SZÉCHENYI ISTVÁN EGYETEM
GÉPÉSZMÉRNÖKI, INFORMATIKAI ÉS
VILLAMOSMÉRNÖKI KAR

DESIGN, BUILD AND MEASURE OF DC/DC BUCK CONVERTER

Project work

FOR VEHICLE ENGINEER STUDENTS

Version: 1.0

Széchényi István University
Department of Automation

1. Introduction, preparation

The project will involve calculating a DC/DC buck converter parameters and using the calculations to build that. A right operation will be checked by measuring.

1.2 Design example

We would like to design a DC/DC buck converter with the following parameters:

- Input voltage (V_{in}) = 24VDC;
- Output voltage (V_{out}) = 5VDC;
- Frequency (f) = 535kHz;
- Output current (I_{out}) = 2A;
- Output voltage ripple (ΔV_{out}) = 50mV.

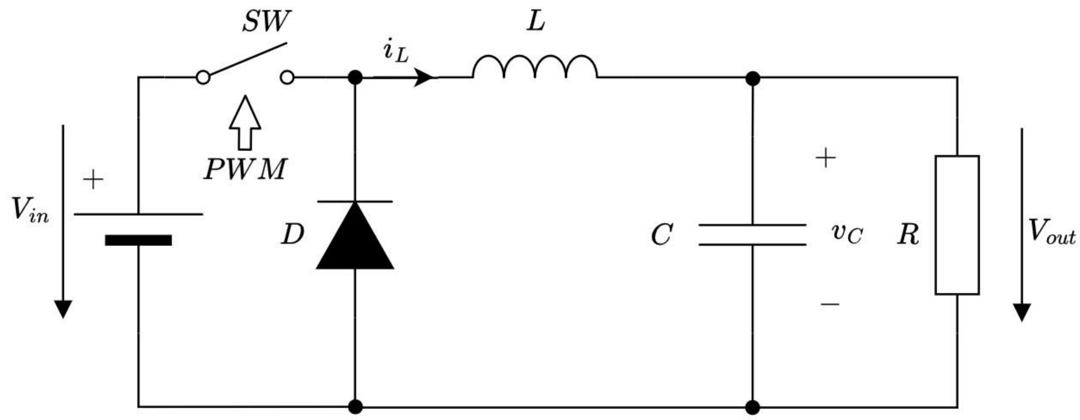


Fig.1. DC/DC buck converter shematic

The duty cycle is given by:

$$D = \frac{V_{out}}{V_{in}} = \frac{5V}{24V} \cong 0.21 = 21\%$$

The period time of switching frequency is:

$$T = \frac{1}{f} = \frac{1}{535kHz} = 1.86us$$

The on time is:

$$t_{on} = D \cdot T = 0.21 \cdot 1.86us = 0.39us$$

The off time is:

$$t_{off} = (1 - D) \cdot T = 0.21 \cdot 1.86us = 1.47us$$

What do these mean? (Fig. 1.)

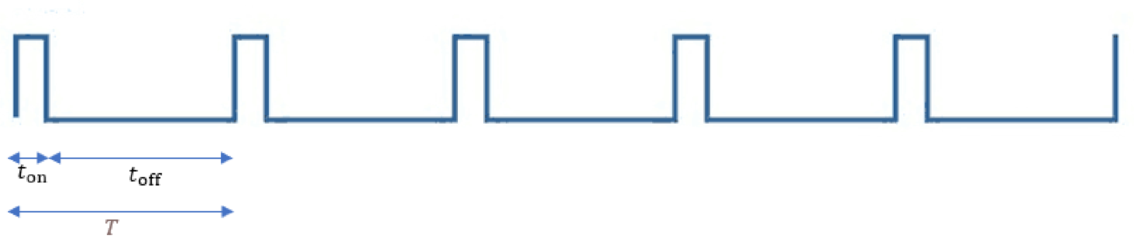


Fig.1. PWM signal properties

It is good design practice to keep the inductor ripple current (*di*) to about 40% of the output current, accordingly with a 2A load this implies a ripple current of 800mA (ΔI_L). Increasing the ripple current increases output ripple, but means we can use a smaller value and size of inductor.

The current ramp in the inductor during t_{on} time is represented by:

$$\frac{V_{in} - V_{out}}{L} = \frac{di}{dt_{on}}$$

Or in t_{off}:

$$\frac{V_{out}}{L} = \frac{di}{dt_{off}}$$

So, we can calculate the inductor value (e.g. during the ton interval):

$$L = (V_{in} - V_{out}) \cdot \frac{dt_{on}}{di} = (12V - 5V) \cdot \frac{0.39us}{0.8A} \cong 10uH$$

The capacitor ESR (Equvivalent Serial Resistance):

$$ESR \leq \frac{\Delta V_{out}}{2 \cdot \Delta I_L} = \frac{50mV}{2 \cdot 0.8A} = 31.25m\Omega$$

For example, let's choose 20mΩ. Fig. 2. helps to understand the esr. ESR is the DC resistance of capacitors, which is independent of frequency. There value is typically mΩ.

Fig.2. Capacitor ESR

Calculating the capacitor:

$$C = \frac{\Delta I_L}{(\Delta V_{out} - \Delta I_L \cdot ESR) \cdot 8 \cdot f_{sw}} = \frac{0.8A}{(50mV - 0.8A \cdot 20m\Omega) \cdot 8 \cdot 535kHz} = 5.4uF$$

We apply a little bit bigger capacitance, so we choose 10uF.

The output load resistance in case of 2A:

$$R_{load} = \frac{V_{out}}{I_{out}} = \frac{5V}{2A} = 2.5\Omega$$

1.2.1 Summary

Input Parameters	Value
Input voltage	24VDC
Output voltage	5VDC
Output voltage ripple	1%
Output current	2A
Switching frequency	535kHz
Output parameters	**Value**
Duty cycle	0.21
Inductor	10mH
Inductor ripple current	0.8A
Capacitor	10uF
Output load	2.5Ω

1.3 Simulation of design example converter

The designed DC/DC buck converter with the calculated paramters simulated by Ltspice. It is a free circuit simulator program. The goal is to check the correctness of the calculated parameters using the simulation. See the lecture for more details.
The converter circuit diagram is shown in fig. 3.

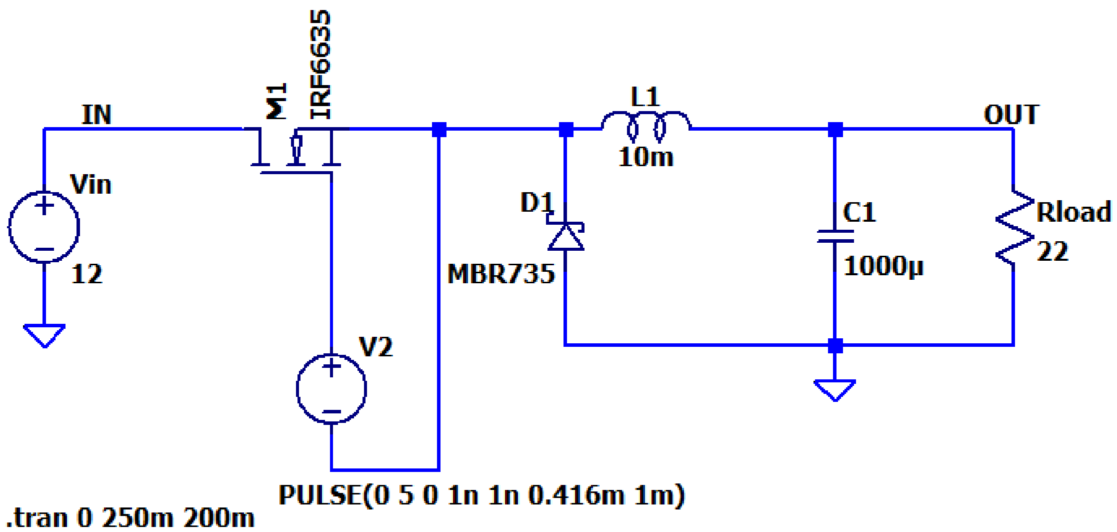


Fig.3. DC/DC buck converter circuit diagram in Ltspice

Firstly, check the output voltage and current of the designed converter from the 0s:

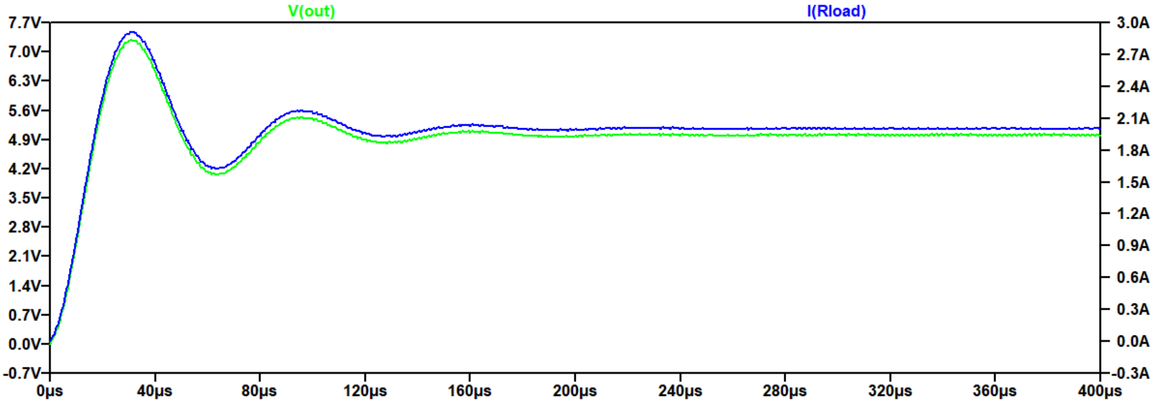


Fig.4. Output voltage and current of buck converter

The output voltage of the buck converter is equals to 5V and the output current is equals to 2A, which we calculated before.

Let's see the inductor current and voltage!

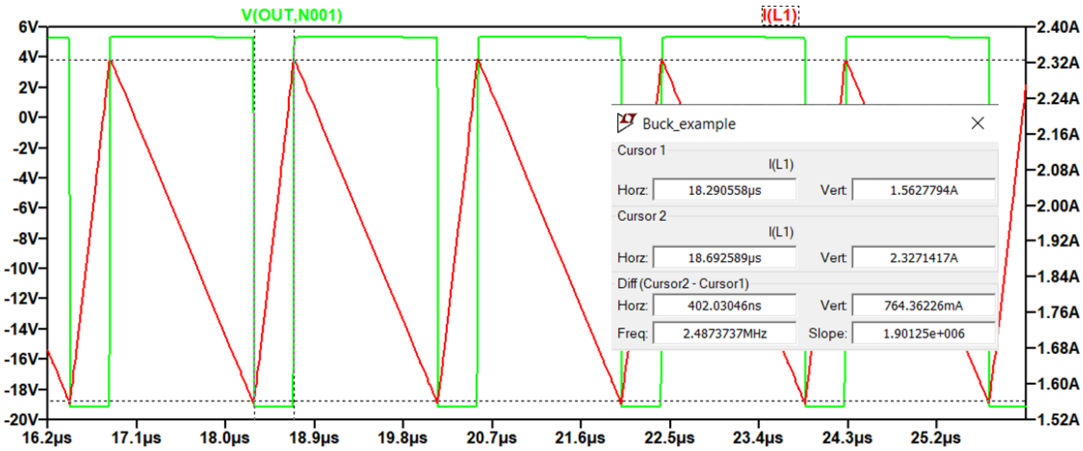


Fig.5. Inductor current of buck converter

The the current ripple of the coil is 764mA (we calculate 800mA), the on time is 0.402us. The voltage at coil change between 5V and -19V. The values are right, because we measure in simulation from the output.

Let's check the output parameters (V_{out}, I_{out}):

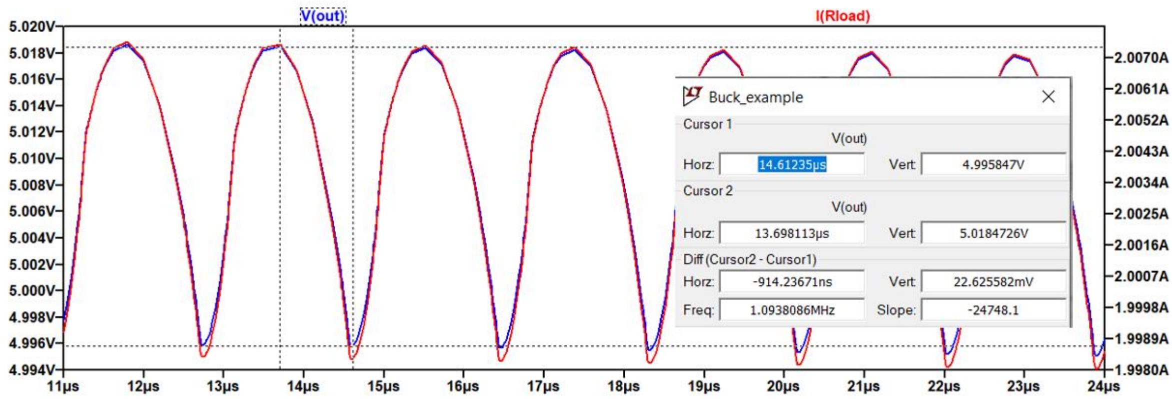


Fig.6. Output current and voltage of buck converter in steady state

The output voltage ripple is 22.6mV, which is below 50mV. It is okay, because we have chosen higher value capacitor.

2. Self-work

The design properties of self work are the following:

Input Parameters	Value
Input voltage	12VDC
Output voltage	5VDC
Output current	2A
Switching frequency	~1kHz
Output voltage change	~1%

2.1 Tinkercad simulation

Before you actually build the designed Buck converter, you need to simulate it using the "Tinkercad" simulation environment. [Tinkercad]is a free and easy to use online application for 3D modelling, (basic) circuit simulation, coding and microcontroller-based application simulations. For this course we assume you are already familiar with Tinkercad from your pervious studies. If not the "Help for the project work" document for the Measurement and data processing course contains some basic introduction. That document will also be helpful in the case of writing the Arduino code for this project.

First you will need to build the circuit in Tinkercad. Use the Power Supply block (Fig. 7) as an input of the converter.

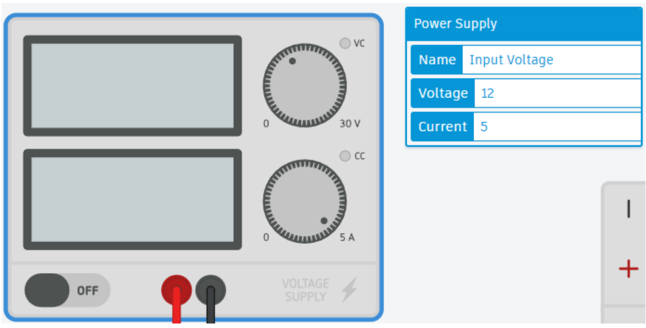


Fig.7. Power Supply block

To make the simulation stable you will need to connect a 50kΩ parallel resistance to the diode, a 100kΩ parallel and a 0.01 Ω series resistance to the inductor and a 100 kΩ parallel and 31.25 mΩ series resistor to the capacitor, as you can see on Fig. 8. If you are having trouble running the simulation it is possible that you need to play around a bit with the value of these component. These additional resistors make the simulation more realistic. In real life there is no perfect inductance and capacitance, all real component have some series resistance, paralell conductance and some parasitic capacitance and inductance. If you place a perfectly inductive or capacitive component in a simulation it can couse oscillations and divergence in the simulation, you can damp these using small series and large parallel resistances. Nonlinear components like diodes (exponential) can couse similar problems. In the case of the capacitor you can set the calculated ESR value with the series resistance.

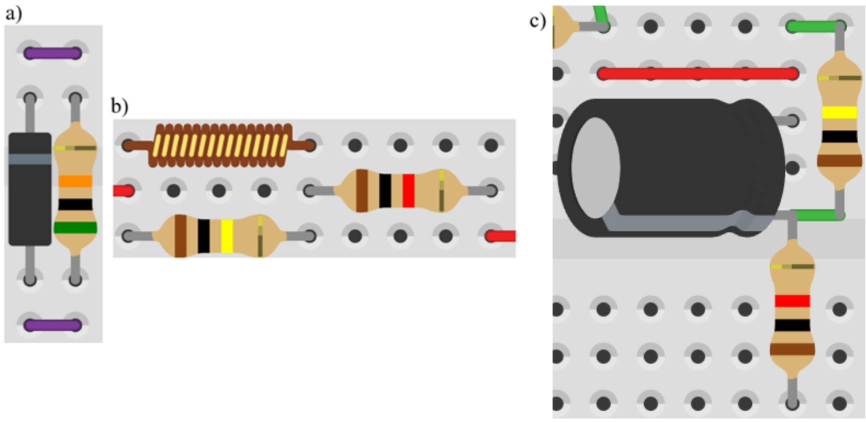


Fig.8. a) Diode model with a parallel resistor b) Inductor model with a series and parallel resistance c) Capacitor model with series and parallel resistance to make the simulation stable and account for the parasitics in real components.

You need to use the Oscilloscope component to measure voltage waveforms, for example the ripple of the output voltage. The voltage scaling of the oscilloscope is automatic, but you will need to set the Time Per Division according to the measured signal. You can measure the waveform of the inductor current ripple on the serial resistance of it. To measure the RMS value of voltage or current you have to use the Multimeter block. For that block you can set either to measure voltage (RMS), current (RMS) or resistance. You can see the Oscilloscope and Multimeter blocks on figure 9.

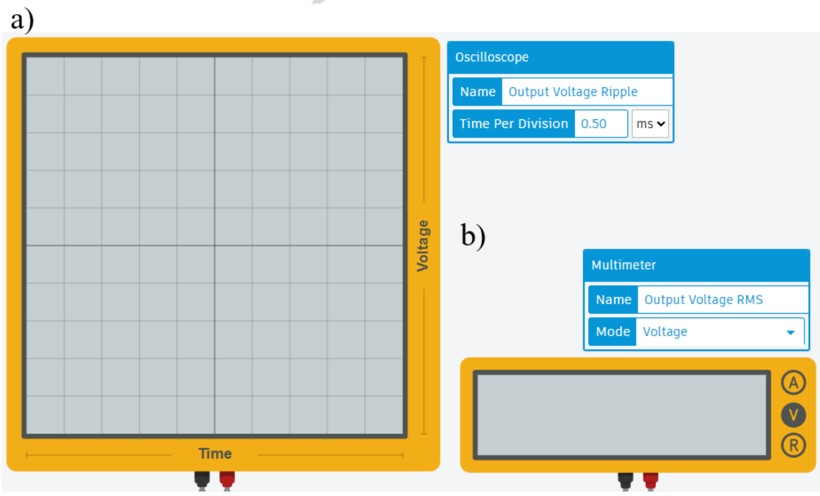


Fig.9. a) Oscilloscpe block b) Multimeter block.

To produce the PWM control signal for the MOSFET you need to use an Arduino Uno. Connect either one of the GND pin to the Source of the MOSFET and connect the generated PWM signal to the Gate of the device. For generating the PWM signal you will need to use the analogWrite() function. This function has two inputs, the first is the pin number on which you want to generate the PWM signal (you can use any one of the PWM compatible digital pins with a "~" sign next to them), and the second is an integer number between 0 and 255 to set the duty cycle of the signal. The Arduino board uses it's timer peripherals to generate the PWM signal. The Uno has 3 timers called timer0, timer1 and timer2. When you use the analogWrite() function by default the output pins connected to timer0 and timer1 generate an approximately 500 Hz PWM signal and the ones connected to timer2 generate about 1 kHz PWM. For the project task you need an approximately 1 kHz signal so use pin 5 or 6. Also by default all of the timers are set to 8 bit, that's why you need to provide an integer value between 0 and 255 for the analogWrite() function. You can read more about Arduino Uno's timers on this site and see the difference between the pins connected to different timers on this Tinkercad example. To adjust the duty cycle, use a potenciometer and the analogRead() function. Send the achieved duty cycle value to the serial monitor. You can find an example in Tinkercad on this link. Be carefull that you need to provide an integer value for the analogWrite() function, so you need to map the 0-1023 input value from the analogWrite(), to 0-255. You can do this with the map() function.

If you have a working model in Tinkercad you can download the arduino code and use it for the real built converter. If you need more help feel free to ask during the labs.

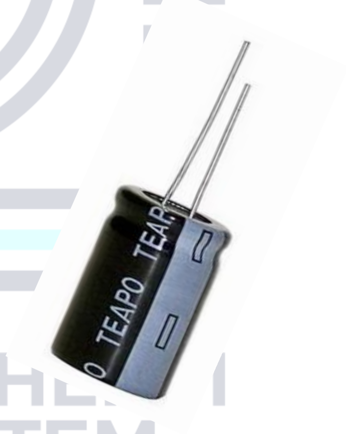
2.1 Building the Buck converter

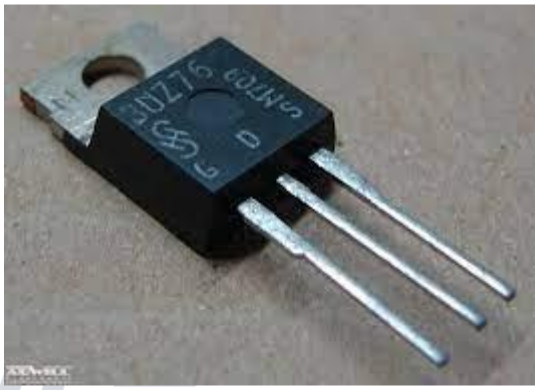
The circuit will be realized on a test panel/breadboard. It is shown in Fig. 10.

Fig.10. Breadboard for circuit

We have the following electronic parts to use:

Arduion UNO	

USB cable for programming	
Inductor	
Capacitor	
Load resistor	

Diode	
MOSFET transistor	
Arduino wire	
Potentiometer	

Using of breadboard will be presented in detail during a lecture.

2.3 Measure the circuit

During the tests, we will measure the *output voltage* and the *duty cycle*. We will use a digital multimeter for the former and an oscilloscope for the latter. See the use of measuring instruments.

Fig.11. Output voltage in case of duty cycle is 0.5

The duty cycle:

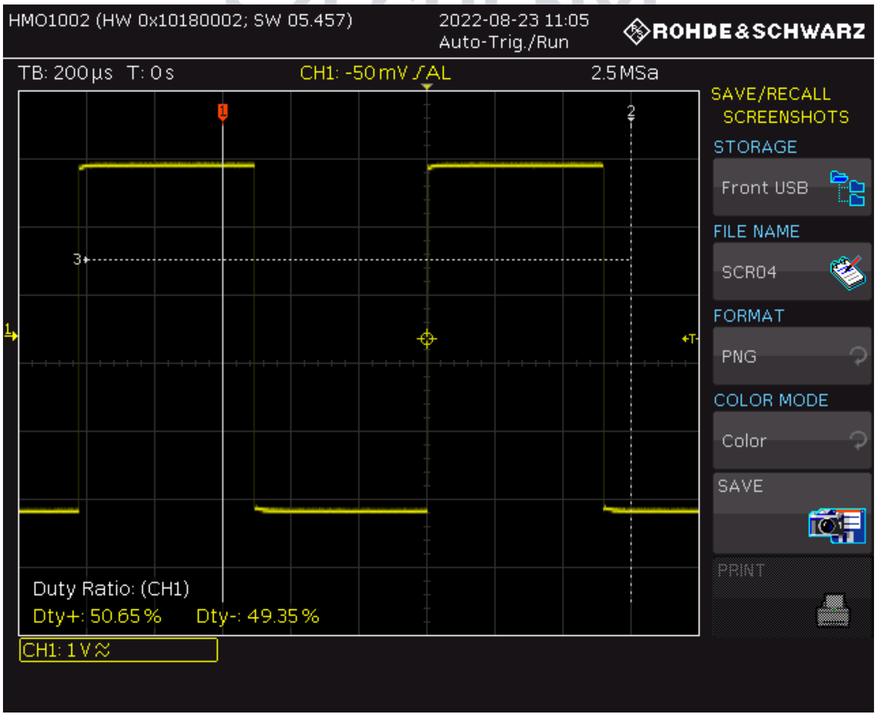


Fig.12. Duty cycle of converter (D=0.5065)

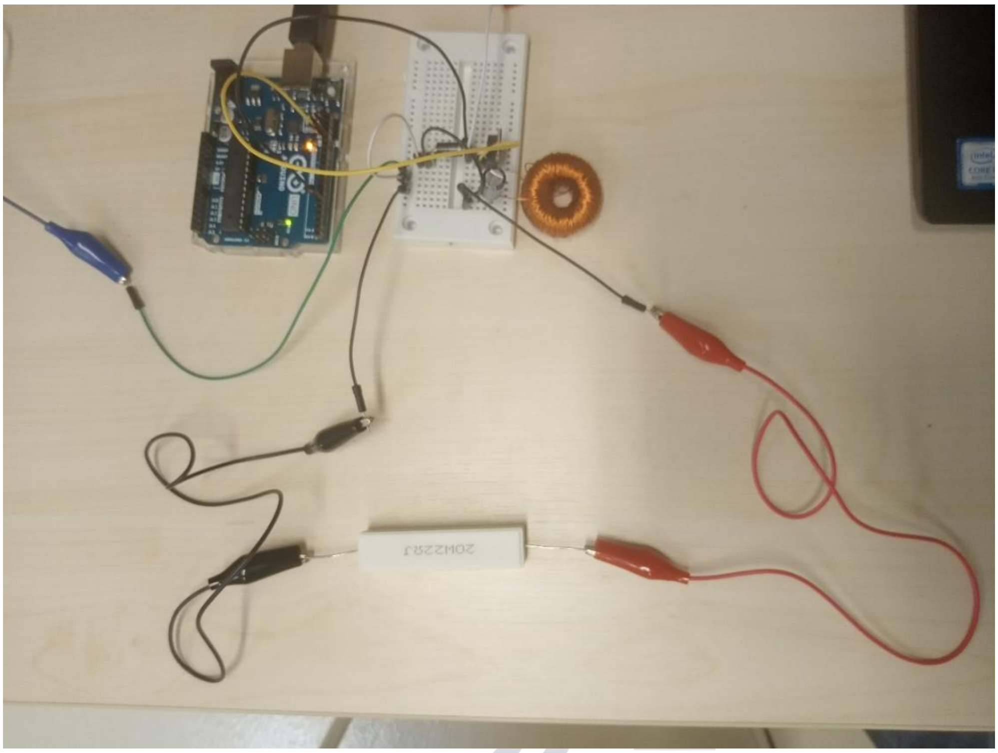


Fig.13. The measurement arrangement

2.4 If the converter doesn't work…

- Check the PWM signal the digital output of Arduino (with scope)!
- Check the output voltage of power supply (it should be 12VDC)!
- Check the ground connections! The ground of Arduino isn’t the same like the circuit ground!
- Check the connection of the electric parts in breadboard.

2.5 Project output (evaluation)

The converter should be working: if we rotate the potentiometer, the output voltage of the converter should be increasing/decreasing.

2.6 Questions (optional)

With the calculated duty cycle, the output voltage will not be accurate. Why?