MIT
MANIPAL

EMBEDDED SYSTEMS

DIGITAL VAULT USING LPC1768

~TEAM 8

TABLE OF CONTENTS

PREFACE

Dear reader,

It is with great pleasure that we present this project to you. Through this project, we aim to justify the implemenation of LPC1768 by creating a digital vault.

We have spent countless hours researching and analyzing the information presented in this document. We would like to express our gratitude to Dr. Prashant Barla, for guiding us through the subject and making us capable enough to complete the project.

Thank you for taking the time to read this project.

INTRODUCTION

The LPC1768 digital vault is a system that uses an LPC1768 microcontroller to securely store and retrieve a PIN, which can only be accessed by entering the correct password. The system also includes a keyboard and a LCD display.

This project utilizes the LPC1768 microcontroller, keyboard and LCD to create a digital vault that can securely store the PIN and can be accessed using a password entered through a keyboard.

The LPC1768 microcontroller is an ideal choice for this project due to its low power consumption, small form factor, and high processing power. The keyboard and LCD display will be interfaced with the microcontroller using GPIO pins.
The system should be reliable, secure, power-efficient, and cost-effective.

OBJECTIVE

The LPC1768 digital vault is a system that uses an LPC1768 microcontroller to securely store and retrieve a PIN, which can only be accessed by entering the correct password.

The user interacts with the system by entering a password using the keyboard. The entered password is then displayed on the LCD display. If the entered password matches the stored password, the user is prompted if they want to display PIN or change PIN, indicating that access has been granted.

If the password is incorrect, the LCD will show a message indicating that the access has been denied, along with the number of attempts available. When the user enters invalid password 5 times consecutively, the vault will be locked.
A master key will be required to unlock the vault.

OBJECTIVE

The system architecture consists of the LPC1768 microcontroller, which is responsible for managing the inputs and outputs of the system. The keyboard is used to enter the password, which is read by the microcontroller. The entered password is then compared to the stored password, which is stored in the microcontroller's memory.

The LCD display is used to show the entered password and any error messages that may occur. The system also includes a power-efficient design to ensure long battery life.

Overall, the LPC1768 digital vault provides a secure and reliable way to store and retrieve a PIN. It is easy to use and cost-effective, making it an ideal solution for personal and commercial applications that require secure data storage.

COMPONENTS USED

01 LPC 1768

The LPC1768 is a 32-bit ARM Cortex-M3 microcontroller designed for embedded applications.

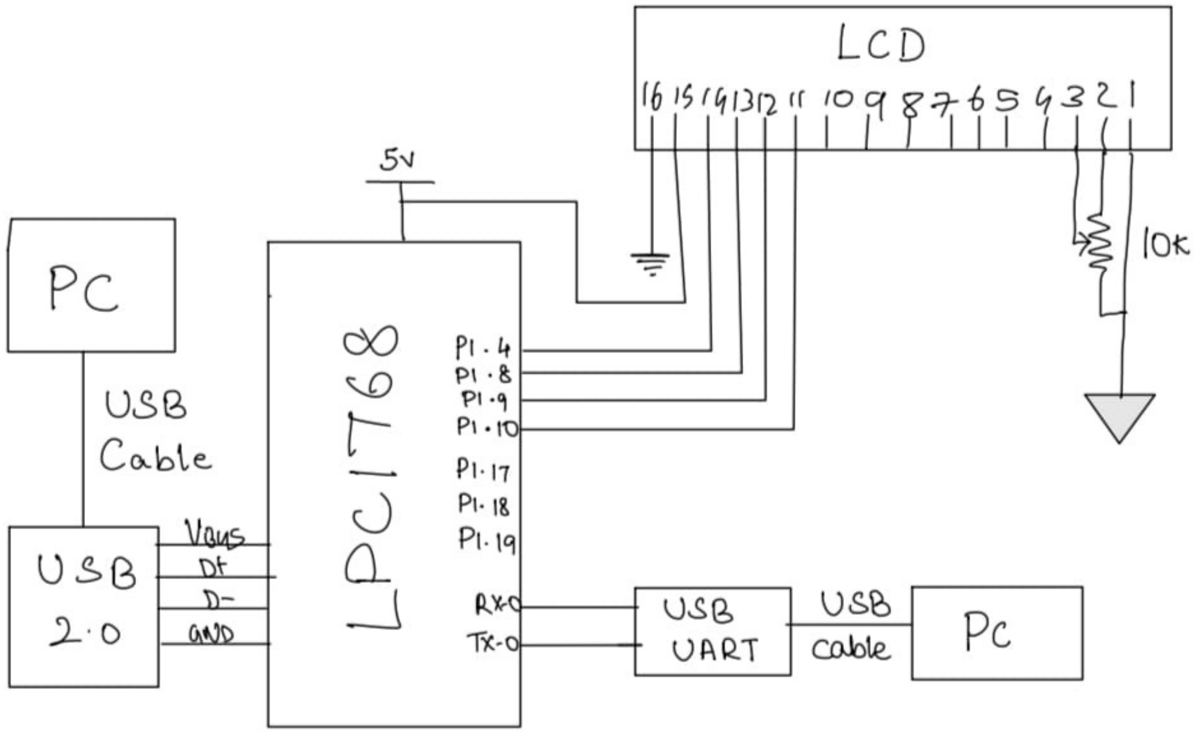
02 Keyboard

The input for the digital vault is taken through the keyboard.

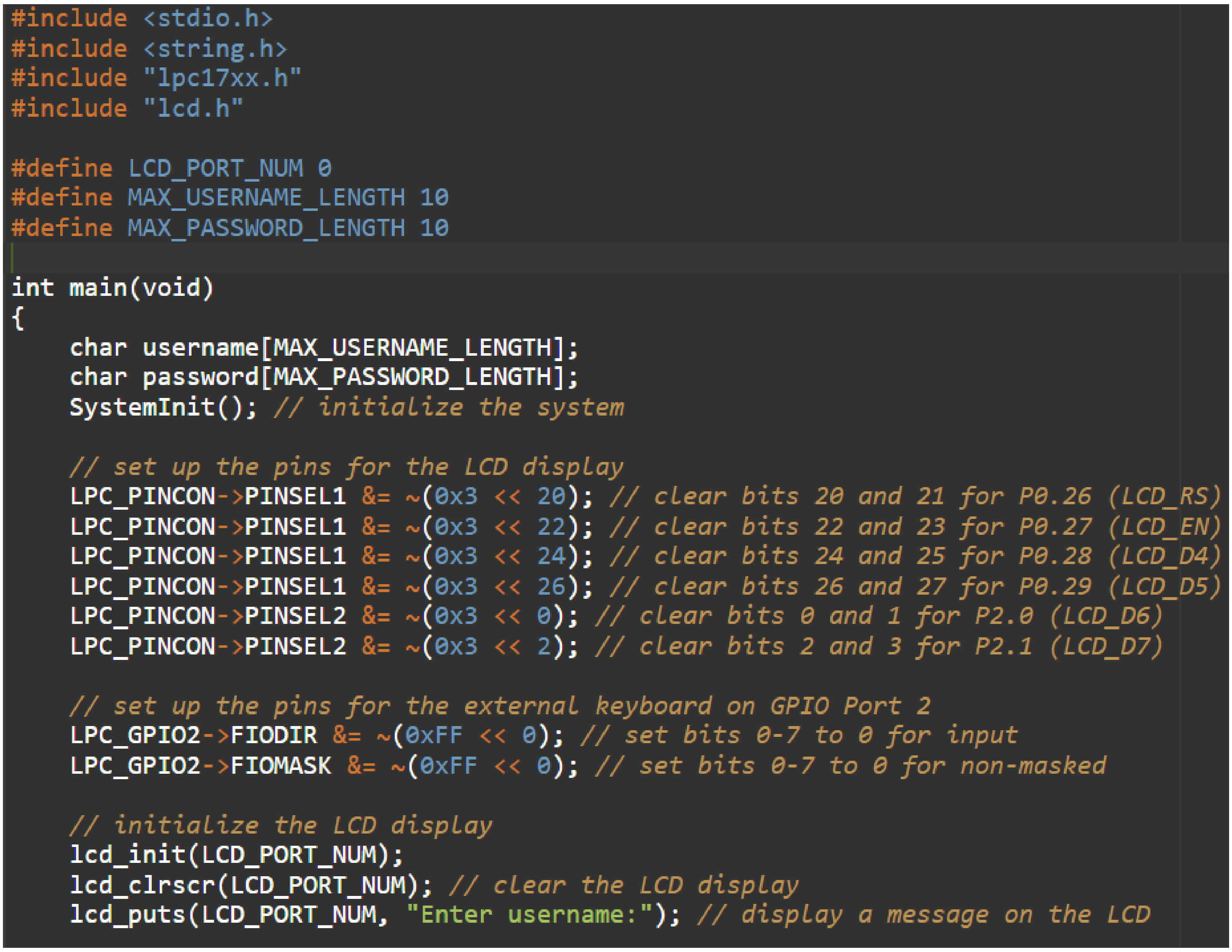
03 LCD Display

The prompt and the output is displayed via LCD Display.

BLOCK DIAGRAM

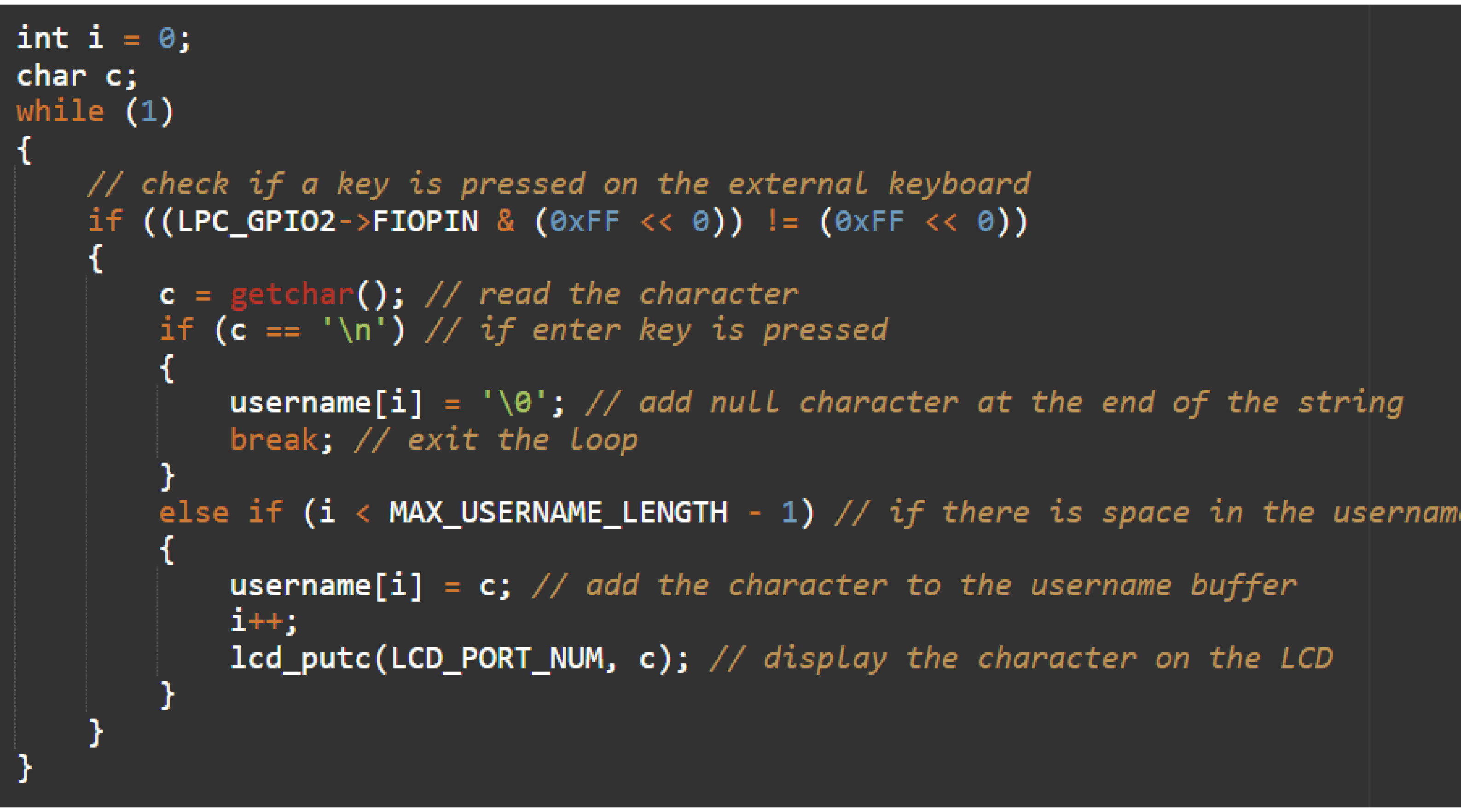


CODE



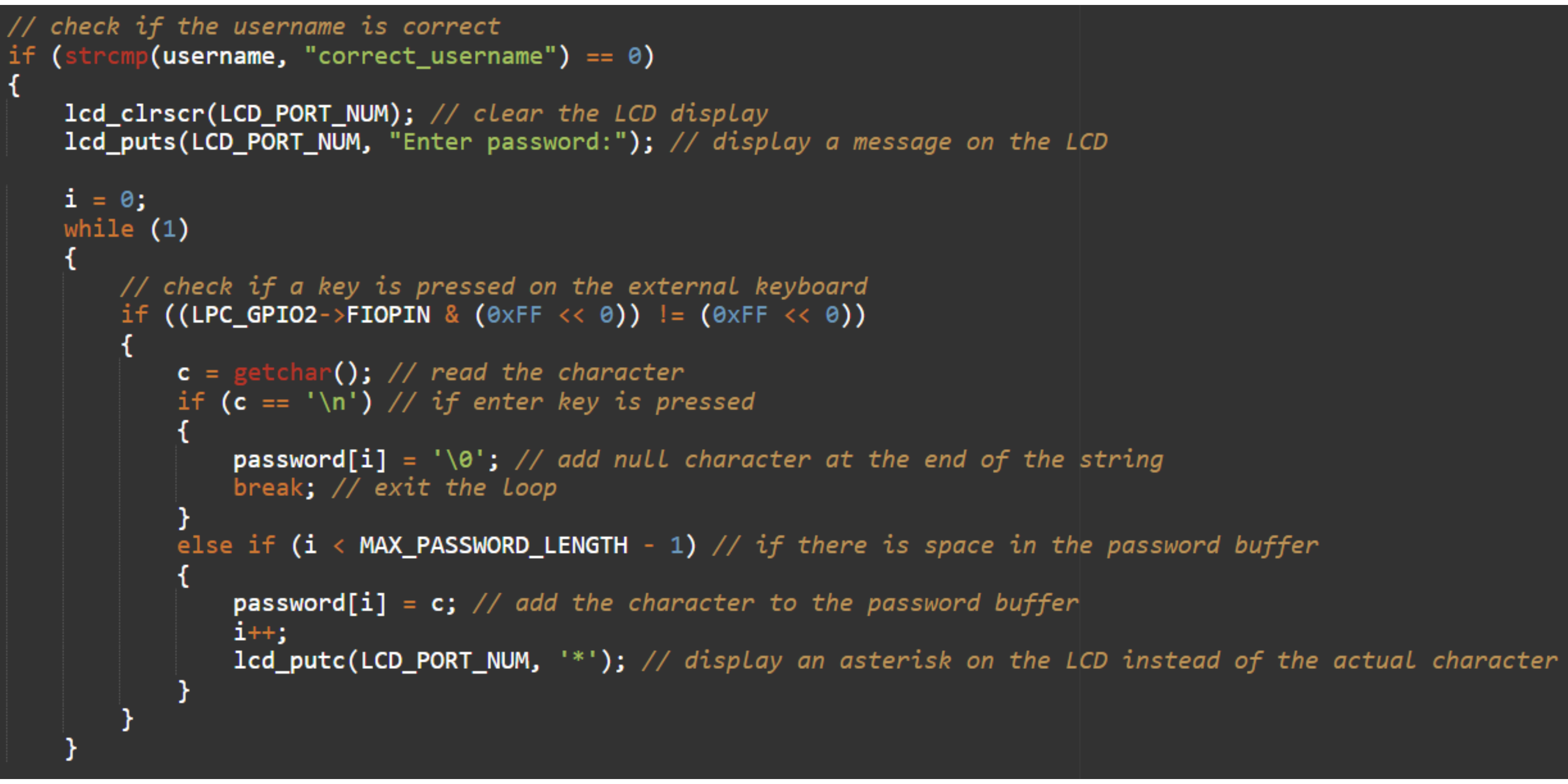
```c
#include <stdio.h>
#include <string.h>
#include "lpc17xx.h"
#include "lcd.h"

#define LCD_PORT_NUM 0
#define MAX_USERNAME_LENGTH 10
#define MAX_PASSWORD_LENGTH 10

int main(void)
{
    char username[MAX_USERNAME_LENGTH];
    char password[MAX_PASSWORD_LENGTH];
    SystemInit(); // initialize the system

    // set up the pins for the LCD display
    LPC_PINCON->PINSEL1 &= ~(0x3 << 20); // clear bits 20 and 21 for P0.26 (LCD_RS)
    LPC_PINCON->PINSEL1 &= ~(0x3 << 22); // clear bits 22 and 23 for P0.27 (LCD_EN)
    LPC_PINCON->PINSEL1 &= ~(0x3 << 24); // clear bits 24 and 25 for P0.28 (LCD_D4)
    LPC_PINCON->PINSEL1 &= ~(0x3 << 26); // clear bits 26 and 27 for P0.29 (LCD_D5)
    LPC_PINCON->PINSEL2 &= ~(0x3 << 0); // clear bits 0 and 1 for P2.0 (LCD_D6)
    LPC_PINCON->PINSEL2 &= ~(0x3 << 2); // clear bits 2 and 3 for P2.1 (LCD_D7)

    // set up the pins for the external keyboard on GPIO Port 2
    LPC_GPIO2->FIODIR &= ~(0xFF << 0); // set bits 0-7 to 0 for input
    LPC_GPIO2->FIOMASK &= ~(0xFF << 0); // set bits 0-7 to 0 for non-masked

    // initialize the LCD display
    lcd_init(LCD_PORT_NUM);
    lcd_clrscr(LCD_PORT_NUM); // clear the LCD display
    lcd_puts(LCD_PORT_NUM, "Enter username:"); // display a message on the LCD
```

CODE



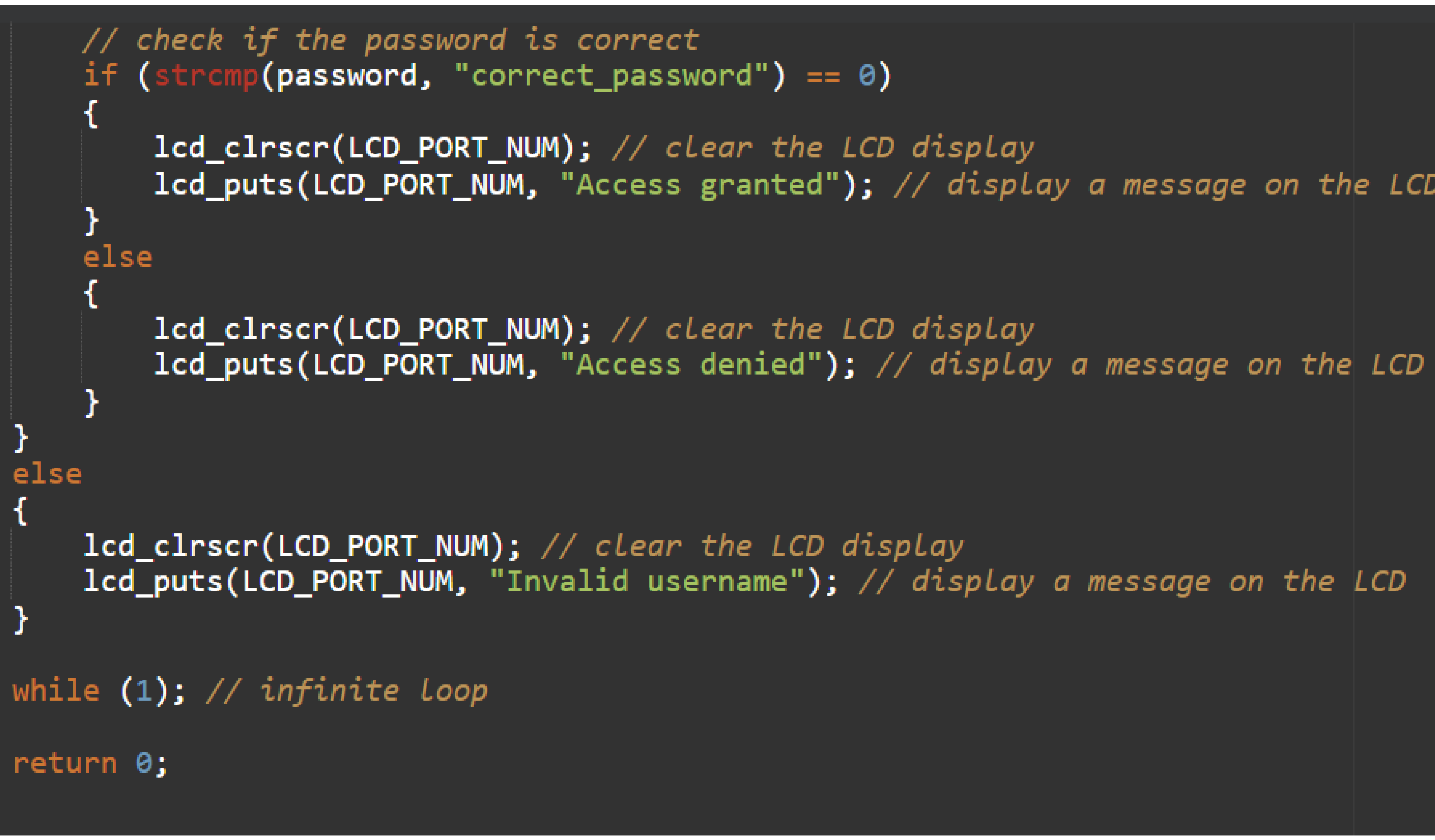
```
int i = 0;
char c;
while (1)
{
    // check if a key is pressed on the external keyboard
    if ((LPC_GPIO2->FIOPIN & (0xFF << 0)) != (0xFF << 0))
    {
        c = getchar(); // read the character
        if (c == '\n') // if enter key is pressed
        {
            username[i] = '\0'; // add null character at the end of the string
            break; // exit the loop
        }
        else if (i < MAX_USERNAME_LENGTH - 1) // if there is space in the usernam
        {
            username[i] = c; // add the character to the username buffer
            i++;
            lcd_putc(LCD_PORT_NUM, c); // display the character on the LCD
        }
    }
}
```

CODE



```c
// check if the username is correct
if (strcmp(username, "correct_username") == 0)
{
    lcd_clrscr(LCD_PORT_NUM); // clear the LCD display
    lcd_puts(LCD_PORT_NUM, "Enter password:"); // display a message on the LCD

    i = 0;
    while (1)
    {
        // check if a key is pressed on the external keyboard
        if ((LPC_GPIO2->FIOPIN & (0xFF << 0)) != (0xFF << 0))
        {
            c = getchar(); // read the character
            if (c == '\n') // if enter key is pressed
            {
                password[i] = '\0'; // add null character at the end of the string
                break; // exit the loop
            }
            else if (i < MAX_PASSWORD_LENGTH - 1) // if there is space in the password buffer
            {
                password[i] = c; // add the character to the password buffer
                i++;
                lcd_putc(LCD_PORT_NUM, '*'); // display an asterisk on the LCD instead of the actual character
            }
        }
    }
```

CODE



```
    // check if the password is correct
    if (strcmp(password, "correct_password") == 0)
    {
        lcd_clrscr(LCD_PORT_NUM); // clear the LCD display
        lcd_puts(LCD_PORT_NUM, "Access granted"); // display a message on the LC
    }
    else
    {
        lcd_clrscr(LCD_PORT_NUM); // clear the LCD display
        lcd_puts(LCD_PORT_NUM, "Access denied"); // display a message on the LCD
    }
}
else
{
    lcd_clrscr(LCD_PORT_NUM); // clear the LCD display
    lcd_puts(LCD_PORT_NUM, "Invalid username"); // display a message on the LCD
}

while (1); // infinite loop

return 0;
```

TEAM MEMBERS

G GAUTAM DATTA

210905394

60

ARYAN SINGH

210905099

17

NIKUNJ AGRAWAL

210905155

28

KSHITI SHETTY

210905137

25

SAMIK PUJARI

210905380

56

BIBLIOGRAPHY

- https://www.keil.com/dd/chip/4868.htm#:~:text=The%20NXP%20(founded%20by%20Philips,Interrupt%20Controller%2C%20Eight%20channel%20General
- https://wglabz.in/16x2-lcd-with-lpc1768-arm-microcontroller-in-depth/
- https://os.mbed.com/handbook/USBHostKeyboard