Lab 5   
Pengzhao Zhu   
Section: 112D

Initial Pseudocode (Lab 5)

Part A1 Pseudocode:

Call CLK function;

Call USART function;

character= “U”;

While () {

OUT\_CHAR(character);

}

\*Executable IN\_CHAR function- function return a received value

\*Executable OUT\_CHAR function- function takes a value and write to DATA register

\*Executable CLK function- function to set up 32Mhz clock.

\*Executable USART function- function to set up USART

Part A2 Pseudocode:

Call CLK function;

Call USART function;

Set Red LED as output

Turn off Red Led

CHECK:

Character= IN\_CHAR();

OUT\_CHAR(character);

if ((character != 'R') && (character !='r')) {

goto CHECK;

}

character=IN\_CHAR();

OUT\_CHAR(character);

if ((character != 'E') && (character != 'e')){

goto CHECK;

}

character=IN\_CHAR();

OUT\_CHAR(character);

if ((character != 'D') && (character != 'd')){

goto CHECK;

}

Toggle Red LED

goto CHECK;

\*Executable IN\_CHAR function- function return a received value

\*Executable OUT\_CHAR function- function takes a value and write to DATA register

\*Executable CLK function- function to set up 32Mhz clock.

\*Executable USART function- function to set up USART

Part B Pseudocode:

Void spi\_init(void) {

Select bit transmission order

Initialize control signals and SPI signals used by the uPad

}

Uint8\_t spi\_write(uint8\_t data){

Transmit a byte of data from XEMGA

While(transmission not complete) ;

Return data;

}

Uint8\_t spi\_read(void) {

Red a single byte of data from connect slave device

Return data;

}

Part C Pseudocode:

Call CLK function

Call SPI Initialization function

Character=0x53

While(1) {

Spi\_write(character);

}

Void spi\_init(void) {

Select bit transmission order

Initialize control signals and SPI signals used by the uPad

}

Uint8\_t spi\_write(uint8\_t data){

Transmit a byte of data from XEMGA

While(transmission not complete) ;

Return data;

}

Uint8\_t spi\_read(void) {

Red a single byte of data from connect slave device

Return data;

}

\*Executable code to configure 32Mhz clock

PART D Pseudocode

Call CLK function

Call SPI Initialization function

Call function to initialize LSM

Int a= WHO\_AM\_i;

Call accel\_read(a);

Call accel\_write(uint8\_t reg\_addr, uint8\_t data);

While(1) {  
}

\*Executable function- void accel\_write(uint8\_t reg\_addr, uint8\_t data)

\*Executable function- void accel\_read(uint8\_t reg\_addr)

\*Executable function- function to configure 32 Mhz clock

\*Executable function- Function to Configure LSM {

Enable device signal

Initialize size of data to be send.  
}

PART E Pseudocode

Call CLK function

Call SPI Initialization function

Call function to initialize LSM

Call function to initialize accelerometer

\*Executable function- void accel\_write(uint8\_t reg\_addr, uint8\_t data)

\*Executable function- void accel\_read(uint8\_t reg\_addr)

\*Executable function- function to configure 32 Mhz clock

\*Executable function- Function to Configure LSM {

Enable device signal

Initialize size of data to be send.  
}

void accel\_init(void) {

Perform a software reset of the LSM (setting a certain bit within the CTRL\_REG4\_A register)

Set up LSM interrupt signal to be triggered upon accelerometer completing an acceleration force measurement

Enable LSM to measure forces in the X,Y, Z coordinate plane

}

PART F Pseudocode

Call CLK function

Call SPI Initialization function

Call function to initialize LSM

Call function to initialize accelerometer

Plot the data in the X Direction

Plot the data in the Y Direction

Plot the data in the Z Direction

While(1) {

}

\*Executable function- void accel\_write(uint8\_t reg\_addr, uint8\_t data)

\*Executable function- void accel\_read(uint8\_t reg\_addr)

\*Executable function- function to configure 32 Mhz clock

\*Executable function- Function to Configure LSM {

Enable device signal

Initialize size of data to be send.  
}

void accel\_init(void) {

Perform a software reset of the LSM (setting a certain bit within the CTRL\_REG4\_A register)

Set up LSM interrupt signal to be triggered upon accelerometer completing an acceleration force measurement

Enable LSM to measure forces in the X,Y, Z coordinate plane

\*Executable function- function to transmit a stream of sensor data via USB Serial PORT

\*Executable ISR- ISR to run when LSM interrupt signal is triggered upon accelerometer completing an force measuremen