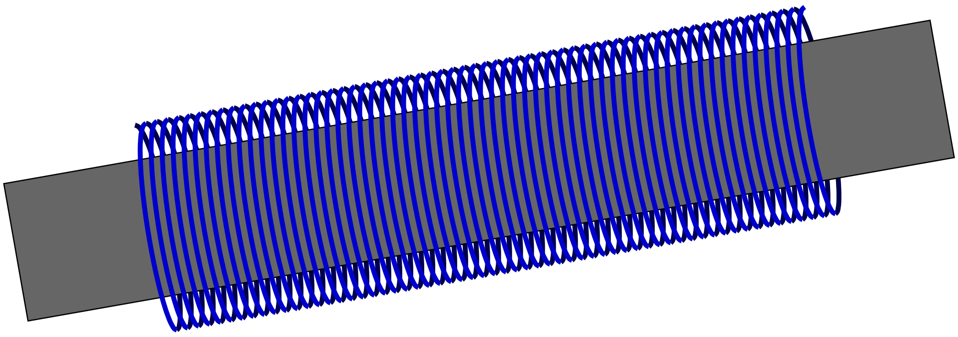
D1

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 `Il Matto` and controls the output voltage through pulse-width-modulation. You will also develop a second C-program that runs on a host PC and interfaces with your embedded program on `Il Matto` by serial communication. The user-interface of the hosted C-program 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.



Schedule

Preparation Time	:	See notes *D1: Modelling and Simulation* + 4×3 hours
Lab Time	:	5×3 hours

Items provided

Tools	:	Hot air guns
Components	:	100 mm × 10 mm Ferrite Rod, STP55NF06L MOS-FET 60V/55A, 1N5819 Schottky Diode 40V/1A, 680Ω 3W 5% Resistor, 150Ω 3W 5% Resistor, 100Ω 3W 5% Resistor, 0.22Ω 3W 5% Resistor, 2 × 22kΩ 0.25 W, 2 × 4.7kΩ 0.25 W, 220μF 63V Electrolyte Capacitor, 2 × Tactile Switch, ≈6 cm of 18 mm 3:1 Heatshrink tubing
Equipment	:	Multimeter, Oscilloscope
Software	:	`avr-gcc, ASCII text editor`

Items to bring

`Il Matto`

Identity card

Laboratory logbook

Toolkit

Version: January 16, 2015

Klaus-Peter Zauner
Electronics and Computer Science
University of Southampton

Before entering the laboratory you should read through this document and complete the preparatory tasks detailed in section 2.

> **Academic Integrity** – *If you wish you may undertake the preparation jointly with other students. If you do so it is important that you acknowledge this fact in your logbook. Similarly, you will probably want to use sources from the internet to help answer some of the questions. Again, record any sources in your logbook.*

You will undertake the exercise working with your laboratory partner. During the exercise you should use your logbook to record your observations, which you can refer to in the future – perhaps to write a formal report on the exercise, or to remind you about the procedures. As such it should be legible, and observations should be clearly referenced to the appropriate part of the exercise. As a guide the ✎ symbol has been used to indicate a mandatory entry in your logbook. However, you should always record additional observations whenever something unexpected occurs, or when you discover something of interest.

For each Task you should create a new directory so that you have a working version of every program at the end of the lab. Remember to place comments in your code.

You will be marked using the standard laboratory marking scheme; at the beginning of the exercise one of the laboratory demonstrators will mark your preparatory work and at the end of the exercise you will be marked on your progress, understanding and logbook.

Notation

This document uses the following conventions:

✎	An entry should be made in your logbook
	Care should be exercised
`command input`	Command to be entered at the command line
Remarkable text	A point of note

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.

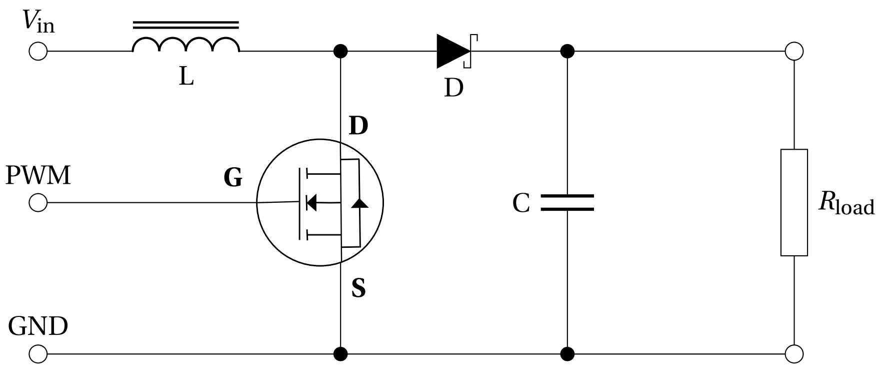


FIGURE 1: Boost circuit schematic with PWM controlled MOS-FET as switch

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[1] 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. For every day there are two requirements:

1. Your logbooks will be marked at the end of each laboratory session, but there will be no mark for the category *Preparation*. To give you an indication of the expectations, the sections for Tuesday to Friday 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.
2. Before 23:59 on the day of the laboratory you are required to submit a brief electronic progress form for the day and—except on Monday—also a an archive (ZIP or tar.gz) of the current state of all the code you have developed. You will find the form for each day and a link to the submission system on the D1 notes page at `https://secure.ecs.soton.ac.uk/notes/ellabs/1/d1/`.

[1] Personal Computer

The mark of each day will contribute with equal weighting (20%) to your final D1 Mark.[2] 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.*

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 four laboratory sessions your 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[3], 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[4]. Leave 10 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.

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.

[2] The mark for Monday is from the modelling and simulation part.
[3] Do not drop or knock it, it may break.
[4] For 60 turns you will need 2.35 m wire.

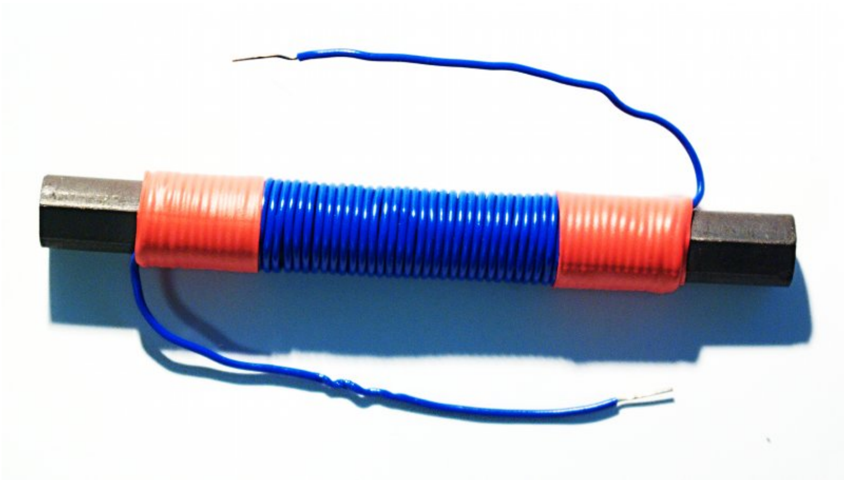


FIGURE 2: Hand-wound coil with 60 turns on 100 mm × 10 mm diameter ferrite rod. Try to avoid damage to the wire insulation (here visible in the foreground) when using the hot-air gun. Leave longer leads than on the sample shown,

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

3.2 Circuit Construction

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[5].

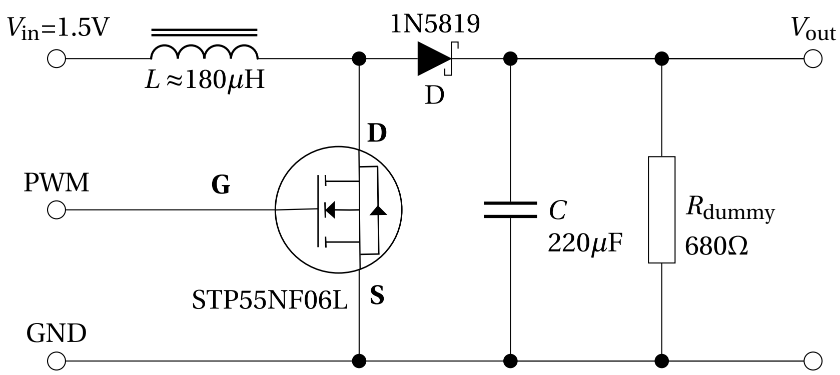


FIGURE 3: Basic circuit for first tests. The terminal PWM is driven by PD7 of the IlMatto board.

[5] Diodes with a higher current limit do not fit in the holes of the breadboard.

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.

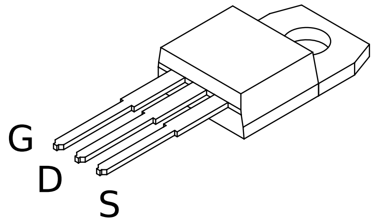


FIGURE 4: Pinout of the STB55NF06L MOS-FET

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 MOS-FET) of your circuit. Connect the bench-top power supply to your circuit[6]. What do your 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. 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.

This is *the safe current limit* for your cicuit. 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.

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.

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

[6] Do *not* use the `Il Matto` board to power your boost circuit!

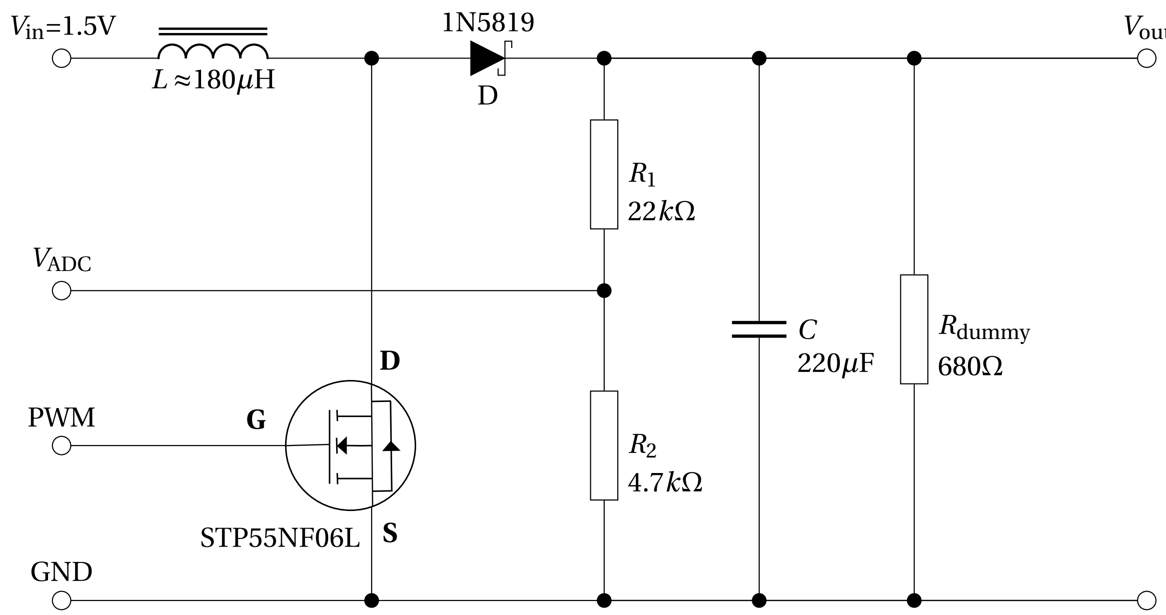


FIGURE 5: Circuit for test with ADC. The PWM terminal is driven by PD7 of the IlMatto board.

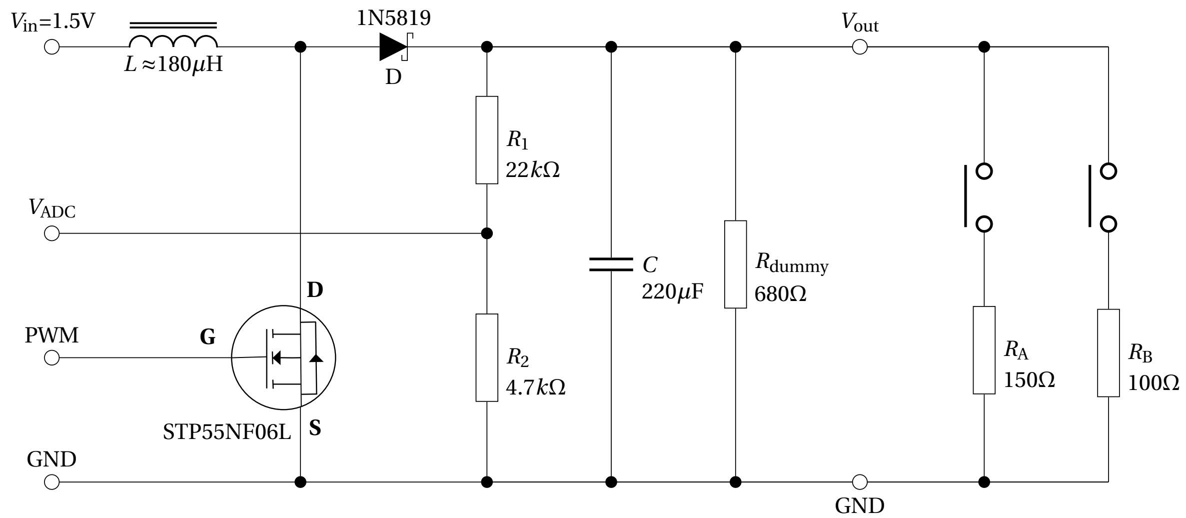


FIGURE 6: Final circuit for tests with varying load. For a high load on the output both load resistors R_1 and R_2 can be switched in parallel.

to measure the current flowing through the coil or the current flowing through the load. For current measurements, insert the 0.22 Ω resistor in series with the path in which you want measure the current. Then use the ADC to measure the voltage across this resistor.

3.3 Circuit Testing

This test should establish weather 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

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 (Wednesday)

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

Level 4 In addition to level 3, the host and embedded system can exchange floating point numbers and do so in such a way that the communication will not block other activities (e.g., the controller on the embedded computer or the indication of the output voltage on the host).

Level 3 A host program has been written that can correspond in both directions with an embedded program. Suitable I/O methods are in place on both the host and the embedded computer to demonstrate the reliability of the communication.

4.3 Integration (Thursday)

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

Level 4 In addition to level 3, extensive support from the user interface to tune the controller: setting of all configuration parameters through interface, easy switching among sets of parameters. Display of a range of information in a clear layout (e.g., deviation from setpoint over time, efficiency). User interaction does not degrade controller performance.

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). Possibly a menu to select between two different control algorithms.

4.4 Tuning (Friday)

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

Level 4 Sophisticated control strategy, behaviour throughout operating range well characterised and documented (e.g, oscilloscope images). Very good performnce of the controller under load change between 1–12 V. Protection against fault sitatuations (e.g., short circuit).

Level 3 A fast and smooth acting controller that can cope well with load change and setpoint change for a range from 3-8 V.

The goal of the tunig is to achieve a short settling time (Δt). 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).

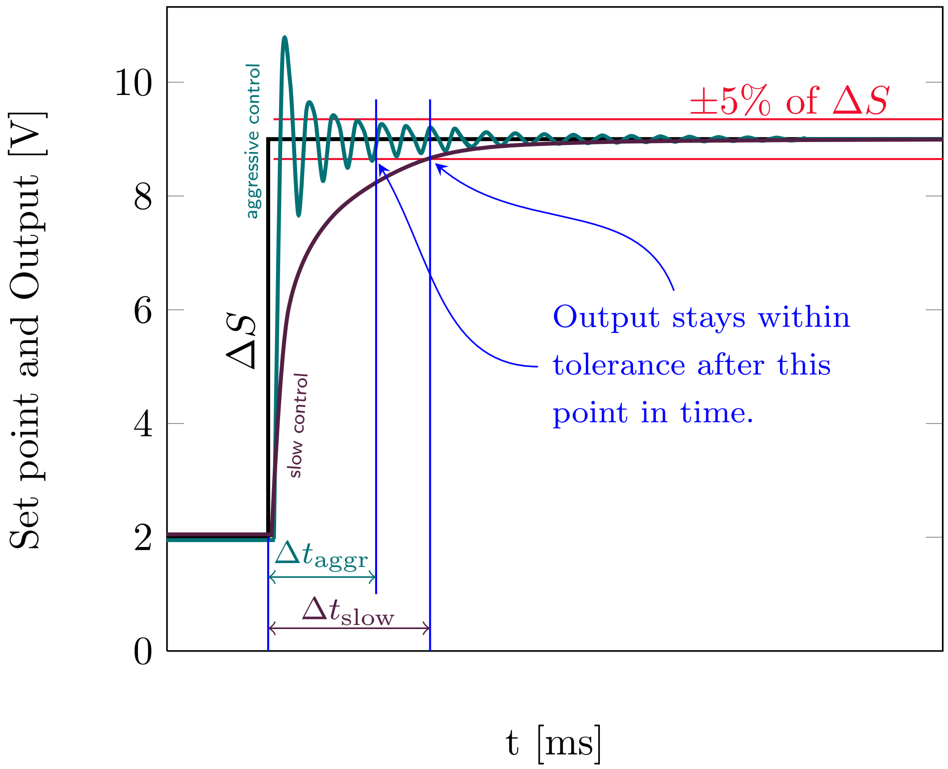


FIGURE 7: Settling time for the boost converter control. After a step-change ΔS in the setpoint for the desired output, the control steers the output towards the new setpoint. The response for two controllers is illustrated. The aggressive controller results in overshoot and oscillations. The slow controller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