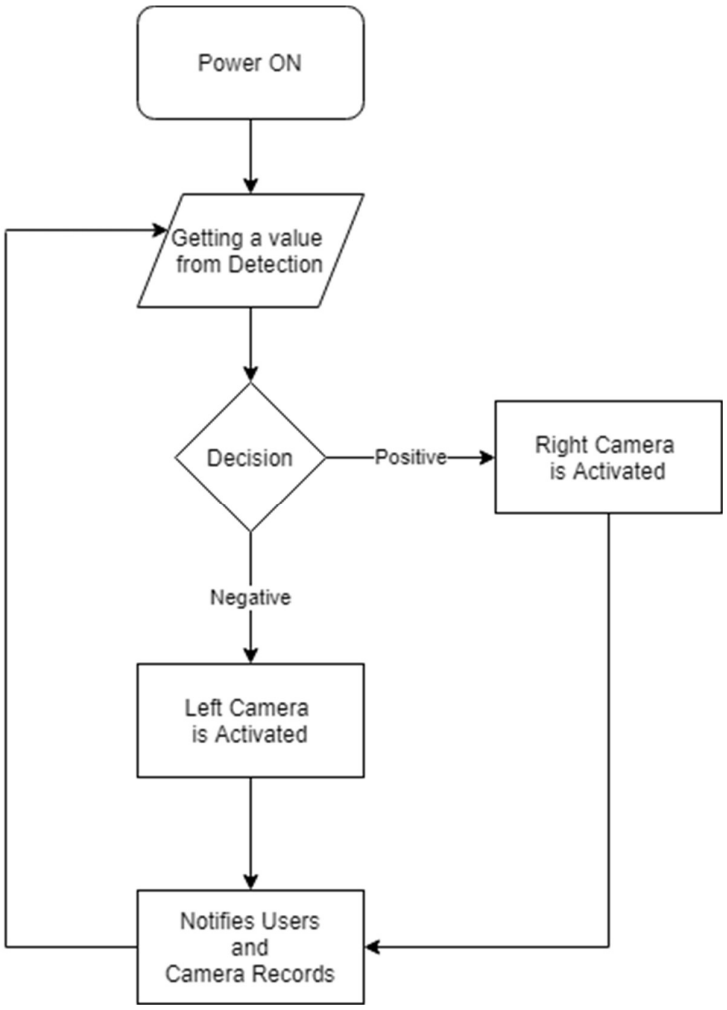
TARGET COUNTER WITH DIRECTION TRACKING

Content

Content 2
Flow Chart 3
Implementation 4
- Arduino to sensors 4
- Arduino to PLC 5
- PLC to Cameras 6

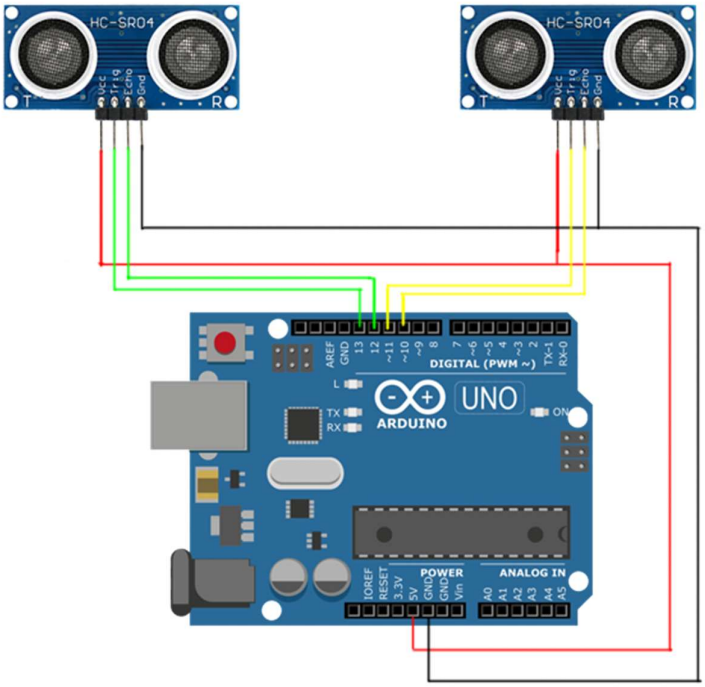
Things to do after installation 7
Arduino Code 8
PLC Ladder 12
Cool Term Software 14
Python for Laptop 16
- Alerting Device (Target-CT-seerver) 17
- Laptop (Target-CT-Client) 18

Flow Chart


Power ON
Getting a value from Detection
Decision
Positive
Right Camera is Activated
Negative
Left Camera is Activated
Notifies Users and Camera Records

Implementation

- Arduino to sensors



Wiring as shown in the picture

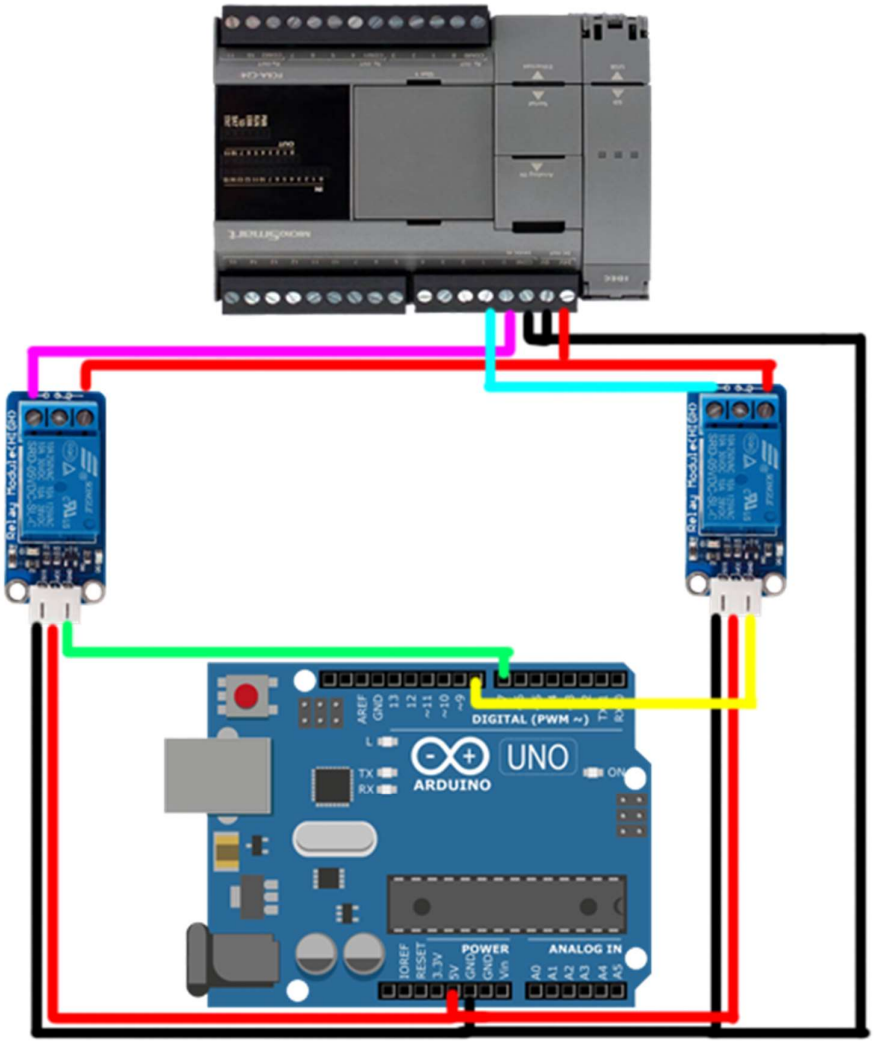
Left Ultrasonic

- Trig = pin13
- Echo = pin12

Right Ultrasonic

- Trig = pin11
- Echo = pin10

- Arduino to PLC



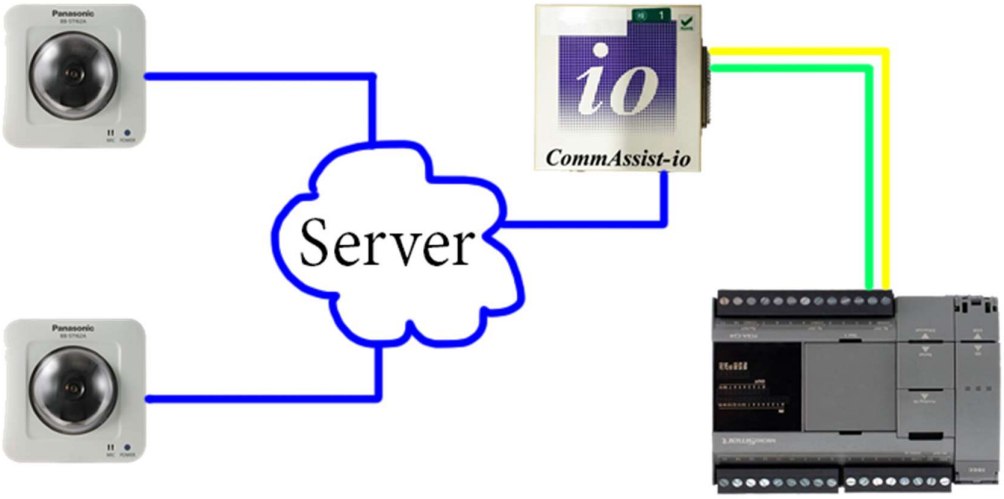
Left Relay Module

- DC+/DC- = 5V/GND @Arduino
- IN = pin7
- NO (normal open) = 24V @PLC
- COM = I01

Right Relay Module

- DC+/DC- = 5V/GND @Arduino
- IN = pin8
- NO (normal open) = 24V @PLC
- COM = I02

- PLC to Cameras

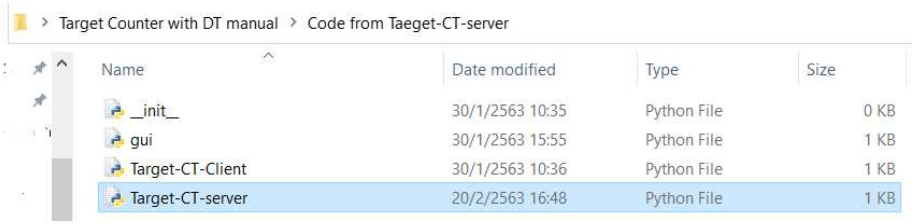


Things to do after installation

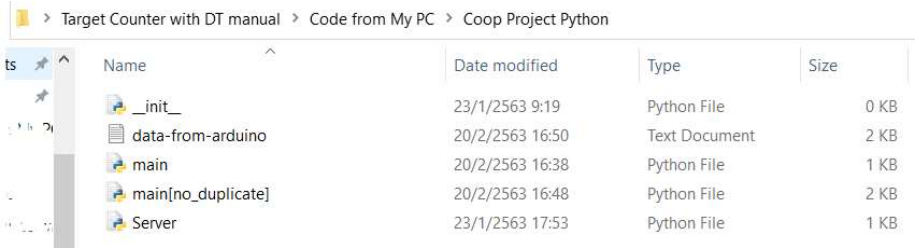
*** for more detail, please go to each topic ***

*** Must follow these steps, the method cannot be skipped ***

0. Download important files from this link https://www.dropbox.com/sh/22xyp9sj75p5m7m/AAB39TC0NVP1vzgsFr-SDeCva?dl=0
1. After finish calibration and installation, upload an Arduino code (do not use serial monitor in Arduino IDE software).
2. Run Cool Term software to capture message.
3. Run the python code, "Target-CT-server.py" in the alerting Device



4. Run the Python code, "main.py" in the Laptop that connects to the Arduino



Arduino Code

The code is in the file you downloaded, following this direction – Target Counter with DT manual → Code from My PC → Target_Counter_US_plc → Target_Counter_US_plc.ino

1. Header section

```
#include <HCSR04.h>

float left, right;
float calculated_value;
int Threshold = 50;
int count1 = 0, count_RL = 0, count_LR = 0;

/* If the sensors do NOT work correctly, shuffle numbers of left and right
  left(11,10) and right(13,12) → left(13,12) and right(11,10) */
UltraSonicDistanceSensor distanceSensor_left(11, 10);
UltraSonicDistanceSensor distanceSensor_right(13, 12);
int out_l2r = 7; //right camera
int out_r2l = 8; //left camera
```

- "Threshold" is the range of number to avoid an error from Ultrasonic Sensors.
- Choose one if the results alert incorrect results,
  Shuffle the pairs of "Ultrasonic sensor pins number" (code)
  Shuffle the physical Ultrasonic sensors – Left shuffles with Right (real)

2. Void setup

```
void setup() {
  pinMode(out_l2r, OUTPUT);
  pinMode(out_r2l, OUTPUT);
  Serial.begin(9600);
  Serial.println("Ready to work!");
}
```

- Nothing can be changed
- There are 2 OUTPUTS from the Arduino, pin 7 and 8 respectively.

3. Void loop

   For the safety, please do not change anything here

```
void loop() {
  left = distanceSensor_left.measureDistanceCm();
  right = distanceSensor_right.measureDistanceCm();

  //key formula
  calculation();

  /* if the number from sensors (not from the key formula) is negative the result is noop (no operation),
    if the result is not negative it will go to the conditions loop */
  if ((left < 0) || (right < 0)) {
    //noop
  }
```

- "left" and "right" are the sensors value in centimeter
- "calculation();" : this will call a function to calculate the key value

```
void calculation() {
  //key formula
  calculated_value = left - right;
}
```

DO NOT get confuse of READ value and CALCULATED value,
-READ value is from sensors, the value from sensors can NEVER be negative.
-CALCULATED value is from the formula (mathematics), so it can be negative

4. Conditions

```
if ((left < 0) || (right < 0)) {
  //noop
}
else { //conditions
  if (abs(calculated_value) < Threshold) {
    digitalWrite(out_l2r, LOW);
    digitalWrite(out_r2l, LOW);
  } else if (calculated_value < 0) {
    Serial.print("Animal is approaching! Be careful!\n");
    //      Serial.print("Left to Right\n");
    digitalWrite(out_l2r, HIGH);
    digitalWrite(out_r2l, LOW);
    count1 += 1;
    count_LR += 1;
    show data();
  } else {
    Serial.print("Animal is Leaving!\n");
    //      Serial.print("Right to Left\n");
    digitalWrite(out_r2l, HIGH);
    digitalWrite(out_l2r, LOW);
    count1 += 1;
    count_RL += 1;
    show data();
  }
```

There are 2 IF-ELSE loops,

First IF-ELSE: to avoid error from sensor if the READ value is negative

Second IF-ELSE: it is in ELSE of the first if-else loop, there are 3 states

1. when the absolute value of CALCULATED value is less than the Threshold.
2. when the value is less than Threshold.
3. when the value is greater than Threshold.

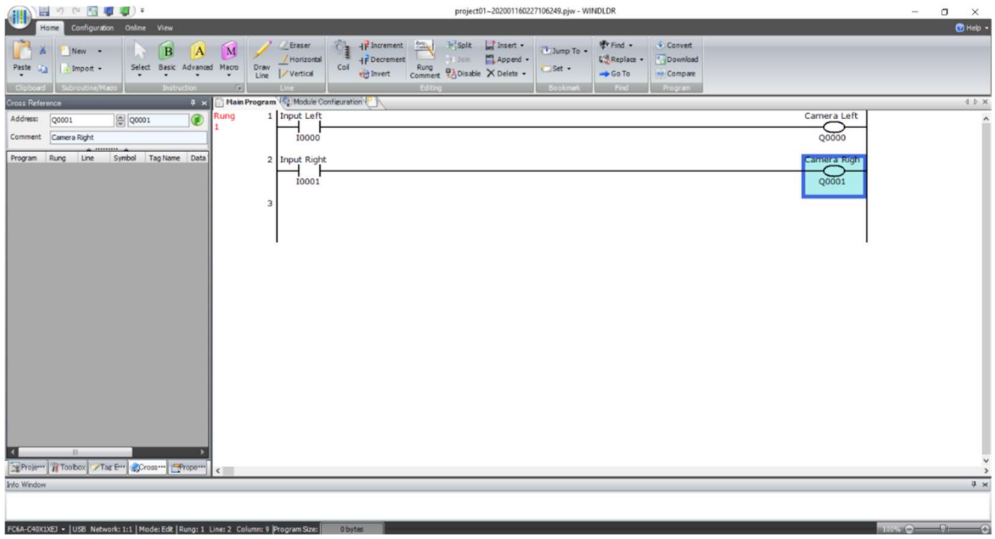
5. show_data(); function

***** This is so important because, it works with Cool Term software ******

Cool Term software will capture every message that shows in the Serial.print and Serial.println to the text file that will be used in the alerting system.
So if you want to see some value for calibration you can delete // sign in front of Serial.print and Serial.println to show the data you want to see, if you finish calibration, please keep // back

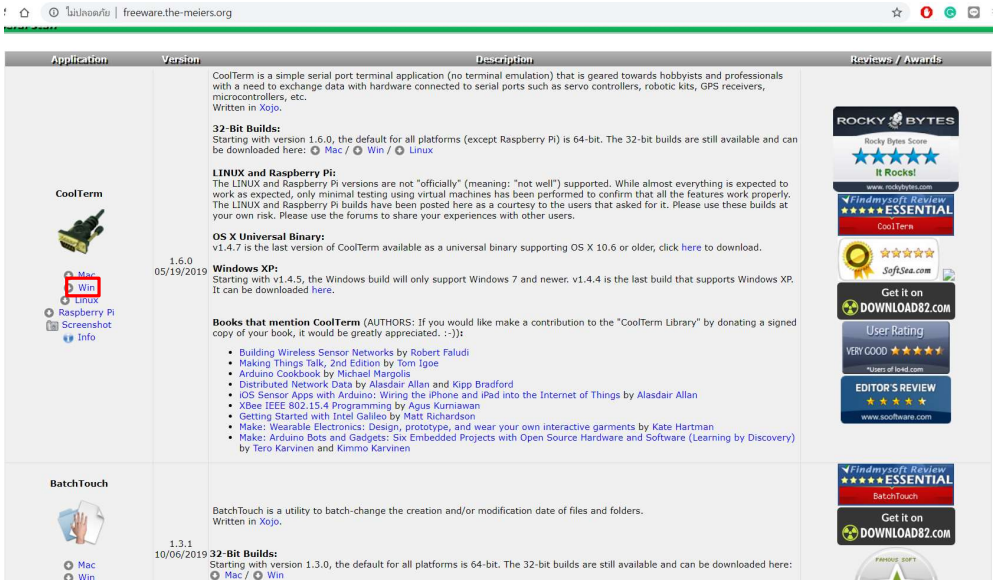
```
void show_data() {
  //print values of ultrasonic sensors
  //  Serial.print("Left: ");
  //  Serial.print(left);
  //  Serial.print("  Right: ");
  //  Serial.println(right);
  //print counter
  //  Serial.print("   Count_L2R : ");
  //  Serial.print(count_LR);
  //  Serial.print("   Count_R2L : ");
  //  Serial.print(count_RL);
  //  Serial.print("   Count(both directions) : ");
  //  Serial.println(count1);
  //print the result value
  //  Serial.print("Calculated_Value: ");
  //  Serial.println(calculated_value);
  delay(3000);
  //Serial.println("Ready for the next value\n");
}
```

PLC Ladder

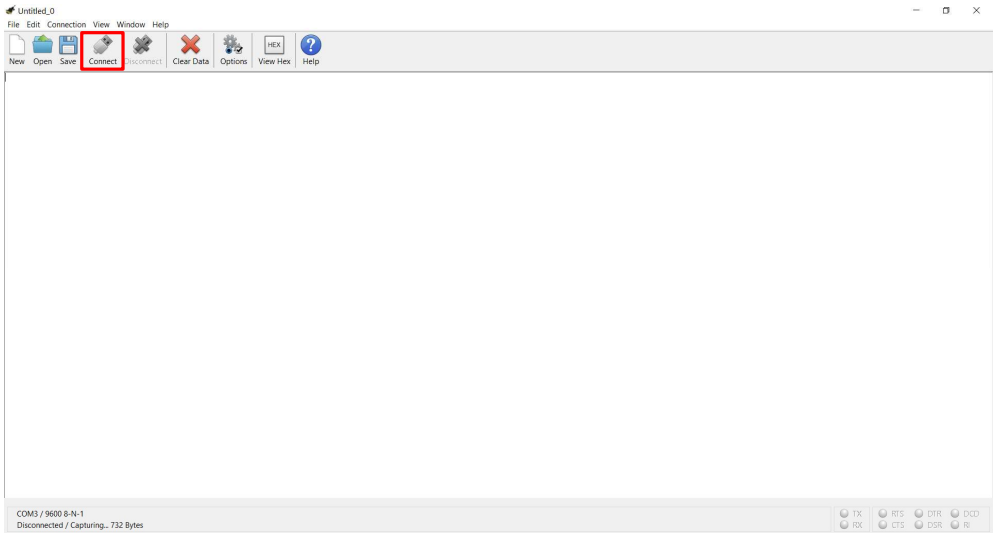

Main Program
Module Configuration
Rung 1
1 Input Left
I0000
Camera Left
Q0000
2 Input Right
I0001
Camera Right
Q0001
3

Cool Term Software

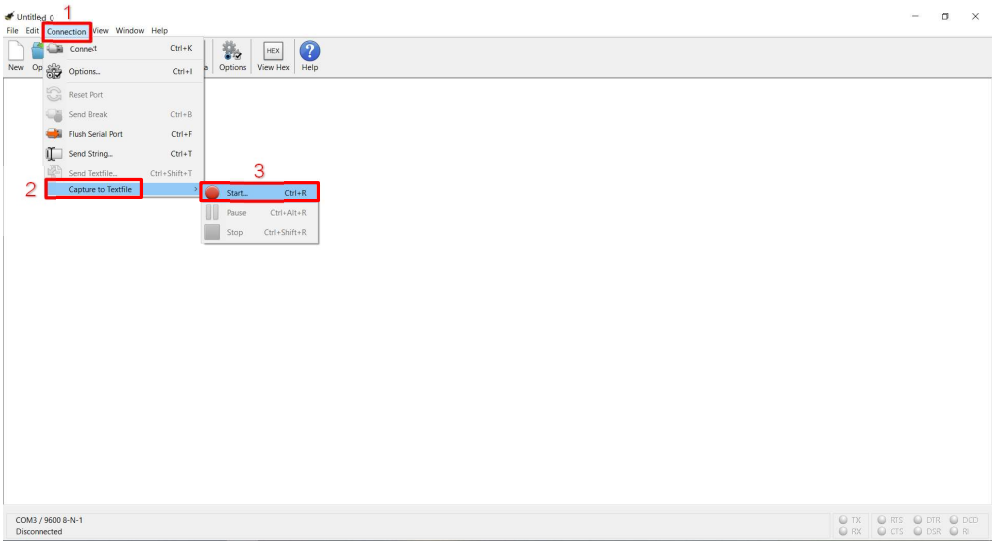
1. Download a free software from this link then install the software
   https://freeware.the-meiers.org/



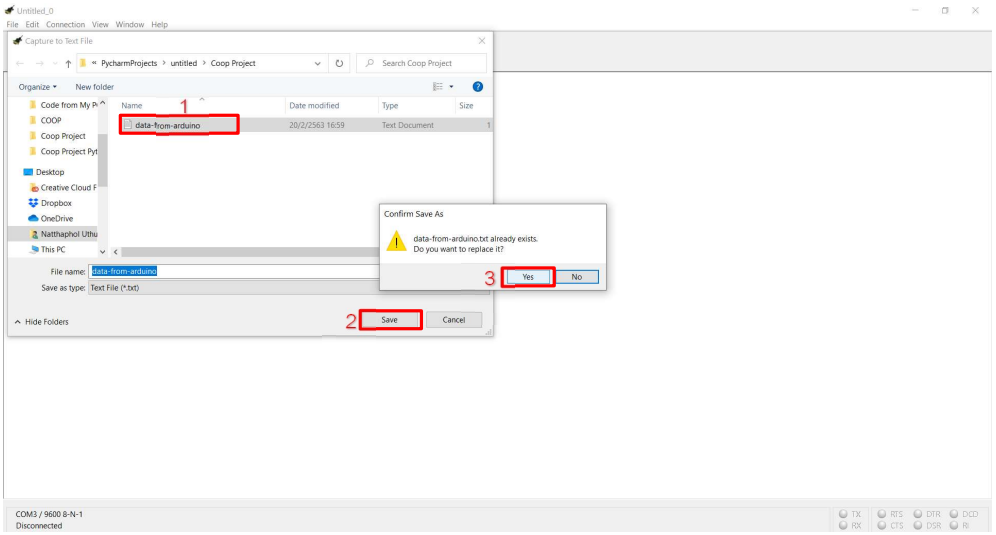
2. After Arduino is connected to PC, open the software then click "Connect"



3. Follow these steps: Connection → Capture to Textfile → Start



4. Select the text file name "data-from-arduino" → Save → Yes
*** The text file must be in the same folder with the Python code ***

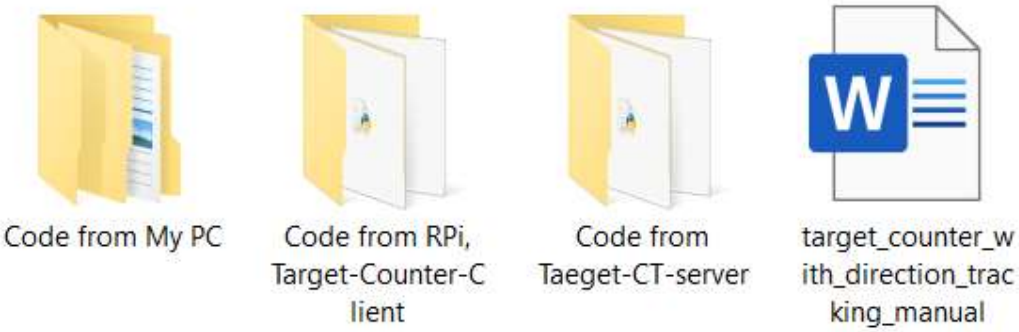


Cool Term software will capture every message that shows in the Serial.print and Serial.println to the text file that will be used in the alerting system.

Python for Laptop

*** The text file and the Python code must be in the same folder ***

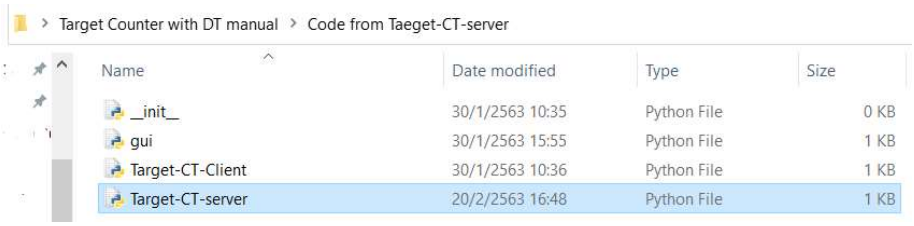
Name	Date modified	Type	Size
__init__	23/1/2563 9:19	Python File	0 KB
data-from-arduino	20/2/2563 16:50	Text Document	2 KB
main	20/2/2563 16:38	Python File	1 KB
main[no_duplicate]	20/2/2563 16:48	Python File	2 KB
Server	23/1/2563 17:53	Python File	1 KB



In the python codes you can modify whatever you want it to be, I was a new learner (beginner) of Python programming.

- # Alerting Device (Target-CT-seerver)

  For the Alerting device, in folder "Code from Target-CT-server" there are these files but only "Target-CT-server.py" is used



The red highlight IP address should be changed for each device because the IP address of each device is different, the IP address MUST be IP address of the alerting device NOT the Laptop that connects to Arduino.

```
import socket

def Server_func():
    hnp = ("192.168.10.130",5050)
    serv = socket.socket()
    serv.bind(hnp)

    serv.listen(2)
    print("Waiting for the connection...")

    target_ct,addr = serv.accept()
    print("Connected from : " + str(addr))

    while True:
        data_target_ct = target_ct.recv(1024).decode('utf-8')
        if data_target_ct != "No motion":
            print(data_target_ct)
        reply_message = "OK"
        target_ct.send(reply_message.encode('utf-8'))
    target_ct.close()

if __name__ == "__main__":
    Server_func()
```

- # Laptop (Target-CT-Client)

For the Laptop that connects to the Arduino, in folder "Code from my PC" there are these files but only "main.py" is used

PLC_IDEC_Ladder	20/2/2563 17:32	File folder	
Target_Counter_US_plc	20/2/2563 17:32	File folder	
__init__	23/1/2563 9:19	Python File	0 KB
data-from-arduino	20/2/2563 16:50	Text Document	2 KB
main	20/2/2563 16:38	Python File	1 KB
main[no_duplicate]	20/2/2563 16:48	Python File	2 KB
Server	23/1/2563 17:53	Python File	1 KB

The red highlight IP address should be changed for each device because the IP address of each device is different, the IP address MUST be IP address of the alerting device NOT the Laptop that connects to Arduino.

```python
import socket
from time import sleep

def main_function():
    host = "192.168.10.130"
    port = 5050
    server = socket.socket()
    server.connect((host,port))

    while True:
        with open("data-from-arduino.txt", "r") as f:
            read_line = f.read().splitlines()
            last_line = read_line[-1]
            print(last_line)
        f.close()

        server.send(last_line.encode('utf-8'))
        message_server = server.recv(1024).decode('utf-8')
        print(message_server)
        sleep(3)
    server.close()


if __name__ == "__main__":
    main_function()
```