[Programa] TFT

Objetivo:

Utilizar o *ESP32-WROOM-32E DevKitC v4* [2] para desenhar num monitor TFT.

Lista de material:

- *ESP32-WROOM-32E DevKitC v4*
- Breadboards
- *Adafruit 1.44" Color TFT (ST7735)*
- Fios de ligação
- Cabo USB micro

Contextualização:

Adafruit 1.44" Color TFT

O monitor TFT (Thin-film transistor, [6]) de 1.44 polegadas da Adafruit (*ST7735 Adafruit 1.44" Color TFT*, [1]) faz uso do barramento de comunicação SPI (*Serial Peripheral Interface* [5]) de 4 fios possuindo ainda *buffer* de dados endereçável por pixel dedicado.

7_tft_eg.png

Serial Peripheral Interface

O *Serial Peripheral Interface* (SPI) é um barramento que permite, por exemplo, a comunicação entre um microcontrolador com diversos outros componentes [5, 3].

Ainda que a sua designação possa variar, e nem todas as ligações se tenham que usar num dado exemplo, o barramento SPI considera 4 sinais lógicos:

SCLK/SCK	*Serial Clock*
MOSI/SDI	*Main Out Sub In / Serial Data In*
MISO/SDO	*Main In Sub Out / Serial Data Out*
CS/SS	*Chip Select*

7_spi_esp32.png	7_spi_tft.png

7_spi_eg.png

Procedimento:

Escolhendo os terminais 13, 14, 15, 25, 26, GND e 3.3V do ESP32 como referência deve considerar-se o seguinte ilustração de montagem:

7_montagem_overview.png

Instruções:

(para uma primeira utilização do ESP32, sugere-se o tutorial *Primeiro Programa* [4])

- Montar o circuito esquematizado anteriormente (ver Secção **Montagem exemplificativa** , para mais detalhe)
- Ligar a placa ESP32 ao computador por via de cabo USB micro
- Abrir o IDE Arduino
- Usando a ferramenta de gestão de bibliotecas do IDE Arduino procurar e instalar a biblioteca: ***Adafruit_GFX*** [7,8].

7_arduino_aa.png

- Repetir a tarefa anterior para as bibliotecas ***Adafruit_ST7735*** e ***Adafruit_BusIO***;

7_arduino_bb.png	7_arduino_cc.png

- Selecionar o dispositivo ESP32 adequado
- Copiar o seguinte código

```
/*
* TFT
*/
#include <SPI.h>□□□ // Biblioteca relacionada com SPI
#include <Adafruit_GFX.h>    // Biblioteca relacionada com primitivas de desenho
#include <Adafruit_ST7735.h> // Biblioteca específica para ST7735
□
// definição dos terminais a considerar
#define TFT_CS         15
#define TFT_DC         25
#define TFT_MOSI       13
#define TFT_SCLK       14
#define TFT_RST        26
```

```
// definição de cores de interesse
#define BLUE     0x001F
#define GREEN    0x07E0
#define WHITE    0xFFFF
#define BLACK    0x0000

// variáveis várias
int m_pos = 0;
int m_s = 128;
int m_contador = 0;
int m_o_contador = 65;
char cc = m_o_contador;

// inicialização de classe relacionada com ST7735
Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS, TFT_DC, TFT_MOSI, TFT_SCLK, TFT_RST);

// função de escrita
void testdrawtext(char *text, uint16_t color, uint8_t size) {
  tft.setCursor(51, 49);
  tft.setTextColor(color);
  tft.setTextWrap(true);
  tft.setTextSize(size);
  tft.print(text);
}

// a função setup é executada pontualmente quando o botão reset é premido ou
// a placa é alimentada:
void setup(void) {
  Serial.begin(115200);
  Serial.print(F("inicialização ..."));

  tft.initR(INITR_144GREENTAB); // inicialização específica do tft 128x128 1.44" usado
  tft.setRotation(1); // definição de rotação do monitor
  tft.fillScreen(WHITE); // desenho no ecrã na sua globalidade

  cc = m_o_contador;
  testdrawtext(&cc,BLACK,5); // escrita no ecrã

  delay(500);
```

```
Serial.print(F(" completa"));
}

// a função loop é executada sucessivamente, uma e outra vez, e por aí em diante
void loop() {
Serial.print(F("iteração"));
Serial.print(m_contador, DEC);
m_contador = m_contador + 1;

tft.drawRect(m_pos,m_pos,m_s,m_s,BLUE); // desenho de linha de quadrado azul
delay(10);

m_pos = m_pos + 1;
m_s = m_s - 2;
if (m_pos >= 64){
m_pos = 0;
m_s = 128;

tft.fillScreen(WHITE);
m_o_contador = m_o_contador + 1;
if (m_o_contador > 90) {
m_o_contador = 65;
}
cc = m_o_contador;

testdrawtext(&cc,BLACK,5); // escrita de texto no ecrâ
delay(500);
}

tft.drawRect(m_pos,m_pos,m_s,m_s, GREEN);
delay(10);
}
```

- Compilar e carregar o código para a placa ESP32

7_arduino.png

- Verificar o resultado