Oscar

	Report Section	%	E	S	D	U	
1	Introduction, background, purpose etc.	5				x	0.0
2	Identify and Formulate problem/design goals (1a)	10		x			7.5
3	Develop design/solution (1b)	15			x		7.5
4	Acquire background knowledge (e.g., Data-sheets, papers, etc.) (7a)	10				x	0.0
5	Describe design judgements regarding global, economic, environmental, and societal contexts (4b)	10				x	0.0
6	Conduct Experiments, acquire data* (6a)	15		x			11.3
7	Conduct Simulations, make predictions* (6b)	0					0.0
8	Interpret Results, draw inferences* (6c)	10		x			7.5
9	Iterate, based on tests, simulations, results to improve design	10	x				10.0
10	Summary/Conclusion	5		x			3.8
	Total	**90**					**47.5**
	20-pt scale						**11.0**

Do you have a schematic? Or at least a block diagram?

Maybe even something at the PSoC level?

Code?

A person should be able to reproduce your results with the invformation in the report.

It's a good start, and youv'e got it working, which is the hard part, but I need more in the sections above that are in the "U" category.

Report

April 18, 2024

1 Lab 2 Report

This lab is about developing a USB Mouse for PSOC-5 using a potentiometer joystick and an i2C joystick from SparkFun

Problems for Potentiometer and I2C combined:

UART would not connect. Solution: remade project after installing KIT for Psoc 5

Not showing up on COM port. Solution: Hard reset the board by holding the reset switch at least 5 seconds.

ADC values were not displaying. Solution: sprintf on the psoc does not display floats properly. Typecast the value to int

No x or y data being put in mouse data array. Solution: Used double temp variable to calculate value then put that in mouse array

Tweaking values for usability. Normalized the adc output to between -40 and 40, which was too high for usable cursor movement. Lowered it incrementally until I settled on -10 to 10. The joystick did not need a software deadzone as the physical deadzone was plenty.

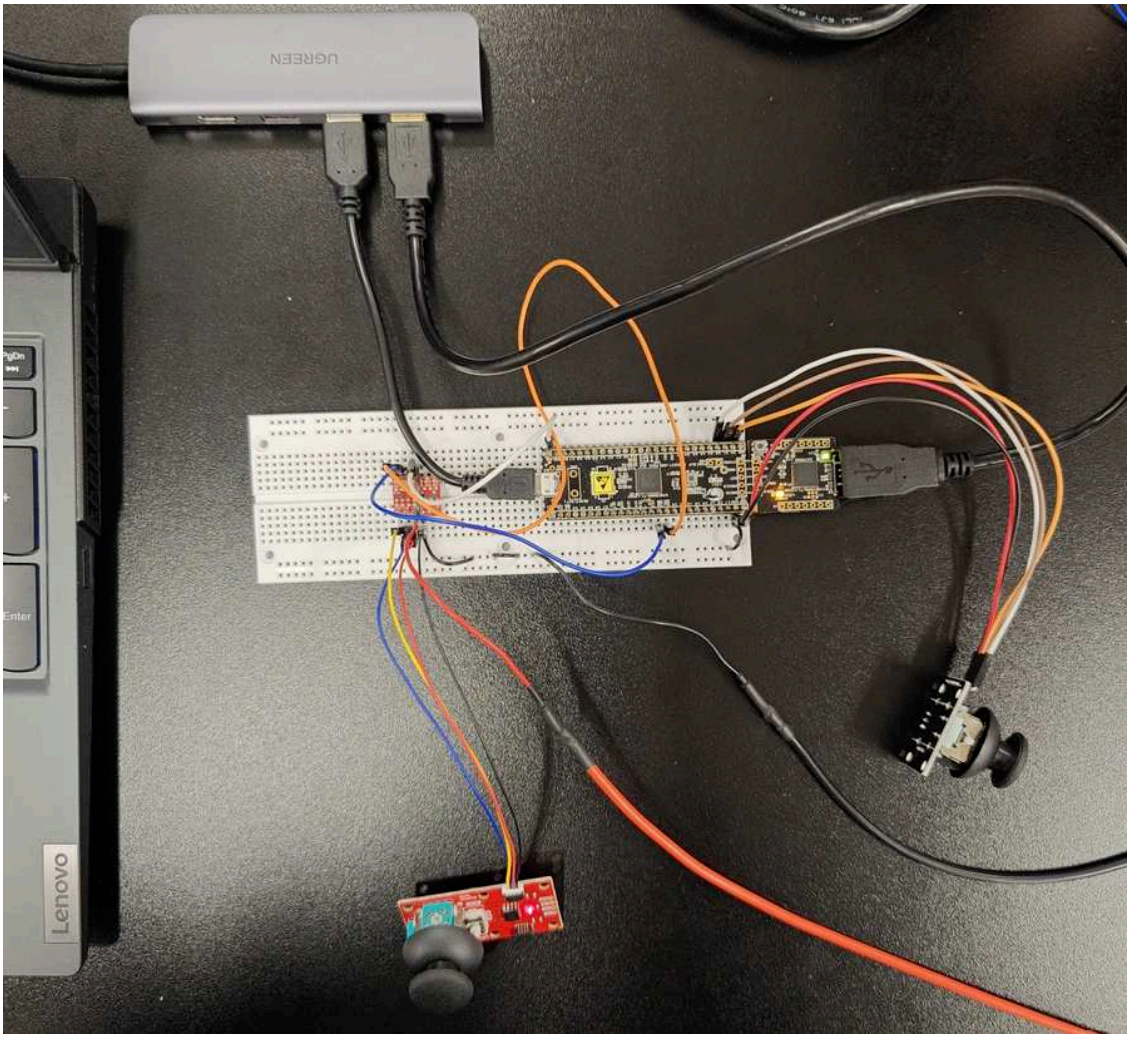
The raw ADC output is 7 bits unsigned. To normalize it, divide by 2^7 = 128 then multiply by the max value desired. In my case, I used 10. To center the data on 0 when the joystick was neutral, subtracted 10.

Mouse circuit.

The potentiometer mouse only used the ADC channels and power to read the joystick position.

The I2C joystick used the I2C bus and a level shifter to interact with the 3.3V signals from the sparkfun joystick

Good documentation of trouble shooting and iterative design.



Both joysticks functioning

The potentiometer joystick uses the ADC on the PSOC to read the voltage values on each of the joystick axes, then converts that into values that are sent through USB to the connected computer. The joystick is able to move the cursor horizontally and vertically as well as left click.

The I2C joystick, also connected to the PSOC, passed signals through a level shifter to control the cursor in the same way as the potentiometer joystick.

Demo video in demo

I started the project having only the ADC framework from Lab 1 to read the potentiometer joystick voltages. I also made a copy of the included example project for a USB Mouse, which moved the cursor in a square path to demo USB functionality.