
Računalnik v pouku Fizike

Cenovno dostopen merilni sistem z Arduino UNO
krmilnikom

DR. DAVID RIHTARŠIČ

2019-04-15

Kazalo

1	Merilni sistemi	4
1.1	Importance of real-life experiments for students	4
1.2	Parts of a DAQ System [[NationalInstruments]]	4
1.3	What Is a Sensor?	4
1.4	What Is a DAQ Device?	5
1.5	What Is a Computer's Role in a DAQ System?	5
2	Merilne naprave	6
2.1	Vernier's DAQs	6
2.2	Arduino Data Acquisition System	8
2.3	Requirements	10
2.4	Pros & Cons	11
3	Teoretične osnove	12
3.1	Frekvenca vzorčenja	12
3.2	Digitalizacija	13
3.3	Resolucija in ločljivost	14
3.4	Točnost in preciznosti (natančnost)	14
3.5	Normalna porazdelitev	15
3.6	Povprečna vrednost	16
3.7	Standardni odklon	16
3.8	Območje zaupanja	16
4	Arduino UNO	17
4.1	Arduino UNO/NANO pinout	18
4.2	Osnove programiranja krmilnika	19
5	Merjenje s krmilnikom Arduino UNO	21
5.1	Merjenje napetosti	22
5.2	Izračun napetosti	22
5.3	Normalna porazdelitev meritev	22
5.4	Koeficienti normalne porazdelitve	23
5.5	Časovne meritve	24
5.6	Uporaba digitalnih vhodov	24
5.7	Hitrost	26
5.8	Pospešek	26

6	Prikazovanje podatkov	27
6.1	Serijski izpis	27
6.2	Grafični izris	28
6.3	Izpis na LCD	28
7	Linearizacija meritvenih vrednosti (praznenje kondenzatorja)	30
7.1	Vezje in zajem podatkov	30
7.2	Linearizacija	31
7.3	Koeficienti regresijske premice	31
8	Senzorji	32
8.1	Občutljivost	32
8.2	Delilniki napetosti s spremenljivim uporom	32
9	Umerjanje (temperaturnega) senzorja	33
9.1	Umeritveni postopek	34
9.2	Interpolacija	35
9.3	Izpis temperature (stand-alone DAQ)	36
10	Projektna naloga	37
11	Primeri	38
11.1	Merjenje srčnega utripa	38
11.2	Točkovanje	39
11.3	Točke:	39
	Viri in literatura	40

1 Merilni sistemi

Data acquisition is the process of sampling signals that measure real world physical conditions and converting the resulting samples into digital numeric values that can be manipulated by a computer [[Wikipedia]]. Data acquisition systems, abbreviated by the acronyms DAS or DAQ, typically convert analog waveforms into digital values for processing. The components of data acquisition systems include: - Sensors, to convert physical parameters to electrical signals. - Signal conditioning circuitry, to convert sensor signals into a form that can be converted to digital values. - Analog-to-digital converters, to convert conditioned sensor signals to digital values.

Data acquisition applications are usually controlled by software programs developed using various general purpose programming languages such as Assembly, BASIC, C, C++, C#, Fortran, Java, LabVIEW, Lisp, Pascal, etc. Stand-alone data acquisition systems are often called data loggers [[Wikipedia]].

There are also open-source software packages providing all the necessary tools to acquire data from different hardware equipment. These tools come from the scientific community where complex experiment requires fast, flexible and adaptable software. Those packages are usually custom fit but more general DAQ package like the Maximum Integrated Data Acquisition System can be easily tailored and is used in several physics experiments worldwide [[Wikipedia]].

1.1 Importance of real-life experiments for students

- where theory finds its place
 - greater motivation for learning
 - more (learning) input → better understanding
-
- instant data processing
 - explicit presentation of interdependent variables
 - visualization of physical quantities that are detected by human nature

1.2 Parts of a DAQ System [[NationalInstruments]]

1.3 What Is a Sensor?

The measurement of a physical phenomenon, such as the temperature of a room, the intensity of a light source, or the force applied to an object, begins with a sensor. A sensor, also called a transducer, converts a physical phenomenon into a measurable electrical signal. Depending on the type of sensor, its electrical output can be a voltage, current, resistance, or another electrical attribute that varies over

time. Some sensors may require additional components and circuitry to properly produce a signal that can accurately and safely be read by a DAQ device [[NationalInstruments]]. Kot v enačbi ?? in na sliki ??.

(„What Is Data Acquisition? - National Instruments“, b. d.)

1.4 What Is a DAQ Device?

DAQ hardware acts as the interface between a computer and signals from the outside world. It primarily functions as a device that digitizes incoming analog signals so that a computer can interpret them. The three key components of a DAQ device used for measuring a signal are the signal conditioning circuitry, analog-to-digital converter (ADC), and computer bus. Many DAQ devices include other functions for automating measurement systems and processes. For example, digital-to-analog converters (DACs) output analog signals, digital I/O lines input and output digital signals, and counter/timers count and generate digital pulses[NationalInstruments].

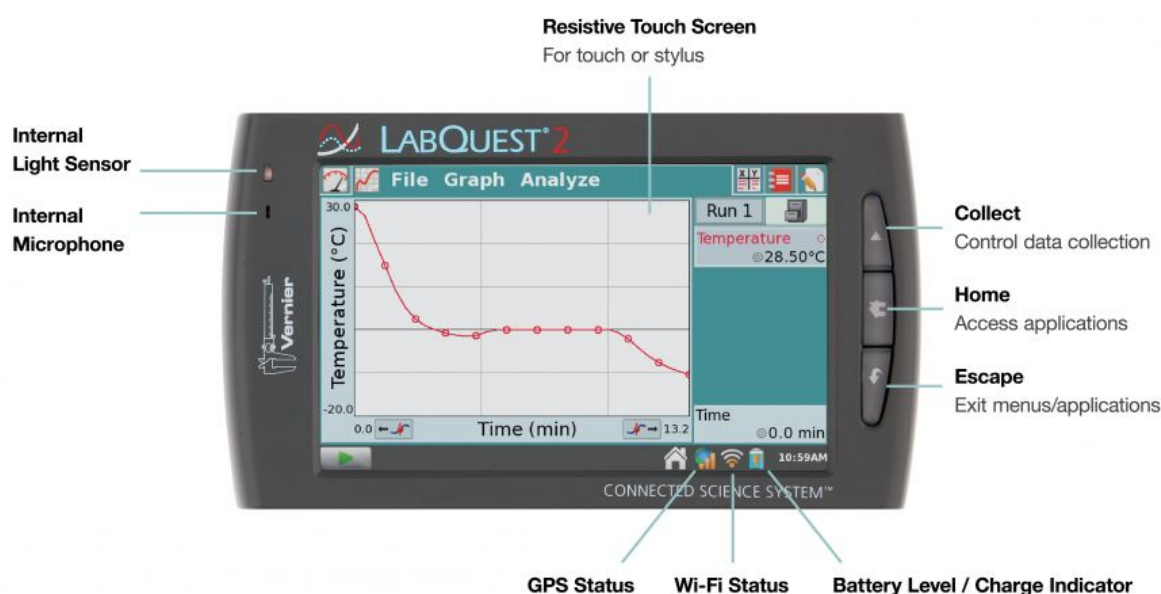
1.5 What Is a Computer's Role in a DAQ System?

A computer with programmable software controls the operation of the DAQ device and is used for processing, visualizing, and storing measurement data. Different types of computers are used in different types of applications. A desktop may be used in a lab for its processing power, a laptop may be used in the field for its portability, or an industrial computer may be used in a manufacturing plant for its ruggedness („What Is Data Acquisition? - National Instruments“, b. d.).

2 Merilne naprave

Pri fizikalnih eksperimentih pogosto uporabljamo merilne naprave, s katerimi želimo potrditi teoretične principe in fizikalne pojave. V šolskem prostoru najpogosteje uporabljamo Vernier-jevo merilno napravo LabQuest Hardware („LabQuest® Hardware and Specifications | Vernier“, b. d.) , ki nam omogoča raznovrstne meritve.

2.1 Vernier's DAQs



Slika 1: Vernier-jeva merilna naprava LabQuest2

Poglejmo si nekaj podatkov o napravi:

- **Display**
 - 11.2 cm x 6.7 cm (13.1 cm diagonal) screen
 - 800 x 480 pixel color display at 188 dpi
 - LED backlight
 - portrait or landscape screen orientation
 - high-contrast mode for outdoor visibility
- **Processor**
 - 800 MHz Application Processor
- **Connectivity**

- Wi-Fi 802.11 b/g/n @ 2.4GHz
- Bluetooth Smart for WDSS and Go Wireless Sensors
- **User Interface**
 - Resistive touch screen
 - Touch and stylus navigation for efficiency and precision
- **Data Acquisition**
 - 100,000 samples per second
 - 12-bit resolution
 - Built-in GPS, 3-axis accelerometer, ambient temperature, light, and microphone
- **Environmental Durability**
 - Operating Temperature: 0 + 45°C
 - Storage Temperature: -30 + 60°C
 - Splash resistant
 - Rugged enclosure designed to withstand a fall from lab bench
- **Size and Weight**
 - Size: 8.8 cm x 15.4 cm x 2.5 cm
 - Weight: 350 g
- **Ports**
 - 5 sensor channels
 - USB port for sensors, flash drives, and peripherals
 - USB mini port
 - DC power jack
 - MicroSD/MMC slot
 - Audio in and out
- **Storage**
 - 200 MB
 - Expandable with MicroSD and USB flash drive
- **Power**
 - Rechargeable, high-capacity battery
 - DC charging/powering through external adapter (included)
- **cena:**
 - \$455

Seveda pa morate dokupiti še senzorje, ki tudi niso cenovno ugodni, saj se njihova cena giblje od \$30 ... naprej.

2.2 Arduino Data Acquisition System

On the market we can find different DAQ systems which are hi-end products and often expensive (from 100 € .. n k€). But we can use Arduino (Uno, nano, ...) as low-cost data acquisition system if we do not need scientific accurate data (for pedagogical purposes).

<- Parts of DAQ[[NationalInstruments]] ->

Na plošči vsebuje mikrokontroler **Atmega328**, ki lahko opravi podobne naloge, kot smo jih opisali v poglavju **Merilne naprave**.

Poglejmo si nekaj karakteristik tega mikrokontroler (Atmel 2017):

- **Advanced RISC Architecture**
 - 131 Powerful Instructions
 - Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 20 MIPS Throughput at 20MHz
 - On-chip 2-cycle Multiplier
- **High Endurance Non-volatile Memory Segments**
 - 32KBytes of In-System Self-Programmable Flash program Memory
 - 1KBytes EEPROM
 - 2KBytes Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data Retention: 20 years at 85°C/100 years at 25°C(1)
 - Optional Boot Code Section with Independent Lock Bits
 - * In-System Programming by On-chip Boot Program
 - * True Read-While-Write Operation
 - Programming Lock for Software Security
- **Atmel® QTouch® Library Support**
 - Capacitive Touch Buttons, Sliders and Wheels
 - QTouch and QMatrix® Acquisition
 - Up to 64 sense channels
- **Peripheral Features**
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Six PWM Channels
 - ADC
 - * 8-channel 10-bit ADC in TQFP and QFN/MLF package

- * Temperature Measurement
- * 6-channel 10-bit ADC in PDIP Package
- * 10-bit Resolution
- * 0.5 LSB Integral Non-Linearity
- * ± 2 LSB Absolute Accuracy
- * 13 - 260 μ s Conversion Time
- * Up to 76.9kSPS (Up to 15kSPS at Maximum Resolution)
- * Six Multiplexed Single Ended Input Channels
- * Two Additional Multiplexed Single Ended Input Channels (TQFP and VFQFN Package only)
- * Temperature Sensor Input Channel
- * Optional Left Adjustment for ADC Result Readout
- * 0 - VCC ADC Input Voltage Range
- * Selectable 1.1V ADC Reference Voltage
- * Free Running or Single Conversion Mode
- * Interrupt on ADC Conversion Complete
- * Sleep Mode Noise Canceler
- Two Master/Slave SPI Serial Interface
- One Programmable Serial USART
- One Byte-oriented 2-wire Serial Interface (Philips I2C compatible)
- Programmable Watchdog Timer with Separate On-chip Oscillator
- One On-chip Analog Comparator
- Interrupt and Wake-up on Pin Change
- **Special Microcontroller Features**
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- **I/O and Packages**
 - 23 Programmable I/O Lines
 - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- **Operating Voltage:**
 - 1.8 - 5.5V
- **Temperature Range:**
 - -40°C to 105°C
- **Speed Grade:**
 - 0 - 4MHz @ 1.8 - 5.5V

- 0 - 10MHz @ 2.7 - 5.5V
- 0 - 20MHz @ 4.5 - 5.5V
- **Power Consumption at 1MHz, 1.8V, 25°C**
 - Active Mode: 0.2mA
 - Power-down Mode: 0.1µA
 - Power-save Mode: 0.75µA (Including 32kHz RTC)

2.3 Requirements

(Checked is required, unchecked is optional)

Hardware:

- ☒ Computer
- ☒ Arduino board (Arduino UNO, Arduino LEONARDO, Arduino NANO, clone products)
- ☒ basic electronics components for sensors
- ☐ arduino set starter for example use this kit
- ☐ arduino sensors set for example use this kit

Software:

- ☒ Arduino IDE download here
- ☐ Python (≥ 3.0)
- ☐ pyserial
- ☐ Ms Excel

2.4 Pros & Cons

Pros	Cons
+ Price (Arduino ~3€, sensors ~2-5€)	- work in progress
+ Accessibility	- no plug&play solutions
+ Versatility	- DIY project
+ Easy importing data into MS Excel	- low sampling rate ~6kHz
+ točne časovne meritve	- 10-bi resolutuin

More about DAQ you can read in further reading...

Kaj ko bi si lahko naredili svojo merilno napravo?

V ta namen smo ustvarili nekaj vsebin na portalu GitHub...

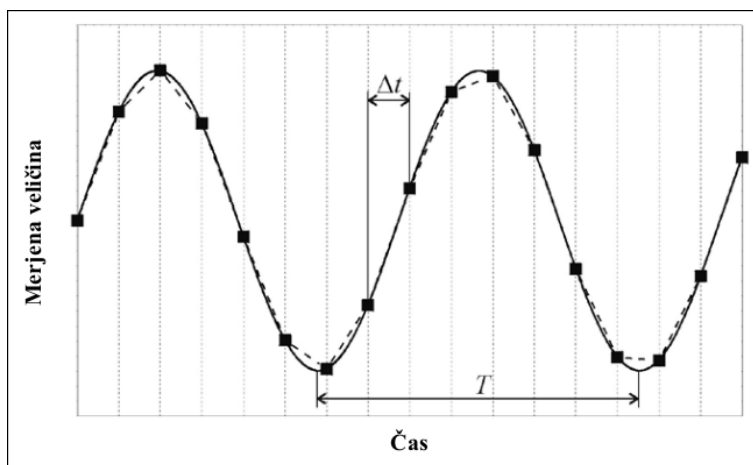
3 Teoretične osnove

3.1 Frekvenca vzorčenja

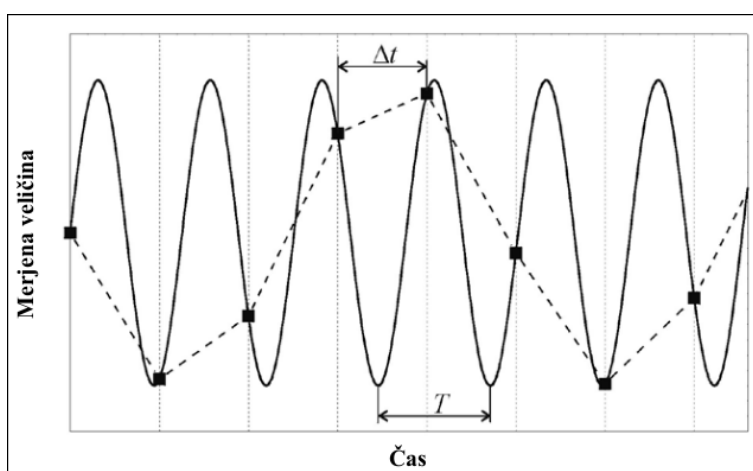
Pri vzorčenju signalov je zelo pomembna frekvenca vzorčenja f_{vz} . Upoštevati moramo Nyquistovo načelo vzorčenja, ki pravi, da je potrebno periodične signale vzorčiti vsaj z 2x večjo frekvenco vzorčenja kot je frekvenca signala f_{sig} (Nyquist 1928), kot to predstavlja slika 2.

$$f_{vz} = 2f_{sig} \quad (1)$$

V nasprotnem primeru lahko dobimo nepravilno reprodukcijo merjenega signala (črtkana krivulja), kot to prikazuje slika 3.



Slika 2: Pravilno vzorčenje sinusnega signala.



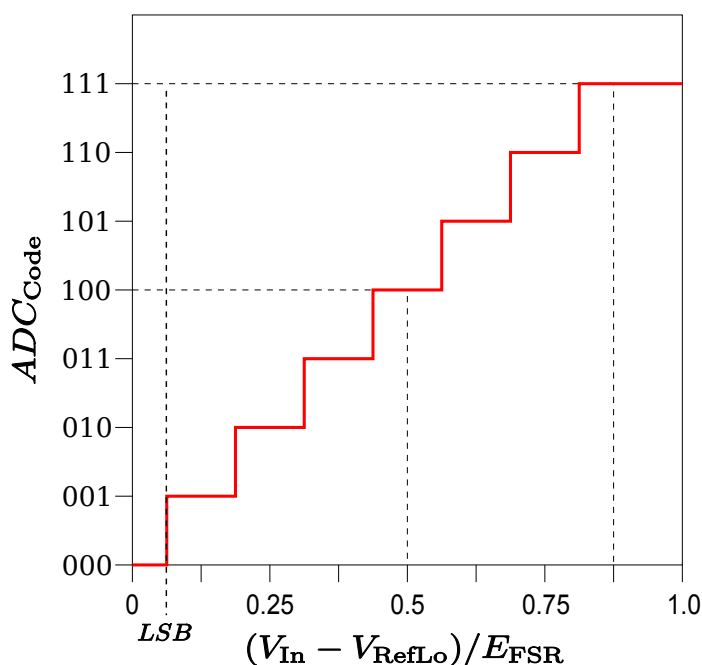
Slika 3: Primer reprodukcije (črtkana krivulja) podvzorčenega signala.

Naloga 1: Frekvenca vzorčenja.

Glede na prejšnje podatke o mikrokrmilniku Atmega328 poiščite podatek o najvišji frekvenci vzorčenja f_{vz} analognih signalov in izračunajte najmanjši čas Δt med dvema vzorčenjema.

3.2 Digitalizacija

Merilni sistemi so opremljeni s t.i. analogno-digitalnimi pretvorniki (ang.: Analog-to-digital converter - ADC), ki pretvarjajo merjeno napetost v neko številsko vrednost. Zelo pogost primer je, ko zvezno napetostno območje od $0,0V - 5,0V$ pretvorimo v številske vrednosti od 0 - 1023. Pri tej pretvorbi ključno vlogo prevzame ADC in njegova **resolucija**. Grafični prikaz take transformacije je prikazan na sliki 4.



Slika 4: Prenosna funkcija ADC pretvorbe (Contributors 2019a).

3.3 Resolucija in ločljivost

Resolucija AD pretvornikov je določena s številom vseh možnih stanj pretvorbe N . Ker so AD pretvorniki napreave prirejene digitalnim tehnologijam, se njihovi podatki izražajo v dvojiški obliki (binarno). Tako na primer AD pretvornik z 10-bitno pretvorbo lahko prikaže:

$$N = 2^B \quad (2)$$

možnih stanj.

Naloga 2: Resolucija AD pretvornika.

Izračunajte s kolikšno resolucijo lahko odčitavamo analogne signale z mikrokrmilnikom Atmega328.

Ločljivost pa je najmanjša razlika med sosednjima digitaliziranimi vrednostima merjene količine. Ta vrednost je odvisna tako od števila možnih stanj N , kakor tudi od območja, ki ga pretvarjamo. Zato bi lahko enačbo 3 zapisali:

$$Ločljivost = \frac{Območje}{N} \quad (3)$$

Naloga 3: Ločljivost AD pretvornika.

Izračunajte kolikšna je ločljivost mikrokrmilnika Atmega328 pri odčitavanju analognih signalov.

3.4 Točnost in preciznosti (natančnost)

Točnost (v različnih virih je poimenovana različno, ang.: validity) je lastnost merilnega sistema, ki predstavlja ustreznost prestavljene meritve glede na njeno realno merjeno vrednost. Navadno jo izra-

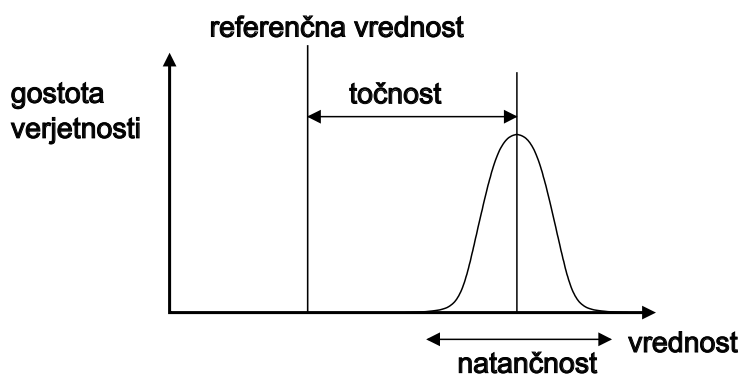
žamo kot relativno napako ϵ v procentualni obliki (enačba 4):

$$\epsilon = \frac{(X - x_n)}{X} \quad (4)$$

Kjer je X realna merjena vrednost in x_n izmerjena vrednost.

Preciznost oz. natančnost (zopet v različnih literaturah poimenovana različno, ang.: reliability) je sposobnost merilnega sistema reprodukcije iste merjene (referenčne) vrednosti z enakimi izmerjenimi vrednostmi. V mnogih primerih se izkaže, da gre v tem primeru za naključno napako merjenja in to vrednost lahko ponazarjamo s standardnim odklonom merilnega postopka (enačba 7).

V splošnem bi lahko točnost in natančnost predstavili z grafom na sliki 5 (Contributors 2019c).



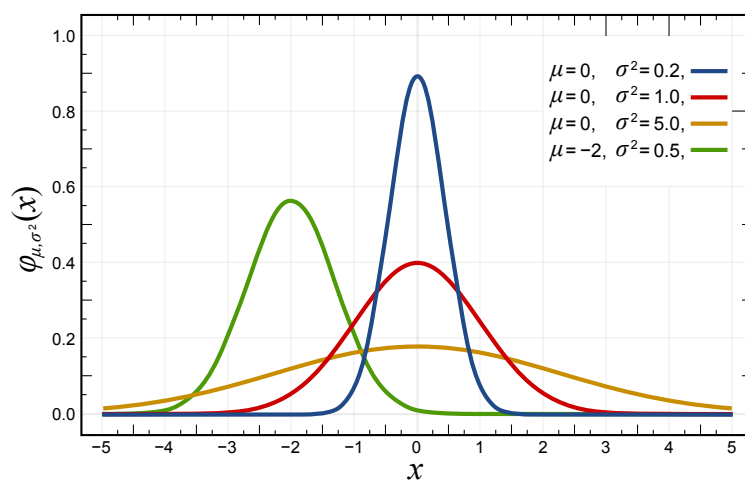
Slika 5: Točnost in natančnost meritev.

3.5 Normalna porazdelitev

Kadar imamo v merilnem sistemu opravka z naključnimi napakami, meritve lahko predstavimo s krivuljo normalne porazdelitve - v splošnem imenovane Gaussova porazdelitev. Zapišemo jo v obliki enačbe 5.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (5)$$

Kjer je μ povprečna vrednost in σ^2 varianca. Nekaj različnih krivulj lahko vidimo na sliki 6 (Contributors 2019b).



Slika 6: Primeri normalne verjetnostne porazdelitve.

3.6 Povprečna vrednost

$$\bar{x} = \frac{\sum x_n}{n} \quad (6)$$

3.7 Standardni odklon

$$\sigma = \sqrt{\frac{\sum_{n=1}^N (x_n - \bar{x})^2}{n - 1}} \quad (7)$$

3.8 Območje zaupanja

Z intervalom zaupanja predstavlja območje meritev, v katerem se naključna meritev pojavi z neko verjetnostjo. Pri normalni porazdelitvi se izkaže, da je v območju $\bar{x} \pm 1\sigma$ kar 68% vseh meritev, pri $\bar{x} \pm 2\sigma$ jih je 95% in pri $\bar{x} \pm 3\sigma$ celo 99,7%. Tako območje $\pm a\sigma$ imenujemo območje zaupanja. Najpogosteje se v praksi uporablja območje zaupanja s koef. $a = 1,96$, v katerem bomo zanesljivo našli 95,00% meritev.

4 Arduino UNO

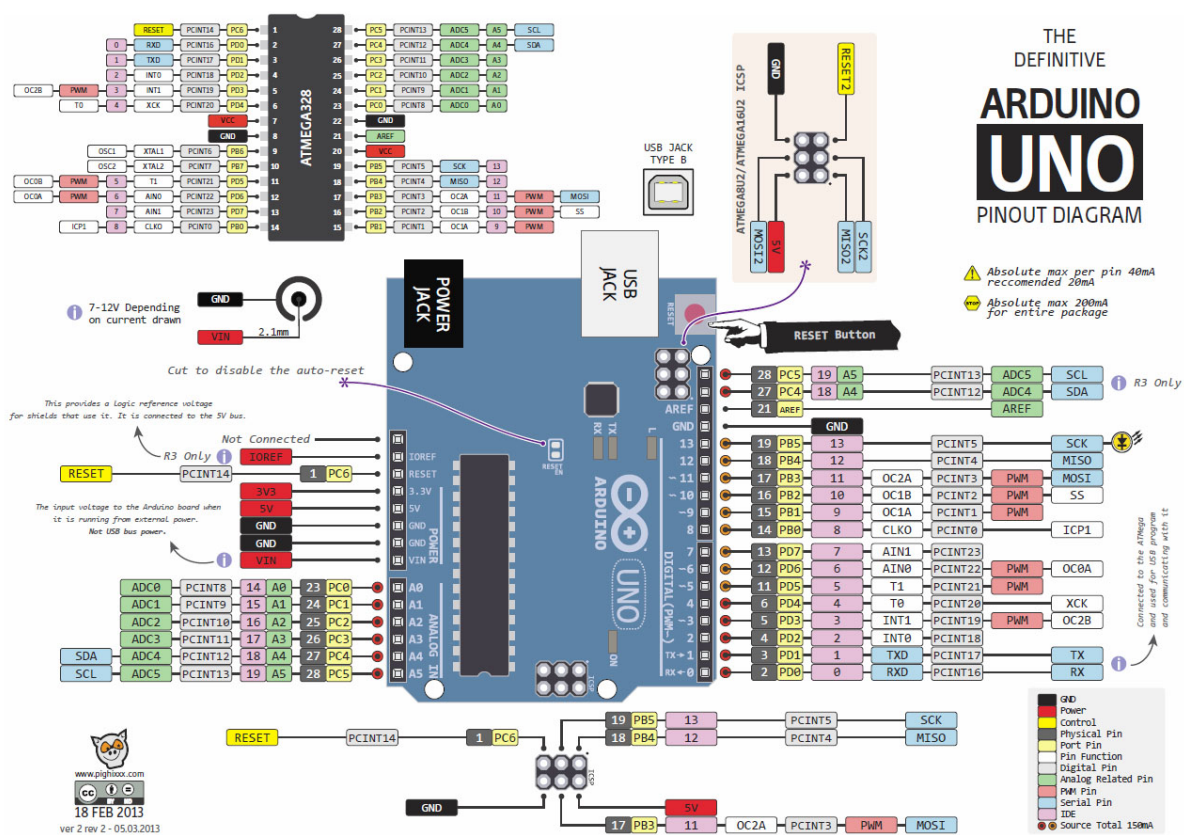
Arduino Uno je:

The UNO is the best board to get started with electronics and coding. If this is your first experience tinkering with the platform, the UNO is the most robust board you can start playing with. The UNO is the most used and documented board of the whole Arduino family. („Arduino Uno Rev3“, b. d.)

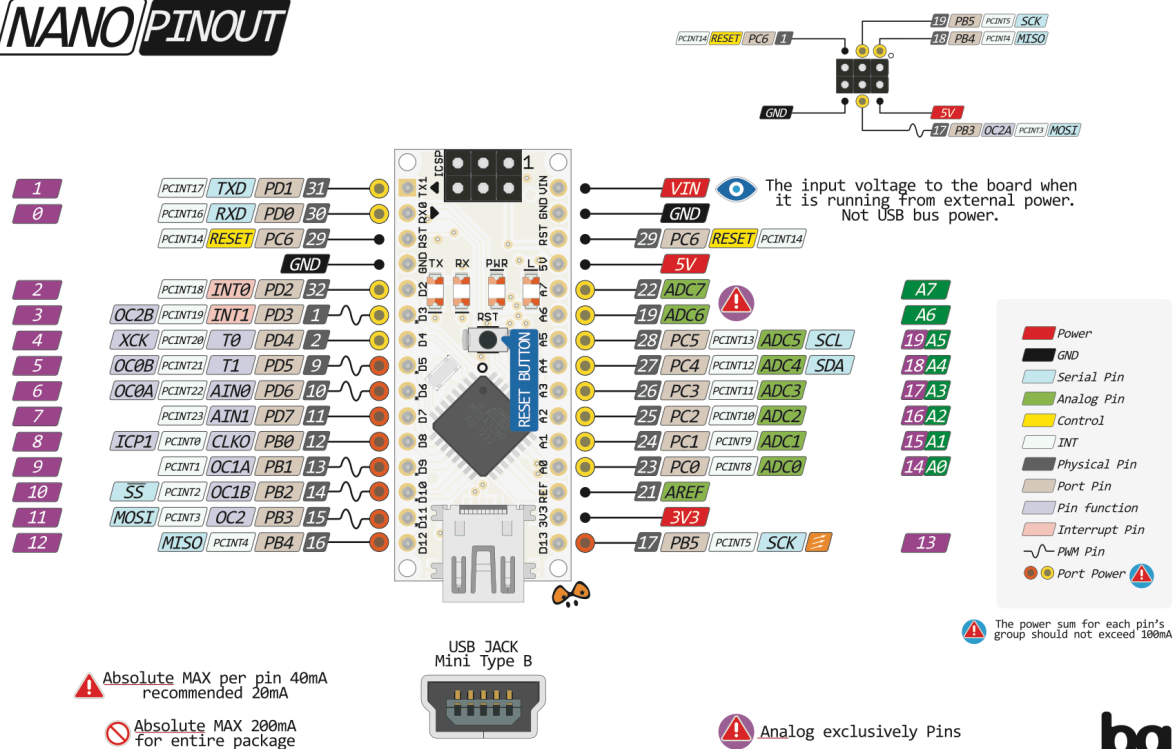


Slika 7: Arduino Uno

4.1 Arduino UNO/NANO pinout



NANO PINOUT



Slika 9: Razporeditev priključkov na krmilniku Arduino NANO (Theman 2015).

Kot lahko opazimo, so funkcije priključkov na sliki 8 in sliki 9 enaki. To ni nenavadno, saj sta krmilnika po električni zgradbi enaka, razlikujeta se le v fizični izvedbi.

4.2 Osnove programiranja krmilnika

Naloga 4: Preskus strojne in programske opreme.

Pravilno priključite krmilnik Arduino na računalnik in nastavite programsko okolje Aruino IDE ter sprogramirajte krmilnik s testnim programom **BLINK.INO**. Iz nastavitve programskega okolja prepišite vaše nastavitve za:

1. krmilnik: _____,
2. komunikacijska vrata: _____,
3. mikrokrmilnik: _____.

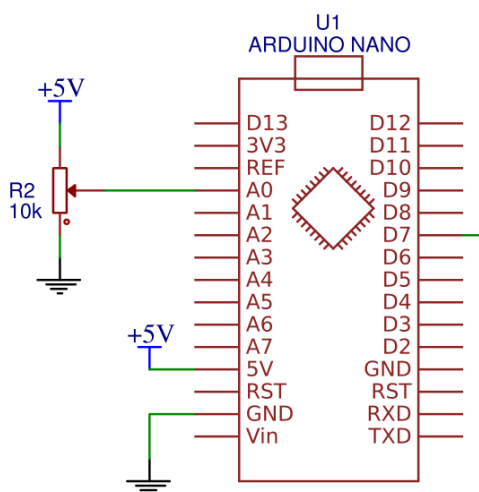
Testni program BLINK.INO

Nastavite in delovanje krmilnika lahko preverimo s testnim programom **blink.ino**. Na krmilniku Arduino UNO je na priključek **13** priključena svetleča dioda prav za ta namen.

```
1 void setup() {  
2   pinMode(LED_BUILTIN, OUTPUT);  
3 }  
4  
5 void loop() {  
6   digitalWrite(LED_BUILTIN, HIGH);  
7   delay(1000);  
8   digitalWrite(LED_BUILTIN, LOW);  
9   delay(1000);  
10 }
```

5 Merjenje s krmilnikom Arduino UNO

Krmilnik Atmega328 ima vgrajen ADC pretvornik, s katerim lahko odčitavamo analogne napetosti v območju $U_{ADC} = [0..5]V$. Te napetosti lahko odčitavamo na priključkih krmilnika, ki so označeni z **A0..A7**. Zato moramo senzorje priključiti tako kot prikazuje slika 10.



Slika 10: priključitev potenciometra na analogni priključek krmilnika.

Naloga 5: Merjenje napetosti.

Zvežite vezje po shemi na sliki 10 in sprogramirajte krmilnik tako, da boste na ekran računalnika izpisovali izmerjene vrednosti.

Napišite v kakšnem intervalu so bile izmerjene vrednosti:

ADC = [_____, _____].

5.1 Merjenje napetosti

```
1 int adc_value;
2 void setup() {
3     Serial.begin(9600);
4 }
5
6 void loop() {
7     adc_value = analogRead(0);
8     Serial.println(adc_value);
9     delay(100);
10 }
```

5.2 Izračun napetosti

Naloga 6: Izračun napetosti.

Napišite program za merjenje napetosti z ADC vmesnikom. V spodnji prostor pa vpišite le programske vrstice, ki ste jih uporabili za izračun napetosti.

5.3 Normalna porazdelitev meritev

Naloga 7: Normalna porazdelitev meritev in njeni parametri.

Z Arduino krmilnikom izmerite 100 meritev neke poljubne napetosti. Nato te meritve preverite še z volt-metrom, kar naj predstavlja vašo referenčno vrednost. Meritve vnesite v program za delo s tabelami in z ustreznimi funkcijami izračunajte, rezultat pa vpišite na črte:

- izmerjena referenčna vrednost: _____,
- povprečno vrednost meritev: _____,
- točnost predstavite z relativno napako: _____,
- preciznost meritev pa podajte s standardnim odklon: _____.

```
1 void setup() {
2   Serial.begin(9600);
3   int i = 0;
4   int adc_value = 0;
5   for (i=0;i<100;i++){
6     adc_value = analogRead(0);
7     float voltage = (float)adc_value * 5 / 1023;
8     Serial.println(voltage, 4);
9     delay(10);
10  }
11 }
12 void loop() {
13 }
```

5.4 Koeficienti normalne porazdelitve

Sploščenost

```
1 =KURT(Range)
```

Premaknjenost

```
1 =SKEW(Range)
```

Povprečna vrednost

```
1 =AVERAGE(Range)
```

Standardna napaka

```
1 =STEYX(y-Range,x-Range)
```

Interval zaupanja

```
1 =CONFIDENCE(Signif., Std.Dev., Sample Size)
```

5.5 Časovne meritve

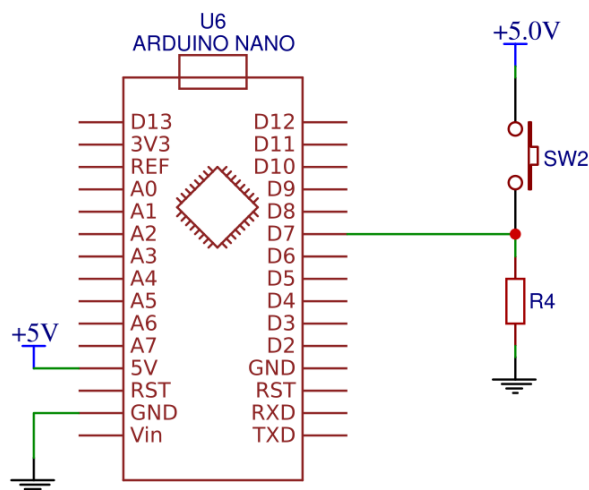
```
1 unsigned long time;
2
3 void setup() {
4   Serial.begin(9600);
5 }
6 void loop() {
7   Serial.print("Time: ");
8   time = millis();
9
10  Serial.println(time); //prints time since program started
11  delay(1000);          // wait a second so as not to send massive
    amounts of data
12 }
```

Preverite tudi:

```
1 time = micros();
```

5.6 Uporaba digitalnih vhodov

V povezavi s časovnimi meritvami pogosto uporabljamo digitalne vhode. Le-te lahko najdemo na priključkih **D0..D13** in tudi na **A0..A7**. Primer enostavne vezave tipke na krmilnik prikazuje slika 11.



Slika 11: Vezava tipke na digitalni vhod krmilnika Arduino.

Naloga 8: Časovni odziv človeške reakcije.

Napišite program za merjenje hitrosti človeškega odziva in zvežite vezje, ki ga prikazuje slika 11. Po naključnem času naj zasveti lučka na krmilniku ArduinoUNO. Nato pa naj program nemudoma shrani trenutni čas v *start_time* in ko uporabnik pritisne tipko naj si program ponovno shrani čas v *stop_time*. Program naj izračuna razliko časov in ga prikaže v na računalniku. Napredno: Če znate program popravite tako, da ne bo omogočal goljufanja.

```
1 void setup() {
2   pinMode(13,OUTPUT);
3   pinMode(7,INPUT);
4   Serial.begin(9600);
5   Serial.println("Start...");
6   randomSeed(analogRead(0));
7 }
8
9 void loop() {
10  digitalWrite(13,LOW);
11  delay(random(5000,10000));
12  digitalWrite(13,HIGH);
13
14  unsigned long start_time = micros();
15  unsigned long stop_time = 0;
16
17  while (digitalRead(7)==0){
18    stop_time = micros();
19  }
20
21  unsigned long time_div = stop_time - start_time;
22  Serial.println(time_div);
23
24  while (digitalRead(7)==1){
25    delay(200);
26    Serial.println("spusti tipko...");
27  }
28 }
```

5.7 Hitrost

Naloga 9: Merjenje hitrosti s svetlobnimi vrati.

Uporabite ali sestavite dvojna svetlobna vrata. Nato napišite program, ki bo izmeril razliko v času ko se svetlobni snop na enih in drugih svetlobnih vratih prekine. Izmerite razdaljo med vratoma in izračunajte hitrost.

Napredno: Če zante program popravite tako, da bo občutljiv na spremembo vhodnega signala.

```
1 void setup() {
2   pinMode(7,INPUT);
3   pinMode(8,INPUT);
4   Serial.begin(9600);
5   Serial.println("Start...");
6 }
7
8 void loop() {
9   unsigned long start_time = 0;
10  unsigned long stop_time = 0;
11
12  while (digitalRead(7)==0){
13    start_time = micros();
14  }
15
16  while (digitalRead(8)==0){
17    stop_time = micros();
18  }
19
20  unsigned long time_div = stop_time - start_time;
21  Serial.println(time_div);
22
23  Serial.println("Nova maritev...");
24 }
```

5.8 Pospešek

6 Prikazovanje podatkov

Prikazovanje podatkov je ena od ključnih nalog vseh merilnih sistemov. To lahko naredimo na različne načine:

1. v obliki tabele na zaslon računalnika,
2. v grafični obliki,
3. na zaslon samostojne merilne naprave ...

Naloga 10: Izpis podatkov.

Prikažite podatke na različne načine:

1. v obliki tabele,
2. v grafični obliki in
3. na LCD zaslon merilne naprave.

6.1 Serijski izpis

Spodnji primer programske kode prikazuje kako lahko z Arduino krmilniki izpisujemo osnovne podatke neke meritve. Izpisovali bomo dva podatka v ločena stolpca. Stolpca bosta ločena s tabulatorjem, kar predstavlja označba `\t`. V prvem stolpcu bomo izpisovali trenutni čas (`millis()`), v drugega pa vrednost analogne pretvorbe (`analogRead(0)`).

```
1 void setup() {  
2   Serial.begin(9600);  
3   Serial.println("t[ms]\tADC");  
4 }  
5  
6 void loop() {  
7   Serial.print(millis());  
8   Serial.print("\t");  
9   Serial.println(analogRead(0));  
10  delay(100);  
11 }
```

6.2 Grafični izris

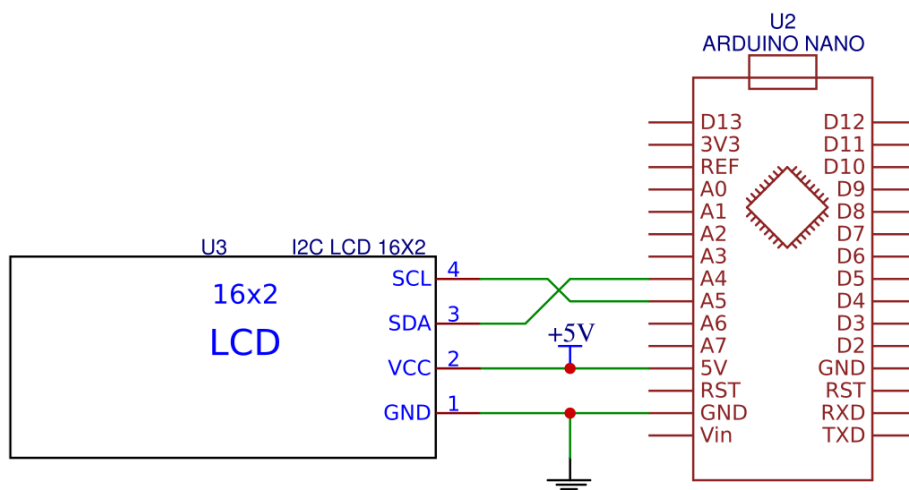
Zelo nazorno je spremljati izris podatkov v grafičnem načinu. Do neke mere nam to funkcijo ponuja programsko okolje Arduino IDE, vendar v zelo okenjeni obliki. Preskusite naslednji program:

```
1 void setup() {  
2   Serial.begin(9600);  
3 }  
4 void loop() {  
5   Serial.print("0,1023,");  
6   Serial.println(analogRead(A0));  
7   delay(10);  
8 }
```

Kot lahko opazimo računalniku pošiljamo 3 različne števila: 0, 1023 in ADC vrednost. Prvi dve nam slušita kot območje, saj bi v nasprotnem primeru program sam prilagajal skalo na y osi (ang.: autofit).

6.3 Izpis na LCD

Če želimo narediti merilno napravo, ki bo delovala samostojno, bomo verjetno potrebovali dodaten ekran za prikaz izmerjenih vrednosti. Zato ga moramo priključiti na krmilnik Arduino, kot kaže slika 12.



Slika 12: Vezava I²C LCD-ja na krmilnik Arduino.

Ena naj enostavnejših rešitev je, da krmilniku Arduino UNO dodamo LCD ekranček z I²C vodilom. Ker

je I²C vodilo zasnovano tako, da nanj lahko priključimo več naprav, moramo za vsako napravo poznati njen naslov. Le tega lahko pridobimo od proizvajalca naprave, ga nastavimo sami ali pa z ustreznim programom pregledamo vse naslove priključenih naprav (programska koda (Arduino 2019)). Za uporabo LCDja na podatkovnem vodilu I²C bomo potrebovali knjižnico **LiquidCrystal_I2C**, ki si jo lahko presnamete s spletne strani Francisco Malpartida.

```
1 #include <Wire.h>
2 #include <LiquidCrystal_I2C.h>
3 LiquidCrystal_I2C DaqLcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
4
5 void setup() {
6     DaqLcd.begin(8, 2);
7 }
8
9 void loop() {
10     int i = 0;
11     long ch = 0;
12     for (i=0; i<64; i++){
13         ch += analogRead(7);
14     }
15
16     float val = ch/64.0;
17     DaqLcd.clear();
18     DaqLcd.print(val);
19 }
```

7 Linearizacija meritvenih vrednosti (praznjenje kondenzatorja)

Izvedli bomo poskus, pri katerem bomo lahko meritve odvisne spremenljivke (napetost) opravili pri različnih vrednostih neodvisne spremenljivke (čas). Iz rezultatov bomo skušali izluščiti kapaciteto kondenzatorja (kot iskano konstantno sistema). Ker je fizikalni pojav nelinearen, ga bomo najprej linearizirali, in šele nato izračunali iskano konstanto (kapaciteto kondenzatorja). Tako lahko v izračun enostavno vključimo vse meritve...

Naloga 11: Izračun vrednosti s pomočjo regresijske premice.

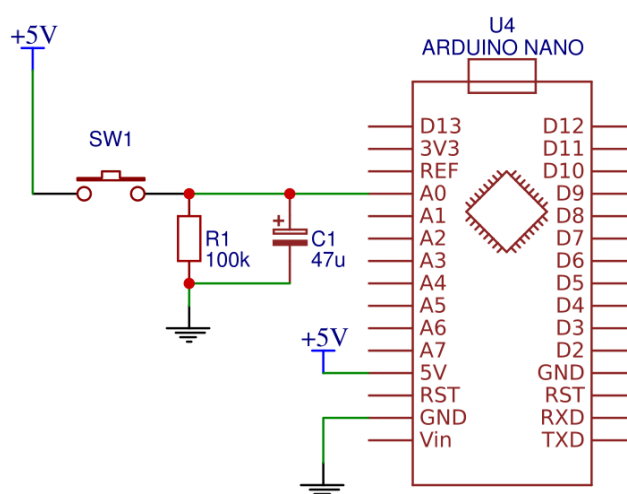
Izvedite poizkus praznjenja kondenzatorja in izmerite $U_C(t)$. Podatke linearizirajte tako, da jim boste lahko dodelili regresijsko premico.

Izračunajte:

- regresijski koeficient: R^2 : _____,
- enačba smernega koeficienta linearizacije:
- vrednost smernega koeficienta:
- vrednost kapacitete kondenzatorja:

7.1 Vezje in zajem podatkov

Vezje sestavimo po sliki 13:



Slika 13: Vezava vezja za zajem napetosti na kondenzatorju.

7.2 Linearizacija

$$f(x) = kx + n \quad (8)$$

$$U_C(t) = U_0 e^{-\frac{t}{RC}} \quad (9)$$

$$\ln U_0 = -\frac{1}{RC}t + \ln U_C \quad (10)$$

ali ...

$$\ln\left(\frac{U_C}{U_0}\right) = -\frac{1}{RC}t \quad (11)$$

7.3 Koeficienti regresijske premice

```
1 =SLOPE(y-Range,x-Range)
```

```
1 =INTERCEPT(y-Range,x-Range)
```

```
1 Korelacija med x in y.  
2 =CORREL(y-Range,x-Range)
```

```
1 Korelacija med y_0 in y.  
2 =RSQ(y-Range,x-Range)
```

8 Senzorji

- toleranca uporov

8.1 Občutljivost

Ob predpostavki, da je določena meritev linearna na celotnem merilnem področju merilnega instrumenta, lahko izrazimo občutljivost instrumenta kot kvocient med spremembo izmerjene in spremembo merjene spremenljivke. Občutljivost analognega instrumenta določa razmerje med linearnim pomikom indikatorja lege in spremembo merjene spremenljivke, ki omenjeni pomik povzroči.

8.2 Delilniki napetosti s spremenljivim uporom

Pri takih vrstah senzorjev je izrednega pomena, da izberemo primeren referenčni upor R_{Ref} , saj le-ta vpliva na občutljivost senzorja. Izhodno napetost za temperaturni senzor s termistorjem in referenčnim uporom lahko izračunamo po enačbi 12.

$$U_{izh} = \frac{U_0 R_{Ref}}{R_{Ref} + R_{NTC}} \quad (12)$$

Če bomo ta temperaturni senzor uporabljali v temperaturnem intervalu $T \in [T_{min}, T_{max}]$, se bo njegova upornost spreminjala v intervalu $R_{NTC} \in [R_{NTC_{max}}, R_{NTC_{min}}]$. Ker želimo, da bi senzor imel v tem območju kar največjo občutljivost lahko sestavimo enačbo 13, za spremembo izhodne napetosti pri intervalu T .

$$\Delta U_{izh} = U_{izh}(T_{max}) - U_{izh}(T_{min}) \quad (13)$$

Če poiščemo maksimum funkcije enačbe 13 za R_{Ref} se izkaže, da je največja občutljivost senzorja pri referenčnem upor, ki ga podaja enačba 14:

$$R_{Ref} = \sqrt{R_{NTC_{min}} R_{NTC_{max}}} \quad (14)$$

Delilnik napetosti z notranjim uporom

Delilnik napetosti lahko naredimo tudi z uporom, ki ga mikrokrmilnik že vsebuje. Ker je ta upor vezan proti napajanju ga imenujemo **pull-up resistor**. V našem primeru ima ta upor upornost $R_{PULL-UP} \approx 10k\Omega$. Ker njegova vrednost ni točna je manj primeren za uporabo pri analognih senzorjih, vendar za izdelavo digitalnih lahko nadomesti nepotrebne vezave.

9 Umerjanje (temperaturnega) senzorja

- glej poglavje o delilnikih
- naredite temp. senzor, katerega odziv bo največji za temperaturno območje $T_{NTC} = [15, 95]^{\circ}C$.

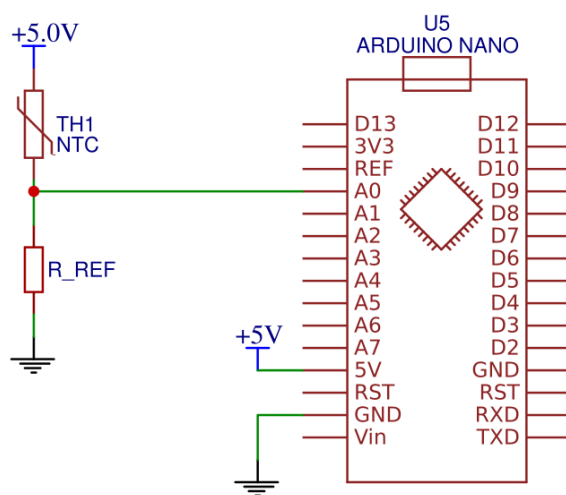
Naloga 12: Dimenzioniranje senzorja

Podrobno si pogledajte pogl. 8 in dimenzionirajte temperaturni senzor z NTC termistorjem za območje $T = [15, 95]^{\circ}C$. Izračunajte ali izmerite ključne vrednosti za dimenzioniranje senzorja:

$$R_{NTC, T=15^{\circ}C} = \underline{\hspace{2cm}},$$

$$R_{NTC, T=95^{\circ}C} = \underline{\hspace{2cm}},$$

$$R_{REF} = \underline{\hspace{2cm}}.$$



Slika 14: Vezava temperaturnega senzorja.

9.1 Umeritveni postopek

Naloga 13: Umerjanje temperaturnega senzorja.

Naredite umeritveni postopek in predstavite tabelo s podatki.

T[°C] | U[V]

```
1 String inputString = "";           // a String to hold incoming data
2 boolean stringComplete = false;    // whether the string is complete
3
4 void setup() {
5   Serial.begin(9600);
6   Serial.println("#####");
7   Serial.println("# INSTRUCTIONS:");
8   Serial.println("# 1. Insert temperature (example: 22.4) #");
9   Serial.println("# 2. then press ENTER");
10  Serial.println("# 3. repeat steps 1. and 2. ...");
11  Serial.println("# When you are done with calibration:");
12  Serial.println("# 4. Copy data to spreadsheet.");
13  Serial.println("# 5. Add some polynome to the points.");
14  Serial.println("# 6. Insert coefficients into file:");
15  Serial.println("# 'temperature.ino'");
16  Serial.println("#####");
17  Serial.println("T[°C] Value");
18  inputString.reserve(20);
19 }
20
21 void loop() {
22   if (stringComplete) {
23     Serial.print(inputString);
```

```
24     Serial.print(" ");
25     inputString = "";
26     stringComplete = false;
27
28     long CH1 = 0;
29
30     int i;
31     for (i = 0; i < 128; i++) {
32         CH1 += analogRead(A7);
33         delay(1);
34     }
35     float Vol_CH1 = float(CH1/128.0);           // calibration in ADC
36         values...
37     //float Vol_CH1 = CH1*5.0/1023; // calibration in Voltage...
38     Serial.println(Vol_CH1);
39 }
40
41 void serialEvent() {
42     while (Serial.available()) {
43         char inChar = (char)Serial.read(); // get the new byte:
44         if (inChar == '\n') {
45             stringComplete = true;
46         } else {
47             inputString += inChar;           // add it to the inputString:
48         }
49     }
50 }
```

9.2 Interpolacija

Naloga 14: Interpolacija

Dobljene podatke vnesite v programsko orodje za delo s tabelami in podatke izrišite v grafični obliki. Nato dodajte ustrezen trend in izpišite enačbo prenosne funkcije.

Prenosna funkcija:

9.3 Izpis temperature (stand-alone DAQ)

Naloga 15: Izpis temperature

V programskem orodju za tabele (Microsoft Excel ali LibreOffice Calc) podatkom priredimo ustrezno krivuljo in odčitamo koeficiente... Nato napišemo program:

```
1 //          k0          k1          k2          k3          k4          k5
2 float k[6] = { -74.91, 530.963E-3, -1.685E-3, 3.253E-6, -3.128E-9,
3               1.227E-12};
4
5 void setup() {
6   pinMode(15, OUTPUT);
7   digitalWrite(15, HIGH);
8   pinMode(16, OUTPUT);
9   digitalWrite(16, LOW);
10  pinMode(14, INPUT); //A0
11  // initialize the serial communication:
12  Serial.begin(9600);
13 }
14
15 void loop() {
16   // send the value of analog input 0:
17   int i;
18   float CH1 = 0;
19   for (i = 0; i < 64; i++) {
20     CH1 += analogRead(A7);
21     delay(3);
22   }
23   float Vol_CH1 = float(CH1 / 64.0);
24
25   float Temp_CH1 = 0;
26   for (int i = 0; i <= 5; i++) {
27     Temp_CH1 += k[i] * pow(Vol_CH1, i);
28   }
29   Serial.println(Temp_CH1);
30   //delay(1000);
31 }
```

10 Projektna naloga

Naloga 16: Samostojna postavitve fiz. eksp. s predstavitevijo rezultatov.

Sestavite poljuben fizikalni eksperiment, ga opremite s primernimi senzorji in podatke zajemite s krmilnikom Arduino. V primernim programskem okolju podatke obdelajte in jih številsko in grafično prikažite.

Dokumentirajte:

1. kratek opis eksperimenta,
2. vezje,
3. postavitev eksperimenta,
4. programsko kodo,
5. meritve,
6. obdelavo podatkov in
7. predstavitev rezultatov.

11 Primeri

11.1 Merjenje srčnega utripa

```
1  #include <Wire.h>
2  #include <LiquidCrystal_I2C.h>
3  LiquidCrystal_I2C DaqLcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
4
5  int sensorPin = 7;
6  double alpha = 0.75;
7  int period =100;
8  double change =0.0;
9  double minval = 0.0;
10
11 void setup() {
12     DaqLcd.begin(8, 2);
13     Serial.begin(9600);
14 }
15
16 void loop() {
17     int i = 0;
18     long ch = 0;
19     for (i=0;i<64;i++){
20         ch += analogRead(7);
21     }
22
23     float val = ch/64.0;
24     Serial.println(val);
25 }
```

Kazalo Nalog:

1	Frekvenca vzorčenja.	13
2	Resolucija AD pretvornika.	14
3	Ločljivost AD pretvornika.	14
4	Preskus strojne in programske opreme.	19
5	Merjenje napetosti.	21
6	Izračun napetosti.	22
7	Normalna porazdelitev meritev in njeni parametri.	22
8	Časovni odziv človeške reakcije.	25
9	Merjenje hitrosti s svetlobnimi vrati.	26
10	Izpis podatkov.	27
11	Izračun vrednosti s pomočjo regresijske premice.	30
12	Dimenzioniranje senzorja	33
13	Umerjanje temperaturnega senzorja.	34
14	Interpolacija	35
15	Izpis temperature	36
16	Samostojna postavitve fiz. eksp. s predstavitvijo rezultatov.	37

11.2 Točkovanje

Vse naloge od 1 .. 15 se točkujejo s $t_{1..15} = [0, 1, 2]$ točke.

Naloga 16 pa se točkuje s $t_{16} = [0, 5, 10]$ točkami za vsako od postavk:

Naloga: **16.1** (kratak opis eksperimenta) =

Naloga: **16.2** (vezje) =

Naloga: **16.3** (postavitev eksperimenta) =

Naloga: **16.4** (programsko kodo) =

Naloga: **16.5** (meritve) =

Naloga: **16.6** (obdelavo podatkov in) =

Naloga: **16.7** (predstavitve rezultatov) =

11.3 Točke:

$t_{1..15} = \dots\dots\dots /30$

$t_{16} = \dots\dots\dots /70$

Skupaj:

$t_{RPF} = \dots\dots\dots /100$

Viri in literatura

Arduino. 2019. „Arduino Playground - I2cScanner“. 2019. <https://playground.arduino.cc/Main/I2cScanner/>.

„Arduino Uno Rev3“. b. d. <https://store.arduino.cc/usa/arduino-uno-rev3>.

Atmel. 2017. „ATMEGA328P“. 2017. http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf.

Contributors, Wikipedia. 2019a. „Analog-to-digital converter“. 2019. https://en.wikipedia.org/wiki/Analog-to-digital_converter.

———. 2019b. „Normal distribution“. 2019. https://en.wikipedia.org/wiki/Normal_distribution.

———. 2019c. „Točnost in natančnost“. 2019. https://sl.wikipedia.org/wiki/To%C4%8Dnost_in_natan%C4%8Dnost.

„LabQuest® Hardware and Specifications | Vernier“. b. d. <https://www.vernier.com/products/interfaces/labq/hardware/>.

Nyquist, H. 1928. „Certain Topics in Telegraph Transmission Theory“. *Transactions of the American Institute of Electrical Engineers* 47 (2): 617–44. <https://doi.org/10.1109/t-aiee.1928.5055024>.

Theman, Dan. 2015. „Can't Get I2C to Work on an Arduino Nano? (Pinout Diagrams)“. 2015. <https://bigdanzblog.wordpress.com/2015/01/30/cant-get-i2c-to-work-on-an-arduino-nano-pinout-diagrams>.

„What Is Data Acquisition? - National Instruments“. b. d. <http://www.ni.com/data-acquisition/what-is/>.