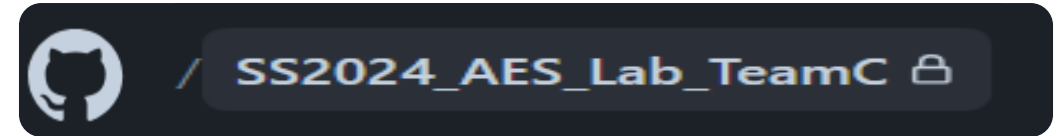


Industrial IoT. Image Source: [IoT Innovations | Top 33 Industrial IoT Device Examples \(bytebeam.io\)](#).



**Teammates (alphab.):**

- Akram Md
- Richard Jimenez

**Module:** Advanced Embedded Systems – SuSe24

**Examiners:** Prof. Dr. Hayek

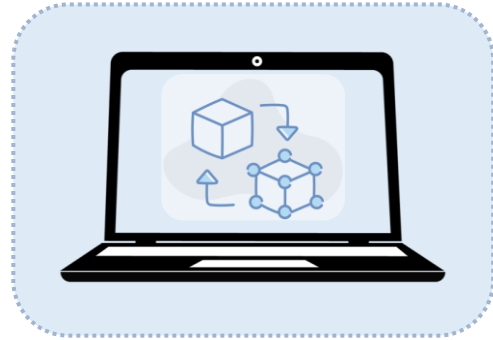
**ELE** – Hochschule Hamm-Lippstadt

July 2<sup>nd</sup> 2024

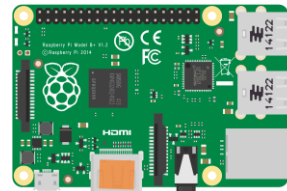
# IoT Graphical Block Diagram

Source: Own  
Creation.

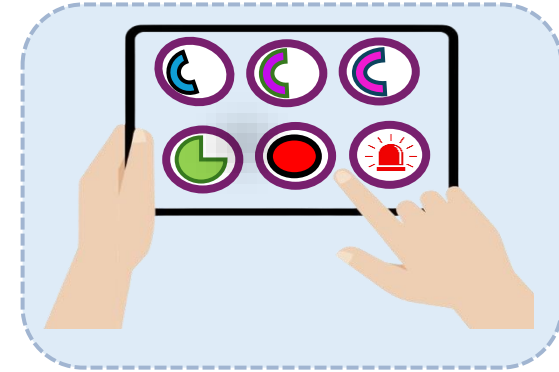
## IIOT APPLICATION



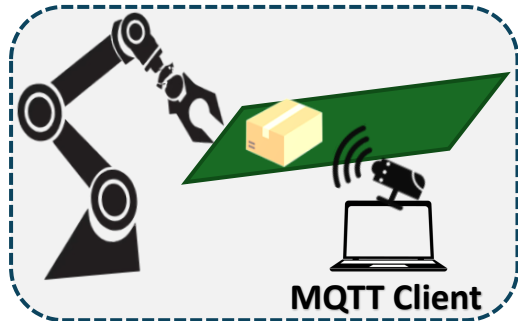
MQTT Client



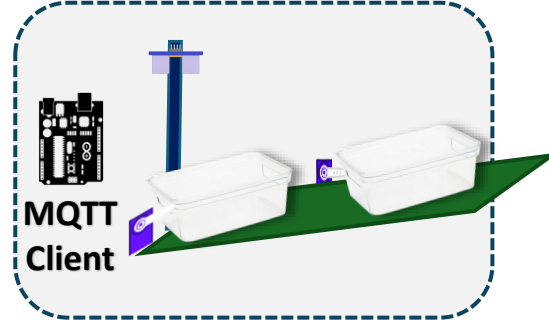
MQTT Broker



MQTT Client



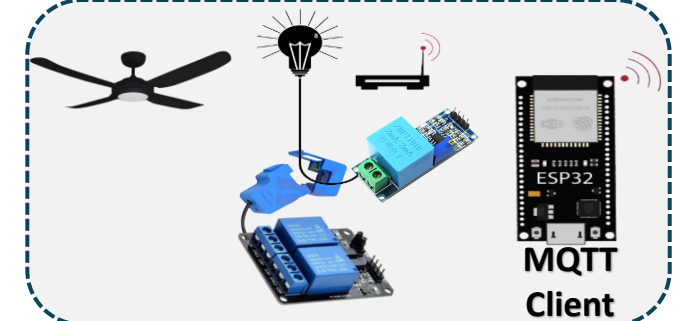
Quality Control



Stock Management



Safety and Security system



Power Monitoring and controlling system

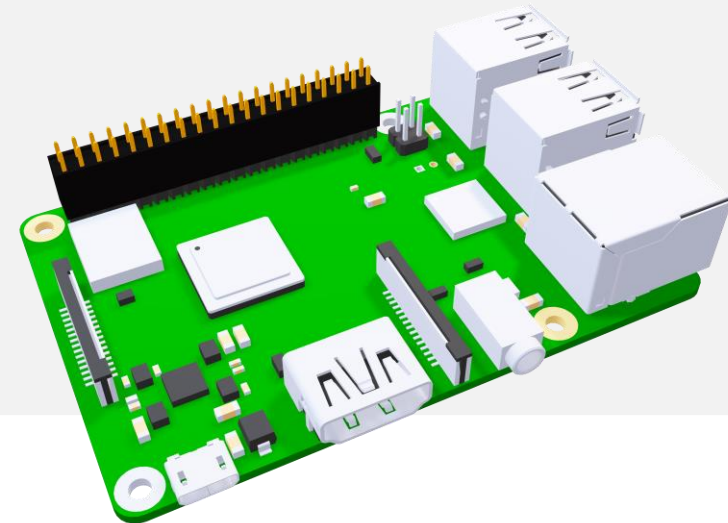
# IIoT – MQTT Broker

```
rejimenezg@raspberrypi: ~  
File Edit Tabs Help  
rejimenezg@raspberrypi:~ $ mosquitto_sub -d -t "bin/updates"  
Client (null) sending CONNECT  
Client (null) received CONNACK (0)  
Client (null) sending SUBSCRIBE (Mid: 1, Topic: bin/updates, QoS: 0, Options: 0x0000)  
Client (null) received SUBACK  
Subscribed (mid: 1): 0  
Client (null) sending PINGREQ  
Client (null) received PINGRESP  
Client (null) received PUBLISH (d0, q0, r0, m0, 'bin/updates', ... (1 bytes))  
1  
Client (null) sending PINGREQ  
Client (null) received PINGRESP  
Client (null) sending PINGREQ  
Client (null) received PINGRESP  
Client (null) sending PINGREQ  
Client (null) received PINGRESP  
Client (null) sending PINGREQ  
Client (null) received PINGRESP  
Client (null) sending PINGREQ  
Client (null) received PINGRESP  
Client (null) sending PINGREQ  
Client (null) received PINGRESP  
Client (null) received PUBLISH (d0, q0, r0, m0, 'bin/updates', ... (1 bytes))  
1  
Client (null) received PUBLISH (d0, q0, r0, m0, 'bin/updates', ... (1 bytes))  
1  
Client (null) received PUBLISH (d0, q0, r0, m0, 'bin/updates', ... (1 bytes))  
1  
Client (null) received PUBLISH (d0, q0, r0, m0, 'bin/updates', ... (1 bytes))  
1  
Client (null) sending PINGREQ  
Client (null) received PINGRESP  
Client (null) sending PINGREQ
```

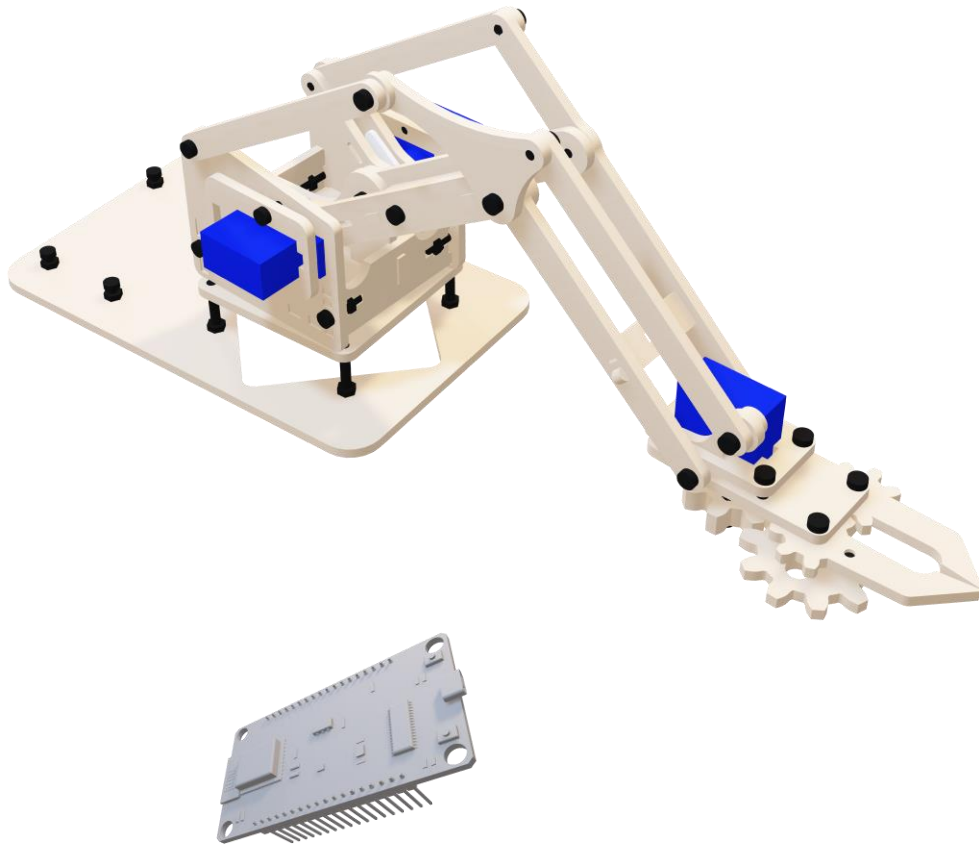
## Mosquitto MQTT Broker:

- Raspberry Pi 3

Tools and SW: Mosquitto MQTT, VNCC, PuTTY

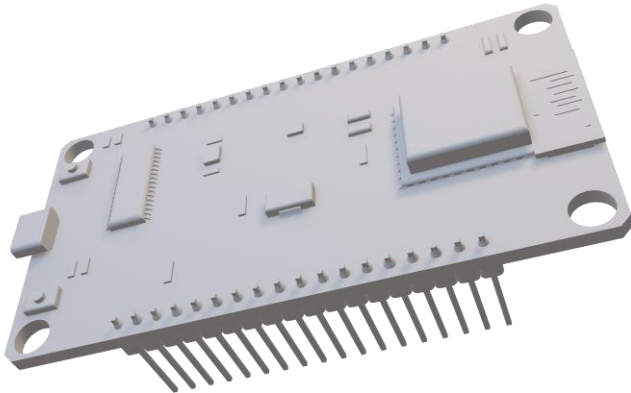
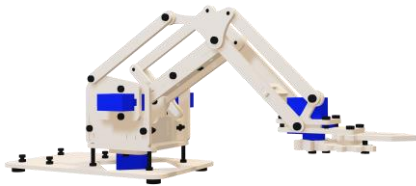


# IIoT – Quality Control



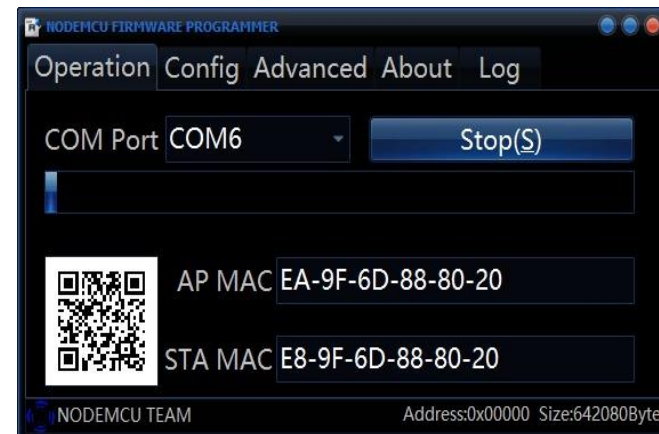
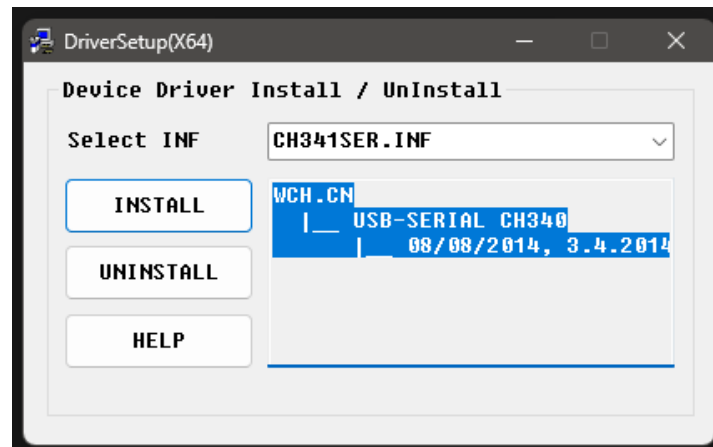
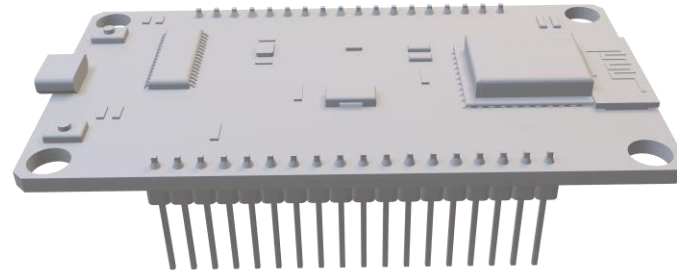
- Embedded control system with IIoT integration: Real-time servo control, data analytics, and remote operation capabilities

# IIoT – Quality Control

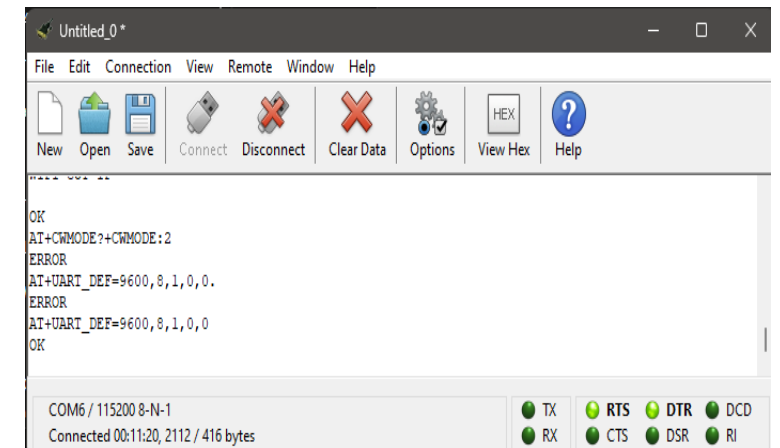


- Embedded control system with IIoT integration: Real-time servo control, data analytics, and remote operation capabilities
- ESP8266 microcontroller with MicroPython: Wi-Fi-enabled, low-cost solution for IoT connectivity and rapid prototyping

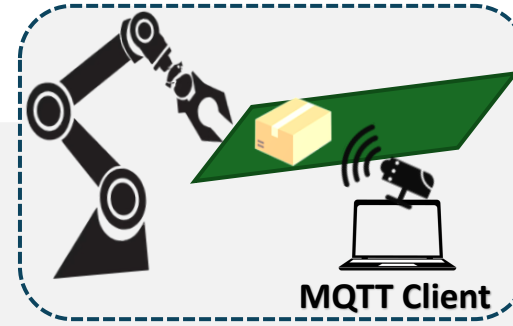
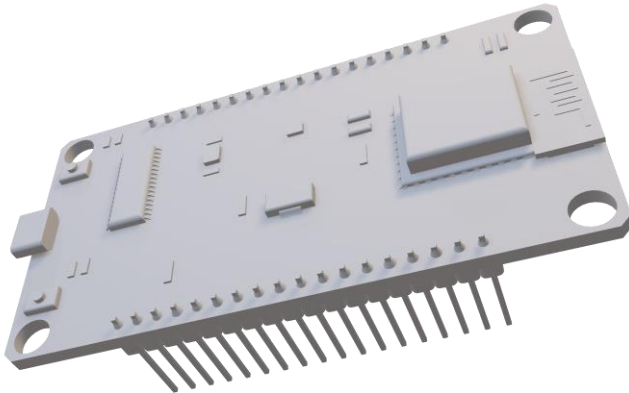
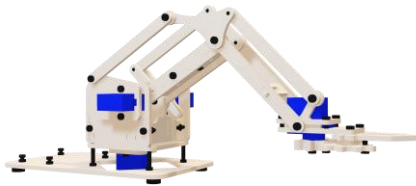
# ESP8266 Extra Setup



Source: Own  
Creation.



# IIoT – Quality Control



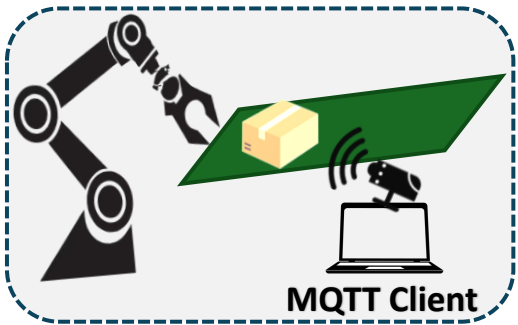
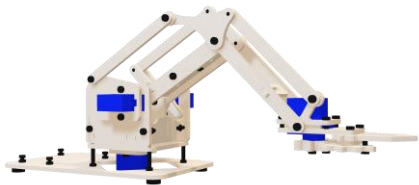
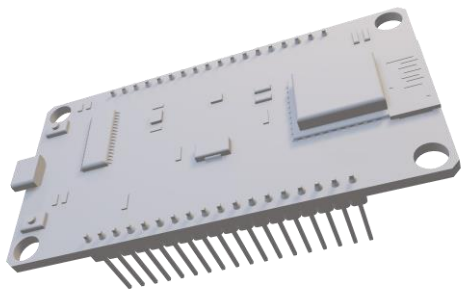
Quality Control

## Development Tools for ESP8266 and MicroPython:

- VSCode and Pymakr extension
- PyCharm and MicroPython extension
- uPyCraft IDE



# IIoT – Quality Control



Quality Control

Three sources files:

```
▼ RoboticarmSources
  boot.py          5, U
  main.py          U
  umqttsimple.py   U
```

Source: Own  
Creation.



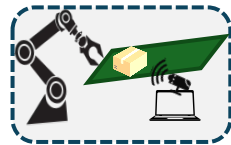
# IIoT – Quality Control Main Sources Code

```

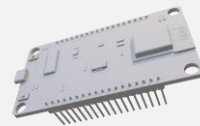
boot.py 5, U
Final_Project_sources > RoboticarmSources > boot.py > ...
1
2 import time
3 from umqttsimple import MQTTClient
4 import ubinascii
5 import machine
6 import micropython
7 import network
8 import esp
9 esp.osdebug(None)
10 import gc
11 gc.collect()
12
13 ssid = 'aeshotspot' #'UPCBF3B25D'
14 password = 'aeshotspot' #'Jmhpsapyv23j'
15 mqtt_server = '192.168.0.59' #
16 mqtt_user = ''
17 mqtt_pass = ''
18
19 #EXAMPLE IP ADDRESS
20 #mqtt_server = '192.168.1.144'
21 client_id = ubinascii.hexlify(machine.unique_id())
22 topic_sub = b'bin/updates'
23 topic_pub = b'bin/updates'
24
25 last_message = 0
26 message_interval = 5
27 counter = 0
28
29 station = network.WLAN(network.STA_IF)
30
31 station.active(True)
32 station.connect(ssid, password)
33
34 while station.isconnected() == False:
35     pass
36
37 print('Connection successful')
38 print(station.ifconfig())

```

Source: Own  
Creation.



MQTT Client  
Quality Control



```

main.py 9
Final_Project_sources > RoboticarmSources > main.py > ...
1 from machine import Pin, PWM
2 import time
3
4 # Servo setup
5 servo_pin1 = Pin(2) # D4
6 servo_pin2 = Pin(4) # D2
7 servo_pin3 = Pin(5) # D1
8
9 servo1 = PWM(servo_pin1, freq=50)
10 servo2 = PWM(servo_pin2, freq=50)
11 servo3 = PWM(servo_pin3, freq=50)
12
13 # Global flag to control servo movement
14 continue_movement = False
15
16 def set_angle(servo, angle):
17     duty = int(((angle / 180) * 115) + 40)
18     servo.duty(duty)
19
20 def move_servos():
21     global continue_movement
22     continue_movement = True
23     for i in range(1):
24         if not continue_movement:
25             break
26
27         # Servo 1
28         set_angle(servo1, 0)
29         print("Servo 1 moved to 0 degrees")
30         time.sleep(1)
31         if not continue_movement:
32             break
33         set_angle(servo1, 90)
34         print("Servo 1 moved to 90 degrees")
35         time.sleep(1)
36         if not continue_movement:
37             break

```

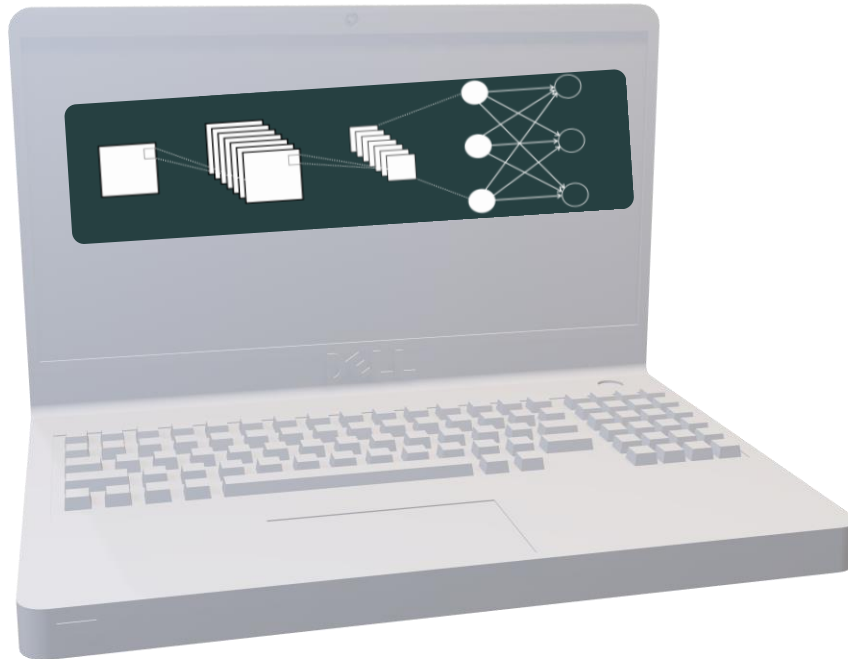
```

main.py 9
Final_Project_sources > RoboticarmSources > main.py > ...
84 def sub_cb(topic, msg):
85     print('Starting servo movement')
86     move_servos()
87     elif msg == b'1':
88         print('Stopping servos')
89         stop_servos()
90
91
92 def connect_and_subscribe():
93     global client_id, mqtt_server, topic_sub
94     client = MQTTClient(client_id, mqtt_server, user=mqtt_user, password=mqtt_pass)
95     client.set_callback(sub_cb)
96     client.connect()
97     client.subscribe(topic_sub)
98     print('Connected to %s MQTT broker, subscribed to %s topic' % (mqtt_server, topic_sub))
99     return client
100
101
102 def restart_and_reconnect():
103     print('Failed to connect to MQTT broker. Reconnecting...')
104     time.sleep(10)
105     machine.reset()
106
107
108 try:
109     client = connect_and_subscribe()
110 except OSError as e:
111     restart_and_reconnect()
112
113 while True:
114     try:
115         client.check_msg()
116         if (time.time() - last_message) > message_interval:
117             msg = b'Hello #%d' % counter
118             client.publish(topic_pub, msg)
119             last_message = time.time()
120             counter += 1
121     except OSError as e:
122         restart_and_reconnect()

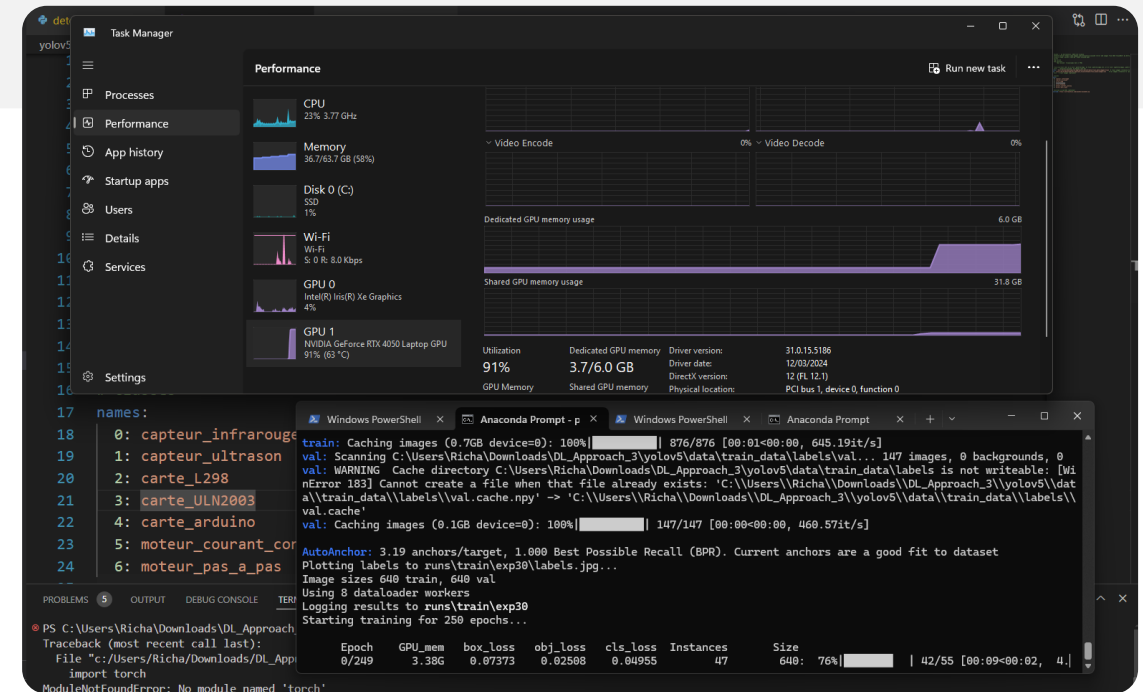
```

Source: Own  
Creation.

# IloT – Quality Control - CNN



- CNN Algorithm for Object Detection Task: Trained 6 classes for electronic component identification

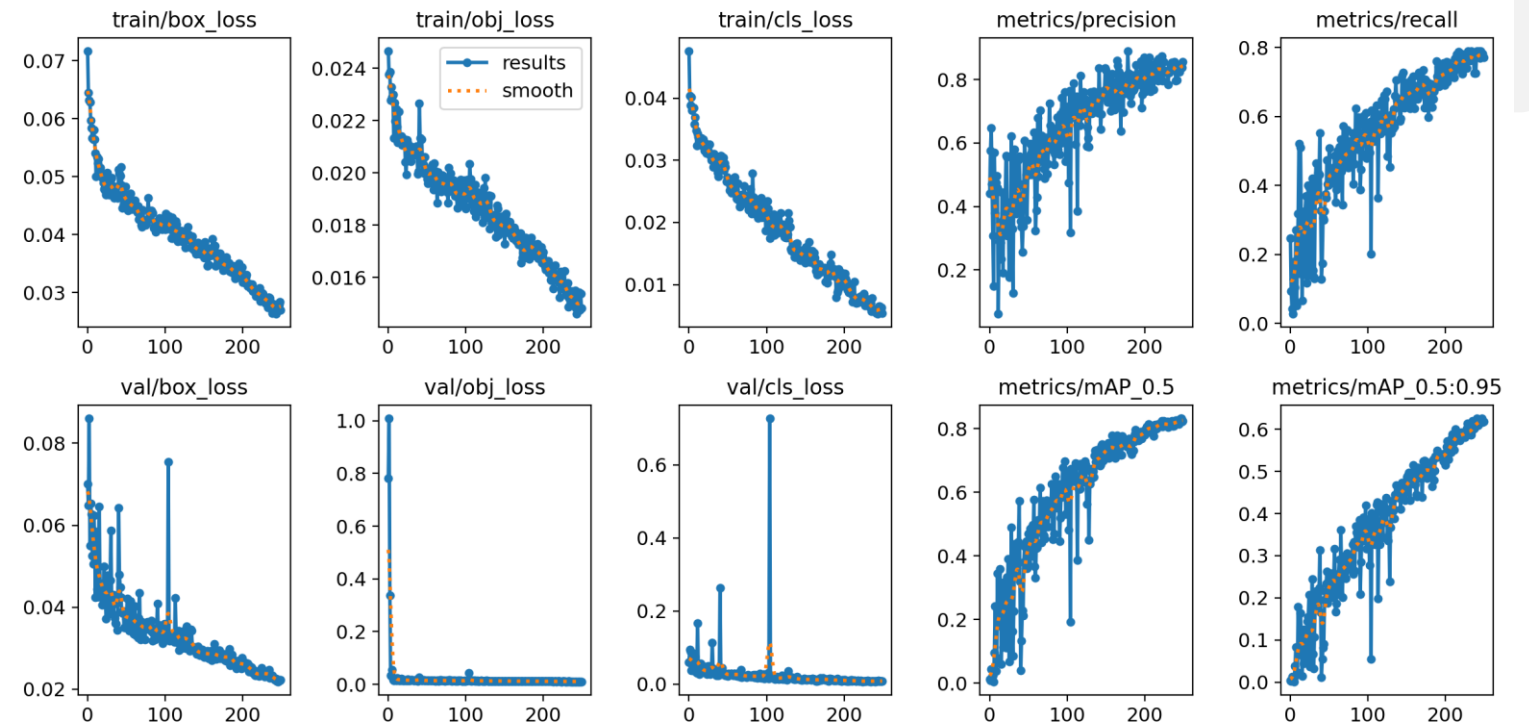


Source: Own  
Creation.

# IloT – Quality Control - CNN



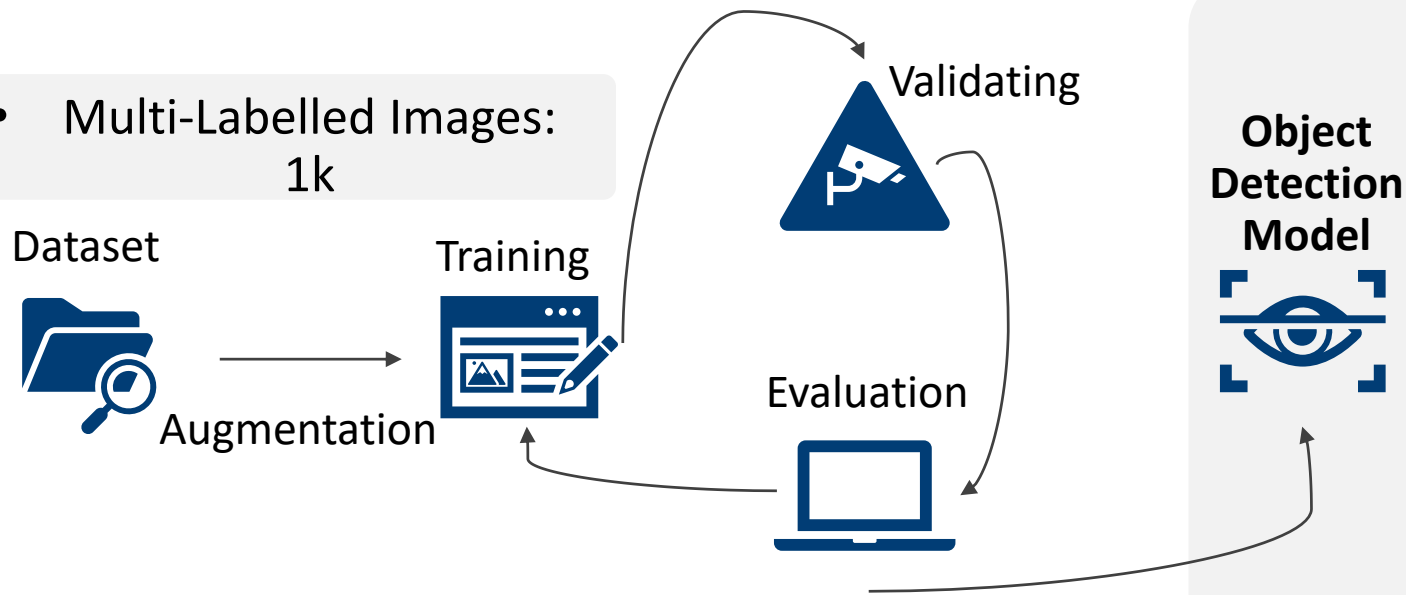
- CNN Algorithm for Object Detection Task: Trained 6 classes for electronic component identification



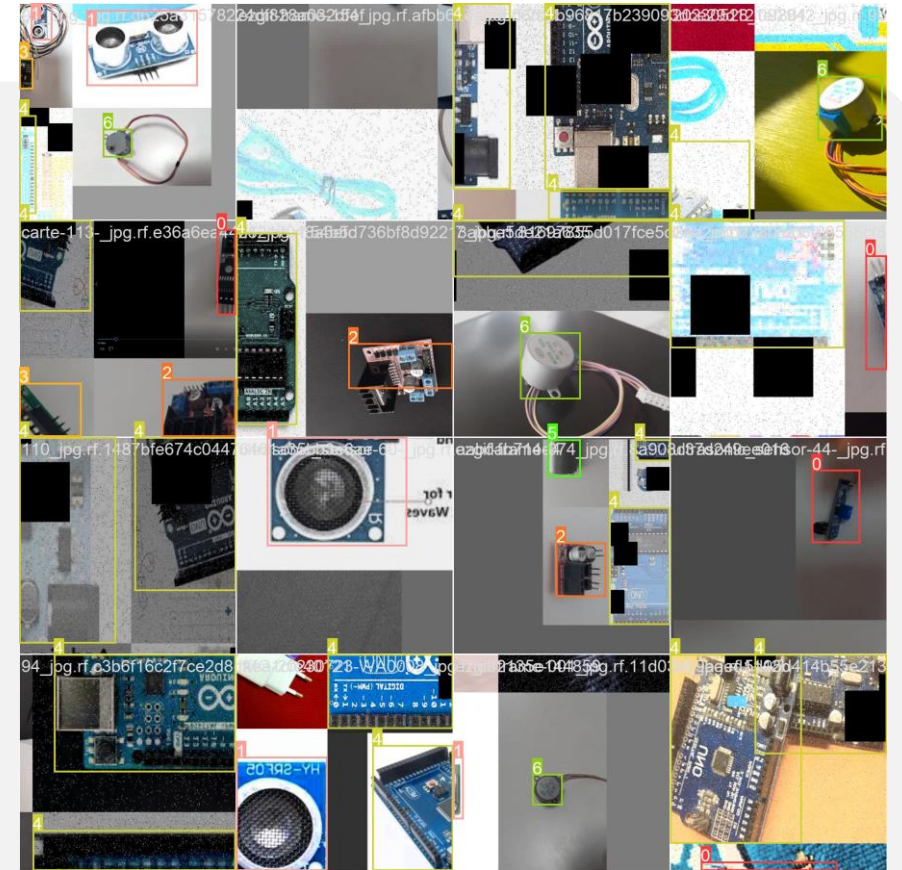
Source: Own Creation.

# Object Detection Methodology

- Multi-Labelled Images:  
1k

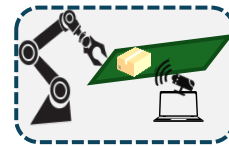


Source: Own Creation.



# IloT – Quality Control Main Sources Code

```
defect.py 3.1M, M
defect.py > ...
28 yoloV5S_paddle_model # PADDLEPaddle
29
30
31 import argparse
32 import csv
33 import os
34 import platform
35 import sys
36 from pathlib import Path
37
38 import torch
39 import paho.mqtt.client as mqtt
40
41 # MQTT Variables
42 MQTT_HOST = "192.168.0.59"
43 MQTT_PORT = 1883
44 MQTT_KEEPALIVE_INTERVAL = 5
45 MQTT_TOPIC = "bin/updatescam"
46
47 # on_connect event Handler
48 def on_connect(mosq, obj, rc):
49     print("Connected to MQTT Broker")
50
51 # on_publish event Handler
52 def on_publish(client, userdata, mid):
53     print("Message Published...")
54
55 # Initiate MQTT Client
56 mqttc = mqtt.Client()
57
58 # Register Event Handlers
59 mqttc.on_publish = on_publish
60 mqttc.on_connect = on_connect
61
62 # Connect with MQTT Broker
63 mqttc.connect(MQTT_HOST, MQTT_PORT, MQTT_KEEPALIVE_INTERVAL)
64
65 FILE = Path(__file__).resolve()
66 ROOT = FILE.parents[0] # YOLOV5 root directory
67 if str(ROOT) not in sys.path:
68     sys.path.append(str(ROOT)) # add ROOT to PATH
69 ROOT = Path(os.path.relpath(ROOT, Path.cwd())) # relative
70
71 from ultralytics.utils.plotting import Annotator, colors, save_one_box
72
73 from models.common import DetectMultiBackend
74 from utils.dataloaders import IMG_FORMATS, VID_FORMATS, LoadImages, LoadScreenshots, LoadStreams
75 from utils.general import (LOGGER, Profile, check_file, check_img_size, check_imshow, check_requirements, colorstr, cv2,
76                             increment_path, non_max_suppression, print_args, scale_boxes, strip_optimizer, xyxy2xywh)
77 from utils.torch_utils import select_device, smart_inference_mode
78
79 @smart_inference_mode()
80 def run(
```



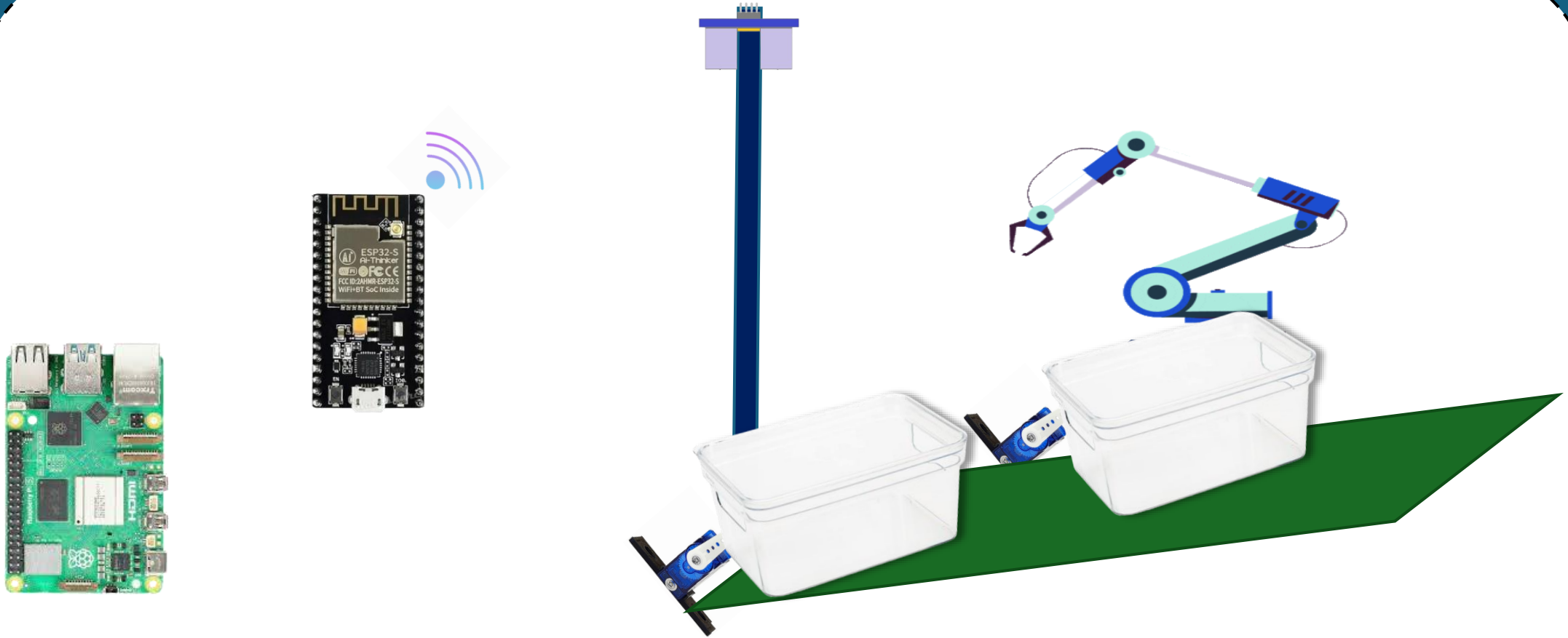
MQTT Client  
Quality Control



```
defect.py 3.1M, M
defect.py > ...
82 def run(
83     weights=ROOT / 'yolov5s.pt', # model path or triton URL
84     source=ROOT / 'data/images', # file/dir/URL/glob/screen/0(webcam)
85     data=ROOT / 'data/coco128.yaml', # (optional) dataset.yaml path
86     imgsz=[640], # inference size h,w
87     conf_thres=0.25, # confidence threshold
88     iou_thres=0.45, # NMS IOU threshold
89     max_det=1000, # maximum detections per image
90     device='', # cuda device, i.e. 0 or 0,1,2,3 or cpu
91     view_img=False, # show results
92     save_txt=False, # save results to *.txt
93     save_csv=False, # save results in CSV format
94     save_conf=False, # save confidences in --save-txt labels
95     save_crop=False, # save cropped prediction boxes
96     nosave=False, # do not save images/videos
97     classes=(), # filter by class: --classes 0, or --classes 0 2 3
98     agnostic_nms=False, # class-agnostic NMS
99     augment=False, # augmented inference
100     visualize=False, # visualize features
101     update=False, # update all models
102     project=ROOT / 'runs/detect', # save results to project/name
103     name='exp', # save results to project/name
104     exist_ok=False, # existing project/name ok, do not increment
105     line_thickness=3, # bounding box thickness (pixels)
106     hide_labels=False, # hide labels
107     hide_conf=False, # hide confidences
108     half=False, # use FP16 half-precision inference
109     dnn=None, # use OpenCV DNN for ONNX inference
110     vid_stride=1, # video frame-rate stride
111     opt=argparse.Namespace()
112 ):
113     # Load model
114     device = select_device(device)
115     model = DetectMultiBackend(weights, device=device, verbose=False)
116     stride = max(model.stride.values)
117     size = (imgsz[0] + stride - 1) // stride * stride
118     s = f'{size}x{size}'
119     # Pre-process
120     img = cv2.imread(source)
121     img = cv2.resize(img, size, interpolation=cv2.INTER_LINEAR)
122     img = img.transpose(2, 0, 1)
123     img = img.astype(np.float32)
124     img = img / 255.
125     # Inference
126     times = [time.perf_counter()]
127     with torch.no_grad():
128         pred = model.predict(img)
129     times.append(time.perf_counter())
130     # Post-process
131     pred = non_max_suppression(pred, conf_thres, iou_thres, classes, agnostic_nms, max_det=max_det)
132     # Results
133     s = f'{s}\n{len(list(save_dir.glob('labels/*.txt')))} labels saved to {save_dir / 'labels'}' if save_txt else ''
134     # Print results
135     LOGGER.info(f'Speed: %.1fms pre-process, %.1fms inference, %.1fms NMS per image at shape {(1, 3, *imgsz)}' % tuple(times))
136     if update:
137         strip_optimizer(weights[0]) # update model (to fix SourceChangeWarning)
138
139 def parse_opt():
140     parser = argparse.ArgumentParser()
141     parser.add_argument('--weights', nargs='+', type=str, default=ROOT / 'yolov5s.pt', help='model path or triton URL')
142     parser.add_argument('--source', type=str, default=ROOT / 'data/images', help='file/dir/URL/glob/screen/0(webcam)')
143     parser.add_argument('--data', type=str, default=ROOT / 'data/coco128.yaml', help='(optional) dataset.yaml path')
144     parser.add_argument('--imgsz', '--img', '--img-size', nargs='+', type=int, default=[640], help='inference size h,w')
145     parser.add_argument('--conf-thres', type=float, default=0.25, help='confidence threshold')
146     parser.add_argument('--iou-thres', type=float, default=0.45, help='NMS IOU threshold')
147     parser.add_argument('--max-det', type=int, default=1000, help='maximum detections per image')
148     parser.add_argument('--device', default='', help='cuda device, i.e. 0 or 0,1,2,3 or cpu')
149     parser.add_argument('--view-img', action='store_true', help='show results')
150     parser.add_argument('--save-txt', action='store_true', help='save results to *.txt')
151     parser.add_argument('--save-csv', action='store_true', help='save results in CSV format')
152     parser.add_argument('--save-conf', action='store_true', help='save confidences in --save-txt labels')
153     parser.add_argument('--save-crop', action='store_true', help='save cropped prediction boxes')
154     parser.add_argument('--nosave', action='store_true', help='do not save images/videos')
155     parser.add_argument('--classes', nargs='+', type=int, help='filter by class: --classes 0, or --classes 0 2 3')
156     parser.add_argument('--agnostic-nms', action='store_true', help='class-agnostic NMS')
157     parser.add_argument('--augment', action='store_true', help='augmented inference')
158     parser.add_argument('--visualize', action='store_true', help='visualize features')
159     parser.add_argument('--update', action='store_true', help='update all models')
160     parser.add_argument('--project', default=ROOT / 'runs/detect', help='save results to project/name')
161     parser.add_argument('--name', default='exp', help='save results to project/name')
162     parser.add_argument('--exist-ok', action='store_true', help='existing project/name ok, do not increment')
163     parser.add_argument('--line-thickness', default=3, type=int, help='bounding box thickness (pixels)')
164     parser.add_argument('--hide-labels', default=False, action='store_true', help='hide labels')
165     parser.add_argument('--hide-conf', default=False, action='store_true', help='hide confidences')
166     parser.add_argument('--half', action='store_true', help='use FP16 half-precision inference')
167     parser.add_argument('--dnn', action='store_true', help='use OpenCV DNN for ONNX inference')
168     parser.add_argument('--vid-stride', type=int, default=1, help='video frame-rate stride')
169     opt = parser.parse_args()
170     opt.imgsz *= 2 if len(opt.imgsz) == 1 else 1 # expand
171     print_args(vars(opt))
172     return opt
173
174 def main(opt):
175     check_requirements(ROOT / 'requirements.txt', exclude=('tensorboard', 'thop'))
176     run(**vars(opt))
177
178 if __name__ == '__main__':
179     opt = parse_opt()
180     main(opt)
```

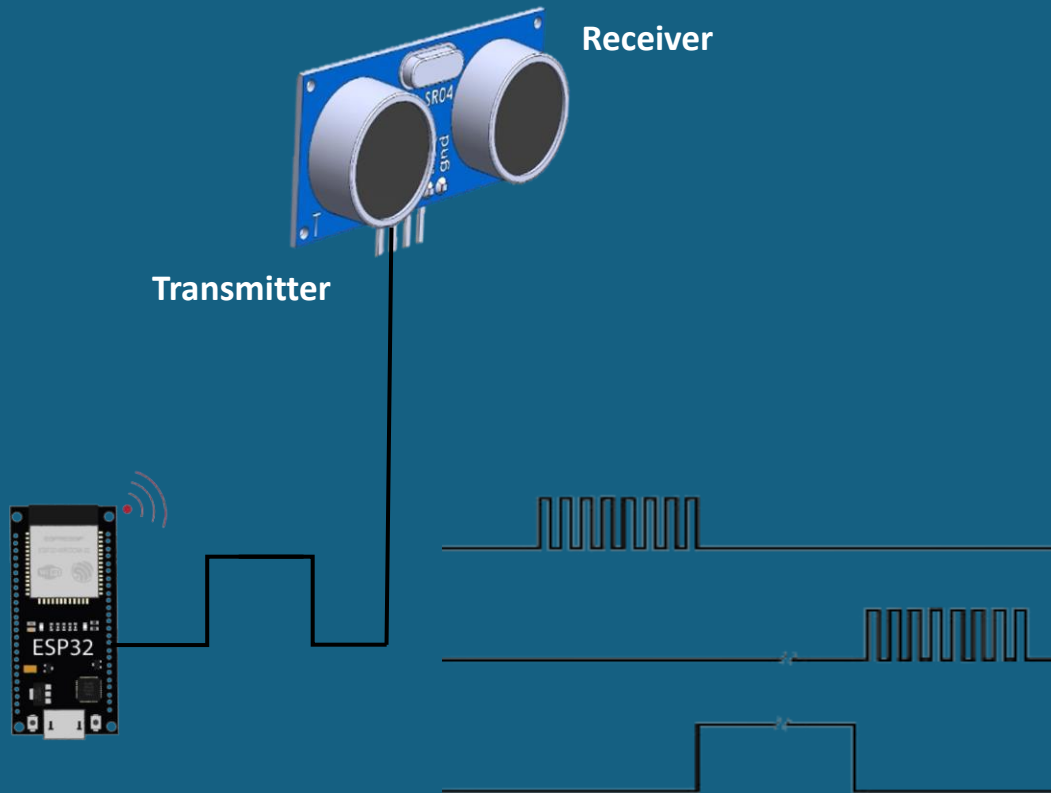


# Stock Management with MQTT



# Used Components

## Ultra Sonic distance Sensor



Echo pulse high time

It is also total travel time

$$\text{Distance} = [340 \text{ m/s} * \text{Echo pulse high time}] / 2$$





# Used Components

## Servo motors

- First Response
- Precise Control
- Programmable



## ESP32

- Smaller Size
- Widely available
- Less cost
- Easy to use



# Final implementation of Stock Management

Output Serial Monitor X

Message (Enter to send message to 'ESP32 Dev Module' on 'COM6')

```
Attempting MQTT connection...connected
Distance >= 10 cm: Sending 1
Distance >= 10 cm: Sending 1
Distance >= 10 cm: Sending 1
Distance < 10 cm: Sending 0
Attempting MQTT connection...connected
Distance < 10 cm: Sending 0
Distance < 10 cm: Sending 0
Distance < 10 cm: Sending 0
Distance >= 10 cm: Sending 1
Attempting MQTT connection...
```

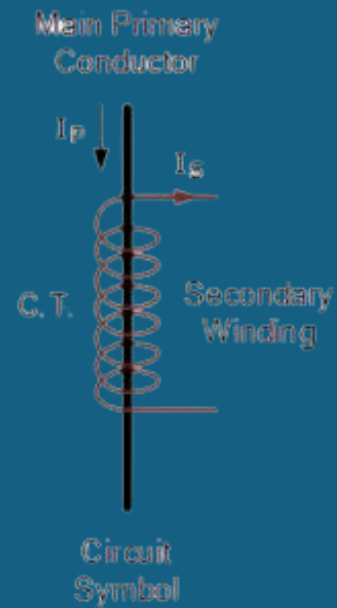
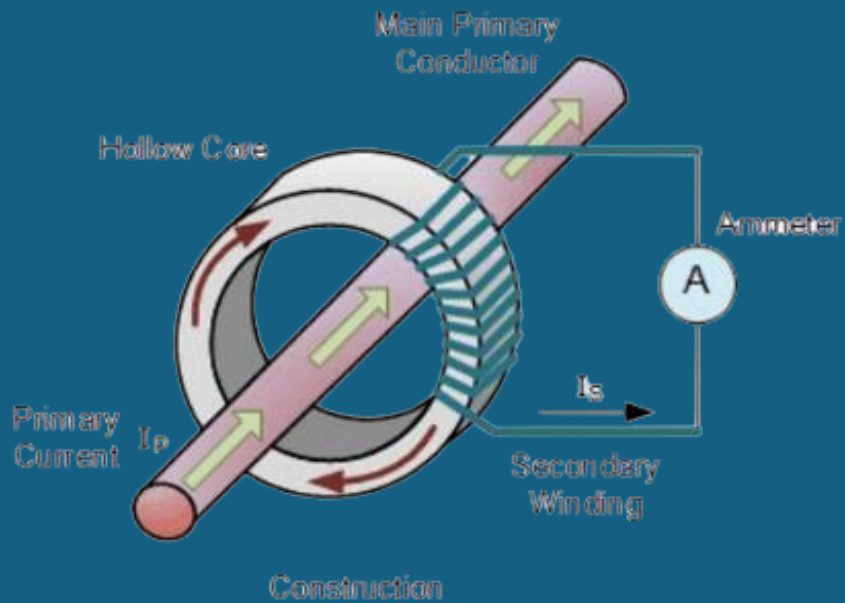
# Power Monitoring and controlling system



# Used Components

## SCT-013 Current Sensor

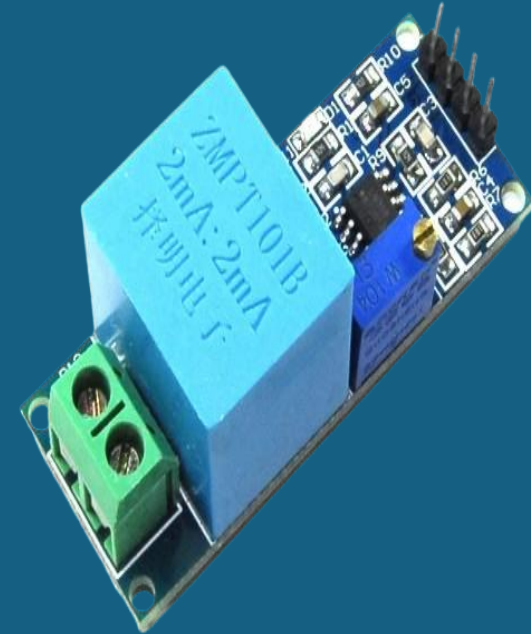
- 0-30A AC Current
- -25 °C~+70 °C
- Low price



# Used Components

## ZMPT101B AC Single Phase Voltage Sensor

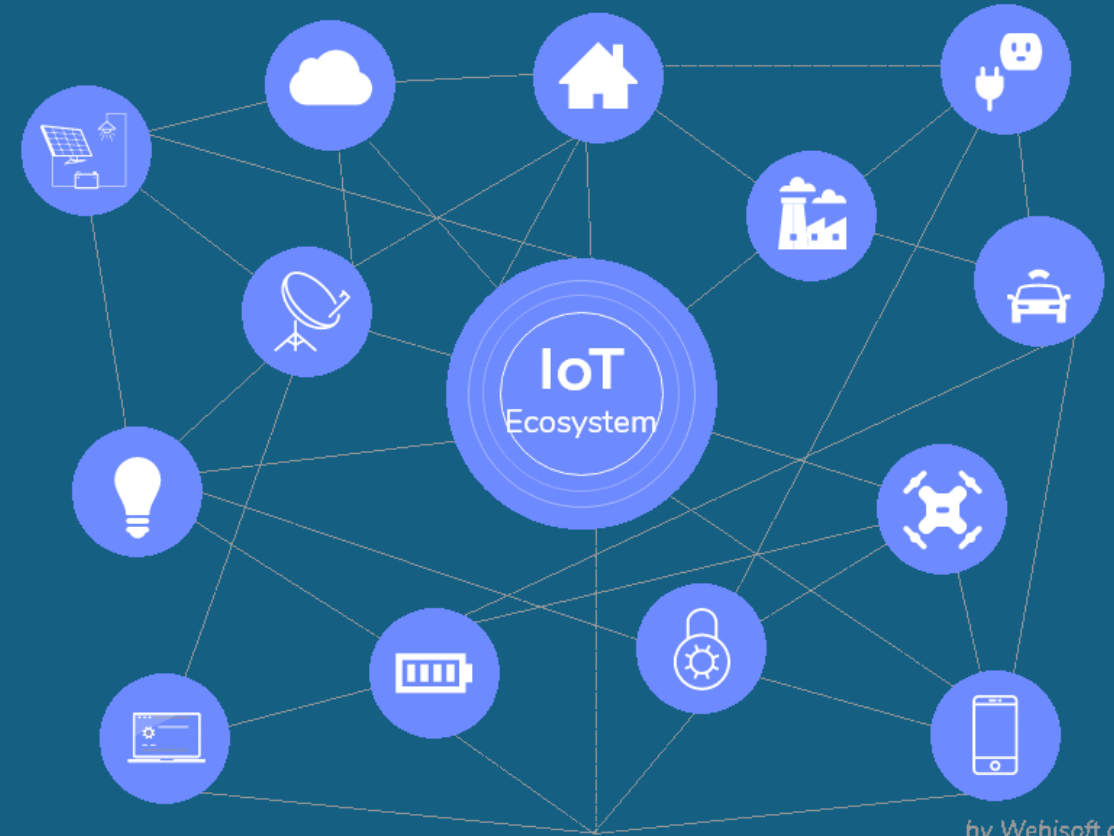
- Voltage up to 250 volts can be measured
- $-40\text{ }^{\circ}\text{C} \sim +70\text{ }^{\circ}\text{C}$
- Supply voltage 5 volts to 30 volts



# Used Components

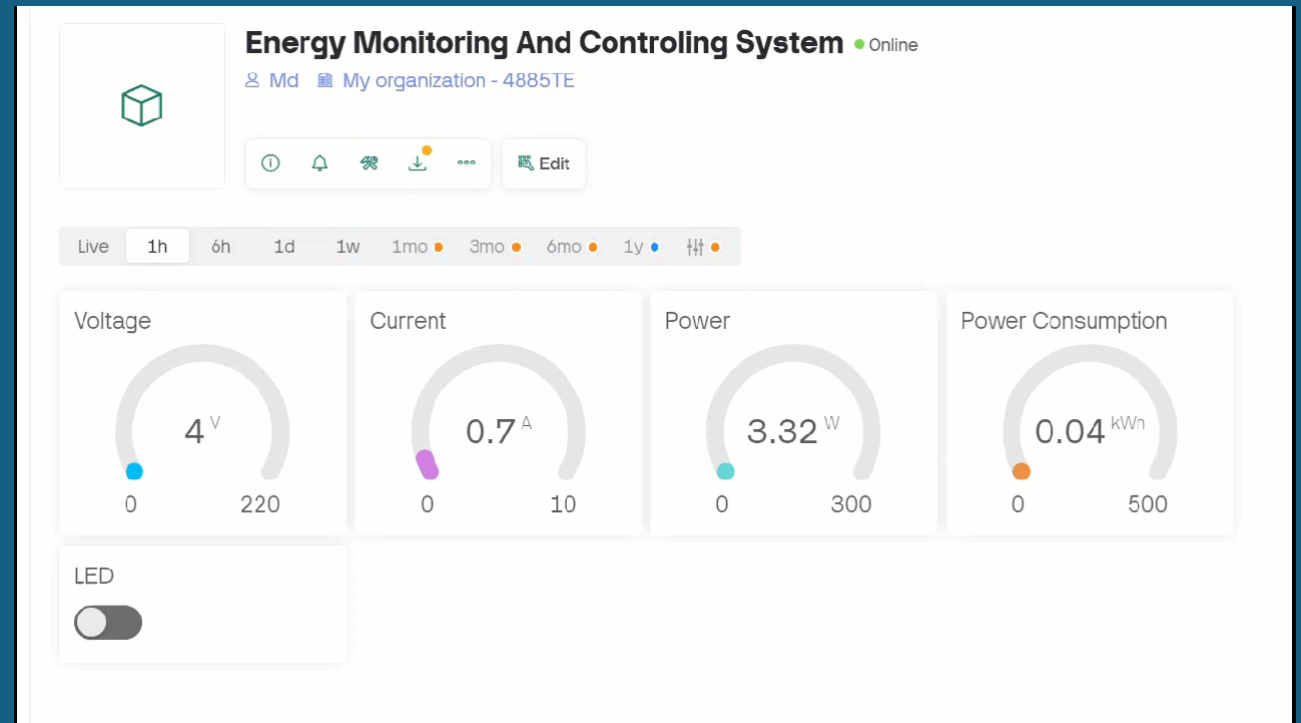
## Blynk IOT

- Easy to use
- Publishing and subscribing both possible
- Free for two projects



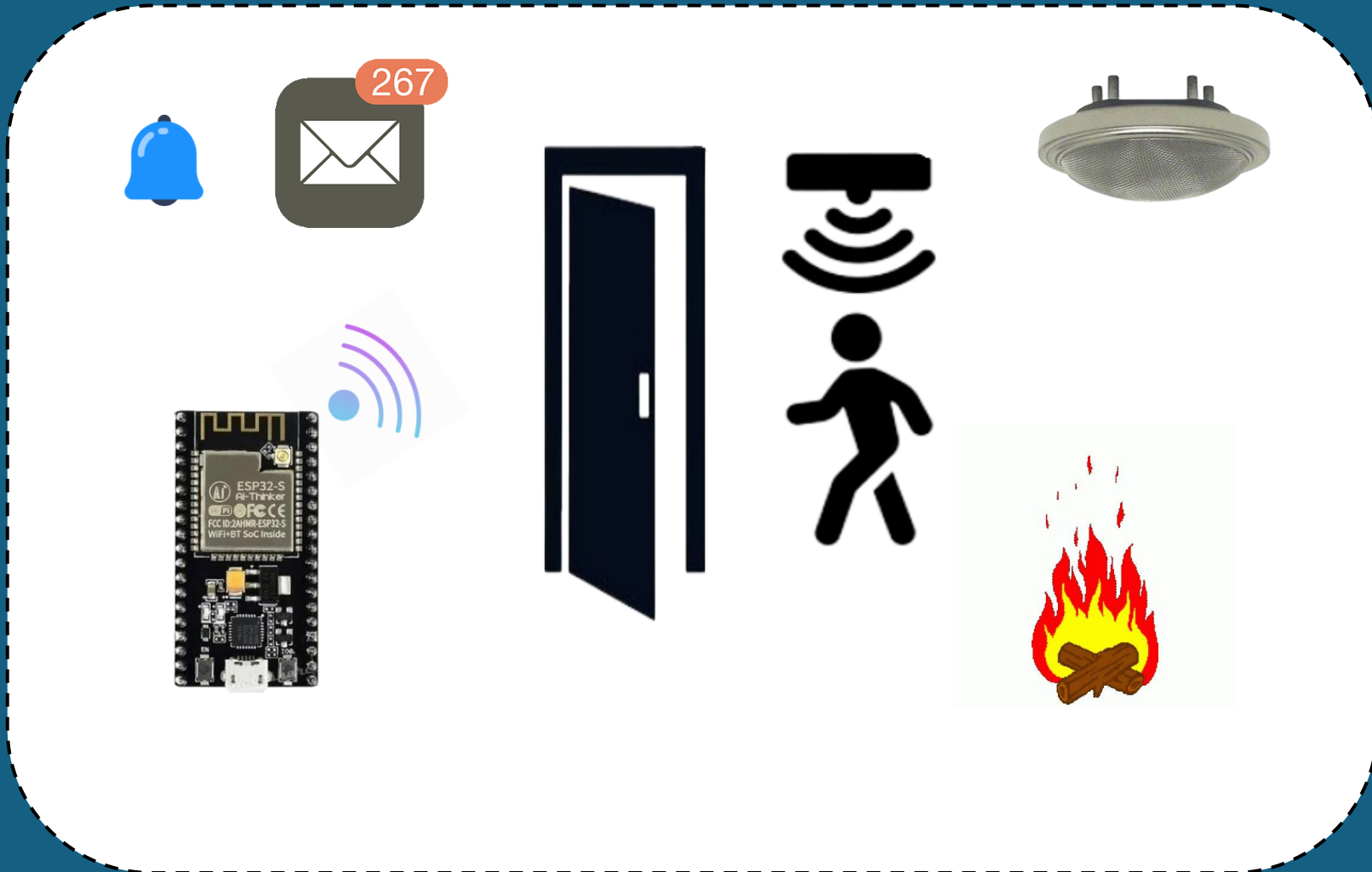
# Final implementation of power monitoring and control system

Output Serial Monitor x			
Message (Enter to send message to 'ESP32 Dev Module' on 'COM6')			
Vrms: 88.26V	Irms: 1.0277A	Power: 90.7128W	kWh: 0.03471kWh
Vrms: 75.14V	Irms: 1.1272A	Power: 84.6940W	kWh: 0.03483kWh
Vrms: 69.57V	Irms: 0.8916A	Power: 62.0299W	kWh: 0.03491kWh
Vrms: 62.87V	Irms: 0.9025A	Power: 56.7424W	kWh: 0.03499kWh
Vrms: 56.82V	Irms: 0.7410A	Power: 42.1066W	kWh: 0.03505kWh
Vrms: 53.11V	Irms: 0.7856A	Power: 41.7225W	kWh: 0.03511kWh
Vrms: 50.04V	Irms: 0.8247A	Power: 41.2697W	kWh: 0.03516kWh
Vrms: 48.01V	Irms: 0.7705A	Power: 36.9967W	kWh: 0.03522kWh
Vrms: 46.52V	Irms: 0.7765A	Power: 36.1255W	kWh: 0.03527kWh
Vrms: 44.87V	Irms: 0.7277A	Power: 32.6547W	kWh: 0.03531kWh
Vrms: 43.22V	Irms: 0.6872A	Power: 29.6976W	kWh: 0.03535kWh





# Safety and security system



# Used Components

## MQ2 Gas Sensor

- Sensitivity
- Only 5V operating voltage
- Low Cost



# Final implementation of power monitoring and control system

