System Design for Mechtronics Enigeering

Team 32, Wingman, SmartVault
Edward He
Erping Zhang
Guangwei Tang
Peng Cui
Peihua Jin

January 18, 2023

1 Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

2 Reference Material

This section records information for easy reference.

2.1 Abbreviations and Acronyms

symbol	description
Mechtronics Enigeering	Explanation of program name
[... —SS]	[... —SS]

Contents

List of Tables

List of Figures

3 Purpose

This Document mainly talks about the design of the project, including the behavior, variables and interfaces used in the design. It will also talk about the design of the hardware component of the object, with some electrical components used and some communication protocols in the design.

4 Scope

The system will be designed to track the movement of the object to get the latest location information about it so that the user can always get the desired output. The user will be able to login and start the program through their own username and password. Then the information about the object will be detected through some image processing algorithms and will be stored into certain files. The user can locate desired objects through the searching interface by providing several searching keys.

4.1 Context Diagram

The following pictures shows the design of the context diagram of the project. In this diagram, the user can interact with the SmartVault by logging in and provide key information about the object and SmartVault will output searching results to the user. There will be a camera located in the room that will keep sending images to SmartVault used for image processing. The motor will interact with SmartVault so to change the angular position of the camera. SmartVault will send or update information stored in the database. It will also extract desired information from the database.

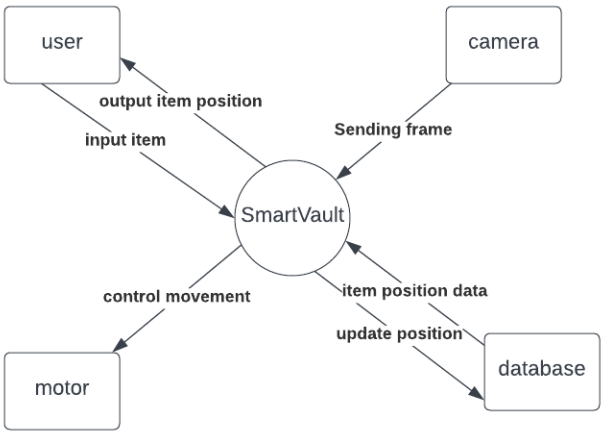


Figure 1: The Picture of Use Case Diagram

5 Project Overview

The purpose of this project is to create a system that help the user to locate their belongs in a certain space. SmartVault will allow user to login and start the camera, by detecting the movement of the object dynamically, it will keep recording and updating positional information about the objects that are detected in the camera. Then user can detect the position of the object by sorting through the database with certain searching information.

5.1 Normal Behaviour

SmartVault will start operating once the user login successfully and provide technical support when the user needs. It will store the position information of objects in the database for further use. The movement of the camera is controlled by the motor so that the monitoring angle will always in best angle. It will automatically detect the movement of object using image processing method. THIS LINE IS USED FOR OTHER MOCULE. TALKS ABOUT THE REST OF DEFINITION OF HARDWARE PART. To achieve different behaviors, different components are describes in the table below with their purpose.

Table 1: The Table of Division of Components and Purpose

Component Name	**Component Purpose**
Login	Manage login information and Technical Support Information
Database	Stores the position information of object detected inside the room
Image Process	Identifies movement of object and takes screen shot
Motor	Control the angle of the camera monitoring the room
Description	As part of the project hard requirement for Mechatronics group, the design can not be all software based

5.2 Undesired Event Handling

[How you will approach undesired events —SS]

5.3 Component Diagram

The picture shown below presents different components known as modules in this project. All the modules are divided into two big module group: Software Module and Hardware Module.

Figure 2: The Picture of Component Diagram

5.4 Connection Between Requirements and Design

To achieve the requirements mentioned in Software Requirement Specification Document, some specific designs are made and described in the table shown below.

Table 2: The Table of Connection Between Requirements and Design

Design Actions	**Requirements**
Image Processing Method	IPR1, IPR2, IPR3, IPR4,
Database	IPR5, IPR6, IPR7, IPR8, IPR9, RFR1, CAR1, LOR1
Customization Button	UIR1, UIR2
Vedio shown in the window	UIR4
Protection Cover	APR1, APR2
Text Prompt	EUR2, LER1, LER2, RFR2
Visualized Window	EUR2, UPR1, ACR1
Setting Username and Password	SCR2, AER1, INR1, INR2, PRR1, PRR2, CPR1
Technical Support Window	MAR1
Motor Communication Protocol	SCR3, RAR1

6 System Variables

[Include this section for Mechatronics projects —SS]

6.1 Monitored Variables

Table 1: Monitored Variables				
Monitor Name	Monitor Type	Range	Units	Comment
Detected object position	float	0-100	percent	This is the relative location of the detected object related to the frame image.
Detected object characteristic	String	TBD	TBD	This is the feature of the detected object
Detected object last seen time	time	Any Date	month-day-hour-min	This is the time in the database that a object was moved last time
Human activity and position in the camera frame	float	0-100	percent	This is the human detection position relative to the frame image

6.2 Controlled Variables

Table 2: Controlled Variables				
Controlled Name	Controlled Type	Range	Units	Comment
Degree of movement of the motor	float	0-360	Degree	This is the degree of the camera
Direction of movement of the motor	integer	0 or 1	NA	This is the direction of the movement of camera, 0 is left and 1 is right
Speed of rotation of motor	float	TBD	Degree/sec	This is the angular velocity of the camera

6.3 Constants Variables

Table 3: Constants Variables				
Constant Name	Constant Type	Value	Units	Comment
Angle per step	float	TBD	Degree/step	This is the angle movement stepper motor will move after 1 signal
Height of the Camera	float	TBD	mm	This is the distance between the lens of camera and the bottom of the mount
Resolution	Integer	1920x1080	Pixel	This is the resolution of the camera
Arduino input voltage	float	9.0	V	This is the input voltage of the Arduino board

7 User Interfaces

Two user interfaces will be used for this project, one is for the user to login and the other is used for searching the position information of the desired object. This section will mainly talks about these two interfaces in the following paragraphs.

7.1 Login Interface

The Login Interface is used to show let the user start the program. It will also provide the contact information of the technical support. The picture shown below describes the FSM of the Login Interface. When the user starts the program ans enters correct username and password, the original window quits and comes up with the Search Window. If the user enters wrong username or passsword, the wondown will not change and asks the user to retry. If the user want to get the technical support, the Technical Support will come up.

Figure 3: The Design of Welcome Window

WHen it comes to the Design of Window, the first one is the Welcome Window. The Welcome Window asks the user to input the username and password. If the user enters correct username and password, the window is closed and the Confirmation Windown will come up, which will be described in the paragraphs below. The Technical Support Window

gives the emails of each team member. The Design of Welcome and Technical Support Window is shown in the pictures below.

Figure 4: The Design of Welcome Window

Figure 5: The Design of Technical Support Window

7.2 Searching Interface

The Searching Interface is used to help the user to locate the position of the object. The picture shown below describe the FSM of the project. After the program starts, the image processing method will be used to record initial condition of the object detected in the roon through the image taken by the camera. The program will wait for further changes. The motor will rotate the camera if the user detected is not in the certer of the camera or certain percentage of area of the images is blocked. When the movement of an object is detected, if it is moved by human, the program will track the movement of hands and update the information stored in the database. If it is moved by other objects, the program will only update it final state. The data base willl only record the object that moves in the area. When the user want to search certain object, the program will allow the user to input some information about the object like the approximate time. Then a list of pictures meets the information will be provided and wait for the user to choose. If the desired object is within the list and the user has confirmed it, the algorithm will finish. If the object is not found, the initial pictures will be pulled out and let the user to choose. The object is marked "Taken out" if the object is still not found.

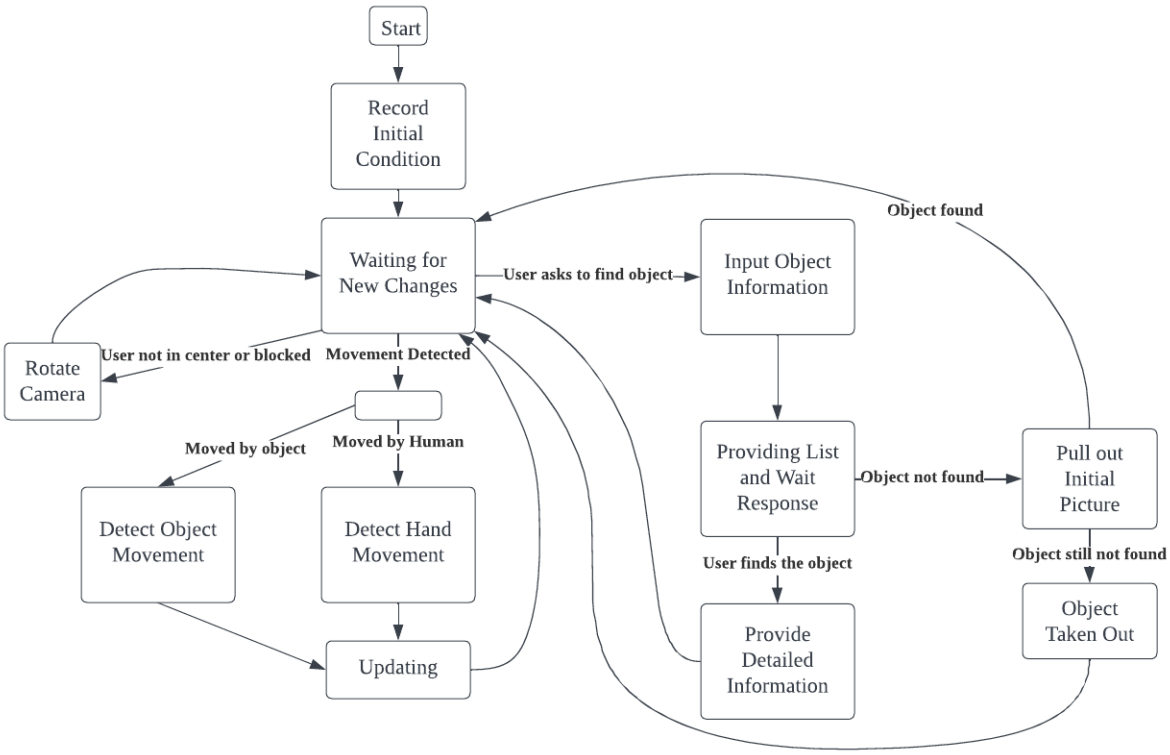


Figure 6: The Finite State Machine of Searching Interface

For the searching interface of this project, the window will come up after the use has successfully logging into the system. A simple design of that interface is shown in the figure below. On the left hand side the images taken by the camera will be shown. On the right hand side, the object-searching algorithm will be used. If the user want to search for one desired object, the system will ask the user to input several informations about that object. After user has finished entering the information and press search, a new window will appear. It will provide several pictures that meets the input information. After the user has confirmed the final result, the result window will come up with the information that the user needs about the object. If the user press "Object not Found ?" button, another confirmation window with the same payttern will come up but with pictures that present the initial condition. An error window is designed to tell the user that the object may be taken out by human. It will aso leave a button to return to the Searching Window.

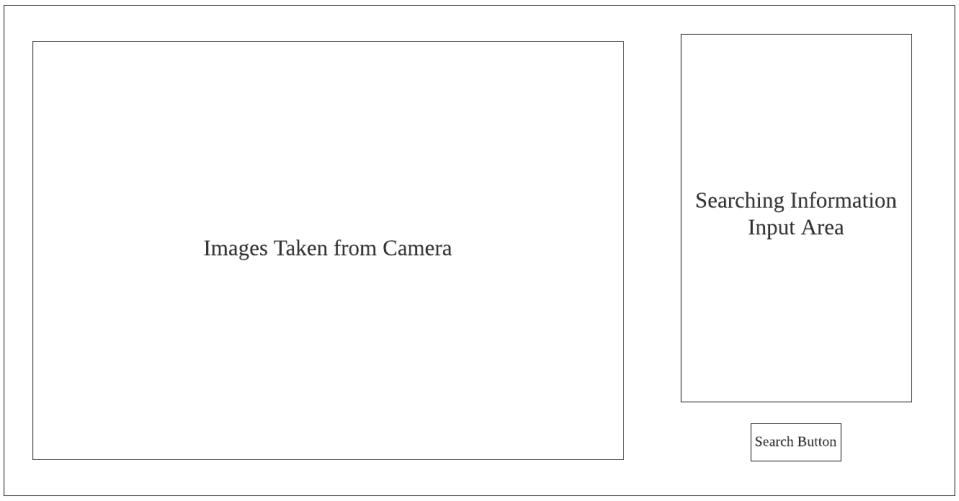


Figure 7: The Design of Searching Window

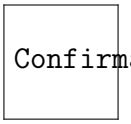


Figure 8: The Design of Confirmation Window

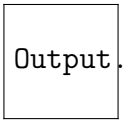


Figure 9: The Design of Output Window

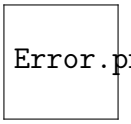


Figure 10: The Design of Error Window

8 Design of Hardware

The hardware part consists of motor, Arduino contoller board, camera and mount.Arduino controller board is used to communicate with PC and driven by the functionality input

by the user. Two stepper motors will be used for horizontal motion and vertical motion respectively. Camera is directly connected to PC to display the video in the user interface. Mount is designed to meet the requirement which can fix both the camera and the motors and capable of three-dimension movement. Further details will be described in below subsections.

8.1 Stepper Motors

Stepper motors are directly connected to the Arduino controller board to receive corresponding signal and rotate to the assigned position. In this section, its physical dimensions will be mainly described. The dimension figure is shown below.

Figure 11: Motor figure

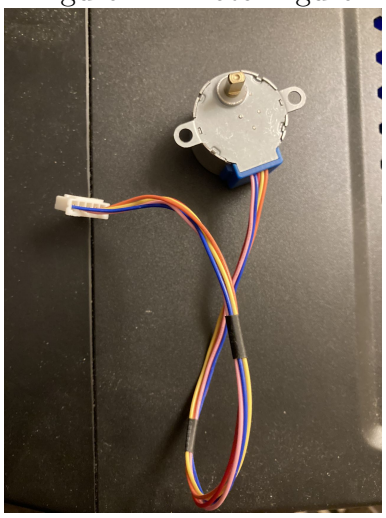
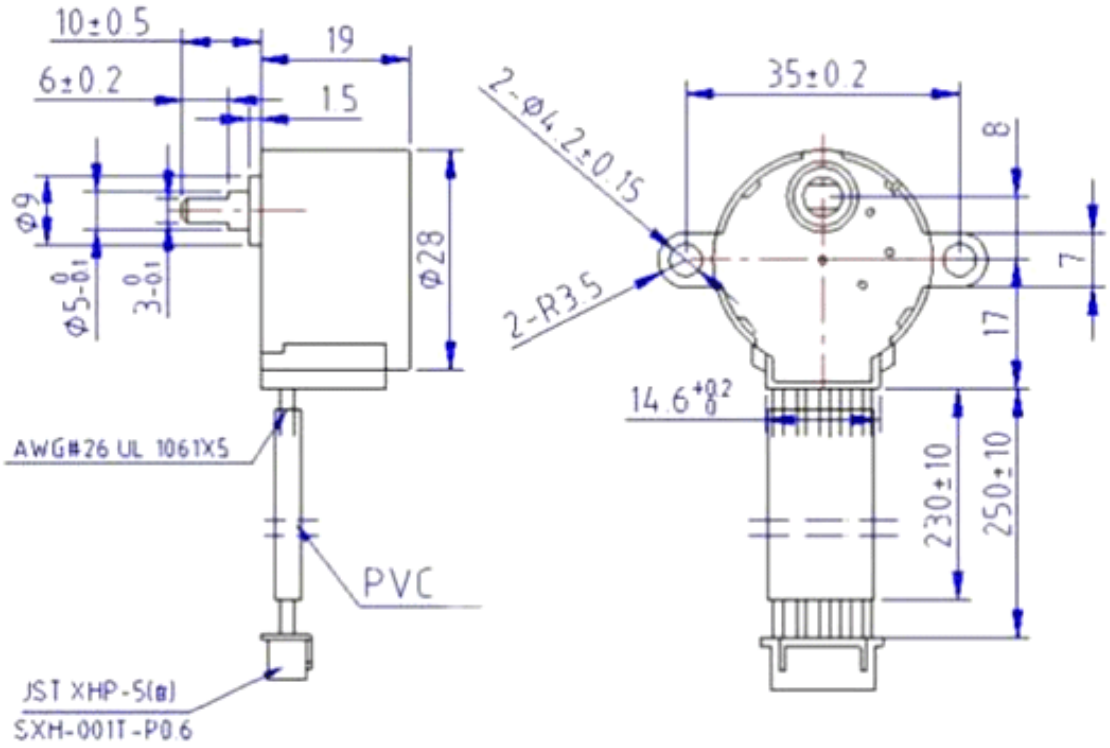


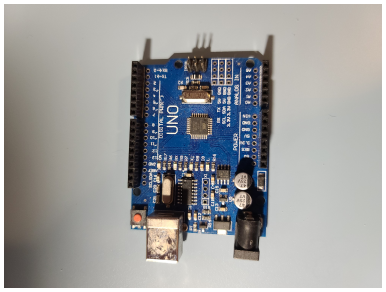
Figure 12: Motor dimension



8.2 Arduino controller board

Arduino controller board is connected to PC and the stepper motors.It is responsible for sending signals to the stepper motors in order to move the camera to desired angle. The board is shown below. As it is not in the moving part of the hardware system, its dimensions are not taken into consideration for initial design.

Figure 13: Arduino controller board



8.3 Camera

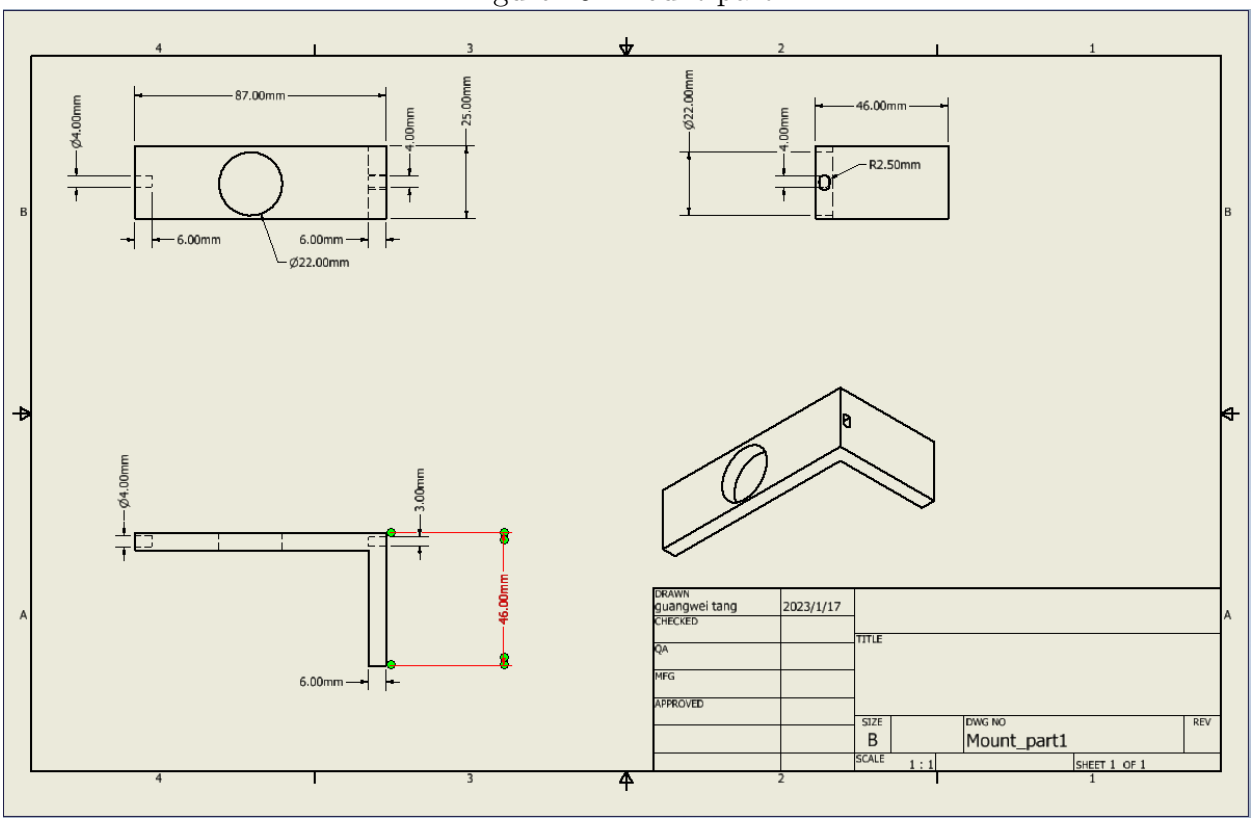
Camera is connected to the PC for video display and fixed to mounting bracket in order to be driven by the stepper motors. The camera is shown below. Under our design and cost consideration, the protruding camera part's radius is taken into consideration.

Figure 14: Camera

8.4 Mounting bracket

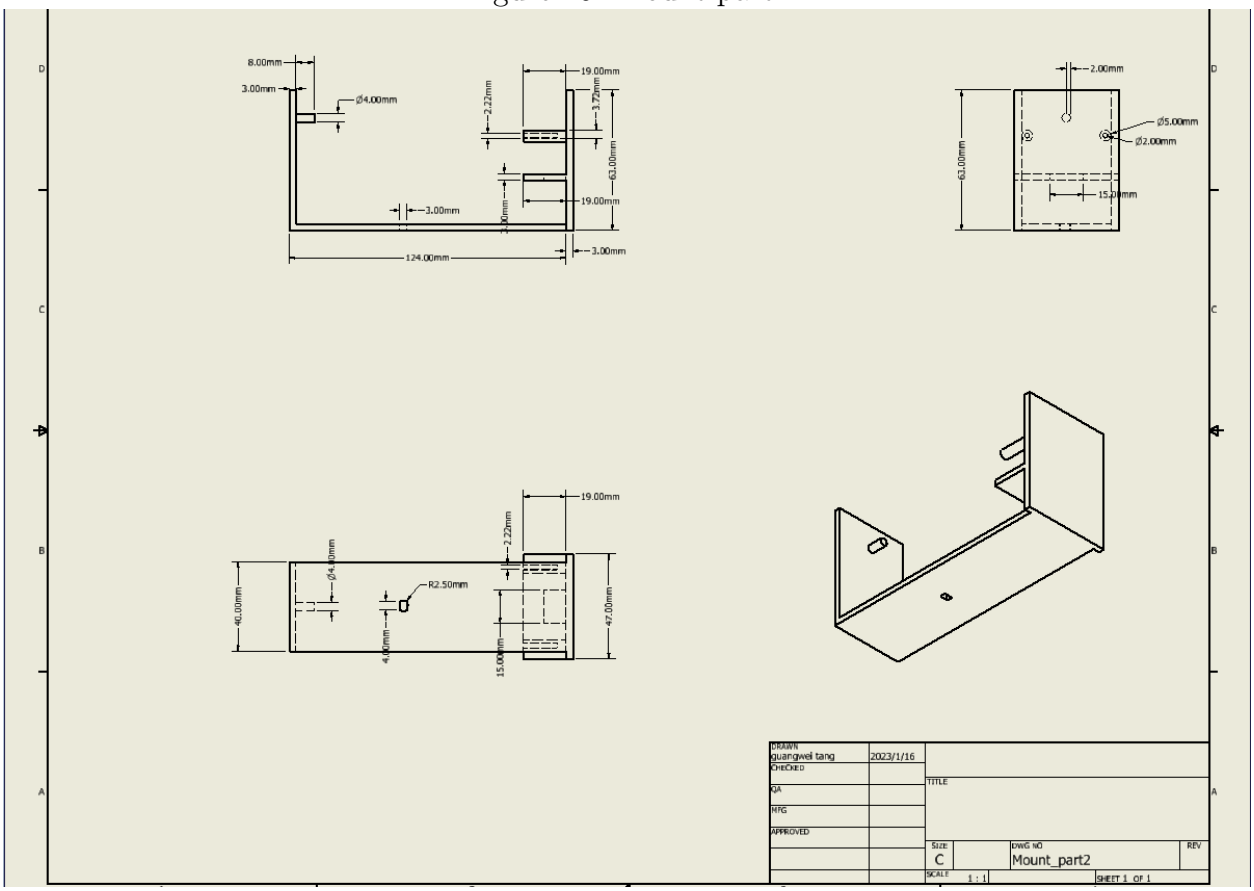
The mounting bracket is designed to meet the design requirement. It consists of two parts. First part is used to fix the camera and allow vertical motion. One hole is constructed on one side of the wall to fit the stepper motor's actuator. The dimensions can be found in the figure below.

Figure 15: Mount-part1



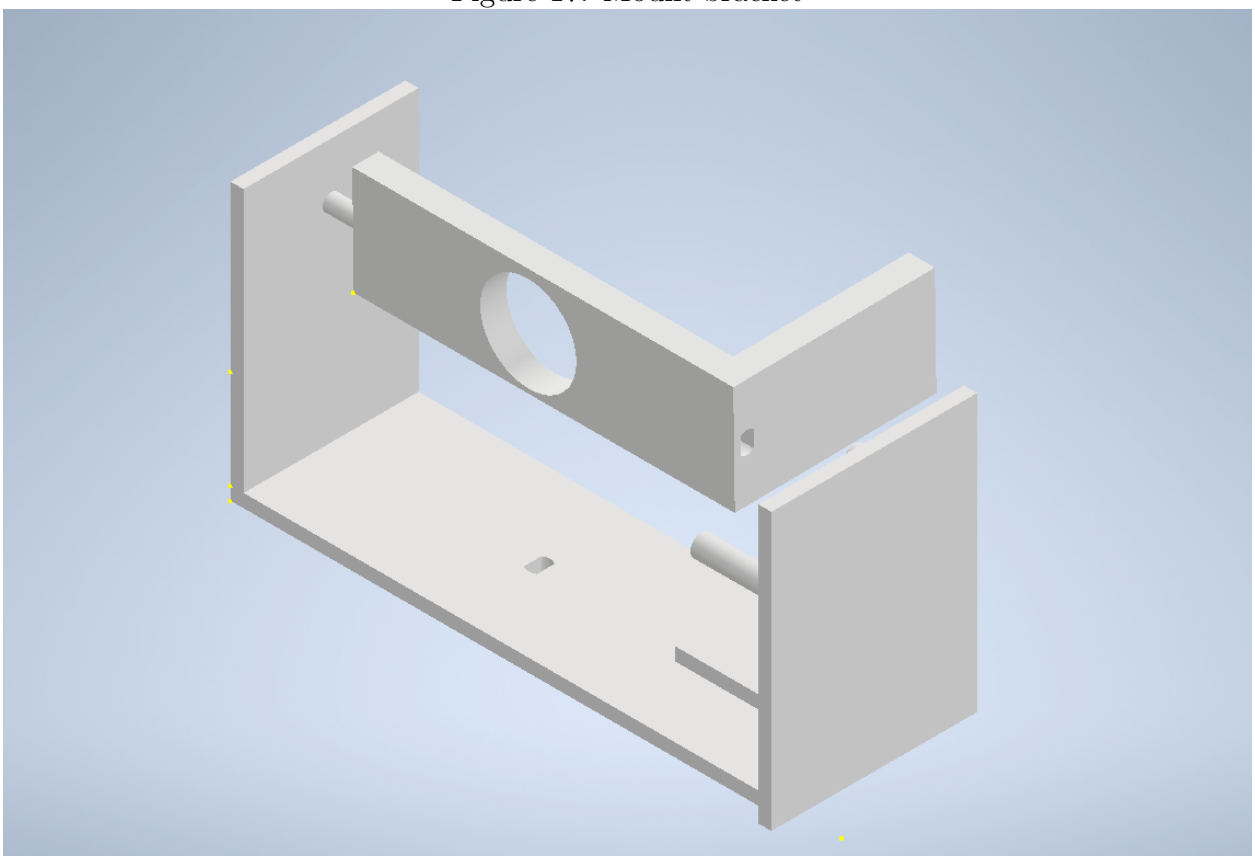
Second part is used to fix the first part and driven by the second stepper motor to allow horizontal motion.One hole is constructed at the bottom to fit the stepper motor's actuator. The dimensions can be found in the figure below.

Figure 16: Mount-part2



The whole configuration of the mounting bracket is shown below.

Figure 17: Mount bracket



9 Design of Electrical Components

The electrical components of the system include the Arduino controller board,stepper motor driver board, camera and stepper motors. The electrical schema is shown below:

Figure 18: Board-motor connection

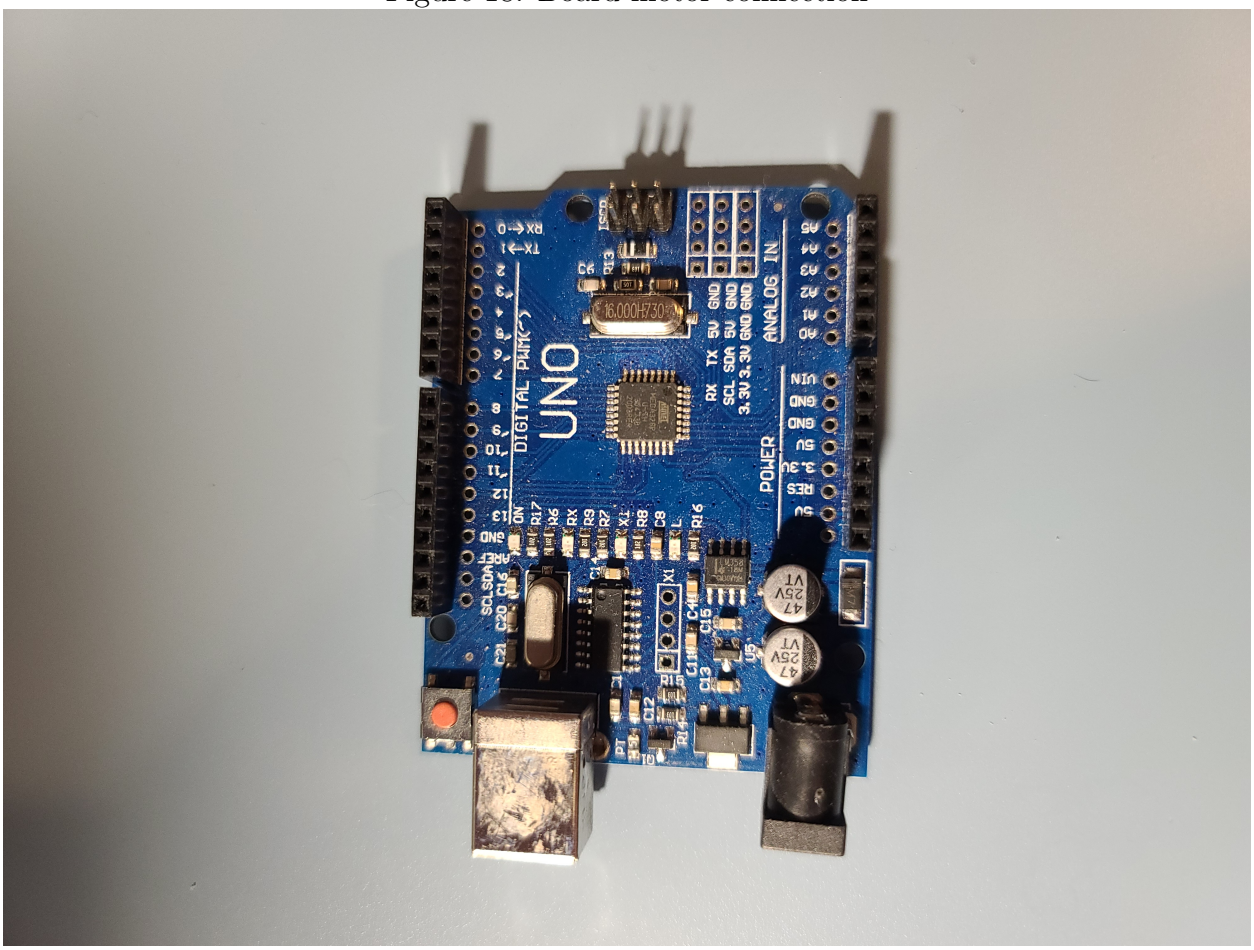
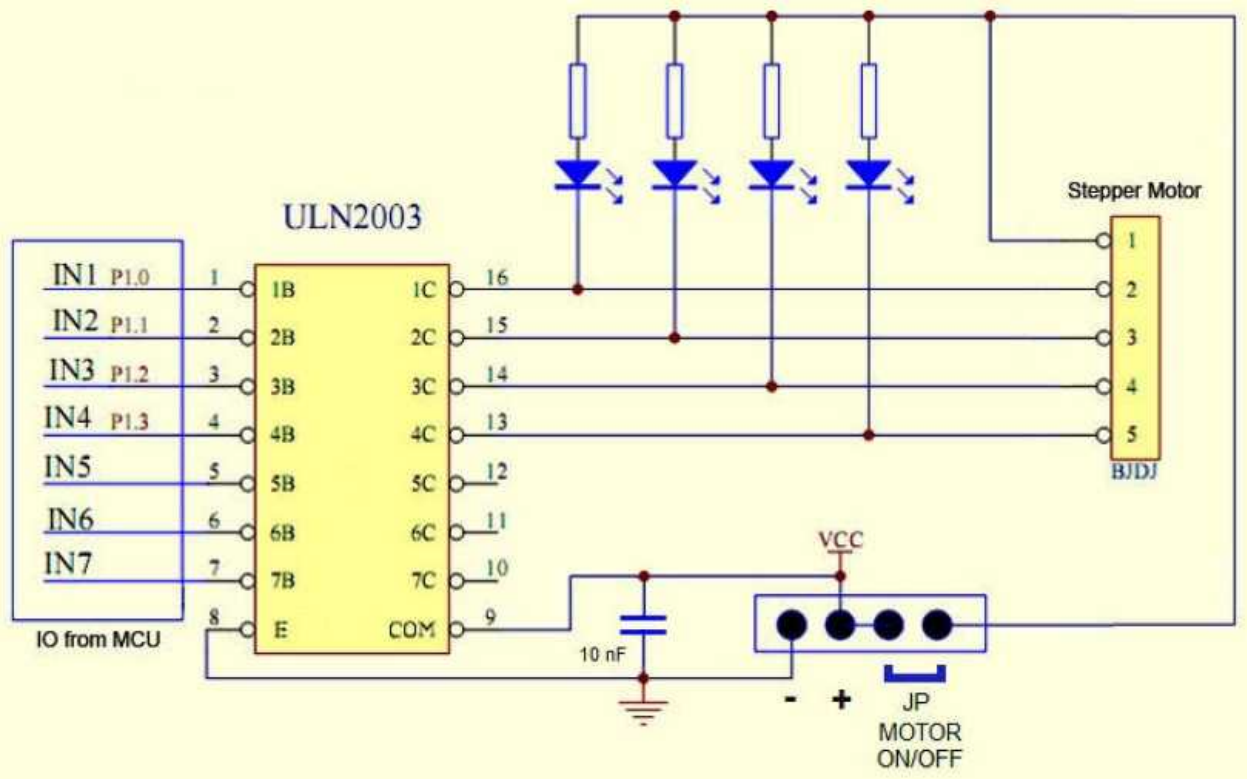


Figure 19: Stepper motor driver board schematic



10 Design of Communication Protocols

The communication protocol between the hardware and software is serial communication protocol. The software system acts as the master and the hardware acts as slave. The communication is a one direction communication, the software system sends the serial command to the Arduino controller board, and the board control the stepper following the serial commands.

11 Timeline

[Schedule of tasks and who is responsible —SS]

A Interface

[Include additional information related to the appearance of, and interaction with, the user interface —SS]

B Mechanical Hardware

Mechanical hardware consists of two mounting brackets. First part fix the position of the camera and connect with the stepper motor who controls vertical motion. Second part is connected with the stepper motor who controls horizontal motion. Part one is fixed to the part two by a rod. The horizontal-motion stepper motor act as the base of the system. The mounting bracket will be 3D-printed with PLA to minimize the weight.

C Electrical Components

The electrical components include 1 Arduino board, 2 stepper motors, 2 ULN2003 drivers and 1 camera. The motor driver simplify the control process of the stepper motor and save the physical space of the hardware. The Arduino board send signal to control the stepper motor.

D Communication Protocols

The serial communication protocol in the project guarantee the communication between software and hardware. The project uses python serial library to send the serial signal to the Arduino board. The communication is one-way direction which means the Arduino only receive the signal from software system but not send any signal back to the system.

E Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Problem Analysis and Design. Please answer the following questions:

1. What are the limitations of your solution? Put another way, given unlimited resources, what could you do to make the project better? (LO_ProbSolutions)

2. Give a brief overview of other design solutions you considered. What are the benefits and tradeoffs of those other designs compared with the chosen design? From all the potential options, why did you select documented design? (LO_Explores)