

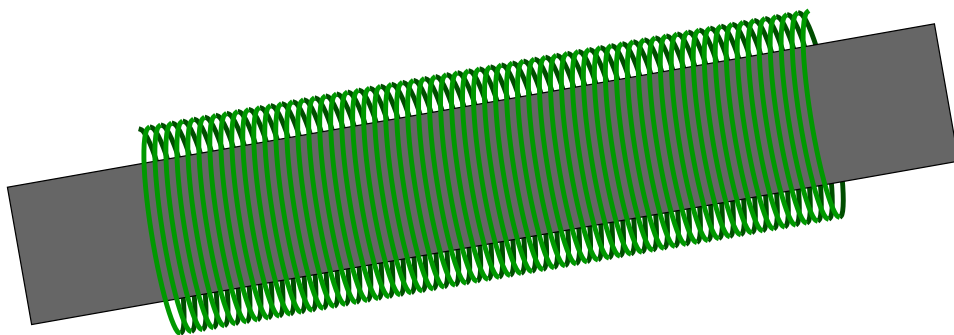
# D1

---

## Boost Converter Design Project: Implementation

---

In the D1 project you will build a power supply that boost 1.5V to up to 12 V. You will study the behaviour of an idealized boost circuit with a simulation. You will use the knowledge gained from the simulation-study to develop a C-program that runs on I1 Matto and controls the output voltage through pulse-width-modulation. You will also interfaces with your embedded program on the I1 Matto by serial communication. The user-interface on the serial terminal will allow for setting a desired output voltage and for displaying the current output voltage. After building your computer-controlled power supply you will tune and evaluate its performance. You may choose to add additional features according to your ability.



## Schedule

---

Preparation Time : See notes *D1: Modelling and Simulation* +  $3 \times 3$  hours

Lab Time :  $4 \times 3$  hours

## Items provided

---

Tools : Hot air guns

Components : 100 mm  $\times$  10 mm Ferrite Rod, STP55NF06L MOS-FET 60V/55A, 1N5819 Schottky Diode 40V/1A,  $680\Omega$  3W 5% Resistor,  $150\Omega$  3W 5% Resistor,  $100\Omega$  3W 5% Resistor,  $0.22\Omega$  3W 5% Resistor,  $2 \times 22k\Omega$  0.25 W,  $2 \times 4.7k\Omega$  0.25 W,  $220\mu\text{F}$  63V Electrolyte Capacitor,  $2 \times$  Tactile Switch,  $\approx 6$  cm of 18 mm 3:1 Heatshrink tubing

Equipment : Multimeter, Oscilloscope

Software : avr-gcc, ASCII text editor

## Items to bring

---

Il Matto

FTDI C232HM Cable

Breadboard

Identity card

Laboratory logbook

Toolkit

Version: January 17, 2020

©2012–2020

Klaus-Peter Zauner

Electronics and Computer Science

University of Southampton

# 1 Introduction

This project builds on the various skills you have acquired in the exercises C1 to C11. You will build a computer controlled power supply based on a boost converter of the type shown in Figure 1.

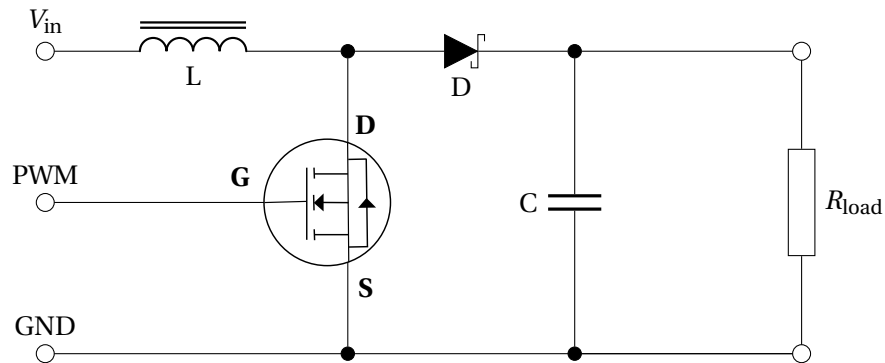


FIGURE 1: Boost circuit schematic with PWM controlled MOS-FET as switch

## 1.1 Outcomes

At the end of the exercise you should be able to:

- Understand the principles of switch-mode voltage conversion
- Design and implement an embedded system that is controlled by a PC<sup>1</sup> through serial communication
- Use a microcontroller to continuously monitor an analogue input signal and respond with appropriate output in real-time
- Implement a closed-loop control-scheme with a microcontroller and evaluate its performance

## 1.2 Assessment

This project is assessed in stages.

During each laboratory session your work completed so far will be marked in your logbook, but there will be no mark for the category *Preparation* on the sticker. On Monday the Preparation (Matlab Simulation) will be marked, on Tuesday the work from Monday will be marked, on Wednesday the work from Tuesday will be marked. On Thursday the overall project will be marked. To give you an indication of the expectations, the sections for Monday to Thursday below indicate typical features for a project at level 3 and at level 4. Marks 0–2 are for partial achievement of level 3 features. Projects with exceptional features or outstanding performance might be judged to reach quality/progress level 5. The demonstrators will provide feedback on your performance each day while marking.

---

<sup>1</sup>Personal Computer

The mark for the simulation (Monday) will contribute 20%, the marks for your implementation (Tuesday to Thursday) will contribute 80% to your final D1 Mark. As stated in the Laboratory hand book, the weighting of the components of each mark (here *Logbook*, *Progress & Quality*, and *Understanding*) is not necessarily equal. The final marks will be moderated for fairness in the context of the overall results of the cohort.

**Project** – *This is a project and not a series of normal laboratory exercises. You will find that the instructions are less prescriptive and you will need to develop your own approach to achieving the objective. You have substantially more freedom in solving the problem—and substantially less support from the demonstrators than in a laboratory exercise. There is no unique correct solution for this project. Set your targets according to your skills and prepare a fallback plan in case you run into difficulties.*

Take a look at the D1 wiki pages—there may be comments and hints for each day and you are invited to contribute to these with suggestions that may be of use to others. In case you make use in your project of suggestions posted by your peers you need to credit them as a source, like any other sources.

## 2 Preparation

### 2.1 First Laboratory Session

The preparation for this exercise is described in a separate document: *D1: Modelling and Simulation*; your logbook will be marked in the first lab session according to this separate document.

You are not required to make additional preparations in your logbook, but you are expected to read all of the present notes before the start of the lab.

For the following three laboratory sessions you should plan your own work to be performed during the session in your logbook according to what is most useful to you—this preparation will not be marked.

## 3 Laboratory Work

### 3.1 Winding a Coil

Wind a coil on your ferrite rod<sup>2</sup>, using Figure 2 as a guide. The coil should be wound tight and all in one direction. Leave at least 5 mm space at the end of the rod and wind at least 60 turns<sup>3</sup>. Leave

<sup>2</sup>Do not drop or knock it, it may break.

<sup>3</sup>For 60 turns you will need 2.35 m wire.

15 cm leads at both ends of the coil. You are provided with heat shrink tubing that can be slipped over the coil ends and heated with hot air to secure the windings in place.

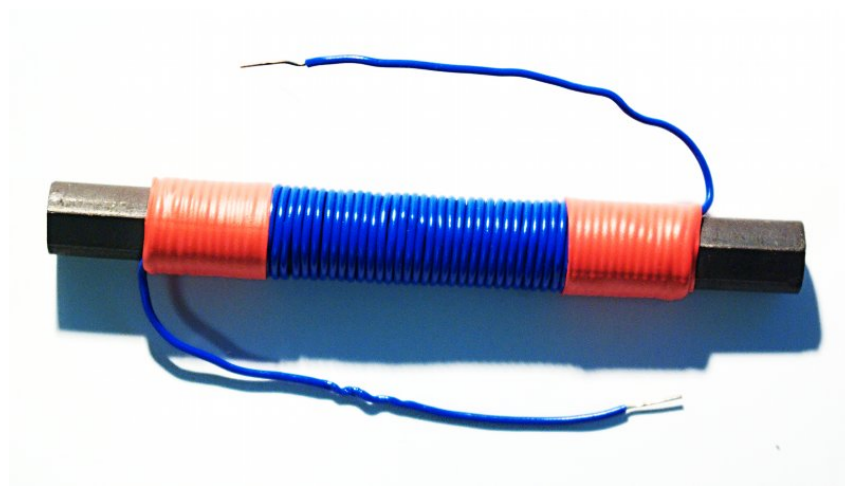


FIGURE 2: Hand-wound coil with 60 turns on 100 mm × 10 mm diameter ferrite rod. Try to avoid damage to the wire insulation (here visible in the foreground) when using the hot-air gun. Leave longer leads than on the sample shown!

If you decide to wind more than one layer: make sure the ends of the coil are thin enough to fit in the heatshrink tube. You can wind the outer layer with a shorter length. Note in your logbook how many turns you have wound and whether you wound more than one layer.

Use the inductance meter in the lab to measure the inductance of your coil and note the value in your logbook.

## 3.2 Circuit Construction

### WARNING:

- Do not supply more than 1.5 V or more than 2 A to the boost circuit.
- Make sure the Resistor  $R_{\text{dummy}}$  is at all times correctly connected parallel to the capacitor C.

*If these precautions are not followed the voltage at the capacitor can exceed the safe level of 40 V. (Note that the IlMatto board will be damaged already when the output reaches 18 V.)*

### 3.2.1 Basic boost circuit

Assemble the circuit shown in Figure 3, but reserve space on the breadboard for the two push button switches that will be required later (see Figure 6). The diode is marked with a ring that indicates the cathode. On the circuit diagram the cathode is the line on the tip of the arrow. Your circuit uses a Schottky diode because it can switch faster and there is less voltage drop across a Schottky diode

than across a regular silicon diode. Please note that *the diode you have has a current limit of 1 A*. If you power the circuit with the bench top supply and short-circuit the output you can damage the Schottky diode<sup>4</sup>.

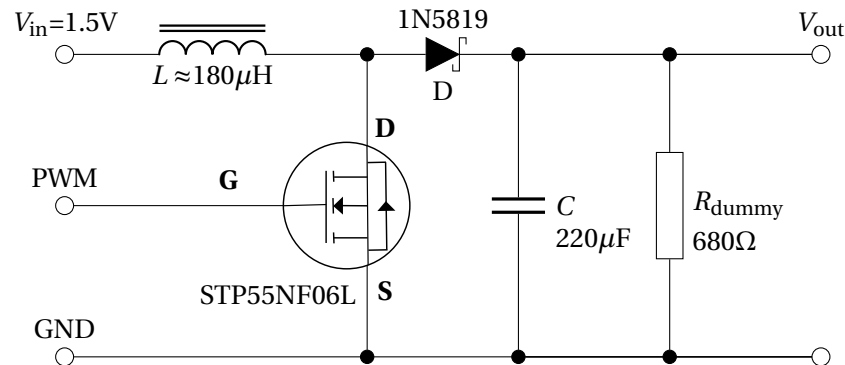


FIGURE 3: Basic circuit for first tests. The terminal PWM is driven by PD7 of the IlMatto board.

The MOS-FET is sensitive to electrostatics: touch a grounded surface (e.g. the metal case of an instrument or PC) before you handle it and then insert it in the bread-board. Do not handle it unnecessarily. The risk of electrostatic damage is much reduced when the MOS-FET is installed in the circuit.

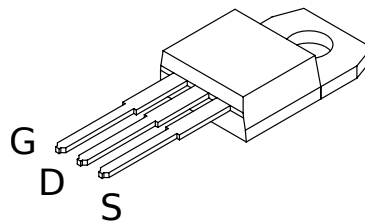


FIGURE 4: Pinout of the STB55NF06L MOS-FET

After you have convinced yourself that your circuit is correctly assembled, the first step is to test whether it can boost the input voltage. Set the bench-top power supply to 1.5 V and current limit to 200 mA. Connect the oscilloscope with a 10× probe  $V_{out}$ . Ground the PWM input (gate of the MOSFET) of your circuit. Connect the bench-top power supply to your circuit.<sup>5</sup> What do you observe at the oscilloscope?

Compile the `embedded_boost.c` program and download it to the IlMatto board. Make sure you have your linker setup with the options for printf with floating point numbers as you did in C9. The compiler needs the options: `-Wl,-u,vfprintf -lprintf_flt -lm`

Connect GND of the boost circuit (see Figure 3) with the ground of the IlMatto board.<sup>6</sup> Connect PWM of the boost circuit with Pin 7 on Port D of the IlMatto board.

Use the oscilloscope with a 10× probe to monitor the output voltage at  $V_{out}$ . Raise the current limit at the bench-top power supply until the output reaches 15V. Note the current limit in your logbook.



<sup>4</sup>Diodes with a higher current limit do not fit in the holes of the breadboard.

<sup>5</sup>Do *not* use the Il Matto board to power your boost circuit!

<sup>6</sup>This is the reference potential for the PWM signal—of course GND of the boost circuit must still be connected to the power supply.

This is *the safe current limit* for your circuit. Do not raise the current limit beyond this except for short tests during the final tuning phase when you are sure your control software is reliably limiting the input current and output voltage. You otherwise risk to damage your IlMatto board (and possibly a computer connected to it) or the semiconductors in your circuit. However, for high-load tests with a working control, *do not forget to increase the current limit*: do not expect more than about 60–80% of the input power at the output.



### 3.2.2 Measuring output voltage

The next step is to use the IlMatto board not only for producing the PWM input to the boost converter, but also to monitor the output voltage delivered at the load.

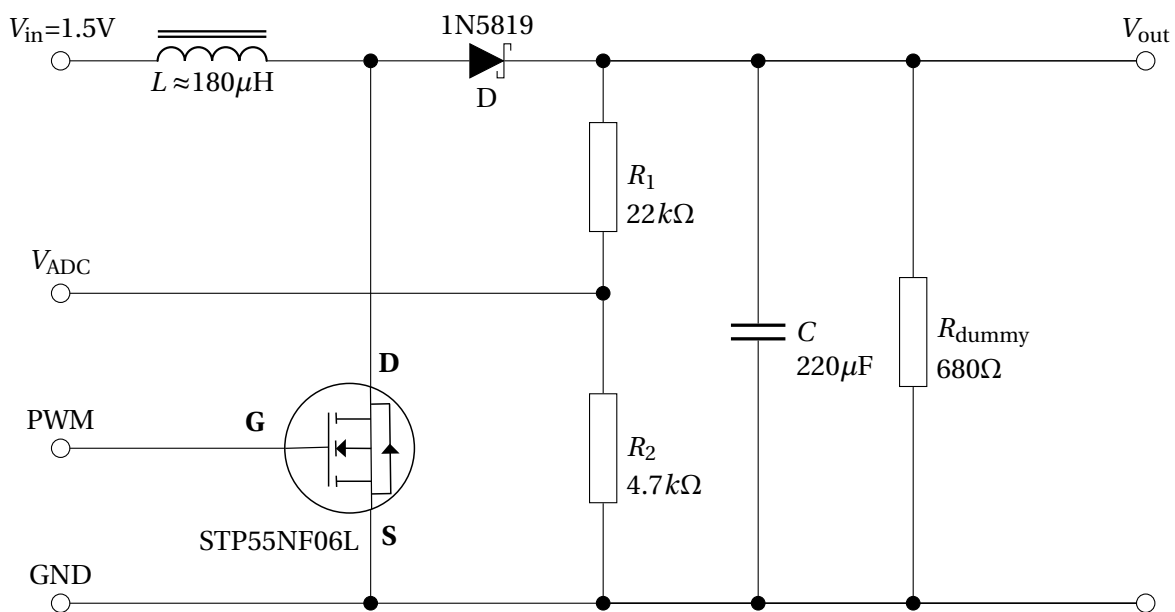


FIGURE 5: Circuit for test with ADC. The PWM terminal is driven by PD7 of the IlMatto board.

Extend your circuit as shown in Figure 5 to include a voltage divider made from  $R_1$  and  $R_2$ .

Connect the oscilloscope at  $V_{ADC}$ . What voltage do you measure? Is this voltage safe for your IlMatto board? Note the voltage in your logbook.



Complete your circuit with the two switchable load resistors  $R_A$  and  $R_B$  as shown in Figure 6. Check which contacts on the switch are connected when it is depressed—your assumptions may be wrong.

You have now completed the circuit you need for this project. The remaining resistors can be used for optional extensions: either to use a second ADC channel to measure the voltage at the input or to measure the current flowing through the coil or the current flowing through the load. For current measurements, insert the  $0.22\ \Omega$  resistor in series with the path in which you want measure the current. Then use the ADC to measure the voltage across this resistor.

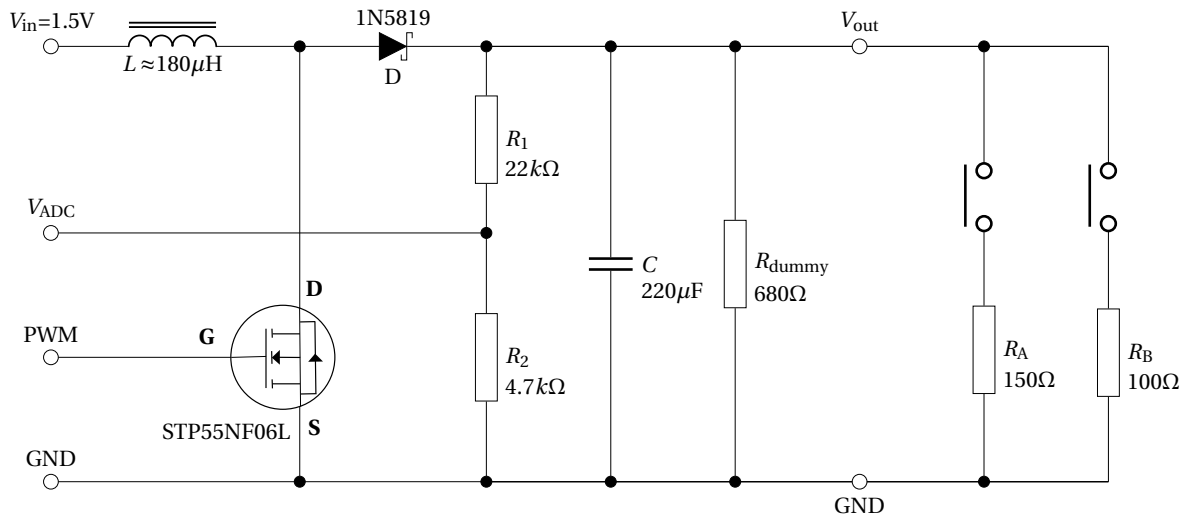


FIGURE 6: Final circuit for tests with varying load. For a high load on the output both load resistors  $R_1$  and  $R_2$  can be switched in parallel.

### 3.3 Circuit Testing

This test should establish whether your circuit works and also provide you with some code that can be a starting point for your software development.

Connect PD0 on the IlMatto board with the USB serial cable signal TXD (orange) and PD1 on the IlMatto board with RXD (yellow) of the USB serial cable. Also connect GND (black).

Open a serial terminal and set the terminal to 9600 baud, one stop bit, no parity, no flow-control. Connect the  $V_{ADC}$  terminal of the boost circuit to the ADC input at PA0 of the IlMatto board. You should now be able to see the voltage measured by the ADC of the IlMatto board as output on the serial terminal. Note the voltage that is shown in your logbook.



## 4 Programming

Make sure you inform the demonstrator each day about:

- ▶ Any part of your work *not* independently performed by yourself and its sources.
- ▶ What your aim was and how you went about implementing it.
- ▶ Any special features you have implemented, or tried to implement.

### 4.1 Control (Tuesday)

From a typical project marked at level 3/4 one would expect:

**Level 4** The embedded system can be set at runtime (e.g., through buttons, rotary encoder, or by communicating through a terminal application) to a desired setpoint in the range from 2–10 V.



Upon any change of the setpoint the system will quickly establish the new output voltage *and* will be able to hold this point well under load change. Error states are indicated (e.g., an LED shows deviation from the setpoint).

**Level 3** The embedded system can be configured at compile time for a desired setpoint in the range between 3–8 V. The system will establish an output voltage near the desired set point. Upon an increase or a decrease in load, the system will adjust the output voltage to be near the desired setpoint again within at most three seconds.

## 4.2 Communication and User Interface (Wednesday)

From a typical project marked at level 3/4 one would expect:

**Level 4** In addition to level 3: Display of a range of information in a clear layout or graphical (e.g. deviation from setpoint over time, efficiency). Furthermore, the user interface has special features, e.g. using the TFT display of the Il Matto, or using a custom host program (instead of a serial terminal) running on the PC.

**Level 3** A serial terminal on a PC can be used to control the embedded program over a bi-directional serial connection. The serial terminal shows the state of the power supply (at minimum the output voltage and the setpoint) and allows the user to change the setpoint.

## 4.3 Integration and Tuning (Thursday)

From a typical project marked at level 3/4 one would expect:

**Level 4** In addition to level 3: Protection against fault situations (e.g., short circuit). User interface allows tuning of controller parameters, possibly with change of control algorithm. User interaction does not degrade controller performance. Behaviour throughout operating range well characterised and documented (oscilloscope images). Good performance of the controller over a wide range (see below).

**Level 3** A convenient user interface to adjust the setpoint and observe the output voltage. Indication of user error and system error (e.g. does not reach setpoint). Reliable control of output voltage under load change can be demonstrated for different set points.

**The goal of the tuning is to achieve a short settling time ( $\Delta t$ ).** This should be the case for both, changes in load and changes of the set point. Ideally this should hold over the whole regulated range of the output voltage (c.f. Figure 7).

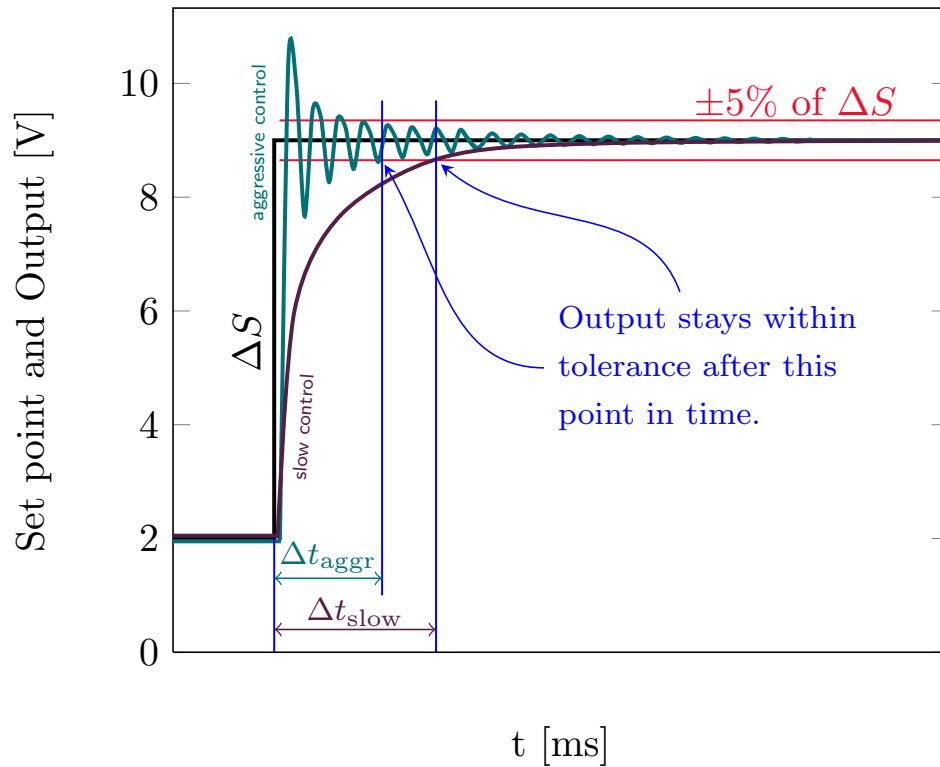


FIGURE 7: Settling time for the boost converter control. After a step-change  $\Delta S$  in the setpoint for the desired output, the control steers the output towards the new setpoint. The response for two controllers is illustrated. The aggressive controller results in overshoot and oscillations. The slow controller does not overshoot and smoothly approaches the new setpoint. The settling time  $\Delta t$  of the controller is the time from the step change in the setpoint to the time from which on the output remains within the tolerance band of  $\pm 5\%$  of the step size  $\Delta S$  around the new set point. In the example illustrated  $\Delta S = 9\text{ V} - 2\text{ V} = 7\text{ V}$  and accordingly the output has to remain within the band  $8.65\text{ V} - 9.35\text{ V}$  ( $9\text{ V} - 5\% \cdot 7\text{ V} = 8.65\text{ V}$  and  $(9\text{ V} + 5\% \cdot 7\text{ V}) = 9.35\text{ V}$ ). What is good controller performance in general depends on the application, however, here we decided to use the settling time as a criteria.