Question 1

Not yet graded / 1 pts

1. PWM Waveform Generation

1. Set up a timer to generate a 3 kHz square wave with a 33% duty cycle.
2. Output this waveform on one of the timer ports.
3. Measure waveform on the timer port using the AD2 oscilloscope. Record a video showing the waveform measurement, with the measured frequency clearly displayed, and upload to Canvas.

exam-q1.zip

Question 2

Not yet graded / 1 pts

2. Push Button LED Cycling

1. Configure P4.0 as a digital input
2. The switch S1 is connected to P4.0 on the EXP board. Enable the internal pull-up resistors for the switch.
3. Set P4.0 to interrupt on a falling edge (i.e. an interrupt occurs when the user presses the button). Enable local and global interrupts.
4. Write code to cycle through the LEDs connected to PJ.0, PJ.1, PJ.2, PJ.3, P3.4, P3.5, P3.6, and P3.7 each time the button is pressed. Specifically, each time the button is pressed, the current LED turns off and the next LED turns on. (Hint: use a state variable)
5. Record a video showing this behavior and upload to Canvas.

3-Axis Shock Sensor

1. Set up the ADC to periodically sample data from all three axes of the accelerometer (Ports A13, A14, A15) and shift the 10-bit results to 8-bit values.
2. Select a threshold acceleration value for each axis. If acceleration on each axis exceed the threshold acceleration (i.e. shock detected), turn on the LED on PJ.0, PJ.1, and PJ.2 for X, Y, and Z axes respectively.
3. Hold the LED on for ~2 s and then turn it off.
4. Each time the LED turns on (i.e. for each axis that detected a shock event), transmit a single packet to report the threshold acceleration and measured acceleration for that axis. No other packets should be transmitted while the LED is still on. Use the following packet format:

Start byte	Axis byte	Threshold data byte	Acceleration data byte
0xFF	0x01, 0x02, or 0x03	0x00 to 0xFE	0x00 to 0xFE

5. Place the MSP430EXP board on a table, tap it to create accelerations in the X, Y, and Z axes. Show the correct LED is lit for each axis.
6. Use the MSP430 Serial Communicator to check the transmitted message is reasonable and the acceleration data byte is greater than your defined threshold.
7. Record a video and upload to Canvas.

Question 4

Not yet graded / 1 pts

4. Inclinometer

1. Set up timer A to trigger an interrupt every 10 ms (i.e. 100 Hz).
2. With each timer interrupt, trigger the ADC to sample from the X-axis of the accelerometer.
3. Set up a circular buffer to store the previous 160 ms of data (i.e. 16 samples) from the X-axis of the accelerometer.
4. With each new data point, calculate the running average of the previous 16 data points by simply summing the results in a 16-bit variable.
5. Set up TB1.1 to output a PWM waveform to P3.4 to control the brightness of LED5.
6. Use the average X-axis acceleration to control the TB1.1 PWM duty cycle, which will control the brightness of LED5. You will have to scale the acceleration result and/or the PWM range to generate an appreciable brightness change between 0 and 90°. Hint: use left and right shifts to multiply and divide by 2.
7. Record a video of your inclinometer and upload to Canvas.

Question 5

Not yet graded / 1 pts

5. Computer Controlled Glowing Orb

1. Set up TB1.1 to output a PWM signal to P3.4.
2. Write code to slowly vary the brightness of the LED back and forth to create a glowing orb effect. Hint: vary the PWM duty cycle in the infinite loop and use a delay loop with _NOP() statements to generate a delay time to hold each brightness state. Remember to declare your variables as 'volatile'.
3. Set up the UART to operate at 9600 baud, 8, N, 1, to accept incoming data.
4. Use a circular buffer to temporarily store the following message packet:

Start byte	Command byte	Data byte
0xFF	0x01, 0x02, or 0x04	0x00 to 0xFE

5. Create a glowing orb with the following behavior:
    1. If Command byte == 0x01: the orb brightness changes from dark to bright, and then repeats
    2. If Command byte == 0x02: the orb brightness changes from bright to dark, and then repeats
    3. If Command byte == 0x04: the orb goes dark
    4. The data byte controls the oscillation period. Note: You may need to scale the data byte to make the oscillation period appreciable.

Record a video of your glowing orb and upload to Canvas.