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.Mobile software design and development

2.Data transfer with cameras

Austin Carroll

Mechanical Lead:

1.Camera Mount

2.Positioning system

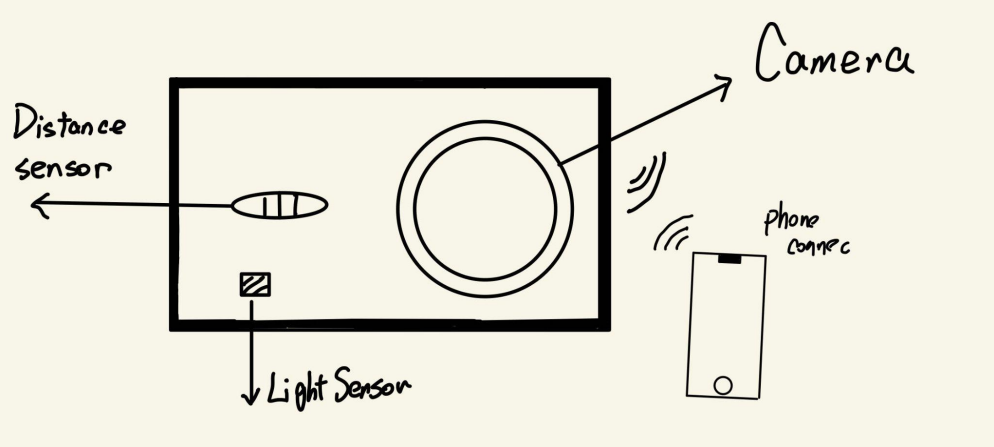
3. 3-D printing

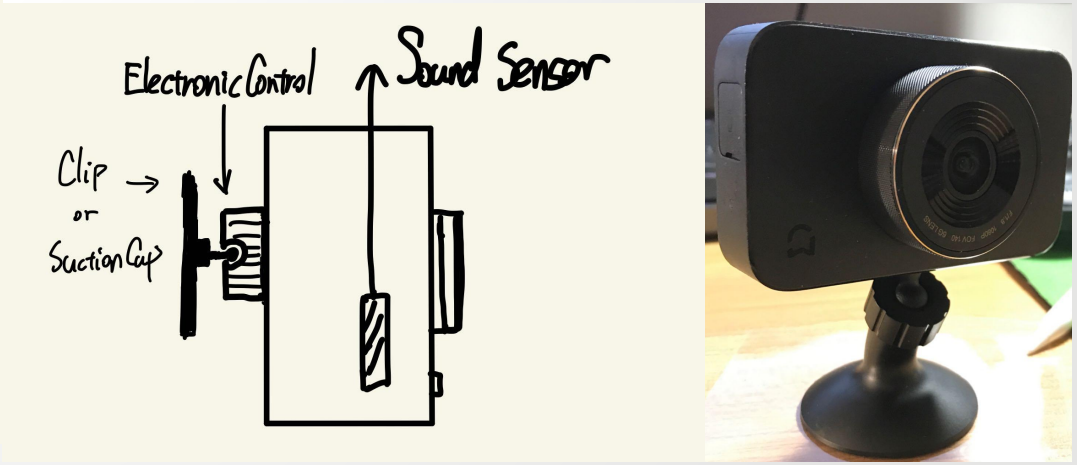
Project Goal Review

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.

Design Sketches:





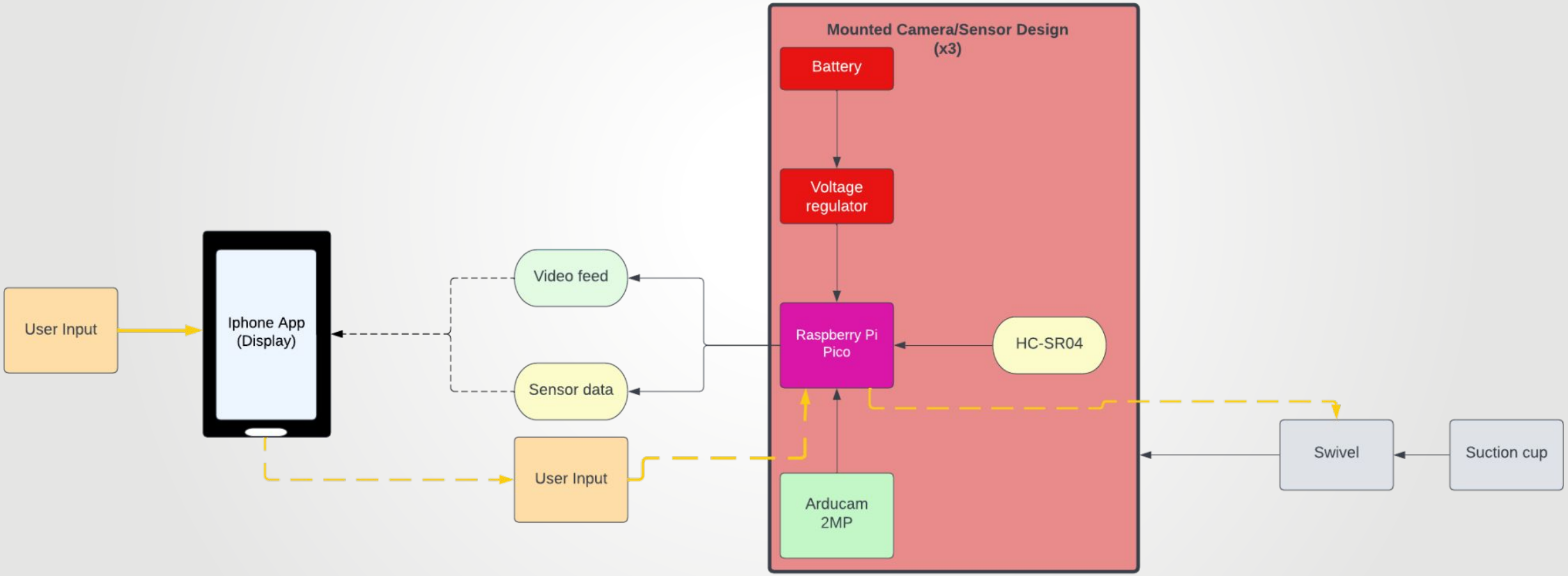
Specifications and Testing - Qualitative

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.

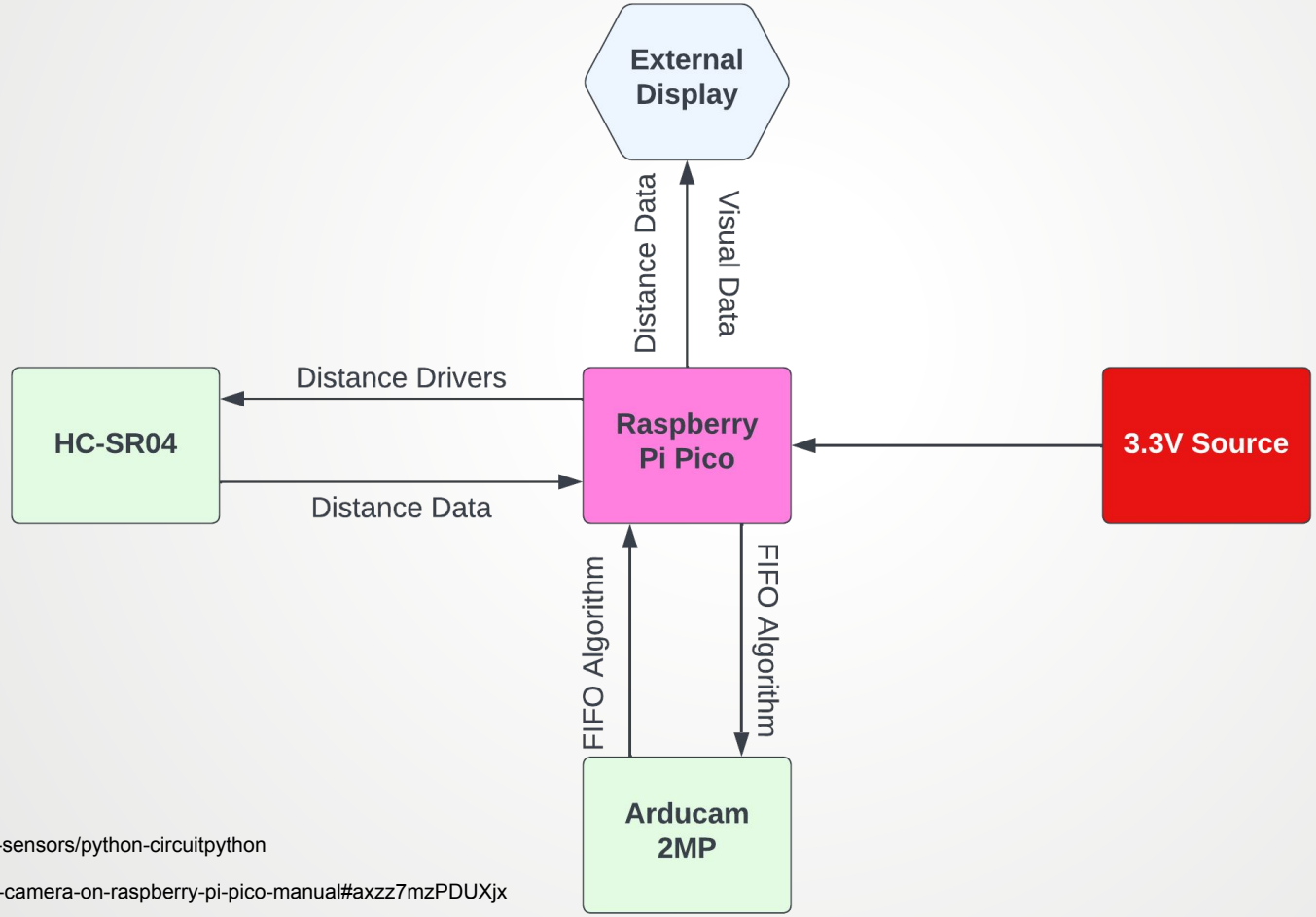
Specifications and Testing - Quantitative

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 10 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 100 x 100 pixels	Measure the resolution of the camera feed after data transfer using external software

Hardware Block Diagram



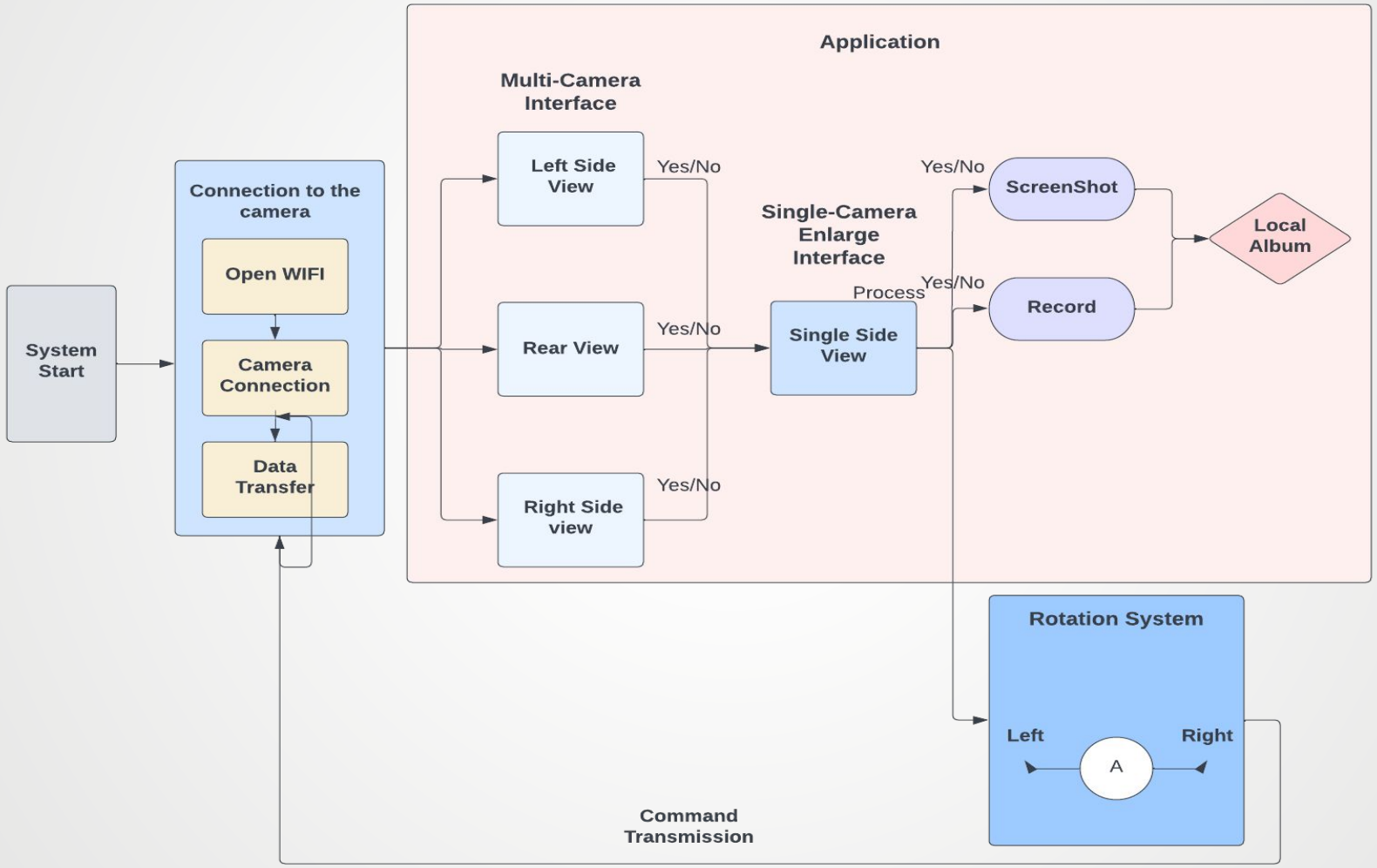
Hardware Block Diagram – Camera/Sensor System



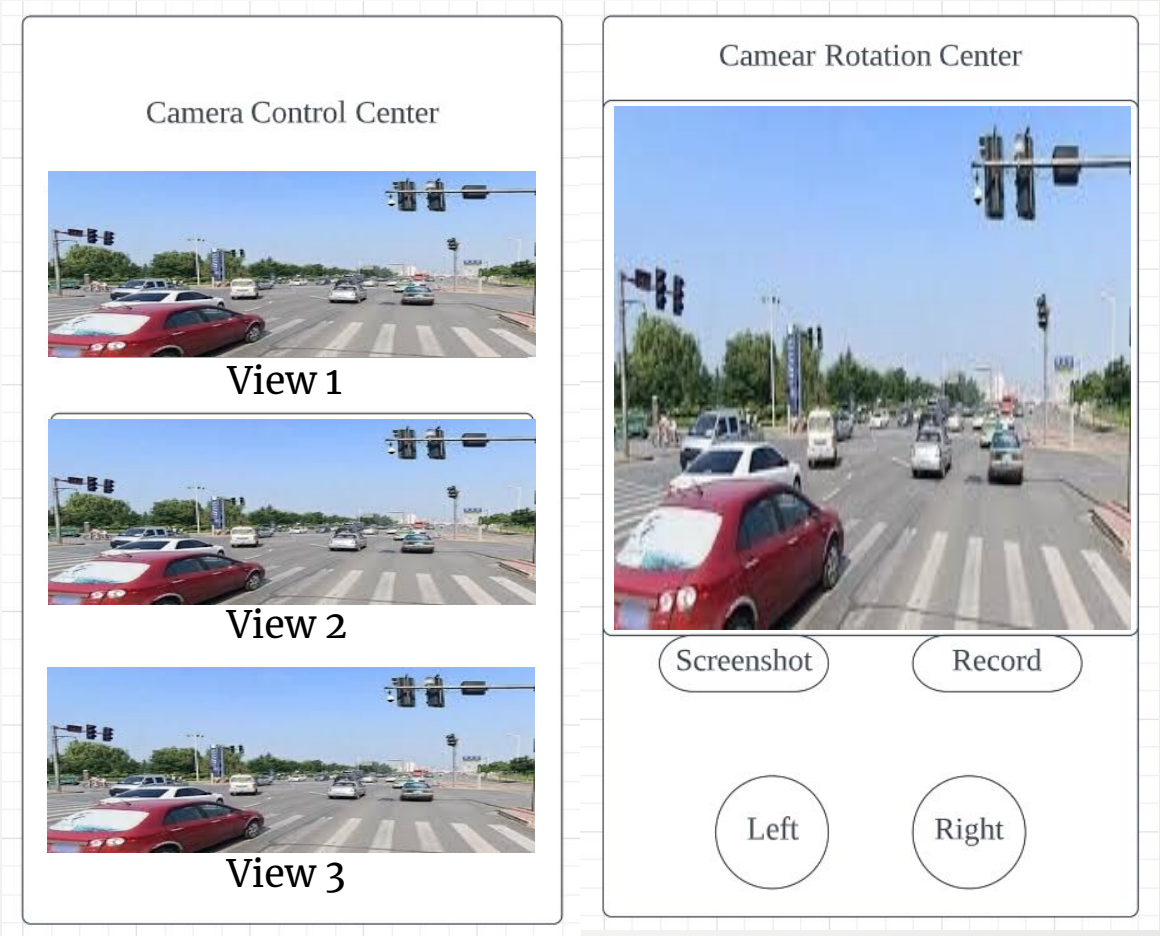
https://learn.adafruit.com/ultrasonic-sonar-distance-sensors/python-circuitpython

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

Software Diagram



Software Design Sketch



BE REVOLUTIONARY™

MDR Deliverables

University of Massachusetts Amherst

MDR Deliverables

- ❑ **Camera/Sensor Subsystem**
  - ❌ Cameras will have night vision capabilities
    - ✅ Cameras work in low-light conditions
  - ✅ Subsystem will transmit video, photo, and distance data to a display
- ❑ **Mobile App**
  - ❌ Cameras can be accessed and controlled through mobile app
- ❑ **Wireless Capabilities**
  - ✅ Microcontroller correctly receives the data or sends data directly to the user via WIFI
    - ❌ Transferring images or videos
    - ✅ Transfer of commands or text data
  - ✅ Demonstrate that data can be sent from pico to external device
- ❑ **Camera/Sensor Subsystem Rotation**
  - ✅ Camera Mounts can pan horizontally
- ❑ **PCB**
  - ✅ Designed
  - ✅ Fabricated
  - ✅ Populated
  - ✅ Tested
  - ❌ Working

MDR Accomplishments/Challenges – Camera/Sensor Design

Camera Subsystem Hardware

- Finished Camera/Sensor System
  - Arducam 2MP
  - Raspberry Pi Pico
  - HC-SR04
- Initial camera uses too much power and too large
  - Arducam 5MP IR Switch → Arducam 2MP
  -
- 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

Hardware Used and Power Considerations

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

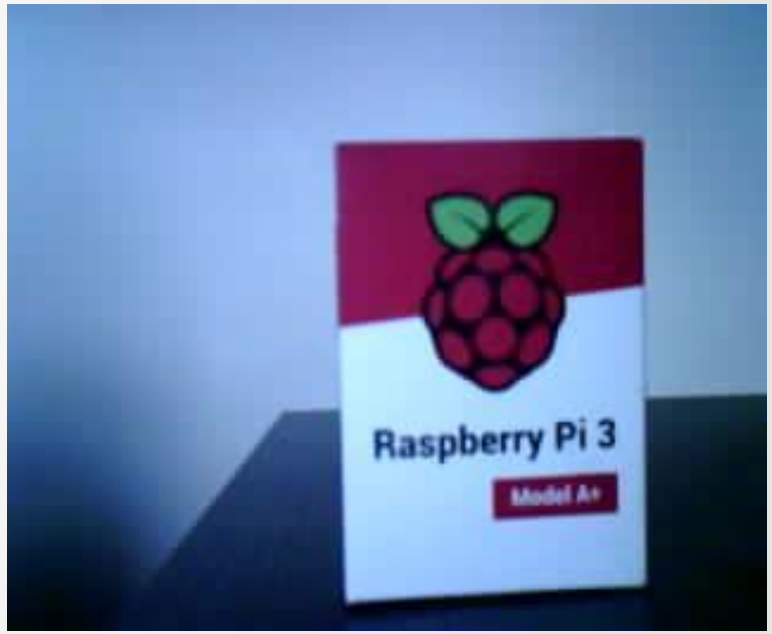
- Motor Subsystem
  - NEMA-17 Stepper Motor - 12V/ 350 mA 4.2 W
  - A4988 Stepper Motor Driver
    - -Operating Voltage - 12 V
    - -Logic Voltage - 5 V
  - Arduino Uno - 5V/ 15mA 0.075 W
  - 100 uF capacitor
  - 2 x 6V Battery Pack in series

Camera Subsystem Verification

Low Resolution
160x120 pixels (30fps)

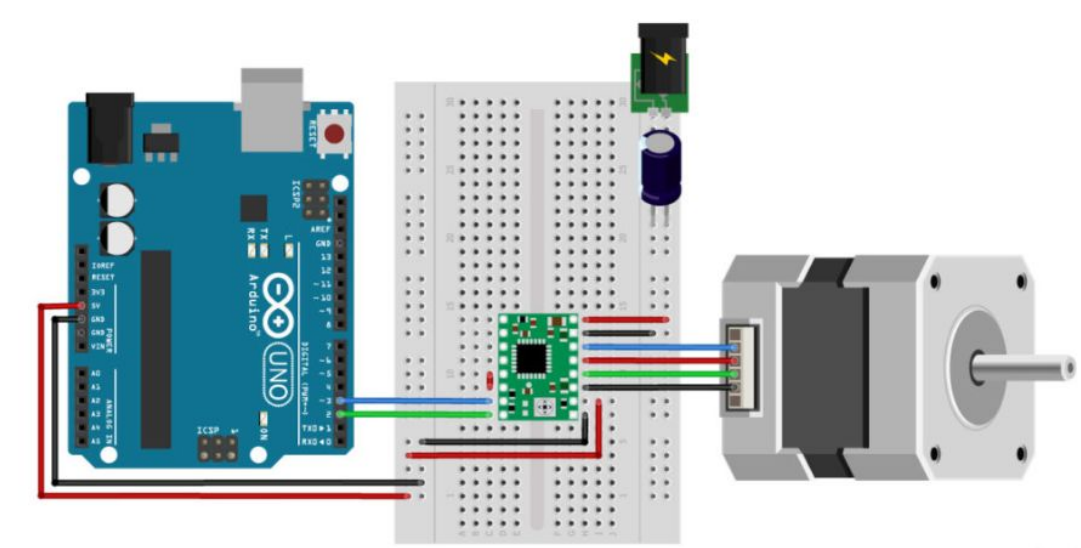
High Resolution
1600x1200 pixels (2fps)

Low Light



Motor Subsystem

Schematic



https://www.makerguides.com/a4988-stepper-motor-driver-arduino-tutorial/

Arduino Code

```
// Define pin connections & motor's steps per revolution
const int dirPin = 2;
const int stepPin = 3;
const int stepsPerRevolution = 200;

void setup()
{
  // Declare pins as Outputs
  Serial.begin(9600);
  pinMode(stepPin, OUTPUT);
  pinMode(dirPin, OUTPUT);
}
void loop()
{
 if (Serial.available())
 {
  char Serial_data =Serial.read();
  Serial.println(Serial_data);
  if (Serial_data == 'c')
  {
    digitalWrite(dirPin, HIGH);

  for(int x = 0; x < stepsPerRevolution; x++)
  {
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(2000);
    digitalWrite(stepPin, LOW);

  }}
  else if (Serial_data == 'a')
 {
  digitalWrite(dirPin, LOW);

  for(int x = 0; x < stepsPerRevolution; x++)
  {
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(2000);
    digitalWrite(stepPin, LOW);

 }} else
{
   digitalWrite(dirPin,LOW);
    digitalWrite(stepPin,LOW);
}}}
```

Motor Subsystem Cont.

Reference Voltage for A4988

Vref = Imax * 8 * Rs

Imax of Motor = 350 mA

Rs = 0.05 ohm

Vref = 0.14 V

Adjusted to 0.08

Motor Specs

1.8 degree angle per step

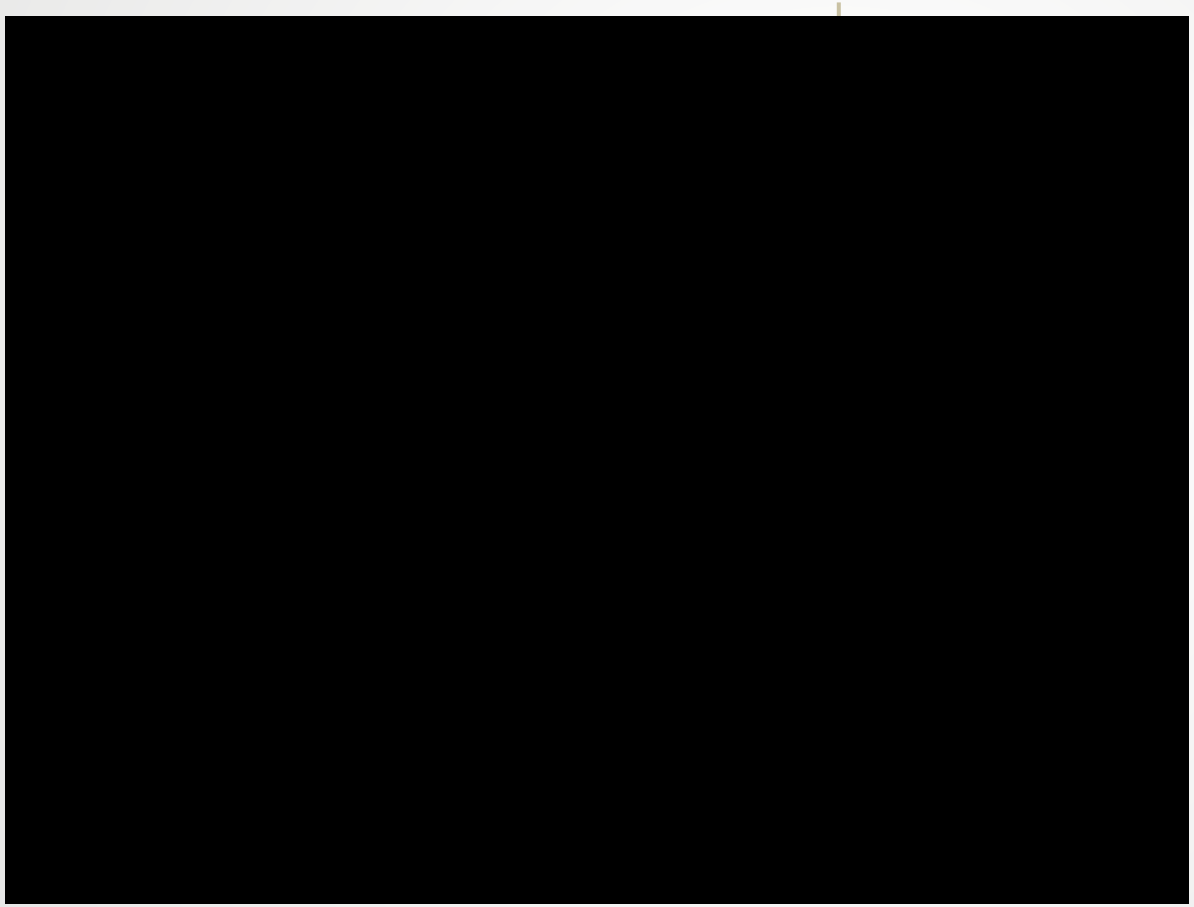
200 rpm

3.71 lb-in of torque

Driver Pin Connections

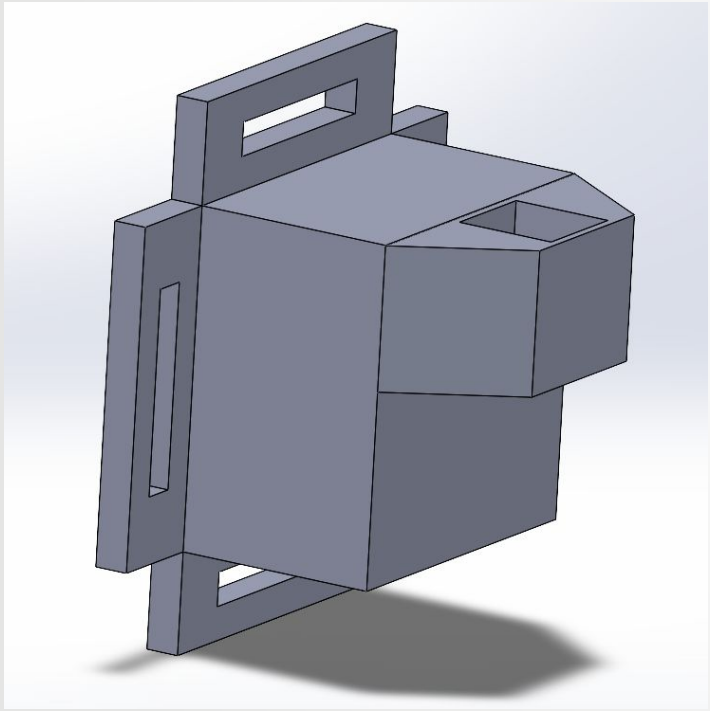
A4988	Connection
VMOT	8-35V
GND	Motor ground
SLP	RESET
RST	SLP
VDD	5V
GND	Logic ground
STP	Pin 3
DIR	Pin 2
1A, 1B, 2A, 2B	Stepper motor

Motor Demo

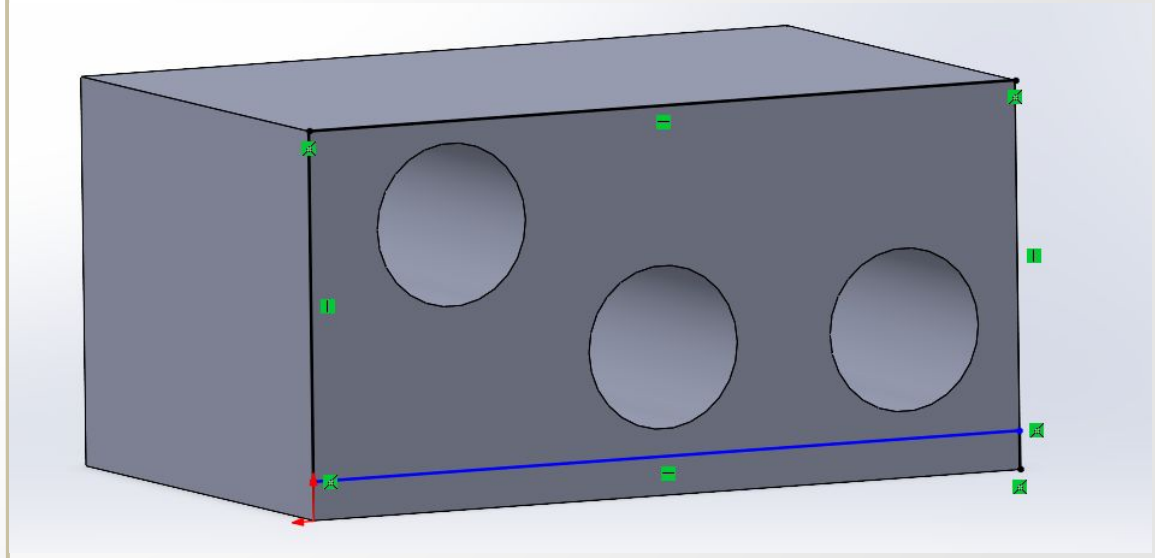


Casing Concept Model

Motor, Power Supply, PCB Housing



Camera and Distance Sensor Housing

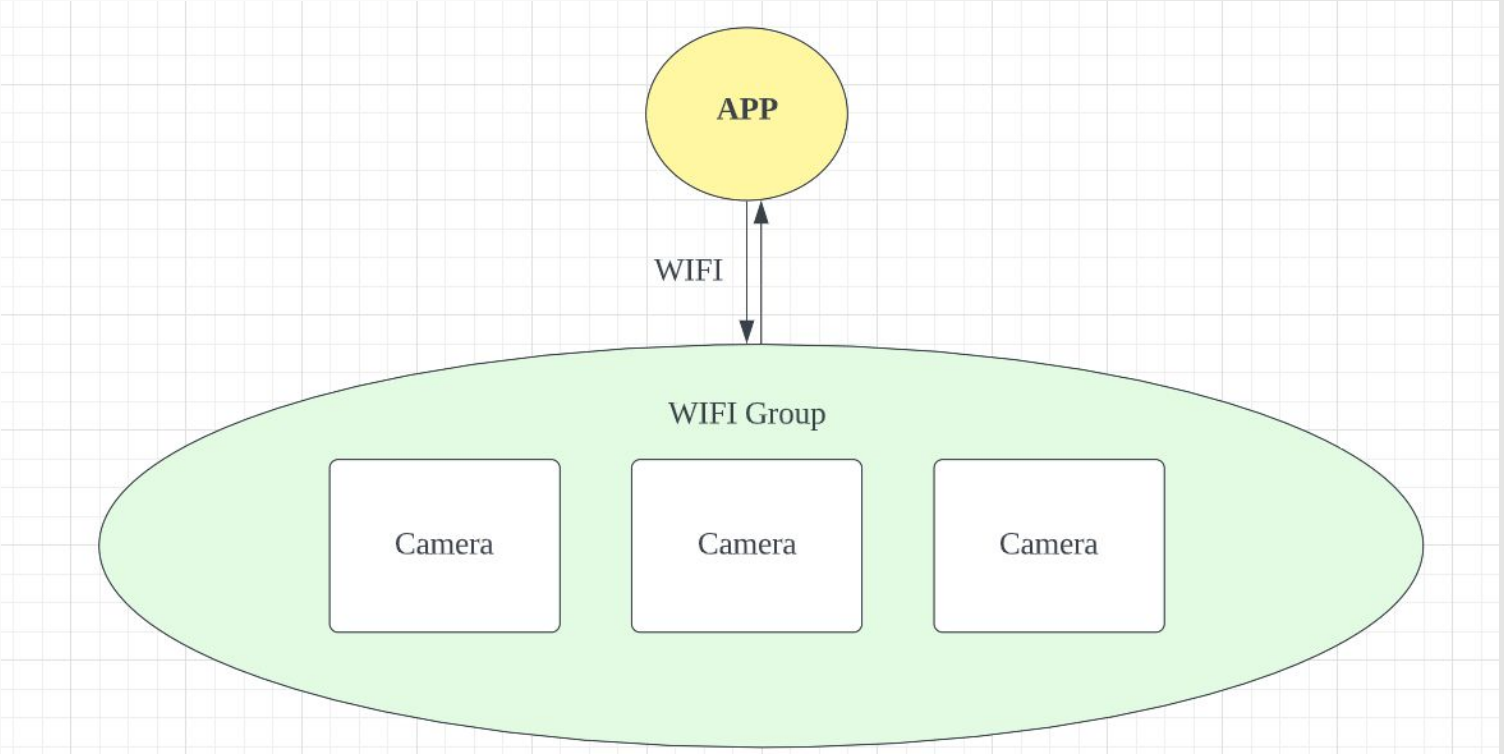


Wirelessly Data Transfer

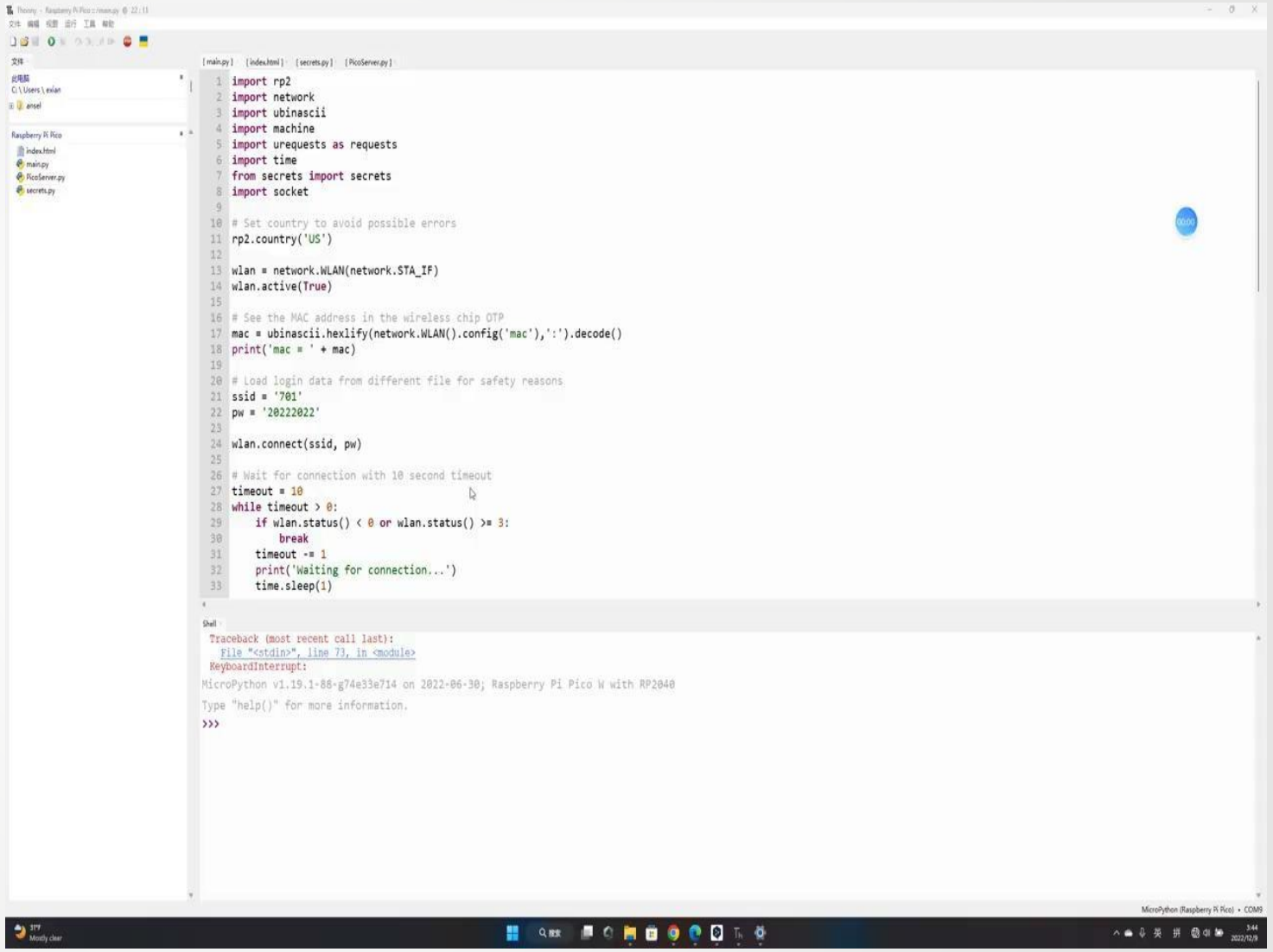
WIFI Direct

WiFi Direct is built upon the same WiFi technology used by most modern consumer electronic devices to communicate with wireless routers.

Transmission speed is 100 times faster than Bluetooth



Code demo

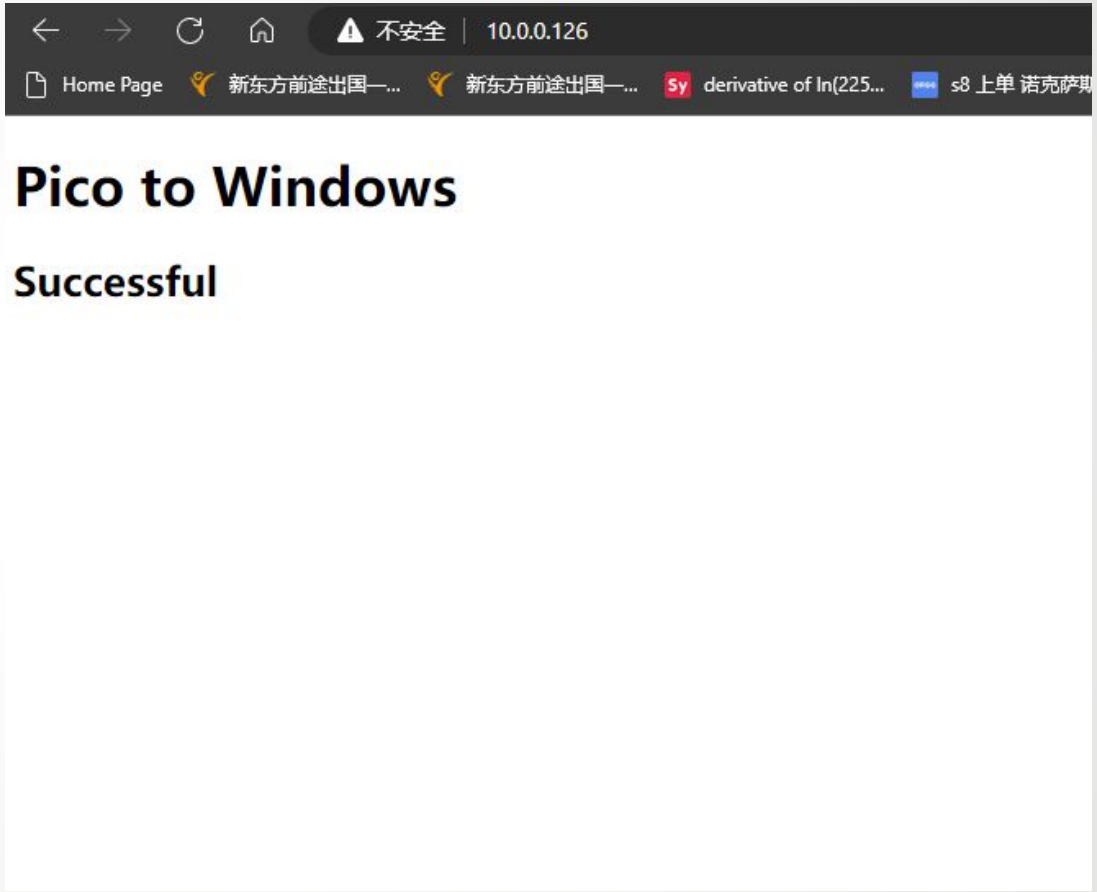

main.py
index.html
secrets.py
PicoServer.py
import rp2
import network
import ubinascii
import machine
import urequests as requests
import time
from secrets import secrets
import socket
# Set country to avoid possible errors
rp2.country('US')
wlan = network.WLAN(network.STA_IF)
wlan.active(True)
# See the MAC address in the wireless chip OTP
mac = ubinascii.hexlify(network.WLAN().config('mac'),':').decode()
print('mac = ' + mac)
# Load login data from different file for safety reasons
ssid = '701'
pw = '20222022'
wlan.connect(ssid, pw)
# Wait for connection with 10 second timeout
timeout = 10
while timeout > 0:
if wlan.status() < 0 or wlan.status() >= 3:
break
timeout -= 1
print('Waiting for connection...')
time.sleep(1)
Shell
Traceback (most recent call last):
File "<stdin>", line 73, in <module>
KeyboardInterrupt:
MicroPython v1.19.1-88-g74e33e714 on 2022-06-30; Raspberry Pi Pico W with RP2040
Type "help()" for more information.
>>>

Wirelessly Data Transfer (Windows)

```python
import rp2
import network
import ubinascii
import machine
import urequests as requests
import time
from secrets import secrets
import socket

# Set country to avoid possible errors
rp2.country('US')

wlan = network.WLAN(network.STA_IF)
wlan.active(True)

# See the MAC address in the wireless chip OTP
mac = ubinascii.hexlify(network.WLAN().config('mac'),':').decode()
print('mac = ' + mac)

# Load login data from different file for safety reasons
ssid = '701'
pw = '20222022'

wlan.connect(ssid, pw)

# Wait for connection with 10 second timeout
timeout = 10
while timeout > 0:
    if wlan.status() < 0 or wlan.status() >= 3:
        break
```

Shell

```
>>> %Run -c $EDITOR_CONTENT

  mac = 28:cd:c1:00:d5:29
  Connected
  ip = 10.0.0.126
  Listening on ('0.0.0.0', 80)
  Client connected from ('10.0.0.19', 60390)
  Client connected from ('10.0.0.19', 60389)
  Client connected from ('10.0.0.19', 60391)
```



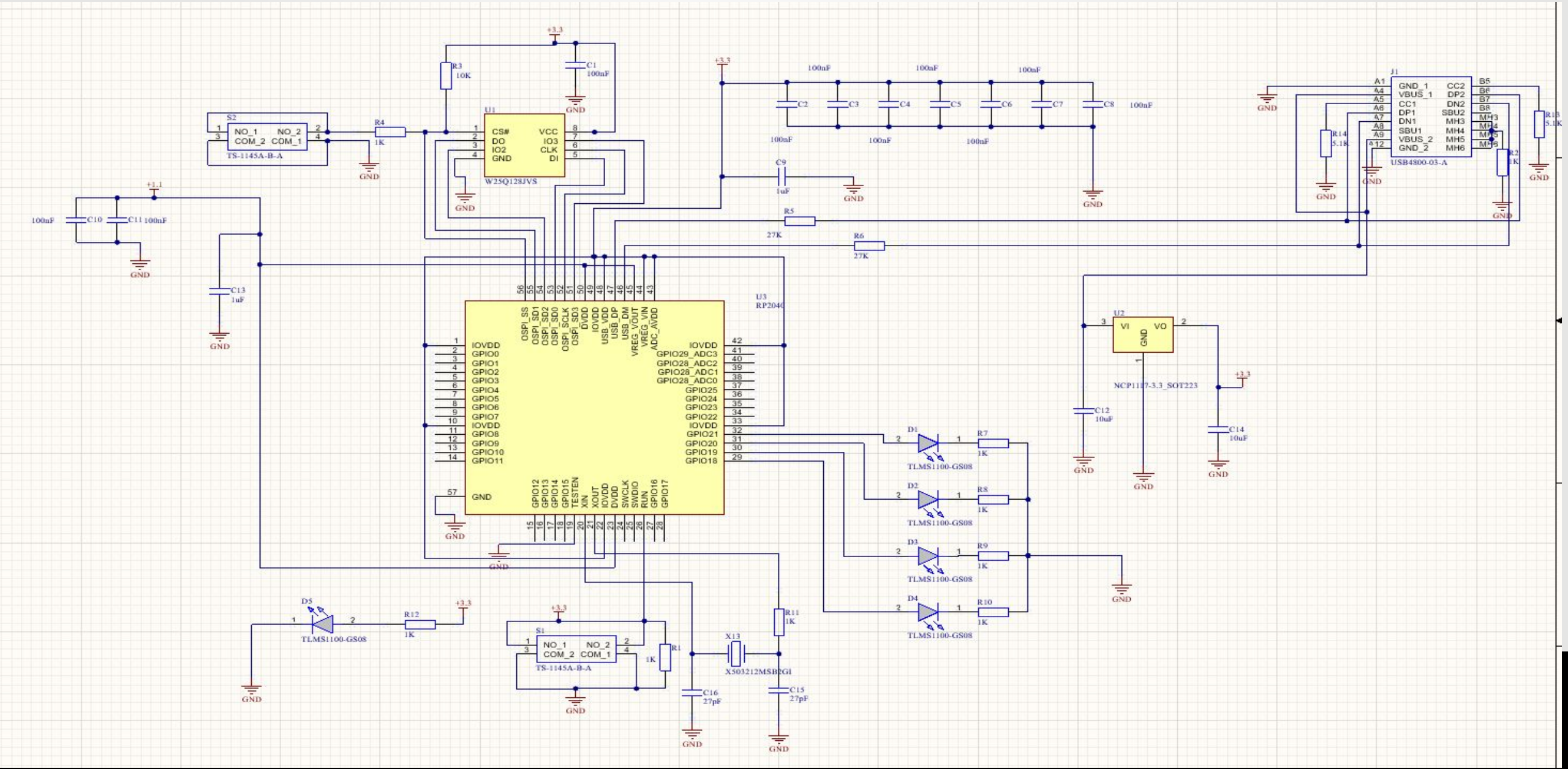
Wirelessly Data Transfer (PICO)

```python
import socket
import network
import machine

ssid = 'MicroPython-AP'
password = '123456789'

led = machine.Pin("LED",machine.Pin.OUT)

ap = network.WLAN(network.AP_IF)
ap.config(essid=ssid, password=password)
ap.active(True)

while ap.active() == False:
  pass

print('Connection successful')
print(ap.ifconfig())

html = """<!DOCTYPE html>
<html>
    <head> <title>Pico W</title> </head>
    <body> <h1>Pico to Windows (pico as server)</h1>
           <h2>Successful</h2>
        <p>Hello from Pico W.</p>
    </body>
</html>
"""

addr = socket.getaddrinfo('0.0.0.0', 80)[0][-1]
s = socket.socket()
s.bind(addr)
s.listen(1)

print('listening on', addr)
led.off()

# Listen for connections
while True:
```

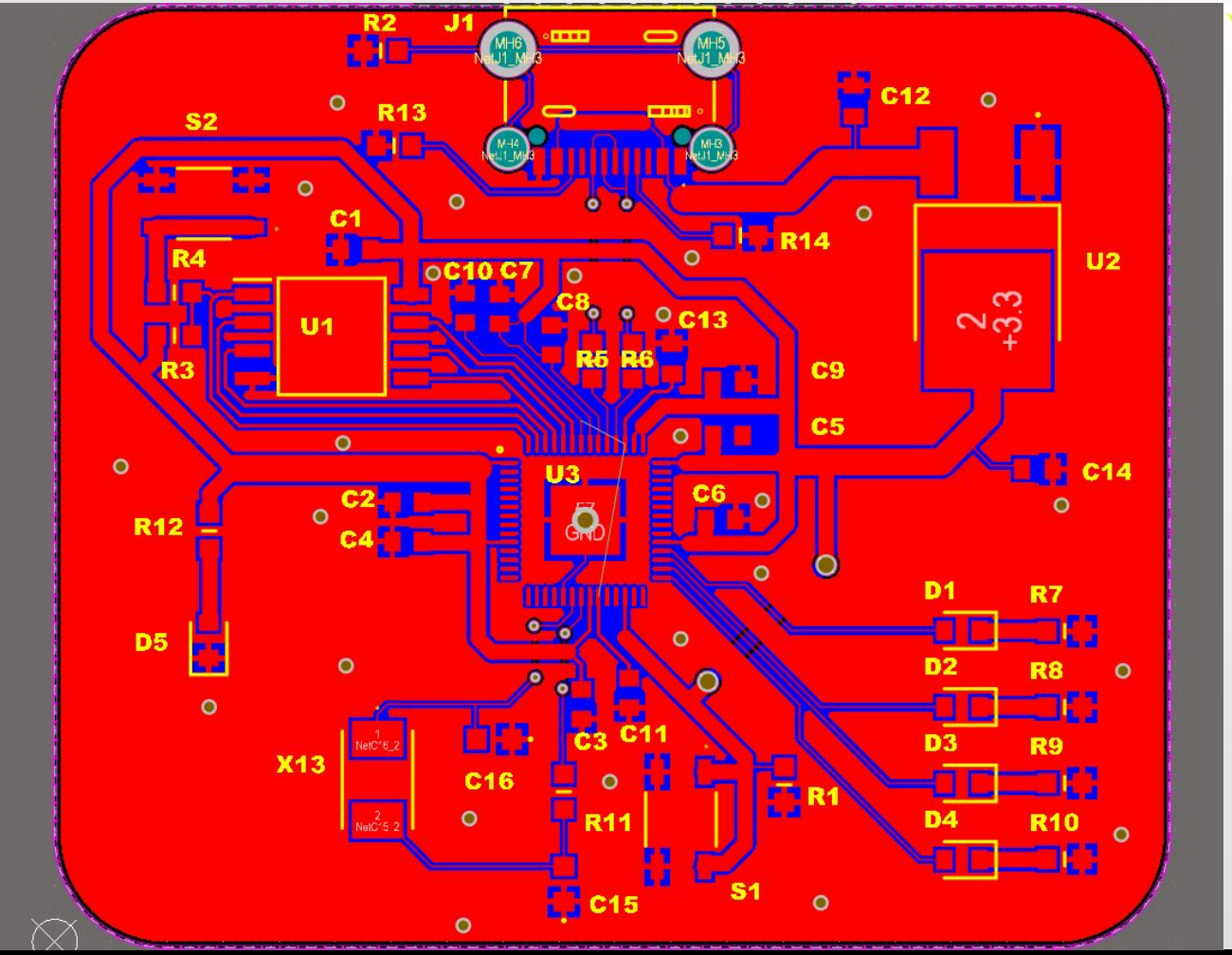
Shell

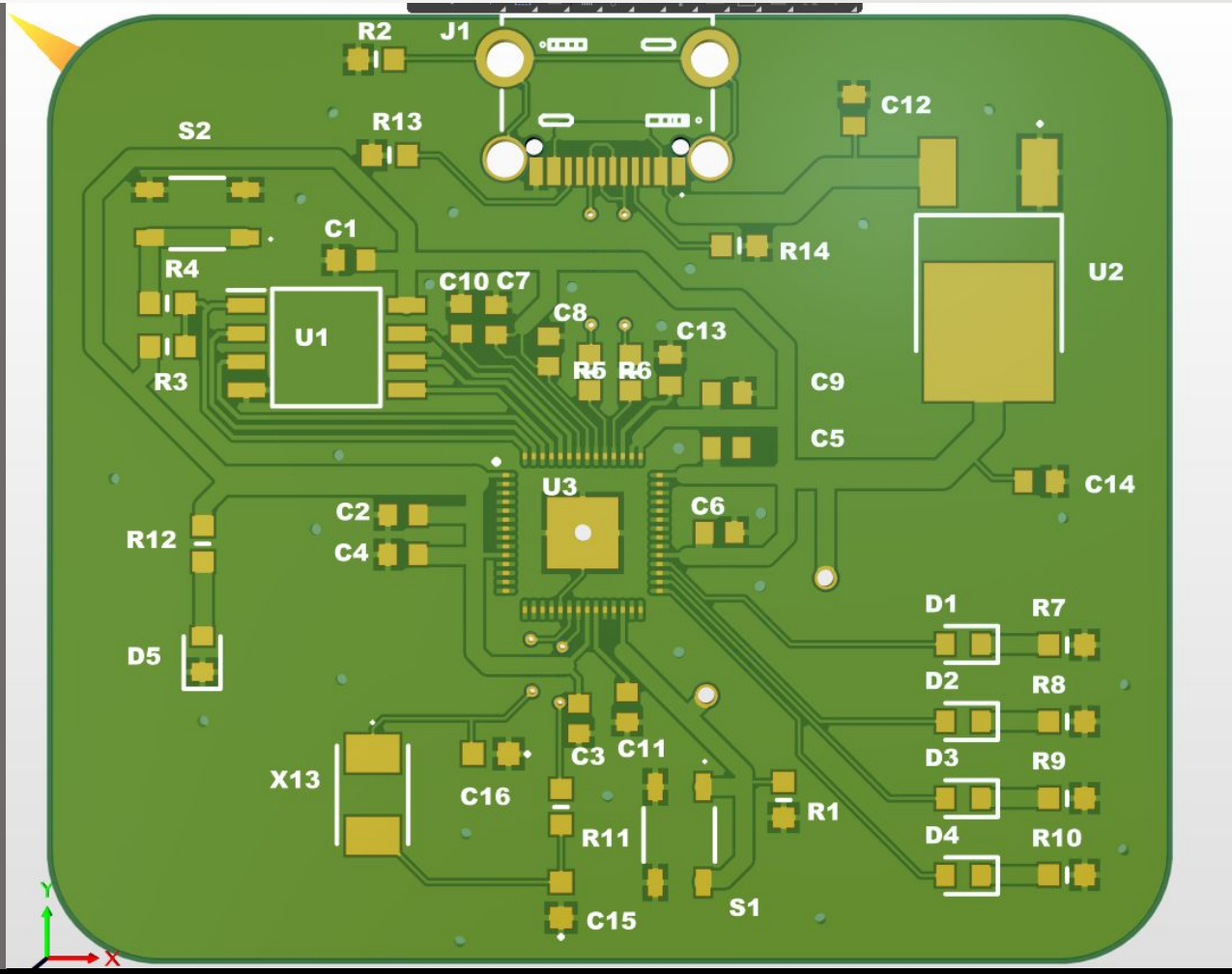
```
>>> %Run -c $EDITOR_CONTENT

  Connection successful
  ('192.168.4.1', '255.255.255.0', '192.168.4.1', '8.8.8.8')
  listening on ('0.0.0.0', 80)
  client connected from ('192.168.4.16', 59438)
  b'GET / HTTP/1.1\r\nHost: 192.168.4.1\r\nConnection: keep-al
  ike Gecko) Chrome/107.0.0.0 Safari/537.36 Edg/107.0.1418.62\
  oding: gzip, deflate\r\nAccept-Language: zh-CN,zh;q=0.9,en;q
  client connected from ('192.168.4.16', 59439)
```

PCB Schematic

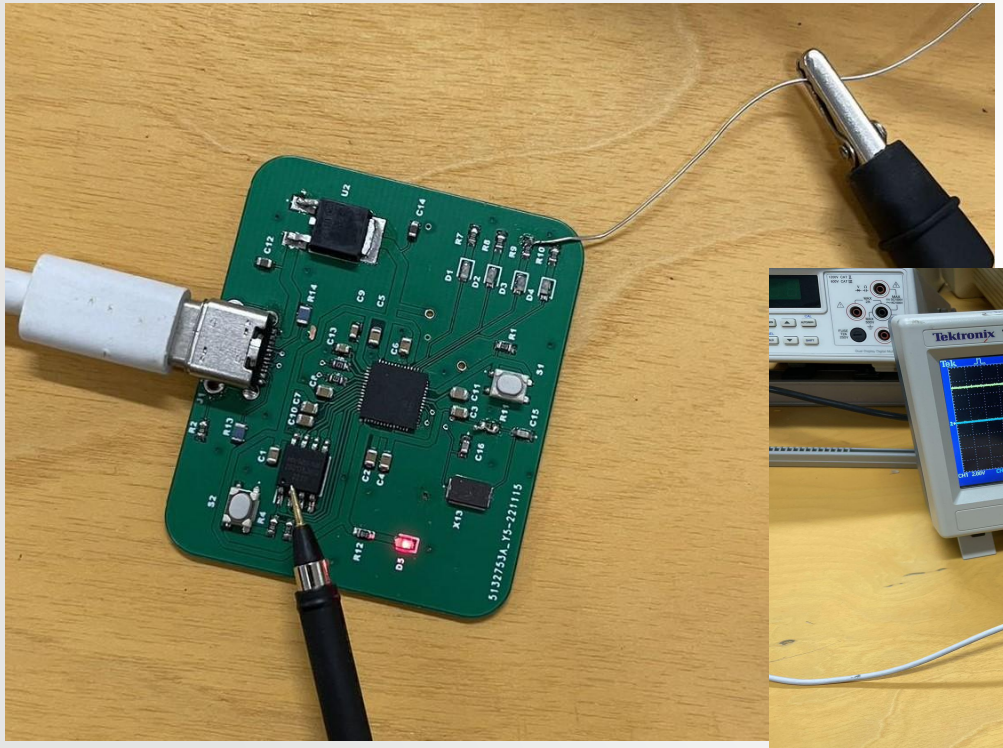


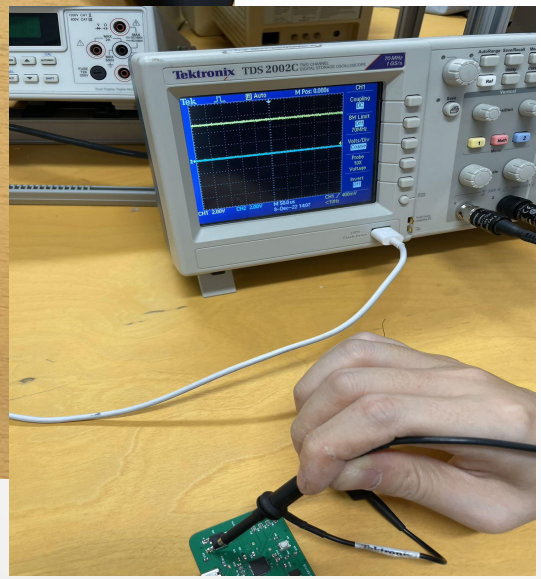
PCB layout

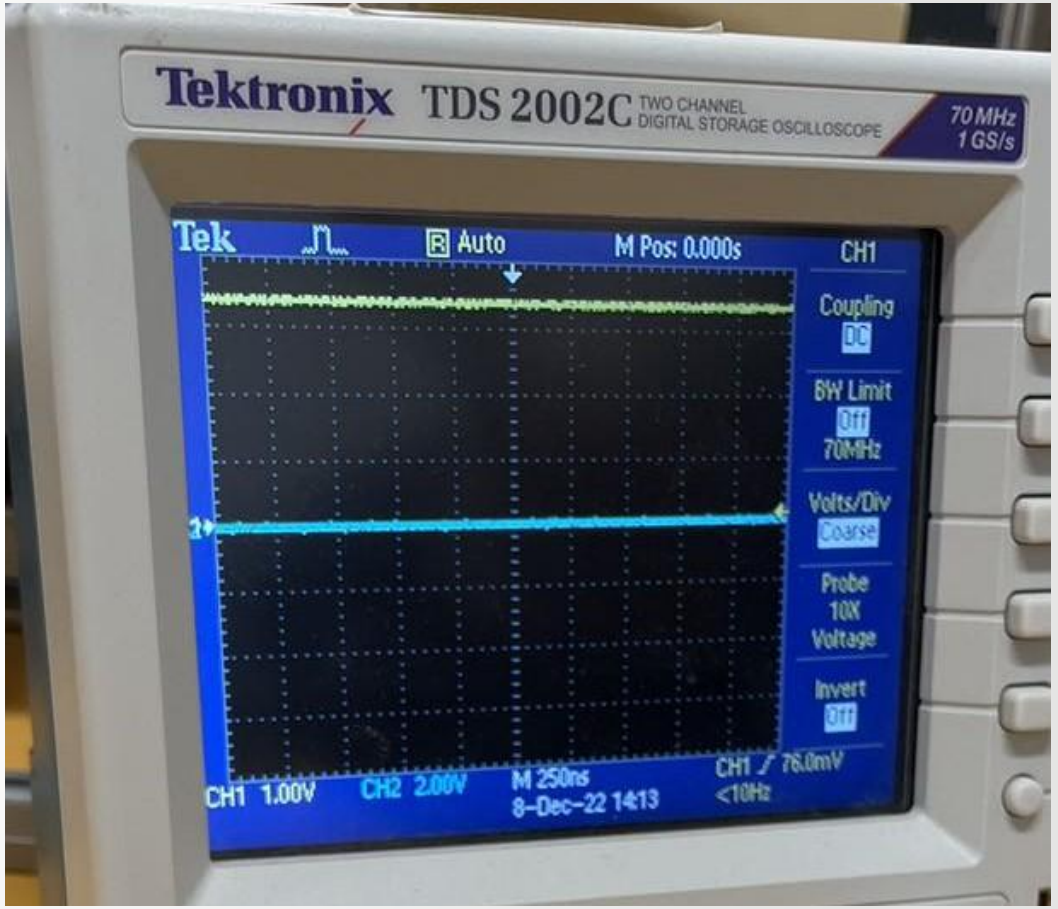

R2
J1
MH6
MH5
S2
R13
C12
C1
R4
R14
U2
C10 C7
C8
C13
U1
R5 R6
R3
C9
C5
U3
C14
C2
C6
R12
C4
D1
R7
D5
D2
R8
D3
R9
X13
C3
C11
C16
R1
D4
R10
R11
S1
C15


R2
J1
S2
R13
C12
C1
R4
R14
U2
C10 C7
C8
C13
U1
R5 R6
R3
C9
C5
U3
C14
C2
C6
R12
C4
D1
R7
D5
D2
R8
D3
R9
X13
C3
C11
C16
R1
D4
R10
R11
S1
C15

Troubleshooting



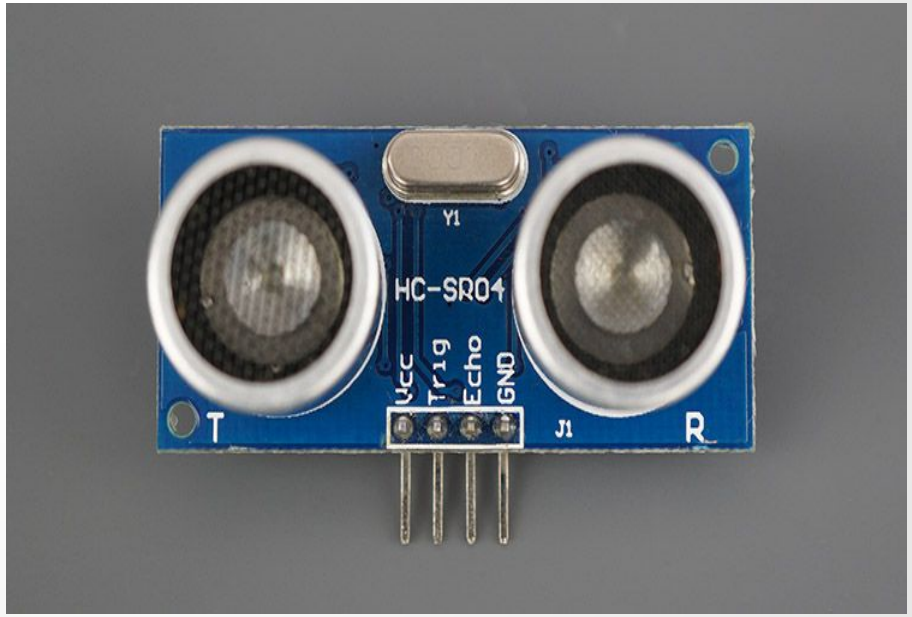




CDR Deliverables—PCB

Wifi module

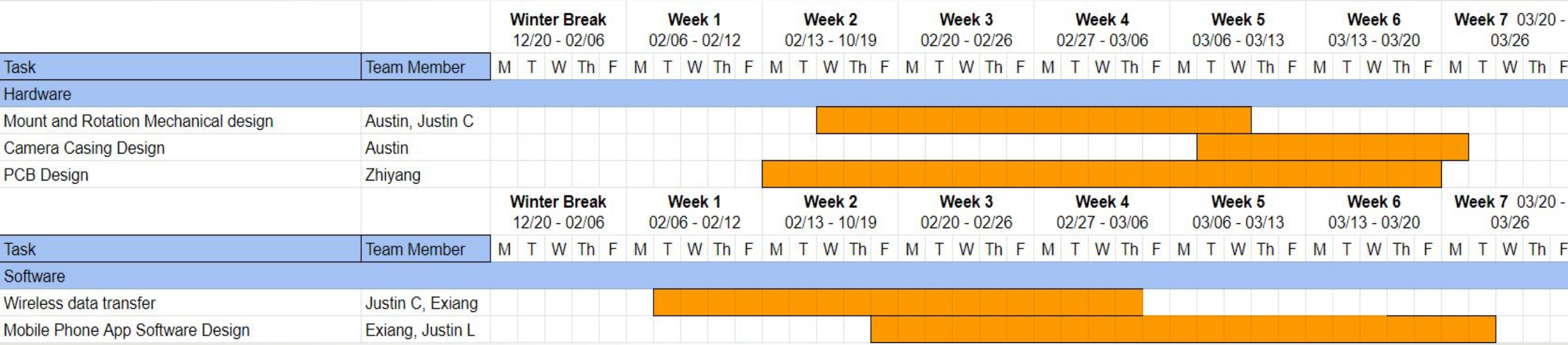
Ultrasonic distance sensor



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 mobile devices
- **Mobile App**
  - Prototype app that receives visual feed and distance data

Gantt Chart


Winter Break 12/20 - 02/06
Week 1 02/06 - 02/12
Week 2 02/13 - 10/19
Week 3 02/20 - 02/26
Week 4 02/27 - 03/06
Week 5 03/06 - 03/13
Week 6 03/13 - 03/20
Week 7 03/20 - 03/26
M T W Th F
Task
Team Member
Hardware
Mount and Rotation Mechanical design
Austin, Justin C
Camera Casing Design
Austin
PCB Design
Zhiyang
Software
Wireless data transfer
Justin C, Exiang
Mobile Phone App Software Design
Exiang, Justin L

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
• A4988 Stepper Motor Driver Carrier x6	• $15
• Adafruit NFMA-17 12V 350 mA Stepper Motor x3	• $42
• 3D Printed Parts	• $0

SDP Budget Used: $151.41
Total: ~$175

Future Parts List

• Lithium Battery	• $30
• Arducam 2MP x2	• $50
• Heavy Duty Suction Cup	• $10
• Custom PCBs	• $150
• 3D Printed Parts	• $0
• 5V Stepper Motors x3	• $15

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.

University of Massachusetts Amherst