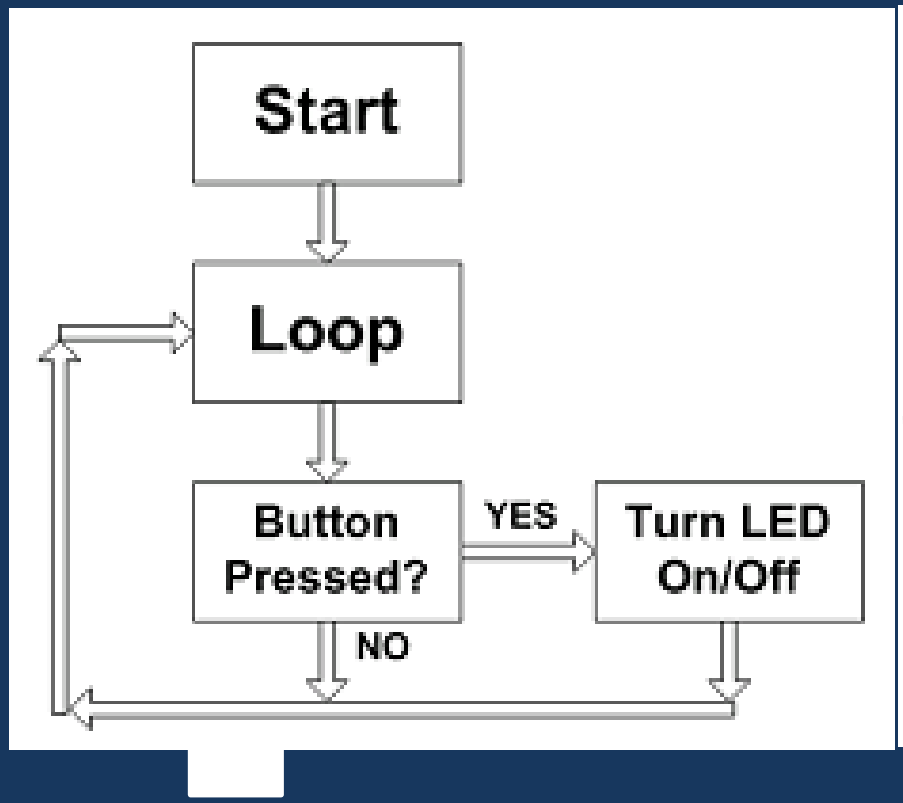
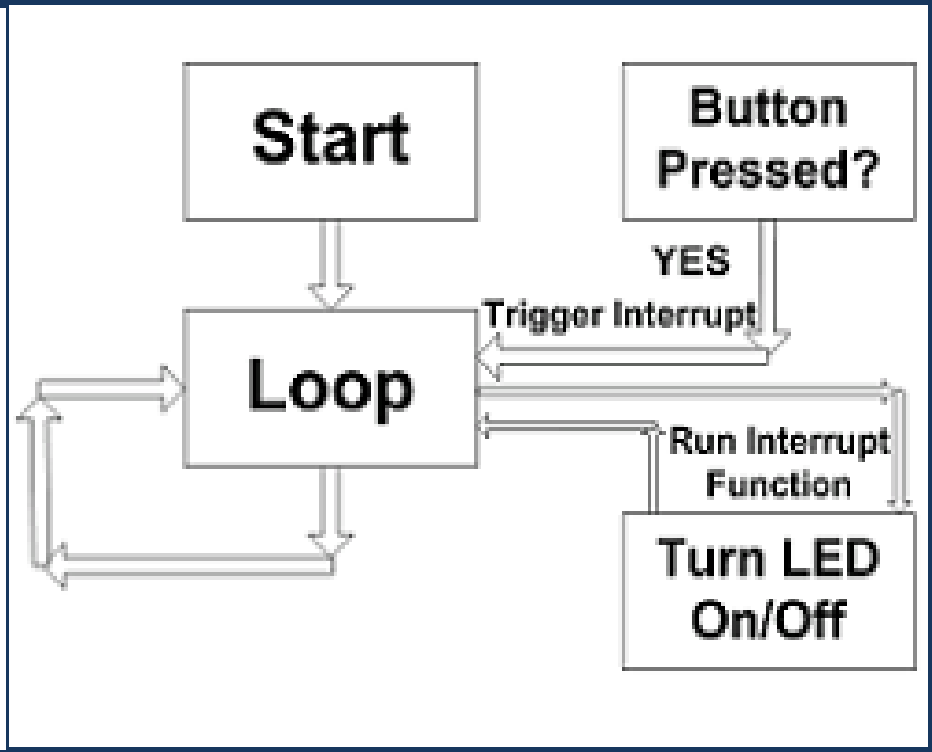
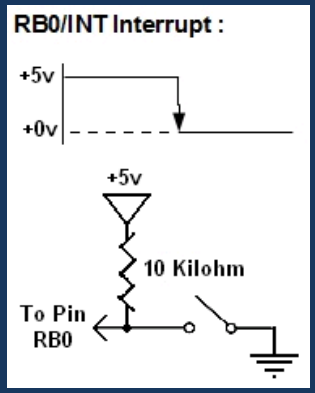
# Polling vs interrupt



# Hardware interrupt…

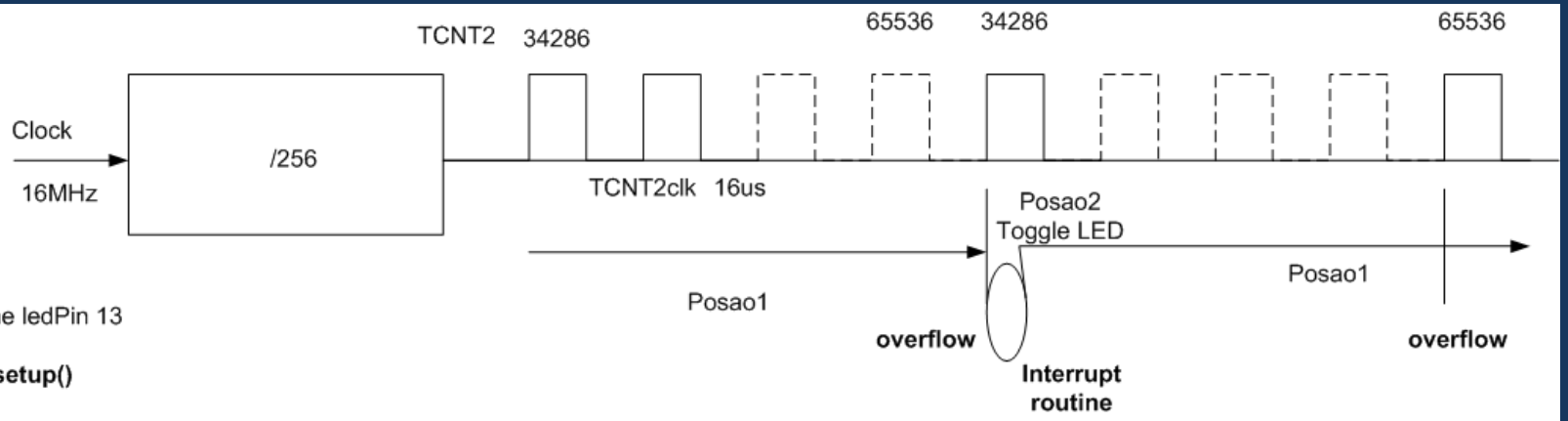
**Switch LED on in polling**

```
const int buttonPin = 2;     // the number of the pushbutton pin
const int ledPin =  13;      // the number of the LED pin

// variables will change:
int buttonState = 0;         // variable for reading the pushbutton
status

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}

void loop() {
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  }
  else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```

**Switch LED on in interrupt**

```
const int buttonPin = 2;     // the number of the pushbutton pin
const int ledPin =  13;      // the number of the LED pin

// variables will change:
volatile int buttonState = 0;         // variable for reading the pushbutton
status

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
  // Attach an interrupt to the ISR vector
  attachInterrupt(0, pin_ISR, CHANGE);
}

void loop() {
  // Nothing here!
}

void pin_ISR() {
  buttonState = digitalRead(buttonPin);
  digitalWrite(ledPin, buttonState);
}
```

# Software interrupt




```
#define ledPin 13

void setup()
{
pinMode(ledPin, OUTPUT);

// initialize Timer1
noInterrupts(); // disable all interrupts
TCCR1A = 0;
TCCR1B = 0;

TCNT1 = 34286; // preload timer 65536-16MHz/256/2Hz
TCCR1B |= (1 &lt;&lt; CS12); // 256 prescaler
TIMSK1 |= (1 &lt;&lt; TOIE1); // enable timer overflow interrupt
interrupts(); // enable all interrupts
}

ISR(Timer1_OVF_vect) // interrupt service routine that wraps a user defined
function supplied by attachInterrupt
{
TCNT1 = 34286; // preload timer
digitalWrite(ledPin, digitalRead(ledPin) ^ 1);
}

void loop()
{
// your program here…
}
```

# Integrisani i u realnom vremenu upravljani sistemi

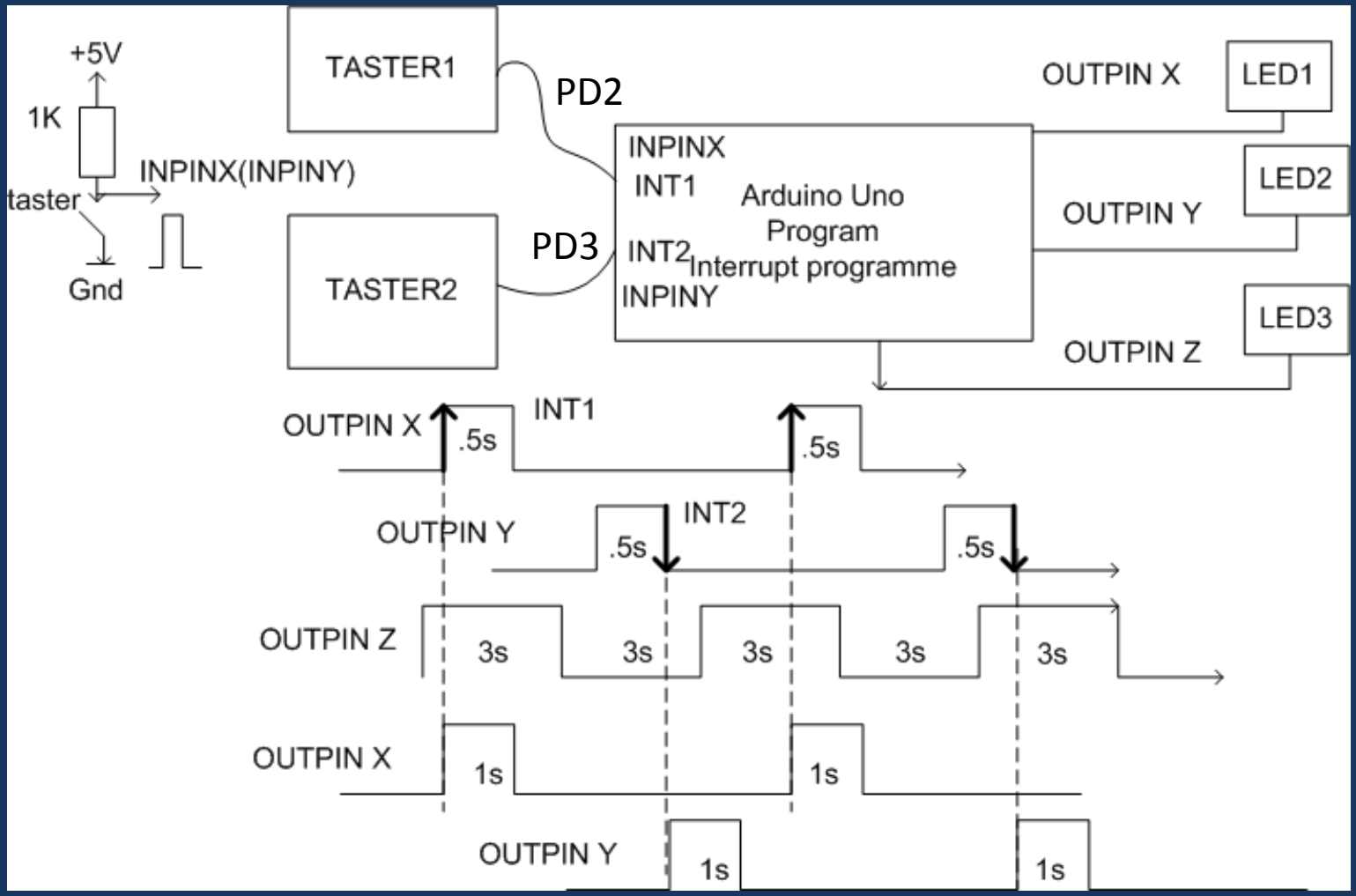
## Vježba 1 (Poling vs Interapt)

a) Povezati Taster1 i Taster2 na ulazne pinove Arduina, koji imaju i ulogu INT0 i INT1, kao sto je prikazano na slici. LED3 se pali (blinka ) iz glavne petlje programa u trajanu 3s sa pauzom 3s, period 6s. Napraviti program.
b) Iz glavne petlje se pritiskaju TASTER1 i TASTER2 u trajanju 0.5s , random, a nakon njihovog pritiskanja se pale LED1 i LED2 u trajanju 1s u zavisnoti da li je pritisnut TASTER1 ili TASTER2.
c) Komentarisati program i situaciju?
d) Napraviti program koji ipak pouzdano pali LED1 i LED2, u glavnoj petlji, bez obzira na duzinu blinka LED3.
e) Dati tasteri generisu hardverske interapte INT0 i INT1 koji pale LED1 ili LED2 u trajanju 1s. Paziti na ivicu na koju se generise interupti INT0 i INT1. Trajanje paljenja 1s (LED1 i LED2) generisati pomocu TIMER interrupta bez obzira da li je LED3 upaljena ili ugasena. Napraviti program.

**Priloziti kod i video objasnjavajuci gornje slucajeve.**

Procitati: ostali materijali interrupt primjer1, skripta Tanskovic,
HARVER INT http://apeg.ac.me/nastava/Arduino%20Hardware%20Interrupts%20Tutorial.pdf ,
TIMER Interrupt: http://www.instructables.com/id/Arduino-Timer-Interrupts/
http://apeg.ac.me/nastava/VJEZBA2%20Hardverski%20interapti%20-%20Arduino.pdf

# Integrisani i u realnom vremenu upravljani sistemi



# Integrisani i u realnom vremenu upravljani sistemi

- **Vjezba 2 (Semplovanje u polling metodu)**

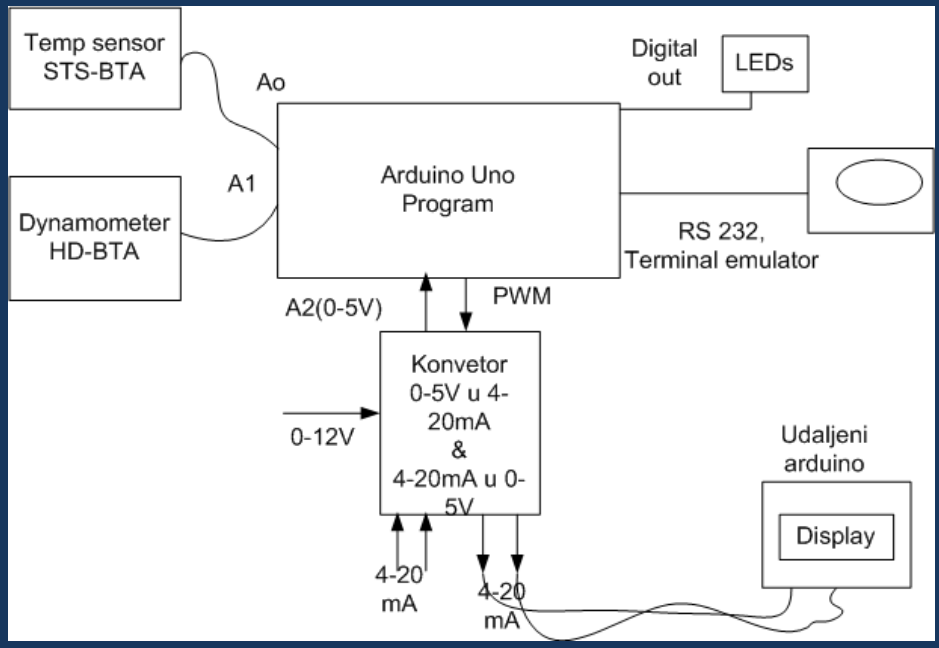
**a)** Povezivanje temperaturnog senzora na Arduino, prikaz izmjerene temperature na terminal emulatoru i monitoru grafičkom i poredjenje sa alarmnom vrijednoscu (kada se pali LED). Semplovanje u poling metodu.

**b)** Dinamometra na Arduino, prikaz izmjerene vrijednosti sile na terminal emulatoru i monitoru grafičkom i poredjenje sa alarmnom vrijednoscu (kada se pali LED). Semplovanje u poling metodu.

c) Ponoviti a) i b) za slucaj senzora provodnosti

d) Ponoviti a) i b) za slucaj PH senzora.

**e)** Mijenjati frekvenciju odabiranja od 0.01s do 1s.




SENZORI:
https://www.vernier.com/files/manuals/sts-bta/sts-bta.pdf
https://www.vernier.com/files/manuals/hd-bta/hd-bta.pdf
https://www.vernier.com/products/sensors/ph-sensors/ph-bta/
https://www.vernier.com/products/sensors/conductivity-probes/con-bta/

Konverter industrijski visenamjenski

http://apeg.ac.me/nastava/Converter%20ind%20multi%20choice.pdf

# Integrisani i u realnom vremenu upravljani sistemi

- **Vjezba 3  ( (Semplovanje na interrupt nivou)**

Ponoviti vjezbu 2 sa semplovanjem na interrapt nivou, koristiti timer interrupt.  Frekvenciju odabiranja zadavati prego serijske komunikacije (serijskog interapta)