2020F_ESE 3014_1

SEMESTER: 3rd

INSTRUCTOR: Linchen Wang

LAB: 5

DUE DATE: 1 Nov 2020 - 23:55

SUBMITTED DATE: 31 Oct 2020

STUDENTS: Amandeep Singh (C0765434)

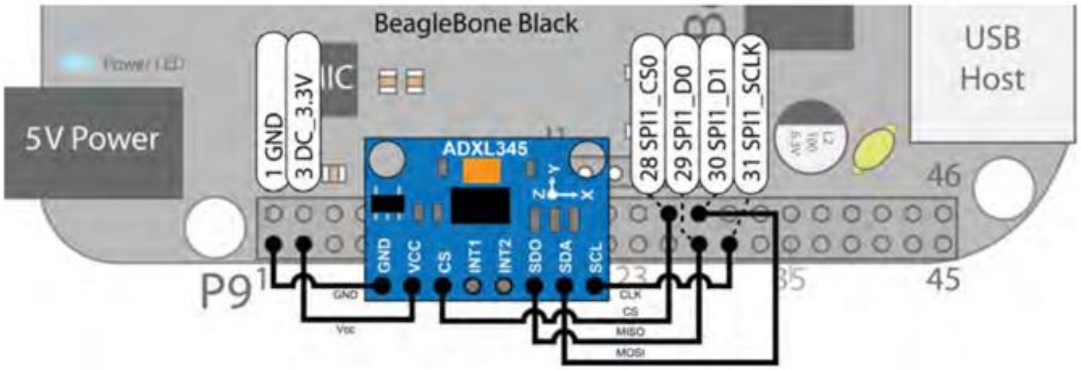
Lab-5

INTRODUCTION

In this Lab we are going to setup SPI interfacing with the BBB as master and ADXL345 as slave.

DESCRIPTION

- Pin Configuration:
  We use the following pin configuration for setting up communication between ADXL345 and BBB.



- You should have the spi overlays functional on you beaglebone

```
debian@beaglebone:~$ cd /dev
debian@beaglebone:/dev$ ls ./spi*
./spidev0.0  ./spidev0.1  ./spidev1.0  ./spidev1.1

./spi:
0.0  0.1  1.0  1.1
debian@beaglebone:/dev$
```

- To use these overlays we first have to disable the hdmi port on the beaglebone as the spidev1.0 and 1.1 are in use of it. So in order to disable it we navigate to "/boot/uEnv.txt" and add the following and reboot the debeaglebone

  ➔ optargs=quiet capemgr.disable_partno=BB-BONELT-HDMI,BB-BONELT-HDMIN
  ➔ $sudo reboot

- Now we just have to write the code to print the X,Y,Z values:
  NOTE: for the entire code please refer to the Appendix.

1.
```
char *pointer = "location of the SPI overlays";          //this segment of the code is
if ((t = open_file_to_read (pointer , O_RDWR)) < 0)      //to read the file using the
    {                                                    //pointer to the file O_RDWR
        Print (" unable to access the ADXL345 ");        //and if it's empty it means
        exit();                                          //that the pins were
    }                                                    //not accessed.
```

2.
```
int x = ((data[1] & 0x0X) * 256 + (data[0] & 0xFX));  //here we read the value from the
                                                      //registers and put them in x
if(x > 511)                    // this Is to simply limit the values of x co-ordinate
    {
        x -= 1024;             //this step is to ensure we get negative values on the axis
    }                          //as well
```
NOTE: the same code goes for obtaining both the y and z co-ordinates.

3.
```
do
    {
        Step 1;  // we place the above steps in a super loop
        Step 2;
    } while(1);
```

CONCLUSION

- Lab-5 was really important to understand how the SPI interfacing of ADXL345 accelerometer sensor works and it can used in the future for our capstone project.

APPENDIX

Youtube video link (please watch it on a phone for better experience):
https://www.youtube.com/watch?v=vg5lt4FEpHU

- CODE:

```
//HEADERFILES
#include <stdio.h>              //printf and scanf
#include <stdlib.h>             //for opening and closing a file
#include <linux/spi-dev.h>      //to read the spidev-overlays
#include <sys/ioctl.h>          //for ioctl function
#include <fcntl.h>              //for sleep() function
#include <unistd.h>             //for read and write funtions
```

```
//MAINFUNCTION of SPI.c
void main()
{
do
   {
    printf("\n");
    int t;
    char *ptr = "/dev/spidev1.1";
    if ((t = open(ptr, O_RDWR)) < 0)
            {
                    printf("unable to access ADXL345. \n");
                    exit(1);
            }

    ioctl(t, I2C_SLAVE, 0x53);

    char config[2]={0};
    config[0] = 0x31;
    config[1] = 0x08;
    write(t, config, 2);
    sleep(1);

    char rgstr[1] ={0x32};
    write(t, rgstr, 1);
    char data[6] ={0};
    if(read(t, data, 6) != 6)
            {
                    printf("Error 404: Something went wrong\n");
                    exit(1);
            }
    else
            {
                    int x = ((data[1] & 0x03) * 256 + (data[0] & 0xFF));
                    if(x > 511)
                            {
                                    x -= 1024;
                            }

                    int y = ((data[3] & 0x03) * 256 + (data[2] & 0xFF));
                    if(y > 511)
                            {
                                    y -= 1024;
                            }
```

```
                int z = ((data[5] & 0x03) * 256 + (data[4] & 0xFF));
                if(z > 511)
                        {
                                z -= 1024;
                        }

                printf("X , Y , Z : %d , %d , %d \n", x,y,z);
        }
    }while(1);
}
```