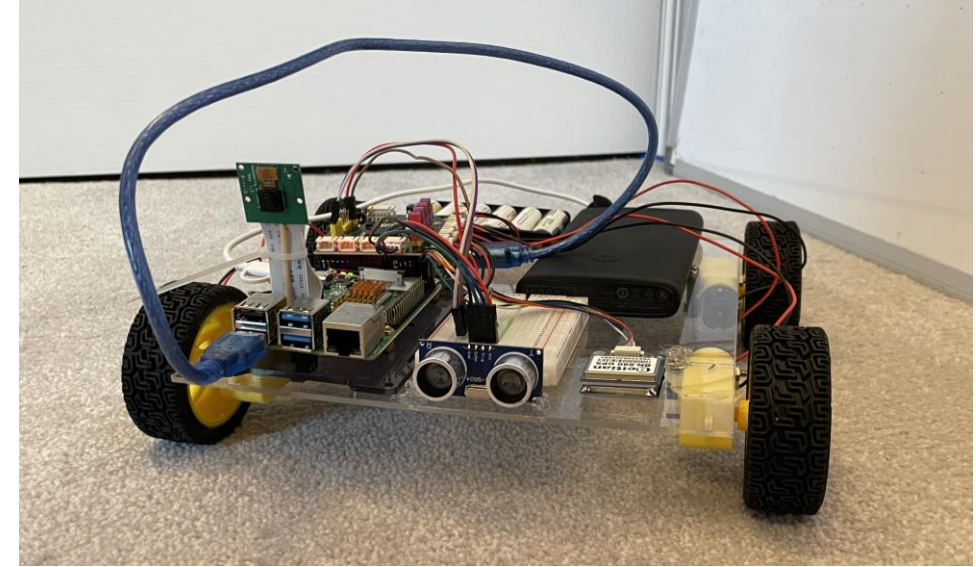


Contactless Delivery Robot

Boost the Business | Save Manual Work | Restrain the Spread of Virus

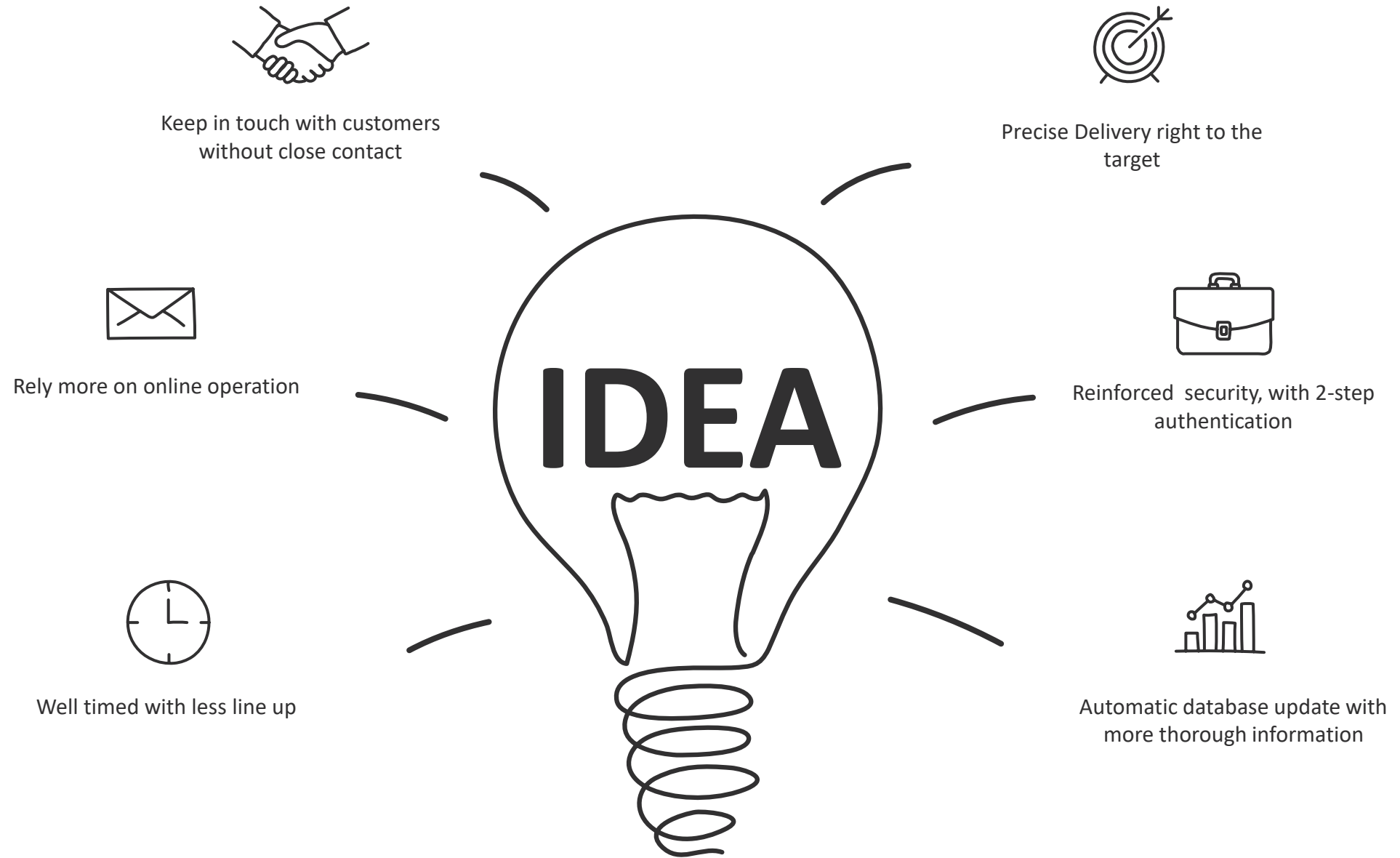


Group Member:

Zhuoxin Ma | Yuchen Miao | Lang Sun | Yichen Xiao

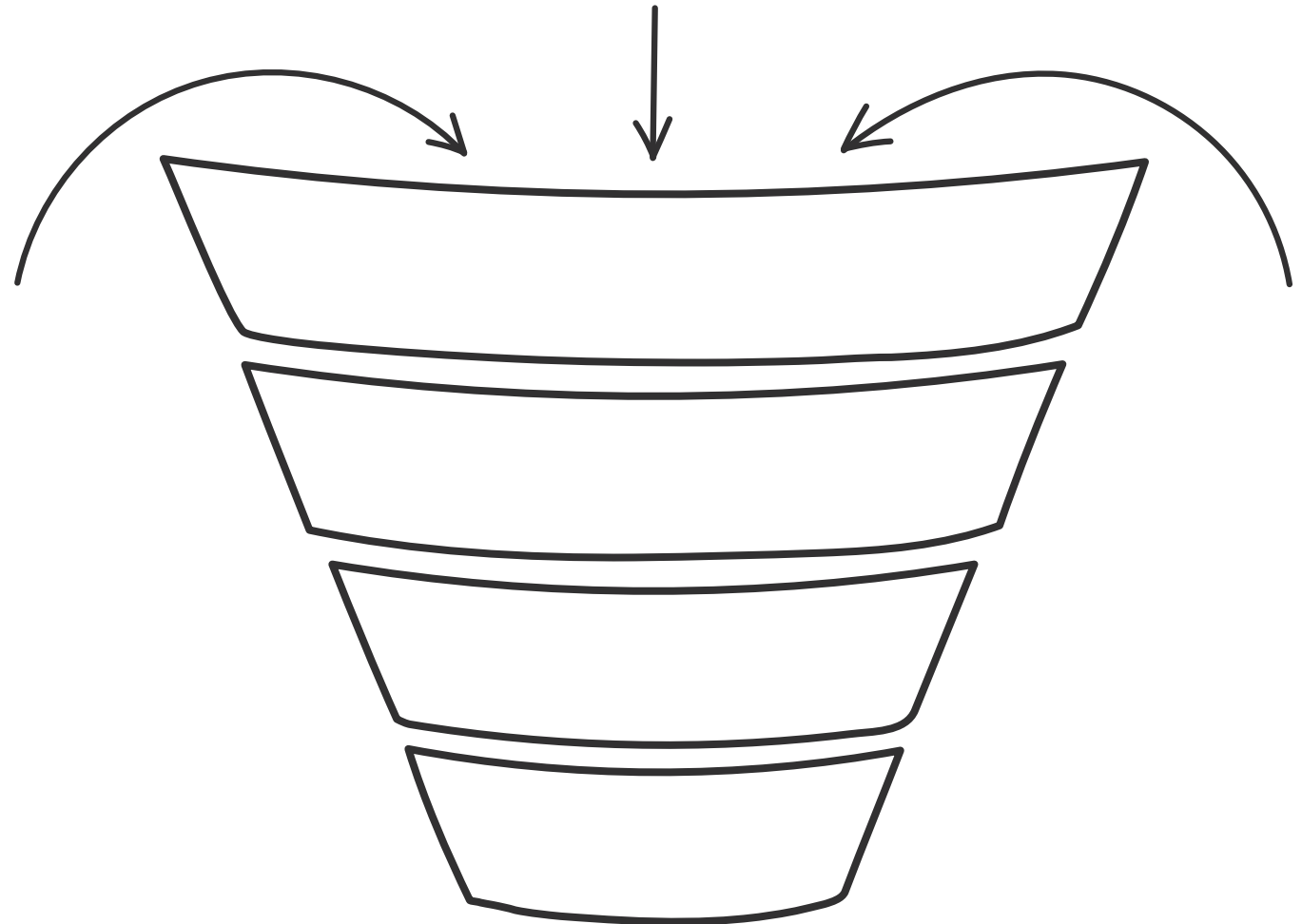
Supervisor: Professor P.X. Liu



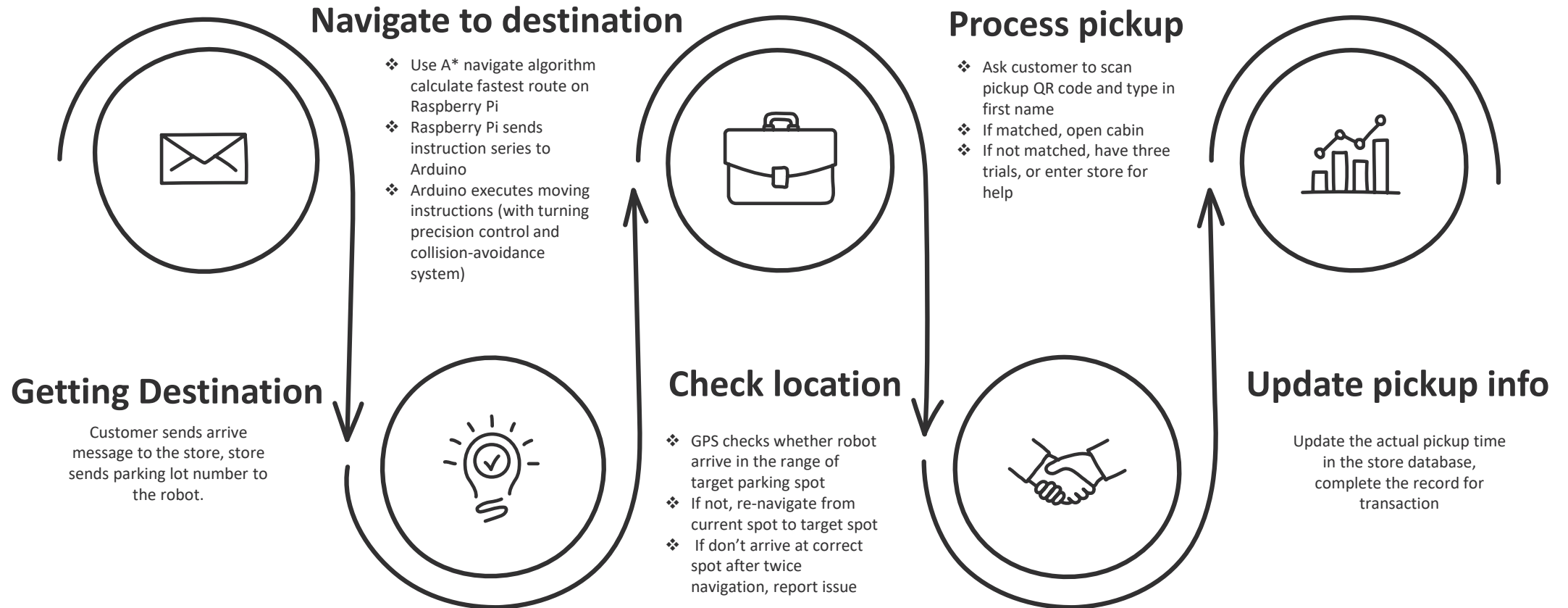


Assumptions

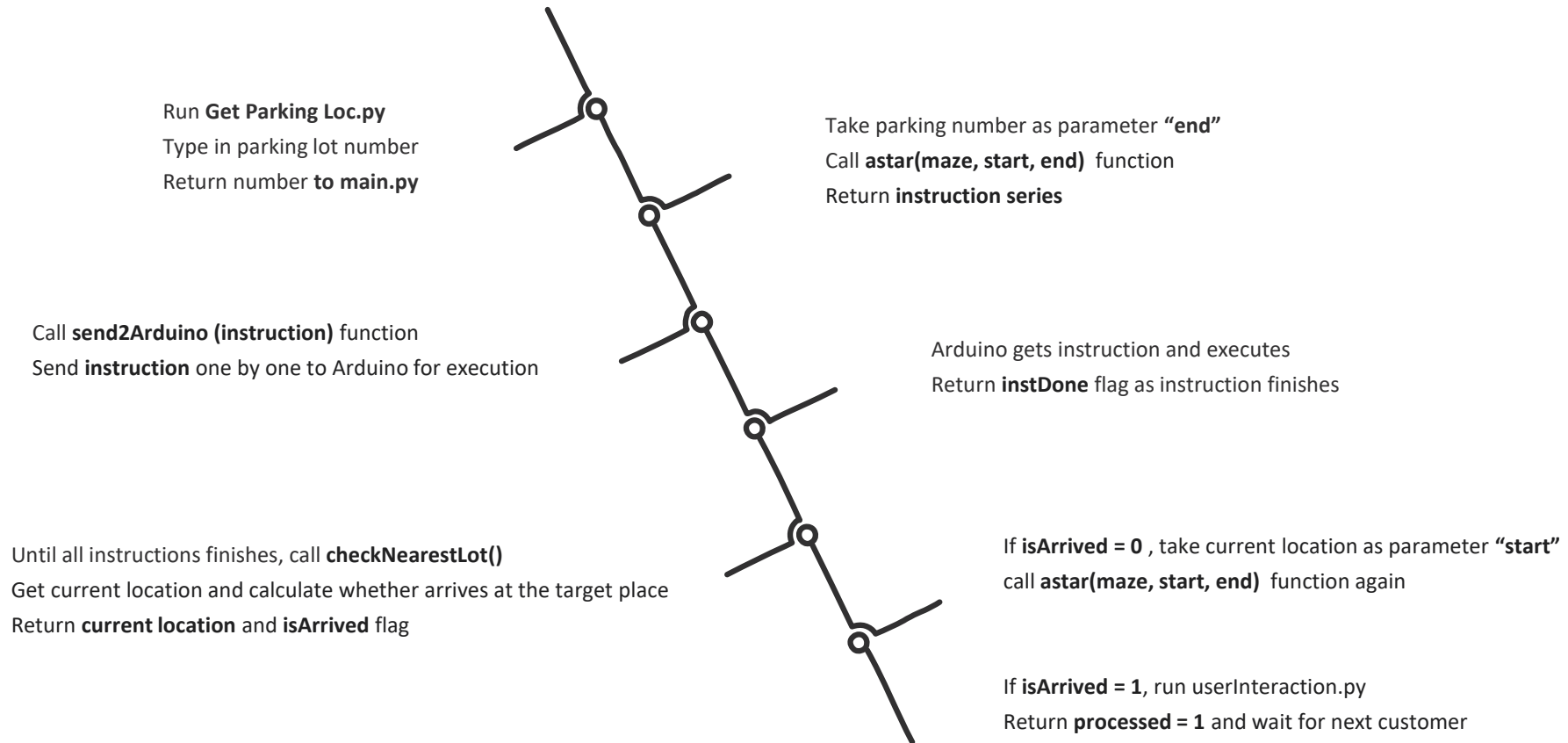
- The robot we design is used as the “**carrier engine**” or the “**control centre**” for the **delivery machine**. Products are preloaded in the cabins carried or dragged by the robot.
- All the parking space are set as **static obstacles** (the collision-avoidance system only deals with **moving obstacles or dynamic obstacles**)
- The store is using **scheduled pick-up** mode, assuming allocating 6 pickups during each 15 minutes. Products are **preloaded** in the cabins carried by the robot.
- The map of parking lot is **preloaded** in the navigation system.
- The parking lot is **completely open-air**. GPS signal will not be affected or blocked.
- The QR code for product pickup is issued and sent to the customer prior to scheduled pick up time.



Work Flow



Coding Flow



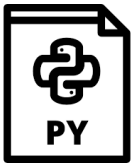
Getting Parking Lot Number

— Zhuoxin Ma



Message

| Scan QR code |
| Enter parking lot number |
| send out message |



Python Script

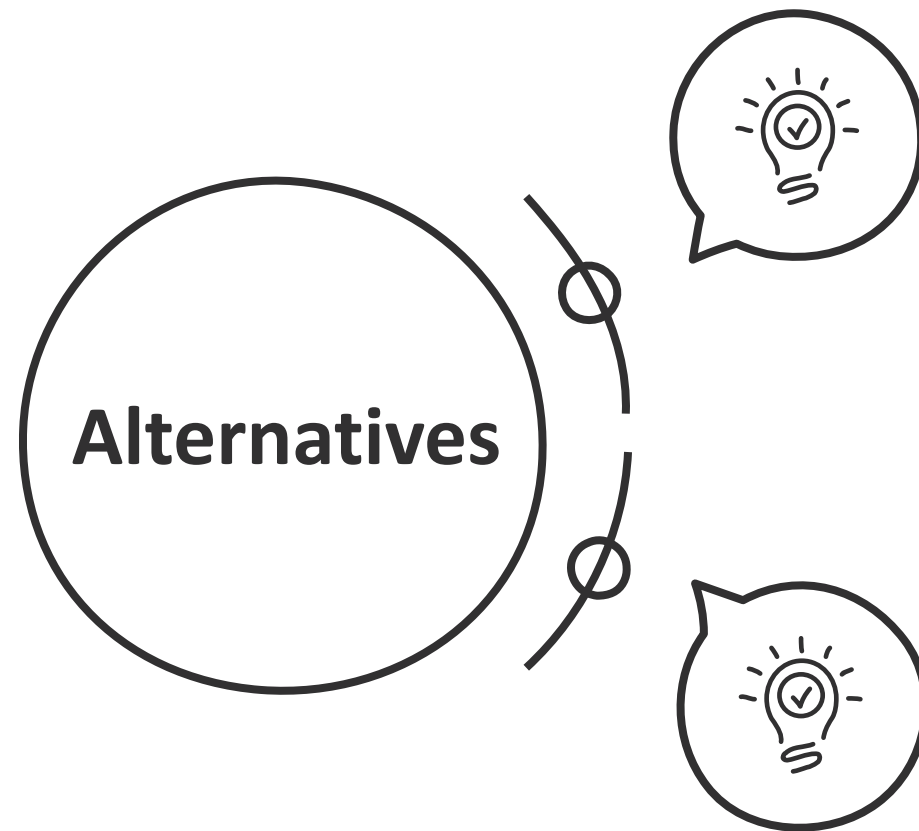
| staff in store enter parking lot number received |
| pass parking lot number to main.py as destination |



```
1 #send parking lot number to pi
2 def get_parking_loc():
3     loc = input("Enter customer's parking location: ")
4     return loc
```

```
1 parking_loc = get_parking_loc();
2 print(parking_loc)
```

```
Enter customer's parking location: 6
6
```



Python directly receive SMS message

- Get an SMS enabled Plivo phone number
- Create a Flask application to handle incoming SMS message (using Flask and Plivo python packages)
- Extract parking lot number directly from the message received
- Reference: <https://www.plivo.com/blog/receive-respond-sms-python-flask-plivo/>

Interact with a Web

- Create a simple webpage to allow user type in their parking lot number
- Generate a QR code for the URL of the webpage

Navigate to Destination

Lang Sun (Raspberry Pi)
Zhuoxin Ma (Arduino)
Yuchen Miao (Data Transfer)
Yichen Xiao (Turning Precision)

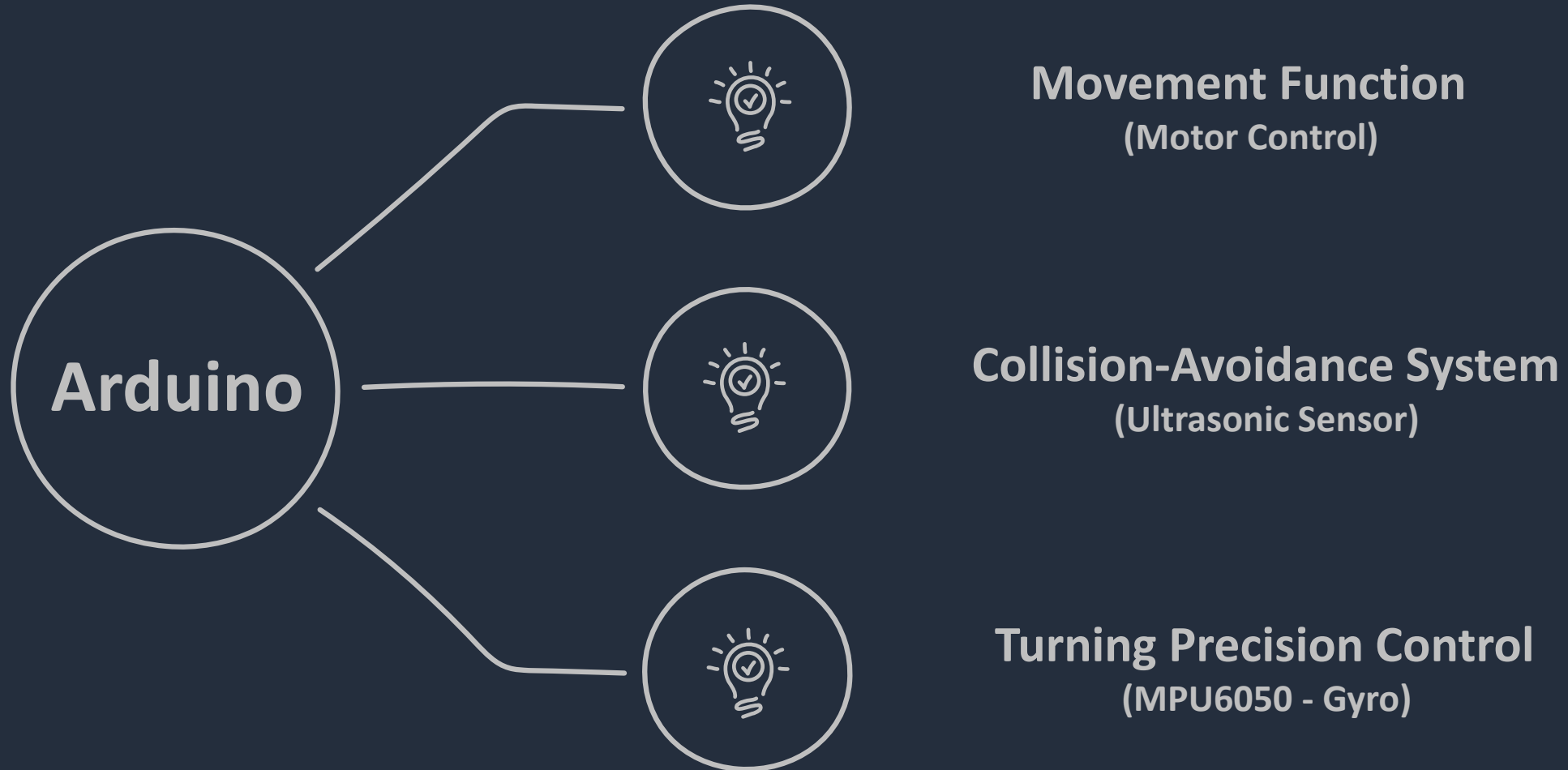
Data Transfer Between Pi and Arduino

 Yuchen Miao

Arduino Execution

Zhuoxin Ma
Yichen Xiao

Arduino Execution



Movement Function

Pin Definition

```
//pin definition for motor
int ML_Ctrl = 4;    //define the direction control pin of B motor
int ML_PWM = 5;    //define the PWM control pin of B motor
int MR_Ctrl = 2;    //define the direction control pin of A motor
int MR_PWM = 9;    //define the PWM control pin of A motor
```

Setup

```
Serial.begin (9600);

//motor pin setup
pinMode(ML_Ctrl, OUTPUT); //Right side motor direction
pinMode(ML_PWM, OUTPUT); //Right side motor direction
pinMode(MR_Ctrl, OUTPUT); //Left side motor direction
pinMode(MR_PWM, OUTPUT); //Left side motor speed
```

Forward Function

```
void forward(){ // move forward
    ultrasonic_avoid();
    digitalWrite(ML_Ctrl, HIGH);
    analogWrite(ML_PWM, 255);
    digitalWrite(MR_Ctrl, LOW);
    analogWrite(MR_PWM, 175);
    delay(move_time);
}
```

Stop Function

```
void stop(){ //stop
    digitalWrite(ML_Ctrl, HIGH);
    analogWrite(ML_PWM, 0);
    digitalWrite(MR_Ctrl, HIGH);
    analogWrite(MR_PWM, 0);
    delay(200);
}
```

Backward Function

```
void backward(){ // turn around
    leftturn();
    leftturn();
}
```

Movement Function

Left turn Function

```
void leftturn(){ // turn left
    turningAngle=0;
    while(1){
        timer = millis();

        // Read normalized values
        Vector norm = mpu.readNormalizeGyro();
        turningAngle = turningAngle + norm.XAxis * timeStep;
        delay((timeStep*1000) - (millis() - timer));
        digitalWrite(ML_Ctrl, HIGH);
        analogWrite(ML_PWM, testSpeed2);
        digitalWrite(MR_Ctrl, HIGH);
        analogWrite(MR_PWM, testSpeed2);
        //Serial.println(turningAngle);
        if(turningAngle<86&&turningAngle>=0){
            digitalWrite(ML_Ctrl, HIGH);
            analogWrite(ML_PWM, testSpeed2);
            digitalWrite(MR_Ctrl, HIGH);
            analogWrite(MR_PWM, testSpeed2);
        }else if (turningAngle>94){
            digitalWrite(ML_Ctrl, LOW);
            analogWrite(ML_PWM, testSpeed1);
            digitalWrite(MR_Ctrl, LOW);
            analogWrite(MR_PWM, testSpeed1);
        }else{
            digitalWrite(ML_Ctrl, HIGH);
            analogWrite(ML_PWM, 0);
            digitalWrite(MR_Ctrl, HIGH);
            analogWrite(MR_PWM, 0);
            break;
        }
    }
}
```

Right turn Function

```
void rightturn(){ // turn right
    turningAngle=0;
    while(1){
        timer = millis();

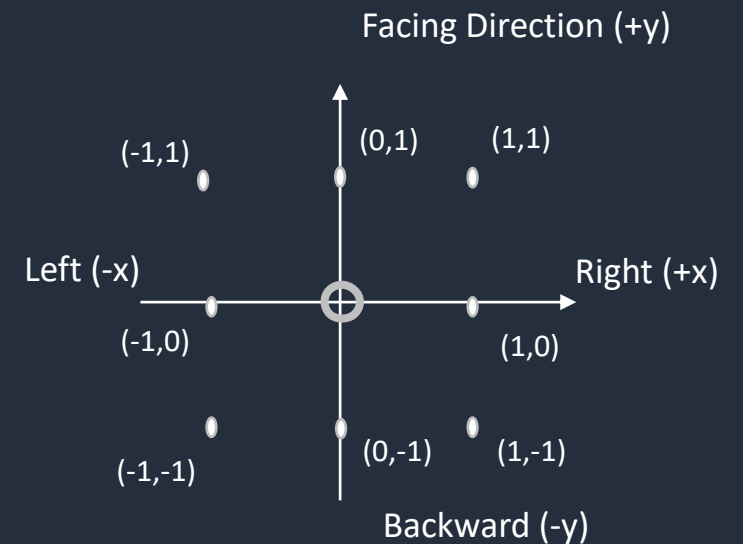
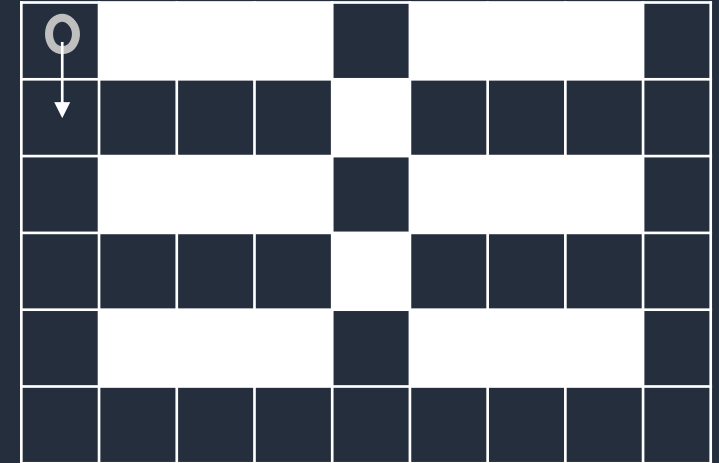
        // Read normalized values
        Vector norm = mpu.readNormalizeGyro();
        turningAngle = turningAngle + norm.XAxis * timeStep;
        delay((timeStep*1000) - (millis() - timer));
        digitalWrite(ML_Ctrl, LOW);
        analogWrite(ML_PWM, testSpeed1);
        digitalWrite(MR_Ctrl, LOW);
        analogWrite(MR_PWM, testSpeed1);
        //Serial.println(turningAngle);

        if(turningAngle>-88&&turningAngle<=0){
            digitalWrite(ML_Ctrl, LOW);
            analogWrite(ML_PWM, testSpeed1);
            digitalWrite(MR_Ctrl, LOW);
            analogWrite(MR_PWM, testSpeed1);
        }else if (turningAngle<-92){
            digitalWrite(ML_Ctrl, HIGH);
            analogWrite(ML_PWM, testSpeed1);
            digitalWrite(MR_Ctrl, HIGH);
            analogWrite(MR_PWM, testSpeed1);
        }else{
            digitalWrite(ML_Ctrl, HIGH);
            analogWrite(ML_PWM, 0);
            digitalWrite(MR_Ctrl, HIGH);
            analogWrite(MR_PWM, 0);
            break;
        }
    }
}
```

Movement Function

Execution Function

```
//next section defines how arduino execute instruction sent from pi
void execute_move(int x1, int y1){ // execute the move instruction from raspberry pi
    if (x1 == -1){
        if (y1 == -1){
            stop(); leftturn(); stop(); forward();
            stop(); leftturn(); stop(); forward();
            stop(); backward(); stop();
        }else if (y1 == 0){
            stop(); leftturn(); stop(); forward();
            stop(); rightturn(); stop();
        }else if (y1 == 1){
            stop(); leftturn(); stop(); forward();
            stop(); rightturn(); stop(); forward();
            stop();
        }
    }else if (x1 == 0){
        if (y1 == -1){
            stop(); backward(); stop(); forward();
            stop(); backward(); stop();
        }else if (y1 == 0){
            stop();
        }else if (y1 == 1){
            stop(); forward(); stop();
        }
    }else if (x1 == 1){
        if (y1 == -1){
            stop(); rightturn(); stop(); forward();
            stop(); rightturn(); stop(); forward();
            stop(); backward(); stop();
        }else if (y1 == 0){
            stop(); rightturn(); stop(); forward();
            stop(); leftturn(); stop();
            //instDone = true;
        }else if (y1 == 1){
            stop(); rightturn(); stop(); forward();
            stop(); leftturn(); stop(); forward();
            stop();
        }
    }
}
```



Future Improvement

Add Component for Orientation Calibration

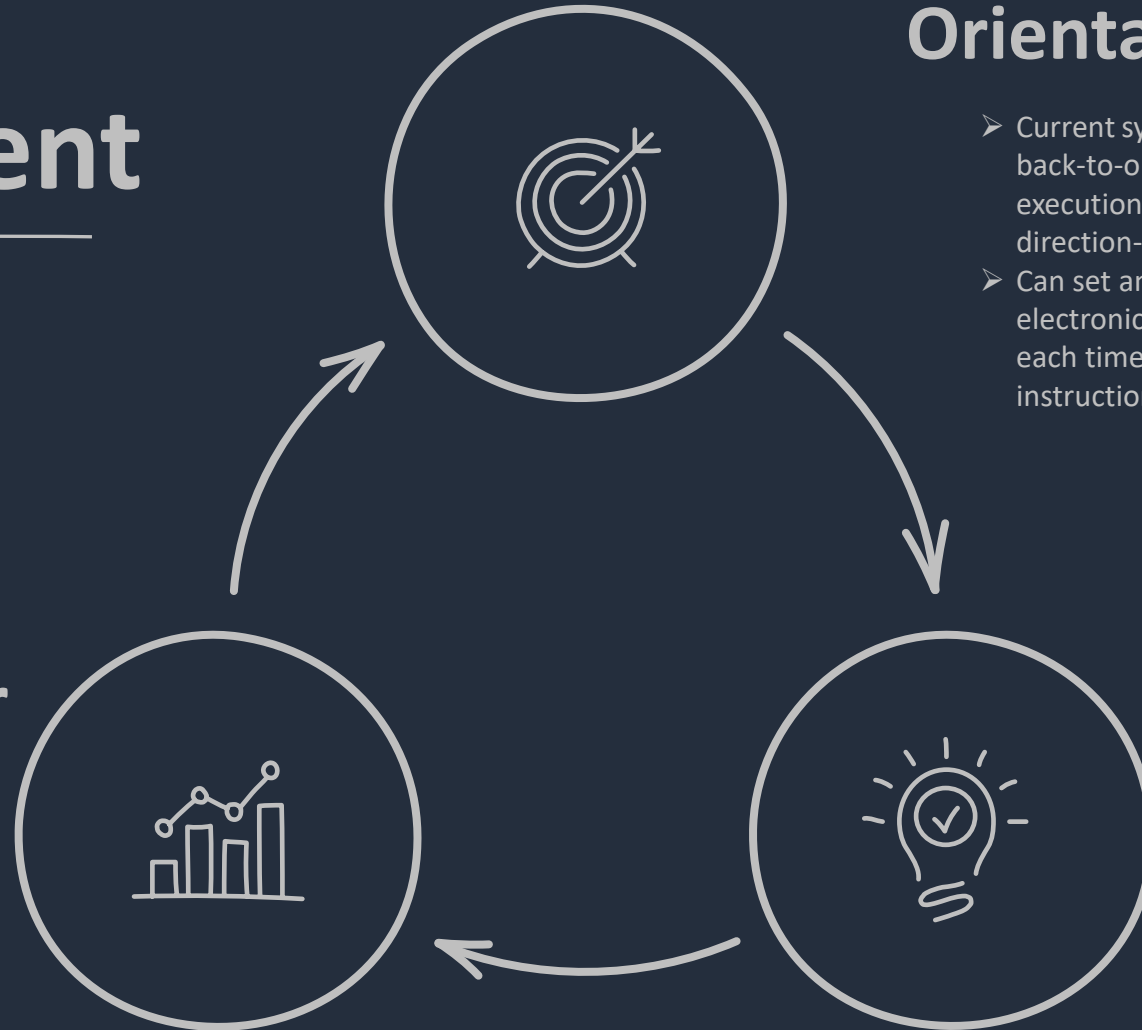
- Current system only hard-code the direction-back-to-original after each movement execution. Don't have solution for the case if direction-back-to-original fail to work.
- Can set an original direction and try using electronical compass to calibrate the direction each time after executing movement instruction.

More Precise Motor

- Current motor can only use PWM signal to adapt the speed which is not precise.
- Use more powerful and more precise motor like **stepper motor** which can decrease movement error between each individual motor (less deviation when moving and turning).

More Sturdy and Durable Frame

- Current Frame is soft and handmade, and the holes used to hang the motors are hand-drilled. Imprecision and errors accumulate fast along with movement.
- Use more sturdy materials, better design the position for each component, manufacture the hanging points for parts more precisely.



Collision-Avoidance System (Ultrasonic Sensor)

Specification

- Ranging Distance: 2cm – 400cm
- Resolution: 0.3cm
- Measuring Angle: 15 degree



Pin Definition

```
//pin definition for motor
int ML_Ctrl = 4;    //define the direction control pin of B motor
int ML_PWM = 5;    //define the PWM control pin of B motor
int MR_Ctrl = 2;    //define the direction control pin of A motor
int MR_PWM = 9;    //define the PWM control pin of A motor

//pin definition for ultrasonic sensor
int trigPin = 12;
int echoPin = 13;
```

Setup

```
//ultrasonic sensor pin setup
pinMode (trigPin, OUTPUT);
pinMode (echoPin, INPUT);
```

Pin Name	Port Connection
Vcc	soldered +5V Vcc
Trigger	digital PWM input
Echo	digital PWM output
Ground	soldered common ground

Collision-Avoidance System

Check Obstacle Function

```
// functions for ultrasonic sensor
double read_ultrasonic(){
    double time;
    // initialize: turn off the signal for 5 microseconds
    digitalWrite (trigPin, LOW);
    delayMicroseconds (5);

    // turn on the sensor and let it emit the signal for 10 microseconds (8 pulses)
    digitalWrite (trigPin, HIGH);
    delayMicroseconds (10);
    // turn off
    digitalWrite (trigPin, LOW);

    // set echopin active to receive the signal sent reflected back
    // pulseIn calculate the time used between sending out and receiving the ultrasonic signal
    duration = pulseIn (echoPin, HIGH);
    // calculate the distance in cm
    cm = (duration/2)/29.1;

    return cm;
}
```

Avoid Obstacle Function

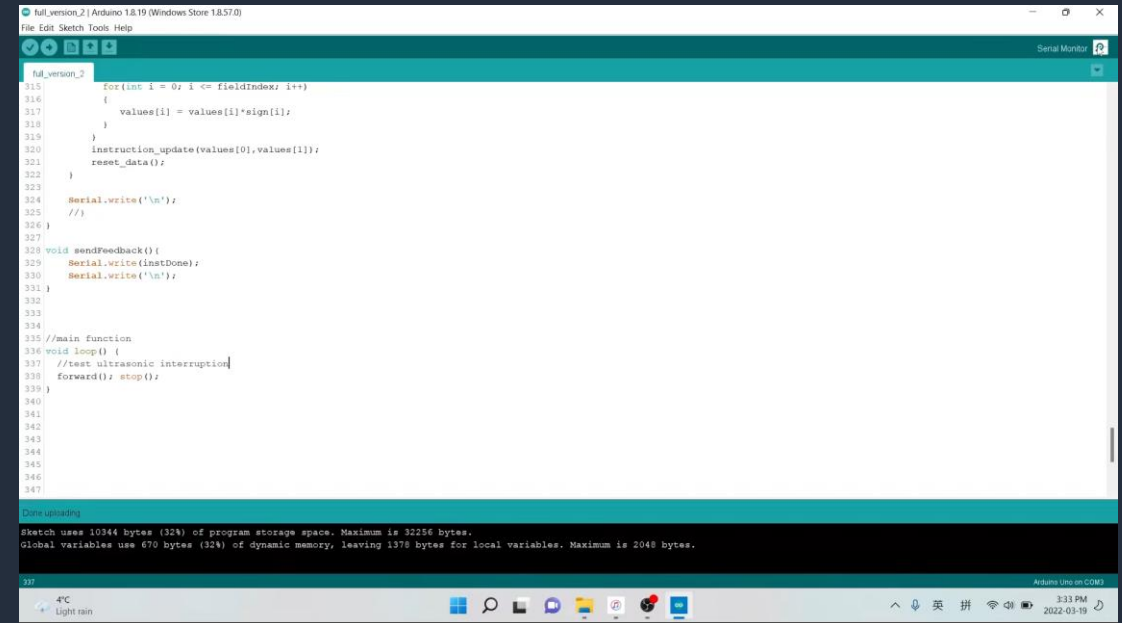
```
void ultrasonic_avoid(){
    Serial.println("Checking obstacles!");
    //Serial.println();
    delay(2000);
    //Serial.println();
    //forward();

    distance = read_ultrasonic();
    Serial.println(distance);
    //Serial.println();
    delay(2000);
    if (distance < safe_dist){
        Serial.println("Distance less than 50 cm, obstacle ahead detected!");
        //Serial.println();
        delay(2000);
        while (distance < safe_dist){
            distance = read_ultrasonic();
            Serial.println("Wait until obstacle disappear!");
            Serial.println(distance);
            delay(1000);
        }
        Serial.println("Interrupt finish! Go!");
    }
}
```

Forward Function

```
void forward(){ // move forward
    ultrasonic_avoid();
    digitalWrite(ML_Ctrl, HIGH);
    analogWrite(ML_PWM, 255);
    digitalWrite(MR_Ctrl, LOW);
    analogWrite(MR_PWM, 175);
    delay(move_time);
}
```

Demo



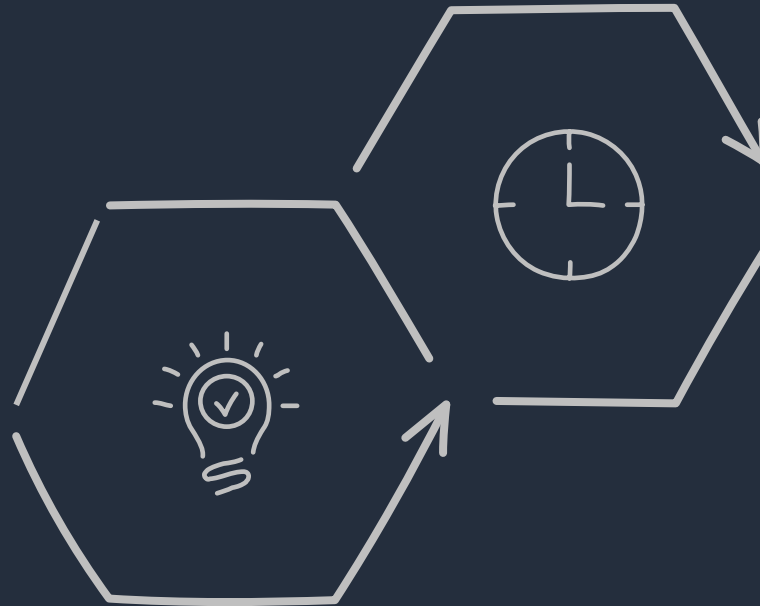
```
full_version_2
115   for(int i = 0; i <= fieldIndex; i++)
116   {
117       values[i] = values[i]*sign[i];
118   }
119   }
120   instruction_update(values[0], values[1]);
121   reset_data();
122   }
123
124   Serial.write('\n');
125   //}
126 }
127
128 void sendFeedback(){
129     Serial.write(instDone);
130     Serial.write('\n');
131 }
132
133
134
135 //main function
136 void loop() {
137     //test ultrasonic interruption
138     forward(); stop();
139 }
140
141
142
143
144
145
146
147

Data monitoring
Sketch uses 10344 bytes (32%) of program storage space. Maximum is 32256 bytes.
Global variables use 670 bytes (32%) of dynamic memory, leaving 1370 bytes for local variables. Maximum is 2048 bytes.
```

Future Improvement

Multi-directional Obstacle-Avoidance

- Only one ultrasonic sensor is used on our robot, and according to specification, the measuring angle is just 15 degree which means the detection range is limited.
- Can place more sensors on the robot with different facing angle to more effectively avoid obstacles in multiple directions.



Simultaneous Detection

- It's hard to do "multi-threading" on Arduino which means it's difficult to detect the obstacle while the robot is moving.
- We hard coded the collision-avoidance in the forward() function to make sure it is safe to move before start moving.
- Can try to add another Arduino board to handle the collision avoidance functionality.

Turning Precision (MPU6050, Gyro)



Yichen Xiao

Turning Precision (MPU6050, Gyro)

► Right Turn

▶ **Left Turn**

Full version 2 | Arduino 1.8.19 (Windows Store 1.8.57.0)

File Edit Sketch Tools Help

```

1  full_version_2
2
3  for(int i = 0; i <= fieldIndex; i++)
4  {
5      values[i] = values[i]*sign[i];
6  }
7  }
8  instruction_update(values[0],values[1]);
9  reset_data();
10 }
11
12 Serial.write('\n');
13 //}
14 }
15
16 void sendFeedback(){
17     Serial.write("instDone");
18     Serial.write('\n');
19 }
20
21 //main function
22 void loop() {
23     rightturn(); stop(); forward(); stop();
24 }
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

```

Done uploading

Sketch uses 11122 bytes (34%) of program storage space. Maximum is 32256 bytes.
Global variables use 554 bytes (27%) of dynamic memory, leaving 1494 bytes for local variables. Maximum is 2048 bytes.

Arduino Uno on COM3

4°C
Light rain

3:12 PM
2022-03-19

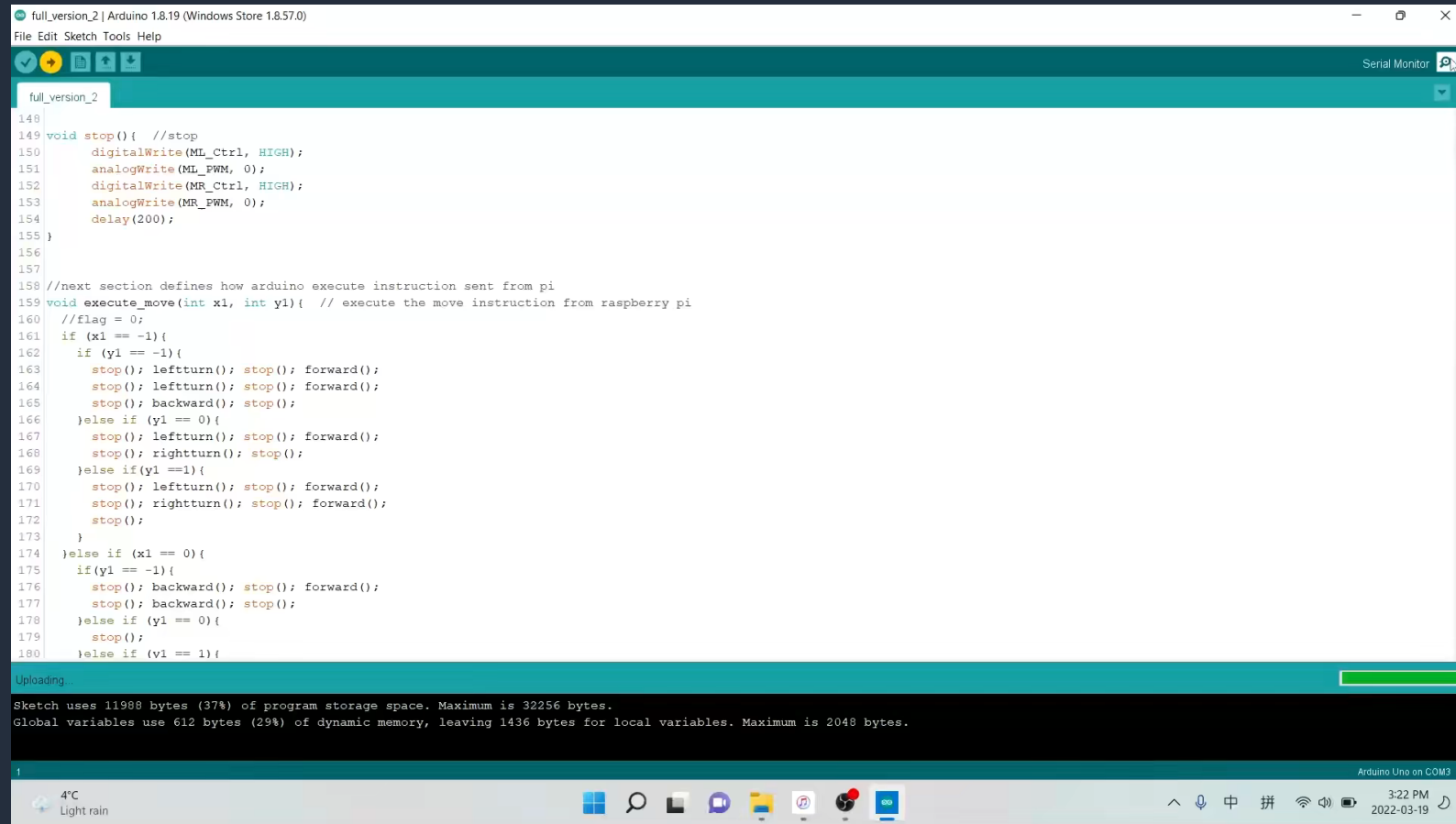
The image shows the Arduino IDE interface. At the top, the title bar reads "full_version_2 | Arduino 1.8.19 (Windows Store 1.8.57.0)". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for opening files, saving, undo, redo, and other IDE functions. The main workspace displays a C++ sketch for a servo motor. The code is as follows:

```
326  
327     Serial.write('\n');  
328     //}  
329 }  
330  
331 void sendFeedback() {  
332     Serial.write("instDone");  
333     Serial.write('\n');  
334 }  
335  
336  
337  
338 //main function  
339 void loop() {  
340     leftTurn(); stop(); forward(); stop();  
341 }  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353
```

The status bar at the bottom of the IDE shows "Done uploading". Below this, a message states: "Sketch uses 11130 bytes (34%) of program storage space. Maximum is 32256 bytes. Global variables use 554 bytes (27%) of dynamic memory, leaving 1494 bytes for local variables. Maximum is 2048 bytes." The bottom of the screen shows the Windows taskbar with the Start button, task view, and several pinned applications. The system tray on the right shows the weather as "4°C Light rain", the time as "3:06 PM", and the date as "2022-03-19".

Turning Precision (MPU6050, Gyro)

Multiple Turn



```
full_version_2 | Arduino 1.8.19 (Windows Store 1.8.57.0)
File Edit Sketch Tools Help

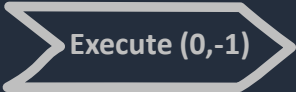
full_version_2
148
149 void stop() { //stop
150     digitalWrite(ML_Ctrl, HIGH);
151     analogWrite(ML_PWM, 0);
152     digitalWrite(MR_Ctrl, HIGH);
153     analogWrite(MR_PWM, 0);
154     delay(200);
155 }
156
157
158 //next section defines how arduino execute instruction sent from pi
159 void execute_move(int x1, int y1) { // execute the move instruction from raspberry pi
160     //flag = 0;
161     if (x1 == -1) {
162         if (y1 == -1) {
163             stop(); leftturn(); stop(); forward();
164             stop(); leftturn(); stop(); forward();
165             stop(); backward(); stop();
166         } else if (y1 == 0) {
167             stop(); leftturn(); stop(); forward();
168             stop(); rightturn(); stop();
169         } else if (y1 == 1) {
170             stop(); leftturn(); stop(); forward();
171             stop(); rightturn(); stop(); forward();
172             stop();
173         }
174     } else if (x1 == 0) {
175         if (y1 == -1) {
176             stop(); backward(); stop(); forward();
177             stop(); backward(); stop();
178         } else if (y1 == 0) {
179             stop();
180         } else if (y1 == 1) {
181             stop();
182         }
183     }
184 }

Uploading...

Sketch uses 11988 bytes (37%) of program storage space. Maximum is 32256 bytes.
Global variables use 612 bytes (29%) of dynamic memory, leaving 1436 bytes for local variables. Maximum is 2048 bytes.

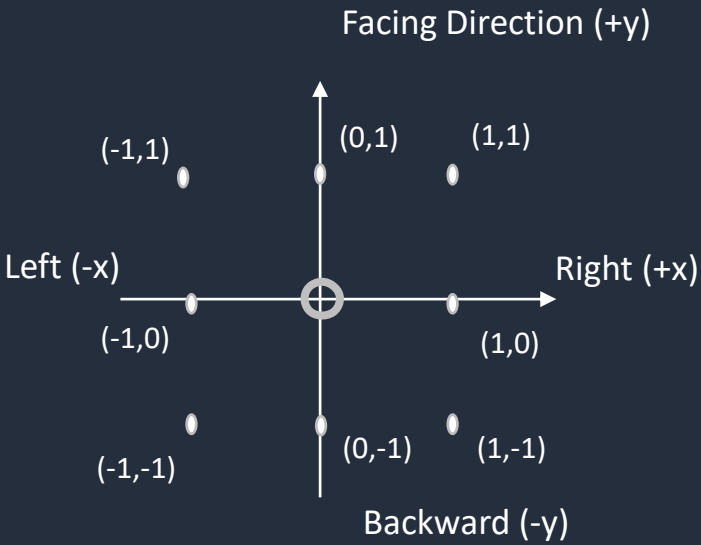
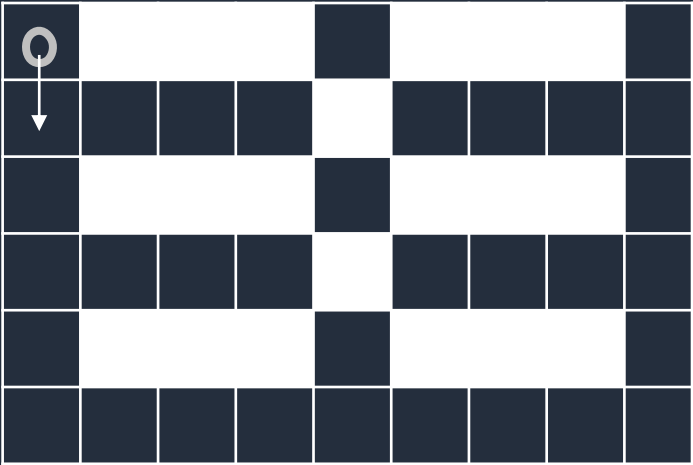
1
Arduino Uno on COM3
4°C
Light rain
3:22 PM
2022-03-19
```

Execution_Move Demo



```
full_version_2 | Arduino 1.8.19 (Windows Store 1.8.57.0)
File Edit Sketch Tools Help

full_version_2
313 else
314 {
315   for(int i = 0; i <= fieldIndex; i++)
316   {
317     values[i] = values[i]*sign[i];
318   }
319 }
320 instruction_update(values[0],values[1]);
321 reset_data();
322 }
323
324 Serial.write('\n');
325 //)
326 )
327
328 void sendFeedback(){
329   Serial.write(instDone);
330   Serial.write('\n');
331 }
332
333
334
335 //main function
336 void loop() {
337   //test execute move with gyro and ultrasonic (single instruction)
338   execute_move(0,-1);
339   stop();
340   delay(1000);
341 }
342
343
344
345
Done uploading
Sketch uses 11524 bytes (35%) of program storage space. Maximum is 32256 bytes.
Global variables use 678 bytes (33%) of dynamic memory, leaving 1370 bytes for local variables. Maximum is 2048 bytes.
349 Arduino
```



Execution_Move Demo

Execute (-1,1)

```
full_version_2 | Arduino 1.8.19 (Windows Store 1.8.57.0)
File Edit Sketch Tools Help

full_version_2
310     }
311     Serial.write(ch);
312 }
313 else
314 {
315     for(int i = 0; i <= fieldIndex; i++)
316     {
317         values[i] = values[i]*sign[i];
318     }
319 }
320 instruction_update(values[0],values[1]);
321 reset_data();
322 }
323
324 Serial.write('\n');
325 //)
326 }
327
328 void sendFeedback(){
329     Serial.write(instDone);
330     Serial.write('\n');
331 }
332
333
334
335 //main function
336 void loop() {
337     //test execute move with gyro and ultrasonic (single instruction)
338     execute_move(-1,1);
339     stop();
340 }
341
342
```

Done uploading

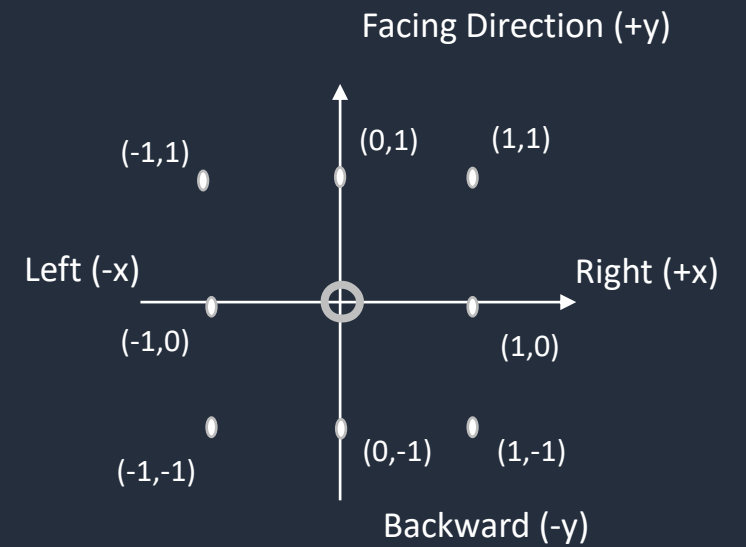
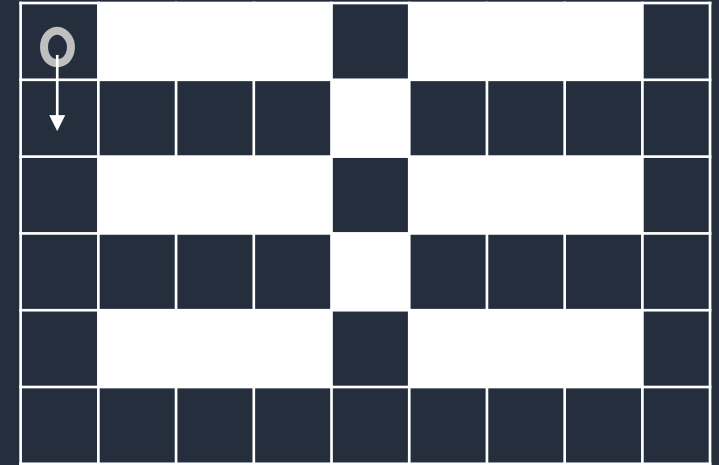
Sketch uses 11970 bytes (37%) of program storage space. Maximum is 32256 bytes.
Global variables use 678 bytes (33%) of dynamic memory, leaving 1370 bytes for local variables. Maximum is 2048 bytes.

25

4°C
Light rain

Arduino Uno on COM3

3:52 PM
2022-03-19



Check Location (GPS)

— Yuchen Miao



User Interaction

— Zhuoxin Ma



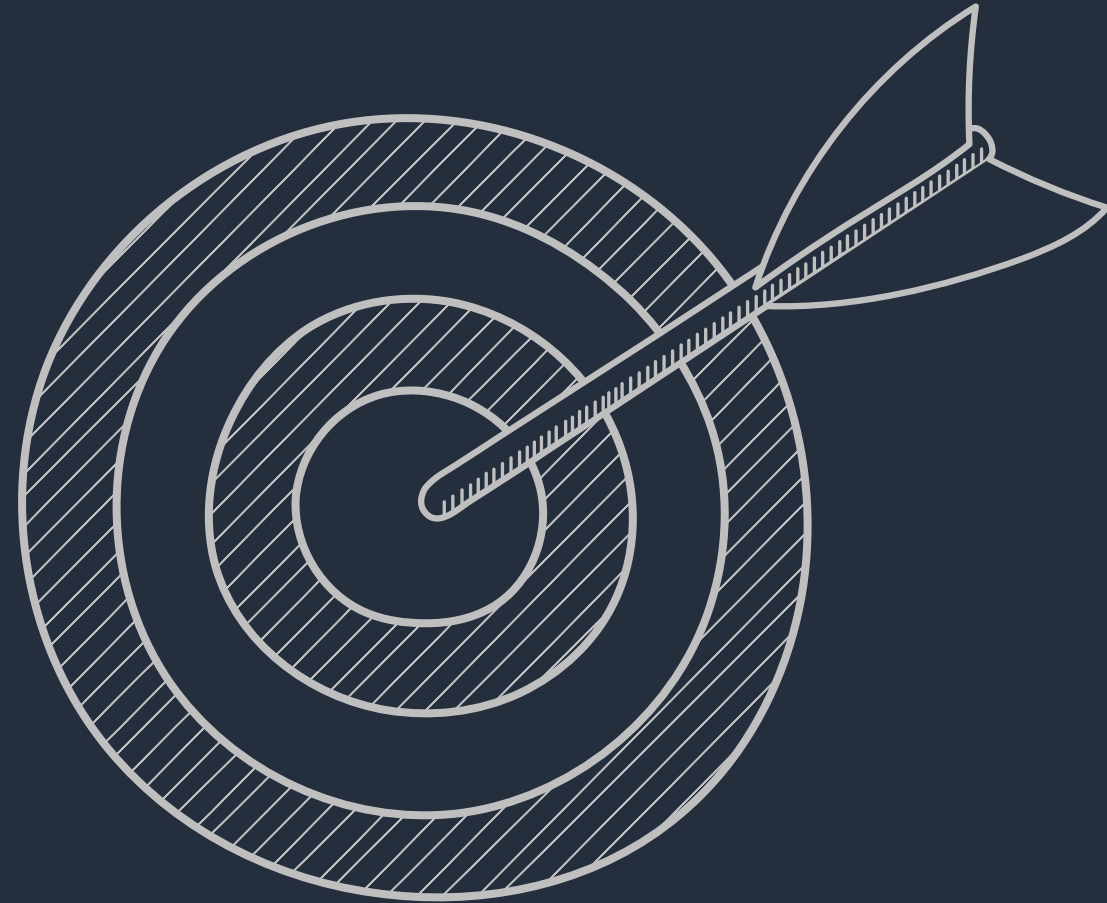
Process Pickup

- Use two-step verification, require QR code and customer first name
- Reduce the cases of misdrawing goods



Update Pickup Info

- Automatically record pickup time in the system
- Improve store database and e-commerce system



Process Pickup

Order Number	Order Time	Product Name	Product Code	Buyer	Scheduled Pick Up Time	Cabin #	Actual Pick Up Time
0	1 13:20, 14-2-2021	iphone 12	01023	Matt Iqra	slot 3	1	NaN
1	2 15:30, 14-2-2021	iphone 12 pro	39480	Ansari Mishaal	slot 3	2	NaN
2	3 15:43, 14-2-2021	iphone 12 pro	85743	Appadoo Disha	slot 3	3	NaN
3	4 16:40, 14-2-2021	logi keyboardxxx	58934	Bishop Zac	slot 3	4	NaN
4	5 17:02, 14-2-2021	ipad pro	84839	Phil Fred	slot 3	5	NaN
5	6 19:06, 14-2-2021	macbook pro	39407	Bell Johnathan	slot 3	6	NaN
6	7 19:47, 14-2-2021	logi mousexxx	93504	Bruca Tudor	slot 7	NaN	NaN
7	8 20:09, 14-2-2021	iphone 12	59348	Mirza Nabeel	slot 4	NaN	NaN
8	9 20:18, 14-2-2021	iphone 12 pro max	98954	Megan Wailly	slot 6	NaN	NaN
9	10 21:09, 14-2-2021	iphone 12	30299	Muir Ben	slot 8	NaN	NaN

Step 1

- Ask customer to scan QR code sent by the store
- Extract product number
- Find order in the data frame
- If no matched order, let customer enter store for help
- If there is matched order, process step 2

Step 2

- Require customer to enter first name which should be the same name they entered when placing the order
- Customer can try three times, if fail exceed three times, let customer try again later or tell them enter store for help

Other Situations

- If customer come during the time which is not the schedule pick up time, let customer enter store for help

Process Pickup

Get_order_info()

Extract the product number from result got by scanning the QR code

```
1 def get_order_info():
2     print("Please scan your product QR code!")
3     decode = scan_product_qrcode()
4     product_code = decode.split(": ")[-1]
5     return product_code
```

match()

Check if customer name matches with the product number

```
1 def match():
2     customer_name = input ("Please type in your first name:")
3     get_name = operate_info.loc[idx[0]].at["Buyer"]
4     if customer_name.lower() == get_name.split(' ')[0].lower():
5         return True
6     else:
7         return False
```

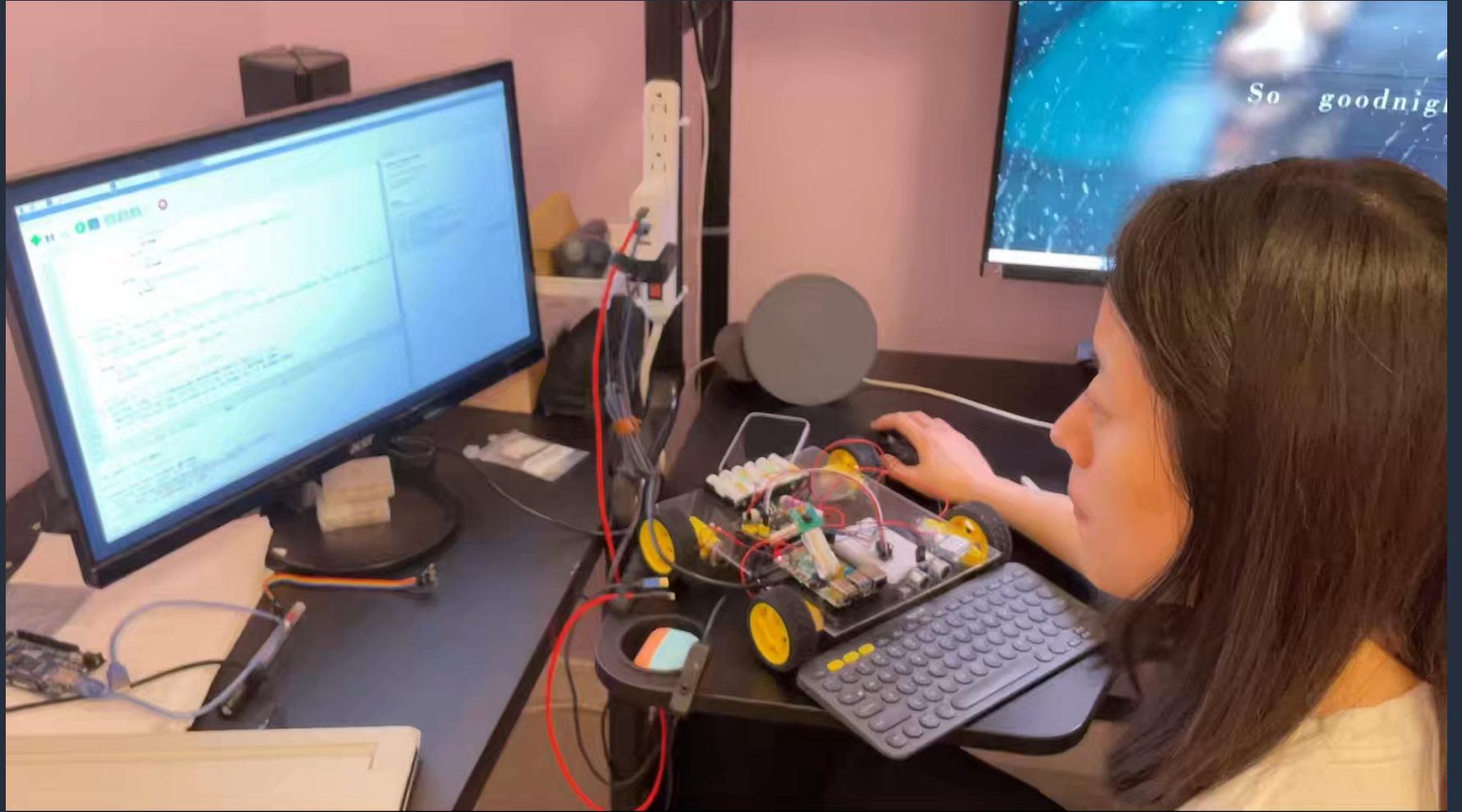
Process Pickup + Update Info

Main function

```
1 #initialize
2 process = 0
3 picked = 0
4
5 # delivery robot arrived!
6 if isarrived == 1:
7     # ask customer to scan their QR code
8     product_code = get_order_info()
9     # find the order from system database
10    idx = operate_info.index[operate_info['Product Code'] == product_code].tolist()
11
12    # check if product info matches with customer name in the system
13    get_name = operate_info.loc[idx[0]].at["Buyer"]
14    if not idx:
15        print("Your order can not be found in the system! Please enter store and ask for help!")
16    else:
17        if match():
18            print("Processing...")
19            process = 1
20        else:
21            t=0
22            for i in range (0,3):
23                t+=1
24                print("Customer name and order don't match !")
25                request = input ("Try typing first name again ? (Y/N)")
26
27                if request in ["N","n","no","NO","No"]:
28                    print("Thanks. See you.")
29                    break
30                elif request in ["Y","y","yes","YES","Yes"]:
31                    get_name = operate_info.loc[idx[0]].at["Buyer"]
32                    if match ():
33                        print("Processing...")
34                        process = 1
35                        break
36                    elif t == 3 :
37                        print("Exceed maximum trial times. Try again later.")
38                        break
39                else:
40                    print("Invalid input!")
41                    break
42
43    # if match: process "open cabin"
44    if process == 1:
45        cabin_num = operate_info.loc[idx[0]].at["Cabin #"]
46        # if customer does not come during the scheduled pick up time
47        if cabin_num == "NaN":
48            print("Your order is not ready for pick up! Please come during scheduled time slot or enter store for help!")
49        else:
50            print("Opening cabin #" + cabin_num)
51            picked = 1
52
53    # update pick up time into system database
54    if picked == 1:
55        pickup_time = str(datetime.datetime.now()).split(".")[0]
56        operate_info.at[idx,'Actual Pick Up Time'] = pickup_time
57        print('Item has been picked up by customer at ' + pickup_time)
58
```

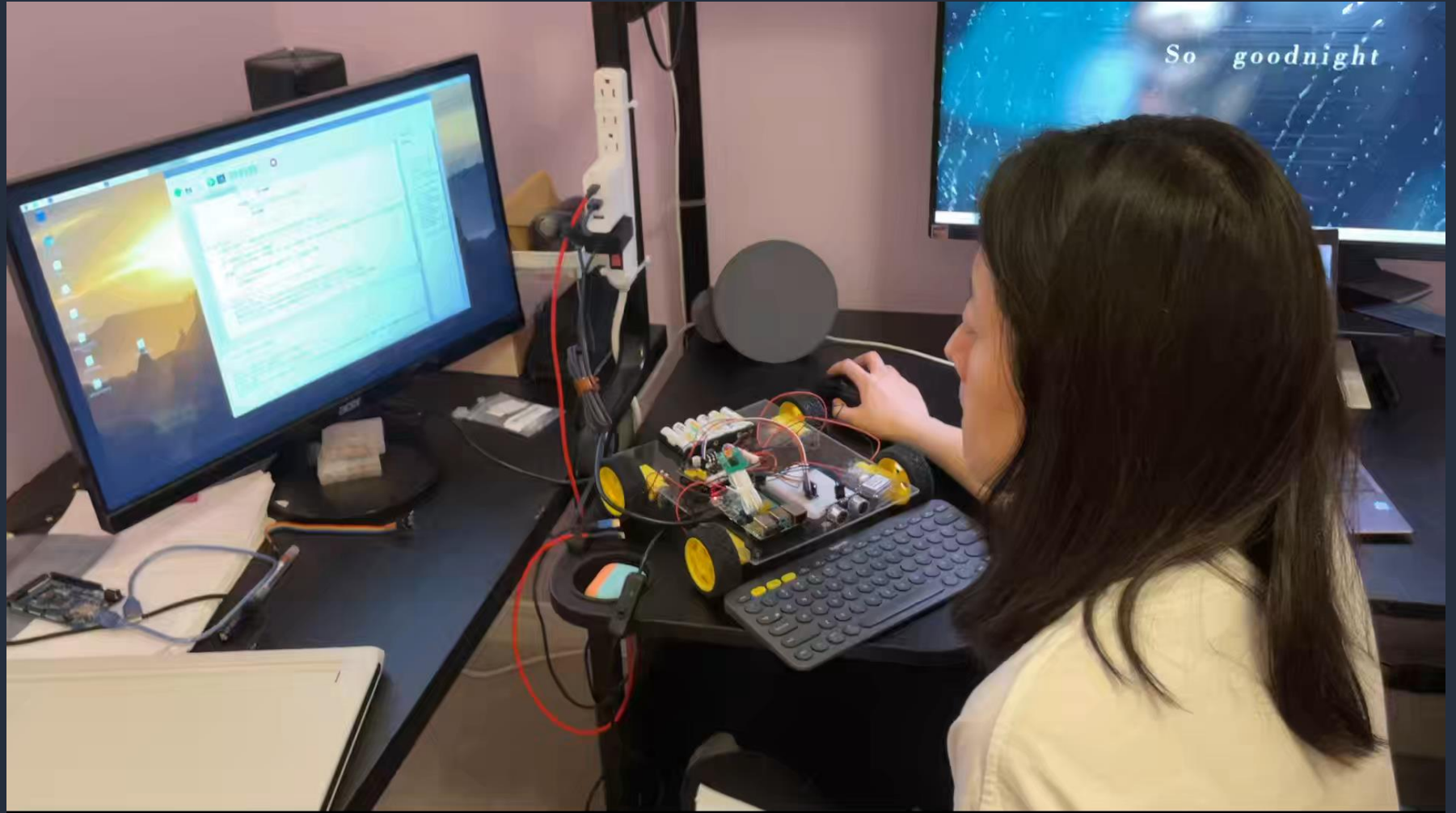
User Interaction Demo

Customer come during
scheduled time



User Interaction Demo

Customer doesn't come
during scheduled time



Conclusion



**Zhuoxin Ma
Yuchen Miao
Lang Sun
Yichen Xiao**

Potential Improvement

INFODATA 03

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim



Find a better operating logic, allowing customer to come anytime after the order is processed



INFODATA 04

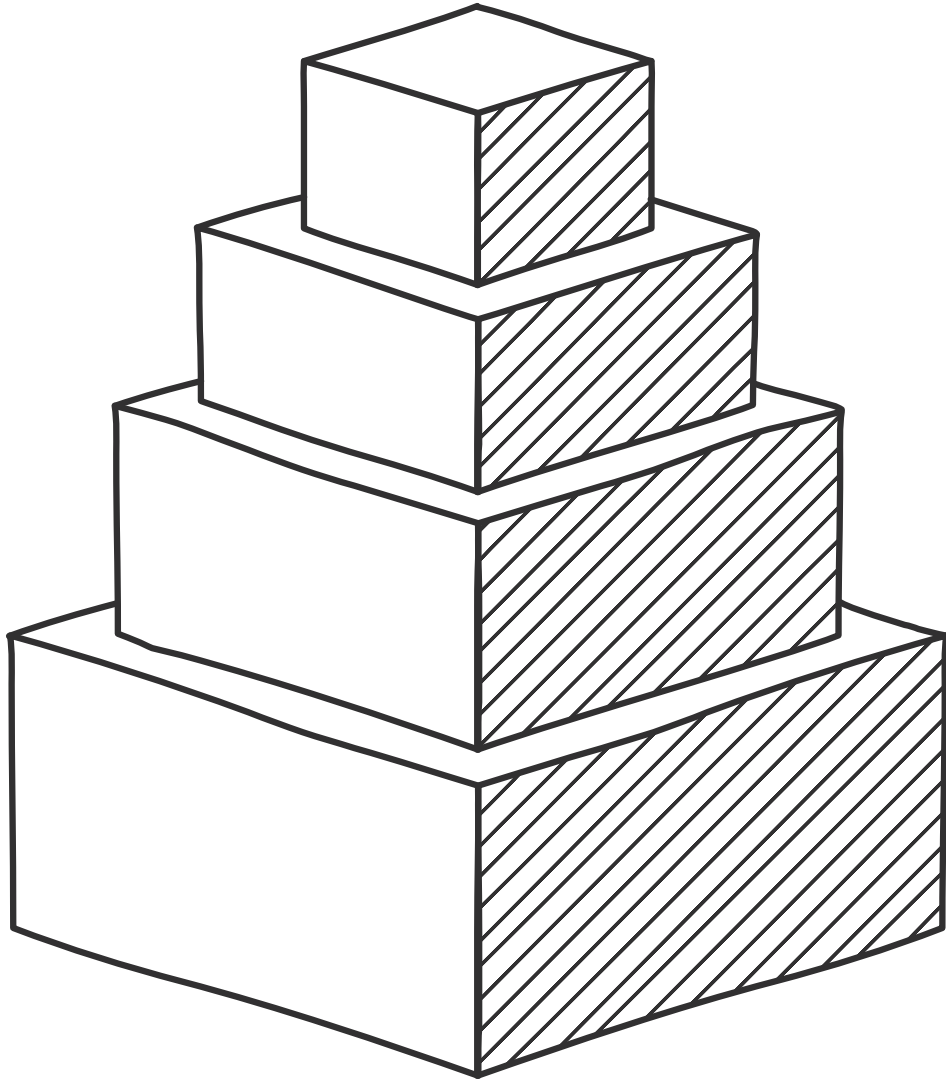
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim



INFODATA 06

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim





Thanks for watching

Thanks everyone for the hard work!

Thanks Professor Liu for all the help!