

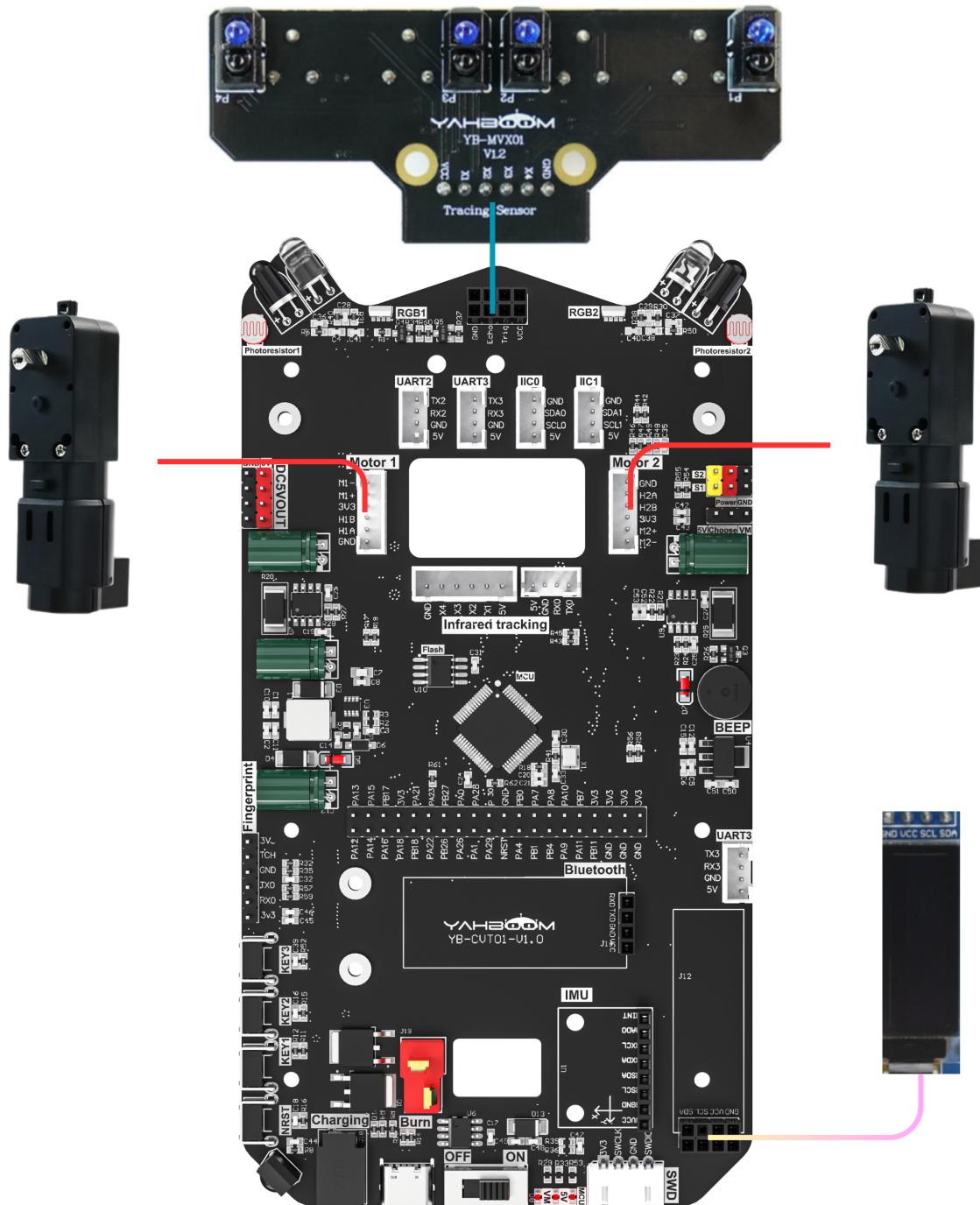
Drawing a Circle - 4 Channel Infrared Tracking Sensor Module

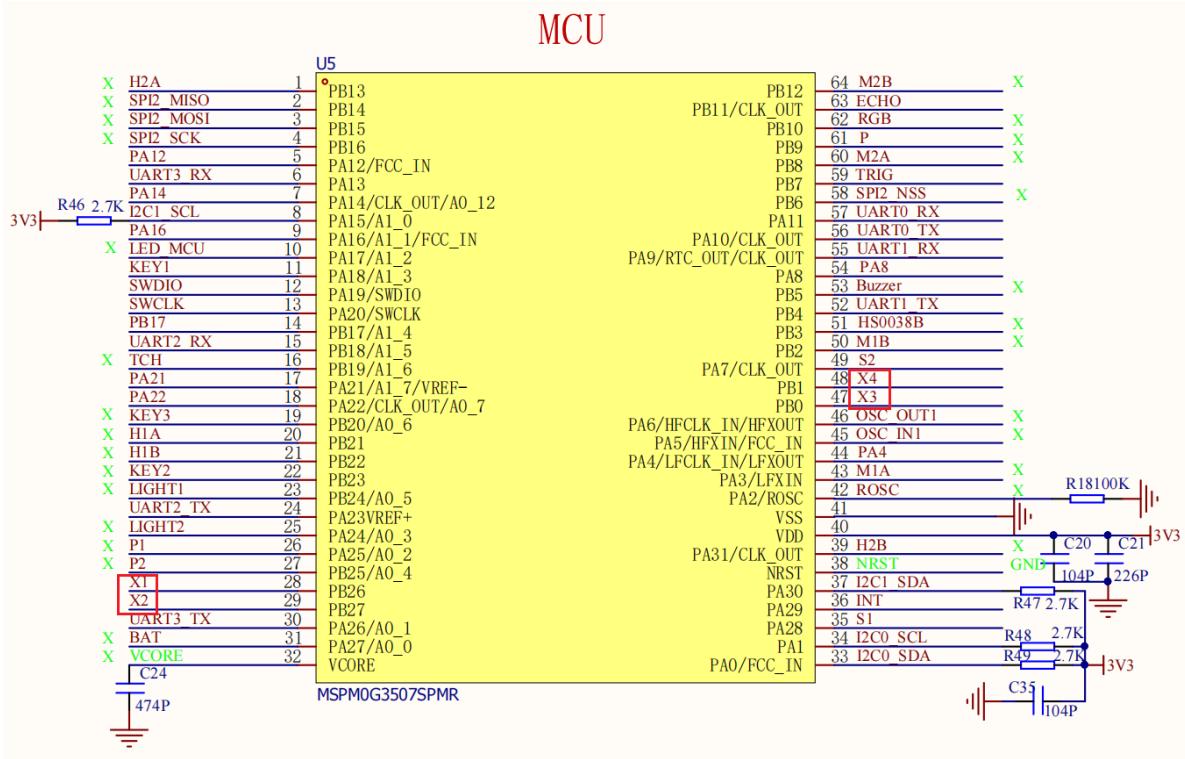
Drawing a Circle - 4 Channel Infrared Tracking Sensor Module

1. Hardware Connection
2. Partial Code Analysis
3. Main Functions
4. Experimental Phenomenon

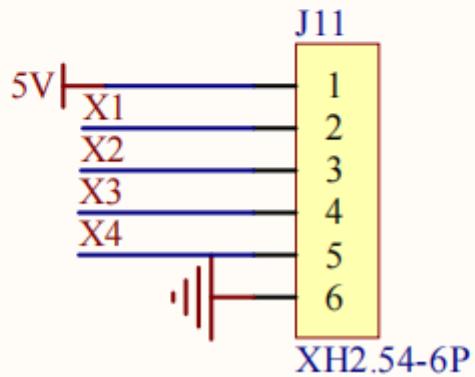
1. Hardware Connection

4 Channel Infrared Tracking Sensor Module	MSPM0G3507
5V	5V
GND	GND
X1	X1
X2	X2
X3	X3
X4	X4



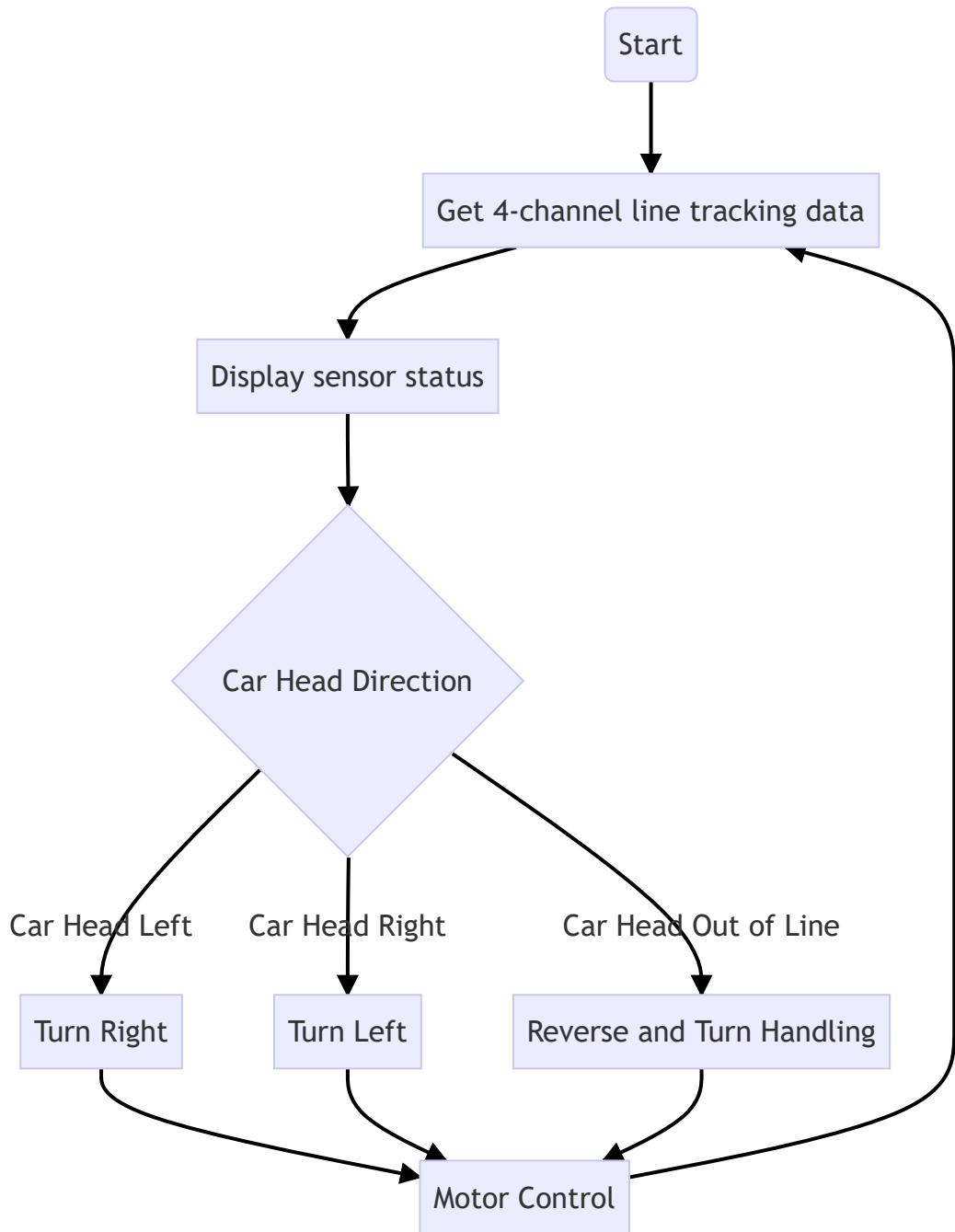


Four-channel line patrol interface



2. Partial Code Analysis

Control Principle



Four_linenwalking.c

```

//带死区处理的位置式PID / Positional PID with dead zone processing
float APP_IR_PID_calc(int8_t actual_value)
{
    float IRTrackTurn = 0;
    int8_t error;
    static int8_t error_last=0;
    static float IRTrack_Integral;//积分 / Integral

    error=actual_value;

    IRTrack_Integral +=error;
  
```

```

//位置式pid / Positional PID
IRTrackTurn=error*IRTrack_Trun_KP
    +IRTrack_Trun_KI*IRTrack_Integral
    +(error - error_last)*IRTrack_Trun_KD;

}

if (IRTrackTurn > (MAX_SPEED - MOTOR_DEAD_ZONE))
    IRTrackTurn = (MAX_SPEED - MOTOR_DEAD_ZONE);
if (IRTrackTurn < (MOTOR_DEAD_ZONE - MAX_SPEED))
    IRTrackTurn = (MOTOR_DEAD_ZONE - MAX_SPEED);
return IRTrackTurn;
}

//得到四路循迹模块的数据 / Get data from 4-channel line tracking module
void Four_GetLinewalking(int *LineL1, int *LineL2, int *LineR1, int *LineR2)
{
    *LineL1 = Linewalk_L1_IN;
    *LineL2 = Linewalk_L2_IN;
    *LineR1 = Linewalk_R1_IN;
    *LineR2 = Linewalk_R2_IN;
}

//画地为牢循迹反逻辑 / Drawing a circle line tracking reverse logic
void Linewalking(void)
{
    int8_t err = 0;
    Four_GetLinewalking(&LineL1, &LineL2, &LineR1, &LineR2); //获取黑线检测状态
    / Get black line detection status
    if(LineL1 == 0 && LineR1 == 0)
    {
        err = 70;
        // 应用PID控制 这里根据实际情况调节 / Apply PID control, adjust according
        to actual situation here
        pid_output_IRR = (int)(APP_IR_PID_Calc(
            -err));
        Motion_Car_Control(-IRR_SPEED, 0, pid_output_IRR);
        delay_ms(400);
        return ;
    }
    else if(LineL2 == 0 && LineR2 == 1) err = -30; // 左传感器检测到黑线，右传感器
    未检测到黑线 / Left sensor detects black line, right sensor does not detect black
    line
    else if(LineL2 == 0 && LineL1 == 0) err = -40; // 左传感器和中间传感器都检
    测到黑线 / Left sensor and middle sensor both detect black line

    else if(LineR2 == 0) err = 30; // 右传感器检测到黑线 / Right sensor detects
    black line
    else if(LineR2 == 0 && LineR1 == 0) err = 40; // 右传感器和中间传感器都检测
    到黑线 / Right sensor and middle sensor both detect black line
    else if(LineL2 == 1 && LineR2 == 1) err = 0; // 两侧传感器都未检测到黑线 /
    Both left and right sensors do not detect black line
    else err = 0;

    // 应用PID控制 / Apply PID control
    pid_output_IRR = (int)(APP_IR_PID_Calc(
        -err));

    // 控制小车运动 / Control car movement
}

```

```

    Motion_Car_Control(IRR_SPEED, 0, pid_output_IRR);

}

```

3. Main Functions

LineWalking

Function Prototype	void LineWalking(void)
Function Description	Four-sensor line tracking control based on "drawing a circle" logic: gets four-sensor status, calculates deviation value err based on different detection conditions. When LineL1 and LineR1 both detect black lines, triggers special control (reverse speed + PID adjustment with delay); other conditions set err based on left/right sensor status, controls car movement after PID calculation (straight line speed is IRR_SPEED)
Input Parameters	None
Return Value	None

4. Experimental Phenomenon

After connecting the car properly, connecting the OLED module, and burning the program to MSPM0, place the car on a white background with black lines map, the car will move in a circular pattern within the bounded area, and sensor data will be displayed on the OLED

Where L2, L1, R1, R2 represent the four sensor data from left to right

