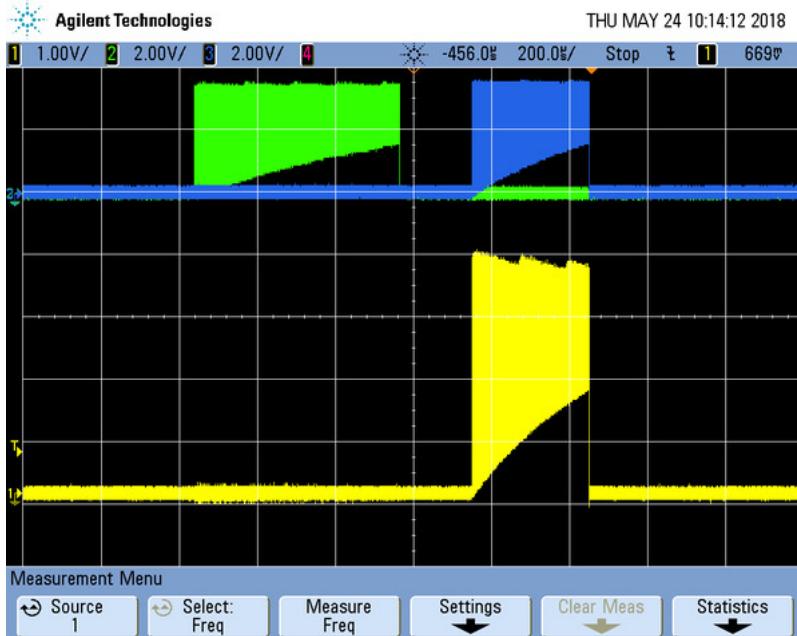


Hello,

I'm working on a project for a customer that uses a STM32F051C8T6. We're running two redundant capacitive sensors on the same I/O group and using V1.4.4 of the Touch Sense Library in our application.

I inherited the setup from another engineer, and since it was already working well enough for development haven't had to modify the sensor logic yet. We're gearing up for production and have final housings so I decided to properly tune the thresholds and make sure everything looked ok with the TSC. When I put all of the signals on a scope I noticed something strange, perhaps it's nothing, but I can't explain it so I wanted a second opinion. Here is the scope capture:



Channel 1 (Yellow) - Sampling Cap (C_s)

Channel 2 (Green) - TKEY Sensor 0 Electrode (C_{x0})

Channel 3 (Blue) - TKEY Sensor 1 Electrode (C_{x1})

Everything is working pretty well in our system at the application level, but I don't understand why I'm not seeing the C_s voltage charge up when Channel 0 is acquired. Is this normal? Where is the charge from C_{x0} going? If it's not being connected to C_s when it's discharging I would expect a shorter cycle than C_{x1} , not the longer one seen here. Something seems off to me...

Both channels are part of the same IO group, I've double-checked all of the IO parameters and everything is being configured correctly. Each channel is placed into its own "bank" within the TSL. Everything seems to work ok, when I don't have the scope probes connected (they add 10-15 pF of parasitic capacitance when connected) my reference value on channel 0 reads ~1300, and my reference value on channel 1 is ~1150. Initially I was just chalking this up to mechanical / electrical tolerance stack up, but now I'm not so sure. The electrodes are both the same size, and have very similar routing on the PCB, they are identical, redundant sensors. Our sampling capacitor is 10 nF.

Here's some relevant configuration snippets:

tsl_user.c:

```
//=====
// Channels
//=====
// Source and Configuration (ROM)
CONST TSL_ChannelSrc_T MyChannels_Src[TSLPRM_TOTAL_CHANNELS] =
{
    // Bank 0
    { CHANNEL_0_SRC, CHANNEL_0_IO_MSK, CHANNEL_0_GRP_MSK },
    // Bank 1
    { CHANNEL_1_SRC, CHANNEL_1_IO_MSK, CHANNEL_1_GRP_MSK }
};

// Destination (ROM)
CONST TSL_ChannelDest_T MyChannels_Dest[TSLPRM_TOTAL_CHANNELS] =
{
    // Bank 0
    { CHANNEL_0_DEST },
    // Bank 1
    { CHANNEL_1_DEST }
};

// Data (RAM)
TSL_ChannelData_T MyChannels_Data[TSLPRM_TOTAL_CHANNELS];
//=====
// Banks
//=====
// List (ROM)
CONST TSL_Bank_T MyBanks[TSLPRM_TOTAL_BANKS] = {
    {&MyChannels_Src[0], &MyChannels_Dest[0], MyChannels_Data, BANK_0_NBCHANNELS, BANK_0_MSK_CHANNELS, BANK_0_MSK_GROUPS},
    {&MyChannels_Src[1], &MyChannels_Dest[1], MyChannels_Data, BANK_1_NBCHANNELS, BANK_1_MSK_CHANNELS, BANK_1_MSK_GROUPS}
};

//=====
// TouchKeys
//=====
// Data (RAM)
TSL_TouchKeyData_T MyTKeys_Data[TSLPRM_TOTAL_TKEYS];
// Parameters (RAM)
TSL_TouchKeyParam_T MyTKeys_Param[TSLPRM_TOTAL_TKEYS];
```

tsl user.h:

tsl_conf_stm32f0xx.h:

```

=====
// Number of elements
=====

/** @defgroup Common_Parameters_Number_Of_Elements 01 - Number of elements
 * @{
 */

/** Total number of channels in application (range=1..255)
 */
#define TSLPRM_TOTAL_CHANNELS (2)

/** Total number of banks in application (range=1..255)
 */
#define TSLPRM_TOTAL_BANKS (2)

/** Total number of "Extended" TouchKeys in application (range=0..255)
 */
#define TSLPRM_TOTAL_TOUCHKEYS (2)

/** Total number of "Basic" TouchKeys in application (range=0..255)
 */
#define TSLPRM_TOTAL_TOUCHKEYS_B (0)

/** Total number of "Extended" Linear and Rotary sensors in application (range=0..255)
 - Count also the 1-channel linear sensor used as TouchKey
 */
#define TSLPRM_TOTAL_LINROTS (0)

/** Total number of "Basic" Linear and Rotary sensors in application (range=0..255)
 - Count also the 1-channel linear sensor used as TouchKey
 */
#define TSLPRM_TOTAL_LINROTS_B (0)

/** Total number of sensors/objects in application (range=1..255)
 - Count all TouchKeys, Linear and Rotary sensors
 */
#define TSLPRM_TOTAL_OBJECTS (2)

/** @} Common_Parameters_Number_Of_Elements */

=====
// GPIO configuration
=====

/** @defgroup STM32F0xx_Parameters_GPIO_Config 01 - TSC GPIOs Configuration
 * @{
 */

/** TSC GPIOs Configuration selection (range=0..1)
 */

```

```

- 0: Manual. The TSC GPIOs configuration must be done by the application code.
- 1: Automatic. The TSLPRM_TSC_GROUPx_IOy parameters below must be filled up.
    The TSC GPIOs configuration is automatically done by the STMTouch driver.

/*
#define TSLPRM_TSC_GPIO_CONFIG (1)

//++ DO NOT CHANGE THESE VALUES ++++++
// These defines must be applied to the TSLPRM_TSC_GROUPx_IOy parameters below.
#define NU      (0) // Not Used IO
#define CHANNEL (1) // Channel IO
#define SHIELD (2) // Shield IO (= Channel IO but not acquired)
#define SAMPCAP (3) // Sampling Capacitor IO
//++++

// If TSLPRM_TSC_GPIO_CONFIG=1 assign each TSLPRM_TSC_GROUPx_IOy parameters below.
// If TSLPRM_TSC_GPIO_CONFIG=0 these parameters are ignored.

#define TSLPRM_TSC_GROUP1_IO1  NU      // PA0
#define TSLPRM_TSC_GROUP1_IO2  NU      // PA1
#define TSLPRM_TSC_GROUP1_IO3  NU      // PA2 - K3
#define TSLPRM_TSC_GROUP1_IO4  NU      // PA3

#define TSLPRM_TSC_GROUP2_IO1  NU      // PA4
#define TSLPRM_TSC_GROUP2_IO2  NU      // PA5
#define TSLPRM_TSC_GROUP2_IO3  NU      // PA6 - K9
#define TSLPRM_TSC_GROUP2_IO4  NU      // PA7

#define TSLPRM_TSC_GROUP3_IO1  NU      // PC5
#define TSLPRM_TSC_GROUP3_IO2  CHANNEL // PB0
#define TSLPRM_TSC_GROUP3_IO3  CHANNEL // PB1
#define TSLPRM_TSC_GROUP3_IO4  SAMPCAP // PB2

#define TSLPRM_TSC_GROUP4_IO1  NU      // PA9
#define TSLPRM_TSC_GROUP4_IO2  NU      // PA10
#define TSLPRM_TSC_GROUP4_IO3  NU      // PA11
#define TSLPRM_TSC_GROUP4_IO4  NU      // PA12

#define TSLPRM_TSC_GROUP5_IO1  NU      // PB3
#define TSLPRM_TSC_GROUP5_IO2  NU      // PB4
#define TSLPRM_TSC_GROUP5_IO3  NU      // PB6
#define TSLPRM_TSC_GROUP5_IO4  NU      // PB7

#define TSLPRM_TSC_GROUP6_IO1  NU      // PB11
#define TSLPRM_TSC_GROUP6_IO2  NU      // PB12
#define TSLPRM_TSC_GROUP6_IO3  NU      // PB13
#define TSLPRM_TSC_GROUP6_IO4  NU      // PB14

// Warning: this group is available on some devices only.
#define TSLPRM_TSC_GROUP7_IO1  NU      // PE2
#define TSLPRM_TSC_GROUP7_IO2  NU      // PE3
#define TSLPRM_TSC_GROUP7_IO3  NU      // PE4
#define TSLPRM_TSC_GROUP7_IO4  NU      // PE5

// Warning: this group is available on some devices only.
#define TSLPRM_TSC_GROUP8_IO1  NU      // PD12
#define TSLPRM_TSC_GROUP8_IO2  NU      // PD13
#define TSLPRM_TSC_GROUP8_IO3  NU      // PD14
#define TSLPRM_TSC_GROUP8_IO4  NU      // PD15

/** @} STM32F0xx Parameters GPIO_Config */

```

Am I over analyzing this and it's just something with the hardware preventing me from seeing Cs charge up on the first acquisition? Or is there actually an issue here?

Thanks

Josh