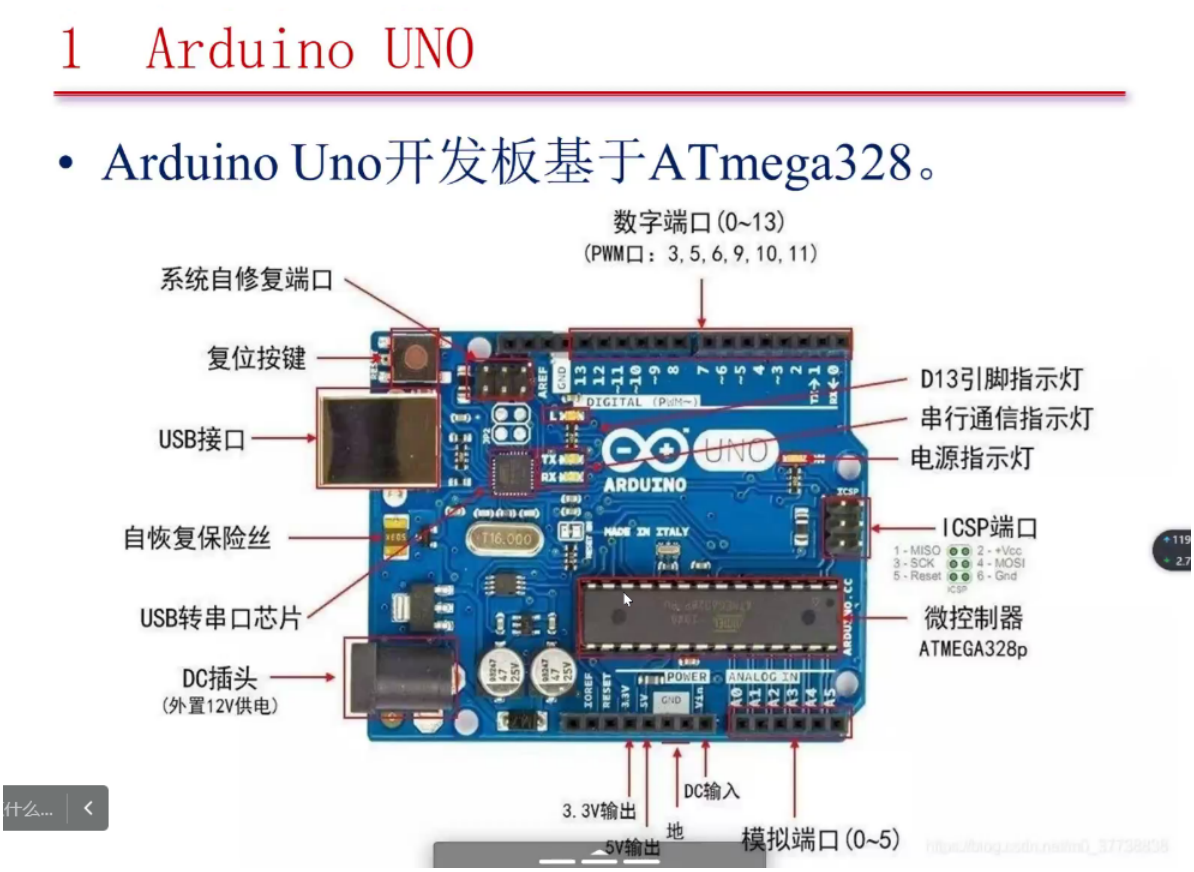
电子积木

软件: Arduino IDE, 系统: Arduino UNO, 语言: Arduino Language.

参考资料: Arduino 开发, 从零开始学.

参考网站: www.arduino.cc



数字引脚：0-13，

模拟引脚：14-19.

输出: 3, 9, 10, 11

变量类型

类型	字节	信息
byte	1	0-255
word	2	0-65536
boolean	1	false, true
string	-	

类型	字节	信息
String	-	
array		
void	0	一个标识符

我的电脑:

右边是 COM6.

proteus 里的元件名

- 电阻 resistor
- 虚拟串口监视器 COMPIM

  Arduino 板子 0 号与 1 号与之 2 号和 3 号连接.
- 开关: switch
- 按钮: button
- WS2812
  - 引脚: VDD 为电源, DOUT 为数据输出, VSS 为地, DIN 为数据输入.
  - FastLED.h
- 光敏电阻 torch_ldr
- 温度传感器 LM35
- 蜂鸣器: sounder, speaker
- 数码管: 7seg-com-anode
- 热敏电阻: NTC
- 液晶显示器: LCD, AGM1232G.
- 共阴数码管驱动芯片 MAX 7219.
- 液晶显示器: SSD1306 (显示的更多)
- 直流电机驱动芯片 L293D.

HC-SR501

LED 灯 "长正短负"

共阳型 LED 数码管低亮高灭, 八段需要九个引脚.

D1

数码管

七段数码管: abcdefgh

八段数码管: 多了一个显示小数点的发光二极管单元 DP (Decimal Point)

共阳极: 只引出两个阳极和各个阴极引脚.

共阴极 GND

显示字符	共阳极	共阴极	显示字符	共阳极	共阴极
0	C0H	3FH	8	80H	7FH
1	F9H	06H	9	90H	6FH
2	A4H	5BH	A	88H	77H
3	B0H	4FH	B	83H	7CH
4	99H	66H	C	C6H	39H
5	92H	6DH	D	A1H	5EH
6	82H	7DH	E	86H	79H
7	F8H	07H	F	8EH	71H

多位数码管: MAX7219 (Proteus 元件) 是共阴数码管驱动芯片, 可以同时驱动 8 个共阴数码管或 LED 点阵, 控制接口是串行接口.

- MAX7219 示例

```
#include <max7219.h>
MAX7219 max7219;
void setup() {
    Serial.begin(9600);
    max7219.Begin();    // 注意首字母大写
}
void loop() {
    String temp;        // 首字母大写
    char temp2[8];
    int y;

    // 显示数字或字母
    max7219.DisplayText("95.67F", 1);   // 1 为右对齐
    delay(3000);
    max7219.Clear();    // 清屏, 注意大写
    max7219.DisplayText("95.67F", 0);   // 0 为左对齐
    delay(3000);
    max7219.Clear();

    // 十进制数数 (计时)
    for (float x = 0; x < 1; x += 0.1) {
        temp = String(x);   // 转为 String 类型
        temp.toCharArray(temp2, temp.length()); // 转为字符数组
        // 若包含 <stdio.h>, 上两行可以直接用 sprintf()
        max7219.DisplayText(temp2, 1);
        Serial.println(x);
        delay(500);
    } // 不 delay 的话速度快, 但也不是快的离谱
```

```
    max7219.Clear();

    // 逐位显示字母
    max7219.DisplayChar(7, 'H', 0);     // 左数第 7 位，从 0 开始
    max7219.DisplayChar(6, 'E', 0);
    max7219.DisplayChar(5, 'L', 0);
    max7219.DisplayChar(4, 'L', 0);
    max7219.DisplayChar(3, 'O', 0);
    delay(1000);
    max7219.Clear();
    max7219.DisplayText("World", 0);
}
```

- MAX7219 用法总结

```
#include <max7219.h>
#define LEFT 0      // 这样更清晰
#define RIGHT 1
Max7219 max7219;    // 定义实例，其成员函数首字母大写
max7219.Begin();    // 在 setup() 中
max7219.Clear();    // 清屏
max7219.DisplayText(chars, justify);    // justify 为 0 (左) 或 1 (右)
max7219.DisplayChar(pos, ch, justify);  // pos 显示的位置，从 0 计数

// 其它可能会用到的函数
float x = 32;
str = String(x);    // 转为 String 类型
str.toCharArray(s, str.length());   // 转为字符数组并存储至 s 中
```

液晶显示器

LCD (Liquid Crystal Display), 在 Proteus 中的元件为 LCD1602.

编号	符号	引脚说明	编号	符号	引脚说明
1	VSS	电源地	9	D2	Data I/O
2	VDD	电源正极	10	D3	Data I/O
3	VL	液晶显示偏压信号	11	D4	Data I/O
4	RS	数据/命令选择端（H/L）	12	D5	Data I/O
5	R/W	读/写选择端（H/L）	13	D6	Data I/O
6	E	使能信号	14	D7	Data I/O
7	D0	Data I/O	15	BLA	背光源正极
8	D1	Data I/O	16	BLK	背光源负极

LCD1602	—>	Arduino UNO	说明
GND	—>	GND	接地
VCC	—>	5V	5V电源
V0	—>		连接3脚继电器中间，用于调节对比度

LCD1602	—>	Arduino UNO	说明
RS	—>	12	随便接一个输出口，方便接线、画图
R/W	—>	GND	接地，写模式
EN	—>	11	随便接一个输出口，方便接线、画图
D0~D3	—>		4位工作模式，不使用
D4~D7	—>	5、4、3、2	其它口也行，方便接线、画图
BLA	—>		背光，电源正极，可选
BLK	—>		背光，接地，可选

- LCD1602 实例

```
#include <LiquidCrystal.h>
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2; // 引脚
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
void setup() {
    lcd.begin(16, 2);   // LCD 的列数与行数
    lcd.println("Hello, world!");
}
void loop() {
    lcd.noDisplay();    // 关闭
    delay(500);
    lcd.display();      // 开启
    delay(500);
}
```

Arduino 实例

- 键盘
- 按键
- 计算器
- 时钟
- 数字温度计
- 直流电机 / 马达控制
- 步进电机
- 伺服电机

时钟

- AdvDS1307Clockbasic

```
#include <Wire.h>
#include "DS1307.h"
DS1307 clock;
void setup() {
    Serial.begin(9600);
    clock.begin();  // 开始计时
    clock.fillByYMD(2022, 10, 22);
```

```
    clock.fillByHMS(19, 23, 30);
    clock.fillDayOfWeek(THU);
    clock.setTime();    // 写入时间至 RTC 芯片
}
void loop() {
    printTime();    // 自己写的函数
    delay(1000);
}
void printTime() {
    clock.getTime();
    Serial.print(clock.hour, DEC);
    Serial.print(":");
    Serial.print(clock.minute, DEC);
    Serial.print(":");
    Serial.print(clock.second, DEC);
    Serial.print(" ");
    Serial.print(clock.month, DEC);
    Serial.print("/");
    Serial.print(clock.dayOfMonth, DEC);
    Serial.print("/");
    Serial.print(clock.year + 2000, DEC);
    Serial.print(" ");
    switch (clock.dayOfWeek) {
        case MON: Serial.print("MON"); break;
        case TUE: Serial.print("TUE"); break;
        case WED: Serial.print("WED"); break;
        case THU: Serial.print("FRI"); break;
        case SAT: Serial.print("SAT"); break;
        case SUN: Serial.print("SUN"); break;
    }
    Serial.println(" ");
}
```

- AdvDS1307ClockLCD

```
#include <Wire.h>
#include "DS1307.h"
#include "HD44780_LCD_PCF8574.h"
DS1307 clock;
HD44780LCD lcdTime(2, 16, 0x27);
void setup() {
    Serial.begin(9600);
    clock.begin();
    clock.fillByYMD(2022,10, 19);
    clock.fillByHMS(19, 12, 35);
    clock.fillDayOfWeek(THU);
    clock.setTime();    // 写入时间
    lcdTime.PCF8574_LCDInit(LCDCursorTypeOff);  // 去除光标
    lcdTime.PCF8574_LCDClearScreen();       // 清屏
    lcdTime.PCF8574_LCDBackLightSet();  // LCD 初始设置
}
void loop() {
    printTime();
    delay(500);
}
```

```
void printTime() {
    char hour[3], minute[3], second[3];
    char month[3], dayOfMonth[3], year[5];

    clock.getTime();
    sprintf(hour, "%02d", clock.hour);
    sprintf(minute, "%02d", clock.minute);
    sprintf(second, "02d", clock.second);
    sprintf(month, "02d", clock.month);
    sprintf(dayOfMonth, "%02d", clock.dayOfMonth);
    sprintf(year, "%d", clock.year + 2000);

    lcdTime.PCF8574_LCDGOTO(2, 4);
    lcdTime.PCF8574_LCDSendString(hour);
    lcdTime.PCF8574_LCDSendChar(':');
    lcdTime.PCF8574_LCDSendString(minute);
    lcdTime.PCF8574_LCDSendChar(':');
    lcdTime.PCF8574_LCDSendString(year);
    lcdTime.PCF8574_LCDSendChar('-');
    lcdTime.PCF8574_LCDSendString(month);
    lcdTime.PCF8574_LCDSendChar('-');
    lcdTime.PCF8574_LCDSendString(dayOfMonth);

    Serial.print(clock.hour, DEC);
    ..... // 同上
}
```

- 用法总结

```
#include <Wire.h>
#include "DS1307.h"
#include "HD44780_LCD_PCF8574.h"
DS1307 clock;
HD44780LCD lcdTime(2, 16, 0x27);

// 初始化 clock 实例，用于获取时间
clock.begin();
clock.fillByYMD(2022,10,19);
clock.fillByHMS(19,12,35);
clock.fillDayOfWeek(THU);
clock.setTime();

// 初始化 lcdTime 实例，用于显示时间
lcdTime.PCF8574_LCDInit(LCDCursorTypeOff);
lcdTime.PCF8574_LCDClearScreen();
lcdTime.PCF8574_LCDBackLightSet(true);

// 获取时间
clock.getTime();

// 使用 lcdTime
lcdTime.PCF8574_LCDGOTO(2, 4);          // 移动光标至第 2 行第 4 列
lcdTime.PCF8574_LCDSendString(s);       // 输出字符数组
lcdTime.PCF8574_LCDSendChar(ch);        // 输出字符
```

参考示例

[LCD 液晶显示器]

[MAX7912 自带示例及其电路]

[时钟基础]

[时钟]

我已经完全不会 cpp 了, 刚才测试了点东西, 放这儿了

```
#include <iostream>
#include <cmath>
using namespace std;

class Test
{
public:
  Test(int a = 0): a(a) {}
  virtual void show() {cout << a << endl;}
protected:
  int a;
};

class Test2: public Test
{
public:
  Test2(int a = 0, char b = 0): Test(a), b(b) {}
  void show() {cout << a << ' ' << b << endl;}
protected:
  int b;
};

typedef struct testS {
    int a = 1;
} testS;

int main()
{
    Test t(1);
    t.show();
    Test2 t2(2, 3);
    t2.show();                      // protected 的用法
    struct testS ts1, ts2 = {2};    // 要用大括号
    ts1 = testS{2};                 // 这样也可以哦
    cout << ts1.a;
    return 0;
}
```

时钟设计: RTC

使用器件

1. 使用串口监视器 (可以没有).
2. 使用 DS1307 获取时间 (实例名 clock).
3. 使用 HD44780LCD 显示时间 (实例名 lcdTime).
4. 使用 LiquidCrystal 显示其它信息 (实例名为 lcdInfo), 如星期几, 当前所在模式, 闹钟是否打开.

注 尝试过了, 不能使用 max7219 显示星期, 一方面完整单次最多有 9 个字母, 另一方面很多字母都无法显示, 或者较难辨认. 星期的单词中有些就无法显示.

引脚数

1. 串口监视器: 2 个数字引脚.
2. clock: 2 个数字引脚, 2 个模拟引脚.
3. lcdTime: 6 个数字引脚, 其中有 2 个与 DS1307 共用
4. 按钮: 4 个数字引脚 (不能与其它的重了).
5. lcdInfo: 6 个数字引脚, 均与 lcdTime 共用.

共花去 2 个模拟引脚和 12 个数字引脚.

数字钟功能

有五个功能模块

1. 显示时间, 日期与星期.
2. 设置时间, 日期 (闪烁).
3. 闹钟模式.
4. 倒计时模式.
5. 计时模式.

之后可以考虑 am, pm.

逻辑思路

没有 ctime 等库函数 (自己安装呢?), 所以打算自己造轮子.

由 clock 获取当前时间 currentTime, 再加上设置时间而改变的时间 deltaTime, 得到校正后的时间 correctedTime.

其中 deltaTime 包含两个整型数据, 一天内的秒数 (0 - 86400) 和天数 (-3660000 - 3660000). 由于闰年稍有复杂, 这里不再深入优化, 一个月一个月的处理.

按键简介

buttonSet: 进入设置; 下一设置

buttonEnter: 确定, 进入 / 退出.

buttonPlus: 加; 计次

buttonMinus: 减; 重置

具体逻辑

- (buttonSet) - 显示时间 (mode = 0)
- buttonSet - 时间设置 (mode = 1)
  - buttonEnter - 设置年 (submode = 1)

    注意 0 的位置留给了 "无设置", 我认为这对于减少疑惑、快速上手、便于使用来说, 是有必要的.

    - buttonPlus - 加一年
    - buttonMinus - 减一年
  - buttonEnter - 设置月 (submode = 2)
  - buttonEnter - 设置日 (submode = 3)
  - buttonEnter - 设置时 (submode = 4)
  - buttonEnter - 设置分 (submode = 5)
  - buttonEnter - 重置秒 (submode = 6)
    - buttonPlus / buttonMinus - 重置秒
  - buttonEnter - 退出时间设置 (submode = 0)
- buttonSet - 闹钟设置 (mode =2)
  - buttonEnter - 闹钟设置时 (submode = 1)
    - buttonPlus - 加一小时
    - buttonMinus - 减一小时
    - buttonSet - 启用闹钟 (可显示点) (isAlarmOn = true)
    - buttonSet - 禁用闹钟 (不显示点) (isAlarmOn = false)
  - buttonEnter - 闹钟设置分 (submode = 2, 没有必要精确到秒)
  - buttonEnter - 退出闹钟设置 (submode = 0)
- buttonSet - 倒计时模式 (mode = 3)
  - buttonEnter - 倒计时设置 (submode = 1)
    - buttonSet - 设置小时 (subsubmode = 1)
      - buttonPlus - 加一小时
      - buttonMinus - 减一小时
    - buttonSet - 设置分 (subsubmode = 2)
    - buttonSet - 设置秒 (subsubmode = 3)
  - buttonEnter - 倒计时 (submode = 2)

        - buttonSet - 开始倒计时 (subsubmode = 1)
        - buttonSet - 停止倒计时 (subsubmode = 2)
        - buttonMinus - 结束倒计时 (subsubmode = 3)
        - buttonPlus - 重置倒计时 (subsubmode = 0)
    - buttonEnter - 退出倒计时 (submode = 0)
    - ~~buttonEnter - 进入倒计时模式 (submode = 0)~~
        - ~~buttonSet - 倒计时设置 (submode = 1)~~
            - ~~buttonEnter - 设置小时 (subsubmode = 1)~~
                - ~~buttonPlus - 加一小时~~
                - ~~buttonMinus - 减一小时~~
            - ~~buttonEnter - 设置分 (subsubmode = 2)~~
            - ~~buttonEnter - 设置秒 (subsubmode = 3)~~
        - ~~buttonSet - 倒计时 (submode = 2)~~
            - ~~buttonEnter - 开始倒计时 (subsubmode = 1)~~
            - ~~buttonEnter - 停止倒计时 (subsubmode = 2)~~
            - ~~buttonMinus - 结束倒计时 (subsubmode = 3)~~
            - ~~buttonPlus - 重置倒计时 (subsubmode = 0)~~
    - ~~buttonEnter - 退出倒计时 (submode = 0)~~
- buttonSet - 计时模式 (mode = 4)
    - buttonEnter - 开始计时 (submode = 1)
    - buttonPlus - 计次
    - buttonEnter - 停止计时 (submode = 2)
    - buttonMinus - 复位 (submode = 0)
- buttonSet - 显示时间 (mode = 0)

```
// 模式
0 SHOW TIME
1 SET TIME
    0 NULL
    1 YEAR
    2 MONTH
    3 DAY
    4 HOUR
    5 MINUTE
    6 SECOND
2 ALARM MODE
    0 NULL
    1 HOUR
    2 MINUTE
3 COUNT DOWN
    0 NULL
    1 SET
        1 HOUR
        2 MIN
    2 WORK
        0 READY
        1 START
        2 STOP
```

```
        3 END
4 TIMER MODE
    0 NULL
    1 START
    2 STOP


// 星期
MON
TUE
WED
THU
FRI
SAT
SUN

// 时间
Morning (06:00 - 11:00)
Noon (11:00 - 14:00)
Afternoon (14:00 - 17:00)
Evening (17:00 - 20:00)
Night (20:00 - 06:00)

// 闹钟
ALARM ON
ALARM OFF
```

变量定义

```
#include <Wire.h>
#include <LiquidCrystal.h>
#include "DS1307.h"
#include "HD44780_LCD_PCF8574.h"

//--------------------类，变量，结构体的定义-----------------------
// 按键引脚与实例定义
// 串口监视器接数字引脚 0 和 1

const int sounderPin = 1;

const int buttonPin[4] = {6, 7, 8, 9};
// 0: buttonSet 引脚 6
// 1: buttonEnter 引脚 7
// 2: buttonPlus 引脚 8
// 3: buttonMinus 引脚 9

// 关闭是低电平，抬起是高电平
int buttonState[4] = {HIGH, HIGH, HIGH, HIGH};
bool isButtonClicked[4] = {false, false, false, false};
int reading[4] = {HIGH, HIGH, HIGH, HIGH};
int lastReading[4] = {HIGH, HIGH, HIGH, HIGH};

// 为了代码的易读性，定义如下引用
bool &buttonSet = isButtonClicked[0];
bool &buttonEnter = isButtonClicked[1];
```

```
bool &buttonPlus = isButtonClicked[2];
bool &buttonMinus = isButtonClicked[3];

// 按钮的防抖动机制
unsigned long lastDebounceTime[4] = {0, 0, 0, 0};
unsigned long debounceDelay = 50;

// 获取时间与显示时间
DS1307 clock;  // 模拟引脚 A4, A5, 共用数字引脚 11, 12
HD44780LCD lcdTime(2, 16, 0x27);  // 共用数字引脚 11, 12

// 显示其它信息
const int rs = 13,en = 10;  // 数字引脚 10, 13
const int d4 = 5,d5 = 4,d6 = 3,d7 = 2;  // 数字引脚 2, 3, 4, 5
LiquidCrystal lcdInfo(rs, en, d4, d5, d6, d7);

// 状态设置 (currentState 和 alarmMode)
// 只用一次, 就直接用 struct 了, 也不 typedef 了
// 也可以在结构体定义时直接初始化
struct ClockState {
    int mode;
    int submode;
    int subsubmode;
} currentState = {0, 0, 0};

bool isAlarmOn = false;

// 时间记录
class Time {
public:
    Time(int hour = 0, int minute = 0, int second = 0)
        : hour(hour), minute(minute), second(second) {}
public:
    int hour;
    int minute;
    int second;
} adtime, ddtime, alarmTime, countDownTime, countTime;
// 即计时上升的 ascending time, 倒计时下降的 descending time 和闹钟时间

class TimeWithDate: public Time {
public:
    TimeWithDate(int year = 0, int month = 0, int day = 0, int hour = 0, int
minute = 0, int second = 0)
        : year(year), month(month), day(day), Time(hour, minute, second) {}
protected:
    int year;
    int month;
    int weekday;
    int day;
} currentTime, correctedTime;

class DayAndSecond {
public:
    DayAndSecond(int day = 0, int second = 0)
        : day(day), second(second) {}
```

```
    void addDay(int deltaDay) {day += deltaDay;}
    void addSecond(int deltaSecond) {
        second += deltaSecond;
        day += second / (3600 * 24);
        second %= 3600 * 24;
    }
    int getDay() {return day;}
    int getSecond() {return second;}
protected:
    int day;
    int second;
} deltaTime;

int daysOfMonth[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
//                month: 1   2   3   4   5   6   7   8   9  10  11  12
```

倒计时的弃用设计

buttonEnter

buttonExit

buttonNext

buttonLast

- buttonEnter - 进入设置 (设置时间)
    - buttonEnter - 设置年
    - buttonNext - 设置月
    - buttonNext - 设置日
    - buttonNext - 设置时
    - buttonNext - 设置分
    - buttonNext - 重置秒
    - buttonExit - 退出时间设置
- buttonNext - 闹钟
- buttonNext - 倒计时
- buttonNext - 计时
- buttonExit - 退出

课程参考

proteus蜂鸣器不响-百度经验 (baidu.com)

(14条消息) Arduino学习(五) 蜂鸣器实验*JoStudio的博客-CSDN博客*arduino蜂鸣器停止发声

(14条消息) proteus 蜂鸣器系列*weixin_41980642的博客-CSDN博客*proteus蜂鸣器

(14条消息) proteus仿真有源/无源蜂鸣器*Doee hc的博客-CSDN博客*proteus蜂鸣器怎么找

RGB颜色代码表🎨 (rapidtables.org)

啃芝士的视频. ⭐

(14条消息) Arduino ide tab缩讲和自动缩讲 4 字节 设置*Yangs_noerr的博客-CSDN博客*arduino tab