Environmental Data Acquisition Device

Contents

[1.0 IOIO Development Board 1](#_Toc403123153)

[2.0 GPS 2](#_Toc403123154)

[2.1 PMB-648 2](#_Toc403123155)

[2.2 Wiring 2](#_Toc403123156)

[2.3 Code 2](#_Toc403123157)

[3.0 Temperature Sensor 2](#_Toc403123158)

[3.1 TMP36 2](#_Toc403123159)

[3.2 Wiring 2](#_Toc403123160)

[3.3 Code 3](#_Toc403123161)

[4.0 Humidity Sensor 3](#_Toc403123162)

[4.1 HIH-4030 3](#_Toc403123163)

[4.2Wiring 3](#_Toc403123164)

[4.3 Code 3](#_Toc403123165)

[5.0 Ozone Sensor 3](#_Toc403123166)

[5.1 MiCS-2610 3](#_Toc403123167)

[5.2 Wiring 4](#_Toc403123168)

[5.3 Code 4](#_Toc403123169)

[6.0 Nitrogen Dioxide Sensor 4](#_Toc403123170)

[6.1 MiCS-2710 4](#_Toc403123171)

[6.2 Wiring 4](#_Toc403123172)

[6.3 Code 4](#_Toc403123173)

[7.0 Case 5](#_Toc403123174)

[8.0 Battery 5](#_Toc403123175)

# 1.0 IOIO Development Board

The IOIO board is a development board that provides hardware I/O to Android applications. The board features and PIC microcontroller and peripherals such as GPIO, PWM, ADC, I2C, SPI, and UART. The environmental data acquisition device uses ADC and UART to interface with the sensors. The PIC microcontroller gathers data from the sensors over the peripherals and sends it to the android device over an attached Bluetooth dongle or USB cable. This is all done by sending commands to the PIC microcontroller over Bluetooth. The IOIO board does nothing unless it is receiving commands from the Android device. This limits its use for applications where the phone is often out of range of the IOIO board. To send commands to the board, the IOIOLibAndroid and IOIOLibBT libraries need to be added to the android project. The libraries provide simple commands for accessing the board’s peripherals. In the android application a class needs to be created that extends BaseIOIOLooper. BaseIOIOLooper has a setup method used for instantiating peripherals and a loop method used for reading the values of the peripherals. The setup method is run when the board first connects to the android device while the application is running. After the setup method finishes, the loop method is repeated for the remainder of the application run time.

# 2.0 GPS

## 2.1 PMB-648

The GPS used in both devices is a PMB-648 module. This module tracks up to 20 satellites and updates at least once per second. It has a built in antenna that allows for some indoor use depending on the building structure. The module requires 3.3V – 5V DC, and consumes 65mA @ 5V DC. Data is provided in the NMEA0183 v2.2 format which contains the RMC sentence used in this application. The data strings are transferred using the UART protocol.

## 2.2 Wiring

The yellow wire is the TTL TX pin that the data transfers on this is connected to one of the UART pins on the IOIO board. The black wire is ground, and the red wire is connected to 5V.

## 2.3 Code

In the setup method the UART connection is established using the openUart method. In the loop method there is a check to make sure data it is sending data. If the module is sending data it is read from the InputStream and line by line using the ‘$’ symbol as an indicator for a new line. Once a new line is received there is a check to make sure that it is a RMC sentence. If it isn’t it repeats the above process. If it is a RMC sentence, the loop is exited.

# 3.0 Temperature Sensor

## 3.1 TMP36

The temperature sensor used is the TMP36 device. The TMP36 is a low voltage sensor that requires 2.7V – 5.5V DC, and consumes less than 50uA of current. It scales linearly with temperature with ±2 degrees Celsius accuracy. The output pin provides an analog voltage value that can then be converted to temperature.

## 3.2 Wiring

The outer pins are connected to 3.3V and ground. The center pin is the output that is connected to one of the ADC pins on the device.

## 3.3 Code

There is no setup code for the Temperature sensor because there would be too many analog inputs left open if they were all opened in the setup code. This previously caused an error where the Bluetooth communication couldn’t keep up with the open connections. In the loop method the analog input needs to be opened, read, and then closed. This process is followed by a slight delay to prevent overloading the Bluetooth communication. The voltage value read is then converted to a temperature by using the following equation.

# 4.0 Humidity Sensor

## 4.1 HIH-4030

The humidity sensor used is the HIH-4030 on a breakout board. The HIH-4030 measures relative humidity therefore an external temperature reading is required to get the actual humidity. The board requires 4V – 5.8V DC while consuming 200uA. NOTE: The datasheet claims the board needs at least 4V but other sources have tested and it works well at 3.3V. The sensor can report the relative humidity to ±3%. The output pin provides an analog voltage value that can then be converted to relative humidity.

## 4.2Wiring

The outer pins are connected to 3.3V and ground. The center pin is the output that is connected to one of the ADC pins on the device.

## 4.3 Code

There is no setup code for the Humidity sensor because there would be too many analog inputs left open if they were all opened in the setup code. This previously caused an error where the Bluetooth communication couldn’t keep up with the open connections. In the loop method the analog input needs to be opened, read, and then closed. This process is followed by a slight delay to prevent overloading the Bluetooth communication. The voltage value read is then converted to a relative humidity value by using the following equations.

– 25.8

# 5.0 Ozone Sensor

## 5.1 MiCS-2610

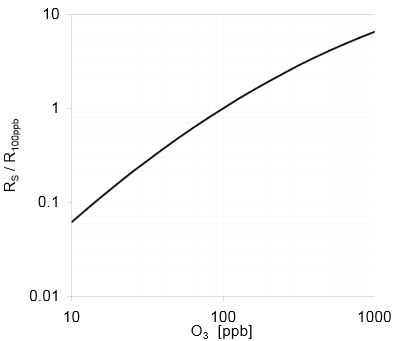
The ozone sensor used is the MiCS-2610. The MiCS-2610 has one circuit for heating the sensor up, and one circuit that measures the O3 in the air. For best operation the heater circuit is powered by 80mW with an internal resistance of 58Ω -78Ω. The sensor circuit is powered by 5V DC and needs to be placed in series with another resistor to measure the resistance of the internal sensor circuit. From additional reading the external resistor value should be close to the internal resistance of the sensor at low O3 concentrations (3kΩ -60kΩ). The internal resistance changes with the O3 concentration in the air within the range of 10-1000 ppb. This sensor is technically not accurate until calibration is completed. This was never completed properly so the readings should only be used to detect changes in the concentration.

## 5.2 Wiring

The heater circuit consists of pins 1 and 3. Pin 1 is connected to ground. Pin 3 is connected to a 75Ω resistor that is connected to the 5V supply. Pins 2 and 4 make up the sensor circuit. Pin 2 is connected to 5V supply. Pin 4 is connected to a 3.3kΩ resistor that is connected to ground. Between pin 4 and the resistor is the analog output value that needs to be connected to the ADC pin on the IOIO board.

## 5.3 Code

There is no setup code for the Ozone sensor because there would be too many analog inputs left open if they were all opened in the setup code. This previously caused an error where the Bluetooth communication couldn’t keep up with the open connections. In the loop method the analog input needs to be opened, read, and then closed. This process is followed by a slight delay to prevent overloading the Bluetooth communication. The voltage value read is then converted to a relative ozone concentration value by using a table, based on the chart below, which maps the sensor circuit resistance to the concentration in ppb.



# 6.0 Nitrogen Dioxide Sensor

## 6.1 MiCS-2710

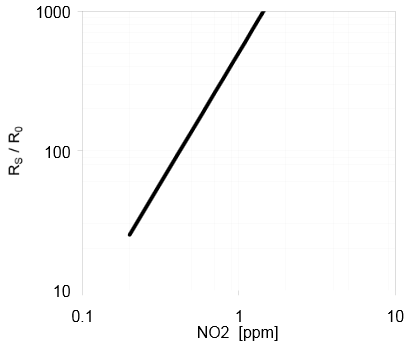
The nitrogen dioxide sensor used is the MiCS-2710. The MiCS-2710 has one circuit for heating sensor up, and one circuit that measures the NO2 in the air. For best operation the heater circuit is powered by 43mW with an internal resistance of 59Ω -73Ω. The sensor circuit is powered by a maximum of 2.5V DC and needs to be placed in series with another resistor to measure the resistance of the internal sensor circuit. From additional reading the external resistor value should be close to the internal resistance of the sensor at low NO2 concentrations (2.2kΩ -8kΩ). The internal resistance changes with the NO2 concentration in the air within the range of 0.05-5 ppm. This sensor is technically not accurate until calibration is completed. This was never completed properly so the readings should only be used to detect changes in the concentration.

## 6.2 Wiring

The heater circuit consists of pins 1 and 3. Pin 1 is connected to ground. Pin 3 is connected to a 220Ω resistor that is connected to the 5V supply. Pins 2 and 4 make up the sensor circuit. Pin 2 is connected to connected to a voltage divider between two 12k resistors that cut the voltage to the maximum of 2.5V required to power the sensor. Pin 4 is connected to a 2.2kΩ resistor that is connected to ground. Between pin 4 and the resistor is one of the analog output values that needs to be connected to the ADC pin on the IOIO board. Since a voltage divider doesn’t provide a steady voltage another wire connects pin 2 to the ADC. This way an accurate voltage drop over the sensor can be read.

## 6.3 Code

There is no setup code for the Ozone sensor because there would be too many analog inputs left open if they were all opened in the setup code. This previously caused an error where the Bluetooth communication couldn’t keep up with the open connections. In the loop method the analog input needs to be opened, read, and then closed. This process is followed by a slight delay to prevent overloading the Bluetooth communication. The two voltage values are then converted to a relative nitrogen dioxide concentration value by using a table, based on the chart below, which maps the sensor circuit resistance to the concentration in ppm.



# 7.0 Case

The case for the device is 3d printed from solid works files. The dimensions are roughly 120mm x 83mm x 41mm for the newer device and 108mm x 83mm x 41mm for the older device. The reason for this size distance is that the IOIO board updated and replaced the USB port with a micro USB port. The change requires an adapter cable to plug in the Bluetooth dongle which takes up additional space.

# 8.0 Battery

The device is powered using 4 AA NiMh rechargeable batteries in series with one another. They each provide roughly 2500mAh of charge and 1.2V. In series they add up to 4.8V which is enough to satisfy the IOIO board’s voltage requirement. The entire sensor unit uses roughly 300mA of current so the device can theoretically run for 8 hours on the 4 AA batteries.