UNIVERSITY OF ILLINOIS
URBANA-CHAMPAIGN

ECE 445 Final Project
Group 23 - Drink Dispenser Robot

Megan Cheng (mxcheng2), Andrew Jung (crjung2), Ethan Cao (ecao4)

Fall 2025

Introduction - Service Robotics

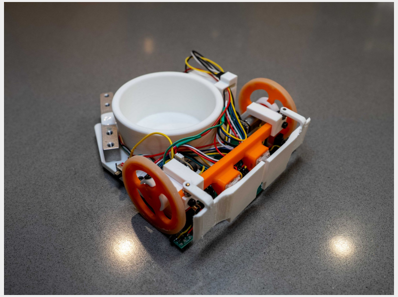
Group 23 - Drink Dispenser Robot

Overarching Design

Robot

Transports Water

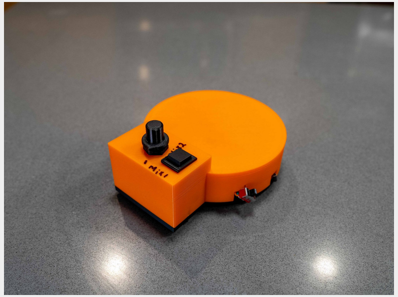
- Edge Detection
- IR Receiving
- Collision detection



Coaster

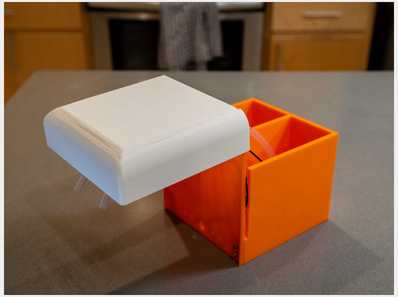
User Interface

- Button to call/send robot
- Button to start dispensing
- Dial to adjust drink Ratio



Dispenser

- 2 Pumps
- Dispenses different amounts depending on ratio

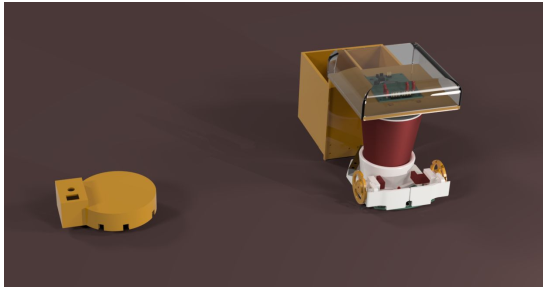


Group 23 - Drink Dispenser Robot

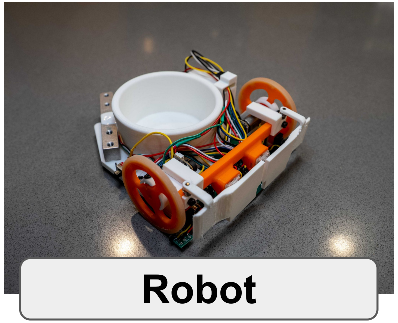
Overarching Design

Steps

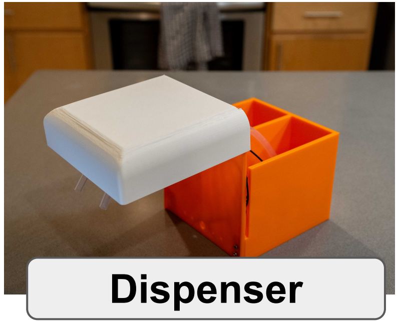
1. Press button: call robot
2. Robot arrives at coaster
3. User chooses ratio and press button
4. Robot moves/arrives at dispenser
5. Press button: Dispense ratio of liquid
6. Robot delivers drink back to coaster



Coaster



Robot

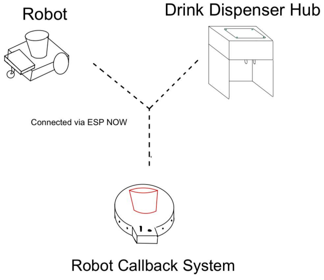


Dispenser

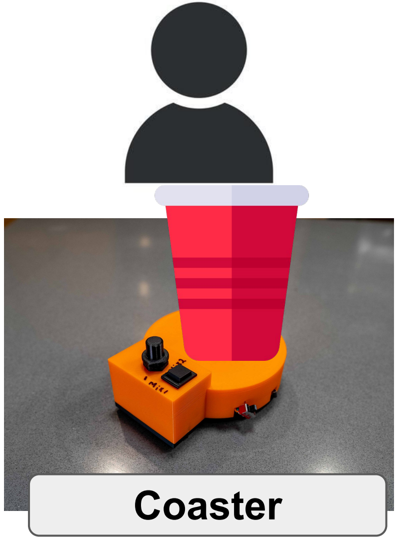
Overarching Design

Steps

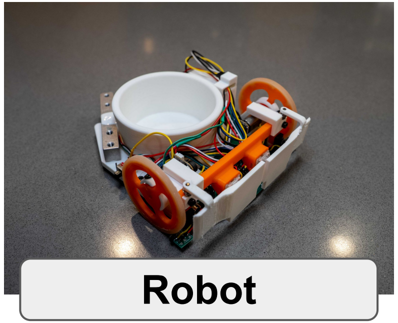
1. Press button: call robot
2. Robot arrives at coaster
3. User chooses ratio and press button
4. Robot moves/arrives at dispenser
5. Press button: Dispense ratio of liquid
6. Robot delivers drink back to coaster



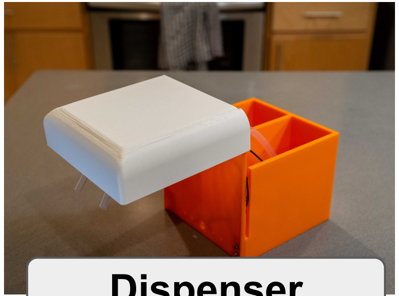
*More details below with sizing for physical design and communication protocol



Coaster



Robot



Dispenser

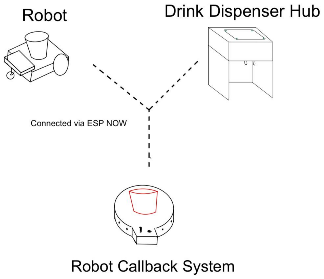
Group 23 - Drink Dispenser Robot

Overarching Design

70% water
30% extract

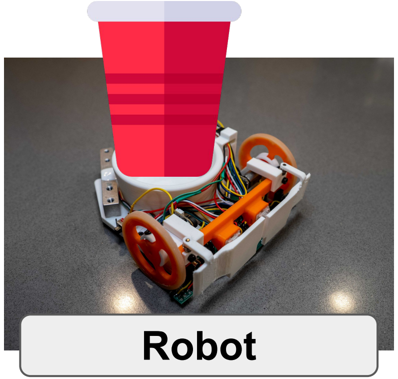
Steps

1. Press button: call robot
2. Robot arrives at coaster
3. User chooses ratio and press button
4. Robot moves/arrives at dispenser
5. Press button: Dispense ratio of liquid
6. Robot delivers drink back to coaster



*More details below with sizing for physical design and communication protocol

Coaster


Robot

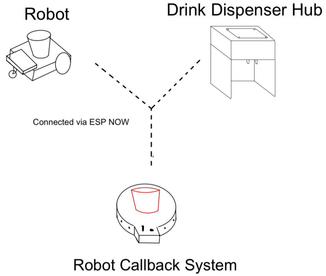
Dispenser

Group 23 - Drink Dispenser Robot

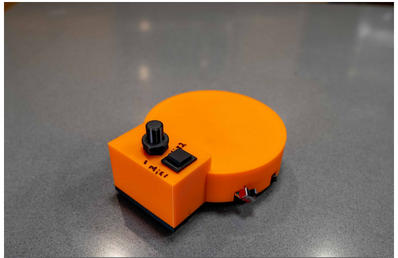
Overarching Design

Steps

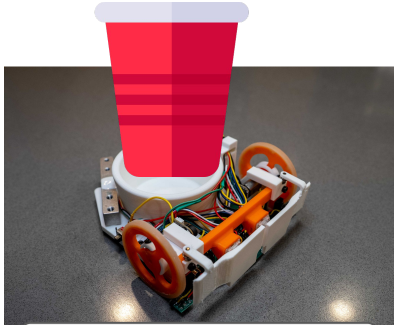
1. Press button: call robot
2. Robot arrives at coaster
3. User chooses ratio and press button
4. Robot moves/arrives at dispenser
5. Press button: Dispense ratio of liquid
6. Robot delivers drink back to coaster



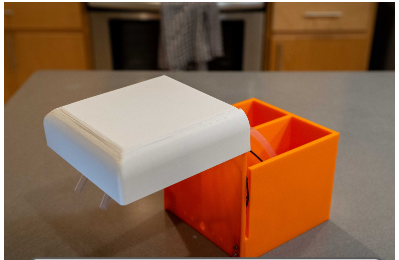
*More details below with sizing for physical design and communication protocol



Coaster



Robot



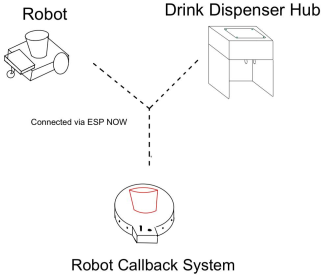
Dispenser

Group 23 - Drink Dispenser Robot

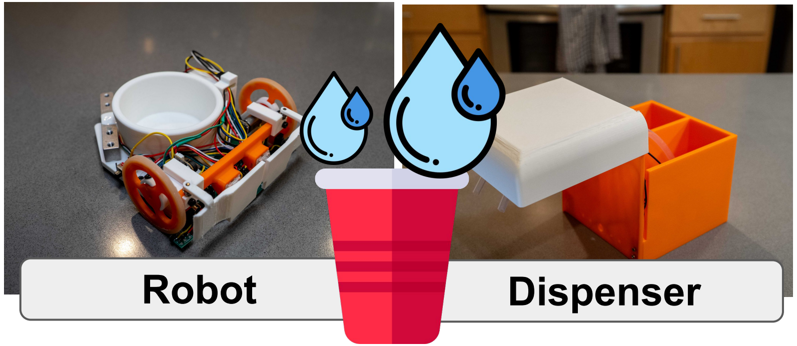
Overarching Design

Steps

1. Press button: call robot
2. Robot arrives at coaster
3. User chooses ratio and press button
4. Robot moves/arrives at dispenser
5. Press button: Dispense ratio of liquid
6. Robot delivers drink back to coaster



*More details below with sizing for physical design and communication protocol

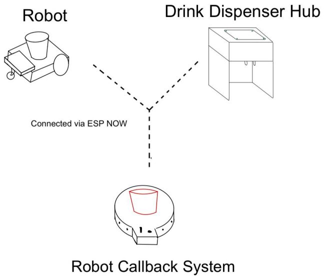


Group 23 - Drink Dispenser Robot

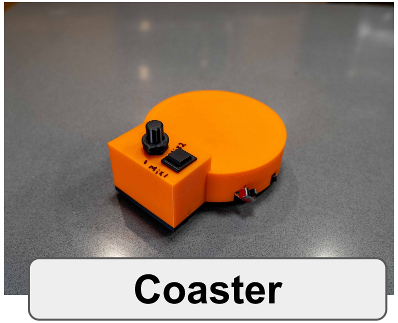
Overarching Design

Steps

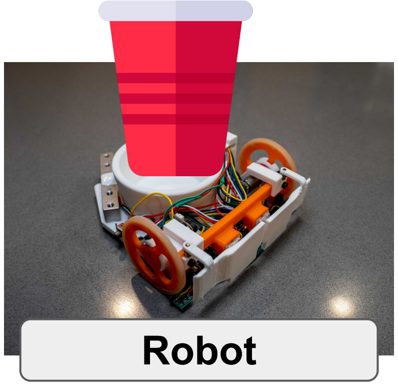
1. Press button: call robot
2. Robot arrives at coaster
3. User chooses ratio and press button
4. Robot moves/arrives at dispenser
5. Press button: Dispense ratio of liquid
6. Robot delivers drink back to coaster



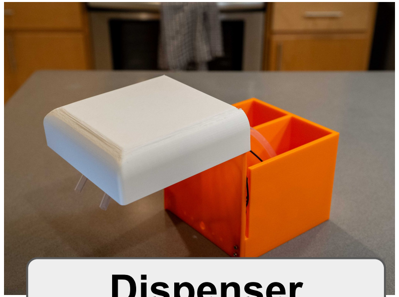
*More details below with sizing for physical design and communication protocol



Coaster



Robot



Dispenser

High-Level Requirements

- The robot will retrieve the drink within 90 seconds on a clear tabletop without any obstacles. The introduction of obstacles shall not inhibit functionality but may increase service time.
- The dispenser shall be able to fill a standard red solo cup. No more than 1 mL of liquid will be spilled on the table for every 5 drink retrieval cycles.
- The robot will not get stuck on the dispensing station/coaster or fall off of the table.

Ethical Concerns

Shorting Hazards

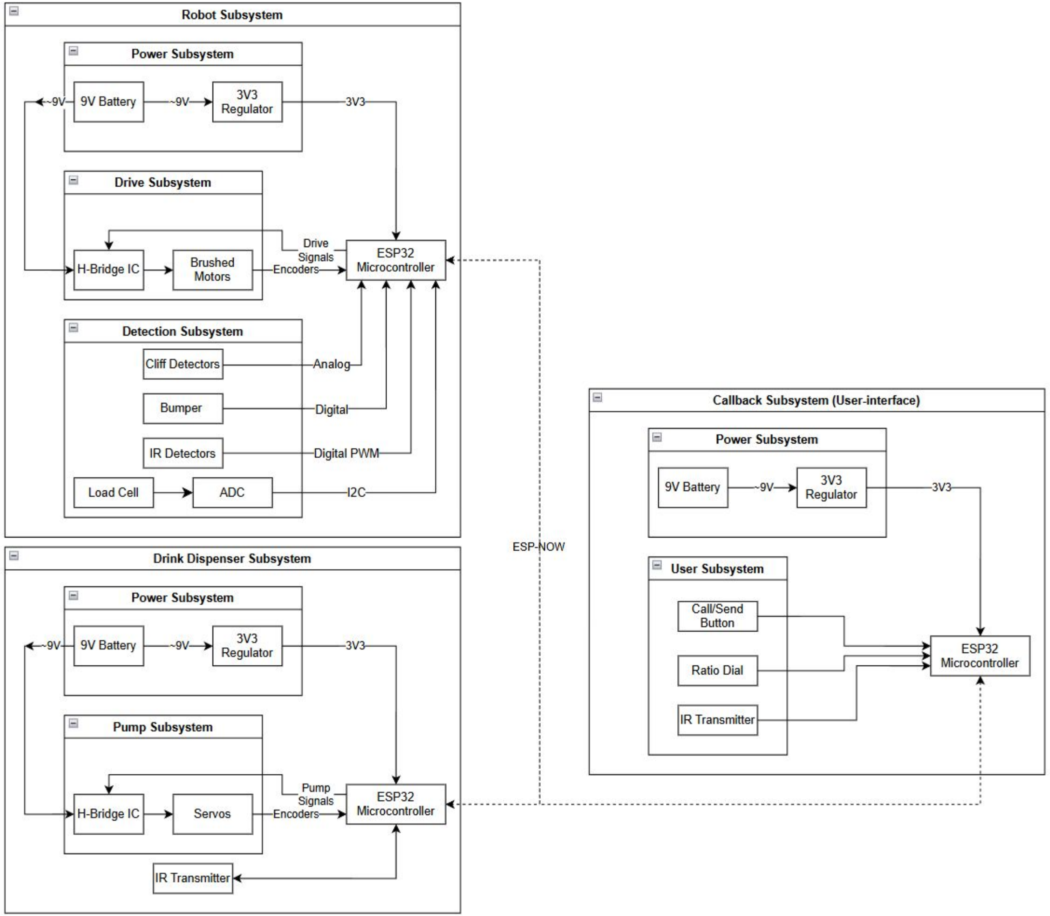
- Elevating PCBs as highest point
- Separate liquid storage
- 3D Printed Elements to protect sensors and PCBs
- Avoiding higher voltage/wall power

Food Safety

- Proper coating for food safe materials

Group 23 - Drink Dispenser Robot

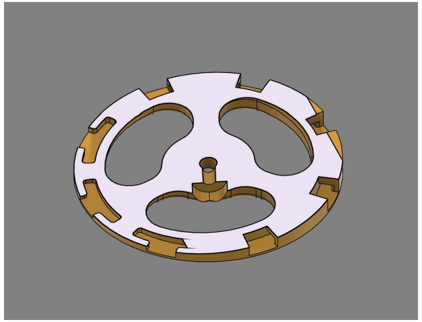
Design Overview

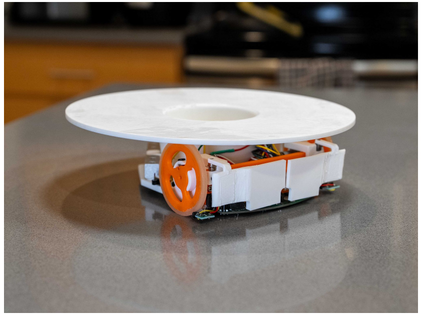


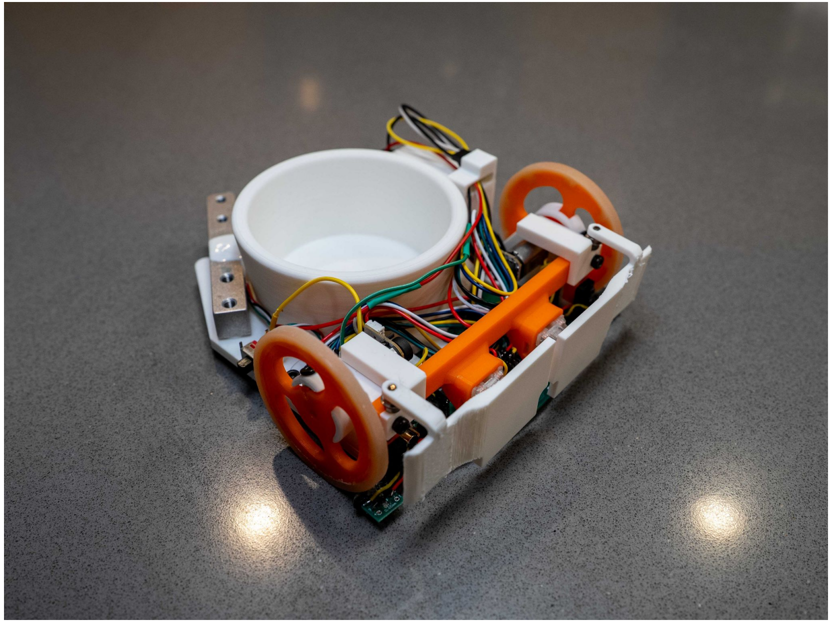
Robot Subsystem

Robot Hardware - Mechanical Design

- Cast urethane wheels
- Integrated cup holder
- Stabilized Bumpers
- Safety Hat

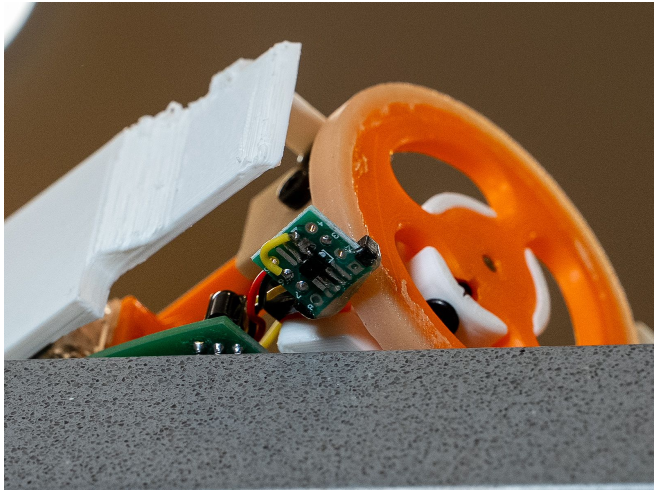






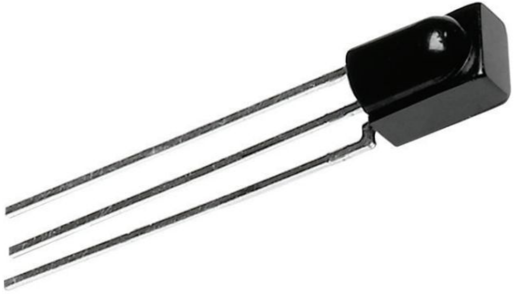
Robot Hardware - Cliff Detection

- Used to not fall off the table
- Packaged IR emitter and Receiver
- Common line follow sensor
- Mounted directly in front of wheels
- Pros: Cheap, easy to use
- Con: Does not work on dark surfaces

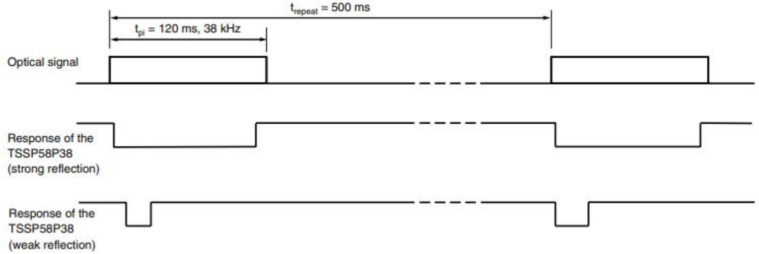


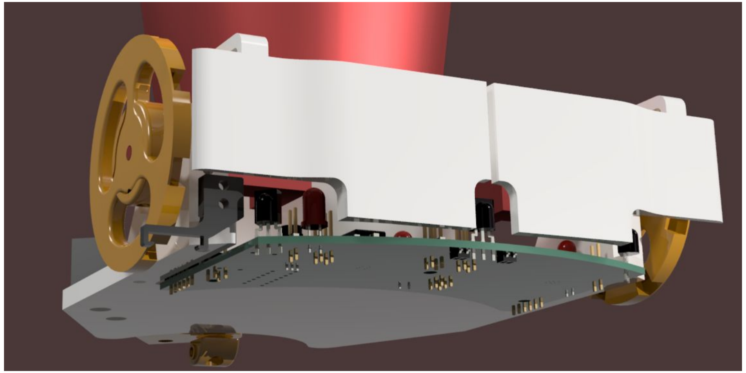
Robot Hardware - IR Detection

- Used to locate coaster and dispenser
- Commonly found on TVs
- Uses a pulsed, modulated signal
- Expresses light intensity using pulse width
- Read using hardware interrupts



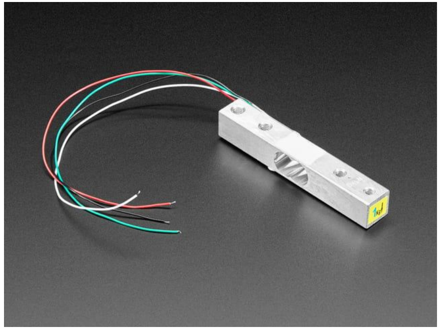
Example of a signal pattern:

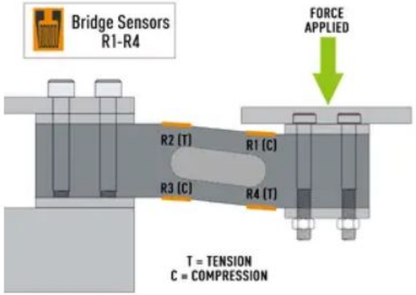

t_{repeat} = 500 ms
t_{pi} = 120 ms, 38 kHz
Optical signal
Response of the TSSP58P38 (strong reflection)
Response of the TSSP58P38 (weak reflection)

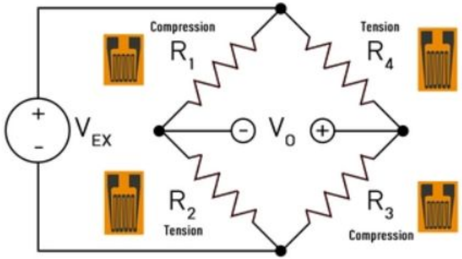


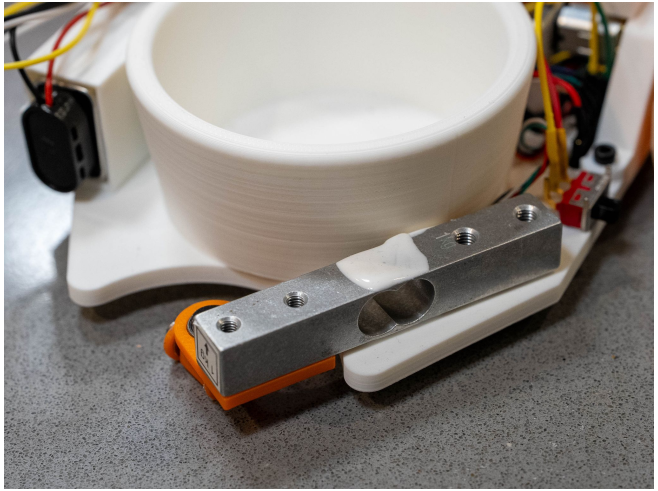
Robot Hardware - Load Cell

- Used to determine cup fill level
- Measures deflection in a beam
- Produces small voltage changes
- Output read using high precision ADC



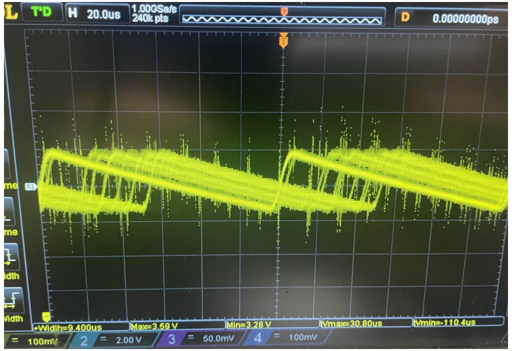


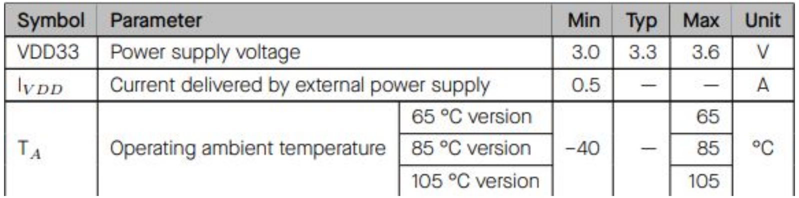




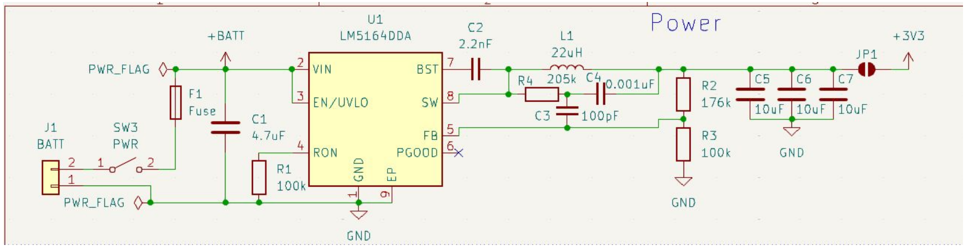
Robot Hardware - Power Supply

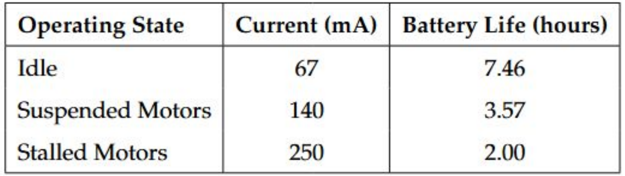
- Switching Regulator
- Outputs 3.3V (3.28V - 3.59V)
- Ripple is within Microcontroller Range
- Fuse for OC Protection





Symbol	Parameter		Min	Typ	Max	Unit
VDD33	Power supply voltage		3.0	3.3	3.6	V
I_{VDD}	Current delivered by external power supply		0.5	—	—	A
T_A	Operating ambient temperature	65 °C version	−40	—	65	°C
		85 °C version			85	
		105 °C version			105	



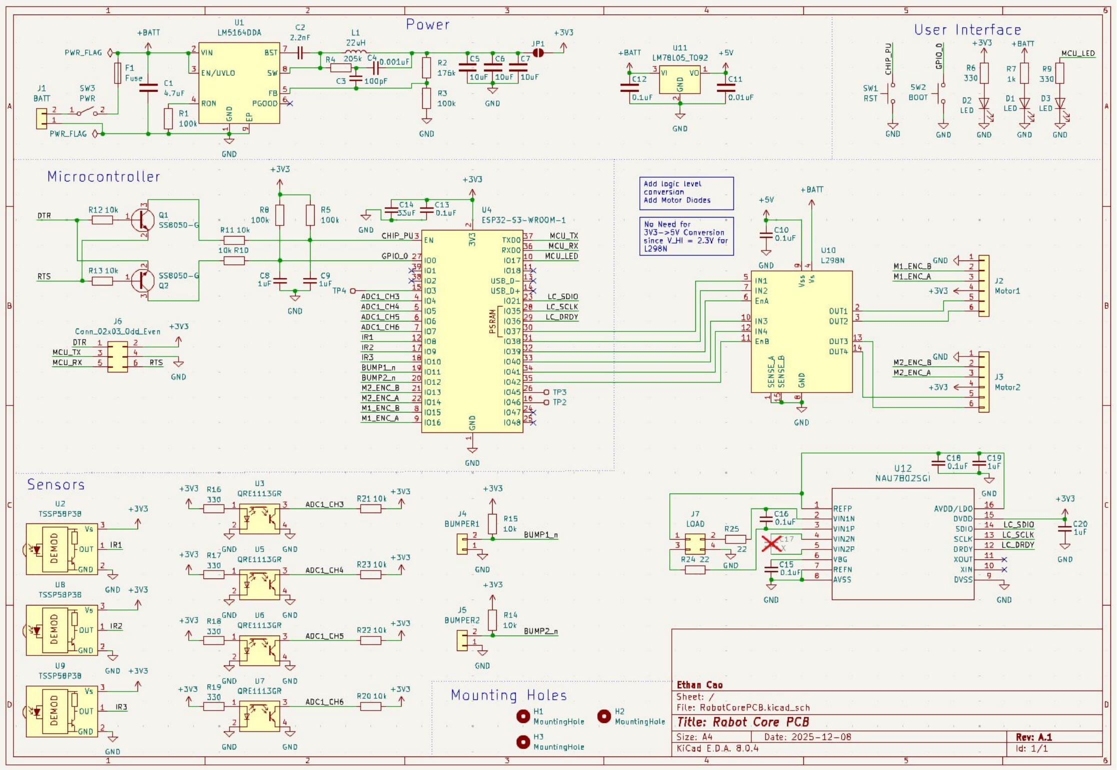


Operating State	Current (mA)	Battery Life (hours)
Idle	67	7.46
Suspended Motors	140	3.57
Stalled Motors	250	2.00

Table 1: Current Consumption and Estimated Battery Life for a 500 mAh Battery

Robot Hardware - PCB

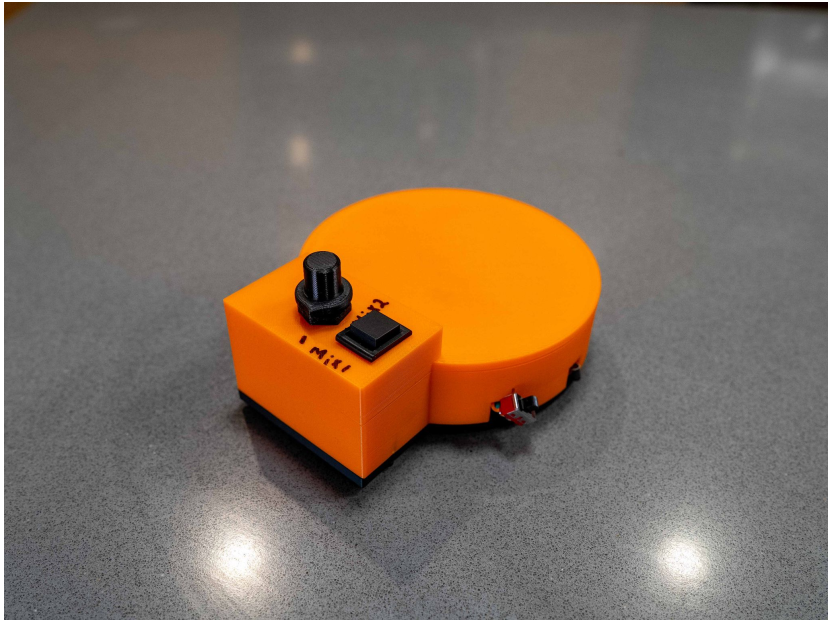
- Switching regulator
- ESP32-S3
- H-Bridge Motor Driver
- IR Detectors
- Cliff Sensors
- Load Cell ADC



Coaster Subsystem

Coaster Hardware - Mechanical Design

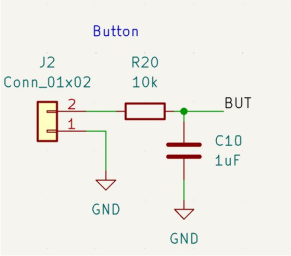
- Radially spread out IR LEDs
- 3D printed covered potentiometer for dial
- Tactile switch for call/send function
- Circular surface to place drinks
- Optimally compact

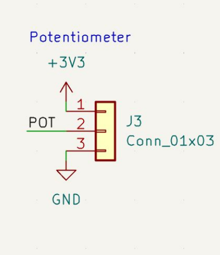


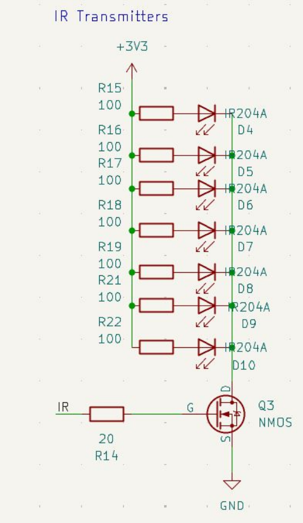
Group 23 - Drink Dispenser Robot

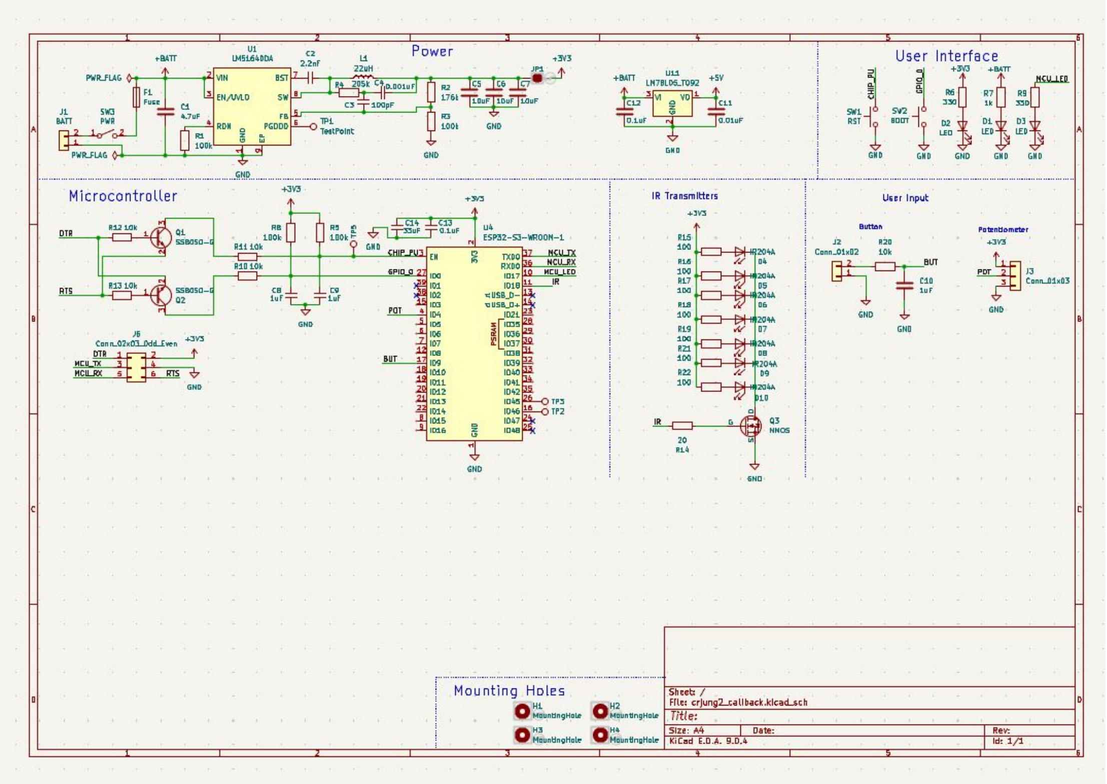
Coaster Hardware - PCB

- 7 IR Transmitters
- Potentiometer
- Button





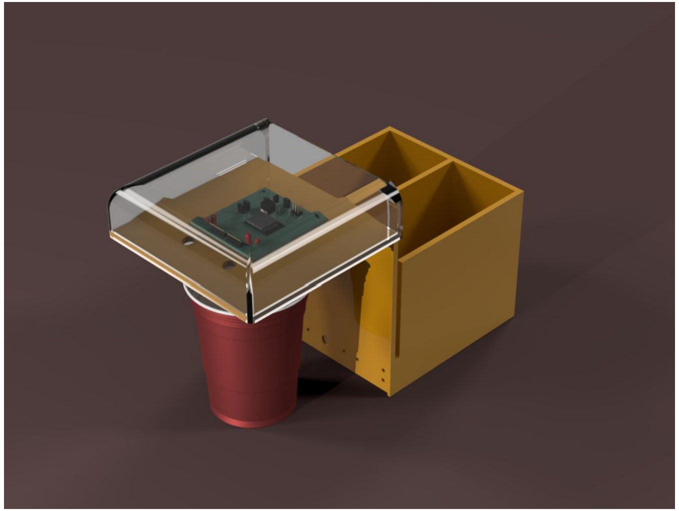




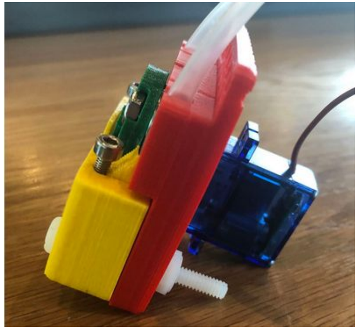
Dispenser Subsystem

Dispenser Hardware - Mechanical

- Liquid Storage
- Submersible Pump
- IR LED
- Water Short Protection



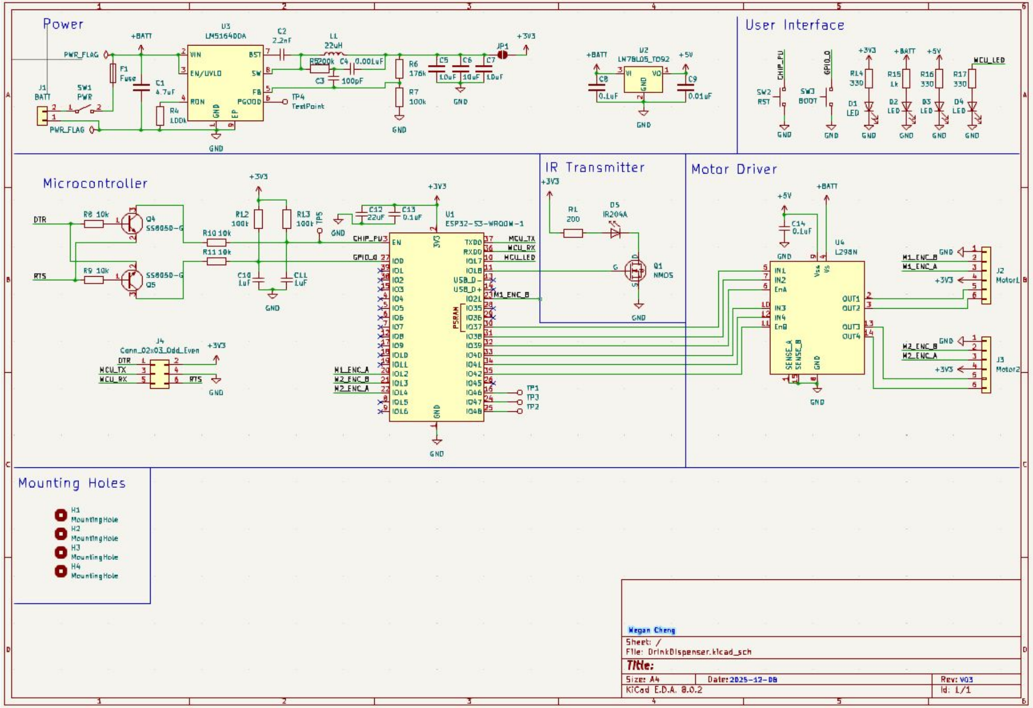
Previous Iteration



Current Iteration

Dispenser Hardware - PCB

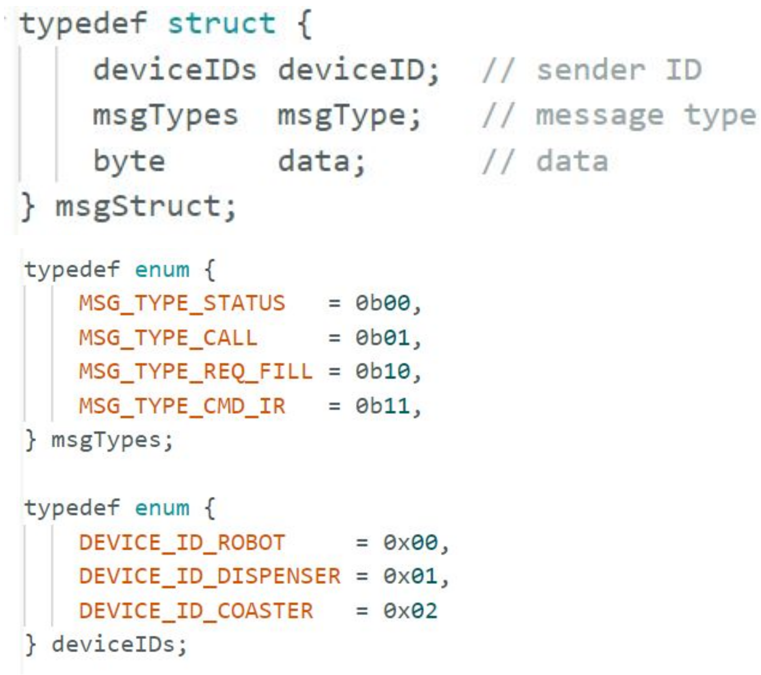
- ESP32-S3
- Pump + H-Bridge Motor Drivers
- IR Receiver



Software

Overall Software - ESP NOW Communication

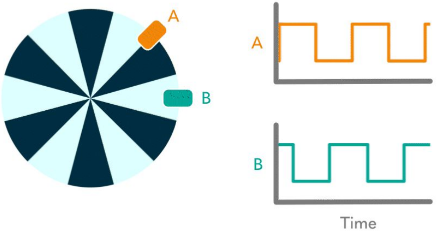
- Custom Message Protocol
  - 2-byte frame:
  - [2 bits SenderID | 2 bits MsgType | 1 byte Data]
  - Devices: Robot (0), Dispenser (1), Coaster (2)
- Key Message Types
  - STATUS: device state, IR state, location info
  - CALL: robot movement commands (0x00, 0x01, 0x02)
  - REQ_FILL: drink ratio (4-bit) sent from coaster → dispenser
  - CMD_IR: signals IR-beacon start (coaster/dispenser)
- All sent via broadcast to FF:FF:FF:FF:FF:FF

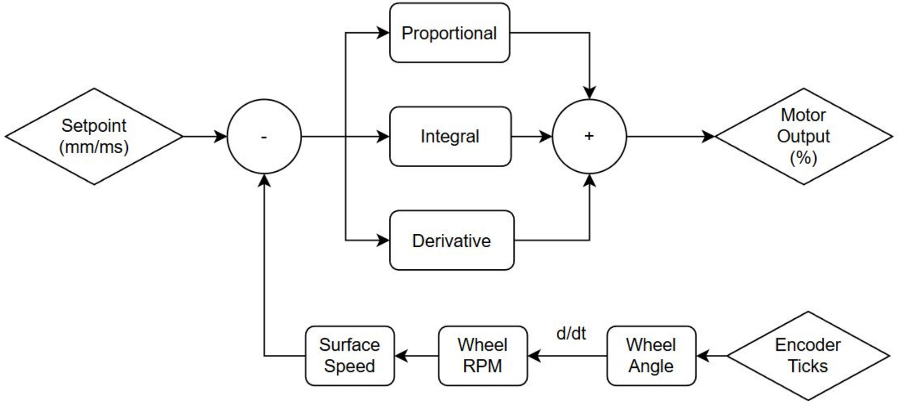


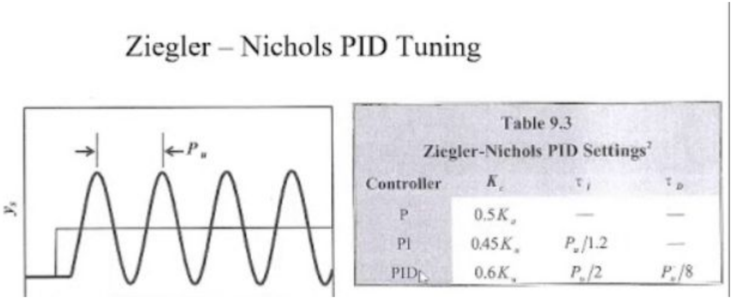
```
typedef struct {
    deviceIDs deviceID;   // sender ID
    msgTypes  msgType;    // message type
    byte      data;       // data
} msgStruct;

typedef enum {
    MSG_TYPE_STATUS   = 0b00,
    MSG_TYPE_CALL     = 0b01,
    MSG_TYPE_REQ_FILL = 0b10,
    MSG_TYPE_CMD_IR   = 0b11,
} msgTypes;

typedef enum {
    DEVICE_ID_ROBOT     = 0x00,
    DEVICE_ID_DISPENSER = 0x01,
    DEVICE_ID_COASTER   = 0x02
} deviceIDs;
```

Robot Software - Wheel PID Turning

- Wheel position by counting encoder ticks
- Wheel velocity using derivative
- Velocity controlled with PID loop
- Allows precise velocity control regardless of environment







Ziegler – Nichols PID Tuning

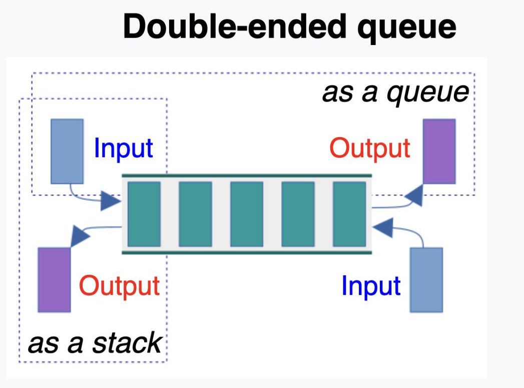
Table 9.3

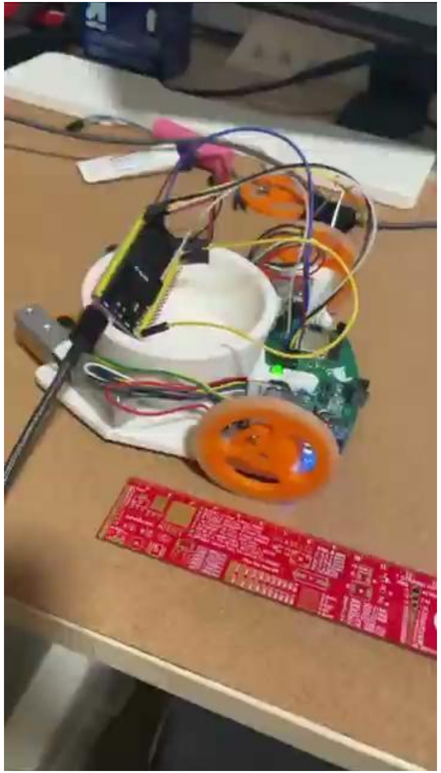
Ziegler-Nichols PID Settings[2]

Controller	K_c	τ_I	τ_D
P	$0.5K_u$	—	—
PI	$0.45K_u$	$P_u/1.2$	—
PID	$0.6K_u$	$P_u/2$	$P_u/8$

Robot Software - Motion

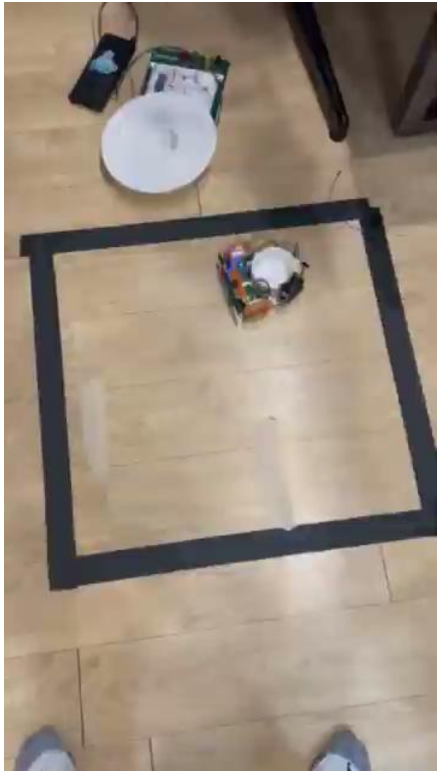
- Add paths to queue
- Robot executes one by one
- Immediate motions can be pushed to front of queue





Robot Software - Cliff Detection

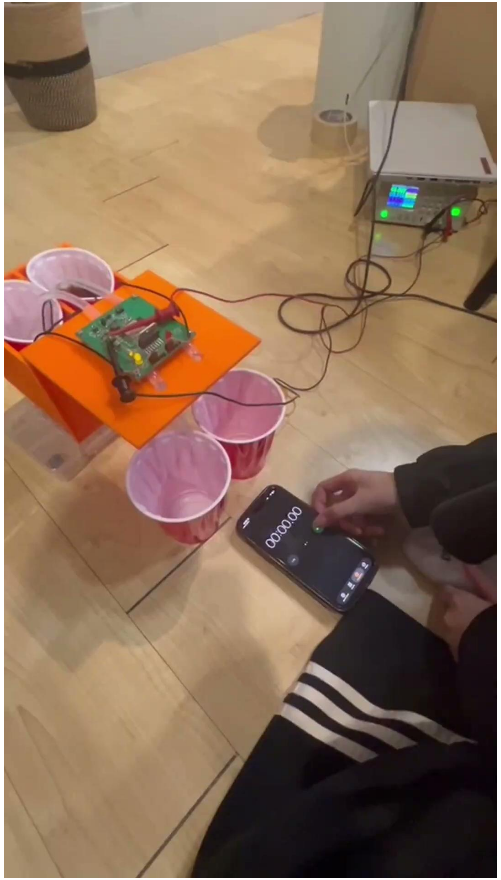
- Robot drives in a straight line
- Executes reverse + turn at the rising edge of cliff sensor



Dispenser Software

- Pump ratio
    - Time-based (10 sec)
    - Delay movement (spill protection)
    - Average liquid weight 167-170g per cup
- Share status

Dispenser Status	bit[1:0]
DISP_STATUS_IDLE	0
DISP_STATUS_WAITING	1
DISP_STATUS_DISPENSING	2



Coaster Software

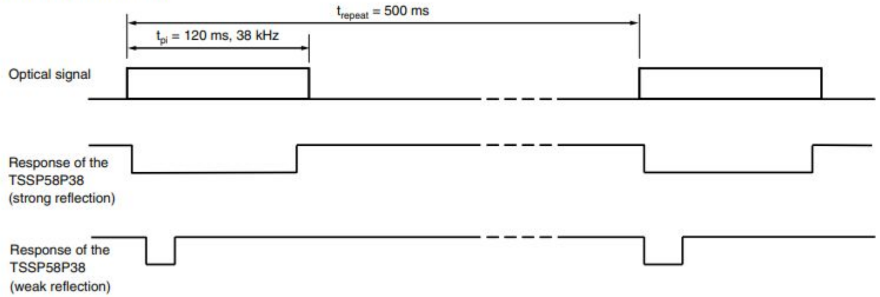
- Button FSM
  - 0 on startup, 1 & 2 repeated (e.g. 0 1 2 1 2 …)
- Button Handling
  - 50 ms software debouncing
  - Latched press detection (no double-triggers)
- Ratio Selection (4-bit)
  - ADC (0–4095)
  - Mapped to ratio 0–15
- IR Transmission
  - LEDC PWM at 38 kHz
  - 40 ms ON / 60 ms OFF IR pulsing

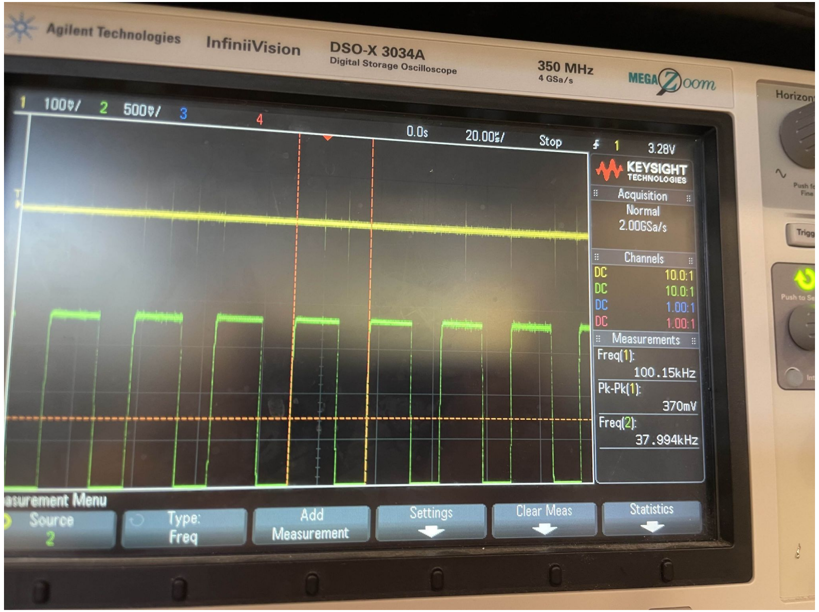
Button Status bit[1:0]	bit[1:0]
0	First press calls robot to coaster (only once on startup)
1	Sends robot to dispenser Communicate ratio value
2	Dispense sequence Wait 4 Seconds Stop dispense sequence Robot Back to coaster

R&V - IR Sensors

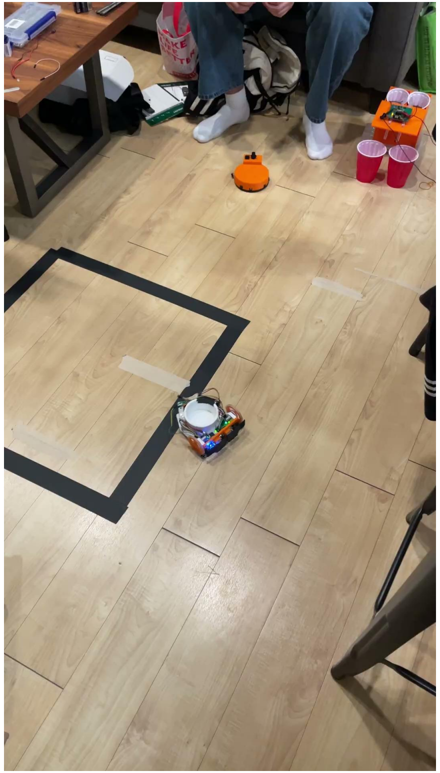
- Modulated 38kHz output from IR LED
- 40 ms ON / 60 ms OFF IR pulsing

Example of a signal pattern:


t_{repeat} = 500 ms
t_{pi} = 120 ms, 38 kHz
Optical signal
Response of the TSSP58P38 (strong reflection)
Response of the TSSP58P38 (weak reflection)



Integration - IR Tracking



Full Demo + Successes

- 38 kHz Modulation across multiple IR transmitters
- Robot movement
- Cliff Detection
- Wireless Communication
- Coaster Control
- Pump Ratios
- Stable Power Source

Challenges

- Pump Design Iterations
- IR Accuracy
  - Low dynamic range
  - Hard to determine directions
  - IR Interference
  - IR Strength tuning

Conclusions

Search Algorithm

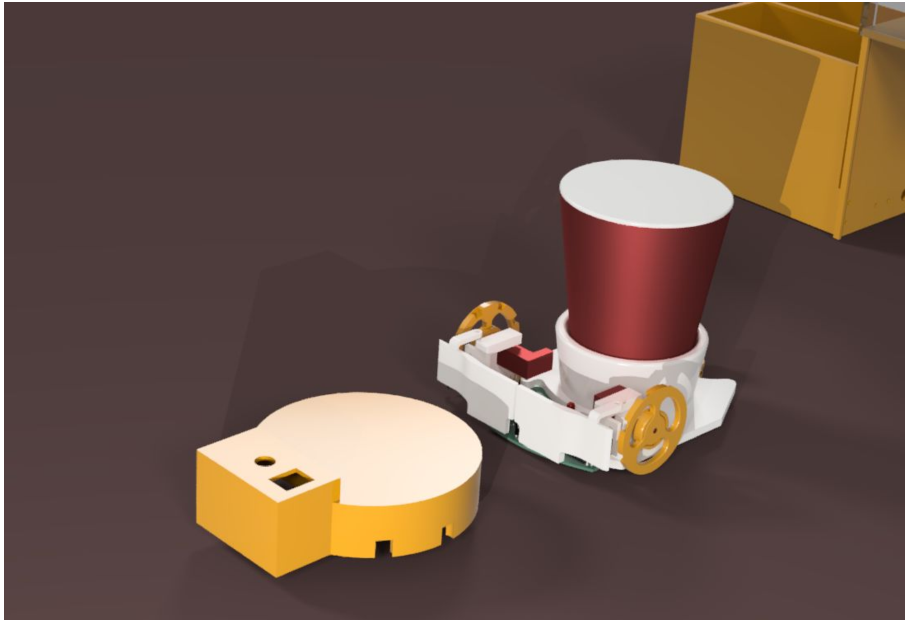
Integration w/ Wireless Queue

Pump Prototyping

IR Accuracy

Future Work

- Other methods of location finding
  - Accuracy and closer range
- Additional coaster/call station integration
- Switching to robot over to a RTOS
  - Timing accuracy



Questions?

The Grainger College of Engineering

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN