2-6-BasicADC

Welcome back to Cypress Academy, PSoC 6 101. In this video I am going to show you how to use some of the analog features of the PSoCC 6. Specifically the SAR ADC, the voltage reference and the OpAMP which for some strange reason we call the CTM or CTB in our documentation.

To use the analog resources, we really need an analog signal for an input to the PSoC 6. Conveniently enough, On the CY8CKIT-028-EPD EInk shield that came with your CY8CKIT-062-BLE there is a thermistor which is perfect for taking analog measurements.

So let’s get started! Create a new project, I’ll call it 2-6-ADC. For this project I want to read the voltages of the thermistor circuit using the ADC and then convert those reading into a temperature and finally print all of the data onto on a terminal connected to the UART.

If you haven’t used a thermistor before, it is a temperature dependent variable resistor. In other words if you know the current through it, then you can calculate the temperature with a crazy nasty formula…. More on that in a minute … In order to use a thermistor, you put it in series with another precision resistor… one that doesn’t change very much with temperature… typically 0 point 1%. Let me show you a picture from the schematic of the e-ink shield schematic.

You can see that the thermistor is connected to A0 and A1, and the reference resistor is connected to A2 and A3… and the intermediate node in the resistor stack between them is shorted.

Typically the best way to do a thermistor measurement is ratiometric… meaning the absolute value of the voltages aren’t important rather, but the ratio of the voltages. This is how you get rid of common mode noise and offset in the measurement. The SAR ADC in the PSoC 6 can be configured to do differential measurements which enable these ratiometric calculations.

Lets get to the schematic.

First, Drop the uart onto the schematic next drop the adc and four analog pins.

Now I am going to show you something new. In all of the examples that I have show you so far you have added components to your project. All of these components have been INSIDE of the PSoC. Well just like putting a comment in your c-program … the PSoC Creator team gave you a similar ability in the schematic. You can add “Off-chip components” sometimes called “annotation componetns”. Thse components don’t actually DO anything to your psoc creator project… but they help you understand what your system looks like. These components … and wires are always in blue… and let me say it again… they don’t do anything to your project. They are just there for documentation.

Lets add annotations components for the thermistor and reference resistor to our schematic so that you can see what is going on. First, click on the “Off chip” tab in the component catalog… you find find a bunch of different annotation components which you can use for documentation. Let drag and drop a resistor and a thermistor… notice how they are blue … remember the blue wires and components are for documentation only.

Now, edit the resistor, change the value to 10K and turn off the instance name

Next edit the thermistor… set to 10K and turn off the instance name.

Now, lets look at the schematic again. Notice that there is no power source for the resistor stack? So… how does that work? Well we are going to power the stack from the PSoC by driving a logic high onto that A0 pin. All right Alan you mean to say that our pins can be both an analog input and a digital output at the same time… Yup. Sure enough. Let me show you.

Edit the first pin and change the name to A0, turn on the external connection, turn on the digital output and set to high… what this does is makes it so that this pin is both a digital output … so I can drive a one or a 3.3.v onto it… but at the same time it is also an analog input which can be routed to the differential input on the sar… How cool is that? The other new thing is that by turning on the external connection it will give you a terminal to hook the annotation components.

Now… remember from the schematic that there was not a ground on the stack. Well we will use the PSOC to connect a ground… also known as a logic low. How do we do that? Same way we did the high side. So, Edit the and change the name to A3, turn on the external connection and turn on the digital out and set the initial state to low

The other two pins are normal analog input pins… but lets change their names to A1 and A2 and turn on the external connections

Now lets wire it up… notice that while I am doing this the wires that I connect to the external pins are blue… remember that they don’t do anything in the project… they are just there for documentation.

The next thing that I want to do is configure the ADC. Notice that it was already setup for 2 channel differential measurements… first Ill double click it. Now with the default settings, the adc can only do differential measure from from minus vref to plus vreg… and that is only plus minus 1.2v .. but we know that we need to measure higher voltages…. So first change the VREF to the Vdda input to increase the range. When I do this, I am able to measure plus minus 3.3v … that will work…. Next I want to turn on averaging… then select 256 samples… averaging is effectively putting on a big low pass filter.. meaning it gets rid of noise. Notice that it slowed down the speed of the SAR.

Now that the schematic is done, Ill assign the pins… UART to P50 P51 … A0 to P10.0, A1 to P10.1 A2 to P10.2 and A3 to P10.3

I want to use printf .. so now Ill change the build settings to include stdio retargeting.

Now… run Generate application

I told you earlier that the actual temperature value of a thermistor is calculated with a big gnarly equation. Well… in PSoC 4 we have a thermistor calculator library… but I noticed that the lilbrary isn’t in PSoC 6… at least for now… so how do I get it there. Well… lets make psoc 4 project... file-new project … psoc 4… Ill call it p4therm… now add the thermistor to the project… now run generate application… and after a few seconds you will see a directory called Thermistor in the generated source… that is exactly what we need. So Ill copy the thermistor.h and then paste it into my cm4 header file… and then copy the thermistor.c and paste it into my cm4 source files.

Now… I will make the 2-6-adc project the active project again so that I don’t accidently hose myself… then Ill make one small change to the Thermistor.h… delete these two includes and just include “project.h”

Next Ill change stdio\_user.h to include project.h … and use the uart\_1\_hw…

And finally edit some firmware.

OK.. lets include stdio and Thermistor.h

Then in the main function I need to start the uart, start the adc… tell the adc to start running continuously…

In the main loop… I need to declare a couple of floats to be the voltage of the two inputs… and counts for the integer values that come back from the SAR..

Then I read the ADC channel 0 and assign it to the countReference.. remember channel 0 is the reference resistor… then do the same thing for thermistor.

Next Ill convert the counts from the two channels into volts.. this isn’t needed… but I do it just so that I can printout the voltage.

Ill call the thermistor library function to find out the resistance of the thermistor

Then convert that value into a temperature by calling the library function … this function actually returns temperature in 100ths of a degree celcuis… .and Id like to have it in degrees celcuis… so divide by 100.

Finally Ill print out the whole thing… and do a delay and then loop back to the start.

Now for the moment of truth. Hit program.

When I open up the terminal… I can see that 5 times a second I am seeing the voltages… and temperature.. how cool is that.

Now When I put my finger on the thermistor… it warms up … and yes I can see the new values.

Next Ill take my fluke meter with the thermos couple… and look there the fluke says nearly the same value as the psoc 6…

Finally lets measure the actual voltages… and look they are nearly the same as well.

What happens if you have a noisy power supply? Well this whole thing depends on your measurement of the reference resistor and the thermistor being taken with exactly the same input voltage… if the current changes because of power supply noise, you will endup with a less accurate reading. Id like to fix that. Let me show you how you can use the interal reference of the chip and an opamp to improve things.

First, Copy the the project and then paste it.

Now Rename it to 2—6-ADC\_OPAMP\_VREF

Change A0 back to just an analog pin. Fix the external component wire

Inside of the chip there is a very accurate reference voltage called VREF. In fact this is the signal that the SAR uses to get accurate measurement. But you can access this signal in your design. To do that Add the vref signal to your schematic.

To get the vreg signal to an output pin… first ill buffer it with an op amp. To do that Add an opamp to the project

Make it a follower… also known as an analog buffer and change it to output to a pin.

Wire it to A0

This time we know that the maximum voltage on the thermistor or the resistor will be less than one volt… so we change the adc vref setting to be the system bandgap.. which will give us a more accurate measurement.

Now we need to startup the opamp in the main\_cm4.c

Program it.

When I look at terminal program I see that I get input voltages around 0.6v at room temperature…

You can post your comments and questions in our PSoC 6 community or as always you are welcome to email me at alan\_hawse@cypress.com or tweet me at @askioexpert with your comments, suggestions, criticisms and questions.