

UTouch

Arduino and chipKit Universal TFT touchscreen library

Manual



PREFACE:

This library was made to complement UTFT to provide touch screen functionality.

You can always find the latest version of the library at
<http://electronics.henningkarlsen.com/>

If you make any modifications or improvements to the code, I would appreciate that you share the code with me so that I might include it in the next release. I can be contacted through
<http://electronics.henningkarlsen.com/contact.php>.

For version information, please refer to **version.txt**.

REGARDING CALIBRATION:

All touch screens will have slight variations. It is therefore important that you calibrate your particular touch screen for the best possible performance.

To calibrate your touch screen you will need to run the UTouch_Calibration sketch supplied in the examples of the library.

Before you compile and upload the sketch there are a couple of things you must do.

1. Make sure you have uncommented the correct section for your development board
2. Make sure the UTFT display model code is correct for your display module
3. Make sure the TOUCH_ORIENTATION define is correct. You can find a list of the correct parameter for all the tested displays in the *UTouch_Supported_display_modules* PDF.

Further instructions will be given on screen when you run the sketch.

Remember that if you have more than one touch display module you may have to run the calibration on each module.

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Defined Literals:

Orientation
For use with InitTouch()
PORTRAIT: 0
LANDSCAPE: 1

Precision
For use with setPrecision()
PREC_LOW: 1
PREC_MEDIUM: 2
PREC_HI: 3
PREC_EXTREME: 4

Functions:

UTouch(TCLK, TCS, TDIN, TDOUT, IRQ);

The main class of the interface.

Parameters: TCLK: Arduino pin for Touch Clock
 TCS: Arduino pin for Touch Chip Select
 TDIN: Arduino pin for Touch Data input
 TDOUT: Arduino pin for Touch Data output
 IRQ: Arduino pin for Touch IRQ
Usage: UTouch myTouch(15,10,14,9,8); // Start an instance of the UTouch class

InitTouch([orientation]);

Initialize the touch screen and set display orientation. If the library is used together with UTFT the orientation should be set to the same orientation for both libraries.

Parameters: orientation: <optional>
 PORTRAIT
 LANDSCAPE (default)
Returns: Nothing
Usage: myTouch.InitTouch();// Initialize the touch screen

dataAvailable();

Check to see if new data from the touch screen is waiting.

Parameters: None
Returns: Boolean: true means data is waiting, otherwise false
Usage: check = myTouch.dataAvailable() // See if data is waiting

read();

Read waiting data from the touch screen. This function should be called if dataAvailable() is true. Use getX() and getY() to get the coordinates.

Parameters: None
Returns: Nothing
Usage: myTouch.read(); // Read data from touch screen
Notes: After calling read(), raw data from the touch screen is available in the variables TP_X and TP_Y. Do not use these if you do not know how to handle the raw data. Use getX() and getY() instead.

getX();

Get the x-coordinate of the last position read from the touch screen.

Parameters: None
Returns: Integer
Usage: x = myTouch.getX(); // Get the x-coordinate

getY();

Get the y-coordinate of the last position read from the touch screen.

Parameters: None
Returns: Integer
Usage: y = myTouch.getY(); // Get the y-coordinate

setPrecision(precision);

Set the precision of the touch screen.

Parameters: precision: **PREC_LOW, PREC_MEDIUM, PREC_HI, PREC_EXTREME**
Returns: Nothing
Usage: myTouch.setPrecision(PREC_MEDIUM); // Set precision to medium
Notes: Higher precision data will take longer to read, so take care when using PREC_HI or PREC_EXTREME with fast-moving input.