

# IDD001 Lecture 3:

## C 语言的三个和弦

```
void setup()  
{  
    Serial.begin(9600);  
    Serial.println("hello, you.");  
}  
  
void loop()  
{  
  
}
```

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

欲速则不达。--《论语》

千里之行始于足下。-- 老子 (2500 BC)

# 和计算机交谈

- Hey, Arduino, please blink the LED on pin 13, time interval 500 milliseconds.
  - Arduino: "yes, master."
  - 未来的作业！Can we do it?
- What's the difference between Forth and other high-level languages? To put it very briefly: it has to do with the compromise between man and computer. A language should be designed for the convenience of its human users, but at the same time for compatibility with the operation of the computer.<sup>[1]</sup>

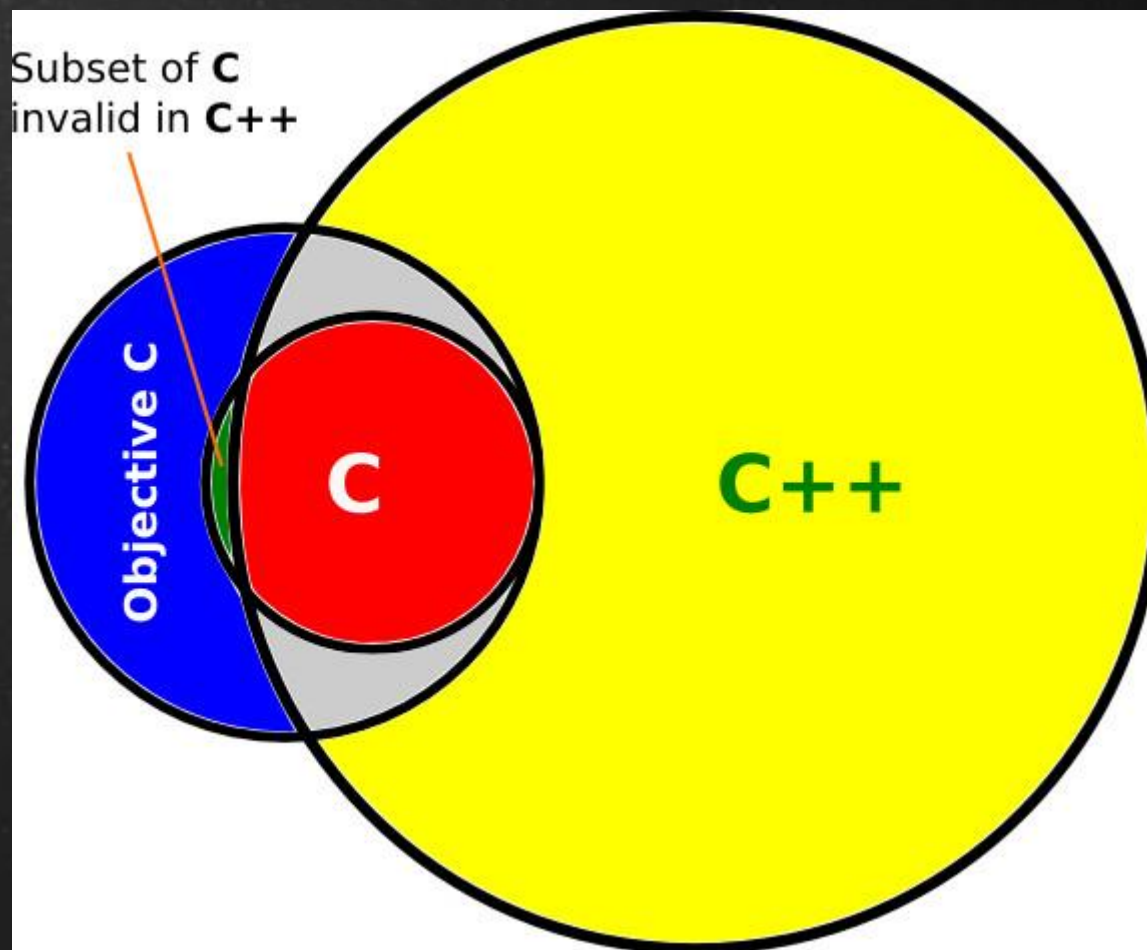
[1] Introduction for Beginners: What is a Computer Language? (Online book "Starting Forth")

# C 语言简介

- 1972 年诞生于 Bell Lab, Dennis Ritchie
  - 有趣的事:Dennis 在编译器里植入后门
  - 影响了很多后世语言, Java, PHP, Verilog...
- 高级语言的岁数
  - FORTRAN, 1957, 60 岁
  - LISP, 1958, 59 岁
  - COBOL, 1959, 58 岁
- Unix, GNU/Linux, Windows, TCP/IP 协议都用 C 编写
- 以及大量重要的软件
- 在嵌入式系统中大量应用
  - Arduino 也是一台计算机
- 几乎每种计算机都支持 C 编程
- 可移植性好(如果你写的是标准 C)
- 可直接操纵底层硬件



# C, C++, Arduino 语言



图片来源 [Stackoverflow](#)

# Arduino 语言

- Arduino IDE 底层是调用 gcc 编译器
  - 同时支持 C 和 C++
- 对 C++ 的特性部分支持
- 可以完全用 C 语言对 Arduino 编程
- 从 C 开始, 可过渡到面向对象
- Serial 库是一个典型的 C++ 库

# setup() 和 loop()

`void setup()` // setup 只在上电时运行一次

```
{  
  Serial.begin(9600); //设置波特率, 单位 bit/s  
  Serial.println("hello, you."); //打印到串口并断行  
}
```

`void loop()` // 永远循环, 直到断电

```
{  
  
}
```

# 注释

- 写给人看, 编译器和 CPU 完全忽略它
- 注释很重要
  - 给自己看, 给别人看
- 单行注释
- 多行注释 以 `/*` 开始, 以 `*/` 为结束

```
// 这里我们让 LED 闪烁 10 次然后停止
```

```
/*
```

```
    Function name : LCD_Init
```

```
    Returns      : None
```

```
    Parameters   : None
```

```
    Purpose      : Initialize LCD_displayData buffer.  
                  Set up the LCD (timing, contrast, etc.)
```

```
*/
```

Thus, programs must be written for people to read, and only incidentally for machines to execute. -- SICP



# 变量

- 内存中的单元格
  - 变量有一个名字, 方便我们引用
  - 变量占用多少个内存单元?
  - 变量的值可以被修改
- 例子

```
unsigned char seconds = 0;  
// 变量名: seconds  
// 占用内存: 一个 byte (i.e. 8 bits)  
// 变量的值: 0
```

Range: 0 to 255

0

seconds

# 全局变量和局部变量

## 例子

`uint8_t counter = 0;` //全局变量, 定义在所有函数之外, 在整个程序的任何地方可访问

```
void setup() {  
    int i; //局部变量, 定义在某个函数内部, 只在这个函数中能访问
```

```
    for(i = 0; i < 8; i++)  
    {  
        pinMode(i, OUTPUT);  
    }  
}
```

```
void loop() {  
  
    counter = counter + 1; //使用了全局变量  
    i = 100; // 出错  
    delay(1000);  
    ...  
}
```

# 数据类型

- C 语言是有类型的编程语言
  - 变量是什么类型, 在内存中占用几个 byte
  - 变量的取值范围
  - 问题: 一个整数 int 在内存用几个字节?
- 有些编程语言没有类型, 如 Python

推荐相关阅读 [Data Types in Arduino](#)

## 数据类型(2)

Data Type	Size in Bits	Data Range/Usage
unsigned char	8-bit	0 to 255
char	8-bit	-128 to +127
unsigned int	16-bit	0 to 65,535
int	16-bit	-32,768 to +32,767
unsigned long	32-bit	0 to 4,294,967,295
long	32-bit	-2,147,483,648 to +2,147,483,648
float	32-bit	$\pm 1.175e-38$ to $\pm 3.402e38$
double	32-bit	$\pm 1.175e-38$ to $\pm 3.402e38$

数据会溢出, 归 0

$2^{32} - 1 = 4294967295$  毫秒

1 秒 = 1000 毫秒

$(4294967295/1000)/(24*60*60) \approx 50$  天

上面的计算和 millis() 函数有关

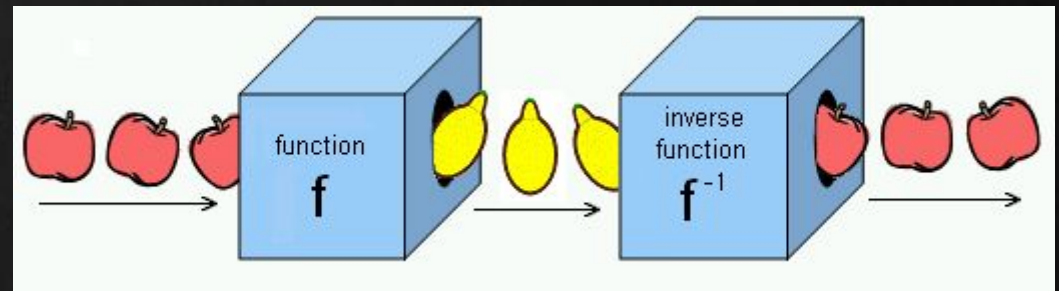


溢出  $1 + 1 = 0$

- 课堂作业：用已有知识构想一个测试实例
  - 在 Arduino 上测试
  - LED?
  - 串口打印？
  - 加法？

# function 函数/函式

- 把一块代码放在一起，给它起个名字
- 函数一旦写好，就可以在各个地方反复调用它
- 库函数，随时可用的工具库
- 每个东西都有一个名字
  - subroutine
  - procedure
  - function
  - routine
  - method
  - subprogram
  - callable unit
  - word (in Forth)
  - Whatever!



Pigs → [function] → sausages → [function<sup>-1</sup>] → Pigs

# function 演示

// 功能：把时间变量写到 8 个 LED 上显示

// 输入参数：data, 要显示的 byte 值

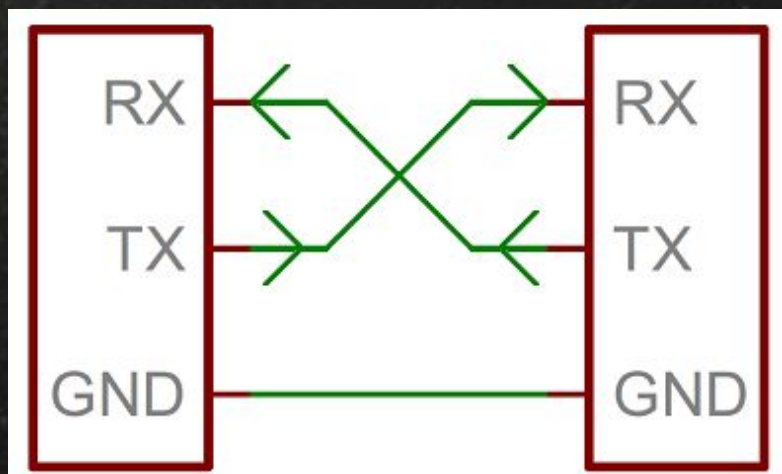
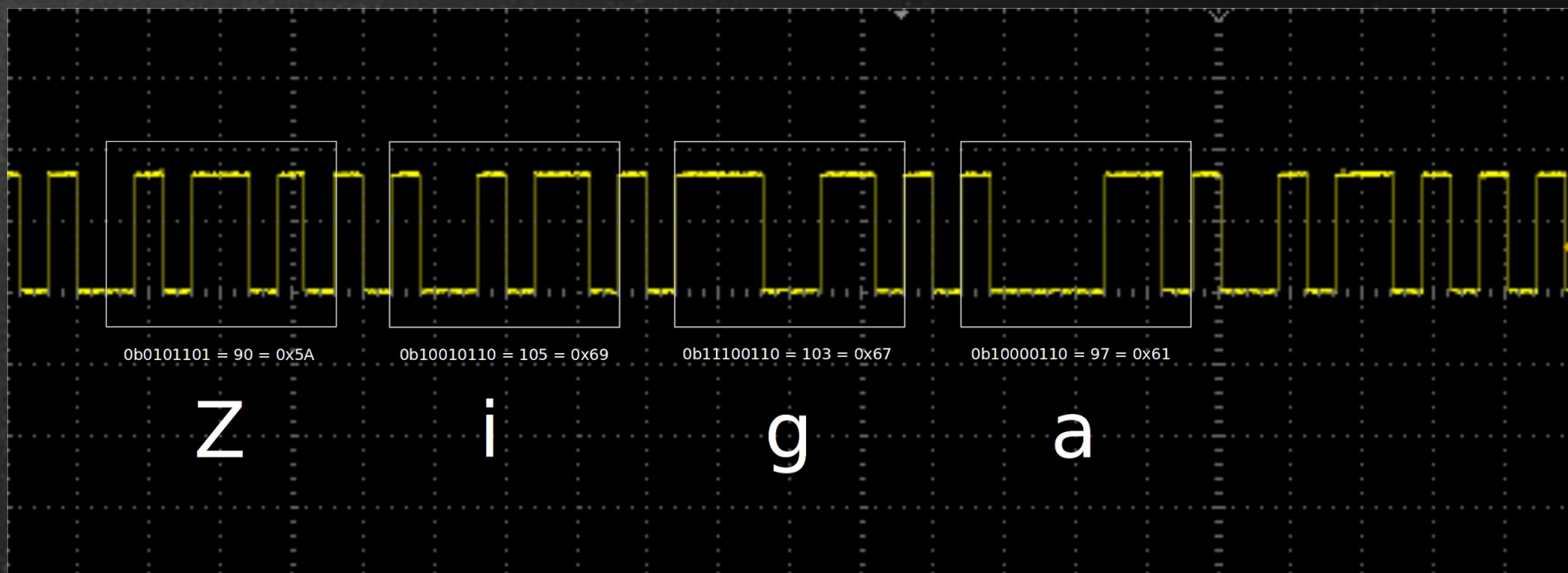
// 返回值：无

```
void put_byte(uint8_t data)
{
    digitalWrite(7, _BV(7) & data);
    digitalWrite(5, _BV(6) & data);
    digitalWrite(6, _BV(5) & data);
    digitalWrite(4, _BV(4) & data);
    digitalWrite(3, _BV(3) & data);
    digitalWrite(2, _BV(2) & data);
    digitalWrite(1, _BV(1) & data);
    digitalWrite(0, _BV(0) & data);
}
```

// 功能：把两个整数相加，返回和

```
int add_two(int a, int b)
{
    int c;
    c = a + b;
    return c;
}
```

# 一个古老而有用的工具：串口



- 简单强大的调试工具
- 有些传感器用串口作为数据接口
  - 演示 CO2 Sensor
  - GPS 模块
- 以后我们会看到更多例子



# Arduino 的串口函数

- 可以随时打印变量的值, 知道程序中运行的状态
- Arduino 向外打印数据
  - `Serial.print(string or variable name);`
  - `Serial.println(string or variable name);`<sup>[1]</sup>
- Arduino 从串口读取数据
  - `Serial.read();`
- 串口函数的详细使用说明和例子  
<https://www.arduino.cc/en/Reference/Serial>

[1] println 后面的 ln 是什么意思？这是历史原因造成的。Pascal 语言有 write 和 writeln 函数，write 输出一个字符串后光标停留在末尾，writeln (ln 是 "line" 的缩写) 打印完字符串后把光标移到下一行的开始处。别的语言继承了这个名字。

# Arduino 串口 - 演示

```
// control the onboard LED via the serial port
char incomingByte = 0;    // for incoming serial data

void setup() {
    pinMode(13, OUTPUT);
    // opens serial port, sets data rate to 9600 bps
    Serial.begin(9600);
}

void loop() {
    // send data only when you receive data:
    if (Serial.available() > 0) {
        // read the incoming byte:
        incomingByte = Serial.read();

        // control the LED according to the received letter
        if (incomingByte == 'h')
            digitalWrite(13, HIGH);
        else if (incomingByte == 'l') // letter l not number 1
            digitalWrite(13, LOW);

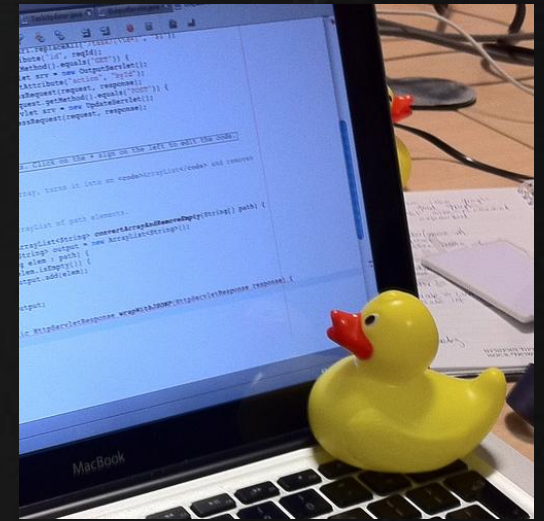
        Serial.print("Done! Master.");
    }
}
```

# 编程解决问题的思路

- 把大问题分解成小问题
  - 在编程/读别人代码时你会发现这个现象
  - 演示：读电表数据的 python 代码
- 分而治之 (Divide and Conquer)
  - 一种写算法的范式，目前还用不到
- 先别急于优化程序
  - Make it Work.
  - Make it Right.
  - Make it Fast.

# 编程提示

- 看书和例程时手工敲入代码，一边敲一边思考
  - 别总是 copy & paste
- Learning by doing.
- Don't panic.
  - 代码出错别害怕，从错误学习，debug, debug and debug.
- 用纸和笔帮助思考
- 小黄鸭调试法(一只龙虾也行)
- 选择一个好的文本编辑器，如 vim, Emacs, Github Atom, etc.
  - 更快的编辑程度，语法高亮，自动化
  - 演示
  - 以尽量提高自己的工作效率为原则，别卷入 Editor war
- Eclipse, MS Visual Studio 也行, Arduino IDE 并不好用





# 作业

- 尝试、阅读、思考 BlinkWithoutDelay 例子
  - File -> Examples -> 02.Digital -> BlinkWithoutDelay
  - 熟悉第 4 个和弦: millis() 函数
- 用 millis() 函数改进 Binary Time Counter
  - delay() 会阻塞

推荐相关阅读 [Multi-tasking the Arduino - Part 1 - Adafruit](#)

# 课后阅读

- Arduino 电子书架
  - <http://volthauslab.com/ebooks-2/>
  - 注意那个《20 个不可思议的 Arduino 项目》PDF
- 《用 Arduino 和 LED 入门 C 语言》教程编写中
  - 预计国庆节期间上线
  - <https://github.com/atommann/IDD001-fall-2017>

## 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>