

Универзитет у Београду
Електротехнички факултет
Катедра за електронику

Интегрисани рачунарски системи

Лабораторијска вежба

Рад са портовима

Београд, 2022.

1 Циљ вежбе

Циљ ове вежбе је да се студенти креирањем неколико асемблерских програма упознају са развојним окружењем, асемблерским преводиоцем и коришћењем дигиталних портова путем полирања и прекида.

На располагању је софтвер *Code Composer Studio 10* (може млађи, а може и старији), а дизајн се имплементира на TI LaunchPad MSP-EXP430F5529LP плочи са MSP430F5529 микроконтролером.

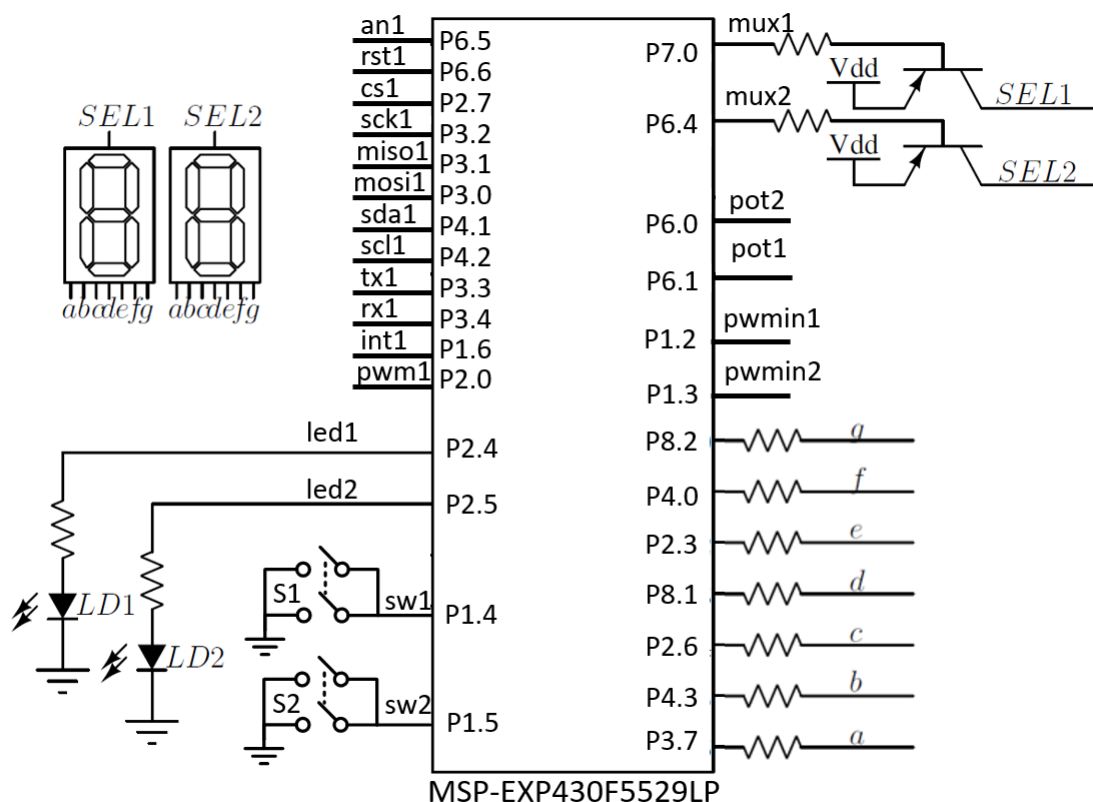
2 Припрема за рад

Покренути *Code Composer Studio 10* и у жељеном директоријуму креирати *workspace* са називом *lab*.

У *workspace*-у *lab* креирати пројекат под називом *lab1_gggg_bbbb*, у којем ћете реализовати тражену функционалност, а где су *gggg* и *bbbb* година и број вашег индекса.

Студентима се саветује да прво прочитају целокупан текст вежбе са разумевањем, пре него што започну имплементацију решења.

3 Шема развојне плоче са релевантним периферијама



Слика 1: Шема развојне плоче са потребним периферијама

4 Задатак

- 4.1 (5 поена) У оквиру пројекта креирати асемблерски програм који обезбеђује исписивање задате хексадецималне цифре на седмосегментни LED дисплеј. Да би се то реализовало потребно је у аутоматски креиран шаблон за асемблерски пројекат убацити део кода за иницијализацију портова као излазних, део кода за индексирање табеле и креирати табелу приказа за сегменте за сваку од 16 цифара хексадецималног бројног система. Седмосегментни дисплеј је са заједничком анодом. Листинг одговарајућих делова кода које је потребно убацити у пројекат је:

декларација табела сегмената за цифре од 0..9

НАПОМЕНА: *потребно је допунити табеле са стањима сегмената за хексадецималне цифре A..F*

НАПОМЕНА: *На рачунским вежбама, на снимку Вежбе 1_4 се помиње да постоји инверторски степен пре сегмената дисплеја. Ово не важи за актуелни хардвер, па се стога вредности из табеле константи морају инвертовати како би се жељени дисплеј укључио. Ово је већ учињено у коду који следи*

```
.sect .const
segtab_p2 .byte 0x48 ; tabela segmenata c i e
           .byte 0x40 ; na pinovima 2.6 i 2.3
           .byte 0x08
           .byte 0x40
           .byte 0x40
           .byte 0x48
           .byte 0x40
           .byte 0x48
           .byte 0x40

segtab_p3 .byte 0x80 ; tabela segmenta a
           .byte 0x00 ; na pinu 3.7
           .byte 0x80
           .byte 0x80
           .byte 0x00
           .byte 0x80
           .byte 0x80
           .byte 0x80
           .byte 0x80
           .byte 0x80

segtab_p4 .byte 0x09 ; tabela segmenata b i f
           .byte 0x08 ; na pinovima 4.3 i 4.0
           .byte 0x08
           .byte 0x08
           .byte 0x09
           .byte 0x01
           .byte 0x01
           .byte 0x08
           .byte 0x09
           .byte 0x09

segtab_p8 .byte 0x02 ; tabela segmenata g i d
           .byte 0x00 ; na pinovima 8.2 i 8.1
           .byte 0x06
           .byte 0x06
           .byte 0x04
           .byte 0x06
           .byte 0x06
```

```
.byte 0x00
.byte 0x06
.byte 0x06
```

кôд програма

```

bis.b #0x48,&P2OUT ; inicijalizacija portova 2,3,4,8
bis.b #0x48,&P2DIR ; pinovi su izlazni sa pocetnom vrednosti '1'
bis.b #BIT7,&P3OUT ; svi segmenti su ugaseni nakon inicijalizacije
bis.b #BIT7,&P3DIR
bis.b #0x09,&P4OUT
bis.b #0x09,&P4DIR
bis.b #0x06,&P8OUT
bis.b #0x06,&P8DIR
bic.b #BIT0,&P7OUT ; pocetna vrednost '0', aktivacija signala SEL1
bis.b #BIT0,&P7DIR ; pin 7.0 je izlazni sa gore definisanom vrednosti

lab mov.w #0x03,R10 ; vrednost koja se ispisuje na displej
bis.b #0x48, &P2OUT ; deaktiviranje segmenata c i e na portu 2
bic.b segtab_p2(R10), &P2OUT ; indeksiranje tabele segmenata i ispis na displej
bis.b #BIT7, &P3OUT ; deaktiviranje segmenta a na portu 3
bic.b segtab_p3(R10), &P3OUT ; indeksiranje tabele segmenata i ispis na displej
bis.b #0x09, &P4OUT ; deaktiviranje segmenata b i f na portu 4
bic.b segtab_p4(R10), &P4OUT ; indeksiranje tabele segmenata i ispis na displej
bis.b #0x06, &P8OUT ; deaktiviranje segmenata d i g na portu 8
bic.b segtab_p8(R10), &P8OUT ; indeksiranje tabele segmenata i ispis na displej
jmp lab ; skok na labelu - za potrebe debugovanja/simulacije
nop
```

По завршетку, верификовати да програм успешно ради. Уколико студент не успе да уради тачку 4.5, треба документовати функционалност ове тачке путем кратког видео-записа. Видео-запису доделити смислено име, рецимо **zadatak_4_1**.

- 4.2** (10 поена) Написати асемблерски кôд који обезбеђује да је док је тастер S1 (на прилагодној плочици означен са S3) притиснут, LE диода LD1 (на прилагодној плочици означена са L3) укључена и обрнуто. Кôд који обавља тражену функцију је:

```

bic.b #BIT4,&P1DIR ; pin 1.4 je ulazni, taster S1
bis.b #BIT4,&P1REN ; dozvola pull up/down otpornika
bis.b #BIT4,&P1OUT ; aktivacija pull up otpornika
bic.b #BIT4,&P2OUT ; pocetna vrednost LD1 je '0'
bis.b #BIT4,&P2DIR ; pin 2.4 je izlazni , LD1

loop bit.b #BIT4,&P1IN ; testira bit 4 porta 1 (taster S1)
jz ledon
bic.b #BIT4,&P2OUT ; iskljucuje diodu LD1
jmp loop
ledon bis.b #BIT4,&P2OUT ; ukljucuje diodu LD1
jmp loop
```

По завршетку, верификовати да програм успешно ради. Уколико студент не успе да уради тачку 4.3, треба документовати функционалност ове тачке путем кратког видео-записа. Видео-запису доделити смислено име, рецимо **zadatak_4_2**.

- 4.3** (25 поена) Модификовати програм из тачке **4.2** тако да се на исти начин као што се тастером S1 контролише LE диода LD1 , и тастером S2 (на прилагодној плочици означен са S4) контролише LE диода LD2 (на прилагодној плочици означена са L4).

По завршетку, верификовати да програм успешно ради и документовати функционалност путем кратког видео-записа. Видео-запису доделити смислено име, рецимо `zadatak_4_3`.

- 4.4** (10 поена) Написати асемблерски код који обезбеђује детекцију притиска тастера `S1` и на сваки детектовани притисак мења стање ЛЕ диоде `LD1`. Функционалност реализовати коришћењем прекида порта `P1`. Да би се то остварило потребно је омогућити прекид на пину 4 порта `P1`, написати прекидну рутину која мења стање укључености ЛЕ диоде, и напунити табелу вектора прекида са адресом прекидне рутине. Главни програм у овом случају не врши никакву функцију већ се врти у бесконачној петљи. У наставку су дати делови програма који реализују тражену функционалност:

иницијализација табеле прекида

```
.sect    .int47
.short  PORT1_ISR
```

прекидна рутина

```
PORT1_ISR    bit.b    #BIT4, &P1IFG      ; provera porekla zahteva za prekid
              jz       exit
              mov      #0xffff,R9        ; cekanje da se nakon pritiska tastera
wait          dec      R9                 ; stanje smiri
              jnz      wait
              bit.b    #BIT4,&P1IN        ; testiranje da li je taster jos uvek pritisnut
              jz       Ton
Toff          jmp      exit
Ton           xor.b    #BIT4,&P2OUT        ; menja stanje diode LD1
exit          bic.b    #BIT4,&P1IFG        ; brisanje flega koji oznacava na kom se pinu dogodio prekid
              reti
```

главни програм

```
bic.b    #BIT4,&P1DIR        ; pin 1.4 je ulazni, taster S1
bis.b    #BIT4,&P1REN        ; dozvola pull up/down otpornika
bis.b    #BIT4,&P1OUT        ; aktivacija pull up otpornika
bic.b    #BIT4,&P1IFG        ; inicijalno brisanje flega prekida
bis.b    #BIT4,&P1IE         ; dozvola prekida na pinu 4 porta 1
bis.b    #BIT4,&P1IES        ; opadajuca ivica generise prekid
bic.b    #BIT4,&P2OUT        ; pocetna vrednost LD1 je '0'
bis.b    #BIT4,&P2DIR        ; pin 2.4 je izlazni , LD1
nop
bis.b    #GIE,SR             ; dozvola svih prekida
nop
jmp      $                   ; jump to current location '$'
```

- 4.5** (30 поена) Модификовати програм из тачке **4.4**, тако да се поред промене стања укључености ЛЕ диоде `LD1` при притиску тастера `S1` инкрементира вредност бројача у регистру `R10`. У главном програму додати део кода који на основу нижих 4 бита у регистру `R10` (маскирањем са `0x0f`) генерише испис на седмосегментни LED дисплеј.
- 4.6** (20 поена) Модификовати тачку **4.5** додавањем прекидне рутине за тастер `S2`, тако да се притиском на тај тастер мења стање диоде `LD2` и врши декрементирање броја у регистру `R10`.

По завршетку, верификовати да програм успешно ради и документовати функционалност путем кратког видео-записа. Видео-запису доделити смислено име, рецимо `zadatak_4_6`.

5 Правила рада

Лабораторијска вежба се ради самостално у кућним условима на подељеним развојним системима. Преписивање или коришћење туђег кода биће санкционисано.

Поред кода који обезбеђује тражену функционалност неопходно је водити рачуна о изгледу (идентификацији) кода, уредности кода, као и јасном коментарисању сваке инструкције. Изглед кода и коментари ће се узимати у разматрање приликом бодовања лабораторијске вежбе.

Пројекат `lab1_gggg_bbbb` са свим фајловима, видео-снимак на коме је приказана реализација тачке **4.3**, као и видео-снимак на коме је приказана крајња функционалност система (пролазак кроз све цифре на дисплеју, приказано инкрементирање, декрементирање, утицај тастера на стања диода из тачака **4.4**, **4.5** и **4.6**) запаковати у `lab1_gggg_bbbb.zip` и послати на адресу `cvele@etf.rs` са насловом мејла **13E044IRS_L1**. Дозвољено је и слање линка ка решењу, уколико је архива постављена на неки дељени диск или сервис.

Рок за предају решења лабораторијске вежбе је **недеља, 10.04. у 20:00 часова**.