

编 号：_____

审定成绩：_____

重庆邮电大学 毕业设计（论文）

设计（论文）题目： 基于 CAN 总线的直流电机驱动模块

学 院 名 称： 自动化学院

学 生 姓 名： 丁凡

专 业： 物联网工程

班 级： 0851201

学 号： 2012212719

指 导 教 师： 李勇

答辩组 负责人： _____

填表时间： 2016 年 05 月

重庆邮电大学教务处制

摘 要

众所周知，现在汽车相关的行业已经取得了飞速的发展，而在汽车行业中运用到非常广泛的就是 CAN 总线。CAN 总线的发展、使用要追溯到上世纪 90 年代，伴随着汽车中越来越多的控制单元的应用，传统意义的点对点的网络布线形式出现了很大的弊端，取而代之的就是更加简单的 CAN 总线网络布线。

本文就是模拟汽车中的 CAN 总线，设计出基于 CAN 总线的驱动模块的设计。其主要的要求是能够让上位机和下位机之间可以通过 CAN 总线进行数据的传输，并且下位机在得到信息后能够驱动直流减速电机转动。

本文的上位机和下位机的，也就是主机和从机，都采用的是 STM32F103c8t6。在主机和从机之间是采用 CAN 总线进行通信，CAN 收发器采用的是 SN65HVD230 芯片；在上位机中设计有 5 个按键，当按下不同的按键时，上位机将这代表的特定信息通过 CAN 总线控制器将它发送出去；当下位机接收到数据时，根据相应的信息控制相应的电机以不同的方式转动。最后，为了让设计更加的生动、形象，将下位机通过电机驱动 L9110S 芯片驱动的 6 路直流电机置于挖掘机模型中，能够让电机进行联动，并为挖掘机设计了前进、后退、转动、挖掘等动作。

【关键词】 CAN 总线 网络布线 电机联动 STM32F103c8t6 SN65HVD230 L9110S

ABSTRACT

As we all know, now related to the automotive industry has made rapid development, and in the automotive industry is applied to a very wide range of CAN bus. Development of CAN bus use dates back to the 1990s, when accompanied by an increasing number of cars in a control unit in the form of ad-hoc network cabling in the traditional sense of the emergence of a serious shortcoming, it is replaced by a simpler CAN bus network wiring.

This article is to simulate the vehicle CAN bus which is designed based on CAN bus driver module. The main requirement is the ability to be able to transmit data via CAN bus between the host computer and the next crew, the next crew and after obtaining information capable of driving DC gear motor rotation.

Upper and lower main control board paper machine are produced using STM32F103c8t6 microprocessor. Between the upper and lower machine is the use of CAN bus communication, CAN transceiver chip is SN65HVD230; PC design has the key means by pressing different buttons represent different information, and then host those machines send information via CAN bus protocol packetized data to the network; when the corresponding lower computer receives the data, depending on the motor rotation control different information in different ways. Finally, in order to make the design more vivid, the lower computer chip driven by a motor driven L9110S 6-channel DC motor placed excavator model, it is possible to rotate the motor linkage, and the excavator design forward, backward, turn, mining and other activities.

【Key words】 CAN bus Network Cabling Motor Linkage STM32F103c8t6 SN65HVD230
L9110S

目 录

前 言	1
第一章 概述	2
第一节 选题意义与设计要求	2
第二节 CAN 总线的发展和现状	3
第二章 CAN 总线挖掘机总体设计方案	5
第一节 挖掘机总体方案结构分析	5
第二节 挖掘机模型的本体结构	6
第三节 本章小结	8
第三章 CAN 总线挖掘机的硬件设计	9
第一节 概述	9
第二节 总体电路的设计	9
第三节 STM32 最小系统	11
第四节 外围电路设计	15
第五节 本章小结	20
第四章 CAN 总线挖掘机的底层驱动设计	21
第一节 底层软件任务分析	21
第二节 关键模块驱动程序设计	22
第三节 本章小结	27
第五章 CAN 总线挖掘机整体的调试和实现	28
第一节 关键部分的测试	28
第二节 本章小结	30
结 论	31
致 谢	32
附 录	34
一、英文原文:	34

二、英文翻译:	43
三、源程序	50

前 言

现如今，科技技术最大的特点就是机械制造技术，信息技术和信息控制技术的结合，汽车行业的智能化就是其中的一个运用^[1]。要想实现汽车的智能控制，除了需要计算机技术和智能算法高效的结合，还需要控制总线的运用。如何将汽车中越来越多的电气设备和控制单元相互通信，无论是动力系统的控制，还是汽车的行驶、转向、制动等，或者是其内部的参数的传输都是需要网络总线的。针对汽车电气系统如此复杂而又庞大的系统，并且都是要集中控制的，因此 CAN（Controller Area Network）总线在汽车电子的生产建设和科研工作中国发挥着越来越大的作用。

CAN 总线技术最开始是在军工等领域，现如今已经运用到汽车等领域，汽车上的 CAN 总线网络可以十分有效的将车上的各种传感器联系起来，实现数据的传递，也更加方便快捷的实现驾驶室对于汽车的控制^[2]。从某种意义上讲，CAN 是一种十分可靠的汽车网络总线，它可以减少共享汽车中几乎所有的信息，能够提高车辆系统的可靠性并且降低成本，使得汽车更加稳定、安全。想必以后 CAN 总线通信协议一定会在汽车电子控制系统领域有更加广泛的运用。

第一章 概述

第一节 选题意义与设计要求

一、选题意义

在汽车行业取得越来越大的进步时，越来越多的控制节点出现在汽车中，要实现各个节点之间的通信，将所有的节点通过总线相互关联起来是非常有现实的问题，而 CAN 总线就是不二之选。CAN 总线是一种运用非常广泛的现场总线，目前在国际上应用的相当的广泛，它是由德国 Bosch 公司为了处理车辆中各个传感器、控制单元等之间数据交换与传输而开发的串行通信协议^[3]。它已经渐渐地成为了汽车电子行业、计算机控制系统、嵌入式控制技术等领域已经成为了标准总线。CAN 总线能够将各个节点之间的数据实时性的通信，开发周期短，实用性强，由此 CAN 总线的功能和作用可见一斑，它已经逐步成为了汽车总线的代名词。

在过去的汽车行业中，主要是采用一种点对点的通信方式，它是需要大量的线路将各个传感器和控制单元串联起来。这使得线路复杂，布线繁琐。伴随而来的就是，各个节点之间的交流更加不流畅，不能够更好的实现控制，使得控制的运算、指令、算法更加的复杂。而采用 CAN 总线网络布局就使得各个控制单元之间的布线更加的简洁。它不仅可以做到数据信息交流、还可以减少线路中导线的使用，节约了成本，同时 CAN 总线也提高了汽车系统的稳定性和可靠性。下图 1-1 是过去采用的点对点的通信网络布线，图 1-2 是 CAN 总线的布线方式，从两图的对比中可以看出 CAN 优点是那么的明显，它仅仅就两根双绞线就可以将网络中的控制单元接入到总控制系统中，充分的降低了车体的重量，改善了汽车的整体性。

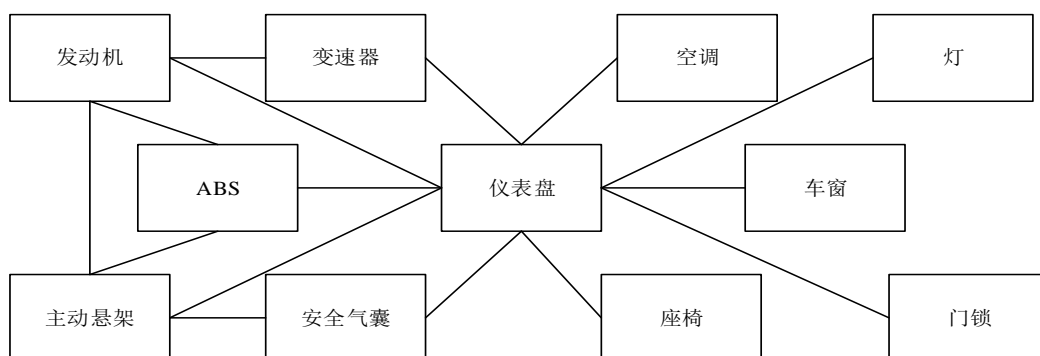


图 1-1 点对点网络布线

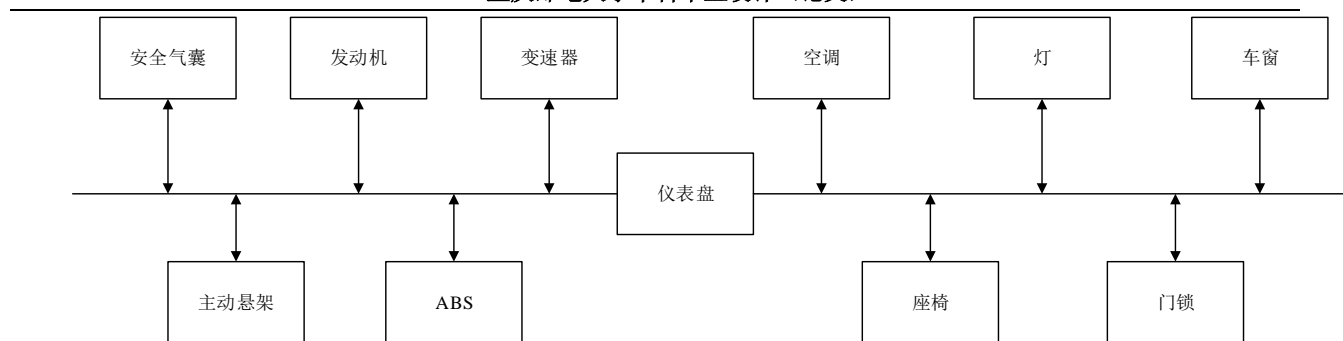


图 1-2 CAN 总线布线

本设计是基于 CAN 总线的 6 路直流减速电机控制的挖掘机，是一个简化是汽车控制系统的。主机单片机与从机单片机都是有 CAN 控制器，在于 CAN 接收器结合，就可以实时的传输了。从机是可以控制电机转动的。6 个电机直接与挖掘机连接，在某一时候，当不同的电机正转或者反转时，挖掘机就以不同的方式运动了。本设计只是 CAN 总线的一个简单的运用，仅仅是一个基础，后续可以在其基础上加上更多的节点和更多的各种功能的传感器，实现更多的运用。

二、毕业设计要求

毕业设计是理论与实践的结合，顺利完成它，是需要将我们在大学中所学知识的运用，充分考验我们的学习能力和知识的迁移能力；它更是我们是否能够顺利毕业的保障。因此，只有我们认真的对待它，这样才可以让我们在以后的学习和工作中有很好的基础。

本次毕业设计，我将完成如下要求：

- ①对以前学习过的知识在进行回顾和不会的在学习。了解毕业设计相关的基础知识以及对问题进行分析，了解其目的、对设计有个大概的认识。进一步学习所需用到的模块的硬件电路和软件原理。
- ②认真学习硬件电路设计的绘制和仿真软件的学习，提高对这些软件的使用熟练度。
- ③掌握相关的化 PCB 图的软件，设计和绘制主控板，并且自己打板。
- ④掌握相关编写程序代码的软件，尽量能够熟练运用。
- ⑤复习相关的数电和模电的知识，对要用到的进行系统复习。
- ⑥学习并掌握使用的主控制芯片。
- ⑦能够提高自己的动手能力，能够通过查找相关书籍、网络资源解决相关的难题。
- ⑧能够对自己的程序代码有清楚的认识，可以尽量做到算法的简便。
- ⑨善于与他人交流，面对复杂问题的时候，能够虚心的向他人请教，采纳有用的意见。

第二节 CAN 总线的发展和现状

一、CAN 总线的发展

论述 CAN 总线的发展要追溯到上世纪 90 年代，当时为了满足汽车行业对于实用性平台的需求，汽车领域厂商将各个车辆的重要部分逐渐模块化，汽车厂商们开始思考如何在车辆的这样的模块之间、各个控制单元之间建立一个信息数据相互交流、传输的平台，如何建立一个能够通用标准化的车辆控制模块接口。从此，汽车总线技术就此诞生了，该总线技术就是如何能够将复杂的、大量是数据进行交流与传输；如何能够简化加入大量模块、传感器和控制单元后线路；如何能够更加降低成本。自此之后，CAN 总线技术就逐步的运用到全球各大汽车制造商中，他们都使用 CAN 总线技术，使得汽车中的模块和零部件都能够更好的串联起来。现在各个汽车厂商不仅使用 CAN 总线技术，而且他们还更进一步的建立了自己的实验室，对其进行更加透彻的研究。

相对于全球的 CAN 总线的发展，由于我国在汽车电子等领域发展的较为晚，相对于全球平均水平存在着一定的差距，但是我们还是在这上面还是有着相当的进步，并且投入的精力也是很大的。我国最开始研究 CAN 总线是由于外国的 CANopen 通讯电子协议后才产生了强烈的兴趣并在高校中进行研究。现如今 CAN 总线技术在我国国产车中有着广泛的运用，并且还引进国外的技术进行改进和研究，研制出自己的产品。

二、CAN 总线的未来

现在，我们都知道 CAN 总线有着非常大的优点，它已经逐步成为了世界范围内使用的最多、最为广泛的总线形式，几乎可以说是世界总线的一种标准了。并且现在它已经不只是在汽车领域的使用，而且逐步轮船等其他工业领域有着巨大的运用。不仅如此，现在随着互联网技术的不断发展，相信在未来它一定可以运用到更多的领域，适用于更多的产品，例如：家居、医疗等。

第二章 CAN 总线挖掘机总体设计方案

第一节 挖掘机总体方案结构分析

一、工业上基于 CAN 总线的汽车的结构与功能

在工业的生产中，基于 CAN 总线的汽车主要是由机械部分、传感部分、控制部分等三个主要的部分构成的^[4]，在其基础上我们可以进行更进一步的划分，大致可以分为执行系统、电机驱动部分、控制单元系统、感知系统、决策系统和软件部分、人-机-环境交互系统等。它们的每一部分基本上都是通过 CAN 总线相互连接通信的，相互传输要传递的数据的。其大致的结构图如图 2-1 所示：

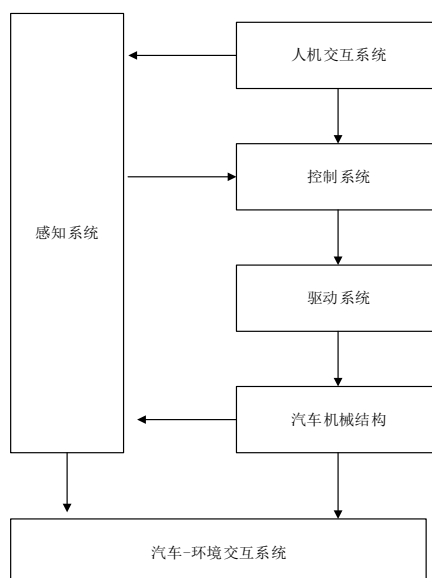


图 2-1 汽车系统结构图

1、执行系统

汽车的执行通俗的讲，就是整个汽车总控是如何执行具体的指令的系统。它主要的功能就是当驾驶员将下一步要做到操作传达给该系统，它就可以再将这些传达给其他的控制系统，例如驱动，传感等系统。

2、驱动系统

驱动系统主要是如何向汽车的执行机构提供动力的设备。按照驱动方式的不同，主要可以将其划分为液压式、气动式、电动式和特种式^[5]。每一种驱动方式都是有着自己的优缺点的，都是需要在实践情况下根据实际的问题采用最为妥善的驱动方式。

3、控制系统

汽车中的控制系统可以说是汽车在运行过程中的中枢部分，它是有可以具体的控制驱动系统，让汽车的控制系统能够在一定的时间里执行所下的指令和规定的要求。常见的控制系

统有语音控制、视频控制和计算机控制。当我们论述到它具体的功能时，它就如同是当行驶人在实时的操作，能够将它立即的反馈给中枢，中枢控制系统做出指令程序的信号，从而控制汽车的执行机构完成相应的动作。

4、感知系统

感知系统主要是针对于现在正在研发的智能汽车，因为在它们的内部有很多的测试车内环境的传感器。通过这些单元感知内部的信息，并且可以传给主控制器。

5、决策系统和软件部分

决策系统是工业汽车根据感知系统得打的数据信息做出当前情况下最为正确、最有利、损失最小的决策，它是可以自己进行推断，学习以及自动调整控制算法的软硬件支持部分。对于软件部分来说，其实就是整个汽车系统的控制端的代码算法等。

6、人-机-环境交互系统

这是对于目前正在研发的智能汽车而提出的，它是基于汽车与人，人以及与环境以及车与环境三者的变化关系，做到相互联系、相互协调的系统。主要是当三者中任何一个发生变化时，都可以给其他两个提供信息交流，做到随时随地的交换信息。

二、CAN 总线挖掘机总体功能分析

根据现在汽车行业的发展和 CAN 总线的运用，借鉴在实际生活中挖掘机在操作时的动作要求、动作方式、动作幅度和整体性并且结合 STM32 的系统资源以及在毕业设计的要求，本文将基于 CAN 总线的挖掘机的主机和从机的功能分析如下：

①设计的挖掘机可以正常的完成所要执行的动作，其控制的主机和从机的主控芯片是 STM32，它们的外部硬件电路能够正常的工作。

②挖掘机的前进、后退或者是转向和挖掘等都是通过从机 STM32 的指令来控制 6 路直流电机的正转或者是反转来实现的。

③综合思考了电机和驱动控制芯片的实际的结构和在电路板上的放置的情况，最终是设计了 6 个相同的硬件接口。

④为了能够更加具体的模拟工业汽车中 CAN 总线的运用，本文简化了工业中多节点的接入，仅仅是采用了一对一的模式，也就是只包括了主机和从机。具体是主机可以实时的采用按键的方式对从机进行控制，之间是通过 CAN 总线进行数据、指令发送的。

⑤本次毕业设计的电源部分是采用电池串联供电的方式，总体的电池的输输出大约有 7.2 伏。它主要是给 STM32，电机驱动等主要的元器件供电。但是它们所要的工作电压是不同的，就需要采用 3.3V 的稳压芯片。

⑥主机和从机都是通过 J-Link 下载。

⑦编写挖掘机所要完成的简单动作的程序并进行调试。

第二节 挖掘机模型的本体结构

本次毕业设计的主要目的是基与 CAN 总线的驱动主板的设计，它主要是要驱动 6 路直流减速电机，然而这 6 个电机都是放置在挖掘机模型中的，因此我的毕业设计的主要目的是能够通过驱动电机来让挖掘机完成相应的动作。挖掘机模型全身都是有塑料支撑结构；由五节干电池串联提供 7.2V 电源；整个模型大致可以分为履带式的下部行走体，可旋转的上部回转体和工作装置（动臂、斗杆、铲斗）等组成，大概有十来处关节，由 6 个直流电机配合减速齿轮组来来连接各个关节来运动。其中它的电机分布位置如图 2-2 所示：

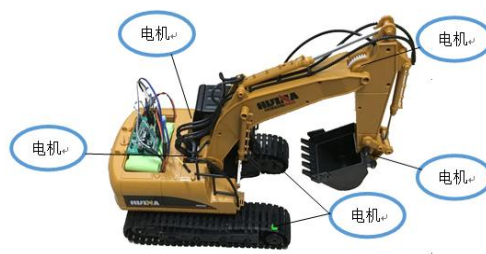


图 2-2 挖掘机电机分布图

一、挖掘机履带式下部行走体机械结构

本设计的挖掘机模型的底盘结构是模仿实际生产生活中的挖掘机的底盘结构，它是履带式的，它的底盘是一个整体的结构。其有两条履带式的底盘，是成中心对称的，其运动方式是像坦克的，能够前进、后退、转动。其动力来源是分别位于两侧的直流减速电机和减速齿轮组成的传动机构，它固定在底盘的两侧。

二、挖掘机的上部回转体机械结构

本设计的挖掘机的上部回转体的机械结构相对而言是较为简单的，它主要是通过一个直流电机和减速齿轮，并且它的减速齿轮是和底盘结构能够固定在一起的，这样就能够当电机旋转的时候，就可以以底盘为参考位置来做到旋转，并且它的设计是能够做到以某一方向，例如顺时针旋转 360 度的。

三、挖掘机的工作装置的机械结构

本次毕业设计的挖掘机模型是仿照实际生活中的挖掘机的机械结构，它主要是由动臂、斗杆、铲斗组成的^[6]。并且它的这三部分分别配置了一个直流减速电机，是可以对它们就行独立的操作的，这也让它在做出挖掘的动作时更加的具体、形象。它就像是一个人的手臂似的。动臂相比于人的手臂，斗杆类似于手肘，铲斗就如同手掌一样，它在挖掘是就如同手抓东西一样。它的动力来源也是三个电机的旋转。在我看来，在实际的操作中，动臂是能够以较大的角度来让转动整个装置；斗杆就是更加细致的就行旋转，让操作员能够将其移动到指定的位置；铲斗是通过电机的正反旋转来形象的展示挖掘机的工作。

第三节 本章小结

本章主要讲述了挖掘机模型的总体设计方案，其中主要包括了目前工业实际生产生活中汽车的设计中包含的主要系统，此外还针对本毕业设计主要是基于 CAN 总线的挖掘机驱动设计，简单的介绍了所要完成的挖掘机驱动总体的设计需求，对总体功能更进行了分析。此外就是介绍了使用的挖掘机模型的机械结构，主要是挖掘机的下部、中部的等用于行进和转向的结构，再就是挖掘机的挖掘的手臂，主要是其结构中的三个电机的介绍。最后就是介绍它们是怎样结合，就可以达到目的。

第三章 CAN 总线挖掘机的硬件设计

第一节 概述

对于我们来说，硬件电路的正确设计是非常关键的，它的好坏直接决定了软件部分能否正确运行。同样软件编写的好坏同样也能够影响硬件设计所要达到的目标。硬件电路的设计一般是在最初阶段，在设计总体规划的时候就需要我们能够对硬件电路有着充分的认识，这样能够到后期可以根据不同的需求还可以做出一定的改进。这就需要在前期的分析准备充分，硬件设计的改动、返工就比较少，就可以有跟充裕的时间去调试软件部分。在现在，我们已经从以前的各种零散的元器件组成的电路变成了可以采用模块化的电路，使用集成度高的模块，这种情况下大大的使我们的任务减轻了，并且硬件设计所需要的时间也同步减少，这为软件编写节约了大量的时间。同时它也为我们广大的学生提供了方便，我们可以快速的入门，不必对硬件电路的每一个细节都非常的懂，只需要对主控芯片进行认真学习，对各个模块只用了解其功能。为了减少硬件电路设计的返工，我们在运用时应该注意一下方面：

①在前期针对所要使用的主控芯片 STM32 进行学习。主要时了解和使用的 CAN 控制器和普通 I/O 口的配置。

②设计时要设计出没用到的 I/O 口接口。在设计时，往往有时候容易出现没有考虑的情况，多留些接口，以后可以在外接，不用大规模改动电路。硬件电路往往不是一次就可以成功的，往往在调试软件过程中会有新的想法，如添加其他模块或者控件，要是没有通用的外接接口的话，就没有余地，就又有可能要重新设计，重新制板，这样是非常耗时耗力的，影响到后续程序的调试，耽误毕业设计的进程。

③对主机和从机的主板设计尺寸有个大概的计划，要综合考虑到每个模块的尺寸，以便可以在完成板子焊接的时候更加方便，有余地。

④模块的选择要充分的配合主控的单片机 MCU，要考虑到模块是否能够适用于选择的芯片，所用的芯片能否正常的驱动起所用的模块，能否让它稳定的工作。

⑤电源是整个设计中的关键部分，如果没有一个正常输出的电源，是不可能为系统提供它所需要功率的。最终就会影响到芯片和所用的模块、电机驱动、电机等的正常工作。除此之外，各个模块和芯片在工作时也是会相互影响的，这也是要考虑的。

第二节 总体电路的设计

本文的基于 CAN 的挖掘机的主机和从机的总体的硬件电路的结构框图大致如图 3-1，主机的原理图如图 3-2，从机的原理图如图 3-3，主机和从机的实际生产制作的电路板见附录。其中主机和从机的原理图中的单片机都是 ARM CORTEX M3 内核的 STM32F103C8T6；

L1~L6 是 6 路电机驱动；JLINK 为程序下载接口；VP230 是 CAN 总线协议的接收芯片；K1、K、K3、K4、K5 是主机中的按键部分。

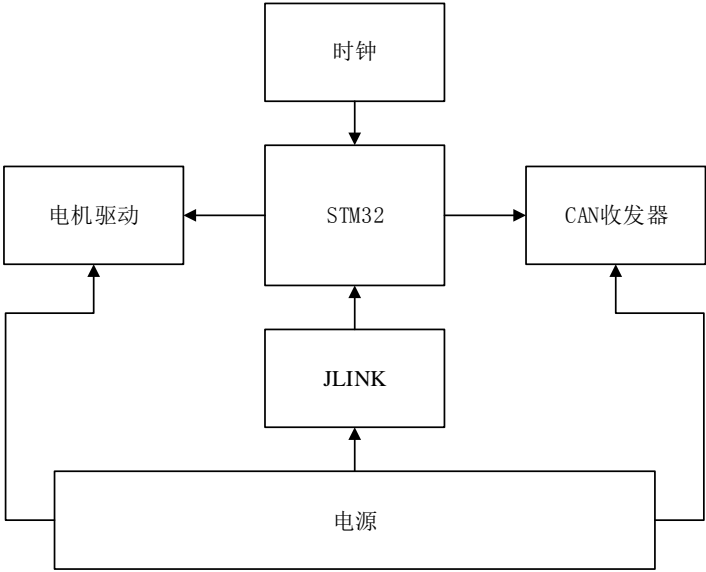


图 3-1 硬件电路总体结构图

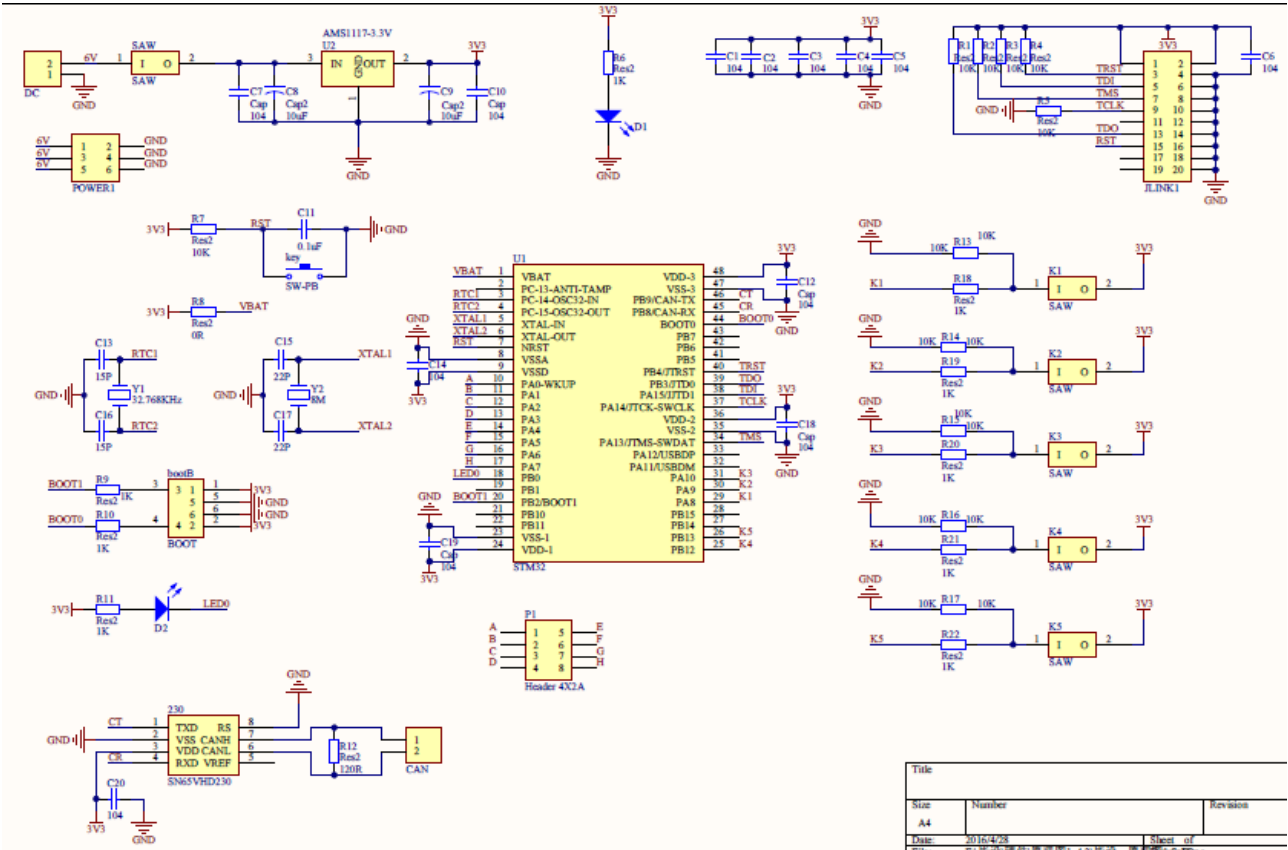


图 3-2 主机原理图

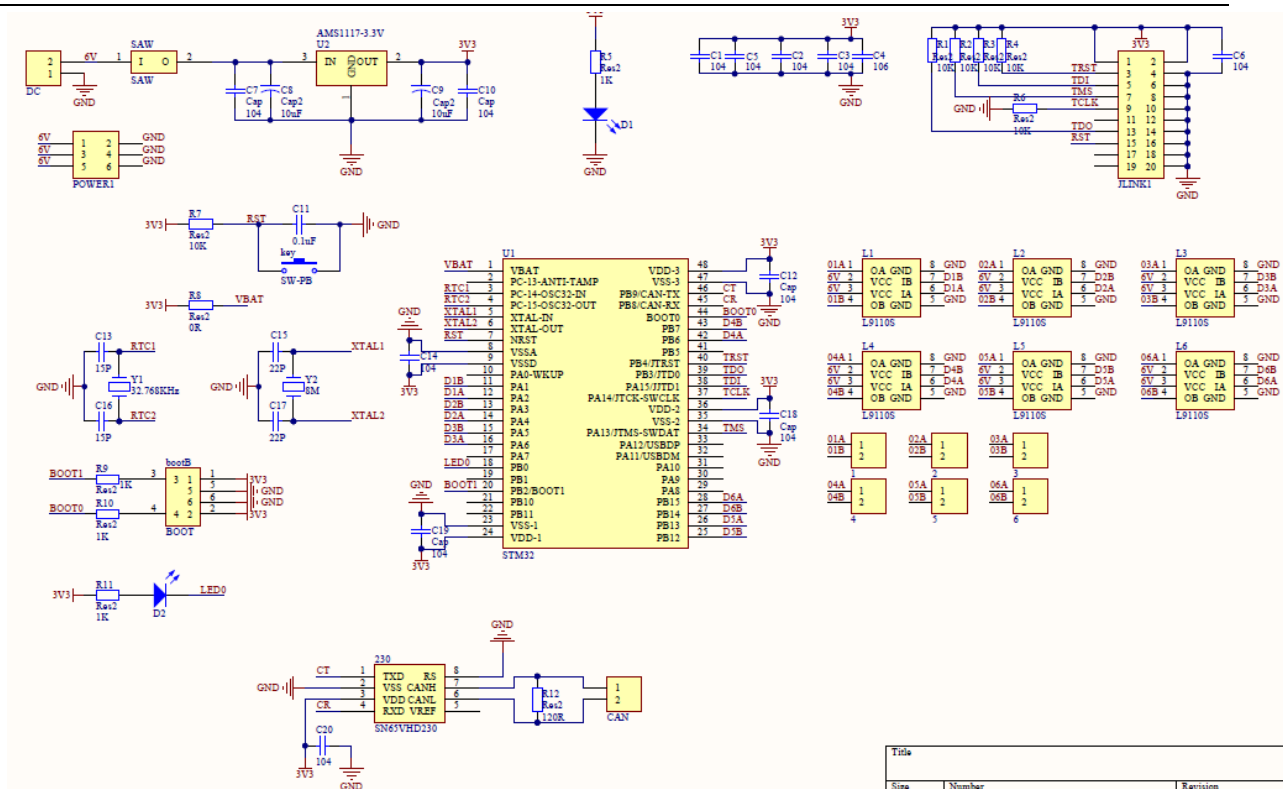


图 3-3 从机原理图

第三节 STM32 最小系统

单片机是一种高度集成的电路芯片，采用超大规模集成技术把微处理器（CPU）、随机数据存储器（RAM）、只读存储器（ROM）、以及输入/输出电路（I/O口）等电路高度集成到单片芯片上，构成一个最小而且功能完善的计算机系统^[7]。使用STM32相关的芯片，在一定程度上可以简化任务。因为TI公司为我们提供了很多的库函数，很多函数的接口，能够简化我们的编程难度。还有就是现在的单片机都是高度集成化的，高效的，更好的实现控制。

随着嵌入式的快速发展以及在各行业的深入应用，人们对电子设备的微型化、智能化的要求也越来越高，在工程控制、机械加工、航空航天等领域，单片机都扮演着非常重要的角色。因此，对于我们大学生，在完成毕业设计的时候要尽量多的去查资料，主动学习，这样才能够锻炼我们的能力，当以后再更进一步的学习或者工作的时候才能够更好的适应，更容易入门。

随着 MCU 技术的运用,与各种控制模块的结合也让这种 MCU 的应用无所不在,其涉及的领域也是越来越多。它经过飞速的发展,现在有的系统已经不在只是 8 位数据线的单片机了,有的已经是 16 位或者 32 位单片机了,甚至有的已经达到了 64 位,都已经是有着极高运算速度。

一、最小系统原理图

没有最小系统是不可能让系统正常工作的，然而最小系统主要是由 MCU 和电源电路、时钟电路、复位电路四个部分组成的^[8]。对于系统来说，只要有了功能完整的最小系统，就可以在其基础上设计其他的外接电路。本文对于最小系统的简介如下。

1、STM32 主控单片机

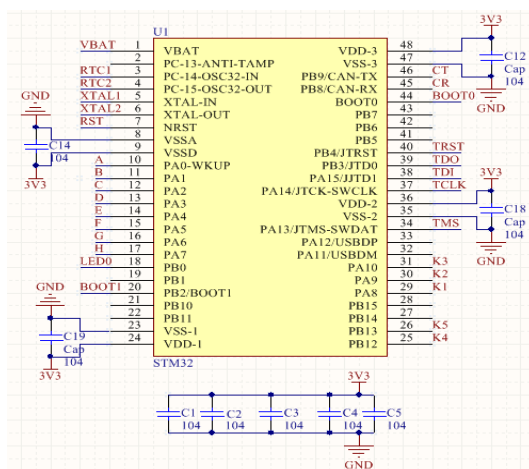


图 3-4 STM32 主控

本设计采用的是 STM32F103c8t6 芯片，它是 TI 公司设计的，这种芯片是基于的 ARM 的架构，是 M3 的内核^[9]。STM32F103XX 系列的单片机微处理器最高可以达到的操作频率为 72MHz，有着很快的运算速度。现在，像这种微控制器已经在生产和生活中运用非常广泛，并且在学生的学习中也很常见。

本次的毕业设计采用的是 STM32F103c8t6 主控芯片，它的封装是贴片的，共有 48 个引脚。图 3-4 就是 STM32F103c8t6 芯片的管脚的连接图，我们可以看到，在每一个电源端和地之间都接了一个 104 的去耦电容。这样的目的是能够滤波，以保证芯片的正常稳定的工作。同时在芯片下面的 5 个 104 的电容，它们的作用是滤波的，能够保证输入到芯片的信号质量更高。

2、STM32 复位电路

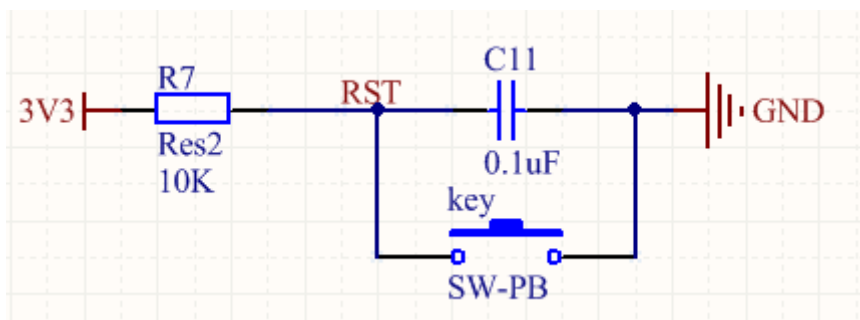


图 3-5 复位电路

在本设计的主机和从机的控制板上都是采用的 STM32 最常见的复位电路的一种：按键复位，也就是外部 RESET 复位。在从机的电机控制主板上，通过按键的方式，当我们需要程序复位时，在重新执行，就可以按下按键，及时的复位。操作方便，简单。因此本设计的复位电路如图 3-5 所示。

3、电源电路

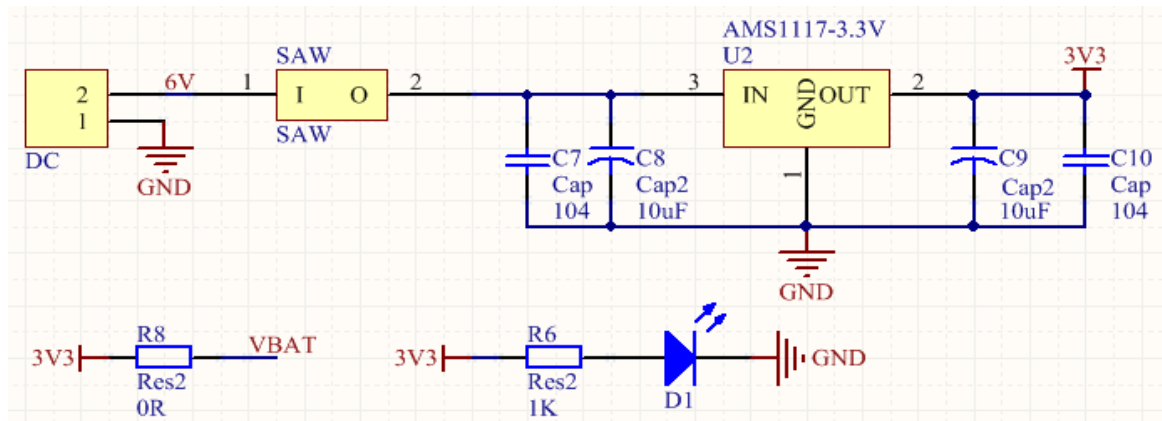


图 3-6 电源电路

对于 STM32F103XX 系列的单片机，在对其电源部分设计的时候，根据单片机的性能。为了保证芯片的正常的运行，是需要为其的 4 个不同的引脚提供电源。其中主要是：

- ①自身的内核和连接的模块所需要的 3.3V 的电源。
- ②内部稳压器所需要的 3.3V 电源。
- ③模拟部分所需要的 3.3V 电源。
- ④VBAT 的 3.3V 电源。

AMS117-3.3 是一个正向低压降稳压器，实质上是一个低漏失电压调整管，它是由一个 PNP 驱动和一个 NPN 驱动组成的。漏失电压的定义是： $V_{DROP}=V_{BT}+V_{SAT}$ 在 1A 电流下压降是 1.2V。该稳压芯片有两个差异很大的版本，不用的版本所能够稳到的电压有所不同的，而且它们的精度也是不同的。本设计中主机和从机的控制芯片需要的是 3.3V 的电压时，因此我们最后选择的是 AMS1117-3.3。对于 AMS1117 系列稳压器主要特点：

- ①该芯片的输出电压是有很多种版本，精度为百分之一左右。
- ②该芯片输出的电压、电流稳定，精度高。
- ③限流功能，过热时会自动切断。
- ④温度范围是：-40℃~125℃。

本系统采用的是总共 5 节干电池串联起来为主机和从机供电，大致的电源部分的硬件结构如图 3-6 所示。首先是经过设置的 6 脚自锁开关，它是可以控制系统主板整个的电源的；然后在经过两个电容来进行滤波；最后就是 7V 左右的电压输入到稳压芯片，而在输出的就是 3.3V 左右的电压了。STM32F103c8t6 具有独立的模拟单源和数字电源，考虑到降低噪声和出错的几率比不是很大，我们可以为了电路的简化，模拟地和数字地没必要进行隔离，本次设计都是将它们共同接到公共的 GND 上去。AMS117-3.3 的稳压芯片的输出的电流可以达到 800mA，它是完全可以承担单片机和各个模块的供电的。

4、时钟晶振电路

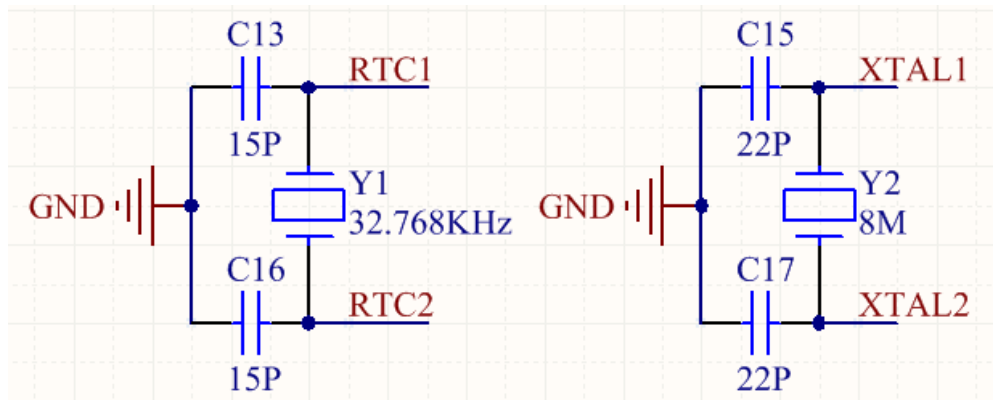


图 3-7 时钟晶振电路

我们都知道，在一个单片机系统中，时钟晶振是多么的重要，它好比是人的心脏，它是维持系统运行是必不可少的。在以后的程序设中，当使用到定时器时，只有有效的晶振能让定时器准确，从图 3-7 中可以很明显的知道 STM32 的时钟晶振的连接的电路图。在 STM32 中一般要连接两个晶振，一个是 RTC，一个是普通的时钟。这里的晶振选用的是 32.768KHz 和 8M 的；电容分别并联的 15P 和 22P 的。

二、STM32 的功能及简介

STM32 系列相关的芯片，可以说都是大同小异的，基本上我们在生活中使用的都是 ARM 的结构。此外，对于 TI 公司生产的这类 MCU 的最大的特点就是它是低耗、高效的、便宜，对于日常的生活的嵌入式产品的主控可以说是不二之选。STM32 主系统主要是有四个驱动单元和四个被动单元组成的，四个驱动单元分别是内核 DCode 总线、系统总线、通用 DMA1 和通用 DMA2；四个被动单元是 AHB 到 APB 的桥、内部 FLASH 闪存器、内部 SRAM、FSMC^[10]。除此之外它本身也是有着很多的资源的，其中具体包括 64KB SRAM、128KB FLASH、2 个基本定时器、4 个通用定时器、2 个高级定时器、3 个 SPI、2 个 IIC、5 个串口、1 个 USB、1 个 CAN、3 个 12 位 ADC、1 个 SDIO 接口、1 个 FSMC 接口、USB 主/从/OTG、以太网、正交编码器接口以及 36 个通用 I/O 口等。下图 3-8 是 STM32F103c8t6 的内部系统结构图：

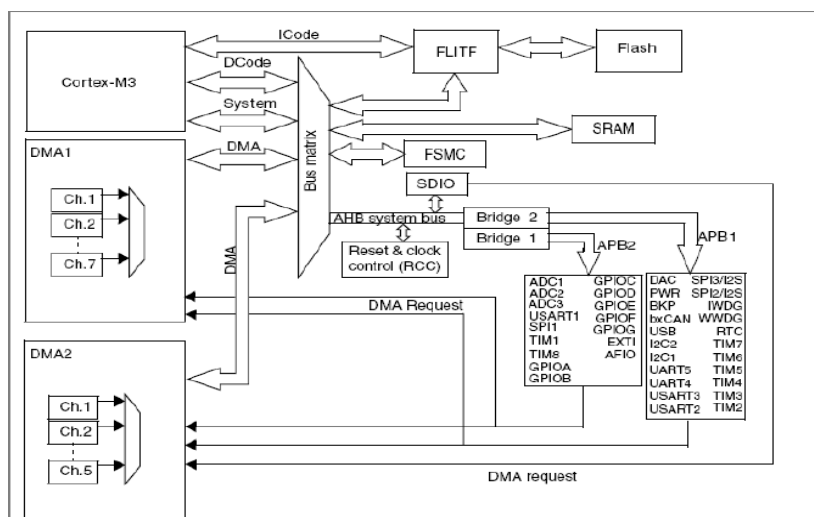


图 3-8 STM32F103c8t6 系统内部结构

1、片内 SRAM

STM32 内部有着 64KB 的片内 SRAM，这么多片内存储空间，可以有效的增加数据的总量，程序的读取速度，可以有更高的吞吐量。

2、串行接口

USB 串口和 DMA 串行通信方式。

2 个 UART 串口、一个 485 串口通信总线。

CAN 控制器，带有 2 个通道。

一个 SPI 控制器，有多种通信方式。

2 个 SSP 控制器，它们是可以支持多种通信协议的。其中一个是可以选择有于 SPI，并且和 SPI 有着共同的中断引脚。

2 个增强型的 IIC 总线接口。

3、其他外设

STM32 的内部资源除了片内 SRAM 和各种的串行通信的控制器，它还有其他各种的定时器等，还有丰富的 I/O 资源，不仅可以连接外围的模块，使它功能更多，还可以输出 PWM 波，可以用于控制电机等的转速。

第四节 外围电路设计

一、CAN 总线接收器电路设计

本设计的 CAN 总线通信是这次毕业设计重中之重，它的正确通信是保证上位机和下位机也就是设计中的主机和从机之间能否正常通信的基础。在基于 CAN 总线控制的挖掘机的设计时，我们的选用是 STM32 系列的单片机作为 CAN 总线的控制器，使用 SN65VHD230（以下用 VP230 表示）作为 CAN 总线的收发器。因为主机和从机的 CAN 总线和收发器的连接图和从机的是相同的，因此这里只给出了从机的连接图。具体的连接如下图 3-9 所示：

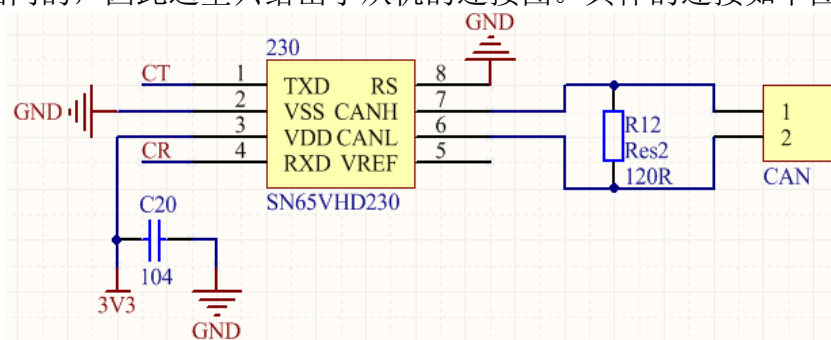


图 3-9 CAN 收发器电路连接图

从上面的 STM32 的内部的 CAN 控制器与 CAN 收发器 VP230 的连接硬件电路图中可以看出 VP230 的 TXD 和 RXD 引脚分别是连接在 STM32 的 PB8/CAN-TX 和 PB9/CAN-RX 的引脚上，而 VSS 是直接连接在电源 3.3V 上的，只不过还在它与 GND 的连接上多添加了一

个去耦电容，其主要的目的是去除其它设备在运行时的产生的电信号的干扰。同时最后可以看出 VP230 的 CANH 和 CANL 引脚是去连接同样有着 CANH 和 CANL 的线路的，这样就可以将两个甚至是多个的具有 CAN 总线结构的器件接入到 CAN 总线的网络中。而在每一个 VP230 的芯片的 CANH 和 CANL 引脚间都是需要并联上一个 120 欧的终端电阻的，其主要的目的是因为在高频的信号在传输的时候，CAN 总线收发器的发出的信号波长是很短的，在终端是很容易生成反射波，这样就会干扰 CAN 收发器的正常的工作，所以在每一个终端上都加上一个 120 欧姆的电阻进行阻抗的匹配比便使每一个收发器的 CANH 和 CANL 的连接线路上的电压能够清楚的进行辨别，这样就能够及时的避免发送和接收信号的干扰，保证 CAN 收发器的正常的工作。

其实在进行关于 CAN 总线的设计的时候，我们可以很清楚的选择到 STM32 作为 CAN 控制器的主控芯片，因为它的性能和质量都是能让我们值得保证。还有就是在对 CAN 收发器的驱动芯片进行选择，在完成本次毕业设计时，当选择什么作为 CAN 收发器的驱动时，还是看了很多的资料的。其实现在市面上对于我们这样的毕业设计相关规模的芯片，主要是 TI 公司的 VP230 和 PHILIPS 公司的 TJA1050 或者 TJA1040，在经过查阅资料 and 比较性能后，考虑到它们的功能、作用、功耗和性能都相差无几，最后从价格入手，选择了相对便宜的 VP230。

下面简单的对 VP230 做个介绍。VP230 现在经常是用于有着较大抗干扰性，在不同的速率和不同的波特率下都是有着很好的接收和发送的能力，它主要的特点是：

①具有着很好的兼容性，完全能够适应 ISO11898 标准和 ISO115645 标准，完全具有向下兼容的特点，适应多个不同版本的 CAN 总线协议。

②支持多个节点的加入，最多可以接入 120 个节点，能够充分的适应网络的多元化，保持网络的丰富性。

③在通信的时候，节点与节点之间的通信数据交流很快，最快的时候可以达到 1 兆每秒。

④具有自动保护电路的功能，在低电流的时候可以进行自动等待模式，充分的节约了资源，此外它还有抗热拔插的功能，可以做到热保护。

⑤它在工作的时候是采用差分信号接收的，是不断地识别 CANH 和 CANL 上的电平信号的。可以做到大范围的抗共模干扰、抗电磁干扰、降低射频干扰，从而做到保护 CAN 总线的目的。

在实际的操作中，我们在使用 VP230 的时候，CAN 控制器的发送 TX 和接收 RX 分别是和 CAN 收发器的 TXD 和 RXD 连接的，此外还需要 RS 进行 VP230 工作模式的选择，其中 VP230 主要有三种工作模式，其如下表 3-1 所示

表 3-1 VP230 的工作模式

VRS	工作模式
$VRS \geq 0.75V_{cc}$	等待模式
10 千欧-100 千欧接地	斜率控制模式

在本次毕业设计中，我们可以清楚的从 3-11 的硬件电路图中看到，RS 端是直接连接的 GND，因此我们是选择的高速模式，它可以非常保障收发器以最快的速度到达最快的通信速率，能够促使主机与从机之间在通信时，数据信息的传递是高效的。

二、JLINK 程序下载电路的设计

本毕业设计采用的是 20 脚的 JLINK，其中它的主要的硬件连接如图 3-10 所示。从这当中，我们可以看到该 JLINK 的调试接口连接主要是 TRST、TDI、TMS、TCLK 和 RST 这几个引脚的连接。当程序正确的下载后，在 KEIL 中就可以直接进行 JLINK 的 DEBUG 了。并且我们使用 JLINK 进行调试的主要的目的是它是可以进行在线调试的，这样就可以帮助我们最快的找到程序的 Bug。

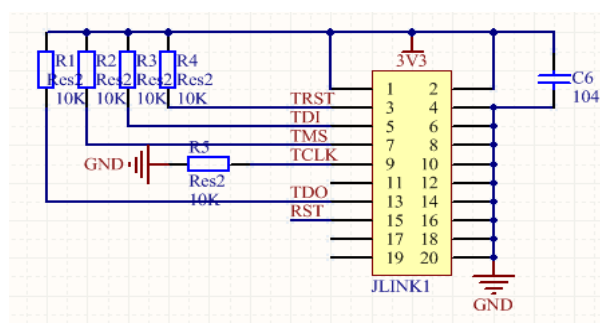


图 3-10 JLINK 接口电路

在通常的 JLINK 的调试程序中，它是经常配合各种编程软件；此外支持多种基于 ARM 的处理器，可以与 IAR、Keil 等编译环境配合使用，快捷、简单，容易上手。其主要特点及产品规格如下：

- ①可以与 IAR，Keil 等编程软件使用。
- ②支持多种 MCU 的多种内核的芯片。
- ③下载速度最高可以到达 1MB/s；甚至其的最高频率可以达到 15MHz。
- ④采用的是 USB 供电，电压要求是 1.2V-1.5V，需要的电流需要小于 50mA。
- ⑤完全的即插即用。
- ⑥具有 J-link 的网络调试的插口，完全可以在进行网络方面的测试时，使用 J-link 进行调试。
- ⑦标准的 20 芯 JTAG 仿真插头。
- ⑧同时它也是可以支持 14 芯的 JTAG 仿真接头。

三、直流电机驱动电路的设计

挖掘机模型采用的是 6 个 130 直流减速电机来提供它的动力的，同时也是 130 减速电机和减速齿轮的配合，才能完成挖掘机的前进、后退、左转、右转并且完成挖掘的动作。对于在电机驱动的选择上，我是根据具体电机的功率，所需的电压等，选择的是 L9110S 驱动芯

片。如图 3-11 所示是从机的主控芯片和电机驱动芯片的硬件电路的连接方式。其中可以非常清楚的看到 L1-L6 是 6 路电机驱动芯片，采用的是 7V 左右的电压直接对它进行供电，输入端是直接可以和 STM32 单片机的 I/O 进行连接的，输出端同时也是直接与电机的两条线路进行连接的。

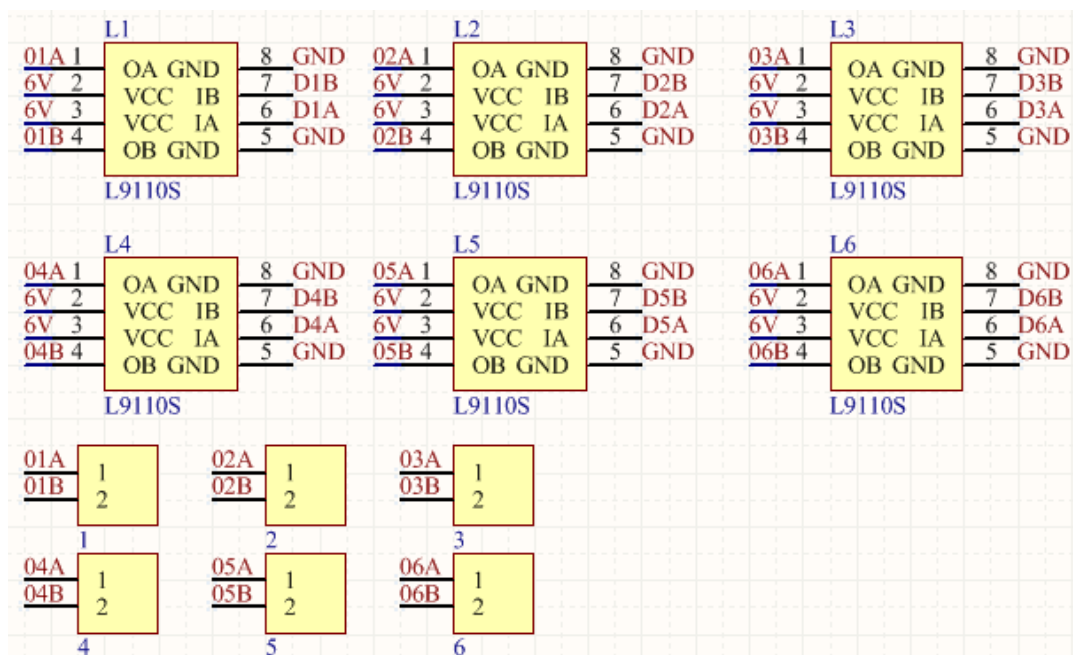


图 3-11 电机驱动电路

在本次设计所用的驱动芯片是一种能够驱动直流电机转动的高度集成的驱动芯片。它的稳定性非常的高，一般都是可以保证电机的正常的运行的。它的驱动能力也是很强的，大约可以输出 800mA 左右的电流。并且该芯片是有强的抗干扰性。由于现在的 L9110S 的功能是如此的强大，目前，它经常会用于玩具车等领域，有着很大的前景。

现在，我们已经知道了 L9110S 有着很多的用途，那么我们应该怎样才能够正确的使用该芯片。我们应该知道它是怎样工作的、电气规则是怎样的、内部的结构图是怎样的、驱动控制的逻辑真值表是怎样的，只有做到了这些，我们才能够叫做学会了该驱动的原理，并不是指局限于使用它。

L9110S 驱动芯片各引脚的功能和如何使用、驱动控制的逻辑真值表和内部结构的功能图分别如下表 3-2、表 3-3 和图 3-12 所示：

表 3-2 L9110S 引脚功能图

1	OA	A 路输出	5	GND	地线
2	VCC	电源电压	6	IA	A 路输入
3	VCC	电源电压	7	TAIB	B 路输入
4	OB	B 路输出	8	GND	地线

表 3-3 L9110S 驱动控制逻辑真值表

IA	IB	OA	OB
H	L	H	L
L	H	L	H
L	L	L	L
H	H	Z（高阻抗）	Z（高阻抗）

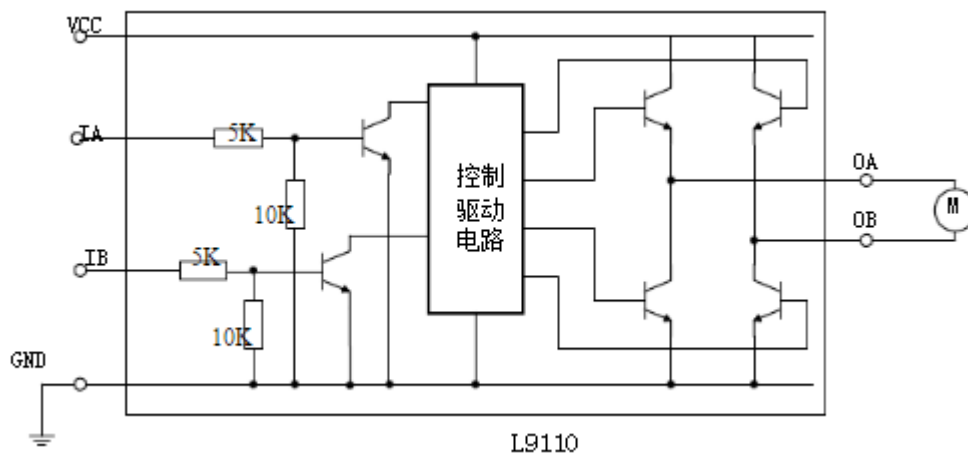


图 3-12 L9110S 内部结构图

130 电机是一种直流减速电机，它通常是运用于各种玩具车、或者一些小型的风扇中。在本毕业设计中就是采用 130 直流减速电机配合减速齿轮来为挖掘机提供动力，并且设计时利用软件做到了定时、延时和控制电机的正反转，可以简单的做到了控制挖掘机的前进和后退，并且还能够基本上完成挖掘机的挖掘动作。下图 3-13 是 130 电机和减速齿轮组。



图 3-13 130 电机和减速齿轮组

130 减速直流电机的具体参数如下：

- ①额定转矩：6.31G.CM（NM）；
- ②额定功率：0.75W；
- ③额定电压：3.0V；
- ④额定电流：0.86A；
- ⑤额定转速：8530r/min；
- ⑥外形尺寸：25*20mm；

⑦效率：45.6%。

四、主机按键控制电路的设计

本次的毕业设计中，为了能够更加形象的模拟出挖掘机在正常工作时的控制情景，我在本次设计中运用了 5 个按键来模拟实际的情况，因为每个按键得到连接方式都是相同的，这里选出 1 个按键和主机的 STM32 主控制器之间的硬件电路连接的电路图如下图 3-14 所示：

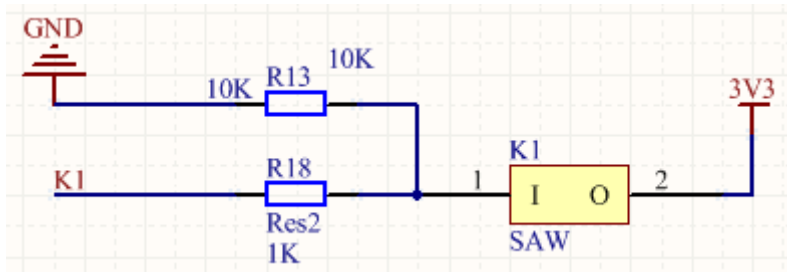


图 3-14 按键电路图

我们可以从前面的原理图中知道，SAWK1-SAWK5 分别是 5 个按键，再就是 K1-K5 分别连接到 STM32 的 5 个普通的 I/O 口，这样就可以保证正常的连接，再就是电路是采用的下拉的接入方式，下拉电阻是 1 千欧姆。电路是非常简单明了的，我的主要目的是当按键按下时就可以代表不同的信号，例如是挖掘机的前进或者后退信息，这种信号或者是数据可以通过 CAN 总线进入到网络的，这样就发送给了从机，从机做出相应的动作。这种使用按键来模拟的方式是很方便的模拟了挖掘机实际操作时的指令。

第五节 本章小结

在本章中主要是先对整个硬件电路进行了一个总的概述，让我们有个总体的认识；再是具体的介绍了毕业设计中用到的 STM32F103XX 系列的芯片，其中主要是介绍了它的最小系统中包含电路，再是对 STM32 芯片的功能和资源做了简单的介绍；最后是比较详细的介绍了围绕主机和从机的主控制芯片 STM32 的外围硬件电路的设计，主要是简单的介绍了在整个毕业设计中最为重要的 CAN 总线收发器的硬件连接，其次是下载程序和调试 Bug 时用到的 J-LINK 电路，再是从机所用到的驱动芯片 L9110S 与 STM32、6 路电机的连接，最后是主机中 STM32 和按键的连接电路。只有在让硬件电路设计好后，让它们每个部分、每一个模块、每一个结构都可以恰到好处的配合，才可以发挥出最大的功效，才可以最大程度上体现后续软件的设计，才可以近乎完美的设计出符合要求的挖掘机。

第四章 CAN 总线挖掘机的底层驱动设计

第一节 底层软件任务分析

在上一章中详细的介绍了本次设计中的主要硬件电路的设计情况，在这里主要是讲述我们主机和从机之间是怎样通过 CAN 总线协议进行通信的，从机和电机驱动之间的接口程序是怎样设计的。我们在设计软件的底层代码的时候，要充分的考虑到硬件设计这样的原因，这样才能写出功能完美的软件，发挥硬件的功能。那么要想设计出比较完美的程序，那么我们应该对本设计需要完成什么进行分析，看懂底层软件设计的目的，任务是什么，这样才可以将挖掘机所要表现的东西全部依托程序的控制程序呈现出来。

对底层软件设计时进行任务分析是软件设计之前的一个程序的规划。在进行设计分析时，先要明确知道程序大概的流程，明确先做什么，在做什么^[11]。然后就是在有一个总体的认识的情况下，对主机和从机的控制主芯片和驱动和 CAN 总线有个初步的认识，在自己的学习能力之内的运用合适的模块。最后将所有模块可能会用到的寄存器和芯片的硬件资源都做一个统一的规划，以便后续程序设计的需要。

本次设计的主要任务是硬件驱动控制电路板的设计。首先是要实现主句和从机之间可以通过 CAN 总线进行数据交流。再是从机主控制器 STM32 能够正常的驱动 6 路 130 直流电机的正反转，完成 STM32 与驱动 L9110S 的接口程序。最后是主机中按键扫描程序的编写，能够将按下的是哪一个按键的信息反馈给主机的 STM32 控制器。只有将这三个部分都完成了，只要可以通过按键来控制任意一个电机正转或者反转，我们就可以进行完善，让挖掘机模型在电机驱动的情况下做出多种动作，让挖掘机可以动起来，可以展现出它的趣味性，程序只是涉及到了挖掘机运动程序的底层开发，主要是对 CAN 总线协议的运用，如何发送出正确的数据报文，并且在接收端也可以接收并且正确的解析出报文的信息，对驱动电路的驱动控制，内部的定时、延时和按键扫描等，这里根据以上对本次设计的软件部分底层任务的分析，可以大致的设计出主机和从机程序的流程图如图 4-1 和 4-2 所示：

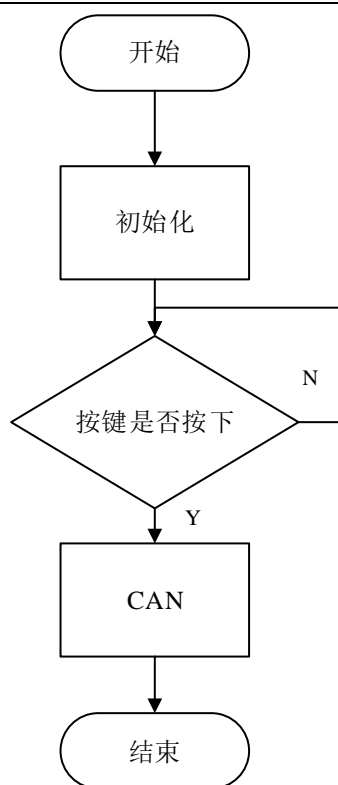


图 4-1 主机流程图

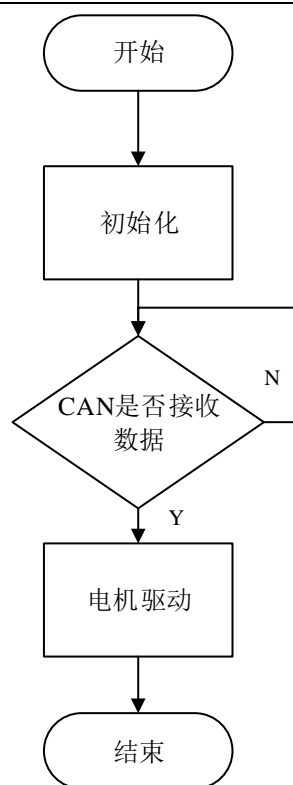


图 4-2 从机流程图

第二节 关键模块驱动程序设计

一、STM32 的初始化

在对软件部分进行编写的时候，必须要知道 STM32 对整个软件部分实现的作用，没有它或者它出错是不可能完成设计的，我们所有的运用都是在它的基础上完成的。因此在进行程序编写的时候，最基础的、最容易出错的就是 STM32 的初始化的设置。系统的配置主要包含系统的时钟，功耗管理和睡眠模式等，外围的基础电路的配置就是所有其他模块的初始化、功耗管理、引脚所要连接的 I/O 的配置和相应的寄存器的配置^[12]。在跑程序的时候，SystemInit(void)函数里面就包含了 STM32 系统的所有的初始化工作，接着就是一些所要用的模块或者部件的初始化了，具体的初始化源代码见附录。

其实我们在一般的程序设计的时候，一般都是会用到 TI 公司的 STM32F103XX 系列单片机的库函数的，这样其实就很方便了，很多函数我们是可以不用去写的，是可以直接去调用的。因此在 STM32 初始化的时候，我们也是直接用它的接口函数就可以，只需要完成时钟配置，flash 配置，寄存器使能，中断配置和其他相应的系统资源配置，没有使用的或者没有必须的是不用对其进行初始化的。

二、CAN 总线协议的配置

CAN 总线协议可以说是本毕业设计的最重要的，因为本次设计的所有通信都是建立在这上面的，只有主机和从机的 CAN 收发器可以正常的发送和接受的数据，并且 STM32 的 CAN

控制器可以识别这些信息，才可以进行下一步的其他工作。下面，这里将简要的介绍主机和从机通信的 CAN 总线协议的配置。

在本设计中主机和从机之间是采用的是的标准是 ISO11898，CAN 控制器上的原理是根据 CANH 和 CANL 两条线上的电位差，显性电平用 0，隐性电平用 1，这样就是将通信中要代表的模拟信号用 0 和 1 的方式转换为数字信号，在线路上进行传递。这样就是通过线路上电平的变化，将这个信息所代表的信号传递出去。

CAN 总线在进行数据信息的传递的时候，所有的数据信息都是通过帧的形式来进行通信的，其中其主要的帧类型如下表 4-1 所示：

表 4-1 CAN 协议帧类型

帧类型	帧用途
数据帧	发送单元向接收单元传递的数据的帧
遥控帧	接收单元向具有相同的 ID 的发送单元请求数据的帧
错误帧	检测到错误而发出指令的帧
过载帧	用于还没有准备就绪接收数据的帧
间隔帧	将数据帧和遥控帧分开的帧

这里，本文对数据帧做简单的介绍，因为它是 CAN 收发器接收到数据信息时，再到 CAN 控制器，对它进行解析，判断它是否是需要的包的^[13]。数据帧主要是有 7 个段组成的，其中包括帧起始、仲裁段、控制段、数据段、CRC 段、ACK 段、帧结束^[14]。

下面对本文中用到做判断的段进行解释，在发送数据的时候，要发送一个数据帧是需要配置很多东西，因为按照 CAN 总线的不同的标准，设置也是不同的。这些很困难的配置，其实很多是不用自己配置的，一般 STM32 中的控制器是当我们在初始化函数的时候就给我们配置好了。我们只需要在程序中对数据帧的控制段、数据段 进行判断就可以了。

下面是主机和从机的基于 STM32 的 CAN 控制器的配置程序：

```
static void CAN_Mode_Config(void)
{
    CAN_InitTypeDef      CAN_InitStructure;
    CAN_DeInit(CAN1);
    CAN_StructInit(&CAN_InitStructure);
    CAN_InitStructure.CAN_TTCM=DISABLE;
    CAN_InitStructure.CAN_ABOM=ENABLE;
    CAN_InitStructure.CAN_AWUM=ENABLE;
    CAN_InitStructure.CAN_NART=DISABLE;
    CAN_InitStructure.CAN_RFLM=DISABLE;
    CAN_InitStructure.CAN_TXFP=DISABLE;
```

```

    CAN_InitStructure.CAN_Mode = CAN_Mode_Normal;
    CAN_InitStructure.CAN_SJW=CAN_SJW_2tq;
    CAN_InitStructure.CAN_BS1=CAN_BS1_6tq;
    CAN_InitStructure.CAN_BS2=CAN_BS2_3tq;
    CAN_InitStructure.CAN_Prescaler =4;
    CAN_Init(CAN1, &CAN_InitStructure);
}

```

这个函数的主要目的是对主机和从机的主控芯片的 CAN 控制器进行配置。设置它们涉及的相关的 CAN 的寄存器。主要的功能是将其设置接收和发送数据包的过滤器组为 0，过滤器的位宽为 32，还有就是在过滤器中的 32 位的数据要其中的高低 16 位进行匹配。

在从机的 CAN 总线的接收中断中，其就是对数据帧的比较，其中有扩展 ID、扩展模式、发送的数据和数据长度。这里只选出了一段代码，是按键的一种情况的信息和标识量的程序。具体见附录源程序。

```

void USB_LP_CAN1_RX0_IRQHandler(void)
{
    CAN_Receive(CAN1, CAN_FIFO0, &RxMessage);
    if((RxMessage.ExtId==0x1314) && (RxMessage.IDE==CAN_ID_EXT)
        && (RxMessage.DLC==2) && ((RxMessage.Data[1]|RxMessage.Data<<8)==0x1111))
    {
        getnum = 0x11;
    }
}

```

三、软件延时程序

在本设计中的软件延时，是根据单片机外接的时钟晶振的大小来达到振荡周期，从而可以知道在 CM3 内核中的 SysTick 定时器的初始值，这样就可以准确的去设定延时函数 void Delay_ms(u32 nms)或者是 void Delay_us(u32 nus)。在我的毕业设计中，有两处要用到延时，一是按键扫描是的去抖，二是挖掘机做动作时电机要转动多少度用到了延时。因为挖掘机的运动基本上都是通过控制直流电机来实现的，并且机械结构很简单，为了能够让挖掘机更加形象的展示它的运动效果，使用延时函数是不可或缺的。在程序的编写是，采用是在每个动作要延时，是让程序跑空指令来实现的。这样做到目的是操作是很简单的，并且效果是杰出的，而且是高效的，可以通过此方法来达到目的。综合考虑，在本次设计中，我们设计的是 50ms 为基本单位函数。在调用时，是直接改变函数中 N 的大小，N 是多少，就是多少个单位的延时。不同的数值，就会让电机转动不同的角度，从而就可以配合起来完成不同的动作了。

其中所用到的延时函数具体的子函数如下：

```
void Delay(uint N)
{
    uint i,j,k;
    for(j=0;j<N;j++)
        for(i=0;i<1000;i++)
            for(k=0;k<1000;k++);
}
```

四、电机驱动程序

在本次设计的挖掘机中，基本上所有的动作都是通过直接控制电机的转动来实现的。在设计的时候，对 6 个电机都进行了标号，在挖掘机所要完成的每一套动作中，根据挖掘机动作的步骤，设计出电机转动的顺序和正反转。这样就可以代表出挖掘机相关的关节的状态。

在设计程序的时候，本设计一共是设计了 6 种挖掘机的运动情况，考虑到它一共是 6 个电机带动的，为了让每个电机都可以展示它的正反转，本设计设计了大体上的 3 中动作。首先是位于履带式底部的两个电机，通过这两个电机的正反转就可以表现挖掘机的前进和后退；然后是位于中部的电机，它可以让挖掘机左转和右转；最后是挖掘机手臂上的 3 个电机的一起转动，来实现挖掘机的挖土动作，这里就主要是系统的向下和抬手臂。

接下来抽取在挖掘机的控制过程中最复杂的手臂的挖掘动作的程序，并且对该程序做简单的介绍。其主要的功能程序见下，其他部分的程序，例如挖掘机的前进和后退、左转和右转，见附录中的源程序出。

```
void way_go()
{
    wawa1_down();
    Delay(0xffffFF);
    wawa1_stop();
    wawa2_up();
    Delay(0xffffFF);
    wawa2_stop();
    wawa3_up();
    Delay(0x4fffFF);
    wawa3_stop();
}
void way_back()
{
    wawa1_up();
    Delay(0xffffff);
```

```

wawa1_stop();
wawa2_down();
Delay(0xfffff);
wawa2_stop();
wawa3_down();
Delay(0x4ffff);
wawa3_stop();
}

```

在本段代码中，其实都主要是调用已经写好的子函数的。从本次的设计只，主要是模拟挖掘机的工作情况的，其中本设计这这一步骤中主要是设计了两个阶段的，一个就是挖掘机动臂向下挖的动作，是函数 void way_go(); 再就是挖掘机动臂向上抬的过程，是函数 void way_back()。在这两个大函数中调用的是分别控制着手臂中 3 个电机的正反转和延时函数，这样就可以很好的控制挖掘机的这个动作了，效果也是很显著的。

五、按键扫描程序

按键扫描程序的主要功能是相当于主机中一个指令的作用，相对于前面已经讲述了 CAN 总线协议，本设计就是将当按下某一个按键的时候，它所代表的信息将由主机中的 STM32 的 CAN 控制器通过 CAN 总线将其发送出去。这样在从机得到信息的时候，就可以在通过从机的 STM32 来控制电机转动了。

在本设计中，主要是采用了 5 个按键，它们就好比是实际中挖掘机控制人员的操作杆。它们在硬件中的分布是上、中、下、左、右和中间，它们代表的意义分别是对应于挖掘机的前进、后退、左转、右转和手臂的挖掘。在主机的程序中，就是不断的就行扫描，然后当有按键按下的时候，就给这个信息有 STM32 发送出去，从而达到控制的目的。其中一部分按键控制程序如下：

```

if( Key_Scan(GPIOB,GPIO_Pin_12) == KEY_ON )
{
    Keynum2 = 1;
    Keynum1 = 0;
    Keynum3 = 0;
    Keynum4 = 0;
    Keynum5 = 0;
    CAN_SetMsg();
    CAN_Transmit(CAN1, &TxMessage);
}
u8 Key_Scan(GPIO_TypeDef* GPIOx,u16 GPIO_Pin)
{

```

```
if(GPIO_ReadInputDataBit(GPIOx,GPIO_Pin) == KEY_ON )
{
    Delay(10000);
    if(GPIO_ReadInputDataBit(GPIOx,GPIO_Pin) == KEY_ON )
    {
        while(GPIO_ReadInputDataBit(GPIOx,GPIO_Pin) == KEY_ON);
        return  KEY_ON;
    }
    else
        return KEY_OFF;
}
else
    return KEY_OFF;}
```

第三节 本章小结

本章主要是详细的介绍了控制挖掘机的主机和从机之间的主要的底层程序的编写，其中主要是包括主机和从机之间的 CAN 总线协议的配置；从机通过 STM32 控制 6 路直流电机的接口程序；主机按键扫描的程序；还有就是整个系统中缺一不可的延时函数。这些主要的底层软件，各个控制模块效果的结合，这样就可以实现挖掘机的各种动作，从而保证整个系统的正常运行。

第五章 CAN 总线挖掘机整体的调试和实现

第一节 关键部分的测试

硬件电路设计和底层代码的正确设计虽然是确保系统正常运行的关键，但是硬件测试还需要对电路中用到的元器件和导线的连接都要进行仔细的排查，软件测试还需要对代码进行检查^[15]。只有确保了所用的模块和芯片都是正常的，每一段代码都是正确的。因此对关键的部分测试时相当重要的。其中主要是在检测电路的时候，下面是排查时主要的常见的问题：

①焊接电路和元件是否正确，确保没有出现焊接不牢和焊接短路等情况。

②元器件本身是有问题的，特别是元器件早就是有问题的，或者是实际通电过程中出现了击穿或者短路的情况。

③各个外设模块的连接是有问题的，或者是没有将其电路共地。

④电路设计本身是有问题的，有些电路设计时没有考虑到实际的电信号的干扰，滤波做的不是很好。

在本设计中，在做好硬件之后，同样也是按照要求对其进行了关键的部分的硬件测试，经常在测试的工程都是需要和简单的软件程序结合起来测试的。下面对这些关键的部分的测试做出说明。

一、电源电路测试

在主机和从机的电源电路都是相同的，电源的稳定输出可以确保系统的正常运行。在对电源模块进行设计是，主机和从机都是 7 伏左右的电源，然后通过稳压模块到芯片的工作电压。在测试的时候主要是要利用万能表去测试主机和从机在上电之后关键节点之间的电位差，在于理论上的值相比较，只要在合理的误差范围内，都是可以接受的。如果测出的电位差存在很大的误差，就需要对具体的元件进行检测，考虑是电源电路的关键的模块是否出现了短路或者是断路，一旦发现是其中的某一种原因，就要马上更换器件了。

在本设计中，制作好的 PCB 板子在焊接好后，经过测试，主机和从机的电源供电模块和各个关键的模块和器件的电压值都是接近于理论值的，是满足要求的，是能够保证主控板稳定工作的。

二、JLINK 接口电路测试

在本次设计中，主机和从机都是采用的是 JLINK 下载电路，它是主板中最重要的部分之一，它是保证编写的软件可以正确调试、运行的基础，只有下载电路正确，才可以保证程序是下载到单片机里面的。

在本次设计中，考虑到驱动电机的转动，其实其本质就是配置 STM32 的 I/O 口。所以在测试系统的时候，就选用的是利用 STM32 来控制点亮一个 LED 灯来测试板子是否可以正常的进行下载程序和正常的进行 CAN 总线的通信。

在本次设计中，主机的板子和从机的板子的 LED 灯是采用同样的电路连接方式的，是可以同样的程序来测试 JLINK 是否可以正常的下载程序的，只要 LED 灯可以点亮，就说明可以正确的运行，JLINK 的接口电路是具有可靠性的。

其中 LED 灯的电路如下图 5-1 所示：

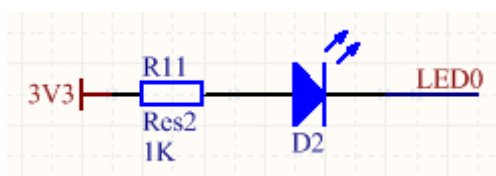


图 5-1 LED 灯

经过测试，在下载程序后，主机和从机上的 LED 灯都是可以正常点亮的，这说明 JLINK 是能够正常的下载程序的。

三、CAN 总线测试

在本次设计中，CAN 总线协议的配置和 CAN 总线的正确通信是整个设计的最关键的部分。为了测试主机与从机的正常通信，在这个测试的时候，主要的思想是，利用主机的按键，当其按下的时候，通过 CAN 总线协议将这个信号发出去，这样当从机接收到这个信号的时候，就点亮 LED 灯。在这个测试中，不仅测试了 CAN 总线是否可以正常通信，还测试了按键的部分是否同样也是可行的。

在经过测试之后，当主机按下按键的时候，从机上的 LED 灯正常点亮。最后得出主机与从机是可以通过 CAN 总线通信的，并且按键的电路连接也是正确的。

四、电机驱动测试

本设计是要通过驱动 6 路电机的转动来控制挖掘机的运动的，那么要测试每一个电机驱动是否可以正常的工作就显得很重要了。考虑到 6 个电机驱动其实电路都是相似的，那么就可以采用相似的程序去检测它们的功能是否正确，只不过是每次都要改变驱动芯片的输入 I/O 口。

主要思路是，先选出一片 L9110S 的驱动芯片进行测试，通过 STM32 编写一个控制电机的程序，观察在输出端是否可以驱动电机。如果可以，就说明驱动的连接电路是可以正常工作的；若不可以，则进一步改进，检查。这只是检测了一块驱动芯片，接着用同样的方法逐一检测其他 5 路驱动。

经过测试，得出 6 路电路都是可以在 L9110S 的驱动芯片下正常工作的。

第二节 本章小结

本章主要是将主机和从机的所有的硬件电路和相关的软件来进行逐一的测试和调试的，其中主要是包括主机和从机的电源电路、主机与从机的 JLINK 下载电路、主机与从机的 CAN 总线的通信、主机的按键连接电路和从机的电机驱动电路。在测试的过程中或者是逐一对它们进行测试或者是几部分一同测试，最终都保证了硬件电路和软件驱动的程序的正确性。最后就可以将每个部分都结合起来，就完成了本设计。

结 论

从开始着手毕业设计到现在完成论文，不到两个月的时间，作为学生这个身份的最后一份作业算是初步完成了，此刻不免难以言尽的复杂情绪萦绕在我心头，既有完成的一丝喜悦，同时完成毕业设计又意味着学生时代的完结，因此喜悦中又夹杂着一丝落寞。不过归根究底，本次设计算是对我大学的最后一次考验，同时也是我的一次再学习，在提高的过程。在毕业设计中，我为能用上四年所学而欣慰，同时我深深的感受到了基础知识的重要性。专业课学习时，老是感觉所学知识大而空，与实际应用相差太远，这种急功近利的思想使自己对一些专业课的学习有所放松，在毕业设计的过程中，我深深的体会到了“书到用时方恨少”的含义，碰到很多情况都是这个我曾经学过，知道一点，但是又不能很好的理解或者解决问题。在以后的学习生活中切不可急于求成而忽略了夯实基础。“万丈高楼从地起”，厚积而薄发，切实做到理论联系实际，学以致用。

针对本次的毕业设计，我设计的是基于 CAN 总线的直流电机驱动控制模块。这道题的设计的主要的目的是，在现在，汽车行业已经逐步的走向了智能化，而随着在汽车中的控制模块的增多，在汽车中的网络布线运用最广泛的就是 CAN 总线。所以本设计就是学习和运用最基础的 CAN 总线节点之间的通信。针对于本次的毕业设计，采用的是两个节点，一个是作为主机，一个作为从机，主机和从机的主控选择的是 STM32F103c8t6，其中主机的主控板在检测到按键按下时，可以通过 CAN 总线将这个信号发送给从机，并且在从机上 CAN 收发器 VP230 在得到信息时，可以在根据信号驱动电机转动。考虑到整体系统的完整性和观赏性。在设计的时候录，我采用的是从机要同时能够驱动 6 路电机，并且将 6 个电机都放置在一个挖掘机模型中。并且我设计了电机的联动，可以当按键按下不同的键后，就可以驱动挖掘机做出相应的动作了，例如：前进、后退、转向和挖掘。

最后，我觉得毕业设计其实好存在一些问题，有多原因是因为自己能力还不足，程序编写能力不足，只是大致上完成了任务的要求。但是，我还在学到了很多，做任何事，不论多么困难，都是要坚持，只有这样，才能够获得最后的成功。

致 谢

写到此处论文也差不多要画上句号，我的大学本科也差不多要画上句号，我不会忘记这短短的几个月时间，我徜徉在书海里查找资料，想起被一个难以解决的问题困扰时的烦躁，想起豁然开朗克服困难时的喜悦，这段经历看似荆棘密布，枯燥无味，实则蕴含着无尽的宝藏。在此我由衷感谢李老师对我的指导和关心，从开始选定题目我还没准备好，老师就已经为我制定了详细而科学的计划安排，并且实时关注研究的进展。在遇到困难时不管是老师还是同学都给我提供了巨大的帮助，我敬佩他们的学识，同时也激励着我向他们学习。当然，此刻需要感谢的不仅仅是帮助我完成毕业设计的人，回顾这大学四年，我需要感谢的人太多了，当我运用到某些知识，脑海里经常会浮现某个老师上课的画面，以后这些都将只能存在于记忆里，感谢这些老师，这将会是我一生最宝贵的财富。

参考文献

- [1] 金娜, 龚娜. 机械智能化控制研究 (J). 中国科技博览, 2012, (11): 204—204.
- [2] 饶运涛, 王进宏. 现场总线 CAN 原理与应用技术 [M]. 北京: 北京航空航天大学出版社, 2007.
- [3] 肖相春. 基于 CAN 总线技术在汽车的实际应用 (J). 装备制造技术, 2011, (8): 85—85.
- [4] 严辉, 叶声华, 刘常杰等. CAN 总线技术在车身视觉检测系统中的应用 (J). 工业仪表与自动化装置, 2004, (1): 26—28.
- [5] 毕霜, 杨勇. 现代电子技术在汽车应用的发展趋势 (J). 中国科技纵横, 2011, (17): 82—82.
- [6] 周亚辉. 液压挖掘机工作装置性能分析和预保护结构动态仿真研究 (D). 东北大学, 2008.
- [7] 向敏, 程安宇, 罗志勇等. 微控制器原理和应用 [M]. 北京: 人民邮电出版社, 2012.
- [8] 张毅刚, 彭喜元. 单片机原理及运用应用 [M]. 北京: 高等教育出版社, 2008.
- [9] 李志明, 徐石明, 丁孝华等. STM32 嵌入式系统开发实战指南 [M]. 北京: 机械工业出版社, 2013.
- [10] 卢有亮. 基于 STM32 的嵌入式系统原理与设计 [M]. 北京: 机械工业出版社, 2014.
- [11] 曾祥春. 任务分析的应用研究 (D). 宁波大学, 2007.
- [12] 勾璇. 基于 iCAN 协议的嵌入式工业控制模块设计 (D). 内蒙古工业大学, 2013.
- [13] 陈金香. 现场总线的低层协议实现 (D). 内蒙古工业大学, 2013.
- [14] 李振林. 浅谈 CAN 总线汽车车载网络 (J). 商情, 2014, (43): 23—24.
- [15] 沈孝凯. 浅谈软件测试自动化管理系统的设计和实现 (J). 魅力中国, 2014, (5): 113—113.

附 录

一、英文原文：

Design Method of CAN BUS Network Communication Structure for Electric Vehicle

Li Ran, Wu Junfeng, Wang Haiying

dept. of Automation

Harbin University of Science & Technology

Harbin, China

zeemoolr@126.com

Abstract

According to the communication structure of CANBUS network on EV, this paper works out a SAE J1939 application layer protocol meet the system functional requirements, and designs the software and hardware for the system. First, design the CAN BUS work node for EV, including the master node, the light node, air conditioning node, doors and instrument node, etc, and draw the CAN BUS topology diagram. Meanwhile, according to the concrete situation of EV, work out an application layer protocol that consistent with the SAE J1939 protocol, and the information allocation table and message structure chart of CAN BUS network node is also presented. Secondly, design the hardware and software for the CAN BUS communication network. Hardware interface circuit mainly consist of CAN communication controller SJA1000, high-speed opt coupler 6N137 and CAN BUS driver 82C250, and design schematic circuit diagram for CAN bus system hardware. The software designs for CAN BUS network are mainly the design of CAN BUS data communication and exchange between nodes, and communication processing for switch signal, analog signal. The design of software communication module includes CAN initialization unit, message sending unit, message receiving unit and the interrupt service unit. Finally, the CAN BUS network communication system the paper designed is applied to the new energy bus produced by CENS Energy-Tech Co., Ltd. which is dedicated in the Shanghai World Expo. From the battery

voltage data collected when EV are running, we can see that the system is accurate, stable without number lost, frame dropping and transfer error in data communication. The design has practical value promoting the application of new energy vehicles.

Keywords: CAN BUS; Communication Structure; SAE J1939

I. INTRODUCTION

CAN BUS is a serial data communication protocol invented by German BOSCH Corporation in the early 80s to realize the data exchange between numerous controllers and measuring instruments in modern automobile. It is a multi-master bus, the communication medium can be a double stranded wire, coaxial cable or optical fiber. Communication speed is up to 1MBPS. Bus communication interface integrates the CAN protocol physical level and the data link layer function, and it can complete the framing of the communication of data processing, including the position filled, the block data code, the circulation redundant check, the priority distinguished and other works[1]. CAN communication protocol feature is to encode the data block. Length of the data is up to 8 bytes, which can meet the electric bus control commands, working status and test data requirement. Meanwhile, the 8 bytes will not take the bus for a long time, so it ensures real-time communication. CAN protocol adopts CRC testing and provides the appropriate error processing, ensure that the data communication is reliable. Now, CAN BUS electric bus technology has become an indispensable part of data communication in the bus [2]. According to the CAN BUS communication network structure, this paper developed a SAE J1939 application layer protocol meets the system requirements, and designed the system software and hardware.

II. CAN BUS COMMUNICATION NETWORK NODE DESIGN

A. CAN Bus Electrical Characteristics

CAN transmission medium formed by the two, One is called high-level transmission line CANH. Another is called low-level transmission line CANL, Ground voltage respectively as VCANH and VCANL. The difference between them is called difference voltage V_{diff} .

B. Hierarchical structure of CAN BUS

According to CAN BUS network structure, Classified into the following five levels, as shown in Fig. 1.

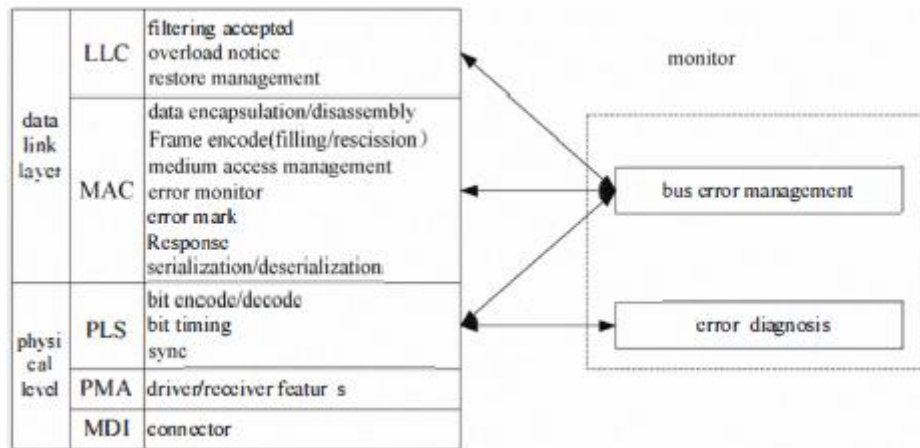


Figure 1. Hierarchical structure of CAN BUS

C. CAN Message Transmission and Frame Type

CAN message use frame as transmission unit. In the CAN 2.0B specification is provided in standard format data frame, Given in two different 2.0B frame format, The difference is the length of different identifiers: Frame with n-bit identifier called the standard frame, the frame contains 29 identifiers is called extended frame. Message transmission has the following four different types of frames: data frame, remote frame, error frame, overload frame.

D. CAN BUS Bits Regularly And Node Synchronous Technology

CAN controller oscillator maximum oscillator tolerance is 1.58%. In the same CAN network, to ensure that each node can communicate properly, the timing of message must be on the corresponding setting. CAN bit rate is divided into four parts, Sync segment, communication segment, Phase buffer segment 1, Phase buffer segment 2. Sync segment used for synchronization bus of different nodes, it was defined as a time range. Communication segment is used for Compensation physical delay time within the network. Phase buffer segment is used to compensate errors for the edge of stage [3]

E. CAN BUS Node Design

CAN BUS system of electric vehicle include master node, lamplight node, air conditioning node, door node, Instrument node and so on. CAN BUS topography of electric vehicle is as shown in Fig. 2. Master node is the central pivot of the body control system, responsible for data communications with all other nodes and handles information, master node mainly from the driver to accept some of the input signal switch. Each child node to detect a variety of quantities such as switch quantity and analog, after process the child node packaged Messages with all the necessary information namely a message sent to the master node, the master node according to the information contained in Messages corresponding processing, including after logic relationship processed then packaged into messages sent to the child node implementation.

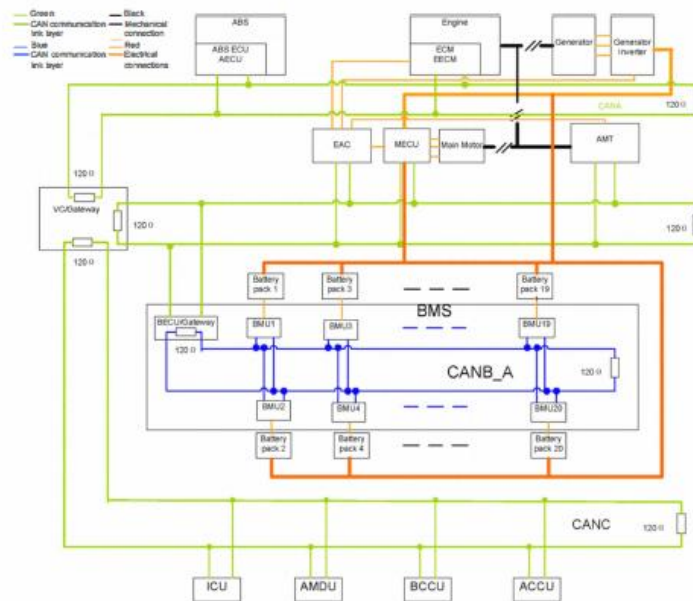


Figure 2. CAN BUS topography of electric vehicle

III. DESIGN OF SAE J1939 APPLICATION LAYER

A. Characters of Application Layer Protocols

SAE J1939 communication protocols (J1939, short for SAE J1939) is developed as a high-level network communication protocol CAN by Control and Communications Network Subcommittee, Society of Automotive Engineer-Trucks and Buses Electrical and Electronics Committee. J1939 protocol defines four layer, the physical layer, data link layer, network layer and application layer. Transfer function of J1939 protocol is a part of data link layer, which has two main functions: information package and restructuring, connection management. In order to send each individual frame to the receiving node to reorganize correctly, the first byte of data field is defined as the package serial number, from 1 to 255. Serial numbers starting from 1 until the end of messages are packaged and sent out in order. For targeted news of multiple packages, the maximum time of the transmission nodes between each packet does not exceed delay 200ms. Each packet must include the 7-byte in original message, and the last one contains 8 bytes of data which includes a byte sequence number that at least one byte is a relevant data to the parameter group and the remaining set to FF. Each data packet is transmitted the receiving end in order, arranged by serial number to compose of a long data and sent to the nodes [4].

B. Units Design of SAE J1939 application layer protocols

System bus communication has the speed of 250Kbps. CAN BUS network packet structure diagram is shown in Fig3.

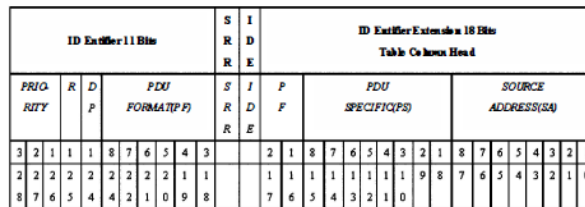


Figure 3. CAN BUS network message structure

The protocol uses 29 extended frames, and the application layer protocol is developed consistent with SAE J1939 protocol according to the specific circumstances of electric bus, whose CAN BUS network node ID information allocation table is shown in Tab. I.

Among them, the priority level being the three positions, it can have 8 priorities; R generally is fixed at 0; DP is fixed at 0; -bit PF is the message code; 8-bit is the destination address or group expansion; 8-bit SA is the source address of sending this message. Data page bit DP is used for selecting data pages. Page 0 contains all the current definitional news, and Page 1 will be used for expansion. DP in this protocol is defined as 0. If the PF domain value is between 0-239 (PDU1), PS field contains the destination address. If the PF domain value is between 240 and 255 (PDU2), PS domain includes the expansion group of PF [5] [6].

TABLE I. CAN BUS NETWORK NODE FUNCTIONAL DESCRIPTION AND INFORMATION ALLOCATION TABLE

Serial number	CAN node name	CAN ID address
1	Stall sign	0x18FFCE00 Byte 1~6
2	Before and after the fog lamp switch symbol	0x18FECA00 Byte 1~4
3	Left and right turn signal sign	0x18FECA00 Byte 5~8
4	Oil warning light symbols	0x18FEEE00 Byte 1
5	Oil Inventory	0x18FEEE00 Byte 5,6
6	Water Temperature	0x18FEEE00 Byte 1,2
7	Oil Pressure Gauge	0x18FEF00 Byte 3,4
8	Speedometer	0x18FF83D0 Byte 1,2
9	Engine tachometer	0x0CF00400 Byte 4,5

IV. CAN BUS INTERFACE CIRCUIT MODULE DESIGN

CAN BUS interface circuit which consists of CAN communication controller sjal000, high-speed photocopiers and CAN BUS driver 82C250. By this operation what read from internal register and write to the internal register, Primary controller can set mode that communication of CAN BUS, it can achieve receive and send data. sjal000 achieve logical encoding and logical decoding when it transmit data, sjal000 improve the capability of the CAN BUS difference send

and difference receive data by CAN BUS driver 82C250. Tx and Rx of sjal000 connect with 82C250 after it pass higher-speed photocopyers 6N137 in order to enhance the capability of the node of CAN BUS anti-jamming and void each other crosstalk. By the way, CAN BUS driver use DCIDC with isolated function, it has achieved to make communication to isolate CAN BUS, it enhance reliability of system. CAN BUS hardware of system circuit of principle is showed in the Fig. 4.

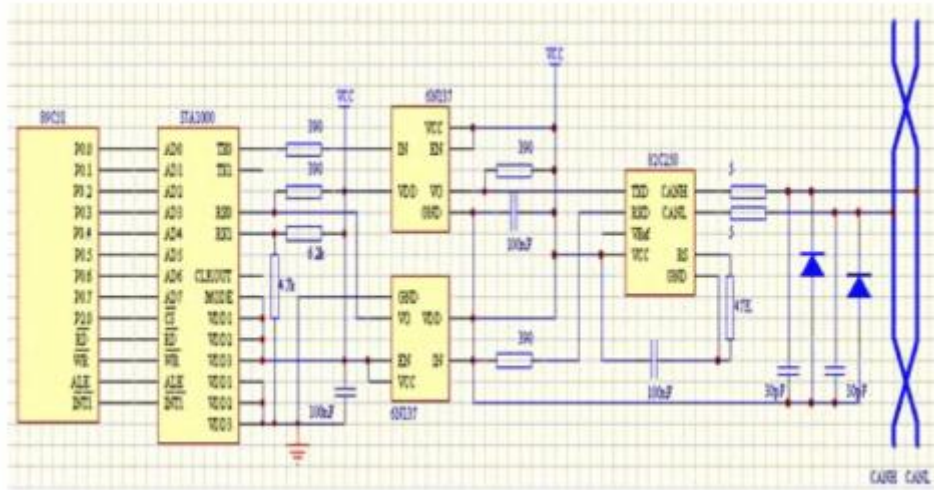


Figure 4. CAN BUS hardware of system circuit of principle

V. CAN BUS NETWORK SOFTWARE DESIGN

The software part of main program unit is divided into: CAN initialization unit, message sending unit, message receiving unit and interrupt service unit. The communication module is responsible for data communication and exchange between each CAN BUS node all over the vehicle, and it also processes some switch-signal and analog signal communications. How to make the software for communication module stable, reliable and real-time is our top priority.

A. CAN Controller Initialization

CAN controller initialization is mainly some sets for its internal register, such as the digital timer set and the mailbox initialization. When initializing the CAN registers in the singlechip, the system immediately clear the read and write buffer, configures the clock frequency register, the code receivingsending register and the interrupt enable register after poweron. After initialization, the single-chip is in normalcommunication status and is ready to work. The initializationcode is as follows:

```
void canCirBufInit(CANCIRBUF *ptCanCirBuf, TCAN DATA
*ptCanFrameBuf, unsigned char ucLength)
{
ptCanCirBuf->uIWriteIndex= 0; ptCanCirBuf->uIReadIndex=
```

```

ptCanCirBuf->bIsFull= false;
ptCanCirBuf->uILength= ucLength;
ptCanCirBuf->ptCanFramBuf = ptCanFrameBuf;
for (unsigned int i = 0; i < ucLength; i++)
ptCanCirBuf->ptCanFramBufli].isExt = 1;
ptCanCirBuf->ptCanFramBufli].rxRTR = 0;
ptCanCirBuf->ptCanFramBufli].dlc = 0;
ptCanCirBuf->ptCanFramBufli].id = 0;
for (unsigned int j = 0; j < 8; j++ )
ptCanCirBuf->ptCanFramBufli].dataU] = 0;
}

```

B. CAN Message Sending

After initialization, the single-chip is in work mode. The CAN message sending adopts inquiry mode. Every 20ms, it checks the data sending buffer for command packet, if there is no command packet, it will return to main program waiting for the next check, and if there is command packet, the CAN controller sending status is on, it sends data in CAN BUS to buffer, the flow of program design is shown in Fig. 5.

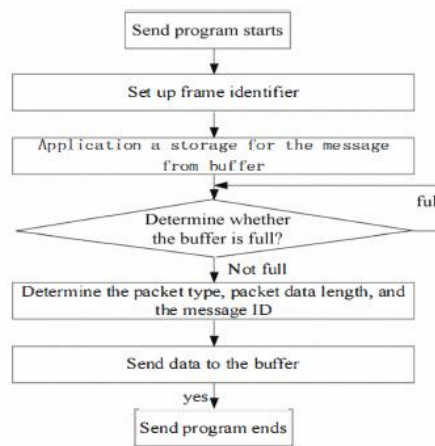


Figure 5. CAN BUS messaging program design process

C. CAN Message Receiving

There are two ways for messages receiving: inquiry mode and interrupt mode. The design adopts the interrupt mode. The single-chip of QY9263K system checks the status register by waiting for interrupting signal. When the interrupt arises, it checks the status flag RBS in buffer for new message. When RBS is 0, it means no new message, keeps waiting. When RBS is 1, it means new message received, the single-chip releases receiving buffer to read the message. The receiving flowchart is shown in Fig. 6.

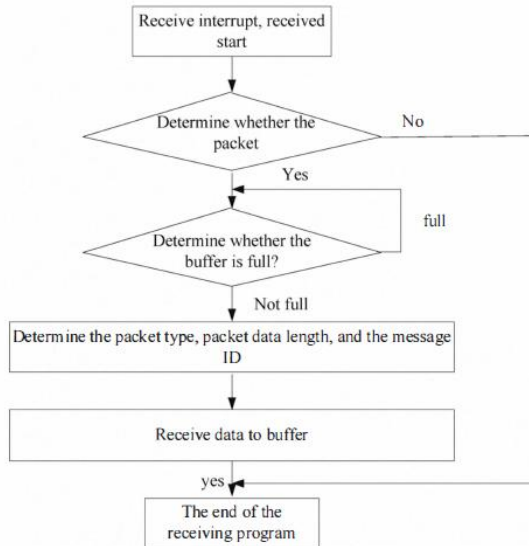


Figure 6. CAN message reception process flow chart design process

VI. DESIGN VERIFICATION

After design the CAN BUS network, including software and hardware, we should verify stability of system functionality, reliability and data consistency. After the system power, select the battery pack combination, a total of 7 groups, on the ody on the electric bus. Record the data that bus instrument receives on the CAN BUS data buffer every 5 minutes. And record the collecting results as follows in Tab. II. Verified that a large number of real-time packet information sanded from the vehicle CAN node is accurate, stable, and no lost number, frame dropping, the number of strings of communication such as abnormal. Stability of system functionality, reliability and data consistency requirements are complied with.

TABLE II. PART OF BATTERY VOLTAGE DATA (UNIT V)

Time	Battery pack 01	Battery pack 02	Battery pack 03	Battery pack 04	Battery pack 05	Battery pack 06	Battery pack 07
00	26.61	26.596	26.62	26.609	26.602	26.624	26.611
05	26.602	26.596	26.616	26.607	26.605	26.62	26.603
10	26.59	26.588	26.6	26.595	26.59	26.612	26.587
15	26.566	26.572	26.584	26.575	26.574	26.592	26.579
20	26.546	26.536	26.552	26.553	26.539	26.572	26.554
25	26.546	26.536	26.556	26.554	26.546	26.556	26.554
30	26.508	26.502	26.518	26.52	26.506	26.531	26.518
35	26.479	26.47	26.49	26.476	26.47	26.495	26.485
40	26.439	26.441	26.464	26.448	26.445	26.464	26.447
45	26.41	26.401	26.42	26.408	26.405	26.422	26.415
50	26.363	26.363	26.386	26.366	26.362	26.385	26.37

VII. CONCLUSIONS

Real-time, reliability and flexibility, all these characteristics make CAN BUS an indispensable network communication technology applied in automobile networkcommunication field. This article based on the analysis of the CAN bussing technique, designed CAN BUS working nodes based on which designed the SAE 11939 application layer design. It also gives the basic software and hardware implementations of CAN BUS network. Finally, tests results show that the hardware circuit of the CAN BUS network communication structure is stable and software design is reasonable, achieved the goal of this design.

二、英文翻译：

CAN 总线网络在电动汽车通信结构的设计方法

摘要

根据 CAN 的通信结构在电动汽车总线网络的运用,本文设计了 SAEJ1939 在应用层协议满足系统的功能需求和设计的软件和硬件的系统。首先,设计电动汽车 CAN 总线工作节点,包括主节点,灯光节点,空调节点,门和仪表节点等,并绘制 CAN 总线型拓扑图。同时,根据电动汽车的具体情况,制定出一个应用层协议一致所述 SAEJ1939 协议,并且信息分配 CAN 总线网络的信息结构图节点也提出了。其次,设计硬件和软件 CAN 总线通信网络。硬件接口电路主要由 CAN 的通信控制器 SJA1000,高速的选择耦合器 6N137 与 CAN 总线驱动 82C250 和设计用于 CAN 总线系统的硬件示意电路图。软件设计的 CAN 总线网络主要有 CAN 总线的数据沟通和交流的节点,以及用于 switchsignal 通信处理模拟信号。软件设计通信模块包括 CAN 初始化单元,消息发送单元,消息接收单元和中断服务单位。最后,CAN 总线网络所设计的节点被利用到该通信系统通过细胞能源科技有限公司,生产的新能源车在上海世博会专用。当电动车在运行时,我们收集到的电池电压数据可以看到,该系统是准确的,稳定的,没有数丢失,帧丢失和传输错误数据通讯。

【关键词】CAN 总线 通信结构 SAE J1939

第一章 简介

CAN 总线是一种串行数据通信协议,它是由德国 BOSCH 公司在 80 年代初发明的,能够实现众多的控制器和测量器之间的数据交换。它是一个多主总线,通信介质可以是一个双绞线,同轴电缆或光纤。通信速度可达 1M/BPS。总线通信接口集成 CAN 协议的物理层和数据链路层的功能,并且它可以完成数据处理的通信框架,包括填充的位置,块的数据代码,所述循环冗余校验,优先级区分和其他作品[I]。CAN 通信协议的特点是编码数据块。数据的长度最多为 8 个字节,可满足电动公交车的控制命令,工作状态和测试数据的要求与此同时,8 个字节将不采取很长一段时间的总线,因此它可以确保实时通讯。CAN 协议采用 CRC 检验和提供相应的错误处理,确保数据通信是可靠的。现在,CAN 总线电动公交车技术已经成为数据的不可缺少的一部分在总线通讯[2]。根据 CAN 总线通信网络结构,本文开发了 SAEJ1939 应用层协议符合系统要求,并且设计的系统软件和硬件。

第二章 CAN 总线通信网络节点设计

A、CAN 总线电气特性

CAN 传输介质由两个形成的，其中一个高级别传输线路 CANH。另一种是低级别的传输线 CANL，对地电压作为分别 CANH 和 CANL。他们之间的不同电压被称为差电压。

B、CAN 总线的分层结构

根据 CAN 总线的网络结构，分为五个等级：MDI、PMA、PLS、MAC、LLC。如图 1：

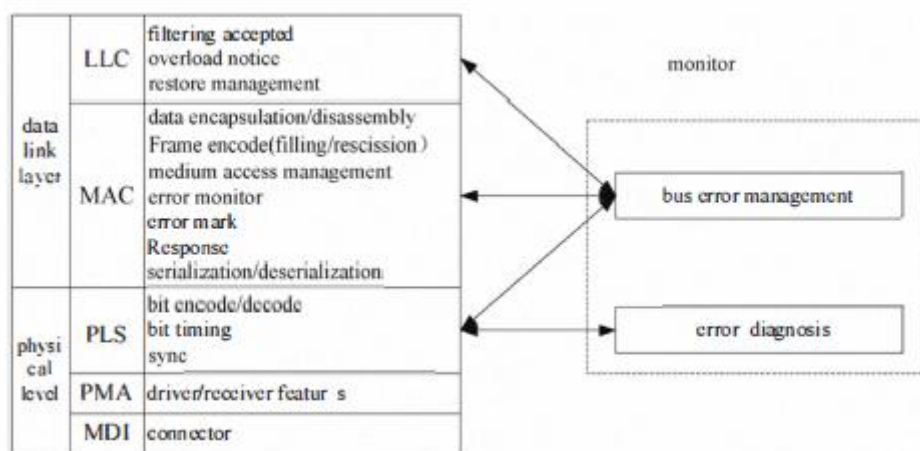


图 1 CAN 总线结构

C、CAN 报文发送和帧类型

CAN 报文是使用帧作为传输单位的。在 CAN2.0B 规范标准格式中，定义了两个不同的 2.0B 帧格式，所不同的是标识符的长度、框架与标识符。包含 11 个标识符的称为标准帧，包含 29 个标识符的帧称为扩展帧。消息传输有以下四种不同类型的帧：数据帧，远程帧，错误帧，超载帧。

D、CAN 总线节点的同步技术

CAN 控制器振荡器的最大振荡容乃 1.58%。在相同的 CAN 网络，以确保每个节点可以正常通信，消息的定时必须是对相应的设置。CAN 位速率被分成四个部分，同步段，传播段，相位缓冲段 1，相位缓冲段 2，同步段不同节点有着同步总线，它被定义为一个时间范围。通信段用于补偿网络内的物理延迟时间。相位缓冲段用于阶段[3]的边缘补偿误差。

E、CAN 总线节点的设计

电动汽车的 CAN 总线系统包括主节点，灯光节点，空调节点，节点大门，仪器节点等。主节点是所述主体的中央枢轴控制系统，负责与所有数据通信其他节点和处理信息，主节点主要来自驾驶员接受一些输入信号开关的。每个孩子节点以检测各种量的如开关数量和模拟，

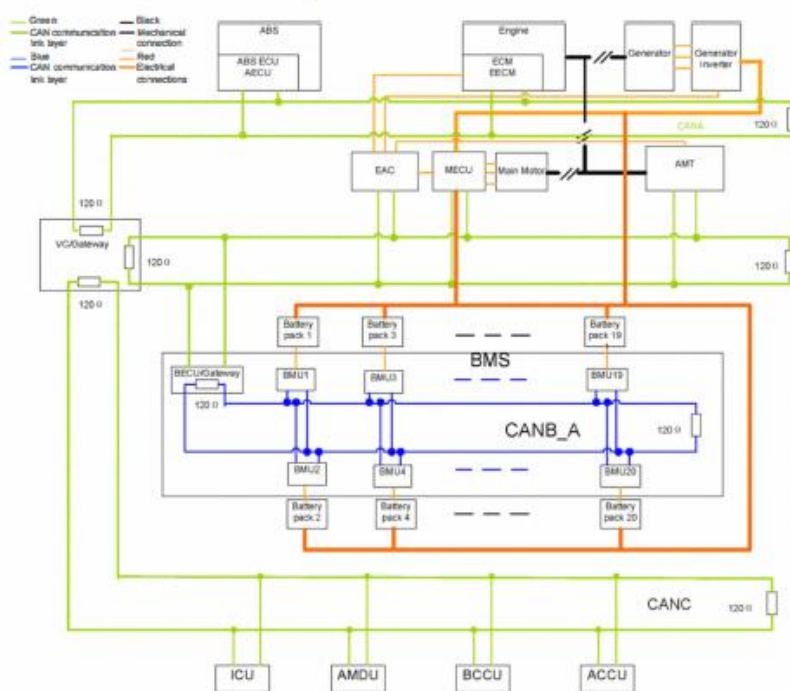


图 2 CAN 总线电气特性

第三章 设计 SAEJ1939 APPLICA 高层次网络通信

A、特征

SAEJ1939 通信协议(J1939, 简称 SAEJ1939)被开发作为一种高层次的网络通信协议可以通过控制通信网络小组委员会, 协会汽车工程师, 卡车和客车电气电子委员会。J1939 协议定义了四个层, 分别是物理层, 数据链路层, 网络层和应用层。J1939 协议的传输功能是数据链路的一部分, 它主要有两个功能: 信息包和重组, 连接管理。为了互送个别帧到接收节点正确地重新组织, 数据字段的第一字节被定义为包序列数, 为 1~255 的序列号开始 froml 直到消息最终被打包并以发送出去。对于多个包的有针对性的消息, 最大时每个数据包不超过延迟之间的传输节点 200 毫秒。每个数据包必须包括在原始的 7 字节消息, 并且最后一个包包含 8 字节的数据的哪一个字节序列号的至少一个字节是一个相关的数据的参数组和其余的集到 FF。每个数据分组被发送接收端按顺序按序号排列以构成一个长数据的和发送的到节点[4]。

B、设计思路

系统总线的通信速度达到 250KB/S，一个总线报文的结构如图 3：

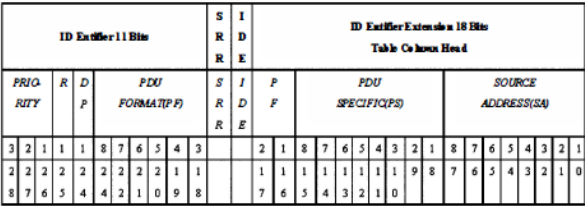


图 3 CAN 总线报文结构

该协议采用 29 扩展帧，并且应用层协议程序开发与 SAE J1939 协议一致根据电动公交车的具体情况，其 CAN 总线网络节点的 ID 信息分配表在选项卡中。其中，优先级是在三个位置，它可以有 8 个优先级；通常固定为 0；DP 固定为 0；8 位 PF 是消息的代码；8 位是目的地地址或者组扩展；8 位 SA 是发送该源地址信息。数据页位的 DP 用于选择的数据页。第 0 页包含了所有当前的定义消息，页 1 将用于扩展。DP 在此协议 defined 为 0。若 PF 域值是 0-239（PDU1），PS 域之间包含目标地址。若 PF 域值 240 和 255（PDU2）之间，PS 域包括 PF 扩张组[5] [6]。

表 1 CAN 总线节点功能描述

Serial number	CAN node name	CAN ID address
1	Stall sign	0x18FFCE00 Byte 1~6
2	Before and after the fog lamp switch symbol	0x18FECA00 Byte 1~4
3	Left and right turn signal sign	0x18FECA00 Byte 5~8
4	Oil warning light symbols	0x18FEEE00 Byte 1
5	Oil Inventory	0x18FEEE00 Byte 5,6
6	Water Temperature	0x18FEEE00 Byte 1,2
7	Oil Pressure Gauge	0x18FEEF00 Byte 3,4
8	Speedometer	0x18FF83D0 Byte 1,2
9	Engine tachometer	0x0CF00400 Byte 4,5

第四章 CAN 总线接口电路模块设计

CAN 总线接口电路组成的 CAN 通信控制器 sjal000，高速度复印机和 CAN 总线驱动器 82C250。通过这一操作读什么从内部寄存器和写入到内部寄存器，主控制器可以设置模式，可以支持 BUS 通信，它可以实现接收和发送数据。sjal000 实现逻辑编码和逻辑解码，当它发送数据，sjal000 提高 CAN 总线发送差异的能力，通过不同的 CAN 总线驱动器 82C250 接收数据。Tx 和 sjal000 的接收与 82C250 连接后，通过更高速复印机 6N137 为了增强的能力 CAN 总线抗干扰的节点，并导致相互串扰。顺便说一句，CAN 总线驱动使用 DCIDC 孤

立功能它已经实现，使通信隔离 CAN 总线，它增强系统的可靠性。具体的硬件电路的连接如图 4:

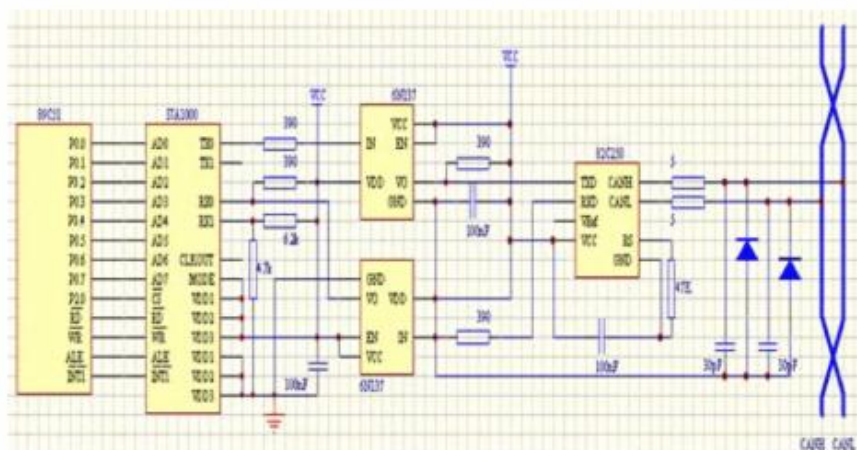


图 4 CAN 总线硬件连接

第五章 CAN 总线网络软件设计

主程序单元的软件部分分为：CAN 初始化单元，信息发送单元，消息接收单元和中断服务单元。通信模块负责数据的沟通与交流每个 CAN 总线节点遍布车辆，在它们之间处理一些开关信号和模拟信号通信。如何使软件通信模块稳定，可靠，实时性是我们的首要优先。

A、CAN 控制器的初始化

主要是一些集其内部寄存器，如数字计时器设定和邮箱初始化。当单芯片初始化 CAN 寄存器系统立即清除读写缓冲区 configures 时钟频率寄存器，代码发送接收寄存器和中断通电后使能寄存器。初始化之后，单芯片是在正常通信状态，并准备工作，初始化代码如下：

```
void canCirBufInit(CANCIRBUF *ptCanCirBuf, TCAN DATA
*ptCanFrameBuf, unsigned char ucLength)
{
    ptCanCirBuf->uIWriteIndex= 0;
    ptCanCirBuf->uIReadIndex=0;
    ptCanCirBuf->bIsFull= false;
    ptCanCirBuf->uILength= ucLength;
    ptCanCirBuf->ptCanFramBuf = ptCanFrameBuf;
    for (unsigned int i = 0; i < ucLength; i++){
        ptCanCirBuf->ptCanFramBufli].isExt = 1;
        ptCanCirBuf->ptCanFramBufli].rxRTR = 0;
    }
}
```

```

ptCanCirBuf->ptCanFramBufli].dlc = 0;
ptCanCirBuf->ptCanFramBufli].id = 0;
for (unsigned int j = 0; j < 8; j++)
ptCanCirBuf->ptCanFramBufli].data[j] = 0;}
}

```

B、CAN 报文发送

初始化之后，单芯片处于工作模式。该 CAN 报文发送协议使用查询方式。每 20ms，它检查数据发送缓冲器对命令包，如果存在没有命令包时，它将返回主程序等待在下一检查，如果有命令包时，CAN 控制器发送状态时，它在 CAN 总线将数据发送到缓冲器。CAN 总线发送流程图如 5 所示：

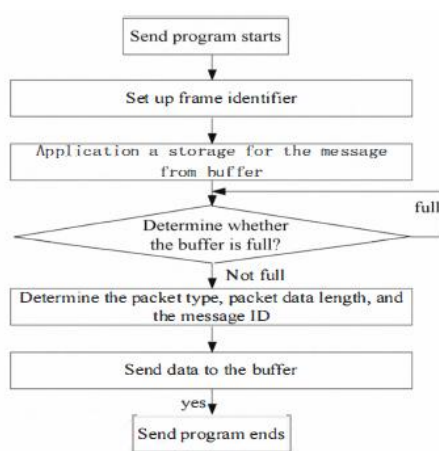


图 5 CAN 总线发送流程图

C、CAN 报文接收

两种方式接收消息：查询方式和中断模式。该设计采用中断方式。该系统的单芯片通过检查状态寄存器等待中断信号。当中断出现时，它检查状态标志 RBS 在缓冲区新消息。什么时候 RBS 是 0，这意味着没有新的消息，一直等待。当 RBS 是 1，这意味着接收到的新邮件时，单芯片释放接收缓冲区中读取消息。其中该过程的接收的流程图如图 6 所示：

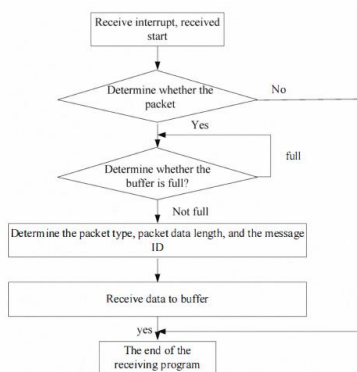


图 6 CAN 总线接收流程图

第六章 设计验证

设计 CAN 总线网络的软件和硬件后，我们应该验证系统的稳定性功能性，可靠性和数据的一致性。记录到数据总线仪器，接收到 CAN 总线数据缓冲区，每 5 分钟一次。有大量的实时数据包从车辆的 CAN 节点发出，信息是准确的，稳定的，并且无丢失的数目，丢帧。通信字符串如发现异常，就得就行修改。稳定性、系统功能性，可靠性和数据一致性必须要遵守。在表 2 中 CAN 总线的测试的一些实验数据：

表 2 部分测试数据

Time	Battery pack 01	Battery pack 02	Battery pack 03	Battery pack 04	Battery pack 05	Battery pack 06	Battery pack 07
00	26.61	26.596	26.62	26.609	26.602	26.624	26.611
05	26.602	26.596	26.616	26.607	26.605	26.62	26.603
10	26.59	26.588	26.6	26.595	26.59	26.612	26.587
15	26.566	26.572	26.584	26.575	26.574	26.592	26.579
20	26.546	26.536	26.552	26.553	26.539	26.572	26.554
25	26.546	26.536	26.556	26.554	26.546	26.556	26.554
30	26.508	26.502	26.518	26.52	26.506	26.531	26.518
35	26.479	26.47	26.49	26.476	26.47	26.495	26.485
40	26.439	26.441	26.464	26.448	26.445	26.464	26.447
45	26.41	26.401	26.42	26.408	26.405	26.422	26.415
50	26.363	26.363	26.386	26.366	26.362	26.385	26.37

第七章 总结

在汽车网络通信应用技术通信领域中，实时性，可靠性和灵活性，这些特性是 CAN 总线的一个不可缺少的网络因素。本文分析了 CAN 的技术，CAN 总线工作在此基础上设计了节点的 SAE11939 申请层设计。这也给了基本的软件和硬件，CAN 总线网络的实现提供了条件。最后，测试结果表明，CAN 总线的硬件，电路网络通信结构的稳定性和软件设计合理。

三、源程序

1、主机主要程序

①按键扫描：

```
int main(void)
{
    CAN_Config();
    LED_GPIO_Config();
    Key_GPIO_Config();
    while(1)
    {
        if( Key_Scan(GPIOA,GPIO_Pin_10) == KEY_ON )
        {
            Keynum1=1;
            Keynum2 = 0;
            Keynum3 = 0;
            Keynum4 = 0;
            Keynum5 = 0;
            CAN_SetMsg();
            CAN_Transmit(CAN1, &TxMessage);
        }
        if( Key_Scan(GPIOB,GPIO_Pin_12) == KEY_ON )
        {
            Keynum2 = 1;
            Keynum1 = 0;
            Keynum3 = 0;
            Keynum4 = 0;
            Keynum5 = 0;
            CAN_SetMsg();
            CAN_Transmit(CAN1, &TxMessage);
        }
        if( Key_Scan(GPIOA,GPIO_Pin_9) == KEY_ON )
        {
            Keynum3 = 1;
            Keynum1 = 0;
```

```

Keynum2 = 0;
Keynum4 = 0;
Keynum5 = 0;
CAN_SetMsg();
CAN_Transmit(CAN1, &TxMessage);
}
if( Key_Scan(GPIOB,GPIO_Pin_13) == KEY_ON )
{
Keynum4 = 1;
Keynum1 = 0;
Keynum2 = 0;
Keynum4 = 0;
Keynum5 = 0;
CAN_SetMsg();
CAN_Transmit(CAN1, &TxMessage);
}
if( Key_Scan(GPIOA,GPIO_Pin_8) == KEY_ON )
{
Keynum5 = 1;
Keynum1 = 0;
Keynum2 = 0;
Keynum3 = 0;
Keynum4 = 0;
CAN_SetMsg();
CAN_Transmit(CAN1, &TxMessage);
}

```

②信息发送

```

void CAN_SetMsg(void)
{
    TxMessage.ExtId=0x1314;
    TxMessage.IDE=CAN_ID_EXT;
    TxMessage.RTR=CAN_RTR_DATA;
    TxMessage.DLC=2;
    if(Keynum1 == 1)
    {

```



```
TxMessage.Data[0]=0x11;
TxMessage.Data[1]=0x11;
}
if(Keynum2 == 1)
{
    TxMessage.Data[0]=0x22;
    TxMessage.Data[1]=0x22;
}
if(Keynum3 == 1)
{
    TxMessage.Data[0]=0x33;
    TxMessage.Data[1]=0x33;
}
if(Keynum4 == 1)
{
    TxMessage.Data[0]=0x44;
    TxMessage.Data[1]=0x44;
}
if(Keynum5 == 1)
{
    TxMessage.Data[0]=0x55;
    TxMessage.Data[1]=0x55;
}}
```

2、从机主要程序

①电机驱动：

```
int main(void)
{
    USART1_Config();
    CAN_Config();
    LED_GPIO_Config();
    while(1)
    {
        if(getnum == 0x11)
        {
            dianji_qianjin();
        }
    }
}
```

```

    Delay(0xffffFF);
    stop();
}
if(getnum == 0x22)
{
    dianji_houtui();
    Delay(0xffffFF);
    stop();
}
if(getnum == 0x33)
{
    zuo();
    Delay(0xffffFF);
    zhuan_stop();
}
if(getnum == 0x44)
{
    you();
    Delay(0xfeFFFF);
    zhuan_stop();
}
if(getnum == 0x55)
{
    way_go();
    Delay(0xffffFF);
    way_back();
}
}
}
}

```

②从机 CAN 总线接收中断

```

void USB_LP_CAN1_RX0_IRQHandler(void)
{
    CAN_Receive(CAN1, CAN_FIFO0, &RxMessage);
    if((RxMessage.ExtId==0x1314) && (RxMessage.IDE==CAN_ID_EXT)
        && (RxMessage.DLC==2) &&
        ((RxMessage.Data[1]|RxMessage.Data[0]<<8)==0x1111))
    { getnum = 0x11; }
}

```

```

if((RxMessage.ExtId==0x1314) && (RxMessage.IDE==CAN_ID_EXT)
    && (RxMessage.DLC==2) &&
((RxMessage.Data[1]|RxMessage.Data[0]<<8)==0x2222))
{ getnum =0x22; }
if((RxMessage.ExtId==0x1314) && (RxMessage.IDE==CAN_ID_EXT)
    && (RxMessage.DLC==2) &&
((RxMessage.Data[1]|RxMessage.Data[0]<<8)==0x3333))
{ getnum = 0x33; }
if((RxMessage.ExtId==0x1314) && (RxMessage.IDE==CAN_ID_EXT)
    && (RxMessage.DLC==2) &&
((RxMessage.Data[1]|RxMessage.Data[0]<<8)==0x4444))
{ getnum = 0x44; }
if((RxMessage.ExtId==0x1314) && (RxMessage.IDE==CAN_ID_EXT)
    && (RxMessage.DLC==2) &&
((RxMessage.Data[1]|RxMessage.Data[0]<<8)==0x5555))
{ getnum = 0x55; }}

```

3、CAN 总线接口程序

```

static void CAN_Mode_Config(void)
{
    CAN_InitTypeDef      CAN_InitStructure;
    CAN_DeInit(CAN1);
    CAN_StructInit(&CAN_InitStructure);
    CAN_InitStructure.CAN_TTCM=DISABLE;
    CAN_InitStructure.CAN_ABOM=DISABLE;
    CAN_InitStructure.CAN_AWUM=DISABLE;
    CAN_InitStructure.CAN_NART=DISABLE;
    CAN_InitStructure.CAN_RFLM=DISABLE;
    CAN_InitStructure.CAN_TXFP=DISABLE;
    CAN_InitStructure.CAN_Mode = CAN_Mode_Normal;
    CAN_InitStructure.CAN_SJW=CAN_SJW_2tq;
    CAN_InitStructure.CAN_BS1=CAN_BS1_6tq;
    CAN_InitStructure.CAN_BS2=CAN_BS2_3tq;
    CAN_InitStructure.CAN_Prescaler =4;
    CAN_Init(CAN1, &CAN_InitStructure);
}

```

```
static void CAN_Filter_Config(void)
{
    CAN_FilterInitTypeDef  CAN_FilterInitStructure;
    CAN_FilterInitStructure.CAN_FilterNumber=0;
    CAN_FilterInitStructure.CAN_FilterMode=CAN_FilterMode_IdMask;
    CAN_FilterInitStructure.CAN_FilterScale=CAN_FilterScale_32bit;
    CAN_FilterInitStructure.CAN_FilterIdHigh= (((u32)0x1314<<3)&0xFFFF0000)>>16;

    CAN_FilterInitStructure.CAN_FilterIdLow=
(((u32)0x1314<<3)|CAN_ID_EXT|CAN_RTR_DATA)&0xFFFF;
    CAN_FilterInitStructure.CAN_FilterMaskIdHigh= 0xFFFF;
    CAN_FilterInitStructure.CAN_FilterFIFOAssignment=CAN_Filter_FIFO0 ;

    CAN_FilterInitStructure.CAN_FilterActivation=ENABLE;
    CAN_FilterInit(&CAN_FilterInitStructure);
    CAN_ITConfig(CAN1, CAN_IT_FMP0, ENABLE);
}
```

4、主机和从机主板



主机



从机