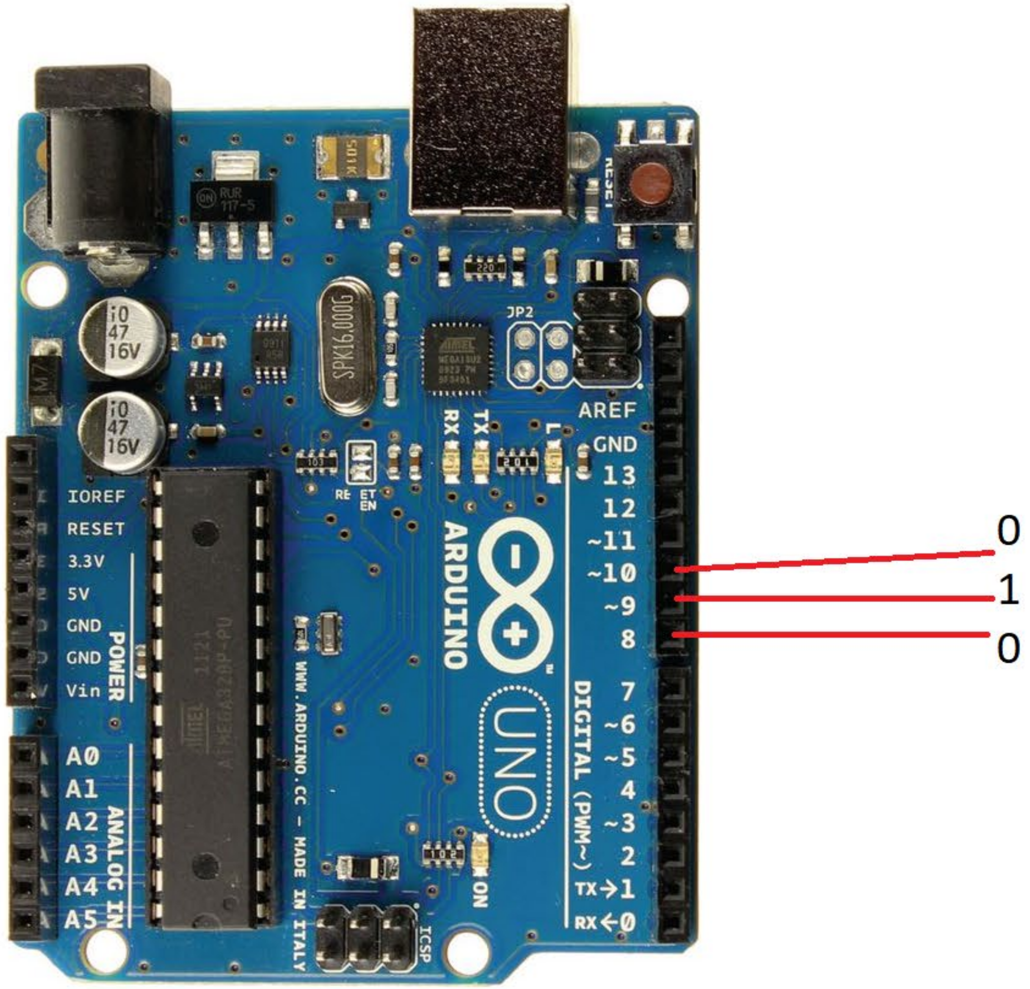
Serial communication between components

Using the input pins themselves as bits (aka parallel)

Pros:
-Simple and intuitive to use
All that is needed to do is read each pin state

Cons:
-Uses up more pins as more bits are needed
-Code can look convoluted
ex: if digitalRead(8)||digitalRead(9)||digitalRead(10)
sure we can store the state in a variable to have it look cleaner
ex2: if (pin1||pin2||pin3) ==1

-Limited to reading 2^n input combinations
n=pins dedicated to be read only



Using serial

Pros:

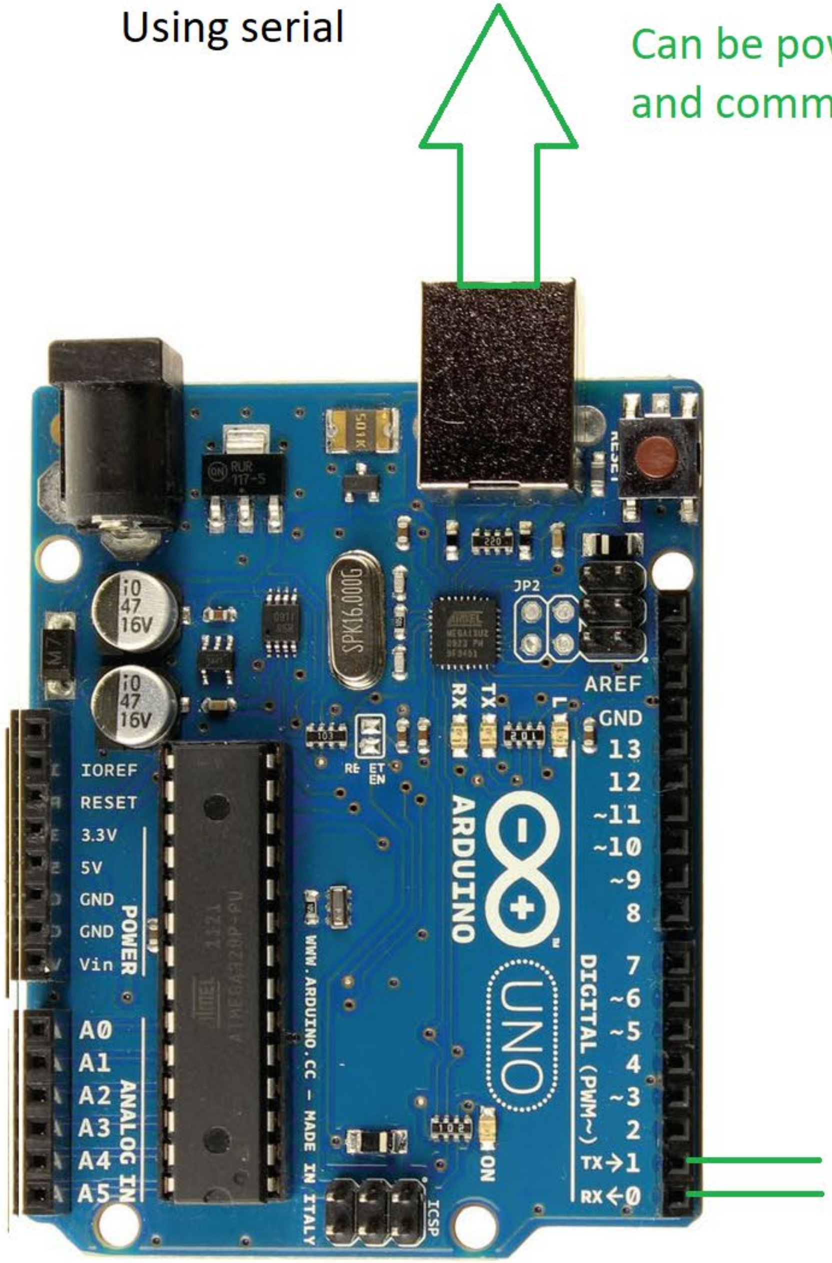
-Only requires 2 pins to read many inputs

-Don't really need to use pins 0 and 1 to communicate, can also use USB communication which is more convenient when using a raspberry pi.

-Eliminates the need of additional wires

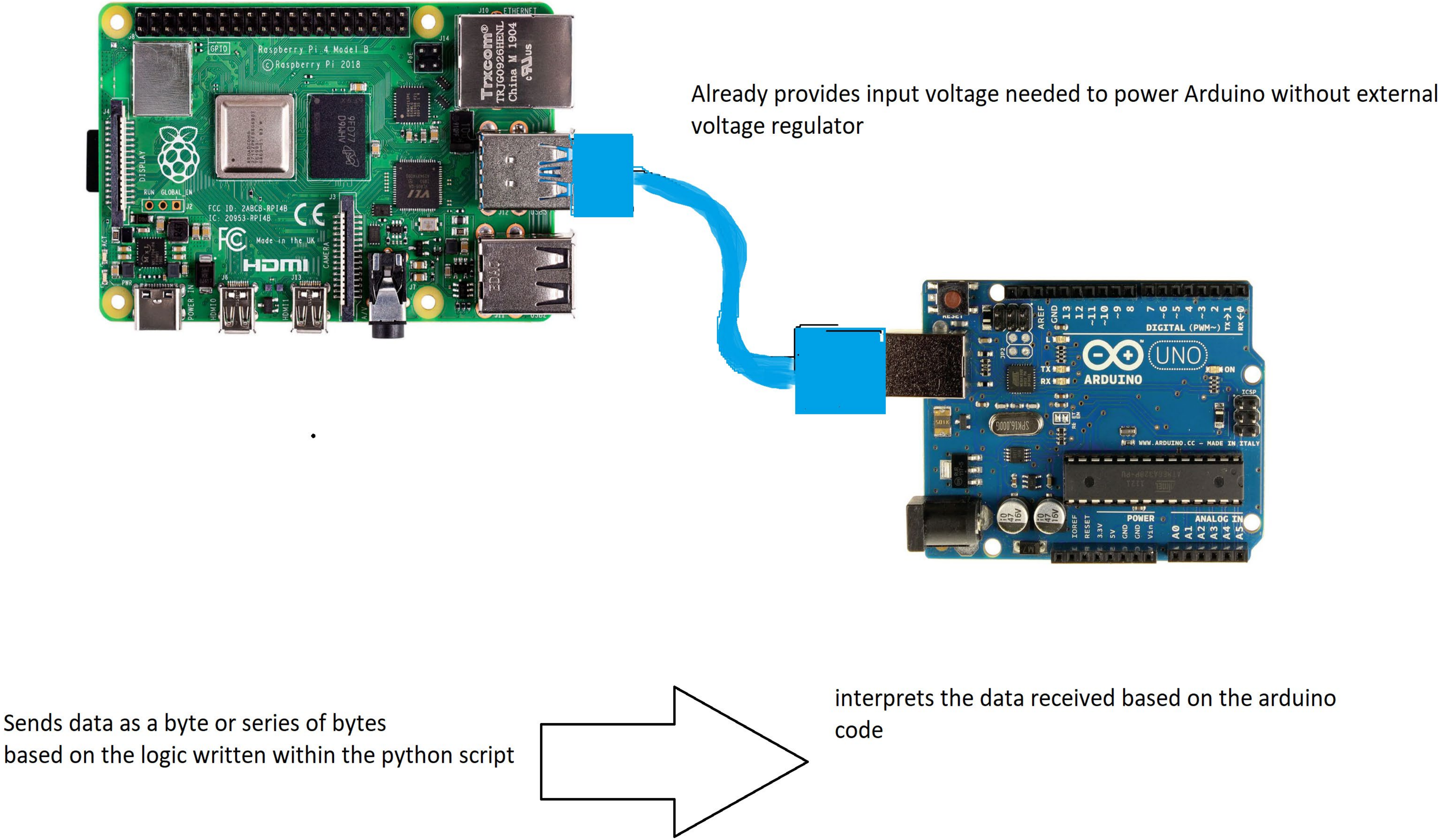
Cons:

-Pins 0 and 1 cannot be used for anything if Serial communication is used



Can be powered by USB and communicate through it

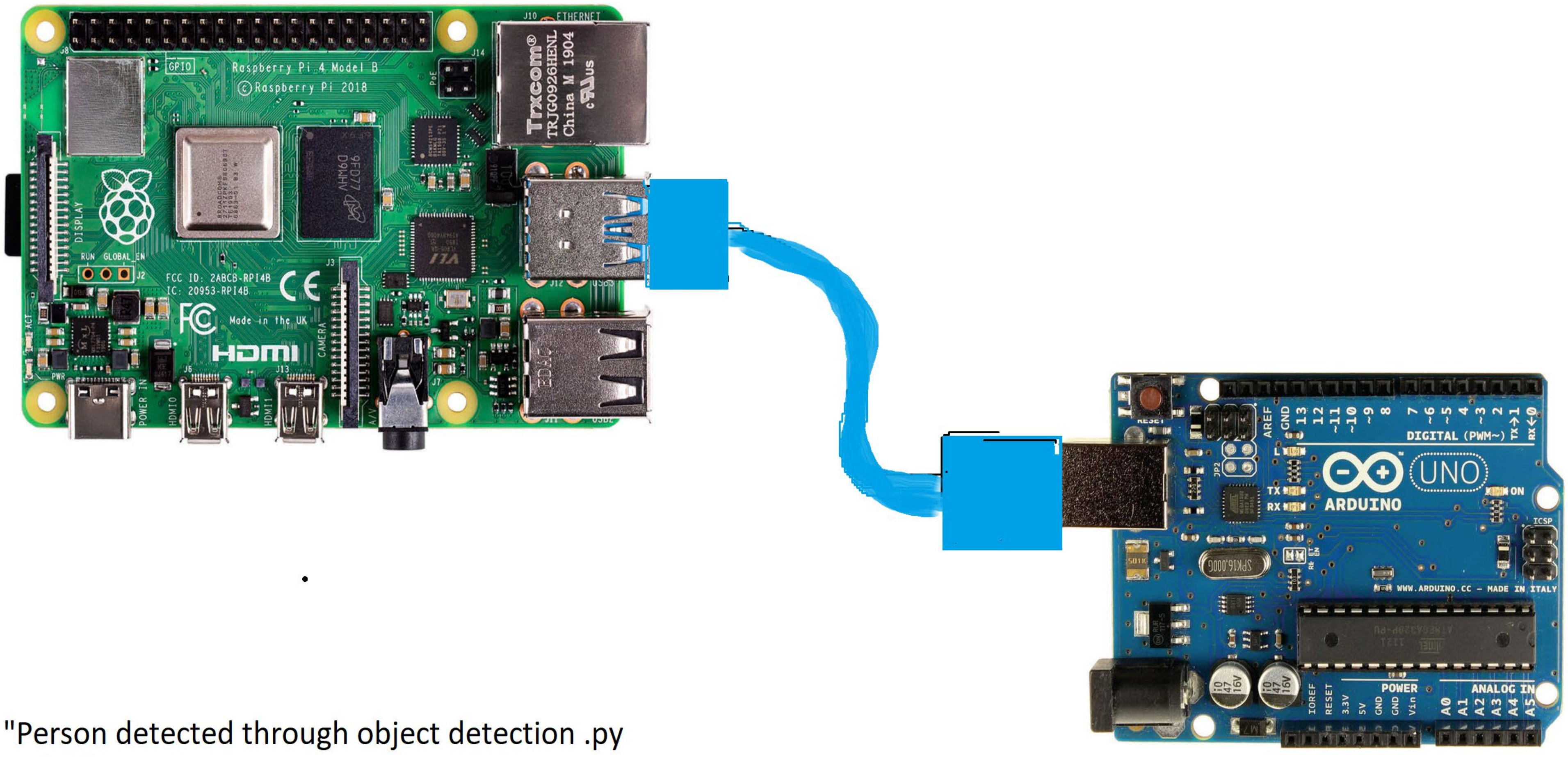
Can't use these pins for anything other than serial



Already provides input voltage needed to power Arduino without external voltage regulator

Sends data as a byte or series of bytes
based on the logic written within the python script

interprets the data received based on the arduino code

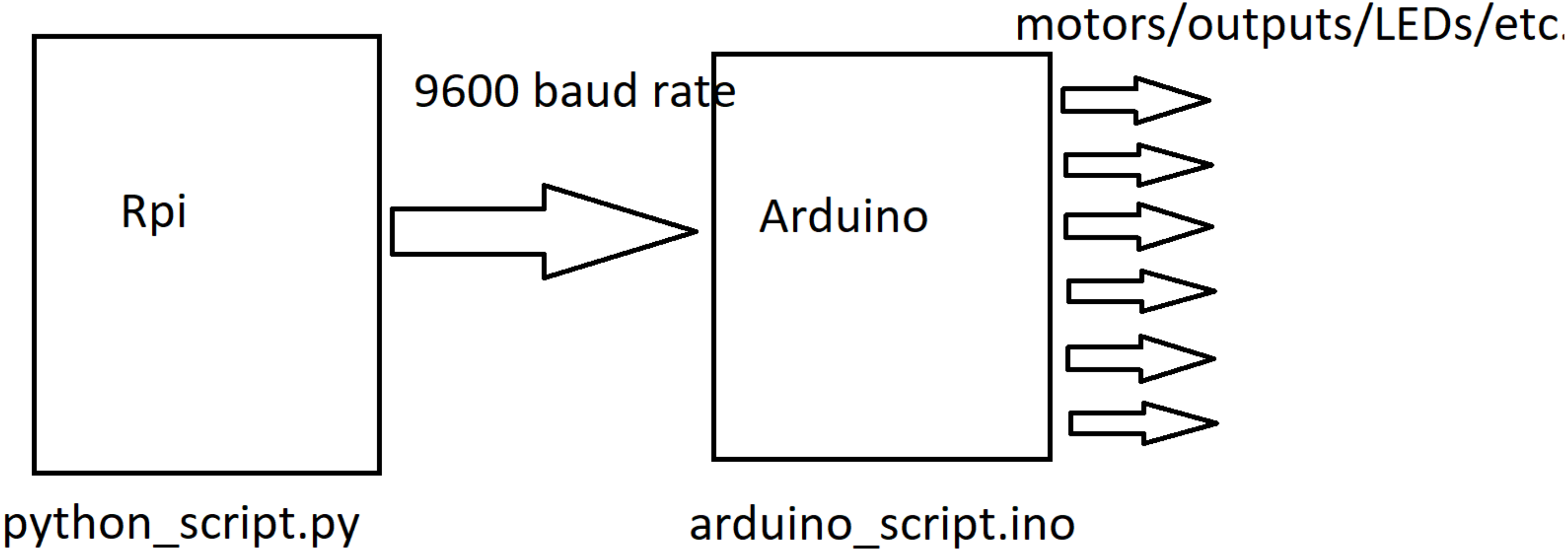


- "Person detected through object detection .py script"
- send b'0' through serial

- if b'0' is received move arm in a waving motion

Demo

- https://www.youtube.com/watch?v=6ESwdYmw_Lk



motors/outputs/LEDs/etc.

9600 baud rate

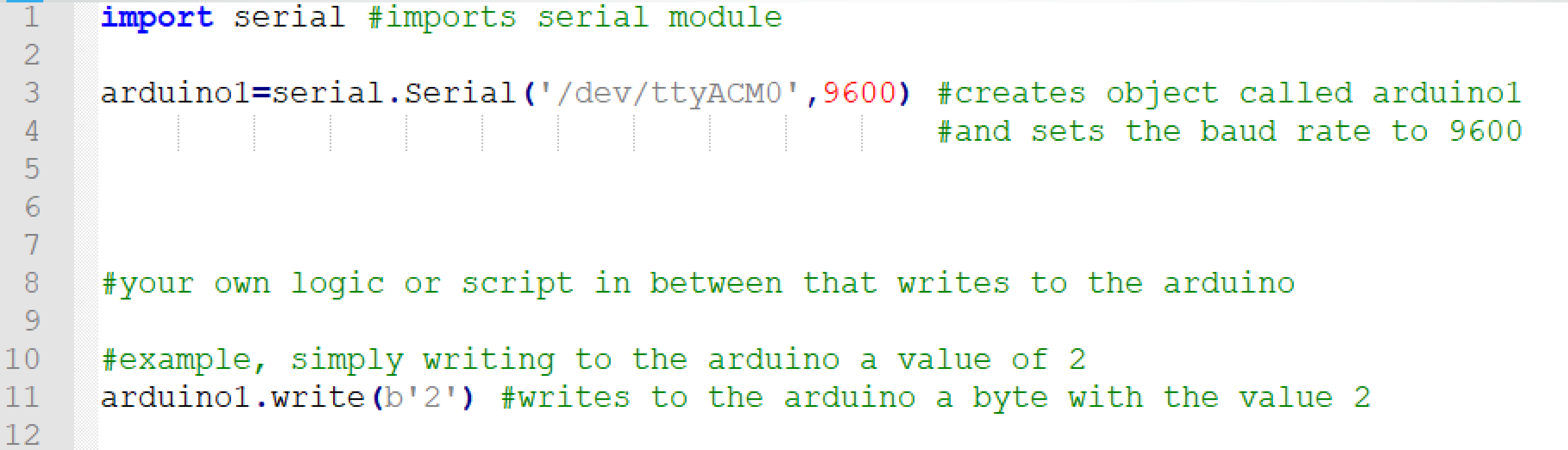
Rpi

Arduino

python_script.py

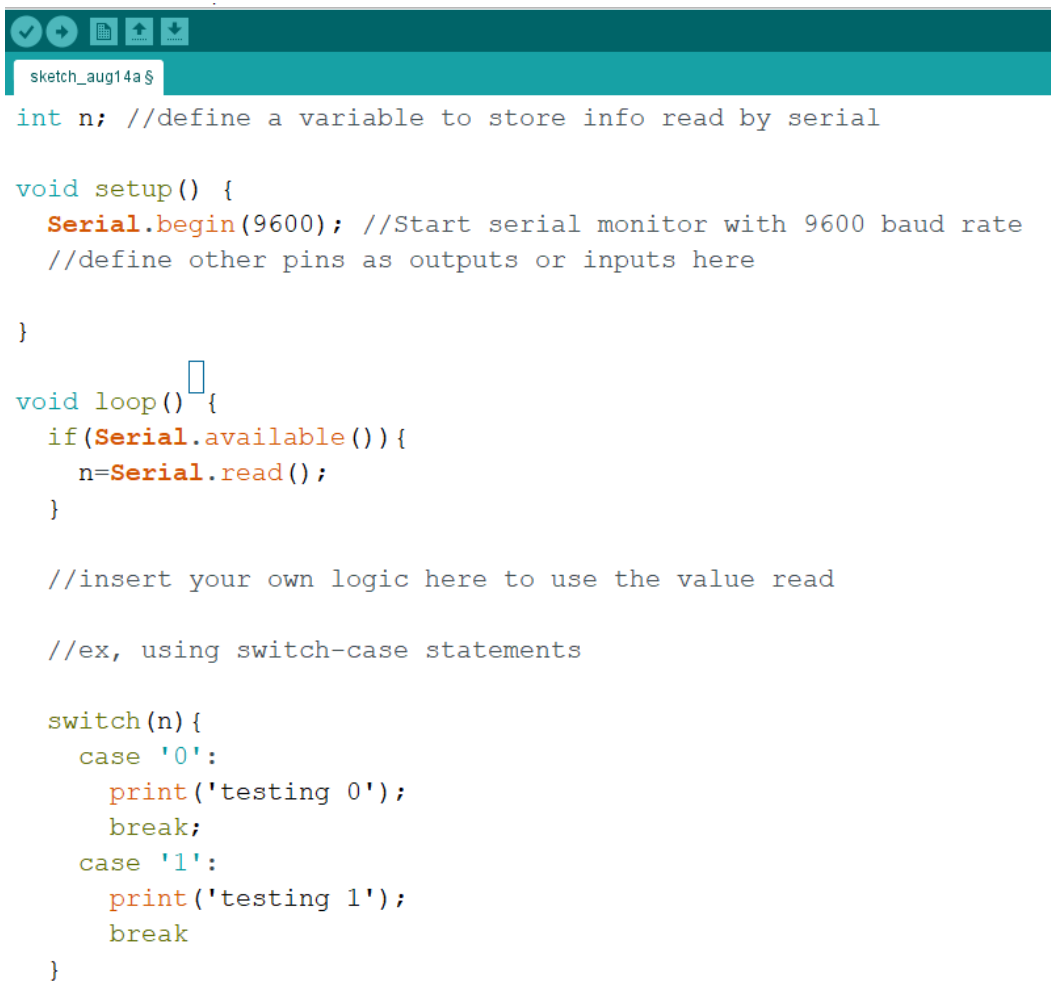
arduino_script.ino

Basic components of communication: Raspberry pi send



```
import serial #imports serial module

arduino1=serial.Serial('/dev/ttyACM0',9600) #creates object called arduino1
                                            #and sets the baud rate to 9600



#your own logic or script in between that writes to the arduino

#example, simply writing to the arduino a value of 2
arduino1.write(b'2') #writes to the arduino a byte with the value 2

```

Arduino read



```
int n; //define a variable to store info read by serial

void setup() {
  Serial.begin(9600); //Start serial monitor with 9600 baud rate
  //define other pins as outputs or inputs here

}

void loop() {
  if(Serial.available()){
    n=Serial.read();
  }

  //insert your own logic here to use the value read

  //ex, using switch-case statements

  switch(n){
    case '0':
      print('testing 0');
      break;
    case '1':
      print('testing 1');
      break
  }
```

Resources to learn more:

- Arduino specific language reference https://www.arduino.cc/reference/en/language/functions/communication/serial/
- How it works https://learn.sparkfun.com/tutorials/serial-communication/serial-intro
- Examples using Arduino and Raspberry Pi https://classes.engineering.wustl.edu/ese205/core/index.php?title=Serial_Communication_between_Raspberry_Pi_%26_Arduino