

**EE464 Hardware Project Report**

Date of Submission : 02.06.19

Ekin Alp Biçer & Mert Elmas & Onur Külahlıoğlu

2030385 2089357 2031102

Table of Contents

1 - Introduction 2

2 - Design 2

2-a- Circuit Schematic 2

2-b-Theory and Selection of Design Parameters 2

2-c- Control Algorithm 2

2-d- Simulation Results 2

3 - Materials and Cost 2

3-a- Materials List 2

3-b- Actual & Total Cost 2

4 - Implementation 2

5 - Results 2

5-a- Operation & Deliverables 2

5-b-Actual Design Parameters 2

Inductance 2

Turns Ratio 2

Input Voltage & PWM Relationship 2

5-c- Temperature Results 2

5-d- Stability Results 2

6 - Challenges 2

7 - Conclusion 2

8 - Appendix 2

# 

# 1- Introduction

In this report, Hardware Project, namely Forward Converter #3 for EE-464 course is documented. It explains the design process, material selection & cost analysis, obtained results, challenges faced during the project and includes a conclusion that summarizes the whole process. The specifications for the converter are explained in table 1. In order to get this input and output characteristics the PWM applied on the switches or switches must be regulated, which is done by an Arduino Nano module. As the input voltage increases PWM ratio decreases and vice versa. Also the line regulation is controlled by the controller algorithm as well. Other specifications such as output voltage peak to peak ripple is regulated through the choice of the inductor.

As expected, all these ideas needed to be implemented on a physical body. Amongst a choice of two, the other being implementing on a PCB layout, the project was assembled on stripboard which was troublesome on occasion to work with. All the problems that we have endured will be explained throughout this report as well as the design process, simulations, cost analysis and the outcome of the final implementation.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Minimum Input Voltage (V) | Maximum Input Voltage (V) | Output Voltage (V) | Output Power (W) | Output Voltage Peak to Peak Ripple (%) | Line Regulation (%) | Load Regulation (%) |
| 24 | 48 | 10 | 40 | 2 | 2 | 2 |

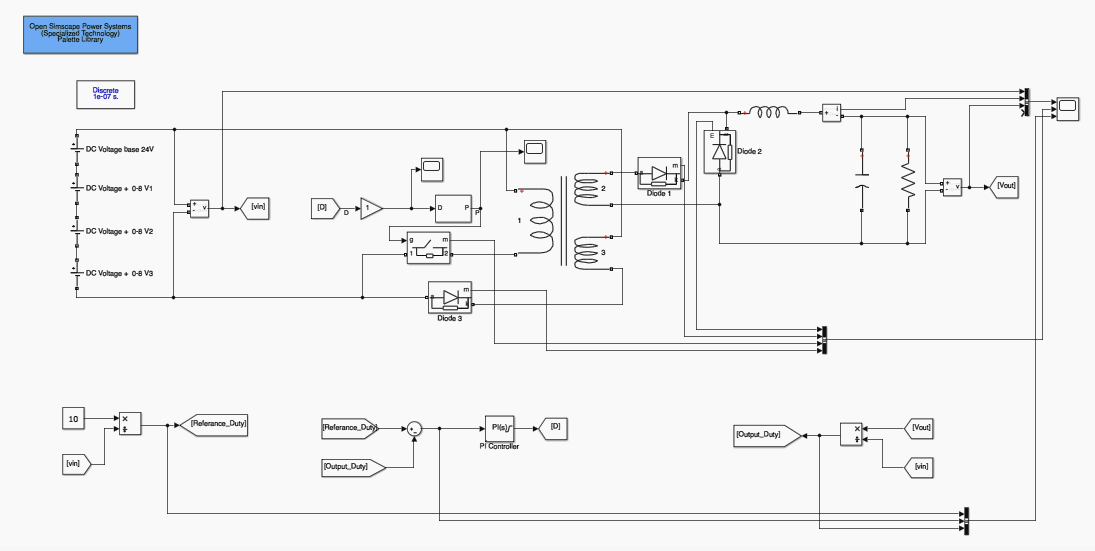
***Table 1:*** *Specifications for the Forward converter* #3 *design.*

# 

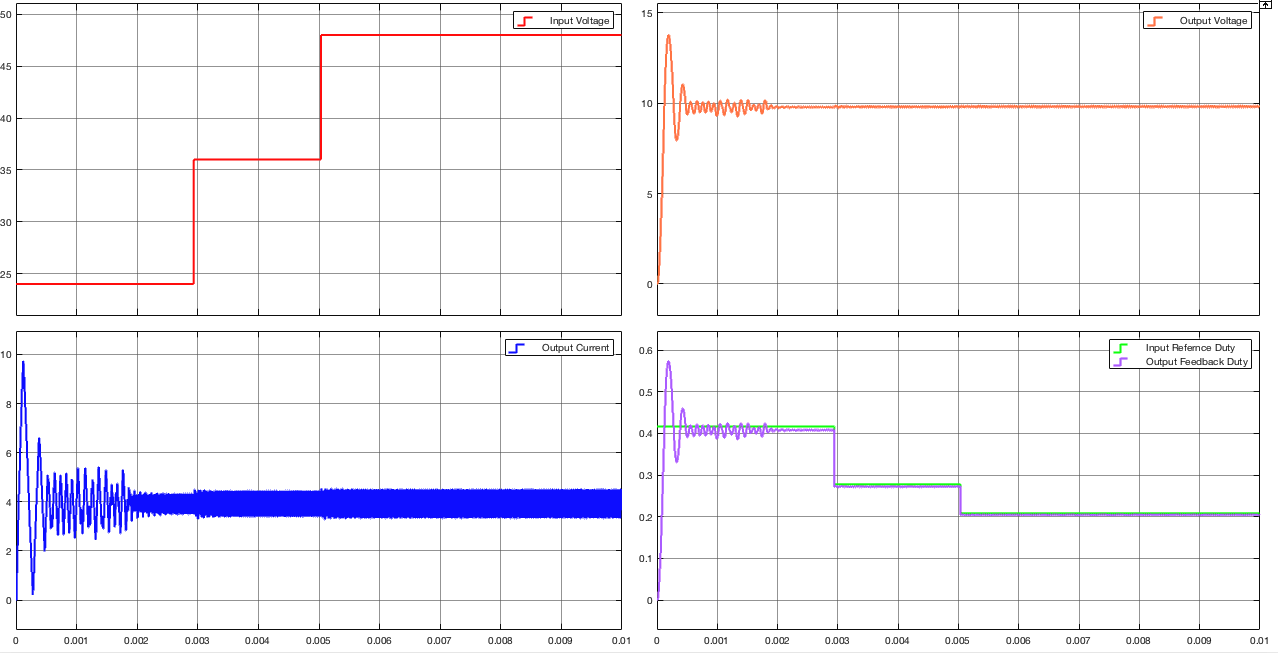
# 2- Design

For the previous project which was named EE464 Simulation Project 2, an earlier stage of the design to be implemented was explored. In figure 1 the schematic for it can be seen.

However, the actual design shown in the demonstration was quite different than the one in figure 1. It was decided that it was necessary to add a second switch, and a snubber capacitor because it was not possible to pump back the energy back to the source effectively enough with the 3rd winding and hence the MOSFET and the diode were heating up alarmingly (boiling spits and melting solder). Secondly, the Arduino algorithm needed to realize the pid controller on the first design could not be implemented therefore it was changed to something simpler. For these reasons, we decided to include another simulation schematic to show a design that is more similar to the one on the demonstration.



***Figure 1:*** *Simulink schematic of the Forward converter design with feedback and PI controller.*

**

***Figure 2:*** *Input voltage, output voltage, inductor current and the duty cycle characteristics of the of the Forward converter design with feedback and PI controller.*

|  |  |
| --- | --- |
|  |  |

***Figure 3:*** *Steady state output voltage and Inductor current ripples of the Forward converter design with feedback and PI controller.*

2-a- Circuit Design with a simple PWM controller

\*\*\*\*\* 2 switch

\*\*\*\*\* 3rd winding

2-b- Theory and Selection of Design Parameters

2-c- Control Algorithm

The algorithm that we used is a simple Arduino code that compares the feedback from the output voltage with the reference voltage. It should be noted that the main difference between this implementation and the one we had in mind in the first design is that this version does not include a feedforward loop that could otherwise generate unstability. The other major difference is that PI controller that the first simulink design had could not be included in this final implementation which was the other main reason why it was necessary to change the controller algorithm in the first place. In the table below, this Arduino code can be seen.

|  |
| --- |
| void setup() {  Serial.begin(9600);  pinMode(Vcontrol, INPUT);  pinMode(feedback, INPUT);  pinMode(PWM, OUTPUT);  TCCR2B = TCCR2B & B11111000 | B00000001; // pin 3 and 11 PWM frequency of 31372.55 Hz  }  void loop() {  int cont = analogRead(Vcontrol);  int output = analogRead(feedback);  dummy = pwm;  if (output > cont)  {  if (pwm < 60)  {  pwm = dummy -0.5;  pwm = constrain(pwm, 1, 127);  } else  {  pwm = dummy-1;  pwm = constrain(pwm, 1, 127);  }  }  if (cont > output)  {  if (pwm < 60)  {  pwm = dummy+0.5;  pwm = constrain(pwm, 1, 153);  } else  {  pwm = dummy + 1;  pwm = constrain(pwm, 1, 153);  }  }  analogWrite(PWM,pwm);  Serial.print(" PWM = ");  Serial.print(pwm);  Serial.print(" , Feedbac: ");  Serial.print(output);  Serial.print(" , control: ");  Serial.println(cont);  } |

***Table 2:*** *Arduino code used in the demonstration..*

As it is obvious from this code that if the output voltage is smaller than the reference voltage, generated duty cycle must increase and if the situation is the other way around the generated duty must decrease while having a limit on the output as below 60%.

Also another notable feature of this code is that it employs a soft starting technique as it increases the duty cycle 0.32 percent (100\*0.5/153) per cycle.

2-d- Simulation Results

2-e- Design Preferences

2switch

Digital controller instead of analog

RC Snubber

# 3- Materials and Cost Analysis

3-a- Material List for the final product

|  |  |
| --- | --- |
| ***Name of the Component*** | ***Number in the Design*** |
| Arduino Nano | 1 |
| Pertinax | 1 (13cm x 25cm) |
| Various passive circuit elements like resistances and capacitors | Numerous |
| Inductor : (core, carcass, windings) | 1 |
| Transformer (core, carcass, windings) | 1 |
| Diodes (U1660g) | 3 |
| MOSFET | 2 |
| TLP250 | 2 |
| Input & Output sockets | 4 |
| AWG 21 cable | 1m |

***Table 3:*** *Initial expected component list before implementation.*

3-b- Cost Analysis

This chapter includes the cost analysis of the project. In table 4, the cost of an individual product is documented whereas in table 5, the total expenditure of the overall project is documented.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Total Cost***   |  |  | | --- | --- | | ***Name of the Component*** | ***Number used and the cost*** | | Arduino Nano | 1x30 =30 TL | | Pertinax | 1x15 = 15 TL | | Various passive circuit elements like resistances and capacitors (already had) | Kx0 = 0 TL | | Inductor (leased only) | 1x0 =0 TL | | Transformer (leased only) | 1x0 = 0 TL | | Diodes (U1660g) | 3x7.5 = 22.5 TL | | MOSFET | 2x5 =10 TL | | TLP250 | 2x6.95 =13.9 TL | | Input & Output sockets | 4x1= 4TL | | AWG 21 cable (in 1m unit) | 1x1 = 1 TL | | Total | 96.4 TL |   ***Table 4:*** *Expenditure on the final product.* | ***Actual Expenditure***   |  |  | | --- | --- | | ***Name of the Component*** | ***Number used and the cost*** | | Arduino Nano | 3x30 =90 TL | | Arduino Uno | 2x35 =70 TL | | Pertinax :1 | 1x15 = 15 TL | | Various passive circuit elements like resistances and capacitors | Kx0 =0 TL | | Inductor (leased only) | 1x0 = 0 TL | | Transformer (leased only) | 1x0 = 0 TL | | Diodes (U1660g) | 8x7.5 = 60 TL | | Diodes IN5822 | 1x20 = 20 TL | | MOSFET | 5x5 = 25 TL | | TLP250 : 2 | 2x6.95 =13.9 TL | | Input & Output sockets | 4x1= 4 TL | | AWG 21 cable(in 1m unit) | 3x1 =3 TL | | Total | 300,9 TL |   ***Table 5:*** *Total expenditure after implementation.* |
|  |  |

# 4- Implementation

# 5- Results

5-a- Operation & Deliverables

Efficiency

Compactness

Soft switching

Closed Loop Digital Controller

Isolated Output

Actual Converter Schematic

Input/Output Relationship

Controller

5-b-Actual Design Parameters

Inductance

Turns Ratio

Input Voltage & PWM Relationship

5-c- Temperature Results

5-d- Stability Results

***Figure 1:*** *Schematic of the Forward converter design.*

# 6- Challenges

TLP250 Optocoupler

Arduino

During the implementation

MOSFET

External Batteries

Soldering

Finding the Suitable Circuit Elements

--capacitor for the output of the optocoupler, resistances for proper voltage division

Diode slow diode→ high switching loss

RC snubber

***Figure 1:*** *Schematic of the Forward converter design.*

# 7- Conclusion

In this report, all the design stages and implementation process are documented. It began by addressing the design specification and continues with the design and simulation of the initial solution that we have come up with than the actual simulink model of the actual design is included. Afterwards, it includes the results that we have obtained throughout the design process and the demonstrations and cost analysis of the project. Finally, the report concludes with the challenges that were faced and this chapter.

# 8- Appendix