Innovative Teaching Approaches in development of Software Designed Instrumentation and its application in real-time systems

Workshop for presenting book and new developed courses

Multiplier Event (E6)

Co-funded by the
Erasmus+ Programme
of the European Union

Project partners

- Faculty of Technical Sciences, University of Novi Sad
- Faculty of Electrical Engineering and Information Technologies, Ss. Cyril and Methodius University
- Zagreb University of Applied Sciences
- School of Electrical Engineering, University of Belgrade
- Faculty of Physics, Warsaw University of Technology



Project goal

- Innovative teaching materials,
- Practice Books with examples,
- Easy-access web platform for e-learning,
- Video lectures,
- New advanced courses,
- Improving existing courses,
- Seminars and workshops to promote LabVIEW language and prepared intellectual outputs,
- Regional student competitions that will be monitored by industrial representatives,
- Academies and industry collaboration in improving study programs.

Intellectual outputs covered by WUT

- Moodle platform for Virtual instrumentation Real-time Control systems courses,
- Video materials of lectures
- Modification of course "Principles of the Virtual Instruments Design",
- Course "Advanced LabVIEW Applications",
- "LabVIEW and Open Source Solutions" book.

Moodle platform

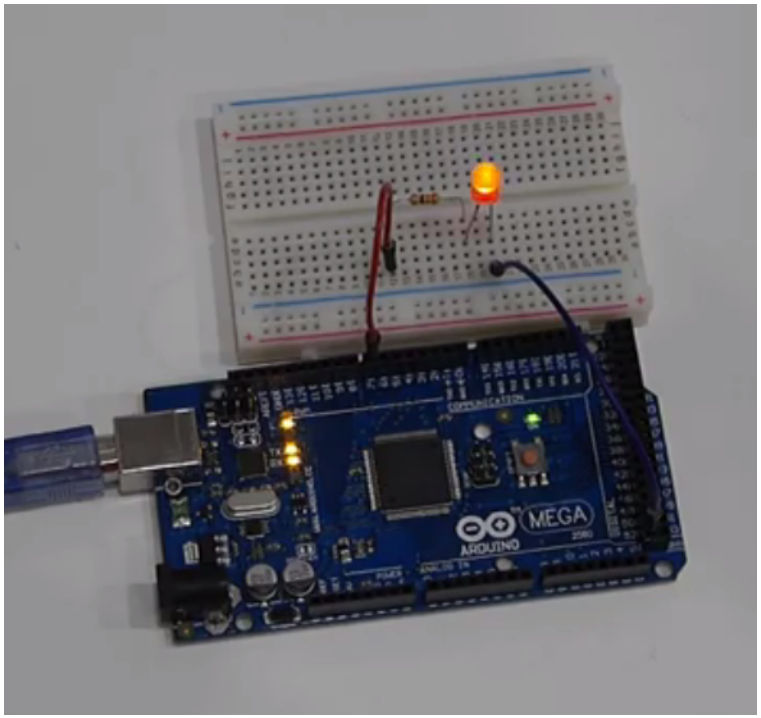
1. "Principles of the Virtual Instruments Design" course
   - Course covers the topic to CLAD exam (*Certified LabVIEW Associate Developer*) organized by National Instruments,
   - Course is divided in two sessions: theoretical description and test,
   - Course is in Polish language and has about 60 participants so far.
2. "Advanced LabVIEW Applications" course
   - First free, on-line advanced course for students,
   - Course prepares students to second exam: CLD (*Certified LabVIEW Developer*),
   - Course is divided in two sessions: theoretical description and test,
   - Course is in English language.

The moodle platform website:

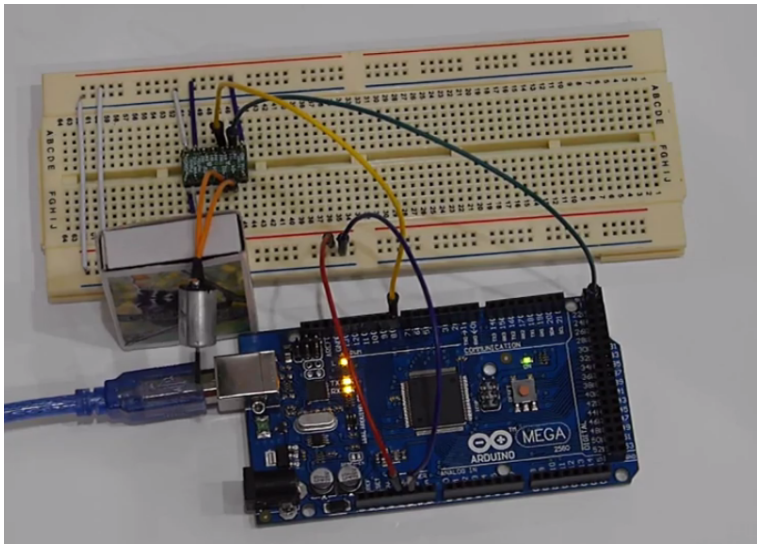
`http://itasdi.uns.ac.rs/`

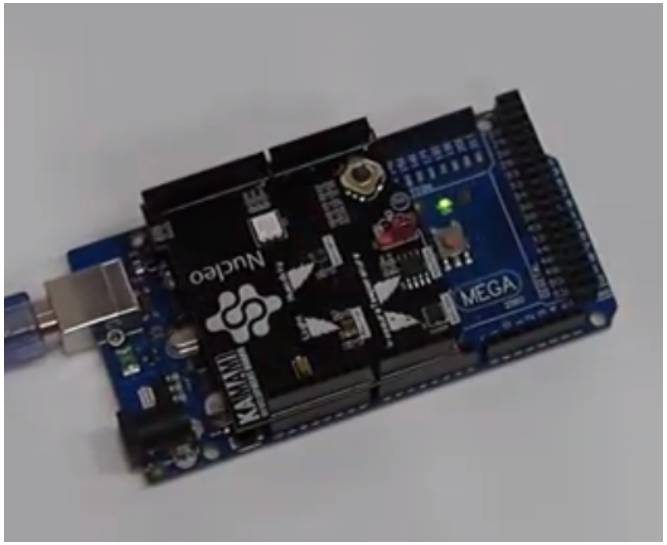
Video materials of lectures

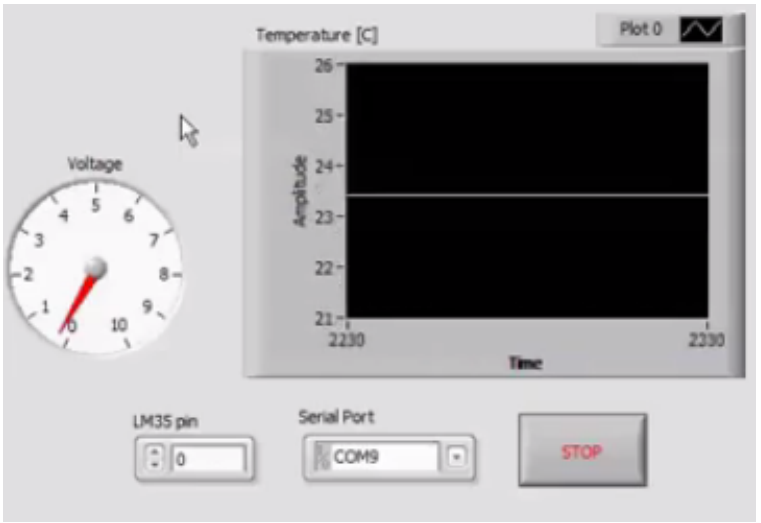
- Local variables in LabVIEW
- Global variables in LabVIEW
- Functional Global Variables in LabVIEW
- Master/Slave Design Pattern
- Producer/Consumer design pattern
- LabVIEW: Arduino platform

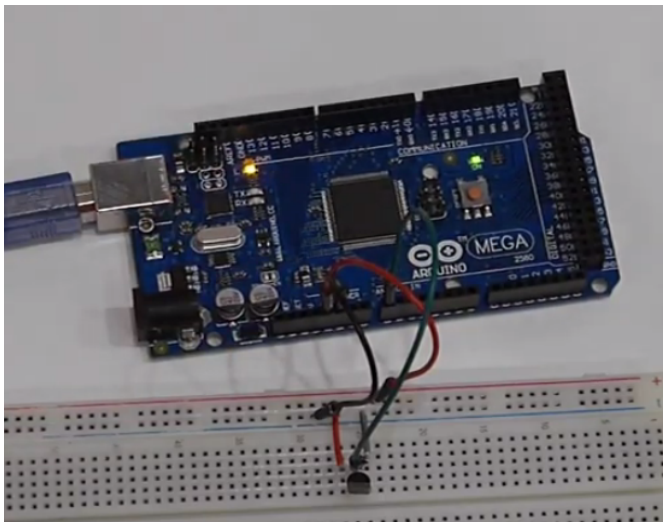


Video materials of lectures









Principles of the Virtual Instruments Design

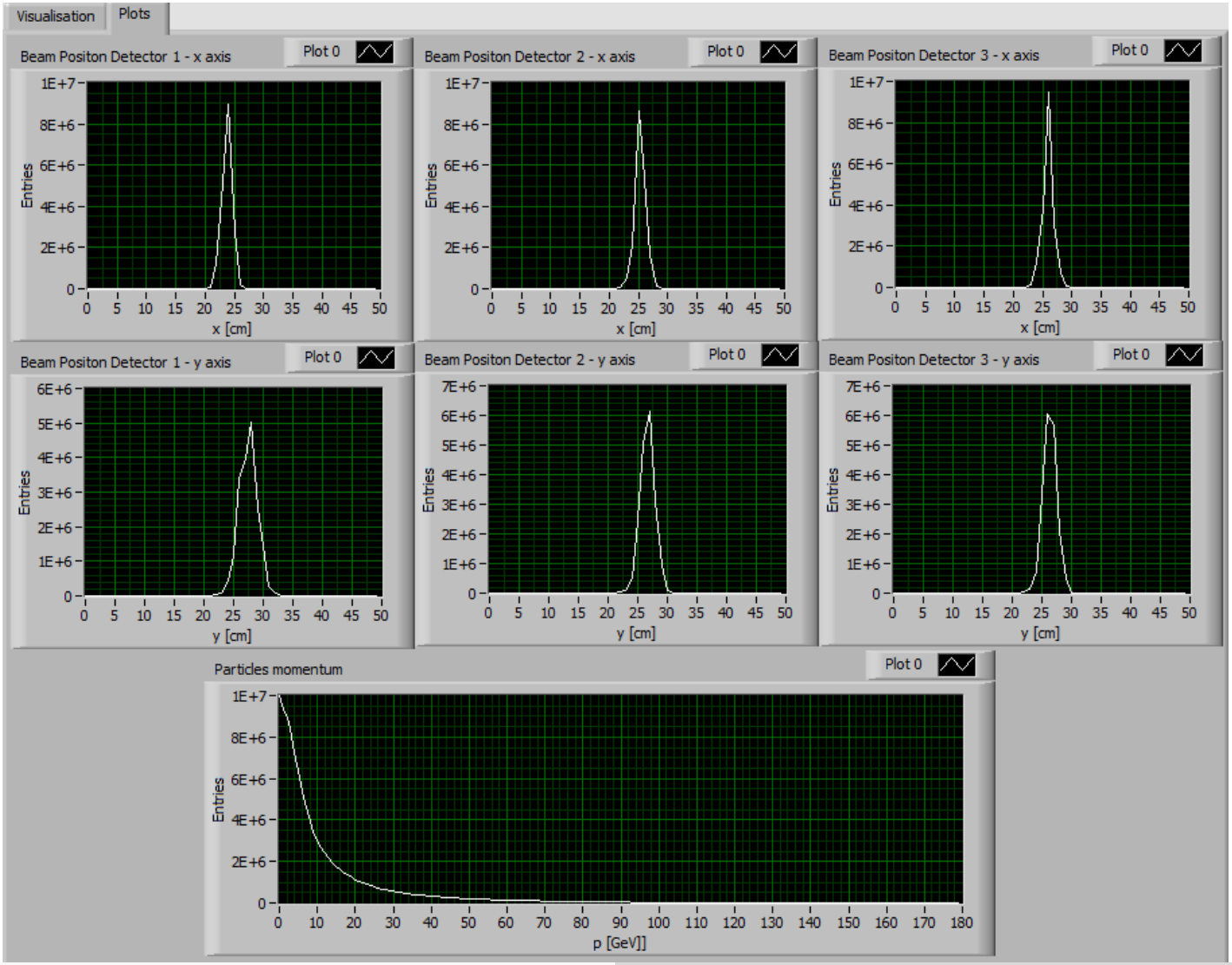
- Modified laboratory exercise: Lab8. Property Nodes, Zmienne współdzielone i funkcjonalne,
- New laboratory exercise: Lab13. Współpraca LabVIEW z Arduino,
- Modified lectures,
- Many, new practical examples for lectures,
- Materials prepared in cooperation with partners:
  1. Faculty of Technical Sciences, University of Novi Sad,
  2. Faculty of Electrical Engineering and Information Technologies, Ss. Cyril and Methodius University,
  3. School of Electrical Engineering, University of Belgrade.

Advanced LabVIEW Applications

- New practical course for students, which is continuation of Principles of the Virtual Instruments Design,
- New practical exercises:
  1. Laboratory no. 1 - Sequentional state machine
  2. Laboratory no. 2 - Master/Slave Design Pattern
  3. Laboratory no. 3 - Producer/Consumer Design Pattern
  4. Laboratory no. 4 - Producer/Consumer User Event Design Pattern
  5. Laboratory no. 5 - Queued Message Handle
  6. Laboratory no. 6-7 User Interface
  7. Laboratory no. 8 - Creating documentation, management and error logging
  8. Laboratory no. 9 - Timing
  9. Laboratory no. 10 - Student project
- New lectures and examples
- Materials prepared in cooperation with partners:
  1. School of Electrical Engineering, University of Belgrade.
  2. Zagreb University of Applied Sciences

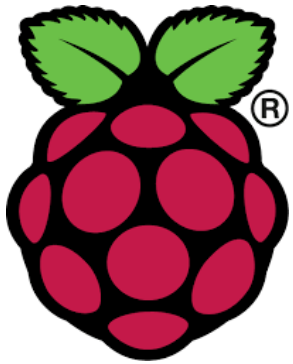
Example of student project

Example of student project



"LabVIEW and Open Source Solutions" book

- First book about connecting LabVIEW with open-source devices e.g. Arduino board, Raspberry Pi,
- The book consists of chapters:
  1. *Introduction*
  2. *Introduction to LabVIEW*
  3. *LabVIEW interfacing techniques*
  4. *Open source platforms*
  5. *LabVIEW Packages*
- More than 60 practical example codes are added to book as appendix,
- The book is available here.

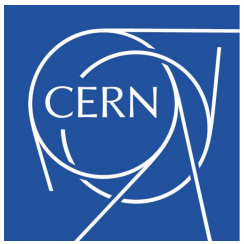


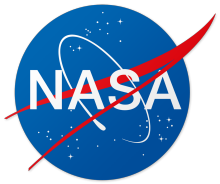
Introduction to LabVIEW

What is the LabVIEW language?

- **LabVIEW - Laboratory Virtual Instrument Engineering Workbench**,
- Graphical language designed to communicate with hardware by National Instruments (NI),
- LabVIEW allows to create graphical user interface easily,
- NI prepared many drivers for hardware, which are used in LabVIEW.

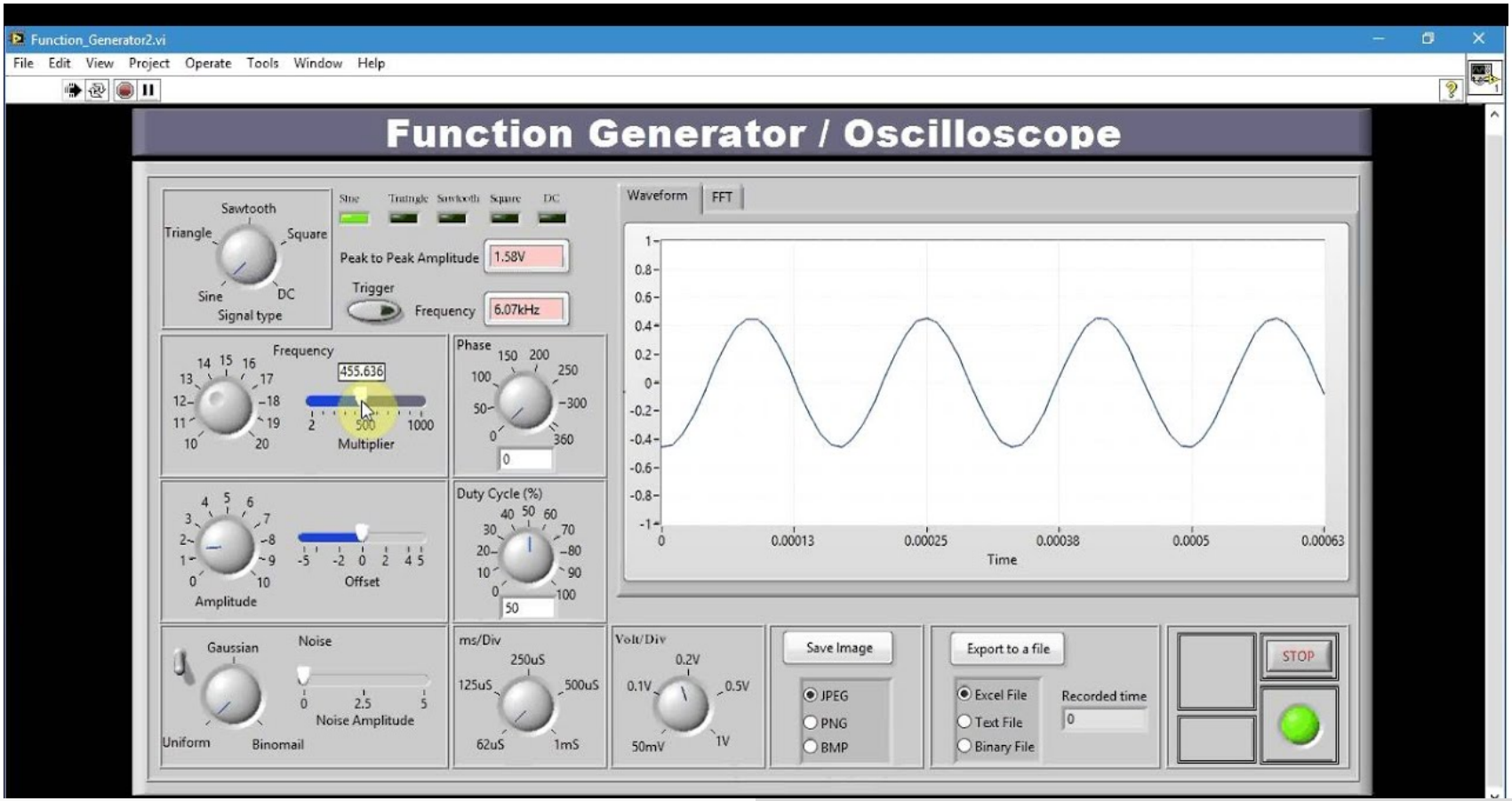
Who use LabVIEW?





Virtual instruments

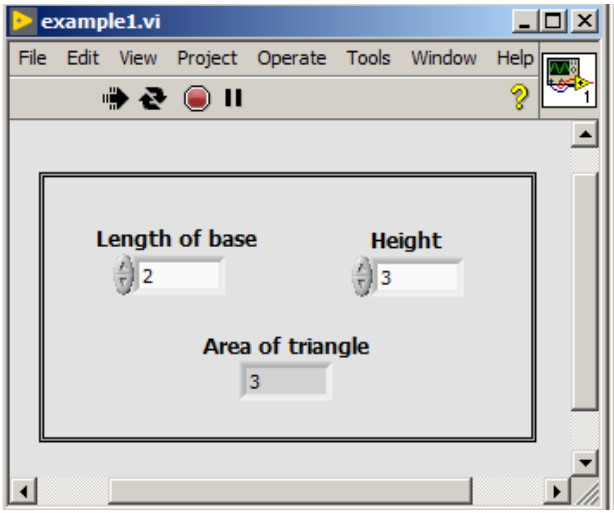
- The graphical user interface can look similar to real device so that they are called **virtual instruments** (**VI**).



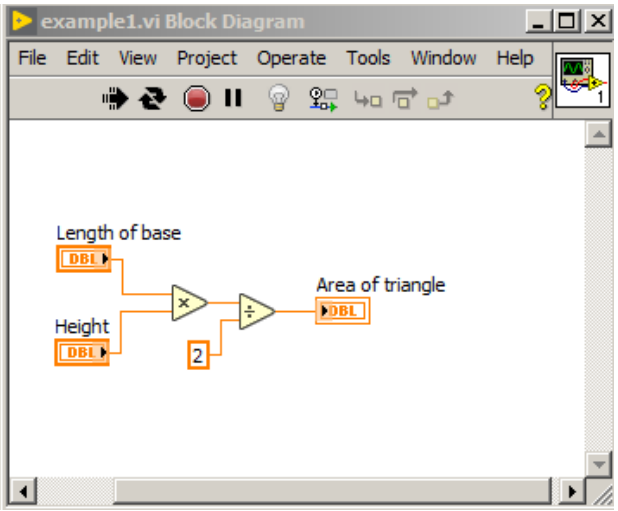
Parts of Virtual instruments

The VI consists of:

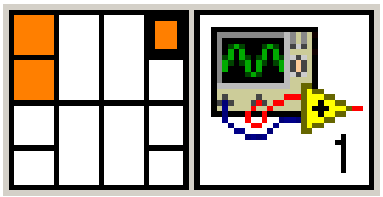
1. **Front panel** - graphical user interface,
2. **Block diagram** - source of program,
3. **Icon and connector pane**.



Front panel



Block diagram

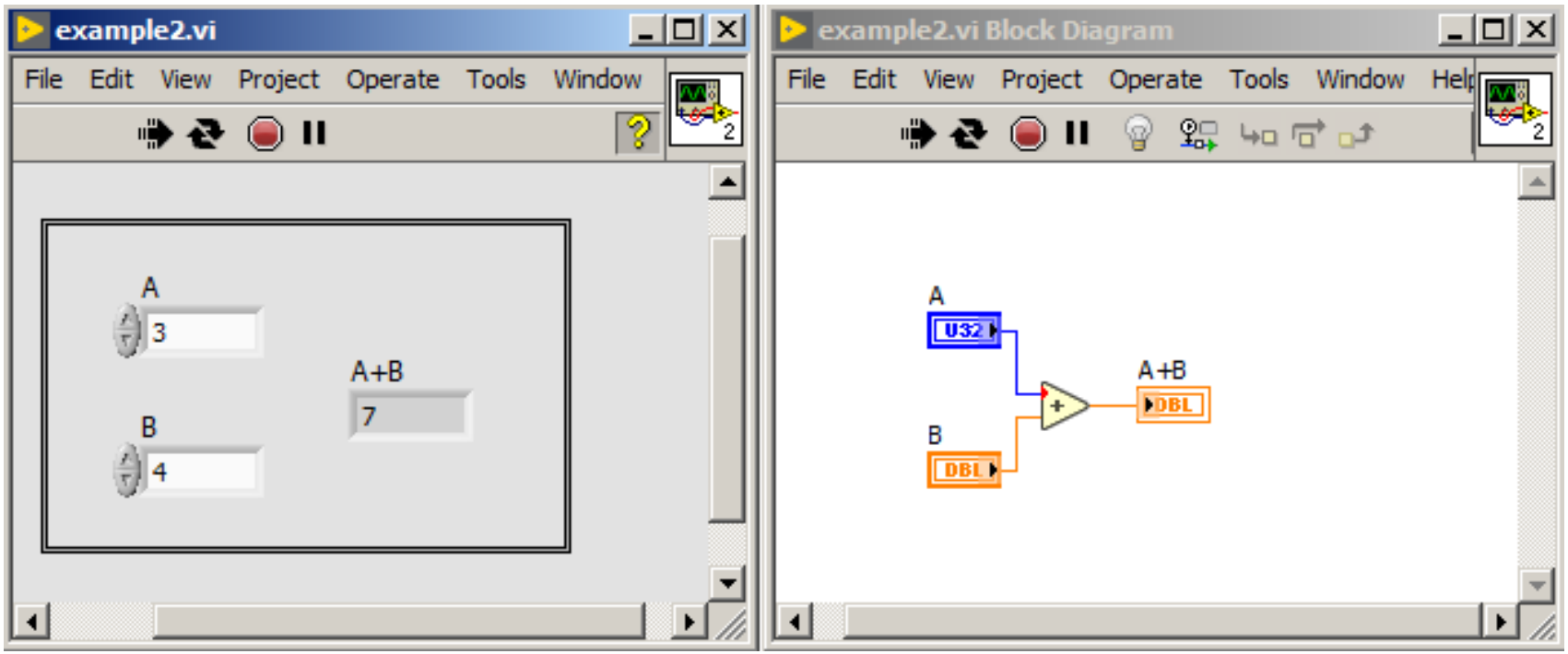


Icon and connector pane

First program

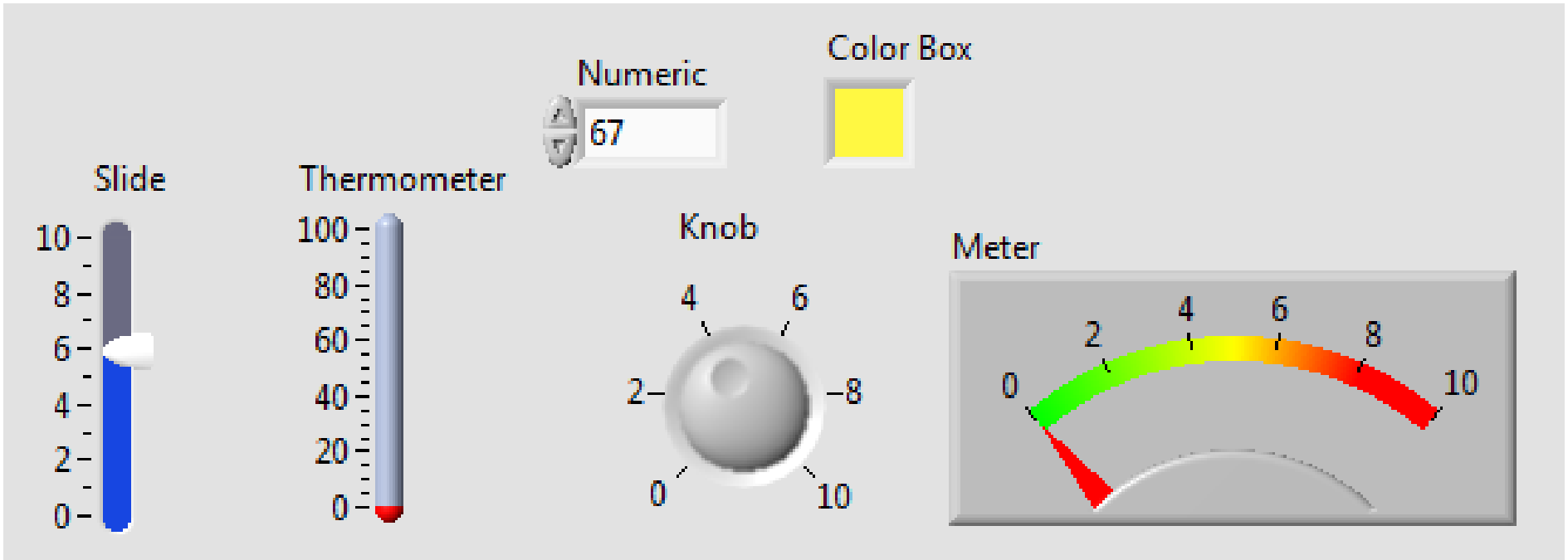
Type of terminals:

- **Controls** - input terminals,
- **Indicators** - output terminals.



Basic data types - numerical data

- Numerical data is represented by numbers. Numbers can be integer (e.g 2, -6, 158), real (e.g. -1.4, 0.23) or complex.

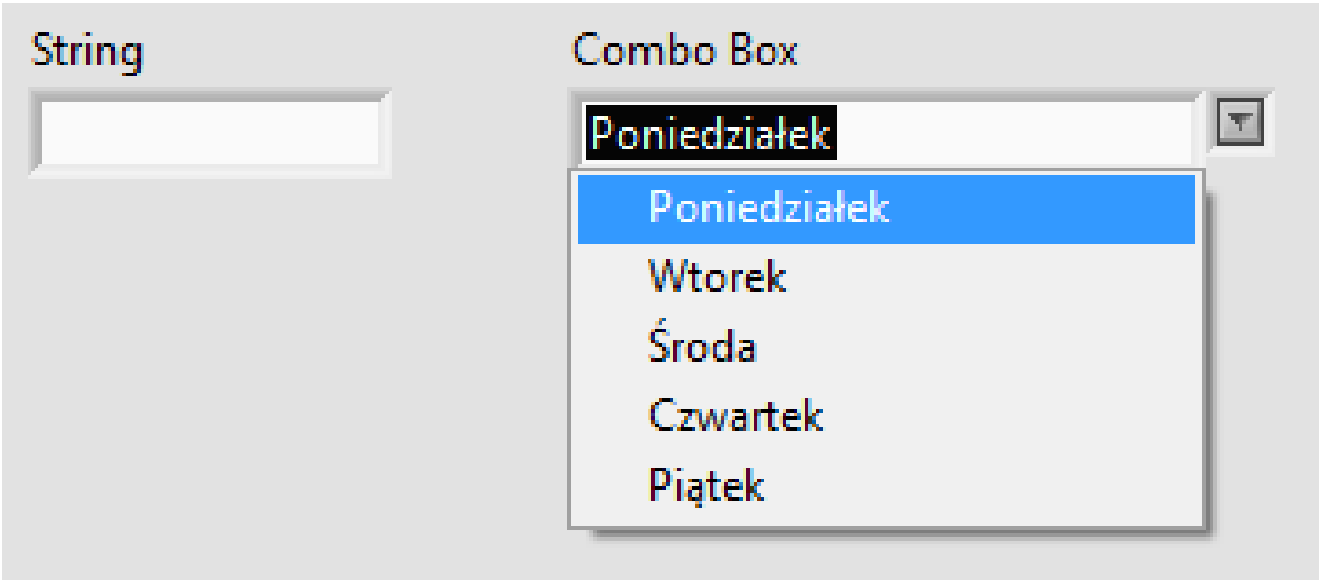


Basic data types - Boolean data

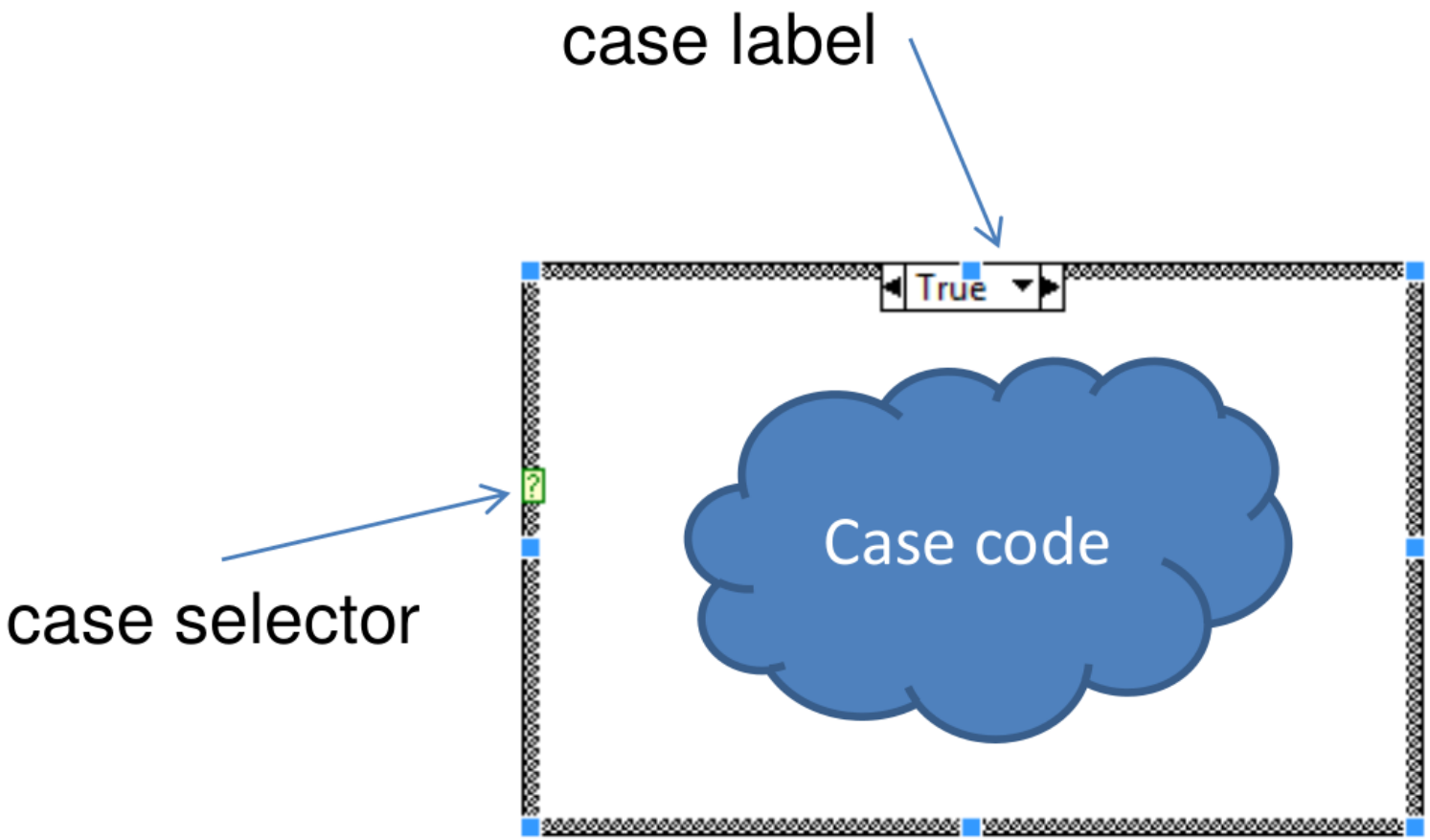
- Boolean data has only two possible values: *True* or *False*.

Basic data types - strings

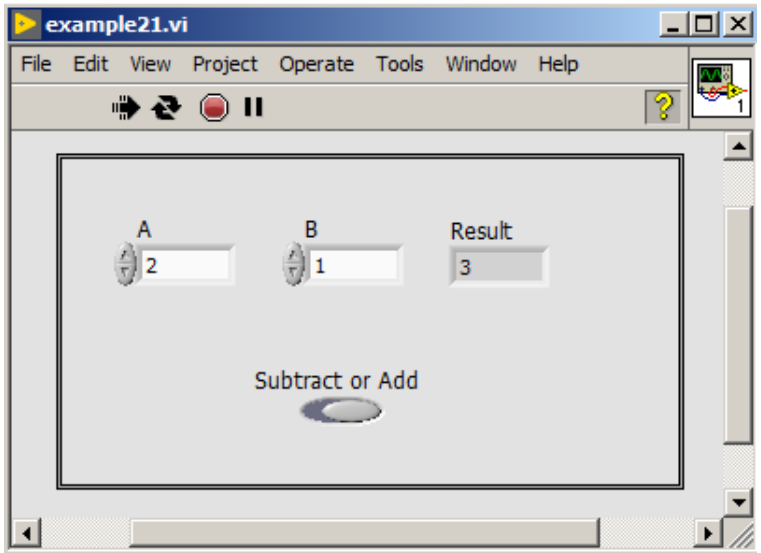
- String is a sequence of characters encoded using ASCII.

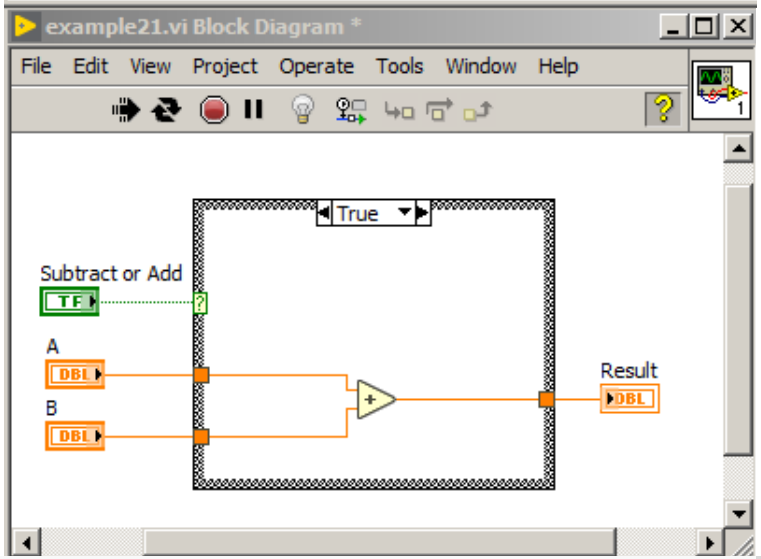


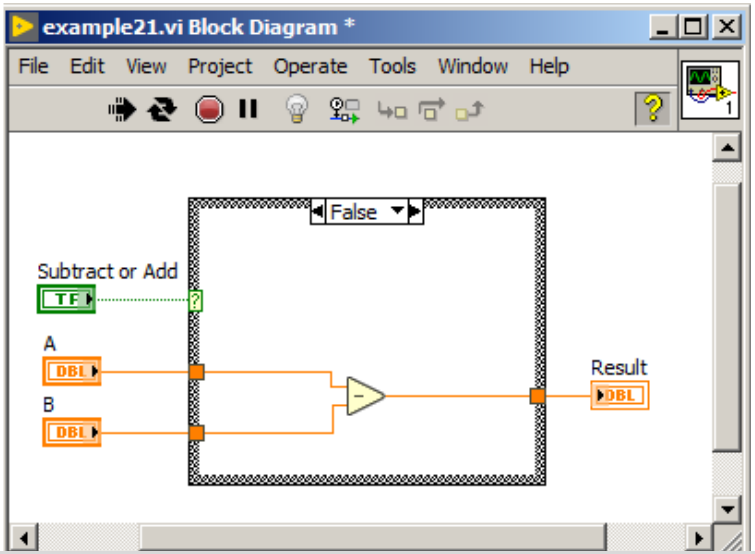
Case structure



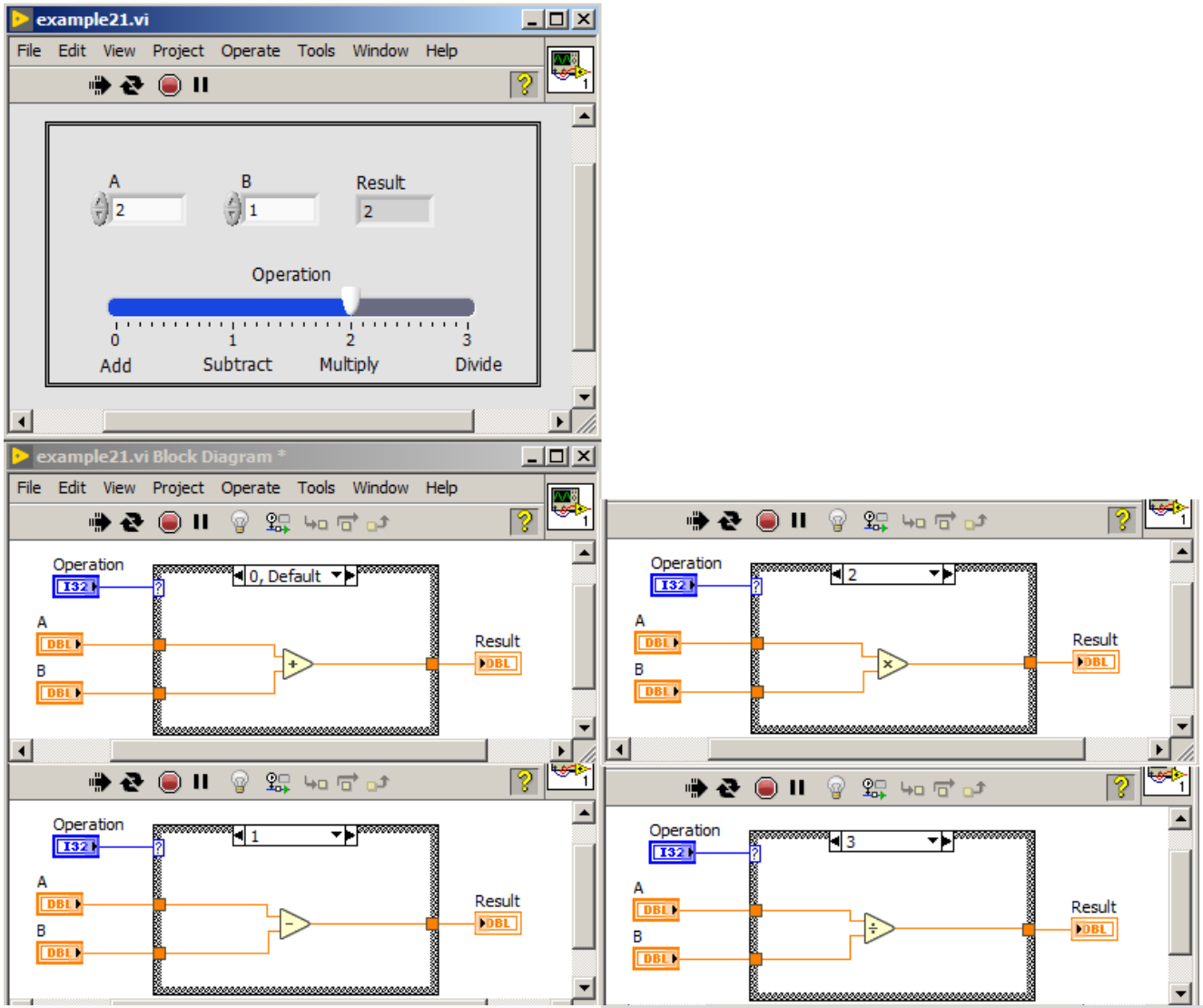
Case structure



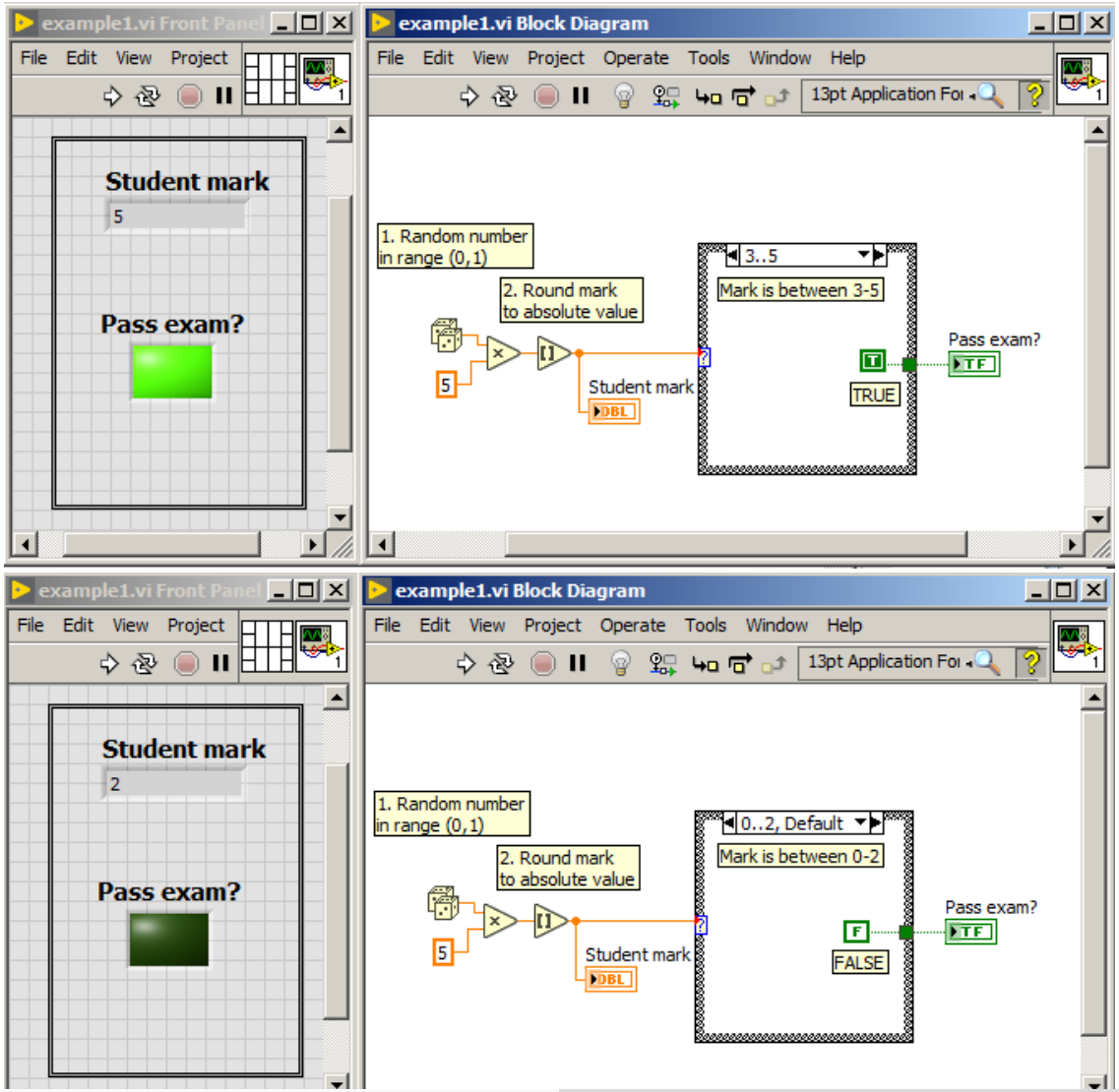




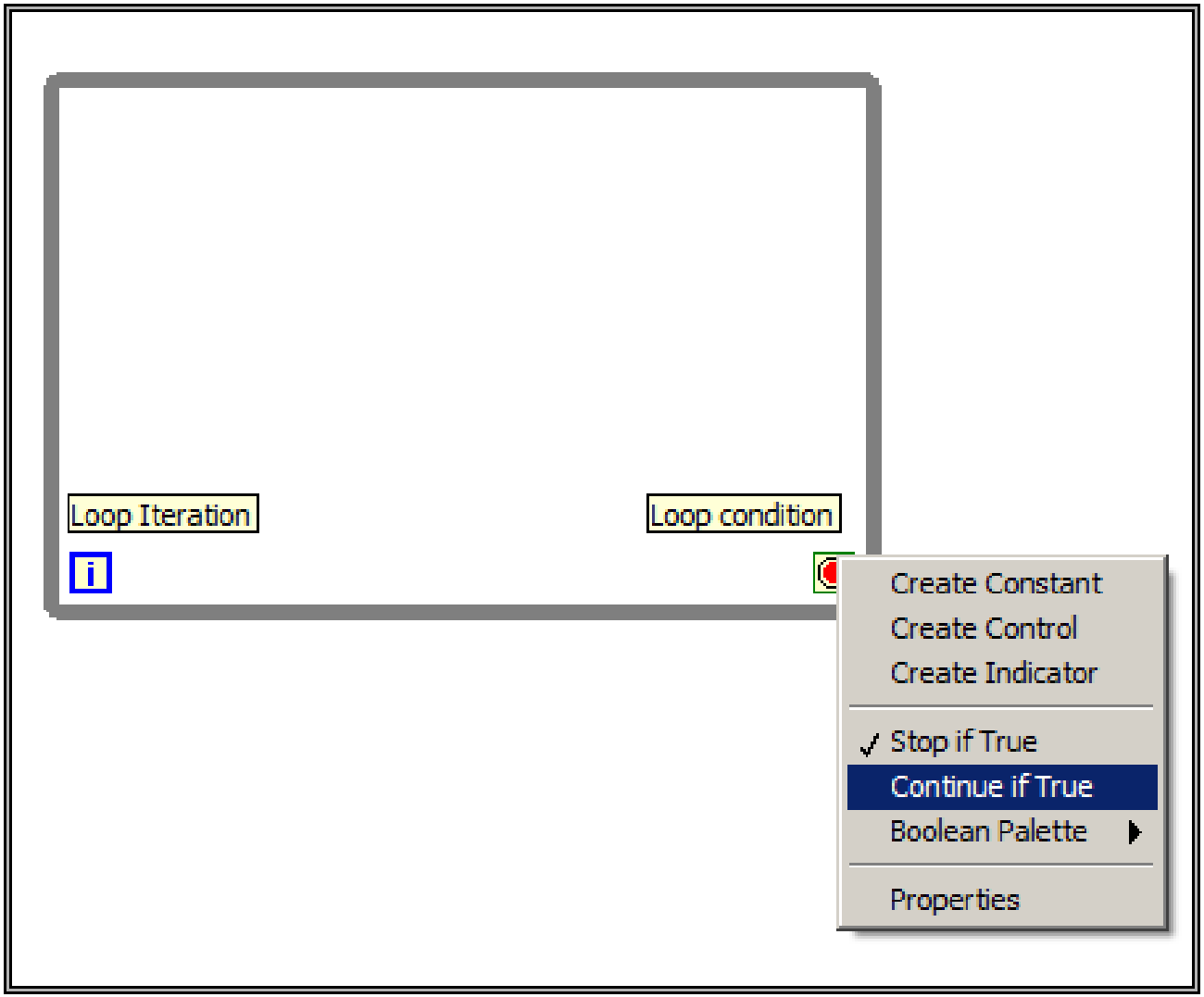
Case structure



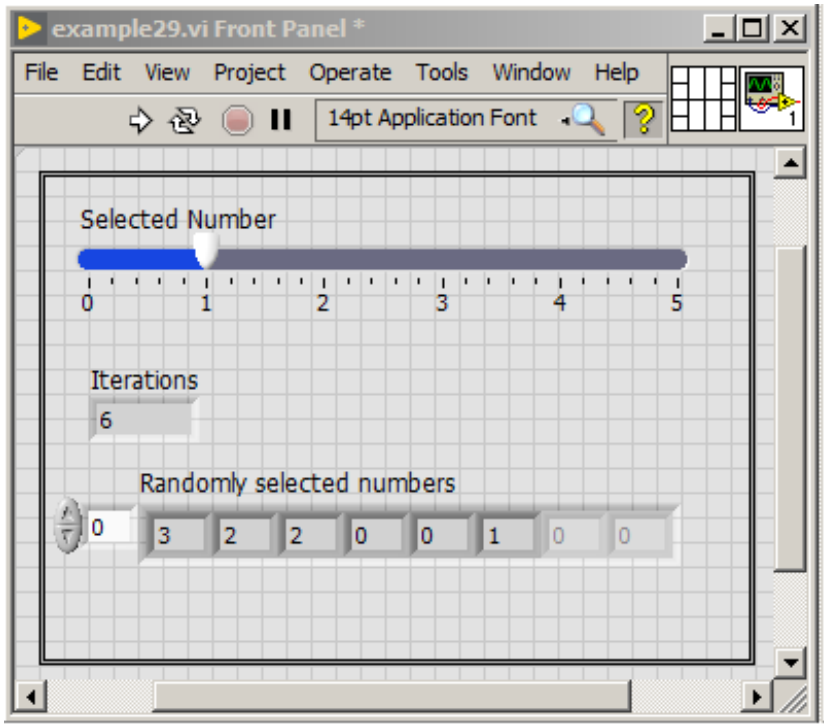
Case structure

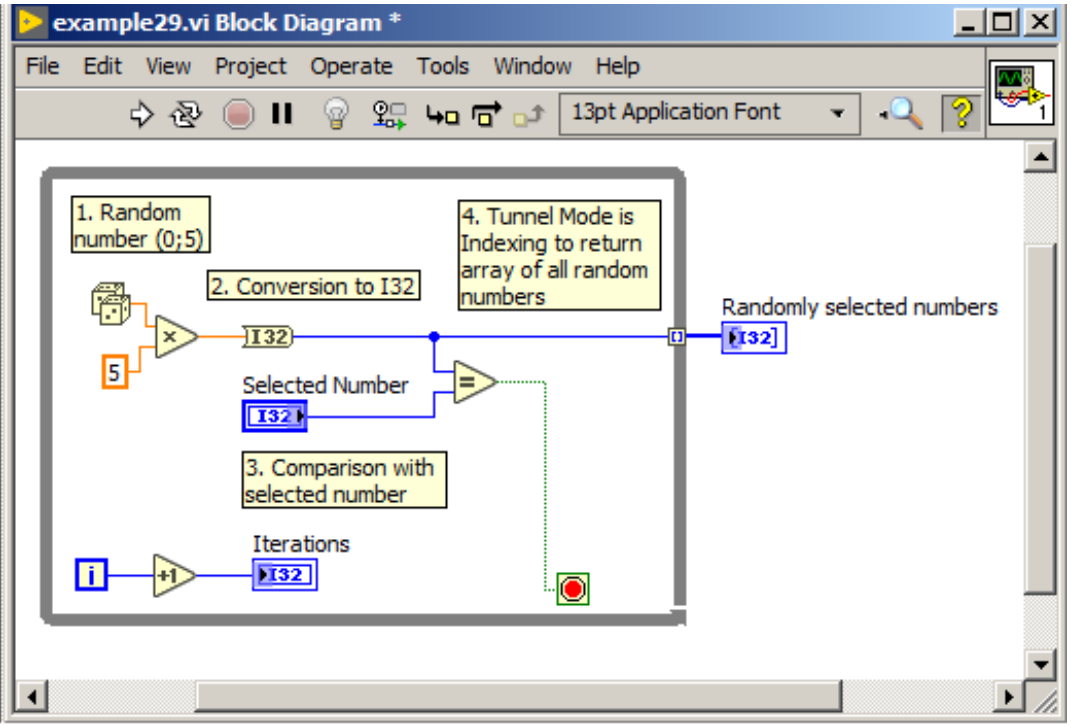


While loop

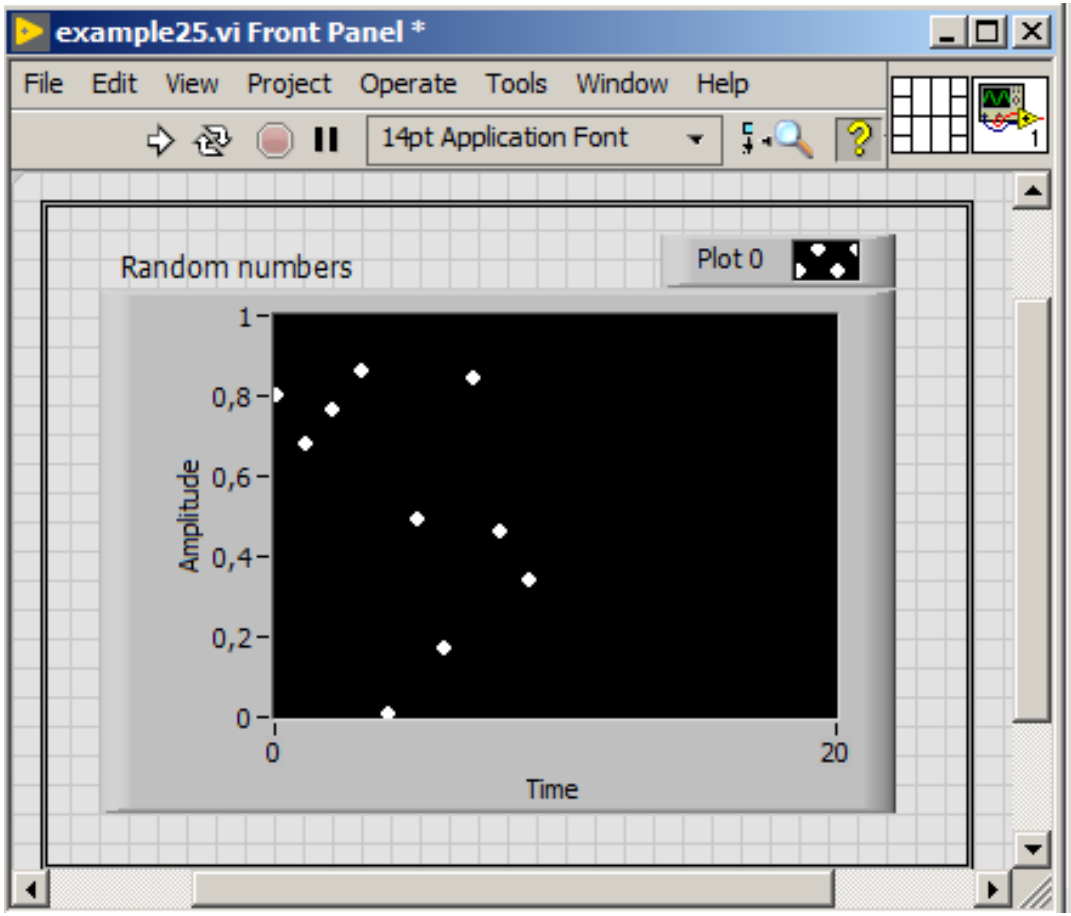

Loop Iteration
Loop condition
i
Create Constant
Create Control
Create Indicator
Stop if True
Continue if True
Boolean Palette
Properties

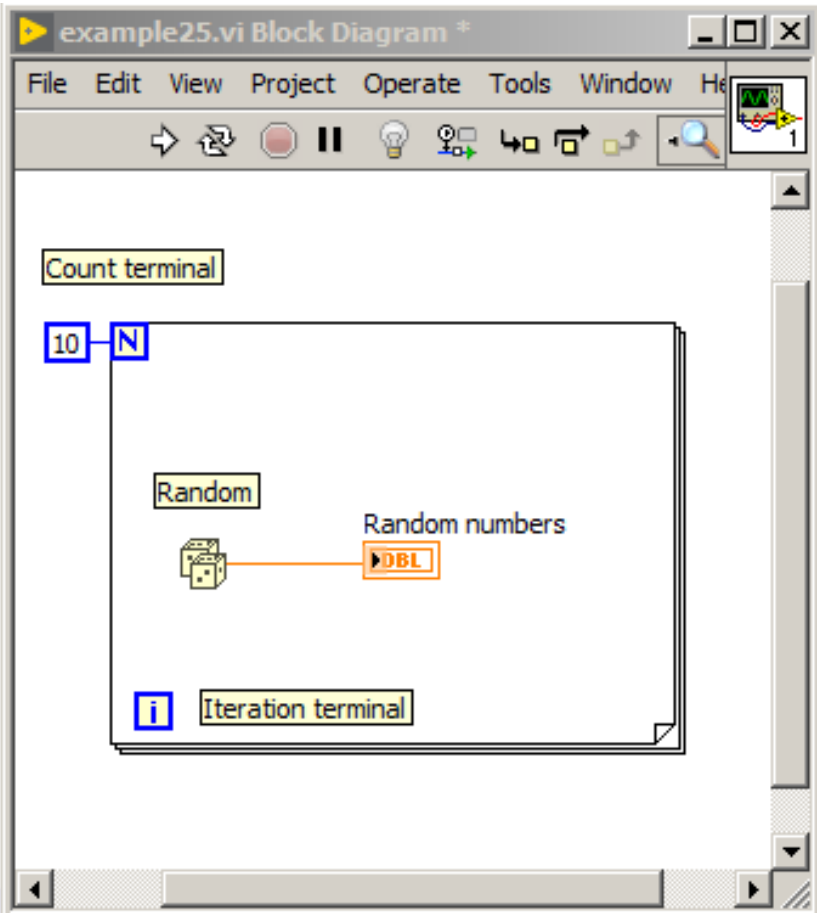
While loop


example29.vi Front Panel *
File Edit View Project Operate Tools Window Help
14pt Application Font
Selected Number
0 1 2 3 4 5
Iterations
6
Randomly selected numbers
0
3 2 2 0 0 1 0 0


example29.vi Block Diagram *
File Edit View Project Operate Tools Window Help
13pt Application Font
1. Random number (0;5)
2. Conversion to I32
4. Tunnel Mode is Indexing to return array of all random numbers
Randomly selected numbers
I32
5
Selected Number
I32
3. Comparison with selected number
Iterations
i
+1
I32

For loop


example25.vi Front Panel *
File Edit View Project Operate Tools Window Help
14pt Application Font
Random numbers
Plot 0
1
0,8
0,6
0,4
0,2
0
Amplitude
0
20
Time


example25.vi Block Diagram *
File Edit View Project Operate Tools Window
Count terminal
10
N
Random
Random numbers
DBL
i
Iteration terminal

LINX library

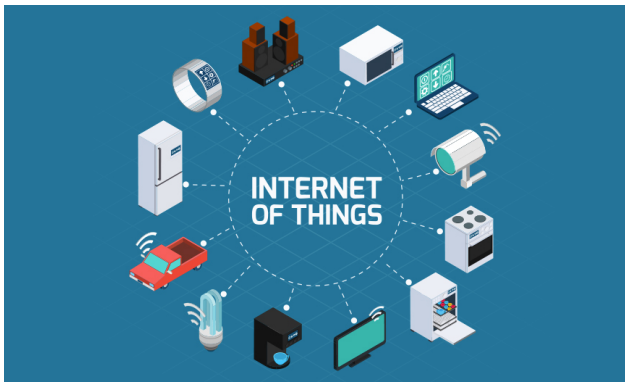
What is the Arduino?

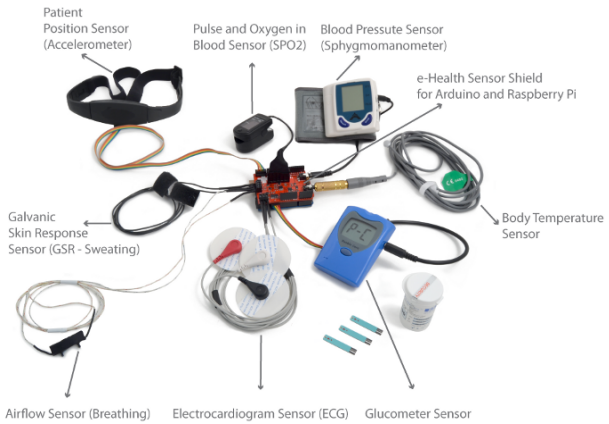
- Arduino is an open-source electronic platform, which heart is the 8-bit microcontroller Atmel AVR.
- This platform was created as a simple and inexpensive tool for prototyping for students without the electronic and programming background.

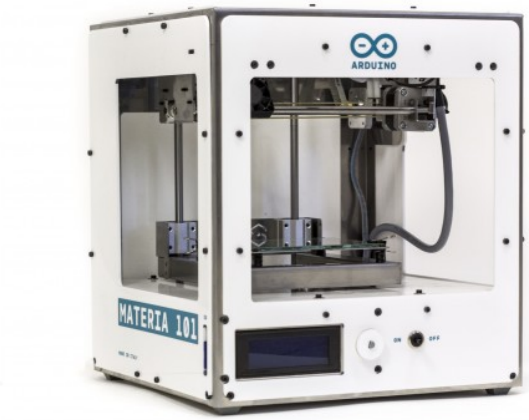
Fig.: The Arduino Mega 2560 board. Source: arduino.cc

What is Arduino?

- By its simplicity, it gained supporters around the world, especially in so-called makers (`https://makezine.com/`, `https://hackaday.com/`, `https://www.thingiverse.com/`).
- Due to the growing demand Arduino began to develop by adding new versions of boards dedicated to various purposes, e.g. Internet of Things (IoT), medical measurements, construction of 3D printers or embedded systems.







Why using the Arduino?

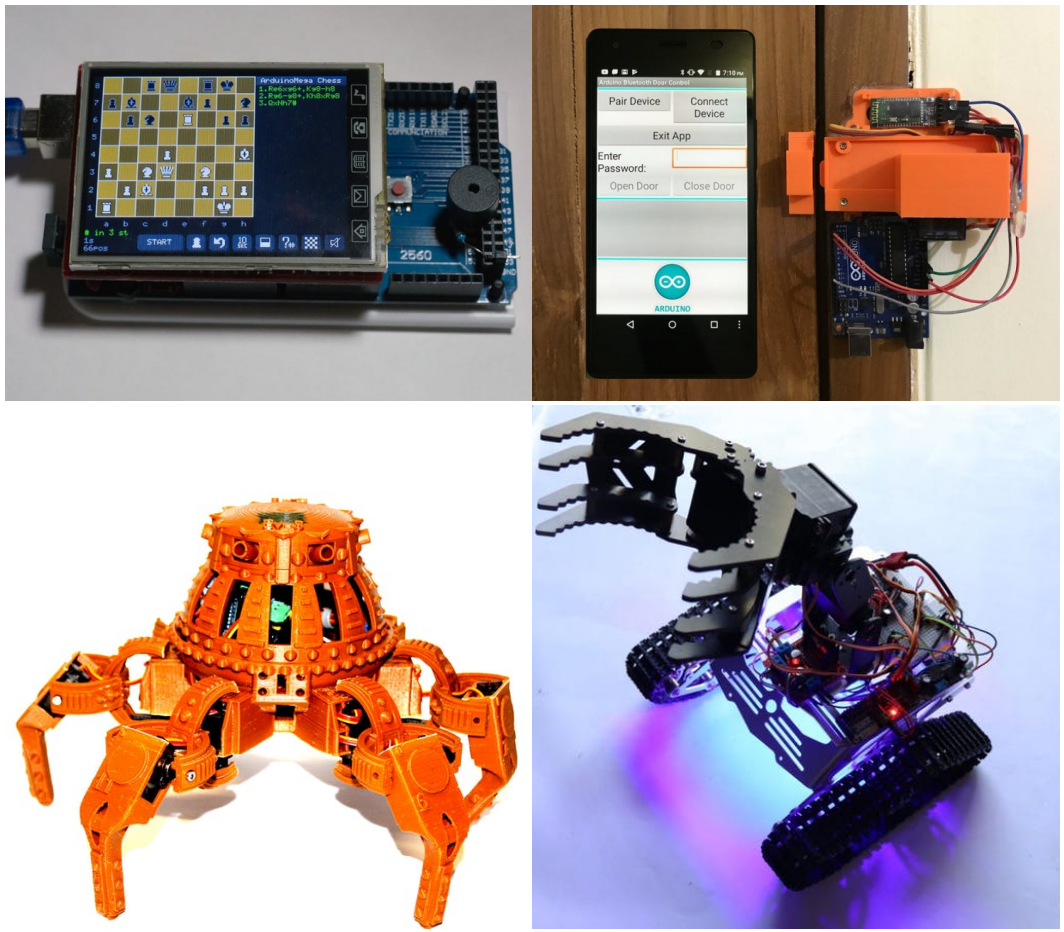


Fig.: Source: https://www.instructables.com, https://create.arduino.cc, https://interestingengineering.com.

LINX library

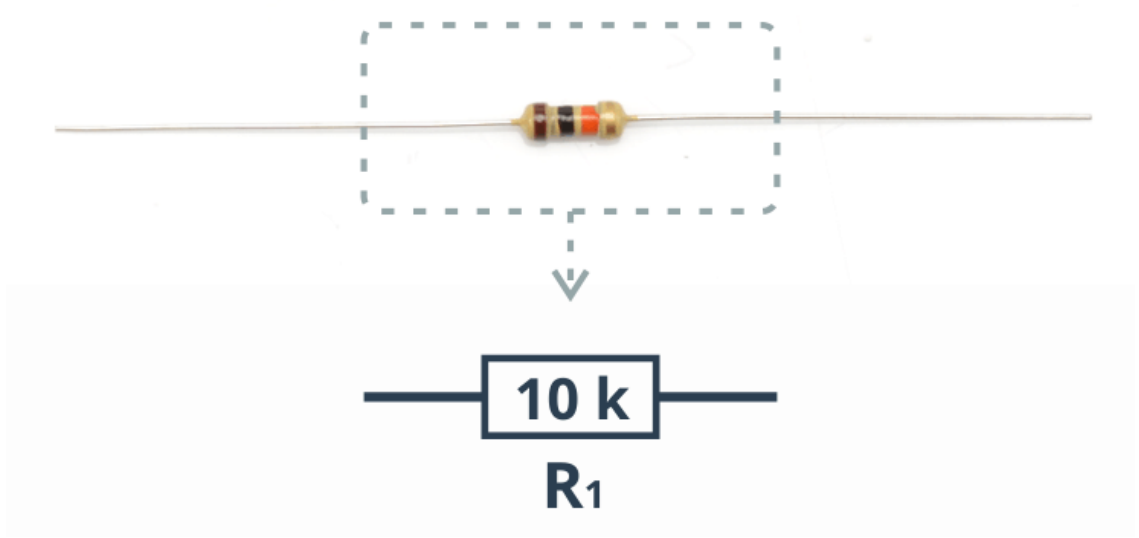
- LINX library can be used to communicate with Arduino, Raspberry Pi, myRIO and chipKIT,
- LINX was development by LabVIEW MakerHub (organisation, which supports makers movement),

LINX library

- The firmware to Arduino board can be upload by choosing: **Tools** → **MakerHub** → **LINX** → **LINX firmware Wizard...**

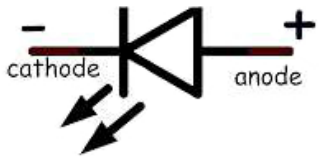
The basic elements - resistor

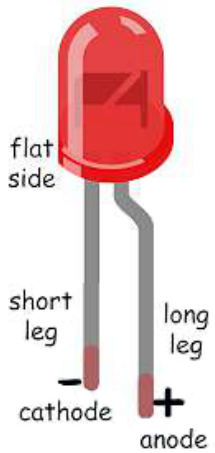
- **Resistor** is a two terminal element, which converts part of current to heat energy.
- This element is used to limit the current flowing through elements, e.g. can work for limiting current of LED diode.
- The higher value of resistor is used, the smaller current will flow through a circuit. This is described by the Ohm law: $I = U/R$.



The basic elements - LED diode

- The **LED** diode converts the electric current to the light.
- The LED diode is a polarized element, what means that current should flow through the LED only in one direction.
- Anode should be connected to the higher voltage and cathode to a lower voltage e.g. to ground (GND).
- Anode is the longer terminal of LED diode. Cathode is shorter terminal.





The basic elements - LED diode

- The typical LED diode voltage drop is about 1.7 V (U_{LED}) and current through LED should not exceed 20 mA.
- The Arduino boards generate the 5 V (U_{DO}) on the digital outputs so that resistor should be used to limit the current.
- The value of resistance should be between (220;3300) Ω according to the equation:

$$R = \frac{U_{DO} - U_{LED}}{I} \tag{1}$$

The basic elements - breadboard

- Breadboard is a very useful device which helps to connect other elements.
- It should be used at the beginning of electronic circuit creation to check circuit behaviour.

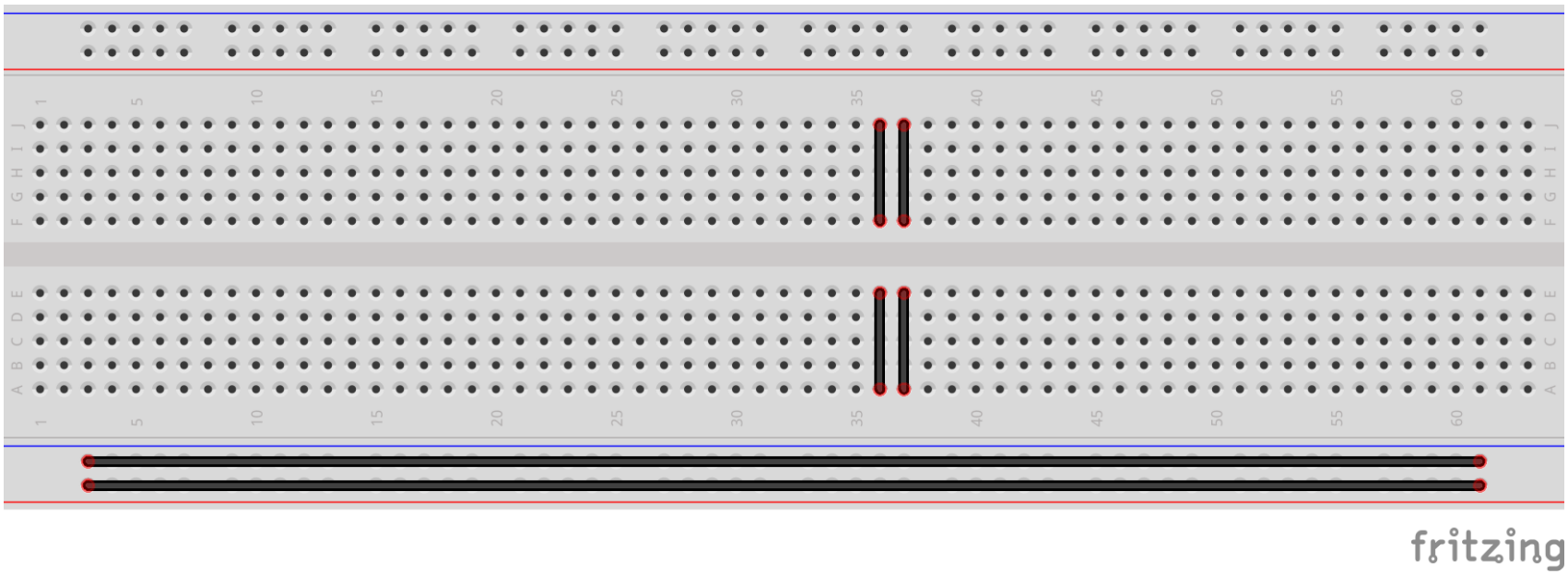
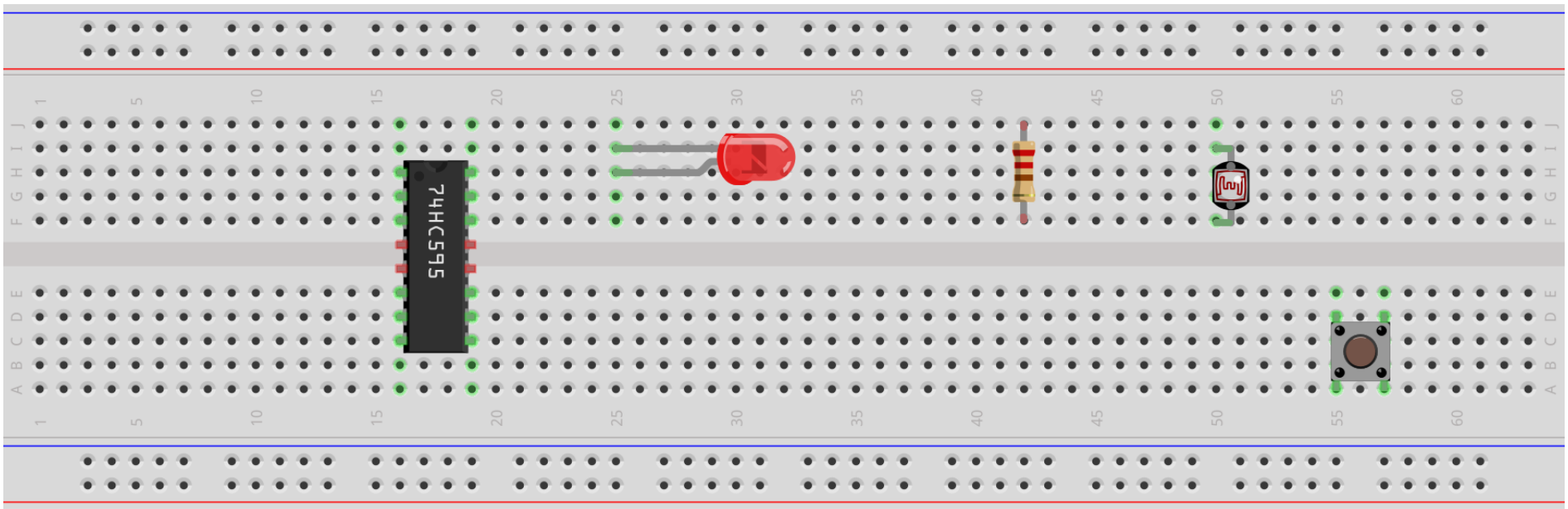


Fig.: An example of breadboard. The black lines show, which holes are connected.

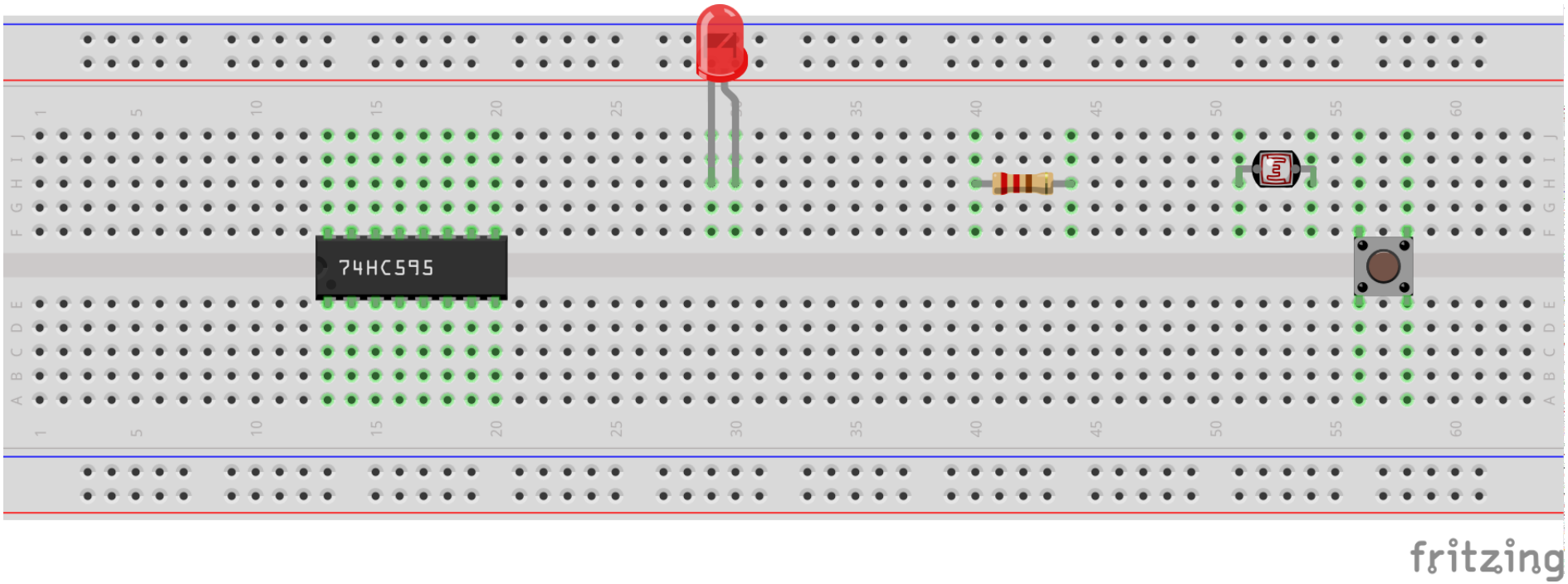
The basic elements - breadboard

- In figure below, **wrong placed** elements are shown. All terminals of elements are shorted so elements will not work.

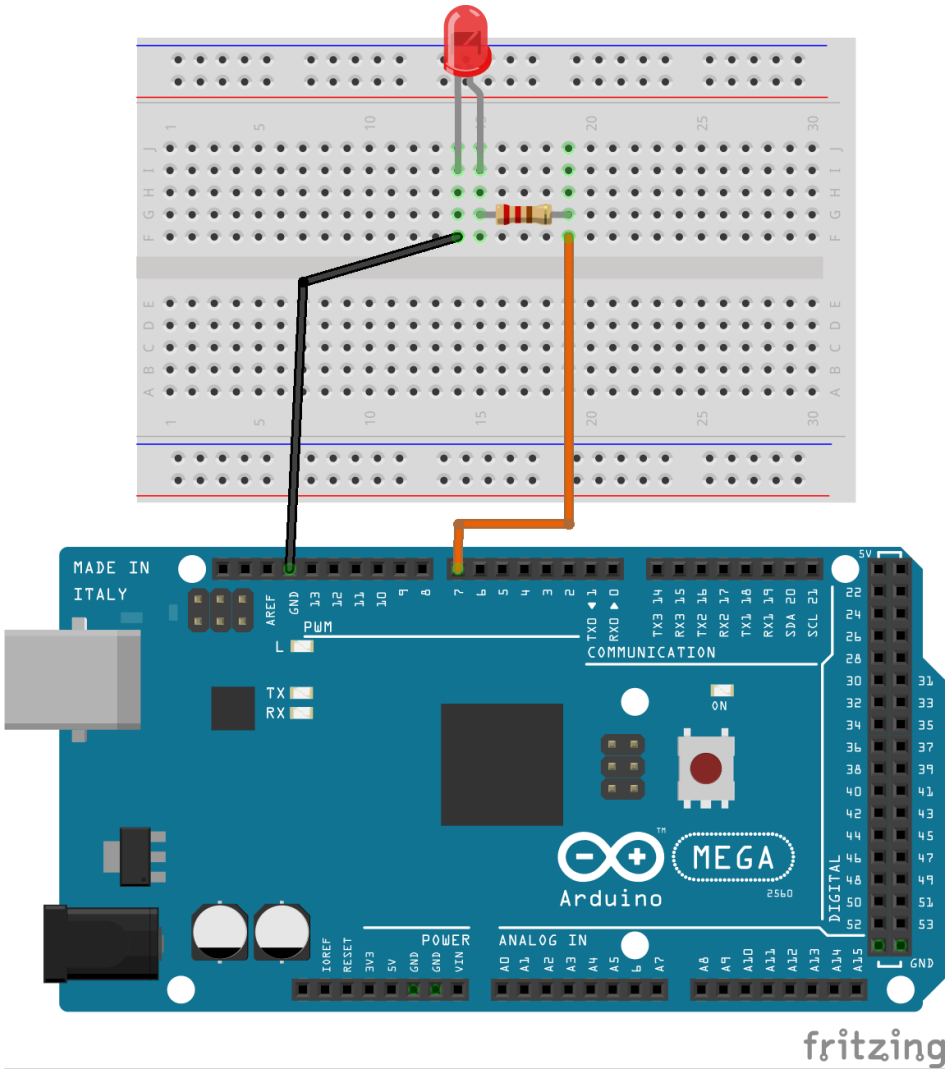


The basic elements - breadboard

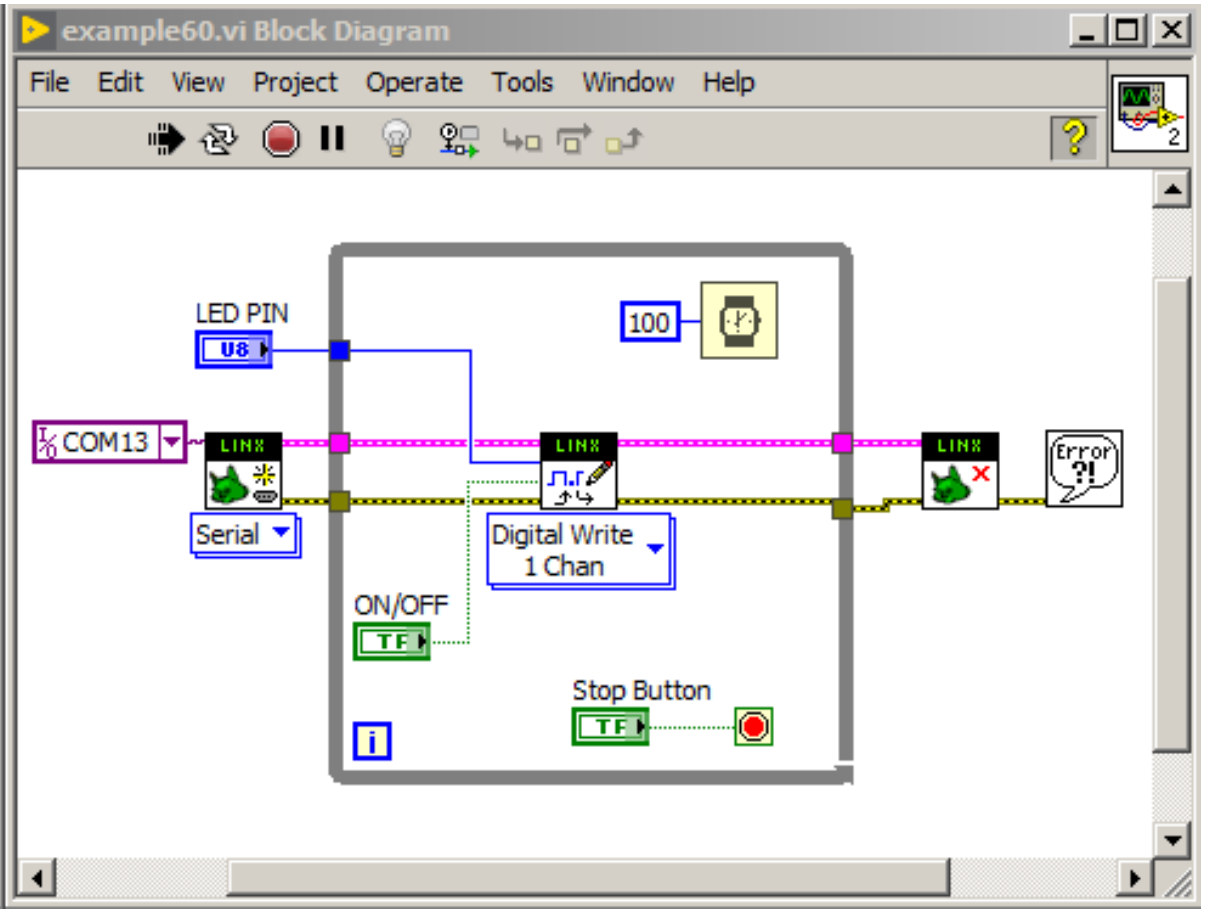
- In figure below correctly placed elements on breadboard are shown.



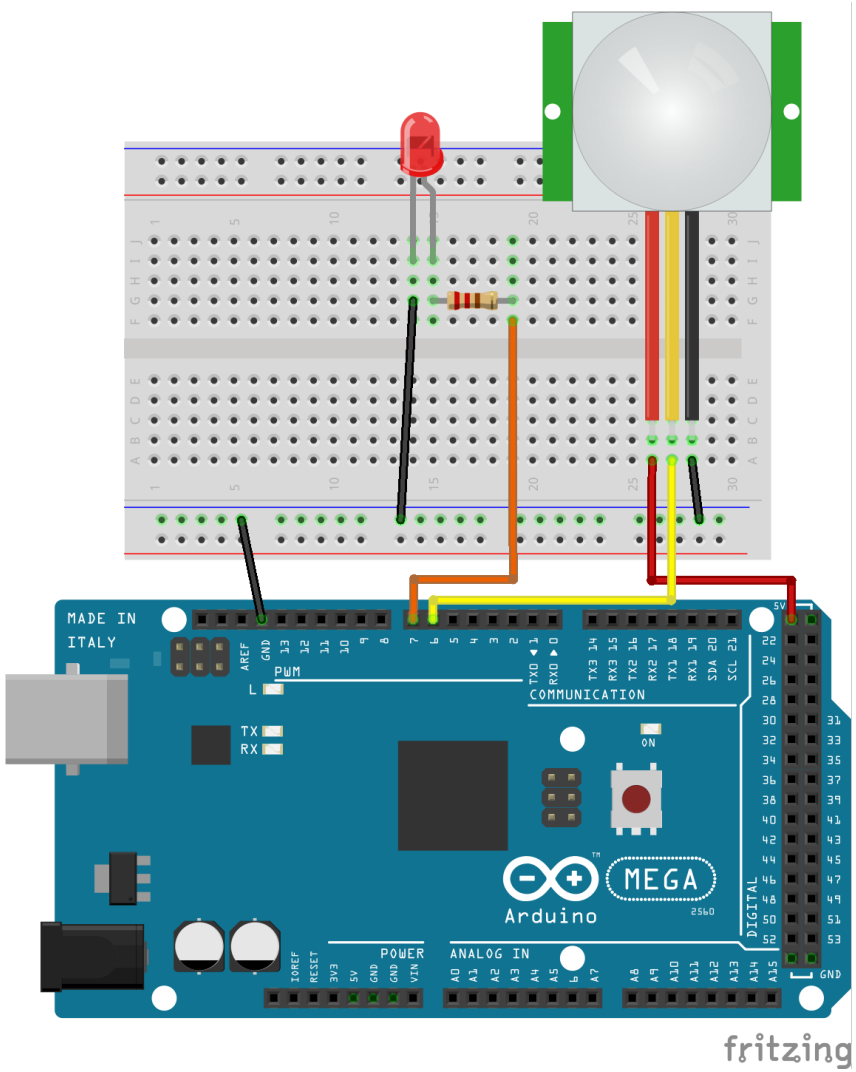
Basic functions - *Digital Write*



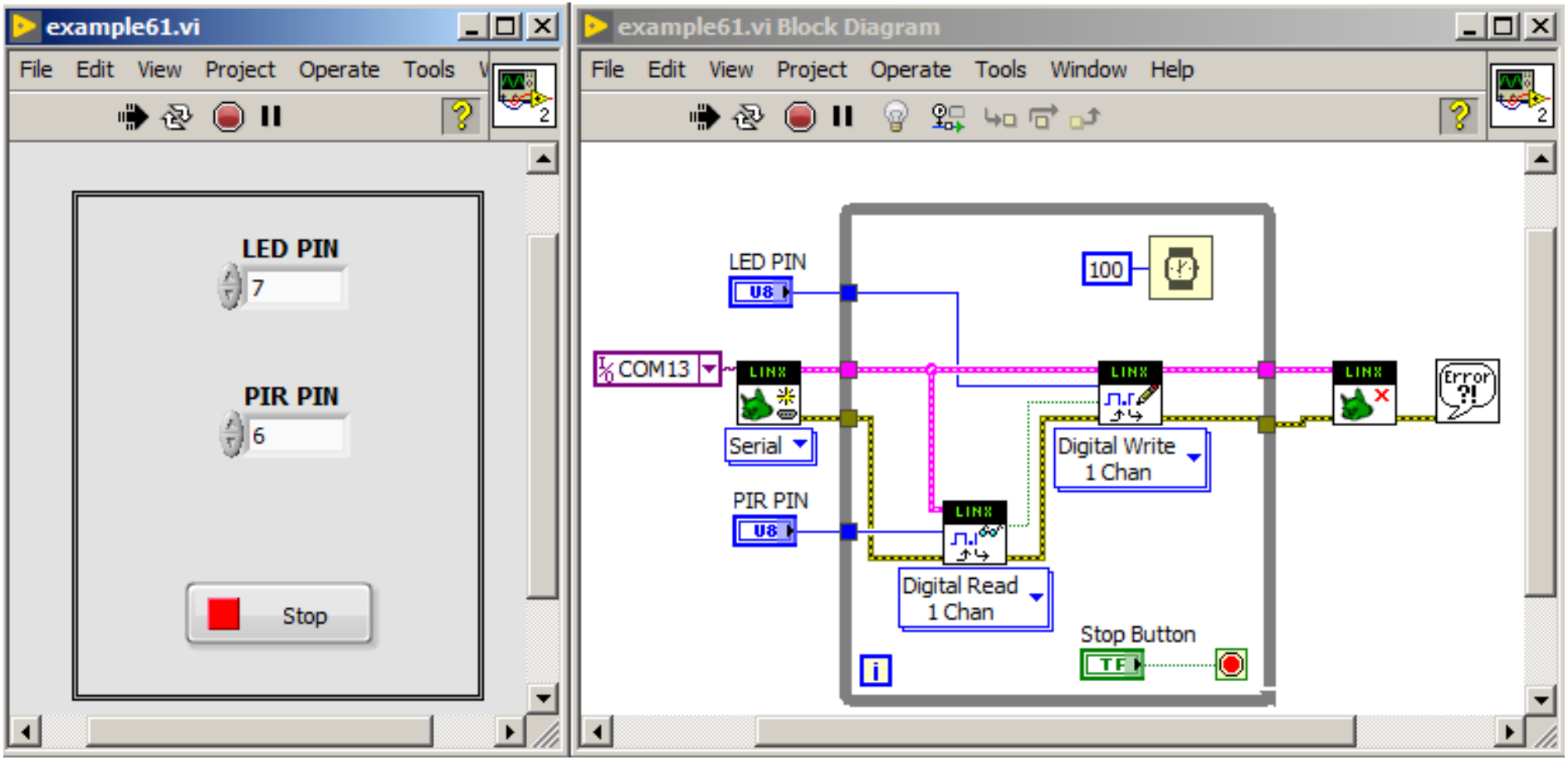
Basic functions - *Digital Write*



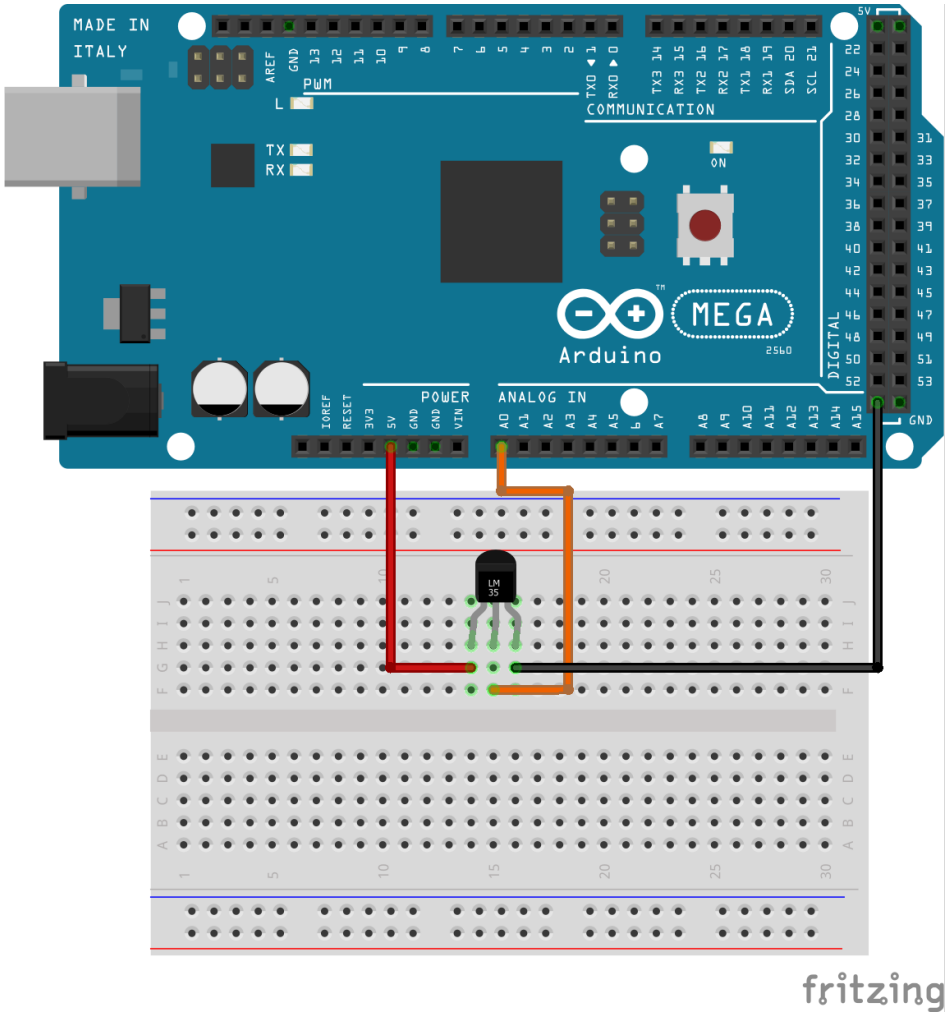
Basic functions - *Digital Read*



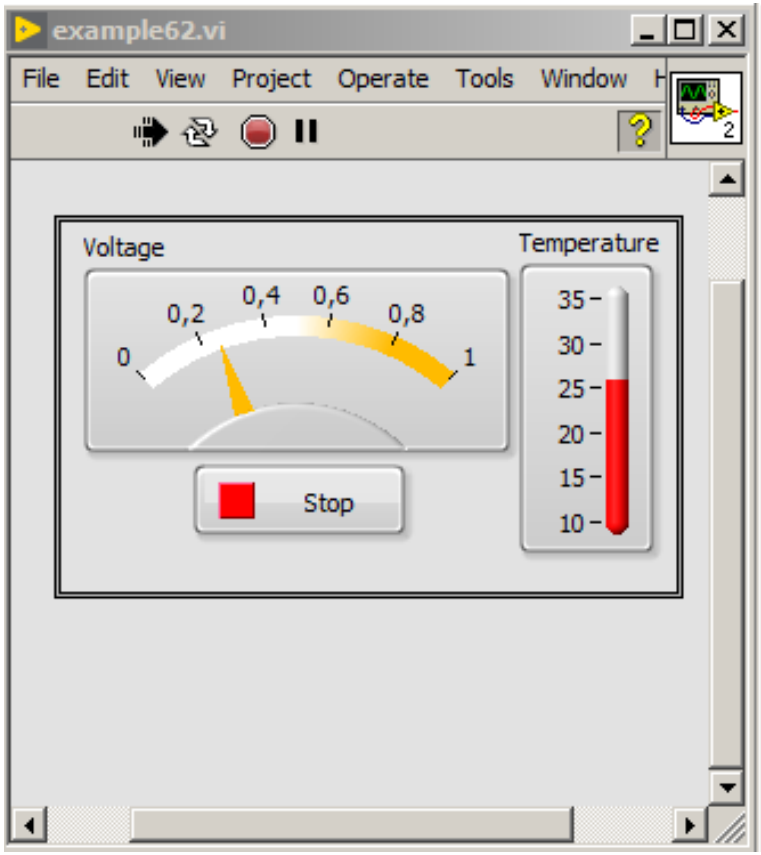
Basic functions - *Digital Read*

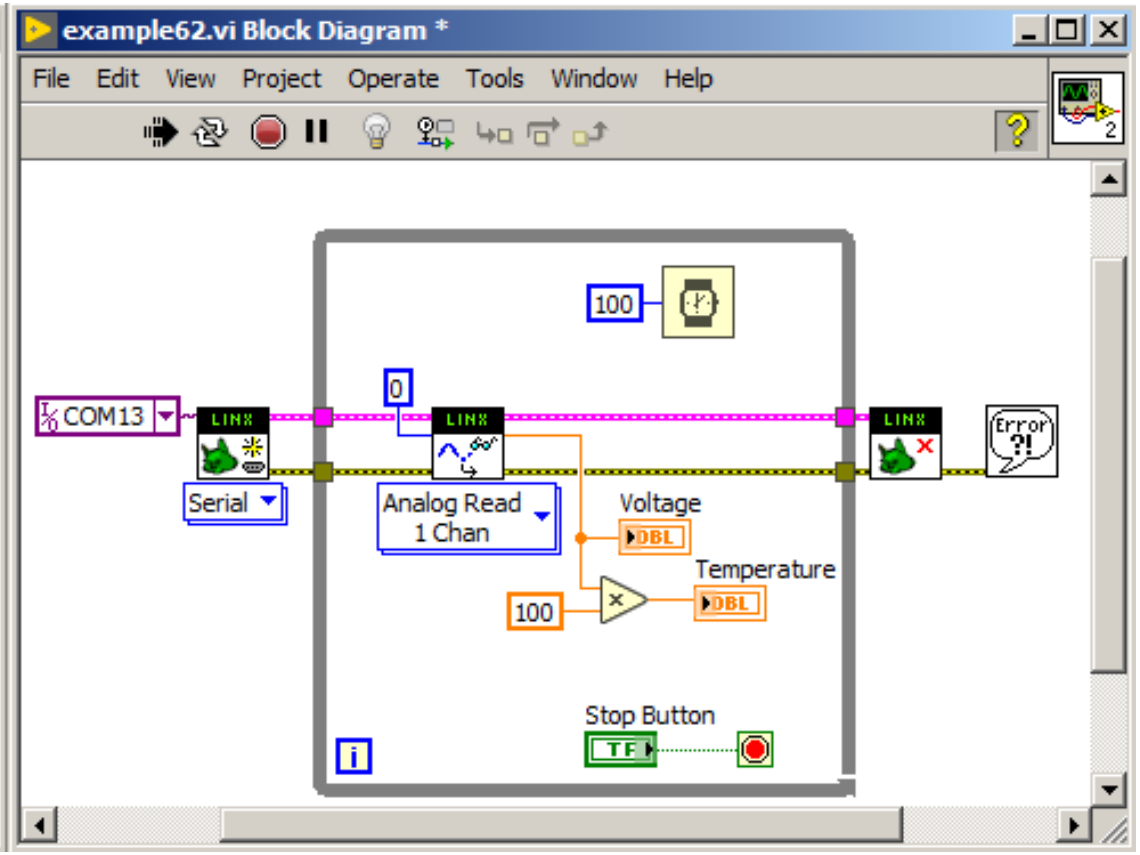


Basic functions - *Analog Read*

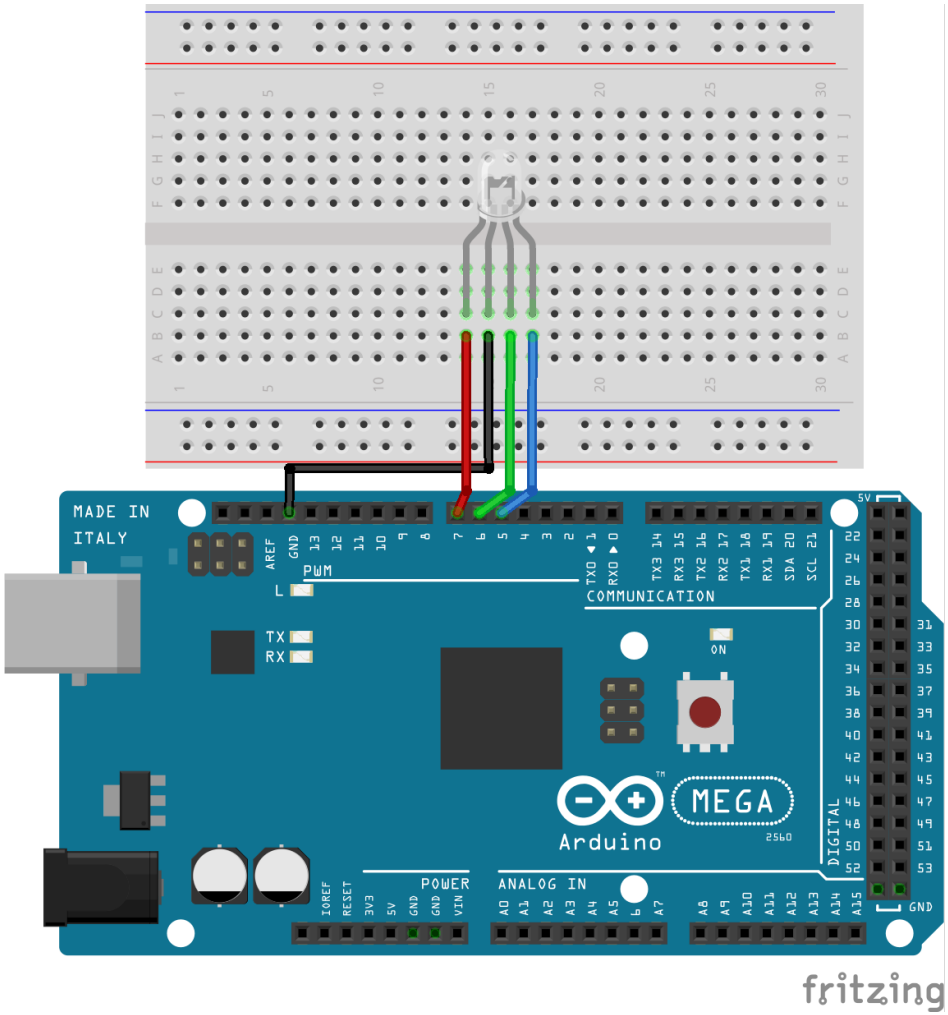


Basic functions - *Analog Read*

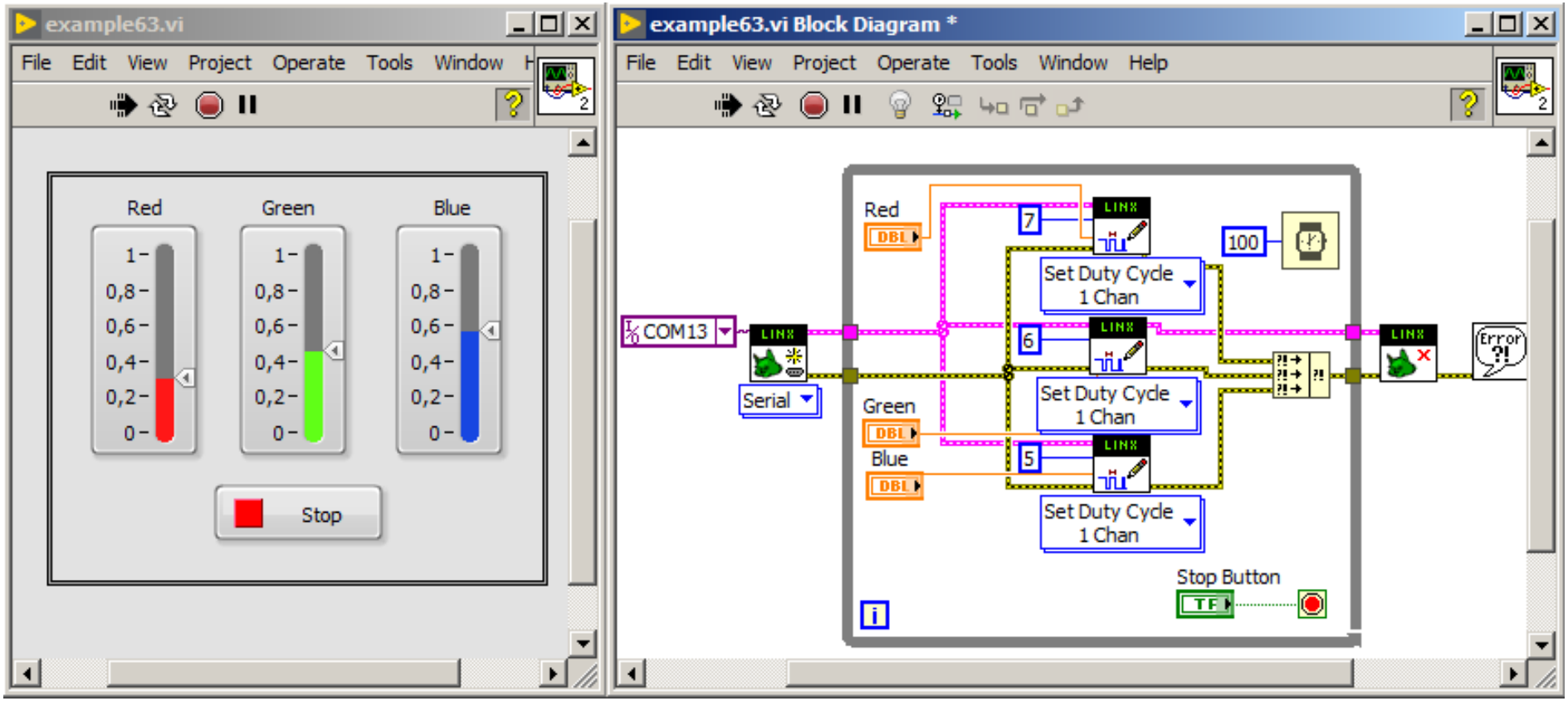




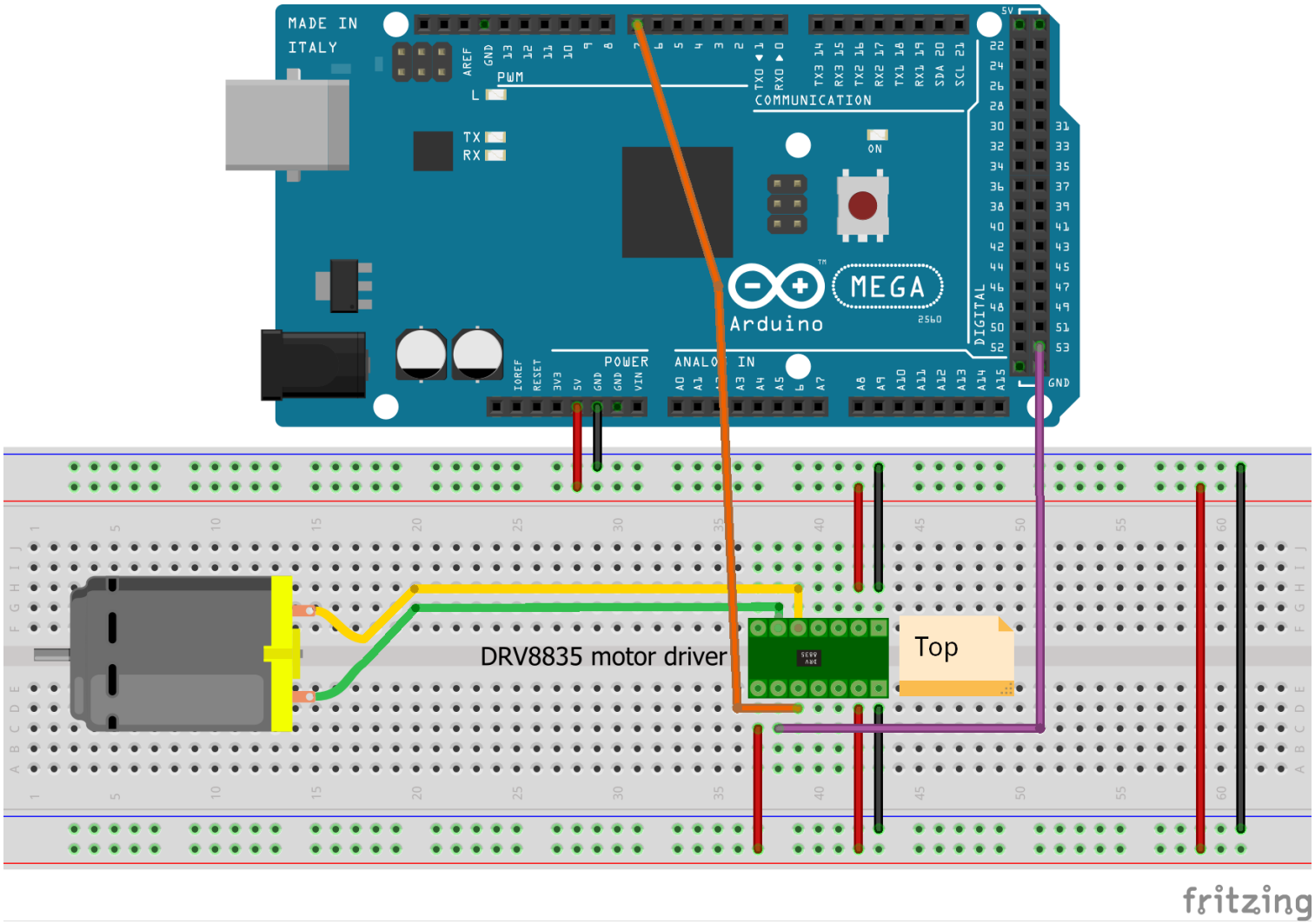
Basic functions - *PWM Set Duty Cycle*



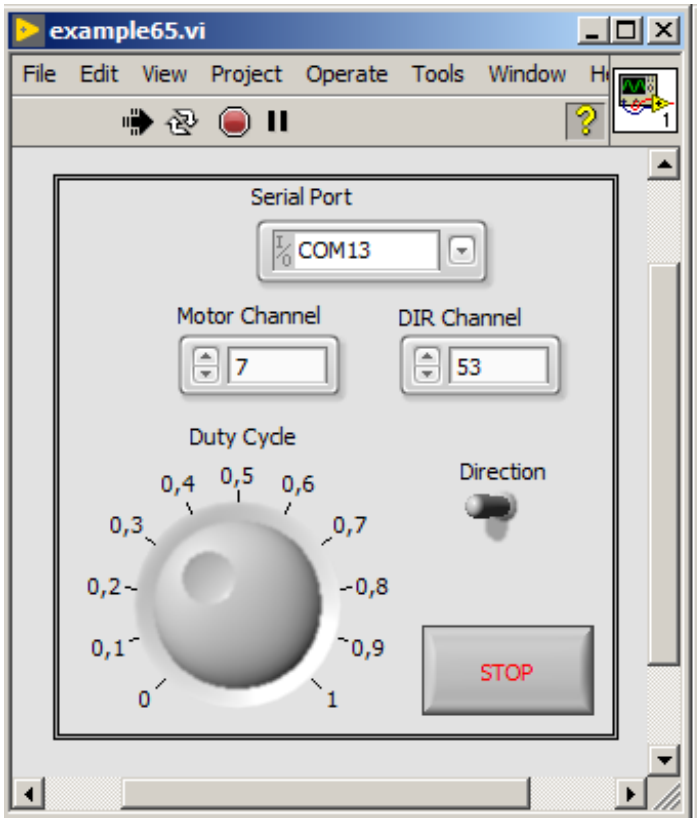
Basic functions - *PWM Set Duty Cycle*

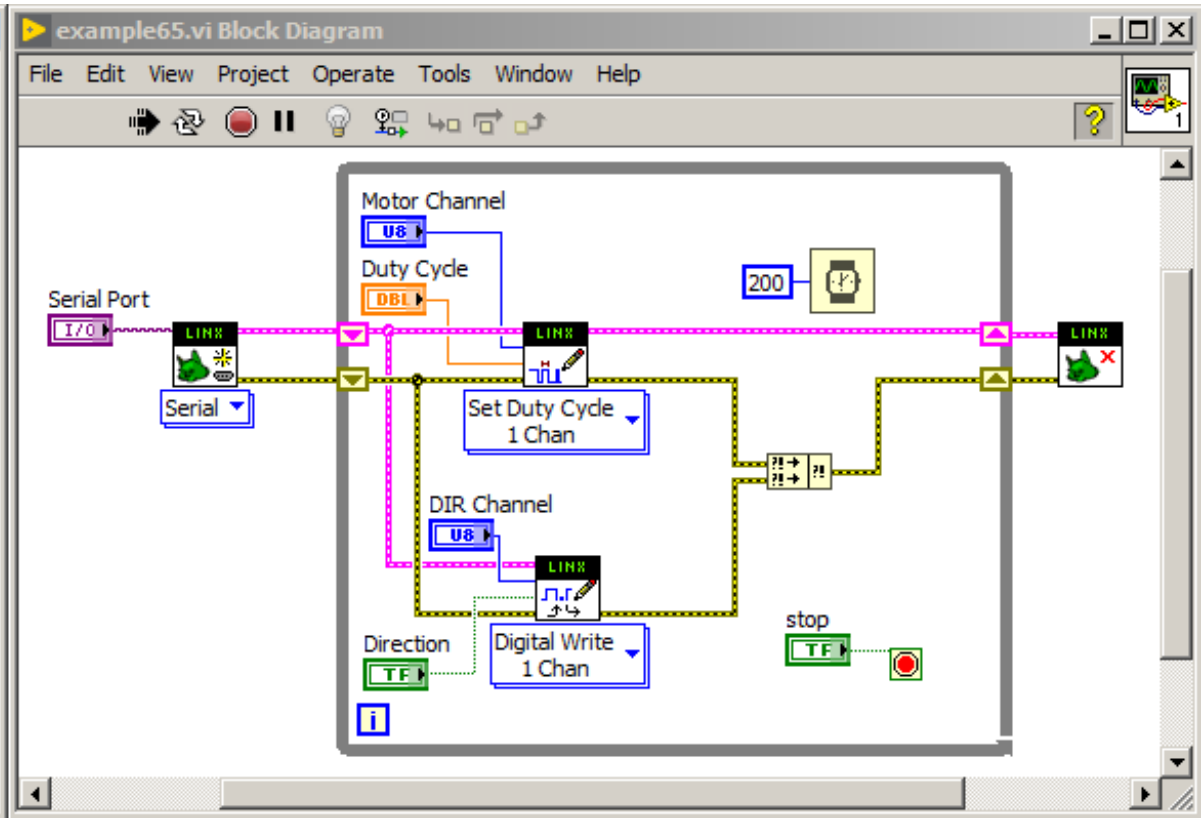


Basic functions - *PWM Set Duty Cycle*



Basic functions - *PWM Set Duty Cycle*





Thank you for your attention!

It's time for practical exercise

We invite you to room 229 in Faculty of Physics!