

CS-226 Mini Project Report

OVERVIEW

Report contains the working of an arduino ping pong gaming console, things which are required to be taken care of while replicating this project and links to the working code and reference used.

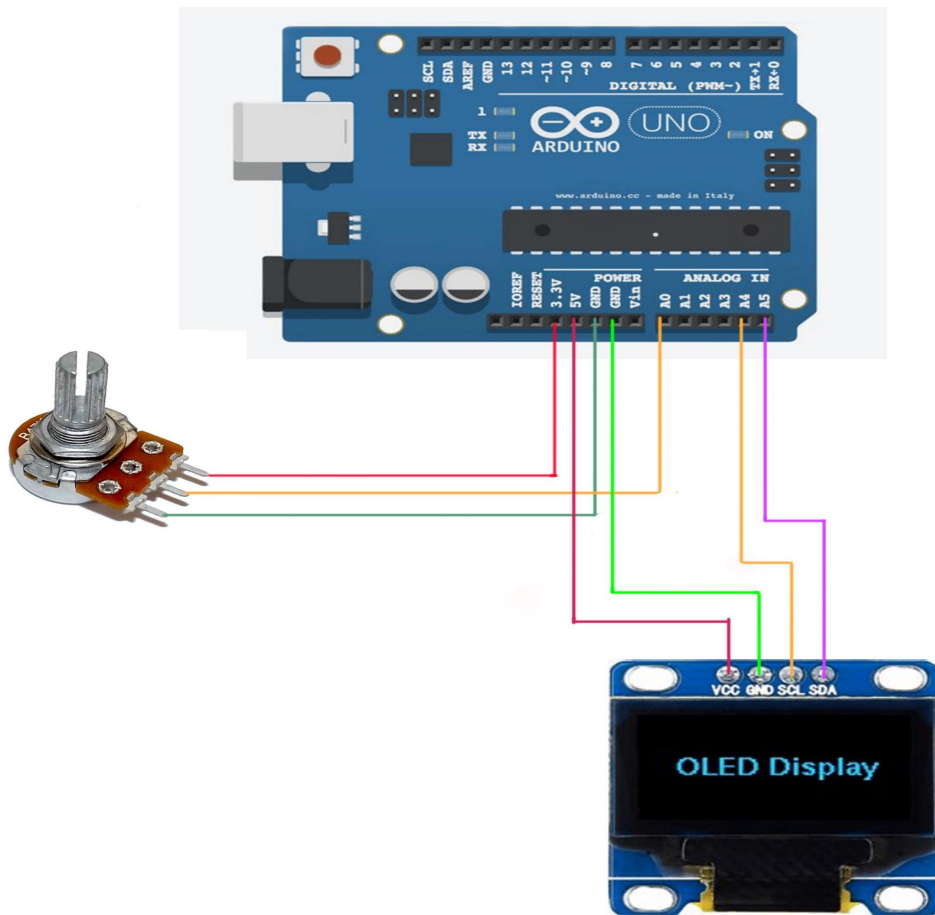
Objective:

To build a fun retro ping pong gaming console .

Things Used:

1. Arduino Uno microcontroller
2. SSD1306 OLED 128x64 I2C display
3. Potentiometer (100 k Ω)
4. Jumper wires

Setup:



Code:

<https://github.com/jenishmonpara/Arduino-ping-pong-gaming-console/blob/main/code.ino>

Code Overview:

SPI library allows us to communicate with SPI devices, with the Arduino as the master device. Wire library allows us to communicate with I2C devices(our display). Adafruit library has some builtin functions to display common shapes and characters on a display.

The variables carry the purpose by intuition of their names. In line 47, `if(time > ball_update)` checks if the last time the ball's position was updated was `BALL_RATE` prior to current time or not and then it updates the ball's position using some comparisons used to detect collisions. In line 90, `if(time > paddle_update)` checks if the last time the human's paddle position was updated was `PADDLE_RATE` prior to current time or not. The CPU's position update was given an additional lag by the amount `cpulag` because it was not possible to score a point by the human player otherwise.

The code checks the ball's current position and accordingly adjusts the CPU's paddle position. Player's paddle position is adjusted on the basis of `POTPIN` reading which is then mapped from 0 - 500 to the range 1 - (63 - `PADDLE_HEIGHT`). The reason I chose 500 instead of 1023 (as expected because of the upper bound of analog signals) is that I was not able to shift my paddle towards the bottom of the screen. This meant that I was not getting the mapped value of (63 - `PADDLE_HEIGHT`) and thus analog reading of 1023. So after some hit and trial, 500 seemed to be working. You may have to use 200 - 600 or 0-1023 or 300-700 depending upon the potentiometer.

Lastly parameters passed in `display.setCursor()`, `display.drawPixel()` and starting and ending limits of some `for` loops inside the functions `check_winner()`, `displayscores()`, `erasescores()` and `drawCourt()` should be determined on the basis of your display dimensions.

Reference:

<https://www.youtube.com/watch?v=YbztpjIDKHE>