EZBackUp
Team 20

Justin Charles, Justin Lam, Zhiyang Jin, Exiang Zhou, Austin Carroll

University of Massachusetts Amherst BE REVOLUTIONARY™

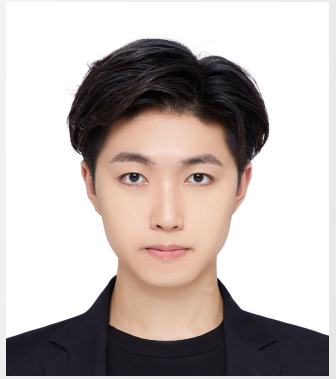
Team 20:

Christopher V. Hollot
Faculty Advisor

Justin Charles
Computer Engineer

Austin Carroll
Mechanical Engineer

Zhiyang Jin
Electrical Engineer



Exiang Zhou
Computer Engineer

Team Roles

Justin Charles

Logistics lead:

1. Sensor and Camera hardware/software design

2. Communicate with team and course coordinators

Zhiyang Jin

PCB lead:

1.Breadboard design and PCB design.

Exiang Zhou

Software Lead:

1.WIFI establishment and data transfer with cameras

2.Mobile software design and development

Austin Carroll

Mechanical Lead:

1.Camera Mount

2.Positioning system

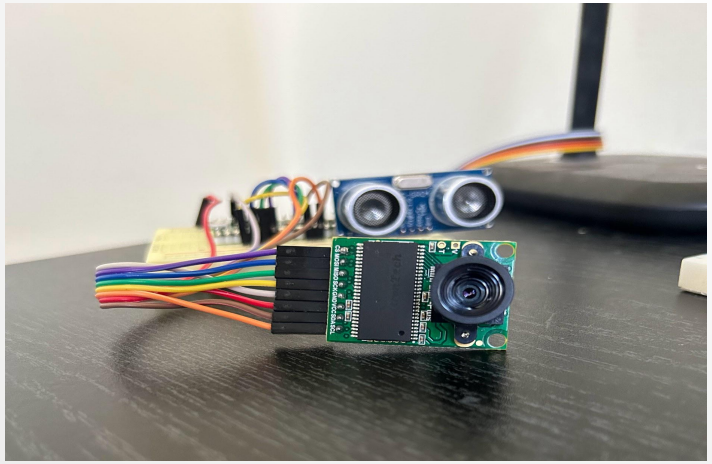
3. 3-D printing

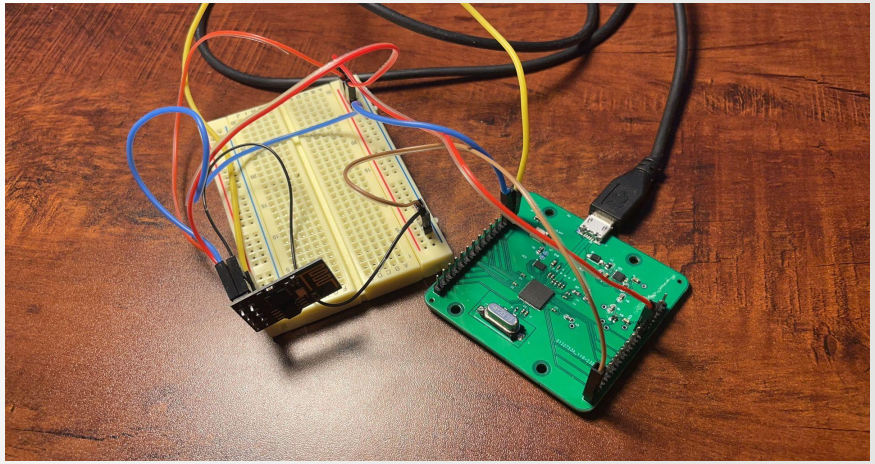
Project Goal Review

Problem: Driving is a task that many Americans undertake daily, it is a necessary function of human life today and will continue to be prevalent into the future. While this may be true, there will always be a need for added safety.

Goal: We aim to create a backup camera system that is easy and accessible to install and use, while also being convenient and adding safety for the users driving experience.

Prototype:





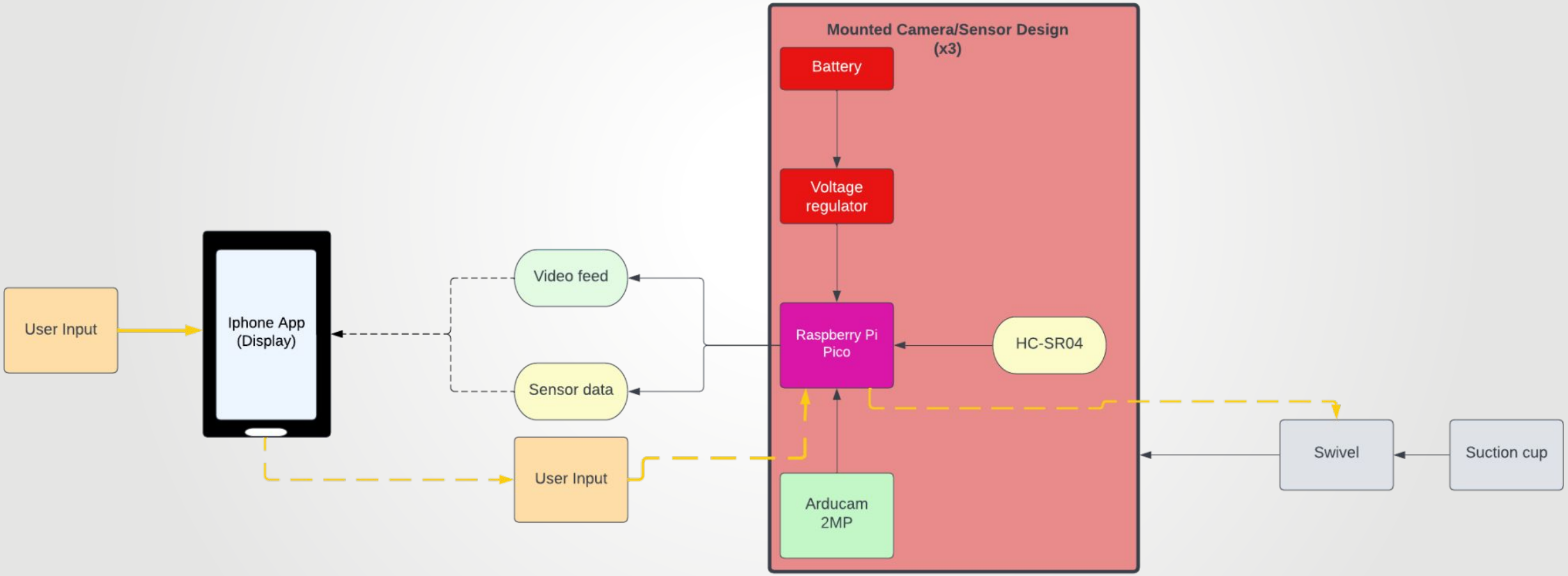
Specifications and Testing - Qualitative

System Specification	Test Plan
System will use up to 3 video systems wirelessly connected to smartphone	Inspect that up to 3 video feeds/sensors/motors will work on display
System display will show up to 3 video feeds, one feed may be chosen at a time	Inspect app to check that video feeds can be viewed and changed by user
System will provide distance and audio to the user via smartphone	Inspect app to check that distance and audio is output
Camera will have low light capabilities	Inspect video feed at night and check that objects are visible
Camera Systems will be self powered	inspect that the system will work being self powered
Camera/Sensor system will be mountable and dismountable to vehicle	Mount system on vehicle and test by driving
System will be easy to set up	Survey 10 people with setup, ask to rate on a scale of 1-10 complexity of set up. <3 should be chosen.

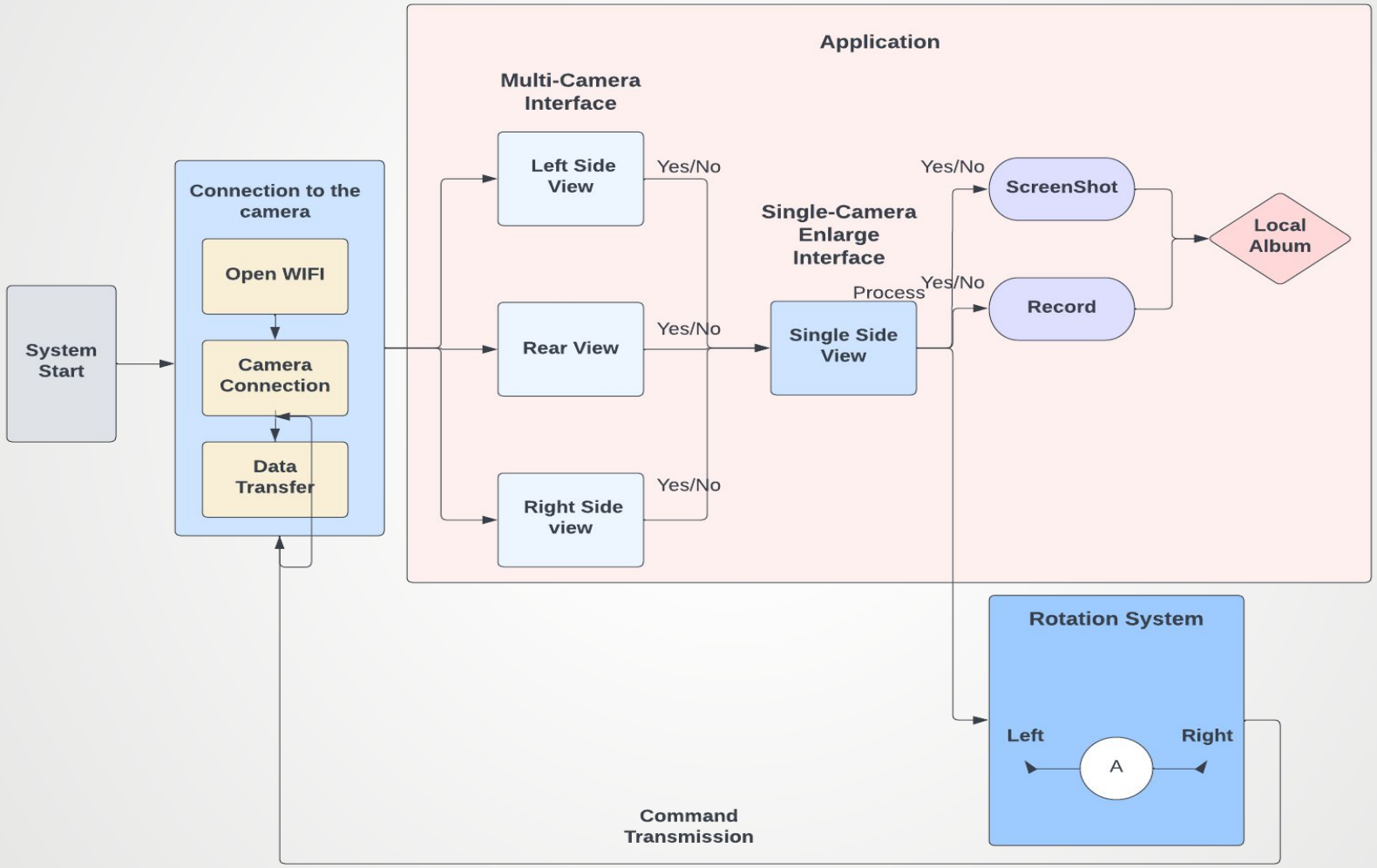
Specifications and Testing - Quantitative

System Specification	Test Plan
System will give a slow audio alert when ≤2ft and an increasingly faster alert when ≤1ft	Get distance using sensors, manually measure distance
System cameras will rotate in intervals of 30 degrees on a horizontal axis	Measure angle change when given rotation input for cameras
Individual camera systems will hold power for a total of 8 hours	measure change in power over a day to estimate power loss
Video Feed will have a frame rate of at least 30 fps	Measure the frames per second of the camera feed after data transfer using external software
System distance Sensors can detect objects at least 10m away	Get distance using sensors, manually measure distance
System minimum resolution will be 160 x 120 pixels	Measure the resolution of the camera feed after data transfer using external software

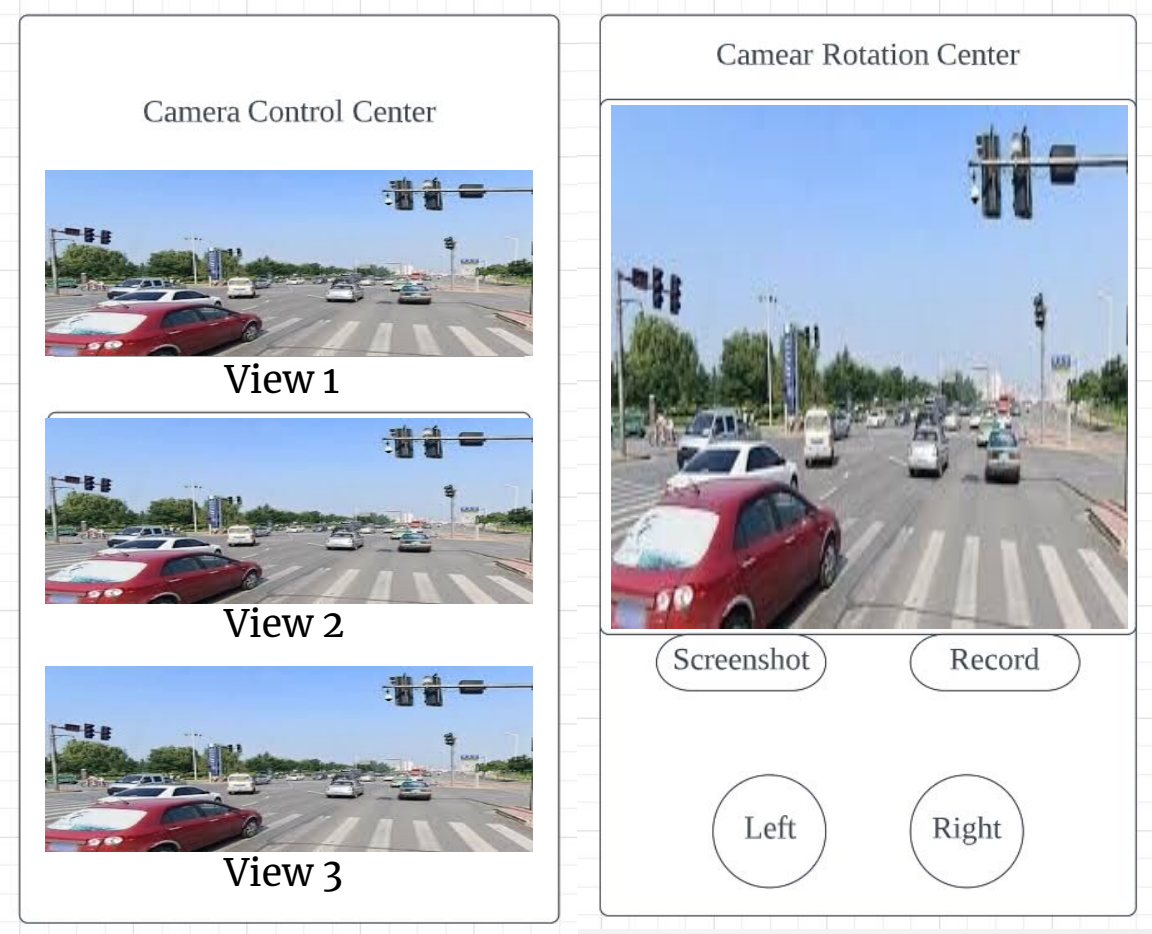
Hardware Block Diagram



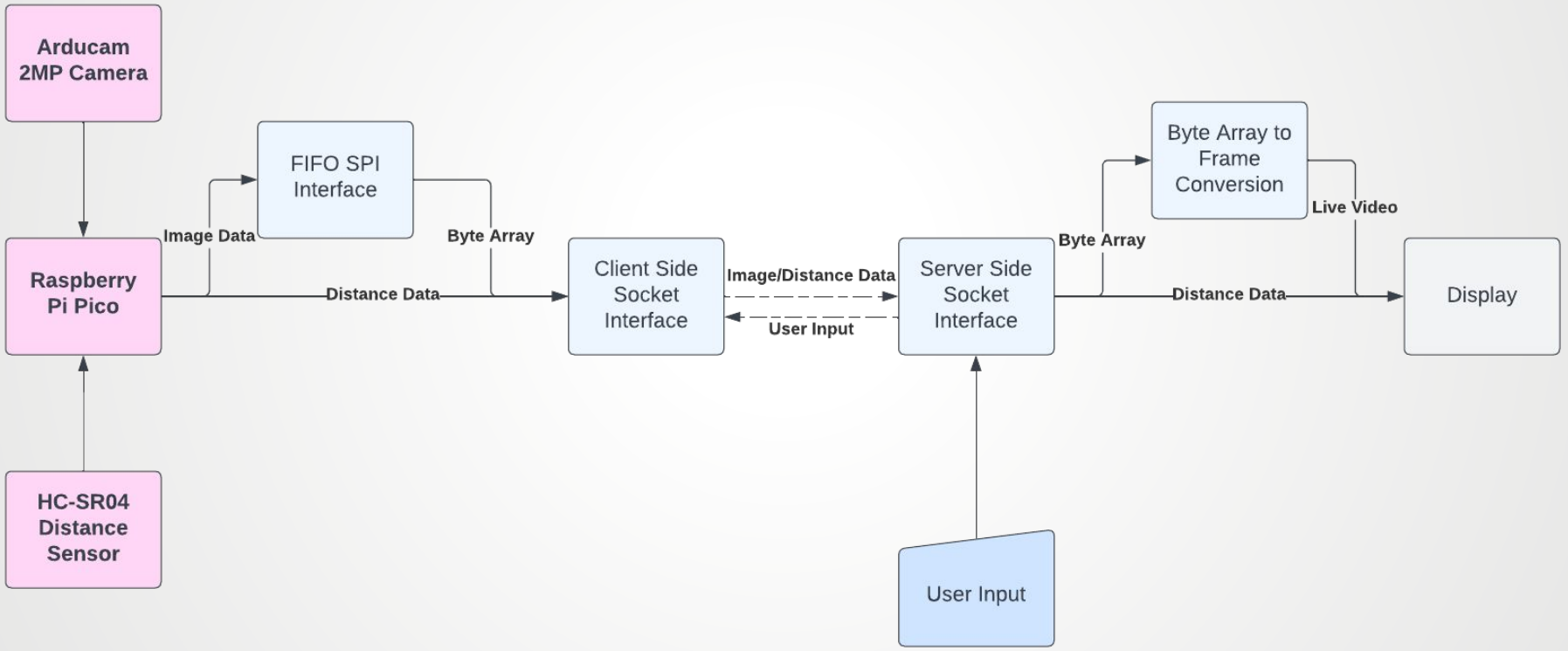
Software Diagram



Multi-cameras Interface Sketch (APP)



Block Diagram – Camera/Sensor System



https://learn.adafruit.com/ultrasonic-sonar-distance-sensors/python-circuitpython

https://manuals.plus/arducam/ov2640-mini-2mp-spi-camera-on-raspberry-pi-pico-manual#axzz7mzPDUXjx

BE REVOLUTIONARY™

CDR Deliverables

University of Massachusetts Amherst

CDR Deliverables

- ❑ **Mounting System**
  - ✅ Wireless Casing for all components
  - ✅ Casings capable of panning in 45 degree segments
- ❑ **Data Transfer**
  - ✅ Enables wireless real-time video transmission between cameras and external devices
  - ✅ wirelessly transfer distance data to external devices
- ❑ **PCB**
  - ✅ A working RP2040 minimal design
  - ✅ Add Wifi chip to the RP2040 minimal design
- ❑ **Mobile App**
  - ❌ Prototype app that receives visual feed and distance data

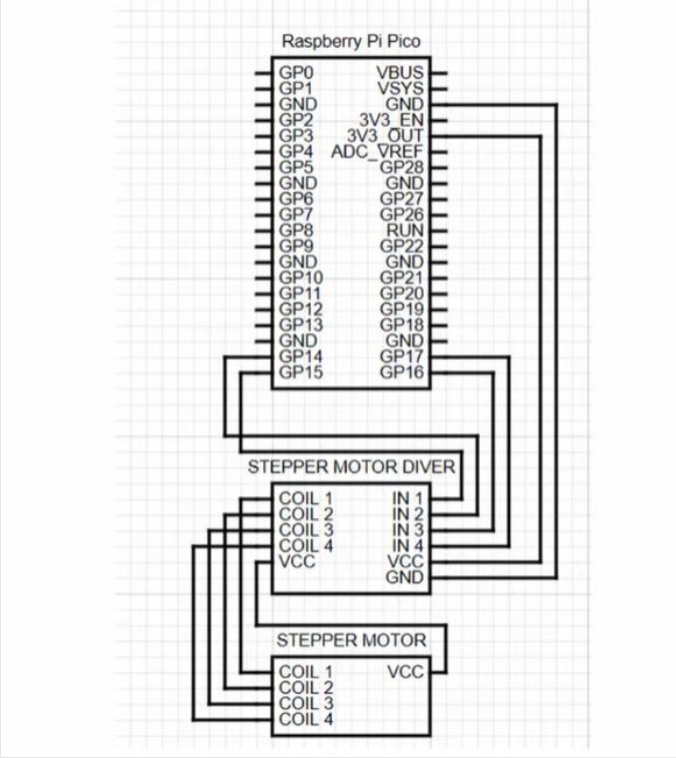
Hardware Used and Power Considerations

- Camera Subsystem: Total ~ 0.89W/Hr
  - Raspberry Pi Pico W - 5V/93mA → 0.465W
  - HC-SR04 - 5V/15mA → 0.075W
  - Arducam 2MP 5V/70mA → 0.35W

- Motor Subsystem
  - 24BJY-48 Stepper Motor - 5V
  - ULN2003 Stepper Motor Driver - 5V
  - Raspberry Pi Pico W - 3.3 V
  - 3 x 6V Battery Pack

Motor Subsystem

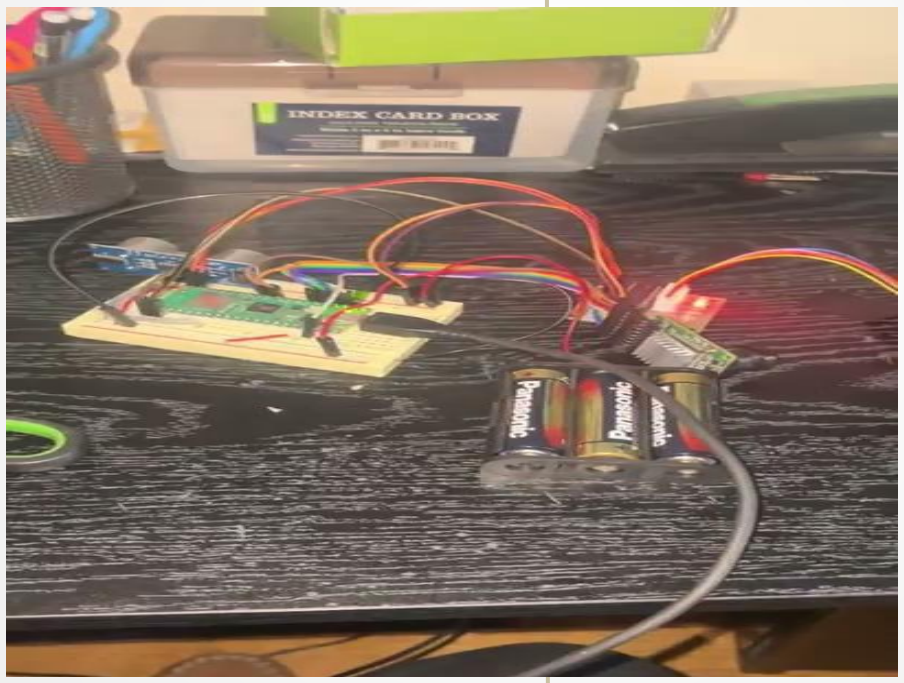
Schematic



Code

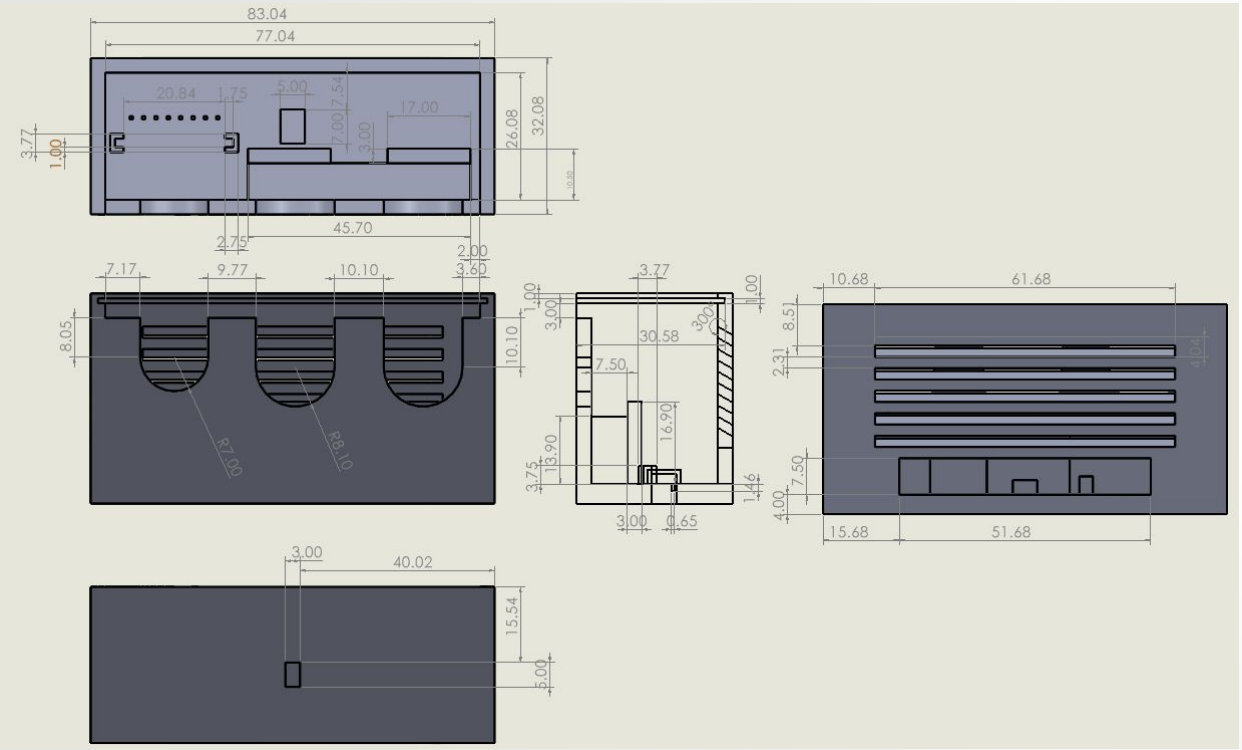
```python
import time
import board
import digitalio

IN1 = digitalio.DigitalInOut(board.GP15)
IN2 = digitalio.DigitalInOut(board.GP14)
IN3 = digitalio.DigitalInOut(board.GP16)
IN4 = digitalio.DigitalInOut(board.GP17)


half_step_sequence = [
    [1, 0, 0, 0],
    [1, 1, 0, 0],
    [0, 1, 0, 0],
    [0, 1, 1, 0],
    [0, 0, 1, 0],
    [0, 0, 1, 1],
    [0, 0, 0, 1],
    [1, 0, 0, 1]
]

STEP_DELAY = 0.002


reference_angle = 0
current_angle = 0


def step(direction):
    global current_angle
    IN1.direction = digitalio.Direction.OUTPUT
    IN2.direction = digitalio.Direction.OUTPUT
    IN3.direction = digitalio.Direction.OUTPUT
    IN4.direction = digitalio.Direction.OUTPUT
    if direction == "clockwise":
        for step in range(170):
            for pin in range(2):
                setattr(IN1, "value", half_step_sequence[step%8][0])
                setattr(IN2, "value", half_step_sequence[step%8][1])
                setattr(IN3, "value", half_step_sequence[step%8][2])
                setattr(IN4, "value", half_step_sequence[step%8][3])
                time.sleep(STEP_DELAY)
            current_angle += 30
            #current_angle %= 360
    elif direction == "counterclockwise":
        for step in reversed(range(170)):
            for pin in range(2):
                setattr(IN1, "value", half_step_sequence[step%8][0])
                setattr(IN2, "value", half_step_sequence[step%8][1])
                setattr(IN3, "value", half_step_sequence[step%8][2])
                setattr(IN4, "value", half_step_sequence[step%8][3])
                time.sleep(STEP_DELAY)
            current_angle -= 30
            #current_angle %= 360

ecount = 0
qcount = 0
while True:
    user_input = input("Enter 'w' to return to reference angle, 'e' for clockwise or 'q' for counterclockwise: ")
    if user_input == "w":
        count = ecount + qcount
        if count < 0:
            for i in range(abs(count)):
                step("clockwise")
        else:
            for i in range(count):
                step("counterclockwise")
        count = 0
        ecount = 0
        qcount = 0
    elif user_input == "e":
        step("clockwise")
        ecount = ecount + 1
    elif user_input == "q":
        step("counterclockwise")
        qcount = qcount - 1
```

Motor Demo

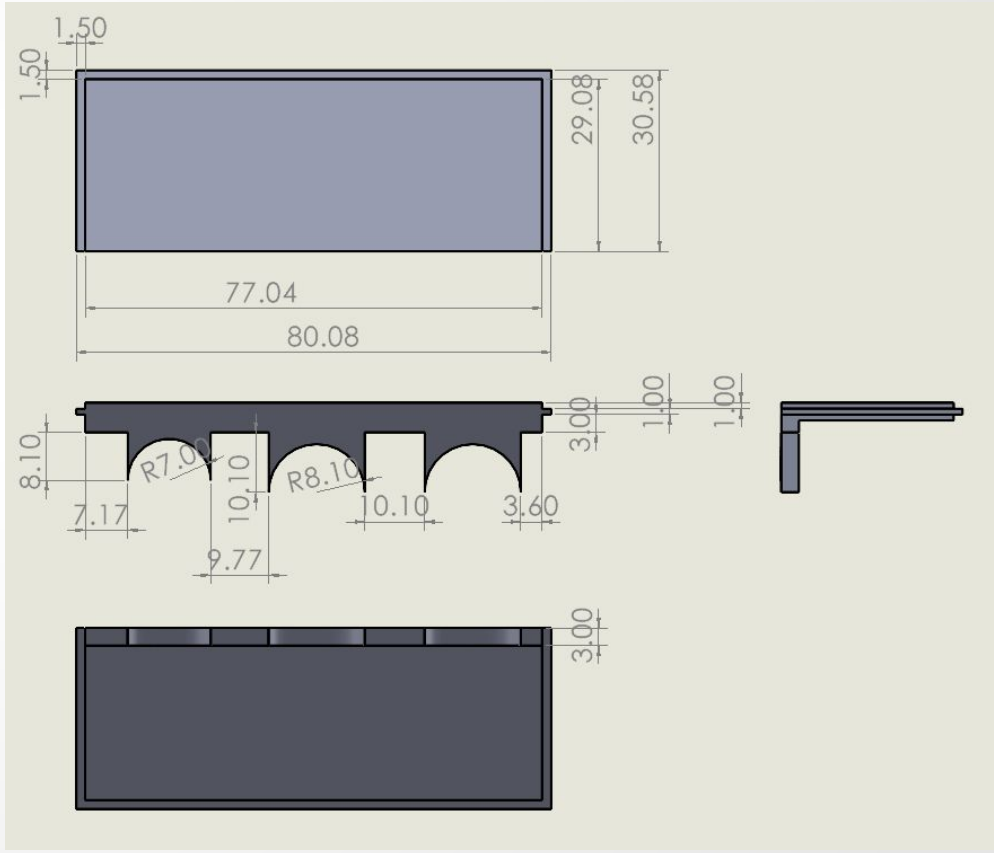


Casing Model

Camera and Distance Sensor Housing

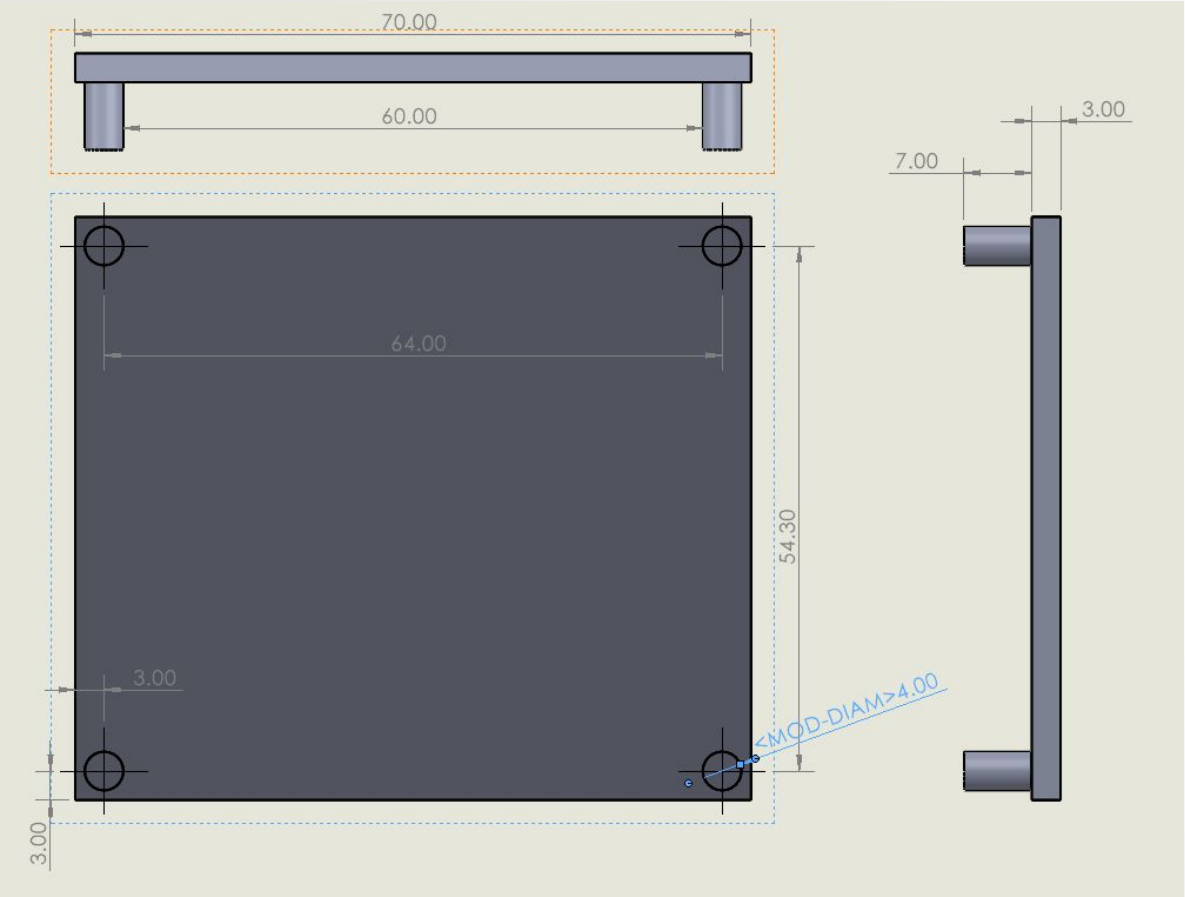


Camera and Distance Sensor Housing Cover

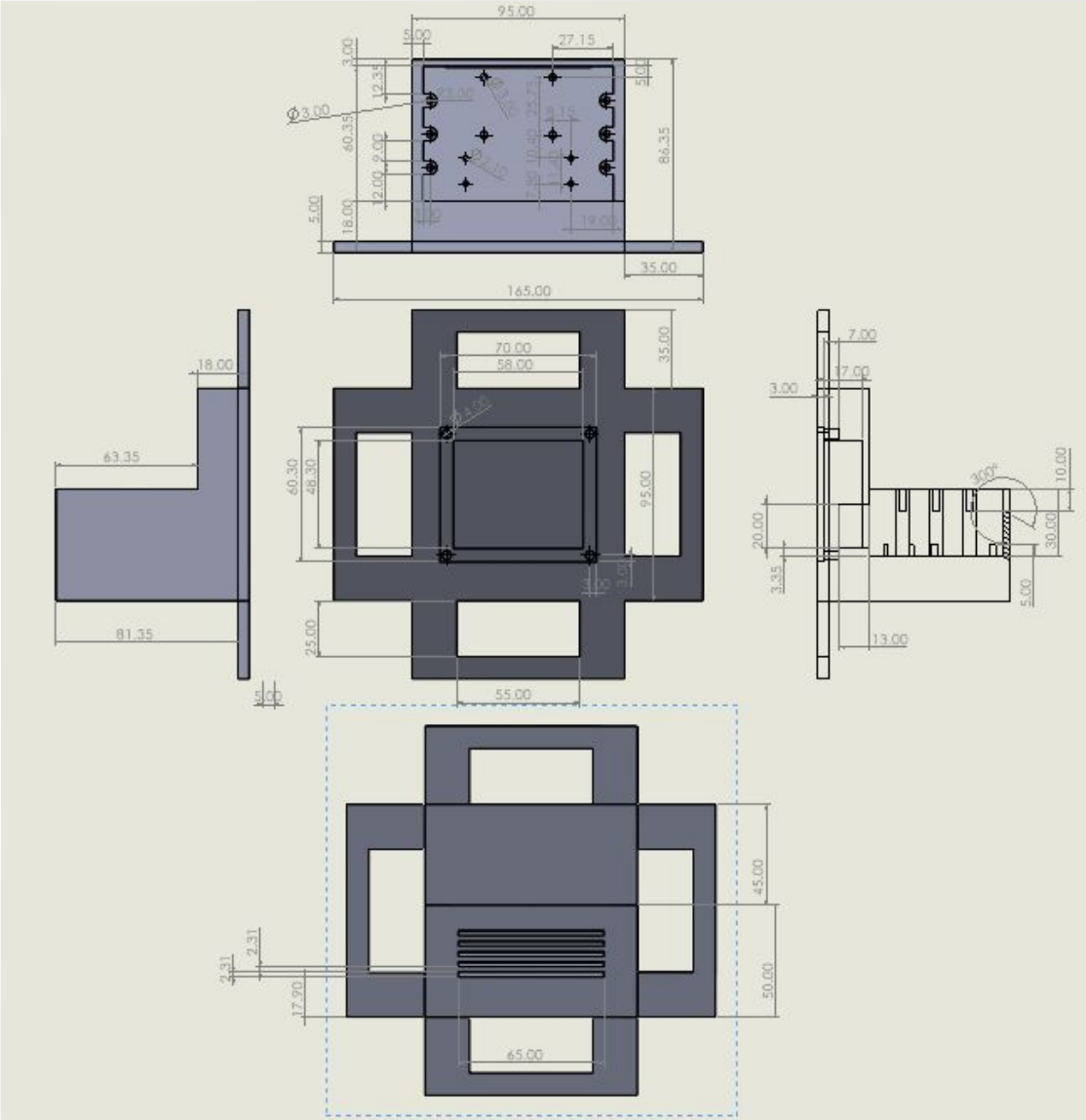


Casing Model Cont.

Battery Cover

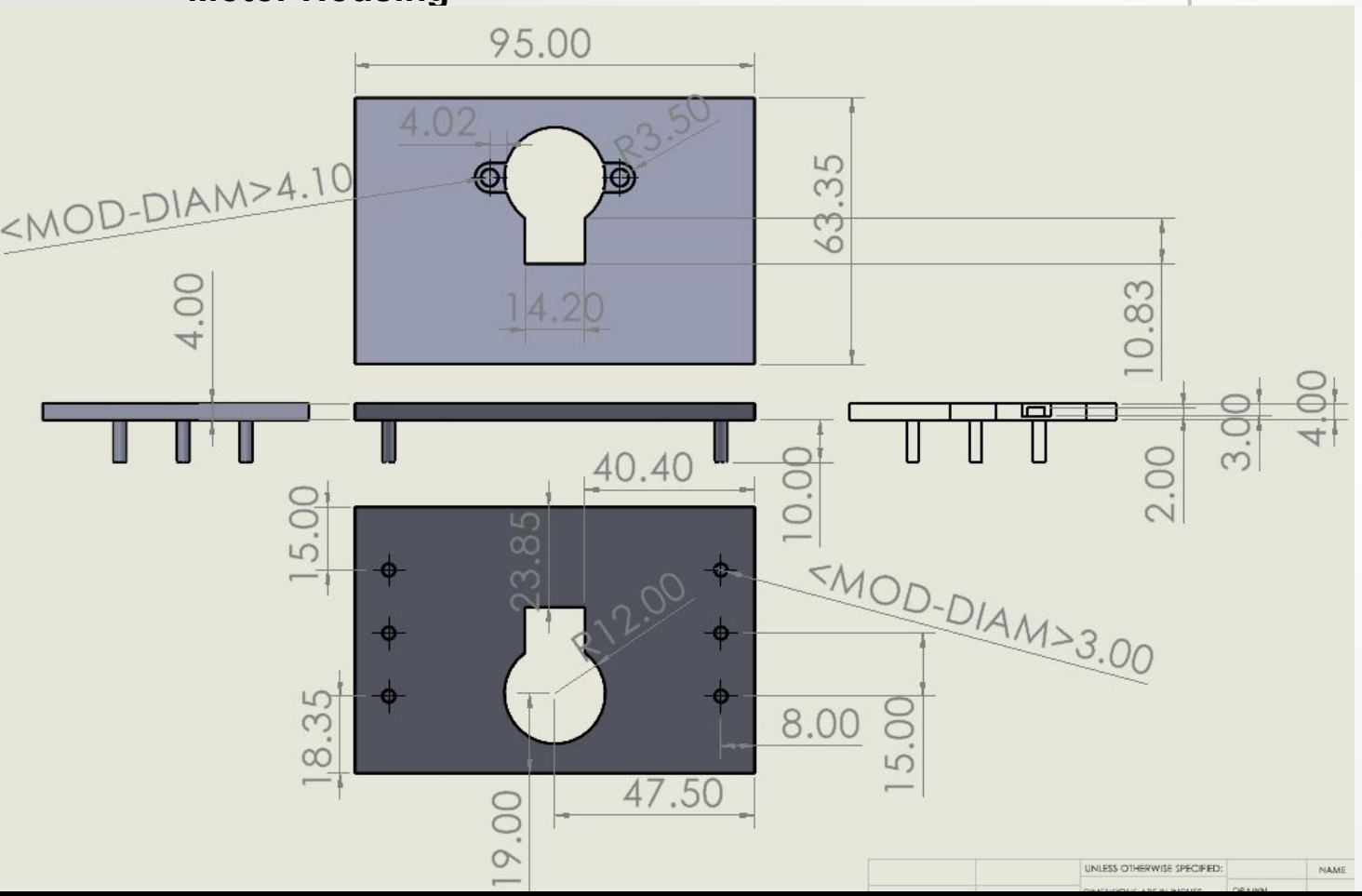


Pico and Driver Housing

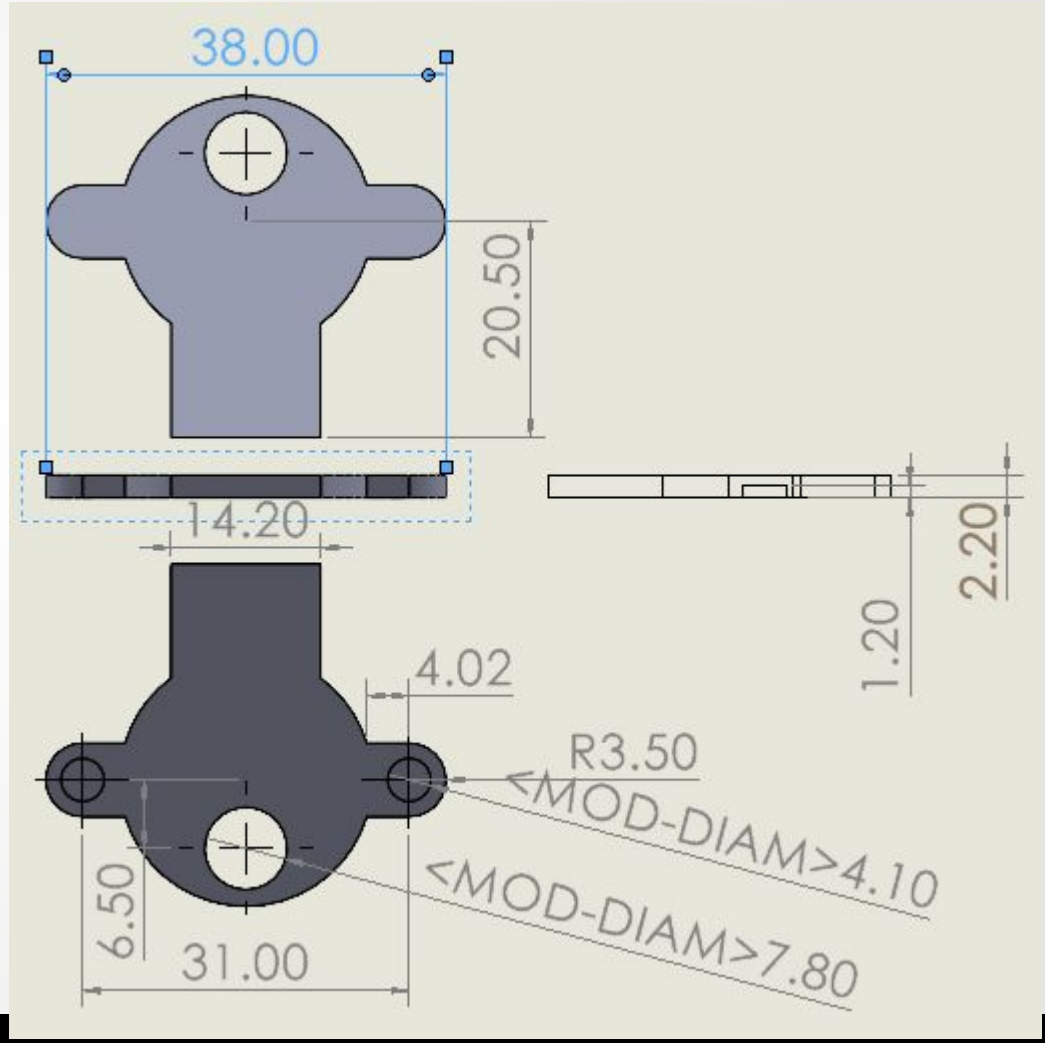


Casing Model Cont.

Motor Housing



Motor Cover



CDR Accomplishments/Challenges – Camera/Sensor Design

Camera Subsystem Software

- Created a socket interface to communicate camera subsystem data between Pico and an external device
  - live video feed, although it is rather slow
  - distance sensor data
- This took a lot of time and effort between team members and impeded some of the work on the app development
- HC-SR04 Capabilities
  - Measuring angle: 15 degrees
  - Accurate Ranging Distance: 2 cm - 400 cm

Dimensions: 80mm x 35mm
Weight: ~120g
Price: $28

Dimensions: 24.4 mm x 34.1 mm
Weight: 60g
Price: $25

Wirelessly Data Transfer

Socket in Python（Server - PC/Mobile App）

A so[illegible]of a two-way com[illegible]veen two pro[illegible] network.

```
import socket
import cv2
import numpy
```

- [illegible] number
- Server and clients

```
HOST = "172.20.10.8"
PORT = 80
```

```
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.bind((HOST, PORT))
    s.listen()
    print("Accepting connections")
    while True:
        conn, addr = s.accept()
        try:
            while True:

                data = conn.recv(30000)
                if data:
                    data2 = numpy.frombuffer((data+B'\xff'+B'\xd9'), dtype='uint8')
                    image = cv2.imdecode(data2, cv2.IMREAD_COLOR)  # 图像解码
                    if image is not None and image.size > 0:
                        cv2.imshow('Live Video', image)
                    else:
                        pass
                    if cv2.waitKey(10) & 0xFF == ord('q'):
                        break
                else:
                    break

        except BlockingIOError:
            pass
```

Wirelessly Data Transfer

Socketpool in C-Python (Clients - PCB/Pico W)

```python
import time as utime
import busio
import board
import usb_cdc
import digitalio
#import serial
from Arducam import *
from board import *
import adafruit_hcsr04
from adafruit_bus_device.spi_device import SPIDevice

import wifi
import socketpool
import ssl
import time


HOST = "172.20.10.8"
PORT = 80

# Connect to wifi
print("Connecting to wifi")
wifi.radio.connect("iPhone (3)", "88888888")
pool = socketpool.SocketPool(wifi.radio)
print("Creating Socket")

```

```python
    with pool.socket(pool.AF_INET, pool.SOCK_STREAM) as s:
        #print("Connecting")
        s.connect((HOST, PORT))
        #print("Sending")
        while True:
            mycam.spi.readinto(buffer,start=0,end=once_number)
            usb_cdc.data.write(buffer)
            sent = s.send(buffer)
            utime.sleep(0.00015)
            count+=once_number

            if count+once_number>lenght:
                count=lenght-count
                mycam.spi.readinto(buffer,start=0,end=count)
                usb_cdc.data.write(buffer)
                mycam.SPI_CS_HIGH()
                mycam.clear_fifo_flag()
                break
```

Wireless Data Transfer – Distance/Motor Data

On the server, we wait for the user to press a key:

- D is to receive distance data
- W is to return to our reference angle
- E is to rotate clockwise
- Q is to rotate counter clockwise

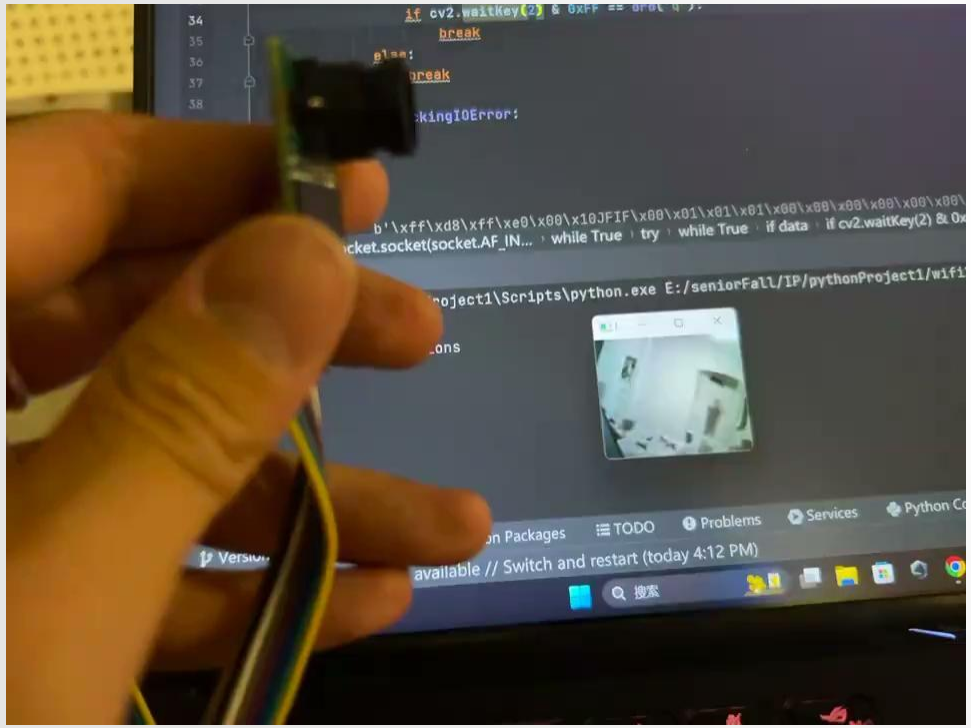
The client will receive the request and perform the corresponding operations.

When user input is detected, we send the corresponding byte to the client, asking for data to be sent back to the server, or for a command to be run

```
buff = bytearray(1)
numbytes = s.recv_into(buff)
if buff == b'1':
    #print(buff)
    send = s.send(str(sonar.distance/100) + " meters")
elif buff == b'2':
    scount = ecount + qcount
    if scount < 0:
        for i in range(abs(scount)):
            step("clockwise")
    else:
        for i in range(scount):
            step("counterclockwise")
        scount = 0
        ecount = 0
        qcount = 0
elif buff == b'3':
    step("clockwise")
    ecount = ecount + 1
elif buff == b'4':
    step("counterclockwise")
    qcount = qcount - 1
```

Camera Subsystem Verification

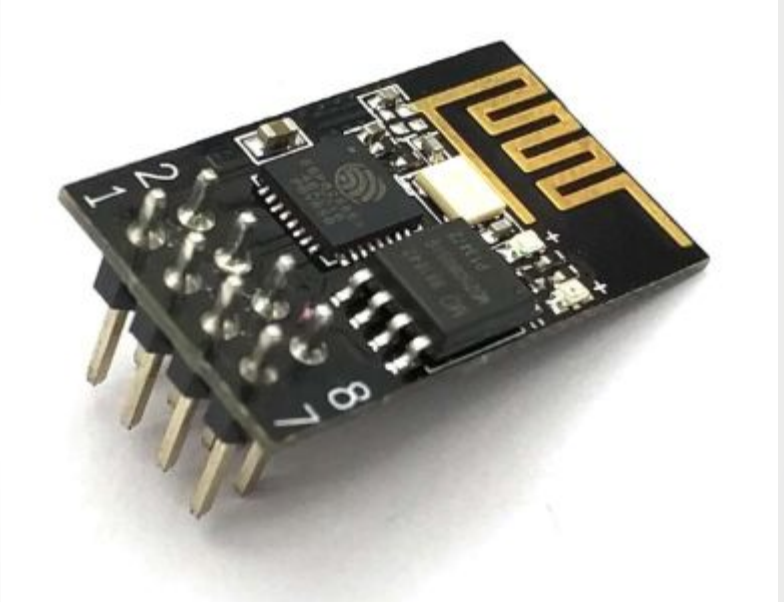
Live video demo



Distance sender demo

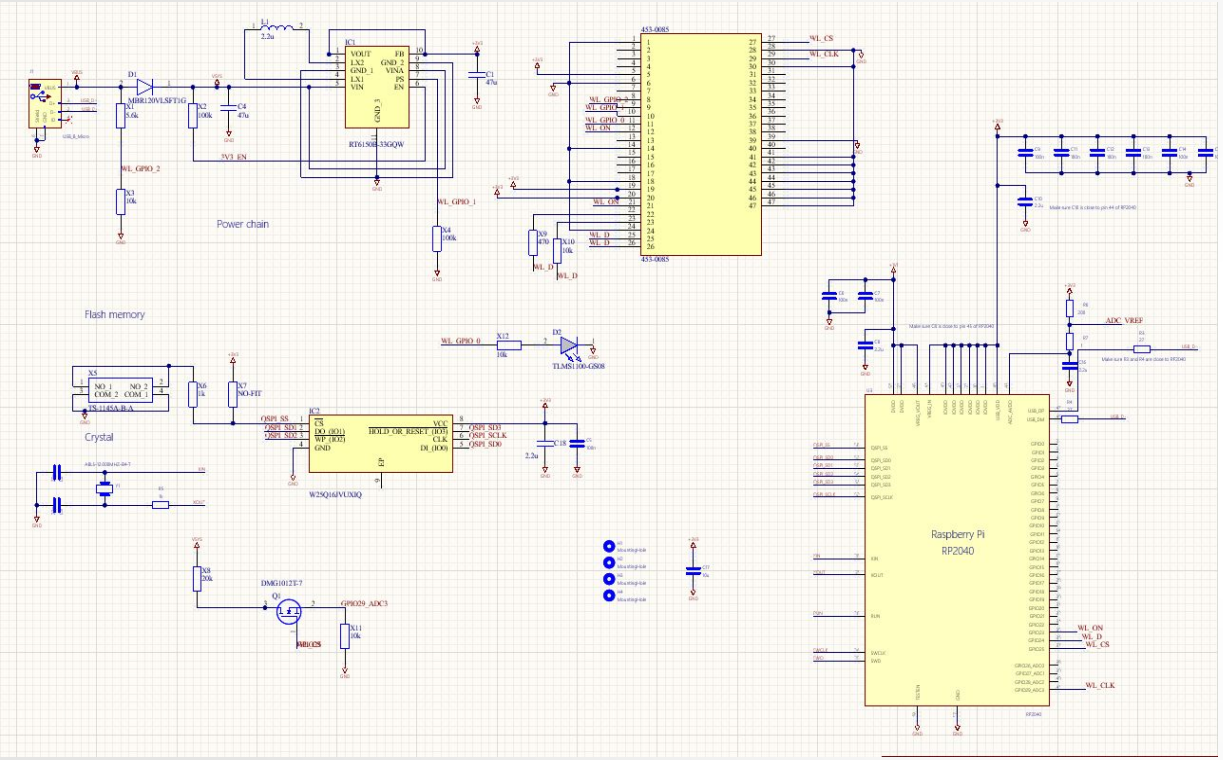
Wifi chip

CYW43439 chip

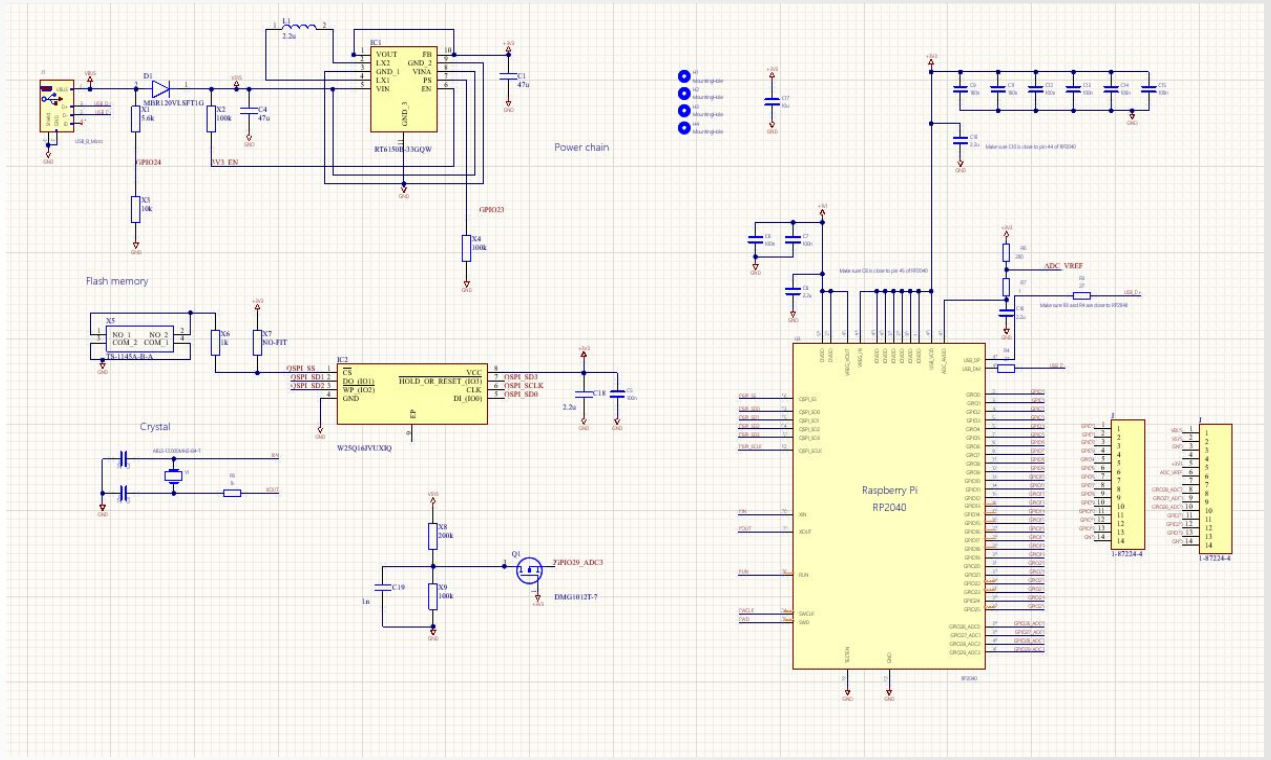


ESP-01 chip

PCB Schematic

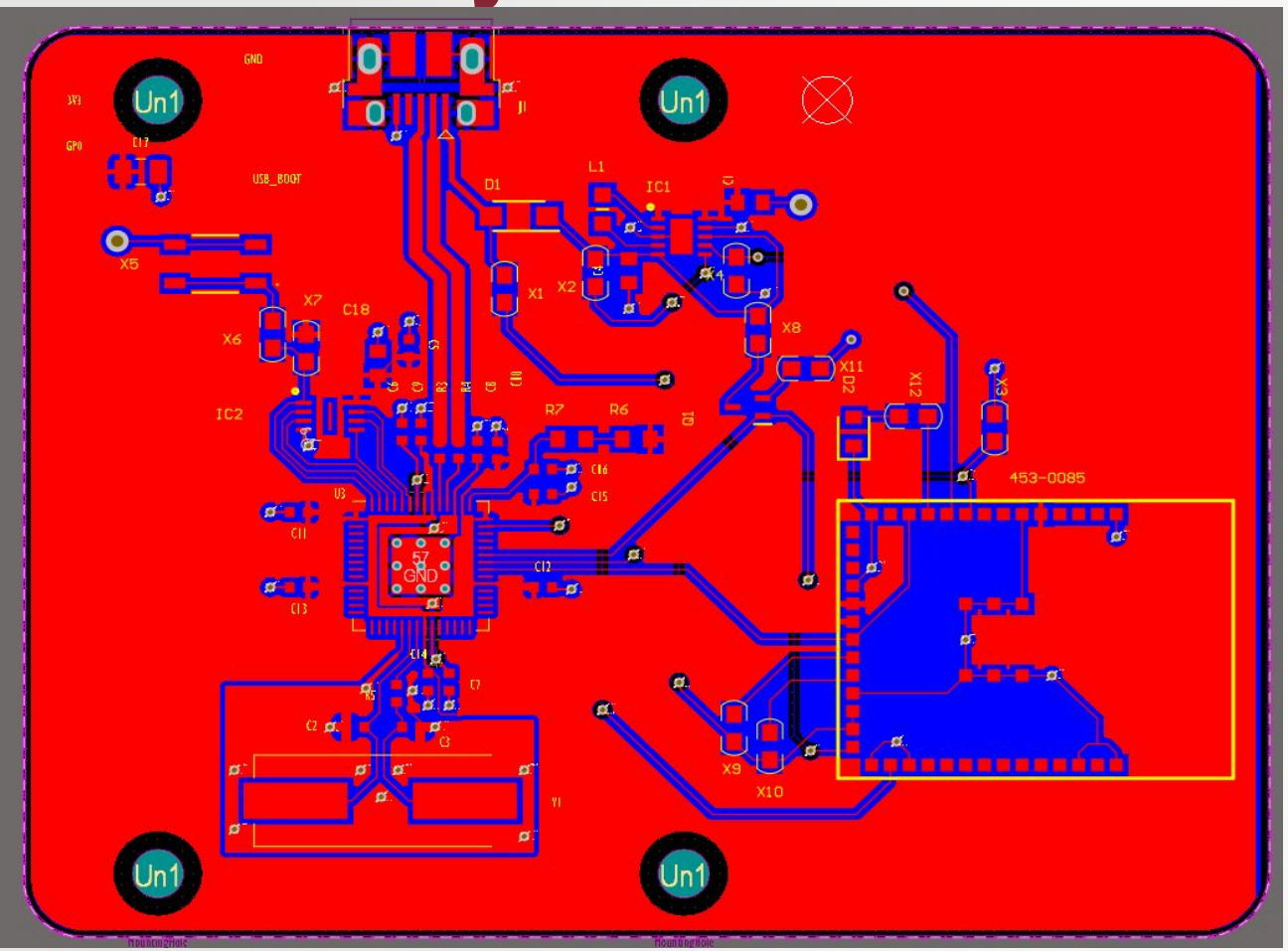


CYW43439 chip

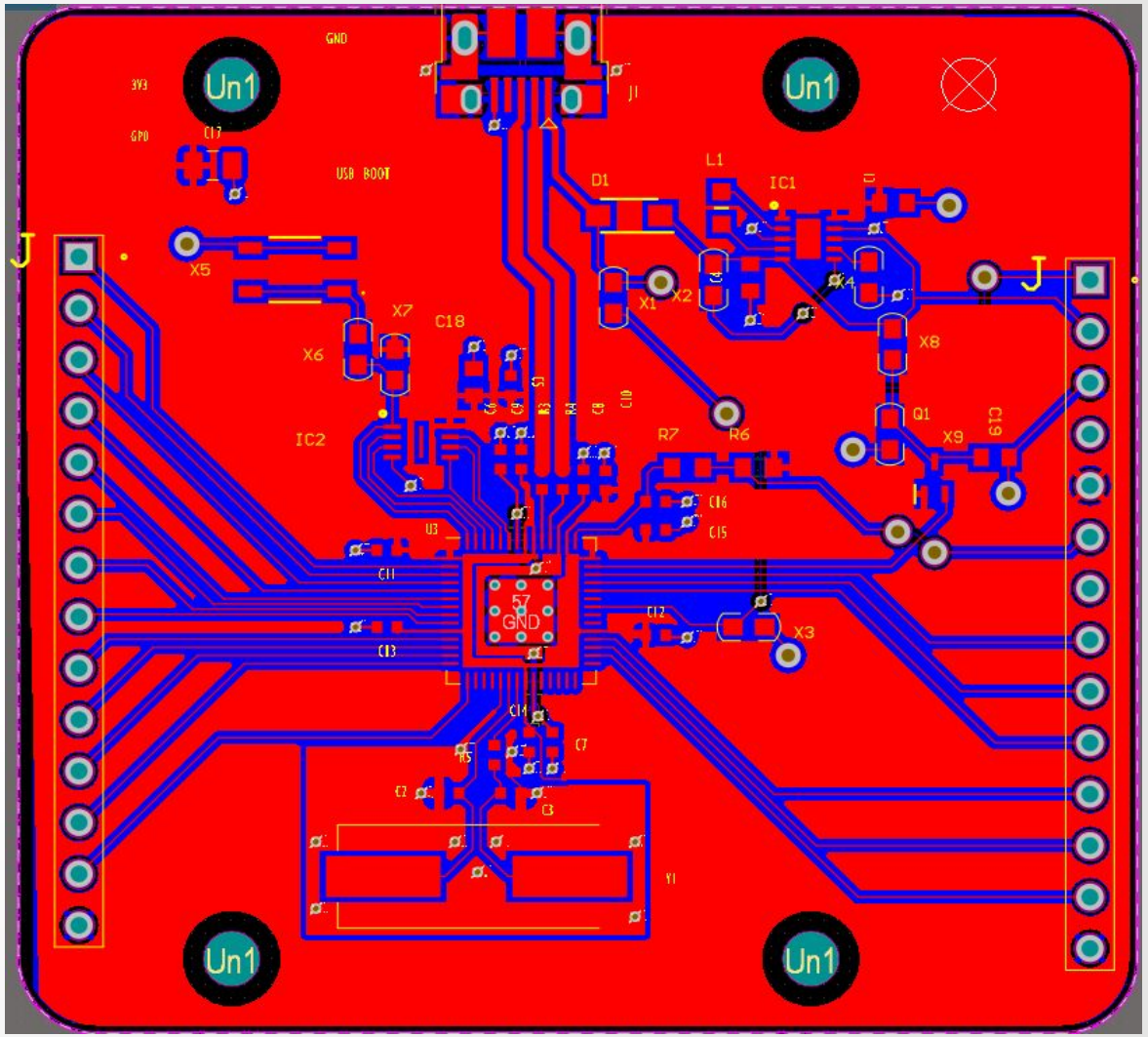


ESP-01 chip

PCB layout

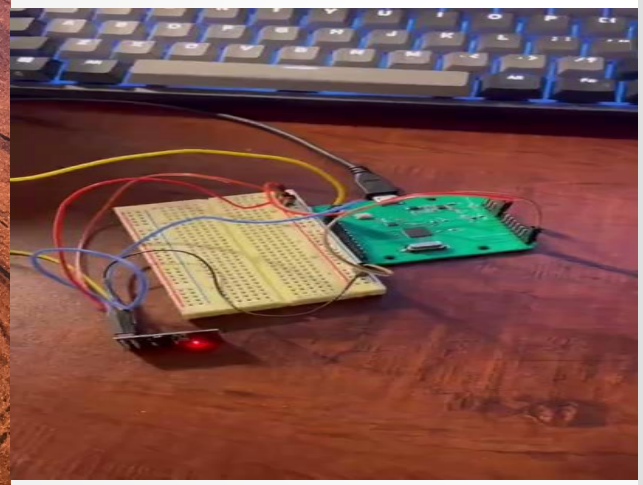


CYW43439 chip



ESP-01 chip

PCB demo



```
print("Try to connect with the WiFi..")

while (1):
    wifi_status = esp01.connectWiFi("yichong","umass15l")
    if "WIFI CONNECTED" in wifi_status:
        print("The ESP8266 has successfully connected to the WiFi...")
        break
    elif "WIFI DISCONNECT" in wifi_status:
        print("ESP8266 failed to connect to the WiFi. Retrying..")
        time.sleep(5)
    else:
        print("ESP8266 is attempting to connect to the WiFi..")
        time.sleep(5)

print("\r\n\r\n")
```

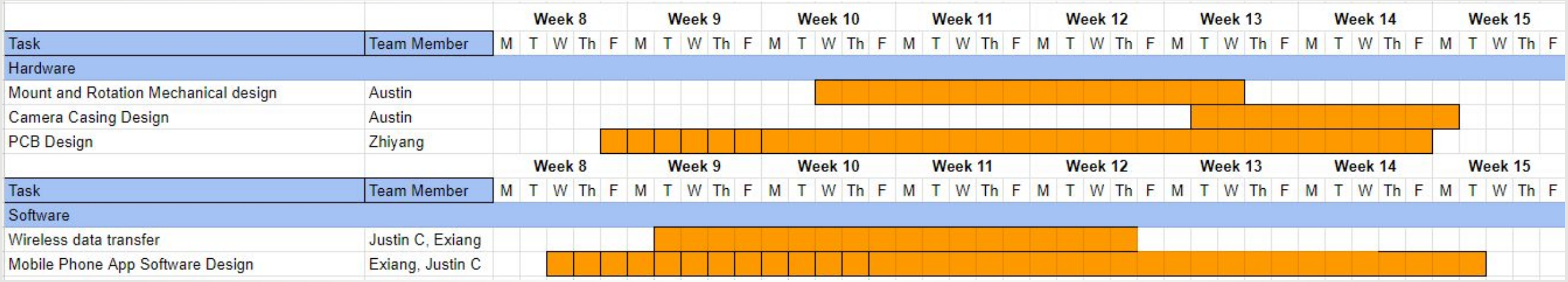
FPR Deliverables—PCB

Power supply

FPR Deliverables

- **Mounting System**
  - Final version of wireless housing models with built in power supplies
- **Mobile App**
  - Prototype app that receives visual feed and distance data
  - Will be able to wirelessly send commands to the microcontroller via GUI
    - Rotate Camera, Take screenshot, Show Distance
- **Camera Subsystem**
  - Will have three camera subsystems that work in parallel to send data to server

Gantt Chart



Task	Team Member	Week 8					Week 9					Week 10					Week 11					Week 12					Week 13					Week 14					Week 15				
		M	T	W	Th	F	M	T	W	Th	F	M	T	W	Th	F	M	T	W	Th	F	M	T	W	Th	F	M	T	W	Th	F	M	T	W	Th	F	M	T	W	Th	F
Hardware																																								
Mount and Rotation Mechanical design	Austin																																							
Camera Casing Design	Austin																																							
PCB Design	Zhiyang																																							
		Week 8					Week 9					Week 10					Week 11					Week 12					Week 13					Week 14					Week 15				
Task	Team Member	M	T	W	Th	F	M	T	W	Th	F	M	T	W	Th	F	M	T	W	Th	F	M	T	W	Th	F	M	T	W	Th	F	M	T	W	Th	F	M	T	W	Th	F
Software																																								
Wireless data transfer	Justin C, Exiang																																							
Mobile Phone App Software Design	Exiang, Justin C																																							

Parts List

• Raspberry Pi Pico x3	• $12
• HC-SR04 Distance Sensor x3	• $18
• Arducam Day&Night Vision Camera	• $28
• Arducam 2MP	• $25
• Ordered PCBs	• $35
• 24BJY48 x 3	• $0
• ULN2003 Driver x3	• $0
• 3D Printed Parts	• $0

SDP Budget Used: $151.41
Total: ~$175

Future Parts List

• Lithium Battery	• $30
• Arducam 2MP	• $25
• Heavy Duty Straps	• $10
• Custom PCBs	• $150
• 3D Printed Parts	• $0

Total: $255

Q&A

University of Massachusetts Amherst

Thank You!

University of Massachusetts Amherst

Works Cited

Proximity Sensing:

- https://ascencione.com/proximity-sensor-on-a-car-automobile/#:~:text=They%20are%20mounted%20on%20all,of%20up%20to%2010%20feet
- https://mycardoeswhat.org/safety-features/parking-sensors/
- https://www.chevrolet.com/support/vehicle/driving-safety/parking/front-rear-park-assist

Back Up Camera Collision Decrease:

- https://www.iihs.org/topics/bibliography/ref/2130

Arducam Research

- https://www.arducam.com/

Camera Panning system

- https://www.youtube.com/watch?v=hEBjbSTLytk

Background Information

- https://www.rhoadsandrhoads.com/blog/avoid-an-accident-and-injuries-with-safer-towing-and-trailering/#:~:text=The%20National%20Highway%20Traffic%20Safety,trailer%2C%20or%20an%20extra%20load

Similar Solutions

- https://www.amazon.com/Wireless-Waterproof-License-Monitor-Trailer/dp/B0768TW5MW
- https://bulepods.com/product/1080p-hd-mini-wireless-mini-camera-camcorder-wifi-outdoor-home-security-dvr/?gclid=Cj0KCQjw166aBhDEARIsAMEyZh6-ME4_35CjWQOa4GCF8a1MQw9MExEK2QYDPwgObFe4msGaK2f1U-YaAkMjEALw_wcB
- https://www.tadibrothers.com/products/9-monitor-with-wireless-mounted-rv-backup-camera?gclid=Cj0KCQjw166aBhDEARIsAMEyZh5TeRzjKDRa7v83kdlWjn4xN1IDf_P8eycc5BjxjW0tsCPysb3zWzAaArC_EALw_wcB
- https://www.walmart.com/ip/WiFi-HD-Wireless-Car-Rear-View-Cam-Wireless-Backup-Camera-Waterproof-Camera-for-Cars-Trucks-Vans-Pickups-SUVs-WiFi-Backup/769954848?wmlspartner=wlpa&selectedSellerId=18988

University of Massachusetts Amherst

Works Cited

Arducam 2MP setup:

- https://manuals.plus/arducam/ov2640-mini-2mp-spi-camera-on-raspberry-pi-pico-manual#axzz7mzPDUXjx

Distance Sensor setup:

- https://learn.adafruit.com/ultrasonic-sonar-distance-sensors/python-circuitpython

Thonny Pico Setup

- https://www.tomshardware.com/how-to/raspberry-pi-pico-setup#:~:text=Connect%20the%20Raspberry%20Pi%20Pico,Click%20Ok%20to%20close.

Raspberry Pi Pico hardware example:

- Raspberry Pi Documentation - Raspberry Pi Pico and Pico W

University of Massachusetts Amherst