# YAPG - Yet Another Pong Game

*Term Project*
*of*

**CSE 316**
**Microprocessors, Microcontrollers, and Embedded Systems Sessional**

*Submitted by*

**Shehabul Islam Sawraz - 1805088**
**Mohammad Abrar Nafee Akhand - 1805089**
**Mobaswirul Islam - 1805090**

*Under the kind guidance of*

**Dr. Md. Monirul Islam**
**Dr. A. B. M. Alim Al Islam**
**Masum Mushfiq**

**Department of Computer Science and Engineering,**
**Bangladesh University of Engineering and Technology**

# Introduction

For our CSE-316 term Project, we created YAPG - Yet Another Pong Game. Pong is a simple "tennis-like" game that features two paddles and a ball. When one of the players misses hitting the ball with his paddle, the other gets a point. The goal is to defeat the opponent by being the first one to gain 9 points.

# Motivations

Pong is one of the very first computer games ever created. For our first hardware project, we wanted to honor this piece of history by recreating it with Microcontroller(ATMega32).

# Description

In our project, we framed the game in two LED Matrices (16x8 display). The players have to use their hand to move the paddle. We used Sonar to track the distance. There is an LCD Display to show the current status of the game i.e. the current scores of the players. We also used Vibration motors and Buzzers to give the players auditory and haptic feedback whenever they get points.

# Instruments

| **Instrument** | **Quantity** |
|---|---|
| Microcontroller(ATMega32) | 1pc |
| LED Matrix | 2pc |
| LCD Display | 1pc |
| SONAR Sensor(HC-SR04) | 2pc |
| Buzzer | 2pc |
| Vibration Motor | 2pc |
| Decoder(74HC154) | 1pc |

# Block Diagram

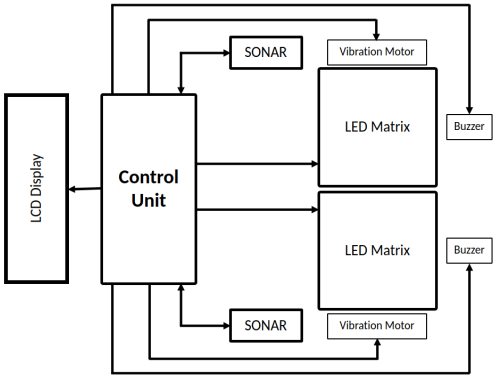


Figure 1: YAPG - Block Diagram

# Circuit Diagram

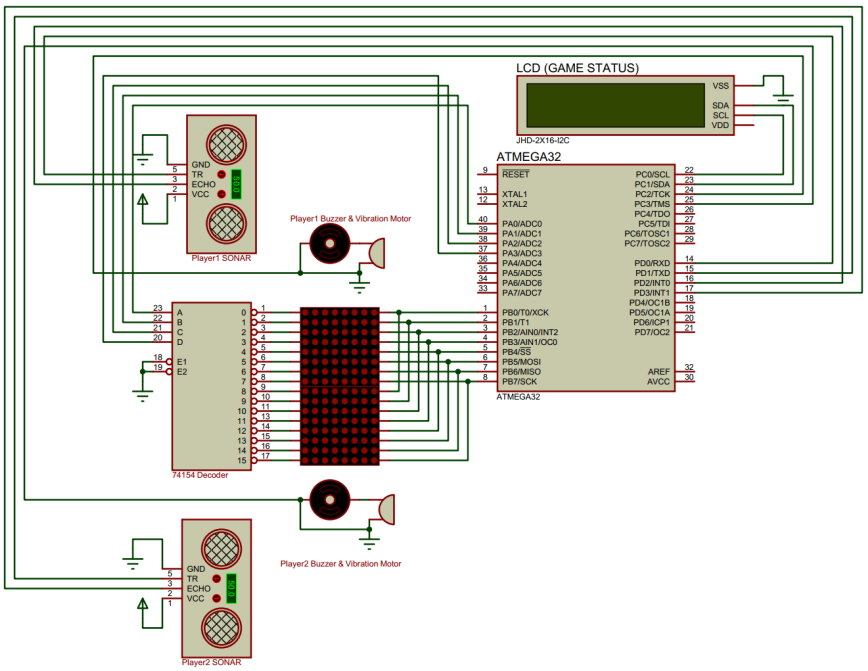


Figure 2: YAPG - Circuit Diagram

## Difficulties

1. As we have used two led matrices, we had to deal with a 16x8 matrix. There were total *16 rows and 16 columns*. But there were limited pins in ATMega32. So to utilize the pins efficiently, we, at first, shorted the columns (8 columns) and used a 74HC154 **decoder** to select rows one by one. In total, we used *12 pins* to operate two LED matrices. We could have used shift register instead of decoder.

2. Our initial plan was to use 7 segment displays to show the scores. But they required a lot of pins to operate. So we used LCD display(that uses I2C communication protocol) to do it instead. This lessened our number of required pins by 12.

3. A lot of I2C LCD libraries were available online but almost all of them were for Arduino. So we took help from our seniors and used a library they had found.

4. As we used ultrasonic sonar sensors, we had to use **external hardware interrupts**. We decided to call our interrupt service response on *both rising and falling edges* so only **INT0** and **INT1** were available to use as INT2 can only support one edge response. So we were limited to two pins for interrupts. But to implement both horizontal and vertical movements, we had to use 4 sensors and that would require 4 pins. We tried to to it with multiplexers but for some reason (might have been propagation delay) it just would not work. So, later we restricted pad movements to horizontal axis only.

5. As our board's dimension was only 16x8, it seems there were these cases when the ball kept moving in a loop if the players don't move their pad. To resolve this *loop issue*, in code, we tried to detect a loop and give the ball a random movement(after collision with pad).