物联网实验箱

实验手册



目 录

1.	物联网	网实验箱介绍1
	1.1.	【核心板介绍】1
	1.2.	【硬件电路介绍】1
		1.2.1. 电源接口1
		1.2.2. 指示灯
		1.2.3. 处理器
		1.2.4. 调试口2
		1.2.5. 处理器
		1.2.6. FLASH
		1.2.7. 温湿度
		1.2.8. 气压
		1.2.9. 继电器
		1.2.10. 六轴
		1.2.11. 485 总线2
		1.2.12. AD/DA
		1.2.13. 纽扣电池
		1.2.14. TF 卡
		1.2.15. 数码管2
		1.2.16. LORA
		1.2.17. SIM 卡
		1.2.18. WiFi
		1.2.19.4G 模组
		1.2.20. 按键
		1.2.21. RFID
		1.2.22. 蜂鸣器
		1.2.23. 光照采集
		1.2.24. 声音模块
		1.2.25. 微波感应
		1.2.26. 红外收发
		1.2.27. 舵机
		1.2.28. 存储器
		1.2.29. OLED 屏幕
		1.2.30. 风扇
		1.2.31. 蓝牙
		1.2.32. 2.4G
		1.2.33. 指纹模块
2.	开发现	不境安装3



2.1. 编译器安装	3
2.2. ST-Link 驱动安装	7
2.3. 程序编译和程序下载	7
基础实验	10
3. LED 灯应用实验	10
3.1. 【实验内容】	10
3.2. 【实验目的】	10
3.3. 【实验平台】	10
3.4. 【实验原理】	10
3.4.1. LED 灯	10
3.4.2. LED 串联电阻计算	10
3.4.3. LED 电路分析	11
3.5. 【实验步骤】	11
3.6. 【实验现象】	12
4. 蜂鸣器应用实验	13
4.1. 【实验内容】	13
4.2. 【实验目的】	13
4.3. 【实验平台】	13
4.4. 【实验原理】	13
4.4.1. 蜂鸣器	13
4.4.2. 蜂鸣器电路分析	14
4.5. 【实验步骤】	14
4.6. 【实验现象】	14
5. 按键应用实验	15
5.1. 【实验内容】	15
5.2. 【实验目的】	15
5.3. 【实验平台】	15
5.4. 【实验原理】	15
5.4.1. 按键电路及检测原理	15
5.4.2. 按键电路分析	16
5.4.3. 串口电路分析	16
5.5. 【实验步骤】	17
5.6. 【实验现象】	
6. 继电器实验	19
6.1. 【实验内容】	19
6.2. 【实验目的】	19
6.3. 【实验平台】	19
6.4. 【实验原理】	19
	-



	6.5.	【实验步骤】	
	6.6.	【实验步骤】	
7.	电平输	入实验	
	7.1.	【实验内容】	
	7.2.	【实验目的】	
	7.3.	【实验平台】	
	7.4.	【实验原理】	
	7.5.	【实验步骤】	
	7.6.	【实验现象】	
8.	风扇实	[验	
	8.1.	【实验内容】	
	8.2.	【实验目的】	
	8.3.	【实验平台】	
	8.4.	【实验原理】	
	8.5.	【实验步骤】	
	8.6.	【实验现象】	
9.	人体感	这应应用实验	
	9.1.	【实验内容】	
	9.2.	【实验目的】	
	9.3.	【实验平台】	
	9.4.	【实验原理】	
	9.5.	【实验步骤】	
	9.6.	【实验现象】	
10	. 定时	器应用实验	
	10.1.	【实验内容】	31
	10.2.	【实验目的】	31
	10.3.	【实验平台】	31
	10.4.	【实验原理】	31
	10.5.	【实验步骤】	32
	10.6.	【实验现象】	32
11	. OLED	显示屏应用实	验
	11.1.	【实验内容】	33
	11.2.	【实验目的】	33
	11.3.	【实验平台】	33
	11.4.	【实验原理】	33
	11.5.	【实验步骤】	34
	11.6.	【实验现象】	34
12	. SHT20	温湿度应用实	ç 验



	12.1.	【实验内容】	
	12.2.	【实验目的】	
	12.3.	【实验平台】	
	12.4.	【实验原理】	
	12.5.	【实验步骤】	
	12.6.	【实验现象】	
13.	MPU60	50 应用实验.	
	13.1.	【实验内容】	
	13.2.	【实验目的】	
	13.3.	【实验平台】	
	13.4.	【实验原理】	
	13.5.	【实验步骤】	
	13.6.	【实验现象】	
14.	MPU60	50-DMP 应用	实验48
	14.1.	【实验内容】	
	14.2.	【实验目的】	
	14.3.	【实验平台】	
	14.4.	【实验原理】	
	14.5.	【实验步骤】	
	14.6.	【实验现象】	
15.	BMP18	0应用实验	
	15.1.	【实验内容】	
	15.2.	【实验目的】	
	15.3.	【实验平台】	
	15.4.	【实验原理】	
	15.5.	【实验步骤】	
	15.6.	【实验现象】	
16.	EEPRON	и应用实验	55
	16.1.	【实验内容】	
	16.2.	【实验目的】	
	16.3.	【实验平台】	
	16.4.	【实验原理】	
	16.5.	【实验步骤】	
	16.6.	【实验现象】	
17.	电源检	浏实验	
	17.1.	【实验内容】	
	17.2.	【实验目的】	
	17.3.	【实验平台】	



	17.4.	【实验原理】	
	17.5.	【实验步骤】	59
	17.6.	【实验现象】	
18.	光感检	〕测实验	61
	18.1.	【实验内容】	61
	18.2.	【实验目的】	61
	18.3.	【实验平台】	61
	18.4.	【实验原理】	61
	18.5.	【实验步骤】	
	18.6.	【实验现象】	
19.	声音枪	〕测实验	
	19.1.	【实验内容】	
	19.2.	【实验目的】	
	19.3.	【实验平台】	
	19.4.	【实验原理】	64
	19.5.	【实验步骤】	65
	19.6.	【实验现象】	65
20.	滑动变	泛阻器实验	
	20.1.	【实验内容】	
	20.2.	【实验目的】	
	20.3.	【实验平台】	
	20.4.	【实验原理】	
	20.5.	【实验步骤】	67
	20.6.	【实验现象】	67
21.	RTC 实	验	
	21.1.	【实验内容】	
	21.2.	【实验目的】	
	21.3.	【实验平台】	
	21.4.	【实验原理】	
	21.5.	【实验步骤】	
	21.6.	【实验现象】	
22.	舵机实	兴验	
	22.1.	【实验内容】	
	22.2.	【实验目的】	
	22.3.	【实验平台】	
	22.4.	【实验原理】	
	22.5.	【实验步骤】	
	22.6.	【实验现象】	



23.	NFC应	用实验	
	23.1.	【实验内容】	
	23.2.	【实验目的】	
	23.3.	【实验平台】	
	23.4.	【实验原理】	
	23.5.	【实验步骤】	
	23.6.	【实验现象】	
综合	含实验		
24.	楼道灯	「控实验	
	24.1.	【实验内容】	
	24.2.	【实验目的】	
	24.3.	【实验平台】	
	24.4.	【实验原理】	
	24.5.	【实验步骤】	
	24.6.	【实验现象】	
25.	密码锁	《实验	
	25.1.	【实验内容】	
	25.2.	【实验目的】	
	25.3.	【实验平台】	
	25.4.	【实验原理】	
	25.5.	【实验步骤】	
	25.6.	【实验现象】	
26.	环境监	涵实验	
	26.1.	【实验内容】	
	26.2.	【实验目的】	
	26.3.	【实验平台】	
	26.4.	【实验原理】	
	26.5.	【实验步骤】	
	26.6.	【实验现象】	
27.	公交卡	亲验	
	27.1.	【实验内容】	
	27.2.	【实验目的】	
	27.3.	【实验平台】	
	27.4.	【实验原理】	
	27.5.	【实验步骤】	
	27.6.	【实验现象】	
28.	闹钟实	<验	
	28.1.	【实验内容】	



28.2.	【实验目的】	88
28.3.	【实验平台】	88
28.4.	【实验原理】	88
28.5.	【实验步骤】	88
28.6.	【实验现象】	88



1. 物联网实验箱介绍

1.1.【核心板介绍】

XXX

1.2.【硬件电路介绍】

1.2.1. 电源接口

电源输入为12V,接口如下图所示,请使用原配的电源适配器。



SW9 为电源总开关,这里加了电源指示灯 LED5,在接上电源以后,打开开关, LED5 亮表示电源输入正常,灯不亮请检查输入电源是否有问题。



- 1.2.2. 指示灯
- 1.2.3. 处理器
- 1.2.4. 调试口
- 1.2.5. 处理器
- 1.2.6. FLASH
- 1.2.7. 温湿度
- 1.2.8. 气压
- 1.2.9. 继电器
- 1.2.10.六轴
- 1.2.11.485 总线
- 1.2.12. AD/DA
- 1.2.13.纽扣电池
- 1.2.14.TF 卡
- 1.2.15. 数码管
- 1.2.16.LORA
- 1.2.17.SIM 卡
- 1.2.18. WiFi
- 1.2.19.4G 模组
- 1.2.20.按键
- 1.2.21. RFID
- 1.2.22.蜂鸣器
- 1.2.23. 光照采集



1.2.24. 声音模块

1.2.25. 微波感应

1.2.26.红外收发

1.2.27. 舵机

- 1.2.28.存储器
- 1.2.29.OLED 屏幕
- 1.2.30.风扇
- 1.2.31. 蓝牙
- 1.2.32.2.4G
- 1.2.33.指纹模块

2. 开发环境安装

2.1. 编译器安装

编译环境是我们开发过程中的基础,首先我们来安装编译环境。

2.1.1. 打开提供的软件安装包,有以下三个文件;

W mdk511a.exe	2016/11/23 16:56	应用程序	308,243 KB
📲 mdkcm511a.exe	2016/11/23 16:58	应用程序	357,735 KB

安装编译器,双击 mdk511a.exe 启动安装,默认配置,选择下一步,直至安装完成;



Setup MDK-ARI	VI V5.11a ×
Welcome to Keil MDK-ARM Release 6/2014	
This SETUP program installs:	
MDK-ARM V5.11a	
This SETUP program may be used to update a previous p However, you should make a backup copy before procee	roduct installation. ding.
It is recommended that you exit all Windows programs bef	ore continuing with SETUP.
Follow the instructions to complete the product installation	
Keil MDK-ARM Setup	
	<

2.1.2. 安装芯片库,双击 mdkcm511a.exe 启动安装,默认配置,选择下一步,直至安装完成;

Setup MDK Cortex-M Lega	acy Device Support V5.11a
Welcome to Keil MDK-ARM Release 6/2014	
This SETUP program installs: MDK Cortex-M Legacy Device Suppo	ort V5.11a
This SETUP program may be used to update a previo However, you should make a backup copy before pr	ous product installation. oceeding.
It is recommended that you exit all Windows programs Follow the instructions to complete the product install	s before continuing with SETUP. ation.
— Keil MDK-ARM Setup	<< Back Next >> Cancel

2.1.3. 找到编译器图标,右键以管理员身份运行,点击 File 选项下面的 License Management, 复制 CID 代码;



V	
1 1 1 ガ ガ ガ ガ (の)	
SkyDrive Pro(S) 打开文件所在的位置(I)	
 ▲ 通过OO发送到 ● 以管理员身份运行(A) 	<u> </u>
固定到"开始"屏幕(P) ਡੋ Edit with Notepad++	
使用 Microsoft Defender扫描 2010 添加到圧缩文件(A)	
	_
发送到(N)	\rightarrow
剪切(T) 复制(C)	

右键快捷方式,以管理员方式运行

Name:	fsdf ly4287100@hotmail.com	CID: CIF1V-VZECU	1
Company:	dfgsdf	Get LIC via Internet	
Product	License ID Code (LIC)/Product variant	Support Period	
MDK-ARM	Plus TRND5-I4PCQ-WM7AZ-CLCLF-3AF1P-C9VDR-KMD	R9 Uninstalled	
	se ID Code (I IC):	Add LIC Uninstall	

复制 CID 代码

2.1.4. 运行 keygen.exe,将复制的代码粘贴到 CID 框中,同时选择 ARM,点击 generate,即 生成一串代码,将其复制下来;



🗲 Keil Generic Keygen - EDGE 🛛 🕅
-Keygen License Details CID: CIF1V-VZECU Targe
Prof. Developers Kit (Plus)
Generate Exit

2.1.5. 打开 KEIL,将其粘贴到 License management 中的 New license ID 里面,点击 Add LIC.出现红色边框所示,说明破解成功了。到这里安装破解已经结束了。

License Manag	gement	×
Single-User I	icense Floating License Floating License Admini	strator FlexNet License
Customer Inf	omation isdf ly4287100@hotmail.com	Computer ID CID: CIF1V-VZECU
Company: Email:	dfgsdf [fsdf@dfsd	Get LIC via Internet
Product MDK-ARM P	License ID Code (LIC)/Product variant Support P lus 47GU5-GMTRT-Z90G9-4ZEBT-ICFLU-JWJ4S Expires: D	Dec 2032
New License	e ID Code (LIC):	Add LIC Uninstall
Evaluate M	DK Professional Close	Help



2.2. ST-Link 驱动安装

2.2.1. 安装 ST-Link 驱动,双击 dpinst_amd64.exe 安装,默认配置,点击下一步完成安装;2.2.2. 插入 ST-Link,在设备管理器中检查是否有设备,有红色方框所示,则驱动安装成功:

4	设备管理器
文件(F) 操作(A) 查看(V) 帮助(H)	
▷ 🕁 IDE ATA/ATAPI 控制器	
▷ 🛄 处理器	
▷ 👝 磁盘驱动器	
▷ 存储控制器	
▷ 🖶 打印队列	
▷ 🦢 电池	
▷ 🚏 端囗 (COM 和 LPT)	
▶ 1 ● 计算机	
▷ 🛄 监视器	
▷ 🔤 键盘	
▶ 🚯 蓝牙	
▷ 🖏 人体学输入设备	
▷ 圓 软件设备	
声音、视频和游戏控制器	
▷ 🖉 鼠标和其他指针设备	
> 🖕 通用串行总线控制器	
🔺 闄 通用串行总线设备	
🖡 STM32 STLink	
> 🗟 图像设备	

2.3. 程序编译和程序下载

- **2.3.1.** ST-Link 和爱望物联网开发板连接,分别连接 ST-Link SWCLK,GND,SWDIO 和爱望物联 网开发板 CLK, GND, DIO,注意一一对应;
- 2.3.2. 打开流水灯工程,点击配置控件(如图所示),在 debug 下点击 setting,查看配置 是都和图片所示配置一致,不一致则修改为一致;

¥5	C:\Users\Administrator\Desktop\实验箱课程\第一节:流水灯上程\LED.uvproj - µVision
File Edit View Project Flash Debug Peripherals To	ools SVCS Window Help
□ □	陰 陰 陵 章 章 //彡 //浪 遊 Close_Buzzer
	配置控件
Options for Target	t 'STM 3 2F103RE'
Device Target Output Listing User C/C++ As	sm Linker Debug Utilities
C Use Simulator Settings	
Image: Image: Contract of the start in t	Load Application at Startup Initialization File: Load Application at Startup Edit
Restore Debug Session Settings	Restore Debug Session Settings
✓ Breakpoints ▼ Toolbox	✓ Breakpoints ✓ Toolbox
✓ Watch Windows & Performance Analyzer	Vatch Windows
Memory Display Viswer	I ✓ Memory Display Viewer
CPU DLL: Parameter: SARMCM3.DLL - REMAP	Driver DLL: Parameter: SARMCM3.DLL
Dialog DLL: Parameter: DCM.DLL pCM3	Dialog DLL: Parameter: TCM.DLL pCM3
0K Canc	el Defaults Help



WDIO 0x1B	A01477 ARM	l CoreSight SW-I	DP	Up
				Dow
 Automatic D 	etection	ID CODE:		
C Manual Con	figuration _D	vevice Name:		
Add De	lete Update	IR len:		
	Cache	Options	-Download On	tions
Autodetect	▼	iche Code	□ Verify Cod	le Download
	Ca	che Memory	Download	to Flash
	Automatic D Manual Cont Add Del	Automatic Detection Manual Configuration Add Delete Update Add Delete Update Cache V Cache V Cache V	Automatic Detection ID CODE: Manual Configuration Device Name: Add Delete Update IR len: Cache Options Autodetect Cache Code Cache Memory	Automatic Detection ID CODE: Manual Configuration Device Name: Add Delete Update IR len: Cache Options Cache Options Cache Code Cache Code Cache Memory

2.3.3. 查看 FlashDownload 选项, flash 存储空间是否为 512k(芯片型号不一样,存储空间不一样);

		Corte	ex-M Target Di	river Setup		
ebug Trace	Flash Download	I]				
Download F	unction C Erase Full Chip C Erase Sectors C Do not Erase ag Algorithm	o I⊽ Program I⊽ Verify I⊽ Reset ar	n S nd Run	/l for Algorithm	Size: 0x10	000
Descriptic STM32F10	ın Ix High-density	Device Size 512k	Device Type On-chip Flash	Address R 08000000H - 0	ange 807FFFFH	
		Ac	S dd Rer	tart:	Size:	

2.3.4. 编译工程,点击 Rebuild 编译工程,编译无错误警告;

Ws								C:\L	Jsers\A	dminist	rato	or\Desktop\∄	实验箱课程	』(第一1	古:流水	、 灯工	[程\LE	ED.uvp	oroj - µ	Vision
File	Edit	View	Project	Flash	Debug	Peripherals	Tools	SVCS	Window	Help										
	2		X D		9 0	$\leftrightarrow \Rightarrow \mid p$	12 1	£ 19. I		//= //_R	1	Close_Buzzer	~	🗟 🇖	0	• •	8	e	- 4	•
٩		🖄 🛸		STME	2F103RE	v	st 🛃	5 E -	è 🔶 😫	2										
Project	:			<u> </u>	ф.	× 🚺 🔝 🖬	nain.c													
				Reb	uild															



Build Output							
Build target 'STM32F103RE'							
linking							
Program Size: Code=75960 RO-data=7116 RW-data=992 ZI-data=49440							
FromELF: creating hex file							
".\output\LED.axf" - 0 Error(s), 0 Warning(s).							

2.3.5. 下载程序,点击 load 控件,所图所示则为下载成功,程序运行,查看流水灯效果。

V								C:\U	Jsers\Ac	Iministrat	or\Desktop\多	实验箱课程\第一节:流水灯工程\LED.uvproj - µVision
File	Edit	View	Project	Flash	Debug	Peripherals	Tools	SVCS	Window	Help		
	💕 🖌	10	X Da	(9.0	← → 🍖	12.1	27 2		//= //👳 🛛 🗖	Close_Buzzer	🚽 🗟 🥐 🎯 🔶 🔿 🔗 🐘
٩		š 🧼		STM3	2F103RE	>	<u> </u>	8 6.	🔶 🐡 🔞	2		
					$\overline{}$	oad						

若弹出图下框,请选择否。

ST-LINK Firmware Upgrade									
Old ST-LINK firmware detected. Do you want to upgrade it?									
No									
	grade ×								

Build Output

Application	running		
Load "C:\\Us	sers\\Adm	inist	trator\\Desktop
Erase Done.			
Programming	Done.		
Verify OK.			
Application	running		



基础实验

3. LED 灯应用实验

3.1.【实验内容】

编写程序,控制实验平台的发光二极管 LED1、LED2、LED3、LED4,使它们有规律的点亮和 熄灭。代码分别在仿真器控制下、和脱离仿真环境运行。

3.2. 【实验目的】

掌握 stm32f103zet6 芯片 I/O 控制器、及系统初始化相关部分的配置方法; 掌握如如何通过 realview MDK 环境将代码烧写到目标板。

3.3.【实验平台】

爱望实验箱

3.4. 【实验原理】

3.4.1. LED 灯

发光二极管与普通二极管一样是由一个 PN 结组成,也具有单向导电性。当给发光二极管加 上正向电压后,从 P 区注入到 N 区的空穴和由 N 区注入到 P 区的电子,在 PN 结数微米分别 与 N 区的电子和 P 区的空穴复合,产生自发辐射的荧光。常用的是发红光、绿光或黄光的 二极管。发光二极管的反向击穿电压大于 5 伏。它的正向伏安特性曲线很陡,使用时必须串 联限流电阻以控制通过二极管的电流。

3.4.2. LED 串联电阻计算

发光二极管的压降是比较固定的,通常红色为 1.6V 左右,绿色有 2V 和 3V 两种,黄色和橙 色约为 2.2V,蓝色为 3.2V 左右。对于常用的几毫米大小的二极管,其工作电流一般在 2 毫 安至 20 毫安之间,电流越大亮度越高,用电源电压减去二极管的压降,再除以设定的工作 电流,就得出限流电阻的阻值

串联 R=(U-LED 压降) / 设定的工作电流



3.4.3. LED 电路分析

利用 GPIO0、GPIO1、GPIO2、GPIO3 三个 IO 口, 假如外部电源给 LED 供电, GPIO0 口输出 低电平时(0), LED 灯就会亮, IO 口输出高电平(1), LED 就会灭。同理, GPIO1、GPIO2、 GPIO3 输出低电平时(0), LED 灯就会亮, IO 口输出高电平, LED 就会灭。

3.5.【实验步骤】

打开实验箱。



图 3.5.1

第一步:连接下载器。





图 3.5.2



第二步:确保下载器连接正确后,接通电源,按下电源总开关白色按钮。

图 3.5.3

第三步,编译并下载程序。观察实验结果。

3.6.【实验现象】

可以看到 LED1、LED2、LED3、LED4 按照 100ms 的时间间隔依次变化状态。



4. 蜂鸣器应用实验

4.1.【实验内容】

编写程序,控制实验平台的蜂鸣器,有规律控制蜂鸣器的叫和不叫。代码分别在仿真器控制下、和脱离仿真环境运行。

4.2.【实验目的】

掌握 stm32f103zet6 芯片 I/O 控制器、及系统初始化相关部分的配置方法; 掌握如如何通过 realview MDK 环境将代码烧写到目标板。

4.3.【实验平台】

爱望实验箱

4.4.【实验原理】

4.4.1. 蜂鸣器

蜂鸣器是一种一体化结构的电子讯响器,采用直流电压供电,广泛应用于计算机、打印机、 复印机、报警器、电子玩具、汽车电子设备、电话机、定时器等电子产品中作发声器件。蜂 鸣器主要分为压电式蜂鸣器和电磁式蜂鸣器两种类型。

实验箱板载的蜂鸣器是电磁式的有源蜂鸣器,如图所示:



图 5.1 有源蜂鸣器

这里的有源不是指电源的"源",而是指有没有自带震荡电路,有源蜂鸣器自带了震荡电路, 一通电就会发声;无源蜂鸣器则没有自带震荡电路,必须外部提供 2~5Khz 左右的方波驱 动,才能发声。



4.4.2. 蜂鸣器电路分析



STM32 的单个 IO 最大可以提供 25mA 电流(来自数据手册),而蜂鸣器的驱动电流是 30mA 左右,两者十分相近,但是全盘考虑,STM32 整个芯片的电流,最大也就 150mA,如果用 IO 口直接驱动蜂鸣器,其他地方用电就得省着点了…所以,我们不用 STM32 的 IO 直接驱动蜂鸣器,而是通过三极管扩流后再驱动蜂鸣器,这样 STM32 的 IO 只需要提供 不到 1mA 的电流就足够了,随着 Q4 的导通,Q18 也会导通,所以 LED24 是伴随着蜂鸣器 的响应而亮灭的。

4.5. 【实验步骤】

打开实验箱。

- 第一步,连接下载器。
- 第二步,确保下载器连接正确后,接通电源,按下电源总开关白色按钮。

第三步,编译并下载程序。观察实验结果。

4.6.【实验现象】

可以听到蜂鸣器按照 1s 周期声响和不响,同时 LED24 伴随着亮和灭。



5. 按键应用实验

5.1.【实验内容】

编写程序,按动实验箱的 SW2、SW3、SW4、SW5、SW6,通过串口打印提示按键信息。 代码分别在仿真器控制下、和脱离仿真环境运行。

5.2.【实验目的】

掌握按键接口电路 掌握查看原理图方法 掌握按键扫描编程方法 掌握按键抖动处理方法 掌握串口使用方法

5.3.【实验平台】

爱望实验箱

5.4.【实验原理】

5.4.1. 按键电路及检测原理

当按下按键开关的按钮,它的弹片就会因为受到弹力作用而发生形变向下触摸焊片,并 成功的使开关里的四个引脚相接通,从而实现电路的接通。当撤离外力的时分弹片就会恢复 到本来的状况,引脚之间也不会接通,所以电路就会处于一种闭合的状况。

当按键未被按下时,此时 MCU 的 IO 口检测到高电平;当按键被按下时,此时检测到低 电平.上拉电阻是为了保证按键未被按下时处于一个固定的高电平。



5.4.2. 按键电路分析





处理按键输入信号的电路。主要功能是消除按键抖动造成的重复信号输入,带 CPU 的设备通常用软件消除抖动,不带 CPU 的通常用单稳态电路,简单的用 RC 电路,不稳定。

5.4.3. 串口电路分析



图 6.4.3.1





图 6.4.3.2

从电路图可以看出, stm32f103zet6 的 PA9(TX)和 PA10(RX), 连接到 CH340N 芯片上, CH340N 的 D+和 D-连接到 USB 接口的两个脚, CH340N 芯片是一种 USB 转串口芯片,它可以将 USB 接口转换为 UART 串口接口,使计算机可以通过 USB 接口和单片机等设备进行通信。

5.5.【实验步骤】

打开实验箱。

第一步,连接下载器。

第二步,确保下载器连接正确后,接通电源,将 USB 接上 J17,另一端接入 pc,按下电源总开关白色按钮。





图 5.5.1

第三步,编译并下载程序。观察实验结果。

5.6.【实验现象】

打开串口助手, 按动按键, 观察串口助手的打印信息与按键是否一致。



图 6.6.1



图 6.6.2



6. 继电器实验

6.1.【实验内容】

编写程序,按动实验箱的 SW2(DOWN)、SW6(UP),控制继电器的闭合和断开。代码 分别在仿真器控制下、和脱离仿真环境运行。

6.2.【实验目的】

掌握按键接口电路 掌握查看原理图方法 掌握按键扫描编程方法 掌握按键抖动处理方法 掌握按键抖动处理方法 掌握按键抖动处理方法

6.3.【实验平台】

爱望实验箱

6.4.【实验原理】

继电器(Relay),也称电驿,是一种电子件,它具有控制系统(又称输入回路)和被控制系统(又称输出回路),通常应用于自动控制电路中,它实际上是用较小的电流去控制较大电流的一种"自动开关"。故在电路中起着自动调节、安全保护、转换电路等作用。本节实验就是通过按键来控制两个继电器的开和关。当 stm32f103zet6 的 PF6 或 PF10 检测到低电平(按键按下), PF2 或 PF3 就改变一次的电平状态,原来是高电平就会变成低电平,原来是低电平就会变成高电平。





图 7.4.1



图 7.4.2





6.5.【实验步骤】

打开实验箱。

第一步,连接下载器。

第二步,确保下载器连接正确后,接通电源,按下电源总开关白色按钮。

第三步,编译并下载程序。观察实验结果。

6.6.【实验步骤】

按下 SW6 或者 SW2,相应的继电器会翻转状态,可以通过 LED22 和 LED27 体现出来, 另外,也可以外部电路接上连接器端子的 NC(常闭)或 NO(常开)与 GND,感受一下, 小信号控制大信号的操作。



图 7.6.1





7. 电平输入实验

7.1.【实验内容】

编写程序,通过高、低电平接入光耦的输入,控制 LED1、LED2 的亮灭。代码分别在仿 真器控制下、和脱离仿真环境运行。

7.2. 【实验目的】

掌握 GPIO 输入输出使用方法 掌握查看原理图方法 掌握光耦使用方法

7.3.【实验平台】

爱望实验箱

7.4.【实验原理】

光电耦合器(optical coupler, 英文缩写为 OC)亦称光电隔离器,简称光耦。光电耦合器以光为媒介传输电信号。它对输入、输出电信号有良好的隔离作用,所以,它在各种电路中得到广泛的应用。目前它已成为种类最多、用途最广的光电器件之一。光耦合器一般由三部分组成:光的发射、光的接收及信号放大。输入的电信号驱动发光二极管(LED),使之发出一定波长的光,被光探测器接收而产生光电流,再经过进一步放大后输出。这就完成了电一光一电的转换,从而起到输入、输出、隔离的作用。由于光耦合器输入输出间互相隔离,电信号传输具有单向性等特点,因而具有良好的电绝缘能力和抗干扰能力。本节实验就是通过光耦的信号检测,从而改变 LED1、LED2 的亮灭状态,当光耦输入电压 3.3V 时,光耦内部 三极管导通,GPIO 检测到低电平,从而通过程序控制 LED 熄灭,当光耦输入 0V 时,光耦内 部三极管载止,GPIO 检测到高电平,从而通过程序控制 LED 熄灭,当光耦输入 0V 时,光耦内











图 8.4.2





图 8.4.3

7.5.【实验步骤】

打开实验箱。

第一步,连接下载器。

第二步,确保下载器连接正确后,接通电源,按下电源总开关白色按钮。

第三步,编译并下载程序。观察实验结果。

7.6.【实验现象】

不接任何电平线的情况下,LED1 和 LED2 亮着。将 GND 或者 3.3V 通过杜邦线接入绿色 端子的 IPUT1 或者 INPUT2,可以看到 LED1 或者 LED2 的亮灭。

	LED1 状态	LED2 状态
INPUT1 (GND)	亮	不变
INPUT1 (3.3V)	灭	不变
INPUT2 (GND)	不变	亮
INPUT2 (3.3V)	不变	灭



图 8.6.1



8.风扇实验

8.1.【实验内容】

编写程序,通过 GPIO 口输出,控制 mos 管的开关,从而控制直流风扇的转动和停止。 代码分别在仿真器控制下、和脱离仿真环境运行。

8.2. 【实验目的】

掌握 GPIO 输出使用方法 掌握查看原理图方法 掌握三极管、MOS 管使用方法 掌握直流风扇的使用方法

8.3.【实验平台】

爱望实验箱

8.4. 【实验原理】

直流风扇,是通过直流电压和电磁感应,由电能转化成机械从而带动风叶转动的风扇。 分析一下电路图,可以看到是通过 PC5 来控制 Q12 的开关,从而控制 Q11 的开关。如果 PC5 输出高电平(3.3V),Q12 导通,PMOS 管 Q11 的 Vgs 小于阈值(可以查看 mos 管 的 datasheet),Q11 导通,则风扇转动,反之风扇停止转动。为何我们不通过 GPIO 直接 驱动风扇呢?因为我们的 stm32f103zet6 的 IO 口驱动能力有限,不足以驱动这种功率器件。



图 9.4.1





图 9.4.2

8.5.【实验步骤】

打开实验箱。

第一步,连接下载器。

第二步,确保下载器连接正确后,接通电源,将风扇的接头接入板子背面的接头 J50, 按下电源总开关白色按钮。

第三步,打开工程,编译并下载程序。观察实验结果。

8.6.【实验现象】

可以看到直流风扇每隔 2s 转动或者停止,同时伴随着 LED23 的亮和灭。



图 8.6.1



9.人体感应应用实验

9.1.【实验内容】

编写程序,通过 RCWL-0516 微波雷达感应开关/人体感应传感器检测人体的移动,通过 LED4 (红灯)显示效果。代码分别在仿真器控制下、和脱离仿真环境运行。

9.2.【实验目的】

掌握 GPIO 输入使用方法 掌握查看原理图方法 掌握人体感应模块使用方法

9.3.【实验平台】

爱望实验箱

9.4.【实验原理】

RCWL0516 传感器主要使用微波多普勒雷达方法来感知移动物体,而不是嗅探来自移动 人的黑体辐射。该传感器的灵敏度范围约为七米。一旦这个传感器被激活,那么 TTL 电平 OUT 引脚将从低电压切换到高电压并持续一段固定的时间,然后返回到其非活动状态。

序号	名称	管脚定义
1	3V3	3V3 电源输出,100mA 对外部
		的供电能力
2	GND	电源地
3	OUT	控制输出,检测到移动物体输
		出高电平
4	VIN	输入电压, 4-28V
5	CDS	使能控制芯片,低于 0.7V 的
		话,OUT 引脚输出低电平,
		如果大于 3.3V, 检测正常工
		作。CDS 管脚外接光敏电阻,
		可以在白天关闭模块功能。

模块的输出管脚定义:


本节通过检测人体感应模块(图 9.4.1)的 4 脚输出电平,从而去控制 LED4(图 9.4.2)的状态。



图 9.4.1



图 9.4.2

9.5.【实验步骤】

打开实验箱。

第一步,连接下载器。

第二步,确保下载器连接正确后,接通电源,将风扇的接头接入板子背面的接头 J50, 按下电源总开关白色按钮。

第三步,打开工程,编译并下载程序。观察实验结果。

9.6.【实验现象】

我们用手在人体感应模块上方舞动,可以看到红色 LED4 亮起,当手完全离开感应模块, 红色 LED4 就会灭掉。





图 9.6.1



10. 定时器应用实验

10.1. 【实验内容】

编写程序,使用 TIM6 产生一个 1s 的周期,并通过串口 1 打印出来周期更新的信息。代码分别在仿真器控制下、和脱离仿真环境运行。

10.2. 【实验目的】

掌握定时器使用方法 掌握查看原理图方法 掌握串口使用方法

10.3. 【实验平台】

爱望实验箱

10.4. 【实验原理】

基本定时器 TIM6 和 TIM7 各包含一个 16 位自动装载计数器,由各自的可编程预分频器驱动。这 2 个定时器是互相独立的,不共享任何资源。

这个可编程定时器的主要部分是一个带有自动重装载的 16 位累加计数器,计数器的时 钟通过一个预分频器得到。

软件可以读写计数器、自动重装载寄存器和预分频寄存器,即使计数器运行时也可以操 作。

时基单元包含:

- 计数器寄存器(TIMx_CNT)
- 预分频寄存器(TIMx_PSC)
- 自动重装载寄存器(TIMx_ARR)

自动重装载寄存器是预加载的,每次读写自动重装载寄存器时,实际上是通过读写预加 载寄存器实现。根据 TIMx_CR1 寄存器中的自动重装载预加载使能位(ARPE),写入预加载 寄存器的内容能够立即或在每次更新事件时,传送到它的影子寄存器。当 TIMx_CR1 寄存 器的 UDIS 位为'0',则每当计数器达到溢出值时,硬件发出更新事件;软件也可以产生更新 事件;关于更新事件的产生。

本节实验就是通过设置预分频寄存器和自动重装载寄存器的值,使定时器产生1s的中断。TIM6的时钟为72M,再根据我们设计的 arr 和 psc 的值,就可以计算中断时间了。 计算公式如下:

Tout= ((arr+1)*(psc+1))/Tclk;



其中:

Tclk: TIM6 的输入时钟频率(单位为 MHz)。 Tout: TIM6 溢出时间(单位为 us)。

10.5. 【实验步骤】

打开实验箱。

第一步,连接下载器。

第二步,确保下载器连接正确后,接通电源,将 USB 接上 J17,另一端接入 pc,按下电源总开关白色按钮。

第三步,编译并下载程序。观察实验结果。

10.6. 【实验现象】

打开串口助手,配置相关参数,可以看到串口每隔1s打印周期更新信息。

文件(f) 編集(f) 編集(f) 編集(f) 編集(f) ● ① ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	≤ 友善串口调试助手		Constant C	
Image: Construction of the second state of the second	文件(F) 编辑(E) 视图(V) 工具(T) 帮助	i(H)		
#□102# [20:32:40.814] Timer Update □□102# [20:32:40.814] Timer Update □□1500 □ □□1500 □ □□1500 □ □□1500 □ □□1500 □ □□1500 □ □□1500 □ □□1500 □ □□1500 □ □□1500 □ □□1500 □ □15000 □ □15000 □ □15000 □ □15000 □ □15000 □ □15000 □ □1500 □ □1500 □ □1500 □ □1500 □ □1500 □ □1500 □ □1500 □ □1500 □ □1500 □ □1500 □ □1500 □ □1500 □ □1500 □ □1500 □ □1500 □ □1500		· — 🛅 🕸		
iii □ USB-SERIAL CH340 (COMS) ↓ iiii □ USB-SERIAL CH340 (COMS) ↓ iiiii □ USB-SERIAL CH340 (COMS) ↓ iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii		[28-22-66 916] Timor Undato		
ym i ou Sinkk Cive Coundy wh#ac i wh#ac i <		[20:32:40.814] Timer Update		
		[20:32:48.814] Timer Update		
	波特率 115200 👤	[20:32:49.814] Timer Update		
	数据位 8 ▼	[20:32:50.814] Timer Update		
(Kaald) Jadd Image		[20:32:51.814] Timer Update		
停止位 1 「 流 控 None 「 波 控 None 「 (6.10) (1.10) (7.10) (1.10)		[20:32:53.814] Timer Update		
流 控 Mone [28:32:55.816] Timer Update [20:32:55.816] Timer Update [20:32:55.815] Timer Update [20:32:55.815] Timer Update [20:33:08.816] Timer Update [20:33:08.816] Timer Update [20:33:08.816] Timer Update [20:33:08.816] Timer Update [20:33:08.816] Timer Update [20:33:08.816] Timer Update [20:33:08.816] Timer Update [20:33:07.816] Timer Update [20:33:08.817] Timer Update [20:33:08.817] Timer Update [20:33:10.817] Timer Update [20:33:10.817] Timer Update [20:33:11.817] Timer Update [20:33:12.817] Timer Update [20:33:12.817] Timer Update [20:33:12.817] Timer Update [20:33:12.817] Timer Update [20:33:12.817] Timer Update [20:33:11.817] Timer Update [20:33:12.817] Timer Update [20:33:12.817] Timer Update [20:33:12.817] Timer Update [20:33:12.817] Timer Update [20:33:12.817] Timer Update [20:33:12.817] Timer Update [20:33:10.817] Timer Update	停止位 1 🗾	[20:32:54.814] Timer Update		
[28:32:55.815] Timer Update [28:32:57.815] Timer Update [28:32:58.815] Timer Update [28:32:58.815] Timer Update [28:32:59.815] Timer Update [28:33:08.816] Timer Update [28:33:08.817] Timer Update [28:33:08.817] Timer Update [28:33:08.817] Timer Update [28:33:11.817] Timer Update [28:33:11.817] Timer Update [28:33:12.817] Timer Update </td <td>流控 None ▼</td> <td>[20:32:55.816] Timer Update</td> <td></td> <td></td>	流控 None ▼	[20:32:55.816] Timer Update		
撤收设置 [28:32:57.815] Timer Update [28:32:59.815] Timer Update [28:32:59.815] Timer Update [28:32:59.815] Timer Update [28:33:60.816] Timer Update [28:33:60.816] Timer Update [28:33:60.816] Timer Update [28:33:60.817] Timer Update [28:33:60.817] Timer Update [28:33:10.817] Timer Update [28:33:10.817] Timer Update [28:33:11.817] Timer Update [28:33:12.817] Timer Update [28:33:12.817] Timer Update [28:33:12.817] Timer Update [28:33:12.817] Timer Update [28:33:12.817] Timer Update [28:33:12.817] Timer Update [28:33:12.817] Timer Update		[20:32:56.815] Timer Update		
Comparison Comparison </td <td>接收设置</td> <td>[20:32:57.815] Timer Update</td> <td></td> <td></td>	接收设置	[20:32:57.815] Timer Update		
□ 自动执行 □ ② ③ ③ ③ ③ ③ ③ ③ ⑤ ③ ⑤ ⑤ ⑤ ⑤ ⑤ ⑤ ⑥ ⑤ ⑤ ⑤ ⑥ ⑥ ⑤ ⑥ ⑤	● ASCTT C Her	[20:32:59.815] Timer Update		
□ 目初興打 □ 20:33:01.816] Timer Update □ 显示发送 □ 20:33:02.816] Timer Update □ ② ③ ③ ○ ③ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○		[20:33:00.816] Timer Update		
豆示发送 [20:33:02.816] Timer Update 「豆示对间 [20:33:03.816] Timer Update 皮送设置 [20:33:05.816] Timer Update (20:33:06.816] Timer Update [20:33:06.816] Timer Update [20:33:06.816] Timer Update [20:33:06.816] Timer Update [20:33:06.817] Timer Update [20:33:06.817] Timer Update [20:33:08.817] Timer Update [20:33:08.817] Timer Update [20:33:10.817] Timer Update [20:33:10.817] Timer Update [20:33:11.817] Timer Update [20:33:11.817] Timer Update [20:33:12.817] Timer Update [20:33:12.817] Timer Update [20:30:12.817] Timer Update [20:30:12.817] Timer Update [20:30:12.817] Timer Update [20:30:12.817] Timer Update [20:30:12.817] Timer Update [20:30.817] Timer Update [20:30:12.817] Timer Update [20:30.817] Timer Update [20:30:12.817] Timer Update [20:30.817] Timer Update		[20:33:01.816] Timer Update		
▼ 显示时间 [28:33:83.816] 11mer Update 20:33:04.817] Timer Update [20:33:06.816] Timer Update [20:33:06.816] Timer Update [20:33:06.816] Timer Update [20:33:06.816] Timer Update [20:33:06.817] Timer Update [20:33:06.817] Timer Update [20:33:10.817] Timer Update [20:33:11.817] Timer Update [20:33:12.817] Timer Update [20:30.817] Timer Update [20:30.917] Timer Update	□ 显示发送	[20:33:02.816] Timer Update		
	☑ 显示时间	[20:33:03.810] limer update		
发送设置 [20:33:06.816] Timer Update ○ ASCII ○ Hex □ 自动重发 1000 … ms [20:33:07.816] Timer Update [20:33:07.816] Timer Update [20:33:07.816] Timer Update [20:33:07.816] Timer Update [20:33:10.817] Timer Update [20:33:11.817] Timer Update [20:33:12.817] Timer Update [20:		[20:33:05.816] Timer Update		
• ASCII • Hex • 自动重发 1000 • ms • ms • E20:33:60.817] Timer Update [20:33:10.817] Timer Update [20:33:11.817] Timer Update [20:33:12.817] Timer Upda	│ ┌ 发送设置 ─────	[20:33:06.816] Timer Update		
		[20:33:07.816] Timer Update		
[20:33:99.87] Timer Update [20:33:10.817] Timer Update [20:33:11.817] Timer Update [20:33:12.817] Timer Update [20:30] Timer Update [20:30] Timer Update [20:30] Timer Update	□ 自动重发 1000 ÷ ms	[20:33:08.817] Timer Update		
[20:33:11.817] Timer Update [20:33:12.817] Timer Update [20:30:12.817] Timer Update <td></td> <td>[20:33:09.817] limer Update</td> <td></td> <td></td>		[20:33:09.817] limer Update		
[28:33:12.817] Timer Update 定 发送 [A1-1] COM5 OPENED, 115200, 8, NONE, 1, OFF Rx: 32,886 Bytes Tx: 0 Bytes		[20:33:11.817] Timer Update		
		[20:33:12.817] Timer Update		-
发送				_
★ ★ COM5 OPENED, 115200, 8, NONE, 1, OFF Rx: 32,886 Bytes				
COM5 OPENED, 115200, 8, NONE, 1, OFF Rx: 32,886 Bytes Tx: 0 Bytes				发送
COM5 OPENED, 115200, 8, NONE, 1, OFF Rx: 32,886 Bytes Tx: 0 Bytes				
, COM5 OPENED, 115200, 8, NONE, 1, OFF Rx: 32,886 Bytes Tx: 0 Bytes		R1-:		
	COM5 OPENED, 115200, 8, NONE, 1, C	, IFF Rx: 32,886 Bytes Tx: 0 Bytes		

图 10.6



11. OLED 显示屏应用实验

11.1. 【实验内容】

编写程序,使用 SPI 通信,驱动 OLED 显示字符。代码分别在仿真器控制下、和脱离仿 真环境运行。

11.2. 【实验目的】

掌握 SPI 使用方法 掌握查看原理图方法 掌握 OLED 的使用方法

11.3. 【实验平台】

爱望实验箱

11.4. 【实验原理】

4 线串口模式使用的信号线有如下几条:

CS: OLED 片选信号(实验箱使用的模块内部已经接 gnd 了)。

RST(RES): 硬复位 OLED。

DC: 命令/数据标志(0,读写命令;1,读写数据)。

SCLK: 串行时钟线。在 4 线串行模式下, D0 信号线作为串行时钟线 SCLK。 SDIN: 串行数据线。在 4 线串行模式下, D1 信号线作为串行数据线 SDIN。

在 4 线串行模式下,只能往模块写数据而不能读数据。

在 4 线 SPI 模式下,每个数据长度均为 8 位,在 SCLK 的上升沿,数据从 SDIN 移入到 SSD1306,并且是高位在前的。DC 线还是用作命令/数据的标志线。在 4 线 SPI 模式下,写操作的时序如图 11.4.1 所示:



图 11.4.1



对于 OLED 的 IC 芯片 SH1106 的使用,按照这个流程: 复位 SH1106->关闭显示->清零显示->开启显示

我们看这个 oled.c 文件的 OLED_WriteByte 函数实现了 OLED 的写操作, OLED_Init 函数就是实现了 OLED 的初始化。

11.5. 【实验步骤】

第一步,打开实验箱,连接下载器。

第二步,确保下载器连接正确后,接通电源。

第三步,编译并下载程序。观察实验结果。

11.6. 【实验现象】

可以看到 OLED 显示屏每隔 1s 更新数字(数字最大只能显示到 255, 想一想是为什么)。



12. SHT20 温湿度应用实验

12.1. 【实验内容】

编写程序,使用 IIC 通信,驱动 SHT20 温湿度传感器测量环境下的温度和湿度。代码分别在仿真器控制下、和脱离仿真环境运行。

12.2. 【实验目的】

掌握 IIC 使用方法 掌握查看原理图方法 掌握 SHT20 的使用方法

12.3. 【实验平台】

爱望实验箱

12.4. 【实验原理】

IIC(Inter-Integrated Circuit)总线是一种由 PHILIPS 公司开发的两线式串行总线,用于 连接微控制器及其外围设备。它是由数据线 SDA 和时钟 SCL 构成的串行总线,可发送和 接收数据。

在 CPU 与被控 IC 之间、IC 与 IC 之间进行双向传送,高速 IIC 总线一般可达 400kbps 以上。

I2C 总线在传送数据过程中共有三种类型信号, 它们分别是: 开始信号、结束信号和 应答信号。

开始信号: SCL 为高电平时, SDA 由高电平向低电平跳变,开始传送数据。

结束信号: SCL 为高电平时, SDA 由低电平向高电平跳变, 结束传送数据。

应答信号:接收数据的 IC 在接收到 8bit 数据后,向发送数据的 IC 发出特定的低电 平脉冲, 表示已收到数据。CPU 向受控单元发出一个信号后,等待受控单元发出一个应答 信号,CPU 接收到应答信号后,根据实际情况作出是否继续传递信号的判断。若未收到应 答信号,由判断为受控单元出现故障。IIC 总线时序如图 12.4.1:







简单介绍一下 SHT20, 能测量的温度范围是-20^{~100} 度, 湿度范围是 5%^{~90%}, 在测量过 程中,有两种模式,保持主机模式和非保持主机模式。主机模式:测量过程中,SCL 线被封 锁(由传感器控制),在测量时,SHT20将 SCL 拉低制主机进入等待状态,当释放 SCL 线, 表示传感器内部工作接收,可以继续进行数据传送;非保持主机模式:测量过程中,SCL 线 是开发状态,可进行其它通信,可以在总线上处理其它 12C 总线通信任务。

命令	释义	代码
触发T测量	保持主机	11100011
触发 RH 测量	保持主机	11100101
触发T测量	非保持主机	11110011
触发 RH 测量	非保持主机	11110101
写用户寄存器		11100110
读用户寄存器		11100111
软复位		11111110

注: T 代表温度, RH 代表湿度。

开始传输后,随后先传输首字节包括12C设备地址(7bit)和一个SDA方向位(R:1,W:0)。 一个时钟发送一个位,在第8个下降沿之后,通过拉低SDA引脚(ACK位为0),只是传感 器数据接收正常,在发出测量命令之后(11100011代表温度测量,11100101代表相对湿度 测量,这种为保持主机模式,图12.4.1),MCU必须等待测量完成。

36





图 12.4.3 (非保持主机模式)

相对湿度转换公式:

$$RH = -6 + 125 \times \frac{S_{RH}}{2^{16}}$$

温度转换公式:



$$T = -46.85 + 175.72 \times \frac{S^{1}}{216}$$

本节所使用的的硬件包括 SHT20(图 12.4.1)的保持主机模式和 OLED(图 12.4.2)。

温湿度SHT20



图 12.4.1



图 12.4.2

12.5. 【实验步骤】

第一步,打开实验箱,连接下载器。

- 第二步,确保下载器连接正确后,接通电源和 USB 线(J17)。
- 第三步,打开工程编译并下载程序。观察实验结果。
- 第四步,将 J63 和 J64 的跳帽对应到到温湿度。





图 12.5.1

12.6. 【实验现象】

可以看到 OLED 显示了温度和相对湿度,如图 12.6.1:



图 12.6.1

同时把 USB 线接入电脑,通过串口工具也可以看到温湿度值,如图 12.6.2。



■ 友善串口调试助手			7.0	
文件(F) 编辑(E) 视图(V) 工具(T) 帮助	(H)			
□ ∞ ▶ ■ ○ +	· — 🛅 🕸			
串口设置	Temp: 14 1°ີ Humi: 74 6	12		
端 🛛 USB-SERIAL CH340 (COM6) 🚽	Temp: 14.1°C, Humi: 74.0	,~~ %		
波特案 115200 🔍	Temp: 14.1°C, Humi: 74.0	1% 19		
	Temp: 14.1°C, Humi: 74.0	1%		
	Temp: 14.1°C, Humi: 74.1°	%		
校验位 None 🔽	Temp: 14.1°C, Humi: 74.1 Temp: 14.1°C, Humi: 74.1	% %		
	Temp: 14.1°Č, Humi: 74.1	8		
流控 None	Temp: 14.1℃, Humi: 74.1 Temp: 14.1℃, Humi: 74.1	% %		
	Temp: 14.1 $\widecheck{ ext{C}}$, Humi: 74.1	%		
C ASCTI C Nu	Temp: 14.1℃, Humi: 74.1 Temn: 14.1℃, Humi: 74.1	% %		
C 白計協行				
		_		
发送设置				
• ASCII O Hex				
□ 自动重发 1000 🕂 ms				
				_
				发送
	n1-:			•
COM6 OPENED, 115200, 8, NONE, 1, C	, FF Rx: 378 Bytes Tx:	0 Bytes		

图 12.6.2



13. MPU6050 应用实验

13.1. 【实验内容】

编写程序,使用 IIC 通信,驱动 MPU6050 测量六轴数据。代码分别在仿真器控制下、 和脱离仿真环境运行。

13.2. 【实验目的】

掌握 IIC 使用方法 掌握查看原理图方法 掌握 mpu6050 的使用方法

13.3. 【实验平台】

爱望实验箱

13.4. 【实验原理】

MPU6050 是 InvenSense 公司推出的全球首款整合性 6 轴运动处理组件,相较于多组件方案,免除了组合陀螺仪与加速器时之轴间差的问题,减少了安装空间。

MPU6050 内部整合了 3 轴陀螺仪和 3 轴加速度传感器,并且含有一个第二 IIC 接口,可用于连接外部磁力传感器,并利用自带的数字运动处理器(DMP: Digital Motion Processor) 硬件加速引擎,通过主 IIC 接口,向应用端输出完整的 9 轴融合演算数据。有了 DMP, 我们可以使用 InvenSense 公司提供的运动处理资料库,非常方便的实现姿态解算,降低了运动处理运算对操作系统的负荷,同时大大降低了开发难度。

MPU6050 的特点包括:

① 以数字形式输出 6 轴或 9 轴(需外接磁传感器)的旋转矩阵、四元数(quaternion)、欧 拉角格式(Euler Angle forma)的融合演算数据(需 DMP 支持)

② 具有 131 LSBs/°/sec 敏感度与全格感测范围为±250、±500、±1000 与±2000°/sec 的 3 轴角速度感测器(陀螺仪)

③ 集成可程序控制,范围为±2g、±4g、±8g 和±16g 的 3 轴加速度传感器

④ 移除加速器与陀螺仪轴间敏感度,降低设定给予的影响与感测器的飘移

⑤ 自带数字运动处理(DMP: Digital Motion Processing)引擎可减少 MCU 复杂的融合演算数据、感测器同步化、姿势感应等的负荷

⑥ 内建运作时间偏差与磁力感测器校正演算技术,免除了客户须另外进行校正的需求

⑦ 自带一个数字温度传感器

⑧ 带数字输入同步引脚(Sync pin)支持视频电子影相稳定技术与 GPS

⑨ 可程序控制的中断(interrupt),支持姿势识别、摇摄、画面放大缩小、滚动、快速下降



中断、high-G 中断、零动作感应、触击感应、摇动感应功能

⑩ VDD 供电电压为 2.5V±5%、3.0V±5%、3.3V±5%; VLOGIC 可低至 1.8V± 5%

① 陀螺仪工作电流: 5mA, 陀螺仪待机电流: 5uA; 加速器工作电流: 500uA, 加速器省 电模式电流: 40uA@10Hz

- (12) 自带 1024 字节 FIFO, 有助于降低系统功耗
- (13) 高达 400Khz 的 IIC 通信接口
- (4) 超小封装尺寸: 4x4x0.9mm (QFN)

MPU6050 的内部框图如图 13.4.1 所示



图 13.4.1

其中,SCL 和 SDA 是连接 MCU 的 IIC 接口,MCU 通过这个 IIC 接口来控制 MPU6050,另外还有一个 IIC 接口:AUX_CL 和 AUX_DA,这个接口可用来连接外部从 设备,比如磁传感器,这样就可以组成一个九轴传感器。VLOGIC 是 IO 口电压,该引脚 最低可以到 1.8V,我们一般直接接 VDD 即可。AD0 是从 IIC 接口(接 MCU)的地址 控制引脚,该引脚控制 IIC 地址的最低位。如果接 GND,则 MPU6050 的 IIC 地址是: 0X68,如果接 VDD,则是 0X69,注意:这里的地址是不包含数据传输的最低位的(最低 位用来表示读写)!!在本节实验中, AD0 接 GND,因而选择 MPU6050 的 IIC 地址 是 0X68(不含最低位),IIC 通信的时序我们上一个实验已经介绍过,这里就不再细说了。



接下来,我们介绍一下利用 STM32F1 读取 MPU6050 的加速度和角度传感器数据(非中断方式),需要哪些初始化步骤:

1) 初始化 IIC 接口

MPU6050 采用 IIC 与 STM32F1 通信,所以我们需要先初始化与 MPU6050 连接的 SDA 和 SCL 数据线。

2) 复位 MPU6050

这一步让 MPU6050 内部所有寄存器恢复默认值,通过对电源管理寄存器 1(0X6B) 的 bit7 写 1 实现。 复位后,电源管理寄存器 1 恢复默认值(0X40),然后必须设置该寄存 器为 0X00,以唤醒 MPU6050,进入正常工作状态。

3) 设置角速度传感器(陀螺仪)和加速度传感器的满量程范围

这一步,我们设置两个传感器的满量程范围(FSR),分别通过陀螺仪配置寄存器(0X1B) 和加速度传感器配置寄存器(0X1C)设置。我们一般设置陀螺仪的满量程范围为±2000dps, 加速度传感器的满量程范围为±2g。

4) 设置其他参数

这里,我们还需要配置的参数有:关闭中断、关闭 AUX IIC 接口、禁止 FIFO、设置 陀螺仪采样率和设置数字低通滤波器 (DLPF)等。本章我们不用中断方式读取数据,所以 关闭中断,然后也没用到 AUX IIC 接口外接其他传感器,所以也关闭这个接口。分别通过 中断使能寄存器 (0X38)和用户控制寄存器 (0X6A)控制。MPU6050 可以使用 FIFO 存 储传感器数据,不过本章我们没有用到,所以关闭所有 FIFO 通道,这个通过 FIFO 使能 寄存器 (0X23)控制,默认都是 0 (即禁止 FIFO),所以用默认值就可以了。陀螺仪采样 率通过采样率分频寄存器 (0X19)控制,这个采样率我们一般设置为 50 即可。数字低通 滤波器(DLPF)则通过配置寄存器 (0X1A)设置,一般设置 DLPF 为带宽的 1/2 即可。

5) 配置系统时钟源并使能角速度传感器和加速度传感器

系统时钟源同样是通过电源管理寄存器 1(0X6B)来设置,该寄存器的最低三位用于 设置系统时钟源选择,默认值是 0(内部 8M RC 震荡),不过我们一般设置为 1,选择 x 轴陀螺 PLL 作为时钟源,以获得更高精度的时钟。同时,使能角速度传感器和加速度传感 器,这两个操作通过电源管理寄存器 2(0X6C)来设置,设置对应位为 0 即可开启。

至此,MPU6050 的初始化就完成了,可以正常工作了(其他未设置的寄存器全部采用 默认值即可),接下来,我们就可以读取相关寄存器,得到加速度传感器、角速度传感器和 温度传感器的数据了。不过,我们先简单介绍几个重要的寄存器。

首先,我们介绍电源管理寄存器 1,该寄存器地址为 0X6B,各位描述如图 13.4.2 所示:

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
6B	107	DEVICE _RESET	SLEEP	CYCLE	1.1	TEMP_DIS		CLKSEL[2:0]	

图 13.4.2

其中,DEVICE_RESET 位用来控制复位,设置为 1,复位 MPU6050,复位结束后, MPU硬件自动清零该位。SLEEEP 位用于控制 MPU6050 的工作模式,复位后,该位为 1, 即进入了睡眠模式(低功耗),所以我们要清零该位,以进入正常工作模式。TEMP_DIS 用



于设置是否使能温度传感器,设置为 0,则使能。最后 CLKSEL[2:0]用于选择系统时钟源,选择关系如表 13.4.1 所示:

CLKSEL[2:0]	时钟源
000	内部 8M RC 晶振
001	PLL, 使用 X 轴陀螺作为参考
010	PLL, 使用 Y 轴陀螺作为参考
011	PLL,使用 Z 轴陀螺作为参考
100	PLL, 使用外部 32.768Khz 作为参考
101	PLL, 使用外部 19.2Mhz 作为参考
110	保留
111	关闭时钟,保持时序产生电路复位状态

表 13.4.1

默认是使用内部 8M RC 晶振的,精度不高,所以我们一般选择 X/Y/Z 轴陀螺作为参考的 PLL 作为时钟源,一般设置 CLKSEL=001 即可。

接着,我们看陀螺仪配置寄存器,该寄存器地址为:0X1B,各位描述如图 13.4.3 所示:

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1B	27	XG_ST	YG_ST	ZG_ST	FS_S	EL[1:0]	-		-

图 13.4.3

该寄存器我们只关心 FS_SEL[1:0]这两个位,用于设置陀螺仪的满量程范围:0,±250°/S; 1,±500°/S; 2,±1000°/S; 3,±2000°/S; 我们一般设置为 3,即±2000°/S,因为陀螺仪的 ADC 为 16 位分辨率,所以得到灵敏度为: 65536/4000=16.4LSB/(°/S)。

接下来,我们看加速度传感器配置寄存器,寄存器地址为:0X1C,各位描述如图 13.4.4 所示:

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1C	28	XA_ST	YA_ST	ZA_ST	AFS_S	EL[1:0]		-	

图 13.4.4

该寄存器我们只关心 AFS_SEL[1:0]这两个位,用于设置加速度传感器的满量程范围: 0, ±2g; 1, ±4g; 2, ±8g; 3, ±16g; 我们一般设置为 0,即±2g,因为加速度传感器 的 ADC 也是 16 位,所以得到灵敏度为: 65536/4=16384LSB/g。

接下来,我看看 FIFO 使能寄存器,寄存器地址为:0X23,各位描述如图 13.4.5 所示:

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
23	35	TEMP_ FIFO_EN	XG_ FIFO_EN	YG_ FIFO_EN	ZG_ FIFO_EN	ACCEL _FIFO_EN	SLV2 _FIFO_EN	SLV1 _FIFO_EN	SLV0 _FIFO_EN

图 13.4.5

该寄存器用于控制 FIFO 使能,在简单读取传感器数据的时候,可以不用 FIFO,设置 对应位为 0 即可禁止 FIFO,设置为 1,则使能 FIFO。注意加速度传感器的 3 个轴,全



由 1 个位(ACCEL_FIFO_EN)控制,只要该位置 1,则加速度传感器的三个通道都开启 FIFO 了。

接下来,我们看陀螺仪采样率分频寄存器,寄存器地址为:0X19,各位描述如图 13.4.6 所示:

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
19	25		SMPLRT_DIV[7:0]						

图 13.4.6

该寄存器用于设置 MPU6050 的陀螺仪采样频率,计算公式为:

采样频率 = 陀螺仪输出频率 / (1+SMPLRT_DIV)

这里陀螺仪的输出频率,是 1Khz 或者 8Khz,与数字低通滤波器(DLPF)的设置有关,当 DLPF_CFG=0/7 的时候,频率为 8Khz,其他情况是 1Khz。而且 DLPF 滤波频率 一般设置为采样率的一半。采样率,我们假定设置为 50Hz,那么 SMPLRT_DIV=1000/50-1=19。

接下来,我们看配置寄存器,寄存器地址为: 0X1A,各位描述如图 13.4.7 所示:

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1A	26	-	-	EXT_SYNC_SET[2:0]		D	LPF_CFG[2:0]	

图 13.4.7

这里,我们主要关心数字低通滤波器(DLPF)的设置位,即:DLPF_CFG[2:0],加速 度计和陀螺仪,都是根据这三个位的配置进行过滤的。DLPF_CFG 不同配置对应的过滤情 况如表 13.4.2 所示:

	加速度	モ 传感器	1	角速度传感器				
DLPF_CFG[2:0]	Fs=	Fs=1Khz		(陀螺仪)				
	带宽(Hz)	延迟(ms)	带宽(Hz)	延迟(ms)	Fs (Khz)			
000	260	0	256	0.98	8			
001	184	2.0	188	1.9	1			
010	94	3.0	98	2.8	1			
011	44	4.9	42	4.8	1			
100	21	8.5	20	8.3	1			
101	10	13.8	10	13.4	1			
110	5	19.0	5	18.6	1			
111	伤	昭	保	留	8			

表 13.4.2

这里的加速度传感器,输出速率(Fs)固定是 1Khz,而角速度传感器的输出速率(Fs),则根据 DLPF_CFG 的配置有所不同。一般我们设置角速度传感器的带宽为其采样率的一半,如前面所说的,如果设置采样率为 50Hz,那么带宽就应该设置为 25Hz,取近似值 20Hz,就应该设置 DLPF_CFG=100。

接下来,我们看电源管理寄存器 2,寄存器地址为:0X6C,各位描述如图 13.4.8 所示:



Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
6C	108	LP_WAKE	CTRL[1:0]	STBY_XA	STBY_YA	STBY_ZA	STBY_XG	STBY_YG	STBY_ZG

图 13.4.8

该寄存器的 LP_WAKE_CTRL 用于控制低功耗时的唤醒频率,本章用不到。剩下的 6 位,分别控制加速度和陀螺仪的 x/y/z 轴是否进入待机模式,这里我们全部都不进入待机模式,所以全部设置为 0 即可。

接下来,我们看看陀螺仪数据输出寄存器,总共有 6 个寄存器组成,地址为:0X43~0X48, 通过读取这 6 个寄存器,就可以读到陀螺仪 x/y/z 轴的值,比如 x 轴的数据,可以通过读 取 0X43(高 8 位)和 0X44(低 8 位)寄存器得到,其他轴以此类推。

同样,加速度传感器数据输出寄存器,也有 6 个,地址为:0X3B~0X40,通过读取这 6 个寄存器,就可以读到加速度传感器 x/y/z 轴的值,比如读 x 轴的数据,可以通过读取 0X3B(高 8 位)和 0X3C(低 8 位)寄存器得到,其他轴以此类推。

最后,温度传感器的值,可以通过读取 0X41(高 8 位)和 0X42(低 8 位)寄存器得到, 温度换算公式为:

Temperature = 36.53 + regval/340

其中, Temperature 为计算得到的温度值, 单位为℃, regval 为从 0X41 和 0X42 读到的温度传感器值。

13.5. 【实验步骤】

第一步,打开实验箱,连接下载器。

- 第二步,确保下载器连接正确后,接通电源和 USB 线(J17)。
- 第三步,打开工程编译并下载程序。观察实验结果。

第四步,将 J63 和 J64 的跳帽对应到到六轴,如图 13.5.1。



图 13.5.1



13.6. 【实验现象】

可以看到 OLED 显示了 mpu6050 的温度、加速度值和角速度值,如图 13.6.1。



图 13.6.1



14. MPU6050-DMP 应用实验

14.1. 【实验内容】

编写程序,使用 MPU6050 的 DMP 计算俯仰、横滚、航向角。代码分别在仿真器控制下、 和脱离仿真环境运行。

14.2. 【实验目的】

掌握 IIC 使用方法 掌握查看原理图方法 掌握 mpu6050 的 DMP 使用方法

14.3. 【实验平台】

爱望实验箱

14.4. 【实验原理】

经过上一节节的介绍,我们可以读出 MPU6050 的加速度传感器和角速度传感器的原始数据。不过这些原始数据,对想搞四轴之类的初学者来说,用处不大,我们期望得到的是 姿态数据,也就是欧拉角:航向角(yaw)、横滚角(roll)和俯仰角(pitch)。有了这三 个角,我们就 可以得到当前四轴的姿态,这才是我们想要的结果。

要得到欧拉角数据,就得利用我们的原始数据,进行姿态融合解算,这个比较复杂,知 识点比较多,初学者不易掌握。而 MPU6050 自带了数字运动处理器,即 DMP,并且, InvenSense 提供了一个 MPU6050 的嵌入式运动驱动库,结合 MPU6050 的 DMP,可以将 我们的原始数据,直接转换成四元数输出,而得到四元数之后,就可以很方便的计算出欧拉 角,从而得到 yaw、roll 和 pitch。

使用内置的 DMP,大大简化了四轴的代码设计,且 MCU 不用进行姿态解算过程,大 大降低了 MCU 的负担,从而有更多的时间去处理其他事件,提高系统实时性。

使用 MPU6050 的 DMP 输出的四元数是 q30 格式的,也就是浮点数放大了 2 的 30 次方倍。在换算成欧拉角之前,必须先将其转换为浮点数,也就是除以 2 的 30 次方,然 后再进行计算,

计算公式为: q0=quat[0]/q30;//q30 格式转换为浮点数 q1=quat[1]/q30; q2=quat[2]/q30; q3=quat[3]/q30;

//计算得到俯仰角/横滚角/航向角



pitch=asin(-2*q1*q3+2*q0*q2)*57.3;//俯仰角 roll=atan2(2*q2*q3+2*q0*q1,-2*q1*q1-2*q2*q2+1)*57.3; //横滚角 yaw=atan2(2*(q1*q2+q0*q3),q0*q0+q1*q1-q2*q2-q3*q3)*57.3;//航向角

其中 quat[0]~ quat[3]是 MPU6050 的 DMP 解算后的四元数,q30 格式,所以要除以 一个 2 的 30 次方,其中 q30 是一个常量: 1073741824,即 2 的 30 次方,然后带入公 式,计算出欧拉角。上述计算公式的 57.3 是弧度转换为角度,即 180/π,这样得到的结果 就是以度(°)为单位的。关于四元数与欧拉角的公式推导,这里我们不进行讲解,感兴趣 的朋友,可以自行查阅相关资料学习。

14.5. 【实验步骤】

第一步,打开实验箱,连接下载器。

第二步,确保下载器连接正确后,接通电源和 USB 线(J17)。

第三步,打开工程编译并下载程序。观察实验结果。

第四步,将 J63 和 J64 的跳帽对应到到六轴,如图 14.5.1。



图 14.5.1

14.6. 【实验现象】

可以看到 OLED 显示了三个角度值,晃动一下实验箱,感受一下三个角度的变化,如图 14.6.1。





如图 14.6.1



15. BMP180 应用实验

15.1. 【实验内容】

编写程序,使用 BMP180 的测量当前环境下的压强、温度和海拔高度。代码分别在仿真器控制下、和脱离仿真环境运行。

15.2. 【实验目的】

掌握 IIC 使用方法 掌握查看原理图方法 掌握 BMP180 使用方法

15.3. 【实验平台】

爱望实验箱

15.4. 【实验原理】

BMP180 是一款数字气压传感器,由德国公司 Bosch Sensrtec 开发。它采用了微电机系统(MEMS)技术,可以测量大气压力和温度。该传感器采用低功耗设计,可通过 12C 总线与 微控制器或单片机通信。BMP180 具有高精度和快速响应的特点,适用于气压计、天气预报、 高度测量、室内导航等应用领域。它的工作温度范围广泛,从-40 摄氏度到+85 摄氏度。此 外, BMP180 还提供了温度补偿功能,可以减少温度对气压测量的影响。

BMP180 内包含有电阳式压力传感器、AD 转换器和控制单元,其中控制单元包括了 EEPROM 和 IIC 接口。读取 BMP180 时会直接传送没有经过补偿的温度值和压力值,而在 EEPROM 中则储存了 176 位单独的校准数据,这些数据将对读取的温度压力值进行补偿。176 位的 EEPROM 被划分为 11 人字,每个字 16 位,这样就包合有 11 个校准系数。每个器件模 块都有自己单独的校准系数,在每一次计算温度压力数据之前,单片机就应该先读出 EEPROM 中这些校准数据,然后再开始采集数据温度和压力数据。和所有的 IIC 总线器件一 样,BMP180 也有一个器件的固定地址,根据其数据手册,出厂时默认 BMP180 的从机地址 为 0xEE(写入方向),或 0xEF (读出方向)。温度数据 UT 和压力数据 UP 都存储在寄存器的第 0 到 15 位之中,压力数据 UP 的精度还可扩展至 16~19 位,如图 15.4.1。





图 15.4.1

要获得温度数据,必须先向控制寄存器(地址 0xF4)写 0x2E,然后等待至少 4.5ms,才可以从地址 0xF6 和 0xF7 读取十六位的温度数据。同样,要获得气压数据,必须先向控制寄存器(地址 0xF4)写 0x34 (0x74.0xB4、0xF4),然后等待至少 4.5ms,才可以从地址 0xF6 和 0xF7 读取 16 位的气压数据。如图 15.4.2 (OSS 为过采样分辨率):

Measurement	Control register value (register address 0xF4)	Max. conversion time [ms]
Temperature	0x2E	4.5
Pressure (oss = 0)	0x34	4.5
Pressure (oss = 1)	0x74	7.5
Pressure (oss = 2)	0xB4	13.5
Pressure (oss = 3)	0xF4	25.5



图 15.4.2

读取气压大致流程为:开始 ->读取 EEPROM 校准值 ->读取温度值 ->读取气压值 ->校准 海拔高度计算公式:

altitude = 44330 ×
$$\left(1 - \left(\frac{p}{p_0}\right)^{\frac{1}{5.255}}\right)$$

15.5. 【实验步骤】

第一步,打开实验箱,连接下载器。

第二步,确保下载器连接正确后,接通电源和 USB 线(J17)。

第三步,打开工程编译并下载程序。观察实验结果。

第四步,将 J63 和 J64 的跳帽对应到到气压,如图 15.5.1。



图 15.5.1

15.6. 【实验现象】

可以看到 OLED 显示了 BMP180 的压强、温度和海拔高度三个值,如图 15.6.1。





图 15.6.1



16. EEPROM 应用实验

16.1. 【实验内容】

编写程序,使用 AT24C02 存取数据。代码分别在仿真器控制下、和脱离仿真环境运行。

16.2. 【实验目的】

掌握 IIC 使用方法 掌握查看原理图方法 掌握 AT24C02 使用方法

16.3. 【实验平台】

爱望实验箱

16.4. 【实验原理】

EEPROM 芯片型号为 AT24C02。该芯片的总容量是 256 个字节,该芯片通过 IIC 总 线与外部连接,我们本章就通过 STM32 来实现 24C02 的读写,IIC 时序已在先前实验介 绍过了 AT24C02 的地址为 7 位高四位为 1010,低三位由 A0/A1/A2 的电平决定,除去这七位 外的一位 (R/W)用来选择读写方向其为最低位,它与地址无关。

EEPROM



图 16.4.1

16.5. 【实验步骤】

第一步,打开实验箱,连接下载器。

第二步,确保下载器连接正确后,接通电源和 USB 线(J17)。

第三步,打开工程编译并下载程序。观察实验结果。



16.6. 【实验现象】

可以看到测试值每隔 1.5s 自加 1,OLED,更新写入和读出的值,可以看到两个值是一致的,如图 16.6.1。



图 16.6.1



17. 电源检测实验

17.1. 【实验内容】

编写程序,使用 ADC 采样电源电压值。代码分别在仿真器控制下、和脱离仿真环境运行。

17.2. 【实验目的】

掌握 ADC 使用方法 掌握查看原理图方法 掌握测量大电压(超过 GPIO 的输入电压)的使用方法

17.3. 【实验平台】

爱望实验箱

17.4. 【实验原理】

STM32 拥有 1~3 个 ADC (STM32F101/102 系列只有 1 个 ADC),这些 ADC 可以独立使用,也可以使用双重模式(提高采样率)。STM32 的 ADC 是 12 位逐次逼近型的模拟数字转换器。它有 18 个通道,可测量 16 个外部和 2 个内部信号源。各通道的 A/D 转换可以单次、连续、扫描或间断模式执行。ADC 的结果可以左对齐或右对齐方式存储在 16 位数据寄存器中。 模拟看门狗特性允许应用程序检测输入电压是否超出用户定义的高/ 低阀值。

STM32F103 系列最少都拥有 2 个 ADC,我们选择的 STM32F103ZET 包含有 3 个 ADC。STM32 的 ADC 最大的转换速率为 1MHz,也就是转换时间为 1us(在 ADCCLK=14M,采样周期为 1.5 个 ADC 时钟下得到),不要让 ADC 的时钟超过 14M, 否则将导致结果准确度下降。

STM32 将 ADC 的转换分为 2 个通道组:规则通道组和注入通道组。规则通道相当 于你正常运行的程序,而注入通道呢,就相当于中断。在你程序正常执行的时候,中断是可 以打断你的执行的。同这个类似,注入通道的转换可以打断规则通道的转换,在注入通道 被转换完成之后,规则通道才得以继续转换。

通过一个形象的例子可以说明:假如你在家里的院子内放了 5 个温度探头,室内放了 3 个温度探头;你需要时刻监视室外温度即可,但偶尔你想看看室内的温度;因此你可以使用 规则通道组循环扫描室外的 5 个探头并显示 AD 转换结果,当你想看室内温度时,通过一 个按钮启动注入转换组(3 个室内探头)并暂时显示室内温度,当你放开这个按钮后,系统又 会回到规则通道组继续检测室外温度。从系统设计上,测量并显示室内温度的过程中断了测 量并显示室外温度的过程,但程序设计上可以在初始化阶段分别设置好不同的转换组,系统 运行中不必再变更循环转换的配置,从而达到两个任务互不干扰和快速切换的结果。可以设想一下,如果没有规则组和注入组的划分,当你按下按钮后,需要从新配置 AD 循环扫描的通道,然后在释放按钮后需再次配置 AD 循环扫描的通道。

上面的例子因为速度较慢,不能完全体现这样区分(规则通道组和注入通道组)的好处, 但在工业应用领域中有很多检测和监视探头需要较快地处理,这样对 AD 转换的分组将简 化事件处理的程序并提高事件处理的速度。

STM32 其 ADC 的规则通道组最多包含 16 个转换,而注入通道组最多包含 4 个通 道。关于这两个通道组的详细介绍,请参考《STM32 参考手册的》第 155 页,第 11 章。 STM32 的 ADC 可以进行很多种不同的转换模式,这些模式在《STM32 参考手册》的第 11 章也都有详细介绍,我们这里就不在一一列举了。我们本章仅介绍如何使用规则通道的单次 转换模式。

STM32 的 ADC 在单次转换模式下,只执行一次转换,该模式可以通过 ADC_CR2 寄存器的 ADON 位(只适用于规则通道)启动,也可以通过外部触发启动(适用于规则通道 和注入通道),这是 CONT 位为 0。

以规则通道为例,一旦所选择的通道转换完成,转换结果将被存在 ADC_DR 寄存器中, EOC(转换结束)标志将被置位,如果设置了 EOCIE,则会产生中断。然后 ADC 将停止, 直到下次启动。

下面介绍 ADC 的具体配置:

- 1) 开启 PA 口时钟和 ADC1 时钟,设置 PA1 为模拟输入。
- 2) 初始化 ADC1 参数,设置 ADC1 的校准、工作模式以及规则序列的相关信息。
- 3) 读取 ADC 值。

通过分压计算可以得到电源电压值(想一下为什么用两个1k的电阻,而不用1个2k的电阻), 如图 17.4.1:





图 17.4.1

17.5. 【实验步骤】

第一步,打开实验箱,连接下载器。

第二步,确保下载器连接正确后,接通电源。

第三步,打开工程编译并下载程序。观察实验结果。

17.6. 【实验现象】

可以看到 OLED 显示电源电压值,如图 17.6.1。





图 17.6.1



18. 光感检测实验

18.1. 【实验内容】

编写程序,使用 ADC 采样,获取光敏电阻的变化。代码分别在仿真器控制下、和脱离 仿真环境运行。

18.2. 【实验目的】

掌握 ADC 使用方法 掌握查看原理图方法 掌握光敏电阻的使用方法

18.3. 【实验平台】

爱望实验箱

18.4. 【实验原理】

光敏电阻为一种无结 (PN 结)器件,光敏特性主要利用了光导电体导电特性。 当光敏 电阻在受到光的照射时,如果光子能量大于本征半导体材料的禁带宽度,则价带中的电子吸 收一个光子后就足以跃迁到导带,并产生一个自由电子和一个自由空穴,从而使其导电性能 增加,电阻值下降。光照停止,自由电子与空穴逐渐复合,电阻又恢复原值。光敏电阻的电 阻值会随着入射光的强弱而发生改变;即当入射光较强时,光敏电阻的阻值会明显减小;当 入射光变弱时,光敏电阻的电阻会显著增大,如图 18.4.1。





图 18.4.1

18.5. 【实验步骤】

第一步,打开实验箱,连接下载器。

第二步,确保下载器连接正确后,接通电源。

第三步,打开工程编译并下载程序。观察实验结果。

18.6. 【实验现象】

用手遮挡光敏电阻的光线,可以看到 OLED 显示电压的变化,如图 18.6.1。





图 18.6.1



19. 声音检测实验

19.1. 【实验内容】

编写程序,使用 ADC 采样,获取声音检测的变化。代码分别在仿真器控制下、和脱离 仿真环境运行。

19.2. 【实验目的】

掌握 ADC 使用方法 掌握查看原理图方法 掌握声音传感器的使用方法

19.3. 【实验平台】

爱望实验箱

19.4. 【实验原理】

声音传感器的工作原理基于压电效应。当声波撞击传感器表面时,传感器内部的压电材 料产生微小的电荷变化,这种变化可被测量并转换为数字信号。此外,声音传感器还需要前 置放大器来增强电荷变化信号,并过滤掉其他类型的干扰信号。如图 19.4.1



图 19.4.1


第一步,打开实验箱,连接下载器。

- 第二步,确保下载器连接正确后,接通电源。
- 第三步,打开工程编译并下载程序。观察实验结果。

19.6. 【实验现象】



20. 滑动变阻器实验

20.1. 【实验内容】

编写程序,使用 ADC 采样,获取滑动变阻器的变化。代码分别在仿真器控制下、和脱 离仿真环境运行。

20.2. 【实验目的】

掌握 ADC 使用方法 掌握查看原理图方法 掌握滑动变阻器的使用方法

20.3. 【实验平台】

爱望实验箱

20.4. 【实验原理】

ADC 的原理在先前章节介绍了,下面介绍滑动变阻器的原理,它的工作原理是通过改变 接入电路部分电阻线的长度来改变电阻的,如图 20.4.1。









20.6. 【实验现象】



21. RTC 实验

21.1. 【实验内容】

编写程序,使用 RTC,在 OLED 显示日期和时间,实现一个简单的时钟。代码分别在仿 真器控制下、和脱离仿真环境运行。

21.2. 【实验目的】

掌握 RTC 使用方法 掌握查看原理图方法 掌握 OLED 的使用方法

21.3. 【实验平台】

爱望实验箱

21.4. 【实验原理】

STM32 的实时时钟(RTC)是一个独立的定时器。STM32 的 RTC 模块拥有一组连续 计数的计数器,在相应软件配置下,可提供时钟日历的功能。修改计数器的值可以重新设置 系统当前的时间和日期。

RTC 模块和时钟配置系统(RCC_BDCR 寄存器)是在后备区域,即在系统复位或从待机 模式唤醒后 RTC 的设置和时间维持不变。但是在系统复位后,会自动禁止访问后备寄存器 和 RTC,以防止对后备区域(BKP)的意外写操作。所以在要设置时间之前, 先要取消备份 区域(BKP)写保护。





图 21.4.1

RTC 由两个主要部分组成(参见图 21.4.1),第一部分(APB1 接口)用来和 APB1 总 线相连。此单元还包含一组 16 位寄存器,可通过 APB1 总线对其进行读写操作。APB1 接 口由 APB1 总线时钟驱动,用来与 APB1 总线连接。另一部分(RTC 核心)由一组可编程计 数器组成,分成两个主要模块。第一个模块是 RTC 的预分频模块,它可编程产生 1 秒的 RTC 时间基准 TR_CLK。RTC 的预分频模块包含了一个 20 位的可编程分频器(RTC 预分 频器)。如果在 RTC_CR 寄存器中设置了相应的允许位,则在每个 TR_CLK 周期中 RTC 产 生一个中断(秒中断)。第二个模块是一个 32 位的可编程计数器,可被初始化为当前的系统 时间,一个 32 位的时钟计数器,按秒钟计算,可以记录 4294967296 秒,约合 136 年左 右,作为一般应用,这已经是足够了的。

RTC 还有一个闹钟寄存器 RTC_ALR,用于产生闹钟。系统时间按 TR_CLK 周期累 加并与位,比较匹配时将产生一个闹钟中断。

RTC 内核完全独立于 RTC APB1 接口,而软件是通过 APB1 接口访问 RTC 的预分 频值、计数器值和闹钟值的。但是相关可读寄存器只在 RTC APB1 时钟进行重新同步的 RTC 时钟的上升沿被更新,RTC 标志也是如此。这就意味着,如果 APB1 接口刚刚被开 启之后,在第一次的内部寄存器更新之前,从 APB1 上读取的 RTC 寄存器值可能被破坏 了(通常读到 0)。因此,若在读取 RTC 寄存器曾经被禁止的 RTC APB1 接口,软件首 先必须等待 RTC_CRL 寄存器的 RSF 位(寄存器同步标志位,bit3)被硬件置 1。

RTC 正常工作的一般配置步骤如下:



- 1) 使能电源时钟和备份区域时钟。
- 2) 取消备份区写保护。
- 3) 设置 RTC 的分频,并初始化 RTC。

- 第一步,打开实验箱,连接下载器。
- 第二步,确保下载器连接正确后,接通电源。
- 第三步,打开工程编译并下载程序。观察实验结果。

21.6. 【实验现象】

可以看到 OLED 显示了时间,实现了一个简易时钟的功能。如图 21.6.1



图 21.6.1



22. 舵机实验

22.1. 【实验内容】

编写程序,使用 PWM 输出,控制舵机转动角度。代码分别在仿真器控制下、和脱离仿 真环境运行。

22.2. 【实验目的】

掌握 PWM 使用方法 掌握查看原理图方法 掌握舵机的使用方法

22.3. 【实验平台】

爱望实验箱

22.4. 【实验原理】

脉冲宽度调制(PWM),是英文 "Pulse Width Modulation"的缩写,简称脉宽调制,是利 用微处理器的数字输出来对模拟电路进行控制的一种非常有效的技术。简单一点,就是对脉 冲宽度的控制。

STM32 的定时器除了 TIM6 和 7。其他的定时器都可以用来产生 PWM 输出。其中 高级定时器 TIM1 和 TIM8 可以同时产生多达 7 路的 PWM 输出。而通用定时器也能同 时产生多达 4 路的 PWM 输出,这样,STM32 最多可以同时产生 30 路 PWM 输出!这 里我们仅利用 TIM3 的 CH3 产生一路 PWM 输出。如果要产生多路输出,大家可以根据 我们的代码稍作修改即可。

舵机主要是由外壳、电路板、驱动马达、减速器与位置检测元件所构成。其工作原理是 由接收机发出讯号给舵机,经由电路板上的 IC 驱动无核心马达开始转动,透过减速齿轮将 动力传至摆臂,同时由位置检测器送回讯号,判断是否已经到达定位。位置检测器其实就是 可变电阻,当舵机转动时电阻值也会随之改变,藉由检测电阻值便可知转动的角度。通过改 变脉冲的宽度和频率来改变电机的转速和转向。PWM 驱动是舵机常用的一种驱动方式,其电 路简单,控制精度高。舵机使用 PWM 方式控制时,需要将控制信号周期设定在 20ms,每个 周期内有一个脉冲,舵机会根据脉冲宽度转动到相应的角度位置,如图 22.4.1。







第一步,打开实验箱,连接下载器。

第二步,确保下载器连接正确后,接通电源。

第三步,打开工程编译并下载程序。观察实验结果。

22.6. 【实验现象】



23. NFC 应用实验

23.1. 【实验内容】

编写程序,使用 SPI,实现 NFC 卡的数据存取。代码分别在仿真器控制下、和脱离仿真环境运行。

23.2. 【实验目的】

掌握 SPI 使用方法 掌握查看原理图方法 掌握 NFC 存取数据的使用方法

23.3. 【实验平台】

爱望实验箱

23.4. 【实验原理】

SPI 是英语 Serial Peripheral interface 的缩写,顾名思义就是串行外围设备接口。是 Motorola 首先在其 MC68HCXX 系列处理器上定义的。SPI 接口主要应用在 EEPROM, FLASH,实时时钟,AD 转换器,还有数字信号处理器和数字信号解码器之间。SPI,是一 种高速的,全双工,同步的通信总线,并且在芯片的管脚上只占用四根线,节约了芯片的管 脚,同时为 PCB 的布局上节省空间,提供方便,正是出于这种简单易用的特性,现在越来 越多的芯片集成了这种通信协议,STM32 也有 SPI 接口。下面我们看看 SPI 的内部简明 图 24.4.1:



图 24.4.1



SPI 接口一般使用 4 条线通信: MISO 主设备数据输入,从设备数据输出。 MOSI 主设备数据输出,从设备数据输入。 SCLK 时钟信号,由主设备产生。 CS 从设备片选信号,由主设备控制。

从图中可以看出, 主机和从机都有一个串行移位寄存器, 主机通过向它的 SPI 串行寄存器写入一个字节来发起一次传输。寄存器通过 MOSI 信号线将字节传送给从机, 从机也将自己的移位寄存器中的内容通过 MISO 信号线返回给主机。这样, 两个移位寄存器中的内容就被交换。 外设的写操作和读操作是同步完成的。如果只进行写操作, 主机只需忽略接收到的字节; 反之, 若主机要读取从机的一个字节, 就必须发送一个空字节来引发从机的传输。

SPI 主要特点有:可以同时发出和接收串行数据;可以当作主机或从机工作;提供频率 可编程时钟;发送结束中断标志;写冲突保护;总线竞争保护等。

SPI 总线四种工作方式 SPI 模块为了和外设进行数据交换,根据外设工作要求,其输出串行同步时钟极性和相位可以进行配置,时钟极性(CPOL)对传输协议没有重大的影响。如果 CPOL=0,串行同步时钟的空闲状态为低电平;如果 CPOL=1,串行同步时钟的空闲状态为高电平。时钟相位(CPHA)能够配置用于选择两种不同的传输协议之一进行数据传输。如果 CPHA=0,在串行同步时钟的第一个跳变沿(上升或下降)数据被采样;如果 CPHA=1,在串行同步时钟的第二个跳变沿(上升或下降)数据被采样。SPI 主模块和与之通信的外设备时钟相位和极性应该一致。

本节实验使用的 RFID 硬件原理图如图 24.4.2:



RFID



图 24.4.2

MF RC522 是应用于 13.56MHz 非接触式通信中高集成度读写卡系列芯片中的一员。是
NXP 公司针对"三表"应用推出的一款低 电压、低成本、体积小的非接触式读写卡芯片,
是智能仪表和便携式手持设备研发的较好选择。容量为 8K 位 EEPROM,分为 16 个扇区,每
个扇区为 4 块,每块 16 个字节,以块为存取单位,每个扇区有独立的一组密码及访问控制。
卡分为 16 个扇区,每个扇区由 4 块(块 0、块 1、块 2、块 3)组成,(我们也将 16
个扇区的 64 个块按绝对地址编号为 0~63,存贮结构如下图所示:

	块 0				数据块	0
扇区 0	块 1				数据块	1
	块 2				数据块	2
	块 3	密码A	存取控制	密码 B	控制块	3
	块 0				数据块	4
扇区1	块1				数据块	5
	块 2				数据块	6
	块 3	密码A	存取控制	密码 B	控制块	7



			:			
			:			
			:			
	0				数据块	60
扇区 15	1				数据块	61
	2				数据块	62
	3	密码A	存取控制	密码 B	控制块	63

1、 第0扇区的块0(即绝对地址0块),它用于存放厂商代码,已经固化,不可更改。

2、 每个扇区的块 0、块 1、块 2 为数据块,可用于存贮数据。

数据块可作两种应用:用作一般的数据保存,可以进行读、写操作;用作数据值,可以进行初始化值、加值、减值、读值操作。 时期去与读写器的通讯如图 24 4 2

射频卡与读写器的通讯如图 24.4.3:



图 24.4.3



第一步,打开实验箱,连接下载器。

第二步,确保下载器连接正确后,接通电源。

第三步,打开工程编译并下载程序。观察实验结果。

23.6. 【实验现象】

连接好 USB 线,电脑打开串口工具并配置好串口参数。将 IC 卡,靠近读写器,可以看 到串口工具打印的信息,如图 24.6.1。现一想,为什么第一次靠近的时候为什么并没有写入 成功(程序代码是写入块地址 16,写入数据每次自加)?



图 24.6.1



综合实验

24. 楼道灯控实验

24.1. 【实验内容】

编写程序,使用声音检测和人体感应,控制继电器的开关,从而控制灯泡的亮灭。代码 分别在仿真器控制下、和脱离仿真环境运行。

24.2. 【实验目的】

掌握人体感应模块使用方法 掌握声音传感器方法 掌握继电器的使用方法

24.3. 【实验平台】

爱望实验箱

24.4. 【实验原理】

在基础实验 9 和 19 中,我们分别用人体感应模块声音检测模块做了对应的实验,本节 实验就是使用这两个模块,当检测到人体移动(图 26.4.1)或者检测到声音(图 26.4.2), 就打开继电器,3s 后再关闭继电器(图 26.4.3)。



图 26.4.1



图 26.4.2



图 26.4.3

第一步,打开实验箱,连接下载器。

第二步,确保下载器连接正确后,接通电源。

第三步,打开工程编译并下载程序。观察实验结果。

24.6. 【实验现象】

当用手在人体感应模块上挥动或者发出声音,可以听到继电器打开,同时 LED27 亮,3s



后继电器关闭,LED27 灭。



25. 密码锁实验

25.1. 【实验内容】

编写程序,使用按键设置密码,控制继电器的开关。代码分别在仿真器控制下、和脱离 仿真环境运行。

25.2. 【实验目的】

掌握按键使用方法 掌握查看原理图方法 掌握 OLED 的使用方法 掌握继电器控制的使用方法

25.3. 【实验平台】

爱望实验箱

25.4. 【实验原理】

密码锁是锁的一种,开启时用的是一系列的数字或符号,文字密码锁可分为:机械密码 锁、数字密码锁等等。密码锁的密码通常都只是排列而非真正的组合。本节实验就是利用基 础实验的按键功能、OLED 显示和继电器模块。当输入密码与设置的密码一致时,则打开继 电器(如图 27.4.1)。



图 27.4.1



第一步,打开实验箱,连接下载器。

第二步,确保下载器连接正确后,接通电源。

第三步,打开工程编译并下载程序。观察实验结果。

25.6. 【实验现象】

上电开始,需要设置密码,使用 SW2、SW4、SW6,SW2 和 SW6 为数字加减键,SW4 为数字确认键,总共 6 个数字,如图 27.6.1。当设置好后,OLED 会显示 rtc 时间,如图 27.6.2,当需要输入密码时,需要按下 SW4 进入输入模式,然后依次输入 6 个数字的密码,当输入密码正确,继电器会打开,同时 LED27 会亮,如图 27.6.3。



图 27.6.1



图 27.6.2





图 27.6.3



26. 环境监测实验

26.1. 【实验内容】

编写程序,使用 SHT20、BMP180 和光敏电阻,监测环境的各项数据。代码分别在仿真器控制下、和脱离仿真环境运行。

26.2. 【实验目的】

掌握 SHT20 使用方法 掌握 BMP180 使用方法 掌握查看原理图方法 掌握直流风扇的使用方法

26.3. 【实验平台】

爱望实验箱

26.4. 【实验原理】

在基础实验,分别使用 SHT20 测量温湿度,BMP180 传感器测量气压和海拔高度,光敏 电阻表征环境光度,其中原理在这里就不再赘述。本节实验使用各种传感器测量环境数据, 并在 OLED 上显示,温度超过 25℃开启风扇,低于 23℃关闭风扇,如图 29.4.1。



图 29.4.1

26.5. 【实验步骤】

第一步,打开实验箱,连接下载器。

第二步,确保下载器连接正确后,接通电源。



第三步,打开工程编译并下载程序。观察实验结果。

26.6. 【实验现象】

需要将J64的温湿度和气压都接上短接帽,如图29.6.1,可以看到各项数据已显示到OLED 显示屏,如图29.6.2,将 SHT20 加热到25 度,可以看到风扇转动。



图 29.6.1



图 29.6.2



27. 公交卡实验

27.1. 【实验内容】

编写程序,使用 NFC 卡和读卡器实现一个公交卡充值和扣费的实验。代码分别在仿真器控制下、和脱离仿真环境运行。

27.2. 【实验目的】

掌握 NFC 卡使用方法 掌握 OLED 使用方法 掌握查看原理图方法

27.3. 【实验平台】

爱望实验箱

27.4. 【实验原理】

本节实验所使用的的模块包括 RFID、OLED、按键、蜂鸣器,这些模块的原理已经在基础实验介绍过了,这里就不再赘述。该实验主要实现一个公交卡的功能,初次充值金额,IC 卡刷一次,每次扣费2元。

27.5. 【实验步骤】

第一步,打开实验箱,连接下载器。 第二步,确保下载器连接正确后,接通电源。 第三步,打开工程编译并下载程序。观察实验结果。

27.6. 【实验现象】

上电后将 IC 卡放到读卡器位置, 空闲状态按 OK 键充值, 上下键设置金额, 如图 30.6.1。 刷一次扣费 2 元并蜂鸣器鸣叫提醒, 余额不足则提示并鸣叫 2 次。按 OK 键充值, 充值金额 最多 999 元。





图 30.6.1



28. 闹钟实验

28.1. 【实验内容】

编写程序,使用 RTC 实现一个闹钟功能。代码分别在仿真器控制下、和脱离仿真环境运行。

28.2. 【实验目的】

掌握 RTC 使用方法 掌握 OLED 使用方法 掌握查看原理图方法

28.3. 【实验平台】

爱望实验箱

28.4. 【实验原理】

在基础实验讲过 RTC 的闹钟功能,本次就是利用 stm32 内部的 rtc 闹钟功能,当设置的 闹钟时间与 RTC 时间重合,蜂鸣器就会响。

28.5. 【实验步骤】

28.6. 【实验现象】