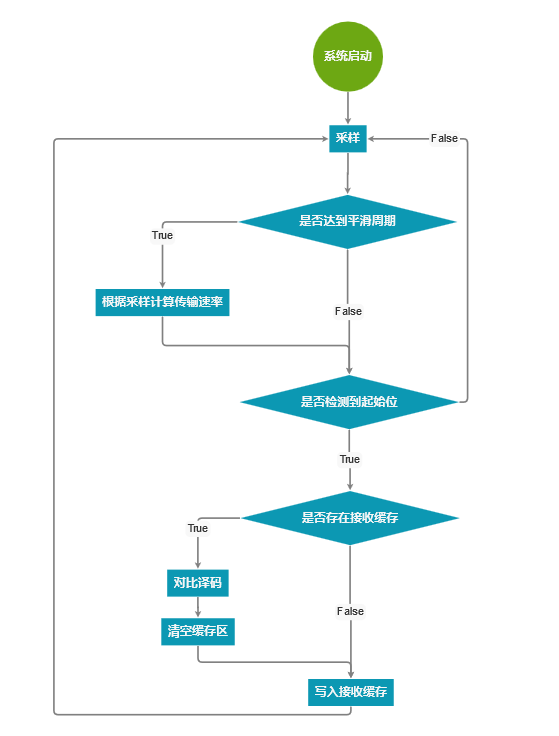
# 部分摩尔斯电报译码器 LHW

## 工作流程



## 自适应传输速率以及译码

**传输速率:**

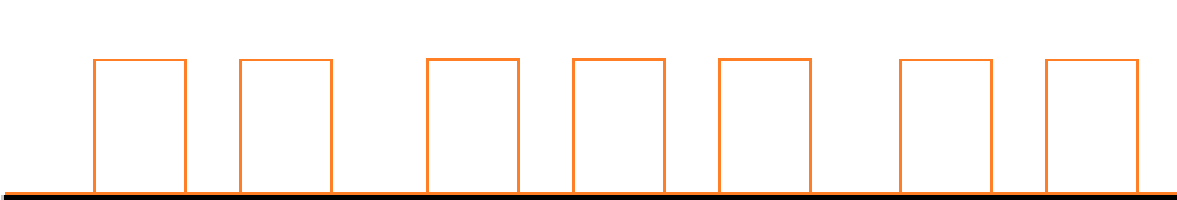
M /--/

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，而LL是所谓的数据头(时间比较长的低电平状态)

为了识别出信号的长短，我们首先需要知道信号长短的时间，于是乎我们可以在开机时进行一定量的数据采样，因为每个完整的信号短都包含了长和短的情况，可以计算出长和短信号时间的平均值，根据这个平均值对长短信号分两个数列存储，最后用平滑算法算出大概的延迟时间，这样就做到自动确定通讯速率!

**译码:**

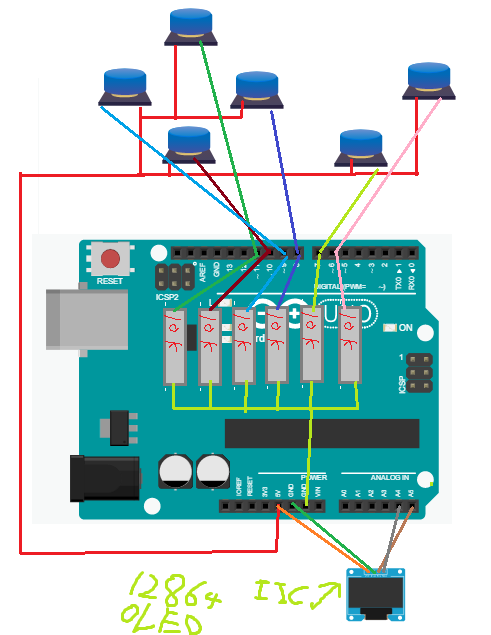
由上文可知，我们获取到了信号长短的延迟，并且能够对比长短和电平来识别出数据

当检测到数据头后假如存在上一个字符的缓存会进行翻译。

翻译过程就是根据LS,HS,HL状态分为0,1,2 然后对比码库

如果没有存在缓存，那么进行从检测到数据头开始把接收到的信号存储在缓存里

## 电路图



## 完整代码

#include "U8glib.h"//引用U8G头文件

U8GLIB\_SSD1306\_128X64 u8g**(**U8G\_I2C\_OPT\_NONE**);**//设置设备名称：I2C-SSD1306-128\*64(OLED)

//如果屏幕不同请自行修改 如果第一次使用u8glib并且遇到显示不正确，请自行修改u8g配置

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

变量

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

int x**,** y**;** int Buffer**[**128**];** //

int Time **=** 0**;** //计时

int DPin **=** A0**;** //数据接收口

byte RXB**[**10**];** //接收缓冲区 0为中间间隔 1为短高 2为长高

int SA**[**10**];** //初始数据平均值

int DA**;**

int HH**[**10**];** //平滑数据 最高的长度

int AH**,** AL**;** //平滑后的数据

byte TH**,** TT**,** TRX**,** MT**;** //位置

bool ET **=** **false;** //电平是否变化

byte BootOk **=** 0**;** //初始化阶段 0为学习通讯速率 1为计算基准值 2为完成通讯速率学习 3为检测到数据起始符 4为工作状态

byte DFT **=** 35**;** //数据正方向容错值

bool WI **=** **false;** //工作状态

//码库

long MH**[**36**]** **=** **{**

1020202020**,**

1010202020**,**

1010102020**,**

1010101020**,**

1010101010**,**

2010101010**,**

2020101010**,**

2020201010**,**

2020202010**,**

2020202020**,**

1020000000**,**

2010101000**,**

2010201000**,**

2010100000**,**

1000000000**,**

1010201000**,**

2020100000**,**

1010101000**,**

1010000000**,**

1020202000**,**

2010200000**,**

1020101000**,**

2020000000**,**

2010000000**,**

2020200000**,**

1020201000**,**

2020102000**,**

1020100000**,**

1010100000**,**

2000000000**,**

1010200000**,**

1010102000**,**

1020200000**,**

2010102000**,**

2010202000**,**

2020101000

**};**

char ME**[**36**]** **=** **{**

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**,**

**};**

char MS**[**32**]** **=** **{**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**,};** //信息显示区

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

只循环一次

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

void setup**()**

**{**

pinMode**(**DPin**,** INPUT**);** //初始化接收pin口

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

u8g**.**setFont**(**u8g\_font\_04b\_03r**);**

u8g**.**firstPage**();**

**do** **{**

u8g**.**setPrintPos**(**13**,** 33**);**

u8g**.**print**(**"Automatic detection rate"**);**

**}** **while** **(** u8g**.**nextPage**()** **);**

**}**

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

不停循环

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

void loop**()**

**{**

scanning**();** //扫描

**}**

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

显示

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

void Draw**()** **{**

u8g**.**firstPage**();**

**do** **{**

int i **=** 0**;**

**for** **(**int py **=** 0**;** py **<** 2**;** py**++)** **{**

**for** **(**int px **=** 0**;** px **<** 16**;** px**++)** **{**

u8g**.**setPrintPos**(**px **\*** 8 **+** 2**,** py **\*** 8 **+** 6**);**

u8g**.**print**((**char**)**MS**[**i**]);**

i**++;**

**}**

**}**

**}** **while** **(** u8g**.**nextPage**()** **);**

**}**

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

清除数据缓冲

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

void CM**()** **{**

**for** **(**int i **=** 0**;** i **<** 10**;** i**++)** RXB**[**i**]** **=** 0**;**

**}**

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

起始符处理

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

void Tse**()** **{**

//检测到数据起始符

String TXT**;**

**for** **(**int i **=** 0**;** i **<** 10**;** i**++)** **{** //把接收区缓存合成字符串

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

**}**

Serial**.**println**(**TXT**);**

**for** **(**int i **=** 0**;** i **<** 35**;** i**++)** **{** //匹配数据库信号

**if** **(**String**(**MH**[**i**])** **==** TXT**)** **{**

MS**[**MT**]** **=** ME**[**i**];** //把信息写入 显示缓冲 区

MT**++;**

**if** **(**MT **>** 31**)** MT **=** 0**;** //如果显示区溢出，覆盖旧的信息

Draw**();**

**}**

**}**

CM**();** //清除接收区缓存

TRX **=** 0**;** //接收区缓存地址重置

**}**

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

译码

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

void translation**()**

**{**

**if** **(**Time **>=** AH **-** DFT **&&** Time **<=** AH **+** DFT **&&** **!**ET**)** **{**

Tse**();**

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

**if** **(**TRX **>** 9**)** TRX **=** 0**;** //数据接收错误 初始化

**if** **(!**ET**)** **{** //低电平

**if** **(**Time **>=** AL **-** DFT **&&** Time **<=** AL **+** DFT**)** **{**

RXB**[**TRX**]** **=** 0**;**

TRX**++;**

**}**

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

**if** **(**Time **>=** AL **-** DFT **&&** Time **<=** AL **+** DFT**)** **{**

RXB**[**TRX**]** **=** 1**;**

TRX**++;**

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

**if** **(**Time **>=** AL **+** DFT **&&** Time **<=** AH **+** DFT**)** **{**

RXB**[**TRX**]** **=** 2**;**

TRX**++;**

**}**

**}**

**}**

**}**

**}**

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

扫描

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

void scanning**()** **{**

**if** **(**digitalRead**(**DPin**)** **!=** ET**)** **{** //电平发生改变

Time **=** 0**;** //重置计时

ET **=** digitalRead**(**DPin**);** //重置电平状态

**while** **(**digitalRead**(**DPin**)** **==** ET**)** **{** //检测电平是否再次改变

**if** **(**Time **>=** 10000**)** **{** //大于10秒接收超时

/\* Serial.print(millis());

Serial.println(" : TimeOut !");\*/

**goto** EndRx**;** //超时

**}**

**if** **(**RXB**[**TRX **-** 1**]** **!=** 0 **&&** Time **>=** AH **+** 30**)** **{**

//可能是通讯结尾 并且不会再有数据传送

Tse**();**

**}**

Time**++;** //增加计时

delay**(**1**);** //延时1ms

**}**

EndRx**:**

**if** **(**Time **<** 10000**)** **{**

//如果没有超时

Serial**.**print**(**millis**());**

Serial**.**print**(**" : "**);**

Serial**.**print**(**ET**);**

Serial**.**print**(**" "**);**

Serial**.**println**(**Time**);**

//学习波形

**if** **(**BootOk **==** 0**)** **{**

SA**[**TH**]** **=** Time**;**

**if** **(**TH **>** 8**)** **{**

BootOk **=** 1**;**

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

TH**++;**

**}**

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

**if** **(**BootOk **==** 1**)** **{**

//计算信号基准值

**for** **(**int i **=** 0**;** i **<** 10**;** i**++)** DA **=** DA **+** SA**[**i**];** //相加

DA **=** float**(**DA **/** 10**);** //获取平均值

BootOk**++;**

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

**if** **(**BootOk **==** 2**)** **{**

**if** **(**Time **>** DA**)** **{**

HH**[**TT**]** **=** Time**;** //写入最高平滑

TT**++;**

**if** **(**TT **>** 9**)** **{**

TT **=** 0**;**

AH **=** 0**;**

**for** **(**int i **=** 0**;** i **<** 10**;** i**++)** AH **=** AH **+** HH**[**i**];** //相加

AH **=** float**(**AH **/** 10**);** //获取平均值

AL **=** float**(**AH **/** 3**);** //获取平均值

BootOk**++;**

**}**

**}**

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

translation**();** //译码

**}**

**}**

**}**

**}**

**}**

**}**