Prekidi kod Arduina

Internet stvari 2023. - II termin

Nenad Petrović

Univerzitet u Nišu, Elektronski fakultet

nenad.petrovic@elfak.ni.ac.rs, kancelarija 323

Uvod

- Prekid (engl. *interrupt*) koristimo u situacijama kada osluškujemo da li je došlo do određenog događaja u spoljašnjoj sredini, pri čemu želimo da mikrokontroler oslobodimo sa ciljem da obavlja neki drugi zadatak, bez propuštanja unosa, umesto da se blokira i vrti u beskonačnoj petlji dok vrši prozivku
- Bilo da je u pitanju pritisak tastera, zvučni senzor koji treba da registruje klik ili infracrveni senzor koji detektuje ubacivanje novčića u automat, upotrebom prekida se izbegava konstanta prozivka u programu (bez da se obavlja neki drugi, koristan zadatak), što rezultuje boljim iskorišćenjem (ograničene) procesorske moći, ali i pojednostavljenjem glavnog programa
- U Arduino programskom jeziku, od ključnog značaja za upotrebu prekida je funkcija **attachInterrupt()**
- Različiti digitalni pinovi kao izvori prekida, zavisno od Arduino modela

Pregled pinova za prekide po modelima

MODEL	DIGITALNI PINOVI ZA PREKIDE
Uno, Nano, Mini, other 328-based	**2, 3**
Uno WiFi Rev.2, Nano Every	**Svi digitalin pinovi**
Mega, Mega2560, MegaADK	**2, 3, 18, 19, 20, 21 (pinovi 20 i 21 nisu dostupni za upotrebu u okviru prekida dok se koriste za I2C komunikaciju)**
Micro, Leonardo, other 32u4-based	**0, 1, 2, 3, 7**
Zero	**Svi digitalni, osim 4**
MKR familija ploča	**0, 1, 4, 5, 6, 7, 8, 9, A1, A2**
Nano 33 IoT	**2, 3, 9, 10, 11, 13, A1, A5, A7**
Nano 33 BLE, Nano 33 BLE Sense	**Svi pinovi**
Due	**Svi digitalin pinovi**
101	**Svi digitalni pinovi (Samo pinovi 2, 5, 7, 8, 10, 11, 12, 13 rade u režimu CHANGE)**

Parametri attachInterrupt

- Varijante poziva
  - attachInterrupt(digitalPinToInterrupt(pin), ISR, mode) (preporučeno)
  - attachInterrupt(interrupt, ISR, mode) (ne propručuje se)
  - attachInterrupt(pin, ISR, mode) (samo SAMD ploče, Uno WiFi Rev2, Due, i 101)
- interrupt (int)
  - Broj interapta, ne slaže se nužno sa brojem pina
  - Preporučljivo da se korsiti **digitalPinToInterrupt(pin)** da bi se na osnovu pina pribavio odgovarajući broj interapta na koji se zapravo mapira
- pin
  - Broj pina na Arduino ploči
- ISR
  - Uslužna prekidna rutina koja se poziva kada dođe do interapta
  - Predstavlja funkciju bez parametara i ne vraća ništa
- Mode
  - Definiše kada dolazi do okidanja prekida
  - Postoje 5 mogućih vrednosti
    - **LOW**
      - Aktivira se prekid kada pin ima vrednost 0
    - **CHANGE**
      - Kada god se vrednost pina promeni
    - **RISING**
      - Kada god pin promeni vrednost od 0 na 1
    - **FALLING**
      - Kada ide od 1 na 0
    - **HIGH**
      - Samo za Due, Zero i MKR1000 ploče
      - Uvek kada pin ima vrednost 1

Rutine za opsluživanje prekida
Interrupt Service Routines (ISR)

- Poseban tip funkcija koje imaju pojedina ograničenja
- Ne smeju da imaju parametre, niti da vraćaju nešto
- Generalno, gleda se da budu što kraće i brže
- Ako sketch koristi više ISR
  - Samo jedna u istom trenutku
  - Drugi prekidi se izvršavaju po završetku prethodnog, zavisno od prioriteta
- Obično se preko globalnih promenljivih prenose podaci između ISR i glavnog programa

Izvori prekida i ISR rutine

- Šablon

```
ISR(SOURCE_vect)
{
...
}
```

INT0_vect	External Interrupt Request 0
INT1_vect	External Interrupt Request 1
INT2_vect	External Interrupt Request 2
INT3_vect	External Interrupt Request 3
INT6_vect	External Interrupt Request 6
PCINT0_vect	Pin Change Interrupt Request 0 (from PCINT0 through PCINT7)
TIMER1_CAPT_vect	Timer/Counter1 Capture Event
TIMER1_COMPA_vect	Timer/Counter1 Compare Match A
TIMER1_COMPB_vect	Timer/Counter1 Compare Match B
TIMER1_COMPC_vect	Timer/Counter1 Compare Match C
TIMER1_OVF_vect	Timer/Counter1 Overflow
TIMER0_COMPA_vect	Timer/Counter0 Compare Match A
TIMER0_COMPB_vect	Timer/Counter0 Compare Match B
TIMER0_OVF_vect	Timer/Counter0 Overflow
SPI_STC_vect	SPI Serial Transfer Complete
USART1_RX_vect	USART1, Rx Complete
USART1_UDRE_vect	USART1 Data register Empty
USART1_TX_vect	USART1, Tx Complete
ANALOG_COMP_vect	Analog Comparator
ADC_vect	ADC Conversion Complete
EE_READY_vect	EEPROM Ready
TIMER3_CAPT_vect	Timer/Counter3 Capture Event
TIMER3_COMPA_vect	Timer/Counter3 Compare Match A
TIMER3_COMPB_vect	Timer/Counter3 Compare Match B
TIMER3_COMPC_vect	Timer/Counter3 Compare Match C
TIMER3_OVF_vect	Timer/Counter3 Overflow
TWI_vect	2-wire Serial Interface
SPM_READY_vect	Store Program Memory Read
TIMER4_COMPA_vect	Timer/Counter4 Compare Match A
TIMER4_COMPB_vect	Timer/Counter4 Compare Match B
TIMER4_COMPD_vect	Timer/Counter4 Compare Match D
TIMER4_OVF_vect	Timer/Counter4 Overflow
TIMER4_FPF_vect	Timer/Counter4 Fault Protection Interrupt

Napomene o prekidnim rutinama

- sei()/cli()
  - Globalno omogućiti / onemogućiti prekide
- delay() ne radi
  - I sam koristi prekide
- micros() radi najviše 1-2ms
- milis() se ne povećava
- delayMicroseconds()se ponaša normalno
  - Ne koristi nikakve brojače i prekide
- Serijski podaci koji se primaju dok se opslužuje prekid mogu biti izgubljeni
- **Volatile** mora da stoji za sve one **globalne** promenljive koje menjamo **unutar tela ISR**

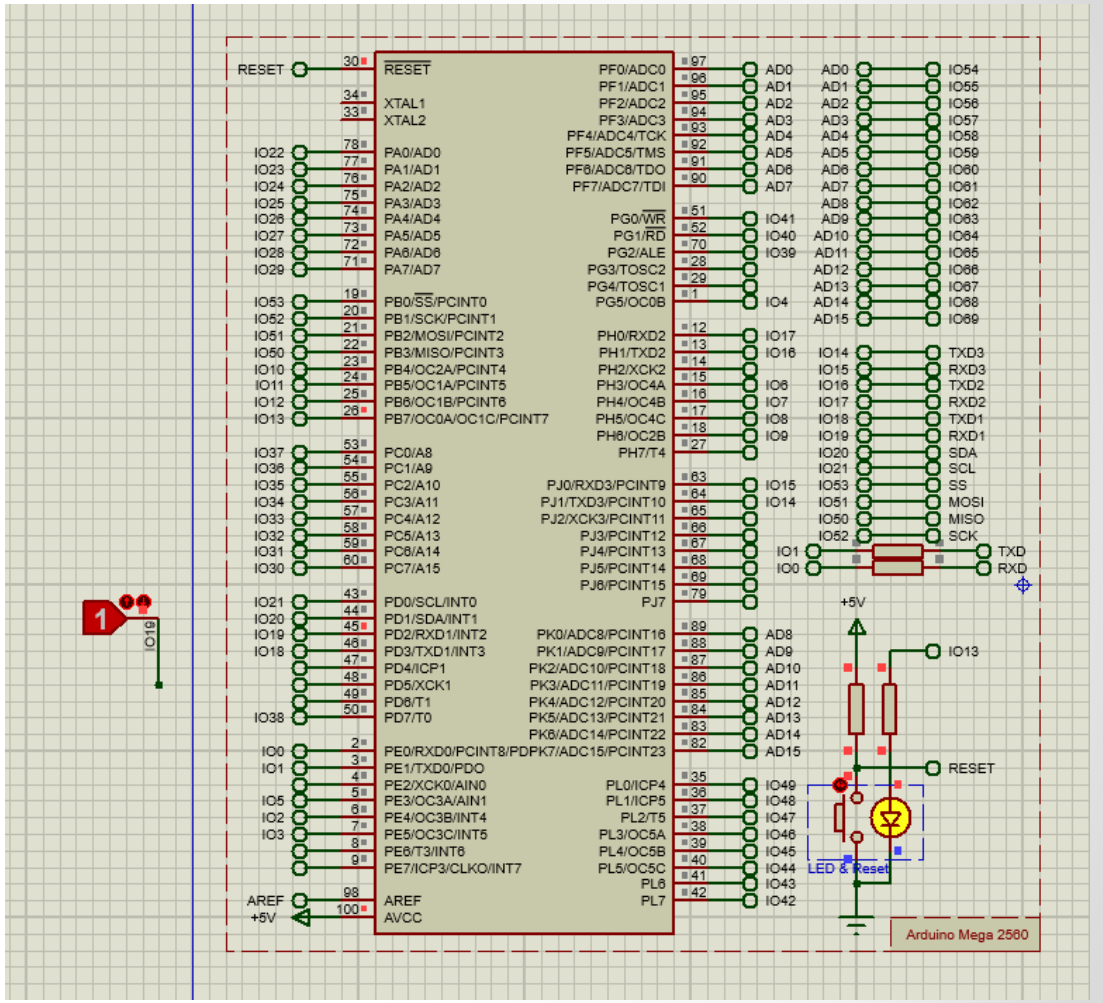
Primer 2

- Implementirati Arduino program koji promenom vrednosti nekog od pinova pali i glasi LED diodu prozivkom

```
const byte ledPin = 13;
const byte buttonPin = 19;
byte buttonValue;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}

void loop() {
  buttonValue = digitalRead(buttonPin);
  if (buttonValue == LOW){
     digitalWrite(ledPin, LOW);
  }
  else if (buttonValue == HIGH){
     digitalWrite(ledPin, HIGH);
  }
}
```

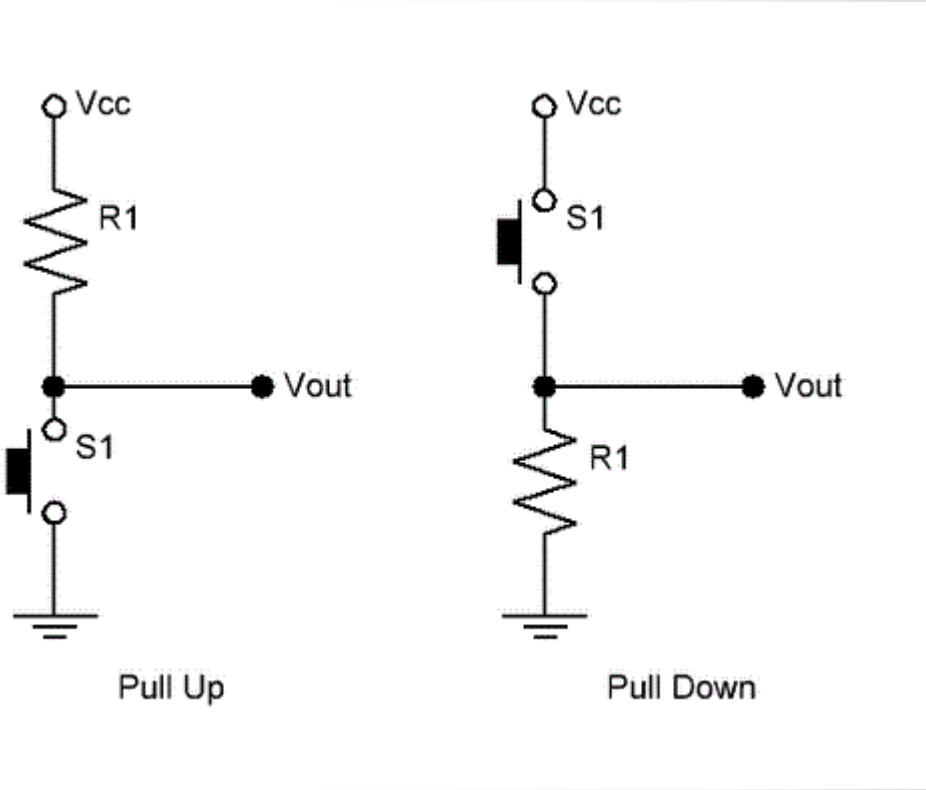


Pojam pull-up i pull-down otpornika

- Stanje visoke impedanse
  - Kada ulazni pin koji koristimo nije podešen na HIGH ili LOW, njegovo stanje "pluta"
  - Izbegavamo ovo stanje, dovodi potencijalno do nepredvidivosti u radu
- Kao rešenje – pull-up i pull-down otpornici
  - Ovo nisu posebne vrste otpornika
  - Omogućavaju da zadržimo željeno stanje logičkog kola, bez obzira na uslove
  - Arduino pin je vezan preko ovog otpornika na konstantnu vrednost
    - +5V napajanje: pull-up
    - GND (masa): pull-down

Nastavak: Pull-up i pull-down otpornici

- "Pull-up"
  - Levo na slici, omogućava držanje digitalnog ulaza (pina) na visokom naponskom nivou – HIGH
  - Vcc predstavlja izvor napajannja od 5V koji odgovara HIGH
  - Vout je izvod koji povezujemo na pin Arduina konfigurisanog kao ulazni
  - Sve dok ne pritisnemo taster S1, pin Arduina će biti HIGH
  - Kada pritisnemo – pin se povezuje sa masom i postavlja na niski naponski nivo LOW
  - Zbog osobine držanja HIGH stanja na pinu se zove pull-up
- "Pull-down"
  - Prikazan desno, služi održavanje LOW vrednosti
  - Povezan na masu (tačku nultog potencihala)
  - Onog trenutka kada bude pritisnut S1, pin se povezuje sa izvorom napjanja Vcc i dovodi na stanje HIGH



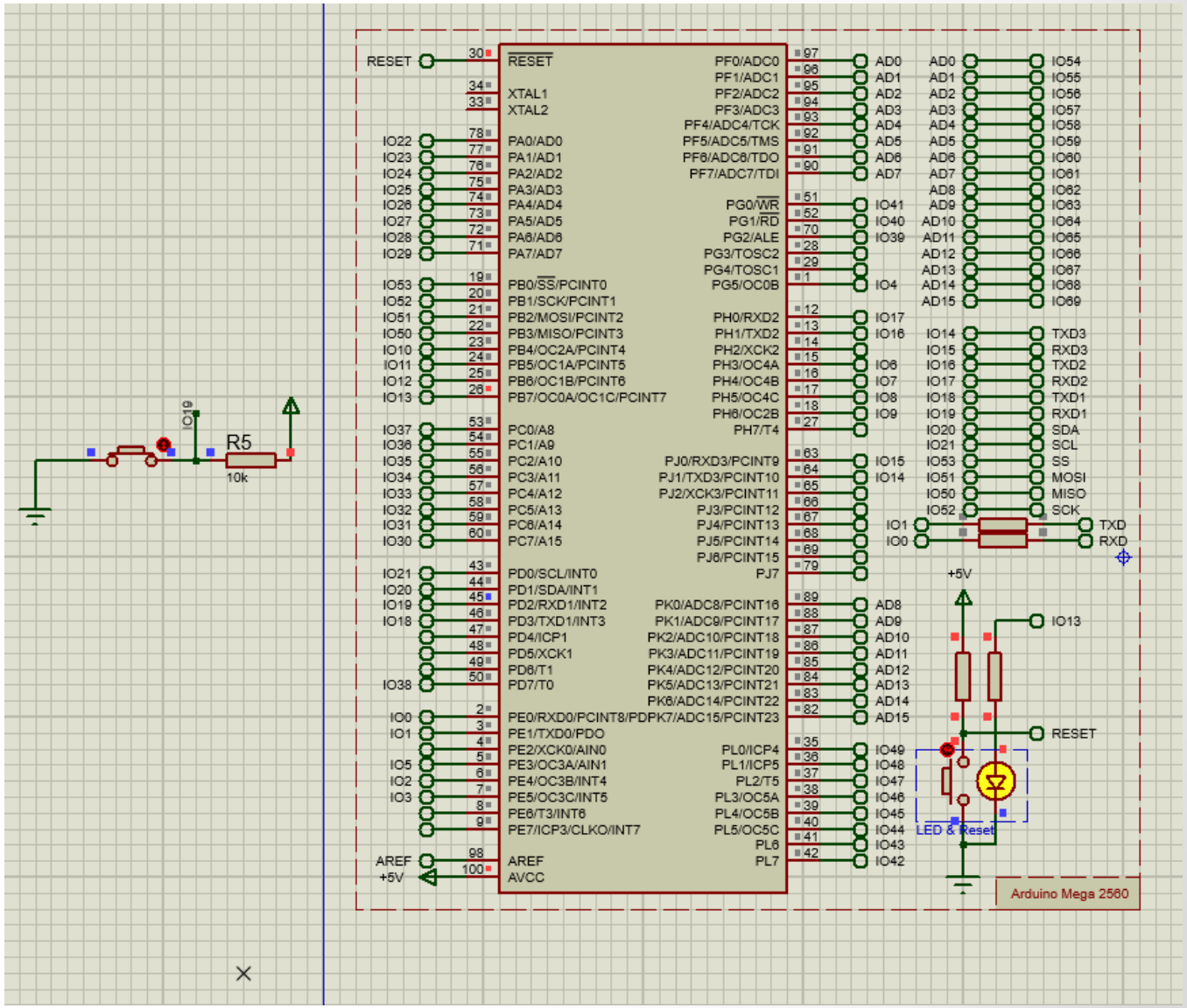
Primer 2b

- Implementirati Arduino program koji promenom vrednosti nekog od pinova pali i glasi LED diodu prozivkom – taster aktivan na 0
  - Pull-up otpornik

```
const byte ledPin = 13;
const byte buttonPin = 19;
byte buttonValue;
byte state = LOW;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}

void loop() {
  buttonValue = digitalRead(buttonPin);
  if (buttonValue == HIGH){
     state=LOW;
     digitalWrite(ledPin, LOW);
  }
  else if (buttonValue == LOW){
     state=HIGH;
     digitalWrite(ledPin, HIGH);
  }
}
```



Primer 3

- Implementirati Arduino program koji promenom vrednosti nekog od pinova koji je izvor prekida pali i glasi LED diodu.

```
const byte ledPin = 13;
const byte interruptPin = 19;
volatile byte state = LOW;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(interruptPin, INPUT);
  attachInterrupt(digitalPinToInterrupt(interruptPin), blink, CHANGE);
}

void loop() {
}

void blink() {
  state = !state;
  digitalWrite(ledPin, state);
}
```