基于PIC12F675的倒车雷达设计

DESIGN OF VEHICLE BACKING RADAR BASED ON PIC12F675

专　　业：　计算机科学与技术（汽车嵌入式系统）
姓　　名：　刘　星
指 导 教 师：
申请学位级别：　学　士
论文提交日期：　2012年6月10日
学位授予单位：　天津科技大学

摘　要

文从可行性分析、需求分析、各部件工作原理、软件环境的搭建、程序设计、测试等方面，阐述了基于单片机 PIC12F675 的倒车雷达的开发过程。

车雷达使用 PIC12F675 芯片进行总体制御，GP2Y0A02YK(sharp)传感器进行测距、蜂鸣器进行报警。在 windows 7 系统下，使用 MPLAB 以及 HI-TECH 编写基于 C 语言的源代码，使用 Proteus 7 professional 进行模拟调试，使用 K150 PROGRAMMER 下载套件进行文件传输。实现了体积小、外观简洁、可调整式全方位侦测、高敏感度汽车报警雷达。

倒车雷达使车辆前进、后退不存在视觉死角，可以提高驾驶的安全性。

关键词：倒车雷达；　嵌入式；　PIC12F675

ABSTRACT

This paper illuminates the development process of the VehicleBackingRadar based on the chip microcomputer PIC12F675 from the aspects of feasibility, demand, operational principles of the components, software environments, program design and tests, and so on.

The VehicleBackingRadar mentioned in this paper equips the chip PIC12F675 as master control and the sensor GP2Y0A02YK (sharp) for distance measurement and the buzzer for warning. The whole warning system involves source code programmed by MPLAB and HI-TECH and based on C, the debugging simulation by proteus 7 professional, the downloading and transfer of kits by K150 PROGRAMMER. In this way, we can make VehicleBackingRadar smaller, more compact and omnibearingly and more sensitive.

The improvement of sensitivity of VehicleBackingRadar can make driving more safely by eliminating the blind zones。

Keywords: Vehicle Backing Radar； Core Chip Controlling Embedded System； PIC12F675

目　　录

第一章　引 言.....1

第一节　背景调查.....1

第二节　背景分析.....5

第三节　汽车倒车雷达的目的和意义.....7

第二章　可行性分析.....9

第一节　技术可行性.....9

第二节　经济可行性.....10

第三节　操作可行性.....10

第四节　计划可行性.....10

第三章　需求分析.....12

第一节　人们对倒车雷达的需求.....12

第二节　芯片功能要求.....12

第三节　对其它器件的要求.....12

第四节　对运行环境及软件要求.....14

第四章　开发原理及详细环境搭建.....17

第一节　蜂鸣器的工作原理.....17

第二节　测距传感器 GP2Y0A02YK 的工作原理.....19

第三节　芯片 PIC12F675 的工作原理.....21

第四节　本倒车雷达的设计原理总述.....29

第五节　实验环境的搭建.....30

第五章　汽车倒车雷达的详细设计.....33

第一节　倒车雷达设计准备.....33

第二节　倒车雷达的详细设计.....33

第三节　成品制作.....43

第六章　功能检验及相关数据测定.....45

总　结.....46

参考文献.....47

致　谢.....48

第一章 引 言

第一节 背景调查

随着人们对汽车人性化要求的提高，汽车嵌入式科技逐渐成为市场的热门，一辆汽车如果没有人性化的嵌入式系统进行辅助，无论从性能，舒适度，还是性价比方面，都会大打折扣，大家认为外国的车比国产车舒服，比国产车性能优异，很大部分是因为外国车具有尖端的车载嵌入式技术。为了进行论述，下面分两方面进行调查。

一、调查各种车载嵌入式科技

（一）独立控温系统

在一些高档车上，为了满足车内不同位置上成员对车内空气温度情况的不同要求，往往将车内空间划分为几个独立的温区，各个独立的温区可以进行不同的温度调节。仔细考虑便可得知其人性化之处，好比在炎热的夏季，车内有老人小孩，如果把空调开至 20 度上下这种年轻人青睐的“舒爽”温度，老人和小孩有可能会不适，如果迁就老人、小孩，作为驾驶者的青年人，可能会因为炎热，不适等原因，心情烦躁，交通事故的发生率也会随之升高。所以说，一个小的嵌入式技术，会另汽车提升到一个很高的档次。概念图如图 1-1 所示。

图 1-1 独立控温系统概念图

（二）GPS 导航系统

现如今，GPS 系统已经是耳熟能详的设备了。虽然 GPS 系统最初是在军事中被使用，但很快也在民用方面有了很高的普及度与很完善的发展。通过民用商业通信卫星，GPS 系统也被引入到汽车内，为驾驶人员指路，从而发展成为了车载导航系统。另一方面，一般被预先制定好的地图，常常会随着街道的发展更新变得无法使用，又难以辨认，往往给人以错误的指令。所以，发展至今，人们更倾向于利用卫星来完成导航以及定位的功能。对于一些强化了语音功能、指路、

定位搜索查询等功能的 GPS 系统更是驾驶员得力的助手，驾驶员可以通过电话，语音交流等方式直接找到目的地的具体位置，性能优良的导航系统甚至会在汽车行驶过程中，以电子语音的方式给驾驶员正确的方向指示，准确的告诉驾驶者转向角度，最短路线选择等等，而不再是只能提示正南正北的“笨蛋”导航仪，而且完全可以实现语音互动。

GPS 由三个独立的部分组成：

1．空间部分：21 个工作卫星，3 个备用卫星。

2．地面支支持系统：1 个主控站，3 个注入站，5 个监测站。

3．用户设备部分：接收 GPS 卫星发射信号以获得必要的导航和定位信息，经数据处理，完成导航和定位工作。GPS 接收机硬件一般由主机、天线和电源组成，操作界面如图 1-2 所示。

图 1-2　GPS 导航界面示例图

（三）HUD 抬头数字显示

抬头数字显示仪(Heads Up Display)，是指把汽车仪表的信息显示在前挡风玻璃上，又称为平视显示系统，它可以使驾驶员不必低头看仪表，就可以了解到当前车辆的相关信息。抬头数字显示仪技术，最初也是应用于军事，是给飞行员用，目的是可以让飞行员不必转头分散注意力就能在挡风玻璃上看到所需的信息。其有优点是，驾驶员不必低头，就可以看到信息，从而避免分散对前方道路的注意力。驾驶员可以避免在抬头低头过程中避免眼部与颈部的疲劳。总之，这种显示系统的作用是提高汽车的安全性。当然这种系统成本昂贵，目前只在少数高档车中采用，如图 1-3 所示。

（四）感应雨刷

感应雨刷的工作原理[1]是用非常灵敏的传感器（雨量 sensor）来感应雨滴的大小，经过汽车中央处理芯片的处理，自动制御雨刷器的工作速度，为驾驶者提供与天气对应的最适视野，从而大大避免了由于视觉问题而引起交通事故的可能性。其工作原理概况如下。

目前被广泛应用两种主流传感器是光学传感器和电容传感器。光学传感器的

依据是光的折射。具体原理是，光学传感器中有一个发光二级光，这个发光二级管会不停的发射侦测光线（锥形发射），如果此时没有下雨，挡风玻璃处于干燥状态的话，所发出的探测光绝大部分会折射回此传感器。但是当下雨天的时候，挡风玻璃上有雨滴，有一部分光会被发散掉，从而导致折射回传感器的光线数量变少，光学传感器是通过这个原理来侦测雨量的。光学传感器的精准程度取决于传感器接受折射光的接收面积，面积越大，传感器就越精准，高精准度的传感器甚至可以这侧到雨滴的数量。

另一种是电容传感器，它是利用水与玻璃介电常数的不同来起到侦测作用的，（水的介电常数为 80，玻璃的介电常数为 2）。一般情况下，是把两条指状金属板放入挡风玻璃的内外层之间，本组指状金属板联通，但不与其他组联通。当不下雨时，内外金属板之间的介电常数为玻璃的介电常数，当降雨时，根据雨量的不同，介电常数会发生变化，从而测量降雨量，如图 1-4 所示。

图 1-3 HUD 抬头数字显示

图 1-4 感应雨刷的雨量传感器

二、调查尖端车载雷达

随着科技的不断发展，现在人们说的车载雷达绝不仅仅是只有声音报警功能的雷达报警器，而是带有更多自动化计算，可视化显示的新一代雷达，虽然名字

没有改变，同样是叫车载雷达，但是其功能已经变得无比强大。

（一）现代汽车载雷达改良之一：倒车视频影像系统

倒车影响系统实际上就是通过在车尾事先安装好的后视摄像头，来採取车后乃至车两侧的路况信息，以视频的方式传递到车前屏幕，供驾驶者观看，比传统单纯的雷达警报更加安全可靠，更加容易理解。

倒车雷达是通过声波或光波探头採取信息，通过声音提示报警的，人对于声音的敏感程度远远小于对图像的敏感程度。所以同步影像的倒车视频影响系统更加安全，易于接受。而且相比一般的倒车影像系统来说，更高端的倒车视频影像系统，更嵌入了倒车引导功能，屏幕上会出现几条虚拟的线，随着方向盘的转动，这些线也会跟着移动，为倒车提供可靠的引导，这项功能是诸多女性驾驶员与新手驾驶员的福音，显示界面如图 1-5 所示。

图 1-5 倒车视频影像系统界面

（二）现代汽车载雷达改良之二：倒车警示系统

在停车场内，一辆直着开进的车与一辆突然倒出车位的车，是很容由于视觉死角等原因发生碰撞的，倒车警示系统就是为了避免这种情况而设计的，介绍如图 2-3，2-4 所示。

图 2-3 倒车警示系统说明 1

图 2-4 倒车警示系统说明 2

在这之上，如果驾驶员是带着方向倒车的话，当车后有要行驶过来的车辆时，系统会自动判断此车辆会不会对你造成影响，如果不会，那么系统会默认为无危险，从而不会给出多余、错误的提示，由此看来这套系统还是非常聪明的。而且，一般给出的提示中，都会附带有别的有用信息，类似于车辆来的方向，车辆的距离等等，而不是单纯的“有车来了”这种粗略的提示。

第二节 背景分析

一、现代汽车嵌入式科技在我国的情况及相关分析

（一）独立控温系统

我国普通车辆并没有普及这项技术，这仍是稍高档次车中，高配版才可以享有的技术，然而这项技术随着日本丰田，日产等依靠车内科技出名的生产厂家在中国市场的技术资源投放，已经离普通老百姓的生活越来越近，相信在不久的将来，这项技术会成为每车必备的东西。

发展前景：B

发展程度：完善趋于饱和。

（二）GPS 导航系统

非常普及的技术，但是也分三六九等，我国目前中等车辆所配设的 GPS 定位系统属于中等水平，与日本等国家有着一定的差距。但是犹豫其关系到卫星，通信，人工智能等科技领域，所以虽然普及度很高，但随着其他这些领域技术的革新，GPS 技术发展的空间空前巨大，与此同时，我国想要快速提高这项技术也有这相当的困难，因为卫星，通信，人工智能这些方面的技术还远远达不到发达国家的标准。

发展前景：A

发展程度：完善中不断革新。

（三）HUD 抬头数字显示

一般来说从军用改到民用的非免费的东西，都是为追求高级感和奢华感而生的，HUD 抬头数字显示就是其中之一。所以，我国国内，无论是外国车还是国产车，几乎看不到这项技术，但这项技术的优秀之处不可否认，那就是方便，加帅气！可是由于对材料以及车辆配件量产公司的要求较为苛刻，使得离普及它还很遥远。

发展前景：A

发展程度：不成熟。

（四）感应雨刷

只存在于部分高级车辆，并且属于未来车辆的技术之一，它和 HUD 抬头数字显示的状况相差不多，虽然感应雨刷能给驾驶员以良好的雨中视野，但由于对技术上以及材料上的要求过大，现在还处于开发期。

发展前景：A

发展程度：不成熟。

（五）倒车视频影像系统以及倒车警示系统

这两项技术已经在国内普及了，尤其是中高档汽车，通过运用更好的显示器等二级设备，是这两项技术变得更加完美。已经很少出现误报警，不报警，影视故障等情况。

发展前景：B

发展程度：饱和等待创新。

（六）泊车引导系统（自动泊车）

又是一个处于生机勃勃的发展阶段的未来技术，由于对传感器，自动制御，人工智能等方面有苛刻的要求，使得它在现阶段普及成为了很困难的事情，但是发展的大方向已经明确，对其他国家性技术（比如卫星）要求并不是很高，是今后几年中汽车嵌入式发展的大热门。但是由于这项技术如果不达到非常完善，会造成很大的损失赔偿问题，想在 3-5 年内在中国普及，恐怕是不可能的。

发展前景：A

发展程度：有很好的前景，正处发展旺盛阶段。

二、对我国汽车业来说，嵌入式技术发展的战略意义

过去 30 年中，很多汽车厂家改良优化自己的汽车都是依靠汽车电子技术的发展。新型车种的研发以及车内设备的更新换代，70%-80%都是来源于车内电子设备的更新，车内电子设备在汽车制造成本与价值中占有越来越大的比例，平均占有水平已从 10%增加到了 40%，高档次车辆中，更是达到了 60%的地步。而

汽车电子设备发展的核心与关键技术，正是电车载入式系统技术，随着人们对车辆安全性、舒适性等人性化特性要求的增高，嵌入式技术成了汽车发展的真正动力。而这种新的市场格局，也为我国国产汽车行业挑战国外汽车品牌创造了新的契机。在很多人的印象里，国产汽车品牌与国际汽车大品牌，在技术方面有着无法逾越的鸿沟，其实事实就是这样，由于国外汽车品牌在技术上，营销上，市场上，品牌效应上的长期积累，使国内汽车品牌难以望其项背。但是随着汽车电子技术突飞猛进的新潮流，作为国产汽车厂家，更熟悉，更了解国人所需，更容易创造出适合中国人的汽车，这就为国产汽车厂商提供了与国际汽车厂商争夺中国本土市场的最佳条件与最优契机。因此，推动汽车嵌入式产业的发展不仅可以帮助我国自主汽车产业在中国市场赢得一席之地，还能增加国产汽车的名声、信誉等，对我国自主汽车产业有着加强信心的极大积极作用。此外，汽车嵌入式属于软件技术产品，软件技术产品并不受国际贸易壁垒影响，如果发展的好，进军国际市场也是轻而易举的事。相对于提升汽车制造技术而言，发展汽车嵌入式技术所需要的成本与社会资源更少，更适应于现阶段中国的市场规则。另一方面，发展好微控相关技术，更能带动起整个产业链的发展。依靠车载嵌入式，首先，中国的国产汽车市场将得到很大提升，随之而提升的必定有设备配件，设备维护等副产业，同时，与汽车嵌入式相似的嵌入式相关产业，类似于只能空调，只能手机等等，必定也会随之受益，从而飞速发展，使我国的嵌入式产业，汽车产业等相关产业链更加拥有活力。与此同时，汽车嵌入式产业开发属于智力密集型产业，发展此产业正好与我国产业调整政策方针相吻合。因此，大力发展汽车电子嵌入式产业不但可以提高中国汽车行业在中国市场乃至国际市场的地位，还能够在国策方针的支撑下，巩固中国的经济基础，可谓一举多得。

三、我国汽车嵌入式技术已取得的骄人成绩

中国在 2009 年提出了出了“核高基” 产业扶植计划，汽车嵌入式行业被列为重点支持的产业之一。在国家政策与资金的大力支持下，我国核高基专项的汽车电子基础软件课题在短短不到两年时间内，已初显成果，目前已推出了一系列已被国际认证并被承认的嵌入式基础产品，实现了国产嵌入式基础软件的零突破，并初步完成了与汽车制造厂商的融合，产品已经应用在一汽大众、上海一汽、奇瑞等十余款国产车型上，装机台套已总数已达到几十万，并开始向欧洲各著名厂商输出。

第三节 汽车倒车雷达的目的和意义

在没有高清摄像机，显示器的情况下，想达到提醒驾驶者，避开障碍，提高驾车安全系数的目的，可以用小型单片机制御电路内嵌蜂鸣器，来完成报警，通

过报警频率的大小，来告诉驾驶者障碍物与车体的距离以及是否需要转向避障。在有限的条件下，花最少的钱，通过所学的知识，制作一个合理、有效、反应灵敏、抗干扰的倒车雷达，来提高驾车安全性。用最小的成本，有效地完成实际所需，是这次毕设的主要目的。

倒车雷达设计从策划，选材，编程，测试，到最后的焊接，完成每一步都需要很扎实很系统的专业知识，所以，努力学习这些知识并试着熟练掌握它将是本次倒车雷达设计对我的最大意义。

第二章 可行性分析

第一节 技术可行性

一、技术分析

本次倒车雷达设计属于单片机制御应用[2]范畴，各大电子网站上都有相关教程，编程样例。由于汽车嵌入式只是电子嵌入式研究的一个方向，所以作为类比，我找了很多其他的类似电控实例以供参考，比如电子控温，定时信号输出，远程遥控等，这些技术已经完全成熟，有系统的教程，通过它们完全可以类推到自己的设计中去，使本次毕设的技术学习方面实际可行。

硬件方面，选择一个带有 TIMER 计时器加 A/D 转换功能，输出在 5V 的标准 PIC 外加一个输入电压在 5V 以下的传感器便可。PIC 以及传感器的使用方法均有 DATASHEET 可供详细参考，只要使用自己掌握的 C 语言就完全可以编写倒车警报程序了。

二、风险分析

由于本次毕设本身的目的就是采用低成本完成任务，所以从实验器材到编写程序所需的软件，总共加起来也不到一百五十元人们币，也没有涉及到他人的相关信息，所以就开发来说，应该可以称为零风险。

但是如果想到后期使用的风险的话，如果本倒车雷达长期未经检查与除尘，或经常受到震荡的话，其敏锐度以及精确度有可能会下降，从而报错警或报警晚，会给使用者造成经济损失。而且，此产品在开发时虽考虑到了外部干扰对单片机功能的影响，也进行了相对代码优化，但实际情况下仍有被干扰而失灵的情况（例：强阳光，车内电信号噪声过大等）。不过使用者只要正常使用，其风险性还是很小的，所以风险分析方面也是完全可行。

三、资源分析

倒车雷达的资源分为硬件资源与软件资源，硬件资源为芯片 PIC12F675，传感器 GP2Y0A02YK，蜂鸣器，7805 三端稳压器，0.33uf 电容，0.1uf 电容，导线若干，小型 电路板，恒温烙铁及焊接套件，220v-12v 变压器，K150 单片机程序下载器，芯片座。这些东西可以在淘宝网上订购，均属于常见器件。软件资源为，Proteus 仿真模拟器，K150 下载器配套软件，MPLAB 单片机开发软件，DATASHEET(PDF 版)，HI-TECH 外挂单片机库环境。这些软件在相关网站均可下载。综上，资源方面完全可行。

第二节 经济可行性

一、硬件成本

传感器 GP2Y0A02YK 为 50 元，PIC12F675 加底座为 6 元，导线一捆 1 元，万用表家里自备，电烙铁及套件 70 元，电路板+电容等小零件 6 元。

二、教材成本

教材主要来源于 51 单片机论坛，中国电子科技论坛等电控方面论坛和芯片本身附带的 DATASHEET，所以教材方面是零成本。

三、劳动成本

组装，焊接等完全是自己动手，并没有找焊接厂代工，所以人力劳动成本为零。产品效果达到预期要求，反应灵敏，免疫一般干扰，有着良好的经济性。

第三节 操作可行性

一、试验，调试时的操作可行性

由于汽车雷达的电源是来源于汽车车内电瓶的，车内是 12V 直流电源，要模拟车内电源，需要将正负极导线保持与变压器相连，然后叫助手帮忙拿障碍物挡住传感器的前端，并移动，来测试装置是否正常工作以及可以正常工作的距离范围。

再拿到室外，做同样的试验来测试装置是否在一般日光下可以正常工作。所以，不需要过于危险复杂的操作，完全可行。

二、装置的操作可行性

此汽车雷达装置是装配在车后的自动报警装置，所以投入使用时，不需要任何特别的操作，就可以自行运作。

第四节 计划可行性

一般的嵌入式开发分为如下几个步骤：

1．了解需求（即：明确功能要求）。
2．进行开发预测。
3．根据需求选择合适的芯片。
4．根据芯片选择其他需要的器件。
5．查询各器件 DATASHEET。
6．查询相关教程，学习并掌握基于 C 语言的开发语言。
7．编写源程序。

8．在仿真软件（Proteus）进行第一次调试（仿真调试）。

9．在面包板上进行第二次调试（实际调试）。

10．将面包板处于不同的实际状况下进行第三次调试（干扰调试）。

11．根据情况改良源代码，检查与开发预测的吻合度，再次进行干扰调试（反复这三步直到可以完全抵抗一般干扰）。

12．器件排布、焊接。

13．成品测试。

14．装盒。

表 4-1 毕设时间安排

阶段	内容	时间
第一阶段	（1）-（4）	2 月 25 日-4 月 3 日
第二阶段	（5）-（7）	4 月 4 日-4 月 20 日
第三阶段	（8）-（11）	4 月 21 日-4 月 30 日
第四阶段	（12）-（14）	5 月 1 日-6 月 1 日

按照要求，此次毕设的开发时间为 2012 年 2 月 25 日至 2012 年 6 月 1 日，大致的时间安排如表 4-1 所示。

从以上安排来看，时间充裕，计划明确，有着绝对的可行性。

如上所述，分别从技术，经济，操作，计划四方面进行了分析，得出此次倒车雷达设计有着绝对的可行性。它满足低成本，高性能的标准，是一个在有限经济条件，有限技术条件下，满足预测需求的优良产品。从而，本倒车雷达设计是可行的。

第三章 需求分析

第一节 人们对倒车雷达的需求

为了了解人们对倒车雷达的需求，通过问卷调查的形式，得出了以下结论：

1．有效距离必须满足在车体距障碍物 15cm 到 100cm 之前。

2．可在正常日光下使用。

3．反应需要很灵敏。

4．小巧方便，超低耗电。

5．监测范围要广，至少在 120°范围左右。

第二节 芯片功能要求

根据人们对倒车雷达的需求，本次倒车雷达设计中的单片机芯片必须具备以下功能[3]。

（一）A/D 转换功能

A/D 转换功能是此次汽车倒车雷达设计的核心功能。顾名思义，A/D 转换即把信号从模拟信号（A）转换为数字信号（D）或把数字信号转换成模拟信号，就本次毕设来说，A/D 转换功能等用途是，把传感器测到距离远近的模拟信号转换成数字信号，来使芯片识别，从而完成对数据的处理。

（二）TIMER 计时器功能

此功能起到定时、延时的作用，因为把 A/D 转换函数等实验代码写在中断服务程序中，所以可以避免单片机过快过频繁执行 A/D 转换功能，同时也可控制蜂鸣器发声频率，是一个非常重要的辅助功能。

（三）尽量少的引脚数

在具备 1、2 功能的同时，还需要一个尽量少引脚，尽量小巧的芯片。这样做的好处是既可以节省空间，又可以省电，而省电就意味着耗电少，耗电少就意味着电力发热少，发热少就意味着寿命长、稳定。所以，PIC12F675 是一个很好的选择。

第三节 对其他器件的要求

一、三端稳压器

三端稳压器的作用是把较高的电压转成一个较低的定电压输出，车辆中的电源电压是 12V，芯片和传感器的输入电压是 5V，所以选择 7805，即：满足把 12V 稳压电压转换成 5V 稳压并恒压输出[4]。

二、传感器

1．必须在 15-100cm 之间有效。

2．必须反应灵敏。

3．必须抵抗一般干扰。

4．输出电压必须在芯片的引脚额定电压可承受范围之内（即：5V 之内）。

5．传感器的电源电压是 5V。

6．转换函数必须是连续函数并在有效区域内尽量单调变化。

7．有充足的 DATASHEET 介绍和相关配置信息，电气规格等文本资料以便于查询和比对。

8．作为一个关键部件，最重要的要求就是，它必须质量好。

综上要求，进行多家比对后，最终选择了夏普日本原装进口的 GP2Y0A02YK。转换函数图，如图 3-1 所示。装配图，如图 3-2 所示。

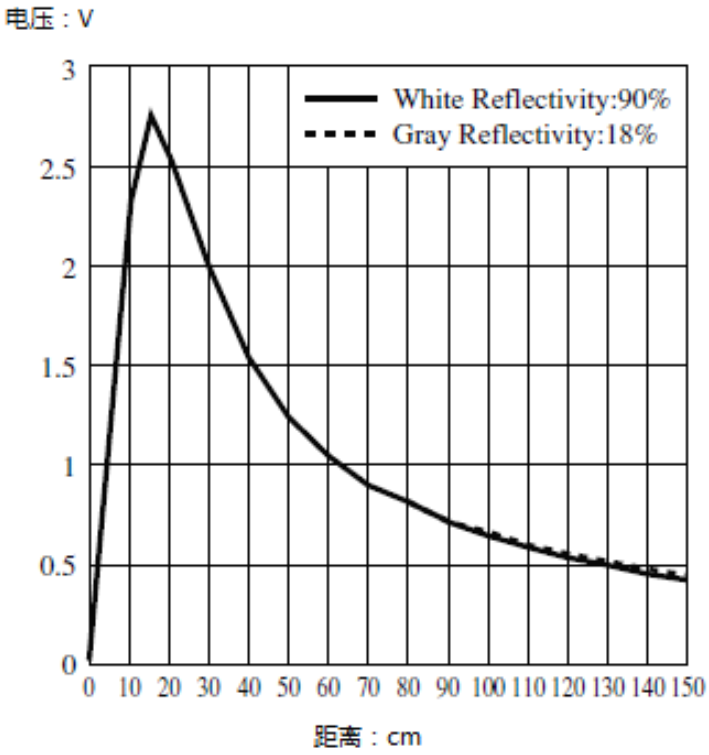

图 3-1 传感器转换函数图

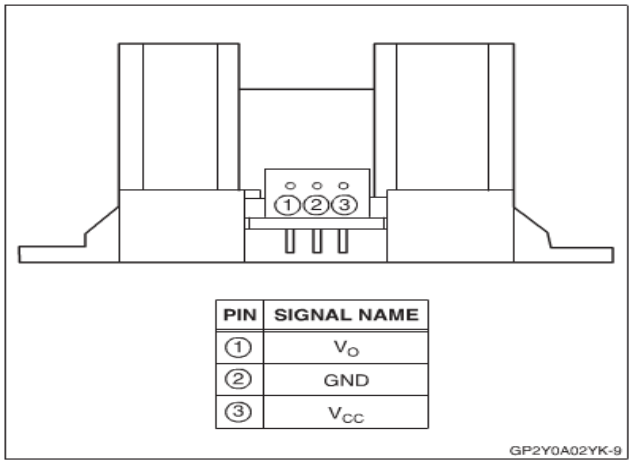

图 3-2 传感器装配图

三、蜂鸣器

要求输入电压为 5V，并无特殊要求。

四、电容、导线、焊接板及装配盒

1．电容：根据三端稳压器 7805 的需求，电容需要 0.1uf 和 0.33uf 各一个。

2．导线：导线为普通铜丝导线即可。

3．焊接板：条形分行导通型焊接板为宜，如图 3-3 所示。不宜用无导通焊接板，如图 3-4 所示。

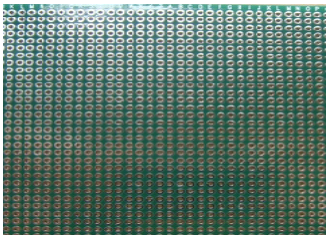

图 3-3 无导通板

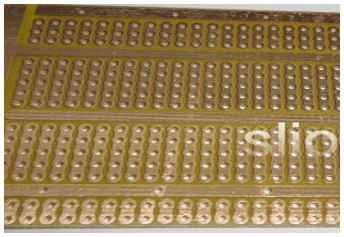

图 3-4 分行导通

4．装配盒：深 3-5cm，长 10cm，宽 8cm 左右即可。

第四节 运行环境要求

一、运行环境

倒车雷达运行环境为车内即可。

二、相关软件：

（一）MPLAB 开发环境介绍

MPLAB 集成开发环境（IDE）是综合的编辑器、项目管理器和设计平台，适用于使用 Microchip 的 PICmicro®系列单片机进行嵌入式设计的应用开发。MPLAB IDE 是适用于 PICmicro®系列单片机和 dsPICTM 数字信号控制

器，基于 Windows®操作系统的集成开发环境。

MPLAB 的功能简述：

1．使用内置编码器创建和编码源代码。

2．汇编以及编译和链接源代码。

3．通过软件内置模拟器观察程序流程调试可执行逻辑；或运用 MPLAB ICE 2000 和 MPLAB ICE 4000 仿真器、MPLAB ICD 2 在线 Debuger 实时调试可执行逻辑。

4．用模拟器或者仿真器测量时间。

5．在观察窗口中查看变量。

6．使用 MPLAB ICD 2、PICSTART® Plus 或 PRO MATE® II 器件编程器烧写固件。

7．使用 MPLAB IDE 在线帮助准确快速找出问题的答案。

MPLAB IDE 系统要求：

1．PC 兼容的奔腾（PENTIUM®）级系统。

2. 操作系统：Microsoft Windows 98 SE，Windows 2000 SP2，Windows NT&reg 及 SP6，Windows ME 或 Windows XP。

3．64 MB 内存（推荐使用 128MB）。

4．45 MB 硬盘空间。

5．Internet Explorer 5.0 或更高版本。

（二）仿真软件 Proteus 7 介绍

Proteus 系列软件不仅拥有其它 EDA 工具的仿真功能，还能对单片机及其他外围器件进行仿真。它是目前最好最全的仿真单片机及外围器件的工具。虽然目前在国内推广刚起步，但已受到单片机发烧友、从事单片机教学的老师、致力于单片机开发应用等的工作者的青睐。Proteus 是世界上著名的 EDA 工具 (仿真软件)，从原理图布图、代码调试到单片机与外围电路协同仿真，一键切换到 PCB 设计，真正实现了从设计最初到最后的完整仿真。是目前世界上唯一将电路仿真软件、PCB 设计软件和虚拟模型仿真软件三合一的设计平台。其处理器模型支持型号为 8051、HC11、PIC10/12/16/18/24/30/DsPIC33、AVR、ARM、8086 和 MSP430 等，2010 年又增加了 Cortex 和 DSP 系列处理器，并持续增加其他系列处理器模型。在编译方面，它也支持 IAR、Keil 和 MPLAB 等多种编译器。

Proteus 7 的功能简述：

1．原理布图

2．PCB 自动或者人工布线

3．SPICE 电路仿真

4．互动的电路仿真：用户甚至可以实时采用诸如 RAM，ROM，键盘，马

达，LED，LCD，AD/DA，部分 SPI 器件，部分 IIC 器件。

Proteus 7 的具体功能特性简介。

智能原理图设计（ISIS）：

1．丰富的器件应用库：超过 27000 种元器件，可方便创建新元件；

2．智能器件搜索：通过模糊搜索功能可以快速定位所需要的器件；

3．智能化连线功能：自动连线功能使连接导线简单快捷，很大缩短绘图时间；

4．支持总线结构：使用总线器件和总线布线使电路设计简明清晰；

5．可输出高质量图纸：通过个性化设置，可以生成印刷质量的 BMP 图纸，可以方便地供 WORD、POWERPOINT 等多种文档使用。

完善的电路仿真功能（Prospice）：

1．ProSPICE 混合仿真：基于工业标准 SPICE3F5，实现数字/模拟电路的混合仿真。

2．超过 27000 个仿真器件：可以通过内部原型或使用厂家的 SPICE 文件自行设计仿真器件，Labcenter 也在不断地发布新的仿真器件，还可导入第三方发布的仿真器件。

3．多样的激励源：包括直流、正弦、脉冲、分段线性脉冲、音频（使用 wav 文件）、指数信号、单频 FM、数字时钟和码流，还支持文件形式的信号输入[5]。

4．丰富的虚拟仪器：13 种虚拟仪器，面板操作逼真，如示波器、逻辑分析仪、信号发生器、直流电压/电流表、交流电压/电流表、数字图案发生器、频率器/计数器、逻辑探头、虚拟终端、SPI 调试器、I2C 调试器等。

5．生动的仿真显示：用色点显示引脚的数字电平，导线以不同颜色表示其对地电压大小，结合动态器件（如电机、显示器件、按钮）的使用可以使仿真更加直观、生动。

6．高级图形仿真功能（ASF）：基于图标的分析可以精确分析电路的多项指标，包括工作点、瞬态特性、频率特性、传输特性、噪声、失真、傅立叶频谱分析等，还可以进行一致性分析。

（三）K150 程序下载器配套软件

K150 软件套件是 K150 编程下载器的配套软件，其目的是辅助电脑连接下载器，并识别下载器上的芯片类型，把编写好的 XXX.hex 文件写入芯片中去。是一个简易，廉价，实用的下载器程序。

第四章　开发原理及详细环境搭建

第一节 蜂鸣器工作原理

一、总述

通过单片机的输出引脚输出高电平来驱动蜂鸣器发声，通过内部程序控制单片机输出高电平的持续时间与间隔，来完成报警。所以，下文将从蜂鸣器的结构原理、发声原理、驱动原理上来详细介绍其报警功能是如何实现的[6]。

二、原理介绍

（一）蜂鸣器的结构原理

1．压电式蜂鸣器：

压电式蜂鸣器的主要组成部分为压电蜂鸣片、多谐振荡器、阻抗匹配器以及、共鸣箱等组成。有些压电式蜂鸣器外壳上还装有发光二极管。多谐振荡器由晶体管或集成电路构成。当电源接通后（1.5V~15V 稳压电压），多谐振荡器起振，输出 1.5kHZ~2.5kHZ 的音频信号，来推动压电蜂鸣片发声，如图 4-1 所示。

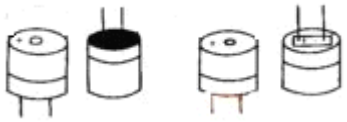

图 4-1 压电式蜂鸣器结构原理图

2．电磁式蜂鸣器：

电磁式蜂鸣器由振荡器、电磁线圈、磁铁、振动膜片及外壳等组成。接通电源后，振荡器产生的音频信号电流通过电磁线圈，使电磁线圈产生磁场。振动膜片在电磁线圈和磁铁的相互作用下，周期性地振动发声，如图 4-2 所示。

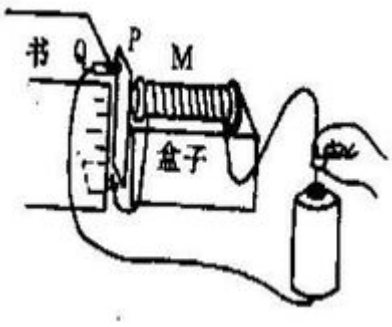

图 4-2 电磁式蜂鸣器结构原理图

（二）蜂鸣器的发声原理

两种蜂鸣器的结构虽然不一样，但是发声的原理却是相同，都是靠一个外来输入的音频振动信号，使自身的发声装置周期性的震动，从而发出不同频率的声音，即：传统意义上的声音尖锐与沙哑之分。

附：音频振动信号说明：通俗意义上来讲，音频振动信号就是通一下电，然后等一小段时间，再通一下电，反复循环。

（三）单片机驱动蜂鸣器的原理

1．PWM 输出口直接驱动：

PWM 输出口直接驱动方式是利用单片机方波输出口本身输出一定的方波来直接驱动蜂鸣器。在单片机寄存器设置中可设定 PWM 引脚输出，一般为编辑占空比和周期，通过如上操作，制定与蜂鸣器相称的方波，只要打开 PWM 输出，方波 输出口就会输出该频率的方波，此时就可以直接驱动蜂鸣器了。比如频率为 5000Hz 的蜂鸣器的驱动，可知周期为 200μs，这样只需要把方波周期设置为 200μs，占空比电平设置为 100μs，就能生成一个频率为 5000Hz 的方波，通过这个方波和三极管的放大电路功能就可以驱动蜂鸣器工作了。

2．I/O 定时翻转电平驱动：

这种 I/O 口翻转控制定时器方式较麻烦一些，需要用 Timer 或延时函数来计算时间翻转电平，产生的波形就可以用来驱动蜂鸣器了。比如为 5000Hz 的蜂鸣器的驱动，可以知道周期为 200μs，这样单片机的 I/O 口每 100μs 翻转一次电平就可以产生一个频率为 5000Hz，duty 比（占空比）为 1/2 的方波，便可以驱动这个蜂鸣器了（根据需要运用放大电路）。

（四）等效原理图

蜂鸣器工作的等效原理图如图 4-3 所示。

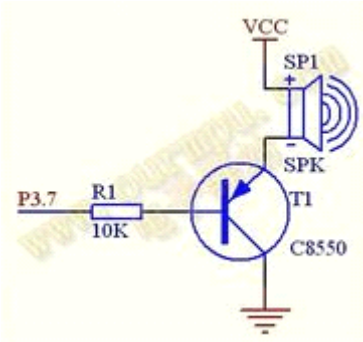

图 4-3 等效原理图

（五）蜂鸣器的种类

1．有源蜂鸣器：

简而言之，不需要音频信号也能发声，只通电就可以。

2．无源蜂鸣器：

简而言之，需要单片机给音频信号，才可以发声。

第二节　测距传感器 GP2Y0A02YK 的工作原理

一、总述

顾名思义，测距传感器的意义就是测量车体与障碍物的距离，测量以后还要把此距离以等效电压的方式传送给芯片，芯片经过相应的处理得到对应每个距离的二进制数，共其他功能调用[7]。

二、原理介绍

（一）GP2Y0A02YK 红外测距传感器原理说明

1．红外线介绍：

红外线又称红外光，它具有反射、折射、散射、干涉、吸收等性质。任何物质，只要它本身具有一定的温度（高于绝对零度），都能辐射红外线。红外线传感器测量时不与被测物体直接接触，因而不存在摩擦，并且有灵敏度高，反应快等优点。

2．GP2Y0A02YK 的构造及工作原理：

GP2Y0A02YK 的主要由红外线发射头和红外线接收头，距离等效电压输出端，正极端，负极端构成。GP2Y0A02YK 在工作时，一直重复着”发射红外线，接收反射回来的红外线”这个循环，内部的计算单元会根据返回红外信号的强度，时间等来计算出距离。而这个距离，通过内部转换，会以等效电压的表现形式，通过等效电压输出端传递给芯片。这个等效电压会在芯片中被 A/D 转换，从而得到一个一个对应着距离二进制数，以供芯片处理。

（二）结构图以及距离对应等效电压的函数图

1．外部结构图：GP2Y0A02YK 的外部结构图如图 4-4 所示。

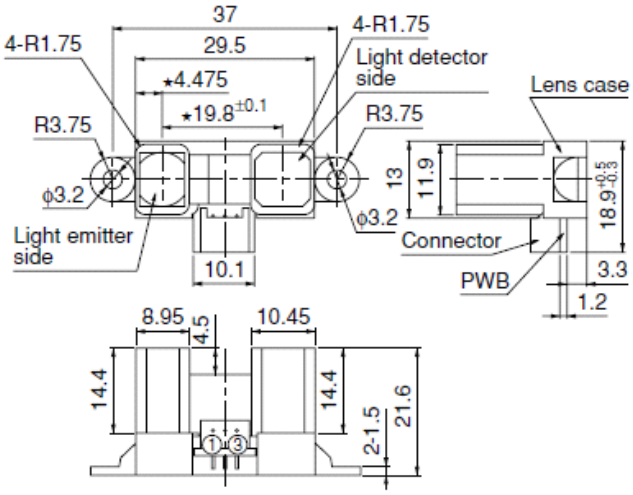

图 4-4 外部结构图

2．内部结构图：GP2Y0A02YK 的内部结构图如图 4-5 所示。

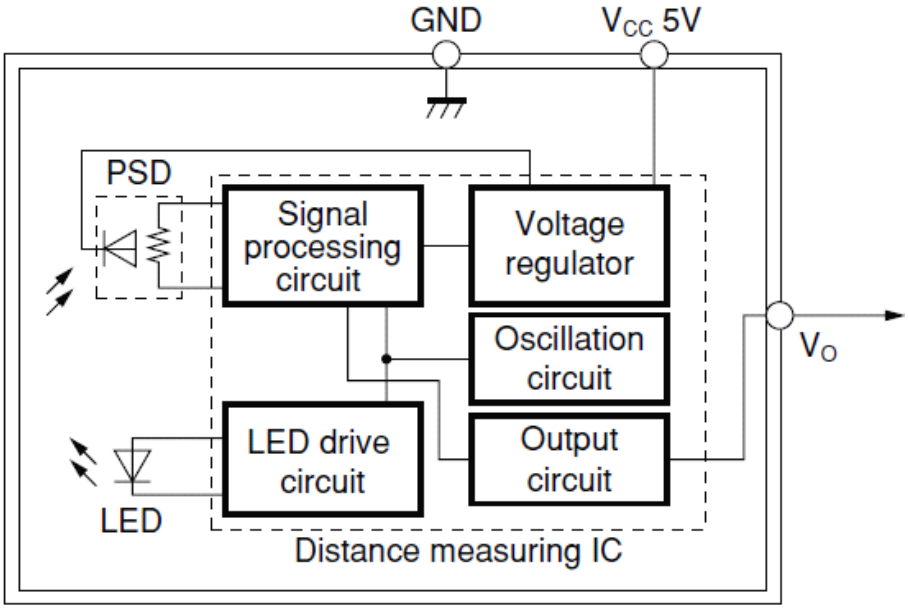

图 4-5 内部结构图

3．函数图：转换函数图如图 4-6 所示。

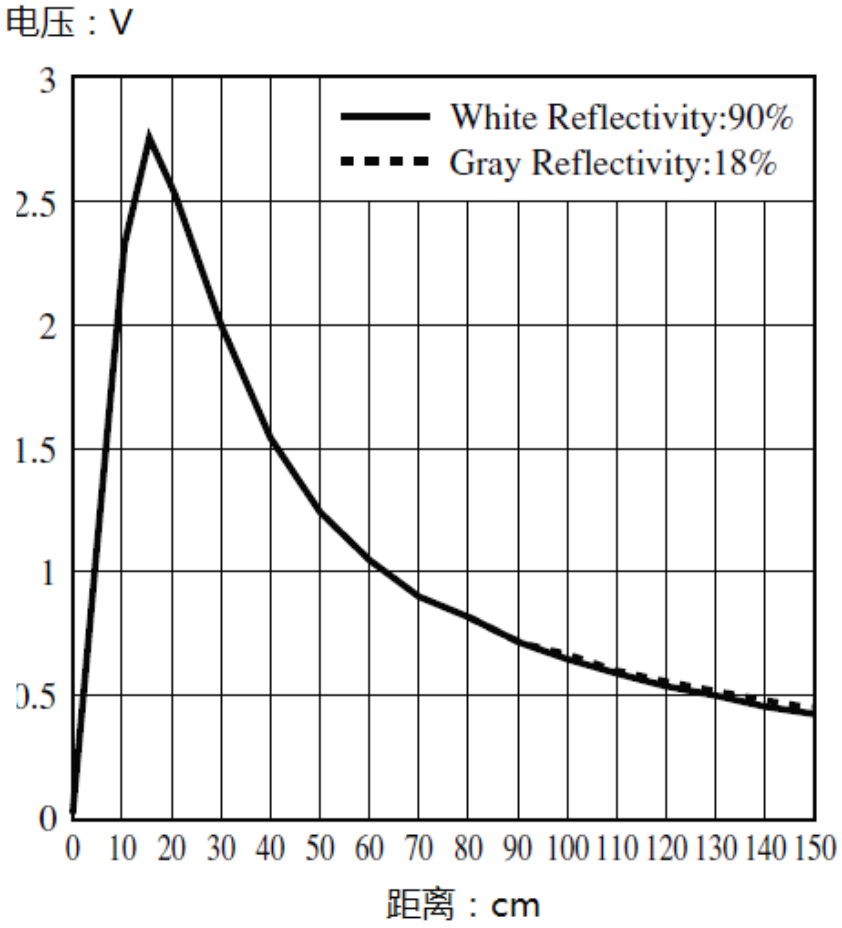

图 4-6 转换函数图

第三节 芯片 PIC12F675 的工作原理

一、PIC12F675 的概况

（一）高性能的 RISC CPU

1．仅需学习 35 条指令。

2．除了跳转指令以外所有指令都是单周期的。

3．工作速度：

（1）DC - 20 MHz 振荡器/ 时钟输入。

（2）DC - 200 ns 指令周期。

4．中断功能。

5．8 级深度硬件堆栈。

6．直接、间接和相对寻址方式。

（二）特殊单片机功能

1．可选择内部和外部振荡器：

（1）高精度内部 4 MHz 振荡器，其出厂时精度已校准为± 1%。

（2）可使用晶振和谐振器作为外部振荡器。

（3）电压典型值为 3.0V 时，将 CPU 从休眠模式唤醒需 5 μs。

2．省电休眠模式。

3．宽工作电压范围，从 2.0V 到 5.5V。

4．工业级和扩展级温度范围。

5．低功耗上电复位（POR）。

6．上电延时定时器（PWRT）和振荡器起振定时器（OST）。

7．欠压检测（BOD）。

8．带有独立振荡器的看门狗定时器，可以保证可靠的运行。

9．复用 MCLR 输入引脚。

10．引脚电平变化可触发中断。

11．独立的可编程弱上拉功能。

12．可编程代码保护。

13．高耐久性的闪存/EEPROM 存储单元：

（1）闪存耐写次数达 100，000 次。

（2）EEPROM 耐写次数达 1，000，000 次。

（3）闪存/ 数据 EEPROM 的数据保持期>40 年。

（三）低功耗功能

1．待机电流：

当电压为 2.0V 时，典型值为 1 nA。

2．工作电流：

（1）当频率为 32 kHz 时，典型值为 8.5 μA。

（2）当频率为 1 MHz 时，典型值为 100 μA。

3．看门狗定时器电流：

当电压为 2.0V 时，典型值为 300 nA。

4．Timer1 振荡器电流：

当频率为 32 kHz、电压为 2.0V 时，典型值为 4 μA。

（四）外设功能：

1．6 个具有独立方向控制功能的 I/O 引脚。

2．高灌/ 拉电流能力，可直接驱动 LED。

3．模拟比较器模块带有：

（1）一个模拟比较器。

（2）片上可编程比较器参考电压（CVREF）模块。

（3）来自器件输入引脚的可编程输入复用。

（4）可外部访问比较器输出。

4．模数转换器模块 (PIC12F675)：

（1）10 位分辨率。

（2）可编程的 4 通道输入。

（3）参考电压输入。

5．Timer0：带有 8 位可编程预分频器的 8 位定时器/计数器。

6．增强的 Timer1：

（1）带有预分频器的 16 位定时器/ 计数器。

（2）外部选通输入模式。

(3)如果已选用 INTOSC 模式的话，那么在 LP 模式中可选择 OSC1 和 OSC2 作为 Timer1 的振荡器。

7．通过两个引脚可实现在线串行编程(ICSPTM)。

二、封装信息及各引脚功能

PIC12F675 封装信息及各引脚功能如图 4-7 所示。

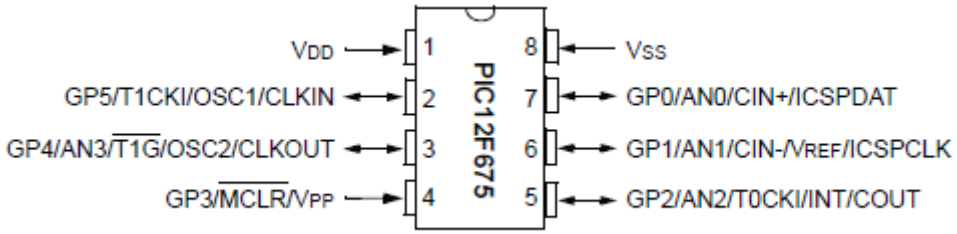

图 4-7 装配图

三、电气规范

PIC12F675 的电气规范如表 4-3 所示。

表 4-3 绝对最大额定值

偏置电压下的环境温度	-40° C 至+125° C
储存温度	-65° C 至+150° C
VDD 相对于 VSS 的电压	0.3 至+6.5V
MCLR 相对于 Vss 的电压	-0.3 至+13.5V
其他引脚相对于 VSS 的电压	-0.3V 至（VDD + 0.3V）
总功耗(1)	800 mW
VSS 引脚的最大输出电流	300 mA
VDD 引脚的最大输入电流	250 mA
输入箝位电流，IIK （VI < 0 或 VI > VDD）	± 20 mA
输出箝位电流，IOK （Vo < 0 或Vo >VDD）	± 20 mA
任一 I/O 引脚的最大输出灌电流	25 mA
任一 I/O 引脚的最大输出拉电流	25 mA
所有 GPIO 的最大灌电流	125 mA
所有 GPIO 的最大拉电流	125 mA

注 1：功耗按如下公式计算：

PDIS = VDD x {IDD - Σ IOH} + Σ {(VDD-VOH) x IOH} + Σ (VOl x IOL)。

四、功能模块介绍

总述：本次试验中将用到三个功能模块[8]。

1．GPIO端口模块：

任何单片机与外界交换信息都是通过GPIO口(即单片机的引脚)实现的，此次倒车雷达设计中，设置第7引脚为输入引脚并带有A/D转换的特殊功能，来接收传感器传来的电压信息。设置6引脚为输出引脚，用来间断输出5V电压，控制蜂

鸣器工作。

2．Timer0模块：

从产品连接电源开始，不停的记数，当记数器溢出时产生中断，从而运行中断函数，然后再从头记数，溢出后再产生下一个中断。本次倒车雷达设计中，A/D转换函数、报警延迟函数等函数被写在了中断函数中，所以，Timer0计数器产生中断后，方可运行这些函数。这样，既保证了单片机A/D转换等功能不进行得过于频繁，也为报警延迟函数提供了初步延迟服务。

3．A/D转换功能模块：

每次Timer0产生中断以后，7引脚便会实时启用A/D转换功能，此时内部A/D转换器便会通过第7引脚把从传感器传来的距离电压数据转换为单片机可看懂的10位2进制数。供其他函数使用。

接下来，将详细介绍每个功能模块的具体工作原理与相关设置。

（一）GPIO 端口

1．GPIO 端口概述：

GPIO 端口通俗意义上讲就是芯片的引脚，对它进行设置以后，就可以让各种数据在指定好的引脚上写入或写出。本次设计中，将 7 号引脚作为传感器的输入口，6 号引脚作为输出口。

可用的通用 I/O （输入/输出）引脚多达 6 个。根据使能外设的不同，部分或所有引脚不一定能用作通用 I/O 引脚。一般来说，当使能某一外设时，与它相关的引脚可能不能用作通用 I/O 引脚。

2．与 GPIO 端口相关的两个寄存器----GPIO 和 TRISIO：

GPIO 是一个 6 位宽的双向端口，它掌管着芯片上六个引脚的相关功能及传输方向。其相应数据方向寄存器为 TRISIO。将 TRISIO 中的相应位置 1 可将相应的 GPIO 引脚设置为输入引脚（即可将相应的输出驱动器设为高阻模式）。将 TRISIO 中相应位清零可将相应的 GPIO 引脚设置为输出引脚（即将输出锁存器中的数据置于所选定的引脚上）。GP3 是个例外，它只能作为输入引脚，其 TRISIO 位始终读作 1。读 GPIO 寄存器将读取引脚的状态，而写该寄存器将会写入端口锁存器。所有写操作都是读－修改－写操作。因此，写一个端口就意味着读该端口引脚，修改读取值，然后再写入端口数据锁存器。

3．引脚的其它功能：

PIC12F629/675 上的所有 GPIO 引脚都具有引脚上电平变化触发中断的功能，且所有的 GPIO 引脚（除 GP3 以外）都具备弱上拉功能。

如果按上述所说，把某一引脚的 TRISIO 寄存器设置完毕，即：确定此引脚的方向。再通过其他的寄存器设置，就可以启动这个引脚的其他功能。例：设置 GPIO3 引脚的 TRISIO 寄存器为 1，再通过设置 ADCON0 寄存器，使能 GPIO3

引脚的 A/D 转换功能的话，GPIO3 口就可以作为 A/D 转换的输入口了。

每个引脚都有很强大的功能，想启动这些功能，就需要熟悉技术手册，查看相对应的功能使能控制寄存器，找到相对应的控制寄存器后，就可以顺利开启引脚的附加功能了，寄存器设置表如图 4-8，4-9 所示。

（二）Timer0模块

1．Timer0概况：

倒车雷达中的Timer0是一个计时器，它会从被使能开始，一直按照内部时钟（参照PIC12F675概况）的上升沿或下降沿计数，当计数完毕后产生中断。产生中断的目的是进入中断服务程序，从而进行A/D转换等工作。由于此单片机执行一条指令的时间是1/4M秒，所以中断功能还可以令A/D转换等相关功能进行的不过于频繁。

GPIO-GPIO 寄存器（地址：05h）

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
bit 7							bit 0

bit 7-6: **未用位**：读作 0

bit 5-0: **GPIO<5:0>**：通用 I/O 引脚。
1 = 端口引脚电平 >VIH
0 = 端口引脚电平 <VIL

图注：			
R = 可读位	W = 可写位	U = 未用位，读作 0	
- n = POR 时的值	1= 置位	0= 清零	x= 未知位

图 4-8 GPIO 寄存器设置图

TRISO — GPIO 三态寄存器（地址：85h）

U-0	U-0	R/W-x	R/W-x	R-1	R/W-x	R/W-x	R/W-x
—	—	TRISIO5	TRISIO4	TRISIO3	TRISIO2	TRISIO1	TRISIO0
bit 7							bit 0

bit 7-6: **未用位**：读作 0

bit 5-0: **TRISIO<5:0>**：通用 I/O 三态控制位
1 = GPIO 引脚被配置为输入引脚（三态）
0 = GPIO 引脚被配置为输出引脚。
注： TRISIO<3> 始终读做 1。

图注：			
R = 可读位	W = 可写位	U = 未用位，读作 0	
- n = POR 时的值	1= 置位	0= 清零	x= 未知位

图4-9 TRISO寄存器设置图

2．Timer0产生中断：

当TMR0 寄存器定时器/ 计数器产生从FFh 至00h 的溢出时，产生Timer0中断。此溢出将T0IF 位置位。可以通过清零T0IE 位（INTCON<5>）来屏蔽该

中断。在重新允许中断之前，必须在软件中用Timer0 模块的中断服务程序将T0IF位（INTCON<2>）清零。

3．预分频器：

预分频器的目的是加大计数的间隔时间，使得产生中断等功能不会进行的过于频繁，其效果由预分频比来决定。

4．相关寄存器及功能：

如图4-10所示，通过设置OPTION_REG来设定Timer0的相关功能，在代码中，可以通过“命令T0IE=1”来使能中断，通过“命令TOIF=0”来给中断旗标位置0。

OPTION_REG—OPTION 寄存器（地址：81h）

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{GPPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

bit 7　**$\overline{\text{GPPU}}$：GPIO 上拉使能位**
1 = GPIO 上拉被禁止
0 = GPIO 上拉通过端口锁存值使能

bit 6　**INTEDG：中断边沿选择位**
1 = 在 GP2/INT 引脚的上升沿发生中断
0 = 在 GP2/INT 引脚的下降沿发生中断

bit 5　**T0CS：TMR0 时钟源选择位**
1 = GP2/T0CKI 引脚上的传输
0 = 内部指令周期时钟（CLKOUT）

bit 4　**T0SE：TMR0 时钟源边沿选择位**
1 = GP2/T0CKI 引脚的下降沿递增
0 = GP2/T0CKI 引脚的上升沿递增

bit 3　**PSA：预分频器分配位**
1 = 将预分频器分配给 WDT
0 = 将预分频器分配给 TIMER0 模块

bit 2-0　**PS2:PS0：预分频比率选择位**

位值	TMR0 比率	WDT 比率
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

图4-10 OPTION寄存器设置图

（三）A/D转换模块

1．A/D转换功能简介：

倒车雷达设计中，由传感器传进来的代表车体与障碍物距离的数据是一组连续变化的电压（模拟信号），通过A/D转换功能，将此模拟信号转变为一个一个独立的2进制数（数字信号），供单片机识别。

模数转换器（A/D）可以将模拟输入信号转换为表示该信号的一个10 位2 进制数。PIC12F675 有四个模拟输入通道，并复用到一个采样保持电路。采样保持电路的输出与模数转换器的输入相连。模数转换器通过逐次逼近比较产生二进制

数，并将结果存入10 位寄存器。可用软件选择转换所使用的参考电压为VDD 或者是VREF引脚提供的电压。

2．A/D 的配置及操作：

（1）ADCON0—A/D 控制寄存器，如图4-11所示。

（2）ANSEL— 模拟选择寄存器，如图4-12所示。

（3）ADRESH（高8位）与ADRESL（低8位）—10位转换结果寄存器，如图4-13所示。

3．具体操作与使能A/D：

根据上述数据，如果需要使能一个A/D转换功能并使之进行下去，就需要对如上2个控制A/D的寄存器进行配置，根据不同的需求，用命令ADCON0=XXXXXXXX以及ANSEL=XXXXXXXX，来设置寄存器。通过ADRESH与ADRESL来得到换算结果，并根据自己的习惯将转换的来的10位2进制数存放在两个拼起来的寄存单元中，供函数使用。

这样，A/D转换功能就会在不需要人为控制的情况下，不停的自行运作了，从而可以不断的获得传感器传来的距离信号，使单片机了解什么时候应该报警，什么时候应该停止报警，什么时候应该加快报警频率。

4．A/D的采集时间：

精确的掌握A/D转换的采集时间是精确控制单片机的关键，因为A/D采集是在每个中断发生后进行的，一秒钟内中断发生很多次，故A/D转换的次数也很多，所以必须精确了解单片机的A/D采集时间，配合单片机执行单位语句的时间，再参考机内延迟函数与发生中断所耗时间，才可以在应对不同的功能需求时，保证单片机处理的灵敏性，即时性以及速度合理性。采集时间公式如图4-14所示。

ADCON0—A/D 控制寄存器（地址：1Fh）

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	VCFG	—	—	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON
bit 7							bit 0

bit 7 **ADFM**：A/D 结果格式选择位
1 = 右对齐
0 = 左对齐

bit 6 **VCFG**：参考电压位
1 = 选用 VREF 引脚上的电压
0 = 选用 VDD

bit 5-4 **未用位**：读作 0

bit 3-2 **CHS1:CHS0**：模拟通道选择位
00 = 通道 00 （AN0）
01 = 通道 01 （AN1）
10 = 通道 02 （AN2）
11 = 通道 03 （AN3）

bit 1 **GO/$\overline{\text{DONE}}$**：A/D 转换状态位
1 = A/D 转换正在进行。将该位置 1 可启动 A/D 转换。
当 A/D 转换完成以后，该位由硬件自动清零。
0 = A/D 转换已完成 / 未进行。

bit 0 **ADON**：A/D 转换器状态位
1 = A/D 转换器模块正在运行
0 = A/D 转换器被关闭且不消耗工作电流

图注：			
R = 可读位	W = 可写位	U = 未用位，读作 0	
- n = POR 时的值	1= 置位	0= 清零	x= 未知位

图4-11 ADCON0寄存器设

ANSEL— 模拟选择寄存器（地址：9Fh）

U-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-1	R/W-1
—	ADCS2	ADCS1	ADCS0	ANS3	ANS2	ANS1	ANS0
bit 7							bit 0

bit 7 **未用位**：读作 0。

bit 6-4 **ADCS<2:0>**：A/D 转换时钟选择位
000 = FOSC/2
001 = FOSC/8
010 = FOSC/32
x11 = FRC （由专用内部振荡器产生的时钟，其频率的最大值为 500 kHz）
100 = FOSC/4
101 = FOSC/16
110 = FOSC/64

bit 3-0 **ANS3:ANS0**：模拟选择位
（可将 AN<3:0> 引脚的功能分别选为模拟或是数字）
1 = 模拟输入：分配引脚为模拟输入 (1)
0 = 数字 I/O ：分配引脚为端口引脚或特殊功能引脚

注 1： 将引脚设置为模拟输入端将自动禁止数字输入电路、弱上拉和引脚电平变化中断等功能。相应的 TRISIO 位必须设置为输入模式以允许从外部控制引脚电压。

图注：			
R = 可读位	W = 可写位	U = 未用位，读作 0	
- n = POR 时的值	1= 置位	0= 清零	x= 未知位

图4-12 ANSEL寄存器设置图

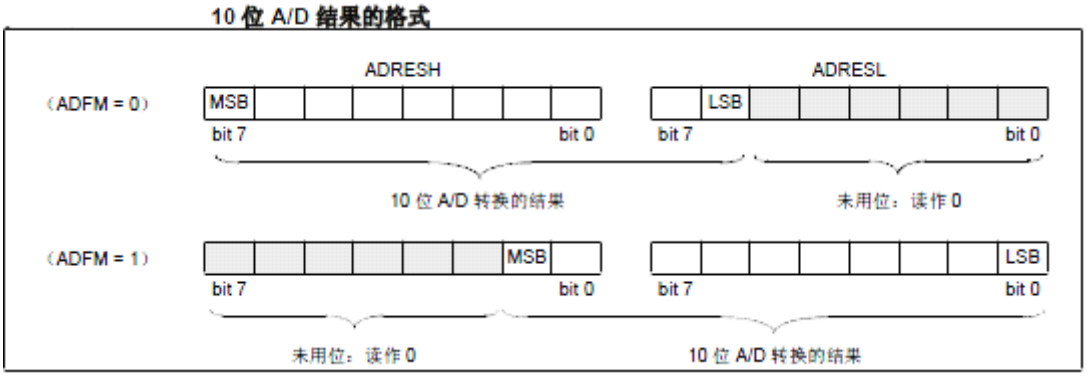

图4-13 结果寄存器设置图

$$
\begin{aligned}
T_{ACQ} &= \text{放大器稳定时间} + \\
&\quad \text{保持电容充电时间} + \\
&\quad \text{温度系数} \\
&= T_{AMP} + T_C + T_{COFF} \\
&= 2\mu s + T_C + [\ (\text{温度} -25^\circ C)\ (0.05\mu s/^\circ C)\ ] \\
T_C &= C_{HOLD}\ (R_{IC} + R_{SS} + R_S) \times (1/2047) \\
&= -120pF\ (1k\Omega + 7k\Omega + 10k\Omega) \times (0.0004885) \\
&= 16.47\mu s \\
T_{ACQ} &= 2\mu s + 16.47\mu s + [\ (50^\circ C -25^\circ C)\ (0.05\mu s/^\circ C) \\
&= 19.72\mu s
\end{aligned}
$$

图4-14 A/D采集时间说明图

第四节 倒车雷达的设计原理总述

至本节之前，详细介绍了倒车雷达所需各个器件单独的工作原理及细则，在了解了这些基础知识以后，本节开始介绍倒车雷达的各个器件被连接、运用的工作原理和设计理念。

（一）明确工作目的

工作目的是制作一个可以在 15cm 至 100cm 之间灵敏报警的倒车雷达，把这句话用单片机语言理解的话，就变为：

1．单片机需要知道现在车体与障碍物的距离——我需要一个在 15cm 至 100cm 之间灵敏工作的传感器。

2．单片机需要报警——我需要启动并控制一个蜂鸣器。

（二）实现单片机的需要

（一）中所述，明确了工作目的和单片机的需要，作为设计者，需要做的就是实现单片机的需要，让它正常的去工作，让它接管对其他器件的控制权。

单片机并不是人类，所以虽然给它一个测距传感器，但它并不能直接看明白距离的远近，所以需要把“距离”这种人类可以看懂的信号，转换成一种单片机看得明白的信号。故需要A/D转换功能。传感器把不同的距离按照特定的公式换算成与距离相对应的电压，然后从输出端传入芯片。此时，芯片之中特定的GPIO引脚被设定成A/D转换接收通道（参考第三节中内容），此引脚接收到传感器传来的不同电压信号，把这些电压信号按照特定公式转换为一个10位的二进制数，把这个10位二进制数存入通用寄存器，供芯片识别。到此为止，就完成了信号转变的任务，单片机也终于可以看懂距离的远近了。

为了让单片机在检测到不同的距离后，做出不同的反应，写程序的时候，需要在中断服务函数中写几个if函数，使单片机在参照不同距离时可以自行选择，即：根据车体与障碍物之间距离，加快或放慢报警频率。

明确了单片机如何识别距离，如何处理数据以后，还必须明确如何驱动蜂鸣器。首先驱动蜂鸣器的原理很简单，通电就行。但是，想清楚听到悦耳动听的“滴滴”的声音，必须调整输出给蜂鸣器的音频振动信号（参照第三节蜂鸣器相关介绍）。想要调整此信号，就得了解如何通过单片机输出音频振动信号。其实很简单，就是让某个GPIO引脚输出一个高电平，然后稍等一会，翻转为低电平，反复此操作即可。可以实现这个操作的代码也就随之诞生：

```
beep()
{
GPIO1=1;
    __delay_ms(100);
    GPIO1=0;
}
```

这个代码的意义是，让蜂鸣器响100ms，然后停止，听觉印象就是“滴”的一声，然后戛然而止。那么，该如何加快或减慢报警的频率呢？这还需要后续函数对蜂鸣器的发声间隔时间进行掌控。比如：

```
if (v>100)
    {
        for (;t>0;t--)
        {__delay_ms(1);}
        beep();
    }
```

这段代码的beep()为上述的发声函数，而for循环中的函数就是发声的间隔时间。

此间隔时间根据t的大小在改变，所以在车体与障碍物距离变小时，把t的值

相应改小，就可以缩短发声间隔时间，加速报警了。至此，基于单片机PIC12F675的倒车雷达的相关工作原理和设计理念就介绍完毕了。

第五节 实验环境的搭建

一、软件环境搭建

1．Windows 7系统：无需特意搭建，已自备。

2．MPLAB IDE 编程环境：

首先下载MPLAB IDE 8.80版，解压缩，点击安装后，如图4-15所示。

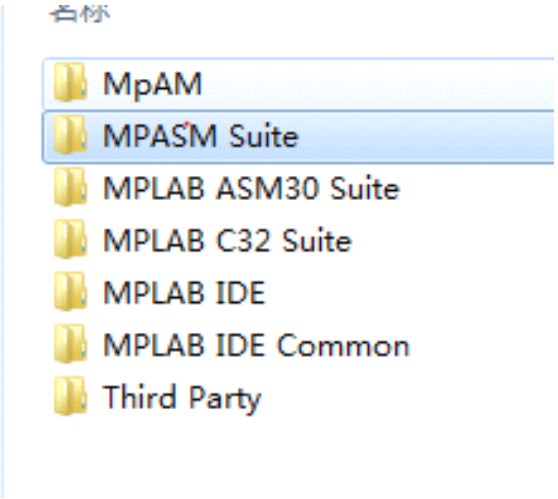

图4-15 MPLAB安装图

可以看见里面有很多PIC相关的库文件和对应芯片信息。打开MPLAB IDE中的core文件夹，便可以运行MPLAB编程环境了。

在编程之前，所以还需要安装HI-TECH，如图4-16所示。

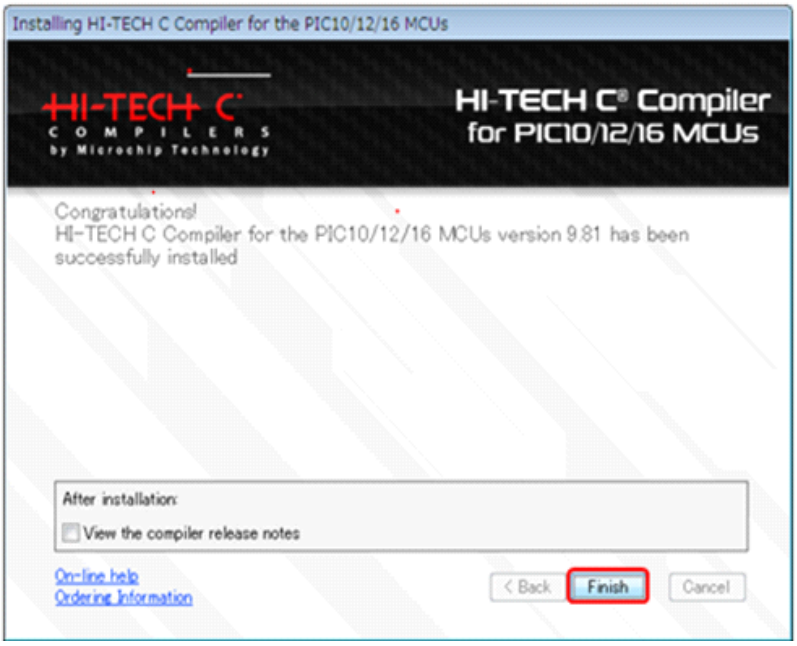

图4-16 HI-TECH安装图

HI-TECH安装完成后，进入MPLAB可以编写程序了。MPLAB具体使用方法请参照文献 。

3．Protues 7环境搭建：

（1）下载安装后就可直接使用，不需要外挂任何软件，方便快捷。

（2）进入软件以后，里面有很多选项，是不同的模拟器件，首先选出所需的芯片PIC12F675，如图4-17所示。

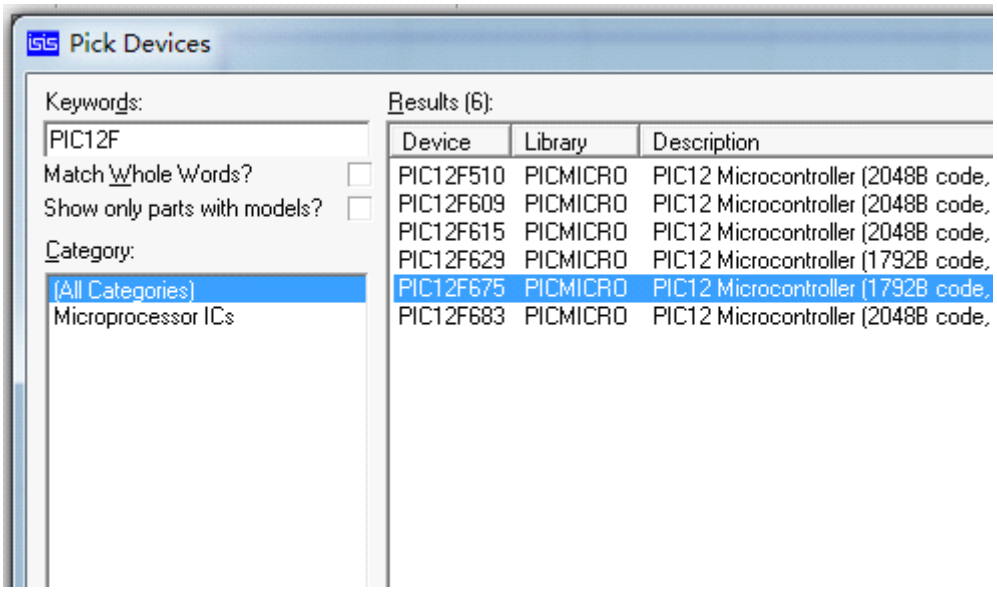

图4-17 Protues中的芯片选择

（3）调整大小，然后根据需求选择其他器件并连接。

（4）双击芯片导入.hex文件。

（5）完成准备工作，进行仿真，观察结果，如图4-18所示。

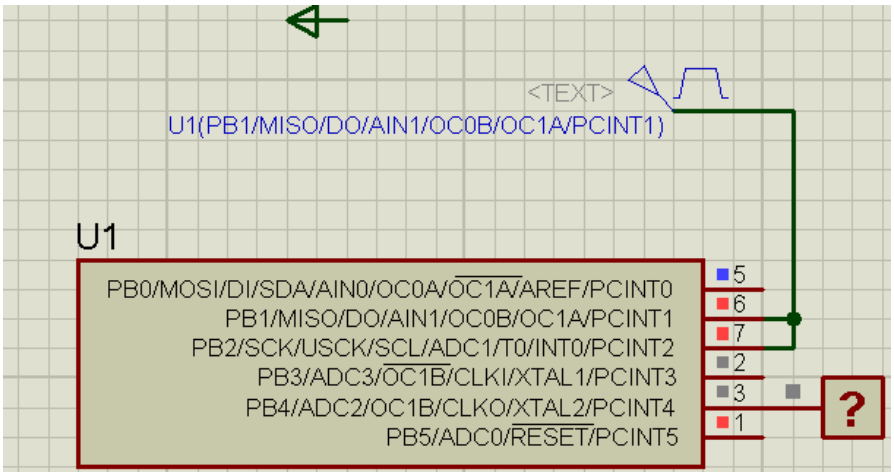

图4-18 器件连接图

4．K150下载器环境搭建：

安装软件，将下载器通过USB口连接电脑。本软件会对相应com口的下载器进行识别，如果此时已经将芯片放置在下载器上，那么软件会读出下载器以及芯片的相关信息和寄存器状态等等，此时将已经配置好的.hex文件拖拽至窗口中，再点击编程键，便可完成对芯片的写入。至此，倒车雷达的实验环境搭建完全介绍完毕。界面如图4-19所示。

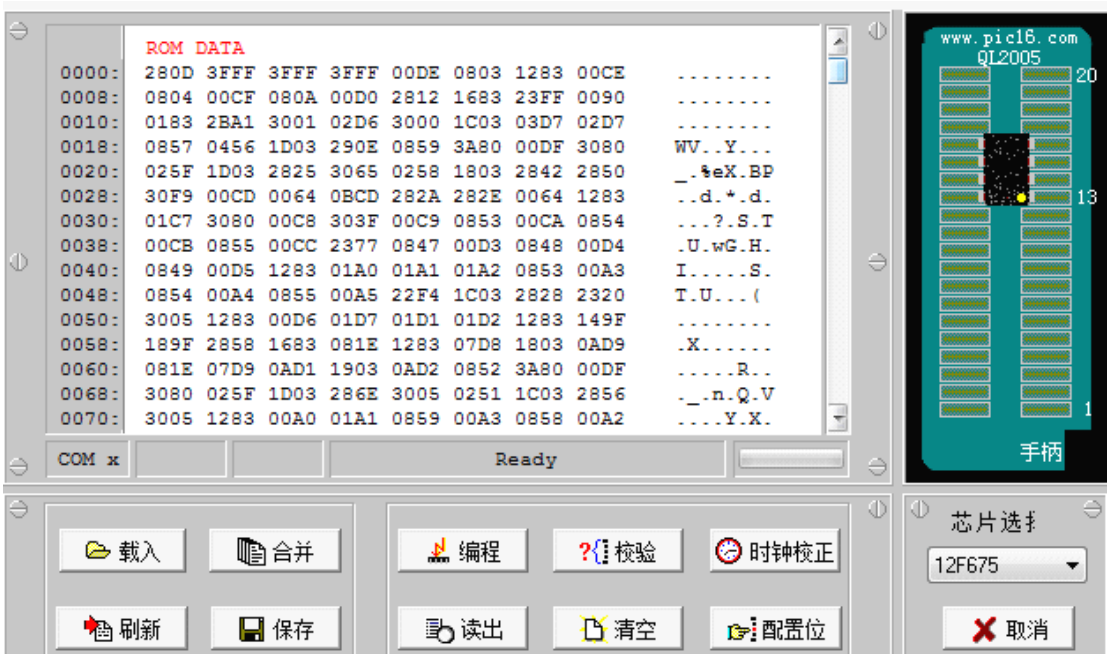

图4-19 K150工作界面

第五章 汽车倒车雷达的详细设计

第一节 倒车雷达设计准备

一、 硬件准备

1．实验器件：PIC12F675（芯片），芯片座，GP2Y0A02YK（传感器），三端稳压器 7805，实验板，导线若干，电容（0.33uf 和 0.1uf），蜂鸣器（有源）。

2．辅助器件：恒温电烙铁及电焊配套套件（包括焊锡，镊子，架台等等），高质量万用表，示波器，220V-12V 变压器，面包板。

二、软件准备

1．系统软件：Windows 7

2．支持软件：MPLAB IDE 8.80，Protues 7，K150 配套软件，HI-TECH C 环境。

三、设计目的

制造一个倒车警报，在 15cm 到 100cm 之间灵敏报警，声音适宜，检测范围大，能抵抗一般干扰。

四、构想与预测

1．用传感器测量距离，传输给芯片。

2．用芯片完成模数转换，数据处理的任务，并通过输出控制蜂鸣器。

3．在一般日光下可以正常工作。

五、实验教材

PIC12F675的DATASHEET，GP2Y0A02YK的DATASHEET，各种电子论坛上的编程案例与回帖。

第二节 倒车雷达的详细设计

一、 环境搭建

在 Windows 7 中先后安装 MPLAB IDE（附加 HI-TECH C），Proteus 7 professional，K150下载器配套软件（具体搭建方法请参照第四章）。

二、编写代码

（一）明确原理

想让一个单片机正常工作，必须明确如何配置它，以及通过什么来使能它。这就是了解并掌握一块芯片的关键。

1．配置单片机的理念：

想要对单片机下达指令，必须明确每条指令的具体对象是谁，单片机是很多小功能模块的集成体，所以编写每一条指令的时候，都需要明确把这条指令放在哪个模块中，才能起到作用。所以，必须仔细通篇阅读DATASHEET，通过了解每个模块特有的寄存器叫什么名字，如何配置，与其他功能寄存器有无联系，是否需要进行特殊初始化等，来明确编写指令的方法。概括的说，配置单片机进行工作，是通过配置很多不同的寄存器来实现的。也就是说，特定寄存器中的每一位都是单片机实现此特定功能的重要参考依据。而且，这些单独的位，在指令中也可以被单独编写或重新定义，他们通常用于控制该功能模块的开启与关闭，或作为旗标使用。

2．配置单片机的关键两步：

（1）初始化：初始化分为初始化CONFIG以及初始化各相关寄存器。初始化CONFIG是对单片机全局上的一种初始化，就和计算机中配置设定BIOS的作用有些相似，指定了硬件工作的初态和一些基本设定，用语句__CONFIG (xxxx);来配置（置顶）。相关寄存器的初始化是对每一个与功能模块相对应的寄存器的初始化，根据PIC12F675的DATASHEET中的介绍进行配置即可（参照第四章）。

（2）编写程序：程序用C语言来编写，如果想对某个功能进行控制的时候，就给相应寄存器的相应位赋值。例：如果想暂时屏蔽中断可写命令 T0IE=0；此时开始，中断功能失效。想再度使用中断功能的时候就用指令TI0E=1来把Timer0中断使能[9]。

（二）明确相关寄存器

1．全局设定：__CONFIG。用来初步配置单片机，类似于电脑中配置BIOS的作用。

2．Timer0模块：OPTION，TMR0，INTCON，TRISIO。用来配置Timer0的工作模式，来启动芯片的中断功能。

3．A/D转换模块：ADCON0，ANSEL。用来配置A/D的工作模式，来完成模数转换功能。

4．GPIO模块：TRISIO，GPIO。用来配置引脚的输入、输出及相关其他功能。

（三）编写代码

1．配置config：

全局配置，关闭WDT等没用的功能模块，开启A/D转换等有用的相关功能。代码：

```
__CONFIG (0x194);
```

CONFIG寄存器设置明细如图5-1所示。

2．初始化：

使所需的功能模块进入就绪状态，随时可以对其进行控制与修改。

代码：

TRISIO = 0b00000001;//GPIO0被配置为输入引脚，如图5-2所示。

GPIO = 0b00000000;//初始均为低电平，如图5-3所示。

ADCON0 = 0b10000001;//A/D结果右对齐，A/D模块运行，如图5-4所示。

ANSEL = 0b00010001;//A/D时钟选择Fosc/8, GPIO0引脚为模拟输入，如图5-5所示。

CONFIG—配置字（地址：2007h）

R/P-1	R/P-1	U-0	U-0	U-0	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
BG1	BG0	—	—	—	$\overline{CPD}$	$\overline{CP}$	BODEN	MCLRE	$\overline{PWRTE}$	WDTE	F0SC2	F0SC1	F0SC0
bit 13													bit 0

bit 13-12 **BG1:BG0**：BOD 和 POR 电压的带隙校准位 (1)
00 = 最低带隙电压
11 = 最高带隙电压

bit 11-9 **未用位**：读作 0

bit 8 **$\overline{CPD}$**：数据代码保护位 (2)
1 = 禁止数据存储器代码保护
0 = 使能数据存储器代码保护。

bit 7 **$\overline{CP}$**：代码保护位 (3)
1 = 禁止程序存储器代码保护
0 = 使能程序存储器代码保护

bit 6 **BODEN**：欠压检测使能位 (4)
1 = 使能 BOD
0 = 禁止 BOD

bit 5 **MCLRE**：GP3/$\overline{MCLR}$ 引脚功能选择 (5)
1 = GP3/$\overline{MCLR}$ 引脚功能为 $\overline{MCLR}$
0 = GP3/$\overline{MCLR}$ 引脚功能为数字 I/O，$\overline{MCLR}$ 内部连接到 VDD

bit 4 **$\overline{PWRTE}$**：上电延时定时器使能位
1 = 禁止 PWRT
0 = 使能 PWRT

bit 3 **WDTE**：看门狗定时器使能位
1 = 使能 WDT
0 = 禁止 WDT

bit 2-0 **FOSC2:FOSC0**：振荡器选择位
111 = RC 振荡器：GP4/OSC2/CLKOUT 引脚为 CLKOUT 功能，GP5/OSC1/CLKIN 引脚上连接 RC
110 = RC 振荡器：GP4/OSC2/CLKOUT 引脚为 I/O 功能，GP5/OSC1/CLKIN 引脚连接 RC

图5-1 CONFIG设置字说明图

GPIO-GPIO 寄存器 (地址：05h)

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
bit 7							bit 0

bit 7-6: **未用位**：读作 0

bit 5-0: **GPIO<5:0>**：通用 I/O 引脚。
1 = 端口引脚电平 >VIH
0 = 端口引脚电平 <VIL

图注：			
R = 可读位	W = 可写位	U = 未用位，读作 0	
- n = POR 时的值	1= 置位	0= 清零	x= 未知位

图5-2 GPIO寄存器设置字说明图

OPTION= 0x84;//GPIO上拉禁止，INT引脚的下降沿触发，Timer0产生的中断在下降沿触发，选用内部时钟，预分频器给Timer0用，预分频比为1/8，计数器模式，明细如图5-6所示。

T0IE = 1; //Timer0中断使能。

GIE = 1; //全局中断使能。

TRISO — GPIO 三态寄存器 （地址：85h）

U-0	U-0	R/W-x	R/W-x	R-1	R/W-x	R/W-x	R/W-x
—	—	TRISIO5	TRISIO4	TRISIO3	TRISIO2	TRISIO1	TRISIO0
bit 7							bit 0

bit 7-6: **未用位**：读作 0

bit 5-0: **TRISIO<5:0>**：通用 I/O 三态控制位
1 = GPIO 引脚被配置为输入引脚 （三态）
0 = GPIO 引脚被配置为输出引脚。
注： TRISIO<3> 始终读做 1。

图注：			
R = 可读位	W = 可写位	U = 未用位，读作 0	
- n = POR 时的值	1= 置位	0= 清零	x= 未知位

图5-3 TRISO寄存器设置字说明图

ADCON0—A/D 控制寄存器 （地址：1Fh）

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	VCFG	—	—	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON
bit 7							bit 0

bit 7 **ADFM**：A/D 结果格式选择位
1 = 右对齐
0 = 左对齐

bit 6 **VCFG**：参考电压位
1 = 选用 VREF 引脚上的电压
0 = 选用 VDD

bit 5-4 **未用位**：读作 0

bit 3-2 **CHS1:CHS0**：模拟通道选择位
00 = 通道 00 （AN0）
01 = 通道 01 （AN1）
10 = 通道 02 （AN2）
11 = 通道 03 （AN3）

bit 1 **GO/$\overline{\text{DONE}}$**：A/D 转换状态位
1 = A/D 转换正在进行。将该位置 1 可启动 A/D 转换。
当 A/D 转换完成以后，该位由硬件自动清零。
0 = A/D 转换已完成 / 未进行。

bit 0 **ADON**：A/D 转换器状态位
1 = A/D 转换器模块正在运行
0 = A/D 转换器被关闭且不消耗工作电流

图注：			
R = 可读位	W = 可写位	U = 未用位，读作 0	
- n = POR 时的值	1= 置位	0= 清零	x= 未知位

图5-4 ADCON0寄存器设置字说明图

ANSEL— 模拟选择寄存器 （地址：9Fh）

U-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-1	R/W-1
—	ADCS2	ADCS1	ADCS0	ANS3	ANS2	ANS1	ANS0
bit 7							bit 0

bit 7 **未用位**：读作 0。

bit 6-4 **ADCS<2:0>**：A/D 转换时钟选择位
000 = FOSC/2
001 = FOSC/8
010 = FOSC/32
x11 = FRC （由专用内部振荡器产生的时钟，其频率的最大值为 500 kHz）
100 = FOSC/4
101 = FOSC/16
110 = FOSC/64

bit 3-0 **ANS3:ANS0**：模拟选择位
（可将 AN<3:0> 引脚的功能分别选为模拟或是数字）
1 = 模拟输入：分配引脚为模拟输入 (1)
0 = 数字 I/O ：分配引脚为端口引脚或特殊功能引脚

注 1： 将引脚设置为模拟输入端将自动禁止数字输入电路、弱上拉和引脚电平变化中断等功能。相应的 TRISIO 位必须设置为输入模式以允许从外部控制引脚电压。

图注：			
R = 可读位	W = 可写位	U = 未用位，读作 0	
- n = POR 时的值	1= 置位	0= 清零	x= 未知位

图5-5 ANSEL寄存器设置字说明图

3．蜂鸣器发声函数

根据蜂鸣器发声原理（参考第四章），让6号引脚（GPIO1）为高电平，令蜂鸣器发声并持续100ms。

发生函数代码：

```
void beep()
{

T0IE=0;                    //暂停Timer0中断
GPIO1=1;                   //令GPIO1引脚输出高电平（5V）
__delay_ms(100);           //延迟100ms
GPIO1=0;                   //令GPIO引脚输出低电平
T0IE=1;                    //使能Timer0中断

}
```

4．中断函数

此中断函数中主要包括：A/D转换函数，报警延迟函数，距离状况分段函数 。

（1）A/D转换函数

通过将GODONE位置1，启动A/D转换，将5次转换的结果相加取平均(用来消除噪音干扰)写入“v”中，供下面的if函数使用。

（2）报警延迟函数

让for循环，运行t次，一次可粗略估计为1ms，所以可以通过改变t的数值来控制此for循环的运行次数，来满足不同的实际情况下的时间延迟需求。从而控制报警间隔的长短。

（3）距离状况分段函数

根据电压距离转换函数图（参考第四章），距离越近递增函数的斜率越大，即：变化的速度随着距离的缩短而加大，所以检测距离段也随之减小，从而增加检测精度。

v<30，距离在150cm以外。

30<=v<40，距离在100cm-150cm之间。

40<=v<=200，距离在60cm-100cm之间。

200<v<=300，距离在30cm-60cm之间。

v>300，距离在15cm-30cm之间。

此分段函数的说明如流程图5-7所示。

OPTION_REG—OPTION 寄存器（地址：81h）

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{GPPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

bit 7　$\overline{\text{GPPU}}$：GPIO 上拉使能位
1 = GPIO 上拉被禁止
0 = GPIO 上拉通过端口锁存值使能

bit 6　INTEDG：中断边沿选择位
1 = 在 GP2/INT 引脚的上升沿发生中断
0 = 在 GP2/INT 引脚的下降沿发生中断

bit 5　T0CS：TMR0 时钟源选择位
1 = GP2/T0CKI 引脚上的传输
0 = 内部指令周期时钟（CLKOUT）

bit 4　T0SE：TMR0 时钟源边沿选择位
1 = GP2/T0CKI 引脚的下降沿递增
0 = GP2/T0CKI 引脚的上升沿递增

bit 3　PSA：预分频器分配位
1 = 将预分频器分配给 WDT
0 = 将预分频器分配给 TIMER0 模块

bit 2-0　PS2:PS0：预分频比率选择位

位值	TMR0 比率	WDT 比率
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

图5-6 OPTION设置字说明图

代码：

```
static void interrupt in (void)
{
    Cnt3mS--;                          //保证启动时间。
     if (Cnt3mS == 0)
     {
        if (v>100)
         {
            for (;t>0;t--)             //根据t的大小，决定“滴”声的间隔时间
            {__delay_ms(1);}
            beep();
         }
                          //消噪处理，5次A/D转换的结果取平均值
         Cnt3mS = 5;
         for(int g=0;g<5;g++)
        {
```

```
        GODONE = 1;
          while(GODONE){}
          v+=(ADRESH<<8)+ADRESL;
      }
v/=5;
if(v<=30)t=0;
if (v<=40) t=-5*v+200;
if (v>40&&v<=200) t=-1.313*v+1053;
if (v>200&&v<=300) t=-0.4*v+870;
if (v>300) t=-3*v+1668;
  }
  T0IF = 0;                    //中断旗标位置0
}
```

5．主函数：

代码：

```
void main()
{
                              //相关初始化，上文已介绍过.
TRISIO = 0b00000001;
GPIO = 0b00000000;
ADCON0 = 0b10000001;
ANSEL = 0b00010001;
CMCON = 0b00000000;
OPTION= 0x84;
TMR0 = 0;
T0IE = 1;
GIE = 1;

while(1)
  { }                          //主函数主体
}
```

图5-7 if函数流程图

6．完整代码：

```
#include <pic.h>
#include <pic12rf675.h>
#include <htc.h>
#define _XTAL_FREQ 4000000
static unsigned int Cnt3mS=5;
static    float t=100;
static int v=10;
CONFIG部分
初始化部分
中断函数部分
蜂鸣器发声部分
主函数部分
```

简述：通电后Timer0马上开始计时，当溢出后产生中断，进入中断服务函数，进行A/D转换等工作，根据if分段函数来控制t的大小，从而控制蜂鸣器。

7．点击build 编译并建立.hex文件。

三、进行仿真

1．在Protues中选择好芯片PIC12F675，如图5-7所示。

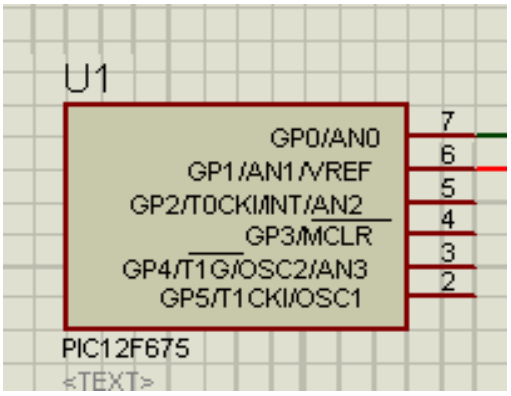

图5-7 仿真流程图1

2．找到DC电压输出部件，如图5-8所示。

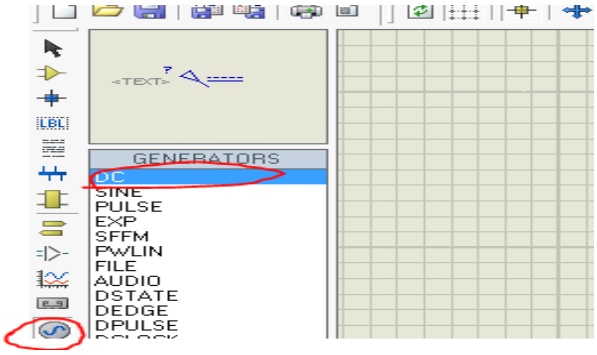

图5-8 仿真流程图2

3．找到输出监测部件，如图5-9所示。

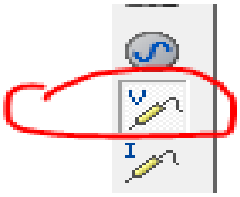

图5-9 仿真流程图3

4．连接好各部件，如图5-10所示。

5．导入hex文件，开始仿真。Vol给3V以下不同的电压（模拟感器输入距离等效电压），6引脚接出来的监测箭头就会以不一样的速度不断闪烁（模拟蜂鸣器）。

四、烧写芯片

经过仿真，确定代码没有问题以后，可以开始烧写工作。放置好芯片，下载器连接电脑，打开 K150。

1．点击载入按钮，载入刚才的.hex 文件。

2．再点击编程，完成下载，如图 5-11 所示。

3．到此为止，单片机 PIC12F675 已经具有自我工作的能力了。

五、面包板测试

1．按照电路图 5-12 所示，把器件连接在面包板上，进行初步实测。

2．将面包板移至室外，测试在正常阳光强度下是否正常工作。

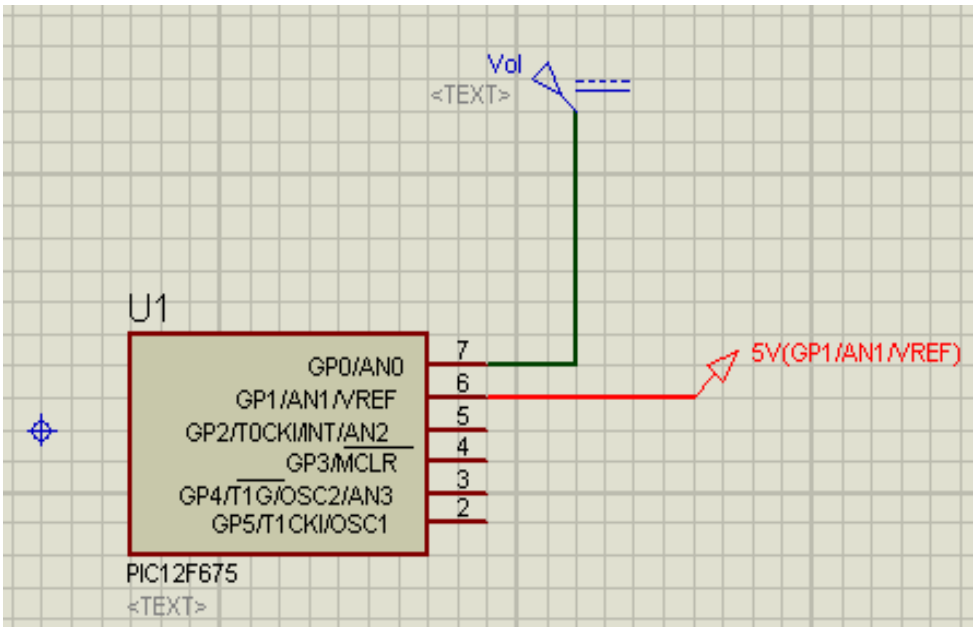

图5-10 仿真流程图4

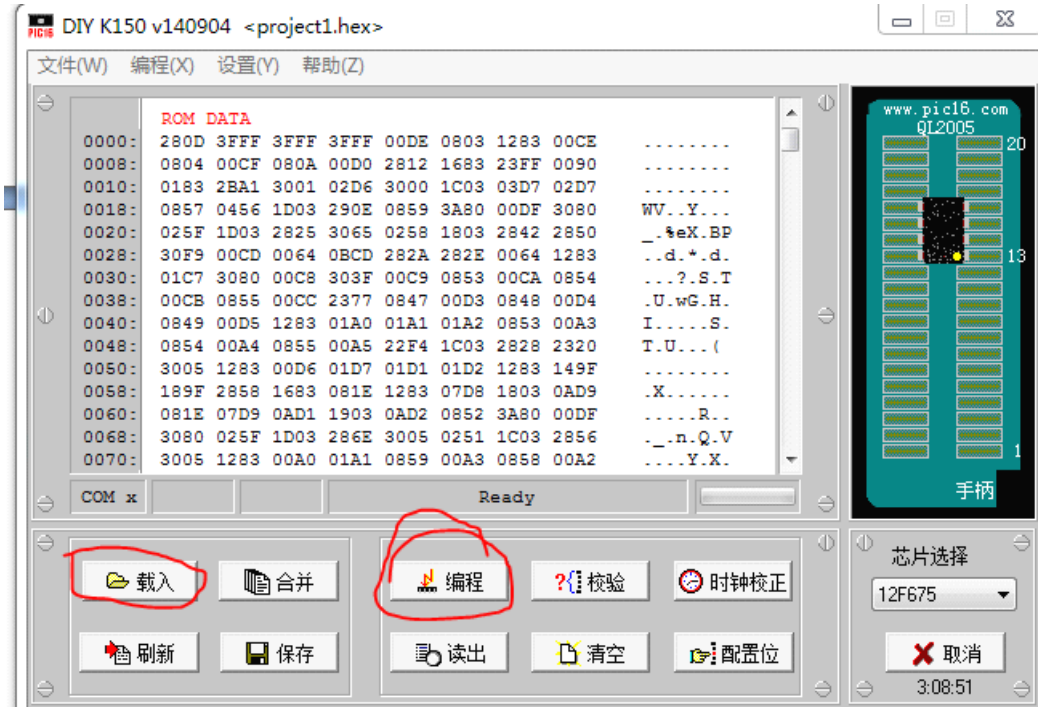

图 5-11 K150 下载界面

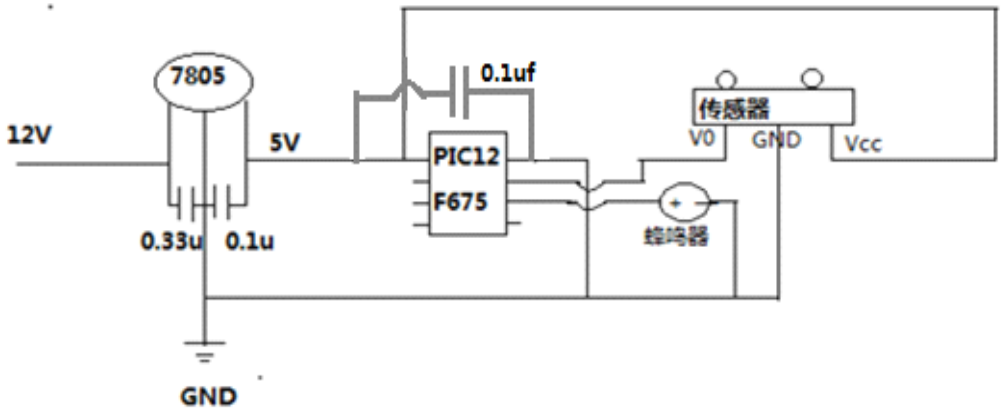

图 5-12 电路图设计

第三节 成品制作

一、焊接

1．准备好焊接台及其相关用具。

2．恒温烙铁调制 430℃-450℃。

3．合理配置电路板，优先焊接 PIC12F675 和三端稳压器。

4．焊接电容。

5．焊接蜂鸣器。

6．焊接导线。

二、装盒

1．绷带加固，如图 5-13 所示。

2．支脚组装。

3．外盒钻孔，切割处理，如图 5-14 所示。

4．封装进盒，如图 5-15，5-16 所示。

三、用具整理

1．将烙铁断电并搁置，15 分钟左右以后，用手轻触烙铁尖端，若以还原至常温，方可收纳回工具箱。

2．将导线，电容，实验板，富裕的芯片座等等器件分别放回工具箱。并将实验室所用器具归位。

3．将万用表装盒，将剥线钳，镊子等工具放回工具箱。

4．清扫桌面线头及焊接垃圾。

四、成品展示

倒车雷达成品如图 5-17 所示。

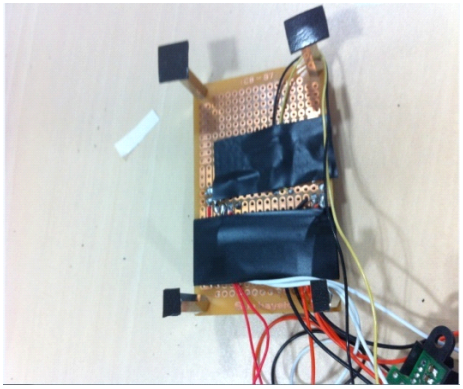

图 5-13 绷带加固

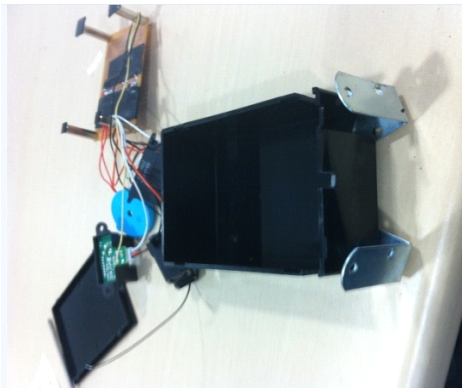

图 5-14 钻孔组装

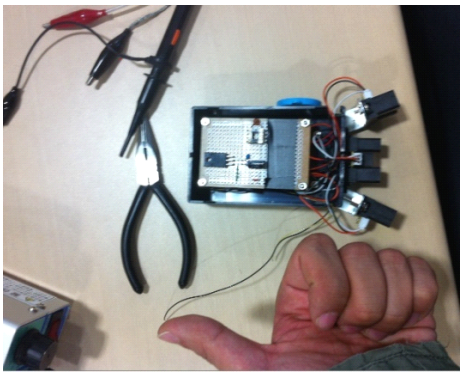

图 5-15 装盒 1

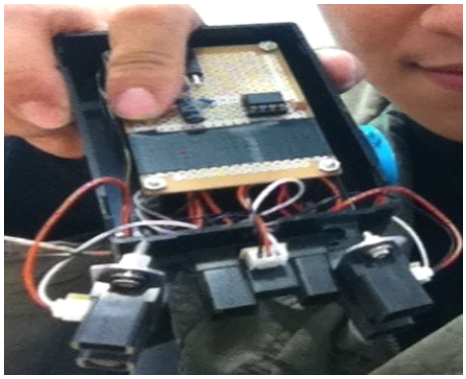

图 5-16 装盒 2

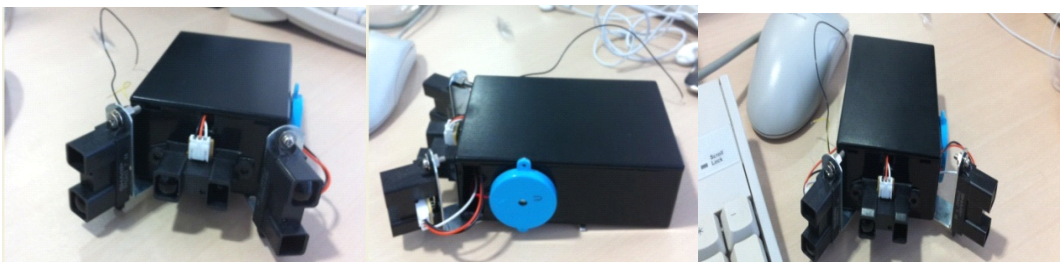

图 5-17 展示图

第六章 功能检验及相关数据测定

一、与实验预测对照，调查是否满足所需功能性

（一）物理方面需求对照

小而便利，侦测范围广，至少要在 120° 以上，如图 6-1 所示。

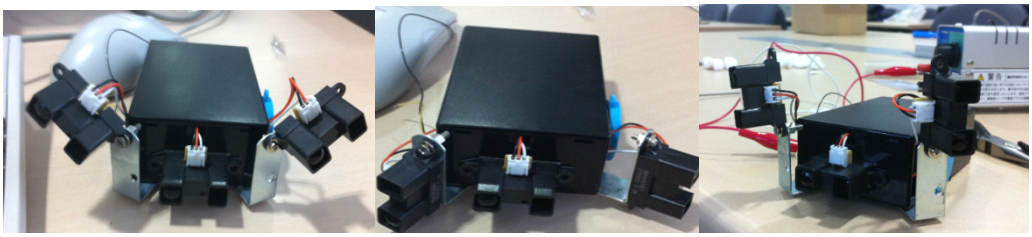

图 6-1 实物成品展示图

实测：铁片和探头可独立进行转动，可以满足一切探测范围，探测角度需求 。

小盒大小：长约 10cm，宽约 6cm，高约 4cm。

结果：满足实验预测与设计需求。

（二）非物理功能方面对照

在 15cm-100cm 之间灵敏报警，声音适宜。检验如表 6-2 所示。

表 6-2 实测报警数据表格

距传感器距离	平均报警频率
130cm-100cm	每 1.70 秒左右响一次
100cm-80cm	每 1.40 秒左右响一次
80cm-70cm	每 0.80 秒左右响一次
70cm-60cm	每 0.54 秒左右响一次
60cm-50cm	每 0.37 秒左右响一次
50cm-30cm	每 0.26 秒左右响一次
30cm-20cm	每 0.20 秒左右响一次
20cm-15cm	每 0.20 秒左右响一次
15cm 以下	无反应

实测：可以在 15cm-100cm 之间有效灵敏测距报警，声音听起来很舒适。

结果：满足实验预测与设计需求。

总　结

文从可行性分析、需求分析两方面对基于 PIC12F675 的倒车雷达进行了初步设计与预测。为了达到价格低，体积小，反应灵敏，侦测范围广等要求，本倒车雷达采用了 GP2Y0A02YK（sharp）高敏感度测距传感器进行测距，采用了 PIC12F675 芯片进行数据处理与总体制御，采用了小型蜂鸣器进行报警。

倒车雷达设计中，首先通过配置寄存器 GPIO 和寄存器 TRISO 来对 GPIO 端口进行设定，将 7 引脚定为输入引脚并使能 A/D 转换功能，将其他引脚定为输出引脚。然后通过配置寄存器 OPTION 和对位 T0IF 和位 T0IE 的操作来控制 Timer0 工作。同时对寄存器 ADCON0 和 ANSEL 进行配置并对位 GODONE 进行适当操作，来控制 A/D 转换模块工作。由传感器传来的代表距离的模拟信号通过 7 引脚进入芯片，经 A/D 转换后作为参考数据被芯片中的 if 分段函数使用，又通过延时函数等函数的辅助，来控制芯片 6 引脚输出高电平的频率。从而启动蜂鸣器报警并控制蜂鸣器的报警频率。最后，将倒车雷达各部件焊接在小型实验板上，组装进入小盒中，并采用旋转式底座来固定传感器。经过最终测试，检验倒车雷达在正常日光下的工作状况，确定无异常状况发生，便可完成实验。软件环境方面，在本倒车雷达设计中，通过 windows 7 系统下的 MPLAB、HI-TECH C 软件环境编写程序，通过 Proteus 7 进行初步仿真，通过 K150 套件进行文件传输。通过如上方法，成功的实现了性价比高、体积小、侦测范围广、灵敏度高、设计独具个性的倒车雷达系统。

过本次毕业设计，我得到了很大收获。我深刻了解到，从事嵌入式方面的工作，哪怕是很简单的实验，也需要有很广很扎实的基础知识，因为我将面临很多实际问题，而想解决这些实际问题，软件方面和硬件方面都须要储备足够的知识，而且，这些知识绝不仅仅是书本上学到的理论知识。

参考文献：

[1]Mark Rumreich．Car Stereo CookBook[M]．Mcgraw-Hill，2005：145-189．

[2]严雨，廉洁. AVR 单片机 C 语言应用 100 例[M]. 北京：电子工业出版社，2012：87-233.

[3]孟祥莲．嵌入式系统原理及应用教程[M]．北京：清华大学出版社，2007：334-366.

[4]康华光．电子技术基础.模拟部分（第五版）[M]．北京：高等教育出版社，2009：50-53.

[5]高橋久，見城尚志．図解一分かる電子回路[M]．講談社，2010：67-238.

[6]周润景，张丽娜，丁莉. 基于 PROTEUS 的电路及单片机设计与仿真(第 2 版) [M]. 北京：北京航天航空大学出版社，2010:300-405.

[7]Sharp．GP2Y0A02YK-DATASHEET[N]．Sharp Electrionic Components，2004，3，2(15).

[8]Mirochip．PIC12F675-DATASHEET[N]．MICROCHIP，2004，1-9.

[9]克尼汉．C 程序设计语言[M]．北京：机械工业出版社，2006：179-223.

[10]彭伟．单片机 C 语言程序设计实训 100 例——基于 PIC+Proteus 仿真[M]．北京：电子工业出版社，2011：300-547.

[11]沈建良. ATmega128 单片机入门与提高[M]. 北京：北京航空航天大学出版社，2009: 302-400.

[12]Ivor Horton．Ivor Horton’s Beginning Visual C++ 2010 [M]．Wrox，2010：579-720.

[13]夏永祥，史志才. C/C++常见编程问题及解答[M]. 北京：清华大学出版社，2012: 70-102.

[14]Richard Hammond．Car Science[M]．Dorling Kindersley，2011：108-299.

[15]John Turner, Ph.D．Automotive Sensors (Sensor Technology)[M]．Momentum Press，2009: 154-362.

致　谢

倒车雷达的最初构思到通过 PIC12F675 实现，我的指导教师宁爱军老师给与了我非常大的帮助。宁老师认真负责的态度以及对知识、对项目设计的洞察力帮助我克服了在开发过程中遇到的重重困难。在宁老师的指导下，我不但很快的学习到了新的知识，而且对软硬件开发有了新的理解。

谢天津科技大学的所有老师们，让我在这大学四年中学到了今后赖以生存的专业知识。

谢所有帮助过我的同学们。你们让我的大学生活更加丰富多彩。

谢我的父母对我无私的支持。

想我的大学生活，四年中的一切仿佛如昨日一样清晰可见，感谢老师和同学们给我的人生中留下精彩一页。

后，再一次感谢所有关心和帮助过我的人们，谢谢！