

一、下载相关编程软件

下载安装软件

目前最新的IAR for ARM为v6.30，支持更多的Kinetics系列芯片，因此推荐大家更新，避免因版本太低而出现不兼容，甚至出现异常错误的情况。

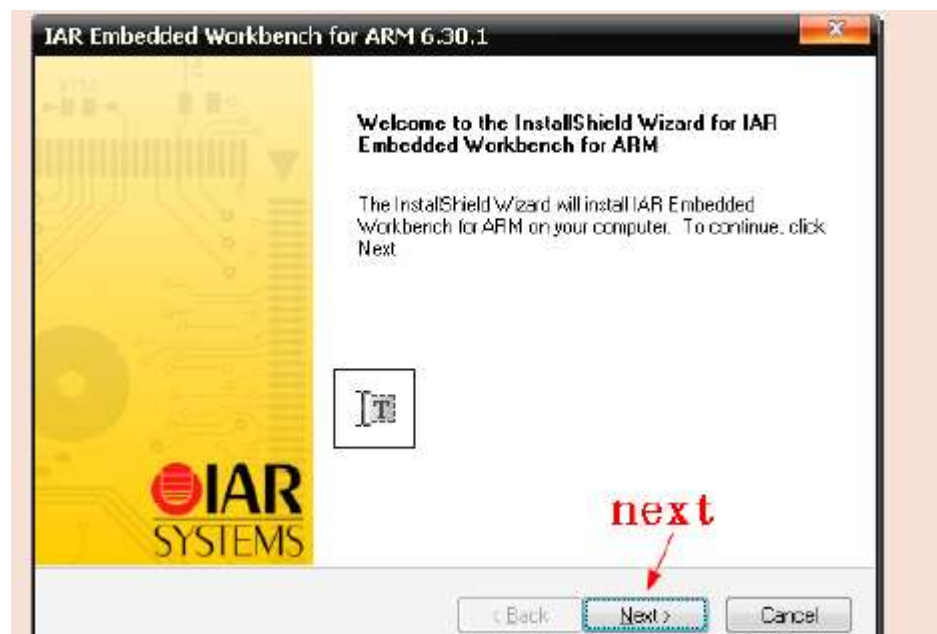
安装 IAR 详细过程

1. 下载后解压文件，打开目录，运行安装文件：

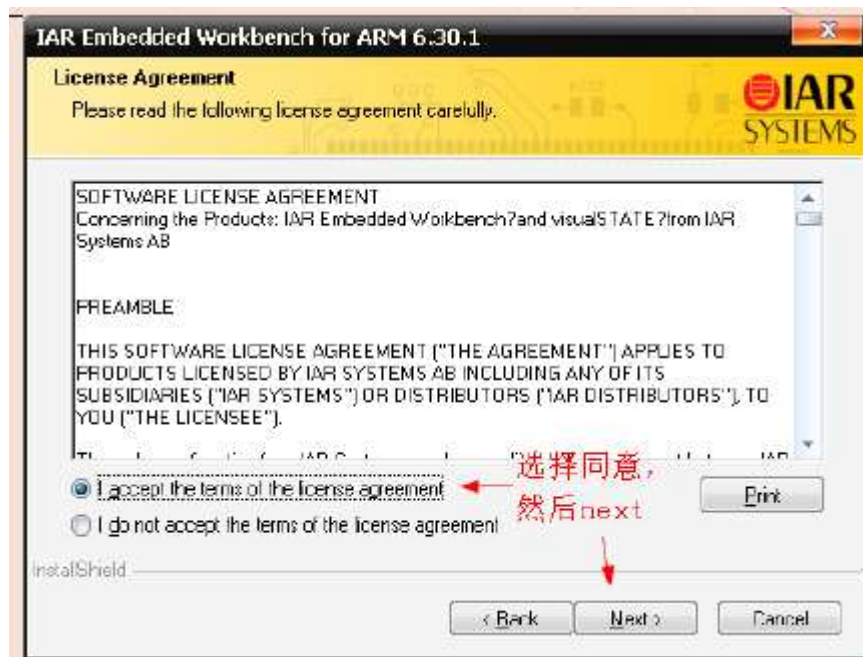


i.

2. 选择安装 IAR Embedded Workbench



选择同意，然后点击next



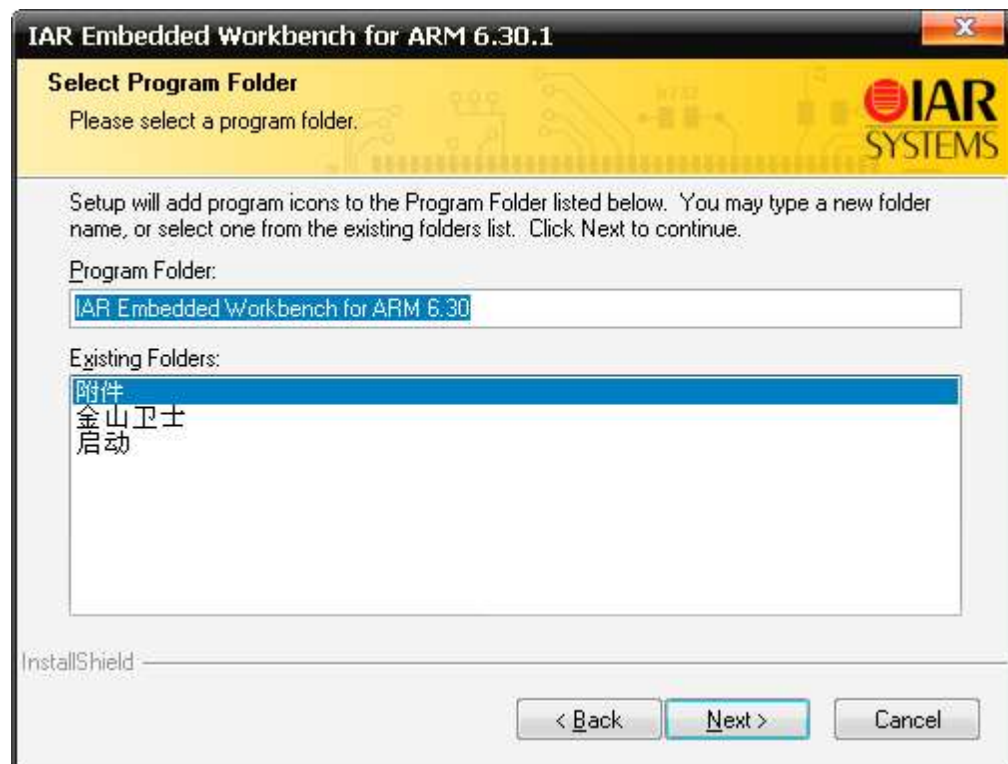
需要填入key

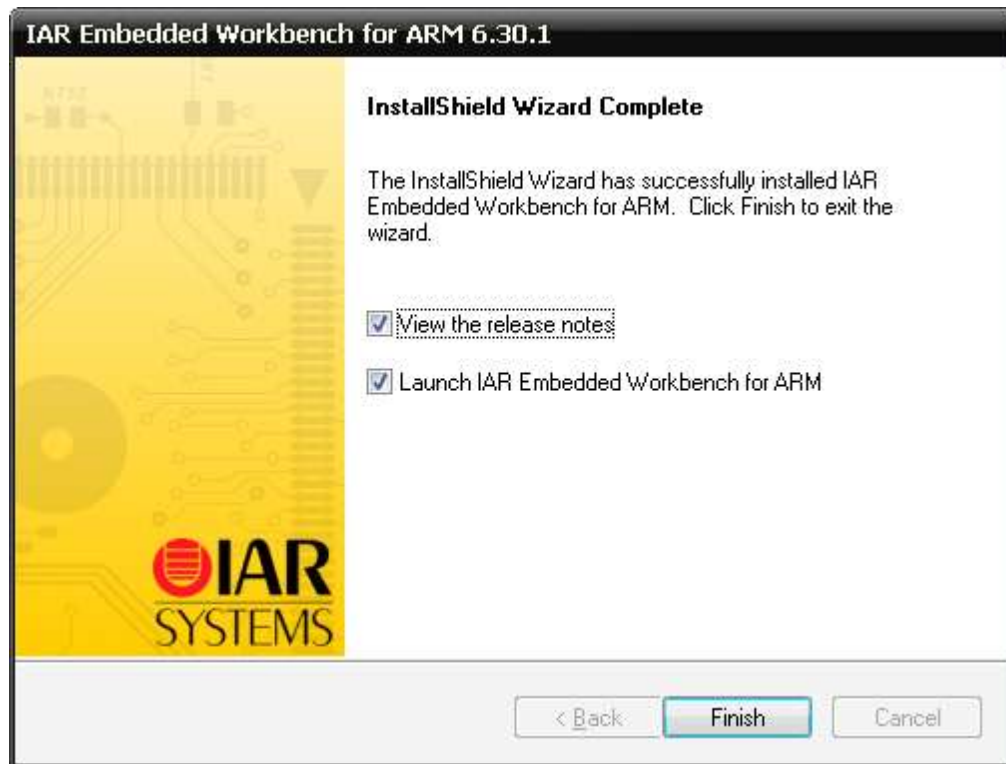


打开软件目录，找到注册机文件，并运行它。

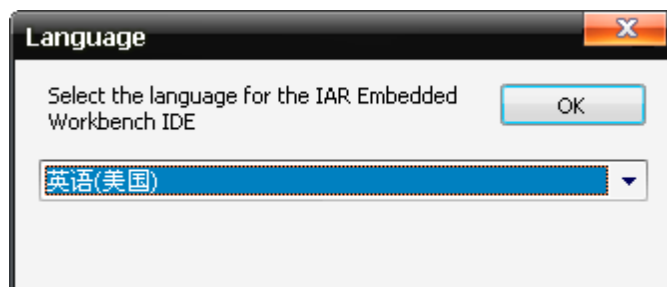
按下任意键，会自动打开license.txt文件，并提示：搜索 "EWARM" ，找到所在行的key。

设置「开始」菜单文件夹，还是继续Next,,

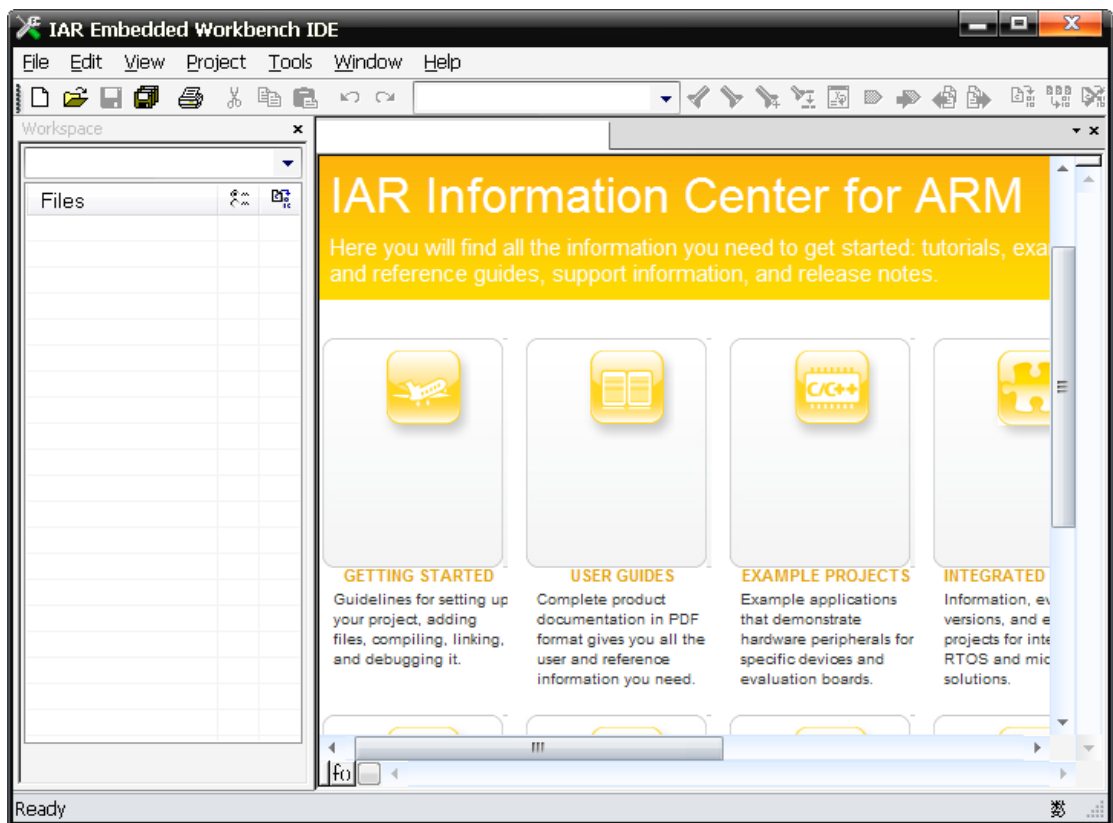




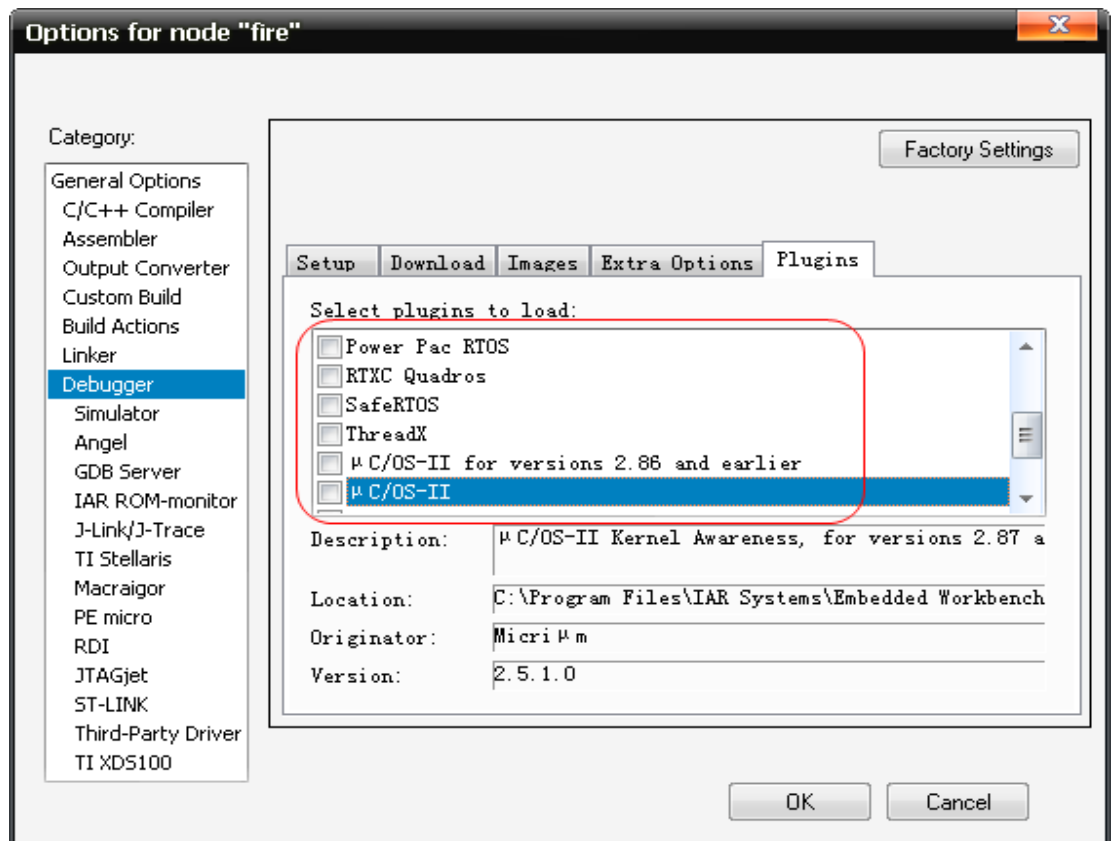
运行IAR，语言选择，按OK结束



IAR的编程界面：

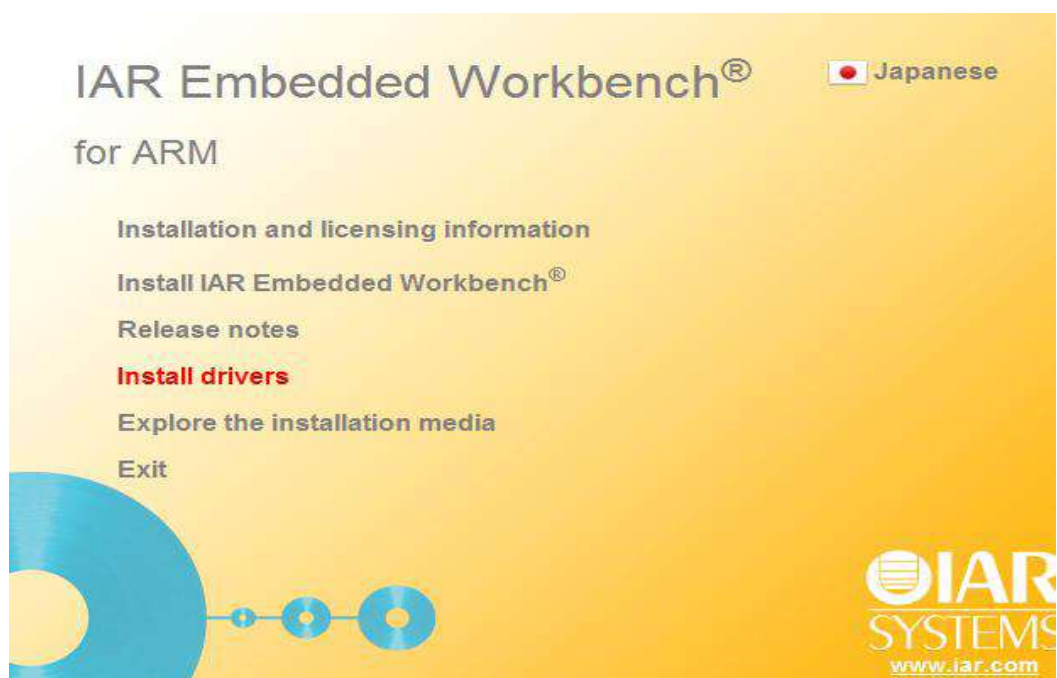


相对于Keil For ARM、CodeWarrior 而言，IAR for ARM的编程界面是最简单的，编译效率高，在嵌入式系统的调试方面提供了可供调试的插件



1. 安装仿真器驱动

1. 又换会原来的安装导航界面：选择Install drivers



选择了用jlink作为仿真调试器，因此这里选择jlink。

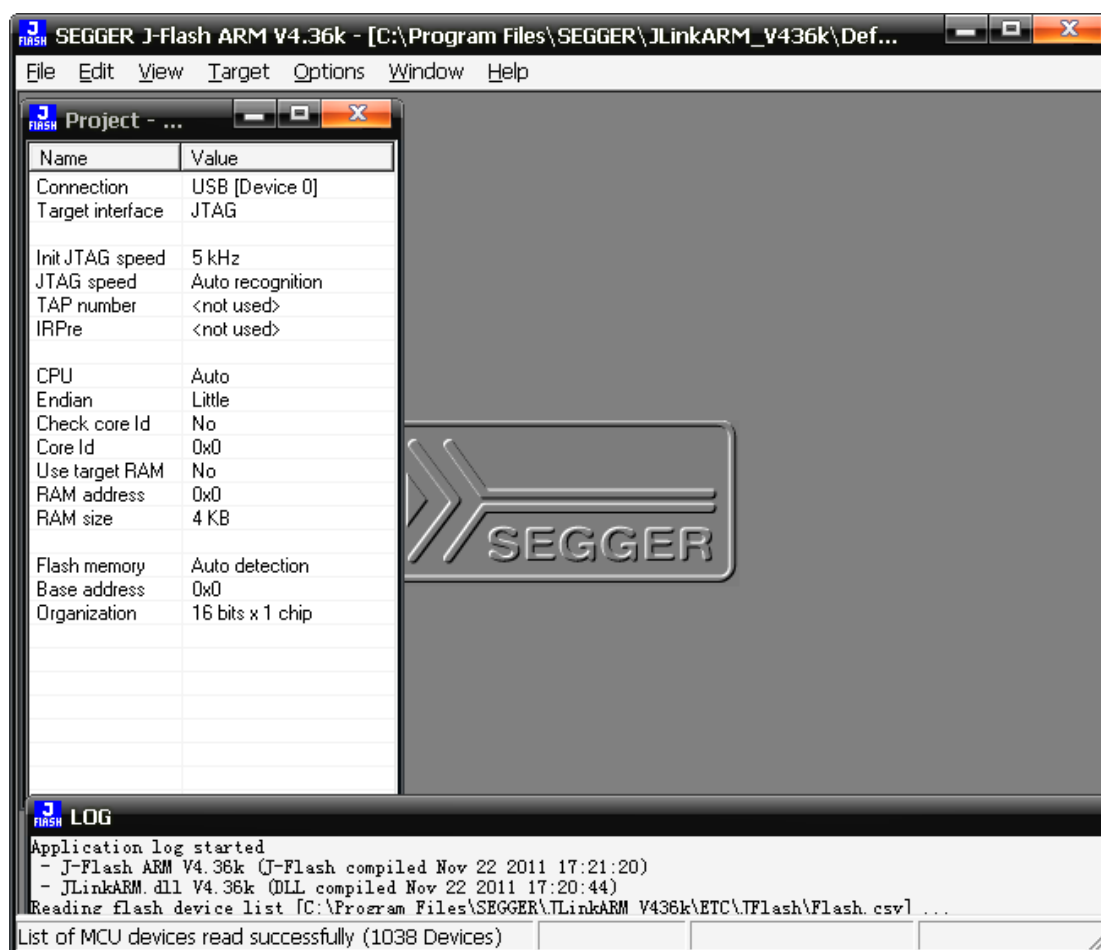


运行安装文件



运行后，会自动安装驱动，不会有其他提示。

另外，为了可以使用擦除芯片等功能，我们可以选择用segger公司的 j-link:



安装时，一路狂Next就行...

安装后，在开始菜单里找到 J-Flash ARM，运行它即可



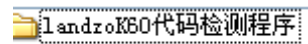
二、如果利用相关软件打开相关摄像头组相关程序和例程

二、模块检测

在车辆组装过程中，因为人为原因，可能出现各个问题，需要在利用一些电气设备和程序的情况下，进行模块检测。在模块检测的时，各个模块间的线，先不插入插座中，待各个模块检测完后，在按照说明连接线路。

- 1、 检测的电源模块。检测方法:给电源模块正常供电后，用万用表测量各个端子的电压是否正常。具体电压，需要查看电源模块说明书。

- 2、 检测核心板，核心板的检测， 需要打开代码检测程序。



，用jlink给系统板供电，下载该程序代码，程序下载后，核心板上指示灯会闪烁。如果接上串口，波特率设置为115200，会不断的发数据出来。可以证明系统版是能够正常工作的。把系统板插到主板上。

- 3、 检测摄像头。连接电源线到主板上，查看核心板的power灯是否亮着。把摄像头排线接到主板上，用万用表测试一下供电电压是不是5V。如果供电正常，把排线插到摄像头上，看摄像头的指示灯是否亮。用视频卡查看，摄像头的图像。下载摄像头测试程序，看串口是否有采集的AD值输出。例如：

- 4、 检测电机驱动。下载电机检测程序，连接好电源线和驱动线。查看电机是否正反转。

- 5、 检测舵机和舵机对正。机械安装好后，先把舵机上面的黑色小圆盘从舵机上拔出来。改装舵接的连接线，舵连接线的改装方法参考主板原理图。把舵机连接线插入主板上，下载舵机测试程序，舵机测试程序会让舵机左右转动。如果舵机能够正常工作，下载舵机对中程序，让舵机对中，安装上黑色小圆盘。

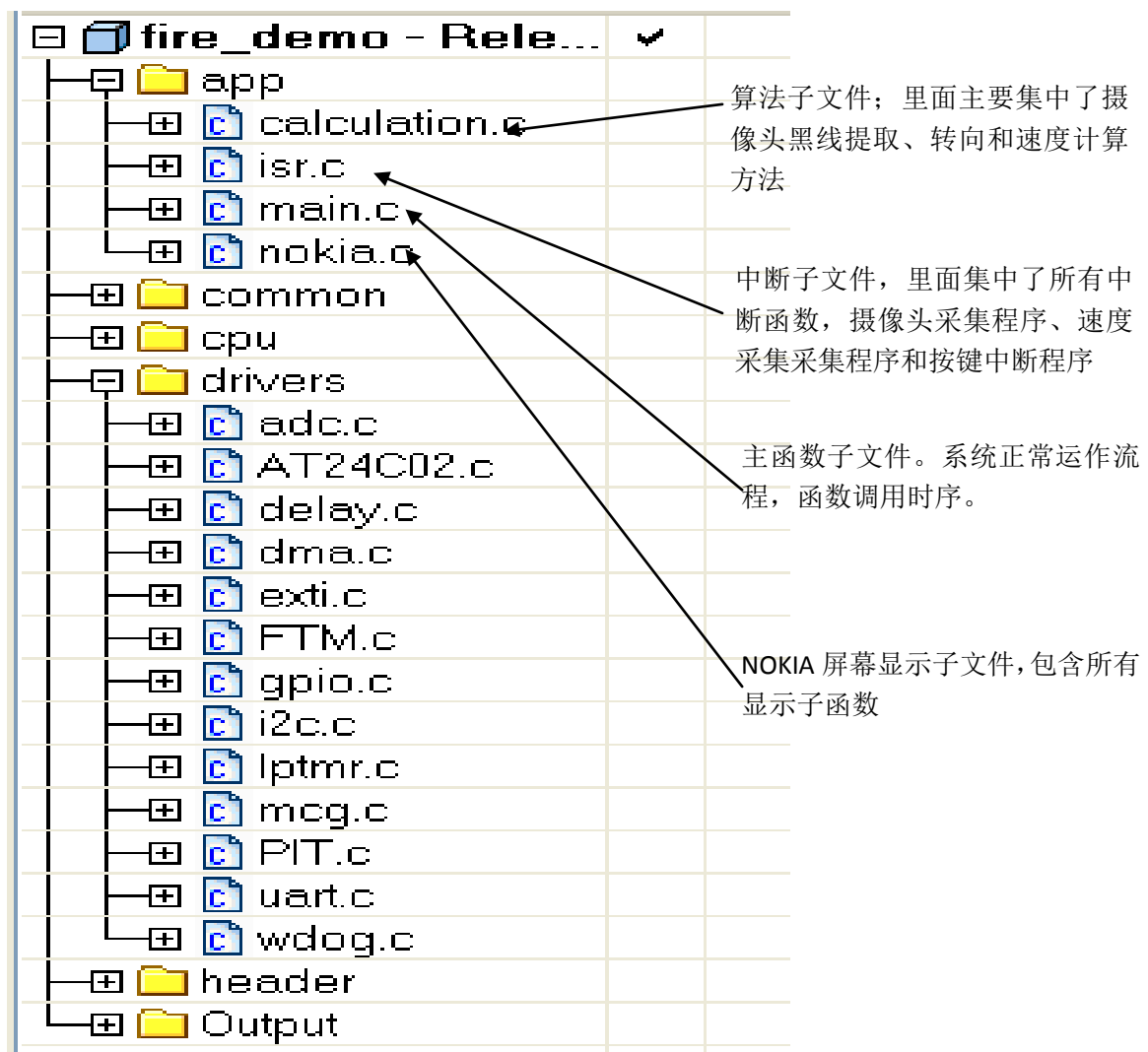
- 6、 检测码盘。安装好码盘后，根据码盘的线的定义和主板的原理图进插座的压制。把插头插入插座中，用示波器连接地和码盘的信号线。转动车轮，查看有没有脉冲信

号。**如图**。如果没有脉冲信号，先检测编码器的供电是否正常。在检查一下码盘是否安装的中心对称，码盘是否能够切割对管。在用白纸遮挡一下对管，看是否有脉冲跳变。

3)、摄像头软件主要包括那几个部分，每个部分的功能和作用以及这几个部分是如何联系在一起的。

①主控制程序

1、子文件介绍



函数介绍

1)、**main函数介绍**: 当程序硬件初始化完成后, 就进入到main函数中, main函数分三部分: 变量初始化、模块初始化、while(1)循环算法。第一个while(1)循环是用按键设置程序参数。第二个while(1)循环为算法模块。

程序详细介绍:

While()部分

1、 计算黑线位置和转向值;

```

/-----
DMA模块数据采样完成后，施移工作。
这里可以与计算黑线联合起来处理，可
*****/

if(DMA_Over_Flg == 1)
{
    DMA_Over_Flg = 0 ;

    if(ALineCal < ALineOverCout ){

        CruteLineAD[ ALineCal ] = Line_calculate(
                                                ALastLeftLinAryy,ALa

```

2、 采集测试和计算PWM值

```

count = LPTMR0_CNCR;
lptmr_counter_clean();
count = LPT_INT_count * LIN_COUT + count;
LPT_INT_count = 0;

PWMCount = SpeedPID( count,0x15);
// FTM_CnV_REG(FTMx[FTM0], CH2) = 300;
PWMC = PWMCount*100/313 ;

if(PWMC > 50)
    PWMC = 50 ;
FTM_PWM_Duty(FTM0 , CH1,0);
FTM_PWM_Duty(FTM0 , CH2,PWMC);

```

2)、calculation 文件函数介绍:

void LCD_KEY_init (void)

按键初始化函数，初始化按键用的定时器和 IO 初始化。

void CCD_INC (void)

摄像头程序初始化，初始化 IO 和 DMA 模块。

void PWM_init (void)

电机驱动 PWM 初始化

void KYELCD(void)

按键处理和显示程序

void LCDTIME(void)

显示程序

```
uint8    Line_calculate(uint8_t    lintcount,uint8_t    LINFA,uint8_t*  
LineLeftAryy,uint8_t*  LineRightAryy, uint8_t * lastLeftAryy, uint8_t  
*lastRightAryy )
```

摄像头二值划程序，该程序主要分：数值二值划，查找黑线位置。

```
int16_t  Turn_Value (u8 LinCout ,u8 *LeftAryy , u8 *RingAryy ,u8  
*CurrlinAD , u8 *linefav)
```

计算转向位置

```
int16_t  SpeedPID(uint16_t speedCount,uint16_t  AmSpeed)
```

电机 PWM 计算。

Isr 中断子文件介绍

void USART1_IRQHandler(void)

USART 中断程序

void PIT1_IRQHandler(void)

PTI1 中断程序

void PIT0_IRQHandler(void)

PIT0 中断程序

void PORTD_IRQHandler(void)

IO 中的 D 口中断程序,该程序里面有摄像头的场中断和行中断,
有 4 个主板的按键中断程序。

void PIT2_IRQHandler(void)

PIT2 是按键定时器中断程序

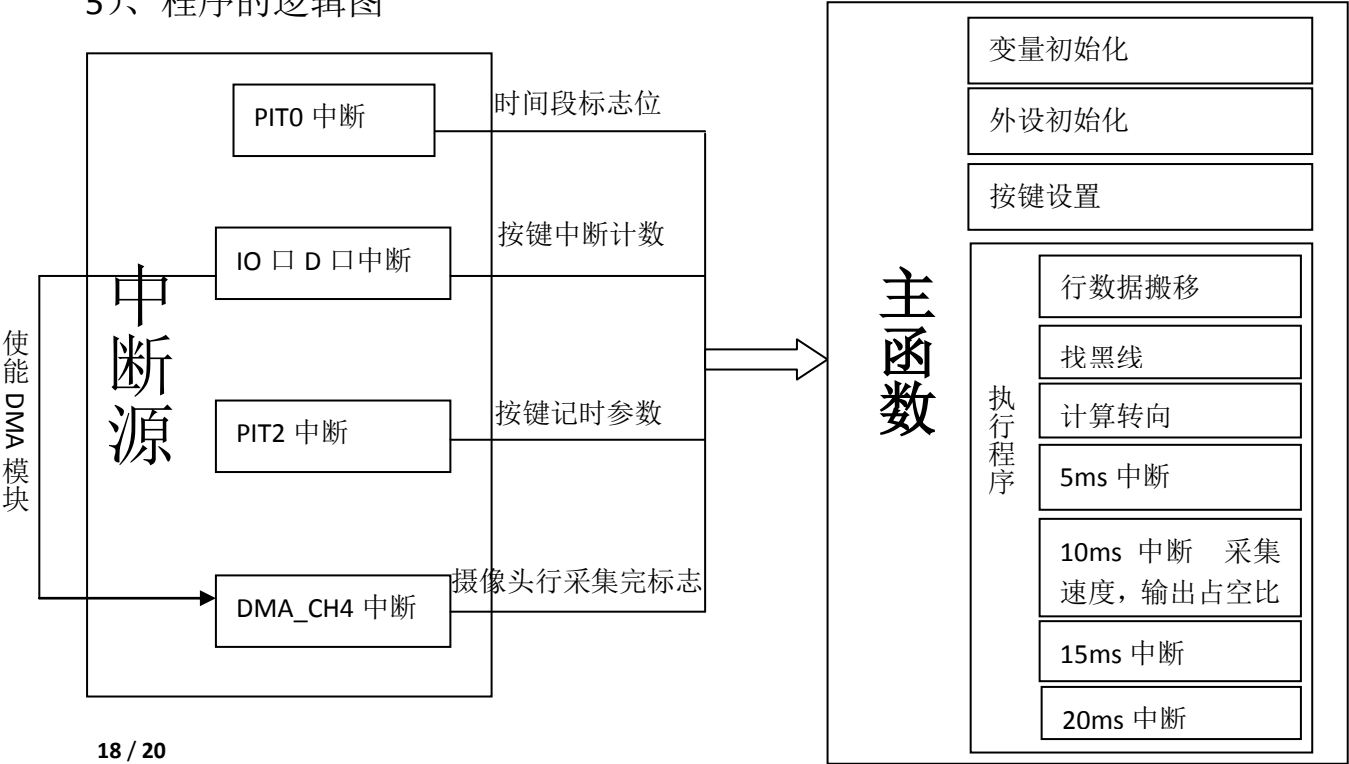
void DMA_CH4_Handler(void)

摄像头 DMA 模块中断

void LPT_Handler(void)

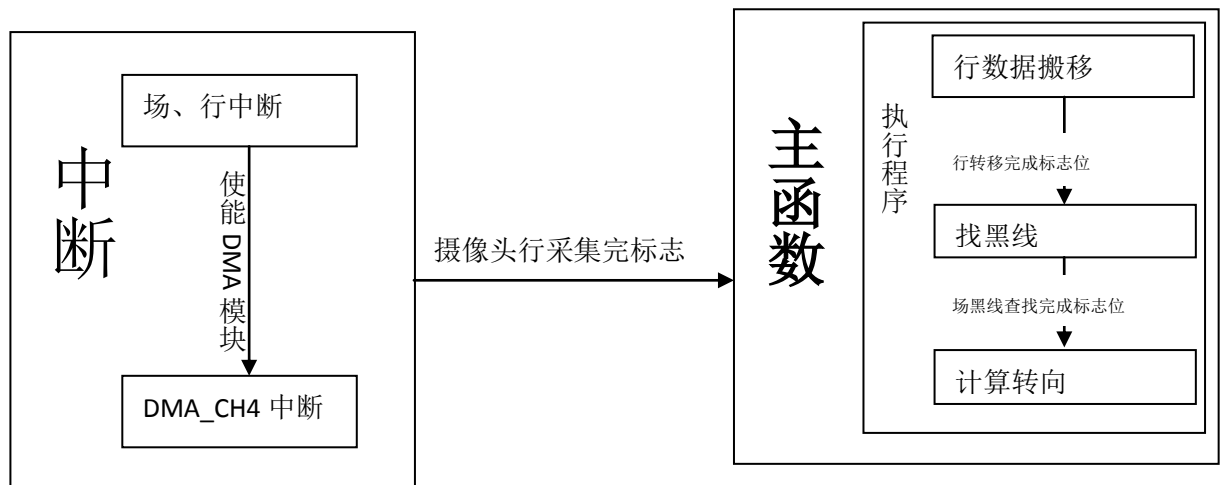
脉冲累加器中断

5)、程序的逻辑图



主要时序详解：

摄像头数据采集处理和转向值计数



6)、程序后面怎么进行升级改进，如何和现在的程序进行搭配连接。

1、采集行数增加。本历程只是在摄像头程序采集中，只用了三行数据，实际的摄像头数据采集，应该为十几行，该行数，应该根据计算程序的大小来定。根据一场数据的时间内，采集多少程序，整个执行程序，能够运行完。需要根据每个人的程序测试。

2、找黑线，本历程的找黑线程序，是先二值划，再根据双线特征，进行黑线查找。同学们，可以根据自己的工作对该程序进行代码优化，能够提升整个程序的运行效率，有精力的同学，应该在此次下一番功夫，因为每行都需要进行黑线查找，运行次数最多的程序。

3、计算转向值，本历程只是用了简单的 **MAP**，进行转向。同学们应该根据整场黑线的趋势，给出一个合适的转向值。在用 **PID** 算法，算出转向 **PWM**。计算转向值程序，是需要根据道路的实际情况结合摄像头前瞻和舵机响应时间，计算出来的值。需要花费大量的时间进

行算法的揣摩。

4、速度算法。本历程中只是用了简单的 PID 值，同学们在做算法时候，需要根据自己赛道是直道和转弯半径给出合适的目标车速。还需要对车速响应做出调整，这里主要是看需要不需要刹车和加速时候的 PID 值。

5、对机械部分的调整。这个同学们需要结合自己的算法，看需要调整车的状态，让车能够更快的响应车速和转向。从而使自己的车速能够迅速提高。