

摘 要

随着经济的快速发展,传统能源储量的不足问题越来越突出。人们逐渐认识到应该寻求一条可持续发展之路,因此寻找可再生能源与替代能源成为经济发展的必由之路。而太阳能是一种取之不尽用之不竭的清洁可再生能源,是目前世界上主要能源探明储量的一万倍。因而太阳能必然成为未来最理想的绿色能源。

我国幅员辽阔,全年太阳照射能量巨大,但是太阳能的利用效率不够充分,这主要是由于太阳光的间歇性和空间分布不断变化,传统的太阳能设备无法对太阳进行实时跟踪。所以开发一套随太阳的角度不断变化的智能设备显得尤为必要。

嵌入式技术以其体积小、成本低、速度快、集成度高、功耗低和可靠性强等特点越来越受到人们的普遍欢迎。本文正是基于嵌入式技术,提出了一种新的智能追光原理。此装置能大大提高太阳光的利用效率。不仅具有巨大的环保价值而且具有广阔的市场前景。

本文是以 ARM9为中央处理器,以 WinCE 为操作系统。在硬件设计方面,本文详细分析了智能追踪系统的组成和工作原理,完成了摄像头传感器模块、电机控制模块和调试接口模块的设计。软件方面给出了 WinCE 系统的定制和移植、驱动程序的开发等内容。

通过以上工作,该系统能够精确的计算任意时刻和地点太阳的高度角和方向角,使太阳能电池板以及其他太阳能利用设备时刻处于太阳光直射下,增加太阳光的利用率达到30%以上。能够充分的满足便携式太阳光智能追踪设备的需求。

关键词: ARM9, 智能追踪, WinCE 操作系统, 嵌入式

ABSTRACT

With the rapid development of economic, the problem of the shortage of conventional energy reserves is becoming increasingly prominent. People gradually come to realize that a sustainable development way is supposed to be found. So looking for a renewable energy or alternative energy becoming the only way of the development of economy. As the solar energy is an inexhaustible supply of clean and renewable energy, and it's reserves is 10,000 times as much as the world's proven reserves of major energy. Therefore, the solar energy will inevitably become the best green energy resources.

Our vast country has tremendous energy of solar radiation throughout the year. However, its utilization is inadequate. This is mainly due to the intermittent nature of sunlight and the everchanging spatial distribution. Traditional solar equipment can not tracking the sun in real-time. So it is particularly essential to invent an intelligent equipment that can continuous changing the angle with the sun.

Embedded Technology is becoming more and more widely welcomed with the feature of small size, low cost, high speed, high integration, low power consumption and reliability, etc. This article is just based on embedded technology, we proposed a new theory of sunlight intelligent tracking. This device can greatly improve the utilization efficiency of solar energy. Not only has the huge value in environmental protection but also broad market prospect.

This article is based on ARM9 with Windows CE operating system. In the aspect of hardware design, this paper makes a detailed analysis of the composition and working principle of this intelligent system. Completely finished the design of camera sensor module, motor control module and debug interface module. The customization and porting of WinCE system and driver development was given in software aspect.

Through the above work, the system can accurately calculate the altitude angle and azimuth angle of the sun in any time and any place. Making the solar panels and other utilizations of solar energy equipment always in direct sunlight, which the efficiency of sunlight utilization will reach up to 30% or more. Greatly satisfying the needs of the portable intelligent sunlight tracking equipment.

Key words: ARM9 , Intelligent Tracking , WinCE OS , Embedded Technology



学位论文的主要创新点

- (一) 本为着眼于绿色能源的开发和利用,能大大提高太阳能资源的利用率。具有广阔的市场前景和经济价值。
- (二) 本文采用图像处理技术,提出了一种新的计算太阳高度角和方向角的原理。
- (三) 基于嵌入式技术,以 S3C2440 为开发板,WinCE 为操作系统,控制更加精确。
- (四) 双轴跟踪,能提高太阳能利用率达到 35%。并且没有累计误差,不受时间和地域的限制。设计了一套完整的智能追光系统。

第一章 绪论

1.1 课题的研究背景和意义

1.1.1 人类面临的能源现状

现在人类正在处于一个高度发达的工业社会，高速发展的工业为我们提供了越来越多的舒适和便利。但是，由于世界资源的过渡采伐，世界能源不足问题越来越突出。根据经济部能源委员会资料显示，在技术与成本的限制下，预计世界石油蕴藏量只可再开采四十年，天然气可开采六十二年，煤炭可开采二百二十七年，而核能发电的燃料源自铀矿，预计尚可开采七十七年，考虑到核燃料回收再处理后可以重复运用，其使用年数可增加五十倍，也就是可以达到约三千八百年^[1]。

正在世界能源蕴藏量不断减少的同时，随着现代工业、农业、国防和科技事业的急速发展，各国对能源的消耗量却在显著的增加，年增长率达到 5%-6%，就是说十几年就要翻一番。人类进入二十世纪以来，世界能源消耗量在成倍的增加。进入二十一世纪之后，能源供应更加紧张，再加上世界的能源分布不平衡，必然导致能源危机的不断发生^[2-3]。

更加应该引起我们注意的是这些化石燃料燃烧产生大量的二氧化碳和含氧硫化物，直接导致地球气温升高和酸雨的产生，生态环境正在急剧恶化，这些问题严重影响着人类的生存和发展。所以越来越多的人开始注意到，大力开发和利用可再生能源、走可持续发展之路已经迫在眉睫。

1.1.2 太阳能资源的优势

太阳能是一种取之不尽用之不竭的清洁可再生能源。据记载，人类利用太阳能已有 3000 多年的历史^[4]，但是长期以来一直发展缓慢。随着二十世纪能源危机的出现和环境污染的加剧，太阳能资源以其巨大的优势迎来了太阳能利用的新高潮。

(1) 普遍性：太阳能资源处处都可以利用，没有地域的限制，无论陆地或海洋，无论高山或岛屿，处处皆有，可直接开发和利用，且无须开采和运输。这对缓解运输压力，解决偏远山区或海上以及岛屿的能源供应有着独特的优越性。

(2) 储量巨大：每年到达地球表面上的太阳辐射能为 17.8 亿千瓦，约相当于 130 亿吨煤，是全世界目前发电量的 8 万倍。在当今世界可开发能源中，其总量最大。

(3) 无污染性：太阳能资源分为太阳能光伏和太阳热能。光伏板组件是一种暴露在阳光下便会产生直流电的发电装置，由几乎全部以半导体材料制成的薄层固体光伏电池组成。而太阳热能科技将阳光聚合，并运用其能量产生热水、蒸汽和电力。所以太阳能不会产生有毒有害物质，对于实现环境保护、走可持续发展之路显得更加可贵。

(4) 长久性：根据目前太阳产生的核能速率估算，氢的贮量足够维持上百亿年，而地球的寿命也约为几十亿年，从这个意义上讲，可以说太阳的能量是用之不竭的。

(5) 经济性：随着科技的进步，太阳能应用成本已经大大下降，而且随着人们对太阳能资源的认识逐步加强，许多专家和学者一致赞同太阳能是二十一世纪最清洁、最廉价的能源^[5-6]。

因此太阳能资源是可替代能源中最引人注目、开发研究最多、应用最广的清洁能源。虽然太阳能还有分散性、不稳定性、效率低和成本高的缺点，但是太阳能以其独特的优势必然成为未来最理想的绿色能源。

1.1.3 我国太阳能资源的分布情况

我国幅员辽阔，太阳能辐射资源比较丰富。据估算，我国陆地表面每年接收到的太阳辐射能量为 $5 \times 10^{19} \text{KJ}$ ，约相当于 1700 亿吨标准煤。全国各地太阳能年辐射总量达 $3350 \sim 8370 \text{MJ/cm}^2 \cdot \text{a}$ ，中值为 $5860 \text{MJ/cm}^2 \cdot \text{a}$ ^[7]。但是由于我国气候、地理环境条件的影响，太阳能资源分布具有明显的地域性。

我国华北，华东大部，东北部分地区以及西部地区的年日照时间大于 2000h，辐射总量高于 $5860 \text{MJ/cm}^2 \cdot \text{a}$ ，是我国太阳能资源丰富或较丰富的地区，面积较大，约占全国总面积的 2/3 以上，具有利用太阳能的良好条件。西南以及华南大部地区虽然太阳能资源条件较差，但仍有一定的利用价值^[8-9]。

1.1.4 本课题的研究意义

本课题研究一种太阳光自动追踪装置，它能够大大提高太阳能的利用率。该装置根据不同时间、不同地点的太阳方位自动调节采光板的角度，进而对太阳进行实时跟踪。使采光板始终垂直于太阳光的入射方向，最大限度的利用太阳能资源。这种装置符合可持续发展战略的思想，对提高太阳能的利用率有着非常重大的意义，具有巨大的环保价值和经济价值。

1.2 太阳能资源的开发和利用

1.2.1 太阳能资源的开发

我国早在 3000 年前的西周时期就利用“阳隧”取火，“阳隧”实际上就是凹面镜。但是将太阳能作为一种能源和动力加以利用，只有 300 多年的历史。20 世纪的 100 年间，太阳能科技的发展大体经历了七个历史阶段^[10]。我国是在 20 世纪 50 年代开始对太阳能利用进行研究。70 年代中期以来，随着世界兴起的太阳能研究和利用热潮，我国太阳能利用的研究项目范围不断扩大、水平不断提高。但是由于太阳能资源的分散性和不稳定性，致使太阳能的利用率很低。因此如何更充分提高太阳能的利用率成为广大研究人员急需解决的问题^[11]。本课题也正是针对这一现实问题提出来的。

1.2.2 太阳能资源的利用方式

目前，人类直接利用太阳能还处于初级阶段，主要有太阳能暖房、太阳灶、太阳能热水器、太阳灶、太阳能光电技术等方式^[12]。虽然我国太阳能开发利用在许多方面取得很大进展，但发展仍然参差不齐。

(1) 太阳能暖房：因寒冷地区冬季气温非常低，室内必须有暖气设备，为了节省煤炭资源，便利用太阳能取暖，这种方法已经使用多年而且越来越普遍。其过程为太阳辐射热传导，经收集器内的工作流体将热能储存，再供热至暖房^[13]。利用这种方式可以种植花卉、蔬菜、瓜果、培育优良品种、疗养院用来日光浴室进行医疗等等。

(2) 太阳能热水器：早期最广泛的太阳能应用是用于将水加热。目前世界各地的太阳能热水装置达数百万种。它的主要原件有收集器、储存装置和循环管路三部分。按照循环方式的不同可以分为自然循环式和强制循环式两种。我国是目前世界上最大的太阳能热水器生产国，技术已属国际先进水平，但是成本有待进一步降低。

(3) 太阳能发电：就是直接将太阳能转变成电能，并将电能存储在电容器中，以备需要时使用，这也是人类大规模利用太阳能的希望所在。目前世界上太阳能电池仍以晶体硅电池为主。我国从 1958 年就开始研制太阳能电池，在国际上具有一定影响力，但是在生产成本、效率水平、成品率以及工艺的稳定性等方面与国外差距较大，太阳能光电利用有待进一步开发^[14]。

(4) 太阳能集热器：是太阳能热系统中，接受太阳辐射并向传热工质传递热量的装置。按传热工质可以分为液体集热器和空气集热器。

除此之外还有太阳能干燥技术,我国在这方面的研究与应用发展较晚,70年代全国仅有7座太阳能干燥器。太阳灶理论研究已经初步成熟。我国研制出的太阳灶种类繁多,已经推广使用的太阳灶达14万台,居世界之首,在国际上产生较大影响。但是太阳能热发电、太阳能海水淡化、太阳能制冷和空调、高温太阳炉等技术还有待进一步发展^[15-16]。

1.3 太阳能跟踪装置的种类

地球和太阳的距离大约为1.5亿公里,而地球的直径大约1.3万公里,所以可以认为太阳光以平行线到达地球表面,所以及时调节太阳能采集器的角度使其处于太阳光的直射下,可以实现对太阳的实时追踪。实验证明,在相同条件下,采用自动追踪设备比固定装置太阳能的利用率可以提高35%^[17]。

太阳能跟踪器主要有以下两种分类方式。

1.3.1 按追踪方式

太阳能跟踪器按照追踪方式可以分为光电追踪方式和视日运动轨迹追踪方式两种。前者是闭环的随机系统,后者是开环的控制系统。

光电追踪是利用光敏传感器。这些传感器按照一定的规则排放在遮光板周围,当阳光的角度发生变化时,遮光板的阴影发生偏移,受到阳光照射的传感器和处于阴影中的传感器就会产生差值电流,然后经过处理器比较,分析出太阳的方向,再由伺服机构调整角度,使追踪装置对准太阳完成追踪^[18]。这种方式灵敏度高,而且设计结构简单。

视日运动轨迹系统根据追踪系统的轴数可以分为单轴追踪和双轴追踪两种。

美国的Blackace在1997年研制了单轴跟踪器,这种跟踪装置根据赤道坐标系下太阳运行的原理完成东西方向的自动跟踪,但南北方向通过手动调节,接收器的接收率提高了15%^[19]。采用这种追踪方式,一天之中只有正午时分太阳光与采光器相垂直,而在其它时间光线都是倾斜照射的。

如果能够在太阳高度和赤纬角的变化上都能够追踪太阳就可以获得更多的太阳能,全追踪即双轴追踪就是根据这样的要求而设计的。1998年美国加州成功的研究了ATM双轴跟踪器^[20],使太阳光接收率进一步提高。

1.3.2 按传感器元件

(一)机械式跟踪器

这是一种被动式的跟踪装置，可以有单轴和双轴两种形式。这种跟踪装置通过电机以恒定的速度带动太阳能采集器运动来实现对太阳运行轨迹的跟踪^[21]。这种跟踪装置的主要优点是结构简单，便于制造，并且该装置的控制系統也十分简单，但主要缺点是跟踪精度不够。

(二) 压差式跟踪器

采用这种跟踪装置时，太阳能采集器南北放置。为了得到太阳的角度信息，在反射镜周边设有一组空气管作为时角的跟踪传感器，根据太阳偏移时空气管因受阳光照射不同产生的压差带动采光板跟踪太阳^[22]。这种跟踪器的灵敏度非常高，清晨太阳刚刚升起三到五分钟，太阳能采集器就可以对准太阳。

这种跟踪器在实际中应用很广。优点是结构简单、制作费用低。缺点是刚度低，没有足够的工作空间，且一般只用于单轴追踪。

(三) 控放式太阳跟踪器

这种方式原理复杂而且只能对太阳方位角进行单向追踪。它是在太阳能采集器西侧安放一偏重，作为其向西转动的动力，并利用控放式自动跟踪装置对此动力的释放加以控制，使太阳能采集器随着太阳的西偏而转动^[23]。其优点是实时跟踪，成本低廉，可以不用外接电源，使收集到的能源充分转化利用。主要缺点是机构只能做单轴跟踪，而且存在跟踪过度、刚度较低的问题。不适应野外恶劣的环境。

(四) 计算机控制的全自动跟踪装置

随着电子技术的发展，许多国家研发出了计算机控制的全自动太阳跟踪器。这种跟踪装置具有全天候、全自动、跟踪精度高、无累积误差、不绕线的优点，可以满足太阳辐射观测的需要，大大减轻观测人员的劳动强度^[24]。

(五) 用于天文观测和气象台的太阳跟踪装置

除了在太阳能电池板以及其他太阳能采集器的用途以外，太阳跟踪装置还经常用于天文台和气象台对太阳的跟踪观察。这种跟踪装置主要应用于科研，由于精度非常高，所以造价也非常昂贵，因与本课题所涉及的应用领域不同，所以不再介绍。

1.4 嵌入式技术的发展

嵌入式系统在 IEEE（国际电气和电子工程师协会）中的标准定义是：An Embedded System is the devices to control, monitor, or assist the operation of equipment, machinery or plant.但是通用的定义为：嵌入式系统是以应用为中心，以计算机技术为基础，并且软硬件可裁剪，应用于对功能、可靠性、成本、体

积、功耗有严格要求的专用计算机系统^[25]。

一般地，我们把嵌入式系统的体系结构分为四个部分：嵌入式处理器、嵌入式外围设备、嵌入式操作系统和嵌入式应用软件。如图 1-1 所示。

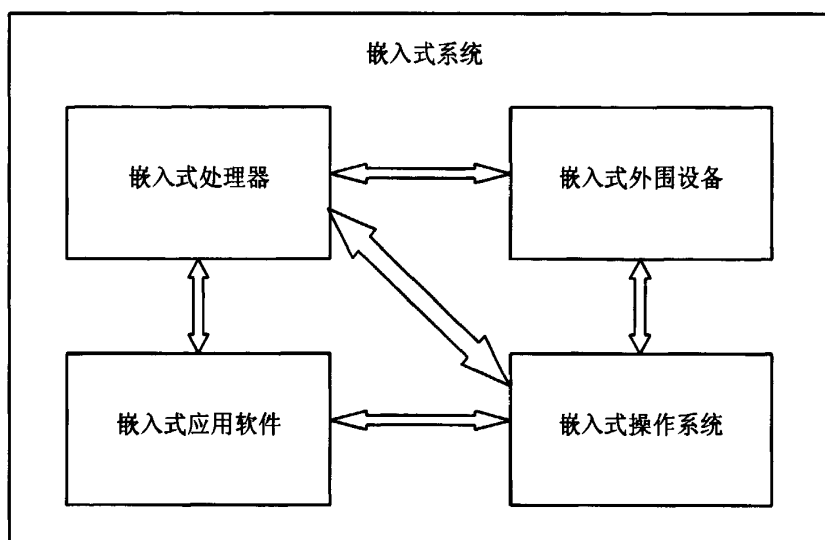


图 1-1 嵌入式系统的组成

嵌入式技术的发展主要经历了以单片机（SCM）、微控制器（MCU）、系统级芯片（SoC）为代表的三个阶段^[26]。

1.4.1 嵌入式处理器

嵌入式处理器是嵌入式系统的核心，是控制、辅助系统运行的硬件单元。范围极其广阔。嵌入式处理器一般分成以下四类。

（一）嵌入式微控制器（Microcontroller Unit，MCU）：典型代表是单片机，就是将整个计算机系统集成到一块芯片中，主要完成信号控制的功能。

（二）嵌入式微处理器（Microprocessor Unit，MPU）：微处理器的最大特点是单片化，体积大大减小，从而使功耗和成本下降、可靠性提高。

（三）嵌入式 DSP 处理器（Digital Signal Processor，DSP）：独特的微处理器，该器件是以数字信号来处理大量的信息。它的工作原理首先是接收模拟信号，然后转换为数字“0”或“1”，最后对数字信号进行修改和删除等处理。

（四）片上系统（System On Chip，SOC）：在单个芯片上集成一个完整的系统，对所有或部分必要的电子电路进行包分组的技术^[27]。

1.4.2 嵌入式外围设备

嵌入式外围设备是指中心控制部件以外，用于完成存储、通信、调试、显示等辅助功能的其他部件，主要包括以下几类。

(一)实时时钟：用来提供可靠的时钟信息，包括时分秒和年月日。即使系统处于关机或停电状态，实时时钟通过后备电池供电也能继续正常工作。

(二)存储设备：提供执行程序 and 存储数据所需空间，分为 RAM 和 ROM 两种。

(三)输入设备：向计算机输入数据和信息的设备。是计算机与用户或其他设备通信的桥梁，分为小型键盘和触摸屏两种。

(四)输出设备：是人与计算机交互的一种部件，用于数据的输出，分为 LED 和 LCD 两种。

(五)外设接口：就是连接外设的接口，分为并行接口和串行接口。

(六)通讯接口：是指中央处理器和标准通信子系统之间的接口，分为有线通信接口和无线通信接口^[28-29]。

1.4.3 嵌入式应用软件

它是针对某一特定的应用领域，而且基于一种固定的硬件开发平台的软件，可以用来实现客户预先设定的任务。因为用户的任务存在很大的差异，所以有些嵌入式应用软件需要特定嵌入式操作系统的支持。嵌入式应用软件不仅要求准确性、安全性和稳定性等能够满足实际应用的需要，而且还要尽可能的进行优化，以减少系统资源的消耗，降低硬件成本^[30]。

1.4.4 嵌入式操作系统

嵌入式操作系统是嵌入式系统的灵魂，它能够有效管理复杂的系统资源，完成进程管理、处理器调度、存储管理、设备管理、中断处理等任务，而且与通用操作系统比它还有内核小、专用性强、系统小而精、使用多任务操作系统、有专门的开发配套工具等特点^[31]。

嵌入式系统的发展经历了 30 年的历史，大致经历了无操作系统阶段、简单操作系统阶段、实时操作系统阶段和面向 Internet 四个阶段^[32]。

目前可选的成熟的操作系统有很多，例如 Linux、VxWorks、Windows CE、μC/OS 等。其中 Windows CE 操作系统具有丰富的硬件驱动程序，多媒体功能极其强大，实时性也较为突出，支持 x86、ARM、MIPS 等架构的 CPU。所以本课题选取的是 Windows CE 操作系统。

在市场和技术的双重推动下，嵌入式系统技术未来的发展将呈现出以下趋势。

- (一) 互联网成为必然趋势。
- (二) 小尺寸、微功耗和低成本。
- (三) 提供精巧的多媒体人机界面。
- (四) 无所不在的智能^[33]。

1.5 本文的研究内容

本文所研究的太阳能自动追光系统是以 S3C2440 芯片为处理器，以 WinCE 为操作系统的自动控制系统。整个装置主要由电路部分和机械部分两大部分组成。但是机械装置不是本论文讨论重点，所以本论文主要对电路的硬件和软件两大部分进行讨论。具体研究如下几方面的内容。

- (一) 提出一种新的计算太阳高度角和方位角的原理，采用图像处理技术，原理简单，计算精确。
- (二) 双轴跟踪，并采用嵌入式技术对采光板实时调整，追光更加精确，没有累计误差，并不受时间和地域的限制。
- (三) 硬件电路包括摄像头电路，步进电机电路以及调试接口电路。
- (四) 软件部分包括操作系统的定制，驱动电路的编写。

第二章 太阳光智能追踪系统的设计方案

在本课题研究过程中，为了开发出能实时跟踪太阳光的装置，首先必须解决如何追光，所以本章首先介绍如何确定太阳和地球的相对位置，这是追光的前提。为了便于描述，我们的追踪装置是对太阳能电池板为例进行的论述。

2.1 实验原理

在地平坐标系上，我们规定地球上某一观测点作为原点，用 O 来表示，即太阳能电池板的位置。对于原点 O ，某一时刻该位置相对于太阳的高度角和太阳方向角可以分别用 α 、 β 来表示。只要确定了太阳高度角和方向角就能准确确定地球上任意一点与太阳之间的相对位置，进而调整太阳能电池板所在平面，使其垂直于太阳光线来达到最大限度利用太阳能的目的。

为了正确找到任意时刻和时间的太阳高度角和方向角，我们在水平面上垂直立一个标杆，则标杆在阳光下必然形成影子，以物体的底部位置为原点，以正南方向为 X 轴，正东方向为 Y 轴建立坐标系，如图 2-1 所示。

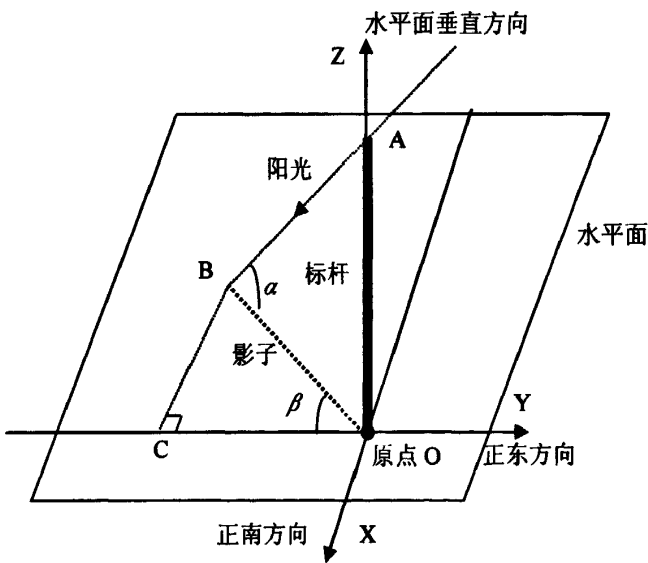


图 2-1 实验原理图

水平面坐标系下，影子与 Y 轴的夹角正是太阳方位角 β 。而高度角 α 的大小又可以通过预先测量的物体高度与影子的长度计算求得。

设水平面上垂直物体标杆为 AO，形成影子 BO，建立直角坐标系。在坐标系上我们可以通过作图和反三角变换可以很方便的得到高度角和方向角的大小。

$$\alpha = \arctan \frac{AO}{BO}, \quad \beta = \arctan \frac{BC}{OC} \tag{2-1}$$

2.2 传感器部分的设计

传感器采用摄像头采集日影图像，示意如图 2-2 所示。

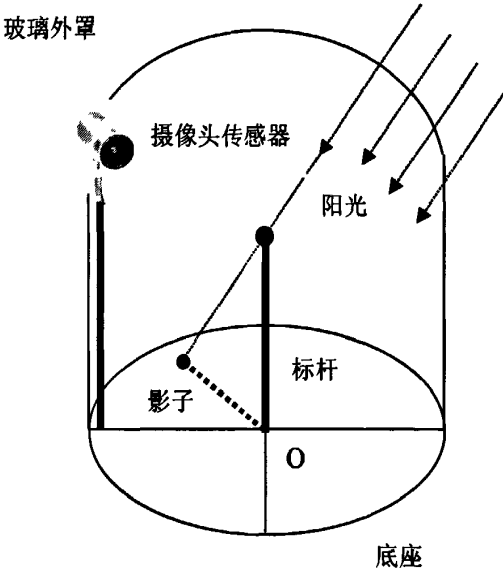


图 2-2 传感器设计图

传感部分主要有四部分组成：底座、指针、玻璃罩和摄像头。安装时摄像头支座与指针组成的直线正对当地的正南正北方向。当阳光照射过来时，指针形成的影子正好被摄像头捕捉到，这样就能把图像信息传递给处理器处理。底座可以固定在平坦的地面上，玻璃罩的作用是防止尘土弄脏摄像头。

2.3 计算高度角和方向角

由摄像头拍摄的图像如 2-3 所示。

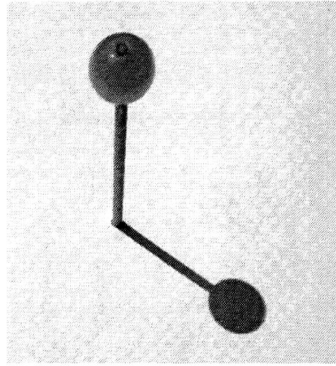


图 2-3 传感器采集的图像

为了能更精确的测量标杆影子的长度，我们在标杆顶端增加了一个球形部分。图像首先经过去灰处理，再经过二值化处理之后的图如 2-4。

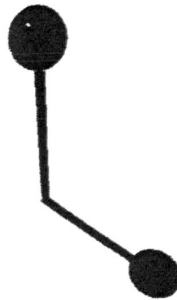


图 2-4 二值化后的图像

再经过去噪处理之后的图像数据存入一个二维数组，用数字“1”和“0”表示。“1”代表背景，“0”代表影子。本次试验中最关键的一点在于找到图 2-1 中的点 B，因为我们事先设定视野中正中间的点为原点 O。因此只要找到了点 B，高度角和方向角就会很轻易的求得了。因此我们设计的球形部分的中心就是 B 点，这样我们扫描数组中每一个 $m \times n$ (n 为奇数) 的矩阵，计算每个矩阵的行和列的数值的和，和最小的矩阵的正中间点我们视为图 2-1 中的点 B。

在对传感器采集到的图像信号进行二值化处理时，我们采取浮动阈值法，这是因为考虑到环境光线对传感器影响很大，所以设置系统的图像阈值 ϵ 采用以

下方式进行分割^[34]：

$$f(x, y) = \begin{cases} 1 & u(x, y) > \varepsilon \\ 0 & u(x, y) \leq \varepsilon \end{cases} \quad (2-2)$$

其中 $u(x, y)$ 为原始数据， $f(x, y)$ 为二值化后的数据， ε 为图像阈值。

2.4 采光板的角度调整原理

前后两次计算得到的高度角和方向角的差值是太阳能电池板角度调整的判断条件。在本系统中，我们可以采取两种方法进行控制。

1. 分布律法

我们以高度角为例，其原理是这样的：我们假设前后两次测得的高度角分别为 α_1 、 α_2 ，差值为 $\Delta\alpha$ （ $\Delta\alpha$ 可以为正，也可以为负）。然后依据差值 $\Delta\alpha$ 的分布情况对太阳能电池板角度的调整进行分类，我们把它分为如下公式的五种情况^[35]：

$$\theta_{(x)} = \begin{cases} \theta_1 & \Delta\alpha < \varepsilon_1 \\ \theta_2 & \varepsilon_1 \leq \Delta\alpha < \varepsilon_2 \\ \theta_3 & \varepsilon_2 \leq \Delta\alpha < \varepsilon_3 \\ \theta_4 & \varepsilon_3 \leq \Delta\alpha < \varepsilon_4 \\ \theta_5 & \varepsilon_4 \leq \Delta\alpha < \varepsilon_5 \end{cases} \quad (2-3)$$

其中 $\theta_{(x)}$ 是当前采光板的角， $\theta_1 \sim \theta_5$ 是调整后的角。

2. 反馈法

由于机构的方位轴和俯仰轴上都安装有光电编码器反馈系统，我们就可以很容易的检测得到其转动的角度信息。原理是这样的：假设计算得到的方向角和高度角差值分别为 $\Delta\alpha$ 、 $\Delta\beta$ （ $\Delta\alpha$ 、 $\Delta\beta$ 可以为正，也可以为负）。由正负号我们可以控制电机是正转还是反转，然后控制电机转动，直到得到反馈系统的停止信号为止。

反馈系统的光电编码器可以很精确的计算太阳能电池板转过的角度，如果两个角度恰好为 $\Delta\alpha$ 和 $\Delta\beta$ 时，就会输出停止信号，通知电机停止转动。这种方法比前一种控制更加精确，且不会出现误差积累的情况。

为了得到准确的信息，每种方法我们都可以多次测量取平均值。以此来保

证步进电机的转动。由于两相电机互不干扰，所以两个方向的转动可以同时进行。

2.5 机械结构的设计

本文采用的是地平坐标系的跟踪方法，它是根据当地日影计算出此时太阳的高度角和方位角，从而控制步进电机实现电池板跟踪太阳的目的。这种方法是双轴跟踪，所以采光效率更高。跟踪装置的机械机构如图 2-5 所示^[36-37]。

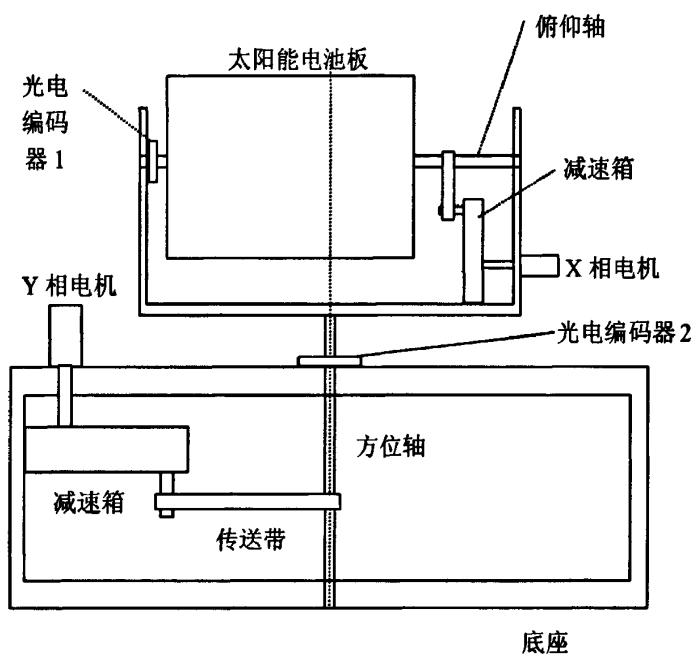


图 2-5 太阳能跟踪装置机械结构设计图

安装时处于水平方向的底座正对当地的南北方向。

X、Y 两相步进电机经过减速箱减速后输出力矩更大，转动角度更加精确，经过传送带可以带动方位轴和俯仰轴的转动，每个轴上面均带有光电编码器，处理器以此来检测两个轴的位置，进而确定太阳能电池板的方向。

跟踪器实现自动跟踪的原理：当太阳光线发生偏移的时候，控制器发出控制信号驱动两相步进电机转动，减速箱经过传送带带动两个轴转动一定的角度，最终在 X、Y 两相步进电机的共同工作下实现对太阳方位角和高度角的跟踪。

2.6 本章总结

本章阐述的内容是系统的总体设计方案。首先分析了本课题的实验原理，提出了一种新的可行办法，其次分析了具体实施的细节，包括机械部分的设计和传感器部分的设计等。本章从宏观上描述了系统的总体设计。下两章将从电路角度分别对硬件和软件进行分析。

第三章 系统硬件设计

太阳能智能追光系统是一整套系统共同协调完成工作的。前一章主要讲了本实验的原理和设计思想，本章将介绍硬件电路的设计。

3.1 系统硬件总体设计方案

硬件电路主要有五大模块组成：中央处理器、摄像头模块、存储器模块、调试接口模块和步进电机模块。系统总体设计框图如图 3-1 所示。

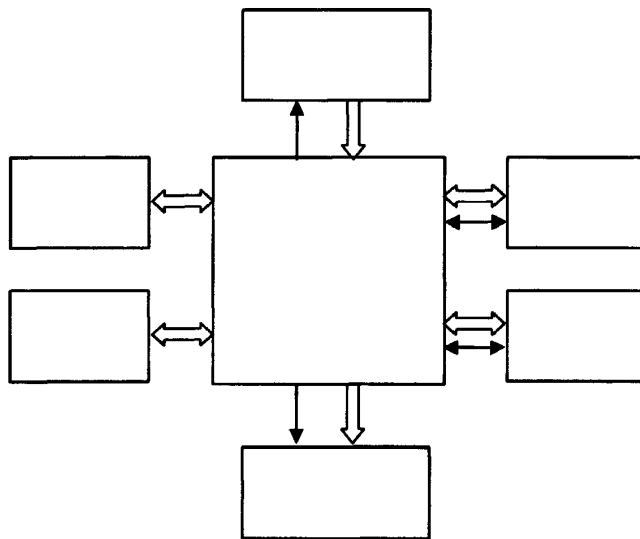


图 3-1 系统硬件框图

系统上电后首先进行初始化，无误后读取系统时间判断是否是白天，并读取摄像头图像信息判断光照强度是否适当。如果是黑夜，或者阴天，系统进入休眠状态以节省电能。如果是在早上 6:00 到晚上 18:00 之间且光照强度足够，进入系统主程序。中央处理器会每两个小时读取一次摄像头信息，判断此时太阳高度角、方位角。然后控制步进电机的转动，使采光板实时处于阳光直射下，到了晚上光照强度很弱，进入休眠状态，等到第二天早上系统能够自动唤醒。主程序流程图如图 3-2 所示。

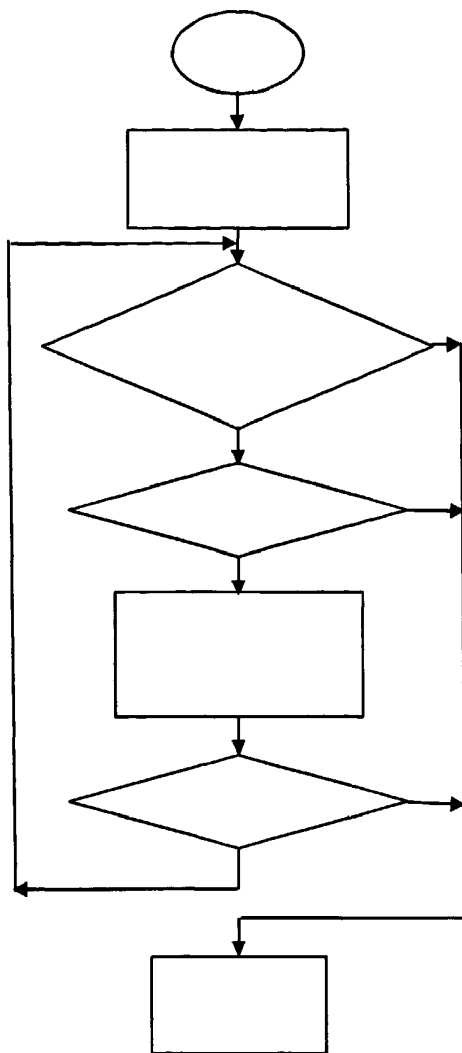


图 3-2 系统主程序流程

下面分别对各个模块进行详细介绍。

3.2 S3C2440 介绍

本课题采用的是由飞凌嵌入式技术有限公司设计生产的一款基于 ARM9 的嵌入式开发平台，其布局和走线经过专业人士精心设计，工作非常可靠，可稳定运行在 400MHz，接口丰富，功能强大，适用于各种手持设备，消费电子和工业控制设备的开发。具有高性能、低功耗、接口丰富和体积小等优良特性^[38-39]。OK2440 开发板硬件资源非常丰富，开发板及其硬件资源如图 3-3、3-4 所示。

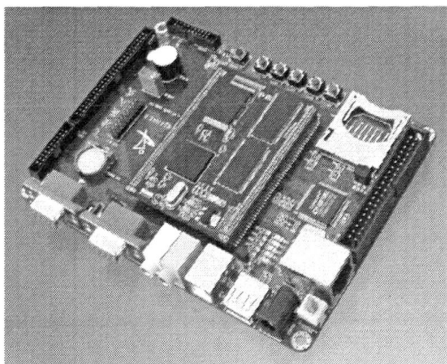


图 3-3 飞凌 OK2440 开发板

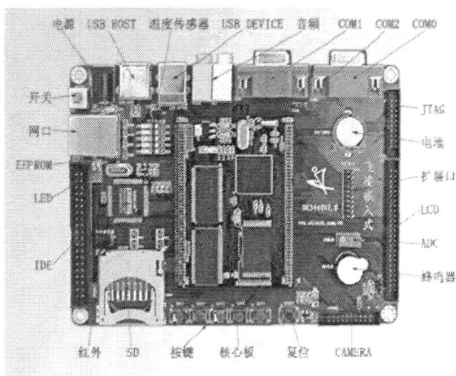


图 3-4 飞凌 OK2440 芯片板硬件组成

3.3 步进电机控制设计

3.3.1 步进电机的选择

选择步进电机时首先要知道机械和时间这两个要素。机械要素是指负载转矩 T_L 和负载惯量 J_L ，时间要素是指加速时间 t_1 和 t_2 (即从 t_1 开始加速到 t_2)，运行时间 $t^{[40]}$ 。

在本实验中，我们为两个步进电机安装上了不同减速比的减速箱，这样可以增加电机的输出力矩。

实际使用时，由于考虑到装置的摩擦力很小，控制系统对步进电机的转速要求不高，因此在参考了步进电机厂家的相关资料后，决定选用细分后步距角为 0.045° 系列的步进电机。

在查阅了步进电机的产品参数资料后，考虑到所选用的电机和驱动器的特性通常留有 1.5~2 倍的余量，所以本系统选用常州双杰电子有限公司生产的

57BYGH8404、42BYGH5403 两种步进电机。表 3-1 列出两种电机的主要相关参数。

表 3-1 步进电机参数表

型号	步距角	电流	电阻	电感	静力矩	引线	重量
57BYGH8404	1.8°	3.0A	0.5Ω	1.8mH	1.66N·m	4	1050g
42BYGH5403	1.8°	1.8A	1.2Ω	1.8mH	0.49N·m	4	340g

图 3-5 为电机外型。

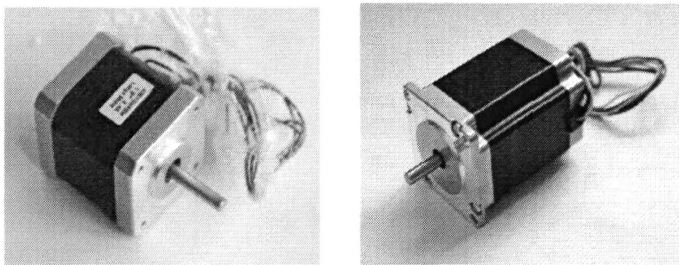


图 3-5 步进电机

3.3.2 步进电机电路

步进电机驱动电路设计如图 3-6 所示。

控制器部分是我们上面讲到的 ARM9 芯片 S3C2440，其标称工作频率 400MHz，运算速度快、性能优越，是一款低功耗，高性价比的工程用处理器，在嵌入式开发中应用比较广泛，也是太阳能自动追光装置高性能的保证。

光电编码器可以精确的反馈转轴的转速和当前的相对位置，系统用此确定方向轴和俯仰轴的定位。两个光电编码器分别安装在两个转轴上，在太阳能电池板旋转过程中，为控制系统提供三相光电编码器返回值，转轴旋转一周，A、B 相返回 1000 个脉冲信号，Z 相返回一个脉冲信号，三相脉冲返回值反馈给 CPU，对转轴位置的反馈精度可达 0.36°，设计中采用 CPLD 的计数器单元 Count1 到 Count4 分别对两相光电编码器的 A 相和 Z 相进行计数操作^[41]。

CPLD 模块采用 Altera 公司 MAX II 系列的 EPM1270 芯片。EPM1270 成本低、质量优异、功耗低，适合接口交接、I/O 扩展、器件配置等。CPLD 为整个追光系统提供了 4 个计数器和 2 个定时器^[42]。

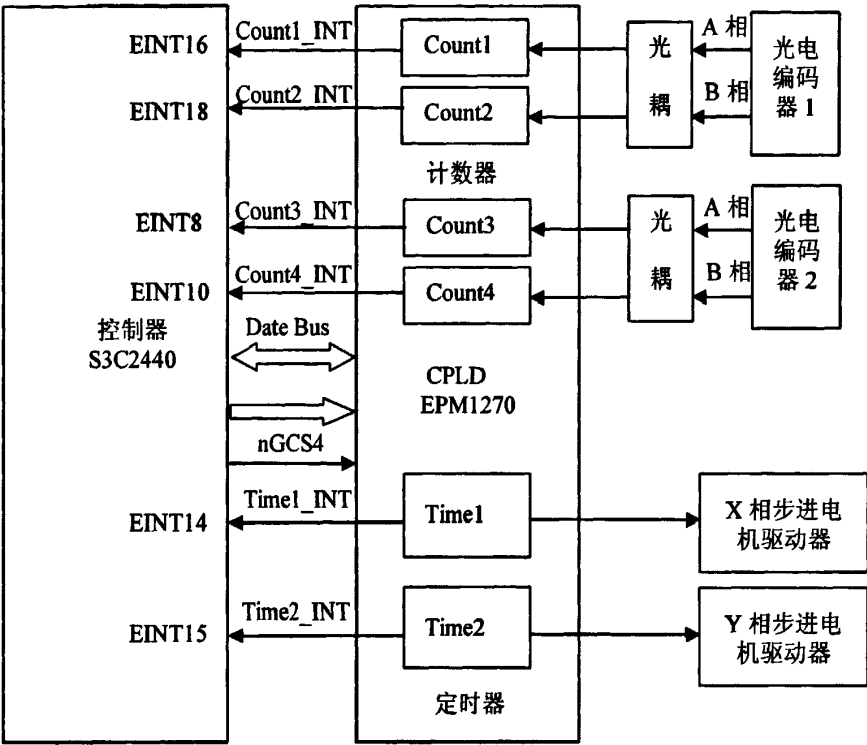


图 3-6 步进电机控制系统硬件框图

3.4 摄像头传感器的设计

OV9650 是新近推出的一款低电压 CMOS 彩色图像传感器，130 万像素，通过 SCCB 接口设置内部寄存器。支持 SXGA, VGA, QVGA, QQVGA, CIF, QCIF, QQCIF 等多种图像输出格式，包含自动曝光控制，自动增益控制，自动白平衡等功能，最大速率在 SXGA 格式时为 15fps，在 VGA 格式时为 30fps^[43]。

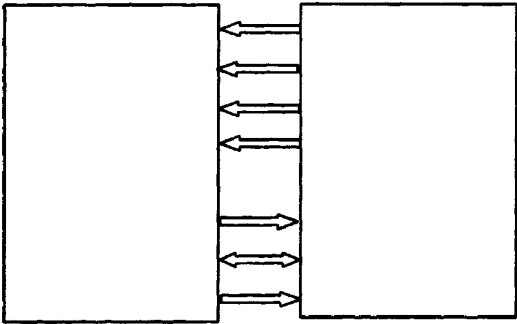


图 3-7 CAM IF 硬件接口

图 3-7 为 CMOS9650 摄像头与 S3C2440 硬件接口设计框图，其中对摄像头的控制部分中，我们只需将 IIC 总线的 SDA 和 SCL 信号线与控制器相连接即可^[44]。

3.5 调试接口设计

3.5.1 串口调试接口

OK2440 内部集成了 3 个 UART(Universal Asynchronous Receiver/Transmitter, 通用异步收发器)接口控制器。本系统设计了一个调试串口用于向开发平台输出调试信息。串行通信是微计算机领域一种常见的近距离通信手段,因使用方便,编程简单而广泛使用。UART 可以产生中断请求或 DMA 请求，以便在 CPU 和 UART 之间传递数据^[45]。

在程序的调试过程中，串口起到了非常重要的通信作用，程序在开发板上运行时，它可以随时在电脑上显示其运行情况，一旦系统出现故障，也可以随时通过串口对程序进行调试，超级终端界面如图 3-8 所示。

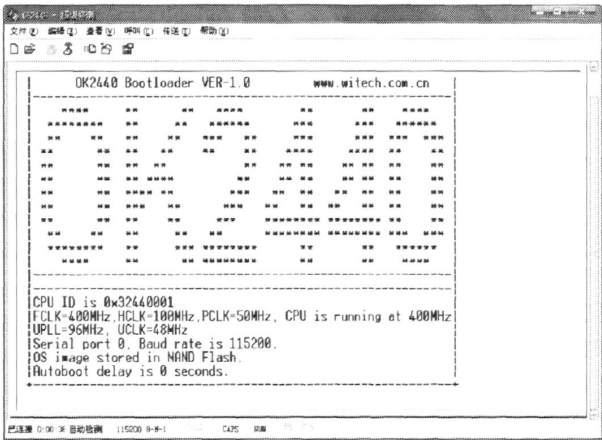


图 3-8 超级终端串口调试界面

3.5.2 JTAG 调试接口

JTAG(Joint Test Action Group, 联合测试行为组织)是一种国际标准测试协议，主要用于芯片内部测试和系统仿真与调试。它通常在芯片内部封装了 TAP(Test Access Port, 测试访问接口)，通过专用的 JTAG 测试工具对内部节点进行测试^[46]。

JTAG 仿真器连接比较方便，它可以通过现有的 JTAG 边界扫描与 ARM CPU 核进行通信，属于完全非插入式(不占用片上资源)调试^[47]。它无需目标存储器。在系统的开发过程中，我们用下载软件 Flash Programmer 将生成的十六进制文件通过 JTAG 接口下载到 Flash 芯片的地址单元里。当系统上电时，就可以启动程序了。

20 针的 JTAG 接口电路如图 3-9 所示。

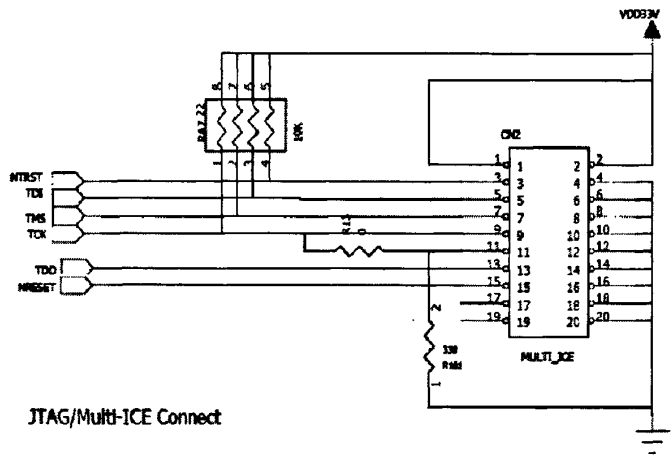


图 3-9 20 针 JTAG 接口电路

3.6 本章小结

本章主要阐述了太阳能智能追光系统的硬件部分的设计。首先介绍了硬件电路的总体设计，然后分别对中央处理器、摄像头模块、调试接口模块和步进电机模块这四个模块分别进行详细介绍。

第四章 Windows CE 操作系统的定制和移植

本文选用 Windows CE 操作系统，嵌入式操作系统最大的特点在于系统的组件可选择，根据系统需求和应用背景选择不同的操作系统组件。

4.1 几种常见的嵌入式操作系统

一般情况下，嵌入式操作系统一般可以分为两类，一类是面向控制、通信等领域的实时操作系统，如 WindRiver 公司的 VxWorks、ATI 的 Nucleus、QNX 系统软件公司的 QNX、ISI 的 pSOS 等；另一类是面向消费电子产品的非实时操作系统，如 WinCE、Linux 等^{[48][49]}。

4.1.1 VxWorks

VxWorks 是美国风河公司（WindRiver）于 1983 年设计开发的一种实时嵌入式操作系统（RTOS），是目前嵌入式系统中使用最广泛、市场占有率最高的操作系统^[50]。它支持多种处理器，如 x86、i960、Sun Sparc、Motorola MC68xxx、MIPS RX000、POWER PC 等，大多数的 VxWorks API 是专有的。

VxWorks 的实时性做得非常好，其系统本身的开销也很小，进程调度、进程间通信、中断处理等系统公用程序精练而有效，延迟很短。在多任务机制中 VxWorks 对任务的控制采用了占先式（preemptive priority scheduling）和轮转调度（round-robin scheduling）机制，充分保证了可靠的实时性^[51]。

4.1.2 Windows CE

Windows CE 是所有源代码全部由微软自行开发的嵌入式新型操作系统。Windows CE 操作系统提供与 PC 类似的界面和主要应用程序，在 Windows CE 平台上可以使用同样的函数、同样的界面风格，使绝大多数的应用软件只需简单的修改和移植就可以在 Windows CE 平台上继续使用^[32]。这样熟悉 PC 机 Windows 系统的用户能很快就会学会使用它。

Windows CE 操作系统的基本核心需要至少 200KB 的 ROM，它支持 Win32 API 的子集，支持多种的用户界面硬件，同时提供熟悉的开发模式和工具。

4.1.3 嵌入式 Linux

自由免费软件 Linux 是一个类似于 Unix 的操作系统。嵌入式 Linux 由于代码开放及强大的网络功能，在嵌入式产品的开发中具备巨大的潜力。

嵌入式 Linux 的优点如下：

（一）Linux 是由很多体积小且性能高的微内核系统组成。

（二）Linux 具有非常完备的工具链，在嵌入式系统开发的过程中，可以跨越仿真工具的障碍。

（三）使用嵌入式 Linux，一旦软硬件能够支持正常的串口功能，即使不用仿真器，也可以很好地开发和调试程序。

（四）Linux 具有广泛的硬件支持特性。

（五）实时性是嵌入式操作系统的基本要求^[53-54]。

4.1.4 Palm OS

由 Palm Computing 公司的嵌入式操作系统，目前最大的应用在 PDA(Personal Digital Assistant)，是市场占有率最高的 PDA 操作系统，Palm 操作系统架构非常简洁，因为少去了很多功能，如内存管理、多任务等等，使得 Palm 可以非常不耗系统资源，硬件需求低，连带的整体耗电量便可压缩到非常低，因此采用 Palm 操作系统的 PDA 都有待机时间长的优点。

4.2 Windows CE 操作系统简介

Microsoft Windows CE 是一个开放且多样化的 32 位嵌入式操作系统。被广泛地应用于设计各种移动和数字设备。

Windows CE 采用层状的结构，可移植性好，组件化可剪裁，强调编程接口和支持上层应用等。从上而下可分为四层：应用层，操作系统服务层，OEM 硬件适配层，硬件层^[55]。其结构如图 4-1 所示。

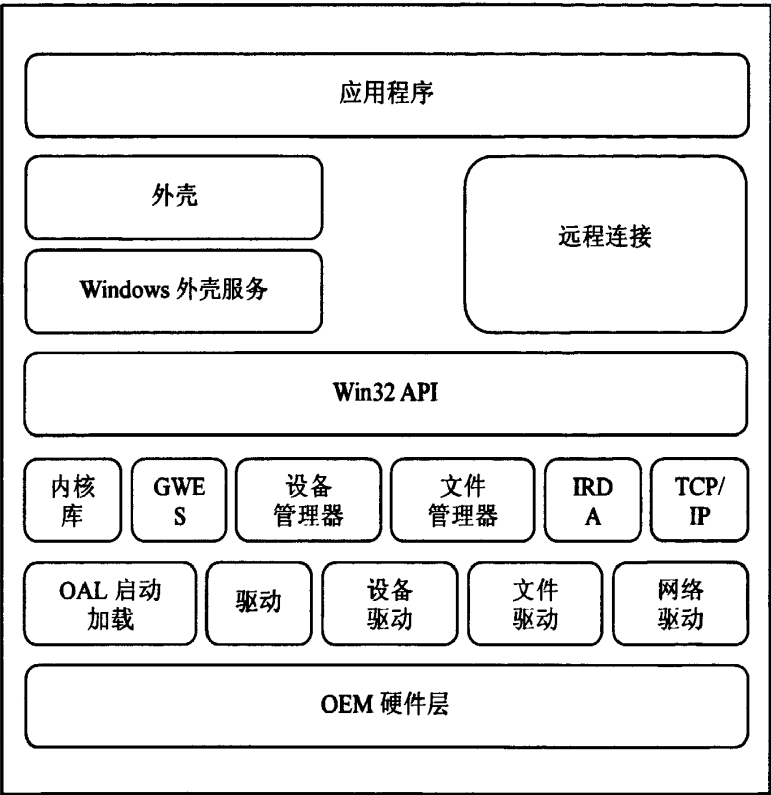


图 4-1 WindowsCE 系统结构图

4.3 构建基于 OK2440 的 Windows CE 5.0

Windows CE 嵌入式操作系统功能强大，可移植性强，可以广泛应用于各种需要，实现多任务相对较复杂的基于不同微处理器为核心的实际系统开发中。对于各实际应用开发，需要根据所选微处理器以及系统功能需求，对 Windows CE 嵌入式操作系统进行移植，并对各应用中没有用到的功能进行裁剪，以缩减程序代码量，节约存储器空间，使 Windows CE 操作系统高效运行于所设计系统中^[56]，Windows CE 嵌入式操作系统的移植与定制如图 4-2。

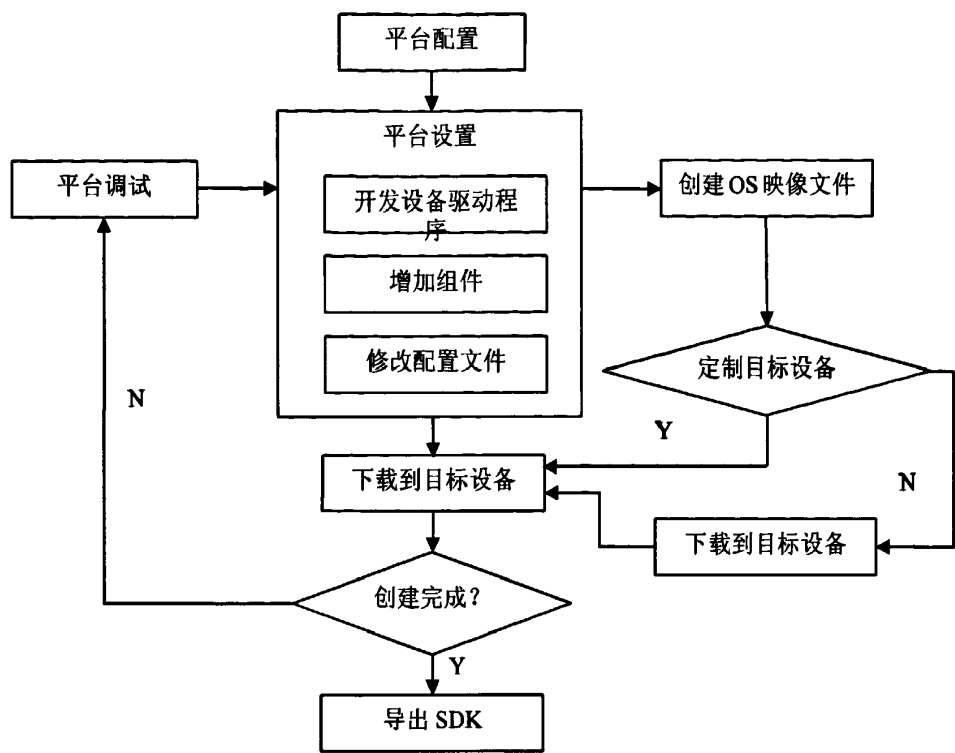


图 4-2 操作系统移植流程

4.3.1 安装基于 OK2440 的 BSP 包

首先在 PC 机上安装 Windows CE 5.0 开发环境，它的安装不仅包括运行于不同 CPU 构架的 WinCE 5.0 操作系统软件包，还包括其开发工具 Platform Builder 5.0(简称 PB5)。

微软的 Platform Builder 5.0 集成研发环境包含一系列研发工具，利用它提供的配置文档和调试工具能够快速定制适合目标平台的 WinCE 操作系统，只要适当修改和配置 PB5 的配置文档就能够裁剪和添加相关组件，和配置各种设备驱动程序。

要使 PB5 编译 WinCE 内核映像，需要安装对应目标板的板级支持包，即 BSP(Board Support Package)，并进行一些配置。BSP 是介于主板硬件和操纵系统之间的一层，属于操纵系统的一部分，其主要目的是为了支持操纵系统，使之能够更好的运行于硬件主板。BSP 是相对于操纵系统而言的，不同的操纵系统对应于不同定义形式的 BSP^[57]。

安装完成之后，会在 PB5 的目录查看器上看到“OK2440: ARMV4I”列。如

图 4-3 所示。

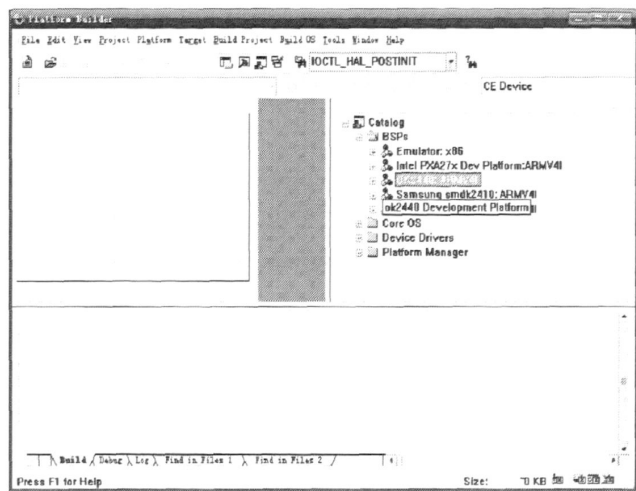


图 4-3 安装 BSP 包后 Platform Builder 5.0 运行界面

4.3.2 内核定制

通过 PB 构建操作系统时，一般是通过 PB 所提供的菜单来设定其配置参数，对系统进行配置。方法是展开上图中右边 Catalog 列表在要添加的模块上点击右键，在右键菜单里选择 Add to OS Design 即可，在这里我们可以选择添加的组件有：

- (一)添加 USB 鼠标和键盘支持。
- (二)添加 USB 存储设备的支持。
- (三)添加 SD 卡支持。
- (四)添加注册表保存功能。
- (五)添加图片浏览器。
- (六)添加 MP3、MP4 支持。

工程设置和驱动添加完成以后，接下来开始编译 Windows CE 5.0 操作系统映像。编译成功就可以产生 Windows CE 5.0 的 2 个二进制映像文件“nk.bin”和“nk.nb0”。将“nk.nb0”下载到 Windows CE 5.0 到 ARM2440 开发板上，重启开发板，这样 Windows CE 5.0 操作系统就可以在开发板上运行了。

4.4 建立应用程序开发环境

4.4.1 SDK 的导出与安装

我们可以把定制好的内核工程导出为 SDK（Software Development Kit）安装文件，它用来提供给应用开发人员，里面主要包括与定制平台有关的头文件、库和一些文档等内容。方法是重新启动 PB5，打开已经创建好的 OK2440 操作系统设计平台，通过软件开发包导出定制操作系统的 SDK 软件开发工具包。导出的文件为 OK2440_SDK.msi。这个是一个标准的 Windows installer 安装文件，双击鼠标即可安装。

4.4.2 安装 eMbedded Visual C++ 4.0

本系统中 Windows CE 5.0 平台上的应用程序开发采用 eMbedded Visual C++ 4.0 (简称 EVC)来开发。是用于创建 WinCE 应用程序的一个集成开发环境。应用程序开发是相对于驱动程序和内核而言的。EVC 的界面和 VC 差不多，开发者可以为 WinCE 开发 Win32、MFC、ATL 应用程序。采用 C/C++语言编写 Windows 应用程序，可以方便的调用 WIN32 API 函数^[58]。应用程序开发流程如图 4-4 所示。

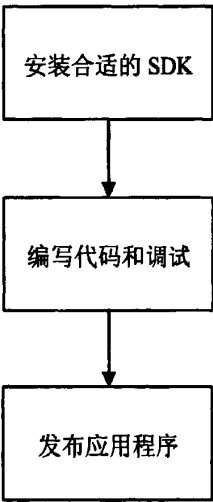


图 4-4 WinCE 下应用程序开发流程

4.5 使用微软 ActiveSync 实现与 PC 机的同步通讯

可以使用微软公司提供的一些工具，并在进行了一些必要的设置之后，就可以在安装了 WinCE 操作系统的移动设备和 Windows 桌面系统之间进行通讯连接，从而可以实现文件上传下载，远程调试等功能。

Microsoft ActiveSync 是一款基于 Windows Mobile 的设备同步软件。在系统调试阶段，用 USB 将开发板和 PC 机连接起来，就可以直接对开发板进行文件管理了。不过需要提前在开发板 BSP 工程文件中加入 ActiveSync 组件，在 PC 端安装了 ActiveSync 软件才可以。

4.6 本章小结

本章首先介绍了几种常用的嵌入式操作系统，然后重点介绍了 Windows CE 操作系统。通过对几种不同的操作系统的比较，我们最终选择了 Windows CE 操作系统。最后给出了详细的 Windows CE 操作系统的移植和开发流程。从而对 Windows CE 操作系统有了更深入的了解。

第五章 驱动程序的开发

使用嵌入式操作系统时，有时需要自己开发设备的驱动程序。在本课题中，系统需要开发摄像头的驱动程序，步进电机模块需要的 GPIO 驱动程序。本章将重点介绍摄像头的流接口驱动程序开发。

5.1 Windows CE 5.0 驱动程序开发基础知识

5.1.1 Windows CE 5.0 驱动模型分类

目前，Windows CE 5.0 提供了 4 种设备驱动程序模型，其中两种是专用于 WinCE 的模型，另外两种模型来自其他操作系统^[59]。

a) 基于 WinCE 的两种驱动程序模型是本机设备驱动程序和流接口驱动程序。

b) 另外两种模型是通用串行总线(USB)驱动程序和网络驱动接口规范(NDIS)驱动程序^[60]。

各种驱动对应的驱动程序模型如图 5-1 所示。

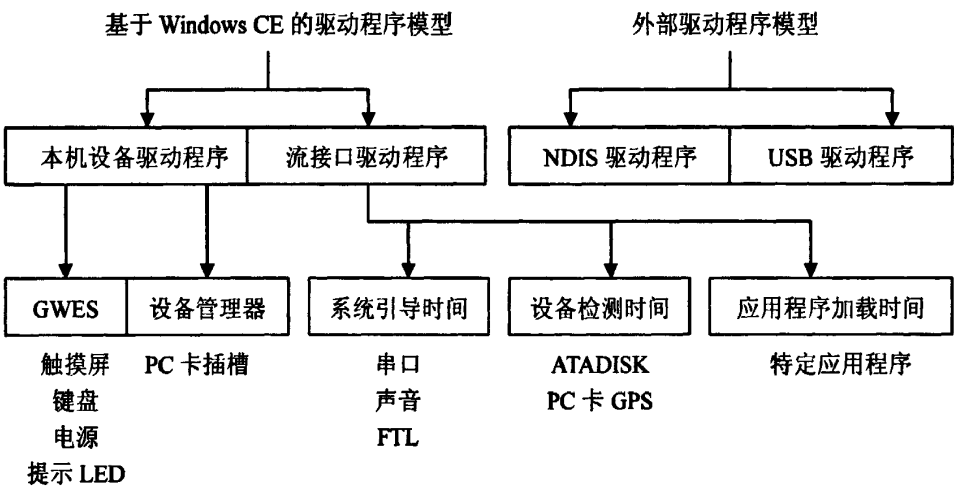


图 5-1 各种常用驱动对应的驱动程序模型

如果我们将 WinCE 系统移植到目标平台上，那么在这个平台上，必须为已建

立的设备提供驱动程序，例如键盘、显示器，这些类型的设备对操作系统都有一定的接口。因为这些接口是专门用于 WinCE 的，所以这类设备的驱动程序称作本机设备驱动程序^[61]。

流接口驱动程序是一般类型的设备驱动程序。流接口驱动程序表现为在用户一级的动态链接库 DLL，用来实现一组固定的函数称为流接口函数，这些流接口函数使得应用程序可以通过文件系统访问这些驱动程序^[62]。

5.1.2 Windows CE 5.0 驱动程序的系统结构

无论使用基于 WinCE 的哪种驱动程序，其提供的样本驱动程序都可分为两种类型：单片驱动程序和分层驱动程序^[63]。

微软提供的许多驱动程序都是采用了分层的结构，如图 5-2 所示。

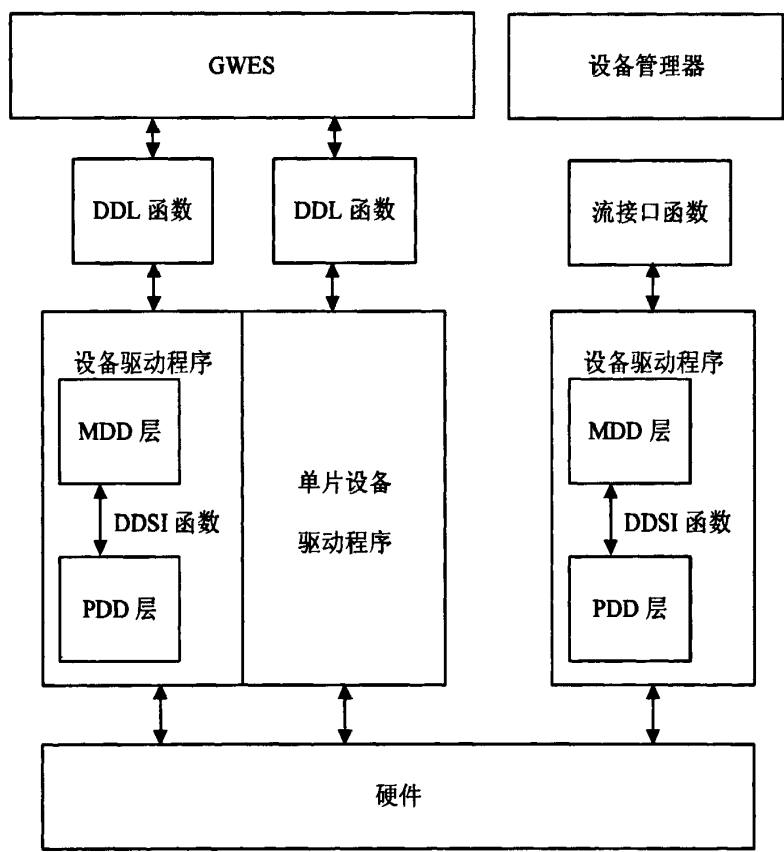


图 5-2 单片驱动和分层驱动

5.2 流接口驱动程序编写方法

流接口驱动程序是以动态链接库的形式存在的，具有固定的入口点函数，由设备管理器统一加载、管理和卸载^[64]。它的主要任务是把外设的使用传递给应用程序。图 5-3 说明了流接口驱动程序的工作结构。

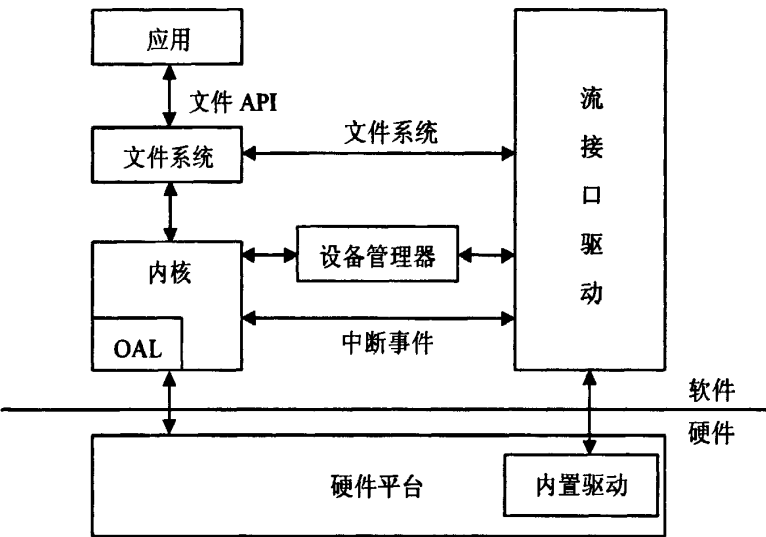


图 5-3 流接口驱动程序与操作系统和硬件的关系

对于每个流接口必须使用一组标准的函数，这些函数通常叫做流接口驱动程序的 dll 接口^[65]，如表 5-1 所示。

表 5-1 流接口驱动程序要实现的入口点函数及其功能

函数名	功能
CIS_Init	Device.exe 初始化图像采集模块时调用该函数
CIS_Deinit	Device.exe 卸载图像传感器驱动时调用该函数
CIS_IOControl	应用程序开发利用 DeviceControl()函数来调用该函数
CIS_Open	打开该驱动时，应用程序用 CreateFile()函数来调用该函数
CIS_Close	关闭该驱动时，应用程序用 CloseHandler()函数来调用该函数
CIS_Read	图像传感器驱动打开后，应用程序通过 ReadFile()函数调用该函数
CIS_Write	图像传感器驱动打开后，应用程序通过 WriteFile()函数来调用该函数
CIS_Seek	对设备数据指针进行操作时，应用程序通过 SetFilePointer()函数调用该函数
CIS_PowerUp	重启系统时调用该函数
CIS_PowerDown	挂起系统前调用该函数

5.3 OV9650 传感器流接口驱动设计

寄存器配置是驱动设计中的非常重要的内容,要使驱动正常工作,就必须对寄存器进行正确的配置。

驱动中涉及到很多寄存器的配置。S3C2440 芯片中的各寄存器组具有连续的物理地址,在 <S3C2440.h>头文件中以结构体形式给予了定义。经虚拟地址映射后,各寄存器组仍然具有连续地址,寄存器的访问可以采取两种方法。

(一) 使用变量访问其结构体成员的方法。

(二) 使用直接访问地址空间的方法^[66]。

5.3.1 流接口函数的实现

5.3.1.1 初始化函数 CIS_Init

编写流接口的驱动程序,实际上就是编写流接口函数。OV9650 初始化函数大体上包括三个操作步骤^[67]:

第一步:为寄存器分配虚拟地址

在 WinCE 系统中,若想要访问硬件的物理地址,就必须首先将其物理地址空间映射到虚拟地址空间。

本实验中,映射虚拟地址时首先使用 Virtual Alloc 在应用程序的地址空间中自动分配一块虚拟地址空间,用到的主要控制寄存器如下:

1. GPIO 控制寄存器 IOPreg;
2. CAMIF 控制寄存器 CAMreg;
3. 中断控制寄存器 INTreg;
4. 时钟电源管理特殊寄存器 CLKPWRreg;
5. IIC 控制寄存器 IICreg。

其次使用 VirtualCopy 来将硬件设备寄存器的物理地址映射成虚拟地址。

第二步:初始化 CAMIF 和图像传感器

可分为如图 5-4 所示七个步骤。

(一) 初始化 CAMIF 接口

CamGpioInit()函数首先配置 GPJCON 控制寄存器的 GPJ0—GPJ12 为 IO 口, s2440IOP->rGPJCON=0x2aaaaaa。该组寄存器就可以配置成 CAMIF 功能引脚了,其次将 CAMIF 初始化为 0,即 s2440IOP->rGPJDAT=0,最后使能上拉 s2440IOP->rGPJUP=0。

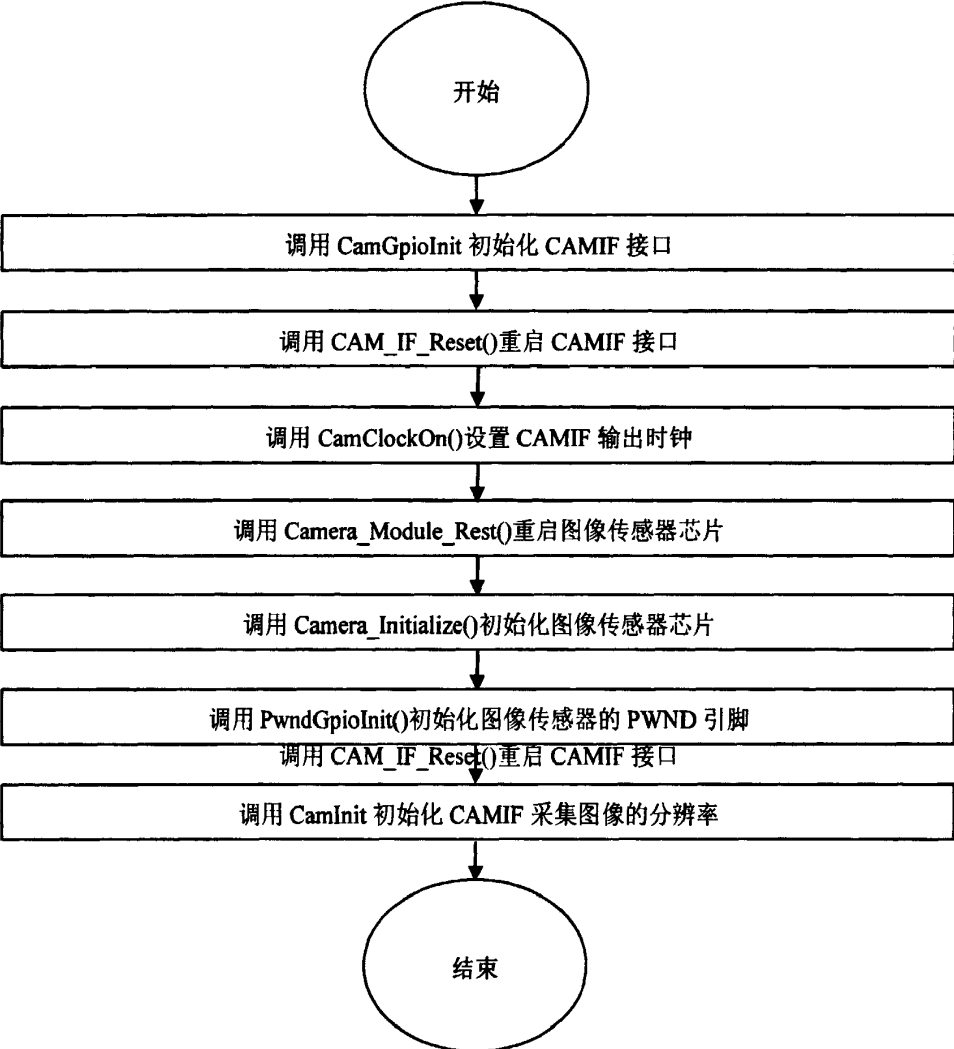


图 5-4 图像采集设备的初始化过程

(二) 重启 CAMIF

调用 CAM_IF_Reset()函数，部分代码如下：

```
s2440IOP->rGPJDAT|=1<<12;
s2440IOP->rGPJDAT&=~(1<<12);    //即 GPJ12:CAMRESET=0
```

(三) 调用 CamClockOn()设置 CAMIF 输出时钟，部分代码如下：

```
s2440PWR->rCLKCON|=(1<<19);
s2440PWR->rCAMDIVN=(s2440PWR->rCAMDIVN&~(0x1f)|(1<<4)|(divide&0
xf);//CAMCLK is divided
```

(四) 重启图像传感器

Camera_Module_Reset()函数将 CAMIF 全局控制寄存器 rCIGCTRL 的第 30 位设置为 1，等待一段时间，然后再将其置为 0，再等待，摄像头模块初始化，代

码如下：

```
s2440CAM->rCIGCTRL|= (1<<30);
s2440CAM->rCIGCTRL&=~(1<<30);
```

（五）初始化图像传感器

初始化图像传感器流接口函数 CIS_Init 时，OV9650 摄像头中的两个引脚 SDA 和时钟线 SCL 分别接 GPE14 和 GPE15 这两个 I/O 端口，并且把 SCL 宏定义为 14，SDA 宏定义为 15。

这样通过宏定义函数 MAKE_HIGH(x)和 MAKE_LOW(x)用于将 GPEDAT 寄存器的 X 位分别设为 1 和 0，使 X 位对应接口的状态分别为高电平和低电平。

IIC 总线产生起始条件、结束条件和进行数据传输的线路时序如图 5-5 所示。

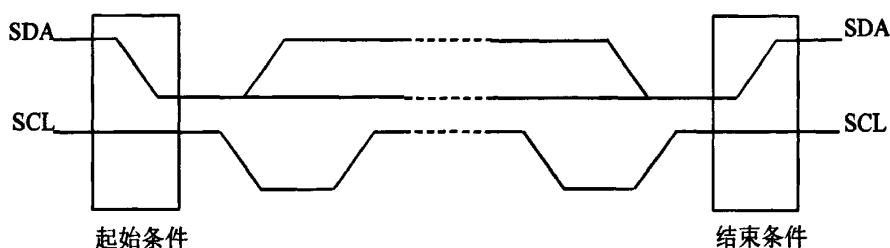


图 5-5 IIC 总线起始和结束条件

这样我们可以编写函数，起始条件函数 `sccb_start`，结束条件函数 `sccb_stop`，发送 1 比特数据函数 `sccb_write_bit`。

0xB1600044 为 GPEDAT 寄存器的虚拟地址。等待时间根据不同的图像传感器芯片长短可能会有所不同，在这里我们设置其为 600ns。

这样函数 `sccb_write_bit` 实现发送 1 比特数据的方法如下：

//延时函数

```
void Delay(int time)
```

```
{
```

```
    volatile int i,j=0;
```

```
    //s2440.h 中 #define S2440FCLK (405000000)
```

```
    volatile static int loop=S2440FCLK/100000;
```

```
    for(;time>0;time--)
```

```
    for(i=0;i<loop;i++)
```

```
    {
```

```
        j++;
```

```
    }
```



```

}
//起始条件:
void_inline sccb_start(void)
{
    *(volatile unsigned*)0xB1600044|=1<<14;    //置 SCL 为高
    *(volatile unsigned*)0xB1600044|=1<<15;    //置 SDA 为高
    Delay(2);    //延时等待
    *(volatile unsigned*)0xB1600044&=~(1<<15);    //置 SDA 为低
    Delay(2);    //延时等待
    *(volatile unsigned*)0xB1600044&=~(1<<14);    //置 SCL 为低
    Delay(2);
}
//结束条件
void_inline sccb_stop(void)
{
    *(volatile unsigned*)0xB1600044&=~(1<<15);    //置 SDA 为低
    Delay(2);
    *(volatile unsigned*)0xB1600044|=1<<14;    //置 SCL 为高
    Delay(2);
    *(volatile unsigned*)0xB1600044|=1<<15;    //置 SDA 为高
    Delay(2);
}
数据传输的相关代码实现:
//传输 1 bit 数据:
void_inline sccb_write_bit(unsigned char bit)
{
    if(bit) *(volatile unsigned*)0xB1600044|=1<<15;
    else *(volatile unsigned*)0xB1600044&=~(1<<15);
    Delay(2);
    *(volatile unsigned*)0xB1600044|=1<<14;
    Delay(2);
    *(volatile unsigned*)0xB1600044&=~(1<<14);
    Delay(2);
}
//传输 1 byte 数据

```

```
void inline sccb_writechar(unsigned char data)
{
    for(int i=0;i<8;i++)
    {
        sccb_write_bit(data&0x80);
        data<<=1;
    }
    sccb_write_bit(1);
}
```

驱动程序通过 IIC 向 OV9650 发送寄存器配置数据的主要过程包括：产生起始条件、发送从设备地址、发送数据、产生停止条件^[68]。设计流程如图 5-6 所示。

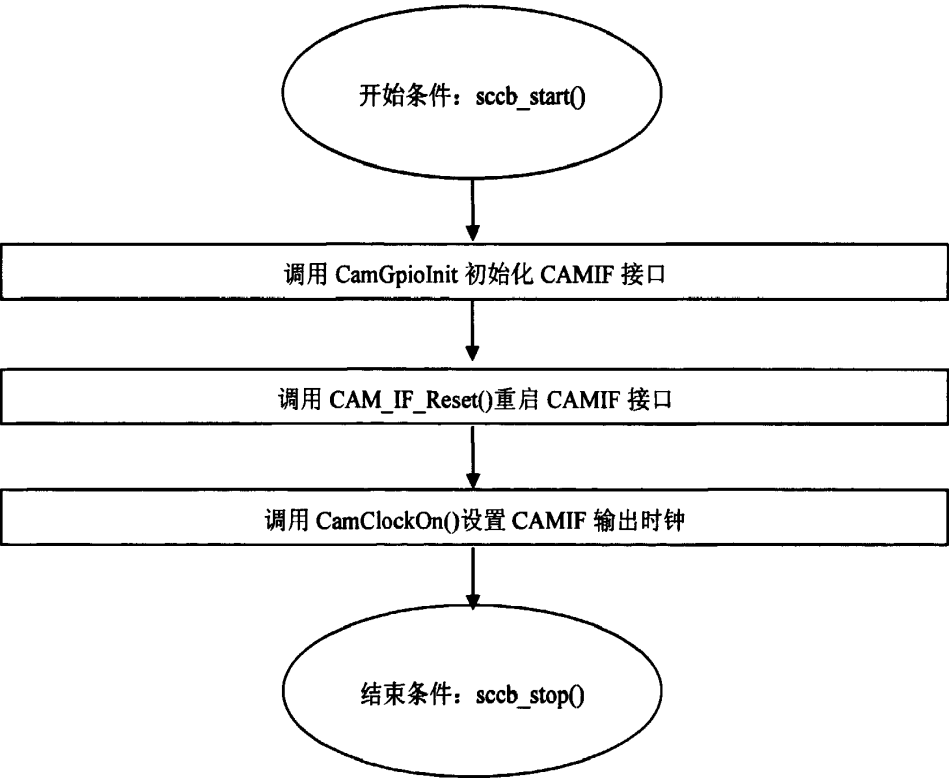


图 5-6 OV9650 寄存器的数据写入流程

```
//传输 1 byte 有效数据
static void sccb_sendbyte(unsigned char subaddr, unsigned char data)
//写入寄存器配置数据函数
{
```

```

    sccb_start();
    sccb_writechar(OV9650_SCCB_ID); //发送设备的写入地址
    sccb_writechar(subaddr); //发送具体地址
    sccb_Writechar(data); //发送配置数据
    sccb_stop();
    Delay(8);
}

```

对摄像头寄存器进行配置时，首先定义一个结构体数组 `struct OV9650{int subaddr; int value;}`，用于对所有寄存器信息进行配置。其中 `subaddr` 表示 OV9650 内部寄存器地址，`value` 表示寄存器中的数值。最后设计一个循环对所有寄存器进行配置：

```

for(int i=0;i<OV9650_REGS;i++)
{
    if(ov9650_reg[i].subaddr==CHIP_DELAY)
        Delay(ov9650_reg[i].value);
    else
        sccb_sendbyte(ov9650_reg[i].subaddr & 0xff, ov9650_reg[i].value & 0xff);
}

```

（六）初始化图像传感器的 PWND 引脚

在完成 OV9650 内部寄存器配置工作之后，我们需要设置 PWND 管脚为 0。将 PWND 管脚连接在 S3C2440 的 GPG12 管脚上，然后设置 GPG12 管脚为输出功能，并将其设置输出为低电平：

```

s2440IOP->rGPCON&=~(0x3<<24);
s2440IOP->rGPCON|=(0x1<<24);
s2440IOP->rGPGDAT&=~(1<<24);

```

（七）初始化图像分辨率

如果需要设置图像分辨率，这是我们用到函数 `CamInit`。它是这样定义的：
`Void CamInit(U32 CoDstWidth , U32 CoDstHeight , U32 PrDstWidth , U32 PrDstHeight, U32 WinHorOffset, U32 WinVerOffset, U32 CoFrameBuffer, U32 PrFrameBuffer)`

第三步：初始化中断服务线程

流程如图 5-7 所示。

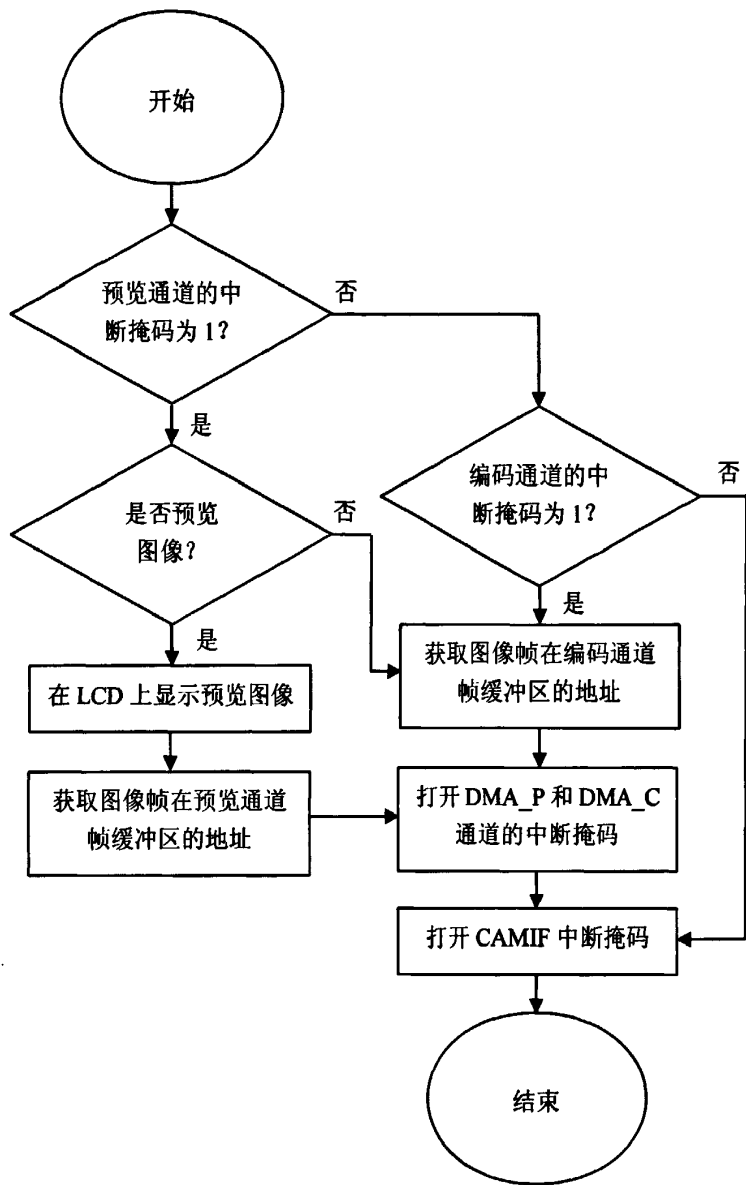


图 5-7 图像捕获中断服务线程的流程

首先我们创建一个全局变量 `CameraEvent`，使用到的函数为 `CreateEvent`，然后我们将其与摄像头逻辑中断号 `SYSINTR_CAM` 相关联。函数代码如下：

然后需要创建图像捕获线程 `CameraCaptureThread`，并调用函数 `WaitForSingleObject(CameraEvent, INFINITE)` 等待 CAMIF 中断事件的发生。

如果有中断产生，首先读取中断掩码寄存器 `INTSUBMSK` 中 `DMA_C`、`DMA_P` 的中断掩码值，判断中断请求是由 CAMIF 单元的哪条通道发出。如果打开的通道是 P 通道则预览，如果是 C 通道则拷贝。

//使能 C 通道中断服务掩码：

```

s2440INT->rINTMSK &=~BIT_CAM;
s2440INT->rINTSUBMSK &=~(BIT_SUB_CAM_C);
//使能 P 通道中断服务掩码:
s2440INT->rINTMSK &=~BIT_CAM;
s2440INT->rINTSUBMSK &=~(BIT_SUB_CAM_P);
//同时使能 P 通道和 C 通道中断服务掩码:
s2440INT->rINTMSK &=~BIT_CAM;
s2440INT->rINTSUBMSK &=~(BIT_SUB_CAM_C|BIT_SUB_CAM_C);

```

5.3.1.2 去初始化函数 CIS_Deinit

卸载驱动程序时调用此函数进行去初始化操作，一般用来释放驱动所占用的资源、停止 IST 等。

本驱动中 CIS_Deinit 进行的主要操作如下：

```

s2440INT->rINTMSK|=BIT_CAM;
s2440INT->rINTSUBMSK|=(BIT_SUB_CAM_P|BIT_SUB_CAM_C);
Camif_Capture(CAPTURE_OFF, CAPTURE_OFF);
CloseHandle(CameraThread);
VirtualFree((void*)s2440IOP, sizeof(IOPreg), MEM_RELEASE);
VirtualFree((void*)s2440CAM, sizeof(CAMreg), MEM_RELEASE);
VirtualFree((void*)s2440INT, sizeof(INTreg), MEM_RELEASE);
VirtualFree((void*)s2440PWR, sizeof(CLKPWRreg), MEM_RELEASE);
VirtualFree((void*)s2440IIC, sizeof(IICreg), MEM_RELEASE);

```

5.3.1.3 I/O 控制函数 CIS_IOControl

设置 CAMIF 图像分辨率的代码如下：

```

//在 camera.h 中定义
#define CAM_IOCTL_IMAGE_SIZE 0x521
//在 camera.cpp 中进行功能实现
case CAM_IOCTL_IMAGE_SIZE:
    if(*pBufOut==1) image_size=1;
    else if(*pBufOut==2) image_size=2;
//先禁止图像通道中断服务掩码并关闭通道
s2440INT->rINTMSK|=BIT_CAM;
s2440INT->rINTSUBMSK|=(BIT_SUB_CAM_P|BIT_SUB_CAM_C);
Camif_Capture(CAPTURE_OFF, CAPTURE_OFF);

```

```

//调用 CamInit 设置图像分辨率
if(image_size==1)
    CamInit(QCIF_XSIZE, QCIF_YSIZE, QCIF_XSIZE, QCIF_YSIZE, 112, 20,
    COPIFRAMEBUFFER_B, COPIFRAMEBUFFER_A);
else if(imag_size==2)
    CamInit(CIF_XSIZE, CIF_YSIZE, QCIF_XSIZE, QCIF_YSIZE, 112, 20,
    COPIFRAMEBUFFER_B, COPIFRAMEBUFFER_A);

```

5.3.1.4 打开设备函数 CIS_Open

应用程序中调用 CreateFile 时, 文件系统会调用 CIS_Open 函数。打开已注册的设备文件。所以可在此函数中对一些参数配置默认初始化值, 也可以依据设计者的意图进行其他的初始化操作。

5.3.1.5 关闭设备函数 CIS_Close

应用程序调用 CloseHandle 来关闭正在使用的图像传感器驱动, 可依据需要在此函数中进行关闭驱动前的相关动作。

5.3.1.6 关闭设备电源函数 CISPowerDown

要想停止对设备供电时应用程序就会调用这个函数, 可以不再需要采集图像时将设备关闭, 以节省用电和提高设备使用寿命。

5.3.1.7 打开设备电源函数 CIS_PowerUp

应用程序通过 I/O 控制代码 IOCTL_POWER_SET 来调用 CIS_PowerDown 和 CIS_PowerUP 这两个流接口函数。

CIS_PowerUp 函数的操作分为五部分:

打开 CAMIF 的电源; 初始化 CAMIF 接口; 重启 CAMIF 接口; 设置 CMIF 输出时钟; 设置 CAMIF 采集图像的分辨率。

5.3.2 将驱动加入操作系统

图像传感器驱动程序动态链接库 camera.dll 是由五个文件编译而成的, 通过对这五个文件的编译, 可以生成驱动程序的动态链接库文件 camera.dll^[69]。这五个文件分别是 camera.h、camif.h、OV9650.h、camera.cpp、iic.cpp。

接下来我们要做的工作是动态加载、卸载设备驱动。可以采取以下三种形式加载流接口驱动程序:

(一) 系统启动时, 会加载注册表中已经列出的流接口驱动程序;

(二)设备管理器程序自动检测外围设备与基于 WinCE 系统平台的连接时加载;

(三) 可以通过调用系统提供的函数 `ActivateDeviceEx()` 手动加载驱动程序。

下面我们采用第一种形式加载,使自己所编写的驱动在系统启动时自动载入。

首先编辑 DRIVERS 目录下的 `dirs` 文件,用文本编辑器打开 `dirs` 文件。找到“`DIRS=`”等式,插入 `DIRS=camera\`之后保存该文件。则 PB 在编译操作系统时,会自动编译 `camera` 驱动程序。

其次打开 `Platform.bib` 文件,在该文件的最后,FILES 之前加入如下代码:

```
camera.dll  $(_FLATRELEASEDIR)\camera.dll      NK  SH
```

SH 指明该文件的属性;

S 指明为系统文件;

H 指明为隐藏文件。

最后打开 `Platform.reg` 文件。在其最后加入如下内容,这样就可以在操作系统启动时,自动加载该驱动了。

```
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\Camera]
```

```
"Prefix"="CIS"
```

```
"Dll"="camera.dll"
```

```
"Index"=dword:1
```

```
"Order"=dword:1
```

其中: `Prefix` 表示图像传感器流接口驱动程序的前缀;

`Dll` 表示驱动程序的 `dll` 文件名称;

`Order` 表示驱动加载的顺序;

`Index` 表示同类设备驱动的实例被加载时来相互区分的索引号。

`Order` 值为大于等于 0 的整数,当系统中存在多个相同前缀名驱动时,其 `Order` 值越小驱动越被先加载^[70]。

做好以上工作后,重新下载新的操作系统映像。并将其烧写到开发板当中。如在超级终端中看到启动信息,说明该驱动已被加载了。

5.4 步进电机驱动程序的设计

与摄像头驱动程序设计相似。步进电机驱动程序的设计首先需要我们确定其文件前缀为 `XYM(XYstepping Motor)`,其次用文本编辑器简历五个文本文件,文件名分别为: `XYsteppingMotor.c`、`XYsteppinMotor.h`、`XYsteppingMotor.def`、`makefile`

和 source。完成了这些文本的编辑之后，就可以用 PB5 编译本驱动程序了。其过程与摄像头驱动程序一样。故不再重复。步进电机流接口驱动函数的实现流程图如图 5-8 所示。

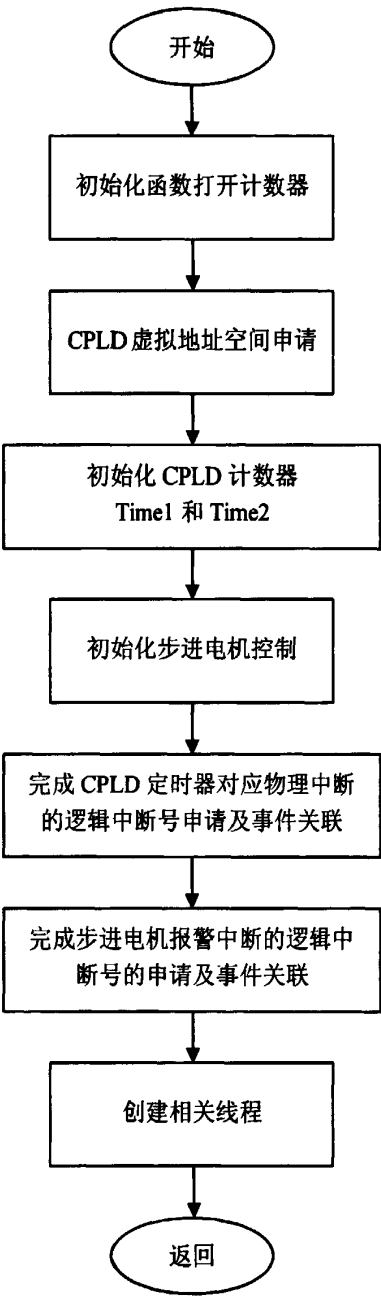


图 5-8 步进电机驱动初始化流程图

5.4.1 函数的初始化

CPLD 有两个主要的功能：一是利用定时器单元产生 PWM 波为步进电机驱动器提供控制脉冲；二是利用其中的计数器单元对光电编码器返回值进行计数。CPLD 对步进电机提供的定时器和计数器资源分配如表 5-1 所示^[71]。

表 5-1 CPLD 对步进电机提供的定时器和计数器资源分配情况

路数	Timer/Counter	名称	功能
1	CPLD T1	CW_X	提供 X 轴电机正向步进脉冲
		CCW_X	提供 X 轴电机反向步进脉冲
2	CPLD T2	CW_Y	提供 Y 轴电机正向步进脉冲
		CCW_Y	提供 Y 轴电机反向步进脉冲
3	CPLD C1	COUNTA	光电编码器 1 A 相计数
4	CPLD C2	COUNTZ	光电编码器 1 Z 相计数
5	CPLD C3	COUNTA	光电编码器 2 A 相计数
6	CPLD C4	COUNTZ	光电编码器 2 Z 相计数

在本系统中 CPLD 主要是负责接受 CPU 传来的数据，然后解析，最后发出控制电机的信号。CPLD 产生的 PWM 控制脉冲，可以精确灵活控制输出脉冲的频率和占空比。

```
为访问 CPLD 的控制寄存器，须为其分配虚拟地址空间。其代码如下：
v_CPLDRegs=
(volatile    CPLD_REG*)VirtualAlloc(0,sizeof(CPLD_REG),MEM_RESERVE,
PAGE_NOACCESS);           //申请空间
if（申请空间失败）{返回错误；}
else
{
    VirtualCopy((PVOID)v_CPLDRegs,(PVOID)(S3C2440A_BASE_REG_PA_
CPLD>>8),sizeof(CPLD_REG),PAGE_PHYSICAL|PAGE_READWRITE|PAGE_NO
CACHE));           //映射空间
    if（映射空间失败）
    {
        VirtualFree((PVOID)v_CPLDRegs,0,MEM_RELEASE);
        //要释放申请的空间
        v_CPLDRegs=NULL;
```

```

    }
}

```

初始化 CPLD 的 Time1 和 Time2, Time1 和 Time2 根据我们的要求定时产生 PWM 波, 用于分别对 X, Y 两相步进电机提供驱动脉冲。

通过 GPIO 的初始化操作, 初始化步进电机控制总线, 使采光板的角度处于预备状态。并且关闭步进电机, 防止步进电机转动。电机使能和禁止函数如下:

```

#define X_PD_EN_ON(v_IOPregs->GPJDAT|=(0x01<<3));// X 相电机使能
#define X_PD_EN_OFF(v_IOPregs->GPJDAT&=(0x01<<3));// X 相电机禁止
#define Y_PD_EN_ON(v_IOPregs->GPJDAT|=(0x01<<4));// Y 相电机使能
#define Y_PD_EN_OFF(v_IOPregs->GPJDAT&=(0x01<<4));// Y 相电机禁止

```

5.4.2 中断的申请与处理

在 Windows CE.NET 中, 中断被分为核心的 ISR 和用户线程 IST 两个阶段来完成。当一个中断产生时, 内核 ISR 将禁止所有同级别或者优先级较低的中断。这时剩下的工作就留给 IST 来完成了^[72]。

本驱动引入的中断主要是为了让 CPU 知道电机操作完成, 以便对步进电机进一步操作。

为 CPLD 的物理中断源注册逻辑中断, 通过 KernelIOControl() 发送 IOCTL_HAL_REQUEST_SYSINTR 命令获得与 g_Time1_Irq 对应的逻辑中断号, 并与事件 Event_MotorX 相关联, 程序代码如下:

```

KernelIOControl(IOCTL_HAL_REQUEST_SYSINTR,&g_Time1_Irq,sizeof(UINT32),& Event_MotorX,sizeof(UINT32),NULL)

```

```

InterruptInitialize(Event_MotorX,local_WaitEvent_MX [],0,0)

```

同理可以获得 Y 相步进电机和光电编码器的逻辑中断源的申请以及关联事件。

在 XYM_Init() 函数中通过函数 CreateThread() 建立中断处理线程, 函数 WaitForMultipleObjects 将线程挂起, 等待即将发生的事件, 若有事件发生, 首先判断是哪个事件发生, 然后作相应处理。其函数结构如图 5-9 所示。

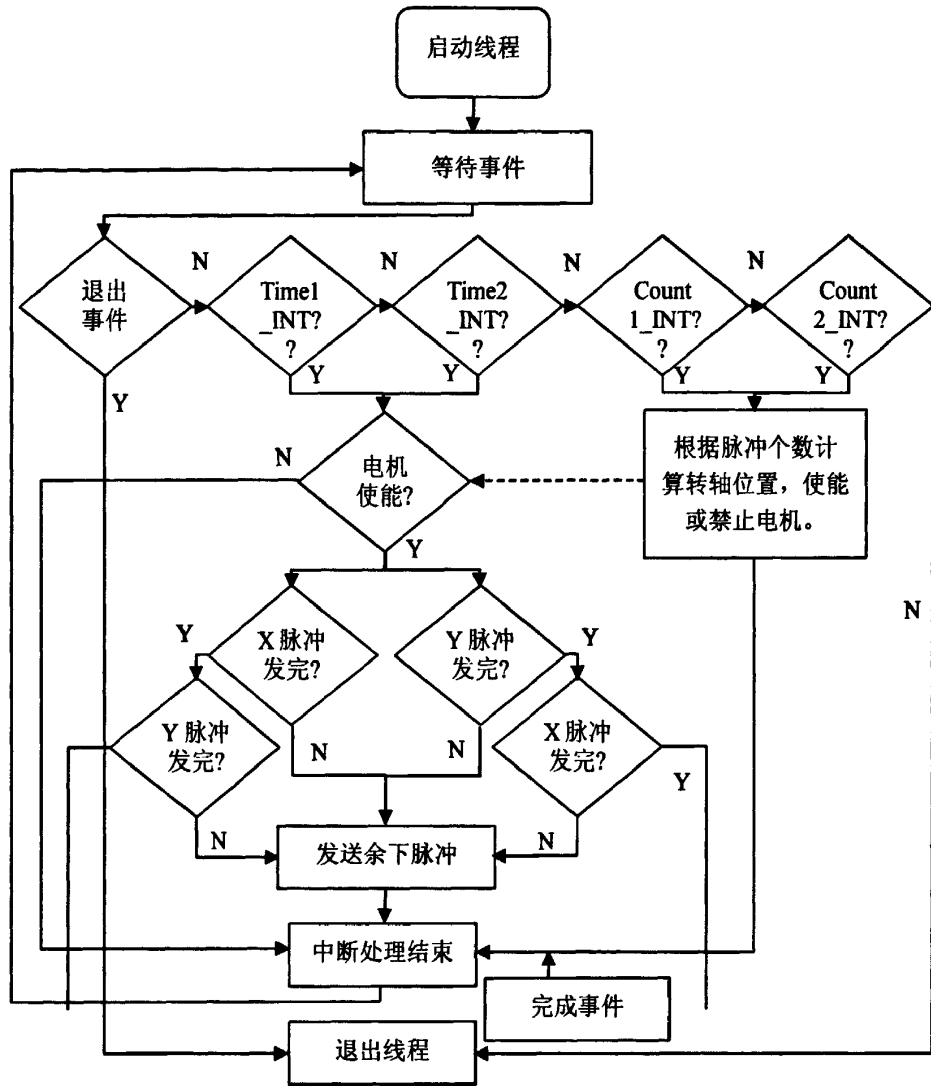


图 5-9 中断处理线程函数

5.5 本章小结

本章首先介绍了 WinCE 操作系统驱动开发的基础知识, 重点讲述了摄像头驱动程序的开发以及步进电机驱动程序。经过调试证明本实验软硬件设计合理, 运行稳定, 可以实现对太阳的实时追踪。

第六章 总结与展望

太阳能资源的充分利用,有利于整个社会的环境保护。而太阳能又是一种取之不尽用之不竭、清洁无污染的能源。因此研发出一种能提高太阳能利用率的设备,无论是从能源的角度,还是商业发展的角度都将成为科学工作者要解决的问题。

本课题的研究内容:

(一) 查阅了相关资料,分析了国内外在智能追光系统方面的成就,以及用到的技术情况。最终确定本课题采用嵌入式技术开发此装置。

(二) 本课题主要在于设计一种能实时随太阳角度变化而变化的装置,因此如何感知太阳角度的变化至关重要。本文主要工作重点首先放在了如何确定任意时刻地球和太阳的相对方位,在总结前人的技术基础上,确定了采用图像处理的方法来测量太阳高度角和方向角的变化。

(三) 设计了从智能追光系统的原理、电机控制方案等硬件部分,到操作系统的定制和移植、驱动程序的开发等软件模块,还设计了机械结构部分。实验证明能提高太阳能利用率 30%以上。

虽然本实验从理论上论述了实验的可行性,并且给出了完整的设计方案,实验证明确实能大大提高太阳能的利用率,但是仍然存在很多不足,表现在以下几个方面:

(一) 系统功能还需完善。因为本系统大多需要野外作业,而本系统并没有增加太阳能蓄电池的供电设备,这样势必造成供电困难的问题。所以还需要解决在室外条件下给系统供电的问题。特别是步进电机的供电,因为电机需要带动电池板的转动,所需的功率很大,需要今后作进一步改善。

(二) 智能追光系统普遍存在成本高的问题,主要表现在双轴追踪系统需要两个电机,这是增加成本的主要原因。所以从商业角度来看,要想把此装置投入市场还有待改进,这也是今后的主要研究内容。

参考文献

- [1] HENRIK L. Renewable energy strategies for sustainable development [J]. Energy, 2007, 18: 202-205.
- [2] 李申生. 物理学与太阳能[M]. 广西: 广西教育出版社, 1999, 126-174.
- [3] ARDEHALI M M, SHAHERSTANI M, CHARLES C. Energy simulation of solar assisted absorption system and examination of clearness index effects on auxiliary heating[J]. Energy Conversion and Management, 2007, 18: 312-318.
- [4] 戴闻. 太阳能利用前景光明[J]. 物理, 2003, 12: 425-432.
- [5] 徐机玲, 蔡玉高. 太阳能利用新突破[J]. 瞭望, 2004, 6: 213-217.
- [6] KALOGIROU S A. Industrial application of PV/T solar energy systems[J]. Applied Thermal Engineering, 2007, 2: 124-131.
- [7] 赵丽伟. 太阳自动追踪系统的研究[D]. 吉林: 吉林大学, 2007, 12-15.
- [8] 毛桂生. 太阳能电池板自动追踪系统的研究[D]. 广州: 华南理工大学理学院, 2010, 412-415.
- [9] 赵媛, 赵慧. 我国太阳能资源及其开发利用[J]. 经济地理, 1998, 1: 6-14.
- [10] 余海. 太阳能利用综述及提高其利用率的途径[J]. 能源研究与利用, 2004, 2: 123-126.
- [11] 罗运俊, 何梓年, 王长贵. 太阳能利用技术[M]. 北京: 化学工业出版社, 2005, 14-18.
- [12] 练亚纯. 太阳能的利用[M]. 北京: 北京人民出版社, 1975, 15-18.
- [13] 岑幻霞. 太阳能热利用[M]. 北京: 清华大学出版社, 1997, 21-26.
- [14] 言惠. 太阳能—21 世纪的能源[J]. 上海大中型电机, 2004, 3: 321-324.
- [15] 姚伟. 太阳能利用与可持续发展[J]. 中国能源, 2005, 3: 233-236.
- [16] 王炳忠. 太阳能—未来能源之星[M]. 北京: 高等教育出版社, 1990, 567-569.
- [17] 郭廷玮. 太阳能的利用[M]. 上海: 科学技术文献出版社, 1984, 23-25.
- [18] CROCKETT R G, NEWBOROUGH M. Electrolyser-based energy management: a means for optimising the exploitation of variable renewable-energy resources in stand-alone applications [J]. Solar Energy, 1997 4: 345-348.
- [19] 樊峰鸣, 马良涛. 单轴太阳能跟踪系统的研究[J]. 河南城建高等专科学校校报, 2000, 12: 326-332.
- [20] 郑小年, 黄巧燕. 太阳追踪方法及应用[J]. 能源技术, 2003, 1: 14-19.

- [21] 孙茵茵, 鲍剑斌, 王凡. 太阳自动跟踪系统的研究[J]. 机械设计与制造, 2005, 2: 146-150.
- [22] 高钟毓, 王永梁. 机电控制工程[M]. 北京: 清华大学出版社, 1994, 34-38.
- [23] 陆利生. 单轴太阳能自动跟踪器液压传动系统的设计[J]. 新能源, 1994, 2: 39-43.
- [24] Edwards B P. Computer based sun following system[J]. Solar Energy, 1978, 2: 45-50.
- [25] 陈文智. 嵌入式系统开发原理与实践[M]. 北京: 清华大学出版社, 2005, 12-16.
- [26] 何宗键. WinCE 嵌入式系统[M]. 北京: 北京航空航天大学出版社, 2006, 45-54.
- [27] 俞建新, 王健, 宋健建. 嵌入式系统基础教程[M]. 北京: 机械工业出版社, 2008, 34-38.
- [28] 周立功. ARM 嵌入式系统基础教程[M]. 北京: 北京航空航天大学出版社, 2008, 68-73.
- [29] KAMAL R. Embedded Systems: Architecture, Programming and Design[M]. 北京: 清华大学出版社, 2010, 34-39.
- [30] 周航慈. 嵌入式系统软件设计基础丛书[M]. 北京: 北京航空航天大学出版社, 2011, 87-93.
- [31] LEWIN A R. So You Wanna Be an Embedded Engineer: The Guide to Embedded Engineering, from Consultancy to the Corporate Ladder[M]. 北京: 人民邮电出版社, 2011, 56-67.
- [32] 佘肇乾. 嵌入式系统硬件体系设计[M]. 北京: 北京航空航天大学出版社, 2007, 67-79.
- [33] 邱铁. ARM 嵌入式系统结构与编程[M]. 北京: 清华大学出版社, 2009, 145-159.
- [34] 孙少林, 马志强, 汤伟. 灰度图像二值化算法研究[J]. 价值工程, 2010, 2: 56-62.
- [35] 卓晴, 黄开胜, 邵贝贝. 学做智能车[M]. 北京: 北京航空航天大学出版社, 2007, 567-572.
- [36] 梁勇, 梁维铭. 太阳能任意方位跟踪系统设计[J]. 电测与仪表, 2008, 3: 111-114.
- [37] 宋开峰. 新型太阳跟踪装置机构的研究[D]. 天津: 河北工业大学, 2004, 12-16.
- [38] 陈静静. 基于 ARM9 的疲劳驾驶检测系统的设计[D]. 天津: 天津工业大学理学院, 2010, 14-18.
- [39] 梁迪. GPS 在太阳能追光系统中的应用[D]. 北京: 北京交通大学电子信息工程学院, 2009, 78-82.
- [40] 任超. 太阳自动跟踪装置控制系统的设计[D]. 武汉: 武汉理工大学机电工程学院, 2007, 54-59.

- [41]王典洪,熊月华,梁娟.基于 WinCE 的步进电机驱动程序设计[J].嵌入式操作系统应用,2007,2:212-217.
- [42]曲明波,白瑞林,刘洋.WinCE 下高速工业绣花机主轴系统驱动设计[J].计算机工程与应用,2009,3:343-347.
- [43]黎燕霞.基于 S3C2440 的视频采集驱动设计[J].仪器仪表用户,2009,1:12-16.
- [44]冉荣.模型小车路径识别 [J].仪器仪表用户,2009,1:34-35.
- [45]朱世佳.基于嵌入式的太阳能智能追光系统设计[D].北京:北京交通大学电子信息工程学院,2009,21-24.
- [46]吴静.太阳自动跟踪系统的研究[D].重庆:重庆大学,2008,13-16.
- [47]荣锐.基于 ARM9 的视频传输系统研究[D].湖北:武汉科技大学,2009,14-19.
- [48]邵震宇.基于 WinCE 操作系统的 GPS 接收机的研究[D].天津:天津工业大学电信学院,2009,23-25.
- [49]郑屹帆.基于 SEP3203 微处理器的 ASIX OS 操作系统内核的研究与优化[D].南京:东南大学,2007,38-42.
- [50]田力.实时嵌入式系统 VxWorks 安全机制的研究与实现[D].南京:南京航空航天大学研究生院,2009,22-27.
- [51]宋俊龙.基于 VxWorks 下 Mesh 网络设备中桥接模块的研究与开发[D].哈尔滨:哈尔滨工业大学研究生院,2009,67-69.
- [52]赵修敏.基于嵌入式系统的二维码识别[D].天津:天津工业大学理学院,2009,36-45.
- [53]孙亚军.基于 ARM 的嵌入式 Linux 平台的研究与实现[D].沈阳:东北大学,2009,34-37.
- [54]唐磊.基于 S3C2440 ARM Linux 图像采集系统的研究[D].武汉:武汉科技大学,2008,12-17.
- [55]夏天.基于 Windows CE 的嵌入式工控组态软件数据库系统的设计与实现[D].武汉:武汉科技大学,2006,56-59.
- [56]蔡卫明.ARM9_WinCE 嵌入式图像采集系统开发技术研究与实现[D].重庆:重庆理工大学电子信息及其自动化学院,2009,21-27.
- [57]张云.基于 ARM 和 WinCE 的电台可视化人及交互接口的设计与实现[D].长沙:国防科学技术大学,2009,18-23.
- [58]黄丹.基于 WinCE 的可编程智能控制对象的设计与实现[D].上海:上海交通大学,2007,5-12.
- [59]程磊.基于 ARM9 的车载多媒体中断的设计与实现[D].苏州:苏州大学,2010.
- [60]章蔚中.基于 ARM920T 嵌入式网络控制器的设计与实现[D].南昌:南昌大学,2009,3-7.

- [61] 张飞. 基于 WinCE 的大型触摸屏控制器的设计与实现[D]. 无锡: 江南大学, 2008, 23-28.
- [62] 屈英. 基于 Windows CE 的嵌入式视频监控终端的设计与实现[D]. 西安: 西北工业大学, 2003, 35-40.
- [63] 王丹丹. 基于磁弹效应法的索力检测技术研究与仪器设计[D]. 南京: 东南大学, 2010, 24-29.
- [64] 赵勃波. 基于 WinCE 的磨床 AEMS 监控系统的设计与实现[D]. 浙江: 浙江大学, 2010, 35-37.
- [65] 孟建民. 嵌入式指纹采集系统研究[D]. 南京: 南京航空航天大学, 2008, 21-25.
- [66] 李浩. 基于 ARM9 的视频监控系统的设计与开发[D]. 无锡: 江南大学, 2009, 64-69.
- [67] 刘军民. 基于 ARM 的嵌入式网络视频监控系统设计[D]. 大连: 大连理工大学, 2009, 12-16.
- [68] 刘万志. 基于 ARM9 的图像采集系统设计[D]. 北京: 北京交通大学, 2008, 21-26.
- [69] 周立功. ARM&WinCE 实验与实践[M]. 北京: 北京航空航天大学出版社, 2007, 18-21.
- [70] 施刚. 基于 ARM 和 Win CE 的教学实验平台研发[D]. 沈阳: 东北大学, 2008, 11-17.
- [71] 曲明波. 基于 WinCE 的高速工业绣花机运动控制的研究与应用[D]. 无锡: 江南大学, 2008, 32-36.
- [72] 白瑞林. 基于 WinCE5.0 的高速绣花机步进电机驱动[J]. 计算机工程, 2009, 3: 346-348.

发表论文和参加科研情况

- [1] ChAOWEI SHI, Yan CUI. System of Sunlight Tracking Automatically Based On S3C2440[J]. Proceedings of the 2011 2nd International Conference on Information Technology and Scientific Management, 2011, 03: 1486-1489.

致 谢

转眼间两年多的研究生生活很快就要过去了，但是在这里的每一分每一秒对知识的探索将是我一生最宝贵的财富。

在这即将离开母校之际，我想对我最敬爱的崔艳老师道一声“谢谢，老师您辛苦了！”因为在这两年多的时间里，崔老师曾给过我无数次的细心指导。本论文从开始确定课题到搜集资料，从软件模拟调试，到最终完成历时一年多时间。而每一个环节始终都留下了崔老师的点点心血。崔老师学识渊博、治学严谨给了我很大帮助。同时崔老师对学生负责任、敬业奉献的态度也成为我今后生活的楷模和榜样。

同时我还要谢谢理学院的所有老师，他们不仅在专业技能方面提供了很大帮助，而且在为人处事，对工作和生活积极热情方面都使我受益匪浅。感谢两位舍友以及所有同学们的帮助，是他们在课题陷入困境的时候陪我度过难关，最终给我的研究生生活画上了圆满的句号，也在我的生命历程中留下了美好的回忆。

感谢我的父母和亲人的关心，他们陪伴我一路走过了近二十年的学习生涯，正是他们的大力支持与鼓励才给了我前进的无穷动力。

感谢所有帮助我、支持我、关心我的人，我一定不会辜负大家对我的殷切期待，走好人生的每一步。

二零一一年十二月

