

Ngày 29 tháng 5 năm 2023

Buổi sáng

Xin nghỉ

Buổi chiều

Xin nghỉ

Ngày 30 tháng 5 năm 2023

Buổi sáng

Buổi chiều (Làm báo cáo thực tập VĐK)

- Ngắt, bộ định thời và bộ đếm: định nghĩa ngắt, timer/counter, cách sử dụng ngắt/ timer/ counter trong Arduino: ngắt ngoài, ngắt timer. Mỗi trường hợp có ít nhất một ví dụ

Ngày 31 tháng 5 năm 2023

Buổi sáng

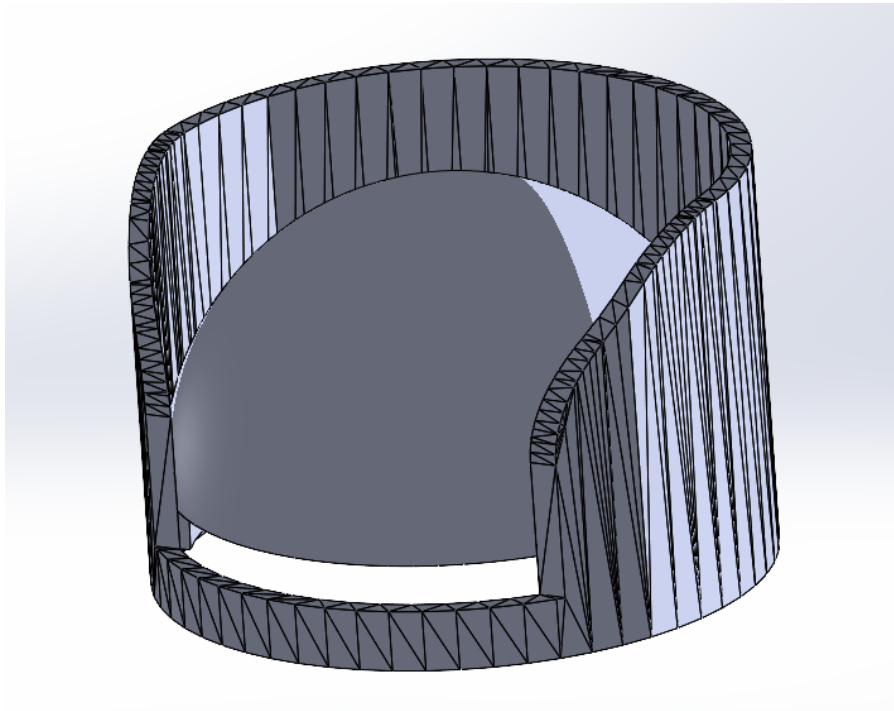
- Làm báo cáo thực tập

Buổi chiều

- Nghỉ

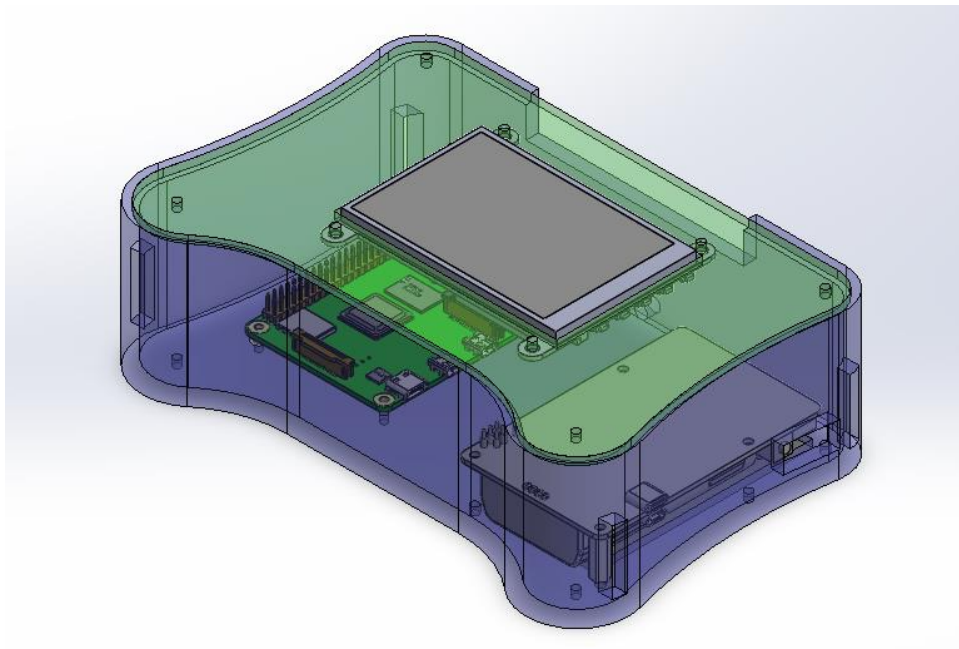
Ngày 1 tháng 6 năm 2023

Buổi sáng



Hình 1 mũ robot nhện

Buổi chiều



Hình 2 tay cầm robot nhện

Ngày 2 tháng 6 năm 2023

Buổi sáng

Tối ưu code

```
float pTerm, iTerm, dTerm;
int error, previousError;
float kp = 15, ki = 0, kd = 15; // 15
float output;
int integral, derivative;
int motor1newSpeed;
int motor2newSpeed;

int sensorPins[5] = {A1, A2, A3, A4, A5};
int motorPins[6] = {12, 6, 13, 11, 3, 5};
int s = 120; // max 120
int motor1Speed = s; //Default r
int motor2Speed = s; //Default l

void setup() {
    for (int i = 0; i < 5; i++) {
        pinMode(sensorPins[i], INPUT_PULLUP);
    }

    for (int i = 0; i < 6; i++) {
        pinMode(motorPins[i], OUTPUT);
    }

    Serial.begin(9600);
}

void calculateError() {
    int sensorValues[5];
    sensorValues[0] = digitalRead(A1);
    sensorValues[1] = digitalRead(A2);
    sensorValues[2] = digitalRead(A3);
    sensorValues[3] = digitalRead(A4);
    sensorValues[4] = digitalRead(A5);

    int errorValues[13][5] = {
        {0, 1, 1, 1, 1}, // 6
        {0, 0, 1, 1, 1}, // 5
        {0, 0, 0, 1, 1}, // 4
        {0, 0, 0, 0, 1}, // 3
        {1, 0, 1, 1, 1}, // 2
        {1, 0, 0, 1, 1}, // 1
    }
}
```

```

    {1, 1, 0, 1, 1},    // 0
    {1, 1, 0, 0, 1},    // -1
    {1, 1, 1, 0, 1},    // -2
    {1, 0, 0, 0, 0},    // -3
    {1, 1, 0, 0, 0},    // -4
    {1, 1, 1, 0, 0},    // -5
    {1, 1, 1, 1, 0},    // -6  100 30
};

for (int i = 0; i < 13; i++) {
    bool match = true;
    for (int j = 0; j < 5; j++) {
        if (sensorValues[j] != errorValues[i][j]) {
            match = false;
            break;
        }
    }
    if (match) {
        error = i - 6;
        break;
    }
}
}

void pidCalculations() {
    pTerm = kp * error;
    integral += error;
    iTerm = ki * integral;
    derivative = error - previousError;
    dTerm = kd * derivative;
    output = pTerm + iTerm + dTerm;
    previousError = error;
}

void changeMotorSpeed() {
    // Check if all IR readings are 0
    if ((digitalRead(A1) == 0) && (digitalRead(A2) == 0) && (digitalRead(A3) == 0)
    && (digitalRead(A4) == 0) && (digitalRead(A5) == 0)) {

        analogWrite(3, 0);
        analogWrite(5, 0);
        digitalWrite(12, 0);
        digitalWrite(6, 0);
        digitalWrite(13, 0);
        digitalWrite(11, 0);
    }
}

```

```

} else {

    // Change motor speed of both motors accordingly
    motor2newSpeed = motor2Speed + output;
    motor1newSpeed = motor1Speed - output;
    int speedThresholds[] = {210, 195, 180, 165};

    for (int i = 0; i < sizeof(speedThresholds) / sizeof(speedThresholds[0]);
i++) {
        if (motor1newSpeed == speedThresholds[i]) {
            motor1newSpeed = s;
        }
        if (motor2newSpeed == speedThresholds[i]) {
            motor2newSpeed = s;
        }
    }

    // Constrain the new speed of motors to be between the range 0-255
    motor2newSpeed = constrain(motor2newSpeed, 0, 255);
    motor1newSpeed = constrain(motor1newSpeed, 0, 255);
    // Set new speed and run motors in the forward direction
    analogWrite(5, motor2newSpeed);
    analogWrite(3, motor1newSpeed);
    digitalWrite(12, 1);
    digitalWrite(6, 1);
    digitalWrite(13, 0);
    digitalWrite(11, 0);
}
}

void loop() {
    calculateError();
    pidCalculations();
    changeMotorSpeed();
    Serial.println(output);
    Serial.print("motor1newSpeed: ");
    Serial.println(motor1newSpeed);
    Serial.print("motor2newSpeed: ");
    Serial.println(motor2newSpeed);
    //delay(1000);
}

```

Buổi chiều

Chạy thử xe

- Xe chạy với tốc độ 95 tăng 120 xe chạy theo line nhưng đến vòng cua bị lệch khỏi vòng line
- Giảm tốc độ của xe

```
int speedThresholds[] = {210, 195, 180, 165};

for (int i = 0; i < sizeof(speedThresholds) / sizeof(speedThresholds[0]);
i++) {
    if (motor1newSpeed == speedThresholds[i]) {
        motor1newSpeed = s;
    }
    if (motor2newSpeed == speedThresholds[i]) {
        motor2newSpeed = s;
    }
}
```