1. 红外遥控使用

### 红外按键的编码

#ifndef UNIT\_TEST

#include <Arduino.h>

#endif

#include <IRremoteESP8266.h>

#include <IRrecv.h>

#include <IRutils.h>

//An IR detector/demodulator is connected to GPIO pin 14(D5 on a NodeMCU

//board).

uint16\_t RECV\_PIN = 14;

IRrecv irrecv(RECV\_PIN);

decode\_results results;

void setup() {

Serial.begin(115200);

irrecv.enableIRIn(); //Start the receiver

}

void loop() {

if (irrecv.decode(&results)) {

//print() & println() can't handle printing long longs. (uint64\_t)

serialPrintUint64(results.value, HEX);

Serial.println("");

irrecv.resume(); //Receive the next value

}

delay(100);

}

测试到的键值和自己硬件带的遥控器有关系，可以根据这些键值做操作

第1行

----------------------

FFA25D静音

FF629D停止

FFE21D播放/暂停

第2行

----------------------

FF22DD设置

FF02FD向上

FFC23D退出

第3行

----------------------

FFE01F向左

FFA857确认

FF906F向右

第4行

----------------------

FF9867向下

### 红外遥控控制灯亮与灭

接线：led灯正极接到D6

#ifndef UNIT\_TEST

#include <Arduino.h>

#endif

#include <IRremoteESP8266.h>

#include <IRrecv.h>

#include <IRutils.h>

//An IR detector/demodulator is connected to GPIO pin 14(D5 on a NodeMCU

//board).

uint16\_t RECV\_PIN = 14;

IRrecv irrecv(RECV\_PIN);

decode\_results results;

//LED Pins

int LEDpin = D6;

void setup() {

Serial.begin(115200);

irrecv.enableIRIn(); //Start the receiver

pinMode(LEDpin, OUTPUT);

digitalWrite(LEDpin, LOW);

}

void loop() {

if (irrecv.decode(&results)) {

//print() & println() can't handle printing long longs. (uint64\_t)

serialPrintUint64(results.value, HEX);

Serial.println("");

if (results.value == 0xFF22DD) //若接收到按键设置按下的指令，打开LED

{

digitalWrite(LEDpin, HIGH);

}

else if (results.value == 0xFFC23D) //接收到退出按键按下，关闭LED

{

digitalWrite(LEDpin, LOW);

}

irrecv.resume(); //Receive the next value

}

delay(100);

}

1. 超声波测距

#define Trig D2 //引脚Tring 连接 IO D2

#define Echo D3 //引脚Echo 连接 IO D3

float cm; //距离变量

float temp; //

void setup() {

Serial.begin(115200);

pinMode(Trig, OUTPUT);

pinMode(Echo, INPUT);

}

void loop() {

//给Trig发送一个低高低的短时间脉冲,触发测距

digitalWrite(Trig, LOW); //给Trig发送一个低电平

delayMicroseconds(2); //等待 2微妙

digitalWrite(Trig, HIGH); //给Trig发送一个高电平

delayMicroseconds(10); //等待 10微妙

digitalWrite(Trig, LOW); //给Trig发送一个低电平

temp = float(pulseIn(Echo, HIGH)); //存储回波等待时间,

//pulseIn函数会等待引脚变为HIGH,开始计算时间,再等待变为LOW并停止计时

//返回脉冲的长度

//声速是:340m/1s 换算成 34000cm / 1000000μs => 34 / 1000

//因为发送到接收,实际是相同距离走了2回,所以要除以2

//距离(厘米) = (回波时间 \* (34 / 1000)) / 2

//简化后的计算公式为 (回波时间 \* 17)/ 1000

cm = (temp \* 17) / 1000; //把回波时间换算成cm

Serial.print("Echo =");

Serial.print(temp);//串口输出等待时间的原始数据

Serial.print(" | | Distance = ");

Serial.print(cm);//串口输出距离换算成cm的结果

Serial.println("cm");

delay(500);

}

1. DHT11温湿度传感器

//温湿度的数据针脚

#define DHT11PIN D5

//温湿度库对应的头文件

#include <dht11.h>

//全局变量

dht11 DHT11;

//数据

String strData;

//初始化函数

void setup()

{

//设置串口的波特率

Serial.begin(115200);

//设置针脚端口的模式

pinMode(DHT11PIN, OUTPUT);

}

//循环执行

void loop()

{

//读取温湿度的数据

int chk = DHT11.read(DHT11PIN);

//温度

float temp = (float)DHT11.temperature;

Serial.print("Tep: ");

Serial.print(temp);

Serial.println("C");

//湿度

float humi = (float)DHT11.humidity;

Serial.print("Hum: ");

Serial.print((float)DHT11.humidity);

Serial.println("%");

delay(500);

}

五、步进电机

#include <Stepper.h>

//change this to the number of steps on your motor

#define STEPS 512

//定义引脚的连接

Stepper stepper(STEPS, D5, D6, D7, D8);

//上一次的读取模拟量

int previous = 0;

//初始化电机转动速度

void setup()

{

stepper.setSpeed(20);

}

//主循环

void loop()

{

//读取AD值，在LY-51S可以直接通过电位器调节

int val = analogRead(A0);

//把速度设置为这次和上次读取的差值

stepper.step(val - previous);

//记下这次读取值，方便下次比较

previous = val;

}