

SECURED ROOM ACCESS SYSTEM



■ AKSHAY MATHUR

Security is a prime concern in our day-to-day life. And access control system forms a vital link in a security chain.

The microcontroller-based digital lock presented here is an access control system that allows only authorised persons to access a restricted area. When someone tries to enter the restricted area by entering invalid passwords continuously, the system locks itself and can be unlocked only by the master user.

The system comprises a small electronic unit with a numeric keypad,

which is fixed outside the entry door to control a solenoid-operated lock. When an authorised person enters a predetermined number (password) via the keypad, the relay energises for a limited time to unlock the solenoid-operated lock, so door can be pushed/pulled open. At the end of the preset delay, the relay de-energises and the door gets locked again. A prompt message is displayed on the LCD module.

Circuit description

The system uses a compact circuitry built around AVR microcontroller ATmega8535. The ATmega8535 is a low-power CMOS 8-bit

microcontroller based on the AVR-enhanced RISC architecture. It provides the following features: 8 kB of in-system programmable Flash memory with read-while-write capabilities, 512-byte EEPROM, 512-byte SRAM, 32 general purpose I/O lines, 32 general-purpose working registers, three flexible timer/counters with compare modes, and internal and external interrupts. The built-in power-on-reset circuitry of the microcontroller eliminates the need for external power-on-reset circuit.

Switch S3 is used to reset the system, which is accessible only to the master user. Port D (PD0 through PD7) is interfaced with the numeric keypad.

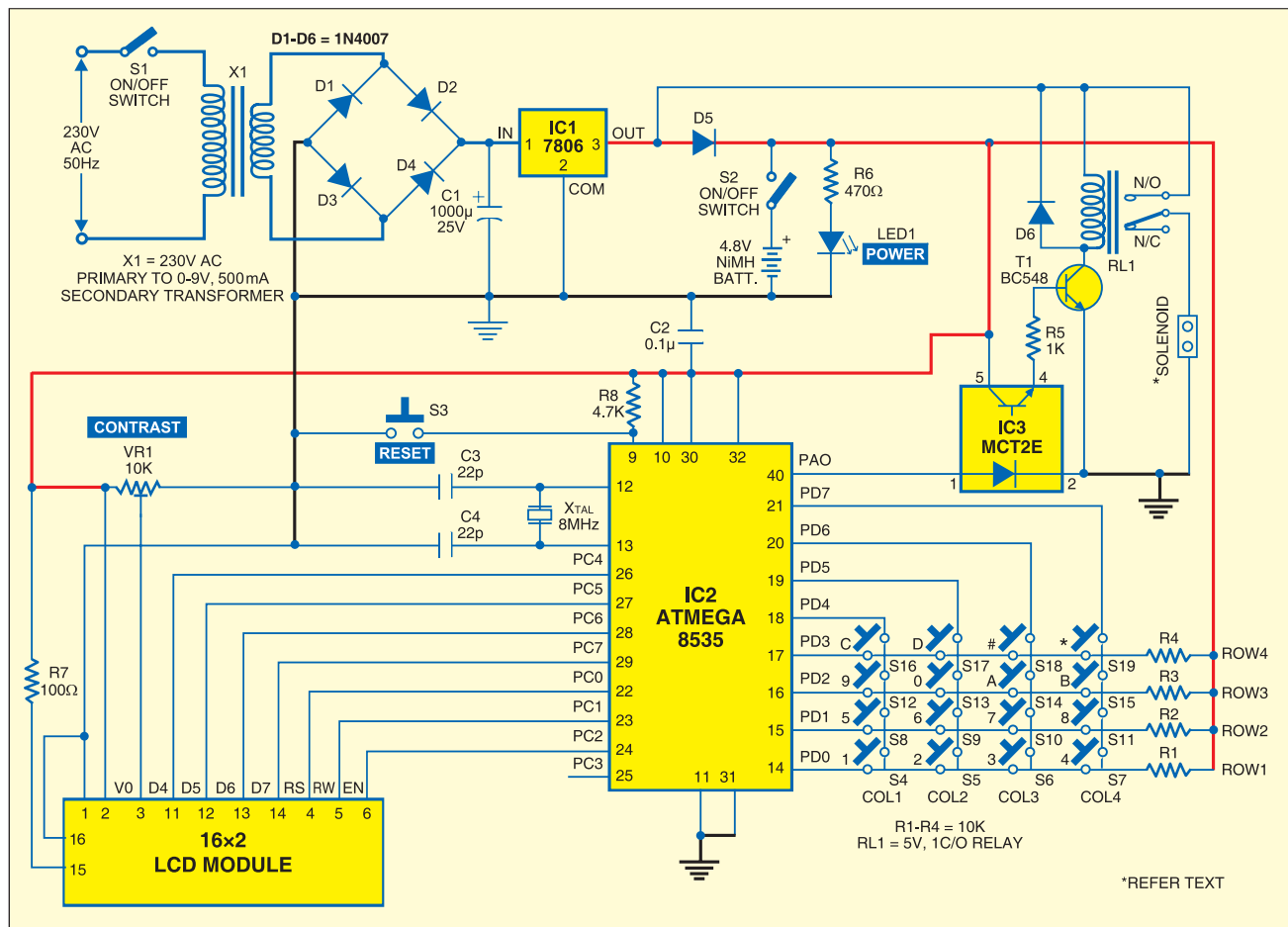


Fig. 1: Secured Room Access System

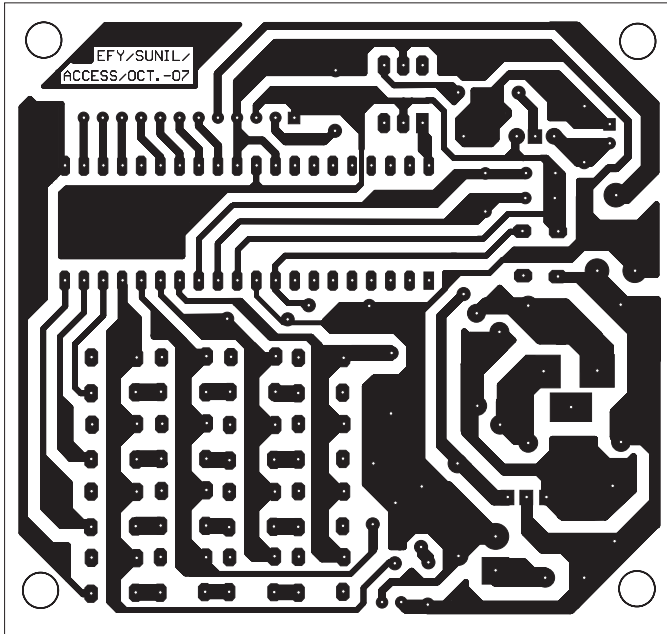


Fig. 2: A single-side, actual-size PCB layout for secured room access system

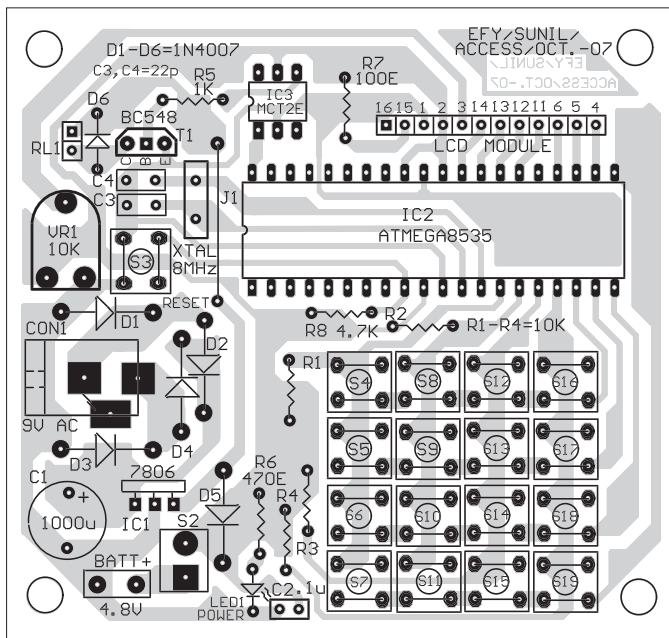


Fig. 3: Component layout for the PCB

Port C is interfaced with a 16-x2-line LCD. Four pins (PC4 through PC7) of Port C are used as data lines for the LCD module and three lines (PC0 through PC2) are used for controlling the LCD. Pin 40 (PA0) of port A is connected to the relay driver circuit through optocoupler MCT2E (IC3) and transistor T1.

When port pin PA0 goes high, the internal transistor of IC3 drives tran-

sistor T1 into saturation and relay RL1 energises. As the solenoid valve is connected through normally-closed (N/C) contact of the relay, the solenoid coil de-energises and the gate is locked. An 8MHz crystal is used with two 22pF capacitors for providing clock. Preset VR1 is used to adjust the contrast of the LCD.

The 230V, 50Hz AC mains is stepped down by transformer X1 to deliver a secondary output of 9V, 500 mA. The transformer output is rectified by a full-wave bridge rectifier comprising diodes D1 through D4, filtered by capacitor C1 and regulated by IC 7806 (IC1). Use adequate heat-sink for 7806 as the solenoid draws a high current. LED1 glows when power is 'on' and resistor R6 acts as the current limiter.

A 16-key numeric keypad for password entry is connected to the microcontroller. The keypad is also used for password change and application of master password when required. To economise the use of I/O pins, we have used here only eight pins for scanning and sensing 16 keys.

The keypad is arranged in a 4x4 matrix. There are four scan lines/pins, which are set in output mode, and four sense keys, which are used as input lines to the microcontroller.

At a small time interval, the microcontroller sets one of the four scan lines as low and the other three scan lines as high. Then it checks for the status of sense lines one by one at the intersection of a specific scan line and sense line to find out if any key has been pressed.

Similarly, after a small time interval, the next scan line is made low and remaining three scan lines are taken high, and again all three sense lines are checked for low level. This way the microcontroller checks which of the 16 keys is pressed.

Due to the high speed of the microcontroller, the status of different keys is checked in less than 100 ms and a key press is detected and identified. As the keys are pressed manually by the user, this delay of 100 ms is not noticeable. The net result is that you save on I/O pins of the microcontroller by sacrificing almost nothing.

When a person wants to enter the

PARTS LIST

Semiconductor:

- IC1 - 7806 6V regulator
- IC2 - ATmega8535 AVR microcontroller
- IC3 - MCT2E optocoupler
- T1 - BC548 npn transistor
- D1-D6 - 1N4007 rectifier diode
- LED1 - 5mm light-emitting diode

Resistors (all 1/4-watt, ±5% carbon):

- R1-R4 - 10-kilo-ohm
- R5 - 1-kilo-ohm
- R6 - 470-ohm
- R7 - 100-ohm
- VR1 - 10-kilo-ohm preset

Capacitors:

- C1 - 1000µF, 25V electrolytic
- C2 - 0.1µF ceramic disk
- C3, C4 - 22pF ceramic disk

Miscellaneous:

- X1 - 230V AC primary to 9V, 500mA secondary transformer
- S1, S2 - On/off switch
- S3-S19 - push-to-on tactile switch
- X_{TAL} - 8 MHz crystal
- RL1 - 6V, 1C/O relay
- Batt. - 4.8 volt rechargeable battery
- LCD module 16 X 2 line
- 6 volt operated solenoid lock

room, he enters the 6-digit password, say '123456.' If the password matches successfully, the gate is unlocked for 15 seconds.

If you want to change the user password (123456) and enter the master password '291279,' the system will ask you to change the user password. On successfully entering the password, pin A0 of port A becomes high for 15 seconds, because of which transistor T1 starts conducting through the emitter of the optocoupler and the relay

energises. The connection between the solenoid lock and the power supply is broken and the door is unlocked for 15 seconds.

An actual-size, single-side PCB for secured room access system (Fig. 1) is shown in Fig. 2 and its component layout in Fig. 3.

Software

The software for the AVR microcontroller is written in 'C' language and compiled using Code Vi-

sion AVR 'C' compiler. Since this compiler does not have library functions for the keypad, place 'kbd.h' file in the INC folder of the installation folder and 'kbd.lib' in the LIB folder of 'cavr' folder. This file is included in the program and the same can be used.

EFY note. All the software files relating to this article have been included in this month's EFY-CD.

The author is robotics engineer at Erokit Technologies Pvt Ltd. He can be contacted through 'akshaymathur39@yahoo.com.'

SOURCE PROGRAM

```
#asm
.equ __lcd_port=0x15
#endasm
eeprom long int pass_store= 123456,
master_password= 291279;
#include <mega8535.h>
#include <lcd.h>
#include <stdio.h>
#include <delay.h>
#include <kbd.h>
#define relay PORTA.0

int i,j,k,fail=0;
long int id_value, pass_value, pass[6];
bit match=0;
void password()
{
i=0;
do{
delay_ms(50);
while(!kbd_read()){
j=kbd_read();
if(j<11)
{if(j==10) j=0; pass[i]=j;
lcd_gotoxy(5+i,1);
lcd_putsf(" "); i++;}
} while(i<6);
delay_ms(100);
pass_value=pass[5]+ 10*pass[4]+ 100*pass[3]+
1000*pass[2]+ 10000*pass[1]+ 100000*pass[0];
}
void pass_change()
{
lcd_clear(); lcd_gotoxy(0,0);
lcd_putsf("Enter New ");
lcd_gotoxy(0,1);
lcd_putsf("Password:");
delay_ms(50); k=0;
do{
delay_ms(50);
while(!kbd_read()){
j=kbd_read();
if(j<11)
{if(j==10) j=0;
pass[k]=j;
k++;}
} while(k<6);
delay_ms(100);
pass_store=pass_value;
}
}
void unlock()
{
relay=~relay;lcd_clear();
lcd_gotoxy(1,0);
lcd_putsf("Door Unlocked");
lcd_gotoxy(2,1);
lcd_putsf("Please enter");
delay_ms(300);
for(k=0;k<20;k++)
{lcd_clear();
delay_ms(50);
lcd_gotoxy(1,0);
lcd_putsf("Door Unlocked");
lcd_gotoxy(2,1);
lcd_putsf("Please enter");
delay_ms(50);}
relay=~relay; return;
}
void main()
{
PORTA=0x00;
DDRA=0xFF;
kbd_init();
lcd_init(16);
lcd_clear();
lcd_gotoxy(4,0);
lcd_putsf("Welcome!");
delay_ms(100);
label1: lcd_clear();
lcd_gotoxy(1,0);
lcd_putsf("Enter Password:");
delay_ms(50);
password();
if(pass_value==pass_store){
match=1;
}
else if(pass_value==master_password) {
match=1;
pass_change();
match=0;
fail=0;
goto label1;
}
else {match=0;
}
if(match==1)
{
match=0;
fail=0;
delay_ms(100);
unlock();
goto label1;
}
else
{
lcd_clear();
lcd_putsf("Invalid Password");
match=0;
fail=fail+1;
delay_ms(200);
if(fail==3)
{
lcd_clear();
lcd_gotoxy(1,0);
lcd_putsf("Contact System ");
lcd_gotoxy(1,1);
lcd_putsf("Administrator");
}
else
{
goto label1;
}
}
}
}

```