**RH850\_F1L测试文档**

V01

2017-05-11

**发布历史：**

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **变更内容** | **发布者** |
| 2017-05-11 | V01 | 初版建立ready | Tan Fumin |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**目录**

[1. 上电时序 5](#_Toc482273301)

[1.1. 测试准备 5](#_Toc482273302)

[1.2. 上电时序的测试监控 5](#_Toc482273303)

[2. 时钟模块 8](#_Toc482273304)

[2.1. 内部高速时钟频率和幅值验证 8](#_Toc482273305)

[2.1.1. 参数配置 8](#_Toc482273306)

[2.1.2. 实际测试波形与分析 8](#_Toc482273307)

[2.2. 内部低速时钟频率和赋值验证 9](#_Toc482273308)

[2.2.1. 参数配置 9](#_Toc482273309)

[2.2.2. 实际测试波形与分析 10](#_Toc482273310)

[2.3. 外部晶振起振时间和等待时间验证 10](#_Toc482273311)

[2.3.1. 参数配置 11](#_Toc482273312)

[2.3.2. 实际测试波形与分析 12](#_Toc482273313)

[2.4. 内部PLL时钟验证 14](#_Toc482273314)

[2.4.1. 参数配置 14](#_Toc482273315)

[2.4.2. 实际测试波形 15](#_Toc482273316)

[2.5. 本模块测试结果 16](#_Toc482273317)

[3. 看门狗模块 17](#_Toc482273318)

[3.1. 测试准备 17](#_Toc482273319)

[3.2. 看门狗的超时时间测试验证 17](#_Toc482273320)

[3.2.1. 参数配置 17](#_Toc482273321)

[3.3. 模块测试结果 21](#_Toc482273322)

[4. OSTM模块 22](#_Toc482273323)

[4.1. 测试准备 22](#_Toc482273324)

[4.2. 模块测试结论 23](#_Toc482273325)

[5. 休眠唤醒模块 24](#_Toc482273326)

[5.1. 测试准备 24](#_Toc482273327)

[5.2. 测试结果 25](#_Toc482273328)

[6. Port模块 27](#_Toc482273329)

[6.1. P\_Port验证 27](#_Toc482273330)

[6.1.1. P\_Port测试验证步骤 27](#_Toc482273331)

[6.1.2. P\_Port参数配置 28](#_Toc482273332)

[6.1.3. P\_Port测试结果 29](#_Toc482273333)

[6.2. J\_Port验证 30](#_Toc482273334)

[6.2.1. J\_Port测试验证 30](#_Toc482273335)

[6.2.2. J\_Port参数配置 30](#_Toc482273336)

[6.2.3. J\_Port测试结果 32](#_Toc482273337)

[6.3. A\_Port验证 33](#_Toc482273338)

[6.3.1. A\_Port测试验证步骤 33](#_Toc482273339)

[6.3.2. 具体配置如下 33](#_Toc482273340)

[6.3.3. A\_Port测试结果 35](#_Toc482273341)

[6.4. 模块测试结论 35](#_Toc482273342)

[7. 外部中断模块 36](#_Toc482273343)

[7.1. 测试准备 36](#_Toc482273344)

[7.2. 测试结果 37](#_Toc482273345)

[8. ADC模块 39](#_Toc482273346)

[8.1. 测试准备 39](#_Toc482273347)

[8.1.1. 设定ADC模块， 39](#_Toc482273348)

[8.2. 测试结果 40](#_Toc482273349)

[9. PWGA模块 43](#_Toc482273350)

[9.1. PWGA测试准备 43](#_Toc482273351)

[9.2. 测试波形 44](#_Toc482273352)

[9.3. 模块测试结果 48](#_Toc482273353)

[10. TAUx模块 49](#_Toc482273354)

[10.1. 定时器功能、计数器功能和脉宽测量功能的验证 49](#_Toc482273355)

[10.1.1. 验证方法 49](#_Toc482273356)

[10.1.2. 参数设定 50](#_Toc482273357)

[10.1.3. 测试结果 51](#_Toc482273358)

[10.2. PPG的主节点与从节点验证及PPG特征接口验证 52](#_Toc482273359)

[10.2.1. 验证方法 52](#_Toc482273360)

[10.2.2. 测试结果 53](#_Toc482273361)

[10.3. 分频器输出的验证 55](#_Toc482273362)

[10.3.1. 验证方法 55](#_Toc482273363)

[10.3.2. 测试结果 55](#_Toc482273364)

[11. RLIN模块的UART 功能 56](#_Toc482273365)

[11.1. 测试准备 56](#_Toc482273366)

[11.2. 测试结果 57](#_Toc482273367)

[12. IIC模块 59](#_Toc482273368)

[12.1. 测试准备 59](#_Toc482273369)

[12.2. 测试结果 61](#_Toc482273370)

[13. CSIH模块 63](#_Toc482273371)

[13.1. 测试准备 63](#_Toc482273372)

[13.2. 测试结论 65](#_Toc482273373)

[14. CSIG模块 67](#_Toc482273374)

[14.1. 测试准备 67](#_Toc482273375)

[14.2. 测试结果 68](#_Toc482273376)

[14.2.1. 缓存结果 68](#_Toc482273377)

[14.2.2. 实际波形 69](#_Toc482273378)

[14.2.3. 测试结论 70](#_Toc482273379)

[15. DMA模块 71](#_Toc482273380)

[15.1. 测试准备 71](#_Toc482273381)

[15.2. 测试结果 73](#_Toc482273382)

[15.2.1. 测试波形 73](#_Toc482273383)

[15.2.2. 测试结论 74](#_Toc482273384)

[16. 附录 75](#_Toc482273385)

[16.1. 图表 75](#_Toc482273386)

# 上电时序

测试目的：验证MCU上电时序的具体动作所需时间

测试范围：VCC供电开始到用户级别的函数调用接口

测试限制条件：MCU内部初始化，汇编代码执行前的时间无法具体测试得出

## 测试准备

备注：option byte 设定需要设定为： 0x9AAFFFFF

看门狗为软件启动，看门狗窗口为100%，看门狗的复位时间通过配置文件进行，MCU内部的电压监测功能开启。

## 上电时序的测试监控

A：汇编开始执行时，设定监控引脚，以监测VCC到软件执行的具体时间

B：内部RAM清除完成以后，设定监控引脚，以监测RAM清除的执行时间

C：进入主main函数时，设定监控引脚，以监测lib函数的执行时间

D：进入用户级别的函数时，设定监控引脚，以监测MCU资源的初始化执行时间

芯片说明书参考如图 1‑1芯片上电时序，通过图 1‑2上电测试波形我们可以知道在软件代码起来以前，MCU内部已经开始执行内部晶振启动且等待稳定，片上flash的初始化等动作。

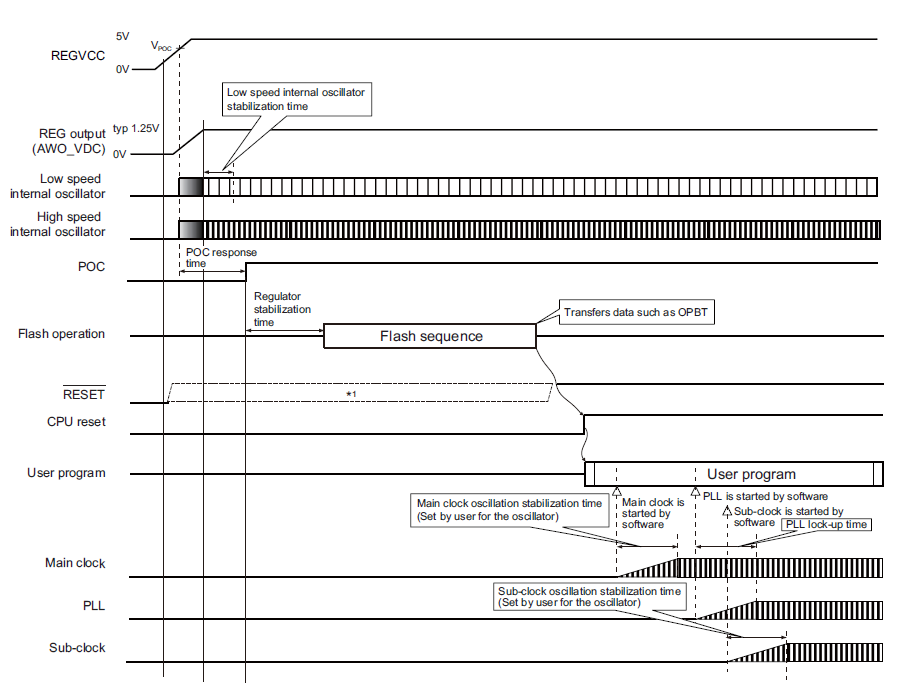


图 1‑1芯片上电时序

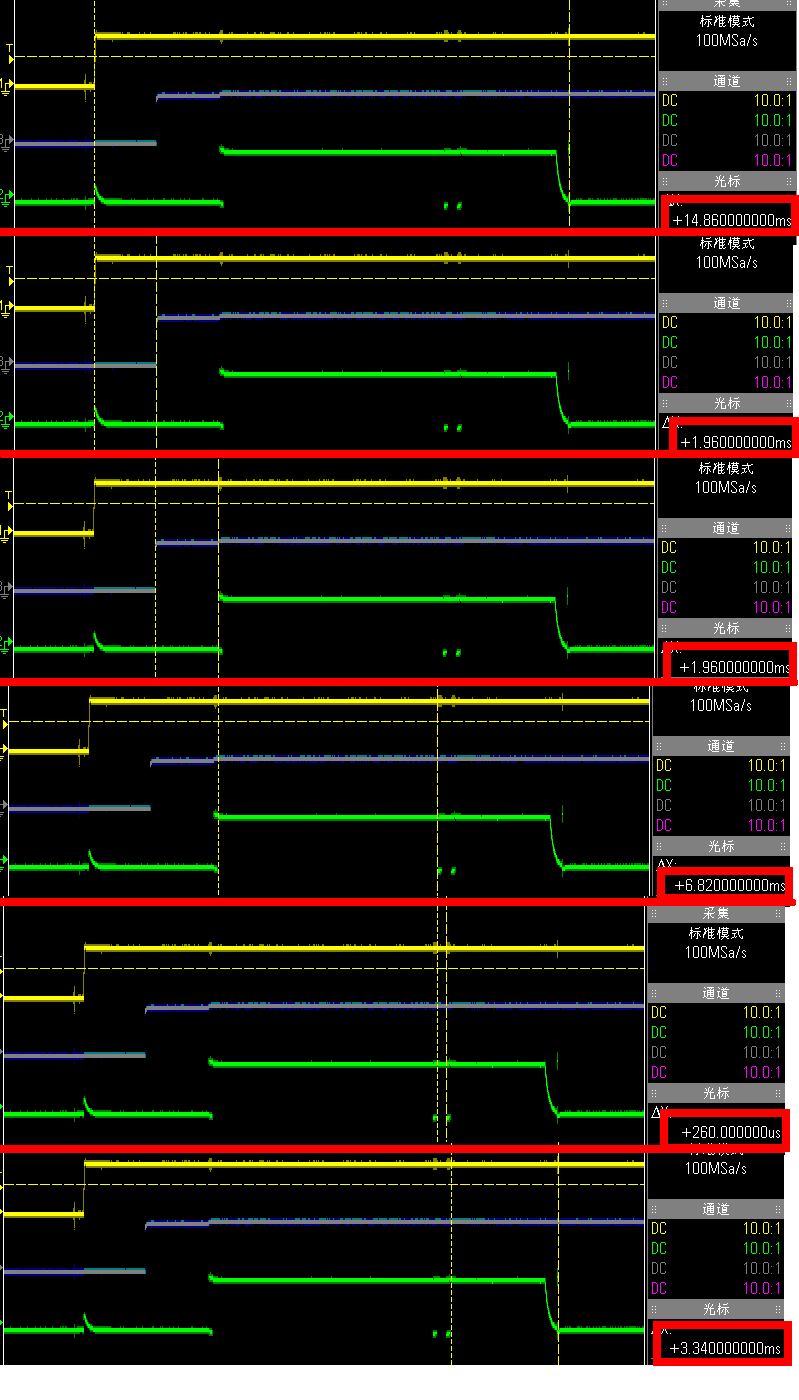


图 1‑2上电测试波形

# 时钟模块

测试目的：验证时钟模块的实际运行状态与配置值是否符合

测试范围：所有能监测的MCU内部时钟

测试限制条件：时钟输出模块的引脚输出最大的频率为10MHz

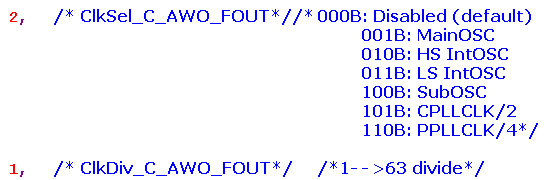
## 内部高速时钟频率和幅值验证

### 参数配置

配置时钟输出的引脚功功能，使用时钟输出选项



配置不同的时钟源及其分频系数：采用不分频，直接输出内部高速晶振



### 实际测试波形与分析

经过测试如图 2‑1内部高速时钟波形，可得内部时钟频率为7.7MHz，与标称值8MHz有4%的误差，不能作为实际项目的时钟源使用，否则MCU的时钟基准将有误差！！

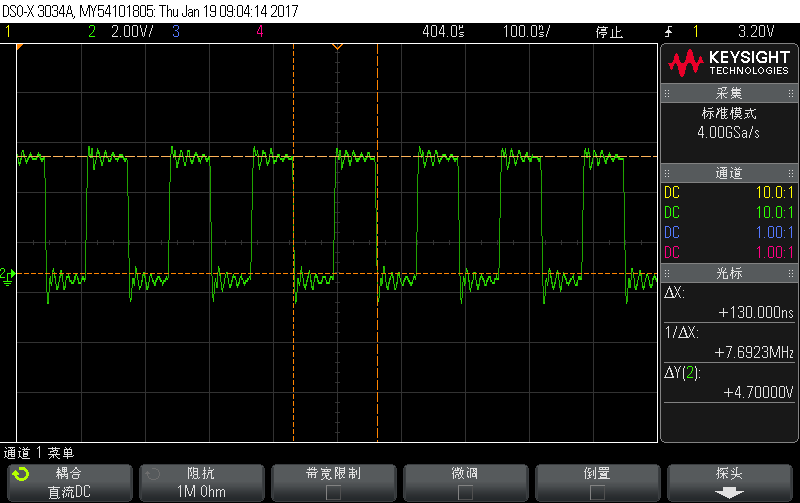


图 2‑1内部高速时钟波形

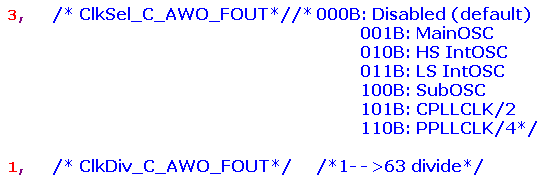
## 内部低速时钟频率和赋值验证

### 参数配置

配置时钟输出的引脚功功能，使用时钟输出选项



配置不同的时钟源及其分频系数：采用不分频，直接输出内部低速晶振



### 实际测试波形与分析

经过测试如图 2‑2内部低速时钟波形，内部时钟频率为250kHz，与标称值240kHz误差为4%。

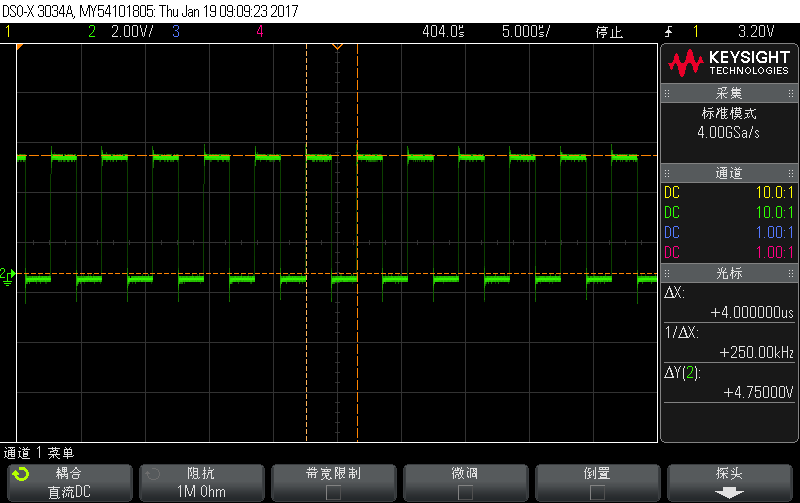


图 2‑2内部低速时钟波形

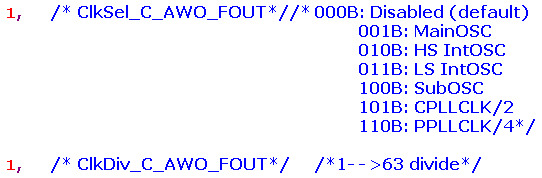
## 外部晶振起振时间和等待时间验证

### 参数配置

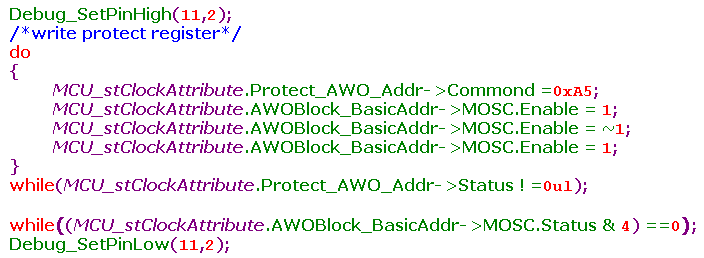
1. 配置时钟输出的引脚功功能，使用时钟输出选项



1. 配置不同的时钟源及其分频系数：采用不分频，直接输出外部晶振



1. 采用一个监控引脚监控晶振稳定时间的设定是否与期望值一致。



### 实际测试波形与分析

#### 晶振起振时间验证

经过测试如图 2‑3晶振起振波形，晶振起振时间为400us。所以，在工程应用中，晶振稳定时间的等待配置不能小于400us：

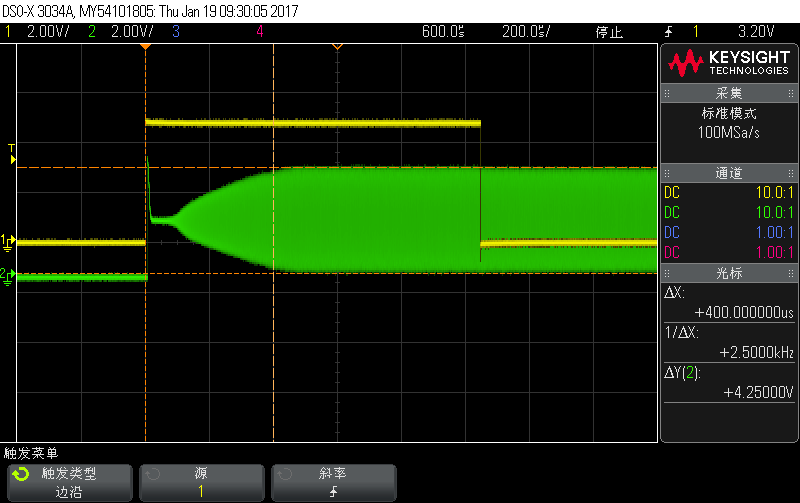


图 2‑3晶振起振波形

#### 晶振稳定时间验证

晶振配置的验证设定值为基于内部晶振频率8MHz时配置1000us，实际测试得出内部晶振的频率为7.6Mhz，所以实际测试值比设定值要长，测试结果 = 1000\*(8MHZ/7.6MHZ)=1052us; 符合实际的计算结果

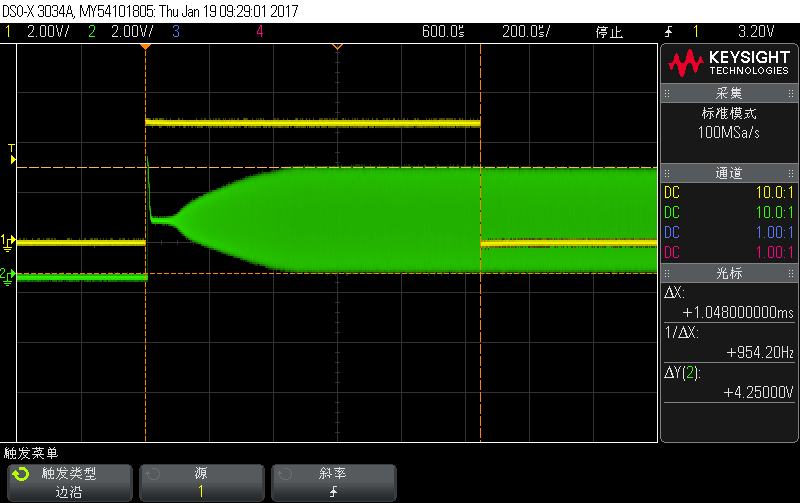


图 2‑4晶振起振和稳定波形

#### 外部晶振频率验证

标称值为16MHz，实际测试频率为16MHz，符合实际结果如图 2‑5实际外部晶振波形。

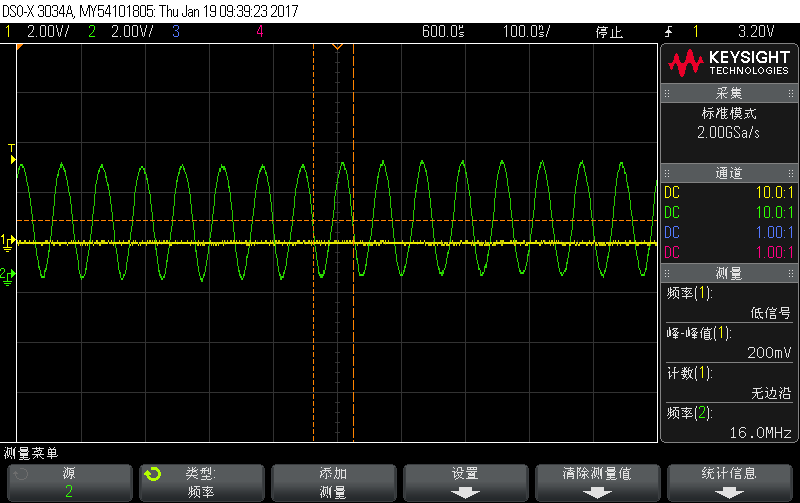


图 2‑5外部晶振波形

## 内部PLL时钟验证

### 参数配置

RH850内部的PLL输出为2路同频率的分支如图 2‑6RH850内部PLL输出，所以只需要对其中一路进行测试验证即可。

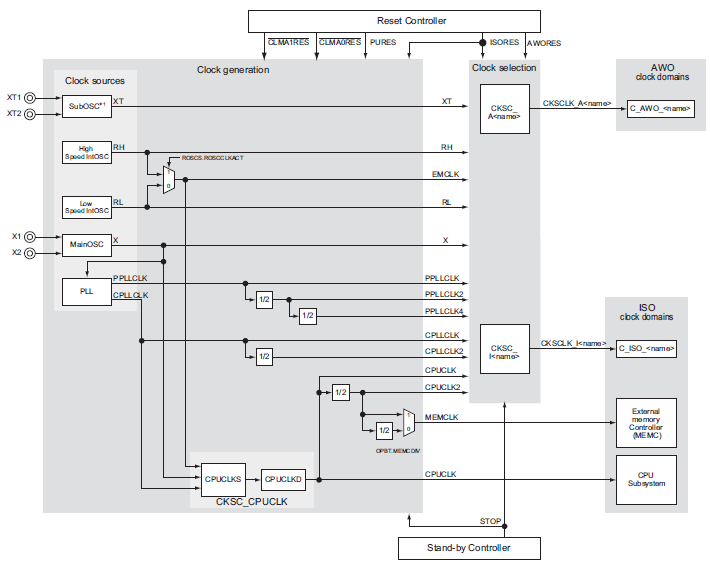
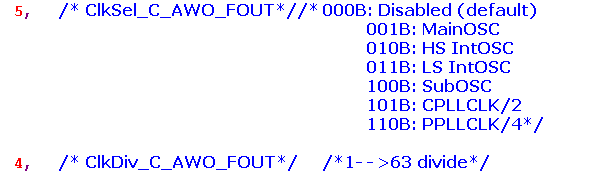


图 2‑6RH850内部PLL输出

1. 设定时钟输出选择为外设的2分频，输出分频选择4分频



1. 设定PLL模块的输出频率为64MHz



### 实际测试波形

实际测试如图 2‑7内部PLL时钟波形  
测试分析：由于PLL设定64Mhz，经过2\*4分频以后的计算值为64MHz/(2\*4)=8Mhz，实际测试波形输出符合计算值。

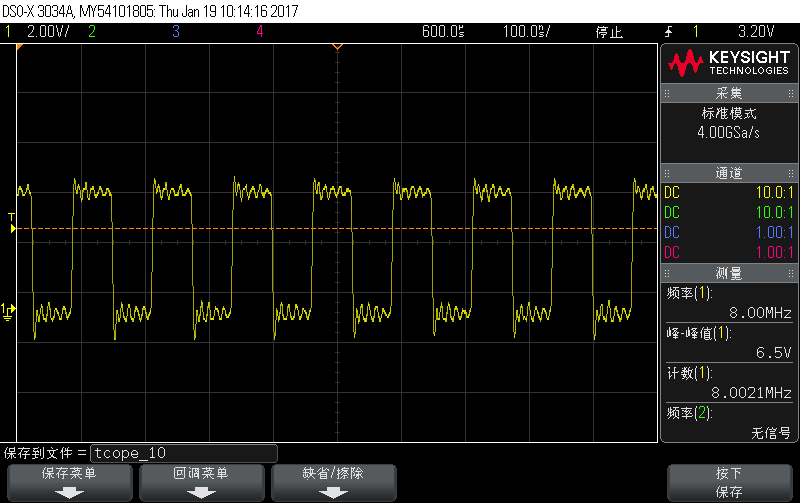


图 2‑7内部PLL时钟波形

## 本模块测试结果

本模块进行的验证测试，实际测试结果符合设定目标值。时钟模块AWO，ISO及基础的MainOSC,内部高速晶振，内部低速晶振的参数设定与实际测试结果是相符合的。下面的接口函数功能实现正常：

1. MCU\_u16ClockInit()
2. MCU\_vClockInit\_Basic()
3. MCU\_vClockInit\_AWO()
4. MCU\_vClockInit\_ISO()

# 看门狗模块

测试目的：验证看门狗模块的实际运行状态与配置值是否符合

测试范围：片内全部看门狗单元watchdog\_0，watchdog\_1

测试限制条件： 测试时，为防止中途复位，可以采用看门狗中断功能进行

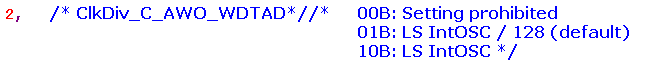
## 测试准备

在Option Byte中开启看门狗使能，且启动方式为软件触发，测试时看门狗使用FENMI功能，用于测试看门狗超时的时间长度，正常项目使用时开启reset功能

## 看门狗的超时时间测试验证

### 参数配置

1. 在时钟配置模块，将看门狗的时钟源都采用内部240kHz的低速时钟，watch\_dog0时钟配置Watch\_dog1时钟配置只能使用默认值，为内部低速时钟。



1. 使用烧录软件，对芯片的optbyte0 进行设定如图 3‑1芯片optbyte0设定，将其设定为0xBAAFFFFF，使能2个看门狗，使用软件方式触发开始，在看门狗初始化时，进行监控引脚的设定，看门狗采用中断方式进行测试。

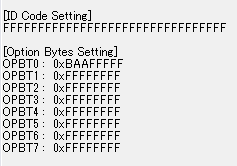
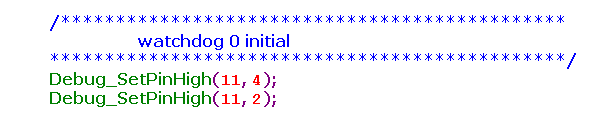


图 3‑1芯片optbyte0设定





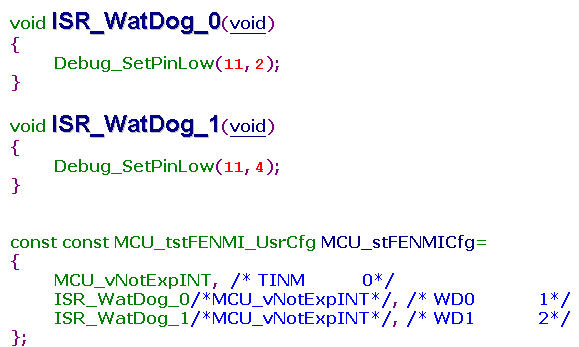


1. 对各个看门狗的越界时间进行配置：



1. 由于看门狗的中断属于FENMI中断，使用看门狗的中断方式进行越界周期的测试。

在诊断服务函数FENMI中配置看门狗越界时的引脚监控状态，在进入MCU\_vDebugMain函数以后，不对看门狗进行喂狗



1. 参数设定的时间为：

Watch\_dog0 越界时间 = 2^（3+9）/ 240kHz = 17ms

Watch\_dog1 越界时间 = 2^（6+9）/ 240kHz = 136ms

#### 实际测试波形与分析

看门狗模块的定时器时间监控的波形如图 3‑2和图 3‑3，可知设定值与计算值符合

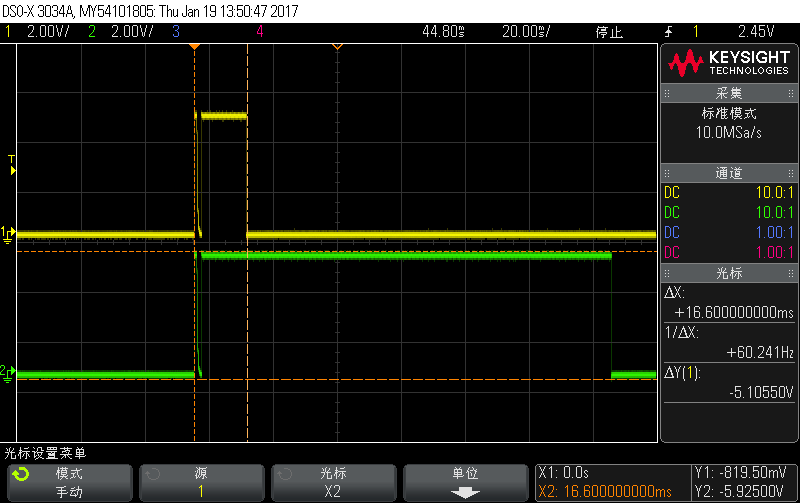


图 3‑2看门狗0定时器监控波形

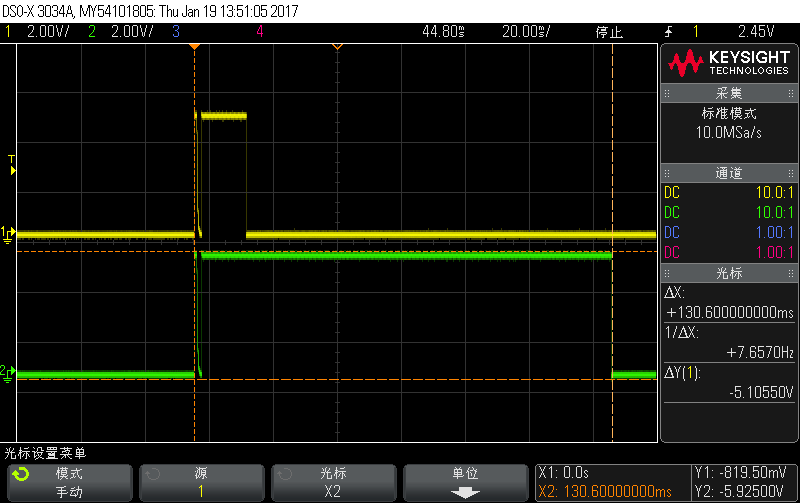


图 3‑3看门狗1定时器监控波形

将看门狗越界的响应改为复位模式时，测试引脚输出的实际电平如图 3‑4，

只对WD1进行喂狗和WD0，WD1都不喂狗的波形相同，因为WD0越界复位，

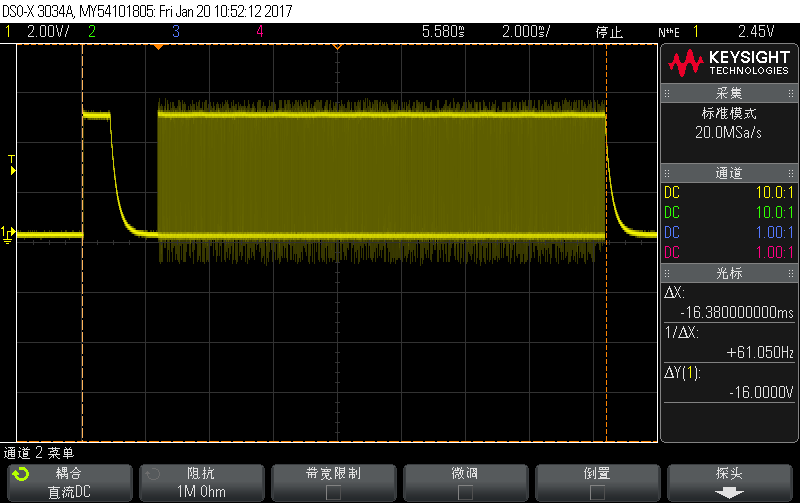


图 3‑4复位模式下只喂WD1的测试波形

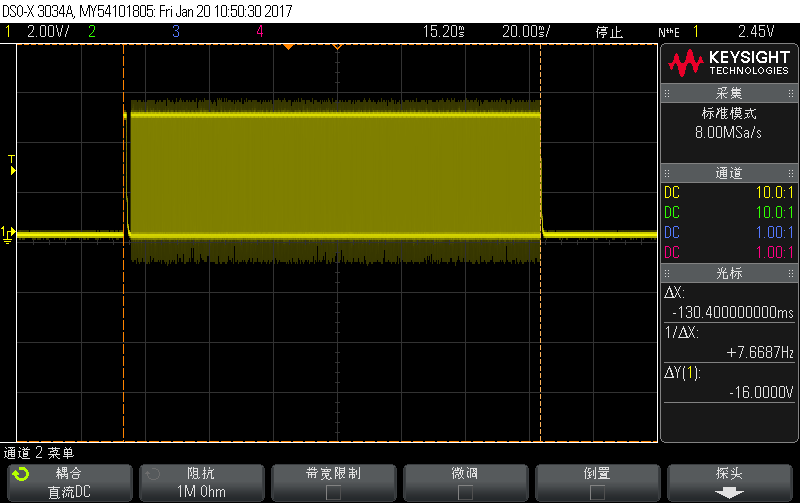


图 3‑5复位模式下只喂WD0的测试波形

## 模块测试结果

本模块进行的验证测试，实际测试结果符合设定目标值。在越界时间内喂狗时，MCU不会产生复位信号或者中断信号。本模块的optbyte中关于看门狗的相关的配置测试正常，optbyte设定为软件触发启动方式的配置正常，看门狗的时间参数配置与实际测试时间相符，看门狗的中断方式响应正常，即可印证下面接口功能正常：

1. MCU\_vWatchDogInit()；
2. MCU\_vWatchDogClean();

# OSTM模块

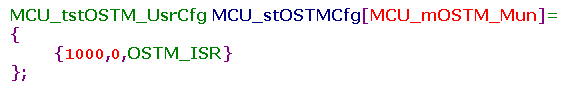
测试目的：验证OSTM模块的实际运行状态与配置值是否符合

测试范围：OSTM的启动，停止，变更中断间隔

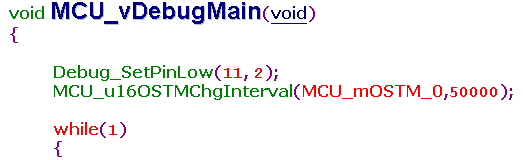
测试限制条件： 无

## 测试准备

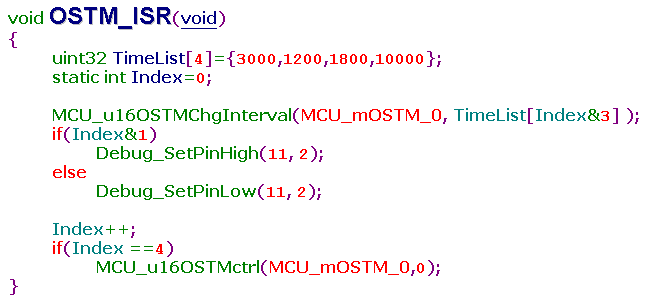
1. 将CPU频率设定为PLL到64MHz，OSTM使用的是CPU频率的1/2，即为32MHz
2. 设定OSTM的中断服务函数，且复位时不启动



1. 在进入DEBUG循环前开启OSTM，且设定时间中断为50ms



1. 在中断服务函数中，设定不同的时间间隔，且完成以后，关闭定时器



## 模块测试结论

查看具体的检测硬件输出的波形与设定值是否吻合，分析实际测试的波形如可知图 4‑1，所有的设定和实际的测试结果均吻合

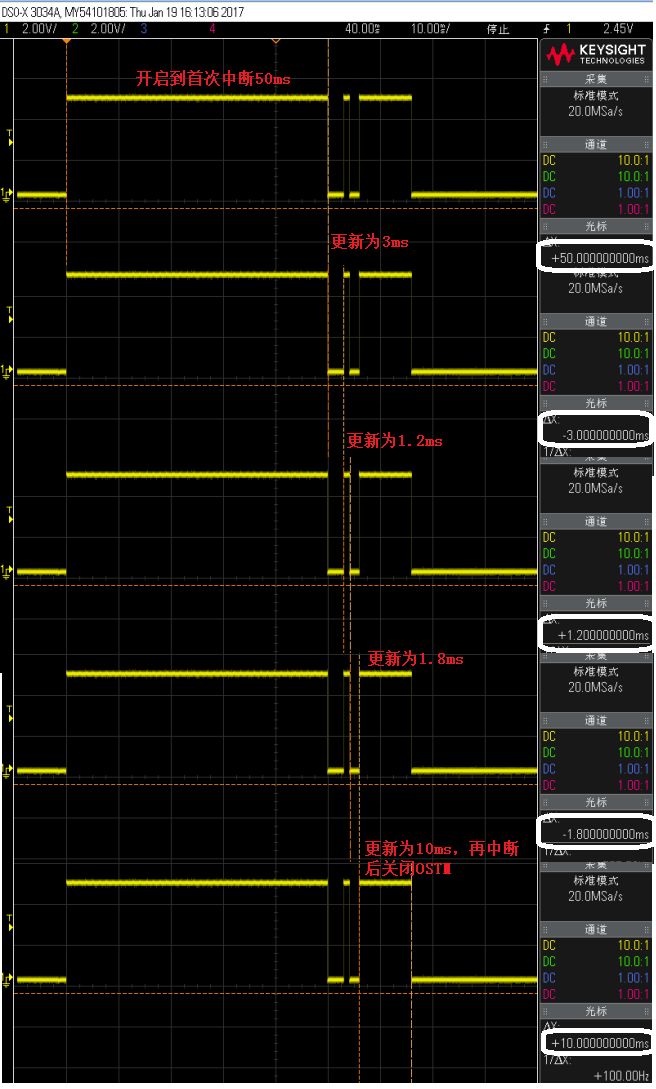


图 4‑1OSTM验证波形

# 休眠唤醒模块

测试目的：验证MCU的实际运行状态与配置值是否符合

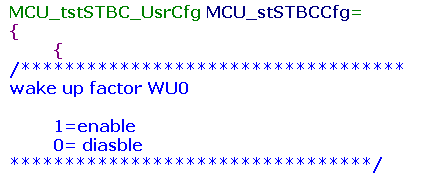
测试范围：定时器的启动，停止，外部中断触发，MCU的运行状态

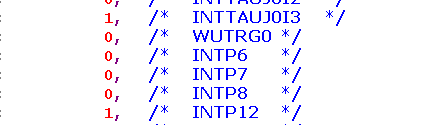
测试限制条件：

## 测试准备

总体测试的方案为使用两个唤醒中断，1个为1秒钟定时器中断唤醒，唤醒以后操作监控引脚，再次进入休眠，1个为外部中断唤醒，唤醒以后执行正常代码，使用while（1）对另外一个监控引脚操作，直到定时器中断以后再次进行申请休眠。

1. 设定唤醒中断源使能

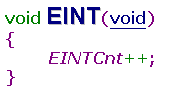


1. 使用TAUJ0I3做定时器唤醒功能，使用外部中断12做手动唤醒功能
2. 配置定时器和外部中断

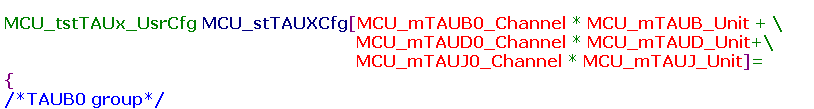


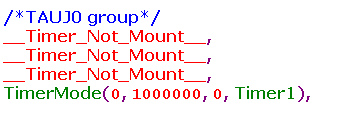


1. 中断函数正常运行代码

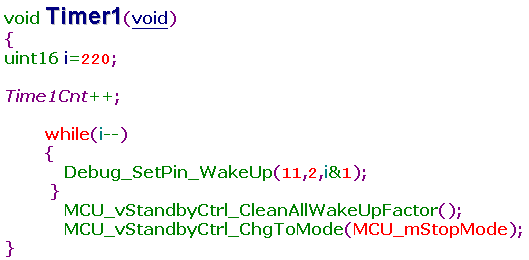


1. 定时器配置

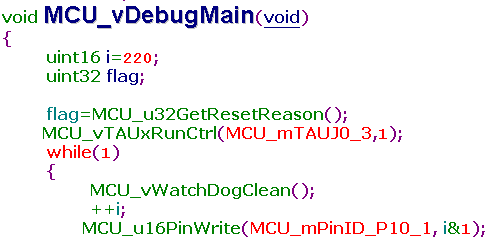




1. 定时器进行引脚监控和再次休眠请求

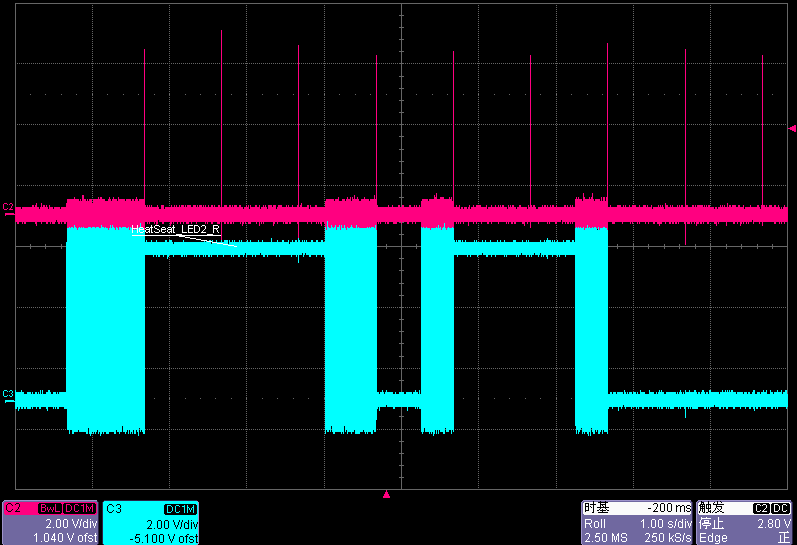


1. Main函数里面进行正常运行状态下的引脚监控



## 测试结果

1. 运行到main函数时，开启定时器，并且正常执行while（1）里面的代码
2. 定时器中断进入，对监控引脚进行操作以后进入休眠模式，此时介绍执行代码
3. 每一次定时器中断进入都会唤醒MCU，并且在中断里面对监控引脚进行操作，再次进入休眠
4. 在休眠时，手动触发外部中断，外部中断唤醒MCU，且执行while（1）的代码，直到定时器进入休眠
5. 由此反复外部触发以唤醒MCU，运行代码，到定时器中断时，再次进入休眠



# Port模块

测试目的：验证MCU的实际运行状态与配置值是否符合

测试范围：定时器的启动，停止，外部中断触发，MCU的运行状态

测试限制条件：

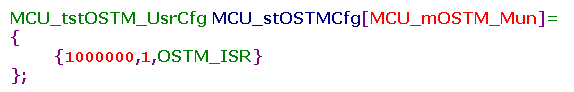
## P\_Port验证

### P\_Port测试验证步骤

1. 启用定时器1s，配置PWGA\_27号PPG并且启动，将P11\_4，P11\_2相连，P11\_3用于输出端口监测
2. 在中断进行端口配置和引脚输出，P11\_4输出电平，P11\_2读入电平，while(1)内将P11\_2的读入数据从P11\_3端口输出
3. 再次进入中断时，设定0.5s定时器，P11\_4输入电平，P11\_2输出电平，while(1)内将P11\_4的读入数据从P11\_3端口输出
4. 再次进入中断时，将P11\_2配置成PWGA功能输出模式，while(1)内将P11\_4的读入数据从P11\_3端口输出

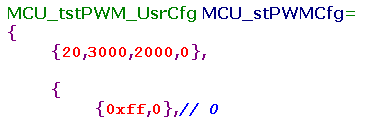
### P\_Port参数配置

#### 定时器配置：



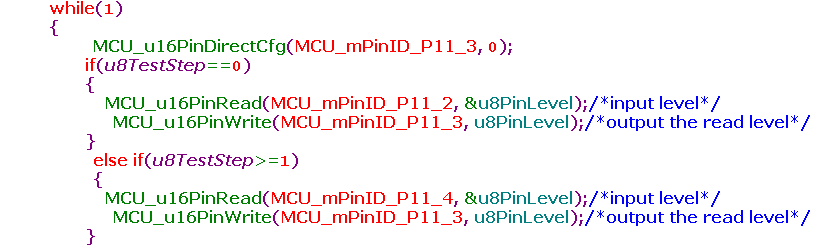


#### PWGA设置





#### Debug\_main设置



### P\_Port测试结果

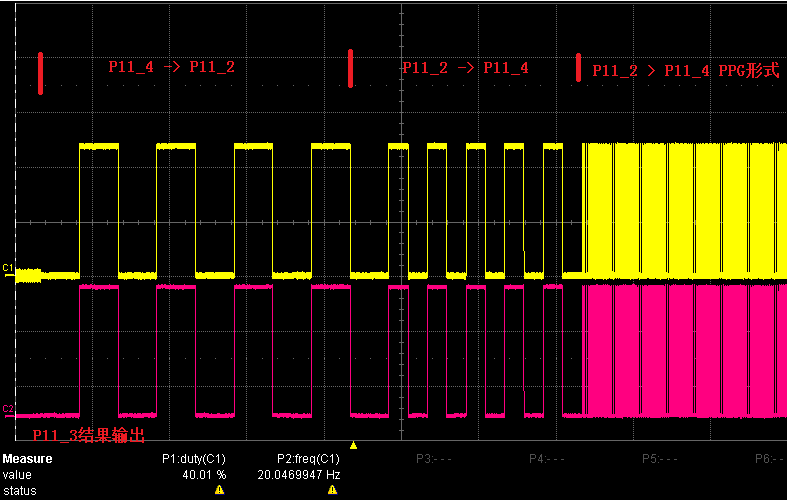


图 6‑1P Port验证波形

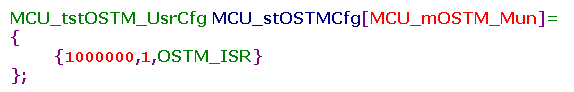
## J\_Port验证

### J\_Port测试验证

1. 启用定时器1s，配置TAUJ0\_3号PPG并且启动，将JP0\_4，JP0\_5相连，JP0\_3用于输出端口监测
2. 在中断进行端口配置和引脚输出，JP0\_4输出电平，JP0\_5读入电平，while(1)内将JP0\_5的读入数据从JP0\_3端口输出
3. 再次进入中断时，设定0.5s定时器，JP0\_4输入电平，JP0\_5输出电平，while(1)内将JP0\_4的读入数据从JP0\_3端口输出
4. 再次进入中断时，将JP0\_5配置成PPG功能输出模式，while(1)内将JP0\_4的读入数据从JP0\_3端口输出

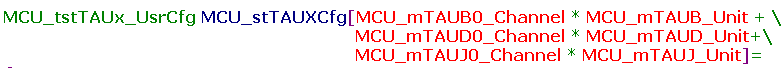
### J\_Port参数配置

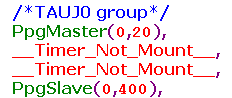
#### 定时器配置：





#### TAUJ设置





#### Debug\_main中设置：



### J\_Port测试结果

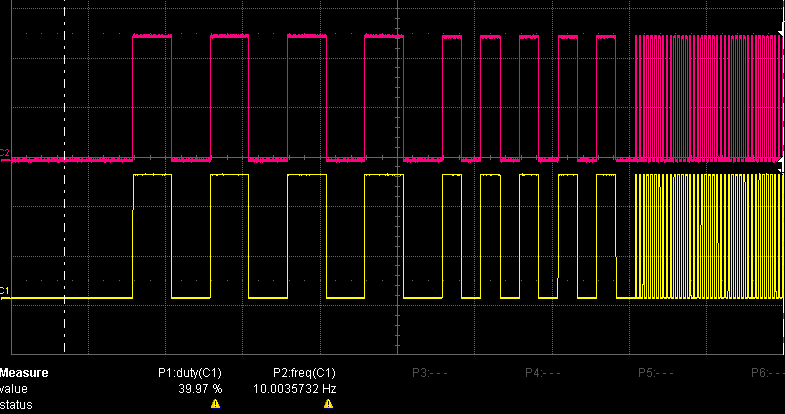


图 6‑2J\_Port验证波形

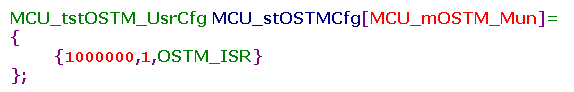
## A\_Port验证

### A\_Port测试验证步骤

1. 启用定时器1s，将AP1\_5，AP0\_15相连，AP1\_1用于输出端口监测
2. 在中断进行端口配置和引脚输出，AP1\_5输出电平，AP0\_15读入电平，while(1)内将AP0\_15的读入数据从AP1\_1端口输出
3. 再次进入中断时，设定0.5s定时器，AP1\_5输入电平，AP0\_15输出电平，while(1)内将AP1\_5的读入数据从AP1\_1端口输出

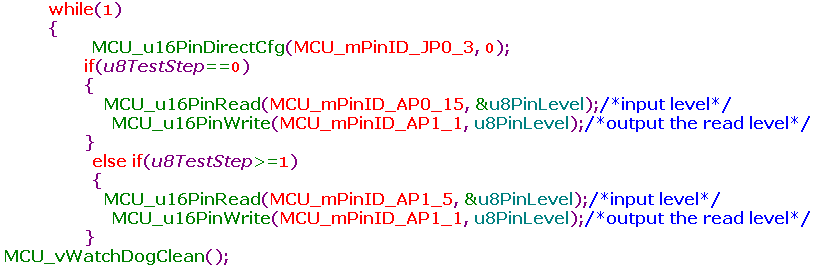
### 具体配置如下

#### 定时器配置





#### Debug\_main中设置：



### A\_Port测试结果

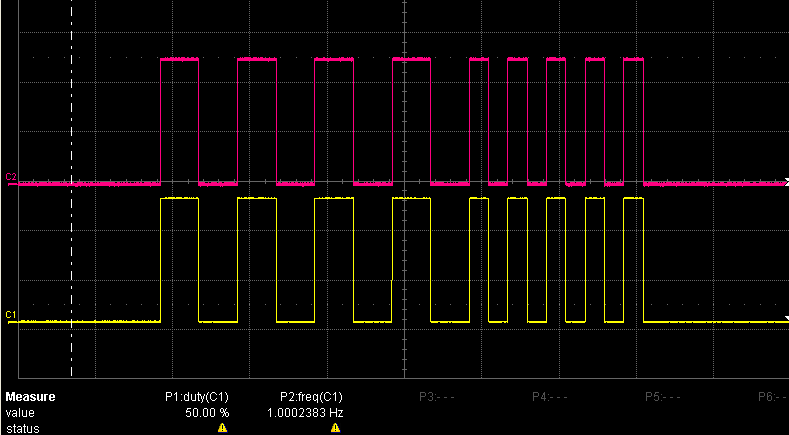


图 6‑3A\_Pprt测试波形

## 模块测试结论

经过测试，以下接口验证通过：

1. MCU\_u16PinDirectCfg
2. MCU\_u16PinRead
3. MCU\_u16PinWrite
4. MCU\_u16PinAltFunCfg
5. MCU\_u16PortInit

# 外部中断模块

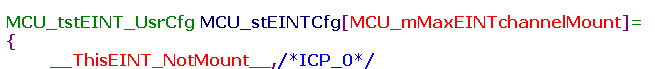
测试目的：验证MCU的实际运行状态与配置值是否符合

测试范围：定时器的启动，停止，外部中断触发，MCU的运行状态

测试限制条件：

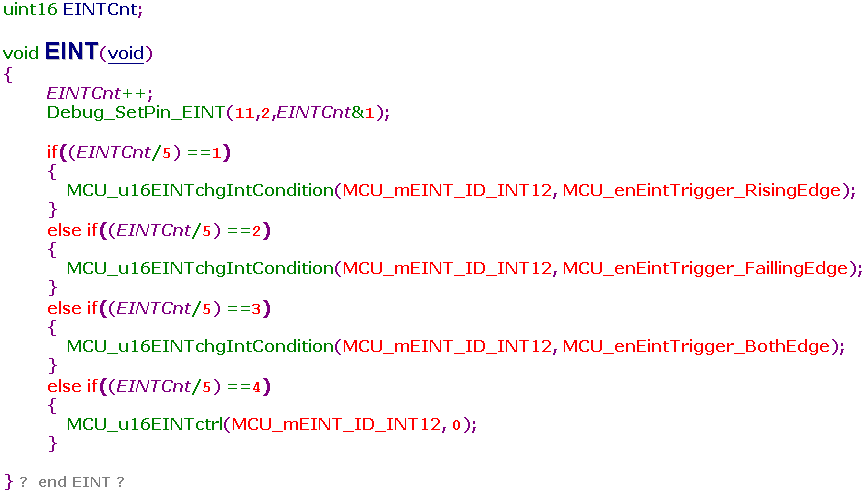
## 测试准备

1. 配置外部中断

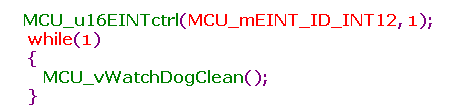




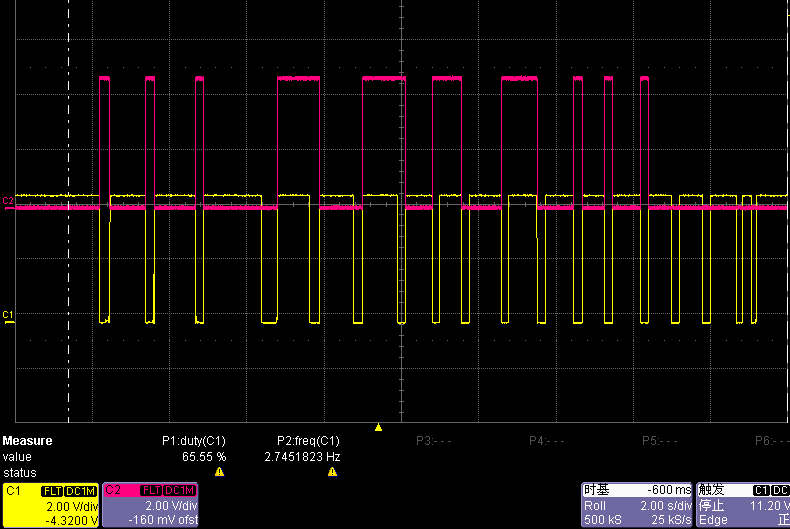
1. 中断服务测试函数的编写，使用每中断5次变更一次触发条件，最后则关闭中断请求的使能.



1. 主函数入口，使能外部中断检测



## 测试结果



A:双边

B:上升沿

C:上升沿

D:双边

E:关闭

引脚输入

中断监控

图 7‑1外部中断验证波形

A：外部中断初始化以后，设定中断的触发条件为双边沿

B：5次双边沿触发以后，将中断触发条件为修改为上升沿

C：5次上升沿触发以后，将中断触发条件为修改为下降沿

D：5次下降沿触发以后，将中断触发条件为修改为双边沿

E：5次双边沿触发以后，将中断触发条件关闭，此时以后将不再响应中断

以上实际测试符合设计条件。

# ADC模块

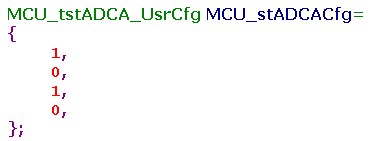
测试目的：验证MCU的ADC转换与实际引脚输入的波形是否符合

测试范围：ADC采样的结果

测试限制条件：ADC模块本身采用单次采样的方式，start一次且采集完成以后，ADC自动停止采样，直到下次start函数被调用，且ADC采样不使用中断方式

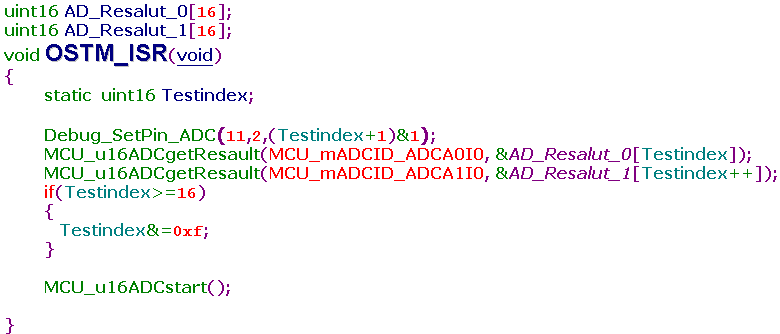
## 测试准备

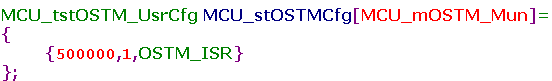
### 设定ADC模块，



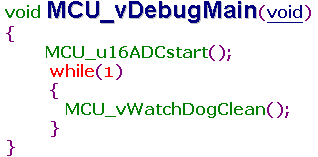
备注：引脚的特殊功能不需要配置功能选择，自动使能特殊功能，非特殊功能则需要配置相应的功能选择

1. 配置一个定时器用于监控数字采样时的模拟输入的电压值，0.5秒采样一次，使用手动输入变阻器的方式进行测试





1. Main函数运行时启动一次采样



## 测试结果

1. 仿真监控的结果如下，下图为ADC结果与转换以后的采集电压

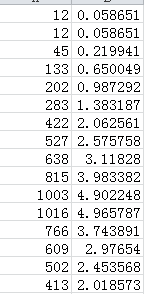
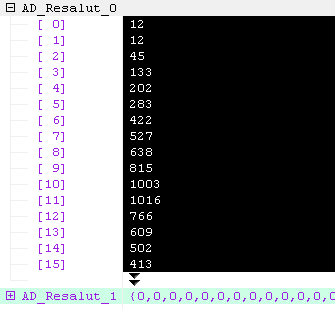


图 8‑1ADC仿真结果

1. 输入波形图与电压

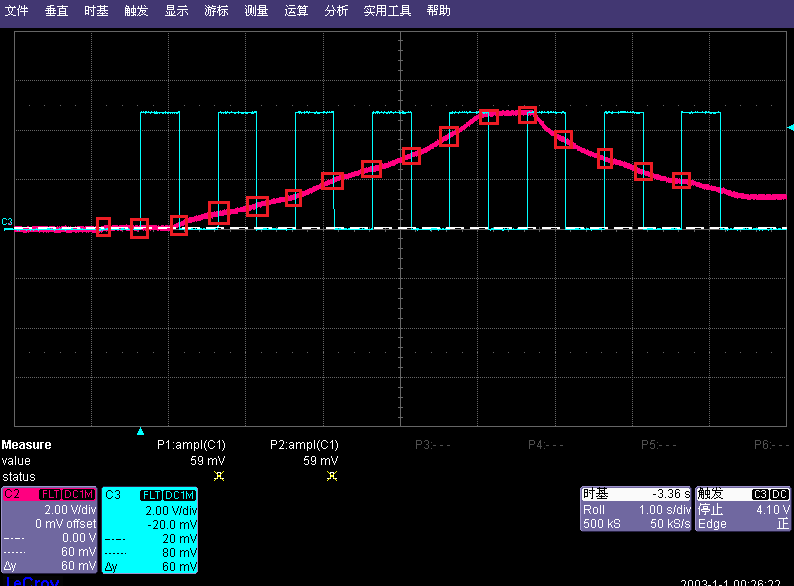
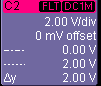
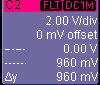


图 8‑2输入波形与



由上可知，ADC采样的结果与实际输入的模拟电压相对应，以下接口实现的功能：

1. MCU\_u16ADCinit
2. MCU\_u16ADCstart
3. MCU\_u16ADCgetResault

# PWGA模块

测试目的：验证PWGA模块的实际运行状态与配置值是否符合

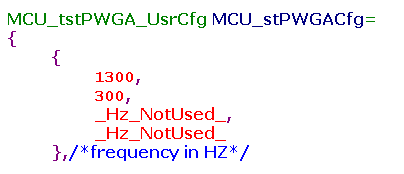
测试范围：PWGA模块的占空比设定

测试限制条件：

1. 由于PWGA的设定频率为离散的点，为了参数设定与实际的输出频率一致，驱动不提供改变设定频率的接口，各个离散的频率输出点请参见excel计算，请避免使用不在点上的频率，以免造成PWGA无法输出期望频率值
2. 占空比的比例为1000 = 100%

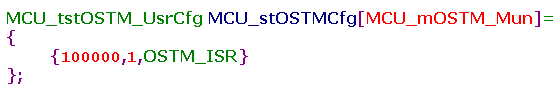
## PWGA测试准备

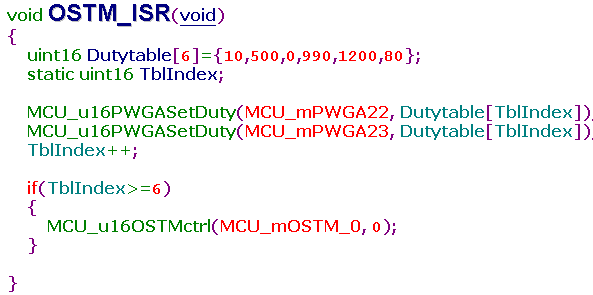
1. 设定PWGA的频率和复位时候的输出占空比



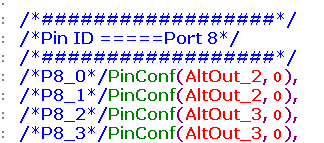


1. 设定一个定时器用于改变期望的占空比





1. 设定引脚的PWGA输出功能选择



## 测试波形

1. 复位的占空比输出为：

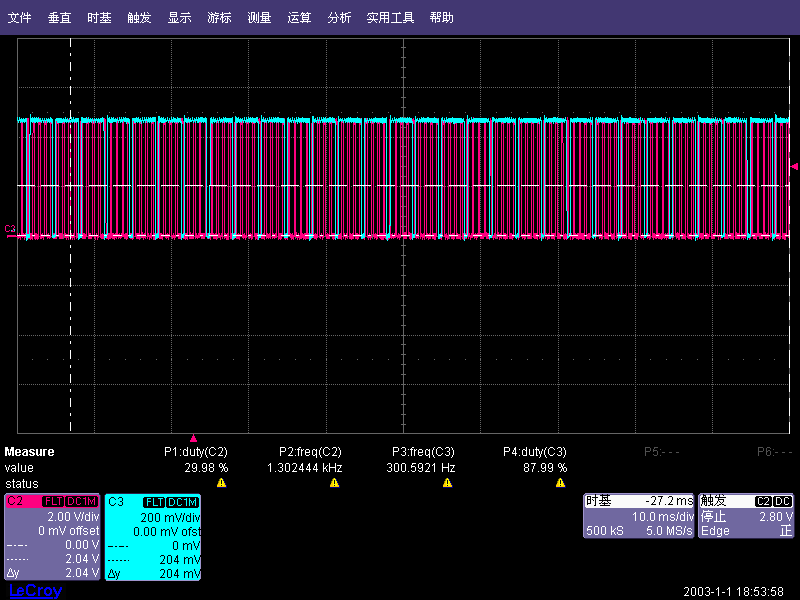
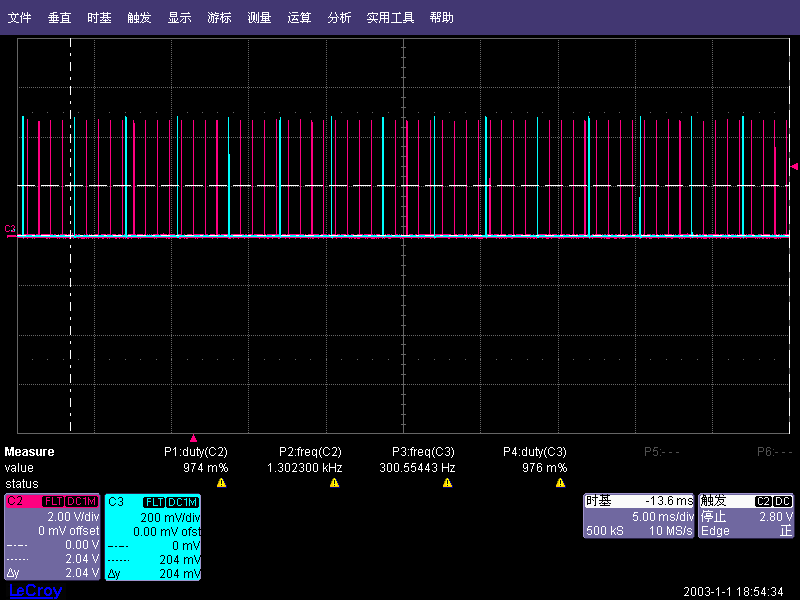
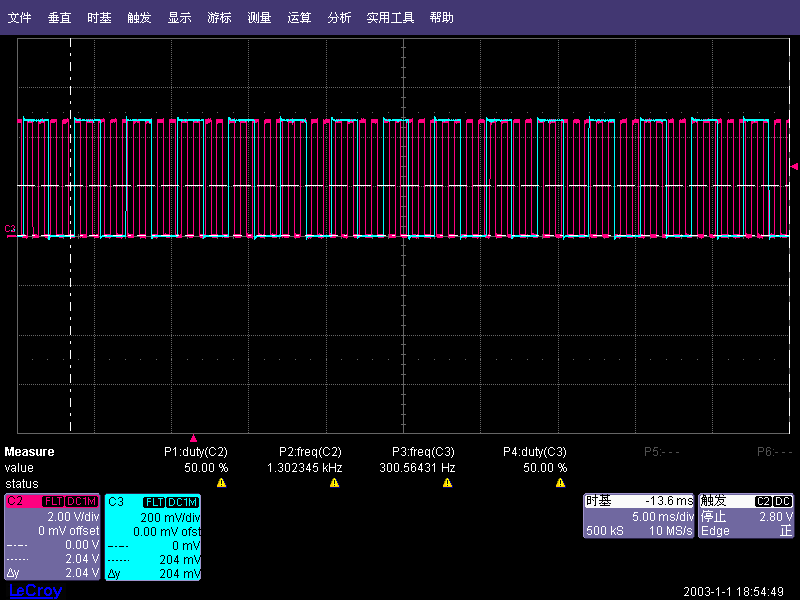
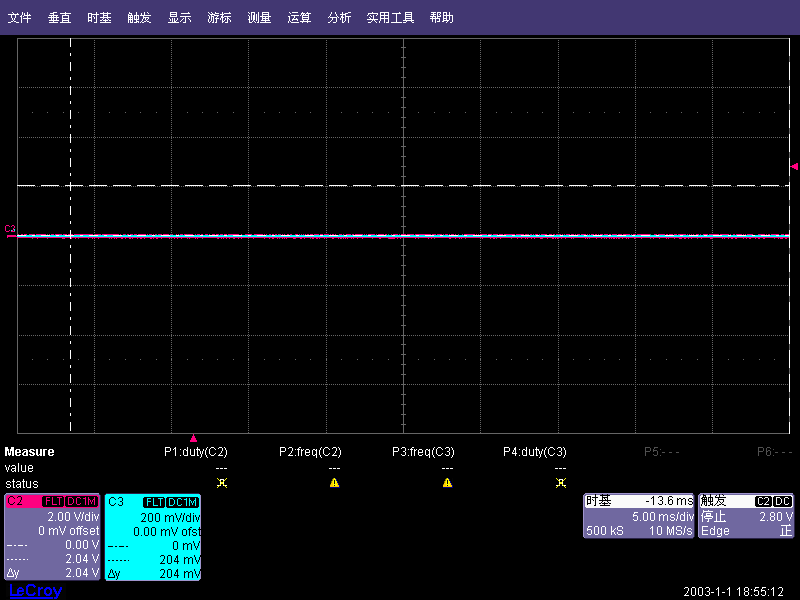


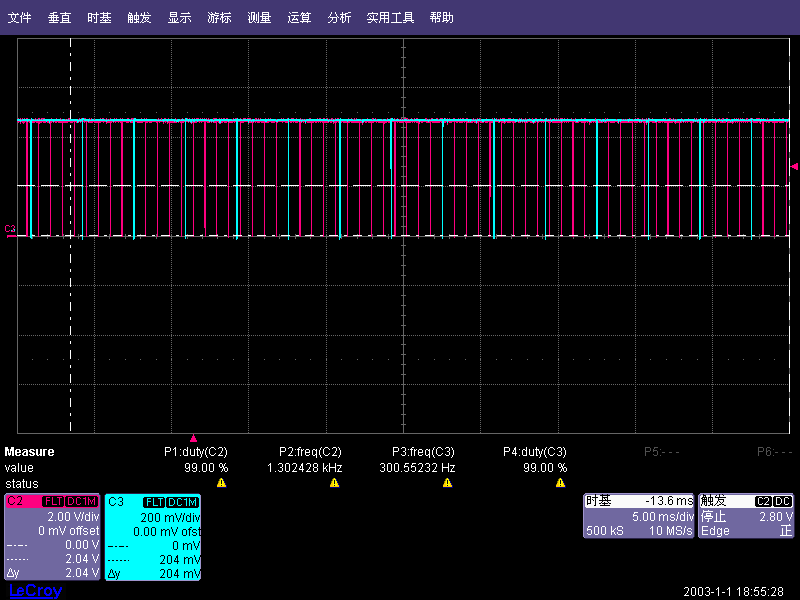
图 9‑1复位占空比

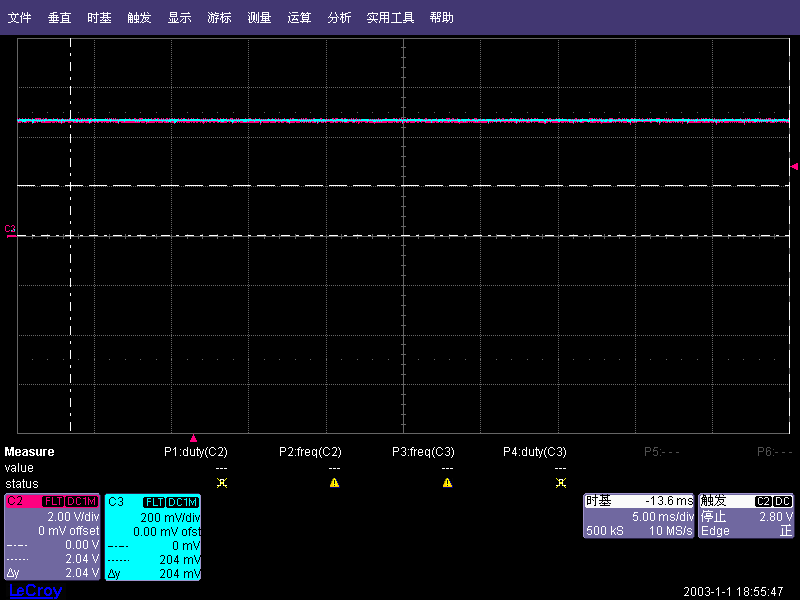
1. 在定时器内变更的占空比一次为：

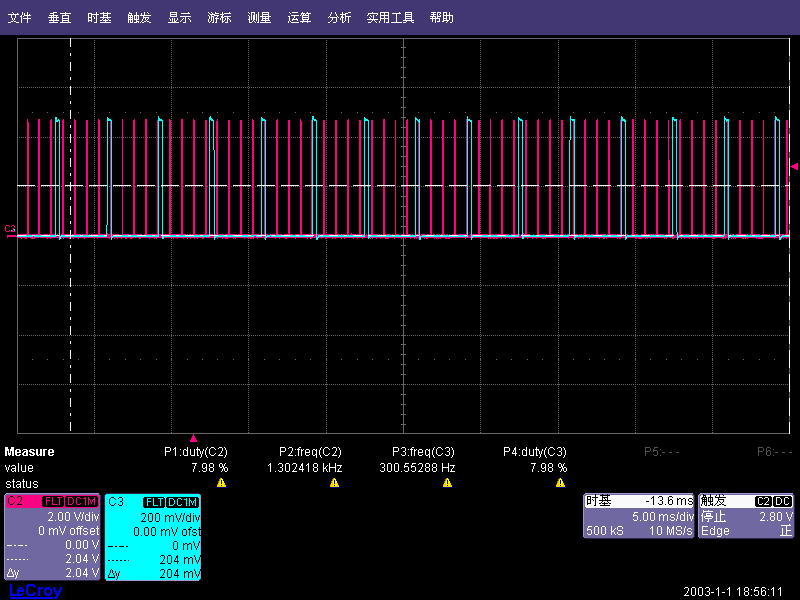












## 模块测试结果

由设定参数和实际测试的波形输出可知，以下接口函数实现了基本功能：

1. MCU\_u16PWGAinit
2. MCU\_u16PWGASetDuty

# TAUx模块

测试目的：验证TAUB,TAUD,TAUJ模块的实际运行状态与配置值是否符合

测试范围：TAUx模块的各个功能

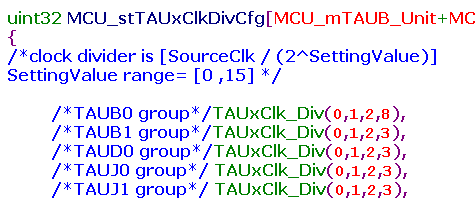
测试限制条件：

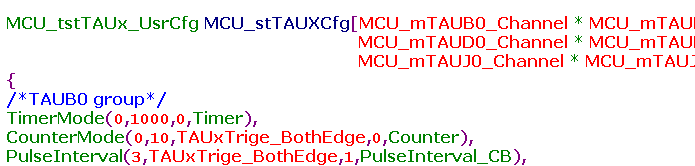
## 定时器功能、计数器功能和脉宽测量功能的验证

### 验证方法

1. 设定各个定时器的功能，TAUB0\_0位定时器，TAUB0\_1位计数器，TAUB0­\_2位脉宽测量功能，
2. 测试思路为：
   1. 定时器中断内对P11\_2进行定时取反
   2. 将P11\_2的输出波形连接到TAUB0\_1的输入引脚P10\_11上
   3. 此时计数器进行计数，在计数满以后，进入中断，在中断内对P11\_3进行取反操作，同时改变计数器个数，以判定TAUx的工作状态
   4. P11\_3波形再连接到TAUB0\_2的输入引脚P0\_8，触发TAUB0\_2的脉宽测量功能，在脉宽测量功能的中断内，检测测试值与波形是否符合

### 参数设定











### 测试结果

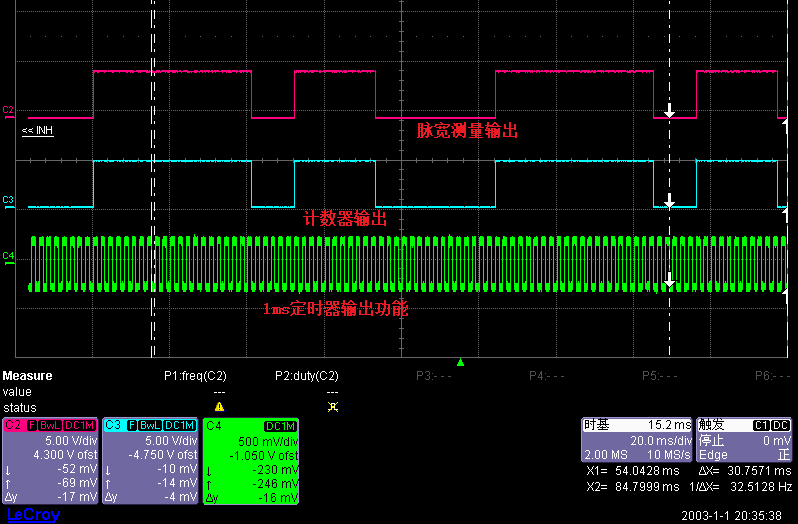


图 10‑1总体波形

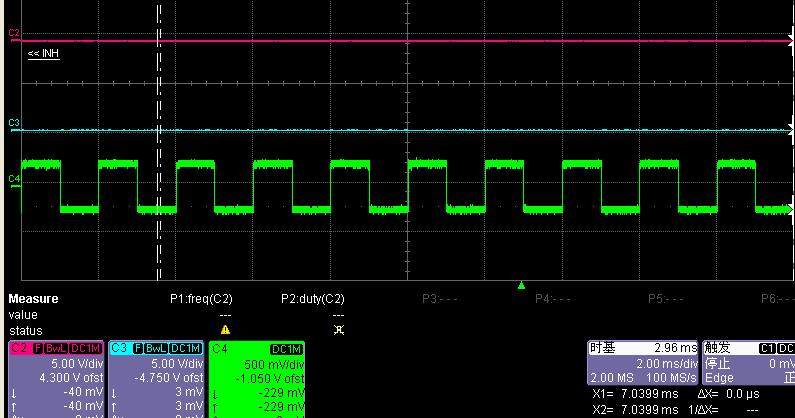


图 10‑21ms定时器波形

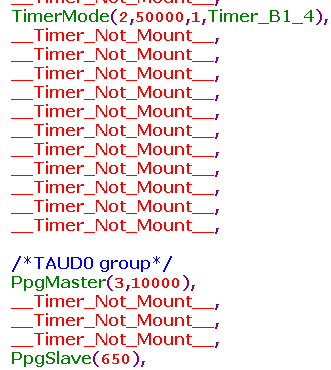
由于脉宽测量的设定是双边沿，所以其中断输出与输入时一致的，在计数器改变以后，输出波形与设定值变化一致。以下接口验证正常：

1. MCU\_vTAUxRunCtrl
2. MCU\_u32TAUxGetCounter
3. MCU\_u32TAUxGetTime
4. MCU\_u16TAUxSetCounter

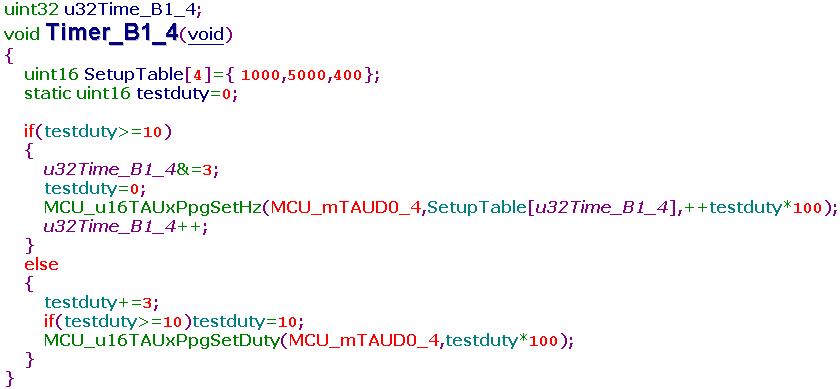
## PPG的主节点与从节点验证及PPG特征接口验证

### 验证方法

1. 设定TAUB1\_4定时器用于变更PPG属性的定时，且设定一组PPG，主节点为TAUD0\_0，从节点为TAUD0\_4（主节点只能是偶数通道），复位状态为10kHZ，占空比为65%



1. 设定定时器器中变更的参数，



1. 设定功能选择的引脚



### 测试结果

根据软件参数的设定，PPG实际输出如下：具体波形细节请参考波形文件

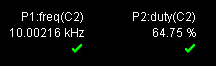


图 10‑3复位时刻









图 10‑4基于10KHZ频率的变更占空比







图 10‑5基于1KHZ频率的变更占空比









图 10‑6基于5KHZ频率的变更占空比







图 10‑7基于0.4KHZ频率的变更占空比

其中100%占空比和0%占空比由于示波器无法识别读数而没有截图

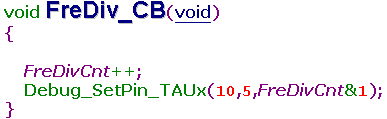
PPG功能的一下接口得到验证：

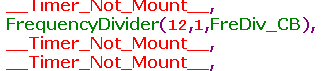
1. MCU\_u16TAUxPpgSetHz
2. MCU\_u16TAUxPpgSetDuty

## 分频器输出的验证

### 验证方法

设定分频器参数，使用TAUD0\_13进行测试，对应输入引脚为P10\_6，直接使用输入源直接使用TAUD0\_4输出的10khz的PPG信号进行分频输出测试，将P0\_1连接到P10\_6，直接使用P10\_5进行分频输出的检测，由此验证输入与输出的分频比是否为设定的值









### 测试结果

分频器的设定值与实际测量值相符合

# RLIN模块的UART 功能

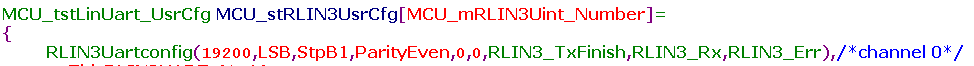
测试目的：验证RLIN3模块的UART功能实际运行状态与配置值是否符合

测试范围：RLIN3模块的UART功能

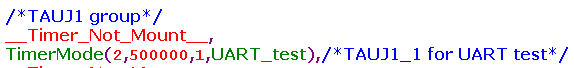
测试限制条件：

## 测试准备

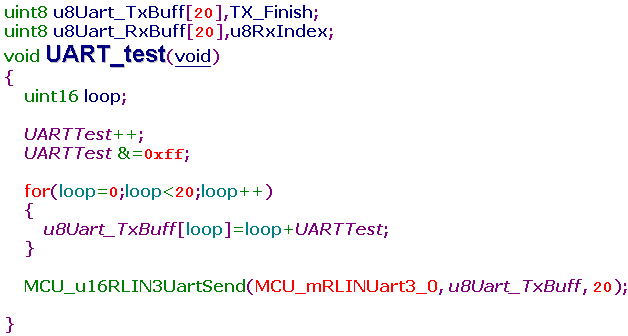
1. 设定UART功能相关的参数



1. 用一个定时器TAUJ1\_1，用于触发UART的发送指令



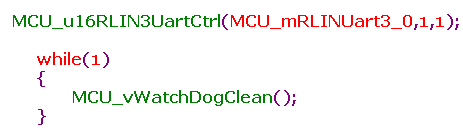
1. 在定时器中断内，进行对数据的更新和发送动作，



1. 对UART引脚功能进行配置



1. 在准备前进行UART的收发控制进行全使能



1. 在UART的引脚，RX,TX通过一个带电阻的导线连接，做自发自收的测试，观测波形与实际发送的数据，接受的数据是否一致

## 测试结果

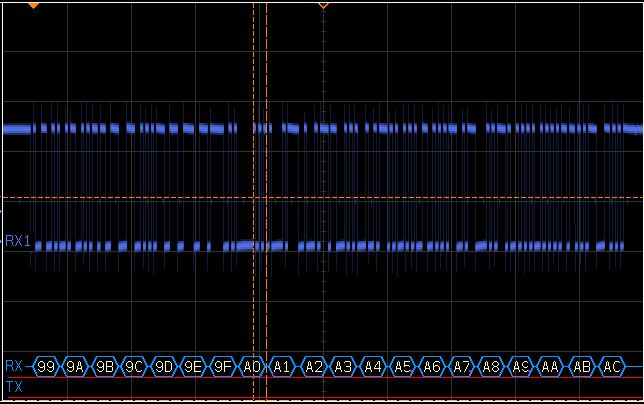


图 11‑1示波器进行解析的数据

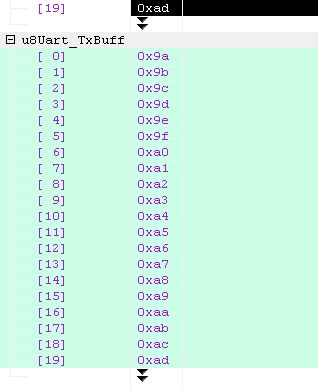


图 11‑2发送端的缓存数据

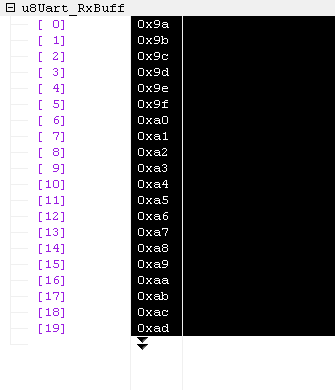


图 11‑3接收端的实际收到的数据

测试结论：

RLIN3的UART功能测试正常，收发数据正常

# IIC模块

测试目的：验证IIC模块的实际运行状态与配置值是否符合

测试范围：IIC模块的主节点工作方式

测试限制条件：从节点模式由于应用场景较少，没有进行实现，本次测试使用M24C16（EEPROM）芯片的读写操作进行IIC的通信测试。具体EEPROM的参数和控制细节请参考M24C16的芯片说明书。

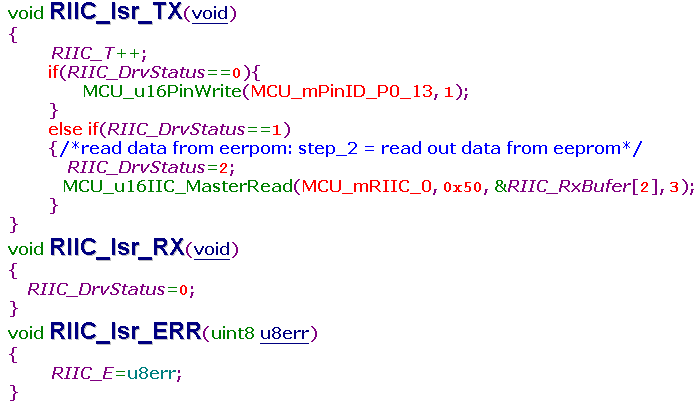
## 测试准备

1. 设定一个定时器用于定时触发IIC工作

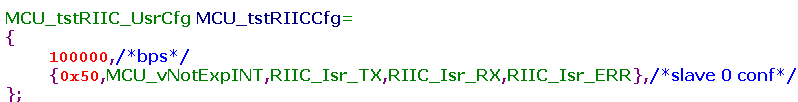


1. IIC的测试程序





1. IIC配置为100kbps，发送完成时产生中断，由EEPROM的规格书可知，此芯片的IIC地址为0x50，所以配置如下：



1. 引脚配置



1. 测试流程

向EEPROM的0x14地址写入数据，在下一次定时器到来时从该地址读出内部数据进行验证

## 测试结果

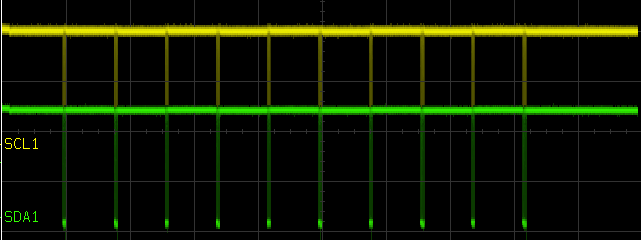


图 12‑1整体数据

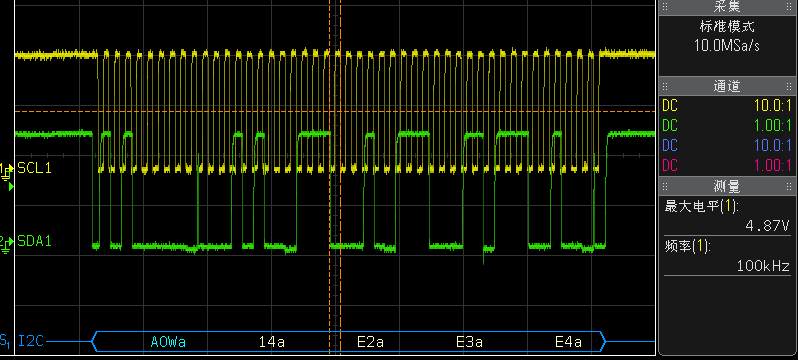


图 12‑2数据写入局部放大



图 12‑3数据读出局部放大

测试结论：

经过测试，IIC模块驱动的EEPROM可以正常工作，IIC配置参数与实际测试值相吻合

# CSIH模块

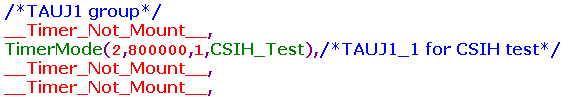
测试目的：验证IIC模块的实际运行状态与配置值是否符合

测试范围：IIC模块的主节点工作方式

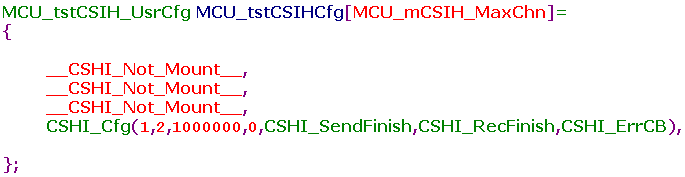
测试限制条件：由于外设的只能同时读写或者写入，没有只读的操作，所以通信的只读接口函数没有测试，具体细节请参考TLE94106手册

## 测试准备

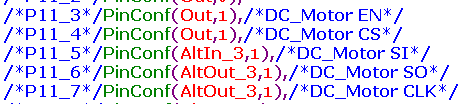
1. 设定一个定时器触发CSIH工作



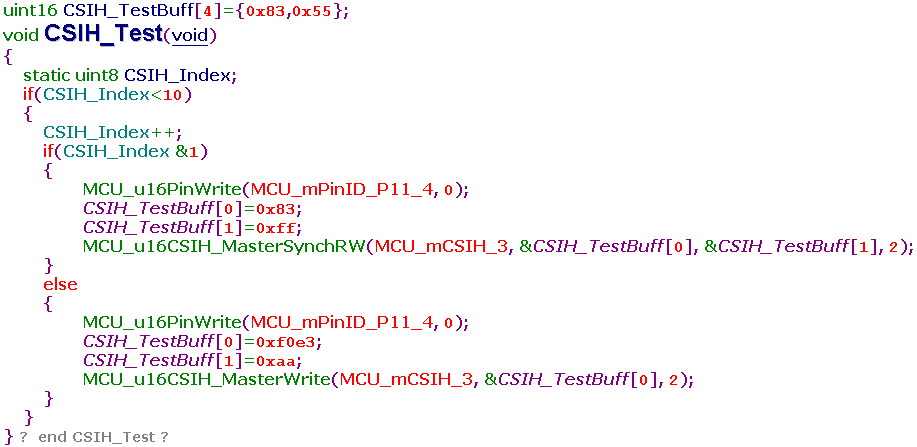
1. 配置CSIH的属性



1. 配置相应引脚的功能选择



1. 在定时器中断回调函数中设定测试流程



1. 函数执行结果

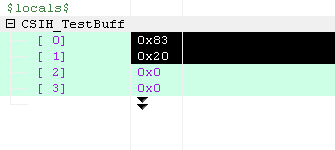


图 13‑1执行完同步读写以后的缓存结果

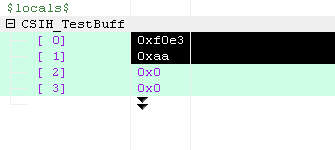


图 13‑2执行完写入以后的缓存结果

1. 实际引脚输出波形

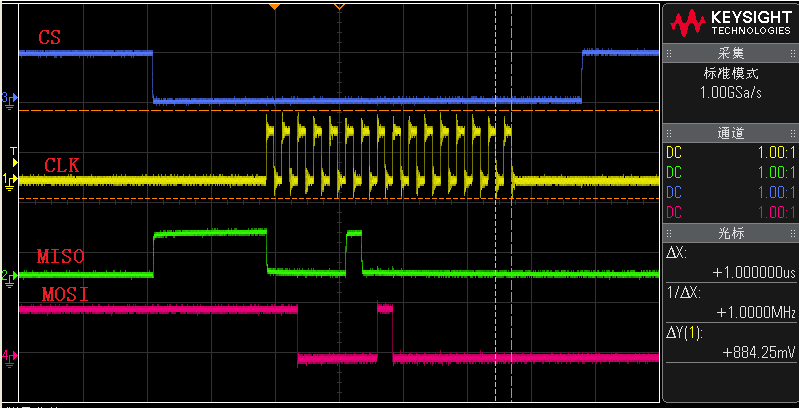


图 13‑3执行同步读写的波形

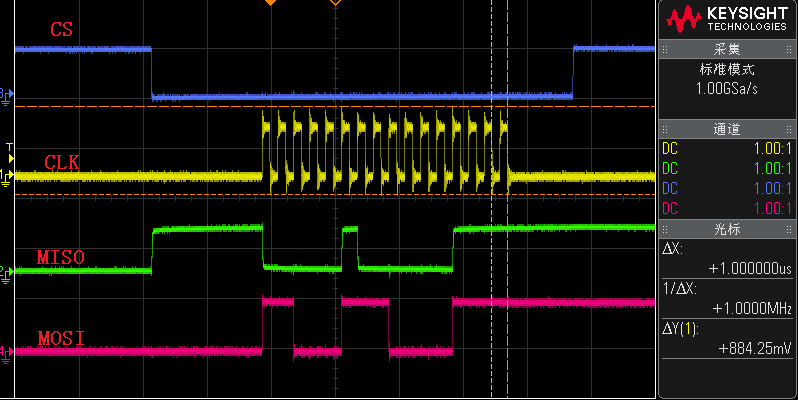


图 13‑4执行写入时的波形

## 测试结论

进过测试，以下接口函数测试正常：

1. MCU\_u16CSIH\_MasterSynchRW
2. MCU\_u16CSIH\_MasterWrite

# CSIG模块

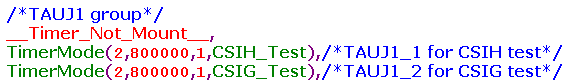
测试目的：验证IIC模块的实际运行状态与配置值是否符合

测试范围：IIC模块的主节点工作方式

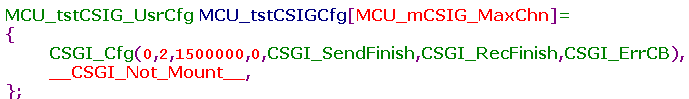
测试限制条件：由于外设的只能同时读写或者写入，没有只读的操作，所以通信的只读接口函数没有测试，具体细节请参考E520.01手册

## 测试准备

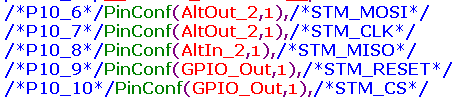
1. 设定一个定时器用于触发CSIG收发工作



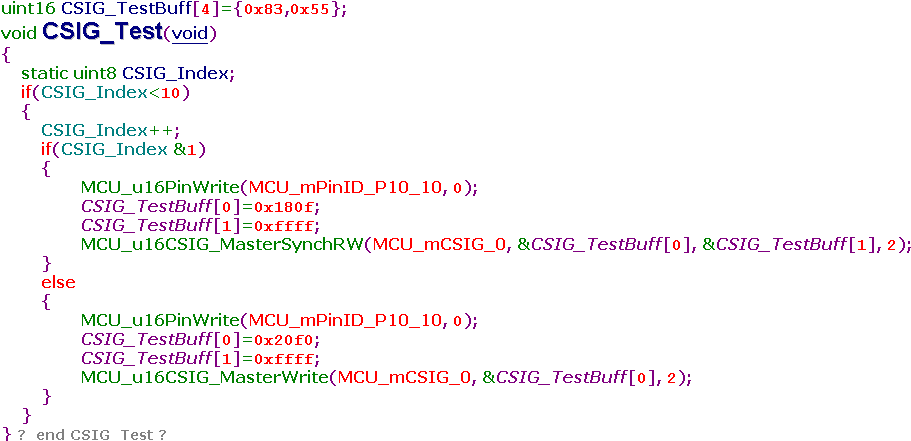
1. 根据外设手册配置CSIG属性



1. 配置引脚功能选择



1. 在定时器回调函数中设定测试流程



## 测试结果

### 缓存结果

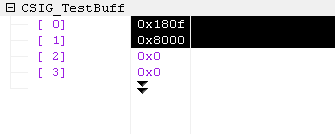


图 14‑1执行同步读写的缓存结果

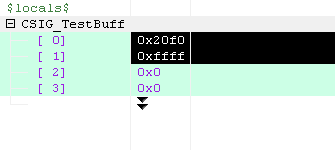


图 14‑2执行只写入数据的缓存结果

### 实际波形

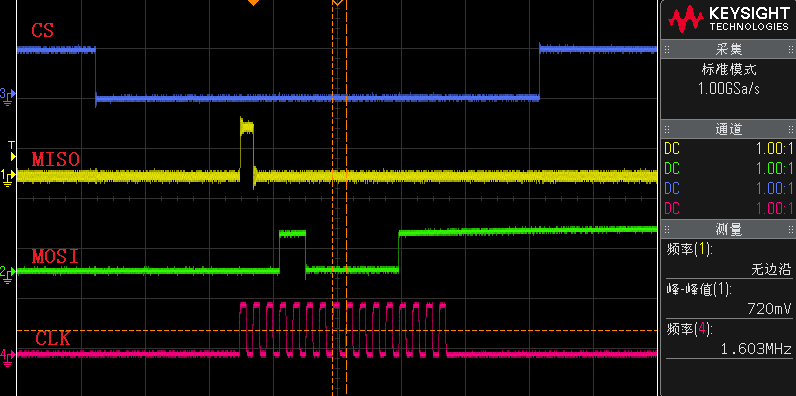


图 14‑3同步读写的测试波形

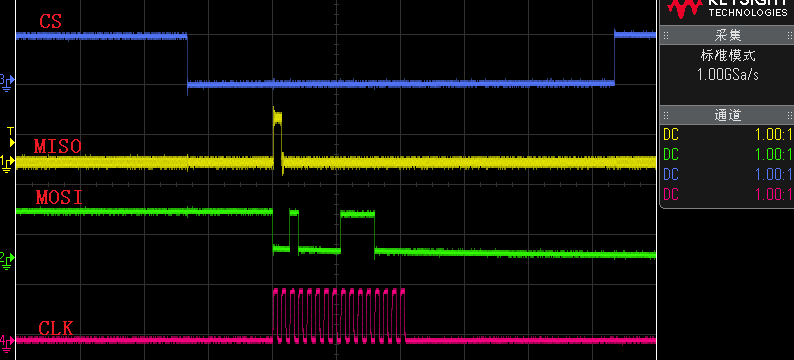


图 14‑4只写的测试波形

### 测试结论

进过测试，以下接口函数测试正常：

1. MCU\_u16CSIH\_MasterSynchRW
2. MCU\_u16CSIH\_MasterWrite

# DMA模块

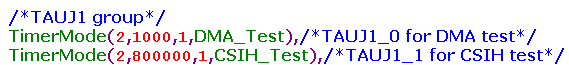
测试目的：验证IIC模块的实际运行状态与配置值是否符合

测试范围：IIC模块的主节点工作方式

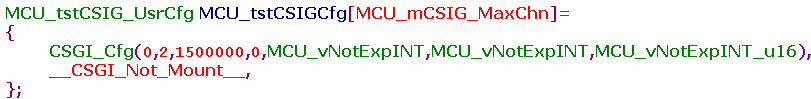
测试限制条件：DMA测试时，对于串口中断触发的情况，需要对中断顺序安排正确，在仿真情况下，可能测试与设想的不一致，那是因为DMA绕过CPU执行数据传输导致

## 测试准备

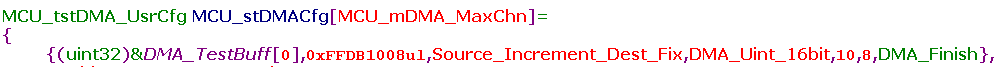
1. 设定一个定时器触发DMA工作



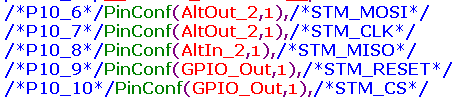
1. 配置CSIG的工作属性，不使用CSIG的中断，使用DMA中断



1. 配置DMA属性，其中源地址是需要发送的数据缓存头地址，目标地址是CSIG的发送寄存器地址，源地址发送时，地址指向递增方向，目标地址保持不变



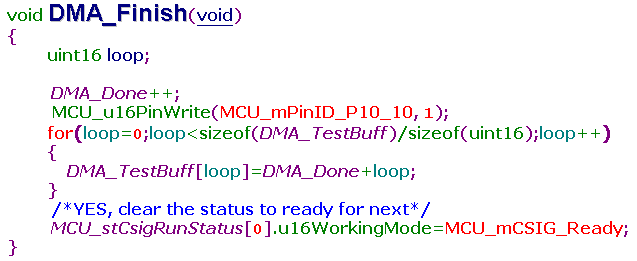
1. 配置CSIG相关的引脚功能选择



1. 在DMA发送完成中断里面设定测试流程



1. 设定发送完成函数，并且操作监控引脚



## 测试结果

### 测试波形

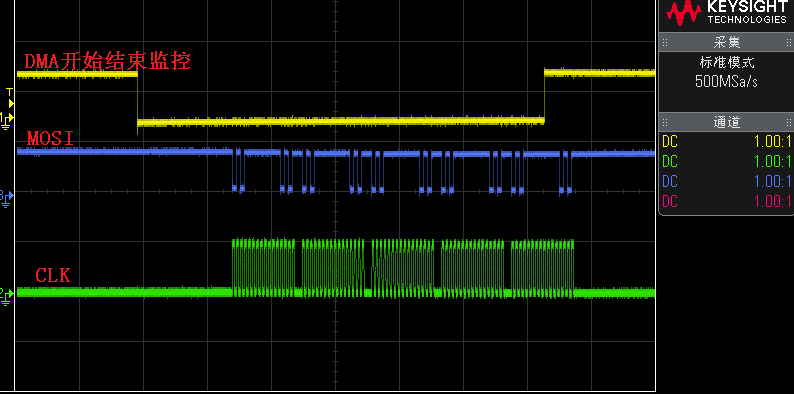


图 15‑1发送10byte时候的测量波形

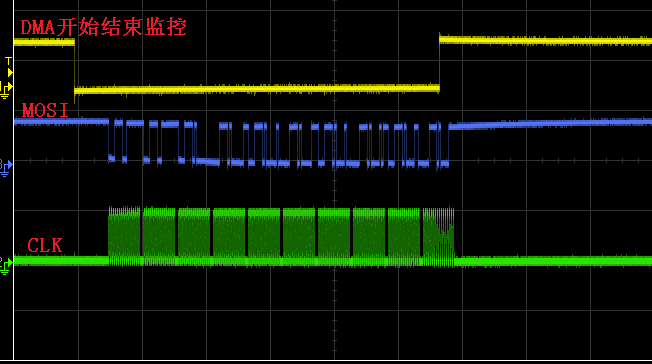


图 15‑2发送20byte时候的波形

### 测试结论

1. DMA测试符合预期，下列的接口函数功能得到实现
   1. MCU\_u16DMAsetup
   2. MCU\_u16DMADefaultSetup
2. 使用DMA和串行接口时，DMA中断是在最后一个单元数据搬运完成就产生，此时数据还处于，加监控引脚所示，最后面的数据正在发送，DMA中断就已经产生，此时不能将正在使用的串口关闭或者停止，否则最后一个单元的数据将无法发送
3. 使用DMA时，被控制的ISR不应该开启，如使用CSIG发送中断触发DMA，则CSIG的发送中断就不能被开启
4. 参考DMA的使用流程如下（串口模式）：
   1. 初始化被控制的模块，但是不使能该模块的中断
   2. 初始化DMA，开启DMA发送完成中断
   3. 通过操作使硬件中断请求发送（DMA根据设定自己开始工作，直到完成）
   4. 产生DMA中断
   5. 清除被控制模块的中断请求位（慎重），重新开启被控制模块的中断请求（被控制模块完成最后一个单元的工作，产生模块的中断）
   6. 产生模块中断，在模块的中断服务函数中做相应的事情

# 附录

## 图表

[图 1‑1芯片上电时序 4](#_Toc482209475)

[图 1‑2上电测试波形 5](#_Toc482209476)

[图 2‑1内部高速时钟波形 7](#_Toc482209477)

[图 2‑2内部低速时钟波形 7](#_Toc482209478)

[图 2‑3晶振起振波形 9](#_Toc482209479)

[图 2‑4晶振起振和稳定波形 9](#_Toc482209480)

[图 2‑5实际外部晶振波形 10](#_Toc482209481)

[图 2‑6RH850内部PLL输出 11](#_Toc482209482)

[图 2‑7内部PLL时钟波形 12](#_Toc482209483)

[图 3‑1芯片optbyte0设定 13](#_Toc482209484)

[图 3‑2看门狗0定时器监控波形 15](#_Toc482209485)

[图 3‑3看门狗1定时器监控波形 15](#_Toc482209486)

[图 3‑4复位模式下只喂WD1的测试波形 15](#_Toc482209487)

[图 3‑5复位模式下只喂WD0的测试波形 16](#_Toc482209488)

[图 4‑1OSTM验证波形 18](#_Toc482209489)

[图 6‑1P Port验证波形 23](#_Toc482209490)

[图 6‑2J\_Port验证波形 25](#_Toc482209491)

[图 6‑3A\_Pprt测试波形 27](#_Toc482209492)

[图 7‑1外部中断验证波形 29](#_Toc482209493)

[图 8‑1ADC仿真结果 31](#_Toc482209494)

[图 8‑2输入波形与 31](#_Toc482209495)

[图 9‑1复位占空比 34](#_Toc482209496)

[图 10‑1总体波形 39](#_Toc482209497)

[图 10‑21ms定时器波形 39](#_Toc482209498)

[图 10‑3复位时刻 40](#_Toc482209499)

[图 10‑4基于10KHZ频率的变更占空比 41](#_Toc482209500)

[图 10‑5基于1KHZ频率的变更占空比 41](#_Toc482209501)

[图 10‑6基于5KHZ频率的变更占空比 41](#_Toc482209502)

[图 10‑7基于0.4KHZ频率的变更占空比 42](#_Toc482209503)

[图 11‑1示波器进行解析的数据 44](#_Toc482209504)

[图 11‑2发送端的缓存数据 44](#_Toc482209505)

[图 11‑3接收端的实际收到的数据 45](#_Toc482209506)

[图 12‑1整体数据 47](#_Toc482209507)

[图 12‑2数据写入局部放大 47](#_Toc482209508)

[图 12‑3数据读出局部放大 48](#_Toc482209509)

[图 13‑1执行完同步读写以后的缓存结果 50](#_Toc482209510)

[图 13‑2执行完写入以后的缓存结果 50](#_Toc482209511)

[图 13‑3执行同步读写的波形 50](#_Toc482209512)

[图 13‑4执行写入时的波形 51](#_Toc482209513)

[图 14‑1执行同步读写的缓存结果 53](#_Toc482209514)

[图 14‑2执行只写入数据的缓存结果 53](#_Toc482209515)

[图 14‑3同步读写的测试波形 53](#_Toc482209516)

[图 14‑4只写的测试波形 54](#_Toc482209517)

[图 15‑1发送10byte时候的测量波形 56](#_Toc482209518)

[图 15‑2发送20byte时候的波形 57](#_Toc482209519)