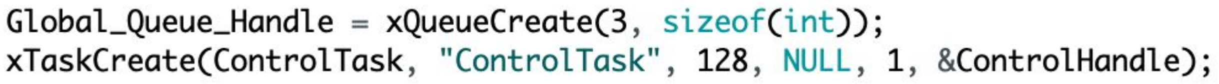
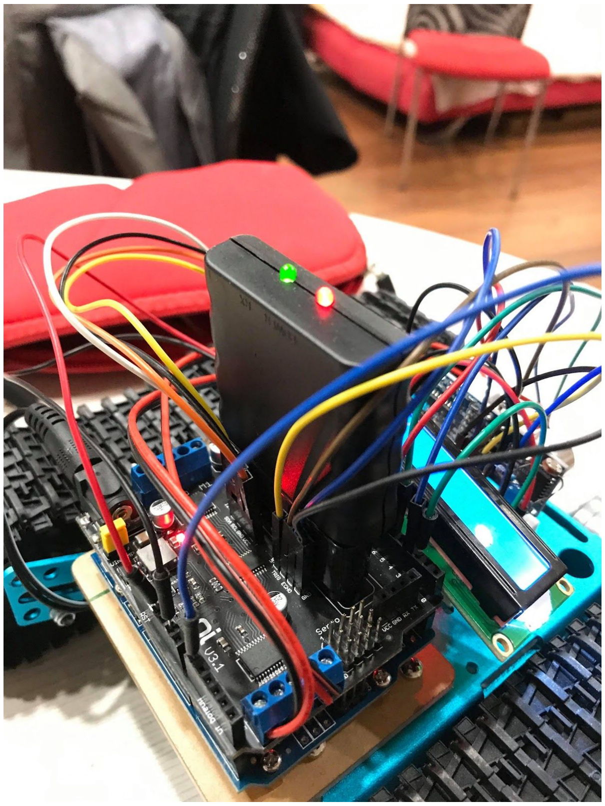
Final Project
106034061
曾靖渝

1. Project Title : Mini Tank
2. Goal & Implementations:
    a. RTOS

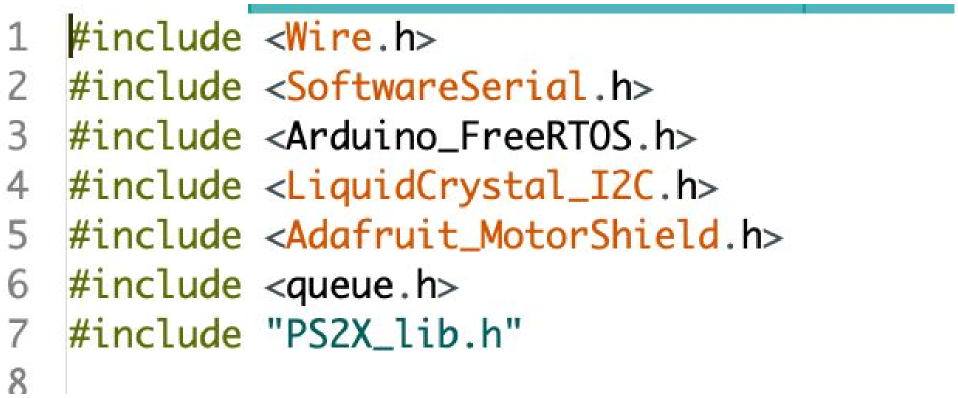


```
Global_Queue_Handle = xQueueCreate(3, sizeof(int));
xTaskCreate(ControlTask, "ControlTask", 128, NULL, 1, &ControlHandle);
```

    b. Moving (10%):
        i. Use PS2 Controller & Arduino Sensor shield to control the tank



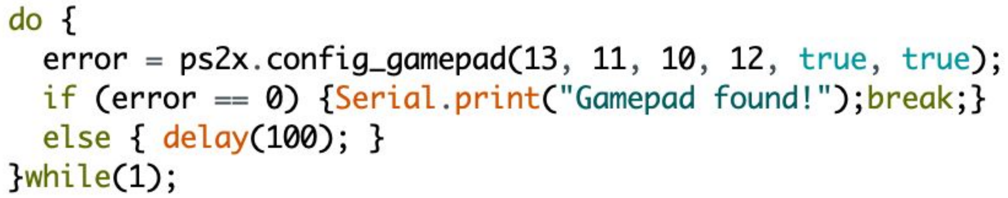
        ii. Include “motor shield library”



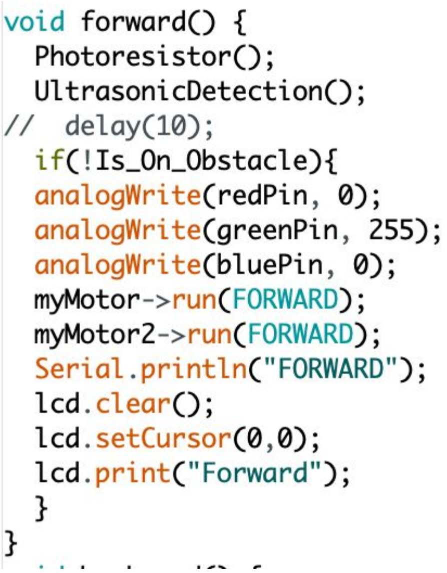
```
#include <Wire.h>
#include <SoftwareSerial.h>
#include <Arduino_FreeRTOS.h>
#include <LiquidCrystal_I2C.h>
#include <Adafruit_MotorShield.h>
#include <queue.h>
#include "PS2X_lib.h"

```

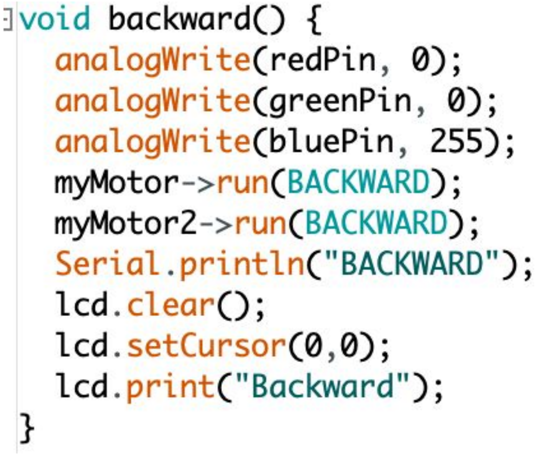
iii. Send the instructions to the motherboard.



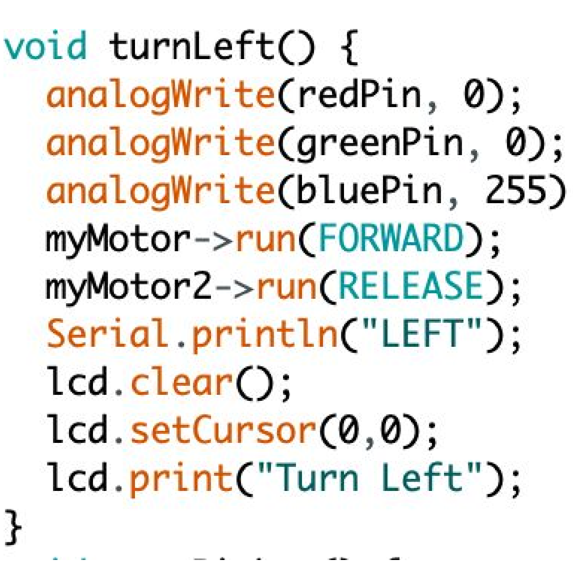
```
  do {
    error = ps2x.config_gamepad(13, 11, 10, 12, true, true);
    if (error == 0) {Serial.print("Gamepad found!");break;}
    else { delay(100); }
  }while(1);
```



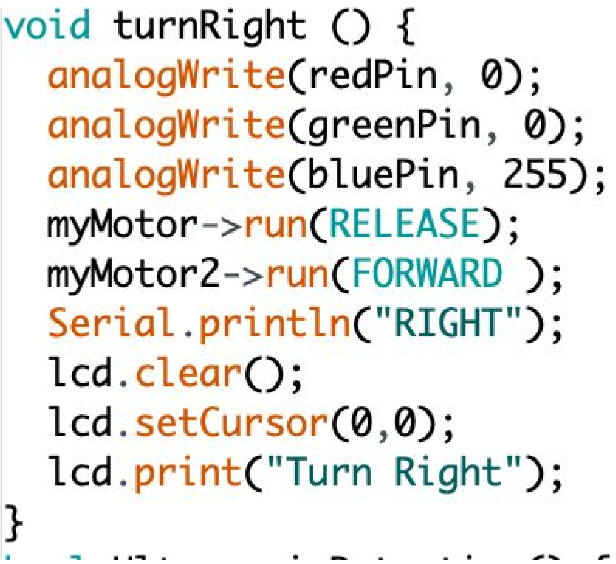
```
void forward() {
  Photoresistor();
  UltrasonicDetection();
//  delay(10);
  if(!Is_On_Obstacle){
  analogWrite(redPin, 0);
  analogWrite(greenPin, 255);
  analogWrite(bluePin, 0);
  myMotor->run(FORWARD);
  myMotor2->run(FORWARD);
  Serial.println("FORWARD");
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Forward");
  }
}
```



```
void backward() {
  analogWrite(redPin, 0);
  analogWrite(greenPin, 0);
  analogWrite(bluePin, 255);
  myMotor->run(BACKWARD);
  myMotor2->run(BACKWARD);
  Serial.println("BACKWARD");
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Backward");
}
```

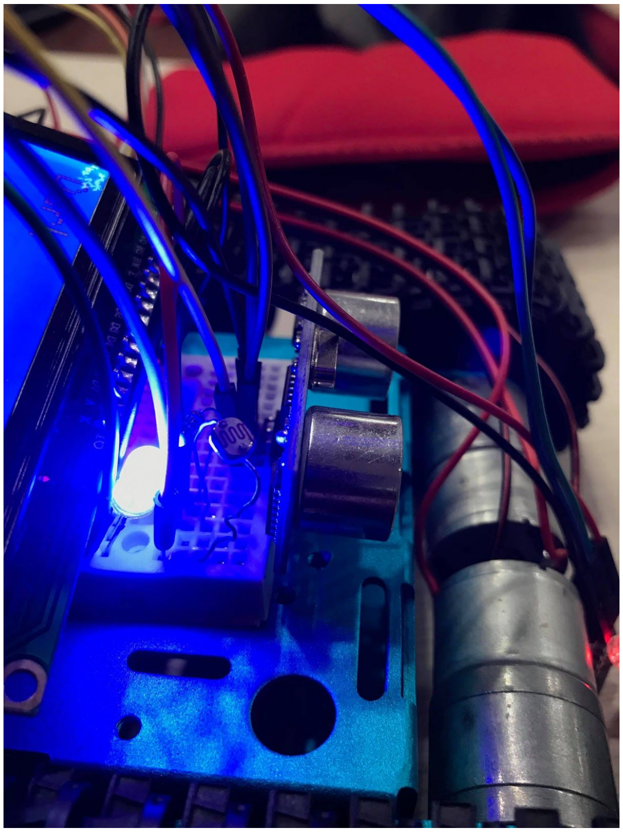


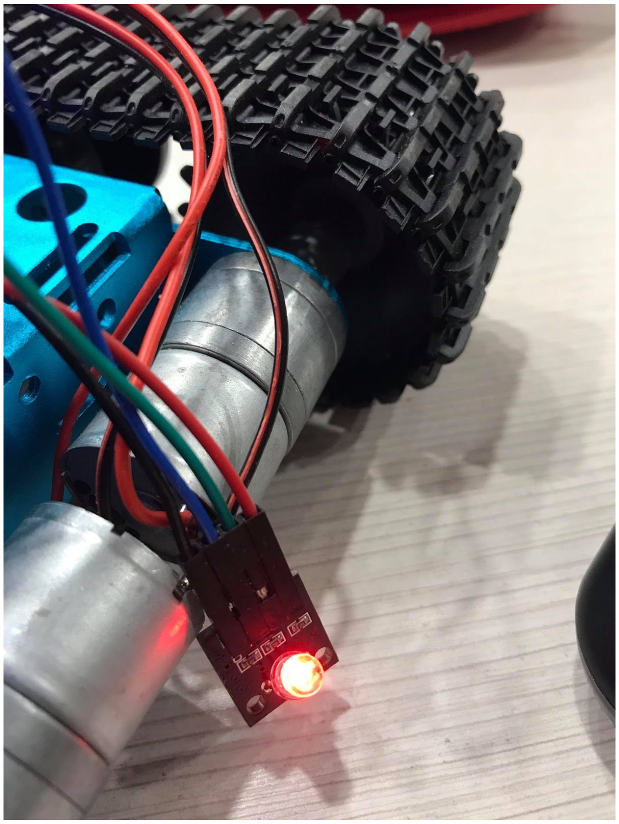
```
void turnLeft() {
  analogWrite(redPin, 0);
  analogWrite(greenPin, 0);
  analogWrite(bluePin, 255)
  myMotor->run(FORWARD);
  myMotor2->run(RELEASE);
  Serial.println("LEFT");
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Turn Left");
}
```



```
void turnRight () {
  analogWrite(redPin, 0);
  analogWrite(greenPin, 0);
  analogWrite(bluePin, 255);
  myMotor->run(RELEASE);
  myMotor2->run(FORWARD );
  Serial.println("RIGHT");
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Turn Right");
}
```

c. HeadLight:
    i. Use LED to provide the light
    ii. Control LED by the photoresistance

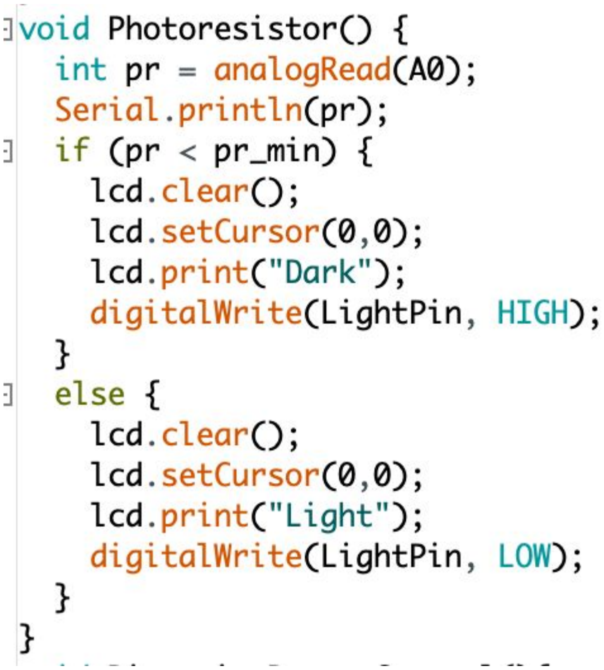




d. Instruction RGB:
    i. Use RGB to show the moving situation

e. Photoresistance
    i. Detect the variance of light

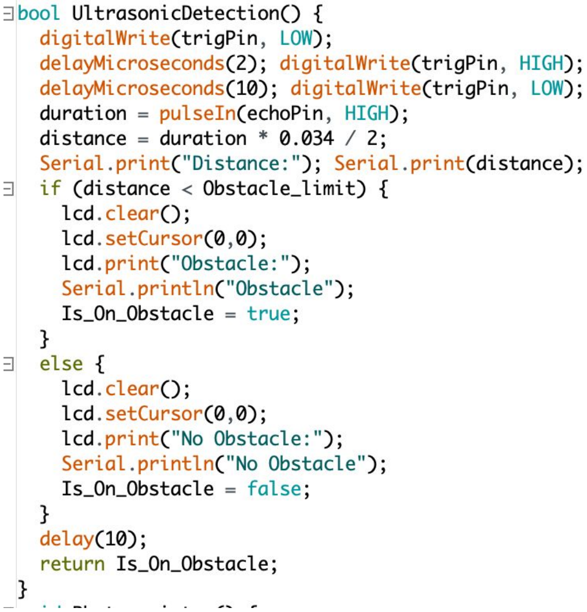
ii. Control the headlight



```
void Photoresistor() {
  int pr = analogRead(A0);
  Serial.println(pr);
  if (pr < pr_min) {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Dark");
    digitalWrite(LightPin, HIGH);
  }
  else {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Light");
    digitalWrite(LightPin, LOW);
  }
}
```

f. Timer:
   i. Control the signal sended by PS2 Controller.

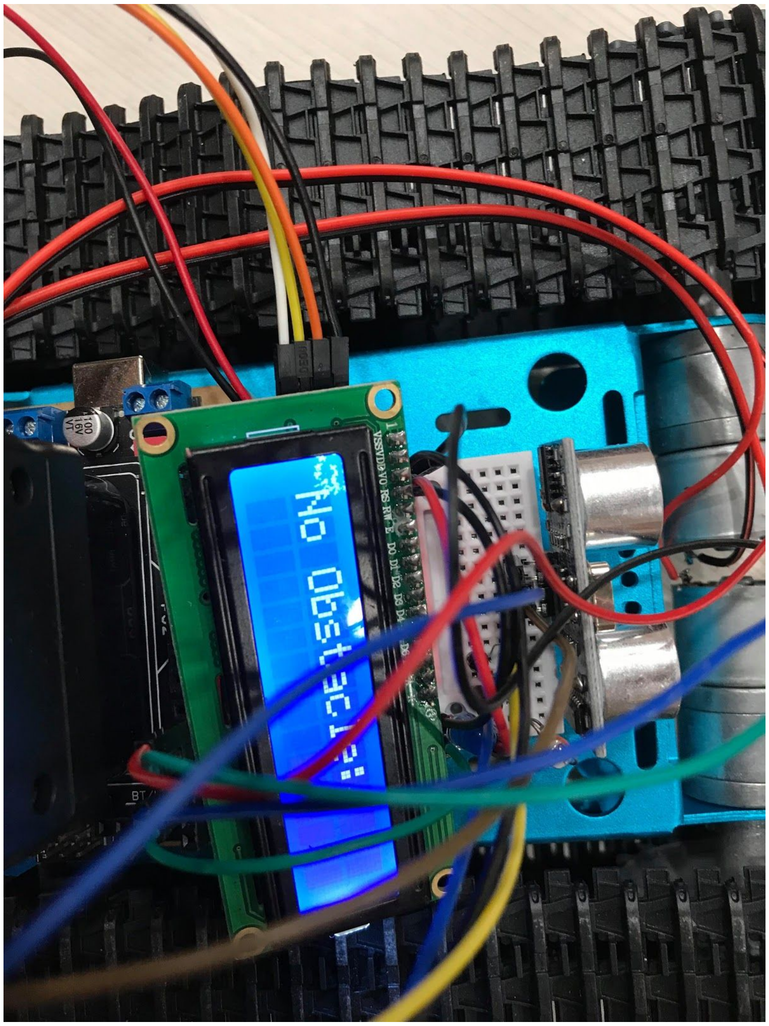
g. Ultrasensor
   i. Use to detect the obstacles
   ii. Control the moving
   iii. Control LCD



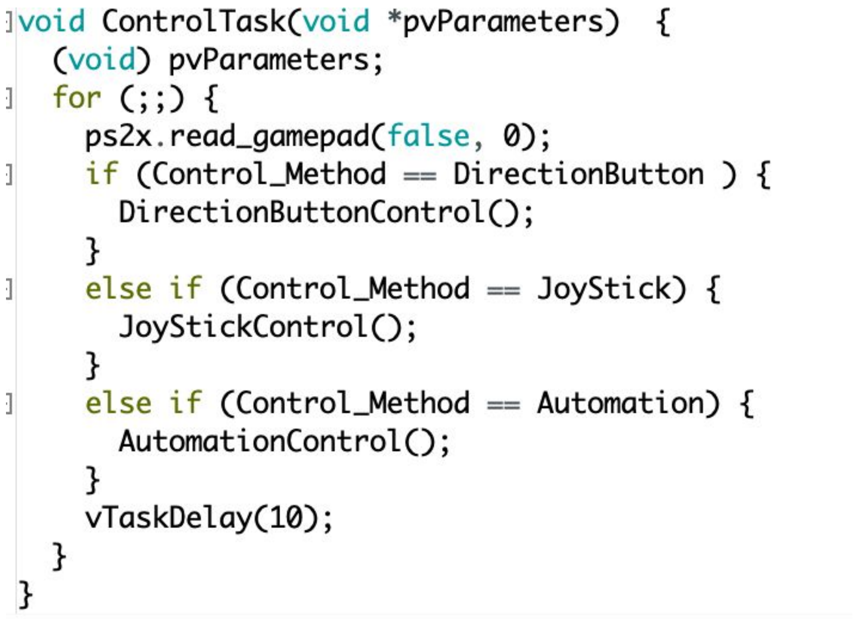
```
bool UltrasonicDetection() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2); digitalWrite(trigPin, HIGH);
  delayMicroseconds(10); digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;
  Serial.print("Distance:"); Serial.print(distance);
  if (distance < Obstacle_limit) {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Obstacle:");
    Serial.println("Obstacle");
    Is_On_Obstacle = true;
  }
  else {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("No Obstacle:");
    Serial.println("No Obstacle");
    Is_On_Obstacle = false;
  }
  delay(10);
  return Is_On_Obstacle;
}
```

h. LCD:
   i. Show the light/Dark
   ii. Obstacle /No Obstacle

iii. Forward/Backward/Turn Left/ Turn Right



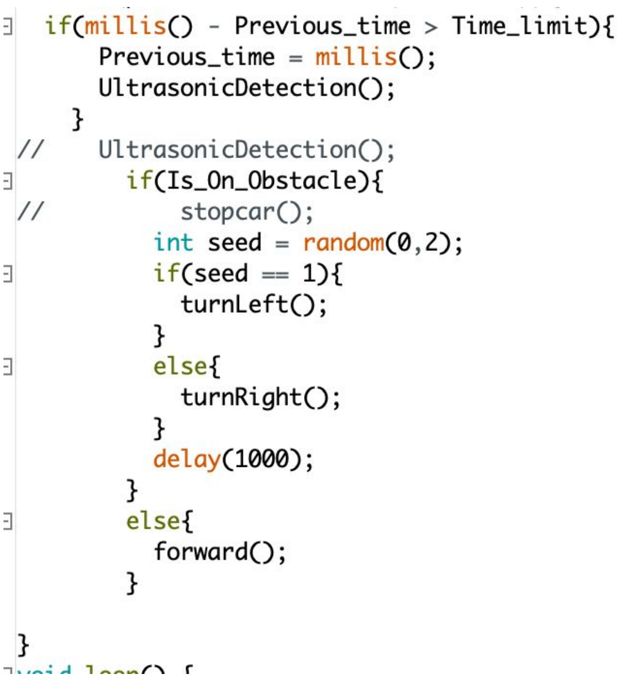
3. 3 Modes



```
void ControlTask(void *pvParameters)  {
  (void) pvParameters;
  for (;;) {
    ps2x.read_gamepad(false, 0);
    if (Control_Method == DirectionButton ) {
      DirectionButtonControl();
    }
    else if (Control_Method == JoyStick) {
      JoyStickControl();
    }
    else if (Control_Method == Automation) {
      AutomationControl();
    }
    vTaskDelay(10);
  }
}
```

a. Direction Button Mode:
   i. Control Tank By Button
   ii. Fixed Speed
   iii. Easy Controllable
b. JoyStick Mode
   i. Control Tank By joystick
   ii. Difference Speed Variance
   iii. Linear Accelerator the Tank by JoyStick
   iv. Turn Left/Right by Joystick (like steering wheel)
   v. Like Racing Car

c. Automation:
   i. Automatic Moving
   ii. If detect obstacles, turn Left/Right in Random.

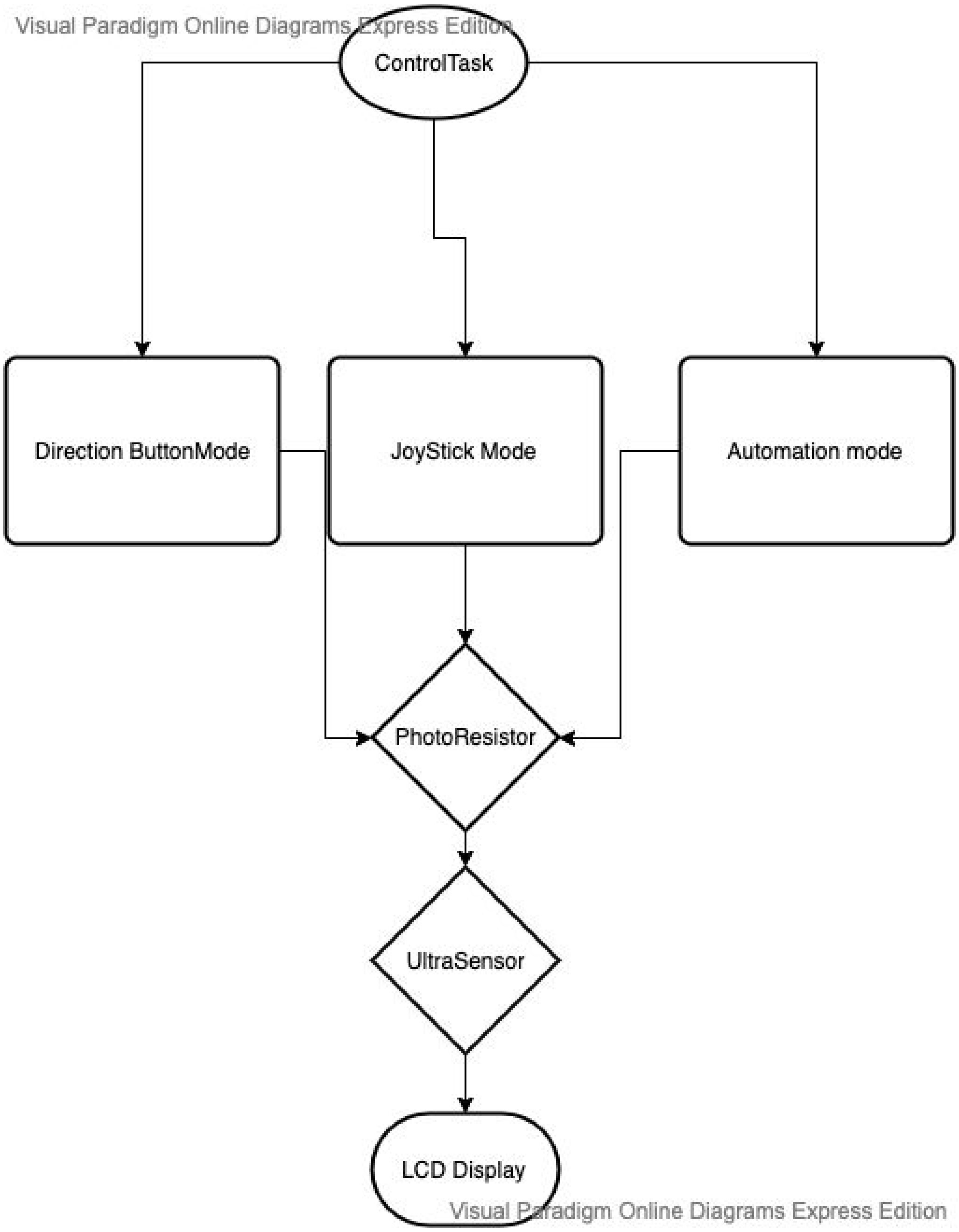


```
  if(millis() - Previous_time > Time_limit){
      Previous_time = millis();
      UltrasonicDetection();
    }
//    UltrasonicDetection();
        if(Is_On_Obstacle){
//          stopcar();
          int seed = random(0,2);
          if(seed == 1){
            turnLeft();
          }
          else{
            turnRight();
          }
          delay(1000);
        }
        else{
          forward();
        }

}
void loop() {
```

4. Program Flow Chart

a.


ControlTask
Direction ButtonMode
JoyStick Mode
Automation mode
PhotoResistor
UltraSensor
LCD Display