LAB 2 – POWER CONVERSION USING ARDUINO

Group Information

Name	Student ID	Name	Student ID

Report Submission

Each group only requires to submit one report. Waveforms, calculations and answers should be laid out neatly and correctly. Marks will be lost for missing and illegible work. Submit your report via UTSOnline. Note the deadline of submission on UTSOnline.

Declaration of Originality

The work contained in this assignment, other than that specifically attributed to another source, is that of the author(s). It is recognised that, should this declaration be found to be false, disciplinary action could be taken and the assignments of all students involved will be given zero marks. In the statement below, I have indicated the extent to which I have collaborated with other students, whom I have named.

Statement of Collaboration and Signatures

Marks

Pre-lab work	
Lab work	
Total marks	

1 Purpose

Digital control of switching power converters has become increasingly important due to the advancements and affordability of micro-controller technology. It facilitates complex mathematic calculation, decision making and intefacing with other IT devices. It is well suited for integration of renewable energy systems (RES) into the electricity network or even standalone RES. The purpose of this lab is to explore digital pulse-width modulation (DPWM) for power converters through simple micro-controller programming. We will use the DPWM to study the step-down (buck) converter in continuous conduction mode (CCM). Successful completion of this lab will build a solid foundation for subsequent labs and group project focusing on maximum power point tracking (MPPT) for PV system.

2 Program an Arduino Nano

2.1 What is Arduino?

Arduino is an inexpensive microcontroller based electronic board. It has easy-to-use hardware and software platforms and both of them are open-source and extensible. It uses Arduino programming language that is based on Java, the reference can be found at https://www.arduino.cc/reference/en/.

2.2 Programming Editors

Arduino IDE is an cross-platform development tool for Aruidno boards. It supports Windows, Mac and Linux operating systems. **Arduino Web Editor** is an alternative programming tool. It allows users to upload their sketches to the cloud and code online. Both the Arduino desktop IDE and the Web Editor can be found from https://www.arduino.cc/en/Main/Software. In this lab manual, **Arduino IDE** is used and explained. A detailed guide has been provided on the website called **Getting Started with Arduino Web Editor on Various Platforms** in helping using the Arduino Web Editor.

2.3 Board, Processor and Port Selection

After installed the Arduino IDE, run the program. The specific Arduino board, processor and port will need to be selected, as shown in Fig. 1.

- Click on the **Tools** tab then the **Board** tabs, select **Arduino Nano** from the drop-down menu and click.
- Click on the **Tools** tab then the **Processor** tabs, select **ATmega328P** from the drop-down menu and click.
- Click on the **Tools** tab then the **Port** tabs, select the availabled **Com** from the drop-down menu and click. Note that this step will be completed when the Arduino board is connected.

2.4 Programming Arduino Nano

Arduino Nano is a compact, breadboard-friendly board. The top and bottom view of the board are shown in Figs. 2 and 3 respectively. More details can be found from https://store.arduino.cc/usa/arduino-nano. The Nano has 6 PWM output pins, which are pin 3, 5, 6, 9, 10 and 11. They provide 8-bit PWM output capability using the

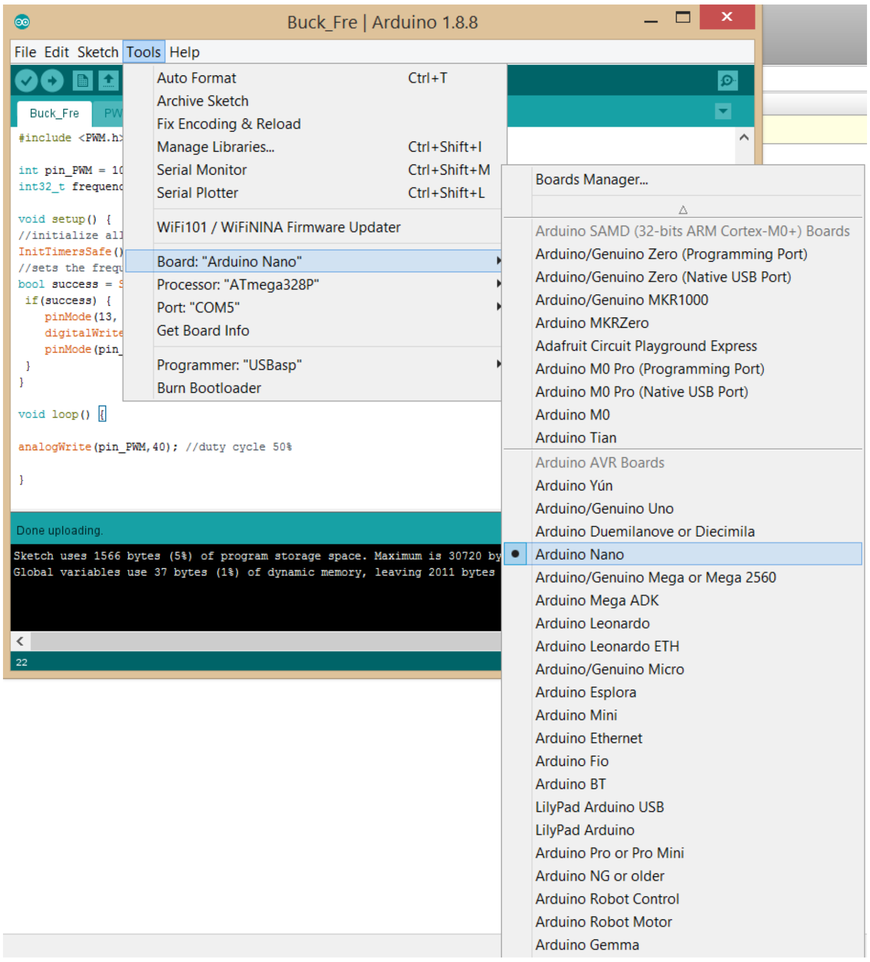


Figure 1: Arduino IDE, choose Arduino board, processor, and port.

command called **analogWrite**. They have different default frequencies and are controlled by the timers 0, 1 and 2. The frequencies cannot be changed by users if using one simple function/command provided by the Arduino programmming language. An approach to solve this problem is to include a PWM library.

The PWM library facilitates users to change the PWM frequencies as desired. In the meantime the **analogWrite** function is no longer defaulted by the 8-bit resolution but it becomes a period dependent value. The period of frequency and analog duty cycle can be determined by equations (1) and (2). The system clock frequency f_{clk} of the ATmega328P is 16MHz, the prescaler N is 1, and d is the duty cycle. A sample code can be found in the Laboratory Exercises folder, and the code is shown in Fig. 4. A 100kHz frequency is set for Pin 10. And based on equation (1), you should be able to calculate the period counter (OCR1x where x = A is for Pin 9 and x = B is for Pin 10) of the frequency. The duty cycle value for **analogWrite** can be calculated based on equation (2).

Notice: In this lab, you will only need to change the frequency value in line **int32 _ t frequency = [value];** and the analogWrite duty cycle value in line **analogWrite(pin _ PWM, [value])**. You can use either Pin 9 or Pin 10, as in the sample code, we have only set timer 1 (i.e. OCR1A for Pin 9, OCR1B for Pin 10) which controls Pins 9 and 10 PWM outputs; they share the same frequency. Otherwise you will need to set timer 0 or timer 2 for other PWM output pins.

$$\text{Frequency (Hz)} = \frac{f_{clk}}{2 \times N \times (1 + \text{OCR1x})} \tag{1}$$

$$\textbf{AnalogWrite}\ \text{Duty Cycle} = (1 + \text{OCR1x}) \times d \tag{2}$$

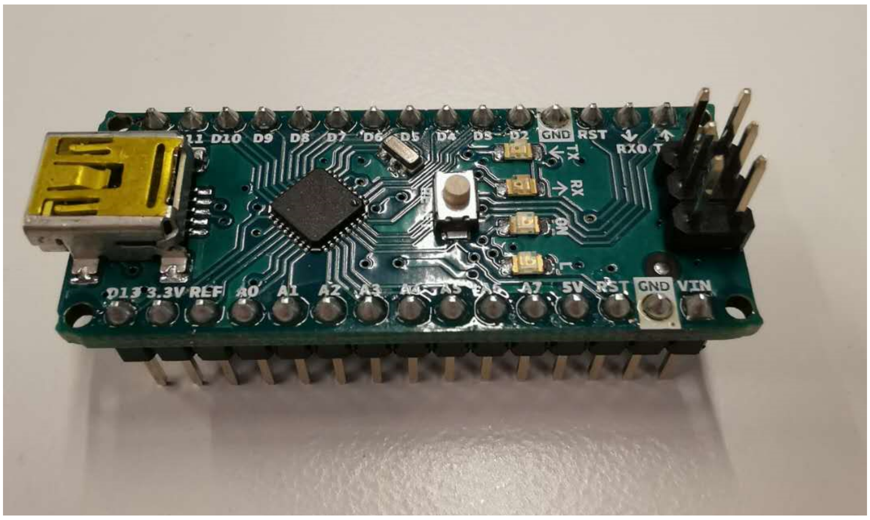


Figure 2: Arduino Nano top view.

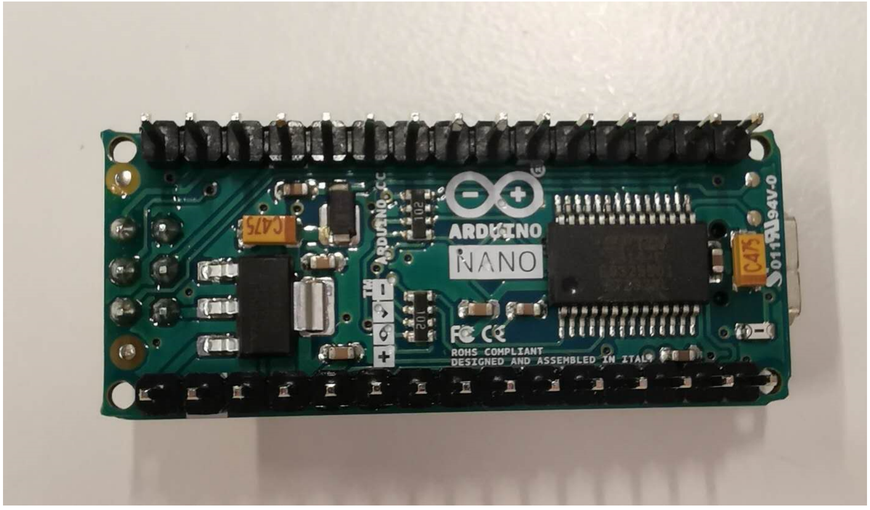


Figure 3: Arduino Nano bottom view.

2.5 Uploading code to Arduino

Once you have made the connections for Arduino and laptop. On the Arduino IDE, click on **Verify** to check if there are syntax errors in your code. Once verified, click on **Upload** to upload the code to the Arduino chip. Once it is finished, a "Done uploading" will appear, as shown in Fig. 5 to indicate the code is successfully written on the chip.

3 Pre-lab Work

i) Complete Sections 2.2, 2.3 and 2.4 [Individual or Group work].

ii) Construct a synchronous buck converter (i.e. the free-wheeling diode is replaced by a MOSFET. Another gate pulse is added and connected to this MOSFET. Note the timing of this gate pulse as compared with that of the other MOSFET.) in LTSpice and simulate the converter using the following parameters and values in Table 1.

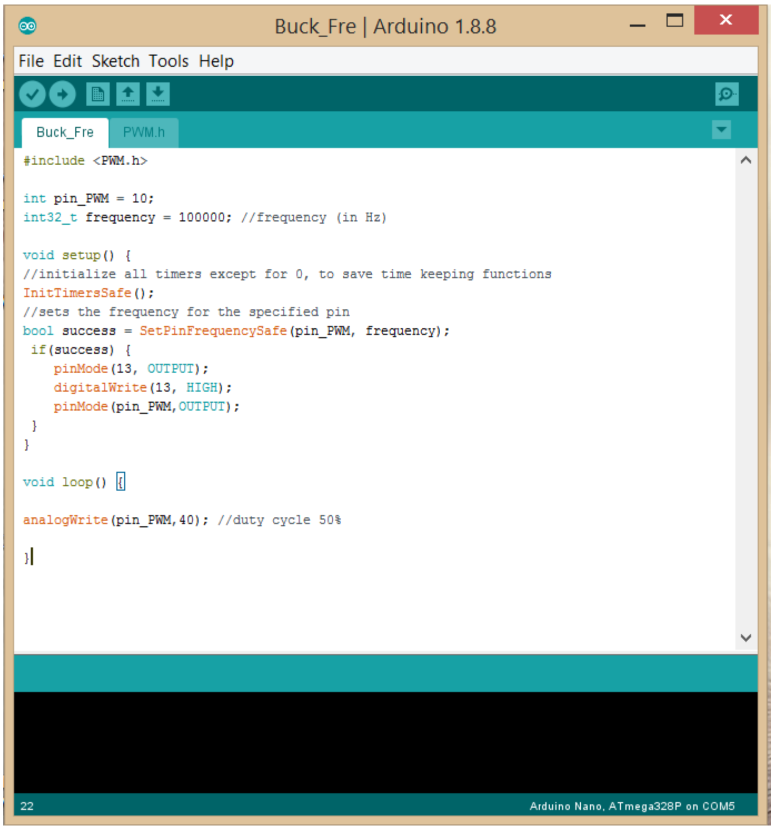


Figure 4: Arduino sample code to generate a PWM signal via pin 10 with 100kHz frequency and 50% duty cycle.

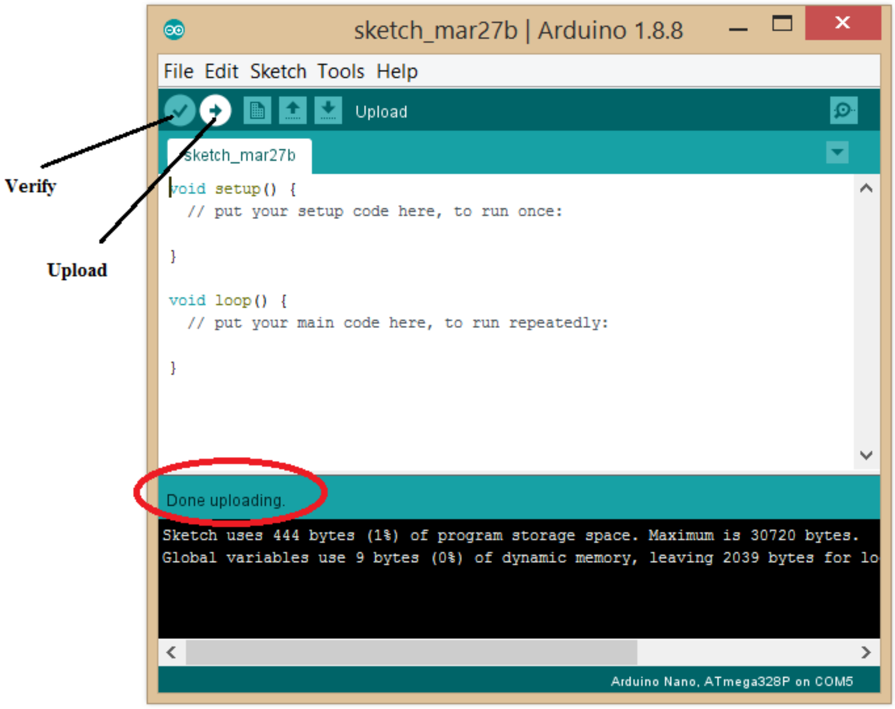


Figure 5: Code is successfully downloaded to the Arduino chip.

Parameters	Values
Input voltage V_{in}	24V
Inductor L	330μH
Output capacitor C	100µF
Duty cycle	50%
Switching frequency	100kHz
Output resistor R_L	33Ω

Table 1: Circuit and operation parameters for a buck converter.

iii) Show the circuit and simulation results (i.e. input voltage, input current, inductor current, mid-point voltage between upper and lower MOSFETs and output voltage)

4 Equipment

- 1 × Assembled testing board
- 1 × Breadboard
- 1 × Laptop computer with Arduino IDE installed or Web Editor Account created (each group to prepare)
- 1 × Mini-B USB download cable
- 1 × Arduino Nano board
- 1 × Digital Storage Oscilloscope (DSO)
- 2 × 10:1 Oscilloscope Voltage Probe
- 1 × Two channel Power Supply up to 30V 1A (set to 24V)
- 1 × Current Probe with amplifier
- 1 × Selectable resister bank
- 1 × Capacitor (100uF/50V)
- 1 × Digital Multimeter

5 Experiments

In this lab exercise, Sections 5.1 focus on programing the Arduino and generating the digital PWM signal. With the PWM generation ready, you will use this signal to study the synchronous buck converter in Sections 5.2 to 5.5.

5.1 Digital PWM Generation

a) Connect the USB cable between the computer and the Arduino Nano board. Run Arduino IDE or equivalent editor program. (**Warning: DO NOT connect the benchtop power supply 24V to the assembled testing board at this stage yet.**)

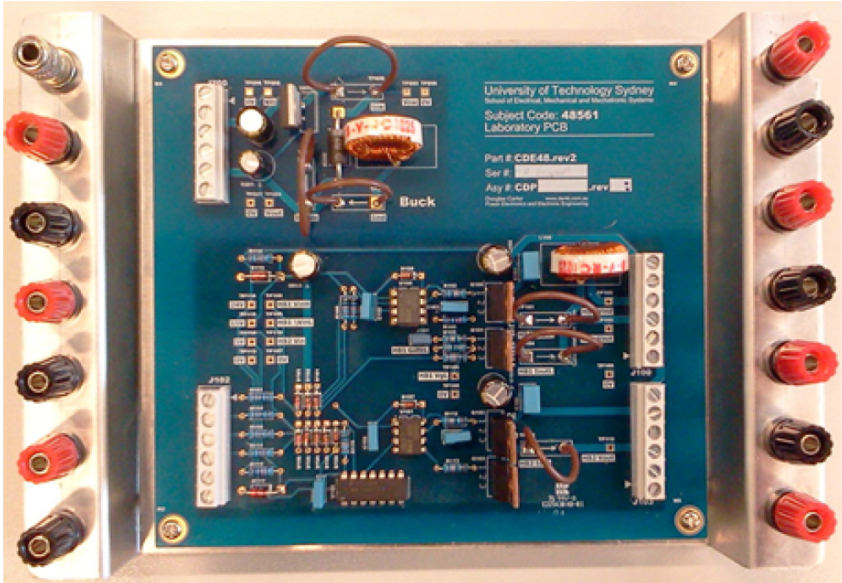


Figure 6: Assembted testing board.

b) Upload an empty file to the Arduino first to make sure the communication between Arduino and the laptop is proper.

c) Download the **PWM.h** file available in UTSOnline subject laboratory exercises folder into the same place where you saved your current program.

d) Click on the **Sketch** tab then the **Include Library** tabs, select **Add .ZIP Library** from the drop-down menu and click.

e) A dialog box will pop up. On the File of type field, select **All Files** then select the PWM.h file to be included in the programming editor.

f) Copy the sample code, and program the Arduino to produce PWM pulses at 100kHz and 50% duty cycle according to your pre-lab work in Section 2.

g) Observe and record the PWM signal from the PWM pin (D10) of the Arduino board using the DSO.

5.2 Setting up the step-down synchronous buck converter

a) Set one of the outputs of the power supply to 24V **without connecting to any equipment**.

b) Turn off the power supply.

c) Check if the jumper J101 of the assembled testing board is in place.

d) Connect the 100µF capacitor and the resistor bank (select 33Ω) to the output (HB1 Vind) and ground terminals of the assembled testing board. Refer to Appendix A for details. Use a breadboard for this step if necessary.

e) Connect the ground pin of the probe to the ground of the testing board. Refer to Appendix A for details.

f) Connect PWM and 0V from the breadboard to pins 1 and 2 (positive, i.e. shorting pins 1 and 2) and pin 6 (negative) on J102 respectively of the assembled testing board.

g) Connect the power supply 24V (power supply is off) via a digital meter (for current measurement) and 0V pins/sockets to the board. Refer to Appendix B for details.

5.3 First test of the buck converter

a) After checking the connections (please ask tutor to double check), turn on the power supply.

b) Observe the PWM signal (using as main triggering signal on DSO).

c) Use other channels to observe the mid-point (TP102), inductor current, and output voltage. Record the waveforms of all four channels including the PWM signal.

d) Record the input voltage, input current (digital meter), output current (average inductor current in DSO) and output voltage (DSO).

5.4 Effect of switching frequency on the converter

a) Turn off the power supply.

b) Change the code to produce 50% duty cycle at 50kHz.

c) Observe the PWM signal on the DSO to confirm the switching frequency and duty cycle.

d) Turn on the power supply.

e) Observe and record the PWM signal, voltage at TP102, inductor current and output voltage.

f) Record the input voltage, input current (digital meter), output current (average inductor current in DSO) and output voltage (DSO).

g) Repeat steps a) to f) but using 75kHz for PWM generation.

5.5 Effect of duty cycle on the converter

a) Turn off the power supply.

b) Change the code to produce 25% duty cycle at 100kHz.

c) Observe the PWM signal on the DSO to confirm the switching frequency and duty cycle.

d) Turn on the power supply.

e) Observe and record the PWM signal, voltage at TP102, inductor current and output voltage.

f) Record the input voltage, input current (digital meter), output current (average inductor current in DSO) and output voltage (DSO).

g) Repeat steps a) to f) but using 75% duty cycle at 100kHz for PWM generation.

6 Analysis

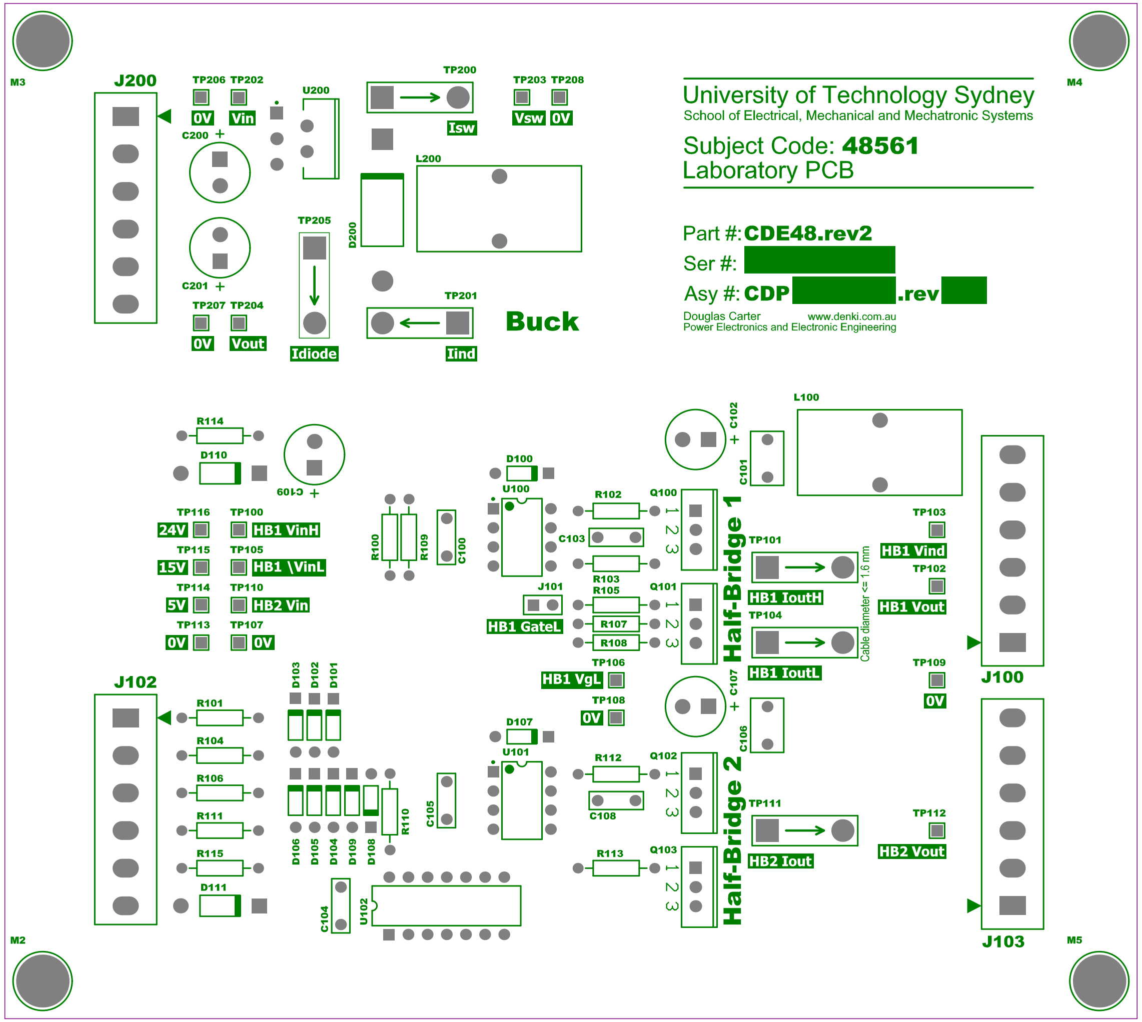
i. Compare and explain the difference of the inductor current and output voltage between the experiment and the simulation in pre-lab at 100kHz and 50% duty cycle.

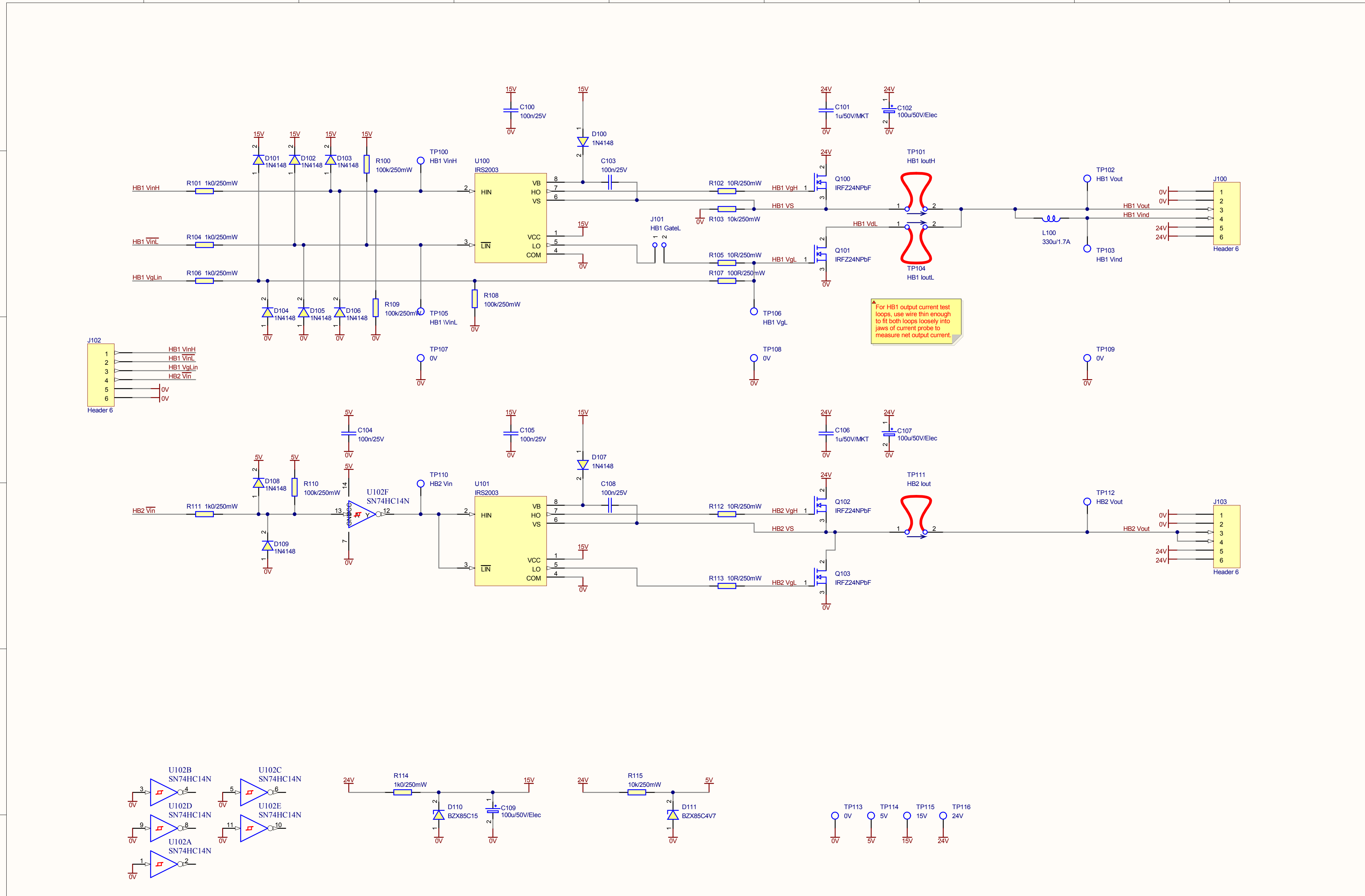
ii. Investigate the effects of the duty cycle and switching frequency on the converter performance, e.g. voltage conversion ratio, inductor ripple current amplitude, efficiency, etc.

Appendix A

(1) Top view of the Assembled Testing Board PCB

(2) Schematic diagram of the Assembled testing Board


University of Technology Sydney
School of Electrical, Mechanical and Mechatronic Systems
Subject Code: 48561
Laboratory PCB
Part #: CDE48.rev2
Ser #:
Asy #: CDP .rev
Douglas Carter
www.denki.com.au
Power Electronics and Electronic Engineering
Buck
Half-Bridge 1
Half-Bridge 2
Cable diameter <= 1.6 mm
J200
J100
J102
J103
M2
M3
M4
M5
U200
L200
D200
C200
C201
TP200
TP201
TP202
TP203
TP204
TP205
TP206
TP207
TP208
0V
Vin
Vout
Isw
Vsw
Iind
Idiode
L100
U100
U101
U102
Q100
Q101
Q102
Q103
D100
D101
D102
D103
D104
D105
D106
D107
D108
D109
D110
D111
J101
C100
C101
C102
C103
C104
C105
C106
C107
C108
C109
R100
R101
R102
R103
R104
R105
R106
R107
R108
R109
R110
R111
R112
R113
R114
R115
TP100
TP101
TP102
TP103
TP104
TP105
TP106
TP107
TP108
TP109
TP110
TP111
TP112
TP113
TP114
TP115
TP116
24V
15V
5V
HB1 VinH
HB1 \VinL
HB2 Vin
HB1 GateL
HB1 VgL
HB1 IoutH
HB1 IoutL
HB1 Vind
HB1 Vout
HB2 Iout
HB2 Vout



All resistors 1%/250mW unless noted otherwise.
All capacitors 10%/50V unless noted otherwise.

Title: Power Electronics Experimental Board University of Sydney, Technology			
Document No.: DE45	Date: 14 Apr 2011	Drawn By: DRHC	
Rev: 2	Size: A3	Sheet No.: 1 / 3	Checked By: DGD	Approved By: DRHC

Douglas Carter
BA PhD Camb, MIEEE
Power Electronics and Electronic Engineering
1/17 Albert Street Edgecliff NSW 2027 Australia
P +61 413 010 838 E douglas.carter@denki.com.au

File: G:\ArtShokai\Projects\PR15 UTS, Labs\Design1 Capture\Electronic\DE45 PE experiment\BridgeConverters.SchDoc