P6-RC3-5b- Basic Motion Sensor

Welcome back to Cypress Academy, PSoC 6 101. In the last video I showed you how to use the PSoC 6 I2C master to interface with the Bosch BMI160 accelerator

Add the bosch library to your project

Make a folder called BMI\_driver

Add the existing items to it

Change build settings cm4 compiler … add additional include directories… then click add… then new then add the bmi\_driver directory

Add the i2c master to the schematic

Change it to master

Add a digitial output pin called led8… this pin wil be ON when you are in capsense mode and off when you are in motion mode

Assign the pins … p6[0] & P6[1]

Generate application

To start the firmware I am going to create motionTask.h. So, right click add new item.. then header file… then motionTask.h

First ill add #pragma once.

When I originally built this application, I had the capsense task and the motion task running at the same time and independently of each other… I thought that this would be great… but they fought with each other… and the robot arm went crazy and I had to use the kill switch. So I decided that what I would do is make the system be in two modes, either capsense mode or motion mode. FreeRTOS has the concept of event groups which is essentially and thread safe global variable.

The way this is going to work is that when the remote control is sitting still and flat for “a while” it will put them system in capsense mode. If you pickup the kit then it will go into motion controller mode.

To implement this you need to include the event\_groups.h… then make a definition for the event group called systemInputMode. this will actually be instantiated in main cm4.c. Now I make a bitmask for the two modes… the double left arrow is the shift operator.

Finally, ill define the motion task.

Now I want to update the main\_cm4.c to include the new stuff. First include the motionTask … and the event\_groups.h.

Next Ill make a variable for the Eventgroup called systeminputmode

Then in main ill initialize the event group, set the current mode to capsense and turn off the led8… the last change is main is to start the motionTask

In the capsenseTask.c I need to make two small changes, first, include the motionTask.h and seocnd to only call the writePositionFunction when you are in capsense mode

Finally the main event… I need to create the motionTask.c. Next I will copy from the basic motion sensor project main\_cm4. Specifically Ill copy all of the includes all the way through the top of main… then paste it into the motionTask.c

At the top I need to add includes for bleTask.h and motionTask.h.

Now scroll all the way down to the motion task. I told you earlier that I will control the mode of the system based on the motion. If the board hasn’t moved in about 1 second then Ill move the system into capsense mode. So I need to declare a variable which will keep track of the last time the board moved… called lastMovement.

In order to calculate the desired motor position I want to consider the x and y acceleration…. Meaning how the board is being held. In order to make the math easier I will cap the acceleration at plus minus 1 g.

Next I build a little converstion that will turn -1g to +1 g into 0 to 100% … right over my shoulder you should see the maths…

The next bit of code is used to keep track of the last movement… meaning if the board is noy flat and level meaning within 3 percent of 50% and 50% then update the lastMovement variable.

And if it has been more than 1000ms since the last movement set the system mode to capsense and turn on the led8

Otherwise set the mode to motion and turn off the led

Finally if we are in motion mode… send the motion information.

All right lets test this thing… hit program… and after a bit you can see the remote control turns on… connects… see the red light turn on… and the led8 turns on because the control is flat.

Now ill slide my finger back and forth… and look the robot arms moves.

Now Ill pick up the remote control… and look it turns off the capsense mode and you can see the arm move in both axis…

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.