



名称：Clock Shield 用户手册

版本：v0.9b

日期：2013 年 4 月 15 日

版权所有，翻版必究

目录

1 宝贝描述.....	3
2 详细参数.....	3
3 接口描述.....	3
4 使用方法.....	4
4.1 实验准备：硬件、程序库安装.....	5
4.2 实验过程：动手往往学得更多，创意就这样产生.....	5
4.2.1 【例程 1】：流水灯.....	5
4.2.2 【例程 2】：按键控制 LED.....	7
4.2.3 【例程 3】：按键控制蜂鸣器音调高低.....	8
4.2.4 【例程 4】：环境光越强，LED 越暗.....	8
4.2.5 【例程 5】：热敏测温，数码管显示温度.....	8
4.2.6 【例程 6】：数码管上的字符在流动.....	9
4.2.3 【例程 7】：综合实验，实时时钟显示控制.....	10
5 配件清单.....	12

1 宝贝描述

Clock Shield (即数码管时钟模块) 板载 RTC 时钟、数码管驱动、可做流水灯 (指示灯) 的 LED、温度传感器、光敏电阻, 硬件资源丰富, 是想踏入开源硬件殿堂大门的初学者理想的扩展板, 结合 Arduino 兼容主板是学生们的理想实验平台。其特点如下:

1. 直接插在 Arduino 兼容的主板即可使用;
2. 板载实时时钟芯片 DS1307, 只占用两根信号线;
3. Arduino 配套实验例程丰富。

若使用其他单片机系统 (51/AVR/PIC/MSP430 等), 则其需具备以下硬件资源:

1. 10 个数字引脚 (4 个控制 LED, 3 个控制按键, 2 个控制数码管驱动芯片, 1 个控制蜂鸣器)
2. 2 个模拟输入引脚 (用于读取热敏和光敏的信号电压)
3. 1 个 I2C 接口 (与实时时钟芯片 DS1307 通信)

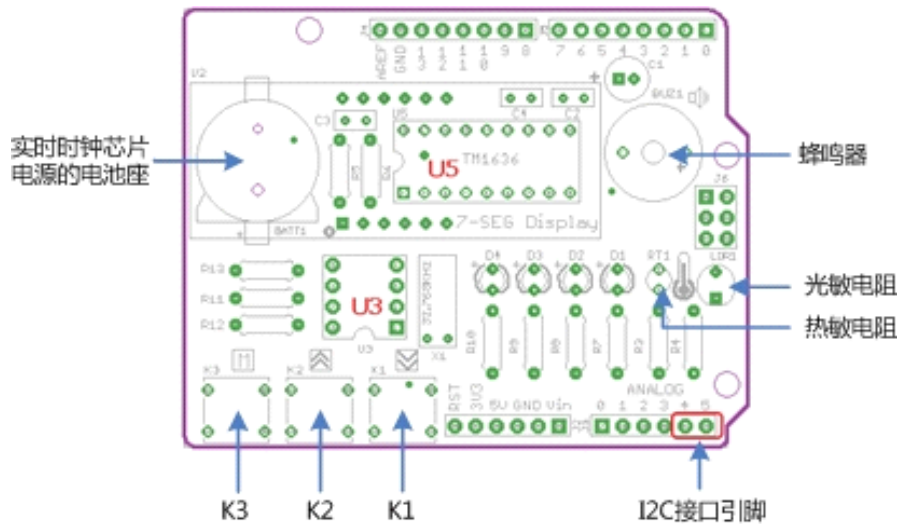
2 详细参数

项目	最小值	典型值	最大值	单位
电压	4.5	5	5.5	V
电流	—	80	—	mA
尺寸	74x59x30			mm
重量	35			g

主控板提供给 Clock Shield 的电源电压建议是 5V, 信号电压 (信号的高电平) 5V、3.3V 均可工作。

3 接口描述

Clock Shield 的接口如下图所示:



硬件资源：

电池座：卡上的电池给实时时钟芯片供电，使得板子掉电都可继续计时；

蜂鸣器：5V 的直流蜂鸣器，可做闹钟、按键提示音；

光敏电阻：感应环境光的强度；

热敏电阻：检测环境的温度；

K1~K3：可直接被单片机扫描的轻触按键；

U5：TM1636，即 4 位 8 段共阳数码管驱动芯片；

U3：DS1307，即实时时钟芯片，计时范围到 2100 年；

D1~D4：蓝，绿，红，红的 3mm 的插件 LED。

数码管：4 位 8 段共阳带时钟点的数码管，而小数点是不显示的，只作方向识别用途。

占用 Arduino 的引脚资源：共占用 9 个数字引脚、2 个模拟输入引脚、1 个 I2C 接口

D2: 控制蓝色的 LED1；

D3: 控制绿色的 LED2；

D4: 控制红色的 LED3；

D6: 控制蜂鸣器；

D5: 控制红色的 LED4；

D7: 连接 TM1636 的时钟引脚 SCLK；

D8: 连接 TM1636 的数据引脚 DIO；

D9: 控制按键 K1；

D10: 控制按键 K2；

D11: 控制按键 K3；

A0: poll readings from temperature sensor;

A1: poll readings from light sensor;

A4: 连接 DS1307 的 I2C 数据引脚 SDA ;

A5: 连接 DS1307 的 I2C 时钟引脚 SCL。

4 使用方法

Clock Shield 的程序库、用户手册可从开智电子的百度网盘

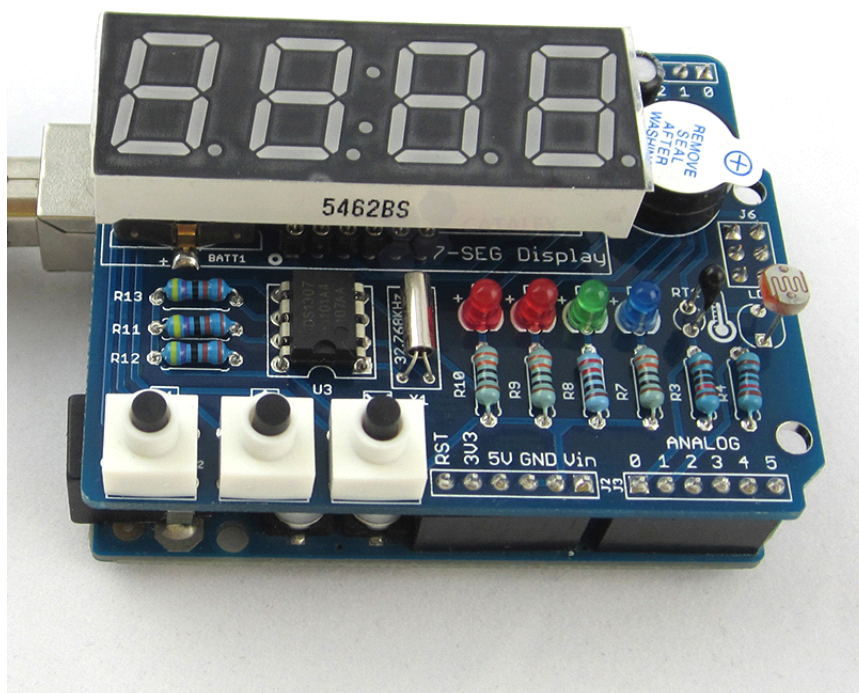
<http://pan.baidu.com/share/link?shareid=426437&uk=1966479381> **下载，包括的 6 个单一例**

程和一个综合例程概览如下：

1. 4 个 LED 流水灯功能，流动速度可控；
2. 3 个按键控制 4 个 LED 显示；
3. 2 个按键控制蜂鸣器发出不同音调的声音；
4. 环境的光强控制 LED 的亮度；
5. 数码管显示热敏电阻测到的温度；
6. 数码管显示流动的字符；
7. 可调节带闹钟的实时时钟。

4.1 实验准备：硬件、程序库安装

1、将 Clock Shield 插到 Arduino UNO 或者其他兼容主控板上，并用 USB 线将主控板连接到电脑。若是第一次使用 Arduino 主控板，还需要查找到 Arduino IDE 目录下的 drivers 目录进行驱动更新即可。本次演示使用 Arduino UNO R2 主控板（若没有可用其他版本或者兼容版本）。



2、本次演示用的 Arduino IDE 版本是 Arduino-1.0，也可用新的版本 1.0.1 等等。将百度网盘下载的 Clock Shield Libraries for Arduino 1.0+.rar 压缩包直接解压到自己的 Arduino IDE 目录下如..\arduino-1.0\libraries。MsTimer2、TimerOne_v9 这两个关于使用单片机的 timer1、timer2 都是从 Arduino 官网下载的，而 TM1636 是数码管驱动 IC 的底层库，ClockShield 则是针对 Clock Shield 的应用库，例程都是在该库下。**注意：每次 libraries 目录下新增了库都需要将 IDE 重启才生效，即关闭所有由该 IDE 打开的程序文件，再打开 IDE。**

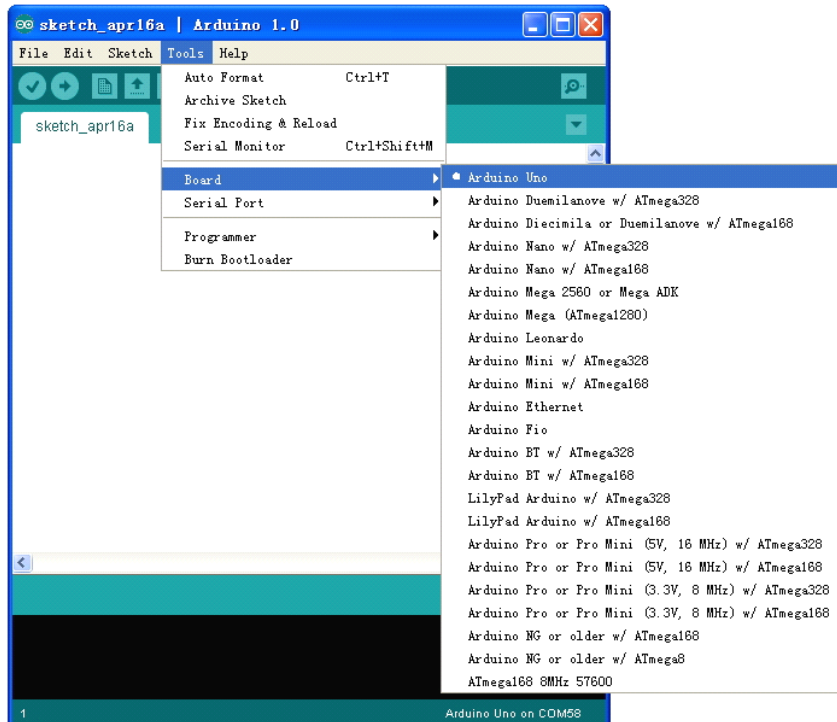
4.2 实验过程：动手往往学得更多，创意就这样产生

为减少 Arduino 初学者的心里障碍，接下来的例程介绍由易到难，期望能通过简单的实验学到一些知识，迸发一些解决生活问题的创意。

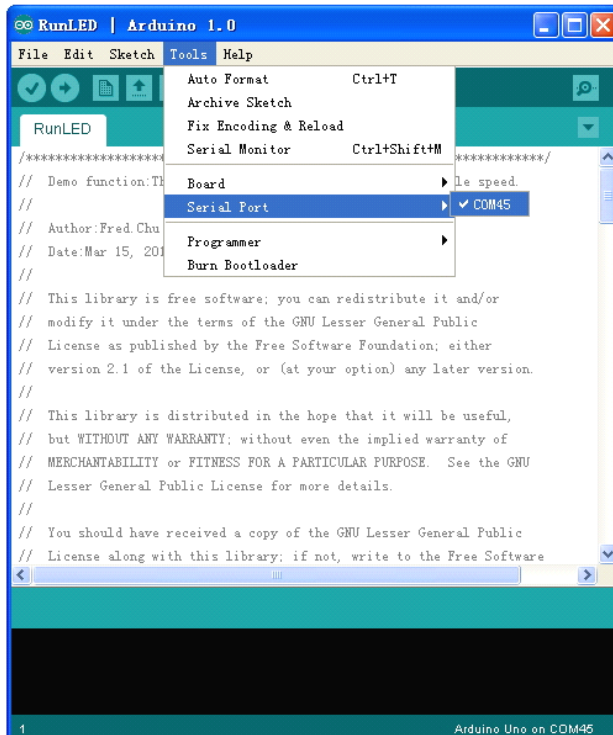
4.2.1 【例程 1】：流水灯

流水灯对电子爱好者来说就如程序编程的 Hello world 那样简单而又必不可少的一个入门练习，流水灯往往就延伸出其他各种 LED 变换花式。

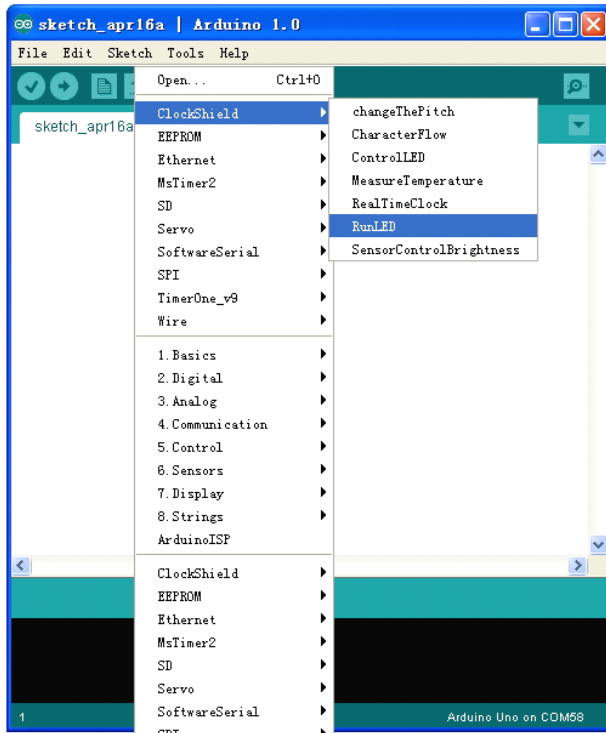
1).先要选择主控板 Arduino UNO。



2).选择 PC 和主控板连接的串口号



3).从 IDE 中打开流水灯例程“ RunLED”



4). 点击烧录按钮即可将打开的例程烧录到主控板上，即可看到 4 个 LED 从右到左流动，流动速度在程序中可调



4.2.2 【例程 2】：按键控制 LED

该例程的主角依然是 LED，但不再是一根筋地流动，而是利用按键添加了一些互动。

1).从 IDE 中打开例程“ ControlLED” , 方法同例程 1。

2).烧录例程到主控板,方法同例程 1。首先会看到 4 个 LED 都熄灭的;现在按 Menu 按键 (K3) 即可进入开始状态,此时 4 个 LED 会同时闪烁,如果有些意外还没看到此现象请再按 K3;然后按 K2 (增加亮灯数量) 或者 K1(减少亮灯数量);然后然后再按 K3 则会回到一片昏暗的状态——LED 立刻熄灭。

4.2.3 【例程 3】: 按键控制蜂鸣器音调高低

1).从 IDE 中打开例程“ changeThePitch” , 方法同例程 1。

2).烧录例程到主控板,方法同例程 1。K2 变成提高音调按键, K1 变成降低音调按键,仅是音调的高低变化,而没有准确到具体的调。

4.2.4 【例程 4】: 环境光越强, LED 越暗

实验步骤:

1).从 IDE 中打开例程“ SensorControlBrightness” , 方法同例程 1。

2).烧录例程到主控板,方法同例程 1。光敏电阻 LDR1 附近的光强越强,LED2(程序默认)就越暗,光强约弱, LED2 就越亮。**注意: 其中 LED2 可改成 LED4, 只有 LED2 和 LED4 才连接到 Arduino 的 PWM 口, 亮度才可调节。**

可以想到光敏控制 LED 这种装置在家里也有用处, 在开灯开关附近装一个, 房间灯还没开时, LED 可作指示灯亮着 (亮度可以调节到适用程度), 开了灯的时候, LED 又熄灭了, 降低功耗。

4.2.5 【例程 5】: 热敏测温, 数码管显示温度

温度和板载热敏电阻的阻值之间的关系公式如下

$$T = \frac{1}{\ln(R_{NTC} / R_{ZERO}) / B + \frac{1}{298.15}} - 273.15$$

其中 T 就是测得的摄氏温度, RNTC 为根据 Arduino 读到 NTC 热敏电阻的一端电压数据及分压原理计算得出的热敏电阻的阻值; RZERO 为热敏电阻的初始阻值 (25℃温度下), 值为 10kΩ; B 就是由其材料决定的值, 为 3975。这公式和值在程序里会用到。

实验步骤：

- 1).从 IDE 中打开例程” MeasureTemperature” ，方法同例程 1。
- 2).烧录例程到主控板，方法同例程 1，即可看到数码管会显示热敏电阻测到的温度，如下

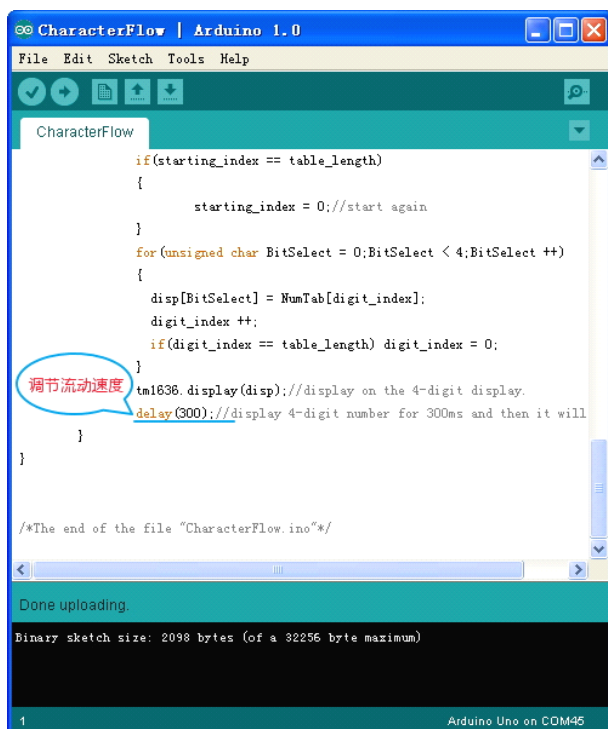
图所示。



4.2.6 【例程 6】：数码管上的字符在流动

实验步骤：

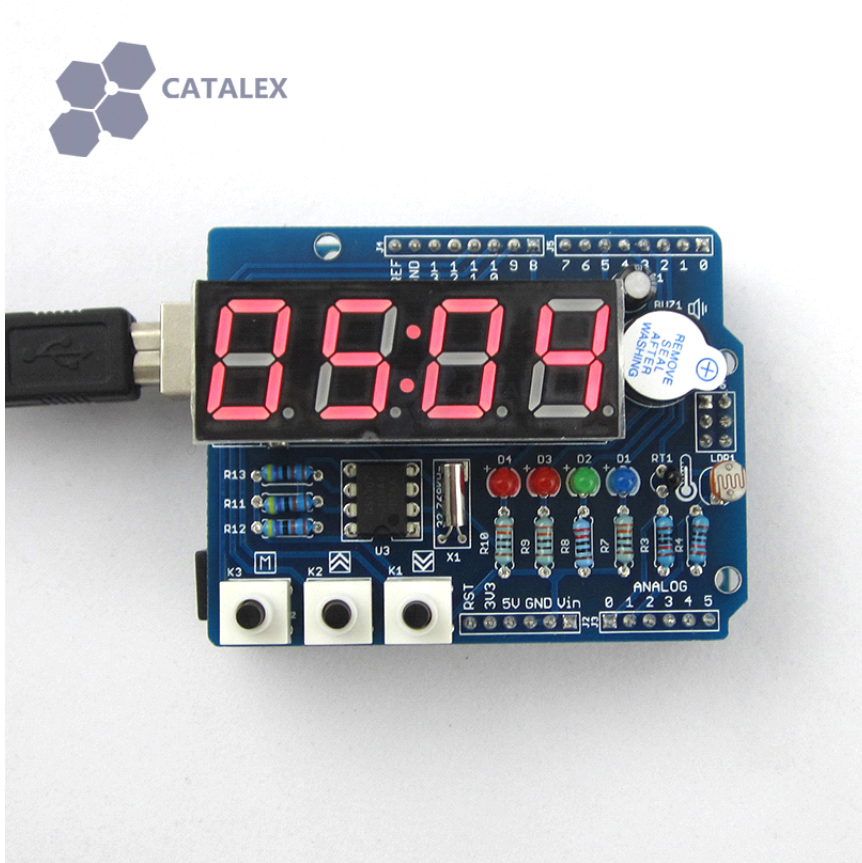
- 1).从 IDE 中打开例程” MeasureTemperature” ，方法同例程 1。
- 2).烧录例程到主控板，方法同例程 1，即可看到 16 个字符 0~9，A、b、C、d、E、F 在 4 位数码管上从右到左依次流动，流动速度由延时长短决定，延时越长，流动越慢，程序一角如下图所示。



4.2.3 【例程 7】：综合实验，实时时钟显示控制

实验步骤：

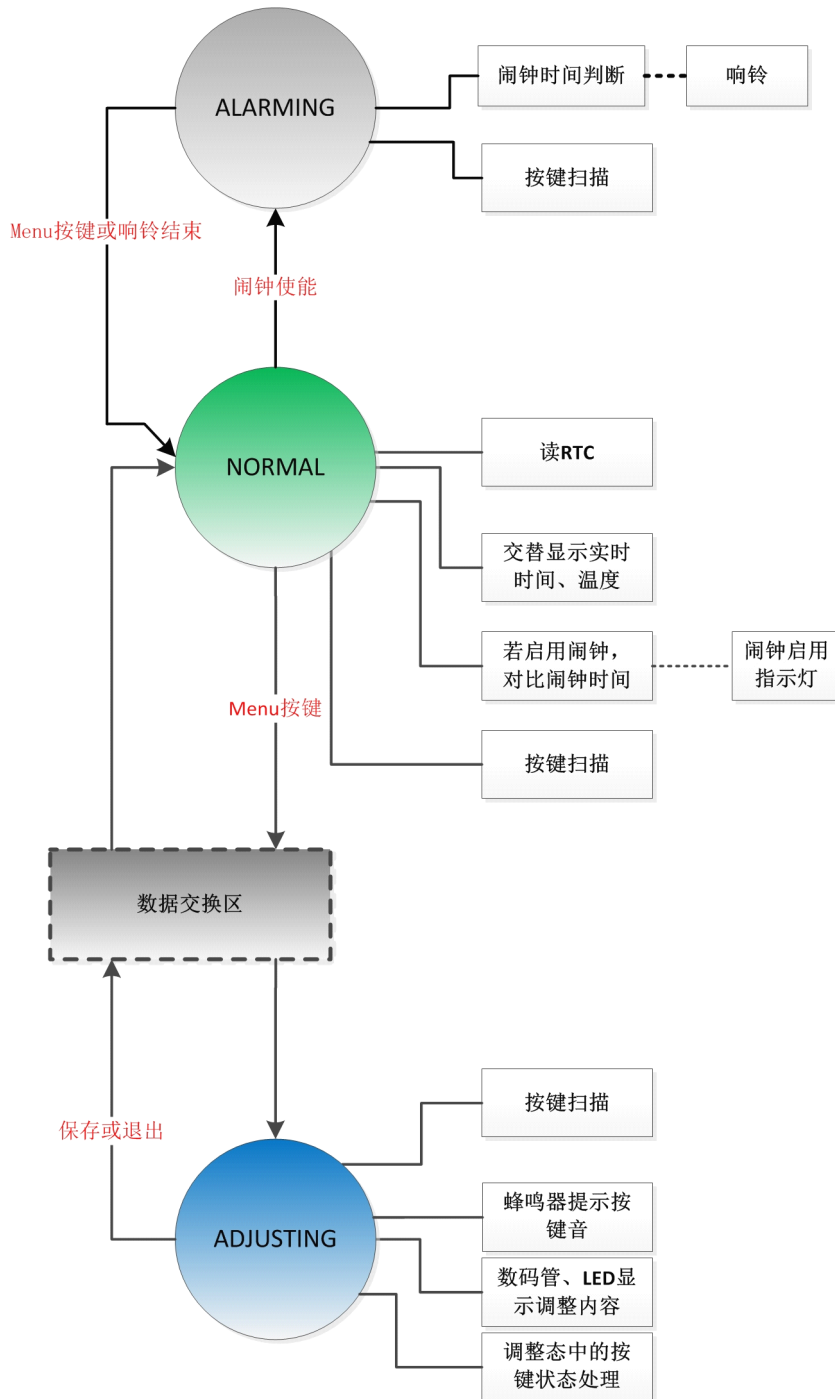
- 1).从 IDE 中打开例程“ RealTimeClock” ,方法同例程 1。
- 2).烧录例程到主控板，方法同例程 1，即可看到数码管会显示 10s 时间，然后再显示温度，这样交替显示，如下图所示。按键 K3 是“菜单”按键，K2 是“增”按键，K1 是“减”按键。K3 按键可依此进入时钟时针、分针调节，闹钟时针、分针调节，闹钟是否启用调节，显示亮度调节等状态。蜂鸣器是闹钟和按键的发声源。



关于例程程序：

1.整个系统状态的变换如下图所示：NORMAL – ADJUSTING – ALARMING

- 常态下是 NORMAL，在闹钟使能且闹钟时间到时进入 ALARMING 状态，然后会响一分钟闹铃或者按了 Menu 按键才会结束 ALARMING 状态，并恢复到 NORMAL。
- 在 NORMAL 状态下按下 Menu 按键即会复制一份现状态下的一些变量值给 ADJUSTING 中用到的变量，然后进入 ADJUSTING 状态。长时间无输入即会不保存设置然后退出该状态恢复到 NORMAL；每次按 Menu 按键都会转换其内部的状态，如闹钟小时调整、闹钟分针调整...保存退出，成功保存退出后则将 ADJUSTING 中变量值覆盖 NORMAL 状态中的响应变量。



2. 一个类函数实现中调用另一个类的函数

由于 Clock Shield 上面带有数码管驱动电路，而之前已经创建一个驱动数码管的类，以驱动芯片命名 TM1636，而在 Tick Shield 文件中就定义了一个 tm1636 对象，可供 TickShield 函数方便调用数码管显示的函数。

3. 引脚资源使用均在 ClockShield.h 中声明

一般情况下，Shield 中的引脚资源都已经确定，无法更改，都会在相应的头文件声明该

Shield 已占用的引脚，方便别人扩展，但还有个隐藏的引脚资源可能会被忽略，就是 wire.h 文件就是说明占用 I2C 接口的，SPI.h 就是说明占用 SPI 接口的，TM1636 tm1636(7,8)定义对象时也说明占用 D7/D8。

由于使用了 I2C 接口，所以该 Shield 不能兼容 Arduino MEGA2560(简称 MEGA)，Arduino UNO R2 的 I2C 在 A4/A5，而 MEGA 的在 D20/D21。

4. 闹钟时间保存在主控板内置 EEPROM 中

实时时间由 RTC 芯片保存，但其并没有闹钟时间，所以需要将闹钟时间保存于内置的 EEPROM 中，在开机时都会读取该闹钟，每次设置闹钟并保存后也会更新到 EEPROM 中。

5 配件清单

1 个 Arduino 扩展板 Clock Shield v0.9b。

1 份 Clock Shield 的原理图 pdf 文件和用户文档电子版（在线传输、邮件传输或百度网盘）。

注：开智电子百度网盘网址 <http://pan.baidu.com/share/link?shareid=426437&uk=1966479381>。