Target counter with direction Tracking

# Content

[Content 2](#_Toc33178596)

[Flow Chart 3](#_Toc33178597)

[Implementation 4](#_Toc33178598)

[ Arduino to sensors 4](#_Toc33178599)

[ Arduino to PLC 5](#_Toc33178600)

[ PLC to Cameras 6](#_Toc33178601)

[Things to do after installation 7](#_Toc33178602)

[Arduino Code 8](#_Toc33178603)

[PLC Ladder 12](#_Toc33178604)

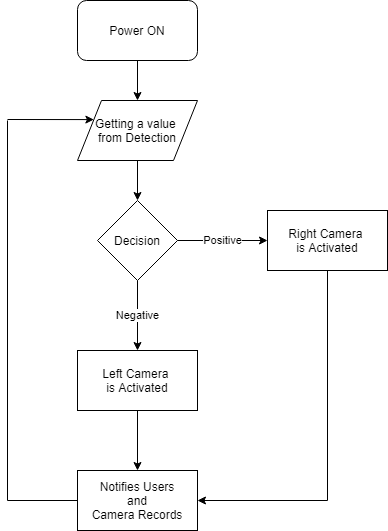
[Cool Term Software 14](#_Toc33178605)

[Python for Laptop 16](#_Toc33178606)

[ Alerting Device (Target-CT-seerver) 17](#_Toc33178607)

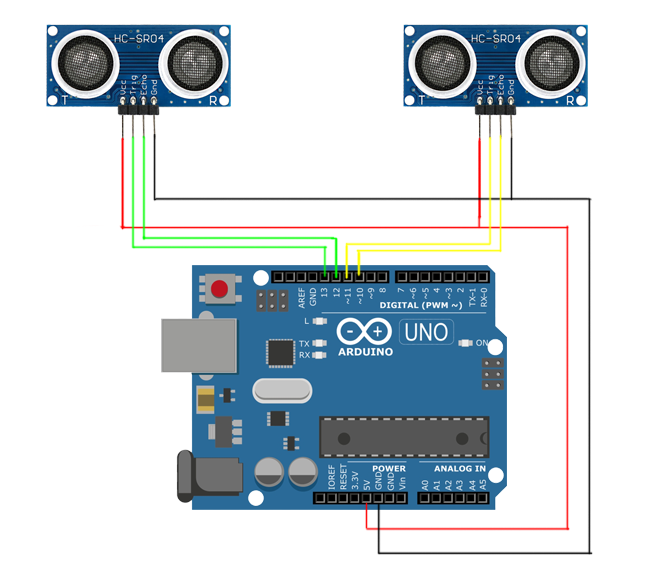
[ Laptop (Target-CT-Client) 18](#_Toc33178608)

# Flow Chart



# Implementation

## Arduino to sensors



Wiring as shown in the picture

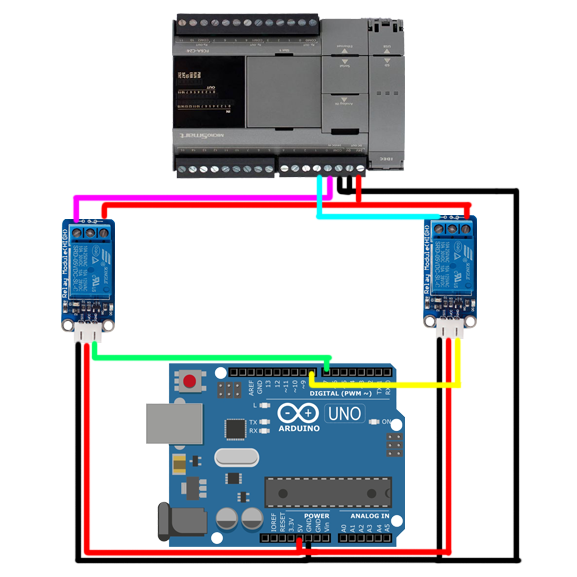
Left Ultrasonic

* + 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

รูปภาพประกอบด้วย อุปกรณ์อิเล็กทรอนิกส์

คำอธิบายที่สร้างขึ้นโดยอัตโนมัติ

# Things to do after installation

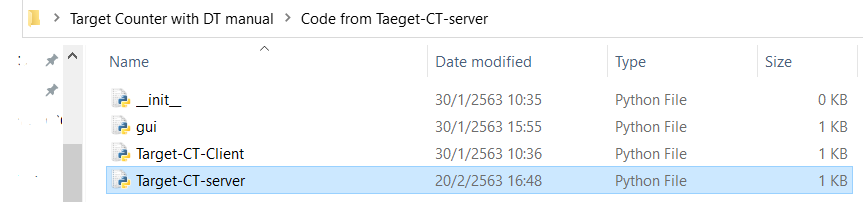
\*\*\* for more detail, please go to each topic \*\*\*

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

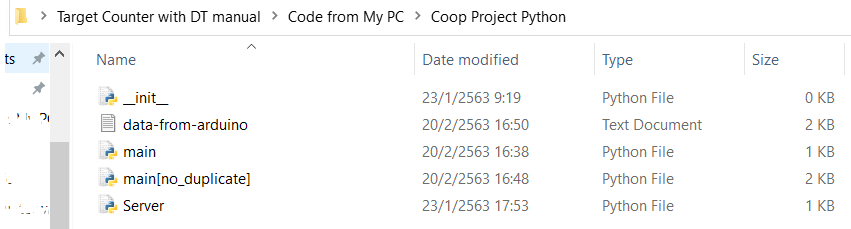
1. Download important files from this link

<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



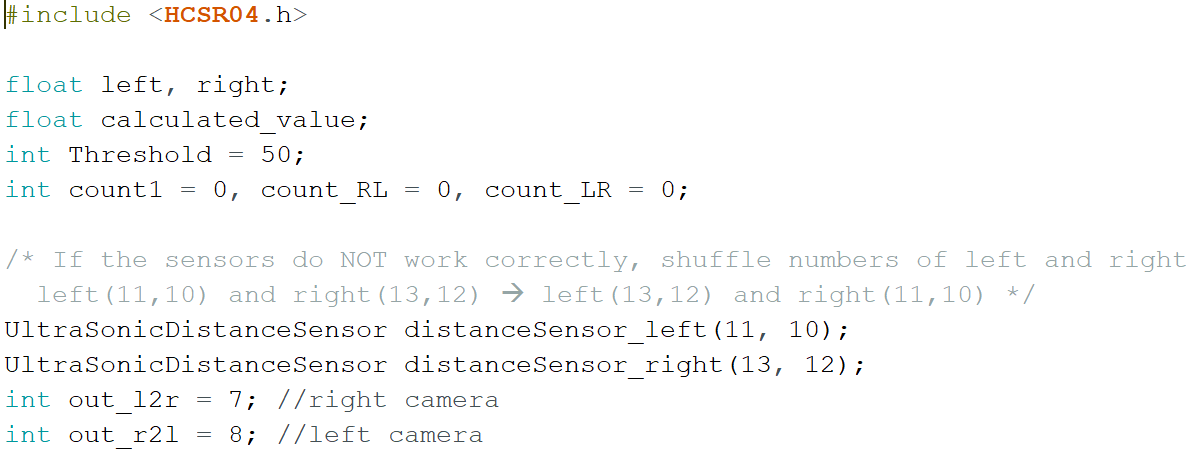
1. Run the Python code, “main.py” in the Laptop that connects to the Arduino



# 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

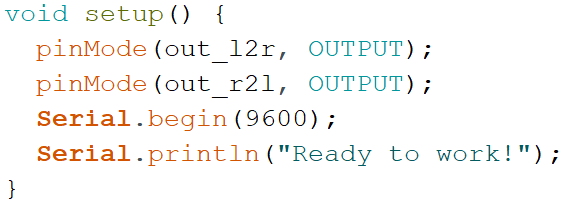


* “Threshold” is the range of number to avoid an error from Ultrasonic Sensors.
* Choose one if the results alert incorrect results,

Shuffle the pairs of “Ultrasonic sensor pins number” (code)

Shuffle the physical Ultrasonic sensors – Left shuffles with Right (real)

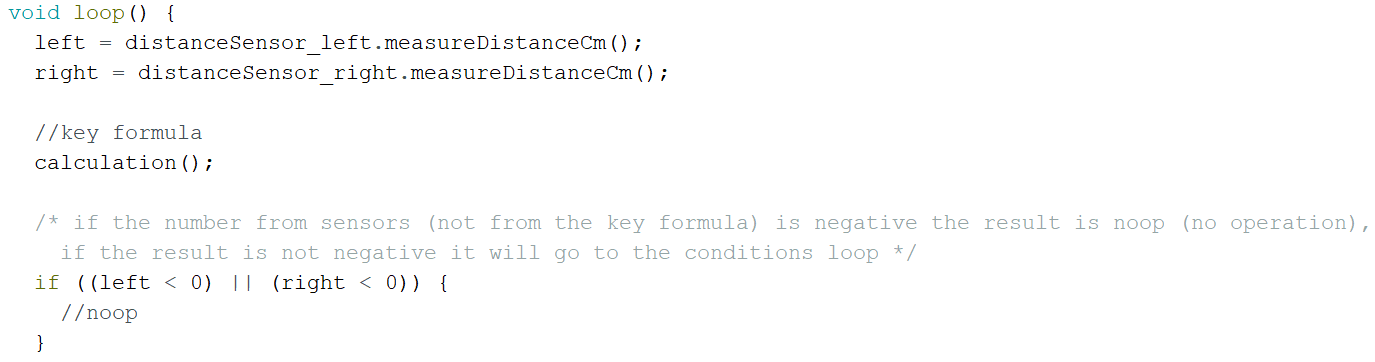
1. Void setup



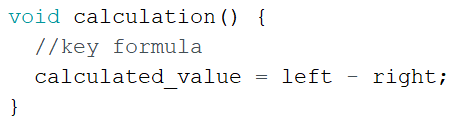
* Nothing can be changed
* There are 2 OUTPUTS from the Arduino, pin 7 and 8 respectively.

1. Void loop

For the safety, please do not change anything here



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



**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**

1. Conditions



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.

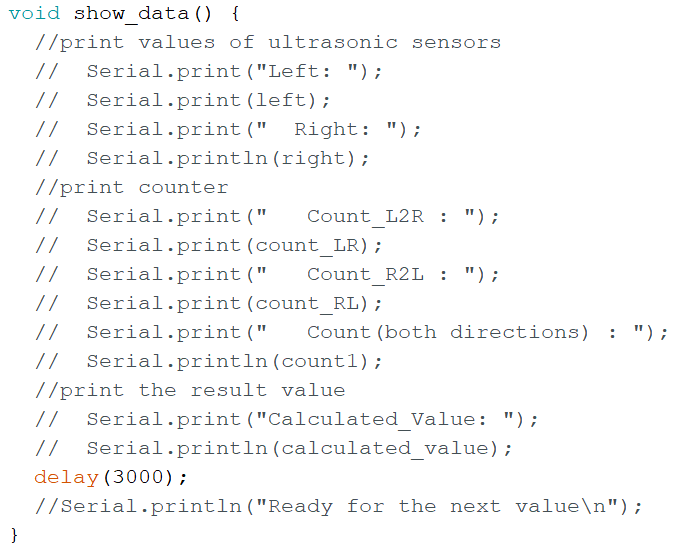
3. when the value is greater than Threshold.

1. show\_data(); function

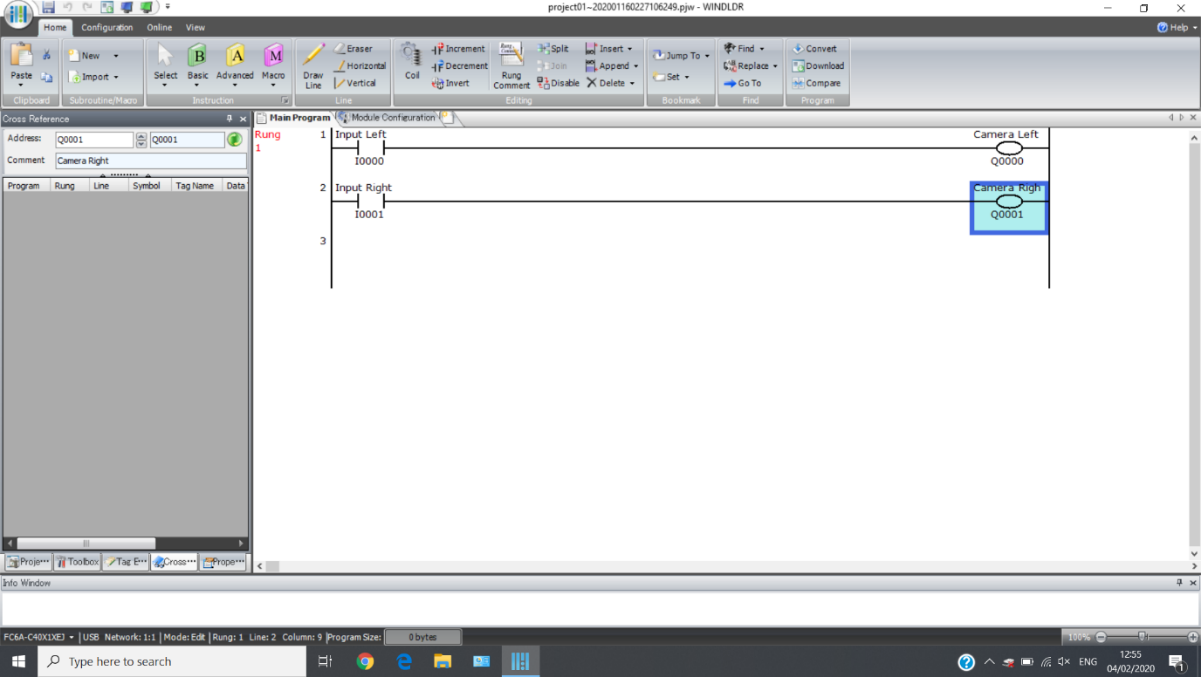
**\*\*\* This is so important because, it works with Cool Term software \*\*\*\***

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



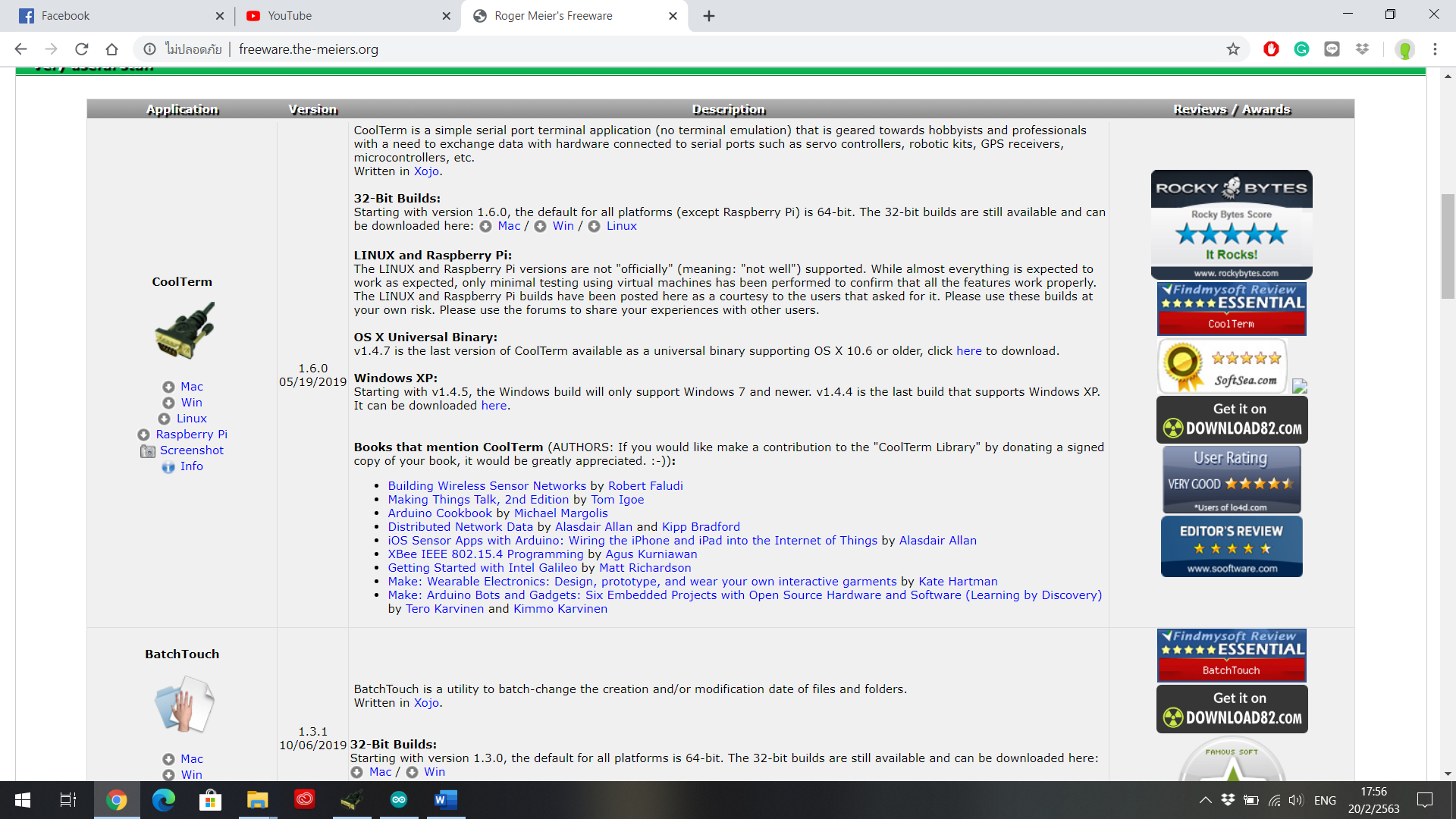
# PLC Ladder



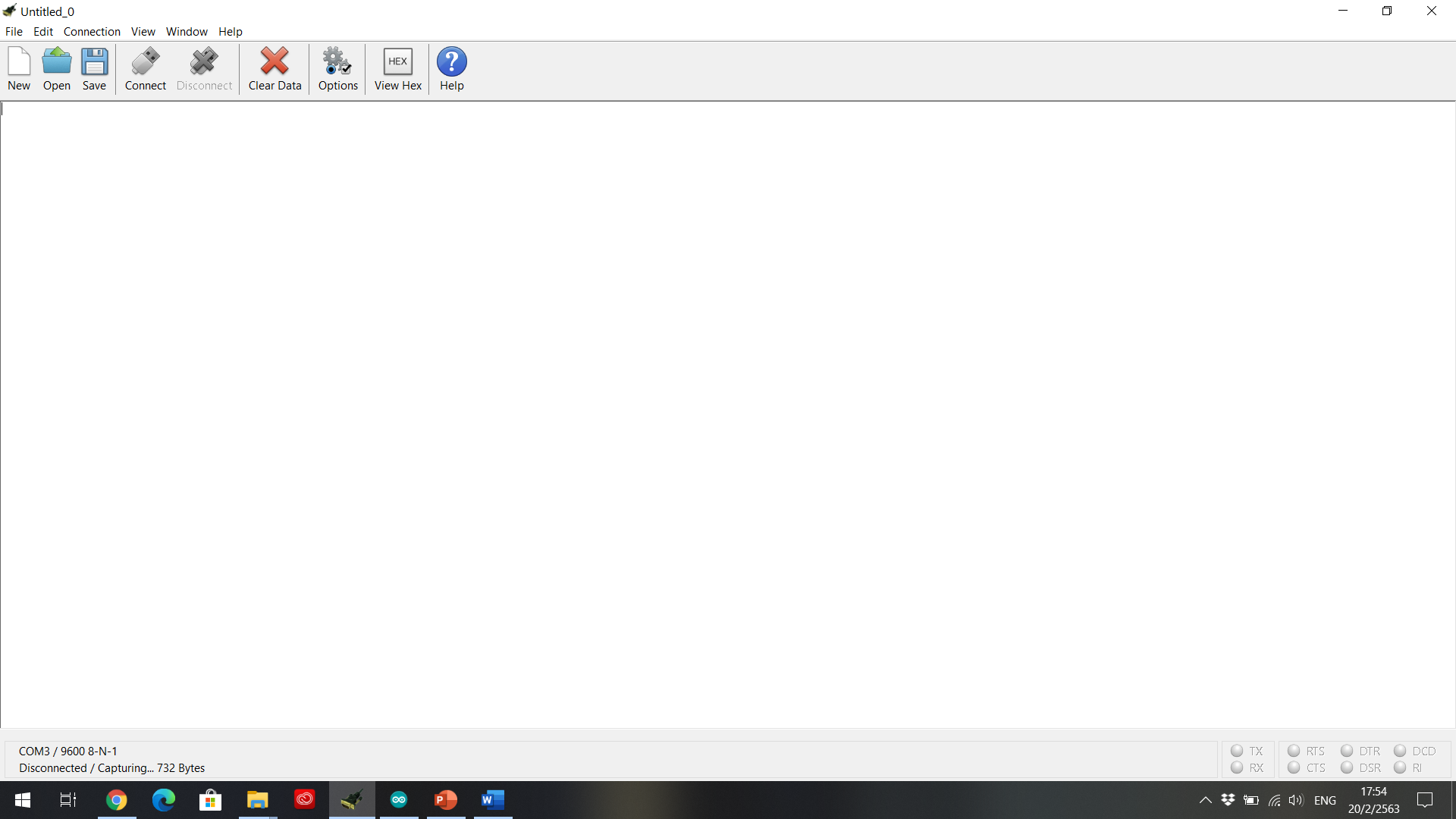
# Cool Term Software

1. Download a free software from this link then install the software

<https://freeware.the-meiers.org/>

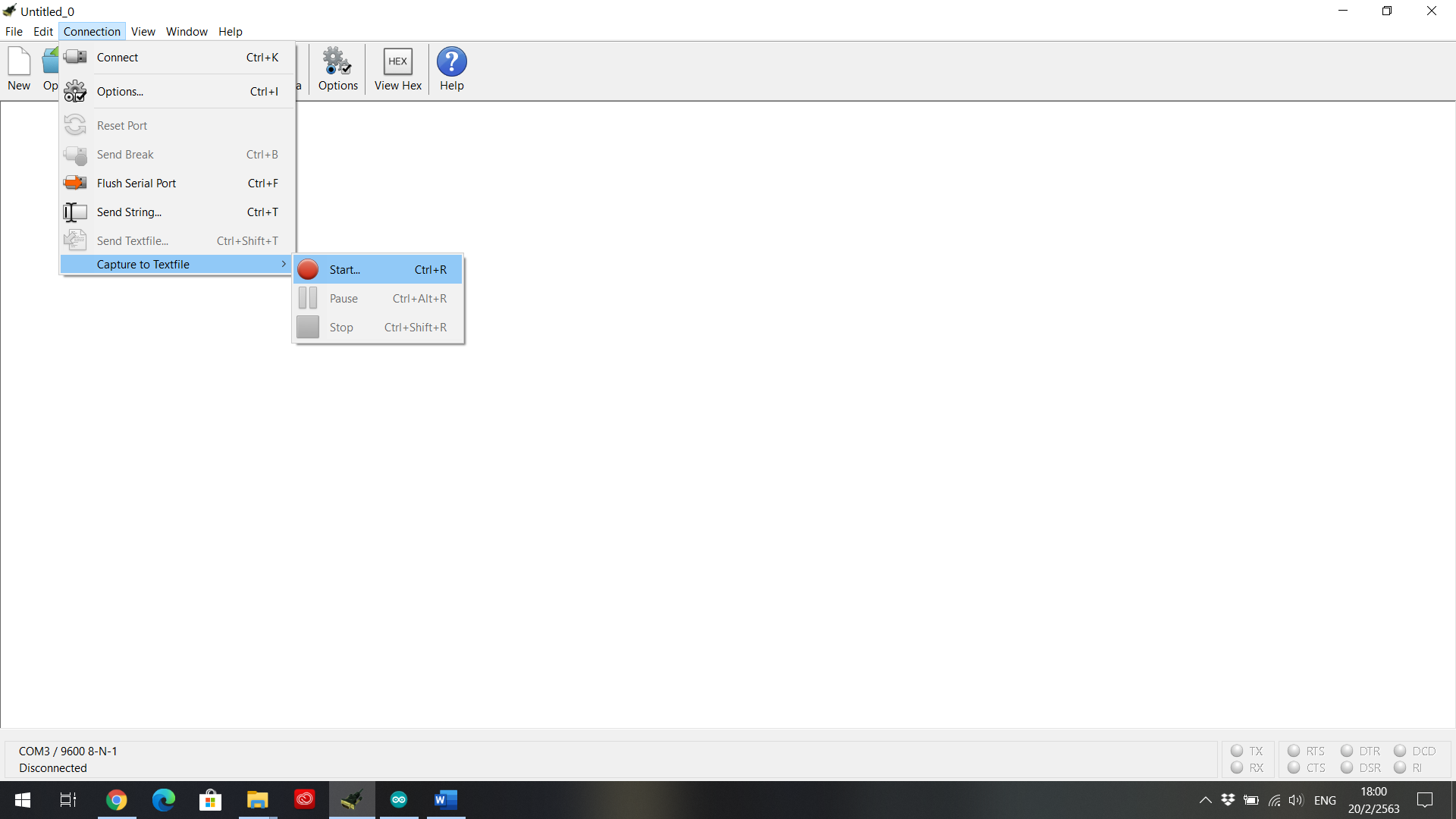


1. After Arduino is connected to PC, open the software then click “Connect”



1. Follow these steps: Connection 🡪 Capture to Textfile 🡪 Start

1

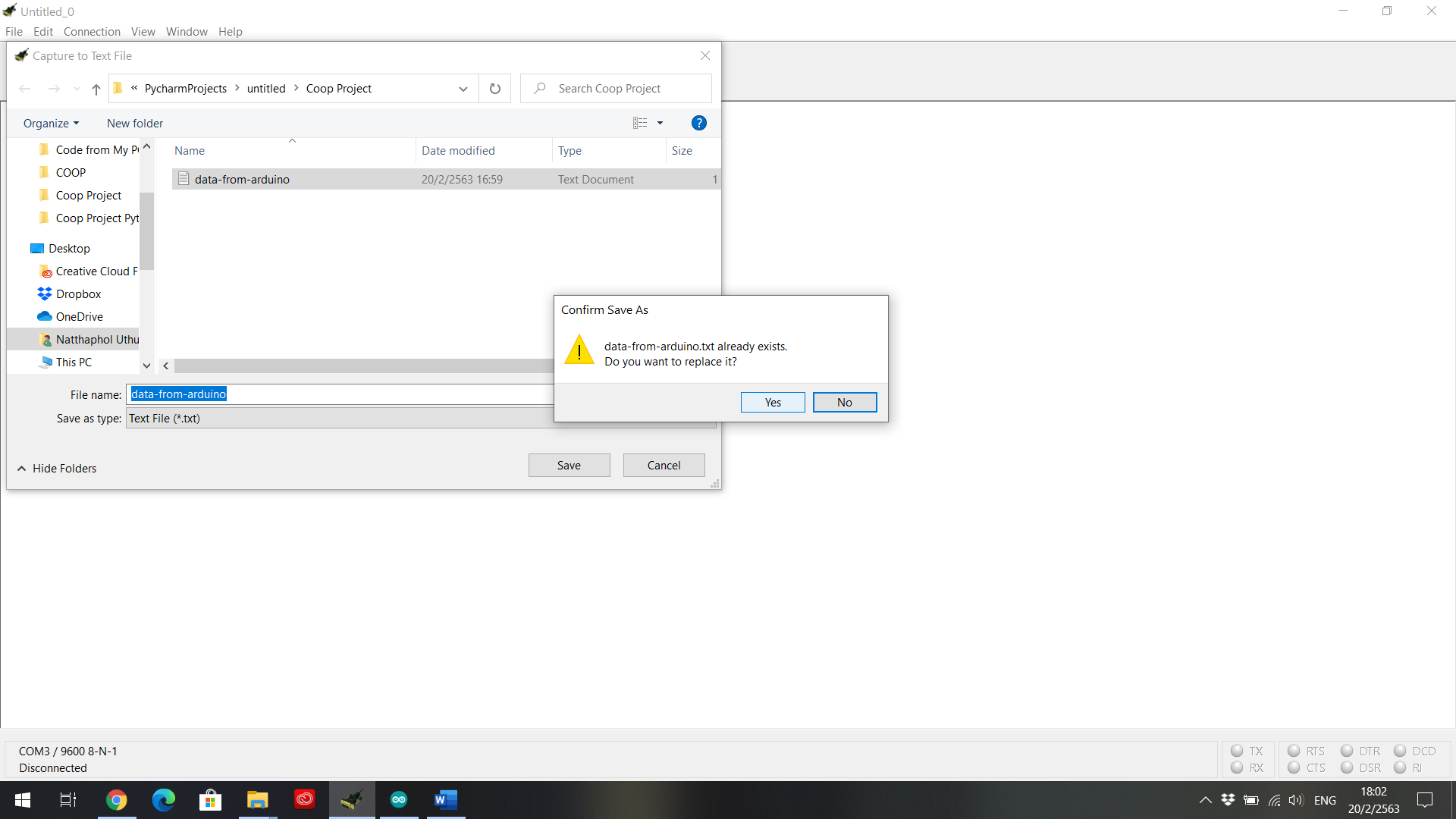


3

2

1. Select the text file name “data-from-arduino” 🡪 Save 🡪 Yes

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



3

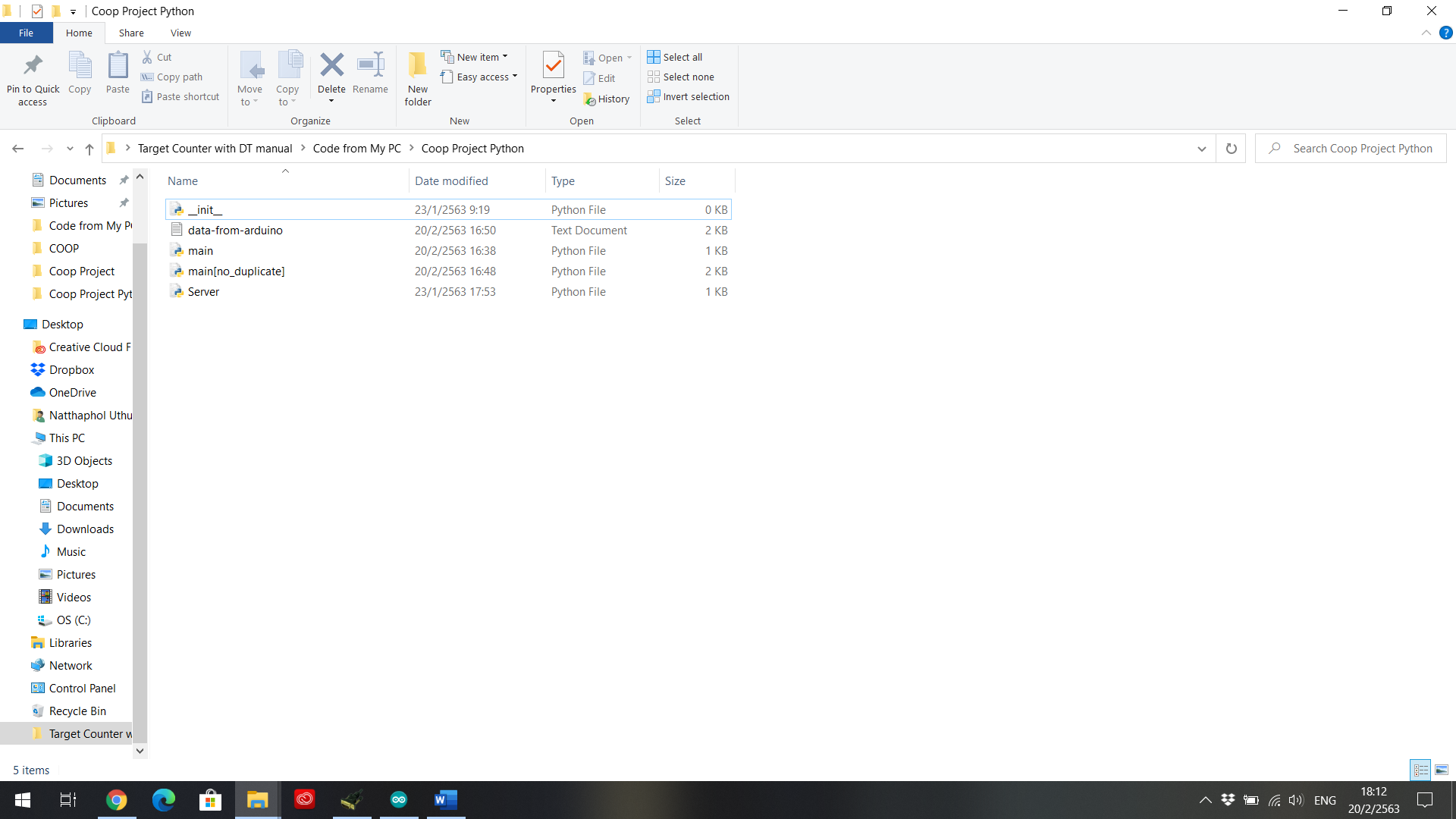
2

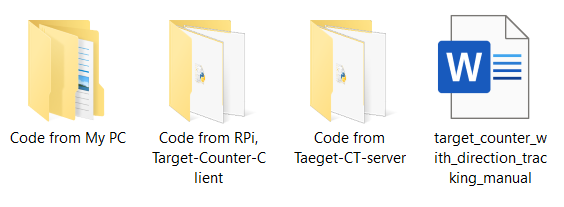
1

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 \*\*\*



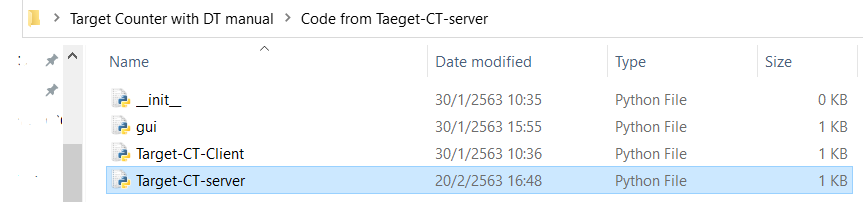


In the python codes you can modify whatever you want it to be, I was a new learner (beginner) of Python programming.

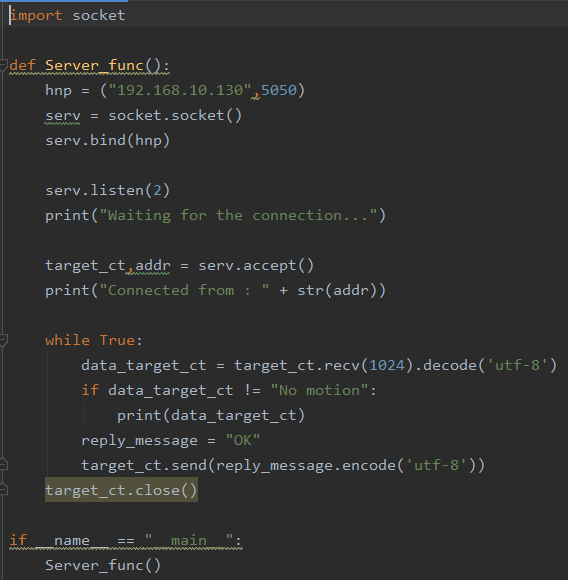
## Alerting Device (Target-CT-seerver)

For the Alerting device, in folder “Code from Target-CT-server”

there are these files but only “Target-CT-server.py” is used



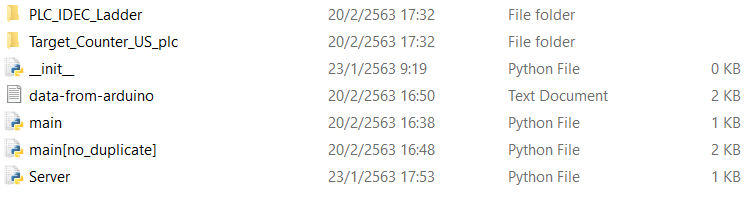
The red highlight IP address should be changed for each device because the IP address of each device is different, the IP address MUST be IP address of the alerting device NOT the Laptop that connects to Arduino.



## 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



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.

