

# 智慧农业项目开发手册\_(IIC)V1

2019年7月11日 19:53

## 功能需求

1. 视频监控
2. 电磁阀控制
3. 温湿度监控
4. 4G联网

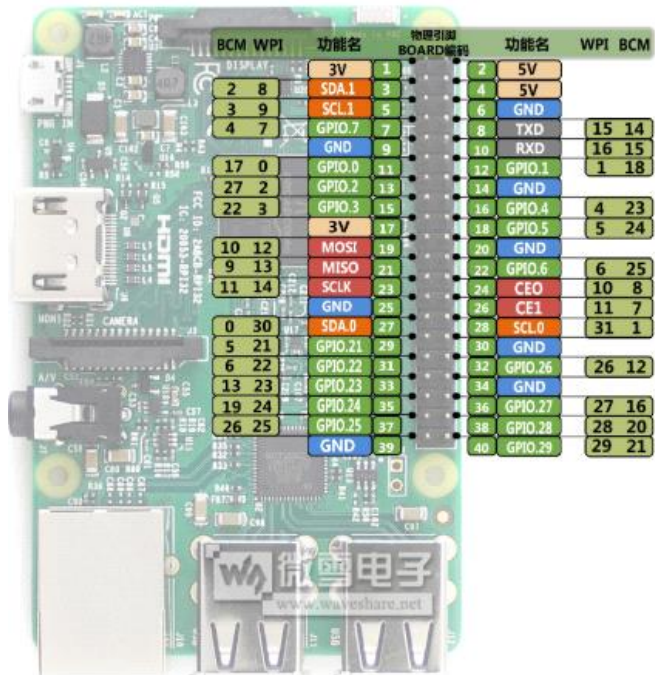
## 硬件设备

1. 树莓派
2. 温湿度传感器
  - a. DHT11传感器
  - b. DHT22传感器
  - c. 注意传感器包含了电阻，单个DHT不包含电阻
3. 树莓派摄像头
  - a. Pi Camera
  - b. USB Camera ( Linux适用的，免驱类型 )
4. 继电器
  - a. 单路继电器
  - b. 多路继电器 ( 相对单路只是统一使用一对供电接口 )
5. 4G模块
  - a. 4G网卡托+4G上网卡
  - b. 华为4G模块 ( ? 还未使用 ) +4G上网卡
6. USB式GPS
7. 面包板即杜邦线
  - a. 公对公
  - b. 公对母 ( 需要较多 )
  - c. 母对母 ( 几乎不用 )

## 电脑端软件

SDFormatter	格式化TF卡
Win32DiskImager	向TF卡中写入树莓派系统
WinSCP	用于和树莓派之间传输文件，需提前打开树莓派的SSH配置
Xshell	用于SSH连接树莓派，需提前打开树莓派的SSH配置

## 树莓派硬件



## 1. 树莓的三种引脚编码方式：BCM，BOARD，writingPi

- BCM：面包板上显示的是BCM模式的接口编号，BCM编号侧重CPU寄存器
- BOARD：BOARD为物理编号，从左上角开始为1，右下角为40，从左到右，从上到下：左边奇数，右边偶数：1-40
- writingPi：侧重实现逻辑，把扩展GPIO端口从0开始编号，这种编号方便编程。
- 在树莓派的terminal中输入gpio readall即可查看对应引脚示意图

BCM 编码方式	wpi 编码方式	功能名	物理接口						功能名	wpi 编码方式	BCM 编码方式
BCM	wPi	Name	Mode	V	-B Plus- Physical	V	Mode	Name	wPi	BCM	
2	8	3.3v			1	2		5v			
3	9	SDA.1	ALT0	1	3	4		5v			
4	7	SCL.1	ALT0	1	5	6		0v			
		GPIO. 7	IN	1	7	8	0	ALT0	TxD	15 14	
		0v			9	10	1	ALT0	RxD	16 15	
17	0	GPIO. 0	IN	0	11	12	0	IN	GPIO. 1	1 18	
27	2	GPIO. 2	IN	0	13	14		0v			
22	3	GPIO. 3	IN	0	15	16	0	IN	GPIO. 4	4 23	
		3.3v			17	18	1	OUT	GPIO. 5	5 24	
10	12	MOSI	ALT0	0	19	20		0v			
9	13	MISO	ALT0	1	21	22	1	OUT	GPIO. 6	6 25	
11	14	SCLK	ALT0	1	23	24	1	ALT0	CE0	10 8	
		0v			25	26	1	ALT0	CE1	11 7	
0	30	SDA.0	ALT0	1	27	28	1	ALT0	SCL.0	31 1	
5	21	GPIO.21	IN	1	29	30		0v			
6	22	GPIO.22	IN	1	31	32	0	IN	GPIO.26	26 12	
13	23	GPIO.23	IN	0	33	34		0v			
19	24	GPIO.24	IN	0	35	36	0	IN	GPIO.27	27 16	
26	25	GPIO.25	IN	0	37	38	0	IN	GPIO.28	28 20	
		0v			39	40	0	IN	GPIO.29	29 21	
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM	

## 2. 使用树莓派连接面包板时，注意连接正反

- 使用万能表测试面包板的两端线头的连通情况（测试最边角），确定排线连接方向
- 测试面板版上的孔的连通性
- 使用万能表电压测试，测接好的面包板上年的电压
- 使用万能表连通测试（信号图标），测是否连通

## 3. 面包板连接传感器时，要看好所需电压和接口线

- 传感器VCC端接5v或3v
- 传感器负极接地
- 传感器数据端接GPIO口

## 4. 继电器需要供电，另一个口接GPIO用于控制继电器

- 继电器com口接电源正极
- 继电器NC（normal close）口用于断电时保持连接状态的情况
- 继电器NO（normal open）用于断电时保持断开状态的情况

## 树莓派软件

### Linux的一些基本使用命令

netstat -apn	查看所有的进程和端口使用情况
--------------	----------------

## 一、安装前的配置

1. 第一次开启树莓派先连入显示器，待配置完成后再使用7寸屏和SSH
2. 使用指定烧录软件安装好树莓派后，注意地区设置为中国，否则无法使用wifi
3. 更新树莓派，安装必要软件

sudo apt-get update	升级
sudo apt-get upgrade	
sudo apt-get install vim	安装vim
sudo apt-get install git	安装git
sudo apt-get install cmake	
sudo apt-get install python-dev	
sudo apt-get install build-essential	

4. 启动SSH服务

sudo raspi-config	选择Interface Options中开始SSH
-------------------	---------------------------

或者直接开始菜单中，配置树莓派，开启SSH

5. 检查USB设备是否被识别

ls /dev	若有USB摄像头，则结果中有video0
lsusb	所有usb设备名称

## 二、使用摄像头

### 基本功能

sudo raspi-config	选择Interface Options中开启Camera
sudo reboot	重启生效

### 测试

raspistill -t 1000 -o image.jpg	1秒钟（时间单位为毫秒）延迟后拍摄一张照片，并命名为image.jpg
raspistill -t 1000 -o image.jpg -w 640 -h 480 -q 5	拍摄一张自定义大小和帧率的照片
raspivid -o myvideo.h264	拍摄一段视频：默认是视频长度为5s，分辨率为1920*1080，帧率：17
raspivid -o myvideo.h264 -t 10000 -w 640 -h 480	拍摄一段视频：分辨率为640*480 时间为10s

### 使用网页查看摄像头视频

工具1：mjpg-streamer

安装必要库：

```
sudo apt-get install subversion
sudo apt-get install libjpeg8-dev
sudo apt-get install imagemagick
sudo apt-get install libv4l-dev
```

安装工具：

```
sudo git clone https://github.com/jacksonliam/mjpg-streamer.git
cd mjpg-streamer/mjpg-streamer-experimental
make all
sudo make install
```

使用：

./mjpg_streamer -i "./input_uvc.so" -o "./output_http.so -w ./www"	1.使用USB摄像头
./mjpg_streamer -o "output_http.so -w ./www" -i "input_raspicam.so"	2.使用Pi Camera
浏览器中：http://raspberrypi-address:8080/?action=stream	只能在同一局域网中使用

参考：<http://shumeipai.nxez.com/2017/05/14/raspberry-pi-mjpg-streamer-installation.html>

也可以在公网中使用[https://blog.csdn.net/Meteor\\_s/article/details/81124086](https://blog.csdn.net/Meteor_s/article/details/81124086)

工具2：pistreaming

sudo apt-get install ffmpeg python3-picamera python3-ws4py	安装依赖包
git clone <a href="https://github.com/waveform80/pistreaming.git">https://github.com/waveform80/pistreaming.git</a>	
cd pistreaming	
python3 server.py	开启服务
浏览器中：http://raspberrypi-address:8082/?action=stream	智能使用Pi Camera，除非修改server.py中的内容 可以在广域网中使用（树莓派使用的是广域网）

使用此工具，若树莓派使用的是局域网的ip，需要在此局域网的路由器设置端口转发，保证任意网段浏览器使用路由器的ip+指定转发端口能够访问树莓派

关于http和https：<https://blog.csdn.net/hixzb/article/details/78107811>

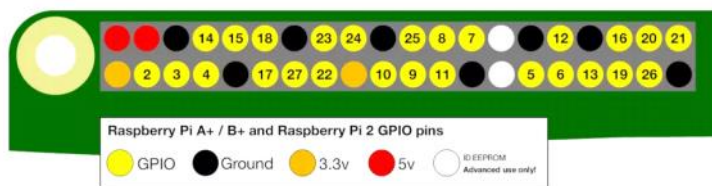
### 三、安装7寸屏幕驱动

sudo rm -rf LCD-show	移除原有文件夹
git clone <a href="https://github.com/goodtft/LCD-show">https://github.com/goodtft/LCD-show</a>	
chmod -R 755 LCD-show	
cd LCD-show	
sudo ./LCD7B-show	1.使用7寸720p驱动
sudo ./LCD-hdmi	2.使用hdmi驱动

### 四、控制GPIO

GPIO (General Purpose Input/Output) 的意思就是 通用型输入输出

参考：<https://blog.csdn.net/chentuo2000/article/details/81051645>



电源和地不用多说，3.3v 就是 3.3v输出，5v 就是 5v输出，地就是地。我们没办法通过软件编程的方法对它们进行设置，这也是为什么它们没有编号。能软件编程的就是亮黄色的 GPIO 引脚了。

对于GPIO引脚的两种使用方式

- 1.设置为输出，输出高电平(3.3v)或输出低电平(0v)
- 2.设置为输入，这时可以接收信号，这一信号不光可以来自机械开关，还可以是各式各样的传感器或另一台电脑或其他设备。

安装wiringPi，GPIO控制库函数（基于c语言）

sudo apt-get install wiringpi	编译和安装wiringPi
-------------------------------	---------------

wiringPi：<https://wenku.baidu.com/view/74b7c14b14791711cd791707.html>

gpio -v	
---------	--

gpio readall	查看所有引脚电平
gpio mode 1 out	设置[以writePi编号为1]的GPIO（即GPIO1口）口为输出模式
gpio mode -g 18 out	设置[以BCM编号为18]的GPIO（即GPIO1口）口为输出模式
gpio read 1	获取当前GPIO1口的电平（0或1）
gpio write 1 0	设置当前GPIO1口的电平为0（低电平）

★ 调试各个GPIO口的时候可以调出gpio readall查看对应口的输入输出模式

首先我们把LED和树莓派连接。LED的正极串联一个1KΩ电阻接树莓派的GPIO18(pin12)，负极接地

#### 使用shell命令控制GPIO

sudo echo 18 >/sys/class/gpio/export	> 是IO重定向符号，IO重定向是指改变linux标准输入和输出的默认设备，指向一个用户定义的设备。echo 18 > export就是把18写入到export文件中。
cd /sys/class/gpio/gpio18  ls	查看GPIO18引脚（在Linux中设备都以文件的形式，引脚也是设备）
sudo echo out > direction	设置GPIO18为输出模式
sudo echo 1 > value	向value文件中输入1，GPIO输出高电平，LED点亮
sudo echo 0 > value	向value文件中输入0，GPIO输出低电平，LED熄灭

#### 使用Shell脚本控制GPIO

sudo nano led_on_off.sh	新建名为led_on_off.sh的脚本
echo \$1 > /sys/class/gpio/export  echo out > /sys/class/gpio/gpio\$1/direction  echo 1 > /sys/class/gpio/gpio\$1/value  sleep 5 #延时5秒  echo 0 > /sys/class/gpio/gpio\$1/value  echo \$1 > /sys/class/gpio/unexport	脚本中写入 说明：shell脚本可传入参数，例如\$1代表第1个参数，\$2代表第2个参数
sudo chmod +x ledonoff.sh	执行权限
sudo ./ledonoff.sh 18	运行，控制引脚18 LED点亮，持续5秒钟关闭

#### 使用python控制GPIO

树莓派系统本身集成的RPi.GPIO类库

python	进入python
import RPi.GPIO as GPIO	导入python类库RPi.GPIO，命名为别名为GPIO
GPIO.setmode(GPIO.BCM)	编码方式，基于BCM
GPIO.setup(18,GPIO.OUT)	18为输出模式
GPIO.output(18,GPIO.LOW)	GPIO18输出低电平，LED熄灭
GPIO.output(18,GPIO.HIGH)	GPIO18输出高电平，LED点亮
GPIO.cleanup()	进行清理
Ctrl+D	推出python

也可以编写python脚本程序控制GPIO

## 六、传感器数据收集与阿里云IOT

读取DHT11温湿度传感器数据

参考：

<http://shumeipai.nxez.com/2014/10/10/raspberry-dht11-get-temperature-data.html>

<http://shumeipai.nxez.com/2018/05/16/dht11-temperature-and-humidity-sensor-raspberry-pi.html>

DHT11有三个Pin脚，需要在电源和数据脚之间串联一个上拉电阻(4.7K-10K)，通常情况下，购买DHT11模块的话都自带了这个电阻  
湿度检测范围：20-80% (5% 精度)

温度检测范围：0-50°C (±2°C 精度)

DHT Pin	Signal	Pi Pin
1	3.3V	1
2	Data/Out	11 (GPIO17)
4	Ground	6 or 9

两种方式读取DHT11数据

- 通过高低电平收集40bit数据，特定的信号协议，比较烦不使用
- 使用Adafruit DHT 库

### ★ 安装Adafruit DHT库

sudo git clone <a href="https://github.com/adafruit/Adafruit_Python_DHT.git">https://github.com/adafruit/Adafruit_Python_DHT.git</a>	
cd Adafruit_Python_DHT	
sudo python setup.py install sudo python3 setup.py install	给 Python 2 和 Python 3 安装该库

Adafruit例子

cd Adafruit_Python_DHT cd examples python AdafruitDHT.py 11 17	Adafruit 提供了示例程序 这两个参数分别表示 DHT11 和数据引脚所接的树莓派 GPIO 编号。成功的话会输出
--	---

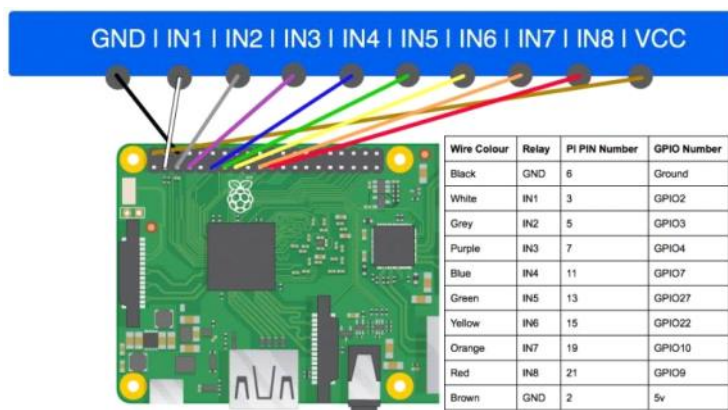
```
1 import Adafruit_DHT
2
3 # Set sensor type : Options are DHT11,DHT22 or AM2302
4 sensor=Adafruit_DHT.DHT11
5
6 # Set GPIO sensor is connected to
7 gpio=17
8
9 # Use read_retry method. This will retry up to 15 times to
10 # get a sensor reading (waiting 2 seconds between each retry).
11 humidity, temperature = Adafruit_DHT.read_retry(sensor, gpio)
12
13 # Reading the DHT11 is very sensitive to timings and occasionally
14 # the Pi might fail to get a valid reading. So check if readings are valid.
15 if humidity is not None and temperature is not None:
16     print('Temp={0:0.1f}*C Humidity={1:0.1f}%'.format(temperature, humidity))
17 else:
18     print('Failed to get reading. Try again!')
```

控制继电器

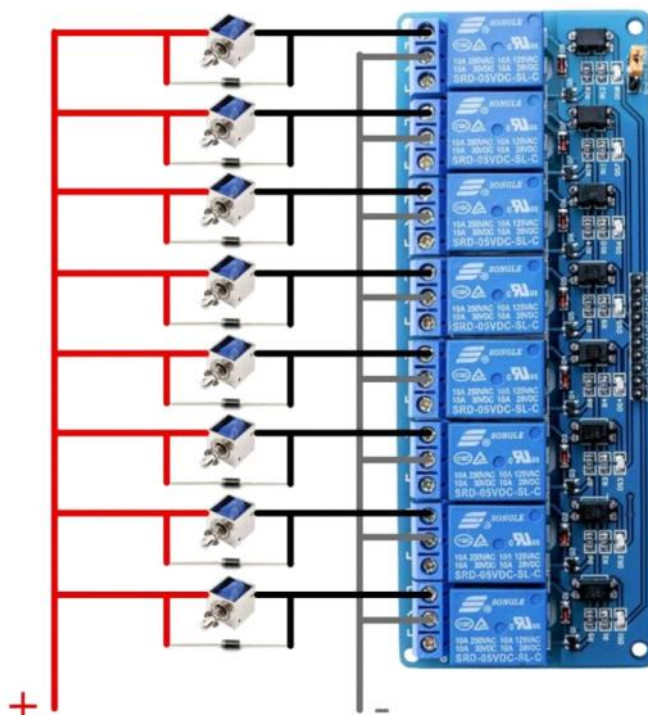
! 注意继电器就接3.3v

八路继电器：<http://shumeipai.nxez.com/2019/06/10/a-raspberry-pi-powered-junk-drum-machine.html>





## 12v Circuit Diagram



继电器有八个通道，可以独立的开、关八个电磁阀。每个通道包含四个连接器，其中三个在高压电路中使用，稍后会看到。剩下一个连接到低压电路的 IN 引脚上。当树莓派的 GPIO 引脚发送 5V 到给定通道 IN 引脚上时，继电器将连接到相应的 12v 电路。

### 安装阿里云IoT客户端SDK

参考：

<https://www.cnblogs.com/yefanqiu/p/10063554.html>

<https://www.yuque.com/cloud-dev/iot-tech/rz6fpl>

首先设备接入IOT，需要申请设备码

阿里云物联网平台：<https://iot.console.aliyun.com/product>

产品	功能定义	增加需要的属性
	在线调试	

设备	运行状态	查看数据
	在线调试	

#### 设备端开发

pip install aliyun-python-sdk-iot-client	阿里云Python SDK
pip install paho-mqtt	