



# Primenjeni softverski inženjering



## ARDUINO, vežbe 1

Namenski Računarski sistemi

# Osnovni ciljevi vežbi

- Osetiti draž direktnog programiranja nad fizičkom mašinom
- Upoznati osnovne UI periferije i metode njihove kontrole
- Razumeti koncept Arduina kao ploče za razvoj
- Upoznati detalje i ograničenja Arduino programskog razvojnog okruženja
- Razumeti i principe implementacije istog na PIC32 (MIPS) platformi
- Razmotriti uklapanje asemblerskog i C koda
- Dakle, da vidimo kako to radi (ispod haube)
- Na vežbama će biti korišćeno:
  - Arduino platforma, razvojna ploča uC32
  - Basic IO Shield za uC32 razvojnu ploču
  - Arduino integrisano razvojno okruženje (IDE)

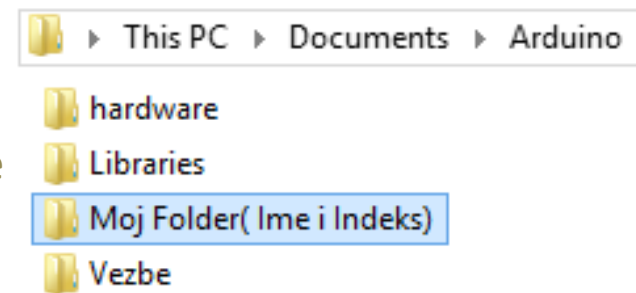
# Literatura

- Ideja je da se koristi originalna dokumentacija proizvođača (*datasheets*)
  - Realnost inženjerskog rada
- PIC32 mikrokontroler (*Microchip*):
  - 61132B\_PIC32ReferenceManual.pdf – opšti opis za celu familiju (CPU, Mem, IO)
  - 61143H.pdf – detalji (adrese) za konkretni CPU PIC32Mx3xx uc32 ploče
- *uc32* i *Basic IO Shield* dokumentacija (*Digilent*):
  - Referentni priručnik (*rm*) i shema (*sch*) svake od ploča
  - uC32 Pinout Table.xls – lista pinova na ploči
- Arduino, instalacija, dokumentacija i tutorijali:
  - <http://chipkit.net/wiki> chipKIT Platform Wiki (naša prva opcija)
  - [www.arduino.cc](http://www.arduino.cc) (Learning→Reference) opšte o Arduino okruženju

# Arduino instalacija

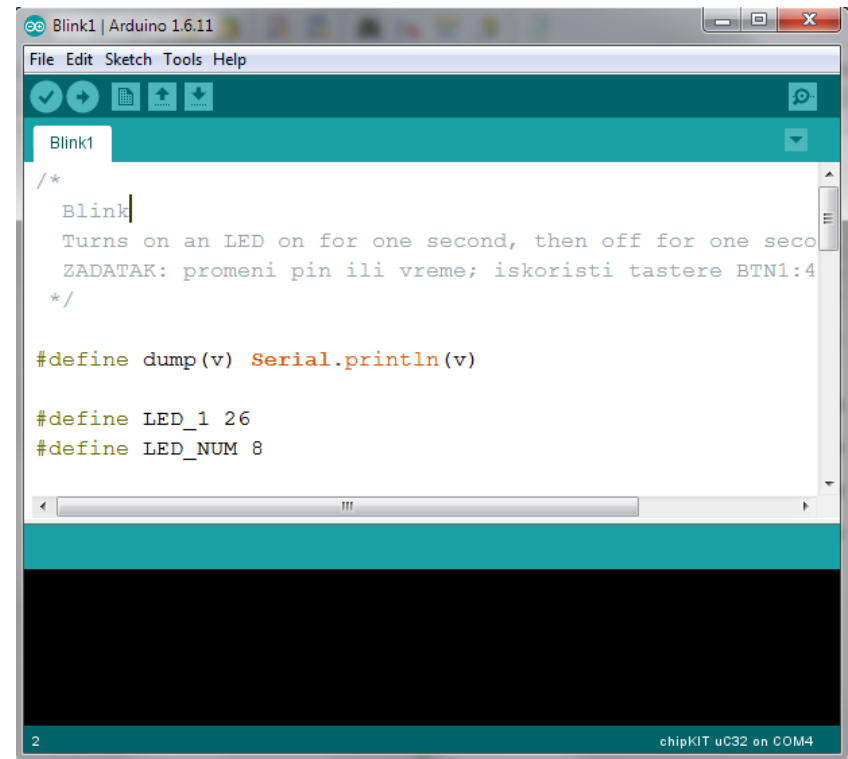
## (gde šta stoji)

- Hardware – uc32 programsko jezgro i osnovne biblioteke
  - izvorni kod (c, cpp, aSm), skripte za kompajliranje linkovanje
  - proučiti strukturu foldera i pogledati gde šta stoji
  - deo ovog foldera će biti proučen na vežbama
- Libraries – korisničke biblioteke, za specifične add-on ploče
  - IO Shield : EEPROM, OLED, Temp
- Vežbe – polazni kod za svaku od vežbi
  - Primeri: Blink, Uart1, i Asm
- Moj Folder – koji treba da napravi svako (ime/indeks) za sebe
  - Tu čuvati svoje programe



# Upoznavanje Arduino programskog okruženja

- Start meni – sve naredbe okruženja
- Traka sa alaktama
  - Kompajliranje koda
  - Upload generisanog koda
  - Novi projekat
  - Otvaranje postojećeg dokumenta
  - Čuvanje dokumenta
  - Serijski monitor
- Centralni deo za pisanje koda
  - Ovde se piše kod, obično u C-u
- Statusni deo
  - Sadrži podatke o kompajliranju
- Dodatna traka sa podacima
  - broj reda u editoru, verzija pločice i port



# Podešavanje Arduino programskog okruženja

- Prilikom prvog pokretanja programskog okruženja, potrebno je podesiti radni direktorijum za okruženje
  - U meniju **File/Preferences/** u opciji **Sketchbook location** podesiti radni direktorijum (na vežbama postaviti na **c:\Arduino** ili **c:\Arduino\Arduino**)`
  - Po potrebi restartovati okruženje da bi povuklo drajvere iz ovog foldera
- Nakon svakog sledećeg priključenja pločice chipKIT uC32 potrebno je:
  - U meniju **Tools/Board/** izabrati razvojnu pločicu (na vežbama je to **chipKIT uC32**). Ukoliko je već izabrana, preskočiti ovaj korak
  - Priključiti osnovnu pločicu na računar preko USB kabla
  - U meniju **Tools/Port/** izabati port na kojem se nalazi pločica (obično **COM4, COM5,...**)
- Nakon svega ovoga, pločica je spremna za rad 😊

# Struktura programa i API primitiva za rad sa digitalnim ulazima/izlazima

- Program se sastoji iz:

- Definicija i deklaracija globalnih vrednost
- Funkcije setup() koja se izvršava samo jednom na početku programa
- Funkcije loop() čije izvršavanje počinje nakon funkcije setup() i ponavlja se do beskonačnosti

- Naredbe:

- pinMode(pin, mode) – postavlja određeni pin na određeni mod (INPUT, OUTPUT, INPUT\_PULLUP)
- digitalWrite(pin, vrednost) – postavlja određeni pin na određenu vrednost (LOW, HIGH)
- digitalRead(pin) – čita vrednost pina. Vraća LOW ili HIGH
- delay(ms) – zaustavlja proces na određeni broj milisekundi

```
/* Komentari  
*/
```

```
// globalne definicije
```

```
void setup()  
{  
    // Kod za inicijalizaciju  
}
```

```
void loop()  
{  
    // kod koji se ponavlja  
}
```

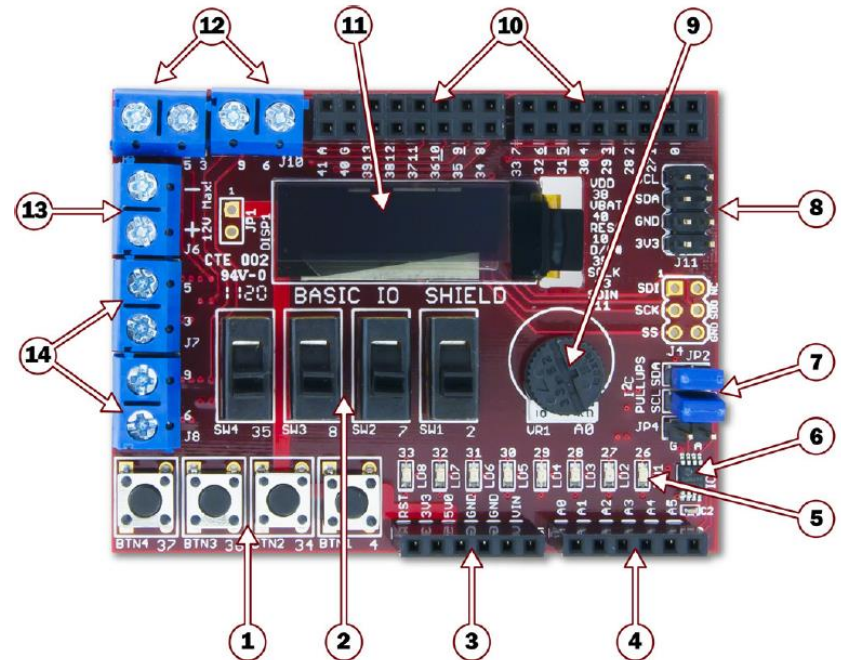
# Serijski monitor

- Koristi se za serijski ulaz/izlaz, najčešće za rad sa tastaturom i displejem
- Uključuje se klikom na poslednju ikonicu u traci sa alatkama
- Naredbe za korišćenje serijskog monitora:
  - `Serial.begin(brzina)` – inicijalizuje serijski mobitor na određenu brzinu komunikacije, najčešće 9600 bps
  - `Serial.println(vrednost)` - ispisuje vrednost na serijski monitor
  - `Serial.read()` – vraća učitani karakter, tj. njegov ascii kod
  - `Serial.available()` – vraća broj raspoloživih bajtova za čitanje, učitanih u bafer. Ako nema učitanih karaktera, vraća 0.



# Basic IO Shield

- Omogućuje neke osnovne ulazno/izlazne operacije
- Sadrži (nepotrebno je izostavljeno):
  1. Dugmadi BTN1,BTN2, BTN3 i BTN4 (pinovi 4, 34, 36, 37)
  2. Prekidači SW1, SW2, SW3 i SW4 (pinovi 2, 7, 8, 35)
  5. Led diode LD1, ..., LD8 (pinovi 26..33)
  6. IC2 senzor za temperaturu
  9. Potenciometar (A0)
  11. OLED displej



# Zadaci za vežbu

- Zadatak 1: Napisati program koji pali i gasi led diode LD8 i LD2 na pinovima 33 i 27. Pin 33 se i pali i gasi u jednom ciklusu loop-a, a pin 27 se pali ili gasi u zavisnosti od stanja prekidača SW2 na pinu 7.
- Zadatak 2: Napisati program koji pali lampice na pinovima od 26 do 33 u krug. Ukoliko je upaljen taster SW2 na pinu 7, rotirati lampice od manjih ka većim pinovima, a u suprotnom od većih ka manjim

# Zadaci za samostalnu vežbu

- Zadatak 3: Napisati program koji pali lampice na pinovima od 26 do 33, tako što prvo pali lampice 26 i 33, zatim 27 i 32, itd... sve dok se lampice ne sudare. Nakon toga, lampice paliti u obrnutom redosledu, pa sve ponavljati. U zavisnosti od stanja prekidača SW na pinu 7, podesiti jednu od dve brzine rotacije
- Zadatak 4: Napisati program koji broji koliko je upaljenih prekidača na pinovima 2, 7, 8 i 35, i u zavisnosti od toga pali duplo veći broj lampica, počevši od lampice sa brojem 26.