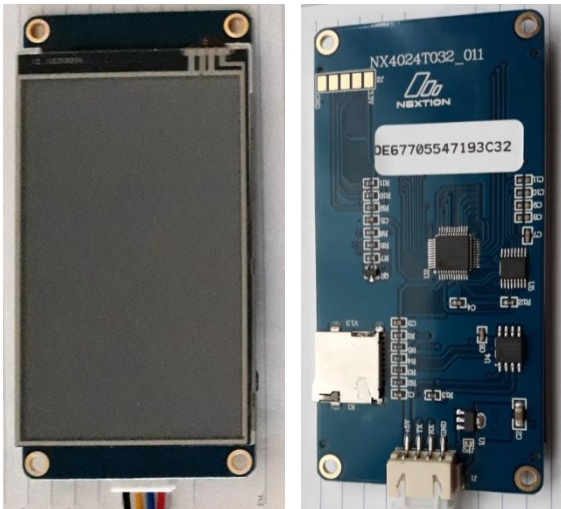


56 il display Nextion – the Nextion display – (some notes at section end)



Il Nextion e' un display touch screen di tipo TFT, dotato di un microprocessore per la gestione della parte grafica. Comunica con l'esterno attraverso una connessione seriale a 9600 baud, per cui puo' essere collegato a qualunque microcontrollore.

Sotto l'aspetto operativo l'interfaccia viene disegnata tramite un editor operante su di un PC e sul PC provata e compilata. Il file prodotto in compilazione viene poi copiato su di una scheda microSD da inserire nel Nextion. Al momento dell'accensione del display, il contenuto della scheda microSD si auto installa e da quel momento (dopo aver tolto l'alimentazione, estratta la scheda microSD e riattivata l'alimentazione), il Nextion e' pronto ad interagire con il microcontrollore.

Due parole sul Nextion

I display Nextion sono touchscreen e rappresentano una soluzione per monitorare e controllare i processi poiche' possono sostituire monitor, tastiere e pulsanti. Esistono diversi moduli Nextion, la cui dimensione varia da 2.4 pollici fino a 7 pollici. Ogni Nextion e' dotato di un microcontrollore ARM che controlla il display e si occupa di generare i pulsanti, creare testo, memorizzare immagini o modificare lo sfondo. Per disegnare la parte grafica, si utilizza il Nextion Editor, che consente di aggiungere all'interfaccia pulsanti, indicatori, barre di avanzamento, etichette di testo e altro.

In questo progetto si utilizza il Nextion base da 3,2 pollici, che e' la dimensione piu' utilizzata e per la quale e' piu' facile trovare esempi open source.

Il Nextion editor

Per lavorare con il Nextion bisogna innanzi tutto installare il Nextion editor, **versione LTS**, gratuitamente scaricabile da qui: <https://nextion.ithead.cc/> selezionando Resources > Download > Nextion Editor LTS.

Questo progetto

Il progetto si concretizza nella creazione di un'interfaccia grafica composta da due schermate. Nella prima si gestisce l'accensione, lo spegnimento e la luminosita' di un led gestito da Arduino e nella seconda si espone la temperatura ambientale, rilevata da un sensore di tipo LM35, collegato sempre ad Arduino. In pratica Arduino gestisce il sensore ed il led mentre il Nextion offre i mezzi (pulsanti e cursori) per pilotarli e visualizzarne i dati.



Qui il filmato del progetto: [Arduino ed il Nextion](#)

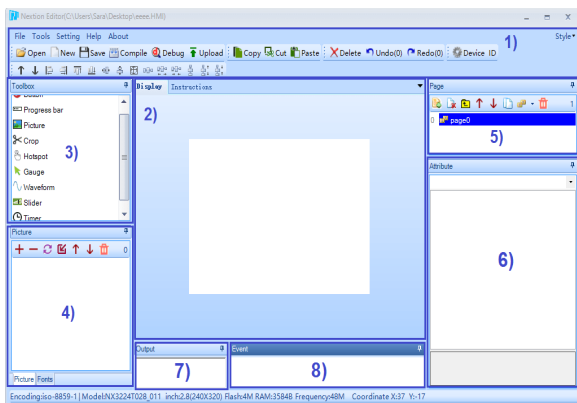
Arduino: il display Nextion – the Nextion touchscreen

In questa nota non saranno illustrate tutte le funzionalità del Nextion, ma ci si limiterà a costruire una semplice interfaccia in grado di interagire con Arduino.

Ecco le risorse coinvolte nella definizione della grafica:

- File .HMI, e' il file di progetto, il file "sorgente" generato dall'editor, che può essere richiamato e modificato;
- Immagine di sfondo (in questo caso delle immagini JPG); se il progetto prevede più interfacce grafiche possono essere utilizzate differenti immagini scelte dall'utente. Per razionalità sarebbe opportuno collocarle fisicamente nella stessa cartella in cui è presente il file .HMI e tutti gli altri file del progetto, in modo da avere un'unica cartella di progetto;
- file .TFT prodotto, su PC dalla compilazione della parte grafica e da trasferire, tramite una microSD, sul Nextion.

L'editor, informazioni generali



1 riga dei menu

2 area principale: qui si aggiungono i componenti e si crea l'interfaccia grafica

3 toolbox: qui si trovano i componenti da aggiungere (fotografie, pulsanti, cursori e altro)

4 font e fotografie da importare nel progetto

5 area delle pagine: qui si selezionano le pagine da creare o modificare

6 attributi: qui sono evidenziati e modificati gli attributi di ogni componente grafico utilizzato nel progetto

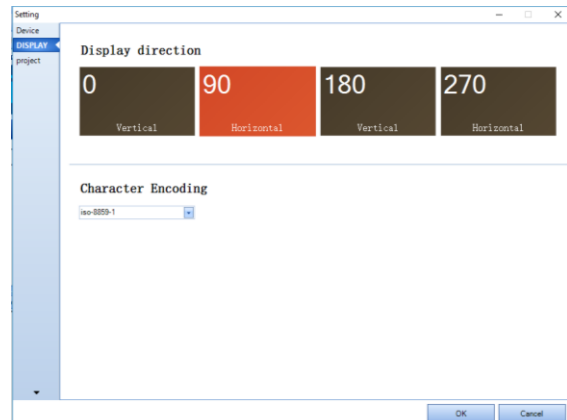
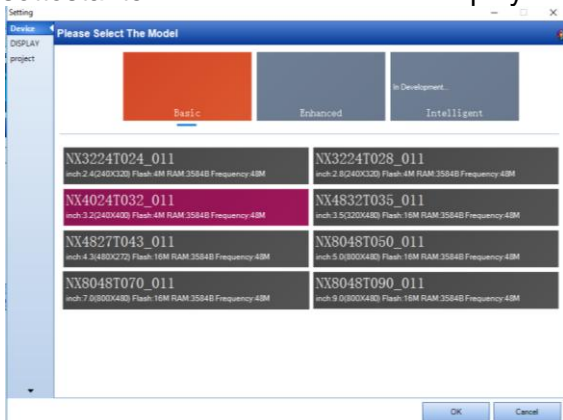
7 finestra di esposizione degli errori di compilazione

8 finestra degli eventi: qui possono essere inserite istruzioni da eseguire in determinate situazioni

Creare un nuovo progetto con l'editor del Nextion

Per creare un nuovo progetto bisogna aprire l'editor, andare in file->new, creare un nuovo file, dare un nome e salvarlo.

Subito dopo il salvataggio si aprirà un menu' con il quale scegliere il modello del display (il codice del modello è stampato sul retro del display) e, cliccando su "DISPLAY" (nel menu' a sinistra) il tipo di orientamento, orizzontale oppure verticale, che si intende utilizzare. Nell'esempio sottostante è stato selezionato il display da 3,2 pollici con orientamento orizzontale.



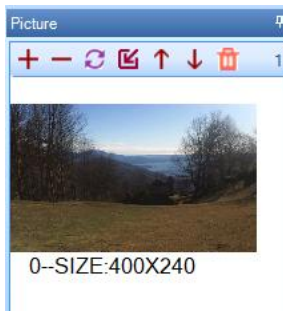
Una volta selezionato modello e orientamento, si preme OK e si torna all'editor. In alternativa è possibile aprire un progetto per il quale è già stato creato un file HMI. I progetti già creati appaiono nel riquadro "recent projects" presente nell'area di lavoro visibile all'apertura dell'editor oppure sono selezionabili tramite il menu file -> open

Aggiungere lo sfondo

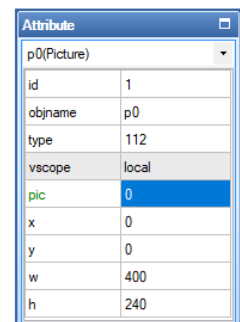
Iniziamo aggiungendo un'immagine di sfondo. Si deve disporre di un'immagine che ha le esatte dimensioni dello schermo. In questo progetto si sta usando un display da 3,2 pollici e quindi sarà necessaria un'immagine da 400 x 240 pixel (nel menu di selezione del tipo di display sono presenti anche le dimensioni in pixel).

Per creare un'immagine della dimensione voluta è possibile, ad esempio, utilizzare la funzione "ridimensiona" presente nel menu "home" di paint. A destra l'immagine che si è deciso di utilizzare come sfondo della prima pagina di questo progetto.

Una volta individuata l'immagine premere il bottone "+" presente nella zona "picture" (in basso a sinistra dell'editor) e selezionare



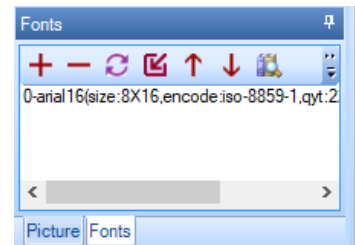
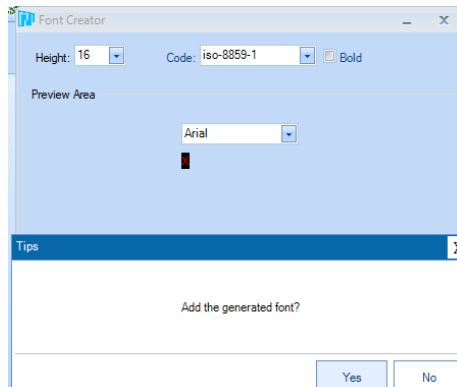
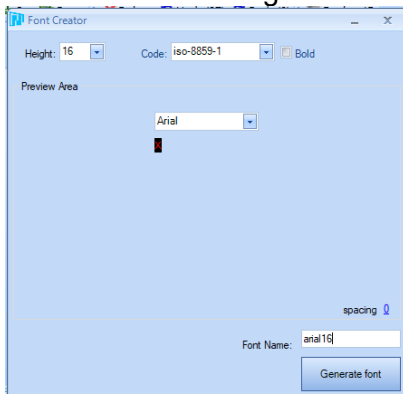
l'immagine desiderata. L'immagine sarà acquisita dall'editor e gli verrà dato un codice (in questo caso "0"). Ora si entra nel "toolbox", e si preme sul componente "picture". Verrà aggiunto un quadrato nella zona di editor e compariranno, sulla destra, gli attributi. Bisogna modificarli inserendo il punto di partenza: $x = 0$ ed $y = 0$, e le dimensioni $w = 400$ e $h = 240$. Ora, facendo doppio clic sulla zona a destra del campo "pic" (scritto in verde, nella zona attributi), appare l'elenco dei possibili sfondi e si



seleziona lo sfondo che si desidera. Si preme ok e la foto diventa lo sfondo del progetto (o di questa pagina del progetto)

Aggiungere un set di caratteri (un font)

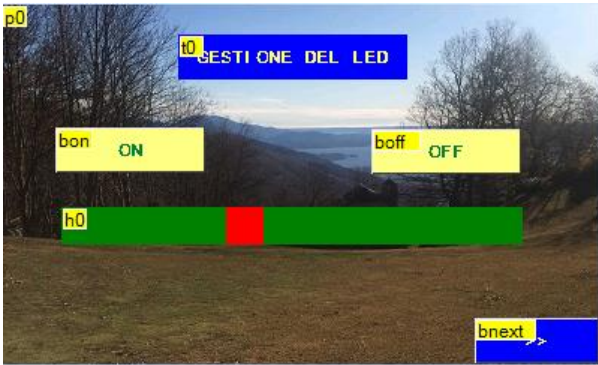
Per scrivere del testo nell'interfaccia grafica è necessario aggiungere un font all'editor. Entrando in tools->font generator (in alto sulla riga dei menu) si apre una finestra con la quale è possibile selezionare il font desiderato e le sue dimensioni. Una volta definito il font gli si dà un nome (in questo caso è stato chiamato arial16 e lo si salva. Il font sarà aggiunto alla libreria dei font, selezionabile nell'angolo in basso a sinistra.



Creare una pagina

A questo punto è possibile aggiungere al progetto i componenti desiderati, tenendo presente che tutti i componenti hanno un attributo chiamato objname e che è opportuno dare un nome comprensibile ad ogni componente.

Arduino: il display Nextion – the Nextion touchscreen



In questo progetto, per la parte che riguarda la gestione del led, sono stati utilizzati, oltre allo sfondo, cinque componenti:

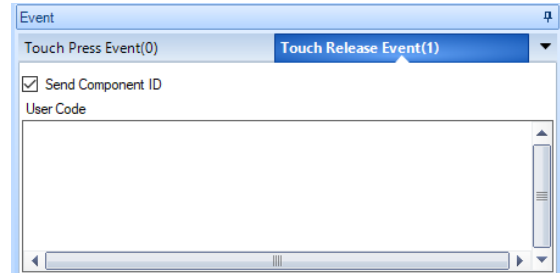
- un testo, chiamato “t0”, che conterrà il titolo, da scrivere nel campo *txt* degli attributi
- tre bottoni, uno ON ed uno OFF rinominati “bon” e boff” per accendere e spegnere il led, ed un terzo bottone chiamato “bnext” per accedere alla pagina successiva
- un cursore, chiamato “h0”, per la gestione dell’intensita’ luminosa del led

Nell’aggiungere i componenti bisogna definire o comunque verificare gli attributi poiche’ ad esempio, se si aggiunge un cursore e’ opportuno dargli 255 come valore massimo (*maxval*), che e’ il valore massimo gestibile da una porta PWM di Arduino.

Per quanto riguarda i bottoni ed i cursori, bisogna anche istruire il display circa il momento in cui inviare un segnale ad Arduino. Per inviare un segnale ad Arduino (che sara’ poi il codice del bottone premuto) bisogna selezionare il bottone e poi, nella zona “*event*” il *touch release event*, e riempire la casella “*send componet ID*”.

In questo modo il display mandera’ ad Arduino un segnale nel momento in cui si togliera’ il dito dal bottone.

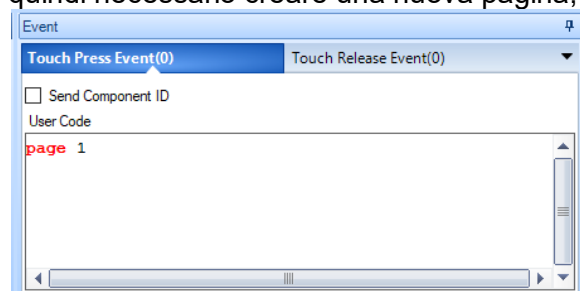
Attribute	
h0(Slider)	
id	3
objname	h0
type	1
vscope	local
mode	horizontal
sta	solid color
psta	solid
bco	34784
pco	63488
wid	25
hig	25
val	100
maxval	255
minval	0
x	17
y	179
...	...



*Nella figura sopra e’ riportata la zona “event” riferita ad un pulsante (con la zona *send componet ID* opportunamente attivata) mentre a sinistra e’ riportata la zona attributi riferita al cursore. Notare il *maxval* settato a 255*

Creare una nuova pagina e navigare

Oltre a gestire un led, questo progetto prevede anche di visualizzare la temperatura ambientale rilevata da un sensore LM35 collegato ad Arduino. E’ quindi necessario creare una nuova pagina, dedicata al sensore di temperatura. Per crearla e’ sufficiente premere l’icona “add” nella zona “page”, nell’angolo in alto a destra. Premendo l’icona viene creata una nuova pagina cui viene dato il nome page 1; per navigare da una pagina all’altra si utilizza un bottone che e’ stato chiamato “bnext” posizionato nell’angolo in basso a destra della pagina 0. Per attivarlo bisogna aggiungere il comando *page 1* nella zona eventi di detto bottone (vedi immagine a destra).



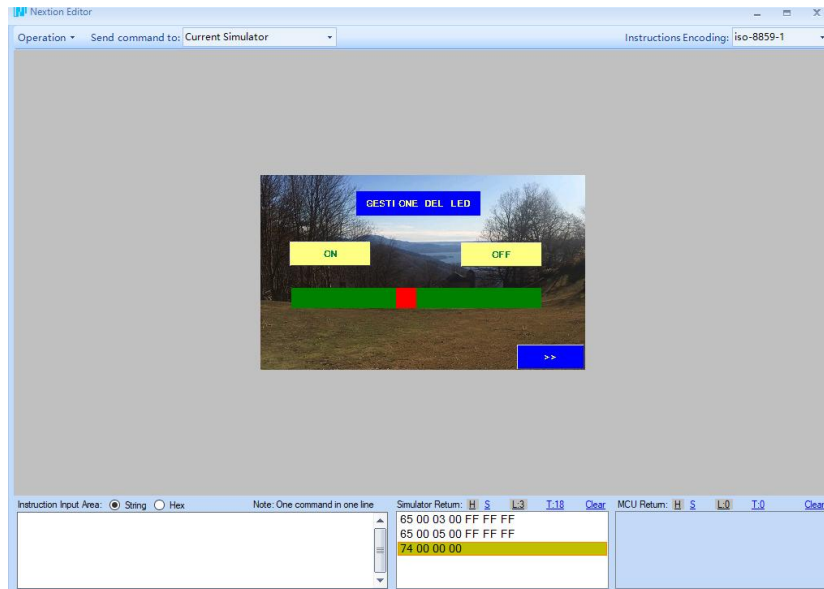
La nuova pagina deve essere disegnata con le medesime modalita’ utilizzate per la creazione della precedente pagina. A sinistra l’immagine della nuova pagina, con i campi necessari alla esposizione della temperatura ambientale.

Il campo temp, nel quale deve essere esposta la temperatura rilevata dal sensore LM35, deve essere dichiarato semplicemente come un campo testo. Sara’ poi il Nextion, a riempirlo con i dati ricevuti da Arduino

Provare il sistema usando il simulatore

Una volta disegnata l'interfaccia e' possibile provarla usando il simulatore, attivabile premendo il tasto **Debug** presente nella riga dei menu'.

Premendo il tasto Debug si apre una nuova finestra nella quale appaiono le interfacce, che possono essere "navigate" ed utilizzate premendo i bottoni e facendo scorrere il cursore. Il debug consente anche di verificare, nella finestrella in basso sulla destra, cio' che il Nextion invia ad Arduino quando si preme un bottone o si fa scorrere il cursore.



Compilare e caricare il progetto sul Nextion

Per trasferire il progetto sul display e' necessario:

- compilare il progetto, premendo il tasto **Compile** presente nella riga dei menu';
- inserire nel PC una scheda microSD formattata in FAT32
- premere il tasto "open build folder" del menu' File per aprire la cartella contenente il file .tft del progetto
- copiare il file ed incollare il file .tft nella scheda microSD
- inserire la scheda microSD nel Nextion e poi alimentarlo
- appariranno sul NEXTION alcuni messaggi che informano sull'avanzamento del caricamento del programma ed alla fine la dicitura **Check data... 100%** seguita da **Update Succeeded!**
- Togliere l'alimentazione dal Nextion, togliere la scheda microSD e ridare alimentazione. Se tutto e' andato bene sul nextion dovrebbe apparire l'interfaccia del progetto.

Scrivere il programma per Arduino

Scrivere un programma che interagisca con il Nextion e' meno complesso di quanto sembra. Ci si puo' fare un'idea esaminando gli esempi presenti nella libreria del Nextion installata su Arduino (vedere piu' sotto le modalita' di installazione della libreria). Nella libreria sono presenti esempi che comprendono sia il progetto da caricare sull'editor che il programma da caricare su Arduino. Gli esempi sono accessibili andando in: documenti > Arduino > libraries > ITEDALIB_Arduino_Nextion > examples

In ogni caso, al di la' degli esempi, la prima cosa da fare e' predisporre una tabella nella quale siano presenti tutti gli elementi che devono poi interagire con Arduino. Questa la tabella di questo progetto:

Arduino: il display Nextion – the Nextion touchscreen

Objname	Tipo	pagina	Numero ID
bon	pulsante	0	3
boff	pulsante	0	4
h0	cursore	0	5
temp	testo	1	4
bupdate	pulsante	1	5

Nello scrivere il programma bisogna tenere presente che l'unica istruzione da inserire nella zona di loop e': `nexLoop(nex_listen_list);` e che ogni altra routine e' lanciata dal nextion, tramite le dichiarazioni presenti nella zona di setup. Ad esempio la dichiarazione:

`bon.attachPop(accendiled, &bon);` presente nel "setup" lancia la routine "accendiled" nel momento in cui viene premuto, sul nextion, il tasto "bon".

Bisogna quindi progettare il programma tenendo presente che ogni attivita' deve essere eseguita o richiamata da routine lanciate dal nextion.

Una volta scritto o copiato il programma, bisogna ricordarsi di togliere l'alimentazione al Nextion quando si carica il programma su Arduino poiche' sia il Nextion che il PC utilizzano contemporaneamente la connessione seriale di Arduino e potrebbero crearsi interferenze

Prima di compilare il programma bisogna, scaricare ed installare, se non gia' fatto la libreria Nextion per Arduino, reperibile qui: [Nextion library for Arduino – ITEADLIB Arduino Nextion](#)

Per installare la libreria e' necessario:

- disimpaccare il file scaricato
- rinominare la cartella da: ITEADLIB_Arduino_Nextion-master a: ITEADLIB_Arduino_Nextion
- copiare la cartella nel repository delle librerie di Arduino (documenti->arduino->libraries)
- riaprire (qualore fosse gia aperta) l'IDE di Arduino

La libreria non e' configurata per Arduino UNO. Per utilizzarla bisogna:

- aprire, in libreria, la cartella ITEADLIB_Arduino_Nextion
- trovare ed aprire il file `NexConfig.h` (per aprirlo e' possibile usare il blocco note)
- commentare la riga 27 (deve diventare: `//#define DEBUG_SERIAL_ENABLE`)
- commentare la riga 32 (deve diventare: `//#define dbSerial Serial`)
- modificare la riga 37 in `#define nexSerial Serial`
- salvare il file `NexConfig.h`

Questo il risultato finale:

```
/**
 * @file NexConfig.h
 *
 * Options for user can be found here.
 *
 * @author Wu Pengfei (email:<pengfei.wu@itead.cc>)
 * @date 2015/8/13
 * @copyright
 * Copyright (C) 2014-2015 ITEAD Intelligent Systems Co., Ltd. \n
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU General Public License as
 * published by the Free Software Foundation; either version 2 of
 * the License, or (at your option) any later version.
 */
#ifndef __NEXCONFIG_H__
#define __NEXCONFIG_H__

/**
 * @addtogroup Configuration
 * @{
 */

/**
 * Define DEBUG_SERIAL_ENABLE to enable debug serial.
```

Arduino: il display Nextion – the Nextion touchscreen

```
* Comment it to disable debug serial.
*/
//#define DEBUG_SERIAL_ENABLE

/**
 * Define dbSerial for the output of debug messages.
 */
//#define dbSerial Serial

/**
 * Define nexSerial for communicate with Nextion touch panel.
 */
#define nexSerial Serial
#ifdef DEBUG_SERIAL_ENABLE
#define dbSerialPrint(a) dbSerial.print(a)
#define dbSerialPrintln(a) dbSerial.println(a)
#define dbSerialBegin(a) dbSerial.begin(a)
#else
#define dbSerialPrint(a) do{}while(0)
#define dbSerialPrintln(a) do{}while(0)
#define dbSerialBegin(a) do{}while(0)
#endif

/**
 * @}
 */

#endif /* #ifndef __NEXCONFIG_H__ */
```

Nota: Questo esercizio e questa nota sono parte di una serie che vede protagonisti Arduino ed alcuni dei componenti ad esso collegabili. Per la maggior parte degli esercizi e' anche disponibile un filmato su youtube:

- [Esercizi facenti parte della raccolta](#)
- [Filmati presenti su youtube](#)
- [Informazioni su arduino e sui componenti collegabili \(PDF scaricato nell'area di download\)](#)
- [Breve manuale di programmazione \(PDF scaricato nell'area di download\)](#)

Per eventuali chiarimenti o suggerimenti sul contenuto di questa scheda scrivere a giocarduino@libero.it

Here some notes about this project, translated by google translator



This project derives from a similar project found on web. The original project is graphically different, but it is however well documented and usefull for a possible replication. Here the link: <https://randomnerdtutorials.com/nextion-display-with-arduino-getting-started/>

Note: This project and this note is part of a series that sees, as main characters, Arduino and some of connectable components. For most projects there is also a video on youtube.

- [Projects collection](#)
- [Movies on youtube](#)
- [About Arduino and components \(italian; pdf will be downloaded in your download area\)](#)
- [Quick programming guide \(almost english; pdf will be downloaded in your download area\)](#)

For any questions or suggestions about this note (and on its english translation), please write to giocarduino@libero.it (simple words and short sentences, please)

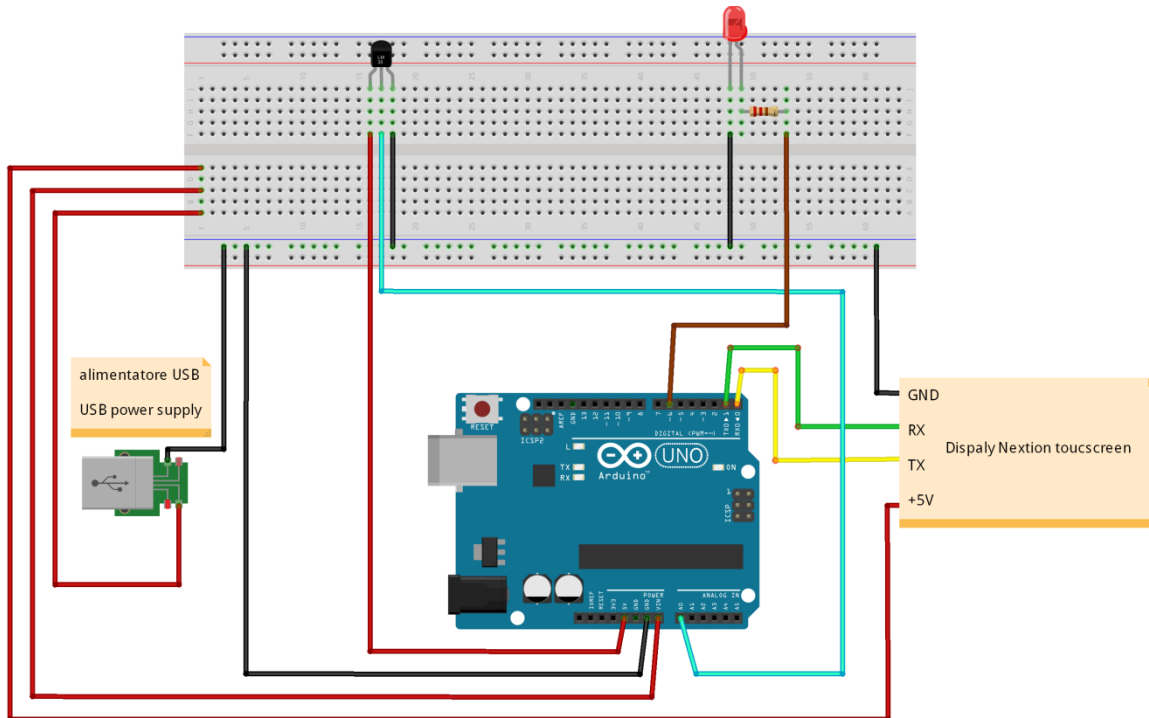
Materiali

- Un display nextion da 3,2 pollici
- Una scheda micro SD
- un led
- un sensore di temperatura di tipo lm35

Arduino: il display Nextion – the Nextion touchscreen

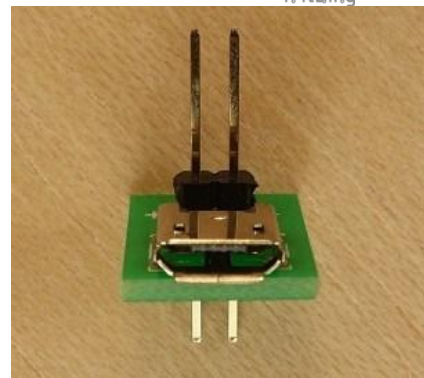
- una resistenza da 220 ohm
- Arduino UNO
- Una breadboard
- Un po' di cavetteria

Schema



Nello schema il sistema e' alimentato da un connettore USB a sua volta connesso ad una fonte di energia, come ad esempio l'alimentatore di un telefonino, in grado di erogare 5 volt ed almeno 2 ampere. A questo scopo viene fornito, insieme al Nextion, un connettore microusb con due pin che, con il connettore alimentato, sono in grado di fornire energia all'intero sistema.

Questo tipo di alimentazione (e cioe' tramite un alimentatore USB e non tramite Arduino) e' motivato dalla possibilita' che un'insufficiente alimentazione possa danneggiare il display. In realta' in questo progetto lo schermo da 3,2 pollici, associato ad altre risorse a basso consumo (un led ed un LM35), assorbono meno di quanto Arduino sia in grado di fornire per cui al di la di quanto previsto dallo schema, nella realizzazione pratica il display e' stato alimentato direttamente da Arduino. Probabilmente in caso di progetti con un maggior numero di componenti o con uno schermo piu' grande e' opportuno seguire lo schema suggerito dal costruttore.



Programma

[Download del programma per Arduino](#)
Arduino program download

[Download del programma per il Nextion](#)
Nextion program download

Arduino: il display Nextion – the Nextion touchscreen

```
/* Attenzione: facendo il copia/incolla dal PDF all'IDE si perde la formattazione del testo. Per
 * rendere piu' facilmente leggibile il programma e' opportuno formattarlo subito dopo il
 * trasferimento nell'IDE, premendo CTRL+T.
 * Warning: cut&paste from PDF to IDE loses formatting. To restore it press CTRL + T.
 */
#include "Nextion.h"
#define ledpin 6 // il led e' connesso alla porta 6
uint32_t cursore = 0; // zona per la "cattura" del valore relativo alla posizione del cursore
float temperatura = 0;
int valore = 0; // zona utilizzata nel calcolo della temperatura
char tem [6] = {0}; // zona in cui la temperatura da trasmettere al nextion
//
// dichiarazione degli oggetti nextion che interagiscono con Arduino; tra parentesi
// il numero di pagina, il numero ID e tra apici il nome dell'oggetto in Nextion
NexButton bon = NexButton (0, 3, "bon");
NexButton boff = NexButton (0, 4, "boff");
NexSlider h0 = NexSlider (0, 5, "h0");
NexText temp = NexText (1, 4, "temp");
NexButton bupdate = NexButton (1, 5, "bupdate");
//
// lista degli oggetti touch
NexTouch *nex_listen_list[] =
{
    &bon,
    &boff,
    &h0,
    &bupdate,
    NULL
};
//
/*
    routine attivata nel momento in cui viene rilasciato il pulsante "bon" - accensione del led
*/
void accendiled(void *ptr)
{
    digitalWrite(ledpin, HIGH);
}
//
/*
    routine attivata nel momento in cui viene rilasciato il pulsante "boff" - spegnimento del led
*/
void spegniled(void *ptr)
{
    digitalWrite(ledpin, LOW);
}
//
/*
    routine attivata nel momento in cui viene mosso e rilasciato il cursore - variazione
dell'intensita' luminosa del led
*/
void intensitaled(void *ptr)
{
    h0.getValue(&cursore);
    analogWrite(ledpin, cursore); // varia la luminosita' del led
}
//
/*
    routine attivata nel momento in cui viene rilasciato il tasto bupdate - aggiorna temperatura
*/
void aggiornatemperatura(void *ptr)
{
    valore = analogRead (0); // rileva la temperatura
    temperatura = (valore * 48.8758) / 100.00; // la trasforma in gradi centigradi
    dtostrf(temperatura, 6, 2, tem); //la inserisce in "tem", in formato stringa
    temp.setText(tem); // trasferisce la temperatura al Nextion
}
//
//
void setup(void)
{
    Serial.begin(9600);
    nexInit();
    // registrazione delle routine da attivate al verificarsi di un evento
    bon.attachPop(accendiled, &bon); // attiva routine accendiled al rilascio del pulsante bon
    boff.attachPop(spegniled, &boff); // attiva routine spegniled al rilascio del pulsante boff
    h0.attachPop(intensitaled); // attiva routine intensitaled al rilascio del cursore h0
    bupdate.attachPop(aggiornatemperatura, &bupdate); // attiva al rilascio di bupdate
}
//
```

Arduino: il display Nextion – the Nextion touchscreen

```
pinMode(ledpin, OUTPUT);
}
//
//
void loop(void)
{
  /* quando si preme un bottone sul Nextion la routine nexLoop (presente nella
   * libreria Nextion di Arduino), lancia la routine corrispondente al bottone premuto
   */
  nexLoop(nex_listen_list);
}
```