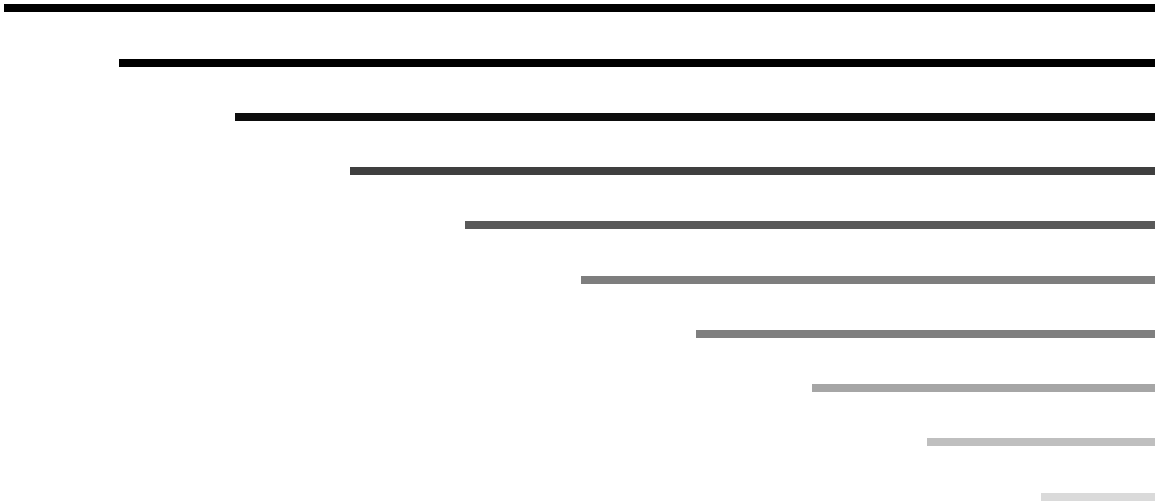


*Embedding it better...*



μTasker Document

**μTasker – Touch Screen**



## **Table of Contents**

1. Introduction.....	3
2. Example of resistive touch screen control with 2 x ADC converter channels and GPIOs..	3
3. Conclusion .....	7

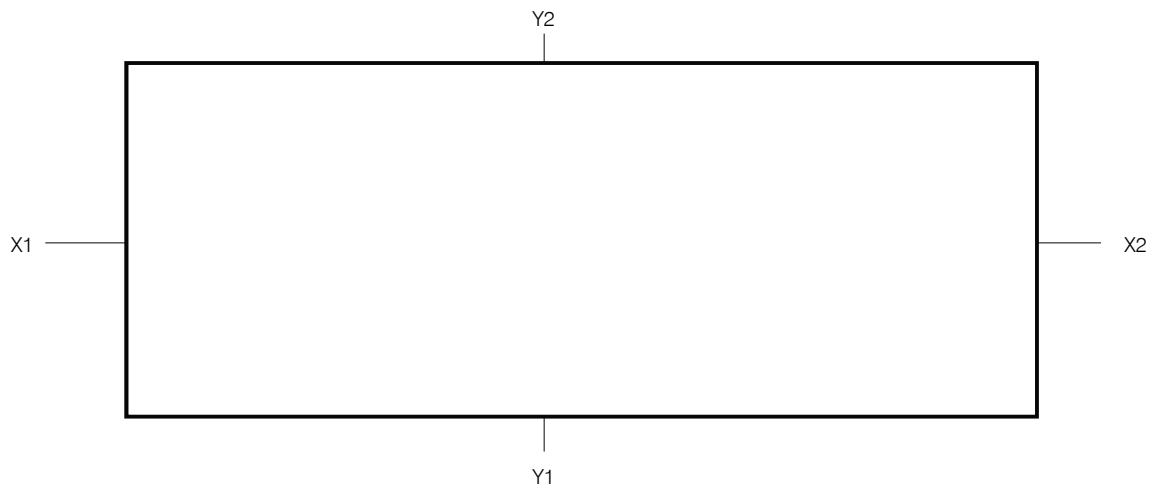
## 1. Introduction

Touch Screens have found their way into many MMIs (Man-Machine Interface) based on display devices. An advantage of touch displays is that the operator can enter key presses or even drawings or text directly on the display screen so that no extra mechanical buttons are required.

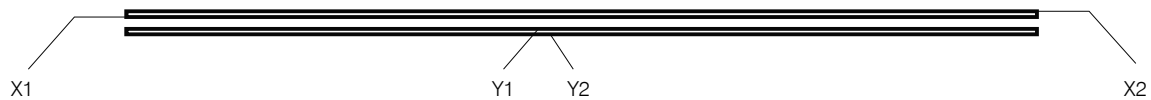
Touch screens can be divided into two main types: resistive and capacitive. Each has their advantages and disadvantage. For example capacitive displays can recognise multiple touch points and so perform new zoom functions by touching the display at two locations and moving these points apart. Capacitive touch screens are however more complicated to control than resistive ones, tend to be more expensive and require extra hardware controllers for the job. They are also not suitable for text input with a pen device as resistive touch screens are.

## 2. Example of resistive touch screen control with 2 x ADC converter channels and GPIOs.

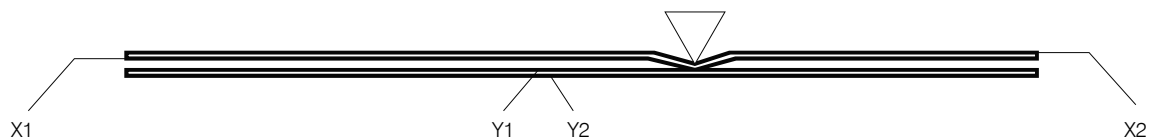
Resistive touch screens have two resistive layers – one has contacts on the left and right extremes (X1 and X2) and the other contacts on the top and bottom extremes (Y1 and Y2). The screen is transparent and is stuck onto a display (eg. LCD or TFT).



When no pressure is applied to the screen the two layers do not make contact.



When contact is applied the two layers will touch at the point of contact.



Note that multiple contact points are not supported since it is then not possible to determine where they are and they will be interpreted as a single point but at an intermediate and incorrect location. By measuring the X and Y resistance at this point it is possible to determine where it is on the display. For example, if a 320 x 240 pixel display is used it is convenient to calculate an equivalent x, y coordinate corresponding to the pixels – the centre of the display will therefore be about 160, 120.

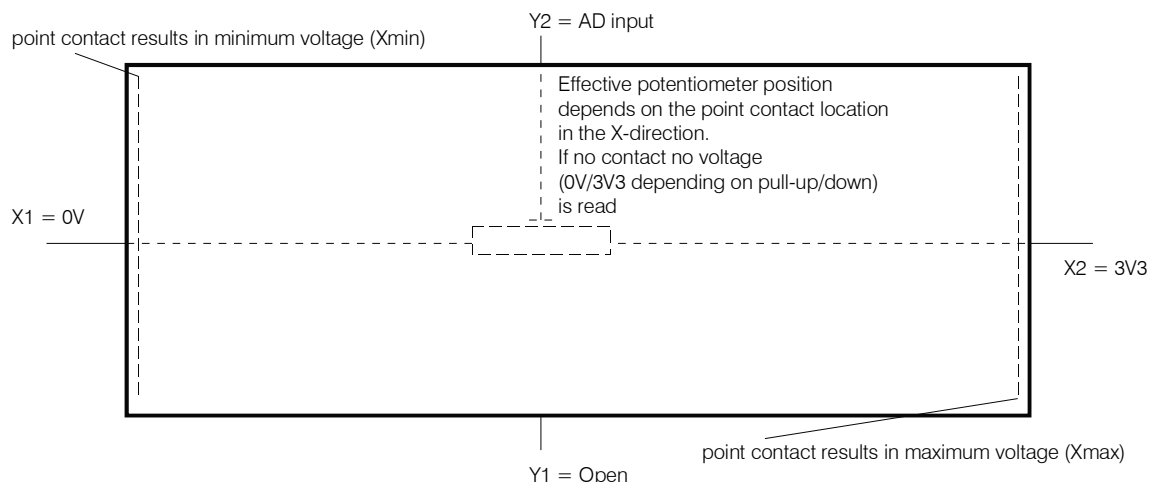
Measurement of the resistance can be performed indirectly by measuring voltages. It is assumed that both of the resistive layers have a switchable voltage applied to them (eg. 3V3) across their lengths. At the same time a processor can measure the voltage using a high-impedance ADC channel input. By alternating the voltage and measurement it is possible to measure the voltages on the X and Y resistive layers and so calculate the x, y contact position.

Resistive touch screen measurements deliver the point of contact between the two resistive layers. Since this is a point representing the point of applied pressure it allows both drawing on the touch screen with either a blunt object (entry pens) and key press locations with finger tips, whereby the centre point of the pressed area can be well determined.

Effects of bouncing during contact and spurious coordinates during the application and release of the pressure point can be filtered out in software by oversampling the measurement.

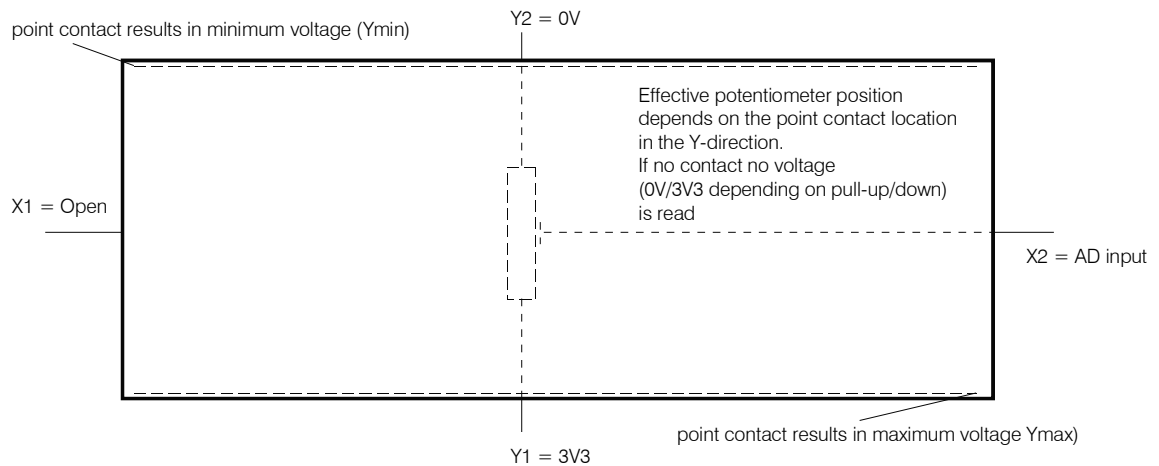
The following diagrams show the action of sampling the touch screen. It assumes 2 ADC channel inputs, which can be multiplexed with general GPIO functions, plus two extra GPIO pins. The GPIOs are used to switch and apply the measurement voltage and also to discharge the screen between samples.

1. All connections (X1, X2, Y1, Y2) are driven with 0V so that the screens are initially discharged.
2. After a delay (eg. 1ms) the 0V driving of the Y screen is stopped and the driving outputs are set to high impedance inputs. A voltage is driven on the X resistive screen (X1 = 0V, X2 = 3V3) using GPIO outputs.



3. After a delay (eg. 1ms) to allow this state to stabilize the sampling of the AD value on the Y resistive layer is started. This can be performed at any end of the Y layer since this is used as high-impedance measurement input to measure the X-position in case of a point contact with the X-resistive layer. In this example Y2 is used as measurement contact.

- After a delay (eg. 1ms) the measured Y-value is saved, the X-layer GPIOs are set back to high impedance inputs and the Y-layer connections are set to drive a voltage across them. Eg.  $Y1 = 3V3$ ,  $Y2 = 0V$ .



- After a delay (eg. 1ms) to allow any point connection value from the Y-layer to the measuring X-layer to stabilize, the sampling of the X-resistive layer is started. This can be performed at any end of the layer since this is used as high-impedance measurement input to measure the Y-position in case of a point contact with the Y-resistive layer. In this example X2 is used as measurement contact.
- After a delay (eg. 1ms) the X-value is saved.  
*At this point a co-ordinate measurement is available and it can be handled (filtered, de-bounced and acted on if a valid touch point).*  
 The Y screen driving is stopped and the GPIOs are set to high impedance inputs. A voltage is driven on the X resistive screen ( $X1 = 0V$ ,  $X2 = 3V3$ ) using GPIO outputs (corresponding to state 2).  
 The process thus continues at step 3 again and repeats (3,4,5,6) for as long as the touch input monitoring is required.

If each delay used during the sequence is 1ms there is a sample result once every 4ms (250 samples a second). The exact values measured depend on the screen and can vary between touch-screens of the same type. Also the voltage driven by the GPIOs can vary from one processor to another, as well as the digital value in relation to the analog reference voltage. This means that usually a calibration sequence is required in order to determine the values of  $X_{min}$ ,  $X_{max}$ ,  $Y_{min}$  and  $Y_{max}$ . Once these are known, and assuming that the resistive screens have a good linearity in the X and Y directions, the pixel location of the touch point can be calculated accurately.

A valid X-position (*Xvalue*) is between *Xmin* and *Xmax*. Its pixel location reference locations) is:

```
(unsigned short)(((unsigned long)((Xvalue - Xmin) * (GLCD_X - 1)))/(Xmax - Xmin));
```

Where *GLCD\_X* is the number of pixels in the X-direction of the display.

Similarly a valid Y-position (*Yvalue*) is:

```
(unsigned short)(((unsigned long)((Yvalue - Ymin) * (GLCD_Y - 1)))/(Ymax - Ymin));
```

Note that the Y-voltage is applied so that the top left corner of the display is (0, 0) and the bottom right corner (319, 239) assuming a 320 x 240 pixel LCD.

Values outside of the valid range can be considered as representing no pressure point. Whether the AD input measures a low or high value in this case depends on the AD input value when it is not connected. There may be pull-up or pull-downs regulating this in the AD itself or else it may be useful to have a weak pull-down to ensure that random valid values cannot be detected.

### 3. Conclusion

This is a provisional document containing a practical description of controlling a resistive touch screen with a general purpose microcontroller.

Modifications:

V0.00 12.08.2009: Initial draft – work in progress. Not officially released.