

```
#include <avr/interrupt.h>
#include <avr/io.h>
#include <avr/portpins.h>
#include <UTFT.h>
UTFTmyGLCD(SSD1289,38,39,40,41);
// Declare which fonts we will be using
//extern uint8_t BigFont[];
extern uint8_t SmallFont[];
const unsigned short Byte00[] PROGMEM = {
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000};
const unsigned short Byte01[] PROGMEM = {
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xFFFF};
const unsigned short Byte02[] PROGMEM = {
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xFFFF, 0x0000};
const unsigned short Byte03[] PROGMEM = {
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xFFFF, 0xFFFF};
const unsigned short Byte04[] PROGMEM = {
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xFFFF, 0x0000, 0x0000};
const unsigned short Byte05[] PROGMEM = {
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xFFFF, 0x0000, 0xFFFF};
const unsigned short Byte06[] PROGMEM = {
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xFFFF, 0xFFFF, 0x0000};
const unsigned short Byte07[] PROGMEM = {
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF};
const unsigned short Byte08[] PROGMEM = {
0x0000, 0x0000, 0x0000, 0x0000, 0xFFFF, 0x0000, 0x0000, 0x0000};
const unsigned short Byte09[] PROGMEM = {
0x0000, 0x0000, 0x0000, 0x0000, 0xFFFF, 0x0000, 0x0000, 0xFFFF};
const unsigned short Byte0A[] PROGMEM = {
0x0000, 0x0000, 0x0000, 0x0000, 0xFFFF, 0x0000, 0xFFFF, 0x0000};
const unsigned short Byte0B[] PROGMEM = {
0x0000, 0x0000, 0x0000, 0x0000, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF};
const unsigned short Byte0C[] PROGMEM = {
0x0000, 0x0000, 0x0000, 0x0000, 0xFFFF, 0xFFFF, 0x0000, 0x0000};
const unsigned short Byte0D[] PROGMEM = {
0x0000, 0x0000, 0x0000, 0x0000, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF};
const unsigned short Byte0E[] PROGMEM = {
0x0000, 0x0000, 0x0000, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000};
const unsigned short Byte0F[] PROGMEM = {
0x0000, 0x0000, 0x0000, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF};

const unsigned short Byte10[] PROGMEM = {
0x0000, 0x0000, 0x0000, 0xFFFF, 0x0000, 0x0000, 0x0000, 0x0000};
const unsigned short Byte11[] PROGMEM = {
0x0000, 0x0000, 0x0000, 0xFFFF, 0x0000, 0x0000, 0x0000, 0xFFFF};
const unsigned short Byte12[] PROGMEM = {
```





```
0x0000, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF};

const unsigned short Byte40[] PROGMEM = {
0x0000, 0xFFFF, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000};
const unsigned short Byte41[] PROGMEM = {
0x0000, 0xFFFF, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xFFFF};
const unsigned short Byte42[] PROGMEM = {
0x0000, 0xFFFF, 0x0000, 0x0000, 0x0000, 0x0000, 0xFFFF, 0x0000};
const unsigned short Byte43[] PROGMEM = {
0x0000, 0xFFFF, 0x0000, 0x0000, 0x0000, 0x0000, 0xFFFF, 0xFFFF};
const unsigned short Byte44[] PROGMEM = {
0x0000, 0xFFFF, 0x0000, 0x0000, 0x0000, 0xFFFF, 0x0000, 0x0000};
const unsigned short Byte45[] PROGMEM = {
0x0000, 0xFFFF, 0x0000, 0x0000, 0x0000, 0xFFFF, 0x0000, 0xFFFF};
const unsigned short Byte46[] PROGMEM = {
0x0000, 0xFFFF, 0x0000, 0x0000, 0x0000, 0xFFFF, 0xFFFF, 0x0000};
const unsigned short Byte47[] PROGMEM = {
0x0000, 0xFFFF, 0x0000, 0x0000, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF};
const unsigned short Byte48[] PROGMEM = {
0x0000, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0x0000, 0x0000, 0x0000};
const unsigned short Byte49[] PROGMEM = {
0x0000, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0x0000, 0x0000, 0xFFFF};
const unsigned short Byte4A[] PROGMEM = {
0x0000, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0x0000, 0xFFFF, 0x0000};
const unsigned short Byte4B[] PROGMEM = {
0x0000, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF};
const unsigned short Byte4C[] PROGMEM = {
0x0000, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0xFFFF, 0x0000, 0x0000};
const unsigned short Byte4D[] PROGMEM = {
0x0000, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF};
const unsigned short Byte4E[] PROGMEM = {
0x0000, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000};
const unsigned short Byte4F[] PROGMEM = {
0x0000, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF};

const unsigned short Byte50[] PROGMEM = {
0x0000, 0xFFFF, 0x0000, 0xFFFF, 0x0000, 0x0000, 0x0000, 0x0000};
const unsigned short Byte51[] PROGMEM = {
0x0000, 0xFFFF, 0x0000, 0xFFFF, 0x0000, 0x0000, 0x0000, 0xFFFF};
const unsigned short Byte52[] PROGMEM = {
0x0000, 0xFFFF, 0x0000, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0x0000};
const unsigned short Byte53[] PROGMEM = {
0x0000, 0xFFFF, 0x0000, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0xFFFF};
const unsigned short Byte54[] PROGMEM = {
0x0000, 0xFFFF, 0x0000, 0xFFFF, 0x0000, 0xFFFF, 0x0000, 0x0000};
const unsigned short Byte55[] PROGMEM = {
```



```
const unsigned short Byte6C[] PROGMEM = {
0x0000, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0x0000, 0x0000};
const unsigned short Byte6D[] PROGMEM = {
0x0000, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF};
const unsigned short Byte6E[] PROGMEM = {
0x0000, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000};
const unsigned short Byte6F[] PROGMEM = {
0x0000, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF};

const unsigned short Byte70[] PROGMEM = {
0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0x0000, 0x0000};
const unsigned short Byte71[] PROGMEM = {
0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0x0000, 0xFFFF};
const unsigned short Byte72[] PROGMEM = {
0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0x0000};
const unsigned short Byte73[] PROGMEM = {
0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0xFFFF};
const unsigned short Byte74[] PROGMEM = {
0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0x0000, 0x0000};
const unsigned short Byte75[] PROGMEM = {
0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0x0000, 0xFFFF};
const unsigned short Byte76[] PROGMEM = {
0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0x0000};
const unsigned short Byte77[] PROGMEM = {
0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF};
const unsigned short Byte78[] PROGMEM = {
0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0x0000};
const unsigned short Byte79[] PROGMEM = {
0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0xFFFF};
const unsigned short Byte7A[] PROGMEM = {
0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0x0000};
const unsigned short Byte7B[] PROGMEM = {
0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF};
const unsigned short Byte7C[] PROGMEM = {
0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0x0000};
const unsigned short Byte7D[] PROGMEM = {
0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF};
const unsigned short Byte7E[] PROGMEM = {
0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000};
const unsigned short Byte7F[] PROGMEM = {
0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF};

const unsigned short Byte80[] PROGMEM = {
0xFFFF, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000};
const unsigned short Byte81[] PROGMEM = {
0xFFFF, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xFFFF};
```







```
const unsigned short ByteAF[] PROGMEM = {
0xFFFF, 0x0000, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF};

const unsigned short ByteB0[] PROGMEM = {
0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0x0000, 0x0000};
const unsigned short ByteB1[] PROGMEM = {
0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0x0000, 0xFFFF};
const unsigned short ByteB2[] PROGMEM = {
0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0x0000};
const unsigned short ByteB3[] PROGMEM = {
0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0xFFFF};
const unsigned short ByteB4[] PROGMEM = {
0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0x0000, 0x0000};
const unsigned short ByteB5[] PROGMEM = {
0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0x0000, 0xFFFF};
const unsigned short ByteB6[] PROGMEM = {
0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0x0000};
const unsigned short ByteB7[] PROGMEM = {
0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF};
const unsigned short ByteB8[] PROGMEM = {
0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0x0000};
const unsigned short ByteB9[] PROGMEM = {
0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0xFFFF};
const unsigned short ByteBA[] PROGMEM = {
0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0x0000};
const unsigned short ByteBB[] PROGMEM = {
0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF};
const unsigned short ByteBC[] PROGMEM = {
0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0x0000};
const unsigned short ByteBD[] PROGMEM = {
0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF};
const unsigned short ByteBE[] PROGMEM = {
0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000};
const unsigned short ByteBF[] PROGMEM = {
0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF};

const unsigned short ByteC0[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000};
const unsigned short ByteC1[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xFFFF};
const unsigned short ByteC2[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0x0000, 0x0000, 0x0000, 0xFFFF, 0x0000};
const unsigned short ByteC3[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0x0000, 0x0000, 0x0000, 0xFFFF, 0xFFFF};
const unsigned short ByteC4[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0x0000, 0x0000, 0xFFFF, 0x0000, 0x0000};
```

```
const unsigned short ByteC5[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0x0000, 0x0000, 0xFFFF, 0x0000, 0xFFFF};
const unsigned short ByteC6[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0x0000, 0x0000, 0xFFFF, 0xFFFF, 0x0000};
const unsigned short ByteC7[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0x0000, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF};
const unsigned short ByteC8[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0x0000, 0x0000, 0x0000};
const unsigned short ByteC9[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0x0000, 0x0000, 0xFFFF};
const unsigned short ByteCA[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0x0000, 0xFFFF, 0x0000};
const unsigned short ByteCB[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF};
const unsigned short ByteCC[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0xFFFF, 0x0000, 0x0000};
const unsigned short ByteCD[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF};
const unsigned short ByteCE[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000};
const unsigned short ByteCF[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF};

const unsigned short ByteD0[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0x0000, 0x0000, 0x0000, 0x0000};
const unsigned short ByteD1[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0x0000, 0x0000, 0x0000, 0xFFFF};
const unsigned short ByteD2[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0x0000};
const unsigned short ByteD3[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0xFFFF};
const unsigned short ByteD4[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0x0000, 0xFFFF, 0x0000, 0x0000};
const unsigned short ByteD5[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0x0000, 0xFFFF, 0x0000, 0xFFFF};
const unsigned short ByteD6[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0x0000};
const unsigned short ByteD7[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF};
const unsigned short ByteD8[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0x0000};
const unsigned short ByteD9[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0xFFFF};
const unsigned short ByteDA[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0x0000};
const unsigned short ByteDB[] PROGMEM = {
```

```
0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF};
const unsigned short ByteDC[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0x0000};
const unsigned short ByteDD[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF};
const unsigned short ByteDE[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000};
const unsigned short ByteDF[] PROGMEM = {
0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF};

const unsigned short ByteE0[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000};
const unsigned short ByteE1[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0x0000, 0x0000, 0xFFFF};
const unsigned short ByteE2[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0x0000, 0xFFFF, 0x0000};
const unsigned short ByteE3[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0x0000, 0xFFFF, 0xFFFF};
const unsigned short ByteE4[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0x0000, 0x0000};
const unsigned short ByteE5[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0x0000, 0xFFFF};
const unsigned short ByteE6[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0xFFFF, 0x0000};
const unsigned short ByteE7[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF};
const unsigned short ByteE8[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0x0000, 0x0000, 0x0000};
const unsigned short ByteE9[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0x0000, 0x0000, 0xFFFF};
const unsigned short ByteEA[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0x0000, 0xFFFF, 0x0000};
const unsigned short ByteEB[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF};
const unsigned short ByteEC[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0x0000, 0x0000};
const unsigned short ByteED[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF};
const unsigned short ByteEE[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000};
const unsigned short ByteEF[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF};

const unsigned short ByteF0[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0x0000, 0x0000};
const unsigned short ByteF1[] PROGMEM = {
```

```

0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0x0000, 0xFFFF};
const unsigned short ByteF2[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0x0000};
const unsigned short ByteF3[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0xFFFF, 0xFFFF};
const unsigned short ByteF4[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0x0000, 0x0000};
const unsigned short ByteF5[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0x0000, 0xFFFF};
const unsigned short ByteF6[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0x0000};
const unsigned short ByteF7[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF, 0xFFFF};
const unsigned short ByteF8[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0x0000};
const unsigned short ByteF9[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0x0000, 0xFFFF};
const unsigned short ByteFA[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0x0000};
const unsigned short ByteFB[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF, 0xFFFF};
const unsigned short ByteFC[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0x0000};
const unsigned short ByteFD[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000, 0xFFFF};
const unsigned short ByteFE[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0x0000};
const unsigned short ByteFF[] PROGMEM = {
0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,};
const byteDataInPins[]={A8,A9,A10,A11,A12,A13,A14,A15};
const byte DISPONOFF_pin=14;
const byte TPB_pin=19;
const byte ACKfrom1802_pin=18;
const byte INT_NOT_pin=A5;
const byte EFX_NOT_pin=A6;
const byte DMAOUT_NOT_pin=A7;
volatile bool ACKfrom1802=0;
volatile bool gotDMAbyte=0;
volatile int DMAbytePtr=0;
volatile int DMAbyte[256];

void TurnACKfrom1802FlagON(){
  ACKfrom1802=1;
  //myGLCD.print("ACKfrom1802=1",CENTER,100);
}

```

```

void getDMAbyte(){
  PORTF=PORTF & 0x7F; //digitalWrite(DMAOUT_NOT_pin,LOW);
  DMAbyte[DMAbytePtr]=PINK;
  gotDMAbyte=1;
}

void setup() {
  pinMode(INT_NOT_pin, OUTPUT);
  pinMode(DMAOUT_NOT_pin, OUTPUT);
  pinMode(EFX_NOT_pin, OUTPUT);
  digitalWrite(INT_NOT_pin,HIGH);
  digitalWrite(DMAOUT_NOT_pin,LOW); //NEEDS TO BE LOW
  digitalWrite(EFX_NOT_pin,HIGH);
  pinMode(7,OUTPUT);
  digitalWrite(7,HIGH);
  myGLCD.InitLCD();
  myGLCD.setFont(SmallFont);
  for (int i=0;i<8;i++){
    pinMode(DataInPins[i], INPUT_PULLUP);
  }
  attachInterrupt(digitalPinToInterrupt(TPB_pin),getDMAbyte,RISING);
  pinMode(ACKfrom1802_pin, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(ACKfrom1802_pin),TurnACKfrom1802FlagON,
  RISING);
  pinMode(TPB_pin, INPUT_PULLUP);
}

unsigned int xPos;
byte yPos;

void loop() {
  digitalWrite(DMAOUT_NOT_pin,LOW);
  digitalWrite(INT_NOT_pin,HIGH);
  digitalWrite(EFX_NOT_pin,HIGH);
  bool eraseSCR=0;
  while (!digitalRead(DISPONOFF_pin)){
    int Xpos=random(200);
    int Ypos=random(200);
    myGLCD.print(" |-----| ",Xpos,Ypos);
    myGLCD.print(" | Turn Display ON | ",Xpos,Ypos+10);
    myGLCD.print(" |-----| ",Xpos,Ypos+20);
    delay(1000);
    myGLCD.clrScr();
    myGLCD.setColor(VGA_WHITE);
    eraseSCR=1;
    for (int i=0;i<256;i++){

```

```

DMAbyte[i]=0;
}
}
if (eraseSCR){
    myGLCD.clrScr();
}
xPos=0;
yPos=0;
ACKfrom1802=0;
DMAbytePtr=0;
digitalWrite(INT_NOT_pin,LOW);
//Wait for ACK signal from 1802 interrupt routine
while (!ACKfrom1802);
digitalWrite(INT_NOT_pin,HIGH);
//Load 256 DMA bytes
while (DMAbytePtr<256){
    //Start clock to initiate DMAOUT pulse
    gotDMAbyte=0;
    PORTF=PORTF | 0x80; //digitalWrite(DMAOUT_NOT_pin,HIGH)
    while (!gotDMAbyte); //Wait for 74273 DMAOUT latch pulse
    myGLCD.setColor(VGA_WHITE);
    switch (DMAbyte[DMAbytePtr]){
        case 0x00:
            myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte00, 5);
            break;
        case 0x01:
            myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte01, 5);
            break;
        case 0x02:
            myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte02, 5);
            break;
        case 0x03:
            myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte03, 5);
            break;
        case 0x04:
            myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte04, 5);
            break;
        case 0x05:
            myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte05, 5);
            break;
        case 0x06:
            myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte06, 5);
            break;
        case 0x07:
            myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte07, 5);
            break;
    }
}
}
}

```

```
case 0x08:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte08, 5);
    break;
case 0x09:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte09, 5);
    break;
case 0x0A:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte0A, 5);
    break;
case 0x0B:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte0B, 5);
    break;
case 0x0C:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte0C, 5);
    break;
case 0x0D:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte0D, 5);
    break;
case 0x0E:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte0E, 5);
    break;
case 0x0F:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte0F, 5);
    break;
case 0x10:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte10, 5);
    break;
case 0x11:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte11, 5);
    break;
case 0x12:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte12, 5);
    break;
case 0x13:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte13, 5);
    break;
case 0x14:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte14, 5);
    break;
case 0x15:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte15, 5);
    break;
case 0x16:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte16, 5);
    break;
case 0x17:
```

```
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte17, 5);
    break;
case 0x18:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte18, 5);
    break;
case 0x19:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte19, 5);
    break;
case 0x1A:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte1A, 5);
    break;
case 0x1B:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte1B, 5);
    break;
case 0x1C:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte1C, 5);
    break;
case 0x1D:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte1D, 5);
    break;
case 0x1E:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte1E, 5);
    break;
case 0x1F:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte1F, 5);
    break;
case 0x20:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte20, 5);
    break;
case 0x21:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte21, 5);
    break;
case 0x22:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte22, 5);
    break;
case 0x23:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte23, 5);
    break;
case 0x24:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte24, 5);
    break;
case 0x25:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte25, 5);
    break;
case 0x26:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte26, 5);
```



```
    break;
case 0x27:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte27, 5);
    break;
case 0x28:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte28, 5);
    break;
case 0x29:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte29, 5);
    break;
case 0x2A:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte2A, 5);
    break;
case 0x2B:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte2B, 5);
    break;
case 0x2C:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte2C, 5);
    break;
case 0x2D:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte2D, 5);
    break;
case 0x2E:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte2E, 5);
    break;
case 0x2F:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte2F, 5);
    break;
case 0x30:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte30, 5);
    break;
case 0x31:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte31, 5);
    break;
case 0x32:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte32, 5);
    break;
case 0x33:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte33, 5);
    break;
case 0x34:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte34, 5);
    break;
case 0x35:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte35, 5);
    break;
```

```
case 0x36:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte36, 5);
    break;
case 0x37:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte37, 5);
    break;
case 0x38:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte38, 5);
    break;
case 0x39:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte39, 5);
    break;
case 0x3A:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte3A, 5);
    break;
case 0x3B:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte3B, 5);
    break;
case 0x3C:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte3C, 5);
    break;
case 0x3D:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte3D, 5);
    break;
case 0x3E:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte3E, 5);
    break;
case 0x3F:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte3F, 5);
    break;
case 0x40:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte40, 5);
    break;
case 0x41:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte41, 5);
    break;
case 0x42:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte42, 5);
    break;
case 0x43:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte43, 5);
    break;
case 0x44:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte44, 5);
    break;
case 0x45:
```

```
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte45, 5);
    break;
case 0x46:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte46, 5);
    break;
case 0x47:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte47, 5);
    break;
case 0x48:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte48, 5);
    break;
case 0x49:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte49, 5);
    break;
case 0x4A:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte4A, 5);
    break;
case 0x4B:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte4B, 5);
    break;
case 0x4C:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte4C, 5);
    break;
case 0x4D:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte4D, 5);
    break;
case 0x4E:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte4E, 5);
    break;
case 0x4F:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte4F, 5);
    break;
case 0x50:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte50, 5);
    break;
case 0x51:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte51, 5);
    break;
case 0x52:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte52, 5);
    break;
case 0x53:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte53, 5);
    break;
case 0x54:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte54, 5);
```

```
    break;
case 0x55:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte55, 5);
    break;
case 0x56:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte56, 5);
    break;
case 0x57:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte57, 5);
    break;
case 0x58:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte58, 5);
    break;
case 0x59:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte59, 5);
    break;
case 0x5A:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte5A, 5);
    break;
case 0x5B:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte5B, 5);
    break;
case 0x5C:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte5C, 5);
    break;
case 0x5D:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte5D, 5);
    break;
case 0x5E:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte5E, 5);
    break;
case 0x5F:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte5F, 5);
    break;
case 0x60:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte60, 5);
    break;
case 0x61:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte61, 5);
    break;
case 0x62:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte62, 5);
    break;
case 0x63:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte63, 5);
    break;
```

```
case 0x64:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte64, 5);
    break;
case 0x65:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte65, 5);
    break;
case 0x66:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte66, 5);
    break;
case 0x67:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte67, 5);
    break;
case 0x68:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte68, 5);
    break;
case 0x69:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte69, 5);
    break;
case 0x6A:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte6A, 5);
    break;
case 0x6B:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte6B, 5);
    break;
case 0x6C:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte6C, 5);
    break;
case 0x6D:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte6D, 5);
    break;
case 0x6E:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte6E, 5);
    break;
case 0x6F:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte6F, 5);
    break;
case 0x70:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte70, 5);
    break;
case 0x71:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte71, 5);
    break;
case 0x72:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte72, 5);
    break;
case 0x73:
```

```
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte73, 5);
    break;
case 0x74:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte74, 5);
    break;
case 0x75:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte75, 5);
    break;
case 0x76:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte76, 5);
    break;
case 0x77:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte77, 5);
    break;
case 0x78:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte78, 5);
    break;
case 0x79:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte79, 5);
    break;
case 0x7A:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte7A, 5);
    break;
case 0x7B:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte7B, 5);
    break;
case 0x7C:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte7C, 5);
    break;
case 0x7D:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte7D, 5);
    break;
case 0x7E:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte7E, 5);
    break;
case 0x7F:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte7F, 5);
    break;
case 0x80:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte80, 5);
    break;
case 0x81:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte81, 5);
    break;
case 0x82:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte82, 5);
```

```
    break;
case 0x83:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte83, 5);
    break;
case 0x84:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte84, 5);
    break;
case 0x85:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte85, 5);
    break;
case 0x86:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte86, 5);
    break;
case 0x87:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte87, 5);
    break;
case 0x88:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte88, 5);
    break;
case 0x89:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte89, 5);
    break;
case 0x8A:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte8A, 5);
    break;
case 0x8B:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte8B, 5);
    break;
case 0x8C:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte8C, 5);
    break;
case 0x8D:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte8D, 5);
    break;
case 0x8E:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte8E, 5);
    break;
case 0x8F:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte8F, 5);
    break;
case 0x90:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte90, 5);
    break;
case 0x91:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte91, 5);
    break;
```

```
case 0x92:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte92, 5);
    break;
case 0x93:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte93, 5);
    break;
case 0x94:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte94, 5);
    break;
case 0x95:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte95, 5);
    break;
case 0x96:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte96, 5);
    break;
case 0x97:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte97, 5);
    break;
case 0x98:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte98, 5);
    break;
case 0x99:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte99, 5);
    break;
case 0x9A:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte9A, 5);
    break;
case 0x9B:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte9B, 5);
    break;
case 0x9C:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte9C, 5);
    break;
case 0x9D:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte9D, 5);
    break;
case 0x9E:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte9E, 5);
    break;
case 0x9F:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, Byte9F, 5);
    break;
case 0xA0:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteA0, 5);
    break;
case 0xA1:
```



```
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteA1, 5);
    break;
case 0xA2:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteA2, 5);
    break;
case 0xA3:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteA3, 5);
    break;
case 0xA4:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteA4, 5);
    break;
case 0xA5:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteA5, 5);
    break;
case 0xA6:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteA6, 5);
    break;
case 0xA7:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteA7, 5);
    break;
case 0xA8:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteA8, 5);
    break;
case 0xA9:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteA9, 5);
    break;
case 0xAA:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteAA, 5);
    break;
case 0xAB:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteAB, 5);
    break;
case 0xAC:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteAC, 5);
    break;
case 0xAD:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteAD, 5);
    break;
case 0xAE:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteAE, 5);
    break;
case 0xAF:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteAF, 5);
    break;
case 0xB0:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteB0, 5);
```

```
break;
case 0xB1:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteB1, 5);
    break;
case 0xB2:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteB2, 5);
    break;
case 0xB3:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteB3, 5);
    break;
case 0xB4:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteB4, 5);
    break;
case 0xB5:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteB5, 5);
    break;
case 0xB6:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteB6, 5);
    break;
case 0xB7:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteB7, 5);
    break;
case 0xB8:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteB8, 5);
    break;
case 0xB9:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteB9, 5);
    break;
case 0xBA:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteBA, 5);
    break;
case 0xBB:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteBB, 5);
    break;
case 0xBC:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteBC, 5);
    break;
case 0xBD:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteBD, 5);
    break;
case 0xBE:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteBE, 5);
    break;
case 0xBF:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteBF, 5);
    break;
```

```
case 0xC0:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteC0, 5);
    break;
case 0xC1:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteC1, 5);
    break;
case 0xC2:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteC2, 5);
    break;
case 0xC3:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteC3, 5);
    break;
case 0xC4:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteC4, 5);
    break;
case 0xC5:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteC5, 5);
    break;
case 0xC6:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteC6, 5);
    break;
case 0xC7:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteC7, 5);
    break;
case 0xC8:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteC8, 5);
    break;
case 0xC9:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteC9, 5);
    break;
case 0xCA:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteCA, 5);
    break;
case 0xCB:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteCB, 5);
    break;
case 0xCC:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteCC, 5);
    break;
case 0xCD:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteCD, 5);
    break;
case 0xCE:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteCE, 5);
    break;
case 0xCF:
```

```
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteCF, 5);
    break;
case 0xD0:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteD0, 5);
    break;
case 0xD1:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteD1, 5);
    break;
case 0xD2:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteD2, 5);
    break;
case 0xD3:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteD3, 5);
    break;
case 0xD4:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteD4, 5);
    break;
case 0xD5:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteD5, 5);
    break;
case 0xD6:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteD6, 5);
    break;
case 0xD7:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteD7, 5);
    break;
case 0xD8:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteD8, 5);
    break;
case 0xD9:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteD9, 5);
    break;
case 0xDA:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteDA, 5);
    break;
case 0xDB:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteDB, 5);
    break;
case 0xDC:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteDC, 5);
    break;
case 0xDD:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteDD, 5);
    break;
case 0xDE:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteDE, 5);
```

```
break;
case 0xDF:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteDF, 5);
    break;
case 0xE0:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteE0, 5);
    break;
case 0xE1:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteE1, 5);
    break;
case 0xE2:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteE2, 5);
    break;
case 0xE3:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteE3, 5);
    break;
case 0xE4:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteE4, 5);
    break;
case 0xE5:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteE5, 5);
    break;
case 0xE6:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteE6, 5);
    break;
case 0xE7:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteE7, 5);
    break;
case 0xE8:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteE8, 5);
    break;
case 0xE9:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteE9, 5);
    break;
case 0xEA:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteEA, 5);
    break;
case 0xEB:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteEB, 5);
    break;
case 0xEC:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteEC, 5);
    break;
case 0xED:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteED, 5);
    break;
```

```
case 0xEE:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteEE, 5);
    break;
case 0xEF:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteEF, 5);
    break;
case 0xF0:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteF0, 5);
    break;
case 0xF1:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteF1, 5);
    break;
case 0xF2:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteF2, 5);
    break;
case 0xF3:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteF3, 5);
    break;
case 0xF4:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteF4, 5);
    break;
case 0xF5:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteF5, 5);
    break;
case 0xF6:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteF6, 5);
    break;
case 0xF7:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteF7, 5);
    break;
case 0xF8:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteF8, 5);
    break;
case 0xF9:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteF9, 5);
    break;
case 0xFA:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteFA, 5);
    break;
case 0xFB:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteFB, 5);
    break;
case 0xFC:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteFC, 5);
    break;
case 0xFD:
```

```
        myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteFD, 5);
    break;
case 0xFE:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteFE, 5);
    break;
default:
    myGLCD.drawBitmap (xPos, yPos, 8, 1, ByteFF, 5);
    break;
}
DMAbytePtr+=1;
xPos+=40;
if (xPos==320){
    xPos=0;
    yPos+=5;
}
}
//Signal 1802 done loading 256 DMA bytes
ACKfrom1802=0;
digitalWrite(EFX_NOT_pin,LOW);
//Wait for ACK signal from 1802 interrupt routine
while (!ACKfrom1802);
digitalWrite(EFX_NOT_pin,HIGH);
delay(20);
}
```