Računalnik v pouku Fizike

Cenovno dostopen merilni sistem z Aruino UNO krmilnikom

DR. DAVID RIHTARŠIČ

2019-04-15 09:28

Kazalo

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 –> beater 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]]

Slika 1: Parts of DAQ

1.2.1 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 [@eq:Ohmov] in na sliki [@fig:ArduinoUNO.jpg].

[@WhatIsDa17:online]

1.2.2 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].

1.2.3 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 [[NationalInstruments]].

1.3 Teoretične osnove

1.3.1 Digitalizacija

1.3.2 Frek. vzorčenja

1.3.3 Resolucija vs ločljivost

1.3.4 Natančnost vs točnost

1.3.5 Napake (naključna, sistemska)

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 @LabQuest30:online , ki nam omogoča raznovrstne meritve.

2.1 Vernier's DAQs

Slika 2: Vernier-jeva merilna naprava LabQuest2

Poglejmo si nekaj podatkov o napravi:

- **Display**
    - 11.2 cm x 6.7 cm (13.1 cm diagonal) screen
    - 00 x 480 pixel color display at 188 dpi
    - ED backlight
    - ortrait or landscape screen orientation
    - igh-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 expencive (from 100 € .. n k€). Buy we can use Arduino (Uno, nano, …) as low-cost data acqusition system if we do not need scientific accurate data (for pedagogical purposes).

[[NationalInstru

2.2.1 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

2.2.2 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 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. @ArduinoU75:online

Slika 3: Arduino Uno

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

Poglejmo si nekaj karakteristik tega mikrokrmilnika [@ATmega328P_Datasheet:online]:

- **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 - 260µs 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)

3.1 Arduino UNO/NANO pinout

Slika 4: Razporeditev priključkov na krmilniku Arduino UNO [@ArduinoUNOpinout:online].

Slika 5: Razporeditev priključkov na krmilniku Arduino NANO [@ArduinoUNOpinout:online].

Kot lahko opazimo, so funkcije priključkov na sliki [@fig:Arduino-UNO-pinout.jpeg] in sliki [@fig:nano.png] enaki. To ni nenavadno, saj sta krmilnika po električni zgradbi enaka, razlikujeta se le v fizični izvedbi.

3.2 Osnove programiranja krmilnika

- Arduino –> Blink

4 Merjenje s krmilnikom Arduino UNO

4.1 Časovne meritve

4.1.1 Hitrost

4.1.2 Pospešek

4.2 Merjenje napetosti

4.2.1 Izračun napetosti

> Vaja: Študentje napišejo program za merjenje napetosti z ADC vmesnikom.

```
void setup() {
  pinMode(15, OUTPUT);
  digitalWrite(15,LOW);
  pinMode(14, INPUT_PULLUP); //A0
  // initialize the serial communication:
  Serial.begin(9600);
}

void loop() {
  // send the value of analog input 0:
  int CH1 = analogRead(A0);

  Serial.print("0,1023,");
  Serial.println(CH1);

  // wait a bit for the analog-to-digital converter to stabilize after the last
  // reading:
  delay(10);
}
```

4.3 Prikazovanje podatkov

4.3.1 Serijski izpis

4.3.2 Grafični izris

4.3.3 Izpis na LCD

5 Statistična obdelava meritev (praznenje kondenzatorja)

5.1 Vezje in zajem podatkov

5.2 Linearizacija

5.3 Povprečna vr.

5.4 Smerni koef.

5.5 Standardna napaka

6 Umerjanje (temperaturnega) senzorja

6.1 Umerjanje

> Vaja: Naredimo umeritven postopek.

```
String inputString = "";         // a String to hold incoming data
boolean stringComplete = false;  // whether the string is complete

void setup() {
  int pin_gnd=16;
  int pin_vcc=15;
  pinMode(pin_gnd, OUTPUT);
  pinMode(pin_vcc, OUTPUT);
  digitalWrite(pin_gnd, LOW);
  digitalWrite(pin_vcc, HIGH);
  // initialize the serial communication:
  Serial.begin(9600);
  Serial.println("###########################################");
  Serial.println("# INSTRUCTIONS:                           #");
  Serial.println("#   1. Insert temperature (example: 22.4) #");
  Serial.println("#   2. then press ENTER                   #");
  Serial.println("#   3. repeat steps 1. and 2. ...         #");
  Serial.println("# When you are done with calibration:     #");
  Serial.println("#   4. Copy data to spreadsheet.          #");
  Serial.println("#   5. Add some polynome to the points.   #");
  Serial.println("#   6. Insert coeficients into file:      #");
  Serial.println("#       'temperature.ino'                 #");
  Serial.println("###########################################");
  Serial.println("T[°C] Value");
  inputString.reserve(20);
}

void loop() {

  if (stringComplete) {
    Serial.print(inputString);
    Serial.print(" ");
    inputString = "";
    stringComplete = false;

    // send the value of analog input 0:
    long CH1 = 0;

    int i;
    for (i = 0; i < 128; i++) {
      CH1 += analogRead(A7);
      delay(1);
    }
    float Vol_CH1 = float(CH1/128.0);          // calibration in ADC values...
    //float Vol_CH1 = CH1*5.0/1023; // calibration in Voltage...
    Serial.println(Vol_CH1);
  }
}

void serialEvent() {
  while (Serial.available()) {
    char inChar = (char)Serial.read();  // get the new byte:
    if (inChar == '\n') {
      stringComplete = true;
    } else {
      inputString += inChar;                // add it to the inputString:
    }
  }
}
```

6.2 Interpolacija

in bi tako takoj naslednji študijski teden (10.5.) lahko vi že pričeli z izvedbo enega od fizikalnih eksperimentov…

6.3 Izpis temperature (stand-alone DAQ)

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

```
//            k0        k1        k2        k3        k4      k5
float k[6] = { -74.91, 530.963E-3, -1.685E-3, 3.253E-6, -3.128E-9, 1.227E-12};

void setup() {
  pinMode(15, OUTPUT);
  digitalWrite(15, HIGH);
  pinMode(16, OUTPUT);
  digitalWrite(16, LOW);
  pinMode(14, INPUT); //A0
  // initialize the serial communication:
  Serial.begin(9600);
}

void loop() {
  // send the value of analog input 0:
  int i;
  float CH1 = 0;
  for (i = 0; i < 64; i++) {
    CH1 += analogRead(A7);
    delay(3);
  }
  float Vol_CH1 = float(CH1 / 64.0);

  float Temp_CH1 = 0;
  for (int i = 0; i <= 5; i++) {
    Temp_CH1 += k[i] * pow(Vol_CH1, i);
  }
  Serial.println(Temp_CH1);
  //delay(1000);
}
```

7 Senzorji

- delilnik
- toleranca uporov

8 Primeri

8.1 Merjenje srčnega utripa

> Vaja: S študenti na pišemo program za merjenje srčnega utripa s pomočjo IR senzorja.

```
#include <Wire.h>¸
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C DaqLcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

int sensorPin = 7;
double alpha = 0.75;
int period =100;
double change =0.0;
double minval = 0.0;

void setup() {
  DaqLcd.begin(8, 2);
  Serial.begin(9600);
}

void loop() {
  int i = 0;
  long ch = 0;
  for (i=0;i<64;i++){
    ch += analogRead(7);
  }

  float val = ch/64.0;
  Serial.println(val);
}
```

9 Viri in literatura