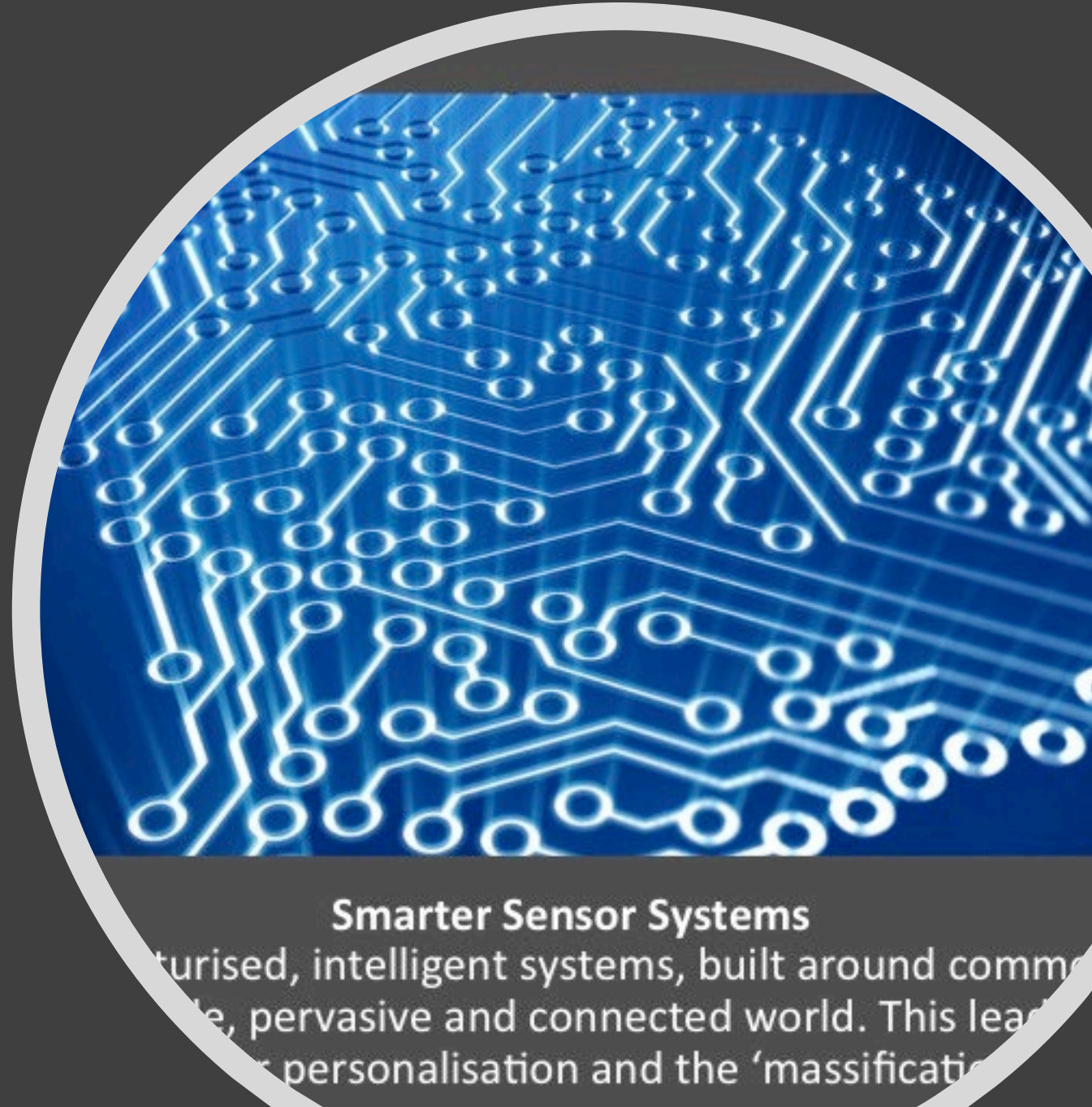


Vincent Claes

# IoT Pilotproject

## SODAQ MBili



### Smarter Sensor Systems

aturised, intelligent systems, built around comm  
e, pervasive and connected world. This lead  
personalisation and the 'massificatio

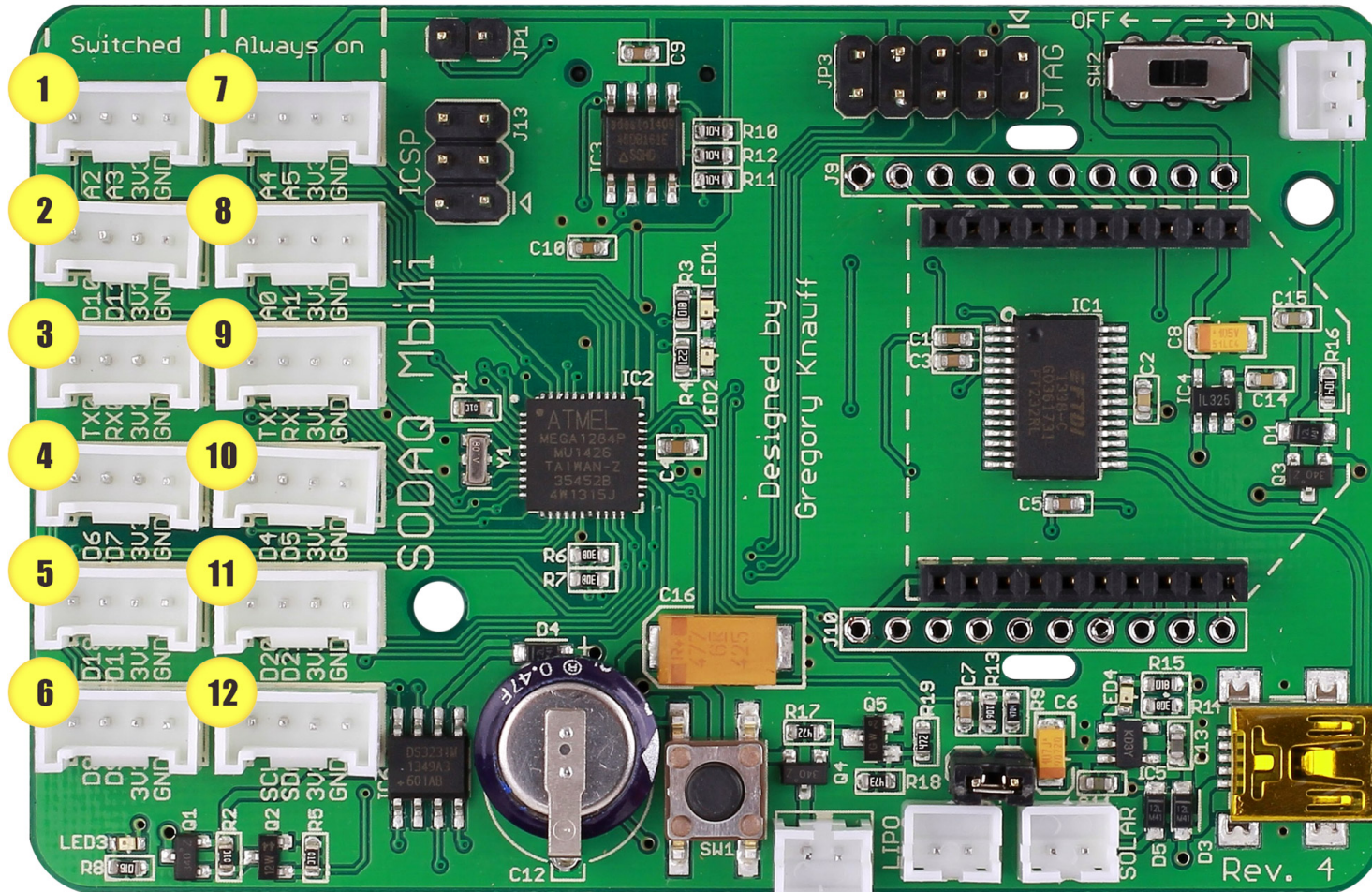
# SODAQ MBili

- Informatie ivm SODAQ Mbili board
- Installatie IDE + Hello World!
- Schrijven van eerste software applicaties

- <https://support.sodaq.com/sodaq-one/sodaq-mbili-1284p/>



# SODAQ Mbili Board



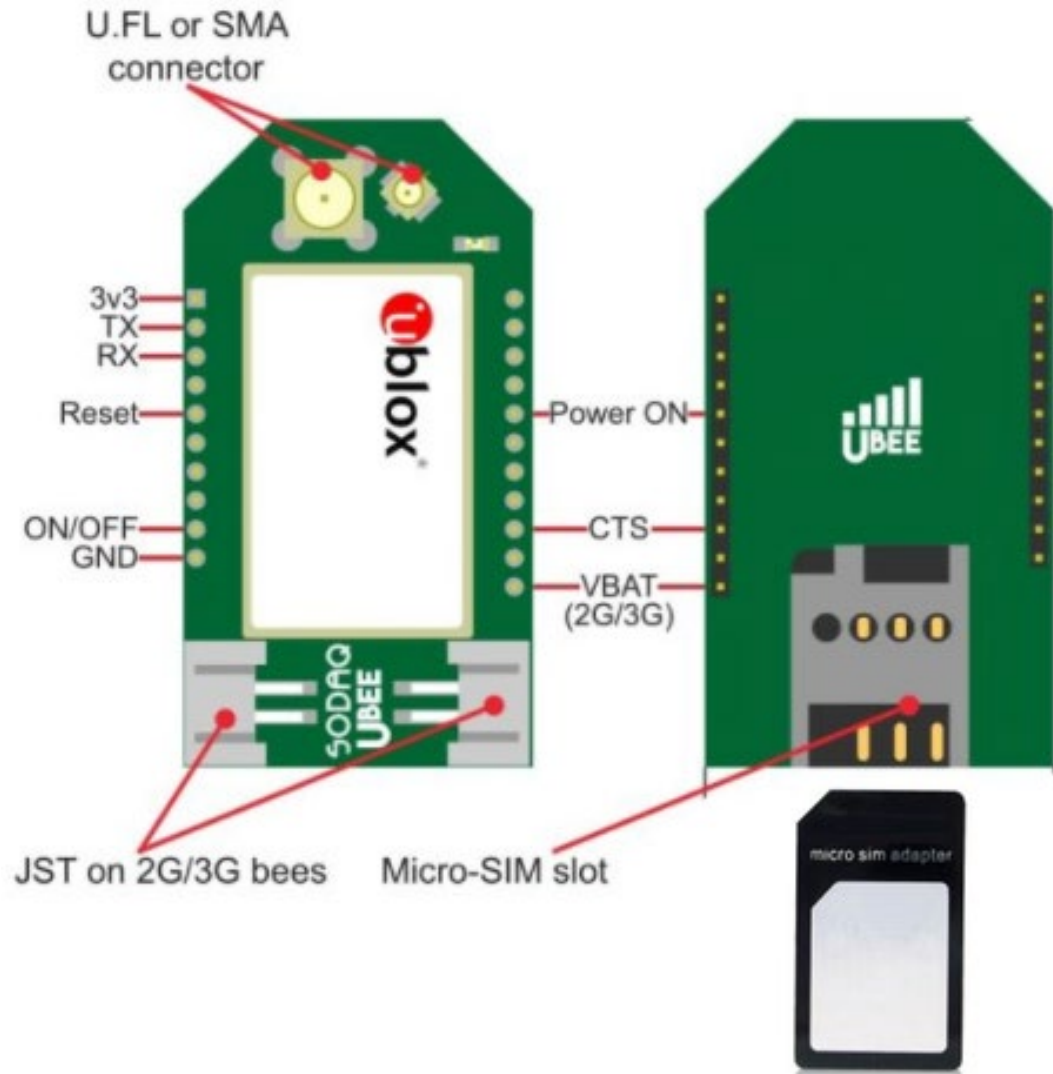
## Switched Column

- 1. Analog: **A2 A3**
- 2. Digital: **D10 D11**
- 3. Serial (USB): **TXD0/RXD0B**
- 4. Digital: **D6 D7**
- 5. Digital: **D18 D19**
- 6. Digital: **D8 D9**

## Always On Column

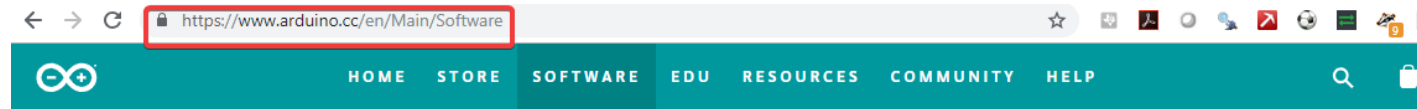
- 7. Analog: **A4 A5**
- 8. Analog: **A0 A1**
- 9. Serial1 (Bee): **TXD1/RXD1**
- 10. Digital: **D4 D5**
- 11. Digital: **D20 D21**
- 12. **I<sup>2</sup>C** socket

# SODAQ UBee




- <https://support.sodaq.com/sodaq-one/ubee/>
- Ublox SARA-N2 – NB-IoT
- Sleep current 4 $\mu$ A

# Arduino IDE Installatie procedure



Download the Arduino IDE



**ARDUINO 1.8.8**  
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.  
This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions.

**Windows** Installer, for Windows XP and up  
**Windows** ZIP file for non admin install

**Windows app** Requires Win 8.1 or 10  
[Get](#)

**Mac OS X** 10.8 Mountain Lion or newer

**Linux** 32 bits  
**Linux** 64 bits  
**Linux** ARM

[Release Notes](#)  
[Source Code](#)  
[Checksums \(sha512\)](#)


**HOURLY BUILDS**

**LAST UPDATE**  
4 February 2019 10:13:19 GMT

Download a **preview of the incoming release** with the most updated features and bugfixes.

[Windows](#)  
[Mac OS X](#) (Mac OSX Mountain Lion or later)  
[Linux](#) 32 bit , [Linux](#) 64 bit , [Linux](#) ARM, [Linux](#) ARM64 (experimental)

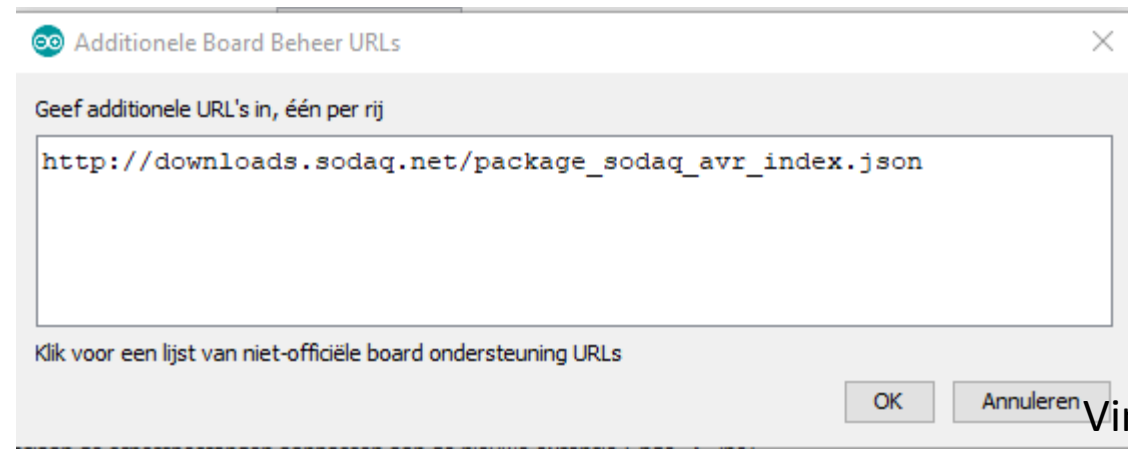
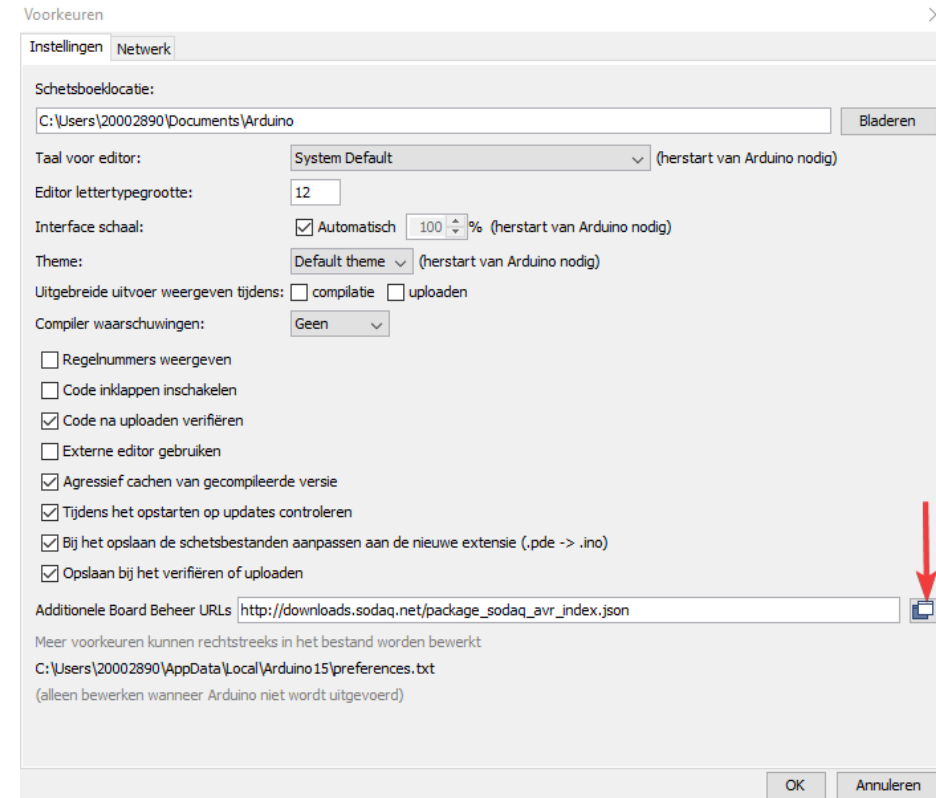
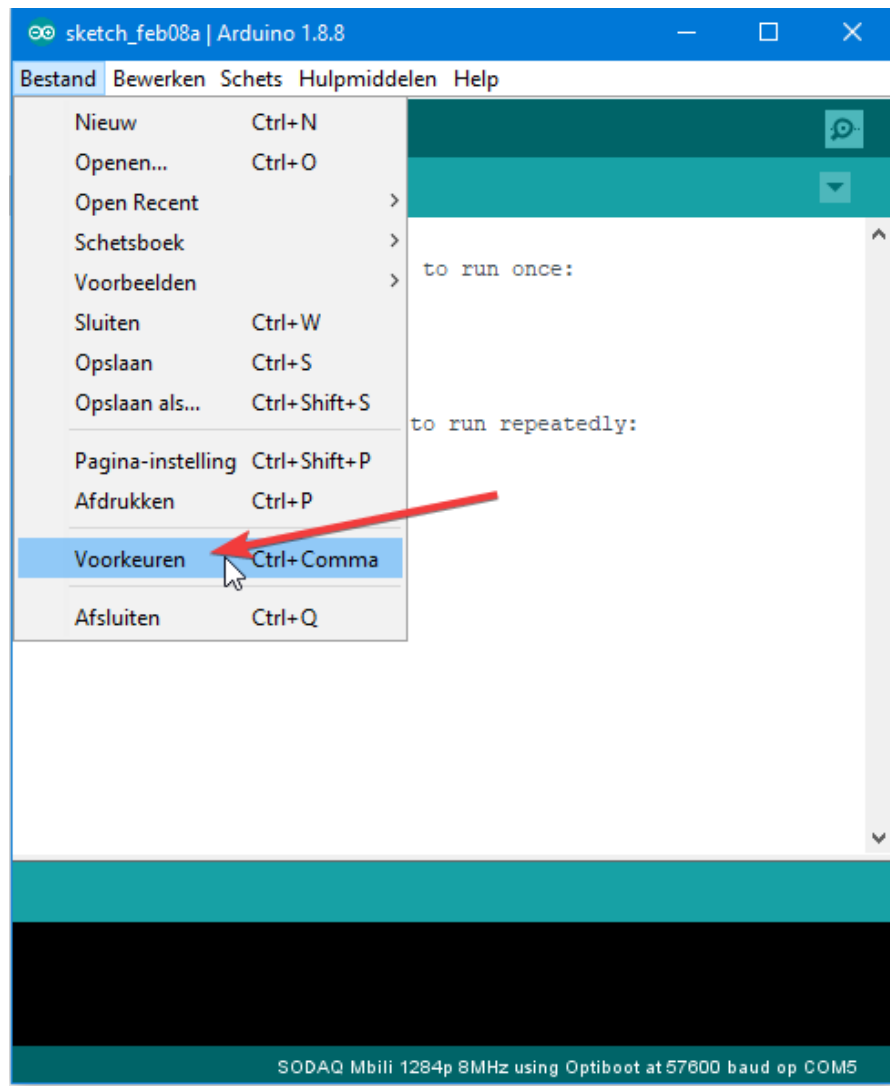
**BETA BUILDS**

 **BETA**

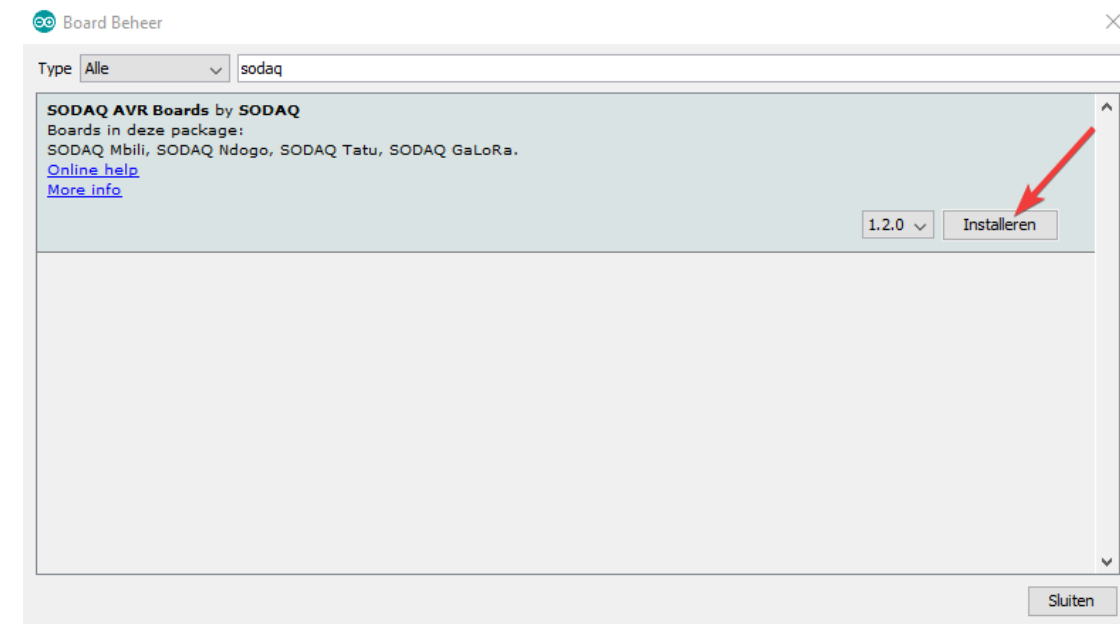
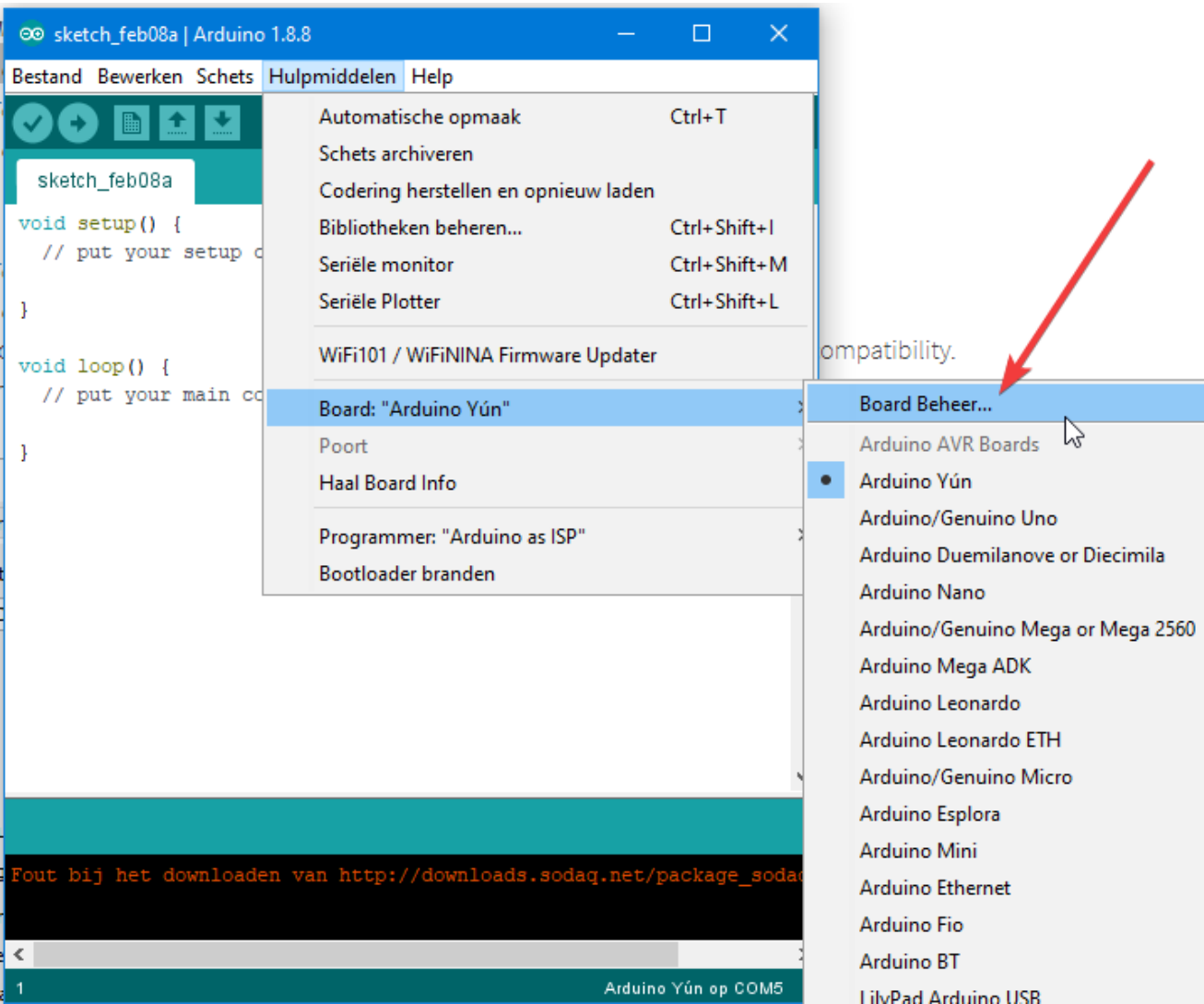
Download the **Beta Version** of the Arduino IDE with experimental features. This version should NOT be used in production.

[Windows](#)  
[Mac OS X](#) (Mac OSX Mountain Lion or later)  
[Linux](#) 32 bit, [Linux](#) 64 bit, [Linux](#) Arm, [Linux](#) Arm64 (experimental)

# Add Mbili Support



# Add Mbili Support

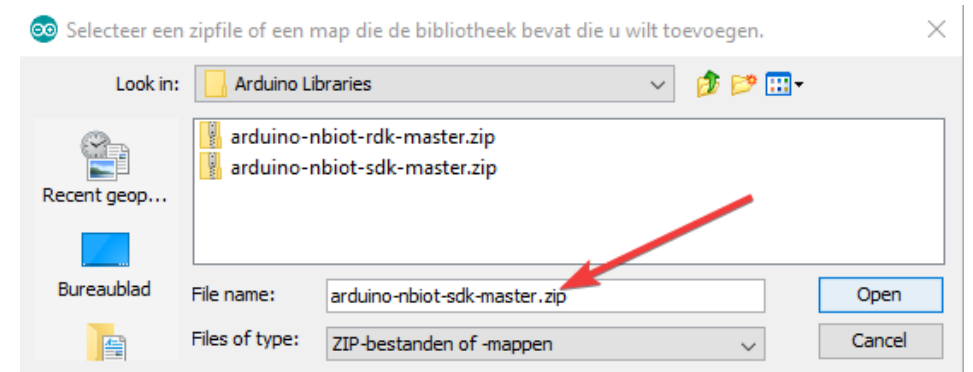
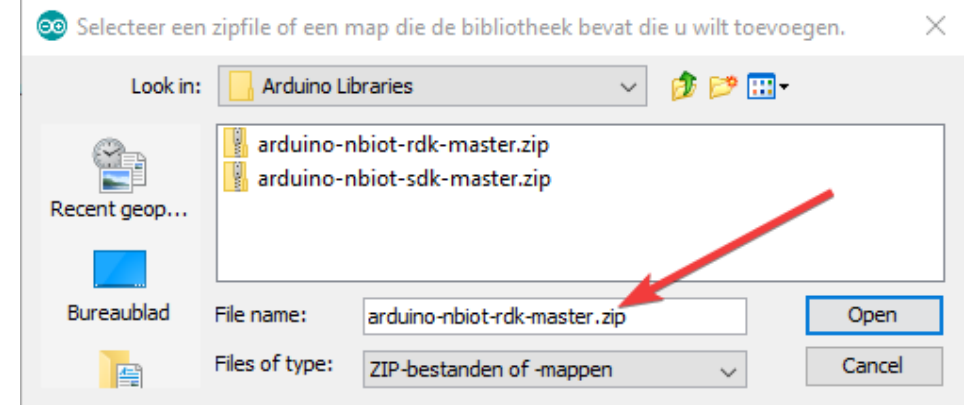
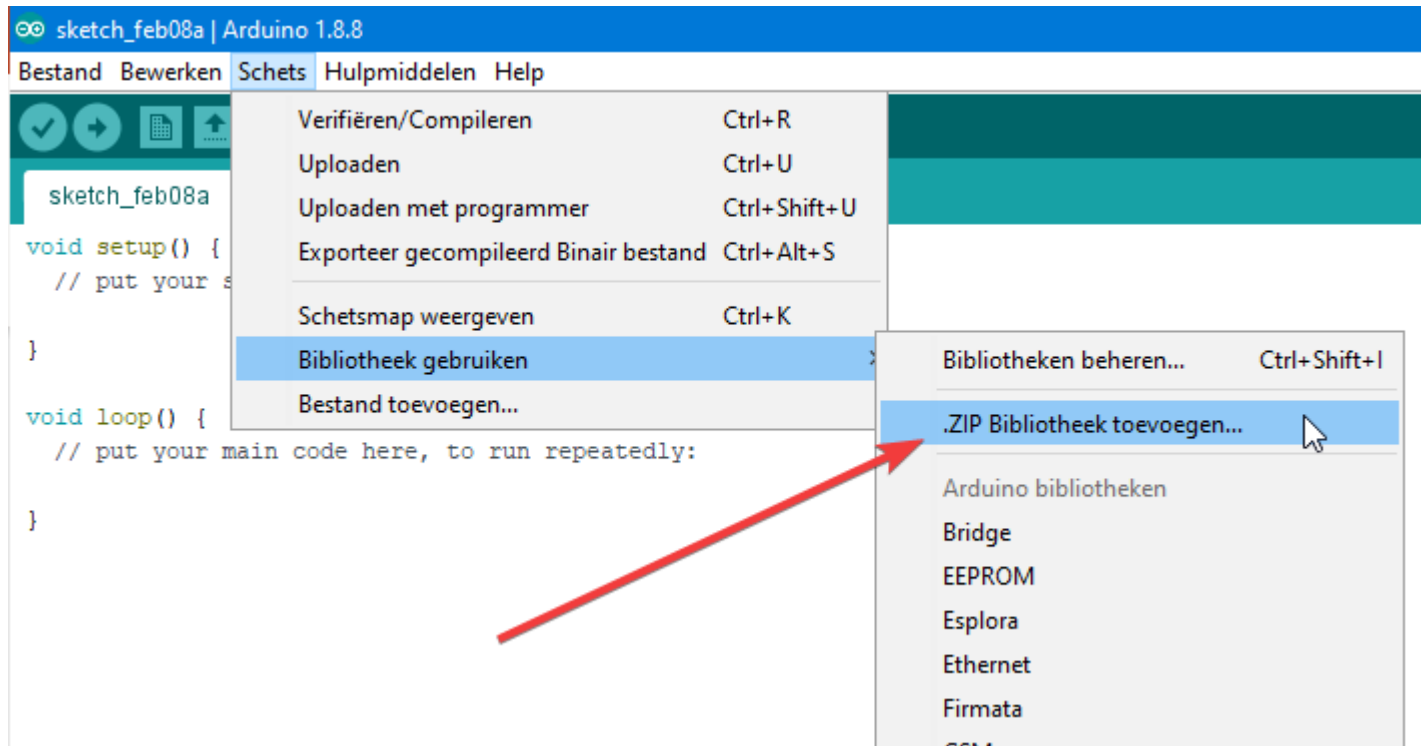




# Add AllThingsTalk Software Libraries to Arduino IDE

- Download to your PC
  - NB-IoT Software Development Kit Library
    - <https://github.com/allthingstalk/arduino-nbiot-sdk/archive/master.zip>
  - NB-IoT Rapid Development Kit Library
    - <https://github.com/allthingstalk/arduino-nbiot-rdk/archive/master.zip>

# Add AllThingsTalk Software Libraries to Arduino IDE



# Example\_01\_Blink.ino

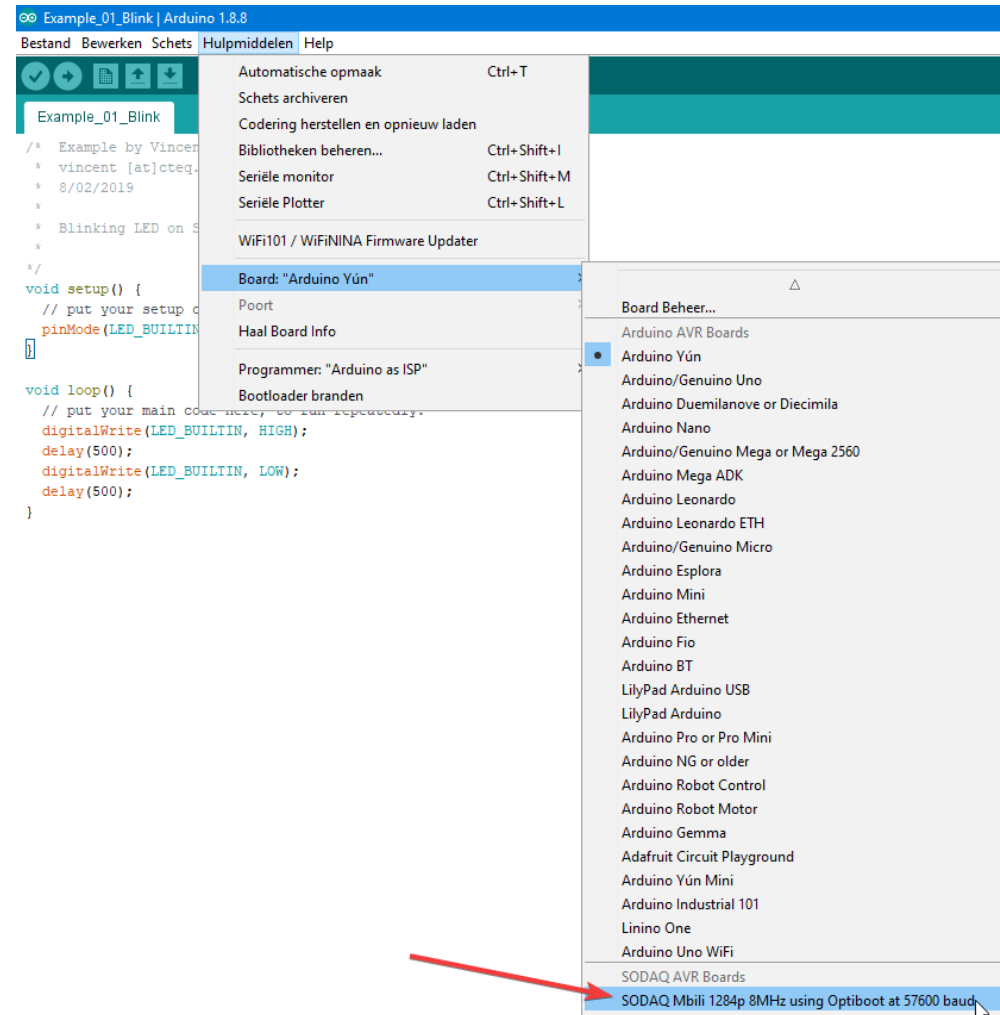


The screenshot shows the Arduino IDE interface with the file 'Example\_01\_Blink' open. The code is as follows:

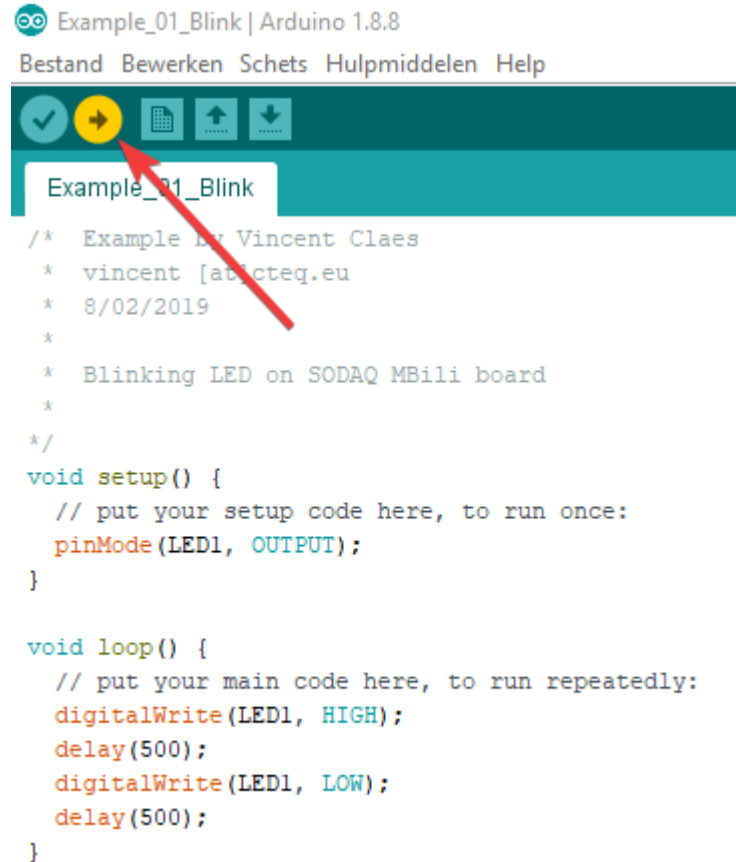
```
/* Example by Vincent Claes
 * vincent [at]cteq.eu
 * 8/02/2019
 *
 * Blinking LED on SODAQ MBili board
 *
 */

void setup() {
  // put your setup code here, to run once:
  pinMode(LED1, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(LED1, HIGH);
  delay(500);
  digitalWrite(LED1, LOW);
  delay(500);
}
```



# Compile and Upload Sketch



```
Example_01_Blink | Arduino 1.8.8
Bestand Bewerken Schets Hulpmiddelen Help

/* Example by Vincent Claes
 * vincent [at] cteq.eu
 * 8/02/2019
 *
 * Blinking LED on SODAQ MBili board
 */

void setup() {
  // put your setup code here, to run once:
  pinMode(LED1, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(LED1, HIGH);
  delay(500);
  digitalWrite(LED1, LOW);
  delay(500);
}
```

Uploaden voltooid.

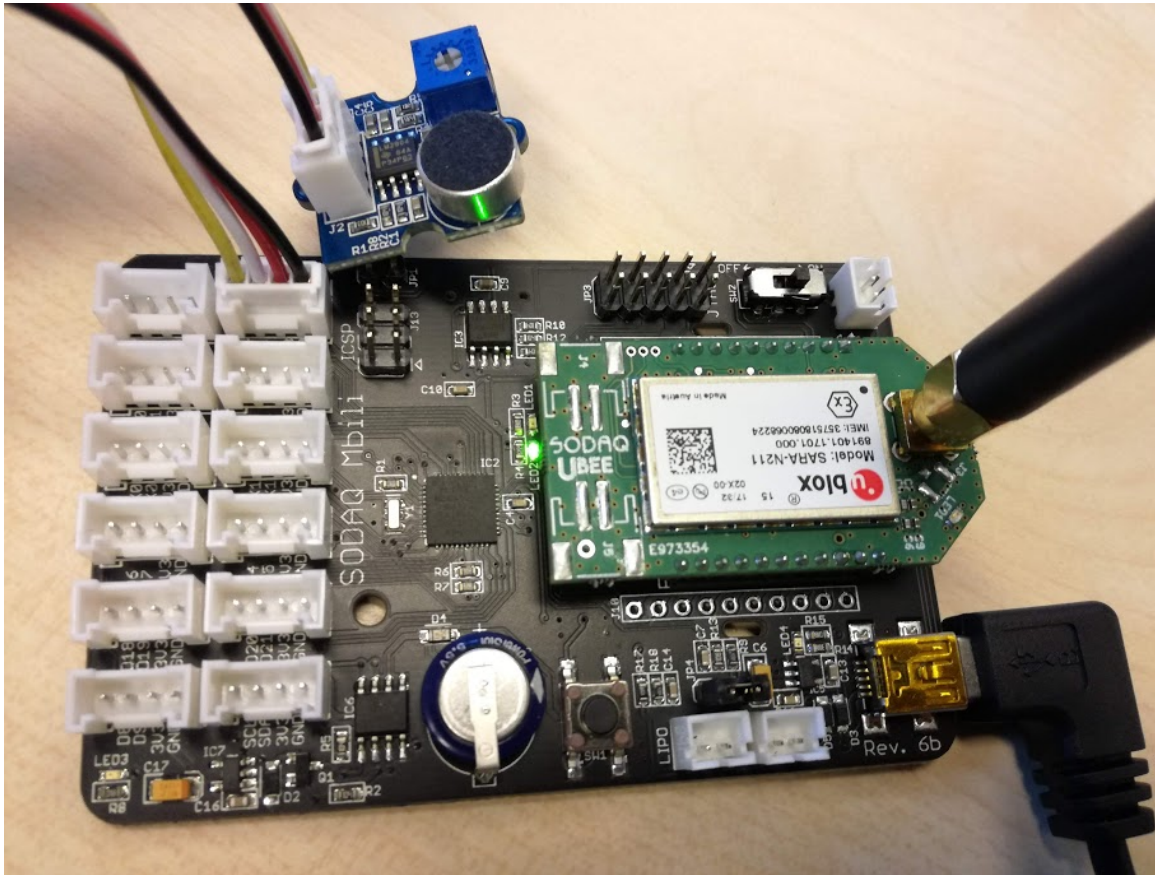
De schets gebruikt 1086 bytes (0%) programma-opslagruimte. Maximum is 130048 bytes.  
Globale variabelen gebruiken 9 bytes van het dynamisch geheugen.



# Exercises

- There is another onboard LED called LED2, try to make an application that alternates between the 2 LEDs every second.
- Create an application that writes “Hello World!” over Serial Port to the computer, you can use the serial terminal from the Arduino IDE to validate your application

# Connecting a Sensor to the MBili



- Connect the Grove – Loudness Sensor to the A4/A5 connector of the SODAQ Mbili board
- [http://wiki.seeedstudio.com/Grove-Loudness\\_Sensor/](http://wiki.seeedstudio.com/Grove-Loudness_Sensor/)

Parameter	Value/Range
Voltage	3.5~10 VDC
Working Frequency	50~2000 Hz
Sensitivity	-48~66 dB
Signal-to-noise Ratio	>58 dB
Output Signal range	Analog Signal (0-1023)

# Example\_02\_Loudness\_Sensor (Analog sensor)

Example\_02\_Loudness\_Sensor | Arduino 1.8.8

Bestand Bewerken Schets Hulpmiddelen Help



Example\_02\_Loudness\_Sensor

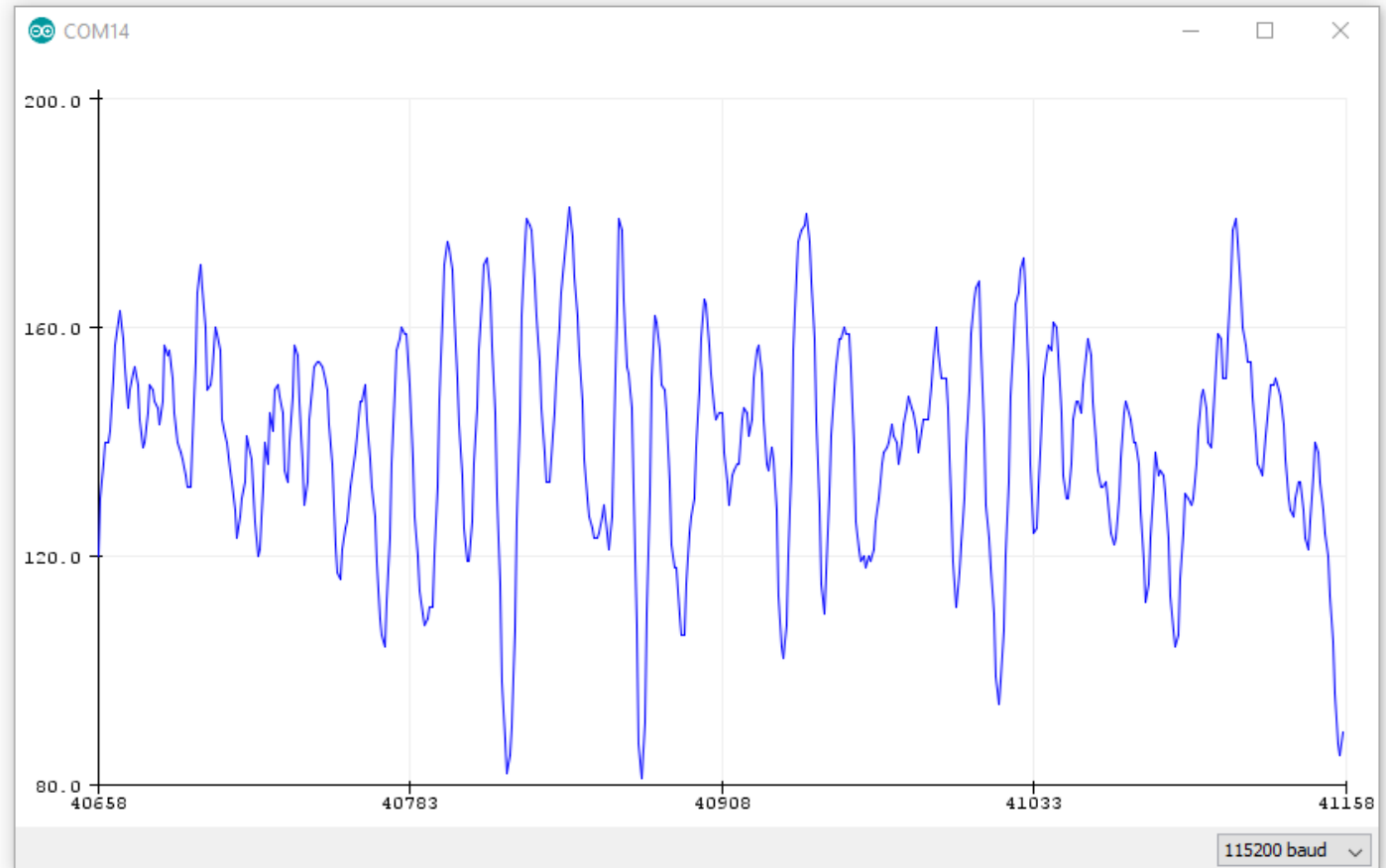
```
/* Example by Vincent Claes
 * vincent [at]cteq.eu
 * 8/02/2019
 *
 * Loudness Sensor on SODAQ MBili board
 */

#define SoundSensorPin A4

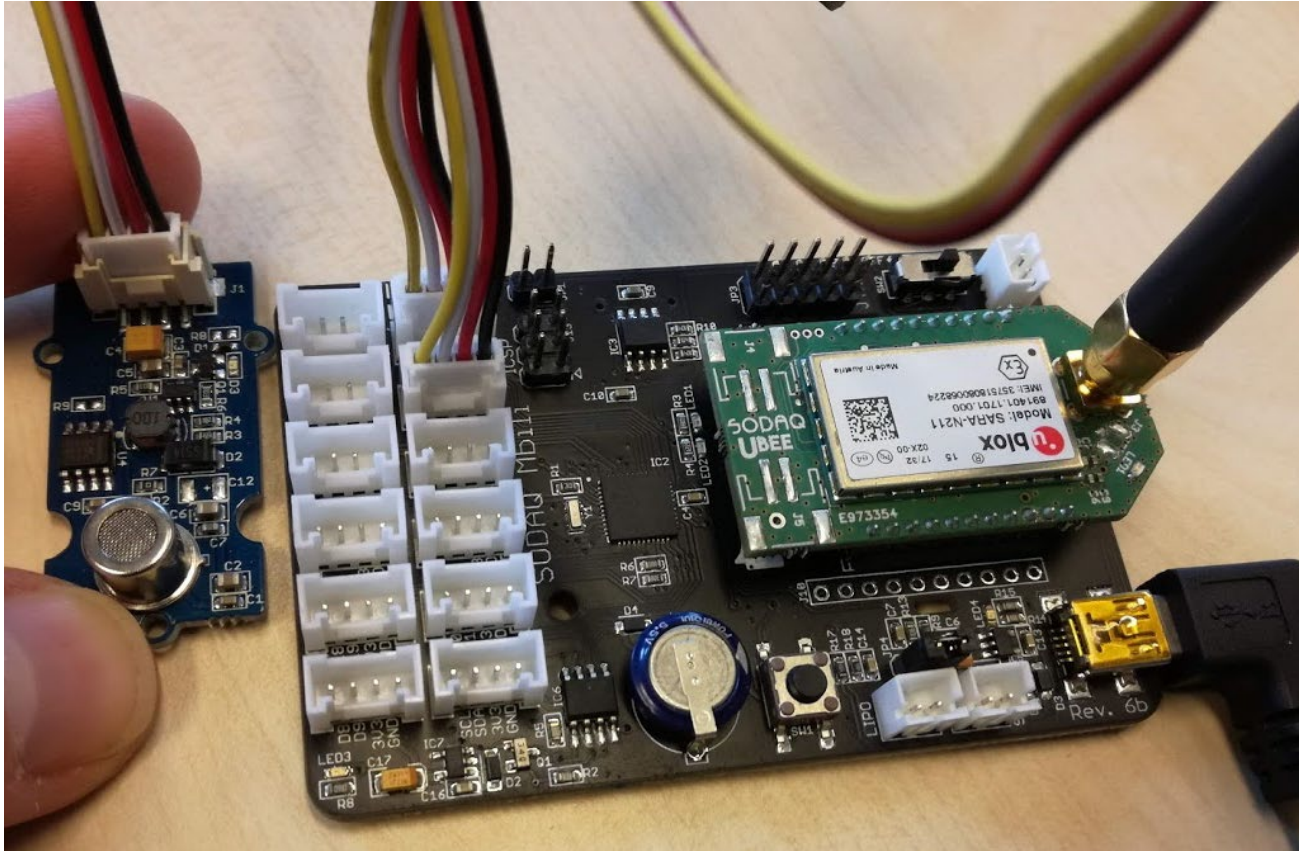
int SoundValue;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  pinMode(SoundSensorPin, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  SoundValue = analogRead(SoundSensorPin);
  Serial.println(SoundValue);
  delay(10);
}
```



# Connecting a Sensor to the MBili



- Connect the Grove – Air Quality Sensor to the A0/A1 connector of the SODAQ Mbili board
- Import the AirQuality Sensor Library Into Arduino IDE
- [http://wiki.seeedstudio.com/Grove-Air Quality Sensor v1.3/](http://wiki.seeedstudio.com/Grove-Air_Quality_Sensor_v1.3/)



# Example\_03\_Air\_Quality (Analoge sensor)

Example\_03\_Air\_Quality\_Sensor | Arduino 1.8.8  
Bestand Bewerken Schets Hulpmiddelen Help



```
/* Example by Vincent Claes
 * vincent [at]cteq.eu
 * 8/02/2019
 *
 * Indoor Air Quality Sensor on SODAQ MBili board
 */
#include "AirQuality2.h"

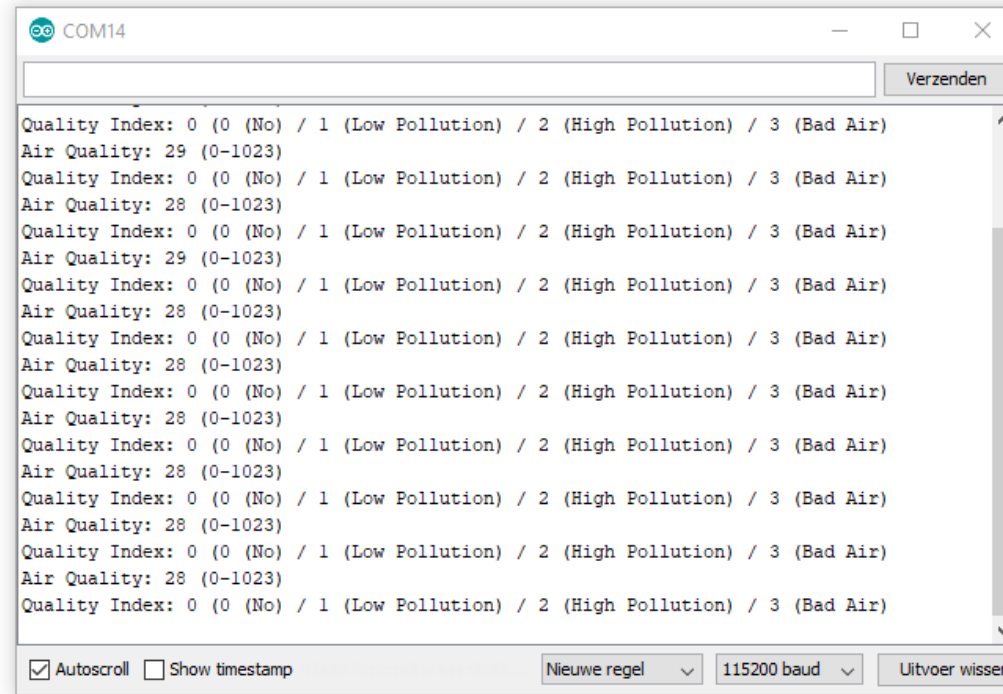
#define AirQualitySensorPin A0

AirQuality2 AirQualitySensor;

int16_t AirQualityValue=-1;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  // pinMode(AirQualitySensorPin, INPUT);
  Serial.println("Before Init...");
  AirQualitySensor.init(AirQualitySensorPin);
  Serial.println("Init Done...");
}

void loop() {
  // put your main code here, to run repeatedly:
  AirQualityValue = AirQualitySensor.getRawData();
  Serial.print("Air Quality: ");
  Serial.print(AirQualityValue);
  Serial.println(" (0-1023)");
  Serial.print("Quality Index: ");
  Serial.print(AirQualitySensor.evaluate());
  Serial.println(" (0 (No) / 1 (Low Pollution) / 2 (High Pollution) / 3 (Bad Air))");
  delay(500);
}
```



# Exercises

- Build an Application to read out the TPH v2 sensor
  - See also: <https://support.sodaq.com/sodaq-one/tpv-v2/>
- Build an Application to read out the Light sensor
- Build an Application to read out the Grove GPS sensor