

**Course** : Diploma in Electronics & Computer Engineering (EGDF20)

**Module**  : Connected System Design Project (EGE205)

**Laboratory No**. : SDL Lab 1

**Laboratory Title** : Environmental Related Sensors

**Objective** : To connect hardware click boards, install python library and write python code

to read different types of environmental sensor click boards.

**Hardware Boards** : BBBW Board with USB Cable x1

: MikroBus Cape x1

: Environment Click x1

: UV 3 Click x1

: Proximity Click x1

**Contents**

1. Environment Click
   1. Understanding of Environment Click Board Hardware Connection
   2. Reading the Sensor Value from the Environment Click Board using Python Code
2. UV 3 Click
   1. Understanding of UV 3 Click Board Hardware Connection
   2. Reading the Sensor Value from the UV 3 Click Board using Python Code
3. Proximity Click
   1. Understanding of Proximity Click Board Hardware Connection
   2. Reading the Sensor Value from the Proximity Click Board using Python Code

# **Environment Click**

## Understanding of Environment Click Board Hardware Connection

**Environment Click** board measures temperature, relative humidity, pressure and VOC (Volatile Organic compounds gases). The click board carries the BME680 environmental sensor from Bosch and it is designed to run on a 3.3V power supply. It communicates with the target microcontroller over SPI or I2C interface.

Some of the application such as testing the indoor air quality, to control HVAC (heating, ventilation, and air conditioning) systems, in a weather station, sports applications and more.

The Environment Click, and its respective schematic are shown in the Figure below.



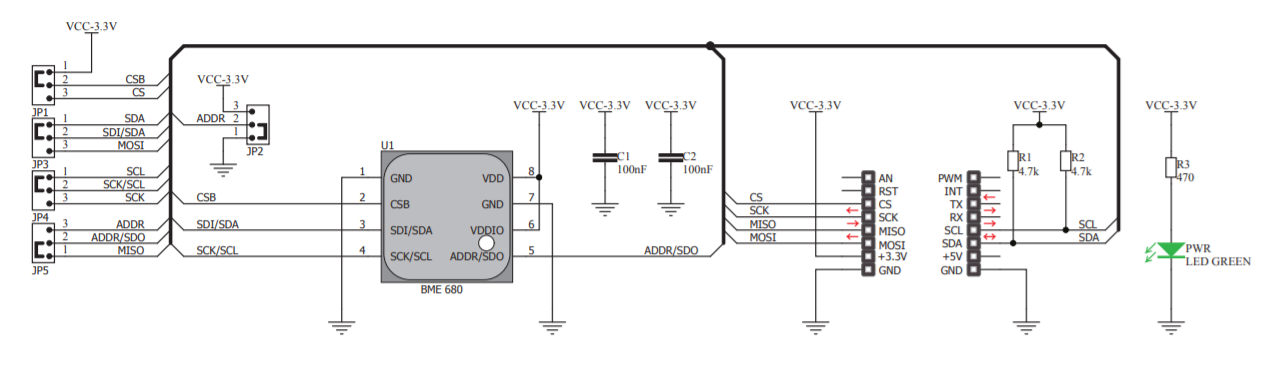


Figure 1.1a: Environment Click and Schematic

1. **Connect** the Environment Click to the mikroBUS cape and BBBW board as shown in the Figure below.

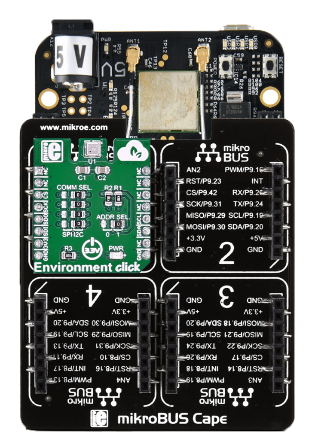


Figure 1.1b: Connecting Environment Click to mikroBUS Cape and BBBW Board

## Reading the Sensor Value from the Environment Click Board using Python Code

**Downloading and Installing Adafruit\_CircuitPython\_BME680 Python Library**

1. **Log in** to the BBBW board through SSH using the default username “**debian**” and password “**temppwd**”.
2. **Type** in the command “**iwconfig**” and **hit** the “Enter” key to ensure that the wlan0 is connected to a particular Wi-Fi’s SSID.
3. **Type** in the command “**pwd**” and **hit** the “Enter” key. It is observed that the current working directory “**/home/debian**” is returned.
4. **Type** in the command “**ls**” and **hit** the “Enter” key. It is observed that the PythonLibrary folder has been created earlier.
5. **Type** in the command “**cd PythonLibrary**” and **hit** the “Enter” key to accesses the PythonLibrary folder.
6. **Type** in the command “**git clone https://github.com/nypege205/Adafruit\_CircuitPython\_BME680.git**” and **hit** the “Enter” key. **Type** in the github username “**nypege205**” and password “**ghp\_Dr3jDaeKJ8fgDH06ZrtG1qUKgsmKux3XffG5**” and **hit** the “Enter” key again to clones a copy of the Adafruit\_CircuitPython\_BME680 Python Library repository from github.
7. **Type** in the command “**ls**” and **hit** the “Enter” key. The Adafruit\_CircuitPython\_BME680 Python Library folder is returned.
8. **Type** in the command “**cd Adafruit\_CircuitPython\_****BME680**” and **hit** the “Enter” key to access the Adafruit\_CircuitPython\_BME680 folder.
9. **Type** in the command “**sudo python3 setup.py install**” and **hit** the “Enter” key to install the Adafruit\_CircuitPython\_BME680 Python Library. Please note that the installation may take up to 5 minutes to complete the whole process.

**Executing the Python Program**

1. **Ensure** that the BBBW board is powered up and connected to the computer through a USB cable. **Open** the web browser (preferably Chrome browser) and **type** “**http://192.168.7.2:3000**” in the address bar.
2. **Right click** on the folder “**MyFirstPythonProject”** and **select** the “**New File**” from the drop-down menu to create a new python file. **Name** the file as “**environment.py**”.
3. **Double click** on the newly created file “**environment.py**” and **enter** the following code into the file under the Editor section.

|  |
| --- |
| import time  import board  import adafruit\_bme680  #Create sensor object, communicating over the board's default I2C bus  i2c = board.I2C()  bme680 = adafruit\_bme680.Adafruit\_BME680\_I2C(i2c, 0x77)  #Singapore mean pressure (hPa) at sea level  bme680.sea\_level\_pressure = 1008.5  #Calibrate the temperature sensor value  temperature\_offset = -5  while True:  print("\nTemperature: %0.1f C" % (bme680.temperature + temperature\_offset))  print("Gas: %d ohm" % bme680.gas)  print("Humidity: %0.1f %%" % bme680.relative\_humidity)  print("Pressure: %0.3f hPa" % bme680.pressure)  print("Altitude = %0.2f meters" % bme680.altitude)  time.sleep(1) |

1. **Click** on the “Run” button located beside the Menu Tab to execute the “**environment.py**” file. It is observed that the temperature, gas, humidity, pressure and altitude values are printed at the Cloud9 console window.

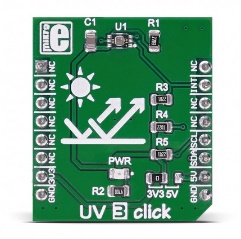
# **UV 3 Click**

## Understanding of UV 3 Click Hardware Connection

**UV 3 Click** board is an advanced ultraviolet (UV) light sensor with I2C protocol interface. The click carries VEML6070 UVA light sensor designed by the CMOS process. It incorporates a photodiode, amplifiers, and analog/digital circuits into a single chip. VEML6070’s adoption of Filtron™ UV technology provides the best spectral sensitivity to cover UV spectrum sensing. It has an excellent temperature compensation and a robust refresh rate setting that does not use an external RC low pass filter.

VEML6070 has linear sensitivity to solar UV light and is easily adjusted by an external resistor. The active acknowledge (ACK) feature with threshold windows setting allows the UV sensor to send out a UVI alert message.

UV 3 Click and its respective schematic are shown in the Figure below.



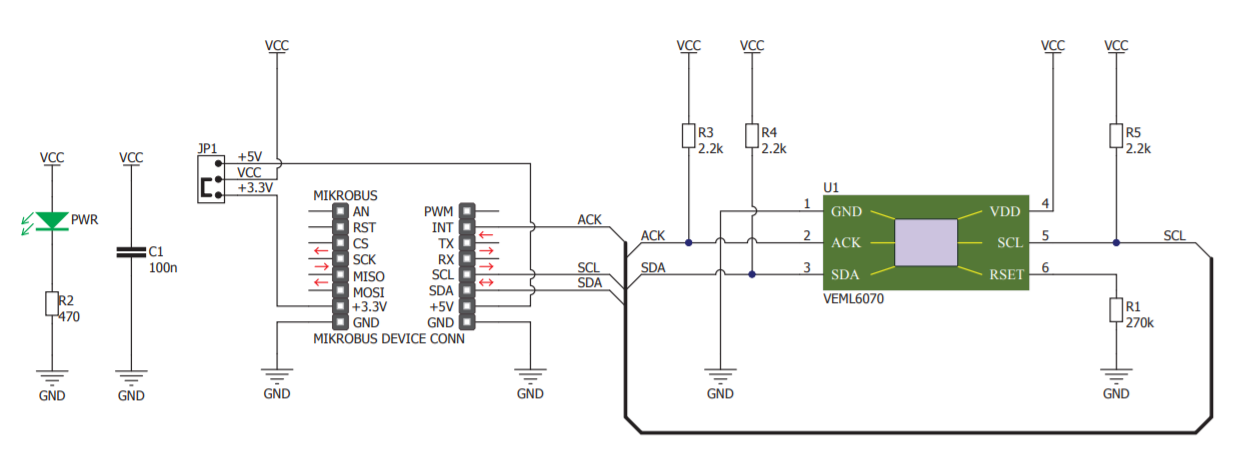


Figure 2.1a: UV 3 Click and Schematic

1. **Connect** the UV 3 Click to the mikroBUS cape and BBBW board as shown in the Figure below.

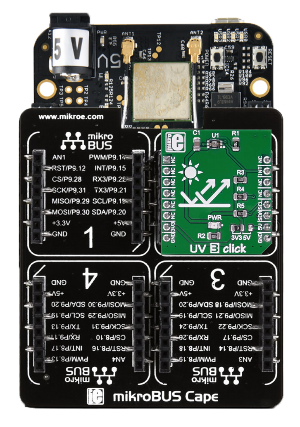


Figure 2.1b: Connecting UV 3 Click to mikroBUS Cape and BBBW Board

## Reading the Sensor Value from the UV 3 Click using Python Code

**Downloading and Installing Adafruit\_CircuitPython VEML6070 Python Library**

1. **Log in** to the BBBW board through SSH using the default username “**debian**” and password “**temppwd**”.
2. **Type** in the command “**iwconfig**” and **hit** the “Enter” key to ensure that the wlan0 is connected to a particular Wi-Fi’s SSID.
3. **Type** in the command “**pwd**” and **hit** the “Enter” key. It is observed that the current working directory “**/home/debian**” is returned.
4. **Type** in the command “**ls**” and **hit** the “Enter” key. It is observed that the PythonLibrary folder has been created earlier.
5. **Type** in the command “**cd PythonLibrary**” and **hit** the “Enter” key to accesses the PythonLibrary folder.
6. **Type** in the command “**git clone** **https://github.com/nypege205/Adafruit\_CircuitPython\_VEML6070.git**” and **hit** the “Enter” key. **Type** in the github username “**nypege205**” and password “**ghp\_Dr3jDaeKJ8fgDH06ZrtG1qUKgsmKux3XffG5**” and **hit** the “Enter” key again to clones a copy of the Adafruit\_CircuitPython\_VEML6070 Python Library repository from github.
7. **Type** in the command “**ls**” and **hit** the “Enter” key. The Adafruit\_CircuitPython\_VEML6070 Python Library folder is returned.
8. **Type** in the command “**cd Adafruit\_CircuitPython\_VEML6070**” and **hit** the “Enter” key to access the Adafruit\_CircuitPython\_VEML6070 folder.
9. **Type** in the command “**sudo python3 setup.py install**” and **hit** the “Enter” key to install the Adafruit\_CircuitPython\_VEML6070 Python Library. Please note that the installation may take up to 5 minutes to complete the whole process.

**Executing the Python Program**

1. **Ensure** that the BBBW board is powered up and connected to the computer through a USB cable. **Open** the web browser (preferably Chrome browser) and **type** “**http://192.168.7.2:3000**” in the address bar.
2. **Right click** on the folder “**MyFirstPythonProject”** and **select** the “**New File**” from the drop-down menu to create a new python file. Name the file as “**uv3.py**”.
3. **Double click** on the newly created file “**uv3.py**” and enter the following code into the file under the Editor section.

|  |
| --- |
| import time  import board  import adafruit\_veml6070  #Create sensor object, communicating over the board's default I2C bus  i2c = board.I2C()  uv = adafruit\_veml6070.VEML6070(i2c)  while True:  uv\_raw = uv.uv\_raw  risk\_level = uv.get\_index(uv\_raw)  print('Reading: {0} | Risk Level: {1}'.format(uv\_raw, risk\_level))  time.sleep(1) |

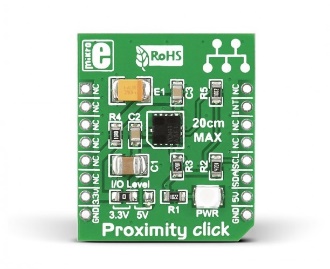
1. **Click** on the “Run” button located beside the Menu Tab to execute the “**uv3.py**” file. It is observed that UV reading and risk level are printed at the Cloud9 console window.

# **Proximity Click**

## Understanding of Proximity Click Board Hardware Connection

**Proximity Click** features VCNL4010 proximity and ambient light sensor. It combines an infrared emitter and PIN photodiode for proximity measurement with a range of up to 20cm. The VCNL4010 has a 16bit resolution which ensures excellent cross talk immunity. It can also be used as ambient light sensor. Proximity click communicates with the target board microcontroller via I2C (SDA, SCL) and INT lines. The board is designed to use 3.3V power supply and 3.3V or 5V I/O voltage levels.

The Proximity Click, and its respective schematic are shown in the Figure below.



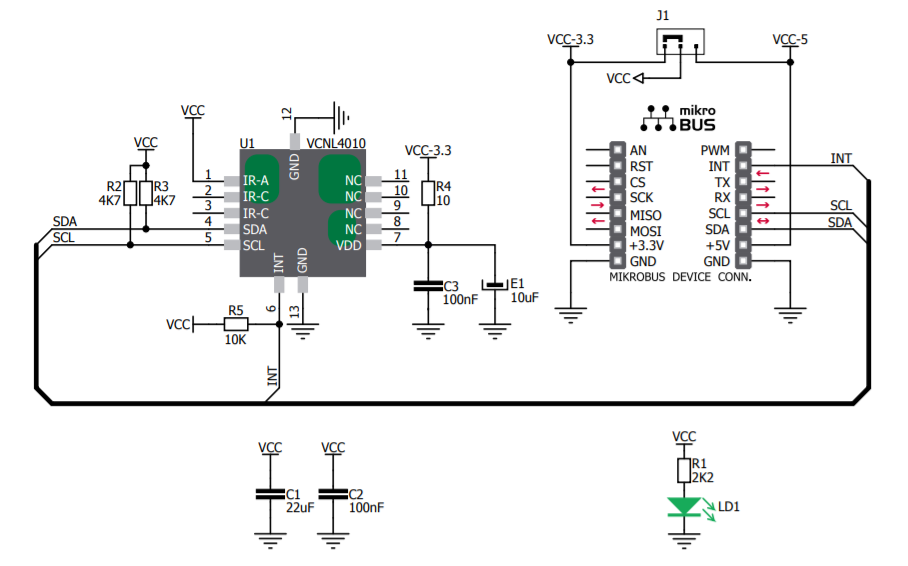


Figure 3.1a: Proximity Click and Schematic

1. **Connect** the Proximity Click to the mikroBUS cape and BBBW board as shown in the Figure below.

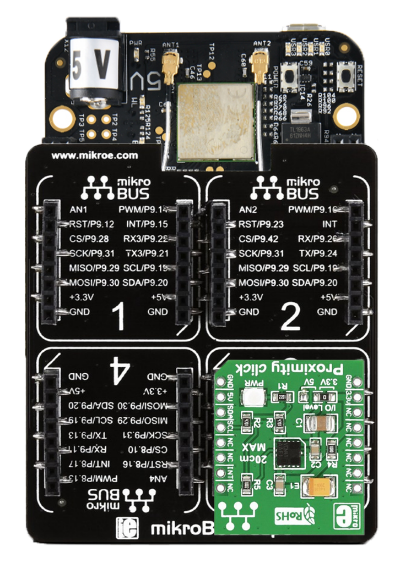


Figure 3.1b: Connecting Proximity Click to mikroBUS Cape and BBBW Board

## Reading the Sensor Value from the Proximity Click Board using Python Code

**Downloading and Installing Adafruit\_CircuitPython\_VCNL4010 Python Library**

1. **Log in** to the BBBW board through SSH using the default username “**debian**” and password “**temppwd**”.
2. **Type** in the command “**iwconfig**” and **hit** the “Enter” key to ensure that the wlan0 is connected to a particular Wi-Fi’s SSID.
3. **Type** in the command “**pwd**” and **hit** the “Enter” key. It is observed that the current working directory “**/home/debian**” is returned.
4. **Type** in the command “**ls**” and **hit** the “Enter” key. It is observed that the PythonLibrary folder has been created earlier.
5. **Type** in the command “**cd PythonLibrary**” and **hit** the “Enter” key to accesses the PythonLibrary folder.
6. **Type** in the command “**git clone https://github.com/nypege205/Adafruit\_CircuitPython\_****VCNL4010.git**” and **hit** the “Enter” key. **Type** in the github username “**nypege205**” and password “**ghp\_Dr3jDaeKJ8fgDH06ZrtG1qUKgsmKux3XffG5**” and **hit** the “Enter” key again to clones a copy of the Adafruit\_CircuitPython\_VCNL4010 Python Library repository from github.
7. **Type** in the command “**ls**” and **hit** the “Enter” key. The Adafruit\_CircuitPython\_VCNL4010 Python Library folder is returned.
8. **Type** in the command “**cd Adafruit\_CircuitPython\_VCNL4010**” and **hit** the “Enter” key to access the Adafruit\_CircuitPython\_VCNL4010 folder.
9. **Type** in the command “**sudo python3 setup.py install**” and **hit** the “Enter” key to install the Adafruit\_CircuitPython\_VCNL4010 Python Library. Please note that the installation may take up to 5 minutes to complete the whole process.

**Executing the Python Program**

1. **Ensure** that the BBBW board is powered up and connected to the computer through a USB cable. **Open** the web browser (preferably Chrome browser) and **type** “**http://192.168.7.2:3000**” in the address bar.
2. **Right click** on the folder “**MyFirstPythonProject”** and **select** the “**New File**” from the drop-down menu to create a new python file. **Name** the file as “**proximity.py**”.
3. **Double click** on the newly created file “**proximity.py**” and **enter** the following code into the file under the Editor section.

|  |
| --- |
| import time  import board  import adafruit\_vcnl4010  i2c = board.I2C()  sensor = adafruit\_vcnl4010.VCNL4010(i2c)  # Main loop runs forever printing the proximity and light level.  while True:  # Proximity has no units and is a 16-bit value. The LOWER the value the further  # an object from the sensor (up to ~200mm).  proximity = sensor.proximity  print("Proximity: {0}".format(proximity))    ambient\_lux = sensor.ambient\_lux  print("Ambient light: {0} lux".format(ambient\_lux))    time.sleep(1.0) |

1. **Click** on the “Run” button located beside the Menu Tab to execute the “**proximity.py**” file. It is observed that the proximity and the ambient lux values are printed at the Cloud9 console window.

*Congratulations! You have successfully completed the SDL Lab. Good job! Take a good break and stay tune for next lab. More exciting lab exercises coming to you!*