**计算机科学与工程学院实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **实验课程名称** | | **信息技术基础实训-树莓派** | | | **实验总成绩** |  |
| **专业** | **计算机科学与技术** | | **班级** | **\*** | **指导教师签字** |  |
| **学号** | **\*** | | **姓名** | **\*** | **实验报告批改时间** |  |
| **实验报告分项成绩**   |  |  |  | | --- | --- | --- | | **序号** | **实验项目** | **成绩** | | **1** | **树莓派应用实践** |  | | **2** | **树莓派系统安装** |  | | **3** | **基于树莓派的程序设计** |  | | **4** |  |  | | **5** |  |  | | | | | | | |
| 通过此次信息技术基础实训—树莓派的学习，我了解到了树莓派的相关知识，以及在Linux环境下运行简单的C语言程序，学习了树莓派GPIO接口和传感器的基本使用方法，并能够通过执行命令控制小灯的闪烁方式。  相较上一次实训内容，本次实训课没有过多困难需要解决，主要困难点即为在连线树莓派与Led小灯的过程较为艰难，经历了几次小灯泡接线后不亮的问题。  此次课程收获颇多，内容较为容易有趣，希望能开展更多类似于此的课程。 | | | | | | |

**实验一**

1. **实验目的**

了解树莓派基础知识以及各种用途，如代替日常桌面计算机文字处理、电子表格等，了解树莓派与Linux系统的关系，以及树莓派的其他功能、存储方式。

（1）从零开始，一步步地下载所需软件

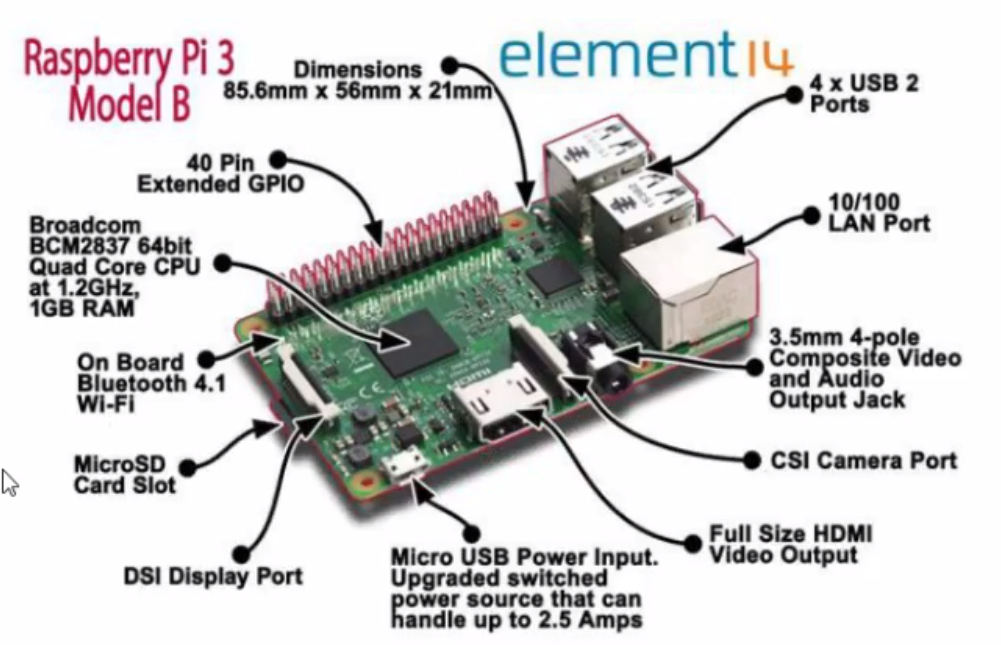
（2）烧录树莓派系统、正确安装Raspberry所需软件并配置基本环境

（3）使用简单的Linux命令安装并更新需要的软件，如音乐播放器、文字处理等软件

（4）在Linux环境下使用命令编辑、编译和调试简单的C语言程序

1. **实验原理**

树莓派端口



USB、网口、音频输出(无输入)、HDMI视频、40个GPIO接口、CPU、SD

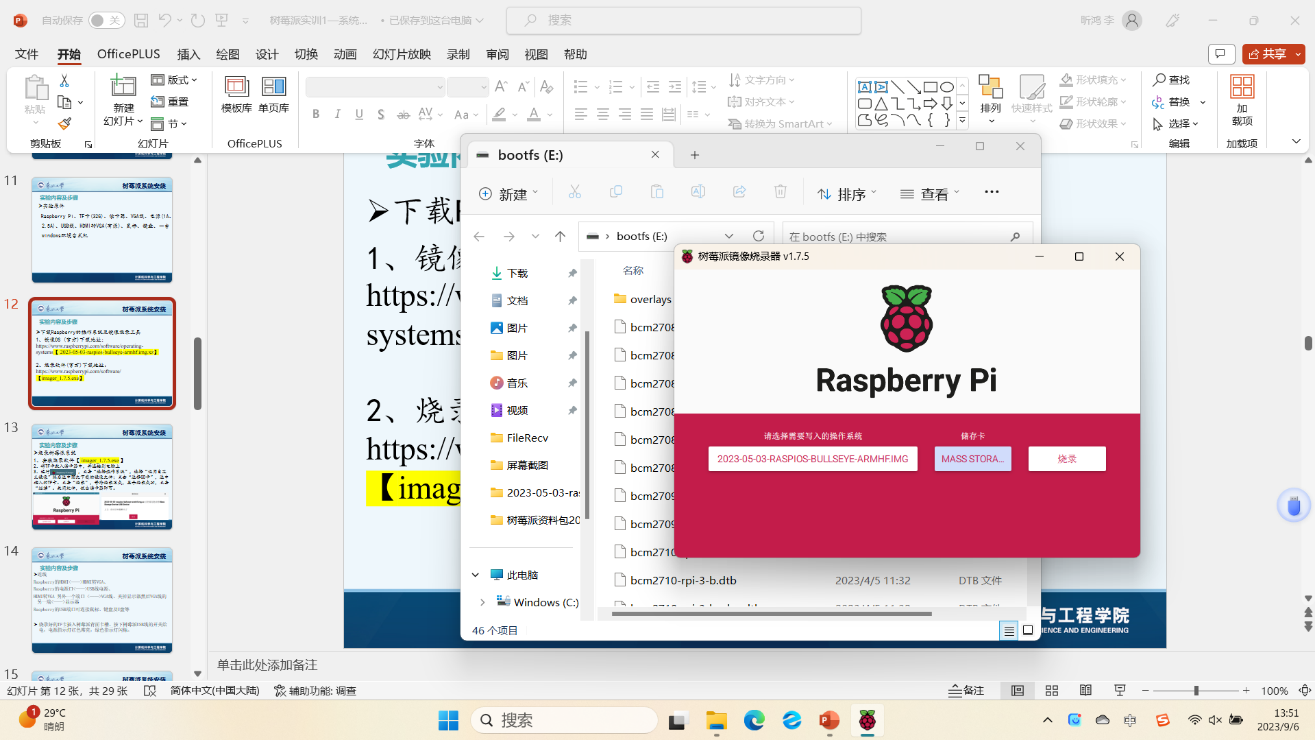
1. **实验内容**

实验原件：

Raspberry Pi、TF卡(32G)、读卡器、VGA线、电源(1A、2.5A)、USB线、HDMI转VGA(有源)、鼠标、键盘、一台windows环境台式机

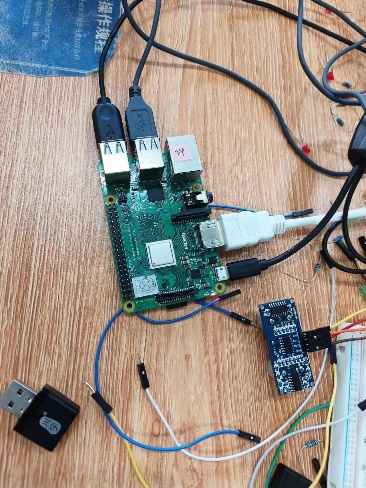
实验步骤：

1. 下载Raspberry的操作系统及镜像烧录工具，烧录树莓派系统。
2. 安装烧录软件【imager\_1.7.5.exe】，将TF卡放入读卡器中，并连接到电脑上，运行点击“选择操作系统”，选择“使用自定义镜像”然后选中预先下载的镜像文件；点击“选择SD卡”，选中插入的TF卡。点击“烧录”，等待烧录完成，显示烧录成功，点击“继续”。关闭软件，拔出读卡器即可。



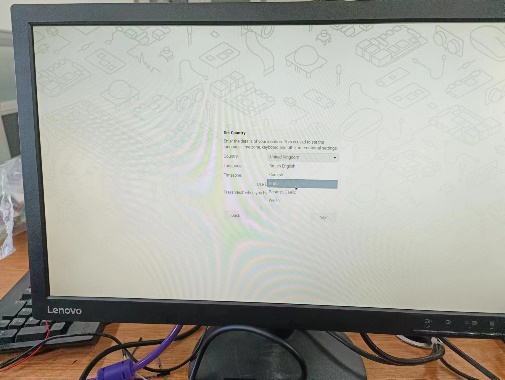
下载镜像文件 插入TF卡 烧录

1. 连线：Raspberry的HDMI<——>HDMI转VGA、 Raspberry的电源口<——>USB线电源、 HDMI转VGA 另外一个端口 <——>VGA线、关掉显示器然后VGA线的另一端<——>显示器Raspberry的USB端口可连接鼠标、键盘及U盘等。



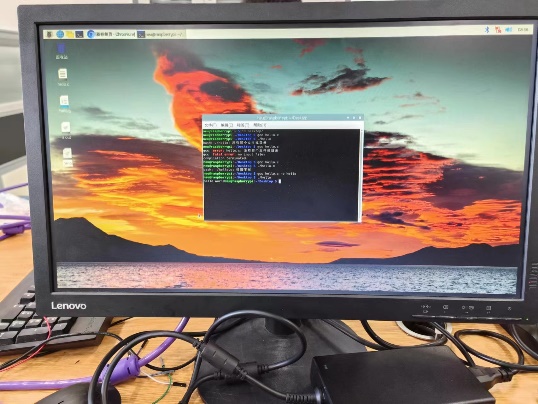
连线HDMI<——>HDMI转VGA、 电源口<——>USB线电源、 HDMI转VGA 另外一个端口 <——>VGA线、显示器连VGA线的另一端<——>显示器Raspberry的USB端口连接鼠标、键盘及U盘，接电源后指示灯闪烁

1. 烧录好的TF卡插入树莓派背面卡槽、按下树莓派USB线的开关给电，电源指示灯红色常亮，绿色指示灯闪烁。
2. 环境配置

连入显示器 配置环境

1. 安装办公软件sudo apt-get install Abiword
2. 编辑、编译和调试简单的C语言程序，在桌面上建立“hello.c”文件，双击打开文件，写入“hello world”程序，并点击保存，打开LX终端，执行如下命令1)cd Desktop/进入桌面 2) gcc hello.c –o hello 编译程序 3)./ hello 执行程序



在LX终端 运行hello world程序

1. **设计思想**

首先通过Raspberry软件烧录TF卡，将数据程序导入内存卡中

接着在桌面建立“hello.c”文件并编辑“hello world”程序，保存

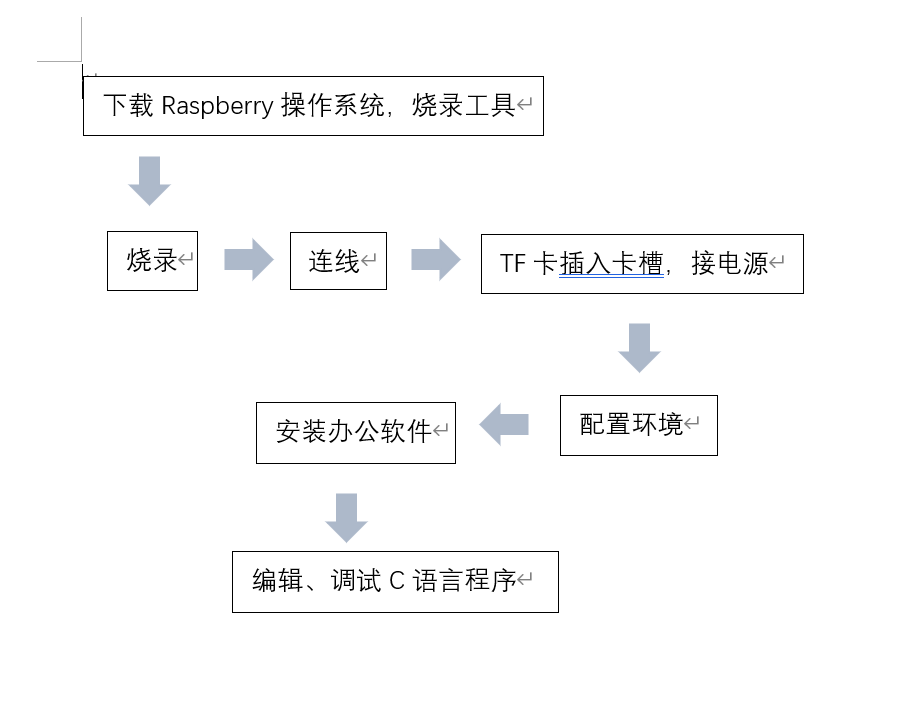
然后通过Linux终端命令运行程序

1. **硬件电路**

连线：Raspberry的HDMI<——>HDMI转VGA、 Raspberry的电源口<——>USB线电源、 HDMI转VGA 另外一个端口 <——>VGA线、关掉显示器然后VGA线的另一端<——>显示器Raspberry的USB端口可连接鼠标、键盘及U盘等。

烧录好的TF卡插入树莓派背面卡槽、按下树莓派USB线的开关给电，电源指示灯红色常亮，绿色指示灯闪烁。

1. **软件设计流程图**

****

设计流程图

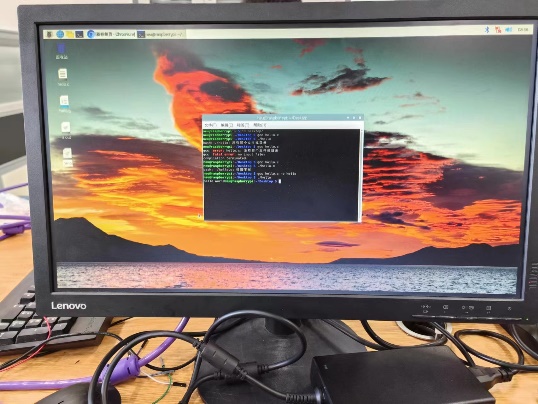
1. **源程序**

1)cd Desktop/进入桌面

2) gcc hello.c –o hello 编译程序

3)./ hello 执行程序

1. **实验结果**



运行出“hello world”结果

1. **思考题（选做题）**
2. 使用Python编写“hello world！”程序在树莓派上运行

第一步创建文件Vi hello.py，在这里.py表示使用Vi编辑器创建一个python格式的程序

第二步在Vi文本编辑器中写入代码#!/usr/bin/python Print "hello world"，代码编写好之后，使用Vi文本编辑器的保存操作进行保存。

第三步运行，在命令终端中输入Python hello.py，然后我们就可以看到hello world输出了，由于python无需编译，所以程序保存后，直接运行即可。

2.如何配置蓝牙功能，使手机与树莓派通过蓝牙传输文件

直接打开VNC viewer登陆进树莓派系统，右上角图标打开蓝牙。点击“Make Discoverable”，使蓝牙处于可被搜索的状态。  
直接打开手机的蓝牙，找到树莓派的蓝牙名称，点击配对，即可。

打开terminal，安装蓝牙相关包

sudo apt-get install pi-bluetooth bluez bluez-firmware blueman

添加用户到蓝牙组（假设用户名为pi）

sudo usermod -G bluetooth -a pi

sudo reboot

启动/增加SPP(Serial Port Profile)  
打开蓝牙配置文件(nano/vi都行)：

sudo nano /etc/systemd/system/dbus-org.bluez.service  
框框中的两行，第一行增加 -C，第二行直接添加

ExecStart=/usr/lib/bluetooth/bluetoothd -C

ExecStartPost=/usr/bin/sdptool add SP

重启树莓派

sudo reboot

蓝牙连接

在树莓派的terminal中输入命令（作用：等待蓝牙设备连接）

sudo rfcomm watch hci0

打开手机蓝牙调试器，搜索并连接树莓派，然后手机给树莓派发送文本数据

**实验二**

1. **实验目的**
2. 学习树莓派设计硬件系统的基本流程方法
3. 掌握树莓派GPIO接口的基本使用方法
4. 学习使用传感器的方法
5. 掌握基于树莓派的程序设计方法
6. 在Linux环境下使用命令编译和执行程序(C语言或Python)
7. **实验原理**

实验原件

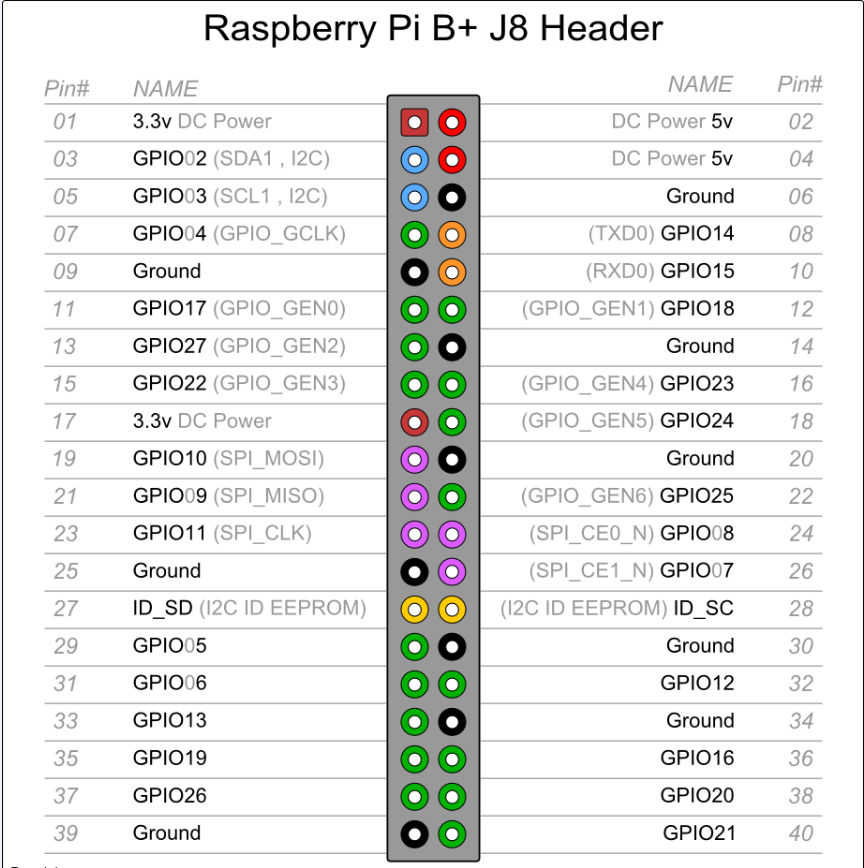
硬件：树莓派、led小灯、杜邦线(公母)、300R电阻(为了防止led短路，电流过大烧坏树莓派，可以在每个led正极处接一个300R电阻)、超声波测距传感器(HC-SR04)、面包板。

软件:

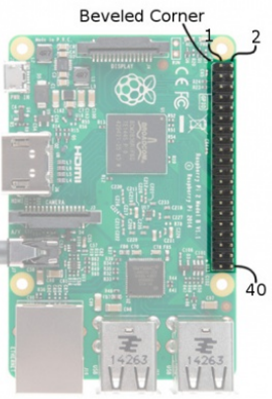
1. 使用C语言编程，如需要安装wringPi包，可通过sudo dpkg -I wiringpi-latest.deb命令安装；
2. 如果使用Python，则需要RPI.GPIO包，可以通过sudon apt-get install python-RPI.GPIO安装，但若使用较新的Debian，就不需要准备了，都自带了。

树莓派GPIO接口

1、GPIO(General Purpose I/O)含义为通用目的输入/输出接口，如01电压



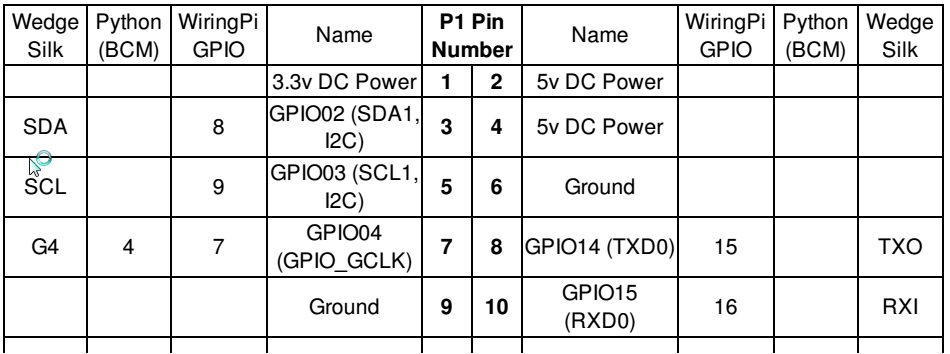
2、40个外接引脚通过软件定义(可与传感器等相连)



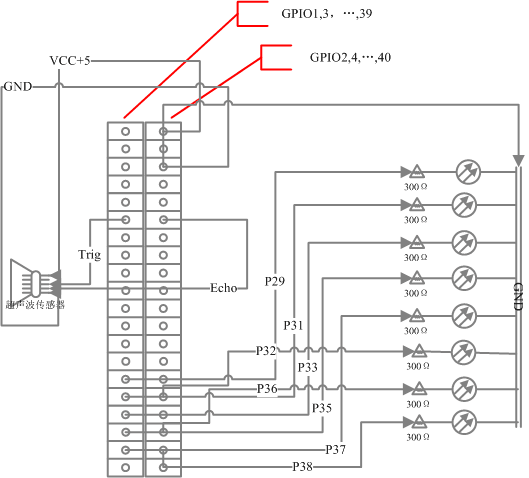
3、不够用时使用树莓派扩展板(智能农业等)

4、Python在GPIO有库

5、引脚说明



6.电路原理图



1. **实验内容**

实验需求

（1）设计硬件电路，并分别用C语言和Python语言分别实现下列需求的程序

1.小灯逐个闪烁（跑马灯程序），用户手离超声波传感器遮挡进入一定范围小灯全亮（20cm或更小距离）；

2.奇数小灯和偶数小灯交替闪烁，用户手离超声波传感器遮挡进入一定范围小灯全灭（20cm或更小距离）

1.连线树莓派与Led小灯

1号led正极接29号(29号引脚出来的杜邦线先接电阻再接led)

2号led正极接31号 (31号引脚出来的杜邦线先接电阻再接led)

3号led正极接33号(33号引脚出来的杜邦线先接电阻再接led)

4号led正极接35号(35号引脚出来的杜邦线先接电阻再接led)

5号led正极接37号(37号引脚出来的杜邦线先接电阻再接led)

6号led正极接32号(32号引脚出来的杜邦线先接电阻再接led)

7号led正极接36号(36号引脚出来的杜邦线先接电阻再接led)

8号led正极接38号(38号引脚出来的杜邦线先接电阻再接led)

所有led负极一起接到raspi的6号接地引脚(对应C语言程序里wiring Pi定义的IO端口21~28，见引脚说明中的第3列到第8列。)

2.连线树莓派与超声波传感器(HC-SR04)

Vcc对应raspi的2号针脚(5v)

Trig对应11号

Echo对应12号

Gnd对应6、9号(接地)。

3.执行程序

执行命令

cd Desktop/ 进入桌面

gcc index1.c –o index1 –lwiringPi

sudo ./index1

执行命令sudo python index1.py

1. **设计思路**

首先通过sudo dpkg -I wiringpi-latest.deb命令安装wringPi包

接着分别将Led小灯和超声波传感器与树莓派连接

最后在Linux终端执行命令，运行index1和index2程序

1. **硬件电路**

1.连线树莓派与Led小灯

1号led正极接29号(29号引脚出来的杜邦线先接电阻再接led)

2号led正极接31号 (31号引脚出来的杜邦线先接电阻再接led)

3号led正极接33号(33号引脚出来的杜邦线先接电阻再接led)

4号led正极接35号(35号引脚出来的杜邦线先接电阻再接led)

5号led正极接37号(37号引脚出来的杜邦线先接电阻再接led)

6号led正极接32号(32号引脚出来的杜邦线先接电阻再接led)

7号led正极接36号(36号引脚出来的杜邦线先接电阻再接led)

8号led正极接38号(38号引脚出来的杜邦线先接电阻再接led)

所有led负极一起接到raspi的6号接地引脚(对应C语言程序里wiring Pi定义的IO端口21~28，见引脚说明中的第3列到第8列。)

2.连线树莓派与超声波传感器(HC-SR04)

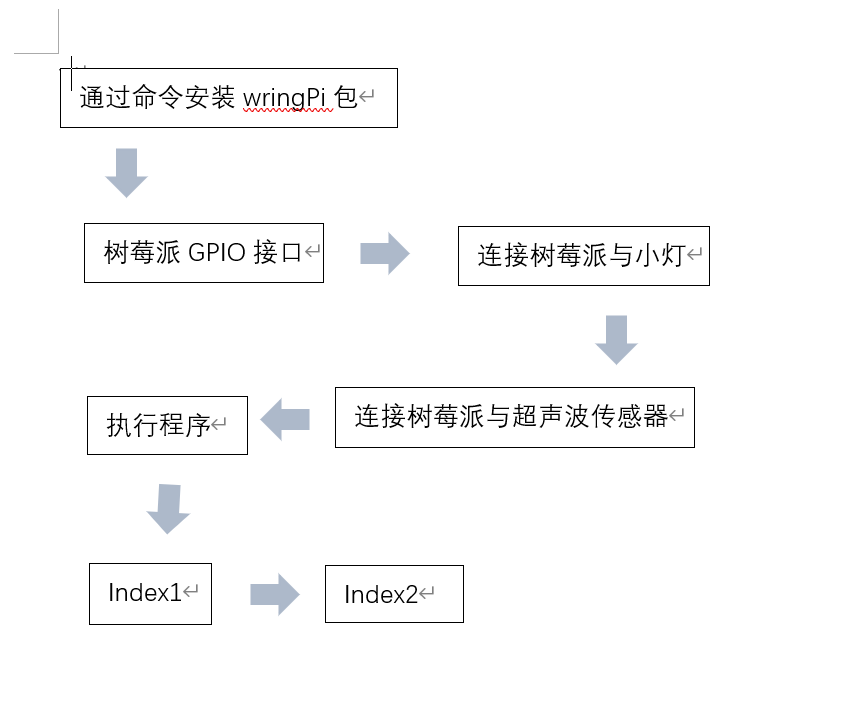
Vcc对应raspi的2号针脚(5v)

Trig对应11号

Echo对应12号

Gnd对应6、9号(接地)。

1. **软件设计流程图**

****

设计流程图

1. **源程序**

Index1

//方案一：8个小灯逐个闪烁（跑马灯程序）,用户手离超声波传感器遮挡进入一定范围小灯全亮

//import the necessary packages

#include <wiringPi.h>

#include <stdio.h>

#include <sys/time.h>

//define the HC\_SR04/led

#define Trig 0

#define Echo 1

#define led1 21

#define led2 22

#define led3 23

#define led4 24

#define led5 25

#define led6 26

#define led7 27

#define led8 28

//init //初始化函数

void ultraInit(void)

{

pinMode(Echo,INPUT); //超声波测距函数

pinMode(Trig, OUTPUT); //trig接口

pinMode(led1, OUTPUT);

pinMode(led2, OUTPUT);

pinMode(led3, OUTPUT);

pinMode(led4, OUTPUT);

pinMode(led5, OUTPUT);

pinMode(led6, OUTPUT);

pinMode(led7, OUTPUT);

pinMode(led8, OUTPUT);

}

//the distance detection, by the HC-SR04

float disMeasure(void) //定义距离函数

{

struct timeval tv1;

struct timeval tv2;

long start, stop;

float dis;

digitalWrite(Trig, LOW); //通入低电平

delayMicroseconds(2); //向树莓派发送2us的脉冲信号

digitalWrite(Trig, HIGH); //通入高电平

delayMicroseconds(10); //向树莓派发送10us的脉冲信号

digitalWrite(Trig, LOW);

while(!(digitalRead(Echo) == 1)); //是否等待波发出

//get the time of now

gettimeofday(&tv1, NULL); //获取当前时间

while(!(digitalRead(Echo) == 0)); //等待波是否接收

//get the time of now

gettimeofday(&tv2, NULL);

//the time(us)

start = tv1.tv\_sec \* 1000000 + tv1.tv\_usec; //将时间全部转化成微秒处理

stop = tv2.tv\_sec \* 1000000 + tv2.tv\_usec;

//the distance

dis = (float)(stop - start) / 1000000 \* 34000 / 2; //计算距离

return dis;

}

int main(int argc, char\* argv[])

{

printf("begin\n");

//the array of the led

int led[8] = {21, 22, 23, 24, 25, 26, 27, 28};

//define a temp value

int temp = 0;

//the distance

float dis;

if(wiringPiSetup() == -1) //初始化树莓派引脚

{ //when initialize wiring failed,print messageto screen

printf("setup wiringPi failed !");

return 1;

}

ultraInit(); //配置初始化

digitalWrite(21, LOW);

digitalWrite(22, LOW);

digitalWrite(23, LOW);

digitalWrite(24, LOW);

digitalWrite(25, LOW);

digitalWrite(26, LOW);

digitalWrite(27, LOW);

digitalWrite(28, LOW);

//loop

for(;;)

{

dis = disMeasure();

if(dis < 20)

{

digitalWrite(led1, HIGH);

digitalWrite(led2, HIGH);

digitalWrite(led3, HIGH);

digitalWrite(led4, HIGH);

digitalWrite(led5, HIGH);

digitalWrite(led6, HIGH);

digitalWrite(led7, HIGH);

digitalWrite(led8, HIGH);

}

else

{

digitalWrite(led1, LOW);

digitalWrite(led2, LOW);

digitalWrite(led3, LOW);

digitalWrite(led4, LOW);

digitalWrite(led5, LOW);

digitalWrite(led6, LOW);

digitalWrite(led7, LOW);

digitalWrite(led8, LOW);

digitalWrite(led[temp], HIGH);

if(temp == 7)

temp = 0;

else

temp += 1;

}

//time delay

delayMicroseconds(1000000);

//print the distance

printf("%0.2f cm\n", dis);

}

return 0;

}

Index2

//方案二：奇数小灯和偶数小灯逐个闪烁，用户手离超声波传感器遮挡进入一定范围小灯全灭（20厘米）

//import the necessary packages

#include <wiringPi.h>

#include <stdio.h>

#include <sys/time.h>

#include <stdbool.h>

//define the HC-SR04/led

#define Trig 0

#define Echo 1

#define led1 21

#define led2 22

#define led3 23

#define led4 24

#define led5 25

#define led6 26

#define led7 27

#define led8 28

//init //初始化函数

void ultraInit(void)

{

pinMode(Echo, INPUT); //超声波测距函数

pinMode(Trig, OUTPUT); //trig接口

pinMode(led1, OUTPUT);

pinMode(led2, OUTPUT);

pinMode(led3, OUTPUT);

pinMode(led4, OUTPUT);

pinMode(led5, OUTPUT);

pinMode(led6, OUTPUT);

pinMode(led7, OUTPUT);

pinMode(led8, OUTPUT);

}

//the distance detection, by the HC-SR04

float disMeasure(void) //定义距离函数

{

struct timeval tv1;

struct timeval tv2;

long start, stop;

float dis;

digitalWrite(Trig, LOW); //通入低电平

delayMicroseconds(2); //向树莓派发送2us的脉冲信号

digitalWrite(Trig, HIGH); //通入高电平

delayMicroseconds(10); //向树莓派发送10us的脉冲信号

digitalWrite(Trig, LOW);

while(!(digitalRead(Echo) == 1)); //是否等待波发出

//get the time of now

gettimeofday(&tv1, NULL); //获取当前时间

while(!(digitalRead(Echo) == 0)); //等待波是否接收

//get the time of now

gettimeofday(&tv2, NULL);

//the time(us)

start = tv1.tv\_sec \* 1000000 + tv1.tv\_usec; //将时间全部转化成微秒处理

stop = tv2.tv\_sec \* 1000000 + tv2.tv\_usec;

//the distance

dis = (float)(stop - start) / 1000000 \* 34000 / 2; //计算距离

return dis;

}

int main(int argc, char\* argv[])

{

printf("begin");

//the array of the led

int led[8] = {21, 22, 23, 24, 25, 27, 28, 29};

//define a temp value

bool temp = true;

//the distance

float dis;

if(wiringPiSetup() == -1) //初始化树莓派引脚

{ //when initialize wiring failed,print messageto screen //初始化接线失败时，将消息打印到屏幕

printf("setup wiringPi failed !");

return 1;

}

ultraInit(); //配置初始化

digitalWrite(21, LOW);

digitalWrite(22, LOW);

digitalWrite(23, LOW);

digitalWrite(24, LOW);

digitalWrite(25, LOW);

digitalWrite(27, LOW);

digitalWrite(28, LOW);

digitalWrite(29, LOW);

//loop

for(;;)

{

dis = disMeasure();

if(dis < 20){

digitalWrite(led1, LOW);

digitalWrite(led2, LOW);

digitalWrite(led3, LOW);

digitalWrite(led4, LOW);

digitalWrite(led5, LOW);

digitalWrite(led6, LOW);

digitalWrite(led7, LOW);

digitalWrite(led8, LOW);

}

else

{

digitalWrite(led1, LOW);

digitalWrite(led2, LOW);

digitalWrite(led3, LOW);

digitalWrite(led4, LOW);

digitalWrite(led5, LOW);

digitalWrite(led6, LOW);

digitalWrite(led7, LOW);

digitalWrite(led8, LOW);

if(temp == true)

{

digitalWrite(led1, HIGH);

digitalWrite(led3, HIGH);

digitalWrite(led5, HIGH);

digitalWrite(led7, HIGH);

}

else

{

digitalWrite(led2, HIGH);

digitalWrite(led4, HIGH);

digitalWrite(led6, HIGH);

digitalWrite(led8, HIGH);

}

temp = !temp;

}

//time delay

delayMicroseconds(1000000); //暂停程序运行

//print the distance

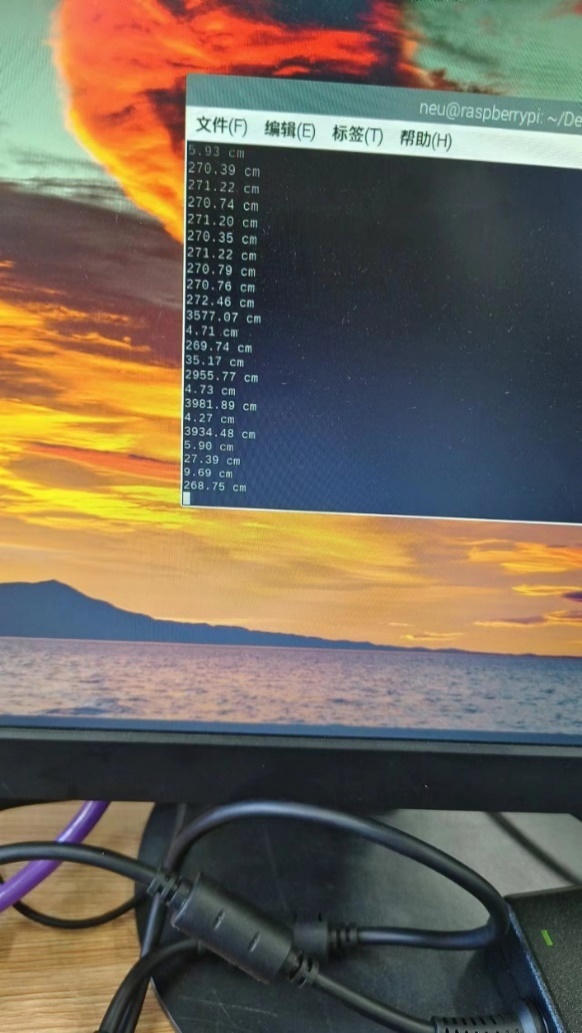
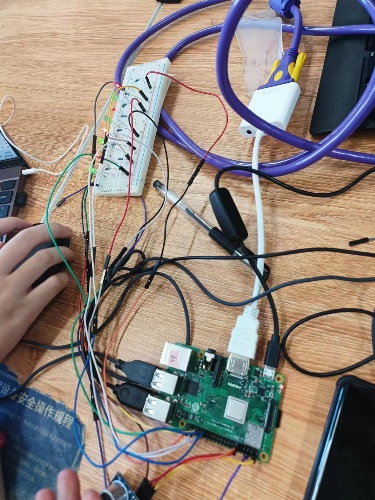
printf("%0.2f cm\n", dis);

}

return 0;

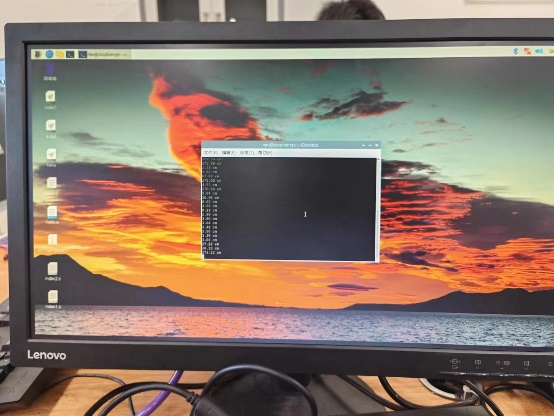
}

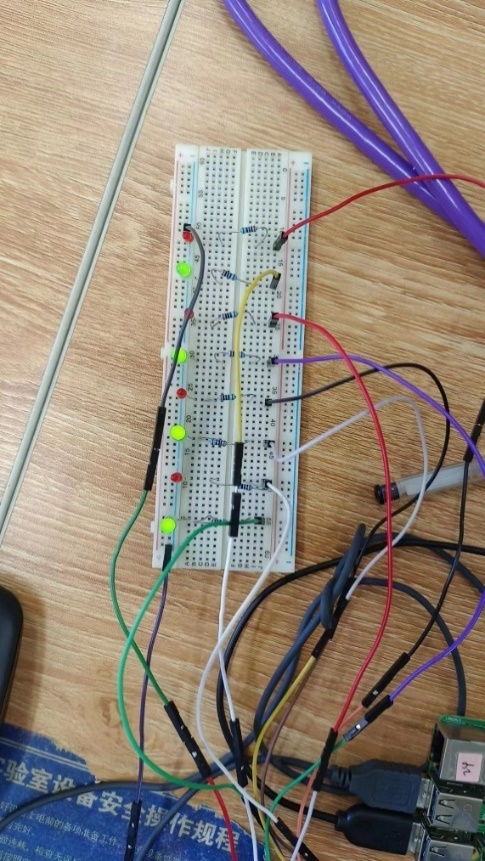
1. **实验结果**

**** 

运行结果1

8个小灯逐个闪烁,手离超声波传感器遮挡进入一定范围小灯全亮



****

运行结果2

奇数小灯和偶数小灯逐个闪烁，手离超声波传感器遮挡进入一定范围小灯全灭

1. **思考题**

1.根据自己的理解简要叙述C语言与Python语言的特点

01编程范式的区别

C语言是一门基于过程的编程语言，强调程序的执行过程，使用函数和指针等概念进行编程。

而Python则是一门基于面向对象编程的语言，强调对象和类的概念。Python同时也支持函数式编程，可以更加灵活地处理函数和数据之间的关系。

02语法结构的区别

C语言的语法结构更为严格，需要使用分号来结束每条语句。

Python则使用缩进来表示代码块，没有分号等符号的限制，代码更加简洁、易读、易维护。

03编程风格的区别

C语言通常采用的是指针和内存地址等底层概念，需要程序员手动进行内存分配和资源管理。

而Python则更强调高级别的语言特性，自动进行垃圾回收和资源管理，同时通过其丰富的标准库，为程序员提供更加方便的编程接口。

04应用场景的区别

C语言通常用于底层的操作系统和设备驱动程序开发，或高性能计算等领域。

而Python通常用于数据分析、人工智能、Web开发等方面，也逐渐在科学计算中广泛使用。