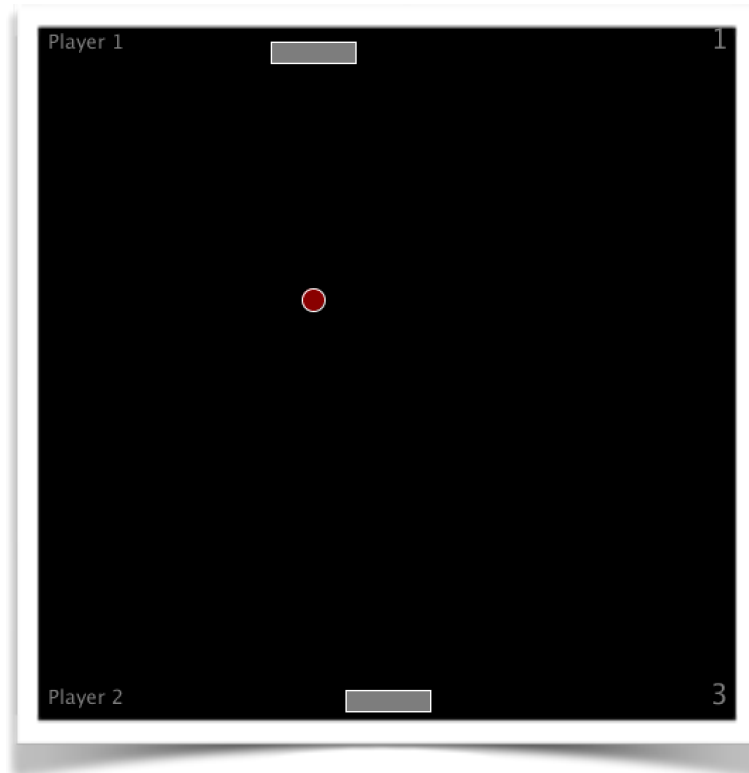


Physical Computing based on Open Software and Hardware Platforms

UPM115

| | |
|--------------------------------------|---|
| 1. Introduction + source code | 2 |
| 2. Rules | 2 |
| 3. Protocol (ARDUINO <-> PROCESSING) | 2 |
| 4. Arduino | 2 |
| 5. Processing | 3 |



1. Introduction + source code

For our project, we have decided to create a new version of well known multiplayer game - pong. User interface is written in processing whilst reading sensors data is handled by Arduino code.

Source code is available at github.com/owlcode/arduino-pong

2. Rules

Two players can steer their paddles at the same time, trying to make the ball fall behind other player's paddle. Game lasts until one of the players collects 5 points. Once that happens we can see a screen confirming who is the winner. Players are using their hands to set the positions of paddles.



3. Protocol (ARDUINO <-> PROCESSING)

1. Movement - LEFT48 / RIGHT78

Each cycle Arduino is printing above information to serial port. It contains information about which player is moving and what is the paddle position. Data gathered from laser sensors.

2. Start game - START

Whenever players are ready to start the game they can push green button. Ball will start to move in random direction.

3. Pause game - PAUSE

Sometimes it gets really tough. You can always press yellow, pause button and make the game just stop. Ball freezes in current position and points are not resetting.

4. Reset game - RESET

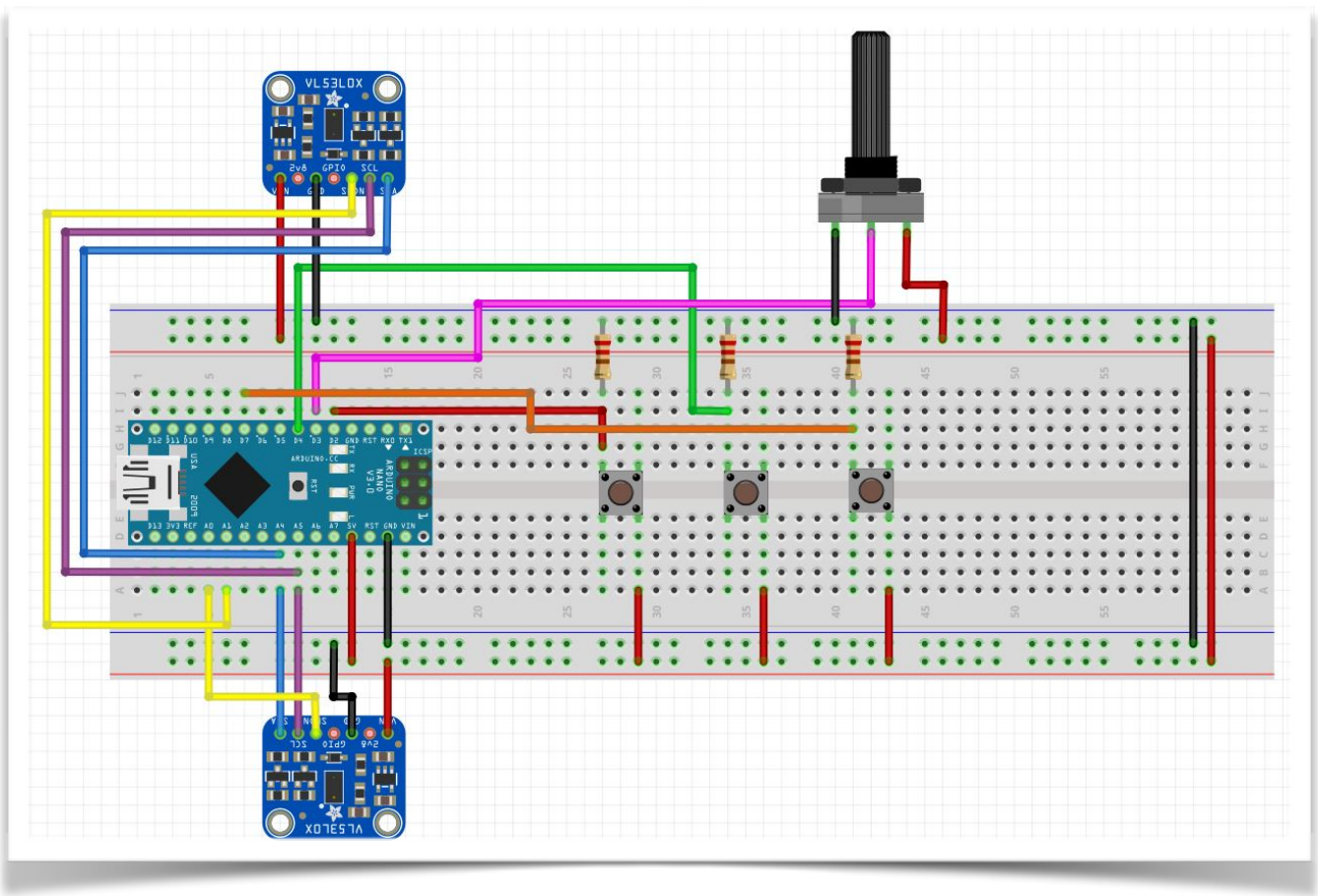
Whenever players want to start game from the beginning and reset points they currently have, they need to press reset button.

5. Ball speed - SPEED4

If the game starts being boring one can change speed of the ball so that it gets more dynamic.

4. Arduino

At first we used ultrasonic sensors HC-SR04. It quickly came out that they are not accurate enough. We were trying to fix it using different programming techniques (e.g. gathering three



readings and taking average value, limiting size of single move that player can take). None of them was good enough so at the end we switched to different sensors.

Laser sensors VL53L0X with its far better accuracy made the game more playable first moment we made them work. Due to I2C communication standard that is used by those sensors we had to create some hacky solutions. Both of them are connected to the same pin and we programmatically changed address for both of them so that we can differentiate which sensors is sending data for us.

5. Processing

Application written in processing is responsible for handling all events sent from Arduino, parsing them and displaying on the screen. It's also capable of holding a multiplayer game without connected Arduino. It's possible to play it using only keyboard.

It calculates collision of particular elements present on board (one ball, two paddles) and also checks if ball is not going out of game board (checks for collision with the wall).