

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

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

**Laboratory No**. : Lab 2b

**Laboratory Title** : Display: Controlling the LED Matrix and OLED using BeagleBone Black Wireless

(BBBW) Board

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

to control different types of display click boards.

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

: MikroBus Cape x1

: 8x8 R Click x1

: OLED B Click x1

**Contents**

1. Controlling the LED Matrix using BeagleBone Black Wireless (BBBW) Board
   1. Understanding of 8x8 (R/G/Y/B) Click Hardware Connection
   2. Controlling the 8x8 (R/G/Y/B) Click using Python Code
2. Controlling the OLED using BeagleBone Black Wireless (BBBW) Board
   1. Understanding of OLED (W/B) Click Hardware Connection
   2. Controlling the OLED (W/B) Click using Python Code
   3. Controlling both the 8x8 R and OLED B Clicks using Python Code

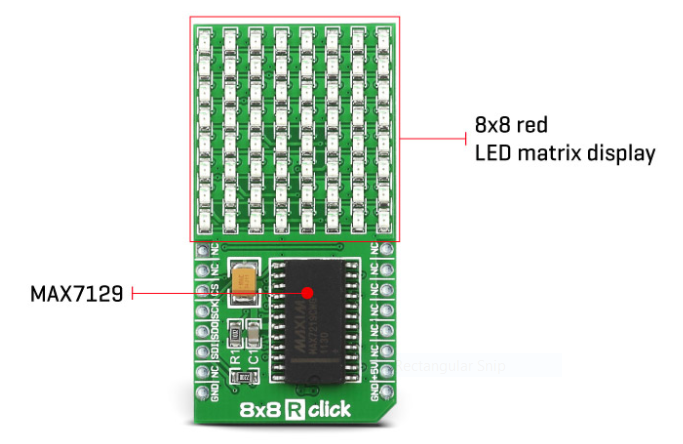
# **Controlling the LED Maxtrix using BeagleBone Black Wireless (BBBW) Board**

## Understanding of 8x8 (R/G/Y/B) Click Hardware Connection

**8x8 R Click** is a 64 LED matrix display Click board, composed of SMD LEDs organized in 8 rows by 8 columns. It has a digital brightness control in 16 steps, it can control every LED in the display matrix independently, it blanks the display on power up to eliminate glitches and it requires a single resistor to control the current through all the LEDs at once, which simplifies the design. 8x8 R Click uses a fast SPI communication protocol, allowing fast display response and no lag.

8x8 R Click can be used as a display or signalization output for a range of applications that are designed to display various information or graphics on the matrix LED display.

The 8x8 R Click, and its respective schematic are shown in the Figure below.



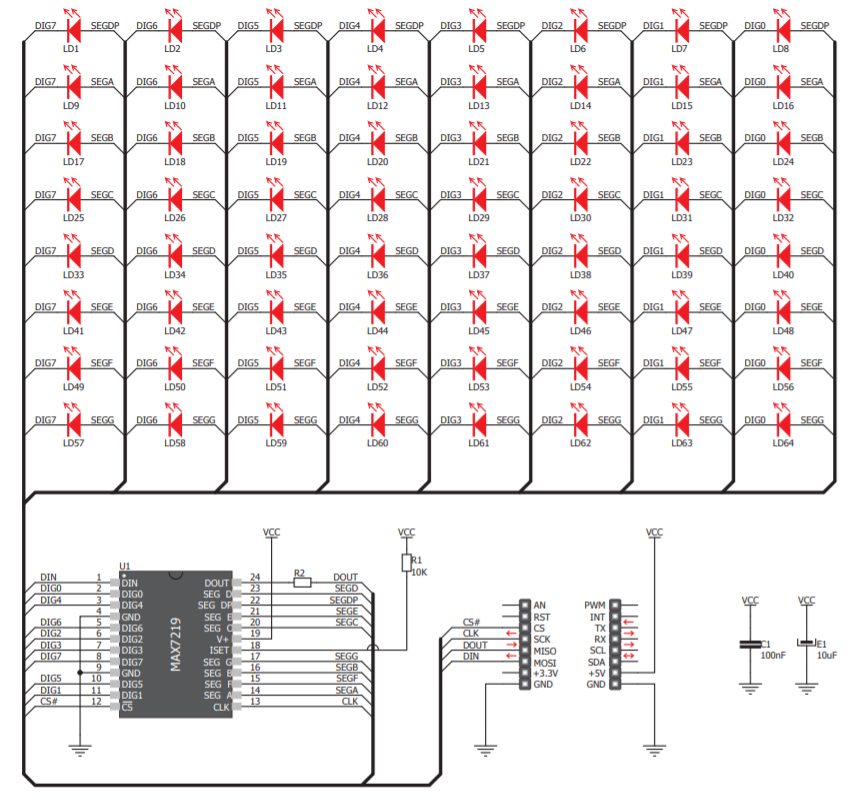


Figure 1.1a: 8x8 R Click and Schematic

1. **Connect** the 8x8 R Click to the mikroBUS cape and BBBW board as shown in the Figure below.

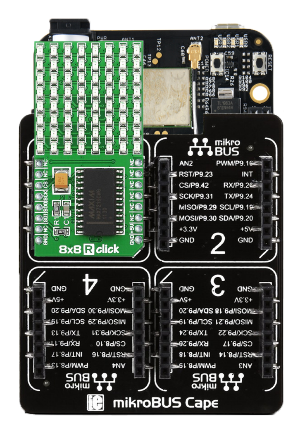


Figure 1.1b: Connecting 8x8 R Click to mikroBUS Cape and BBBW Board

## Controlling the 8x8 R Click using Python Code

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 “**8x8r.py**”.
3. **Double click** on the newly created file “**8x8r.py**” and **enter** the following code into the file under the Editor section.

|  |
| --- |
| from Adafruit\_BBIO.SPI import SPI  G\_SmileyHappyFace = [0b00111100,  0b01000010,  0b10101001,  0b10000101,  0b10000101,  0b10101001,  0b01000010,  0b00111100]  def LedMatrix8x8ClickInit():  L\_Spi1 = SPI(1,0)  L\_Spi1.mode = 0  L\_Spi1.writebytes([0x09, 0x00])  L\_Spi1.writebytes([0x0A, 0x01])  L\_Spi1.writebytes([0x0B, 0x07])  L\_Spi1.writebytes([0x0C, 0x01])  return L\_Spi1  def PrintDisplay(L\_Spi1, DisplayList):  L\_Spi1.writebytes([0x01, DisplayList[0]])  L\_Spi1.writebytes([0x02, DisplayList[1]])  L\_Spi1.writebytes([0x03, DisplayList[2]])  L\_Spi1.writebytes([0x04, DisplayList[3]])  L\_Spi1.writebytes([0x05, DisplayList[4]])  L\_Spi1.writebytes([0x06, DisplayList[5]])  L\_Spi1.writebytes([0x07, DisplayList[6]])  L\_Spi1.writebytes([0x08, DisplayList[7]])    G\_Spi1 = LedMatrix8x8ClickInit()  PrintDisplay(G\_Spi1, G\_SmileyHappyFace) |

1. The code above uses the functions explained below to make the display on the 8x8 R Click possible.

|  |  |
| --- | --- |
| **Function** | **Description** |
| LedMatrix8x8ClickInit() | To initialize the required communication channel (SPI) and return the initialized SPI instance to the function call. |
| PrintDisplay(L\_Spi1, DisplayList) | To print the display on the 8x8 R Click. The function needs to be supplied with an initialized SPI instance to L\_Spi1 and a list that contains the information to be printed. |

1. **Click** on the “Run” button located beside the Menu Tab to execute the “**8x8r.py**” file. It is observed that a smiley Happy Face is printed on the 8x8 R Click.
2. **Try** creating a new list loaded with different bits information to display different graphics on the 8x8 R Click and show it to your lecturer.

# **Controlling the OLED (W/B) using BeagleBone Black Wireless (BBBW) Board**

## Understanding of OLED (W/B) Click Hardware Connection

**OLED B Click** carries a 96 x 39px blue monochrome passive matrix OLED display. The display is bright, has a wide viewing angle and low power consumption. To drive the display, OLED B click features an SSD1306 controller. Its built-in functionalities include contrast control, normal or inverse image display, vertical and horizontal scrolling functions and more. OLED B click can communicate with the target board MCU either through SPI or I2C mikroBUS lines. You switch between output options by resoldering the onboard SEL COMM jumpers (J1, J2 and J3) to the appropriate position.

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



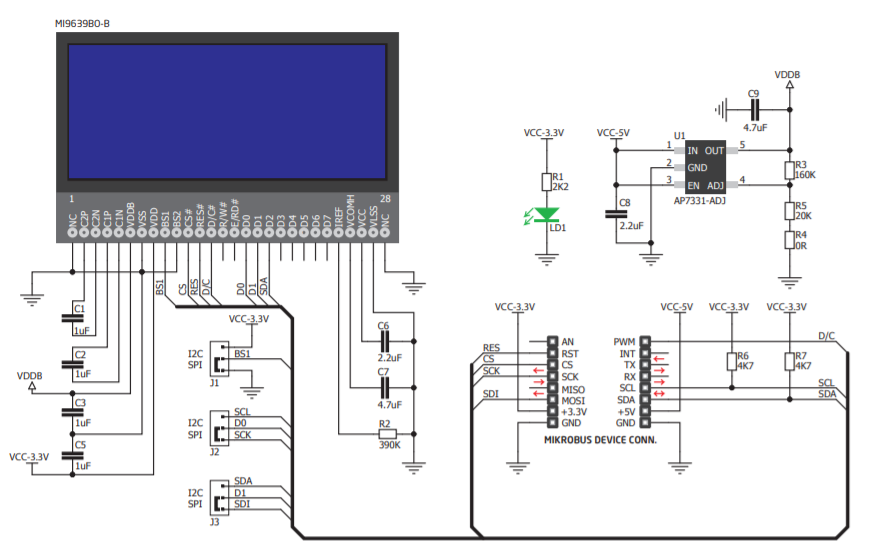


Figure 2.1a: OLED B Click and Schematic

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

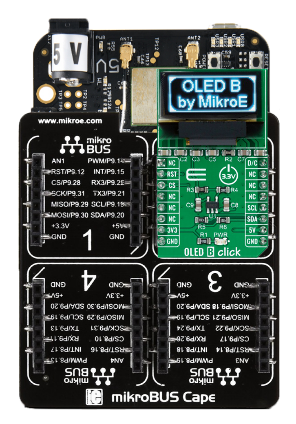


Figure 2.1b: Connecting OLED B Click to mikroBUS Cape and BBBW Board

## Controlling the OLED (W/B) Click using Python Code

**Downloading and Installing Adafruit\_CircuitPython\_SSD1306 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\_SSD1306.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\_SSD1306 Python Library repository from github.
7. **Type** in the command “**ls**” and **hit** the “Enter” key. The Adafruit\_CircuitPython\_SSD1306 Python Library folder is returned.
8. **Type** in the command “**cd Adafruit\_CircuitPython\_SSD1306**” and **hit** the “Enter” key to access the Adafruit\_CircuitPython\_SSD1306 folder.
9. **Type** in the command “**sudo python3 setup.py install**” and **hit** the “Enter” key to install the Adafruit\_CircuitPython\_SSD1306 Python Library. Please note that the installation may take up to 5 minutes to complete the whole process.

**Installing Pillow Python Library using Pip**

1. **Type** in the command “**python3 -m pip install --upgrade Pillow**” and **hit** the “Enter” key to install the Pillow Python Library using **pip** which is the package installer for Python.

**Installing libopenjp2-7 (JPEG 2000 image compression/decompression library) using apt-get**

1. **Type** in the command “**sudo apt-get install libopenjp2-7**” and **hit** the “Enter” key to install the libopenjp2-7 Python Library using **apt-get**.

**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 “**Oledb.py**”.
3. **Double click** on the newly created file “**Oledb.py**” and enter the following code into the file under the Editor section.

|  |
| --- |
| import board  import busio  import digitalio  import adafruit\_ssd1306  from board import SCL, SDA  from PIL import Image, ImageDraw, ImageFont  def OLEDClickInit():  Pin\_DC = digitalio.DigitalInOut(board.P9\_16)  Pin\_DC.direction = digitalio.Direction.OUTPUT  Pin\_DC.value = False  Pin\_RESET = digitalio.DigitalInOut(board.P9\_23)  Pin\_RESET.direction = digitalio.Direction.OUTPUT  Pin\_RESET.value = True  L\_I2c = busio.I2C(SCL, SDA)  return L\_I2c    G\_I2c = OLEDClickInit()  Display = adafruit\_ssd1306.SSD1306\_I2C(64, 32, G\_I2c, addr=0x3C)  ImageObj = Image.new("1", (Display.width, Display.height))  Draw = ImageDraw.Draw(ImageObj)  Draw.rectangle((0, 0, Display.width - 1, Display.height - 1), outline=1, fill=0)  Font = ImageFont.load\_default()  Text = "I Love NYP"  Draw.text((2, 10), Text, font=Font, fill=1)  Display.image(ImageObj)  Display.show() |

1. The code above uses the functions explained below to make the display on the OLED B Click possible.

|  |  |
| --- | --- |
| **Function** | **Description** |
| OLEDClickInit() | To initialize the required GPIO and communication channel (I2C) and return the initialized I2C instance to the function call. |

1. **Click** on the “Run” button located beside the Menu Tab to execute the “**Oledb.py**” file. It is observed that a rectangle outline is printed with a line of text “**I Love NYP**” printed inside.
2. **Try** modifyingthe code to change the text and draw a smaller rectangle and display it on the OLED B Click and show it to your lecturer.

## Controlling both the 8x8 R and OLED B Clicks using Python Code

1. **Connect** both the 8x8 R and OLED B Clicks to the mikroBUS cape and BBBW board as shown in the Figure below.

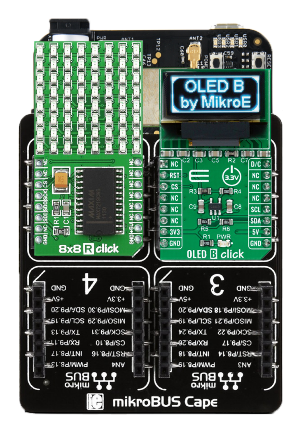


Figure 2.3a: Connecting both The 8x8 R and OLED B Click to mikroBUS Cape and BBBW Board

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 “**8x8\_Oled.py**”.
3. **Double click** on the newly created file “**8x8\_Oled.py**” and start entering code into the file under the Editor section.
4. **Enter** the code that imports all the necessary library to be used in the program as shown below.

|  |
| --- |
| import time  import board  import busio  import digitalio  import adafruit\_ssd1306  from board import SCL, SDA  from PIL import Image, ImageDraw, ImageFont  from Adafruit\_BBIO.SPI import SPI |

1. **Enter** the code of the 4 lists to be used in the program as shown below.

|  |
| --- |
| G\_ClearDisplay = [0b00000000,  0b00000000,  0b00000000,  0b00000000,  0b00000000,  0b00000000,  0b00000000,  0b00000000]  G\_CharacterN = [0b00000000,  0b01111110,  0b00000100,  0b00001000,  0b00010000,  0b00100000,  0b01111110,  0b00000000]  G\_CharacterY = [0b00000000,  0b00000000,  0b01100000,  0b00010000,  0b00001110,  0b00010000,  0b01100000,  0b00000000]  G\_CharacterP = [0b00000000,  0b00000000,  0b00110000,  0b01001000,  0b01001000,  0b01111110,  0b00000000,  0b00000000] |

1. **Enter** the code of the 3 functions to be called in the program as shown below.

|  |
| --- |
| def OLEDClickInit():  Pin\_DC = digitalio.DigitalInOut(board.P9\_16)  Pin\_DC.direction = digitalio.Direction.OUTPUT  Pin\_DC.value = False  Pin\_RESET = digitalio.DigitalInOut(board.P9\_23)  Pin\_RESET.direction = digitalio.Direction.OUTPUT  Pin\_RESET.value = True  L\_I2c = busio.I2C(SCL, SDA)  return L\_I2c  def LedMatrix8x8ClickInit():  L\_Spi1 = SPI(1,0)  L\_Spi1.mode = 0  L\_Spi1.writebytes([0x09, 0x00])  L\_Spi1.writebytes([0x0A, 0x01])  L\_Spi1.writebytes([0x0B, 0x07])  L\_Spi1.writebytes([0x0C, 0x01])  return L\_Spi1  def PrintDisplay(L\_Spi1, DisplayList):  L\_Spi1.writebytes([0x01, DisplayList[0]])  L\_Spi1.writebytes([0x02, DisplayList[1]])  L\_Spi1.writebytes([0x03, DisplayList[2]])  L\_Spi1.writebytes([0x04, DisplayList[3]])  L\_Spi1.writebytes([0x05, DisplayList[4]])  L\_Spi1.writebytes([0x06, DisplayList[5]])  L\_Spi1.writebytes([0x07, DisplayList[6]])  L\_Spi1.writebytes([0x08, DisplayList[7]]) |

1. **Enter** the main code that call the 4 functions to control both the 8x8 R and OLED B Click as shown below.

|  |
| --- |
| G\_I2c = OLEDClickInit()  G\_Spi1 = LedMatrix8x8ClickInit()  Display = adafruit\_ssd1306.SSD1306\_I2C(64, 32, G\_I2c, addr=0x3C)  ImageObj = Image.new("1", (Display.width, Display.height))  Draw = ImageDraw.Draw(ImageObj)  Draw.rectangle((0, 0, Display.width - 1, Display.height - 1), outline=1, fill=0)  Font = ImageFont.load\_default()  while True:  Draw.text((10, 10), 'N', font=Font, fill=1)  Display.image(ImageObj)  Display.show()  PrintDisplay(G\_Spi1, G\_CharacterN)  time.sleep(1)  Draw.text((20, 10), 'Y', font=Font, fill=1)  Display.image(ImageObj)  Display.show()  PrintDisplay(G\_Spi1, G\_CharacterY)  time.sleep(1)  Draw.text((30, 10), 'P', font=Font, fill=1)  Display.image(ImageObj)  Display.show()  PrintDisplay(G\_Spi1, G\_CharacterP)  time.sleep(1)  Draw.rectangle((5, 10, 40, 20), outline=0, fill=0)  Display.image(ImageObj)  Display.show()  PrintDisplay(G\_Spi1, G\_ClearDisplay)  time.sleep(1) |

1. **Click** on the “Run” button located beside the Menu Tab to execute the “**8x8\_Oled.py**” file.
2. **Observe** and **compare** the output on both the 8x8 R and OLED Clicks with your teammates and **consult** your lecturer for advice if it is not the same.
3. **Write** the program output in the white box below for future reference if needed.

|  |
| --- |
| *Right click and select “New comment” to insert your program as a comment.* |

*Congratulations! You have successfully completed the Lab2b. Good job! Take a good break and stay tune for next lab. More exciting lab exercises coming to you!*