Embedded Lab 11

Report for Electronics Lab 11

Morse Transmitter/Receiver

NAME: Bueno Jorge

Nhut Tran

GROUP A

Turku University of Applied Sciences

1. **Devices:**

**\_** Prototyping board.

\_ Multimeter MASTECH MS8217

\_ Resistors

\_ Arduino

\_ Photoresistor

\_ 7-segment LED

1. **Introduction:**

Morse code is a method of transmitting text information as a series of on-off tones, lights, or clicks that can be directly understood by a skilled listener or observer without special equipment. It is named for Samuel F. B. Morse, an inventor of the telegraph.

The transmitter will send the code by using the LED – blinking accordingly international Morse code.

The receiver will receive the code by using the Photoresistor – reading the changing of voltage from Photoresistor.

In this lab, we have used an Arduino to implement for both task and display the codes which have been send from the transmitter to 7-segment LED.

1. **Procedure:**
2. **Algorithms**

According to the requirements of this lab, we will send the codes through the LED and read those by using Photoresistor.

After reading the codes from transmitter, the Arduino will store the codes to array and compare it with the original codes to decryption. Then, it the codes which have been received are correct, the number of codes will be displayed on 7-segment LED

1. **Codes**

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

**\* Lab 11: Morse transmitter/ receiver**

**\* Names: Jorge**

**\* Nhut Tran**

**\* Description: Arduino will transmit a serial signals of a number by LED.**

**\* Using Photoresistor to receive that signal and decrypt those signal to number**

**\* The signal of transmitter could be effect by the light noise when it transmits the signal to Photoresistor**

**\* In this LAB, we have chosen random numbers = 9642**

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

**#define LED\_A1 3**

**#define LED\_B1 4**

**#define LED\_C1 5**

**#define LED\_D1 6**

**#define LED\_E1 7**

**#define LED\_F1 8**

**#define LED\_G1 9**

**#define ledPin 11**

**const byte numbers[] = {B11000000,B11111001,B10100100,B10110000,B10011001,B10010010,B10000010,B11111000,B10000000,B10010000};**

**const int pin1[7] = { LED\_A1,LED\_B1, LED\_C1,LED\_D1,LED\_E1,LED\_F1,LED\_G1 };**

**void displaySevenSeg(int displayCode);**

**byte defaultSignal[4][5]={{1,1,1,1,0},**

**{1,0,0,0,0},**

**{0,0,0,0,1},**

**{0,0,1,1,1}}; //9642**

**byte readingSignal[5];**

**byte decrytionArray[4][5];**

**int readingNumber = 0;**

**int randomNumber();**

**int decrytionNumber = 0, totalNumber, startTime, endTime, totalTime, number;**

**int buttonPlay = 0, x =0;**

**int ledState = LOW; // the current state of the output pin**

**int buttonState; // the current reading from the input pin**

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

**// the following variables are unsigned long's because the time, measured in miliseconds,**

**// will quickly become a bigger number than can be stored in an int.**

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

**unsigned long debounceDelay = 50; // the debounce time; increase if the output flickers**

**void displayCode(int code);**

**void readingCode(int randNumbers);**

**void displayDecrytion(int displayNumer);**

**int decrytionSignal();**

**int randomNumber();**

**int debounceButton(int buttonPin);**

**void setup() {**

**pinMode(ledPin,OUTPUT);**

**pinMode(10,INPUT);**

**for(int i = 0; i<7;i++){**

**pinMode(pin1[i],OUTPUT);**

**}**

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

**Serial.begin(9600);**

**}**

**void loop() {**

**buttonPlay = debounceButton(10); //checking button with debounce function**

**delay(500);**

**if(buttonPlay == 1)**

**{**

**while(x < 1) //allow the program run only one time**

**{**

**number = randomNumber(); //get random number(code) from the random function**

**delay(500); //avoid the program run so fast**

**switch(number) //using this to see which code is sent on Serial monitor**

**{**

**case 0:**

**displayCode(9);**

**break;**

**case 1:**

**displayCode(6);**

**break;**

**case 2:**

**displayCode(4);**

**break;**

**case 3:**

**displayCode(2);**

**break;**

**}**

**readingCode(number); //reading the code from transmitter**

**readingNumber = decrytionSignal(); //decrypte the code what have been recived from transmitter**

**switch(readingNumber) //Display the number(codes) after dcryption**

**{**

**case 0:**

**displayDecrytion(9);**

**displaySevenSeg(numbers[9]);**

**break;**

**case 1:**

**displayDecrytion(6);**

**displaySevenSeg(numbers[6]);**

**break;**

**case 2:**

**displayDecrytion(4);**

**displaySevenSeg(numbers[4]);**

**break;**

**case 3:**

**displayDecrytion(2);**

**displaySevenSeg(numbers[2]);**

**break;**

**}**

**x++;**

**delay(500);**

**}**

**}**

**else**

**{**

**x = 0;**

**}**

**}**

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

**\* Display to 7-segment**

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

**void displaySevenSeg(int displayCode)**

**{**

**for (int x=0;x<7;x++){**

**int value = bitRead(displayCode, x);**

**digitalWrite(pin1[x], value);**

**}**

**}**

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

**\* Display the code which we have sent**

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

**void displayCode(int code)**

**{**

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

**Serial.print(number);**

**Serial.print(" - Number: ");**

**Serial.println(code);**

**}**

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

**\* Display the code which we have received and its decrytion**

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

**void displayDecrytion(int displayNumber)**

**{**

**Serial.print("Decryption Code: ");**

**Serial.print(readingNumber);**

**Serial.print("- Number: ");**

**Serial.println(displayNumber);**

**}**

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

**\* Transmitting the code from transmitter**

**\* Reading the code from Photoresistor**

**\* Insert value of codes to readingSignal[]**

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

**void readingCode(int randNumbers)**

**{**

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

**{**

**delay(500);**

**digitalWrite(ledPin,HIGH);**

**if((analogRead(A1)) < 850)**

**{**

**startTime = millis();**

**}**

**switch(defaultSignal[randNumbers][i])**

**{**

**case 0:**

**delay(500);**

**break;**

**case 1:**

**delay(1000);**

**break;**

**}**

**digitalWrite(ledPin,LOW);**

**//Serial.println(analogRead(A1));**

**if((analogRead(A1)) > 850)**

**{**

**endTime = millis();**

**}**

**delay(500);**

**totalTime = endTime - startTime;**

**if(totalTime > 600)**

**{**

**readingSignal[i] = 1;**

**}**

**else**

**{**

**readingSignal[i] = 0;**

**}**

**}**

**}**

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

**\* Decryte the signals which have been received by compare it with the original codes**

**\* XOR the value of reading code and the value of orginal code**

**\* Getting the value of original number if the value of final results are all zero.**

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

**int decrytionSignal()**

**{**

**int h = 0;**

**for(int j = 0; j < 4; j++)**

**{**

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

**{**

**decrytionArray[j][x] = readingSignal[x] ^ defaultSignal[j][x]; //XOR the value of reading code and the value of orginal code**

**}**

**}**

**for(int j = 0; j < 4; j++)**

**{**

**int totalNumber = 0;**

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

**{**

**totalNumber = totalNumber + decrytionArray[j][i];**

**}**

**if(totalNumber == 0)**

**{**

**decrytionNumber = h;**

**j = 4;**

**break;**

**}**

**h++;**

**}**

**return decrytionNumber;**

**}**

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

**\* Getting random number**

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

**int randomNumber()**

**{**

**int randNumber = random(3);**

**return randNumber;**

**}**

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

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

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

**int debounceButton(int buttonPin)**

**{**

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

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

**lastDebounceTime = millis(); }**

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

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

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

**buttonState = current;**

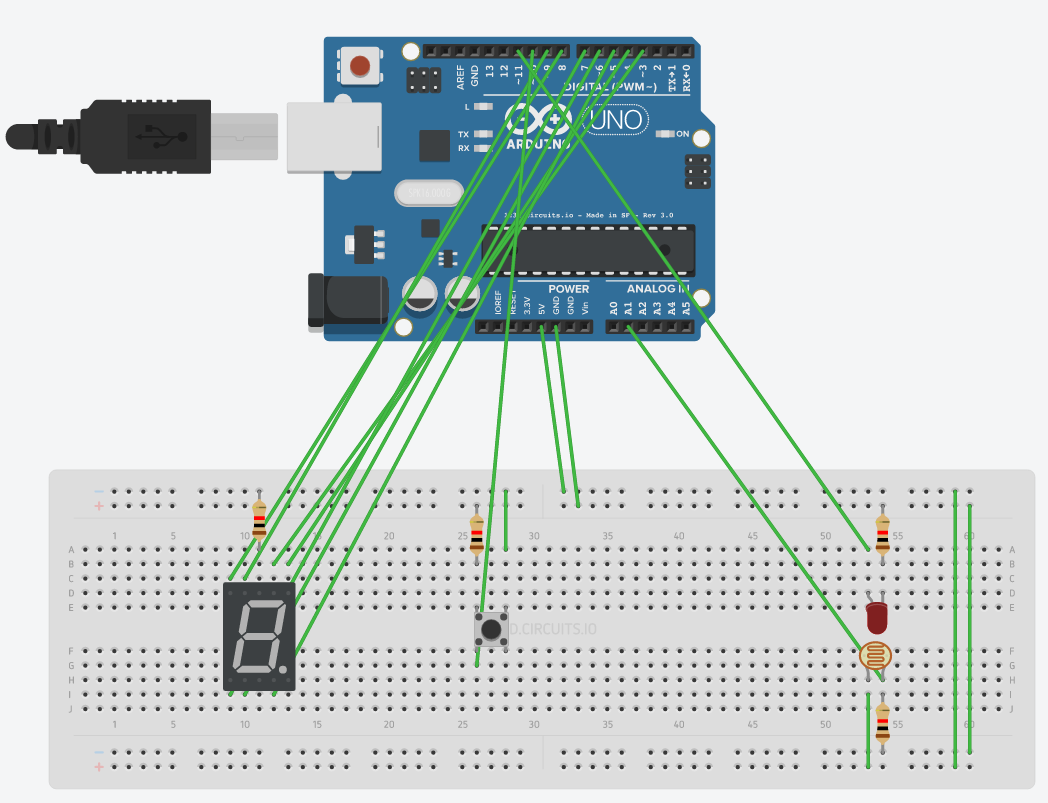
**}**

**}**

**return current;**

**}**

1. **Circuit**

****