Safe Lock Security system

EMBEDED PROJECT

GROUP MEMBERS:

1-Radwa Hassan Sadek
2-Shaahd Mohammed Saber

Project description:

Locker security system contains two AVR atmega32. First microcontroller interface with the user through the keypad and transmit the data through UART. Second microcontroller receive the data from the first microcontroller through UART then check the password and control the operations.

Drivers used in this project:

- **MC1 is called UserInterfaceMC→**
  1. DIO
  2. KEPAD
  3. UART
- **MC2 is called SecurityMC→**
  1. DIO
  2. LCD
  3. GINT
  4. TIMER0
  5. DC MOTOR
  6. UART

Steps of implementation:

- **MC1**:
  1. Initialization of keypad driver.
  2. Initialization of UART.
  3. Assign the keypad pressed key by the user to a variable called →KPD_VAL
  4. Check if the key is pressed then MC1 transmit the value of KPD_VAL to MC2.
- **MC2**:
  1. Define macro for password size and let it equal to 5
  2. Define two global arrays→ (PassArr) for the default password and (checkArr) for the password that user enter to check it.
  3. Define global variable called checkcounter with value 0.
  4. Define global waitflag with value 0.
  5. Declaration of KPD_LCD_FUNC→
     1) Reset the value of the global variable checkcounter.
     2) First LCD print "Enter pass "then go to axis (1,0) to print the password received from the user.

    3) In for loop limited by password size, User enters the password through KPD in MC1 and the pressed key is received then assigned to Rec variable then stored in the checkArr.
    4) At each element user press, LCD print"*" to hide the password.
    5) checkcounter increments if only each element of the password that user enteres which is stored in CheckArr matches each element of the password of the system which is stored in PassArr, so checkcounter will only reach the PassSize if the password is matched.

6. Declaration of Door operation & Wait until close functions:
    1) First clear the lcd screen.
    2) Lcd print "opening door".
    3) DC motor rotate clockwise for 1sec then stop.
    4) Clear lcd screen then print to the user "you are in you have 1minit".
    5) Timer0 calls (wait until close) function→in this function LCD print "door is closing" and the dc motor rotate anticlockwise for 1sec then stop, at the end wait flag become 1.
    6) Enable general interrupt
    7) Initialize timer0 in CTC mode
    8) If waitflag=0 stay do nothing so it will stay until wait until close function ends.
    9) DC motor stop for one minute then wait until close function is called.
    10) Lcd clear screen then prints "Door is closed".
    11) Reset waitflag =0;
    12) Disable general interrupt to prevent call bake function of the timer from looping.

7. Declaration of change password function:
    1) Clear Lcd screen
    2) User changes the default password and enters new password to the system through KPD in MC1.
    3) MC2 receive the new password and assign it in the password array element by element in for loop limited by password size.
    4) The system now has the new password.
    5) Clear LCD screen.
    6) Display password changed in lcd for 0.5sec then clear lcd.
8. Declaration of Alarm function:
    1) Clear LCD
    2) Print "Alarm !"
    3) Define static counter for the alarm to give some time to work in.
    4) Define static variable called toggle.
    5) In for loop toggle the led and buzzer every 0.1sec to give alarm by toggling the value of toggle and display its value on LED and buzzer pins.

9. Declaration of Lock system function:

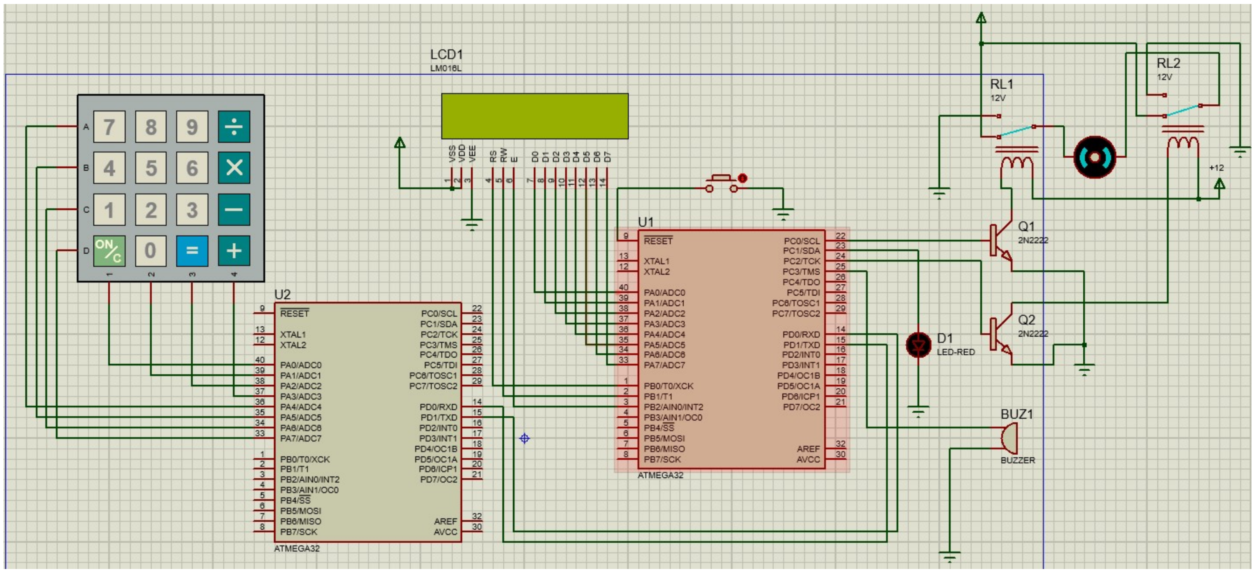
    1) Clear LCD
    2) Turn off the led and buzzer.
    3) Print “screen is locked “in LCD

10. Main:

    1) Declare variable called WrongCounter and initialize its value by zero.
    2) Define pin 1 and 3 of port c as output pins for LED and buzzer.
    3) Initialization of LCD.
    4) Initialization of UART.
    5) Initialization of DC motor
    6) In the infinite while loop→
    7) Call KPD_LCD_FUNC to receive the check password array from the user by UART and check, if it matched check counter reach password size.
    8) When checkcounter reaches passsize, that means password is matched and LCD display “Matched Pass” and “Load options”for 0.5sec then clear screen.
    9) LCD display operations options to the user→ at axis (0,0) “open door-> +” and at axis(1,0)“change pass-> -”.
    10) If the user press + ,that means open door so call (Door_operation) function.
    11) If the user press - ,that means change password so call(change password) function.
    12) If the user press any another key, LCD prints ”wrong choice” so that is the default case.
    13) If checkcounter not equal PassSize, that means not matched password so LCD prints ”Wrong Password” and red led flash for 0.5sec while wrong counter increment.
        - **Note that if the user enter the correct password after wrong password the wrongcounter is reset to 0.**
    14) After three following wrong tries, call (Alarm function) then call (lock screen function) So the whole system is stopped.

Components used:

1) Two ATMEGA32
2) KEYPAD-SMALLCALC
3) 16*2 Alphanumerical LCD→LM016L
4) DC MOTOR
5) Two RELAYS
6) Two NPN Bipolar Transistor → 2N2222
7) One Push button
8) One RED-LED
9) One BUZZER
10) 5 volt power
11) 12 volt power


LCD1
LM016L
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
7 8 9 ÷
4 5 6 ×
1 2 3 −
ON/C 0 = +
A B C D
1 2 3 4
U2
ATMEGA32
U1
ATMEGA32
RESET
XTAL1
XTAL2
PA0/ADC0
PA1/ADC1
PA2/ADC2
PA3/ADC3
PA4/ADC4
PA5/ADC5
PA6/ADC6
PA7/ADC7
PB0/T0/XCK
PB1/T1
PB2/AIN0/INT2
PB3/AIN1/OC0
PB4/SS
PB5/MOSI
PB6/MISO
PB7/SCK
PC0/SCL
PC1/SDA
PC2/TCK
PC3/TMS
PC4/TDO
PC5/TDI
PC6/TOSC1
PC7/TOSC2
PD0/RXD
PD1/TXD
PD2/INT0
PD3/INT1
PD4/OC1B
PD5/OC1A
PD6/ICP1
PD7/OC2
AREF
AVCC
RL1
12V
RL2
12V
+12
Q1
2N2222
Q2
2N2222
D1
LED-RED
BUZ1
BUZZER