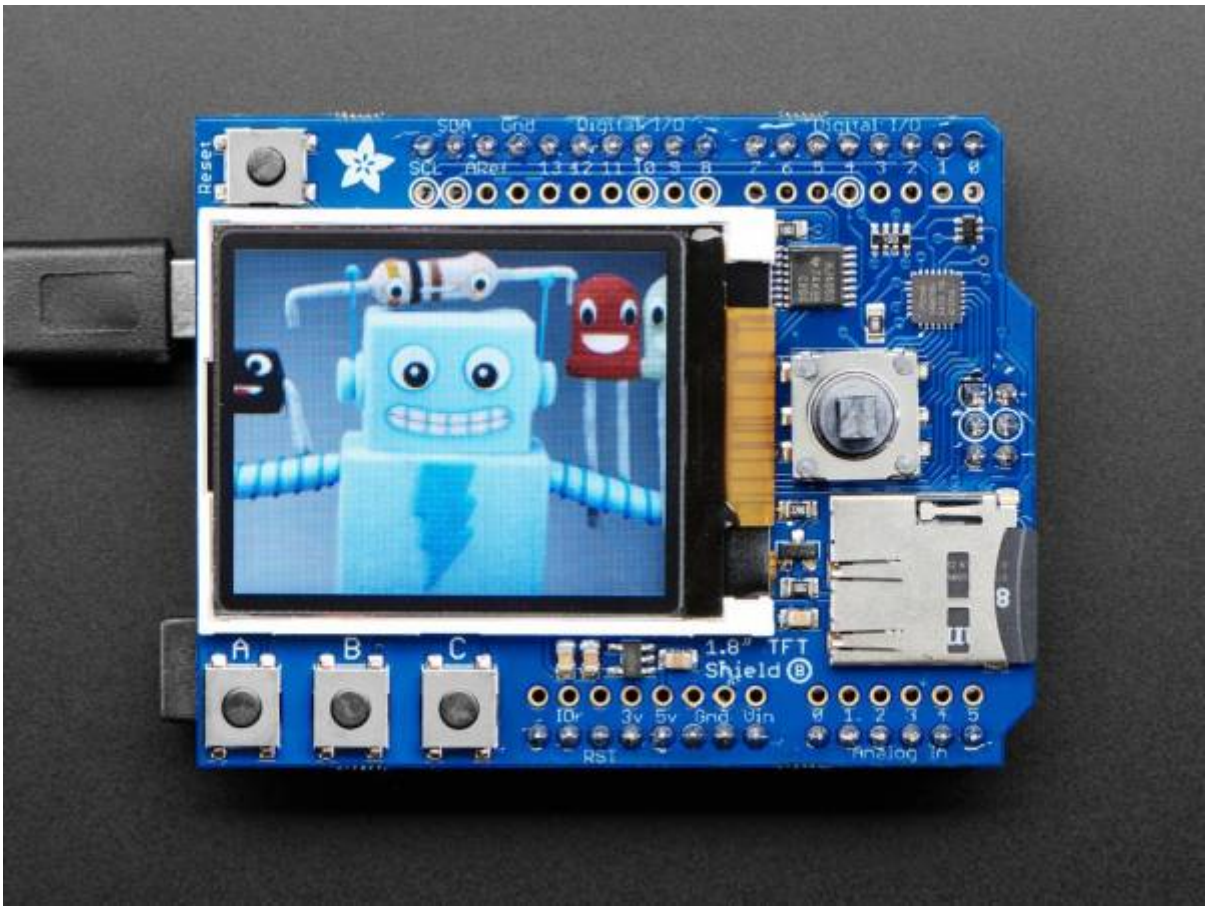
Adafruit 1.8" 128x160 Color TFT LCD display with MicroSD Card v2 - ST7735R (SPI)

[Mise à jour le 24/12/2021]

- **Ressources**
  - **Adafruit**
    - Produit Adafruit 1.8" Color TFT Shield w/microSD and Joystick - v 2.
    - Tutoriel : Testing the Shield
- **Lectures connexes**
  - Les afficheurs graphiques : généralités et primitives
  - Bibliothèque - Adafruit GFX Graphics Library
  - 0,96" 128x64 OLED 2864 Display module - SSD1306 (I2C)
  - Adafruit 1,3" 128x64 OLED FeatherWing - SH1107 + 3 buttons (I2C)
- **Distributeur**
  - **Gotronic** : Shield écran couleur TFT 1,8'' ADA802



1. Description

Cette carte contient un écran **1.8“ TFT**, un connecteur pour carte **microSD**, un **joystick** à **5 directions** et **trois boutons**. Comme l'**écran** n'utilise que 4 broches pour communiquer en **SPI** et possède son propre buffer, il peut facilement être utilisé pour ajouter un écran et une interface à un montage sans limiter la mémoire ou les broches du microcontrôleur.

Ce shield est compatible avec le format 'Arduino R3'. Il est utilisable avec tous les Arduinos ou Metros, y compris les Metros M0 ou M4, Arduino Mega, Zero, etc. La gestion des **boutons** et du **joystick** se fait à l’aide du port **I2C**, de sorte que seules 2 broches sont nécessaires pour communiquer avec les 8 commutateurs.

L’écran couleur 1.8” a **128×160 px**. Le pilote TFT (ST7735R) peut afficher des couleurs sur **18 bits** (**262144 nuances** !).

Il est possible d’afficher des images **bitmaps** couleur à partir d’une carte microSD formatée (**FAT16** / **FAT32**).

Pour afficher du texte, des formes, des lignes, des pixels, etc., le shield utilise les broches **SPI** (**SCK** / **MOSI** / **MISO**), **I2C** (**SDA** & **SCL**) et numérique **#8**. Pour la carte microSD, il a également besoin de **#4**.

- **Caractéristiques**
  - 1.8“ diagonal LCD TFT display
  - Physical dimensions: 2.71” (69mm) width, 2.1“ (53.5mm) height, 0.27” (6.94mm) thickness (top of joystick)
  - 128×160 resolution, 18-bit (262,144) color
  - 4 wire SPI digital interface
  - Built-in microSD slot - uses 2 more digital lines
  - 5V compatible! Use with 3.3V or 5V logic Arduinos
  - Onboard 3.3V @ 150mA LDO regulator
  - 2 white LED backlight, transistor connected. PWM controlled via I2C seesaw chip
  - Comes with header, requires soldering!
  - Display current draw is mostly based on the backlight, with full backlight the current draw is ~100mA, this does not include the SD Card. SD cards can draw 20-100mA based on read/write. Measure current draw in circuit to get precise numbers.
- **Connexions**
  - SCK - Horloge SPI
  - MOSI - SPI Data
  - Digital 10 - Sélection de puce
  - Digital 8 - Sélection de données / commandes
- **Bibliothèques à installer dans l'IDE Arduino ou dans PlatformIO (VSCode)**
  - Adafruit seesaw library
  - Adafruit gfx library

- Adafruit ST7735 and ST7789 library
- SD Built-In

2. Exemple

- *Un premier exemple pour tester le shield*
  Arduino Examples → Examples from Custom Libraries → Adafruit_ST7735_and_ST7789_Library → **seesaw_shield18_test.ino**

initv2.cpp

```
// Programme : seesawshield18_test
// Carte : Arduino Uno
// Shield écran couleur TFT 1,8'' V2
// Fichier : seesawshield18_test.ino (partiel)

#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Adafruit_ST7735.h>
// Assurez-vous d'avoir installé la bibliothèque Adafruit seesaw!
#include <Adafruit_seesaw.h>
#include <Adafruit_TFTShield18.h>

Adafruit_TFTShield18 ss; // Constructeur

// L'écran TFT et la carte SD partagent l'interface SPI.
// Pour la carte Arduino, le bus SPI est disponible sur
// pin 11 = MOSI, pin 12 = MISO, pin 13 = SCK.
#define SD_CS 4    // Sélection de la carte SD sur le Shield V2
#define TFT_CS 10  // Sélection de l'afficheur TFT sur le Shield V2
#define TFT_DC 8   // Données/commandes line de l'afficheur TFT sur le
Shield V2
#define TFT_RST -1 // Le reset de l'afficheur TFT est géré par seesaw !

Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS, TFT_DC, TFT_RST); //
Constructeur

void setup()
{
    Serial.begin(9600);
    while (!Serial)
        ;

    // On désactive l'afficheur et la carte SD
    pinMode(TFT_CS, OUTPUT);
    digitalWrite(TFT_CS, HIGH);
    pinMode(SD_CS, OUTPUT);
    digitalWrite(SD_CS, HIGH);

    // Démarrage de la puce Seesaw
```

```
    if (!ss.begin())
    {
        Serial.println("seesaw ne peut pas être initialisé!");
        while (1)
            ;
    }

    Serial.println("seesaw démarré");
    Serial.print("Version: ");
    Serial.println(ss.getVersion(), HEX);

    // Désactivation du rétroéclairage
    ss.setBacklight(TFTSHIELD_BACKLIGHT_OFF);
    // Reset de l'afficheur TFT
    ss.tftReset();

    // Initialisation de l'afficheur 1.8" TFT
    tft.initR(INITR_BLACKTAB); // Initialisation de la puce ST77355,
fond d'écran  noir

    Serial.println("TFT OK!");
    tft.fillScreen(ST77XX_CYAN); // écran cyan

    // Activation du rétroéclairage (100%)
    // ss.setBacklight(TFTSHIELD_BACKLIGHT_ON);
    // ou 30%
    // ss.setBacklight(TFTSHIELD_BACKLIGHT_ON / 3);
    // ou éclairage progressif
    for (int32_t i = TFTSHIELD_BACKLIGHT_OFF; i <
TFTSHIELD_BACKLIGHT_ON; i += 100)
    {
        ss.setBacklight(i);
        delay(1);
    }
    delay(100);
    tft.fillScreen(ST77XX_RED); // fond d'écran rouge
    delay(100);
    tft.fillScreen(ST77XX_GREEN); // fond d'écran vert
    delay(100);
    tft.fillScreen(ST77XX_BLUE); // fond d'écran bleu
    delay(100);
    tft.fillScreen(ST77XX_BLACK); // fond d'écran noir

    tft.setTextSize(1);
    tft.setTextColor(ST77XX_WHITE); // texte blanc
    tft.setCursor(0, 0);
    tft.print("Presser les boutons");
}

uint8_t buttonhistory = 0;
```

```
void loop()
{
    uint32_t buttons = ss.readButtons();
  tft.setTextSize(3);
  if(! (buttons & TFTSHIELD_BUTTON_DOWN)){
    tft.setTextColor(ST77XX_RED);
    tft.setCursor(0, 10);
    tft.print("Bas ");
    buttonhistory |= 1;
  }
  if(! (buttons & TFTSHIELD_BUTTON_LEFT)){
    tft.setTextColor(ST77XX_YELLOW);
    tft.setCursor(0, 35);
     tft.print("Gauche ");
    buttonhistory |= 2;
  }
  if(! (buttons & TFTSHIELD_BUTTON_UP)){
    tft.setTextColor(ST77XX_GREEN);
    tft.setCursor(0, 60);
    tft.print("Haut");
    buttonhistory |= 4;
  }
  if(! (buttons & TFTSHIELD_BUTTON_RIGHT)){
    tft.setTextColor(ST77XX_BLUE);
    tft.setCursor(0, 85);
    tft.print("Droit");
    buttonhistory |= 8;
  }
  if(! (buttons & TFTSHIELD_BUTTON_1)){
    tft.setTextColor(ST77XX_BLUE);
    tft.setCursor(0, 140);
    tft.print("1");
    buttonhistory |= 16;
  }
  if(! (buttons & TFTSHIELD_BUTTON_2)){
    tft.setTextColor(ST77XX_GREEN);
    tft.setCursor(50, 140);
    tft.print("2");
    buttonhistory |= 32;
  }
  if(! (buttons & TFTSHIELD_BUTTON_3)){
    tft.setTextColor(ST77XX_YELLOW);
    tft.setCursor(100, 140);
    tft.print("3");
    buttonhistory |= 64;
  }
  if (! (buttons & TFTSHIELD_BUTTON_IN)) {
    tft.setTextColor(ST77XX_MAGENTA);
    tft.setCursor(0, 110);
    tft.print("SELECT");
  }
```

```
}
```

TELECHARGER

Code complet avec la gestion de la carte SD
Arduino Examples → Examples from Custom Libraries → Adafruit_ST7735_and_ST7789_Library → **seesaw_shield18_test.ino**

2. Breakout ou Shield V1 (ARCHIVE)

- **Caractéristiques**
  - Alimentation: 3,3 ou 5 Vcc
  - Interface SPI
  - Résolution: 160 x 128 pixels
  - Couleurs: 18 bits (262144 couleurs)
  - Rétro-éclairage à leds
  - Port micro-SD (carte non incluse)
  - Dimensions: 70 x 54 x 12 mm

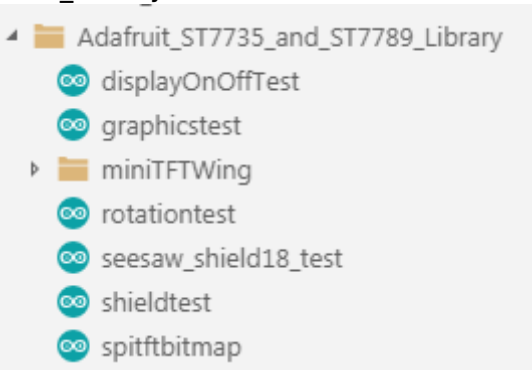
- **Brochage**
  - Compatible Arduino Uno R3.

Désignation	Description	Câblage Uno
MISO	**SPI Master In Slave Out**. Utilisée pour la carte SD mais pas pour l'écran TFT qui est en écriture seule	**MISO(D12)**
MOSI	**SPI Master Out Slave In**. Utilisée pour envoyer des données du microcontrôleur à la carte SD et / ou au TFT	**MOSI(D11)**
SCLK	**SPI** : horloge	**SCK(D13)**
TCS	**SPI** : sélection de la puce TFT	**D10**
DC	**Sélection** : données ou commande pour l'afficheur	**D8**
CCS	**Sélection**. Utilisée pour lire la carte SD	**D4**
RST	**Réinitialisation** de l'afficheur	**D9**

- **Bibliothèques à installer dans l'IDE Arduino**

  - Adafruit gfx library
  - Adafruit ST7735 and ST7789 library
  - TFT Built-In
  - Adafruit ImageReader Library

- **Exemples**
  - *Un premier exemple pour tester le shield*
    Arduino Examples → Examples from Custom Libraries → Adafruit_ST7735_and_ST7789_Library → **shieldtest.ino**



  - *Un autre exemple pour afficher une image **bitmap** située sur la carte SD*
    Arduino Examples → Examples from Custom Libraries → Adafruit_ImageReader_Library → **ShieldST7735.ino**

- **Initialisation**

init.cpp

```
// Programme : DemoST7735
// Carte : Arduino Uno
// Shield écran couleur TFT 1,8'' V1 ou breakout
// Fichier : ST7735.ino (partiel)

// Bibliothèques
#include <Adafruit_GFX.h> // Bibliothèques graphiques
#include <Adafruit_SPITFT_Macros.h>
#include <Adafruit_SPITFT.h>
#include <gfxfont.h>
#include <Adafruit_ST7735.h> // Bibliothèques spécifiques aux matériels
#include <Adafruit_ST7789.h>
#include <Adafruit_ST77xx.h>
#include <SPI.h> // Communication sur un Bus SPI

// Configuration des broches du micro pour commander un afficheur à
ST7735
// Pour la sélection du ST7735, vous pouvez utiliser 2 ou 3 broches
// Ces broches fonctionneront également pour le shield TFT de 1,8 "
#define TFT_CS     10 // Cheap Select (Sélection du composant)
#define TFT_RST    9  // Peut être connectéé à la broche de remise à
zéro (RST) de l'Arduino
                      // dans ce cas, définissez cette broche #define
sur 0!
#define TFT_DC     8  // Sélection d'une Donnée / Commande
```

```
// Option 1 (recommandée): Utiliser l'interface SPI matérielle
// (pour une UNO  SPI CLK = 13 et SPI MOSI = 11). La broche 10 doit
être
// une sortie. C'est beaucoup plus rapide, mais également nécessaire si
vous voulez
// commander une carte micro SD
Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS,  TFT_DC, TFT_RST); //
Constructeur

// Option 2: utiliser n'importe quelle broche (SPI Soft) mais un peu
plus lent!
#define TFT_SCLK 13   // A choisir parmi les GPIO
#define TFT_MOSI 11
//Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS, TFT_DC, TFT_MOSI,
TFT_SCLK, TFT_RST);

void setup()
{
  // Initialisation d'un afficheur TFT 1.8"
  tft.initR(INITR_BLACKTAB);   // initialise un circuit ST7735S
  tft.fillScreen(ST7735_BLACK);
}
```

TELECHARGER

Le programme ***DemoST7735*** complet pour un shield V1 ou un breakout est téléchargeable ici

From:
https://webge.fr/dokuwiki/ - **WEBGE Wikis**

Permanent link:
https://webge.fr/dokuwiki/doku.php?id=materiels:afficheurs:ard1_8shv2

Last update: **2023/08/20 18:43**