



# IFI6203 ROBOOTIKA

Praktikum 6



# Katkestuse kasutamine Arduino juhtimiseks

- Katkestus on signaal, mis ütleb kontrolleri, et hetkel käimas olev tegevus tuleb peatada ja teha midagi muud.
- See, mida tehakse, sõltub katkestusest - katkestusega on seotud funktsioon, mis kutsutakse katkestuse puhul välja.
- Peale katkestusega seotud funktsiooni täitmist jätkab kontrolleri oma tööd sealt, kust see pooleli jäi.

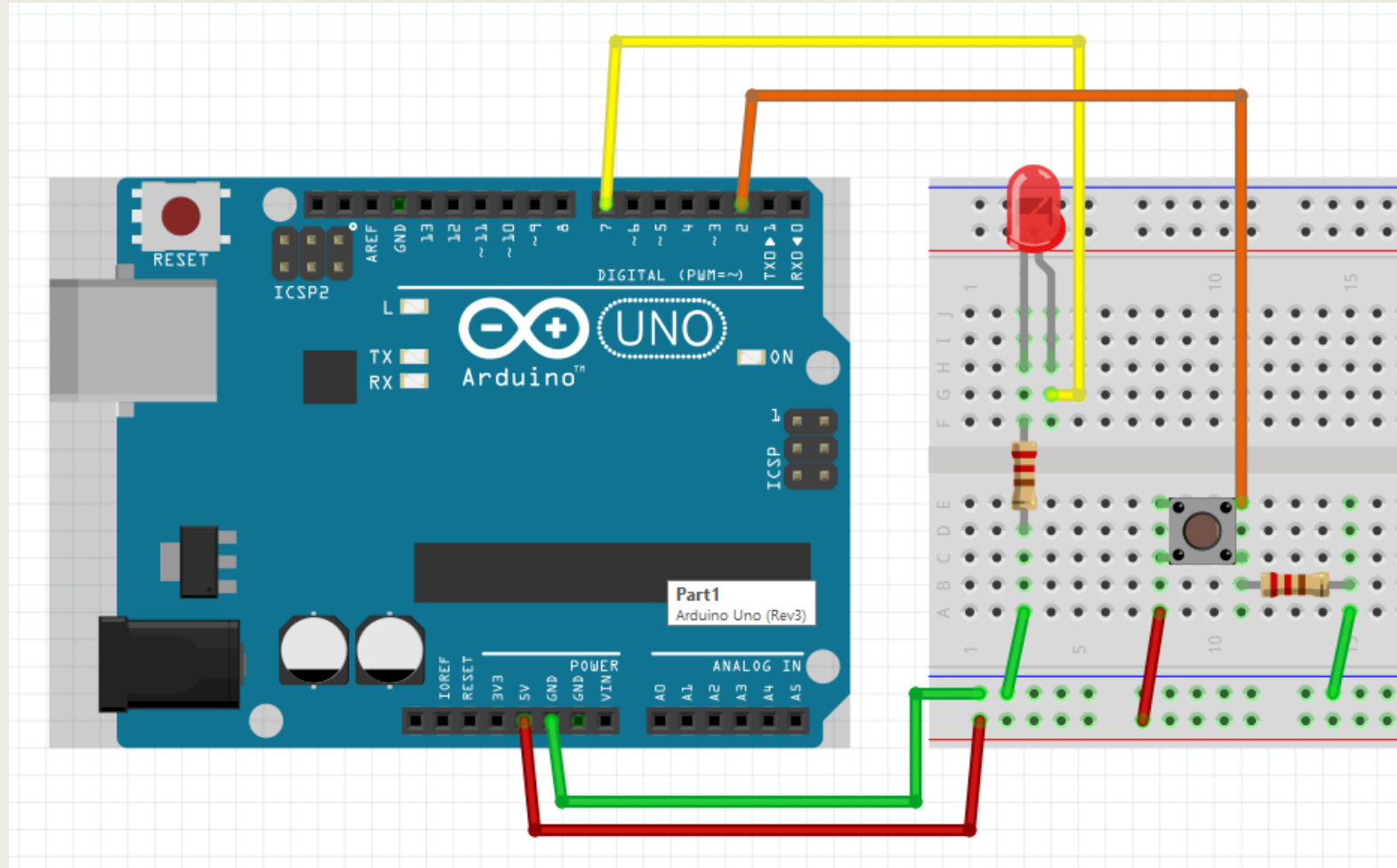
# Katkestused Arduino UNOga

- Arduino UNO puhul on väliste katkestuste vastuvõtmise võimelised digitaalsed viigud 2 ja 3
- Katkestusega seotav funktsioon ei saa võtta vastu parameetreid ega tagasta midagi
- Katkestusega seotava funktsiooni sees ei tööta ajastusega seotud funktsioonid nagu *delay()* ja *millis()*
- Katkestuseda seotava funktsiooni poolt mõjutatavad muutujad tuleb deklareerida kui *volatile*

# Katkestuse päästiksignaaliid

- Arduino UNO tuvastab järgmisi katkestuse päästiksignaale:
  - *LOW* - signaal on LOW (0V)
  - *CHANGE* - signaali olek muutub
  - *RISING* - signaal muutub olekust LOW olekusse HIGH
  - *FALLING* - signaal muutub olekust HIGH olekusse LOW

# Nupp ja LED koos katkestusega



# Katkestuse näitekood

```
int ledPin = 7;
int interruptPin = 2;
volatile byte olek = LOW;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(interruptPin, INPUT);
  attachInterrupt(digitalPinToInterrupt(interruptPin), vilguta, RISING);
}

void loop() {
  digitalWrite(ledPin, olek);
}

void vilguta() {
  olek = !olek;
}
```

# Andmete saatmine Arduino Serial Monitori kaudu

- Arduino IDE Serial Monitor võimaldab võtta vastu ja saata andmeid jadapordi kaudu.
- Jadapordi objektile on puhver (64 baiti), kus andmeid hoitakse. Serial Monitor kirjutab kontrollerilt arvutile saadetud andmed vaikimisi puhvrist kasutajaliidese vastavasse aknasse.
- Arduino IDE liidese kaudu võib andmeid saata ka tagasi kontrollerile.
- Kontrollimaks, kas kontrollerile on andmeid saadetud võib kasutada `Serial.available()` funktsiooni

# Serial Monitori kaudu andmete saatmine Arduinole

```
String k2sk;

void setup() {
  Serial.begin(9600);
  Serial.println("ootan kaske: ");
}

void loop() {
  if (Serial.available() > 0) {
    k2sk = Serial.readString();
    Serial.print("Sain kasu: ");
    Serial.println(k2sk);
  }
  if(k2sk == "vilguta"){
    digitalWrite(LED_BUILTIN, HIGH);
    delay(1000);
    digitalWrite(LED_BUILTIN, LOW);
    delay(1000);
  }
}
```



# Azande Studio

- Azande Studio võimaldab Arduino-le lihtsa graafilise kasutajaliidese ehitamist
- Andmeid saadetakse üle jadaliidese, mis võib toimida üle virtuaalse COM pordi (tekib Arduino ühendamisel USB kaabliga) või Bluetooth ühenduse (kui meil on vastava võimekusega Arduino kontrolleri).
- Azande Studio ja selle Arduino teek on allalaetavad siit:  
<https://zeijlonsystems.se/products/azande/download.html>
- Teek tuleb paigaldada käsitsi
- Azande abimaterjalid:  
<https://zeijlonsystems.se/products/azande/specification/Introduction.html>

# Azande näitekode

```
#include <Azande.h>

//tekitame kasutajaliidese käsuobjekti, mis kutsus funktsiooni setBlinkState boolean parameetriga
define_bool_command(cmdLedBlink, "LED vilgub", setBlinkState, , "Jah", "Ei" );

//tekitame kasutajaliidese sündmuseobjekti, mis kuvab LEDi olekut
define_bool_event(eventLedState, "LEDi olek", , "Põleb", "Ei Põle");

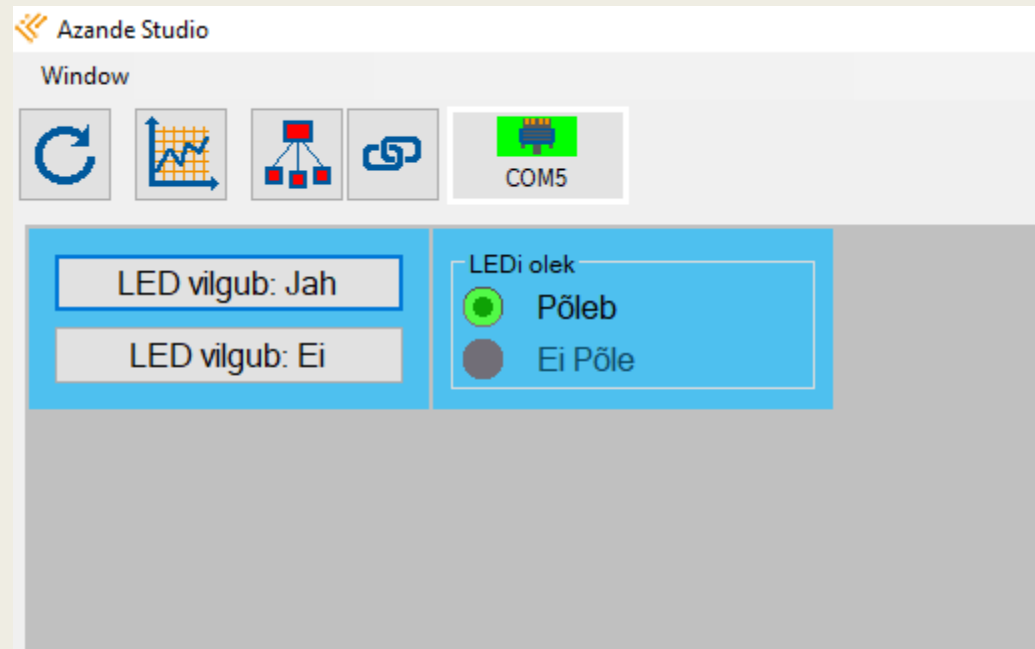
boolean LEDstate=false; //vaikimisi on LED kustus
boolean vilgub=false; //vaikimisi LED ei vilgu
Azande azande(Serial); //tekitame azande andmeühenduse üle jadaliidese

void setup()
{
  Serial.begin(19200); //käivitame jadaliidese
  azande.begin(); //käivitame azande ühenduse
  azande.add(cmdLedBlink); //lisame kasutajaliidese käsuobjekti
  azande.add(eventLedState); //lisame kasutajaliidese sündmuseobjekti
}

void loop()
{
  azande.send(eventLedState, LEDstate); //saadame azande sündmuseobjektile LEDi oleku
  digitalWrite(LED_BUILTIN, LEDstate); //paneme Arduino LEDi tekelikult ka sellesse olekusse
  if(vilgub){ //vastavalt muutuja vilgub olekule muudame LEDi olekut või jätame selle samaks
    LEDstate=!LEDstate;
  }
  azande.readStream(); //Vaatame, kas Azande kasutajaliidest on käsk saadetud ja kui on siis tegeleme nendega
  delay(1000);
}

//seda funktsiooni kutsus välja kasutajaliidese käsuobjekt. Funktsioon seab muutuja "vilgub" väärtuse vastavusse
//kasutajaliidest valituga.
void setBlinkState(boolean value)
{
  vilgub=value;
}
```

# Azande kasutajaliidese näide



# Arduino ja Processing

- Processing on keel, millega on lihtne luua visuaalseid elemente
- Tarkvara on allalaetav siit:
  - <https://processing.org/download/>
- Arduino+Processing:
  - <https://playground.arduino.cc/Interfacing/Processing/>
- Ühendame Arduino Processing tarkvaraga
  - <https://learn.sparkfun.com/tutorials/connecting-arduino-to-processing/all>
- Veel üks näide:
  - <https://electronicshobbyists.com/arduino-processing-tutorial-connect-arduino-to-processing/>