

ESP32 PWM信号的输出

2021年7月4日 14:35

ESP32有一个 PWM 控制器与16个独立的渠道，可以配置产生 PWM 信号与不同的性质。

下面是使用 Arduino IDE 产生 PWM 的步骤:

- 1.定义产生pwm的引脚和选择一个 PWM 通道，从0到15有16个通道。
- 2.然后，你需要设置 PWM 信号的频率。ESP32最高的频率是312.515khz，设置打了也没关系，它会按最高频率输出。
- 3.你还需要设置信号的占空比分辨率: 你的分辨率从1到16位。我们将使用8位分辨率，这意味着您可以控制使用值从0到255来控制占空比，公式是 $dutycycle=(2^{\text{分辨率}})*\text{占空比}$
- 4.接下来，您需要指定信号将出现在哪个 GPIO 或 GPIO 上。为此，你可以使用以下函数:

`ledcAttachPin(GPIO, channel)`

这个函数接受两个参数。第一个是将输出信号的 GPIO，第二个是将产生信号的通道。

5. 最后，为了使用 PWM 来控制 LED 的亮度，你需要用下面这个函数:

`ledcWrite(channel, dutycycle)`

这个函数接受作为参数的通道，正在生成的 PWM 信号和第三步算出来的dutycycle。

注意: 你可以使用任何你想要的引脚，只要它可以作为一个输出。所有可以作为输出的引脚都可以用作 PWM 引脚。

代码:

// 定义pwm引脚的序号，可以自己设定，其他引脚也可以，比如我用的是G13

const int pwmPin = 13; // 13对应于 GPIO13

// 设定 PWM 特性

const int freq = 200000; //频率 最大频率是312.515khz，你设大了也只能输出最大频率，产生的波形感觉还可以

const int pwmChannel = 0; //通道

const int resolution = 8; //分辨率 2的8次方=256

int dutycycle = 0;

void setup(){

// 配置 PWM 功能(初始化)

ledcSetup(pwmChannel, freq, resolution);

// 将通道连接至 GPIO 以便控制

ledcAttachPin(pwmPin, pwmChannel);

n = 256*0.5; // dutycycle=2^resolution*占空比

ledcWrite(pwmChannel, dutycycle);

}

void loop(){

}

连接到示波器可以看到200khz的方波，波形还算可以，没有明显的失真，看下面图:

