《机器人编程基础》
课程报告

实验题目：**Arduino IDE Core 代码分析**

姓名、学号：成子谦，201730681303

学院、班级：软件学院 2017 级 1 班

华南理工大学软件学院

二〇一九年十二月

目录

1. 前言

1.1. Arduino 介绍

Arduino 是一家制作开源硬件和开源软件的公司，该公司负责设计和制造单板微控制器和微控制器包，用于构建数字设备和交互式对象，以便在物理和数字世界中感知和控制对象。Arduino 电路板设计使用各种微处理器和控制器。这些电路板配有一组数字和模拟 I/O 引脚，可以连接各种扩展板、面包板和其他电路。这些电路板具有串行通信接口，包括某些型号上的通用串行总线，也用于从计算机加载程序。微控制器通常使用 C/C++编程语言。除了使用传统的编译工具链之外，Arduino 项目还提供了一个基于 Processing 语言项目的集成开发环境。

1.2. 写作目的

本文主要目的为介绍并分析 Arduino IDE 源码库的 core 部分(以下简称“core 目录”)。

在 Manjaro 系统中，通过执行 yay -S arduino 安装 Arduino IDE 后，源代码存放在~/.arduino15/packages/arduino/hardware/avr/1.8.1 下。我们可以通过执行 sudo find / -name arduino | grep arduino 来找到 arduino 在根目录下的位置。其他 Linux 发行版操作类似。

1.3. 系统环境

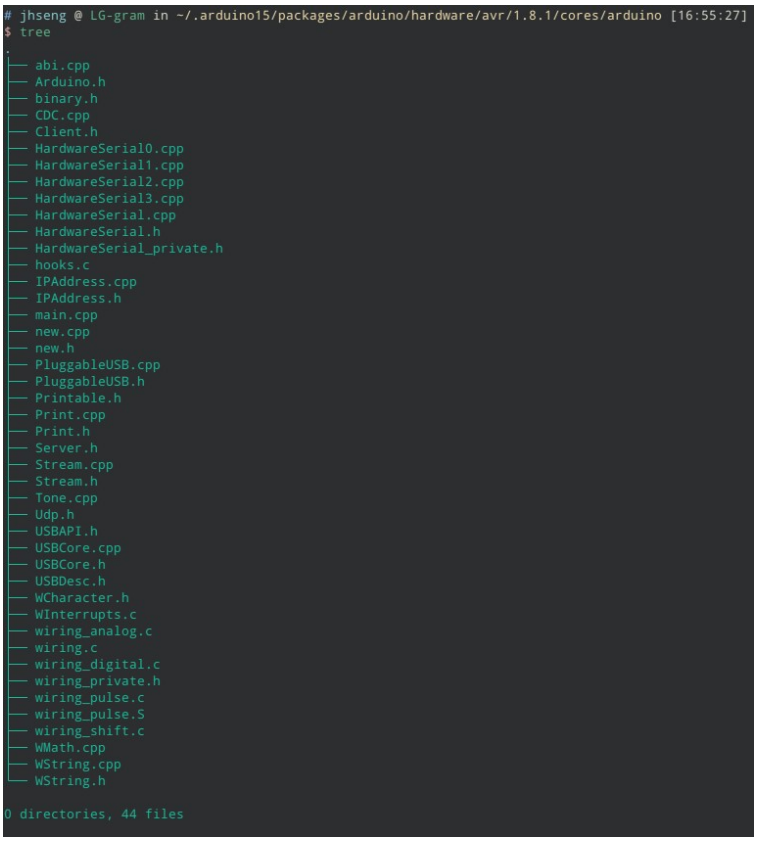
- 操作系统: Manjaro 18.1.4 Juhraya
- 内核版本: x86_64 Linux 4.19.88-1-MANJARO
- Arduino IDE 版本: 1.8.9
- OpenJDK 版本: 11.0.5, 2019-10-15

其中，操作系统和内核版本无要求。Arduino IDE 版本和 JDK 版本不能太老。

2. 代码分析

2.1. Core 部分概览

Arduino IDE 核心部分代码目录结构如图所示：

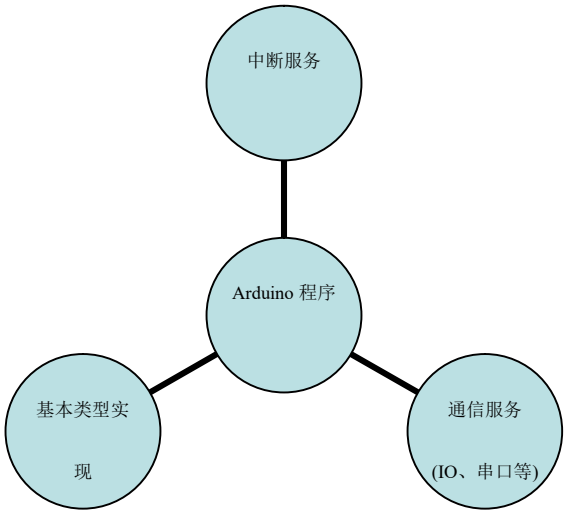


虽然代码都放置在同一文件夹下，显得比较杂乱，但仍可以按照其相关硬件与职责功能，分为以下七类：

类型	描述	相关代码
基本类	作为程序框架以调用其他代码	Arduino.h, main.cpp
基础功能类	提供或重载了基本功能	WCharacter.h, WInterrupts.c, WMath.cpp, WString.h, WString.cpp, binary.h, hooks.c, new.h, new.cpp, abi.cpp
IO 类	声明并定义了基本 IO 功能	Stream.h, Stream.cpp, Print.h, Print.cpp, Printable.h, Server.h
串口服务类	声明并定义了串口功能	HardwareSerial.h,

		HardwareSerial.cpp, HardwareSerial_private.h, HardwareSerial0.cpp, HardwareSerial1.cpp, HardwareSerial2.cpp, HardwareSerial3.cpp
网络类	封装了基于 UDP 协议的网络通信功能	IPAddress.h, IPAddress.cpp, Udp.h, Client.h
USB 串口类	声明并定义了 USB 串口基本功能	USBAPI.h, USBCore.h, USBCore.cpp, USBDesc.h, PluggableUSB.h, PluggableUSB.cpp, CDC.cpp
传感器控制类	提供与传感器进行交互功能	wiring.c, wiring_analog.c, wiring_digital.c, wiring.private.h, wiring_pulse.S, wiring_pulse.c wiring_shift.c, Tone.cpp

从文件名来看，core 部分代码主要实现的是基本类型的定义、IO 操作、串口服务、中断服务、有线操控等功能，其依赖关系呈星形结构。



下面我们从基本类代码出发，对 core 部分代码一探究竟。

2.2. 基本类

2.2.1. Main.cpp

当我们在编写 Arduino 程序时，我们只能看见 void setup()和 void loop()两个函数，main 函数去哪里了？实际上我们可以在 core 部分代码找到它，非常简短：

```
#include <Arduino.h>

// 直接返回
int atexit(void ( * /*func*/ )()) { return 0;}
// 在 Arduino.h 中声明为弱允许用户重新定义。

void initVariant() __attribute__((weak));
void initVariant() { }
// USB 设置
void setupUSB() __attribute__((weak));
void setupUSB() { }

int main(void) {
    init();
    initVariant();
#if defined(USBCON)
    USBDevice.attach();
#endif
    // 运行用户所定义的初始化行为
    setup();
    for (;;) {
        // 运行用户所定义的循环行为
        loop();
        // 执行串口通信
        if (serialEventRun) serialEventRun();
    }
```

```
    return 0;
}
```

main.cpp 只包含了 Arduino.h 这一头文件。抛去为了工程设计合理化而写的内容，main.cpp 实际上只干了四件事情：发现并连接 USB 设备、对 arduino 开发板进行初始化、运行用户定义的循环行为与执行串口通信。

2.2.2. Arduino.h

Arduino.h 是 Arduino 程序最基本的头文件。该头文件定义了 0~255 的二进制书写形式；定义了大部分常用常量（包括高低电平表示、数学类常量等）；针对芯片种类进行专门常量的定义；定义了大量常用函数的宏，包括数学、CPU 时钟、数字取高低位、位读写等函数。除此之外，Arduino.h 还声明了大量函数，如初始化操作、引脚操作、延迟与电平测量、移位读入输出、中断服务、用户自定义初始化与循环操作和扬声器操作等。此外，Arduino.h 还声明了部分 WMath.cpp 中函数的原型。

2.3. 基础功能类

2.3.1. 二进制数表示

Binary.h 为 Arduino.h 所包含，其定义了 0~255 的二进制书写形式。

2.3.2. 字符与字符串

WCharacter.h, WString.h 和 WString.cpp 三个文件声明并定义了常用字符操作、String 类及字符串的创建、修改、查找等常用操作。

2.3.3. 数学

WMath.cpp 声明并定义了随机种子生成、生成随机数、哈希和 word 类型生成函数。

2.3.4. 中断

WInterrupts.c 声明并定义了系统中断操作。它可以匹配中断模式、选择常数并将模式移动到适当位置以启用中断。这使得 Arduino 能实现比较基础的 IO 功能。

2.3.5. 内存管理

new.h 与 new.cpp 重载了 new 与 delete 函数。主要不同点是用 malloc 来实现 new 的过程。

2.3.6. 钩子函数

hooks.c 并没有完成，仅仅只是一个空函数。作者希望用此函数来构建支持协作线程的库或草图。这代表了 Arduino 程序未来的一个发展方向。

2.3.7. 虚拟 CXA 构造与删除

这部分代码存放在 abi.cpp 下，也是空函数。有可能在若干个版本之后继续完善。

2.4. IO 类

2.4.1. 流操作

Stream.h 和 Stream.cpp 声明并定义了基本流操作，包括流的直接读写、整数或其他类型读入、读取特定内容、寻找多个内容相同的流等用法。

2.4.2. 打印服务

Print.h, Print.cpp,Printable.h 与 Server.h 为类类型提供了打印服务，通过从 Printable 派生并实现 printTo 方法，用户可以通过把类的实例传递到 print 方法中进行打印。

2.5. 串口类

2.5.1. 基本功能

HardwareSerial.h, HardwareSerial_private.h 和 HardwareSerial.cpp 声明并定义了基于流实现的串口通信、IO 与中断处理功能。部分功能针对实际芯片不同而有所定制。

2.5.2. 串口实例

HardwareSerial0.cpp, HardwareSerial1.cpp, HardwareSerial2.cpp 与 HardwareSerial3.cpp 分别定义了四个实际串口。当使用到串口的任何元素时，连接器都将拉入整个串口文件。垃圾回收会使得未使用的符号被删除，但串口实例会继续保留。这样做的好处是可以防止连接器首先拉入任何未使用的实例。

2.6. 网络类

2.6.1. IP 协议

IPAddress.h 和 IPAddress.cpp 基于 printable 服务实现了 IP 协议。通过验证 IP 地址，实现了基本的数据传输功能。

2.6.2. UDP 协议

Udp.h 基于流实现了 UDP 协议，包含数据包构造、发送、接收与返回数据包相关信息等服务。

2.6.3. 客户端

Client.h 基于流实现了客户端，使得终端能与主机进行通信。包含可用性检查、数据通信、数据检索、缓冲区刷新与暂停服务功能。

2.7. USB 串口类

2.7.1. 核心功能

USBAPI.h, USBCore.h, USBCore.cpp 与 USBDesc.h 声明并定义了一系列 USB 串口基本服务。包括数据的接收与发送、缓冲区刷新、设置 EP 位、TX 与 RX 的释放、端点服务可用性检查与中断服务。

2.7.2. 可插拔 USB

PluggableUSB.h 与 PluggableUSB.cpp 建立了可插拔 USB 的模型，声明并定义了模型构造、接口获取、描述获取与名称获取等服务。该服务可以使用单例模式实现，具有较高的灵活性。

2.7.3. CDC 通讯

CDC.cpp 声明并定义了一系列 CDC 通讯接口服务与实现机制。支持 CDC 接口获取、CDC 连接建立、可用性检查、数据读写、缓冲区刷新与忙等待功能。

2.8. 传感器控制类

2.8.1. 基础服务

Wiring.c 与 wiring_private.h 通过设置预分频器，统计每个计时器的总毫秒数 0 溢出来设定指定时钟频率，来实现对传感器的时钟信号产生与发送。同时对特定芯片定制服务，使得时钟服务更加精确。

2.8.2. 数据读写

Wiring_analog.c, wiring_digital.c 与 wiring_shift.c 声明并定义了数字信号 IO、pwm 信号 IO 与移位 IO 的服务。满足了对不同传感器信号类型与大小的要求。

2.8.3. 脉冲测量

Wiring_pulse.c 与 wiring_pulse.S 声明并定义了引脚脉冲测量方法。

2.8.4. 计时测量

Tone.cpp 声明并定义了计时器功能，针对 8 位、16 位和 32 位计时器分别实

现了不同的计时服务。除此之外，计时器也引入了中断向量来执行中断程序处理功能。

3. 附录

3.1. Core 部分所有代码(附中文注释)

由于代码量极大，以附件形式给出。

提交文件说明

|- report.doc

|- report.pdf

|- image

    |- tree.png

|- src

    |- abi.cpp

    |- Arduino.h

    |- binary.h

    |- CDC.cpp

    |- Client.h

    |- HardwareSerial.cpp

    |- HardwareSerial.h

    |- HardwareSerial_private.h

    |- HardwareSerial0.cpp

    |- HardwareSerial1.cpp

    |- HardwareSerial2.cpp

    |- HardwareSerial3.cpp

    |- hooks.c

|- IPAddress.h

|- IPAddress.cpp

|- main.cpp

|- new.cpp

|- new.h

|- PluggableUSB.cpp

|- PluggableUSB.h

|- Print.cpp

|- Print.h

|- Server.h

|- Stream.cpp

|- Stream.h

|- Tone.cpp

|- Udp.h

|- USBAPI.h

|- USBCore.cpp

|- USBCore.h

|- USBDesc.h

|- WCharacter.h

|- WInterrupts.c

|- wiring.c

|- wiring_analog.c

|- wiring_digital.c

|- wiring_private.h

|- wiring_pulse.c

|- wiring_pulse.S

|- wiring_shift.c

|- WMath.cpp

|- WString.cpp

|- WString.h