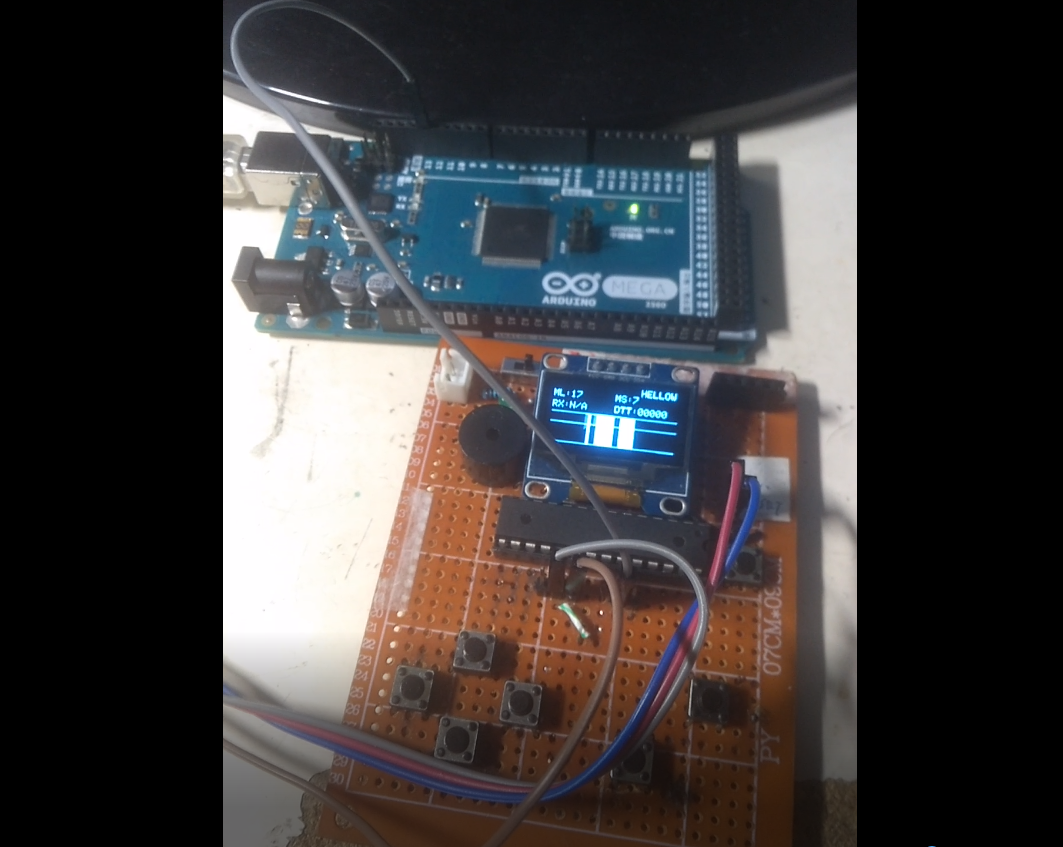
# Arduino 电报译码 以及 自动发报

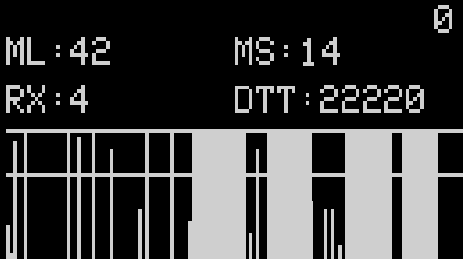
### LHW



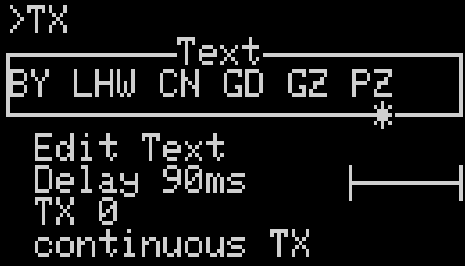
## 操作界面

### 0

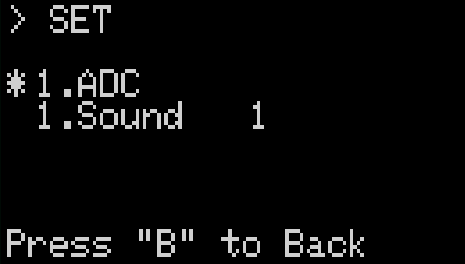
**主菜单**



**自动译码**



**自动发报**



**设置界面**



**设置采样深度**

## 自适应传输速率以及译码

**传输速率:**

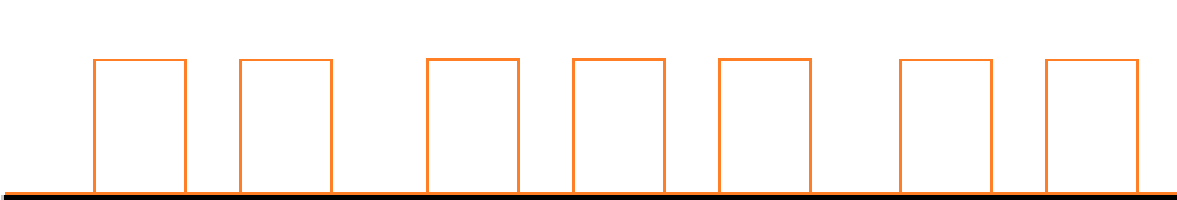
M /--/

M和O的电码

O /---/

↓下图为MO的电码片段

**M O**



LL HL LS HL LL HL LS HL LS HL LL

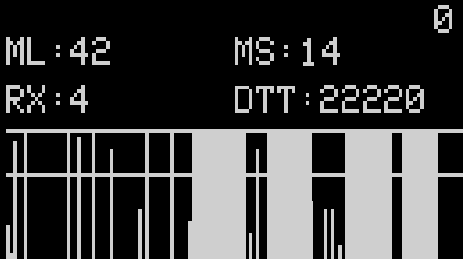
然后我们根据信号的高低电平和长短情况把信号进行分类:

|  |  |  |
| --- | --- | --- |
|  | 高电平 H | 低电平 L |
| 长 L | HL 2 | LL |
| 短 S | HS 1 | LS 0 |

由图可见字母M是由 LL HL LS HL 组成，我们要译码首先要知道电码点和划的间隔，这个通过一个算法可以做到实时计算。

我们把采样到的数据放到一个128长度的缓存中，并且用实时采样数据进行覆盖，在一段无信号的时间后检测到信号视为信号开始，当信号停止后开始译码。

译码会对比码库，找到相同就是了。



在译码界面可以通过上下键调节过滤阀值

低于过滤线的信号会被忽略，建议事先用外部电路处理信号

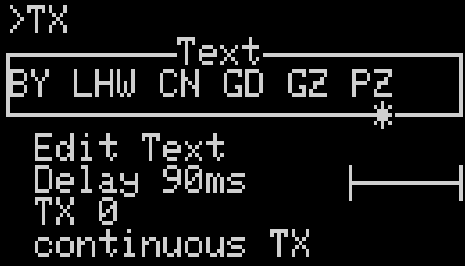
如果中途速率有较大改变可以按A键重置参数，重新计算发送间隔

## 自动发报

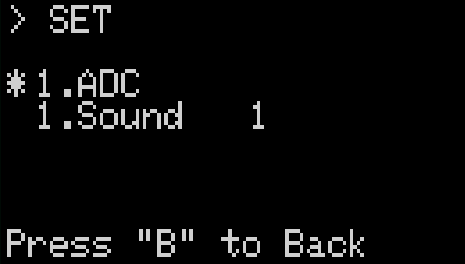
这个过程是译码的相反过程，首先有一个输入框长度为21，用户选择Edit Text后可以通过上下键选择合法字符，左右键移动光标，输入完成A键退出。

当A键按下”TX”或”Continuous TX”后进行发报，程序会在码库中查询输入框的第N个位置对应的电码，随后码库会给出1和2, 前者代表点，后者代表划。

完成查询会让对应引脚输出信号，就这样很简单。



## 设置



**第一个是设置采样深度**

**第二个设置是否开始蜂鸣器**



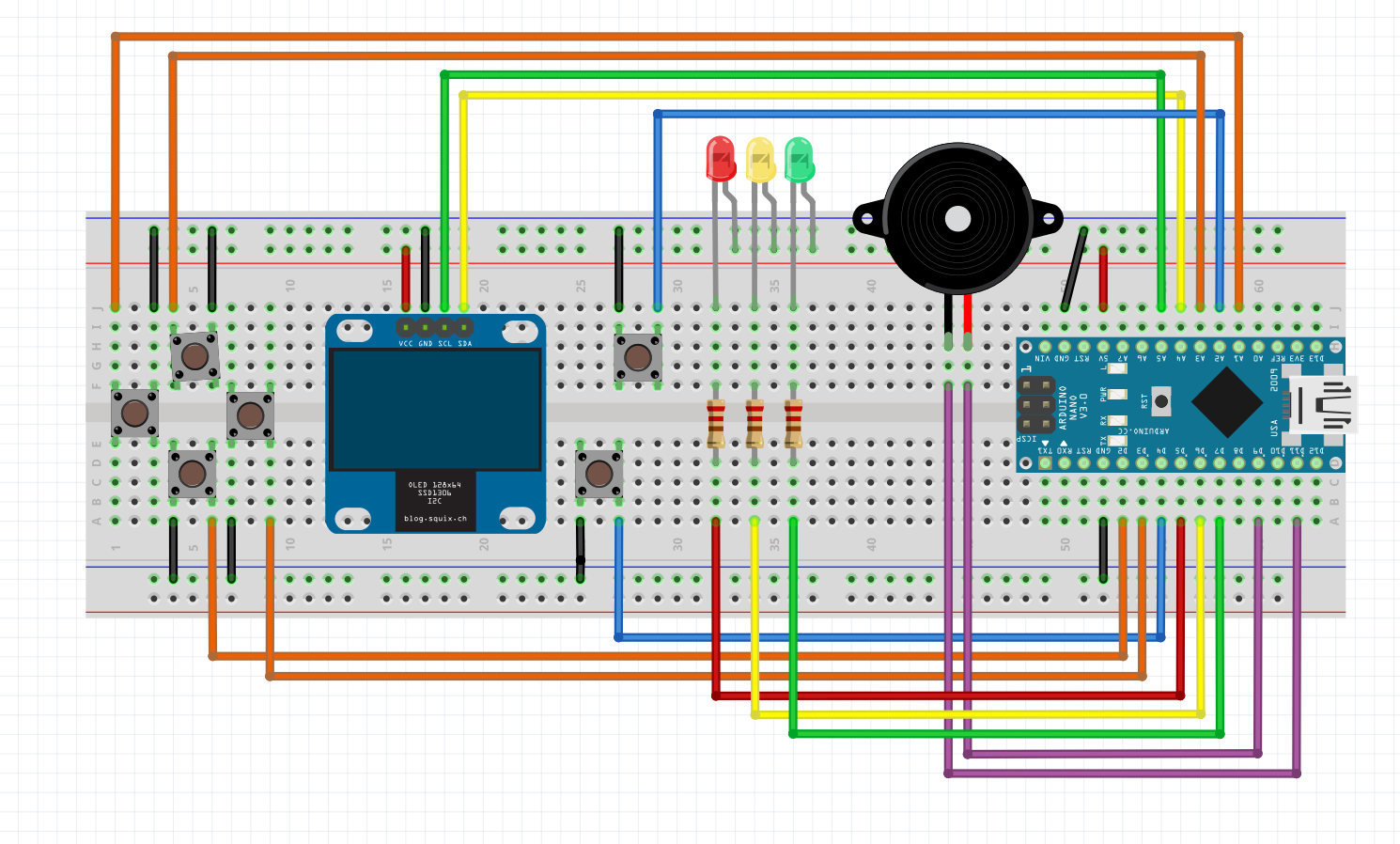
**当然，更好的采样深度是**

**会付出采样率的代价**

**建议配合外部电路处理信号**

**高**

## 电路图

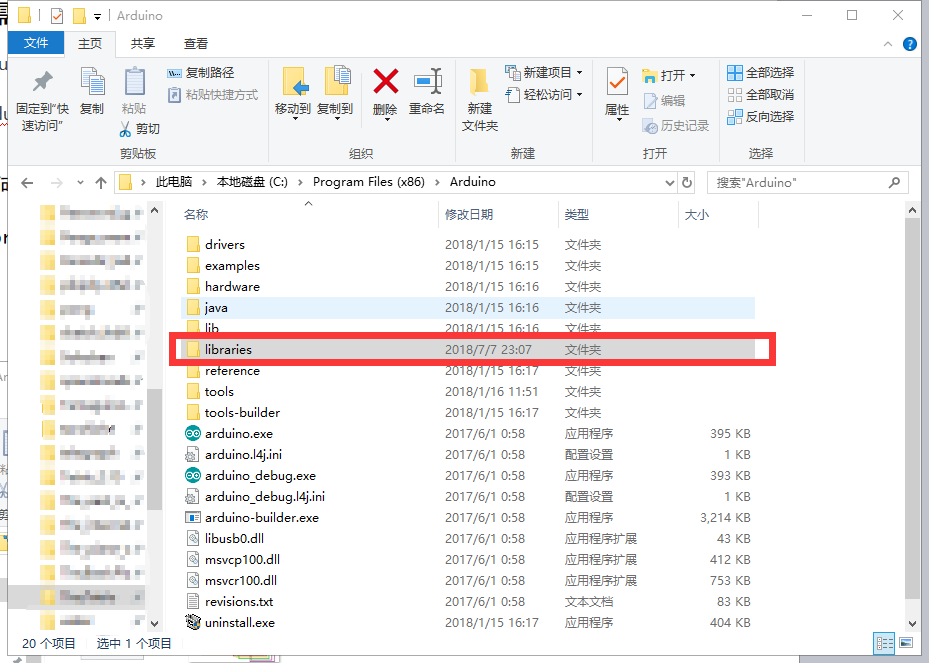


扬声器正       ===>9   
扬声器负       ===>11  
红色LED        ===>5  
蓝色LED        ===>6  
绿色LED        ===>7        (由于Atmega328p 少了一个PWM的缘故 绿色led无法调亮度)  
上键           ===>17 A3  
下键            ===>2  
左键            ===>15 A1  
右键            ===>3  
A键            ===>4  
B键          ===>16 A2  
OLED\_SCL      ===>19 A5  
OLED\_SDA    ===>18 A4

## UNO如何通过编译

所需要的库文件:  
Arduboy2   GITHUB: [https://github.com/harbaum/Arduboy2](https://github.com/harbaum/Arduboy2" \t "https://www.arduino.cn/_blank)  
 ArduboyTones GITHUB: [https://github.com/harbaum/ArduboyTones](https://github.com/harbaum/ArduboyTones" \t "https://www.arduino.cn/_blank)

##### 如何安装库文件：把库文件解压在Arduino IDE安装目录下的libraries文件夹，并重启Arduino IDE 这时候推箱子游戏可以正常编译



### **这个和真正的ARDUBOY有什么不同：** 真正的ARDUBOY采用Arduino leonardo 使用的是ATmega32u4处理器 而这个采用的是UNO或NANO 使用的是ATmega328 处理器

|  |  |  |
| --- | --- | --- |
|  | **ATmega328** | **ATmega32u4** |
| **运行频率** | 16  Mhz | 16  Mhz |
| **RAM** | 2    KB | 2.5 KB |
| **Flash** | 32   KB | 32 KB |
| **EEPROM** | 1 KB | 1 KB |

### 

### 很显然，通过对比，我们不难发现 ATmega328 的RAM少了0.5 KB,这会导致一些游戏无法运行

**由于RAM不足 所以很多游戏运行不了，使用的芯片不一样 一些游戏需要的库会不兼容，进而无法运行相关Arduboy的程序**

## 完整代码

/\*=========================================================

LHW开发

邮箱:1281702594@qq.com

采用CC协议 https://creativecommons.org/licenses/by-nc-nd/2.5/cn/

不得用于毕业设计 比赛 以及其他盈利行为，程序修改后需要以CC协议发布

提示：有个小彩蛋

=========================================================\*/

#include <Arduboy2.h>

Arduboy2 arduboy**;**

BeepPin1 beep**;**

#include <EEPROM.h>

/\*EEPROM:

0 1 2 3 4 5 6 7 8

43 52 41 43 ADCMod

\*/

/\*=========================================================

位图

=========================================================\*/

//const uint8\_t BootA[] PROGMEM = {};

const uint8\_t SC**[]** PROGMEM **=** **{**0x07**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xe0**,** 0x0f**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xf0**,** 0x1f**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xf8**,** 0x3f**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xfc**,** 0x3f**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xfc**,** 0x3f**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xfc**,** 0x3f**,** 0x80**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x01**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0x80**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x3f**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xfc**,** 0x3f**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xfc**,** 0x3f**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xfc**,** 0x3f**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xf3**,** 0xfc**,** 0x3f**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xe1**,** 0xfc**,** 0x3f**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xc0**,** 0xfc**,** 0x3f**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xc0**,** 0xfc**,** 0x3f**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xe1**,** 0xfc**,** 0x3f**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xf3**,** 0xfc**,** 0x1f**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xf8**,** 0x0f**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xf0**,** 0x07**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xe0**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x01**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0x80**,** 0x00**,** 0x00**,** 0x03**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xc0**,** 0x00**,** 0x00**,** 0x03**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xc0**,** 0x00**,** 0x00**,** 0x03**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xc0**,** 0x00**,** 0x00**,** 0x03**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xc0**,** 0x00**,** 0x00**,** 0x01**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0x80**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**};**

const uint8\_t KB**[]** PROGMEM **=** **{**0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x1f**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xe0**,** 0x7f**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xf8**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xfc**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xfc**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xfe**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xfe**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xfe**,** 0xfe**,** 0x1c**,** 0x3c**,** 0x38**,** 0x70**,** 0xf0**,** 0xe1**,** 0xfe**,** 0xfe**,** 0x1c**,** 0x18**,** 0x38**,** 0x70**,** 0x60**,** 0xe1**,** 0xfe**,** 0xfe**,** 0x1c**,** 0x18**,** 0x38**,** 0x70**,** 0x60**,** 0xe1**,** 0xfe**,** 0xfe**,** 0x1c**,** 0x18**,** 0x38**,** 0x70**,** 0x60**,** 0xe1**,** 0xfe**,** 0xfe**,** 0x1c**,** 0x38**,** 0x38**,** 0x70**,** 0x70**,** 0xe1**,** 0xfe**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xfe**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xfe**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xfe**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xfe**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xfe**,** 0xff**,** 0x87**,** 0x87**,** 0x0e**,** 0x0e**,** 0x1c**,** 0x3f**,** 0xfe**,** 0xff**,** 0x83**,** 0x07**,** 0x0e**,** 0x0e**,** 0x1c**,** 0x1f**,** 0xfe**,** 0xff**,** 0x83**,** 0x07**,** 0x0e**,** 0x0e**,** 0x1c**,** 0x1f**,** 0xfe**,** 0xff**,** 0x83**,** 0x07**,** 0x0e**,** 0x0e**,** 0x1c**,** 0x1f**,** 0xfe**,** 0xff**,** 0x87**,** 0x87**,** 0x0e**,** 0x1e**,** 0x1c**,** 0x3f**,** 0xfe**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xfe**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xfe**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xfe**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xfe**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xfe**,** 0xff**,** 0xfc**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xff**,** 0xfe**,** 0xff**,** 0xfc**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xff**,** 0xfe**,** 0xff**,** 0xfc**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xff**,** 0xfe**,** 0xff**,** 0xfc**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xff**,** 0xfe**,** 0xff**,** 0xfc**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xff**,** 0xfe**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xfe**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xfe**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xfe**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xfc**,** 0x7f**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xf8**,** 0x3f**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xf0**,** 0x1f**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xff**,** 0xe0**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**};**

const uint8\_t Kdel**[]** PROGMEM **=** **{**0x1F**,** 0x23**,** 0x57**,** 0x8B**,** 0x97**,** 0x43**,** 0x23**,** 0x1F**};**

const uint8\_t Kok**[]** PROGMEM **=** **{**0x3C**,** 0x42**,** 0x81**,** 0x85**,** 0xA9**,** 0x91**,** 0x42**,** 0x3C**};**

const uint8\_t Kup**[]** PROGMEM **=** **{**0x18**,** 0x3C**,** 0x7E**,** 0xDB**,** 0x99**,** 0x18**,** 0x18**,** 0x18**};**

/\*=========================================================

变量

=========================================================\*/

#define RxPin A0 //RX

#define TxPin 13 //TX

bool SOF**;** //声音

byte ADCMod **=** 0**;** //4-0 8-1 16-2 32-3 128-4 设置采样深度

byte ROOM **=** 0**;** //场景号 主菜单-0 RX界面-1 自定义TX-2 随机TX-3 设置-4 ADC设置-5

byte CB **=** 0**;** //按键选择返回值

byte TFL **=** 21**;** //过滤掉的电平 用于过滤干扰 ROOM-2 中 表示发送状态

byte Buffer**[**128**];** //记录波形

int ML**,** MS**;** //最长最短间隔

byte DT**;**//刷新画面间隔 单位 程序周期

byte JF**;**//跳帧

bool EL **=** **false;** //ROOM-1:电平情况 ROOM-2:是否编辑文本

bool NEL **=** **false;** //现在电平情况 ROOM-2 为TX状态

long CIT1**,** CIT2**;** //开始和上一次时间

int CILTF **=** 10**;** //这个值由ML与LS决定 ROOM-2为发送进度

int DTO **=** 90**;** //接收超时 在TX模式中为最低间隔延迟

byte DTT**[**5**];** //缓存的数据表 0代表在点的间隔低电平 1代表点 2代表划

byte DTL **=** 255**;** //缓存写入的位置 255代为禁用 也就是说没信号 在ROOM-2代表是否连续发信

//码库

const int MH**[**36**]** PROGMEM **=** **{**

12222**,**

11222**,**

11122**,**

11112**,**

11111**,**

21111**,**

22111**,**

22211**,**

22221**,**

22222**,**

12000**,**

21110**,**

21210**,**

21100**,**

10000**,**

11210**,**

22100**,**

11110**,**

11000**,**

12220**,**

21200**,**

12110**,**

22000**,**

21000**,**

22200**,**

12210**,**

22120**,**

12100**,**

11100**,**

20000**,**

11200**,**

11120**,**

12200**,**

21120**,**

21220**,**

22110

**};**

const byte ME**[**36**]** PROGMEM **=** **{**

49**,** 50**,** 51**,** 52**,** 53**,** 54**,** 55**,** 56**,** 57**,** 48**,** 65**,** 66**,** 67**,** 68**,** 69**,** 70**,** 71**,** 72**,** 73**,** 74**,** 75**,** 76**,** 77**,** 78**,** 79**,** 80**,** 81**,** 82**,** 83**,** 84**,** 85**,** 86**,** 87**,** 88**,** 89**,** 90**,**

**};** //字码库对应ascii

byte TmpString**[**21**];** //接收数据缓存显示区

byte EGG**[**10**]** **=** **{**74**,** 73**,** 78**,** 71**,** 74**,** 85**,** 67**,** 65**,** 84**,** 83**};** //哈哈，我才不会告诉你这是什么，或许可以发送来看看？

const unsigned char EggBmp**[]** PROGMEM **=** **{**0xd0**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xf8**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xe8**,** 0x00**,** 0x00**,** 0xc0**,** 0x00**,** 0x00**,** 0x01**,** 0xf8**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xe4**,** 0x00**,** 0x01**,** 0x00**,** 0x00**,** 0x00**,** 0x03**,** 0xf8**,** 0x00**,** 0xc0**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xf2**,** 0x00**,** 0x01**,** 0x20**,** 0x00**,** 0x00**,** 0x05**,** 0xf8**,** 0x00**,** 0xe0**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xf9**,** 0x00**,** 0x01**,** 0x10**,** 0x00**,** 0x00**,** 0x0b**,** 0xf8**,** 0x00**,** 0xf0**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xf8**,** 0xc0**,** 0x01**,** 0x0c**,** 0x00**,** 0x00**,** 0x13**,** 0xf8**,** 0x01**,** 0xf8**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xfc**,** 0x20**,** 0x01**,** 0x03**,** 0x00**,** 0x00**,** 0x27**,** 0xf8**,** 0x00**,** 0xf0**,** 0x00**,** 0x00**,** 0x10**,** 0x00**,** 0x00**,** 0x00**,** 0xfe**,** 0x18**,** 0x01**,** 0x00**,** 0xe1**,** 0x80**,** 0x47**,** 0xf8**,** 0x00**,** 0x77**,** 0x80**,** 0x00**,** 0x18**,** 0x00**,** 0x00**,** 0x00**,** 0xff**,** 0x04**,** 0x01**,** 0x00**,** 0x10**,** 0xc0**,** 0x8f**,** 0xf8**,** 0x00**,** 0x7f**,** 0xc0**,** 0x00**,** 0x18**,** 0x00**,** 0x00**,** 0x00**,** 0xff**,** 0x82**,** 0x00**,** 0x80**,** 0x06**,** 0x21**,** 0x0f**,** 0xf8**,** 0x01**,** 0xff**,** 0xc0**,** 0x00**,** 0x3c**,** 0x04**,** 0x00**,** 0x00**,** 0xff**,** 0xc1**,** 0x9f**,** 0xc0**,** 0x02**,** 0x26**,** 0x1f**,** 0xf8**,** 0x0f**,** 0xfc**,** 0x00**,** 0x70**,** 0x3c**,** 0x06**,** 0x00**,** 0x00**,** 0xff**,** 0xc0**,** 0x60**,** 0x30**,** 0x00**,** 0x28**,** 0x1f**,** 0xf8**,** 0x3f**,** 0xc0**,** 0x01**,** 0xf0**,** 0x3c**,** 0x07**,** 0x01**,** 0x00**,** 0xff**,** 0xe3**,** 0x00**,** 0x00**,** 0x00**,** 0x30**,** 0x3f**,** 0xf8**,** 0x7f**,** 0x3c**,** 0x07**,** 0xf0**,** 0x1c**,** 0x07**,** 0xc3**,** 0x80**,** 0xff**,** 0xf0**,** 0x00**,** 0x00**,** 0x00**,** 0x20**,** 0x7f**,** 0xf8**,** 0x38**,** 0x7e**,** 0x04**,** 0xe0**,** 0x1c**,** 0x07**,** 0xe3**,** 0x00**,** 0xff**,** 0xfe**,** 0x00**,** 0x00**,** 0x00**,** 0x20**,** 0x7f**,** 0xf8**,** 0x00**,** 0xfe**,** 0x04**,** 0xe0**,** 0x7c**,** 0x67**,** 0xf7**,** 0xc0**,** 0xfe**,** 0x7f**,** 0xc0**,** 0x00**,** 0x00**,** 0x00**,** 0xff**,** 0xf8**,** 0x01**,** 0xff**,** 0x0d**,** 0xe3**,** 0x3c**,** 0x2f**,** 0xff**,** 0x00**,** 0xff**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xff**,** 0xf0**,** 0x01**,** 0xce**,** 0x07**,** 0xc3**,** 0xbc**,** 0x3f**,** 0xb5**,** 0x80**,** 0x7f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x01**,** 0xff**,** 0xf0**,** 0x03**,** 0x8e**,** 0x07**,** 0xd3**,** 0xbc**,** 0x3d**,** 0xb0**,** 0x00**,** 0x7f**,** 0x80**,** 0x00**,** 0x00**,** 0x00**,** 0x03**,** 0xff**,** 0xf0**,** 0x01**,** 0x9e**,** 0x06**,** 0x7b**,** 0xbc**,** 0x78**,** 0x8f**,** 0x80**,** 0x7f**,** 0xc0**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xff**,** 0xf0**,** 0x01**,** 0xfc**,** 0x06**,** 0x7f**,** 0xbc**,** 0xfc**,** 0x03**,** 0xc0**,** 0x7f**,** 0xc0**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xff**,** 0xf0**,** 0x01**,** 0xfc**,** 0x07**,** 0xf3**,** 0xfd**,** 0xfc**,** 0x83**,** 0x00**,** 0x7f**,** 0xf0**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xff**,** 0xf0**,** 0x00**,** 0xf2**,** 0x07**,** 0xe0**,** 0x7f**,** 0x8d**,** 0x63**,** 0x60**,** 0x7f**,** 0xfc**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x7f**,** 0xf0**,** 0x00**,** 0x33**,** 0x07**,** 0xf8**,** 0x7e**,** 0x1c**,** 0xd3**,** 0xa0**,** 0x7f**,** 0xf7**,** 0xfc**,** 0x00**,** 0x03**,** 0xfe**,** 0x7f**,** 0xe0**,** 0x00**,** 0x31**,** 0x8d**,** 0xb8**,** 0x7c**,** 0x3c**,** 0xc3**,** 0x60**,** 0x7f**,** 0xff**,** 0xff**,** 0x80**,** 0x0f**,** 0xff**,** 0xbf**,** 0xe0**,** 0x02**,** 0x31**,** 0xff**,** 0xb8**,** 0xf8**,** 0x6c**,** 0x45**,** 0x00**,** 0x7f**,** 0xef**,** 0xff**,** 0xc0**,** 0x1f**,** 0xff**,** 0xff**,** 0xe0**,** 0x0e**,** 0x31**,** 0xf9**,** 0xf0**,** 0xf8**,** 0x0c**,** 0x7e**,** 0xc0**,** 0x7f**,** 0xc7**,** 0xff**,** 0xc0**,** 0x1f**,** 0xff**,** 0xbf**,** 0xe0**,** 0x1e**,** 0x31**,** 0xf8**,** 0xe0**,** 0xf8**,** 0x1c**,** 0x7d**,** 0x80**,** 0x7f**,** 0xc3**,** 0xff**,** 0x80**,** 0x1f**,** 0xfe**,** 0x1f**,** 0xe0**,** 0x3e**,** 0x30**,** 0x00**,** 0x00**,** 0xf8**,** 0x00**,** 0x3b**,** 0x00**,** 0x7f**,** 0x80**,** 0x07**,** 0x00**,** 0x0f**,** 0x00**,** 0x1f**,** 0xc0**,** 0x7c**,** 0x70**,** 0x00**,** 0x00**,** 0xf8**,** 0x00**,** 0x00**,** 0x00**,** 0x3f**,** 0x80**,** 0xe0**,** 0x00**,** 0x00**,** 0x00**,** 0x0f**,** 0xc0**,** 0x0c**,** 0x30**,** 0x00**,** 0x00**,** 0xf8**,** 0x00**,** 0x00**,** 0x00**,** 0x3f**,** 0x03**,** 0x7c**,** 0x00**,** 0x01**,** 0xfe**,** 0x0f**,** 0x80**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0xf0**,** 0x00**,** 0x00**,** 0x00**,** 0x1f**,** 0x06**,** 0xde**,** 0x00**,** 0x06**,** 0xfb**,** 0x07**,** 0x80**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x1e**,** 0x0c**,** 0xcf**,** 0x00**,** 0x0e**,** 0x79**,** 0x83**,** 0x80**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x0c**,** 0x18**,** 0xff**,** 0x80**,** 0x07**,** 0xfc**,** 0xc3**,** 0x03**,** 0xba**,** 0x4e**,** 0xe9**,** 0x31**,** 0x9d**,** 0xc3**,** 0x19**,** 0xb0**,** 0x04**,** 0x19**,** 0xfd**,** 0x00**,** 0x17**,** 0xec**,** 0xc1**,** 0x01**,** 0x13**,** 0x50**,** 0x49**,** 0x4a**,** 0x4a**,** 0x04**,** 0xa5**,** 0x50**,** 0x08**,** 0x18**,** 0xbf**,** 0x00**,** 0x07**,** 0xfc**,** 0xc0**,** 0x81**,** 0x13**,** 0x56**,** 0x49**,** 0x43**,** 0xc9**,** 0x84**,** 0x25**,** 0x50**,** 0x08**,** 0x0c**,** 0x9b**,** 0x47**,** 0x16**,** 0xe9**,** 0x80**,** 0x81**,** 0x12**,** 0xd2**,** 0x49**,** 0x4a**,** 0x48**,** 0x44**,** 0xa5**,** 0x50**,** 0x08**,** 0x06**,** 0x82**,** 0x05**,** 0x0a**,** 0x09**,** 0x81**,** 0x03**,** 0x3a**,** 0x4e**,** 0xc6**,** 0x32**,** 0x4b**,** 0x93**,** 0x19**,** 0x50**,** 0x04**,** 0x03**,** 0x3d**,** 0x07**,** 0x05**,** 0xb2**,** 0x02**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x02**,** 0x00**,** 0x00**,** 0x02**,** 0x01**,** 0xc0**,** 0x02**,** 0x00**,** 0x00**,** 0x06**,** 0x01**,** 0xe4**,** 0xff**,** 0xe0**,** 0x00**,** 0x00**,** 0x02**,** 0x00**,** 0x00**,** 0x77**,** 0xf8**,** 0x00**,** 0x04**,** 0x00**,** 0x00**,** 0x7f**,** 0xe7**,** 0x04**,** 0x08**,** 0x00**,** 0x00**,** 0x00**,** 0x01**,** 0x00**,** 0x00**,** 0x7f**,** 0xf8**,** 0x00**,** 0x08**,** 0x00**,** 0x00**,** 0x40**,** 0x41**,** 0x04**,** 0x08**,** 0x00**,** 0x00**,** 0x00**,** 0x01**,** 0x00**,** 0x00**,** 0x7f**,** 0xf0**,** 0x00**,** 0x08**,** 0x00**,** 0x00**,** 0x08**,** 0x0f**,** 0xfc**,** 0x0e**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x80**,** 0x00**,** 0x7f**,** 0xf0**,** 0x00**,** 0x10**,** 0x00**,** 0x00**,** 0xff**,** 0xc1**,** 0x0c**,** 0x0f**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x40**,** 0x00**,** 0x7f**,** 0xf0**,** 0x00**,** 0x20**,** 0x00**,** 0x00**,** 0x31**,** 0x03**,** 0x8c**,** 0x09**,** 0x80**,** 0x00**,** 0x00**,** 0x00**,** 0x20**,** 0x00**,** 0x3f**,** 0xe0**,** 0x00**,** 0x40**,** 0x00**,** 0x00**,** 0x32**,** 0x03**,** 0x54**,** 0x08**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x10**,** 0x00**,** 0x18**,** 0x10**,** 0x00**,** 0x80**,** 0x00**,** 0x00**,** 0x1e**,** 0x05**,** 0x54**,** 0x08**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x08**,** 0x00**,** 0x10**,** 0x08**,** 0x01**,** 0x00**,** 0x00**,** 0x00**,** 0x0e**,** 0x01**,** 0x44**,** 0x08**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x04**,** 0x00**,** 0x40**,** 0x04**,** 0x06**,** 0x00**,** 0x00**,** 0x00**,** 0x39**,** 0x81**,** 0x5c**,** 0x08**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x02**,** 0x81**,** 0x16**,** 0x02**,** 0x1c**,** 0x00**,** 0x00**,** 0x00**,** 0x60**,** 0x81**,** 0x0c**,** 0x08**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x03**,** 0xf2**,** 0x10**,** 0x02**,** 0x62**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x07**,** 0x40**,** 0x20**,** 0x01**,** 0x84**,** 0x00**,** 0x7f**,** 0xc0**,** 0x20**,** 0x08**,** 0x02**,** 0x10**,** 0xff**,** 0xe2**,** 0x24**,** 0x00**,** 0x1f**,** 0xfc**,** 0x20**,** 0x00**,** 0xff**,** 0x00**,** 0x0c**,** 0x00**,** 0x20**,** 0x18**,** 0x0f**,** 0xfe**,** 0x04**,** 0x07**,** 0xfe**,** 0x00**,** 0x3f**,** 0xf0**,** 0x20**,** 0x00**,** 0xff**,** 0xe0**,** 0x0c**,** 0x07**,** 0xff**,** 0x7f**,** 0xc7**,** 0xf8**,** 0x04**,** 0x06**,** 0x24**,** 0x00**,** 0xff**,** 0xf0**,** 0x1e**,** 0x40**,** 0xff**,** 0xf8**,** 0x0c**,** 0x00**,** 0x20**,** 0x40**,** 0xcf**,** 0x18**,** 0x0e**,** 0x07**,** 0xfe**,** 0x01**,** 0xff**,** 0xf0**,** 0x10**,** 0x20**,** 0x3f**,** 0xf8**,** 0xff**,** 0xe0**,** 0x60**,** 0x40**,** 0xc7**,** 0xfe**,** 0x0f**,** 0x8e**,** 0x00**,** 0x01**,** 0xff**,** 0xf4**,** 0x10**,** 0x00**,** 0x3f**,** 0xfc**,** 0x0e**,** 0x00**,** 0xe0**,** 0x7f**,** 0xc7**,** 0xfe**,** 0x36**,** 0xcf**,** 0xfe**,** 0x03**,** 0xff**,** 0xe4**,** 0x10**,** 0x00**,** 0x3f**,** 0xfc**,** 0x0e**,** 0x01**,** 0xa0**,** 0x40**,** 0xc7**,** 0x90**,** 0x26**,** 0x42**,** 0x64**,** 0x03**,** 0xff**,** 0xe0**,** 0x0d**,** 0x80**,** 0x9f**,** 0xfc**,** 0x1b**,** 0x03**,** 0x20**,** 0x40**,** 0xcf**,** 0xfe**,** 0xc6**,** 0x46**,** 0x7e**,** 0x03**,** 0xff**,** 0xf2**,** 0x07**,** 0x80**,** 0x9f**,** 0xfc**,** 0x31**,** 0x8e**,** 0x20**,** 0x40**,** 0xc3**,** 0xc4**,** 0x06**,** 0x03**,** 0xe4**,** 0x01**,** 0xff**,** 0xf2**,** 0x0c**,** 0xc1**,** 0x3f**,** 0xfc**,** 0x60**,** 0xe0**,** 0x60**,** 0x7f**,** 0xc7**,** 0xfe**,** 0x06**,** 0x07**,** 0x7e**,** 0x01**,** 0xff**,** 0xf1**,** 0xff**,** 0xfe**,** 0x3f**,** 0xfc**,** 0xc0**,** 0x40**,** 0x60**,** 0x40**,** 0xc2**,** 0x64**,** 0x06**,** 0x02**,** 0x64**,** 0x00**,** 0xff**,** 0xf0**,** 0x7f**,** 0xf4**,** 0x3f**,** 0xf8**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x7f**,** 0xfc**,** 0xfb**,** 0xf8**,** 0xff**,** 0xf0**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**,** 0x00**};**

byte EGGN**;**//彩蛋计数

/\*====================================================================

软重启函数

====================================================================\*/

void**(\*** resetFunc**)** **(**void**)** **=** 0**;** //制造重启命令

/\*=========================================================

只循环一次

=========================================================\*/

void setup**()**

**{**

arduboy**.**boot**();**

arduboy**.**setFrameRate**(**30**);**

beep**.**begin**();**

Serial**.**begin**(**115200**);** //初始化串口比特率

pinMode**(**RxPin**,** INPUT**);** //初始化RXpin口

byte ERC **=** 0**;** //EEProm Check

//检查EEPROM

**for** **(**int i **=** 0**;** i **<** 4**;** i**++)** **{**

ERC **=** EEPROM**.**read**(**i**)** **+** ERC**;**

**}**

**if** **(**ERC **!=** 179**)** **{**

arduboy**.**clear**();**

arduboy**.**println**(**F**(**"initialize the EEPROM"**));**

arduboy**.**print**(**ERC**);**

arduboy**.**display**();**

delay**(**1500**);**

//初始化EEPROM

**for** **(**int i **=** 0 **;** i **<** EEPROM**.**length**()** **;** i**++)** **{**

EEPROM**.**write**(**i**,** 0**);**

**}**

EEPROM**.**write**(**0**,** 43**);**

EEPROM**.**write**(**1**,** 52**);**

EEPROM**.**write**(**2**,** 41**);**

EEPROM**.**write**(**3**,** 43**);**

EEPROM**.**write**(**4**,** 0**);** //ADCMod ADC采样深度

EEPROM**.**write**(**5**,** 1**);** //SOF 声音

ERC **=** 0**;**

**for** **(**int i **=** 0**;** i **<** 7**;** i**++)** **{**

ERC **=** EEPROM**.**read**(**i**)** **+** ERC**;**

**}**

**if** **(**ERC **!=** 0**)** resetFunc**();**

**else** arduboy**.**println**(**F**(**"EEPROM writing error"**));**

arduboy**.**display**();**

delay**(**1500**);**

**}**

ADCMod **=** EEPROM**.**read**(**4**);** //读取ADCMod设置

SOF **=** EEPROM**.**read**(**5**);** //读取声音设置

ADCSET**();**

**}**

/\*=========================================================

不停循环

=========================================================\*/

void loop**()**

**{**

**if** **(**SOF **==** **true)** arduboy**.**audio**.**on**();**

**else** arduboy**.**audio**.**off**();**

key**();** //按键检测

**if** **(**ROOM **==** 0**)** JF **=** 0**;**

**else** **if** **(**ROOM **==** 1**)** **{**

JF **=** 2**;**

sampling**();** //采样

Minterval**();** //计算间隔

**}**

**if** **(**DT **==** JF **)** **{** //跳帧节约系统资源

Draw**();**

DT **=** 0**;**

**}** **else** **{**

DT**++;**

**}**

**}**

/\*=========================================================

按键检测

=========================================================\*/

void key**()**

**{**

**if** **((**arduboy**.**pressed**(**A\_BUTTON**)))** **{**

**if** **(**ROOM **==** 0**)** **{**

ROOM **=** CB **+** 1**;**

CB **=** 0**;**

**}** **else** **if** **(**ROOM **==** 4**)** **{**

**if** **(**CB **==** 0**)** **{**

//ADC设置

CB **=** ADCMod**;**

ROOM **=** 6**;**

**}**

**if** **(**CB **==** 1**)** **{**

//SOF设置

SOF **=** **!**SOF**;**

EEPROM**.**write**(**5**,** SOF**);**

**}**

**}** **else** **if** **(**ROOM **==** 2**)** **{**

**if** **(**CB **==** 0 **&&** EL **==** **false)** EL **=** **true;**

**else** **if** **(**CB **==** 2 **&&** EL **==** **false)** **{**

NEL **=** **!**NEL**;**

CILTF **=** 0**;**

DTL **=** 0**;**

**}** **else** **if** **(**CB **==** 3 **&&** EL **==** **false)** **{**

NEL **=** **!**NEL**;**

CILTF **=** 0**;**

DTL **=** 255**;**

**}**

**else** **if** **(**EL **==** **true)** **{**

EL **=** **false;**//退出编辑模式

CB **=** 0**;**

**}**

**}** **else** **if** **(**ROOM **==** 1**)** **{**

ML **=** 0**;**

MS **=** 0**;**

**}**

delay**(**100**);**

**}**

**if** **((**arduboy**.**pressed**(**B\_BUTTON**)))** **{**

**if** **(**ROOM **==** 1 **||** ROOM **==** 4 **||** ROOM **==** 2 **||** ROOM **==** 5**)** resetFunc**();**

**if** **(**ROOM **==** 6**)** **{**

ADCMod **=** CB**;**

EEPROM**.**write**(**4**,** ADCMod**);**

CB **=** 0**;**

ROOM **=** 4**;**

**}**

delay**(**200**);**

**}**

**if** **((**arduboy**.**pressed**(**LEFT\_BUTTON**))** **)** **{**

**if** **(**ROOM **==** 2 **&&** CB **>** 0 **&&** EL **==** **true)** **{**

CB**--;**

delay**(**200**);**

**}** **else** **if** **(**ROOM **==** 2 **&&** CB **==** 1 **&&** DTO **>** 10**)** DTO**--;**

**}**

**if** **((**arduboy**.**pressed**(**RIGHT\_BUTTON**))** **)** **{**

**if** **(**ROOM **==** 2 **&&** CB **<** 20 **&&** EL **==** **true)** **{**

CB**++;**

delay**(**200**);**

**}** **else** **if** **(**ROOM **==** 2 **&&** CB **==** 1 **&&** DTO **<** 2999 **)** DTO**++;**

**}**

**if** **((**arduboy**.**pressed**(**UP\_BUTTON**))** **)** **{**

**if** **(**ROOM **==** 0 **&&** CB **>** 0 **||** ROOM **==** 2 **&&** CB **>** 0 **&&** EL **==** **false** **||** ROOM **==** 4 **&&** CB **>** 0 **||** ROOM **==** 6 **&&** CB **>** 0**)** **{**

CB**--;**

delay**(**50**);**

**}**

**if** **(**ROOM **==** 1 **&&** **(**TFL **<** 31**))** TFL**++;**

**if** **(**ROOM **==** 2 **&&** EL **==** **true)** **{**

**if** **(**TmpString**[**CB**]** **<** 48 **||** TmpString**[**CB**]** **>** 57 **&&** TmpString**[**CB**]** **<** 65 **&&** TmpString**[**CB**]** **>** 90**)** TmpString**[**CB**]** **=** 48**;**

TmpString**[**CB**]++;**

**if** **(**TmpString**[**CB**]** **==** 58**)** TmpString**[**CB**]** **=** 65**;**

**else** **if** **(**TmpString**[**CB**]** **==** 91**)** TmpString**[**CB**]** **=** 32**;**

**else** **if** **(**TmpString**[**CB**]** **==** 32**)** TmpString**[**CB**]** **=** 32**;**

delay**(**100**);**

**}**

**}**

**if** **((**arduboy**.**pressed**(**DOWN\_BUTTON**)))** **{**

**if** **(**ROOM **==** 0 **&&** CB **<** 4 **||** ROOM **==** 4 **&&** CB **<** 1 **||** ROOM **==** 6 **&&** CB **<** 4 **||** ROOM **==** 2 **&&** CB **<** 3 **&&** EL **==** **false)** **{**

CB**++;**

delay**(**50**);**

**}**

**if** **(**ROOM **==** 1 **&&** **(**TFL **>** 0**))** TFL**--;**

**if** **(**ROOM **==** 2 **&&** EL **==** **true)** **{**

**if** **(**TmpString**[**CB**]** **<** 48 **||** TmpString**[**CB**]** **>** 57 **&&** TmpString**[**CB**]** **<** 65 **&&** TmpString**[**CB**]** **>** 90**)** TmpString**[**CB**]** **=** 32**;**

TmpString**[**CB**]--;**

**if** **(**TmpString**[**CB**]** **==** 31**)** TmpString**[**CB**]** **=** 90**;**

**else** **if** **(**TmpString**[**CB**]** **==** 47**)** TmpString**[**CB**]** **=** 32**;**

**else** **if** **(**TmpString**[**CB**]** **==** 64**)** TmpString**[**CB**]** **=** 57**;**

delay**(**100**);**

**}**

**}**

**}**

/\*=========================================================

显示

=========================================================\*/

void Draw**()**

**{**

arduboy**.**clear**();**

**if** **(**ROOM **==** 0**)** **{**

arduboy**.**print**(**F**(**"> MENU"**));**

arduboy**.**setCursor**(**8**,** 16**);**

arduboy**.**println**(**F**(**"1.RX"**));**

arduboy**.**setCursor**(**8**,** 24**);**

arduboy**.**println**(**F**(**"2.TX Custom content"**));**

arduboy**.**setCursor**(**8**,** 32**);**

arduboy**.**println**(**F**(**"3.TEST TOOLS"**));**

arduboy**.**setCursor**(**8**,** 40**);**

arduboy**.**println**(**F**(**"4.SET"**));**

arduboy**.**setCursor**(**8**,** 48**);**

arduboy**.**println**(**F**(**"5.ABOUT"**));**

//显示选择

arduboy**.**setCursor**(**0**,** 8 **\*** **(**CB **+** 2**));**

arduboy**.**println**(**F**(**"\*"**));**

**}**

**if** **(**ROOM **==** 1**)** **{**

DrawWav**();** //显示波形

arduboy**.**setCursor**(**0**,** 0**);**

**for** **(**int i **=** 0**;** i **<** 21**;** i**++)** **{**

arduboy**.**print**(**char**(**TmpString**[**i**]));**

**}**

arduboy**.**setCursor**(**0**,** 8**);**

arduboy**.**print**(**F**(**"ML:"**));**

arduboy**.**print**(**ML**);**

arduboy**.**setCursor**(**64**,** 8**);**

arduboy**.**print**(**F**(**"MS:"**));**

arduboy**.**print**(**MS**);**

arduboy**.**setCursor**(**0**,** 20**);**

arduboy**.**print**(**F**(**"RX:"**));**

**if** **(**DTL **!=** 255**)** arduboy**.**print**(**DTL**);**

**else** arduboy**.**print**(**F**(**"N/A"**));**

arduboy**.**setCursor**(**64**,** 20**);**

arduboy**.**print**(**F**(**"DTT:"**));**

**for** **(**int i **=** 0**;** i **<** 5**;** i**++)** **{**

arduboy**.**print**(**DTT**[**i**]);**

**}**

**for** **(**int i **=** 0**;** i **<** 10**;** i**++)** **{** //增加气氛...我知道这注释很扯...管他的

**if** **(**TmpString**[**i **+** 11**]** **==** EGG**[**i**])** **{**

EGGN**++;**

**}** **else** EGGN **=** 0**;**

**}**

**if** **(**EGGN **==** 10**)** VEGG**();**

**}**

**if** **(**ROOM **==** 2**)** **{**

**if** **(**NEL **==** **true)** **{**

pinMode**(**TxPin**,** OUTPUT**);** //初始化TXpin口

**for** **(**byte i **=** 0**;** i **<** 36**;** i**++)** **{** //匹配数据库信号

**if** **(**byte**(**pgm\_read\_word\_near**(&**ME**[**i**]))** **==** byte**(**TmpString**[**CILTF**]))** **{**

String TXT **=** String**(**pgm\_read\_word\_near**(&**MH**[**i**]));**

**for** **(**byte I **=** 0**;** I **<** 5**;** I**++)** **{**

**if** **(**char**(**TXT**[**I**])** **==** 49**)** **{**

digitalWrite**(**TxPin**,** HIGH**);**

beep**.**tone**(**440**);**

delay**(**DTO**);**

**}** **else** **if** **(**char**(**TXT**[**I**])** **==** 50**)** **{**

digitalWrite**(**TxPin**,** HIGH**);**

beep**.**tone**(**440**);**

delay**(**DTO **\*** 3**);**

**}**

digitalWrite**(**TxPin**,** LOW**);** //间隔信号

beep**.**noTone**();**

delay**(**DTO**);**

**}**

**}**

**}**

delay**(**DTO **\*** 3**);**

CILTF**++;**

**if** **(**CILTF **>=** 21**)** **{**

**if** **(**DTL **!=** 255**)** NEL **=** **!**NEL**;**

CILTF **=** 0**;**

**};**

**}**

arduboy**.**println**(**F**(**">TX"**));**

arduboy**.**drawRect**(**0**,** 12**,** 128**,** 16**,** 1**);**

arduboy**.**setCursor**(**48**,** 8**);**

arduboy**.**println**(**F**(**"Text"**));**

arduboy**.**setCursor**(**1**,** 16**);**

**for** **(**int i **=** 0**;** i **<** 21**;** i**++)** **{**

arduboy**.**print**(**char**(**TmpString**[**i**]));**

**}**

arduboy**.**setCursor**(**8**,** 32**);**

arduboy**.**print**(**F**(**"Edit Text"**));**

arduboy**.**setCursor**(**8**,** 40**);**

arduboy**.**print**(**F**(**"Delay "**));**

arduboy**.**print**(**DTO**);**

arduboy**.**print**(**F**(**"ms"**));**

arduboy**.**drawLine**(**96**,** 44**,** 126**,** 44**,** 1**);** //横线

arduboy**.**drawLine**(**96**,** 40**,** 96**,** 48**,** 1**);** //左侧竖线

arduboy**.**drawLine**(**127**,** 40**,** 127**,** 48**,** 1**);** //右侧竖线

arduboy**.**fillRect**(**96**,** 42**,** map**(**DTO**,** 10**,** 3000**,** 0**,** 32**),** 5**,** 1**);**

arduboy**.**setCursor**(**8**,** 48**);**

arduboy**.**print**(**F**(**"TX "**));**

arduboy**.**print**(**NEL**);**

arduboy**.**setCursor**(**8**,** 56**);**

arduboy**.**print**(**F**(**"continuous TX"**));**

//发信

//显示选择

**if** **(**EL **==** **false)** **{**

arduboy**.**setCursor**(**0**,** 8 **\*** **(**CB **+** 4**));**

arduboy**.**println**(**F**(**"\*"**));**

**if** **(**NEL **==** **true)** **{**

arduboy**.**setCursor**(**6 **\*** CILTF **+** 1**,** 24**);** //在编辑栏左右移动

arduboy**.**println**(**F**(**"\*"**));**

**}**

**}** **else** **{**

arduboy**.**setCursor**(**6 **\*** CB **+** 1**,** 24**);** //在编辑栏左右移动

arduboy**.**println**(**F**(**"\*"**));**

**}**

**}**

**if** **(**ROOM **==** 3**)** **{**

DIAGNOSTIC**();**

**}**

**if** **(**ROOM **==** 4**)** **{**

arduboy**.**print**(**F**(**"> SET"**));**

arduboy**.**setCursor**(**8**,** 16**);**

arduboy**.**println**(**F**(**"1.ADC"**));**

arduboy**.**setCursor**(**8**,** 24**);**

arduboy**.**print**(**F**(**"1.Sound "**));**

arduboy**.**print**(**SOF**);**

//显示选择

arduboy**.**setCursor**(**0**,** 8 **\*** **(**CB **+** 2**));**

arduboy**.**println**(**F**(**"\*"**));**

**}**

**if** **(**ROOM **==** 5**)** **{**

arduboy**.**println**(**F**(**"> ABOUT"**));**

arduboy**.**println**();**

arduboy**.**println**(**F**(**"LHW programming"**));**

arduboy**.**println**(**F**(**"The program adopts CC"**));**

arduboy**.**println**(**F**(**" open source protocol"**));**

arduboy**.**println**(**F**(**"E-M:1281702594@qq.com"**));**

arduboy**.**println**(**F**(**"Version beta2.1"**));**

arduboy**.**println**(**F**(**"Press the B key"**));**

**}**

**if** **(**ROOM **==** 6**)** **{**

arduboy**.**print**(**F**(**"> ADC"**));**

arduboy**.**setCursor**(**8**,** 16**);**

arduboy**.**println**(**F**(**"4 BIT"**));**

arduboy**.**setCursor**(**8**,** 24**);**

arduboy**.**println**(**F**(**"8 BIT"**));**

arduboy**.**setCursor**(**8**,** 32**);**

arduboy**.**println**(**F**(**"16 BIT"**));**

arduboy**.**setCursor**(**8**,** 40**);**

arduboy**.**println**(**F**(**"32 BIT"**));**

arduboy**.**setCursor**(**8**,** 48**);**

arduboy**.**println**(**F**(**"128 BIT"**));**

//显示选择

arduboy**.**setCursor**(**0**,** 8 **\*** **(**CB **+** 2**));**

arduboy**.**println**(**F**(**"\*"**));**

**}**

**if** **(**ROOM **==** 4 **||** ROOM **==** 6**)** **{**

arduboy**.**setCursor**(**0**,** 56**);**

arduboy**.**println**(**F**(**"Press \"B\" to Back"**));**

**}**

arduboy**.**display**();**

**}**

/\*=========================================================

VEGG

=========================================================\*/

void VEGG**()**

**{**

arduboy**.**clear**();**

arduboy**.**drawSlowXYBitmap**(**0**,** 0**,** EggBmp**,** 128**,** 64**,** 1**);**

arduboy**.**display**();**

**while** **(**1**)** **{}**

**}**

/\*=========================================================

清除数据缓冲

=========================================================\*/

void CM**()**

**{**

**for** **(**int i **=** 0**;** i **<** 5**;** i**++)** DTT**[**i**]** **=** 0**;**

**}**

/\*=========================================================

译码

=========================================================\*/

void translation**()**

**{**

**if** **(**CIT2 **>=** ML **-** CILTF **&&** CIT2 **<=** ML **+** CILTF **&&** ML **!=** 0 **&&** EL **==** **true)** **{**

//合法数据 写入缓存

DTT**[**DTL**]** **=** 2**;**

DTL**++;**

**}** **else** **{**

**if** **(**CIT2 **>=** MS **-** CILTF **&&** CIT2 **<=** MS **+** CILTF **&&** MS **!=** 0 **&&** EL **==** **true)** **{**

//合法数据 写入缓存

DTT**[**DTL**]** **=** 1**;**

DTL**++;**

**}** /\*else {

if (CIT2 >= MS - CILTF && CIT2 <= MS + CILTF && MS != 0 && EL == false) {

//合法数据 写入缓存

DTT[DTL] = 0;

DTL++;

}

}\*/

**}**

TOStr**();** //超时合并字符串

**}**

/\*=========================================================

超时合并字符串

=========================================================\*/

void TOStr**()**

**{**

**if** **((** millis**()** **/** 10.0**)** **>=** CIT1 **+** DTO **||** DTL **>** 4 **)** **{** //译码时间超时 确认不在有数据 数据合并

//数据接收完成 查询数据库

String TXT**;**

**for** **(**int i **=** 0**;** i **<** 5**;** i**++)** **{** //把接收区缓存合成字符串

TXT **=** TXT **+** DTT**[**i**];**

**}**

**for** **(**int i **=** 0**;** i **<** 36**;** i**++)** **{** //匹配数据库信号

**if** **(**String**(**pgm\_read\_word\_near**(&**MH**[**i**]))** **==** TXT**)** **{**

Serial**.**println**(**char**(**pgm\_read\_byte\_near**(&**ME**[**i**])));**

//执行显示区数据移位

**for** **(**byte i **=** 0**;** i **<** 20**;** i**++)** **{**

TmpString**[**i**]** **=** TmpString**[**i **+** 1**];**

**}**

TmpString**[**20**]** **=** pgm\_read\_byte\_near**(&**ME**[**i**]);** //写到显示区最后的位置

CM**();**

**}**

**}**

DTL **=** 255**;** //无信号 禁用译码

**}**

**}**

/\*=========================================================

计算间隔

=========================================================\*/

void Minterval**()**

**{**

**if** **(**map**(**Buffer**[**127**],** 63**,** 31**,** 0**,** 31**)** **>=** TFL**)** **{**

beep**.**tone**(**440**);**

NEL **=** **true;**

**}** **else** **{**

beep**.**noTone**();**

NEL **=** **false;** //根据过滤获取电平情况

**}**

**if** **(**NEL **!=** EL**)** **{**

//电平发生了变化

CIT2 **=** **(** millis**()** **/** 10.0**)** **-** CIT1**;**

CIT1 **=** millis**()** **/** 10.0**;** //获取开始计算时的时间

**if** **(**DTL **!=** 255**)** translation**();** //译码

//计算最长和最短电平

**if** **(**EL **==** **true)** **{**

**if** **(**CIT2 **>** MS **)** **{**

**if** **(**CIT2 **>=** ML **-** CILTF **&&** CIT2 **<=** ML **+** CILTF **||** ML **==** 0**)** **{**

ML **=** CIT2**;**

DTO **=** ML **+** CILTF**;**

**}** **else** **{**

**if** **(**CIT2 **>** ML **&&** CIT2 **<=** ML **\*** 4**)** **{**

ML **=** CIT2**;**

DTO **=** ML **+** CILTF**;**

**}**

**}**

**}**

**}** **else** **{**

//低电平

**if** **(**CIT2 **>=** DTO**)** **{**

//在稍微长的时间收到信号 可能是数据开头 到这里代表已经成功学习信号规律

DTL **=** 0**;** //写入位置为0

CM**();**

**}**

**}**

**if** **(**CIT2 **<=** ML **-** CILTF **&&** CIT2 **>** 5**)** **{**

MS **=** CIT2**;**

**}**

**if** **(**ML **!=** 0 **&&** MS **!=** 0**)** **{**

CILTF **=** **((**ML **+** MS**)** **/** 2.0**)** **\*** 0.6**;**

**}**

EL **=** **!**EL**;** //反转电平状态

**}**

**if** **((** millis**()** **/** 10.0**)** **>=** CIT1 **+** DTO **&&** DTL **!=** 255**)** **{**

TOStr**();** //超时

**}**

**}**

/\*=========================================================

采样和移位

=========================================================\*/

void sampling**()**

**{**

//执行数据移位

**for** **(**byte i **=** 0**;** i **<** 127**;** i**++)** **{**

Buffer**[**i**]** **=** Buffer**[**i **+** 1**];**

**}**

//执行采样

Buffer**[**127**]** **=** map**(**analogRead**(**RxPin**),** 0**,** 1023**,** 63**,** 31**);**

**}**

/\*=========================================================

画出波形

=========================================================\*/

void DrawWav**()**

**{**

//显示波形输入

**for** **(**byte i **=** 0**;** i **<** 127**;** i**++)** **{**

arduboy**.**drawLine**(**i**,** 64**,** i**,** Buffer**[**i**],** 1**);**

**}**

arduboy**.**drawLine**(**0**,** 31**,** 128**,** 31**,** 1**);** //波形上端的隔离线

arduboy**.**drawLine**(**0**,** map**(**TFL**,** 0**,** 31**,** 63**,** 31**),** 128**,** map**(**TFL**,** 0**,** 31**,** 63**,** 31**),** 1**);** //干扰过滤线

**}**

/\*=========================================================

ADC深度设置

=========================================================\*/

void ADCSET**()**

**{**

**if** **(**ADCMod **==** 0**)** setP4**();**

**else** **if** **(**ADCMod **==** 1**)** setP8**();**

**else** **if** **(**ADCMod **==** 2**)** setP16**();**

**else** **if** **(**ADCMod **==** 3**)** setP32**();**

**else** **if** **(**ADCMod **==** 4**)** setP128**();**

**}**

/\*=========================================================

ADC深度设置

=========================================================\*/

void setP32**(** **)**

**{**

ADCSRA **|=** **(**1 **<<** ADPS2**);** // 1

ADCSRA **&=** **~(**1 **<<** ADPS1**);** // 0

ADCSRA **|=** **(**1 **<<** ADPS0**);** // 1

**}**

void setP16**(** **)**

**{**

ADCSRA **|=** **(**1 **<<** ADPS2**);** // 1

ADCSRA **&=** **~(**1 **<<** ADPS1**);** // 0

ADCSRA **&=** **~(**1 **<<** ADPS0**);** // 0

**}**

void setP8**(** **)** // prescaler = 8

**{**

ADCSRA **&=** **~(**1 **<<** ADPS2**);** // 0

ADCSRA **|=** **(**1 **<<** ADPS1**);** // 1

ADCSRA **|=** **(**1 **<<** ADPS0**);** // 1

**}**

void setP4**(** **)** // prescaler = 4

**{**

ADCSRA **&=** **~(**1 **<<** ADPS2**);** // 0

ADCSRA **|=** **(**1 **<<** ADPS1**);** // 1

ADCSRA **&=** **~(**1 **<<** ADPS0**);** // 0

**}**

void setP128**(** **)** // 默認就是這樣

**{**

ADCSRA **|=** **(**1 **<<** ADPS2**);** // 1

ADCSRA **|=** **(**1 **<<** ADPS1**);** // 1

ADCSRA **|=** **(**1 **<<** ADPS0**);** // 1

**}** // setP128

/\*=========================================================

测试工具

=========================================================\*/

void DIAGNOSTIC**()** **{**

long T0 **=** millis**()** **/** 1000.0 **+** 5**;**

**while** **(**T0 **>** long**(**millis**()** **/** 1000.0**))** **{**

**if** **((**arduboy**.**pressed**(**B\_BUTTON**)))** resetFunc**();**

arduboy**.**clear**();**

arduboy**.**setCursor**(**0**,** 0**);**

arduboy**.**println**(**F**(**"DIAGNOSTIC"**));**

arduboy**.**print**(**F**(**"Continue for "**));**

arduboy**.**print**(**T0 **-** **(**millis**()** **/** 1000.0**));**

arduboy**.**print**(**F**(**" S"**));**

arduboy**.**setCursor**(**0**,** 56**);**

arduboy**.**println**(**F**(**"Press B Key Cancel"**));**

arduboy**.**display**();**

**}**

//测试按键

byte TB**[**6**];**

**for** **(**T0 **=** 0**;** T0 **<** 6**;** T0**++)** **{**

TB**[**T0**]** **=** 255**;**

**}**

**while** **(**int**(**TB**[**0**]** **+** TB**[**1**]** **+** TB**[**2**]** **+** TB**[**3**]** **+** TB**[**4**]** **+** TB**[**5**])** **!=** 0**)** **{**

**if** **((**arduboy**.**pressed**(**UP\_BUTTON**))** **&&** TB**[**0**]** **!=** 0**)** TB**[**0**]--;**

**if** **((**arduboy**.**pressed**(**DOWN\_BUTTON**))** **&&** TB**[**1**]** **!=** 0**)** TB**[**1**]--;**

**if** **((**arduboy**.**pressed**(**LEFT\_BUTTON**))** **&&** TB**[**2**]** **!=** 0**)** TB**[**2**]--;**

**if** **((**arduboy**.**pressed**(**RIGHT\_BUTTON**))** **&&** TB**[**3**]** **!=** 0**)** TB**[**3**]--;**

**if** **((**arduboy**.**pressed**(**A\_BUTTON**))** **&&** TB**[**4**]** **!=** 0**)** TB**[**4**]--;**

**if** **((**arduboy**.**pressed**(**B\_BUTTON**))** **&&** TB**[**5**]** **!=** 0**)** TB**[**5**]--;**

arduboy**.**clear**();**

arduboy**.**println**(**F**(**"Button test"**));**

arduboy**.**println**();**

arduboy**.**print**(**F**(**"UP "**));**

arduboy**.**println**(**TB**[**0**]);**

arduboy**.**print**(**F**(**"DOWN "**));**

arduboy**.**println**(**TB**[**1**]);**

arduboy**.**print**(**F**(**"LEFT "**));**

arduboy**.**println**(**TB**[**2**]);**

arduboy**.**print**(**F**(**"RIGHT "**));**

arduboy**.**println**(**TB**[**3**]);**

arduboy**.**print**(**F**(**"A "**));**

arduboy**.**println**(**TB**[**4**]);**

arduboy**.**print**(**F**(**"B "**));**

arduboy**.**println**(**TB**[**5**]);**

arduboy**.**display**();**

**}**

delay**(**1000**);**

//测试屏幕

**for** **(**T0 **=** 0**;** T0 **<** 6**;** T0**)** **{**

**if** **((**arduboy**.**pressed**(**A\_BUTTON**)))** **{**

T0**++;**

delay**(**200**);**

**}**

arduboy**.**clear**();**

**if** **(**T0 **==** 0**)** arduboy**.**fillRect**(**0**,** 0**,** 128**,** 64**,** 1**);** **else** **if** **(**T0 **==** 2**)** arduboy**.**drawRect**(**0**,** 0**,** 128**,** 64**,** 1**);** **else** **if** **(**T0 **==** 3 **||** T0 **==** 4**)** **{**

**for** **(**int y **=** 0**;** y **<** 64**;** y**++** **)** **{**

**for** **(**int x **=** 0**;** x **<** 129**;** x**++** **)** **{** //129不解释

**if** **(** **(**x **+** y**)** **&** 1 **)** **{**

arduboy**.**drawPixel**(**x **+** T0 **-** 4**,** y**,** 1**);**

**}**

**}**

**}**

**}** **else** **if** **(**T0 **==** 5**)** **{**

**for** **(**int y **=** 0**;** y **<** 8**;** y**++** **)** **{**

**for** **(**int x **=** 0**;** x **<** 21**;** x**++** **)** **{** //129不解释

arduboy**.**print**(**F**(**"@"**));**

**}**

arduboy**.**println**();**

**}**

**}**

arduboy**.**display**();**

**}**

//EEPROM测试

**for** **(**T0 **=** 0**;** T0 **<** EEPROM**.**length**();** T0**++)** **{**

arduboy**.**clear**();**

byte Ran **=** random**(**255**);**

int EP**,** EF**;**

arduboy**.**println**(**F**(**"EEPROM TEST"**));**

arduboy**.**println**();**

arduboy**.**print**(**F**(**"Add "**));**

arduboy**.**print**(**T0**);**

arduboy**.**print**(**F**(**" / "**));**

arduboy**.**println**(**EEPROM**.**length**()** **-** 1**);**

arduboy**.**print**(**F**(**"Write "**));**

arduboy**.**println**(**Ran**);**

EEPROM**.**write**(**T0**,** Ran**);**

arduboy**.**print**(**F**(**"Read "**));**

arduboy**.**println**(**EEPROM**.**read**(**T0**));**

**if** **(**EEPROM**.**read**(**T0**)** **==** Ran**)** **{**

arduboy**.**println**(**F**(**" PASS"**));**

EP**++;**

**}** **else** **{**

arduboy**.**println**(**F**(**"Fail"**));**

EF**++;**

delay**(**1000**);**

**}**

EEPROM**.**write**(**T0**,** 0**);**

arduboy**.**println**();**

arduboy**.**print**(**F**(**"PASS "**));**

arduboy**.**print**(**EP**);**

arduboy**.**print**(**F**(**" Fail "**));**

arduboy**.**println**(**EF**);**

arduboy**.**display**();**

**}**

delay**(**1000**);**

//测试完成

arduboy**.**clear**();**

arduboy**.**println**(**F**(**"TEST OK!"**));**

arduboy**.**println**();**

arduboy**.**println**(**F**(**"A Key Reset All"**));**

arduboy**.**display**();**

**while** **(!(**arduboy**.**pressed**(**A\_BUTTON**)))** **{}**

resetFunc**();**

**}**