VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANASANGAMA", BELAGAVI : 590018, KARNATAKA, INDIA

Final Year Project Report on

"Raspberry Pi-Based Thermal and Optical Detection System with Automated Targeting and Real-Time Monitoring"

Submitted in Partial fulfilment for the award of
degree Of
Bachelor of Engineering In

ELECTRONICS AND COMMUNICATION ENGINEERING

By

Dhanya R Rao	**4NN21EC012**
Varsha Bhat K	**4NN21EC050**
Sanjan G	**4NN22EC411**
Yashwanth M	**4NN22EC413**

Under the Guidance of
Dr. Manjula A V
Assoc. Prof., Head
NIEIT, Mysuru-18

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

NIE INSTITUTE OF TECHNOLOGY

MYSORE-570018

2024-2025

ESTD-2008

NIE INSTITUTE OF TECHNOLOGY

#50 (part), Hootagalli Industrial area, Koorgalli Village, Mysuru-18

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

Certified that the project work entitled **“Raspberry Pi-Based Thermal and Optical Detection System with Automated Targeting and Real-Time Monitoring”** carried out by **Ms. Dhanya R Rao (4NN21EC012), Ms. Varsha Bhat K (4NN21EC050), Mr. Sanjan G (4NN22EC411), Mr. Yashwanth M (4NN22EC413)** Bonafide students of NIEIT, Mysuru in partial fulfilment for the award of **Bachelor of Engineering in Electronics and Communication** of the Visvesvaraya Technological University, Belgavi during the **year 2024-2025.** It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Signature of the guide	**Signature of the HOD**	**Signature of the principal**
Dr. Manjula A V	Dr. Manjula A V	Dr. Rohini Nagapadma

External Viva

Name of the examiners	Signature with date
1.	
2.	

ESTD-2008

NIE INSTITUTE OF TECHNOLOGY

#50 (part), Hootagalli Industrial area, Koorgalli Village, Mysuru-18

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

DECLARATION

We, Dhanya R Rao (4NN21EC012), Varsha Bhat K (4NN21EC050), Sanjan G (4NN22EC411), Yashwanth M (4NN22EC413) students of VII semester, **Department of Electronics and Communication Engineering, NIE Institute of Technology, Mysuru** declare that the project work titled **"Raspberry Pi-Based Thermal and Optical Detection System with Automated Targeting and Real-Time Monitoring"** has been successfully completed. This work is submitted to Visvesvaraya Technological University, Belgavi-590018, in partial fulfilment of the requirements for the award of Degree of Bachelor of Engineering in Department of Electronics and Communication Engineering during the academic year 2023- 2024. Further the matter embodied in the project report has not been submitted previously by anybody for the award of any degree to any university.

Place: Mysuru
Date:

	Name	USN	Signature with date
1.	Dhanya R Rao	4NN21EC012	
2.	Varsha Bhat K	4NN21EC050	
3.	Sanjan G	4NN22EC411	
4.	Yashwanth M	4NN22EC413	

ACKNOWLEDGEMENT

We sincerely owe our gratitude to all the person who helped us and guided us in completing this project.

We are thankful to Prof. **Dr. Rohini Nagapadma**, principal, NIE Institute of Technology, Mysuru for having supported us in our academic endeavors.

We take this opportunity to express our profound gratitude and deep regards to our guide **Dr Manjula A V**, Associate Professor & HOD, Department of Electronics and Communication Engineering, NIE Institute of Technology, Mysuru.

We are obliged to all **Teaching and Technical staff members** of **Department of Electronics and Communication Engineering, NIE Institute of Technology, Mysuru** for the valuable information providing by them in their respective fields. We are grateful for their co- operation during the period of our project.

Lastly, we thank almighty, our parents and friends for their constant encouragement without which this project would not be possible.

Dhanya R Rao	4NN21EC012
Varsha Bhat K	4NN21EC050
Sanjan G	4NN22EC411
Yashwanth M	4NN22EC413

ABSTRACT

The rapid growth of remote surveillance and security applications has fuelled the demand for advanced detection systems. This project introduces a Raspberry Pi-based system capable of both thermal and optical detection, featuring automated targeting and real-time monitoring capabilities. This system incorporates key components like infrared sensors, cameras, missile model, and a Raspberry Pi microcontroller for intelligent control and data processing. A user-friendly interface with a display provides real-time visualization and control of the system.

At the core of the system, the Raspberry Pi microcontroller processes data from the infrared sensors and camera, analyses images for potential threats, and controls the missile model to automatically track and launch amino at the detected targets. Infrared sensors provide valuable data on heat signatures, while the camera captures visual information for further analysis. The CNN enables precise and agile target tracking, ensuring continuous monitoring of detected threats.

The system's intuitive interface utilizes a display to present real-time video feeds, thermal imagery, and tracking information. User interaction is facilitated through intuitive controls for system configuration and target tracking adjustments. This design emphasizes real-time performance, accuracy, and adaptability, addressing critical aspects of security surveillance, including early threat detection, automated response, and continuous monitoring.

This Raspberry Pi-based detection system showcases the potential of embedded systems in advanced surveillance applications. By integrating thermal and optical sensing with automated targeting and real-time monitoring, it enhances security capabilities, enabling proactive threat assessment and efficient resource allocation.

TABLE OF CONTENTS

Chapter 4

SOFTWARE IMPLEMENTATION

Chapter 5

SYSTEM IMPLEMENTATION

Chapter 6

RESULTS AND CONCLUSION

LIST OF FIGURES

LIST OF TABLES

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANASANGAMA", BELAGAVI : 590018, KARNATAKA, INDIA

Final Year Project Report on

"Raspberry Pi-Based Thermal and Optical Detection System with Automated Targeting and Real-Time Monitoring"

Submitted in Partial fulfilment for the award of
degree Of
Bachelor of Engineering In

ELECTRONICS AND COMMUNICATION ENGINEERING
By

Dhanya R Rao	**4NN21EC012**
Varsha Bhat K	**4NN21EC050**
Sanjan G	**4NN22EC411**
Yashwanth M	**4NN22EC413**

Under the Guidance of
Dr. Manjula A V
Assoc. Prof. and Head

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
NIE INSTITUTE OF TECHNOLOGY
MYSORE-570018

2024-2025

ESTD-2008

NIE INSTITUTE OF TECHNOLOGY

#50 (part), Hootagalli Industrial area, Koorgalli Village, Mysuru-18

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

Certified that the project work entitled **"Raspberry Pi-Based Thermal and Optical Detection System with Automated Targeting and Real-Time Monitoring"** carried out by **Ms. Dhanya R Rao (4NN21EC012), Ms. Varsha Bhat K (4NN21EC050), Mr. Sanjan G (4NN22EC411), Mr. Yashwanth M (4NN22EC413)** Bonafide students of NIEIT, Mysuru in partial fulfilment for the award of **Bachelor of Engineering in Electronics and Communication** of the Visvesvaraya Technological University, Belgavi during the **year 2024-2025.** It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Signature of the guide	**Signature of the HOD**	**Signature of the principal**
Dr. Manjula A V	Dr. Manjula A V	Dr. Rohini Nagapadma

External Viva

Name of the examiners	Signature with date
1.	
2.	

ESTD-2008

NIE INSTITUTE OF TECHNOLOGY

#50 (part), Hootagalli Industrial area, Koorgalli Village, Mysuru-18

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

DECLARATION

We, Dhanya R Rao (4NN21EC012), Varsha Bhat K (4NN21EC050), Sanjan G (4NN22EC411), Yashwanth M (4NN22EC413) students of VII semester, **Department of Electronics and Communication Engineering, NIE Institute of Technology, Mysuru** declare that the project work titled **"Raspberry Pi-Based Thermal and Optical Detection System with Automated Targeting and Real-Time Monitoring"** has been successfully completed. This work is submitted to Visvesvaraya Technological University, Belgavi-590018, in partial fulfilment of the requirements for the award of Degree of Bachelor of Engineering in Department of Electronics and Communication Engineering during the academic year 2023- 2024. Further the matter embodied in the project report has not been submitted previously by anybody for the award of any degree to any university.

Place: Mysuru

Date:

Name	USN	Signature with date
1. Dhanya R Rao	4NN21EC012	
2. Varsha Bhat K	4NN21EC050	
3. Sanjan G	4NN22EC411	
4. Yashwanth M	4NN22EC413	

ACKNOWLEDGEMENT

We sincerely owe our gratitude to all the person who helped us and guided us in completing this project.

We are thankful to Prof. **Dr. Rohini Nagapadma**, principal, NIE Institute of Technology, Mysuru for having supported us in our academic endeavors.

We take this opportunity to express our profound gratitude and deep regards to our guide **Dr Manjula A V**, Associate Professor & HOD, Department of Electronics and Communication Engineering, NIE Institute of Technology, Mysuru.

We are obliged to all **Teaching and Technical staff members** of **Department of Electronics and Communication Engineering, NIE Institute of Technology, Mysuru** for the valuable information providing by them in their respective fields. We are grateful for their co- operation during the period of our project.

Lastly, we thank almighty, our parents and friends for their constant encouragement without which this project would not be possible.

Dhanya R Rao	4NN21EC012
Varsha Bhat K	4NN21EC050
Sanjan G	4NN22EC411
Yashwanth M	4NN22EC413

ABSTRACT

The rapid growth of remote surveillance and security applications has fuelled the demand for advanced detection systems. This project introduces a Raspberry Pi-based system capable of both thermal and optical detection, featuring automated targeting and real-time monitoring capabilities. This system incorporates key components like infrared sensors, cameras, missile model, and a Raspberry Pi microcontroller for intelligent control and data processing. A user-friendly interface with a display provides real-time visualization and control of the system.

At the core of the system, the Raspberry Pi microcontroller processes data from the infrared sensors and camera, analyses images for potential threats, and controls the missile model to automatically track and launch amino at the detected targets. Infrared sensors provide valuable data on heat signatures, while the camera captures visual information for further analysis. The CNN enables precise and agile target tracking, ensuring continuous monitoring of detected threats.

The system's intuitive interface utilizes a display to present real-time video feeds, thermal imagery, and tracking information. User interaction is facilitated through intuitive controls for system configuration and target tracking adjustments. This design emphasizes real-time performance, accuracy, and adaptability, addressing critical aspects of security surveillance, including early threat detection, automated response, and continuous monitoring.

This Raspberry Pi-based detection system showcases the potential of embedded systems in advanced surveillance applications. By integrating thermal and optical sensing with automated targeting and real-time monitoring, it enhances security capabilities, enabling proactive threat assessment and efficient resource allocation.

TABLE OF CONTENTS

Chapter6

RESULTS AND CONCLUSION

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Introduction

Nowadays, the Closed-Circuit Television (CCTV) surveillance system is being utilized in order to keep peace and provide security to people. There are several defects in the video surveillance system, such as: picture is indistinct, anomalies cannot be identified automatically, a lot of storage spaces are needed to save the surveillance information, and prices remain relatively high [21]. The described project uses the design and implementation of a low-cost system monitoring based on Raspberry Pi, a single board computer which follows Motion Detection algorithm written in Python as a default programming environment. In addition, the system uses the motion detection algorithm to significantly decrease storage usage and save investment costs. The algorithm for motion detection is being implemented on Raspberry Pi, which enables live streaming camera along with detection of motion and a thermal camera to detect temperature range of human.

In an era of increasing security concerns, the development of advanced surveillance and targeting systems has become paramount. This project focuses on the design and implementation of a Raspberry Pi-based system capable of real-time thermal and optical detection, automated target acquisition, and remote monitoring.

The system leverages the computational power of the Raspberry Pi 5, coupled with a thermal camera and a webcam, to provide a comprehensive solution for object detection and tracking. By integrating a servo motor with a laser module, the system achieves precise target designation. Furthermore, the utilization of Python and OpenCV enables sophisticated image processing algorithms for object identification and tracking within both thermal and visual spectrums.

This project aims to demonstrate a cost-effective and versatile platform for various applications, including security surveillance, real time monitoring, specified detection and targeting based on user requirements, and industrial automation. The system's ability to operate in real-time and provide remote monitoring capabilities enhances its applicability in diverse scenarios.

1.2 Problem Statement

This project aims to address the limitations of traditional surveillance systems by developing a Raspberry Pi-based system that integrates thermal and optical detection with automated targeting for improved object detection, tracking, and engagement in challenging environments.

1.3 Objectives

- To design thermal detection and optical tracking system.
- To integrate thermal & optical systems.
- To implement automated targeting and develop real-time monitoring.

Chapter 2

LITERATURE SURVEY

[1]. The research paper titled "**Online Object Recognition Using CNN-based Algorithm on High-speed Camera Imaging:** Framework for fast and robust high-speed camera object recognition based on population data cleansing and data ensemble" presents a novel framework for real-time object recognition using convolutional neural networks (CNNs) integrated with high-speed camera imaging technology. The study addresses the critical challenge of achieving both speed and robustness in object recognition systems, particularly in scenarios where high-speed imaging is essential for dynamic and complex environments. The research emphasizes the practical applications of the proposed system in areas requiring real-time responses, such as robotics, surveillance, and industrial automation.

[2]. The paper titled "**Raspberry Pi 5: The New Raspberry Pi Family with More Computation Power and AI Integration**" explores the latest advancements in the Raspberry Pi family, focusing on the introduction of the Raspberry Pi 5. This iteration represents a significant leap in computational capabilities and integrates artificial intelligence (AI) features, making it a robust platform for various applications. A central theme of the research is the role of AI integration in transforming the usability and scope of the Raspberry Pi. With its advanced machine learning capabilities and AI frameworks, the Raspberry Pi 5 is capable of handling tasks such as real-time image recognition, natural language processing, and predictive analytics. This opens up new opportunities in fields like robotics, IoT, and embedded systems.

[3]. The research paper titled **"Person Tracking Control of Mobile Robots Using a Lightweight Object Detection and Tracking System"**. Presents a novel approach to enhancing mobile robots' ability to track individuals in real-time. The authors introduce a lightweight system that integrates deep learning-based object detection with efficient tracking algorithms, specifically designed for mobile robots operating in dynamic environments. The proposed system employs a streamlined neural network architecture for pedestrian detection, ensuring high accuracy while maintaining computational efficiency. This design choice enables the system to run on mobile robots with limited processing

power without compromising performance.

[4]. The research paper titled "**Detection Method to Continue Tracking of Automatic Human Tracking System**" introduces the Ripple Detection Method, a novel approach designed to enhance the robustness of automatic human tracking systems. The Ripple Detection Method addresses challenges such as occlusions and sudden movements by employing a ripple-like propagation mechanism to detect and track individuals across multiple camera nodes. This method ensures continuous tracking by dynamically adjusting to changes in the target's position and the environment. Experimental results demonstrate that the Ripple Detection Method significantly improves tracking accuracy and reliability in complex scenarios, making it a valuable contribution to the field of automated surveillance and human tracking systems.

[5]. The research paper titled "**Face Recognition Using IPCA-ICA Algorithm**" introduces a novel approach to face recognition by combining Incremental Principal Component Analysis (IPCA) with Independent Component Analysis (ICA). This method, referred to as IPCA-ICA, is designed to efficiently compute the principal components of image sequences incrementally, eliminating the need to estimate the covariance matrix, and simultaneously transforming these components into independent directions that maximize non-Gaussian. The IPCA-ICA algorithm operates in real-time by sequentially applying two major techniques: Principal Component Analysis (PCA) and Independent Component Analysis (ICA). Initially, PCA reduces the dimensionality of the image data by identifying the principal components, effectively capturing the most significant features. Subsequently, ICA processes these principal components to extract statistically independent sources, enhancing the representation of facial features critical for recognition tasks.

[6]. The research paper titled "**Multiple Objects Recognition for Industrial Robot Applications**" introduces a vision-based system designed to enhance industrial robots' ability to recognize and manipulate multiple objects, including those with complex shapes and reflective surfaces. The proposed system employs a combination of image processing techniques to achieve accurate object recognition. It utilizes edge detection methods to identify object boundaries, facilitating effective segmentation of objects from the background. Feature extraction algorithms are then applied to analyze the segmented objects, capturing essential characteristics such as shape and texture. This information is

crucial for distinguishing between different objects within the robot's operational environment.

[7]. The research paper titled "**Human Detection using Infrared Thermal Imaging System**" explores the application of infrared thermal imaging for human detection. The authors employed image processing techniques to enhance thermal images and reduce unwanted noise. They utilized the Haar-like feature method to define a square region centered on the tip of the face, enabling effective face detection and tracking. The system continuously adjusts its pan and tilt to keep the detected face centered on the display screen, achieving accurate real-time tracking. In this study, the authors developed a system that utilizes infrared thermal imaging to detect human faces. They applied image processing techniques to enhance the recorded thermal images and remove unwanted noise.

[8]. The research paper titled "**Thermal Imaging Detection System: A Case Study for Indoor Environments**" presents a system designed for detecting and localizing individuals within indoor spaces using thermal imaging technology. This system is particularly beneficial for applications such as people-flow analysis in museums, smart home management, and monitoring hazardous areas like railway platforms. The authors developed a method utilizing FLIR Lepton 3.5 thermal cameras in conjunction with Raspberry Pi 3B+ computers. They created a control and capture library for the Lepton 3.5 and introduced a novel person-detection technique employing the YOLO (You Only Look Once) real-time object detector, which is based on deep neural networks. The thermal unit was designed with automated configuration using Ansible and encapsulated in a custom 3D-printed enclosure.

Chapter 3

HARDWARE REQUIREMENTS

This chapter provides a detailed overview of the hardware components that constitute the core of the Raspberry Pi-based thermal and optical detection system. A careful selection of components is crucial for achieving optimal system performance, reliability, and cost-effectiveness. The following sections detail the specifications and functionalities of each key hardware component, along with their roles within the overall system architecture.

The hardware components used are as follows:
3.1 Thermal Sensor AMG8833
3.2 Web camera
3.3 Laser Module
3.4 Servo Motor
3.5 Motor Drive
3.6 Raspberry Pi
3.7 Missile Module Prototype

3.1 Thermal Senor (AMG8833)

The AMG8833 thermal imaging camera sensor is an 8x8 infrared thermal sensor array. It has a temperature measurement range of 0°C to 80°C (32°F to 176°F). When connected to your microcontroller (or Raspberry Pi), it will return a set of 64 separate infrared temperature readings via I2C. It is compact and simple, and easy to integrate. The sensor only supports I2C and has a configurable interrupt pin that can be triggered when any single pixel is above or below the threshold you set

The Infrared Array Sensor is a thermopile-typed infrared sensor which detects quantity of infrared ray by measuring absolute temperatures in two-dimensional area of 8 x 8 (64 pixels).

Grid-EYE is able to provide thermal images by measuring actual temperature and temperature gradients. Grid-EYE enables detection of multiple persons, identification of positions and direction of movement, almost independent of ambient light conditions without disturbing privacy as with conventional cameras.

Features:

- Temperature detection of two-dimensional area: 8 × 8 (64 pixels).
- Digital output (capability of temperature value output)

- Compact SMD package (adaptively to reflow mounting)
- RoHS compliant.
- Temperature detection on Two-Dimensional Area with8 x 8 pixels(64).
- It can be use on Arduino or compatible (sensors communicate via I2C) or on Python's Raspberry Pi. On the Pi, with the image processing and the SciPy python library, you can insert 8x8 meshes.
- It can detect humans from distances as far as 7 meters (23) feet.
- The sensor only supports I2C, and has a configurable interrupt pin that can fire when any individual pixel goes above or below a threshold that is set.

Specifications:

- Operating Voltage: 1.6V - 3.6V
- Temperature Range: 0°C to 80°C (32°F to 176°F)
- The accuracy is + - 2.5 ° C (4.5 ° F).
- Maximum frame rate of 10 Hz, it is ideal for creating your own body detector or mini camera.
- Dimensions: 25.8mm x 25.5mm x 6.0mm
- Weight: 10g

Applications:

- Energy savings in offices (air-conditioning and lighting controls)
- Digital signage
- Automatic door and elevators
- High performance home appliances (Microwave ovens and air conditioners.

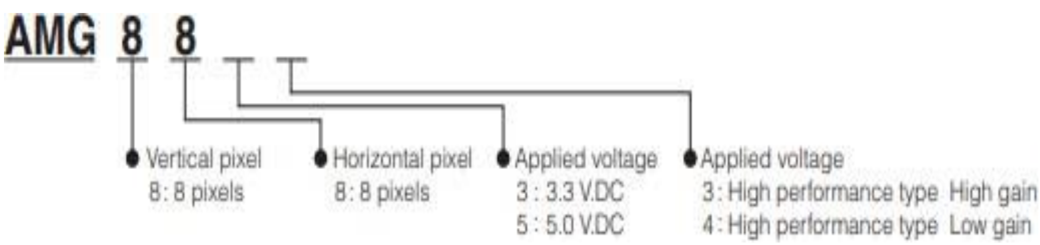


Fig 3.1.1 AMG Series Specification

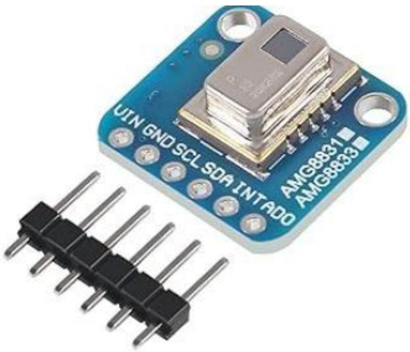


Fig 3.1.2 AMG8833

The thermal sensor plays a crucial role in this project by providing the system with the ability to detect objects based on their temperature differences. In this project, the range is programmed between 26 to 32 degrees. The color values that is assigned in the code is 1024. This is particularly advantageous in scenarios where visual detection is limited, such as low-light conditions, heavy fog, or when targets are camouflaged. By capturing infrared radiation emitted by objects, the thermal sensor generates thermal images that highlight temperature variations, allowing the system to identify and track targets that might be invisible to a standard camera. This enhances the system's overall detection capabilities and its effectiveness in challenging environments.

The internal circuit of AMG8833 works by detecting infrared radiation, amplifying the signals, converting them to digital values, and communicating the data to a microcontroller for further processing and visualization. The heart of the sensor is the Infrared Array Sensor which consists of 64 thermopile elements. Each thermopile generates a small voltage when exposed to infrared radiation emitted by objects in its field of view. The generated voltages from the thermopiles are fed into the Selector. This unit selects the output from one thermopile at a time, allowing the system to read each element sequentially. The selected signal then passes through a Gain Amplifier which amplifies the weak signals from the thermopiles to a level suitable for the subsequent stages. The amplified signals are then fed into an ADC. This circuit converts the analog signals into digital values, which can be easily processed by the microcontroller. The Control Unit manages the overall operation of the sensor. It controls the selector to read each thermopile, configures the gain amplifier, and initiates the ADC conversion. The I2C interface allows SCL and SDA communication with the microcontroller. The AD_SELECT pin is used to select the ADC channel for reading. The INT pin generates an interrupt signal when the ADC conversion is complete.

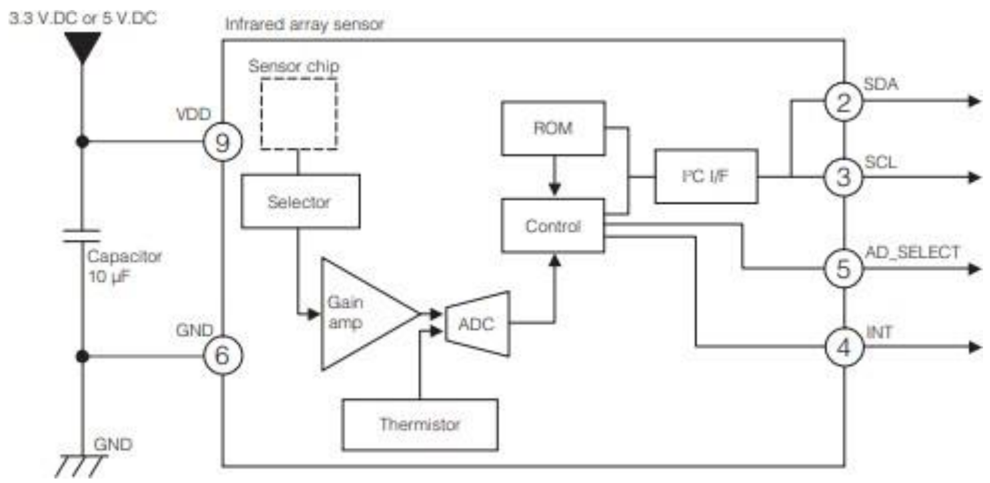


Fig 3.1.3 Internal Circuit Diagram of AGM8833

The thermal image represents the temperature distribution of the objects in the sensor's field of view. The colours of the image correspond to different temperature ranges. Cooler objects are typically represented by blue or purple. Warmer objects are shown in yellow, orange, or red. Intermediate temperatures are often depicted in green. We can analyse the thermal image to determine the temperature of the object in the scene, (here a person).

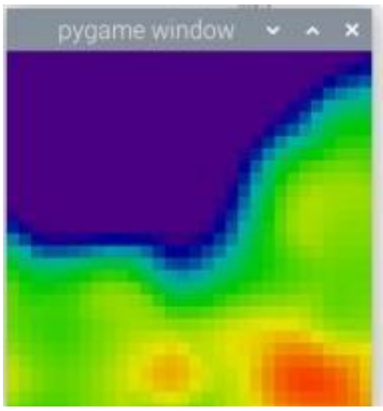


Fig 3.1.4 Thermal image (output of AGM8833)

3.2 Web Camera

Web cam has an important role of detecting and identifying the object within its view range. This project utilizes a Zebronics Crystal Pro Webcam as the primary visual input device. This USB 2.0 webcam offers a 640x480 pixel resolution at 30 frames per second, providing real-time video capture for object detection and tracking. Equipped with a 3P high-quality lens and a built-in microphone, the webcam serves as a cost-effective solution for basic visual data acquisition. While its fixed focus and VGA resolution limit its capabilities in demanding scenarios, it provides a suitable platform for initial prototyping and testing of the object detection and tracking algorithms within the project's scope.With the help of programming, the camera can be operated to detect only a certain object as per the requirements. In this project it has been programmed to detect person, cell-phone, and pen.

Features:

- Video Resolution: 640 x 480 pixels (VGA)
- Frame Rate: 30 frames per second (fps)
- Lens: 3P high-quality lens
- Image Sensor: CMOS
- Interface: USB 2.0
- Built-in Microphone: Yes
- Night Vision: No
- Focus: Fixed focus
- Cable Length: 1.2 meters

The Zebronics Crystal Pro webcam is suitable for basic object detection and tracking tasks. Its VGA resolution might be sufficient for initial testing and prototyping. However, for more advanced applications or higher accuracy requirements, a webcam with higher resolution (720p or 1080p) and features like autofocus would be preferable. It has many pros like: affordable price, plug and play simplicity.

Fig 3.2.1 Web Cam

The basic working of the webcam is given below:

CMOS (Complementary Metal-Oxide-Semiconductor) Sensor is the core of the webcam. Which is common in modern webcams due to its lower power consumption and cost compared to CCD sensors. It captures light and converts it into an electrical signal. The Analog Front-End (AEF) amplifies the weak signals from the sensor, filters out noise, and prepares the signal for the next stage. The Digital Media Processor is the brain of the webcam. It processes the digital image data from the sensor and performs functions: image enhancement (adjusting brightness, contrast, color balance, and sharpness); focus adjustment; compression (reducing bandwidth requirements by compressing data). The SDRAM acts as a temporary storage for the processed image data, and allows the webcam to capture and store a sequence of frames for video output. The Crystal Pro Webcam has a built-in microphone. The audio signals from the microphone are amplified and then converted into digital format by an Analog-to-Digital Converter (ADC). The power management unit regulates the power supply to the various components of the webcam, ensuring stable operation. The USB interface allows the webcam to connect to the computer and transmit the captured video and audio data. The Crystal Pro Webcam uses a USB 2.0 interface. The USB port is the physical connector on the webcam that plugs into the computer's USB port.

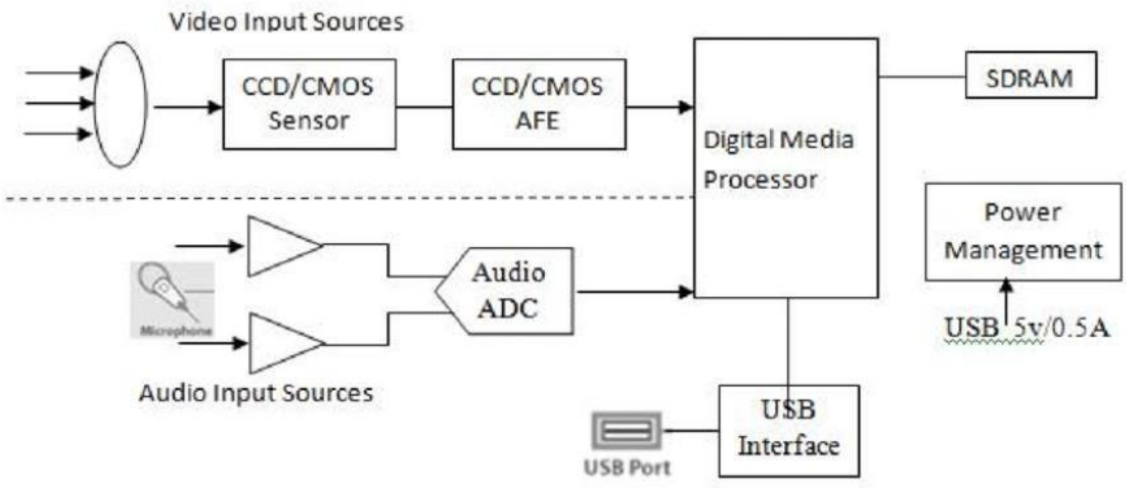


Fig 3.2.2 Block Diagram of Web Cam

The project leverages the camera input as a primary source of visual information for object detection and tracking. The captured video stream from the webcam is continuously processed by the system. OpenCV library plays a crucial role in analyzing the video frames, extracting relevant features, and identifying objects of interest. Image processing techniques, such as color filtering, edge detection, and object recognition algorithms (like Haar Cascades or HOG), are applied to the video stream to detect and locate objects within the field of view. The system then tracks the movement of these detected objects by analyzing their positions in consecutive frames.

The system can detect and identify objects of interest within the camera's field of view in real time. It tracks the movement of detected objects, providing continuous updates on their positions and trajectories. The system displays the captured video feed along with visual cues, such as bounding boxes, to indicate the location and movement of detected objects. This output is integrated with the automated targeting system.

The output window of the webcam is shown in the figure below. The labels include various object categories such as "person" and "cell phone". The label is specified as per the requirements applied in the code.

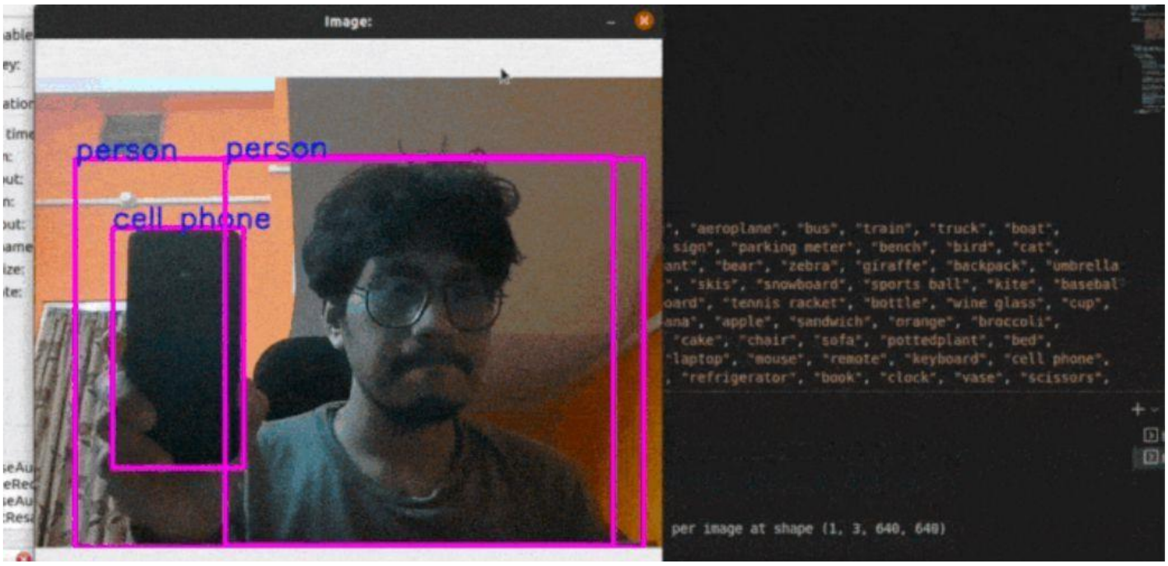


Fig 3.2.3 Output Window of Web Cam

3.3 Laser Module (KY-008)

The project incorporates a laser module for precise target designation. The laser module emits a focused beam of light, allowing for accurate target acquisition and marking. By integrating the laser module with the servo motor and web cam, the system achieves automated target engagement. This enables the system to not only detect and track objects but also precisely direct the laser beam towards the target, simulating a real-world targeting scenario. The laser module is a crucial component in demonstrating the system's capabilities for automated targeting and precision engagement.

The KY-008 Laser Transmitter module is used in this project. It emits a dot shaped, red laser beam and is compatible with Arduino, Raspberry PI, ESP32 and other popular microcontrollers. The main component of the KY-008 Laser sensor module is represented by a laser diode that emits a low power red laser beam. This module consists of a 650nm red laser diode head, a resistor and 3 male header pins.

Operating Voltage	5V
Output Power	5mW
Wavelenght	650nm
Operating Current	< 40mA
Working Temperature	-10°C ~ 40°C [14°F to 104°F]
Board Dimensions	18.5mm x 15mm [0.728in x 0.591in]

Table 3.3.1 KY-008 Specifications

The KY-008 laser module board has three pins. Starting from the pin marked with S, the pins of the laser module are:

- Pin 1: Signal pin to activate and deactivate the laser.
- Pin 2: 5V
- Pin 3: Ground

Fig 3.3.1 KY-008 Pinout.

The module consists of two resistors R1 and R2, where R1 is 10K ohms and R2 is generally 91K ohms. These resistors are used to protect the laser module from burning out due to excessive current. The input digital signal is given through the sig pin, the v+ and GND refer to the power supply and ground connections respectively.

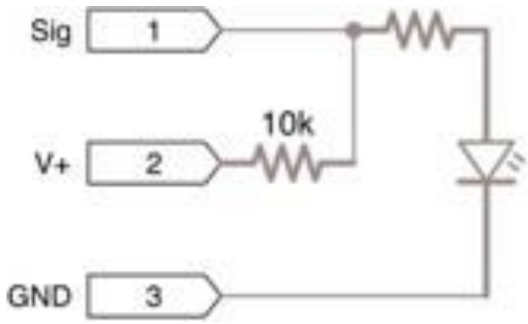


Fig 3.3.2 Internal Circuit Diagram

The KY-008 laser module, a small and compact component, operates by generating a focused beam of coherent light using a semiconductor laser diode. When powered, the module emits a narrow, intense beam of red light. The laser diode within the module is driven by a constant current source, ensuring consistent and stable laser output. The module typically includes a simple lens to collimate the laser beam, improving its directionality and focus.

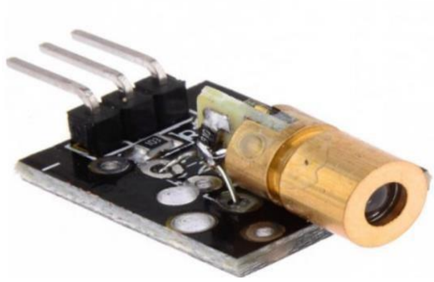


Fig 3.3.3 KY-008 Laser Module

3.4 Servo Motor (SG90)

The SG90 servo motor is a compact and lightweight servo motor commonly used in robotics and hobby projects. It is a type of motor that is powered by a DC source, either from external supply or by a controller. It offers precise rotational control within a specific range of motion, typically 180 degrees. The servo motor receives control signals, typically in the form of Pulse Width Modulation (PWM) signals, which determine the desired angle of rotation. By controlling the PWM signal, the system can accurately position the servo motor to direct the laser module towards the detected target, enabling precise target engagement and demonstrating the system's automated targeting capabilities.

Pulse width modulation or PWM is a commonly used control technique that generates analog signals from digital devices such as microcontrollers. The signal thus produced will have a train of pulses, and these pulses will be in the form of square waves. Thus, at any given time, the wave will either be high or low. Pulse width modulation reduces the average power delivered by an electrical signal by converting the signal into discrete parts. In the PWM technique, the signal’s energy is distributed through a series of pulses rather than a continuously varying (analogue) signal.

Fig 3.4.1 Servo Motor SG90

A servo motor contains a simple setup involving small DC motor, potentiometer, and a control unit. Gears attach the motor to the control wheel. As the motor rotates, the potentiometer’s resistance changes, so the control circuit can precisely regulate the movement and direction of motion. When the motor shaft is at the desired position, the power supplied to the motor is stopped. If not, the motor is turned in the appropriate direction. The desired position is sent via electrical pulses through the signal wire. The motor's speed is proportional to the difference between actual and desired positions. So, if the motor is near the desired position, it will turn slowly. Otherwise, it will turn fast. This is called proportional control. This means the motor will only run as hard as necessary to accomplish the task at hand.

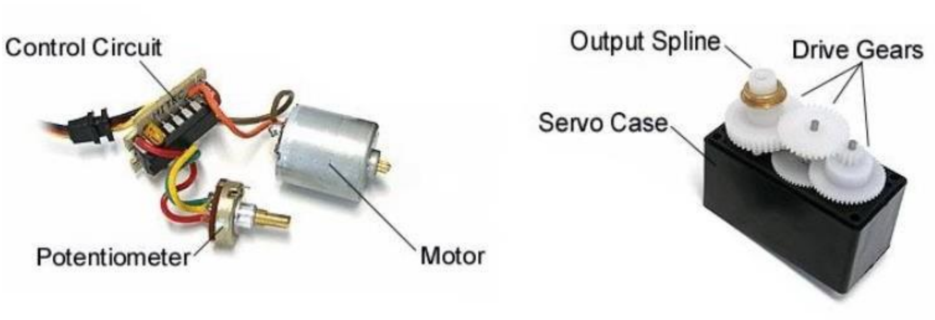


Fig 3.4.2 Servo Motor Control

The PWM sent to the motor determines the position of the shaft, and based on the duration of the pulse sent via the control wire, the rotor will turn to the desired position. The servo motor expects to see a pulse every 20 milliseconds (ms), and the length of the pulse determines how far the motor turns. The PWM signal is a rectangular wave with a fixed period of 20ms (50Hz). The servo's position is determined by the duty cycle of this PWM signal, which is the ratio of the pulse width to the period. A duty cycle between 1-2ms typically corresponds to the minimum position, while 1.5-2ms corresponds to the maximum position.

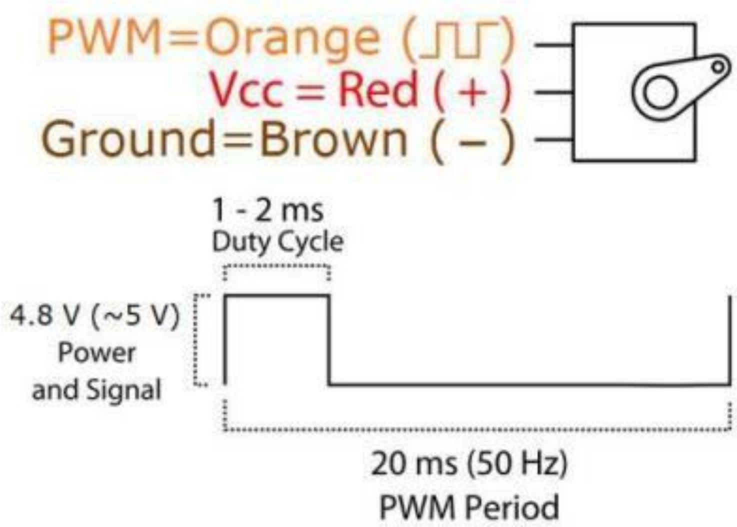


Fig 3.4.3 SG90 Servo Motor Control Signal

The servo motor has three wires: Vcc (red), GND (brown) and PWM Signal (orange). The features of SG90 are as follows:

- Operating Voltage is +5V.
- Torque: 2.5kg/cm
- Operating speed is 0.1s/60 °
- Rotation: 0 ° to 180 °
- Gear type: Plastic
- Weight of the motor: 9gm

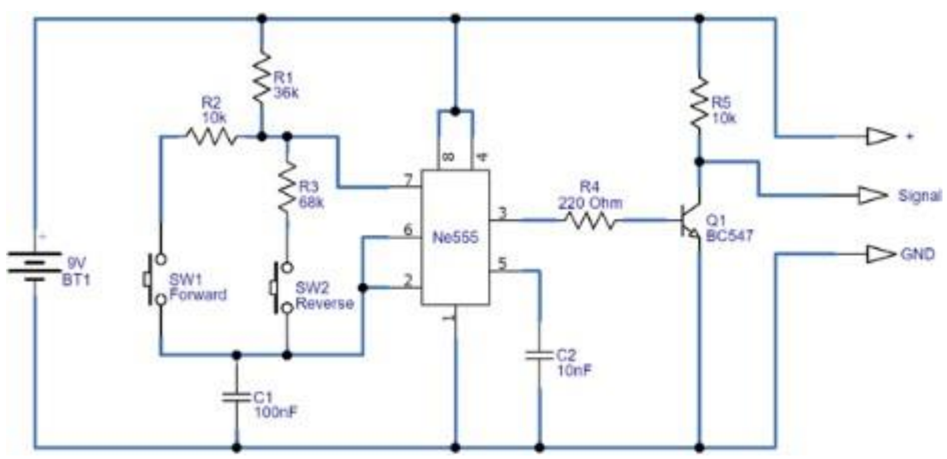


Fig 3.4.4 Internal Circuit of SG90

NAME	WIRE COLOR	DESCRIPTION
PWM	Yellow	Input control signal for the motor
V+	Red	Positive supply terminal
Ground	Brown	Ground terminal

Table 3.4.1 Servo Motor SG90 Pin Details

Applications of the motor include: Used as actuators in many robots like Biped Robot, Hexapod, robotic arm etc. Commonly used for steering system in RC toys, Robots where position control is required without feedback, less weight hence used in multi DOF robots like humanoid robots.

3.5 Motor Drive (Arduino UNO R3)

Motor drives are electronic circuits that control the speed, torque, and position of electric motors. They play a crucial role in a wide range of applications, from industrial automation and robotics to electric vehicles and home appliances. Motor drives convert fixed AC or DC power into variable frequency and amplitude power, allowing for precise control over the motor's performance. This controllability enables energy efficiency, improved performance, and enhanced flexibility in various applications.

The Arduino UNO, a popular microcontroller board, can be effectively utilized as a motor driver for controlling the speed and direction of DC motors or stepper motors. By leveraging the Arduino's digital and analog output pins, PWM (Pulse Width Modulation) signals can be generated to control the motor's speed. Additionally, by strategically controlling the direction pins, the Arduino can reverse the motor's rotation. This versatile approach allows for precise motor control and enables the implementation of various motion control applications, making the Arduino UNO a valuable tool for robotics and automation projects. This project uses Arduino UNO as the motor drive as it is the most compatible. If other drivers are used, it causes the motor to jitter and the working of the model is compromised.

Arduino UNO is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

Fig 3.5.1 Arduino UNO R3 Microcontroller

Features:

- ATMega328P Processor
- Memory: AVR CPU at up to 16 MHz; 32 kB Flash; 2 kB SRAM; 1 kB EEPROM;
- Security: Power On Reset (POR); Brown Out Detection (BOD);
- Peripherals: 2x 8-bit Timer/Counter with a dedicated period register and compare channels; 1x 16-bit Timer/Counter with a dedicated period register, input capture and compare channels; 1x USART with fractional baud rate generator and start-of-frame detection; 1x controller/peripheral Serial Peripheral Interface (SPI); 1x Dual mode controller/peripheral I2C; 1x Analog Comparator (AC) with a scalable reference input; Watchdog Timer with separate on-chip oscillator; Six PWM channels; Interrupt and wake-up on pin change.
- ATMega16U2 Processor: 8-bit AVR® RISC-based microcontroller
- Memory: 16 kB ISP Flash ,;512B EEPROM; 512B SRAM; debugWIRE interface for on-chip debugging and programming.
- Power: 2.7-5.5 volts.

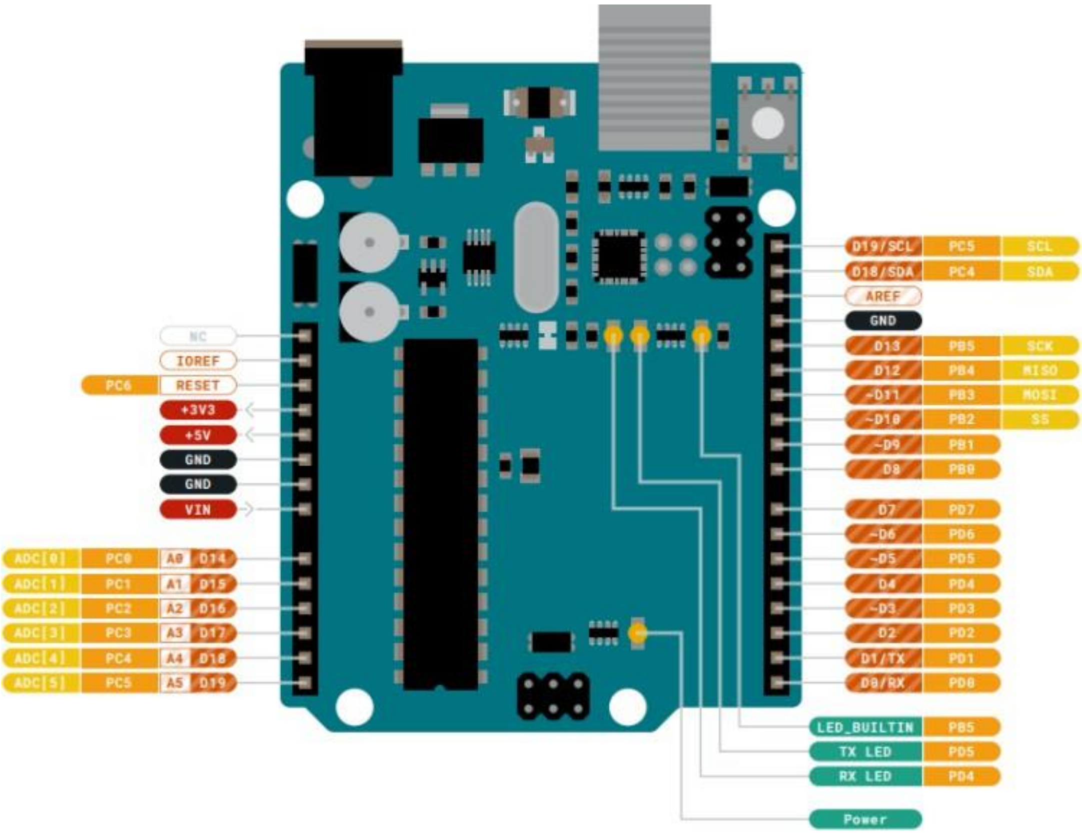


Fig 3.5.2 Pinout of Arduino UNO R3

In this project model, Raspberry Pi acts as the main higher-level logic and as a user interface. The Arduino UNO R3 handles the lower-level control of motors and acts as a motor drive. It responds to the commands from the raspberry pi and acts accordingly. A UART communication between the Raspberry Pi and Arduino is used for the setup.

The Arduino UNO R3 utilizes Pulse Width Modulation (PWM) to control the speed of the motor. PWM involves generating a square wave signal where the "on" time of the signal is varied. By adjusting the "on" time (duty cycle), the average voltage applied to the motor is effectively controlled, thereby regulating its speed. The Arduino UNO R3 has several dedicated PWM pins that can generate these signals. To control the direction of the motor, an H-bridge circuit is typically used. This circuit consists of four transistors arranged in a "H" shape. By switching the transistors on and off in the correct sequence, the direction of current flow through the motor is reversed, causing it to rotate in the opposite direction. The motor is connected to the H-bridge circuit, which is then interfaced with the Arduino UNO. The Arduino controls the transistors in the H-bridge through its digital output pins, determining the direction of current flow and, consequently, the motor's rotation.

3.6 Raspberry Pi 5

The Raspberry Pi 5 serves as the central processing unit of this project, providing the computational power necessary for real-time image processing, motor control, and data analysis. Its 64-bit quad-core processor delivers the performance required for running complex algorithms for object detection, tracking, and targeting in real-time. The 8GB LPDDR5 RAM allows for efficient handling of high-resolution images and video streams captured by the thermal and optical cameras. Furthermore, the Raspberry Pi 5's enhanced connectivity options facilitate seamless communication with other devices and enable remote monitoring and control of the system.

Fig 3.6.1 Raspberry Pi 5

Specification:

Processor:

- Broadcom BCM2712 2.4GHz quad-core 64-bit Arm Cortex-A76 CPU, with Cryptographic Extension, 512KB per-core L2 caches, and a 2MB shared L3 cache.

Features:

- VideoCore VII GPU, supporting OpenGL ES 3.1, Vulkan 1.2 • Dual 4Kp60 HDMI® display output with HDR support

- 4Kp60 HEVC decoder
- LPDDR4X-4267 SDRAM (4GB and 8GB SKUs available at launch)
- Dual-band 802.11ac Wi-Fi®
- Bluetooth 5.0 / Bluetooth Low Energy (BLE)
- microSD card slot, with support for high-speed SDR104 mode
- 2 × USB 3.0 ports, supporting simultaneous 5Gbps operation
- 2 × USB 2.0 ports
- Gigabit Ethernet, with PoE+ support (requires separate PoE+ HAT)
- 2 × 4-lane MIPI camera/display transceivers
- PCIe 2.0 x1 interface for fast peripherals (requires separate M.2 HAT or other adapter)
- 5V/5A DC power via USB-C, with Power Delivery support • Raspberry Pi standard 40-pin header
- Real-time clock (RTC), powered from external battery
- Power button

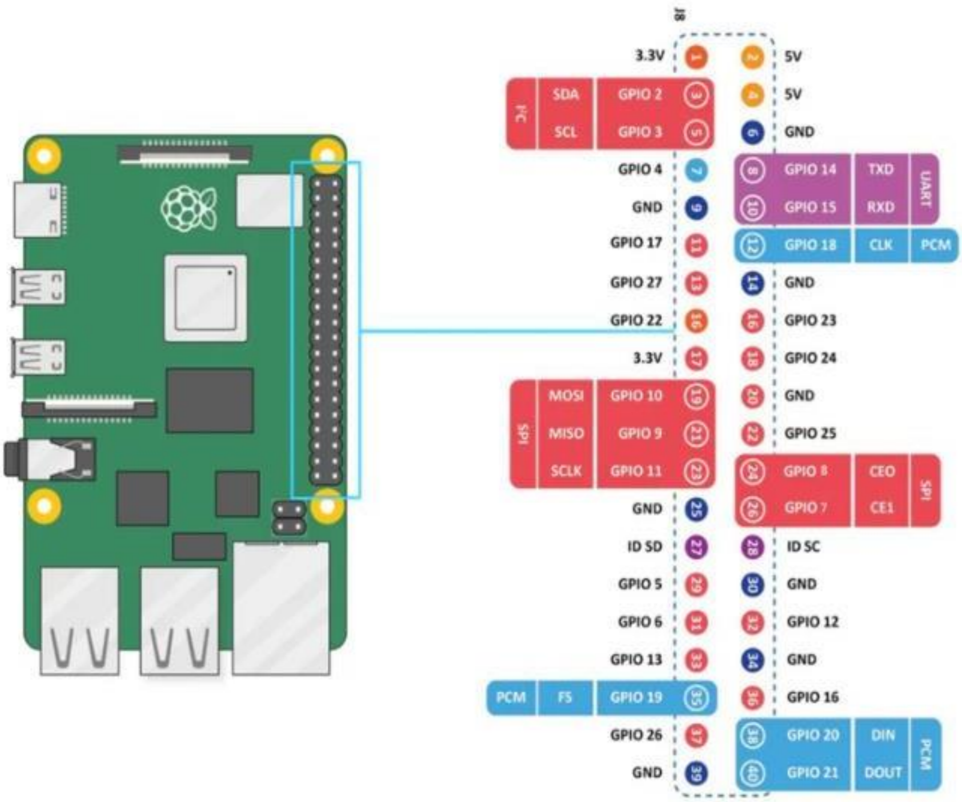


Fig 3.6.2 Raspberry Pi 5 Pinout

In this diagram, we cover the following headers:

- **J8 GPIO Header:** This is the default 40-pin GPIO header.
- **J14 PoE:** Power over Ethernet (PoE) connection.
- **J2 Power Switch:** While not officially confirmed, it may be related to the Real-Time Clock (RTC) battery.
- **J7 Composite Video:** Composite video output (no longer on an audio jack).
- **J17 Fan Header:** This header allows for PWM control and provides feedback for the fan's speed.

The RP1 GPIO bank (IO_BANK0) supports the following functions:

- 5 UART interfaces
- 6 SPI interfaces
- 4 I2C interfaces
- 2 I2S interfaces (including Clock Producer and Clock Consumer instances)
- Registered IO (RIO) interface
- 24-bit DPI output
- 4-channel PWM output
- Stereo PWM audio output (AUDIO_OUT)
- General-purpose clock input and output (GPCLK)
- eMMC/SDIO bus with a 4-bit interface
- Interrupt generation from pin level or edge transitions

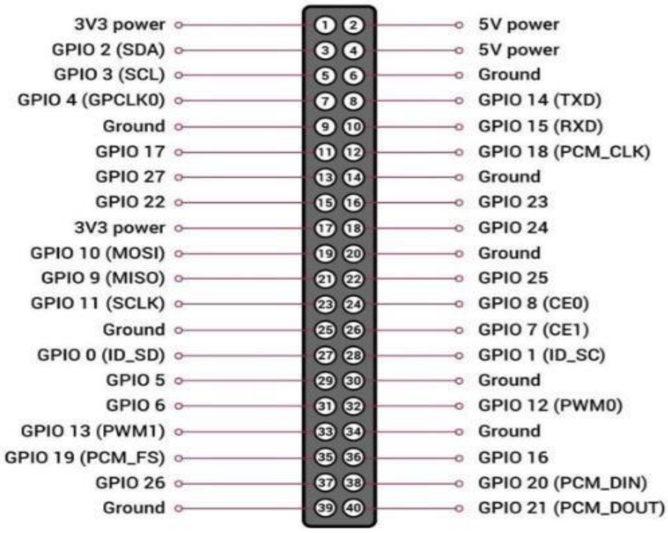


Fig 3.6.3 GPIO Pins

Pin	Physical pin	GPIO pin	Function
1	Pin 1	N/A	3.3V Power
2	Pin 2	N/A	5V Power
3	Pin 3	GPIO 2	I2C SDA (Data Line)
4	Pin 4	N/A	5V Power
5	Pin 5	GPIO 3	I2C SCL (Clock Line)
6	Pin 6	N/A	Ground (GND)
7	Pin 7	GPIO 4	Digital Input/Output pin
8	Pin 8	GPIO 14	UART TXD (Transmit)
9	Pin 9	N/A	Ground (GND)
10	Pin 10	GPIO 15	UART RXD (Receive)
11	Pin 11	GPIO 17	Digital Input/Output pin
12	Pin 12	GPIO 18	PWM (Pulse Width Modulation)
13	Pin 13	GPIO 27	Digital Input/Output pin
14	Pin 14	N/A	Ground (GND)
15	Pin 15	GPIO 22	Digital Input/Output pin
16	Pin 16	GPIO 23	Digital Input/Output pin
17	Pin 17	N/A	3.3V Power
18	Pin 18	GPIO 24	Digital Input/Output pin
19	Pin 19	GPIO 10	SPI MOSI (Master Out Slave In)
20	Pin 20	N/A	Ground (GND)
21	Pin 21	GPIO 9	SPI MISO (Master In Slave Out)
22	Pin 22	GPIO 25	Digital Input/Output pin
23	Pin 23	GPIO 11	SPI SCLK (Clock Line)
24	Pin 24	GPIO 8	SPI CE0 (Chip Enable 0)
25	Pin 25	N/A	Ground (GND)
26	Pin 26	GPIO 7	SPI CE1 (Chip Enable 1)
27	Pin 27	GPIO 0	Digital Input/Output pin
28	Pin 28	GPIO 1	Digital Input/Output pin
29	Pin 29	GPIO 5	Digital Input/Output pin
30	Pin 30	N/A	Ground (GND)
31	Pin 31	GPIO 6	Digital Input/Output pin
32	Pin 32	GPIO 12	PWM (Pulse Width Modulation)
33	Pin 33	GPIO 13	PWM (Pulse Width Modulation)
34	Pin 34	N/A	Ground (GND)
35	Pin 35	GPIO 19	PWM (Pulse Width Modulation)
36	Pin 36	GPIO 16	PWM (Pulse Width Modulation)
37	Pin 37	GPIO 26	Digital Input/Output pin

38	Pin 38	GPIO 20	Digital Input/Output pin
39	Pin 39	N/A	Ground (GND)
40	Pin 40	GPIO 21	Digital Input/Output pin

Table 3.6.1 GPIO Pin Description

The selection of the Raspberry Pi 5 as the central processing unit for this project was driven by its exceptional capabilities. Its powerful 64-bit quad-core processor provides the necessary computational horsepower to handle the intensive image processing tasks involved in real-time object detection and tracking. The 8GB LPDDR5 RAM ensures smooth operation and efficient handling of large image and video data streams generated by the thermal and optical cameras. Moreover, the Raspberry Pi 5's enhanced connectivity options, including faster Ethernet, Wi-Fi 6, and Bluetooth 5.2, facilitate seamless communication with other devices and enable remote monitoring and control of the system. The platform's versatility, coupled with its open-source nature and extensive community support, makes it an ideal choice for this project, allowing for rapid prototyping, development, and future expansion.

3.7 Missile Module Prototype

The project incorporates a handmade missile model prototype to simulate real-world targeting scenarios and evaluate the system's performance. This prototype, designed to mimic the flight characteristics of a missile, provides a tangible platform for testing the system's ability to accurately track and engage targets. By observing the missile model's trajectory and comparing it to the predicted path based on the system's target acquisition and tracking data, the project aims to assess the system's accuracy, precision, and overall effectiveness in a controlled environment.

To make the prototype, a wooden clip was used. A small tooth pick or a 12inch toothpick was used as the amino. The Servo motor on which the proto type is attached runs parallel to the web-cam and thermal cam used. When the target is detected, the toothpick is shot at it. A motor is used to trigger the missile.

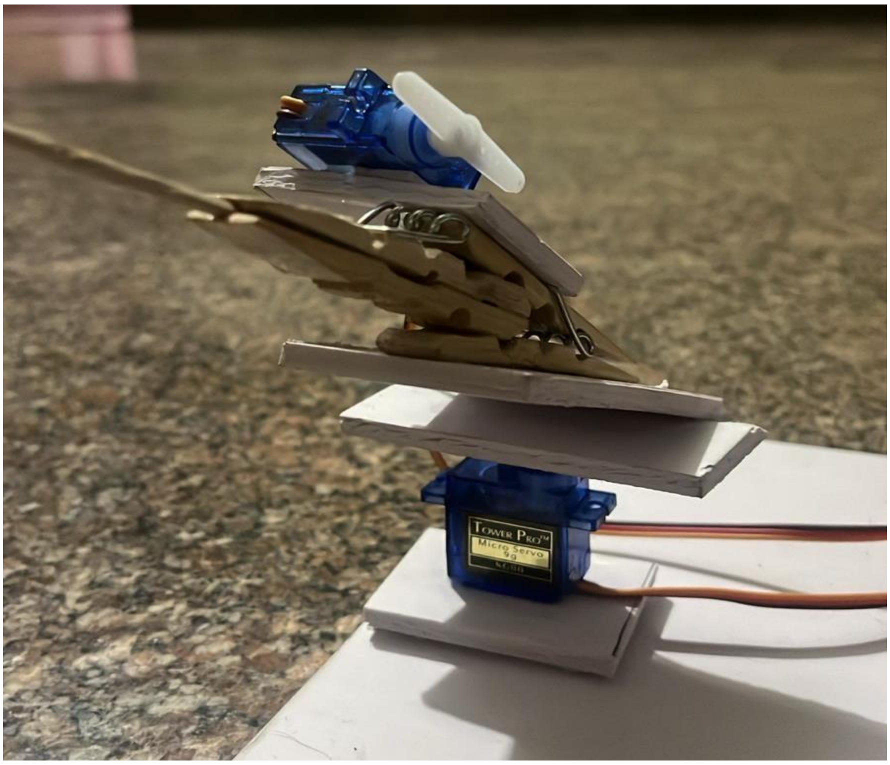


Fig 3.7.1 Missile Prototype

Chapter 4

SOFTWARE IMPLEMENTATION

The successful implementation of this project heavily relies on a robust software framework. Python, with its extensive libraries for image processing, computer vision, and control, forms the backbone of the system. OpenCV, the cornerstone of the software, provides a comprehensive set of functions for image acquisition, manipulation, analysis, and object detection. The integration of NumPy enables efficient numerical computations and array operations, crucial for image processing algorithms. These libraries, combined with the flexibility and ease of use of the Python language, provide the necessary tools for implementing the complex algorithms required for real-time object detection, tracking, and automated targeting.

The software used are:

4.1 OpenCV

4.2 Thonny IDE

4.3 Arduino Software (IDE)

4.4 RealVNC

4.5 Python

4.1 Open CV

OpenCV (Open-Source Computer Vision Library) is a powerful open-source library that provides a comprehensive set of functions for real-time computer vision tasks. It offers a wide range of tools for image and video processing, including image loading and manipulation, feature detection and extraction, object detection and tracking, and motion analysis. In this project, OpenCV plays a crucial role in enabling the system to perform real-time object detection and tracking. It facilitates the analysis of video frames captured by the webcam and thermal camera, allowing the system to identify and locate objects of interest, extract relevant features, and track their movements. OpenCV's extensive functionality and efficiency make it an indispensable tool for implementing the core image processing algorithms required for the successful operation of the system.

4.1.1 Installation of Open CV in Raspberry Pi:

Step 1: Update and upgrade the system

- Open a terminal window on your Raspberry Pi 5.
- Update the package lists:

```
Bash
sudo apt update
```

- Upgrade existing packages:

```
Bash
sudo apt upgrade -y
```

Step 2: Install Build Dependencies

- Install necessary build tools and libraries:

```
Bash
sudo apt install build-essential cmake git unzip
pkg-config
```

- Install libraries required for OpenCV:

```
Bash
sudo apt install libjpeg-dev libpng-dev libtiff-dev
\
libavcodec-dev libavformat-dev libswscale-dev \
libv4l2-dev libxvidcore-dev libx264-dev \
libgtk-3-dev libgtk2.0-dev libcanberra-gtk* \
libgstreamer1.0-dev gstreamer1.0-gtk3 \
libgstreamer-plugins-base1.0-dev \
libgstreamer1.0-libcamera libxvidcore-dev libx264-
dev \
libblas-dev liblapack-dev \
libatlas-base-dev gfortran
```

Step 3: Install NumPy

- Install NumPy, a fundamental library for numerical computing in Python:

```
Bash
sudo apt install python3-numpy
```

Step 4: Instal OpenCV

- Install OpenCV using pip:

```
Bash

sudo pip3 install opencv-python
```

Step 5: Verify Installation

- Create a Python script (e.g., `test_opencv.py`) with the following code:

```
Python
import cv2

img = cv2.imread("path/to/your/image.jpg")  #
Replace with the path to an image

if img is not None:
    print("OpenCV is installed successfully!")
    cv2.imshow("Image", img)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
else:
    print("Error loading image. Check OpenCV
installation.")
```

- Run the script:

```
Bash
python3 test_opencv.py
```

If the image selected for the display is displayed, then OpenCV is installed correctly.

4.2 Thonny IDE

Thonny is a beginner-friendly integrated development environment (IDE) designed specifically for learning and teaching Python programming. Developed with simplicity and clarity in mind, Thonny provides an intuitive interface that minimizes distractions, making it ideal for new programmers who are just starting their coding journey. One of its standout features is the built-in debugger, which allows users to visualize code execution step by step, making it easier to understand complex concepts such as variable scopes, function calls, and loops. Additionally, Thonny offers a simplified management system for Python packages and virtual environments, enabling users to easily install libraries and experiment with isolated environments. Its lightweight design ensures quick installation and startup, while its support for advanced features like autocompletion and code analysis makes it versatile enough for more experienced users. Overall, Thonny serves as a powerful yet accessible tool, bridging the gap between novice programmers and professional coding environments.

4.2.1 Getting Started with Thonny

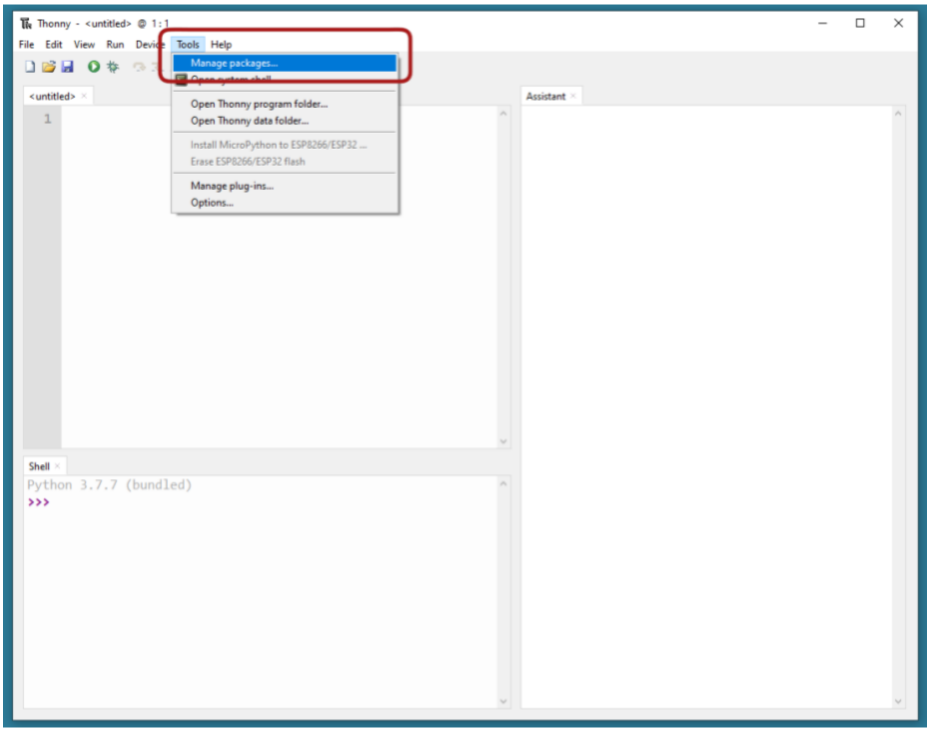
1. Visit https://thonny.org/
2. Use the download links (highlighted in red) to get a Thonny installer for your operating system. To install on Linux, follow the instructions on the website.
3. Locate the downloaded installer (.exe for Windows or .pkg for Mac)
4. Double click to run the installer
5. The default installation options are sufficient.

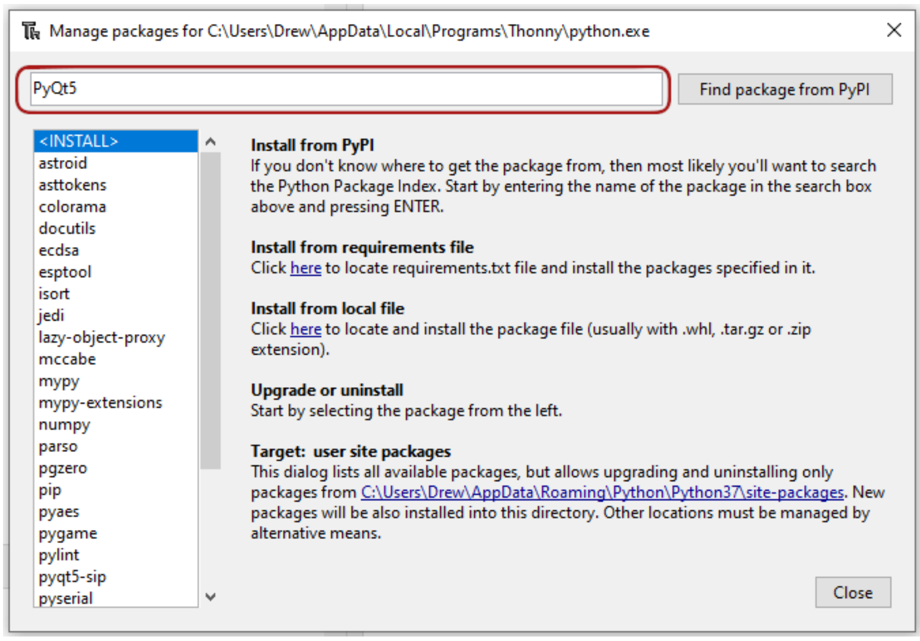
Fig 4.2.1 Thonny Installer Page

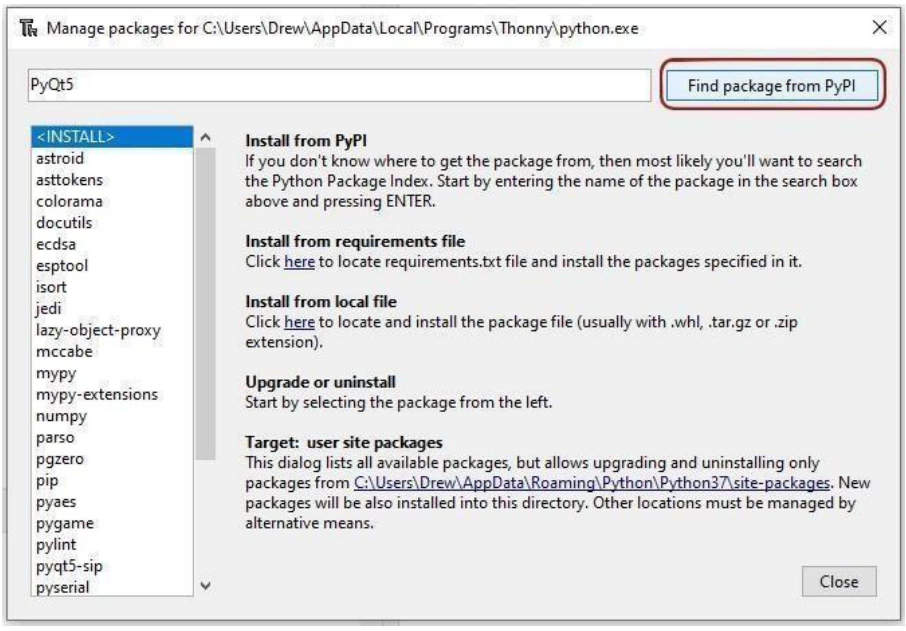
First Time Configuration

1. Run the installed Thonny Application: Click the “Tools” menu and select “Manage packages…” to open the Thonny Pac.
2. In the new Thonny Package Manager Window type the following in the textbox: PyQt5
3. Click the “Find package from PyPI” button
4. When the package is found, click the “Install” button. Installation could take a few minutes.
5. Repeat steps 3-5 for a package named: pgzero
6. Once both packages are installed, close the Thonny Package Manager

7. Thonny is now ready for use.







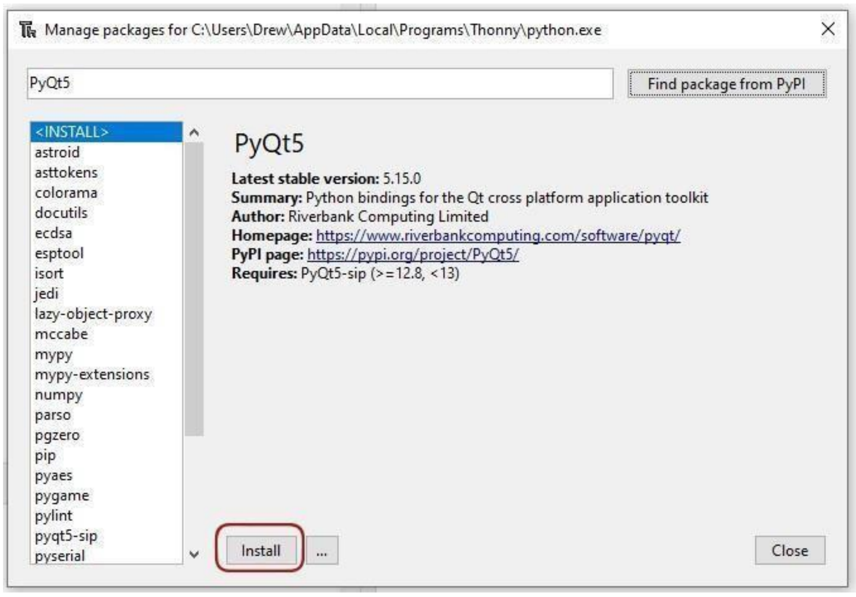
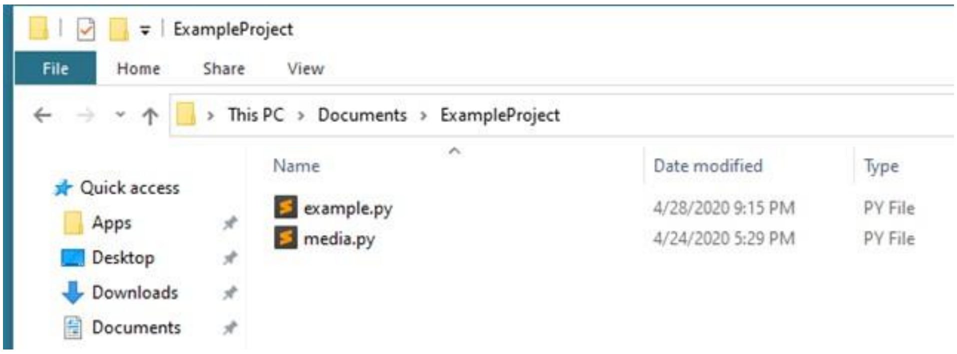


Fig 4.2.2 Thonny Configuration

Using media.py

1. Download media.py here http://csweb.wooster.edu/kbhowmik/cs102/week01/lecture01/media.py
2. Locate the downloaded media.py file



3. Place the media.py file in the same directory as your python code
4. Add the following line to the top of your python code in Thonny:

   import media
5. Each time you want to use a JES function provided by media.py simply write:

   media. [function name] ()

   For example, if you want to use the pickAFile function you would use the follow line of code: media.pickAFile()
6. Don't forget that the function names ARE CASE SENSITIVE (capitalization matters)!

   media.pickAFile() is not the same as media.PickAFile()

4.3 Arduino Software (IDE)

The Arduino Integrated Development Environment (IDE) is a cross-platform application used for writing and uploading code to Arduino and Genuino boards. It's written in Java and based on Processing, a flexible software sketchbook and language for learning electronics and creating digital experiences. The Arduino IDE provides a simple, easy-to-use interface for writing code, compiling it into machine code, and uploading it to the board. It also includes a serial monitor for communicating with the board and debugging code. In this project Arduino UNO is used as the motor driver and this software is necessary to write and control the board as per our requirements.

The Arduino Integrated Development Environment (IDE) is a crucial tool for writing and uploading code to Arduino boards. It features:

- Text Editor: For writing code.
- Message areas feedback and display

- Text Console: Shows output and error messages.
- Toolbar and Menus: Provides quick access to common functions. The Arduino IDE supports Windows, Mac OS X, and Linux. It is written in Java and based on Processing and another open-source software.

Key features of the Arduino IDE include:

- Open Source: Free to use and modify.
- User – Friendly: Suitable for beginners without technical knowledge.
- Multi- Platform: Available for Windows, Mac, and Linux.
- Built-In Functions: Includes commands for debugging, editing, and compiling code.
- Support for Multiple Boards: Compatible with various Arduino modules like Arduino Uno, Mega, Leonardo, and Micro. C and C++ Support: Allows programming in these languages. Hex File Generation: Converts the code into a Hex file for uploading to the microcontroller.

4.3.1 Implementation of Arduino Software (IDE)

To download and implement the latest version open page: https://docs.arduino.cc/software/ide-v1/tutorials/Windows/?_gl=1*jg9vc3*_up*MQ..*_ga*OTEzNzcxMDg3LjE3MzU1Nzc1NTc.*_ga_NEXN8H46L5*MTczNTU3NzU1NC4xLjEuMTczNTU3NzY1OC4wLjAuMzUwNjkzMjg.

There are two options for the installation, the installer (.exe) and the zip packages. When the download is completed, the driver installation is done. The process is shown below.

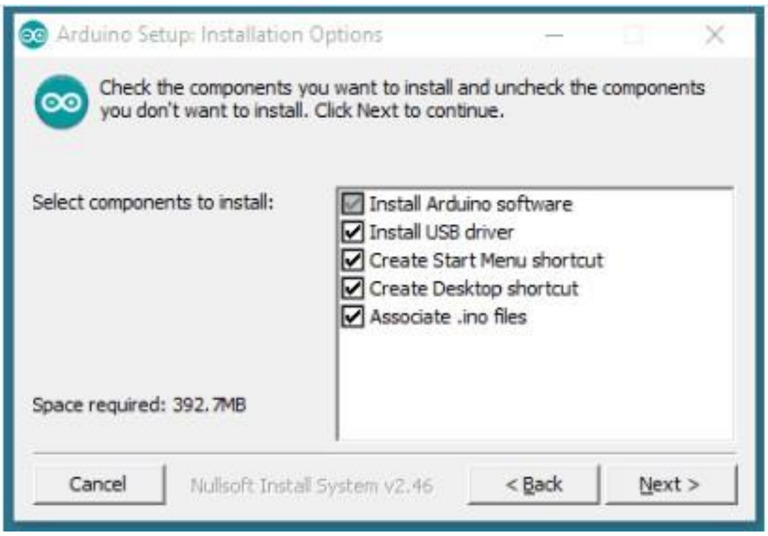


Fig 4.3.1 Step1 Choose the Components to Install

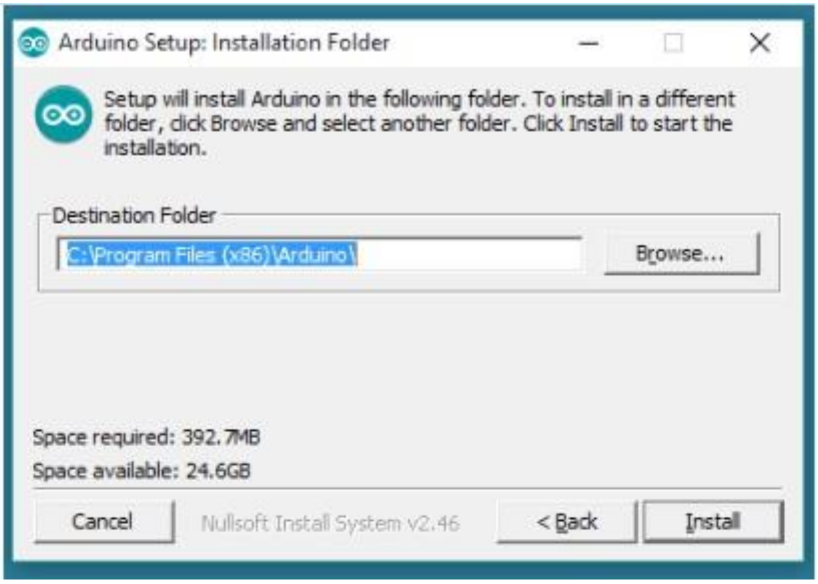


Fig 4.3.2 Step2 Choose the installation directory

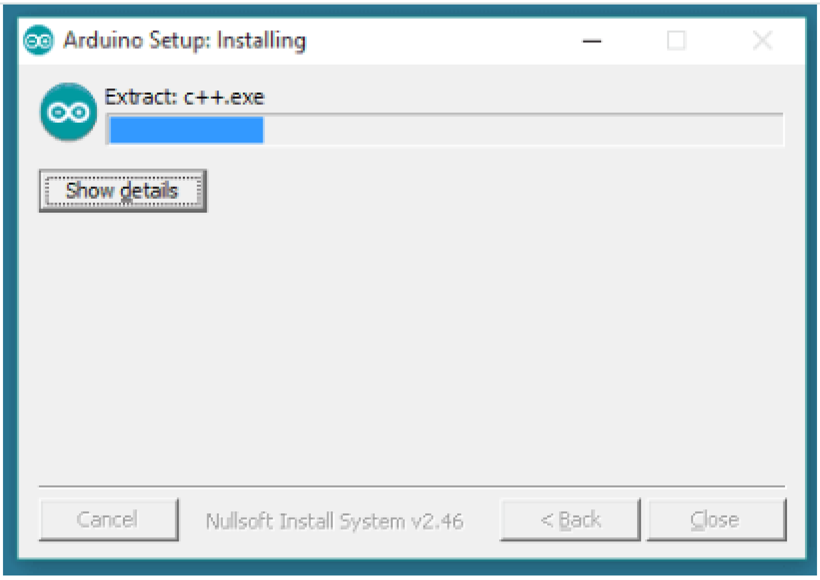


Fig 4.3.3 Installation in progress

4.4 RealVNC Viewer

RealVNC Viewer is a remote desktop software application that allows you to access and control another computer (the "VNC server") from your own device (the "VNC client"). This application allows us to interact with the Raspberry Pi 5. This remote desktop software enables you to access and control the Raspberry Pi's graphical user interface (GUI) from any device with an internet connection. By using RealVNC Viewer, you can conveniently monitor the system's operation, debug code, manage files, and perform other administrative tasks without needing physical access to the Raspberry Pi 5.

On the device we want to control:

- Download the RealVNC Connect setup app.
- Click the **Download for Windows** button.
- Click the **Accept and Install** button.
- Once the download and installation process complete, choose one of:
    - **Start my free trial:** If the user is new to RealVNC Connect and do not already have an account or subscription, to get started with a free 14-day trial
    - **Sign in to continue:** If the user is existing RealVNC Connect user and want to set up this device with your existing account/subscription
- The web browser will open automatically. Follow the on-screen prompts, then click **Open** to return to the **RealVNC Connect Setup app**.

- Review the settings shown, then click **Launch**.

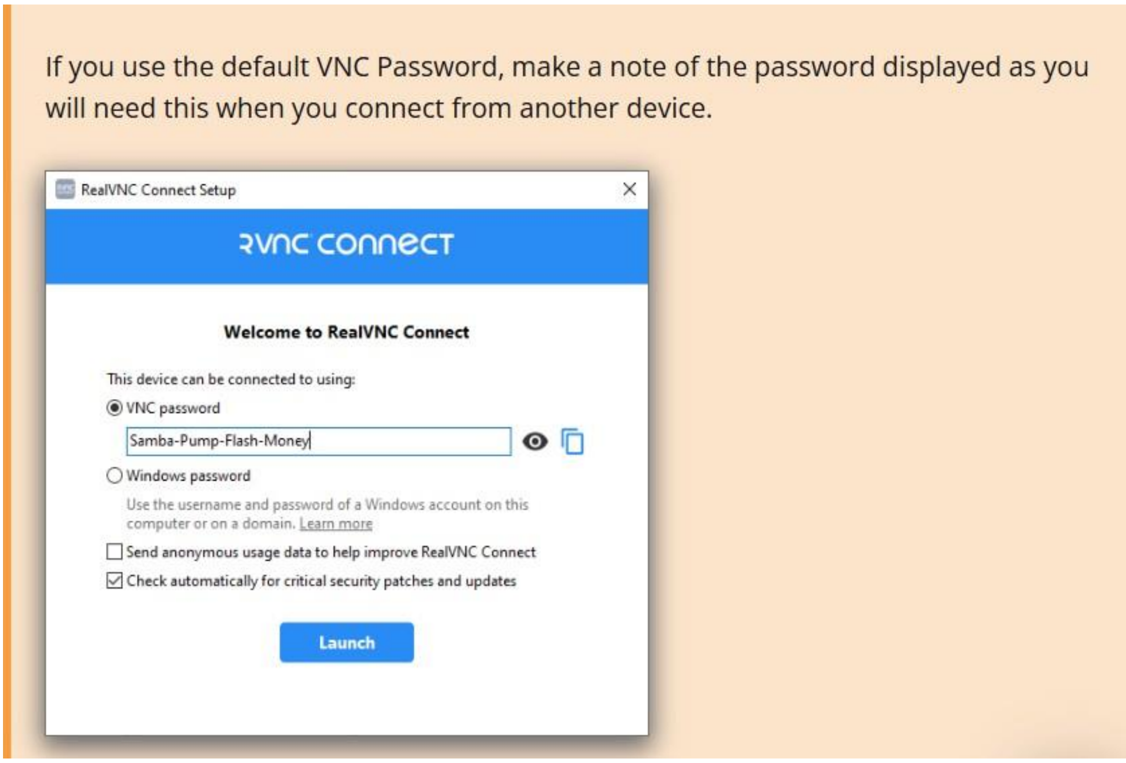


Fig 4.4.1 Sign in Process

On the device we want to control from:

- Download the RealVNC Connect setup app.
- Click the **Download for Windows** button.
- Click the **Accept and Install** button.
- Once the download and installation process complete, choose one of:
  - **Start my free trial:** If the user is new to RealVNC Connect and do not already have an account or subscription, to get started with a free 14-day trial
  - **Sign in to continue:** If the user is an existing RealVNC Connect user and want to set up this device with your existing account/subscription
- Your web browser will open automatically. Follow the on-screen prompts, then click **Open** to return to the **RealVNC Connect Setup app**.
- Review the settings shown, then click **Launch**.
- You will see your remote computer(s) appear automatically in your team:

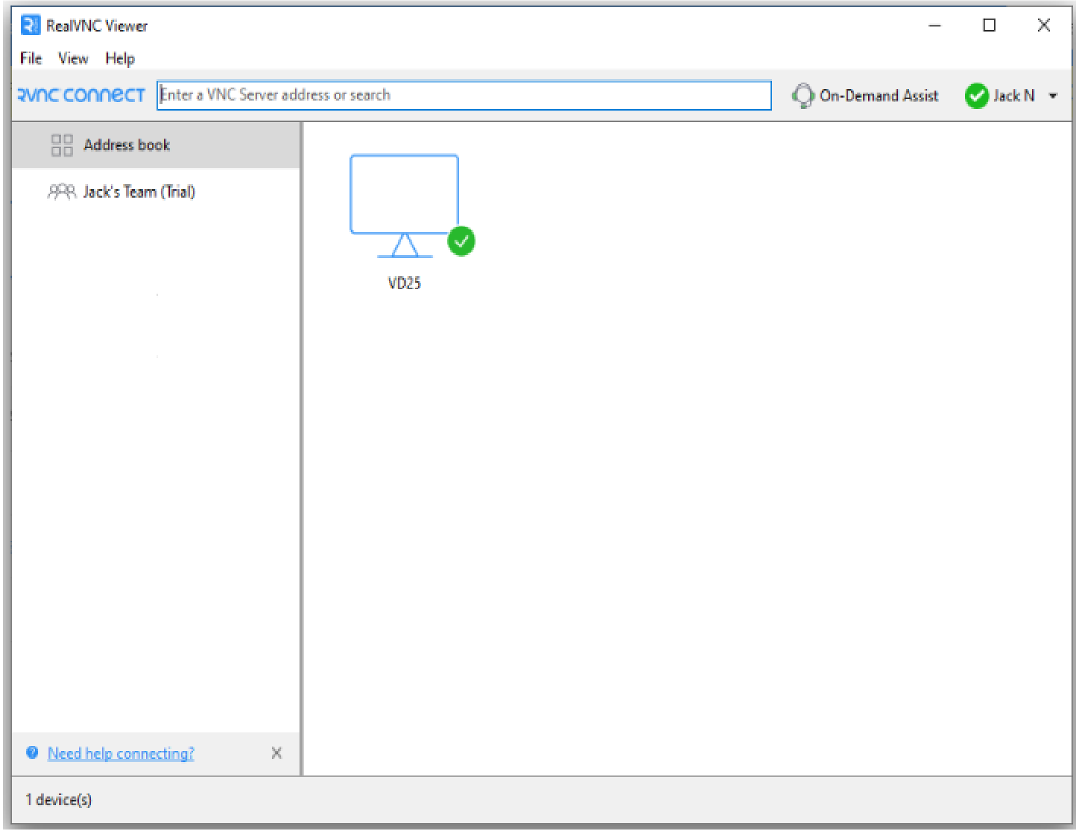


Fig 4.4.2 Connecting Device

Connecting for the first time:

In RealVNC Viewer, double click or tap on the remote computer that you want to connect to. As part of the first connection to a RealVNC Server, RealVNC Viewer will show an Identity Check screen that you can verify wether connected to the right computer.

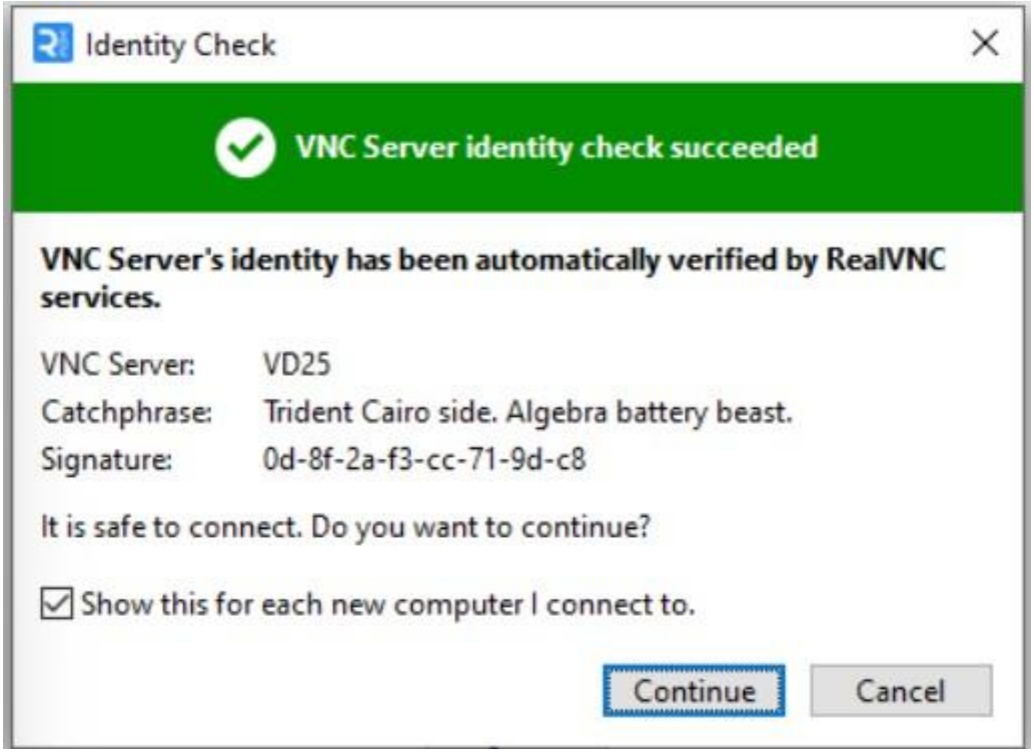
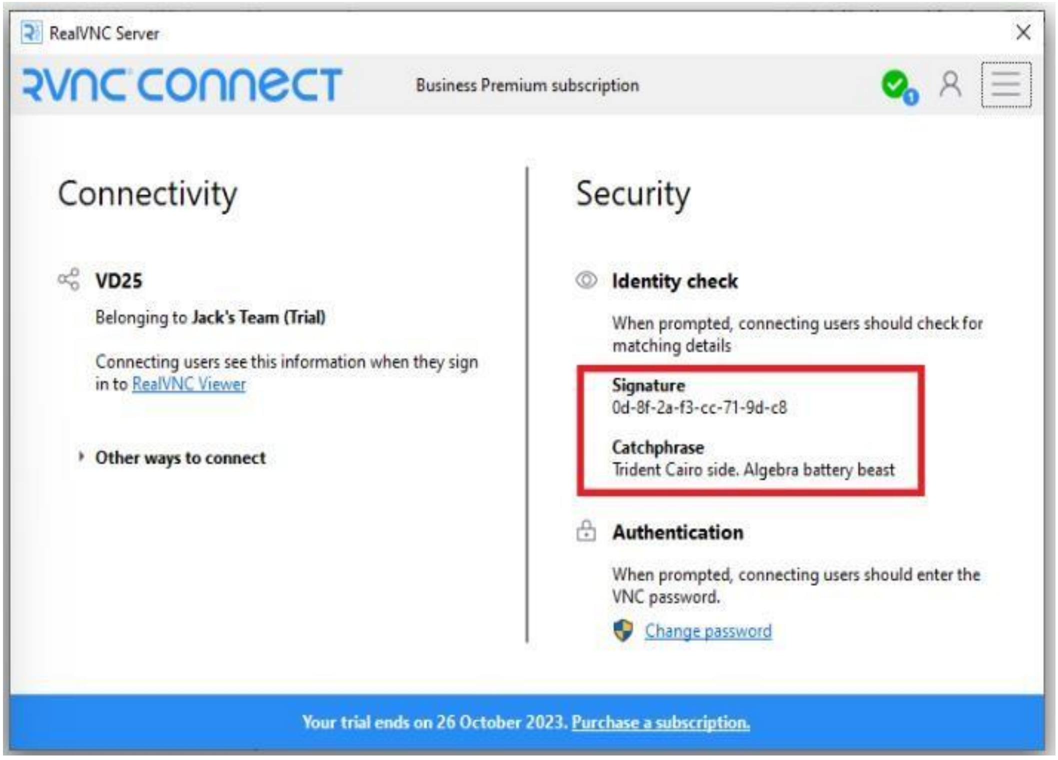


Fig 4.4.3 Connection Succesful

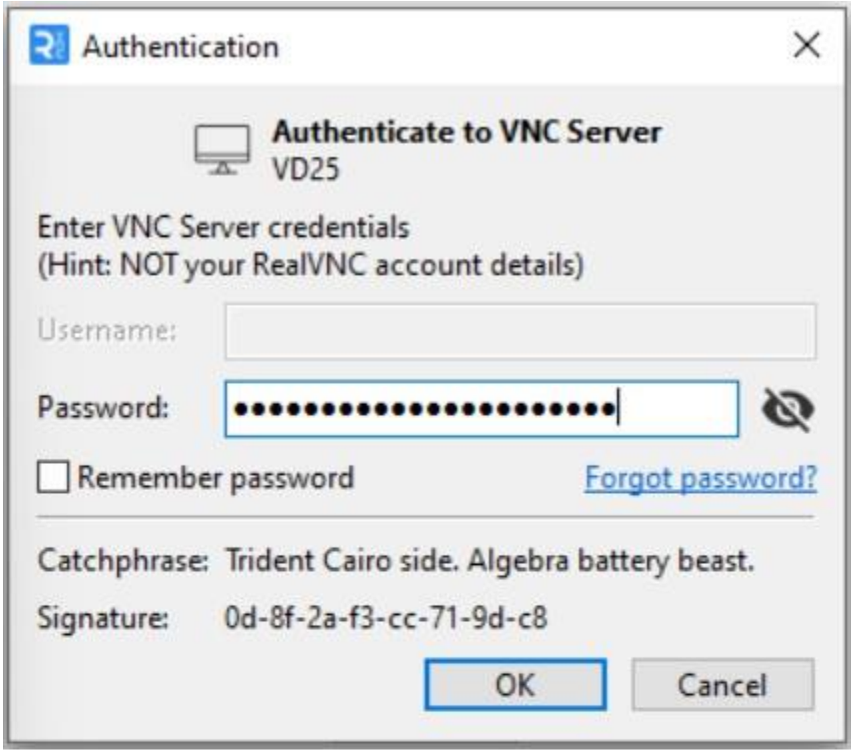
The displayed Catchphrase and Signature against the values shown in RealVNC Server can be checked.



RealVNC Viewer will store the identity which it will automatically check against for future connections to that computer to ensure they match. If they don't, RealVNC Viewer will display a warning to alert you before it makes the connection.

Authenticating:

When we connect to a remote computer, it will be prompted to authenticate. Enter the password previously set.



The integration of RealVNC Viewer significantly enhanced the project's development and testing phase. It provided a convenient and user-friendly means to remotely access and control the Raspberry Pi 5, allowing for real-time monitoring of system behavior, debugging of code, and system administration without the need for physical access to the device. This remote access capability streamlined the development process, facilitated efficient troubleshooting, and ultimately contributed to the successful implementation of the project.

4.5 Python

Python plays a pivotal role in this project, serving as the primary programming language for implementing the system's core functionalities. Its strengths lie in its readability, ease of use, and extensive library support.

Key Features and Benefits:

- **Readability and Maintainability:** Python's clear and concise syntax enhances code readability and maintainability, making it easier to develop, debug, and modify the system.
- **Cross-Platform Compatibility:** Python code can run on various operating systems, including Linux (Raspberry Pi OS), Windows, and macOS, ensuring flexibility and portability.
- **Rich Ecosystem:** Python boasts a vast collection of libraries, including OpenCV, NumPy, and others, that provide powerful tools for image processing, computer vision, and numerical computing.
- **Community Support:** A large and active community provides extensive resources, tutorials, and support, making it easier to find solutions to challenges and learn new concepts.

Chapter 5

SYSTEM IMPLIMENTATION

5.1 Methodology

The project methodology involves a multi-stage approach. First, the Raspberry Pi processes data from the Thermal Imaging Sensor (AMG8833) and the Webcam. This data is analysed using image processing algorithms to detect and track the target. The Raspberry Pi then transmits targeting data to the Arduino. The Arduino, acting as a motor driver, controls the Servo Motor for Camera and Laser Rotation to direct the laser module towards the detected target. Upon confirmation of target lock, the Arduino signals the Missile Firing Servomotor to release the missile. The entire system is powered by a centralized power source.

- **Data Acquisition and Processing:** The Raspberry Pi receives data from the sensors and processes it for target detection and tracking.
- **Target Acquisition and Tracking:** The system calculates the target's position and directs the camera and laser module accordingly.
- **Missile Firing:** Upon target lock, the Arduino signals the missile firing mechanism.
- **Power Management:** The system is powered by a centralized source for efficient operation.

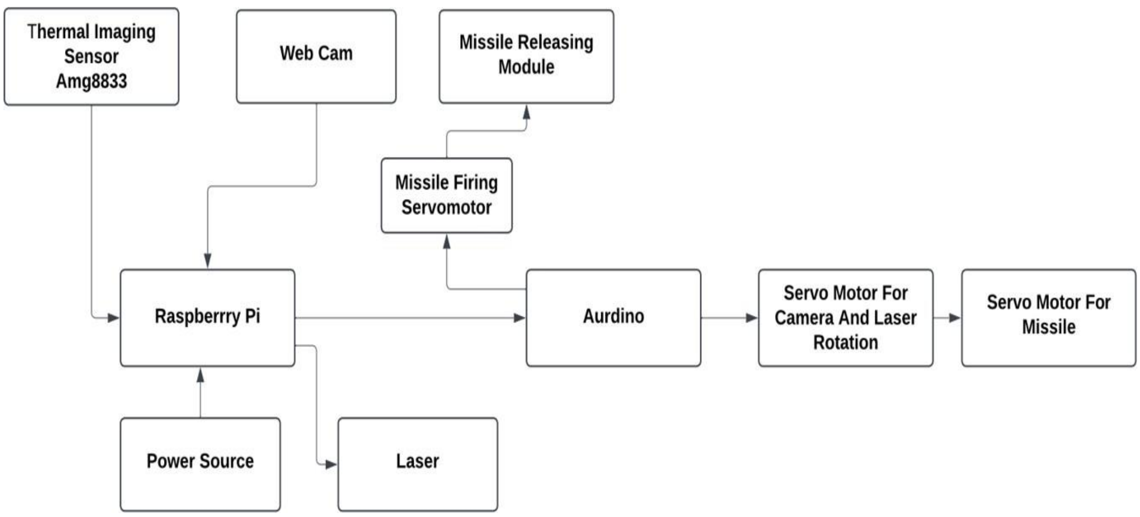


Fig 5.1.1 Block Diagram

This block diagram illustrates the architecture of a Raspberry Pi-Based Thermal and Optical Detection System with Automated Targeting and Real-Time Monitoring. The system integrates various hardware components, including sensors, processing units, and actuators, to achieve automated detection of human movement in the target region and release the missile model toward it. The system is designed around two primary processing units: a Raspberry Pi and an Arduino microcontroller, which handle different aspects of the system's operation.

- Thermal Imaging Sensor (AMG8833): This sensor captures thermal radiation, creating a heat map of the environment. This is crucial for detecting the object in front of the camera. In this project it detects the human body temperature range.
- Web Cam: A standard web camera provides visual input to the system. This visual data is likely used in conjunction with the thermal data for triggering the missile module if needed.
- Raspberry Pi: This serves as the main processing unit for high-level tasks. It receives data from both the thermal imaging sensor and the web cam.
- Missile Releasing Module: This module is the mechanism for deploying the missile. It is a handmade prototype that allows to demonstrate the missile working.
- Missile Firing Servomotor: This servomotor is responsible for triggering the missile module when it receives the signal from the raspberry pi. If a target is detected in the target region, then the servo motor gets the signal from the camera through the raspberry pi and triggers the missile.
- Arduino: This microcontroller acts as motor drive, and controls the servo motors used. It precisely controls the servomotors for camera and laser movement, as well as the missile release and firing mechanisms.
- Servomotor for Camera and Laser Rotation: This servomotor allows for the repositioning of the camera and laser unit. This enables the system of laser, thermal camera and webcammer mounted on it to move in the direction to motion detected.
- Servomotor for Missile: This servomotor is specifically dedicated to controlling the missile's direction. It runs parallel to the thermal camera and web camera.
- Laser: The laser component is used for aiming toward the target body in the region.

System Operation Flow

- The thermal imaging sensor and web cam capture data from the environment.
- This data is sent to the Raspberry Pi for processing and analysis.
- The Raspberry Pi identifies potential targets and makes decisions.

- Commands are sent to the Arduino based on these decisions.
- The Arduino controls the servomotors for camera/laser movement and missile release/firing.
- The laser may be activated for targeting or other purposes.
- The missile is released and potentially guided by the servomotor.

5.2 Circuit Diagram

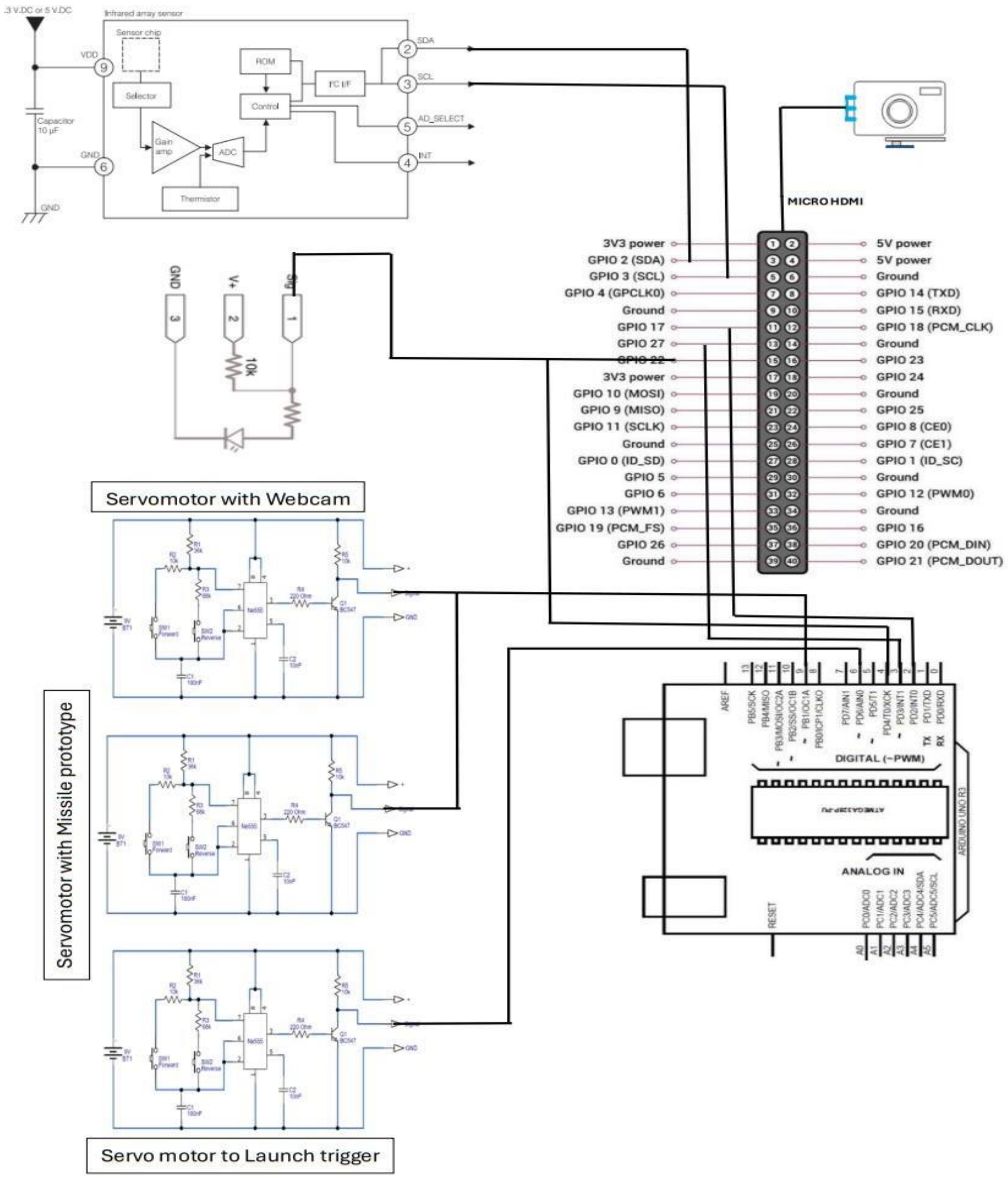


Fig 5.2.1 Circuit Diagram

5.3 Flow Chart

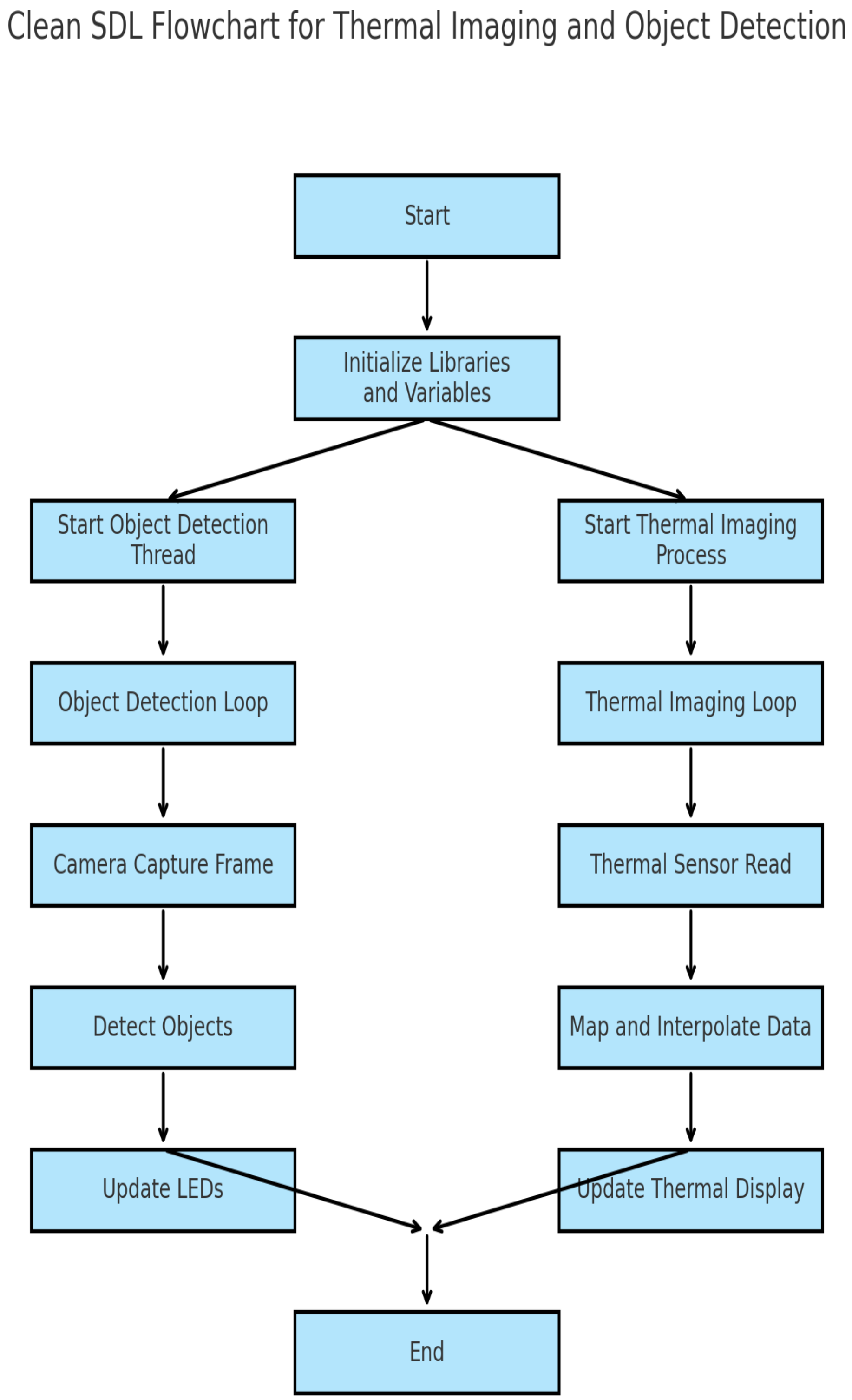


Fig 5.3.1 Flow Chart

Figure 5.3.1 illustrates the software architecture of the object detection and thermal imaging system. The system employs a concurrent processing model, where two separate threads handle object detection and thermal image processing, respectively. This parallel approach ensures real-time performance by preventing blocking between the two tasks.

The object detection thread continuously captures frames from the web camera, applies object detection algorithms. Simultaneously, the thermal imaging thread reads data from the thermal sensor, processes it to create a heat map, and updates the thermal display. Both threads operate independently and continuously, providing simultaneous visual and thermal information about the environment.

The flowchart depicts the sequential steps involved in a system that performs thermal imaging and object detection. It appears to be designed for a system that processes thermal data and camera images concurrently.

Step-by-Step Breakdown:

- Start: The process begins here.
- Initialize Libraries and Variables:
    - This step involves setting up the necessary software libraries and initializing variables required for the system to function. This might include loading OpenCV for image processing, configuring the thermal sensor, and setting initial parameters.
- Start Object Detection Thread and Start Thermal Imaging Process: The flowchart splits into two parallel paths:
    - Object Detection Thread: This thread will run concurrently with the thermal imaging process.
    - Thermal Imaging Process: This path handles the acquisition and processing of thermal data.
- Object Detection Loop:
    - Camera Capture Frame: The system captures a frame from the camera. Detect Objects: Image processing algorithms (like those in OpenCV) are applied to the captured frame to detect objects within the scene.
    - Update LEDs: The detected objects are used to control the state of LEDs, possibly indicating the presence or absence of objects in specific zones.

- Thermal Imaging Loop:
    - Thermal Sensor Read: The system reads temperature data from the thermal sensor.
    - Map and Interpolate Data: The raw thermal data is processed to create a visual representation of the temperature distribution. This might involve mapping raw sensor data to a color palette or interpolating data to create a smoother image.
    - Update Thermal Display: The processed thermal data is displayed on a screen or other output device, providing a visual representation of the temperature distribution in the scene.
- End: After completing the object detection and thermal imaging processes, the system reaches the end of the flowchart.

5.4 Working Principle

This system operates on the principle of real-time object detection, tracking, and automated targeting. It integrates thermal and optical imaging with precise motor control to engage targets effectively.

- **Data Acquisition:** The Thermal Imaging Sensor (AMG8833) captures infrared radiation emitted by objects in the field of view and generates a thermal image. The Webcam captures a visual image of the scene.
- **Image Processing:** The Raspberry Pi receives and processes the thermal and visual images. OpenCV library is utilized to perform image processing tasks, including:
    - **Pre-processing:** Image enhancement, noise reduction, and feature extraction.
    - **Object Detection:** Algorithms like Haar Cascades or HOG are employed to detect objects in both thermal and visual images.
    - **Object Tracking:** Tracking algorithms, such as Kalman filtering or optical flow, are used to follow the movement of detected objects in consecutive frames.
- **Target Classification:** The system analyses the detected objects to classify them as targets or non-targets based on predefined criteria (e.g., size, shape,

temperature).

- **Targeting and Engagement:** Upon detection of a target, the Raspberry Pi calculates the target's position and transmits this information to the Arduino. The Arduino controls the Servo Motor for Camera and Laser Rotation to direct the laser module towards the calculated target position. Once the target is locked, the Arduino signals the Missile Firing Servomotor to initiate the missile release sequence.
- **Real-time Monitoring and Control:** The Raspberry Pi provides real-time monitoring of the system's status, including detected objects, target tracking data, and system performance. A user interface (e.g., developed using Tkinter or Flask) enables remote monitoring and control of the system, allowing for adjustments to parameters and system configuration.
- **Power Management:** The system is powered by a centralized power source, ensuring efficient and reliable operation of all components.

Chapter 6

RESULTS AND CONCLUSION

6.1 Future Scope

The current implementation of the Raspberry Pi-based thermal and optical detection system provides a strong foundation for future enhancements and expanded capabilities. Building upon this platform, several avenues for future research and development can be explored.

- **Machine Learning Integration:** Implement object recognition using deep learning models like CNNs for improved accuracy and robustness. Train models on larger datasets to enhance performance in diverse environments.
- **Enhanced Targeting:** Incorporate predictive targeting algorithms to anticipate target movements. Consider environmental factors like wind speed and direction for more accurate targeting.
- **System Miniaturization:** Design a smaller and more compact system for deployment in various field environments. Integrate the system onto unmanned aerial vehicles (UAVs) or other mobile platforms.
- **Sensor Fusion:** Integrate other sensors like lidar, radar, or GPS for enhanced situational awareness and improved target tracking.
- **Advanced User Interface:** Develop a more user-friendly and interactive graphical user interface (GUI) for system control and data visualization.
- **Wireless Communication:** Explore wireless communication options for remote operation and data transmission.
- **Power Optimization:** Investigate techniques to reduce power consumption for extended operation in the field.

6.2 Result

The project successfully delivers its purpose and shoots the amino at the target when it is detected within the monitoring range. The raspberry pi is the main processor that controls and navigates the entire process. The project model can present these results based on the surrounding:

- **No Object Detected:** When no objects or humans are present within the monitoring range, the system remains in a standby mode, awaiting the detection of a potential target.
- **Non-Target Object Detected:** If an object is detected but does not correspond to the predefined target class (e.g., if the system is configured to target humans, and an animal is detected), the system remains in a monitoring state without engaging the targeting mechanism.
- **Target Object Detected:** Upon detection of a target object from the predefined class, the system initiates the targeting sequence. The laser module is activated, and the system precisely marks the target using the laser beam. Subsequently, a simulated "missile launch" (or a representative action) is triggered to engage and "destroy" the target.

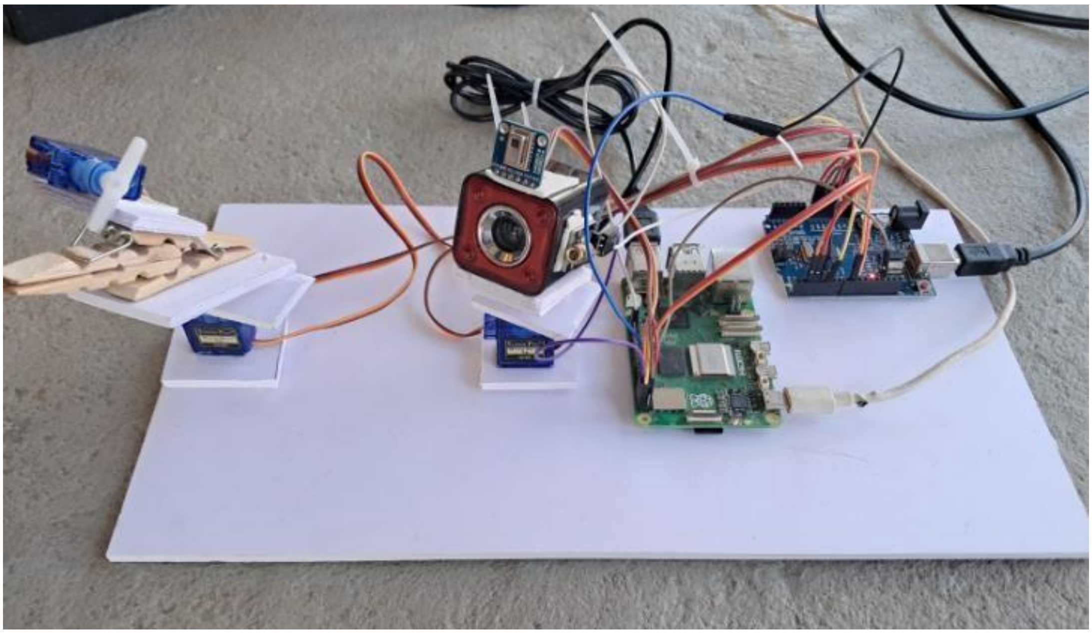


Fig 6.2.1 Project Model

6.3 Advantages, Limitations and Conclusion

ADVANTAGES

- Enhanced Security: Provides 24/7 surveillance, enabling early detection of potential threats.
- Improved Situational Awareness: Offers a comprehensive view of the monitored area, enhancing overall security posture.
- Versatility: Adaptable to various applications, including perimeter security, building surveillance, and critical infrastructure protection.
- Cost-Effectiveness: Utilizes low-cost hardware (Raspberry Pi) and leverages open-source software, making it a more budget-friendly solution compared to traditional systems.
- Leverages Modern Technology: Employs cutting-edge technologies like Raspberry Pi, AI/ML, and thermal imaging for enhanced capabilities.
- Continuous Improvement: Benefits from ongoing advancements in these technologies, allowing for future upgrades and enhancements.
- Remote Monitoring: Enables remote monitoring and control of the system, allowing for efficient management and response.
- Customizable: Can be customized to specific needs by adjusting parameters, integrating with other systems, and implementing advanced AI/ML algorithms.

LIMITATIONS

- Computational Constraints: The Raspberry Pi may have limitations in processing complex AI/ML models in real-time, especially with high-resolution thermal imagery.
- Power consumption can increase during continuous operation and processing, potentially requiring efficient power management.
- Environmental Factors: Extreme weather conditions (e.g., heavy rain, snow) can affect the accuracy of thermal imaging and sensor readings.
- Physical obstructions (e.g., trees, buildings) can hinder the field of view and limit detection capabilities.

- False Positives/Negatives: Environmental noise (e.g., temperature variations, background heat sources) can lead to false alarms or missed detections.
- Limitations in AI/ML algorithms can result in inaccurate predictions or misinterpretations.
- Privacy Concerns: Unrestricted use of surveillance systems can raise concerns about privacy violations.
- Data Security: Proper data security measures are essential to protect sensitive information collected by the system from unauthorized access or breaches.
- Maintenance Requirements: The system may require regular maintenance, including software updates, hardware checks, and calibration of sensors.
- Ethical Considerations: Careful consideration must be given to the ethical implications of using surveillance technology, including potential biases and misuse.
- Susceptibility to Interference: The system may be susceptible to interference from electromagnetic sources or cyberattacks, potentially disrupting its operation.

CONCLUSION

This project successfully demonstrates a Raspberry Pi 5-based thermal and optical detection system integrated with automated targeting and missile firing capabilities. The system effectively monitors the surroundings for intruders. Upon detecting a target, it utilizes a combination of thermal camera, webcam, and AI/ML algorithms to accurately locate and track the target. Subsequently, the system employs a custom-designed missile module, controlled by precision motors, to aim and fire at the designated target. In the absence of intruders, the system maintains a vigilant state, continuously monitoring the environment for potential threats.

This project showcases the potential of embedded systems in developing advanced security and defence applications. While further research and development are necessary to refine the system's accuracy, address potential limitations, and ensure safety protocols, this prototype provides a strong foundation for future iterations of autonomous defence systems.

REFERENCES

[1] S. Namiki, K. Yokoyama, S. Yachida, T. Shibata, H. Miyano and M. Ishikawa, "Online Object Recognition Using CNN-based Algorithm on High-speed Camera Imaging: Framework for fast and robust high-speed camera object recognition based on population data cleansing and data ensemble," 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 2021, pp. 2025-2032, doi: 10.1109/ICPR48806.2021.9413042. keywords: {Image recognition;Sociology;Object detection;Cameras;Real-time systems;Robustness;Pattern recognition;Applications of computer vision;Object recognition;Image classification},

[2] Raspberry Pi 5 : The new Raspberry Pi family with more computation power and AI integration , Mohapmed Fezari (1), Ali Al-Dahoud (2) , (1) Badji Mokhtar Annaba university , dept Electronics Algeria,(2) Al-Zaytoonah university Amman, Jordan.

[3] Y. -C. Chiu, H. -W. Hsu and C. -Y. Tsai, "Person Tracking Control of Mobile Robots Using a Lightweight Object Detection and Tracking System," 2024 8th International Conference on Robotics and Automation Sciences (ICRAS), Tokyo, Japan, 2024, pp. 21-25, doi: 10.1109/ICRAS62427.2024.10654470. keywords: {Deep learning;Target tracking;Pedestrians;Object detection;Detectors;Network architecture;Control systems;Multi-object tracking;pedestrian detection;deep learning;person tracking control;mobile robots},

[4] S. Iwasaki, T. Kawamura, T. Sasama, K. Sugahara and H. Kakiuchi, "Detection method to continue tracking of automatic human tracking system," Proceedings of SICE Annual Conference 2010, Taipei, Taiwan, 2010, pp. 573-574. keywords: {Target tracking;Servers;Mobile agents;Humans;Cameras;Feature extraction;Mobile Agent;Human Tracking System},

[5] R. Jelinsks and M. Gorobetz, "Improved Centroid-based Object Detection and Tracking System for Automated Manufacturing Line," 2022 IEEE 63th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON), Riga, Latvia, 2022, pp. 1-5, doi: 10.1109/RTUCON56726.2022.9978844. keywords:

{Electrical engineering;Manufacturing processes;Machine learning algorithms;Manufactured products;Object detection;Belts;Product design;Computer Vision;Object detection;Object tracking;Automated Manufacture},

[6] P. Manikandan, C. Swedheetha, S. N. C. Thulasi, V. V. R. Chennuru, J. K. R. Vardireddy and Y. V. S. S. Deepak, "Automatic Train Tracking and Announcement System," 2023 5th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2023, pp. 1659-1663, doi: 10.1109/ICAC3N60023.2023.10541459. keywords: {GSM;Target tracking;Urban areas;Message services;Real-time systems;Rail transportation;Satellite images;Voice Announcement system;Train tracking;Display system;GPS location detection;SMS alerting}

[7] I. Dagher and R. Nachar, "Face recognition using IPCA-ICA algorithm," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 6, pp. 996-1000, June 2006, doi: 10.1109/TPAMI.2006.118.
keywords: {Face recognition;Independent component analysis;Principal component analysis;Covariance matrix;Humans;Blind source separation;Eigenvalues and

eigenfunctions;Filters;Image storage;Symmetric matrices;IPCA-ICA;Principal component analysis (PCA);independent component analysis (ICA);principal non-Gaussian directions;image processing;blind source separation.},

[8] H. Nishimura, Y. Ozasa, Y. Ariki and M. Nakano, "Object Recognition by Integrated Information Using Web Images," 2013 2nd IAPR Asian Conference on Pattern Recognition, Naha, Japan, 2013, pp. 657-661, doi: 10.1109/ACPR.2013.140. keywords: {Speech recognition;Image recognition;Object recognition;Speech;Robots;Hidden Markov models;Computational modeling;object recognition;Web;integration;speech;image},

[9] K. Kim, S. Kang, J. Kim, J. Lee, J. Kim and J. Kim, "Multiple objects recognition for industrial robot applications," 2013 10th International Conference on Ubiquitous Robots

and Ambient Intelligence (URAI), Jeju, Korea (South), 2013, pp. 257-259, doi: 10.1109/URAI.2013.6677361. keywords: {Object recognition;Shape;Sensors;Feature extraction;Lighting;Image edge detection;Object segmentation;Object recognition;Edge detection;Pose estimation;Object segmentation},

[10] S. Bandong, Y. Y. Nazaruddin and E. Joelianto, "Container Detection System Using CNN Based Object Detectors," 2021 International Conference on Instrumentation, Control, and Automation (ICA), Bandung, Indonesia, 2021, pp. 106-111, doi: 10.1109/ICA52848.2021.9625663. keywords: {Automation;Trade agreements;Image recognition;Tracking;Instruments;Loading;Object detection;Mobilenet;Resnet;Faster RCNN;Port Automation;Container Detection},

[11] A. Shaik, R. T. Prabu and S. Radhika, "Detection of Face Mask using Convolutional Neural Network (CNN) based Real-Time Object Detection Algorithm You Only Look Once-V3 (YOLO-V3) Compared with Single-Stage Detector (SSD) Algorithm to Improve Precision," 2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), Chennai, India, 2023, pp. 1-6, doi: 10.1109/ACCAI58221.2023.10200890. keywords: {COVID-19;Detectors;Object detection;Real-time systems;Convolutional neural networks;Informatics;Faces;Novel Yolo-v3;Single Stage Detection (SSD);Face Mask Detection;Convolutional Neural Network;COVID - 19;Object Detection},

[12] Human Detection using Infrared Thermal Imaging System Min Goo Lee, Yong Kuk Park, Kyung Kwon Jung Embedded Software Convergence Research Center, Korea Electronics Technology Institute Seongnam-si, Gyeonggi-do, 463-816, Korea.

[13] Thermal Imaging Detection System: A Case Study for Indoor Environments, by Martin Drahansky, Michal Charvat, Ivo Macek, Jitka Mohelnikova

[14] A.M. Anushree Kirthika (2014), "Missile Detection and Automatic Destroy System" Volume 4, Number 1, pp. 1-6.

[15] S. Murakami, Y. Nishida, T. Hori, H. Mizoguchi, Detecting Human Head Location Using a Simply Installed Ultrasonic Radar System, in Proceedings of the 22nd Annual Conference of the Robotics Society of Japan, 1A23(1)-(2), September 2004.

[16] Fu-Kuang Yeh, Kai-Yuan Cheng, and Li-Chen Fu." Variable Structure-Based Nonlinear Missile Guidance/Autopilot Design With Highly Maneuverable Actuators "IEEE transaction on control systems technology,vol.12,no.6,november 2004.

[17] Arduino UNO R3_[Online], available https://docs.arduino.cc/hardware/uno-rev3/

[18] Servo motor SG90 [online] , available http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf

[19] A. Kumari Sirivarshitha, K. Sravani, K. S. Priya and V. Bhavani, "An approach for Face Detection and Face Recognition using OpenCV and Face Recognition Libraries in Python," 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2023, pp. 1274-1278, doi: 10.1109/ICACCS57279.2023.10113066. keywords: {Databases;Face recognition;Surveillance;Passwords;Libraries;Software;Face detection;Face Detection;Face Recognition;Face Alignment;Feature Extraction;Python;OpenCV Library;face Recognition Library},

[20] M. C. Xenya, C. Kwayie and K. Quist-Aphesti, "Intruder Detection with Alert Using Cloud Based Convolutional Neural Network and Raspberry Pi," 2019 International Conference on Computing, Computational Modelling and Applications (ICCMA), Cape Coast, Ghana, 2019, pp. 46-464, doi: 10.1109/ICCMA.2019.00015. keywords: {Cloud computing;Feature extraction;Cameras;Convolutional neural networks;Data models;Training;Monitoring;component;convolutional neural network;cloud computing;raspberry p;intruder detection;MMS alert},

[21] Huu-Quoc Nguyen, Ton Thi Kim Loan, Bui Dinh Mao and Eui-Nam Huh, "Low cost real-time system monitoring using Raspberry Pi," 2015 Seventh International Conference

on Ubiquitous and Future Networks, Sapporo, Japan, 2015, pp. 857-859, doi: 10.1109/ICUFN.2015.7182665. keywords: {Motion detection;Cameras;Streaming media;Surveillance;Boards;Computers;Video surveillance;Raspberry Pi;Motion Detection;streaming},

APPENDIX A

Thermal Sensor AMG8833

Types

Tape and reel package : 1,000 pcs

Product name	Number of pixel	Operating voltage	Amplification factor	Part number
Infrared array sensor Grid-EYE High performance type	64 (Vertical 8 × Horizontal 8 Matrix)	3.3 V.DC	High performance type High gain	AMG8833
			High performance type Low gain	AMG8834
		5.0 V.DC	High performance type High gain	AMG8853
			High performance type Low gain	AMG8854

Rating

Item	Performance	
	High gain	Low gain
Applied voltage	3.3 V.DC±0.3 V.DC or 5.0 V.DC±0.5 V.DC	
Temperature range of measuring object	0 °C to 80 °C +32 °F to +176 °F	−20 °C to 100 °C −4 °F to +212 °F
Operating temperature range	0 °C to 80 °C +32 °F to +176 °F	−20 °C to 80 °C −4 °F to +176 °F
Storage temperature range	−20 °C to 80 °C −4 °F to +176 °F	−20 °C to 80 °C −4 °F to +176 °F

Table1 Types and Rating

Absolute maximum ratings

Item	Absolute maximum ratings	Terminal
Applied voltage	−0.3 V.DC to 6.5 V.DC	VDD
Input voltage	−0.3 V.DC to VDD +0.3 V.DC	SCL, SDA, AD_SELECT
Output sink current	−10 mA to 10 mA	INT, SDA
Static electricity (Human body model)	1 kV	All terminals
Static electricity (Machine model)	200 V	All terminals

Table 2 Absolute Maximum Rating

Characteristics

Item	Performance	
	High performance type High gain	High performance type Low gain
Temperature accuracy	Typical ±2.5 °C ±4.5 °F	Typical ±3.0 °C ±5.4 °F
Human detection distance *1	7 m or less (reference value) 22.966 ft	
NETD *2	Typ. 0.05 °C 32.900 °F 1 Hz Typ. 0.16 °C 32.288 °F 10 Hz	
Viewing angle	Typical 60 °	
Optical axis gap	Within Typical ±5.6 °	
Current consumption	Typical 4.5 mA (normal mode) Typical 0.2 mA (sleep mode) Typical 0.8 mA (stand-by mode)	
Setup time	Typical 50 ms (Time to enable communication after setup) Typical 15 s (Time to stabilize output after setup)	

Table 3 Characteristics

Performance

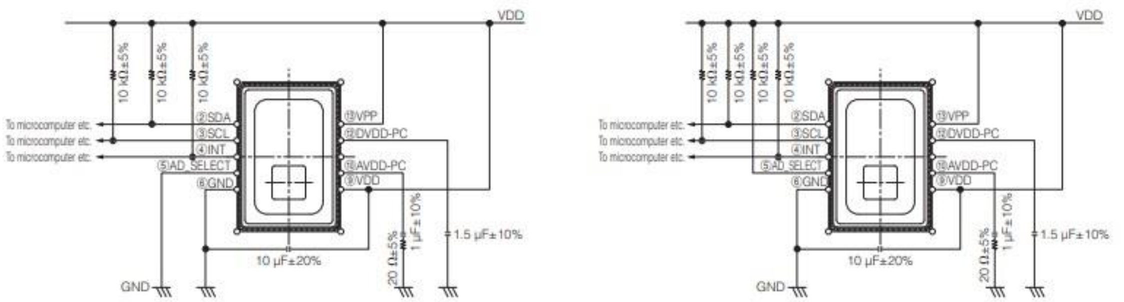
Item	Performance
Number of pixel	64 (Vertical 8 × Horizontal 8 Matrix)
External interface	I^{2}C (fast mode)
Frame rate	Typical 10 frames/sec or 1 frame/sec
Operating mode *1	Normal Sleep Stand-by (10 sec or 60 sec intermittence)
Output mode	Temperature output
Calculate mode	No moving average or Twice moving average
Temperature output resolution	0.25 °C 32.45 °F
Number of sensor address	2 (I^{2}C slave address)
Thermistor output temperature range	−20 °C to 80 °C −4 °F to +176 °F
Thermistor output resolution	0.0625 °C 32.1125 °F

Table 4 Performance

External design details

(1) In case of setting I^{2}C slave address of the sensor 1101000
✻ Connect terminal ⑤ (AD_SELECT) to GND.

(2) In case of setting I^{2}C slave address of the sensor 1101001
✻ Connect terminal ⑤ (AD_SELECT) to VDD.



This circuit is an example to drive infrared array sensor "Grid-EYE", so that we will not take any responsibility of loss which is due to this circuit.

Fig 1 External circuit Diagram of AMG8833

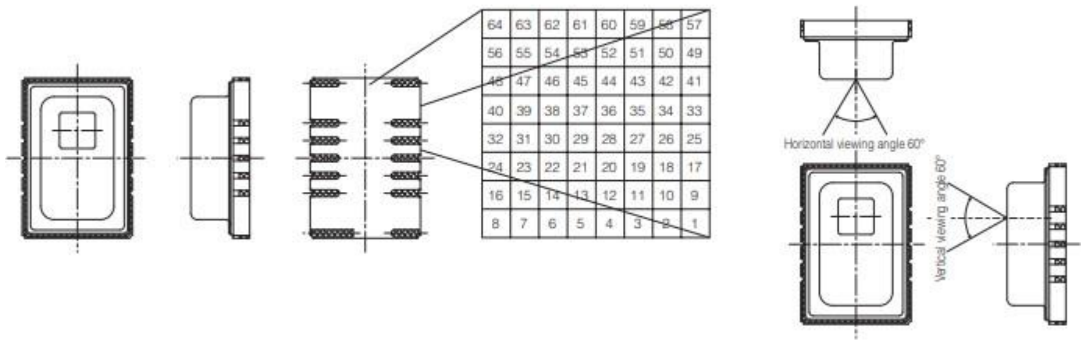
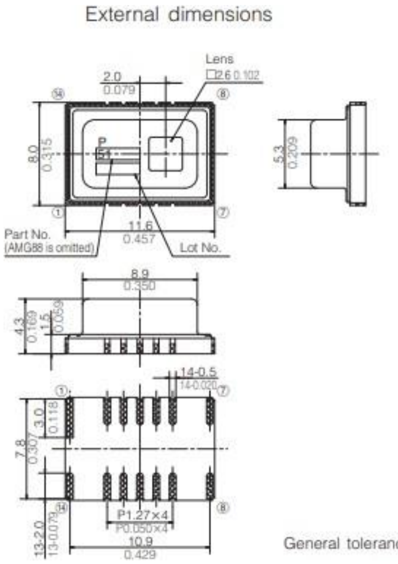
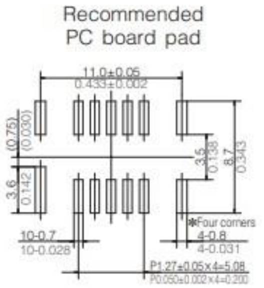


Fig 2 Pixel array and viewing field



Number	Terminal Name	Number	Terminal Name
①	NC	⑧	NC
②	SDA	⑨	VDD
③	SCL	⑩	AVDD-PC
④	INT	⑪	NC
⑤	AD_SELECT	⑫	DVDD-PC
⑥	GND	⑬	VPP
⑦	NC	⑭	NC

Note : Leave terminal "NC (No.①,⑦,⑧,⑪ and ⑭)" unconnected.
Make electrical potential of terminals ⑨ and ⑬ the same.



General tolerance : ±0.2 ±0.08

unit : mm inch

Fig 3 Dimensions

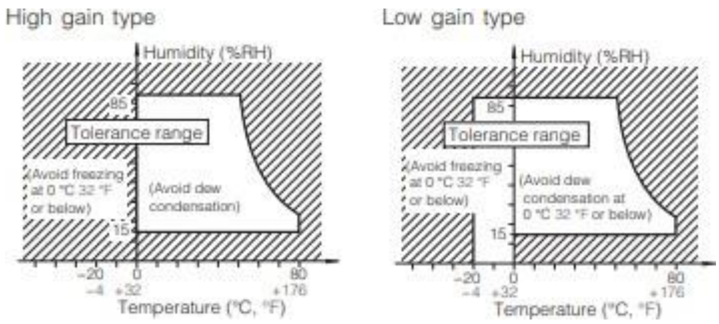
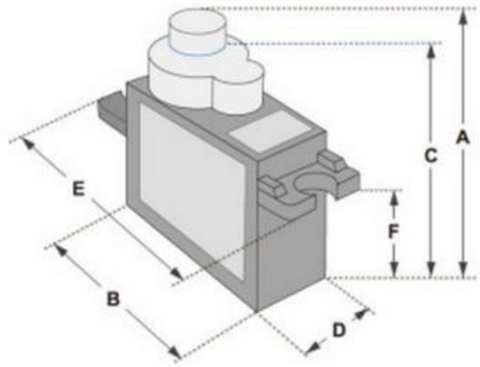


Fig 4 Range of Using ambient temperature

APPENDIX B

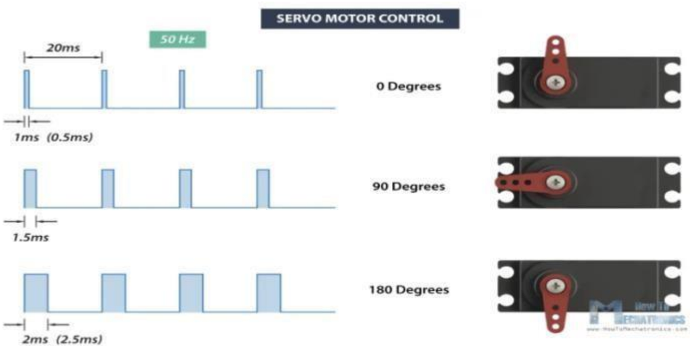
Servo Motor SG90

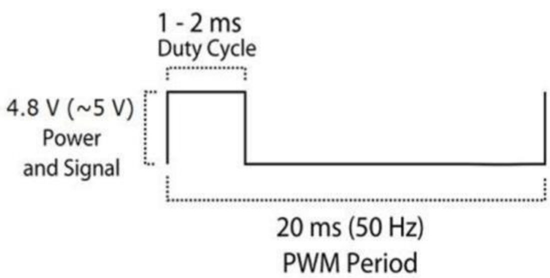


Dimensions & Specifications
A (mm) : 32
B (mm) : 23
C (mm) : 28.5
D (mm) : 12
E (mm) : 32
F (mm) : 19.5
Speed (sec) : 0.1
Torque (kg-cm) : 2.5
Weight (g) : 14.7
Voltage : 4.8 - 6

Fig 5 Servo Motor Dimensions

TP SG90 Specifications: Dimensions (L x W x H) = 0.86 x 0.45 x 1.0 inch (22.0 x 11.5 x 27 mm), Weight = 0.32 ounces (9 grams) Weight with wire and connecter = 0.37 ounce (10.6 grams) ,Stall Torque at 4.8 volts = 16.7 oz/in (1.2 kg/cm) ,Operating Voltage = 4.0 to 7.2 volts ,Operating Speed at 4.8 volts (no load) = 0.12 sec/ 60 degrees ,Connector Wire Length = 9.75 inches (248 mm) The Gear ratio of this Sevo motor gear is 1:64 .





Duty Cycle: The duty cycle of the PWM signal, which is the duration the signal is high within each period, ranges from 1ms to 2ms. This variation in duty cycle is what controls the position of the servo motor:

- 1ms duty cycle: Moves the servo to the 0° position.
- 1.5ms duty cycle: Moves the servo to the 90° position (neutral).
- 2ms duty cycle: Moves the servo to the 180° position.

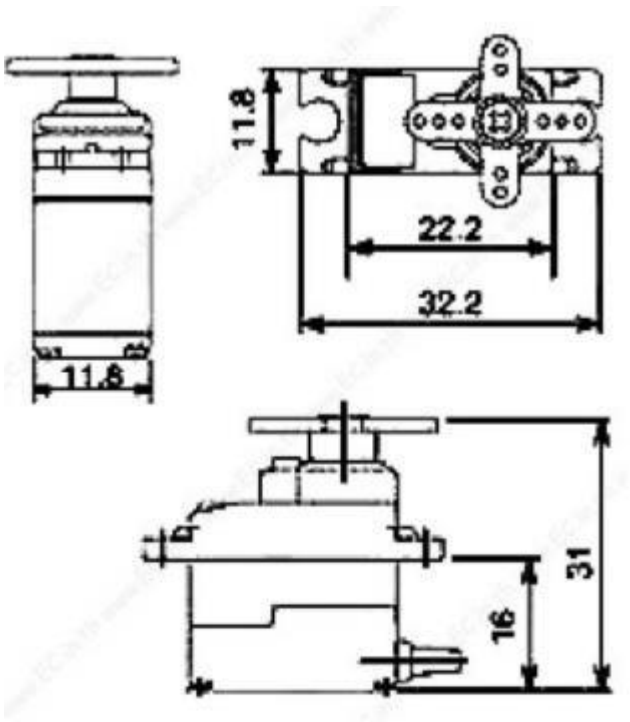


Fig 6 Dimension

APPENDIX C

Arduino UNO R3

Pins	
Built-in LED Pin	13
Digital I/O Pins	14
Analog input pins	6
PWM pins	6
Communication	
UART	Yes
I2C	Yes
SPI	Yes
Power	
I/O Voltage	5V
Input voltage (nominal)	7-12V
DC Current per I/O Pin	20 mA
Power Supply Connector	Barrel Plug
Clock speed	
Main Processor	ATmega328P 16 MHz
USB-Serial Processor	ATmega16U2 16 MHz
Memory	
ATmega328P	2KB SRAM, 32KB FLASH, 1KB EEPROM
Dimensions	
Weight	25 g
Width	53.4 mm
Length	68.6 mm

Table 5 Arduino Working requirements

- It is an ATmega328P based Microcontroller
- The Operating Voltage of the Arduino is 5V
- The recommended input voltage ranges from 7V to 12V
- The i/p voltage (limit) is 6V to 20V
- Digital input and output pins-14
- Digital input & output pins (PWM)-6
- Analog i/p pins are 6
- DC Current for each I/O Pin is 20 mA
- DC Current used for 3.3V Pin is 50 mA
- Flash Memory -32 KB, and 0.5 KB memory is used by the boot loader
- SRAM is 2 KB
- EEPROM is 1 KB
- The speed of the CLK is 16 MHz
- In Built LED
- Length and width of the Arduino are 68.6 mm X 53.4 mm
- The weight of the Arduino board is 25 g

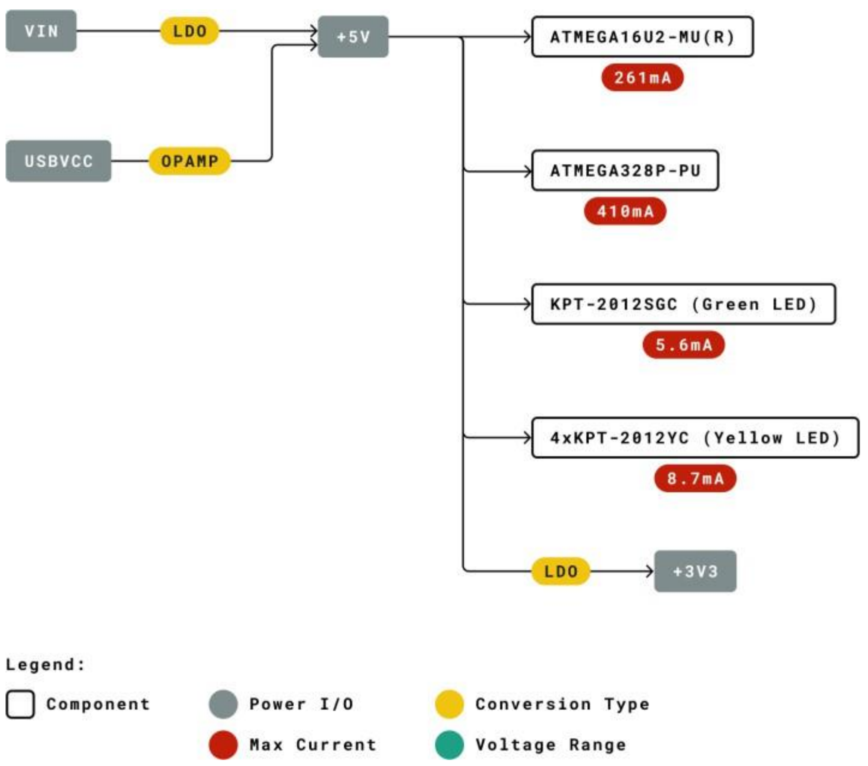


Fig7 Power Tree of Arduino UNO R3

Pin	Function	Type	Description
1	NC	NC	Not connected
2	IOREF	IOREF	Reference for digital logic V - connected to 5V
3	Reset	Reset	Reset
4	+3V3	Power	+3V3 Power Rail
5	+5V	Power	+5V Power Rail
6	GND	Power	Ground
7	GND	Power	Ground
8	VIN	Power	Voltage Input
9	A0	Analog/GPIO	Analog input 0 /GPIO
10	A1	Analog/GPIO	Analog input 1 /GPIO
11	A2	Analog/GPIO	Analog input 2 /GPIO
12	A3	Analog/GPIO	Analog input 3 /GPIO
13	A4/SDA	Analog input/I2C	Analog input 4/I2C Data line
14	A5/SCL	Analog input/I2C	Analog input 5/I2C Clock line

Table 6 JANALOG

Pin	Function	Type	Description
1	D0	Digital/GPIO	Digital pin 0/GPIO
2	D1	Digital/GPIO	Digital pin 1/GPIO
3	D2	Digital/GPIO	Digital pin 2/GPIO
4	D3	Digital/GPIO	Digital pin 3/GPIO
5	D4	Digital/GPIO	Digital pin 4/GPIO
6	D5	Digital/GPIO	Digital pin 5/GPIO
7	D6	Digital/GPIO	Digital pin 6/GPIO
8	D7	Digital/GPIO	Digital pin 7/GPIO
9	D8	Digital/GPIO	Digital pin 8/GPIO
10	D9	Digital/GPIO	Digital pin 9/GPIO
11	SS	Digital	SPI Chip Select
12	MOSI	Digital	SPI1 Main Out Secondary In
13	MISO	Digital	SPI Main In Secondary Out
14	SCK	Digital	SPI serial clock output
15	GND	Power	Ground
16	AREF	Digital	Analog reference voltage
17	A4/SD4	Digital	Analog input 4/I2C Data line (duplicated)
18	A5/SD5	Digital	Analog input 5/I2C Clock line (duplicated)

Table 7 JDIGITAL

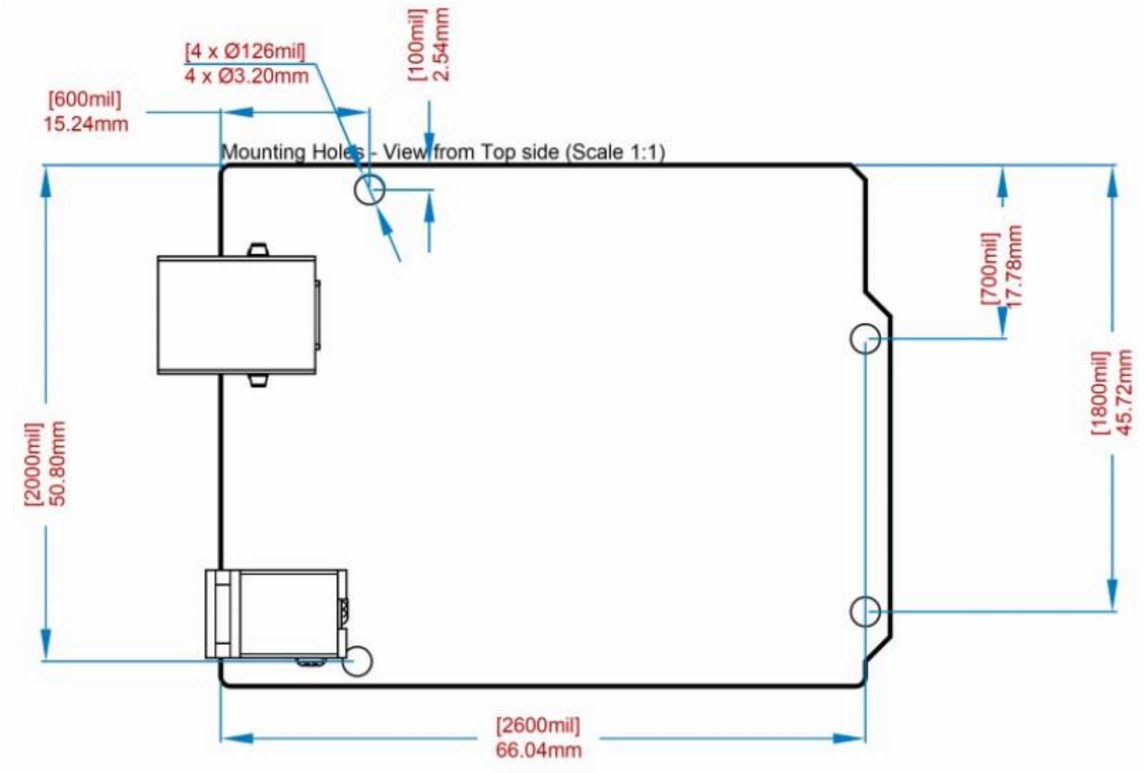


Fig 8 Body outline of Arduino UNO R3

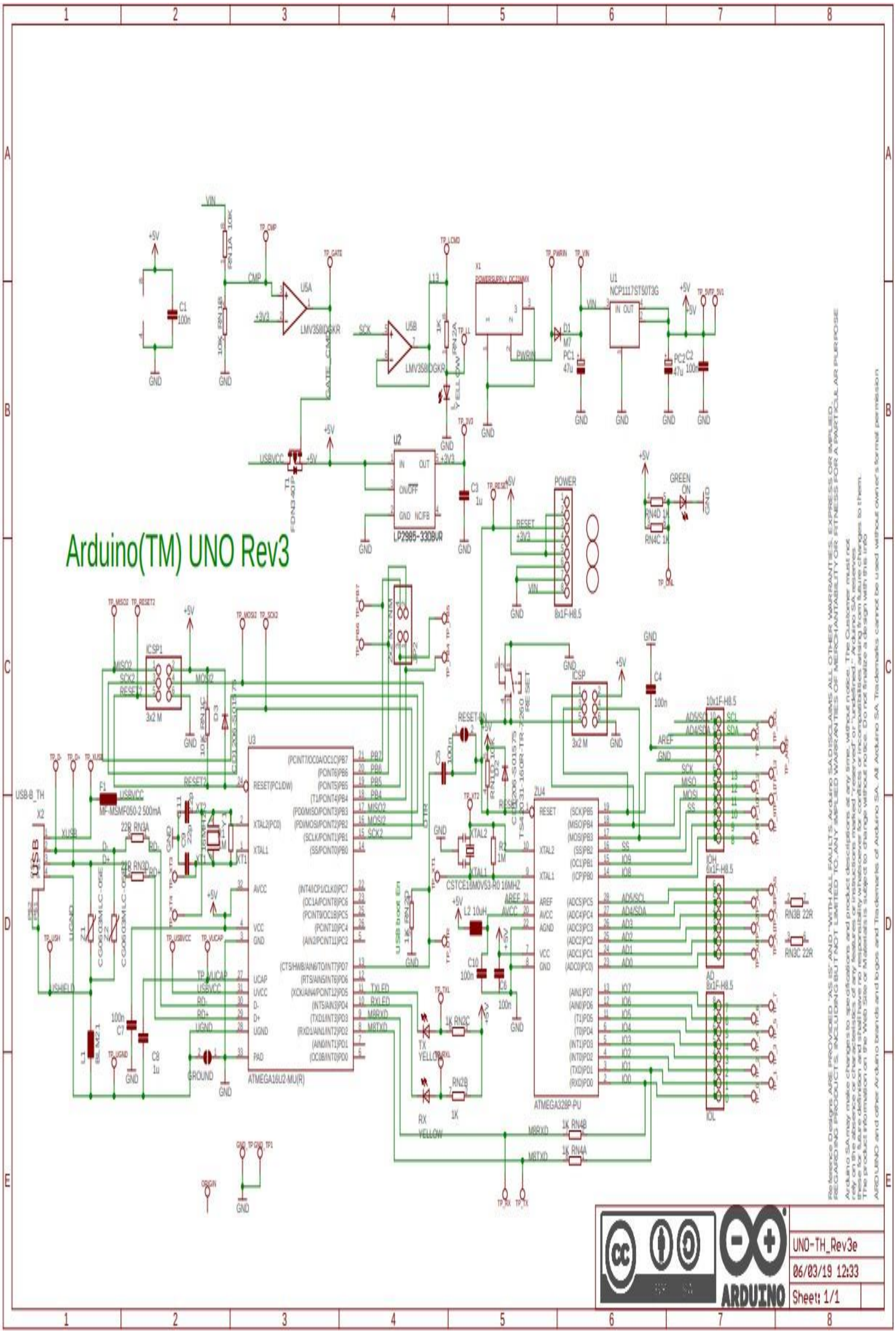


Fig 9 Internal Circuit

APPENDIX D

Raspberry Pi 5

Raspberry Pi is a series of small single board computers (SBCs) developed in the United Kingdom. The original Raspberry Pi computer was developed by the Raspberry Pi Foundation in association with Broadcom. Since 2012, all Raspberry Pi products have been developed by Raspberry Pi Ltd, which began as a wholly-owned subsidiary of the Foundation.

The Raspberry Pi project originally leaned toward the promotion of teaching basic computer science in schools. The original model became more popular than anticipated, selling outside its target market for diverse uses such as robotics, home automation, industrial automation, and by computer and electronic hobbyists, because of its low cost, modularity, open design, and its adoption of the HDMI and USB standards.

The Raspberry Pi became the best-selling British computer in 2015, when it surpassed the ZX spectrum in unit sales.

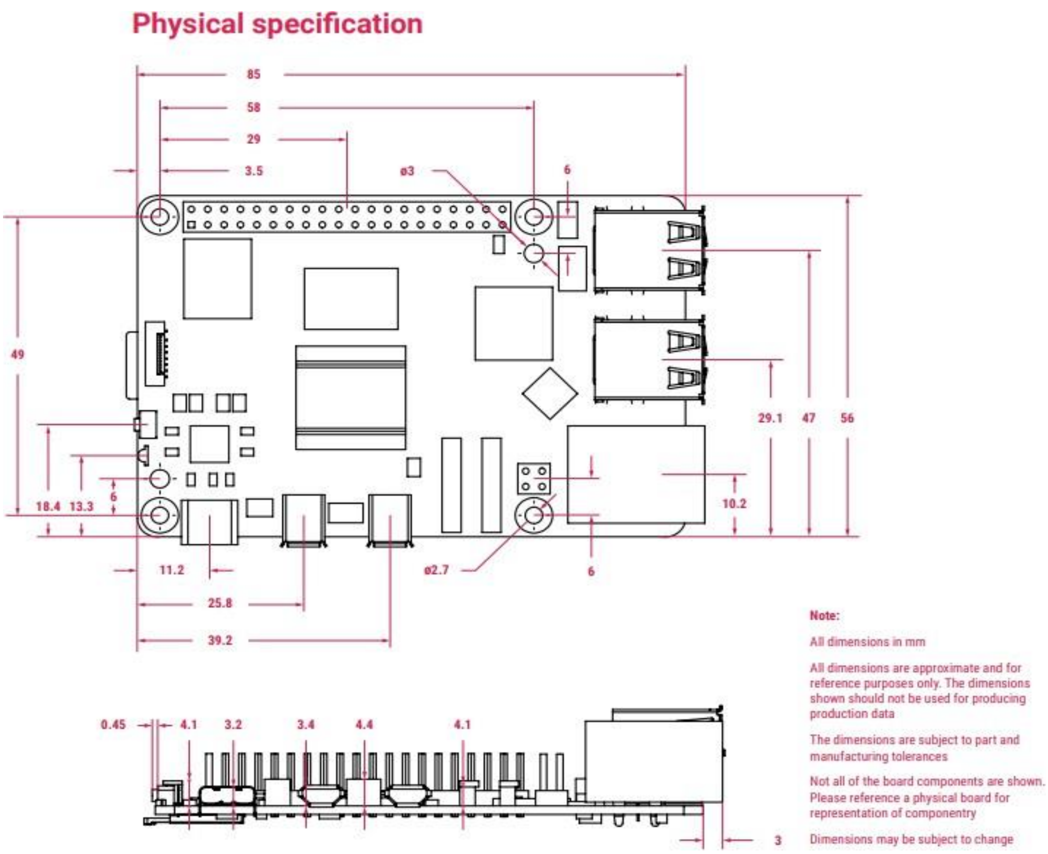


Fig 10 Physical Specification

The **Raspberry Pi 5** was announced in September 2023. It uses a 2.4GHz quad-core 64-bit Arm Cortex-A76 CPU and a Video-Core VII GPU, with the improvements in hardware and software reportedly making the Pi 5 more than twice as powerful as the Pi 4. It has an I/O controller designed in-house, a power button, and an RTC chip (which requires an external battery). At launch, the Pi 5 was available with either 4 or 8 GB of RAM, at US$60 and US$80; a 2 GB variant was released in August 2024 at US$50. The Pi 5 lacks a 3.5 millimetre audio jack, so Bluetooth, HDMI, USB audio or an Audio HAT are the options for audio output.

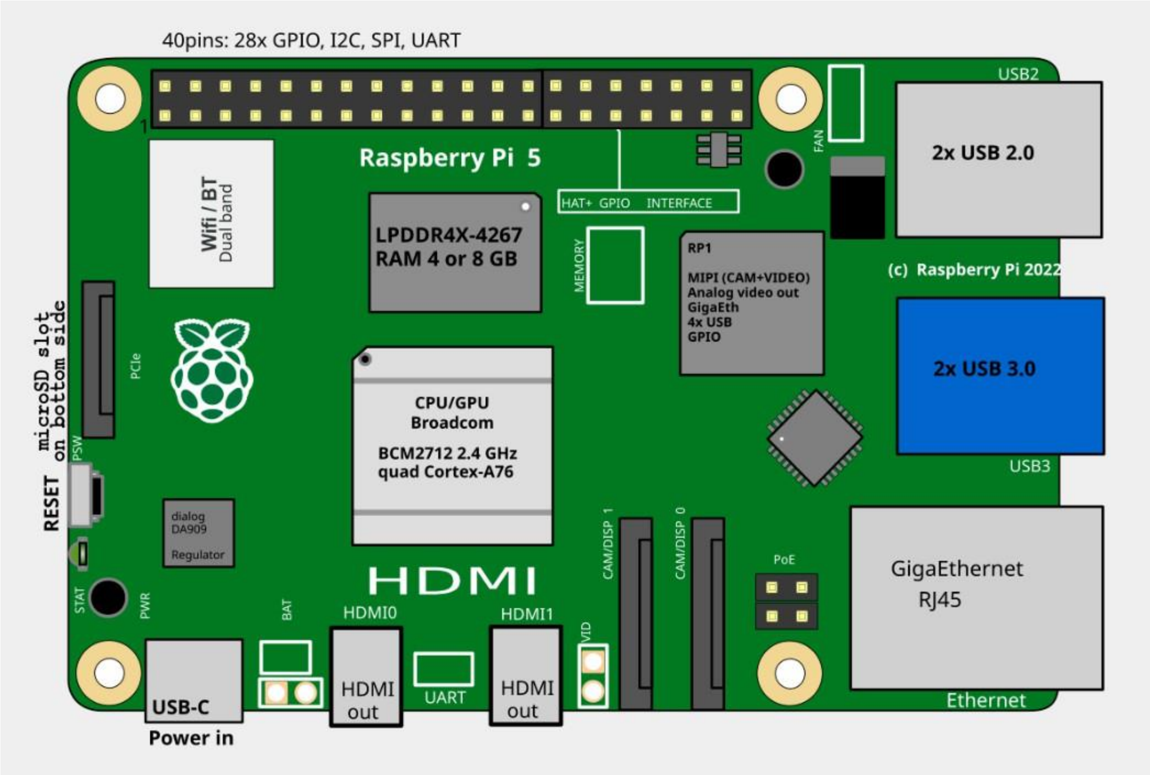


Fig 11 Raspberry Pi 5

APPENDIX E

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being an Apache 2 licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc.

OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups and by governmental bodies.

Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCV's deployed uses span the range from stitching streetview images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

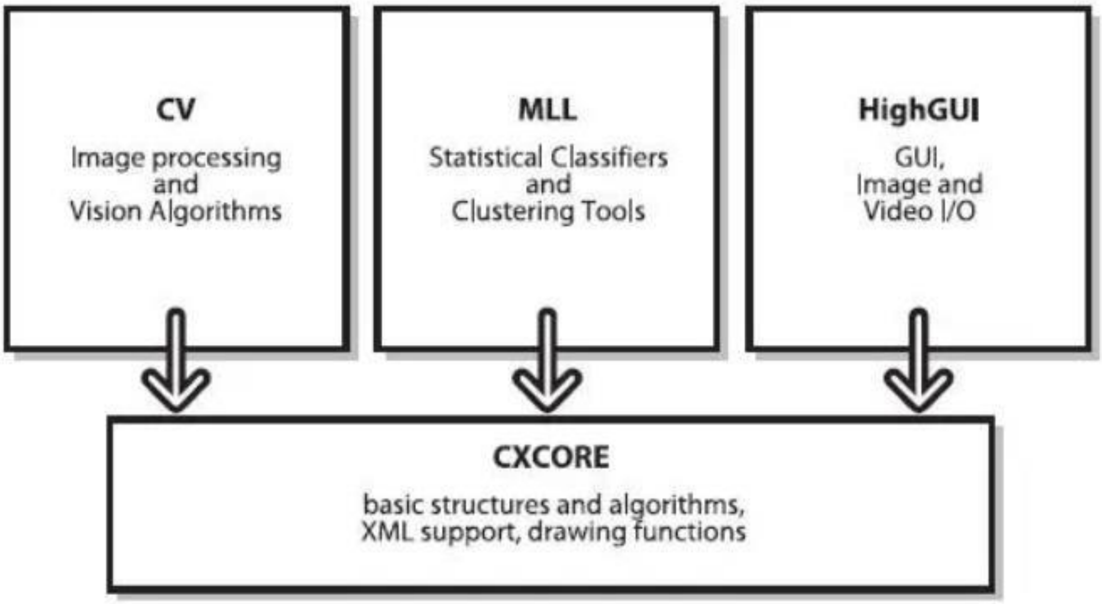


Fig 12 Open CV Architecture

Thonny IDE (Code Used)

```
import os
import math
import time

import numpy as np
import pygame
import busio
import board

from scipy.interpolate import griddata

from colour import Color

import adafruit_amg88xx

i2c_bus = busio.I2C(board.SCL, board.SDA)

# low range of the sensor (this will be blue on the screen)
```

```
MINTEMP = 26.0

# high range of the sensor (this will be red on the screen)
MAXTEMP = 32.0
# how many color values we can have
COLORDEPTH = 1024

os.putenv("SDL_FBDEV", "/dev/fb1")
# pylint: disable=no-member
pygame.init()
# pylint: enable=no-member

# initialize the sensor
sensor = adafruit_amg88xx.AMG88XX(i2c_bus)

# pylint: disable=invalid-slice-index
points = [(math.floor(ix / 8), (ix % 8)) for ix in range(0, 64)]
grid_x, grid_y = np.mgrid[0:7:32j, 0:7:32j]
# pylint: enable=invalid-slice-index

# sensor is an 8x8 grid so lets do a square
height = 240
width = 240

# the list of colors we can choose from
blue = Color("indigo")
colors = list(blue.range_to(Color("red"), COLORDEPTH))

# create the array of colors
colors = [(int(c.red * 255), int(c.green * 255), int(c.blue * 255)) for c in colors]

displayPixelWidth = width / 30
displayPixelHeight = height / 30

lcd = pygame.display.set_mode((width, height))

lcd.fill((255, 0, 0))

pygame.display.update()
pygame.mouse.set_visible(False)

lcd.fill((0, 0, 0))
pygame.display.update()


# some utility functions
def constrain(val, min_val, max_val):
    return min(max_val, max(min_val, val))
```

```
def map_value(x, in_min, in_max, out_min, out_max):
    return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min

# let the sensor initialize
time.sleep(0.1)

while True:
    # read the pixels
    pixels = []
    for row in sensor.pixels:
        pixels = pixels + row
    pixels = [map_value(p, MINTEMP, MAXTEMP, 0, COLORDEPTH - 1) for p in
pixels]

    # perform interpolation
    bicubic = griddata(points, pixels, (grid_x, grid_y), method="cubic")

    # draw everything
    for ix, row in enumerate(bicubic):
        for jx, pixel in enumerate(row):
            pygame.draw.rect(
                lcd,
                colors[constrain(int(pixel), 0, COLORDEPTH - 1)],
                (
                    displayPixelHeight * ix,
                    displayPixelWidth * jx,
                    displayPixelHeight,
                    displayPixelWidth,
                ),
            )

    pygame.display.update()
```