

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

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

**Laboratory No**. : Lab 4a

**Laboratory Title** : Sensor and Actuator: Reading the Digital 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

: Motion Click x1

: Vibra Sense Click x1

: IR Eclipse Click x1

: Reed Click x1

: Buzz 2 Click x1

**Contents**

1. Reading the Digital Sensors Data using BeagleBone Black Wireless (BBBW) Board
   1. Understanding of Motion Click Hardware Connection
   2. Reading Digital Data from Motion Click using Python Code
   3. Understanding of Vibra Sense Click Hardware Connection
   4. Reading Digital Data from Vibra Sense Click using Python Code
   5. Understanding of IR Eclipse Click Hardware Connection
   6. Reading Digital Data from IR Eclipse Click using Python Code
   7. Understanding of Reed Click Hardware Connection
   8. Reading Digital Data from Reed Click using Python Code
2. Controlling the Actuator using BeagleBone Black Wireless (BBBW) Board
   1. Understanding of Buzz 2 Click Hardware Connection
   2. Controlling the Buzz 2 Click using Python Code
3. Controlling the Actuator based on Digital Sensor Data using BeagleBone Black Wireless (BBBW) Board
   1. Developing a Magnetic Door Alert System using Reed and Buzz 2 Click
   2. Tinkering Time

# **Reading the Digital Sensor Data using BeagleBone Black Wireless (BBBW) Board**

## Understanding of Motion Click Hardware Connection

**Motion Click** is a motion detector sensitive only to live bodies. It carries PIR500B, a pyroelectric sensor which generates a voltage when exposed to infrared radiation emitted by live bodies. The white plastic Fresnel lens covering the sensor filters visible light. The signal is processed by a BISS0001 PIR sensor controller which sends an interrupt to the MCU. An onboard potentiometer lets you adjust the detecting range of the sensor (up to 1.7 meters).

Motion Click is ideal for alarm systems, light switch controllers, and similar systems where human presence needs to be detected.

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



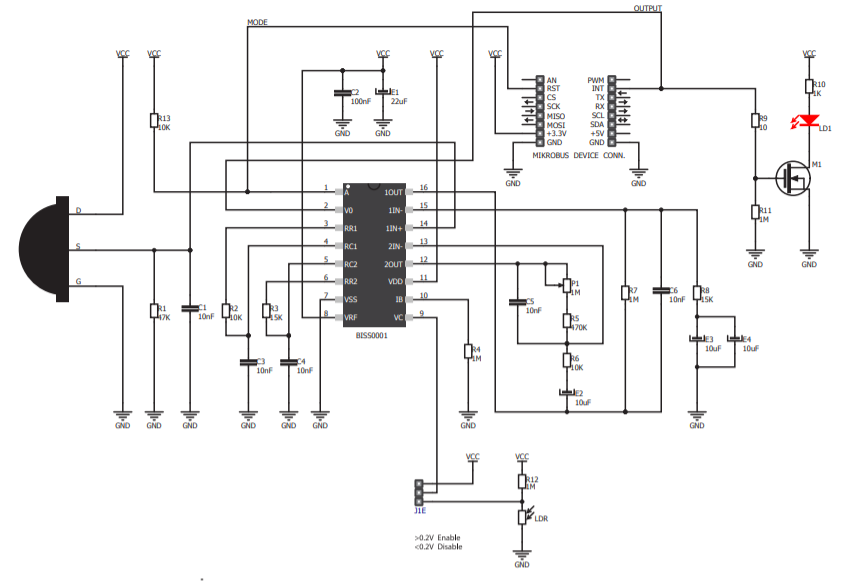


Figure 1.1a: Motion Click and Schematic

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

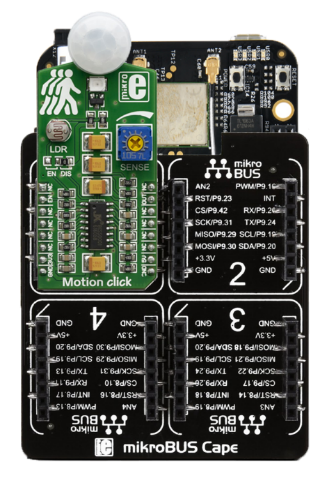


Figure 1.1b: Connecting Motion Click to mikroBUS Cape and BBBW Board

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

|  |
| --- |
| import time  import Adafruit\_BBIO.GPIO as GPIO  GPIO.setup("P9\_15", GPIO.IN)  while True:  if GPIO.input("P9\_15"):  print("Motion is Detected")  else:  print("No Motion is Detected")  time.sleep(0.3) |

1. **Click** on the “Run” button located beside the Menu Tab to execute the “**Motion.py**” file. It is observed that text “**No Motion is Detected**” is printed at the output console window.
2. **Wave** your hand on top of the Motion Click and it is observed that the red Led on the Motion Click is turned on and the text “**Motion is Detected**” is printed at the output console window.
3. **Stop** waving your handand it is observed the red led is turned off and the text “**No** **Motion is Detected**” is printed again at the output console window.

## Understanding of Vibra Sense Click Hardware Connection

**Vibra Sense Click** is a low-cost micro shock vibration sensor with a digital output which can be set as an Interrupt. An onboard potentiometer lets you set the interrupt threshold. It’s a very simple and effective solution for detecting vibrations. The sensing unit consists of a round housing in which a spring is coiled around a metal pin. When exposed to vibration, the spring contacts the pin and closes the switch.

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



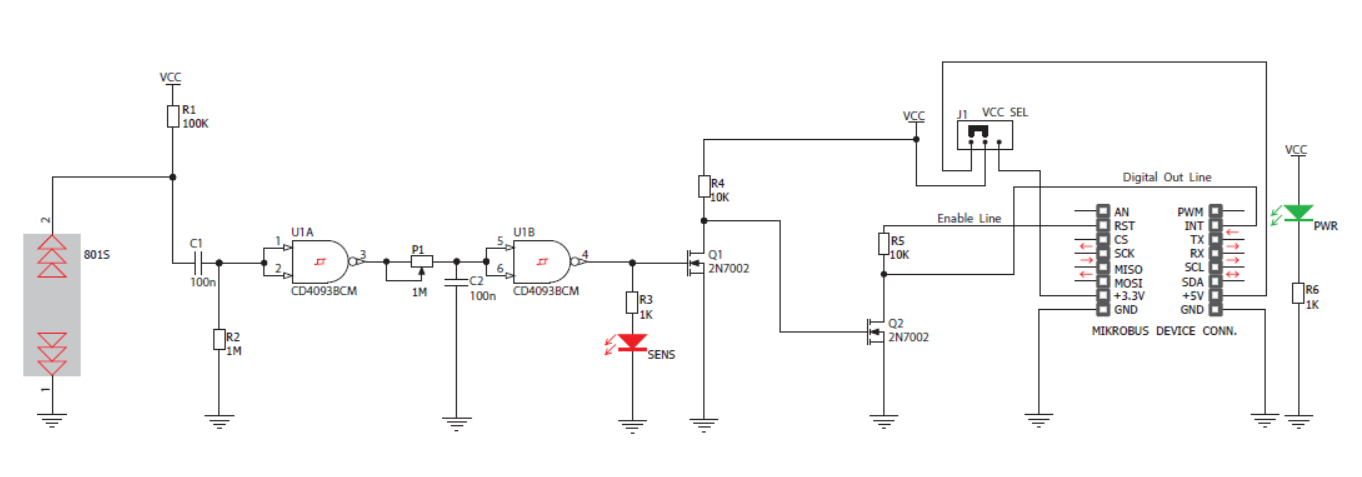


Figure 1.3a: Vibra Sense Click and Schematic

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



Figure 1.3b: Connecting Vibra Sense Click to mikroBUS Cape and BBBW board

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

|  |
| --- |
| import time  import Adafruit\_BBIO.GPIO as GPIO  GPIO.setup("P9\_23", GPIO.OUT)  GPIO.setup("P9\_41", GPIO.IN)  GPIO.output("P9\_23", GPIO.HIGH)  while True:  if GPIO.input("P9\_41"):  print("Vibration is Detected")  else:  print("No Vibration is Detected")  time.sleep(0.3) |

1. **Click** on the “Run” button located beside the Menu Tab to execute the “**VibraSense.py**” file. It is observed that the text “**No** **Vibration is Detected**” is printed at the output at the output console window.
2. **Tune** the Potentiometer on the Vibra Sense Click labelled as P1 towards clockwise to increase the sensitivity of the vibration sensor.
3. **Knock** once on the table where the Vibra Sense Click is seated**.** It is observed that the red led is turned on and the text “**Vibration is Detected**” is printed at the output console window.
4. It is also observed that the text “**No** **Vibration is Detected**” is printed again at the output console window and the red led is turned off once the knock is over.

## Understanding of IR Eclipse Click Hardware Connection

**IR Eclipse Click** carries an EE-SX198 photo interrupter sensor. This sensor consists of an infrared transmitter and receiver facing each other and spaced apart by a 3mm slit. When the beam from the transmitter is eclipsed by placing an object in the slit (like a piece of paper), the sensor is activated (indicated by the onboard INT LED). This type of sensor is typically used in printers, copiers, vending machines and so forth.

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



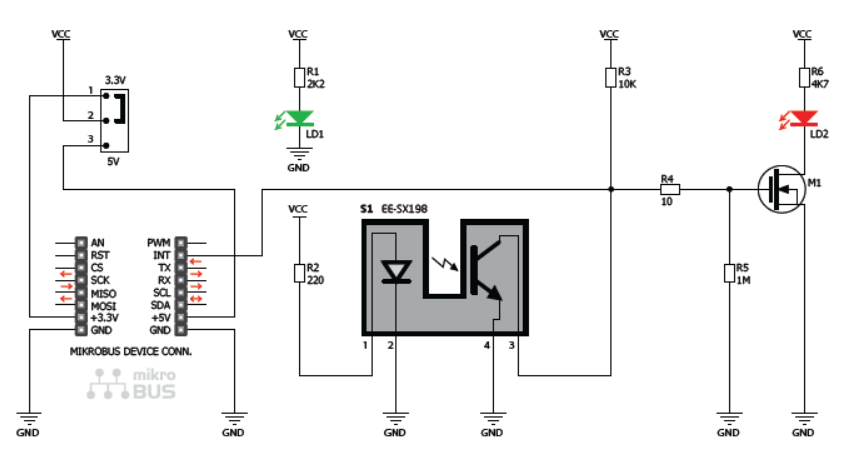


Figure 1.5a: IR Eclipse Click and Schematic

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

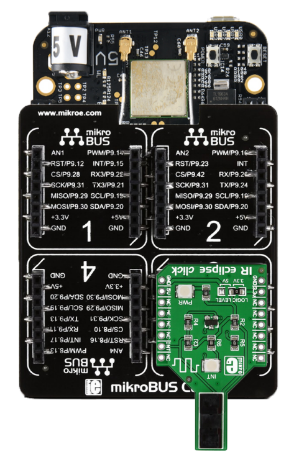


Figure 1.5b: Connecting IR Eclipse Click to mikroBUS Cape and BBBW Board

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

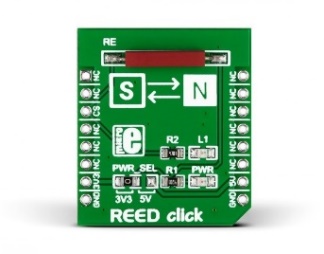
|  |
| --- |
| import time  import Adafruit\_BBIO.GPIO as GPIO  GPIO.setup("P8\_18", GPIO.IN)  while True:  if GPIO.input("P8\_18"):  print("Paper is Detected")  else:  print("No Paper is Detected")  time.sleep(0.3) |

1. **Click** on the “Run” button located beside the Menu Tab to execute the “**IREclipse.py**” file. It is observed that the text “**No Paper is Detected**” is printed at the output console window.
2. **Put** a piece of paper into the slit on the IR Eclipse Click. It is observed that the red led is turned on and the text “**Paper is Detected**” is printed at the output console window.
3. **Take** away the paper and it is observed that the red led is turned off and the text “**No Paper is Detected**” is printed again at the output console window.

## Understanding of Reed Click Hardware Connection

**Reed Click** is a simple board that carries a standard (Single Pole Single Throw Normally Open) reed switch. A reed switch comprises of two thin magnetic contacts sealed inside a casing. One contact is a magnetic north pole, the other a south. The two contacts are separate, until a magnetic field is applied which snaps them close, activating the switch.

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



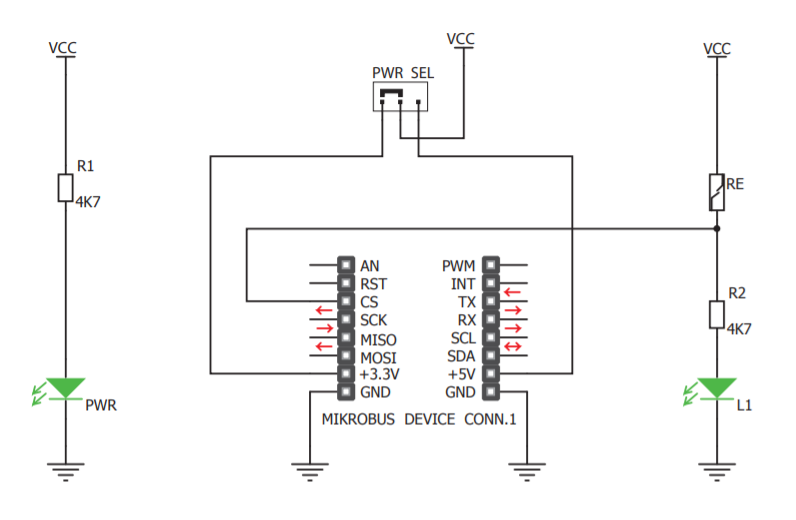


Figure 1.7a: Reed Click and Schematic

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

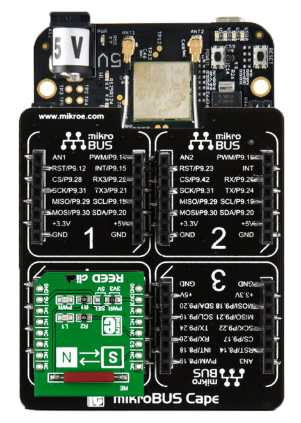


Figure 1.7b: Connecting Reed Click to mikroBUS Cape and BBBW Board

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

|  |
| --- |
| import time  import Adafruit\_BBIO.GPIO as GPIO  GPIO.setup("P8\_10", GPIO.IN)  while True:  if GPIO.input("P8\_10"):  print("Magnet is Detected")  else:  print("No Magnet is Detected")  time.sleep(0.3) |

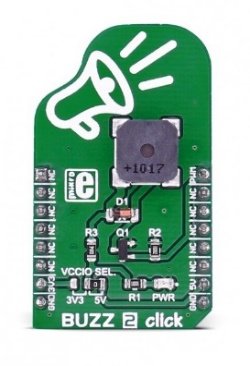
1. **Click** on the “Run” button located beside the Menu Tab to execute the “**Reed.py**” file. It is observed that the text “**No** **Magnet is Detected**” is printed at the output console window.
2. **Bring** a magnet given by the instructor closer to either the south or north pole of the reed switch located on the Reed Click. It is observed that the yellow led is turned on and the text “**Magnet is Detected**” is printed at the output console window.
3. **Bring** away the magnet from the reed switch. It is observed that the yellow led is turned off and the text “**No** **Magnet is Detected**” is printed again at the output console window.

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

## Understanding of Buzz 2 Click Hardware Connection

**Buzz 2 Click** carries the CMT-8540S-SMT magnetic buzzer transducer. The buzzer’s resonant frequency is 4kHz. The CMT-8540S-SMT magnetic buzzer is controlled by the PWM signal where the signal frequency determines the sound pitch, and the duty cycle determines the sound volume.

Buzz 2 Click, and its respective schematic are shown in the Figure below.



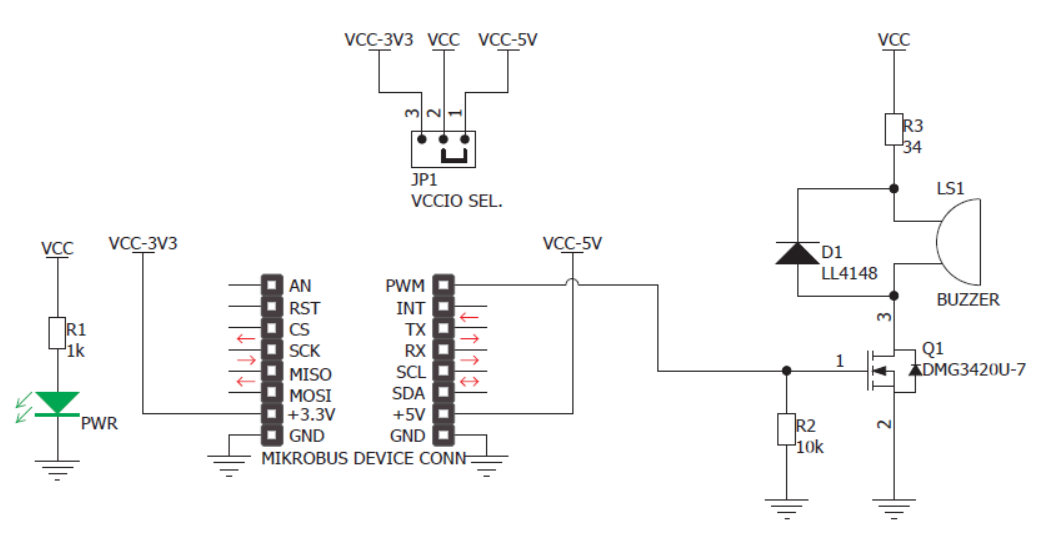


Figure 2.1a: Buzz 2 Click and Schematic

1. **Connect** the Buzz 2 Click to the mikroBUS cape and BBBW board as shown in Figure below.

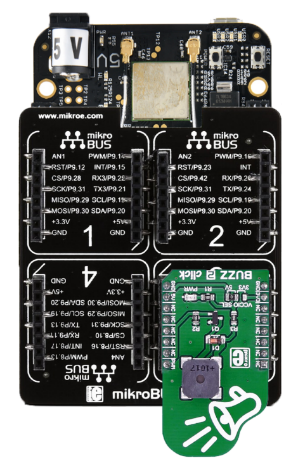


Figure 2.1b: Connecting Buzz 2 Click to mikroBUS Cape and BBBW Board

## Controlling the Buzz 2 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 “**Buzz2.py**”.
3. **Double click** on the newly created file “**Buzz2.py**” and enter the following code into the file under the Editor section.

|  |
| --- |
| import time  import Adafruit\_BBIO.PWM as PWM  PWM.start("P8\_19", 50)  PWM.set\_frequency("P8\_19", 523)  time.sleep(0.5)  PWM.set\_frequency("P8\_19", 587)  time.sleep(0.5)  PWM.set\_frequency("P8\_19", 659)  time.sleep(0.5)  PWM.stop("P8\_19") |

1. **Click** on the “Run” button located beside the Menu Tab to execute the “**Buzz2.py**” file.
2. **Listen** to the frequency tones of “Do Re Me” played by the Buzz 2 Click.

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

## Developing a Magnetic Door Alert System using Reed and Buzz 2 Click

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

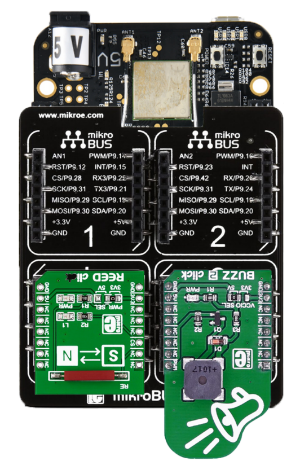


Figure 3.1a: Connecting both the Reed and Buzz 2 Click to mikroBUS Cape and BBBW Board

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

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

|  |
| --- |
| GPIO.setup("P8\_10", GPIO.IN)  PWM.start("P8\_19", 50) |

1. Enter the main code that reads the sensor output from Reed Click and sound the Buzz 2 Click as shown below.

|  |
| --- |
| while True:  if GPIO.input("P8\_10"):  PWM.stop("P8\_19")  else:  PWM.start("P8\_19", 50)  PWM.set\_frequency("P8\_19", 1000)  time.sleep(0.1)  PWM.set\_frequency("P8\_19", 2000)  time.sleep(0.1)  time.sleep(0.3) |

1. **Click** on the “Run” button located beside the Menu Tab to execute the “**MagDoorAlertSystem.py**” file.
2. The buzzer from the Buzz 2 Click will sound due to the absent of the magnet.
3. **Bring** a magnet to closer to either the north of south pole of the reed switch on the Reed Clickto stop the sound from buzzer.
4. **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 1 of the 4 digital sensors clicks to capture an event and alert the user through Buzz 2 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 Lab4a. Good job! Take a good break and stay tune for next lab. More exciting lab exercises coming to you!*