

Presented By:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Name** | **Contact #** | **D1** | **D2** | **D3** | **D4** |
| 19105633 | MOHAMED MAHMOUD ZEIN |  |  |  |  |  |
| 19106978 | MOHAMED ABDALLA |  |  |  |  |  |
| 18106717 | OMAR MAGED |  |  |  |  |  |

[ PROJECT TITLE]

Department : Computer Engineering

Course Code : CC421

Course Name : Microprocessor Systems

Lecturer : Prof. Dr Ahmed Fahmy

Teaching Asst. : Eng. Muhammad Sami

# Summary:

Secure Door Lock System.

The door opens when you enter your PIN code so that only you and the people with that PIN code can open the door and bypass through.

This gives more security than the normal door lock with physical key.

# Individual Tasks/ Responsibilities

|  |  |
| --- | --- |
| **Name** | Mohamed Abdalla Basim |
| **ID** | 19106978 |
| **Tasks** | * Hardware * Software |

|  |  |
| --- | --- |
| **Name** | Mohamed Mahmoud Zein |
| **ID** | 19105633 |
| **Tasks** | * Software |

|  |  |
| --- | --- |
| **Name** | Omar Maged |
| **ID** | 18106717 |
| **Tasks** | * Hardware |

# Introduction

The Door should be only opened when correct password is entered.

It should be open for just a short span of time to let people pass and automatically closes after that.

When a wrong password is entered continuously for 3 times then the LCD should tell about intruder alert and buzz will turns on.

Most of used main components are new to the team members, only the AT89S56 microprocessor and its crystal was familiar to the member, never used the keypad nor push button that’s why they are in the reference page

It’s our first time flashing with Arduino so that the pinout for that is also added to the reference page.

# Methodology

The project works as the following:

When the power is connected it start with locking the door at the beginning and the LCD lights up and tells you to enter the PIN code, and then it scans it and store it in the memory, then compared it with the stored one.

If they are the same the GREEN LED lights up, the LCD prints “SUCCESS” and the motor used as a door lock opens and keeps opened for a while (15sec) and then closes again and the green led lights down again and the scanned PIN be deleted from the memory and the systems returns to the initial state again.

If the PIN didn’t match the stored one a RED LED lights up and the LCD prints “FAILURE” and the lock keeps closed

You have up to three attempts if they all wrong the BUZZER keeps buzzing for a while and the LCD prints “INTRUDER” and then returns back to the initial state,

There is a push button added to the circuit acts as a reset button when you press it you go to the start of the system.

So, to conclude,

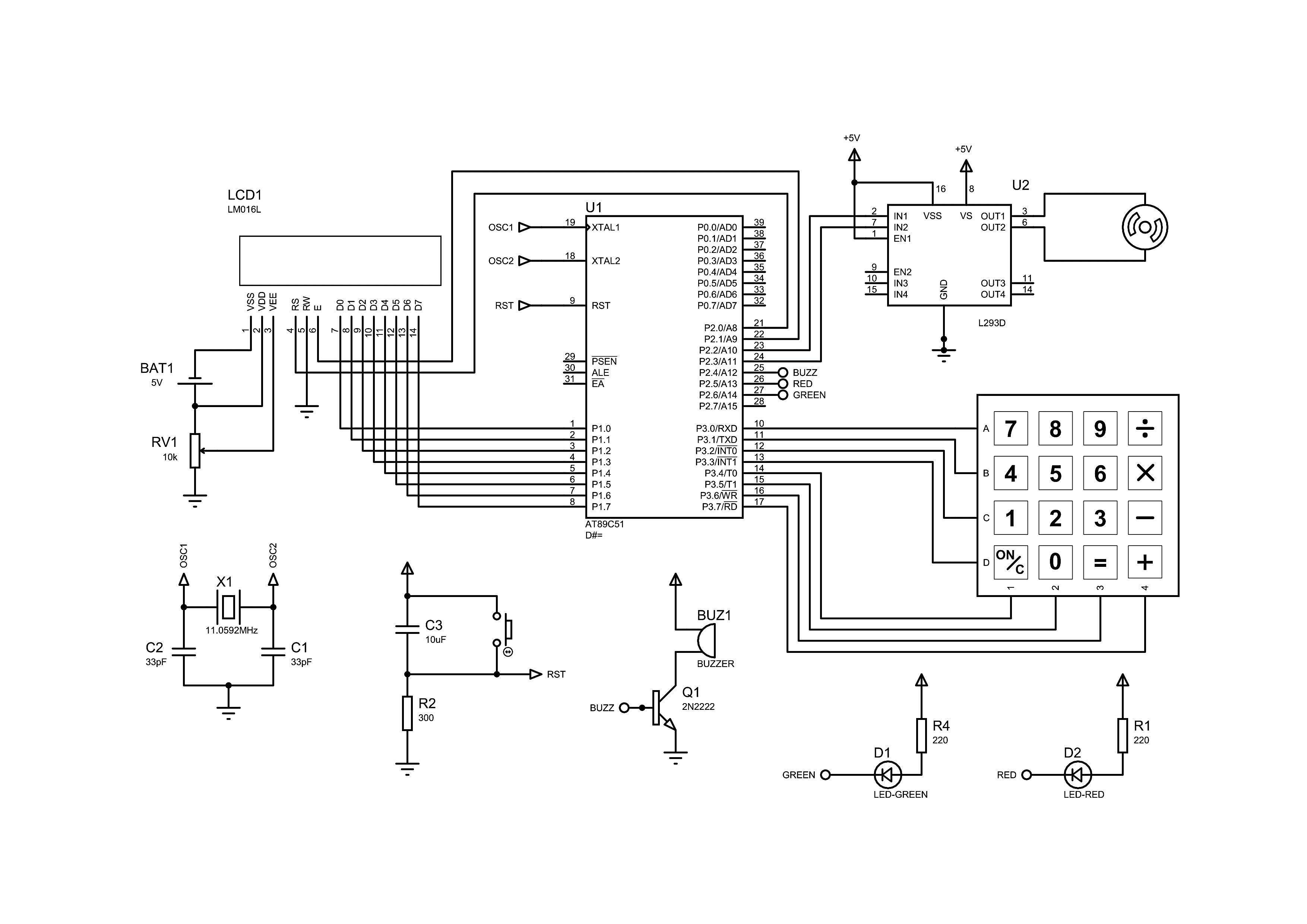
The input is the PIN code and the output is the lock (open/close)

The major component used is the keypad which reads/enters the PIN code and the motor acting as the lock

A driver is used to control the motor cycles and direction weather it opens or closes

The LCD is used to print out a guide words to the user

# Schematic:



# Components and Tools:

LIST OF ALL THE COMPONENTS USED

* Bread board
* 16 X 2 LCD
* 4 X 4 Keypad
* Stepper Motor
* Motor Driver
* Buzzer
* Variable resistance (10K Ohm)
* Crystal (12 MHZ)
* Push Buttons
* Transistor (2N2222)
* Capacitors (10uf -33pf)
* 2xResistances (220 Ohm – 330 Ohm)
* LED (Red - Green)
* Wires & Jumpers
* Arduino (UNO)

# Code Illustration:

Take 4-digit PIN

from user

If PIN match

Turn stepper motor to open door

NO

YES

If number of attempts =0

Start buzzer for 10 times

If user press reset

RESET

Turn stepper motor to close door

Wait 15 second

Check PIN with the

correct one

If does not match

# Code Text:

; MAIN PROJECT

MAIN:

MOV A,#38H

ACALL CMD

MOV A,#0EH

ACALL CMD

MOV A,#01H

ACALL CMD

MOV A,#080H

ACALL CMD

ACALL DEL

ACALL PRT

ACALL KEY\_READ

;END MAIN

PRT: MOV R4,#11

MOV DPTR,#STRING1

S1: CLR A

MOVC A,@A+DPTR

ACALL DAT

INC DPTR

DJNZ R4,S1

MOV A,#0C0H

ACALL CMD

MOV R4,#4

SJMP KEY\_READ

RET

KEY\_READ:

ACALL DEL

JNB P3.4,C1

JNB P3.5,C2

JNB P3.6,C3

JNB P3.7,C4S

CJNE R4,#0,KEY\_READ

JMP BEFORE\_CHECK

C4S:JMP C4

RS BIT P2.0

E BIT P2.1

BUZZ BIT P2.4

RLED BIT P2.5

GLED BIT P2.6

ORG 0000H

MOV R7,#3

CLR P2.2

ACALL W2

CPL P2.2

AGAIN:

MOV P3,#0FFH

CLR P3.0

CLR P3.1

CLR P3.2

CLR P3.3

CLR BUZZ

ACALL W2

SETB BUZZ

SETB GLED

SETB RLED

MOV R0,#02FH

;Input Password is Stored at 030H in memory

MOV R1,#03FH

;original Password is Stored at 040H in memory

MOV R4,#4

MOV DPTR,#PASS

STORED\_PASS:

CLR A

MOVC A,@A+DPTR

INC R1

MOV @R1,A

INC DPTR

DJNZ R4,STORED\_PASS

C3: DEC R4

INC R0

SETB P3.0

JB P3.6,NINE

SETB P3.1

JB P3.6,SIX

SETB P3.2

JB P3.6,THREE

SETB P3.3

JB P3.6,HASH

EIGHT: MOV A,#'8'

ACALL DAT1

ACALL DEL

MOV @R0,#'8'

JMP KEYPAD

FIVE: MOV A,#'5'

ACALL DAT1

ACALL DEL

MOV @R0,#'5'

SJMP KEYPAD

TWO: MOV A,#'2'

ACALL DAT1

ACALL DEL

MOV @R0,#'2'

SJMP KEYPAD

ZERO: MOV A,#'0'

ACALL DAT1

ACALL DEL

MOV @R0,#'0'

SJMP KEYPAD

C4: DEC R4

INC R0

SETB P3.0

JB P3.7,AA

SETB P3.1

JB P3.7,BB

SETB P3.2

JB P3.7,CC

SETB P3.3

JB P3.7,DD

KEY\_SCAN:

C1: DEC R4

INC R0

SETB P3.0

JB P3.4,SEVEN

SETB P3.1

JB P3.4,FOUR

SETB P3.2

JB P3.4,ONE

SETB P3.3

JB P3.4,STAR

C2: DEC R4

INC R0

SETB P3.0

JB P3.5,EIGHT

SETB P3.1

JB P3.5,FIVE

SETB P3.2

JB P3.5,TWO

SETB P3.3

JB P3.5,ZERO

SEVEN: MOV A,#'7'

ACALL DAT1

ACALL DEL

MOV @R0,#'7'

JMP KEYPAD

FOUR: MOV A,#'4'

ACALL DAT1

ACALL DEL

MOV @R0,#'4'

JMP KEYPAD

ONE: MOV A,#'1'

ACALL DAT1

ACALL DEL

MOV @R0,#'1'

JMP KEYPAD

STAR: MOV A,#'\*'

ACALL DAT1

ACALL DEL

MOV @R0,#'\*'

JMP KEYPAD

NINE: MOV A,#'9'

ACALL DAT1

ACALL DEL

MOV @R0,#'9'

SJMP KEYPAD

SIX: MOV A,#'6'

ACALL DAT1

ACALL DEL

MOV @R0,#'6'

SJMP KEYPAD

THREE: MOV A,#'3'

ACALL DAT1

ACALL DEL

MOV @R0,#'3'

SJMP KEYPAD

HASH: MOV A,#'#'

ACALL DAT1

ACALL DEL

MOV @R0,#'#'

SJMP KEYPAD

AA: MOV A,#'A'

ACALL DAT1

ACALL DEL

MOV @R0,#'H'

SJMP KEYPAD

BB: MOV A,#'B'

ACALL DAT1

ACALL DEL

MOV @R0,#'H'

SJMP KEYPAD

CC: MOV A,#'C'

ACALL DAT1

ACALL DEL

MOV @R0,#'H'

SJMP KEYPAD

DD: MOV A,#'D'

ACALL DAT1

ACALL DEL

MOV @R0,#'H'

SJMP KEYPAD

;FOR ENCRYPTION

DAT1: MOV P1, '\*'

SETB RS

SETB E

ACALL DEL2

CLR E

RET

;DELAY

DEL: MOV R1,#255

D2 : MOV R2,#255

D3 : DJNZ R2,D3

DJNZ R1,D2

RET

DEL2: MOV R1,#10

D22 : MOV R2,#20

D33 : DJNZ R2,D33

DJNZ R1,D22

RET

W2: MOV R4,#7

WAIT2:

ACALL DEL

DJNZ R4,WAIT2

RET

W3: MOV R4,#50

WAIT3:

ACALL DEL

DJNZ R4,WAIT3

RET

PASS: DB "1234"

STRING1: DB "ENTER PIN: "

STRING2: DB "SUCCESS"

STRING3: DB "FAILURE"

STRING4: DB "ALERT"

END

FAIL: CLR A

MOVC A,@A+DPTR

ACALL DAT

INC DPTR

DJNZ R4,FAIL

MOV R4,#10

WAIT: ACALL DEL

DJNZ R4,WAIT

DJNZ R7,X

MOV A,#01H

ACALL CMD

MOV A,#080H

ACALL CMD

ACALL DEL

ALERT: MOV R4,#5

MOV DPTR,#STRING4

AL: CLR A

MOVC A,@A+DPTR

ACALL DAT

INC DPTR

DJNZ R4,AL

CLR BUZZ

ACALL W3

SETB BUZZ

JMP AGAIN

RET

X: JMP AGAIN

RET

CMD: MOV P1,A

CLR RS

SETB E

ACALL DEL2

CLR E

RET

;FOR DISPLAY

DAT: MOV P1,A

SETB RS

SETB E

ACALL DEL2

CLR E

RET

KEYPAD: CLR P3.0

CLR P3.1

CLR P3.2

CLR P3.3

JMP KEY\_READ

RET

BEFORE\_CHECK:

MOV A,#01H

ACALL CMD

MOV A,#080H

ACALL CMD

ACALL DEL

MOV R0,#02FH

MOV R1,#03FH

MOV R4,#4

CHECK: INC R0

INC R1

MOV A,@R0

MOV B,@R1

CJNE A,B,FA

DJNZ R4,CHECK

MOV R4,#7

MOV DPTR,#STRING2

SUCCESS:

CLR A

MOVC A,@A+DPTR

ACALL DAT

INC DPTR

DJNZ R4,SUCCESS

CLR P2.3

CPL GLED

ACALL W2

CPL P2.3

ACALL WAIT3

CLR P2.2

ACALL W2

CPL P2.2

JMP AGAIN

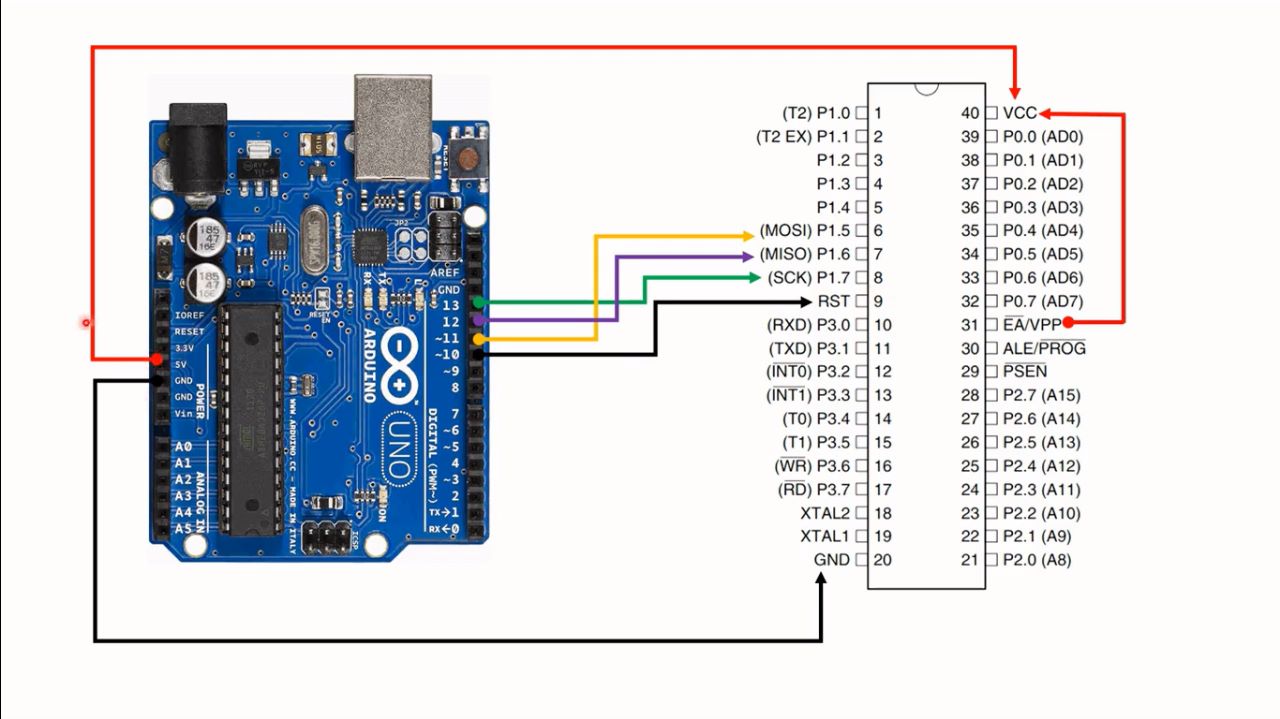
FA: MOV R4,#7

MOV DPTR,#STRING3

CPL RLED

# References:

* <https://youtu.be/Zhsktt0S1Fo>
* <https://www.youtube.com/watch?v=qzqSjv65ubQ>



# Diagram Description automatically generatedA picture containing text, electronics Description automatically generatedDiagram Description automatically generated

**4x4 Matrix keypad**

**AT89S52**