

2019.07.01上午

电子设计小学期工作日的第一个上午，首先我们较为顺利地通过了预习验收，鼓舞了项目开始时的士气。

此外，在等待验收的前前后后的过程中，我们主要使用W3Cschool的arduino教程，对我们所选的主控模块进行了简单的上手热身。主要了解了其整体的程序语法，控制流，IO功能和串口通信调试功能。

接近上午调试结束时，我们还盘点了已有的一些模块。我们现有的模块有LCD显示屏，蓝牙通信模块，基本可以实现数字部分的功能。而模拟部分的模块，大部分传感器仍在配送，电源管理模块可先根据已有的备选芯片进行一定的调试。故而我们敲定了之后的计划，按照电源管理、arduino并行的方法进行调试。而LCD的调试相蓝牙模块调试而言比较简单，故先进行调试；并且另一路对电源管理模块调试的结束后，可以分人手去提前学习一下蓝牙模块的使用。得到近期调试优先度大纲如下：

1. arduino,LCD,串口联调；电源管理模块参数测试
2. 蓝牙模块学习调试。
3. 传感器模块的参数调试与联调。
4. 其他基于分立元件（如光敏电阻）的外围传感电路设计
5. 写数字系统整体代码框架

2019.07.01下午

在经过上午的验收以及上手热身后，我们在下午正式开启了设计与调试，由于大多数传感器还没有送达，我们手中已有的模块是arduino uno主控模块和LCD1602液晶显示模块，为了减少IO的使用，我们特地前往中发电子大厦购买了I2C转接板，将16引脚方便的减少为4引脚和arduino相连接。

在购置回转接板后，我们一方面开始学习LCD1602与arduino的硬件连接方法，以及其各个引脚的说明，并且在利用已有的LiquidCrystal库函数的情况下，尝试进行了字符数据的显示，成果如下：



图 1: LCD字符显示

起初并不能显示，后来我们很快发现是转接板电位器的问题，转接板电位器直接控制了LCD显示的亮度，因此在使用镊子改变电位到合适的亮度后便能观察到字符。在能够显示字符后，我们进一步结合上午的学习进行了串口LCD通信联调，使得在键盘上实时输入字符在LCD上进行显示，这是我们之后显示模块的重要基础，我们拍摄成果的照片如下：

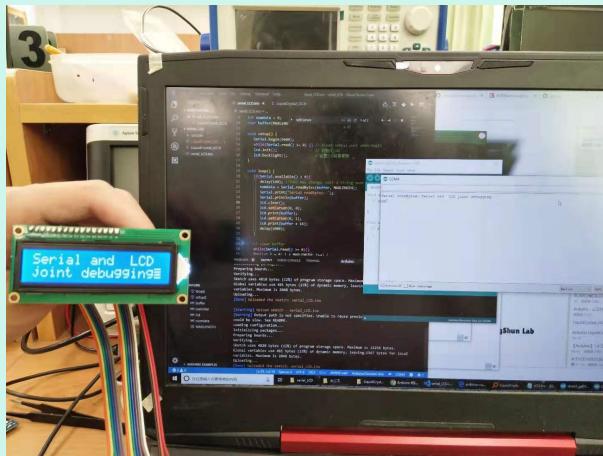


图 2: LCD串口通信联调

另一方面，我们组在调试LCD的同时，对电源管理电路进行了实际的检测，我们计划使用9V的干电池，而恰好在实验室中找到一块电源管理的模块，能够在小于12V输入的情况下，输出5V/3.3V的直流电压，我们类比于电网10%的波动，使用8V ~ 10V的50Hz正弦波作为输出，观察两输出的电压情况，结果十分令人满意，根据示波器的显示，以及自动测量的结果，能够得到纹波非常小的直流电压，并十分接近其标称的输出，记录如下图所示：

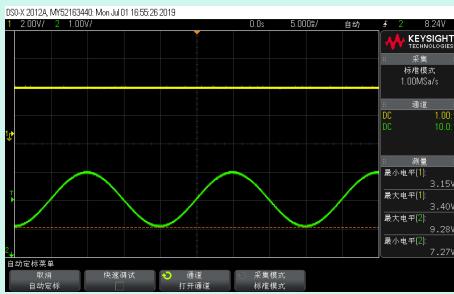


图 3: 3.3V 稳压输出

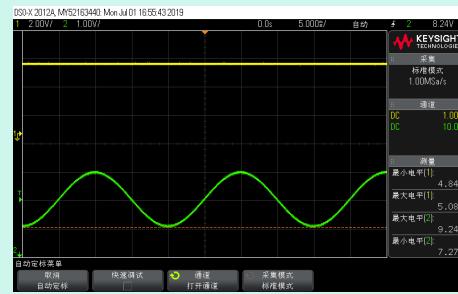


图 4: 5V 稳压输出

其中黄色线为稳压输出，绿色线为输入，验证了该电源管理模块能够提供理想的供电电压，在TI公司的样片到来之前为我们的电源管理提供了替代。

在下午收工以后我们另外找到一片蓝牙模块，计划于明天开启蓝牙模块的调试以及DHT11的湿度模块的调试，并通过LiquidCrystal库编写代码，实现自己需要的函数的头文件。

2019.07.02上午 DHT11温湿模块与蓝牙模块调试

电子设计小学期工作的第二个上午，我们明显加快了调试的进度，在昨天初步调试LCD1602后，今天我们进一步同时开始调试蓝牙模块和DHT11温湿模块。

首先，上午蓝牙模块的调试并不是十分顺利，中间遇到了一些连接的问题，进度稍慢，而温度湿度传感器模块借助于Arduino官网可查找的dht11的库，可以很快的进行实现。我们设置没两秒更新并发送一次数据，可以在串口接收到数据。之后便很方便地将数据

为了验证其正确性，我们将传感器分别在室内与室外进行了测量，结果表明室外比室内温度大约高3°C左右，湿度也稍高于室内，符合实际情况，测量结果如下图5记录：

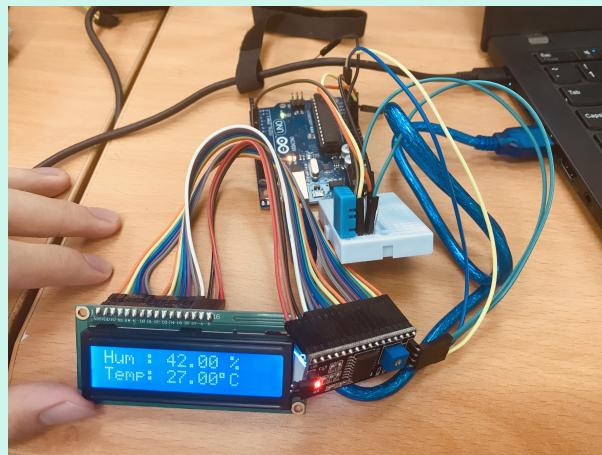


图 5: 温度湿度测量与LCD显示

该LCD显示了当前教室的室温和室内湿度。整体上温度湿度模块由于能够直接输出

校准后的数字信号，所以调试过程比较顺利，只是在单位的输出时“C”的符号难以直接输出，库中的print()函数无法输出该字符，需要去自定义字符，否则会输出响应的日文，因此我最终采用了print((char)233)在其中一格的 5×7 矩阵先输出“度”的上标，再输出C来完成单位的显示。

此外，我们修改了LiquidCrystal_I2C库的内容，将其精简并增加了和串口联调的功能，成为我们可以使用的serial_lcd库，其中的函数实现原理大致与官方库相同。此后该模块恰好能够利用该屏幕进行显示，因此我们考虑了一下多模块测量的显示方法，我们选用TTP226电容触摸开关进行显示内容的选择，不同的开关控制不同内容的显示，因此我们目前手头只有TTP224，并用其进行测试其触摸效果，结果十分令人满意。

而另一模块，蓝牙模块的调试，能够初步和设备进行连接，但基于Arduino进行数据通讯仍具有一定的问题，有待下午进一步调试。

2019.07.02下午 蓝牙模块调试和系统架构细节讨论

在下午我们完成了蓝牙模块的调试，经过排查，发现是使用的蓝牙模块已经进行了配置，和出厂设置不同，厂方给出的设置方案不能直接使用。在简单了解AT指令集后，我们迂回使用USB转TTL的芯片，先对蓝牙模块进行恢复出厂设置，便可按照厂方文档进行调试。按照应用情景，我们将蓝牙模块设置为从机，波特率设置为9600（之后可能会调节到更低以节省功耗），简单测试了手机通过蓝牙模块、电脑通过蓝牙模块与arduino的通信，顺利完成数据的收发并能做出简单的响应，可以预料到能与后续的模块完成连接。

此外，气压传感器BMP180的特性测试和简单实用调试也在今天下午完成。此外在考虑微波雷达探测人体存在的实际情况，我们认为可能需要做一定的角度的扫描检测，故而在实验末尾简单阅读了舵机的工作原理和操作例程，这是一个控制模块，可以预料到调试环节不一定顺利。但是这是一个数字驱动的模块，可以在今晚先将大部分功能点实现，其他的预想功能如根据雷达信号的反旋转舵机，需要之后联调实现。

最后，已经看到的情况是，由于外设较多，arduino的IO资源已经有些紧张，而且各外设的使用方式不一，有直接用IO功能进行读写，有用I2C进行通信。我们敲定方案是用IO转I2C和I2C扩展板，实现一个统一的数据总线结构，方便整体系统调试。另外我们还需要一个数据存储媒介给仪器记录实验数据。所以在今天下午实验结束后，我们还去中关村中发市场进行了相关器件的采购。

2019.07.03上午 舵机、SD卡、LCD12864、红外收发调试

电子设计小学期工作日的第三个上午，我们继续对已有的模块进行测试。我们在昨天的采购时重新购置了一块较大的液晶显示屏，型号为LCD12864，是一块分辨率为 128×64 的显示屏，并带有中文字库，相比与1602具有更大的优势，首先我们先去将其20引脚焊好排针，之后我们利用了已有的一个简单库函数，并且对其进行函数的补充与丰富，例如清除光标等函数，并对其进行了显示的测试，测试结果能够清楚的显示如下图6，但是发现该屏幕对电磁干扰非常敏感，在硬件接线有扰动或者其他电磁干扰的情况下，会出现乱码的情况，这是我们进一步需要解决的问题。

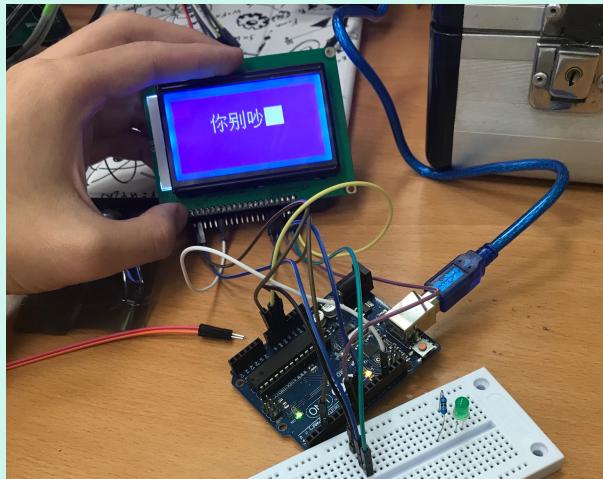


图 6: 12864显示

除此之外，我们调试使得舵机能够正常运转，我们希望借助 180° 旋转的舵机能够帮助微波雷达模块对室内的人体进行感应与检测，扩大其检测范围。在调通舵机后，我们开始了对SD卡存储的调试，初步能够实现通过Arduino将数据发送的SD卡进行存储。并且在舵机模块和SD卡模块的类库的使用中，我们注意到从类库生成的对象较大（一个舵机大约占用动态内存的20%、一个SD文件对象大约占用动态内存的50%），故而我们讨论了两种解决方案，第一是精简类库，减少内存消耗，二是使用 PROGMEM后缀，将变量存储在flashrom中。第二个解决方案用于对读写实时性的要求不高的场景，初步确认SD卡文件对象内存占用大的问题可以使用这个方式得以解决。这是因为将测量数据转存到SD卡的操作本身就是外存文件操作，速度较低。而舵机对象的问题推迟到最后联调时视情况解决。此外我们还进行了红外收发模块的调试和标定：得到编码如下：

```
//result
/*
  CH-    CH     CH+
FFA25D  FF629D  FFE21D
|<<    >>|   >|||
FF22DD  FF02FD  FFC23D
-      +      EQ
FFE01F  FFA857  FF906F
0      100+    200+
FF6897  FF9867  FFB04F
1      2      3
FF30CF  FF18E7  FF7A85
4      5      6
FF10EF  FF38C7  FF5AAS
7      8      9
FF42BD  FF4AB5  FF52AD
*/
*/
```

图 7: 红外遥控编码

并且我们进一步讨论总线的管理方法，由于IO资源较少，而待测量变化频率一般较低，我们希望能够在每次轮询周期内利用一根总线分别选通各传感器进行数据传输，因此我们需要搭建数据选择电路，为此我们在午饭时再次前往中发市场购置数据选择器等器件。

根据课上所学的知识，我们在中发市场增购了八输入数据选择器74151，用做输入总线的设计。此外采购了GPIO转I2C的芯片PCF8574。I2C在LCD1602调试时我们接触到的一个串行通信协议。在下午的正是调试中，我们两路并行进行，一方面对购得的元件进行逻辑测试，另一方面学习理解I2C协议。以下是在PCF8574的数据手册中截图得到的关于I2C的介绍。大约分为主机发送起止命令、选择地址和读写模式，以及从机被读或被写时按8位数据位一位应答位的串行数据传输形式。

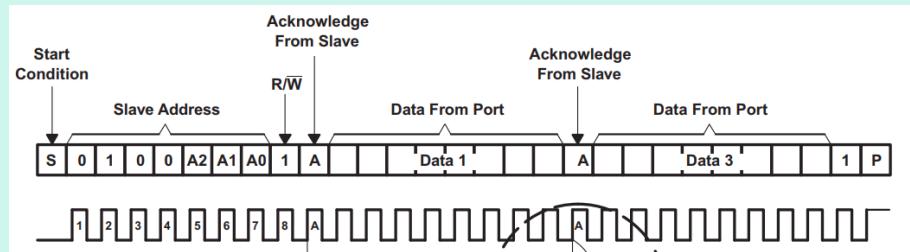


图 8: I2C数据传输举例（从数据手册中摘录）

在简单讨论后，我们决定先试用I2C进行总线管理。主要出于以下几个目的：

1. I2C适用于通信速率低的情景，在轮询传感器时，符合这个特点。
2. I2C是相对主流的通信协议，可以方便后续拓展。
3. I2C协议读写均支持，是更完整意义上的总线形式，比起我们用数据选择器设计的只读总线结构更为合理，也可以把读传感器值和内容显示等读写操作统一在一个时序中，让程序更为统一。
4. I2C协议要求我们对时序电路有更好的理解，我们也想在这方面挑战一下自己。