

基于LGTSDK Builder

## LGT8F690A 快速开发系列教程

---

### 第六篇: I2C接口的使用



本篇为系列教程的第六篇。如果需要了解教程相关的软件硬件环境，请参考本系列教程的第一篇：《LGT8F690A快速开发系列教程第一篇\_急速上手》



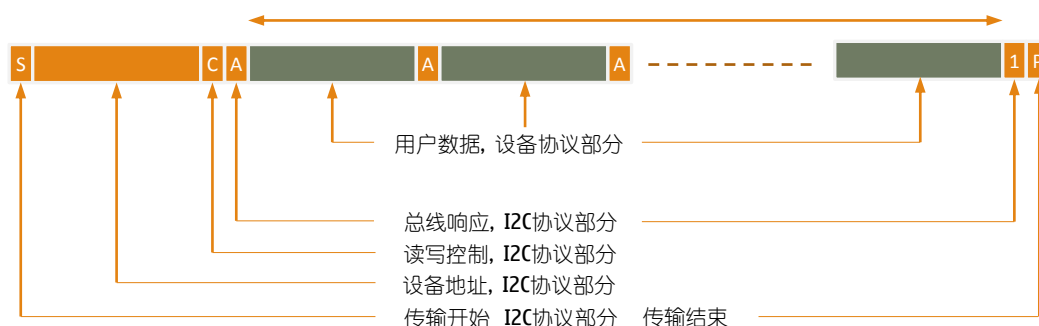
本篇教程只适用于LGTSDK Builder 1.5beta23或者之后的更新版本。

LGT8F690A内部集成一个兼容I2C协议的硬件控制器。可以实现I2C主从设备通讯。控制器内部实现了一个可编程的传输速率控制器，当系统主频为16MHz时，可实现1.6Mbps的最高传输速率。

本篇教程主要是介绍如何使用LGT8F690A的内置I2C控制器实现与其他I2C设备的通讯，将不会详细介绍I2C协议本身。如果需要了解I2C传输协议的更新细节，请参考其他资料。

I2C接口具有非常标准的物理定义和传输时序定义，因此，除非我们需要用软件模拟I2C总线，其他情况下，我们只需要知道I2C传输的数据帧格式即可，不需要了解更底层的细节。

下面是一个比较常见的I2C数据帧：



一个完整的I2C数据帧，包含设备地址，设备数据以及几种传输控制字。其中传输控制部分由I2C控制器根据传输类型的定义自动产生。传输开始后，第一个字节包含了设备地址和读写控制字。这也是I2C协议定义的部分。之后的数据部分，数据的含义以及数据的长度，不属于I2C协议的定义，这部分由实现I2C接口的设备定义。

LGT8F690A控制器实现了以上I2C协议部分。传输控制部分需要在软件的指导下完成。

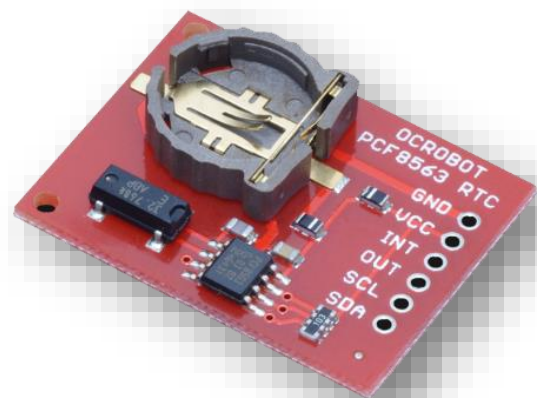
对于主从设备，控制器和应用软件在处理数据帧上有比较大的区别。主设备具有绝对的控制全，启动传输，确定传输类型以及控制传输流程。从设备属于被动设备，只有在收到主设备的控制指令后，才能根据控制指令做出相应的回应。

MCU的应用中，I2C作为主设备访问I2C外设的应用是最常见的。因此我们这里也是先介绍LGT8F690A作为I2C主设备的使用方法。

I2C的外设非常广泛，这里我们选择的测试设备是一个I2C总线的时钟芯片：PCF8563。这是一款常用的时钟芯片，无论是芯片还是模块，都非常容易获得。我这里的PCF8563来自于OCROBOT，在这里也表达对OCROBOT的感谢！

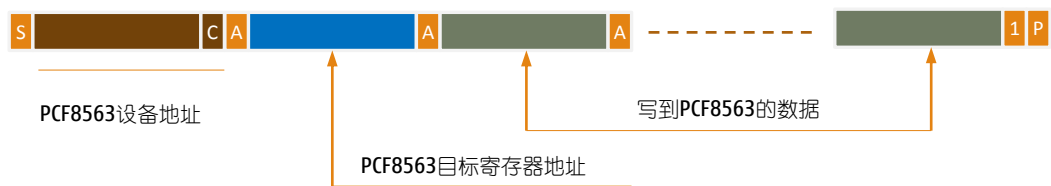
外设与LGT8F690A的连接也非常简单，只需要连接下面4根线即可：

PCF8563	LGT8F690A
VCC	VCC
GND	GND
SCL	RA1
SDA	RA0

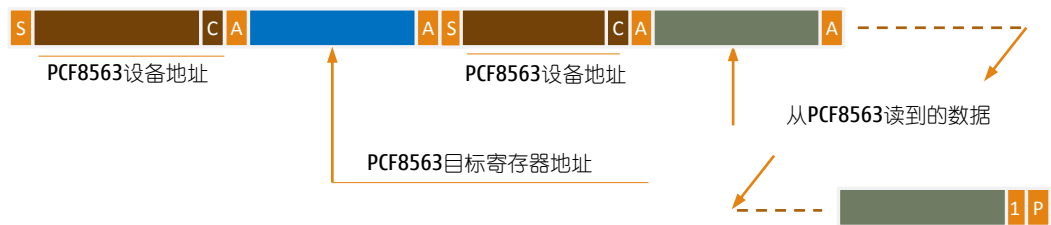


PCF8563是一个标准接口的I2C从设备。支持400K的高速I2C传输速率。除了具有日历功能外，还具有闹钟以及可编程频率输出等。具体请参考PCF8563的数据手册。

PCF8563具有固定的读写地址：读地址(0xA2)，写地址(0xA3)。其中最低位是读写控制位。下面是PCF8563的读写数据帧。这个帧定义与之前我们的I2C协议帧完全一致。不同的是，这里给出了设备协议部分的定义：



PCF8563主机写操作帧结构



PCF8563主机读操作帧结构

从读操作的帧结构可以看到，PCF8563的读操作，实际上由两个I2C协议子帧构成。这两个子帧之间并没有传输停止的控制字，这在I2C协议上也是允许的。

PCF8563读操作的两个I2C子帧中，第一个是设置读操作的起始地址。然后下面一个帧才开始顺序的读出从起始地址开始后的数据。

PCF8563的读写操作都是先设定起始地址，然后连续的发送读写即可。读写的地址是自动累积的。因此PCF8563建议一次读出所有内部寄存器的数据。我们也将使用这种方法。

PCF8563的内部的数据比较多，包括了日历和其他时钟相关的状态和设置。这里，我们仅仅是实例I2C的接口的使用。因此，我们仅仅是读出内部的数据，显示一个时钟。

外设和协议介绍完毕。需要了解详细信息，请参考数据手册。

下面，我们就以LGTSDK Builder为例，示例如何快速的实现对PCF8563的数据访问。

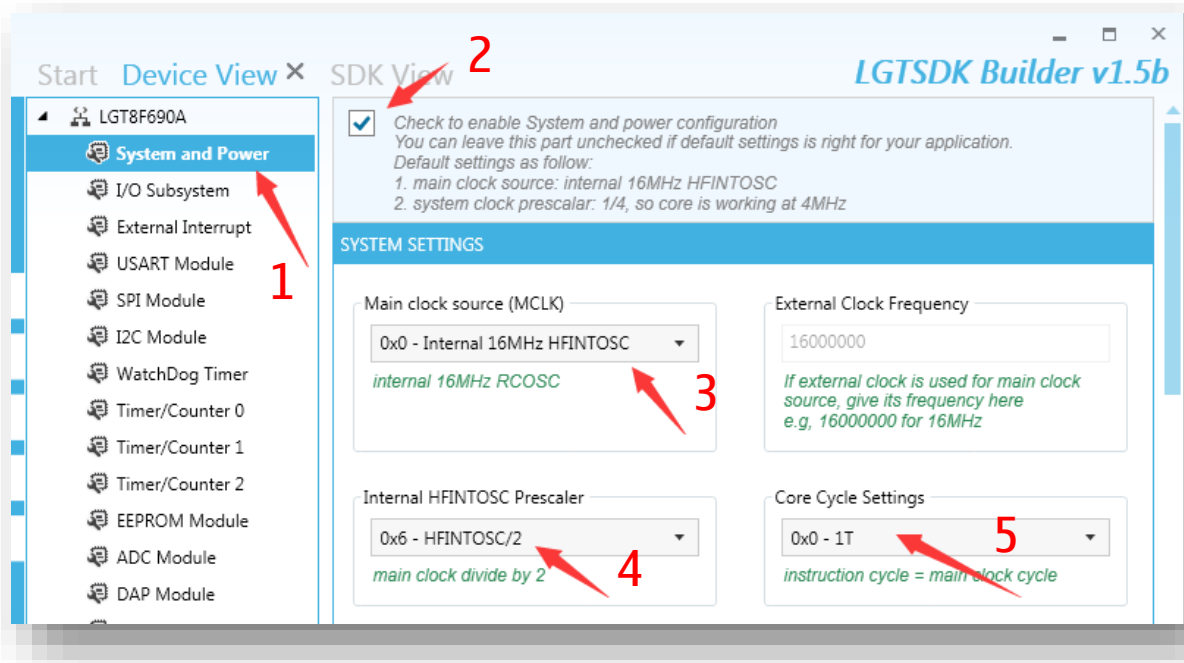
启动LGTSDK Builder，注意要使用 1.5beta23或者以上的版本。

新建一个工程，选择目标芯片：LGT8F690A，然后输入项目名称：lgt8f690a\_i2c

这里先说明下我们将要实现的功能：

我们使用I2C接口，从PCF8563中读出时间信息，然后转换格式化，然后通过串口送到PC上显示！I2C的速率选择400Kbps，USART的波特率选择19200bps。时间每秒钟更新一次，来得及！

首先，因为我们要用到串口，还有实现I2C的速率，所以我们要先配置系统时钟：



## I2C控制器的使用

简单说明下系统时钟的配置：

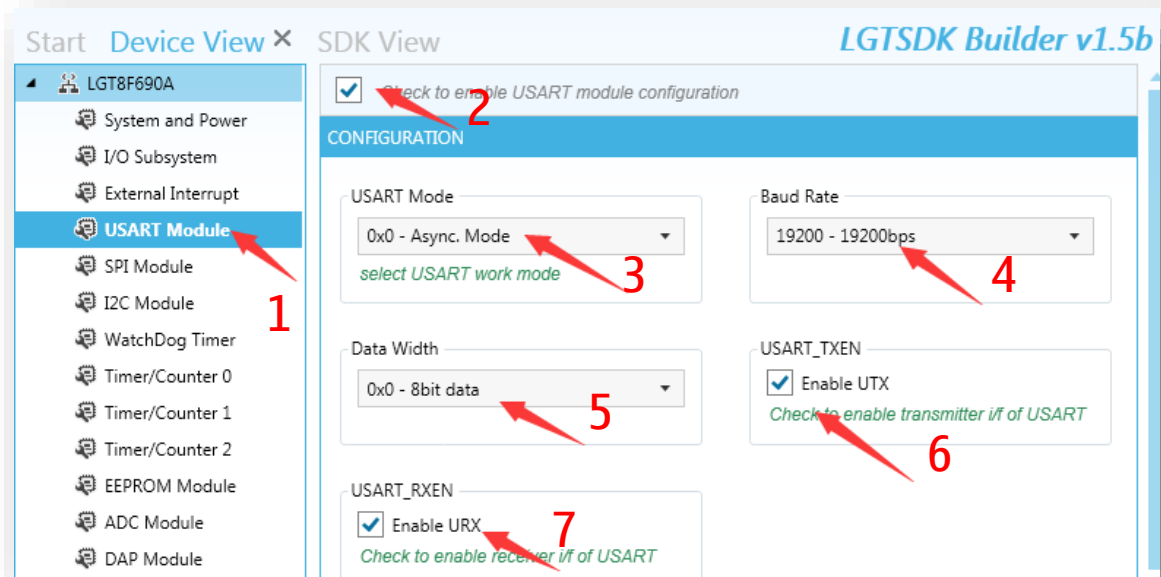
3：我们将使用**16MHz** 的内部**RC**振荡器

4：系统预分频是**2**分频。这样我们最终的系统时钟就是**8MHz**

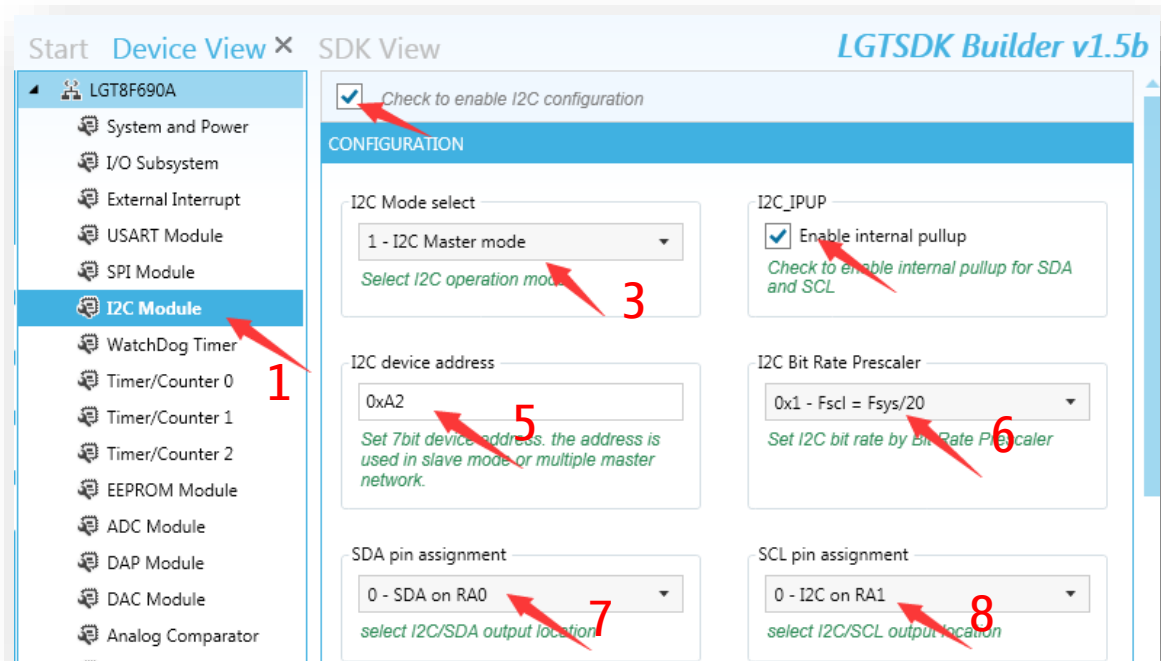
5：因为有计算，我们这里用**1T**模式，让内核跑的更快

系统时钟我们使用的是**8MHz**，这也将是**I2C**控制器的工作时钟。要产生**400K**的传输速率，只要**20**分频就可以了。

下面是串口的配置，这个我们在之前的教程中已经有个介绍，这里只给出配置结果：



最后是配置**I2C**控制了：



I2C配置说明：

- 3：如之前所示，我们选择使用I2C主机模式， PCF8563为从机设备；
- 4：使能I2C接口SDA/SCL引脚的内部上拉电阻，也就是RA0/1的内部上拉电阻；
- 5：设备地址，这里我们给出读地址即可；
- 6：I2C时钟分频器，我们的主时钟是8MHz, 这里选择20分频，产生400K的I2C传输时钟；
- 7：选择I2C总线SDA的引脚分配，这里我们用RA0；
- 8：选择I2C总线SCL的引脚分配，这里我们用RA1；

外设配置到此结束！ 点击[ Device View ] 视图下的 [Build] 按钮， 产生SDK代码。

下面开始写代码了！串口相关的接口函数再之前的教程中已经列出过。这里我们介绍下即将用到的I2C相关的接口函数。同样，你在代码编辑器里面， 输入i2c，也可以在日志窗口中看到所有I2C接口相关的函数。

函数名称	功能说明	使用方法
i2cBeginTransmission()	开始一次I2C总线传输，可以在参数里指定读写控制(I2C_READ/WRITE)	i2cBeginTransmission(I2C_WRITE) 启动I2C写传输
i2cEndTransmission()	结束I2C总线传输，发出STOP	i2cEndTransmission()
i2cSetDevAddress()	设置I2C设备地址。 我们在配置时也可以指定地址	i2cSetDevAddress(0x2A)
i2cWriteBuffer()	向I2C总线上发送数据，可指定数据长度。	i2cWriteBuffer(pBuffer, 16) 发送16个字节到I2C总线
i2cReadBuffer()	从I2C总线读数据。也可以指定读数据的长度	i2cReadBuffer(pBuffer, 16)
i2cWriteByte()	向I2C总线写一个字节	i2cWriteByte(0x55)
i2cReadByte()	从I2C总线读一个字节	Value = i2cReadByte()

下面简单介绍下程序部分要做的工作：

- 1. 参照PCF8563的读数据帧，使用I2C接口函数，组合一个PCF8563的读数据接口函数；
- 2. 由于PCF8563的数据是BCD编码， 我们需要一个转码到DEC格式，用于打印显示；
- 3. 把读到的时分秒数据，格式化一下，通过串口发送到PC

就是这些事情！下面是相关的代码：

## SDK View X

## main.c X

```

10  u8 i2cBuffer[16];
11
12  u8 hour;
13  u8 minute;
14  u8 second;
15
16  u8 bcdToDec(u8 value)
17  {
18      u8 tmp = axu_fdiv16d8(value, 16);
19      tmp = axu_fmul8x8(tmp, 10);
20      axu_fdiv16d8(value, 16);
21      return (tmp + MDXF);
22  }
23
24  void pcf8563_getDataTime(void)
25  {
26      i2cBeginTransmission(I2C_WRITE);
27      i2cWriteByte(0x00);
28      i2cEndTransmission();
29
30      i2cBeginTransmission(I2C_READ);
31      i2cReadBuffer(i2cBuffer, 16);
32      i2cEndTransmission();
33
34      second = bcdToDec(i2cBuffer[2] & 0x7f);
35      minute = bcdToDec(i2cBuffer[3] & 0x7f);
36      hour = bcdToDec(i2cBuffer[4] & 0x3f);
37  }
38

```

存放时，分，秒的变量

BCD转码DEC的接口函数

使用硬件乘除器实现的BCD码转DEC码

读PCF8563数据的接口函数

先写读操作的目标地址，从0x00地址开始

然后连续读16个字节的寄存器数据

时分秒数据依次转码处理

## SDK View X

## main.c X

```

39 void pcf8563_printTime() ← 串口打印从PCF8563读到的时间信息
40 {
41     u8 btmp;
42     u8 cTmp[11];
43
44     btmp = axu_fdiv16d8(hour, 10);
45     cTmp[0] = '0' + btmp;
46     cTmp[1] = '0' + MDXF;
47     cTmp[2] = ':';
48
49     btmp = axu_fdiv16d8(minute, 10);
50     cTmp[3] = '0' + btmp;
51     cTmp[4] = '0' + MDXF;
52     cTmp[5] = ':';
53
54     btmp = axu_fdiv16d8(second, 10);
55     cTmp[6] = '0' + btmp;
56     cTmp[7] = '0' + MDXF;
57     cTmp[8] = '\r';
58     cTmp[9] = '\n';
59     cTmp[10] = '\0';
60
61     usartWriteString(cTmp); ← 通过串口打印!
62 }
63
64 int main(void)
65 {
66     u8 now = 0;
67
68     // Device initialization
69     init_modules();
70
71     // Add your code from here
72     while(1)
73     {
74         pcf8563_getDataTime(); ← 从PCF8563读数据
75
76         if(now != second)
77         {
78             now = second;
79             pcf8563_printTime(); ← 如果时间更新, 打印到串口!
80         }
81     }

```



大功告成！！ 开始编译工程。修改笔误。。。编译成功，产生HEX，下载到最小板测试！！

用串口连接最小板。PC上启动串口助手，或者其他串口通讯工具，可以看到如下输出：

