```c
#include <REG52.H>


unsigned int data tick=0; // clock update preskaler
unsigned char data disp=0; // display refresh preskaler
unsigned char data mode=1; // mode selector
unsigned int data ktimeout=0; // KEY ENTER mode timeout
unsigned int data etimeout=0; // ERROR mod timeout
unsigned char data keynum=0; // broj unesenih tastera
unsigned char data char2snd=0; // broj karaktera za slanje
unsigned char data vreme[6]={0x00,0x00,0x80,0x00,0x80,0x00}; // vreme hh.mm.ss unazad!
unsigned char data id[6]={0,0,0,0,0,0}; // id
unsigned char data adminID[6]={100,100,100,100,100,100}; //adminID
unsigned char data next_display=0; // pokazivac na sledeci displej za osvezavanje
unsigned char data key=0; // a key :)
unsigned char data i;
unsigned char data errornum = 0; // broj uzastopnih gresaka

static unsigned char code disp_mask[6]={0x3E,0x3D,0x3B,0x37,0x2F,0x1F}; // kodovi za paljenje odgovarajuce cif
re, s desna na levo
static unsigned char code keyboard[12]={'1','4','7','c','2','5','8','0','3','6','9','o'};
static unsigned char code DISPoff = 0x3F; // 0x3F gasi sve cifre
static unsigned char code mOK = 0xFF;
static unsigned char code mERR = 0x00;
static unsigned char code mACK = 0x0F;
static unsigned char code mRESEND = 0xF0;
bit admin = 0; // admin mod flag ( za unos adminID-a )
sbit GATE = P0^7; // P0.7 daje signal za otvaranje vrata


void main ( void ) using 0 {

// inicijalizacija
    PCON = 0x00;
    IE = 0x94; // IE =1, omoguceni prekidi za: Timer0, Serial Connection, INT1


    TMOD = 0x22; // tajmeri 0 i 1 u modu 2 ( 8bit auto-reload Timer )
    TCON = 0x5C; // Timer0 = enabled, Timer1 = enabled, INT1 on falling edge
    TH0 = 24; // 11056200/(12*(255-24)) - 4000 interapta u sekundi -> T0int na svakih 0.25ms
    TH1 = 253; // 256 - 11056200/(384*9600) = 253 ( za 9600Bd )
    SCON = 0x50; // mod1 ( 8bit UART, variable baud rate )

    while(1) {}
}


void update_clock() {
    vreme[0] = vreme[0] + 1;
    if ( vreme[0] == 0x0A ) {
        vreme[0] = 0;
        vreme[1] = vreme[1] + 1;
        if ( vreme[1] == 0x06 ) {
            vreme[1] = 0;
            vreme[2] = vreme[2] +1;
            if ( vreme[2] == 0x8A ) {
                vreme[2] = 0;
                vreme[3] = vreme[3] + 1;
                if ( vreme[3] == 0x06 ) {
                    vreme[3] = 0;
                    vreme[4] = vreme[4] + 1;
                    if ( vreme[5] == 2 ) {
                        if ( vreme[4] == 0x84) {
                            vreme[4] = 0;
                            vreme[5] = 0;
                        }
                    } else if ( vreme [4] == 0x8A ) {
                        vreme[4] = 0;
                        vreme[5] = vreme[5] + 1;
                    }
                }
            }
        }
    }
}


void refresh() {
    P0 = P0 | DISPoff;
    switch ( mode )
    {
        case 1:
            // CLOCK mode
            P2 = vreme[next_display];
            P0 = disp_mask[next_display];
            break;
```

```c
        case 2:
            // KEY ENTER mode
            if ( next_display<keynum ) {
            P2 = 0x0A;
            P0 = disp_mask[next_display];
            } else {
            P2 = 0x0F;
            P0 = disp_mask[next_display];
            }
            break;
        case 3:
            // ERROR mode
            P2 = 0x0E;
            P0 = disp_mask[next_display];
            break;
    }
    next_display++;
    if (next_display == 6 ) next_display=0;
}


void timer0() interrupt 1 using 1 {
    tick++;
    if ( tick == 4000 ) {
        // prosla 1s, azuriraj sat
        tick = 0;
        update_clock();
    }
    disp++;
    if ( disp == 8 ) {
        // prosle 2ms, osvezi sledecu cifru
        disp = 0;
        refresh();
    }
    if ( mode == 2 ) {
        // sistem je u KEY ENTER modu
        ktimeout++;
        if ( ktimeout == 20000) {
            // proslo 15s, vrati sistem u CLOCK mod
            mode = 1;
            keynum = 0;
            ktimeout = 20000;
            char2snd = 0;
        }
    } else if ( mode == 3 ) {
        // sistem je u ERROR modu
        etimeout++;
        if ( etimeout == 40000 ) {
            // proslo 30s, ugasi ERROR poruku
            etimeout = 40000;
            mode = 1;
        }
    }
}

void usart() interrupt 4 using 3 {
    if ( SCON^0 == 1 ) {
    // data received
        SCON &= 0xFE;
        i = SBUF;
        if ( i == mOK ) {
        // stigla "open door" komanda
        errornum = 0;
        GATE = 0;
        GATE = 1;
        SBUF = mACK;
        mode = 1;
        } else if ( i == mERR )
        {
           // sitigla "ERROR" komanda
            errornum++;
            SBUF = mACK;
            if (errornum > 3 )
                mode = 3;
            if (errornum < 4 )
                mode = 1;
        } else {
            // nije prepoznata komanda
            SBUF = mRESEND;
        };
    } else if ( SCON^1 == 1 ) {
        // Tx buffer ready
        SCON &= 0xFD;
        if ( char2snd > 0 ) {
            // ima nesto za slanje
```

```c
            char2snd--;
            SBUF = ( id[5-char2snd]);
            if ( char2snd == 0 )
                keynum=0;
        }
    }
}


void key_int() interrupt 2 using 2 {
    if ( mode != 3 ) {
    // ako je ERROR mod, ne reaguj na tastere
    ktimeout = 20000; // reset timeout logic
    key = P1;
    if ( key > 0x0B ) {
    // preferences mod
        IE &= 0xDF;
        switch (key) {
            case 12: // podesavanje sekundi
                    vreme[0]  = vreme[0] + 1;
                    if ( vreme[0] == 0x0A ) {
                        vreme[0] = 0;
                        vreme[1] = vreme[1] + 1;
                        if ( vreme[1] == 0x06 )
                        vreme[1] = 0;
                    }
                break;

            case 13: // podesavanje minuta
                    vreme[2] = vreme[2] + 1;
                    if ( vreme[2] == 0x8A ) {
                        vreme[2] = 0;
                        vreme[3] = vreme[3] + 1;
                        if ( vreme[3] == 0x06 )
                        vreme[3] = 0;
                    }
                break;

            case 14: // podesavanje sati
                    vreme[4] = vreme[4] + 1;
                    if ( vreme[5] == 2 ) {
                        if ( vreme[4] == 0x84) {
                            vreme[4] = 0;
                            vreme[5] = 0;
                        }
                    } else if ( vreme [4] == 0x8A ) {
                        vreme[4] = 0;
                        vreme[5] = vreme[5] + 1;
                    }
                break;

            case 15: // taster za unos adminID-a
                    admin = 1; // predji u admin rezim
                    mode = 2; // i ukljuci KEY ENTER mod
                break;
        }
        IE |= 0x40;
    } else {
    key = keyboard[key];

    switch (key)
    {
        case 'c': // pritisnut je taster "c"
            if ( keynum > 0 ) keynum--;
            break;

        case 'o': // pritisnut je taster "ok"
            if ( keynum == 6 ) {
                if ( admin == 1 ) { //sistem je u admin rezimu
                    for ( i = 0; i<6; i++ ) {
                        adminID[i] = id[i]; // postavi novi adminID
                    }
                    keynum = 0; // isprazni bafer
                    admin = 0; // ugasi admin rezim
                    mode = 1; // vrati se u CLOCK mod
                } else {
                    i = 0;
                    while  ( adminID[i] == id[i] )  // id==adminID?
                    { i++; }
                    if ( i < 6 ) { // id != adminID
                        char2snd=5;
                        SBUF = id[0]; // posalji 1., interapt rutina se pokupiti ostale
                    } else { // id == adminID
                        GATE = 0; // otvori vrata ( na uzlaznu ivicu )
                        GATE = 1;
```

```
                        keynum = 0;
                        errornum = 0;
                        mode = 1;
                }
            }
        }
        break;

    default : // pritisnut je taster 0..9
        if ( keynum < 6 )
            id[keynum++]=key;
    break;
    }
    }

}

}
```