

# MegaHurts TFT SCREEN TOUCH PANEL CALIBRATION UTILITY

## U S E R S   G U I D E

\*

### PROGRAM DESCRIPTION

\*

Alright, you just got the **PIC32 MMB** from *Mikroelektronika* and ready to get programming that awesome application you've been thinking day and night about. Well, here is a little application tool that will help you make yours all the better.

I made this project because *MikroElektronika* has not seen fit to include a similar tool in their arsenal of software or examples (*yet*), and sooner or later, you would realize like I did, there needs to be one. Now you do not have to write one, just use this application as part of your development tool set.

Every *Touch Panel*, (**TP** herein after), is unique and should be tested for proper operation and calibration. This program allows you to do that so you can set calibration constants in your project applications code that gives you the best results for each device and if being programmed as a final product, for your customers, a device that does not require **TP** calibration every time it is started and is accurate and gives stable, consistent operation.

This tool also allows you to test the accuracy of the **TP** and adjust out any instabilities due to **ADC** sensitivity because of Min/Max axis ranges that result in *uneven* division results. This problem can be seen (*if it exists*), on the accuracy testing screen (**Screen2**), by using a **TP** screen *stylus* and pressing it along the test box outline pattern and watching the **X Y** axis values being returned in real time, to see if either is fluctuating wildly between values.

Adjusting an axis's minimum or maximum range value by a small amount (**+ - 1**), at a time and retesting will allow you to "*tune*" out erratic behavior of **ADC** reads and let you concentrate on the rest of your application without worries of **TP** input operations. **Screen3** is the screen that allows you to make these *calibration* adjustments. See the "**SCREENS AND BUTTONS**" section below to learn how each part of this application tool works in detail.

You are welcome to modify this **V-TFT** project in any way you deem to try and make it a better tool or make it do the adjustments automatically.

I would request that if you do, please make it available to the community freely as I have done with this project. You may **NOT** use any part of this code in a commercial product without my permission. This code is to remain in the public domain for free usage of *MikroElektronika* customers and product users.

I hope you have found this tool as useful as I have, Robert.

-(Note:

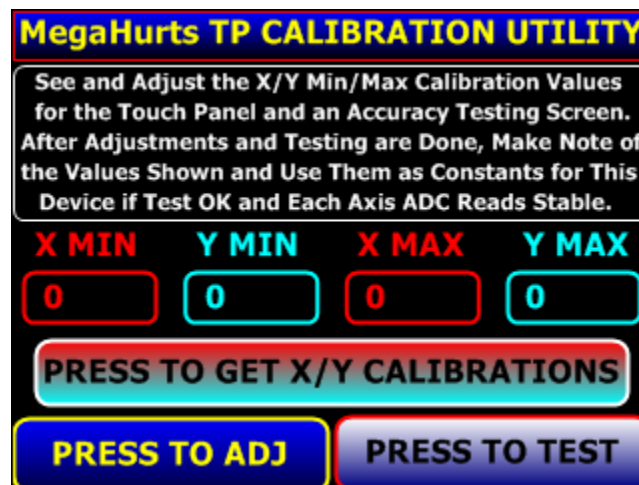
- This version uses a technique called "**self re-programming**" (**RTSP**), in order to allow
- you to try different calibration settings without needing you to reprogram
- the device with different constants each time an adjustment is needed.
- The **PIC32** has the capability to Write data of 32Bit Word lengths to **CODE** memory
- during program execution. It writes any changes to **FLASH** memory, then performs
- a **SW Reset**. Then upon start-up, gets the values from those **FLASH** memory locations
- to use as the **TFT\_TP\_calib\_Const()** during device initialization.
- You can see how this was done by looking at the code in the "**User Code**" sections
- in the "**Screen\_Cal\_Utility\_events\_code.mbas**" project module.)

\*

## SCREENS AND BUTTONS

\*

### Screen1: S T A R T U P



This is the **Main Menu** screen. There are 3 Buttons and the 4 Min/Max XY calibration values display windows below title and short instruction boxes.

\*Press the [**PRESS TO GET X/Y CALIBRATIONS**] Button to load the **X\_min**, **Y\_min**, **X\_max** and **Y\_max** TP calibration constants hard coded in the program at start-up or after a Reset. **DO THIS FIRST**. Then use tools on screens 2 and 3 to **TEST** and **ADJUST** these **values**.

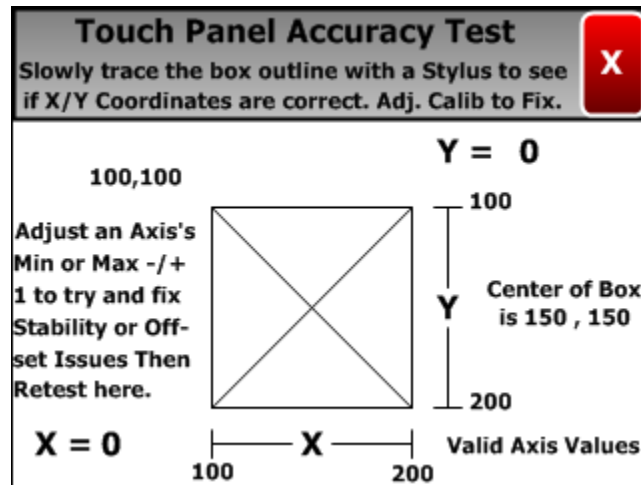
\*Press the [**PRESS TO TEST**] Button to see **Screen2** and perform TP sensitivity and accuracy testing of touch position **X\_coord** and **Y\_coord** results. See **Screen2** description section for details of use and operation.

\*Press the [**PRESS TO ADJ**] Button to see **Screen3** and adjust the calibration values for **X\_min**, **Y\_min**, **X\_max** and **Y\_max**. See **Screen3** description section for details of use and operation.

---

## Screen 2: ACCURACY TEST

---



This is the "**Working**" part of this application. This screen allows you to see and test how well any calibration values for the **X** and **Y** axis minimum and maximum settings work by displaying "*real time*" **XY** axis touch coordinates.

Use of a "**TP Stylus**" pen is recommended to get the absolute *best* results. Using your finger as the input instrument can be done, but a stylus should be used at first to ensure that the best calibration is achieved. Using your finger for input can be tested after proper testing is done using a stylus pen. This is important because of the "*foot print*" size of the input instrument used on the **TP**. For this testing, "*Smaller*" is better.

There are only 2 active touch areas on this screen - the area around the test box drawn +/- 10 *and* inside the entire box and the **EXIT** button at top right of the screen. The [**X**] Exit button returns you to the main menu (**Screen1**) screen. The test box is where you test the accuracy of the calibration settings by touching the box's outline and seeing what **XY** value results are shown. The box is 100 *pixels* by 100 *pixels* big and the top left corner is at 100x100 **XY** screen location. The screen shows details of what values you should expect to see. Make note of what values you see compared to where you are pressing the stylus pen. The **XY** coordinates are only displayed while you are pressing on the **TP**. Once you lift up off the **TP**, the program stops reading and displaying the **XY** axis position pressed and clears the values displayed and waits for the **TP** to be pressed inside active area again.

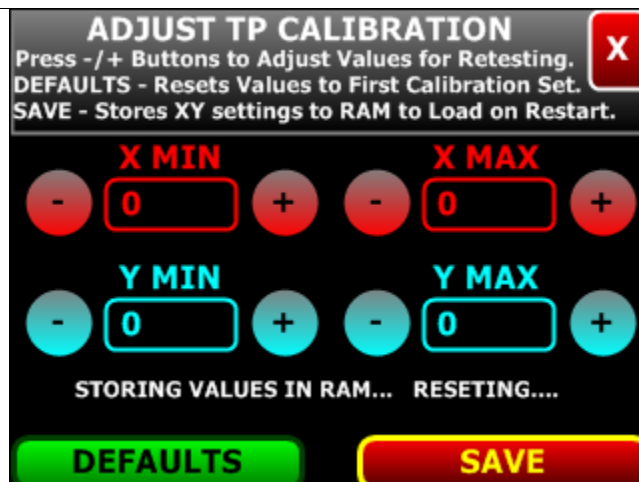
Holding the stylus *steady* and at any non-moving location is how you check for erratic **TP ADC** reads. Check this at *multiple* places inside the test area. Once you have noted the **XY** axis results and behavior, you can exit this screen and from the main menu go to the *Adjustment* screen (**Screen3**), to make any adjustments to the *calibration constants* shown there.

Which way to make any adjustments I leave up to you to decide because there are considerations due to which way the **TP** and **TFT** display are oriented. When I have done much more testing in different hardware and settings, I may do an update to this tool to make it more automatic, but for now I like the manual touch approach so I have to get better familiar with the **HW** and **SW**.

---

### Screen 3: ADJUST TP CALIBRATION

---



*\* The actual screen in the application will say ROM, not RAM. That is how it actually works. -MHs*

This screen shows the results of the manual calibration routines done on device start-up or after a reset and allows you to change any of the **TP XY** axis's minimum and maximum calibration constants by touching the *minus* (-) or *plus* (+) buttons on either side of each parameter display window.

Each press (*click*), of these buttons will either subtract or add *one* (1) to the value shown. Valid range of these numbers is limited to be between *zero* (0) and **65535** (a binary *Word*). Values are usually much lower than the maximum limit of a *Words* capacity though. As stated, each device is unique, hence the need for the calibration setting routines in *Mikroe's TP* libraries, so typical values are not listed by me or anyone for this reason.

\*[**SAVE**] Button at bottom right of this screen to put the currently displayed values in use so you can go to the *Accuracy Testing screen* (**Screen2**), and check the operation of these settings. Press the [**X**] exit button at the top right of screen to *exit* without *activating* any changes made. Once changes have been done to any calibration value, pressing [**SAVE**] button will make the **PIC32** write the values to *Code FLASH* memory so they can be retrieved after the device goes through a Software (**SW**) Reset. Once the device restarts, the new values should now be in use and can be checked by pressing the [**PRESS TO GET X/Y CALIBRATIONS**] Button on **screen1** to see *current* calibration values. Proceed to **Screen2** to test these new settings ( [**PRESS TO TEST**] Button).

\*[DEFAULTS] Button allows you to restore these values back to the *original* values at *start-up* in case you need to start over with trying to get the device calibrated and have made the wrong adjustments.

Of course, this button will only work if the adjustments you have made have not made it impossible for the **TP** library routines to determine where you have touched the **TP** now. If this is the case, a ***RE-Programming*** of the *device* is required in order to get it functioning again. Just start over and make careful, small adjustments of these values.

---

---

*Final note:*

*I have made every effort to make this as easy of a tool to use as possible.*

*It really is very easy to use and not as complicated as these directions are to read, (sorry if so).*

*There are other comments thru-out the rest of this **V-TFT** project code that you can read to see what the program is designed to do.*

*I know this is not a perfect programming example, but you are welcome to examine it anyway to see how it works and you are welcome to post any comments about it on the **Libstock** site or **Mikroe's** forum.*

*It should be very easy to adapt this application to other **HW** devices and other compiler languages. Please post notifications if you do so ok? And make it available for all to use, since you are using my work as a base for it.*

*Just to keep things fair is all. **B^)***

***MegaHurts.***