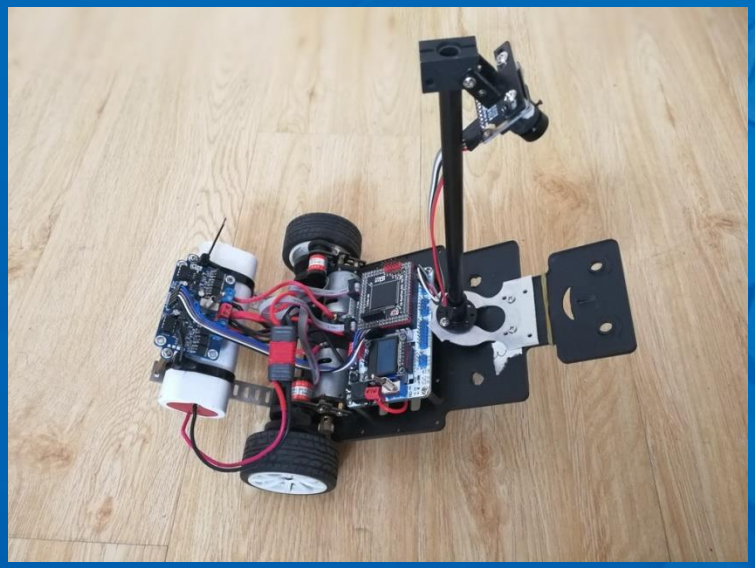
龙邱智能科技
LONGQIUZHINENGKEJI

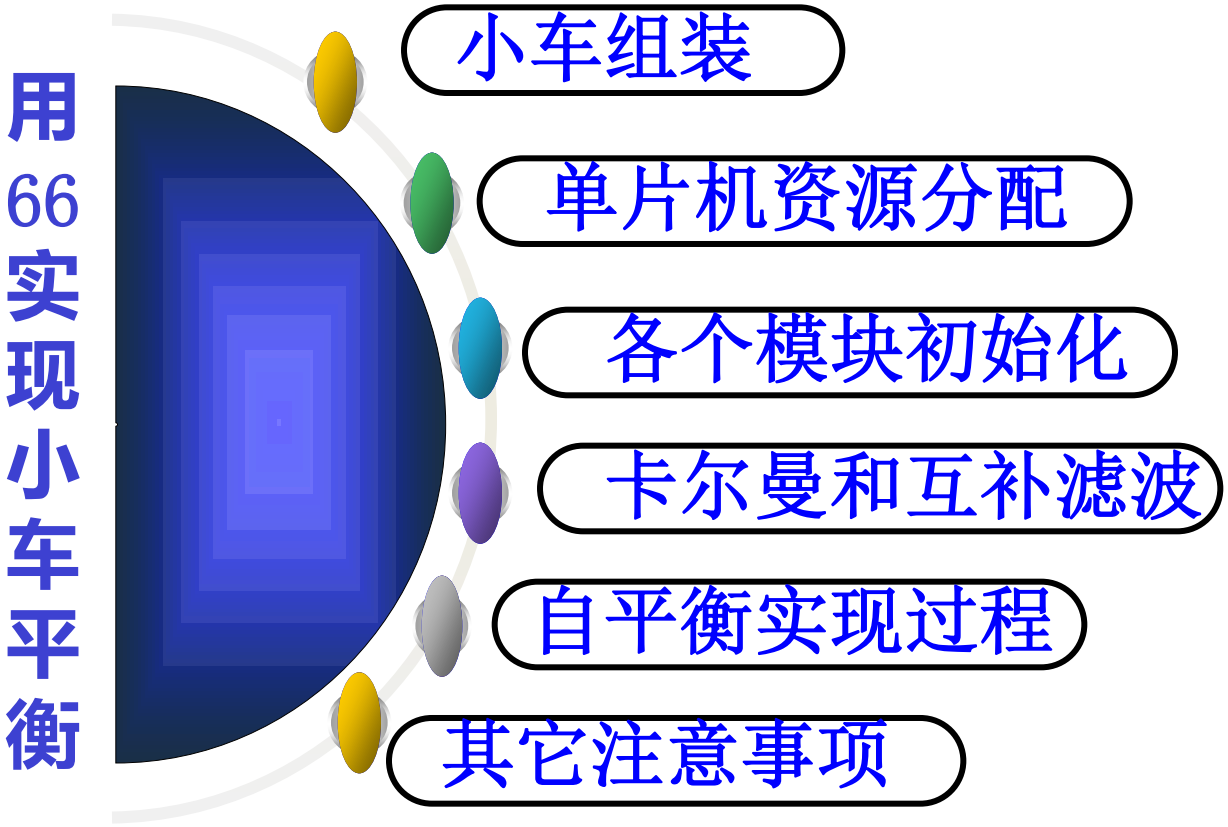
用K66单片机实现小车自平衡

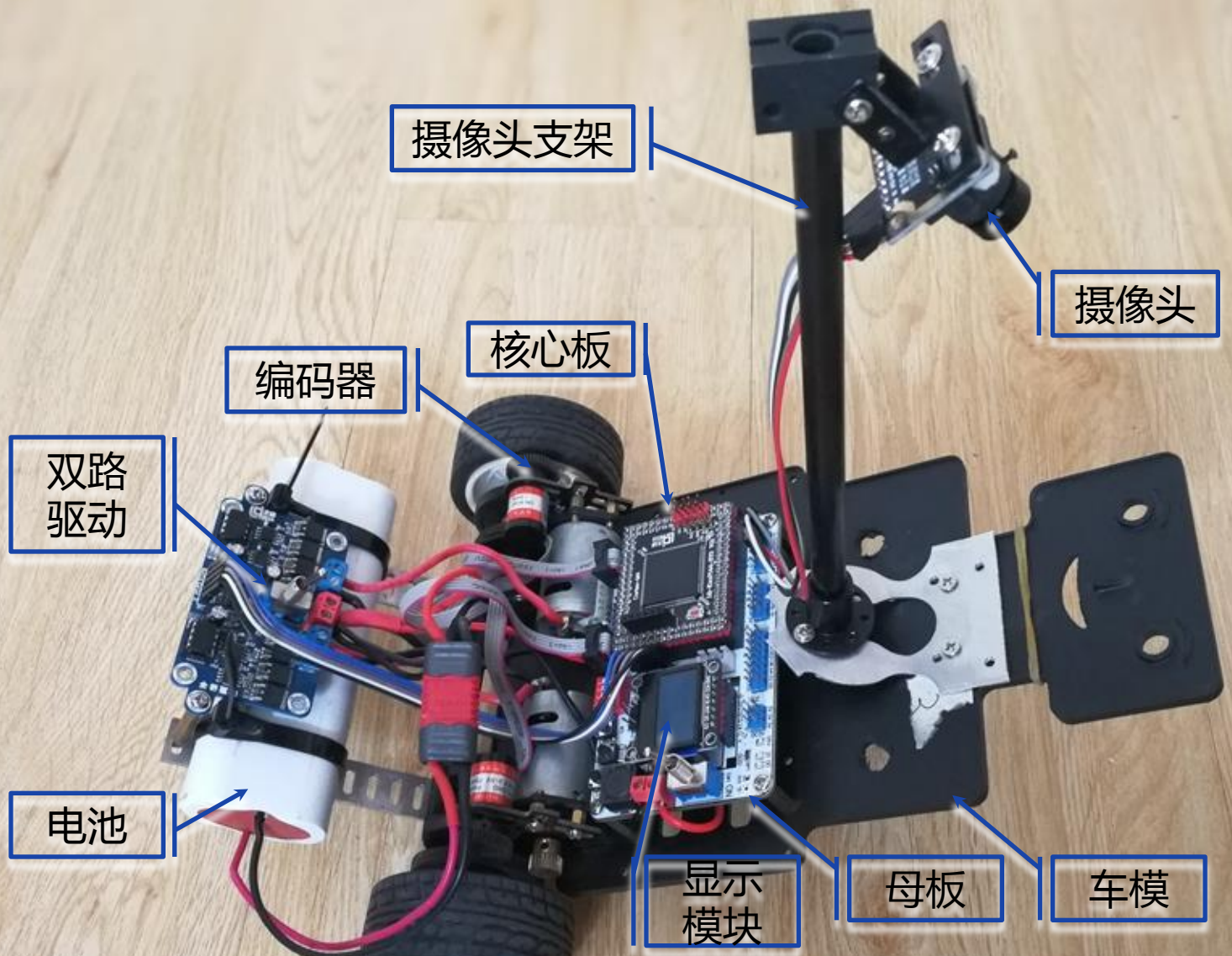


主讲：chiusir

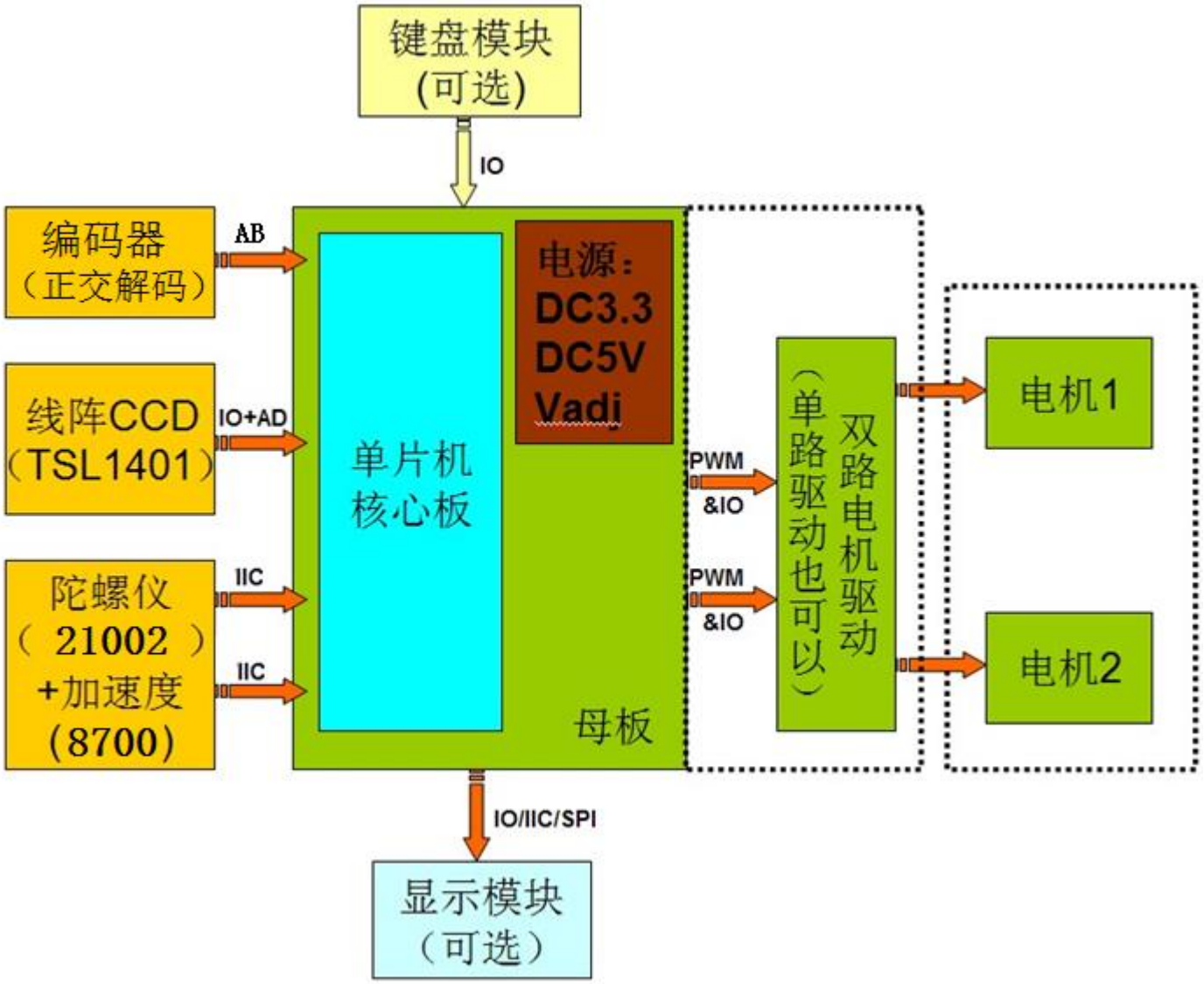
20170312

主要内容




摄像头支架
摄像头
核心板
编码器
双路
驱动
电池
显示
模块
母板
车模

一、自平衡小车构成


键盘模块
(可选)
IO
编码器
（正交解码）
AB
线阵CCD
（TSL1401）
IO+AD
陀螺仪
（ 21002 ）
+加速度
(8700)
IIC
IIC
单片机
核心板
电源：
DC3.3
DC5V
Vadj
母板
PWM
&IO
PWM
&IO
双路电机驱动
（单路驱动也可以）
电机1
电机2
IO/IIC/SPI
显示模块
（可选）

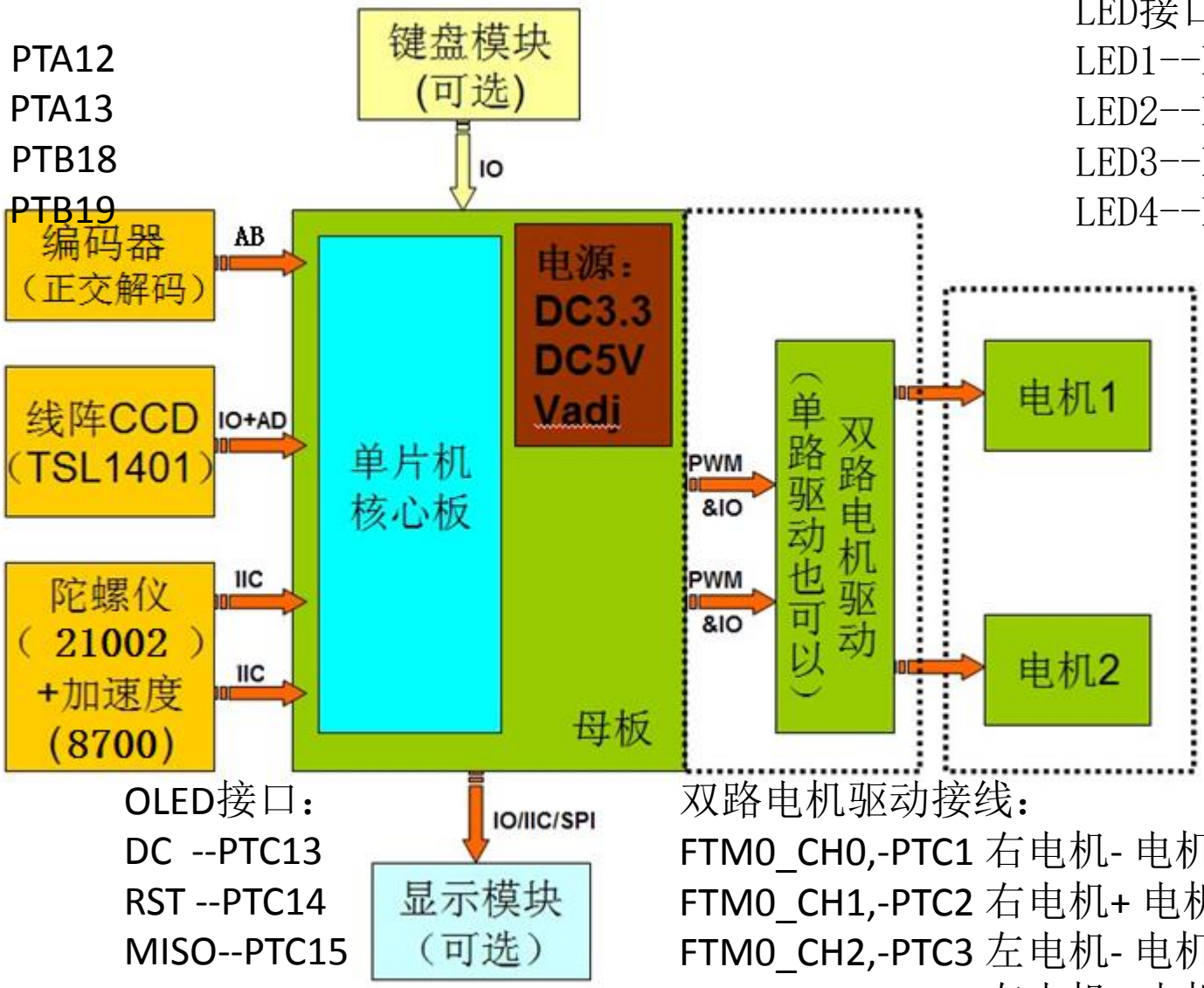
二、K66资源分配

编码器接口：
FTM1_QDPHA PTA12
FTM1_QDPHB PTA13
FTM2_QDPHA PTB18
FTM2_QDPHB PTB19

TSL1401接口：
AD0_SE16
SI PTE3
SCK PTE2

九轴接口：
SCL--PTD8
SDA--PTD9

蓝牙接口：
GND GND
TX PTE24
RX PTE25

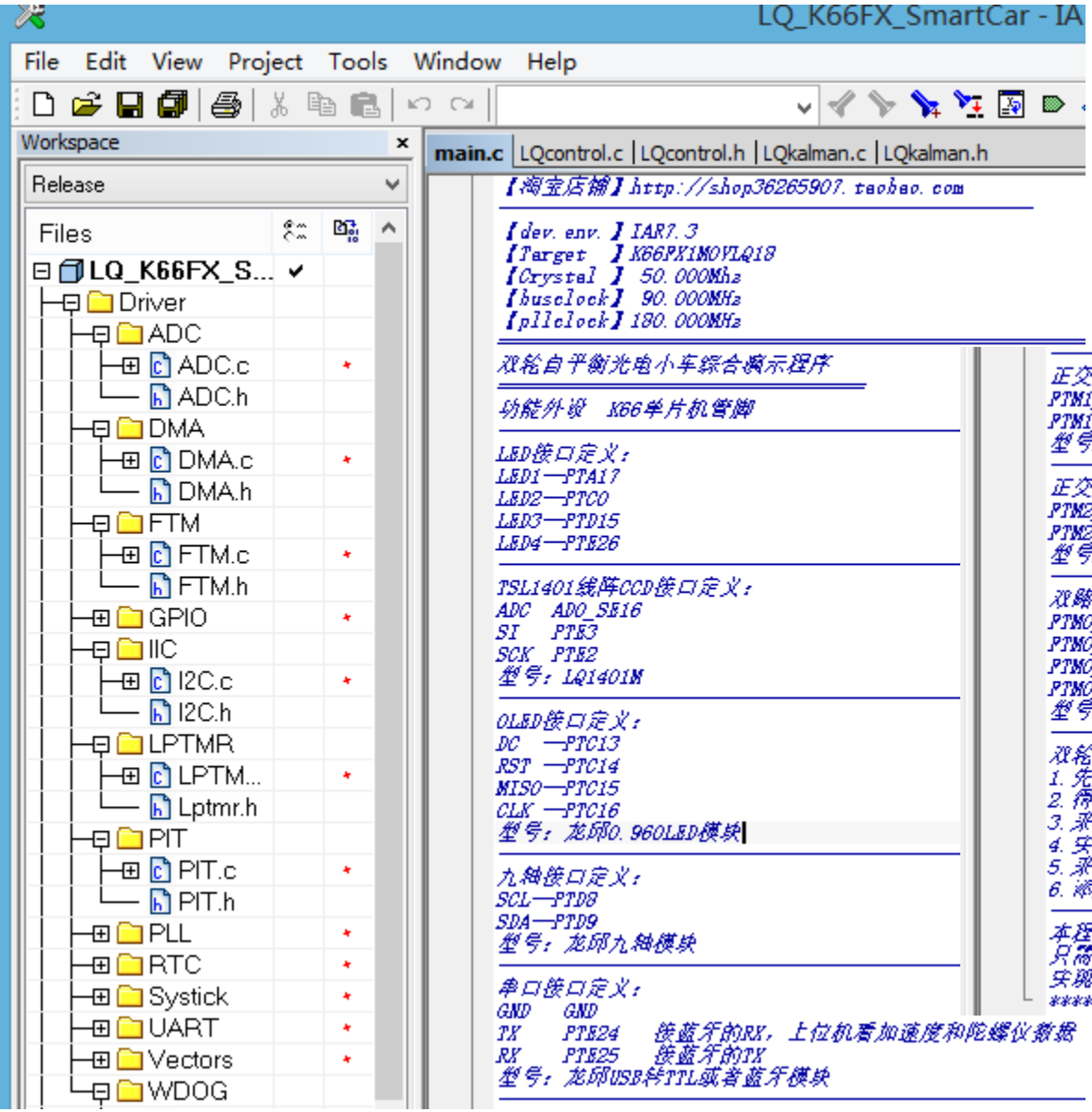


LED接口：
LED1--PTA17
LED2--PTC0
LED3--PTD15
LED4--PTE26

OLED接口：
DC --PTC13
RST --PTC14
MISO--PTC15
CLK --PTC16

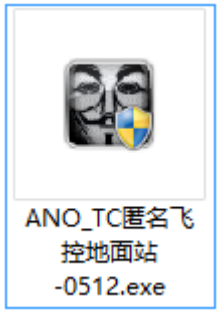
双路电机驱动接线：
FTM0_CH0,-PTC1 右电机- 电机驱动板P1
FTM0_CH1,-PTC2 右电机+ 电机驱动板P2
FTM0_CH2,-PTC3 左电机- 电机驱动板P3
FTM0_CH3,-PTC4 左电机+ 电机驱动板P4

二、K66资源分配


LQ_K66FX_SmartCar - IA
File Edit View Project Tools Window Help
Workspace
Release
Files
LQ_K66FX_S...
Driver
ADC
ADC.c
ADC.h
DMA
DMA.c
DMA.h
FTM
FTM.c
FTM.h
GPIO
IIC
I2C.c
I2C.h
LPTMR
LPTM...
Lptmr.h
PIT
PIT.c
PIT.h
PLL
RTC
Systick
UART
Vectors
WDOG
main.c
LQcontrol.c
LQcontrol.h
LQkalman.c
LQkalman.h
【淘宝店铺】http://shop36265907.taobao.com
【dev.env.】IAR7.3
【Target 】K66FX1M0VLQ18
【Crystal 】 50.000Mhz
【busclock】 90.000MHz
【pllclock】180.000MHz
双轮自平衡光电小车综合测示程序
功能外设 K66单片机管脚
LED接口定义：
LED1—PTA17
LED2—PTC0
LED3—PTD15
LED4—PTE26
TSL1401线阵CCD接口定义：
ADC ADC0_SE16
SI PTE3
SCK PTE2
型号：LQ1401M
OLED接口定义：
DC —PTC13
RST —PTC14
MISO—PTC15
CLK —PTC16
型号：龙邱0.96OLED模块
九轴接口定义：
SCL—PTD8
SDA—PTD9
型号：龙邱九轴模块
串口接口定义：
GND GND
TX PTE24 接蓝牙的RX，上位机看加速度和陀螺仪数据
RX PTE25 接蓝牙的TX
型号：龙邱USB转TTL或者蓝牙模块
正交解码编码器接口：
PTM1_QDPHA PTA12
PTM1_QDPHB PTA13
型号LQ-EMC1418031K 正交解码1K/R
正交解码编码器接口：
PTM2_QDPHA PTB18
PTM2_QDPHB PTB19
型号LQ-EMC1418031K 正交解码1K/R
双路全桥电机驱动接线定义：
PTM0_CH0, -PTC1 右电机- 电机驱动板P1
PTM0_CH1, -PTC2 右电机+ 电机驱动板P2
PTM0_CH2, -PTC3 左电机- 电机驱动板P3
PTM0_CH3, -PTC4 左电机+ 电机驱动板P4
型号：龙邱双路全桥驱动板
双轮平衡车调试流程：
1.先采集加速度和陀螺仪数值，通过ANO_TC匿名飞控地面站-0512.exe上位
2.得到理想角度数据后，结合开环PID控制小车实现站立，（可震荡），
3.采集编码器数值，实现小车原地震荡；
4.实现小车前进后退等速度控制；
5.采集摄像头或者传感器数据，实现小车循迹运行；
6.添加其他的功能，完善优化小车程序。
本程序默认采用蓝底白色赛道，用线阵CCD作为路径识别；
只需要开机时把LQcontrol.h中的QingJiaoZhongZhi标定一下，就可以
实现小车平衡和简单的轨迹识别。

三、各个模块初始化

4.1 上位机


ANO_TC匿名飞
控地面站
-0512.exe


ANO TC
匿名科创
TECHNOLOGY & CREATION
匿名官网
匿名论坛
匿名淘宝
进入地面站

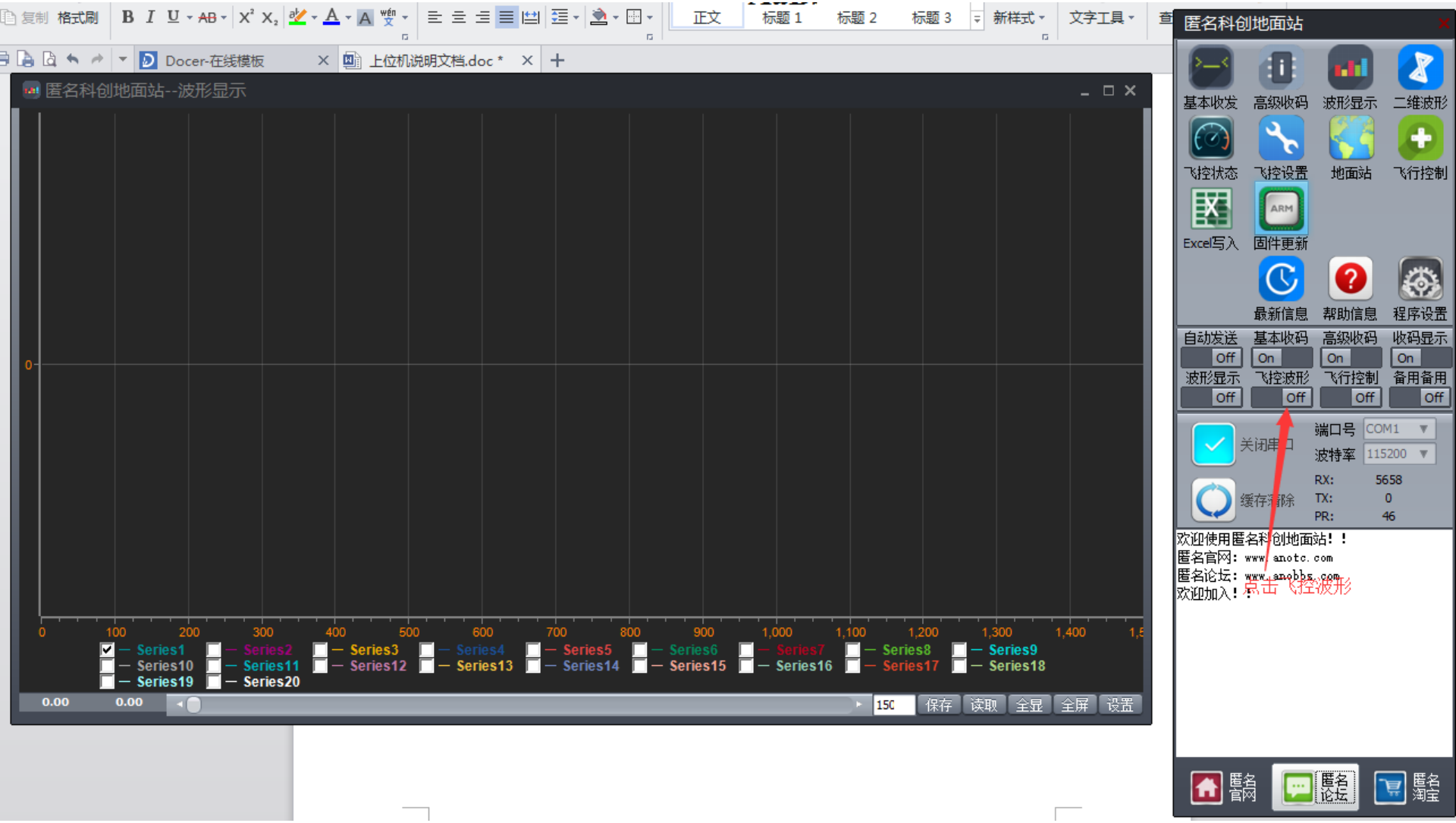
四、两种滤波的调试


匿名科创地面站
基本收发
高级收码
波形显示
二维波形
飞控状态
飞控设置
地面站
飞行控制
Excel写入
固件更新
最新信息
帮助信息
程序设置
自动发送
基本收码
高级收码
收码显示
Off
On
On
On
波形显示
飞控波形
飞行控制
备用备用
Off
Off
Off
Off
打开串口
端口号
COM1
波特率
115200
缓存清除
RX: 0
TX: 0
PR: 0
欢迎使用匿名科创地面站！！
匿名官网：www.anotc.com
匿名论坛：www.anobbs.com
匿名官网
匿名论坛
匿名淘宝

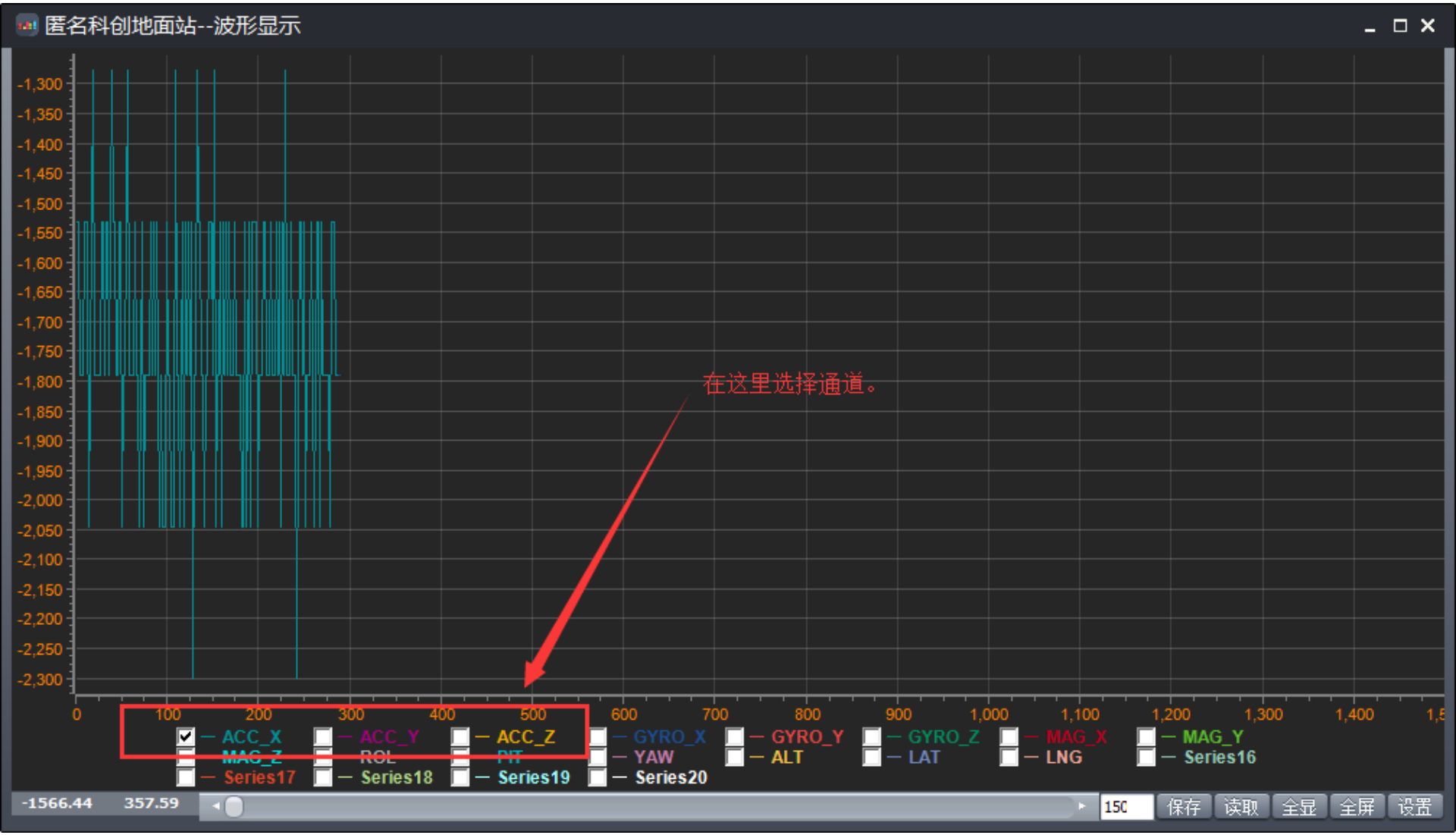

匿名科创地面站
基本收发
高级收码
波形显示
二维波形
飞控状态
飞控设置
地面站
飞行控制
Excel写入
固件更新
最新信息
帮助信息
程序设置
自动发送
基本收码
高级收码
收码显示
Off
On
On
On
波形显示
飞控波形
飞行控制
备用备用
Off
Off
Off
Off
打开串口
端口号
COM1
波特率
115200
缓存清除
RX: 0
TX: 0
PR: 0
欢迎使用匿名科创地面站！！
匿名官网：www.anotc.com
匿名论坛：www.anobbs.com
欢迎加入！！


匿名科创地面站
基本收发
高级收码
波形显示
二维波形
飞控状态
飞控设置
地面站
飞行控制
Excel写入
固件更新
最新信息
帮助信息
程序设置
自动发送
基本收码
高级收码
收码显示
Off
On
On
On
波形显示
飞控波形
飞行控制
备用备用
Off
Off
Off
Off
关闭串口
端口号
COM1
波特率
115200
缓存清除
RX: 6256
TX: 0
PR: 23
欢迎使用匿名科创地面站！！
匿名官网：www.anotc.com
匿名论坛：www.anobbs.com
欢迎加入！！
显示串口收到的数据

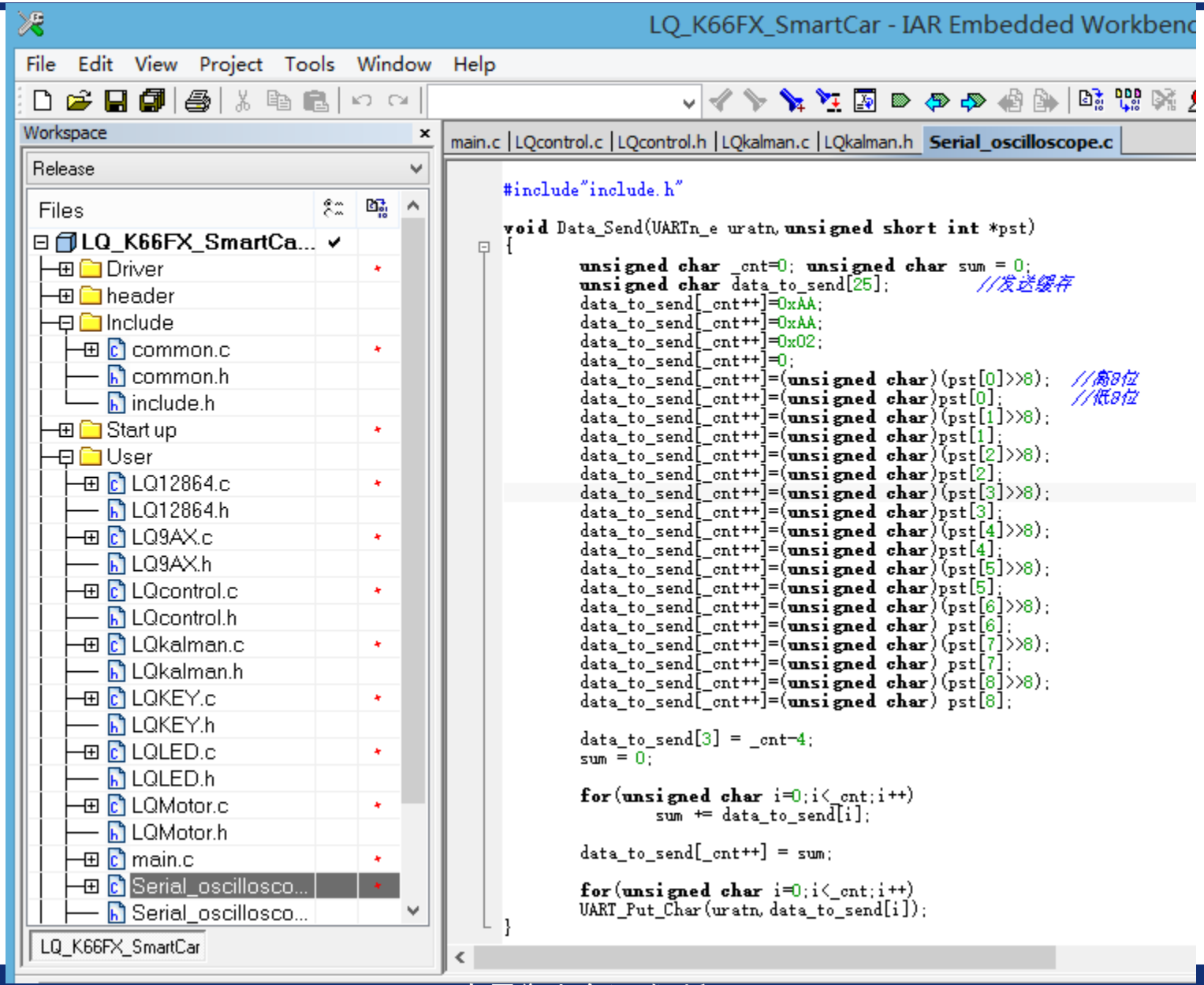
四、两种滤波的调试


匿名科创地面站--波形显示
匿名科创地面站
基本收发
高级收码
波形显示
二维波形
飞控状态
飞控设置
地面站
飞行控制
Excel写入
固件更新
最新信息
帮助信息
程序设置
自动发送
基本收码
高级收码
收码显示
波形显示
飞控波形
飞行控制
备用备用
关闭串口
端口号
COM1
波特率
115200
RX:
5658
TX:
0
PR:
46
缓存清除
欢迎使用匿名科创地面站！！
匿名官网：www.anotc.com
匿名论坛：www.anobbs.com
欢迎加入！！
点击飞控波形
匿名官网
匿名论坛
匿名淘宝
Series1
Series2
Series3
Series4
Series5
Series6
Series7
Series8
Series9
Series10
Series11
Series12
Series13
Series14
Series15
Series16
Series17
Series18
Series19
Series20
保存
读取
全显
全屏
设置

四、两种滤波的调试


匿名科创地面站--波形显示
-1,300
-1,350
-1,400
-1,450
-1,500
-1,550
-1,600
-1,650
-1,700
-1,750
-1,800
-1,850
-1,900
-1,950
-2,000
-2,050
-2,100
-2,150
-2,200
-2,250
-2,300
在这里选择通道。
0
100
200
300
400
500
600
700
800
900
1,000
1,100
1,200
1,300
1,400
ACC_X
ACC_Y
ACC_Z
GYRO_X
GYRO_Y
GYRO_Z
MAG_X
MAG_Y
MAG_Z
ROL
PIT
YAW
ALT
LAT
LNG
Series16
Series17
Series18
Series19
Series20
-1566.44
357.59
150
保存
读取
全显
全屏
设置

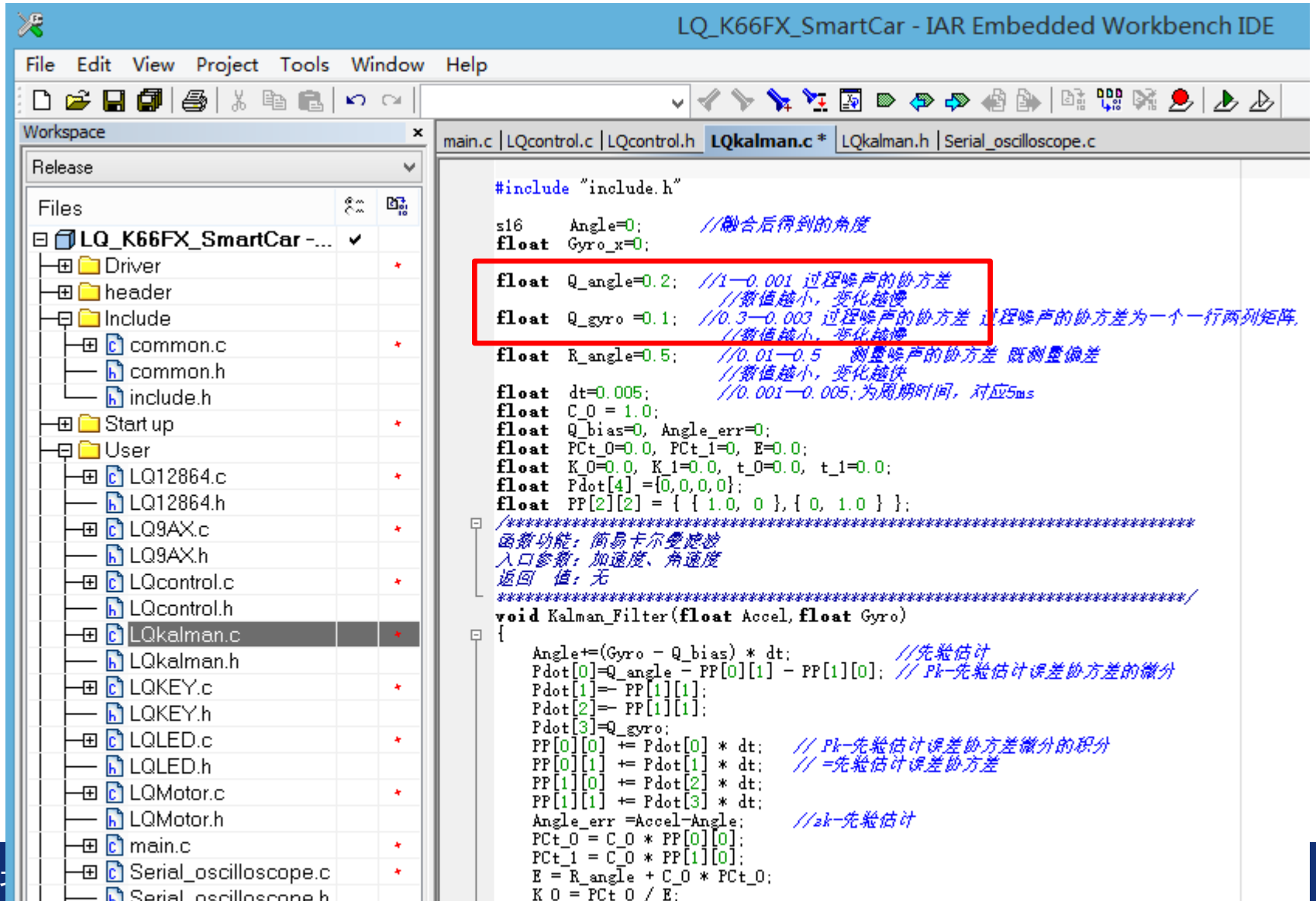
四、两种滤波的调试


LQ_K66FX_SmartCar - IAR Embedded Workbench
File Edit View Project Tools Window Help
Workspace
Release
Files
LQ_K66FX_SmartCa...
Driver
header
Include
common.c
common.h
include.h
Start up
User
LQ12864.c
LQ12864.h
LQ9AX.c
LQ9AX.h
LQcontrol.c
LQcontrol.h
LQkalman.c
LQkalman.h
LQKEY.c
LQKEY.h
LQLED.c
LQLED.h
LQMotor.c
LQMotor.h
main.c
Serial_oscillosco...
Serial_oscillosco...
LQ_K66FX_SmartCar
main.c | LQcontrol.c | LQcontrol.h | LQkalman.c | LQkalman.h | Serial_oscilloscope.c
#include"include.h"
void Data_Send(UARTn_e uratn,unsigned short int *pst)
{
unsigned char _cnt=0; unsigned char sum = 0;
unsigned char data_to_send[25]; //发送缓存
data_to_send[_cnt++]=0xAA;
data_to_send[_cnt++]=0xAA;
data_to_send[_cnt++]=0x02;
data_to_send[_cnt++]=0;
data_to_send[_cnt++]=(unsigned char)(pst[0]>>8); //高8位
data_to_send[_cnt++]=(unsigned char)pst[0]; //低8位
data_to_send[_cnt++]=(unsigned char)(pst[1]>>8);
data_to_send[_cnt++]=(unsigned char)pst[1];
data_to_send[_cnt++]=(unsigned char)(pst[2]>>8);
data_to_send[_cnt++]=(unsigned char)pst[2];
data_to_send[_cnt++]=(unsigned char)(pst[3]>>8);
data_to_send[_cnt++]=(unsigned char)pst[3];
data_to_send[_cnt++]=(unsigned char)(pst[4]>>8);
data_to_send[_cnt++]=(unsigned char)pst[4];
data_to_send[_cnt++]=(unsigned char)(pst[5]>>8);
data_to_send[_cnt++]=(unsigned char)pst[5];
data_to_send[_cnt++]=(unsigned char)(pst[6]>>8);
data_to_send[_cnt++]=(unsigned char) pst[6];
data_to_send[_cnt++]=(unsigned char)(pst[7]>>8);
data_to_send[_cnt++]=(unsigned char) pst[7];
data_to_send[_cnt++]=(unsigned char)(pst[8]>>8);
data_to_send[_cnt++]=(unsigned char) pst[8];
data_to_send[3] = _cnt-4;
sum = 0;
for(unsigned char i=0;i<_cnt;i++)
sum += data_to_send[i];
data_to_send[_cnt++] = sum;
for(unsigned char i=0;i<_cnt;i++)
UART_Put_Char(uratn,data_to_send[i]);
}

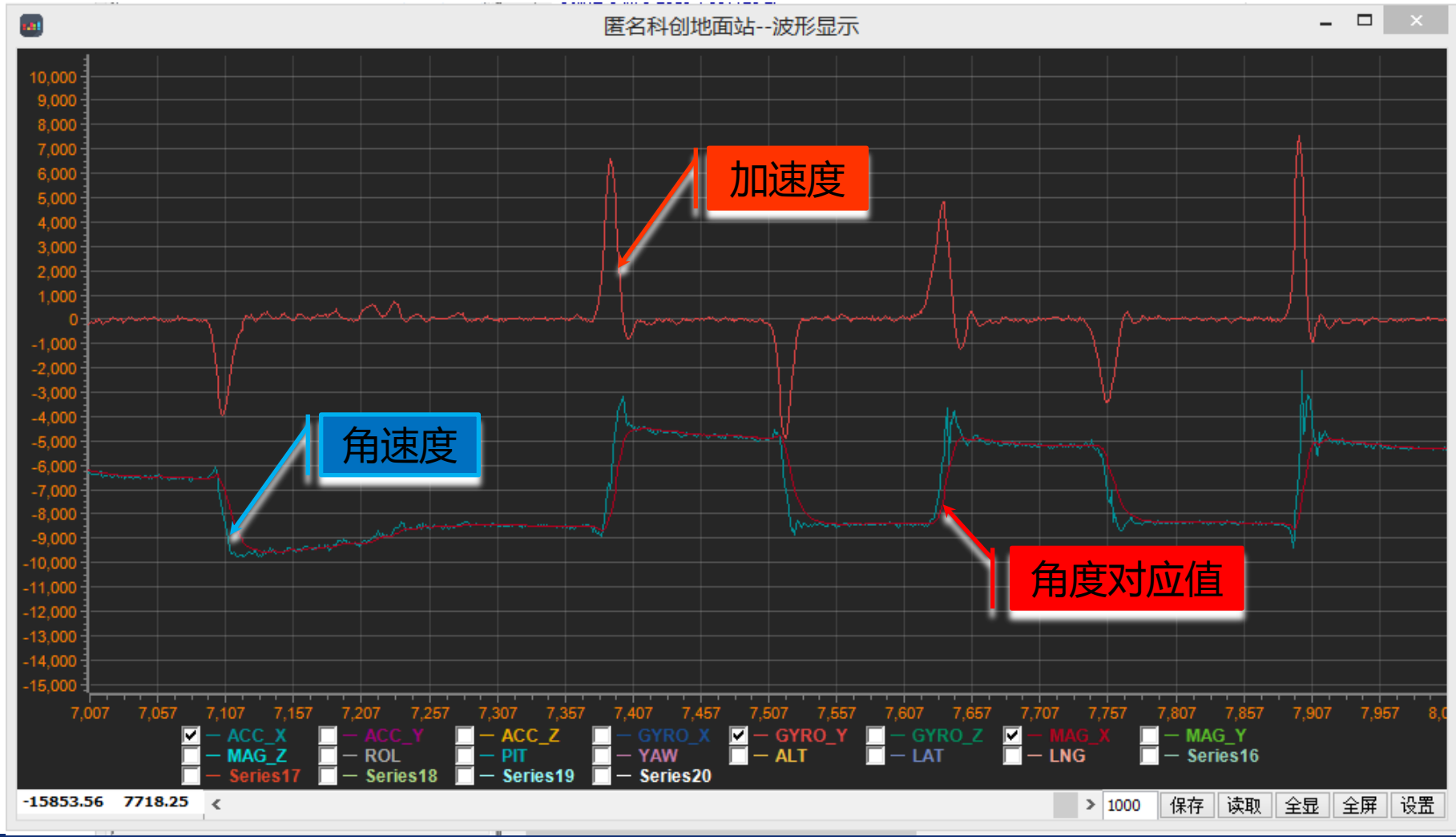
四、两种滤波的调试

龙邱智能科技 LONGQIUZHINENGKEJI

4.2 卡尔曼滤波标定

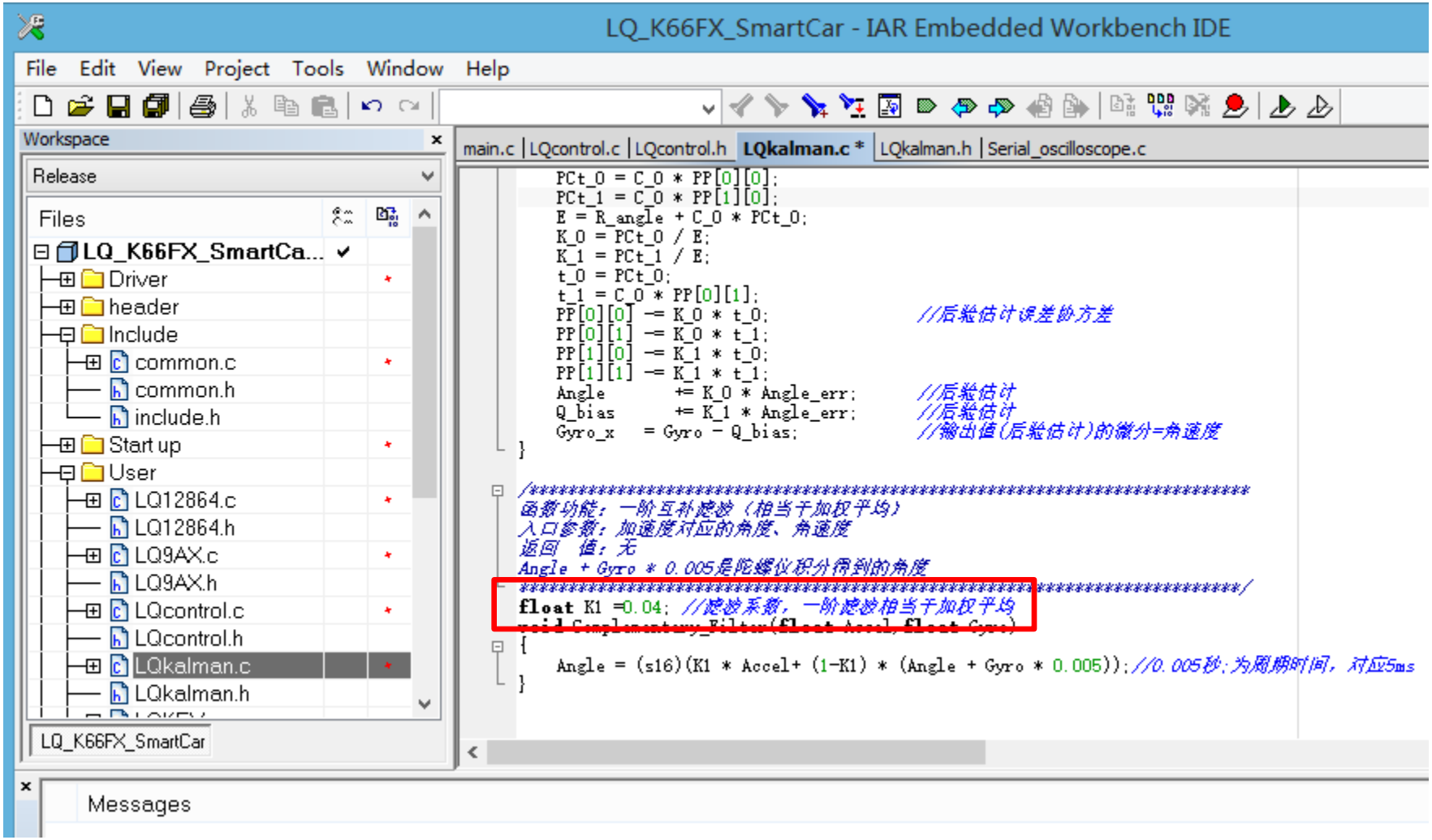


4.2 卡尔曼滤波标定


匿名科创地面站--波形显示
加速度
角速度
角度对应值
ACC_X
ACC_Y
ACC_Z
GYRO_X
GYRO_Y
GYRO_Z
MAG_X
MAG_Y
MAG_Z
ROL
PIT
YAW
ALT
LAT
LNG
Series16
Series17
Series18
Series19
Series20
-15853.56
7718.25
1000
保存
读取
全显
全屏
设置

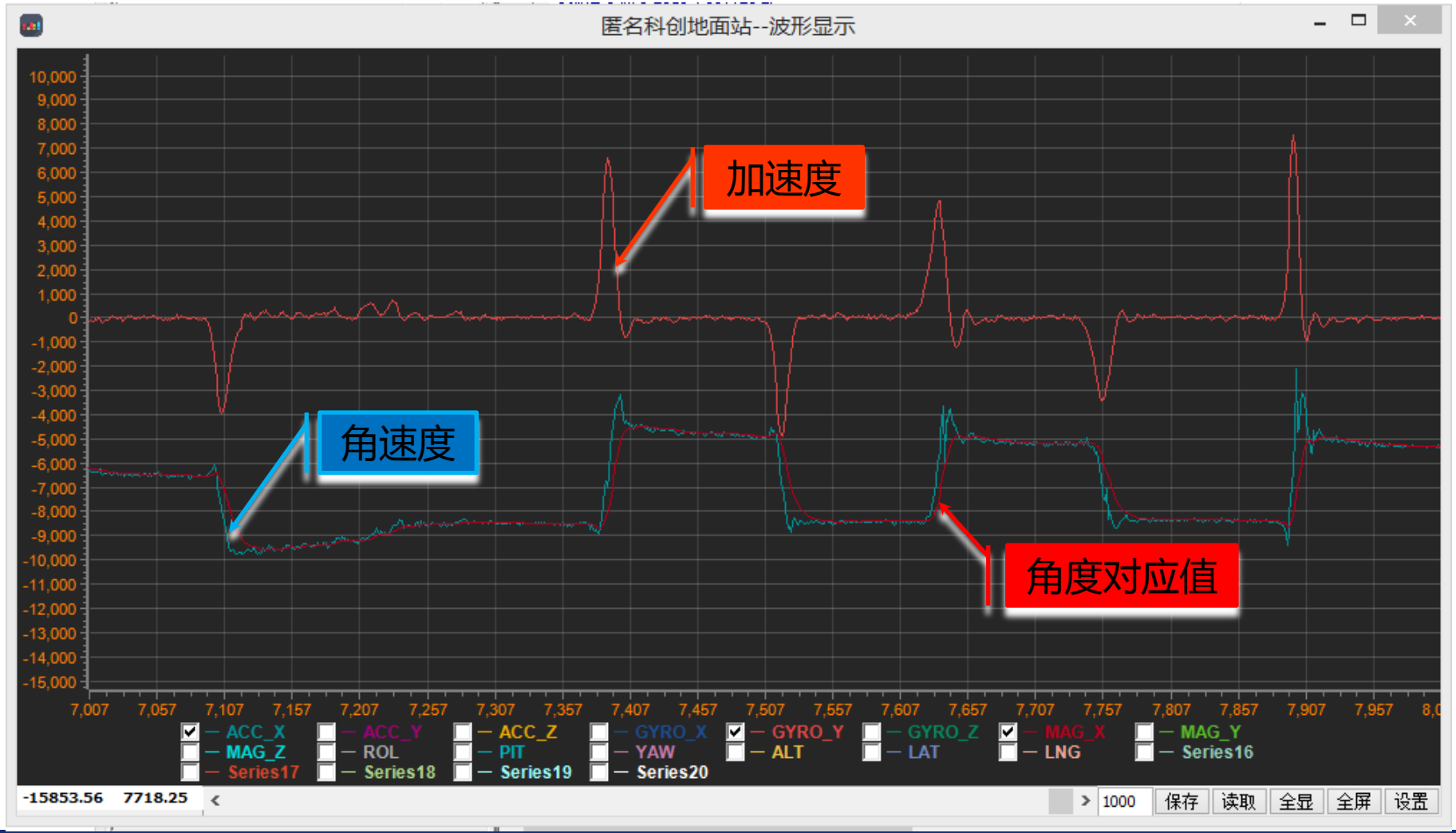
龙邱智能科技 LONGQIUZHINENGKEJI

4.2 互补滤波标定

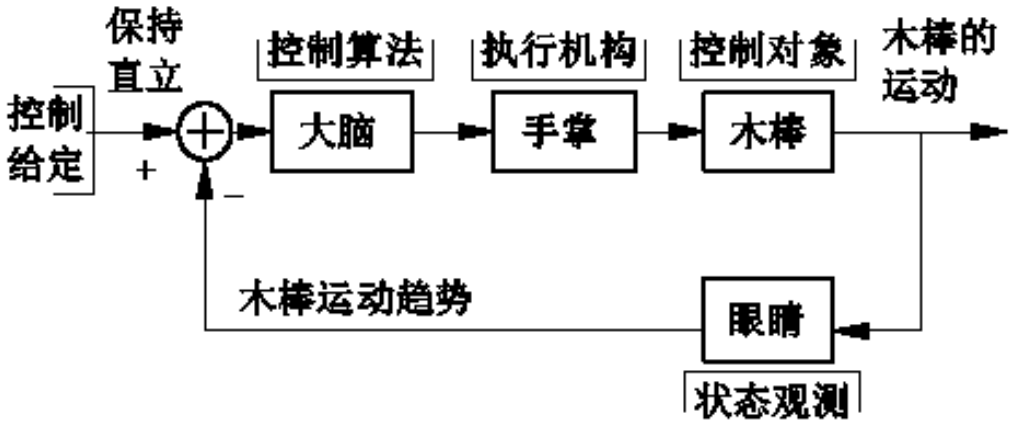


龙邱智能科技 LONGQIUZHINENGKEJI

4.2 互补滤波标定



5.1 自平衡原理



倒立摆的实现

五、自平衡的实现

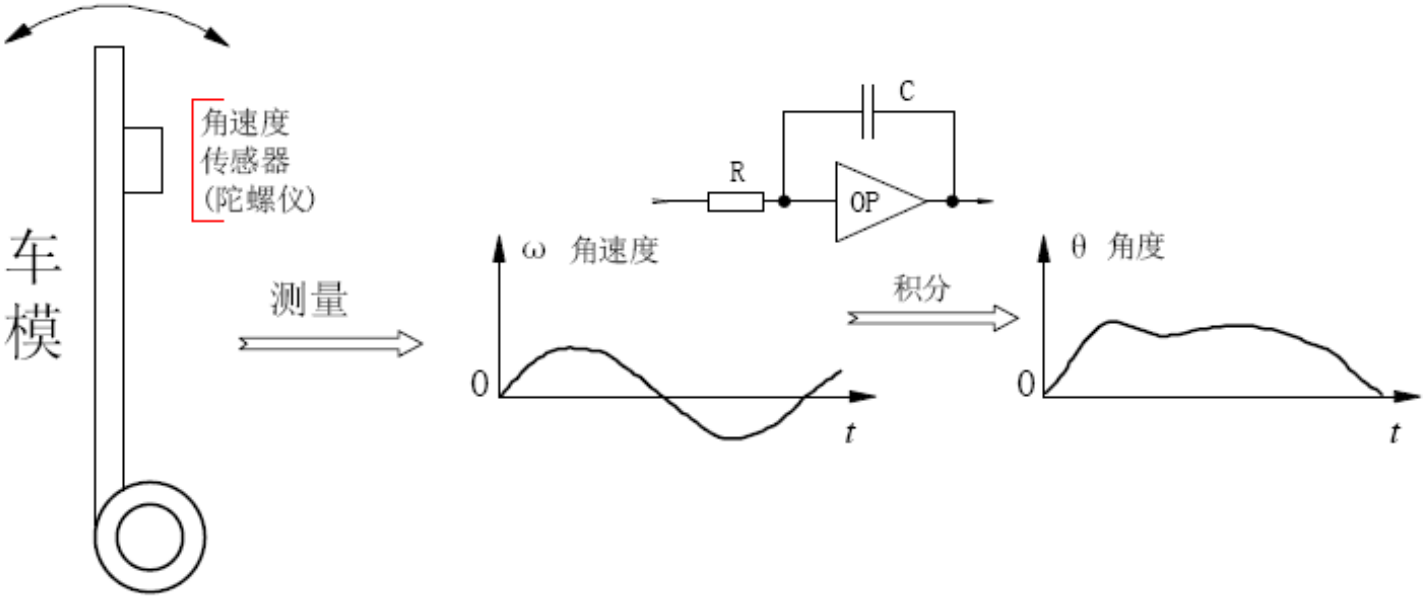
5.1 自平衡原理

★ 车体向左倾斜，车轮向左加速运行。

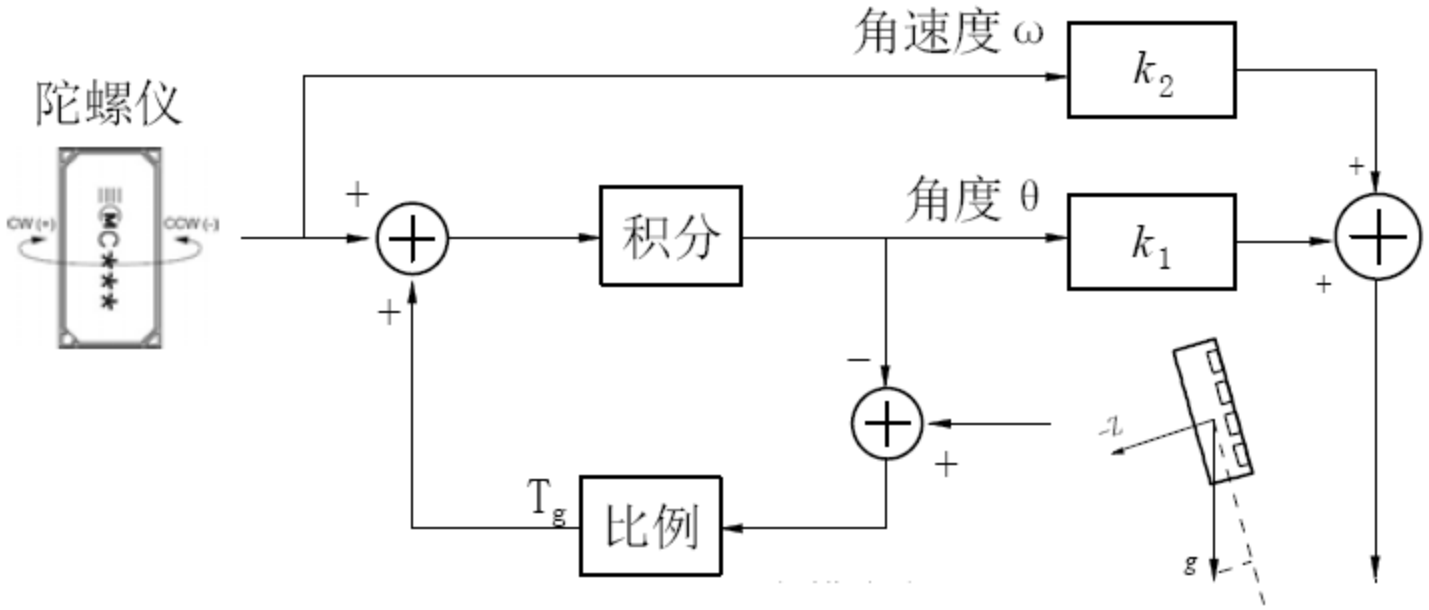
■ 车体垂直车轮保持静止。

○ 车体向右倾斜，车轮向右加速运行。

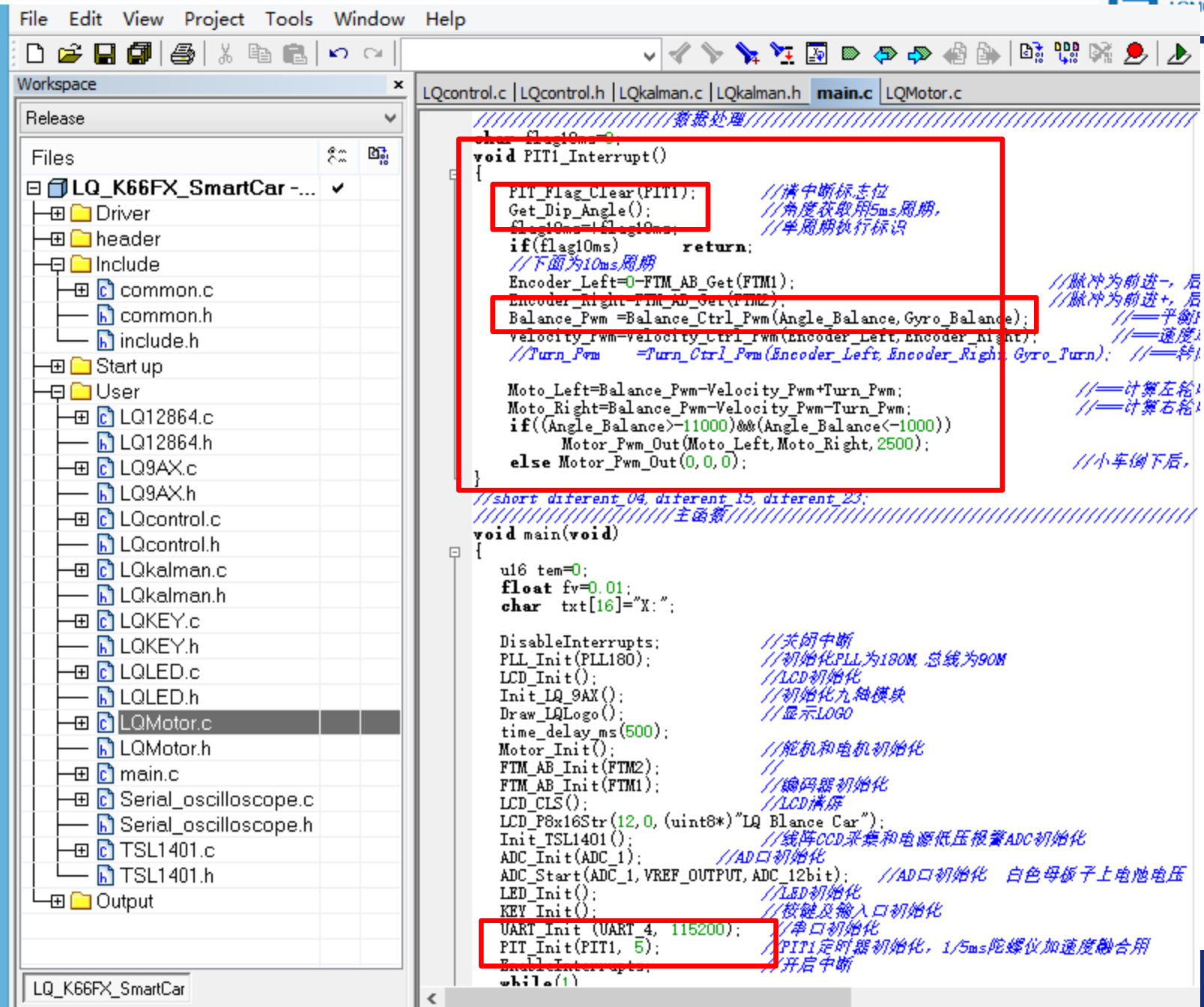
5.1 自平衡原理



5.1 自平衡原理



五、自平衡的实现


File Edit View Project Tools Window Help
Workspace
Release
Files
LQ_K66FX_SmartCar -...
Driver
header
Include
common.c
common.h
include.h
Start up
User
LQ12864.c
LQ12864.h
LQ9AX.c
LQ9AX.h
LQcontrol.c
LQcontrol.h
LQkalman.c
LQkalman.h
LQKEY.c
LQKEY.h
LQLED.c
LQLED.h
LQMotor.c
LQMotor.h
main.c
Serial_oscilloscope.c
Serial_oscilloscope.h
TSL1401.c
TSL1401.h
Output
LQ_K66FX_SmartCar
LQcontrol.c | LQcontrol.h | LQkalman.c | LQkalman.h | main.c | LQMotor.c
//////////////////////数据处理//////////////////////
void PIT1_Interrupt()
{
PIT_Flag_Clear(PIT1); //清中断标志位
Get_Dip_Angle(); //角度获取用5ms周期,
//单周期执行标识
if(flag10ms) return;
//下面为10ms周期
Encoder_Left=0-FTM_AB_Get(FTM1); //脉冲为前进-,后
//脉冲为前进+,后
Balance_Pwm =Balance_Ctrl_Pwm(Angle_Balance,Gyro_Balance); //==平衡
//==速度
//Turn_Pwm =Turn_Ctrl_Pwm(Encoder_Left,Encoder_Right,Gyro_Turn); //==转
Moto_Left=Balance_Pwm-Velocity_Pwm+Turn_Pwm; //==计算左轮
Moto_Right=Balance_Pwm-Velocity_Pwm-Turn_Pwm; //==计算右轮
if((Angle_Balance)>-11000)&&(Angle_Balance<-1000))
Motor_Pwm_Out(Moto_Left,Moto_Right,2500);
else Motor_Pwm_Out(0,0,0); //小车倒下后,
}
//short diferent_04, diferent_15, diferent_23;
//////////////////////主函数//////////////////////
void main(void)
{
u16 tem=0;
float fv=0.01;
char txt[16]="X:";
DisableInterrupts; //关闭中断
PLL_Init(PLL180); //初始化PLL为180M,总线为90M
LCD_Init(); //LCD初始化
Init_LQ_9AX(); //初始化九轴模块
Draw_LQLogo(); //显示LOGO
time_delay_ms(500);
Motor_Init(); //舵机和电机初始化
FTM_AB_Init(FTM2); //
FTM_AB_Init(FTM1); //编码器初始化
LCD_CLS(); //LCD清屏
LCD_P8x16Str(12,0,(uint8*)"LQ Blance Car");
Init_TSL1401(); //线阵CCD采集和电源低压报警ADC初始化
ADC_Init(ADC_1); //AD口初始化
ADC_Start(ADC_1,VREF_OUTPUT,ADC_12bit); //AD口初始化 白色母板子上电池电压
LED_Init(); //LED初始化
KEY_Init(); //按键及输入口初始化
UART_Init (UART_4, 115200); //串口初始化
PIT_Init(PIT1, 5); //PIT1定时器初始化,1/5ms陀螺仪加速度融合用
//开启中断
while(1)

5.2 平衡中心值标定

```
/**************************************************************************
函数功能：小车平衡电机占空比控制，直立PD控制
入口参数：倾角角度、角速度
返回  值：平衡控制用PWM
**************************************************************************/
int Balance_Ctrl_Pwm(float Ang,float Gyro)
{                                   //kp=1.5,独立作用跟倒的方向同向；kd=0.25，独立作用，跟倾斜方向要同向，跟随效果
    float Bias,kp=0.8,kd=0.7;       //调试方法，将其中一个参数值为零，调整另一个，直到能控制车轮转动为止
    int balance;

    Bias=Ang-QingJiaoZhongZhi;      //中值，求出平衡的角度中值 和机械相关
    balance=(int)(kp*Bias+Gyro*kd); //计算平衡控制的电机PWM  PD控制   kp是P系数 kd是D系数
    return balance;
}
```

调试方法：

不开电机，手动将小车维持平衡状态，读取平衡状态的角度值—倾角中心值。

5.3 平衡中的PD标定

```
/**************************************************************************
函数功能：小车平衡电机占空比控制，直立PD控制
入口参数：倾角角度、角速度
返回  值：平衡控制用PWM
**************************************************************************/
int Balance_Ctrl_Pwm(float Ang,float Gyro)
{                                   //kp=1.5,独立作用跟倒的方向同向；kd=0.25，独立作用，跟倾斜方向要同向，跟随效果
    float Bias,kp=0.8,kd=0.7;       //调试方法，将其中一个参数值为零，调整另一个，直到能控制车轮转动为止
    int balance;

    Bias=Ang-QingJiaoZhongZhi;      //中值，求出平衡的角度中值 和机械相关
    balance=(int)(kp*Bias+Gyro*kd); //计算平衡控制的电机PWM  PD控制   kp是P系数 kd是D系数
    return balance;
}
```

调试方法，将其中一个参数值为零，调整另一个，直到能控制车轮转动，最后融合两者.

5.3 平衡中的PD标定

极性标定：

由于平衡过程是“追”的过程，参数方向跟倒向方向一致为正确。否则取反。

大小标定：

Kp，偏大，很“冲”；偏小，无力！

Kd，偏大，抖动；偏小，跟随效果很差

Kp值的大小跟“倾角”大小和倾斜角度有关，如当小车从平衡位置到倾斜15°变化范围是2000，此时速度就要达到最大值，则：

Kp最大值=5000（占空比100%最大值）/2000=2.5，由于电机驱动能力太强，可以适当减小此值。

当Kp值标定好之后，慢慢加大Kd，一直到电机刚要开始抖动为准。

6.1 速度PI标定

LQcontrol.c * | LQcontrol.h | LQkalman.c | LQkalman.h | main.c | LQMotor.c

```
/**************************************************************************
函数功能：速度PI控制 修改前进后退速度，
入口参数：左轮编码器、右轮编码器
返回  值：速度控制PWM
**************************************************************************/
int Velocity_Ctrl_Pwm(int encoder_left,int encoder_right)
{
    static float Velocity,Encoder_Least,Encoder,Movement;
    static float Encoder_Integral,Target_Velocity;
    float kp=-4,ki=-0.02;

    Target_Velocity=60;                          //目标速度，脉冲数量

    Movement=Target_Velocity ;                   //前进标志位置1

    //车速设置，可以通过拨码开关或者按键等设置速度等级
    if(Encoder_Least< 20)                                    //后退
        QingJiaoZhongZhi=QingJiaoZhongZhiTarget+400;
    else
        QingJiaoZhongZhi=QingJiaoZhongZhiTarget+100;

    //速度PI控制
    Encoder_Least =(Encoder_Left+Encoder_Right)-0;           //获取最新速
    Encoder *= 0.8;                                          //一阶低通滤
    Encoder += Encoder_Least*0.2;                            //一阶低通滤
    Encoder_Integral +=Encoder;                              //积分出位移
    Encoder_Integral=Encoder_Integral-Movement;              //接收遥控器
    if(Encoder_Integral>10000)   Encoder_Integral=10000;     //积分限幅
    if(Encoder_Integral<-10000)  Encoder_Integral=-10000;    //积分限幅
    Velocity=Encoder*kp+Encoder_Integral*ki;                 //速度控制
    //Encoder_Integral=0;      //电机关闭后清除积分
    return (int)Velocity;
}
```

调试方法，将其中一个参数值某数，然后Kp ≈200*ki ，标定极性后结合直立环实现整车控制。

Kp最大值=5000（占空比最大）/(100（编码器10ms脉冲数）*50%)

6.1 速度PI标定

极性标定：

Kp单独作用时，转动轮子，如果两轮方向转动相反，说明极性错误；如果两轮方向转动相同（会加速到电机最大转速），说明极性正确；

大小标定：

Kp，偏大，负反馈可能抵消直立环，电机无力甚至不转；偏小，效果差！

Kp最大值=5000（占空比100%最大值）/(100（编码器10ms期望脉冲数）*50%)

Kp ≈200*ki，需结合直立环调整该参数。

龙邱智能科技 LONGQIUZHINENGKEJI

7.1 转向P标定

```
/**************************************************************************
函数功能：转向控制  修改转向速度，请修改Turn_MaxPwm即可
入口参数：左轮编码器、右轮编码器、Z轴陀螺仪
返回  值：转向控制PWM
作    者：
**************************************************************************/
int Turn_Ctrl_Pwm1(int encoder_left,int encoder_right,float gyro)//转向控制
{
    float Kp=4.5;

    //=转向PD控制器=//
    return (int)(RoadPianCha*Kp);//最简单的P控制算法
}
int Turn_Ctrl_Pwm(int encoder_left,int encoder_right,float gyro)//转向控制
{
    float Kp=4.5;
    static float Turn_Target;

    Turn_Target=RoadPianCha+(encoder_left-encoder_right)/10;
    if(Turn_Target>200)  Turn_Target=200;            //转向速度限幅
    if(Turn_Target<-200) Turn_Target=-200;
    //=转向PD控制器=//
    return (int)(Turn_Target*Kp);//最简单的P控制算法
}
int Turn_Ctrl_Pwm2(int encoder_left,int encoder_right,float gyro)//转向控制
{
    static float Turn_Target,Turn;
    float Turn_MaxPwm=88,Kp=5,Kd=0;

    Turn_Target=RoadPianCha-My_Abs(encoder_left-encoder_right)/10;

    if(Turn_Target>Turn_MaxPwm)  Turn_Target=Turn_MaxPwm;          //转向速度限幅
    if(Turn_Target<-Turn_MaxPwm) Turn_Target=-Turn_MaxPwm;

    if(Flag_Remote==0)  Kd=0.5;
    else Kd=0;                                                     //转向的时候陀螺仪反馈

    //=转向PD控制器=//
    Turn=-Turn_Target*Kp -gyro*Kd;                 //结合Z轴陀螺仪进行PD控制
    return (int)Turn;
}
```

调试方法，通过电感，线阵摄像头或者面阵摄像头采集并计算赛道偏差，结合直立环和速度环，实现最终控制。

7.1 转向P标定

极性标定：

Kp转向的极性比较容易标定，根据小车偏离程度来设定即可，如果小车右边，就要实现左转弯；反之，右转弯。

大小标定：

Kp大小范围是根据计算出的偏差值来确定的，比如偏差范围为（-1000，1000），那么：

Kp最大值=5000（占空比100%最大值）/1000=5

然后结合直立环和速度环适当调整该参数达到最佳控制。

Q&A！