Most of the LCD touch displays are connected to the microcontroller either using parallel connections or SPI. This is because a lot of data has to be transferred to build up just one page. And because these displays are completely “thumb”, also after a small change, the whole page has to be built from scratch. Usually, the display routine is the biggest part of the sketch and takes a lot of time to build and debug.

Recently I found an interesting new display concept to be used for Arduinos, ESPs, and other microcontrollers: The Nextion by iteadstudio.com. This is a series of touch displays which include a controller directly attached to the LCD and which can be connected to any microcontroller via a slow serial communication. And these modules are quite cheap, not much more expensive that a touch display of the same size without this feature.

So, they should solve many problems: Nice looking user interface with pictures as background, “smartphone-like” touch screen, real-time reaction to touches without real-time programming on the microcontroller, and 2 pin connection which saves the other pins for our projects. And the setup should be really easy because we can do the design part on the PC with a “what you see is what you get” editor. Sounds great! Let’s check it.

One drawback at the beginning: Unfortunately, the information for the product is not really good and it took me a while till I was able to use it properly. This is also the reason for this tutorial.

First of all, I had to get one. They offer a big variety of sizes from 2.4 to 7 inches. All should behave exactly the same with the exception of the different resolutions. I ordered only a 3.5-inch version and therefore was not able to test the compatibility of the different sizes. You find the link to my source below in the comments.

The display came in a nice box and with the necessary cable. The connector just has the 4 pins 5V, Ground, RX and TX. Communication speed is 9600 bauds. On the back, it has a slot for a micro SD card. That’s all.

To prove that the display has its own controller I start with a simple demo to show you the principle. The link for the demo file is also in the comments.

First, we have to download and install the Nextion editor on our PC. The data files for the editor have the ending \*.HMI. We open the demo file and chose the display type we want to program. Do not forget to also choose the orientation, horizontal in this case. Now, we see the first page of the PC display. On the right size we see, that this HMI file contains 7 different pages from home to “touch area”. On the left bottom we see our resources: Pictures and fonts. And on the left top we see the different tools we can use to create our display. Now let’s check-out our display first. We compile the file and start the debugger. With this debugger, we can simulate the future behavior of the display on the PC screen. Very comfortable. Now let’s go to page “gauges”. Here, we have two gauges and two buttons. Pressing these small buttons with a mouse pointer is easy. But on a 3.5 inch display with a finger, this will be impossible. This is, why we want to make them bigger: To do so, we stop the simulator and go to this page. Here, we either drag- and drop the buttons or we enter the respective values in the Attribute field.

We see an immediate reaction and can adjust till the buttons fit our needs. Just imagine this process if you would program your own design on a thumb display with an Arduino. This would be very time consuming…

Now, our buttons are better and we compile and run the simulator again. We can select the different pages and press the bigger knobs. Everything ok.

As a next step, we want to get the same result on the LCD display itself. To do so, we need a micro SD card and a card reader/writer on our PC. I use a 16GB card, but I tried also a 32 GB card. Both work fine. We close the simulator again and open the build folder. Here, we find the tft file with the same name than the HMI file. This file has to be copied to the SD card. Pay attention, that you delete all other files. The SD card must only have one tft file on it and nothing else.

Now we insert the SD card into the Nextion display. Nothing happens until you power-cycle the display. As soon as the 5V are back, the display starts to convert the tft file and to store it on the display itself. After this work is done, again, nothing happens until you power-cycle the board again. Do not forget to take the SD card out before you do so. Otherwise, you will end up in a conversion cycle again… From now on you do not need the SD card until you want to change the display’s content.

Without the SD card the display really starts with displaying page0 and you can go through the same pages like in the simulation. You see also our bigger buttons on the gauges’ page. Be aware, that we do not need any microcontroller board to do so. Just the display board. And the big data rates are transferred just from the controller to the LCD.

This is nice, but more-or-less useless for my purpose, because I want to use it with an ESP8266 or an Arduino. So, lets connect it to the microcontroller. To do that, we use a simple scenario to start with: We want a slider where the slider position is read by the microcontroller and its value displayed on the display.

To do that, we need to install the Nextion library for Arduino and search for the example “CompSlider”. We find the HMI file for the editor in the libraries folder. If you do not know the path to this folder: Just go to preferences. Here you find the path to your Arduino folder. Add \libraries and you should find the Nextion library folder (the one with the newest date).

Klick on the HMI file, select your board, and compile the file. Now we can start our simulation. If you slide the slider with your mouse, the number does not change. Why?

The number does not change because a connected microcontroller has to change its value. And for the moment, there is no microcontroller connected to our simulator. But we can emulate the microcontroller by typing in the command “t0.txt="60” into the instruction input area. After hitting return, the display shows the number 60. BTW, t0 was the name of the text field.

If we move the slider you see some text appearing on the bottom of the simulator. This text shows the data the Nextion module will send to our microcontroller when we move the slider. But where is this coming from? We close the simulator again and select the slider object. It has the name h0. If you go to the rider “Touch Release Event”, you see, that the “send component ID” is selected. This defines, that a message will be sent via the serial connection when you release the slider. If you uncheck it and run the debugger again, no text will be produced.

We are now ready to load our new file to the SD card and afterwards to the Nextion display. We use the same principle as shown before. The name of the file is CompSlider.tft.

For the moment, I use an Arduino Mega for this Tutorial because it has 2 serial connections. The Nextion Display is connected to Serial2 and we enable debug messages at the normal Serial connection.

Now we are ready to download the Arduino library from Github. I enclose the link in the comments. If we compile and run the example “CompSlider” the display works as expected. If we slide the slider, the number is changed accordingly. If we watch the serial connection, we see, that the Arduino gets the values of the slider and we can do something with it. For example, we can check if the value is above 60 and then remind the user that it is “hot”.

Many other examples come with the library which show you how to use this marvelous display. I enclose also the link to another library for the Nextion and a few links to examples that you get an impression of its possibilities.

I plan to make a second video where I will show you how to create your own display and maybe some other nice features of these displays.

Thank you for watching. I hope, this video was useful or at least interesting for you. Bye