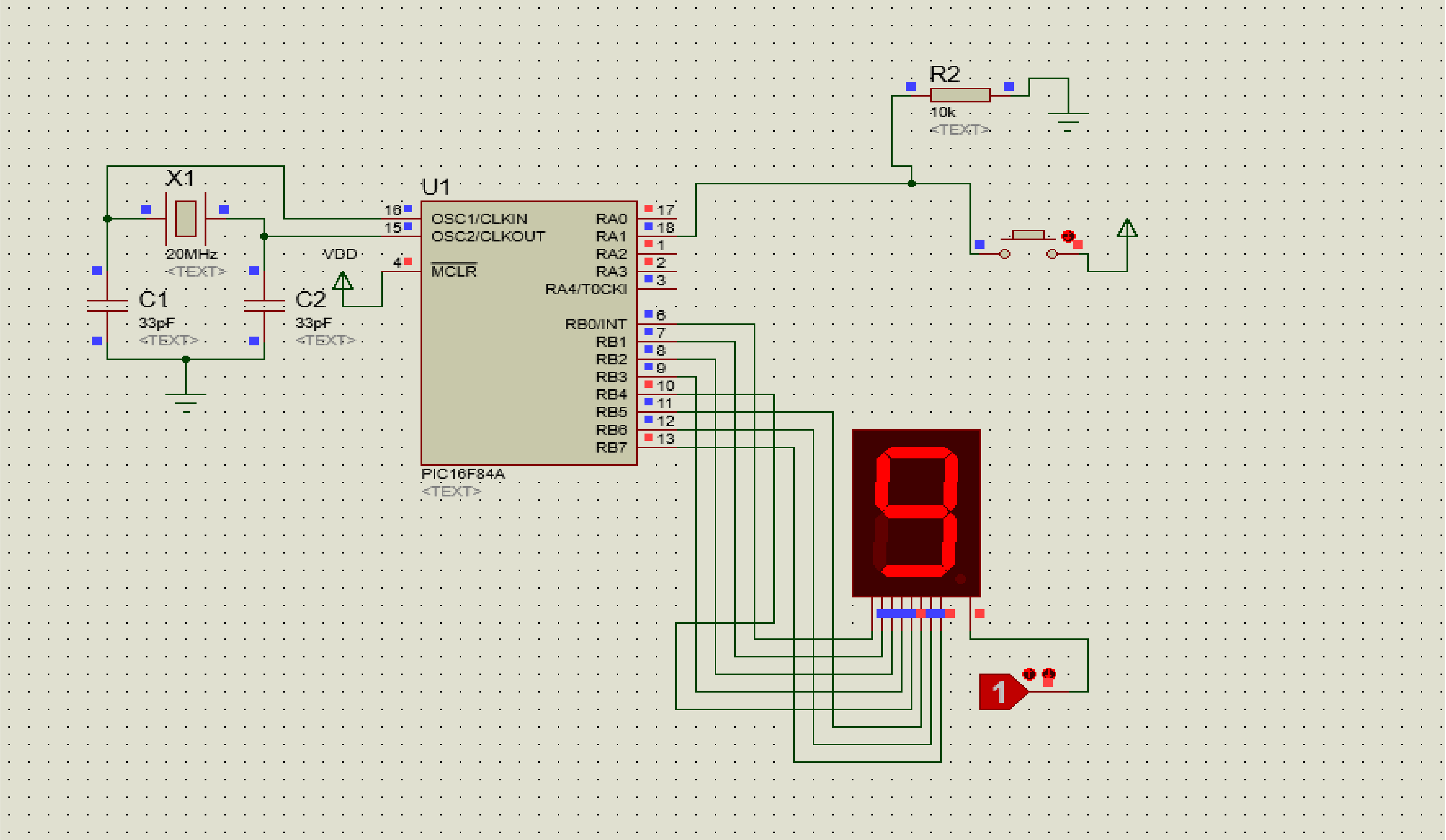
Zadatak 1 - PIC 16F84A i Proteus

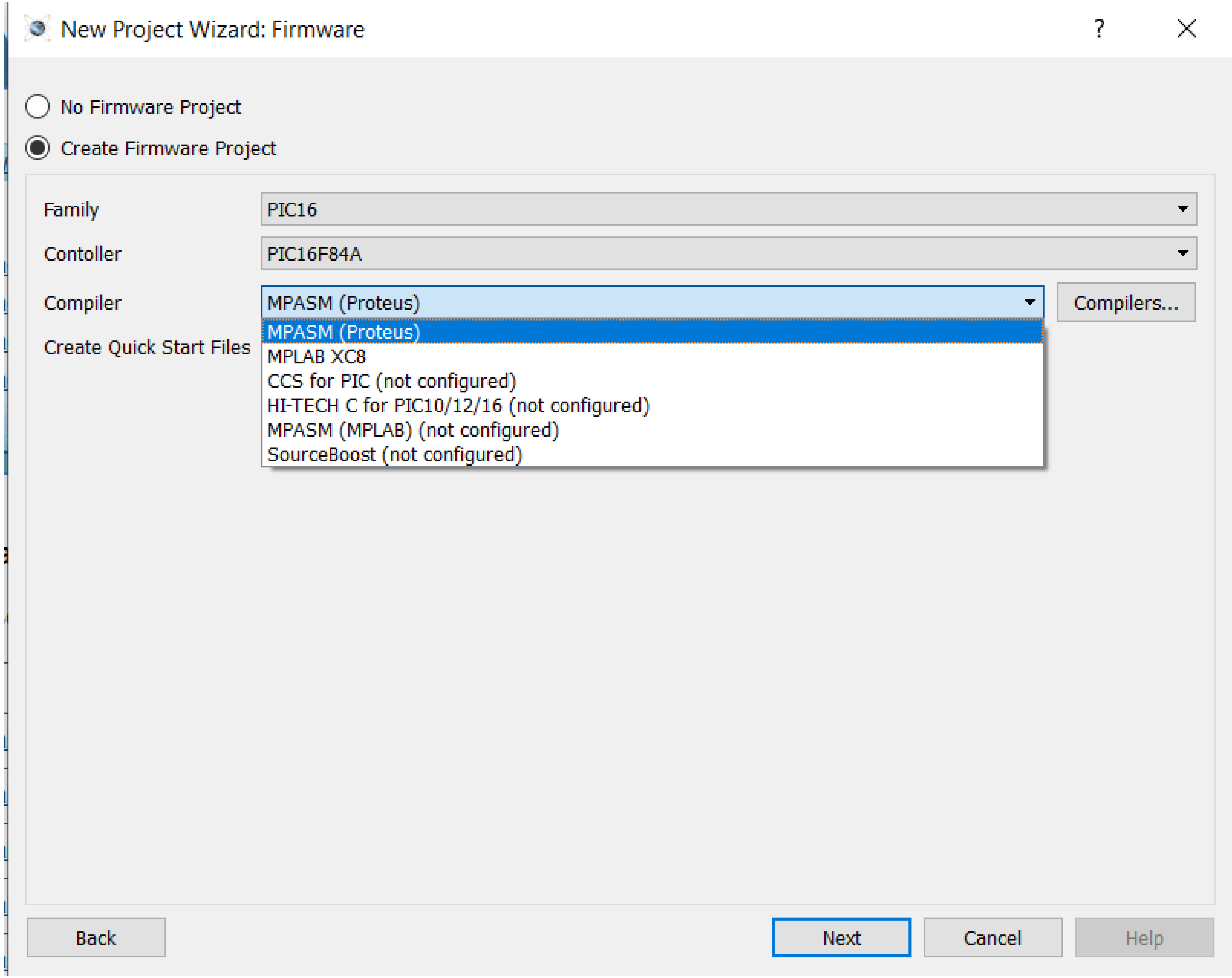


Koraci

- Kreirati firmware projekat
- Odabrati PIC16F84A
- Odabrati programski jezik
    - MPASM – PIC asembler
    - MPLAB XC8 - C
- Oscilatorno kolo
- Podesiti takt PIC-a

Proteus

- Firmware project
- MPASM – assembler
- MPLAB XC8 - C



New Project Wizard: Firmware

○ No Firmware Project

● Create Firmware Project

Family: PIC16

Contoller: PIC16F84A

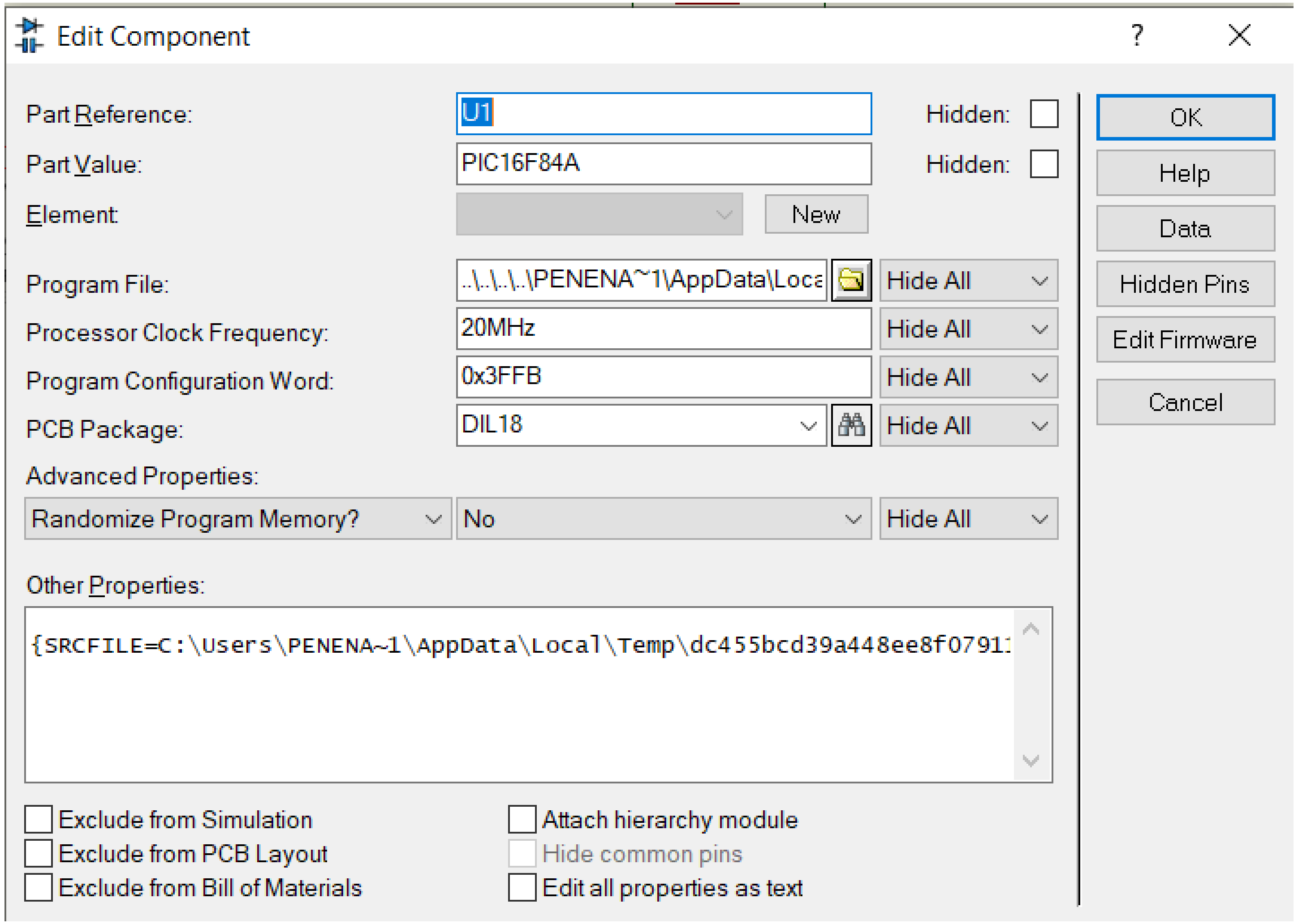
Compiler: MPASM (Proteus)

Create Quick Start Files

?
MPASM (Proteus)
MPLAB XC8
CCS for PIC (not configured)
HI-TECH C for PIC10/12/16 (not configured)
MPASM (MPLAB) (not configured)
SourceBoost (not configured)
Compilers...
Back
Next
Cancel
Help

Podesiti frekvenciju clock-a

- Desni klik na PIC
- Edit Properties



Edit Component

Part Reference: U1 Hidden: ☐

Part Value: PIC16F84A Hidden: ☐

Element:

Program File: ..\..\..\PENENA~1\AppData\Loca Hide All

Processor Clock Frequency: 20MHz Hide All

Program Configuration Word: 0x3FFB Hide All

PCB Package: DIL18 Hide All

Advanced Properties:

Randomize Program Memory? No Hide All

Other Properties:

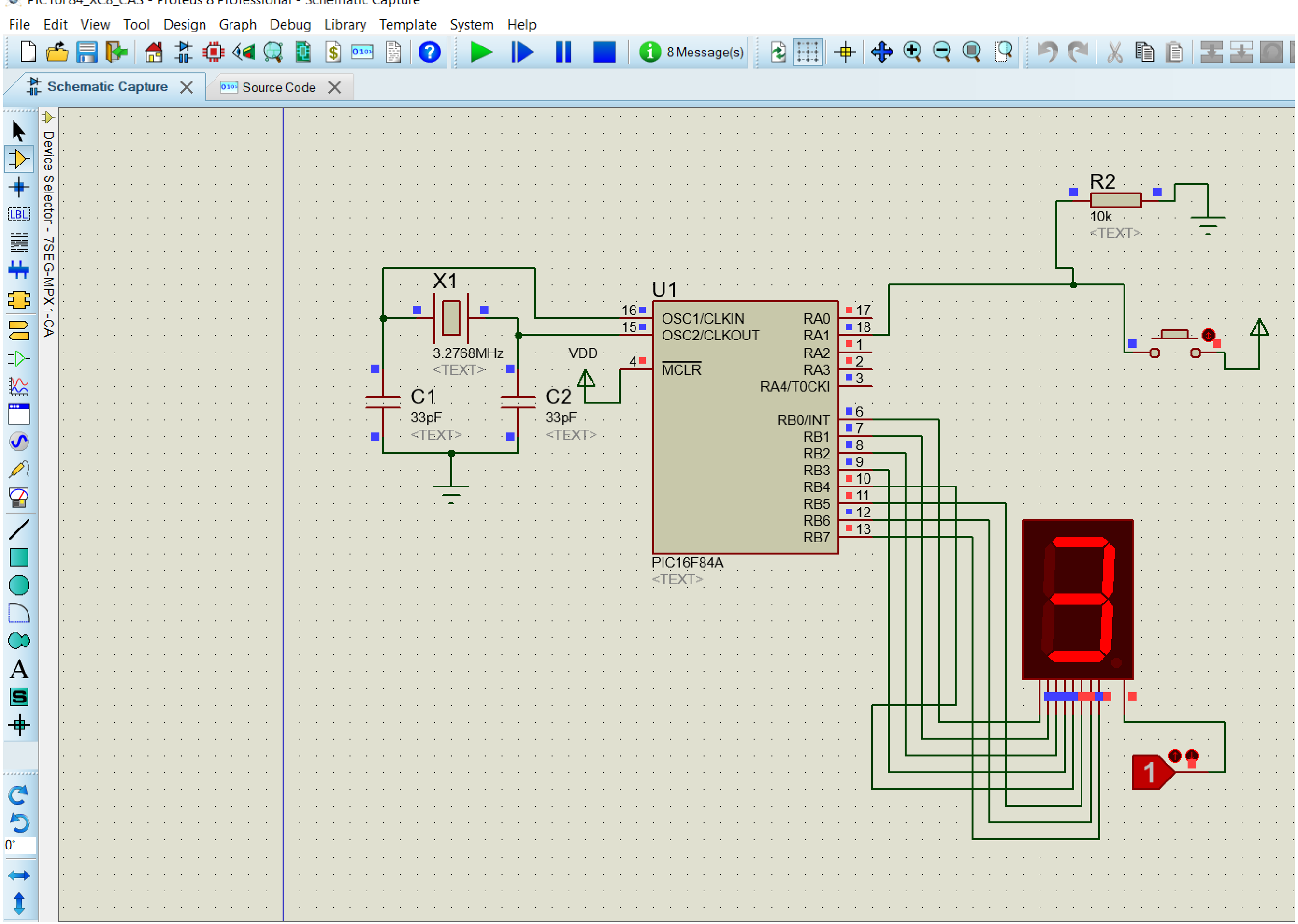
```
{SRCFILE=C:\Users\PENENA~1\AppData\Local\Temp\dc455bcd39a448ee8f0791
```

☐ Exclude from Simulation
☐ Exclude from PCB Layout
☐ Exclude from Bill of Materials
☐ Attach hierarchy module
☐ Hide common pins
☐ Edit all properties as text

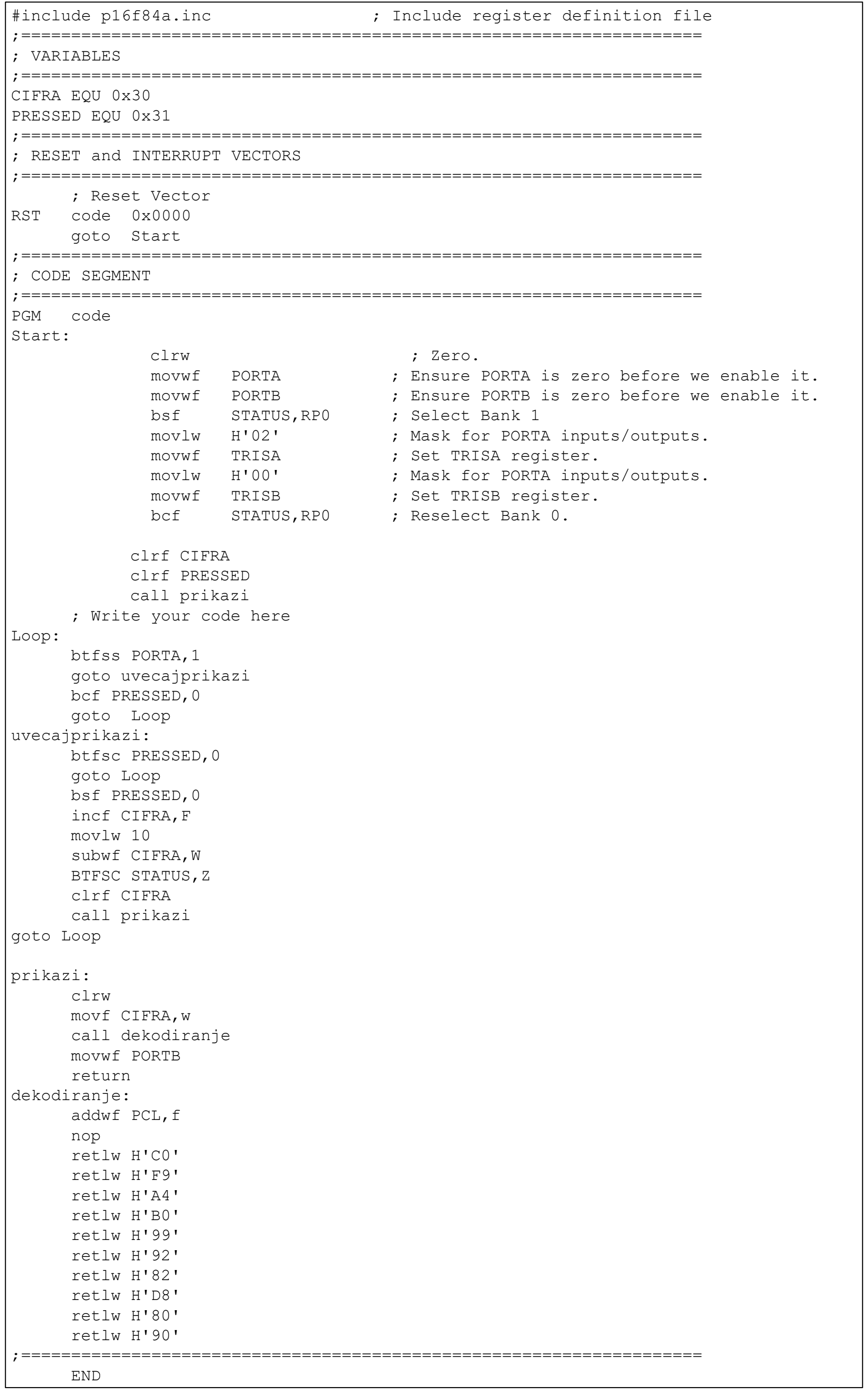
Zadatak 1

- Napisati program na asemblerskom jeziku/XC8 za PIC16F84A koji inkrementira sadržaj 7s displeja pritiskom na taster.
- Pin RA1 povezan je na taster a linije RB0-RB6 na segmente displeja.
- Pritiskom na taster treba inkrementirati sadrzaj prikazan na displeju.
- Kada vrednost prikazna na displeju bude 9 nakon inkrementiranja postaje 0.
- Napisati proceduru za prikaz cifre na displeju a tablicu definicija cifara realizovati programski.
- Početna vrednost prikazana na displeju je nula.

Šema


PIC16F84_XC8_CA3 - Proteus 8 Professional - Schematic Capture
File
Edit
View
Tool
Design
Graph
Debug
Library
Template
System
Help
8 Message(s)
Schematic Capture
Source Code
Device Selector - 7SEG-MPX1-CA
X1
3.2768MHz
<TEXT>
C1
33pF
<TEXT>
C2
33pF
<TEXT>
VDD
U1
16
15
4
OSC1/CLKIN
OSC2/CLKOUT
MCLR
RA0
RA1
RA2
RA3
RA4/T0CKI
RB0/INT
RB1
RB2
RB3
RB4
RB5
RB6
RB7
17
18
1
2
3
6
7
8
9
10
11
12
13
PIC16F84A
<TEXT>
R2
10k
<TEXT>
1
0°

ASM kod



```
#include p16f84a.inc                ; Include register definition file
;====================================================================
; VARIABLES
;====================================================================
CIFRA EQU 0x30
PRESSED EQU 0x31
;====================================================================
; RESET and INTERRUPT VECTORS
;====================================================================
      ; Reset Vector
RST   code  0x0000
      goto  Start
;====================================================================
; CODE SEGMENT
;====================================================================
PGM   code
Start:
              clrw                      ; Zero.
              movwf   PORTA           ; Ensure PORTA is zero before we enable it.
              movwf   PORTB           ; Ensure PORTB is zero before we enable it.
              bsf     STATUS,RP0      ; Select Bank 1
              movlw   H'02'           ; Mask for PORTA inputs/outputs.
              movwf   TRISA           ; Set TRISA register.
              movlw   H'00'           ; Mask for PORTA inputs/outputs.
              movwf   TRISB           ; Set TRISB register.
              bcf     STATUS,RP0      ; Reselect Bank 0.

            clrf CIFRA
            clrf PRESSED
            call prikazi
      ; Write your code here
Loop:
      btfss PORTA,1
      goto uvecajprikazi
      bcf PRESSED,0
      goto  Loop
uvecajprikazi:
      btfsc PRESSED,0
      goto Loop
      bsf PRESSED,0
      incf CIFRA,F
      movlw 10
      subwf CIFRA,W
      BTFSC STATUS,Z
      clrf CIFRA
      call prikazi
goto Loop

prikazi:
      clrw
      movf CIFRA,w
      call dekodiranje
      movwf PORTB
      return
dekodiranje:
      addwf PCL,f
      nop
      retlw H'C0'
      retlw H'F9'
      retlw H'A4'
      retlw H'B0'
      retlw H'99'
      retlw H'92'
      retlw H'82'
      retlw H'D8'
      retlw H'80'
      retlw H'90'
;====================================================================
      END
```

XC8 kod

```
#include<htc.h>

// Config word
__CONFIG(FOSC_HS & WDTE_OFF & PWRTE_ON & CP_OFF);


//CPU takt
//Mora da se definise ako se koristi  __delay_ms()
#define _XTAL_FREQ   20000000

void prikazi(int cifra);



// Main function
void main()
{
   int cifra=0;
   int pressed=0;

   TRISB=0;
   TRISA=0;
   TRISAbits.TRISA1=1;
   prikazi(cifra);
      while(1)
      {
         if(PORTAbits.RA1==1)
         {
            if(pressed==0){
             pressed=1;
             cifra+=1;
             if(cifra>9)
                cifra=0;
             prikazi(cifra);
            }
         } else
         {
            pressed=0;
         }

      }

}
void prikazi(int cifra)
{
   static const int codes[]={0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92, 0x82, 0xd8, 0x80, 0x90};
   if(cifra<10)
      PORTB=codes[cifra];
}
```