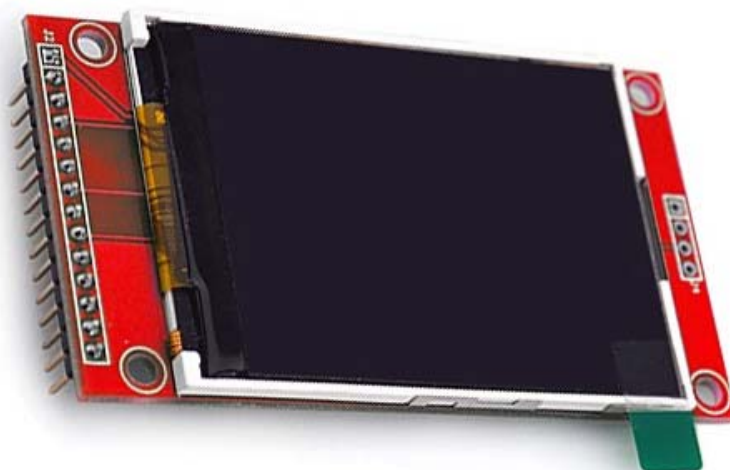


Дисплей сенсорный 2,4 TFT LCD 240x320



Модуль TJCTM24024-SPI позволяет оснастить прибор на основе микроконтроллера цветным (18-битный цвет – 262144 цветов) сенсорным дисплеем с диагональю 2,4 дюйма. Дисплей имеет подсветку. Его удобно пользоваться благодаря большому углу обзора. Контрастность и динамические свойства ЖКИ индикатора [H24TM84A](#) позволяют воспроизводить видео. В состав модуля входит контейнер для CD карты, на которой могут размещаться изображения, видеозаписи и множество фрагментов программ для разгрузки памяти МК. [Дисплей сенсорный 2,4 TFT LCD 240x320](#) имеет ЖКИ контроллер ILI9341 <http://datasheets.gpio.dk/dl/ILI9341.pdf> и контроллер сенсорных функций ADS7843 или [ХРТ2046](#) Контроллер ЖКИ содержит буфер RAM, что снижает требования к микроконтроллеру прибора. Дисплей управляется по последовательной SPI шине. Мощный сервис модуля TJCTM24024-SPI позволяет прекрасно работать с системами Arduino, Raspberry Pi, STM32 Nucleo. Удобный интерфейс великолепно сочетается с МК классов Atmel AVR, 51, PIC и других. Совместим и является глубоким развитием дисплея Nokia LCD 5110 о котором рассказано здесь www.arduino-kit.ru

Характеристики

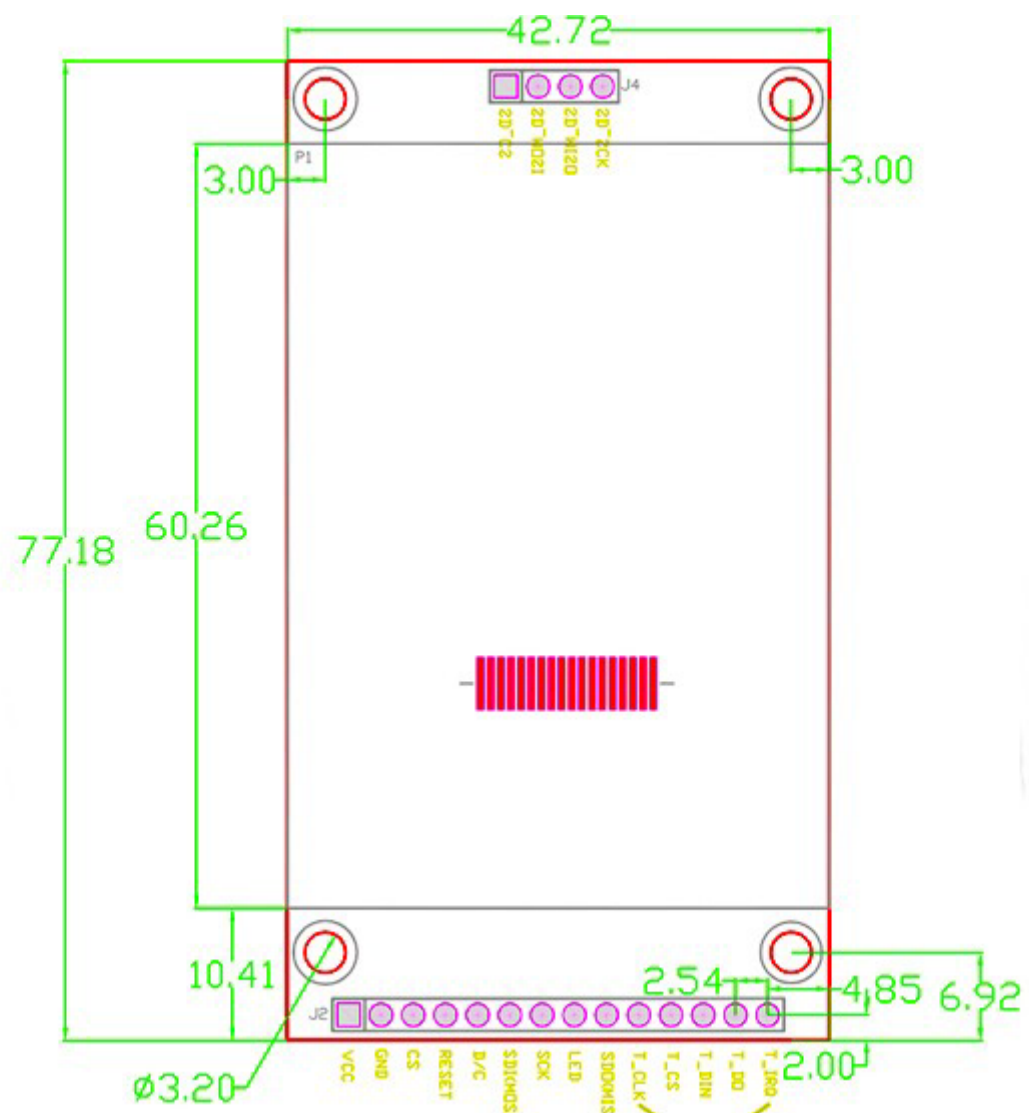
Напряжение питания 3 В или 5 В

Ток потребления при включенной подсветке 100 мА

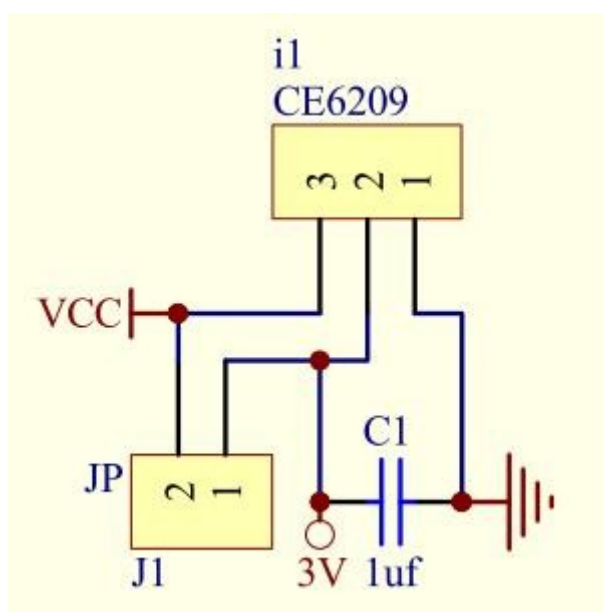
Разрешение экрана 320 x 240 точек

Негативное воспроизведение

Объем SD карты 2 Гбайт



Питание



Цепи питания модуля.

На плате модуля установлена микросхема [CE6209](#) стабилизатора напряжения 3 В для питания модуля от источника 5 В. При питании от источника 3,3 В перемычка J1 должна быть замкнута. Перед включением напряжения питания 5 В проверить отсутствие перемычки J1. Линии питания модуля соединяются с контактом VCC и GND. Питание от 5 В предпочтительнее так как стабилизатор модуля обеспечивает напряжение 3 В более соответствующее требованиям ЖКИ чем напряжение 3,3 В. Подсветка через контакт LED питается напряжением 3 В.

Контакты и сигналы

Интерфейс модуля содержит две отдельные шины SPI. 14-контактный соединитель J2 содержит контакты сигналов интерфейса SPI дисплея: CS, MOSI, MISO, SCK. Они имеют логические уровни с максимальным значением 3 В, а также J2 содержит контакты:

LED – питание подсветки 3 В,
D/C – выбор передачи данных или команд,
RST – сброс, низкий активный уровень.

Соединитель J2 содержит контакты сигналов сенсорного управления, их название начинается с T_. Интерфейс SPI SD-карты выведен на набор контактов J4.

- 1 SD_CS
- 2 SD_MOSI
- 3 SD_MISO
- 4 SD_SCK

При подключении к микроконтроллеру питающемуся от 5 В дисплей сенсорный 2,4 TFT LCD 240x320 подключают через цепи согласования уровней. В противном случае TJCTM24024-SPI испортится. В качестве цепей согласования применяют резисторные делители, диодные ограничители совместно с резисторами, специальные микросхемы и другие схемные решения.

Ознакомительное включение

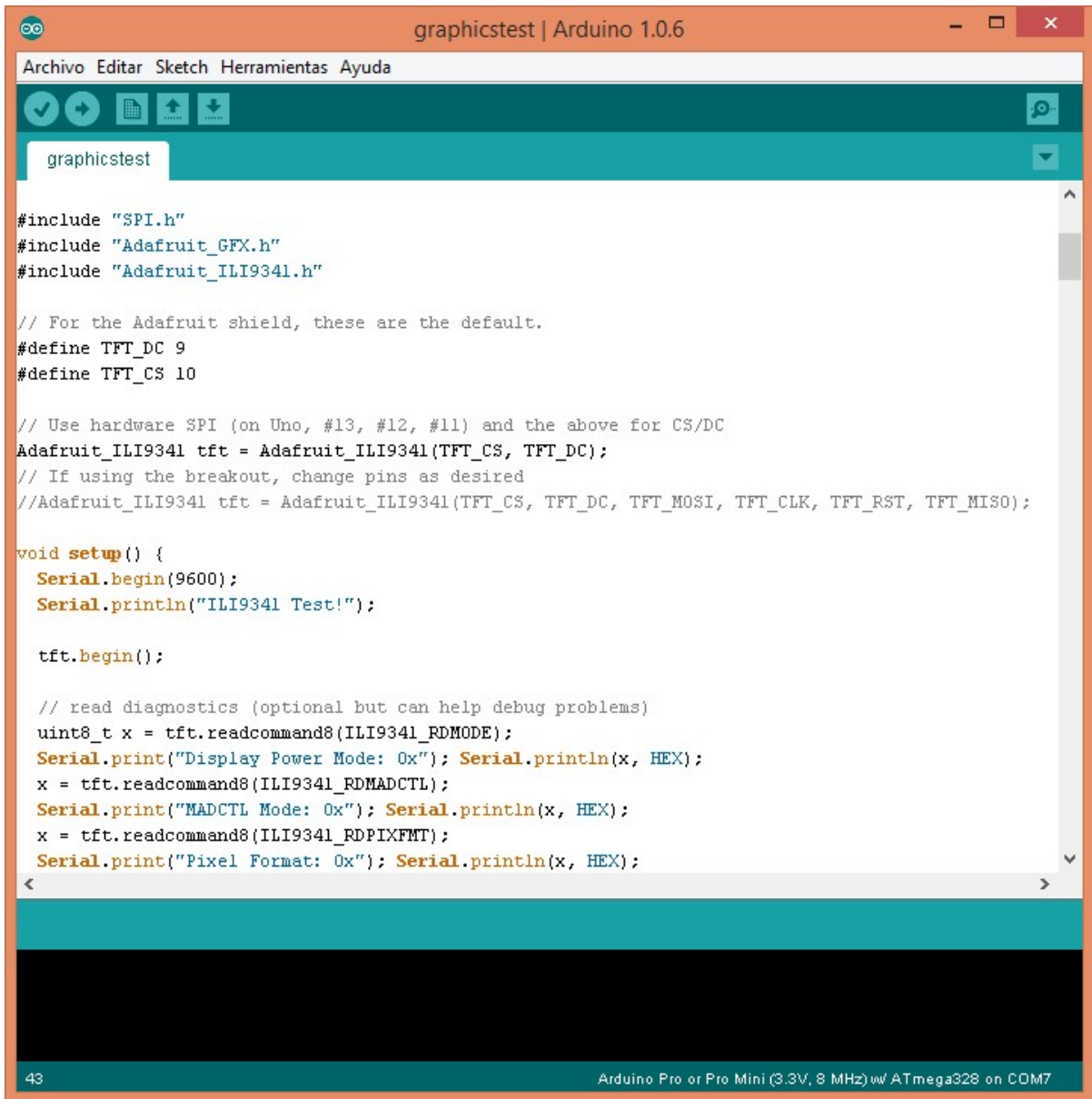
Проверка

Для ознакомления с дисплеем возьмем Arduino Pro Mini, имеющее питание 3,3 В. Подключение одинаковое для нескольких типов Arduino.

Дисплей Arduino UNO, NANO или PRO MINI

SDO (MISO)	контакт 12
LED	3.3 V
SCK	13
SDI (MOSI)	11
D/C	9
RESET	3.3 V
CS	10
GND	GND
VCC	3.3 V

Проведем тестирование программной библиотеки Adafruit ILI9341 нашей программой graphicstest. Предварительно установим программные библиотеки Adafruit ILI9341 https://github.com/adafruit/Adafruit_ILI9341 и Adafruit GFX <https://github.com/adafruit/Adafruit-GFX-Library> Наберем текст программы.



```
graphicstest | Arduino 1.0.6
Archivo Editar Sketch Herramientas Ayuda

#include "SPI.h"
#include "Adafruit_GFX.h"
#include "Adafruit_ILI9341.h"

// For the Adafruit shield, these are the default.
#define TFT_DC 9
#define TFT_CS 10

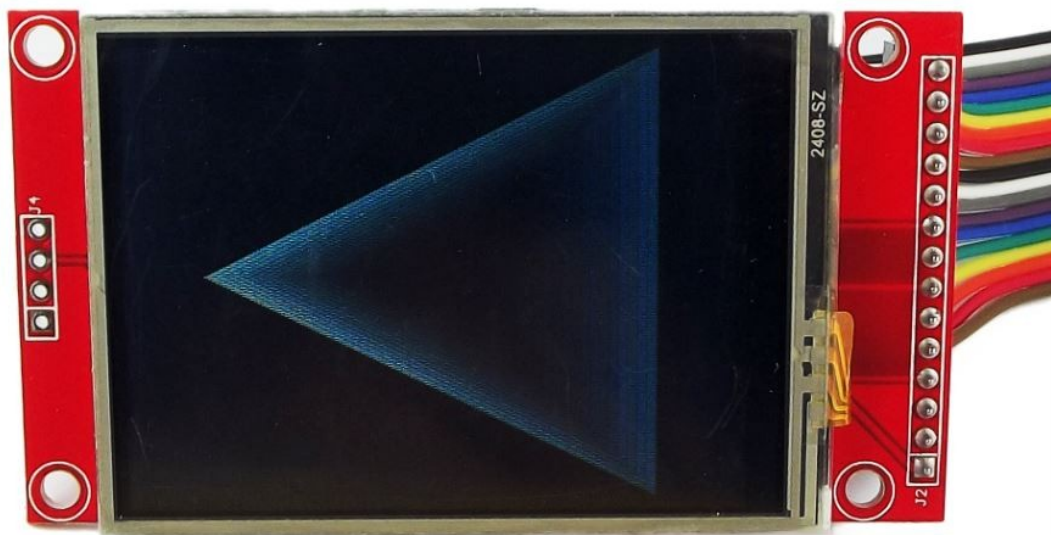
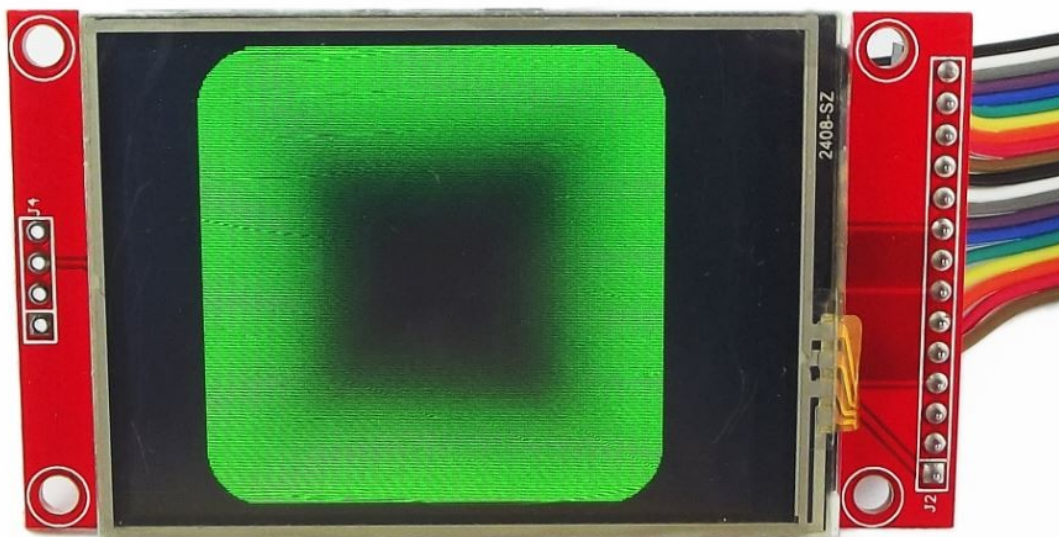
// Use hardware SPI (on Uno, #13, #12, #11) and the above for CS/DC
Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC);
// If using the breakout, change pins as desired
//Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC, TFT_MOSI, TFT_CLK, TFT_RST, TFT_MISO);

void setup() {
  Serial.begin(9600);
  Serial.println("ILI9341 Test!");

  tft.begin();

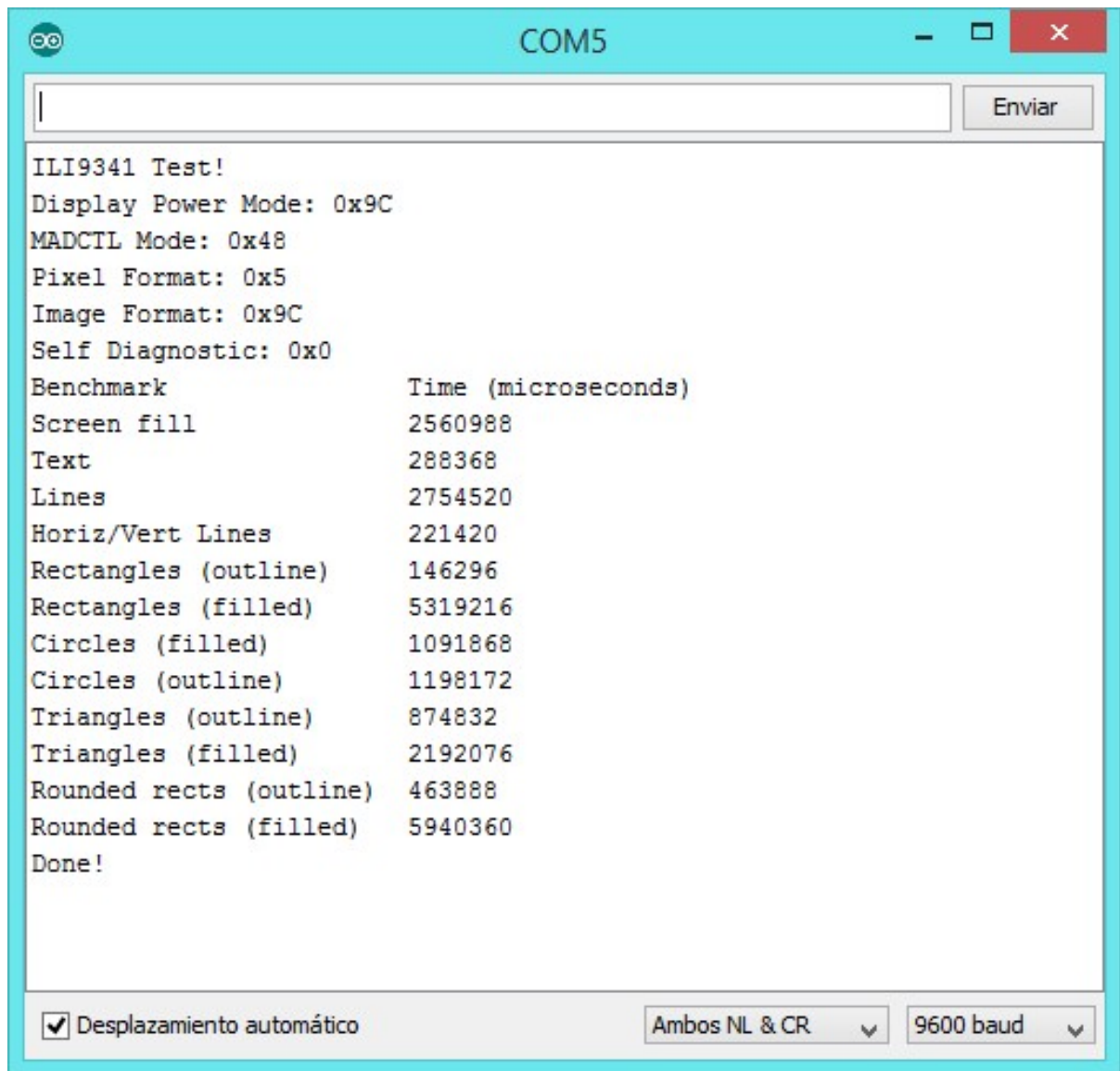
  // read diagnostics (optional but can help debug problems)
  uint8_t x = tft.readcommand8(ILI9341_RDMODE);
  Serial.print("Display Power Mode: 0x"); Serial.println(x, HEX);
  x = tft.readcommand8(ILI9341_RDMADCTL);
  Serial.print("MADCTL Mode: 0x"); Serial.println(x, HEX);
  x = tft.readcommand8(ILI9341_RDPIXFMT);
  Serial.print("Pixel Format: 0x"); Serial.println(x, HEX);
}
```

Следует без изменений компилировать и загрузить программу. Включим соединенные модули. На экране должны появляться различные изображения показанные далее.





Если открыть последовательный монитор, то увидим характеристики информации отображаемой дисплеем.



Перейдем к более полезным примерам.

Индикация значения напряжения на аналоговом входе

Для этого нужно подвижный контакт потенциометра соединить с аналоговым входом AN0, а остальные его контакты соединить с общим проводом и питанием 3 В. Загрузите следующую программу.

```
//Adafruit and SPI libraries
#include
#include
#include

//Pins
#define TFT_DC 9
#define TFT_CS 10
//Create ILI9341 instance
Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC);

void setup(){
```

```

//TFT init
tft.begin();
tft.setRotation(1); //Horizontal
tft.fillScreen(ILI9341_RED);
tft.setTextColor(ILI9341_WHITE);
tft.setTextSize(4);

// We write the text:
tft.setCursor(60,30);
tft.print("Voltage: ");
tft.setTextSize(2);
tft.setTextColor(ILI9341_WHITE);
tft.setCursor(20,200);
tft.print("http://arduino-kit.ru");

// Let we set to a text size 8
tft.setTextSize(8);
tft.setTextColor(ILI9341_GREEN);
tft.setCursor(250,100);
tft.print("V");

}

float voltage=0;
float voltageAnt=0;
void loop(){

// Read the value of volteje
voltage = analogRead(A0) * (5.0 / 1023.0);
if(voltage-voltageAnt>0.01||voltage-voltageAnt<-0.01)
{
tft.setTextColor(ILI9341_RED);
tft.setCursor(30,100);
tft.print(voltageAnt);

tft.setTextColor(ILI9341_GREEN);
tft.setCursor(30,100);
tft.print(voltage);
voltageAnt=voltage;

}
delay(100);
}

```

Назначение программных функции приведенной выше программы.

```

#define TFT_DC 9
#define TFT_CS 10
Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC);

```

Как видно здесь просто указываются контакты Arduino.

Инициализация TFT

```
tft.begin();
```

Ориентация экрана

```
tft.setRotation(1); //Horizontal
```

Значения вводятся от 0 до 3, где 0 – исходное положение, 1 – поворот на 90 °, 2 – поворот на 180 °, 3 – поворот на 270 °.

Цвет фона и цвет текста.

```
tft.fillScreen(background);
```

```
tft.setTextColor(txtColor);
```

FillScreen() заполняет весь экран указанным цветом фона, SetTextColor() устанавливает цвет текста, который будет написан позже. Цвета, которые мы используем:

ILI9341_BLACK, ILI9341_NAVY, ILI9341_DARKGREEN, ILI9341_DARKCYAN,

ILI9341_MAROON, ILI9341_PURPLE, ILI9341_OLIVE, ILI9341_LIGHTGREY,

ILI9341_DARKGREY, ILI9341_BLUE, ILI9341_GREEN, ILI9341_CYAN, ILI9341_RED,

ILI9341_MAGENTA, ILI9341_YELLOW, ILI9341_WHITE, ILI9341_ORANGE,

ILI9341_GREENYELLOW, ILI9341_PINK.

Размер текста, и установка курсора.

```
tft.setTextSize(5);
```

```
tft.setCursor(x,y);
```

Размер текста указывает на толщину шрифта в точках. Положение курсора определено в (x,y). Позиция (0,0) – верхний левый угол.

Распечатать текст.

```
tft.print("Voltage: ");
```

Эта функция наследует все особенности функции printf().

Заметьте, что в программе нет функции удаления текста. Для этого следует написать тот же текст цветом фона. Результатом работы программы должно быть следующее:



Осциллограф из Arduino

Этот пример аналогичен предыдущему, за исключением того, что вместо того чтобы показывать напряжение в виде текста, оно отображается графически. Текст программы далее.

```
//Adafruit and SPI libraries
#include
#include
#include

//Pins
#define TFT_DC 9
#define TFT_CS 10
//Create ILI9341 instance
Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC);

// Variable for the X coordinate
int xPos = 0;

void setup(){
  //Start TFT
  tft.begin();
  tft.setRotation(1); //Horizontal
  limpiar(); // function to clean the screen
}

void loop(){
  // Read the voltage value
  int voltage = analogRead(A0);
  int voltageEscalado = map(voltage, 0, 1023, 0, tft.height());
```

```

// Draw a rectangle of 3x3 pixels
tft.fillRect(xPos, tft.height()-voltageEscalado-1, 3, 3, ILI9341_GREEN);

// And check to see if we reached the edge of the TFT
if(xPos >= tft.width()-1){
  xPos = 0;
  limpiar();
}
else{
  xPos++;
}
delay(5); // Sampling Time
}

void limpiar()
{
  tft.fillScreen(ILI9341_RED);
  tft.setTextSize(2);
  tft.setTextColor(ILI9341_WHITE);
  tft.setCursor(80,5);
  tft.print("OSCILOSCOP");

  tft.setTextColor(ILI9341_WHITE);
  tft.setCursor(20,220);
  tft.print("http://arduino-kit.ru");
}

```

Происходит чтение значения между 0 и значением соответствующим высоте масштаба отображения. Точка отображается в позиции x,y. Программа увеличивает значение координаты x. Когда достигнут конец экрана – экран стирается и начинается отображение с x=0.



Тестирование touch-screen

Установим соединения между контактами:

Дисплей Arduino UNO, NANO или PRO MINI

T_IRQ	7
T_DO	6
T_DIN	5
T_CS	4
T_CLK	3
SDO (MISO)	12
LED	3.3 V
SCK	13
SDI (MOSI)	11
D/C	9
RESET	3.3 V
CS	10
GND	GND
VCC	3.3 V

Новые соединения добавлены только для сигналов касания. Другие соединения остаются такими же, как и в предыдущем примере. Для работы с сенсорным экраном понадобится [библиотека UTouch](#).

```
#include
#include
#include
#include

//Pins TFT
#define TFT_DC 9
#define TFT_CS 10

//Pins Touch
#define t_SCK 3
#define t_CS 4
#define t_MOSI 5
#define t_MISO 6
#define t_IRQ 7

// Instantiating the TFT
Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC);

// Instantiating the Touch
UTouch ts(t_SCK, t_CS, t_MOSI, t_MISO, t_IRQ);

void setup(){
  // TFT configuration
  tft.begin();
  tft.setRotation(1); //Must be 1 or else touchscreen gets lost
  tft.fillScreen(ILI9341_YELLOW); // Yellow Background
```

```

// Touch Settings
ts.InitTouch();
ts.setPrecision(PREC_MEDIUM);
tft.setTextColor(ILI9341_BLUE);
tft.setTextSize(3);
tft.setCursor(80,5);
tft.print("Test Touch ");
tft.setTextColor(ILI9341_RED);
tft.setTextSize(2);
tft.setCursor(20,220);
tft.print("http://arduino-kit.ru");
}

void loop()
{
  long x, y;

  while(ts.dataAvailable())
  {
    ts.read();
    x = ts.getX()+15;
    y = ts.getY()+5;
    if((x!=-1) && (y!=-1))
    {
      int radius = 3;
      tft.fillCircle(x, y, radius, ILI9341_BLACK);
    }
  }
}

```

Функции библиотеки uTouch.

Создание экземпляра Touch.

```

#define t_SCK 3
#define t_CS 4
#define t_MOSI 5
#define t_MISO 6
#define t_IRQ 7
UTouch ts(t_SCK, t_CS, t_MOSI, t_MISO, t_IRQ);

```

Библиотека может работать с любыми контактами кроме контактов Arduino SPI.

Инициализации и настройки сенсорного экрана.

```

ts.InitTouch();
ts.setPrecision(PREC_MEDIUM);

```

После инициализации, используя сенсорный экран InitTouch(), следует установить точность с помощью функции setprecision(), доступны: PREC_LOW, PREC_MEDIUM, PREC_HI и PREC_EXTREME.

Проверка нажатия на экран.

ts.dataAvailable()

Функция dataAvailable() возвращает логическое значение, истина при нажатии на экран, и ложное значение в противном случае.

Ложится на позиции сенсорного экрана

```
x = ts.getX()+15;  
y = ts.getY()+5;
```

Координаты сенсорного слоя и ЖКИ индикатора, как правило, не совпадают. Для их совмещения необходимо добавить или вычесть из координат, что и делается в программе. При работе программы должен появиться желтый фон. Нажмите на экран, должна появиться точка контрастного цвета. Если перемещать нажатие, то мы увидим линию.



Для очистки экрана нажмите кнопку сброса Arduino.

[Техническая документация](#)

[Схема электрическая](#)

[Программирование микроконтроллера класса 51](#)

[Пример использования](#)