**DEPARTMENT OF PHYSICS, UNVERSITY OF COLOMBO**

**EMBEDDED SYSTEMS LABORATORY**

**ELECTRONICS & IT**

**2016**

**TETRIS**

**INDIVIDUAL PROJECT**

**NAME: A.K.Samira Dinendra**

**REG. NO.: 2013/S/13998**

**INDEX NO.: 12119**

**ABSTRACT**

This report is based on the development of game called “Tetris”. The main aim of this project was to build interactive, reliable and user friendly gaming equipment for users to enjoy their leisure time. In realising that goal, a compatible set of hardware devices and software were used.

All the regulations and procedure is programmed according to the standard Tetris game which has been already designed in LCD screen. The main integrated circuit used here is the ATmega32A microcontroller. The power is given by 9 V battery and the display is two 8\*8 LED matrices. The joy-stick is designed separately in order to increase the user’s comfortability when using the equipment. Apart from the physical items software like EAGLE 7.5, Programmers Notepad and Proteus were used when developing the game.

The result of this effort is the successful completion of a user friendly gaming experience with simple but attractive gaming equipment.

**CONTENTS**

[1 INTRODUCTION 4](#_Toc457993519)

[1.1 Context 4](#_Toc457993520)

[1.2 Objectives 4](#_Toc457993521)

[2 METHODology 5](#_Toc457993522)

[2.1 Procedure of Tetris 5](#_Toc457993523)

[2.2 Designing 5](#_Toc457993524)

[2.3 Coding 8](#_Toc457993525)

[3 RESULTS 9](#_Toc457993526)

[4 DISCUSSION 10](#_Toc457993527)

[5 CONCLUSION 11](#_Toc457993528)

[6 ACKNOWLEDGEMENTS 12](#_Toc457993529)

[7 REFERENCES 13](#_Toc457993530)

[8 APPENDIX 14](#_Toc457993531)

[8.1 APPENDIX A – Code 14](#_Toc457993532)

[8.2 APPENDIX B – User interface of the device 17](#_Toc457993533)

**Table of Figures**

[Figure 1‑1- Classical handheld game player 4](file:///D:\UOC\Electronics%20and%20It\sem%201\Microcontrllers%20N%20Embeded%20Syatem\tetris%20project%20report%20.docx#_Toc457993479)

[Figure 2‑1-Schematic of the main PCB 5](file:///D:\UOC\Electronics%20and%20It\sem%201\Microcontrllers%20N%20Embeded%20Syatem\tetris%20project%20report%20.docx#_Toc457993480)

[Figure 2‑2- ATmega32A IC 6](file:///D:\UOC\Electronics%20and%20It\sem%201\Microcontrllers%20N%20Embeded%20Syatem\tetris%20project%20report%20.docx#_Toc457993481)

[Figure 2‑3-Pin configuration of ATmega32A IC 6](file:///D:\UOC\Electronics%20and%20It\sem%201\Microcontrllers%20N%20Embeded%20Syatem\tetris%20project%20report%20.docx#_Toc457993482)

[Figure 2‑4-8\*8 LED display 7](file:///D:\UOC\Electronics%20and%20It\sem%201\Microcontrllers%20N%20Embeded%20Syatem\tetris%20project%20report%20.docx#_Toc457993483)

[Figure 2‑5-Pin configuration 7](file:///D:\UOC\Electronics%20and%20It\sem%201\Microcontrllers%20N%20Embeded%20Syatem\tetris%20project%20report%20.docx#_Toc457993484)

[Figure 2‑6- Schematic of the Joy-Stick 7](file:///D:\UOC\Electronics%20and%20It\sem%201\Microcontrllers%20N%20Embeded%20Syatem\tetris%20project%20report%20.docx#_Toc457993485)

[Figure 2‑7- Main PCB process of soldering components 8](file:///D:\UOC\Electronics%20and%20It\sem%201\Microcontrllers%20N%20Embeded%20Syatem\tetris%20project%20report%20.docx#_Toc457993486)

[Figure 2‑8- Joy-Stick 8](file:///D:\UOC\Electronics%20and%20It\sem%201\Microcontrllers%20N%20Embeded%20Syatem\tetris%20project%20report%20.docx#_Toc457993487)

[Figure 2‑9-Initial coding 8](file:///D:\UOC\Electronics%20and%20It\sem%201\Microcontrllers%20N%20Embeded%20Syatem\tetris%20project%20report%20.docx#_Toc457993488)

[Figure 8‑1- Final product 17](#_Toc457993489)

# INTRODUCTION

## Context

There are many ways of spending leisure in modern society. Out of them playing games made a prominent place out of them due to development of the electronic world. So it is a responsibility of the developer to make an instrument which satisfies the user needs of the leisure and as well as if it is able to improve and develop user’s physical or mental skills, the gaming equipment will be popular among the users. So Tetris is a good example of such games.

Basically, the game Tetris is a game which provides enjoy and as well as improve the mental power of the user who plays the game. There are many methods of playing this game in modern society. The most popular type is the classical handheld game player with LCD display. There are some issues of this equipment and the aim of this project is to identify them and dismiss them in the final product.

Figure 1‑1- Classical handheld game player

Any instrument should be user friendly in order to increase the user response. In the classical game player, basically the display does not tally with the modern electronic items as it is a colour less, small, odd looking one. So, the requirement of a replacement of more attractive display is emerged. And requirement of a more comfortable joy-stick also emerged with compared with the classical one. So they become the main objectives when designing the project.

## Objectives

There were many objectives of implementing this game.

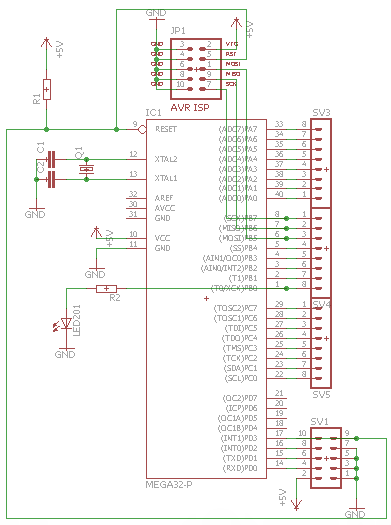
1. Improving the ability of taking decisions quickly of an individual.
2. Improving the thinking pattern.
3. More attractive display than classical gameplay.
4. More comfortable joy-stick.
5. Reducing the heaviness to the hand of the player.
6. Giving a smart look.

# METHODology

## Procedure of Tetris

Tetris is an electronic game let players arrange puzzle pieces in real time as they fell from the stop of the playing field. The resulting design was a game that used four distinctive geometric playing pieces, each made up of four LEDs in the 8\*8 LED matrix. The patterns can be move left, right and can rotate in clockwise. When a row is completely lit with LEDs, then it is dismissed by adding a score. Then the rest will drop down.

## Designing

The design was implemented and simulated through Proteus before designing the schematic. The main controlling IC used here is ATmega32A microcontroller. All the instructions of the game are coded in to this main IC. This is a 40 pin, low-power CMOS 8-bit microcontroller IC with 32 general purpose working registers. All the I/O pins of PORT A, B and C are used to connect to the two LED displays. The eight grounds of each LED display are connected to each other and connected to PORT B. All the 16 anodes are connected to PORT A and C individually. The movable joy-stick was planned to connect to PORT D.

Programming ISP header was also introduced to the main circuit as it can program anytime wanted and it is easy to program at the initial state also. Also external 16 MHz oscillator was also used. The basic design of the schematic was done with EAGLE 7.5.

Figure 2‑1-Schematic of the main PCB

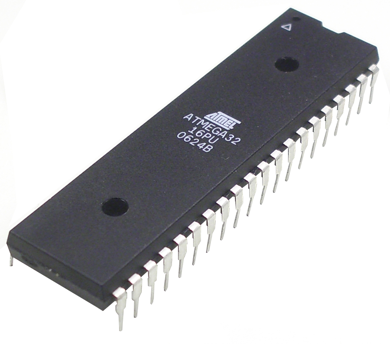


Figure 2‑2- ATmega32A IC

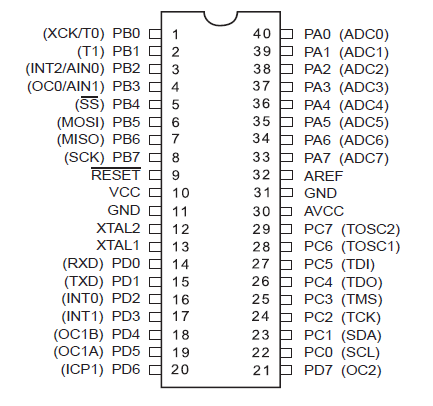


Figure 2‑3-Pin configuration of ATmega32A IC

## ATmega32A Microcontroller Features

* High-performance, Low-power AVR 8-bit Microcontroller
* Advanced RISC Architecture
  + 131 Instructions - Most Single Clock Cycle Execution
  + 32 x 8 General Purpose Working Registers
  + Up to 16 MIPS Throughput at 16MHz
* Non-Volatile Program and Data Memories
  + 32k Bytes of In-System Self-Programmable Flash
  + 1024 Bytes EEPROM
  + 2K Bytes Internal SRAM
* JTAG Interface
  + Boundary-scan Capabilities According to the JTAG Standard
  + Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
* Peripheral Features
  + On-chip Analog Comparator
  + Programmable Watchdog Timer with Separate On-chip Oscillator
  + Master/Slave SPI Serial Interface
  + Two 8-bit Timer/Counters with Separate Pre-scalar, Compare
  + One 16-bit Timer/Counter with Separate Pre-scalar, Compare and Capture mode
  + Real Time Counter with Separate Oscillator
  + Four PWM Channels
  + 8-channel, 10-bit ADC
* Special Microcontroller Features
  + Power-on Reset and Programmable Brown-out Detection
  + Internal Calibrated RC Oscillator
  + External and Internal Interrupt Sources
  + Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
* I/O and Packages
  + 32 Programmable I/O Lines
  + 40-pin PDIP, 44-lead TQFP, and 44-pad MLF
* Operating Voltages
  + 4.5-5.5V for ATmega32
* Speed Grades
  + 0-16MHz for ATmega32
* Power Consumption at 4 MHz, 3V, 35 °C
  + Active: 1.1mA
  + Idle Mode: 0.35mA
  + Power-down Mode: < 1µA

The LED matrices are red colour 8\*8 which has 16 pins, 8 for cathode and other 8for anode separately. They are not properly arranged in the matrix. Before connecting to the microcontroller, each LED should be identified with their anodes and cathodes. And also there are two types which are called row anode method and row cathode method. The matrix here used was row anode one. The forward voltage of an LED was 2 V. the output voltage of the microcontroller was 5 V. so 330 Ω resistors were used in series with each LED to reduce the voltage felt by the anode of the LED.



Figure 2‑4-8\*8 LED display

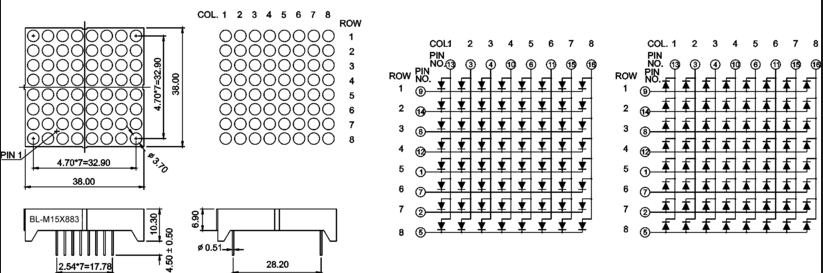


Figure 2‑5-Pin configuration

In the joy-stick pull down resisters and capacitors were used to reduce the bouncing effect. It consists of 5 micro switches for handling the objects. All the components are connected to a header as in the main PCB and the two headers are connected each other by a wire. This feature is introduced to this particular project in order to make the user more comfortable.

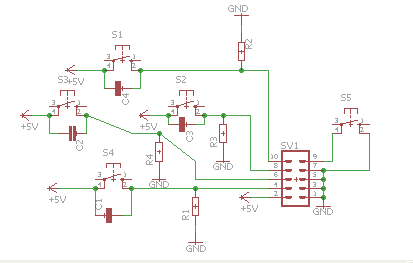


Figure 2‑6- Schematic of the Joy-Stick

After designing the two schematics, the PCB printing layout was created. It was printed and the PCBs were constructed.

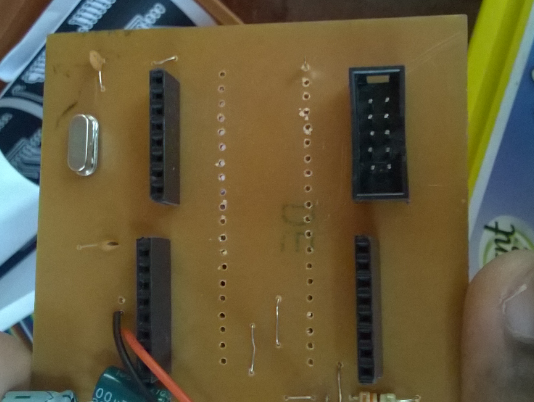




Figure 2‑7- Main PCB process of soldering components

Figure 2‑8- Joy-Stick

## Coding

The coding was initially started with the demonstration board. It was easy with the demonstration board as it provides the four I/O ports separately. The full code of the game is in the appendix.

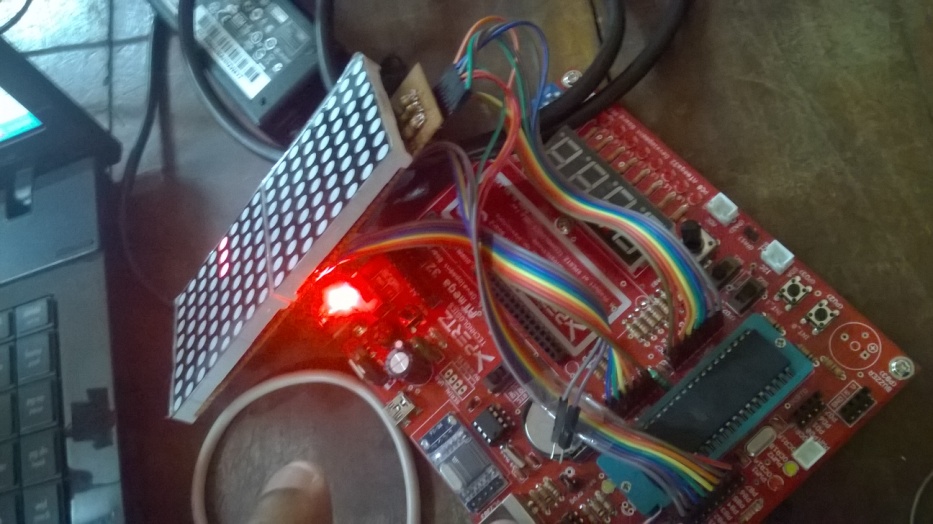


Figure 2‑9-Initial coding

# RESULTS

The outcome of this individual project was a hand held game player operating with a 9 V battery which was able to replace any time. The device is performing up to the theoretical standard that was expected. There is a small issue when dropping down the objects which is the button sometimes doesn’t work as a pause button. Other than that all the other functions are working properly.

Also the device made was very user friendly as it can be use comfortably. The interface was very attractive with the LED bulbs than the classical LCD display. So, the user attraction towards the device increases.

Also, the user can very easily replace the battery if the battery is drained. He only would have to remove the four screws that hold the project box and just replace the old battery with a new battery.

Although it is somewhat expensive than classical game play, it will be able to compete with classical one as the new device holds many attractive features that the old one.

# DISCUSSION

Tetris is a worldwide game which anyone can play without any age restrictions. The main objective of this project is to produce a more attractive and user-friendly gameplay. Basically, the game Tetris is a game which provides enjoy and as well as improve the mental power of the user who plays the game. The overall project was a very successful one as the final product was with its proper working conditions. The major objectives of the attempt were achieved. In this project the physical arrangements were few, but they were smooth to handle. The full project is fertilized with the code as it is the major part of this project. According to the theories followed under embedded system laboratory and knowledge gained from researching the related things this game play was constructed.

There were some more modifications to add which were hoping to develop. One is implementing a seven segment display to display the score after the game is over. It can be implemented in the LED matrix too by developing the code. Also there is a hope to develop an Android application as the joy-stick for increase the user friendliness. Using that player can play the game with his Android phone without using the physical joy-stick provided. Also it was a suggestion by the first test viewers to introduce RGB LED display to the main display in order to change the colour of the completed rows. Most of the things what is going to be developing basically depend on the software base things which can easily be modified by changing the code and write it in the main ATmega32A microcontroller chip. Programming ISP header is also implemented with the main circuit board to make this opportunity easier. So this product is developed which is compatible for any further modifications. That is a very good advantage of this product.

The product is developed under research level and anyone who has concerned about microcontroller programming can add more modifications. This will be a good product than the classical Tetris gameplay.

# CONCLUSION

It can be concluded that this project was a success as the device was working as expected. A very good knowledge on microcontroller programming was obtained. Another advantage was gaining a good knowledge electronic components and the way how to present any device in order to catch the eye of the user or customer. The knowledge gained through this project will be much helpful in my future studies as a Physics student.

# ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to all those who helped even with a single word in the completion of this report. Specially, I would like thank the lecturer in charge of the Embedded Systems Laboratory, Hiran Jayaweera, and the demonstrators Mr.Rasika Manjujeewa and Mr. Deshitha for their greatest support and guidance. I would also extend special gratitude to Thilina Roshan who was very much helpful in making this project a success. Furthermore, I would like to thank all the academic and non-academic staff in the Computational Lab and the Physics department.

# REFERENCES

* <http://electronics.stackexchange.com/questions/236663/how-to-get-an-8x8-led-matrix-pin-out>
* <http://arduino-er.blogspot.com/2015/01/how-to-identify-pin-1-of-8x8-led-matrix.html>
* <http://circuitdigest.com/microcontroller-projects/8x8-led-matrix-with-avr-atmega8>
* <http://dhassel.de/projects/cool/led-matrix-display/>

# APPENDIX

## APPENDIX A – Code

#include <avr/io.h>

#include <util/delay.h>

#define plusA PORTC

#define plusB PORTA

#define ground PORTB

#define disp\_delay 1

#define btn\_left (1<<PD1)

#define btn\_right (1<<PD2)

#define btn\_rot (1<<PD3)

#define btn\_down (1<<PD0)

unsigned char blocks[36]={ 0b11000000,

0b11000000,

0b00000000,

0b11000000,

0b11000000,

0b00000000,

0b11000000,

0b11000000,

0b00000000,

0b11000000,

0b11000000,

0b00000000,

0b11100000,

0b00000000,

0b00000000,

0b10000000,

0b10000000,

0b10000000,

0b11100000,

0b00000000,

0b00000000,

0b10000000,

0b10000000,

0b10000000,

0b11100000,

0b01000000,

0b00000000,

0b10000000,

0b11000000,

0b10000000,

0b01000000,

0b11100000,

0b00000000,

0b00100000,

0b01100000,

0b00100000};

unsigned char area[17]={0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xff};

unsigned int speed = 30;

unsigned int flag\_shape=1;

unsigned int flag\_rot=1;

unsigned int flag\_push=0;

unsigned int flag\_btn=0;

void display (unsigned int h);

void button (void);

void area\_check(void);

void clear\_line(void);

void change\_block(void);

unsigned int stop\_block(unsigned int h);

void game\_over(void);

int main (void){

DDRD&=~(btn\_left|btn\_right|btn\_down|btn\_rot);

DDRA=0xff;

DDRB=0xff;

DDRC=0xff;

while(1){

next:

game\_over();

change\_block();

clear\_line();

speed=30;

for(unsigned int height=0;height!=16;height++){

for(unsigned int ref=0;ref!=speed;ref++){

display(height);

}

if(stop\_block(height)){

goto next;

}

}

}

}

void display (unsigned int h){

for(unsigned int i=0;i!=16;i++){

for(unsigned int r=0;r!=3;r++){

if(i==h+r){

ground =~(blocks[12\*(flag\_shape)+3\*(flag\_rot)+r]>>flag\_push);

}

}

ground &=~(area[i]);

if(i/8==0){

plusB=0x00;

plusA=(1<<i);

}

else{

plusA=0x00;

plusB=(1<<i%8);

}

button();

\_delay\_ms(disp\_delay);

ground=0xff;

}

//ground=0xff;

}

void button (void){

if((PIND&btn\_left)){

if(flag\_btn==0){

if(flag\_push!=5)

flag\_push=flag\_push+1;

flag\_btn=1;

}

}

else if((PIND&btn\_right)){

if(flag\_btn==0){

if(flag\_push!=0)

flag\_push=flag\_push-1;

flag\_btn=1;

}

}

else if((PIND&btn\_down)){

if(flag\_btn==0){

speed=10;

flag\_btn=1;

}

}

else if((PIND&btn\_rot)){

if(flag\_btn==0){

flag\_rot=(flag\_rot+1)%4;

flag\_btn=1;

}

}

else{

flag\_btn=0;

}

}

void clear\_line(void){

for(unsigned int l=0;l!=16;l++){

if(area[l]==0xff){

while(l){

area[l]=area[l-1];

l--;

}

}

}

}

void change\_block(void){

flag\_shape=((flag\_shape+1)\*23+4)%3;

flag\_rot=((flag\_rot+1)\*17+5)%4;

}

unsigned int stop\_block(unsigned int h){

unsigned int skip=0;

for(unsigned int k=3;k!=0;k--){

if(area[h+1]&(blocks[12\*(flag\_shape)+3\*(flag\_rot)+k-1]>>flag\_push)){

for(unsigned int a=0;a!=3;a++){

area[h-a]|=(blocks[12\*(flag\_shape)+3\*(flag\_rot)+a]>>flag\_push);

}

skip=1;

}

}

return skip;

}

void game\_over(void){

if(area[0]!=0x00){

for(unsigned int j=0;j!=16;j++){

area[j]=0x00;

}

}

}

## APPENDIX B - User interface of the device

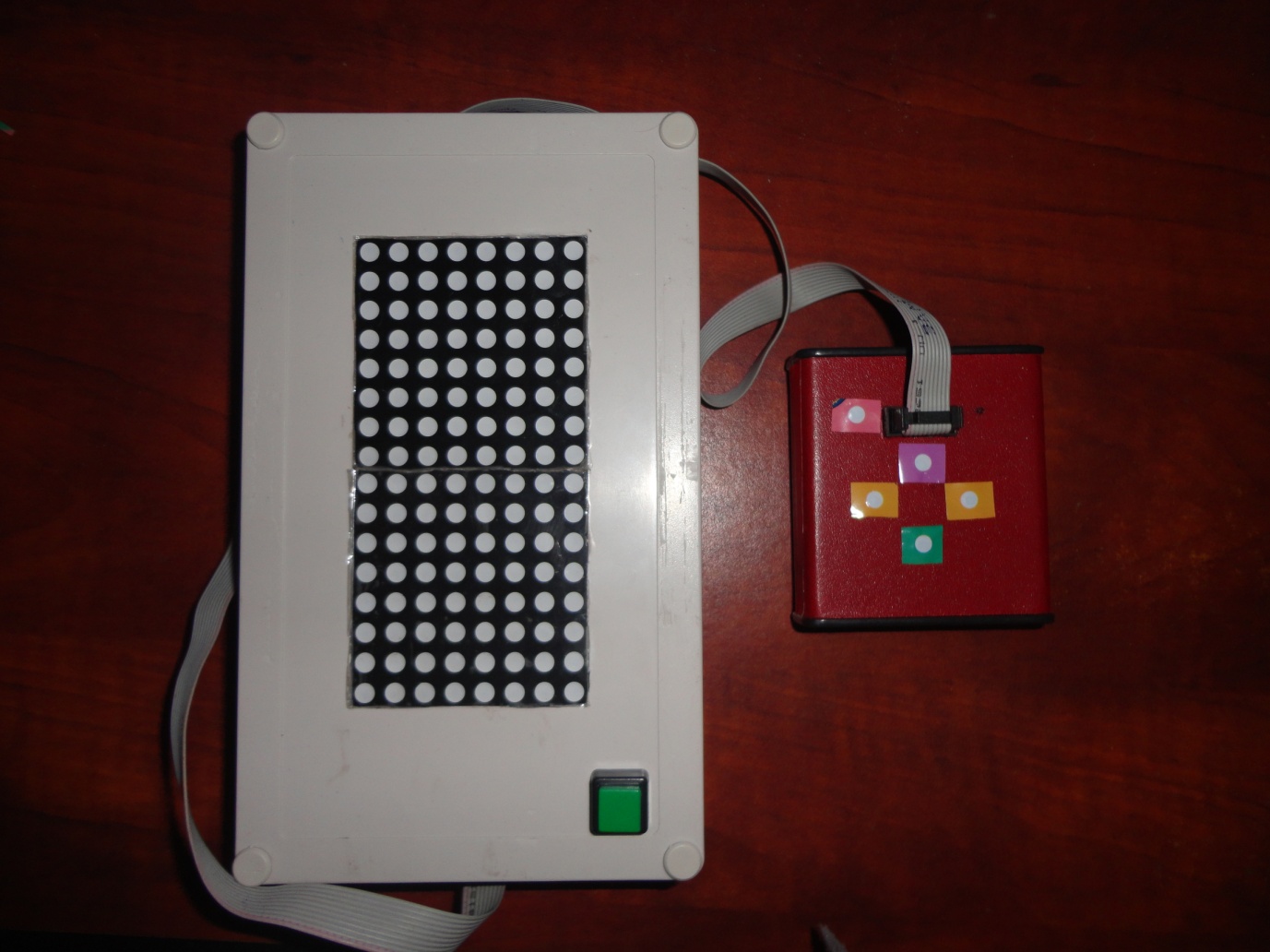


Figure 8‑1- Final product