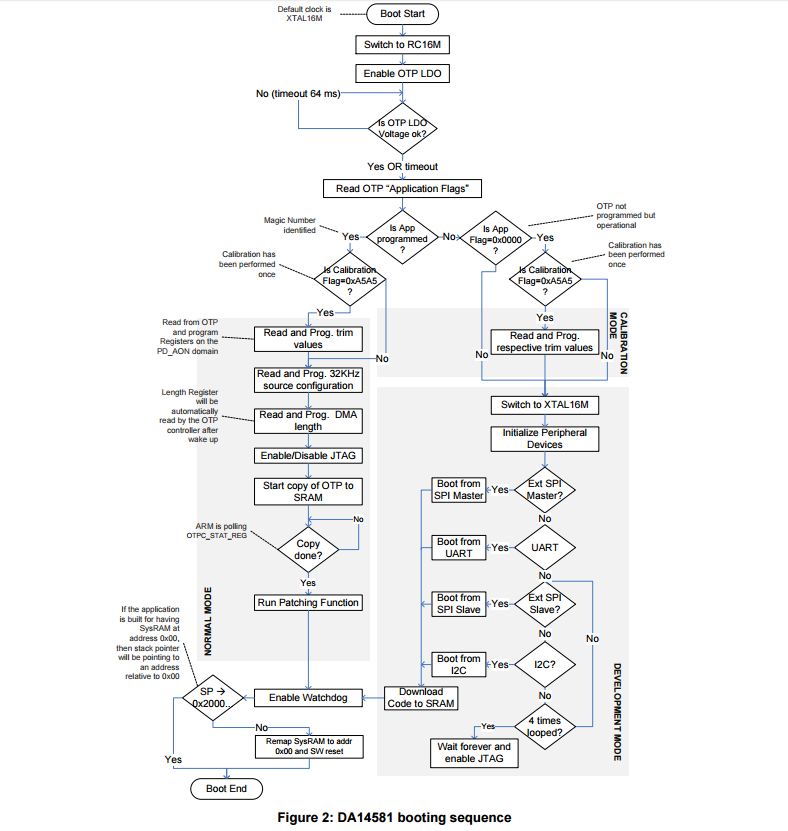
2016.12.20

Code下载有几个模式

1. JLINK直接下到SRAM
2. 外设连接到EEPROM、FLASH，读入Code。不过好像要通过DIALOG的那个软件smartSnippets\_Studio给ROM写入BOOT LOADER先。
3. 写到OTP，同时外设的FLASH模式没用了？？

启动方式



BOOT START（RST复位？？？）后进入引导流程，读取OTP上特定地址的FLAG数据，判断‘’Application Flags‘’，从而决定SRAM里代码的写入源；1、由OTP写入，但是一但决定这个OTP引导方式，便不能更改，因为FLAG在OTP，写了不能改了。2、启动外设接口，按顺序检测外部主SPI、UART、外部从SPI和I2C，寻找引导设备，并由特定的协议，读入Code，下载到SRAM。

2016.12.21

电源总线3V电压用开关电源得来（TPS62736，BUCK），不使用LDO，LDO效率不够，起码锂电池3.7V到3V的压降太大了。

串口转USB好像没什么必要。

IIC需要上拉，功耗增加。

I2C总线空闲的时候，两条信号线应该维持高电平。否则，上拉电阻上会有耗电。特别是在上电过程中，IO线上电平也应保持在高电平状态。也就是说：当Master的I2C使用的是IO软件模拟时，一定要保证该两个IO上电默认均为输入（或高阻）或者输出高电平，切不可默认为输出低电平。IO默认为输入时，可以通过外部上拉电阻将I2C信号线拉至高电平。

2016.12.27

使用DIALOG的工具进行串口下载的时候，会要求用户进行手动复位，这就说的通了。

蜂鸣器用的DET402-G-1

2016.12.28

打算换四层板，好走线一点，求稳。

今天向DIALOG申请到了SDK，有点复杂，先开始了解一下结构。

GAP: Generic Access Profile 通用存取Profile。跟蓝牙的advertising相关

GTL: Generic Transport Layer 通用传输层。当工作于外部主控模式时，用来传输主控到DA14580的数据

HCI : Host Controller Interface 主控接口。当工作于外部主控模式时，DA14580提供给主控的通讯接口

NVDS: Non-Volatile Data Storage 非易失性数据存储器。在DA14580中用来保存蓝牙地址和一些配置信息

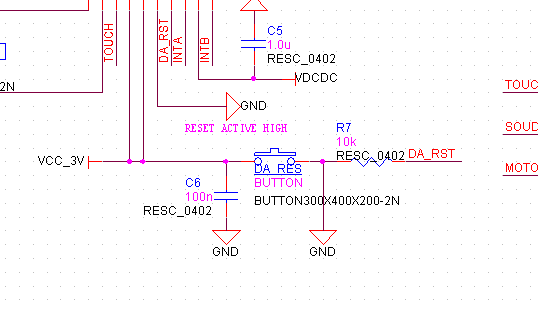
SPotA: Software Patching over the Air 固件空中升级

2016.12.29

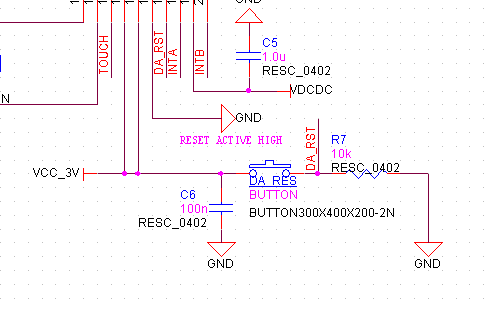
文档AN-B-020\_v1.3\_End product testing and programming guidelines.pdf有提到各种下载方式。

2016.12.31

V220的pcb文件有个错误



RST复位电路接错了。V220已经修正。



2017.1.2

Pcb文件完成了，等待光绘，用cam350进一步检查，以及原理图检查。

2017.1.3

Tps62736和16MHZ 2520的封装错了，以后要注意top view和bottom view。

2017.1.7

PROJECT OPTION的DEVICE一定要指定具体的设备。

在linker里面吧Use Memory Layout from Target Dialog去掉勾，下面.sct选dialog的文件，就可以正常在keil debug。奇怪。

Dialog的文件内容:.sct

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; \*\*\* Scatter-Loading Description File generated by uVision \*\*\*

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

LR\_IROM1 0x20000000 0x00008000 { ; load region size\_region

ER\_IROM1 0x20000000 0x00008000 { ; load address = execution address

\*.o (RESET, +First)

\*(InRoot$$Sections)

.ANY (+RO)

}

RW\_IRAM1 0x20008000 0x00008000 { ; RW data

.ANY (+RW +ZI)

}

}

问题解决，.sct文件是keil自己生成，也就是根据device的选择生成，说到底还是keil要有da14580的支持包。另外还需要将project option->Linker->R/W Base 改为0x20000000，这个还需要多测试。

2017.1.19

下一板led驱动方式改。

USB的VBUS和GND短路了，初步认定是usb下的机械孔内电层POWER SOURCE和GND被孔打穿短路了，通了40A也仅仅是把焊盘烧了，所以短路接触面是相当大的。下一板要绕开机械孔，除了机械孔我也想不到哪里还有可能短路了。

UART console 初步用的一些指令，没加入校验，有空再说：

c0 01 c0 redon

c0 02 c0 redoff

c0 03 c0 greenon

c0 04 c0 greenoff

c0 05 c0 oledon

c0 06 c0 oledoff

OLED SDA SCL搞反了…已修正，以后注意

2017.1.25

DA14583不支持Deep Sleep mode 和 boost模式。

Read tutorial:

Tutorial 1,Modifying Advertising Parameters

BLE has two primary modes of operation: 1-a connected mode which is defined by Profiles,2-an unconnected broadcast mode.

An advert defines the format of the data transmitted in this broadcast mode.

Tutorial 2,Adding Characteristics – Custom Profile Creation

The server -> owner of the data and in most cases is the peripheral device.

The client -> consumer of the data and is typically the central device like smart phone.

/da1458x\_config\_advanced.h

/user\_config\_sw\_ver.h

/user\_modules\_config.h

/user\_callback\_config.h

/user\_config.h

/user\_profiles\_config.h

/user\_custs\_config.h

/user\_periph\_setup.h

/user\_periph\_setup.c

BLE application includes a BLE profile, a BLE profile can have one and more services, a service can have one or more characteristics. A characteristics is the lowest level concept in GATT transactions.

Tutorial 4,Building Custom Profiles

2017.1.28

Service : custom service 1

Characteristics:

BBPG CONTROL->{

Property: Write

len: 32

uuid:

name: BBPG CONTROL

contain: led alarm, souder alarm, motor alarm

BBPG LOSS CHECK->{

Property: Write, read,notify

Len: 32

Uuid:

Namd:BBPG LOSS CHECK

BBPG BREAK CHECK->{

Property: Notify

Len: 32

Uuid:

Name:BBPG BREAK CHECK

TIMER0的ON寄存器是控制中断的周期，M和N是控制PWM的周期

app\_easy\_timer() 最多注册10个回调。

2017.1.30

OLED REMAP设置和写的方向有关。

字节的低位在页PAGE的上方：

0x01

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |

OLED滚动显示字符串完成，但是连续的功能没搞。

asdasdasdABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789:

C里面不同类型比较是会出问题的，例如刚刚两一样值的字符比 一个unsigned一个signed，被判断为不等。

指令约定：## +4 bytes timestamp + 6 bytes command + 1 byte [( ts + a + b + c + d + e +f)%26 +97] + %%

6bytes command : 1byte command type + 1byte command parameter+ 4bytes resvered

command type:{identify, alarm enable}

motor直接用gpio驱动，放弃脑残一样的pwm0

电池检测周期 10s

2017.2.3

由于防拆中断和防剪中断选用的奇葩io口，在开机初始化从flash读code时，表带不能查着。

2017.2.4

睡眠检测->降低发射频率->比如以后要监控犯人身体监控，要周期性发送数据

2017．2．9

今天遇到了一个bug，关于破拆报警，这个timer跑一段时间后会抛出RESET\_MEM\_ALLOC\_FAIL的异常，后来发现没连接的时候才会异常，后来req的创建（KE\_MSG\_ALLOC\_DYN）也放进是否连接的判断里，保证req创建了一定能被消化掉。也就是每次send message都需要alloc，而且alloc后一定要send不能不管，不然出事。

WAKE UP的时候断开连接没警示