中心主题

- 中断关闭，初始化
- init初始化
  - ADC_all_init — 00 01 05 06 二分频 设置为ADC引脚
    - dg_state
      - real
        - dg_state.L_zx_real = 0;
        - dg_state.L_zh_real = 0;
        - dg_state.L_zs_real = 0;
        - dg_state.L_ys_real = 0;
        - dg_state.L_yh_real = 0;
        - dg_state.L_yx_real = 0;
      - max
        - dg_state.L_zx_max = 33;
        - dg_state.L_zh_max = 2183;
        - dg_state.L_zs_max = 0;
        - dg_state.L_ys_max = 0;
        - dg_state.L_yh_max = 1831;
        - dg_state.L_yx_max = 1700;
      - once
        - dg_state.L_zx_once = 0;
        - dg_state.L_zh_once = 0;
        - dg_state.L_zs_once = 0;
        - dg_state.L_ys_once = 0;
        - dg_state.L_yh_once = 0;
        - dg_state.L_yx_once = 0;
      - twice
        - dg_state.L_zx_twice = 0;
        - dg_state.L_zh_twice = 0;
        - dg_state.L_zs_twice = 0;
        - dg_state.L_ys_twice = 0;
        - dg_state.L_yh_twice = 0;
        - dg_state.L_yx_twice = 0;
    - err_steering
      - err_steering.Err_x = 0;
      - err_steering.Err_h = 0;
      - err_steering.Err_s = 0;
      - err_steering.Err = 0;
      - err_steering.Err_last = 0;
      - err_steering.Err_x_last = 0;
      - err_steering.Err_h_last = 0;
      - err_steering.Errsum = 0;
      - err_steering.Errdif = 0;
  - GPIO初始化P13 推挽输出 作为其他电源
  - PWM初始化 — P60 62 64 66 17000hz 50%
  - 舵机初始化 — P74，50hz, 750
  - TIM4定时器初始化 — 20ms中断一次
  - PID初始化
    - pid_steering
      - pid_steering.p_steering = 5;
      - pid_steering.i_steering = 5;
      - pid_steering.d_steering = 5;
      - pid_steering.imax = 5;
      - pid_steering.imin = -5;
      - pid_steering.PID_STEERING_OUT = 750;
    - pid_motor
      - pid_motor.PID_MOTOR_R_OUT = 0;
      - pid_motor.PID_MOTOR_L_OUT = 0;
      - pid_motor.d_motor = 1;
      - pid_motor.i_motor = 0.5;
      - pid_motor.p_motor = 2;
  - 舵机初始化 — Steering_init
    - PWM_Steering_now = 750;
    - PWM_Steering_Max = 850;
    - PWM_Steering_Min = 650;
  - 电机初始化 — Motor_init
    - speed_state
      - speed_state.Outgar_speed_L_ai = 0;
      - speed_state.Outgar_speed_R_ai = 0;
      - speed_state.Strai_speed_L_ai = MFBL*0.1;
      - speed_state.Strai_speed_R_ai = MFBL*0.1;
      - speed_state.Cur_speed_L_ai = 0;
      - speed_state.Cur_speed_R_ai = 0;
      - speed_state.Cross_speed_L_ai = 0;
      - speed_state.Cross_speed_R_ai = 0;
      - speed_state.Rampin_speed_L_ai = 0;
      - speed_state.Rampin_speed_R_ai = 0;
      - speed_state.Ramp_speed_L_ai = 0;
      - speed_state.Ramp_speed_R_ai = 0;
      - speed_state.Ring_speed_L_ai = 0;
      - speed_state.Ring_speed_R_ai = 0;
      - speed_state.Ringin_speed_L_ai = 0;
      - speed_state.Ringin_speed_R_ai = 0;
      - speed_state.Ringout_speed_L_ai = 0;
      - speed_state.Ringout_speed_R_ai = 0;
      - speed_state.Three_speed_L_ai = 0;
      - speed_state.Three_speed_R_ai = 0;
      - speed_state.Threein_speed_L_ai = 0;
      - speed_state.Threein_speed_R_ai = 0;
    - speed_now
      - speed_now.speed_L = 0;
      - speed_now.speed_L_ai = 0;
      - speed_now.speed_R = 0;
      - speed_now.speed_R_ai = 0;
    - PWM_Motor_Max = 9000;
    - PWM_Motor_Min = 1000;
    - PWM_Motor_L_now = 2000;
    - PWM_Motor_R_now = 2000;
  - 编码器初始化 — Encoder_init
    - ctimer_count_init
      - Encoder_L
      - Encoder_R
    - err_motor
  - 无线串口初始化 — WIRELESS_init
- 总中断开启，TIM4中断时执行Control_All
  - road道路判断 — 返回 直走
  - ADC采样计算 — 通过ADC_ReadAverage读取 10次平均 8位ADC
    - 引脚
      - P00
      - P01
      - P05
      - P06
    - 涉及更新结构体
      - DG_State
        - real
        - once
        - twice
      - Err_Steering
        - Err
        - Errsum
        - Errdif
        - Err_last
        - Err_h_last
        - Err_x_last
    - 数据处理流程
      - 更新读取位real值
      - 小于1时赋值1 — 得到过一值real_x
      - 更新第一次归一化 — once_x=(过一值real_x)/(实测最大值) 有几个电感就有几个x
      - 更新第二次归一化 — 求偏差的占比 once_a/(once_a+once_b+...+once_x)
      - 左右电感做差，并计算更新权重误差Err
        - 横电感差
        - 斜电感差 (matlab拟合和距离的关系)
        - 得到误差Err，并赋予权重 — -0.6151*(Err_斜) + 3.3868*(Err_横);
      - 更新累计误差和Errsum — Errsum +=Err;
      - 更新权重误差和上一次权重误差的差 Errdif — Err-Err_last
      - 更新存为上一次误差
        - 上一次权重误差Err_last
        - 上一次横竖误差Err_横斜_last
  - 舵机控制
    - 计算Pid_Steering_Calculate
      - 引脚 — p73引脚
      - 涉及更新结构体
        - Err_Steering
        - PID_Steering — PID_STEERING_OUT
      - 数据处理
        - 限制幅度?
          - pid_steering->imin
          - pid_steering->imax
        - 更新pid_steering->PID_STEERING_OUT — (p_steering * Err) + (i_steering * Errsum) + (d_steering * Errdif);
    - 实际控制STEERING_Control — 数据处理
      - pwm_duty 输出
        - PID_STEERING_OUT+750
        - p73引脚
      - 限制幅度，防止舵机过偏，更新新的PID_STEERING_OUT
        - pid_steering->imin
        - pid_steering->imax
  - 电机控制
    - speedout选择道路结构体，赋值速度
      - 涉及更新结构体
        - Road — Straight
        - SPEED_state
          - Strai_speed_L_ai
          - Strai_speed_R_ai
          - Cur_speed_L_ai
          - Cur_speed_R_ai
        - SPEED_now
          - speed_L_ai
          - speed_R_ai
      - 数据处理 — switch road
        - Straight
          - 更新speed_L_ai — Strai_speed_L_ai
          - 更新speed_L_ai — Strai_speed_R_ai
        - Curve
          - 更新speed_L_ai — Cur_speed_L_ai
          - 更新speed_L_ai — Cur_speed_R_ai
    - speed_cal通过编码器数据计算速度，写入当前速度结构体SPEED_now
      - 涉及更新结构体 — SPEED_now
        - speed_L
        - speed_R
      - 数据处理
        - 读取编码器ctimer_count_read
          - Encoder_L — dat_L
          - Encoder_R — dat_R
        - 判断 if
          - 左轮正转 — 使用dat_L — 更新speed_L — MFBL*(dat_L/(CONTROL_T*DECO));
            - MFBL — 控制PID精度级别 — 10000
            - DECO — 1m脉冲数 — 11587
            - CONTROL_L — 中断控制周期，为定时器20ms — 0.02
          - 右轮正转 — 使用(-1)*dat_R — 更新speed_R — MFBL*(dat_R/(CONTROL_T*DECO));
            - MFBL — 控制PID精度级别 — 10000
            - DECO — 1m脉冲数 — 11587
            - CONTROL_L — 中断控制周期，为定时器20ms — 0.02
          - 反转左 — 使用 (-1)*dat_L — 更新speed_L
          - 反转右 — 使用 dat_R — 更新speed_R
        - 清理计数 — ctimer_count_clean
          - Encoder_R
          - Encoder_L
    - calculate_err_m 计算偏差值写入电机误差结构体
      - 涉及更新结构体
        - SPEED_now
          - speed_R_ai — 读取
          - speed_L_ai — 读取
          - speed_L — 读取
          - speed_R — 读取
        - Err_Motor
          - err_L_m — 更新
          - err_R_m — 更新
          - err_derivative_L_m — 更新
          - err_derivative_R_m — 更新
          - err_derivative2_L_m — 更新
          - err_derivative2_R_m — 更新
      - 数据处理
        - 更新err_R_m — speed_L_ai-speed_L
        - 更新err_L_m — speed_R_ai-speed_R
        - 更新本次误差和上次误差的差err_derivative_L_m — err_L_m-err_last_L_m
        - 更新本次误差和上次误差的差err_derivative_R_m — err_R_m-err_last_R_m
        - 更新上次误差和上上次误差的差err_derivative2_L_m — err_last_L_m-err_past_L_m
        - 更新上次误差和上上次误差的差err_derivative2_R_m — err_last_R_m-err_past_R_m
        - 更新上次误差和上上次误差
          - err_past_L_m = err_last_L_m
          - err_last_L_m = err_L_m
          - err_past_L_m = err_last_L_m
          - err_last_L_m = err_L_m
    - Pid_Motor_Calculate 计算PID输出到电机 — 涉及结构体
      - Err_Motor
        - err_derivative_L_m — 读取
        - err_derivative_R_m — 读取
        - err_L_m — 读取
        - err_R_m — 读取
        - err_derivative2_L_m — 读取
        - err_derivative2_R_m — 读取
      - PID_Motor
        - PID_MOTOR_L_OUT — **输出**
        - PID_MOTOR_R_OUT — **输出**
        - p_motor — 读取
        - i_motor — 读取
        - d_motor — 读取
      - 数据处理
        - PID_MOTOR_L_OUT — (p_motor *err_derivative_L_m) + (i_motor * err_L_m) + (d_motor * derivative2_L_m);
        - PID_MOTOR_R_OUT — (p_motor *err_derivative_L_m) + (i_motor * err_L_m) + (d_motor * derivative2_L_m);
- 分支主题4
- 结构体
  - typedef struct
  - 子主题4
  - 结构体
    - float Err_x;
    - float Err_h;
    - float Err_s;
    - float Err;
    - float Err_last;
    - float Err_x_last;
    - float Err_h_last;
    - float Errsum;
    - float Errdif;