龙芯1b0200 IIC分析

(IIC 的相关简介 感觉之前好像有个东西给过了)

这个是IIC的介绍

https://blog.csdn.net/ZahirGong/article/details/78864226

【对了还有OLED的相关介绍应该也是要放在这里的 SSD1306】

这个是我找到的关于OLED的介绍，但是其实是论坛里的问答：

https://arduino.stackexchange.com/questions/74725/finding-address-of-oled-using-i2c-communication-and-arduino

龙芯1b系列芯片具备三路IIC接口用于芯片间通信，为了防止与GPIO寄存器等出现问题我们这里使用IIC0作为通信接口与OLED通信。

在龙芯1b核心手册中，可以看到IIC1 和IIC2与CAN共用物理接口并通过MUX来选择物理接口的链接协议。但是为了避免将过程复杂化我们直接使用IIC0通道这个拥有独立物理接口的IIC通道。 它的两个物理针脚在开发板的PAD上，标记为：I2C\_SCL 和I2C\_SDA就是IIC0 通道的数据和时钟接口了。

0.96寸OLED支持两种协议分别是IIC和SPI协议。 同样的，这里我们只讨论OLED在IIC协议下的工作状态。

添加头文件，以及在bsp.h中调用函数模块

首先要注意的是LoongIDE 已经提供了关于IIC与外部通讯的相关API函数，可以在龙芯1b开发学习套件用户手册中第四章找到。当然，类似的OLED驱动的项目基于ARM架构也是可以找到作为参考的。

那么我们首先开始解释其中最重要的两个函数：

Write\_IIC\_Data()和Write\_IIC\_Command();

这两个是最重要的两个函数，那么我们从定义开始看。‘

unsigned char cmd\_buf**[**2**]** **=** **{**0x00**,** 0x00**};**

unsigned char data\_buf**[**2**]** **=** **{**0x40**,** 0x00**};**

unsigned int Addr **=** 0b0111100**;**

int rw **=** 0**;**

//secondary driver API function

void Write\_IIC\_Data**(**unsigned char IIC\_Data**){**

//start the transmition

ls1x\_i2c\_send\_start**(**busI2C0**,** Addr**);**

//ask for the write and wait for reply

ls1x\_i2c\_send\_addr**(**busI2C0**,** Addr**,** rw**);**

//get the data to sent

data\_buf**[**1**]** **=** IIC\_Data**;**

//sent the data

ls1x\_i2c\_write\_bytes**(**busI2C0**,** **(**uint8\_t**\*)**data\_buf**,** 2**);**

//data sented close the transmition

ls1x\_i2c\_send\_stop**(**busI2C0**,** Addr**);**

**}**

void Write\_IIC\_Command**(**unsigned char IIC\_Command**){**

//start the transimition

ls1x\_i2c\_send\_start**(**busI2C0**,** Addr**);**

//ask the device and config to write mode

ls1x\_i2c\_send\_addr**(**busI2C0**,** Addr**,** rw**);**

//get the command need to be sent

cmd\_buf**[**1**]** **=** IIC\_Command**;**

//send the command

ls1x\_i2c\_write\_bytes**(**busI2C0**,** **(**uint8\_t**\*)**cmd\_buf**,** 2**);**

//close the trasmition

ls1x\_i2c\_send\_stop**(**busI2C0**,** Addr**);**

**}**

’ 从这部分代码我们基本就实现了龙芯驱动的最基础的功能，我们知道了如何让龙芯给OLED发送指令和数据。注意这里虽然是龙芯作为主机OLED作为从机，但是OLED依旧需要通过IIC回复主机。

那么接下来的主程序就会容易的多了，首先初始化IIC总线，之后初始化OLED中的SSD1306 芯片即可。 然后就只需要将测试的图片输入就足够了。我们直接来看代码

unsigned char show**[][**128**]=**

**{**

**…………**

（**这一部分请见文档第15章 实时时钟的设计与实现 15.4.2部分）**

**};**

int main**()** **{**

//init the IIC bus

int inti\_stu **=** ls1x\_i2c\_initialize**(**busI2C0**);**

//init the OLED (just take this as the standerd init command)

//**这里可以见文档第15章 实时时钟的设计与实现 15.3.8部分**

Write\_IIC\_Command**(**0xae**);**

Write\_IIC\_Command**(**0x20**);**

Write\_IIC\_Command**(**0x10**);**

Write\_IIC\_Command**(**0xb0**);**

Write\_IIC\_Command**(**0xc8**);**

Write\_IIC\_Command**(**0x00**);**

Write\_IIC\_Command**(**0x10**);**

Write\_IIC\_Command**(**0x40**);**

Write\_IIC\_Command**(**0x81**);**

Write\_IIC\_Command**(**0xdf**);**

Write\_IIC\_Command**(**0xa1**);**

Write\_IIC\_Command**(**0xa6**);**

Write\_IIC\_Command**(**0xa8**);**

Write\_IIC\_Command**(**0x3F**);**

Write\_IIC\_Command**(**0xa4**);**

Write\_IIC\_Command**(**0xd3**);**

Write\_IIC\_Command**(**0x00**);**

Write\_IIC\_Command**(**0xd5**);**

Write\_IIC\_Command**(**0xf0**);**

Write\_IIC\_Command**(**0xd9**);**

Write\_IIC\_Command**(**0x22**);**

Write\_IIC\_Command**(**0xda**);**

Write\_IIC\_Command**(**0x12**);**

Write\_IIC\_Command**(**0xdb**);**

Write\_IIC\_Command**(**0x20**);**

Write\_IIC\_Command**(**0x8d**);**

Write\_IIC\_Command**(**0x14**);**

Write\_IIC\_Command**(**0xaf**);**

//show the given picture

unsigned char y**=**0**;**

**for(** y**=**0**;**y**<**8**;**y**++)**

**{**

Write\_IIC\_Command**(**0xb0**+**y**);**

Write\_IIC\_Command**(**0x00**);**

Write\_IIC\_Command**(**0x10**);**

unsigned char x **=** 0**;**

**for(**x**=**0**;**x**<**128**;**x**++)**

**{**

Write\_IIC\_Data**(**show**[**y**][**x**]);**

**}**

**}**

int testFinish **=** 0**;**

**return** 0**;**

**}**