CONSTRUCTION OF MAXIMUM POWER POINT TRACKING WIND-SOLAR HYBRID CHARGE CONTROLLER

Win Zaw Myo Htet[1], Lwan Htet Nay Aung[2]

Abstract

Both wind and solar circuits are constructed by using an Arduino Uno that uses ATmega328P. ACS712 Hall Current Sensor and the voltage divider circuit are used to measure the input and output current and voltage of wind, solar and battery. PWM signal is provided by Arduino Uno to drive the MOSFETs via IR2110 MOSFET driver IC. High ampere dc relay is used to switch between dump load and wind circuit. The 4-lines LCD is used to display the voltage and the current of solar, wind and battery. The A/D module of an Arduino Uno is used to convert the output voltage and current from voltage divider and ACS712 into digital forms which are used as the reference values in the program. When battery is at maximum capacity, the controller will activate a relay that will send the excess power to a dump load resistor.

Keywords: hall current sensor, A/D module

Introduction

Research on MPPT controllers is becoming more popular as the interest in renewable energy technologies rises. In a PV system, when compared to the battery the energy produce by the panel(s) may be over or under. Not to waste energy, a buck or boost converter in a MPPT controller is used to gain higher power production efficiency. With a turbine, it is uncommon to have a steady input source because wind stays never steady. When the wind is strong and steady, it would be ideal to maximize the opportunity to harvest as much energy as possible through a buck converter. DC-DC converters with MPPT that provided the maximum power transfer are implemented. It can change the current and voltage levels so that the maximum power can be extracted from the PV array. Converting a given fixed load to a variable load is changing the voltage and current level. The whole circuit operation is shown as block diagram in Figure 1.

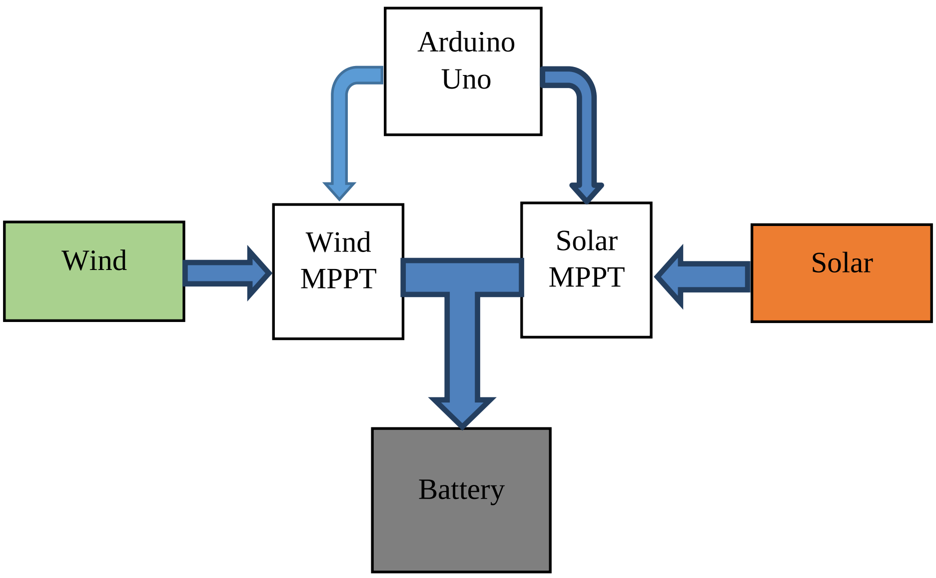

Figure 1 block diagram of MPPT hybrid charge controller

[1] Assistant Lecturer, Department of Physics, Bago University
[2] Lecturer, Department of Physics, Bago University

Controller Design

The diode and the capacitor on the input side filter noise that may be present. The input is read by the voltage divider and then passes through the ACS712 current sensor on both solar and wind. For this controller, the input is fed through a MOSFET if the voltage is high enough to meet the enable voltage, then it flows through the switching node. Pin 7 on the wind and pin 8 on the solar control the enabling and the disabling of the MOSFET. The Arduino begins setting up the hybrid MPPT controller by reading the defined variables and pins. It sets the digital pins 0, 6, 7, 8, and 11 to outputs. PWM signal or High/Low signal is given out by digital pins (0, 6, 7, 8 and 11). Since pins 0, 7, 8 are used as High/Low, they are initially given the Low (off) state not to allow flow from the output. The duty cycle has set limits not to allow either energy flow from the output and keep the bootstrap gate capacitor charged. The Arduino UNO sends a 62.5 kHz PWM signal from pin 6 and a 31.25 kHz PWM signal from pin 11 to the IR2110 high and low side drivers. The driver sends the same signal to the MOSFETs but amplifies the driving voltage to the MOSFET gate. The output voltage and current are measured and compared to the set values assigned within the program.

The duty cycle within the PWM equates to on/off time for the MOSFET. If the voltage from the output of DC-DC converter is above the maximum, the duty cycle will decrease until the output voltage matches the set limit. Conversely, if the DC-DC converter output voltage is less than the minimum limit set, the duty cycle will increase. If the input voltage from the solar panel and/or the wind turbine is less than the disable voltage or the current is less than the minimum, then the DC-DC converter will be disabled. If the system is disabled, it will be constantly checked by the Arduino till the system enabled. If the input voltages are greater than the enable voltage, it turns on the DC-DC converters. Figure 2 shows the circuit schematic for the hybrid controller and Figure 3 shows the photo of the hybrid controller circuit.

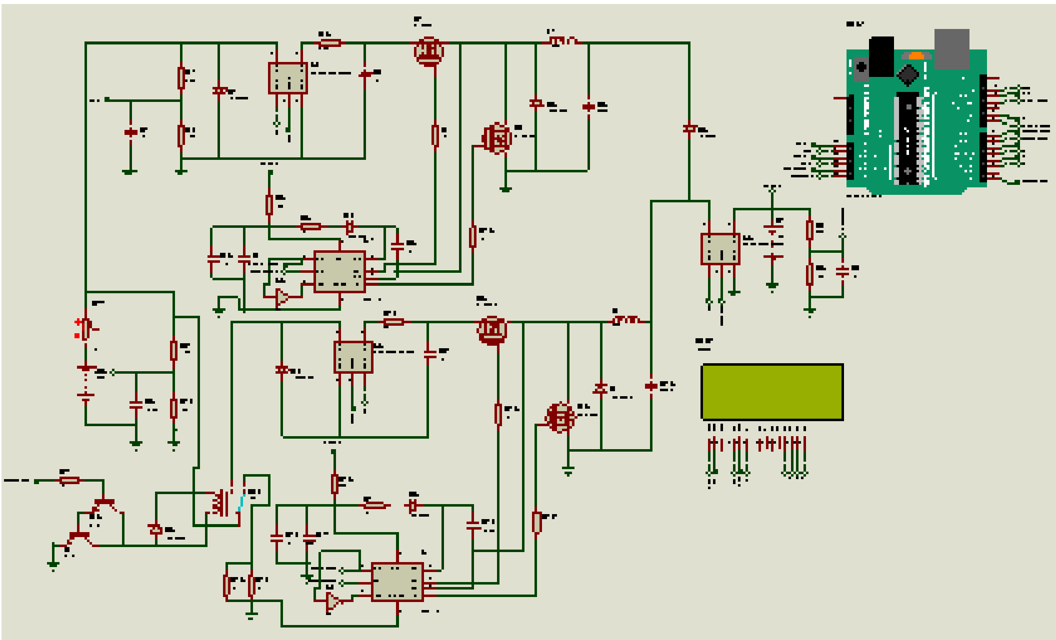

Figure 2 Hybrid MPPT charge controller circuit diagram

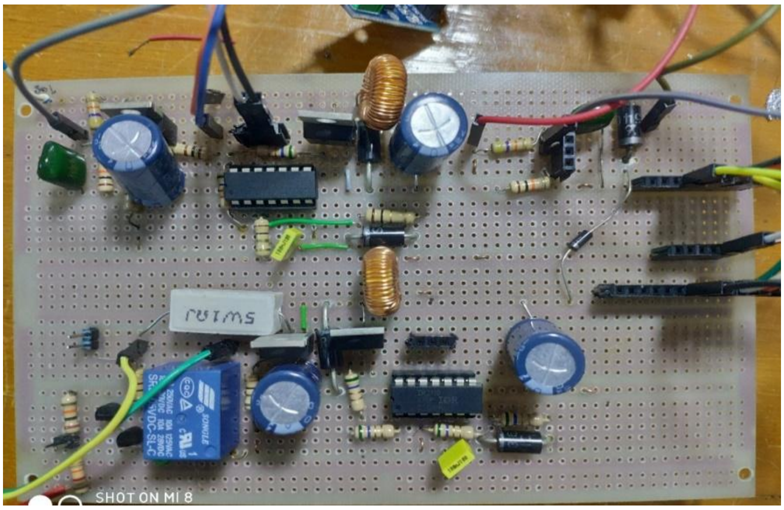

Figure 3 Photo of hybrid MPPT charge controller circuit

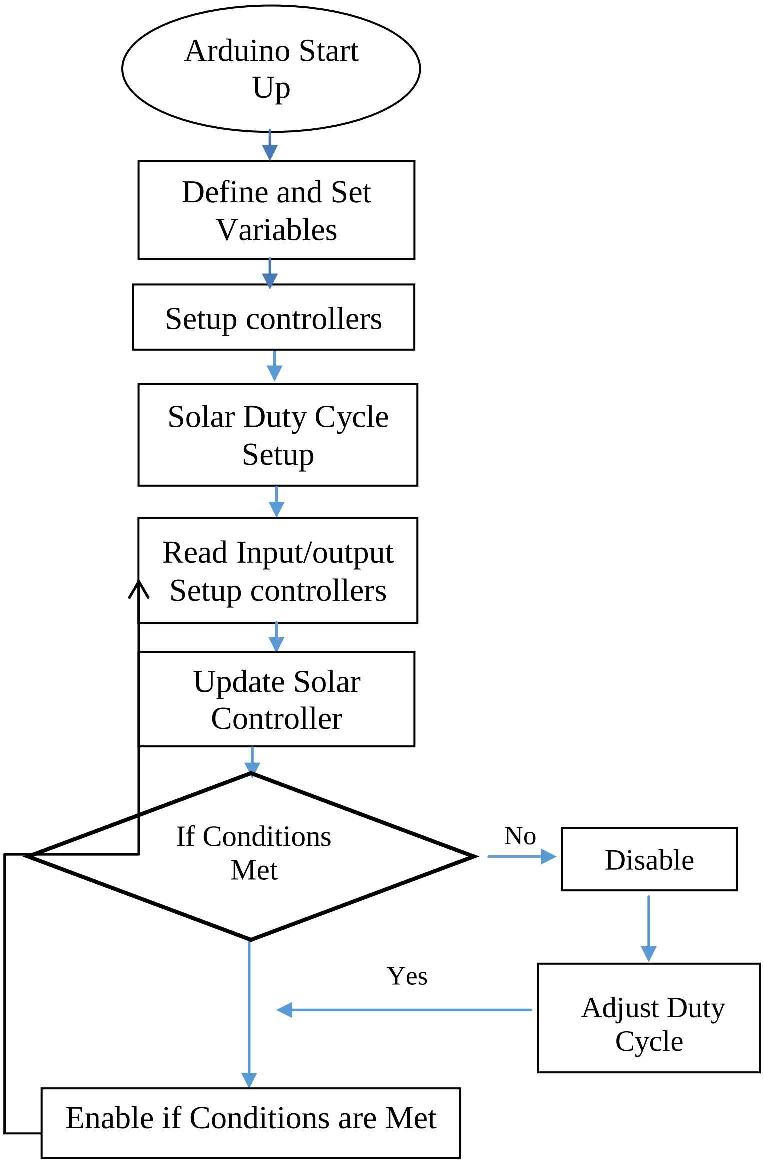

Figure 4 Flow diagram of control sequence

Operating Procedure

Once there is understanding of how the controller is designed to work, the next step is to have the circuit communicate with the controller. When the controller receives power from the UNO, it reads the defined conditions of the input and output sources. It begins to set up the controller by identifying which pins will provide an output and setting the PWM modes. It displays a welcome screen and then begins the infinite loop after setting up the controller. The loop contains three different parts, the first being to read the values at the input and output. Then, it begins the solar and wind updates; the Arduino controls the amount of duty being supplied through each PWM pin to correct the output voltage and input current.

It is necessary to set up a time delay because when operating on Fast PWM, the clock runs 64 times faster than normal. The delay will allow the Arduino to read at a normal pace. The duty cycle for the controllers needs to have a maximum and minimum level to protect the circuit. A voltage divider is used to produce a measurable voltage for the Arduino. The voltage multipliers are implemented to reflect the voltages before being reduced. Using an LCD display to show the values from the input and the output, the library Liquid Crystal is used to work with LCD screen.

The initial setup processes the hybrid MPPT controller goes through upon cycling on. The function pinMode sets the Arduino's pins to produce output. The "enable" pins, such Solar_Enable and Wind_Enable, are not defined to a PWM frequency but to a LOW or HIGH (on or off) through the function digital Write. The PWM pins need to be called upon to produce the fast PWM for pin 6 and the phase shift PWM for pin 11. This is done by setting the TCCR0A and TCCR2A registers then using the Output Compare (OCR0A and OCR2A) pin for the duty cycle, which is initially set to 0.

The enabling and the disabling parameters are set in phase, to ensure that the power is being delivered when it is needed and diverted away when it is not needed. Both buck converters need off time to keep the bootstrap capacitor for the upper MOSFET charged, so the duty cycle is set below 100% to ensure the capacitor is charged. If the bootstrap capacitor is not charged, the gate will turn off, creating a partially conducting state. A synchronous buck can eliminate the possibility of the voltage from the output to flow to the input by enforcing the duty cycle above 50%.

Measuring and Displaying Input and Output Values

The Arduino's microcontroller uses an analog to digital converter (ADC) that reads the voltage and converts it to a number between 0 and 1023. The circuit has a limit of 5V or it will overload. The voltage divider is used to measure anything above 5V, and the ACS712 converts the current to a voltage that is read the same way as the voltage divider. Each value is measured 16 times and then averaged. Once the values are read and averaged, it is then sent to the LCD screen.

Updating Solar and Wind DC-DC Converter

The Arduino begins the loop by reading the input and output values and then updating the controller for the wind and solar DC-DC converters based on the values. The Arduino decides to enable, disable, or update the duty cycle based on the state that the controller is in and the input/output values read. Wind and solar are nearly identical in the program design for disabling, enabling, and updating the duty cycle. The wind DC-DC converter has an additional condition if the turbine is producing more than 50V. The controller will disable the circuit and send it to the dump load resistors; this is to protect the components that are sensitive to voltages above 50V.

The Arduino begins by determining that the DC-DC converters are operating. If they are operating, then it will disable the DC-DC converters if the solar and wind voltages are less than their respective disable voltages, or the current is below the output's minimum. If the operating conditions are met, then the controller checks:

1. If the input voltages are less than the target voltages
2. If the output is greater than the output limit
3. If the input current is greater than the input maximum or
4. If the output current is greater than the output maximum

If one or more of these conditions are met, it will reduce the duty cycle by one until the desired output is met. If the voltages are greater than the target voltages and the output voltage is less than the minimum limit, then it will increase the duty cycle by one until the desired output is met. The Arduino will enable the DC-DC converters if previously disabled and the input is greater than the enable voltage. When in the enable state, it will measure and update the duty cycle by multiplying 255 by the output voltages over the input voltages.

Conclusion

The wind and solar hybrid charge controller circuit acts as a control circuit to regulate the process of wind turbine and solar panel battery charging process. These circuits can be constructed from discrete electronic components. The circuit operation is based on matching the wind and solar cell terminal load voltage to the appropriate number of battery cell units to be charged depending on the total power output of hybrid power source.

The advantage of using hybrid controller is controlling two systems of solar and wind energy simultaneously. The ability to control two systems with one controller is better for an overall production of energy, cost, and manageability at a slight expense to the efficiency. Developing a buck-boost converter for the wind MPPT controller would take advantage of both the lower and higher wind speed. Though the overall efficiency is less than that of a concentrated buck or boost controller, it would have more potential to produce power. To produce higher power from generator inductor size in controller must be increase due to frequency provided by Arduino. A larger inductor means less allowable current to flow before the inductor becomes over saturated, reducing the efficiency of the controller.

Acknowledgements

I would like to thank Professor Dr Khin Khin Win, Head of Department of Physics, University of Yangon and Professor Dr Khin Mar Ohn, Head of Department of Physics, Bago University, for her kind permission to carry out this work. And I also thank Professor Dr Khin Htoo, Department of Physics, Bago University for her encouragement. I am thankful for the valuable guidance and support provided by my co-supervisor Professor Dr Myo Lwin, Department of Physics, University of Yangon. I would like to express my sincere gratitude to my supervisor Dr Zaw Zaw Aung, Lecturer, Department of Physics, University of Yangon for his valuable advice and help. And my thanks also goes to U Thant Shwe, Lecturer, Department of Physics, Bago University, Dr Mie Mie Aye, Lecturer, Department of Physics, Dagon University for their help in programming and suggestion for this work. Without their guidance and feedback, the thesis work could not have been a success.

References

Allegro Micro Systems (2015, Nov) *Fully Integrated, Hall Effect-Based Linear Current Sensor IC with 2.1 kVRMS Isolation and a Low-Resistance Current Conductor*, ACS712 datasheet, [Revised June 2017].

Azevedo, J., Mendoca. F., (July, 2015) "Small scale wind energy harvesting with maximum power tracking," AIMS Energy, Vol.3 no.3 pp.297-315. DOI:10.3934/ energy.2015.3.297.

Choudhary, D., Sexana, (2014) *A.R.DC-DC Buck-Converter for MPPT of PV System, International Journal of Emerging Technology and Advanced Engineering.*

ON Semiconductor (2013, Apr) LC selection Guide for the DC-DC synchronous Buck Converter.

Unknown Authors (2017, July) How to Design Battery Charger Applications that Require External Microcontrollers and Related System-Level Issues, Maxim Integrated https://www.maximintegrated.com/en/app-notes/index.mvp/id/680