2. DOMAČA NALOGA - ODDAJNIK MORSEJEVE ABECEDE

Program je sestavljen iz glavnih funkcij za strukturirano izvajanje željenih operacij morsejevega oddajalnika:

- LED_On

Elementarna funkcija programa, ki spremeni vrednost PIO_CODR na 1, kar povzroči, da se prižge LED na plošči FRI-SMS.

- LED_Off

Elementarna funkcija programa, ki spremeni vrednost PIO_SODR na 1, kar povzroči, da se ugasne LED na plošči FRI-SMS.

- Delay

Funkcija prejme parameter 'Counter', ki določi kolikokrat se izvede notranja zanka. Ta zanka pa uporablja enoto Timer na plošči, katero uporabimo za zakasnitev 1ms.

- XMCHAR

Funkcija prejme znak '.' ali '-'. Na podlagi vhoda preprosto kliče LED_On, Delay in potem LED_Off. Delay je 150 ms za '.' in 300ms za '-'.

- XMCODE

Na vhod dobimo tabelo znakov – string. Za vsako črko kličemo metodo XMCHAR.

- GETMCODE

Za vsako črko v abecedi je definirana kombinacija '.' in '-'. Switch stavek poišče pravo zaporedje morsejeve abecede za poljubno črko angleške abecede. Prvotno sem za ta namen želel uporabiti tabelo, ki vsebuje vse kombinacije morsejeve abecede (index char c – 65), vendar je ob poskusu uporabe tabel FRI-SMS plošča prenehala delovati (brez napak, vendar program ni deloval), zato sem se odločil za pristop s switch.

- XWORD

Za vsako črko vhodnega niza metodi izvedemo funkcijo GETMCODE. Prav tako preverimo, če je vhodni znak zapisan z malo začetnico, če je, potem kličemo metodo GETMCODE s parametrom char c – 32 (32 je konstantna vrednost, s katero po ASCII tabeli preidemo iz male v velike črke, ali obratno).

- find_length

Funkcija služi za pridobitev dolžine vhodnega niza. Prvotno sem uporabil ukaz strlen(niz), vendar ta potrebuje knjižnico string.h, katere FRI-SMS nima nameščene.

- Main

Glavna funkcija programa. V večni zanki preprosto pošiljamo niz »sos« in »SOS«, s katerima preverimo delovanje programa z malimi in velikimi črkami.

KODA PROGRAMA (Dodane funkcije):

```
int find_length(char *input)
{
   int length = 0;
   while(input[length] != '\0')
   {
      length++;
   }
         return length;
}
```

```
void XMCHAR(char c)
{
        if(c == '.')
        {
                LED_On(PORT);
    Delay(150);
        }
        if(c == '-')
        {
                LED_On(PORT);
    Delay(300);
        }
        LED_Off(PORT);
        Delay(150);

}

void XMCODE(char c[])
{
        int i;
        for(i = 0; i < find_length(c); i++)
        {
                XMCHAR(c[i]);
        }
        Delay(150);
}
```

```
char * GETMCODE(char c)
{
        switch(c)
 {
 case 'A':
  return ".-";
  break;

 case 'B':
  return "-...";
  break;

 case 'C':
  return "-.-.";
  break;

 case 'D':
  return "-..";
  break;

 case 'E':
  return ".";
  break;

 case 'F':
  return "..-.";
  break;

 case 'G':
  return "--.";
  break;
```

```
case 'H':
 return "....";
 break;

case 'I':
 return "..";
 break;

case 'J':
 return ".---";
 break;

case 'K':
 return "-.-";
 break;

case 'L':
 return ".-..";
 break;

case 'M':
 return "--";
 break;

case 'N':
 return "-.";
 break;

case 'O':
 return "---";
```

```
  break;

case 'P':
  return ".--.";
  break;

case 'Q':
  return "--.-";
  break;

case 'R':
  return ".-.";
  break;

case 'S':
  return "...";
  break;

case 'T':
  return "-";
  break;

case 'U':
  return "..-";
  break;

case 'V':
  return "...-";
  break;

case 'W':
```

```
    return ".--";
    break;

  case 'X':
    return "-..-";
    break;

  case 'Y':
    return "-.--";
    break;

  case 'Z':
    return "--..";
    break;

  default:
    return ".-";
    break;

  }
}
```

```
void XWORD(char c[])
{
        int i;
        for(i = 0; i < find_length(c); i++)
        {
                char *mc = GETMCODE(c[i]);
   if(mc >= 'a' && mc <= 'z')
    XMCODE(mc - 32);
   else
      XMCODE(mc);
        }
        Delay(1000);
}


int main(void)
{
 initPIO_Port(PORT);
 Init_Timer_Tc0();

 LED_Off(PORT);

 /* Loop for ever */
 while (1) {
    // Deluje za male in velike zacetnice
    XWORD("sos");
    XWORD("SOS");
 };
}
```