



**MICROCHIP**

**Additional Information for  
PIC CVD and CTMU Demos**

**14 November 2011**

***Michael W. Mann***  
***HPMD 32-Bit Application Engineer***



# Real-Time Debugging

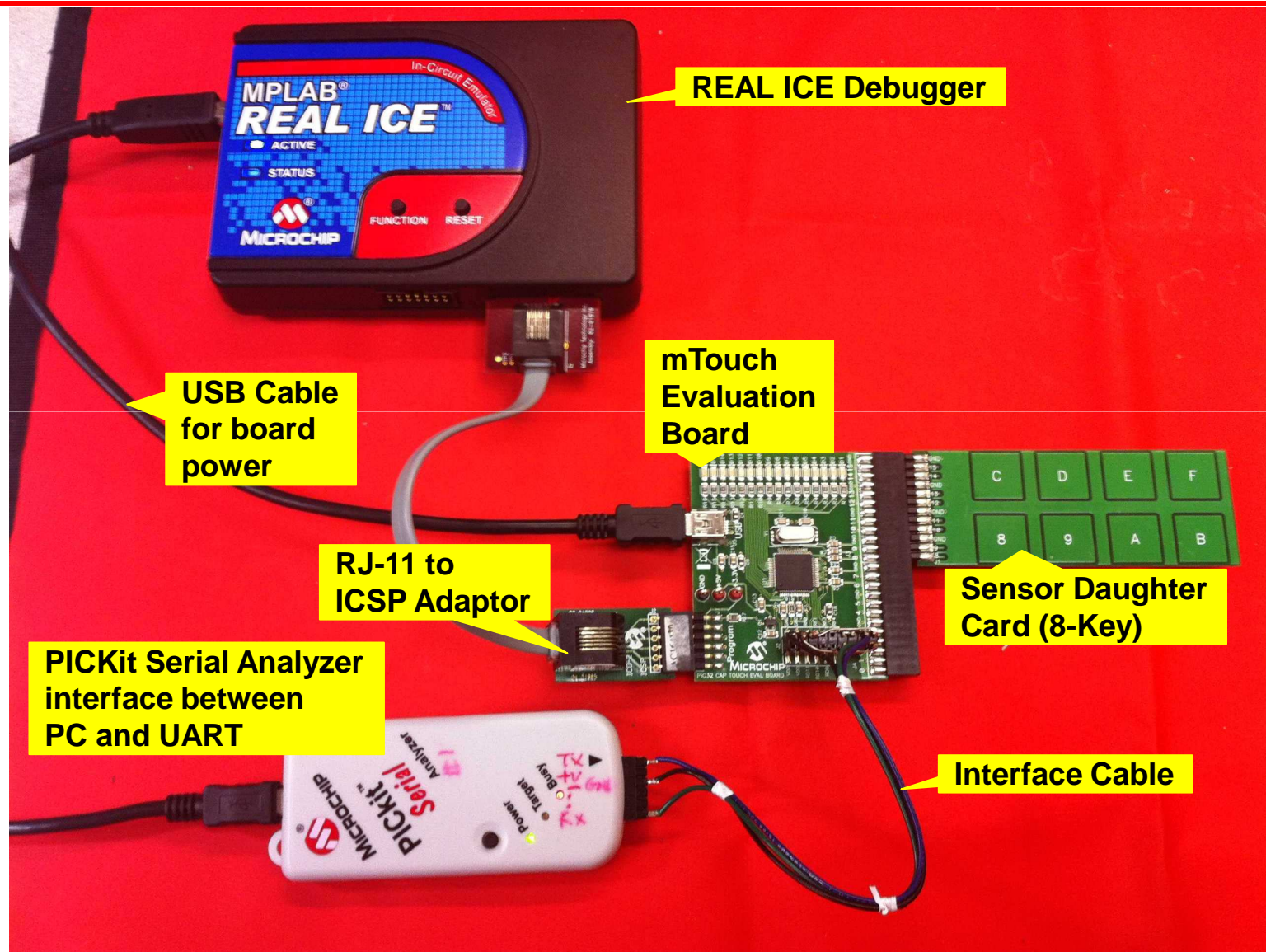
- To assist in debugging button/slider behavior, Microchip has provided Profilab GUI project for use on your PC
- Profilab™ is like LabView™, but much cheaper

<http://www.abacom-online.de/uk/html/profilab-expert.html>

- Standalone Windows mTouch GUI.exe allows use of GUI without purchasing Profilab™
- Profilab™ projects can be customized to meet your debugging needs
  - Plot average voltage and button voltage
  - Model product front panel on a GUI tab



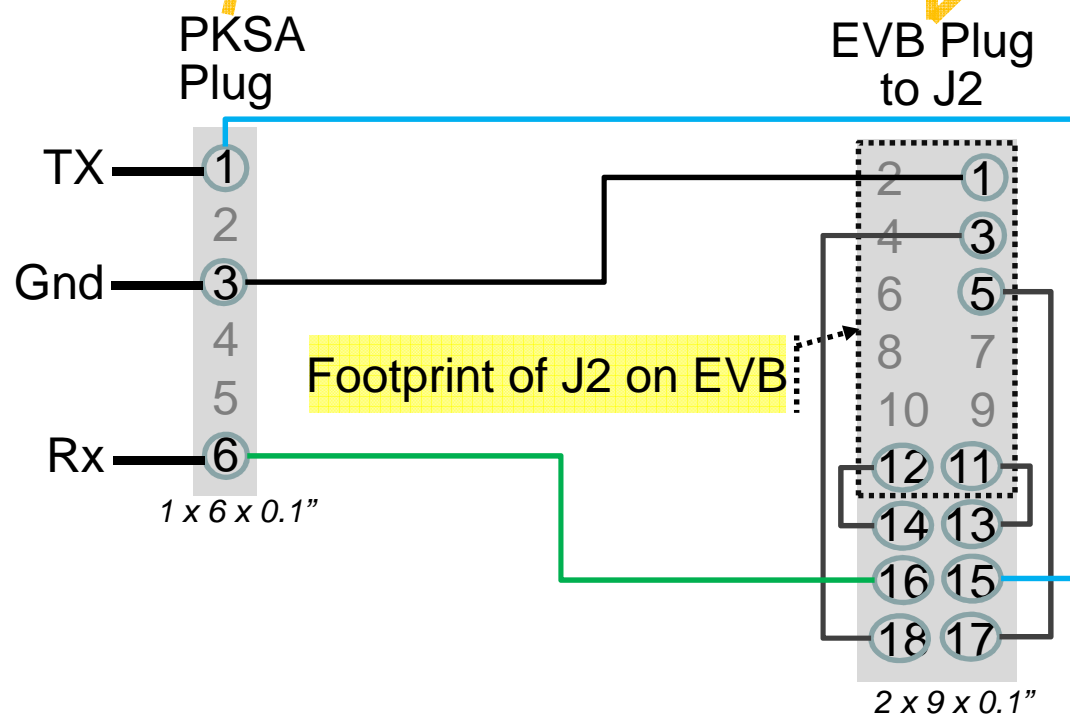
# Typical Bench-Top Setup







# PKSA-EVB J2 Interface Cable



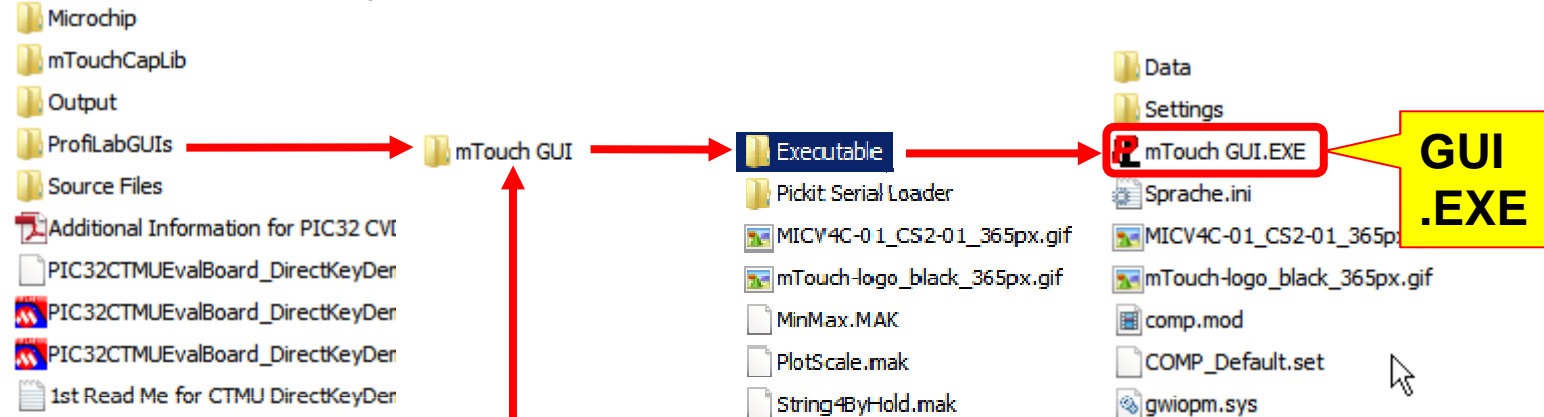
## Jumper Settings

Most Boards	PIC32 CTMU	PIC18F CVD
2 ①	2 ①	2 ①
4 ③	4 ③	4 ③
6 ⑤	6 ⑤	6 ⑤
8 7	8 7	8 7
10 9	10 9	10 9
12 11	12 11	12 11
14 13	14 13	14 13
16 15	16 15	16 15
18 17	18 17	18 17

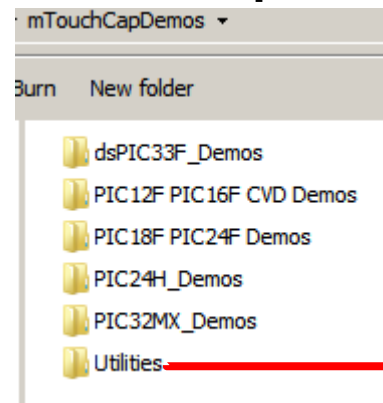


# Finding the .EXE GUI File

## Standalone Project

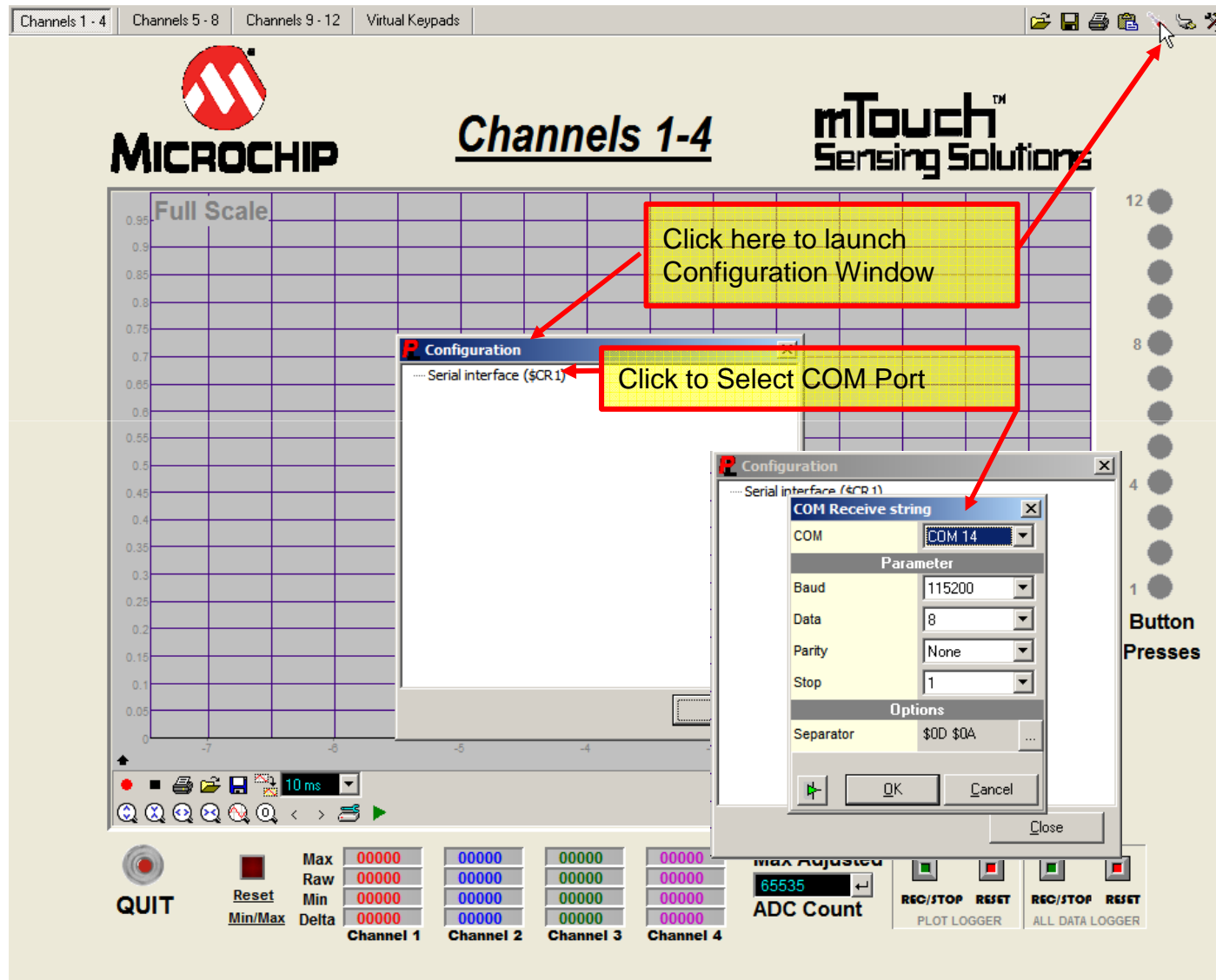


## Microchip Solutions mTouchCapDemos Directory



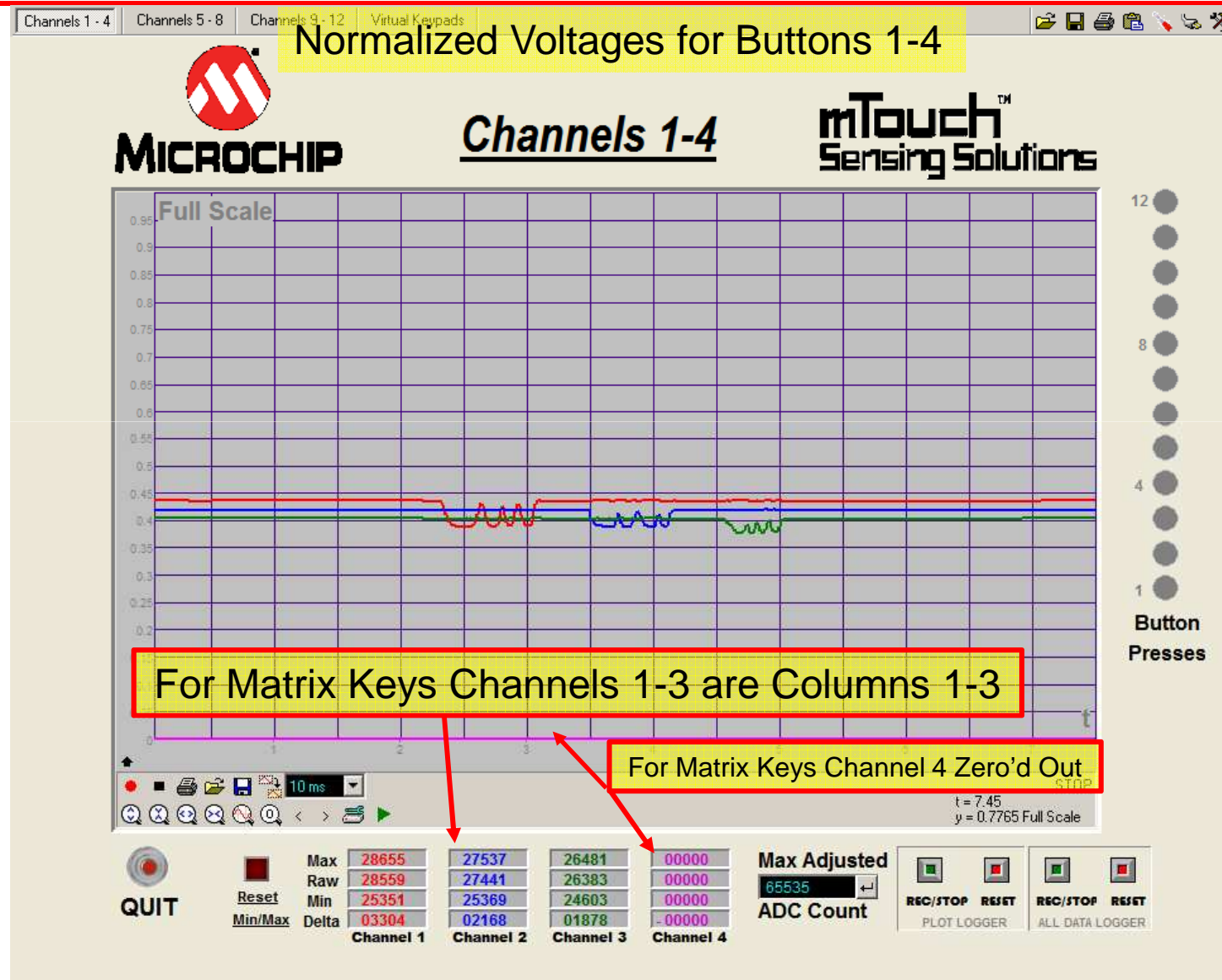


# Setting Up GUI Com Params



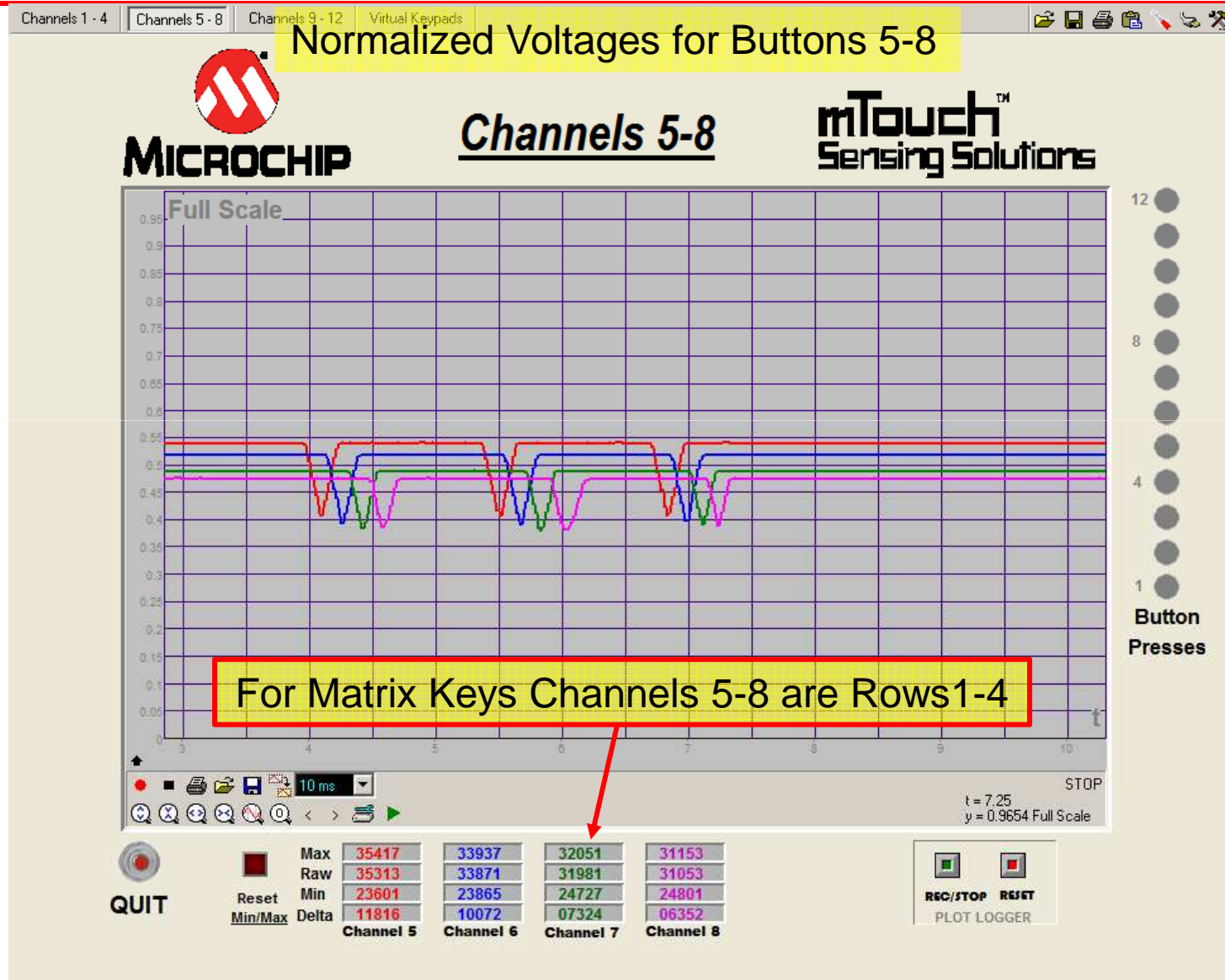


# mTouch GUI Screen 1/3





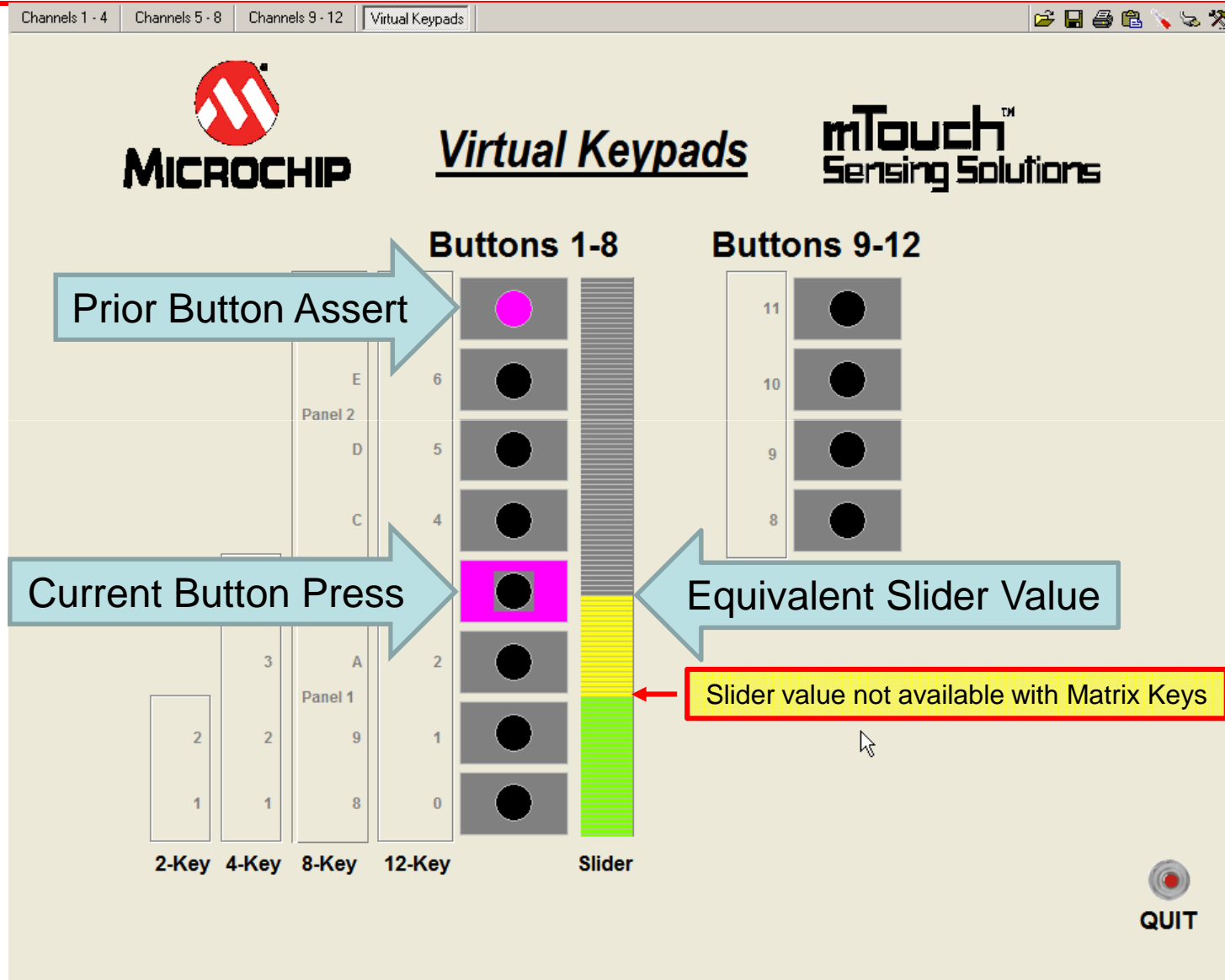
# mTouch GUI Screen 2/3







# mTouch GUI Screen 3/3





# Voltage Plot Screen Controls

Quit data collection and exit GUI

Reset Min/Max for all Channels

Maximum ADC Count

Save voltage plot data to ASCII file

Save all UART data to ASCII file

QUIT

Reset Min/Max

	Channel 1	Channel 2	Channel 3	Channel 4
Max	28655	27537	26481	00000
Raw	28559	27441	26383	00000
Min	25351	25369	24603	00000
Delta	03304	02168	01878	-00000

Delta = Max - Min

Max Adjusted ADC Count

Record or Stop Recording

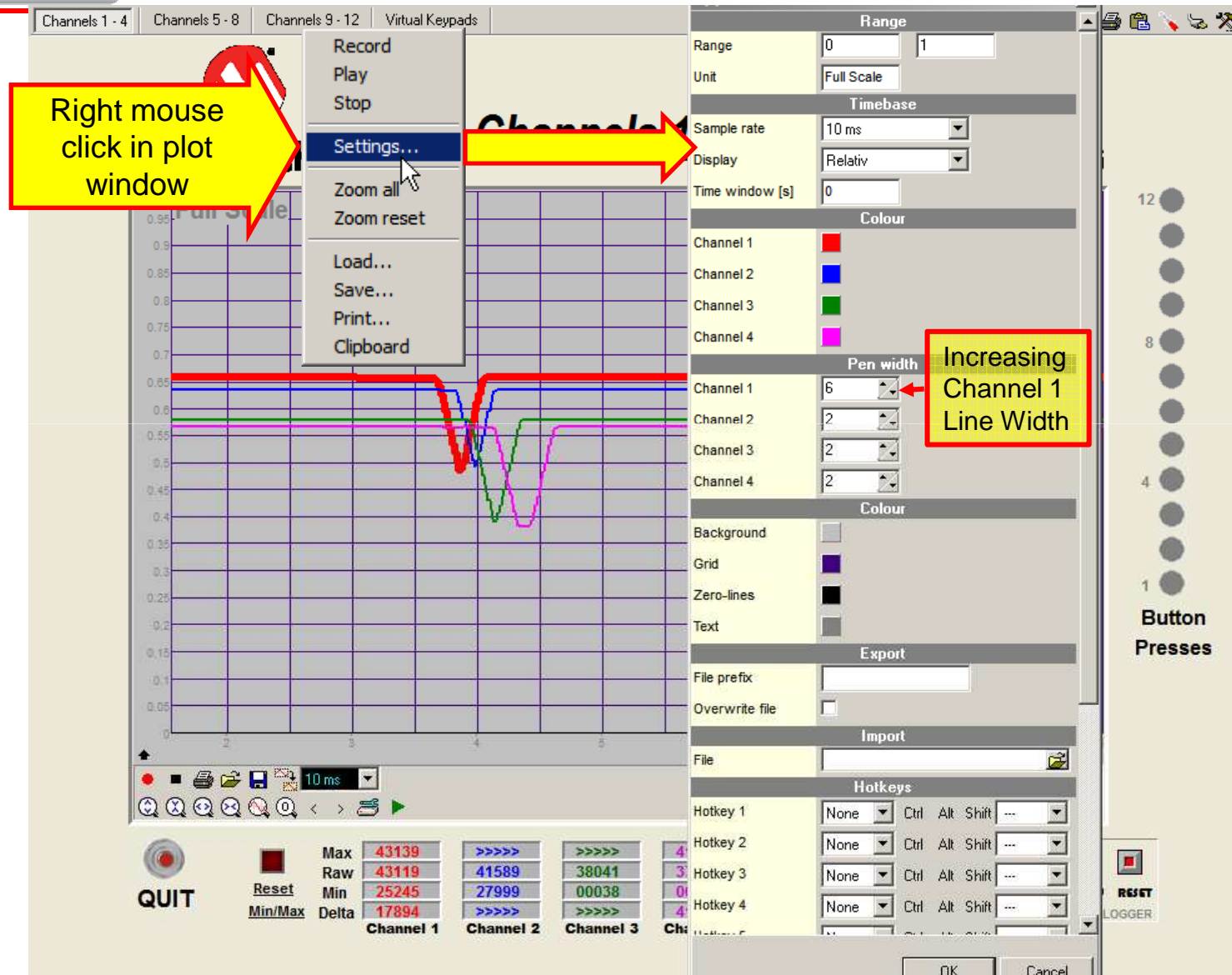
REC/STOP RESET PLOT LOGGER

REC/STOP RESET ALL DATA LOGGER

Erase file and start over

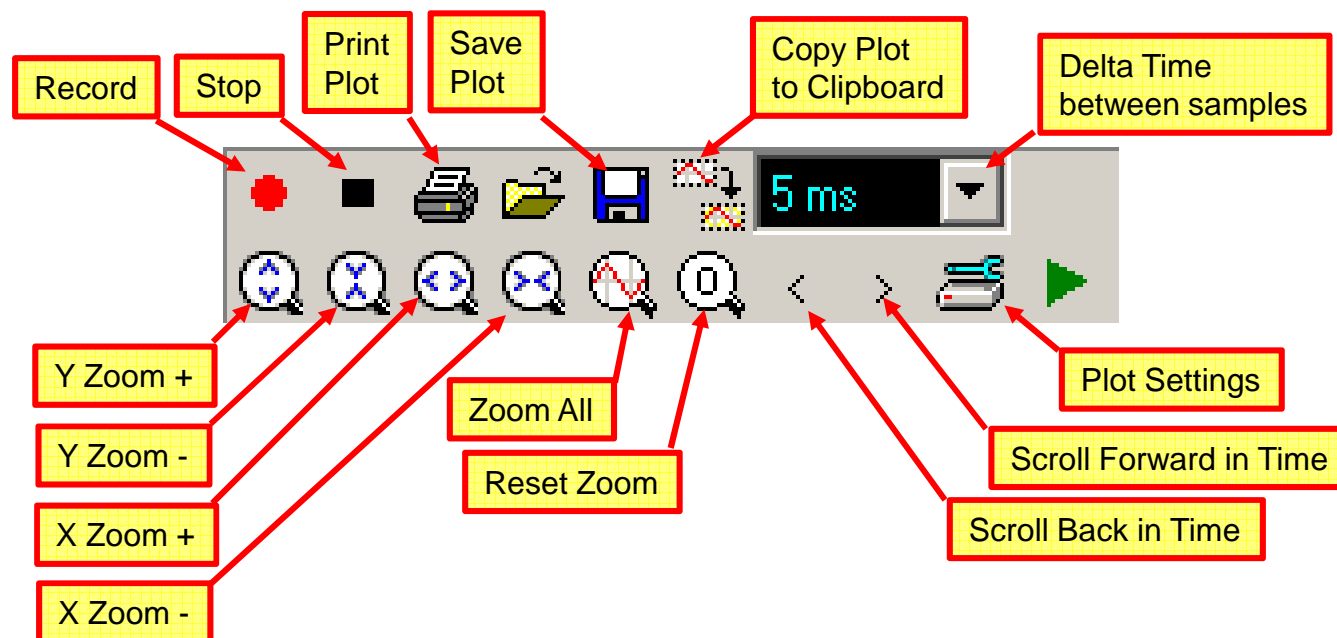


# Configuring Voltage Plots



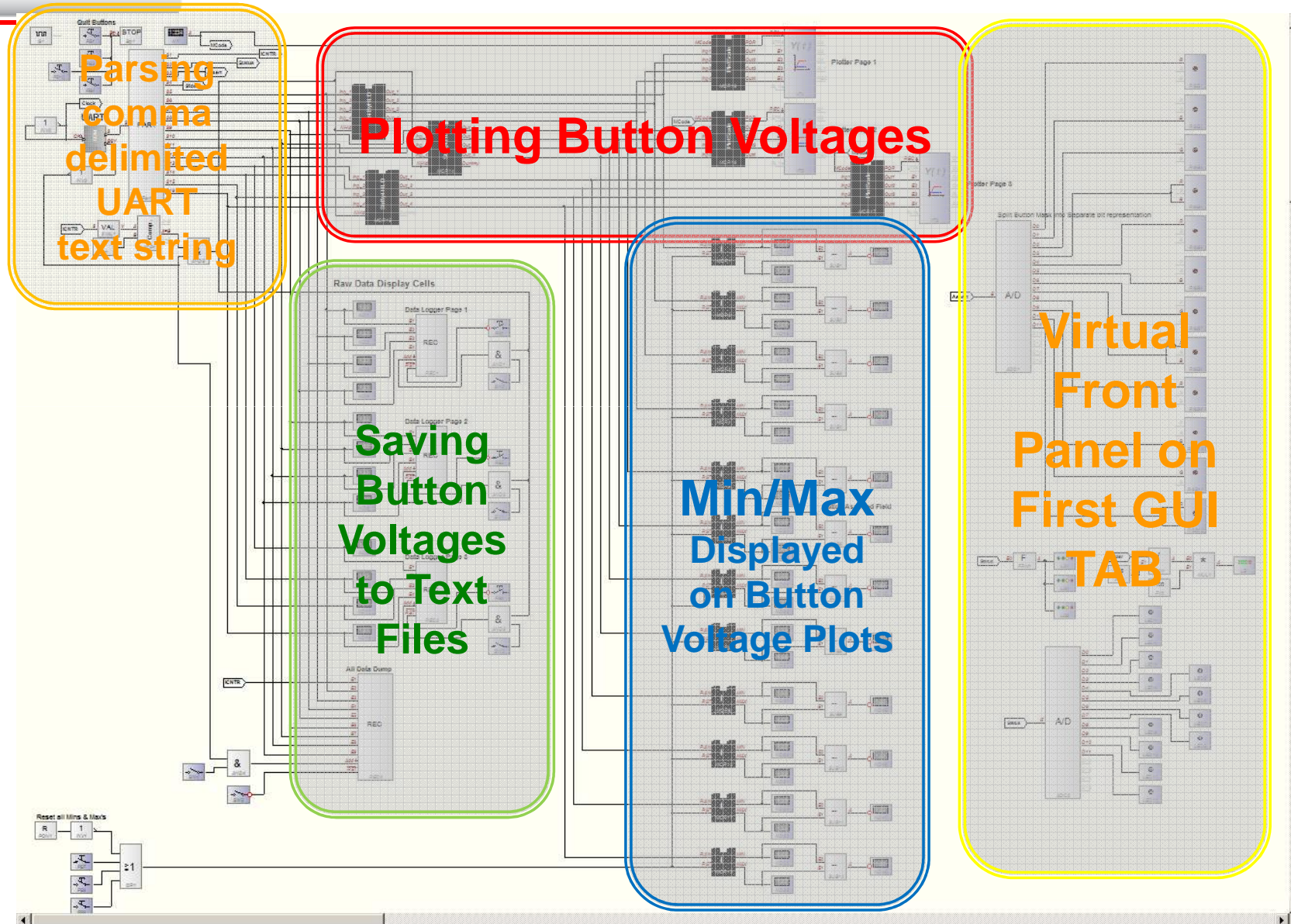


# Plot Controls





# Profilab™ = LabView™ Lite

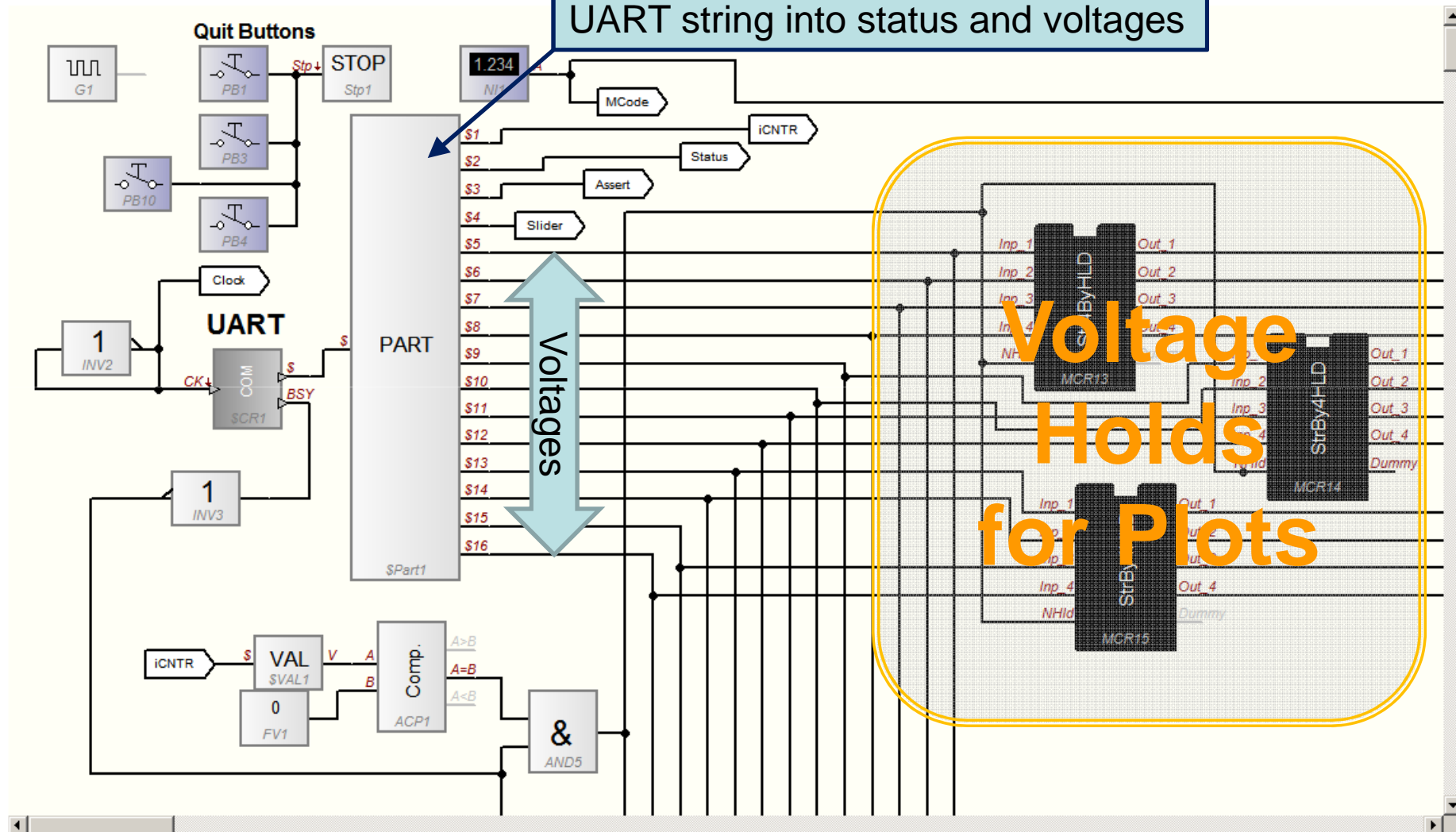




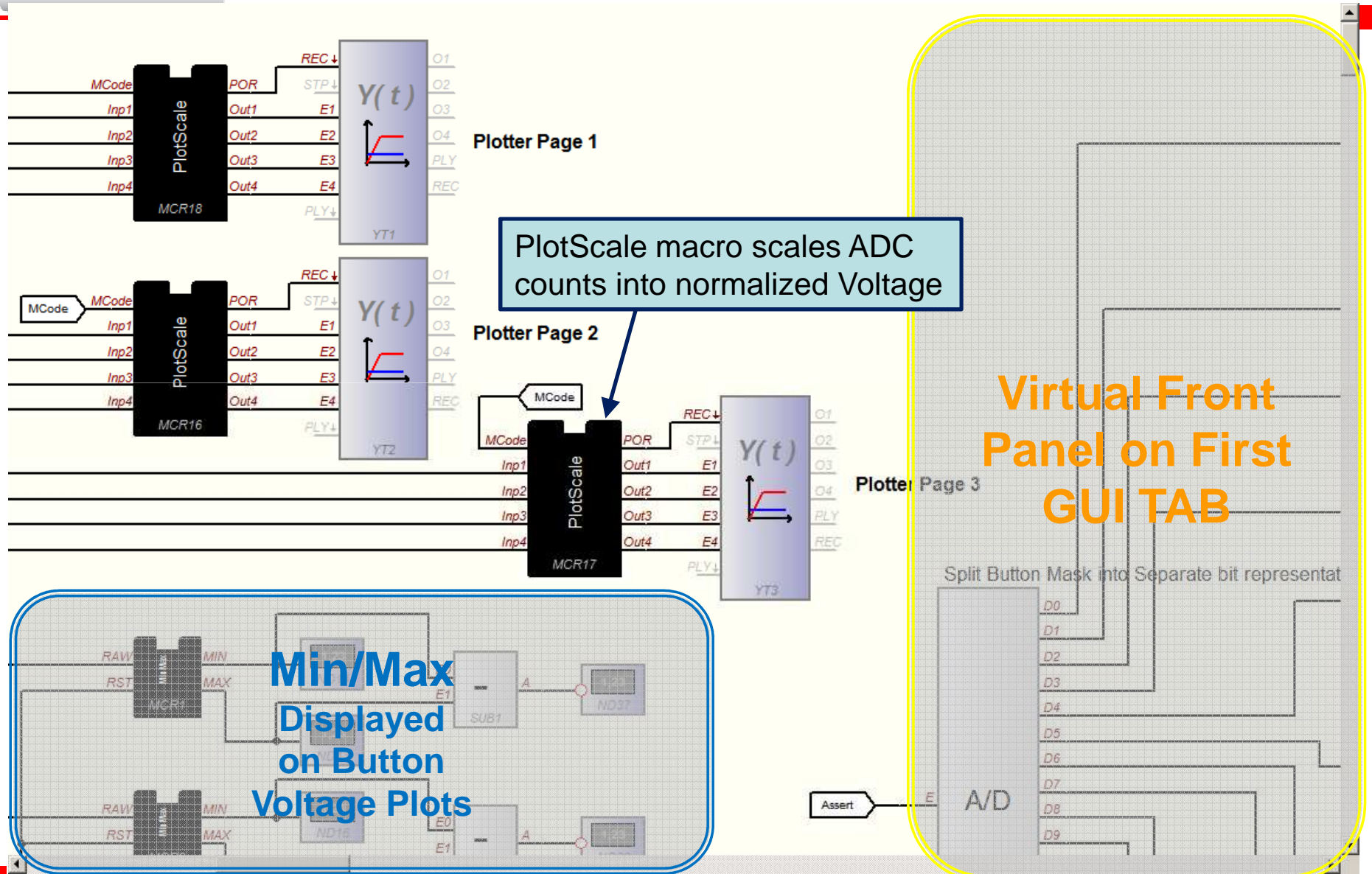


# Parsing Comma Delimited Data

Block decodes comma delimited UART string into status and voltages

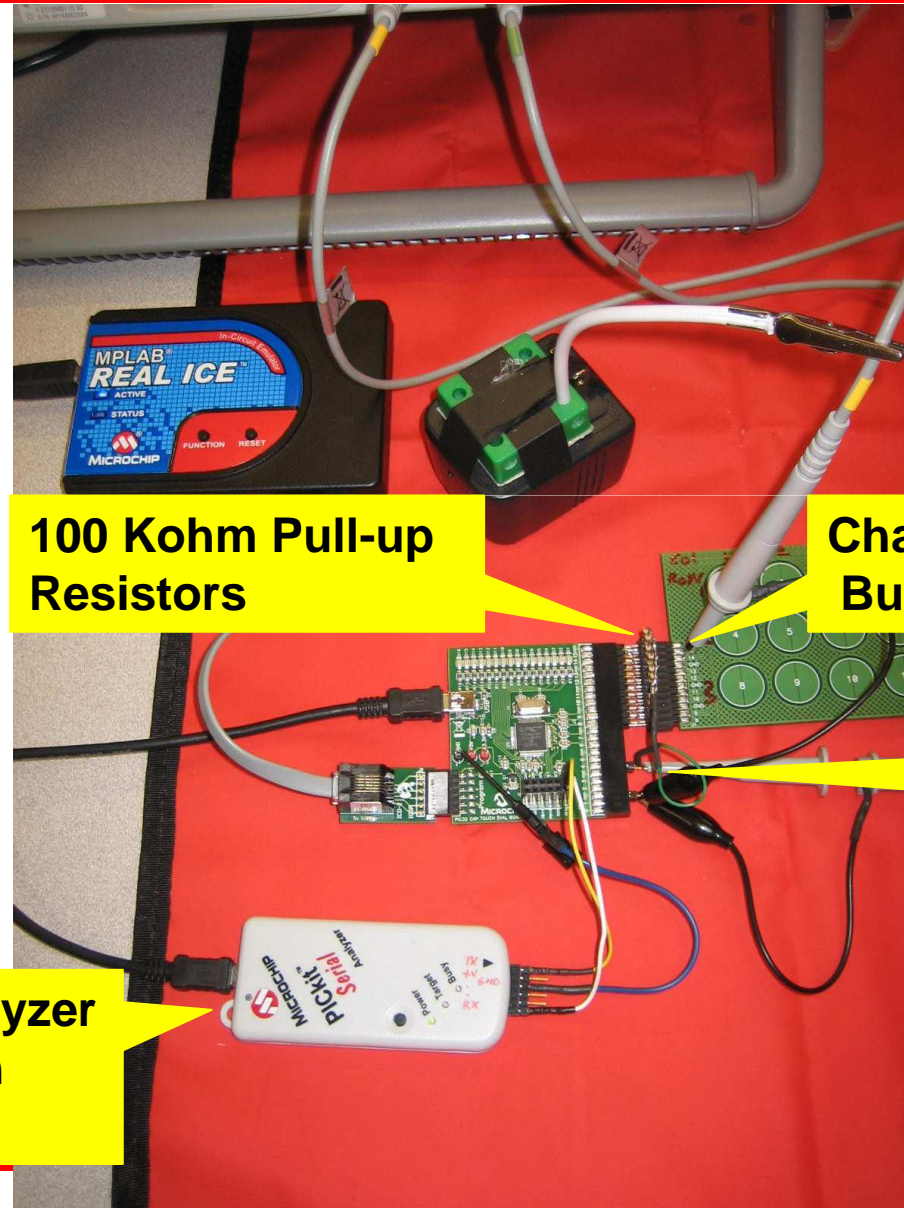


# Scale and Plot Channels





# Benchtop Setup for Differential with Pull-Ups Demo



100 Kohm Pull-up Resistors

Channel 1 on Button Signal

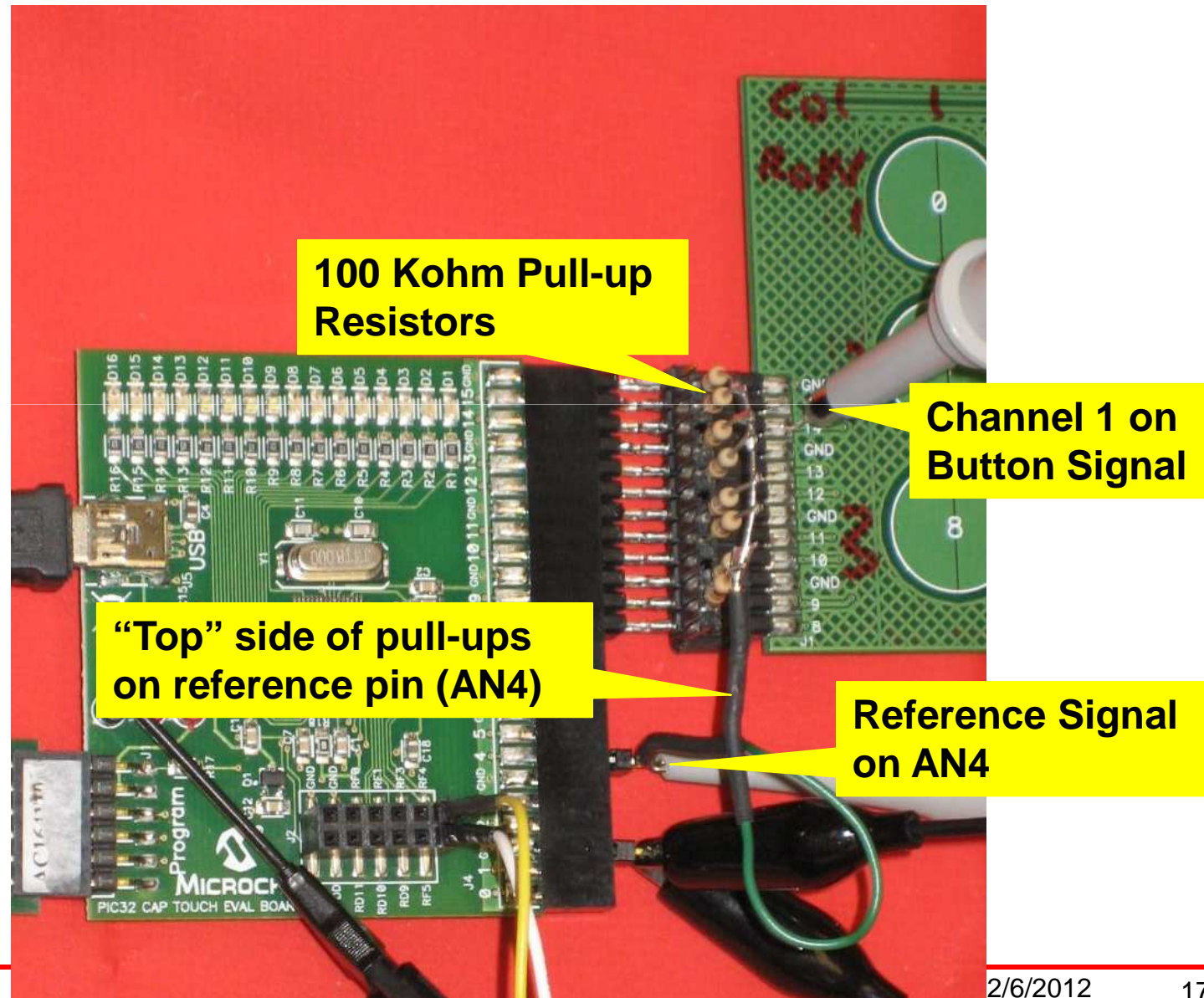
Reference Signal on AN4

PICKit Serial Analyzer interface between PC and UART





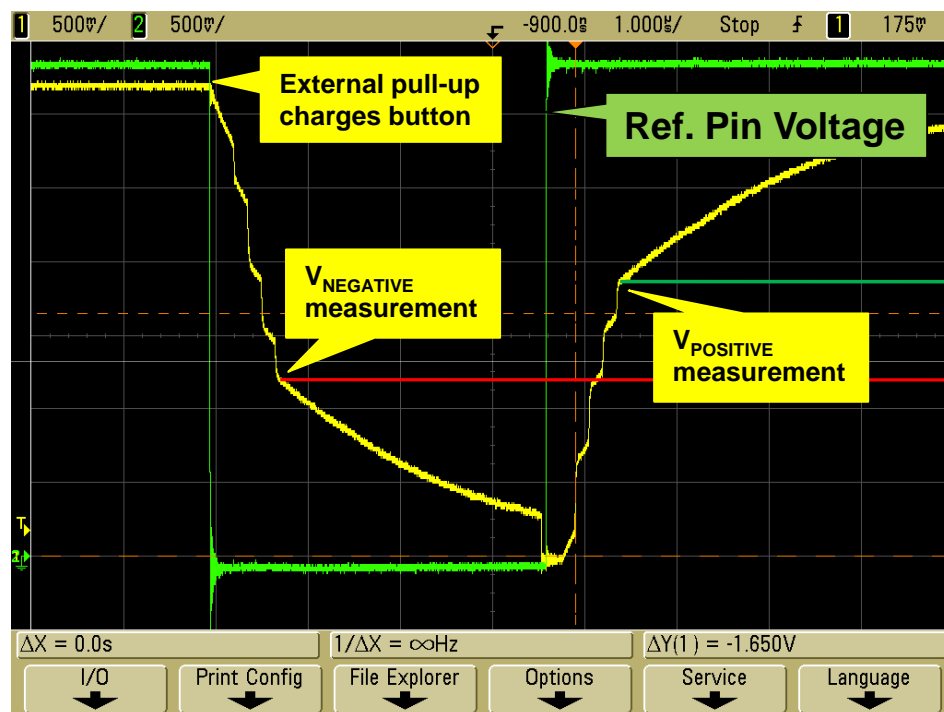
# Close-Up of Differential Setup using Pull-Ups



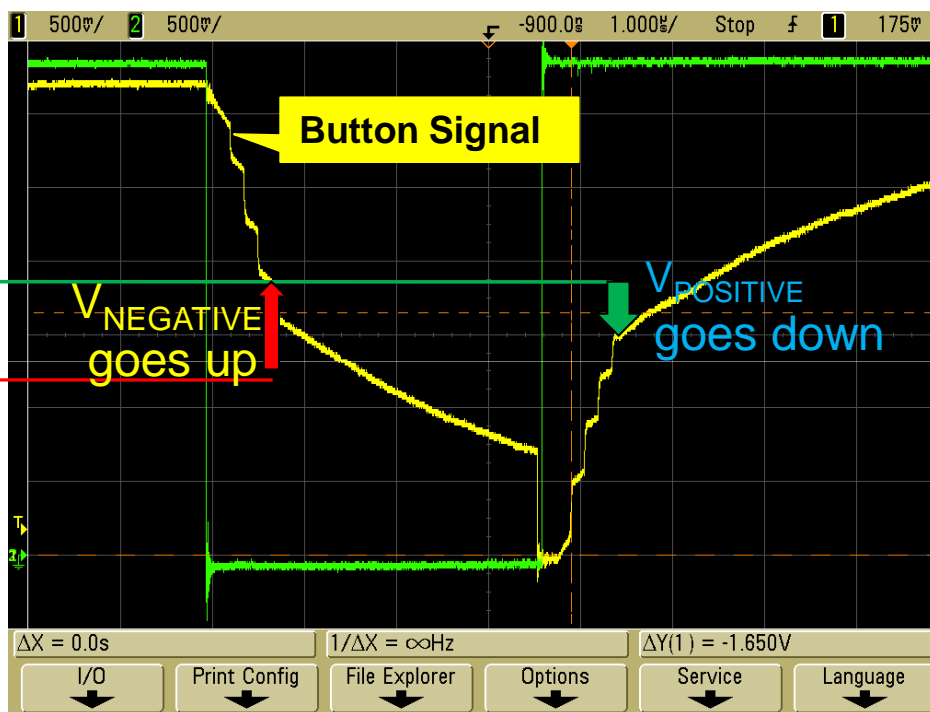


# Example Differential with Pull-Ups 'Scope Signals

## Unasserted



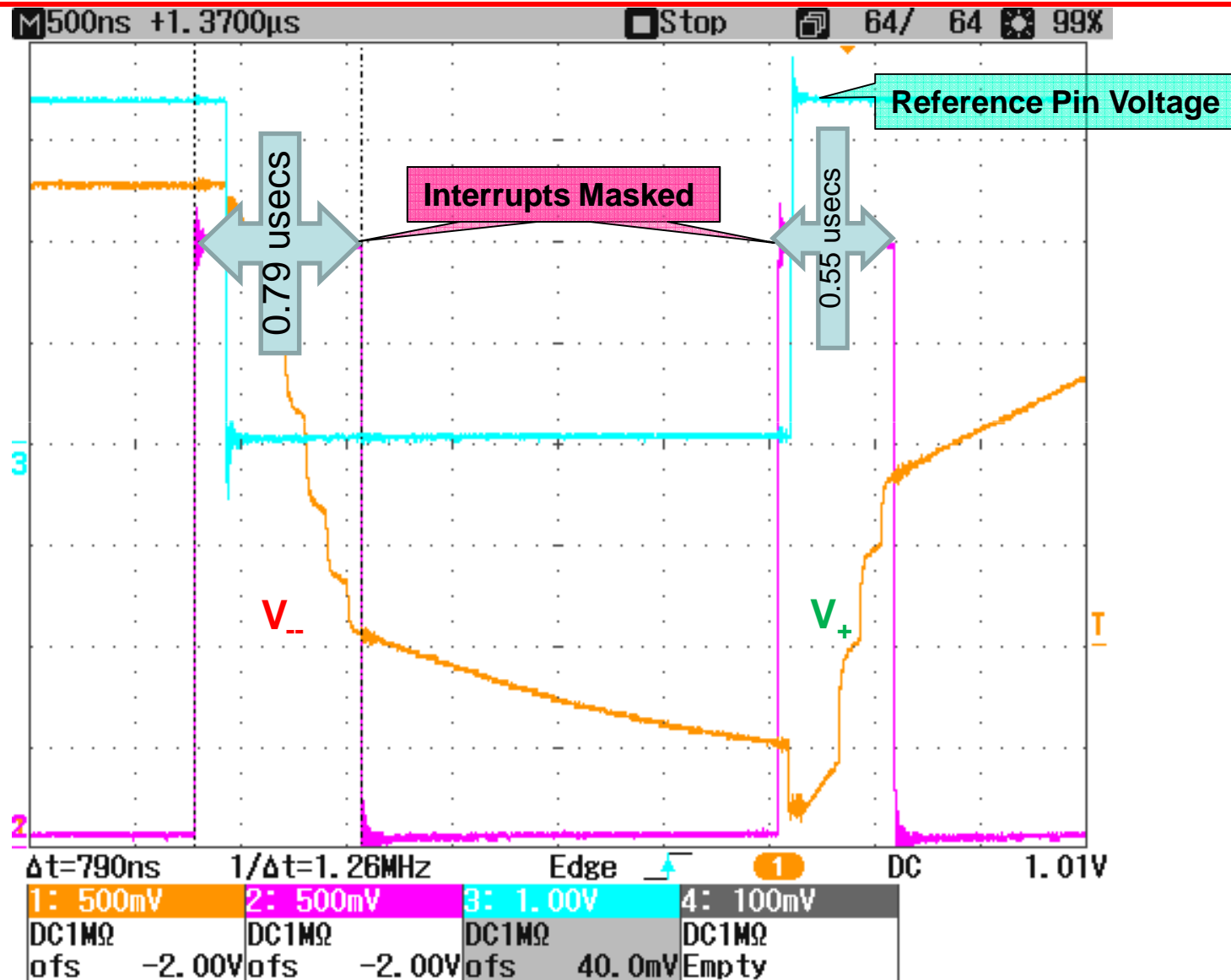
## Asserted

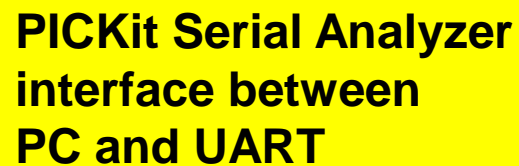






# Timing of Diff'l w. Pull-Ups



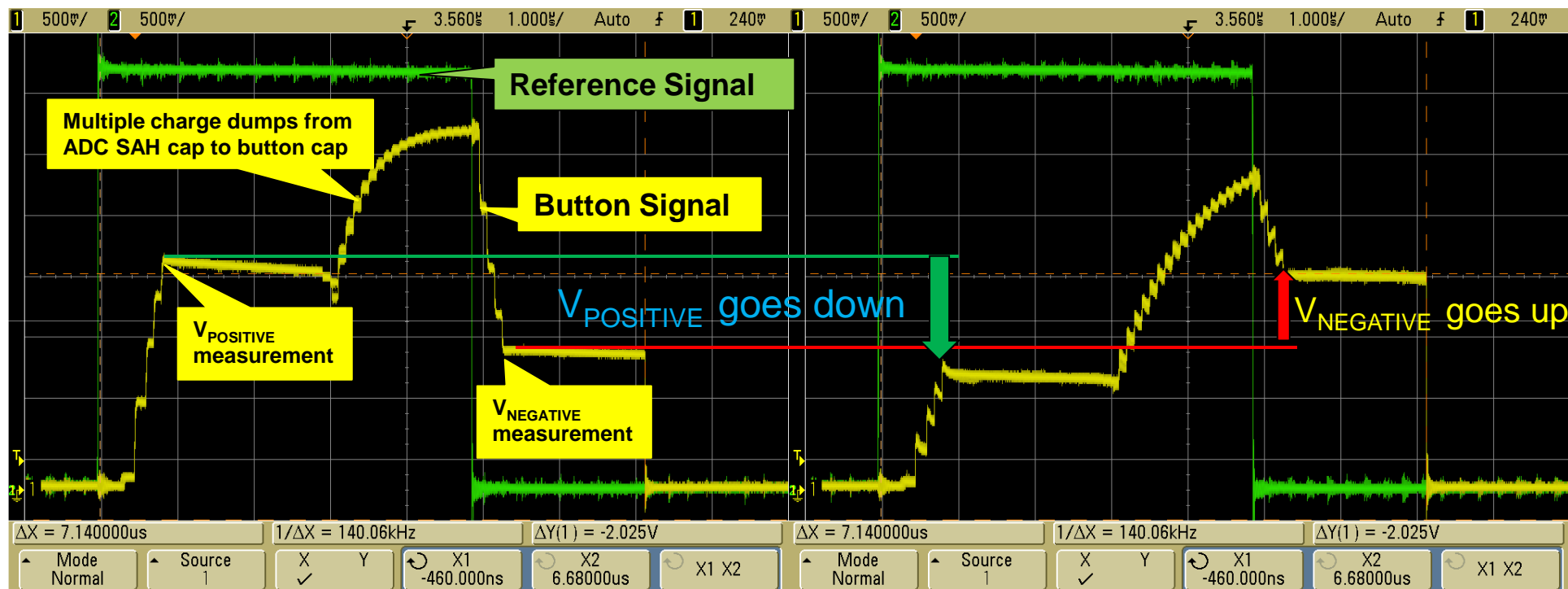




# Example Differential w/o Pull-Ups 'Scope Signals

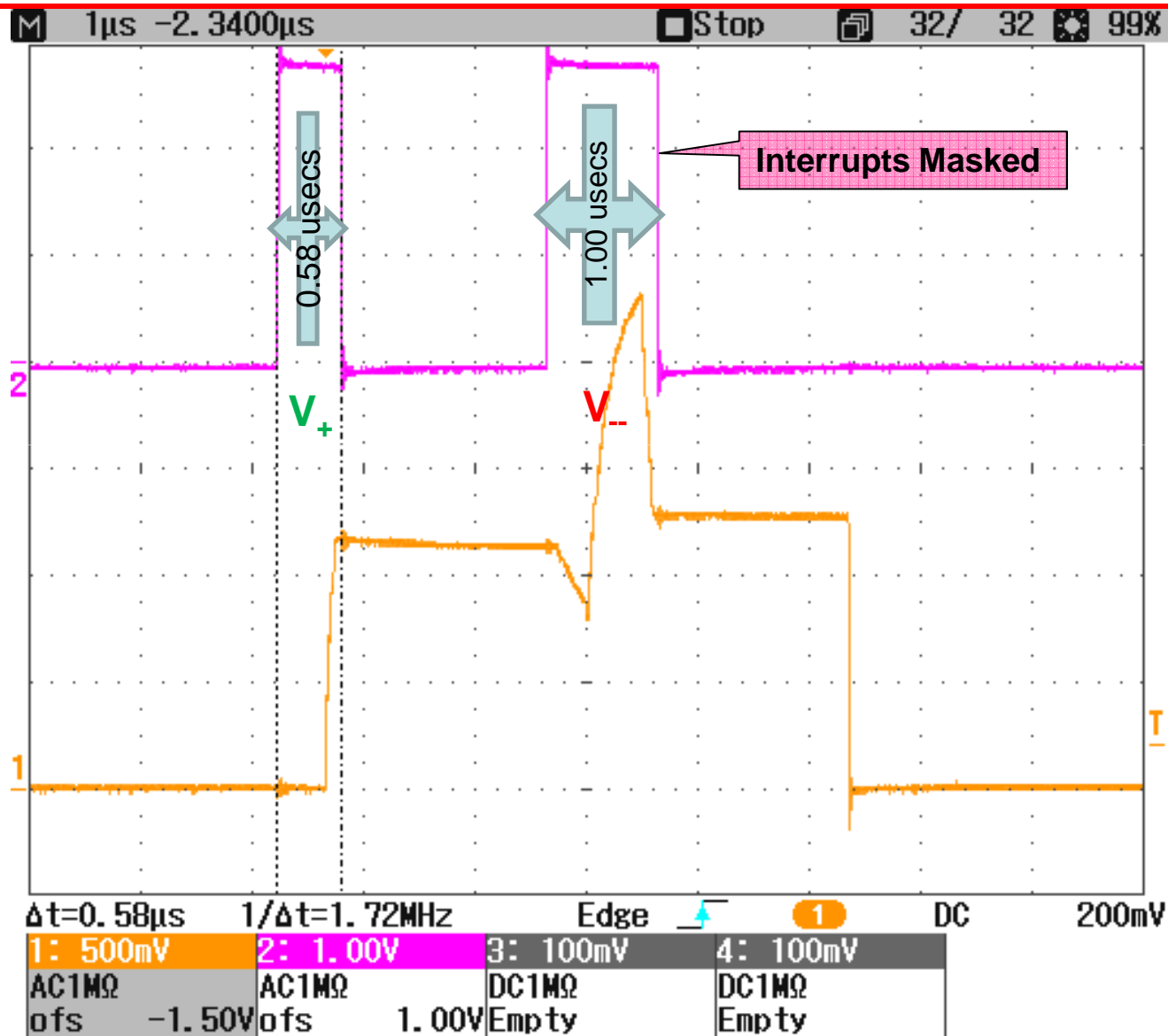
## Unasserted

## Asserted



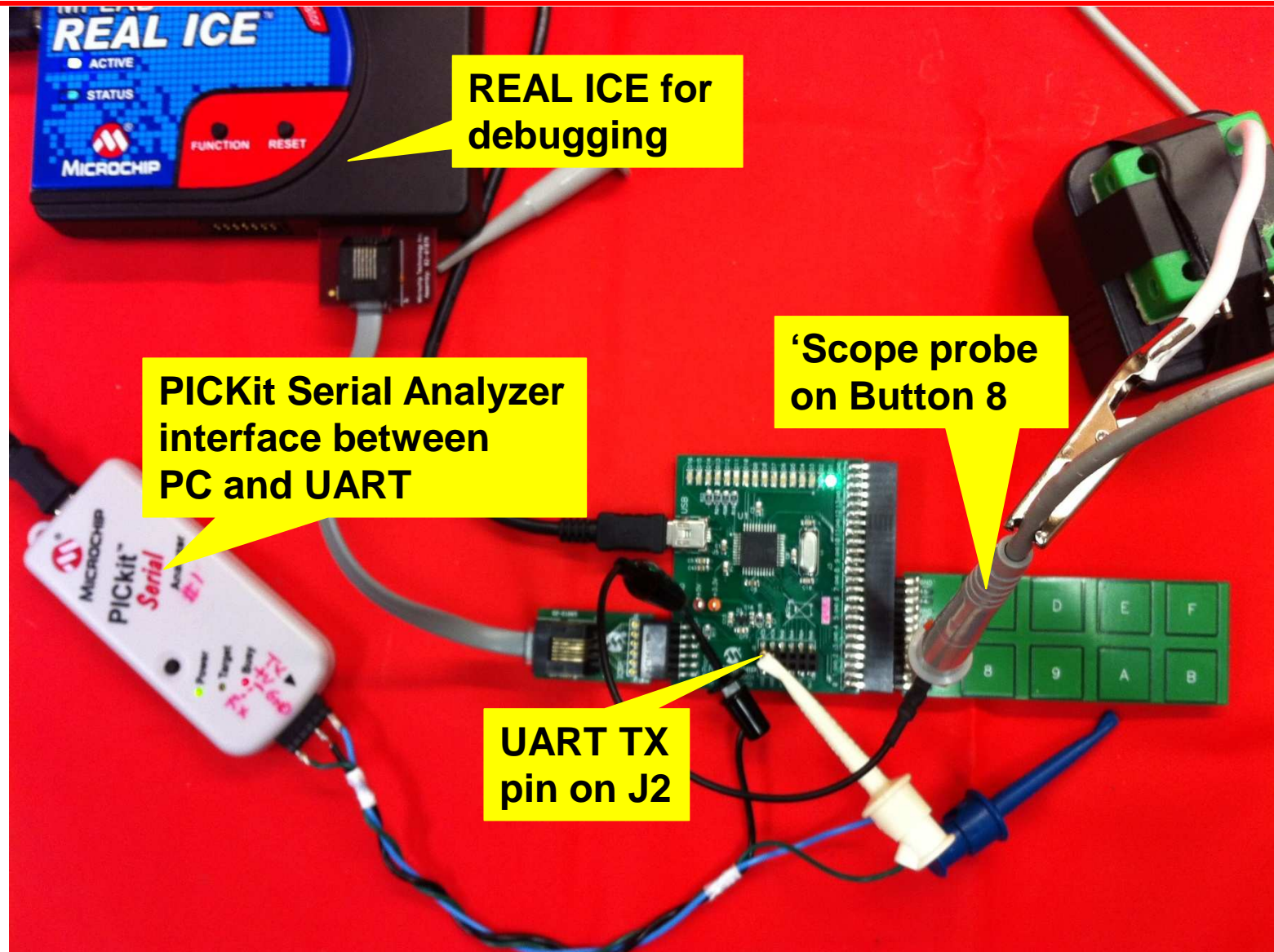


# Differential w/o Pull-Ups Timings





# Benchtop Setup for CTMU Measurements

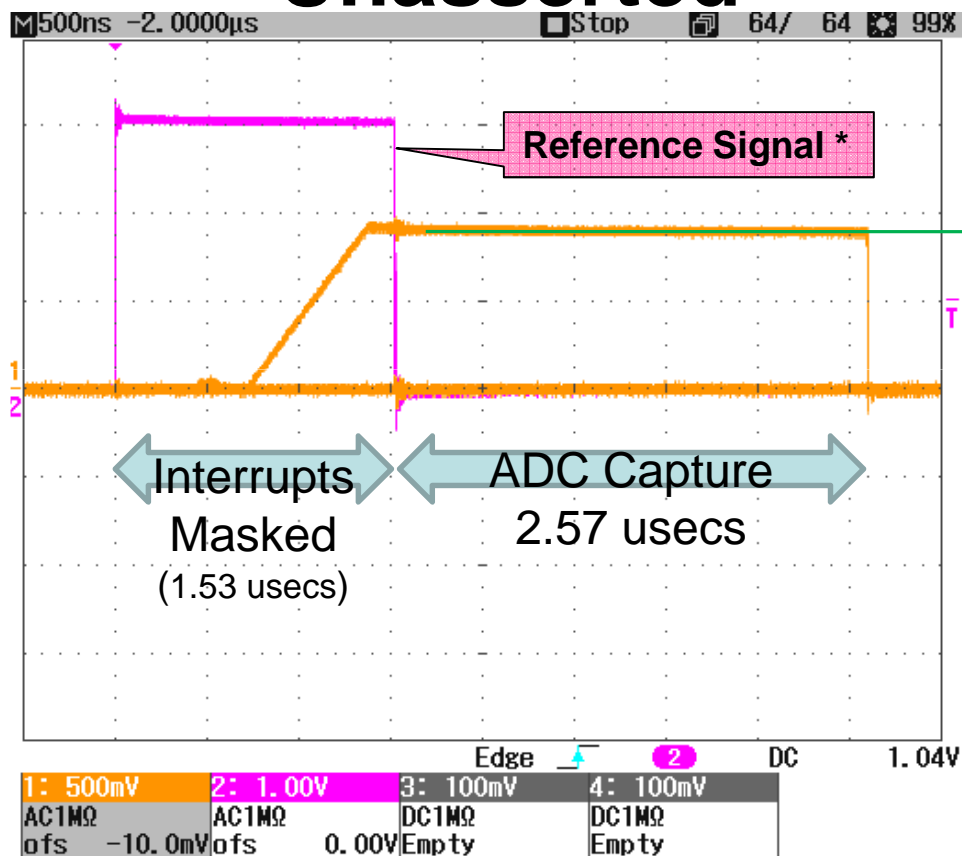




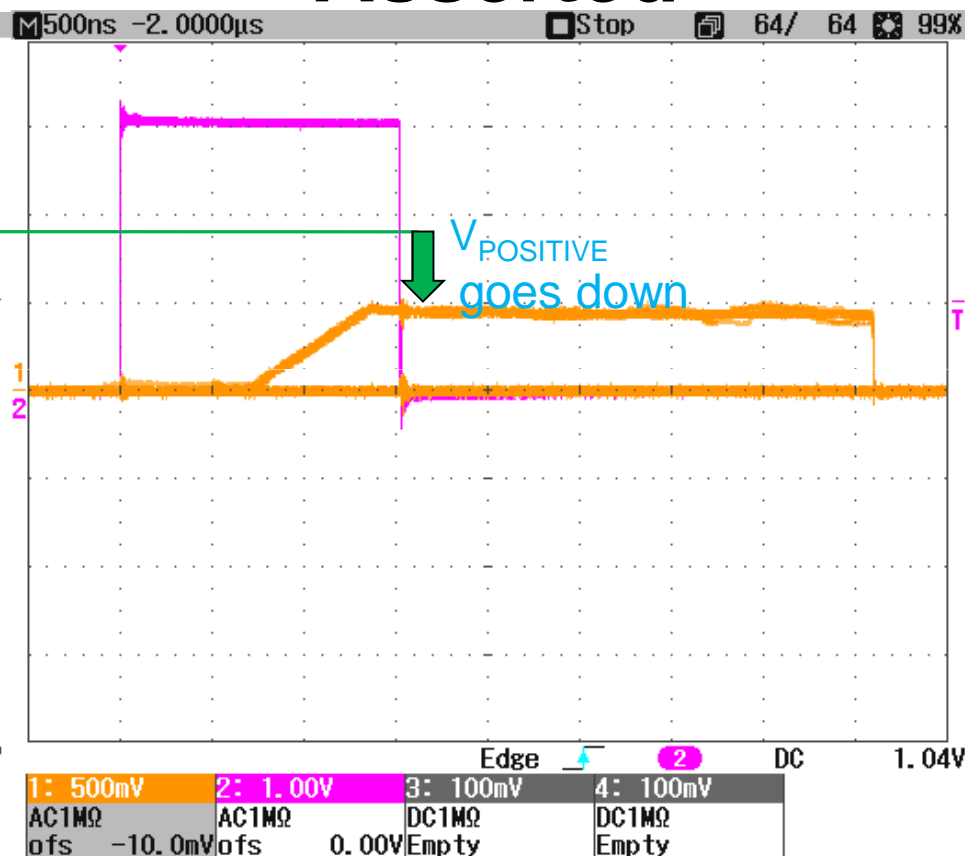


# Example CTMU Measurements

## Unasserted



## Asserted



\* Reference Signal: Interrupt masking, from asm("di") to asm("ei")



## Perf. Metric: Interrupt Masking Duration

- All these examples were taken using the Direct Key daughter card, but with different CVD/CTMU Evaluation Boards
- Each application tuned to provide equivalent cap touch performance

Interrupt Masking Durations [usecs]			
Technique	Vpos	Vneg	Vtotal
Diff w/o Pull-Ups	0.58	1	<b>1.58</b>
Diff w Pull-ups	0.55	0.79	<b>1.34</b>
CTMU	1.53	n/a	<b>1.53</b>



# Interrupt Masking Duration Workarounds

- **What if interrupts won't accommodate these black out periods?**
  - Don't use `asm("di")` and `asm("ei")` to mask interrupts, set `CapTouchMeas` flag instead of `asm("di")`, clear flag instead of `asm("ei")`
  - ISRs set `DumpMeasurement` flag if ISR executes when `CapTouchMeas` is set
  - Timer ISR dumps/ignores cap touch measurement when `DumpMeasurement` is set, then clears the flag

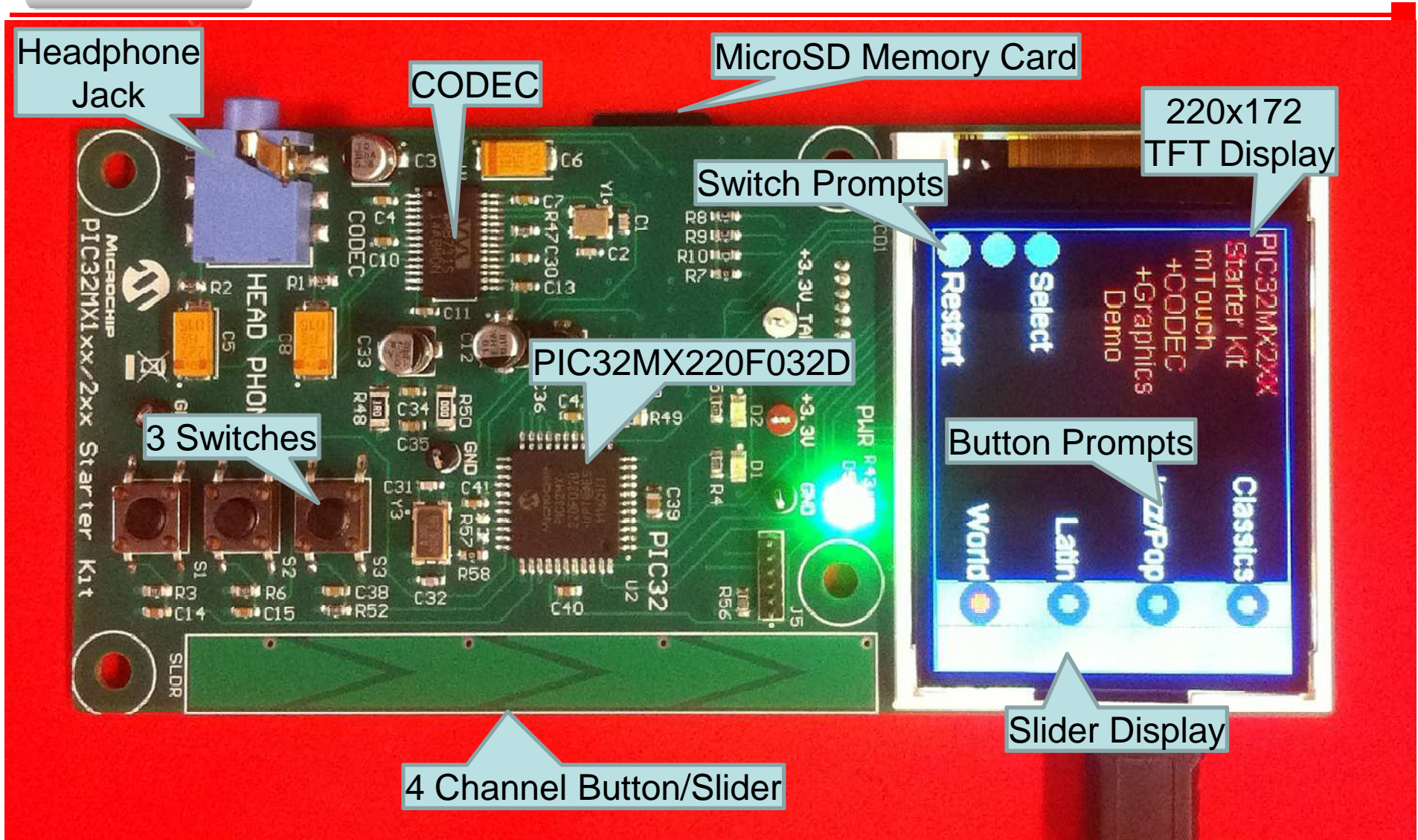


# PIC32MX2xx Starter Kit App

- **CTMU Cap Touch on 4-channel slider**
  - Selects music
  - Selects music volume
- **Three switches**
- **I2S I/F to CODEC**
- **PMP I/F to 220x172 pixel TFT display**
- **SPI I/F to microSD music card**
- **Playback of 44.1 Ksps .WAV files**



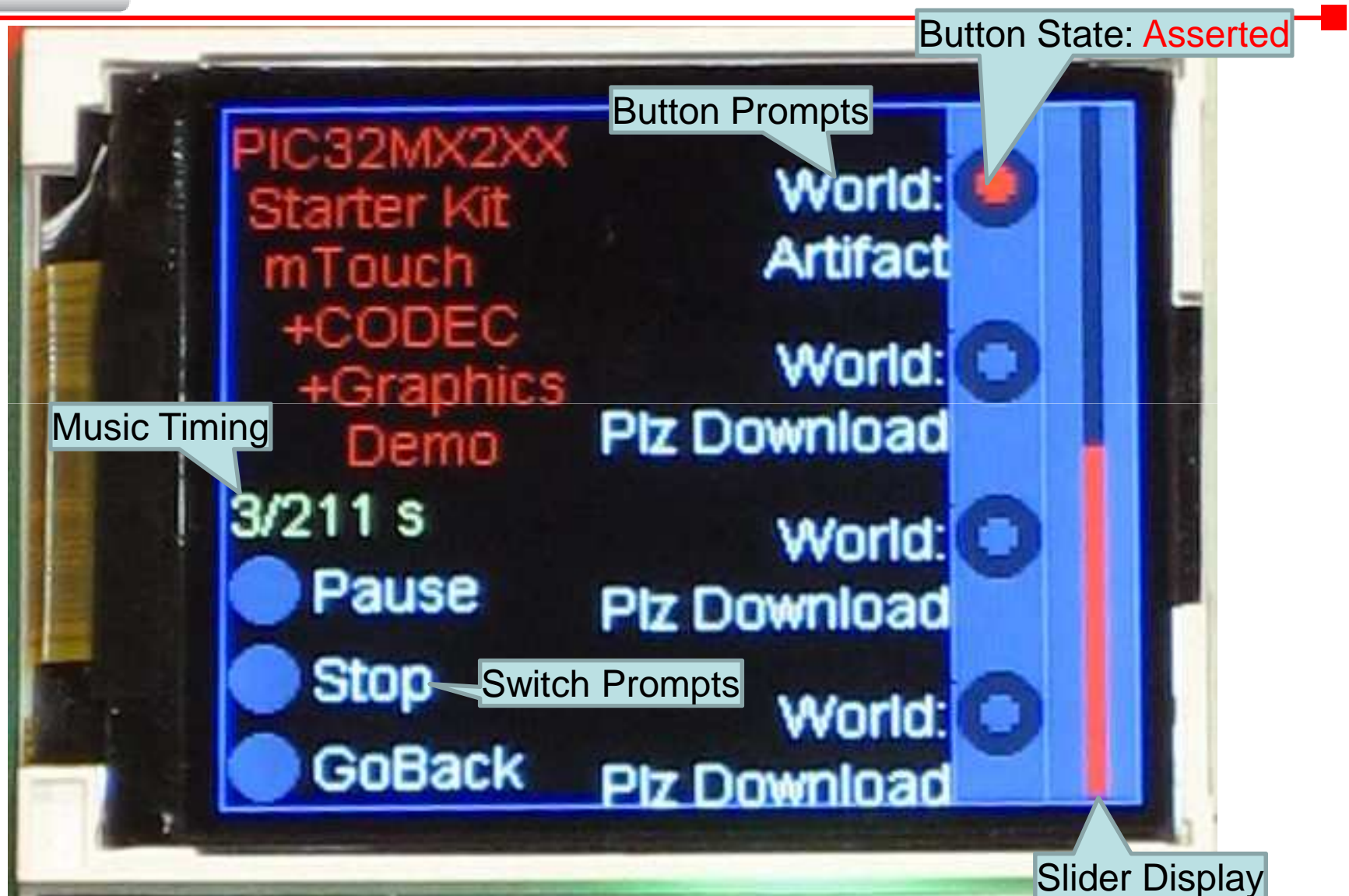
# PIC32MX2xx Starter Kit Board





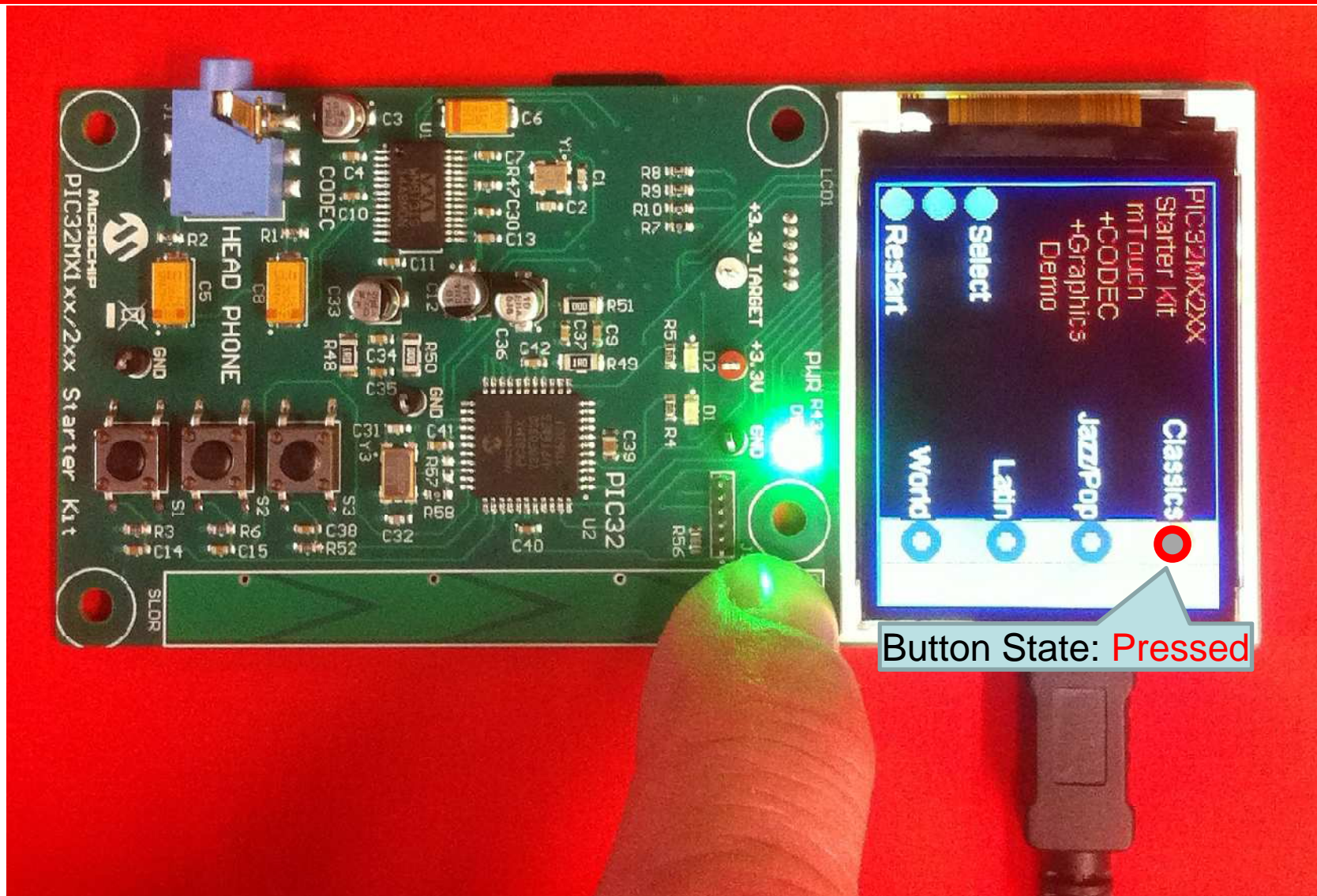


# PIC32MX2xx Starter Kit Display





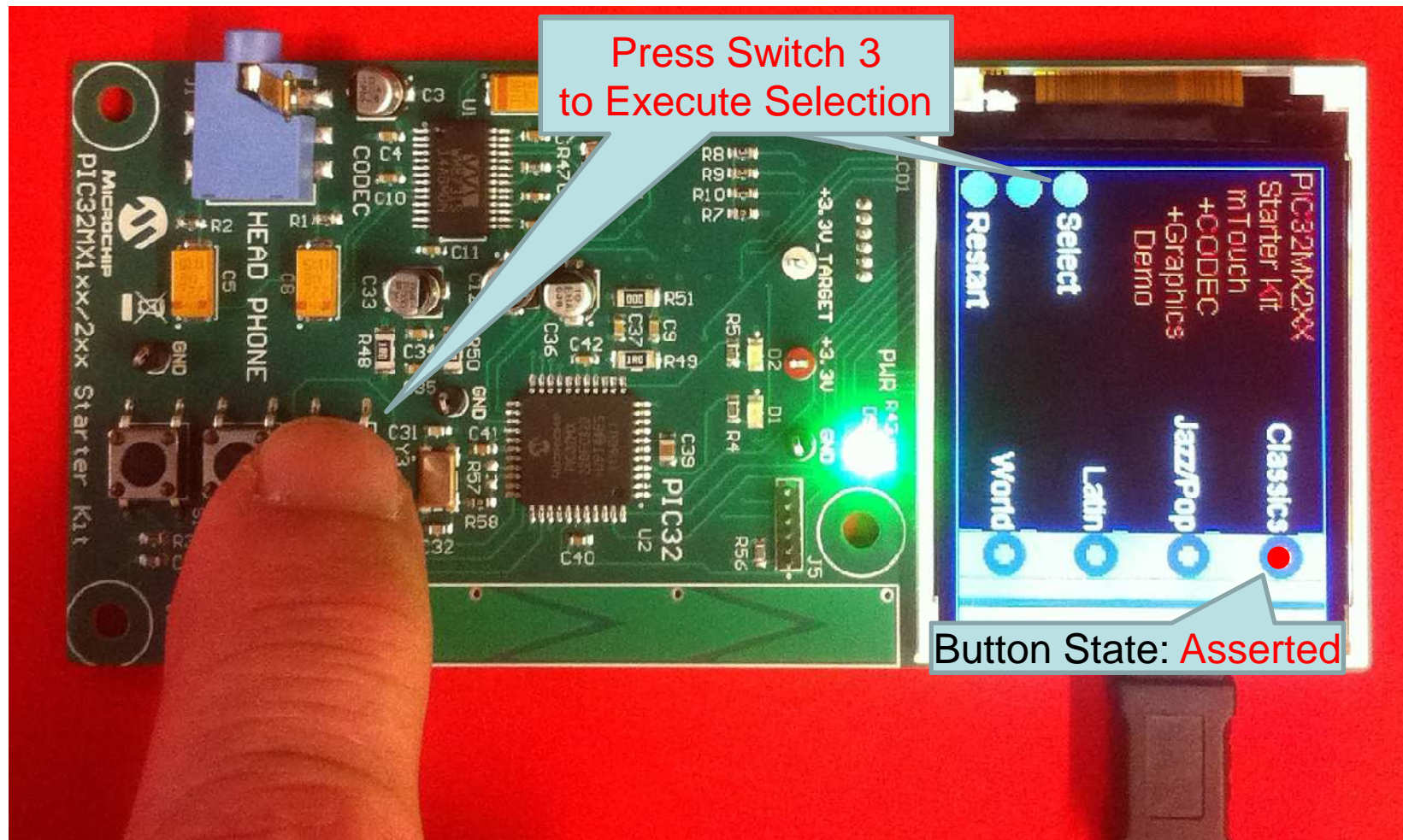
# Selecting Music 1/2







# Selecting Music 2/2





# Cap Touch Tricks 1/2

- Leave interpretation of button/slider status to the main application, only it knows context of the measurements!

```
while(1)
{
    // Update switches
    Switch1St8 = UpdateSwitch( 1, CheckSwitch1(), (char *)0 );
    Switch2St8 = UpdateSwitch( 2, CheckSwitch2(), (char *)0 );
    Switch3St8 = UpdateSwitch( 3, CheckSwitch3(), (char *)0 );

    // Update buttons/slider
    if( ApplicationState != STARTUP &&
        mTouchCapStatus_Check( &CurrentButtonStatus, &CurrentButtonAsserts, &Temp ) )
    {
        mTouchUpdated = TRUE;

        if ( IgnoreSlider != TRUE )
        {
            SliderValue = Temp;
            UpdateSlider( SliderValue );
        }
        if ( IgnoreButtons != TRUE )
        {
            UpdateButton( 1,(CurrentButtonStatus >> 0)&0x1,(CurrentButtonAsserts >> 0)&0x1,(char *)0);
            UpdateButton( 2,(CurrentButtonStatus >> 1)&0x1,(CurrentButtonAsserts >> 1)&0x1,(char *)0);
            UpdateButton( 3,(CurrentButtonStatus >> 2)&0x1,(CurrentButtonAsserts >> 2)&0x1,(char *)0);
            UpdateButton( 4,(CurrentButtonStatus >> 3)&0x1,(CurrentButtonAsserts >> 3)&0x1,(char *)0);
        }
    }
} //end if( mTouchCheckStatus...

switch( ApplicationState )
{
```



# Cap Touch Tricks <sub>2/2</sub>

- Only update screen after completion of button scan to prevent crosstalk between PMP screen pins and button inputs

```
case FILE_PLAYING_SETUP:
    if ( mTouchUpdated == TRUE ) // Only update display when mTouch
    {                               // has been updated and isn't running.
        UpdateSwitch(3,-1, "Pause" );
        UpdateSwitch(2,-1, "Stop" );
        UpdateSwitch(1,-1, "GoBack");

        IgnoreButtons = TRUE;
        IgnoreSlider   = FALSE;
        mTouchCapStatus_Reset(128);
        UpdateSlider(128);
        ApplicationState = FILE_PLAYING;
    }
    else // mTouch is still running don't do anything
    {
        ApplicationState = FILE_PLAYING_SETUP;
    }
    break;

case FILE_PLAYING:
    PMADDRSET = 1<<10;    // Turn on LED
    WM8731Codec_MusicOn(TRUE); // Start playing
```