



Uvod

Za pripremu rješenja zadataka laboratorijskih vježbi na predmetu Ugradbeni sistemi i za njihovu realizaciju u laboratoriji ćemo koristiti različite razvojne alate i razvojne sisteme. Cilj prve laboratorijske vježbe je upoznavanje studenata sa laboratorijom i jednim od razvojnih sistema koje ćemo koristiti.

U ovoj laboratorijskoj vježbi ćemo koristiti razvojni sistem (hardver) *picoETF*, baziran na mikrokontroleru RP2040, te razvojno okruženje Thonny IDE za razvoj koda u MicroPython-u.

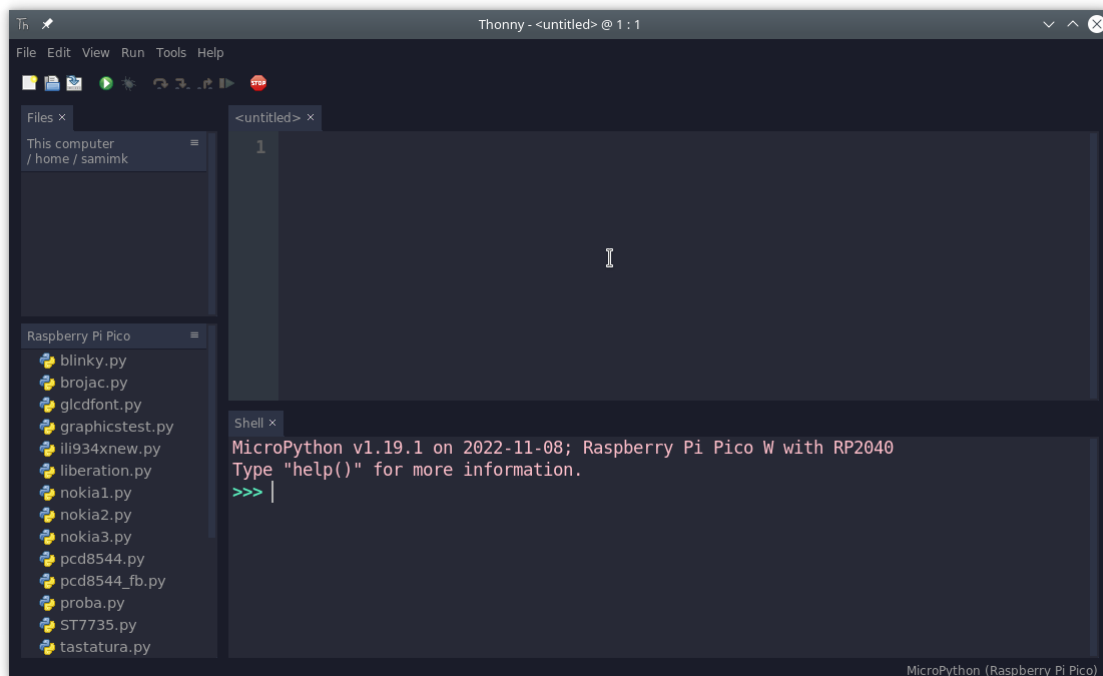
U nastavku će biti data pojašnjenja i svi koraci neophodni za rad sa navedenim sistemom, te kreiranje i izvršavanje aplikacija.

Razvojni sistem picoETF

Razvojni sistem *picoETF* je baziran na mikrokontroleru RP2040, razvijenom od strane Raspberry Pi fondacije. Ovaj mikrokontroler je osnova sve popularnijeg *Raspberry Pi Pico*, koji je korišten i u okviru razvojnog sistema *picoETF*.

Sistem *picoETF* (slika 1) posjeduje:

- četiri tastera,
- osam LED,
- jednu RGB LED,
- headere za:
 - matričnu tastaturu,
 - LCD displej N5110,
 - TFT displej,
 - četverocifreni 7-segmentni displej,
 - linearni senzor temperature LM35DZ,
 - ostale pinove i napajanje mikrokontrolera.



Slika 2: Thonny IDE.

Da bi se koristila klasa `Pin`, potrebno je prvo uvesti modul `machine`, u okviru koga se ova klasa nalazi.

To se postiže izvršavanjem kôda na REPL promptu:

```
>>> import machine
```

U ovom slučaju se klasa `Pin` koristi navođenjem imena `machine.Pin`. Ukoliko se izvrši kôd:

```
>>> from machine import Pin
```

tada je dovoljno samo navođenje imena `Pin`.

ZADACI

Prvi zadatak će biti da se na sistem *picoETF* instalira MicroPython firmware ¹. Instalaciju firmware radite samo na prvoj laboratorijskoj vježbi kako bi se upoznali sa procedurom instalacije. Na narednim vježbama nije potrebna ponovna instalacija firmware.

Ostali zadaci za laboratorijsku vježbu su koncipirani tako da se korištenjem REPL prompta kroz interaktivni rad upoznamo sa radom sa tasterima i LED diodama. Nakon toga će biti potrebno napisati aplikaciju i izvršiti je.

Zadatak 1

Prije instalacije firmware preporuka je da se obriše kompletan sadržaj koji je sačuvan u memoriji mikrokontrolera. Ovo je potrebno uraditi posebno u slučajevima kada mikrokontroler ne reagira na uobičajene metode programiranja.

Procedura brisanja sadržaja i instalacije firmware na razvojnom sistemu *picoETF* je sljedeća:

¹Značenje pojma firmware je pojašnjeno npr. na [linku](#)

- prilikom priključivanja USB kabla na PC računar držati pritisnuto dugme **BOOT SEL** na ploči Raspberry Pi Pico. Nakon inicijalizacije sistema, na PC računaru će se pojaviti novi (virtualni) USB disk,
- preuzeti fajl `flash_nuke.uf2` kojeg je potrebno prekopirati na ovaj USB disk. Navedeni fajl za brisanje sadržaja memorije mikrokontrolera se može preuzeti sa [linka](#),
- nakon obavljenog brisanja u razvojnom okruženju Thonny moguće je provjeriti da li je memorija mikrokontrolera prazna. U donjem desnom uglu biti će dostupna samo **Local Python 3** verzija,
- sada je potrebno preuzeti najnoviji MicroPython firmware za RP2040 sa [linka](#),
- preuzeti firmware se iskopira na virtualni USB disk, nakon čega će biti upisan u programsku memoriju i sistem će se resetovati. Time je postupak instaliranja MicroPython firmware-a okončan i sistemu je moguće pristupiti putem REPL prompta.

Napomena

Kada se USB kabl poveže na PC računar, potrebno je sačekati nekoliko sekundi da se virtualni serijski port inicijalizira. Ukoliko u donjem desnom dijelu prozora ne bude ispisana verzija MicroPython firmware-a i REPL prompt (`>>>`), potrebno je kliknuti na dugme **Stop/Restart backend** u toolbar-u, ili odabrati tu opciju u okviru menija **Run**.

Zadatak 2

Kako je već rečeno, razvojni sistem *picoETF* posjeduje četiri tastera, koji se mogu očitavati korištenjem digitalnih ulaza. Svaki od tastera je povezan na jedan GPIO (General Purpose Input Output) pin mikrokontrolera RP2040. Tako je na GPIO pin GP0 povezan taster T1.

Na REPL promptu izvršiti sljedeću naredbu:

```
>>> t1 = Pin(0,Pin.IN)
```

Nakon izvršavanja ove naredbe, biće kreiran objekat `t1`, koji možemo koristiti za očitavanje stanja tastera. Ukoliko taster T1 nije pritisnut, biće očitana vrijednost `log. 0`, a ukoliko je taster pritisnut, biće očitana vrijednost `log. 1`.

Hint

Ukoliko se radi o digitalnom ulazu, nije neophodno navoditi `Pin.IN`, jer je svaki GPIO inicijalno konfiguriran kao digitalni ulaz. Tako se isti efekat postiže i izvršavanjem kôda:

```
>>> t1 = Pin(0)
```

Očitavanje stanja tastera postićemo izvršavanjem:

```
>>> t1.value()
```

i ukoliko se ova naredba izvrši, biće ispisana vrijednost stanja tastera T1. Ova vrijednost se može pridružiti varijabli:

```
>>> stanje = t1.value()
```

Ova varijabla se dalje može koristiti u programskom kodu, npr.:

```
>>> print(stanje)
```

Zadatak 3

Razvojni sistem *picoETF* posjeduje i 8 LED dioda koje se mogu paliti i gasiti korištenjem digitalnih izlaza. Svaka od LED je povezana na jedan GPIO pin mikrokontrolera RP2040. Tako je LED0 povezana na GPIO pin GP4.

Na REPL promptu izvršiti sljedeću naredbu:

```
>>> led0 = Pin(4,Pin.OUT)
```

Nakon izvršavanja ove naredbe, biće kreiran objekat `led0`, koji možemo koristiti za postavljanje vrijednosti digitalnog izlaza GP4, na kome se nalazi LED0.

Inicijalno je na digitalnom izlazu vrijednost log. 0, tako da LED0 neće svijetliti (LED su povezane u konfiguraciji sa zajedničkom katodom).

LED0 možemo upaliti (na digitalni izlaz GP4 ispisat log. 1) izvršavanjem naredbe:

```
>>> led0.value(1)
```

Hint

Umjesto pomenute naredbe, isti efekat se može postići sa:

```
>>> led0.on()  
>>> led0.high()  
>>> led0(1)
```

Napomena

Logička vrijednost log. 0 se može predstaviti ili kao 0 ili `False`, a logička vrijednost log. 1 kao 1 ili `True`. Tako npr. diodu LED0 će upaliti i izvršavanje naredbe:

```
>>> led0.value(True)
```

Vrijednost digitalnog izlaza je moguće postavljati i na osnovu vrijednosti neke varijable ili na osnovu očitavanja digitalnog ulaza. Tako će se npr. izvršavanjem koda:

```
>>> led0.value(t1.value())
```

postići da će LED0 svijetliti ukoliko je pri izvršavanju naredbe taster T1 bio pritisnut, a biti ugašena ukoliko je prilikom izvršavanja naredbe taster T1 bio otpušten.

Zadatak 4

Umjesto izvršavanja naredbi liniju po liniju, moguće je više linija koda zapisati u fajl (npr. u fajl sa ekstenzijom `.py` (npr. `ime_fajla.py`) i onda sve linije kôda izvršiti pozivom:

```
>>> import ime_fajla
```

Potrebno je napisati programski kôd koji omogućava da se korištenjem `while` petlje neprekidno očitava stanje tastera T4 i na osnovu očitano stanja pali ili gasi dioda LED7.

Napomena

Osnovni elementi sintakse programskog jezika Python se mogu naći npr. na sljedećem [linku](#).

Hint

Izvršavanje programskog kôda se može prekinuti pritiskom na `Ctrl+C`, nakon čega se ponovo aktivira REPL prompt.

Zadatak 5

Nakon što smo se upoznali sa osnovnim načinom korištenja digitalnih ulaza i izlaza, sada možemo napisati i složeniji programski kod.

U okviru ovog zadatka je potrebno napisati programski kod (i snimiti ga u fajl) koji realizira binarni brojač na diodama LED0-LED7. Tastere T1-T4 je potrebno koristiti za upravljanje brojačem, na sljedeći način:

- pritisak na taster T1 povećava stanje brojača,
- pritisak na taster T2 umanjuje stanje brojača,
- pritisak na taster T3 postavlja stanje brojača na 0x00,
- pritisak na taster T4 postavlja stanje brojača na 0xFF.

Hint

Ukoliko na sistemu postoji fajl sa imenom `main.py`, ovaj fajl će se automatski izvršiti prilikom priključivanja sistema na napajanje.

Da bi se postiglo da se ova aplikacija automatski izvrši svaki put kada se *picoETF* priključi na napajanje, bez potrebe da je pozovemo korištenjem REPL prompta, potrebno je ili snimiti kôd u fajl `main.py` ili u okviru fajla `main.py` napisati naredbu:

```
import ime_fajla
```

Na ovaj način će se postići da će se automatski izvršiti kod u fajlu `main.py`, a taj kod će učitati i pokrenuti aplikaciju u fajlu `ime_fajla.py`.

Napomena

U zasebnom dokumentu će biti pojašnjeno kako se Wokwi simulator može koristiti da se u njemu kreira projekat koji omogućava simuliranje ponašanja *picoETF*. Nakon toga se ovaj projekat može koristiti za testiranje razvijenog kôda.

Zadaci za dodatne bodove (1b)

Zadatak 1 (0.25b)

Koristeći REPL prompt testirati rad RGB diode postavljajući različite kombinacije uključenosti R,G,B. U dokumentu sa rasporedom pinova picoETF pogledati koji pinovi se koriste za rad sa RGB. U izvještaju vježbe prikazati koje naredbe ste koristili u REPL promptu za postizanje različitih boja, tj. prikazati koje su zadate vrijednosti izlaza za svaku ostvarenu boju.

Zadatak 2 (0.75b)

Kreirati i učitati programski kôd prema prethodno datim procedurama, a koji treba da omogući promjenu boje RGB LED, pri čemu će se postavljati sve moguće kombinacije boja. Trajanje svake kombinacije emitovane svjetlosti treba da u startu bude 0.1s, a nakon svake iteracije (nakon izmjene svih kombinacija boja na RGB LED) da se uvećava za 0.1s do konačnih 1s. Nakon toga se vraća do početnih 0.1s, sa istim korakom od 0.1s. Navedeno se nakon toga ponavlja. U izvještaju za vježbu priložiti kôd, sa kratkim objašnjenjem ili priložiti kôd sa dovoljnim brojem komentara.

Hint

Titranje kontakata tastera može predstavljati problem prilikom očitavanja njegovog stanja. Utjecaj titranja kontakata se može umanjiti "usporavanjem" izvršavanja kôda, što se postiže korištenjem modula `time` implementiranog u MicroPython-u.

Literatura

- [1] S. Konjicija, E. Sokić (2019) *Ugradbeni sistemi: Hardverrski aspekti, Elektrotehnički fakultet Univerziteta u Sarajevu, ISBN 978-9958-629-77-8*
- [2] S. Konjicija (2022) *Predavanje Ugradbeni sistemi: Digitalni ulazi i izlazi*
- [3] MicroPython projekat *Dokumentacija za MicroPython za RP2040*