Embedded Lab 10

Report for Electronics Lab 10

Black Jack Card Game

NAME: Bueno Jorge

Nhut Tran

GROUP A

Turku University of Applied Sciences

1. **Devices:**

**\_** Prototyping board.

\_ Multimeter MASTECH MS8217

\_ Resistors & buttons

\_ Arduino

\_ LCD display

1. **Introduction:**

Blackjack is a popular American casino game, now found throughout the world. It is a banking game in which the aim of the player is to achieve a hand whose points total nearer to 21 than the banker's hand, but without exceeding 21. Blackjack is played with an international 52-card deck without jokers.

It's fundamentally a two-player game. The player plays against the dealer.

The players' turn

The player can keep his hand as it is (stand) or take more cards from the deck (hit), one at a time, until either the player judges that the hand is strong enough to go up against the dealer's hand and stands, or until it goes over 21, in which case the player immediately loses (busts).

The dealer's turn

The dealer takes more cards or stands depending on the value of the hand. Contrary to the player, though, the dealer's action is completely dictated by the rules. The dealer must take more cards if the value of the hand is lower than 17, otherwise the dealer will stand.

Cards value

• Cards 2-10 are valued at the face value of the card.

• Face cards such as the King, Queen, and Jack are valued at 10 each.

• The Ace card, however, is a special card and which be valued either at 11 or 1.

1. **Procedure:**
2. **Planning**

According to the requirements of this lab, we will have one player and a dealer – computer.

The player will play first, after each turn the player will decide to continuous or stop by pressing playButton or buttonStop. When the player press playButton, the program will take a random number and add it to an array. The game will be stop when the total value of cards of player higher than 21 or when the player hit buttonStop.

The dealer will take card after player. If the player stop while the total value of dealer lower than 17, the dealer will take cards until 17.

After all, the Arduino will calculate the total value of cards of player and dealer. The results will be compared.

The results of cards and the winner will be display on LCD display.

1. **Codes**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\* Black Jack game**

**\* Names: Jorge**

**\* Nhut Tran**

**\* Playing Black Jack with Arduino**

**\* It still has bugs, particularly for LCD display - sometimes it does not work properly on the first round after plugin**

**\* However, when simulate it on Autodesk Circuit website, it works properly**

**\*** <https://circuits.io/circuits/4390493-lcd-black-jack>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**#include <LiquidCrystal.h>**

**LiquidCrystal lcd(12, 11, 5, 4, 3, 2);**

**int buttonPlay = 0;**

**int buttonStop = 0;**

**int count = 0, number, dealerNumber, totalValue, dealerTotal;**

**int value[5], dealerValue[5];**

**const char\* names[5] = {"ace","K","Q","J","N."};**

**int x = 0;**

**int buttonState, State = LOW;**

**int lastButtonState = LOW; // the previous reading from the input pin**

**unsigned long lastDebounceTime = 0; // the last time the output pin was toggled**

**unsigned long debounceDelay = 100; // the debounce time; increase if the output flickers**

**int calculating();**

**int calculatingDealer();**

**int randomNumber();**

**void line1(int times, int cardNumber, int playerValue);**

**void line2(int times,int dealerCard, int dealerValue);**

**boolean debounceButton(int buttonPin);**

**void Player(int playerNumber, int playerValue);**

**void Dealer(int dealerNumber, int valueDealer);**

**void endGame();**

**void winning();**

**void losing();**

**void runningWelcome();**

**void setup() {**

**// put your setup code here, to run once:**

**Serial.begin(9600);**

**pinMode(6, INPUT);**

**pinMode(7, INPUT);**

**randomSeed(analogRead(A0));**

**lcd.begin(16, 2);**

**}**

**void loop() {**

**//checking button with debounceButtion function**

**while(x < 1)**

**{**

**runningWelcome();**

**x++;**

**}**

**buttonPlay = debounceButton(6);**

**buttonStop = debounceButton(7);**

**if(buttonPlay == true)**

**{**

**if(count < 6 && totalValue <= 21 ) //only allow player to draw card if under 5 cards, totalValue under 21**

**{**

**lcd.clear();**

**delay(200); //delay to avoid the program running so fast**

**number = randomNumber();**

**value[count] = number;**

**dealerNumber = randomNumber();**

**dealerValue[count] = dealerNumber;**

**Player(number, value[count]);**

**Dealer(dealerNumber, dealerValue[count]);**

**totalValue = calculating();**

**dealerTotal = calculatingDealer();**

**count++; //increase the times**

**}**

**if(count > 5 || totalValue > 21 || dealerTotal == 21) //if player cards is overflow || totalValue larger than 21 || dealerTotal = 21 before player**

**{**

**losing(); //print totalValue and "Lose"**

**}**

**else if(dealerTotal > 21) //checking dealer value and player value value after each round**

**{**

**winning();**

**}**

**}**

**if(buttonStop == true) //when the player want to stop and checking the result of player and dealer**

**{**

**while(dealerTotal < 17) //if the final result of dealer is under 17, get more cards untill it is over 17**

**{**

**dealerNumber = randomNumber();**

**dealerValue[count + 1] = dealerNumber;**

**dealerTotal = calculatingDealer();**

**count++;**

**}**

**if(totalValue > dealerTotal && totalValue <= 21 )**

**{**

**winning();**

**}**

**else if(totalValue > 21 || dealerTotal > totalValue)**

**{**

**losing();**

**}**

**}**

**}**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\* Player get cards**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**void Player(int playerNumber, int playerValue)**

**{**

**if(number < 4 ) //if random number is under 4, get their name**

**{**

**switch(number) //print its order and name**

**{**

**case 0: //ace**

**value[count] = 1;**

**line1(count, number, value[count]); //print its value**

**break;**

**case 1: //K**

**value[count] = 10; //set value of card K = 10**

**line1(count, number, value[count]); //print its value**

**break;**

**case 2: //Q**

**value[count] = 10; //set value of card Q = 10**

**line1(count, number, value[count]); //print its value**

**break;**

**case 3: //J**

**value[count] = 10; //set value of card J = 10**

**line1(count, number, value[count]); //print its value**

**break;**

**}**

**}**

**else if(number > 4)**

**{**

**line1(count,4,value[count]); //if random number is over 4, get "N." symbol**

**}**

**}**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\* Dealer get cards**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**void Dealer(int dealerNumber, int valueDealer)**

**{**

**if(dealerNumber < 4) //if random number is under 4, get their name**

**{**

**//print its order and name**

**switch(dealerNumber)**

**{**

**case 0: //ace**

**dealerValue[count] = 1;**

**line2(count, dealerNumber, dealerValue[count]); //print its value**

**break;**

**case 1: //K**

**dealerValue[count] = 10; //set value of card K = 10**

**line2(count, dealerNumber, dealerValue[count]); //print its value**

**break;**

**case 2: //Q**

**value[count] = 10; //set value of card Q = 10**

**dealerValue[count] = 10;**

**line1(count, number, value[count]);**

**line2(count, dealerNumber, dealerValue[count]); //print its value**

**break;**

**case 3: //J**

**dealerValue[count] = 10; //set value of card J = 10**

**line2(count, dealerNumber, dealerValue[count]); //print its value**

**break;**

**}**

**}**

**else if(dealerNumber > 4)**

**{**

**line2(count,4,dealerValue[count]); //print its value**

**}**

**}**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\* "Black Jack" welcome words**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**void runningWelcome()**

**{**

**lcd.setCursor(16, 0);**

**lcd.print("BLACK JACK!");**

**delay(1000);**

**for (int positionCounter = 0; positionCounter < 27; positionCounter++) {**

**// scroll one position left:**

**lcd.scrollDisplayLeft();**

**// wait a bit:**

**delay(150);**

**}**

**lcd.clear();**

**}**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\* Winning function**

**\* Display the final result cards of Player and Dealer**

**\* Player WIN**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**void winning() //print "winning" and show their final result**

**{**

**// Using Serial monitor to check the value**

**/\***

**Serial.print("Player value: ");**

**Serial.println(totalValue);**

**Serial.print("Dealer value: ");**

**Serial.println(dealerTotal);**

**Serial.println("Player WIN");**

**\*/**

**lcd.clear();**

**lcd.setCursor(0,0);**

**lcd.print("Player: ");**

**lcd.setCursor(9,0);**

**lcd.print(totalValue);**

**lcd.setCursor(0,1);**

**lcd.print("Dealer: ");**

**lcd.print(dealerTotal);**

**delay(3000);**

**lcd.clear();**

**lcd.setCursor(2,0);**

**lcd.print("Player WIN!!!");**

**lcd.setCursor(2,1);**

**lcd.print("Dealer LOSE");**

**delay(1500);**

**lcd.clear();**

**for(int x = 0; x < 5; x++)**

**{**

**value[x] = 0;**

**}**

**for(int x = 0; x < 5; x++)**

**{**

**dealerValue[x] = 0;**

**}**

**count = 0;**

**number = 0;**

**totalValue = 0;**

**dealerTotal = 0;**

**delay(500);**

**x = 0;**

**}**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\* Losing function**

**\* Display the final result cards of Player and Dealer**

**\* Player LOSE**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**void losing() // print "lose" and show their final result**

**{**

**// Using Serial monitor to check the value**

**/\***

**Serial.print("Player value: ");**

**Serial.println(totalValue);**

**Serial.print("Dealer value: ");**

**Serial.println(dealerTotal);**

**Serial.println("Player LOSE");**

**\*/**

**lcd.clear();**

**lcd.setCursor(0,0);**

**lcd.print("Player: ");**

**lcd.setCursor(9,0);**

**lcd.print(totalValue);**

**lcd.setCursor(0,1);**

**lcd.print("Dealer: ");**

**lcd.print(dealerTotal);**

**delay(3000);**

**lcd.clear();**

**lcd.setCursor(2,0);**

**lcd.print("Dealer WIN!!!");**

**lcd.setCursor(2,1);**

**lcd.print("Player LOSE");**

**delay(1500);**

**lcd.clear();**

**for(int x = 0; x < 5; x++)**

**{**

**value[x] = 0;**

**}**

**for(int x = 0; x < 5; x++)**

**{**

**dealerValue[x] = 0;**

**}**

**count = 0;**

**number = 0;**

**totalValue = 0;**

**dealerTotal = 0;**

**delay(500);**

**x = 0;**

**}**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\* Display cards of Player**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**void line1(int times, int cardNumber, int playerValue) // print the result to Serial monitor**

**{**

**const char \*cardName = names[cardNumber];**

**// Using Serial monitor to check the value**

**/\***

**Serial.print(times);**

**Serial.print(" Player: ");**

**Serial.print(cardName);**

**Serial.print(" ");**

**Serial.println(playerValue);**

**\*/**

**lcd.setCursor(0,0);**

**lcd.print(times);**

**lcd.print(" Player:");**

**lcd.print(cardName);**

**lcd.setCursor(13,0);**

**lcd.print(playerValue);**

**delay(1500);**

**}**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\* Display cards of Dealer**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**void line2(int times,int dealerCard, int dealerValue)**

**{**

**const char \*dealerCardName = names[dealerCard];**

**// Using Serial monitor to check the value**

**/\***

**Serial.print(times);**

**Serial.print(" Dealer: ");**

**Serial.print(dealerCardName);**

**Serial.print(" ");**

**Serial.println(dealerValue);**

**\*/**

**lcd.setCursor(0,1);**

**lcd.print(times);**

**lcd.print(" Dealer:");**

**lcd.print(dealerCardName);**

**lcd.setCursor(13,1);**

**lcd.print(dealerValue);**

**delay(1500);**

**}**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\* Get a random number by using random function in Arduino**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**int randomNumber()**

**{**

**int randomNumber = random(12);**

**return randomNumber;**

**}**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\* calculating the total card's value of player**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**int calculating()**

**{**

**int finalValue = 0;**

**for(int a = 0; a < 5; a++)**

**{**

**finalValue += value[a];**

**}**

**return finalValue;**

**}**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\* Calculating the total card's values of dealer**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**int calculatingDealer()**

**{**

**int finalDealer = 0;**

**for(int a = 0; a < 5; a++)**

**{**

**finalDealer += dealerValue[a];**

**}**

**return finalDealer;**

**}**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\* Checking button function - only change the current's state when button is pressed over 50ms**

**\* Return the current's state to start getting cards**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**boolean debounceButton(int buttonPin)**

**{**

**boolean current = digitalRead(buttonPin);**

**if (current != lastButtonState) {**

**lastDebounceTime = millis(); }**

**if ((millis() - lastDebounceTime) > debounceDelay) {**

**// if the button state has changed:**

**if (current != buttonState) {**

**buttonState = current;**

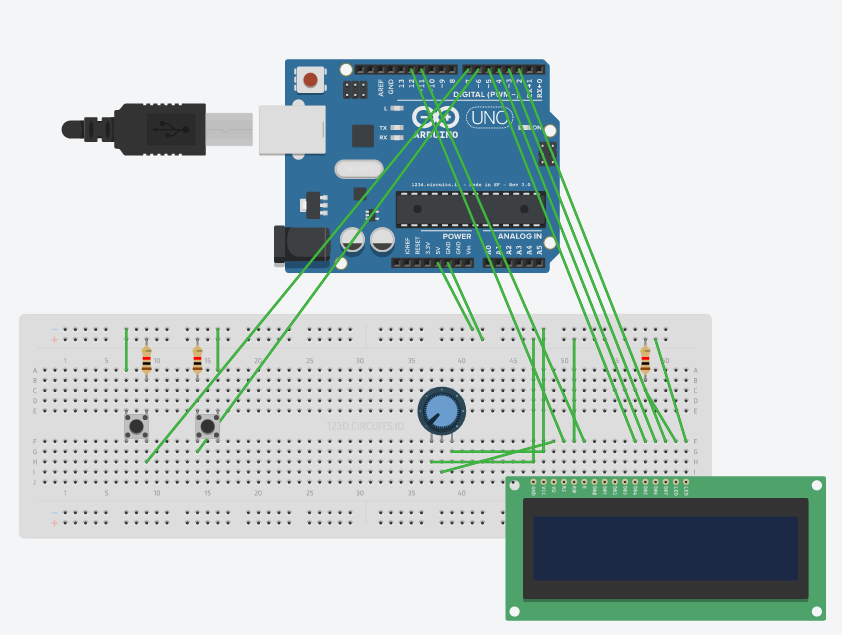
**}**

**}**

**return current;**

**}**

1. **Circuit**

****