Computer Input using Arduino

❖ Introduction:

Microcontrollers (like arduino and raspberry pi) are very powerful tools to carry out cool projects including motion detection, fetching metric/distance data using Ultrasonic sensors, humidity sensor, and what not. But this domain of creativity can further expand to a new regime if the data from microcontrollers can be directly sent to a computer and then manipulated to perform a certain task. This article aims to describe one such method of data transfer, by establishing serial communication between Python and Arduino. Further, some libraries of python for automation, data representation, and data manipulation (which might come handy for carrying out projects in general) are described.

❖ Requirements

1. Microcontroller(arduino uno is described in this article)
2. Python3 (IDE preferred but not required)
3. Arduino IDE
4. Accessories according to project (like ultrasonic sensor for fetching distance)

★ Python Installation (and required libraries)

If python is not already installed installed on your computer below is link for guide to setup python o respective OS:
https://realpython.com/installing-python/

Although it is not necessary, but using an IDE will make things easy for programming and installing libraries. Below are links for guide to setup PyCharm and Jupyter Notebook:
PyCharm:https://www.guru99.com/how-to-install-python.html
Jupyter:https://jupyter.org/

In case you are not familiar with programming in python below are tutorial videos (playlist) which cover basics of python:
https://youtu.be/QXeEoD0pB3E

To establish serial communication between arduino and python there is an easy to use library named “PySerial”.
To install the library from terminal/command prompt use this command:
“pip install pyserial”

To import this library in the Python program use:
“import serial”

Now we need to install arduino IDE to proceed further.

★ Arduino IDE

Arduino IDE is required to export code to the arduino. Programming for arduino is similar to C++ with some modifications. Basics of programming in C++ is enough for this article.
Below is a link to install arduino IDE:
https://www.arduino.cc/en/software

If you are not familiar with using arduino, then below is a link to a video tutorial for learning the basics for the same.
https://youtu.be/NfvAxbu_SyI

After connecting the arduino to the computer, the PORT id of connection is required to communicate with python. It can be found via the “port” section of the “tools” tab of arduino IDE.

★ Arduino program

Arduino can communicate any information to python through serial communication. Therefore, using
“Serial.println(10)”
will pass 10 (however, as a string) to python instead of printing it in the serial monitor.
A simple program on arduino side can be:

```
//Program that prints “HIGH” when button is pressed
//and “LOW” when the button is not pressed.
void setup()
{
  pinMode(3,INPUT);
  digitalWrite(3,LOW);
  Serial.begin(9600);
}

void loop()
{
  if(digitalRead(3)==HIGH){
        Serial.println(1,DEC);
  }else{
        Serial.println(0,DEC);
  }

}
```

★ Python program

To use the “pyserial” library the baud rate and port of connection is required(as found earlier).

Note:Baud rate must be the same for python program and arduino program!!

A simple python program corresponding to the arduino program can be:

```
import serial

ard=serial.Serial("COM3",timeout=1,baudrate=9600)

while True:
    cc=str(ard.readline())
    z=cc[2:][:-5]
    if(z=="1"):
        print("The button is pressed")

    if(z=="0"):
        print("The button is not pressed")
```

In the above program, “COM3” is the port of connection of arduino and “9600” is the baud rate.Usually, a timeout of 1 second(0.1 secs is enough though) is suggested to give time for establishing the connection.
Note: The data from arduino might contain some unnecessary characters and to retrieve the required information, array slicing can be performed
This is how data from arduino can be fetched by python.

Data can even be sent to arduino using :
“ard.write(bytes(x,’utf-8’))”

and can be fetched from arduino program using:
“Serial.read()”

After being able to carry out the above procedure, what is left to exercise is your creativity.

Below are some libraries of python which can be used to carry out interesting projects!

1. Pyautogui

It is an easy to use library to carry out any automation process on the computer, be it keyboard strokes, mouse click, dragging mouse, clicking on a region having a given set of pixels….and a lot more.
Documentation: https://pyautogui.readthedocs.io/en/latest/

2. Numpy

This library is mainly used for array manipulation, and turns out to be ‘necessity’ for any scientific computation. The data fetched from arduino (say, distance from ultrasonic sensor) can be stored and manipulated in an array and the data can even be made into an output file.
Documentation: https://numpy.org/

3. Matplotlib

This library is often used for data representation, be it histograms, bar graphs, pie charts, and a lot more.
Documentation: https://matplotlib.org/
Video tutorial: https://youtu.be/yZTBMMdPOww

Codes for my project(window change upon motion detection):

❖ Arduino Side:

```
int ECHO=11;
int TRIG=10;
float a,b,A;

void setup() {
  pinMode(12, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, INPUT);
  Serial.begin(9600);
  a=getDistance();
  b=a;
  }

float getDistance(){
  float time;
  digitalWrite(TRIG, LOW);
  delayMicroseconds(10);
  digitalWrite(TRIG, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG, LOW);


  time=pulseIn(ECHO,HIGH);

  return time*170*0.000001;
}

bool check1(){
  bool c=true;
  for(int i=0;i<5;i++){
    if(getDistance()>2){c=false;break;}
    delay(100);
    }
   return c;
  }

 bool check2(){
  bool c=true;
  for(int i=0;i<5;i++){
```

```
    if(getDistance()<2){c=false;break;}
    delay(100);
    }
   return c;
  }



void loop() {
  while(true){
    if(check1()){break;}
    delay(100);
    }

    Serial.println(1,DEC);

    digitalWrite(12,HIGH);
    delay(500);
    digitalWrite(12,LOW);

   while(true){
      if(check2()){break;}
      delay(100);
      }
   Serial.println(0,DEC);

   delay(100);

    }
```

- Python side:

```
import pyautogui as auto
import time
```

```
import serial

time.sleep(1)

ard=serial.Serial("COM3",timeout=1,baudrate=9600)

while True:
   cc=str(ard.readline())
   z=cc[2:][:-5]
   if(z=="1"):
       #print("yes")
       auto.hotkey('winleft','tab')
       time.sleep(0.3)
       auto.click(800,250)

   if(z=="0"):
       #print("back")
       auto.hotkey('winleft', 'tab')
       time.sleep(0.3)
       auto.click(600, 250)
```

Note: The print statements can be removed once the debugging is done