دانشگاه صنعتی اصفهان

دانشکده مهندسی برق و کامپیوتر

عنوان: پروژه پایانترم درس ریزپردازنده

OTP Token Device

نام و نام خانوادگی: ملوین مختاری

شماره دانشجویی: ۹۸۳۱۱۴۳

نیم سال تحصیلی: پاییز ۱۴۰۰

مدرّس: سرکار خانم دکتر فرزانه شایق

فهرست مطالب

۱ مقدمه

برای حل مشکلات مختلفی که در زمینه‌های گوناگون فناوری اطلاعات رخ می‌دهد، راه حل‌های متفاوتی ارائه گردیده است. یکی از راه حل‌های ارائه شده که میزان استفاده از آن به طور چشمگیری در حال رشد می‌باشد، استفاده از وسایل یا ماژول های امنیتی است. توکن‌های هوشمند یواس‌بی به عنوان نسل جدیدی از وسایل امنیتی، مشکل هزینه و قیمت بالای سخت افزارهای امنیتی را حل نموده و به وسایلی فراگیر تبدیل شده‌اند. نام دیگر توکن "رمزیاب" می‌باشد.

توکِن امنیتی سخت‌افزاری کوچک است که برای ورود کاربر یک سرویس رایانه‌ای به سامانه به‌کار می‌رود. به عبارت دیگر، این دستگاه یک دستگاه فیزیکی است که در اختیار کاربران مجاز قرار می‌گیرد تا به راحتی بتوانند برای استفاده از یک سیستم کامپیوتری هویت آن‌ها تشخیص داده شود. توکن امنیتی برای اثبات هویت فرد به صورت الکترونیکی استفاده می‌شود (به عنوان مثال نحوه دسترسی به حساب بانکی از راه دور). به علاوه از توکن به جای رمز عبور معمولی برای احراز هویت مشتری که خواهان ورود به سیستم است، بهره می‌برند. به عبارت دیگر به عنوان یک کلید الکترونیکی برای دسترسی عمل می‌کند.

بعضی از توکن‌ها کلیدهای رمزنگاری مانند پسورد را در حافظه خود ذخیره می‌کنند. این توکن‌ها شامل کلید برای وارد کردن شماره شخصی شناسایی و آغاز برنامه توکن برای انجام عملیات ایجاد رمز عبور هستند.

گستره عملکرد و استفاده از این دستگاه بسیار وسیع است و در حال حاضر سازمان املاک و اسناد کشور،کانون سردفتران و دفتریاران، برخی بانک‌ها و ... برای شناسای افراد و مشتریانشان از این سیستم استفاده می‌کنند که امنیت بالایی دارد.

۲ آشنایی با پروژه

با گسترش استفاده از بانکداری الکترونیک و همه گیر شدن آن این روزها بحث امنیت و حفظ اطلاعات و اسرار مشتریان بانک ها اهمیت دوچندانی یافته است.

از آنجا که بانک های متعدد در حال رقابت برای جذب تعداد مشتریان بیشتر و حفظ مشتریانشان هستند انتظار می رود در زمینه جلب اعتماد و امنیت اطلاعات مشتریانشان کوشا باشند.

یکی از راهکارها استفاده از دستگاه های مدرن و به روز است که می توانند در این عرصه تاثیر گذار باشند.خوشبختانه تا کنون برخی از بانک های کشور مبادرت به عرضه دستگاه های رمزیاب کرده اند که سالهاست در نظام بانکداری کشور های جهان امتحان خود را پس داده اند.

با ایده گرفتن از ماهیت و کاربرد های این دستگاه، من به ساخت و طراحی یک دستگاه OTP توکن برای پرداخت های مالی مشتریان یک بانک فرضی پرداختم. این دیوایس، شامل یک صفحه کلید و یک نمایشگر LCD و به عنوان مهمترین قطعه، یک میکروکنترلر (در مورد من ATMEGA32) می باشد.

نحوه کار با دستگاه بسیار ساده و روان است به نحوی که مشتریان بانک ما اعداد را به وسیله صفحه کلیدی که در کنار صفحه نمایش وجود دارد، وارد می‌کنند و سپس شماره شناسایی شخصی یا PIN code برای ورود به توکن را زده و سپس وارد مود سیستمی مورد نظر خواهند شد.

۳ مد های سیستمی

توکن MMMokhtari، دارای چهار مود سیستمی مختلف شامل ایجاد پسورد یکبار مصرف، تعویض رمز ثابت دستگاه، غیر فعال سازی کارت بانکی و منوی خروج می باشد، که در ادامه به بررسی هر کدام میپردازم:

۱.۳ منوی اول: ایجاد رمز پویا

طبق قوانین بانک ما، به منظور استفاده از کارت بانکی خود برای هر گونه تراکنشی، لازم است که از رمز پویا استفاده کنید. برای دریافت این رمز، پس از روشن کردن دستگاه، منوی شماره یک را مشاهده میکنید که همان ساخت پسورد است.
با فشار دادن کلید یک، رمز پیشفرض دستگاه که ۱۲۳۴ است را وارد میکنید و با زدن کلید x، این رمز را تایید میکنید. پس از آن بر روی نمایشگر یک رمز پویا ساخته شده و به مدت ۲ ثانیه قابل رویت است.

۲.۳ منوی دوم: تعویض رمز ثابت مشتری

در ابتدای ساخت و برنامه نویسی توکن های شخصی سازی شده برای هر مشتری، یک رمز ایستا وجود دارد که من آن را ۱۲۳۴ گذاشتم. به منظور تعویض این رمز ثابت لازم است بعد از رویت منوی اول، با فشار دادن کلید + ، به منوی دوم رفته و آنجا است که با منوی تعویض رمز ثابت مشتری مواجه می شویم.
با فشار دادن کلید ۲، به صفحه وارد کردن رمز ثابت هدایت می شوید که در آنجا باید رمز ثابت قدیمی را وارد نمایید و در ادامه کلید x را فشار دهید. اکنون باید رمز جدید را وارد کرده و در نظر داشته باشید که در بعد فشار دادن کلید x این رمز، رمز جدید دستگاه است و در صورت استفاده از رمز اشتباه، پیام "رمز اشتباه است" نمایش داده می شود، به همین دلیل در انتخاب رمز مناسب و به خاطر سپردن آن دقت فرماید.

۳.۳ منوی سوم: غیر فعال سازی کارت بانکی

اگر مشتری تمایل داشت کارت خودش را غیر فعال کند با فشار دادن کلید ۳ در منوی اصلی، با پرسشی مواجه می شود که آیا مطمئن است که میخواهد کارتش را غیر فعال نماید یا خیر و او با انتخاب گزینه صحیح، از انتخاب گزینه اشتباه جلوگیری میکند و در واقع در این منو یک سیستم نهفته تایید دو مرحله ای نیز وجود دارد.
با انتخاب گزینه مورد نظر، کارت مشتری غیر فعال میشود یا صرفا بدون تغییر باقی خواهد ماند.

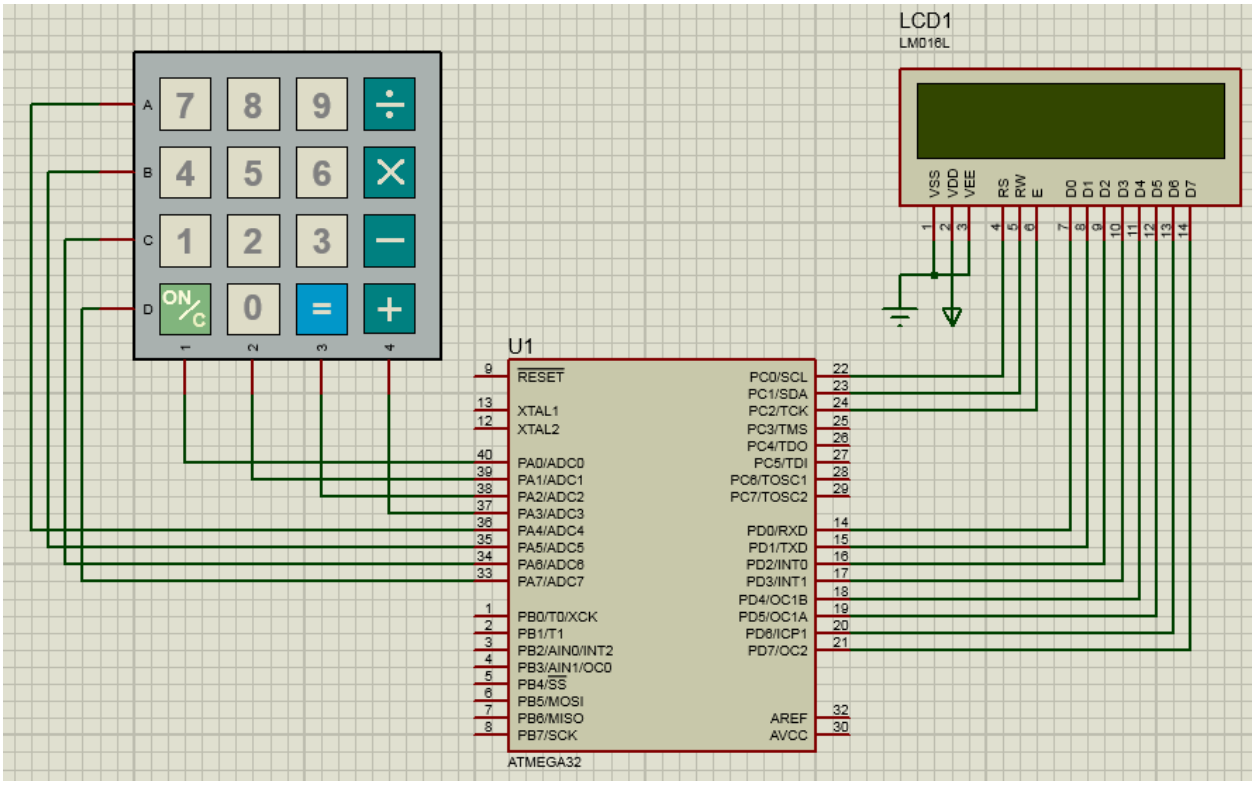
۴.۳ منوی چهارم: خروج

در این دستگاه به منظور خروج امن، یک گزینه برای خروج تعبیه شده است که کاربر با زدن دکمه ۴ میتواند آن را فعال نماید. بدین ترتیب پس از فشرده شدن آن کلید دستگاه خاموش میشود. به منظور روشن سازی دوباره دستگاه صرفا فشردن کلید x کفایت میکند و به محض فشرده شدن این کلید، دستگاه دوباره شروع به کار میکند.

- لازم به ذکر است که به منظور حفظ انرژی و صرفه جویی در مصرف با استفاده از وقفه ها ،سیستمی به منظور خاموش شدن نمایشگر در صورت عدم استفاده از دستگاه پس از چند ثانیه طراحی شده است که دستگاه را به حالت Sleep میبرد و به طریق مشابه با فشردن کلید x دستگاه از همان نقطه پیشین شروع به کار میکند.

۴ تصویر شبیه سازی

نمونه شبیه سازی شده از این دستگاه در نرم افزار Proteus Design Suite به شکل زیر قابل مشاهده است:



- لازم به ذکر است که نمونه سخت افزاری نیز در دانشگاه پیاده سازی و ساخته شد.

۵ کد C میکروکنترلر

میکروکنترلر قلب تپنده پروژه من است. جریانی که باعث تپش هر چه دقیق تر این قلب است، کد این پروژه است. وجود یک قطعه کد منسجم، تمیز و بدون نقص برای هر پروژه ای لازم است که خوشبختانه با دلسوزی ها و تلاش های بی بدیل استاد محترم در تفهیم این مفاهیم، قطعه کد من برای این پروژه از این قاعده مستثنی نبوده و همگی این مولفه ها را به خوبی تبیین می کند.
در ادامه کد اجرای پروژه آورده میشود:

```
/*
 * Project-Code.c
 *
 * Created: 1/25/2022 7:14:01 PM
 * Author : Melvin Mokhtari
 */

#define F_CPU 1000000
#include <avr/io.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
#include <string.h>
#include <util/delay.h>
#include <avr/interrupt.h>

#define LCD_DPRT PORTD
#define LCD_DDDR DDRD
#define LCD_DPIN PIND
#define LCD_CPRT PORTC
#define LCD_CDDR DDRC
#define LCD_CPIN PINC
#define LCD_RS 0
#define LCD_RW 1
#define LCD_EN 2

#define KEY_PRT         PORTA
#define KEY_DDR         DDRA
#define KEY_PIN         PINA
```

```
unsigned char keypad[4][4] = {
        {'7','8','9','/'},
        {'4','5','6','*'},
        {'1','2','3','-'},
        {' ','0','=','+'}
};
unsigned char colloc, rowloc;
unsigned char currentPassword[10]="1234";
unsigned char enterdPassword[10];
char enterdPasswordIndex=0;
unsigned char enterdNewPassword[10];
char  enterdNewPasswordIndex=0;


unsigned char * menus[4]={"GENERATE PASS: 1","CHANGE PASS: 2","DISABLE CARD: 3","EXIT: 4"}

char menuIndex=0;
char state=1;

// 1 => show menu
// 2 => show generate menu
// 3 => password enterd correctly
// 4 => show change password menu
// 5 => password enterd correctly
// 6 => disable card
// 7 => display off

// LCD
void lcdCommand(unsigned char);
void lcdData(unsigned char);
void lcdInit();
void lcdGoToxy(unsigned char, unsigned char);
void lcdPrint(unsigned char *);

// key find
```

```
char keyfind();
void showIntro();
void showLoading();
// main menu
void showMenu();
// generate password
void showGenerateMenu();
void handleSubmitGenerateMenu();
// change password
void showChangePasswordMenu();
void handleSubmitChangePasswordMenu();
void showNewChangePasswordMenu();
void handleSubmitNewChangePasswordMenu();
// disable card
void showDisableCardMenu();
// turn display on/off
void displayOn();
void displayOff();
// initial timer
void initialTimer();

int main(void)
{
        lcdInit();
        showIntro();
        _delay_ms(2000);
        showLoading();
        showMenu();
        srand(time(NULL));
        while(1) {
                char key = keyfind();
                if(state == 1){
                        if(key == '1')
                        showGenerateMenu();
                        else if(key == '2')
                        showChangePasswordMenu();
```

```
        else if(key == '3')
        showDisableCardMenu();
        else if(key == '4')
        displayOff();
        else if(key == '+')
        {
                menuIndex=(menuIndex+1)%4;
                showMenu();
        }

}
else if(state == 2){
        if(key == '*')
        {
                showLoading();
                handleSubmitGenerateMenu();
        }
        else
        {
                initialTimer();
                lcdData(key);
                enterdPassword[enterdPasswordIndex]=key;
                enterdPasswordIndex++;
        }
}
else if(state == 4){
        if(key == '*')
        {
                showLoading();
                handleSubmitChangePasswordMenu();
        }
        else
        {
                initialTimer();
                lcdData(key);
                enterdPassword[enterdPasswordIndex]=key;
```

```
                enterdPasswordIndex++;
        }
}
else if(state == 5){
        if(key == '*')
        {
                showLoading();
                handleSubmitNewChangePasswordMenu();
        }
        else
        {
                initialTimer();
                lcdData(key);
                enterdNewPassword[enterdNewPasswordIndex]=key;
                enterdNewPasswordIndex++;
        }
}
else if(state == 6){
        if(key == '1')
        {
                initialTimer();
                lcdPrint("PROCCESS DONE!");
                _delay_ms(1000);
                showMenu();
        }
        else if(key == '2')
        {
                initialTimer();
                lcdPrint("PROCESS FAILED!");
                _delay_ms(1000);
                showMenu();
        }
}
else if(state == 7){
        if(key == '*')
        {
```

```
                                displayOn();
                        }
                }
        }
}

void lcdCommand(unsigned char command)
{
        LCD_DPRT=command;
        LCD_CPRT &= ~(1<<LCD_RS);
        LCD_CPRT &= ~(1<<LCD_RW);
        LCD_CPRT |= (1<<LCD_EN);
        _delay_us(1);
        LCD_CPRT &= ~ (1<<LCD_EN);
        _delay_us(100);
}

void lcdData(unsigned char data)
{
        LCD_DPRT=data;
        LCD_CPRT |= (1<<LCD_RS);
        LCD_CPRT &= ~(1<<LCD_RW);
        LCD_CPRT |= (1<<LCD_EN);
        _delay_us(1);
        LCD_CPRT &= ~ (1<<LCD_EN);
        _delay_us(100);
}

void lcdInit()
{
        LCD_DDDR = 0xFF;
        LCD_CDDR = 0xFF;
        LCD_CPRT &= ~(1 << LCD_EN);

        _delay_ms(2);
        lcdCommand(0x38); // Initializing to 2 lines & 5x7 font
```

```
        lcdCommand(0x06); // After displaying a character on the LCD, shift cursor to righ
        // lcdCommand(0x0E); // Display on, cursor on
        lcdCommand(0x0C); // Display on, cursor off
        lcdCommand(0x01); // Clear display screen
        _delay_ms(2);
}

void lcdGoToxy(unsigned char x, unsigned char y)
{
        unsigned char firstCharAdr[]={0x80,0xC0,0x94,0xD4};
        lcdCommand(firstCharAdr[y-1] + x - 1);
        _delay_us(100);
}

void lcdPrint(unsigned char *string)
{
        unsigned char i = 0;
        while (string[i]!=0)
        {
                lcdData(string[i]);
                _delay_ms(25);
                i++;
        }
}

char keyfind()
{
        while(1)
        {
                KEY_DDR = 0xF0;           /* set port direction as input-output */
                KEY_PRT = 0xFF;

                do {
                        KEY_PRT &= 0x0F;      /* mask PORT for column read only */
                        asm("NOP");
                        colloc = (KEY_PIN & 0x0F); /* read status of column */
```

```
} while(colloc != 0x0F);

do
{
        do
        {
                _delay_ms(20);              /* 20ms key debounce time */
                colloc = (KEY_PIN & 0x0F); /* read status of column */
                }while(colloc == 0x0F);         /* check for any key press

                _delay_ms (40);              /* 20 ms key debounce time */
                colloc = (KEY_PIN & 0x0F);
        }while(colloc == 0x0F);

        /* now check for rows */
        KEY_PRT = 0xEF;             /* check for pressed key in 1st row */
        asm("NOP");
        colloc = (KEY_PIN & 0x0F);
        if(colloc != 0x0F)
        {
                rowloc = 0;
                break;
        }

        KEY_PRT = 0xDF;          /* check for pressed key in 2nd row */
        asm("NOP");
        colloc = (KEY_PIN & 0x0F);
        if(colloc != 0x0F)
        {
                rowloc = 1;
                break;
        }

        KEY_PRT = 0xBF;          /* check for pressed key in 3rd row */
        asm("NOP");
        colloc = (KEY_PIN & 0x0F);
```

```
                if(colloc != 0x0F)
                {
                        rowloc = 2;
                        break;
                }

                KEY_PRT = 0x7F;         /* check for pressed key in 4th row */
                asm("NOP");
                colloc = (KEY_PIN & 0x0F);
                if(colloc != 0x0F)
                {
                        rowloc = 3;
                        break;
                }
        }

        if(colloc == 0x0E)
        return(keypad[rowloc][0]);
        else if(colloc == 0x0D)
        return(keypad[rowloc][1]);
        else if(colloc == 0x0B)
        return(keypad[rowloc][2]);
        else
        return(keypad[rowloc][3]);
}

void showIntro() {
        lcdGoToxy(1,1);
        lcdPrint(">> MMMokhtari <<");
        lcdGoToxy(1,2);
        lcdPrint(">>My OTP Token<<");
}

void showLoading() {
        char i=0,j=0;
        lcdCommand(0x01);
```

```
        _delay_ms(2);
        lcdGoToxy(17,1);
        lcdPrint("<<<<<");
        lcdGoToxy(17,2);
        lcdPrint("<<<<<");
        for(i=0 ; i<=2 ; i++)
        {
                for(j=0 ; j<=20 ; j++)
                {
                        lcdCommand(0x18);
                        _delay_ms(25);
                }
                lcdCommand(0x02);
                _delay_ms(25);
        }
}

void showMenu() {
        state=1;
        initialTimer();
        lcdCommand(0x01);
        _delay_ms(2);
        lcdGoToxy(1,1);
        lcdPrint(menus[menuIndex]);
        lcdGoToxy(1,2);
        lcdPrint("NEXT: +");
}

void showGenerateMenu() {
        state=2;
        initialTimer();
        lcdCommand(0x01);
        _delay_ms(2);
        lcdGoToxy(1,1);
        lcdPrint("PASSWORD:(*)");
        lcdGoToxy(1,2);
```

```
}

void handleSubmitGenerateMenu(){
        initialTimer();
        lcdCommand(0x01);
        _delay_ms(2);
        lcdGoToxy(1,1);

        if(strcmp(currentPassword, enterdPassword) == 0)
        {
                state=3;
                lcdPrint("YOUR CODE IS:");
                lcdGoToxy(1,2);
                //Generate Temp Pass
                int size =9;
                char str[size];
                const char charset[] = "0123456789";
                if (size) {
                        --size;
                        for (int n = 0; n < size; n++) {
                                int key = rand() % (int) (sizeof charset - 1);
                                str[n] = charset[key];
                        }
                        str[size] = '\0';
                }
                lcdPrint(str);
                memset(enterdPassword, 0, sizeof(enterdPassword));
                enterdPasswordIndex=0;
                _delay_ms(4000);
                showMenu();
        }
        else
        {
                lcdPrint("WRONG PASSWORD!");
                memset(enterdPassword, 0, sizeof(enterdPassword));
                enterdPasswordIndex=0;
```

```
                _delay_ms(1000);
                showMenu();
        }
}

void showChangePasswordMenu(){
        initialTimer();
        state=4;
        lcdCommand(0x01);
        _delay_ms(2);
        lcdGoToxy(1,1);
        lcdPrint("PASSWORD:(*)");
        lcdGoToxy(1,2);
}

void handleSubmitChangePasswordMenu(){
        initialTimer();
        lcdCommand(0x01);
        _delay_ms(2);
        lcdGoToxy(1,1);

        if(strcmp(currentPassword, enterdPassword) == 0)
        {
                showNewChangePasswordMenu();
        }
        else
        {
                lcdPrint("WRONG PASSWORD!");
                memset(enterdPassword, 0, sizeof(enterdPassword));
                enterdPasswordIndex=0;
                _delay_ms(1000);
                showMenu();
        }
}

void showNewChangePasswordMenu(){
```

```
        initialTimer();
        state=5;
        lcdCommand(0x01);
        _delay_ms(2);
        lcdGoToxy(1,1);
        lcdPrint("NEW PASSWORD:(*)");
        lcdGoToxy(1,2);
}

void handleSubmitNewChangePasswordMenu(){
        initialTimer();
        char i=0;
        lcdCommand(0x01);
        _delay_ms(2);
        lcdGoToxy(1,1);
        lcdPrint("PASSWORD CHANGED!");
        memset(currentPassword, 0, sizeof(currentPassword));
        for (i =0 ;i<=9;i++)
        {
                currentPassword[i]=enterdNewPassword[i];
        }
        memset(enterdPassword, 0, sizeof(enterdPassword));
        enterdPasswordIndex=0;
        memset(enterdNewPassword, 0, sizeof(enterdNewPassword));
        enterdNewPasswordIndex=0;
        _delay_ms(1000);
        showMenu();
}

void showDisableCardMenu(){
        initialTimer();
        state=6;
        lcdCommand(0x01);
        _delay_ms(2);
        lcdGoToxy(1,1);
        lcdPrint("SURE?: [Y:1/N:2]");
```

```
        lcdGoToxy(1,2);
}

void displayOn() {
        state=1;
        lcdCommand(0x0C);
        initialTimer();
}

void displayOff() {
        state=7;
        lcdCommand(0x08);
}

void initialTimer(){
        TCNT1=0;
        OCR1A=10000;
        TCCR1A=0x00;
        TCCR1B=0x0D;
        TIMSK=(1<<OCIE1A);
        sei();
}

ISR (TIMER1_COMPA_vect) {
        displayOff();
}
```

۶ کلام آخر

با توجه به شرایط پیچیده و ویژه جهان، عملیات های اینترنتی بخش جدانشدنی از زندگی همه شده است؛ در این میان عملیات هایی که دارای سطح امنیت بالا هستند، احتیاج به احراز هویت دارند تا از ایجاد مشکلات امنیتی جلوگیری کنند. با کمک از این دستگاه رمز ساز یا به اصلاح با استفاده از "MMMokhtari OTP Token" میتوانیم نسبت به مبادله اطلاعات و اسناد خود اطمینان لازم را در بحث امنیت به دست آوریم.