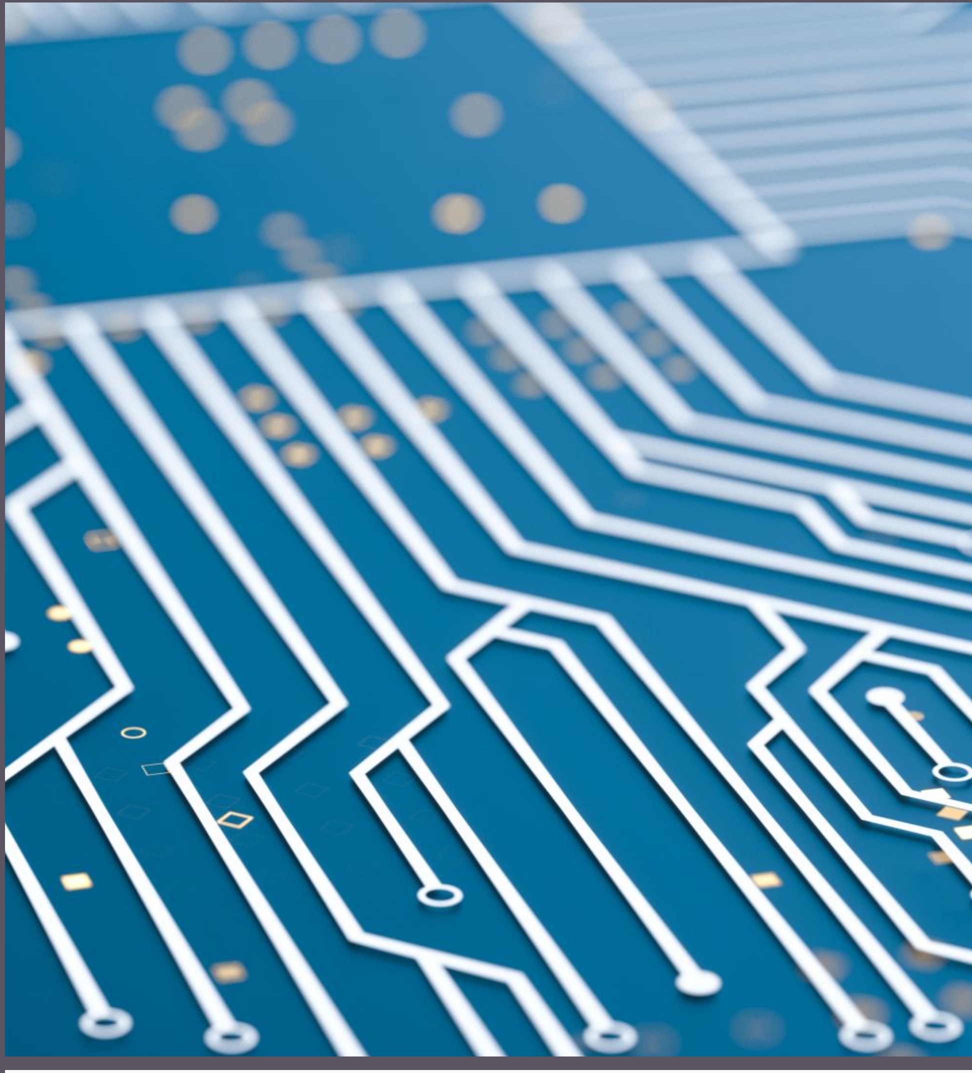
HOME SECURITY SYSTEM

ECE202 – EMBEDDED SYSTEMS HARDWARE



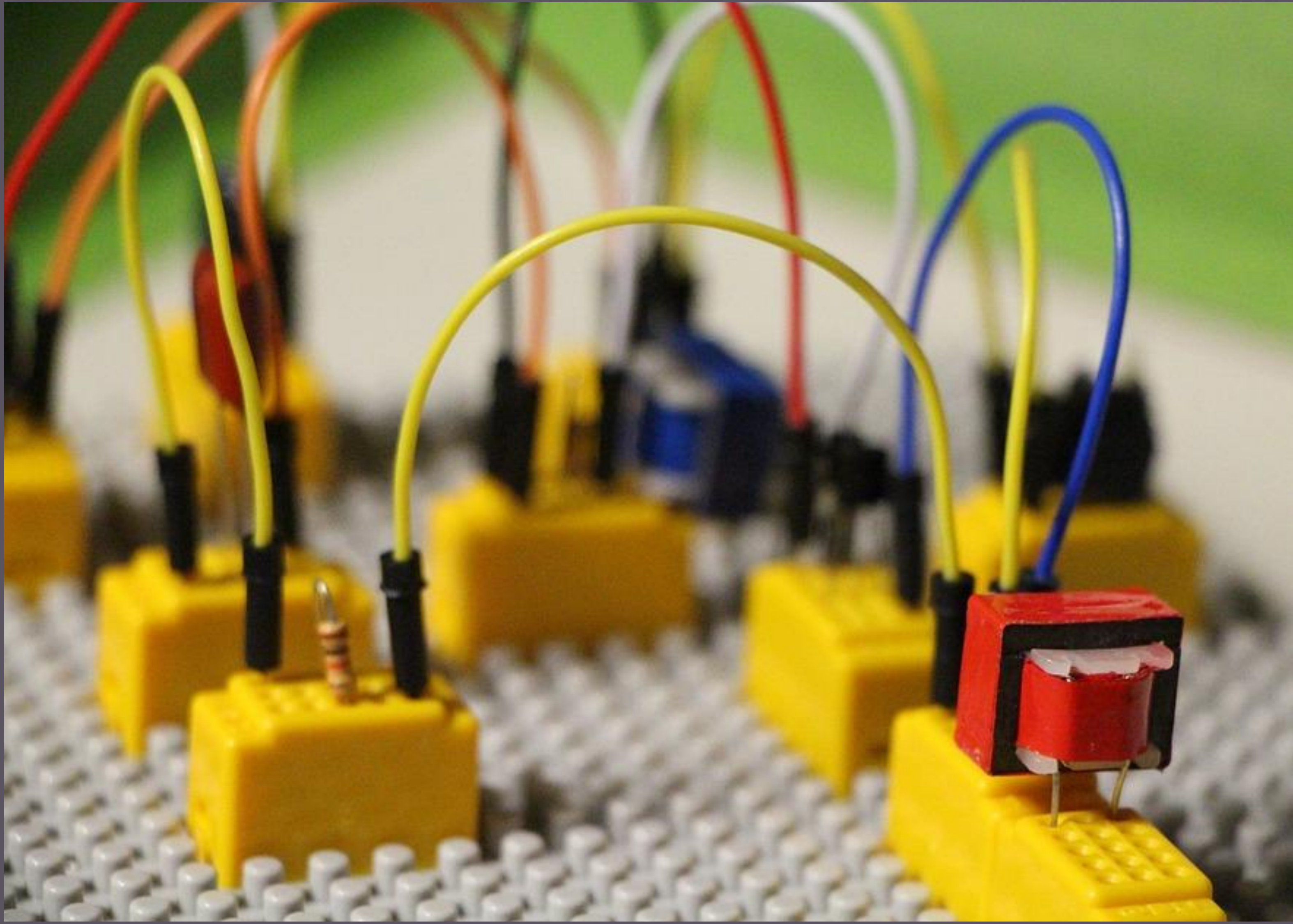
TEAM MEMBERS

GUNJA JASWANTH

D.Y. PRANEETH REDDY

ATHARVAN

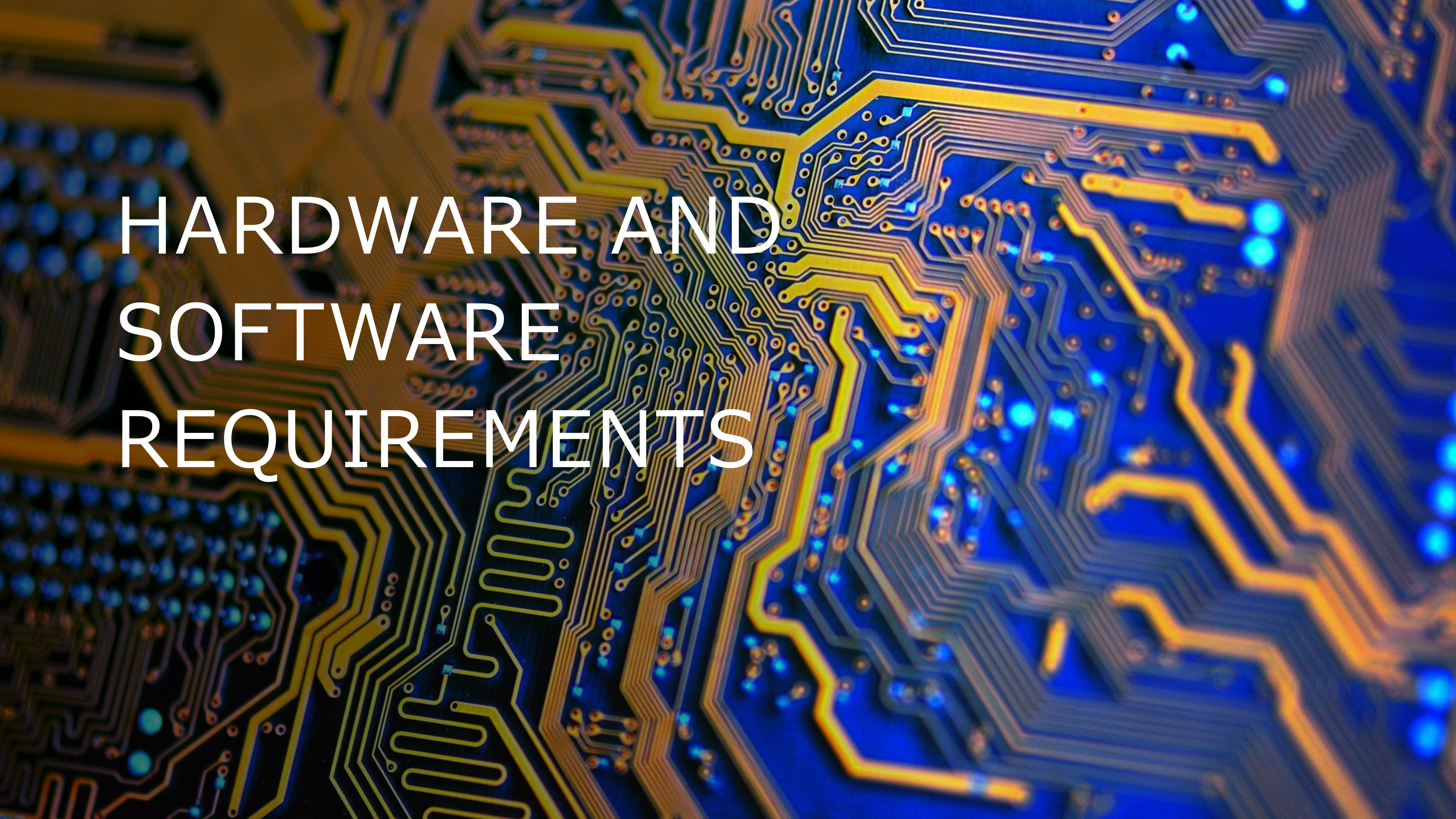
SRI KRISHNA



PROJECT OVERVIEW

PROJECT OVERVIEW

The project aims to detect motion and provide an alert using an STM32 microcontroller. It incorporates a PIR sensor for motion detection and triggers a buzzer to sound an alert when motion is detected. Additionally, a laptop camera is utilized to monitor the surroundings and capture photos whenever motion is detected.



HARDWARE AND SOFTWARE REQUIREMENTS

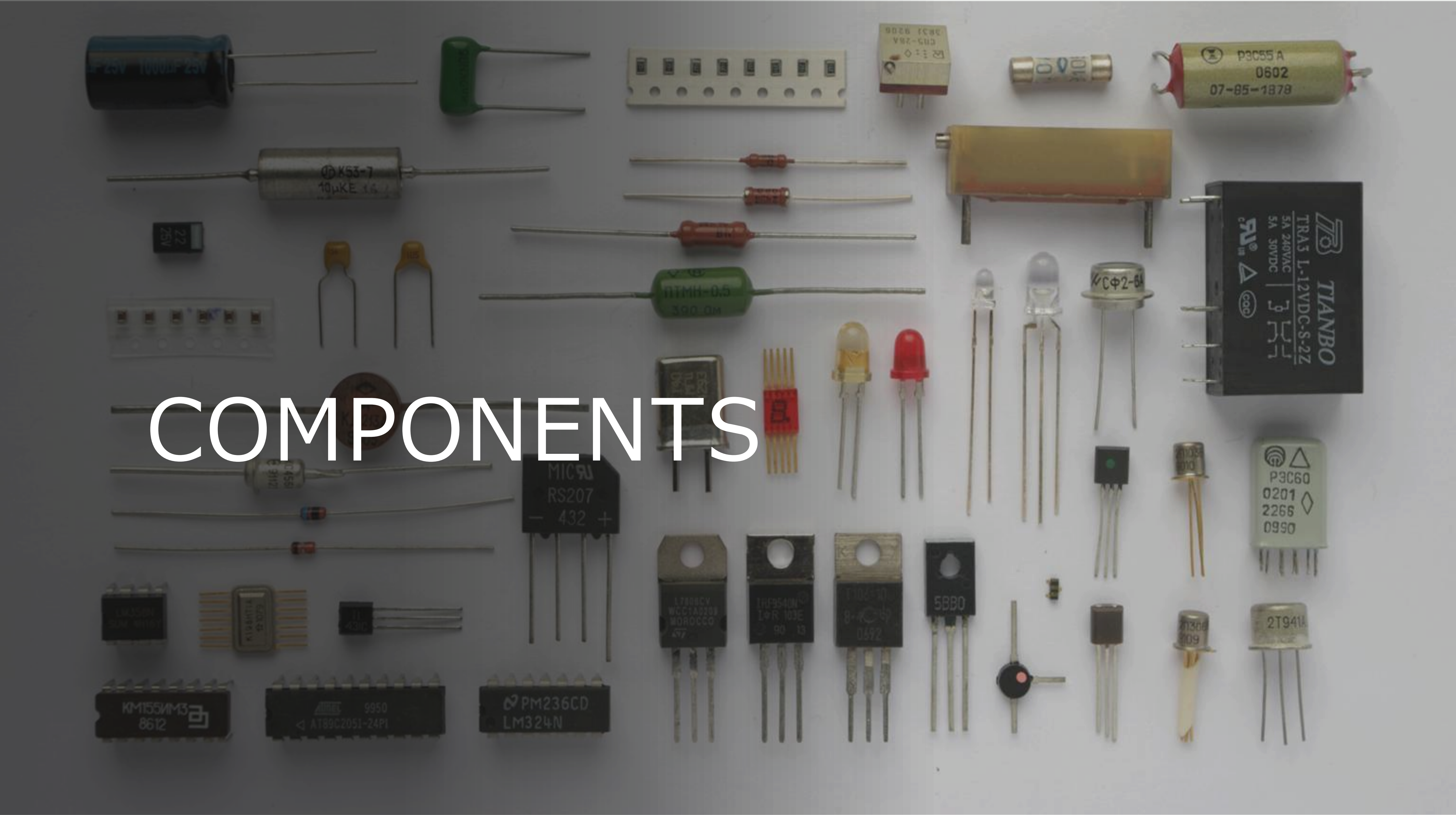
HARDWARE AND SOFTWARE REQUIREMENTS

SOFTWARE REQUIREMENTS:

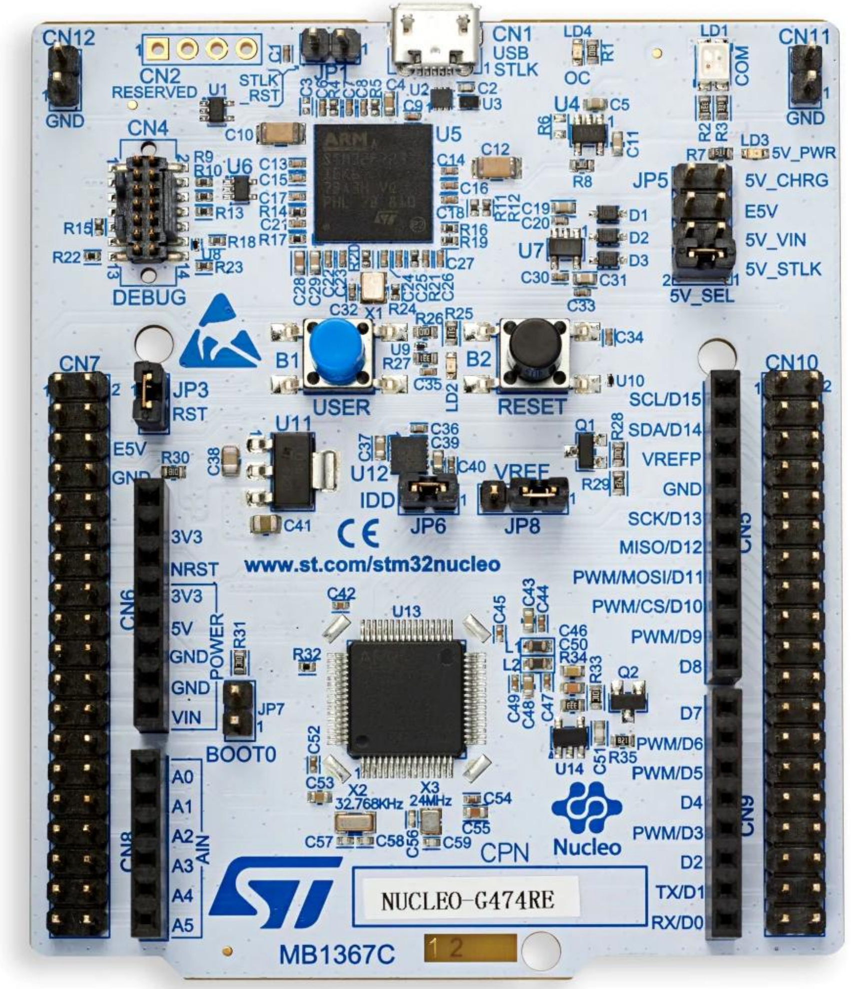
- ➢ STM32 IDE
- ➢ Python Compiler

HARDWARE REQUIREMENTS:

- ➢ STM32 Microcontroller
- ➢ PIR Sensor
- ➢ Buzzer
- ➢ Connecting Wires
- ➢ Breadboard
- ➢ Laptop Camera


1000μF 25V
K53-7
10μKE
P3C55 A
0602
07-85-1878
TIANBO
TRA3 L-12VDC-S-2Z
5A 240VAC
5A 30VDC
ПТМН-0,5
390 Ом
СФ2-6А
MIC
RS207
− 432 +
P3C60
0201
2266
0990
L7808CV
IRF9540N
5BB0
2T941A
КМ155ИМ3
8612
9950
AT89C2051-24PI
PM236CD
LM324N

COMPONENTS



STM32 MICROCONTROLLER

The STM32 microcontroller acts as the central unit, processing signals from the PIR sensor, activating the buzzer for alerts, and coordinating with the laptop camera to capture images. It ensures seamless communication between components, real-time operation, and energy efficiency, making it vital for the system's functionality.

PIR SENSOR

PIR sensors detect motion by sensing changes in infrared radiation caused by warm objects, like humans. They have pyroelectric elements that detect these changes, triggering a signal to activate devices like buzzers, lights, or alarms. PIR sensors are commonly used in security systems, automatic lighting, and smart devices. It plays a key role in the project.

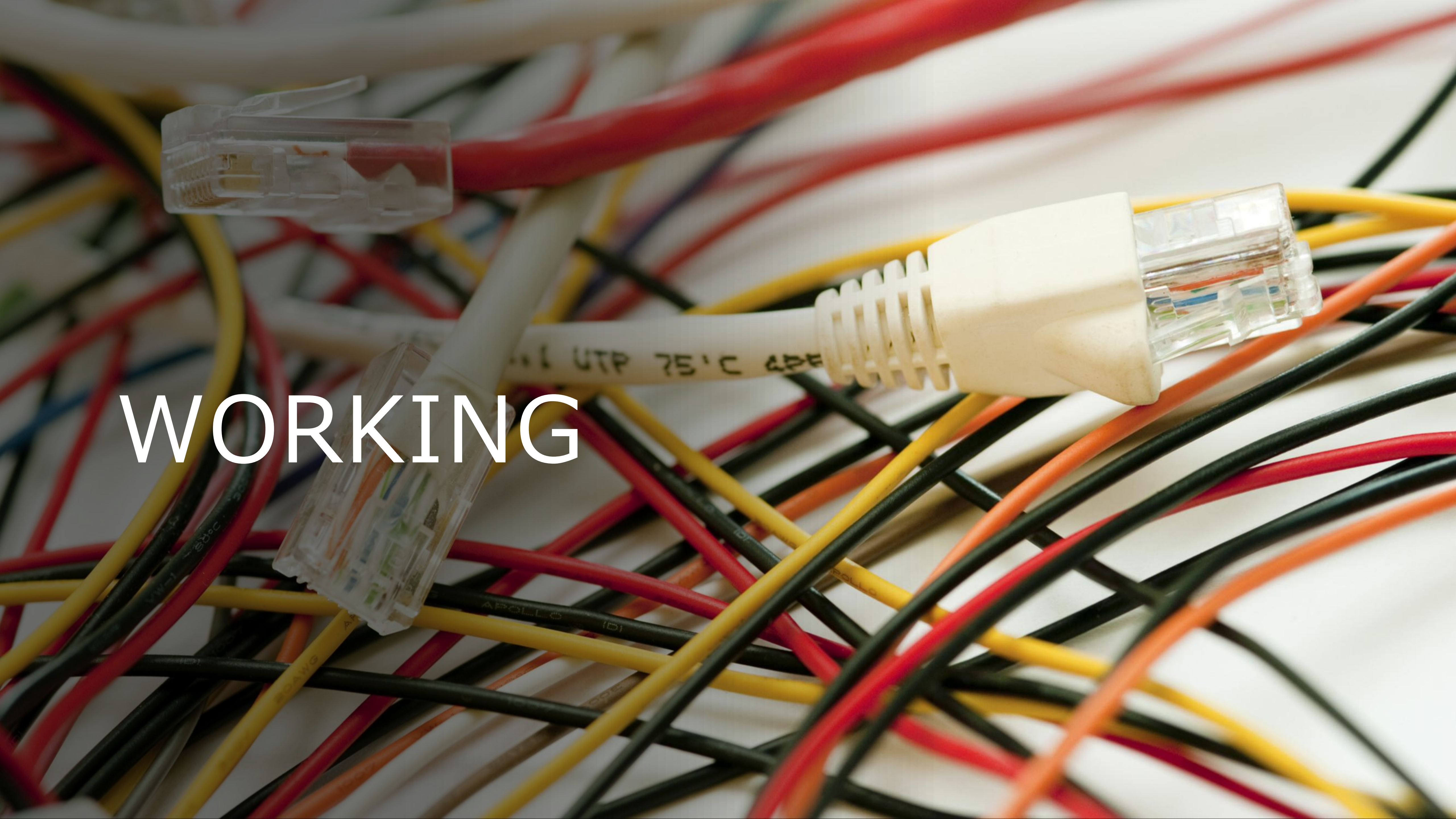
BUZZER

A buzzer is used to indicate whether motion is detected. When body movement or motion is sensed, the buzzer emits a sound, signaling that motion has occurred. If no motion is detected within the sensor's range, the buzzer remains silent. The sensor sends the signal to the buzzer, triggering the sound based on the presence or absence of motion.

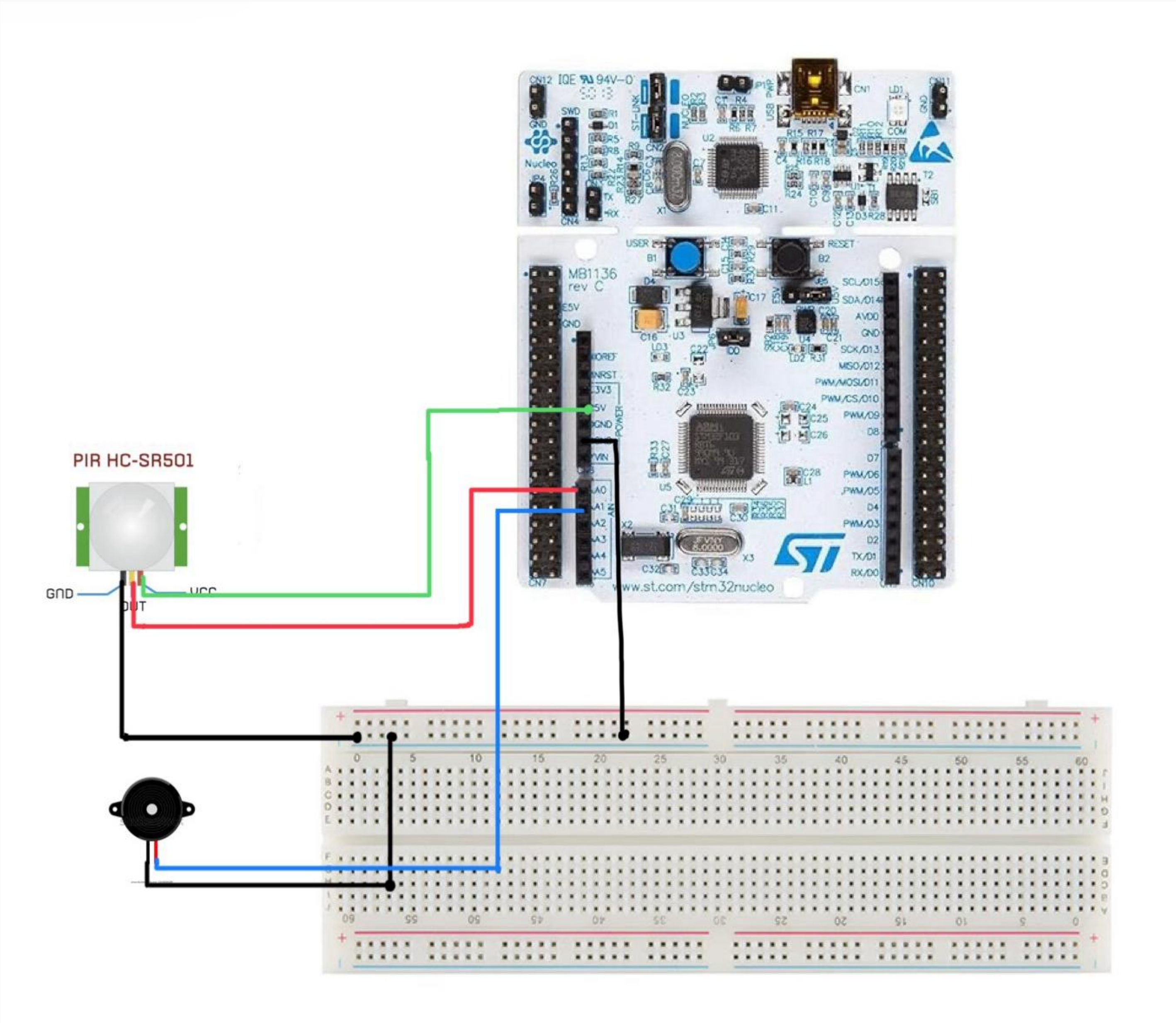
LAPTOP CAMERA

The laptop camera captures snapshots of objects when motion is detected, saving the images while simultaneously monitoring for movement. This process is managed by Python code, which efficiently processes the data.

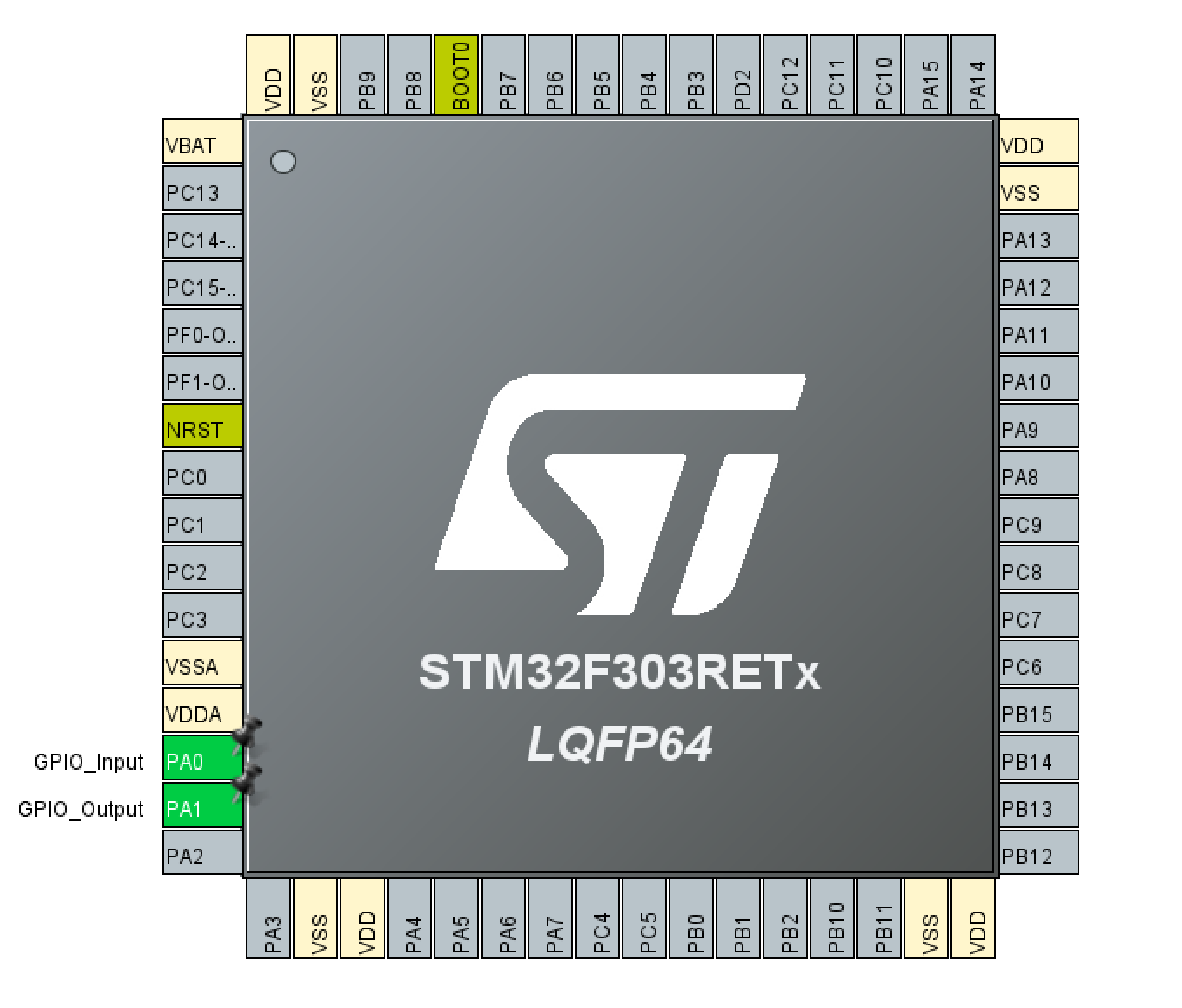
WORKING



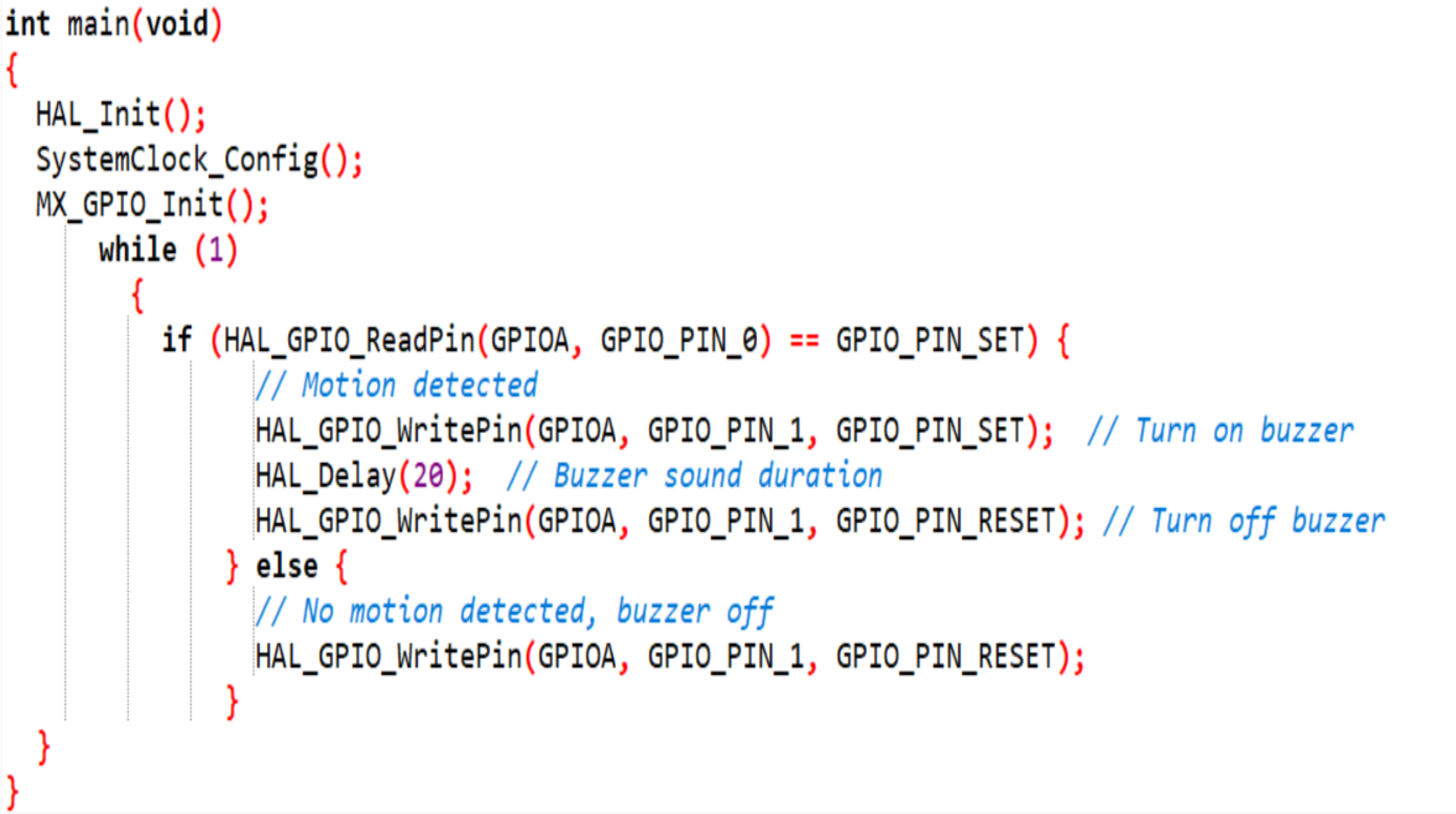
PIN CONNECTIONS



CONFIGURATION IN STM32 IDE



STM CODE



```
int main(void)
{
  HAL_Init();
  SystemClock_Config();
  MX_GPIO_Init();
      while (1)
        {
          if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0) == GPIO_PIN_SET) {
                // Motion detected
                HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_SET);  // Turn on buzzer
                HAL_Delay(20);  // Buzzer sound duration
                HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_RESET); // Turn off buzzer
              } else {
                // No motion detected, buzzer off
                HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_RESET);
              }
  }
}
```

PYTHON CODE

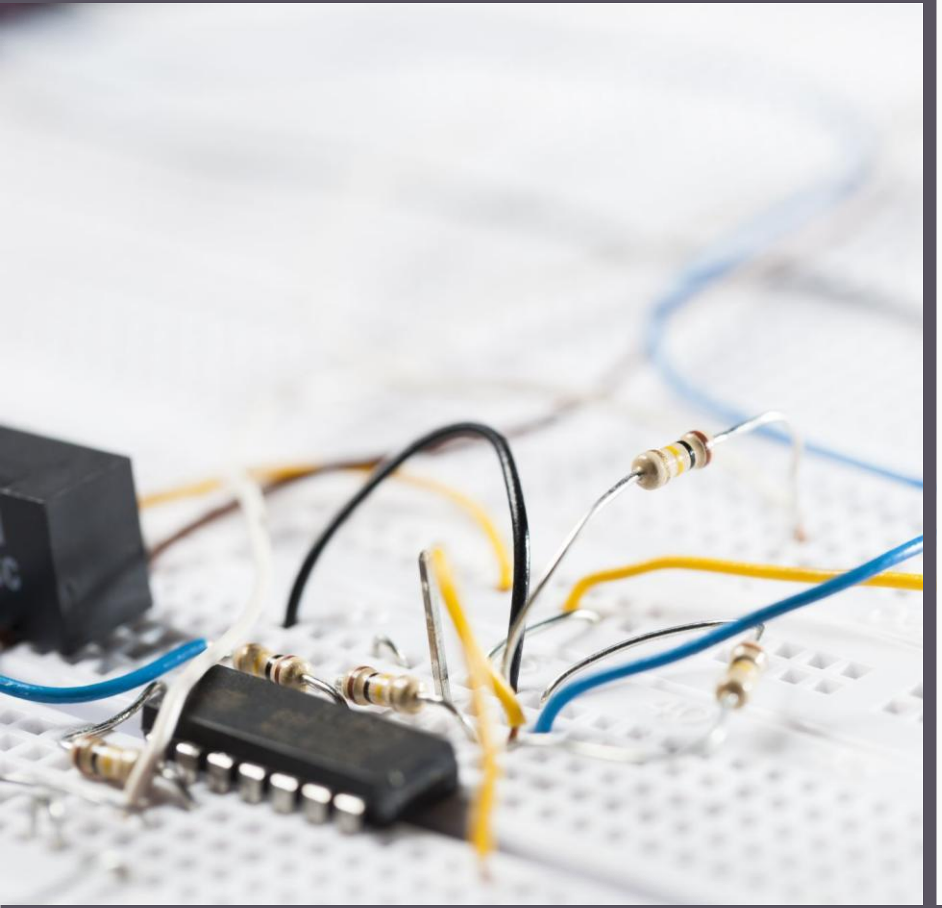
```
import cv2
import time
import datetime
cap = cv2.VideoCapture(0)  # camera intilization
previous_frame = None
while True:
    ret, frame = cap.read()
    if not ret:
        break
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)  # converting frame to grayscale
    gray_frame = cv2.GaussianBlur(gray_frame, (21, 21), 0)
    if previous_frame is None:  # comparing with the initial frame
        previous_frame = gray_frame
        continue

    frame_delta = cv2.absdiff(previous_frame, gray_frame) #basically detects the motion
    thresh_frame = cv2.threshold(frame_delta, 25, 255, cv2.THRESH_BINARY)[1]
    thresh_frame = cv2.dilate(thresh_frame, None, iterations=2)

    contours, _ = cv2.findContours(thresh_frame.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)  #movement detection
    motion_detected = False  # initially
    for contour in contours:
        if cv2.contourArea(contour) < 100:  # Ignore small movements
            continue
        motion_detected = True  # motion is detected
        (x, y, w, h) = cv2.boundingRect(contour)
        cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
    if motion_detected:  # if motion is detected it saves the images
        timestamp = datetime.datetime.now().strftime('%Y-%m-%d_%H-%M-%S')

        # Create a filename using the timestamp
        image_name = f"image_{timestamp}.jpg"
        cv2.imwrite(image_name, frame)

    cv2.imshow("Frame", frame)
    previous_frame = gray_frame
    if cv2.waitKey(1) & 0xFF == ord('q'):  # to exit press q
        break
cap.release()
cv2.destroyAllWindows() #closes al the windows upon pessing q
```



PROBLEMS FACED

- Initially, we were unsure about how to work with the sensor and how it functions.
- After debugging the code in the IDE, the sensor does not trigger an alert immediately when motion is detected(up to 5-8 seconds).
- Additionally, if motion occurs right after the buzzer has been activated, it doesn't trigger the buzzer again immediately(within 2-5 seconds of the first movement).

THANK YOU!

19
U5
C914 J7
C936
RP2 1
2
C935
C916
C934
C974
C976
C972
21
ASSEMBLY
OPTIONS