

电子系统设计实践报告

**智能语音出租车计价器**

**电气 学院 电气工程 系**

**电气2102邓朴达2216113167**

**电气2102苏春洋2211211963**

2024年5月

摘 要

**摘 要：**本实验旨在设计并实现一种智能语音出租车计价器的电子系统，以提高出租车行业的服务质量和效率。系统采用STM32F107VCT6微控制器单元作为核心，集成了语音输出、计价、显示和实时时钟等功能模块。通过控制启动和暂停进行里程计价，系统能够自动计算乘车费用并实时在ILI9341d LCD显示。经过实验验证，系统表现出良好的稳定性和准确性，具有广阔的应用前景。在数据采集方面，本文创新地设计了基于ARM Cortex-M3处理器的数据采集系统，可准确采集车轮转速并计算里程、总价和时速。同时，引入语音报价功能，为乘客提供了更加便捷的服务体验，并提供无线支付的选项，进一步提升了系统的实用性和竞争力。利用DS1302实时时钟功能，系统能够准确地记录时间信息，为计价和报价提供了时间基准，增强了系统的功能完整性和可靠性。

**关 键 词：** STM32F107VCT6；DS1302实时时钟；无线支付；语音报价；ILI9341LCD

目 录

[电子系统设计实践报告 1](#_Toc166873711)

[摘 要 2](#_Toc166873712)

[1 前言 4](#_Toc166873713)

[2 设计任务与需求分析 4](#_Toc166873714)

[3 工作模块及原理 5](#_Toc166873715)

[3.1 ADC及电位器 6](#_Toc166873716)

[3.2 LCD液晶显示 6](#_Toc166873717)

[3.2.1 ILI9341介绍 6](#_Toc166873718)

[3.2.2 ILI9341驱动原理 7](#_Toc166873719)

[3.3 DS1302 7](#_Toc166873720)

[3.3.1 ILI9341介绍 7](#_Toc166873721)

[3.3.2 DS1302驱动原理 8](#_Toc166873722)

[3.4 CN-TTS语音模块 9](#_Toc166873723)

[4 总体方案设计 9](#_Toc166873724)

[4.1 硬件部分 9](#_Toc166873725)

[4.1.1 STM32 最小系统板设计 9](#_Toc166873726)

[4.2 软件部分 12](#_Toc166873727)

[4.2.1 系统总体软件设计 12](#_Toc166873728)

[4.2.2 状态机 14](#_Toc166873729)

[4.2.3 模块设计思路 14](#_Toc166873730)

[附 录 18](#_Toc166873731)

印前将其字体颜色变为白色，在打印预览中看不见即可）：

# 前言

随着科技的不断进步，现代化智能设备在各行各业中的应用日益广泛。在交通运输行业，尤其是出租车服务中，传统的机械式计价器逐渐被电子计价器所取代。电子计价器不仅提高了计价的准确性和可靠性，还大大提升了用户体验。

近年来，智能语音技术的快速发展为出租车计价器的创新带来了新的契机。智能语音系统能够为乘客提供实时语音提示和报价，进一步提高了服务的便捷性和人性化程度。同时，随着移动支付技术的普及，无线支付功能也成为现代出租车计价系统的一个重要组成部分。

本实验旨在设计并实现一种基于STM32F107VCT6微控制器的智能语音出租车计价器电子系统。该系统集成了语音输出、计价、显示、实时时钟和无线支付等多项功能，力求通过先进的电子技术和智能算法，为乘客提供更加高效、准确和便捷的服务。

通过本次实验，我们将探讨如何利用STM32F107VCT6的外部中断功能、数据采集能力以及语音输出和显示技术，来实现一个功能完善且易于使用的智能出租车计价器系统。我们还将评估系统的稳定性和准确性，分析其在实际应用中的可行性和发展前景。

# 设计任务与需求分析

本实验的主要设计任务是开发一种智能语音出租车计价器电子系统。该系统需要满足以下基本功能和技术要求：

实时计价功能：根据车辆行驶的里程和时间，实时计算并显示乘车费用。

语音输出功能：通过语音合成模块，向乘客提供实时的语音报价和提示信息。

显示功能：使用ILI9341 LCD显示屏，显示实时计价信息、里程、时间等数据。

数据采集与处理：通过车轮转速传感器采集数据，并利用STM32F107VCT6微控制器进行处理，计算里程和速度。

实时时钟功能：利用DS13020实时时钟模块，记录和显示当前时间，为计价提供时间基准。

无线支付功能：提供无线支付选项，提升乘客支付的便捷性。

稳定性与准确性：系统需具备高稳定性和高准确性，能够在实际应用中可靠运行。

实时计价：系统应能够根据预设的费率和车辆行驶的里程、时间，实时计算乘车费用。支持启动和暂停计价功能，以应对不同的乘车情况。

语音输出：系统应能够通过语音合成模块，向乘客提供实时的语音报价和提示信息。语音输出应清晰、准确，并支持音量调节功能。

显示信息：系统应能够在ILI9341 LCD显示屏上，实时显示乘车费用、行驶里程、行驶时间和当前时间等信息。显示信息应布局合理，易于乘客读取。

数据采集与处理：系统应能够通过车轮转速传感器，实时采集车辆行驶的速度和里程数据。采集的数据需通过STM32F107VCT6微控制器进行处理，确保数据的准确性。

实时时钟：系统应配备DS13020实时时钟模块，提供准确的当前时间信息。实时时钟应具备高精度和低功耗特点。

无线支付：系统应支持主流的无线支付方式，如NFC或二维码支付，方便乘客支付车费。

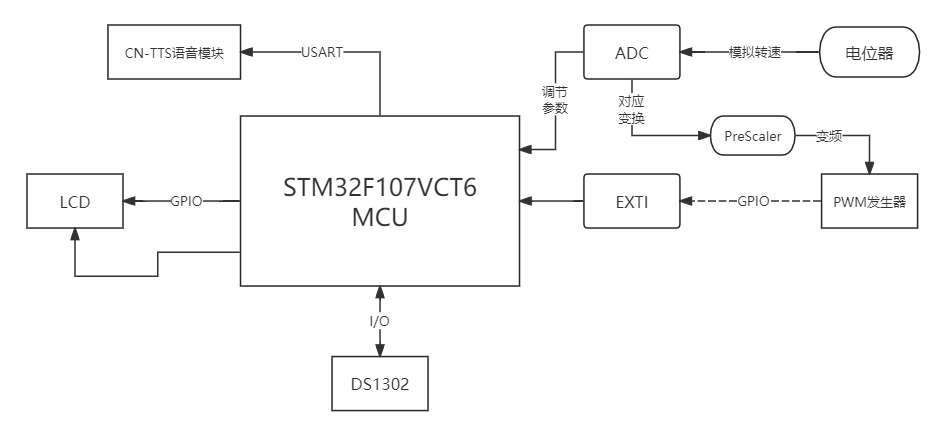


图1系统设计框图

# 工作模块及原理

根据设计要求确定了系统的总体方案，整个系统由单片机、DS1302和EXTI外部中断、LCD显示模块、ADC数据转换模块、语音模块以及按键等部分组成。系统功能原理图如图所示。

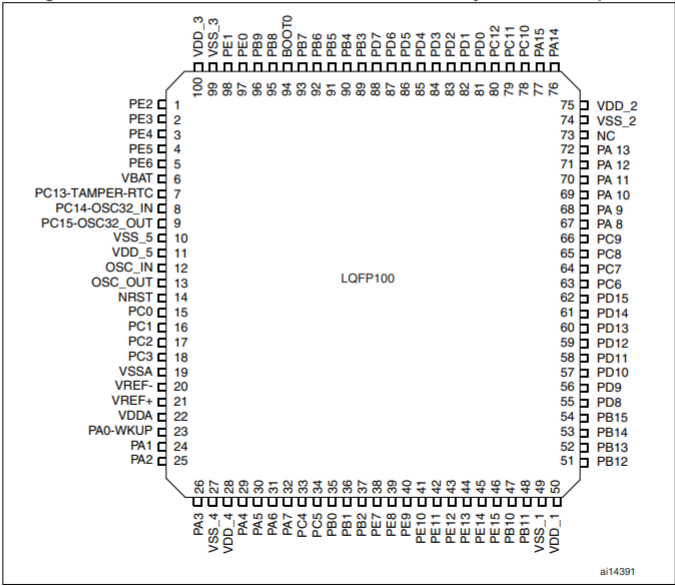


图2.STM32F107xx连接引脚图

## ADC及电位器

电位器是一种常用的可变电阻器，通过旋转或滑动调节器件，可以改变输出电压。它通常用于调节电压、控制音量、调整亮度等。在STM32微控制器系统中，电位器主要用于输入模拟信号，通过ADC（模数转换器）将模拟信号转换为数字信号，以供系统处理。在智能语音出租车计价器的设计中，电位器可以用于以下几种情况:

调节音量：通过电位器调整语音输出的音量，为乘客提供更好的听觉体验。

调节屏幕亮度：通过电位器调节显示屏的亮度，使得在不同光照条件下都能清晰显示信息。

调节其他参数：可以用电位器调节计价系统的各种参数，如基本费率、单位里程费用等。

硬件连接。

在这里我们使用电位器进行车轮变速的方式，通过检测电位器的电压水平引入调节参数，通过ADC将蒂电压水平的模拟量转换为数字量，在调用PWM发生中进行参数的引入，实现变频变速。

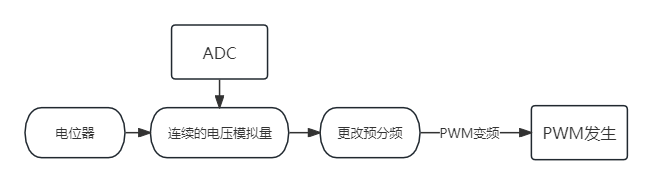


图3. PWM变频输出原理

## LCD液晶显示

### ILI9341介绍

ILI9341是一款性能优越且功能丰富的TFT LCD驱动芯片，非常适合用于智能语音出租车计价器的设计。其多种接口方式和高质量显示效果，为系统提供了稳定可靠的显示解决方案。

ILI9341显示屏在智能语音出租车计价器中主要用于显示实时计价信息、车辆行驶里程、乘车时间等重要数据。其高分辨率和丰富的色彩深度使得显示效果清晰、细腻，提升了用户体验。

通过STM32微控制器与ILI9341显示屏的结合，可以实现：实时显示计价信息、显示语音提示信息的可视化、用户界面的图形化展示。

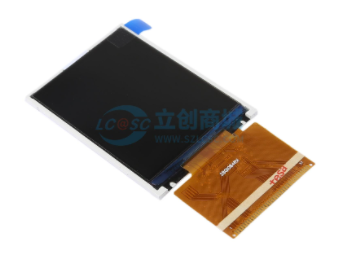


图4. ILI9341 LCD实物

LI9341 LCD显示屏是通过其内部的驱动IC与外部控制器（如STM32微控制器）进行通信和控制的。以下是详细的驱动原理介绍，包括硬件连接、软件配置、数据传输等方面。

### ILI9341驱动原理

1. 驱动架构

ILI9341的驱动架构可以分为几个主要部分：

控制接口：用于与外部微控制器通信，支持并行接口和SPI接口。

图像处理单元：负责接收和处理图像数据，并将其存储在内部的显存中。

时序控制器：生成显示所需的各种时序信号，包括行、列扫描信号。

电源管理模块：提供驱动显示屏所需的各种电压。

显示单元：包括像素阵列，接收图像处理单元传递的数据并实际显示图像。

2. 硬件连接

常见的连接方式是使用SPI接口，这是由于其引脚数量少、速度快、连接简单的优点。以下是典型的硬件连接示意图：

VCC：连接到3.3V电源

GND：接地

CS（Chip Select）：连接到STM32的GPIO引脚，用于选择显示屏

RESET：连接到STM32的GPIO引脚，用于复位显示屏

D/C（Data/Command）：连接到STM32的GPIO引脚，用于区分数据和命令

SDI（MOSI）：连接到STM32的SPI MOSI引脚，用于数据传输

SCK：连接到STM32的SPI时钟引脚

## DS1302

### ILI9341介绍

DS1302是一款低功耗、串行实时时钟（RTC）芯片，具有日历和时钟功能。它可以提供秒、分钟、小时、日、星期、月份和年份的信息，具有闰年自动校正功能。DS1302通过一个简单的串行接口与微控制器进行通信，支持数据的读写操作。该芯片广泛应用于需要准确时间记录的嵌入式系统中，如智能语音出租车计价器。



图5. DS1302芯片实物

1. 主要特性

时间和日历功能：提供秒、分钟、小时、日、星期、月份和年份信息，具有闰年校正功能。

串行接口：简单的三线串行接口（I/O、SCLK、RST）用于与微控制器通信。

低功耗：设计为低功耗设备，可以通过备用电池供电，确保断电后仍能保持时间信息。

数据保持：内置31字节的RAM，用于存储用户数据。

工作电压：2.0V到5.5V

封装形式：常见的8引脚DIP或SOIC封装，方便集成到各种电路设计中。

2. 硬件连接

DS1302与微控制器之间的连接相对简单，通过三线串行接口进行通信。以下是典型的硬件连接示意图：

VCC：连接到系统的3.3V或5V电源

GND：接地

I/O：连接到微控制器的GPIO引脚，用于数据传输

SCLK：连接到微控制器的GPIO引脚，用于时钟信号

RST：连接到微控制器的GPIO引脚，用于复位信号

在智能语音出租车计价器中，DS1302通常连接到STM32微控制器的GPIO引脚，通过软件模拟的串行接口进行通信。

### DS1302驱动原理

DS1302的驱动原理可以分为以下几个主要部分：

1.时钟管理：

DS1302内置的RTC单元负责管理和维护当前的时间和日期。芯片内部的振荡器电路使用外部连接的32.768kHz晶体振荡器产生精确的时间基准。RTC单元通过内部计数器保持时间信息，自动调整日期和处理闰年。

2.串行接口通信

DS1302使用一个简单的三线串行接口与微控制器进行通信。通信过程包括时钟信号（SCLK）、数据输入/输出（I/O）和复位信号（RST）：

SCLK（串行时钟）：由微控制器提供的时钟信号，用于同步数据传输。

I/O（输入/输出）：双向数据线，用于传输命令和数据。在数据传输过程中，微控制器通过这条线发送和接收数据。

RST（复位/片选）：用于启动和结束一次通信。当RST被拉高时，DS1302进入通信模式；当RST被拉低时，DS1302终止通信。

3.数据读写操作：

在与DS1302通信时，微控制器需要发送特定的命令字来执行读写操作。命令字包括读写指令和地址信息：

写操作：微控制器首先通过I/O线发送写命令和目标寄存器地址，然后发送要写入的数据字节。每次写操作结束后，微控制器将RST拉低，以终止通信。

读操作：微控制器首先通过I/O线发送读命令和目标寄存器地址，然后切换I/O线为输入模式，读取DS1302返回的数据字节。读取操作完成后，微控制器将RST拉低，以终止通信。

4.数据保持与电源管理：

DS1302具有内置的电源管理模块，可以连接主电源和备用电池。当主电源断电时，DS1302会自动切换到备用电池供电，继续保持时间和日期信息。这样，即使系统断电，DS1302仍能确保时间信息的连续性和准确性。

5.RAM数据存储：

除了实时时钟功能，DS1302还内置了31字节的静态RAM，可用于存储用户数据。用户可以通过串行接口读写这些RAM数据，为系统提供额外的非易失性存储空间。

## CN-TTS语音模块

CN-TTS是一款高集成度的语音合成模块，可实现中文、英文、数字的语音合成；并且支持用户的命令词或提示音的定制需求。

CN-TTS控制方式简单，是通过TTL串口发送GBK编码的形式，可兼容市面上主流5V或3.3V单片机。

支持任意中文、英文字母、阿拉伯数字的文本合成，并且支持中文、英文字母、数字的混读。

模块支持中文GBK编码集；支持大、小写英文字母。模块采用UART通讯方式，UART串口支持9600 bps，简单易用。支持状态显示用户的控制器能够清楚地了解模块是否正在合成播报，还是空闲状态。

# 总体方案设计

## 硬件部分

### STM32 最小系统板设计

#### 复位电路

复位电路，就是指单片机芯片可以通过外部外部引脚输入复位电平信号，从而使单片机除备份区以外的所有寄存器都恢复为默认值，且芯片内烧录的程序会重新执行。对于STM32103C8T6芯片来说，其复位引脚是7号引脚，也就是NRST引脚。

通过复位电路对STM32芯片系统进行的复位是外部复位，除此之外，STM32芯片内部还有独立看门狗IWDG复位、窗口看门狗WWDG复位、软件SW复位和低功耗管理复位。独立看门狗和窗口看门狗就是一个递减计数器，当计数器的值减小到0的时候，系统就会进行复位。软件SW复位是通过程序在STM32的一个寄存器的SW位赋值来控制复位。低功耗管理复位可以在待机或者停机状态下将STM32的系统复位。除了复位电路的外部复位，其他复位模式并不需要自己设计，本文就仅设计了STM32的外部复位电路。

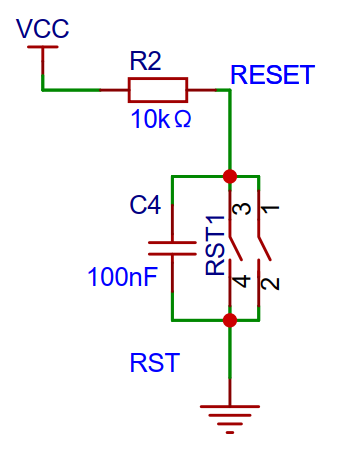


图6. 复位电路

#### 时钟电路

STM32芯片内部是由十分复杂的数字电路以及其他电路组成，需要稳定的时钟脉冲信号才能够保证其正常工作，通过STM32的数据手册可知，STM32芯片要想正常工作，其需要用到时钟的模块有很多。其分别是Contex-M3内核、闪存FLASH、可配置的静态存储器控制器FSMC、SDIO、内置SRAM、DMA直接数据存储、AHB总线、APB1总线及APB2总线。其中Contex-M3需要用到Contex自由运行时钟FCLK和Contex系统时钟SYSCLK。闪存FLASH、内置SRAM、DMA直接数据存储器和AHB总线则需要用到高速时钟HCLK。FSMC和SDIO也有其特有的时钟，分别为FSMCCLK和SDIOCLK。APB1和APB2也有其特有的时钟。由于芯片的内部集成化要求，不可能将每个模块都配置一个时钟源，因此芯片内部对已有的时钟源进行分频或者倍频操作，从而达到各个模块要求的时钟频率。

通过STM32的数据手册得知，STM32的芯片内部有三个时钟源，分别是HSI、LSI、PLL倍频器，除此之外还具备两个可以接外部时钟源的引脚，分别是OSCIN和OSC32IN。其中OSCIN引脚可以接4~16MHZ的晶体振荡器，在本设计中采用8MHZ的晶体振荡器。OSC32引脚通常接入32.768KHZ的晶振。本设计所采用的晶振都为石英晶振，因此外部晶振模块的抗干扰能力和精度相较于系统内部已有的晶振要提升很多，所以正常情况下，模块使用外部晶振。

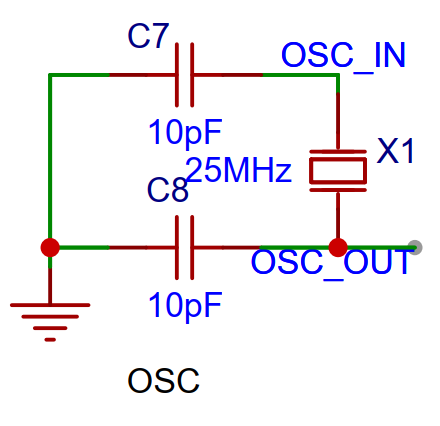


图7. 时钟电路

#### 电源电路

通过对STM32数据手册的查阅可以得知，STM32的芯片工作电压范围为2.0-3.6V。STM32提供给模数转换电路的电压范围为2.4V-3.6V，在对模数转换没有较高精度要求的时候，其可以直接接到芯片的供电电源上，本文所设计的电源电路提供给STM32芯片的电压值为3.3V。

由于此最小系统板的设计采用USB供电，USB的直接供电电压是5V，因此需要设计出一个降压电路模块来将5V电压降到芯片所能够承受的3.3V。本设计在降压电路模块所采用的稳压芯片是RT9193。该稳压芯片具有输入端、参考噪声旁路端口、使能端、接地端和输出端五个引脚。电源电路原理图中输入端接USB提供的5V电压。BP端口接滤波电容。输出端口输出3.3V的稳定电压，输出电路部分接电容C3和C4，其作用为抑制自激振荡和稳定输出电压。

由于芯片中各个电路网络中电流变化时，各个电路模块之间会产生耦合现象，在电源电路模块中产生的耦合电流会对整个最小系统板的正常工作产生影响，因此可以设计一个退耦电容电路来消除各个电路模块之间产生的寄生耦合。本文所设计的退耦电路模块是在电路正极和地之间并联了退耦电容。退耦电路原理图如下。

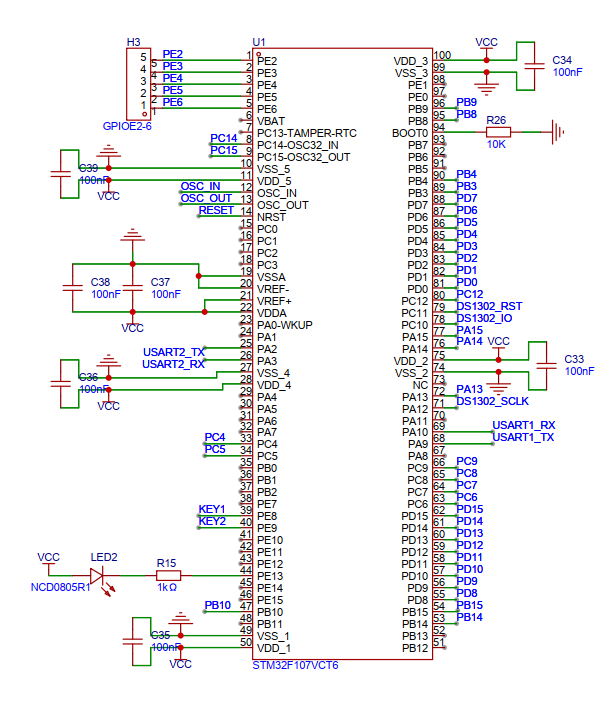


图8. 滤波电容分布

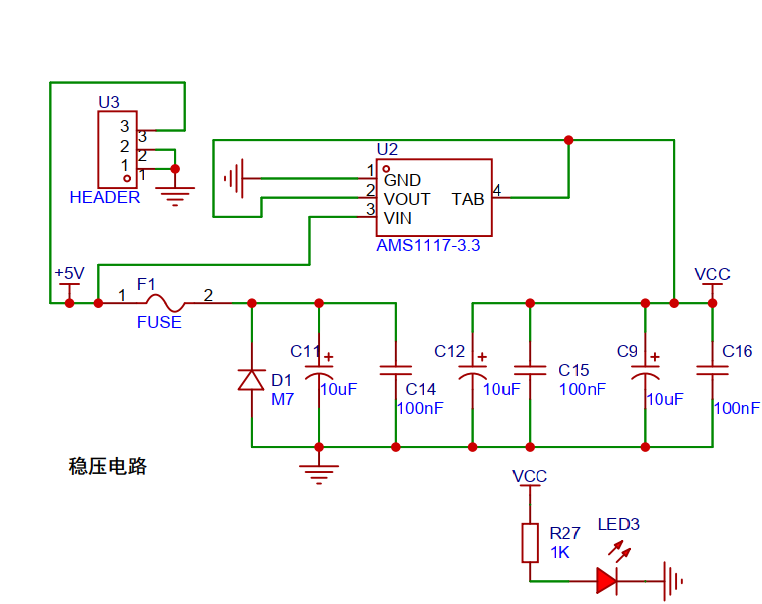


图9. 电源电路

#### BOOT电路

通过STM32的用户手册可以得知，STM32有三种启动状态，其由BOOT1和BOOT0引脚控制，BOOT0和BOOT1的值与启动模式之间的具体关系如下表所示。在本设计中，BOOT0对应芯片的引脚为44号引脚，BOOT1对应引脚为20号引脚。本设计通过一个3\*2的跳线针来作为主体，其1、2号引脚接3.3V电源，3、4号引脚分别通过一个电阻与BOOT0和BOOT1相连接，5、6号引脚接地。需要更改STM32的启动模式时，只需要用跳线帽就可以达到更改BOOT0和BOOT1的值的目的，从而就可以更改启动模式。

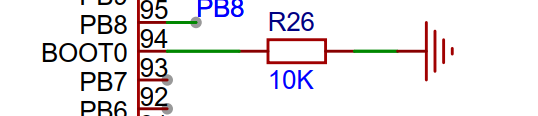


图10. BOOT电路

#### DC-DC转换电路

MP2315是一款内置内部功率 MOSFET 的高频同步整流降压开关模式变换器。可在宽输入范围内实现 3A 的连续输出电流，并具有出色的负载和线路调节能力。MP2315 采用同步工作模式，可在输出电流负载范围内实现更高的效率。电流模式操作提供快速瞬态响应，并简化环路稳定。全面保护功能包括 OCP 和热关断。MP2315 最大限度地减少了现有标准外部元器件的使用，采用节省空间的 8 引脚 TSOT23 封装。

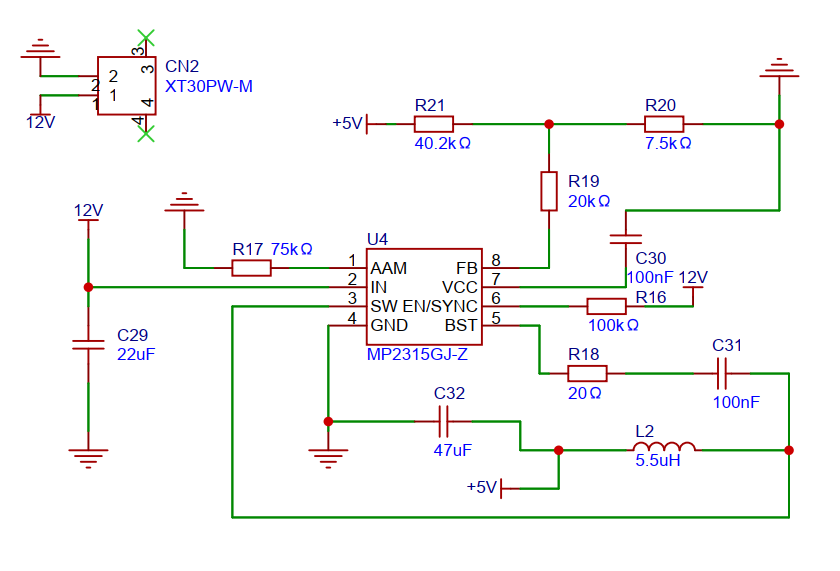


图11. 降压电路

### 管脚接线

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **按键** |  |  | **其他** |  |
| KEY1 | PE8 |  | TP\_CLK | PC12 |
| KEY2 | PE9 |  | TP\_MOSI | PB15 |
| KEY3 | PE10 |  | TP\_MISO | PB14 |
| KEY4 | PE11 |  | TP\_CS | PC9 |
|  |  |  | TP\_PEN | PB10 |
| LCD |  |  | F\_CS | PB8 |
| LCD\_D0 | PD0 |  |  |  |
| LCD\_D1 | PD1 |  |  |  |
| LCD\_D2 | PD2 |  | TIM2\_CH1 | PA0\_WKUP |
| LCD\_D3 | PD3 |  |  |  |
| LCD\_D4 | PD4 |  | RCC\_OSC32\_IN | PC14 |
| LCD\_D5 | PD5 |  | RCC\_OSC32\_OUT | PC15 |
| LCD\_D6 | PD6 |  |  |  |
| LCD\_D7 | PD7 |  | RCC\_OSC\_IN | RCC\_OSC\_IN |
| LCD\_D8 | PD8 |  | RCC\_OSC\_OUT | RCC\_OSC\_OUT |
| LCD\_D9 | PD9 |  |  |  |
| LCD\_D10 | PD10 |  |  |  |
| LCD\_D11 | PD11 |  | **IO** |  |
| LCD\_D12 | PD12 |  | 边沿触发 | PE7 |
| LCD\_D13 | PD13 |  |  |  |
| LCD\_D14 | PD14 |  | **SYS** |  |
| LCD\_D15 | PD15 |  | SYS\_JTDI | PA15 |
| LCD\_BL | PC8 |  | SYS\_JTCK-SWC | PA14 |
| LCD\_RS | PC7 |  | SYS\_JTDO | PB3 |
| LCD\_CS | PC6 |  | SYS\_NJTRST | PB4 |
| LCD\_RD | PC5 |  | SYS\_JTMS\_SWDIO | PA13 |
| LCD\_WR | PC4 |  |  |  |
|  |  |  | **UART** |  |
|  |  |  | USART1\_TX | PA10 |
| **DS1302** |  |  | USART1\_RX | PA9 |
| DS1302\_CE | PC11 |  | USART2\_TX | PA2 |
| DS1302\_DATA | PC10 |  | USART2\_RX | PA3 |
| DS1302\_SCLK | PA12 |  |  |  |
|  |  |  | **LED** |  |
| **ADC** |  |  | LED1 | PB9 |
| ADC1\_IN7 | PA7 |  |  |  |

## 软件部分

### 系统总体软件设计

系统软件框图如图3.1所示。系统总流程图如图3.2所示。

PWM波生成模块产生可控频率的方波用于模拟车轮转速，由主程序中的外部中断接收；ds1302模块产生搭载时间数据的信号，由主程序中的GPIO接收；主程序根据状态机的状态，通过USART串口通讯向语音模块发送指令并接收其回传信息；并通过GPIO向LCD模块发送指令。

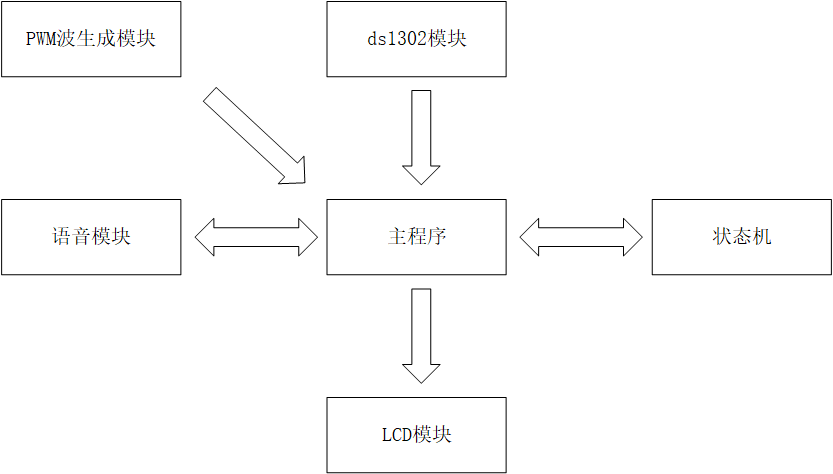


图12.系统软件框图



图13.系统总流程图

### 状态机

系统有两个独立的状态机，分别为计程状态机和用户界面状态机，如图3.3和图3.4所示，其中括号内的内容为初始化赋值，每切换到该状态时只执行一次，并非该状态下不变的量。

计程状态机有初始、计程和完成三个状态。初始状态为系统的闲置状态，计程和总价恒为0。计程状态为系统的运行状态，此时计数器记录来自PWM波生成模块触发的外部中断数，用于计算路程和总价。完成状态为系统的暂停状态，此时计数器停止计数，计程和总价不再改变，并开始播报语音。

用户界面状态机有界面1和界面2两种状态。用户界面1为计程计价界面，显示总价、总公里数、单价、起步价、时间和日期，分为用户界面1初始化函数和用户界面1函数两个函数。用户界面1初始化函数用于显示恒定不变的字，如“总价”“计程”这些字，这些字只需要打印一次。用户界面1函数用于计算和显示一直变化的数字，如具体的总价和时速的值，这些数字需要不断刷新和打印。用户界面2为付费界面，显示微信支付二维码，由于只需要打印一次，故只有用户界面2初始化函数一个函数。

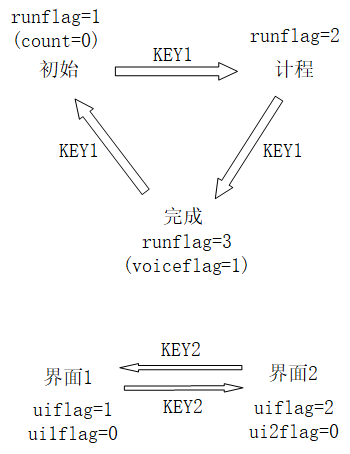


图3.3 计程状态机

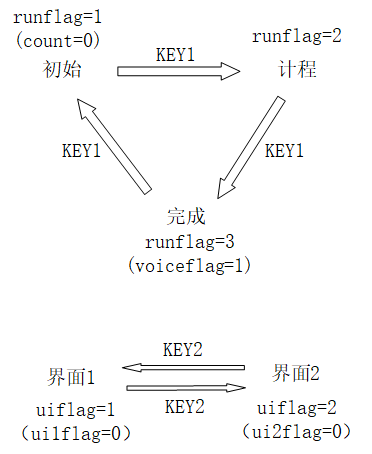


图14. 用户界面状态机

### 模块设计思路

#### PWM波生成模块

此模块用于产生可实时控制频率的方波用于模拟车轮转速。

将一电位器与5V电压连接，其可输出0~5V的电压，电压幅值可由电位器旋钮控制。

ADC模块通过定时器TIM3进行频率为25Hz的采样，参考电压为5V，采样结果通过通过DMA传给芯片并存储到数组ADC\_Value中，在ADC回调函数中进行数据处理，采样5次取平均值减小偶然性误差。

启用TIM2产生PWM波用于模拟车轮转速，其时基时钟为72MHz，计时周期Counter Period设定为35999，比较值Pulse设定为17999，此时TIM时钟进行了36000分频，频率为2000Hz，占空比为50%。预分频值由ADC采样值决定，其计算公式为

预分频值 = 154 + 1846 \* ADC采样值 / 4096

预分频值的范围为154~2000，对应TIM时钟频率13Hz~1Hz。设定出租车轮胎每转一圈为1.2m，则对应汽车时速10km/h~100km/h。预分频值的更新在ADC回调函数中通过宏定义完成。

#### 计数器

外部中断EXTI7用于接收来自PWM波生成模块的PWM波，上升/下降沿触发，每触发一次，用于计算时速的变量countspeed加一，同时，如果状态机为计程状态，用于计算路程的变量count加一。

此外，为了演示方便，设定一个另外的外部中断，由按键KEY3触发，每触发一次，变量count加834，对应路程增加1公里。

#### LCD模块

此模块用于在LCD显示屏上显示汉字、图片和ASCII码。

通过设置光标位置和写入GRAM的方式可对对单个像素点进行设置。通过对汉字和图片取模可以得到其模值数组。汉字默认其纵向长度等于横向宽度。ASCII码默认其纵向长度等于横向宽度的一半。图片专门设定一个结构体，用于存储图片的长度、宽度、和模值数组。

对于模值数组，设定一指针temp，初始指向此模值数组第一个元素的第8位，每设置完一个像素temp向左移动1位，当temp指向该元素的第1位时，下一次移动至下一个元素的第8位，如图3.5所示，如此循环。

对于显示屏，将显示屏的横向设定为x轴，纵向设定为y轴。从给定的初始位置(x0,y0)开始，每从temp中读取一个模值，每设置一个像素点，然后光标位置沿x轴正方向移动一个像素，当检测到当前x值与初始x0的差值(x-x0)等于字体大小size时，光标的x回到x0处，然后沿y轴正方向移动一个像素，如图3.6所示，如此循环。

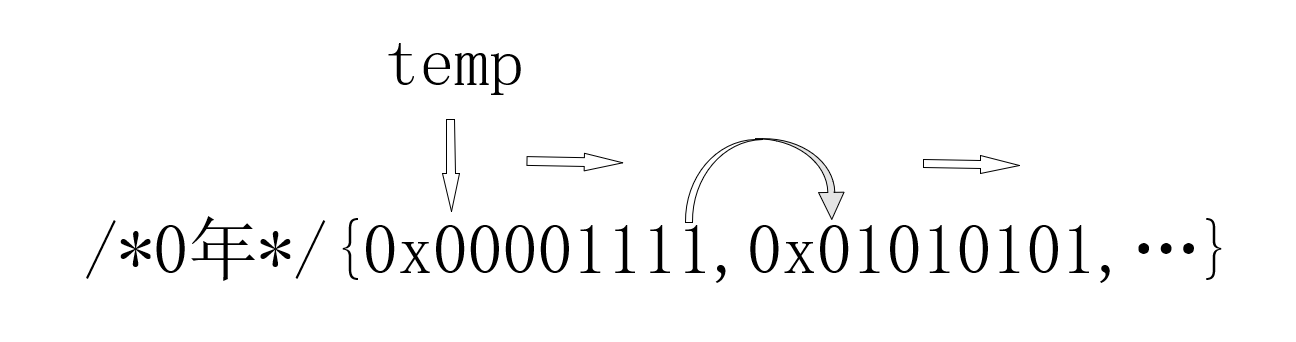


图15.字模数组

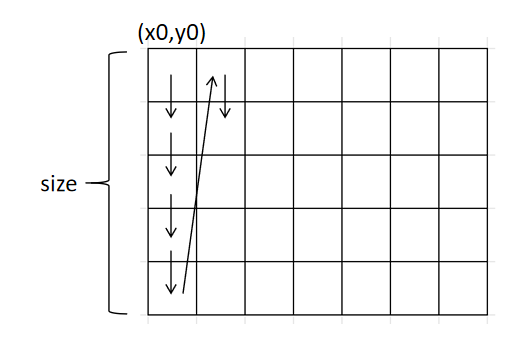


图16.字模取值可视化

#### 路程和总价的计算

首先从ds1320模块中获取当前时间，然后设定起步价、单价和长途价，其中长途价为超公里后单价和空驶费之和。当时间在6：00~23：00时，起步价设定为8.5，单价设定为2，长途价设定为3。为方便进行小数运算，所有数字乘100以消除小数，即起步价设定为850，单价200，空驶费300。当时间在23：00~次日6：00时，起步价设定为950，单价230，长途价345。

计算路程distance。设定轮胎每转一圈，汽车行驶1.2m，则路程distance等于外部中断计数count乘上1.2，同样为方便进行小数运算，让此数字再乘上10，即

distance=count\*12

计算总价price。计算公式为



数值的显示以小数点为基准。例如price/100为元，显示两位在小数点前；price%100为角和分，显示两位在小数点后；遇到角等于零、分不等于零的情况时，角不显示，需要补零。

#### 用户界面的刷新。

启用TIM4，频率为2Hz，实现每0.5秒刷新一次屏幕。检测用户界面标识uiflag的值，若为1，则执行一次用户界面1初始化函数，然后循环执行用户界面1函数；若uiflag的值为2，则执行一次用户界面2初始化函数。

#### 时速的计算

启用TIM5，频率为10Hz。不断读取速度计数变量countspeed的值，连续两次读取的差值即为0.1s内触发的外部中断数，即0.1s内出租车轮胎转过的圈数。为减小偶然性，连续读取5次取平均值。

设定一数组countarray[10]用于存储数据，然后不断读取速度计数变量countspeed的值，每读取一次，数组countarray[10]中的元素整体右移一个元素，然后将countspeed的值填入countarray[0]中，即

countarray[9]=countarray[8];

countarray[8]=countarray[7];

...

countarray[1]=countarray[0];

countarray[0]=countspeed;

则countarray[5]-countarray[0]为0.5s内的圈数，则数组countarray[10]中前五项元素的平均值减去后五项元素的平均值即为连续5次读取的0.5s内圈数的平均值。设定轮胎每转过一圈，出租车行驶1.2m。单位换算1m/s=3.6km/h。故时速计算公式为



其中，speed为时速，SUM( )为求和函数。

#### 语音模块

语音模块使用的是CN\_TTS语音合成模块，用于播报总公里数和总价。此模块使用串口通讯，串口参数（9600，8，N，1），汉字采用GBK编码。在模块使用说明中，模块接收到字符开始播报会回传“Playing”，接收到字符播报完成会回传“End”，但是经过实测，两种情况分别会回传“A”和“O”，以下均以实测数据为准。

通过串口发送给CN\_TTS模块的内容存储在语音内容数组str[50]中。数组str[50]初始值为空，设定一变量h，初始为0，改变str[h]的值，然后h加一，如此循环，即可将数组str填充完整的一句话，具体流程如图3所示。此外，所使用的编辑器无法自动将汉字识别为GBK编码，故需要手动查询所用汉字的GBK编码，查询网站为gbk.hsjsj.cn，其中，由于“角”和“分”的读音不为预期读音，故分别改为“脚”和“芬”的GBK编码。

每当检测到语音标志voiceflag为1时，将voiceflag置零，并开始向CN\_TTS模块发送语音字符，同时通过DMA接受不定长数组，存储在数组receiveData[2]中。在串口回调函数中，当检测到receiveData的第二个元素为“O”，即字符播报完成时，将receiveData[2]、str[50]、h全部清零，进行usart串口初始化以强制中断DMA的接收。且由于不明原因，播报语音后LCD会白屏，故同时进行LCD初始化并将用户界面1标志ui1flag置零。

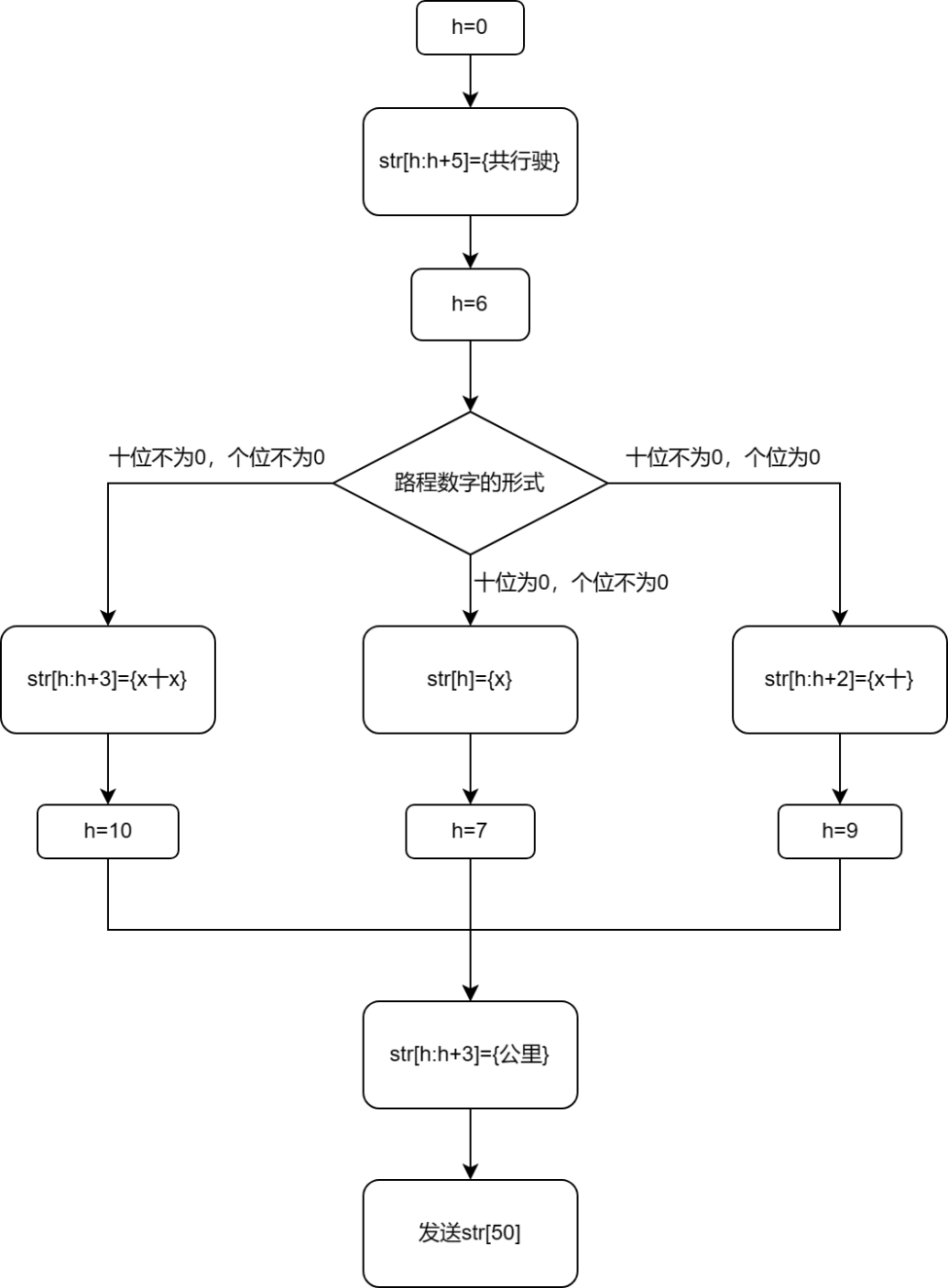
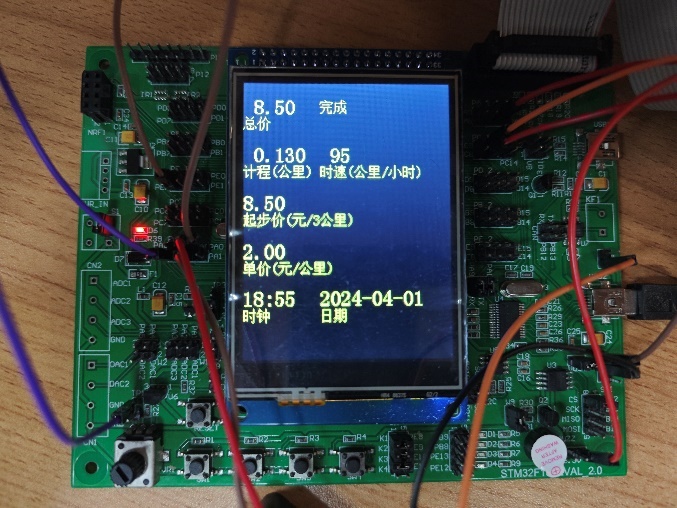


图17.语音模块原理

# 系统调试



# 附 录

