

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

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

**Laboratory No**. : Lab 4b

**Laboratory Title** : Sensor and Actuator: Reading the Analog Sensor Data and Controlling the

Actuator using BeagleBone

**Objective** : To connect hardware click boards and write python code to read data from

different types of sensors and control different types of actuators.

**Hardware Boards** : BBBW Board x1

: MikroBus Cape x1

: IR Distance Click x1

: Flame Click x1

: Mic Click x1

: Force Click x1

: Vibro Motor Click x1

: Buzz 2 Click x1

**Contents**

1. Reading the Analog Sensors Data using BeagleBone Black Wireless (BBBW) Board
   1. Understanding of IR Distance Click Hardware Connection
   2. Reading the Analog Data from IR Distance Click using Python Code
   3. Understanding of Flame Click Hardware Connection
   4. Reading the Analog Data from Flame Click using Python Code
   5. Understanding of Mic Click Hardware Connection
   6. Reading the Analog Data from Mic Click using Python Code
   7. Understanding of Force Click Hardware Connection
   8. Reading the Analog Data from Force Click using Python Code
2. Controlling the Actuator using BeagleBone Black Wireless (BBBW) Board
   1. Understanding of Vibro Motor Click Hardware Connection
   2. Controlling the Vibro Motor Click using Python Code
3. Controlling the Actuator based on Sensor Data using BeagleBone Black Wireless (BBBW) Board
   1. Developing an IR Piano using IR Distance and Buzz 2 Click
   2. Tinkering Time

# **Reading the Analog Sensors Data using BeagleBone Black Wireless (BBBW) Board**

## Understanding of IR Distance Click Hardware Connection

**IR Distance Click** carries Sharp’s GP2Y0A60SZ0F distance measuring sensor, which comprises of an integrated PSD (position sensitive detector), an infrared LED and a signal processing circuit. The measuring range is between 10 and 150 cm.

The sensor on IR Distance Click is not easily influenced by variations caused by the reflectivity of the object whose distance is measured (it won’t be able, however, to detect the distance of a mirror). For optimal operation the lens should be kept clean, because dust, water or oil on its surface can impact the precision of the sensor. With these precautions in mind, the IR Distance Click can be used for designing touch-less switches, various energy-saving devices, but it’s also suitable for robotics.

IR Distance Click, and its respective schematic are shown in the Figure below.



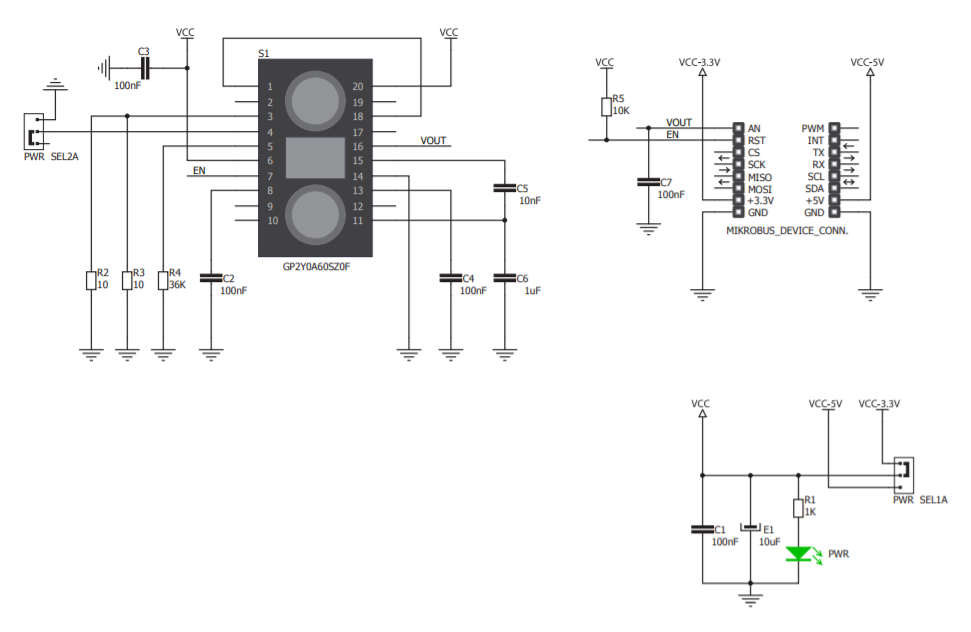


Figure 1.1a: IR Distance Click and Schematic

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

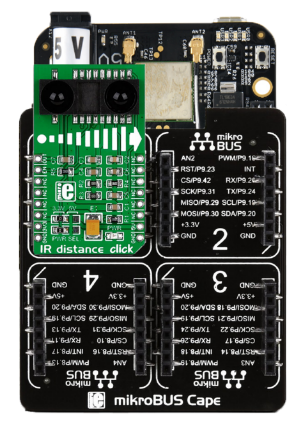


Figure 1.1b: Connecting IR Distance Click to mikroBUS Cape and BBBW Board

## Reading the Analog Data from IR Distance 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 “**IRDistance.py**”.
3. **Double click** on the newly created file “**IRDistance.py**” and enter the following code into the file under the Editor section.

|  |
| --- |
| import time  import Adafruit\_BBIO.ADC as ADC    ADC.setup()  while True:  DigitalValue = ADC.read("P9\_38")  if DigitalValue != 0:  AnalogVoltage = (DigitalValue \* 1.8) \* (2200 / 1200)  DistanceCM = 29.988 \* pow(AnalogVoltage , -1.173)  print("Distance(cm): %f" % DistanceCM)  time.sleep(0.3) |

1. **Click** on the “Run” button located beside the Menu Tab to execute the “**IRDistance.py**” file.
2. **Put** your palm at around 10cm on top of the IR Distance Click. It is observed that the value of the distance value printed at the output console window is approximately 10cm.
3. **Move** away your palm slowly from the IR Distance Click. It is observed that the value of the distance printed at the output console window is getting larger as well.

## Understanding of Flame Click Hardware Connection

**Flame Click** is a fire detection solution. It carries a PT334-6B silicon phototransistor that’s covered in black epoxy and therefore sensitive only to infrared light. To use it as a fire alarm, set up the exact detection threshold through the onboard potentiometer. Otherwise the sensor will output a continuous analog signal.

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



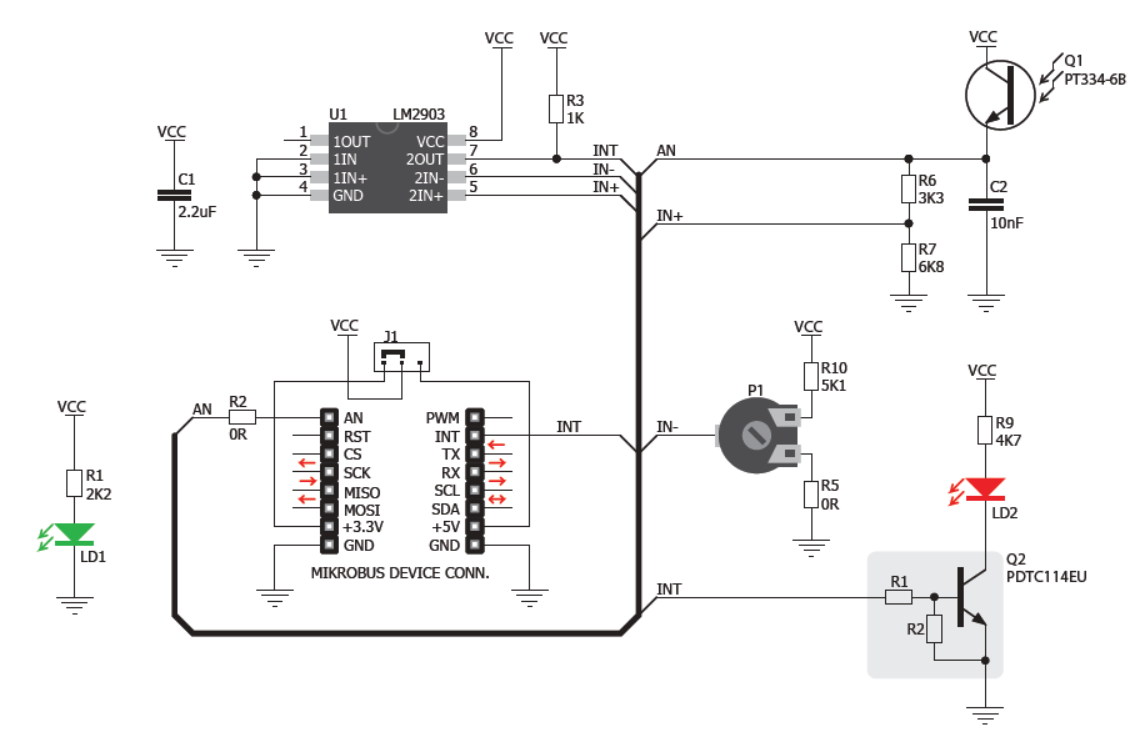


Figure 1.3a: Flame Click and Schematic

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

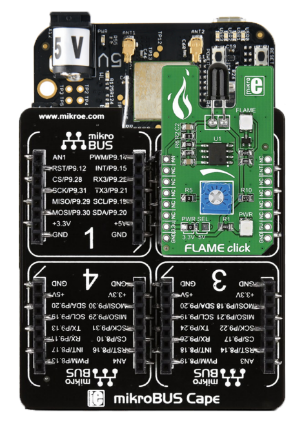


Figure 1.3b: Connecting Flame Click to mikroBUS Cape and BBBW Board

## Reading the Analog Data from Flame 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 “**Flame.py**”.
3. **Double click** on the newly created file “**Flame.py**” and enter the following code into the file under the Editor section.

|  |
| --- |
| import time  import Adafruit\_BBIO.ADC as ADC    ADC.setup()  while True:  DigitalValue = ADC.read("P9\_37")  AnalogVoltage = (DigitalValue \* 1.8) \* (2200 / 1200)  print("Digital Value: %f, Analog Voltage: %f" % (DigitalValue, AnalogVoltage))  time.sleep(0.3) |

1. **Click** on the “Run” button located beside the Menu Tab to execute the “**Flame.py**” file. It is observed that the digital value and analog voltage printed at the output console window are 0.
2. **Tune** the potentiometer to adjust the threshold of the Flame Click.
3. **Light** up a lighter with flame in front of the Flame Click. The red led is turned on when the flame is detected. It is observed that the digital value and analog voltage changes with the amount of flame detected is printed at the output console window.
4. **Move** away the lighter with flame from the Flame Click.The red led is turned off. It is observed that that the digital value and analog voltage printed at the output console window are back to 0.

## Understanding of Mic Click Hardware Connection

**Mic Click** carries the SPQ0410HR5H-B surface mount silicon microphone with maximum RF protection. The SPQ0410HR5H-B is a miniature, high-performance, low power, top port silicon microphone. Using the SiSonic™ MEMS technology, the SPQ0410HR5H-B consists of an acoustic sensor, a low noise input buffer, and an output amplifier. MaxRF protection prevents RF noise in traces from getting into the mic output.

It communicates with the target microcontroller over the analog input pin.

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



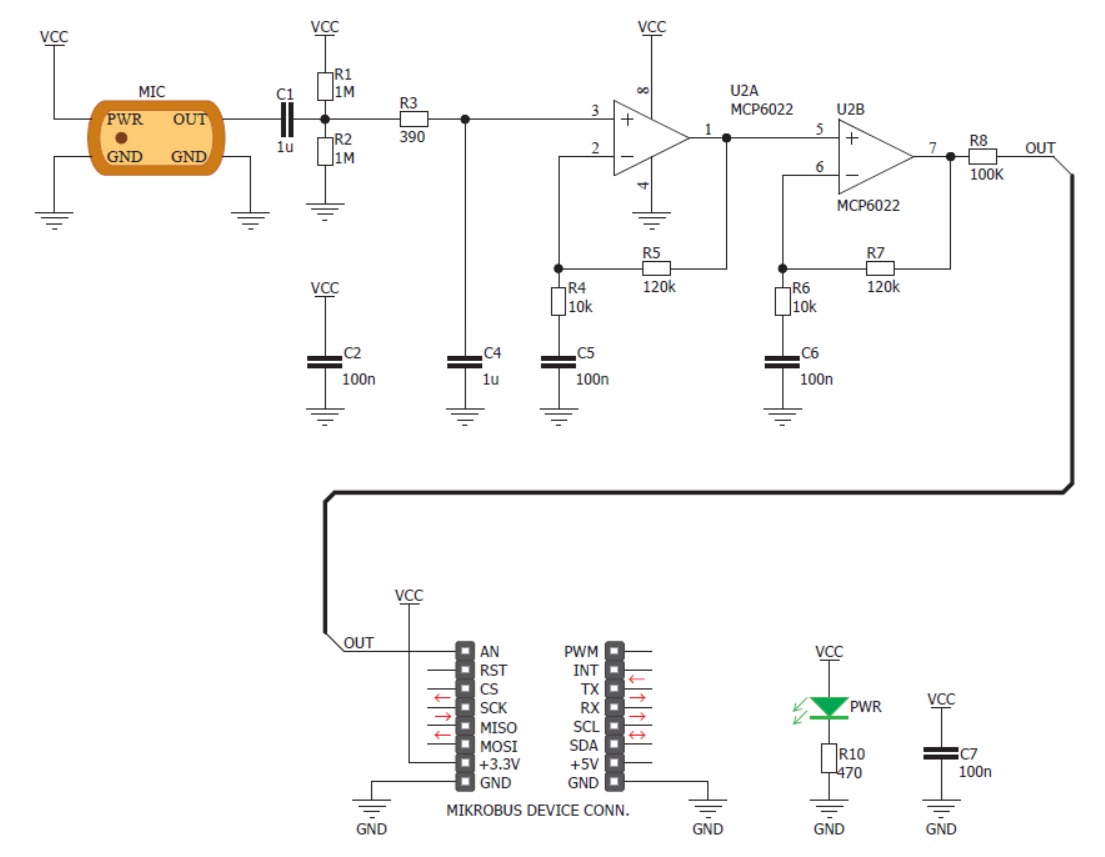


Figure 1.5a: Mic Click and Schematic

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

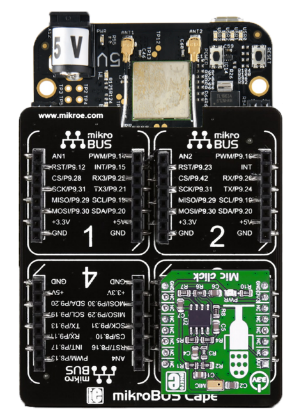


Figure 1.5b: Connecting Mic Click to mikroBUS Cape and BBBW Board

## Reading the Analog Data from Mic 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 “**Mic.py**”.
3. **Double click** on the newly created file “**Mic.py**” and enter the following code into the file under the Editor section.

|  |
| --- |
| import time  import Adafruit\_BBIO.ADC as ADC    ADC.setup()  while True:  DigitalValue = ADC.read("P9\_40")  AnalogVoltage = (DigitalValue \* 1.8) \* (2200 / 1200)  print("Digital Value: %f, Analog Voltage: %f" % (DigitalValue, AnalogVoltage))  time.sleep(0.3) |

1. **Click** on the “Run” button located beside the Menu Tab to execute the “**Mic.py**” file. It is observed that the digital value and analog voltage printed at the output console window are consistent.
2. **Make** some noise in front of the Mic Click.It is observed that the digital value and analog voltage printed at the output console window have a drastic change before settling down.

## Understanding of Force Click Hardware Connection

**Force click** is an add-on board with circuitry for implementing Interlink Electronics’ Force Sensing Resistors with a single zone force sensing resistor included with the click. The Force Sensing Resistor is a thin sensor made of two membranes that are separated by a spacer around the edges. When pressed, the gap between two membranes gets closed. This shorts the two membranes together, with a resistance that is proportional to applied force. The outputted analog voltage is thus a measure of force applied.

The force sensitivity range is from about 0.2N to 20N depending on the implementation.

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



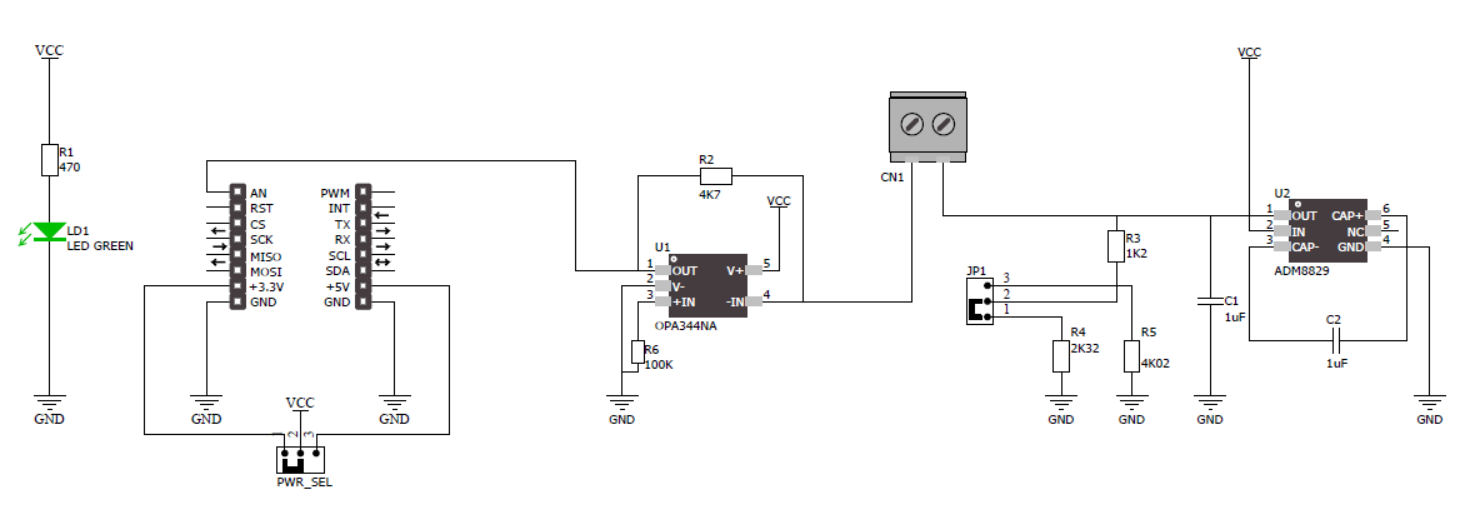


Figure 1.7a: Force Click and Schematic

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

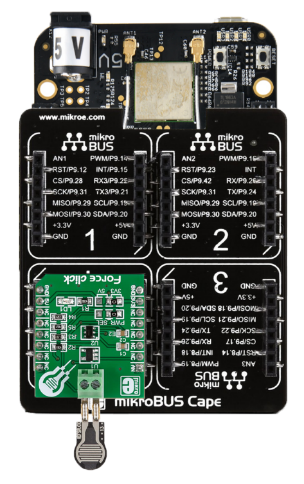


Figure 1.7b: Connecting Force Click to mikroBUS Cape and BBBW Board

## Reading the Analog Data from Force 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 “**Force.py**”.
3. **Double click** on the newly created file “**Force.py**” and enter the following code into the file under the Editor section.

|  |
| --- |
| import time  import Adafruit\_BBIO.ADC as ADC    ADC.setup()  while True:  DigitalValue = ADC.read("P9\_39")  AnalogVoltage = (DigitalValue \* 1.8) \* (2200 / 1200)  print("Digital Value: %f, Analog Voltage: %f" % (DigitalValue, AnalogVoltage))  time.sleep(0.3) |

1. **Click** on the “Run” button located beside the Menu Tab to execute the “**Force.py**” file. It is observed that the digital value and analog voltage printed at the output console window are 0.
2. **Use** 2 fingers to apply different strength on the thin force sensing resistor attached to the Force Click.It is observed that the digital value and analog voltage printed at the output console window are displaying some changes according to the strength applied.

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

## Understanding of Vibro Motor Click Hardware Connection

**Vibro Motor Click** features a compact size Eccentric Rotating Mass (ERM) motor, labeled as C1026B002F. This type of motor is often used for haptic feedback on many small handheld devices, such as the cellphones, pagers, RFID scanners and similar devices. This motor contains a small eccentric weight on its rotor, so while rotating it also produces vibration effect. This kind of motors is sometimes referred to as coin motors, due to its shape.

Besides the vibration motor, the click is also equipped with the DMG3420U, a small MOSFET, which is used to drive the motor. The Vibro Motor Click is an ideal solution for adding a simple, one pin driven haptic feedback on any design.

Vibro Motor Click, and its respective schematic are shown in the Figure below.



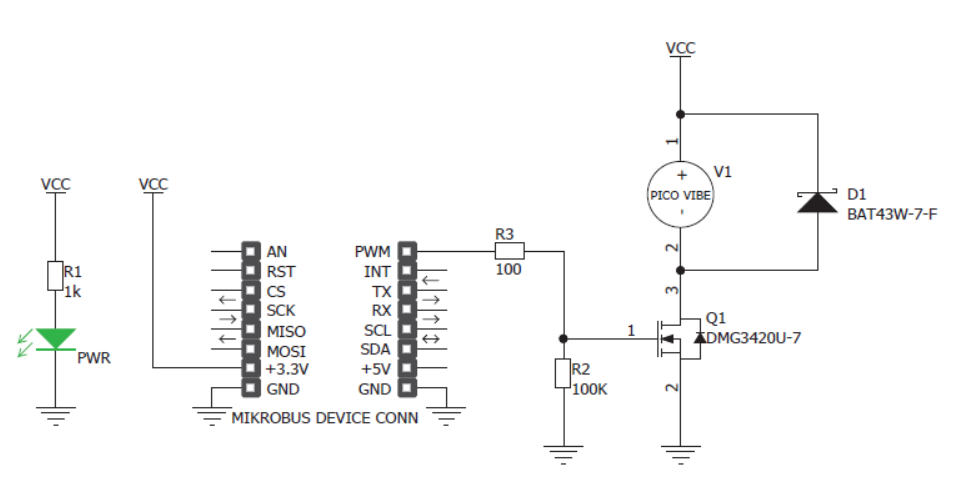


Figure 2.1a: Vibro Motor Click and Schematic

## Controlling the Vibro Motor from 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 “**Vibromotor.py**”.
3. **Double click** on the newly created file “**Vibromotor.py**” and enter the following code into the file under the Editor section.

|  |
| --- |
| import time  import Adafruit\_BBIO.PWM as PWM  PWM.start("P9\_14", 50)  PWM.set\_frequency("P9\_14", 5000)  time.sleep(1)  PWM.set\_frequency("P9\_14", 1)  time.sleep(0.1)  PWM.set\_frequency("P9\_14", 5000)  time.sleep(1)  PWM.set\_frequency("P9\_14", 1)  time.sleep(0.1)  PWM.set\_frequency("P9\_14", 5000)  time.sleep(1)  PWM.set\_frequency("P9\_14", 1)  time.sleep(0.1)  PWM.stop("P9\_14") |

1. **Click** on the “Run” button located beside the Menu Tab to execute the “**Vibromotor.py**” file.
2. **Connect** the Vibro Motor Click to the mikroBUS cape and BBBW board as shown in the Figure below.

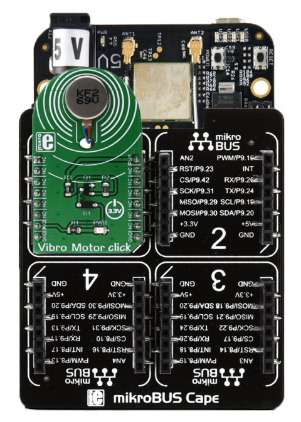


Figure 2.2a: Connecting Vibro Motor Click to mikroBUS Cape and BBBW Board

1. **Click** on the “Run” button located beside the Menu Tab to execute the “**Vibromotor.py**” file again.
2. **Touch** on the Vibro Motor Click and feel the short vibration pulse generated by the Vibro Motor Click.

# **Controlling the Actuator based on Sensor Data using BeagleBone Black Wireless (BBBW) Board**

## Developing an IR Piano using IR Distance and Buzz 2 Click

1. **Connect** both the IR Distance and Buzz 2 Clicks to the mikroBUS cape and BBBW board as shown in the Figure below.

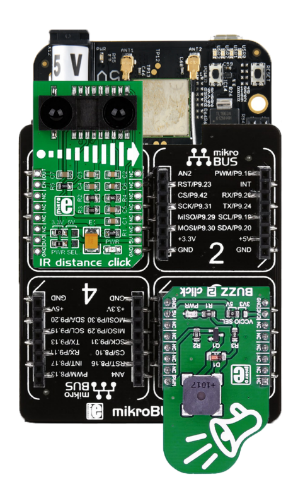


Figure 3.1a: Connecting both the IR Distance and Buzz 2 Click to mikroBUS Cape and BBBW Board

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 “**IRPiano.py**”.
3. **Double click** on the newly created file “**IRPiano.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 Adafruit\_BBIO.ADC as ADC  import Adafruit\_BBIO.PWM as PWM |

1. **Enter** the code to initialize the required peripheral ADC and PWM as shown below.

|  |
| --- |
| ADC.setup()  PWM.start("P8\_19", 50) |

1. **Enter** the main code that reads the user input from IR Distance Click and sound different frequency tones through the Buzz 2 Click as shown below.

|  |
| --- |
| while True:  DigitalValue = ADC.read("P9\_38")  if DigitalValue != 0:  AnalogVoltage = (DigitalValue \* 1.8) \* (2200 / 1200)  DistanceCM = 29.988 \* pow(AnalogVoltage , -1.173)  if DistanceCM > 100:  PWM.stop("P8\_19")  elif DistanceCM > 45:  PWM.start("P8\_19", 50)  PWM.set\_frequency("P8\_19", 1046)  elif DistanceCM > 40:  PWM.start("P8\_19", 50)  PWM.set\_frequency("P8\_19", 988)  elif DistanceCM > 35:  PWM.start("P8\_19", 50)  PWM.set\_frequency("P8\_19", 880)  elif DistanceCM > 30:  PWM.start("P8\_19", 50)  PWM.set\_frequency("P8\_19", 783)  elif DistanceCM > 25:  PWM.start("P8\_19", 50)  PWM.set\_frequency("P8\_19", 698)  elif DistanceCM > 20:  PWM.start("P8\_19", 50)  PWM.set\_frequency("P8\_19", 659)  elif DistanceCM > 15:  PWM.start("P8\_19", 50)  PWM.set\_frequency("P8\_19", 587)  elif DistanceCM > 10:  PWM.start("P8\_19", 50)  PWM.set\_frequency("P8\_19", 523)  time.sleep(0.3) |

1. **Click** on the “Run” button located beside the Menu Tab to execute the “**IRPiano.py**” file.
2. **Put** your palm at 10cm on top of the IR Distance Clickand **move** your palm away slowly.
3. **Listen** to the frequency tones generated by the Buzz 2 Click.
4. **Observe** and **compare** the program output with your teammates and consult your lecturer for advice if it is not the same.
5. **Write** the observation in the white box below for future reference if needed.

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

## Tinkering Time

1. Together with a classmate or two, **think** of a simple application that can use at least 2 of the 8 digital and analog sensors clicks to capture an event and alert the user through either Buzz 2 or Vibromotor Click. **Try** to use other clicks you have learned before to complement your application.
2. **Connect** the selected clicks to the mikroBUS cape and BBBW board.
3. **Create** a python file in Cloud9 IDE and start writing your code.
4. **Present** your complete work to your lecturer for advice.
5. **Share** your work with your other classmates and teach them how you do it if they are interested.

*Congratulations! You have successfully completed the Lab4b. Good job! Take a good break and stay tune for next lab. More exciting lab exercises coming to you!*