- Elettronica Open Source - https://it.emcelettronica.com -

Programmiamo la scheda Arduino Esplora

Posted By *Adriano Gandolfo* On 6 settembre 2013 @ 5:00 In Arduino | 4 Comments

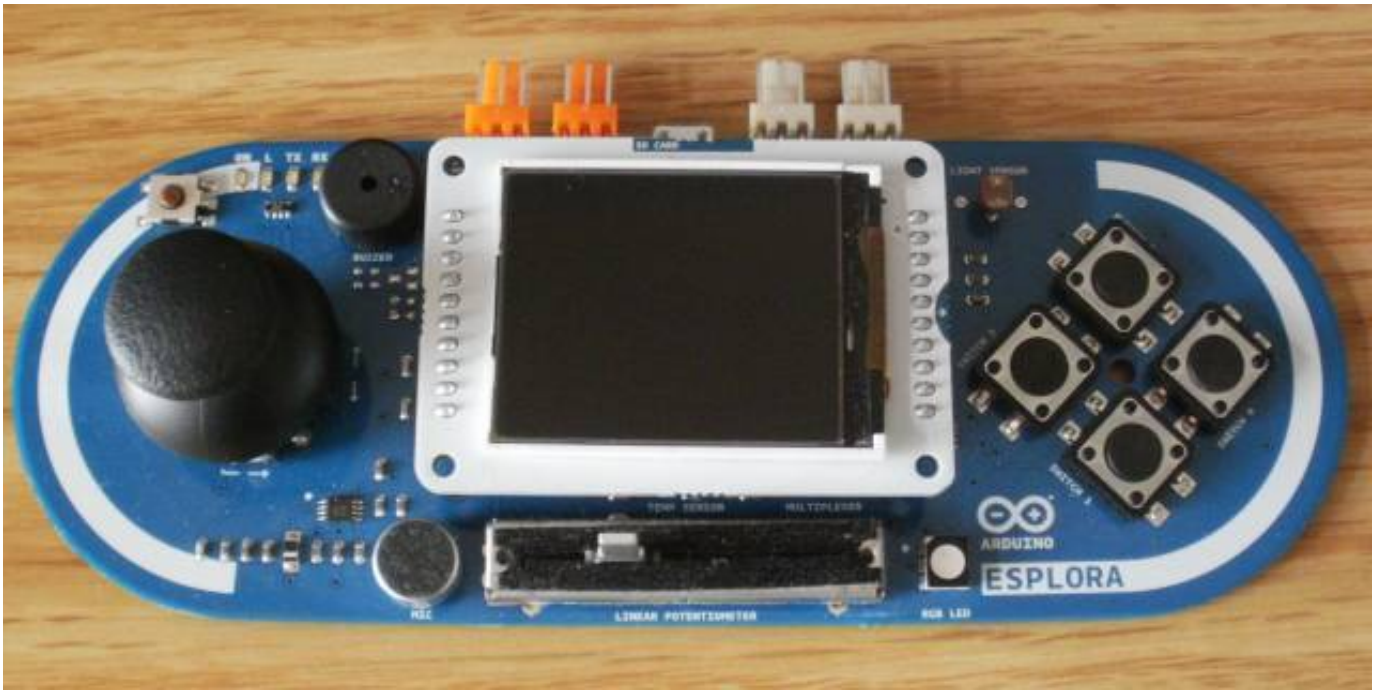
In un precedente articolo è stata presentata l'ultima scheda del mondo Arduino [1], la Esplora. E' stato analizzato come funziona presentandone i vari sensori presenti. In quest'articolo vediamo come installare i driver di gestione e analizziamo la libreria di comandi che ne permettono il suo utilizzo.

La Arduino ESPLORA, la cui presentazione è riportata nell'articolo Scopriamo la nuova scheda Arduino Esplora [2], a prima vista sembra un controller di videogame, in realtà è un dispositivo che ha un piccolo computer chiamato microcontrollore e una serie di ingressi e uscite.

Per gli ingressi: sono disponibili un joystick [3], quattro pulsanti, un sensore di luce, un potenziometro a cursore, un microfono, un sensore di temperatura, e un accelerometro.

Per le uscite c'è un "cicalino" e un led RGB.

Con la ESPLORA è possibile scrivere un software che, utilizzando le informazioni acquisite dagli ingressi, gestisce le uscite della scheda o controlla il computer, proprio come si potrebbe fare con un mouse o una tastiera.



Scaricamento del Software Arduino.

Se siete già possessori di schede Arduino e mantenete costantemente aggiornato il software di gestione, probabilmente potrete saltare questo passo. Nel caso invece, ne foste sprovvisti dovrete scaricare l'ultima versione dell' IDE Arduino dalla pagina di download.

http://arduino.cc/en/Main/Software [4]

Attualmente è disponibile la versione 1.0.5. Al termine del download, salvate il file scaricato in qualsiasi directory. Assicuratevi di conservare la struttura delle cartelle.

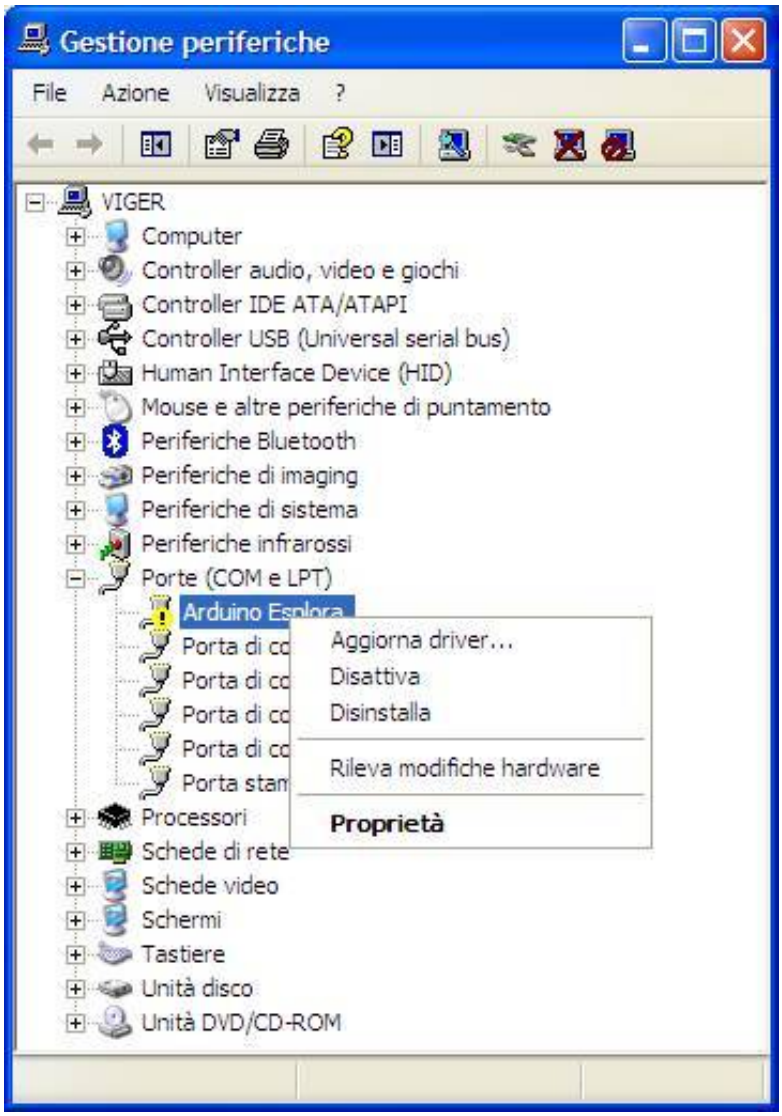
Collegamento della scheda al PC.

La prima operazione da compiere è quella di connettere la scheda al PC: si utilizzerà il cavo USB presente all'interno della confezione, il cavo dispone da un lato di una tipo "A" da un lato e una di tipo "Micro-B" dall'altro. La presa è simile a quella presente su alcuni telefoni cellulari o lettori musicali portatili che utilizzano questo tipo di cavo per il trasferimento dati da/verso il PC. Colleghiamo quindi la scheda Arduino al computer; l'avvenuto collegamento è segnalato dall'accensione del LED di alimentazione verde (etichetta ON) mentre il LED giallo contrassegnato con "L" dopo una prima accensione, dopo circa 8 secondi inizierà a lampeggiare.

Installazione dei driver.

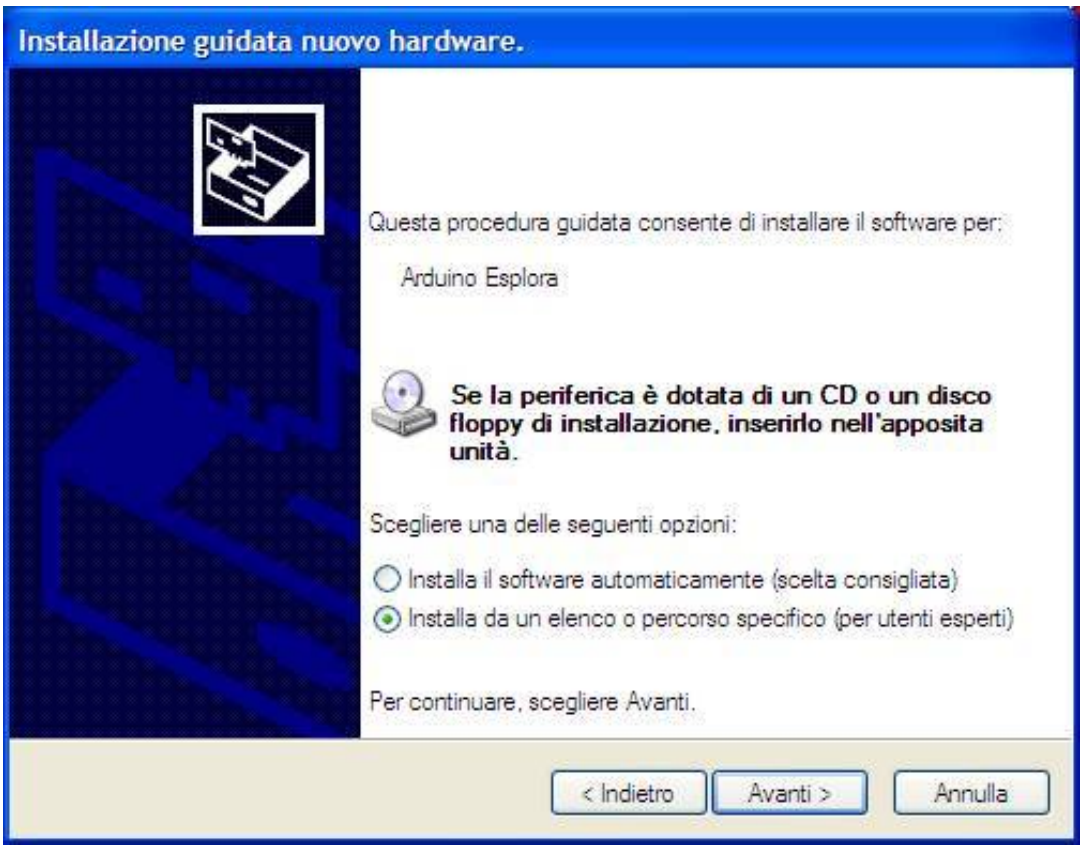
Se è la prima volta che si collega la scheda, sarà necessario l'istallazione dei driver, la procedura è diversa a seconda del sistema operativo. In questa guida vedremo quella per il sistema operativo per Windows XP valida anche per Windows 7, con piccole differenze nelle finestre di dialogo.

Collegate la vostra scheda e attendete sino a che Windows avvia il processo di installazione del driver. Se il programma di installazione non viene avviato automaticamente, individuate la Gestione periferiche di Windows (Start> Pannello di controllo> Hardware) e trovate il riferimento ad Arduino ESPLORA. Con il tasto destro premuto, scegliete Aggiorna driver.

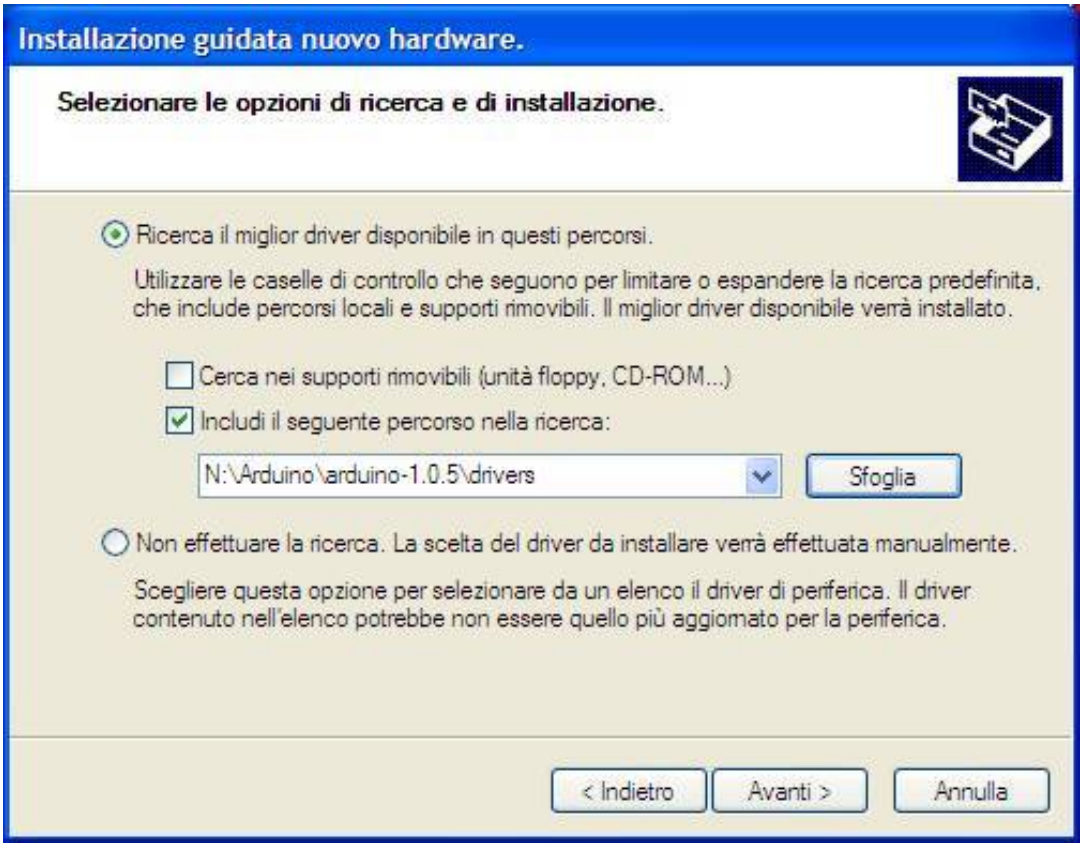


Nella schermata successiva, sarà proposto di consentire a Windows Update di ricercare il software, selezionare "No, non ora" e fare click su Avanti.

scegliere "Installa da un elenco o percorso specifico" e fare clic su avanti

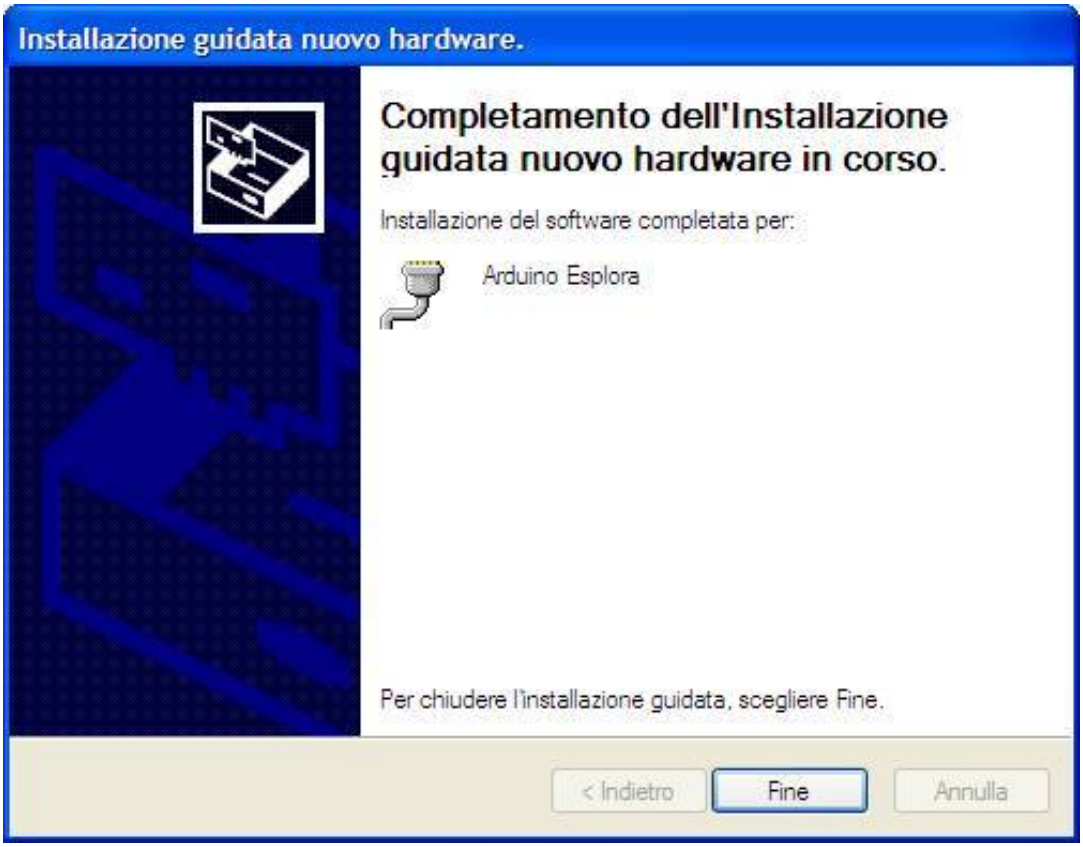


Nella prossima schermata si indicherà la posizione in cui è presente la cartella con il software Arduino che avete appena scaricato. Selezionare la certella driver, quindi fare clic su Avanti.



Si riceverà una notifica che la scheda non ha superato il test di Windows Logo. Fare clic sul pulsante Continua.

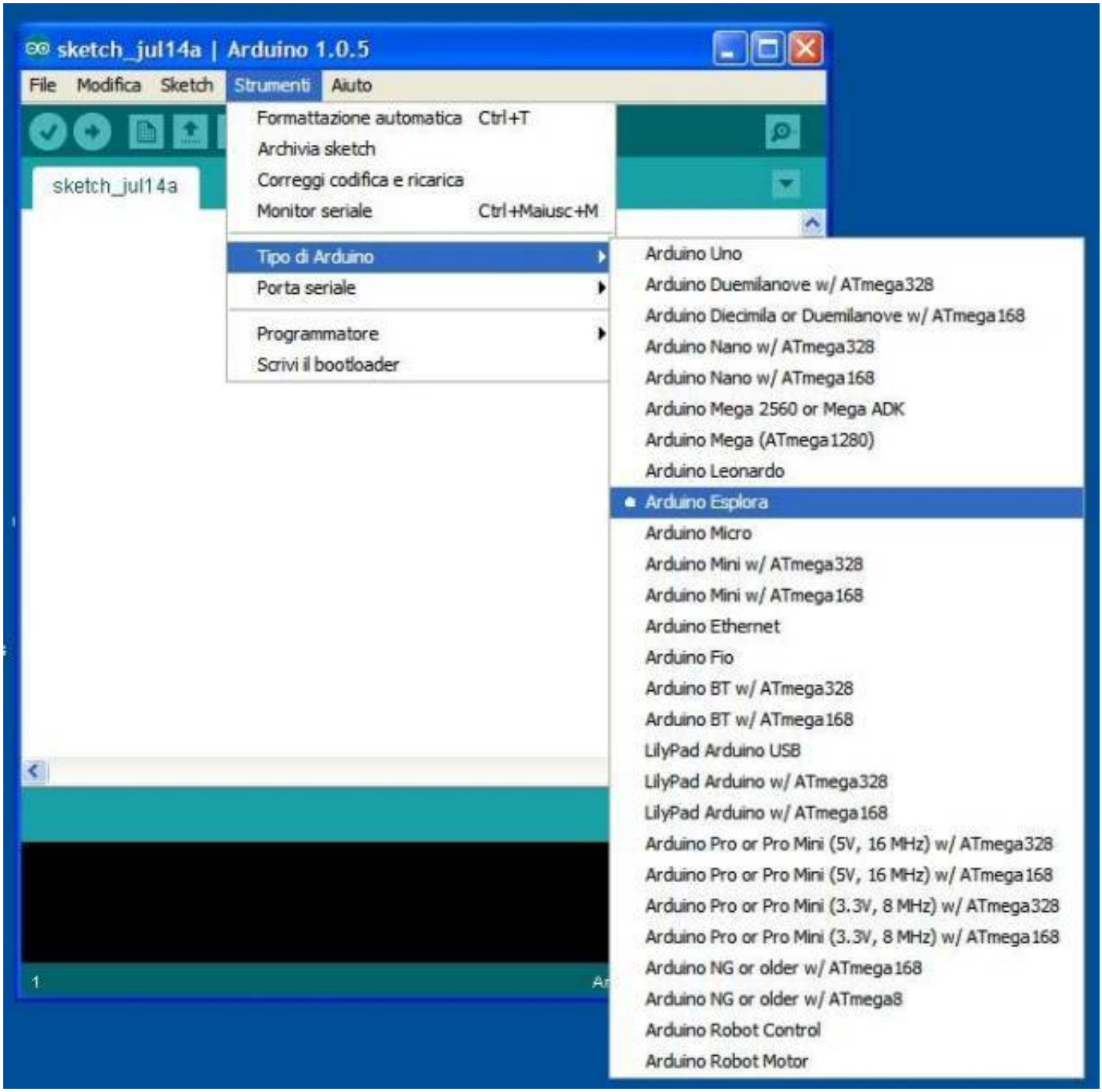
Dopo pochi istanti, una finestra segnalerà che la procedura guidata ha terminato l'installazione del software per Arduino ESPLORA. Premere il pulsante Fine.



Configuriamo l'IDE per utilizzare la Arduino ESPLORA.

Dal momento che l'IDE Arduino è utilizzato per molte schede Arduino diverse, è necessario impostare l'IDE con la scheda ESPLORA.

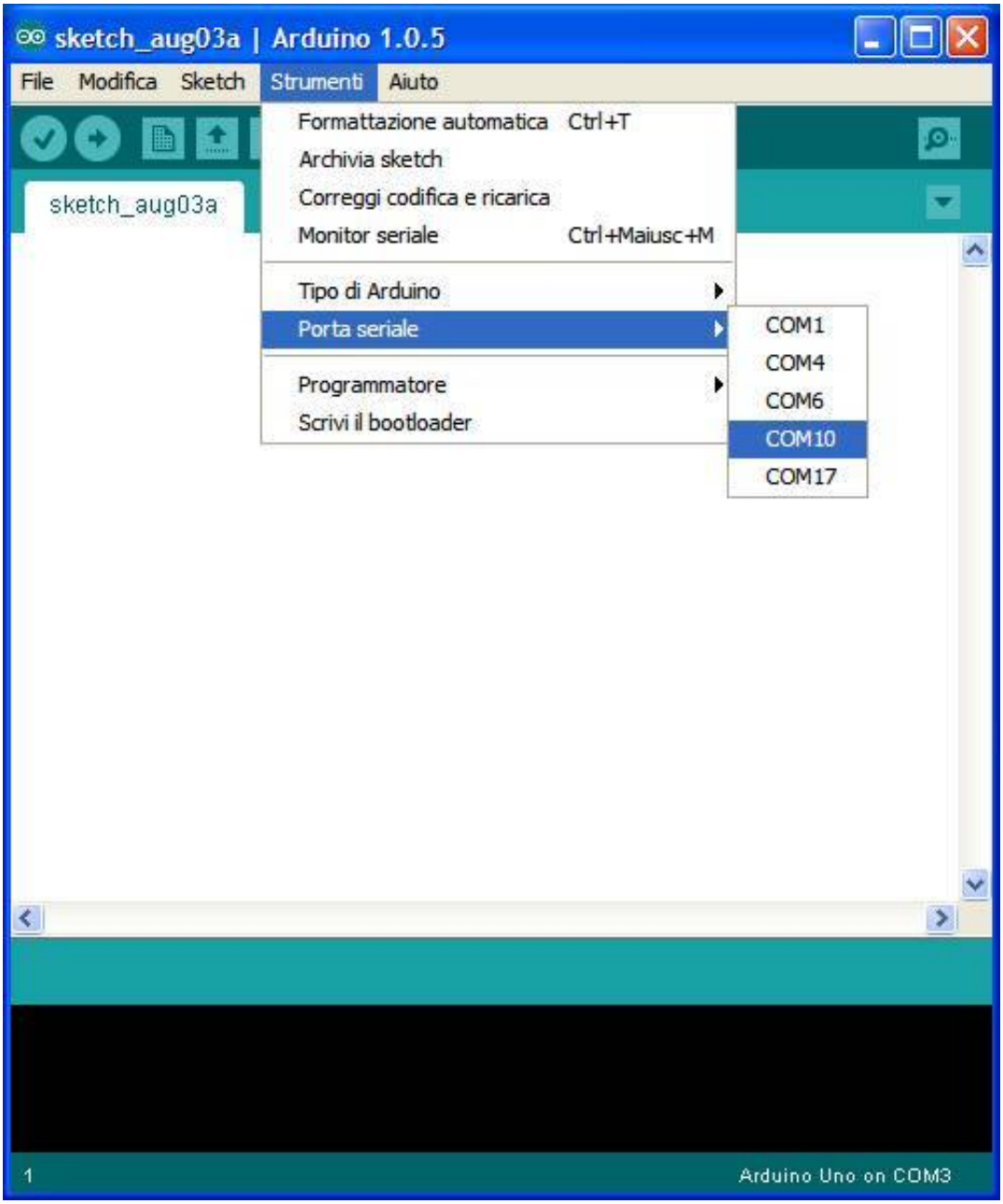
Per selezionare la scheda aprire il menu Strumenti> Tipo di Arduino e scegliere **Arduino ESPLORA**.



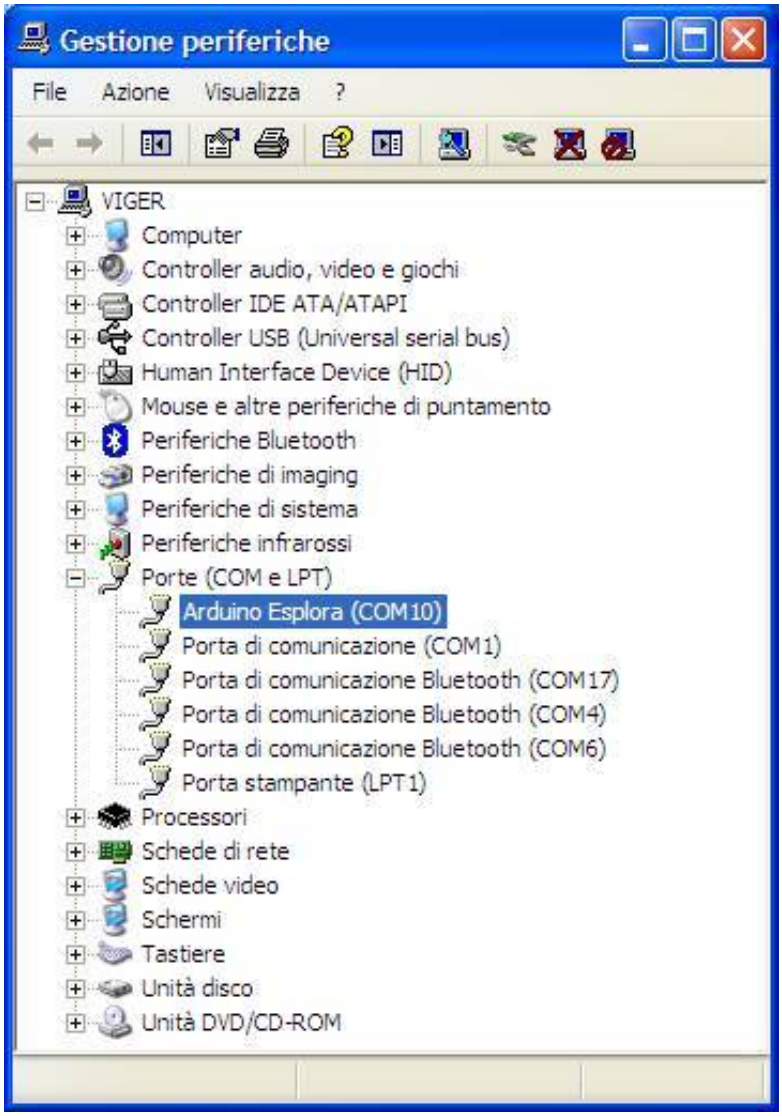
Selezionare la porta USB.

Per programmare la scheda, l'IDE ha bisogno di sapere la porta USB a cui è collegata. Nel menu Strumenti>Porta seriale sono elencate le porte disponibili.

Tra quelle disponibili selezionare quella a cui è collegata la scheda



Il numero della porta è visibile da: Gestione periferiche di Windows (Start> Pannello di controllo> Hardware) e trovate il riferimento a Arduino ESPLORA.



Altro metodo è quello di scollegare ESPLORA e riaprire il menu, la voce che scompare dovrebbe essere la scheda ESPLORA. Collegare nuovamente la scheda e selezionare quella porta seriale.

Le librerie per la gestione della scheda ESPLORA.

Con l'uscita della scheda ESPLORA sono state inserire nell'IDE (dalla versione 1.0.3 in poi) le librerie per la sua gestione, tra cui:

Esplora : con una serie di funzioni per l'interfacciamento con i sensori e gli attuatori montati sulla scheda.
TFT: per la gestione del display 160X128 di cui può essere dotata la scheda.
SD: per la gestione della scheda SD che può essere istallata sul display TFT
Sul sito sono disponibili degli esempi per meglio comprenderne il loro uso.

La libreria Esplora.

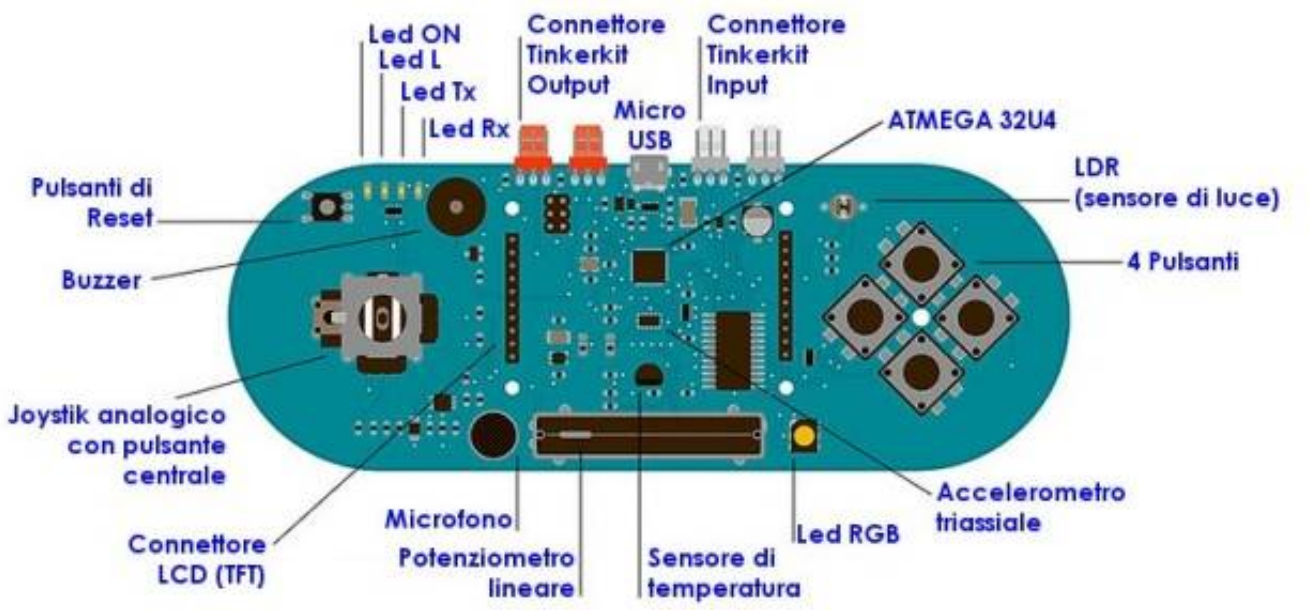
La libreria offre un facile accesso ai dati dai sensori di bordo e fornisce la possibilità di cambiare lo stato delle uscite. Le funzioni della libreria sono accessibili tramite la classe

ESPLORA. (http://arduino.cc/en/Reference/EsploraLibrary [5])

Ricapitolando, i sensori che sono disponibili sulla scheda:

Joystick analogico a due assi.
Pulsante centrale del joystick
4 pulsanti
Un microfono
Un sensore di luce
Un sensore di temperatura
Un zccelerometro a 3 assi
2 connettori di ingresso Tinkerkit
Abbiamo poi i seguenti attuatori:

Un led RGB (Rosso-verde-blu)
Un piezo buzzer
2 connettori di uscita Tinkerkit



Per la gestione delle funzioni disponibili all'interno della libreria troviamo:

Esplora.readSlider()
Esplora.readLightSensor()
Esplora.readTemperature()
Esplora.readMicrophone()
Esplora.readJoystickSwitch()
Esplora.readJoystickButton()
Esplora.readAccelerometer()
Esplora.readButton()
Esplora.readJoystickX()
Esplora.readJoystickY()
Esplora.writeRGB()
Esplora.writeRed()

Esplora.writeGreen()
Esplora.writeBlue()
Esplora.readRed()
Esplora.readGreen()
Esplora.readBlue()
Esplora.tone()
Esplora.noTone()
Analizziamo le varie funzioni: per ognuna è presente la descrizione, la sintassi del comando, gli eventuali parametri da inserire con il loro range e l'eventuale dato che viene fornito dalla funzione.

Esplora.readSlider()

La funzione legge il valore del potenziometro lineare come un numero a 10 bit. Ciò significa che mapperà la tensione d'ingresso compresa tra 0 e 5 volt in valori interi compresi tra 0 e 1023. Questo produce una risoluzione tra le letture di: 5 volt / 1024 unità oppure 0,0049 volt (4,9 mV) per unità.

Sintassi: *Esplora.readSlider()*

Dato fornito int: La posizione del potenziometro lineare, mappato a un valore compreso tra 0 e 1023.

Esplora.readLightSensor()

La funzione legge l'intensità della luce che colpisce il sensore (rappresentato da una fotocellula) come un numero a 10 bit. Ciò significa che mapperà tensioni d'ingresso analogamente al comando Esplora.readSlider()

Sintassi: *Esplora.readLightSensor()*

Dato fornito: int : La quantità di luce che colpisce il sensore, mappato a un valore tra 0 e 1023.

Esplora.readTemperature()

Legge la temperatura dell'ambiente fornito dal sensore e restituisce una lettura sia nella scala Celsius che Fahrenheit, a seconda del parametro passato.

Sintassi: *Esplora.readTemperature(scale)*

Parametri: scale: la scala in uscita, argomenti validi sono: DEGREES_C e DEGREES_F

Dato in uscita: int: La lettura della temperatura in gradi Celsius o Fahrenheit. In Celsius l'intervallo è compreso tra -40 ° C a 150 ° C, la gamma Fahrenheit tra -40 ° F a 302 ° F.

Esplora.readMicrophone()

Legge l'ampiezza fornita dal microfono, restituendo un valore tra 0 e 1023.

Sintassi: *Esplora.readMicrophone()*

Dato fornito int: Il valore dell'ampiezza mappato in una gamma da 0 a 1023.

Esplora.readJoystickSwitch()

Legge lo stato del tasto del joystick e restituisce 0 o 1023. Se si preferisce qualcosa di più coerente con la funzione readButton (), si consiglia di utilizzare la funzione readJoystickButton (). Tale funzione si comporta nello stesso modo ma restituisce LOW quando si preme il tasto del joystick, e HIGH quando non è premuto.

Sintassi: *Esplora.readJoystickSwitch()*

Valore fornito 0 quando il tasto è premuto, 1023 quando il tasto non è premuto (stato normale).

Esplora.readAccelerometer()

Legge i valori forniti dall'accelerometro. Ciascuno dei tre assi è accessibile indipendentemente.

Sintassi: *Esplora.readAccelerometer(axis)*

Parametri: axis : char, determina quale asse deve essere letto.

X_AXIS per leggere il valore dell'asse X.
Y_AXIS per leggere il valore dell'asse Y.
Z_AXIS per leggere il valore dell'asse Z.
Dato fornito int: il valore delle letture sull'asse prescelto. L'accelerometro restituisce zero quando è perpendicolare alla direzione della gravità. Sono forniti valori positivi o negativi quando è mosso in una delle due direzioni dell'asse.

Esplora.readButton()

Legge lo stato di un pulsante e restituisce se è HIGH o LOW

Sintassi: *Esplora.readButton(button)*

Parametri: button: il pulsante associato che si vuole leggere.

Argomento validi sono:

SWITCH_1 oppure SWITCH_DOWN
SWITCH_2 oppure SWITCH_LEFT
SWITCH_3 oppure SWITCH_UP
SWITCH_4 oppure SWITCH_RIGHT

JOYSTICK_DOWN = JOYSTICK_BASE
JOYSTICK_LEFT = JOYSTICK_BASE+1
JOYSTICK_UP = JOYSTICK_BASE+2
JOYSTICK_RIGHT = JOYSTICK_BASE+3
Dato fornito: LOW quando premuto, HIGH quando non premuto (situazione normale).

Esplora.readJoystickY()

Utilizzata per leggere la posizione dell'asse Y del joystick. Quando il joystick è al centro, restituisce zero. Valori positivi indicano che il joystick è spostato in alto, valori negativi quando questo è spostato verso il basso.

Sintassi: *Esplora.readJoystickY()*

Dato fornito:

int: 0 quando il joystick è in mezzo l'asse y; valori tra 1 e 512 quando il joystick è spostato in alto; valori tra -1 e -512 quando il joystick viene spostato verso il basso.

Esplora.readJoystickX()

Utilizzata per leggere la posizione dell'asse X del joystick. Quando il joystick è al centro, restituisce zero. Valori positivi indicano che il joystick è spostato a destra, valori negativi quando è spostato a sinistra.

Sintassi: *Esplora.readJoystickX()*

Dato fornito:

int: 0 quando il joystick è nel mezzo dell'asse x; valori tra 1 e 512 quando il joystick viene spostato verso destra; -1 e -512 quando il joystick viene spostato a sinistra.

Esplora.writeRGB()

Utilizzata per scrivere i valori che rappresentano la luminosità del LED RGB. Mescolando valori diversi, è possibile creare diverse combinazioni di colori.

Sintassi: *Esplora.writeRGB(red, green, blue)*

Parametri:

red: int, valore per modificare la luminosità del led rosso con un range tra 0 e 255
green: int, valore per modificare la luminosità del led verde con un range tra 0 e 255
blue: int, valore per modificare la luminosità del led blu con un range tra 0 e 255
Esplora.writeRed();Esplora.writeGreen();Esplora.writeBlue();

Utilizzata per variare la luminosità dei singoli led contenuti nel led RGB

Sintassi:

Esplora.writeRed(value)
Esplora.writeGreen(value)
Esplora.writeBlue(value)
Parametri:

value: int, valore per modificare la luminosità del led rosso (oppure verde o blu) con un range tra 0 e 255

Esplora.readRed(),readGreen(), readBlue()

Utilizzata per leggere il valore di luminosità di uno dei tre LED RGB.

Sintassi:

Esplora.readRed()
Esplora.readGreen()
Esplora.readBlue()
Dato fornito:

int : valore della luminosità del led rosso (oppure verde o blu) con un range tra 0 e 255

Esplora.tone()

Genera un'onda quadra di frequenza specificata emessa attraverso il buzzer di bordo di ESPLORA. Può essere specificato la sua durata espressa in millisecondi, altrimenti l'onda continua finché non si effettua una chiamata a Esplora.noTone (). Può essere generato un solo tono alla volta. Se il segnale è già in riproduzione, ne viene variata la frequenza. L'uso della funzione Esplora.tone () può interferire con la dissolvenza il LED rosso.

Sintassi: *Esplora.tone(frequency, duration)*

Parametri:

frequency: la frequenza del tono in hertz - unsigned int
duration: la durata del tono in millisecondi (opzionale) - unsigned long
Esplora.noTone()

La funzione interrompe la generazione di un'onda quadra generato tramite la funzione Esplora.tone (). Non ha effetto se non è generato nessun tono.

Sintassi: *Esplora.noTone()*

Libreria TFT

Questa libreria permette alla scheda Arduino Esplora di comunicare con lo schermo LCD TFT.

http://arduino.cc/en/Reference/TFTLibrary [6]

Semplifica il processo di disegno di forme, linee, immagini e testo sullo schermo. La biblioteca estende le librerie Adafruit GFX, e Adafruit ST7735 su cui si basa. Adafruit è la società che ha creato il primo modello di scheda con display TFT, da cui è stata creato il modello ora commercializzato da Arduino.

La biblioteca GFX è responsabile delle routine di disegno, mentre la libreria ST7735 è specifica per lo schermo dell'Arduino TFT. Le aggiunte specifiche sono state progettate per funzionare in modo simile a delle API (Application Programming Interface).

Per impostazione predefinita, lo schermo è orientato in modo da essere più largo che alto. La parte superiore dello schermo è la stessa parte del testo 'SD CARD'. In quest'orientamento, lo schermo è di 160 pixel di larghezza e 128 pixel di altezza.
Quando ci si riferisce alle coordinate sullo schermo, occorre immaginare una griglia. Ogni quadrato della griglia è un pixel. È possibile identificare la posizione dei pixel con coordinate specifiche. Un punto in alto a sinistra avrebbe il valore 0,0. Se questo punto dovesse spostarsi nella parte superiore destra dello schermo, le sue coordinate sarebbero 0, 159, nell'angolo in basso a sinistra, le coordinate sarebbero 127,0, e in basso a destra 127.159.
E' possibile utilizzare lo schermo in orientamento verticale (chiamato anche "portrait"), richiamando il comando EsploraTFT.setRotation (0), a questo punto le coordinate x ed y degli assi cambiano di conseguenza, così come cambiano i dati forniti da screen.width () o screen.height ().
Come abbiamo visto, sulla scheda è presente uno slot per schede SD, che può essere utilizzata attraverso la libreria SD.

La biblioteca TFT si basa sulla libreria SPI per la comunicazione con lo schermo e la scheda SD e deve essere inclusa in tutti gli sketches.

Utilizzo della libreria

Dato che su Arduino ESPLORA la configurazione dei pin di collegamento con il display è fissa, per la gestione sarà sufficiente la creazione di un solo oggetto ESPLORA. Nella parte iniziale dello sketch dovranno essere presenti le seguenti linee #include:

```
#include <Esplora.h>
#include <TFT.h> // Arduino LCD library
#include <SPI.h>
```

Nella cartella della libreria sono presenti alcuni esempi per meglio comprendere l'utilizzo dei comandi.

Esplora TFT Bitmap Logo: Legge un file di immagine da una scheda micro-SD e lo disegna in posizioni casuali.
Esplora TFT Color Picker: Utilizzando il joystick e il dispositivo di scorrimento, è possibile modificare il colore dello schermo TFT
Esplora TFT Etch a Sketch: Un'implementazione ESPLORA del classico disegno a mano libera.
Esplora TFT Graph: Disegna sul display il Grafico dei valori dal sensore di luce.
Esplora TFT Horizon: Disegna una linea di orizzonte artificiale basato sul tilt dall'accelerometro
Esplora TFT Pong: Un'implementazione del classico gioco
Esplora TFT Temperature: visualizza sullo schermo la temperatura rilevata dal sensore a bordo
Ecco i comandi presenti nella libreria:

```
EsploraTFT.begin()
EsploraTFT.background()
EsploraTFT.stroke()
EsploraTFT.noStroke()
EsploraTFT.fill()
EsploraTFT.noFill()
EsploraTFTtext()
EsploraTFT.setTextSize()
EsploraTFT.begin()
EsploraTFT.point()
EsploraTFT.line()
EsploraTFT.rect()
EsploraTFT.width()
EsploraTFT.height()
EsploraTFT.circle()
PImage
EsploraTFT.image()
EsploraTFT.loadImage()
PImage.height()
```

PImage.width()
PImage.isValid

EsploraTFT.begin()

La funzione deve essere chiamata per inizializzare lo schermo prima di disegnare o scrivere qualsiasi cosa.

Sintassi: *EsploraTFT.begin();*

EsploraTFT.background ()

La funzione cancella tutto il contenuto dello schermo con il colore indicato.

Può essere utilizzato all'interno di un loop () per cancellare lo schermo.

Lo schermo ha la capacità di mostrare il colore a 16 bit. Il rosso e il blu hanno 5 bit di risoluzione ciascuno (32 livelli di rosso e blu), il verde, dispone di 6-bit di risoluzione (64 livelli diversi).

Per coerenza con altre applicazioni, la libreria permette di inserire colori con valori di 8 bit per i canali rosso, verde e blu (0-255) e bilancia i colori in modo appropriato.

Sintassi: *screen.background(red, green, blue);*

Parametri:

red : int 0-255
green : int 0-255
blue : int 0-255
Esempio:

```
#include <Esplora.h>  //Libreria Arduino Esplora
#include <SPI.h>      //Libreria SPI
#include <TFT.h>      //Libreria Arduino LCD TFT
Void setup() {
  // inizializza lo scermo
  EsploraTFT.begin();
  // Imposta lo sfondo Bianco
  EsploraTFT.background(255, 255, 255);
  delay (1000);
  // Imposta lo sfondo nero
  EsploraTFT.background(0, 0, 0);
  delay (1000);
  // Imposta lo sfondo rosso
  EsploraTFT.background(255, 0, 0);
  delay (1000);
```

```
  // Imposta lo sfondo giallo
  EsploraTFT.background(255, 255, 0);
  delay (1000);
  // Imposta lo sfondo verde
  EsploraTFT.background(0, 255, 0);
  delay (1000);
  // Imposta lo sfondo ciano
  EsploraTFT.background(0, 255, 255);
  delay (1000);
  // Imposta lo sfondo blu
  EsploraTFT.background(0, 0, 255);
  delay (1000);
  // Imposta lo sfondo magenta
  EsploraTFT.background(255, 0, 255);
  delay (1000);
}
void loop() {
}
```

EsploraTFT.stroke()

Chiamato prima di disegnare un oggetto sullo schermo, imposta il colore delle linee e dei bordi intorno alle forme.

stroke () si aspetta valori a 8 bit per ognuno dei canali rosso, verde e blu, ma lo schermo non visualizza con questa fedeltà. Vedere comando EsploraTFT.background ()

Sintassi: *EsploraTFT.stroke(red, green, blue);*

Parametri:

red : int 0-255
green : int 0-255
blue : int 0-255

EsploraTFT.noStroke

Dopo aver eseguito questo comando, eventuali forme disegnate sullo schermo non avranno un colore del tratto di contorno.

Sintassi: *EsploraTFT noStroke();*

EsploraTFT.fill()

Chiamato prima di disegnare un oggetto sullo schermo, imposta il colore di riempimento di forme e testo.

fill () si aspetta valori a 8 bit per ognuno dei canali rosso, verde e blu, ma lo schermo non

visualizza con questa fedeltà. Vedere comando EsploraTFT.background ().

Sintassi: *EsploraTF.fill(red, green, blue);*

Parametri:

red : int 0-255
green : int 0-255
blue : int 0-255

EsploraTFT.noFill

Dopo aver inseriro questo comando, eventuali forme disegnate sullo schermo non saranno riempite.

Sintassi: *EsploraTFT noFill();*

EsploraTFT.text

Scrive il testo sullo schermo nelle coordinate indicate.

Sintassi: *EsploraTFT.text(text, xPos, yPos);*

Parametri:

text : array di caratteri, il testo da scrivere sullo schermo
xPos : int, la posizione su l'asse x in cui si desidera iniziare a scrivere il testo sullo schermo
yPos : int, la posizione su l'asse y in cui si desidera iniziare a scrivere il testo sullo schermo
Esempio

```
#include <Esplora.h>  //Libreria Arduino Esplora
#include <SPI.h>      //Libreria SPI
#include <TFT.h>      //Libreria Arduino LCD TFT

void setup() {
  // inizializza lo scermo
  EsploraTFT.begin();
  // Imposta lo sfondo nero
  EsploraTFT.background(0, 0, 0);
  //Imposta il colore del testo a bianco
  EsploraTFT.stroke(255,255,255);
  //Scrive il testo sullo schermo in alto a sinistra
  EsploraTFT.text("Testo di prova!", 0, 0);
}
void loop() {
```

```
}
```

EsploraTFT.setTextSize

Consente di impostare la dimensione del testo che segue. La dimensione predefinita è "1". Ogni variazione di dimensione aumenta il testo di 10 pixel di altezza. Cioè, taglia 1 = 10 pixel, dimensione 2 = 20 pixel, e così via.

Sintassi: *EsploraTFT.setTextSize(size);*

Parametri:

size : int 1-5

Esempio

```
#include <Esplora.h>  //Libreria Arduino Esplora
#include <SPI.h>      //Libreria SPI
#include <TFT.h>      //Libreria Arduino LCD TFT

void setup() {
  // inizializza lo scermo
  EsploraTFT.begin();
  // Imposta lo sfondo nero
  EsploraTFT.setRotation(0);
  EsploraTFT.background(0, 0, 0);
  //Imposta il colore del testo a bianco
  EsploraTFT.stroke(255,255,255);
  //Altezza di default
  EsploraTFT.setTextSize(1);
  EsploraTFT.text("H=1", 0, 0);
  //Incremento dell'altezza
  EsploraTFT.setTextSize(2);
  EsploraTFT.text("H=2", 0, 10);
  //Incremento dell'altezza
  EsploraTFT.setTextSize(3);
  EsploraTFT.text("H=3", 0, 30);
  //Incremento dell'altezza
  EsploraTFT.setTextSize(4);
  EsploraTFT.text("H=4", 0, 60);
  //Incremento dell'altezza
  EsploraTFT.setTextSize(5);
  EsploraTFT.text("H=5", 0, 100);
}
void loop() {
```

```
}
```

EsploraTFT.point()

Disegna un punto nelle coordinate indicate. Il primo parametro è il valore orizzontale, il secondo è il valore verticale per il punto.

Sintassi: *EsploraTFT point(xPos, yPos);*

Parametri:

xPos : int, la posizione orizzontale del punto.
yPos : int, la posizione verticale del punto.

EsploraTFT.line()

Disegna una linea tra due punti. Utilizzare EsploraTFT stroke () per cambiare il colore prima di tracciare la linea.

Sintassi: *EsploraTFT.line (xStart, yStart, xEnd, yEnd);*

Parametri

xStart: int, la posizione orizzontale in cui la linea inizia
yStart: int, la posizione verticale in cui la linea inizia
xEnd: int, la posizione orizzontale in cui la linea si conclude
yEnd: int, la posizione verticale in cui la linea si conclude

EsploraTFT.rect()

Disegna un rettangolo sullo schermo TFT. EsploraTFT.rect() accetta 4 argomenti, i primi due sono l'angolo superiore sinistro della forma, gli ultimi due sono la larghezza e l'altezza della forma.

Sintassi: *EsploraTFT.rect(xStart, yStart, width, height);*

Parametri

xStart: int, la posizione orizzontale in cui iniziare il rettangolo
Ystart: int, la posizione verticale in cui iniziare il rettangolo
width: int, la larghezza del rettangolo
height: int, l'altezza del rettangolo

EsploraTFT.width()

Riporta la larghezza dello schermo TFT in pixel.

Sintassi: *EsploraTFT.width();*

Dato fornito:

int : la larghezza dello schermo in pixel

EsploraTFT.height()

Riporta l'altezza dello schermo TFT in pixel.

Sintassi: *EsploraTFT.height();*

Dato fornito:

int : l'altezza dello schermo in pixel

EsploraTFT.circle()

Disegna un cerchio sullo schermo. Il cerchio è disegnato rispetto al suo punto centrale, il che significa che il diametro totale sarà sempre un numero dispari.

Sintassi: *EsploraTFT.circle(xPos, yPos, radius);*

Parametri:

xPos: int, la posizione del centro del cerchio sull'asse x
yPos: int, la posizione del centro del cerchio sull'asse y
radius: int, il raggio del cerchio

PImage

E' la classe di base che trasferisce un'immagine bitmap caricata dalla scheda SD sullo schermo. È necessario includere la libreria SD per utilizzare PImage.

Sintassi: PImage image;

Parametri

Image: il nome dell'oggetto PImage. Ci si riferirà a questo nome successivamente per elaborare le immagini sullo schermo.

EsploraTFT.image

Disegna un'immagine dalla scheda SD allo schermo in una posizione specificata.

Sintassi: *EsploraTFT.image(image, xPos, yPos);*

Parametri:

image: l'istanza denominata di PImage.
xPos: int, posizione sul l'asse x per iniziare a disegnare
yPos: int, posizione sulla asse y per iniziare a disegnare

EsploraTFT.loadImage()

Carica un file immagine dalla scheda SD in un'istanza denominata da PImage.

Sintassi: *EsploraTFT.loadImage(name);*

Parametri:

Nome: array di caratteri, il nome dell'immagine dalla scheda SD che si desidera leggere

PImage.height

La funzione fornisce l'altezza dell'oggetto PImage.

Sintassi: *EsploraTFT image.height();*

Dato fornito:

int: altezza dell'immagine in pixel

EsploraTFT.width

La funzione fornisce la larghezza dell'oggetto PImage.

Sintassi: *EsploraTFT.width();*

Dato fornito:

int : larghezza dell'immagine in pixel

PImage.isValid

La funzione controlla se il file bitmap caricato con riferimenti PImage è valido.

Sintassi: *image.isValid();*

Dato fornito:

Booleano

Programma di test per la scheda Arduino Esplora dotata di display TFT

Il programma proposto permette di testare tutti i sensori e gli attuatori presenti sulla scheda.

Il programma si trova all'interno del file allegato arduino_esplora_test.zip. [7]

Le istruzioni e i valori letti sono mostrati sul display TFT che deve essere obbligatoriamente installato.

Il programma originale è presente a questo [8]link. Questa versione è solamente un adattamento/traduzione delle istruzioni per utilizzarlo con le librerie originali Arduino.
Anche i testi presenti sul display sono stati tradotti.

Appena caricato il programma sarà mostrata una pagina di presentazione da cui scegliere il tipo di prova da effettuare.

Per fare questo occorrerà premere il relativo tasto tra quelli presenti alla destra del display.

Con il tasto 1 si testano i vari sensori, con il tasto 2 si effettua il test del led RGB, mentre il tasto 3 permette il test del buzzer.

Test 1 - Sensori

Quando si premono i pulsanti o si sposta il joystick, la parola appropriata passa dal bianco al verde. Muovendo il potenziometro lineare (slider) i valori varieranno partendo da 1023 a sinistra a 0 sulla destra.

Viene visualizzato il valore del suono rilevato dal microfono, così come il valore di sensore di luce.

Il valore della temperatura è fornito sia in gradi Celsius che Fahrenheit. Infine, sono visualizzati i valori forniti dell'accelerometro (X, Y e Z). Quando si vuole uscire, spingere verso il basso il joystick (non spostarlo, ma spingerlo con decisione verso il basso) per tornare al menu.

Test 2 - Led RGB.

Se dal menu principale si preme il tasto 2 si passa al programma di Test del LED RGB.

I valori del rosso e verde possono essere variati da 0 a 255 spostando il joystick verso

l'alto, il basso, sinistra e destra. Mentre per regolare i valori del blu si dovrà variare la posizione del cursore del potenziometro lineare. Per uscire, premere di nuovo verso il basso il joystick.

Test 3 - Buzzer

Infine, per testare il buzzer selezionare il tasto 3 del menu principale. Si dovrà spostare il cursore del potenziometro lineare per regolare la frequenza. Partendo da frequenze basse con il cursore tutto a sinistra a frequenze più alte spostando il cursore verso destra.

Se non si sposta il cursore il suono si arresterà dopo 5 secondi, per arrestarlo immediatamente premere il tasto 1. Premere di nuovo il joystick verso il basso per accedere al menu.

Programma di test della memoria SD

Un altro programma presente negli allegati (arduino_esplora_test.zip [7]) permette di testare la possibilità di lettura di file immagine bitmapped da mostrare poi sul display TFF.

Per utilizzare il programma occorrerà avere una memoria mini-SD con una capacità da 1 MB, su cui memorizzare i file .bmp allegati.

Si inserirà quindi la memoria nell'apposito alloggiamento presente sul display, nella parte superiore, lato presa USB.

Una volta caricato il programma saranno mostrati ciclicamente i file sul display, l'accensione del led RGB in verde confermerà il riconoscimento corretto del file, l'accensione del led di colore rosso segnalerà un'anomalia.

Filmato illustrativo

Conclusioni

Fatta questa panoramica, ora potrete cominciare ad utilizzare la scheda per poterne saggiare le potenzialità. Per il futuro, nei prossimi articoli sarà presentato un piccolo robot che sarà comandato in remoto tramite questa interfaccia.

Documentazione completa

La documentazione è disponibile, a questo link: arduino_esplora_test.zip. [7]

Sono presenti due cartelle, in una sono contenuti i due programmi di test descritti nell'articolo e mostrati nel filmato: *Esplora_test_TFT.ino*, *EsploraTest_new.ino*, nella

seconda cartella, sono presenti i dieci files immagine da salvare nella memoria SD.

URLs in this post:

[1] Arduino: **https://it.emcelettronica.com/arduino**
[2] Scopriamo la nuova scheda Arduino Esplora:
https://it.emcelettronica.com/scopriamo-nuova-scheda-arduino-esplora
[3] joystick: **https://it.emcelettronica.com/open-source-game-boy-con-arduino**
[4] http://arduino.cc/en/Main/Software: **http://arduino.cc/en/Main/Software**
[5] http://arduino.cc/en/Reference/EsploraLibrary:
http://arduino.cc/en/Reference/EsploraLibrary
[6] http://arduino.cc/en/Reference/TFTLibrary:
http://arduino.cc/en/Reference/TFTLibrary
[7] arduino_esplora_test.zip.:
https://it.emcelettronica.com/files/arduino_esplora_test.zip
[8] questo : **http://21stdigitalhome.blogspot.it/2013/02/arduino-esplora-with-display-exercise.html**