# EE/CPE 3280 Assignment – Tilt-Control Star Blasters

## Program: 100 Points

Replace the joystick controls for Star Blasters with tilt control using a 3-axis accelerometer.

## Basic Operation

Star Blasters is a simple cockpit-view game in which you pilot a spaceship and blast stars. The game shows a view into space out the cockpit window of a space ship. The space background is punctuated by approximately 100 stars. A set of crosshairs highlights the aiming point for your ship’s blaster; the crosshairs may be moved in any direction by tilting the game system. When the user taps the screen once, a blast will be fired at the crosshairs’ position; double-taps will trigger a larger “hyper blast”.

## Functional Specifications

1. Stars. Upon startup and throughout the game, 100-200 stars shall be displayed on a black background. Stars shall be indicated by single pixels colored white and arrayed in a pseudo-random fashion. Stars shall be redrawn if erased by crosshair movement or blasts. Twinkling and scrolling are not required.
2. Crosshairs: A set of crosshairs shall be displayed on the screen. The crosshairs may be any suitable shape or color but shall have a size of between 20 and 40 pixels on each side.
3. Crosshair movement. The crosshairs shall move in the direction indicated by the tilt of the screen. When the game system is level, the crosshairs shall not move. Titling the screen left, right, up or down shall cause the crosshairs to move left, right, up or down, respectively. The crosshair speed of movement shall be proportional to the incline of the screen: A full tilt will cause the crosshairs to traverse the entire screen in approximately one second. It is acceptable if movement stalls during a blast or hyperblast.
4. Blaster. When the user taps the system, the game shall display an “explosion” sequence centered at the location of the crosshairs. The explosion sequence shall be 40-60 pixels in diameter and shall last no more than 1 second. 80% or more single-tap actions shall be recognized correctly.
5. HyperBlaster. When the user double-taps the system, the game shall display an extra large explosion sequence centered at the location of the crosshairs. Size, color, shape and duration of the hyperblast sequence are at the discretion of the programmer. 80% or more double-tap actions shall be recognized correctly.

## Implementation SPECifications

1. The X- and Y-Axes of an ADXL345 Accelerometer shall be sampled at a rate of 20 samples/s used to sense tilt. The sample rate shall be controlled by using a timer interrupt.
2. The *tap* and *double-tap* interrupt functions of an ADXL345 shall be used to sense taps and double-taps.
3. Communications between the ADXL345 and PSoC4 shall be through I2C.

### Important Implementation Notes

1. Start by just reading the accelerometer and displaying the x, y and z values on the screen. When the accelerometer is level, x and y should be near 0 and z should be a value that represents 1g. Tilting the accelerometer should change these values. The value of 1g will depend on the scale you set up using DATA\_FORMAT.
2. The I2C component uses interrupts. Since I2C reads/write will be done inside of the timer interrupt, the I2C interrupt must be at a higher priority (lower priority number) than the timer and tap interrupts. Configure this in the “Interrupts” tab of the \*.cydwr view.
3. After confirming that the PSoC is getting good accelerometer data, set up the crosshairs movement. Save the taps for last – you will need to configure several ADXL345 registers to get the taps and double-taps to work. I recommend using a single interrupt for both taps and double-taps and reading the ADXL345 INTSRC register to determine the source of the interrupt.
4. Main should simply configure the ADXL345 (see PPT notes for details), draw stars, set up the timer and interrupts, and go into a loop.
5. Timer interrupt should sample the Accel, move the crosshairs, and redraw the stars (twinkle them if you want).
6. Tap interrupt should read the ADXL345 Interrupt type and make either a large or small blast.

## TO TURN IN Through Canvas

Turn in the following:

1. Submit a single MS-Word document containing the following through Canvas
   1. Top-level schematic (select all of your drawing, copy and paste it into the Word doc)
   2. main.c (Select all text with ctrl-a, copy and paste into the Word doc)
   3. Modified portions of all ISR .c files (usually just the “Includes” portion and the “Interrupt” portion). Make sure that you have a header at the top of each file that identifies its purpose.
   4. Copy in any other files that your created or modified
2. Demonstrate your program to the instructor during class time on the due date. (Online students may send a video demo to the instructor)
3. Return your accelerometer and any borrowed equipment to the instructor.