PROJECT - 67

WIRELESS SEMI-AUTO OBJECT DETECTION WITH RADAR

GROUP NAME:

project-67

Github:https://github.com/SuChinapat/2568-CPE414-Project-67.git

Project description

แบบจำลองเรดาห์ โดย Servo motorจะหมุน 180 องศา โดยมี Ultrasonic sensor ติดอยู่ข้างบน สามารถใช้Joy stick ควบคุม Servo motor ให้หมุนแบบ manual เพื่อค้นหาวัตถุที่เข้ามาแบบ Real time ได้ และจะมีระบบแสดงหน้าจอเรดาห์ผ่านหน้าเว็บแจ้งเตือนด้วยเสียงด้วย Speaker เมื่อพบวัตถุในระยะ

Project description

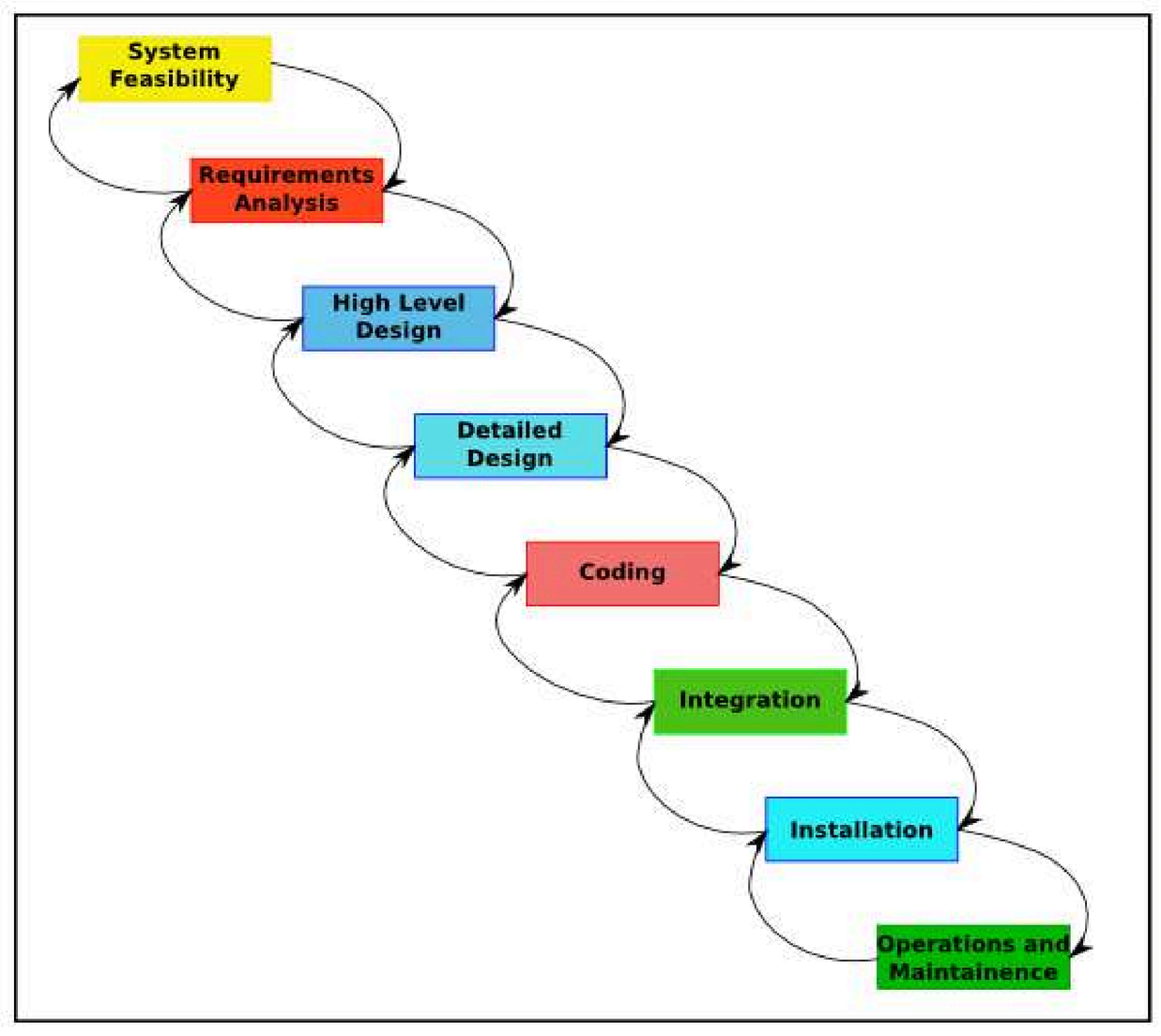
- **requirements**
- 1x Monitor
- 1x Ultrasonic sensor
- 1x Servo motor
- 1x PS2 XY Joy stick Module
- WIFI
- 2x ESP32
- 1x Speaker
- 1x mp3 tf 16p

- **Method Case**
- Servo motor หมุน 180 องศา โดยมี Ultrasonic sensor ติดอยู่ข้างบน
- สามารถใช้Joy stick ควบคุม Servo motor ให้หมุนแบบ manual เพื่อค้นหาวัตถุที่เข้ามาแบบ Real time ได้
- แสดงหน้าจอเรดาห์ ผ่านจอ Monitor และ แจ้งเตือนด้วยเสียงด้วย Speaker เมื่อพบวัตถุในระยะ
- ให้ ESP32 1 ตัวเป็นตัวควบคุมเรดาห์ อีก 1 ตัว เชื่อมต่อกับ Joy stick เพื่อ Remote เข้าไปควบคุมการหมุนของ เรดาห์

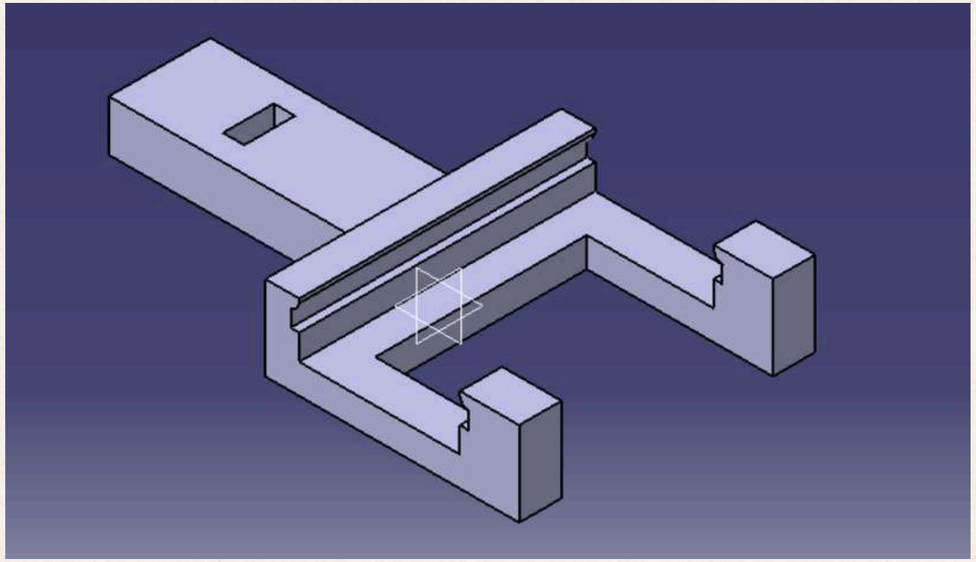
- **Test Case**
- Ultrasonic sensor ตรวจหาวัตถุในระยะ 40cm
- Servo morter ต้องหมุน 0-180 และ 180-0 องศา ตลอดเวลา
- เมื่อเจอวัตถุจะแสดงผลบนหน้าจอ Monitor และ มีเสียงแจ้งเตือน Speaker ภายใน 2 วินาที
- เมื่อใช้ Joy Stick ควบคุม Servo จะต้องหยุดการทำงานแบบอัตโนมัติและ หมุนตาม Joy Stick ภายใน 1 วินาที

Model

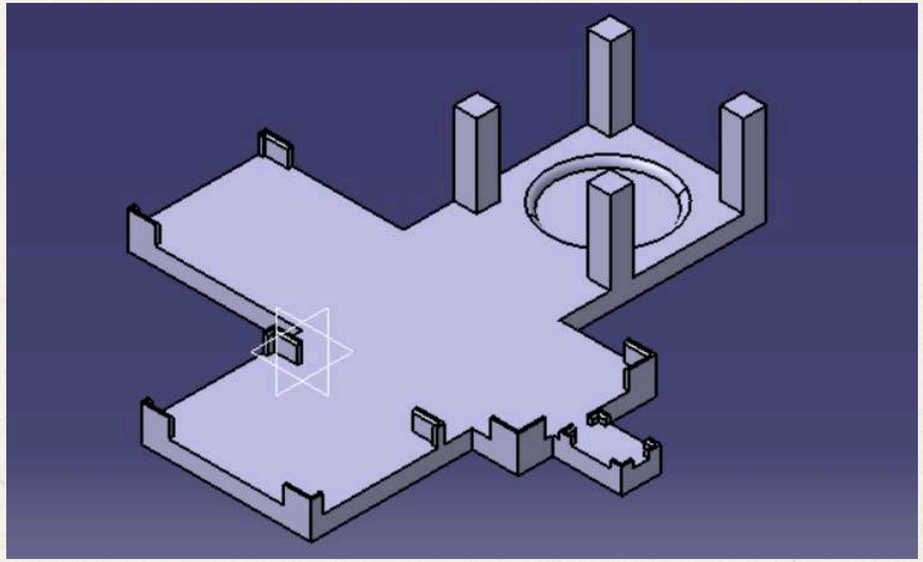
Adaptive waterfall model



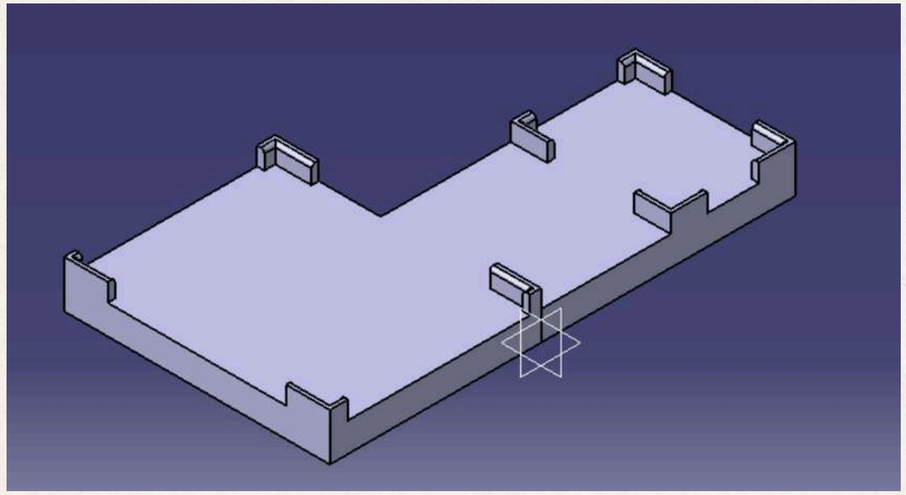
ลักษณะชิ้นงาน



part 1

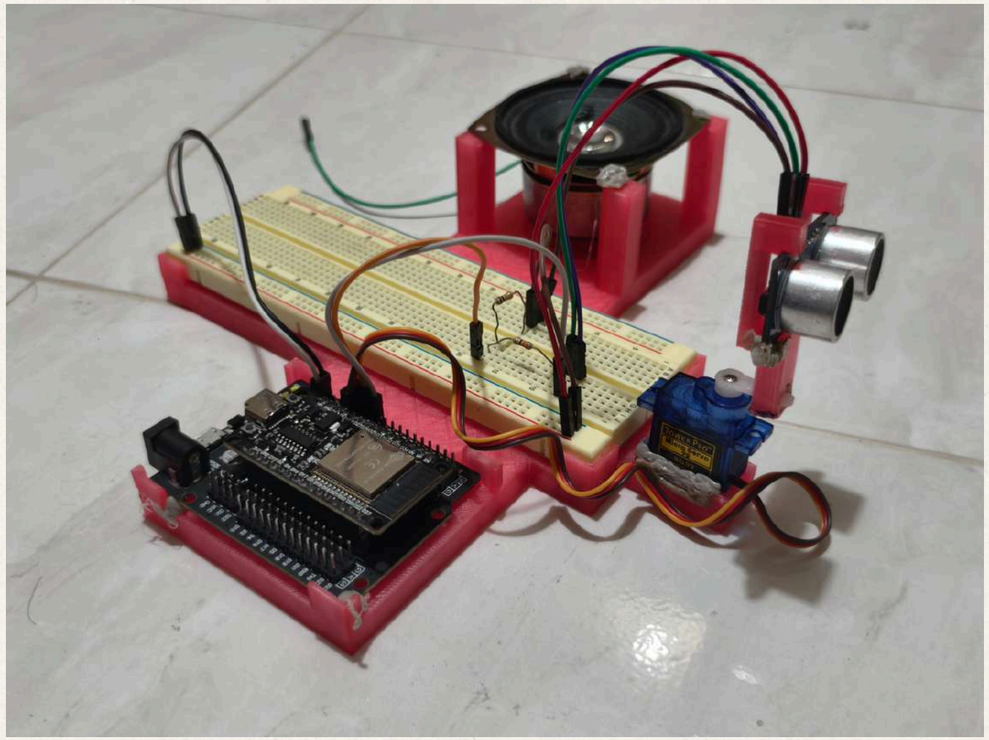


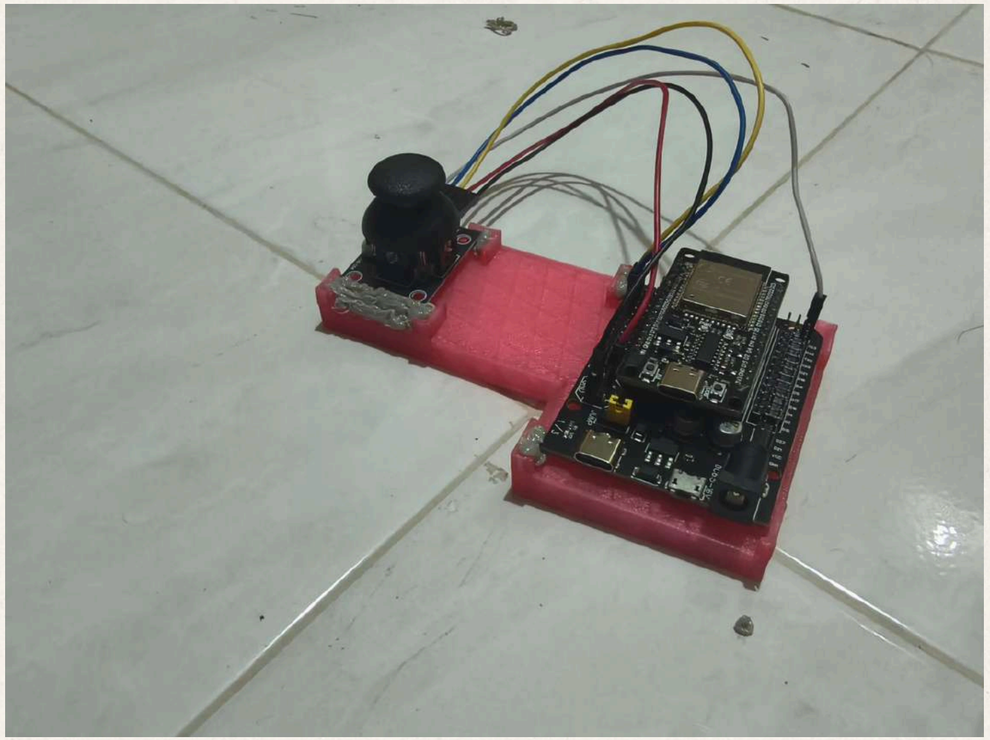
part 2



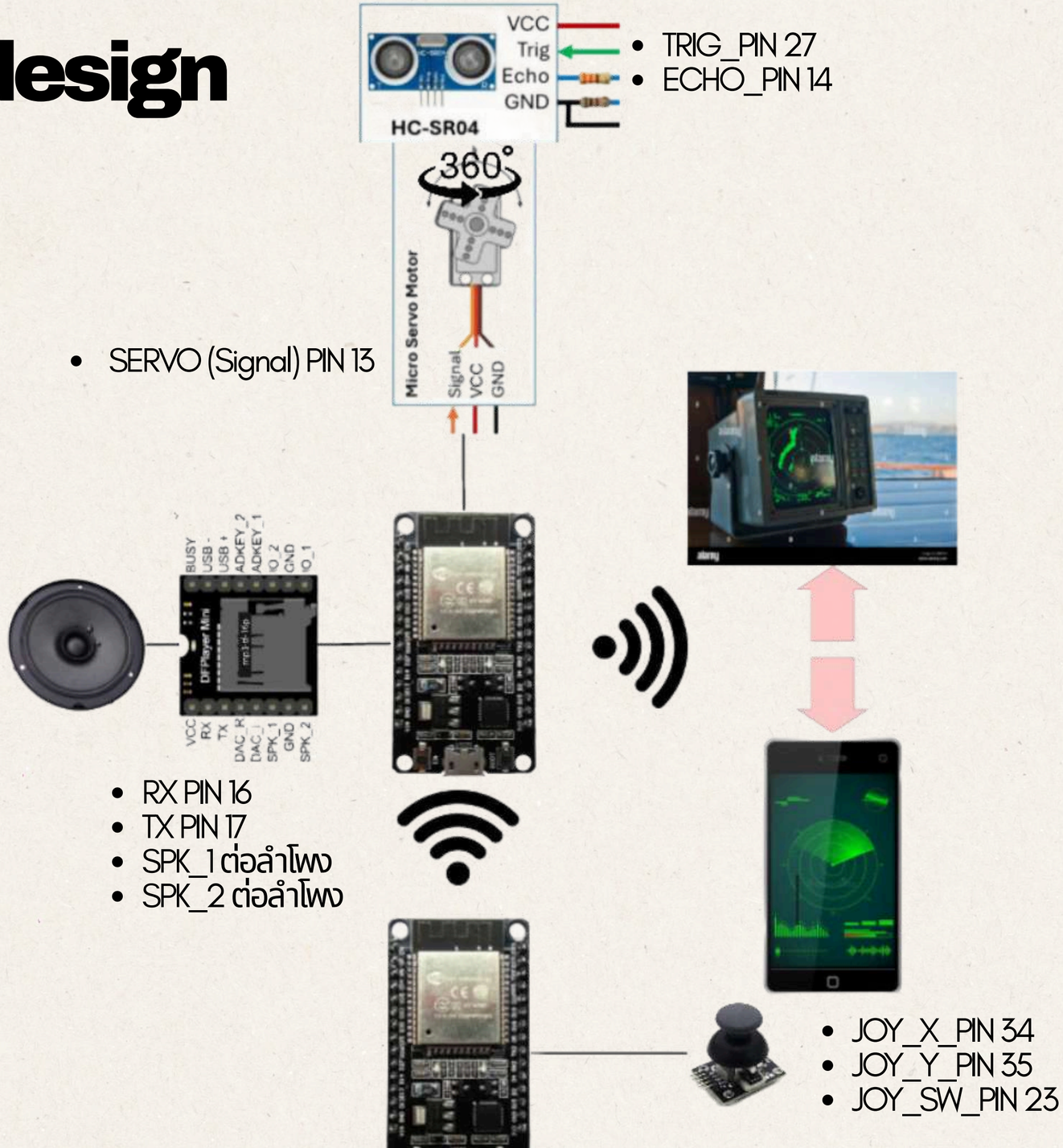
part 3

Hardware design





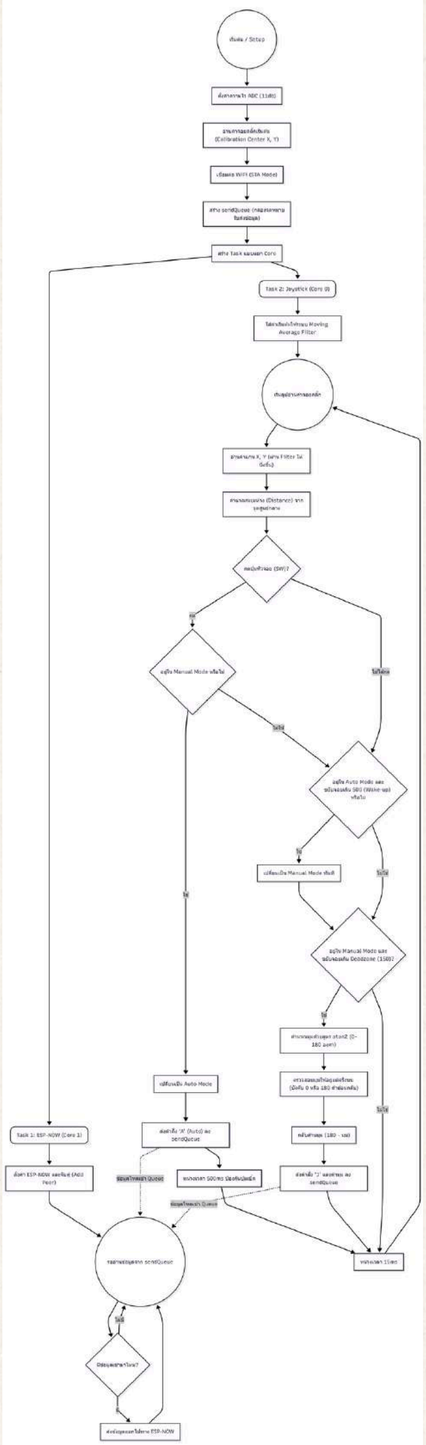
Architectural design and Hardware



- TRIG_PIN 27
- ECHO_PIN 14
- SERVO (Signal) PIN 13
- RX PIN 16
- TX PIN 17
- SPK_1 ต่อลำโพง
- SPK_2 ต่อลำโพง
- JOY_X_PIN 34
- JOY_Y_PIN 35
- JOY_SW_PIN 23

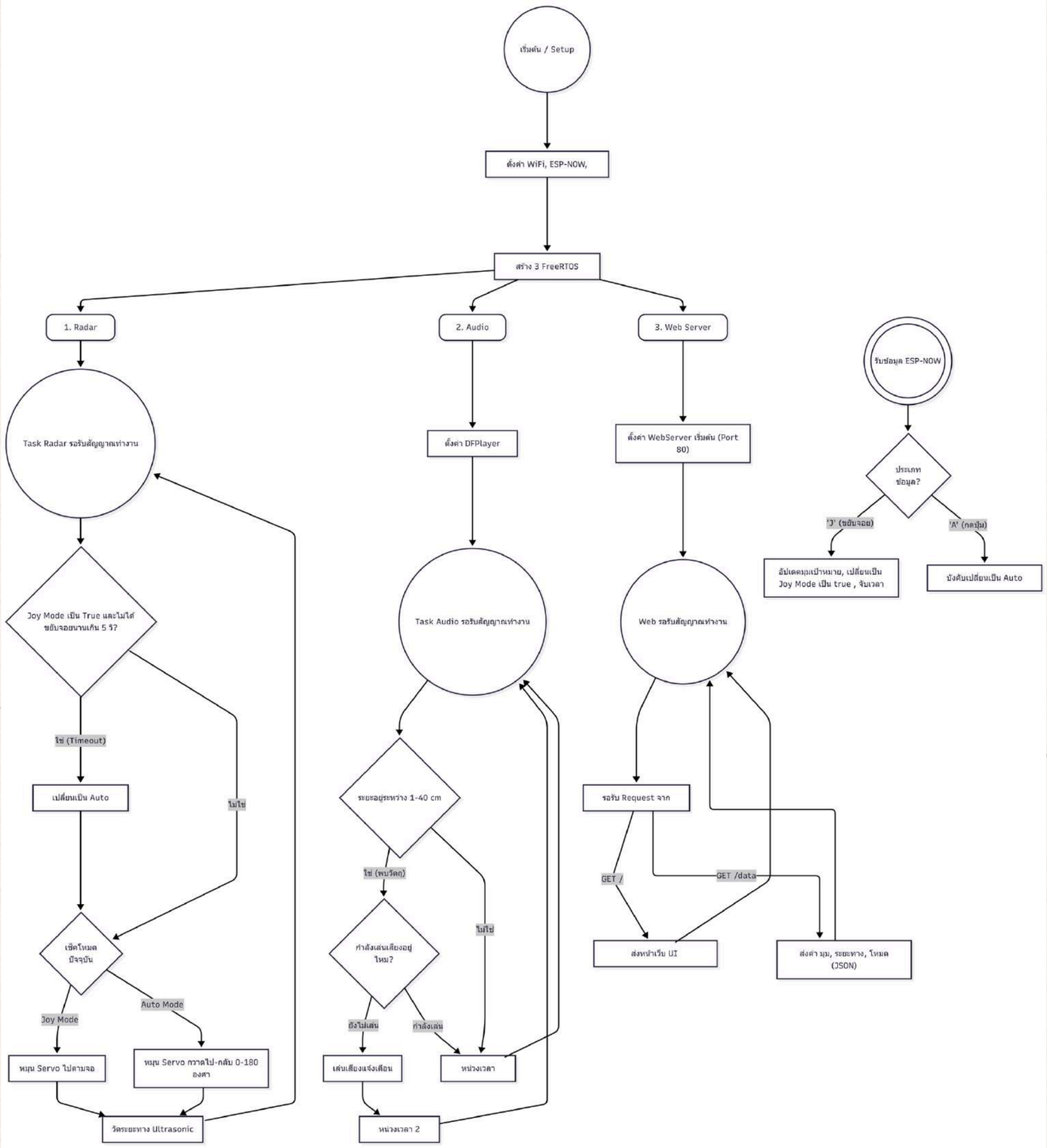
Flow Chart

Sender



Flow Chart

Receiver



Code

Sender

```
while (true)

{

  if (xQueueReceive(sendQueue, &msgToSend, portMAX_DELAY) == pdTRUE)

  {

    esp_now_send(broadcastAddress, (uint8_t *)&msgToSend, sizeof(msgToSend));
  }
}
```

Espnow Task

```
pinMode(JOY_SW_PIN, INPUT_PULLUP);
// Pre-fill filter with center values
for (int i = 0; i < FILTER_SIZE; i++)
{
  readingsX[i] = centerX;
  readingsY[i] = centerY;
}
totalX = (long)centerX * FILTER_SIZE;
totalY = (long)centerY * FILTER_SIZE;
while (true)
{
  int smoothX, smoothY;
  readJoystickSmooth(&smoothX, &smoothY);
  // X and Y mapping
  int mapX = centerX - smoothX;
  int mapY = centerY - smoothY;
  double distance = sqrt((double)mapX * mapX + (double)mapY * mapY);
  // 1. Button Logic (Mode Switch)
  if (digitalRead(JOY_SW_PIN) == LOW)
  {
    if (!inAutoMode)
    {
      inAutoMode = true;
      struct_message msg = {'A', 0};
      xQueueSend(sendQueue, &msg, 0);
      Serial.println(">> SYSTEM: AUTO MODE");
      vTaskDelay(pdMS_TO_TICKS(500));
    }
  }
  // 2. Wake-up Logic
  if (inAutoMode && distance > WAKE_UP_THRESHOLD)
  {
    inAutoMode = false;
    Serial.println(">> SYSTEM: MANUAL MODE");
  }
```

```
if (!inAutoMode && distance > DEADZONE)
{
  // Use atan2 to get the full angle
  double radian = atan2(mapY, mapX);
  int angle = (int)(radian * 180.0 / PI);
  // Map the joystick sweep to a 0-180 servo range
  // We clamp it here so only the upper arc works.
  // make sure that the angle is between 0 and 180
  if (mapY < 0)
  {
    if (mapX > 0)
      angle = 0;
    if (mapX < 0)
      angle = 180;
  }
  // Invert angle range: 0-180 becomes 180-0
  angle = 180 - angle;
  struct_message msg = {'J', angle};
  if (xQueueSend(sendQueue, &msg, 0) == pdTRUE)
  {
    Serial.printf("POS -> X:%d Y:%d | SERVO ANGLE:%d\n", mapX, mapY, angle)
  }
}

vTaskDelay(pdMS_TO_TICKS(15));
}
```

joystick Task

Code

Receiver

```
// ---------------- ESP-NOW CALLBACK ----------------
void OnDataRecv(const uint8_t * mac, const uint8_t *incomingData, int len) {
  memcpy(&myData, incomingData, sizeof(myData));

  // กรณี 1: ขยับจอย (Joystick Move)
  if (myData.type == 'J') {
    int raw = myData.value;

    if (raw > 180) raw = 180;
    if (raw < 0) raw = 0;

    // อัปเดตเป้าหมาย
    joyTargetAngle = raw;
    isJoyMode = true;
    lastJoyMoveTime = millis();
  }
  // กรณี 2: กดปุ่มหัวจอย (Button Press -> Auto)
  else if (myData.type == 'A') {
    isJoyMode = false; // บังคับกลับ Auto ทันที
  }
}
```

OnDataRecv

```
void radarTask(void *parameter) {
  myServo.attach(SERVO_PIN, 500, 2400);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);

  int sweepDir = 1;
  int autoAngle = 90;
  unsigned long lastScan = 0;

  while (true) {
    // 🔄 เช็ค Timeout: ถ้าไม่ขยับจอยนานเกินกำหนด ให้กลับ Auto
    if (isJoyMode && millis() - lastJoyMoveTime > JOY_TIMEOUT) {
      isJoyMode = false;
    }

    if (isJoyMode) {
      // 🎮 MANUAL MODE (แบบมี Smooth)
      // ถ้าอยากให้ไวสุดๆ ให้แก้บรรทัดนี้เป็น: smoothAngle = joyTargetAngle;
      smoothAngle = smoothAngle + (joyTargetAngle - smoothAngle) * 0.25;

      currentAngle = (int)smoothAngle;
      myServo.write(currentAngle);

      // วัดระยะ (ไม่ต้องถี่มาก เดี๋ยว Servo กระตุก)
      if (millis() - lastScan > 60) {
        currentDistance = measureDistanceFast();
        lastScan = millis();
      }
      vTaskDelay(15 / portTICK_PERIOD_MS);
```

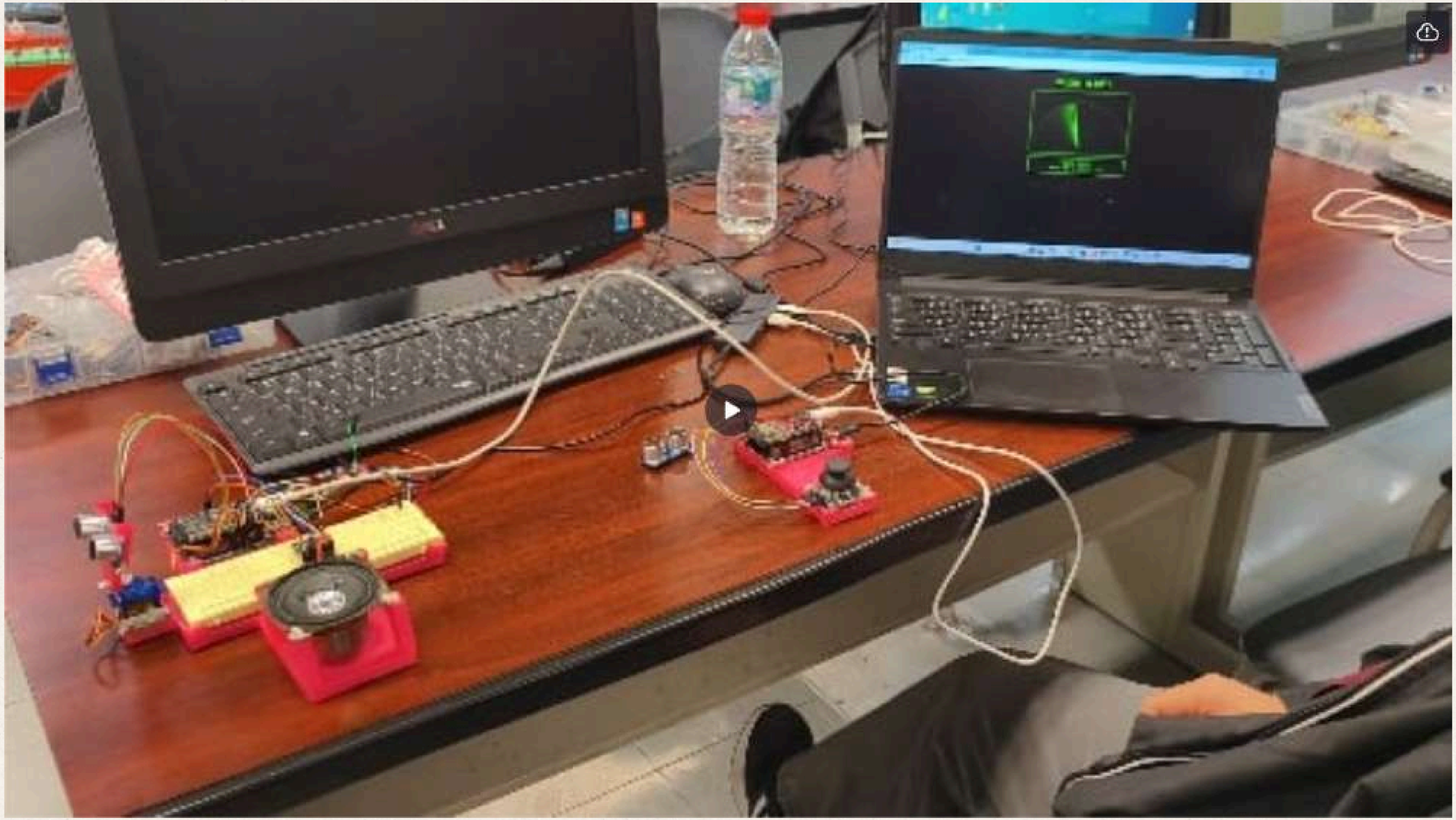
```
    } else {
      // 🔄 AUTO MODE
      autoAngle += sweepDir * 2;
      if (autoAngle >= 180) { autoAngle = 180; sweepDir = -1; }
      if (autoAngle <= 0)   { autoAngle = 0;   sweepDir = 1; }

      // Smooth Auto Movement
      smoothAngle = smoothAngle + (autoAngle - smoothAngle) * 0.2;
      currentAngle = (int)smoothAngle;
      myServo.write(currentAngle);

      currentDistance = measureDistanceFast();
      vTaskDelay(30 / portTICK_PERIOD_MS);
    }
  }
}
```

RadarTask

Test result



Budget

1. PS2 XY Joy stick Module ราคา 21 บาท
2. mp3 tf 16p manual ราคา 50 บาท

รวม 71 บาท

Gantt chart

TASK	RESPONSIBLE	PERIOD	PRIORITY	WEEK 1 (16/01 - 22/01)					WEEK 2 (23/01 - 29/01)					WEEK 3 (30/01 - 05/02)					WEEK 4 (06/02 - 12/02)					WEEK 5 (13/02 - 20/02)				
				FRI	MON	TUE	WED	THU	FRI	MON	TUE	WED	THU	FRI	MON	TUE	WED	THU	FRI	MON	TUE	WED	THU	FRI	MON	TUE	WED	THU
1) Discussion																												
1.1) Detail	Everyone	3	HIGH																									
1.2) Task	Everyone	3	HIGH																									
1.3) Circuit design	Big Pooh	2	MEDIUM																									
1.4) Flowchart	Big Pooh		MEDIUM																									
1.5) Hardware	Big Chat		HIGH																									
2) Coding																												
2.1) Monitor	Big Chat	10	LOW																									
2.2) Ultrasonic sensor	Big Chat	10	HIGH																									
2.3) Servo motor	Big Dan	10	HIGH																									
2.4) ลำโพง	Big Pooh	10	MEDIUM																									
2.5) WIFI	Big Pooh	10	HIGH																									
2.6) PS2 Joy stick	Big Dan	10	HIGH																									
2.7) marge code	Everyone	2	HIGH																									
2.8) test & debug	Everyone	1	MEDIUM																									
3) Conclusion																												
3.1) presentation	Everyone		HIGH																									
			HIGH																									
			HIGH																									
			HIGH	กำลังดำเนินการ		สำเร็จ		แพลน	กำลังดำเนินการแต่ไม่ตรงกับที่แพลนไว้																			

นาย ชยาพันธ์ วิโรจน์ชัยยันต์

members' roles & responsibilities

- Monitor
- Ultrasonic sensor

นาย วีรพงษ์ ศรีสอาด

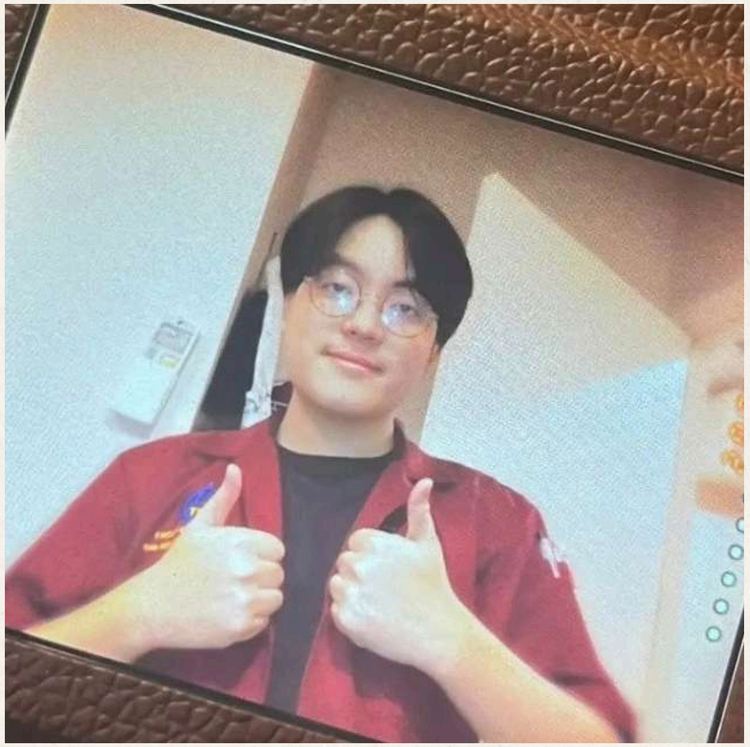
members' roles & responsibilities

- Speaker
- WIFI

นาย ชินพัฒน์ สุพรรณพงศ์

members' roles & responsibilities

- Servo moter
- PS2 joystick



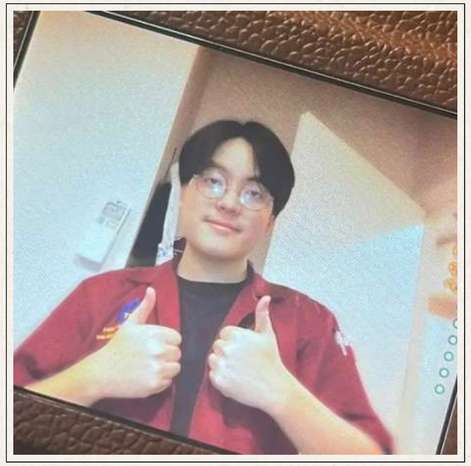
Project members

นาย ชยาพันธ์ วิโรจน์ชัยยันต์

ID: 2311311530

นาย วีรพงษ์ ศรีสอาด

ID: 2311311316



นาย ชินพัฒน์ สุพรรณพงศ์

ID: 2311311480

Thank you