

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

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

**Laboratory No**. : SDL Lab 3

**Laboratory Title** : Optical Related Sensor

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

to read different types of optical related sensor click boards.

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

: MikroBus Cape x1

: Color 7 Click x1

: Spectrometer Click x1

: Grid-eye Click x1

**Contents**

1. Color 7 Click
   1. Understanding of Color 7 Click Board Hardware Connection
   2. Reading the Sensor Value from the Color 7 Click Board using Python Code
2. Spectrometer Click
   1. Understanding of Spectrometer Click Board Hardware Connection
   2. Reading the Sensor Value from the Spectrometer Click Board using Python Code
3. Grid-EYE Click
   1. Understanding of Grid-EYE Click Board Hardware Connection
   2. Reading the Sensor Value from the Grid-EYE Click Board using Python Code

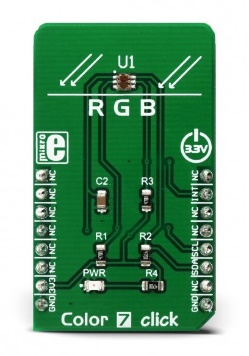
# **Color 7 Click**

## Understanding of Color 7 Click Hardware Connection

**Color 7 Click** is a very accurate color sensing Click board™ which features the TCS3472 color light to digital converter with IR filter, from ams. It contains a 3x4 matrix of photosensitive elements, which can sense red, green, blue and clear light component. Additional IR resistive coating reduces the influence of the IR component of the light spectrum. Four low noise 16bit ADCs ensure the high dynamic range, making this sensor suitable to be used behind dark glass. This sensor offers a fast I2C interface for the communication with the host MCU. Ability to measure light without the influence of the IR makes this device a good choice for Ambient Light Sensing (ALS).

In addition to high accuracy and sensitivity, this color sensor also offers a programmable interrupt pin, used to trigger an interrupt on the host MCU. This allows more efficient controller firmware to be written. Features, such as the high dynamic range, programmable gain and integration time, very high sensitivity, interrupt pin with programmable events, and more, make this Click board™ an ideal solution for LED lighting color management, ambient light sensing for display backlight control, product color verification and sorting, and other similar applications that require an accurate color and ambient light sensing.

Color 7 Click and its respective schematic are shown in the Figure below.



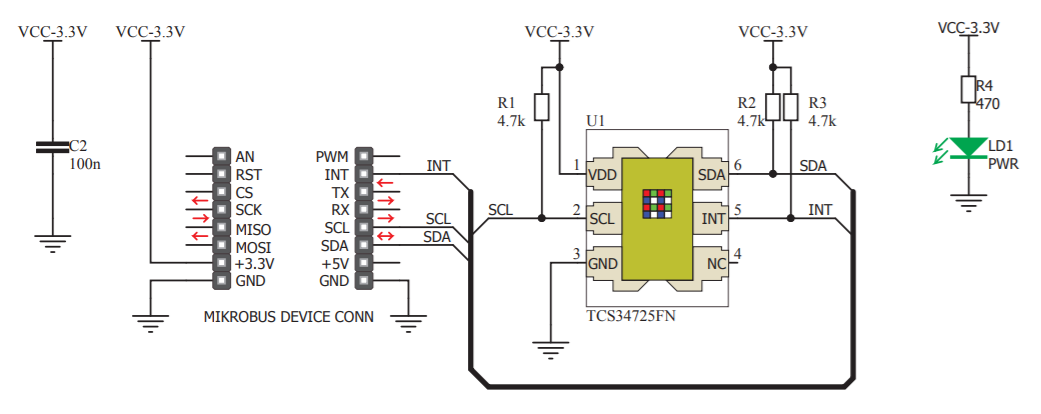


Figure 1.1a: 7 Color Click and Schematic

1. **Connect** the Color 7 Click to the mikroBUS cape and BBBW board as shown in the Figure below.

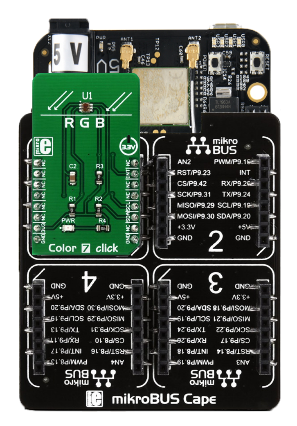


Figure 1.1b: Connecting Color 7 Click to mikroBUS Cape and BBBW Board

## Reading the Sensor Value from the Color 7 Click using Python Code

**Downloading and Installing Adafruit\_CircuitPython\_TCS34725 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\_TCS34725.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\_TCS34725Python Library repository from github.
7. **Type** in the command “**ls**” and **hit** the “Enter” key. The Adafruit\_CircuitPython\_TCS34725 Python Library folder is returned.
8. **Type** in the command “**cd Adafruit\_CircuitPython\_TCS34725**” and **hit** the “Enter” key to access the Adafruit\_CircuitPython\_TCS34725folder.
9. **Type** in the command “**sudo python3 setup.py install**” and **hit** the “Enter” key to install the Adafruit\_CircuitPython\_TCS34725 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 “**color7.py**”.
3. **Double click** on the newly created file “**color7.py**” and enter the following code into the file under the Editor section.

|  |
| --- |
| # Simple demo of the TCS34725 color sensor.  # Will detect the color from the sensor and print it out every second.  import time  import board  import adafruit\_tcs34725  # Create sensor object, communicating over the board's default I2C bus  i2c = board.I2C() # uses board.SCL and board.SDA  sensor = adafruit\_tcs34725.TCS34725(i2c)  # Main loop reading color and printing it every second.  while True:  # Read the color temperature and lux of the sensor too.  temp = sensor.color\_temperature  lux = sensor.lux  print("Temperature: {0}K Lux: {1}".format(temp, lux))  # Delay for a second and repeat.  time.sleep(1.0) |

1. **Click** on the “Run” button located beside the Menu Tab to execute the “**color7.py**” file. It is observed that …………..

# **Spectrometer Click**

## Understanding of Spectrometer Click Board Hardware Connection

**Spectrometer Click** board uses the AS7341 IC, 11-Channel Spectral Sensor Frontend from AMS-AG. This IC features 6 independent optical channels with a dedicated 16-bit light-to-frequency converter. Gain and integration time of the 6 channels can be adjusted with the serial interface. Wait time can be programmed to automatically set a delay between two consecutive spectral measurements and to reduce overall power consumption. The other available channels can be accessed by a multiplexer (SMUX) connecting them to one of the internal ADCs. Also features a 4x4-photodiode array. On top and below the photodiode array there are two photodiodes with dedicated functions such as flicker detection and near-infrared response.

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



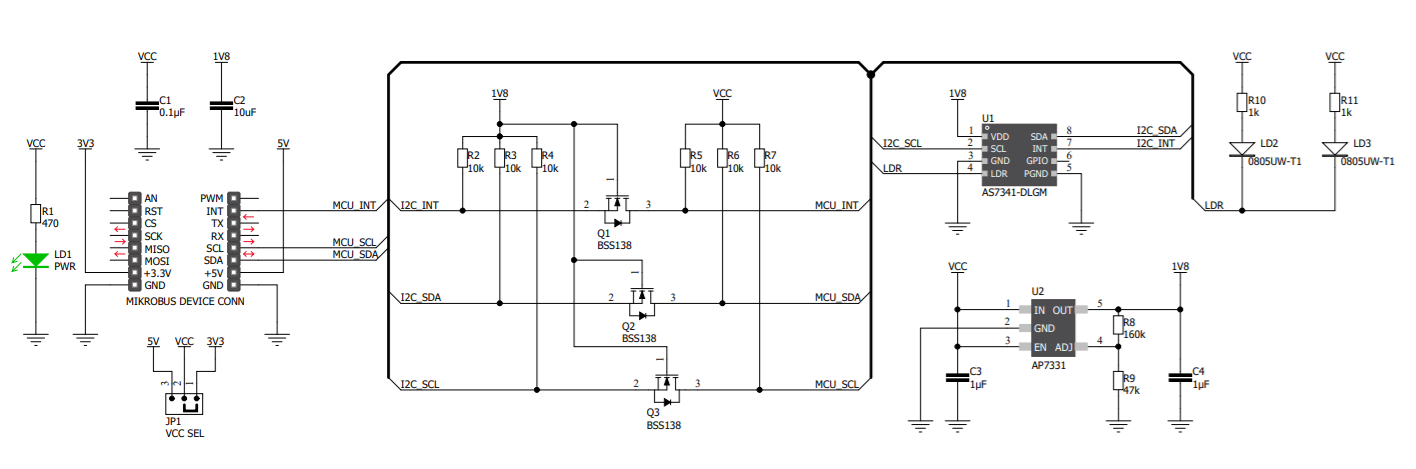


Figure 2.1a: Spectrometer Click and Schematic

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

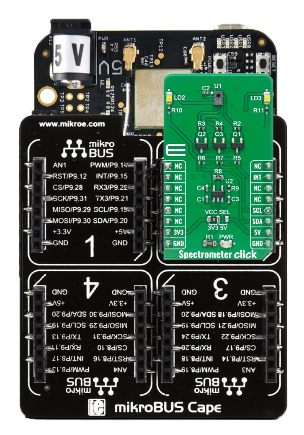


Figure 1.5b: Connecting Spectrometer Click to mikroBUS Cape and BBBW Board

## Reading the Sensor Value from the Spectrometer Click using Python Code

**Downloading and Installing Adafruit\_CircuitPython\_AS7341 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****\_AS7341.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\_AS7341 Python Library repository from github.
7. **Type** in the command “**ls**” and **hit** the “Enter” key. The Adafruit\_CircuitPython\_AS7341 Python Library folder is returned.
8. **Type** in the command “**cd Adafruit\_CircuitPython\_AS7341**” and **hit** the “Enter” key to access the Adafruit\_CircuitPython\_AS7341 folder.
9. **Type** in the command “**sudo python3 setup.py install**” and **hit** the “Enter” key to install the Adafruit\_CircuitPython\_AS7341 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 “**spectrometer.py**”.
3. **Double click** on the newly created file “**spectrometer.py**” and enter the following code into the file under the Editor section.

|  |
| --- |
| from time import sleep  import board  from adafruit\_as7341 import AS7341  i2c = board.I2C() # uses board.SCL and board.SDA  sensor = AS7341(i2c)  def bar\_graph(read\_value):  scaled = int(read\_value / 1000)  return "[%5d] " % read\_value + (scaled \* "\*")  while True:  print("F1 - 415nm/Violet %s" % bar\_graph(sensor.channel\_415nm))  print("F2 - 445nm//Indigo %s" % bar\_graph(sensor.channel\_445nm))  print("F3 - 480nm//Blue %s" % bar\_graph(sensor.channel\_480nm))  print("F4 - 515nm//Cyan %s" % bar\_graph(sensor.channel\_515nm))  print("F5 - 555nm/Green %s" % bar\_graph(sensor.channel\_555nm))  print("F6 - 590nm/Yellow %s" % bar\_graph(sensor.channel\_590nm))  print("F7 - 630nm/Orange %s" % bar\_graph(sensor.channel\_630nm))  print("F8 - 680nm/Red %s" % bar\_graph(sensor.channel\_680nm))  print("\n------------------------------------------------")  sleep(1) |

1. **Click** on the “Run” button located beside the Menu Tab to execute the “**spectrometer.py**” file. It is observed …………………….

# **Grid-EYE Click**

## Understanding of Grid-EYE Click Board Hardware Connection

**Grid-EYE Click** board is a 8x8 thermal array sensor-detector that carries the AMG8853 infrared array sensor from Panasonic. The click is designed to run on either 3.3V or 5V power supply. It communicates with the target MCU over I2C interface.

Use the Grid-EYE click to detect absolute surface temperature without any contact. Or use it to detect the movement of people and objects.

The AMG8853 is made out of 64 individual thermal sensors. It can build an image according to the heat it detects. You don’t need light to form a picture. The temperature measuring range is from -20°C to +100°C. The detecting distance is 5m, the viewing angle 60 degrees. The AMG8853 has a built-in thermistor for suppressing ambient temperature noise. The I2C levels and address are jumper selectable, while the AMG8853 is supplied with designated voltage.

Grid-EYE Click and its respective schematic are shown in the Figure below.



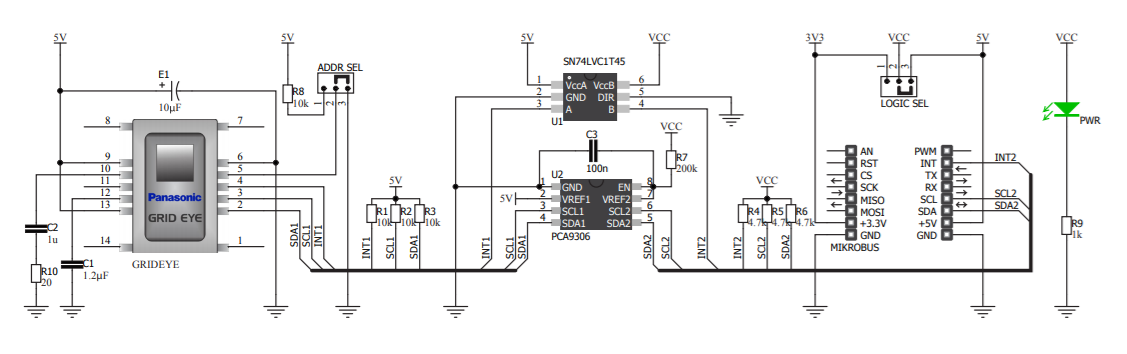


Figure 3.1a: Grid-EYE Click and Schematic

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

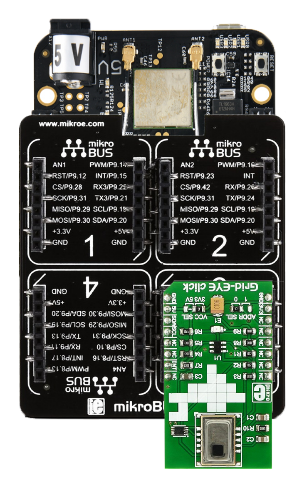


Figure 3.1b: Connecting Grid-EYE Click to mikroBUS Cape and BBBW Board

## Reading the Sensor Value from the Grid-EYE Click using Python Code

**Downloading and Installing Adafruit\_CircuitPython\_AMG88xx 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\_AMG88xx.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\_AMG88xx Python Library repository from github.
7. **Type** in the command “**ls**” and **hit** the “Enter” key. The Adafruit\_CircuitPython\_AMG88xx Python Library folder is returned.
8. **Type** in the command “**cd Adafruit\_CircuitPython\_AMG88xx**” and **hit** the “Enter” key to access the Adafruit\_CircuitPython\_AMG88xx folder.
9. **Type** in the command “**sudo python3 setup.py install**” and **hit** the “Enter” key to install the Adafruit\_CircuitPython\_AMG88xx 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 “**grideye.py**”.
3. **Double click** on the newly created file “**grideye.py**” and enter the following code into the file under the Editor section.

|  |
| --- |
| import time  import busio  import board  import adafruit\_amg88xx  i2c = busio.I2C(board.SCL, board.SDA)  amg = adafruit\_amg88xx.AMG88XX(i2c, 0x68)  while True:  for row in amg.pixels:  # Pad to 1 decimal place  print(["{0:.1f}".format(temp) for temp in row])  print("")  print("\n")  time.sleep(1) |

1. **Click** on the “Run” button located beside the Menu Tab to execute the “**grideye.py**” file. It is observed ………………..

*Congratulations! You have successfully completed the SDL Lab3. Good job! Take a good break and stay tune for next lab. More exciting lab exercises coming to you!*