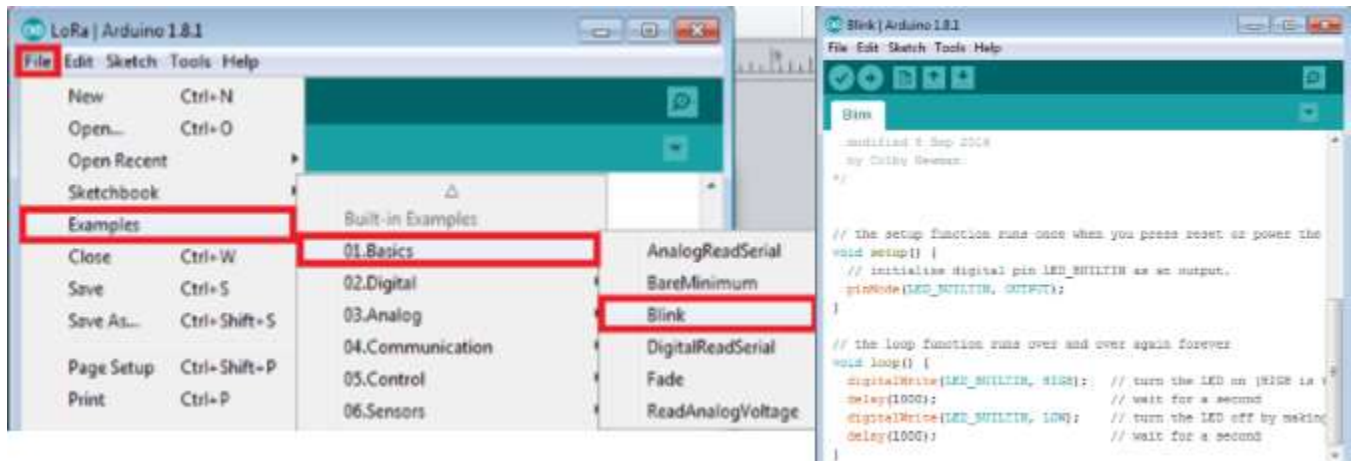
For your reference and if you want to get some knowledge on the Arduino Foundations, you can visit the Arduino Getting Started page <https://www.arduino.cc/en/Guide/HomePage>



Step 6. Explore a Basic Sketch and Blink an LED

The Arduino IDE has some basic examples built-in.

- a. **Open** the basic blink sketch. *File* → *Examples* → *01. Basics* → *Blink*



- b. **Verify/Compile** the basic blink sketch

The first step to getting a sketch ready for transfer over to the Arduino is to Verify/Compile it. That means check it over for mistakes and then translate it into an application that is compatible with the Arduino hardware.

**Verify/Compile**



- c. **Press** the Reset button (near the USB connector) twice within a second to place the ExpLoRer board into **Bootloader Mode** and to mount a new COM port dedicated for uploading the sketch.



- d. **Set** the accompanying COM port number within the Arduino IDE by going to *Tools* → *Port COM*

- e. **Upload** the sketch to the board. The built-in LED (D1) will start to blink every second.



### Usage of the Hardware Serial Ports on the ExpLoRer Starter Kit

The ExpLoRer board has 4 hardware serials defined like this:

- SerialUSB** is for debugging over the USB cable
- Serial** is attached to pin D1/TX and D0/RX of the internal microcontroller
- Serial1** is connected to the RN4871 Bluetooth Low Energy module
- Serial2** is connected to the RN2xx3 LoRaWAN module

The sketch starts directly after uploading new code or when connected to a power source. After opening a Serial Monitor, the code will not reset. The following code may be added to your sketch if you want to wait for a Serial Monitor before pursuing the execution of the program:

```
void setup()
{
  // put your setup code here, to run once:
  // wait for SerialUSB or start after 10 seconds
  while (!(SerialUSB) && (millis() < 10000));
  SerialUSB.begin(57600) ;
  Serial.begin(57600) ;
  Serial1.begin(115200) ;
  Serial2.begin(57600) ;
}

void loop()
{
  // put your main code here, to run repeatedly:
}
```

## Using the Arduino IDE Serial Monitor Window

---



The Arduino IDE has a feature that can be great help in debugging sketches or controlling Arduino from your computer's keyboard. The Serial Monitor is a separate pop-up window that acts as a separate terminal that communicates by receiving and sending Serial data over USB cable in our case. The Serial Monitor can be open by clicking the icon on the far-right side of the bar menu:



Here is an example of a sketch which uses the Serial Monitor for debugging purposes:

```
void setup()
{
  // put your setup code here, to run once:
  // wait for SerialUSB or start after 10 seconds
  while(!SerialUSB || millis() < 10000) ;
  SerialUSB.begin(115200) ; // open a communication @115200bps
  SerialUSB.println("Hello world") ; // print a single line
  for (int myCounter = 0; myCounter < 10; myCounter++)
  {
    SerialUSB.println(myCounter) ; // print myCounter variable
  }
}

void loop()
{
  // put your main code here, to run repeatedly:
}
```

A screenshot of the Serial Monitor window. The window title is "COM5 (SODAQ ExpLoRer)". The output area shows the text "Hello world" followed by a list of numbers from 0 to 9, each on a new line. A green arrow points from the code block on the left to the Serial Monitor window on the right.

## Conclusion

---

This Appendix has familiarized you with the installation and configuration of the Arduino Software (IDE) on a Windows 7 PC, as well as preparing you to use the SODAQ ExpLoRer Starter Kit by installing Board support and the application libraries into the Arduino environment.