

PowerPSoC® – Measuring Channel Current Using the Analog Input Multiplexer (AINX)

Associated Part Family: CY8CLED0xx0x

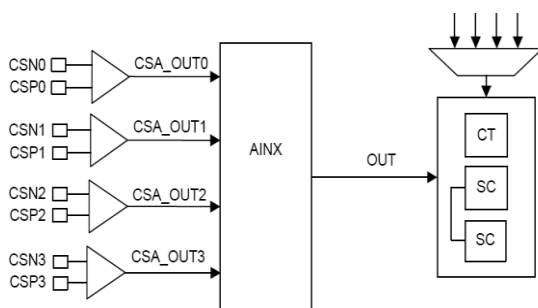
Software: PSoC® Designer™ 5.2

Related Hardware: CY3268

Overview

In this example, we drive two power channels of PowerPSoC, each with a different signal density. The CSA outputs of these channels are routed through the AINX to the analog blocks in the PSoC core.

Figure 1. AINX MUX Connections



The analog block is configured as an ADC and multiplexed between the two channels. The ADC output is converted back to the equivalent current. The measured value is also read using the USB-I²C Bridge. For this, an I²C slave user module is placed in the PowerPSoC controller to pass the equivalent current output data to the I²C master on the bridge. This is then plotted as a 'Current vs. time' graph on the USB-I²C Bridge GUI. This example can be tested on the CY3268 kit.

This example helps the user to get familiar with using the AINX block to monitor the current through the PowerPSoC's output channels. Also, a method to read the current value through the USB-I²C Bridge is provided.

A list of the required user modules and the parameter settings for these modules follows below.

User Module List and Placement

User Module	Placement
ADCINCVR	ASC10, DBB11, DCB03, DCB12
CSD0	DBB00, DBB01, DCB02, ACB01
CSA0	CSA0

CSA1	CSA1
EzI ² C	See note below
HYSTCTRL0	HYSTCTL0
HYSTCTRL1	HYSTCTL1
PRISM0	MOD0
PRISM1	MOD1

Note: EzI²Cs occupies a dedicated hardware block.

User Module Parameter Settings

ADCINCVR		
Parameter	Value	Comments
Input		The CSA output of a particular channel is routed to the ADC input in firmware
Clock Phase	Norm	
Clock	VC2	VC2 = VC1/16 = 187.5 KHz. The clock has to be between 125 kHz and 8 MHz
ADCResolution	8	
CalcTime	100	
DataFormat	Unsigned	

Note: For the project, the RefMux setting is Bandgap +/- Bandgap, so the ADC range is 0 to 2.6 V. Since the resolution is 8 bit, the step size is [2.6 / 255] V.

CSA0		
Parameter	Value	Comments
Bandwidth	Highest	

Note: The same parameters apply for CSA1.

HYSTCTRL0		
Parameter	Value	Comments
Gate Driver strength	Default	Default is the highest gate driver strength
RefHigh	113	See the following notes
RefLow	104	See the following notes
FeedbackInput	CSA0	--
DACVoltageRange	2.6 Step10 mV	The full scale DAC is selected
DimInput	MOD0	--
TriplInput	VGND	--
TimerDelay	10-25 ns	--
Gate Driver	Internal	--

Notes:

- The calculations for the RefHigh and RefLow are based on the peak current, valley current, DAC range, sense resistor and the gain of the CSA. On the CY3268, the Rsense is .22 Ω.
- $\text{RefHigh} = (I_{\text{peak}} * R_{\text{sense}} * \text{CSA gain}) / (\text{DAC_Resolution})$
- $\text{RefLow} = (I_{\text{valley}} * R_{\text{sense}} * \text{CSA gain}) / (\text{DAC_Resolution})$
- Channel 0 is driven with Peak and valley currents of 260 mA and 240 mA, for channel 1 it is 130 mA and 120 mA respectively.
- The parameters for HYSTCTRL1 are calculated similarly.
- For a better understanding of these calculations, refer the application note [PowerPSoC™ Firmware Design Guidelines - AN51012](#)

PRISM0		
Parameter	Value	Comments
ClockScaler	255	SysClk / 255 are the input clock source.
Dimming Resolution	8	--
SignalDensity	0	Is set appropriately in the firmware
Frequency compensation	Disable	--
Compare Type	Less than	--

Note: The same parameters apply for PRISM1.

EzI ² C		
Parameter	Value	Comments
Slave_Addr	0	--
Address_Type	Static	--
ROM_Registers	Disable	--
I ² C Clock	400k	--
I ² C Pin	P1 [0]-P1 [1]	See the following note

Note: On the CY3268, for CY8CLED04D01, the I²C data and I²C clock pins are connected to P1 [0] and P1 [1] respectively.

CSD0		
Parameter	Value	Comments
Finger Threshold	200	See the following note
Noise Threshold	20	--
BaselineUpdate Threshold	200	--
Sensors Autoreset	Enabled	--
Hysteresis	10	--
Debounce	3	--
NegativeNoiseThreshold	20	--
LowBaselineReset	50	--
Scanning Speed	Normal	--
Resolution	12	--
Ref Value	2	--
ShieldElectrodeOut	NA	--

Notes:

- These parameters are calculated based on the tuning of the Capsense buttons. For details on tuning, refer the section 'CSD Step-By-Step Tuning Guide' in the [CSD datasheet](#).
- On the CY3268, the first two CapSense buttons are connected to P1 [4] and P2 [2], the modulator capacitor pin is on P0 [5] and feedback resistor pin is on P1 [5]. The same have been configured in the CSD wizard.

Global Resources

Set RefMux to 'Bandgap +/- Bandgap'

Set VC1 to 8 and VC2 to 16.

Hardware Connections

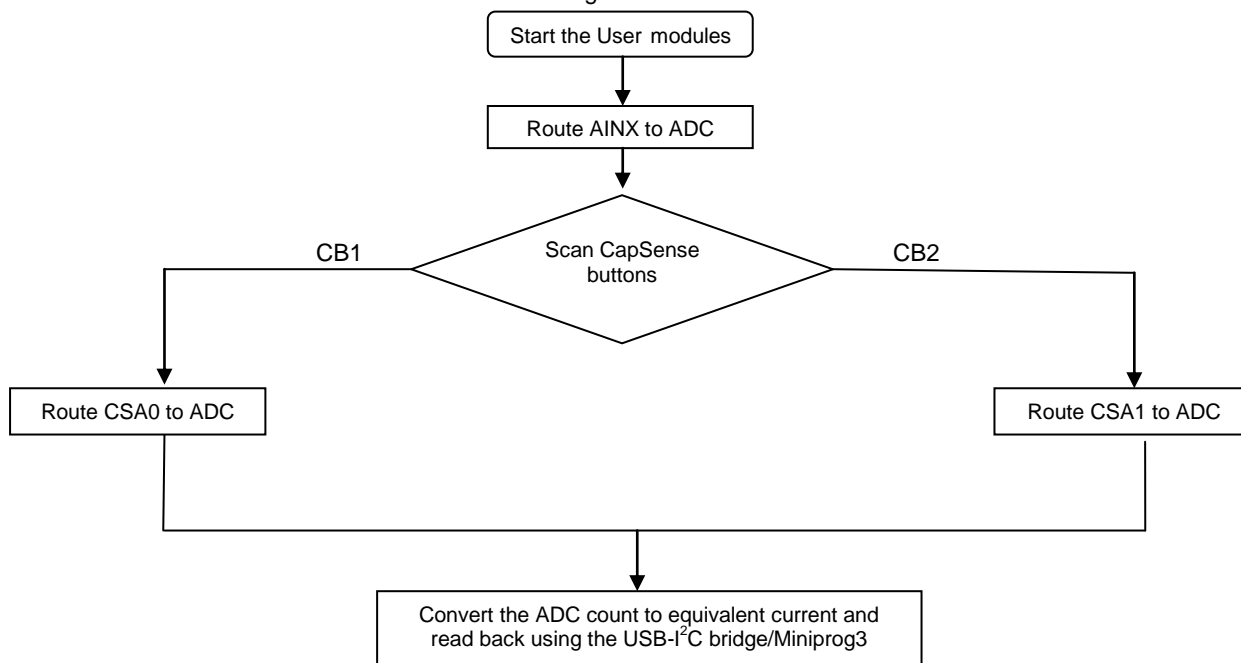
Refer the schematics of the [CY3268](#) board.

Operation

- The example is setup to drive channel 0 and channel 1 with maximum signal density of 255.
- The RefHigh and RefLow limits for channel 0 are 113 and 104; the limits for channel 1 are 56 and 51 respectively. So, the average current in channel 1 should be half of the current in channel 0.
- Internal CSA is used for both the channels. The output of the internal CSA is routed to an ADC.

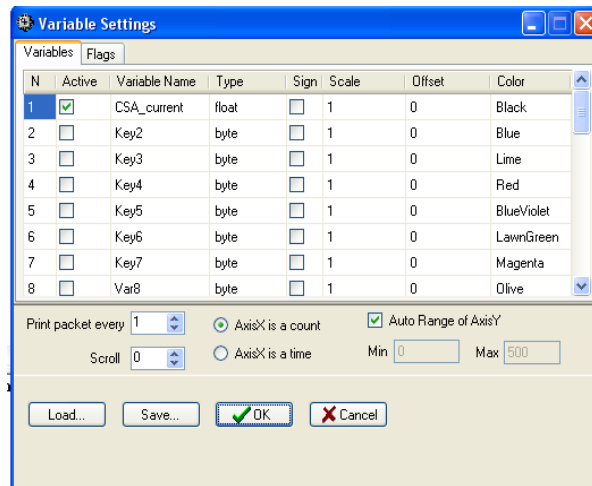
4. The routing of the AINX to the analog block (switched capacitor type) is controlled by the register ASC10CR1. Further, bits 4 and 5 of PAMUX_S4 register determine which of the CSA output is routed through the AINX block. Refer the [PowerPSoC TRM](#) for a detailed description of these registers.
5. In the main routine, when CSB1 (Capsense Button 1) is pressed, the CSA output of channel0 is routed to the ADC through the AINX. Similarly, when CSB2 is pressed, the CSA output of channel1 is routed to the ADC through the AINX.
6. The appropriate CSA output is sampled by the ADC and the ADC count is converted to an equivalent current in mA.
7. This equivalent current value is exposed to the USB-I²C bridge master through the EzI²Cs slave and read back on the GUI.

Figure 2. Flowchart



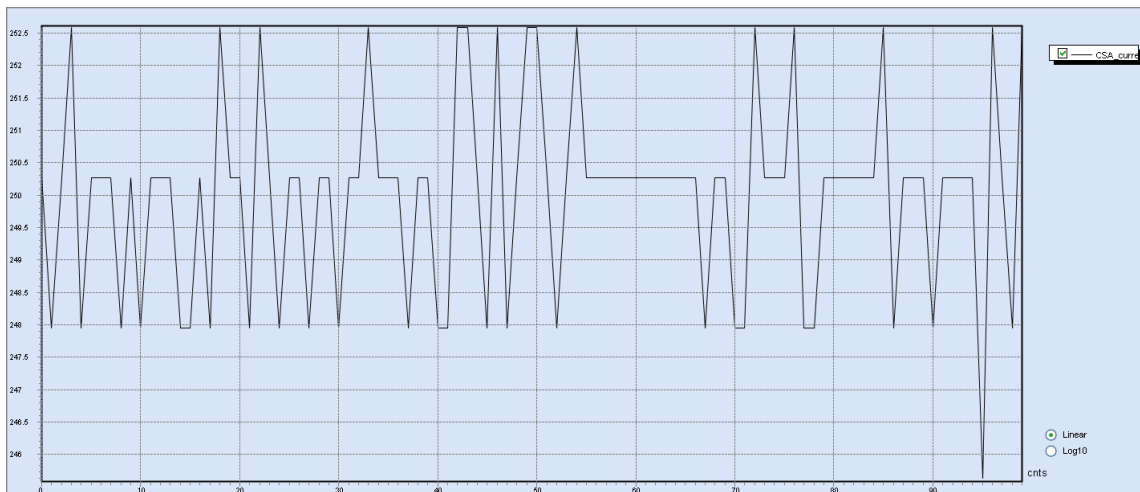
Testing the Project:

1. Connect the provided USB cable from the PC to the MiniProgrammer. Then, connect the MiniProgrammer to header J2 on the CY3268.
2. Open PSoC programmer. Set programming mode to PowerCycle and AutoDetection to On. PSoC programmer can be downloaded <http://www.cypress.com/?rID=38050r>.
3. In PSoC programmer, open the file 'AINX.hex', which is located in the folder \firmware. Press the program button to start programming. The status window should show "Programming Starting".
4. Once the program is downloaded, power the CY3268 using the provided 12 V DC supply.
5. On powering up the board, 2 channels [Red and Green] should turn ON.
6. Download and install the USB- I²C bridge control panel from the link [GUI](#).
7. Connect the USB-I²C Bridge [CY3240] between the PC and the CY3268 kit using the USB cable. For more information on using the bridge control panel, refer to its user guide. Note that Miniprogrammer 3 can also be used a USB - I²C bridge.
8. On the bridge control panel, enter CSA_current as a variable. Note that this variable name and type should match the variable name and type declared in firmware.



9. To read the values on the USB-I²C control panel, use the following format:
r 00 @3CSA_current @2CSA_current @1CSA_current @0CSA_current p
10. Set the 'Repeat Count' to 100 and the Scan period to '1' on the Chart window. Figure 3 is a sample waveform when Channel 0 was selected through CSB1; the y-axis is the current in milli Amps.

Figure 3. Channel 0 Current



Note: The ideal current waveform is a triangular wave, with peak at 260 mA and valley at 240 mA. Here, the distortion is due to measurement offsets, among others.

11. Similarly, channel 1 current can be read by using CapSense button 2 and repeating steps 9 and 10.

Figure 4. Channel 1 Current

