

常量:

- HIGH | LOW 表示数字 IO 口的电平, HIGH 表示高电平 (1), LOW 表示低电平 (0)。
- INPUT | OUTPUT 表示数字 IO 口的方向, INPUT 表示输入 (高阻态), OUTPUT 表示输出 (AVR 能提供5V 电压 40mA 电流)。
- true | false true 表示真 (1), false 表示假 (0)。

结构

- void setup() 初始化变量, 管脚模式, 调用库函数等
- void loop() 连续执行函数内的语句

数字 I/O

- pinMode(pin, mode) 数字 IO 口输入输出模式定义函数, pin 表示为0~13, mode 表示为 INPUT 或 OUTPUT。
- digitalWrite(pin, value) 数字 IO 口输出电平定义函数, pin 表示为0~13, value 表示为 HIGH 或 LOW。比如定义 HIGH 可以驱动 LED。
- int digitalRead(pin) 数字 IO 口读输入电平函数, pin 表示为0~13, value 表示为 HIGH 或 LOW。比如可以读数字传感器。

模拟 I/O

- int analogRead(pin) 模拟 IO 口读函数, pin 表示为0~5 (Arduino Diecimila 为0~5, Arduino nano 为0~7)。比如可以读模拟传感器 (10位 AD, 0~5V 表示为0~1023)。
- analogWrite(pin, value) - PWM 数字 IO 口 PWM 输出函数, Arduino 数字 IO 口标注了 PWM 的 IO 口可使用该函数, pin 表示3, 5, 6, 9, 10, 11, value 表示为0~255。比如可用于电机 PWM 调速或音乐播放。

扩展 I/O

- shiftOut(dataPin, clockPin, bitOrder, value) SPI 外部 IO 扩展函数, 通常使用带 SPI 接口的 74HC595做8个 IO 扩展, dataPin 为数据口, clockPin 为时钟口, bitOrder 为数据传输方向 (**MSBFIRST** 高位在前, **LSBFIRST** 低位在前), value 表示所要传送的数据 (0~255), 另外还需要一个 IO 口做 74HC595的使能控制。
- unsigned long pulseIn(pin, value) 脉冲长度记录函数, 返回时间参数 (us), pin 表示为0~13, value 为 HIGH 或 LOW。比如 value 为 HIGH, 那么当 pin 输入为高电平时, 开始计时, 当 pin 输入为低电平时, 停止计时, 然后返回该时间。

时间函数

- unsigned long millis() 返回时间函数 (单位 ms), 该函数是指, 当程序运行就开始计时并返回记录的参数, 该参数溢出大概需要50天时间。
- delay(ms) 延时函数 (单位 ms)。
- delayMicroseconds(us) 延时函数 (单位 us)。

数学函数

- min(x, y) 求最小值
- max(x, y) 求最大值
- abs(x) 计算绝对值
- constrain(x, a, b) 约束函数，下限 a，上限 b，x 必须在 ab 之间才能返回。
- map(value, fromLow, fromHigh, toLow, toHigh) 约束函数，value 必须在 fromLow 与 toLow 之间和 fromHigh 与 toHigh 之间。
- pow(base, exponent) 开方函数，base 的 exponent 次方。
- sq(x) 平方
- sqrt(x) 开根号

三角函数

- sin(rad)
- cos(rad)
- tan(rad)

随机数函数

- randomSeed(seed) 随机数端口定义函数，seed 表示读模拟口 analogRead(pin)函数。
- long random(max) 随机数函数，返回数据大于等于0，小于 max。
- long random(min, max) 随机数函数，返回数据大于等于 min，小于 max。

外部中断函数

- attachInterrupt(interrupt, , mode) 外部中断只能用到数字 IO 口2和3, interrupt 表示中断口初始0或1, 表示一个功能函数, mode: **LOW** 低电平中断, **CHANGE** 有变化就中断, **RISING** 上升沿中断, **FALLING** 下降沿中断。
- detachInterrupt(interrupt) 中断开关, interrupt=1 开, interrupt=0 关。

中断使能函数

- interrupts() 使能中断
- noInterrupts() 禁止中断

串口收发函数

- Serial.begin(speed) 串口定义波特率函数，speed 表示波特率，如9600，19200等。
- int Serial.available() 判断缓冲器状态。
- int Serial.read() 读串口并返回收到参数。
- Serial.flush() 清空缓冲器。
- Serial.print(data) 串口输出数据。
- Serial.println(data) 串口输出数据并带回车符。

/*****Arduino 语言库文件*****/

官方库文件

- [EEPROM](#) - EEPROM 读写程序库
- [Ethernet](#) - 以太网控制器程序库
- [LiquidCrystal](#) - LCD 控制程序库
- [Servo](#) - 舵机控制程序库
- [SoftwareSerial](#) - 任何数字 IO 口模拟串口程序库
- [Stepper](#) - 步进电机控制程序库
- [Wire](#) - TWI/I2C 总线程序库
- [Matrix](#) - LED 矩阵控制程序库
- [Sprite](#) - LED 矩阵图象处理控制程序库

非官方库文件

- [DateTime](#) - a library for keeping track of the current date and time in software.
- [Debounce](#) - for reading noisy digital inputs (e.g. from buttons)
- [Firmata](#) - for communicating with applications on the computer using a standard serial protocol.
- [GLCD](#) - graphics routines for LCD based on the KS0108 or equivalent chipset.
- [LCD](#) - control LCDs (using 8 data lines)
- [LCD 4 Bit](#) - control LCDs (using 4 data lines)
- [LedControl](#) - for controlling LED matrices or seven-segment displays with a MAX7221 or MAX7219.
- [LedControl](#) - an alternative to the Matrix library for driving multiple LEDs with Maxim chips.
- [Messenger](#) - for processing text-based messages from the computer
- [Metro](#) - help you time actions at regular intervals
- [MsTimer2](#) - uses the timer 2 interrupt to trigger an action every N milliseconds.
- [OneWire](#) - control devices (from Dallas Semiconductor) that use the One Wire protocol.
- [PS2Keyboard](#) - read characters from a PS2 keyboard.
- [Servo](#) - provides software support for Servo motors on any pins.
- [Servotimer1](#) - provides hardware support for Servo motors on pins 9 and 10
- [Simple Message System](#) - send messages between Arduino and the computer
- [SSerial2Mobile](#) - send text messages or emails using a cell phone (via AT commands over software serial)
- [TextString](#) - handle strings
- [TLC5940](#) - 16 channel 12 bit PWM controller.
- [X10](#) - Sending X10 signals over AC power lines

/*****