Project Lab Report

Game: Block Dude

## Overview

I implemented the game Block Dude from the TI graphic calculators where the goal of the puzzle is to get to the door located at the furthest right top light of the LED matrix by finding a way to reach the door.  Block Dude can carry blocks to make steps to reach high places.

Once you win, the matrix will display green lights, and the LCD will display "Complete". If you get stuck and cannot carry blocks, you lost, and can reset the game anytime.

## User Guide

*Display:*
Green Light - Block Dude
Red Lights - Movable Blocks
Blue Lights - Bricks (Not movable)
Blue/Red - Door

*Controls:*
Tilt Left - Block Dude moves one block left.
Tilt Right - Block Dude moves one block right.
Tilt Downwards toward you +  direction the block is in.

e.g., if you want to pick up movable block that's to the right, you tilt in the board down towards you to the right in a diagonal motion.

Tilt in the same direction again to pick up a block

Tilt Upwards toward you + direction

e.g. if you want to move up one step, you tilt the board upwards in the direction you desire in a diagonal motion.

Press button - Turns on game/ Resets

*Special Consideration (Things to be aware):*
When making a tilt, it has to be brought back to the initial state (flat) to receive another input.

## Components

*Main:*

- 8x8 RGB LED Matrix
- ADXL335 - 5V ready triple-axis accelerometer (+-3g analog out)
- ATmega1284-PU

*Other:*

- 5 Shift registers
- 16x2 LCD Display
- Push Button
- Eight 330 Ω Resistors
- Three 10kΩ Resistors
- AVR Atmel Studio 6

**Complexities**

1. 8x8 RGB LED Matrix:

I used all three colors: Green (Block Dude), Red (Movable Blocks), Blue (Non-movable bricks), and Red/Blue/Purple for the Door

For each color, I used a shift register.

2. (ADXL335) accelerometer:

I used several resources to learn how to receive input, program and calibrate.

Resources:

https://www.sparkfun.com/datasheets/Components/SMD/adxl335.pdf
http://learn.adafruit.com/adafruit-analog-accelerometer-breakouts/programming
http://chionophilous.wordpress.com/2011/08/26/accelerometer-calibration-ii-simple-methods/
http://www.engineersgarage.com/embedded/avr-microcontroller-projects/adc-circuit

3. Communication between two microcontrollers via USART, but I ran out of time for this.

I also used shift registers to minimize the number of pins used on the microcontroller.

**Source Files**

*Main*:

FinalProject.c - has the whole program which includes six state machines.

SM1 Displays the Block Dude
SM2 Displays the movable Red Blocks
SM3 Displays the Blue Bricks
SM4 determines the direction in which the board is being tilted via Accelerometer
SM5 determines if the Reset Button is pushed
SM6 checks to see if player won

This file also has functions:

*unsigned short ADC_read(unsigned char ch)* - converts inputs to ADC values *unsigned short equilibrium(unsigned char x, unsigned char y, unsigned char z)* - checks to see if board is flat (t equilibrium)

*void transmit_data(unsigned char data, unsigned char regNum)* - [from lab (modifired)] transmits data to a specified shift register

*Includes:*
blockcheck.h - provides functions to determine if blocks (either red or blue) exist above/below/left/right of block.

io.c / io.h - [from lab] provides functions to use LCD Display

lcd_shift_reg.h - provides the *void transmit_data_lcd(unsigned char data)* function in order to transmit data to the LCD via shift register

scheduler.h - [from lab] provides functions to use the task scheduler in the main function

timer.h- [from lab] provides a timer to be used for the synchronous state machines

**Youtube Link**:

Demo: https://www.youtube.com/watch?v=Et6wDDQ-Jvg

Game Inspiration: https://www.youtube.com/watch?v=ErFXHHYXByk