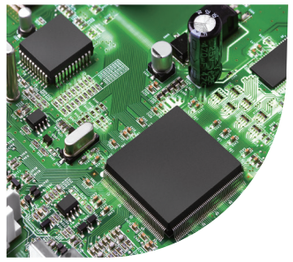
嵌入式系统开发与应用



ISBN 978-7-5775-0092-8
9 787577 500928 >
定价：55.00元

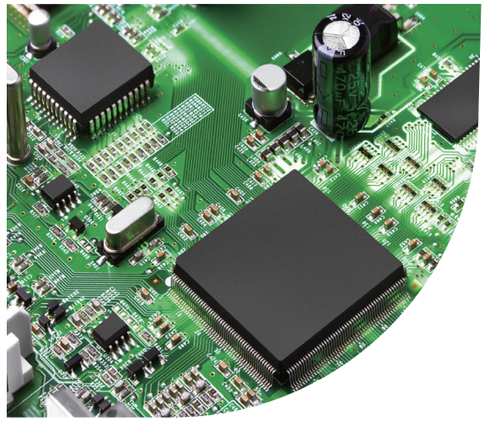
嵌入式系统开发与应用　主编　程小荣

西北工業大學出版社

高等职业教育电子信息系列精品教材

嵌入式系统开发与应用

主　编　程小荣
副主编　马　俊　张宇峰

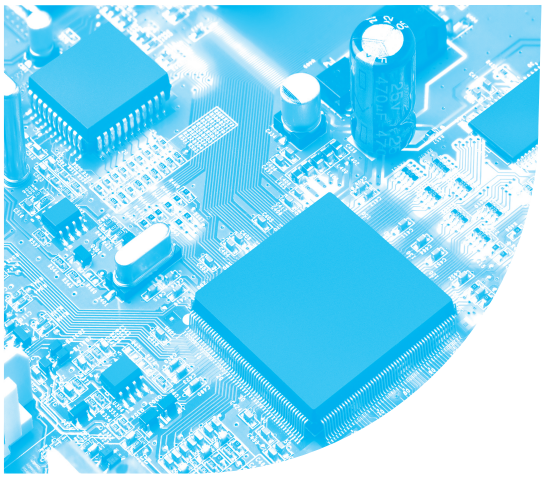


西北工業大學出版社

高等职业教育电子信息系列精品教材

嵌入式系统开发与应用

主　编　程小荣
副主编　马　俊　张宇峰



西北工业大学出版社
西　安

【内容简介】 本书分为14个项目：项目一为项目环境介绍，介绍了扫地机器人硬件平台的组成与原理，以及Keil μ Vision4、ST-LINK驱动等软件环境的搭建与使用；项目二至项目十四涵盖LED（发光二极管）灯设计、蜂鸣器应用、继电器与主控板供电控制、串行通信及OLED（有机发光二极管）显示屏应用等实践内容。除项目一外，每个项目均以任务为导向，结合代码实例与硬件调试步骤，强化学生的动手能力与工程思维意识。

本书既可作为高等职业院校和应用型本科院校电子信息工程、自动化、智能网联汽车等相关专业的教材，也可作为嵌入式系统开发初学者的参考用书。

图书在版编目（CIP）数据

嵌入式系统开发与应用 / 程小荣主编. -- 西安：西北工业大学出版社，2025.8. -- ISBN 978-7-5775-0092-8

Ⅰ. TP332.021

中国国家版本馆CIP数据核字第2025FH3641号

QIANRUSHI XITONG KAIFA YU YINGYONG

嵌 入 式 系 统 开 发 与 应 用

程小荣　主编

责任编辑： 曹　江　　**装帧设计：** 黄燕美

责任校对： 马　丹

出版发行： 西北工业大学出版社

通信地址： 西安市友谊西路127号　　邮编：710072

电　　话： (029)88491757，88493844

网　　址： www.nwpup.com

印 刷 者： 三河市骏杰印刷有限公司

开　　本： 787 mm×1 092 mm　　1/16

印　　张： 16

字　　数： 349千字

版　　次： 2025年8月第1版　　2025年8月第1次印刷

书　　号： ISBN 978-7-5775-0092-8

定　　价： 55.00元

PREFACE 前言

随着物联网、智能硬件及智能网联汽车等技术的快速发展，嵌入式系统开发技术成为现代电子工程领域的核心技术之一。为满足各行业对嵌入式系统开发高素质技术技能型人才的需求，本书以 STM32 单片机为载体，以嵌入式系统开发为核心，结合扫地机器人这一典型设备，系统设计了一套理论与实践并重的教学方案，通过项目引领、任务驱动的教学模式，帮助学生掌握嵌入式系统开发的全流程。

本书系统讲解了嵌入式系统开发的基础知识、硬件平台设计、软件环境配置及典型功能模块开发；以 14 个核心项目为主线，除项目一外，将每个项目分解为多个具体任务，通过“项目目的—项目准备—项目分析—支撑知识—任务”的递进式学习路径，提升学生解决实际工程问题的能力。

本书内容由浅入深，理论与实践紧密结合，并提供丰富的实验程序与工程模板，其编写特色如下。

(1)软硬结合，注重实操。本书详细解析硬件原理图与软件代码，提供 Keil(一款集成开发环境软件)工程模板、驱动配置指南及调试技巧，降低学习门槛。

(2)行业应用导向鲜明。本书基于扫地机器人的真实应用场景，融合红外传感、电机控制、无线通信等工业技术，案例设计紧贴行业需求，能够帮助学生快速提升岗位技能。

(3)模块化设计，灵活适配。本书采用模块化结构设计，项目间既独立成篇又可串联整合，教师可根据课时自由组合教学内容，适应不同层次院校的差异化教学需求。

(4)任务评价体系完善。除项目一外，本书每个项目均设置“任务评价”模块，包含团队合作能力、功能实现展示、问题解决能力等多维度评分标准，便于教师对学生量化考核与学生自我检验，强化学习效果的反馈。

本书由苏州工业职业技术学院程小荣担任主编，其负责全书框架设计及统稿工作，以及项目一至项目四、项目十三、项目十四内容的编写；由苏州工业职业技术学院马俊、张宇峰担任副主编，其中马俊负责硬件平台原理图解析与传感器模块开发，以及项目五至项目七内容

的编写，张宇峰负责串行通信、OLED 显示等软件功能实现，以及项目八至项目十内容的编写；苏州工业职业技术学院李洪群、臧华东参与了本书的编写，其中李洪群负责实验案例设计与调试，以及项目十一内容的编写，臧华东负责代码优化与文档整理，以及项目十二内容的编写。

本书的编写得到了苏州工业职业技术学院汽车工程学院的大力支持，行业专家为本书的编写提供了技术指导，合作企业常州市炜煌科教仪器有限公司为本书的编写提供了硬件资源上的协助，苏州工业职业技术学院的师生为本书内容提供了许多宝贵意见，在此一并表示感谢。同时，笔者参考了相关文献资料，在此对其作者一并表示感谢。

限于笔者水平，书中难免存在不足之处，恳请读者批评指正。

编　者

2025 年 4 月

CONTENTS 目录

项目一 项目环境介绍

一、嵌入式系统概述

嵌入式系统(embedded system)是一种完全嵌入受控器件内部,为特定应用设计的专用计算机系统。它可以用来控制或者监视机器、装置、工厂等大规模设备,被应用于工业控制、交通管理、家庭智能管理等领域。与个人计算机的通用计算机系统不同,嵌入式系统通常执行的是有特定要求的预先定义的任务。

嵌入式系统的核心由一个或几个预先编好程序的,用来执行少数几项任务的微处理器或者单片机组成。与通用计算机能够运行用户选择的软件不同,嵌入式系统上的软件通常是固定不变的,称为固件。

一般的嵌入式系统装置由嵌入式系统和执行装置组成,嵌入式系统(由硬件层、中间层、系统软件层和应用软件层组成)是整个嵌入式系统装置的核心。

1. 硬件层

硬件层中包含嵌入式微处理器、存储器、通用设备接口和 I/O(输入/输出)接口[A/D(模/数)、D/A(数/模)等]。在一片嵌入式微处理器基础上添加电源电路、时钟电路和存储器电路,就构成了一个嵌入式核心控制模块。其中操作系统和应用程序都可以固化在 ROM(只读存储器)中。

2. 中间层

硬件层与软件层之间为中间层,也称为 HAL(硬件抽象层)或 BSP(板级支持包)。它将系统上层软件与底层硬件分离开来,使系统的底层驱动程序与硬件无关。上层软件开发人员无须关心底层硬件的具体情况,根据 BSP 提供的接口即可进行开发。中间层一般包含相关底层硬件的初始化、数据的输入/输出操作和硬件设备的配置功能。

3. 系统软件层和应用软件层

系统软件层由 RTOS(实时多任务操作系统)、文件系统、GUI(图形用户接口)、网络系统及通用组件模块组成。RTOS 是嵌入式系统应用软件的基础和开发平台。

应用软件层由基于实时多任务操作系统开发的应用程序组成,用于控制被控对象。

二、扫地机器人硬件平台介绍

1. 主控板

扫地机器人主控板的布局图如图 1-1 所示，扫地机器人主控板的实物图如图 1-2 所示。

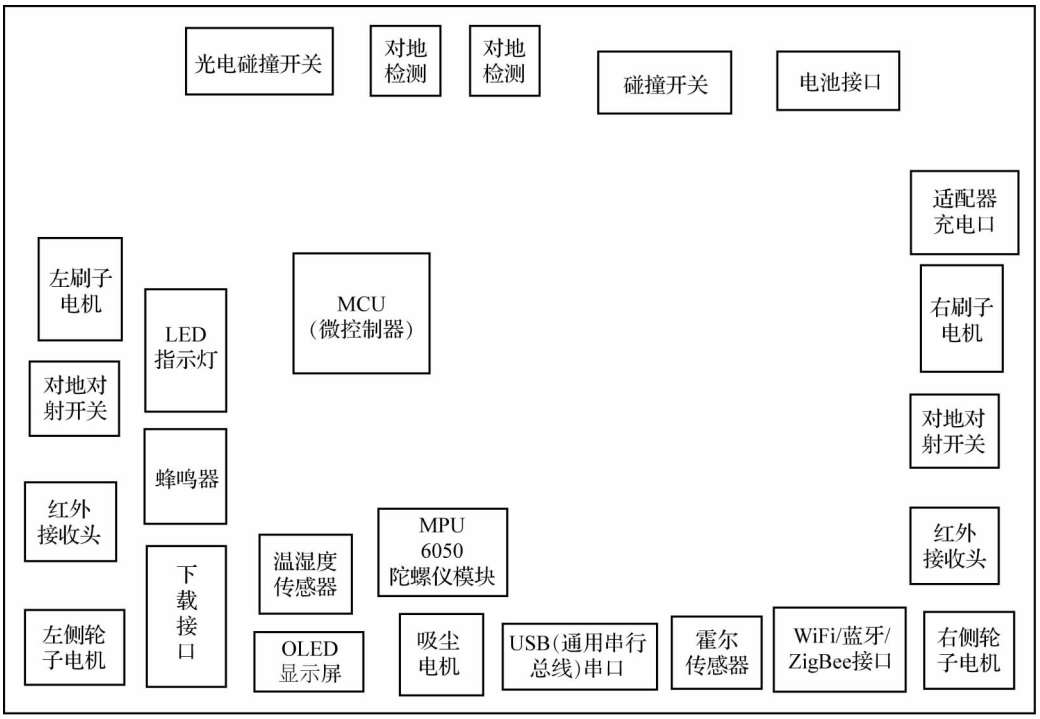


图 1-1　扫地机器人主控板的布局图

各元器件及其功能如下。

对地对射开关：用于检测扫地机器人当前所处的地面环境，防止地面悬空，扫地机器人跌落。

光电碰撞开关：用于检测扫地机器人前方是否有障碍物，避免扫地机器人碰撞障碍物。

复位按键：用于控制扫地机器人程序复位。

刷子电机：用于控制扫地机器人的扫地刷。

蜂鸣器：用于实现扫地机器人的提示功能。

红外接收头：扫地机器人充电回充时用于接收来自充电座的红外导航信息。

轮子电机：用于控制扫地机器人的行动。

下载接口：为扫地机器人主控板提供程序下载接口，此处采用 ST-LINK 下载方式。

OLED 显示屏：可以显示扫地机器人当前的工作状态等信息。

MPU6050 陀螺仪模块：用于指示扫地机器人的位置变化。

吸尘电机：用于吸除地上的灰尘。

USB(通用串行总线)串口：为扫地机器人主控板提供串口和供电功能。

霍尔传感器：用于检测磁场及扫地机器人的水箱是否安装上。

WiFi/蓝牙/ZigBee 接口：WiFi、蓝牙、ZigBee 三种通信模块的公共接口，用于采用无线通信方式控制扫地机器人。

适配器充电口：扫地机器人适配器的供电接口。

电池接口：扫地机器人电池供电、充电座芯片的供电接口。

MCU：扫地机器人的主控芯片，其型号为 STM32F103VET6。

LED 指示灯：用于指示扫地机器人主控板的各种状态。

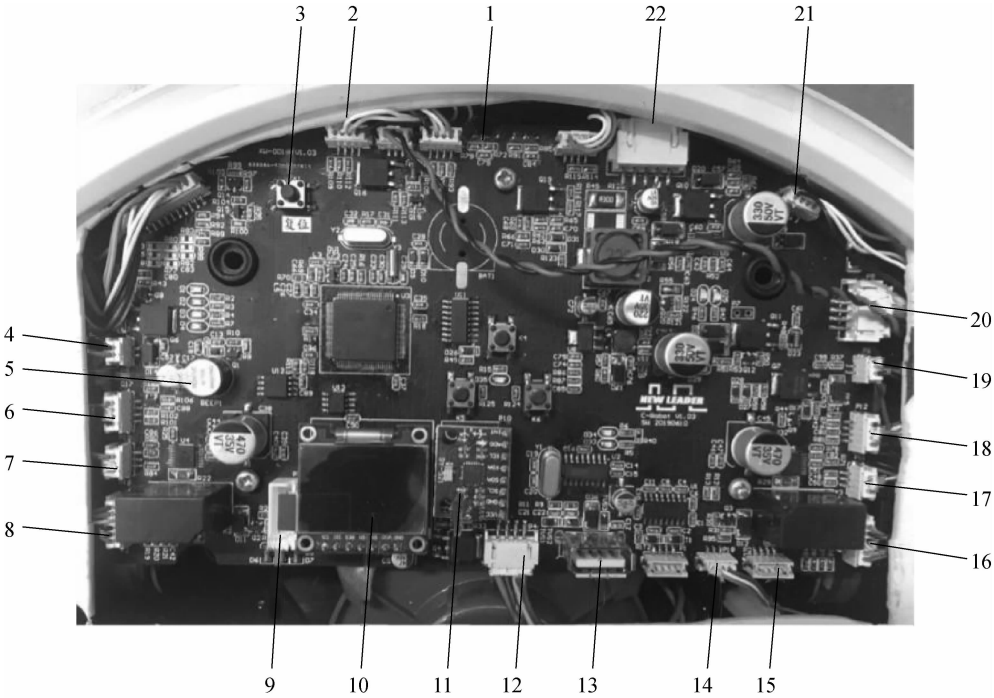


图 1-2　扫地机器人主控板的实物图

1—对地对射开关；2—光电碰撞开关；3—复位按键；4—左侧刷子电机；5—蜂鸣器；6—左侧对地对射开关；7—红外接收头；8—左侧轮子电机；9—下载接口；10—OLED 显示屏；11—MPU6050 陀螺仪模块；12—吸尘电机；13—USB 串口；14—霍尔传感器；15—WiFi/蓝牙/ZigBee 接口；16—右侧轮子电机；17—红外接收头；18—右侧对地对射开关；19—右侧刷子电机；20—适配器充电口；21—红外对射开关；22—电池接口

2. 充电板

扫地机器人充电板的实物图如图 1-3 所示。各组成部分及其功能如下。

(1)电源指示灯:用于指示电源。

(2)红外发射管:通过发射红外载波来指示扫地机器人的当前位置。

(3)下载接口:为扫地机器人充电板提供程序下载接口,此处采用 ST-LINK 下载方式。

(4)LED 灯:指示扫地机器人的位置。

(5)电源接口:为扫地机器人充电板提供电源的接口。

(6)MCU:充电板主控芯片,型号为 STM32F103C8T6。

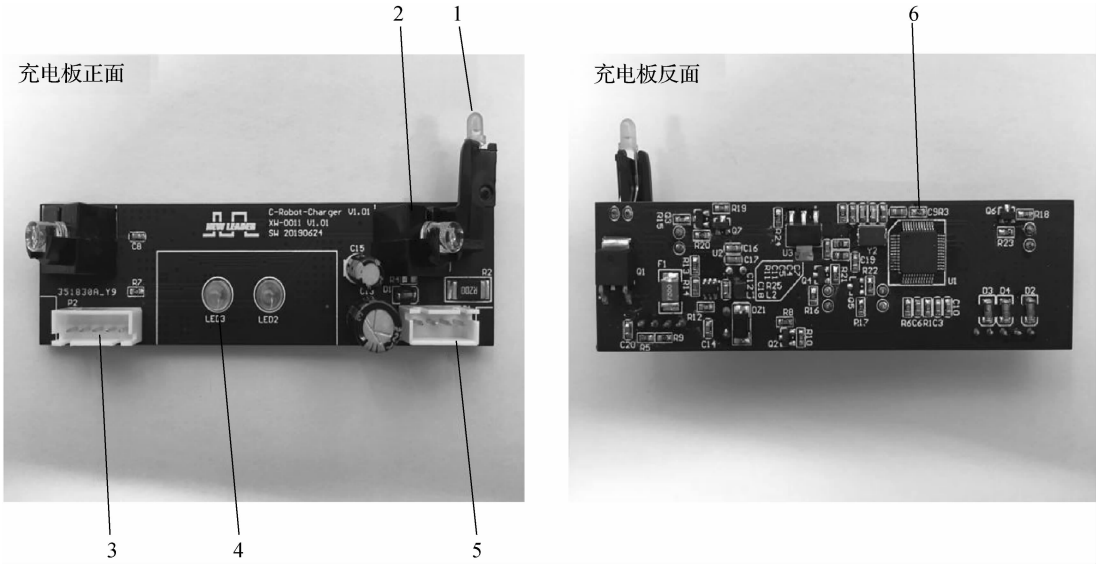


图 1-3　扫地机器人充电板的实物图

1—电源指示灯;2—红外发射管;3—下载接口;

4—LED 灯;5—电源接口;6—MCU

3. 扫地机器人硬件模块原理图

主控板电源原理图、MCU 原理图、Flash(闪存)/EEPROM(电擦除可编程只读存储器)原理图、电机原理图、传感器原理图、外围接口原理图、充电板原理图分别如图 1-4～图 1-10 所示。

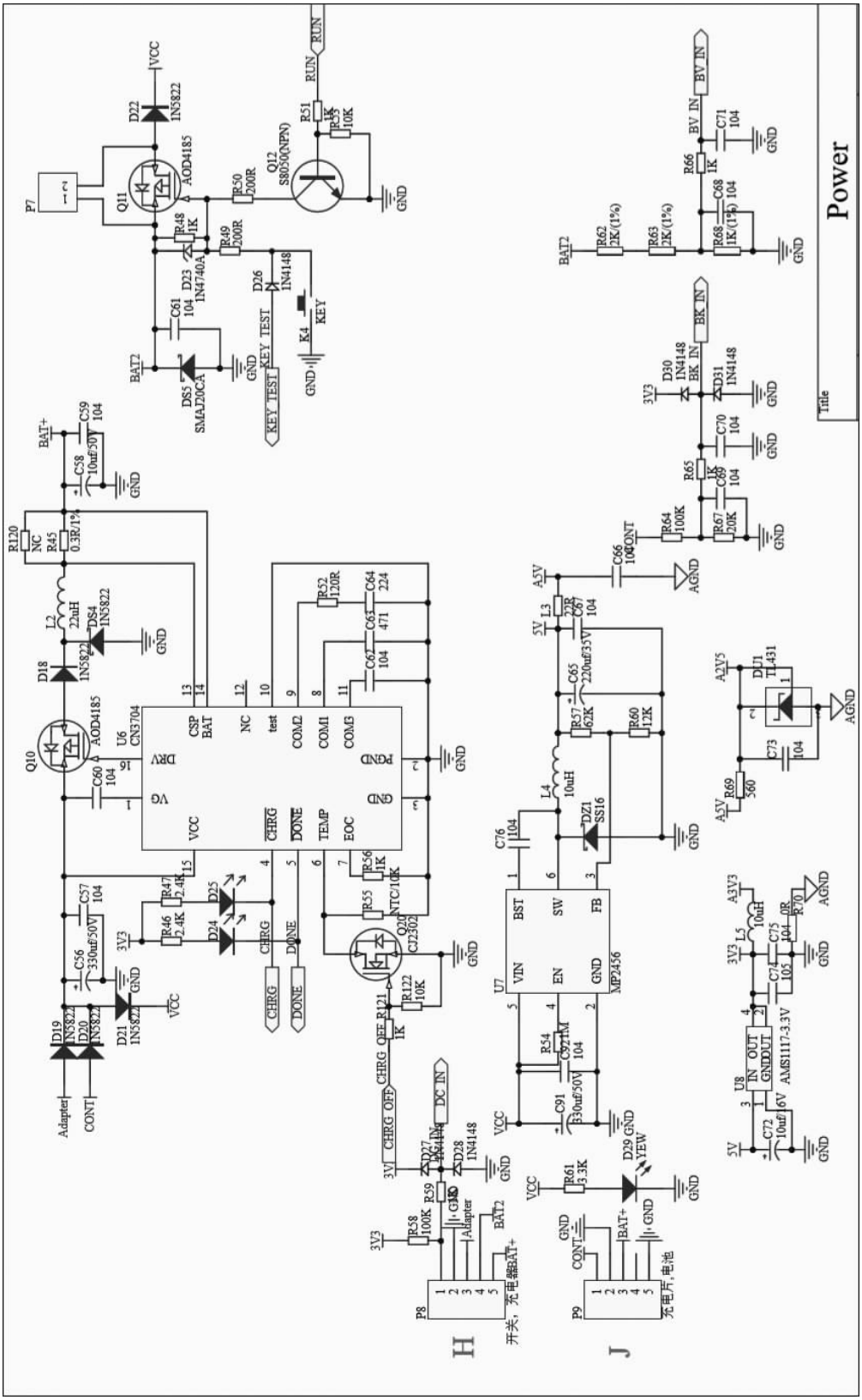


图 1-4 主控板电源原理图

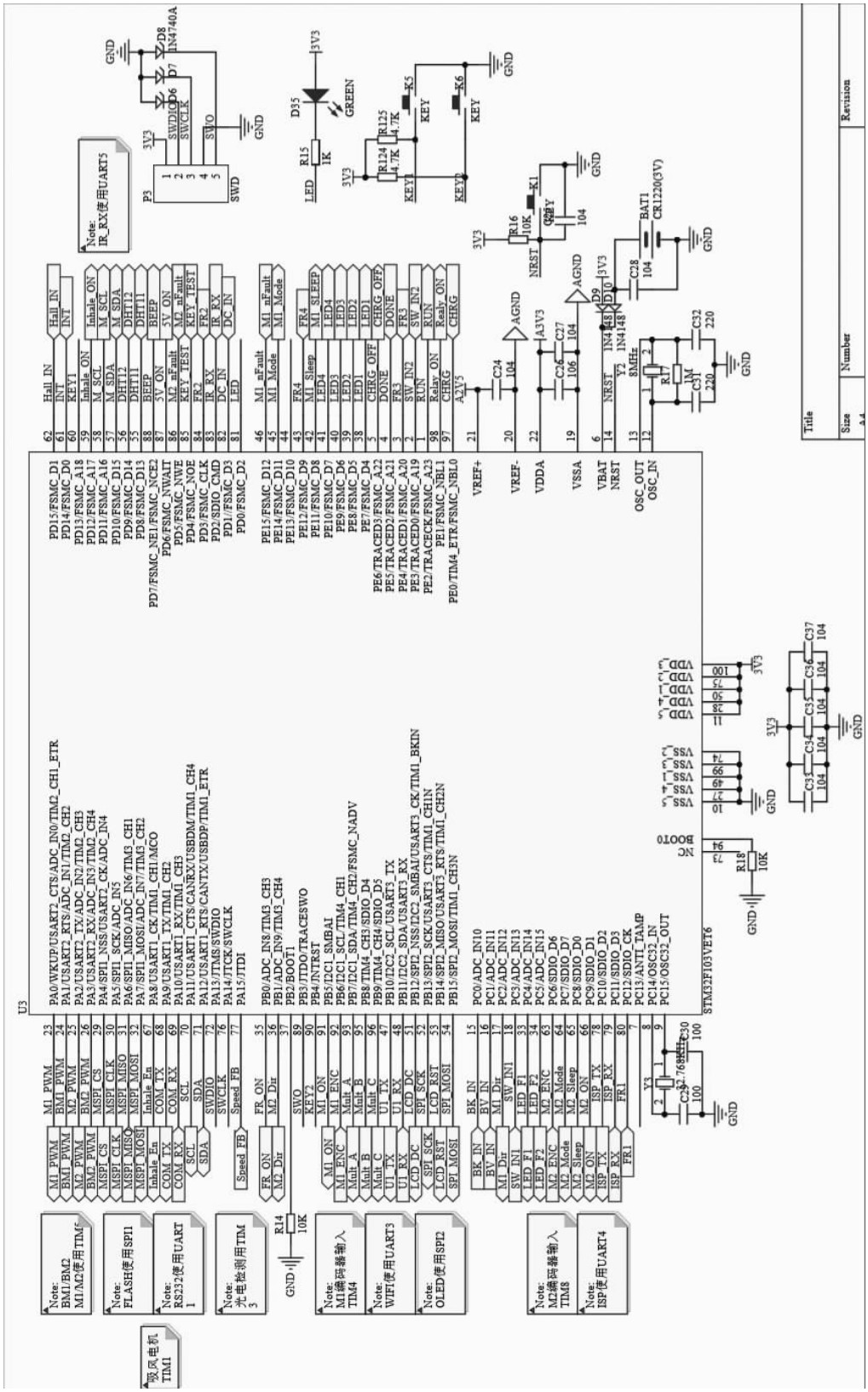

U3
STM32F103VET6
Note: BM1/BM2 M1/M2使用TIM6
Note: FLASH使用SPI1
Note: RS232使用UART 1
Note: 光电检测用TIM 3
Note: M1编码器输入 TIM4
Note: WIFI使用UART3
Note: OLED使用SPI2
Note: M2编码器输入 TIM8
Note: ISP使用UART4
吸风电机 TIM1
Note: IR_RX使用UART5
SWD
P3
LED
D35
GREEN
KEY1
KEY2
KEY
BAT1
CR1220(3V)
Title
Size
Number
Revision
A4

图 1-5　MCU 原理图

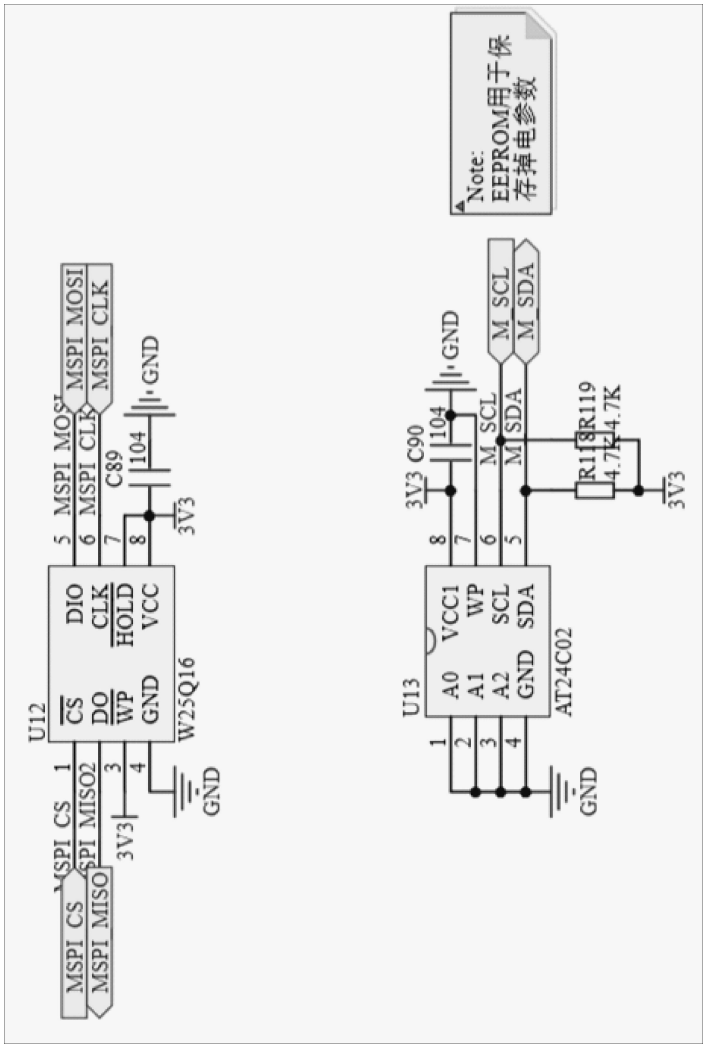


图 1-6　Flash（闪存）/EEPROM（电擦除可编程只读存储器）原理图

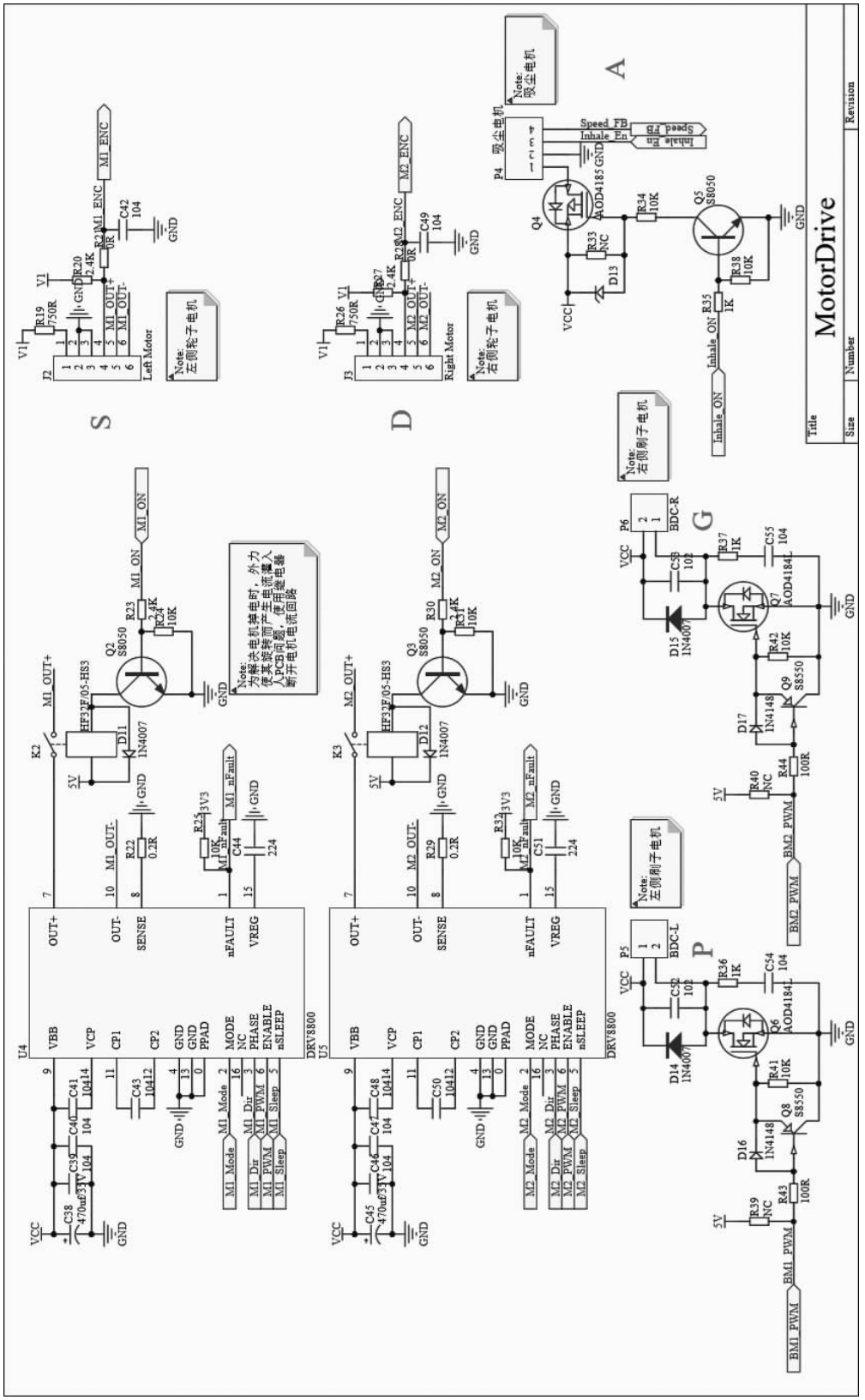

MotorDrive
Left Motor
Right Motor
DRV8800
左侧轮子电机
右侧轮子电机
吸尘电机
左侧刷子电机
右侧刷子电机
为解决电机堵电时，外力使其旋转而产生电流灌入人PCB问题，使用继电器断开电机电流回路

图 1-7　电机原理图

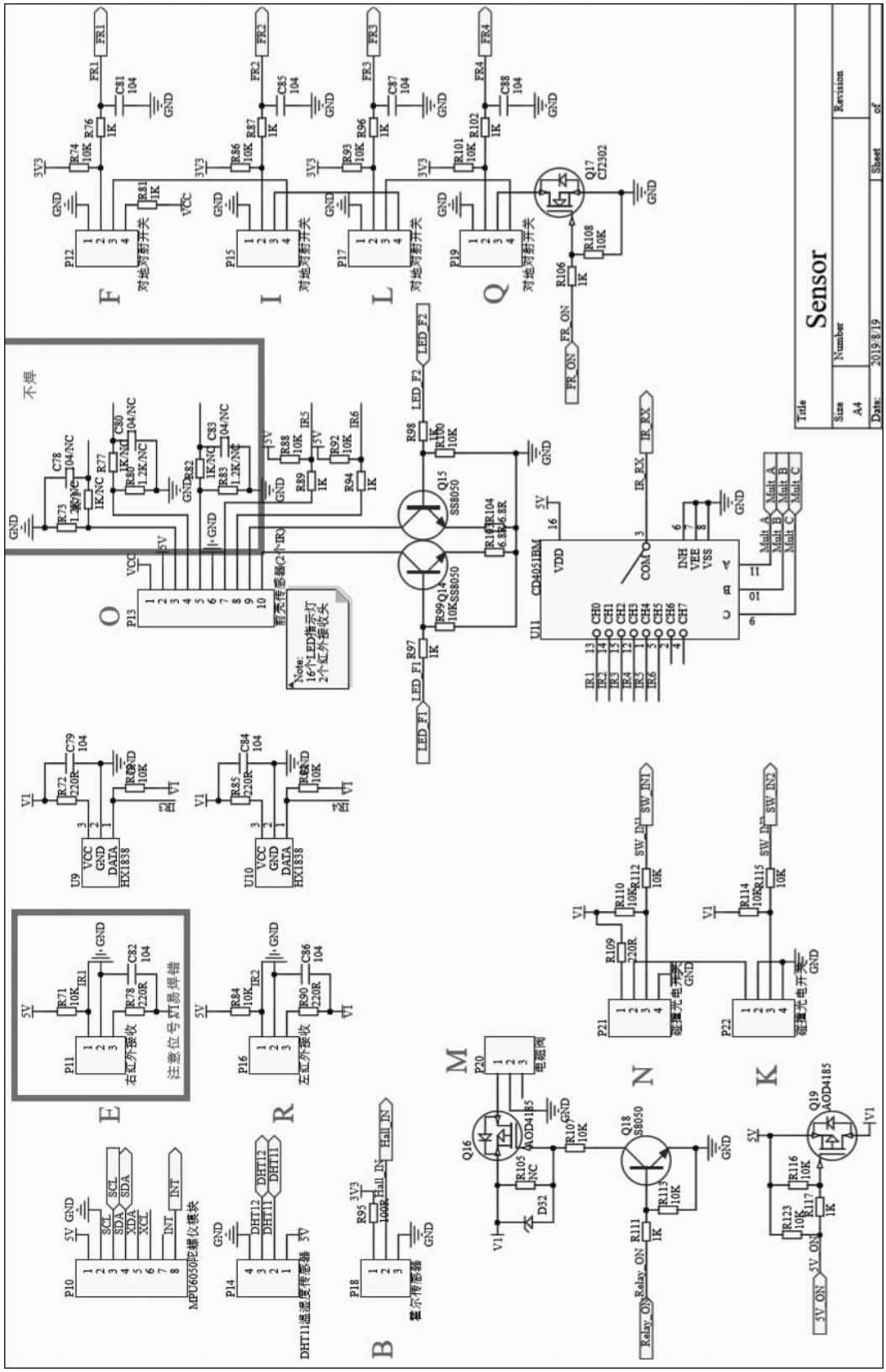

Sensor
Title
Size
A4
Number
Revision
Date: 2019/8/19
Sheet
of
F
I
L
Q
O
E
R
B
M
N
K
对地对射开关
不焊
注意位号对易焊错
右红外接收
左红外接收
MPU6050陀螺仪模块
DHT11温湿度传感器
霍尔传感器
电磁阀
磁控光电开关
Note:
16个LED指示灯
2个红外接收头
CD4051BM
HX1838
SS8050
AOD4185
Q17
C7202
FR_ON
IR_RX
LED_F1
LED_F2
Relay_ON
5V_ON
Hall_IN
SW_IN1
SW_IN2
Mult_A
Mult_B
Mult_C

图 1-8 传感器原理图

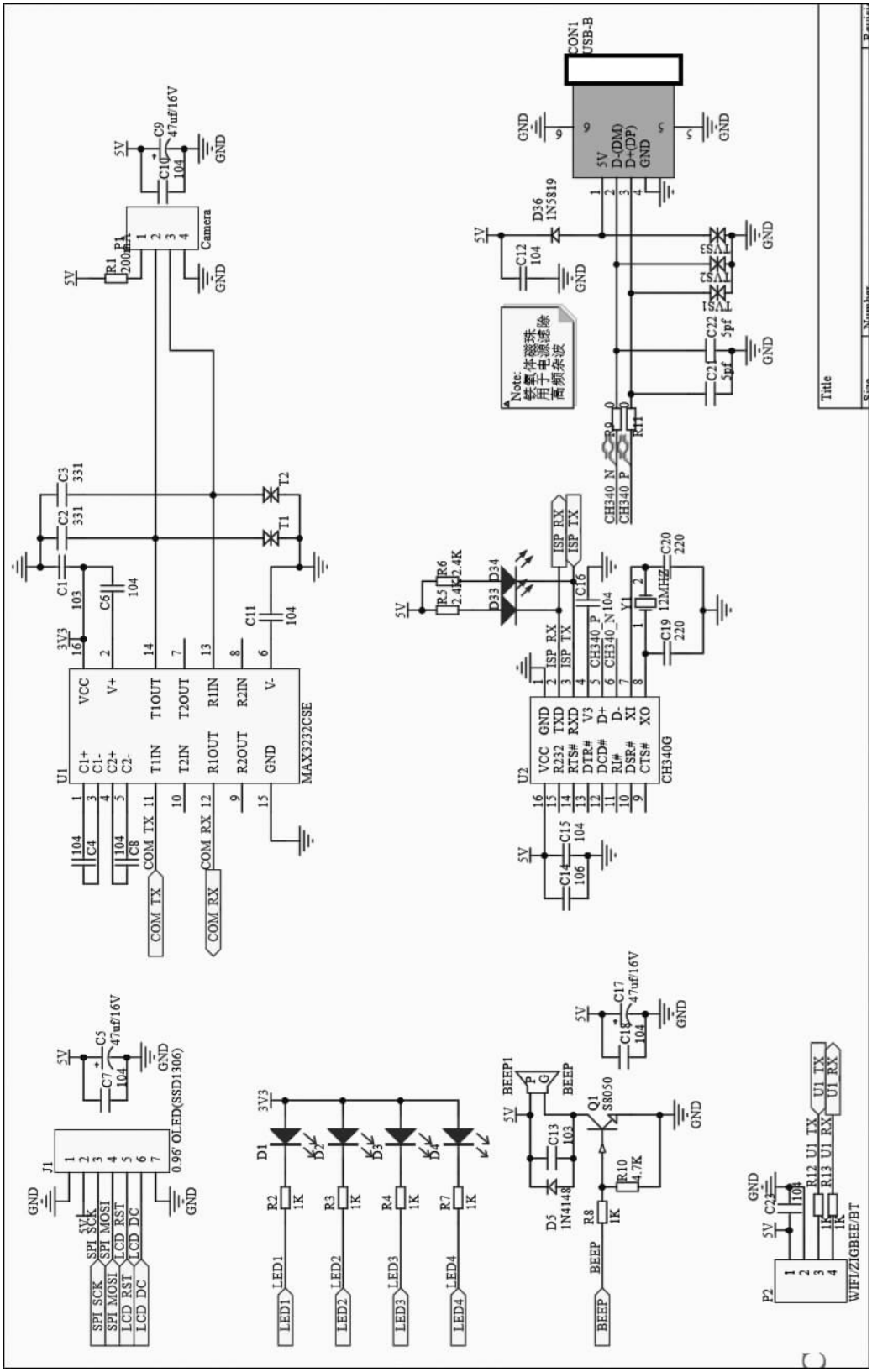

U1
MAX3232CSE
VCC
V+
C1+
C1-
C2+
C2-
T1OUT
T2OUT
T1IN
T2IN
R1OUT
R2OUT
R1IN
R2IN
GND
V-
COM_TX
COM_RX
Camera
P1
R1
200mA
C1
C2
C3
C4
C5
C6
C7
C8
C9
C10
C11
T1
T2
3V3
5V
GND
U2
CH340G
VCC
GND
R232
TXD
RXD
RTS#
V3
DTR#
D+
DCD#
D-
RI#
XI
DSR#
XO
CTS#
ISP_RX
ISP_TX
CH340_P
CH340_N
R5
2.4K
R6
2.4K
D33
D34
C14
C15
C16
C19
C20
Y1
12MHZ
CON1
USB-B
D-(DM)
D-(DP)
D36
1N5819
C12
C21
C22
5pf
R9
R11
CH340_N
CH340_P
Note:
Title
J1
0.96' OLED(SSD1306)
SPI_SCK
SPI_MOSI
LCD_RST
LCD_DC
D1
D2
D3
D4
R2
R3
R4
R7
1K
LED1
LED2
LED3
LED4
BEEP1
BEEP
C13
103
D5
1N4148
R8
R10
4.7K
Q1
S8050
C17
47uF/16V
C18
P2
WIFI/ZIGBEE/BT
U1_TX
U1_RX
R12
R13
C22

图 1-9　外围接口原理图

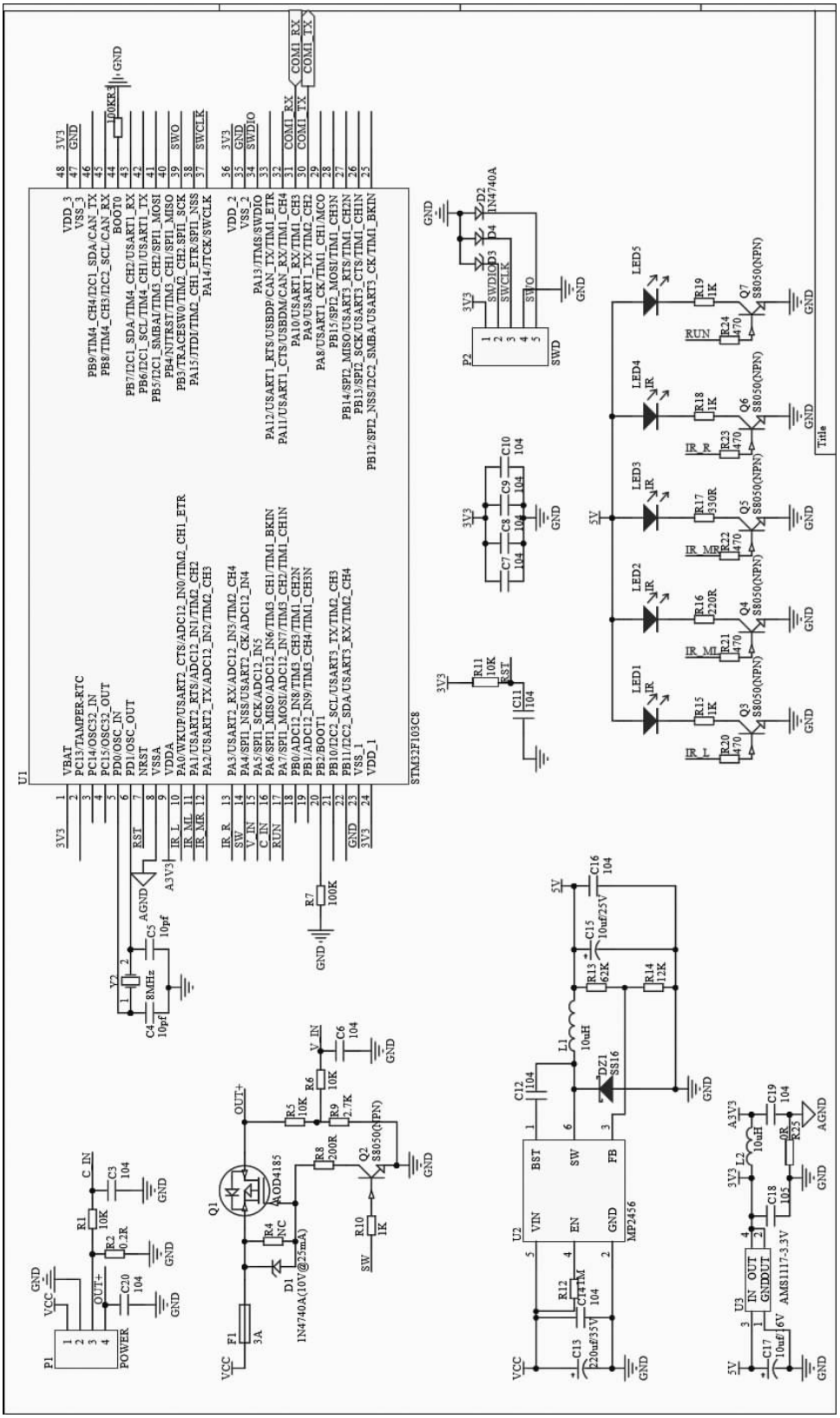

U1
STM32F103C8
VBAT
PC13/TAMPER-RTC
PC14/OSC32_IN
PC15/OSC32_OUT
PD0/OSC_IN
PD1/OSC_OUT
NRST
VSSA
VDDA
PA0/WKUP/USART2_CTS/ADC12_IN0/TIM2_CH1_ETR
PA1/USART2_RTS/ADC12_IN1/TIM2_CH2
PA2/USART2_TX/ADC12_IN2/TIM2_CH3
PA3/USART2_RX/ADC12_IN3/TIM2_CH4
PA4/SPI1_NSS/USART2_CK/ADC12_IN4
PA5/SPI1_SCK/ADC12_IN5
PA6/SPI1_MISO/ADC12_IN6/TIM3_CH1/TIM1_BKIN
PA7/SPI1_MOSI/ADC12_IN7/TIM3_CH2/TIM1_CH1N
PB0/ADC12_IN8/TIM3_CH3/TIM1_CH2N
PB1/ADC12_IN9/TIM3_CH4/TIM1_CH3N
PB2/BOOT1
PB10/I2C2_SCL/USART3_TX/TIM2_CH3
PB11/I2C2_SDA/USART3_RX/TIM2_CH4
VSS_1
VDD_1
VDD_3
VSS_3
PB9/TIM4_CH4/I2C1_SDA/CAN_TX
PB8/TIM4_CH3/I2C2_SCL/CAN_RX
BOOT0
PB7/I2C1_SDA/TIM4_CH2/USART1_RX
PB6/I2C1_SCL/TIM4_CH1/USART1_TX
PB5/I2C1_SMBAI/TIM3_CH2/SPI1_MOSI
PB4/NJTRST/TIM3_CH1/SPI1_MISO
PB3/TRACESWO/TIM2_CH2/SPI1_SCK
PA15/JTDI/TIM2_CH1_ETR/SPI1_NSS
PA14/TCK/SWCLK
VDD_2
VSS_2
PA13/JTMS/SWDIO
PA12/USART1_RTS/USBDP/CAN_TX/TIM1_ETR
PA11/USART1_CTS/USBDM/CAN_RX/TIM1_CH4
PA10/USART1_RX/TIM1_CH3
PA9/USART1_TX/TIM1_CH2
PA8/USART1_CK/TIM1_CH1/MCO
PB15/SPI2_MOSI/TIM1_CH3N
PB14/SPI2_MISO/USART3_RTS/TIM1_CH2N
PB13/SPI2_SCK/USART3_CTS/TIM1_CH1N
PB12/SPI2_NSS/I2C2_SMBA/USART3_CK/TIM1_BKIN
P1
POWER
VCC
OUT+
C_IN
R1 10K
R2 0.2R
C3 104
C20 104
GND
Q1
AOD4185
F1 3A
D1 1N4740A(10V@25mA)
R4 NC
R5 10K
R6 10K
R8 200R
R9 2.7K
R10 1K
Q2 S8050(NPN)
V_IN
C6 104
SW
Y2 8MHz
C4 10pf
C5 10pf
AGND
A3V3
3V3
RST
IR_L
IR_ME
IR_MR
IR_R
V_IN
C_IN
RUN
R7 100K
COM1_RX
COM1_TX
R0KR3
P2
SWD
SWDIO
SWCLK
SWO
D2 1N4740A
D3
D4
C7 104
C8 104
C9 104
C10 104
R11 10K
RST
C11 104
LED1
LED2
LED3
LED4
LED5
IR
R15 1K
R16 220R
R17 330R
R18 1K
R19 1K
R20 470
R21 470
R22 470
R23 470
R24 470
Q3 S8050(NPN)
Q4 S8050(NPN)
Q5 S8050(NPN)
Q6 S8050(NPN)
Q7 S8050(NPN)
5V
U2
MP2456
VIN
EN
BST
SW
GND
FB
C12 104
L1 10uH
DZ1 SS16
R12 CT+1M 104
C13 220uf/35V
R13 62K
R14 12K
C15 10uf/25V
C16 104
U3
AMS1117-3.3V
IN
OUT
GND/DUT
C17 10uf/6V
C18 105
L2 10uH
C19 104
R25 0R
Title

图 1-10 充电板原理图

三、软件环境介绍

1. Keil μ Vision4 软件的安装

在×××目录下运行 mdk. exe 以安装 MDK-ARM，弹出安装界面，如图 1-11 所示，单击 Next 按钮。MDK-ARM 软件开发工具的全称是 Microcontroller Development Kit，是 ARM 基于 Keil μVision 界面推出的针对 ARM7、ARM9、Cortex-M0、Cortex-M1、Cortex-M3、Cortex-R4 等 ARM 处理器的嵌入式软件开发工具。

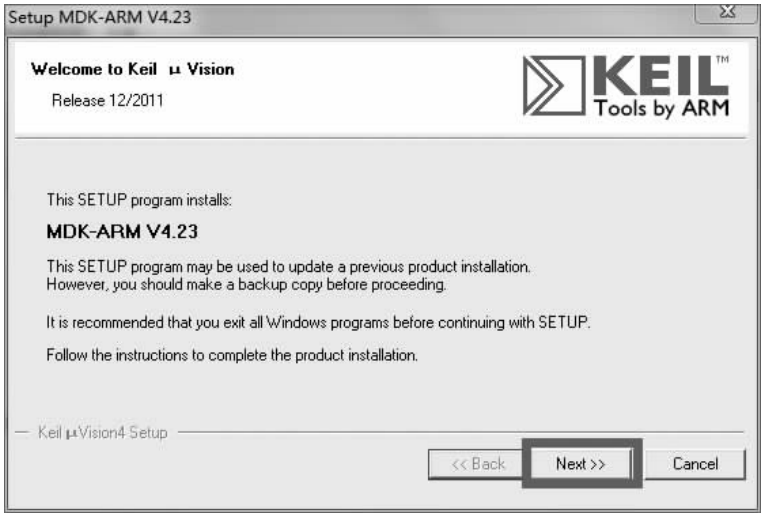


图 1-11　MDK-ARM 安装界面

选中同意安装协议复选框，然后单击 Next 按钮，如图 1-12 所示。

图 1-12　选中同意安装协议复选框

选择安装目录，然后单击 Next 按钮，如图 1-13 所示。

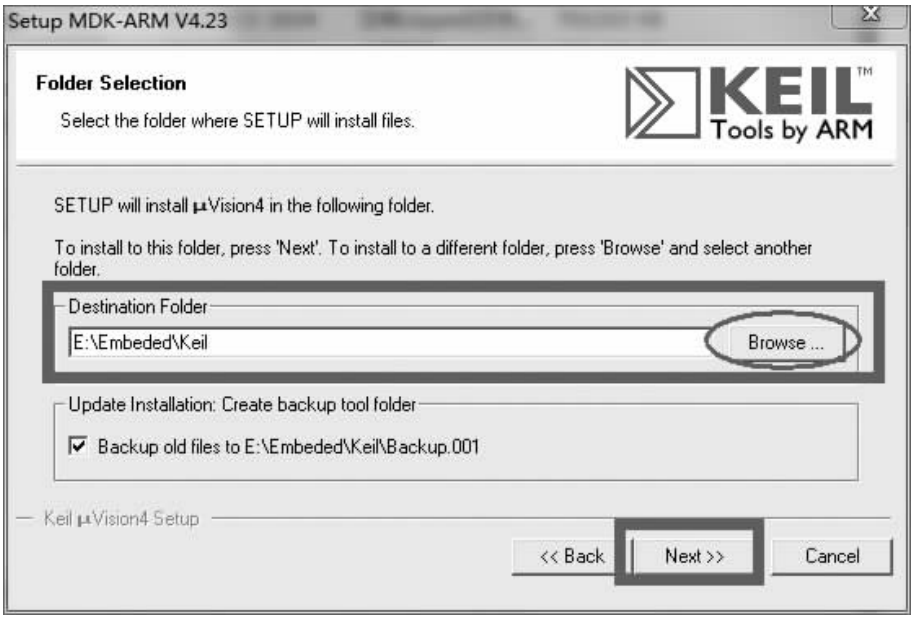


图 1-13　选择安装目录

填写用户信息，然后单击 Next 按钮，如图 1-14 所示。

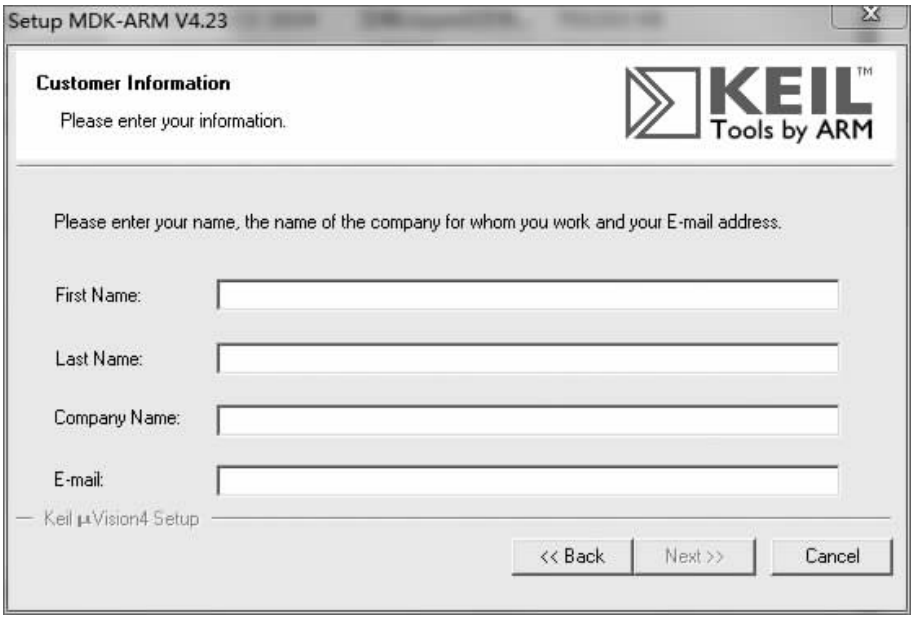


图 1-14　填写用户信息

系统显示正在安装，如图 1-15 所示。

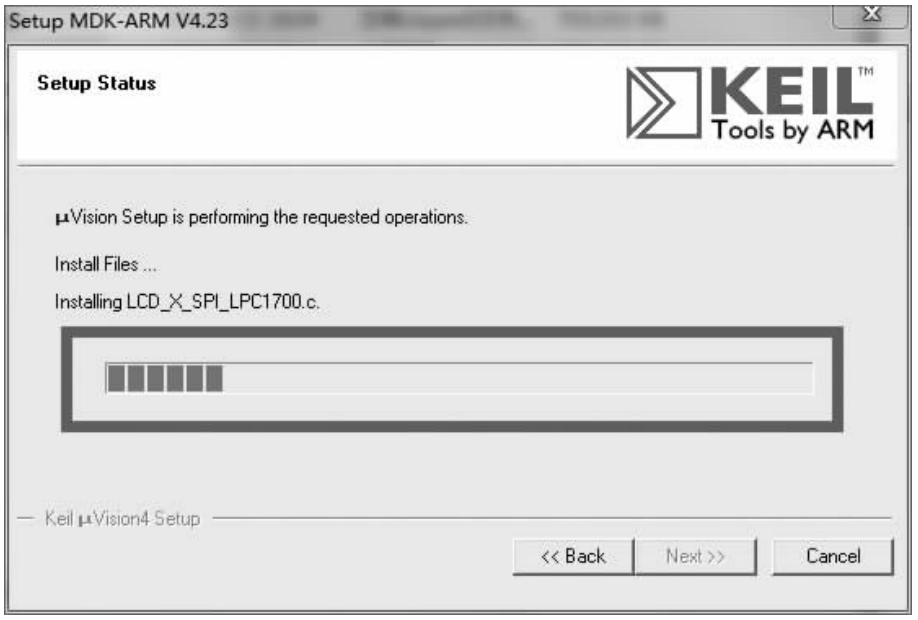


图 1-15　系统显示正在安装

选中两个复选框，单击 Finish 按钮，如图 1-16 所示。

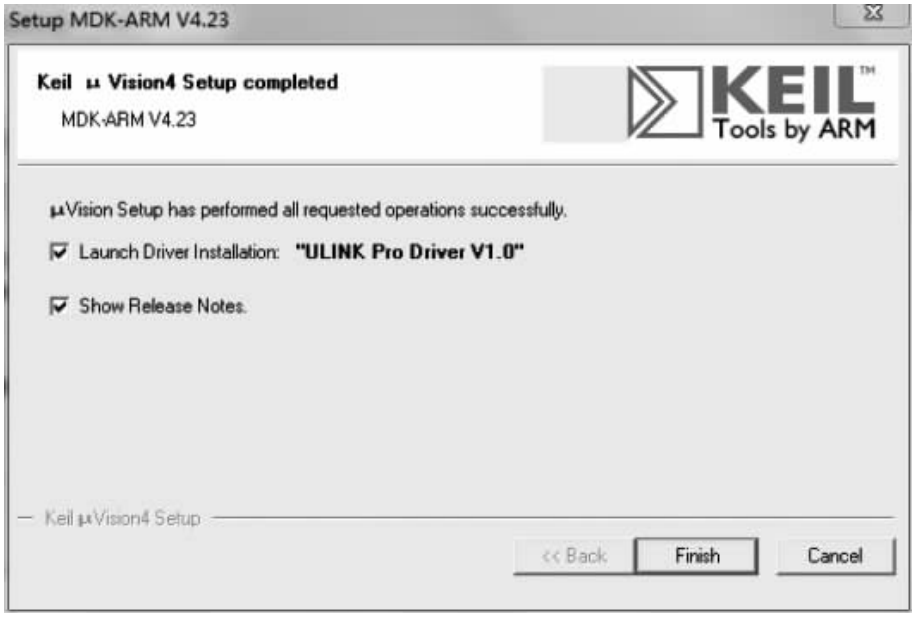


图 1-16　MDK-ARM 安装完成

此时系统自动安装 ULINK 驱动，如图 1-17 所示。ULINK 通常指 ARM 推出的嵌入式系统调试工具，主要用于连接个人计算机(PC)与 ARM 内核微控制器，可在开发过程中实现程序下载、实时调试及运行状态监控等核心功能。

图 1-17　安装 ULINK 驱动

以管理员身份运行 Keil μ Vision4，按照图 1-18、图 1-19 中的步骤①～⑦完成注册。执行“File”→“License Management”命令，复制 CID(字符识别码)，在×××文件夹下找到 License Details，粘贴 CID 并选择 ARM，然后单击 Generate 按钮生成 LIC(授权文件)，最后将生成的 LIC 复制到 License Management 中并点击 Add LIC。

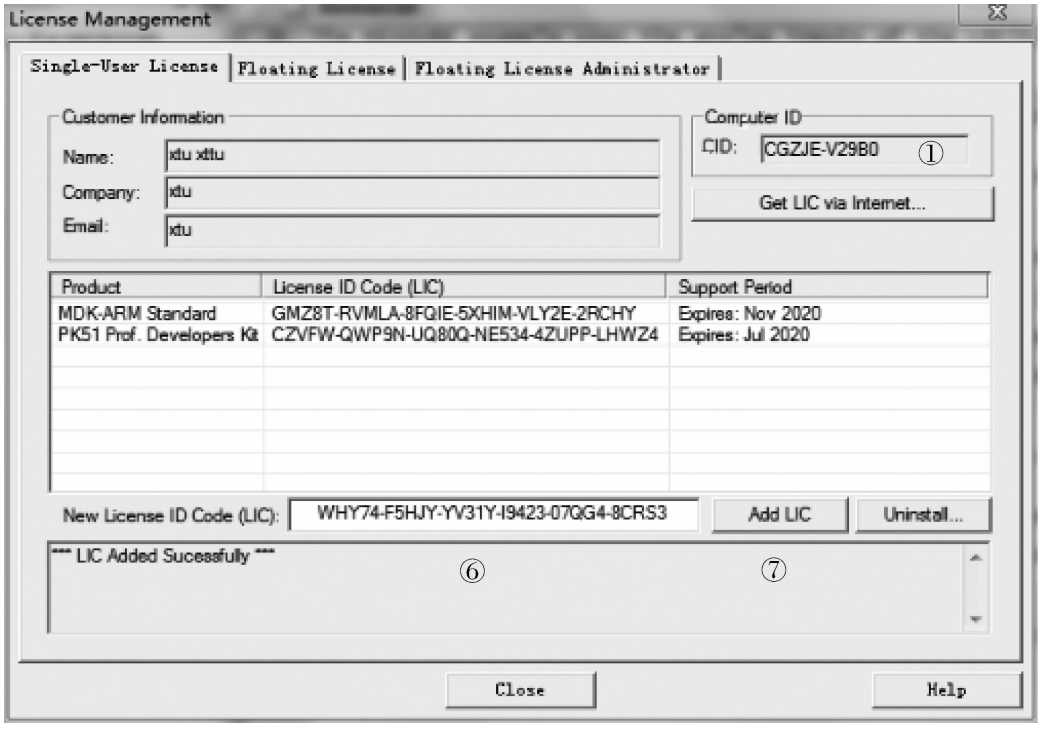


图 1-18　许可证管理

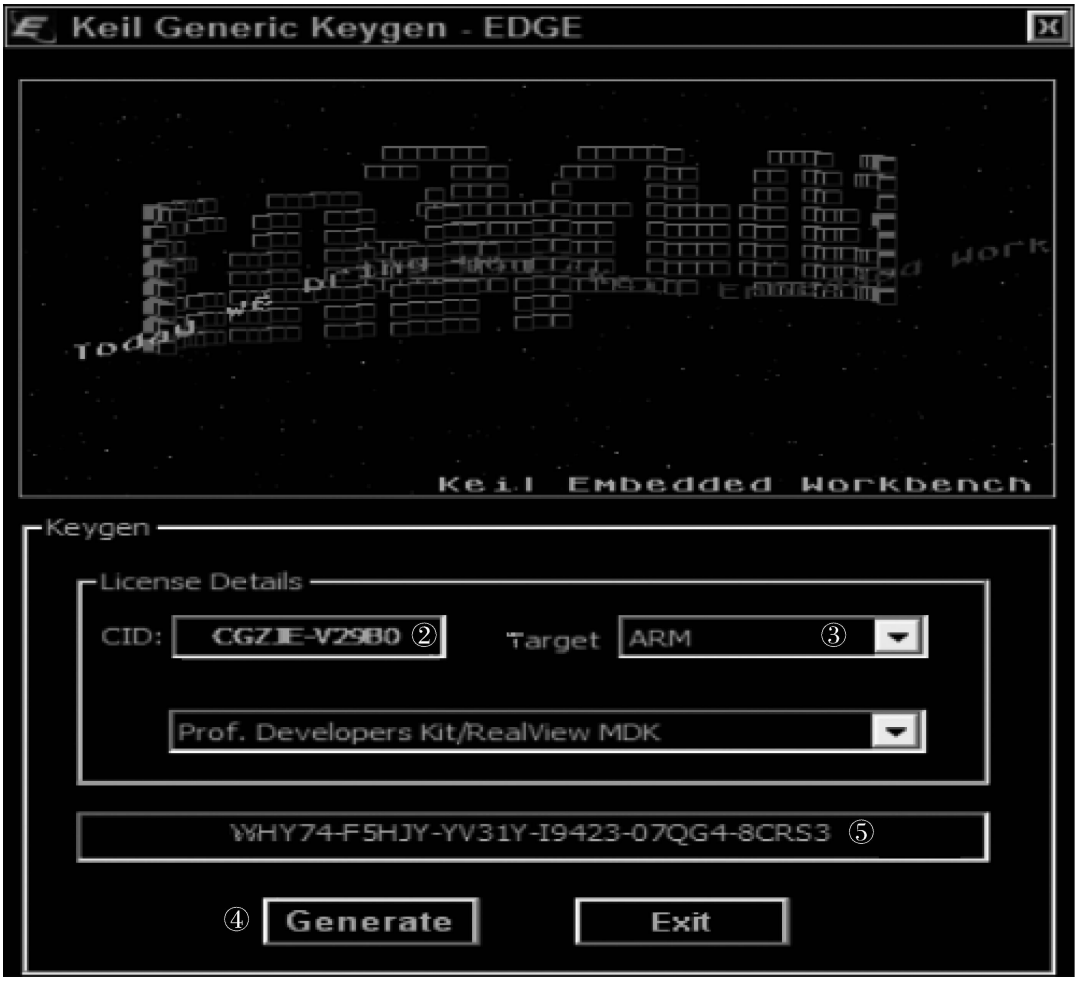


图 1-19　Keil μ Vision4 注册机

2. ST-LINK 驱动的安装

在×××文件夹下找到文件 dpinst_amd64，运行安装。DPINST_AMD64 是 64 位 Windows 系统下的驱动程序自动安装工具，其主要作用是简化 INF（安装信息文件）格式的驱动文件的安装流程，帮助用户或厂商实现驱动文件的快速自动化安装。

单击“下一步(N)”按钮，如图 1-20 所示，然后在弹出的界面单击“完成”按钮，完成安装，如图 1-21 所示。

图 1-20　安装界面

图 1-21　完成安装

安装完成后，若在计算机的设备管理器界面中出现图 1-22 所示的驱动，则表示安装完成。

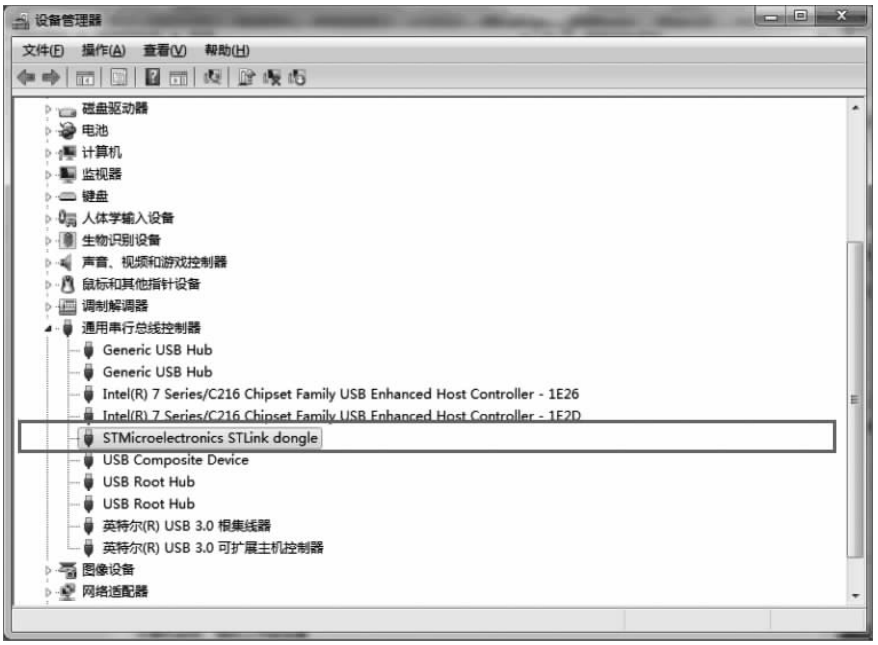


图 1-22　计算机的设备管理器界面

3. CH340G 驱动的安装

在×××文件夹下找到 CH340G 的安装文件 CH341SER. INF，单击“安装”按钮完成安装，如图 1-23 所示。CH340G 是一款 USB 总线适配器芯片，其核心功能是实现 USB 接口与其他通信接口的转换，主要用于解决计算机与串口设备（如单片机、传感器等）之间的通信问题。

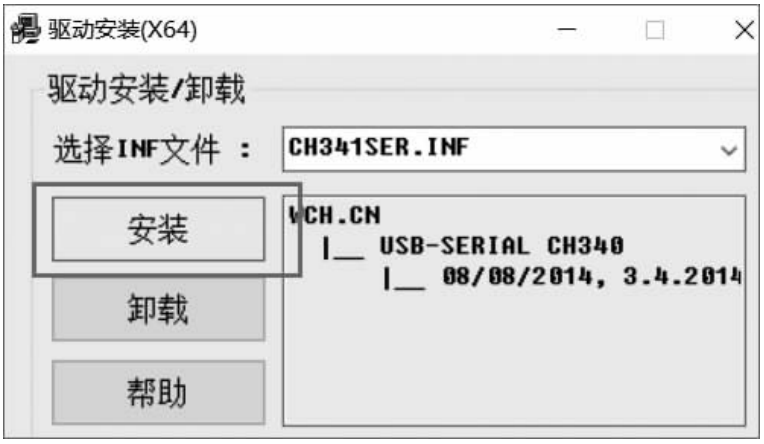


图 1-23　安装 CH340G 驱动

4. Keil μ Vision4 开发环境应用

在安装完 Keil μ Vision4 后，就可以创建工程和配置 MDK(微控制器开发工具)仿真环境了。

1) 新建工程

在桌面创建 TEST 文件夹，然后在 TEST 文件夹中创建以下文件夹，如图 1-24 所示。

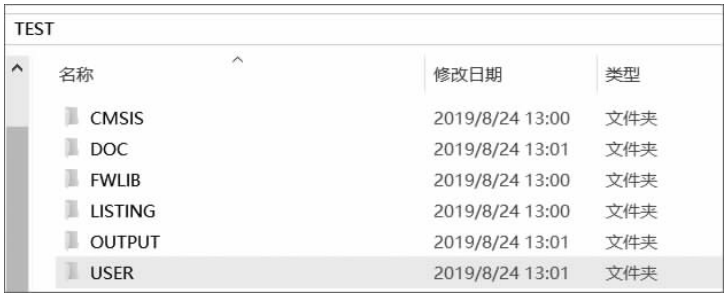


图 1-24 在 TEST 文件夹中创建文件夹

启动 Keil μ Vision4，新建工程文件，将文件命名为 STM-DEMO，并将其保存在文件夹 USER 下。

选择芯片，系统会弹出提示框，询问是否复制 STM32 的启动代码到工程文件中，因为后面可以自己手动添加，故此处单击“否”按钮。

创建完成的工程中没有任何文件，可以把 STM32 库中的文件复制到 TEST 的这几个文件夹中。

(1)USER 中放工程文件。

(2)FWLIB 中放 STM32 库中的 inc 和 src 两个文件夹，用于设置库外驱动。

(3)CMSIS 用来存放 STM32 库自带的启动文件和 M3 系列单片机通用的文件。

(4)OUTPUT 用来存放软件编译后的输出文件。

(5)LISTING 用来存放程序编译后生成的链接文件。

(6)DOC 用来存放与项目相关的文档资料。

回到 Keil μ Vision4 工程界面，右击 Target 1，添加 4 个组，如图 1-25 所示。

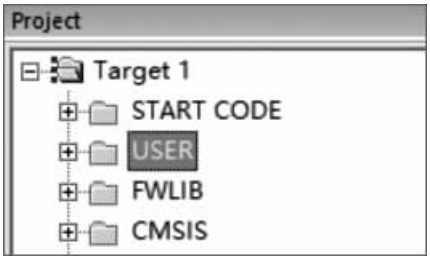


图 1-25 添加 4 个组

分别向各组添加之前导入文件夹中的库文件，如图 1-26 所示。

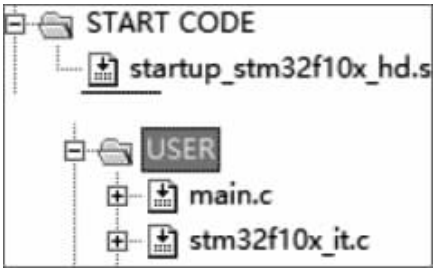


图 1-26　添加库文件

在 FWLIB 文件夹里添加 src 的全部驱动文件(.h 的头文件不用添加)，如图 1-27 所示。

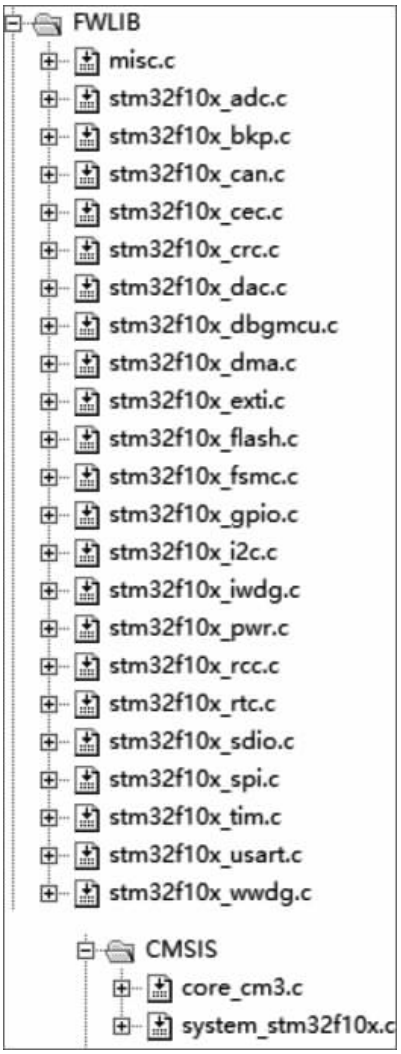


图 1-27　添加 src 的全部驱动文件

2）配置 MDK 工程环境

单击魔术棒图标，如图 1-28 所示。

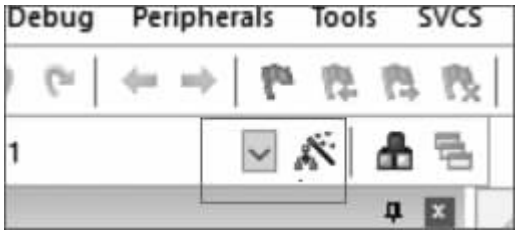


图 1-28　单击魔术棒图标

选择输出路径，如图 1-29 所示。

图 1-29　选择输出路径

选择 Listing 路径，如图 1-30 所示。

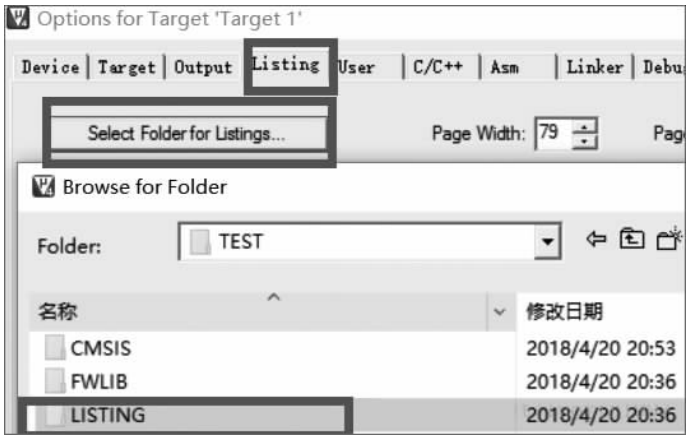


图 1-30　选择 Listing 路径

把从库函数复制的 main.c 文件里的内容全删除，然后输入以下代码。

```
{
  #include "stmf10x.h"
  int main (void)
  {while(1);}
  //add your code here ^-^。
}
```

添加宏定义 USE_STDPERIPH_DRIVER，STM32F10X_HD，如图 1-31 所示。

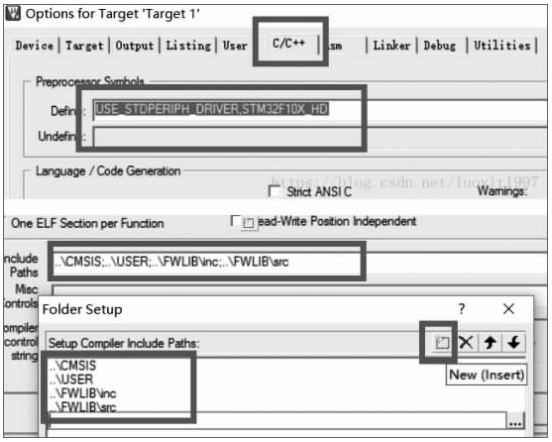


图 1-31　添加宏定义

设置 Output 选项卡，如图 1-32 所示。

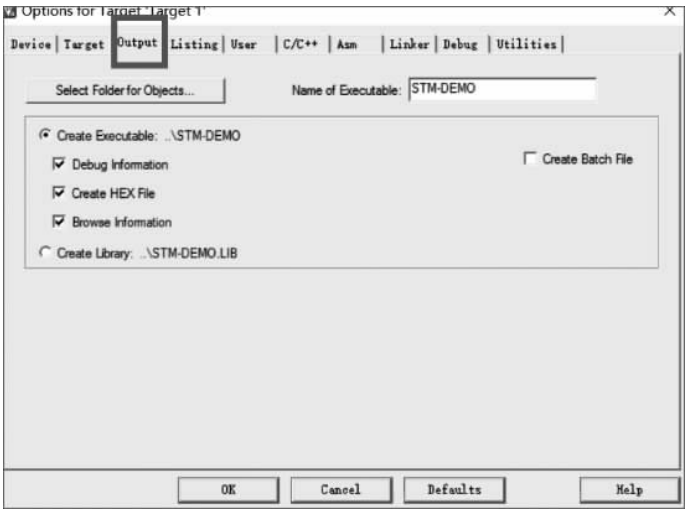


图 1-32　设置 Output 选项卡

设置 Debug 选项卡，如图 1-33 所示。

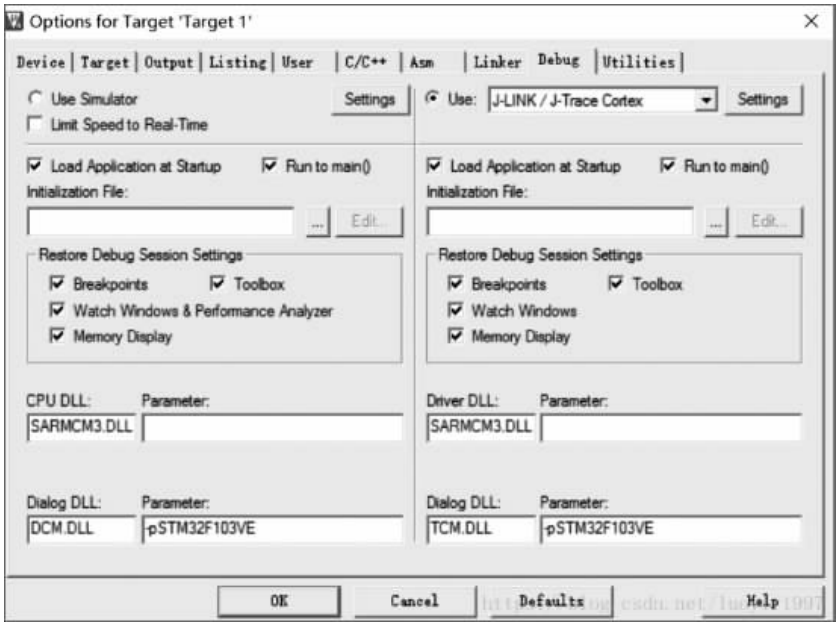


图 1-33　设置 Debug 选项卡

3）配置 ST-LINK 硬件调试

单击魔术棒图标，弹出 Target 1 的设置界面，切换到 Debug 选项卡，单击“Settings”按钮，如图 1-34 所示。

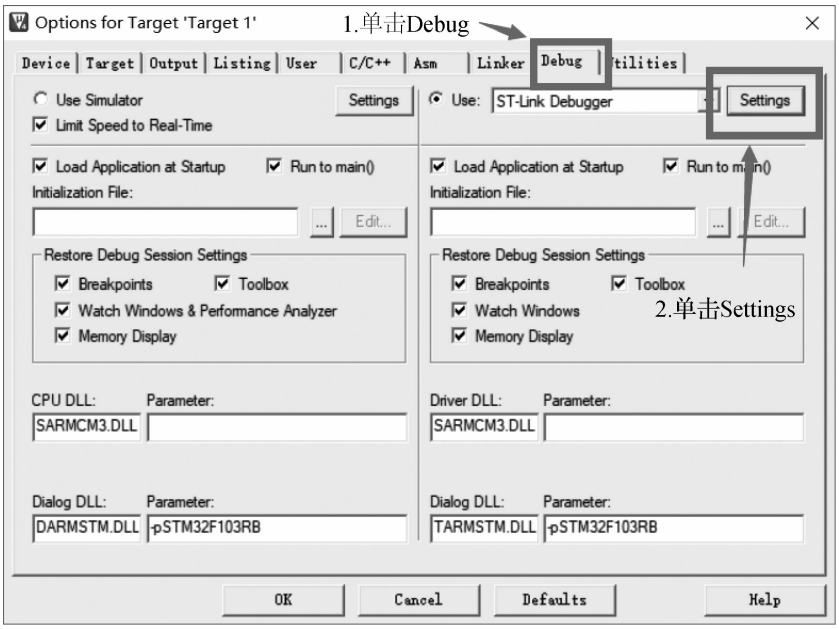


图 1-34　切换到 Debug 选项卡

在弹出的下拉列表中选择调试工具，如图 1-35 所示。

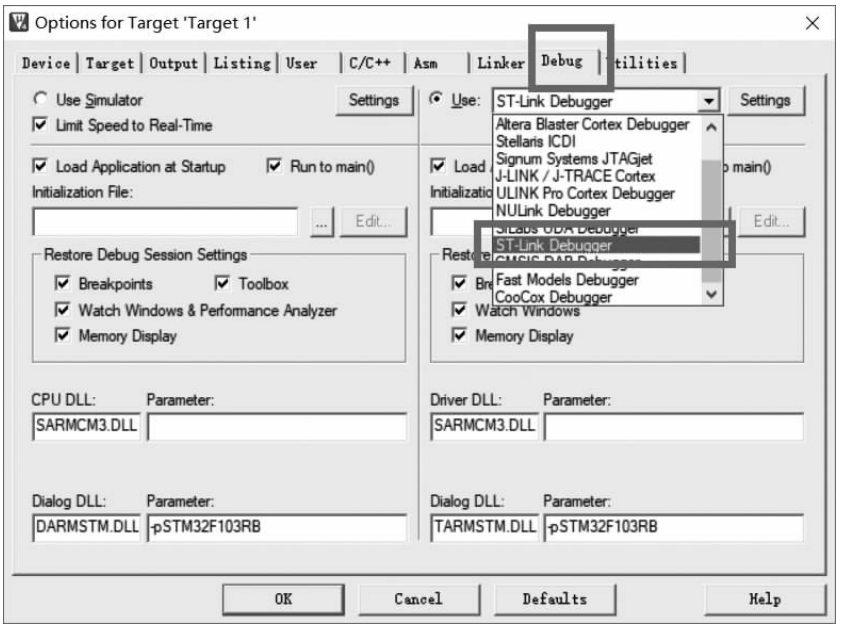
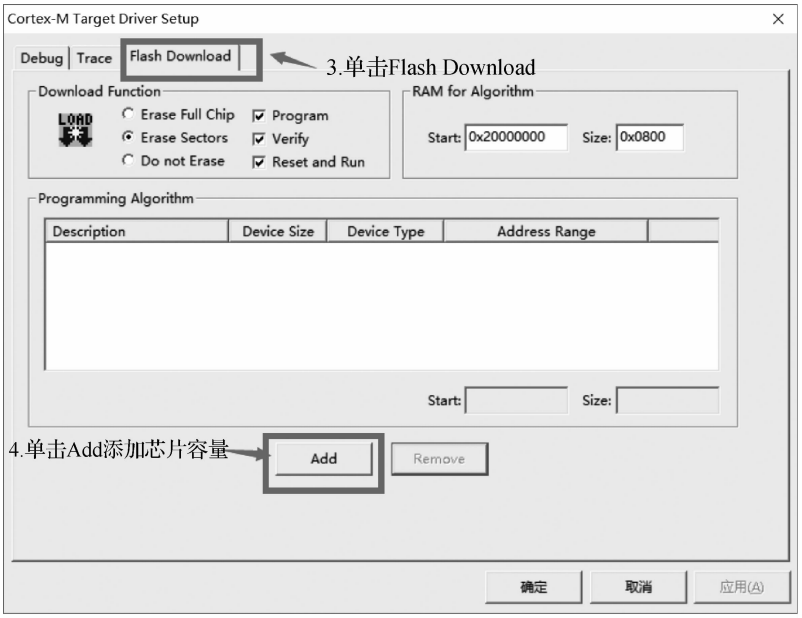


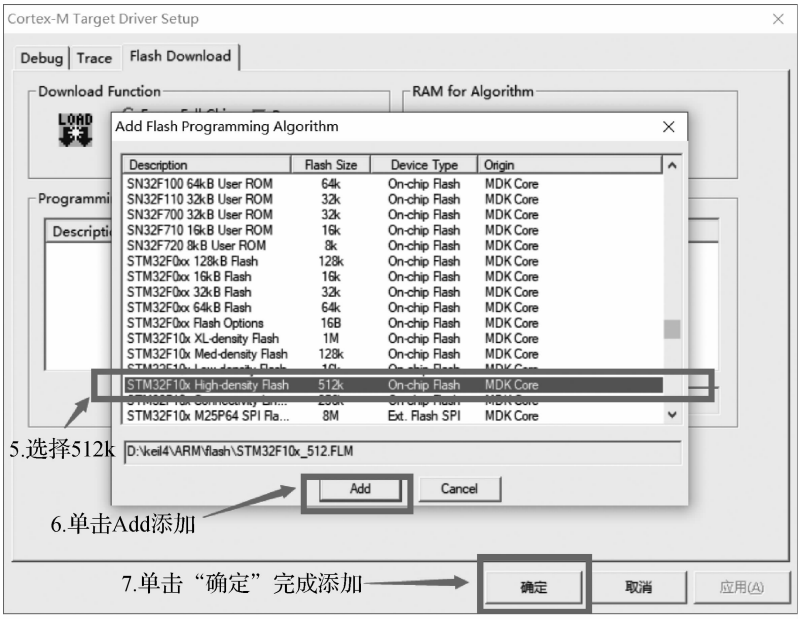
图 1-35　选择调试工具

扫地机器人主控板采用的是大容量芯片，因此需要修改芯片容量。切换到 Flash Download 选项卡，单击“Add”按钮，如图 1-36(a)所示。在弹出的芯片容量选择对话框中选择 512 k 的芯片，单击“Add”按钮，如图 1-36(b)所示。



(a)

图 1-36　芯片容量选择



(b)

续图 1-36　芯片容量选择

5. 连接 ST-LINK

1）编译程序

编译程序的模式有以下 3 种：

translate：编译当下文件，不链接库文件，也不生成任何可执行文件。

build：编译当下修改过的文件，做语法检查，链接动态库，生成可执行文件（常用）。

rebuild：重新编译整个工程，耗时巨大。

程序编译完成后编译栏提示“0 Error(s)，0 Warning(s)”，如图 1-37 所示。这表示工程创建成功，程序编译通过。

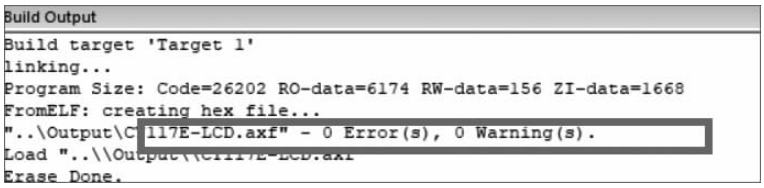


图 1-37　程序编译结果

2）烧写程序

接通主控板电源并连接 ST-LINK。

单击 LOAD 图标，如图 1-38 所示。

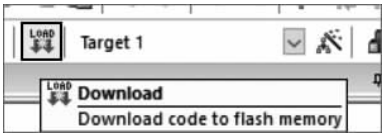


图 1-38　单击 LOAD 图标

程序下载成功后弹出提示框，如图 1-39 所示。

```
Erase Done.
Programming Done.
Verify OK.
Application running ...
```

图 1-39　程序下载成功后弹出提示框

注：Keil μ Vision4 工程创建过程比较复杂且容易出错，TEST 文件夹提供了工程模板，编写程序时可以直接使用工程模板，无须再创建工程和进行相关配置。

项目二 LED灯设计

项目目的

(1)了解 STM32F1 系列单片机的基本结构。

(2)了解 GPIO(通用输入/输出)口的基本特性和使用方法。

(3)掌握位操作知识和使用方法。

项目准备

(1)扫地机器人主控板一块。

(2)ST-LINK 下载器一个。

项目分析

本项目通过 2 个任务简要介绍 LED 的一些基础知识和开发工具。通过本项目的学习，学生能够初步了解和掌握 STM32F1 系列单片机 GPIO 口的特性和使用方法，为后续项目的学习打下坚实的基础。

本项目的教学情境设计见表 2-1。

表 2-1　项目二教学情境设计

序号	任务名称	任务内容	知识要点
任务一	LED 闪烁灯	利用 GPIO 口 控制 LED 灯闪烁	GPIO 口的基本特性 扫地机器人主控板 GPIO 口的使用方法
任务二	LED 流水灯	利用位操作 实现 LED 流水灯效果	位操作原理

支撑知识

一、LED 介绍

LED 是一种能够将电能转化为可见光的固态的半导体器件。半导体器件由两部分组

成：一部分是 P 型半导体，其空穴占主导地位；另一部分是 N 型半导体，其电子占主导地位。将这两种半导体连接起来，它们之间就形成了一个 PN 结。当电流流过这个半导体器件时，电子就会被推向 P 区，在 P 区里电子跟空穴结合，然后就会以光子的形式发出能量，这就是 LED 发光的原理。不同波长的光波，光的颜色也不同，LED 的发光颜色是由形成 PN 结的材料决定的。

二、GPIO 口的基本特性

GPIO 口，即 I/O 口，是嵌入式系统中用于与外部设备交互的核心接口。由于每个开发板上都会设计不同的外围电路，所以 GPIO 口的功能可能有所不同。

GPIO 口可用于单片机与各种外部设备通信，从而实现各种功能，其功能如下。

(1)基本输入输出功能。利用 I/O 输出口可控制灯光、开关，利用 I/O 输入口可监测温度、检测门窗状态等。在机器人控制系统中，GPIO 口可以用来控制电机、接收传感器数据等。

(2)中断。GPIO 口可以通过中断的方式来实现异步通信。通过 GPIO 口的电平变换可以触发一个中断，这意味着当发生中断时，单片机可以立即得到信息并进行相应的处理。

(3)复用各类通信外设功能。GPIO 口还可以配置为 SPI(串行外设接口)、IIC(集成电路总线)、UART(通用异步收发器)等通信接口，从而实现单片机与外部设备的高速数据传输。

三、移位操作知识

在单片机编程中，移位是指将数据在二进制位上进行平移的操作。移位操作可以分为逻辑移位和算术移位两种类型。

逻辑移位是当进行移位操作时，不需要考虑符号位，只关注数据二进制表示的移位方式。逻辑移位分为左移和右移两种方式。左移操作将数据的二进制位向左平移，移出的位被丢弃，右侧空出的位补零。右移操作将数据的二进制位向右平移，移出的位被丢弃，左侧空出的位补零。

算术移位是当进行移位操作时，需要考虑符号位影响的移位方式。算数移位也分为左移和右移两种方式。算术右移操作将数据的二进制位向右平移，移出的位被丢弃。对于无符号数，左侧空出的位补零；对于带符号数，左侧空出的位补符号位。例如，对二进制数 1101 进行算术右移一位操作，其结果为 1110。

移位操作在单片机编程中有广泛的应用，常见的应用场景如下。

(1)数据的位操作。通过移位操作可以方便地对数据的每一位进行读取或设置。

(2)数据的扩展或截断。通过移位操作可以将一个数据的位数进行扩展或截断。

(3)乘除法的优化。在一些特定的情况下,移位操作可以用于乘除法的优化。

(4)循环移位。循环移位可以实现数据的循环左移或循环右移。

(5)移位操作还可以与其他位操作相结合,用于实现一些特定的功能,如位反转、位交换、位取反等。

任务一 LED 闪烁灯

1.任务要求

利用扫地机器人主控板控制 4 盏 LED 灯闪烁,要求如下:4 盏 LED 灯同时点亮,点亮 1 s 后 4 盏 LED 灯同时熄灭,熄灭 1 s 后 4 盏 LED 灯再同时点亮,循环此操作实现闪烁灯效果。

2.知识链接

LED 具有单向导电性,通以 5 mA 左右电流即可发光,电流越大,其亮度越强。但若电流过大,则会烧毁 LED。一般情况下,流过 LED 的电流应控制在 3～20 mA。通常 LED 引脚上会串联电阻,目的是限制通过 LED 的电流,这类电阻也称为限流电阻。当 LED 发光时,可测得它两端的电压约为 1.7 V,这个电压又叫作 LED 的导通压降。

3.任务分析

本任务可分为 3 个模块:主模块(main 函数)、LED 灯模块(初始化、闪烁函数)、delay 延时模块(延时函数)。main 函数作为总函数,用于调用 LED 灯模块和 delay 模块。LED 灯函数中包含两个子模块:led_init 和 led_on_all。led_init 用来初始化点亮 LED 灯需要用到的引脚,led_on_all 用于控制 GPIO 口以实现 LED 灯的闪烁效果。delay 函数用来控制 LED 灯延时闪烁。

4.任务步骤

1)硬件连接

根据 MCU 原理图(见图 1-5)进行硬件连接,连接方式如下。

(1)LED1 连接 GPIOE_7(PE7)引脚。

(2)LED2 连接 GPIOE_8(PE8)引脚。

(3)LED3 连接 GPIOE_9(PE9)引脚。

(4)LED4 连接 GPIOE_10(PE10)引脚。

2)软件编程

(1)编写 LED 灯初始化、闪烁函数,代码如下:

```
#include "led.h"
#include "stm32f10x.h"
/*****************************************************************
```

```
    * 函数名:led_init
    * 描述:配置 LED 灯用到的 I/O 口
    * 入口参数:无
    * 出口参数:无
    * 调用方式:led_init();
    ********************************************************************/
    void led_init(void)
    {
      GPIO_InitTypeDef GPIO_InitStructure;  // 定义结构体
      RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOE, ENABLE);  // 开启 LED 灯的外
设时钟
      GPIO_InitStructure.GPIO_Pin = GPIO_Pin_7 | GPIO_Pin_8 | GPIO_Pin_9 |GPIO_
Pin_10;
      GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
      GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
      GPIO_Init(GPIOE, &GPIO_InitStructure);  //调用库函数初始化 LED 灯
    }
    /********************************************************************
    * 函数名:led_on_all
    * 描述:同时点亮 4 盏 LED 灯
    * 入口参数:无
    * 出口参数:无
    * 调用方式:led_on_all();
    ********************************************************************/
    void led_on_all(void)
    {
      GPIO_ResetBits(GPIOE, GPIO_Pin_7 | GPIO_Pin_8 | GPIO_Pin_9 |GPIO_Pin_
10);  // 点亮 4 盏 LED 灯
    }
    /********************************************************************
    * 函数名:led_off_all
    * 描述:同时熄灭 4 盏 LED 灯
    * 入口参数:无
```