$\underline{A}^2\underline{B}^2$

Ping Pong

ATMEGA32 and Processing based Ping Pong video game.

Project by,

Aravind Kumar-2013105004,

Ashvath Nirmal-2013105005,

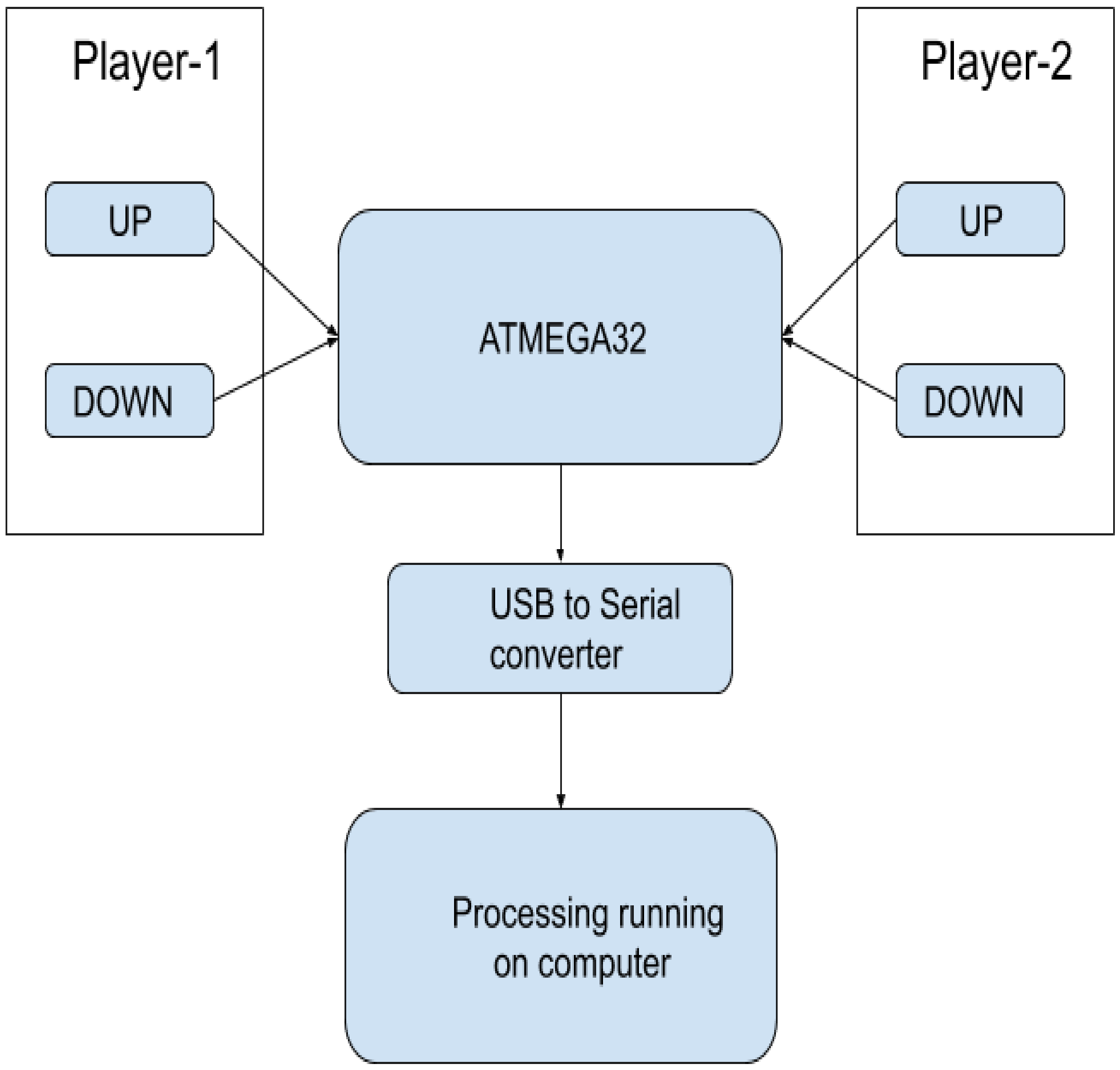
Bharath.A-2013105006.

Table of contents:

Introduction:

Ping Pong is a game between two players where each of one get to handle a paddle and collide with the incoming ball,whichever player making another leave the ball wins the game.In this project **ATMEGA32** microcontroller is used to interface with the buttons to control the paddles and a **PROCESSING** software on computer is used to display the game on the PC.A **USB-Serial converter** is used to interface between computer and the microcontroller.The software part is purely on processing and hardware relies on the Atmega32 microcontroller.

Block diagram:



ATMEGA32:

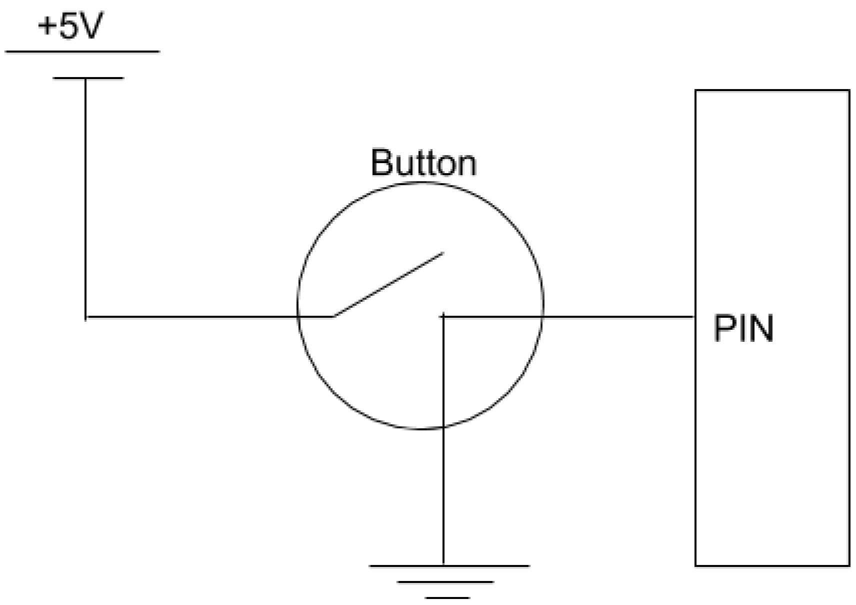
Atmega32 is an 8 bit microcontroller from Atmel corporation.The following are the technical specifications,

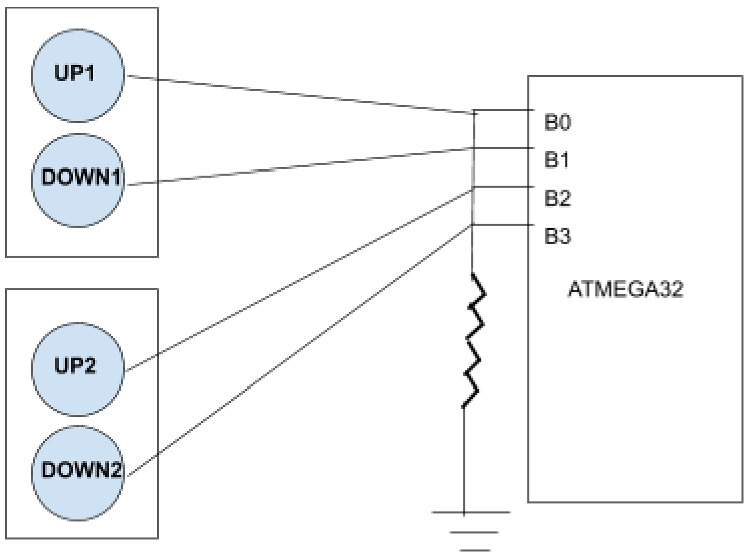
Parameter	Specification
Operating voltage	2.7 - 5.5Volts
Maximum operating frequency	16Mhz,default at 1MHz
I/O pins	40
Ports	A,B,C,D
USART available	1TX and 1RX

The hardware side of this project relies on this microcontroller, where the button interfacing and Serial communication to the computer is carried out.The four buttons(2 for each player) is connected to **PORTB, pins 0,1,2 and 3** by means of **PULL DOWN 1K resistors**.The communication from controller to PC is through **Serial** means, the controller sends data serially according to the button pressed through the TX pin, which is inturn connected to the USB-Serial converter,interfacing will be explained later.

A.Button interfacing:

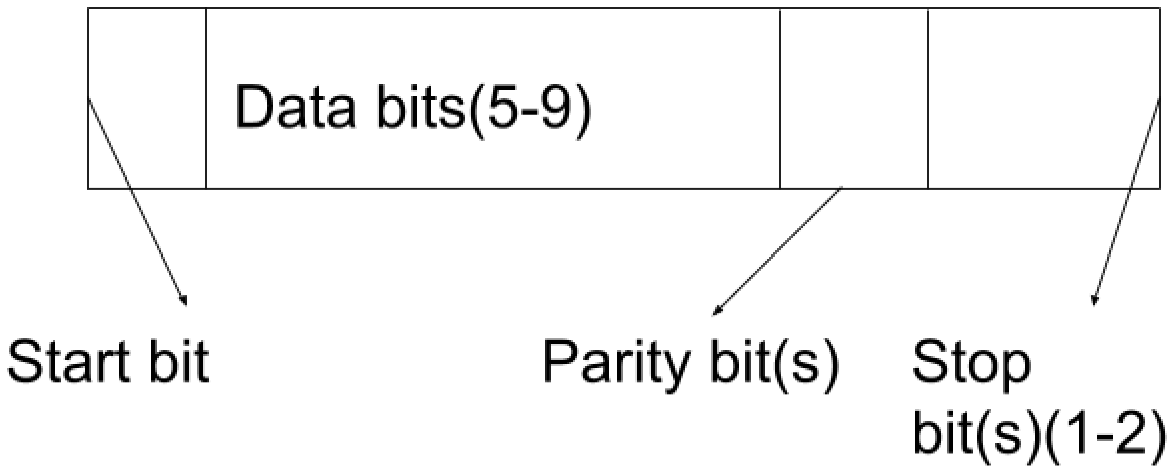
There are four buttons connected to the controller via pull down resistors to PORT B,each of two are used to move the paddle up and down.The circuit diagram for the buttons connection is shown below,





B.Serial Communication:

Serial communication refers to the process of sending the data one after another as the name implies.ATMEGA32 is capable of Serial communication by means of TX and RX pins.Generally there are two types of Serial communication, Synchronus which involves CLK and asynchronous which is void of the CLK.In this projct we are using the controller in asynchronous mode.There is a standard form for transmitting the bits which is accepted universally and is known as **UART-Universal Asynchronous Receiver and Transmitter**.The speed at which the bits change (or) speed at which the bits are transmitted is called the **BAUD RATE**.There is a **specific format** for the data tansmission of **UART**, below is the structure.



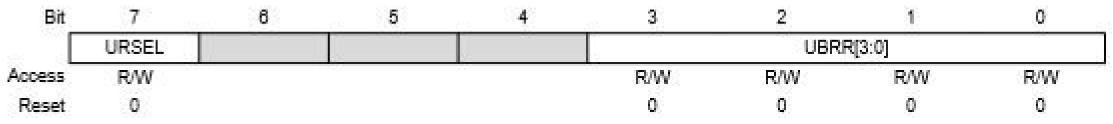
Setting the BAUD rate:

The BAUD rate of the controller is set up by loading the **UBRRH** and **UBRRL** registers in the controller with the number to countdown for the setting.The formula for calculation of number to be loaded in the UBRR and UBRRL registers is,

F_CPU / (16 * baudRate)) - 1

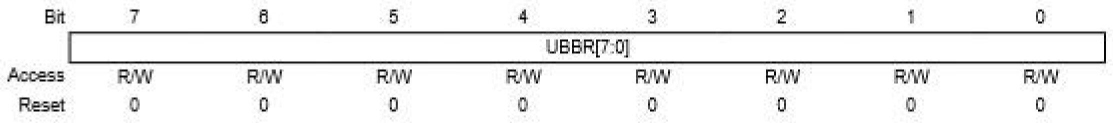
Where the F_CPU is the running clock frequency of the controller,here 1 MHz,Baud rate is baudRate defined above,here it is **2400**.The multiplying factor 16 is used because initially 16 MHz clock is divided to get the default clock of 1 MHz.

UBRRH-Holds 4 Most Significant Bits.



Bit	7	6	5	4	3	2	1	0
	URSEL	▨	▨	▨	UBRR[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

UBRRL-Holds 8 Least Significant Bits.



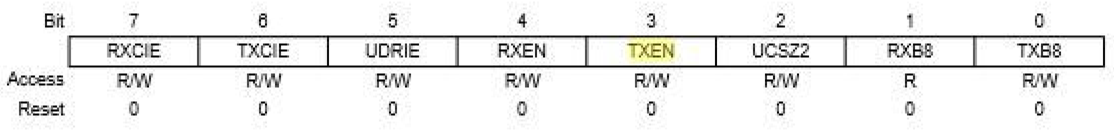
Bit	7	6	5	4	3	2	1	0
	UBBR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

So the number to be loaded in registers is **25**,converted to binary it is 00011001, so
UBRRH=0000
UBRRL=00011001

The **UMSEL** bit in the UCSRC register is used to set the USART in Synchronous mode/Asynchronous mode.

Here **UMSEL=0**,for Asynchronous mode of operation.

Transmitter and receiver bits are enabled in the UCSRB register to start transmitting and receiving.So TXEN and RXEN are set to 1.



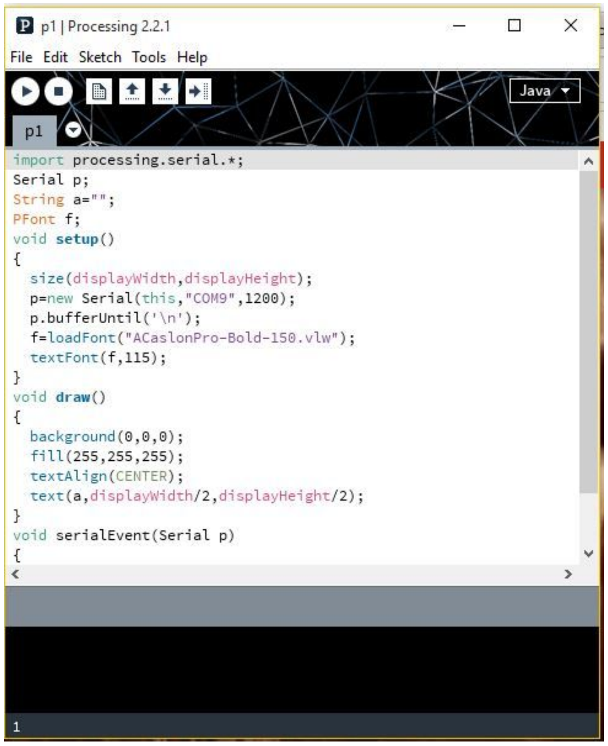
Bit	7	6	5	4	3	2	1	0
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Reset	0	0	0	0	0	0	0	0

Processing:

Processing is an open source programming language and integrated development environment (IDE) built for the electronic arts, new media art, and visual design communities with the purpose of teaching the fundamentals of computer programming in a visual context, and to serve as the foundation for electronic sketchbooks.

The program for the Ping Pong was purely written in processing environment in **C** language.The input is got from the **Serial COM port**.

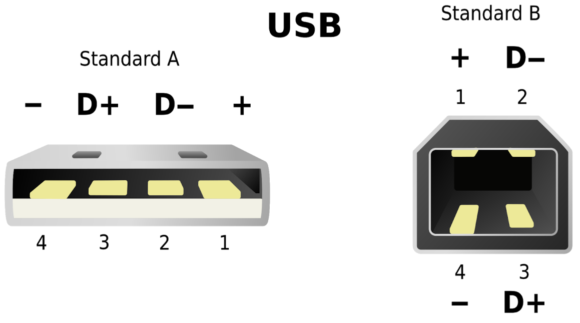
Everything is built into the void loop function and the code for the game is given as an attachment.



```
import processing.serial.*;
Serial p;
String a="";
PFont f;
void setup()
{
  size(displayWidth,displayHeight);
  p=new Serial(this,"COM9",1200);
  p.bufferUntil('\n');
  f=loadFont("ACaslonPro-Bold-150.vlw");
  textFont(f,115);
}
void draw()
{
  background(0,0,0);
  fill(255,255,255);
  textAlign(CENTER);
  text(a,displayWidth/2,displayHeight/2);
}
void serialEvent(Serial p)
{
```

Interfacing:

The micro controller has only TX and RX pins to transmit and receive, but computer uses different convention to read and write data through USB, here comes the issue. So we go for serial to USB converter, this converter has inputs as **TX** and **RX** and outputs data to the **USB**.The USB is one side A type and other side type B.Type A is connected to the PC and other to the converter.



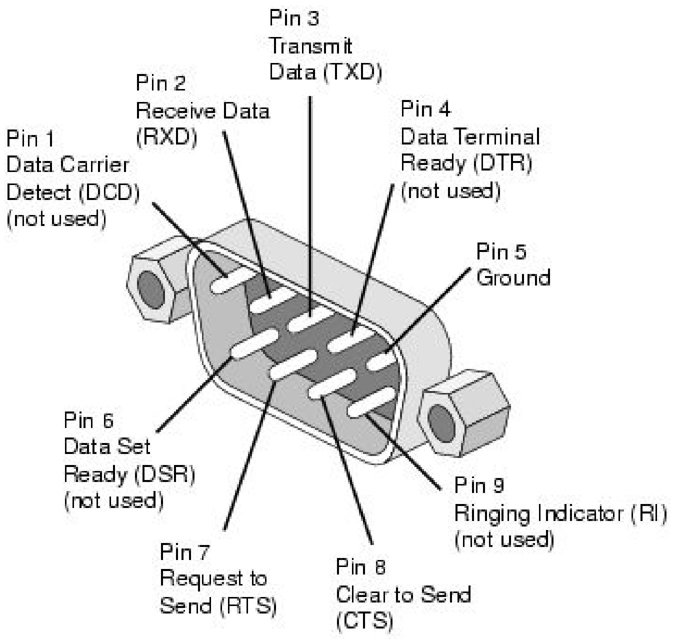
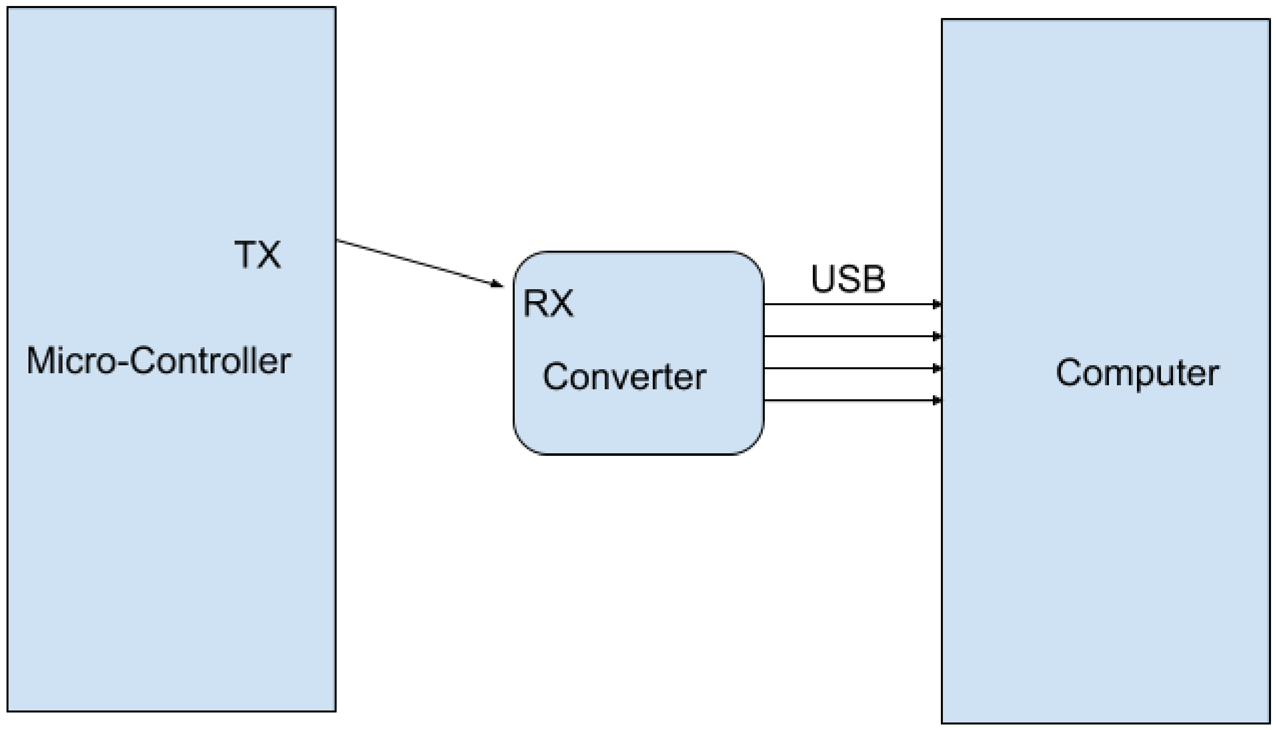


Diagram:



Working:

1.Every port is checked inside the void loop function constantly, if any one of the button is pressed the letters **“U”,”V”,”S”** and **“T”** corresponding to UP1,DOWN1,UP2,DOWN2, where 1 and 2 represents player 1 and player 2.

2.The alphabets is transmitted serially via the TX pin and sent to the PC via USB-Serial converter.

3.The alphabets are checked in the processing environment and the paddles are moved respectively.

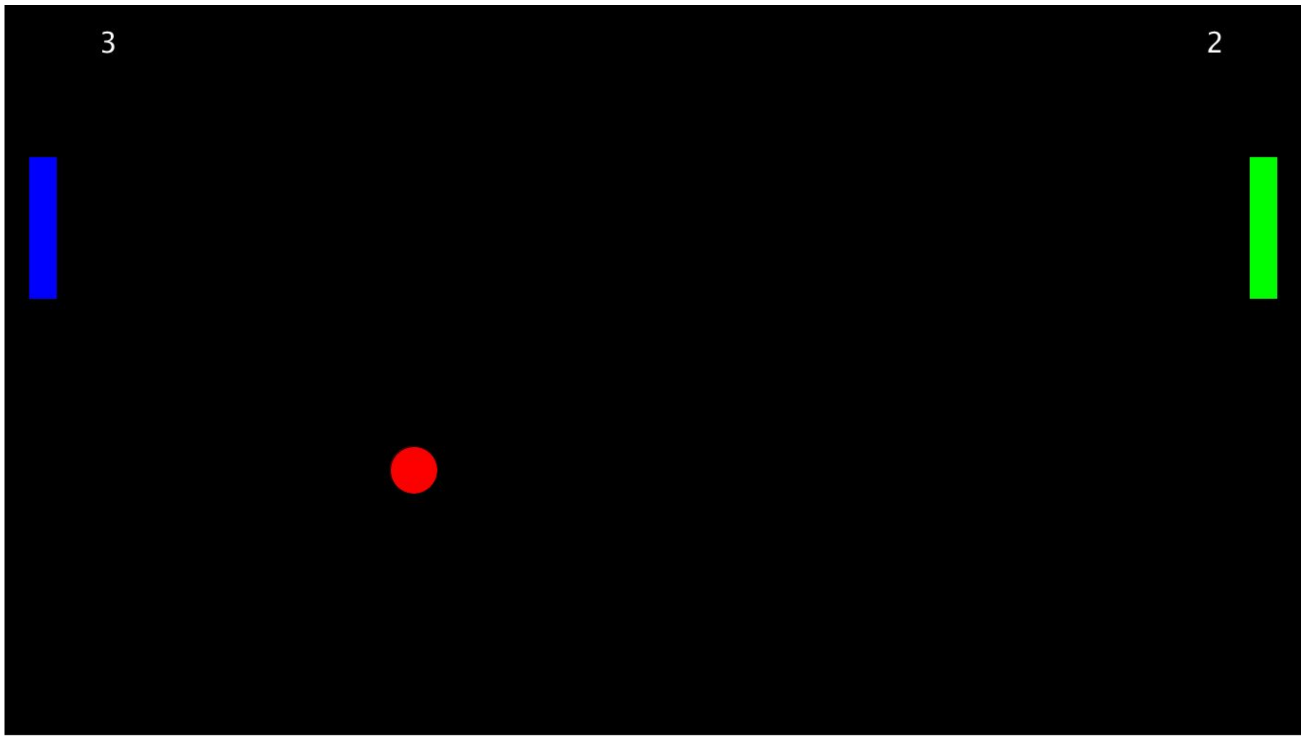
Game Rules:

1.Move the paddles up-down using the buttons in the joy stick.

2.Never leave the ball behind your paddle, if so opponent gets +1 score.

3.Move consistently and tactically to make loose the opponent.

4.And Et Voila, the works. Happy Gaming.



That’s all folks….