

CAB202 Assignment:

Student Name: Hyeonwoo Lee (Leo)

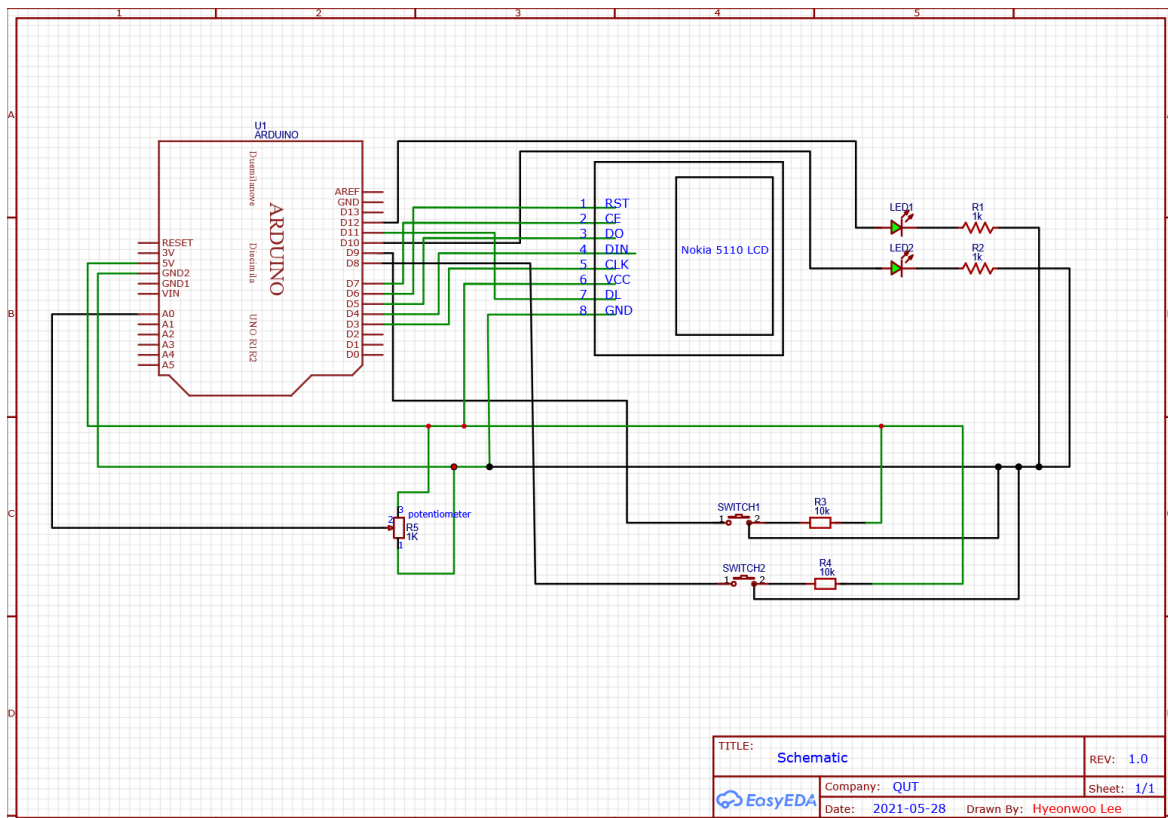
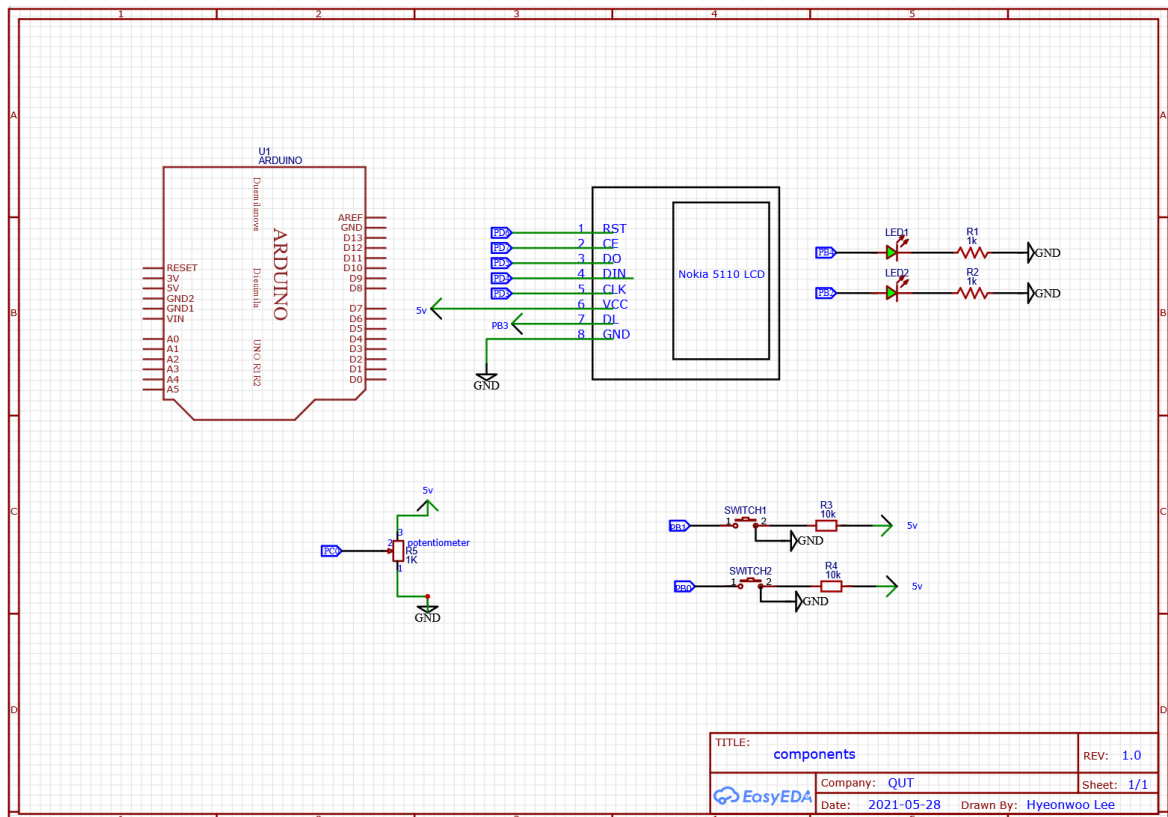
Student Number: n10661531

Introduction

This Arduino controller is designed for gaming purposes. In this device, it provides brick breaker game. If someone wants to play game, they usually sit in front of computer and wait for computer booting. However, sometimes even though we do not have enough time to turn on computer and play a game, we want to play a game. For anyone who wants to spend a bit time without boring, this microcontroller is a good fit. There are also special functions of this device that the normal brick-breaking game does not have, such as LCD brightness control and time limit function. In microcontroller, there are some functions, firstly the LCD shows the game. Secondly, by switch it let users control steps and play the game. Thirdly, there is a potentiometer for handle brightness of lcd. As a fourth function, if ball try to go down, right LED turns on and if it goes up, left LED turns on. Fifth function is putty serial port. If we connect to computer and see serial by putty, it shows the score that user currently playing. The last function is timer function, it limits a time of game, so if user can not finish the game in time, it is game over. Also, the overall game functions are work perfectly.

Digital I/O - Switch	Switch is used for controlling the move box that reflects a ball and it also used to play again or start a game in main page or replay pages. It interacts with lcd's object displaying through programmed code.
Digital I/O – Interrupt-based Debouncing	Even without debouncing, the code could seem to run without a problem, but in fact, it is incomplete when it comes to controlling. This makes it difficult to control the box using switches in the game. So by debouncing, it changed the switch input more stably.
Digital I/O – LED	As mentioned above, while play a game, if direction of ball is going down, right LED turns on and if direction of ball goes up, left led turns on. It is interacting with Arduino through get a direction value of ball on the LCD.
Analog Input – ADC	By potentiometer it gets input value from sensor, and it immediately interact with the brightness of LED which is in LCD. It is available to adjust the brightness of LCD by current value of potentiometer.
Analog Output – PWM	To adjust the brightness of LCD. It is important to handle output data of Arduino. By control with Data, it is available to change brightness which can only be done with PWM. It interacts with potentiometer's input value for brightness adjustment.
Serial I/O – UART	Through connect to computer's putty serial port, it is available to see output that this device shows by UART. It interacts with code that lcd displayed object's action. If ball break a brick, score grows up. If score grows up, it immediately shows current score on serial port.
LCD	LCD is most important part in this program, it interact with switch and potentiometer (ADC) if data comes in, LCD react immediately. Also, lcd drawn object location data required from serial port and led.
Timers (other than debouncing or PWM)	Inside the game, there is a time limit function, which is if times up it goes to game over page. While playing a game, it shows time flow on LCD. If player start new game, the timer goes to zero which means timer is interact with code inside to reset timer.

Schematic



Wiring Instructions

Step 1:

Powering Breadboard

1. Using a wire, place in +5V pin and connect on red positive (+) rail of breadboard
2. Using a wire, place in GND pin and connect to blue negative (-) rail of breadboard

Step 2:

Connecting LED's

1. Using a long wire, place a red LED1 in PB4 (Labelled D12) with the long leg in hole a10 of the breadboard and short leg in hole A9 with a 1k ohm resistor connecting the short leg to GND
2. Using a long wire, place a red LED2 in PB2 (Labelled D12) with the long leg in hole a5 of the breadboard and short leg in hole A4 with a 1k ohm resistor connecting the short leg to GND

Step 3:

Connecting Switch's

1. Using a long wire, place a switch1 in PB1 (Labelled D9) with left two legs in hole A26 of the breadboard with a 10k ohm resistor connecting left two legs to GND and right two legs in hole A24 to connect right two legs to 5v.
2. Using a long wire, place a switch2 in PB0 (Labelled D8) with left two legs in hole A22 of the breadboard with a 10k ohm resistor connecting left two legs to GND and right two legs in hole A20 to connect right two legs to 5v.

Step 4:

Connecting LCD's

0. place LCD on the breadboard to the right of Arduino.

1. RST is on 30f which is first, it connects to PD6 (Labelled D6) with wire in hole i30 of the breadboard.
2. CE is on 29f which is second, it connects to PD7 (Labelled D8) with wire in hole j29 of the breadboard.
3. DO is on 28f which is third, it connects to PD5 (Labelled D3) with wire in hole i28 of the breadboard.
4. DIN is on 27f which is fourth, it connects to PD4 (Labelled D4) with wire in hole j27 of the breadboard.
5. CLK is on 26f which is fifth, it connects to PD3 (Labelled D3) with wire in hole i26 of the breadboard.
7. VCC is on 25f which is sixth, it connects to 5v with wire in hole j25 of the breadboard.
8. DL is on 24f which is seventh, it connects to PB3 (Labelled D11) with wire in hole i24 of the breadboard.
8. GND is on 23f which is eighth, it connects to GND with wire in hole j23 of the breadboard.

Step 5:

Connecting potentiometer's

1. Using a wire, place a potentiometer in PC0 (Labelled D14) with middle leg in hole C34 of the breadboard and right leg in hole A33 connecting to 5v, and left leg in hole A35 connecting to GND

<Additional photo for overall understanding>

