



Application Note

AN2363

Sensor Calibration with PSoC Express™

Author: Dave Funston

Associated Project: Yes

Associated Part Family: CY8C21xxx, CY8C24xxxA, CY8C24794, CY8C27xxx, CY8C29xxx

PSoC Express Version: 2.0

Associated Application Notes: AN2339

Abstract

PSoC Express design elements and custom code can be combined to create a two-point, straight-line calibration curve for a sensor.

Introduction

Use of an analog sensor within a microcontroller application typically requires that some form of calibration be done to compensate for systematic errors in the sensor and in the analog-to-digital conversion. In many cases, a simple straight-line calibration using two known measurand values will suffice. This Application Note shows how PSoC Express design elements combined with custom code can perform a straight-line calibration.

Application Description

The project associated with this note contains an example application that acquires a voltage reading (X) and converts it into a value (Y) that represents the measurand of interest, which could be any physical phenomenon.

The application also includes a mechanism for acquiring calibration data and storing it in RAM. Typically, calibration data is also stored in non-volatile memory (such as Flash memory), but this activity is not within the scope of this application note. Refer to AN2339 “Storing Calibration Factors into Flash Memory Within a PSoC Express Application” for storing data into Flash memory.

Calibration Data Acquisition

The PSoC Express graphical design oversees the entire data point acquisition process to enable this conversion. The application stores two predefined values of Y in interface valuator, which can be modified via an I2C interface. The two Y values are known values at which X will be acquired to establish the relationship between X and Y. The X values are acquired by pushing the buttons GetX1 and GetX2. The application stores these values upon initiation of the button presses (not while the buttons are held down), and the Loop Delay valuator, PrevGetX1 and PrevGetX2, enable the application to capture the button press initiation.

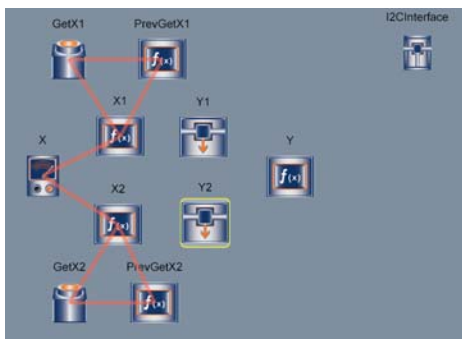


Figure 1. Example Application

There are many alternatives to using button presses to trigger calibration data acquisition. Such examples include:

- Using a state machine. A state transition can trigger the acquisition of a calibration data point. As an example, the state machine can use a single button as input to trigger transitions through a calibration sequence. State machines can also direct various operation modes, so calibration can be a mode separate from normal operation.
- Observing another analog input value for a transition above or below a given threshold.
- Observing an edge transition (low to high or high to low) on a digital input.
- Observing a value change in an interface valuator initiated by communication from an I2C master.

X-to-Y Conversion

The conversion of X to Y is performed in custom code written into the `CustomPostInputUpdate()` function of *custom.c* (which is a source code file generated during the first project build; refer to AN2339 for more information on *custom.c*). The resulting Y value is then available to transfer function valuator and output drivers in the current loop iteration.

The main reason for writing custom code to perform this conversion is to preserve the resolution of Y. When the 16-bit X is entered into a straight-line equation, multiplication of X by a 16-bit slope gives a 32-bit result. Since a 32-bit numeric format does not exist in any PSoC Express graphical design element, the application designer would have to scale down X or the slope in order to avoid overflow while using graphical design elements. The result of this effort is loss of resolution and accuracy in the resulting Y value. The logical alternative is to perform the conversion in 'C' code, which has the 32-bit `long` data type available.

The `CustomPostInputUpdate()` function proceeds as follows:

- Determines if the user is pressing either of the buttons to obtain an X value corresponding to one of the two predefined Y values. If this condition occurs, the Y is set equal to the corresponding predefined Y value. In the case that the user-initiated button pushes on both buttons simultaneously, Y1 is output.
- Places the values of the calibration data, X1, Y1, X2, and Y2, into long variables in preparation for line calculation.

If X1 and X2 are equal, then the line is vertical. Output the maximum number that a 16-bit integer can hold. Otherwise, calculate the output Y value using the point-slope form of a straight-line equation. Convert the 32-bit result into a 16-bit integer.

Conclusion

Often, two-point calibration is a useful part of a microcontroller application. 'C' customization of PSoC Express applications makes calibration possible without losing resolution or accuracy.

About the Author

Name: Dave Funston

Title: Applications Engineer, Staff

Background: Dave Funston supports PSoC Express at Cypress Semiconductor in Lynnwood, WA. He holds a Master's degree in Mechanical Engineering from the University of California at Davis, and has programmed microcontrollers since 1996.

Contact: dfu@cypress.com

Cypress Semiconductor
2700 162nd Street SW, Building D
Lynnwood, WA 98087
Phone: 800.669.0557
Fax: 425.787.4641

<http://www.cypress.com/>

Copyright © 2006 Cypress Semiconductor Corporation. All rights reserved.

PSoC is a registered trademark of Cypress Semiconductor Corp.

"Programmable System-on-Chip," PSoC Designer and PSoC Express are trademarks of Cypress Semiconductor Corp.

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

The information contained herein is subject to change without notice. Made in the U.S.A.