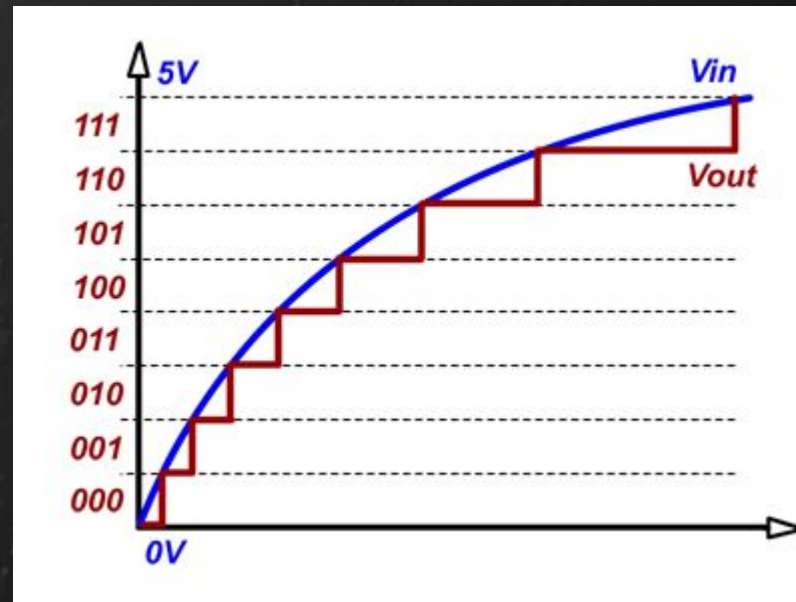


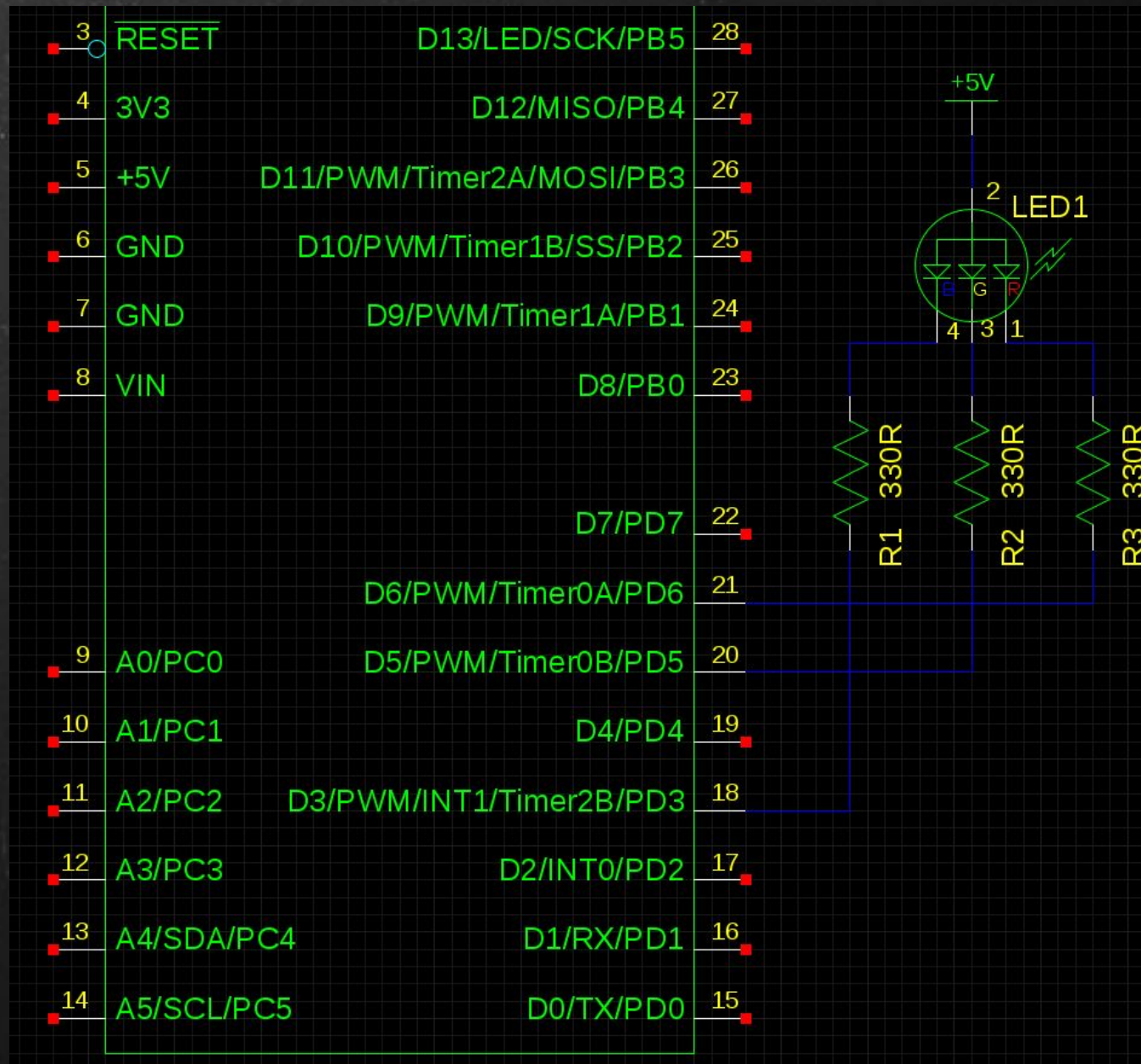
# IDD001 Lecture 5:

## 当模拟遇到数字



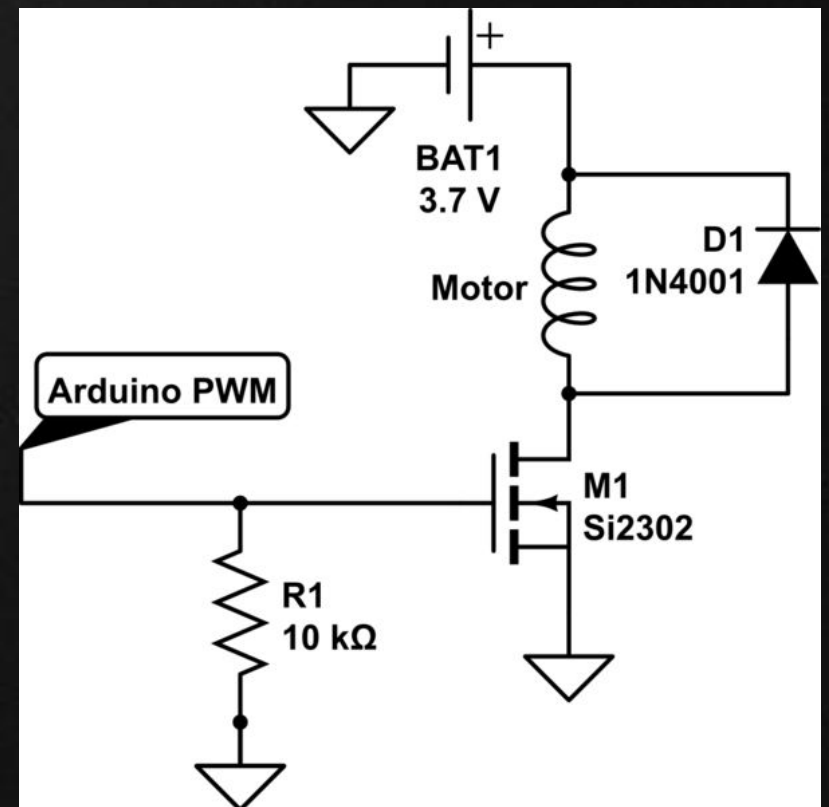
Atommann <[atommann@gmail.com](mailto:atommann@gmail.com)>  
2017 Fall

# PWM



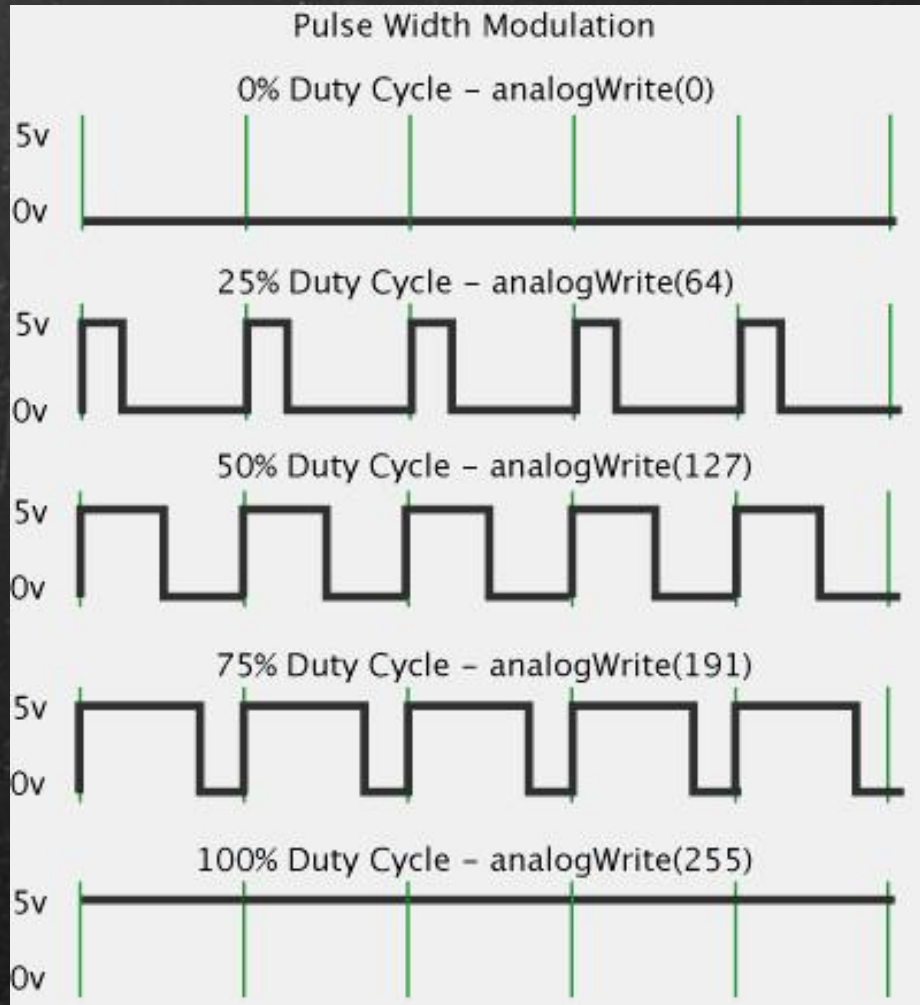
# PWM

- PWM = Pulse Width Modulation
- 是一种只占用一个 I/O 的 Digital to Analog Converter(DAC)
- 一种极有用的技术, 应用广泛, 比如:
  - LED 调光
  - 控制 motor 的速度
  - 电话拨号 DTMF
  - 其它你没想到的地方





# PWM: 原理



`analogWrite(val);`

$$V_{out} = 5V * \frac{val}{255}$$

$$\text{Duty Cycle} = val/255$$

原理:电压/电流做功,取平均值。

## PWM: Arduino 提供的抽象

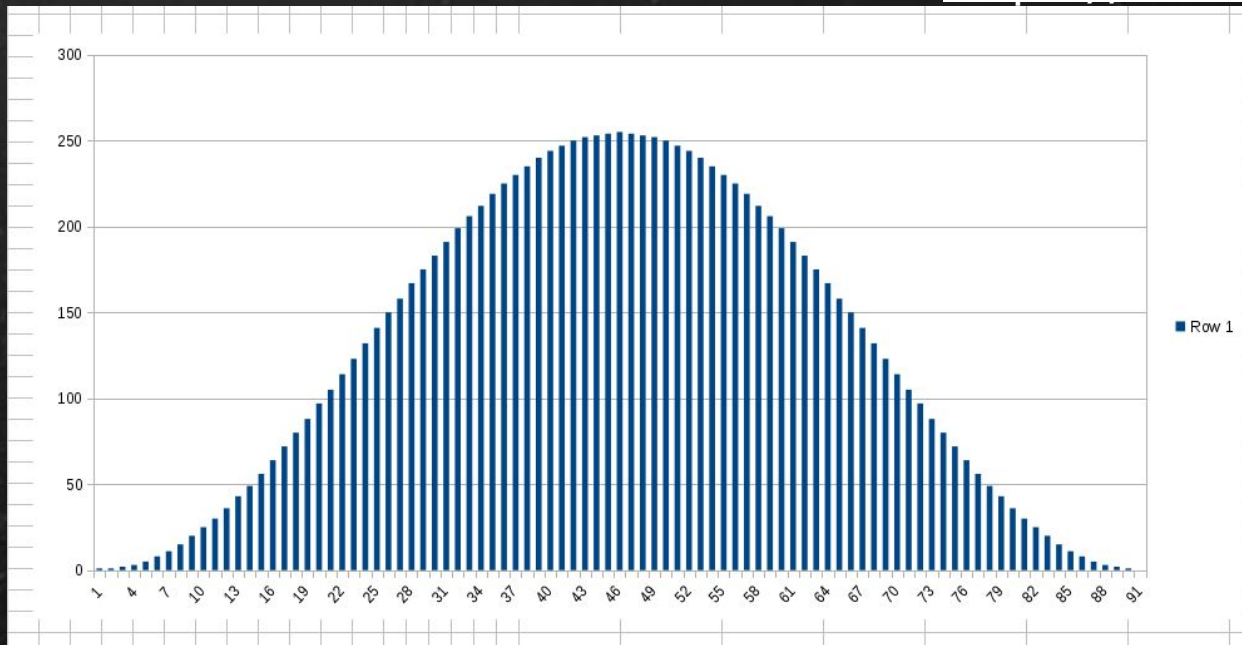
- 函数 `analogWrite()`
- 和 `analogRead()` 没有关系
- 可以想象成输出一个连续的电压值(实际上不是)
- `analogWrite(0)`: 输出 0V
- `analogWrite(255)`: 输出 5V

# PWM: 呼吸灯演示



来源:

<https://www.adafruit.com/icufflinks>



# PWM: 呼吸灯演示(代码)

```
// 呼吸灯
// 在 Uno 上 pin 3, 5, 6, 9, 10, 11 支持 analogWrite()
// 也就是带 ~ 的 pin
#define LED_PIN 3

uint8_t sine_table[] = {
    1, 1, 2, 3, 5, 8, 11, 15, 20, 25,
    30, 36, 43, 49, 56, 64, 72, 80, 88, 97,
    105, 114, 123, 132, 141, 150, 158, 167, 175, 183,
    191, 199, 206, 212, 219, 225, 230, 235, 240, 244,
    247, 250, 252, 253, 254, 255, 254, 253, 252, 250,
    247, 244, 240, 235, 230, 225, 219, 212, 206, 199,
    191, 183, 175, 167, 158, 150, 141, 132, 123, 114,
    105, 97, 88, 80, 72, 64, 56, 49, 43, 36,
    30, 25, 20, 15, 11, 8, 5, 3, 2, 1, 0};

void setup() {
    //pinMode(LED_PIN, OUTPUT); // analogWrite() 不需要把 pin 设置成输出
}

void loop() {
    for (uint8_t i = 0; i < 91; i++) {
        analogWrite(LED_PIN, sine_table[i]);
        delay(50);
    }
}
```



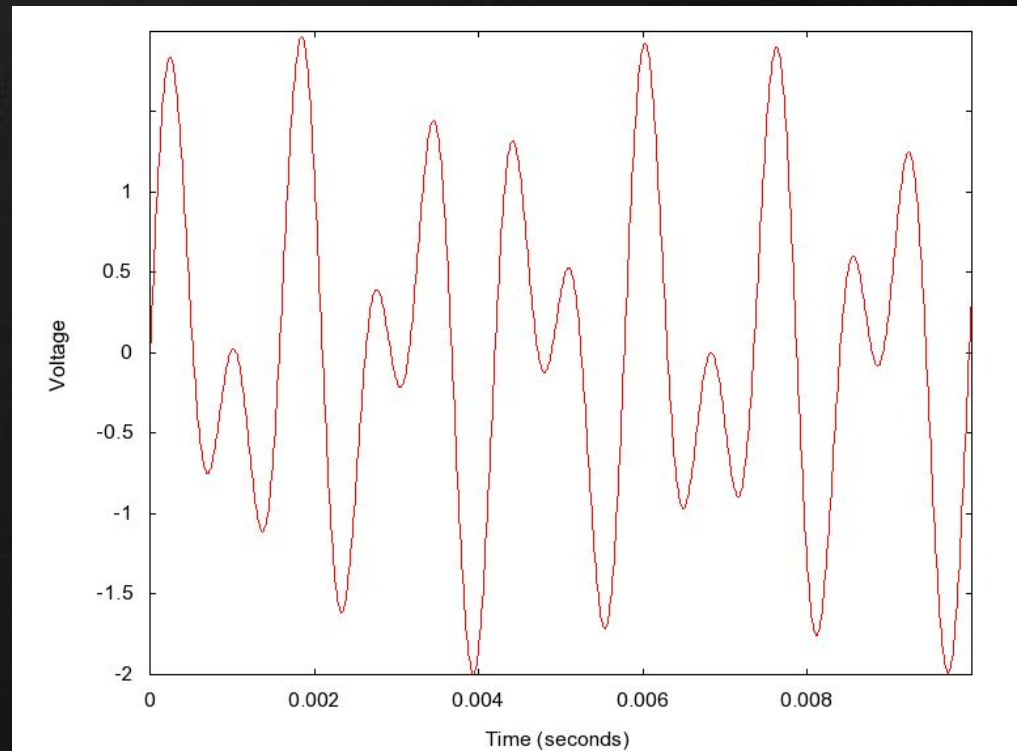
# PWM 应用案例:DTMF(1)

- DTMF = Dual-Tone Multi-Frequency
- 在 Bell Lab 被发明
- 可以用 PWM 来实现
- DTMF 物理攻击

DTMF keypad frequencies (with sound clips)

	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

电话键盘和频率的对应

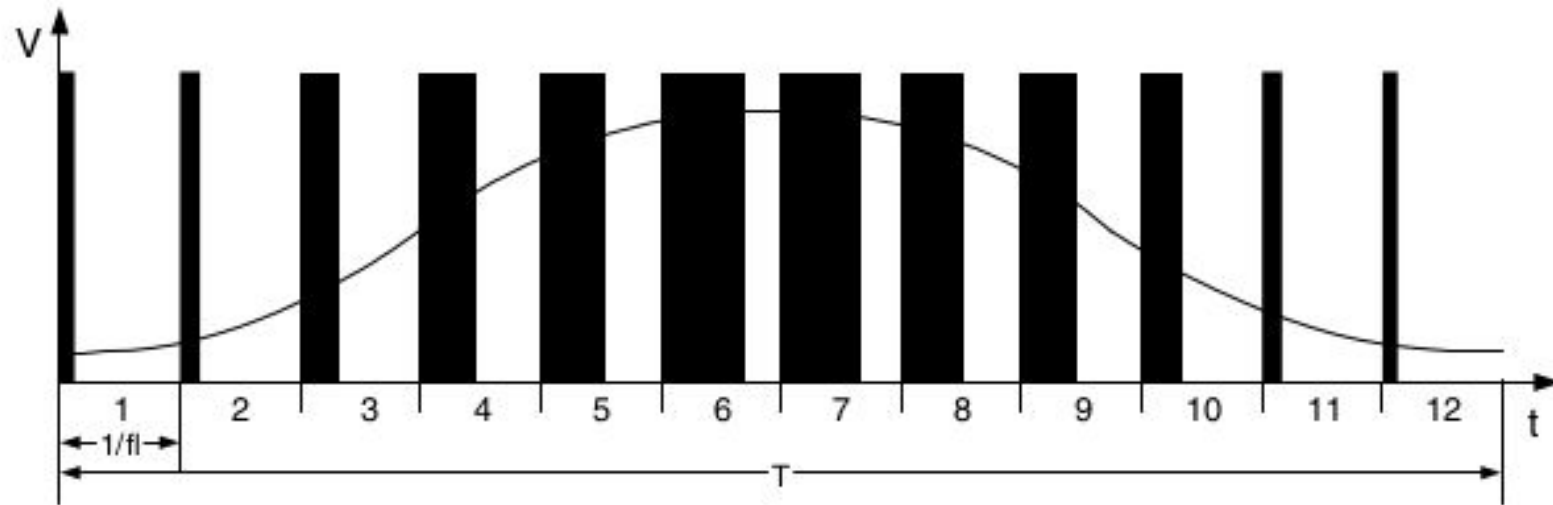


数字 1 的波形



# PWM 应用案例:DTMF(2)

Figure 3. Generating a Sine Wave with PWM



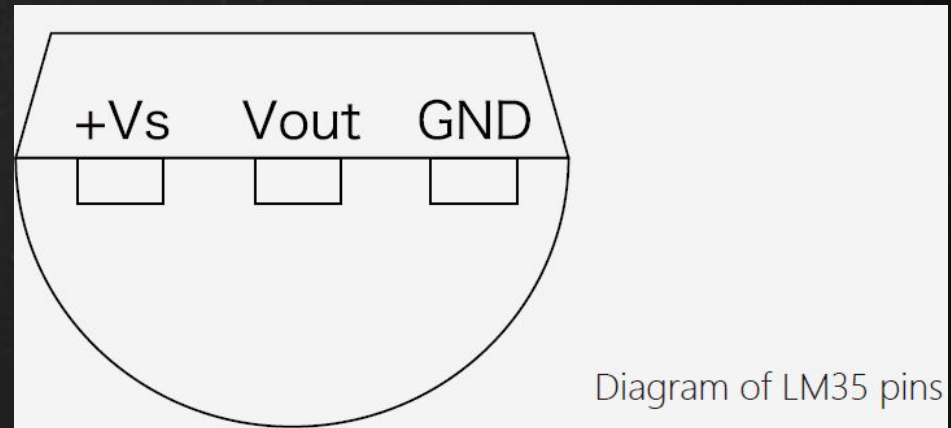
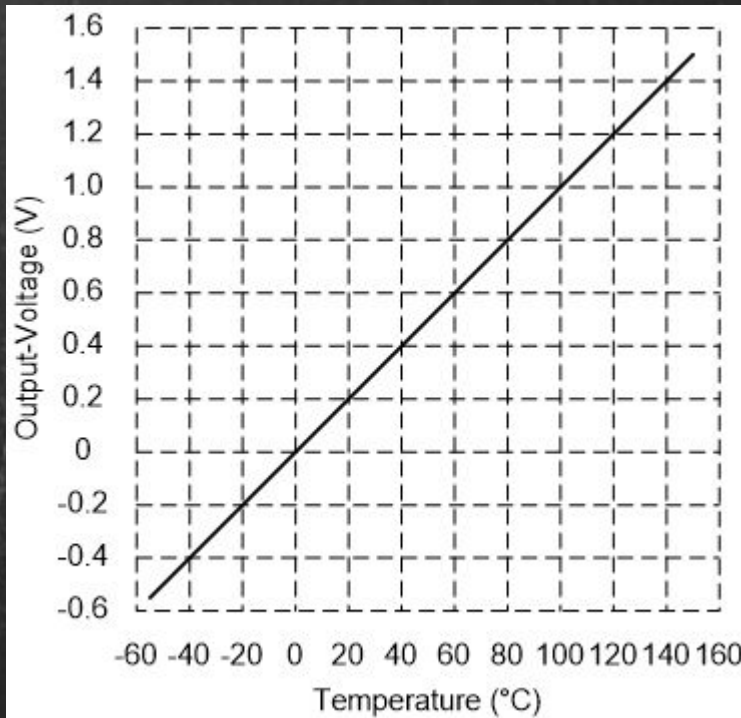
用 PWM 生成 Sine Wave

详见 Appnote AVR314: DTMF Generator

<http://www.atmel.com/images/doc1982.pdf>

挑战:你能用 Arduino 把你的手机拨通吗?

# 实验：用 LM35 读温度



LM35 的 pinout  
Bottom View

LM35 Vout 的输出电压和温度呈线性关系  
可以用万用表测得温度值

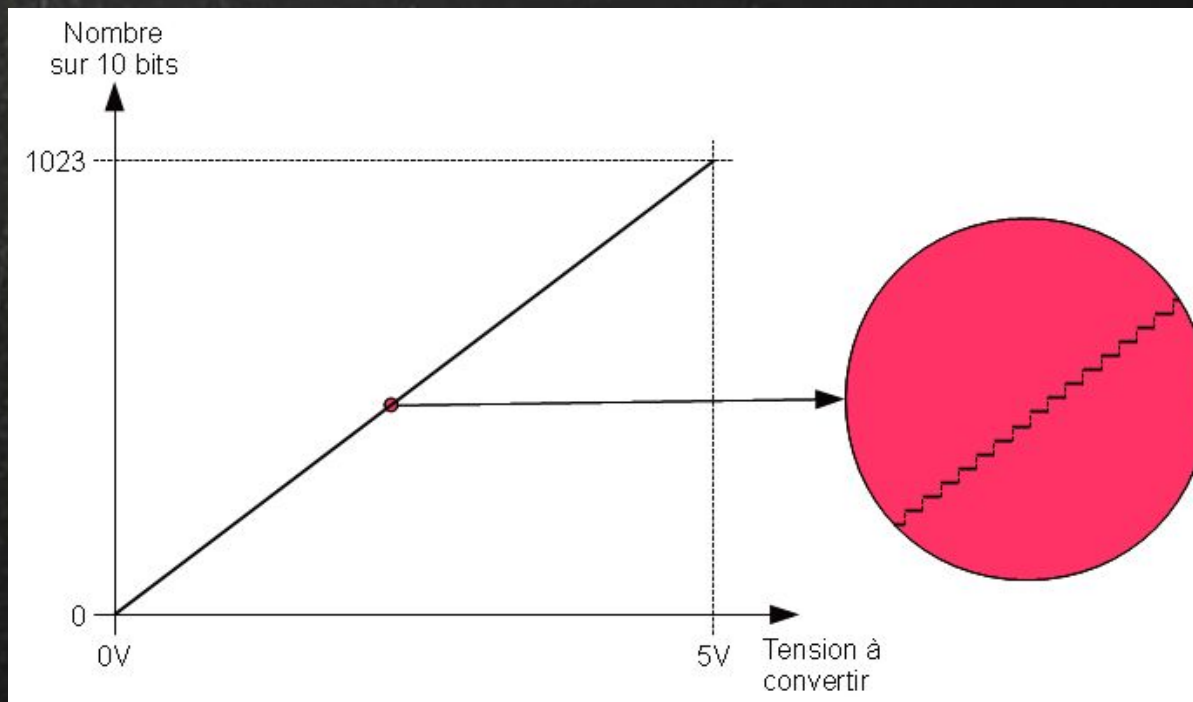
来源：

<http://depokinstruments.blogspot.jp/2016/03/lm35-precision-centigrade-temperature.html>

推荐阅读：<https://learn.adafruit.com/tmp36-temperature-sensor/overview>

# 当模拟遇上数字 - ADC

- ADC = Analog to Digital Converter 即: 模拟/数字转换器, 有时简称为“模数转换器”
- Arduino 提供的抽象
  - `analogRead()`
  - `analogReference()`



- 10位, 0~1023 ( $2^{10}-1$ )
- 10k samples/s, 约 100us
- 把电压值映射到 0 到 1023

# analogReference()

- 参数

- DEFAULT: 默认参考值, 5V on 5V Arduino 或 3.3V on 3.3V Arduino
- INTERNAL: IC 内部参考电压 ATmega168/328 上是 1.1 V
- INTERNAL1V1: 内部 1.1V (Arduino Mega only)
- INTERNAL2V56: 内部 2.56V<sup>[1]</sup> (Arduino Mega only)
- EXTERNAL: 由用户接到 AREF 引脚上 (0 to 5V only)

<https://www.arduino.cc/en/Reference/AnalogReference>

[1]  $256 = 2^8$ , 这不是一个巧合。

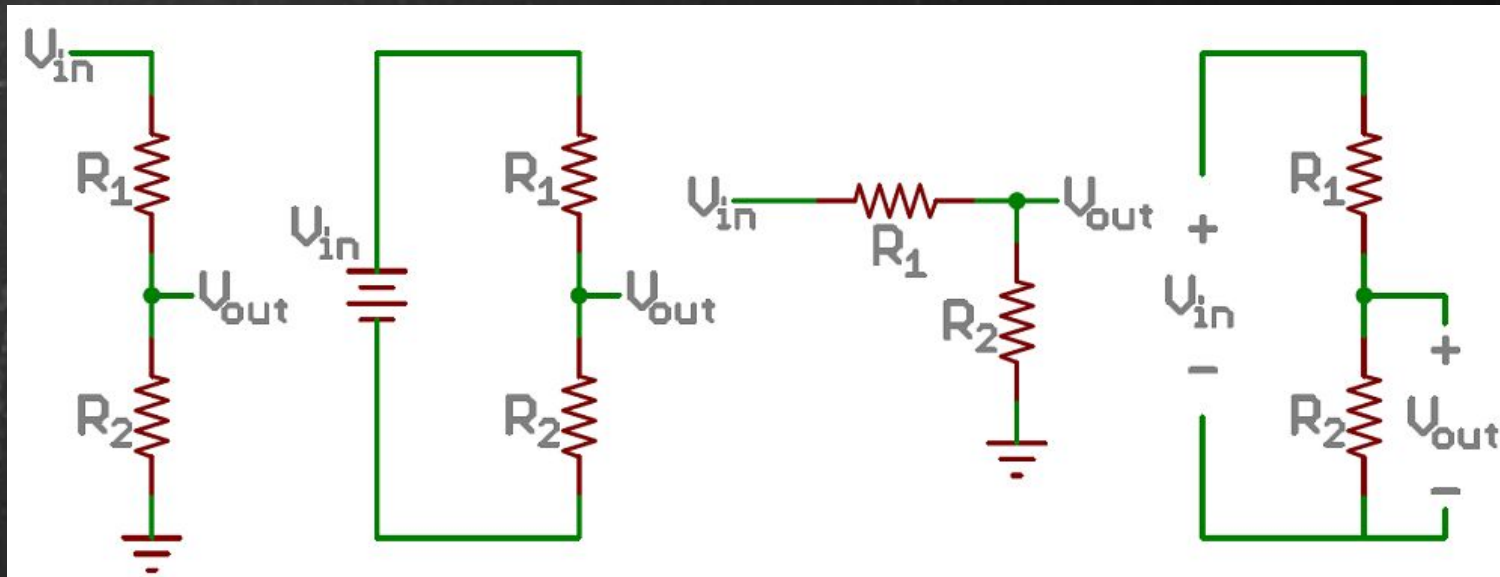


# 演示：用 Arduino 观察 RC 充电曲线

- 可以把 Arduino 作为你的实验工具

<https://atommann.github.io/learn/rc-circuit/rc-circuit.html>

# 分压器 Voltage Divider

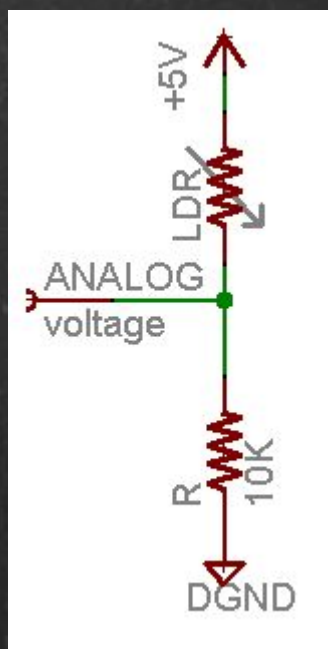


$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

Your turn:  
推导这个公式

来源: <https://learn.sparkfun.com/tutorials/voltage-dividers>

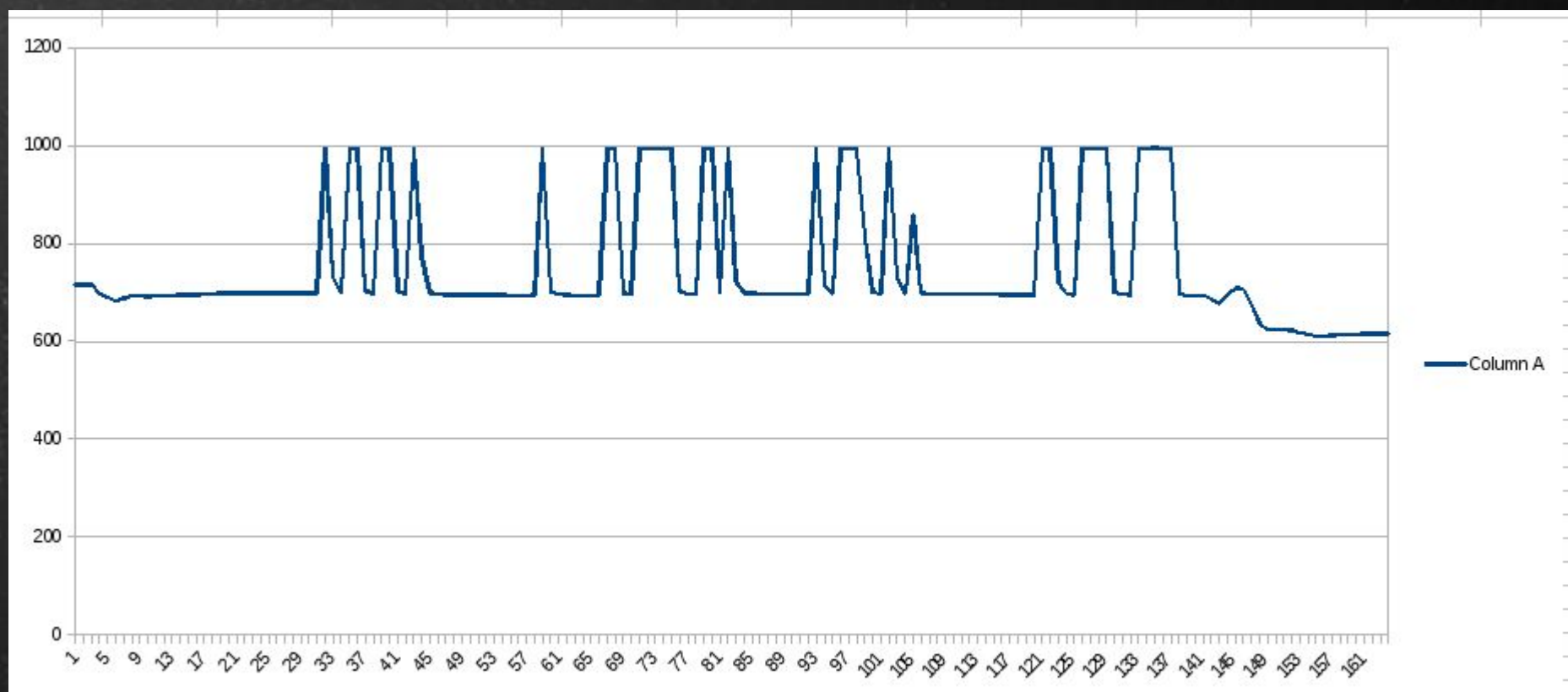
# 课堂实验：读光敏电阻的阻值



Ambient light like...	Ambient light (lux)	Photocell resistance ( $\Omega$ )	LDR + R ( $\Omega$ )	Current thru LDR + R	Voltage across R
Dim hallway	0.1 lux	600 K $\Omega$	610 K $\Omega$	0.008 mA	0.1 V
Moonlit night	1 lux	70 K $\Omega$	80 K $\Omega$	0.07 mA	0.6 V
Dark room	10 lux	10 K $\Omega$	20 K $\Omega$	0.25 mA	2.5 V
Dark overcast day / Bright room	100 lux	1.5 K $\Omega$	11.5 K $\Omega$	0.43 mA	4.3 V
Overcast day	1000 lux	300 $\Omega$	10.03 K $\Omega$	0.5 mA	5V

来源: <https://learn.adafruit.com/photocells/using-a-photocell>

# Photocell 奇思妙想



Your turn!

Do you have any idea that you can do something with this little guy? A scanner?



# 推荐图书:《爱上制作》



# 作业说明

- 写文档
  - 中文, 可以
  - English, OK
  - HTML
  - Github pages
  - 独立思考, You can do it!

## 版权说明 Copyright Issue

本 slide 的很多图片来自 Wikipedia 和 Google 搜索, 我感谢原创作者的杰出贡献, 如有任何违反版权的事宜, 请联系我。

Lot of diagrams/pictures in this slide are from Wikipedia and Google search, I appreciate the author's cool job, if there are some licence issues please email me.

Atommann <[atommann@gmail.com](mailto:atommann@gmail.com)>