# How To Write The Display code

Souhaiel

December 26, 2020

## Contents

# 1 How To Program a Display

1. write a SendCommand and SendData functions
2. initilase all register needed from the data sheet
3. program a setPowerFunction() to privide display on/off
4. dowload a fonts from the page of the address
5. write a draw byte function
6. write a draw bytes function
7. write a draw character function
8. write a draw a text function

## 1.1 write a SendCommand and SendData functions

### 1.1.1 Send Command

1. Find The display command mode Byte From the data sheet , in OLed it is 0x00 (c0 bit = 0 , Dc bit = 0)
2. Send a Buffer contain command mode Byte(0x00) , uint8t command , ....,

### 1.1.2 Send Data

1. Find the Display Data mode Byte From the data sheet, in Oled it is 0x40 ,(c0 bit = 0 , dc bit = 1)
2. Send a Buffer frame contain {Data mode Byte(0x40) , uint8t data , ....,}
3. when send a buffer frame the frame should contain all the pixels width and height . because the display will write in all pixel then repeat from the start position x , y = 0 . that mean if our frame = "hallo world" will charge only 127 in the frame memory the reset of the frame size

= 1024 should be sended to the display with 0x00 . if not than the next transaction the display will write our text directly aftrer the first the text postion "hallo world " then " our new text" . the display will point to the next frame postion after hallo world . to avoid that with should write the 125 buffer size of hallo wrold than write 0x00 in the rest of 1024 to be sure that the display will begin from the the start postion x = 0 y = 0 by the next transaction

### 1.2 initilase all register needed from the data sheet

1. Find all register need from the data sheet like Muliplexer , dataclock , charg pump , Lower column address , upper column address , displayOff , displayOn , ... .
2. sendCommand(register To initilise with values given) . in oled it accepte to send all register and value in one transaction : {Multiplexer reg , value , dipslay colock , value , ... }

### 1.3 program a setPowerFunction() to privide display on off

1. To Enable the diplay Sendcommand(charg pump reg in oled(0x8D) , value to enable the charge pum in oled (0x14),display ON reg in oled(0xAF)

2. To disable the display SendCommand(display OFF reg,charg pump reg (0x8D) , value to disble charg pump (0x10)

### 1.4 dowload a fonts from the page of the address

1. fowload a Table contain all font Bytes for each Character

### 1.5 write a draw a text function

1. Send bytes list for each charchter of the text to the frame Table , that frame table contain the data to be sended to the display

### 1.6 write a draw character function

draw character means to write all bytes of the character to the frame Table , those bytes should be taken from the fonts bytes array of each character . to take a char c fonts index ,

```
c= (c - 32)*6
```

. than this charcter index should be send with drawByte function with

```
&fontsArray[c]
```

and size of the charcter depends on fonts form . in our downaloded fonts size of charcter = 6 . that mean 6 Byte should be readen from fonts array for each character.

### 1.7 write a draw bytes function

copie All Bytes of the character to the frame . that function should take the begin x , y postion

```
(frame[index])
```

. those x , y postion is to find in which pixel should that buffer showed in the display . the frame array should contaion a byte for each pixel that mean if the display 128x64 than frame

```
[128*64  \8]$
```

than each byte of the charcter Bytes should be sended with the draw byte function . that mean data pointer should be progressed each transaction +1 (data++) and x postion should be progessed too x++ to write each byte in one pixel . the y possition should stay constant to write in only one line . these procedure should be repeated while x ¡ width .

### 1.8 write a draw byte function

1. to draw a byte to the frame array we should know the position x , y to find the frame[index].

   ```
   index =  y / 8 * _width(128) + x$
   ```

   if the color of the character should be written in WHITE than :

   ```
   uint16 w = (uint16) byteToBeWritten << ( y % 8)
   frame[index] |= (w & 0xFF)
   ```