摘 要

具体地讲就是研究工作的主要对象和范围， 采用的手段和方法， 得出的结果和重要的结论， 有时也包括具有情报价值的其它重要的信息。 一般字数在200-400字之间

https://www.zhihu.com/question/20028088

本部分正文结束。

**关键字：**关键字1，关键字2，关键字3

ABSTRACT

英文摘要正文

英文摘要结束.

**Key Words:** 英文关键字1,英文关键字2,英文关键字3

目录

[第1章 课题整体框架 1](#_Toc29413)

[1.1 课题任务 1](#_Toc19877)

[1.2 课题要求 1](#_Toc20859)

[1.3 研究意义 1](#_Toc14954)

[第2章 硬件部分和软件部分 2](#_Toc20533)

[2.1 硬件部分 2](#_Toc21308)

[2.1.1 电源模块 2](#_Toc6751)

[2.1.2 最小系统开发板 2](#_Toc15363)

[2.1.3 寻迹红外模块 2](#_Toc31454)

[2.1.4 前轮舵机 2](#_Toc30759)

[2.1.5 TB6612FNG模块说明 5](#_Toc9006)

[2.1.6 0.96OLED显示屏 6](#_Toc2078)

[2.1.7 方式 8](#_Toc31538)

[2.1.8 规范化发 8](#_Toc123)

[2.1.9 规范化 8](#_Toc18558)

[2.2 软件部分 9](#_Toc20110)

[2.2.1 软件工具keil 9](#_Toc12552)

[2.2.2 Oled显示屏专用取模工具 9](#_Toc16072)

[2.2.3 Cubmax 10](#_Toc31852)

[2.2.4 Git 10](#_Toc15237)

[2.2.5 模块流程图 10](#_Toc20506)

[第3章 实现功能与电路设计 11](#_Toc12014)

[3.1 实现功能与设计 11](#_Toc17961)

[3.1.1 实现功能描述 11](#_Toc2311)

[3.2 电路设计 11](#_Toc31660)

[3.2.1 最小系统板电路 11](#_Toc13105)

[3.2.2 红外检测模块电路 12](#_Toc23222)

[3.2.3 TB6612带稳压模块板原理图 13](#_Toc17566)

[3.2.4 0.96OLED显示屏原理图 14](#_Toc17775)

[3.3 软件设计 15](#_Toc10127)

[3.3.1 软件设计思路 15](#_Toc13035)

[第4章 调试难点和问题 16](#_Toc7504)

[4.1 调试中遇到的重点与难点 16](#_Toc5028)

[4.2 解决方案 16](#_Toc16208)

[4.3 实现展示（附上仿真图或实物照片） 16](#_Toc17086)

[第5章 总结 17](#_Toc19918)

[参考文献 18](#_Toc12932)

[想对老师说的话 19](#_Toc24800)

[附录 20](#_Toc13762)

[附录一：06年11月CD电信新华营业厅日缴费顾客半小时到达数据 20](#_Toc30446)

[（略） 20](#_Toc5344)

[附录二：顾客调查问卷 20](#_Toc29658)

[（略） 20](#_Toc21742)

1. 课题整体框架
   1. 课题任务

用STM32f103c8t6作为微控制器，设计一款智能小车。小车能实现自动巡航，可前进后退，转弯。小车具有手动控制模式，能使用远程设备解除智能小车的自动巡航。实现手动控制小车的行进。

* 1. 课题要求

自动巡航的跑道采用黑色线纸组成的一圈半径为25cm环形跑道，小车在自动巡航的时候无需外部控制，可完成一圈巡航。手动控制器与小车分离，且采用合适的无线通信技术跟小车进行通信。小车可以测速，并通过速度反馈控制电机的驱动能力，使速度恒定。小车的转向模拟真实的车辆前轮左右转向。手动控制器需要屏幕显示。且输入方式不可以使用按键输入。

* 1. 研究意义

自动巡航和无人驾驶技术一直是热门的研究技术，前人已经提出了许多优秀的检测方法和算法去实现小车的智能驾驶。但是总体的解决方案价格都是价格高昂的。在某些运用场景上，这些价格高昂的技术就显得十分的鸡肋。所以，本课题在于研究如何利用更低廉的单片机芯片去实现些简易，普通的自动巡航技术在简单赛道上的运用。并结合手动控制来完善自动巡航驾驶的不足，解决无人驾驶潜在的风险和威胁。

1. 硬件部分和软件部分
   1. 硬件部分
      1. 电源模块

小车的电源模块，采用了可充放电的12v锂电池供电，并采用LM2596S DC-DC直流可调的降压稳压电源模块板进行降压处理成3.3v。手柄的电源模块采用了可充放电的3.7v锂电池供电，同样配置了降压模块进行降压。

* + 1. 最小系统开发板

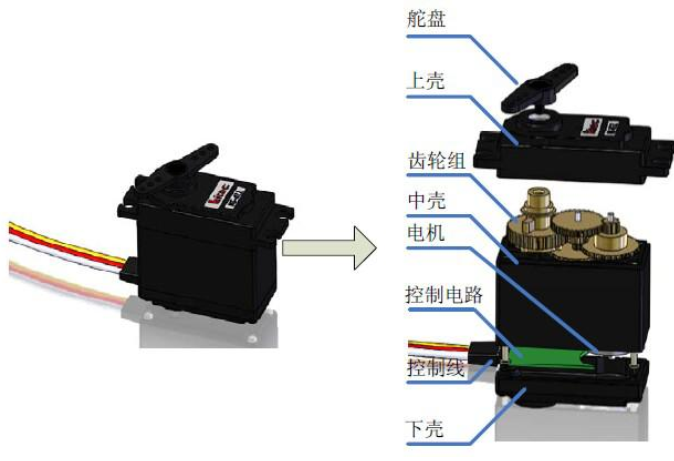
小车和手柄的采用了基于stm32f103c8t6的最小系统控制板。基于ARM Cortex-M 内核STM32系列的三十二位的微控制器嵌入式操作系统，拥有从六万四千或十二万八千字节的闪存flash程序存储器，高达两万字节的SRAM。工作电压两伏到三点六伏，温度区间为负四十摄氏度到八十五摄氏度。拥有最高七十二MHz的工作频率，在存储器的等待周期访问时可达五分之四DMips/MHz。拥有两个I方C接口(支持SMBus和PMBus)。三个USART串口通信接口。两个个SPI通信接口(十八兆比特每秒)。

* + 1. 寻迹红外模块

在小车的前端配置了左右两个红外传感器模块。该红外传感器模块对环境光线的抗干扰能力强，拥有一对红外线发射装置与接收装置，发射装置发射出某一波段的红外线，当发射红外线方向遇到拥有反射红外线能力的反射面时，红外线被反射回来，被接收设备接收，然后通过比较器LM393电路进行信号处理之后，接受信号的LED灯会亮起。同时，输出接口OUT引脚输出一个数字低电平信号。拥有电位器旋钮调节按钮，方便调节红外检测距离，有效距离范围是二到三十厘米，工作电压区间为三点三伏到五伏。因为模块发射端会主动发射红外线去检测,所以影响红外探测距离的重要因素是探测对象的颜色红外反射率和形状。其中黑色对红外线的吸收能力最强。红外探测距离因此会减小。相反白色是反射能力最强的。面积不大的对象的探测距离小,大面积的对象的探测距离大。输出端口的OUT引脚可以直接驱动一个五伏的继电器，也能直接和MCU的IO口引脚连接，；连接方式：VCC-VCC;GND-GND;OUT-IO。电路板尺寸为3.2CM\*1.4CM。

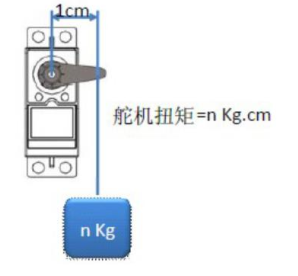
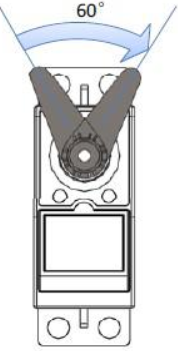
* + 1. 前轮舵机

舵机根据其控制电机的方式可以称为微型伺服马达。因为它早期在模型上主要用于控制模型的舵面，所以市面上又被称为舵机。舵机能够接收一个简易的操作命令，便能够自动旋转到操作命令对应的较为精准的位置，所以十分方便在关节型结构机器产品中运用。可以较为准确的操控和维持输出轴的转动角度。直流电机这样的控制方式又叫做闭环控制，所以舵机更准确的名称为伺服马达，英文名servo。



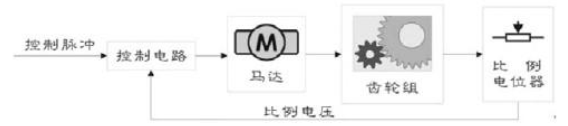
舵机的结构

舵机的主要构成如图 2-1 所示，分别为舵盘、上壳、减速齿轮组、中壳、电机、控制电路、控制线、下壳。舵机的外壳普遍是高分子聚合物材质的，特殊的可能会有金属铝合金材质的外壳。金属外壳能更牢靠的固定在目标地点。伺服马达的齿轮箱的种类有：塑料、混合、金属等差别。塑料的噪音小，成本底，但是强度一般不大；金属的强度虽然很高，但是成本也很高，如果装配的精度不行，就会产生很大的噪音。小扭矩的微舵，扭矩虽然很大，一般所有用塑料齿轮的伺服马达功率密度小，如辉盛的九克金属齿轮微舵主要用在功率密度很高的伺服马达上，比如辉盛的MG995伺服马达却能提供十三千克的扭矩。Hitec甚至用钛合金作为齿轮材料，其强度能提供二十几公斤的扭矩。混合类型的在塑料和金属之间做了平衡，在扭矩不大的部位，如电机输出减速箱的地方用塑料齿轮。

伺服马达的规格和选型：1伺服马达转速，伺服马达无负载的条件下，转速在转过六十度角时，所需时间的长短。伺服马达常见的速度一般在零点零一秒到零点二一秒每六十度之间。2伺服马达转矩，伺服马达扭矩的单位是千克每厘米。

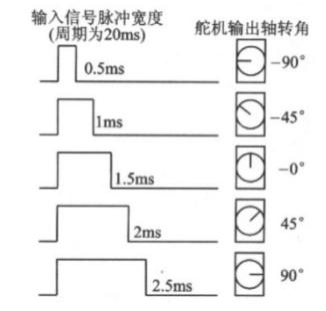
舵机的转速和转矩

舵机的工作理论：舵机是一个微型的伺服控制系统，具体的控制原理可以用下图表示。



舵机的控制原理

通过控制电路接收信号源的控制信号，并驱动电机转动是基本的运行原理；电机的速度被齿轮组成倍数的缩小，电机的输出扭矩则会放大响应的倍数，然后输出给外部；齿轮组跟电位器的最后端口会同时旋转，通过舵机轴的旋转角度的测量；电路板会进行检测，并且依照电位器去判断舵机旋转的角度，最后再操控舵机旋转到对象角度或者停留在对象角度。舵机需要一个外部的控制终端产生的脉宽调制信号去控制舵机旋转角度，舵机脉冲的宽度是控制终端必要的编码信息。舵机的控制脉冲一般周期是二十毫秒，脉宽从零点五毫秒到二点五毫秒，分别对应着旋转到负九十度和正九十度的位置。如下图所示：

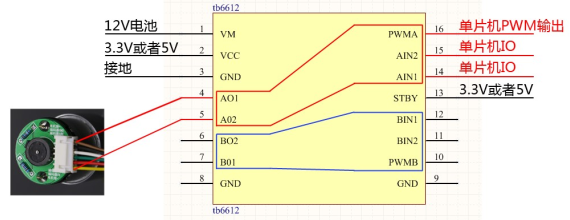
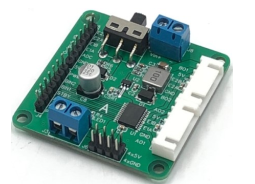


舵机信号脉冲示意图

* + 1. TB6612FNG模块说明

TB6612FNG对比L298N，它无需额外的散热片，拥有更好的热耗性。外围电路比二极管续流电路更简单。增加了电源的滤波电容便能够直接驱动电机，便于降低系统的尺寸。PWM信号的输入频率高达十万赫兹，能够满足绝大部分的设计要求。

带稳压模块版说明：新增五伏稳压电路，支持五伏两安的输出。共有四个这样的引脚可以对外部供电。增加电压测量电路，通过串联一个一万欧姆和一个一千欧姆的电阻，对输入电源进行十一分之一的分压后，可以通过ADC进行采集并计算得到电源电压进行监控。引出电机标准的六个引脚接口，可以直接用排线连接，AB相编码器的信号接到单独的引脚输出。新增电源输出电路，可输出与输入的电源保持一致的电源。含有电源开关，可以对板子供电进行开启和关闭的操作。



TB6612FNG实物图和引脚连接示意图

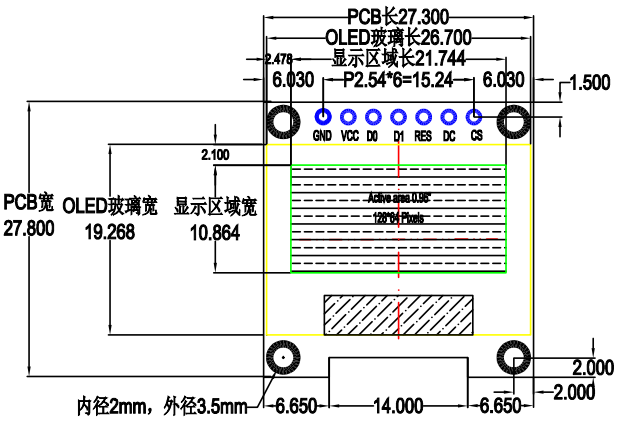
必须要有PWM波输入才有AO1和AO2的信号，只接AIN1和AIN2不会产生AO1和AO2的信号。VM引脚接电源正极，VCC引脚接内部的逻辑供电，模块的GND接电源地，一个接单片机地，STBY引脚置高，模块才能正常运行。上图中蓝色部分的五个引脚控制一路马达，红色部分的控制另外一路马达，AO1和AO2分别接到马达的正极和负极。然后通过PWMA、AIN2、AIN1控制电机。其中 PWMA接到单片机的PWM引脚，一般一万赫兹的PWM即可，并通过改变占空比来调节电机的速度。B路的使用是相同的。下面是真值表：

TB6612FNG输出真值表

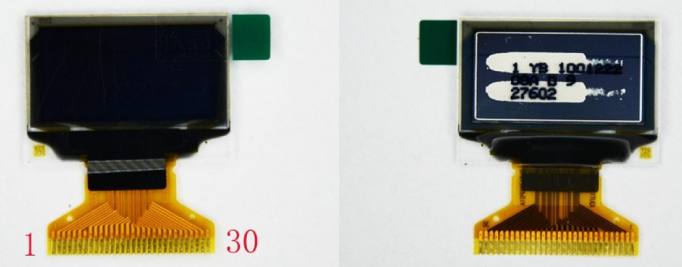
|  |  |  |
| --- | --- | --- |
| AIN1 | 0 | 0 |
| AIN2 | 0 | 1 |
|  | 停止 | 反转 |

* + 1. 0.96OLED显示屏

OLED（Organic Light Emitting Diode），又称作有机发光二极管。LCD都需要背光，而OLED不需要，因为它是自发光的。因此同样的显示，OLED效果要好一些。在此我们使用的是中景园电子的0.96寸OLED黄蓝显示屏，该屏有以下特点：其中黄蓝是屏上1/4部分为黄光，下3/4为蓝；而且是固定区域显示固定颜色，颜色和显示区域均不能修改；白光则为纯白，也就是黑底白字；蓝色则为纯蓝，也就是黑底蓝字。分辨率为128\*64。多种接口方式，通过屏上的 BS0~BS2 来配置。

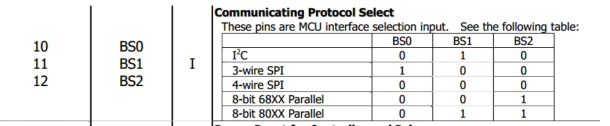


裸屏为 30pin，从屏正面看左下角为 1，右下角为 30；如下图所示：



0.96寸OLED裸屏外观

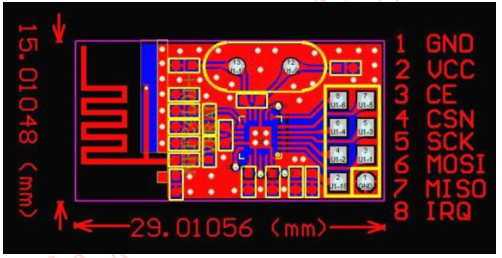
0.96寸OLED驱动IC为 SSD1306；其具有内部升压功能；所以在设计的时候不需要再专一设计升压电路；SSD1306总共八页，每页拥有一百二十八个字节，刚好可以对应128\*64的点阵大小。七针SPI\_OLED有七个引脚，分别为VCC、D0、GDN、RES、DC、D1、CS。由OLED的数据手册我们可以知道 0.96 寸 OLED 裸屏是支持四种五种不同接口的。



通信接口是通过 BS0,BS1,BS2 三个管脚来配置的。

* + 1. NRF24L01 无线模块简介

NRF24L01无线模块采用NRF24L01芯片，该模块拥有以下特征：一百二十五个频道可以选择，满足调频通信和多点通信的需要。免许可证使用的2.4G全球开放的ISM频段。内置CRC检错和点对多点的通信地址控制。高效的GFSK调制技术，两兆bps为最高工作速率，抗干扰能力强。低工作电压区间1.9到3.6V。可设置自动应答，确保数据可靠传输。所以在后面软件编程的时候 SPI 速度不能高于这个最大值。已经被大量使用，目前各方面表现都十分不错。该模块的外形和引脚图如图 2-8 所示：



NRF24L01 模块外观引脚图

nRF24L01可以设置为以下几种主要的模式：

nRF24L01的模式设置

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 模式 | PWR\_UP | PRIM\_RX | CE | FIFO寄存器状态 |
| 接受模式 | 1 | 1 | 1 | - |
| 发送模式 | 1 | 0 | 1 | 数据在TX FIFO寄存器中 |
| 发送模式 | 1 | 0 | 1-0 | 停留在发送模式，直到数据发送完 |
| 待机模式2 | 1 | 0 | 1 | TX FIFO 为空 |
| 待机模式1 | 1 | - | 0 | 无数据传输 |
| 掉电模式 | 0 | - | - | - |

nRF24L01在不同模式下的引脚功能

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 引脚名称 | 方向 | 发送模式 | 接受模式 | 待机模式 | 掉电模式 |
| CE | 输入 | 高电平>10us | 高电平 | 低电平 | - |
| CSN | 输入 | SPI片选使能，低电平使能 | | | |
| SCK | 输入 | SPI时钟 | | | |
| MOSI | 输入 | SPI串行输入 | | | |
| MISO | 三态输出 | SPI串行输出 | | | |
| IRQ | 输出 | 中断，低电平使能 | | | |

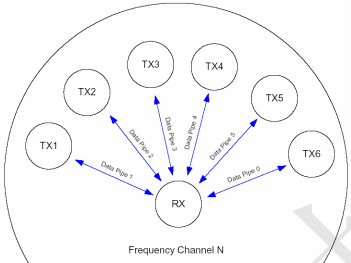
模式的介绍

|  |  |
| --- | --- |
| 待机模式 | 待机模式A可以在保证快速启动的同时并减少系统平均消耗的电流。晶振正常工作。当发送端FIFO寄存器为空并且CE为高电平时进入待机模式B。待机模式B下部分时钟缓冲器处在工作模式。寄存器配置字内容保持不变。 |
| 掉电模式 | 掉电模式由寄存器中PWR\_UP位来控制，进入后，nRF24L01停止工作，各功能关闭，电流消耗最小，但寄存器内容保持不变。 |

nRF24L01的数据包处理方式有：1.ShockBurstTM。2.增强型ShockBurstTM 模式。

ShockBurstTM模式：ShockBurst模式下能够与成本较低的低速单片机连接。该模式减小了通信的平均消耗电流，因为在与单片机低速通信的同时而在无线部分高速通信。在ShockBurstTM发送模式下，会自动生成前导码及 CRC校验。数据发送完毕后，IRQ中断会通知单片机。降低了单片机的查询时间，降低了单片机的工作量并且也降低了软件的开发时间。内部各有三个不同的 RX\_FIFO与TX\_FIFO寄存器。

增强型的ShockBurstTM模式：能够让双向链接协议运行更加简单、高效。经典的双向链接协议为：发送对象规定终端设备在得到数据信号后有应答信号，让发送对象检验是否有数据的丢失。如果数据发生丢失，便重新发送将丢失的数据恢复。该模式无需增加其它工作量便能够同时控制重发功能和应答。

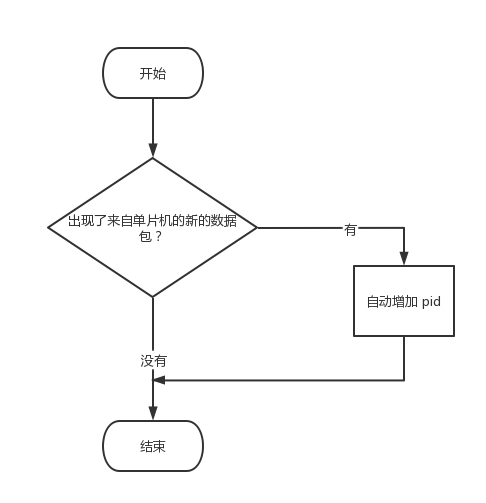


nRF2402在星形网络中的结构图

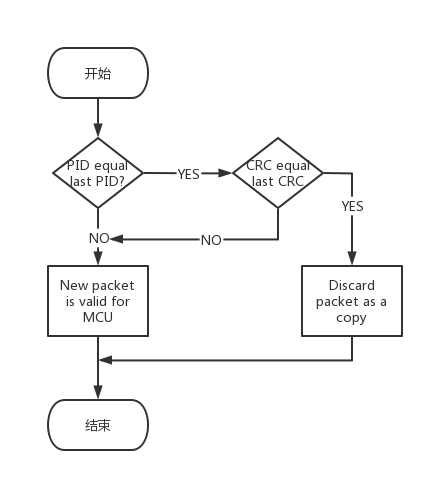
在接收模式下nRF24L01能够接收六路不同通道的数据，见图2-9。每一个通道都可以设置不同的通信地址，但是共用相同的频道。一个设置为接收模式的设备后能够与六个不同的设置为发送模式设备同时进行通讯。而设置为接收模式的设备能够对六个发射端进行身份识别。一到五的通道都拥有八位自身地址和三十二位公用地址。所有的通道都能够设置为增强型ShockBurst模式。通道零是唯一一个能够配置为 四十位自身地址的通道。在发送端，通道零被用做接收应答信号的通道。如果重发次数超过了设定值，则会产生中断。

在发送对象每次从单片机取得一包新数据后数据包识别值加一。

接收方：接收方对新接收数据包的数据包值与上一包进行比较。如果数据包值不同，则认为接收的数据包是新数据包。发送方：每发送一包新的数据则发送方的数据包值加一。

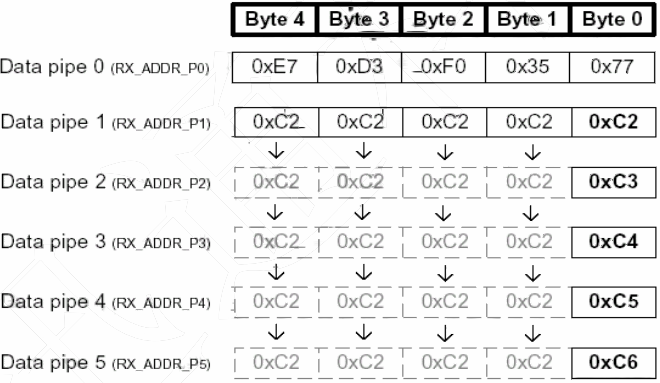


PID的生成



PID的检测

通过SPI接口进行配置CRC校验的长度。接收数据包的附加要求是CRC的校验错误，便不会接收数据包。nRF24L01能够接收六路不同的地址相同频率的数据。每个数据通道都拥有自己的地址，并能够通过寄存器来分别配置。数据通道是通过寄存器配置的，初始状态下只有通道零和数据通道一是开启状态的。一般情况下不会允许不同的通道设置一样的地址。通道0有40位可配置地址。数据通道1~5的地址为：32位共用地址+各自的地址。



通道0-5的地址设置

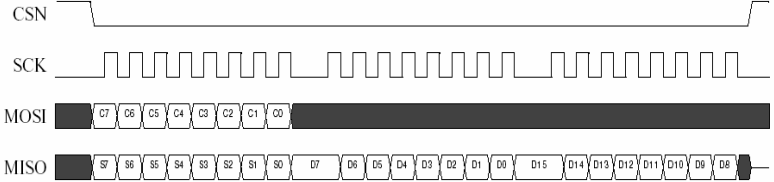
图 2-12 所示的是通道一到五的地址设置方法举例。全部通道可以设置为多达四十位，但是一到五通道的最低位必须不同。

所有寄存器都是通过SPI口进行配置的。SPI 指令格式：命令字+数据字节。

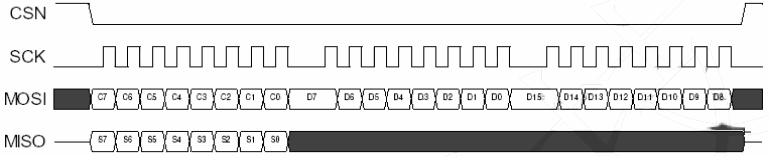
nRF24L01 SPI串行口指令设置

|  |  |  |
| --- | --- | --- |
| 指令名称 | 指令格式 | 操作 |
| R\_REGISTER | 000A AAAA | 读配置寄存器。 AAAAA 指出读操作的寄存器地址 |
| W\_REGISTER | 001A AAAA | 写配置寄存器。 AAAAA 指出写操作的寄存器地址。只有在掉电模式和待机模式下可操作。 |
| R\_RX\_PAYLOAD | 0110 0001 | 读 RX 有效数据： 1-32 字节。读操作全部从字节 0 开始。当读 RX有效数据完成后， FIFO 寄存器中有效数据被清除。应用于接收模式下。 |
| W\_RX\_PAYLOAD | 1010 0000 | 写 TX 有效数据： 1-32 字节。写操作从字节 0 开始。应用于发射模式下。 |
| FLUSH\_TX | 1110 0001 | 清除 TX FIFO 寄存器，应用于发射模式下。 |
| FLUSH\_RX | 1110 0010 | 清除 RX FIFO 寄存器，应用于接收模式下。在传输应答信号过程中不应执行此指令。也就是说，若传输应答信号过程中执行此指令的话将使得应答信号不能被完整的传输。 |
| REUSE\_TX\_PL | 1110 0011 | 重新使用上一包有效数据。当 CE 为高过程中，数据包被不断的重新发射。在发射数据包过程中必须禁止数据包重利用功能。 |
| NOP | 1111 1111 | 空操作。可以用来读状态寄存器。 |

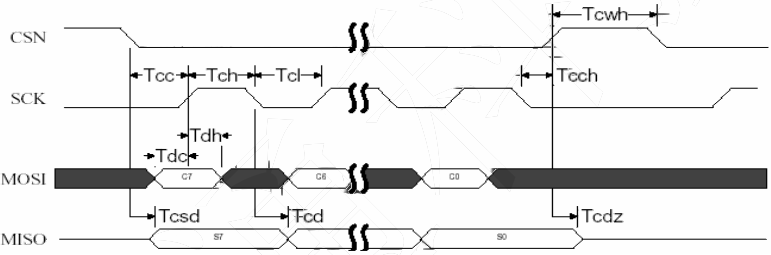
当访问多字节寄存器时首先要读/写的是最低字节的高位。



SPI读操作



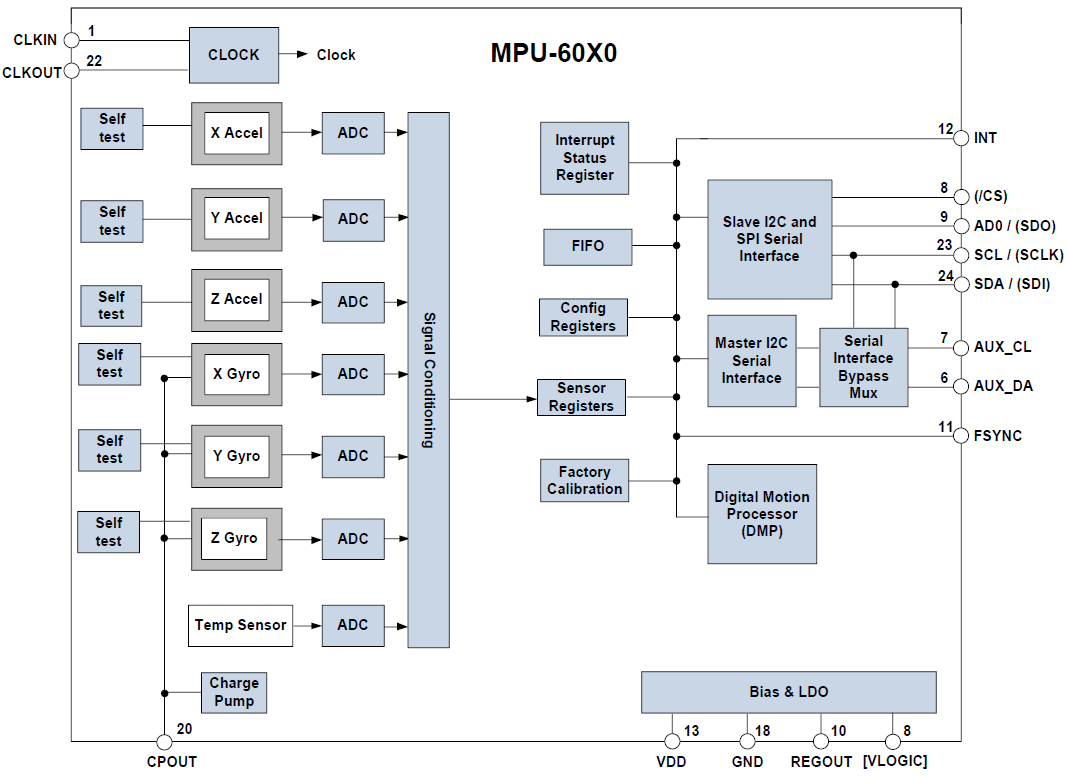
SPI写操作



SPI NOP操作时序图

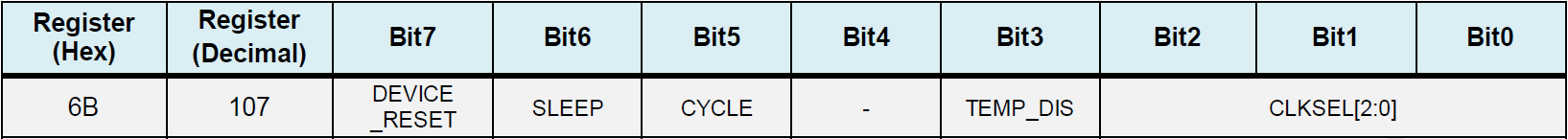
* + 1. MPU6050简介规范化

MPU6050是整合性六轴运动处理组件。



MPU6050框图

AD0=0地址=0X68；AD0=1地址=0X69；



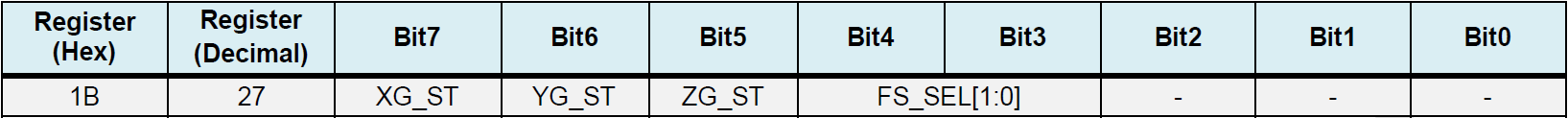
电源管理寄存器1（0X6B）

DEVICE\_RESE=1，复位MPU6050，复位完成后，自动清零。SLEEP=1，进入睡眠模式；SLEEP=0，正常工作模式。如下表所示：

系统时钟源选择表

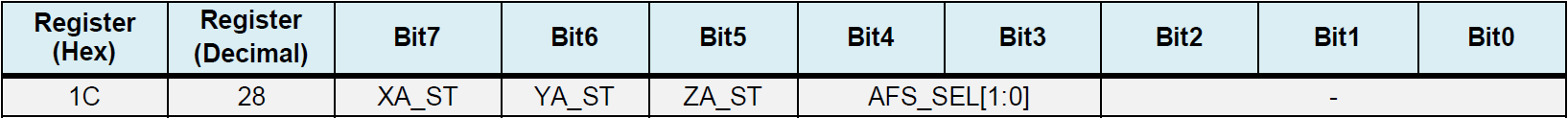
|  |  |
| --- | --- |
| CLKSEL[2:0] | 时钟源 |
| 000 | 内部8M RC晶振 |
| 001 | PLL，使用X轴陀螺作为参考 |
| 010 | PLL，使用Y轴陀螺作为参考 |
| 011 | PLL，使用Z轴陀螺作为参考 |
| 100 | PLL，使用外部32.768Khz作为参考 |
| 101 | PLL，使用外部19.2Mhz作为参考 |
| 110 | 保留 |
| 111 | 关闭时钟，保持时序产生电路复位状态 |

陀螺仪配置寄存器，FS\_SEL[1:0]用于设置陀螺仪的满量程范围。



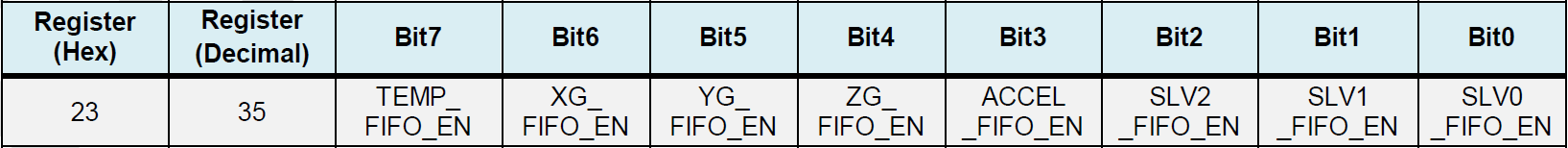
陀螺仪配置寄存器（0X1B）

加速度传感器配置寄存器，AFS\_SEL[1:0]用于设置加速度传感器的满量程范围。



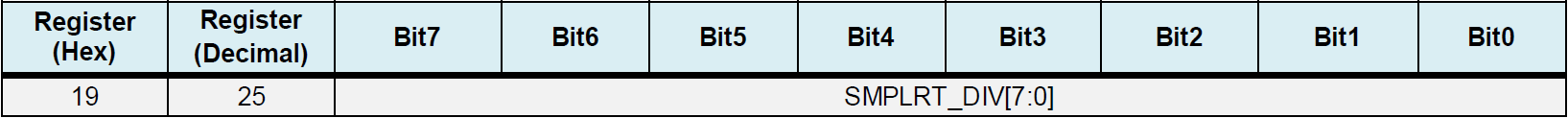
加速度传感器配置寄存器（0X1C）

FIFO使能寄存器用于控制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。

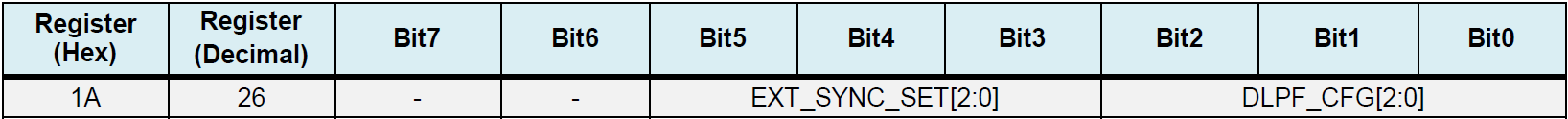


FIFO使能寄存器（0X23）

陀螺仪采样频率等于陀螺仪输出频率除以SMPLRT\_DIV寄存器的值加一，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。



陀螺仪采样率分频寄存器（0X19）



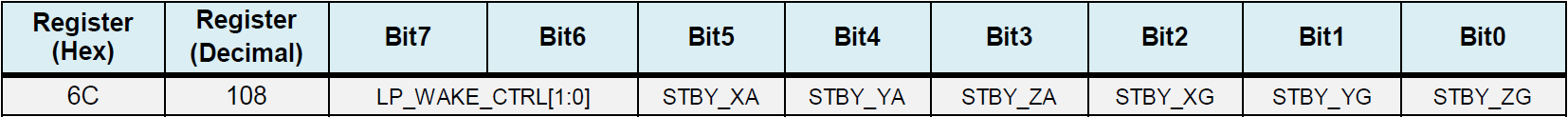
配置寄存器（0X1A）

数字低通滤波器（DLPF）的设置位，如下表：

数字低通滤波器（DLPF）的设置位

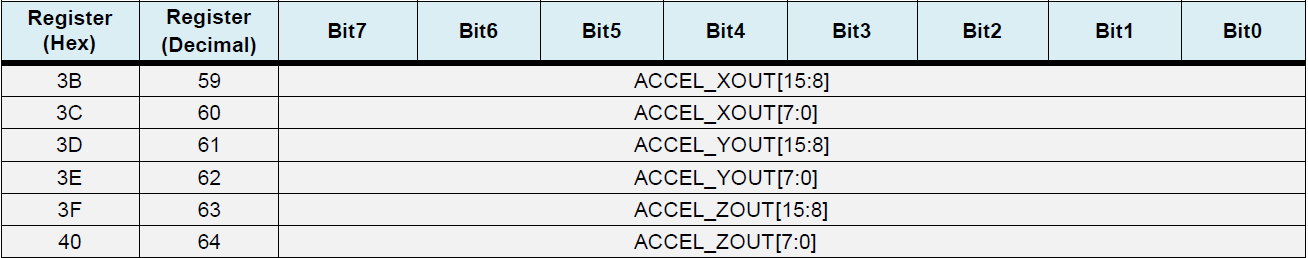
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DLPF\_CFG[2:0] | 加速度传感器Fs=1Khz | | 角速度传感器（陀螺仪） | | |
| 带宽(Hz) | 延迟（ms） | 带宽(Hz) | 延迟（ms） | Fs(Khz) |
| 000 | 260 | 0 | 256 | 0.98 | 8 |
| 001 | 184 | 2.0 | 188 | 1.9 | 1 |
| 010 | 94 | 3.0 | 98 | 2.8 | 1 |
| 011 | 44 | 4.9 | 42 | 4.8 | 1 |
| 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 | 保留 | | 保留 | | 8 |

该寄存器的LP\_WAKE\_CTRL用于控制低功耗时的唤醒。剩下的分别控制加速度和陀螺仪是否进入待机模式。



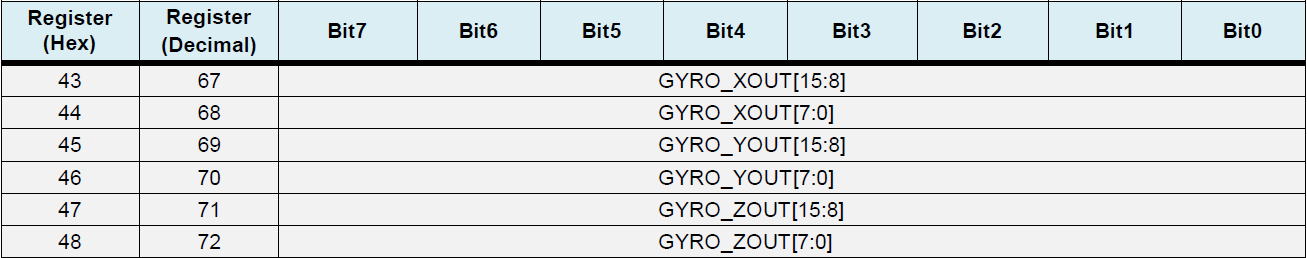
电源管理寄存器2（0X6C）

加速度传感器数据输出寄存器输出X/Y/Z三个轴的加速度传感器值，高字节在前，低字节在后。



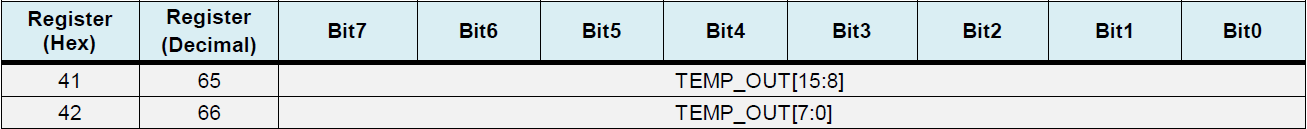
加速度传感器数据输出寄存器（0X3B~0X40）

陀螺仪数据输出寄存器输出X/Y/Z三个轴的陀螺仪传感器数据，高字节在前，低字节在后。



陀螺仪数据输出寄存器（0X43~0X48）

温度换算公式为：Temperature = 36.53 + regval/340单位为℃，regval为从0X41和0X42读到的温度传感器值。



温度传感器数据输出寄存器（0X41~0X42）

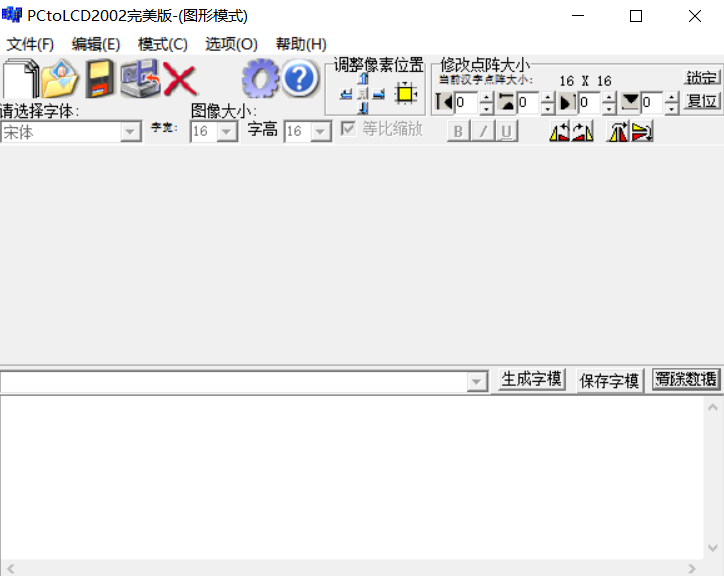
通过前面配置，便可以读取MPU6050的温度、加速度、陀螺仪的数据参数，但是我们在实际的使用过程中希望得到的数据是姿态数据欧拉角：yaw、roll和pitch。利用原始数据，进行姿态融合解算比较复杂，不易掌握，且会浪费大量单片机的计算资源。但是MPU6050本身拥有DMP数字运动处理器，并且InvenSense提供了对应的驱动库，官方DMP驱动库移植，主要是实现这4个函数：i2c\_write、i2c\_read、delay\_ms和get\_ms。

* 1. 软件部分
     1. 软件工具MDK5 简介

MDK 源自德国的 KEIL 公司，是 RealView MDK 的简称。在全球 MDK 被超过 10 万的嵌入式开发工程师使用。 MDK5 由两个部分组成： MDK Core 和 Software Packs。其中，Software Packs 可以独立于工具链进行新芯片支持和中间库的升级。

* + 1. Oled显示屏专用取模工具

该软件可生成中英文数字混合的字符串的字模数据。各种旋转，翻转文字功能。任意调整输出点阵大小，并任意调整字符在点阵中的位置。每行输出数据个数可调。支持四种取模方式：逐行，逐列，行列，列行支持阴码，阳码取模。支持纵向取模。输出数制可选16进制或10进制。动态液晶面板彷真，可调节彷真面板象素点大小和颜色。图形模式下可用鼠标作画，左键画图，右键擦图。



软件界面展示

* + 1. CubeMX

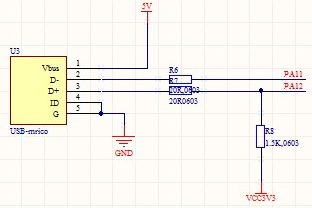
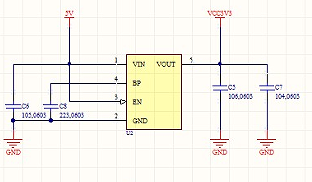
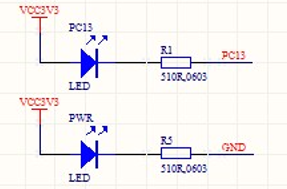
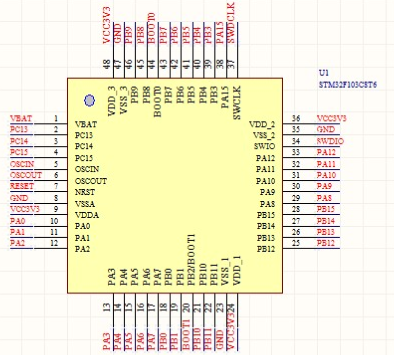
在Cube工具还没出来之前，在ST的MCU开发都是用标准固件库，标准库自推出以来受到ST的使用者的推崇，现在很多公司也都在使用。但是ST官方在2013年后就没有更新版本，ST官方也全力推HAL（Hardware Abstraction Layer）库。它的存在是为了确保 STM32 系列最大的移植性。CubeMX软件可以直接根据开发人员所需的功能勾选，以HAL库文件基础自动产生代码。开发者就可以轻松应用每一个外设。在 ST 官方的声明中，HAL 库是大势所趋，在 ST 公司最新开发的部分芯片中，只有 HAL 库而没有标准库，从这点便可以说明，以后的战略目标是逐渐的转向HAL 库 。相对于标准库来说，在使用 CubeMX 生成代码后，工程项目和初始化代码已经完成。cubeMX直接生产的工程支持IAR，keil，TrueSudio。

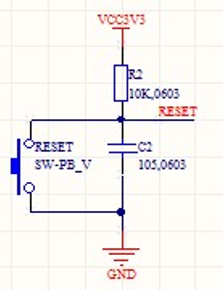
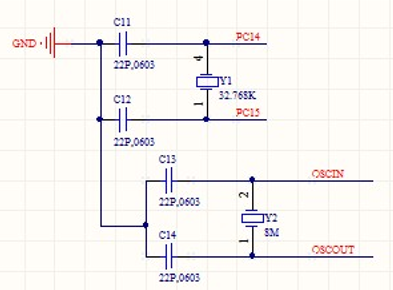
* + 1. Git

Git --- The stupid content tracker, 傻瓜内容跟踪器。Git是用于Linux内核开发的版本控制工具。Git的速度很快，这对于诸如Linux kernel这样的大项目来说自然很重要。Git最为出色的是它的合并跟踪（merge tracing）能力。现在，越来越多的著名项目采用 Git 来管理项目开发。

* + 1. 模块流程图

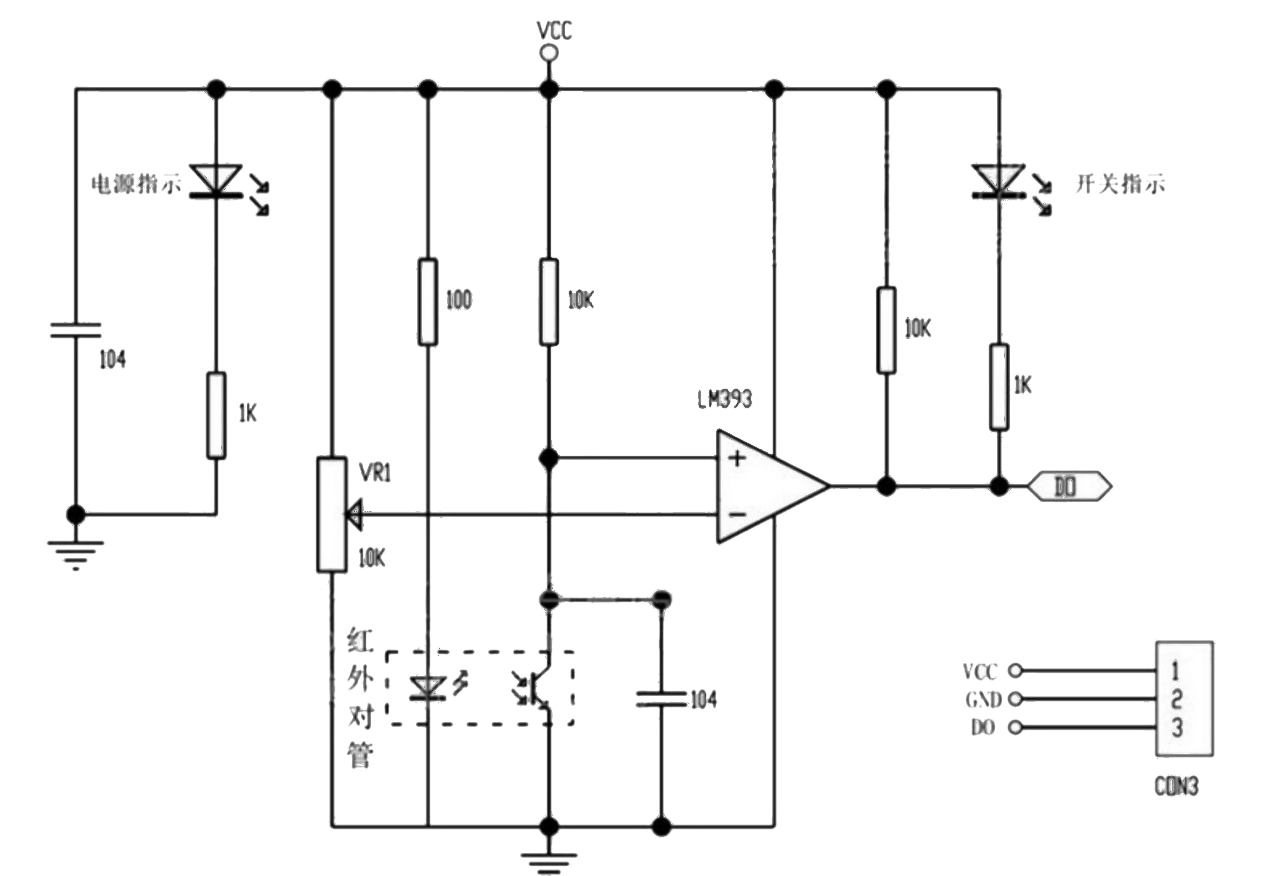
1. 实现功能与电路设计
   1. 实现功能与设计
      1. 实现功能描述
   2. 电路设计
      1. 最小系统板电路





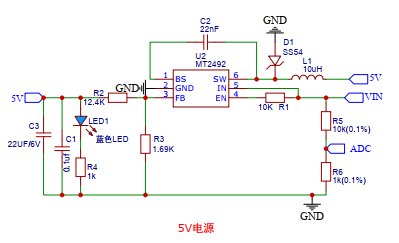
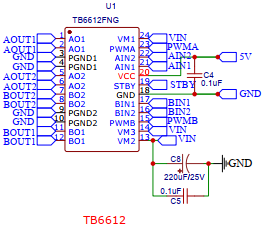
最小系统板C8T6主要部分电路图

* + 1. 红外检测模块电路

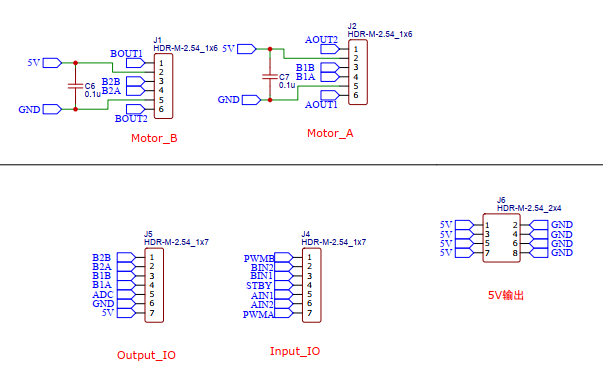


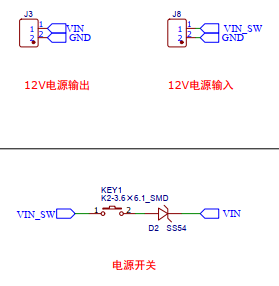
红外检测模块电路图

* + 1. TB6612带稳压模块板原理图



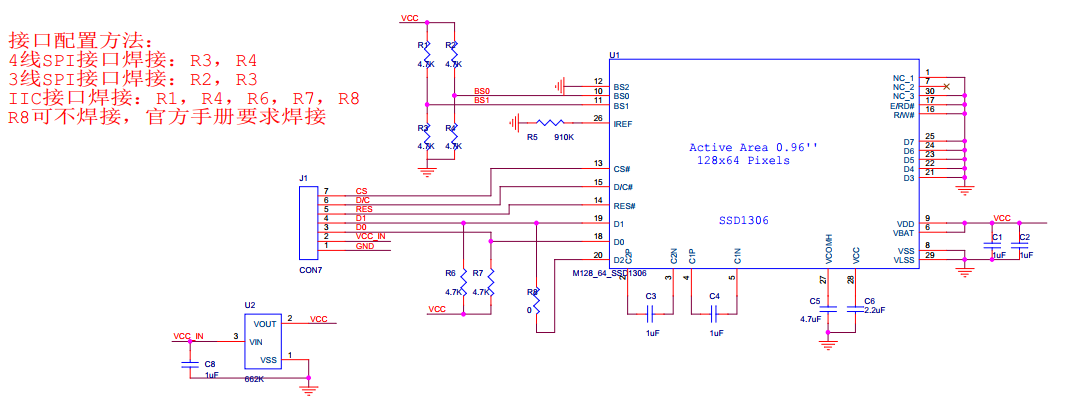
TB6612芯片和电源部分



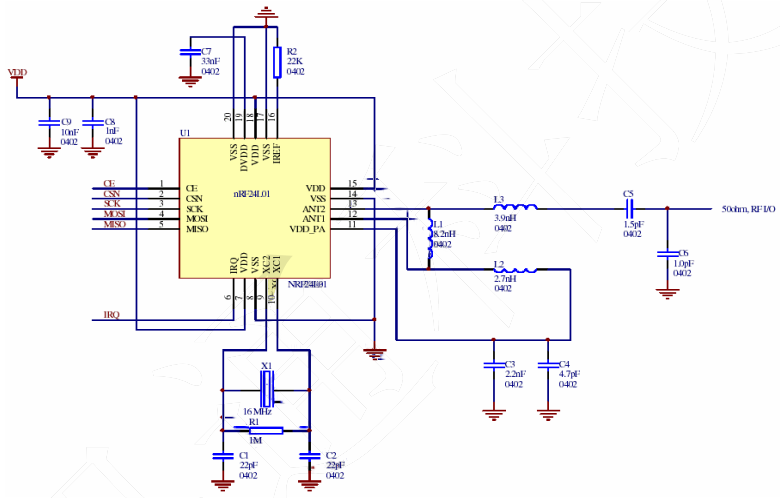


TB6612外接端口部分

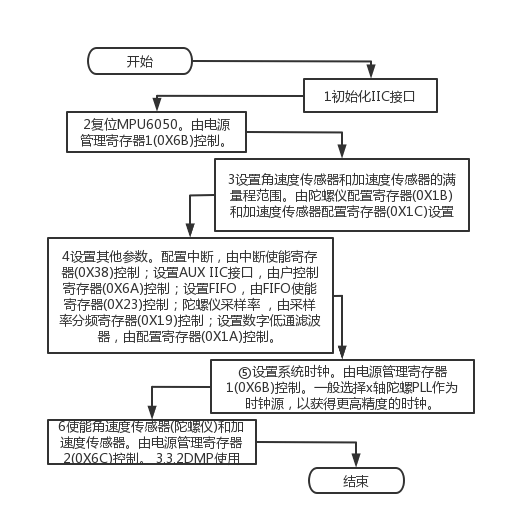
* + 1. 0.96OLED显示屏原理图



* + 1. nRF24L01 单端匹配网络：晶振、偏置电阻、去藕电容部分电路



* 1. 软件设计
     1. MPU6050初始化流程图



MPU6050初始化流程图

* + 1. DMP使用

MPU6050的DMP输出的是姿态解算后的四元数，采用的q30的格式，也就是放大了二的三十次方，要得到欧拉角，就得做一个转换，代码如下：q0=quat[0] / q30;q1=quat[1] / q30;q2=quat[2] / q30;q3=quat[3] / q30; pitch=asin(-2 \* q1 \* q3 + 2 \* q0\* q2)\* 57.3; roll=atan2(2 \* q2 \* q3 + 2 \* q0 \* q1, -2 \* q1 \* q1 - 2 \* q2\* q2 + 1)\* 57.3;yaw=atan2(2\*(q1\*q2+q0\*q3),q0\*q0+q1\*q1-q2\*q2-q3\*q3)\*57.3;quat[0]~quat[3]：是MPU6050的DMP解算后的四元数，q30格式。q30：是一个常量：1073741824，即2的30次方。

57.3：是弧度转换为角度，即180/π，这样结果就是以度（°）为单位的。

* + 1. 寻迹逻辑程序设计
    2. 电机驱动部分程序设计
    3. 通信部分程序设计
    4. 手动控制部分程序设计
    5. CubeMX对底层BSP(板级支持包)的初始化配置

1. 调试难点和问题
   1. 调试中遇到的重点与难点
      1. MPU6050的DMP调试出现FIFO溢出问题。

源代码：

while(1)

{

delay\_ms(1000);

if(mpu\_dmp\_get\_data(&pitch,&roll,&yaw)==0)

{

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

temp=MPU\_Read\_Temp();

MPU\_Read\_Accel(&ax,&ay,&az);

MPU\_Read\_Gyro(&gx,&gy,&gz);

mpu6050\_send\_data(ax,ay,az,gx,gy,gz); usart1\_report\_imu(ax,ay,az,gx,gy,gz,(int)(roll\*100),(int)(pitch\*100),(int)(yaw\*10));

}

}

通过打印信息分析是没有进入if语句，进入mpu\_dmp\_get\_data发现是卡在dmp\_read\_fifo函数if(dmp\_read\_fifo(gyro, accel, quat, &sensor\_timestamp, &sensors,&more))return 1;再进入dmp\_read\_fifo函数是卡在mpu\_read\_fifo\_stream函数，最终定位为如下函数的return语句：if (fifo\_count > (st.hw->max\_fifo >> 1)) { /\* FIFO is 50% full, better check overflow bit. \*/ if (i2c\_read(st.hw->addr, st.reg->int\_status, 1, tmp)) return -1; // printf("0x%02x, ",tmp[0]); if (tmp[0] & BIT\_FIFO\_OVERFLOW) { mpu\_reset\_fifo(); return -2; } }打印读到的int\_status的值为0x13,即为FIFO溢出产生中断。从语句中看出当FIFO的数量大于50%就会进入中断Check状态，计数寄存器的数据过多，所以溢出可能是因为读取速度过慢导致，回看MPU6050采样率设置为50Hz，即为20ms采样一次，while语句中设置了1s的延时，1s的延时期间导致采样数据过多而溢出，所以把延时改为100ms，问题解决。

* + 1. 红外检测过慢，导致小车错过检测跑道时机，而跑出跑道。
    2. Pwm过低，转弯的驱动能力不足。
  1. 解决方案
  2. 实现展示（附上仿真图或实物照片）

1. 总结

参考文献

[1]范书瑞,赵燕飞,高铁成. arm处理器C语言开发应用(第2版).北京:北京航空航天大学出版社,2001.78~80

[2]占跃华,主审,王明文.c语言程序设计(第2版).北京:北京邮电大学出版社,2003.128~132

[3]刘火良,杨森. FreeRTOS内核实现与应用开发实战指南(第1版).北京:机械工业出版社,2003.145~163

[4]徐广伟《RFID在煤矿系统中的应用》博硕《中国优秀硕士学位论文全文数据库》 2010

[5]潘勇《短距离无线数据网络的应用研究》博硕《中国优秀硕士学位论文全文数据库》 2010

[6]刘石亮《基于STM32的摄像机云台系统的设计》博硕《中国优秀硕士学位论文全文数据库》 2018

[7]靳培峰《智能双轮平衡车的设计》博硕《中国优秀硕士学位论文全文数据库》 2017

[8]

[9]

[10]

[11]

[12]

[13]

[14]

[15]

[16]

[17]

[18]

想对老师说的话

附录

附录一：06年11月CD电信新华营业厅日缴费顾客半小时到达数据

（略）

附录二：顾客调查问卷

（略）