Chassis_Controller代码思路

> 更新：2023.3.6
>
> github链接：https://github.com/himcrh/Simple_Chassis_Controller
>
> 有问题可以提出Issues

> 使用操作系统freertos，创建四个进程用于不同方面

Chassis_Run

作用

控制底盘运动，完成任务流程，和电机的控制放在一起，5ms为一个分度值

优缺点

- 运动的逻辑与电机的控制没有分开，调试不好有可能会造成电机控制的阻塞导致其疯转。
- 但通过此种方式可以在调试过程中直观的看到此进程的哪个状态有问题，方便调试

实现方式

通过`Move_Info.Move_Status`进行状态机切换，进入不同的控制进程。

注：`Move_Task`为主进程，其余基本为debug使用

Move_Status_NewTask函数思路

```
switch(Move_Info.Process_Status){
        /*出发到扫二维码*/
        case 0:{
            if(Scan_Qr_Code(&VX,&VY,&VZ))
                Move_Info.Process_Status = 1;
            break;
        }
        case 1:{
            if(Visual_Recognition(&VX,&VY,&VZ))
                Move_Info.Process_Status = 2;
            break;
        }
        ...
```

使用`Move_Info.Process_Status`进行状态机控制，同时其中的每个函数为bool返回值类型，其中也为状态机控制，若满足需求则返回true，退出，进入下一个状态。

```
    switch(Program_Counter){
        case 0:{
            *VY = 2000;
            if(Rt_Line[3].Pin_State)
                Program_Counter = 1;
            break;
        }
        case 1:{
            if(Line_Patrol_VX(VX,VY,VZ,5500,2))
                Program_Counter = 2;
            break;
        }
        case 2:{
            if(Line_Patrol_VY(VX,VY,VZ,5500,2))
                Program_Counter = 3;
            break;
        }
        case 3:{
```

```
            HAL_UART_Receive_IT(&huart1,Uart_Rx_Buffer,9);
            /*扫码成功*/
            if(Uart_Rx_Buffer[7]==0x0d &&
Uart_Rx_Buffer[8]==0x0a){
                function_init = true;
                return true;
            }
            break;
        }
    }
    return false;
```

如这个扫码函数，使用了`Program_Counter`状态机控制运动。达到`case3`，扫码成功则返回`true`，上层状态机会进行状态切换。

Line_Refresh

作用

对巡线灰度管的状态进行侦测，对相关结构体进行更新，此结构体更新会影响底盘运动控制进程中状态的更新。

实现方式

两个关键结构体类型

```
typedef struct Line_IO{

    GPIO_TypeDef * IO_Port;          //Pin Port
    uint16_t IO_Pin;                 //Pin
    bool Pin_State;                  //State
    bool Triggered;                  //是否触发过
```

```
}Line_IO;

typedef struct Handle_IO{

    bool All_Pin_Up;                //全部触发
    bool All_Pin_Down;              //全部未触发
    bool All_Pin_Triggered;         //全部触发过
    int8_t Cur_Counter;             //当前触发个数

}Handle_IO;
```

缺点

由此方法区分了前后左右四个巡线模块，而不是采用一个通用二维数组储存，造成了代码的冗余（即不方便使用循环的方式直接遍历四个方向的巡线）

Arm_Move_Ctrl

作用

机械臂运动控制，状态机的控制方式同`Chassis_Run`进程。同时将控制与逻辑分开，避免长时间阻塞。

Arm_Move_Base

作用

机械臂舵机控制，防止`Arm_Move_Ctrl`中的阻塞对舵机的控制造成影响。