Gravity Byte

Custom Project Final Report

Winter 2018

Michael Tang

Table of Contents

[**Introduction**](https://docs.google.com/document/d/1-xy6VmtY8k6tF-gSGBFLczn0Hhce3uhBp-F_ejkmoHQ/edit#heading=h.n9o44neiquk5)[**2**](https://docs.google.com/document/d/1-xy6VmtY8k6tF-gSGBFLczn0Hhce3uhBp-F_ejkmoHQ/edit#heading=h.n9o44neiquk5)

[**Hardware**](https://docs.google.com/document/d/1-xy6VmtY8k6tF-gSGBFLczn0Hhce3uhBp-F_ejkmoHQ/edit#heading=h.vqmbsuo5p330)[**2**](https://docs.google.com/document/d/1-xy6VmtY8k6tF-gSGBFLczn0Hhce3uhBp-F_ejkmoHQ/edit#heading=h.vqmbsuo5p330)

[Parts List](https://docs.google.com/document/d/1-xy6VmtY8k6tF-gSGBFLczn0Hhce3uhBp-F_ejkmoHQ/edit#heading=h.xtqh2tzibtrr) [2](https://docs.google.com/document/d/1-xy6VmtY8k6tF-gSGBFLczn0Hhce3uhBp-F_ejkmoHQ/edit#heading=h.xtqh2tzibtrr)

[Pinout](https://docs.google.com/document/d/1-xy6VmtY8k6tF-gSGBFLczn0Hhce3uhBp-F_ejkmoHQ/edit#heading=h.8yptvhjzojq8) [3](https://docs.google.com/document/d/1-xy6VmtY8k6tF-gSGBFLczn0Hhce3uhBp-F_ejkmoHQ/edit#heading=h.8yptvhjzojq8)

[**Software**](https://docs.google.com/document/d/1-xy6VmtY8k6tF-gSGBFLczn0Hhce3uhBp-F_ejkmoHQ/edit#heading=h.w9543qe124on) **4**

[**Complexities**](https://docs.google.com/document/d/1-xy6VmtY8k6tF-gSGBFLczn0Hhce3uhBp-F_ejkmoHQ/edit#heading=h.sn48u4uktu3c) **6**

[Completed Complexities:](https://docs.google.com/document/d/1-xy6VmtY8k6tF-gSGBFLczn0Hhce3uhBp-F_ejkmoHQ/edit#heading=h.tp9dnsse4u9p) 6

[Incomplete complexities:](https://docs.google.com/document/d/1-xy6VmtY8k6tF-gSGBFLczn0Hhce3uhBp-F_ejkmoHQ/edit#heading=h.99jbzhpy2eks) 6

[**Youtube Link**](https://docs.google.com/document/d/1-xy6VmtY8k6tF-gSGBFLczn0Hhce3uhBp-F_ejkmoHQ/edit#heading=h.u9cxbem9510r) **7**

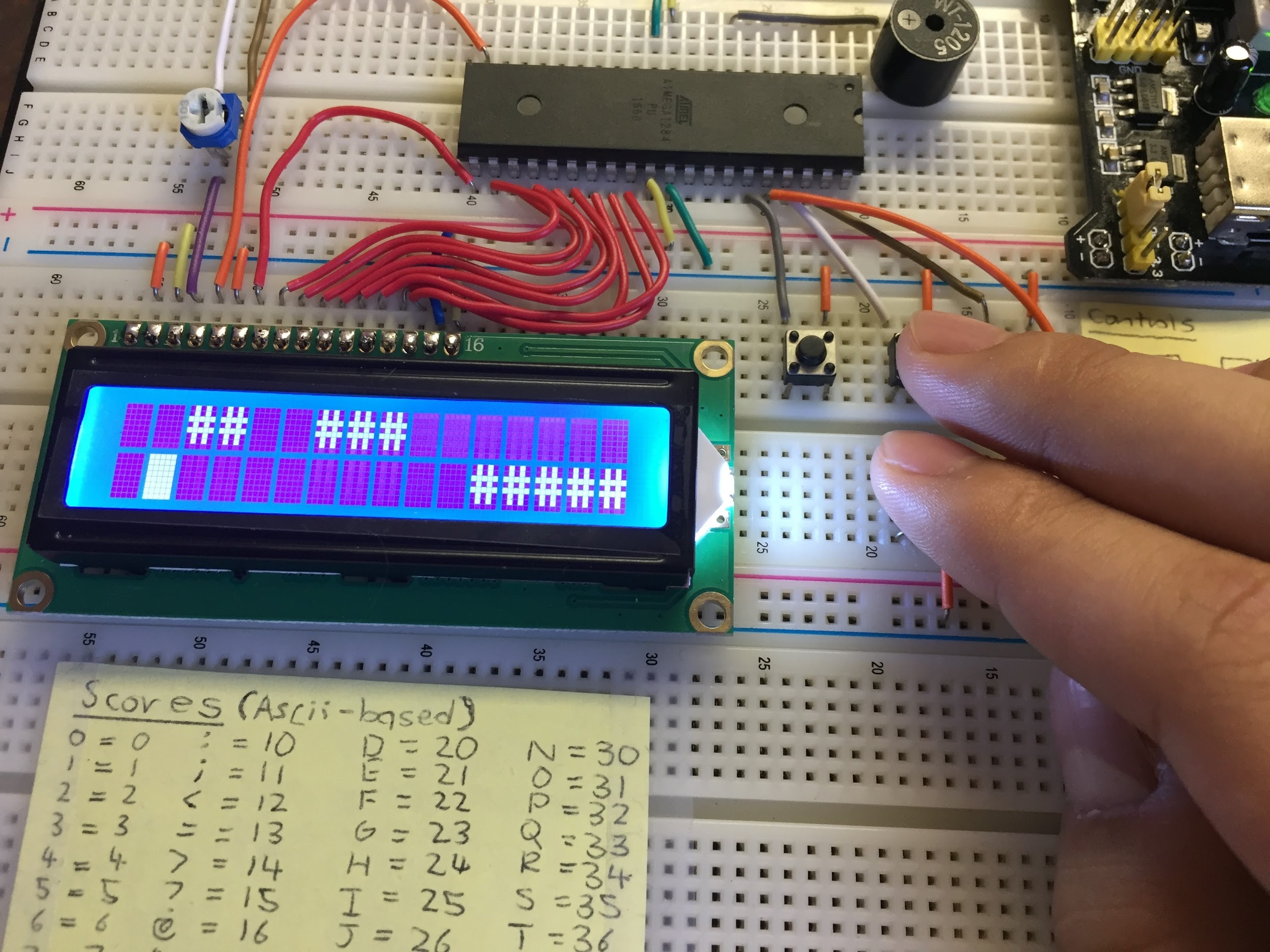
**Source Link 7**

[**Known Bugs and Shortcomings**](https://docs.google.com/document/d/1-xy6VmtY8k6tF-gSGBFLczn0Hhce3uhBp-F_ejkmoHQ/edit#heading=h.grlzpb6vy2cq) **7**

[**Future work**](https://docs.google.com/document/d/1-xy6VmtY8k6tF-gSGBFLczn0Hhce3uhBp-F_ejkmoHQ/edit#heading=h.qbv6f31drpex) **7**

# Introduction

Gravity Byte is a game based off an mobile game called, “Gravity Guy”. In this side-scrolling jumping game, you, the player, manipulate your own gravity to ‘jump’ to several platforms to avoid the obstacles, in this game, ‘#’. Similarly to an arcade game, the goal is to earn the most amount of points as you can, also known as your score. There are 4 buttons that manipulate the game. The first button starts the game(rightmost). The second and third buttons make the character jump up and down respectively(middle buttons). Finally, the last button pauses/resets/ and unpauses the game(leftmost).



# Hardware

## Parts List

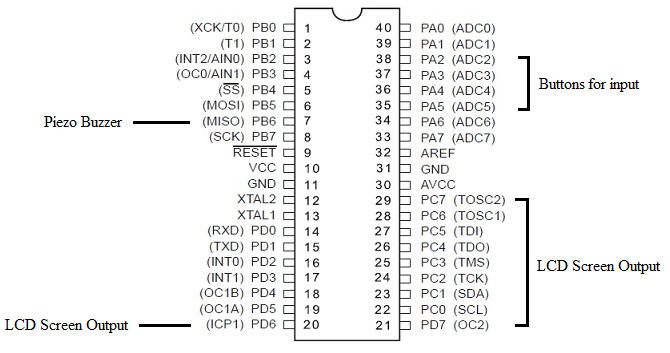
The hardware that was used in this design is listed below.

* ATMega1284 microcontroller
* 16x2 LCD Screen
* Potentiometer
* Buttons
* Piezo Buzzer

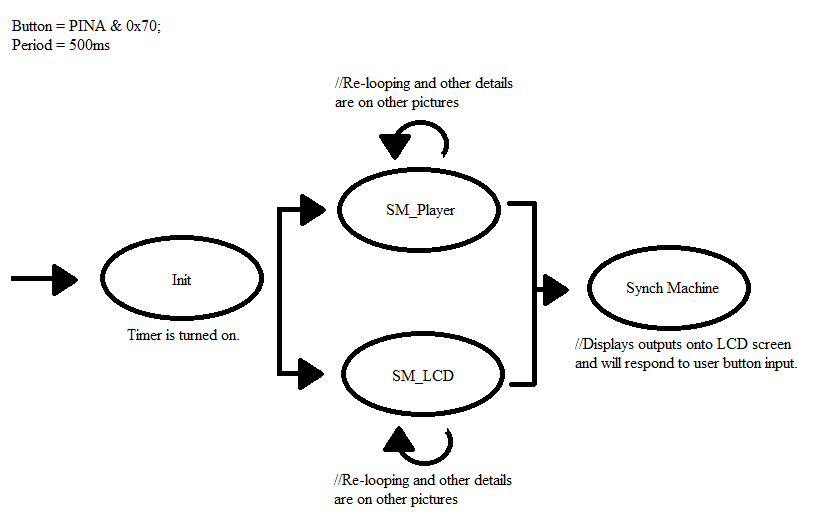
The software that was used in this design is listed below.

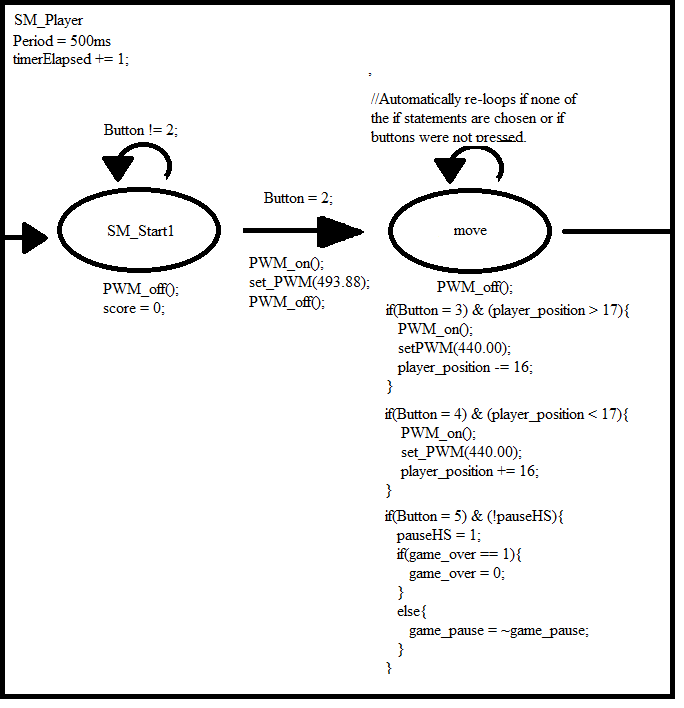
* Atmel Studio 7

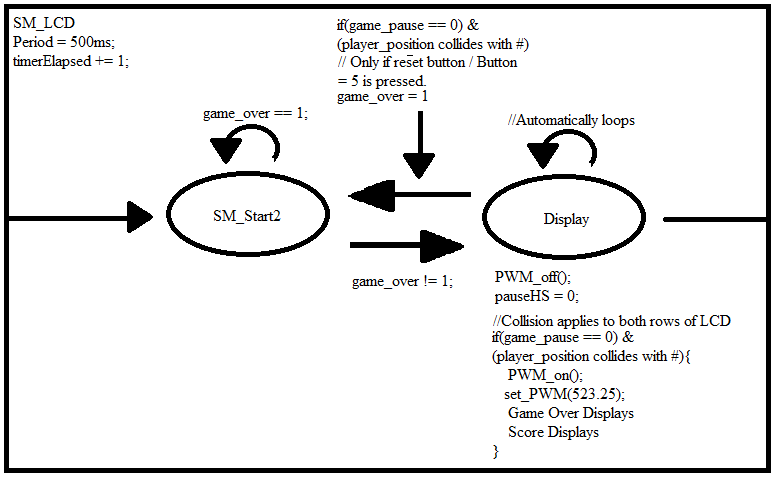
## Pinout



# Software







# Complexities

## Completed Complexities:

* Synchronous SM logic
* Software logic in conjunction with parts.
* Title screen and start button logic.
* Buzzer sounds.
* Reset button logic.
* Score keeping and score resetting.

## Incomplete complexities:

* EEPROM implementation for High Score
* Proper Display of Scores (i.e. Numbers instead of ASII)

# Youtube Link

<https://youtu.be/k5dNHMkE7XY>

Disclaimer: As annotations are discontinued as of May 6th a few years ago, I have added some necessary notes in the description regarding the Piezo Buzzer ”Build-Upon”.

Source Links

<https://drive.google.com/file/d/1QryadwTVfDz8KAcSBhK5p_HybRrfgIaL/view?usp=sharing>

Main.c (This file provides the logic of the jumping game. This file is a product combining code from Lab 10 Part 5, reference PWM code from Lab 9, and self-produced code).

<https://drive.google.com/file/d/1UdGKzZxfMvh7RLAXbcBBCXaQFYYKukBr/view?usp=sharing>

Io.c (This file contains the commands and each of its’ functions to the LCD. This file originates from Lab 10.)

<https://drive.google.com/file/d/1Ox0Srb5hXiabaKJtxv73ooFnIxSoYpst/view?usp=sharing>

Bit.h (This file receives and gives values of bits/input values. This file originates from Lab 4.)

<https://drive.google.com/file/d/1eJRrZjRU9b1GbdWwdLTAv2maQLxZd8CJ/view?usp=sharing>

Timer.h (This files controls the timer functions and commands. This file originates from Lab 4.)

# Known Bugs and Shortcomings

* After the first time playing the game, when you click the start button it doesn’t always start right away, sometimes you need to click the button multiple times. I believe that this is caused by not properly resetting some of my variables.
* The Game Over screen would only write halfway if you die at the top side of the LCD screen, leaving the second row to display an empty space with two ##. I believe that there might have been some error with the transitions between states.
* The Score Display does not properly display a score. Instead of numbers, ASCII characters are displayed instead.

# Future work

* Add an EEPROM for storing the High Score within the system even if the system has been turned off.
* The player sprite will be implemented in the future.
* Create seperate pause and reset buttons.
* Extra Lives and Versus Options will be implemented.
* Properly implement the display of score.