

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/339553290>

Design and Implementation of Embedded Password Based Security Door Lock System using 8051 microcontroller

Article · October 2017

CITATIONS

0

READS

13,066

1 author:



[Emmanuel Ajayi-Smart](#)

Ekiti State University, Ado Ekiti

5 PUBLICATIONS 1 CITATION

SEE PROFILE

Design and Implementation of Embedded Password Based Security Door Lock System using 8051 microcontroller

Ajayi-Smart Emmanuel Abiodun

Department of Electrical & Computer Engineering
Ekiti State University, Ado-Ekiti, Nigeria
ajayismartea@yahoo.com

Abstract

The degree of security is feeble in this current situation. So there is lots of theft, theft taking place in and round the world. Human beings worry to keep any of their valuables in their houses. Henceforth, many people opt to maintain it in banks. However, in this insecure world even banks aren't too secure enough to meet human beings needs. A not unusual man feels his treasured are secured if there may be performance in safety. Therefore these studies can give effective security in minimum price.

Keywords: Security, door, lock, microcontroller, Liquid Crystal Display, password, keypad.

INTRODUCTION

Due to the advancement of science and technology throughout the world, there may be a consequent growth inside the fee and sophistication of crime [7]. As a end result, it is important to make certain safety of oneself and one's treasured belongings. in spite of the usage of mechanical locks, the crime price still has multiplied due to the truth that those locks are effortlessly damaged. Consequently, there may be a want for other varieties of locks mainly electronic ones[9]. This work is of an electronic combination lock with a keyboard to be mounted on the door for keying in the secret code. The code unit, which operates with a 12-switch (matrix) keypad, was designed to control an electromechanical door lock with five-digit code. Unlike other keyboard combination locks this lock is constructed in such a way that once any of the wrong keys is pressed, it resets automatically making it harder for an intruder to break into[4]. It also blows an alarm after 3 error count entry.

The increasing rate of crime, attacks by thieves, intruders, vandals etc., despite all forms of security gadgets and lock

constitute the main factor that prompts the selection of this work.

“Password Based Door Security System using Microcontroller” is used in the places where we need more security[9]. It can also use to secure lockers and other protective doors. The system comprises a number keypad and the keypads a reconnected to the 8 bit microcontroller AT89C51[2][15]. This is one of the popular Microcontrollers. It has only 40 pins and there are 32 input/output lines. The microcontroller has a program reminiscence of 4Kilobytes. The microcontroller constantly display the keypad and if somebody enters the password it will take a look at the entered password with the password which changed into stored inside the memory and if it they are equal then the microcontroller will switch on the corresponding device. The system will allow the person that knows the password and it'll not allow who don't know the password and the machine may even show the men and women who attempt to break the safety barrier.

BACKGROUND

The earliest lock in existence is the Egyptian lock, made of timber, determined with its key in the Palace ruins in Nineveh, in ancient Assyria[10]. Within the 19th century, stage locks, cylinder locks and keyless locks have been invented and stepped forward upon [12]. The primary successful steel key changeable mixture lock was invented by way of James Sargent in 1857[13]. This lock became the prototype of those being utilized in current bank vaults.

In 1958, the first electronic aggregate lock become invented[14]. As next traits were alongside the strains, the locks were advanced upon by way of the improvement of substances and increasing complexity of the operating mechanisms which includes the increasing use of automatic electronic alarm and safety devices[14].

Conventional lock structures using mechanical lock and key mechanism are being changed by using new advanced strategies of locking gadget [1]. those techniques are an integration of mechanical and digital gadgets and extraordinarily sensible. One of the prominent capabilities of those revolutionary lock structures is their simplicity and excessive efficiency. Such an automatic lock gadget includes electronic manipulate assembly which controls the output load through a password. This output load may be a motor or a lamp or some other mechanical/electric load. Here I advanced an electronic code lock machine the use of 8051 microcontroller, which offers manage to the actuating the burden. It's far a simple embedded machine with enter from the keyboard and the output being actuated hence. This device demonstrates a password based totally door lock system wherein once the correct code or password is entered, the door is opened and the

involved character is authorized get entry to the secured region. once more if every other character arrives it'll ask to enter the password. If the password is wrong then door could remain closed, denying the get admission to the man or woman [11]. Principle at the back of the Circuit: the principle aspect in the circuit is 8051 controller. In this challenge four×three keypad is used to go into the password. The password that is entered is as compared with the predefined password. If the entered password is correct then the machine opens the door by rotating door motor and presentations the status of door on liquid crystal display. If the password is incorrect then door is stay closed and displays "pwd is inaccurate" on lcd.

SCOPE

The microcontroller based Door locker is an access control system that allows most effective legal individuals to get entry to a constrained area. The gadget is completely controlled through the eight bit microcontroller AT89C51 which has 4K bytes of flash programmable and erasable study best memory (PEROM). The system has a keypad by using which the password may be entered via it. When they entered password equals with the password saved within the reminiscence then the door gets open. If we entered a wrong password then the Alarm is switched on. The default password is set. There's a grasp code which lets in the proprietor to reset the password or overwrite the modern password. It additionally has a liquid crystal display (display unit) which displays the all of the moves to be carried out and also the comments of such activities.

Arduino UNO

Arduino makes it as easy as possible to program tiny computers called microcontrollers. This microcontroller is based on the ATmega 328. There are total of 20 pins (0-19) out of which 6 are analog

inputs which can also be used as general purpose pins, a ceramic resonator of frequency 16MHz, an USB connection, a power jack and a reset button. It contains everything needed to support a microcontroller [4].

SYSTEM HARDWARE DESIGN

The system was designed with Fig 3 as the block diagram. Password based door lock

circuit design uses five major components – a Microcontroller, a Servo, a 4×4 matrix keypad, a Buzzer and a LCD. Here Aduino microcontroller is used and it is an 8-bit controller. This controller requires a supply voltage of +5V DC. In order to provide regulated 5V DC voltage to the controller we need to use 7805 power supply circuit. We can use 9V DC battery or 12V, 1A adaptor as a power source.

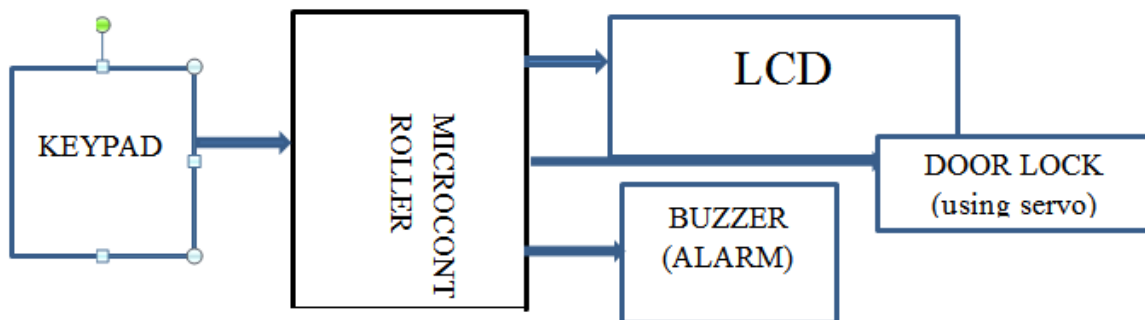


Fig 1: The system block diagram

The block design is subdivided into the under listed sub-unit for easy design, analysis and integration.

- Power Supply unit
- Keypad unit
- Controller unit
- LCD unit
- Lock unit
- Servo motor
- Buzzer

POWER SUPPLY UNIT

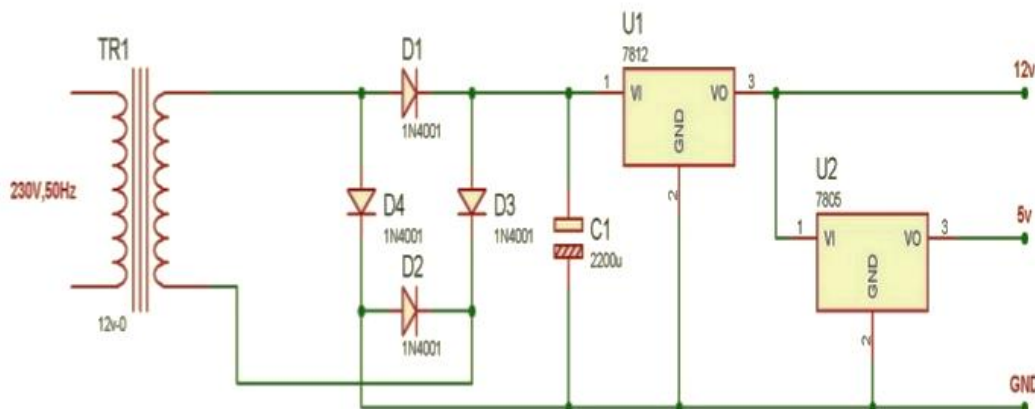


Fig 2: Power Supply

For a proper function of any microcontroller, it's far essential to offer a strong source of supply according to technical specifications by the producer of % microcontroller, deliver voltage have to pass between 2.0V to 6.0V in all variations.

KEYPAD UNIT

A keypad is a common primary input device for embedded microcontrollers. A Keypad actually consists of push buttons or press-to-make switches arranged in a column-by-row matrix (In this case four by three matrix) as shown in the fig 5 below. One terminal of each switch is connected to microcontroller via a pull-up resistor, and the other terminal is connected to ground. When the switch is not pressed, it sends logic '1' (high signal) to the microcontroller else it sends logic '0' (low signal).

Summary about Keypad pins:

1. Maximum operation rating: 24VDC
2. Insulation Resistance: 100M ohm, 30 MA.
3. Interface: 8 pins can be accessed in the form of 4X4 matrix.

The keypad contains four lines, each consisting of three characters. The keypad wires are connected to the controller by the Ports (D2-D9) via pull-resistors each of 100M ohm resistance. The '*' key serves as enter, while the 'A' key enables one to change the codes of the lock. The '#' key can also serve as key 9 for the input code and the '*' key can also serve as key 8 for the input code. 'LOCK' key on the keypad is used to roll the electric motor anti-clock-wisely back to its lock state. The '#' key is the reset key which is used to reset the microcontroller to the default code in case the present code is forgotten.

The keypad unit comprises mainly of the keyboard and its switches each can generate a discrete signal when processed. It is made up of ten switches, of which five will be used as the key in the secret code, another five will serve as the reset switches, and the remaining two will serve as a decoy.

By using the ten switches, the number of possible combinations was calculated. By applying the permutation principle, the numbers of entries that can be made is given by;

$$P = {}^{10}C_5 = \frac{10!}{5!5!} = \frac{10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1}{5 \times 4 \times 3 \times 2 \times 1 \times 5 \times 4 \times 3 \times 2 \times 1} = 252$$

This means that there are 252 ways in which this combination can be set, which means that the probability of an intruder to break the code is around 1 out of 252 ways[16].

CONTROLLER UNIT

The Arduino Uno is a microcontroller board based on the ATmega328. It has fourteen digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a sixteen MHz crystal oscillator, a USB connection, a energy jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter. "Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USBArduinoboard[2].

SPECIFICATION SUMMARY

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins output)	14 (of which 6 provide PWM
Analog Input	Pins 6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory bootloader	32 KB of which 0.5 KB used by
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

POWER OF THE BOARD

The Arduino Uno can be powered thru the USB connection or with an outside power supply. The energy source is chosen automatically.

External (non-USB) energy can come both from an AC-to-DC adapter (wall-wart) or battery.

The adapter may be linked by way of plugging a 2.1mm center-line plug into the board's power jack. Leads from a battery may be inserted within the Gnd and Vin pin headers of the power connector.

The board can perform on an outside supply of 6 to 20 volts. If provided with much less than 7V, but, the 5V pin may also deliver much less than five volts and the board can be unstable. If the use of more than 12V, the voltage regulator can also overheat and damage the board. The encouraged variety is 7 to twelve volts [11].

The power pins are as follows:

- VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- 5V. The regulated power supply used to

power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.

- 3.3V. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- GND. Ground pins.

MEMORY OF THE BOARD

The has Atmega328 32 KB of flash memory for storing code (of which 0,5 KB is used for the boot loader); It has also 2 KB of SRAM and 1 KB of EEPROM .

OVERVIEW OF AN ATMEGA328

The ATmega328P is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega328P achieves throughputs approaching 1 MIPS per MHz allowing the system designed to optimize power consumption versus processing speed[2].

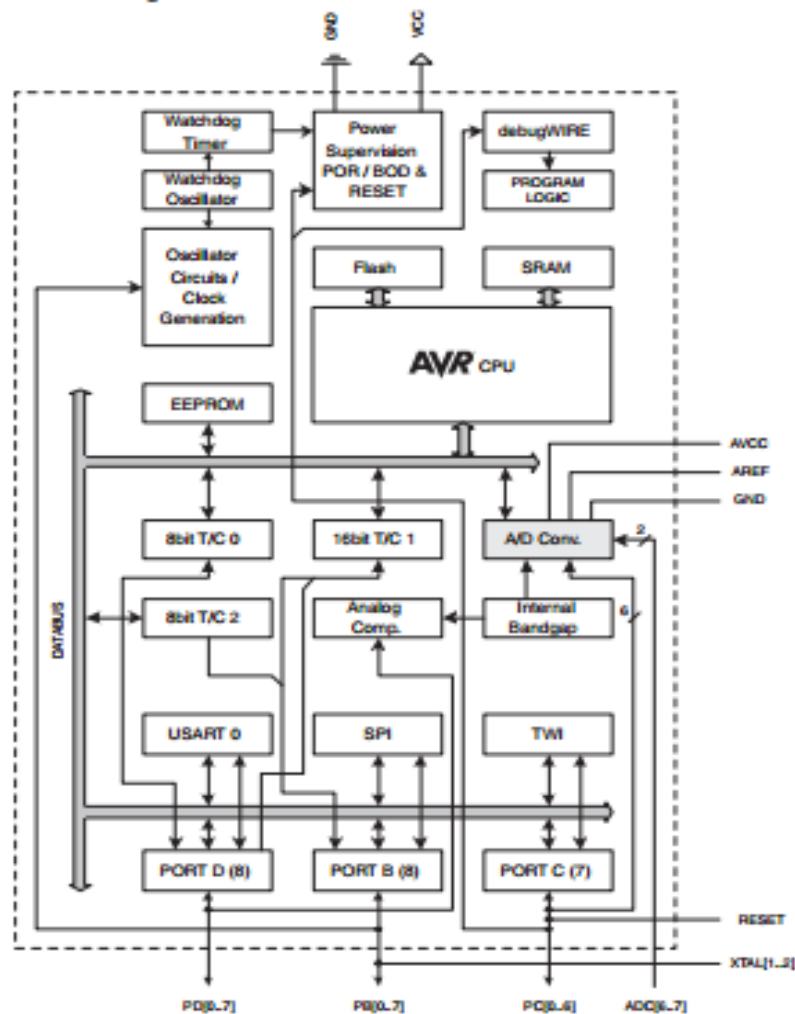
BLOCK DIAGRAM OF AN ATMEGA328

The AVR core combines a rich instruction set with 32 general purpose registers. all of the 32 registers are at once related to the arithmetic logic Unit (ALU), permitting two independent registers to be

accessed in one unmarried preparation finished in one clock cycle. The ensuing architecture is more code green whilst attaining throughputs up to 10 instances faster than conventional CISC microcontrollers.

The ATmega328P offers the following capabilities: 32K bytes of In device Programmable Flash with study-even as-Write abilities, 1K bytes EEPROM, 2K bytes SRAM, 23 popular purpose I/O lines, 32 preferred reason operating registers, 3 bendy Timer/Counters with examine modes, internal and outside interrupts, a serial programmable USART, a byte-orientated 2-twine Serial Interface, an SPI serial port, a 6-channel 10-bit ADC (eight channels in TQFP and QFN/MLF programs), a programmable Watchdog Timer with internal Oscillator, and 5 software program selectable energy saving

modes. The Idle mode stops the CPU at the same time as allowing the SRAM, Timer/Counters, USART, 2-wire Serial Interface, SPI port, and interrupt system to retain functioning. The power-down mode saves the sign in contents but freezes the Oscillator, disabling all other chip capabilities till the following interrupt or hardware reset. In strength-shop mode, the asynchronous timer continues to run, allowing the person to maintain a timer base even as the rest of the tool is snoozing. The ADC Noise reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to limit switching noise in the course of ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the relaxation of the tool is sleeping. This permits very speedy start-up mixed with low power consumption.



The device is manufactured using Atmel's high density non-volatile reminiscence generation. The On-chip ISP Flash permits the program reminiscence to be reprogrammed In-machine via an SPI serial interface, through a traditional non-unstable memory programmer, or via an On-chip Boot software strolling on the AVR core. The Boot program can use any interface to download the application in the software Flash memory. software within the Boot Flash segment will continue to run while the utility Flash segment is updated, providing genuine read-whilst-Write operation. With the aid

of combining an 8-bit RISC CPU with In-machine Self-Programmable Flash on a monolithic chip, the Atmel ATmega328P is a effective microcontroller that offers a noticeably bendy and price powerful technique to many embedded control applications.

The ATmega328P AVR is supported with a complete suite of program and device development tools along with: C Compilers, Macro Assemblers, software Debugger/Simulators In-Circuit Emulators, and evaluation kits.

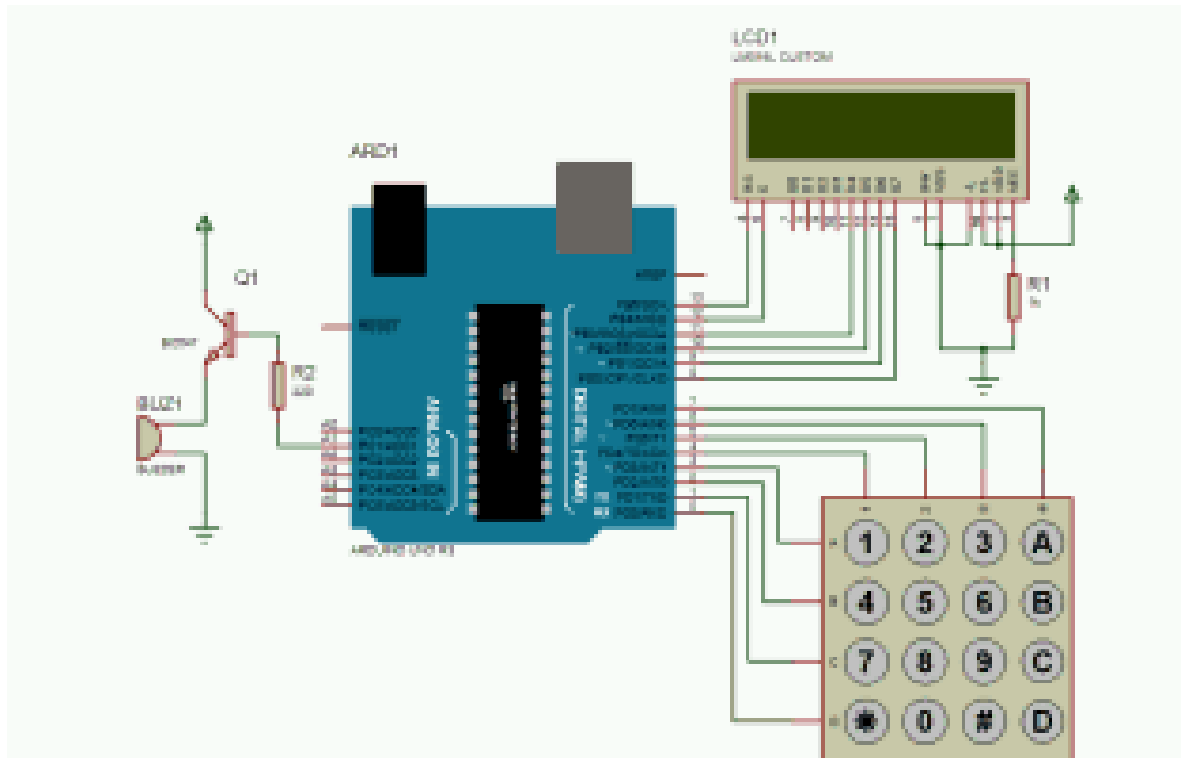


Fig 4: Circuit Diagram

COMMUNICATION IN THE SYSTEM

The Arduino Uno has a number of facilities for communicating with a computer, every other Arduino, or other microcontrollers. The ATmega328 gives UART TTL (5V) serial conversation, that's available on virtual pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial verbal exchange over USB and looks as a digital com port to software program on the laptop. The '8U2 firmware uses the standard USB COM drivers, and no external motive force is wanted. But, on windows, an *.inf record is needed.. The Arduino software includes a serial reveal which allows simple textual statistics to be dispatched to and from the Arduino board. The RX and TX LEDs at the board will flash while records is being transmitted thru the USB-to serial chip and USB connection to the pc (but not for serial verbal exchange on pins 0 and 1).

A software program Serial library lets in for serial conversation on any of the Uno's digital pins.

The ATmega328 also support I2C (TWI) and SPI communique. The Arduino software program consists of a twine library to simplify use of the I2C bus

PROGRAMMING

The Arduino Uno can be programmed with the Arduino software .Select "Arduino Uno w/ATmega328" from the Tools > Board menu (according to the microcontroller on your board). .The ATmega328 on the Arduino Uno comes pre burned with a boot loader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500protocol.

You can also bypass the boot loader and program the microcontroller through the ICSP (In Circuit Serial Programming) header.

AUTOMATIC (SOFTWARE) RESET

In preference to requiring a physical press of the reset button earlier than add, the Arduino Uno is designed in a manner that permits it to be reset via software jogging on a related laptop. One of the hardware go with the flow manipulate traces (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 thru a one hundred Nano farad capacitor. when this line is said (taken low), the reset line drops lengthy enough to reset the chip. The Arduino software program makes use of this capability to assist you to add code with the aid of truly pressing the upload button inside the Arduino surroundings. which means the boot loader could have a shorter timeout, as the decreasing of DTR may be nicely-coordinated with the begin of the add. This setup has different implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the boot loader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data. The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line.

USAGE OF DEVICE

Installation – You will be asked to input 5 digits as password at the *initial boot/reset* of the device. The first 5 digits you input

at installation will be saved as your **SET PASSWORD**. Or the initialized password.

Key * – for unlocking the device. Input correct password and **press *** for Unlocking.

Key # – for locking any time. Just **press #** and you will see the LCD cleared.

Key A – for changing the password. Input the correct password and **Press A**. You will see message asking to **ENTER NEW PASSWORD**. Enter 5 digits as password. The first 5 digits you enter will be SAVED as NEW PASSWORD.

Exceptions – You cannot use keys ‘*’, ‘#’ and ‘A’ inside the password combination. These 3 keys are control keys of the device with specific functions. The program checks for these key presses (at the password setting time – you may see the Set Password() function and look the *condition to check for invalid key press*) and identifies them as **Invalid Keys**. You will have to *input 5 new digits* as password after an Invalid Key press.

Important Variables and Arrays

password – is the array used to save and hold the user defined password.

Check password– is the array used to collect & hold user input. This user input data (in **check[]** array) is compared with **pass[]** array to authenticate password.

Getkey – is the variable used to identify initial entry point of the program. User is asked to SET a 5 Digit Password at installation of Lock. Hence we need a variable to *identify entry* and *loop 5 times* to collect up to 6 digits and save them to **password[]** array. The same variable is later made use of to Change Password. When the key for Changing Password (here ‘A’) is pressed, this variable is simply assigned a zero value (the initial state of variable). This forces the program control to re-enter the Password Setting Loop of the program.

key_id – is the variable used to perceive a key press and perform some actions within the application (that should manifest most effective on a key press). by way of default

this variable is about 0 initial value. Each time a key is pressed in key pad, this variable could be assigned a **value =1**. You may check the **keyscan()** function to see this. This simple trick helps to identify a key press and perform various actions on that key press (based on the value of key press). This variable **is set to zero** at different points in the program (to prevent the value 1 in **key_id** variable being identified as a false key press). You may check them as well.

Note:-col_scan – is the actual variable that gets activated to a **LOW** on key press (hence helps in identifying key press). But this variable is actually a part of the key pad interfacing program **.lcd_setcursor** – is a simple counter variable used to iterate the column position of LCD module. This variable helps to display user input data successively in row 2 of LCD module.

Subroutines used in the Program

SetPassword() – is the subroutine used to SET user defined password. This subroutine may be very depending on the “**Password Setting Loop**” written within the main program. This password placing loop can be iterated at set up of the device (that is on the boot or reset of the tool) for first **5 key presses**. This first five key press can be used to SET the Password. those key presses might be saved to **password[]** array. As mentioned earlier, **entry** is the variable used to iterate the loop **5** times. **key_id** is the variable used identify key press.

Note:- The same “**Password Setting Loop**” is made use of for converting the Password as properly. while key ‘C’ is pressed, the cutting-edge password is checked for. If the input password is matching with modern-day SET password, then **entry** variable might be assigned to **zero** value. This may definitely transfer the manipulate of this system to go into the Password placing Loop again.

keyscan() –is the subroutine to scan keypad for a key press. This subroutine is essentially same as the version 2 code of

interfacing hex keypad to arduino. i have brought a few strains of code wished for this code lock. apart from that, the strains of code on this subroutine is identical as that of interfacing keypad. **keyscan()** subroutine scans for a key press (whenever the function is called from Main program or from other sub routines like Set Password()) and identifies the row and column of the pressed key. If key ‘1’ is pressed, keyscan identifies that key at **row 1** and **column 1** is pressed. Similarly if key ‘6’ is pressed, the keyscan identifies a key is pressed at row 2 and column 3. Whenever a key is pressed, another subroutine named **keypress()** is invoked within the **keyscan()** routine. This **keypress()** routine is used identify the value of key press (say ‘1’, ‘2’, ‘3’ or ‘A’, ‘C’ or ‘D’ etc).

keypress() – as mentioned above is the subroutine to identify value of key press. The **keyscan()** routine identifies which row and column of key pad is pressed. This **row and column** number is passed to **keypress()** routine as parameters (using variable values of i and j).

Check Password() – is the subroutine to check user input password against the SET User Defined Password. The user input data (password to cross check) is collected in the **checkpassword[]** array. This is compared against the SET Password inside **2password[]** array. For comparing. If each digit inside the arrays matches, flag variable will remain zero. If any mismatch occurs, the flag will be set to 1 and loop will break.

SYSTEM REQUIREMENT

The following hardware and software requirements were needed in other to achieve this work.

Software requirement

- Windows 98/2000/XP or any version of windows
- Assembly language

System hardware requirement

The Embedded Password based security door lock is an electronic device that is made up of different kinds of component which are put together to perform a particular task. These components are:

- Microcontroller
- Servo
- 4×4 matrix keypad
- Buzzer
- LCD (liquid crystal display)
- Connecting wires etc.

IMPLEMENTATION

In the implementation once the circuit is powered with a voltage of +5V DC. In order to provide regulated 5V DC voltage to the controller we need to use 7805 power supply circuit. We can use 9V DC battery or 12V, 1A adaptor as a power source.

The microcontroller sends commands to the LCD such that it is ready to accept the data. This data is entered using the keypad. This Keypad module's Column pins are directly connected to pin 5,4,3,2 and Row pins are connected to 9,8,7,6, of arduino. Once the data is entered, it is displayed on the LCD. This data denotes the password. According to the program, the microcontroller again sends commands to clear the LCD and then sends the data to be displayed on the LCD. The password is entered using the keypad and it is checked with the set password. If the passwords match, the microcontroller sends signals to the motor such that pin 11 is at high level and the motor rotates in forward direction. After a certain time delay, the enable pin is grounded by sending a low logic signal from the microcontroller and the motor stops. Again after some certain time delay, the microcontroller sends signals to the motor driver such that pin 11 is at logic low level. The motor now rotates in backward direction.

Now if the passwords do not match, the microcontroller sends a logic low signal to the enable pin of the microcontroller, thus disabling the motor driver and the motor does not runs at all. And one buzzer is connected at pin 10 of arduino. The buzzer is at a high state whenever the password is typed wrongly three (3) times consecutively



Fig 5: Show user enter password

SYSTEM TESTING

Involves the testing of the entire circuitry and cross examine it for errors like short circuits, joining unwanted links, proper 222insertion pin and also checking if board of these pin number but different function is slotted in their proper base. After this check cross examine once again before powering the system.

1. Switch ON the power
 - LCD will display "ENTER PASSWORD"
2. Press 12345 on keypad (default password)
 - The servo will turn.
 - LCD will display "OPEN"
 - Door lock will release.
3. Press any 2-6 number on keypad
4. Press Reset button 'A'
 - LCD will display "Success"
 - Door lock will remain lock.
5. Press clear screen button '#'
6. Repeat the 2-6 number pressed on keypad
 - Door lock will release

RECOMMENDATION

To meet up with the technological demands of the country, the following recommendation should be considered:

1. We can send this data to a remote location using mobile or internet we can implement other related modules like fire sensor, wind sensor.
2. This circuit can be also modified by using EEPROM chip interfaced to the microcontroller and store the entered password in the chip.

CONCLUSION

This digital code lock is very marketable as it is easy to use, relatively inexpensive due to low energy consumption, and especially reliable. This virtual code lock is therefore especially beneficial in applications which include door locks and device locks.

REFERENCE

1. Ankit, J., Anita, S., and Ritu, R., 2016, "Password Protected Home Automation System with Automatic Door Lock," MIT International Journal of Electrical and Instrumentation Engineering. Vol.6, No.1, pg 28-31.
2. Evans, B.W., 2007, Arduino Programming Notebook(first edition). Cambridge University Press. UK 2007, ISBN: 1463698348
3. Graw-Hill, M.C., 1985, Encyclopedia of Science and Technology (5th Edition). Cambridge University Press 1985, ISBN 042507843, pg 2-253.
4. Ishra, A., Arma, I., Ubey, S., and Ubey, K., 2014, "Password Based Security Lock System," International Journal of Advanced Technology in Engineering and Science, Vol.2, No.5.
5. Jgaonkar, N., Dekar, R., and Ndagale, P., 2013, "Automatic Door Locking System," International Journal of Engineering Development and Research, Vol.4, No.1.
6. Koenig, J.A., and Taylor, L., 1998, "Perimeter Security Sensor Technology Handbook," Electronic Security Systems Engineering Division, North Charleston, USA, pg 67-86.
7. Manish, K., Hanumanthappa, M., Kumar, T. V. S., and Amit, K. O., 2016, "Android Based Smart Door Locking System with Multi User and Multi Level Functionalities," International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE). Vol.5 Issue 2 pg115-118.
8. Mishra, A., Sharma, S., Dubey, S., and Dubey, S.K., 2014, "Password Based Security Lock System," International Journal of Advanced Technology in Engineering and Science, Vol. No.2, Issue No.5.
9. Mohammed, A., 2013, "Development of a Microcontroller Based Security Lock for a Car Engine," IOSR Journal of Electrical and Electronics Engineering(IOSR-IEEE) Vol.24, No.6, pg 1-30.
10. Neelam, M., Ruhina, H. and Priyanka, B., 2016, "Automatic Door Locking System," International Journal of Engineering Development and research (IJEDR) Vol.4 Issue 1 pg 495-499.
11. Oladunmoye, M., Oluwatomi, A. A., and Obakin, O., 2014, "Design And Construction of an Automatic Sliding Door Using Infrared Sensor," Computing, Information Systems, Development Informatics & Allied Research Journal Vol. 5 No. 4.
12. Pradnya, R. N., Chaudhari, J.P., Pachpande, S. R., and Rane, K. P., 2016, Literature Survey on Door Lock Security Systems. International Journal of Computer Applications Vol. 153 No.2, pg13-17.
13. Sedhumadhavan, S., and Saraladevi, B., 2014, "Optimized Locking and Unlocking a System Using Arduino,"

- International Journal of Innovative Research in Computer and Communication Engineering, Vol. 2, Issue 11.
14. Shruti, G., Anirban, C., Srijan, D., Tushit, B., and Soham, 2017, "Automated Password Protected Door Lock System," Advances in Industrial Engineering and Management Vol.6, No.1. pg 48-52.
 15. Smith, A. G., 2011, "Introduction to Arduino," A piece of cake, sparkfun electronics Inc. USA, ISBN-13: 978-1463698348, pg 60-80.
 16. Theraja, B. L and Theraja, B. K(2002): A Textbook of Electrical Technology. S. Chand and Company Ltd., New Delhi India 2002, pg 220, 920, 924, 1712-1716.
 17. Weber, M., and Thad, L., 1985, "Alarm Systems and Theft Protection (2nd edition)," Stoneham; Butterworth UK, pg 7-8. ISBN: 0913708119
 18. Zungeru, A.M., 2012, "Design and Implementation of a Low Cost Digital Bus Passenger Counter," International Journal of Innovative Systems Design and Engineering (IJISE), Vol.3 No.4 pg. 29–41.
 19. Zungeru, A.M., Kolo, J. G., and Olumide, I., 2012, "A Simple and Reliable Touch Sensitive Security System," International Journal of Network Security and its Applications (IJNSA) Vol.4 No.5 pg 149-165.