EC382 : Embedded system Lab

Project Report

VIKAS BHATT

(2101223)

TITLE: Digital Combinational Lock with MATLAB Interface

Objective: - The digital lock project aims to create a secure access control system using a PSoC microcontroller, MATLAB GUI, and UART communication. The system allows users to enter a predefined password via a MATLAB graphical user interface (GUI). The password entered by the user is transmitted to the PSoC microcontroller over UART communication. The PSoC microcontroller receives the password, compares it with the predefined password stored in its memory, and grants access if the passwords match.

Components Used :-

- Microcontroller board
- MATLAB software
- USB cable
- LEDs
- LCD screen
- Resistors
- Pushbuttons

Theory:-

1. Combination Locks:-

Combination locks are mechanisms used to secure access to assets or facilities by requiring the input of a specific sequence of symbols or numbers, known as a combination. They are widely employed in various applications due to their simplicity, reliability, and effectiveness in restricting unauthorized access.

2. Digital Combinational Locks:-

Digital combinational locks employ electronic components, such as microcontrollers, to implement secure locking mechanisms. Unlike traditional mechanical combination locks, digital combinational locks utilize digital circuits and algorithms for password authentication and control.

3. Microcontrollers:-

Microcontrollers are integrated circuits (ICs) that contain a processor, memory, and input/output peripherals on a single chip. They are commonly used in embedded systems to control and manage electronic devices or processes. In the Digital Combinational Lock project, a microcontroller serves as the central processing unit responsible for handling password input, authentication, and controlling the lock mechanism.

4. User Interface:-

A user interface provides a means for users to interact with a system or device. In this project, the user interface consists of a MATLAB graphical user interface (GUI) and hardware components such as pushbuttons, LEDs, and an LCD screen. The MATLAB GUI facilitates password input and communicates with the microcontroller via a serial connection. Hardware components provide visual feedback and status indicators to the user.

5. Serial Communication:-

Serial communication is a method of transferring data between electronic devices over a serial interface. It involves transmitting data one bit at a time over a communication channel. In the Digital Combinational Lock project, serial communication is used to establish a connection between the MATLAB GUI and the microcontroller, enabling bidirectional data exchange for password transmission and status updates.

6. Password Authentication:-

Password authentication is the process of verifying the identity of a user by comparing an input password with a predefined correct password. In the project, the microcontroller authenticates passwords received from the MATLAB GUI by comparing them with a predefined set password. Upon successful authentication, the microcontroller triggers the unlocking mechanism, granting access to the secured area.

7. Hardware Components:-

Hardware components such as LEDs and an LCD screen are utilized to provide visual feedback and status indications to the user. LEDs are used to indicate the status of the lock (e.g., locked or unlocked), while the LCD screen displays messages related to password input and authentication.

8. Security Considerations:-

Security is a critical aspect of digital combinational locks, as unauthorized access could compromise the integrity of the system. To enhance security, measures such as encryption of transmitted data, implementation of timeout mechanisms, and protection against brute-force attacks should be considered during system design and implementation.

9. Future Enhancements:-

Future enhancements to the Digital Combinational Lock project could include the integration of additional security features, such as biometric authentication or multi-factor authentication, to further enhance access control and security measures. Additionally, improvements to the user interface and hardware components could be explored to enhance user experience and system functionality.

Code Implementation:-

1. Microcontroller Firmware:-

The firmware running on the microcontroller handles various tasks such as serial communication, password authentication, and control of hardware components. Below is the firmware code implemented in the PSoC microcontroller using Cypress' PSoC Creator environment:

```
// Microcontroller firmware code (PSoC Creator
environment)

#include "project.h"
#include <stdio.h>
#include <string.h>

// Define LED states
#define LED_ON  (1u)
#define LED_OFF (0u)
#define DELAY_TIME_MS 2000

#define PASSWORD_LENGTH 5
#define USBUART_BUFFER_SIZE (64u)

unsigned char setPassword[PASSWORD_LENGTH
+ 1] = "1234#"; // Set password
unsigned char Password[PASSWORD_LENGTH + 1]
= "*****"; // Set password
unsigned char
receivedPassword[PASSWORD_LENGTH + 1] = {0};
// Received password

void processReceivedData() {
   // Process received data here
   // For example, check if received password
matches the set password
   if (strcmp(receivedPassword, setPassword) ==
0) {
      // Password matches, display it on the LCD
      LCD_ClearDisplay();
      LCD_Position(0, 0);
      LCD_PrintString("Password Match:");
      Pin_1_Write(LED_ON);
      Pin_2_Write(LED_ON);
      LCD_Position(1, 0);
      LCD_PrintString(Password);
      CyDelay(DELAY_TIME_MS);

      // Show lock is open
      LCD_ClearDisplay();
      LCD_Position(0, 0);
      LCD_PrintString("Lock is open     ");
      LCD_Position(1, 0);
      LCD_PrintString("Access granted");
   } else {
      // Password does not match, display an error
message on the LCD
      LCD_ClearDisplay();
      LCD_Position(0, 0);
      LCD_PrintString("Invalid Password");
      Pin_1_Write(LED_ON);
      Pin_2_Write(LED_OFF);
      CyDelay(DELAY_TIME_MS);

      // Show access denied message
      LCD_ClearDisplay();
      LCD_Position(0, 0);
      LCD_PrintString("Access denied");
      CyDelay(DELAY_TIME_MS);
      LCD_ClearDisplay();
      LCD_Position(0, 0);
      LCD_PrintString("Enter Correct   ");
      LCD_Position(1,0);
      LCD_PrintString("Password");
   }
}

int main(void) {
   CyGlobalIntEnable; /* Enable global interrupts.
*/
   USBUART_Start(0, USBUART_3V_OPERATION);
/* Start USBUART operation */
   LCD_Start(); // Start LCD

   uint16 count ;
   uint8 buffer[USBUART_BUFFER_SIZE];
   int i;

   for (;;) {
      if(0u != USBUART_IsConfigurationChanged())
{
         if(0u != USBUART_GetConfiguration()) {
            USBUART_CDC_Init();
         }
      }

      if(0u != USBUART_GetConfiguration()) {
         if(0u != USBUART_DataIsReady()) {
            char rcv = USBUART_GetChar();
            receivedPassword[i++]=rcv;
            LCD_PutChar(rcv);

            if (rcv =='#') { // Check if end of password
               receivedPassword[i] = '\0'; // Null-
terminate the string
               processReceivedData(); // Process
received data
               i = 0; // Reset index for next password
            }

            if(0u != rcv) {
               while(0u == USBUART_CDCIsReady()) {
               }
```

```
                USBUART_PutChar(rcv);

                if(USBUART_BUFFER_SIZE == count) {
                   while(0u ==
USBUART_CDCIsReady()) {
                   }
                   USBUART_PutData(NULL, 0u);
                }
             }
          }
       }
    }
}
```

2. MATLAB GUI Code:-

The MATLAB GUI provides a graphical interface for users to input passwords and control the lock mechanism. Below is the MATLAB code for creating the GUI interface:

```
function combined_gui()
   % Create the main figure
   fig = figure('Name', 'Combined GUI',
'NumberTitle', 'off', ...
      'Position', [100, 100, 400, 500], 'MenuBar',
'none', ...
      'ToolBar', 'none', 'Resize', 'off');

   % Create an input field to display the pressed
digits
   input_field = uicontrol('Style', 'edit', 'Position',
[20, 450, 360, 30]);

   % Create buttons for digits 0-9
   for digit = 1:9
      uicontrol('Style', 'pushbutton', 'String',
num2str(digit), ...
         'Position', [20 + 90 * mod(digit-1, 3), 350 -
70 * floor((digit-1) / 3), 70, 70], ...
         'Callback', @(~,~)
update_input_field(input_field, digit));
   end
```

```
   uicontrol('Style', 'pushbutton', 'String',
num2str("*"), ...
         'Position', [20 + 90 * mod(9, 3), 350 - 70 *
floor(9 / 3), 70, 70], ...
         'Callback', @(~,~)
update_input_field(input_field, "*"));

   uicontrol('Style', 'pushbutton', 'String',
num2str(0), ...
         'Position', [20 + 90 * mod(10, 3), 350 - 70 *
floor(10 / 3), 70, 70], ...
         'Callback', @(~,~)
update_input_field(input_field, 0));

   uicontrol('Style', 'pushbutton', 'String',
num2str("#"), ...
         'Position', [20 + 90 * mod(11, 3), 350 - 70 *
floor(11 / 3), 70, 70], ...
         'Callback', @(~,~)
update_input_field(input_field, "#"));

   % Create a clear button
   uicontrol('Style', 'pushbutton', 'String', 'Clear',
...
```

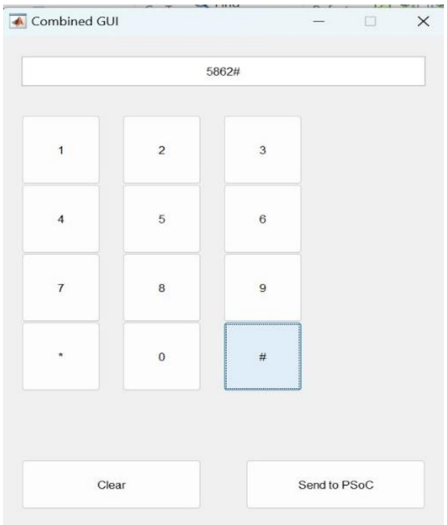
```
      'Position', [20, 20, 160, 50], 'Callback', @(~,~)
clear_input_field(input_field));

   % Create a send button for sending text to PSoC
   uicontrol('Style', 'pushbutton', 'String', 'Send to
PSoC', ...
      'Position', [220, 20, 160, 50], 'Callback',
@(~,~) send_to_psoc(input_field));

   % Serial port configuration
   s = serialport('COM6', 9600);
   configureTerminator(s, "LF");

   % Callback function to send text to PSoC
   function send_to_psoc(input_field)
      % Get the text from the input field

      for i=1:5
         text = get(input_field, 'String');
      % Send text to PSoc
         write(s, text, "string");
      end

      % Update status label
      disp(['Text sent to PSoC: ' text]);
   end

   % Callback function to update the input field
with pressed digits
   function update_input_field(input_field, digit)
      current_text = get(input_field, 'String');
      new_text = [current_text num2str(digit)];
      set(input_field, 'String', new_text);
   end

   % Callback function to clear the input field
   function clear_input_field(input_field)
      set(input_field, 'String', '');
   end
end
```

Results:

Upon completion of the project, we successfully implemented a digital combinational lock with the following key features:

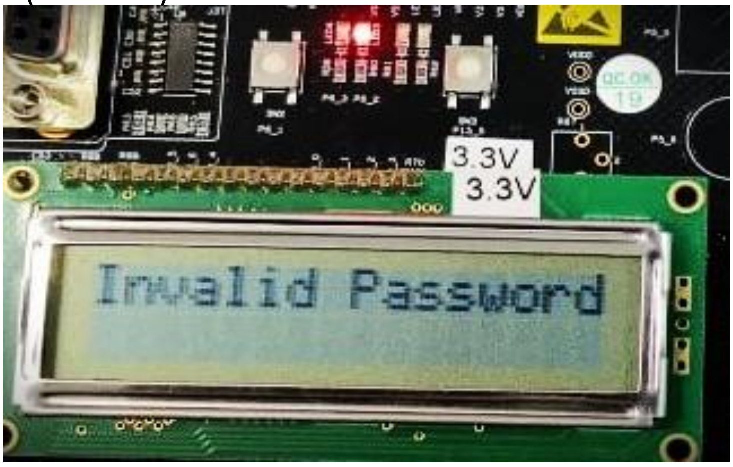
MATLAB GUI interface for password input
Serial communication between MATLAB and the microcontroller
Password authentication mechanism
Control of hardware components such as LEDs and an LCD screen
Secure locking/unlocking functionality

Fig1: Taking Input from MATLAB



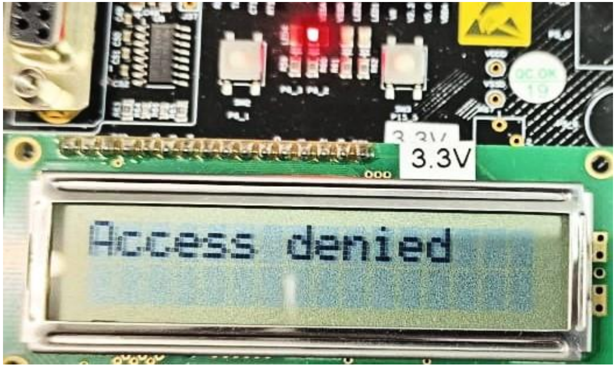
5862#

Fig2: If input password is wrong (ON LCD)



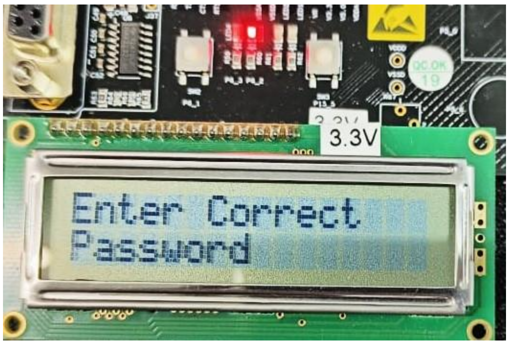
Invalid Password

Fig3: If input password is wrong (ON LCD)



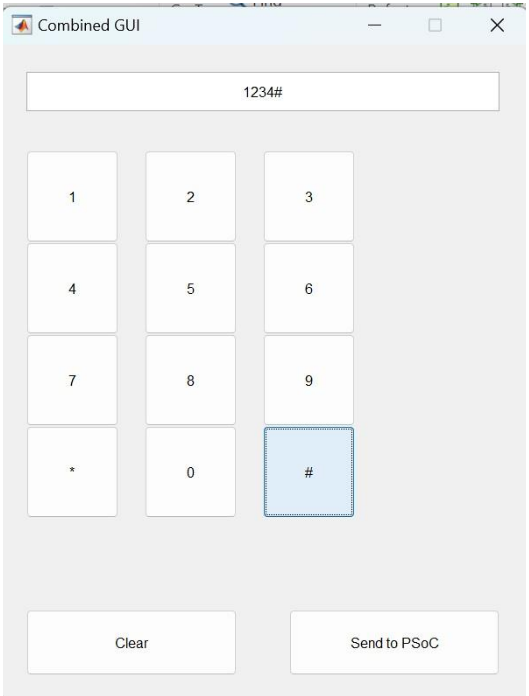
Access denied

Fig4: If input password is wrong (ON LCD)



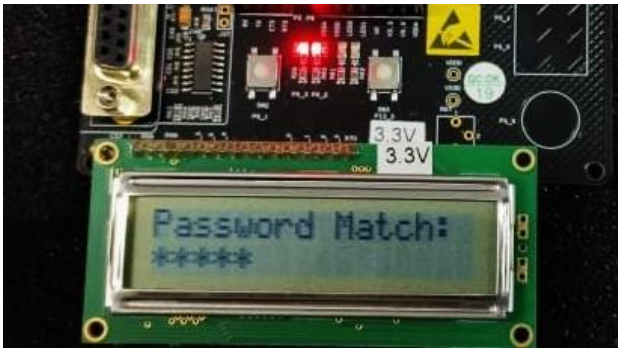
Enter Correct
Password

Fig5: The correct password input



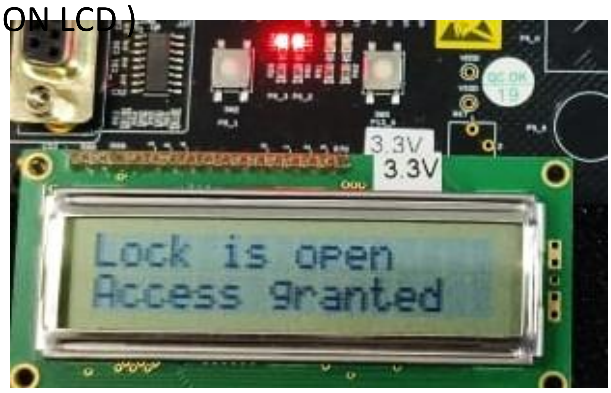
1234#

Fig6: If input password is correct (ON LCD)



Password Match:
*****

Fig7: If input password is correct (ON LCD)



Lock is open
Access granted

Discussion:-

The Digital Combinational Lock project represents a successful integration of hardware and software components to create a secure and user-friendly locking mechanism. By leveraging microcontroller technology, MATLAB graphical user interface (GUI), and serial communication, we have developed a versatile solution for controlling access to secured areas or assets.

Key Achievements:

Secure Locking Mechanism: The implementation of a digital combinational lock ensures reliable access control by requiring users to input a predefined password sequence for authentication.

User-Friendly Interface: The MATLAB GUI provides an intuitive interface for users to input passwords conveniently, enhancing the overall user experience and accessibility.

Visual Feedback: Hardware components such as LEDs and an LCD screen offer visual feedback to users, indicating the status of the lock (e.g., locked or unlocked) and providing feedback on password input.

Serial Communication: Bidirectional serial communication between the MATLAB GUI and the microcontroller facilitates seamless data exchange, enabling password transmission and status updates in real-time.

Future Considerations:

While the current implementation meets the project objectives, several areas for future improvement and expansion can be explored:

Enhanced Security Features: Integration of additional security measures such as encryption for password transmission and biometric authentication can further enhance the security of the system against unauthorized access attempts.

User Interface Enhancements: Continued refinement of the MATLAB GUI interface, including the addition of interactive elements and improved visual design, can enhance user engagement and usability.

Error Handling and Robustness: Implementation of error handling mechanisms and robust serial communication protocols can improve the reliability and resilience of the system, especially in challenging environments or under adverse conditions.

Conclusion:-

In conclusion, the Digital Combinational Lock project demonstrates the successful implementation of a secure and user-friendly locking mechanism through the integration of hardware and software components. By combining microcontroller technology with MATLAB GUI development, we have created a versatile solution that offers reliable access control while prioritizing user experience and convenience.

As technology continues to evolve, the Digital Combinational Lock project serves as a foundation for further innovation in access control systems, with the potential for broader applications in security, automation, and beyond.