

```

#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 byte DataInPins[]={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);
    }
}

```

```

    }
    pinMode(TPB_pin, INPUT_PULLUP);
attachInterrupt(digitalPinToInterrupt(TPB_pin),getDMAbyte,RISING);
    pinMode(ACKfrom1802_pin, INPUT_PULLUP);
attachInterrupt(digitalPinToInterrupt(ACKfrom1802_pin),TurnACKfrom1802FlagON,
RISING);
}

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.clrScr();
        myGLCD.setColor(VGA_WHITE);
        myGLCD.print(" |-----| ",Xpos,Ypos);
        myGLCD.print(" | Turn Display ON | ",Xpos,Ypos+10);
        myGLCD.print(" |-----| ",Xpos,Ypos+20);
        delay(1000);
        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
    ShowBit(DMAbyte[DMAbytePtr] & 0x80);
    xPos+=5;
    ShowBit(DMAbyte[DMAbytePtr] & 0x40);
    xPos+=5;
    ShowBit(DMAbyte[DMAbytePtr] & 0x20);
    xPos+=5;
    ShowBit(DMAbyte[DMAbytePtr] & 0x10);
    xPos+=5;
    ShowBit(DMAbyte[DMAbytePtr] & 0x08);
    xPos+=5;
    ShowBit(DMAbyte[DMAbytePtr] & 0x04);
    xPos+=5;
    ShowBit(DMAbyte[DMAbytePtr] & 0x02);
    xPos+=5;
    ShowBit(DMAbyte[DMAbytePtr] & 0x01);
    xPos+=5;
    DMAbytePtr++;
    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);
}

void ShowBit(int ColorBit){
    if (ColorBit){
        myGLCD.setColor(VGA_WHITE);
    }else{
        myGLCD.setColor(VGA_BLACK);
    }
    myGLCD.fillRect(xPos,yPos,xPos+4,yPos+4);
}

```