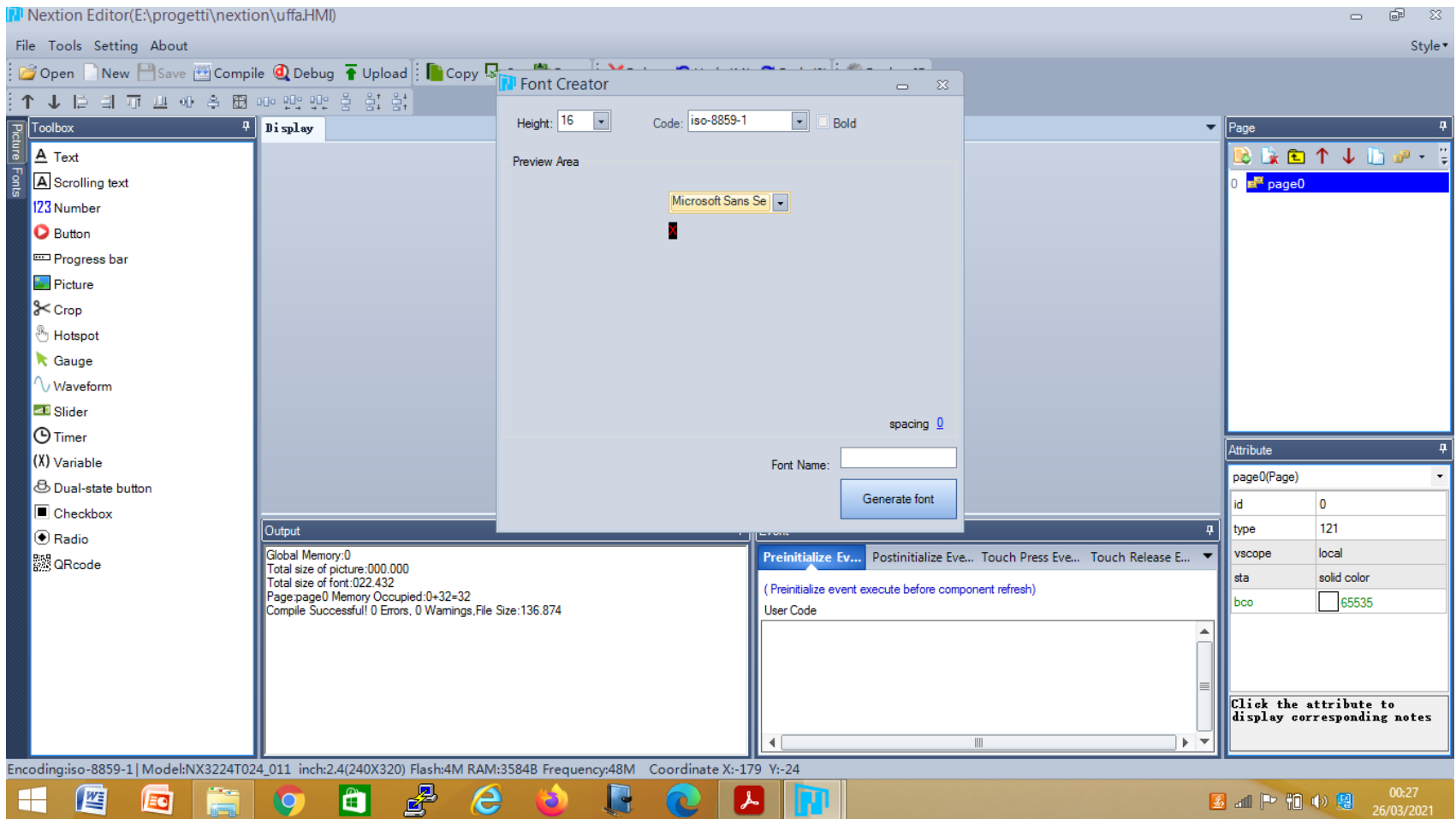


NEXTION

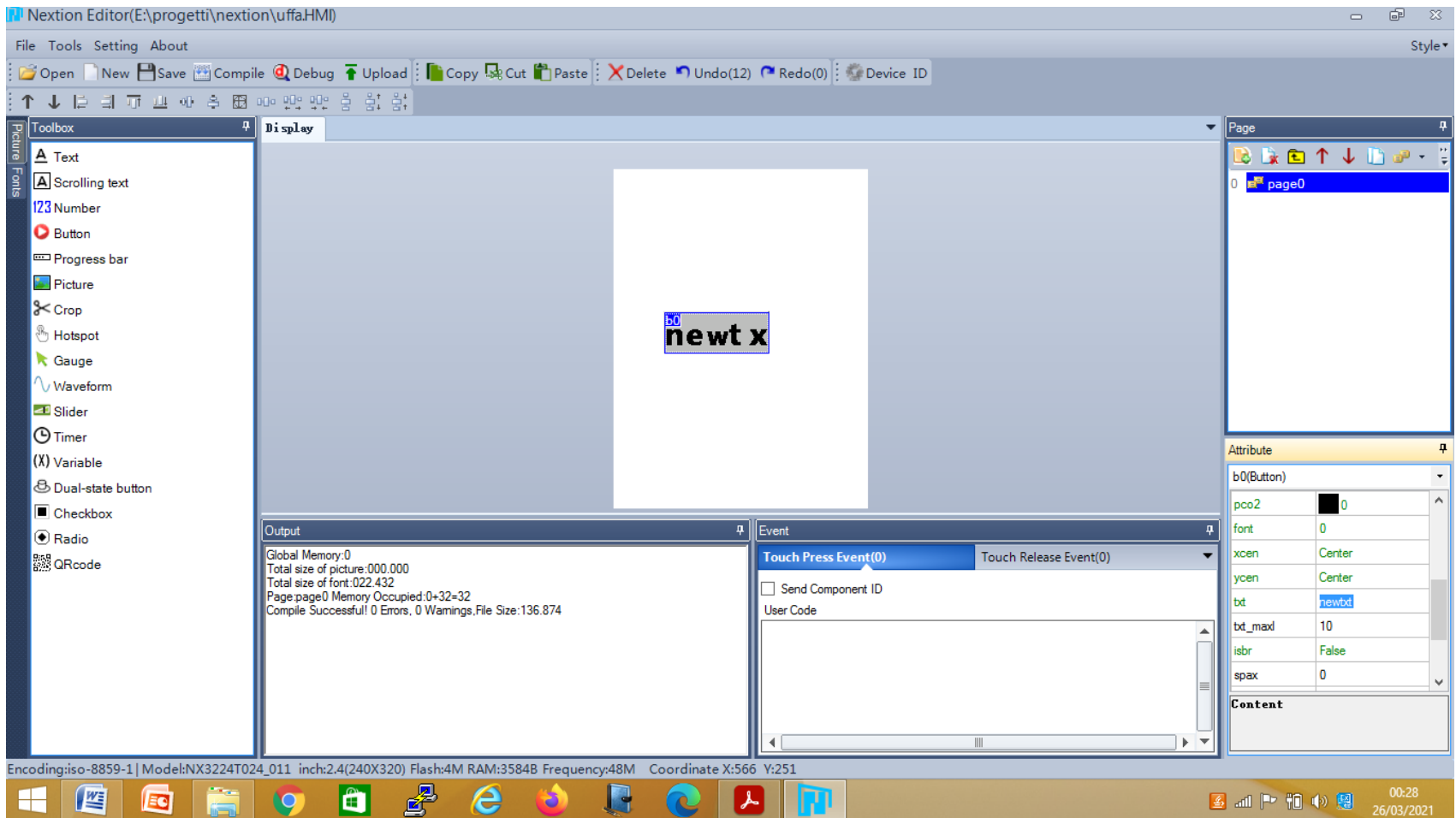
# INTRODUZIONE

- Nextion è una HMI Human Machine Interface, è una interfaccia uomo macchina molto semplice. È un touch screen con interfaccia grafica che comunica con la scheda Arduino tramite seriale a 9600 baud

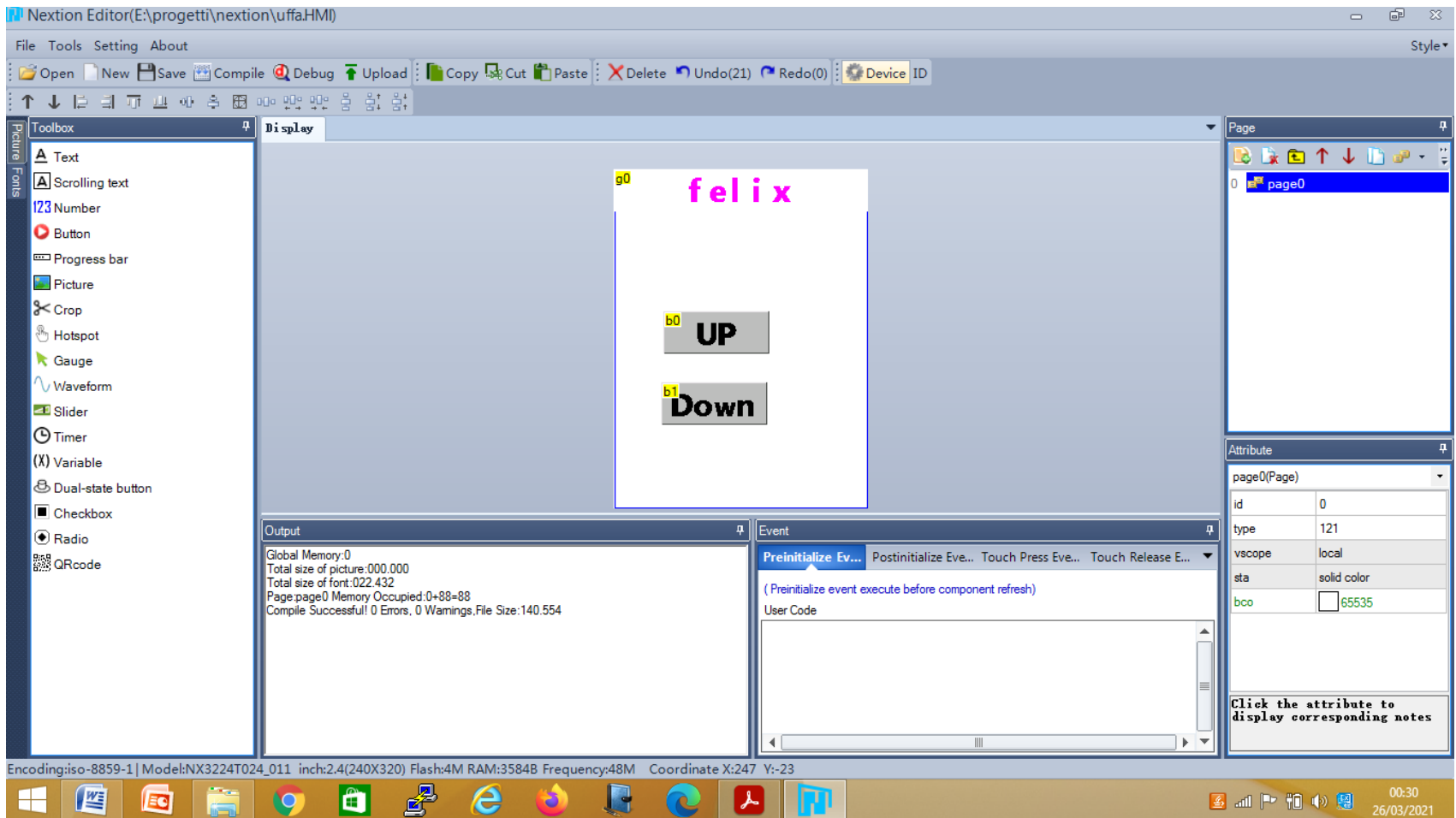
# Editor Nexion



# Inserimento di pulsanti



# Inserimento di pulsanti



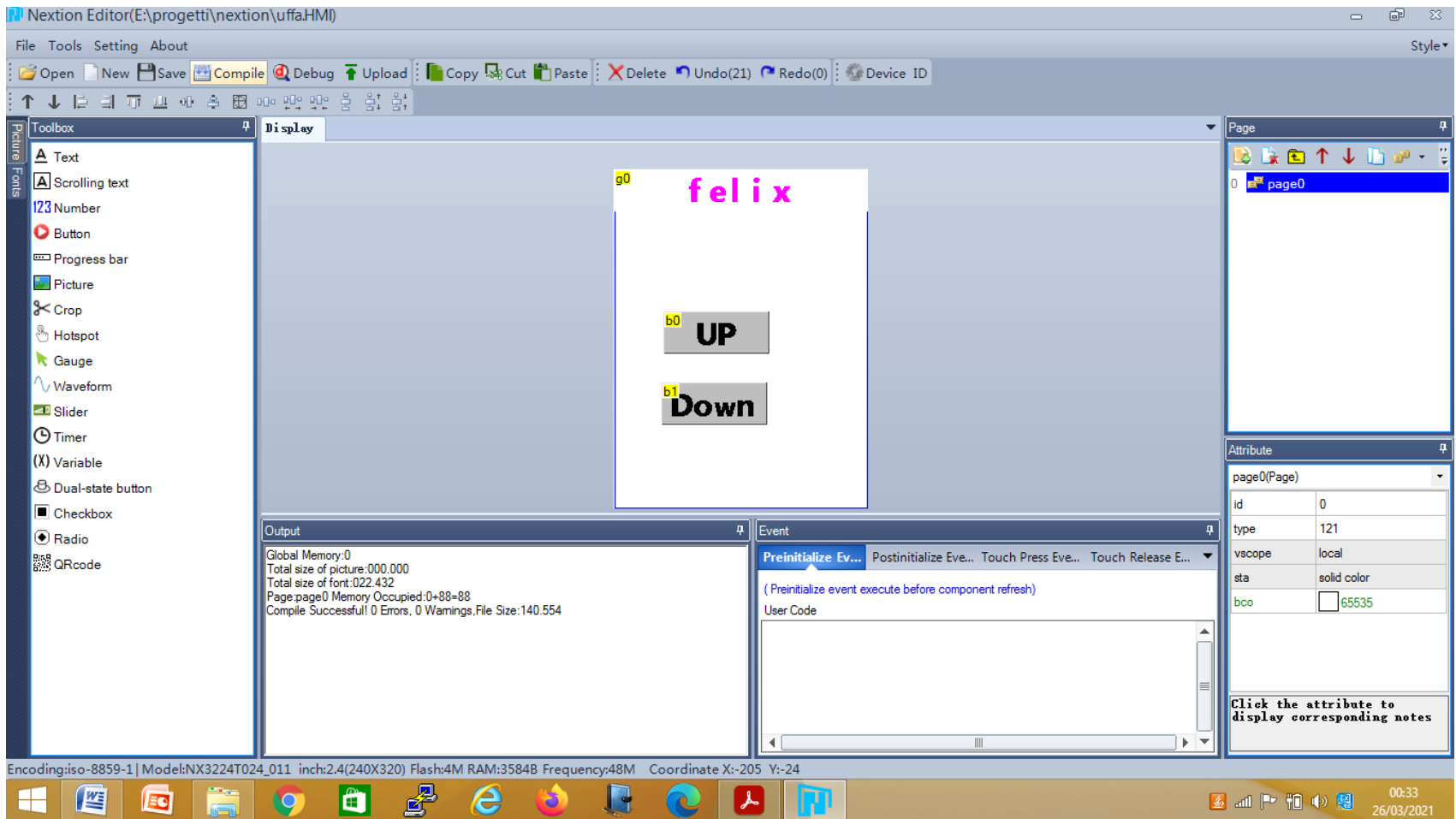
# Inserire immagine

- Si possono inserire immagini anche di sfondo con qualche attenzione.
- **IMPORTANTE!** La dimensione dell'immagine deve essere giusta per il tipo di device. Per il device base, deve essere tipo 308x180.

# Compilazione

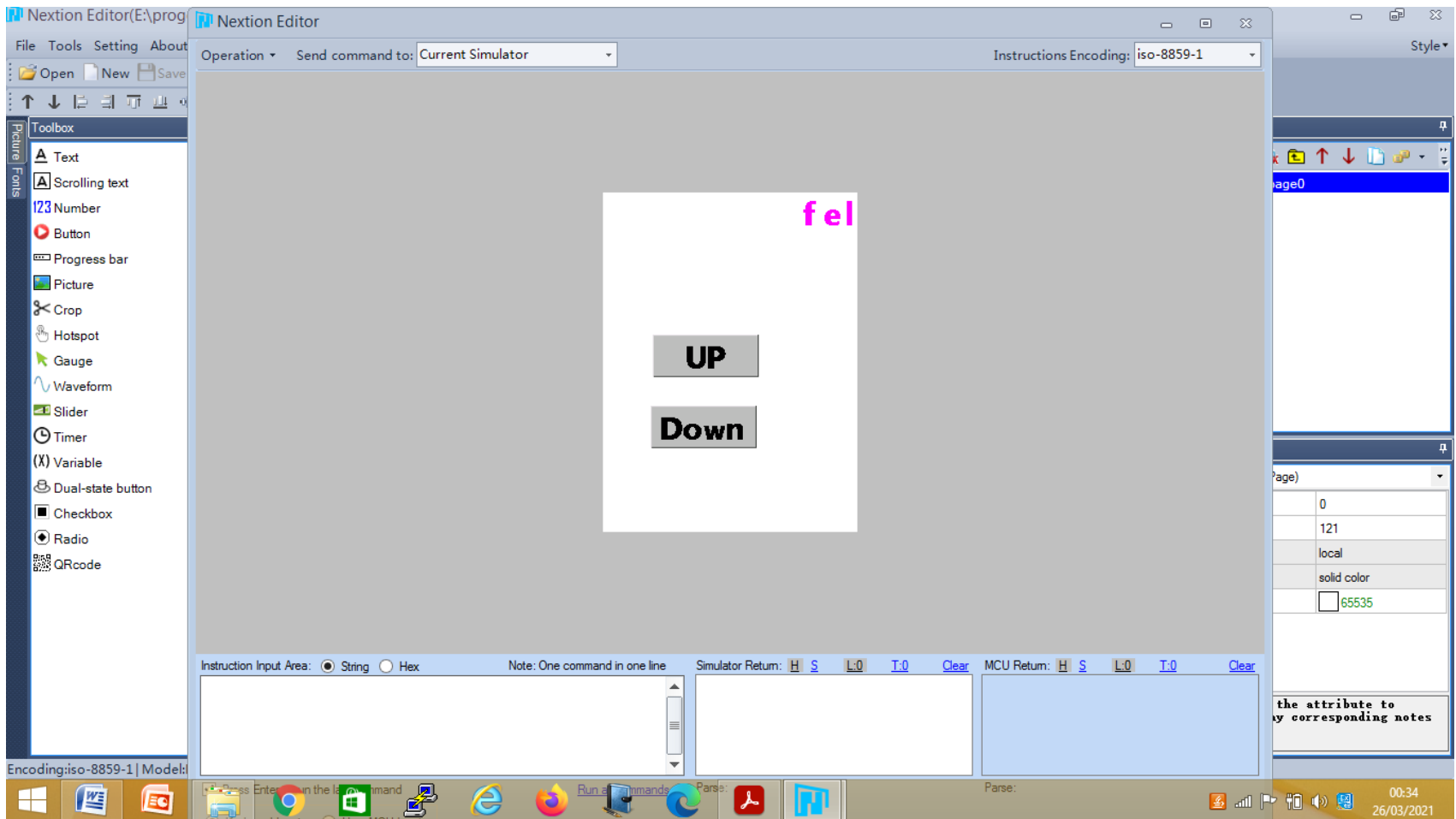
- Il file viene salvato automaticamente con estensione .hmi
- Il file va compilato pigiando sul pulsante “compile”
- Se tutto va a buon fine, verrà creato un file con lo stesso nome ma estensione .TFT
- È possibile fare anche una simulazione

# Compilazione



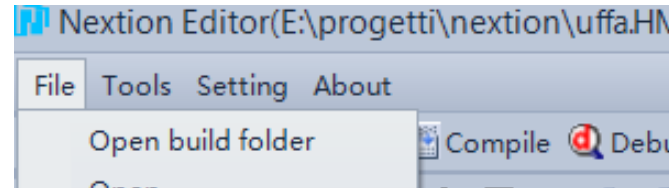
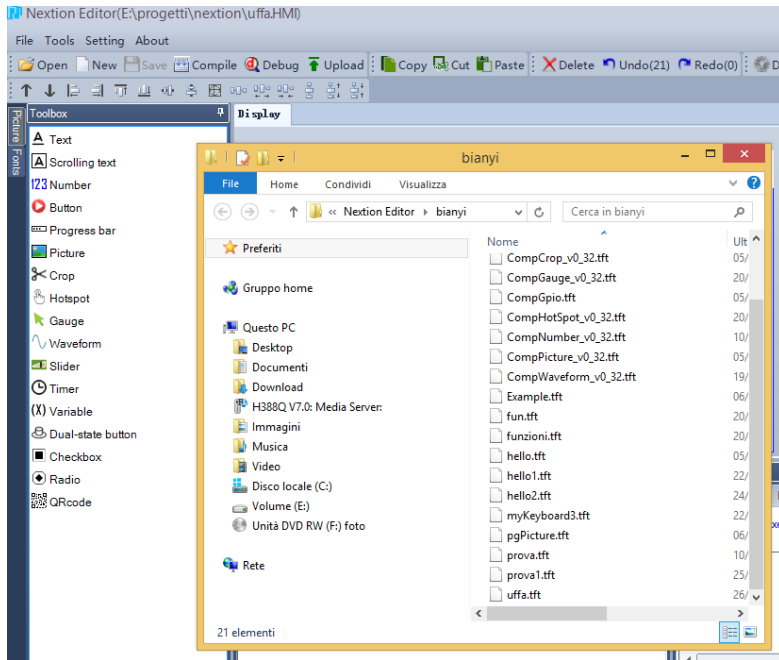


# Simulazione



# Utilizzo HMI

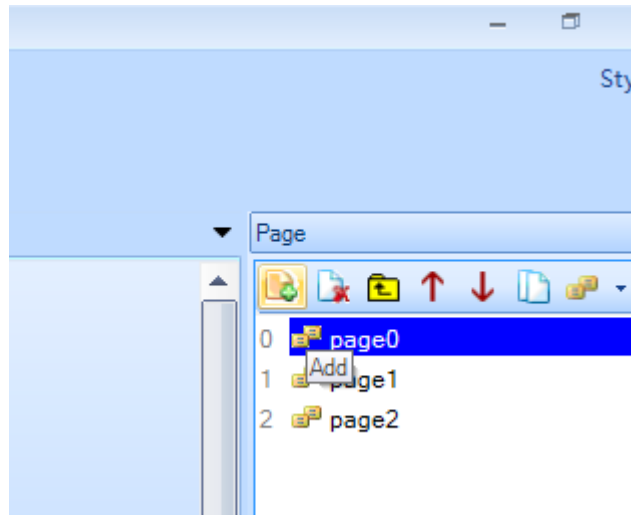
- Per poter usare l'interfaccia creata, bisogna caricare il file compilato (con estensione . TFT) in una SD card



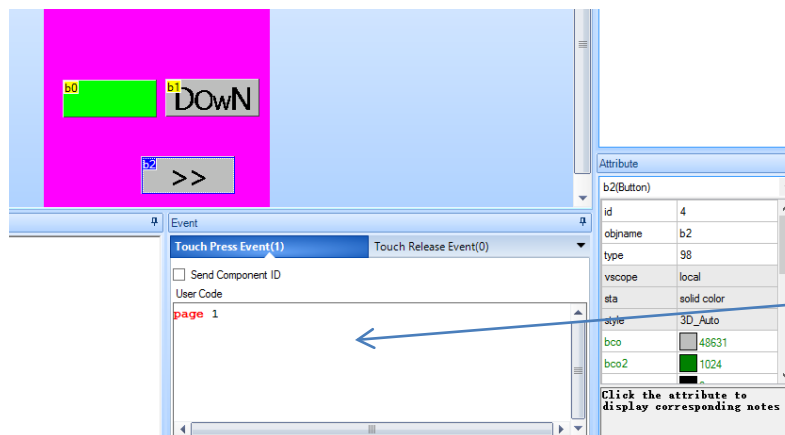
- La microSD viene inserita nello slot della Nextion e, alimentarlo da pc o tramite arduino.
- La scheda Nextion acquisisce il file se tutto funziona bene.
- Mentre il file viene caricato, sullo schermo della HMI compare la percentuale di caricamento.
- Arrivati al 100% bisogna rimuovere l'alimentazione e la microSD dal modulo nextion.

- Si alimenta la scheda Nextion con arduino e si pongono i collegamenti seriali.
- Per far funzionare tutto, bisogna passare un opportuno programma ad Arduino

# Inserire più pagine



Cambiare pagina con il pulsante



Importante è inserire il comando che fa cambiare pagina alla precedente o alla successiva

# Programmare la scheda Arduino

- Per programmare la scheda c'è bisogno di una libreria che però è stata settata solo per Arduino Mega
- . La cartella verrà scaricata con il nome ITEADLIB\_Arduino\_nextion\_master che sarà rinominata come ITEADLIB\_Arduino\_nextion.
- Nella cartella deve essere aggiunto il nome: library.properties che si trova in tutte le cartelle delle librerie relative alle diverse funzioni.

# Modifica del file di libreria

- Nella cartella deve essere aggiunto il nome: `library.properties` che si trova in tutte le cartelle delle librerie.
- Il file deve essere scritto con blocco note e deve essere un file testuale.
- Il codice è il seguente:
- ***nome = ITEADLIB\_Arduino\_Nextion***
- ***version = 0.7.0-beta***
- ***autore = Wu Pengfei***
- ***manutentore = ITEAD Studio***
- ***frase = ITEADLIB\_Arduino\_Nextion***
- ***paragrafo =***
- ***categoria =***
- ***URL di visualizzazione = https://github.com/itead/ITEADLIB\_Arduino\_Nextion***
- ***architetture = \****
-

# Modifica del file di libreria

- 
- Necessario è il cambiamento del file nexconfig.h dove vengono messe a commento le righe colorate in rosso e cambiata una riga colorata in blu
- **/\*\***
- **\* @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.**
- **\*/**

# Modifica del file di libreria

- `#ifndef __NEXCONFIG_H__`
- `#define __NEXCONFIG_H__`
- `/**`
- `* @addtogroup Configuration`
- `* @{`
- `*/`
- `/**`
- `* Define DEBUG_SERIAL_ENABLE to enable debug serial.`
- `* 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.`
- `*/`

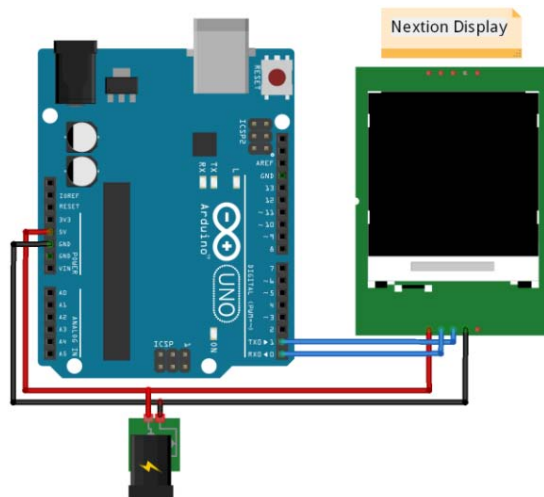
# Modifica del file di libreria

- ***#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\_\_ \*/***



# Collegamento Arduino-Nextion

- La comunicazione tra la scheda Arduino e Nextion avviene tramite i pin di comunicazione seriale di arduino
- 0 RX
- 1 TX
- Il pin RX di Nexion v'è sul pin 1 TX di arduino; il pin TX di Nexion v'è sul pin 0 RX di arduino.
- 



# Programma Arduino

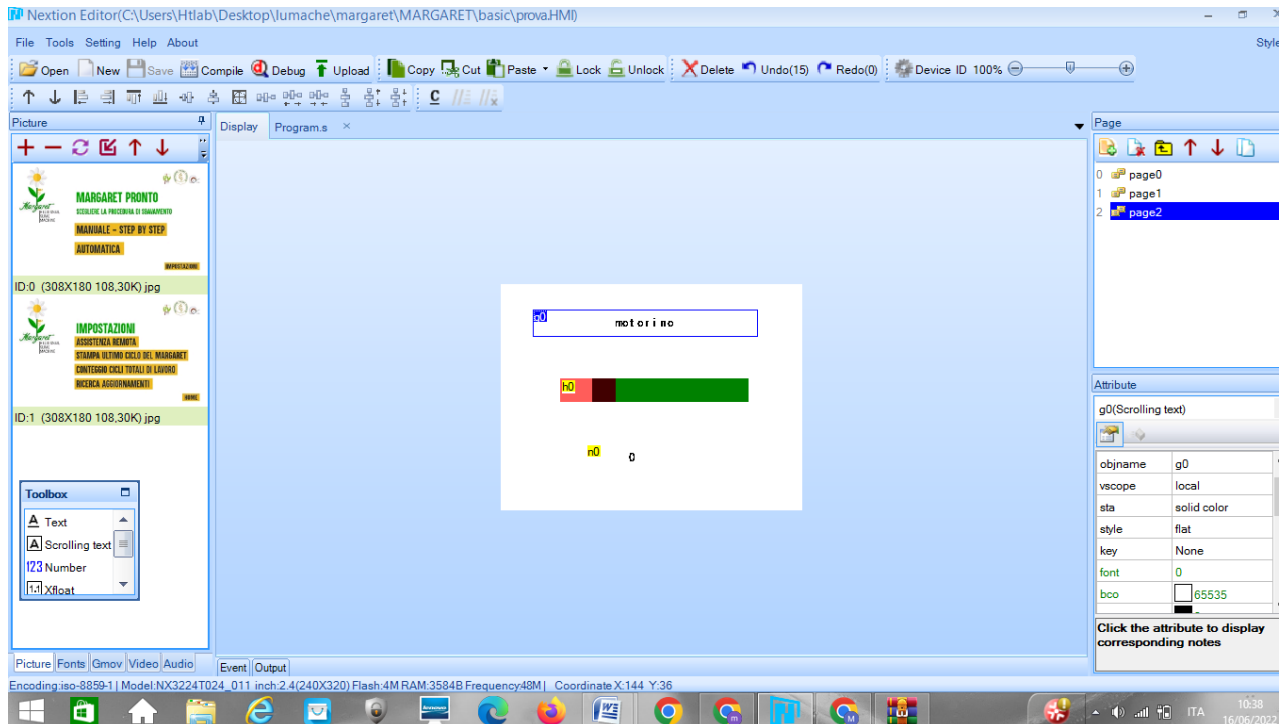
- **Esempio per accendere o spegnere un diodo LED**

```
#include "Nextion.h"
#include<NexButton.h> //se non viene riconosciuto il file, lo si includ
NexButton bOn = NexButton(0, 1, "b0");
NexButton bOff = NexButton(0, 2, "b1"); //importante è che ad ogni funzione della scheda nextion ci sia
// rispettivamente: la pagina, il numero id

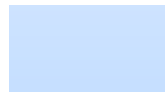
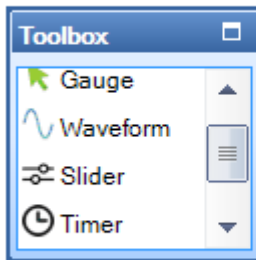
// Register objects to the touch event list
NexTouch *nex_listen_list[] = { &bOn, &bOff,NULL};
void bton(void *ptr){ // ad ogni elemento è associata una funzione
  digitalWrite(13,HIGH);}
void btoff(void *ptr){
  digitalWrite(13,LOW);}
void setup() {
  Serial.begin(9600);
  pinMode(13,OUTPUT);
  //Initialize Nextion Library
  nexInit();
  bOn.attachPop(bton,&bOn);
  bOff.attachPop(btoff,&bOff);}
void loop() {
  nexLoop(nex_listen_list);}
```

# Altre funzioni

- Scrolling text

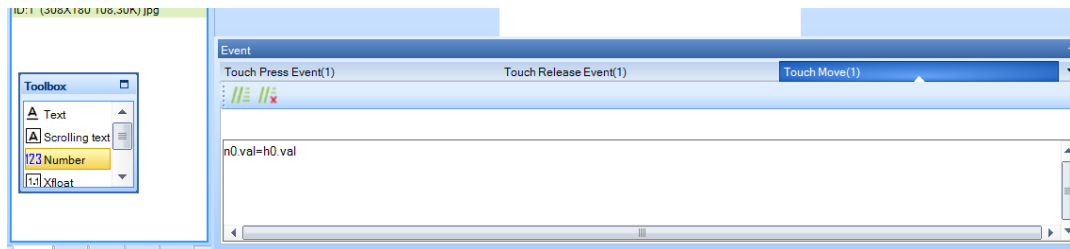


# Altre funzioni: slider



# Altre funzioni: number

- Affinchè possa comparire il valore di slider su un number:



Number è una funzione di testo come text solo è dinamica e fa comparire un number