UNIVERSIDAD PONTIFICIA
ICAI ICADE
COMILLAS
MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

INGENIERO ELECTRÓNICO

SOLAR POWERED WATER FILTRATION AND VENDING SYSTEM

Autor: Lixiang Dong

Director: Bryce Smith

Madrid
Julio, 2018

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESINAS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D. Lixiang Dong

DECLARA ser el titular de los derechos de propiedad intelectual de la obra: Solar Powered Water Filtration and Vending System,

que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar "marcas de agua" o cualquier otro sistema de seguridad o de protección.
b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
e) Asignar por defecto a estos trabajos una licencia Creative Commons.
f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente).*

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

a) Que la Universidad identifique claramente su nombre como autor de la misma
b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
c) Solicitar la retirada de la obra del repositorio por causa justificada.
d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e

intereses a causa de la cesión.

d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusive del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a ……3….. de ……Julio…………………... de ..2018..

ACEPTA

Fdo…董理想……………………………………

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título ...Solar Powered Water Filtration and Vending System...

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico ...4º... es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: 董 Fecha: 29/05/2018

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Beyce Smith Fecha: 29/05/2018

SISTEMA DE FILTRADO Y EXPENDICIÓN DE AGUA ALIMENTADO POR ENERGÍA SOLAR

Autor: Dong, Lixiang.

Director: Smith, Bryce.

Entidad Colaboradora: University of Illinois at Urbana-Champaign

RESUMEN DEL PROYECTO

Motivación

Solar Chapter, una organización sin ánimo de lucro, llevó a cabo este año un proyecto en el que construyeron una bomba de agua solar en el pueblo de Umutnana, situado en la provincia de East Nusa Terengganu, Indonesia. Esto permitió a sus habitantes obtener agua sin tener que desplazarse largas distancias cada día. Sin embargo, esta agua tenía que ser purificado de alguna manera antes de poder ser consumida de manera segura.

Al estar situados en una zona rural, con un limitado acceso a la red eléctrica, sus habitantes hierven el agua usando leña como combustible, un proceso lento y poco eficiente. Pero la mayor preocupación según Solar Chapter, es la cantidad de humo que se inhala producido por la combustión de la leña, resultando en serios problemas respiratorios. A esto, hay que añadirle los problemas medioambientales debido a la tala de árboles para obtener este recurso.

Objetivo

En consideración con este proyecto ya finalizado de Solar Chapter, se ha decidido complementarlo con un sistema de purificación de agua para eliminar la necesidad de esterilizar el agua usando leña como combustible. Se hará uso de membranas de filtrado como elemento de purificación. Se incorporará una bomba amplificadora a la bomba de agua existente para proporcionar suficiente presión para impulsar el agua a través de las distintas membranas de filtrado. El agua filtrada se almacenará en un tanque cilíndrico para su posterior expendición.

El sistema de purificación y expendición de agua funcionará mediante energía solar con el fin de poder ser implementado en entornos no electrificados. La energía generada por el panel solar será almacenada en una batería de plomo-ácido de 12 V. Esta batería se encargará de alimentar la bomba amplificadora dc, válvulas eléctricas, microcontrolador, interfaz de usuario, y sensores.

Metodología

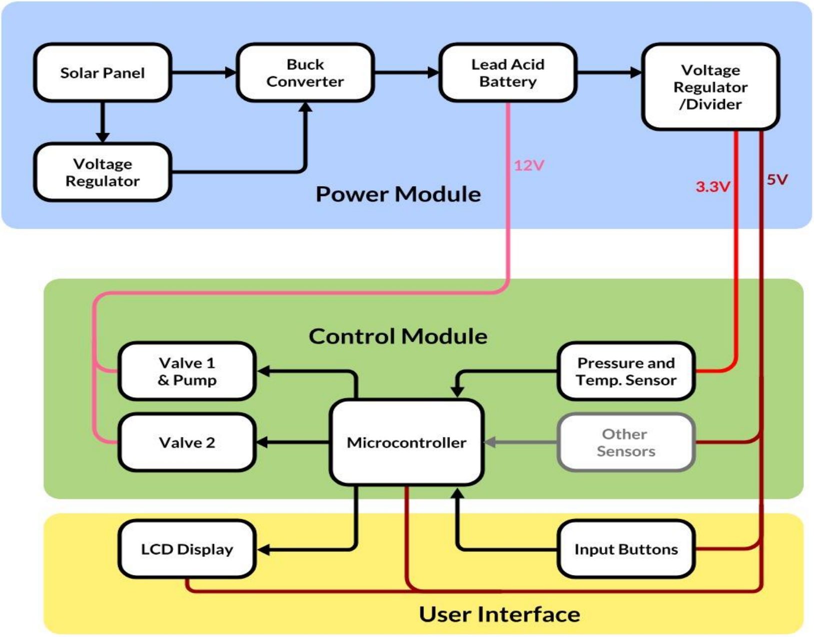


Figura 1. Diagrama de bloques

El sistema de purificación y expendición está compuesto de tres módulos; módulo de potencia, módulo de control, e interfaz de usuario (ver Fig. 1).

1. Diseño

1.1 Módulo de Potencia

- Panel Solar:

La potencia nominal de la bomba amplificadora dc y las válvulas eléctricas empleadas está en torno a los 24 W y 36 W respectivamente. A esto, hay que incluirle una batería de 12 V/12 AH cuya potencia nominal es de unos 36 W, considerando que la corriente de carga máxima permitida de la batería es de 3 A. El sistema en su totalidad requiere de unos 96 W para su funcionamiento. Por lo que se usará un panel solar de 100 W para alimentar el sistema.

- Batería plomo-ácido 12 V:

Se usará una batería de plomo-ácido porque es la solución para almacenamiento más económica. Para mejorar la vida útil de la batería, se cargará usando el voltaje de flotación para compensar la auto descarga. Para una batería de plomo-ácido de 12V, este voltaje esta entre los 13.5 y los 13.8 V [1]. No obstante, el hecho de cargar la batería siempre a la tensión de flotación implica que la batería nunca alcanzaría su capacidad máxima, aunque esto no influye en el funcionamiento del sistema.

- Convertidor DC-DC (Buck):

  Cuando un panel de 100 W recibe suficiente radiación solar, su tensión de salida es mayor a la tensión de la batería de 12 V. Por lo que es necesario un conversor para disminuir estas tensiones a un nivel más adecuado. Teniendo en cuenta estos puntos, se ha diseñado un convertidor Buck con una eficiencia superior al 85% [2]. La eficiencia se ha calculado conociendo la tensión y corriente de entrada mostrado por la fuente de alimentación y conectando una carga electrónica en modo voltaje para conocer los valores de salida.

  El controlador PWM encargado de ajustar el ciclo de trabajo y obtener la tensión de salida deseada es implementado de manera analógica formado por un generador de error, control PI, comparador, y un generador de señal triangular.

1.2 Módulo de Control

- Microcontrolador:

  El microcontrolador utilizado es el ATMega328-pu. Se encarga de recoger la información del sensor de presión, así como de los pulsadores para poder mandar la señal de salida adecuada a la bomba amplificadora, válvulas, y la pantalla LCD. El propósito es mantener siempre unos niveles adecuados de agua en el tanque y controlar el volumen de expendición escogido por el usuario. Para facilitar el entorno de programación, se ha escrito el código en Arduino IDE y cargado al chip utilizando una placa Arduino.

- Válvulas Eléctricas:

  El sistema presenta dos válvulas, una de entrada al tanque y una de salida. La apertura de las válvulas se controla activando un transistor BJT que actúa de interruptor.

- Sensor de Presión:

  El sensor de presión utilizado es el MS5803-14BA, soldado a su tablero de desglose. El tablero es impermeabilizado recubriéndolo de silicona, dejando expuesta la membrana de gel.

- Bomba Amplificadora:

  La bomba amplificadora se alimenta a una tensión de 12 V. Es necesaria para producir suficiente presión para impulsar el agua a través de los distintos tipos de filtros.

1.3 Interfaz de Usuario

Permite al usuario escoger el volumen de agua que desea mediante 3 pulsadores que corresponde a 1, 2, y 3L de agua respectivamente, y un último pulsador para confirmar la opción escogida. Una pantalla LCD muestra información necesaria al usuario.

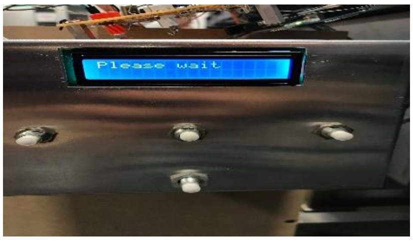


Figura 2. Imagen del interfaz de usuario

Resultados

Se comprobaron la funcionalidad de los distintos módulos por separado. En la figura 3 se muestra las tensiones de entrada y salida del conversor Buck. Como se puede observar, es capaz de disminuir las tensiones de entrada (20 y 15 V respectivamente) a la tensión de salida deseada (13.6V).

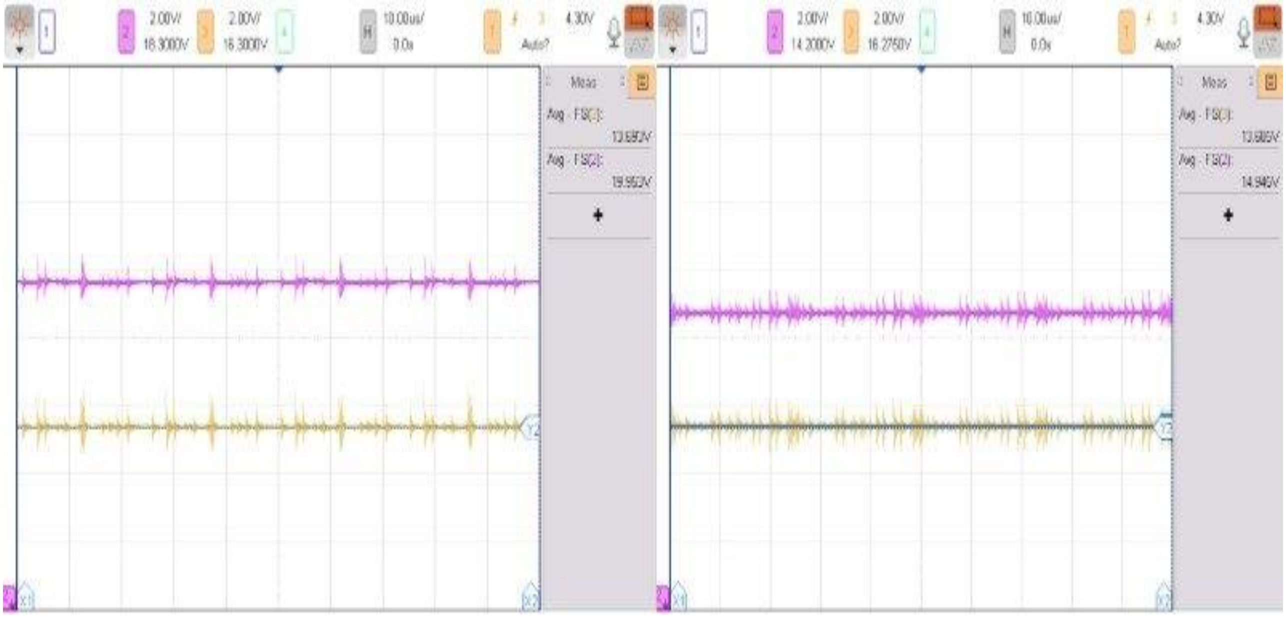


Figura 3. Tensión entrada (señal morada) y tensión salida (señal amarilla) del conversor Buck

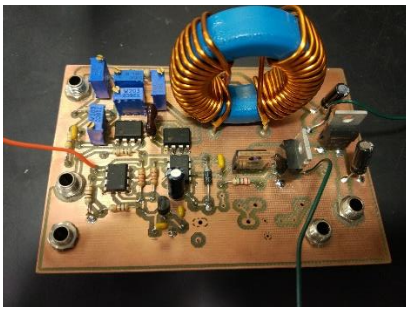


Figura 4. Convertidor Buck

El sensor de presión mide el valor de presión absoluta. Haciendo las conversiones necesarias usando la ecuación (1.1), se comprobó su precisión.

$$P_{abs} = P_{atm} + \rho * g * h \tag{1.1}$$

Para controlar la expendición del agua, se derivó la siguiente fórmula (1.2) para calcular el tiempo de apertura de la válvula, en función del área circular del tanque (A), área de apertura de la válvula (a), nivel de agua antes de la expendición (h_1), y nivel del agua posterior a la expendición (h_2).

$$t = \frac{2*A}{a*\sqrt{2*g}}(\sqrt{h_1} - \sqrt{h_2}) \qquad (1.2)\ [3]$$

Conclusión

Se pudo demostrar la funcionalidad de los distintos módulos individualmente. El conversor Buck proporciona la tensión de carga adecuada para la batería y el sistema de expendición proporciona un volumen de agua correcto dentro de un rango de error permisible. Sin embargo, no se pudo integrar el sistema completo, dada la falta de componentes tales como la batería y las membranas de filtrado.

SOLAR POWERED WATER FILTRATION AND VENDING SYSTEM

EXECUTIVE SUMMARY

Motivation

A non-profit organization named Solar Chapter built at the beginning of 2018 solar powered water pumps in the village of Umutnana, located in the province of East Nusa Terengganu, Indonesia. This allowed the villagers obtain water without the need to walk several kilometers every day. However, this water still needs to be treated before it could be consumed safely.

Located in a rural area with limited access to stable electric infrastructures, people in Umutnana boil water using firewood as fuel, which is a slow and inefficient process. Nonetheless, the biggest concern, according to Solar Chapter, is the amount of smoke inhaled resulted from the combustion of firewood. In addition, there are also environmental issues associated due to the cutting of trees to obtain firewood.

Objective

In consideration with the already finished project by Solar Chapter, it will be complemented with a water purification system to remove the need of sterilizing water using firewood. This purification system will make use of filtration membranes as purifying element. A booster pump will be added to the existing water pump to provide enough pressure to drive water through the different layers of filtration membranes. Filtered water will be stored in a cylindrical tank ready to be dispensed.

The water purification and vending system will be powered using solar energy, so it can be deployed in environments without a stable electric grid. Energy generated from the solar panel will be harnessed and stored in a 12 V lead-acid battery. This battery will power the dc booster pump, electric valves, microcontroller, user interface components, and sensors.

Methodology

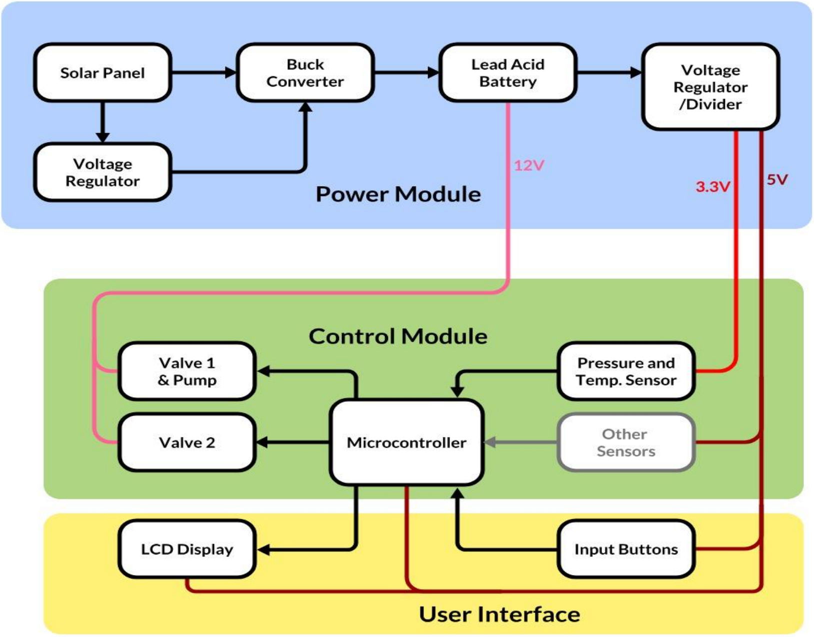


Figure 1. Block diagram

The water filtration and vending system is composed of three modules; power module, control module, and user interface.

1. Design

1.1 Power Module

- Solar Panel:

The power rating of the dc booster pump and the electric valves used are 24 W and 36 W respectively. In addition, there is a 12 V/ 12 AH battery with a power rating of 36 W, considering that the battery's maximum allowed charging current is 3 A. The system requires approximately 96 W to function. Therefore, a 100 W solar panel will be used to power this system.

- 12 V lead-acid battery:

This system will use a lead-acid battery since it is the cheapest way to store energy. To improve the battery's lifespan, it will be charged at the floating charge voltage to compensate self-discharge. For a 12 V lead-acid battery, the floating charge ranges between 13.5 and 13.8 V [1]. However, by charging the battery always at the floating voltage, it implies that the battery will never reach its full capacity. Nonetheless, this does no affect the functionality of the system.

- DC-DC converter Buck:

When a 100 W solar panel receives enough solar radiation, its output voltage is greater than a 12 V battery. Therefore, a converter is required to step-down the voltages to a more suitable level. Considering these points, a buck converter is designed with an efficiency greater than 85 % [2]. The efficiency is calculated by knowing the input voltage and current values displayed through the power supply and the output values displayed by the dc load.

The PWM controller used to adjust the duty ratio and obtain the desired output voltage is an analog implementation composed of an error generator, PI control, and a triangular wave generator.

1.2 Control Module

- Microcontroller:

The chosen microcontroller is the ATMega328.pu. It collects information from the pressure sensor, and input buttons to send the correct output signal to the booster pump, valves, and LCD display. Its purpose is to always maintain a suitable water level in the tank and control the dispensed volume chosen by the user. To facilitate the programming environment, the code was written in Arduino IDE and uploaded to the chip using an Arduino board.

- Electric Valves:

The system has two valves. The opening of the valves is controlled by activating a BJT transistor that acts as a switch.

- Pressure Sensor:

The pressure sensor used is the MS5803-14BA, soldered to its breakout board. The board is waterproofed by coating it in silicon, leaving the membrane gel exposed.

- Booster Pump:

The booster pump is powered at 12 V. It is needed to provide enough pressure to drive the water through the different types of filters.

1.3 User Interface

The user interface allows the user to choose the volume to be dispensed through 3 input buttons, corresponding to 1, 2, and 3 L respectively, and a last button to confirm choice. An LCD display shows the necessary information to the user.

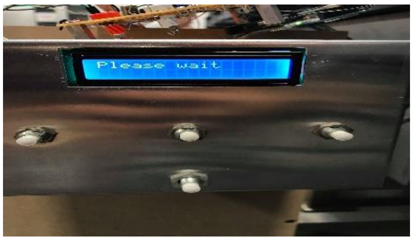


Figure 2. Image of the user interface

Results

The functionality of the different modules was tested separately. Figure 3 illustrates the input and output voltages of the buck converter. As it can see, the converter is able to decrease the input voltages (20 and 15 V respectively) to the desired output voltage (13.6 V).

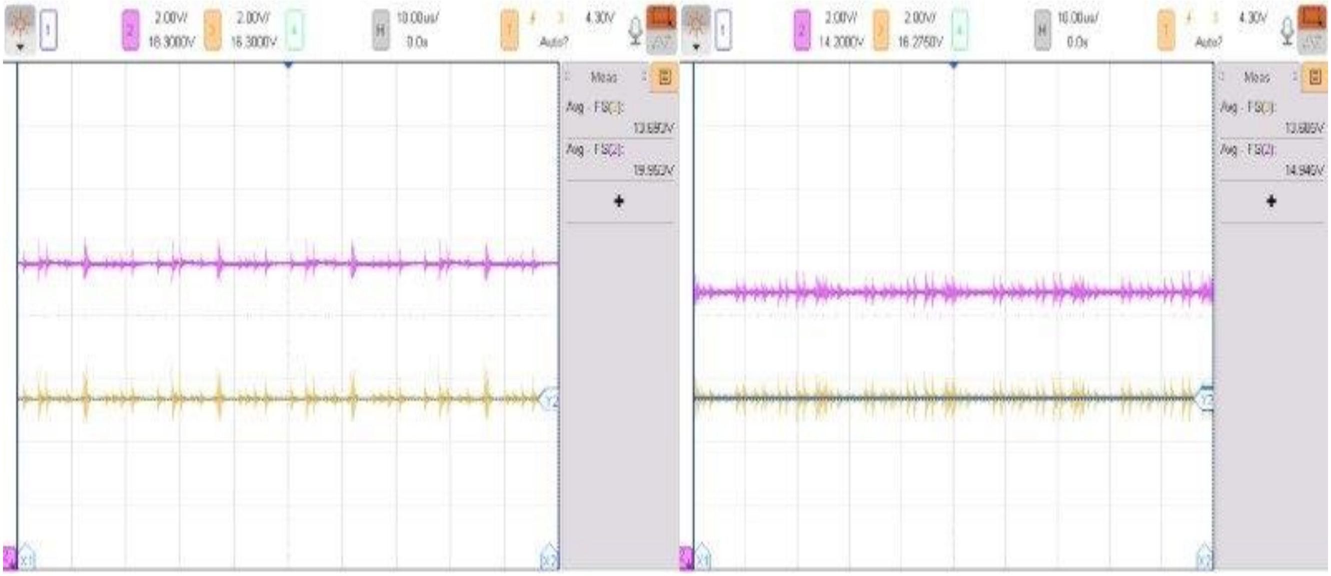


Figure 3. Input voltage (purple signal) and output voltage (yellow signal) of the buck converter

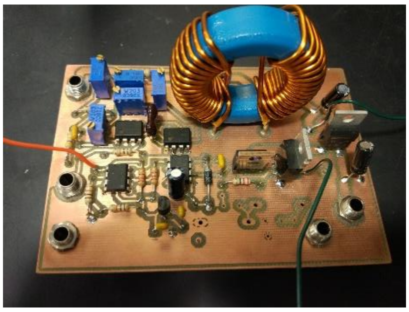


Figure 4. Buck converter

The pressure sensor measures the absolute pressure. Its precision was tested by doing the required conversions using eq. (1.1).

$$P_{abs} = P_{atm} + \rho * g * h \qquad (1.1)\ [3]$$

To control the water expending, the following equation was derived (1.2) to calculate the opening time of the valves, with respect to the circular area of the tank (A), opening area of the valve (a), water level before dispensing (h_1), and water level after dispensing (h_2).

$$t = \frac{2*A}{a*\sqrt{2*g}}(\sqrt{h_1} - \sqrt{h_2}) \qquad (1.2)\ [3]$$

Conclusion

The functionally of the different modules was tested individually. The buck converter supplies the desired charging voltage to the battery and the dispensing system provides the correct volume within an acceptable error range. Nonetheless, the system could not be integrated completely due to the lack of components such as the battery and the filtering membranes.

Reference

[1] B. Isidor, "BU-403: Charging Lead Acid," Available: http://batteryuniversity.com/learn/article/charging_the_lead_acid_battery. [Accessed Feb, 2018].

[2] A. Banarjee. ECE464. Class Lecture, Topic: "DC-DC conversion." College of Electrical and Computer Engineering, University of Illinois Urbana-Champaign, Urbana, IL.

[3] L. M. Mochón, Castro, Mecánica de Fluidos. Class Lecture. School of Engineering, Universidad Pontificia de Comillas ICAI, Madrid, España.

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

INGENIERO ELECTRÓNICO

SOLAR POWERED WATER FILTRATION AND VENDING SYSTEM

Autor: Lixiang Dong

Director: Bryce Smith

Madrid
Julio, 2018

ABSTRACT

This project consists in the design of the power electronics and water flow control of a solar powered water filtration and vending system. It is motivated by Solar Chapter, a non-profit organization created by Indonesian students in the US and the solar water pump project they carried out at the beginning of 2018.

A dc-dc converter is designed to be set between solar panel and 12 V lead-acid battery to allow charge at constant floating voltage. The converter's duty ratio is controlled by employing an analog implementation.

The water flow is controlled using a microcontroller to manage the switching of electric valves and booster pump based on the signal received from a pressure sensor.

TABLE OF CONTENTS

AKNOWLEDGEMENTS

This project would not have been taken place if not for the two great institutions that gave me the opportunity to spend my last year of degree abroad; the Universidad Pontificia Comillas ICAI for providing me the chance to go abroad, and the University of Illinois at Urbana-Champaign for taking me in.

I would like to express my sincere gratitude to Prof. Xiaogang Chen and T.A Bryce Smith for their constant guidance and support in the development of this project.

I would like to thank my professors at ICAI for showing me the fascinating world of engineering, and my friends with whom I shared joy and struggles in pursuing our dreams.

Finally, but most importantly, I would like to thank my parents, for their unconditional love and constant support.

CHAPTER 1

INTRODUCTION

Water is an essential resource only second to oxygen to sustain life. One can survive weeks without food, but only a few days with no drinkable water. Almost 70 % of the earth surface is covered by water. However, only 2.5 % is fresh water, while the rest is made of salt water [1]. Although it is called fresh water, it contains all kinds of impurities and bacteria that need to be got rid of before it can be safely consumed.

While people in developed countries have potable tap water delivered to them through a complex pipeline network, or buy bottled water and all kinds of beverages to satisfy their thirst, people in underdeveloped and developing countries do not even have easy access to fresh water.

According to Solar Chapter, a non-profit organization committed to provide help in remote areas in Indonesia, many people living in rural areas such as Umutnana must walk several kilometers a day to obtain water. To help solve this issue, they decided to build a solar powered water pump with the help of the locals. Now, people in Umatnana have fresh water delivered to their doorstep. However, this water must be purified before it could be consumed.

The villagers spend about two hours a day boiling water using firewood as fuel. The resulting smoke is a major concern. Wood smoke is mainly made of fine particles that settle in the lung and narrow airways when breathed in. This can cause long term effects such as decreased lung functions and chronic bronchitis [2]. To remove the need of using firewood, it has been decided to build a water purification system that relies on solar energy.

1.1 Initial Concept

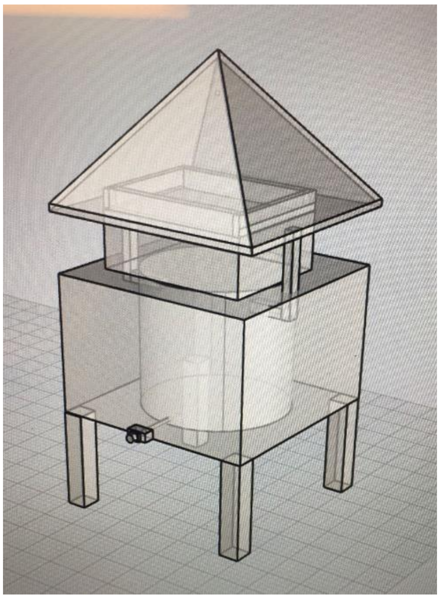


Fig. 1.1 Physical Design

Initially, the idea was to build a distillation system that makes direct use of solar radiation to heat and evaporate water. The top pyramidal structure in Fig. 1.1 would have two sides made of glass that allow solar rays to pass through. The incident sunlight would evaporate feed water in the square container. The other two sides of the pyramid would be of an opaque material that reflect sun light. This makes their interior surface temperature to be low enough to allow water vapor to condense when in contact with them. The condensed water drops would flow down and collected. Since the rate of evaporation using only sunlight is not high enough, an immersed heating mechanism would be used to further heat the water. The system would be powered by a 12 V lead-acid battery of high capacity charged using photovoltaic cells. The reason for using a battery is to offer heating when there is no sunlight. However, this was not a feasible approach considering the amount of energy required to heat water.

The amount of energy required to heat a certain amount of water to a specific temperature can be found by the following expression,

$$Q = m * c * \Delta T \qquad (1.1)[3]$$

The energy required to increase by 30 ºC, 10 litters of water would be 1.26 J. A 12 V 60 Ah battery can provide up to 2.6 J. Therefore, it can only heat 10 litters of water up to 30 ºC only twice when fully charged. Considering the amount of clean water that needs to be produced daily, one would require the battery to fully charge in a short period. This translates into a very high current requirement and robust equipment to support it, which is not realistic.

1.2 Final Concept

Instead of using distillation as the purification method, a more feasible approach would be using different layers of filtration membranes to remove the impurities, such as reverse osmosis and nanofiltration membranes. This approach does not require high power requirement as it only need a booster pump capable of producing enough pressure to push feed water through the filters. An average 12 V DC booster pump draws at maximum 3 A and it can drive pressures up to 100 psi.

Thanks to the low power requirement of a water filtration system, a single 100 W solar panel is enough to power the whole system. The system will include a 12 V lead-acid battery charged by the solar panel. This will allow the system to keep working when there is no sunlight.

The final proposal is to implement a water vending system. By charging an affordable fee for water expended, it ensures people will not exploit the system. The collected money will be used for maintenance and change of filters.

CHAPTER 2

STATE OF ART

2.1 Filtration Membranes

Membrane technology uses semi-permeable materials to separate desirable substances from unwanted particles. Since the separation process does not involve the application of heat, it requires less power than conventional separation processes such as distillation. Depending on the pore size of the membrane, they can be classified as microfiltration (MF), ultrafiltration (UF), nanofiltration (NF), and reverse osmosis (RO)[4].

MF membranes have pore sizes around 0.1 to 10 μm. The separation process is not as simple as allowing particles with sizes smaller than pores to flow through, while retaining the bigger size ones. Since particles are adsorbed onto the pore surface, it reduces the pore size significantly. Therefore, the particles that are separated are much smaller than pore size. MF are used to separate large suspended solids such as bacteria, and colloids from low molecular weight molecules. This type of membranes can be made from both organic materials such as polymers and inorganic materials such as ceramic or aluminum. They are primarily used in tertiary water treatments for potable water production [4].

UF membranes’ pore size ranges between 1 to 100 nm. Like MF membranes, UF are porous, but with an asymmetric structure, smaller pore size and less porosity in the surface. The separation of UF membranes depend on pore size and concentration and size of the macromolecule. There should be at least a 10 times difference in size between the particles to be separated. These membranes are used to remove heavy metals, oil and grease from wastewater, as well as virus removal [4].

RO is a process that separates dissolved solutes from solution when enough pressure is applied at the feed side. NF membranes, or loose RO, have pore sizes between 1-10 nm. Used mostly in groundwater treatment to remove natural organic matter, viruses, and bacteria. Like RO, these membranes can be used to desalinate blackish waters, but they are cheaper to operate and more durable. Unlike Ro, NF membranes can retain essential nutrients present in the water [5].

2.2 Solar Panel

A solar panel is a combination of photovoltaic cells. These cells are electric devices that generate electricity when exposed to light. This is possible because they are composed of two different types of semiconductors, a n-type and a p-type.

P-type semiconductors have a larger hole to electron ratio, while n-type semiconductors have higher electron concentration than hole concentration. When joined together, they form a p-n junction. In the junction, an electric field is formed as holes move to the negative n-side and electrons move to the positive p-side. When the correct wavelength is absorbed by the solar cell, it creates an electron-hole pair. Current flow occurs when the hole is swept across the junction, while the electron completes the external circuit.

An ideal solar cell can be modelled as a current source with a parallel diode, but a more realistic representation is as shown in fig. 2.1.

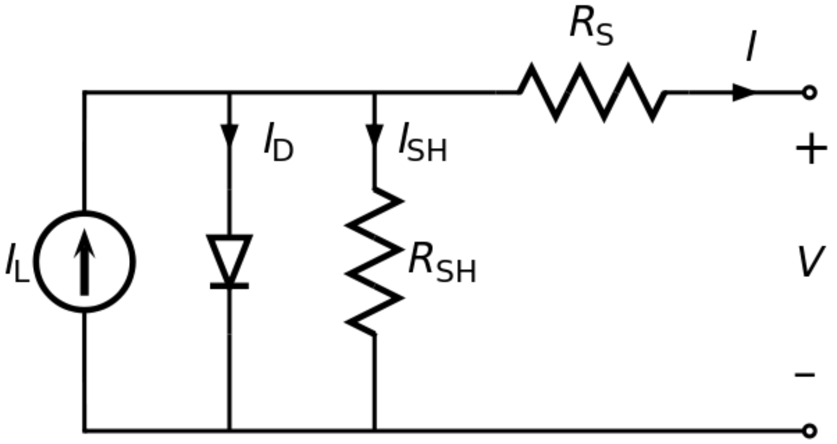


Fig. 2.1 Equivalent circuit of a solar cell

Single solar cells have a very small open-circuit voltage and short-circuit current. Voltage can be increased by connecting the cells in series, while higher currents can be achieved by connecting the cells in parallel. The I-V characteristics of a solar cell is as shown in fig. 2.2. At constant temperature, the current is proportional to the irradiance level. If incident irradiance doubles, the current the cell can output doubles, while the maximum power voltage experience a slight increase. At constant solar irradiance, increasing temperature decreases voltage, resulting in decreased power output. Therefore, to make the most of solar panels, they must be positioned such that they can receive the maximum irradiance level, while being cooled down at the same time.

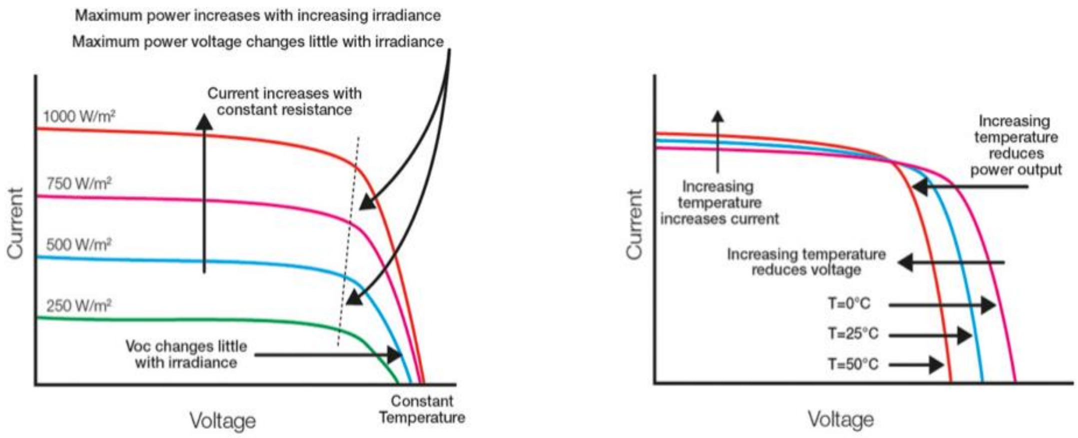


Fig. 2.2 I-V characteristics of a solar cell

2.3 Voltage Regulator

Solar panels can be used to charge a battery. However, batteries must be charged at correct voltage levels to avoid damage. As seen in fig. 2.2, output values of a solar panel are highly reliant on ambient conditions. Hence, it is not safe to connect a solar panel directly to a battery. Instead, voltage regulators can be used to stabilize the output voltage of solar panels.

2.3.1 Linear Regulator

Linear regulators are used to step down higher input voltages to lower output voltages regardless of load current, assuming that they are within the specified range of operation. Simple linear regulators include shunt regulators and series regulators.

Shunt regulators make use of a simple Zener diode as reference. They are used in low-powered circuits where the amount of current is so small that the powered loss is of no concern. This type of regulators is not suitable for voltage conversion applications due to the inefficient Zener diode. The circuit is shown in fig. 2.3, where R_1 is calculated as,

$$R_1 = \frac{V_S - V_Z}{I_Z + I_{R2}} \tag{2.1)[6}$$

And the power dissipated by the Zener is,

$$P_d = \left(\frac{V_S - V_Z}{R_1} - I_{R2}\right) * V_Z \tag{2.2)[6}$$

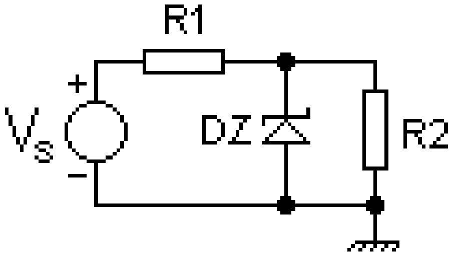


Fig. 2.3 Diagram of simple shunt regulator

Simple series regulators make use of single pass transistor and a Zener diode. They are more efficient than shunt regulators because they do not draw the full current when the load does not require any. Fig, 2.4 shows a simple emitter follower voltage regulator. The output voltage will be as shown in eq. (2.3), where V_{BE} is the voltage drop of the transistor. Since the base current $I_B = \frac{I_E}{\beta+1}$ is a small value, it will have no effect on V_Z with load current changes.

$$V_{out} = V_Z - V_{BE} \tag{2.3)[6}$$

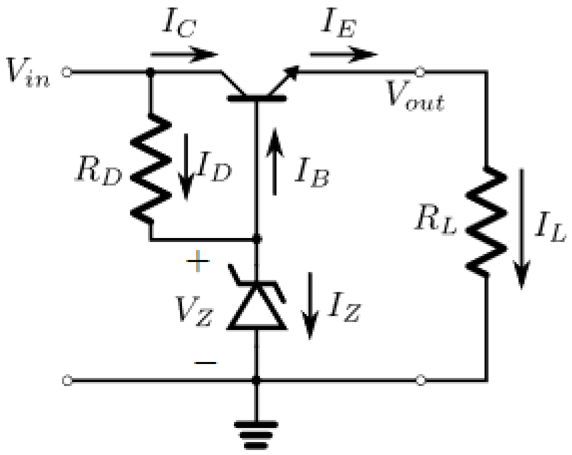


Fig. 2.4 Diagram of simple series voltage regulator

2.3.2 Switching Regulator

Switching regulators are highly efficient because the series components are either conducting, or switched off, and power dissipation is almost negligible. The transistor switches with frequency f_{sw}, or period T_{sw}. It will be on for DT seconds, and off for $(1-D)T$ seconds. D is the duty ratio, which value goes between (0,1).

Unlike linear regulators, switching regulators can either step-up the existing input voltage (Boost Converter) or decrease the input voltage (Buck converter). Some switching regulators have both functionalities such as Buck-Boost converter.

Buck converters are used to step-down higher voltages to lower values. Its circuit schematics is shown in fig. 2.5. Fig. 2.6 shows the circuit based on the switching state of the transistor.

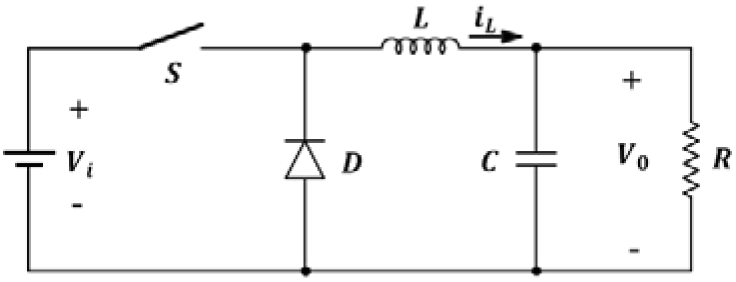


Fig. 2.5 Buck converter

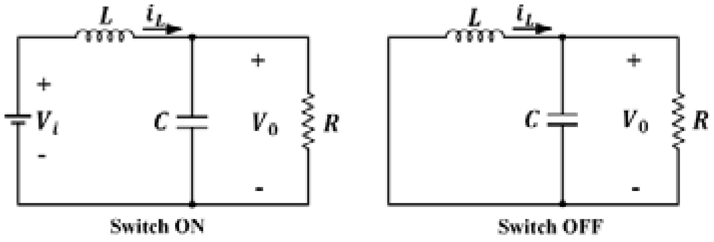


Fig. 2.6 Buck converter switching states

When switch is on, the voltage across the inductor,

$$V_L = V_i - V_o \qquad (2.4)[7]$$

When switched off,

$$V_L = -V_o \qquad (2.5)[7]$$

Since the average voltage across an inductor is 0, then,

$$(V_i - V_o)DT - V_o(1 - D)T = 0 \qquad (2.6)[7]$$

Therefore,

$$V_o = DV_i \qquad (2.7)[7]$$

Since energy is conserved,

$$I_o = \frac{I_i}{D} = I_L \qquad (2.7)[7]$$

Boost converters are dc-dc converters with output voltage greater than the input voltage. Its circuit diagram is shown in fig. 2.7.

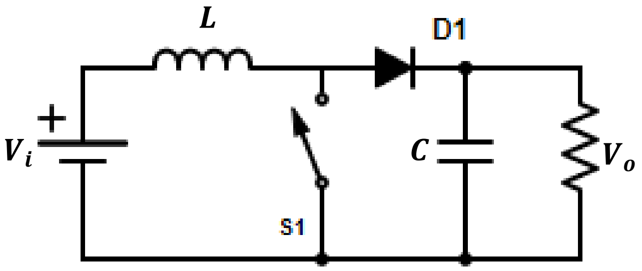


Fig. 2.7 Boost converter

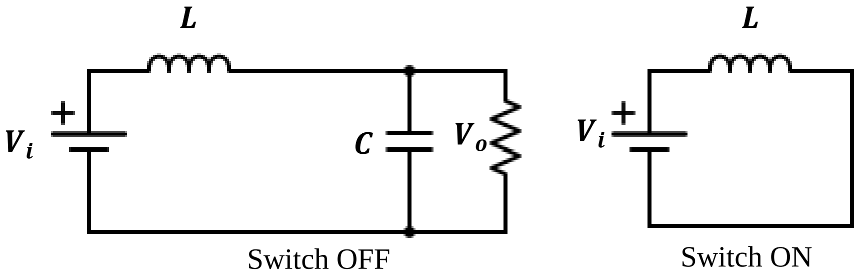


Fig. 2.8 Boost converter switching states

When switch is on, the voltage across the inductor,

$$V_L = V_i \quad (2.8)[7]$$

When switched off,

$$V_L = V_i - V_o \quad (2.9)[7]$$

Since the average voltage across an inductor is 0, then,

$$V_i DT + (V_i - V_o)(1 - D)T = 0 \quad (2.10)[7]$$

Therefore,

$$V_o = \frac{V_i}{1-D} \quad (2.11)[7]$$

2.4 Lead-Acid Battery

Invented in 1859 by Gaston Planté, it was the first commercialized rechargeable battery. Although it is a fairly old technology, it is still widely used nowadays for two main reasons. It is reliable and inexpensive on a cost-per-watt base.

Lead-acid batteries must be charged at the correct voltage levels. High voltages improve battery performance, but it shortens the lifespan because of grid corrosion, while low voltages cause sulfuration on the negative plate. Only when permanently charging it at the floating charge does not cause any damage. This is thanks to the self-discharge nature of the battery. The recommended float voltage is 2.25 to 2.3 V per cell. For a 12 V lead-acid battery, the float voltages would range from 13.5 to 13.8 V [8].

CHAPTER 3

DESIGN

This project consists in the design and implementation of a solar powered water filtration and vending system. The goal is to use a single 100 W solar panel to charge a 12 V lead-acid battery, while powering the rest of the components necessary for the dispensing. The charged 12 V battery will provide power whenever the solar panel is not supplying enough energy.

When enough sunlight is incident in a 100 W solar panel, the panel provides voltages higher than those used to charge a 12 V battery. Fig. 3.1 shows the I-V characteristics of a typical 100 W panel. Therefore, a buck converter will be designed for this project.

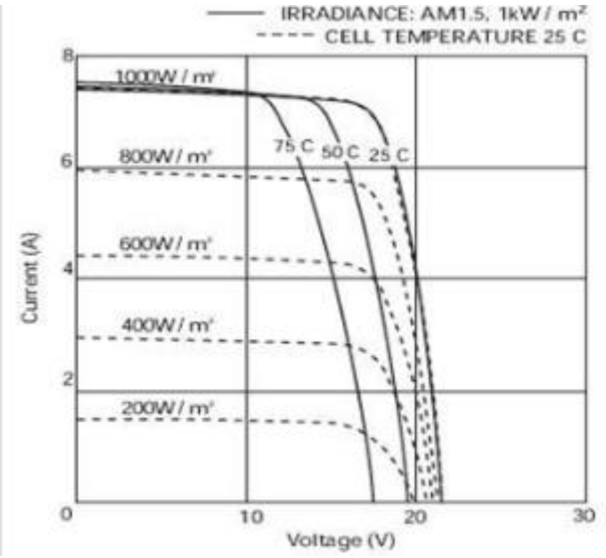


Fig. 3.1 I-V plot of 100 W solar panel

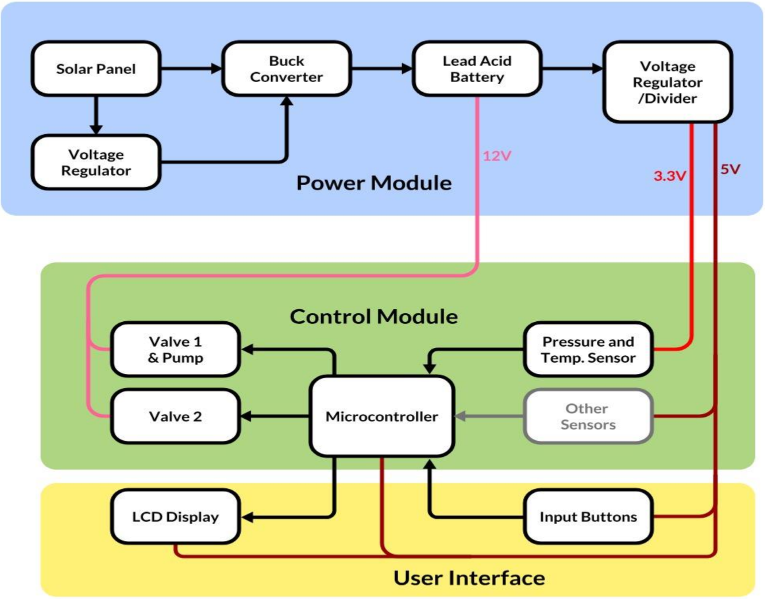


Fig. 3.2 High Level Block Diagram

3.1 Solar panel

As explained in fig. 2.2, the positioning of solar panels is key to obtain the most energy from the sun. The ideal situation is when sunlight is perpendicular to the panel surface, i.e. when $\theta = 0$ from fig. 3.3. To do so, a 2-axis solar tracking solar panel would be the choice. However, for this project, a fixed oriented solar panel will be used.

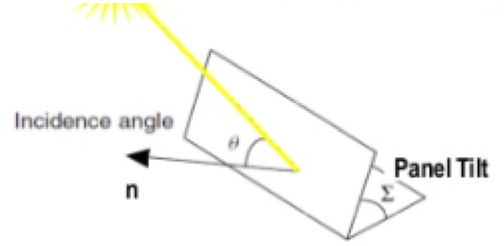


Fig. 3.3 Incident angle θ between a normal n to the collector and incoming solar beam

To determine the tilt angle Σ, it is necessary to find the altitude angle β. For places located in the southern hemisphere such as East Nusa Terengganu, the sun path is shown in fig.

3.5. Hence, the solar panel should be facing north to capture the most sunlight throughout the day.

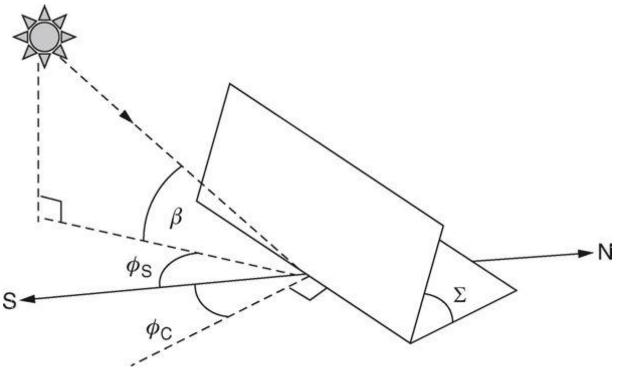


Fig. 3.4 Illustration of collector azimuth angle ϕ_c, tilt angle Σ, solar azimuth angle ϕ_s and altitude angle β

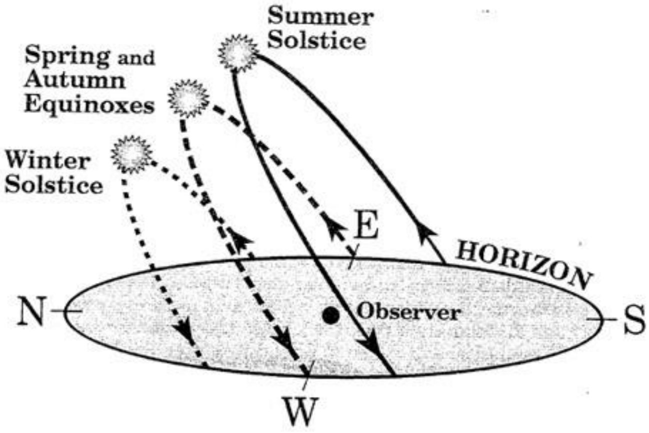


Fig. 3.5 Sun trajectory

The sun is shining the brightest during solar noon, therefore, it is logical to place the panels at an angle $\Sigma = 90 - \beta_N$.

To determine β_N, the following equations are used,

$$\beta_N = 90 - L + \delta \quad (3.1)[9]$$

$$\delta = 23.45 \sin\left[\frac{360}{365}(n - 81)\right], \text{ n=day number} \quad (3.2)[9]$$

From eq. (3.2), it can be seen that δ varies between ± 23.45º. Hence, it would be best to change the tilt angle at the beginning of every season.

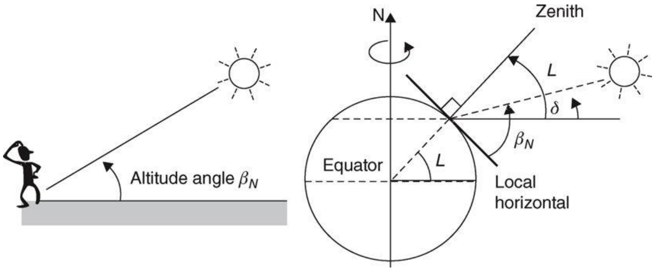


Fig. 3.6 Illustration of altitude angle at solar noon β_N, solar declination δ, and latitude L

3.2 Buck converter

When designing a dc-dc converter, one must take into consideration the efficiency of the converter and the allowable output ripple. An ideal dc-dc converter would provide a constant dc voltage and current at the output. However, the real voltage seen at the output is as shown in fig. 3.5. This triangular wave form is dependent on the switching frequency and size of the energy storing components, i.e. inductor and capacitor. Big sizes of inductor and capacitor, and high switching frequencies would lead to small ripples. However, high switching frequencies and big passive components would result in higher losses.

The proposed requirements for the converter are:

-Ripple less than 1% of the average output value

-Efficiency greater than 80%

-Switching frequency of 50kHz

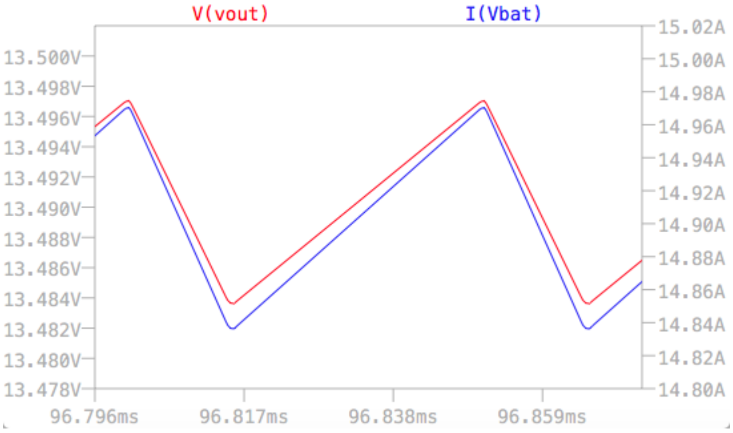


Fig 3.7 Output voltage and current of a Buck converter

3.2.1 Passive component

-Inductor

The voltage across an inductor is expressed as follow,

$$v_L = L\frac{di_L}{dt} \qquad (3.3)[10]$$

As explained in Ch. 2.3, when the transistor is on,

$$V_L = V_i - V_o$$

Therefore, eq. (3.3) can be written as,

$$V_i - V_o = L\frac{\Delta i_L}{DT} \qquad (3.4)[10]$$

where Δi_L is the ripple current. The worst-case scenario is for high input voltages and current. From fig 3.1, the highest output that the panel can supply is roughly 7 A at 18 V. Knowing that 12 V lead-acid battery should be charged between 13.5 - 13.8 V to consider the self-discharge of the battery, the chosen output voltage of the buck converter will be 13.65 V. Knowing this, the resulting duty ratio $D = 0.758$ and $I_L = 9.23\ A$. Hence,

$$L > 0.715\ mH$$

The chosen inductance value is $1.5\ mH$ for safety margins.

An inductor is made of a core with copper wire wounded around it as shown in fig. 3.8. Its theoretical representation consists of a voltage source and two reluctances, one representing the core and the other the gap as seen in fig 3.9.

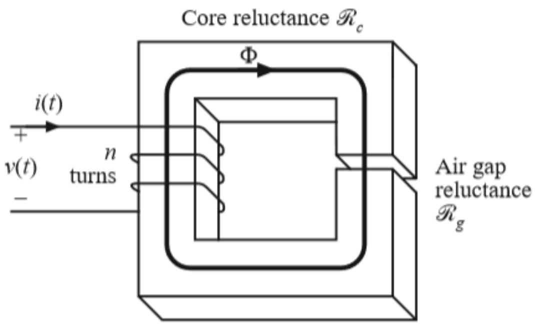

Core reluctance $\mathcal{R}_c$
Φ
$i(t)$
+
$v(t)$
−
n turns
Air gap reluctance $\mathcal{R}_g$

Fig. 3.8 Physical representation of an inductor

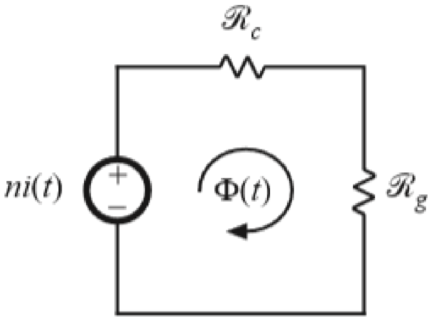


Fig. 3.9 Theoretical representation of an inductor

Solving the magnetic circuit from fig. 3.9,

$$ni = \phi(R_c + R_g) \qquad (3.5)[10]$$

where n is the number of turns, i the winding current, and ϕ is the magnetic flux. Since,

$$R_c \ll R_g$$

$$ni \approx \phi R_g \qquad (3.6)[10]$$

Knowing that $\phi = BA_c$, where B is the magnetic field, and A_c, the core section.

$$ni = BA_cR_g \qquad (3.7)[10]$$

When choosing an inductor, one must consider the maximum allowed current and the core saturation. Hence,

$$nI_{max} = B_{sat}A_cR_g \qquad (3.8)[10]$$

The inductance L can be found using the following equation,

$$L = \frac{n^2}{R_g} \qquad (3.9)[10]$$

Core manufacturers specify the air gap with A_L $(mH/1000)$. Where,

$$A_L = \frac{10B_{sat}^2A_c^2}{LI_{max}^2} \qquad (3.10)[10]$$

Therefore,

$$L = A_L n^2 10^{-9} \ (H) \qquad (3.11)[10]$$

Having chosen the core, the only unknown would be the number of turns,

$$n = \frac{LI_{max}}{B_{sat}A_c} 10^4 \qquad (3.12)[10]$$

By following the above equations, a personalized inductor can be made. However, due to the complexity of making an inductor without the right equipment, a finished product was purchased instead.

-Capacitor

The current across a capacitor can be expressed as follow,

$$i_c = C\frac{dv_c}{dt} \qquad (3.13)[11]$$

When transistor is on, the maximum current flowing into capacitor is $\frac{\Delta i_L}{2}$. Hence,

$$\frac{\Delta i_L}{2} = C\frac{\Delta \mathrm{V_c}}{DT} \qquad (3.14)[11]$$

Knowing that ΔV_c must be less than 1 % of the average output voltage, then,

$$C = 5.13\ \mu F$$

To ensure that the Buck converter will work within the proposed specifications, the selected capacitor will be of 10 μF.

A real capacitor can be represented as a resistor R_{lead} in series with an ideal capacitor in parallel with a resistor R_{leak} as shown in fig. 3.10.

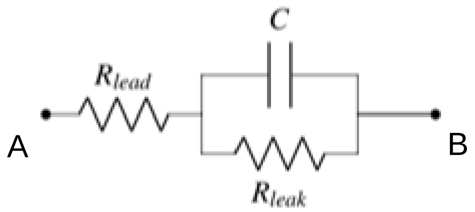


Fig. 3.10 Representation of a real capacitor

The equivalent impedance between A and B is,

$$Z_{AB} = \left(R_{lead} + \frac{1}{w^2C^2R_{leak}}\right) - j\frac{1}{wC} \qquad (3.15)[11]$$

Since, R_{lead} is a negligible value,

$$Z_{AB} = \frac{1}{w^2C^2R_{leak}} = ESR \qquad (3.16)[11]$$

The ESR value will increase the total voltage ripple and dissipate heat. To determine the ESR, the following formula is used,

$$ESR = wCtan\delta \qquad (3.17)[11]$$

where $tan\delta$ is the dissipation factor, and it can be found on the datasheet of the capacitor. For a 10-25V rated capacitor $tan\delta = 0.35$ [12], which gives an ESR value of approximately 1 Ω.

3.2.2 Power Loss

The resulting losses from a dc-dc converter are caused mainly by conduction and switching losses of the power MOSFET transistor and diode. Conduction losses is dependent on the duty cycle, and load current, while switching losses is dependent on switching frequency, and load current. The total losses should include the copper loss from the inductor.

-MOSFET

The conduction loss of MOSFET is,

$$P_{cond,mosfet} = I_L^2DR_{ds,on} \qquad (3.18)[13]$$

where $R_{ds,on}$ is the total resistance between drain and source when the MOSFET is turned on.

The switching losses is caused by the simultaneous presence of drain current and drain-source voltage during each transient as shown in fig 3.11.

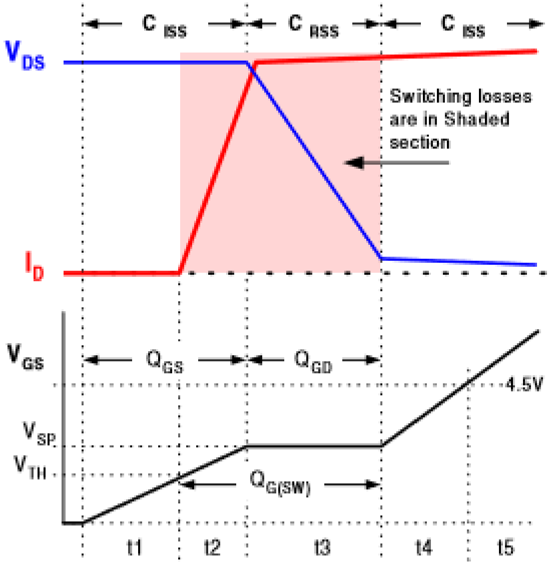


Fig. 3.11 MOSFET switching losses

As illustrated in fig. 3.11, the switching interval is divided into 5 sections. However, the losses only happen during periods t2 and t3, where there is simultaneous presence of voltage and current. The energy loss during each period is approximately the area under the slope. Hence,

$$E_{t2} = t_2 \frac{V_i I_O}{2} \qquad (3.19)[13]$$

$$E_{t3} = t_3 \frac{V_i I_O}{2} \qquad (3.20)[13]$$

Therefore, the total switching loss is,

$$P_{sw} = \frac{1}{2} V_i I_o (t_2 + t_3) f_{sw} \qquad (3.21)[13]$$

<u>-Diode</u>

The conduction losses across the diode is simply the forward voltage drop times the current flowing through is when turned on.

$$P_{cond,diode} = V_F I_d \qquad (3.22)[13]$$

As for the switching losses of the diode, known as reverse recovery losses, occurs when a reverse voltage is applied across the diode. This is caused by the opposite conduction due to the stored charges in the depletion layer of p-n junction. This reverse conduction happens for a time t_{rr}, as illustrated in fig. 3.12

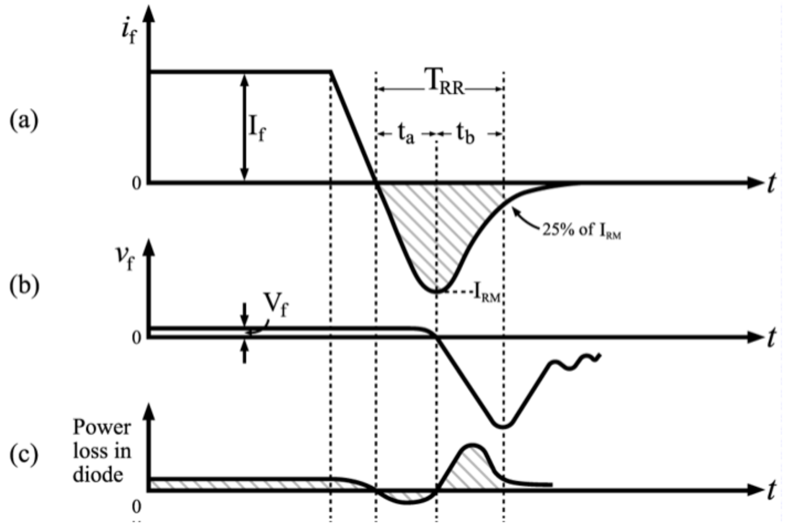


Fig. 3.12 Turn-off characteristics of diode

The reverse recovery loss can be approximated by the following equation,

$$P_{rr} = f_{sw}V_iQ_{rr} \tag{3.23)[13}$$

where Q_{rr} is the shaded area from fig 3.12 a.

-Passive components

A real inductor can be represented as an inductance L and a resistor as shown in fig 3.13.

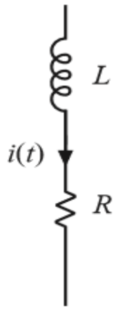


Fig. 3.13 Real inductor representation

The power dissipated by an inductor comes from the copper,

$$P_{cu} = I_L^2 R \tag{3.24}$$

$$R = \rho \frac{l_b}{A_w} \tag{3.25}$$

where ρ is the resistivity of the copper, l_b is the length of the wire, and A_W is the wire section area.

-Total loss

The chosen components for the buck converter their respective characteristics for power loss calculation are:

IRF510 Power MOSFET

- $t_f + t_r = 87\ ns$ [14], $P_{sw} = 0.361\ W$
- $R_{ds,on} = 0.077\ \Omega$ [14], $P_{cond} = 4.975\ W$

DSSK 80-0025 Schottky diode

- No reverse recovery loss, $P_{rr} = 0\ W$
- $V_F = 0.39\ V$ [15], $P_c = 3.6\ W$

B82726S2163N0 Inductor

- $R = 7.1\ m\Omega$, [16] $P_{cu} = 0.605\ W$

The total power dissipated is,

$$P_{tot} = 9.542\ W$$

Since the solar panel provide 100 W, the total efficiency is about 90 %.

3.2.3 PWM generator

The switching frequency and duty ratio is determined by the PWM signal. To control the buck converter output, it is necessary to control the duty ratio by feeding the output voltage to a feedback control mechanism.

-Triangular wave generator

A PWM signal can be generated by comparing a triangular wave with a controlled voltage to generate the desired PWM, as illustrated in fig 3.14. It can be done analogically using a comparator as shown in fig 3.15.

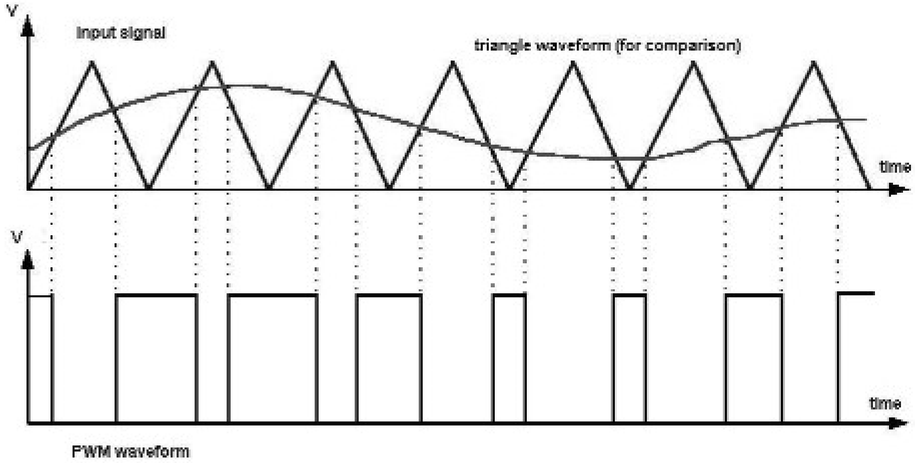
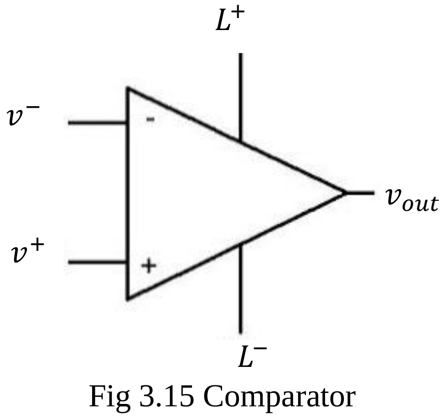


Fig. 3.14 PWM signal generation



$$v_{out} = L^+, if\ v^+ > v^-$$

$$v_{out} = L^-, if\ v^+ < v^-$$

Fig 3.15 Comparator

The triangular wave is generated by using a non-inverting Schmitt trigger to create a square wave, and integrating it using an integrator ramp generator. The resulting triangular wave will be fed back to the non-inverting Schmitt trigger as shown in fig. 3.16 to obtain the square wave.

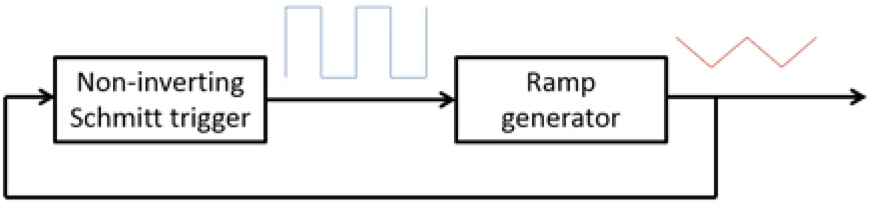


Fig. 3.16 Block diagram for generating triangular wave

A non-inverting Schmitt trigger is a comparator with hysteresis with positive feedback. Fig. 3.17 illustrates its circuit diagram. Just like a comparator, its output will oscillate between the supply voltages of the comparator depending on the input voltages.

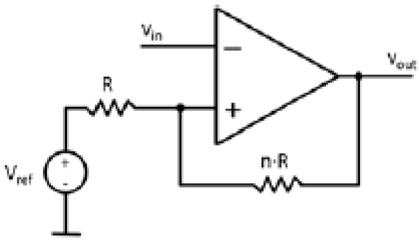


Fig. 3.17 Non-inverting Schmitt trigger

Solving the circuit,

$$v^{+} = V_{ref}\frac{nR}{R+nR} + v_{out}\frac{R}{R+nR} = V_{ref}\frac{n}{n+1} + v_{out}\frac{1}{n+1} \qquad (3.26)[17]$$

If initially $v_{out} = L^{+}$, because $v^{+} > v^{-}$, then,

$$v^{+} = V_{ref}\frac{n}{n+1} + L^{+}\frac{1}{n+1} = V_{u} > v^{-} = v_{in} \qquad (3.27)[17]$$

Hence, if $v_{in} < V_{u}$, then $v_{out} = L^{+}$. This means that V_{U} is the threshold for v_{in} so that v_{out} changes to L^{-}.

If $v_{out} = L^{-}$, because $v^{+} < v^{-}$, then,

$$v^{+} = V_{ref}\frac{n}{n+1} + L^{-}\frac{1}{n+1} = V_{L} > v^{-} = v_{in} \qquad (3.28)[17]$$

Hence, if $v_{in} > V_L$, then $v_{out} = L^-$. This means that V_L is the threshold for v_{in} so that v_{out} changes to L^+.

Knowing that $V_U \neq V_L$, such that $V_U > V_L$, then, the resulting curve is as shown in fig. 3.18.

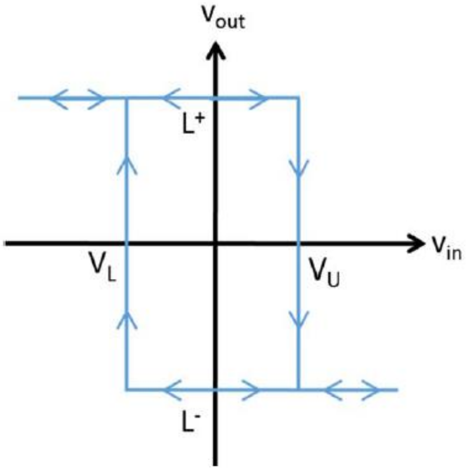


Fig. 3.18 Non-inverting Schmitt trigger output-input curve

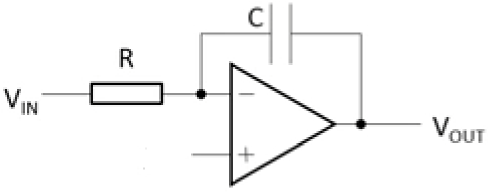


Fig 3.19 Integrator ramp generator

Using the integrator ramp generator, a triangular wave can be generated from a square wave. When a changing input signal is fed into an integrator, the capacitor will charge and discharge in response at a constant rate as shown in fig. 3.20.

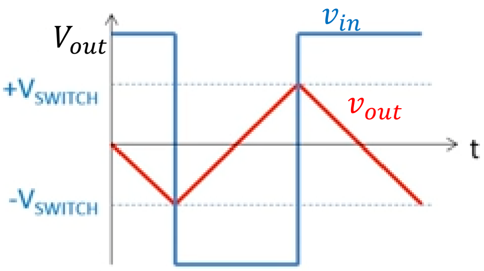


Fig 3.20 Output curve of an integrator

For this project, the idea is to generate a triangular wave with mean value in 2.5 V, amplitude as close to 5 V as possible for maximum resolution, and frequency 50kHz.

To obtain an amplitude of 5 V, the op-amps will be supplied between 5 V and GND. Since most op-amps cannot provide rail-to-rail output voltage, the real amplitude will be below 5 V.

The mean value can be achieved by adding a dc offset of 2.5 V to integrator v^{+} and non-inverting Schmitt trigger v^{-}. Illustrated below is the complete triangular wave generator circuit diagram.

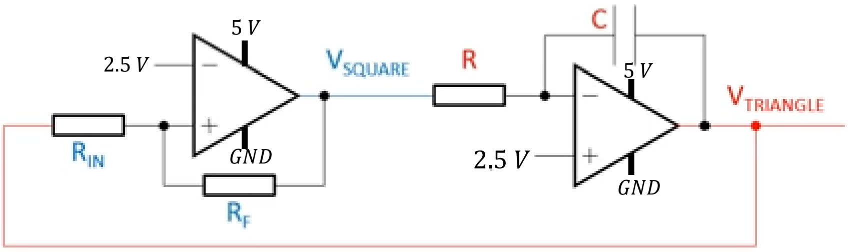


Fig. 3.21 Triangular wave generator

The frequency of the triangular wave can be determined using eq. (3.29).

$$f_{sw} = \frac{R_F}{4R_{IN}} \frac{1}{RC} \tag{3.29}$$ [18]

R_F and R_{IN} relation can be found using eq. (3.27) and eq. (3.28). Ideally, $V_{triangle}$ should go between 0 and 5 V for maximum resolution of triangular wave. When $V_{triangle} = 5\ V$, the comparator output will go from 0 to 5V. When $V_{triangle} = 0\ V$, the comparator output will go from 5 to 0 V. However, due to op-amps limitation, $V_{triangle}$, being the output of an integrator, it can never reach 5 V. For a general-purpose op-amp, the output voltage is capped at around 4.5 V. Hence,

$$2.5 < 4.5\frac{n}{n+1}, n > 1.25$$

The chosen $\frac{R_F}{R_{IN}}$ ratio is 1.5, $R_F = 15\ k\Omega$ and $R_{IN} = 10\ k\Omega$. Using eq. (3.29), the obtained values of R and C are $20\ k\Omega$ and $0.375\ nF$ respectively.

-PI control

To obtain the PWM signal with the desired duty ratio, the generated triangular wave must be compared to a controlled voltage. This is done by comparing the converter output voltage to a reference voltage to generate an error as illustrated in fig 3.22. This error will be corrected using a control mechanism such as PI control.

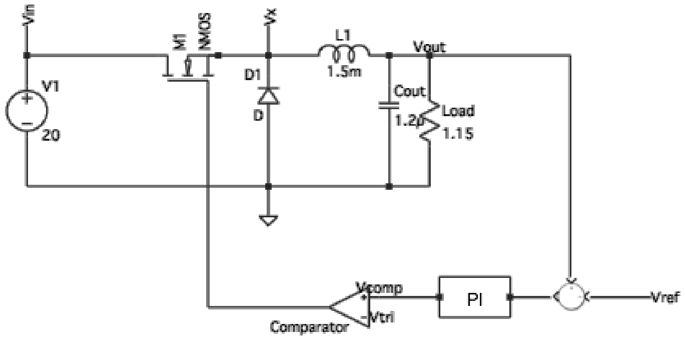


Fig. 3.22 Control concept

The error will be generated by using a differential amplifier as shown in fig. 3.23.

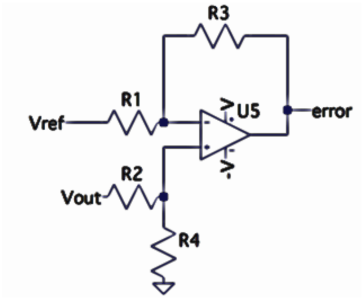


Fig 3.23 Circuit diagram of differential amplifier

The output of a differential amplifier can be found using the following equation,

$$error = V_{out}\frac{R_4}{R_2+R_4}\frac{R_1+R_3}{R_1} - V_{ref}\frac{R_3}{R_1} \qquad (3.30)[18]$$

When,

$$R1 = R2 = R3 = R4,\ error = V_{out} - V_{ref} \qquad (3.31)[18]$$

The resulting error will be fed into the PI (fig. 3.24) to be corrected.

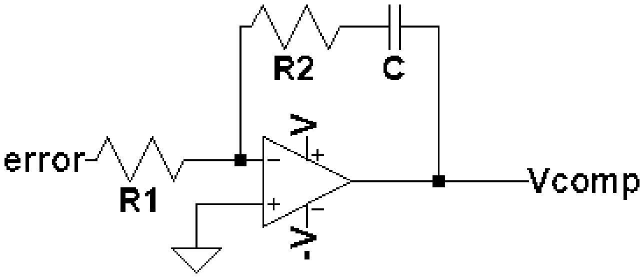


Fig. 3.24 Analog implementation of PI control

The transfer function of the PI control is as follow,

$$\frac{V_{comp}}{error} = -\frac{R_2}{R_1}\left(1 + \frac{1}{R_2 CS}\right)$$

where $\frac{R_2}{R_1}$ is the proportional gain and $\frac{1}{R_1 C}$ the integral gain.

3.3 Dispense Control

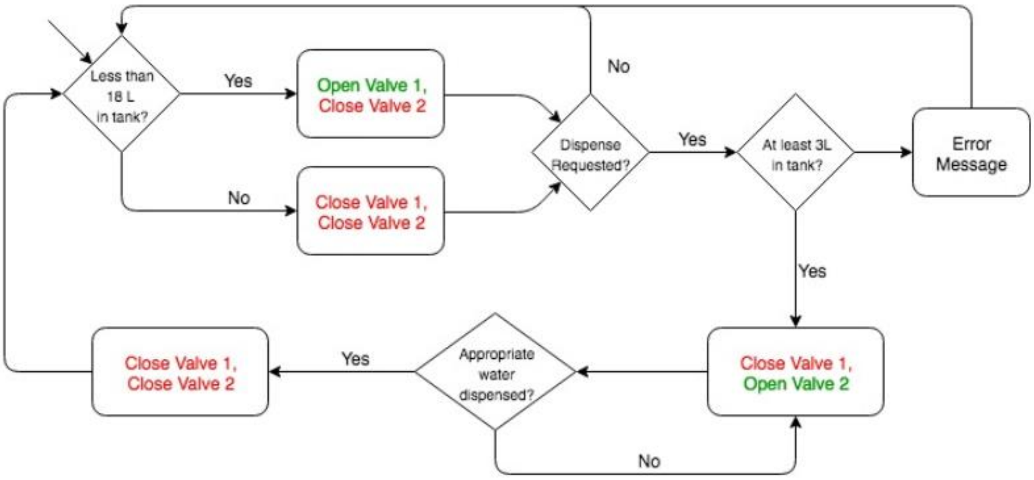

Less than 18 L in tank?
Yes
Open Valve 1, Close Valve 2
No
Close Valve 1, Close Valve 2
No
Dispense Requested?
Yes
At least 3L in tank?
Error Message
Yes
Close Valve 1, Open Valve 2
Appropriate water dispensed?
Yes
Close Valve 1, Close Valve 2
No

Fig. 3.25 Control logic flowchart

The dispensing system is controlled using the ATMega328-pu microcontroller. The code is written in Arduino IDE and uploaded to the chip using an Arduino board.

The microcontroller is connected to a MS5803 pressure sensor to determine water volume in the tank. This sensor is calibrated to measure absolute pressure. Using eq. (3.32), the water level h can be determined.

$$P_{abs} = P_{atm} + \rho g h \tag{3.32}$$

By using a cylindrical tank of uniform radios, the pressure sensor can be used to measure volume.

$$V = \pi R^2 h \tag{3.33}$$

The water flow is controlled using two electric valves. These valves are driven by activating a BJT switch. Its circuit schematics is illustrated in fig. 3.26.

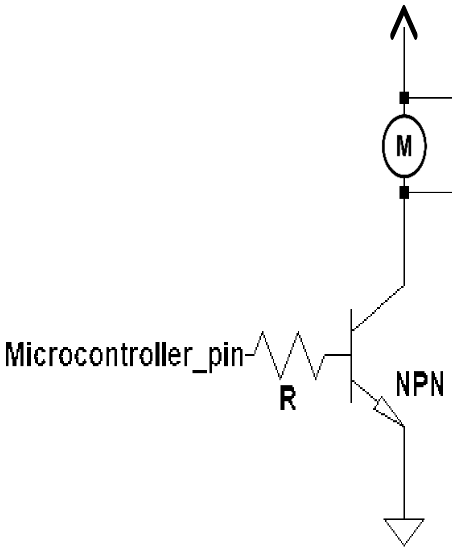


Fig. 3.26 Valve and booster pump switching mechanism

The microcontroller will activate the input valve to allow filtered water to flow in whenever the tank is below 90 % of its capacity and close it otherwise. The output valve is triggered when a confirmation button is pressed after a specific amount of water is selected. The water of volume to dispense is selected by the user through 3 input buttons, which corresponds to 1, 2, and 3 L respectively. These buttons can only be pressed when the tank contains at least 3 L of water. If its capacity is below 3 L, an LCD display will show an error message telling the user to wait.

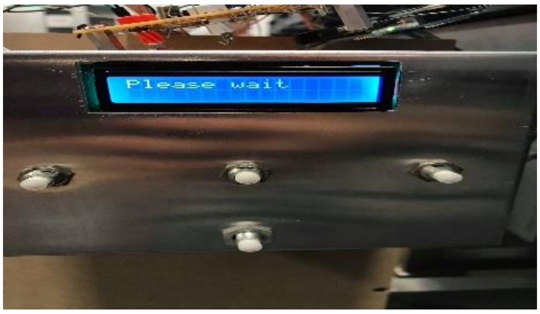


Fig. 3.27 User Interface

When dispensing, the only information that the microcontroller has is the actual volume and the amount to dispense. Knowing that energy is conserved, a relationship between time and volume dispensed can be derived.

When output valve opens, the potential energy is converted to kinetic energy,

$$mgh = \frac{1}{2}mv^2 \qquad (3.34)$$

Knowing that the mass flow rate in a system is also conserved,

$$\frac{dm}{dt} = \dot{m_{in}} - \dot{m_{out}} \qquad (3.36)[19]$$

The input flow is 0. Hence,

$$\frac{dm}{dt} = -\dot{m_{out}} \qquad (3.37)\ [19]$$

Since $m = density\ x\ volume$ and $\dot{m} = area\ x\ velocity$,

$$\frac{d\rho V}{dt} = -\rho a v \qquad (3.38)\ [19]$$

where a is the opening are of the output valve. From eq. (3.34), $v = \sqrt{2gh}$. Hence,

$$\frac{dV}{dt} = a\sqrt{2gh} \qquad (3.39)\ [19]$$

Since $volume = area\ x\ height$,

$$\frac{dAh}{dt} = a\sqrt{2gh} \qquad (3.40)[19]$$

Solving for time, eq. (3.42) is obtained.

$$\int dt = \int_{h_1}^{h_2} \frac{A}{a\sqrt{2gh}} dh \qquad (3.41)[19]$$

$$t = \frac{2*A}{a*\sqrt{2*g}} \left(\sqrt{h_1} - \sqrt{h_2}\right) \qquad (3.42)[19]$$

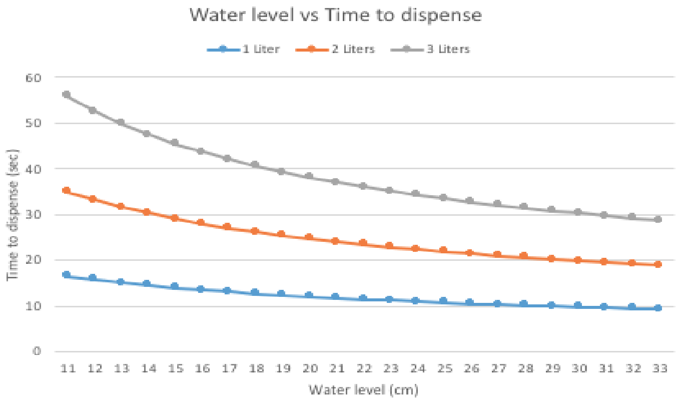


Fig. 3.28 Theoretical curve of time to dispense vs water level before dispensing

The microcontroller requires a supply voltage of 5 V and the system's power supply is a 12 V battery. Hence, a voltage regulator is necessary. Knowing that the microcontroller is a low power component, a linear voltage regulator such as LM7805 can be used without compromising efficiency.

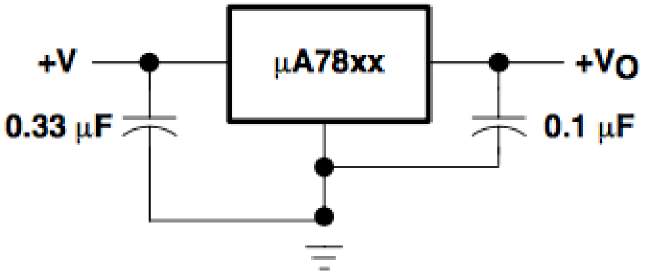


Fig. 3.29 5 V linear regulator

CHAPTER 4

SIMULATION

The simulations were performed using LTSpice XVII.

Fig. 4.1 is the simulation circuit for designed buck converter. It is used to check whether the calculated sizes of passive components fulfill the requirements for ripple.

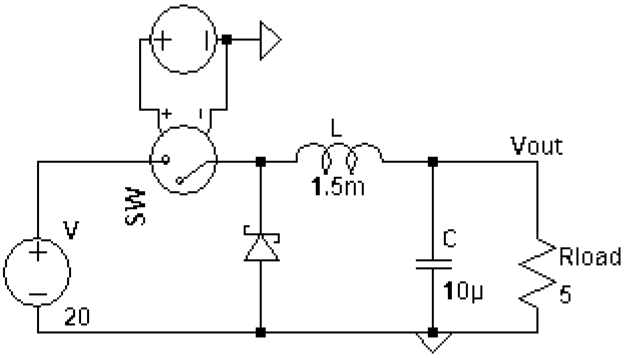


Fig. 4.1 Buck converter simulation circuit

The commands used are:

- PULSE(0 5 0 0 0 13.8u 20u 1000). This command drives the switch at 50kHz with a duty ratio of 0.69 to obtain an average output of 13.8 V with an input of 20 V.
- .model SW SW(Ron=0.077 Roff=1Meg Von=5 Voff=0V). This command sets the resistance of the switch for the on and off state, and the voltage required to turn it on and off.

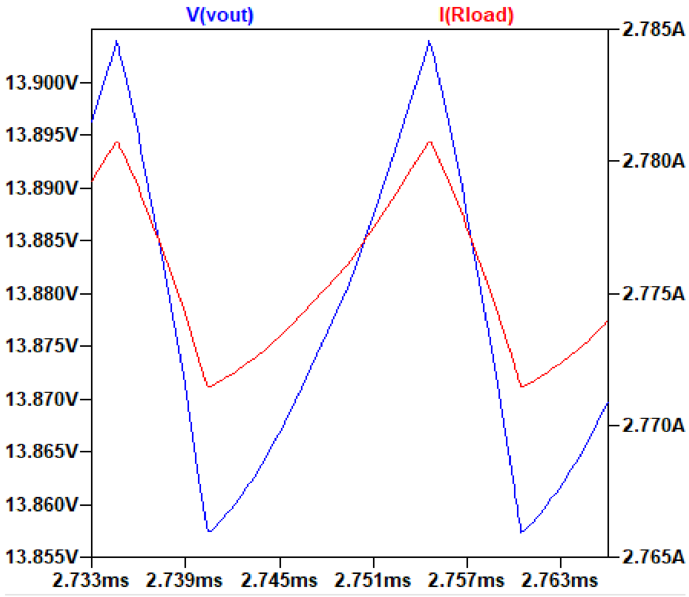


Fig. 4.2 Output voltage and current ripple

From fig. 4.2, it can see that $\Delta v \approx 0.045\ V$ and $\Delta i \approx 10\ mA$. Both are below 1 % of the average value.

Fig. 4.3 illustrates the simulation circuit of the designed triangular wave generator.

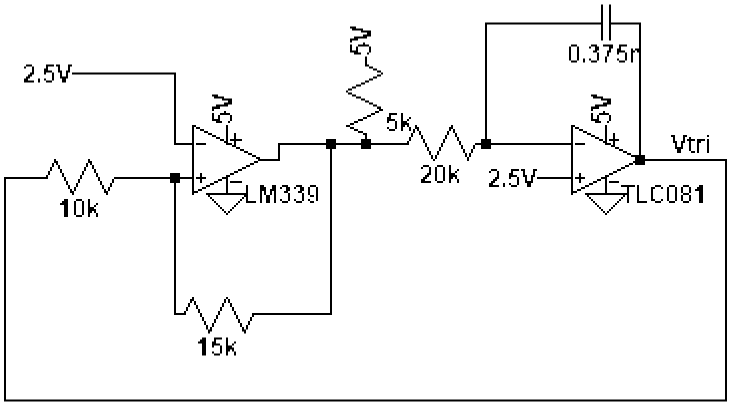


Fig. 4.3 Triangular wave generator

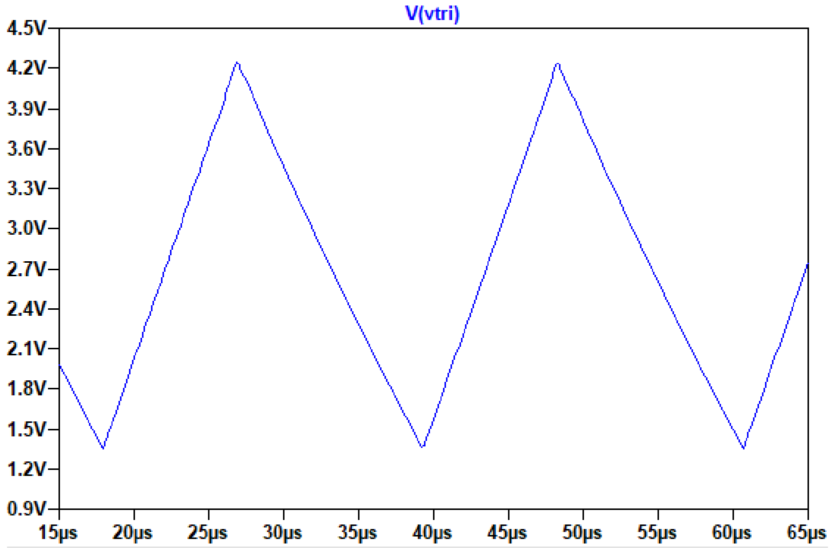
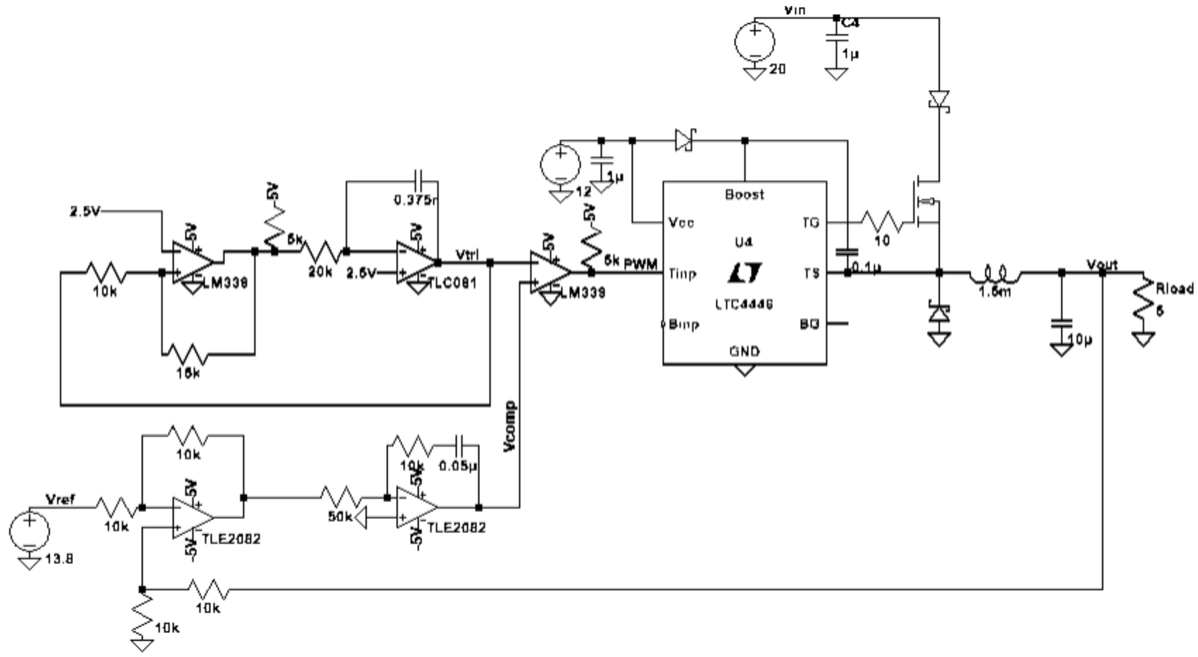


Fig.4.4 Triangular wave generator output

The resulting triangular signal has frequency close to 50 kHz consistent with the calculations. Its amplitude is below 5 V as expected due to the limitations of the op-amps used for simulation.



Fig, 4.5 Complete circuit simulation

Having checked that the triangular wave generator works, the PI feedback loop was added to complete the circuit. A gate driver was included to boost the PWM signal to drive the power MOSFET.

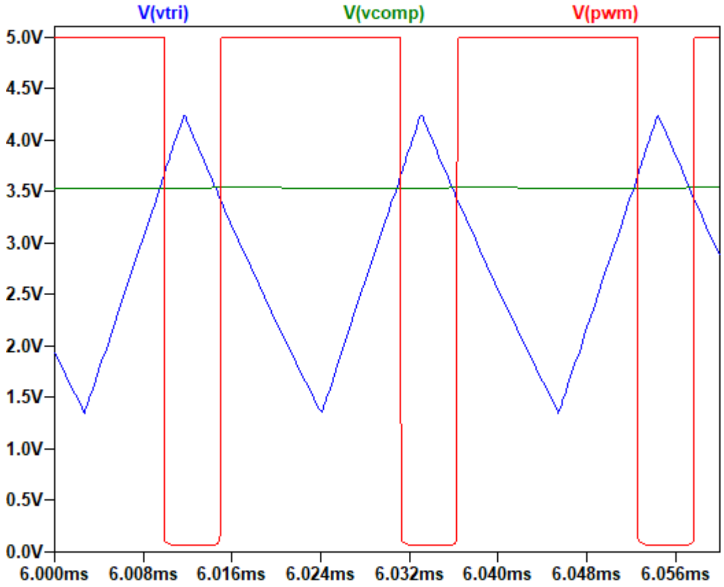


Fig. 4.6 PWM signal

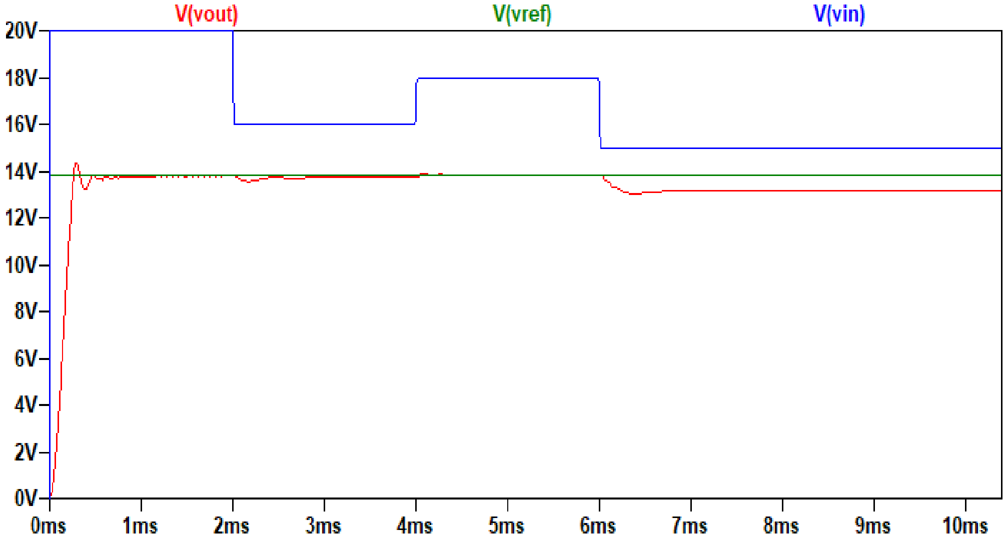


Fig. 4.7 Output response with resistive load

As seen in fig. 4.7, the circuit seems to work when the input voltage is high enough. However, for lower input voltages, the resulting output voltage is lower than the desired reference voltage. This is caused by the output limitations of the op-amp.

From fig. 4.8, it can see that the output from the PI control is capped at 4 V, while the triangular wave has its peak value at 4.2 V (Fig. 4.6). Therefore, the resulting PWM signal from the comparison of both signals will have a maximum duty ratio below 1. The duty cycle's resolution can be increased by reducing the triangular signal's peak value as seen in fig. 4.9, where the desired output voltage is obtained for the same input voltages as those shown in fig. 4.7. Finally, the circuit was tested using a voltage load of 12.8 V to replicate a undercharged battery.

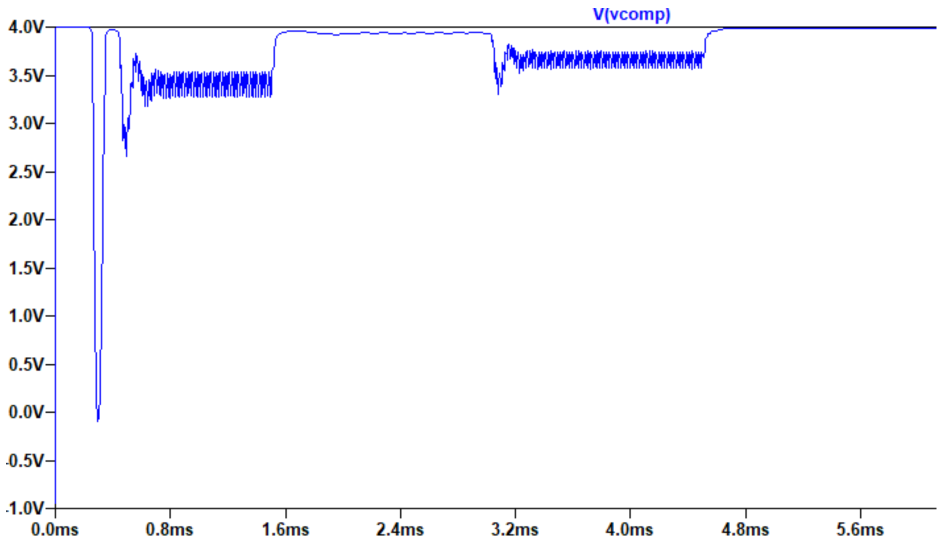


Fig. 4.8 Output of PI control

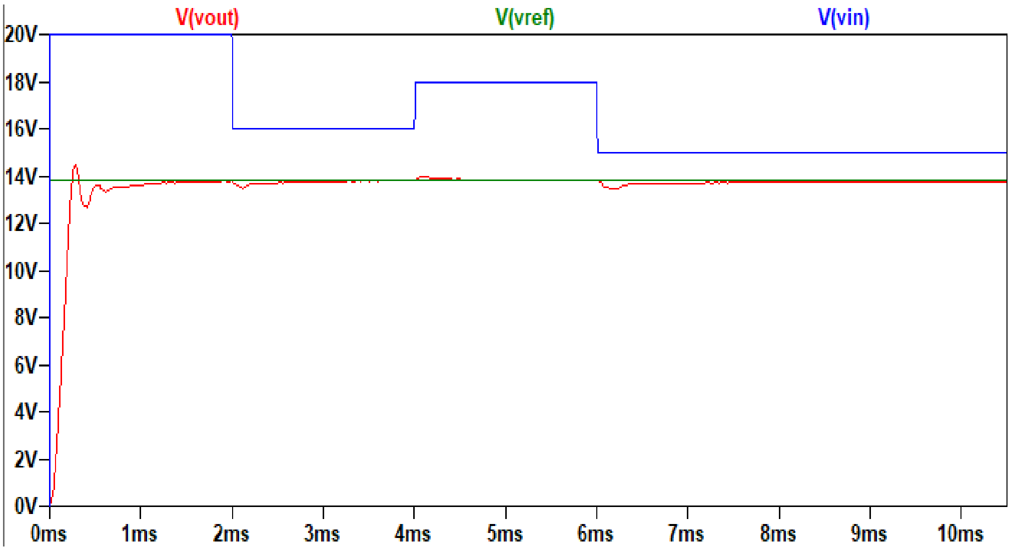


Fig. 4.9 Output response with corrected triangular wave

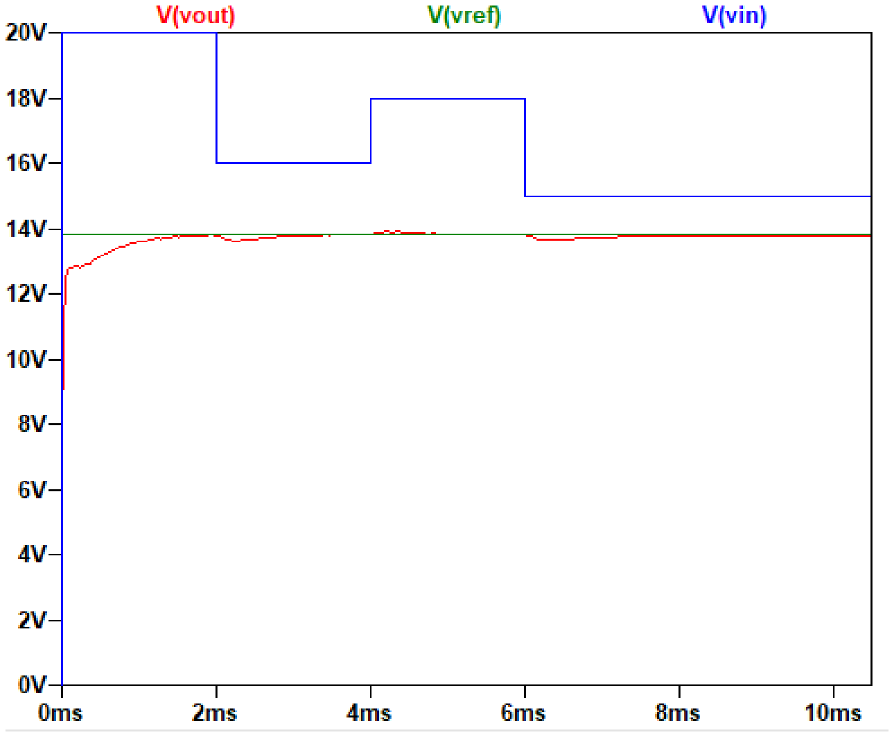


Fig. 4.10 Output response with voltage load

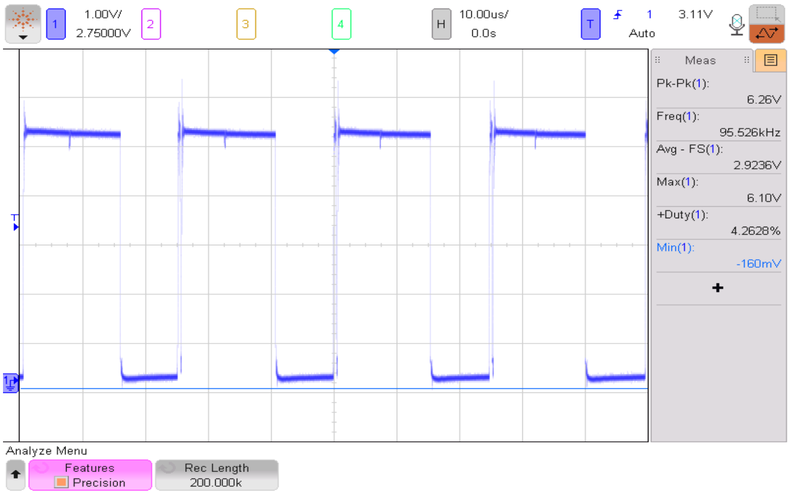
CHAPTER 5

TESTING AND IMPLEMENTATION

5.1 Buck Converter

During breadboard testing, the designed buck converter did not work as expected. Although the reference voltage used was 13.6 V. The output voltage reached its maximum value at 13 V. From the explanation given in Ch.4, low input voltages may not result in desired output voltage since the duty ratio cannot reach its full range due to op-amp limitation. However, during testing, high input voltages were used, but the output voltage was still capped at 13 V. Nevertheless, for references below 13 V, the buck converter could keep the output voltage at the reference point for input voltages above 14.5 V. From this, it can be determined the range of duty cycle is not an issue.

When looking at the PWM signal (fig. 5.1) the measured duty cycle is clearly wrong. This is caused by the double transition observed in the signal. This double transition happens when the comparator changes state because of noise in the comparing signals as illustrated in fig. 5.2. The solution to this problem can be resolved by adding hysteris to the comparator as seen in fig 5.3, where the correct duty cycle is illustrated.



Pk-Pk(1): 6.26V
Freq(1): 95.526kHz
Avg - FS(1): 2.9236V
Max(1): 6.10V
+Duty(1): 4.2628%
Min(1): -160mV

Fig. 5.1 PWM signal with double transition

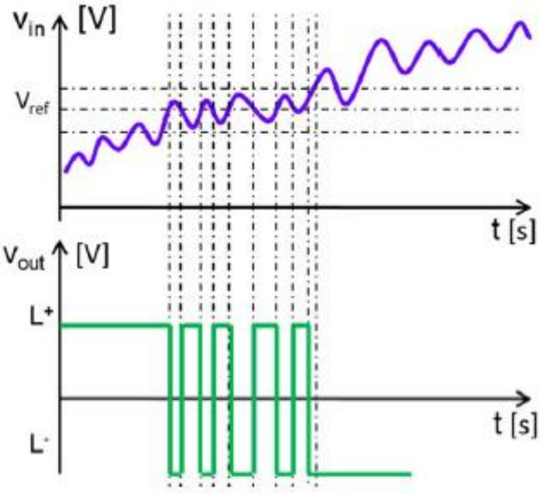
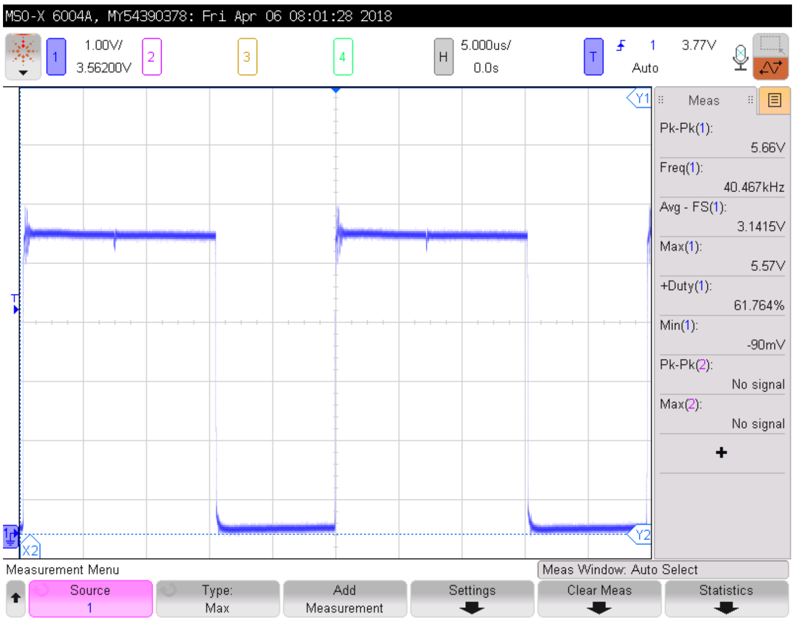


Fig. 5.2 Unwated transition due to noise


MSO-X 6004A, MY54390378: Fri Apr 06 08:01:28 2018

1.00V/
3.56200V

5.000us/
0.0s

1 3.77V
Auto

Pk-Pk(1): 5.66V
Freq(1): 40.467kHz
Avg - FS(1): 3.1415V
Max(1): 5.57V
+Duty(1): 61.764%
Min(1): -90mV
Pk-Pk(2): No signal
Max(2): No signal

Measurement Menu
Meas Window: Auto Select

Source 1
Type: Max

Fig. 5.3 PWM signal with hysteresis

After verifying the designed circuit is functioning as expected, it was mounted on a PCB board. The PCB layout was drawn using Eagle. See Appendix B. The finished Buck converter is shown below.

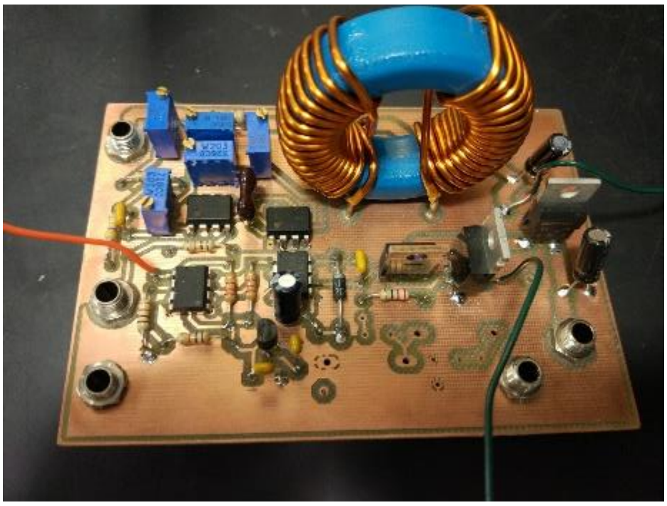
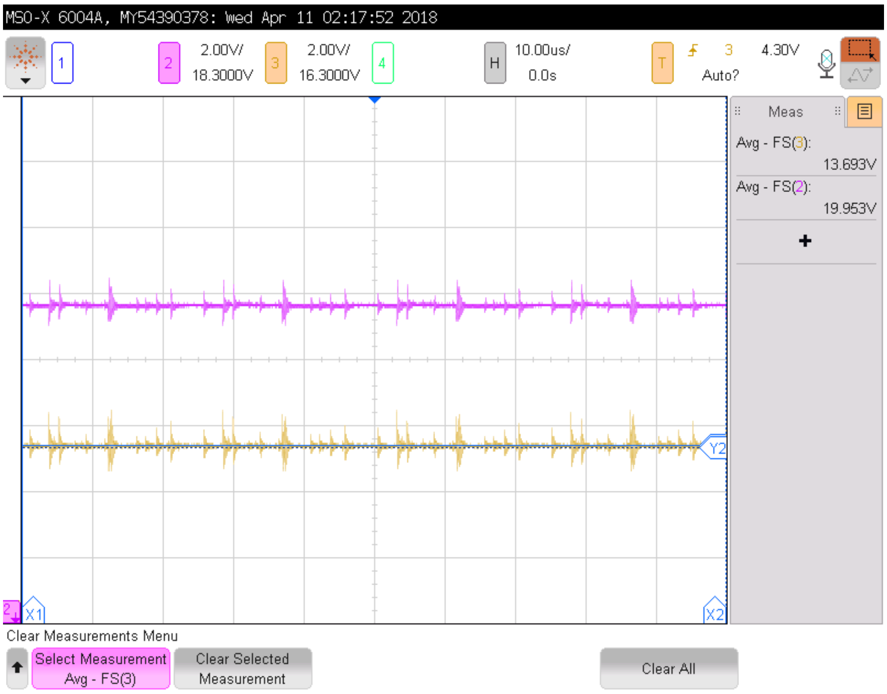


Fig. 5.4 Conversor Buck

The converter was tested by connecting its input to a power supply and measuring its output using an oscilloscope. As it can be seen in fig. 5.5 and 5.6, the converter is capable of stepping down the different input voltages to a stable 13.7 V.



MSO-X 6004A, MY54390378: Wed Apr 11 02:17:52 2018

2 2.00V/ 18.3000V

3 2.00V/ 16.3000V

H 10.00us/ 0.0s

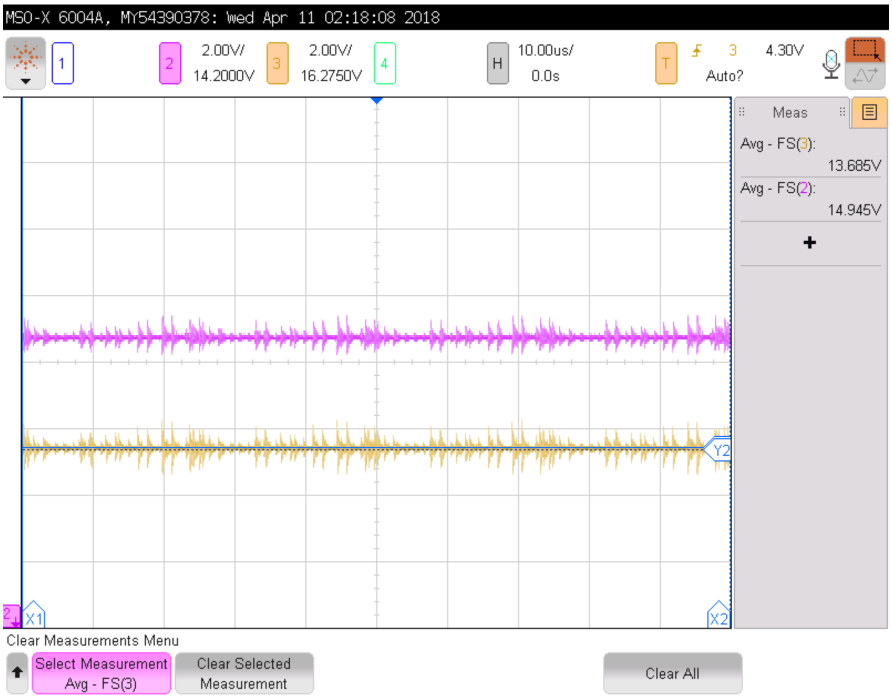
T 3 4.30V Auto?

Meas

Avg - FS(3): 13.693V

Avg - FS(2): 19.953V

Fig. 5.5 Output voltage (yellow signal) at 20 V input voltage



MSO-X 6004A, MY54390378: Wed Apr 11 02:18:08 2018

2 2.00V/ 14.2000V

3 2.00V/ 16.2750V

H 10.00us/ 0.0s

T 3 4.30V Auto?

Meas

Avg - FS(3): 13.685V

Avg - FS(2): 14.945V

Clear Measurements Menu

Fig. 5.6 Output voltage (yellow signal) at 15 V input voltage

To check the efficiency of the buck converter, the input and output voltages and currents were measured. A DC electronic load in voltage mode was used to simulate a 12 V battery. Knowing that a fully charged battery has a voltage of 14 V, the voltage load was set to 12.8 V to reproduce an undercharged battery. Both voltage supply and DC load show the voltage and current values through their screen as illustrated in fig. 5.7.

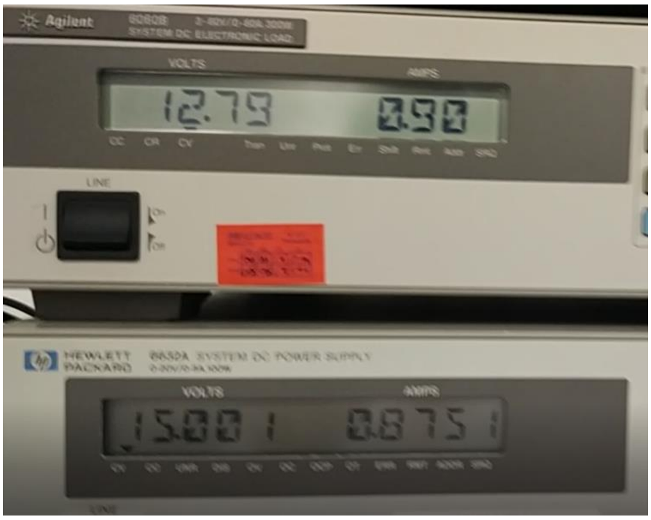


Fig. 5.7

$$\eta = \frac{12.8 * 0.9}{15 * 0.8751} = 0.88$$

5.2 Dispensing system

The key component in the dispensing system is the pressure sensor. The correct dispensing depends on the its precision. To test it, the pressure sensor was placed at the bottom of the water tank and connected to the microcontroller, so that it can send the information to be shown through an LCD display. As it can be seen in fig. 5.9, the sensor provides accurate readings.

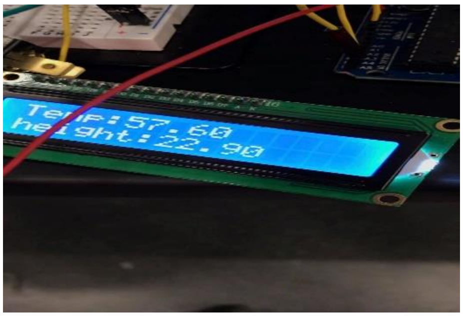
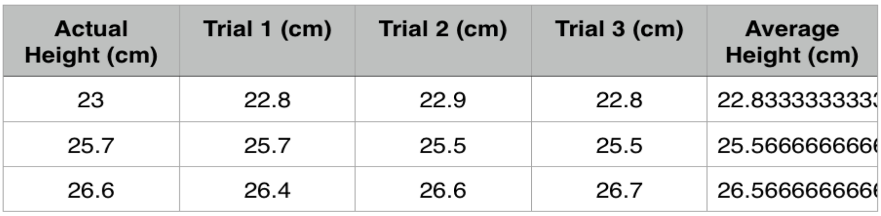


Fig. 5.8 LCD display



Actual Height (cm)	Trial 1 (cm)	Trial 2 (cm)	Trial 3 (cm)	Average Height (cm)
23	22.8	22.9	22.8	22.8333333333
25.7	25.7	25.5	25.5	25.5666666666
26.6	26.4	26.6	26.7	26.5666666666

Fig. 5.9 Pressure sensor test

Due to time constraints, a working PCB board could not be prepared in time for the dispensing system. The components were soldered in a perfboard and connected using jumper wires. The excesive amount of wires made debugging difficult. Nonetheless, the system was able to dispense the correct amount of water chosen through the input buttons.

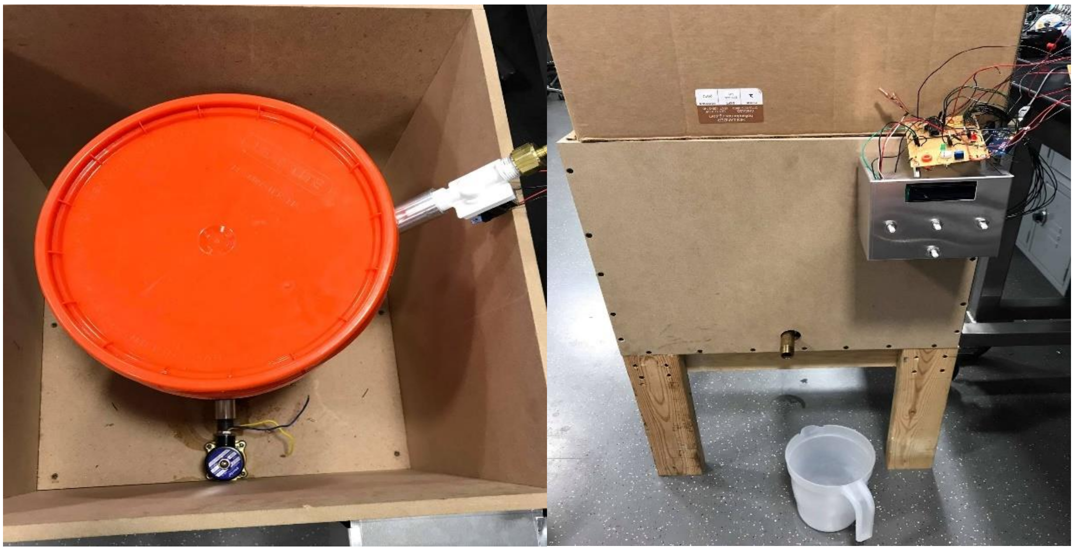


Fig. 5.10 Physical design

CHAPTER 6

CONCLUSION

The objective of this project was to create a water filtration system powered by solar energy to remove the need of using firewood as energy source to sterilize water; reducing the health issues related to the inhalation of wood smoke and diminishing the environmental effects due to the cutting of trees.

To power the system, an efficient buck converter was designed to step-down the unsteady output voltages from the solar panel to a stable floating voltage to charge the 12 V lead-acid battery. However, the functionality of the converter was only tested using a power supply instead of the existing solar panel.

A 100 W solar panel was borrowed. However, it did not perform as expected. The measured open-circuit voltage (V_{oc}) was 40 V instead of the 22 V showed in its specification. As exposure to sunlight increased, V_{oc} kept incrementing, which is contradictory to temperature voltage relationship of PV cells.

The dispensing control was demonstrated successfully. The volume dispensed was within a permissible error margin, considering unaccounted variables such as friction, or delay.

Although the functionality of individual parts of the system was demonstrated successfully, the whole system could not be integrated together since some components such as the battery and the filtration membranes were not ordered. The battery could be simulated using a power supply and digital load, and there was no need to show the filtration capacity of market membranes.

REFERENCES

[1] Kuldeep H. Nayi, Kalpesh V. Modi, “Pyramid solar still: A comprehensive review,” Renewable and Sustainable Energy Reviews, vol. 81, part 1, pp. 136-148.2018, 2018.

[2] Diaz, J. Koff, M. Gotway, S. Nishimura and J. Balmes, "Case Report: A Case of Wood-Smoke–Related Pulmonary Disease", *Environmental Health Perspectives*, vol. 114, no. 5, pp. 759-762, 2006.

[3] E. Huélamo Martínez, “Termodinámica,” Universidad Potificia de Comillas ICAI, 2015.

[4] Singh, R. “Introduction to Membrane,” in *Hybrid Membrane Systems for Water Purification,* Elsevier Science, 2005.

[5] N.M. Momba, “Nanotechnology-based filters for cost-effective drinking water purification in developing countries,” in *Water Purification*, Academic Press, 2017, pp. 184-186.

[6] Hegge, W. Frederick, “A Simple Zener Diode Voltage Regulator,” Journal of the experimental analysis of behavior volume 8, January 1965.

[7] A. Banarjee. ECE464. Class Lecture, Topic: “DC-DC conversion.” College of Electrical and Computer Engineering, University of Illinois Urbana-Champaign, Urbana, IL.

[8] B. Isidor, “BU-403: Charging Lead Acid,” Available: http://batteryuniversity.com/learn/article/charging_the_lead_acid_battery. [Accessed Feb, 2018].

[9] G. M. Masters, “Chapter 4: The Solar Resource,” in *Renewable and Efficient Electric Power Systems,* Ed. 2, Wiley-IEEE Press, May 2013.

[10] R. W. Erickson et al., “Chapter 14: Inductor Design,” in *Fundamentalsof Power Electronics,* New York: Springer Science + Business Media , 2001.

[11] A. Banarjee. ECE464. Class Lecture, Topic: “Real Components.” College of Electrical and Computer Engineering, University of Illinois Urbana-Champaign, Urbana, IL.

[12] “Aluminum Electrolytic Capacitors,” EPCOS AG, pp. 12, Dec. 2016. Accesed on: Apr. 10, 2018 [Online].Available:

https://en.tdk.eu/download/530704/5f33d2619fa73419e2a4af562122e90c/pdf-generaltechnicalinformation.pdf.

[13] G. Lakkast, *MOSFET power losses and how they affect power-supply efficiency. Analog Applications Journal Texas instrument*, 2016, [online] Available: http://www.ti.com/lit/an/slyt664/slyt664.pdf.

[14] Vishay, "Power MOSFET," IRF510 datasheet, Nov. 2015 [Revised Feb. 2017].

[15] IXYS, "Power Schottky Rectifier," DSSK 80-0025 datasheet, 2002.

[16] TDK, "Power Line Chokes," B82726S2163N0 datasheet, Jul. 2012.

[17] C. Rodrígrez-Morcillo, Garcia, Electrónica. Class Lecture, Topic: "Amplificación, Amplificadores, y Amplificadores Operacionales." School of Engineering, Universidad Pontificia de Comillas ICAI, Madrid, España.

[18] A. Banarjee, ECE469, Power Electronics Laboratory, College of Electrical and Computer Engineering, University of Illinois Urbana-Champaign, Urbana, IL.

[19] L. M. Mochón, Castro, Mecánica de Fluidos, Class Lecture. School of Engineering, Universidad Pontificia de Comillas ICAI, Madrid, España.

APPENDIX A: MICROCONTROLLER CODE

```
#include <Wire.h>
#include <SparkFun_MS5803_I2C.h>
#include <LCD.h>
#include <LiquidCrystal.h>
#include <LiquidCrystal_I2C.h>

MS5803 sensor(ADDRESS_HIGH);
#define I2C_ADDR    0x3F
#define BACKLIGHT_PIN     3
#define En_pin  2
#define Rw_pin  1
#define Rs_pin  0
#define D4_pin  4
#define D5_pin  5
#define D6_pin  6
#define D7_pin  7

LiquidCrystal_I2C
lcd(I2C_ADDR,En_pin,Rw_pin,Rs_pin,D4_pin,D5_pin,D6_pin,D7_pin);

//Create variables to store results
float temperature_c, temperature_f;
double pressure_abs, pressure_relative, altitude_delta, pressure_baseline, height,
volume;
const float radius = (11.8/2) ;
const float Pi = 3.14159;
int solenoidPin = 8;
int solenoidPin2 = 9;
const int buttonPin1 = 14;     // the number of the pushbutton pin
const int buttonPin2 = 15;
const int buttonPin3 = 16;
const int buttonPinyes = 17;
int buttonState1 = 0;
int buttonState2 = 0;
int buttonState3 = 0;
int buttonStateyes = 0;
int starttime = 0;
int endtime = 0;
int count = 0 ;

void setup() {
  Serial.begin(9600);
  //Retrieve calibration constants for conversion math.
  sensor.reset();
  sensor.begin();
  pressure_baseline = sensor.getPressure(ADC_4096);
  pinMode(solenoidPin, OUTPUT);
  pinMode(solenoidPin2, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin1, INPUT);
  pinMode(buttonPin2, INPUT);
  pinMode(buttonPin3, INPUT);
  pinMode(buttonPinyes, INPUT);
  digitalWrite(solenoidPin, HIGH);
  lcd.begin (16,2); //  <<----- My LCD was 16x2
  lcd.setBacklightPin(BACKLIGHT_PIN,POSITIVE);
  lcd.setBacklight(HIGH);
  lcd.clear();
  lcd.home (); // go home
}
```

```
void loop() {
  // To measure to higher degrees of precision use the following sensor settings:
  // ADC_256
  // ADC_512
  // ADC_1024
  // ADC_2048
  // ADC_4096
  // read the state of the pushbutton value:
  lcd.setCursor(0,0);
  lcd.print("Temp:" );
  lcd.print(temperature_f);
  lcd.setCursor(0,1);
  lcd.print("height:" );
  lcd.print(height);
  digitalWrite(solenoidPin2, HIGH);
  buttonState1 = digitalRead(buttonPin1);
  buttonState2 = digitalRead(buttonPin2);
  buttonState3 = digitalRead(buttonPin3);
  buttonStateyes = digitalRead(buttonPinyes);  digitalWrite(solenoidPin, HIGH);
  // check the boundary
  if( height < 11 ){
    lcd.clear();
    lcd.print("Please wait");
    delay(5000);
    lcd.clear();
  }
  if( height > 33 ){
    digitalWrite(solenoidPin, LOW);
  }

   //check if the pushbutton is pressed. If it is, the buttonState is HIGH:
  if (buttonState1 == HIGH) {
```

```
  button1() ;
}
else if (buttonState2 == HIGH) {
  button2() ;
}

else if (buttonState3 == HIGH) {
  button3();
}

// Read temperature from the sensor in deg C. This operation takes about
temperature_c = sensor.getTemperature(CELSIUS, ADC_512);

// Read temperature from the sensor in deg F. Converting
// to Fahrenheit is not internal to the sensor.
// Additional math is done to convert a Celsius reading.
temperature_f = sensor.getTemperature(FAHRENHEIT, ADC_512);

// Read pressure from the sensor in pa.
//radius = 1.2789916 dm
pressure_abs = (sensor.getPressure(ADC_4096)* 100) - 98460;
height = (pressure_abs) / 9.81 / 997 * 100   ;
volume = Pi * radius * radius * height ;
// Report values via UART
Serial.print("Temperature F = ");
Serial.println(temperature_f);
Serial.print("Pressure abs (atm)= ");
Serial.println(pressure_abs);
Serial.print("height = ");
Serial.println(height);
Serial.print("volume = ");
Serial.println(volume);
```

```
  delay(200);
}
//
void button1(){
  starttime = millis();
  endtime = starttime;
  lcd.clear();
  while ((endtime - starttime) <=10000){ // do this loop for up to 10000mS
    lcd.setCursor(0,0);
    lcd.print("you've pressed 1 ");
    lcd.setCursor(0,1);
    lcd.print("TRY NEXT YEAR");
    int buttonState1 = LOW;
    endtime = millis();
    buttonState2 = digitalRead(buttonPin2);
    buttonState3 = digitalRead(buttonPin3);
    buttonStateyes = digitalRead(buttonPinyes);
    Serial.print(buttonState2);
    if (buttonState2 == HIGH) {
      button2();
      break ;
    }
    if (buttonState3 == HIGH) {
      button3();
      break ;
    }
    else if (buttonStateyes == HIGH) {
      lcd.clear();
      lcd.print("dispense 1 L");
      digitalWrite(solenoidPin2, HIGH);
      digitalWrite(solenoidPin, LOW);
      count = count + 1;
      float time1 = 2*pow((13/0.37),2)*((sqrt(height*pow(10,-2))-sqrt(height*pow(10,-2)
- (1/(3.14159*pow(1.3,2))*pow(10,-1))))/sqrt(2*9.81));
      Serial.print(time1);
      delay(time1 * 1000);
      digitalWrite(solenoidPin2, LOW);//Switch Solenoid OFF
      digitalWrite(solenoidPin, HIGH);
      break ;
    }
    delay(200);
  }
  lcd.clear();
}

void button2(){
  starttime = millis();
  endtime = starttime;
  lcd.clear();
  while ((endtime - starttime) <=10000){ // do this loop for up to 10000mS
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("you've pressed 2");
    lcd.setCursor(0,1);
    lcd.print("NEW SHOES");
    int buttonState1 = LOW;
    endtime = millis();
    buttonState1 = digitalRead(buttonPin1);
    buttonState3 = digitalRead(buttonPin3);
    buttonStateyes = digitalRead(buttonPinyes);
    if (buttonState1 == HIGH) {
      button1();
      break;
    }
    else if (buttonState3 == HIGH) {
      button3();
      break ;
    }
    else if (buttonStateyes == HIGH) {
      lcd.clear();
      lcd.print("dispense 2 L");
      digitalWrite(solenoidPin2, HIGH);
      digitalWrite(solenoidPin, LOW);
      count = count + 2;
      float time2 = 2*pow((13/0.37),2)*((sqrt(height*pow(10,-2))-sqrt(height*pow(10,-2) - (2/(3.14159*pow(1.3,2))*pow(10,-1))))/sqrt(2*9.81));
      Serial.print(time2);
      delay(time2 * 1000);                        //Dispense 2 L
      digitalWrite(solenoidPin2, LOW);     //Switch Solenoid OFF
      digitalWrite(solenoidPin, HIGH);
      break;
    }
    delay(200);
  }
  lcd.clear();
}

void button3(){
  starttime = millis();
  endtime = starttime;
  lcd.clear();
  while ((endtime - starttime) <=10000){// do this loop for up to 10000mS

    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("you've pressed 3 ");
```

```
   lcd.setCursor(0,1);
   lcd.print("A SURPRISE VISIT");
   int buttonState3 = LOW;
   endtime = millis();
   buttonState1 = digitalRead(buttonPin1);
   buttonState2 = digitalRead(buttonPin2);
   buttonStateyes = digitalRead(buttonPinyes);
   if (buttonState1 == HIGH) {
     button1();
     break;
   }
   if (buttonState2 == HIGH) {
     button2();
     break;
   }
   if (buttonStateyes == HIGH) {
     lcd.clear();
     lcd.print("dispense 3 L");
     digitalWrite(solenoidPin2, HIGH);
     digitalWrite(solenoidPin, LOW);
     count = count + 3;
     float time3 = 2*pow((13/0.37),2)*((sqrt(height*pow(10,-2))-sqrt(height*pow(10,-2)
- (3/(3.14159*pow(1.3,2))*pow(10,-1))))/sqrt(2*9.81));
     Serial.print(time3);
     delay(time3 * 1000);                    //Dispense 2 L
     digitalWrite(solenoidPin2, LOW);     //Switch Solenoid OFF
     digitalWrite(solenoidPin, HIGH);
     break ;
   }
   delay(200);
  }
  lcd.clear();
```

```
}

/**********************************************************************
********
MS5803_I2C.h
Library for MS5803 pressure sensors.
Casey Kuhns @ SparkFun Electronics
6/26/2014
https://github.com/sparkfun/MS5803-14BA_Breakout
The MS58XX MS57XX and MS56XX by Measurement Specialties is a low cost I2C
pressure sensor.  This sensor can be used in weather stations and for altitude
estimations. It can also be used underwater for water depth measurements.
In this file are the function prototypes in the MS5803 class
Resources:
This library uses the Arduino Wire.h to complete I2C transactions.
Development environment specifics:
        IDE: Arduino 1.0.5
        Hardware Platform: Arduino Pro 3.3V/8MHz
        MS5803 Breakout Version: 1.0
**Updated for Arduino 1.6.4 5/2015**
This code is beerware. If you see me (or any other SparkFun employee) at the
local pub, and you've found our code helpful, please buy us a round!
Distributed as-is; no warranty is given.
**********************************************************************
********/
#ifndef SparkFun_MS5803_I2C_h
#define SparkFun_MS5803_I2C_h


#include <Arduino.h>
// Define units for conversions.
enum temperature_units
{
        CELSIUS,
        FAHRENHEIT,
};
// Define measurement type.
enum measurement
{
        PRESSURE = 0x00,
        TEMPERATURE = 0x10
};
// Define constants for Conversion precision
enum precision
{
        ADC_256  = 0x00,
        ADC_512  = 0x02,
        ADC_1024 = 0x04,
        ADC_2048 = 0x06,
        ADC_4096 = 0x08
};
// Define address choices for the device (I2C mode)
enum ms5803_addr
{
        ADDRESS_HIGH = 0x76,
        ADDRESS_LOW  = 0x77
};
//Commands
#define CMD_RESET 0x1E // reset command
#define CMD_ADC_READ 0x00 // ADC read command
#define CMD_ADC_CONV 0x40 // ADC conversion command
#define CMD_PROM 0xA0 // Coefficient location
class MS5803
{
        public:
```

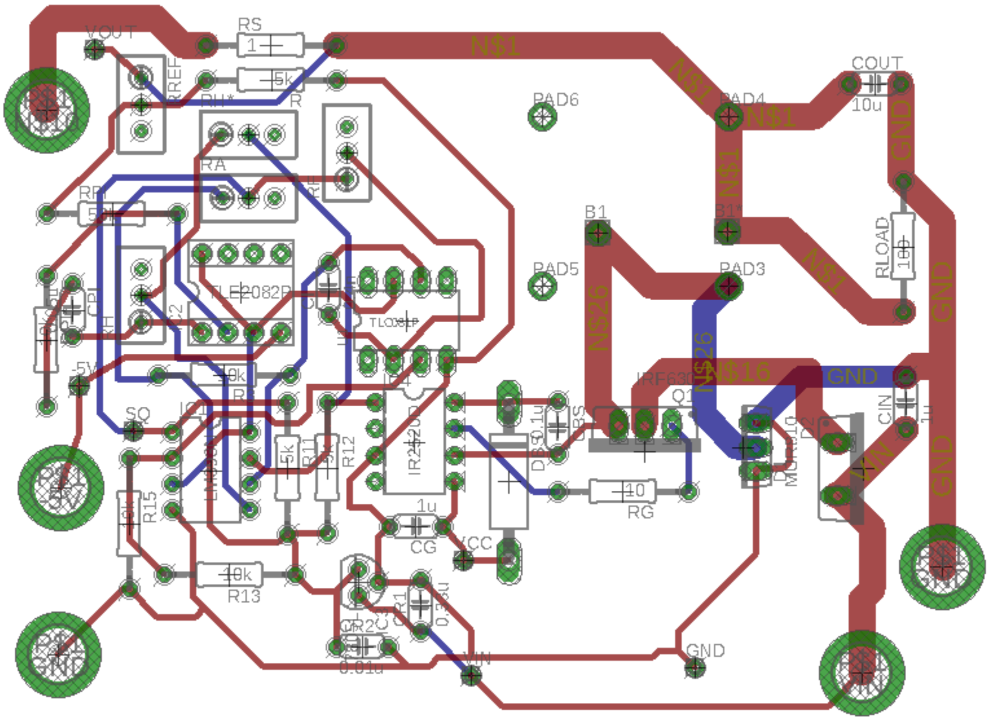
```
        MS5803(ms5803_addr address);

        void reset(void);       //Reset device

        uint8_t begin(void); // Collect coefficients from sensor

        // Return calculated temperature from sensor

        float getTemperature(temperature_units units, precision _precision);

        // Return calculated pressure from sensor

        float getPressure(precision _precision);

    private:

        int32_t _temperature_actual;

        int32_t _pressure_actual;

        ms5803_addr _address;               // Variable used to store I2C device
address.

        uint16_t coefficient[8];// Coefficients;

        void getMeasurements(precision _precision);

        void sendCommand(uint8_t command);    // General I2C send command
function

        uint32_t getADCconversion(measurement _measurement, precision
_precision);    // Retrieve ADC result

        void sensorWait(uint8_t time); // General delay function see: delay()

};

#endif

/******************************************************************
********

MS5803_I2C.cpp

Library for MS5803 pressure sensor.

Casey Kuhns @ SparkFun Electronics

6/26/2014

https://github.com/sparkfun/MS5803-14BA_Breakout

The MS58XX MS57XX and MS56XX by Measurement Specialties is a low cost I2C
pressur

sensor.  This sensor can be used in weather stations and for altitude

estimations. It can also be used underwater for water depth measurements.

In this file are the functions in the MS5803 class

Resources:
```

```
This library uses the Arduino Wire.h to complete I2C transactions.
Development environment specifics:
        IDE: Arduino 1.0.5
        Hardware Platform: Arduino Pro 3.3V/8MHz
        MS5803 Breakout Version: 1.0
**Updated for Arduino 1.6.4 5/2015**
This code is beerware. If you see me (or any other SparkFun employee) at the
local pub, and you've found our code helpful, please buy us a round!
Distributed as-is; no warranty is given.
********************************************************************************
*********/
#include <Wire.h> // Wire library is used for I2C
#include "SparkFun_MS5803_I2C.h"
MS5803::MS5803(ms5803_addr address
// Base library type I2C
{
        Wire.begin(); // Arduino Wire library initializer
        _address = address; //set interface used for communication
}
void MS5803::reset(void)
// Reset device I2C
{
  sendCommand(CMD_RESET);
  sensorWait(3);
}
uint8_t MS5803::begin(void)
// Initialize library for subsequent pressure measurements
{
        uint8_t i;
        for(i = 0; i <= 7; i++)
  {
                sendCommand(CMD_PROM + (i * 2));
		Wire.requestFrom( _address, 2);
		uint8_t highByte = Wire.read();
		uint8_t lowByte = Wire.read();
		coefficient[i] = (highByte << 8)|lowByte;
	// Uncomment below for debugging output
	//	Serial.print("C");
	//	Serial.print(i);
	//	Serial.print("= ");
	//	Serial.println(coefficient[i]);
	}
	return 0;
}
float MS5803::getTemperature(temperature_units units, precision _precision)
// Return a temperature reading in either F or C
{
	getMeasurements(_precision);
	float temperature_reported;
	// If Fahrenheit is selected return the temperature converted to F
	if(units == FAHRENHEIT)
  {
		temperature_reported = _temperature_actual / 100.0f;
		temperature_reported = (((temperature_reported) * 9) / 5) + 32;
		return temperature_reported;
	}
	// If Celsius is selected return the temperature converted to C
	else
  {
		temperature_reported = _temperature_actual / 100.0f;
		return temperature_reported;
	}
}
float MS5803::getPressure(precision _precision)
```

```
// Return a pressure reading units Pa.
{
        getMeasurements(_precision);
        float pressure_reported;
        pressure_reported = _pressure_actual;
        pressure_reported = pressure_reported / 10.0f;
        return pressure_reported;
}
void MS5803::getMeasurements(precision _precision)
{
        //Retrieve ADC result
        int32_t temperature_raw = getADCconversion(TEMPERATURE, _precision);
        int32_t pressure_raw = getADCconversion(PRESSURE, _precision);
        //Create Variables for calculations
        int32_t temp_calc;
        int32_t pressure_calc;
        int32_t dT;
        //Now that we have a raw temperature, let's compute our actual.
        dT = temperature_raw - ((int32_t)coefficient[5] << 8);
        temp_calc = (((int64_t)dT * coefficient[7]) >> 23) + 2000;
        // TODO TESTING  _temperature_actual = temp_calc;
        //Now we have our first order Temperature, let's calculate the second order
        int64_t T2, OFF2, SENS2, OFF, SENS; //working variables
        if (temp_calc < 2000)
        // If temp_calc is below 20.0C
        {
                T2 = 3 * (((int64_t)dT * dT) >> 33);
                OFF2 = 3 * ((temp_calc - 2000) * (temp_calc - 2000)) / 2;
                SENS2 = 5 * ((temp_calc - 2000) * (temp_calc - 2000)) / 8;
                if(temp_calc < -1500)
                // If temp_calc is below -15.0C
                {
```

```
            OFF2 = OFF2 + 7 * ((temp_calc + 1500) * (temp_calc + 1500));
            SENS2 = SENS2 + 4 * ((temp_calc + 1500) * (temp_calc +
1500));
        }
 }
    else
    // If temp_calc is above 20.0C
    {
        T2 = 7 * ((uint64_t)dT * dT)/pow(2,37);
        OFF2 = ((temp_calc - 2000) * (temp_calc - 2000)) / 16;
        SENS2 = 0;
    }
    // Now bring it all together to apply offsets
    OFF = ((int64_t)coefficient[2] << 16) + (((coefficient[4] * (int64_t)dT)) >> 7);
    SENS = ((int64_t)coefficient[1] << 15) + (((coefficient[3] * (int64_t)dT)) >> 8);
    temp_calc = temp_calc - T2;
    OFF = OFF - OFF2;
    SENS = SENS - SENS2
    // Now lets calculate the pressur
    pressure_calc = (((SENS * pressure_raw) / 2097152 ) - OFF) / 32768;
    _temperature_actual = temp_calc ;
    _pressure_actual = pressure_calc ; // 10;// pressure_calc;
}
uint32_t MS5803::getADCconversion(measurement _measurement, precision
_precision)
// Retrieve ADC measurement from the device.
// Select measurement type and precision
{
    uint32_t result;
    uint8_t highByte = 0, midByte = 0, lowByte = 0;
    sendCommand(CMD_ADC_CONV + _measurement + _precision);
    // Wait for conversion to complete
    sensorWait(1); //general delay
	switch( _precision )
	{
		case ADC_256 : sensorWait(1); break;
		case ADC_512 : sensorWait(3); break;
		case ADC_1024: sensorWait(4); break;
		case ADC_2048: sensorWait(6); break;
		case ADC_4096: sensorWait(10); break;
	}
	sendCommand(CMD_ADC_READ);
	Wire.requestFrom(_address, 3);
	while(Wire.available())
	{
		highByte = Wire.read();
		midByte = Wire.read();
		lowByte = Wire.read();
	}
	result = ((uint32_t)highByte << 16) + ((uint32_t)midByte << 8) + lowByte;
	return result;
}
void MS5803::sendCommand(uint8_t command)
{
	Wire.beginTransmission( _address);
	Wire.write(command);
	Wire.endTransmission();
}
void MS5803::sensorWait(uint8_t time)
// Delay function.  This can be modified to work outside of Arduino based MCU's
{
	delay(time);
}
```

APPENDIX B: CIRCUIT LAYOUT



Buck Converter PCB Layout

APPENDIX C: PART COSTS

	Part	Quantity	Unit Cost
Power Module	100 W Solar Panel	1	$109.95
	LM7805 Voltage Regulator	1	$0.70
	DSSK 48-0025B Schottky Diode	1	$2.56
	B82726S2163N0 1.5mH Inductor	1	$9.80
	Various Capacitors	x	x
	Various Resistors	x	x
	12V 12AH Lead-Acid Battery Power Sonic	1	$28.99
	IRF2125 Gate driver	1	$6.24
	TLC081 Op-Amp	3	$1.60
	TLE2082 Op-Amp	2	$3.35
	IRF630 Mosfet	1	$0.94
Control Module	ATMega328p Microcontroller	1	$2.01
	Plastic Water Solenoid Valve	1	$6.95
	Brass Water Solenoid Valve	1	$29.99
	SparkFun Pressure Sensor Breakout - MS5803-14BA	1	$59.95

Filtration Unit	iSpring RCC7 High Capacity Under Sink 5-Stage Reverse Osmosis with booster pump	1	$249.99
I/O	16x2 LCD display	1	$6.48
	Input Buttons	4	$2.00