```
 1: #include <18f4520.h>
 2: #include "PIC18F4520_registers.h"
 3: #device adc=10
 4:
 5: #FUSES NOWDT                    //No Watch Dog Timer
 6: #FUSES WDT128                    //  watch dog timer on
 7: #FUSES HS                      //High power osc < 200 khz
 8: #FUSES NOPROTECT                //Code not protected from reading
 9: #FUSES BROWNOUT                 //Reset when brownout detected
10: #FUSES BORV25                   //Brownout reset at 2.5V
11: #FUSES NOPUT                    //No Power Up Timer
12: #FUSES NOCPD                    //No EE protection
13: #FUSES STVREN                   //Stack full/underflow will cause rese
14: #FUSES NODEBUG                  //No Debug mode for ICD
15: #FUSES NOLVP                    //No low voltage prgming, B3(PIC16) or
16: #FUSES NOWRT                    //Program memory not write protected
17: #FUSES NOWRTD                   //Data EEPROM not write protected
18: #FUSES IESO                     //Internal External Switch Over mode e
19: #FUSES FCMEN                    //Fail-safe clock monitor enabled
20: #FUSES NOPBADEN                   //PORTB pins are configured as analo
21: #FUSES NOWRTC                   //configuration not registers write pr
22: #FUSES NOWRTB                   //Boot block not write protected
23: #FUSES NOEBTR                   //Memory not protected from table read
24: #FUSES NOEBTRB                  //Boot block not protected from table
25: #FUSES NOCPB                    //No Boot Block code protection
26: #FUSES LPT1OSC                  //Timer1 configured for low-power oper
27: #FUSES MCLR                     //Master Clear pin enabled
28: #FUSES NOXINST                    //Extended set extension and Indexed
29:
30: #use delay(clock=20000000)
31:
32: #define use_portb_lcd TRUE
33: #define C_Timeout 2000
34: #include "EXLCD.C"
35:
36: #use rs232(baud=300,parity=E,xmit=PIN_C6,rcv=PIN_C7,bits=7,stop=1 )
37:
38:
39: int i;
40: char a[200],inds = 0;
41: unsigned int16 RSTimeout=0, ind1;
42:
43: #INT_RDA
44: void RX_isr(void)
45: {
46:    set_uart_speed(300,x);
47:    disable_interrupts(int_rda); // int_rda kesmesini pasif yap
48:
49:    if(inds>199)
50:       inds=199;
51:
52:    output_toggle(pin_c1);
53:    a[inds++]=SSPBUF;
54:    RSTimeout=0;
55:
56: }
57:
58: #INT_TIMER2
59: void T2_isr(void)
60: {
61:
62: if(RSTimeout)
63:       RSTimeout--;
64:    output_toggle(pin_c0);
65:
```

```
 66: }
 67:
 68:
 69:
 70: void main ( )
 71: {
 72: setup_adc_ports( NO_ANALOGS );
 73: enable_interrupts(INT_RDA);
 74: enable_interrupts(INT_TIMER2);
 75: enable_interrupts(GLOBAL);
 76: clear_interrupt(int_rda);
 77:
 78: setup_wdt(WDT_OFF);
 79:    setup_timer_0(RTCC_INTERNAL);
 80:    setup_timer_1(T1_DISABLED);
 81:    setup_timer_2(T2_DIV_BY_16,156,2);
 82:   lcd_init ( );
 83:    delay_ms(100);
 84:
 85:    printf(lcd_putc,"\f    WELCOME");
 86:
 87:
 88:
 89:    WHILE ( TRUE )
 90:  {
 91: if(input(pin_d0)==1) {
 92:
 93:          delay_ms(1000);
 94:
 95:         //Wake up komutu kismi
 96:
 97:         output_high(pin_a0);
 98:          output_high(pin_a1);
 99:
100:         for(i=0;i<20;i++) {
101:          putc(0x15);
102:          putc(0x00);
103:          delay_ms(100);
104:          }
105:           output_low(pin_a1);
106:          output_low(pin_a0);
107:
108:         //Wake up komutu kismi bitisi
109:
110:          delay_ms(1000);   // 1.TX
111:          output_high(pin_a0);
112:          output_high(pin_a1);
113:
114:          putc ( 0x2f );
115:          putc ( 0x3f );
116:          putc ( 0x21 );
117:          putc ( 0x0d );
118:          putc ( 0x0a );
119:
120:          delay_ms(6800);
121:                   // 1.TX BITIMI
122:          output_low(pin_a1);
123:          output_low(pin_a0);
124:
125:          delay_ms(500);
126:
127:          output_high(pin_a1);
128:          output_high(pin_a0);
129:                        // 2.TX
130:          putc (0x06);
```

```
131:          putc (0x30);
132:          putc (0x34);
133:          putc (0x30);
134:          putc (0x0d);
135:          putc (0x0a);
136:
137:          delay_ms(8000);
138:                     // 2.TX BITIMI
139: output_low(pin_a1);
140: output_low(pin_a0);
141:                       }
142:
143:
144: if(inds>0 && !RSTimeout)
145: {
146:
147: printf(lcd_putc,"INSIDE");
148:
149: for(ind1 = 0; ind1 < inds; ind1++)
150:    printf(lcd_putc,"%C",a[ind1]);
151:
152: inds=C_Timeout;
153:
154: }
155:
156:
157:  }
158:
159: }
160:
161:
```