LAB TEAM PROJECT REPORT

TUE Group 3

Patrick Gaft, SID: 520432082

Alvin Chen, SID: 500584882

Nicholas Latrou, SID: 520450798

Yancheng Lin, SID: 510022956

University of Sydney
ELEC1004

Contents

Summary/Abstract

The lab team project report is a detailed outline of the work done throughout Labs 3-6 which focuses on the basis of a boost converter and its implementation into a solar energy system.

Introduction

The project team consists of 4 individuals working through each lab and understanding the fundamentals of simple DC-to-DC boost converters and solar system panels. The project report is split into four main headings with information regarding the specificities of the boost converter and its control, the solar energy system, PV controller design and overall team compatibility. This project will present our understanding of using a controller to manage the varying output from a photovoltaic solar panel via breaking down the concept into the mentioned headings.

1 Boost Converter Design and Control

As stated in Lab 3, a boost converter is a DC-to-DC power converter which steps up voltage whilst stepping down current from its input supply to its output load. The boost converter utilises a technique called Pulse Width Modulation (PWM) to control the outputted voltage during the switching, creating a square wave on-off pattern. PWM allows for digital control of the boost converter via controlling the pulse width of the circuit, otherwise known as its duty cycle. Pulse width is the ratioed time during which current is flowing (on) through the circuit compared to when it is not (off). In terms of its effect on the boost converter, the use of a potentiometer will control the PWM element of the circuit which in turn will determine the output voltage based on the duty cycle percentage set for the PWM. Control of the PWM via Arduino code was based off the given templates in Lab 6 as seen below. The operation of the boost converter will be described during its states below (On-Off States).

The **main components** of a boost converter consists of:

- An inductor (Stores current)
- Any number of capacitors (Stores voltage)
- A diode
- A transistor (MOSFET)
- Input supply
- Transistor Power Supply

The observed voltage boost happens instantaneously in reality when utilising a boost converter within a lab setting, but the process can be slowed down tremendously (down to microseconds ~ 100μs) to notice a step-up effect of the voltage. The component which forms the basis of the boost converter is the switch, or MOSFET transistor in this case. Thus allowing for the transitioning between the two states of a boost converter: On and Off states. Additional capacitors can be noticed in boost converter designs as they are utilised as filters to reduce voltage ripples which is the periodic variation of the DC voltage as seen in *Figure 1*.

On State:

As seen in the *Figure 2* circuit diagram, a current is supplied through the input voltage supply which runs through the inductor. When a current passes through an inductor it generates a magnetic field (via Faraday's Law) around the component. The inductor stores the electrical energy (current) in the form of magnetic energy (magnetic field). Theoretically, the inductor will want to return to its natural state by removing its magnetic field when there is no other current flowing through it. During the on period of the switch (transistor), the current is able to flow through the inductor constantly and back into the voltage supply as usual. The current will not flow into the second half of the circuit containing the diode and capacitor, only following the designated path with the switch.

Off State:

During the off state of the switch, the main observation is that current begins to flow through the rest of the circuit not seen during its on state. The current will not flow through the switch. The magnetic field generated by the inductor during the on-period will collapse quickly during this time, and the current within the inductor will dwindle down to 0 amperes (triangle wave graph). Again Faradays Law states a changing magnetic field will result in a changing electric field, hence the voltage spike during this moment. The induced current flow from this point onwards will be from the collapsing magnetic field and not the input voltage supply. Current flows through the diode and capacitor components of the circuitry, finally the voltage spike being stored within the capacitor. The diode acts as a roadblock which halts any reverse current flow from the capacitor through the switch in its on-state and into the power supply.

This process is repeated quickly throughout the instances of on-off periods within the boost converter. With every voltage spike from the collapsing magnetic field of the inductor being stored within the capacitor which ultimately steps up the voltage by an incremental amount. Furthermore, in a real-life setting, the switch is usually in a rapid on and off state of transition, thus the inductor's magnetic field will never collapse fully so there is a build-up effect of voltage on the inductor. This can be seen in *Figure 3*.

The Arduino code in *Figure 4* dictates the PWM control of the boost converter which basically dictates the duty cycle. An external microcontroller in the form of a potentiometer can be used in conjunction with the above code to then change the duty cycle percentages. This will then directly affect the output voltage of the boost converter.

2 Solar Energy System and Panel Performance

The Solar Photovoltaic (PV) system our group used in Lab 6 consisted of the following parts:

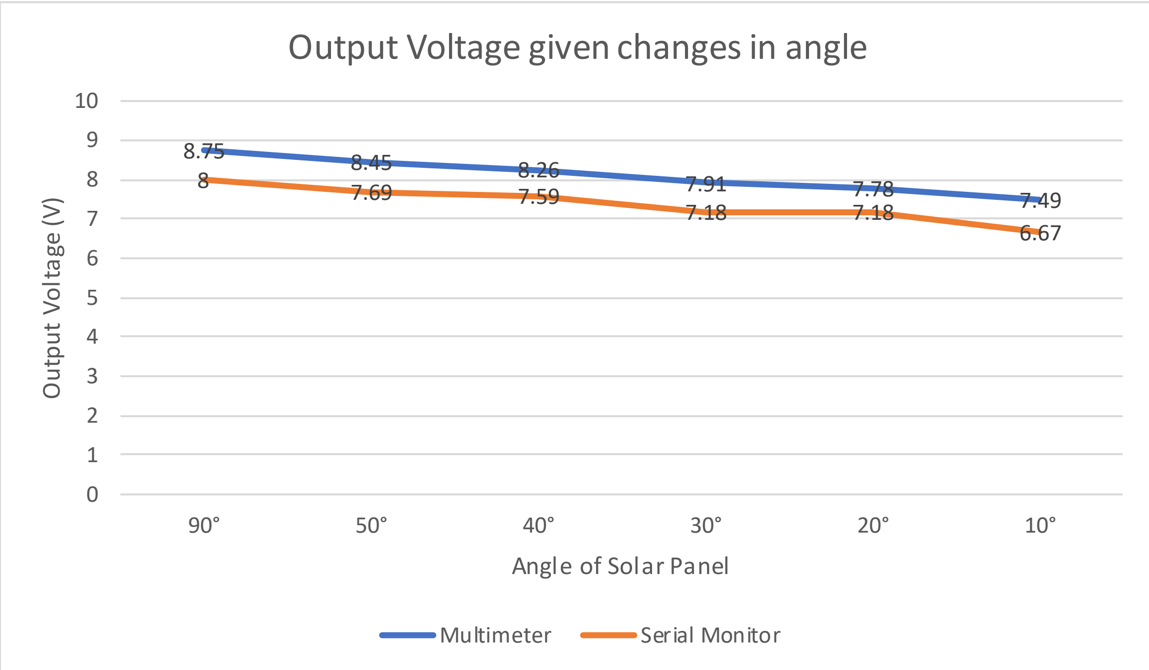
- PV Panel
- DC/DC boost converter
- Arduino MCU
- Personal computer – to access Arduino code
- Heat lamp – source of solar energy
- 2 Voltage sensors: Digital Multimeter and Serial Monitor

The boost converter is a power converter that steps up voltage and steps down current from the input – supply – to the output – load. It utilises Pulse Width Modulation (PWM) to switch the duty cycle of a transistor and vary the output voltage. The Arduino MCU is used in tandem with this device to sense the input & output voltages and generate a PWM signal, allowing the boost converter to maintain a steady output voltage.

The need for a boost converter in a PV energy system stems from the output voltage of these systems typically being low and unreliable, being affected by many conditions such as cloud cover, angle of the sun relative to the panel, temperature, and more. A boost converter steps up the varying low input voltage to a more reliable and stable level.

We recorded how the PV system performance was affected by the variables: angle, distance, and coverage, based on the output voltage (Recorded by both the digital multimeter and serial monitor) and also the changes in observed brightness, ranging from no light to extremely bright.

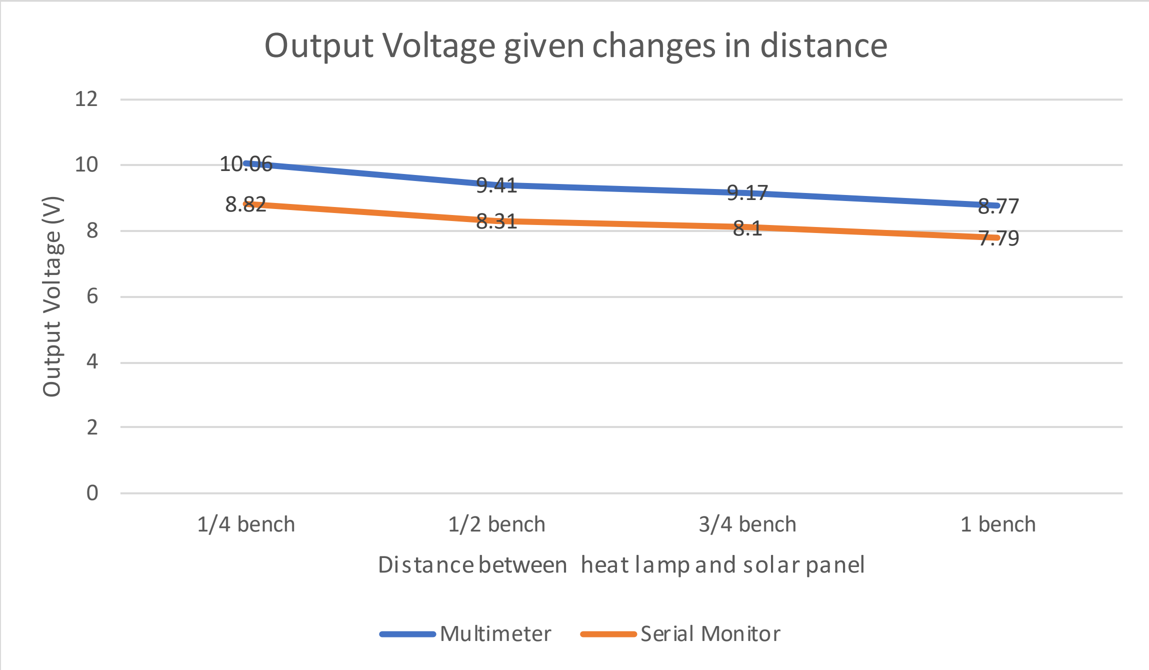
Performance of the system:

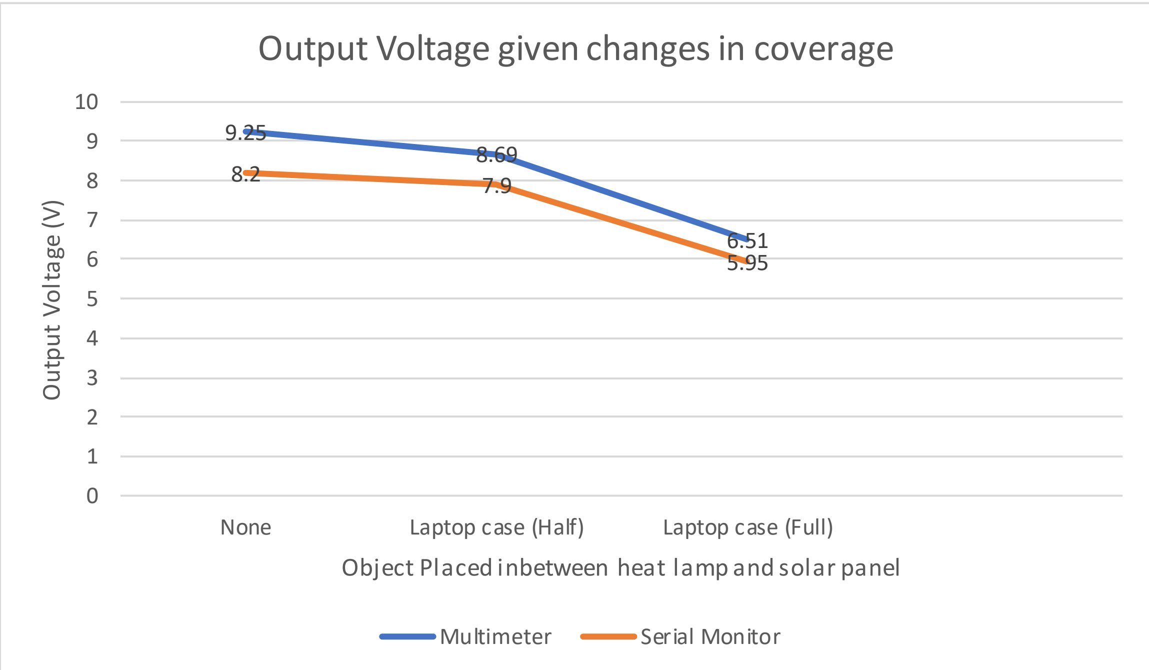


As seen in the chart 1, output voltage steadily fell as the PV Panel was increasingly angled away from the heat lamp. In terms of observed brightness, low brightness was seen at 90°, very low brightness from 50-30°, and no brightness from 20-10°. It is worth noting the differences in recorded voltage between the digital multimeter and the serial monitor, the multimeter recording ~0.71V higher on average.

As seen in chart 2, output voltage fell as the heat lamp was moved further away from the PV panel. At ¼ bench, observed brightness was very high, at ½ bench, high, at ¾ bench, medium, and at 1 bench, low. The same differences in recorded voltage can be seen, this time the multimeter reading 1.06V higher on average.

As seen in chart 3, output voltage decreased as more of the heat lamp was covered by an object. Observed brightness was very high when there was no object, high when half a laptop case was placed in the way, and low when a whole laptop case was placed in the way. The multimeter read 0.79V higher on average.





When the lamp was switched off, the PV system still produced voltage, albeit far lower than when the heat lamp was switched on. This is due to the LED lights being on in the lab and being a less efficient form of light energy than the heat lamp.

Exercise 6:

In exercise 6, the open-circuit voltage and short-circuit current of the solar panel at different angles of the heat lamp and PV panel were measured, in order to understand how much power the PV panel can produce at each orientation. The results are as follows:

Angle (degrees)	Open-circuit voltage (V)	Closed-circuit voltage drop (V)	Closed-circuit current multimeter	Closed-circuit current calculated
90	8.88	6.212	0.003	0.0282
50	8.34	2.002	0.002	0.0091
40	8.20	1.834	0	0.00833
30	7.88	1.784	0	0.0081
20	7.49	0.868	0	0.0039
10	7.16	0.724	0	0.00329

3 PV controller design

The three methods of controlling the photovoltaic (PV) panel's output were adjusting the orientation, adjusting the duty cycle manually, and adjusting the duty cycle automatically.

3.1 Orientation of PV panel

Adjusting the angle at which the photovoltaic panel faces the light source can control the output voltage of the panel. In the system a DC-DC converter is used to generate an output voltage that is higher than the input voltage by converting the current into a higher voltage. Due to the photovoltaic effect the current produced is proportional to the intensity of the incident light, and as the angle of the photovoltaic panel is changed the intensity of light changes, and thus, the output voltage changes. In a real-life scenario, the angle of the photovoltaic panel would be adjusted in accordance with local geography so that the photovoltaic panel receives the greatest intensity of light on average with the path of the sun. Producing a greater output voltage is important as powering a house requires such. Adjusting the angle of the photovoltaic panel is often does as the path of the sun relative to the position of the panel changes throughout the year, necessitating an adjustment of the angle of the panel to achieve a greater output voltage.

3.2 Manually Adjusting PWM Potentiometer

A DC-DC boost converter converts input current into a higher output voltage. Within the DC-DC boost converter's circuit contains a MOFSET. The measure of the amount of time the MOFSET is on compared to off is represented by the duty cycle. This duty cycle is manually controlled by a PWM potentiometer. Fundamentally, a DC-DC boost converter stores energy in the inductor, where the inductor current is only transferred during the off-time of the MOFSET. As the duty cycle is increased the inductor will attempt to generate a greater voltage so that the current may continue flowing. However, the duty cycle can only be increased to a certain point until the output voltage attempted cannot be attained as the input current becomes too low. Additionally, if the duty cycle is too high the current will not have enough time to flow through the entire circuit. Thus, as the duty cycle is increased beyond a certain point will result in a decreased output voltage. The PWM potentiometer is used with a photovoltaic panel to maintain a constant output voltage as extraneous factors such as clouds affect the input voltage. The PWM potentiometer is manually adjusted as the input voltage is lowered as to maintain this constant output voltage. However, the primary limitation of which is that it is manual thus, necessitating constant human involvement, due to how rapidly the light source may be obstructed. Additionally, the PWM potentiometer method maintains the flaws associated with a boost converter, and as input currents of photovoltaic panels are typically low these flaws are especially pronounced.

3.3 Automatically Adjust PWM Potentiometer

The primary flaw of the manually adjusting PWM potentiometer was always the need of human control. This is however solved by computer automated PWM potentiometer control, where the duty cycle is adjusted as to maintain a constant output voltage. Furthermore, the necessary calculations for the duty cycle result in a more accurate voltage that was attainable through human control who in the laboratory was merely observing the brightness of an LED.4

4 Teamwork

Our group is consistent of three on-campus students and two online students. However, for a large portion of the laboratories it was three online students due to extraneous factors.

4.1 Lab 2

Initially during Week 2 communication between online and on-campus students was sparse, this was largely due to the nature of the lab's tasks. Lab 2 Part 1 was a short task and thus, was done by a single individual. Part 2 of the lab mostly necessitated on-campus work with soldering, thus involvement between peers was largely done between on-campus students, where the online students were purely observational. The calculations of Lab 2 were done individually by online students and sent to all through Zoom. The group immediately realized that this lack of communication did not promote equal learning outcomes across all peers and thus, it was deemed as inefficient.

4.2 Lab 3

As a result of prior issues, in the third week we allocated tasks at the start of the experiment. However, the quantity of work assigned to online students was extremely minimal, being a single calculation. The lab instructions were designed in a manner such that all tasks would be completed on a single computer. As such, most of the lab was done by a single individual where others maintained a purely observational role. Furthermore, due to hardware and software availability online being somewhat limited work was again largely done by on-campus students. Moreover, the nature of the tasks was extremely guided and as such opportunity for interaction between peers was again significantly limited. As for the post-lab tests online students seemed oblivious to which, and thus, left the zoom meeting forcing on-campus students to answer all questions.

4.3 Lab 4

By the fourth week, practically every week's trials were dependent on the hardware facilities (Arduino board and so on). From week 4 outwards, it was mostly the on-campus students who shared the screen and the online students who saw the experiments in our group. Week 4's lab specifically involved no contribution from online student's due to the nature of the task involving an Arduino, breadboard, and Arduino software which online students are unable to interact with. Reflecting on the online student's largely observational role the on-campus students made a greater effort to ensure online students could fully understand what on-campus students were doing by directing the camera over the Arduino, constantly sharing the screen, and asking the online members whether they had any questions. Between on-campus students (of which there were only two during this lab) the exercises were split evenly between both students. When the on-campus students encountered problems, they actively sought assistance from the online students and attempted to solve them collectively. However, a major caveat of text-based communication was time lags between questions and response. Therefore, even as on-campus students sought help from online students due to lacking communication and significant delays these problems were usually tackled by the on-campus students. An example of this was the bit-banging process to turn the servo. However, this problem specifically was unable to be addressed by either on-campus or online students. Excluding which the lab exercises were completed promptly. However, yet again online students did not contribute to post-lab tests, and thus the mark was based only on the knowledge of two on-campus students.

4.4 Lab 5

Lab 5 was done entirely by on-campus student's as it required access to an Arduino for all Exercises. This lab was done very efficiently as almost all steps were provided for by the notes and those that were not, were solved quickly by on-campus students. On-campus students-maintained communication via camera and screen sharing constantly throughout the lab, but due to the simplicity of it no issues or questions arose. Between the two on-campus students in attendance, exercises were split evenly. However, yet again online students did not contribute to post-lab tests, and thus the mark was based only on the knowledge of two on-campus students.

4.5 Lab 6

Due to the physicality and pure experimental nature of the laboratory in recording measurements, it was entirely conducted by on-campus students. Rather than splitting the exercises and then completing them individually, this lab was done collaboratively by on-campus students. This was a far more effective process than previous labs promoting significantly more engagement as members were kept occupied throughout the entirety of the lab. The exercises were completed effectively, and data was shared to all members through a word document created by online students. This lab did not include a post-lab test.

4.6 Lab Tests

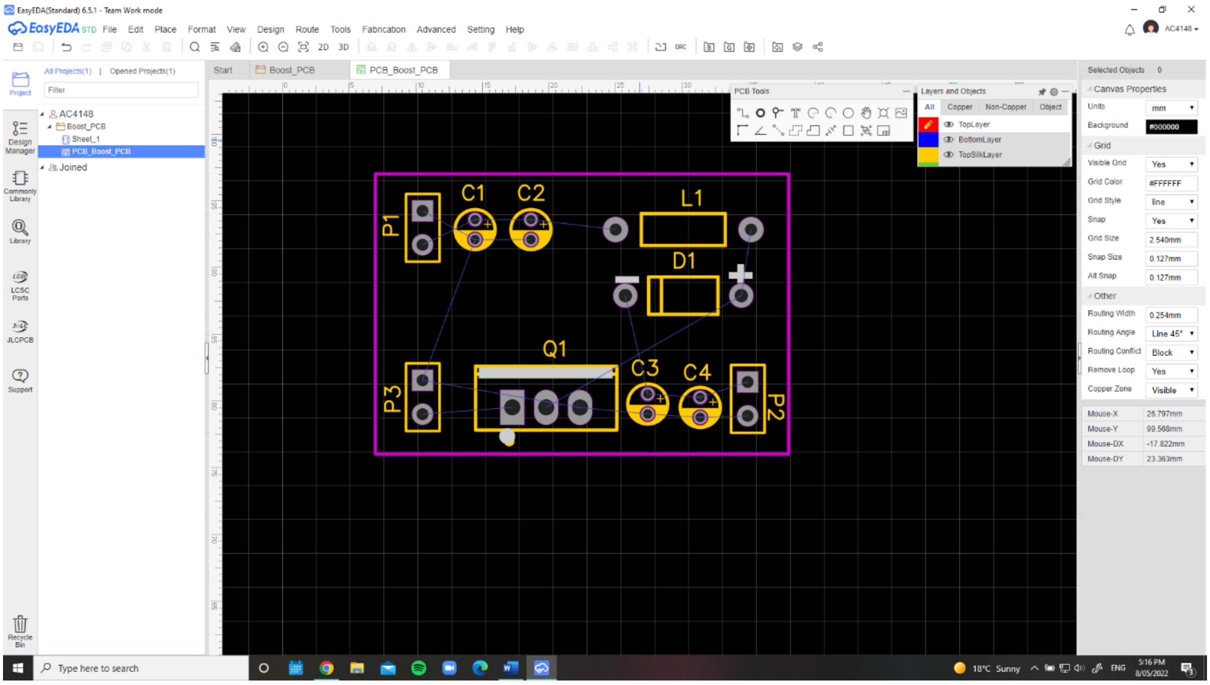
A major issue within the team was contribution to the post-lab test in which online students would have consistently left the zoom leaving on-campus students to solely answer the tests. As such, since the team largely composed of only 2 on-campus student's lab test results, although they consistently were highly marked, they may not be truly representative of the team's competence. This was a major issue within the team as this was not effectively communicated to online students. Moving forward on-campus students will remind online students to remain after the exercises have been completed so that they may contribute to the lab tests. Additionally, on-campus students will further implore engagement in answering which by online students. Between on-campus students

We successfully completed the ELEC1004 project from weeks two to six. During this period, we realized the importance and necessity of teamwork and excellent communication. We are unable to finish the full assignment separately due to the short time allocated for the experiment (3 hours). We would have done better in our group project if we had communicated more actively and given duties from the start. In the following weeks, we will continue to communicate actively and give duties in a systematic manner. This is critical to us since we cherish this opportunity to collaborate on the project and really want to obtain more professional and transferable skills from the Elec1004 project and cooperation.

Conclusion

In conclusion, from week 2 to week 6, we completed our Elec1004 project, which was separated into three main areas comprising boost converter design and control, solar energy system and panel performance and PV controller design. In the lab, our group members worked together online and offline, using a combination of electronic hardware and “Simulink” to simulate circuit diagrams and collect experimental data. During this process, although communication and task allocation were not particularly excellent at the start of the experiment. However, we were able to immediately alter our group's plan and collect the data in a timely and thorough manner in the following weeks. We would have been more efficient and capable of completing the project if we had been explicit about which portion of the trial everyone was working on for each main parts in the first lab.

Appendix



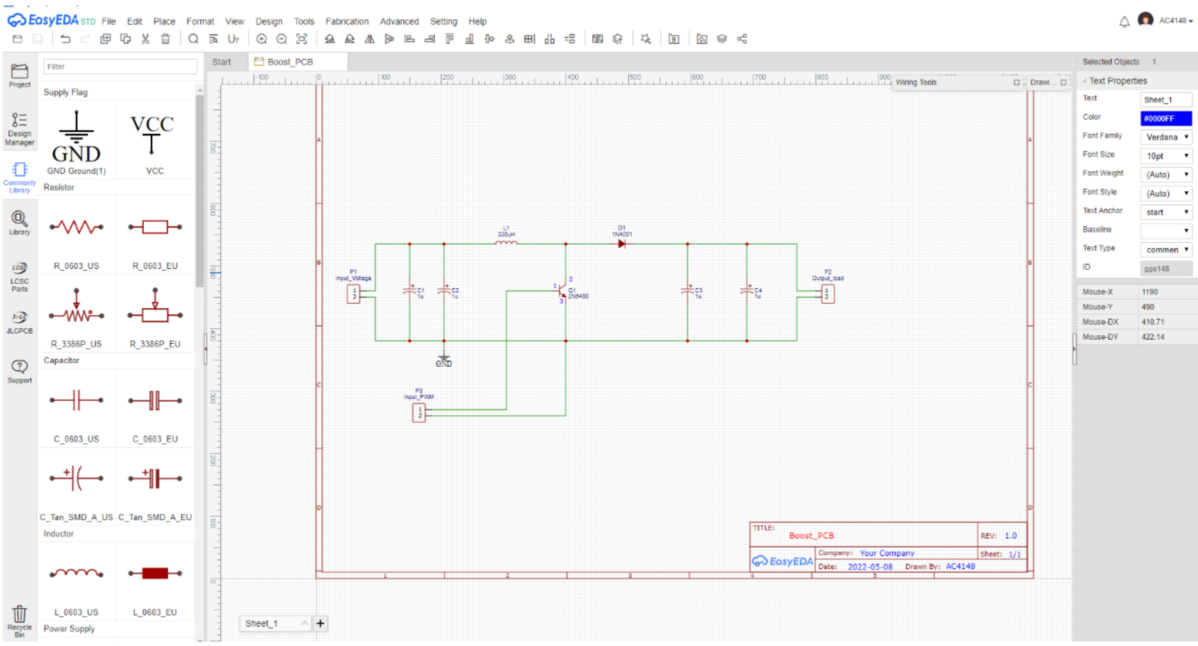


Figure 1. PCB Design

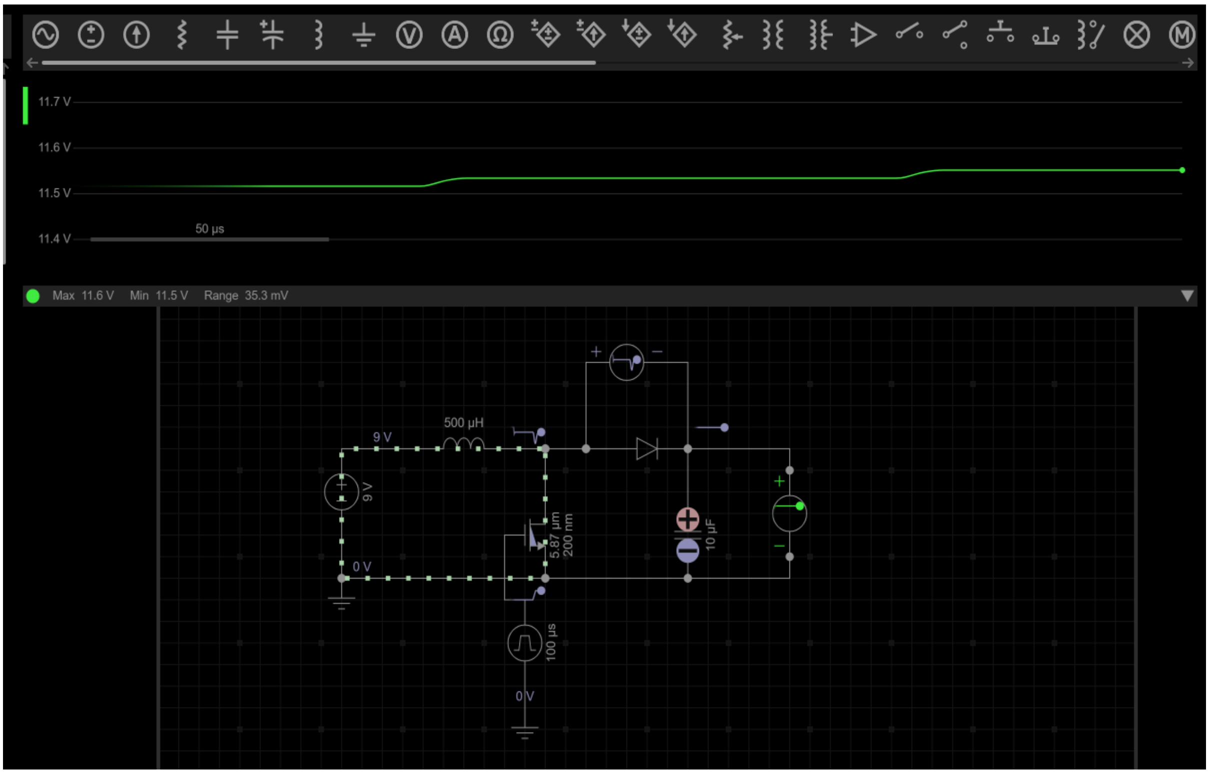


Figure 2. Circuit diagram on-state

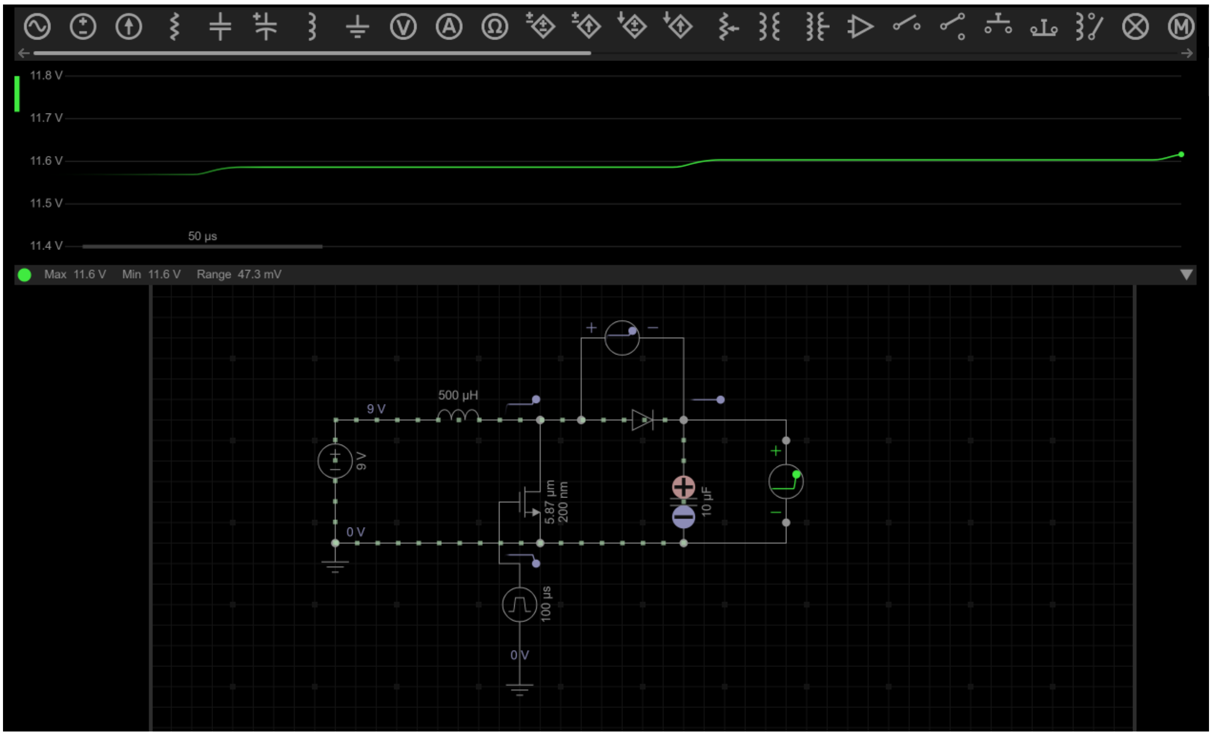


Figure 3. Circuit diagram off-state

```
bool isUNO  = true;           // Uno, Nano, Micro Magician, Mini Driver, Lilly Pad and any other board using ATmega 8, 168 or 328
bool isMEGA = !isUNO;         // Mega1280, Mega2560, MegaADK, Spider or any other board using ATmega1280 or ATmega2560

#define PWM           9
#define potinput      A0
#define outputvoltage A1
#define Button3       11      // Button closest to potentiometer

double potinputval;
double PWMvalue;
double output_sense;
double output_voltage_conv;

double potmin         = 0;    // potmin & potmax can be used to scale the range of the pot (and resulting duty cycle)
double potmax         = 1023; // potmin = 0, potmax = 1023 allows the full range of 0-100% duty cycle
//double potmax       = 255;  // potmin = 0, potmax =  255 limits the range of duty cycle adjustment to 0-25%

double Duty_Cycle     = 0;

int buttonPushCounter = 0;    // counter for the number of button presses
int buttonState       = 0;    // current state of the button
int lastButtonState   = 0;    // previous state of the button

int  interval         = 1000; // interval between successive writes to Serial Monitor
long previousMillis   = 0;
long currentMillis    = 0;

void setup(void)
{
  Serial.begin(9600);
  pinMode(PWM, OUTPUT);
  pinMode(potinput, INPUT);
  pinMode(outputvoltage, INPUT);
  pinMode(Button3, INPUT);

  // Set appropriate timer divisor to increase pin 9 PWM frequency
  if (isMEGA) TCCR2B = TCCR2B & B11111000 | B00000001; // Mega: set timer 2 divisor to 1
```

```
void loop(void)
{
  // Read the potentiometer 10-bit ADC input. Scale it if desired to reduce the functional range of pot.
  potinputval = analogRead(potinput);
  potinputval = map(potinputval, 0, 1023, potmin, potmax); // Can limit effective range of pot to prevent overvoltage

  // Scale 10-bit potinputval to 8-bit PWMval (map full range of pot to full range of PWM)
  PWMvalue = map(potinputval, 0, 1023, 0, 255);

  // Read output voltage 10-bit ADC input. Convert to voltage using Vref = 5V and 20k:1k voltage divider
  output_sense = analogRead(outputvoltage);
  output_voltage_conv = ((output_sense / 1024) * (5))*(21);

  // Read the start/stop button (normal state is HIGH, pushing button pulls output LOW)
  buttonState = digitalRead(Button3);

  // Possibly some debounce code: Need to pass a counter value of 1 or 2 to the switch() function
  if (buttonState != lastButtonState)
  {
    if (buttonState == LOW)
    {
      buttonPushCounter++;
      if (buttonPushCounter > 2 ) // until 1 to 2
      {
        buttonPushCounter = 1;
      }
    }
    delay(10);
  }
  lastButtonState = buttonState;

// ------------------------
// Section 3 - Act on inputs
// ------------------------

  // If circuit is active   (count=1) write data to Serial and write PWM value to output
  // If circuit is inactive (count=2) write message to Serial and write PWM = 0 to output
  switch (buttonPushCounter)
  {
    case 1:
      currentMillis = millis();
      if (currentMillis - previousMillis > interval)
      {
        previousMillis = currentMillis;
        Serial.print("Duty Cycle: ");
        Serial.print((PWMvalue / 255) * (100));
        Serial.print("%");
        Serial.print("    Output_Voltage: ");
        Serial.print(output_voltage_conv);
        Serial.println("V");
      }
      analogWrite(PWM, PWMvalue);
      break;
    case 2:
      currentMillis = millis();
      if (currentMillis - previousMillis > interval)
      {
        previousMillis = currentMillis;
        Serial.println("Boost converter is inactive (PWM = 0)");
      }
      analogWrite(PWM, 0);
      break;

    delay(1);  // delay in between reads for stability
  }
```

Figure 4. PWM controller Arduino code