XC83x

AP08129

inTouch Application Kit - Touch Sliders

Application Note

V1.0, 2012-02

Microcontrollers

Edition 2012-02

Published by
Infineon Technologies AG
81726 Munich, Germany

XC83x
Revision History: V1.0 2012-02

Previous Version(s):

Page	Subjects (major changes since last revision)
–	

We Listen to Your Comments
Is there any information in this document that you feel is wrong, unclear or missing? Your feedback will help us to continuously improve the quality of this document. Please send your proposal (including a reference to this document) to:
mcdocu.comments@infineon.com

Table of Contents

1 Introduction 5

2 Hardware & Program Flow 6
2.1 Hardware 6
2.2 Program Flow 7

3 Sensing Touch on Slider 11
3.1 Slider Position Calculation 11
3.1.1 2-pad Slider 11
3.1.2 3-pad Slider 16
3.1.3 4-pad Slider 22
3.1.4 Library for Position Calculation 29

4 U-SPY 30
4.1 inTouch_Slider.ini 30
4.2 inTouch_Slider_II.ini 32

Appendix - Schematics and Layout 36

References 41

1 Introduction

In today's Human-Machine Interface (HMI) designs, capacitive touch technology is now often more widely used than traditional mechanical buttons. Capacitive touch technology is the more popular choice because it brings flexibility, a high-level of customization, and a significant reduction in overall system cost.

The *inTouch Application Kit* is available to help learn about working with the advanced touch solutions provided by Infineon. Step-by-step tutorials covers the basics of Infineon's touch solutions, while example application code can be used to start developing new touch-related projects.

The *inTouch Application Kit* comprises of a mother board, supplied as a USB stick, and a number of daughter boards. **Figure 1** shows the USB stick with the Slider daughter board.

Among the many different touch input elements that can be designed with capacitive touch technology, the slider is gaining popularity because of the intuitive control it provides. This application note describing the slider daughter board, aims to highlight the ease of implementing a design with Infineon's touch solutions. Topics covered include program flow, touch behavior and touch position calculation algorithm.

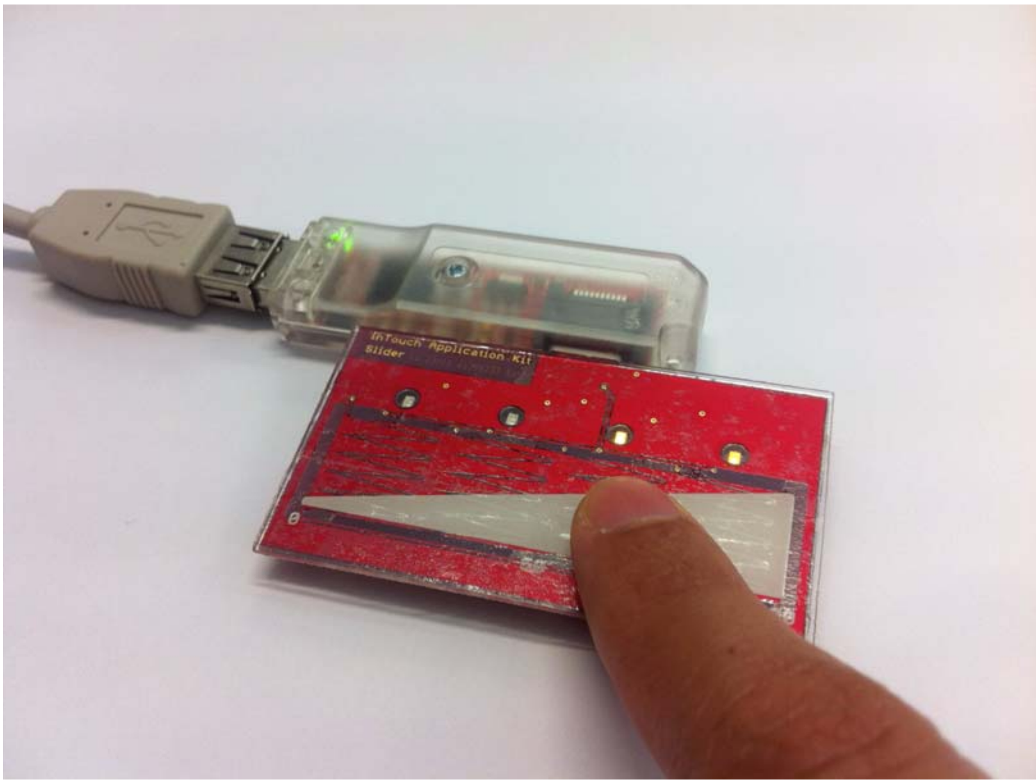

Figure 1 ***inTouch Application Kit* (USB Stick and Slider board)**

2 Hardware & Program Flow

This section describes the hardware used and the connections involved.

2.1 Hardware

Infineon's XC836MT 2FRI (**Figure 2**) is used in this application. The XC836MT is embedded in the *inTouch Application Kit*'s USB stick. For more details regarding the USB stick, please refer to *AP08126: Infineon Touch Solutions - inTouch Application Kit*.

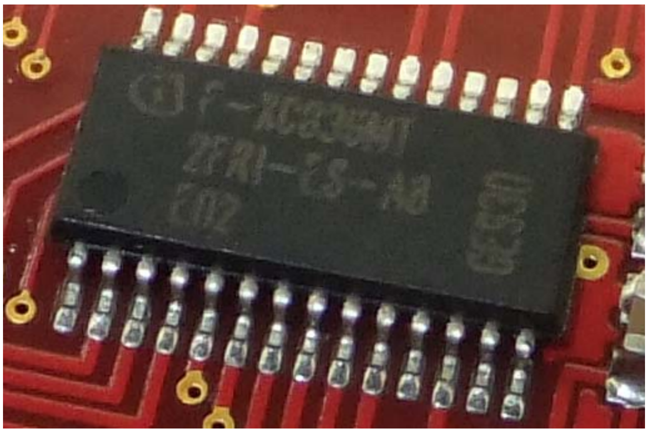

Figure 2 Infineon's XC836MT 2FRI

The *inTouch Slider* and *inTouch Slider II* boards (**Figure 3**) are available as plug-in daughter boards which are part of the *inTouch Application Kit.*

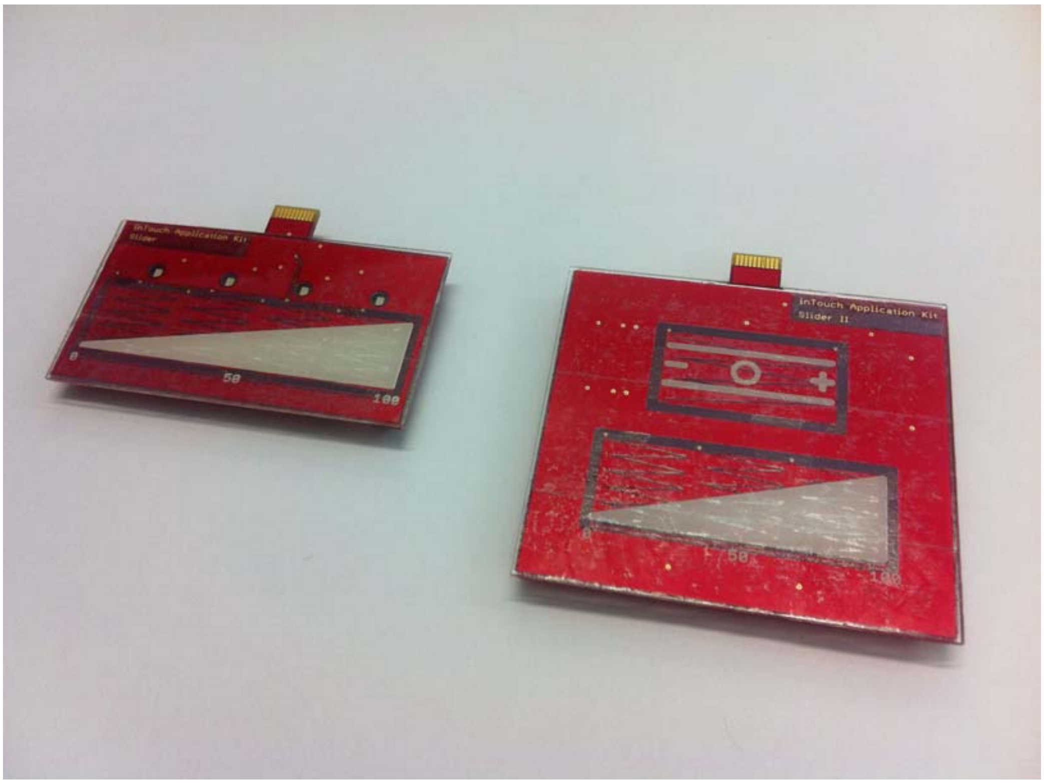

Figure 3 Slider Boards

The *inTouch Slider* board consists of a 4-pad slider (**Figure 3**, left). The *inTouch Slider II* board consists of a 2-pad and a 3-pad slider (**Figure 3**, right). Both slider boards are standard PCBs with a 1mm thick plexiglas cover glued to the board.

The 4-pad slider on the *inTouch Slider* board is connected to 4 LEDTS pad inputs of the XC836. 4 indicator LEDs are each connected to an LEDTS line pin and they share an LEDTS column pin of the XC836. The 2-pad and 3-pad sliders on the *inTouch Slider II* board are connected to 2 and 3 LEDTS pad inputs of the XC836 respectively. The schematics are available in the **Appendix - Schematics and Layout**.

Users can tap or swipe the touch sliders.

2.2 Program Flow

This section presents an overview of the program in terms of the interrupts involved, and then provides the tasks performed in each interrupt service routine. Both programs for *inTouch_Slider* and *inTouch_Slider_II* are essentially the same. The main difference is that the program for *inTouch_Slider* has additional tasks to toggle the indicator LEDs.

In terms of interrupts, the UART interrupt has the highest priority to ensure the smooth transmission of data to U-SPY. The Time Frame interrupt has the medium priority. In this service routine, touch sense related tasks are performed each time pad capacitance has been measured. LED updates (for *inTouch_Slider* board), which are performed in the Time Slice interrupt, have low priority. The Timer 2 (T2) Overflow interrupt is given lowest priority due to its slow frequency. **Figure 4** and **Table 5** provide an illustration of the program overviews for *inTouch_Slider* and *inTouch_Slider_II* boards respectively.

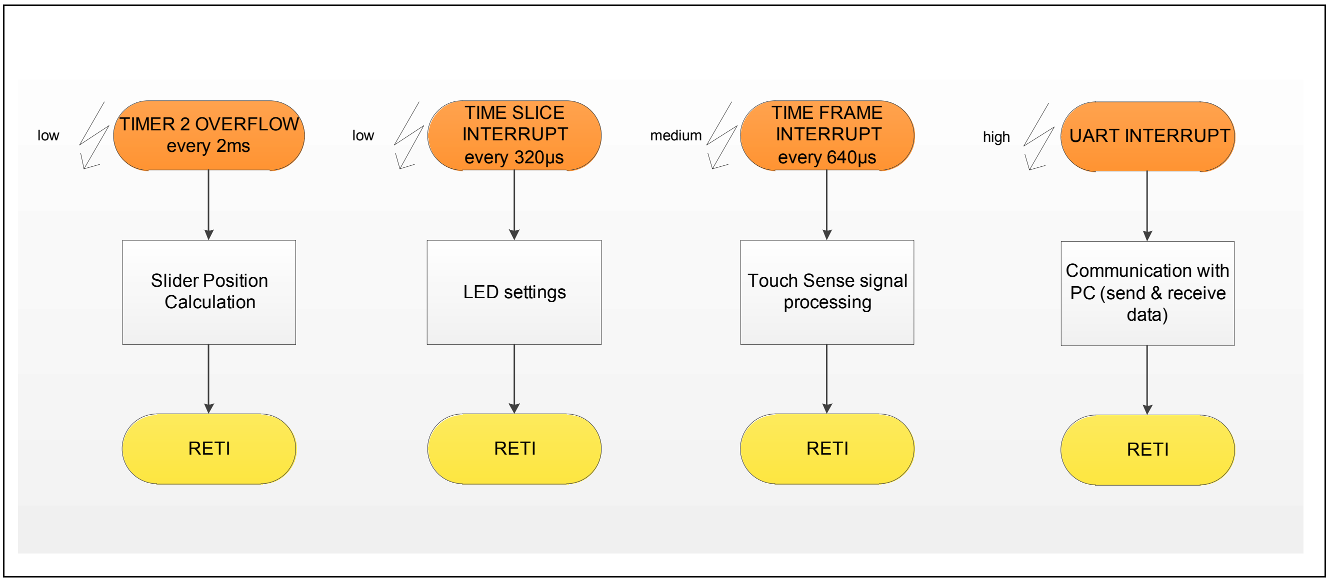

Figure 4 Program Overview for *inTouch Slider* board

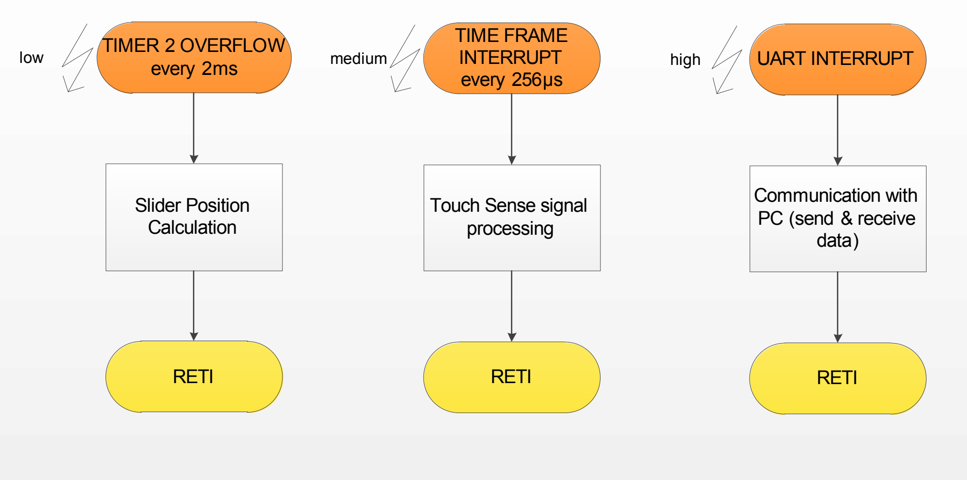

Figure 5 Program Overview for *inTouch Slider II* board

The tasks performed in each interrupt service routine are further illustrated in the flowcharts that follow:

- T2 Overflow Interrupt (**Figure 6**)
  – The T2 module provides a slow time base by generating the T2 Overflow interrupt for calculations necessary to handle the touch slider.
- UART Interrupt (**Figure 7**)
  – The UART module, which is part of the XC800 core, is used for full-duplex UART communication with the PC.
- Time Frame Interrupt (**Figure 8**)
  – The LEDTS module generates this interrupt after every measurement where signal processing and touch detection take place.
- Time Slice Interrupt (**Figure 9**) (*inTouch_Slider* board only)
  – The LEDTS module generates this interrupt after every LED column activation where the pattern for the next LED column is loaded into shadow registers.

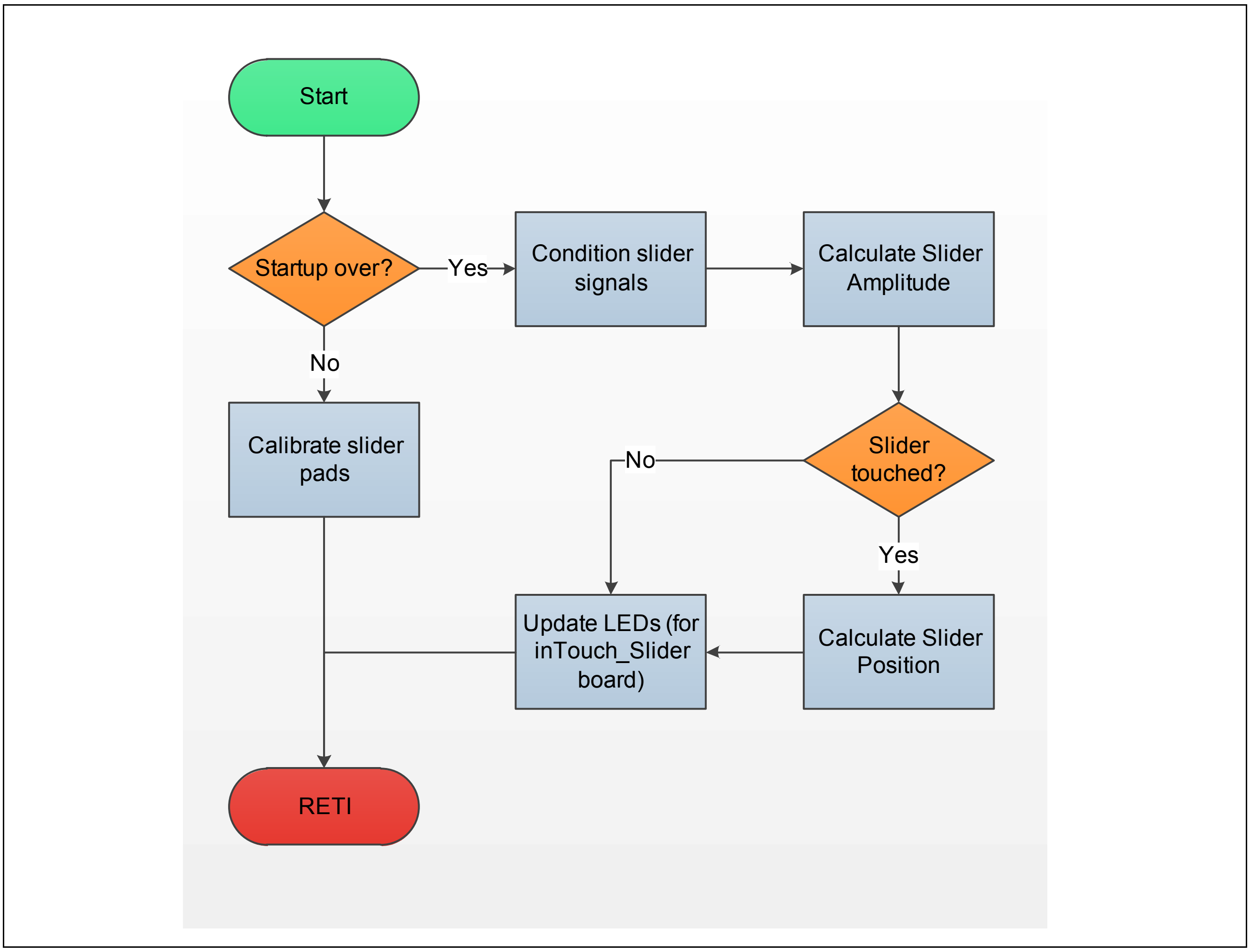

Figure 6 Timer 2 Overflow Interrupt Service Routine

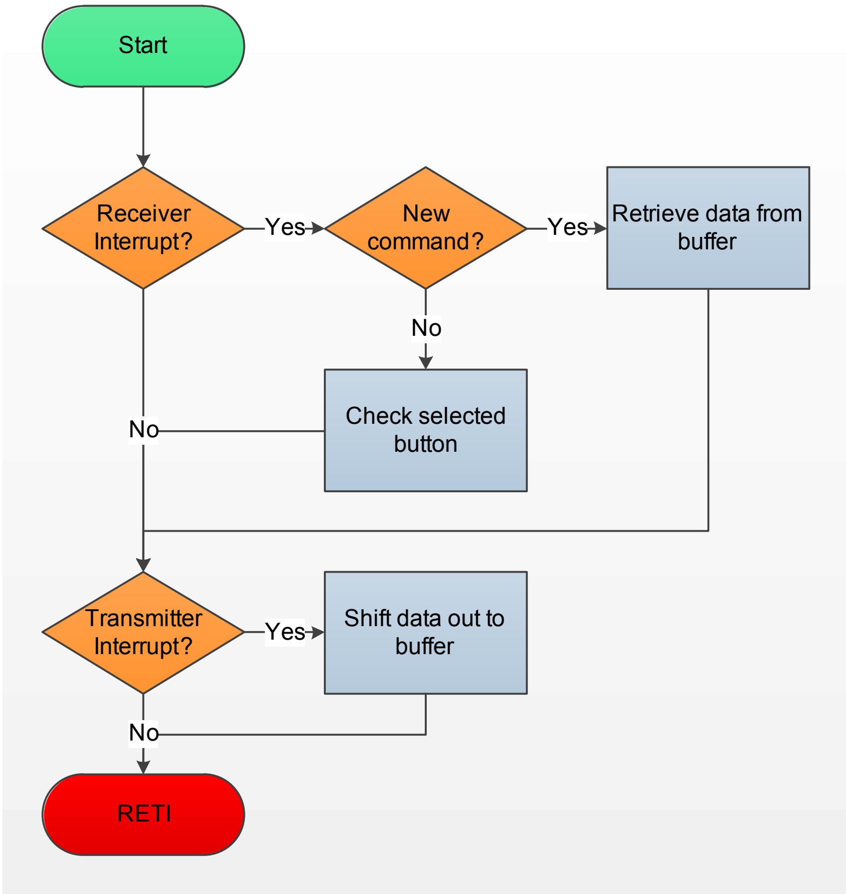

Figure 7 UART Interrupt Service Routine

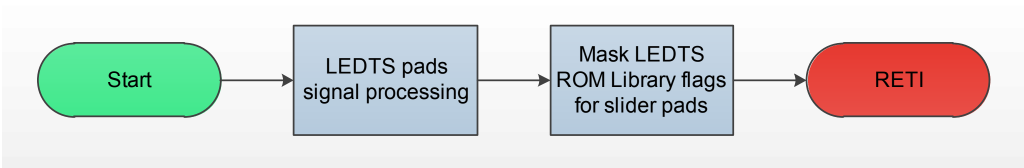

Figure 8 Time Frame Interrupt Service Routine

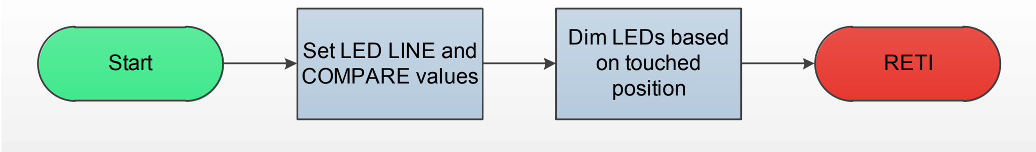

Figure 9 Time Slice Interrupt Service Routine (for *inTouch Slider* only)

3 Sensing Touch on Slider

This section describes how the LEDTS module of the XC836, complemented with a software library, controls the touch slider. The algorithm for calculating the location of touch is also explained in the following section.

The main touch sensing functions, handled by software, are as follows:

- Sample accumulation (ROM library)
- Signal filtering and moving average calculation (ROM library)
- Touch detection (ROM library)
- Touch slider calibration (user software in Flash)
- Signal tuning (user software in Flash)

If properly configured, the LEDTS automatically measures the capacitance of the slider pads. This capacitance increases when a slider pad is touched. A library function in ROM processes the capacitance signals and detects touch on the slider. It does so by accumulating a number of samples and low-pass filtering them to obtain a moving average. The moving average filters noise and is used as a reference to detect sudden changes in capacitance. When the slider is touched or released, a corresponding pad flag in RAM will be set or reset. For more information on the LEDTS ROM Library, please refer to the XC836 User's Manual.

The pad flags for the slider pads are unused (always cleared) in the slider position calculation algorithm. It is the moving averages ("*pad averages*") that are used instead to calculate the position of the touch. The slider pads are automatically calibrated to the same sensitivity and resolution during startup. Once the pad averages are stable, a position calculation algorithm is run if the slider is touched. The calculated position is then used to determine the location of touch, and is shown on the LEDs.

3.1 Slider Position Calculation

This section describes the algorithm for calculating the location of touch on the slider. This section is categorized according to the number of pads forming the touch slider.

3.1.1 2-pad Slider

The two touch pads of the slider are placed in a spatially-interpolated manner as shown in **Figure 10**. The slider is divided into 2 sections for position calculation.

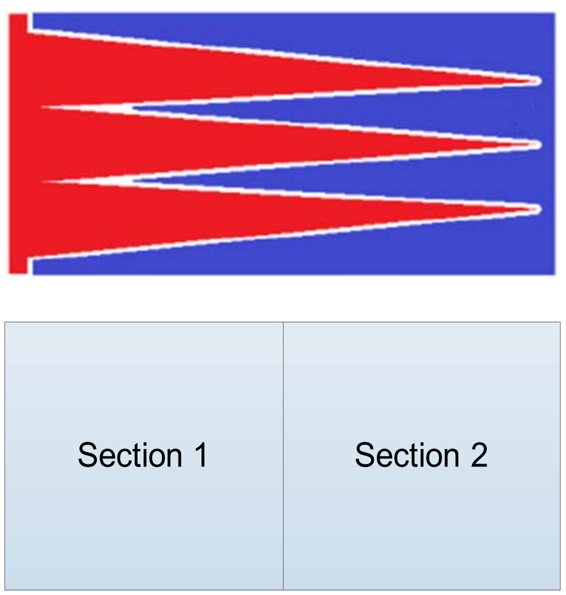

Figure 10 Spatially interpolated 2-pad slider layout and abstraction

If the pads are calibrated to roughly the same sensitivity and a finger slides from left to right with constant speed and constant pressure (constant effective finger area), the pad average signals are expected to behave in a linear manner in this model as seen in **Figure 11**.

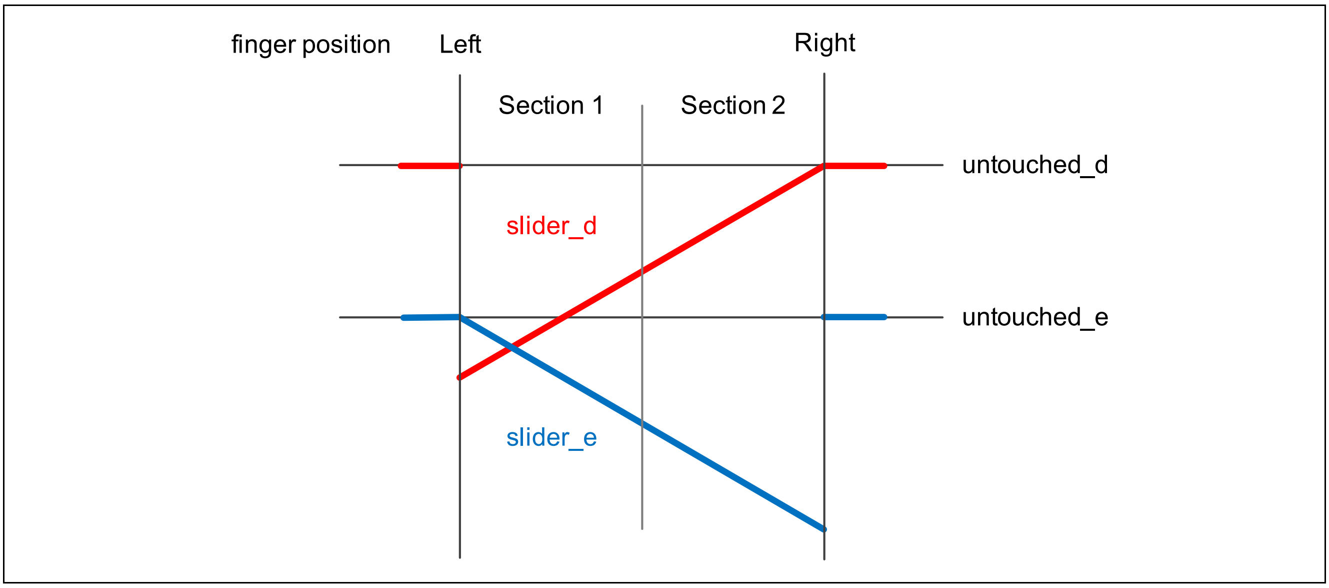

Figure 11 Pad average signals of the two pads when slider is swiped

Values untouched_d and untouched_e are the pad average levels for pads D and E respectively when they are not touched.

If the pads have roughly the same sensitivity, the two signals can be tuned to have a common untouched (UT) level (**Figure 12**). The actual signals can be expected to look like those in **Figure 13**. If the two pads have slightly different sensitivity due to board layout, it may be necessary to manually modify the oscillation windows.

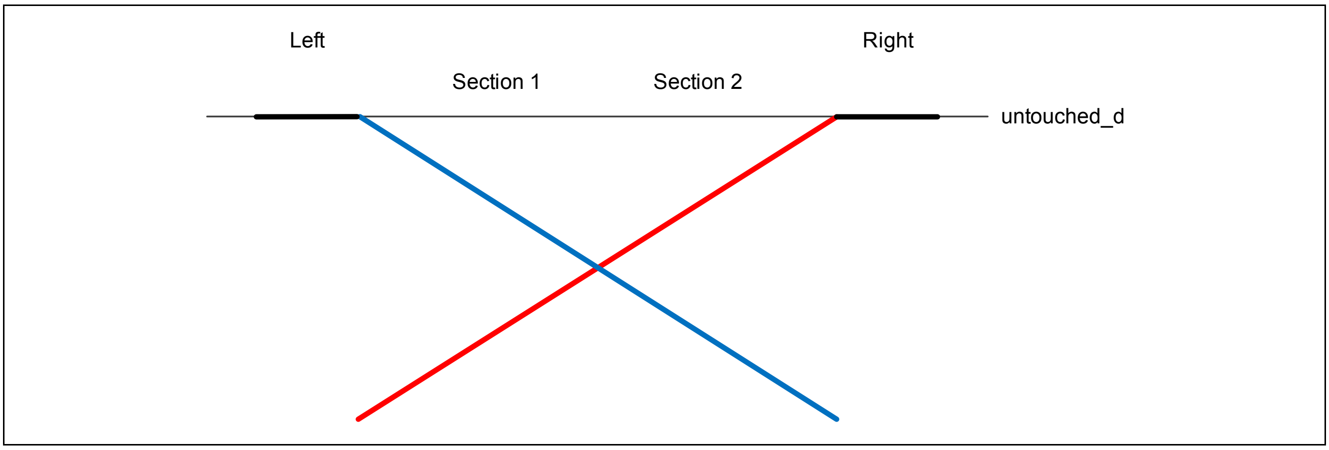

Figure 12 Pad average signals of the two slider pads after tuning

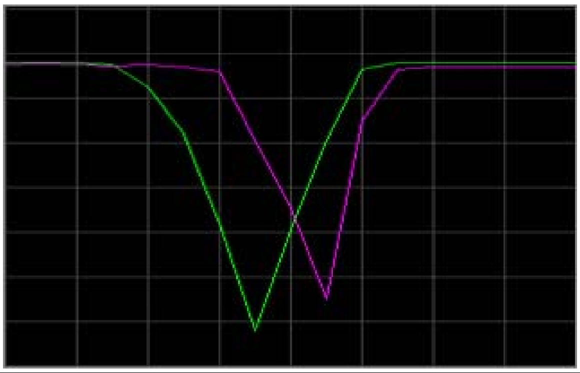

Figure 13 Actual pad average signals after tuning

The, now common, untouched level (UT) is very high compared to the difference between touched and untouched states. To make calculations easier, the signals are transformed near to zero by linear combinations which can be represented by the formulae below. Figure 14 provides an illustration of the transformation.

$$X = UT - B \qquad Y = UT - A$$

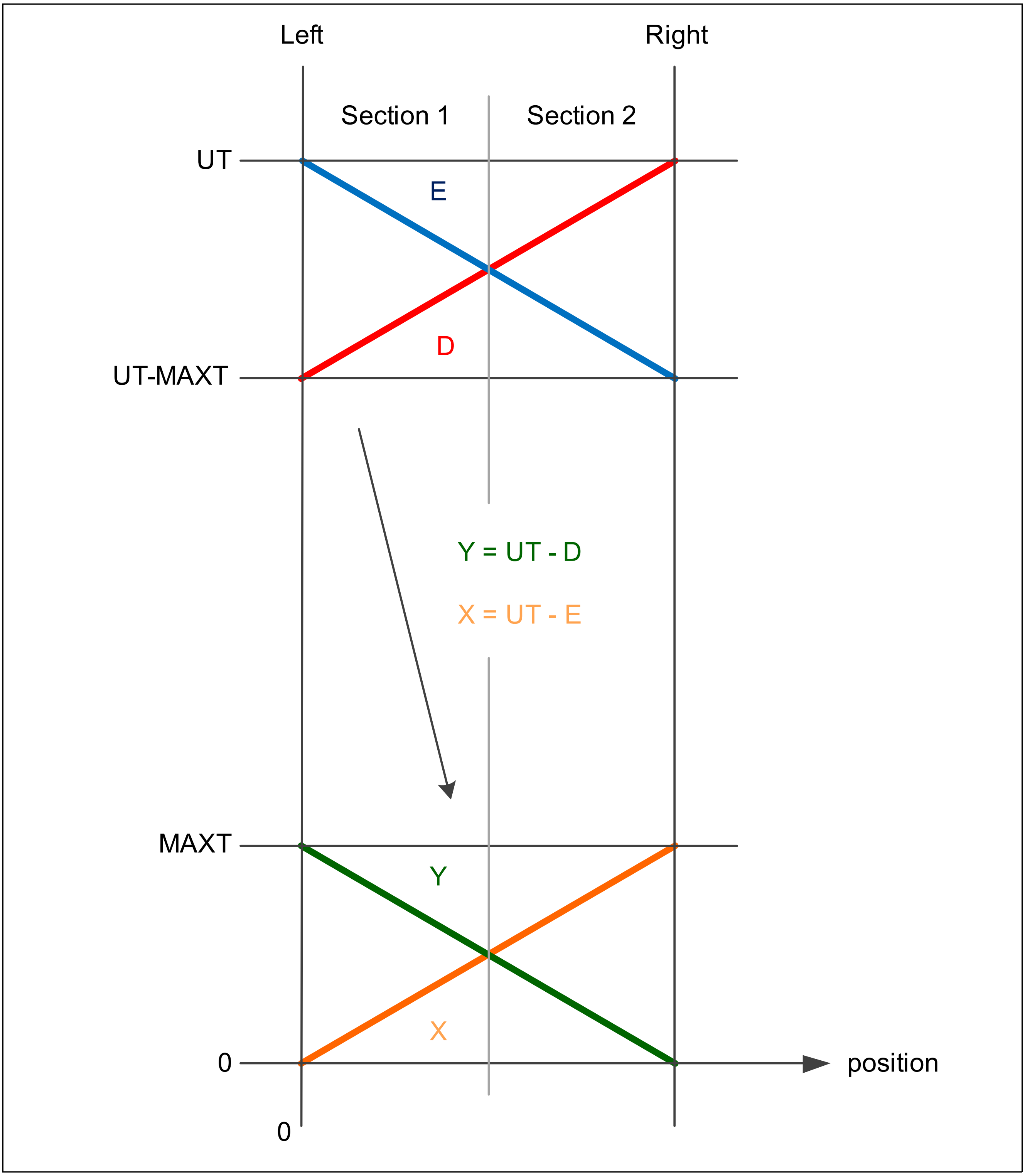

Figure 14 Transformed pad average signals

Before the transformation, Section 1 has two signals between *UT* and *UT-MAXT*. *UT* stands for the untouched level and *UT-MAXT* stands for the signal level when the largest area of the respective pad is touched (this happens at the two extremes).

After the transformation, the *X* and *Y* signals have much lower values.

The two signals can be described as:

$$X = MAXT \times d \quad (1)$$

$$Y = MAXT - MAXT \times d \quad (2)$$

If we rearrange **Equation (1)**, we get $MAXT = \frac{X}{d}$ which we can substitute in **Equation (2)**:

$$Y = \frac{X}{d} - \frac{X}{d}d$$

$$Y + X = \frac{X}{d}$$

$$d = \frac{X}{X+Y} \quad (3)$$

$$d = 1 - \frac{Y}{X+Y} \quad (4)$$

One division is needed to calculate the position; this operation needs the most computing performance. To minimize the error, it is safer to use **Equation (3)** if *X* is larger and **Equation (4)** if *Y* is larger.

A scaling factor of 2^R is added to create a more usable calculated position (**Figure 15** and **Figure 16**). *R* is for resolution and corresponds to the number of left bitshifts in the numerator.

Section 1 $$d = 2^R - \frac{Y \times 2^R}{X+Y}$$

Section 2 $$d = \frac{X \times 2^R}{X+Y}$$

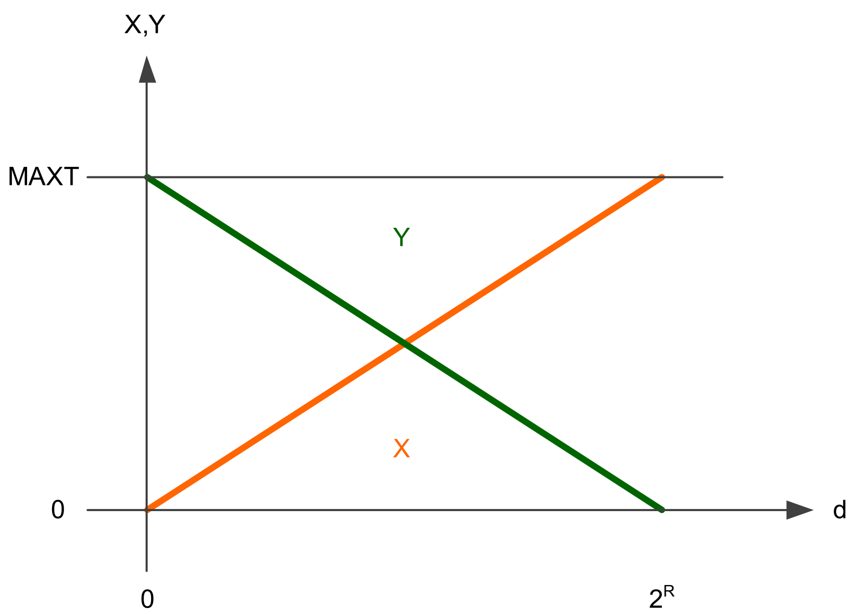

Figure 15 Transformed pad average signals after offsetting and scaling

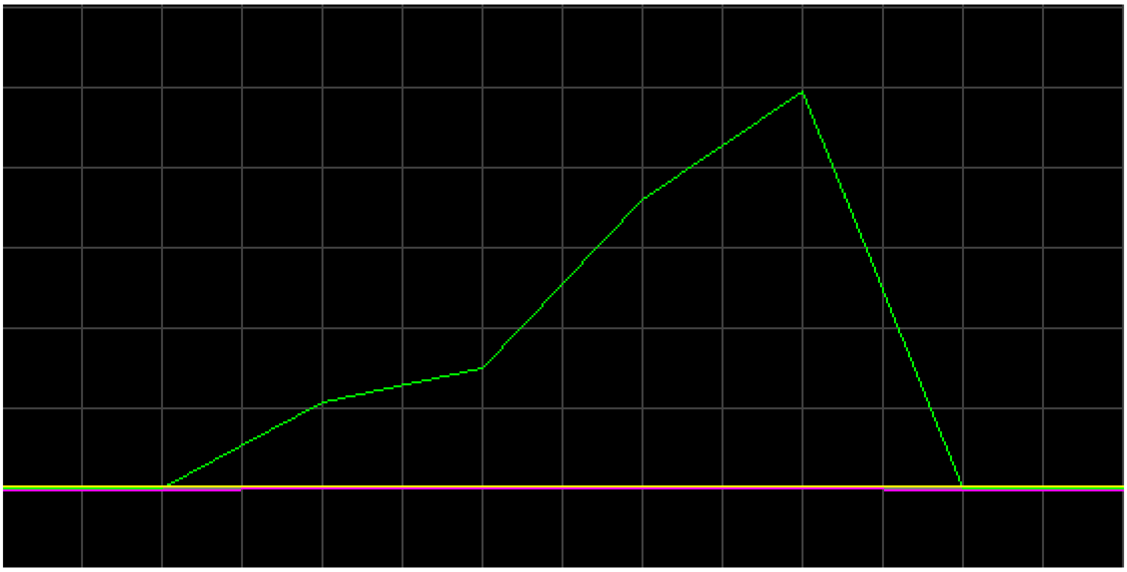

Figure 16 Actual calculated position

3.1.2 3-pad Slider

The three touch pads of the slider are placed in a spatially-interpolated manner as shown in **Figure 17**. The slider is divided into 3 sections for position calculation.

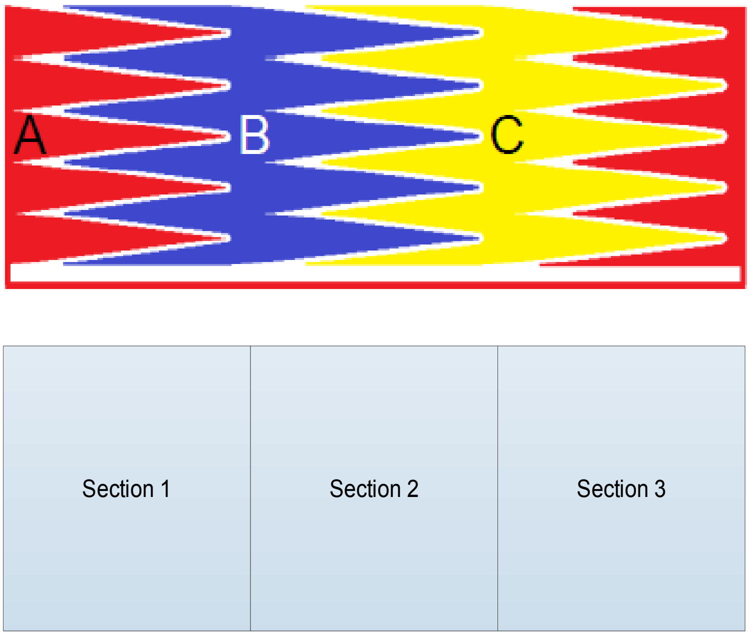

Figure 17 Spatially interpolated 3-pad slider layout and abstraction

If the pads are calibrated to roughly the same sensitivity and the finger slides from left to right with constant speed and constant pressure (constant effective finger area), the pad average signals are expected to behave in a linear manner in this model as seen in **Figure 18**.

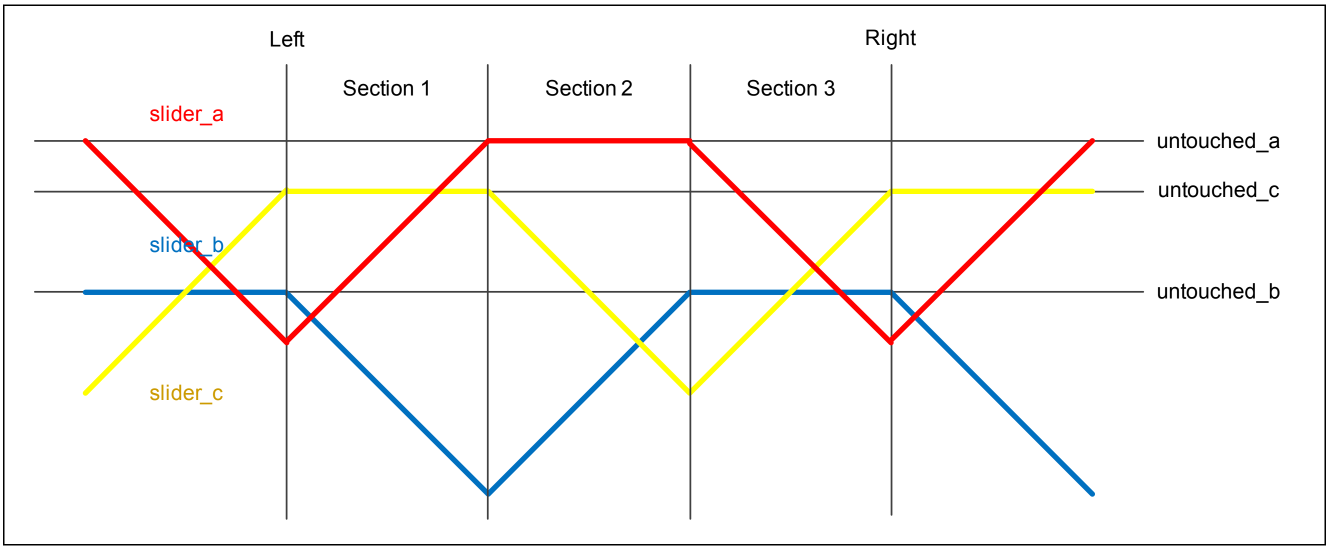

Figure 18 Pad average signals of the three slider pads during swiping

Values untouched_a, untouched_b and untouched_c are the pad average levels for pads A, B and C respectively when they are not touched.

If the pads have roughly the same sensitivity, the three signals can be tuned to have a common untouched level (**Figure 19**). The actual signals can be expected to look like those in **Figure 20**.

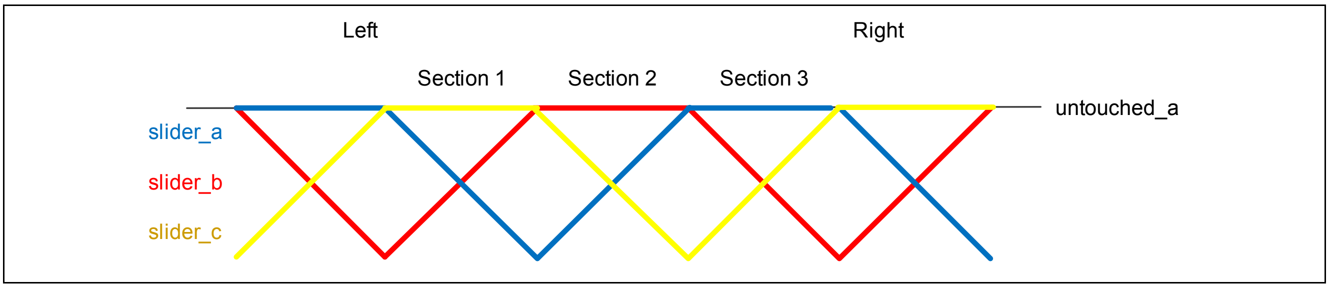

Figure 19 Pad average signals of the three slider pads after tuning

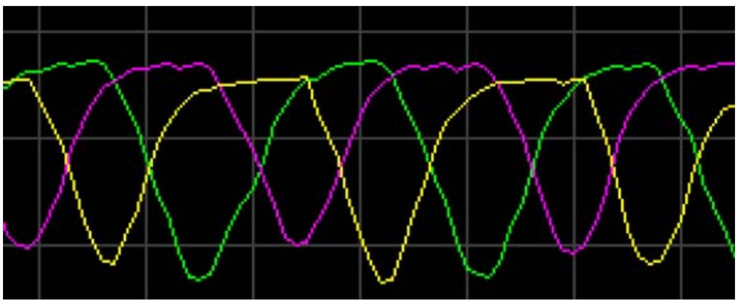

Figure 20 Actual pad average signals after tuning

The, now common, untouched (UT) level is very high compared to the difference between touched and untouched states. To make calculations easier, the signals are transformed near to zero by linear combinations which can be represented by the formulae below. **Figure 21** provides an illustration of the transformation. This transformation also makes the transitions between sections smooth, which is especially important if the three pads have different sensitivity or unstable untouched levels due to imperfect calibration or a changing environment.

$$X = \frac{A + B}{2} - C \qquad Y = \frac{A + C}{2} - B \qquad Z = \frac{B + C}{2} - A$$

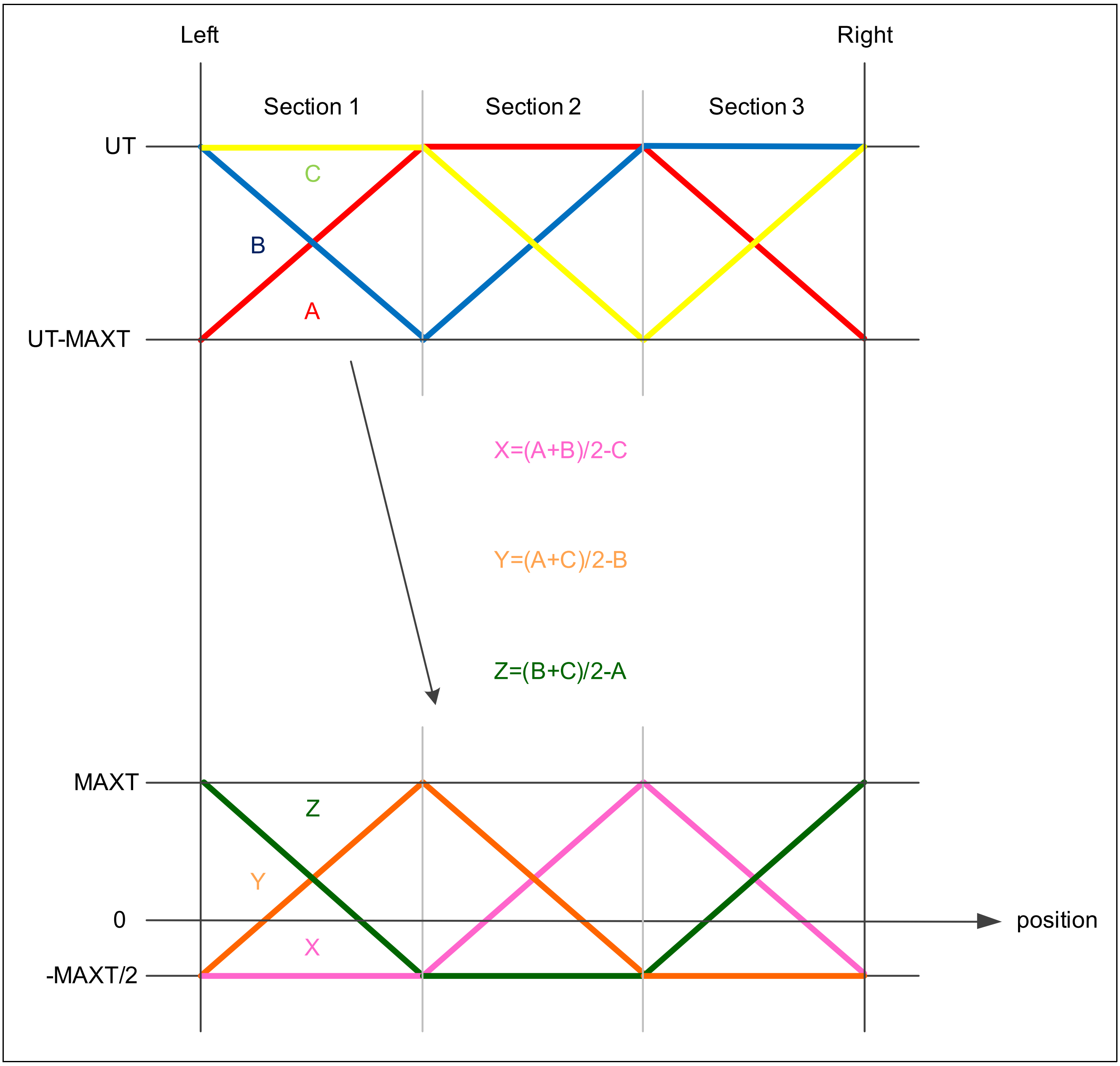

Figure 21 Combined pad average signals

The resulting *X*, *Y* and *Z* signals still have three distinct sections.

Section 1

Before the transformation, Section 1 has three signals between *UT* and *UT-MAXT* (**Figure 22**). *UT* stands for the untouched level and *UT-MAXT* stands for the signal level when the largest area of the respective pad is touched (this happens at section borders).

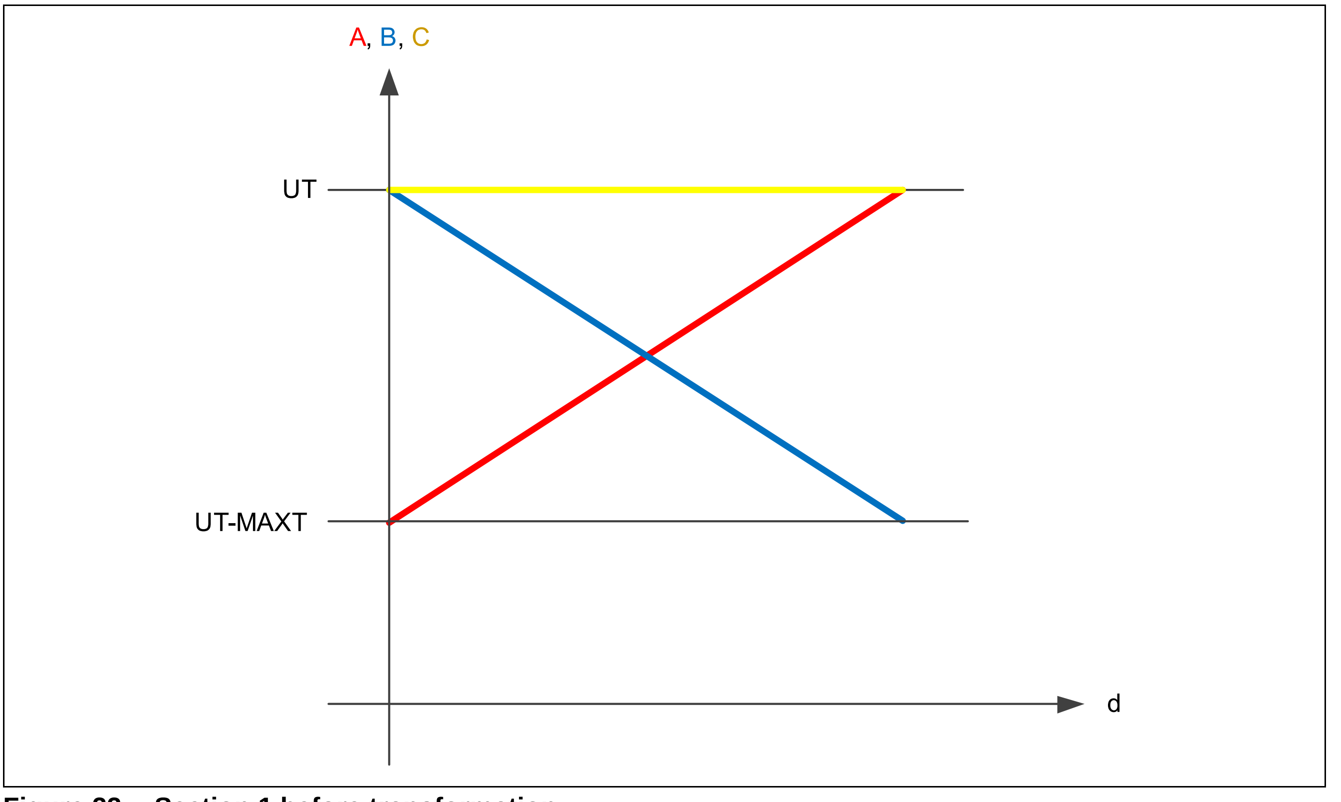

Figure 22 Section 1 before transformation

After the transformation, the *X*, *Y* and *Z* signals have much lower values (**Figure 23**). The position axis has been arbitrarily scaled from -1 to 2 in this region for convenience.

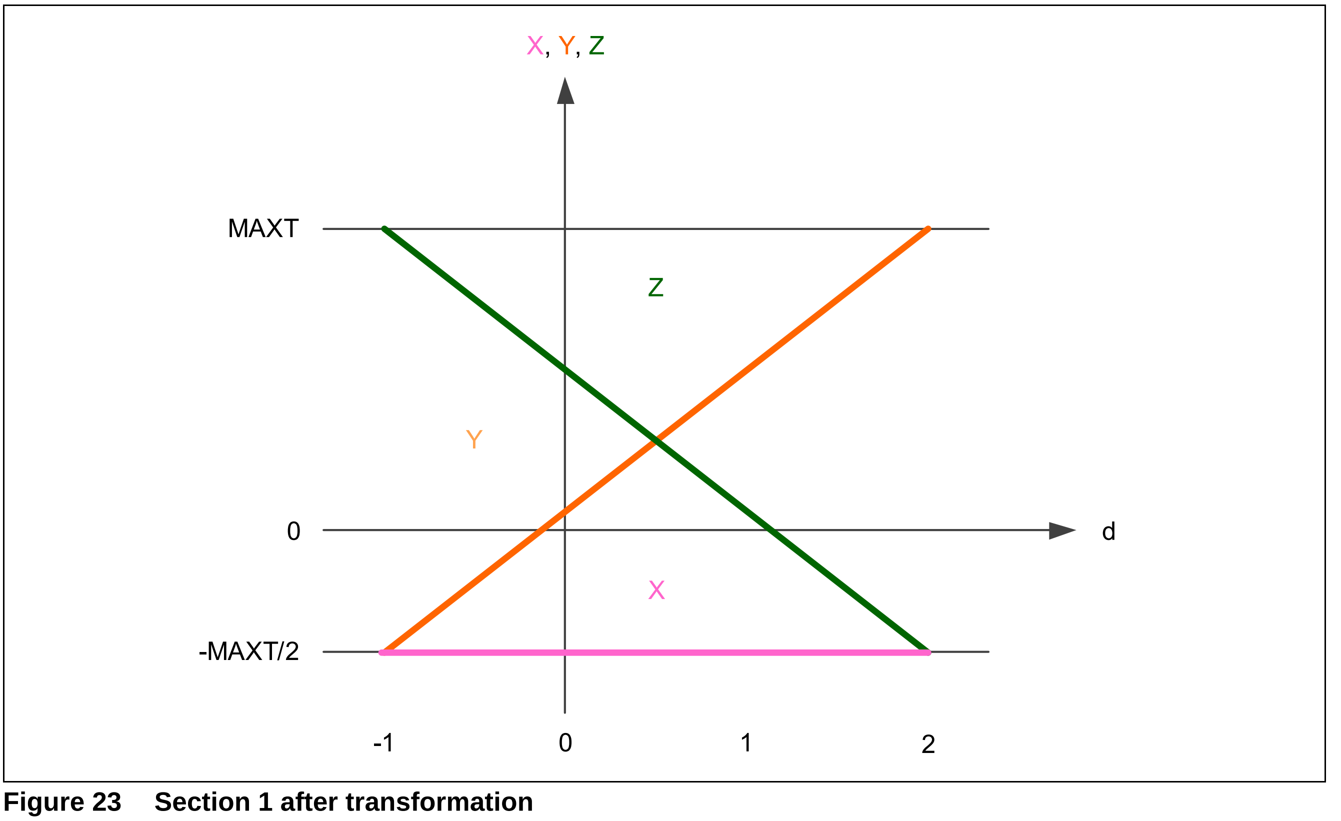

Figure 23 Section 1 after transformation

Signal *X* is constant low in this section so it does not participate in the position calculation. The other two signals can be described as:

$$Y = \frac{MAXT}{2} d \tag{1}$$

$$Z = \frac{MAXT}{2} - \frac{MAXT}{2} d \tag{2}$$

If we rearrange **Equation (1)**, we get $\frac{MAXT}{2} = \frac{Y}{d}$ which we can substitute in **Equation (2)**:

$$Z = \frac{Y}{d} - \frac{Y}{d} d$$

$$Z = \frac{Y}{d}(1 - d)$$

$$d(Y + Z) = Y$$

$$d = \frac{Y}{Y + Z} \tag{3}$$

$$d = 1 - \frac{Z}{Y + Z} \tag{4}$$

One division is needed to calculate the position; this operation needs the most computing performance. To minimize the error, it is safer to use **Equation (3)** if *Y* is larger and **Equation (4)** if *Z* is larger.

An offset of 1 and a scaling factor of 2^R are added to create a more usable calculated position (**Figure 24**). R is for resolution and corresponds to the number of left bitshifts on the numerator.

Section 1 Left

$$d = 2 \times 2^R - \frac{Z \times 2^R}{Y + Z}$$

Section 1 Right

$$d = \frac{Y \times 2^R}{Y + Z} + 2^R$$

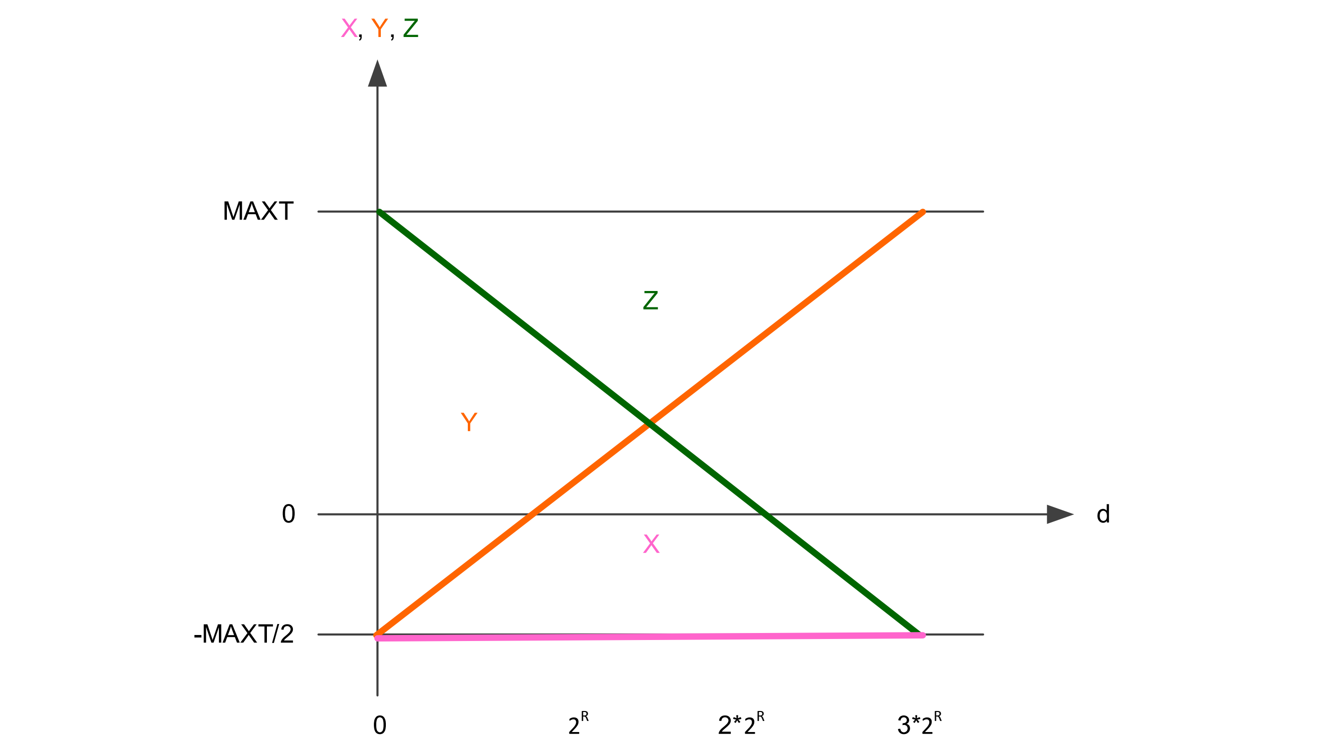

Figure 24 Section 1 after offsetting and scaling

Sections 2 and 3

In these two sections, the position can be calculated in a similar way as in **Section 1**, using the two non-constant signals. Offsets of 4 and 7, and the same scaling factor, can then be added to sections 2 and 3 respectively to get a calculated position of 0..9*2^R.

Section 2 Left $$d = 5 \times 2^R - \frac{Y \times 2^R}{X + Y}$$

Section 2 Right $$d = \frac{X \times 2^R}{X + Y} + 4 \times 2^R$$

Section 3 Left $$d = 8 \times 2^R - \frac{X \times 2^R}{X + Z}$$

Section 3 Right $$d = \frac{Z \times 2^R}{X + Z} + 7 \times 2^R$$

Figure 25 gives an illustration of the calculated position across all 3 sections while **Figure 26** shows the actual calculated position.

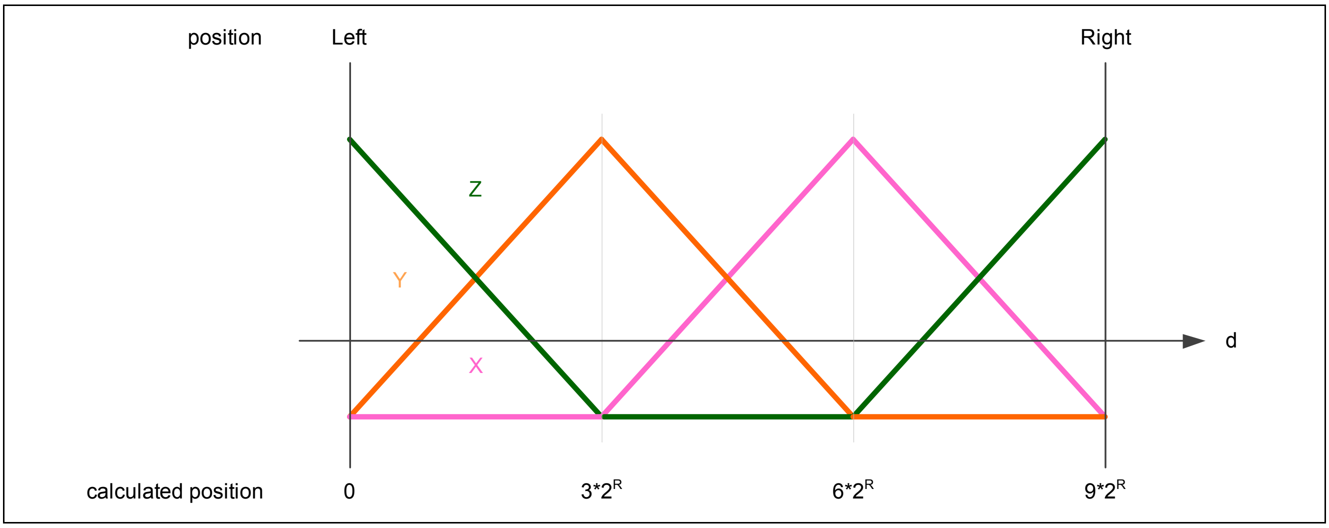

Figure 25 Calculated position vs real position across all sections

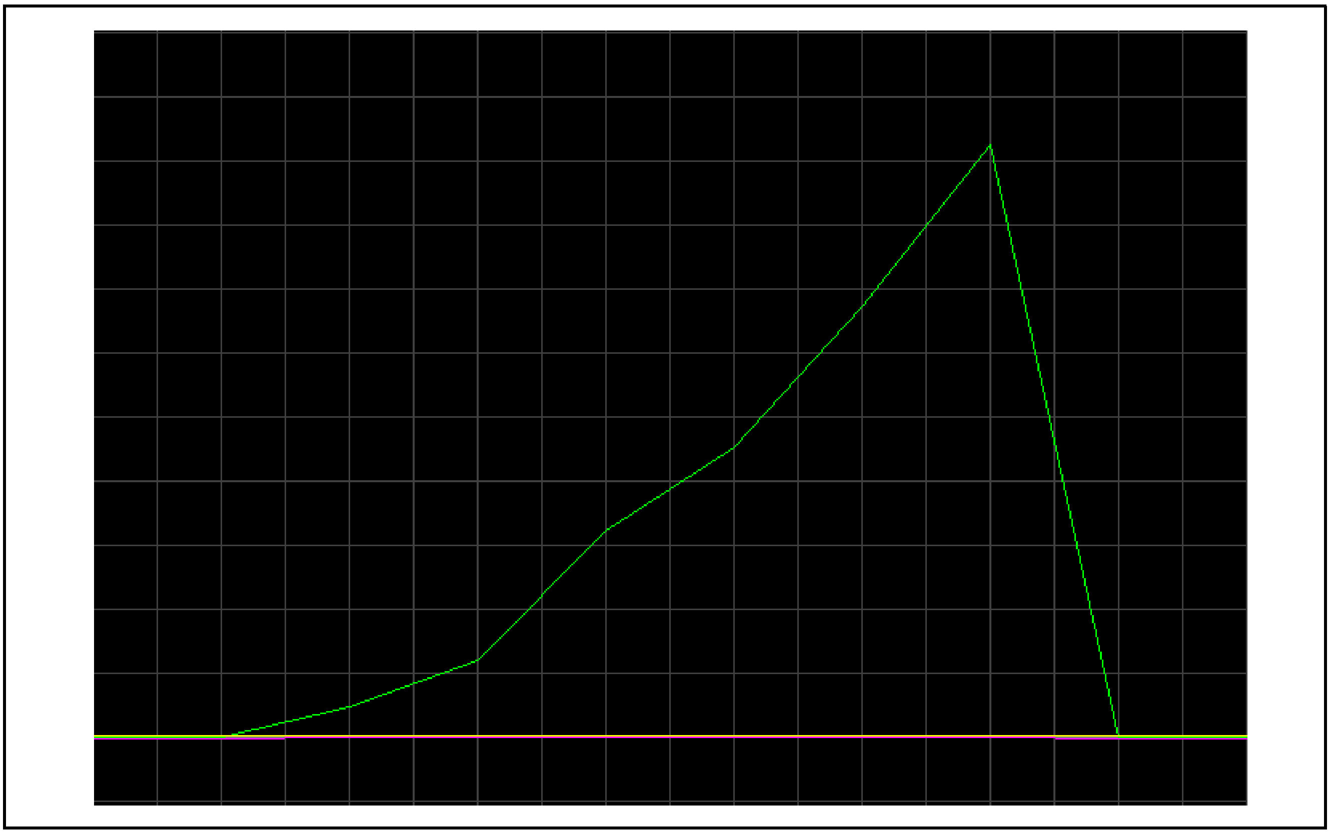

Figure 26 Actual calculated position

3.1.3 4-pad Slider

The four touch pads of the slider are placed in a spatially-interpolated manner as shown in **Figure 27**. The slider is divided into 4 sections for position calculation.

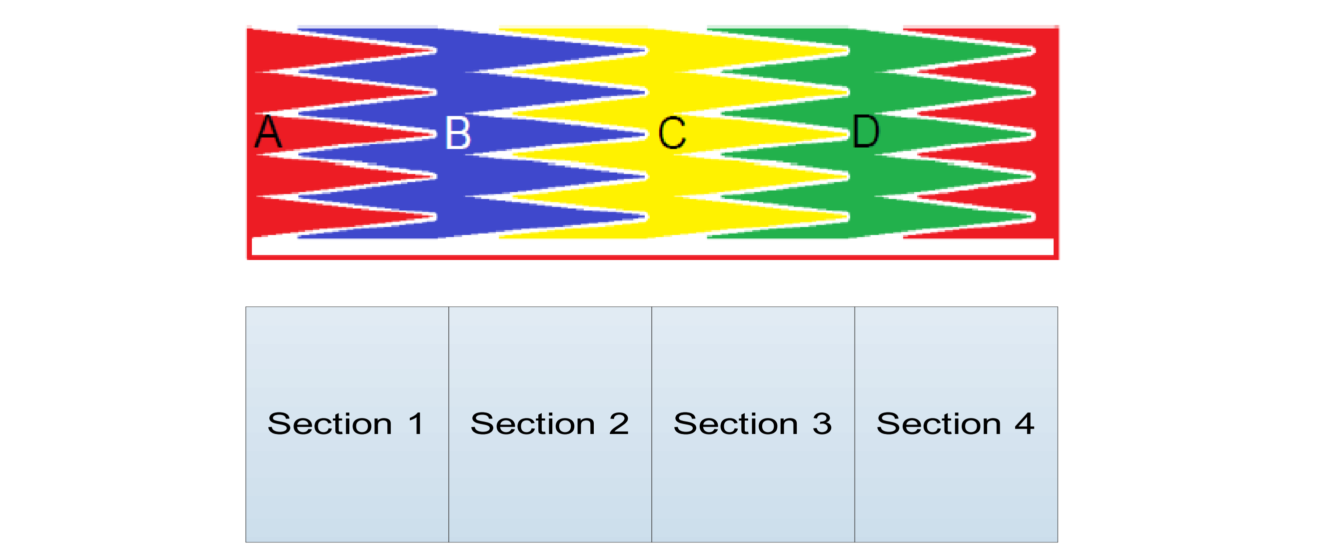

Figure 27 Spatially interpolated 4-pad slider layout and abstraction

If the pads are calibrated to roughly the same sensitivity and the finger slides from left to right with constant speed and constant pressure (constant effective finger area), the pad average signals are expected to behave in a linear manner in this model as seen in **Figure 28**.

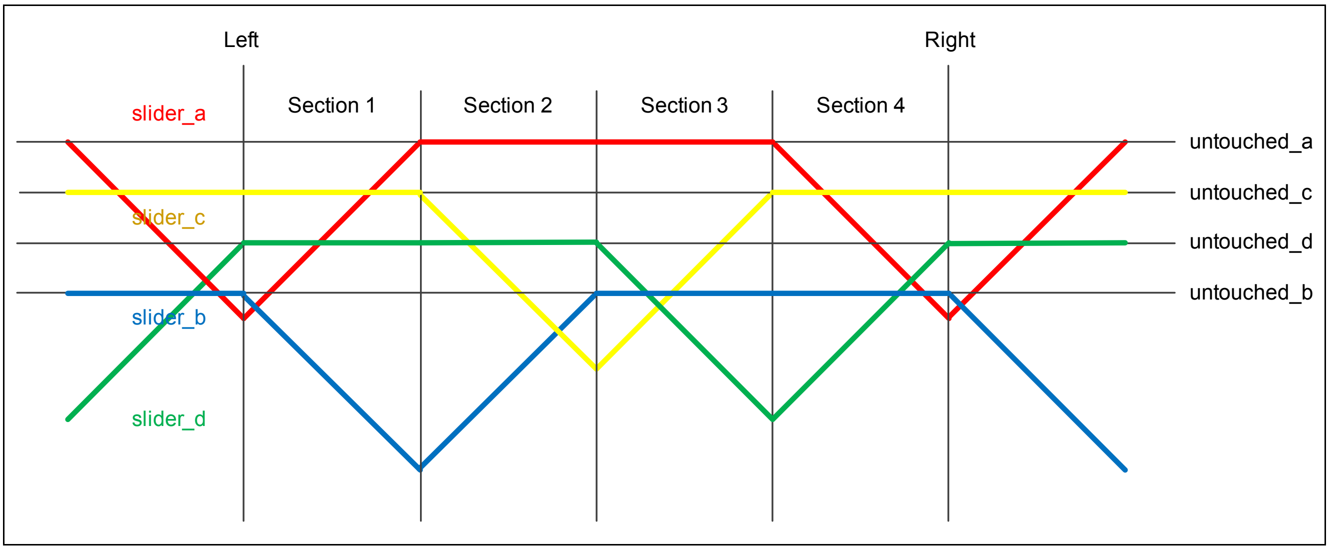

Figure 28 Pad average signals of the four slider pads during swiping

Values untouched_a, untouched_b, untouched_c and untouched_d are the pad average levels for pads A, B, C and D respectively when they are not touched.

If the pads have roughly the same sensitivity, the four signals can be tuned to have a common untouched level (**Figure 29**). The actual signals can be expected to look like those in **Figure 30**.

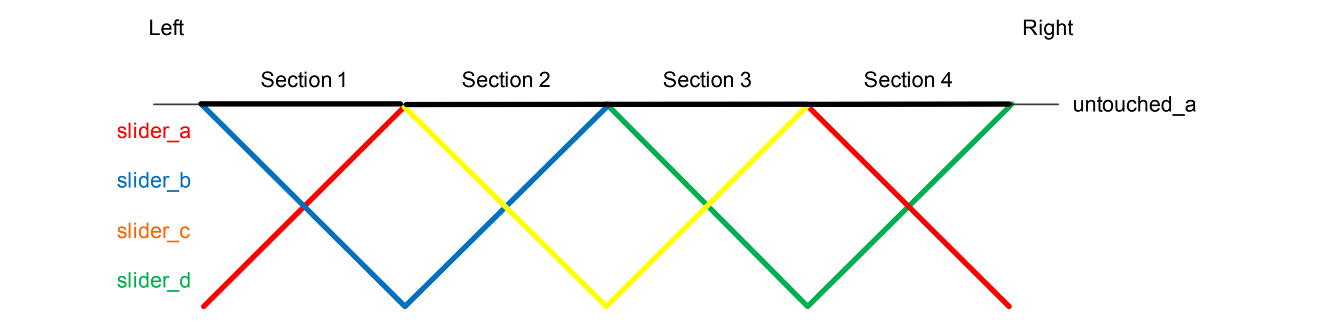

Figure 29 Pad average signals of the four slider pads after tuning

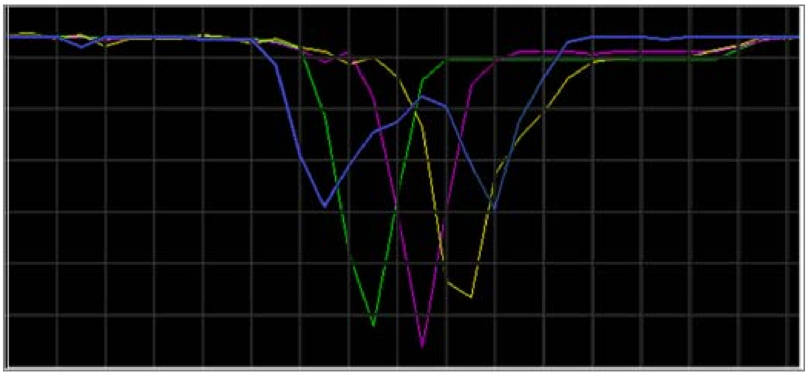

Figure 30 Actual pad average signals after tuning

The, now common, untouched level is very high compared to the difference between touched and untouched states. To make calculations easier, the signals are transformed near to zero by linear combinations which can be represented by the formulae below. **Figure 31** provides an illustration of the transformation. This transformation also makes the transitions between sections smooth, which is especially important if the four pads have different sensitivity or unstable untouched levels due to imperfect calibration or a changing environment.

$$w = \frac{A + B + C}{3} - D \qquad X = \frac{A + B + D}{3} - C \qquad Y = \frac{A + C + D}{3} - B \qquad Z = \frac{B + C + D}{3} - A$$

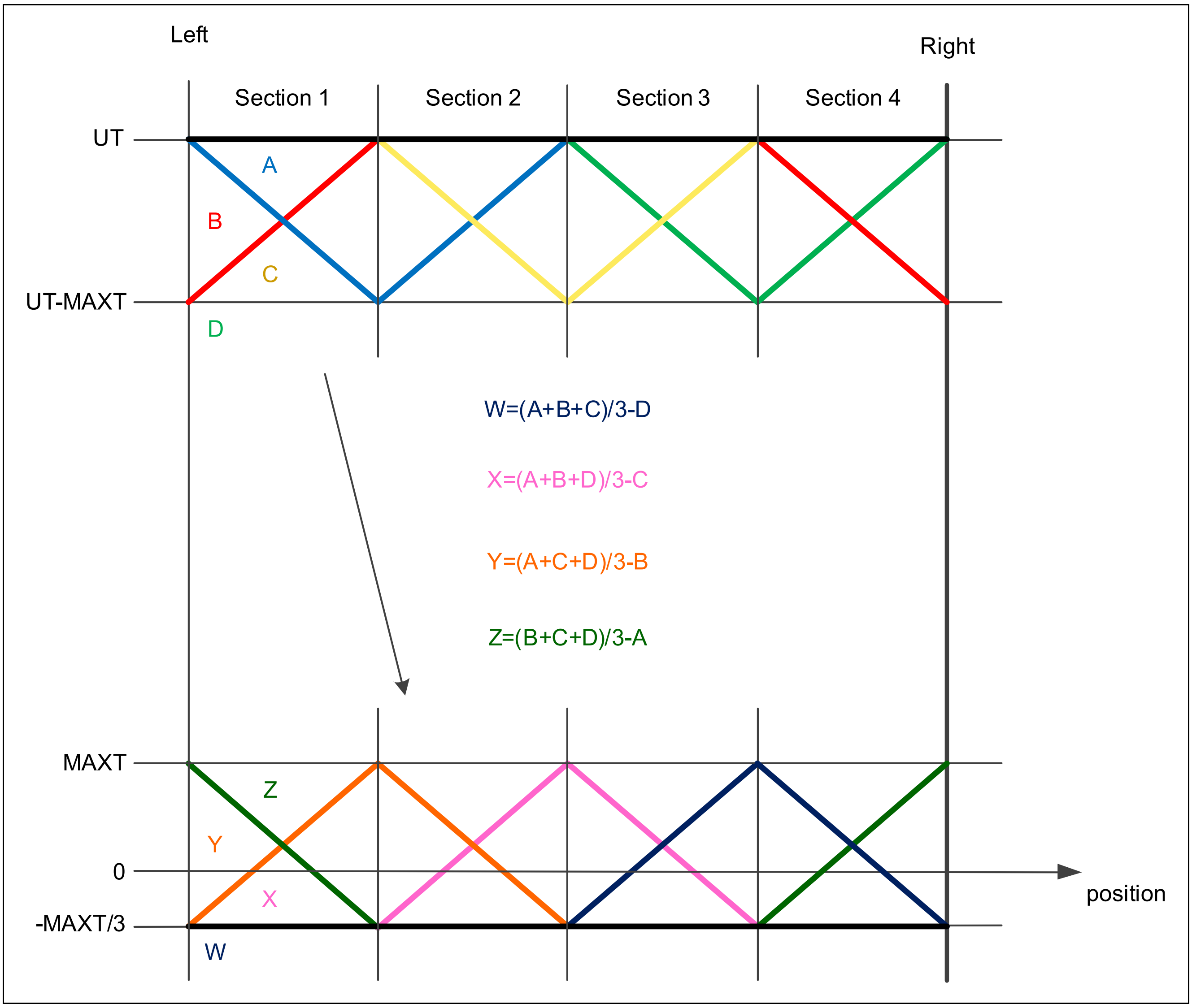

Figure 31 Combined pad average signals

The resulting *W*, *X*, *Y* and *Z* signals still have four distinct sections.

Section 1

Before the transformation, Section 1 has three signals between *UT* and *UT-MAXT* (**Figure 32**). *UT* stands for the untouched level and *UT-MAXT* stands for the signal level when the largest area of the respective pad is touched (this happens at the section borders).

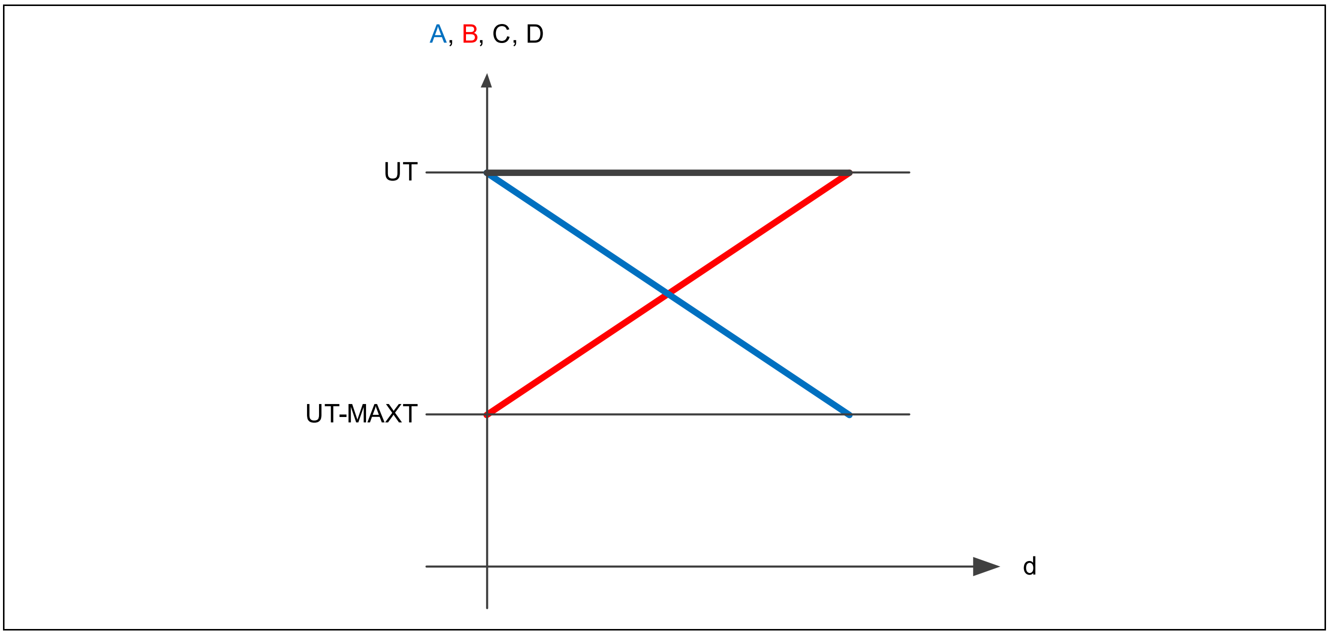

Figure 32 Section 1 before transformation

After the transformation, the *W*, *X*, *Y* and *Z* signals have much lower values (**Figure 33**). The position axis has been arbitrarily scaled from -0.5 to 1.5 in this region for convenience.

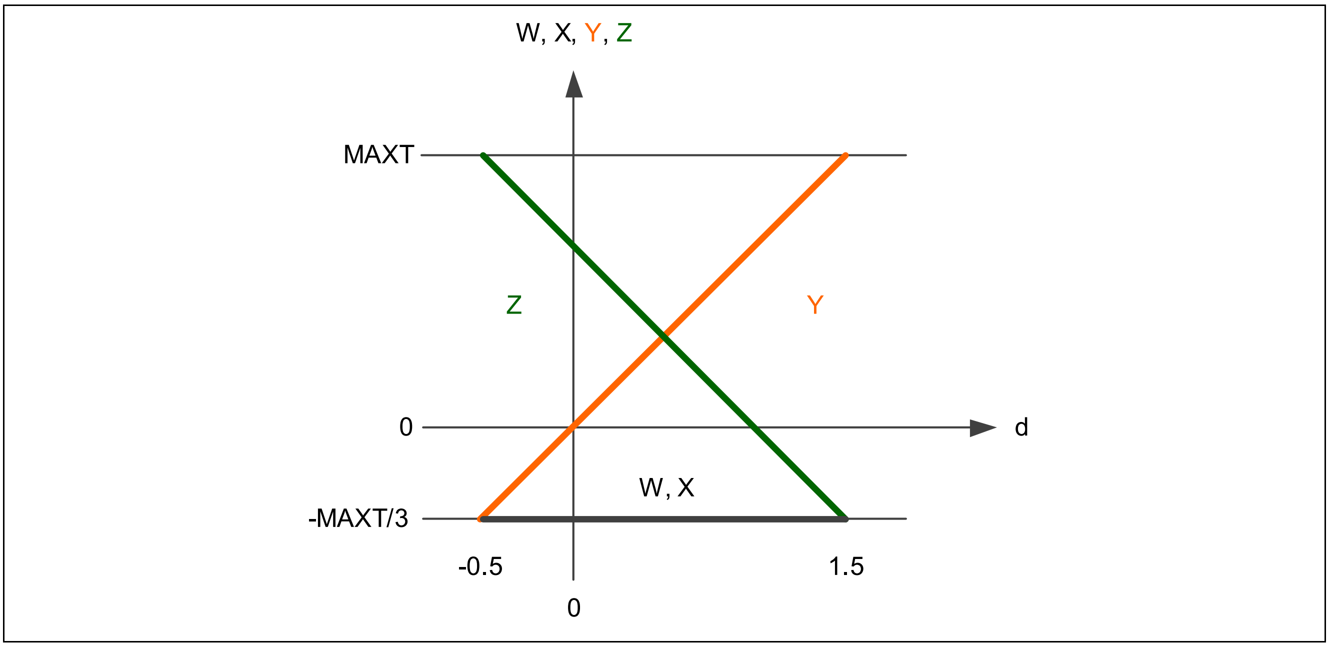

Figure 33 Section 1 after transformation

Signals *W* and *X* are constant low in this section so they do not participate in the position calculation. The other two signals can be described as:

$$Y = 2\frac{MAXT}{3}d \tag{1}$$

$$Z = 2\frac{MAXT}{3} - 2\frac{MAXT}{3}d \tag{2}$$

If we rearrange **Equation (1)**, we get $2\frac{MAXT}{3} = \frac{Y}{d}$ which we can substitute in **Equation (2)**:

$$Z = \frac{Y}{d} - \frac{Y}{d}d$$

$$Z = \frac{Y}{d}(1-d)$$

$$d(Y+Z) = Y$$

$$d = \frac{Y}{Y+Z} \tag{3}$$

$$d = 1 - \frac{Z}{Y+Z} \tag{4}$$

One division is needed to calculate the position; this operation needs the most computing performance. To minimize the error, it is safer to use **Equation (3)** if *Y* is larger and **Equation (4)** if *Z* is larger.

An offset of 0.5 and a scaling factor of 2^R are added to create a more usable calculated position (**Figure 34**). *R* is for resolution and corresponds to the number of left bitshifts on the numerator.

Section 1 Left $$d = 1.5 \times 2^R - \frac{Z \times 2^R}{Y+Z}$$

Section 1 Right $$d = \frac{Y \times 2^R}{Y+Z} + 0.5 \times 2^R$$

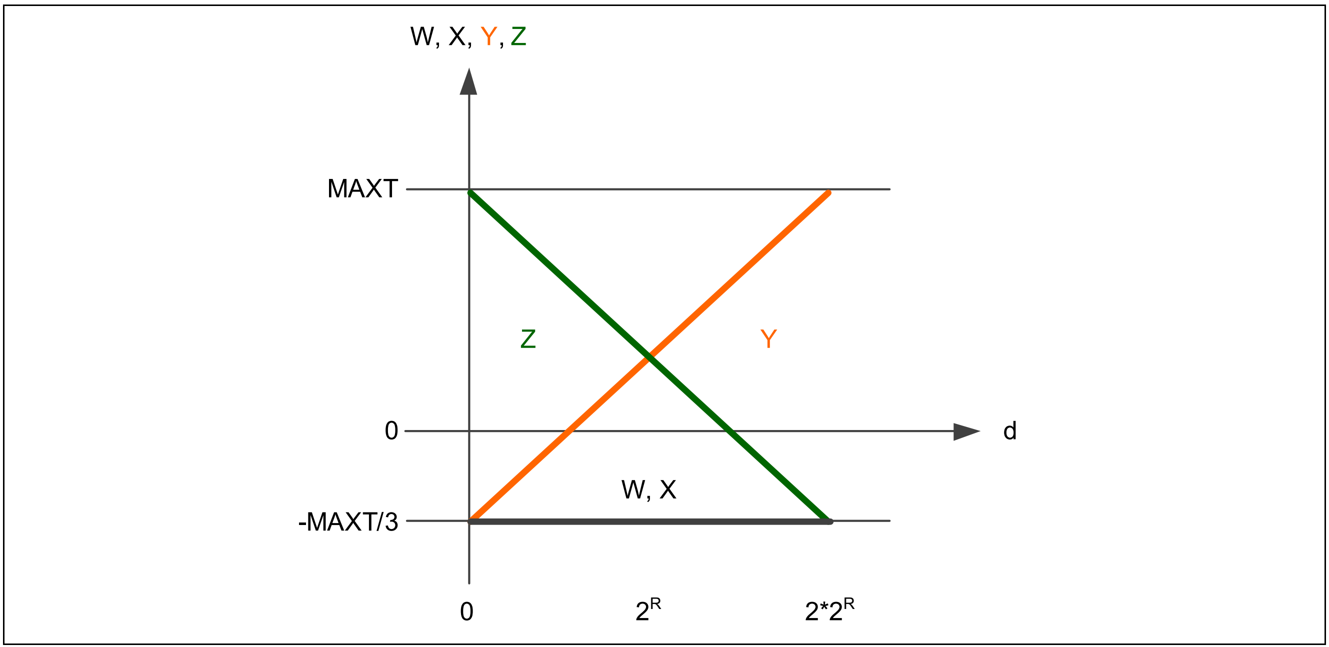

Figure 34 Section 1 after offsetting and scaling

Sections 2, 3 and 4

In these three sections, the position can be calculated in a similar way as in **Section 1**, using the two non-constant signals. Offsets of 2.5, 4.5 and 6.5, and the same scaling factor, can then be added to sections 2, 3 and 4 respectively to get a calculated position of $0..8{*}2^R$.

Section 2 Left $$d = 3.5 \times 2^R - \frac{Y \times 2^R}{X + Y}$$

Section 2 Right $$d = \frac{X \times 2^R}{X + Y} + 2.5 \times 2^R$$

Section 3 Left $$d = 5.5 \times 2^R - \frac{X \times 2^R}{X + Z}$$

Section 3 Right $$d = \frac{Z \times 2^R}{X + Z} + 4.5 \times 2^R$$

Section 4 Left $$d = 7.5 \times 2^R - \frac{X \times 2^R}{X + Z}$$

Section 4 Right $$d = \frac{Z \times 2^R}{X + Z} + 6.5 \times 2^R$$

Figure 35 gives an illustration of the calculated position across all 4 sections while **Figure 36** shows the actual calculated position.

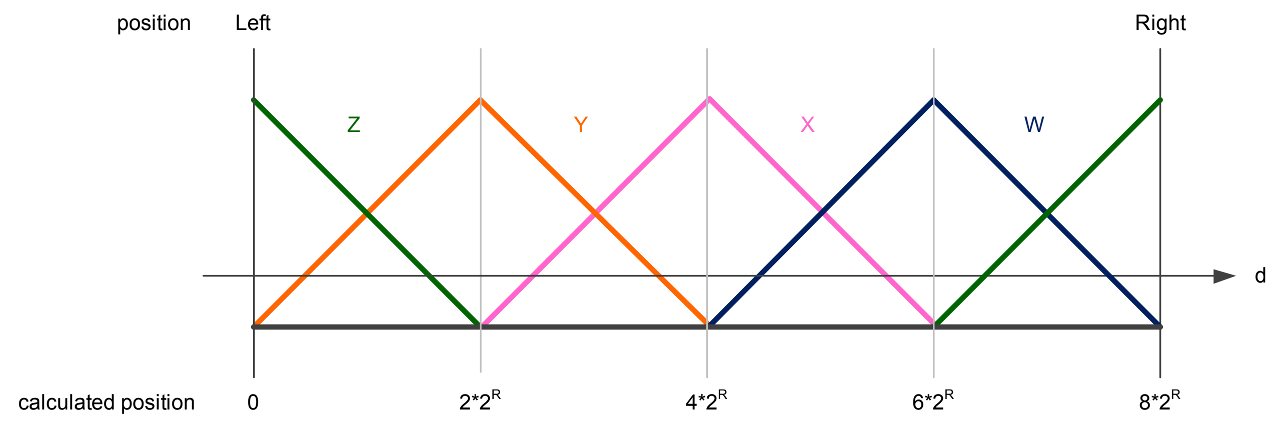

Figure 35 Calculated position vs real position across all sections

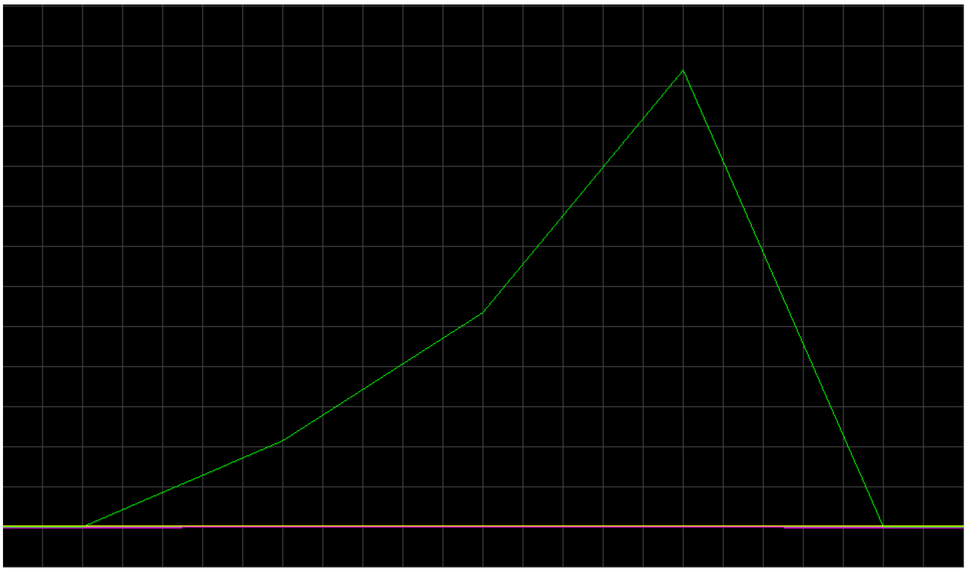

Figure 36 Actual calculated position

3.1.4 Library for Position Calculation

Infineon provides a function library for position calculation. The resolution, which was explained in earlier sections, is user-selectable from 1 to 8. The XC82xMx and XC83xMx microcontrollers have a Multiplication/Division Unit (*MDU*) for hardware acceleration. If the *MDU* is used for the division necessary to calculate the position, the resolution is fixed at 8. The execution is faster and code size is smaller than without hardware acceleration. The disadvantage is that the *MDU* increases the microcontroller's current consumption almost 1mA.

4 U-SPY

Two settings files, **inTouch_Slider.ini** and **inTouch_Slider_II.ini** have been configured for the *inTouch_Slider* and *inTouch_Slider_II* boards respectively.

4.1 inTouch_Slider.ini

This settings file (**Figure 37**) is customized to allow the user to monitor the calculated slider position and the parameters of the Touch Slider Library, while running the demonstration program.

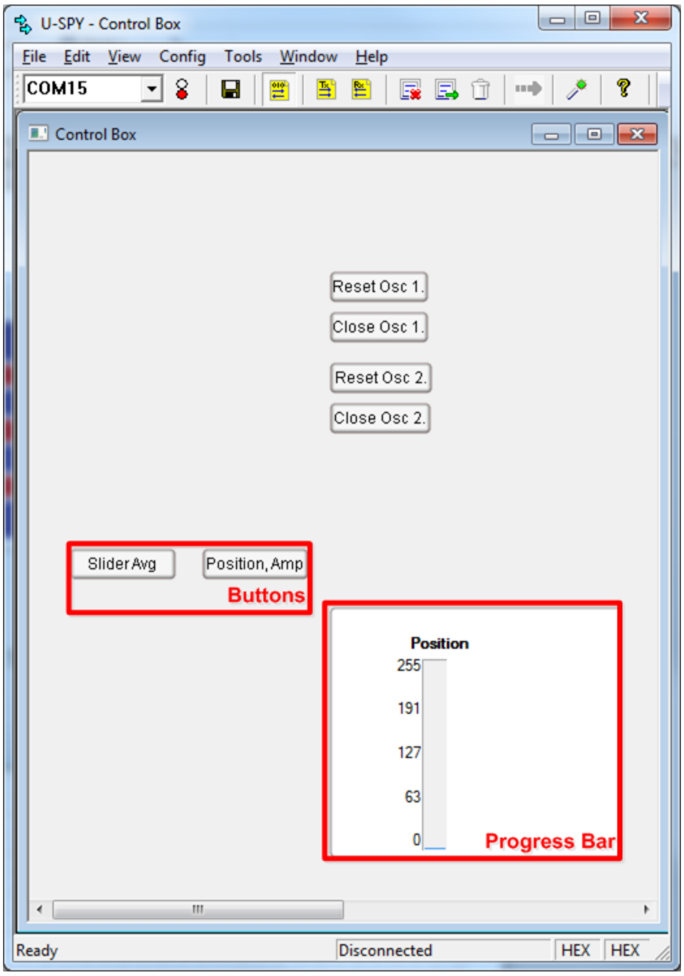

Figure 37 inTouch_Slider.ini User Interface

Buttons

The buttons in this settings file are used to select the signal(s) to be monitored. The format of the data transmitted for the buttons is shown in the following table (**Table 1**):

Table 1 Transmit Data Format for Buttons

	D0	D1
Value (hex)	08	XX
Description	I.D. number	Button number

The data received by the microcontroller will be used to determine the signals that will be transmitted to U-SPY for display on the **Oscilloscope**.

Progress Bar

The progress bar displays the calculated slider position. The format of the transmitted data for the progress bar is as follows (**Table 2**):

Table 2 Transmit Data Format for Progress Bar

	D0	D1	D2	D3
Value (hex)	A2	XX	XX	XX
Description	I.D. number	Progress Bar Index	Calculated Position (High Byte)	Calculated Position (Low Byte)

Oscilloscope

The oscilloscope function allows the user to monitor up to 3 signals at a time (**Figure 38**). A total of 3 oscilloscopes are available. In this application, only 2 oscilloscopes are used. If the “Slider Avg” button is selected, 4 signals will be displayed (3 signals on 1 oscilloscope and 1 signal on another). If the “Position, Amp” button is selected, 2 signals will be displayed on 1 oscilloscope. The format of the transmitted data for the oscilloscope is as follows (**Table 3**):

Figure 38 U-SPY Oscilloscope

Table 3 Transmit Data Format for Oscilloscope

	D0	D1	D2	D3	D4	D5	D6	D7
Value (hex)	A4	01	XX	XX	XX	XX	XX	XX
Description	I.D. number	Scope number	Signal 1 high byte	Signal 1 low byte	Signal 2 high byte	Signal 2 low byte	Signal 3 high byte	Signal 3 low byte

As mentioned in the previous section, the user is able to monitor two different types of signals in this settings file. The signals displayed are as follows (**Table 4**: Slider Avg Mode, **Table 5**: Position, Amp Mode):

Table 4 Signals Displayed for Slider Avg Mode

Oscilloscope 1			
	Signal 1	**Signal 2**	**Signal 3**
Description	Slider_B Current Pad Average	Slider_C Current Pad Average	Slider_D Current Pad Average
Colour	Green	Pink	Yellow
Oscilloscope 2			
	Signal 1	**Signal 2**	**Signal 3**
Description	Slider_A Current Pad Average	None	None
Colour	Green	Pink	Yellow

Table 5 Signals Displayed for Position, Amp Mode

	Signal 1	Signal 2	Signal 3
Description	Slider Position	Slider Amplitude	None
Colour	Green	Pink	Yellow

4.2 inTouch_Slider_II.ini

This settings file (**Figure 39**) is customized to allow the user to monitor the calculated slider position and the parameters of the Touch Slider Library, while running the demonstration program.

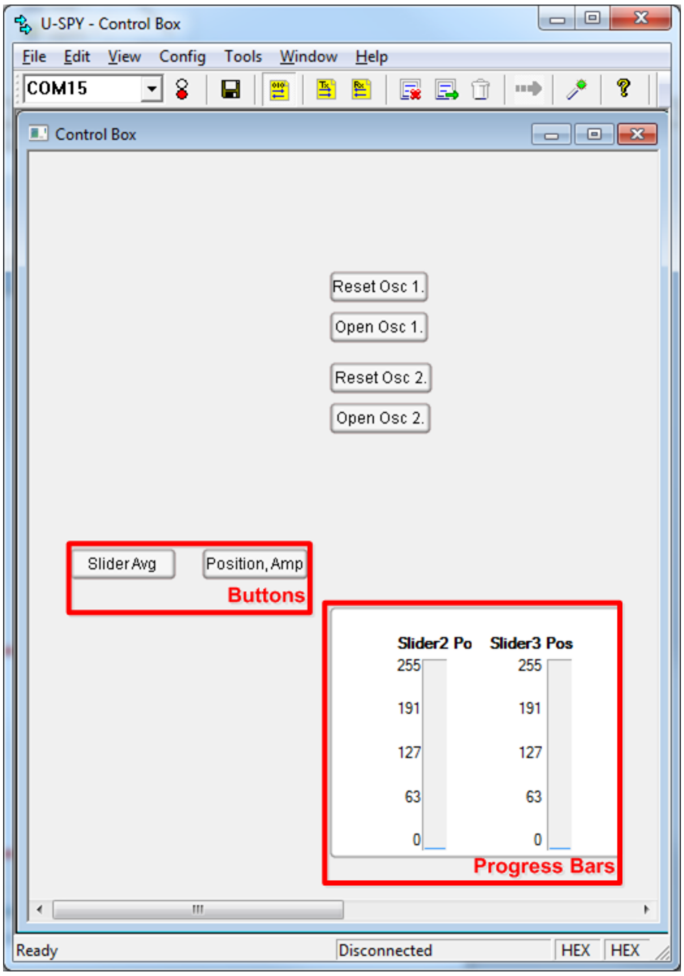

Figure 39 inTouch_Slider_II.ini User Interface

Buttons

The buttons in this settings file are used to select the signal(s) to be monitored. The format of the data transmitted for the buttons is shown in the following table (**Table 6**):

Table 6 Transmit Data Format for Buttons

	D0	D1
Value (hex)	08	XX
Description	I.D. number	Button number

The data received by the microcontroller will be used to determine the signals that will be transmitted to U-SPY for display on the **Oscilloscope**.

Progress Bars

The progress bars display the calculated slider positions for the 2-pad and 3-pad sliders. The format of the transmitted data for the progress bar is as follows (**Table 7**):

Table 7 Transmit Data Format for Progress Bar

	D0	D1	D2	D3
Value (hex)	A2	XX	XX	XX
Description	I.D. number	Progress Bar Index	Calculated Position (High Byte)	Calculated Position (Low Byte)

Oscilloscope

The oscilloscope function allows the user to monitor up to 3 signals at a time (**Figure 40**). A total of 3 oscilloscopes are available. In this application, only 2 oscilloscopes are used. If the "Slider Avg" button is selected, 5 signals will be displayed (2 signals on 1 oscilloscope for 2-pad slider and 3 signals on another for 3-pad slider). If the "Position, Amp" button is selected, 4 signals will be displayed on 2 oscilloscope (2 signals each). The format of the transmitted data for the oscilloscope is as follows (**Table 8**):

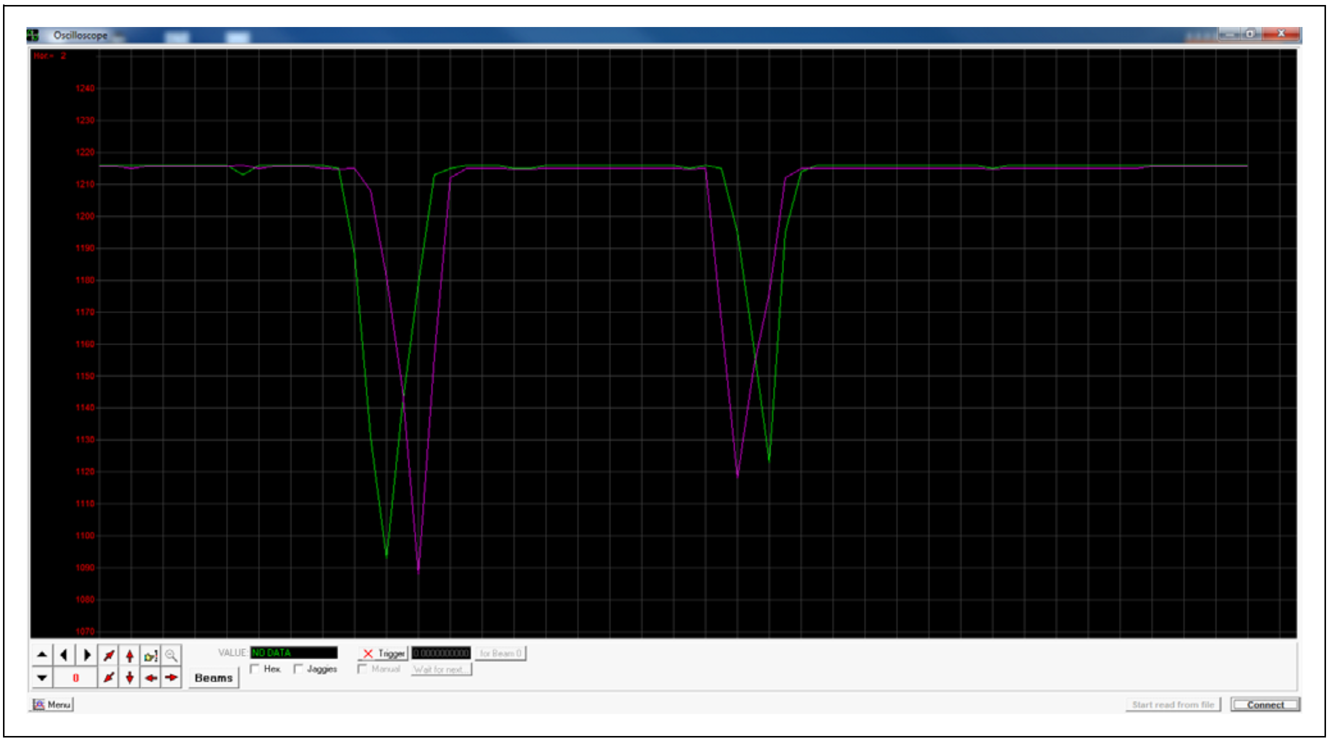

Figure 40 U-SPY Oscilloscope

Table 8 Transmit Data Format for Oscilloscope

	D0	D1	D2	D3	D4	D5	D6	D7
Value (hex)	A4	01	XX	XX	XX	XX	XX	XX
Description	I.D. number	Scope number	Signal 1 high byte	Signal 1 low byte	Signal 2 high byte	Signal 2 low byte	Signal 3 high byte	Signal 3 low byte

As mentioned in the previous section, the user is able to monitor two different types of signals in this settings file. The signals displayed are as follows (**Table 4**: Slider Avg Mode, **Table 5**: Position, Amp Mode):

Table 9 Signals Displayed for Slider Avg Mode

Oscilloscope 1			
	Signal 1	**Signal 2**	**Signal 3**
Description	Slider_D Current Pad Average (2-pad Slider)	Slider_E Current Pad Average (2-pad Slider)	None
Colour	Green	Pink	Yellow
Oscilloscope 2			
	Signal 1	**Signal 2**	**Signal 3**
Description	Slider_A Current Pad Average (3-pad Slider)	Slider_B Current Pad Average (3-pad Slider)	Slider_C Current Pad Average (3-pad Slider)
Colour	Green	Pink	Yellow

Table 10 Signals Displayed for Position, Amp Mode

Oscilloscope 1			
	Signal 1	**Signal 2**	**Signal 3**
Description	2-pad Slider Position	2-pad Slider Amplitude	None
Colour	Green	Pink	Yellow
Oscilloscope 2			
	Signal 1	**Signal 2**	**Signal 3**
Description	3-pad Slider Position	3-pad Slider Amplitude	None
Colour	Green	Pink	Yellow

Appendix - Schematics and Layout

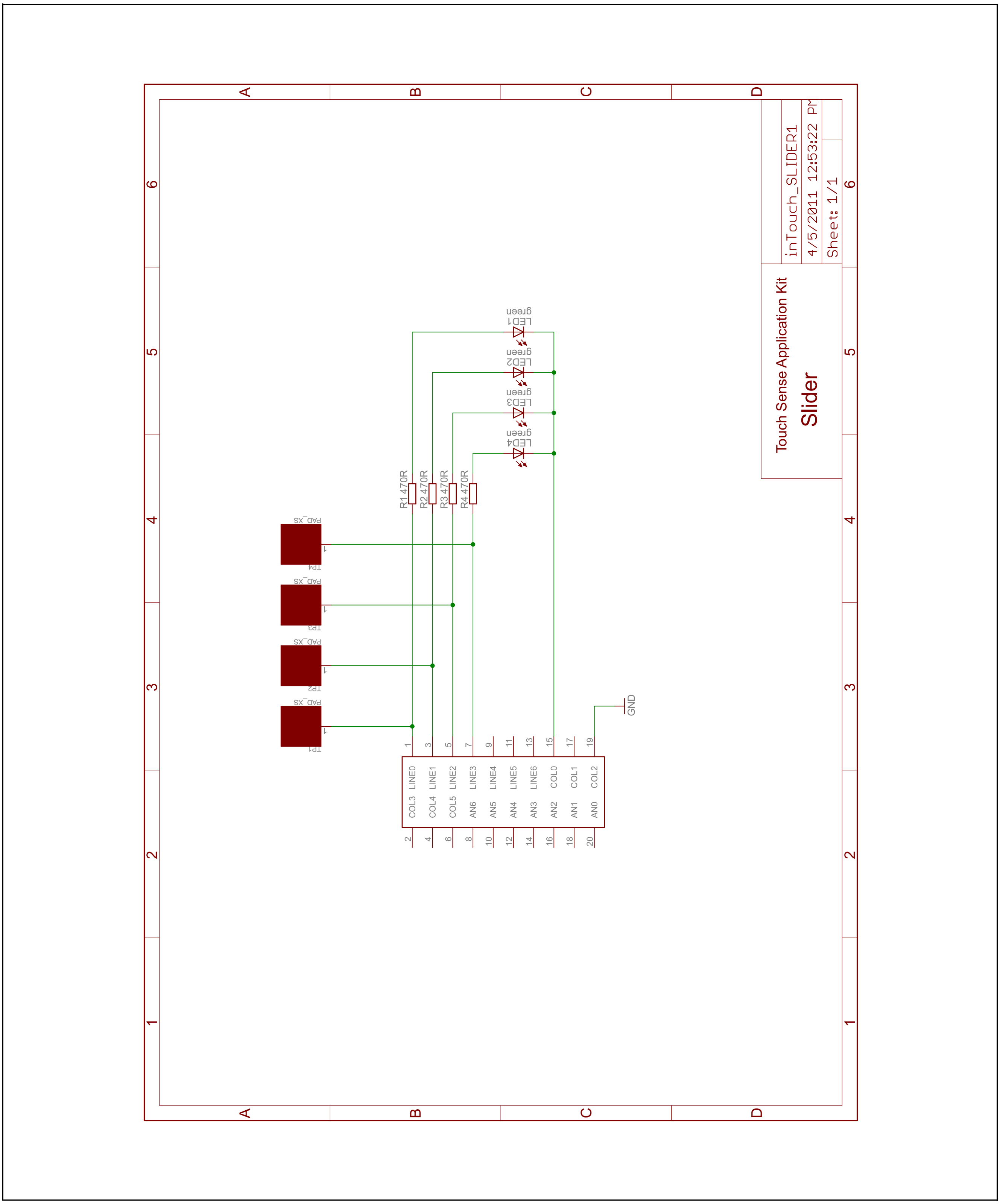

Figure 41 ***inTouch Slider*** **Board Schematics**

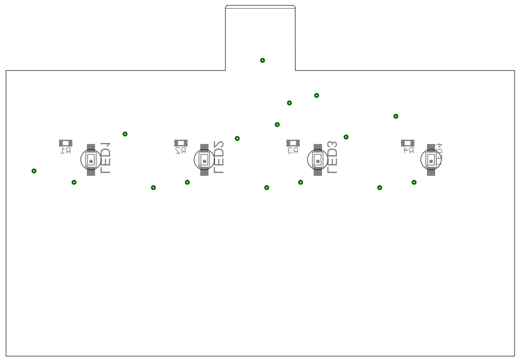

Figure 42 *inTouch Slider* Board Componenet Bottom Layout

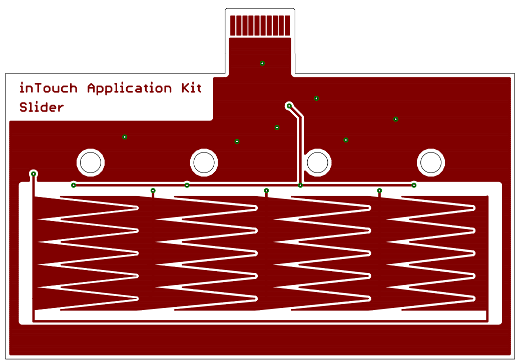

Figure 43 *inTouch Slider* Board Top Layout

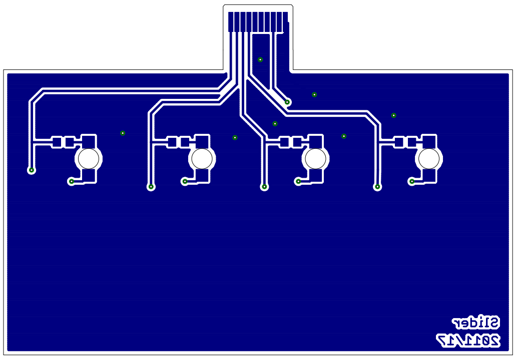

Figure 44 ***inTouch Slider*** **Board Bottom Layout**

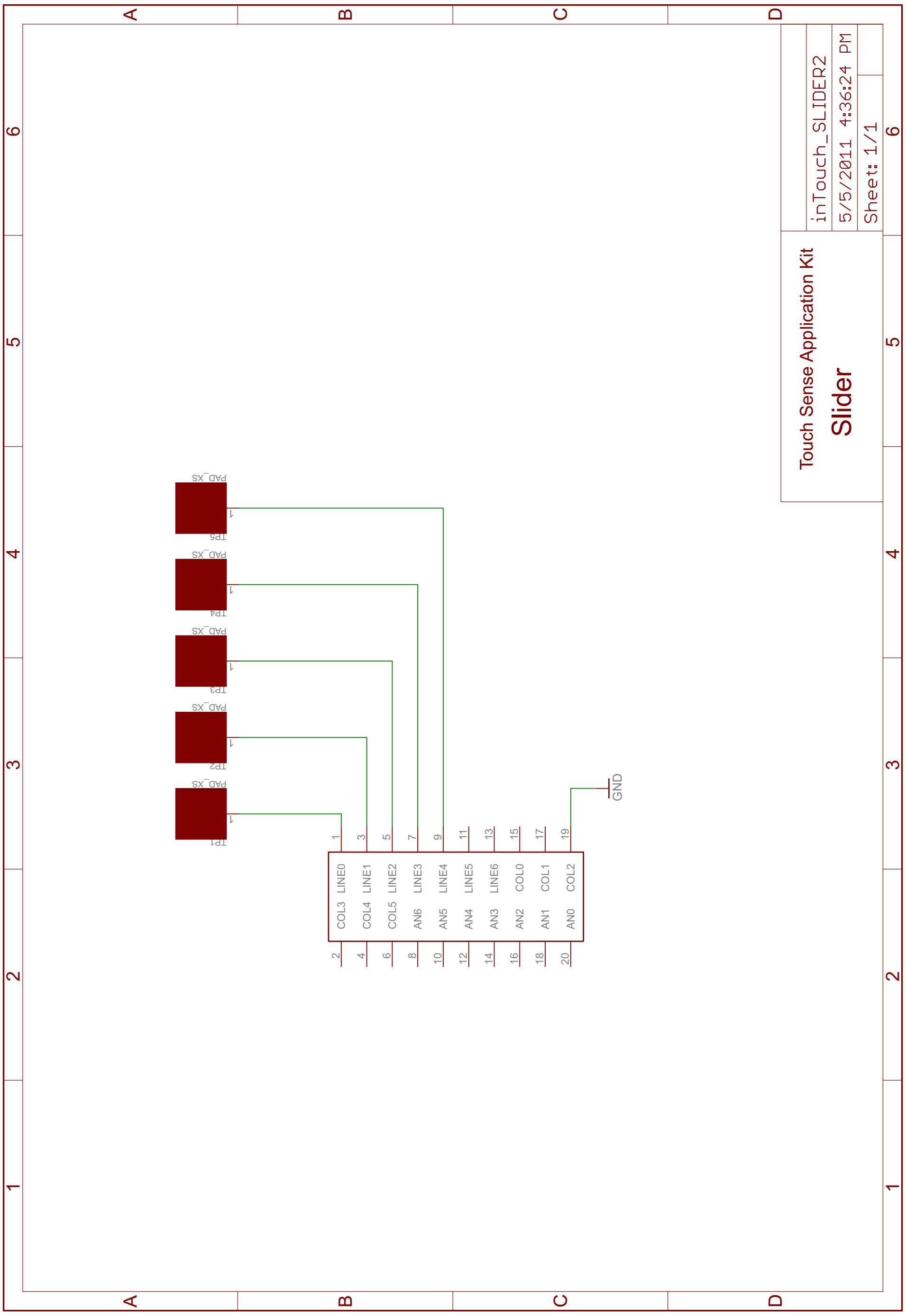

Figure 45 ***inTouch Slider II*** **Board Schematics**

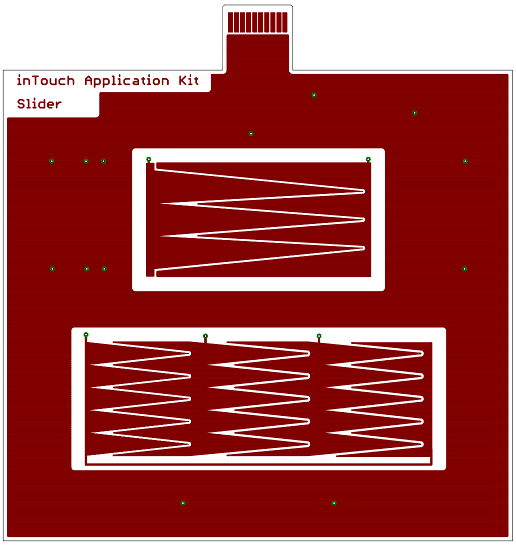

Figure 46 ***inTouch Slider II* Board Top Layout**

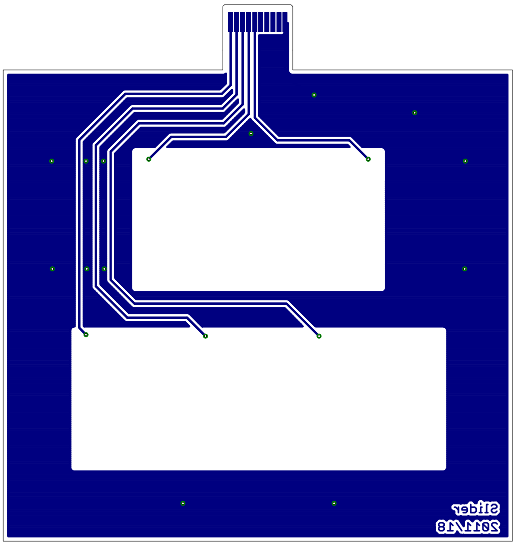

Figure 47 ***inTouch Slider II* Board Bottom Layout**

References

The list below provides resources that may be useful to the user.

1. User’s Manual - *XC83x; 8-Bit Single-Chip Microcontroller*
2. Application Note - *AP08100* - *Configuration for Capacitive Touch-Sense Application*
3. Application Note - *AP08110* - *Design Guidelines for XC82x and XC83x Board Layout*
4. Application Note - *AP08113* - *Capacitive-Touch Color Wheel Implementation*
5. Application Note - *AP08115* - *Design Guidelines for Capacitive Touch-Sensing Application*
6. Application Note - *AP08121* - *Infrared Remote Controller with Capacitive Touch Interface*
7. Application Note - *AP08122* - *16-Button Capacitive Touch Interface with XC836T*
8. Application Note - *AP08124* - *XC82/83x Design Guidelines for Electrical Fast Transient (EFT) Protection in Touch-Sense Applications*
9. Application Note - *AP08126* - *Infineon Touch Solutions - inTouch Application Kit*
10. Application Note - *AP08127* - *inTouch Application Kit - Buttons*
11. Application Note - *AP08128* - *inTouch Application Kit - Touch Wheel*
12. Application Note - *AP08130* - *inTouch Application Kit - LED Matrix*
13. Link to *XC83x-Series* - **www.infineon.com/xc83x**
14. Link to *Solutions for advanced touch control* - **www.infineon.com/intouch**

w w w . i n f i n e o n . c o m

Published by Infineon Technologies AG