

### Show All 12 Items

Welcome to my solar charge controller tutorials series.I have posted two version of my PWM charge controller.If you are new to this please refer my earlier tutorial for understanding the basics of charge controller.

1.

[Version-1](#)

2. [Version-2](#)

This instructable will cover a project build for a Arduino based Solar MPPT charge controller.It has features like: LCD display,Led Indication,Wi Fi data logging and provision for charging different USB devices.It is equipped with various protections to protect the

circuitry from abnormal condition.

The microcontroller used in this controller is Arduino Nano. This design is suitable for a 50W solar panel to charge a commonly used 12V lead acid battery. You can also use other Arduino board like Pro Mini,Micro and UNO.

Now a days the most advance solar charge controller available in the market is Maximum Power Point Tracking (MPPT).The MPPT controller is more sophisticated and more expensive.It has several advantages over the earlier charge controller.It is 30 to 40 % more efficient at low temperature.But making a MPPT charge controller is little bit complex in compare to PWM charge controller.It require some basic knowledge of power electronics.

I put a lot of effort to make it simple, so that any one can understand it easily.If you are aware about the basics of MPPT charge controller then skip the first few steps.

The Maximum Power Point Tracker (MPPT) circuit is based around a synchronous buck converter circuit..It steps the higher solar panel voltage down to the charging voltage of the battery. The Arduino tries to maximize the watts input from the solar panel by controlling the duty cycle to keep the solar panel operating at its Maximum Power Point.

### **Specification of version-3 charge controller :**

- 1.Based on MPPT algorithm
2. LED indication for the state of charge
3. 20x4 character LCD display for displaying voltages,current,power etc
4. Overvoltage / Lightning protection
5. Reverse power flow protection
6. Short Circuit and Over load protection
7. Wi Fi data logging
- 8.USB port for Charging Smart Phone /Gadgets

### **Electrical specifications :**

- 1.Rated Voltage= 12V
- 2.Maximum current = 5A
- 3.Maximum load current =10A
4. In put Voltage = Solar panel with Open circuit voltage from 12 to 25V

This project is consists of 40 steps.So for simplicity I divided the entire project in to small sections.Click on the link which you want to see.

1. Basics on MPPT charge controller
2. Buck circuit working and design calculation
3. Testing the Buck Circuit
4. Voltage and Current Measurements
- 5.LCD display and LED indication

6. Making the Charging Board
7. Making the Enclosure
8. Making the USB Charging Circuit
9. Wi Fi Data Logging
10. MPPT algorithm and flow chart

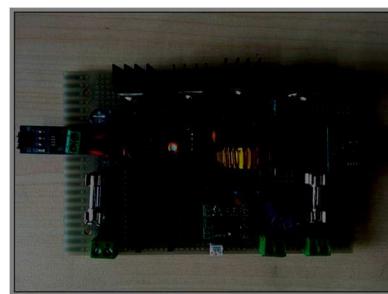
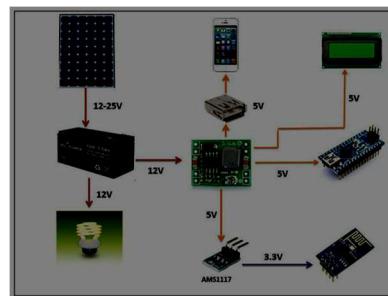
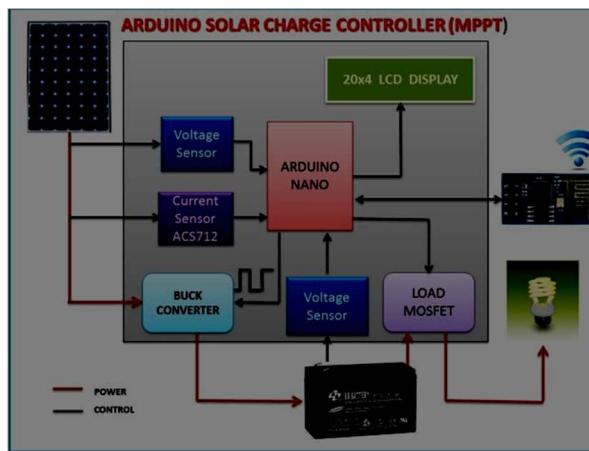
*I have spent a lot of time to make this charge controller project.If you like it ,please vote for me in all the competitions.It will be very helpful for me.Thank you in advance.*



Remove these ads by [Signing Up](#)

## Step 1: PARTS AND TOOLS REQUIRED:







### Show All 12 Items

Welcome to my solar charge controller tutorials series.I have posted two version of my PWM charge controller.If you are new to this please refer my earlier tutorial for understanding the basics of charge controller.

1.

[Version-1](#)

2. [Version-2](#)

This instructable will cover a project build for a Arduino based Solar MPPT charge controller.It has features like: LCD display,Led Indication,Wi Fi data logging and provision for charging different USB devices.It is equipped with various protections to protect the circuitry from abnormal condition.

The microcontroller used is in this controller is Arduino Nano. This design is suitable for a 50W solar panel to charge a commonly used 12V lead acid battery. You can also use other Arduino board like Pro Mini,Micro and UNO.

Now a days the most advance solar charge controller available in the market is Maximum Power Point Tracking (MPPT).The MPPT controller is more sophisticated and more expensive.It has several advantages over the earlier charge controller.It is 30 to 40 % more efficient at low temperature.But making a MPPT charge controller is little bit complex in compare to PWM charge controller.It require some basic knowledge of power electronics.

I put a lot of effort to make it simple, so that any one can understand it easily.If you are aware about the basics of MPPT charge controller then skip the first few steps.

The Maximum Power Point Tracker (MPPT) circuit is based around a synchronous buck converter circuit..It steps the higher solar panel voltage down to the charging voltage of the battery. The Arduino tries to maximize the watts input from the solar panel by controlling the duty cycle to keep the solar panel operating at its Maximum Power Point.

#### Specification of version-3 charge controller :

- 1.Based on MPPT algorithm
2. LED indication for the state of charge
3. 20x4 character LCD display for displaying voltages,current,power etc
4. Overvoltage / Lightning protection
5. Reverse power flow protection
6. Short Circuit and Over load protection
7. Wi Fi data logging

8.USB port for Charging Smart Phone /Gadgets

#### **Electrical specifications :**

- 1.Rated Voltage= 12V
- 2.Maximum current = 5A
- 3.Maximum load current =10A
4. In put Voltage = Solar panel with Open circuit voltage from 12 to 25V

This project is consists of 40 steps.So for simplicity I divided the entire project in to small sections.Click on the link which you want to see.

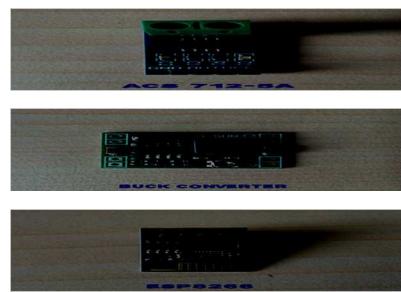
1. Basics on MPPT charge controller
2. Buck circuit working and design calculation
3. Testing the Buck Circuit
4. Voltage and Current Measurements
5. LCD display and LED indication
6. Making the Charging Board
7. Making the Enclosure
8. Making the USB Charging Circuit
9. Wi Fi Data Logging
10. MPPT algorithm and flow chart

*I have spent a lot of time to make this charge controller project.If you like it ,please vote for me in all the competitions.It will be very helpful for me.Thank you in advance.*



Remove these ads by [Signing Up](#)

## **Step 1: PARTS AND TOOLS REQUIRED:**



Show All 24 Items

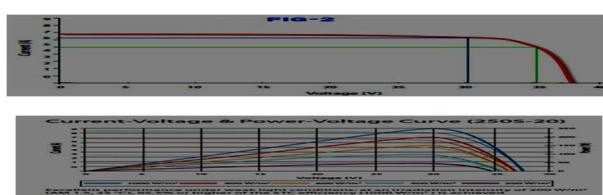
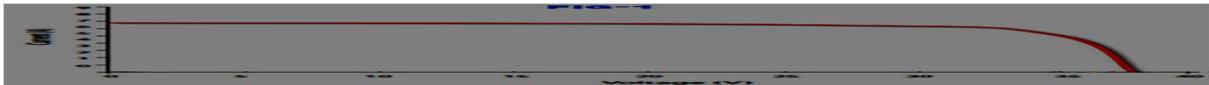
1. Arduino Nano ([eBay](#))
2. Current Sensor ([ACS712-5A](#))
3. Buck Converter ([LM2596](#))
4. Wifi Module ([ESP8266](#))
5. LCD display ([20x4 I2C](#))
6. MOSFETs (4x [IRFZ44N](#))

7. MOSFET driver ( **IR2104** )
8. 3.3V Linear regulator ( **AMS 1117** )
9. Transistor ( **2N2222** )
10. Diodes ( 2x **IN4148** , 1 x **UF4007** )
11. TVS diode ( 2x **P6KE36CA** )
12. Resistors ( 3 x 200R , 3 x 330R, 1 x 1K, 2 x 10K, 2 x 20K, 2x 100k, 1x 470K )
13. Capacitors ( 4 x 0.1 uF, 3 x 10uF ,1 x 100 uF ,1x 220uF)
14. Inductor ( 1x **33uH -5A** )
15. LEDs ( 1 x Red ,1 x Yellow ,1 x Green )
16. Prototype Board
17. Wires and Jumper wires ( **Female -Female** )
18. Header Pins ( **Male Straight ,female , Right angle** )
19. DIP Socket ( **8 pin** )
19. Screw Terminals ( 3 x **2 pin** ,1 x **6pin** )
20. Fuses ( 2 x **5A**)
21. Fuse Holders (**2 nos**)
22. Push Switch ( **2 nos**)
23. Rocker /Toggle Switch ( **1 no**)
24. Female USB port ( **1no**)
25. JST connector ( **2pin male -female** )
26. Heat Sinks
27. Enclosure
28. Plastic Base and studs
29. Screws/Nuts/Bolts

#### **TOOLS REQUIRED :**

1. Soldering Iron
2. Glue Gun
3. Dremel
4. Hobby Knife
5. Wire Cutter
6. Wire Stripper
7. Screw Driver
8. Ruler and pencil

## Step 2: Basics on MPPT charge controller



A solar panel will generate different voltages depending on the different parameters like :

- 1.The amount of sun light
- 2.The connected load
- 3.The temperature of the solar panel.

Throughout the day, as the weather changes, the voltage produced by the solar panel will be constantly varying. Now, for any given voltage, the solar panel will also produce a current (Amps). The amount of Amps that are produced for any given voltage is determined by a graph called an **IV curve**, which can be found on any solar panel's specification sheet and typically looks like the figure-1 shown above.

In the above figure-2, the blue line shows a solar panel voltage of 30V corresponding to a Current of about 6.2A. The green line shows a Voltage of 35V corresponds to a current of 5A.

We know that  $\text{Power} = V \times I$

In the picture shown above as you move along the red curve above you will find one point where the Voltage multiplied by its corresponding Current is higher than anywhere else on the curve. This is called the solar panel's Maximum Power Point(**MPP**).

### What Is MPPT ?

MPPT stands for Maximum Power Point Tracking. MPPT charge controllers used for extracting maximum available power from PV module under certain conditions. In order to understand this in details we first need to look at the power curve characteristics of a solar panel. Look at the image shown above. I have downloaded this images from web to explain the MPPT. Till now we have seen that the maximum power point (MPP) of a solar panel lies at the knee of the current and voltage curve.

A 12V solar panel is not really a 12V panel at all. Its really a somewhere in between 12V and 21V panel depending on what load is connected to it and how bright the sunlight is. The panel has an internal resistance which changes dynamically with differing irradiance levels. Solar panels will only deliver their rated power at one specific voltage and load, and this voltage and load moves around as the sunlight intensity changes.

For example take a solar panel rated at 100 watts, 18V at 5.55 amps.

The 18 V at 5.5 amps means that the Solar panel wants to see a load of  $18/5.5 = 3.24$  ohms.

With any other load the panel will deliver less than 100 watts. So if a static load is connected directly to a panel and its resistance is higher or lower than the panels internal resistance at MPP, then the power drawn from the panel will be less than the maximum available.

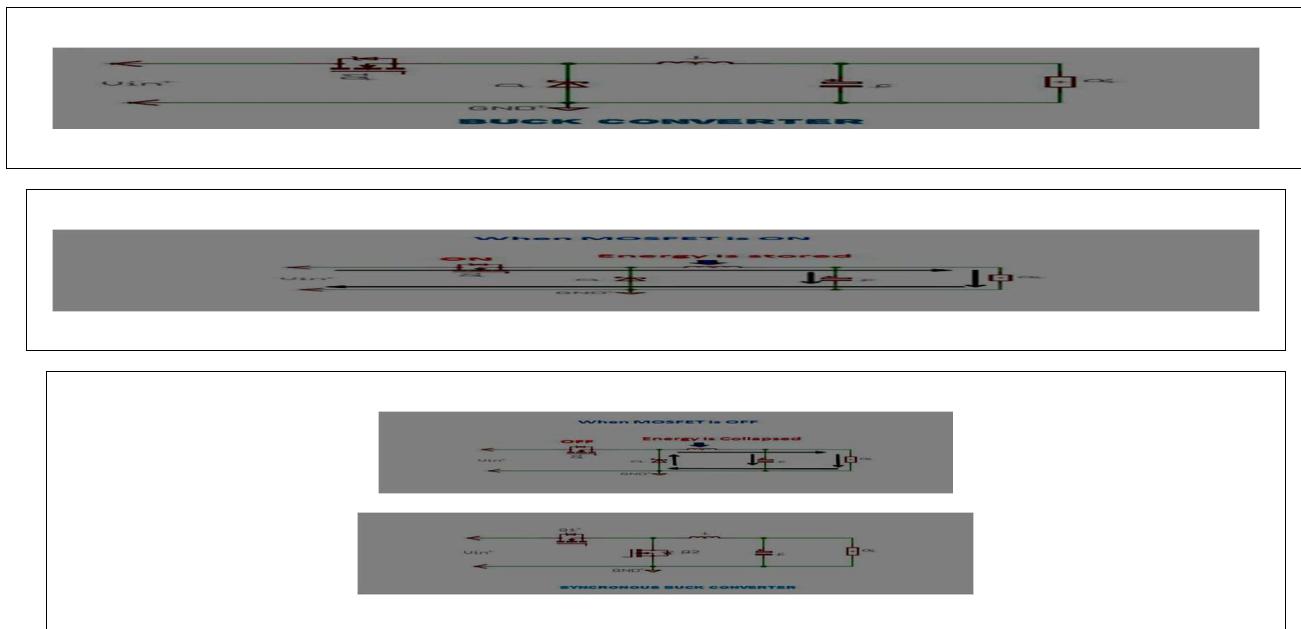
Taking a simple example say we connected the above 100W panel directly to a 12V lead acid battery, the panel voltage would be dragged down near to the load voltage of the battery as the batteries resistance is lower than the panels, but the current stays the same at 5.55 amps. This happens because Solar Panels behave like current sources, so the current is determined by the available sunlight.

Now the power ( $P$ ) =  $V \times I = 12 \times 5.55 = 66.6$  W. So the Solar panel is now behaving like a 66 watt panel.

This equates to a loss of  $100W - 66.6W = 34W$  ( 33.4%).

This is where MPPT comes into play. MPPT circuits can be based on various switch mode power supply (SMPS) topologies, they generally have a fixed frequency but varying duty cycle. The duty cycle is controlled via an algorithm so as to track the changing MPP.

### Step 3: BUCK CONVERTER WORKING



A buck converter is basically a DC to DC converter. The main principle at work in a buck converter, is the tendency for an inductor to resist changes in current. A buck converter output voltage will always be lower or the same as the input voltage. A simplified schematic of a buck converter is shown in the above picture.

#### Working Principle :

##### When the MOSFET is ON

When the mosfets is ON, current flows through the inductor (L) in a clockwise direction

into the load (R) and also charging the output capacitor (C). At this point the voltage on the cathode of the diode is positive, therefore the diode (D) is blocking any flow of current and is said to be reverse biased. The initial current flow into the load from  $V_{in}$  is slow, as energy is stored in the inductor as its magnetic field increases. So during the ON state of the mosfet, energy is stored in the inductor.

### **When the MOSFET is OFF**

When the mosfet is switched OFF, the voltage across the inductor is reversed. The inductors magnetic field begins to collapse, this collapse releases the stored energy allowing current to flow from the inductor into the load. The diode now has a negative voltage on the cathode so becomes forward biased. Therefore the inductors discharge current flows in a clockwise direction through the load and back through the diode. Once the inductors energy has fallen below a certain threshold, the load voltage falls and the capacitor becomes the main source of current, ensuring the load is still supplied until the next switching cycle begins. To ensure continuous conduction mode the inductor is must not be fully discharged before the mosfet is switch on again, and the cycle repeats.

### **What is Synchronous Buck Converter ?**

The synchronous design simply replaces the diode with a second mosfet, this eliminates the losses incurred by the forward voltage drop across the diode, thus making the circuit more efficient. This is slightly more complex to implement, as the second mosfet switching needs to be carefully timed with the switching of the first mosfet. It is essential to ensure that both are never on at the same time, or the current will have a direct path to ground, effectively causing a short circuit. The mosfet switching is effectively 180 degrees out of phase, with a short delay period between each transition referred to as a Dead-Band.

## **Step 4: BUCK CONVERTER DESIGN**

The first step was to design the buck converter circuit, this is determined by the output parameters of the system and its load. It designed for a 50W solar panel, it was decided to aim for a 12 V output ( battery voltage)

When calculating a buck circuit the frequency of operation, inductor size and output capacitor size are important, as they determine the current and voltage ripple size. It is desirable to have as smaller current and voltage ripple as possible.

A general rule is the higher the frequency the smaller the size of the inductor and output capacitor, and a smaller inductor and capacitor size generally lowers the system cost. However higher PWM frequencies decrease the system efficiency due to switching losses in the mosfets, so a trade off has to be reached which meets the design constraints of the end system.

For this design a PWM frequency of 50kHz was chosen.

From the earlier discussion we have conclude that a buck converter is consist of

- 1.Inductor

2.Capacitor

3.MOSFETS

In the next few steps I will discuss how to choose these components.

## Step 5: INDUCTOR CALCULATION



Calculating the inductor value is most critical in designing a buck converter. First, assume the converter is in continuous current mode (CCM), which is usually the case. CCM implies that the inductor does not fully discharge during the switch-off time. The following equations assume an ideal switch (zero on-resistance, infinite off-resistance and zero switching time) and an ideal diode.

### Assume

We are designing for a 50W solar panel and 12V battery

Input voltage ( $V_{in}$ ) = 15V

Output Voltage ( $V_{out}$ ) = 12V

Output current ( $I_{out}$ ) =  $50W/12V = 4.16A = 4.2A$  (approx)

Switching Frequency ( $F_{sw}$ ) = 50 KHz

Duty Cycle ( $D$ ) =  $V_{out}/V_{in} = 12/15 = 0.8$  or 80%

### Calculation

$$L = (V_{in} - V_{out}) \times D \times 1/F_{sw} \times 1/dI$$

Where  $dI$  is Ripple current

For a good design typical value of ripple current is in between 30 to 40 % of load current.

Let  $dI = 35\%$  of rated current

$$dI = 35\% \text{ of } 4.2 = 0.35 \times 4.2 = 1.47A$$

$$\text{So } L = (15.0 - 12.0) \times 0.8 \times (1/50k) \times (1/1.47) = 32.65\mu H = 33\mu H \text{ (approx)}$$

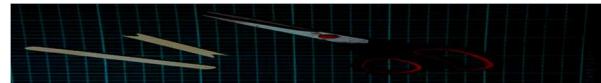
$$\text{Inductor peak current} = I_{out} + dI/2 = 4.2 + (1.47/2) = 4.935A = 5A \text{ (approx)}$$

So we have to buy or make a toroid inductor of 33 $\mu$ H and 5A.

You can also use a buck converter design [calculator](#)

So 33uH is enough for our design.

## Step 6: HOW TO WIND A TOROIDAL INDUCTOR



### Show All 7 Items

I have collected a bunch of toroidal cores from old computer power supply. So I thought to made the inductor at my home. Though it took a lot of time to make, but I learned a lot and enjoyed during making. These are few tricks what I learned during the making, so that you can make it easily.

#### How to Wind the wire :

Winding by hand is very painful for skin as well as you can't make the winding so tight. So I made a simple tool from popsicle stick for winding the toroidal core. This simple tool is very handy and you can make perfect and tight winding. Before making the inductor you have to know the core specification and number of turns.

The important parameters of toroidal core are

1. Outer diameter(OD)
2. Inner diameter(ID)
3. Height (H)

#### 4.Al value

As I did not know the part number,I used a indirect method to identify it.First I measure the OD and ID of the unknown core by using my vernier caliper,it was around

OD= 23.9mm (.94") , ID= 14.2mm(.56") ,H= 7.9mm( .31") and yellow white in color.

I used a toroid core chart (page-8) to identify the unknown core.I have attached this toroid size chart in the bellow.It contains a lot of information for the inductor design.The PDF version is attached bellow.

#### Finding the part number :

I searched the **Physical dimension** table from the chart. From the table it was found that the core is **T94**

#### Finding the mix number :

The color of the core is indication for mix number.As my core is is yellow/white in color,it is confirmed that the mix number is 26

So the unknown core is T94-26

#### Finding Al value :

From the Al value table for a T94-26 core it is 590 in uH/100 turns.

After selecting the core now time to find out the number of turns required to obtain the desired inductance.

**Number of turn (N) = 100 x sqrt( desired inductance in uH / Al in uH per 100 turns)**

$$\Rightarrow N = 100 \sqrt{33/590} = 23.65 = \text{approximately 24 turns}$$

You can also use [this](#) online calculator for finding the number of turns.Only you have to know the part number and mix number.

Then I wind a 20 AWG copper wire (24 turns) around the the toroid core.At the both end of the winding leave some extra wire for connection lead.After this remove the enamel insulation from the lead. I used my leatherman file for removing the insulation. See the above picture for better understanding.

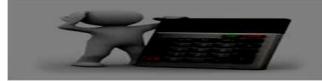
**Note :** Making a good inductor is not so simple.I am still in learning stage.If you are not so confident I will recommend to buy a ready made inductor.



toroid core size.pdf4 MB

### Step 7: CAPACITOR CALCULATION





Output capacitance is required to minimize the voltage overshoot and ripple present at the output of a buck converter. Large overshoots are caused by insufficient output capacitance, and large voltage ripple is caused by insufficient capacitance as well as a high equivalent-series resistance (ESR) in the output capacitor. Thus, to meet the ripple specification for a buck converter circuit, you must include an output capacitor with ample capacitance and low ESR.

### Calculation :

The output capacitor ( $C_{out}$ ) =  $dI / (8 \times F_{sw} \times dV)$

Where  $dV$  is ripple voltage

Let voltage ripple(  $dV$  ) = 20mV

$C_{out} = 1.47 / (8 \times 50000 \times 0.02) = 183.75 \mu F$

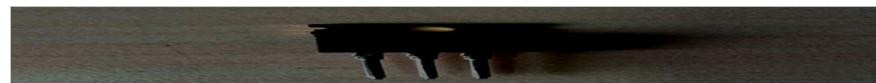
By taking some margin, I select 220uF electrolytic capacitor.

The equations used for calculation of inductor and capacitor is taken from a article [LC Selection Guide for the DC-DC Synchronous Buck Converter](#)



LC selection.PDF 889 KB

## Step 8: MOSFET SELECTION



The vital component of a buck converter is MOSFET. Choosing a right MOSFET from the variety of it available in the market is quite challenging task.

These are few basic parameters for selecting right MOSFET.

**1. Voltage Rating :**  $V_{ds}$  of MOSFET should be greater than 20% or more than the rated voltage.

**2. Current Rating:**  $I_{ds}$  of MOSFET should be greater than 20% or more than the rated current.

**3.ON Resistance (Rds on) :** Select a MOSFET with low ON Resistance (Ron)

**4.Conduction Loss :** It depends on Rds(ON) and duty cycle.Keep the conduction loss minimum.

**5.Switching Loss:** Switching loss occurs during the transition phase.It depends on switching frequency,voltage ,current etc.Try to keep it minimum.

These are few links where you can get more information on selecting the right MOSFET.

1.[MOSFET selection for Buck Converter](#)

2.[A simple guide to selecting power MOSFETs](#)

In our design the maximum voltage is solar panel open circuit voltage(Voc) which is nearly 21 to 25V and maximum load current is 5A.

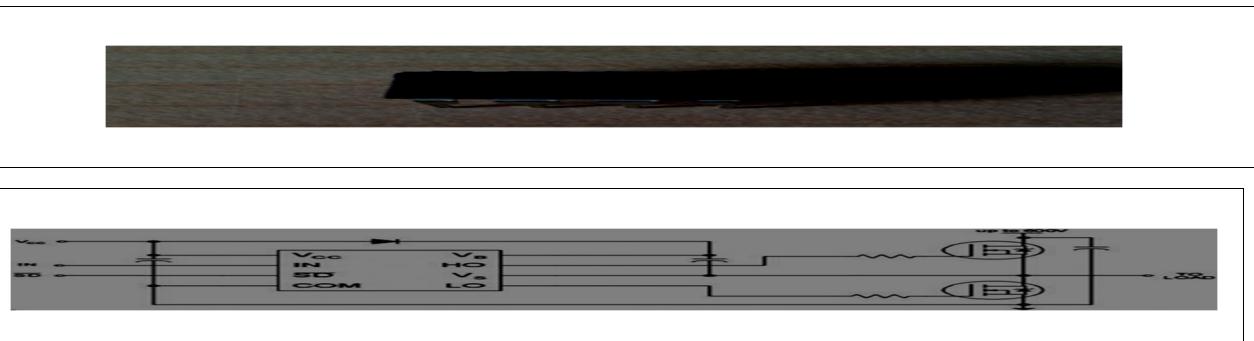
I have chosen IRFZ44N MOSFET. The Vds and Ids value have enough margin as well as it has low Rds(On) value.

You can check the other parameters of IRFZ44N from the [data sheet](#)



IRFZ44N DATA SHEET.pdf104 KB

## Step 9: MOSFET DRIVER



Why we need a gate driver ?

A Mosfet driver allows a low current digital output signal from a Microcontroller to drive the gate of a Mosfet. A 5 volt digital signal can switch a high voltage mosfet using the driver.A MOSFET has a gate capacitance that you need to charge so that the MOSFET can turn on and discharge it to switch off,the more current you can provide to the gate the faster you switching on/off the mosfet, that is why you use a driver.

For more details you can read about [MOSFET Basics](#)

For this design I am using a IR2104 Half Bridge driver. The IC takes the incoming PWM signal from the micro controller, and then drives two outputs for a High and a Low Side MOSFET.

### How to use it ?

From the data sheet I have taken the image shown above.

**Input :**

First we have to provide power to the gate driver. It is give on Vcc (pin-1) and its value is in between 10-20V as per data sheet.

The high frequency PWM signal from Arduino goes to IN (pin-2) . The shut down control signal from the Arduino is connected on SD ( pin 3).

### **Output :**

The 2 output PWM signals are generated from HI and LO pin. This gives the user the opportunity to fine tune the dead-band switching of the MOSFETs.

### **Charge Pump Circuit :**

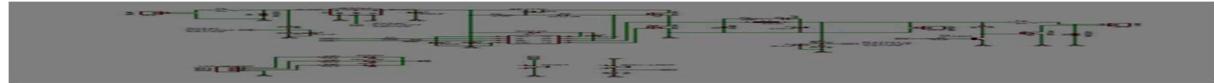
The capacitor connected between VB and VS along with the diode form the charge pump. This circuit doubles the input voltage so the high switch can be driven on. However this bootstrap circuit only works when the MOSFETs are switching.

The data sheet of IR2104 is attached [here](#)



**IR2104 data sheet.pdf** 137 KB

## **Step 10: SCHEMATIC AND WORKING**



The input power connector to the solar panels is the screw terminal JP1 and JP2 is the output screw terminal connector to the battery. The third connector JP3 is connection for the load.

F1 and F2 are the 5A safety fuses.

The buck converter is made up of the synchronous MOSFET switches Q2 and Q3 and the energy storage devices inductor L1 and capacitors C1 and C2. The inductor smooths the switching current and along with C2 it smooths the output voltage. Capacitor C8 and R6 are a snubber network, used to cut down on the ringing of the inductor voltage generated by the switching current in the inductor.

The third MOSFET Q1 is added to allow the system to block the battery power from flowing back into the solar panels at night. In my earlier charge controller, this is done by a diode in the power path. As all diodes have a voltage drop a MOSFET is much more efficient. Q1 turns on when Q2 is on from voltage through D1. R1 drains the voltage off the gate of Q1 so it turns off when Q2 turns off.

The diode D3 (UF4007) is an ultra fast diode that will start conducting current before Q3 turns on. It is supposed to make the converter more efficient.

The IC IR2104 is a half bridge MOSFET gate driver. It drives the high and the low side MOSFETs using the PWM signal from the arduino (Pin -D9) .The IR2104 can also be shut down with the control signal (low on pin -D8) from the Arduino on pin 3. D2 and C7 are part of the bootstrap circuit that generates the high side gate drive voltage for Q1 and Q2. The software keeps track of the PWM duty cycle and never allows 100% or always on. It

caps the PWM duty cycle at 99.9% to keep the charge pump working.

There are two voltage divider circuits( R1,R2 and R3,R4) to measure the solar panel and battery voltages.The out put from the dividers are feeds the voltage signal to Analog pin-0 and Analog pin-2 .The ceramic capacitors C3 and C4 are used to remove high frequency spikes.

The mosfet Q4 is used to control the load.The driver for this mosfet is consists of a transistor and resistors R9 ,R10.

The diode D4 and D5 are TVS diodes used for over voltage protection from solar panel and load side.

The current sensor ACS712 sense the current from the solar panel and feeds to the Arduino analog pin-1.

The 3 LEDs are connected to the digital pins of the microcontroller and serve as an output interface to display the charging state.

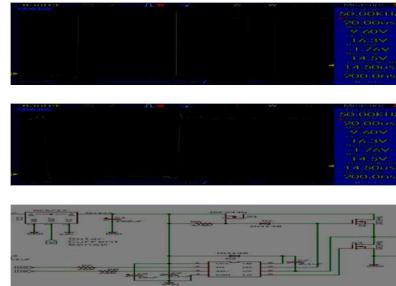
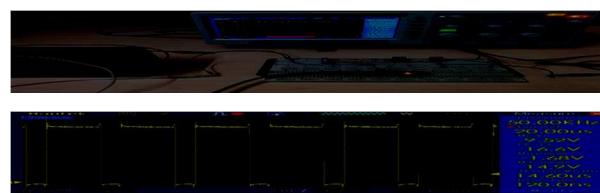
Reset switch is helpful if the code gets stuck.

The back light switch is to control the back light of LCD display.



[mppt controller\\_schema.pdf](#) 31 KB

## Step 11: Test the gate driver and MOSFETs Switching



Hey I think I have talked a lot on the theory.So lets do some practical.

As I have told earlier the heart of the MPPT charge controller is Buck Converter. As per me if your buck converter circuit work perfectly. You can do the rest thing easily. So first lets test the Mosfets switching and the driver.

Before soldering ,I request to do it on a bread board.I have blown lot of MOSFETs during my testing. So be careful during the connection.

Connect the everything as per schematic given above. Now you can omit the TVS diode,current sensor and voltage divider.

After connecting everything test the resistance between the input rail. It should be several KOhm. If you get resistance bellow 1K then recheck the circuit connection.

Upload the test sketch to the Arduino. The code in the form of text file is attached bellow.

Then connect the scope in between the source of Q1 and GND.

The result should be a PWM with frequency 50KHz.

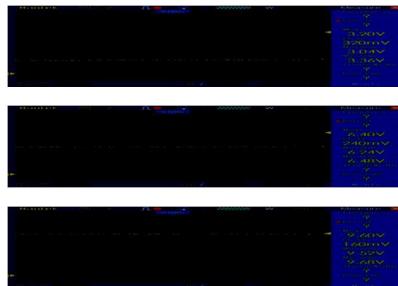
The waveform obtained during my testing are shown above.

If everything goes right then proceed to complete the bulk converter circuit.( i.e adding inductor and capacitor)



test\_code.txt 843 bytes

## Step 12: Test the Buck Converter



### Show All 8 Items

In the previous steps we have calculated the inductor and capacitor rating. Now it is time to using and testing it.

Add the 33uH inductor and 100uf input and 220uF out put electrolytic capacitor as per schematic. You can also use 0.1uF ceramic capacitors parallel with input and output capacitors. It will give better result. But it is not mandatory.

Then make the snubber circuit by using a 0.1uF ceramic capacitor and 200ohm resistor.

Again check the resistance in between the input rail. It should be order of K ohm.

Now give power to the input rail and Arduino.

Connect the probe of your scope in between the output capacitor.

The result is shown above . The out put should be a steady DC.

$$V_{out} = \text{Duty Cycle} \times V_{in}$$

For example if i give 50% duty cycle to a 12 input supply, the output should be 6V in the scope.

After confirmed that everything working fine, now we can add the blocking mosfet Q1. It is used to block reverse power from battery to the solar panel during night.

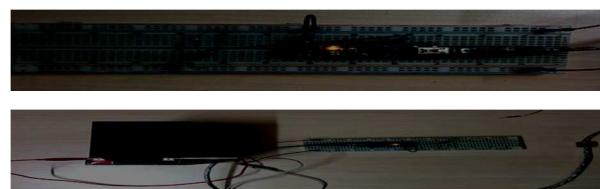
Add the third mosfet Q3 as per schematic. Then place the 470k resistance and diode IN4148.

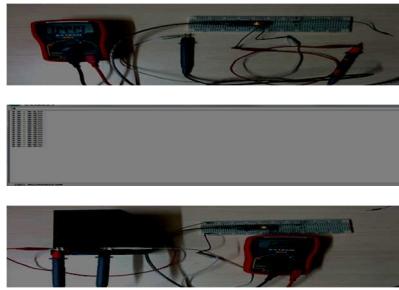
Again check the output it should be same.

At last put the scope in between the gate of Q1 and Gnd.

Do you know ? you have done the most critical part of this project.

## Step 13: VOLTAGE MEASUREMENT





### Show All 7 Items

#### Voltage Measurement :

As you may well know, Arduino's analog inputs can be used to measure DC voltage between 0 and 5V (when using the standard 5V analog reference voltage) and this range can be increased by using two resistors to create a voltage divider. The voltage divider decreases the voltage being measured to within the range of the Arduino analog inputs. We can use this to measure the solar panel and battery voltages.

For a voltage divider circuit

$$V_{out} = R_2/(R_1+R_2) \times V_{in}$$

$$V_{in} = (R_1+R_2)/R_2 \times V_{out}$$

The `analogRead()` function reads the voltage and converts it to a number between 0 and 1023

#### Example code :

```
// read the input on analog pin 0 ( You can use any pin from A0 to A5)
int Value = analogRead(A0);
Serial.println(value);
```

The bove code gives an ADC value in between 0 to 1023

#### Calibration :

We're going to read output value with one of the analog inputs of Arduino and its `analogRead()` function. That function outputs a value between 0 (0V in input) and 1023 (5V input) that is 0,0049V for each increment (As  $5/1024 = 0.0049V$ )

$$V_{in} = V_{out} * (R_1+R_2)/R_2 ; R_1=100k \text{ and } R_2=20k$$

$$V_{in} = \text{ADC count} * \mathbf{0.0049} * (\mathbf{120/20}) \text{ Volt} // \text{Highlighted part is Scale factor}$$

**Note :** This leads us to believe that a reading of 1023 corresponds to an input voltage of exactly 5.000 volts.

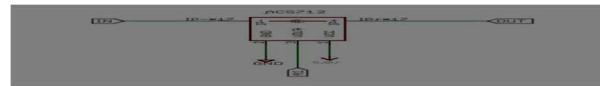
*In practical you may not get 5V always from the arduino pin 5V .So during calibration first measure the voltage between the 5v and GND pins of arduino by using a multimeter, and use  $IADC = \text{measured voltage}/1024$  instead of  $5/1024$*

Check your voltage sensor by a test code attached bellow



Voltage\_mesurement\_code.ino 784 bytes

## Step 14: CURRENT MEASUREMENT



For current measurement I used a Hall Effect current sensor ACS 712 (5A).

The ACS712 sensor read the current value and convert it into a relevant voltage value, The value that links the two measurements is sensitivity. You can find it on the [datasheet](#).

**As per data sheet for a ACS 712 (5A) model :**

1. Sensitivity is 185mV/A.
2. The sensor can measure positive and negative currents (range -5A...5A),
3. Power supply is 5V
4. Middle sensing voltage is 2.5V when no current.

**Calibration:**

Value = (5/1024)\*analog read value

// If you are not getting 5V from arduino 5V pin then, value = ( Vmeasured/1024 ) \* analog read value

// Vmeasured is the voltage in between Arduino pin 5V and GND. You can measure it by a multimeter.

But as per data sheets offset is 2.5V (When current zero you will get 2.5V from the sensor's output)

Current in amp = (value-2.5)/0.185

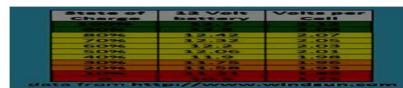
Test it by a sample code for ACS712 attached bellow.



ACS\_712\_code.ino

791 bytes

## Step 15: LCD Display and LED Indication



LCD display :

A 20X4 char LCD is used for monitoring solar panel, battery and load parameters. For simplicity a I2C LCD display is chosen. It needs only 4 wires to interface with the arduino. In my earlier design the LCD was consuming lot of power. The main cause was LCD back light. So I add a push switch to control the back light. By default the back light will be in off condition. If the user press the switch then it will be on for 15 secs and again goes off.

Vcc--> 5V , GND-->GND, SDA-->A4 and SCL-->A5

Column-1 : Solar panel voltage, Current and Power

Column-2 : Battery Voltage, Charger state and SOC

Column-3 : PWM duty cycle and load status

For testing the LCD download the test code attached below.

You download the library from [LiquidCrystal\\_I2C](#) .

### LED indication :

Red ,Green and Yellow leds are used to indicate the battery voltage level.

Low Voltage -- > Red led

Normal Voltage --> Green Led

Fully Charged --> Yellow Led



LCD\_code\_test.ino2 KB

## Step 16: HARDWARE AND SOLDERING



Before soldering you should clear about the Power and Control Signal. Do not mix up between them. Otherwise you will fry everything.

### Power Signal :

1. Solar panel -> Fuse -> Current sensor -> Mosfets Q1,Q2 ,Q3 -> Inductor -> Battery.
2. Battery -> Fuse -> Load -> Mosfet Q4

### Control Signals :

- 1.Signal from the different Sensors to Arduino
2. Signals from the Arduino to the Mosfet drivers,LED,LCD
3. Signal between the Arduino and ESP8266

I used red and black thick wires ( 0.5 to 0.75 sq mm) for power and ground connections respectively.

All the colored thin wires are for control signals.

**Tips:** Print the PDF format Schematics before soldering. Keep it in front of you during soldering for reference.

## Step 17: Drill Holes for mounting



First hold the prototype board by a vice.

Then drill 4 holes (3mm) at the 4 corners of the prototype board.

## Step 18: Add the Input and out put terminals :



First solder the three screw terminals for solar panel,battery and load connection.

The left one is for solar panel,middle one is for battery and the right one is for load connection.

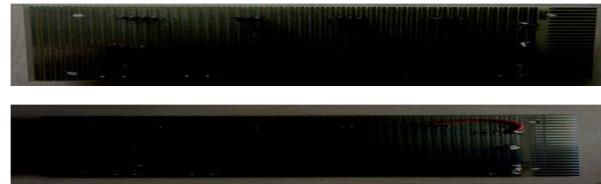
## Step 19: Add the Fuse Holders



On the extreme left and right solder the two fuse holders.( One in the solar panel side and other on the load side)

Then connect the left terminal of the solar screw terminal with one leg of the fuse holder.

## Step 20: Solder the MOSFETS and Input Capacitor



Solder all the 4 MOSFETs with equally spaced on the top of the prototype board.(Leave some space to putting the heat sinks)

Then add the input 100uF capacitor.I left some space in between the fuse holder and Capacitor for installing the current sensor later.

Solder connecting wires as follows :

Between positive terminal of input capacitor(C) and source of mosfet Q1.

Between drains of mosfet Q1 and Q2.

Then in between source of Q2 and drain of Q3.

## Step 21: Mounting the Arduino Nano





### Show All 7 Items

First cut two rows of female and male header pin with 15 pins in each.I used a diagonal nipper to cut the headers.

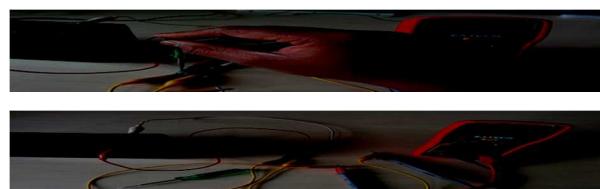
Then solder the male header pins.Be sure the distance between the two rails fits the arduino nano.

Leave two rows on each side of the female header and then solder the two male headers.

Then short the corresponding male and female pins.Though I forgot this during my soldering.

The female headers is used to mount the Arduino nano and male headers are used for external connection with the Arduino.

## Step 22: Make The Power supply





### Show All 11 Items

To run the Arduino ,different sensors,LED,LCD and the wifi module( ESP8266 ) we need power.

Except ESP8266 module all the others can be run by 5V power supply.The ES8266 module need power not more than 3.7V. It is recommended to run it on 3.3V. Though Arduino Nano have 3.3V pin but it can not provide sufficient power ( around 200mA to 300mA) to run the ESP8266 module.So we need a separate 3.3V power supply which can provide at least 300mA current.

#### 5V Power Supply :

In my previous version I used a LM7805 linear voltage regulator to step down the battery voltage to 5V for the power supply.But it produces a lot of heat during its working.So I used a high efficient buck converter in this design.

Adjust the output voltage of buck converter :

First connect the battery on the input terminal of the buck converter and adjust the potentiometer to get 5V out put.

See the above picture.

Cut 4 pcs of male header with 2pins in each.Solder the headers as per the holes given in the converter.

Place the converter on the above 4 header pin and solder on the top.Be sure the input side is toward the battery screw terminal.

Add the output capacitor(C2) near to the battery screw terminal.The positive terminal of the capacitor should be on the left.

Then connect the input of the buck converter to the battery screw terminal and output to the 5V and GND pin of the Arduino Nano.At this stage you can check it.Place the Arduino nano on the header pin and connect the 12V battery to the screw terminal.If everything is correct then Arduino power led should glow.

Finally add two rows of male header pins to the side of Arduino 5V and GND pin for external connection.

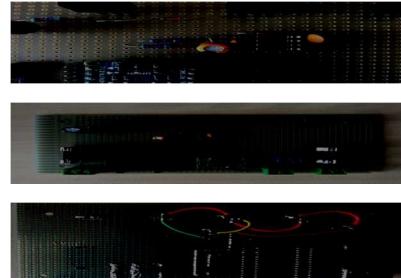
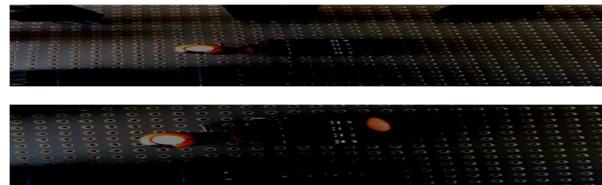
#### 3.3V Power Supply :

I am planning to use a voltage regulator AMS1117 to step down from 5V to 3.3V.

Solder the voltage regulator first, then add two 10uF capacitors. One on the input and other on the output side.

See the above schematic.

## Step 23: Solder the Mosfet Driver circuit



### Show All 9 Items

First solder the 8 pins DIP socket just above the arduino header pins.

Add 10uF capacitor and and a 0.1uF capacitor in between the pin-1 and pin-4.

Solder the diode (D2) in between pin -1 and 8.The diode cathode should be connect to the pin-8.

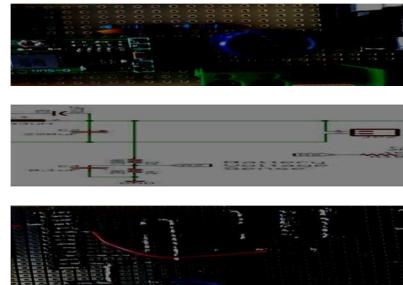
Solder the capacitor (C7) in between pin-8 and pin-6.

Solder two 200ohm resistors ( R7 and R8) just side to the pin-2 and pin-3.

Solder one 470K resistor (R1) near to the mosfet Q1 and a diode (D1) in between gates of mosfets Q1 and Q2.The diode cathode connects to the gate of Q1.

After this complete the circuit by soldering wires as per the schematics.

## Step 24: Solder the Voltage Sensors



#### Show All 7 Items

Solder solar panel voltage divider near to the fuse and battery voltage divider near to the output capacitor.

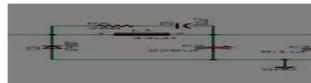
Then solder two ceramic capacitors ( C3 and C4) across the 20k resistors.

Then solder a wire between middle point of the solar panel side voltage divider and arduino pin A0.

Finally solder a wire between middle point of the battery side voltage divider and arduino pin A2.

### Step 25: Solder The Inductor and Snubber Circuit





First solder the resistor (R6) and capacitor (C8) in series just above the output capacitor( C2).

Then solder the inductor parallel to it.

Inductor is the heavier component in the entire circuit.To sit it firmly, apply glue at the base.

Then solder the ultra fast diode (D3) .

## Step 26: Solder the Load Mosfet (Q4) Driver

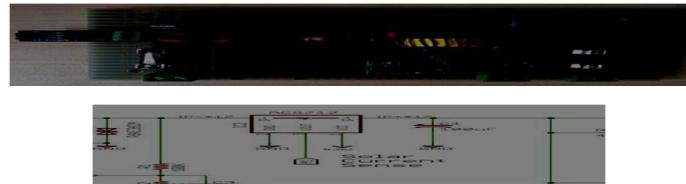


Solder the 2N2222 transistor near the gate of the mosfet (Q4).

Then add a 10k resistor (R9) near to the collector and a 1k resistor( R10) near to the base.

Then connect the points as per schematic.

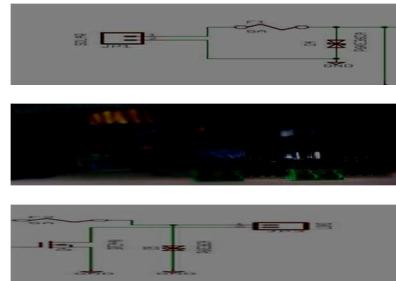
## Step 27: Adding The Current Sensor



Solder two thick wire in between the solar panel side fuse and capacitor (C1).

Then screw the wire in to the ACS712 screw terminal.

## Step 28: Solder the TVS diodes

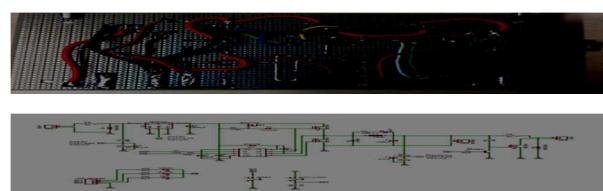


I do not have spare TVS diode. So I solder it later. You can solder it earlier also.

One TVS diodes, D4 near the connector JP1 and D5 near the connector JP3.

**Note :** I am using bidirectional TVS diode. So no polarity mark is there.

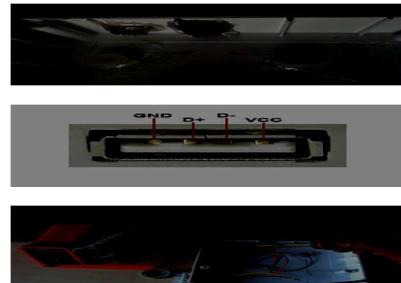
## Step 29: Connect the GND



After soldering all the components, connect all the grounds (GND) shown in the schematic.

I am using thick black wires.

## Step 30: Make the USB Charging Circuit



### Show All 9 Items

The buck converter used for power supply can deliver maximum current 3A. So the power supply have sufficient margin for charging the USB gadgets.

#### Make the Circuit :

Solder the male JST connector near to the buck converter and connect two pins with positive ( 5V )and negative

( GND ) out of the converter.See the picture.

Insert the USB port and switch in to the slots made earlier.Then apply hot glue surround them.

Solder the red wire (+ ve ) of the JST connector to one terminal of the switch.Then solder a small red wire between another terminal of switch and USB Vcc terminal.Finally solder the black wire (-ve ) of the JST connector to the USB GND.

For USB pin out see the above picture.

You can make this step earlier also.

## Step 31: Make the Wifi Module Circuit



First cut 2 female header with 4pins in each.

The solder it side by side near the load side fuse holder.

Complete the circuit as per schematic.

Be careful about when you solder this module. Voltage more than 3.7 V kill this module as it operates at 3.3 V .

Even the serial lines should not exceed this voltage.I am planning to use a 3.3 V regulator ( AMS1117 ) to power this module. A voltage divider circuit is used to drop the arduino Tx ( 5V ) to ESP8266 3.3 V ( RX ).

### Setting up the ESP8266 :

The first thing you want to do with ESP8266 is to establish communication.You can see this [example project](#) for setting up the ESP8266.Then connect it to your WiFi router.

Hey now you are ready to upload your data to the web.

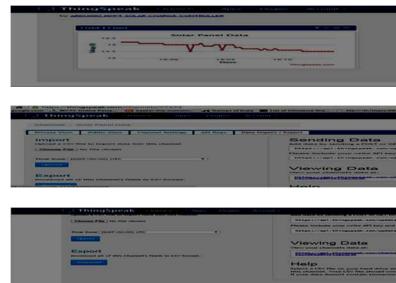
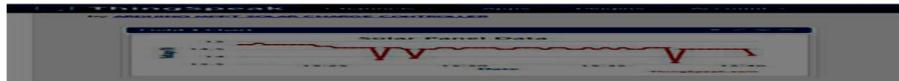
**Note:** I have not completed this part.Once completed,I will post the details along with code.

You can see the following projects to get some idea to use ESP8266 for data uploading to web.

<http://www.instructables.com/id/ESP8266-Wifi-Tempe...>

<http://www.element14.com/community/groups/internet...>

## Step 32: WiFi Data logging and Scientific Exploration



As the solar panel are installed at remote location, monitoring systems parameter is vital for us. This gives me the idea to add the data logging feature to my controller.

The WiFi module( ESP8266 ) automatically uploads live power generation, voltage, Current data to the Web( <https://thingspeak.com/> ). Then the web application graph and tabulate data in live. You can download the feeds from the website in the form of a Xcel sheet. Then explore these data for further analysis.

Go to Data Import/Export and then click on Download. See the above pics.

If you are app developer, then develop apps for Android, iPhone and Windows Mobile to see these useful data. If you make please share me. I am not a developer.

### Scientific Exploration of Data :

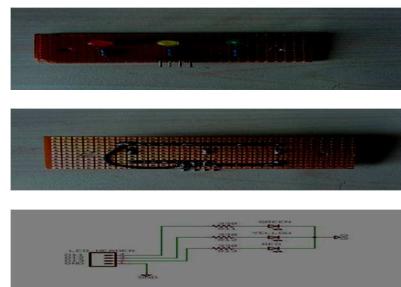
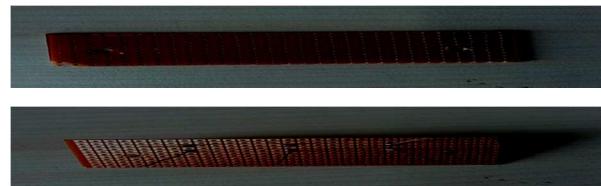
Science can be defined as the practice of observing the natural world, and trying to make objective sense of it by uncovering facts or cause-and-effect relationships.

By data logging, you can collect solar photo voltaic (PV) data that otherwise would be invisible to your senses. These are fun, interesting, and practical things to do- but doing them by themselves is not doing science. Suppose you'd like to learn more from what you uncover. Maybe you'd like to measure PV system parameters and atmospheric conditions ( temperature and solar irradiance ) over days or weeks, and then search similar solar installations within a geographical area. Compare daily data with another system, past date or postcode in different parts of your neighborhood, county, or state.



feeds.csv 21 KB

## Step 33: Make The LED Panel



Take a small size rectangular prototype board and drill holes at both end for mounting on the enclosure.

Solder the Leds with equally spaced.

Then solder the 330 ohm resistors (R11,R12 and R13) and 4pin male headers.

Finally complete the circuit as per schematics.

## Step 34: Make the Back light and Reset Switch





Take 5 female -female jumper wires and cut one side headers in all.

Insert heat shrink tube in all jumper wires.

#### **Reset Switch :**

Solder two jumper wires directly to the two pin of the push switch.

#### **Back Light Switch :**

Solder two jumper wires to the two pins of the switch.

Solder a 10k resistor to any one pin of the switch.

Then solder a jumper wire to the other end of the resistor.

Finally cover the joints with heat shrink tube and apply hot air.

### **Step 35: Prepare the Enclosure**



[Show All 12 Items](#)

I used a 6" x 8" plastic enclosure.

Mark the LCD,USB and Switch sizes .Then cut out the rectangular portion by using a dremel. Finally finish the edges by a hobby knife.

Then mark the mounting holes position for LCD,LED panel,Switches and External screw terminal by a pencil.

Drill holes at all the marked position.

**Note :** The holes size for LED is 5mm ,switches are 7mm and all other are 3mm.

## Step 36: Make the external connection Terminal



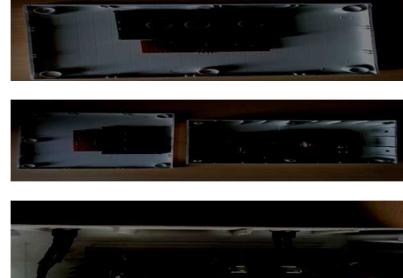
The external connector is used for outside access of all the 3 screw terminals in the controller board.

Mark the hole positions for mounting and 6 wires.

Then screw the wires in all the terminals.Use different color to distinguish between positive and negative terminal.

## Step 37: Mount Everything





To mount the controller board I used 4 plastic bases.Screw the main board over the base.

Mount the LCD and Led panel by screw and bolts.

Then mount the two switches.

## Step 38: Connect all the panel and switches



After mounting everything connect the panels,switches and external connector.

Use female-female jumper wires for connecting the panels.

Refer schematics for connection.

Finally box up the enclosure.

## Step 39: Software and Algorithm



The Maximum Power Tracker uses an iterative approach to finding this constantly changing MPP. This iterative method is called Perterb and Observe or hill climbing algorithm. To achieve MPPT, the controller adjusts the voltage by a small amount from the solar panel and measures power, if the power increases, further adjustments in the direction are tried until power no longer increases.

The voltage to the solar panel is increased initially, if the output power increase, the voltage is continually increased until the output power starts decreasing. Once the output power starts decreasing, the voltage to the solar panel decreased until maximum power is reached. This process is continued until the MPPT is attained. This result is an oscillation of the output power around the MPP.

The full software is attached bellow.

 **MPPT\_Code.ino** 23 KB

## Step 40: Conclusion

I have tried my best to make this instructable. Till now I am learning more on MPPT. So if I have done any mistakes please forgive me and raise a comments. I will rectify it as soon as possible.

I love getting feedback on my projects! The earlier version charge controllers has received a ton of feedback, and many users have posted pictures of their build. If you follow this Instructable and make your own controller, please share pictures and videos.

At last, I would like to give very special thanks to **timnolan**. As I have learned and used several things from his design.

For more updates and new projects subscribe me.

Thank you so much for reading my instructable.

1. Arduino Nano ([eBay](#))

2. Current Sensor ([ACS712-5A](#))

3. Buck Converter ([LM2596](#))

4. Wifi Module ([ESP8266](#))

5. LCD display ([20x4 I2C](#))

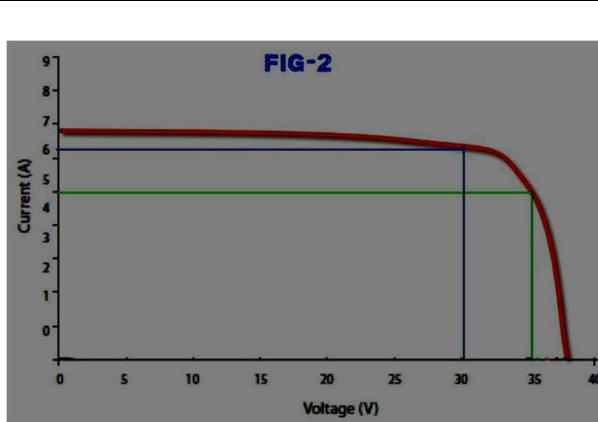
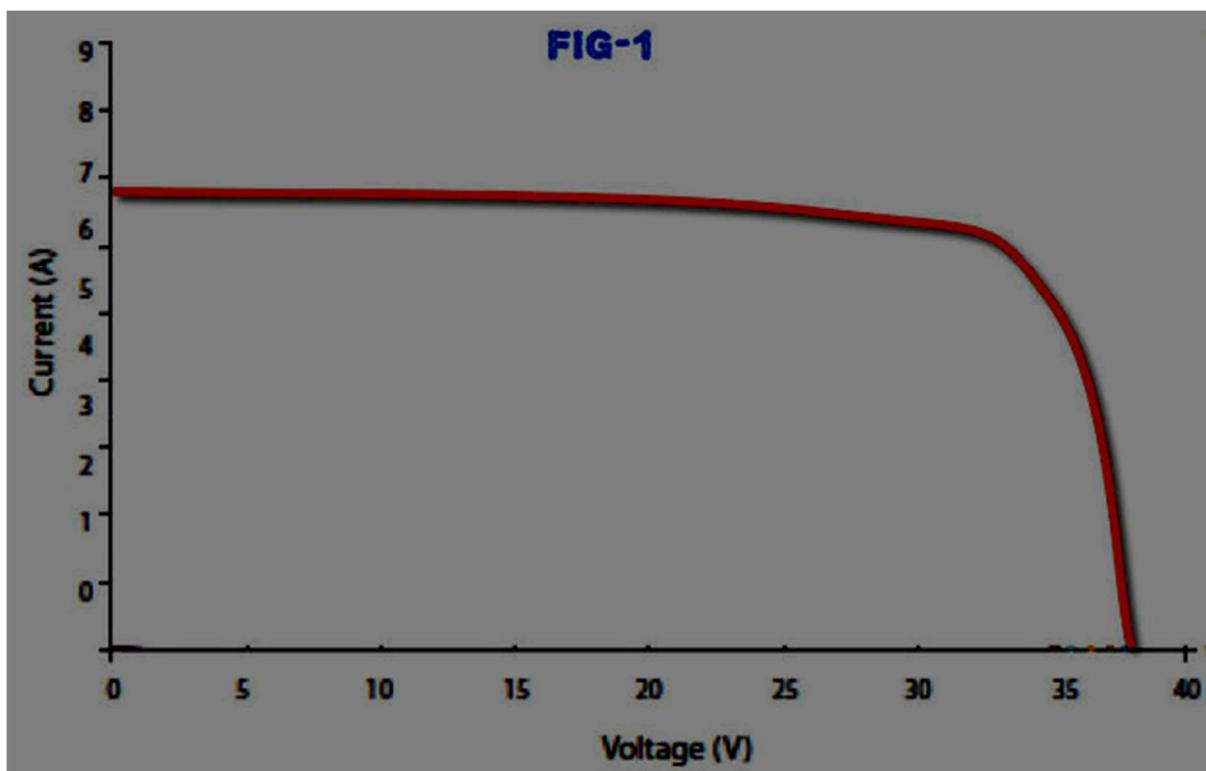
6. MOSFETs ( 4x [IRFZ44N](#) )

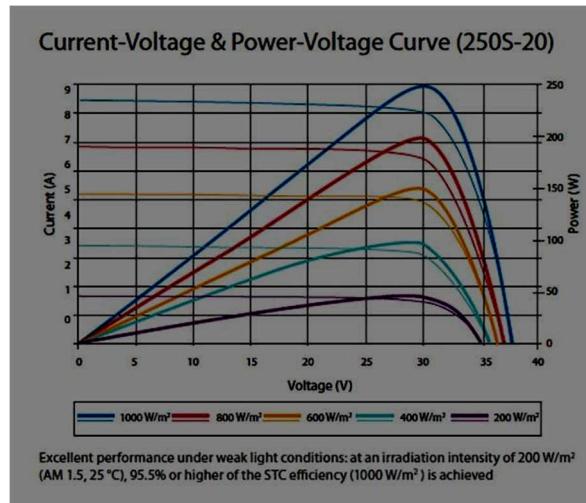
7. MOSFET driver ( **IR2104** )
8. 3.3V Linear regulator ( **AMS 1117** )
9. Transistor ( **2N2222** )
10. Diodes ( 2x **IN4148** , 1 x **UF4007** )
11. TVS diode ( 2x **P6KE36CA** )
12. Resistors ( 3 x 200R , 3 x 330R, 1 x 1K, 2 x 10K, 2 x 20K, 2x 100k, 1x 470K )
13. Capacitors ( 4 x 0.1 uF, 3 x 10uF ,1 x 100 uF ,1x 220uF)
14. Inductor ( 1x **33uH -5A** )
15. LEDs ( 1 x Red ,1 x Yellow ,1 x Green )
16. Prototype Board
17. Wires and Jumper wires ( **Female -Female** )
18. Header Pins ( **Male Straight ,female , Right angle** )
19. DIP Socket ( **8 pin** )
19. Screw Terminals ( 3 x **2 pin** ,1 x **6pin** )
20. Fuses ( 2 x **5A**)
21. Fuse Holders (**2 nos**)
22. Push Switch ( **2 nos**)
23. Rocker /Toggle Switch ( **1 no**)
24. Female USB port ( **1no**)
25. JST connector ( **2pin male -female** )
26. Heat Sinks
27. Enclosure
28. Plastic Base and studs
29. Screws/Nuts/Bolts

#### **TOOLS REQUIRED :**

1. Soldering Iron
2. Glue Gun
3. Dremel
4. Hobby Knife
5. Wire Cutter
6. Wire Stripper
7. Screw Driver
8. Ruler and pencil

## Step 2: Basics on MPPT charge controller





A solar panel will generate different voltages depending on the different parameters like :

- 1.The amount of sun light
- 2.The connected load
- 3.The temperature of the solar panel.

Throughout the day, as the weather changes, the voltage produced by the solar panel will be constantly varying. Now, for any given voltage, the solar panel will also produce a current (Amps). The amount of Amps that are produced for any given voltage is determined by a graph called an **IV curve**, which can be found on any solar panel's specification sheet and typically looks like the figure-1 shown above.

In the above figure-2, the blue line shows a solar panel voltage of 30V corresponding to a Current of about 6.2A. The green line shows a Voltage of 35V corresponds to a current of 5A.

We know that  $\text{Power} = V \times I$

In the picture shown above as you move along the red curve above you will find one point where the Voltage multiplied by its corresponding Current is higher than anywhere else on the curve. This is called the solar panel's Maximum Power Point(**MPP**).

### What Is MPPT ?

MPPT stands for Maximum Power Point Tracking. MPPT charge controllers used for extracting maximum available power from PV module under certain conditions. In order to understand this in details we first need to look at the power curve characteristics of a solar panel. Look at the image shown above. I have downloaded this images from web to explain the MPPT. Till now we have seen that the maximum power point (MPP) of a solar panel lies at the knee of the current and voltage curve.

A 12V solar panel is not really a 12V panel at all. Its really a somewhere in between 12V and 21V panel depending on what load is connected to it and how bright the sunlight is. The panel has an internal resistance which changes dynamically with differing irradiance levels. Solar panels will only deliver their rated power at one specific voltage and load, and this voltage and load moves around as the sunlight intensity changes.

For example take a solar panel rated at 100 watts, 18V at 5.55 amps.

The 18 V at 5.5 amps means that the Solar panel wants to see a load of  $18/5.5 = 3.24$  ohms.

With any other load the panel will deliver less than 100 watts. So if a static load is connected directly to a panel and its resistance is higher or lower than the panels internal resistance at

MPP, then the power drawn from the panel will be less than the maximum available.

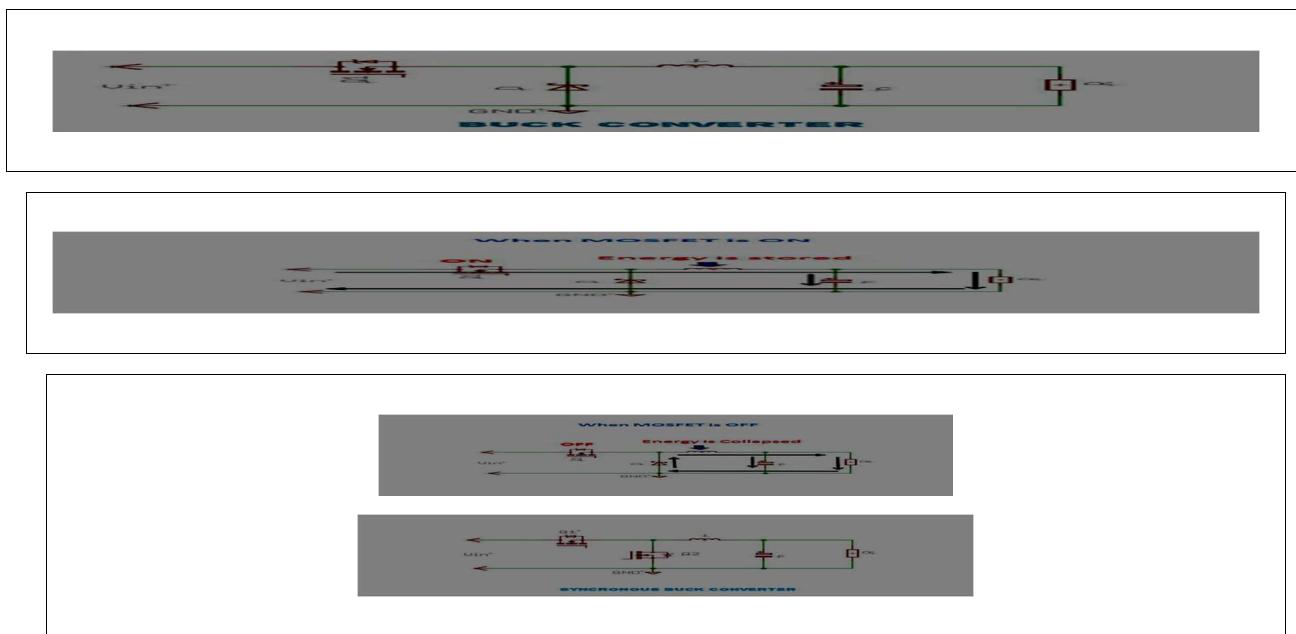
Taking a simple example say we connected the above 100W panel directly to a 12V lead acid battery, the panel voltage would be dragged down near to the load voltage of the battery as the batteries resistance is lower than the panels, but the current stays the same at 5.55 amps. This happens because Solar Panels behave like current sources, so the current is determined by the available sunlight.

Now the power ( $P = V \times I = 12 \times 5.55 = 66.6\text{W}$ ). So the Solar panel is now behaving like a 66 watt panel.

This equates to a loss of  $100\text{W} - 66.6\text{W} = 34\text{W}$  ( 33.4%).

This is where MPPT comes into play. MPPT circuits can be based on various switch mode power supply (SMPS) topologies, they generally have a fixed frequency but varying duty cycle. The duty cycle is controlled via an algorithm so as to track the changing MPP.

### Step 3: BUCK CONVERTER WORKING



A buck converter is basically a DC to DC converter. The main principle at work in a buck converter, is the tendency for an inductor to resist changes in current. A buck converter output voltage will always be lower or the same as the input voltage. A simplified schematic of a buck converter is shown in the above picture.

#### Working Principle :

##### When the MOSFET is ON

When the mosfets is ON, current flows through the inductor (L) in a clockwise direction into the load (R) and also charging the output capacitor (C). At this point the voltage on the cathode of the diode is positive, therefore the diode (D) is blocking any flow of current and is said to be reverse biased. The Initial current flow into the load from  $V_{in}$  is slow, as energy is stored in the inductor as it's magnetic field increases. So during the ON state of the mosfet, energy is stored in the inductor.

## **When the MOSFET is OFF**

When the mosfet is switched OFF, the voltage across the inductor is reversed. The inductors magnetic field begins to collapse, this collapse releases the stored energy allowing current to flow from the inductor into the load. The diode now has a negative voltage on the cathode so becomes forward biased. Therefore the inductors discharge current flows in a clockwise direction through the load and back through the diode. Once the inductors energy has fallen below a certain threshold, the load voltage falls and the capacitor becomes the main source of current, ensuring the load is still supplied until the next switching cycle begins. To ensure continuous conduction mode the inductor is must not be fully discharged before the mosfet is switch on again, and the cycle repeats.

## **What is Synchronous Buck Converter ?**

The synchronous design simply replaces the diode with a second mosfet, this eliminates the losses incurred by the forward voltage drop across the diode, thus making the circuit more efficient. This is slightly more complex to implement, as the second mosfet switching needs to be carefully timed with the switching of the first mosfet. It is essential to ensure that both are never on at the same time, or the current will have a direct path to ground, effectively causing a short circuit. The mosfet switching is effectively 180 degrees out of phase, with a short delay period between each transition referred to as a Dead-Band.

## **Step 4: BUCK CONVERTER DESIGN**

The first step was to design the buck converter circuit, this is determined by the output parameters of the system and it's load. It designed for a 50W solar panel, it was decided to aim for a 12 V output ( battery voltage)

When calculating a buck circuit the frequency of operation, inductor size and output capacitor size are important, as they determine the current and voltage ripple size. It is desirable to have as smaller current and voltage ripple as possible.

A general rule is the higher the frequency the smaller the size of the inductor and output capacitor, and a smaller inductor and capacitor size generally lowers the system cost. However higher PWM frequencies decrease the system efficiency due to switching losses in the mosfets, so a trade off has to be reached which meets the design constraints of the end system.

For this design a PWM frequency of 50kHz was chosen.

From the earlier discussion we have conclude that a buck converter is consist of

- 1.Inductor
- 2.Capacitor
- 3.MOSFETS

In the next few steps I will discuss how to choose these components.

## Step 5: INDUCTOR CALCULATION



Calculating the inductor value is most critical in designing a buck converter. First, assume the converter is in continuous current mode (CCM), which is usually the case. CCM implies that the inductor does not fully discharge during the switch-off time. The following equations assume an ideal switch (zero on-resistance, infinite off-resistance and zero switching time) and an ideal diode.

### Assume

We are designing for a 50W solar panel and 12V battery

Input voltage ( $V_{in}$ ) = 15V

Output Voltage ( $V_{out}$ ) = 12V

Output current ( $I_{out}$ ) =  $50W/12V = 4.16A = 4.2A$  (approx)

Switching Frequency ( $F_{sw}$ ) = 50 KHz

Duty Cycle ( $D$ ) =  $V_{out}/V_{in} = 12/15 = 0.8$  or 80%

### Calculation

$$L = (V_{in} - V_{out}) \times D \times 1/F_{sw} \times 1/dI$$

Where  $dI$  is Ripple current

For a good design typical value of ripple current is in between 30 to 40 % of load current.

Let  $dI = 35\%$  of rated current

$$dI = 35\% \text{ of } 4.2 = 0.35 \times 4.2 = 1.47A$$

$$\text{So } L = (15.0 - 12.0) \times 0.8 \times (1/50k) \times (1/1.47) = 32.65\mu H = 33\mu H \text{ (approx)}$$

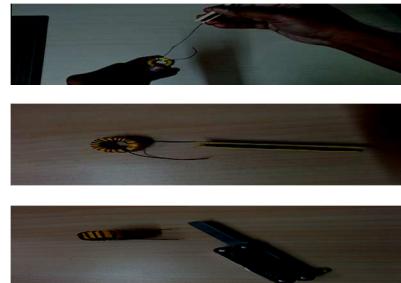
$$\text{Inductor peak current} = I_{out} + dI/2 = 4.2 + (1.47/2) = 4.935A = 5A \text{ (approx)}$$

So we have to buy or make a toroid inductor of 33 $\mu$ H and 5A.

You can also use a buck converter design [calculator](#)

So 33 $\mu$ H is enough for our design.

## Step 6: HOW TO WIND A TOROIDAL INDUCTOR



### Show All 7 Items

I have collected a bunch of toroidal cores from old computer power supply. So I thought to made the inductor at my home. Though it took a lot of time to make, but I learned a lot and enjoyed during making. These are few tricks what I learned during the making, so that you can make it easily.

#### How to Wind the wire :

Winding by hand is very painful for skin as well as you can't make the winding so tight. So I made a simple tool from popsicle stick for winding the toroidal core. This simple tool is very handy and you can make perfect and tight winding. Before making the inductor you have to know the core specification and number of turns.

The important parameters of toroidal core are

1. Outer diameter(OD)
2. Inner diameter(ID)
3. Height (H)
4. Al value

As I did not know the part number, I used an indirect method to identify it. First I measure the OD and ID of the unknown core by using my vernier caliper, it was around

OD= 23.9mm (.94") , ID= 14.2mm(.56") , H= 7.9mm( .31") and yellow white in color.

I used a toroid core chart (page-8) to identify the unknown core. I have attached this toroid size chart in the bellow. It contains a lot of information for the inductor design. The PDF version is attached bellow.

## Finding the part number :

I searched the **Physical dimension** table from the chart. From the table it was found that the core is **T94**

## Finding the mix number :

The color of the core is indication for mix number. As my core is yellow/white in color, it is confirmed that the mix number is 26.

So the unknown core is T94-26

### Finding Al value :

From the Al value table for a T94-26 core it is 590 in uH/100 turns.

After selecting the core now time to find out the number of turns required to obtain the desired inductance.

**Number of turn (N) =  $100 \times \sqrt{\text{desired inductance in uH} / \text{Al in uH per 100 turns}}$**

$$\Rightarrow N = 100 \sqrt{33/590} = 23.65 = \text{approximately 24 turns}$$

You can also use [this](#) online calculator for finding the number of turns. Only you have to know the part number and mix number.

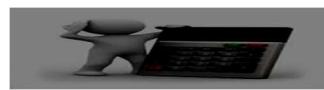
Then I wind a 20 AWG copper wire (24 turns) around the the toroid core. At the both end of the winding leave some extra wire for connection lead. After this remove the enamel insulation from the lead. I used my leatherman file for removing the insulation. See the above picture for better understanding.

**Note :** Making a good inductor is not so simple.I am still in learning stage.If you are not so confident I will recommend to buy a ready made inductor.



toroid core size.pdf4 MB

## Step 7: CAPACITOR CALCULATION



Output capacitance is required to minimize the voltage overshoot and ripple present at the output of a buck converter. Large overshoots are

caused by insufficient output capacitance, and large voltage ripple is caused by insufficient capacitance as well as a high equivalent-series resistance (ESR) in the output capacitor. Thus, to meet the ripple specification for a buck converter circuit, you must include an output capacitor with ample capacitance and low ESR.

### Calculation :

The out put capacitor ( Cout)=  $dI / (8 \times F_{sw} \times dV)$

Where dV is ripple voltage

Let voltage ripple( dV ) = 20mV

Cout=  $1.47 / (8 \times 50000 \times 0.02) = 183.75 \mu\text{F}$

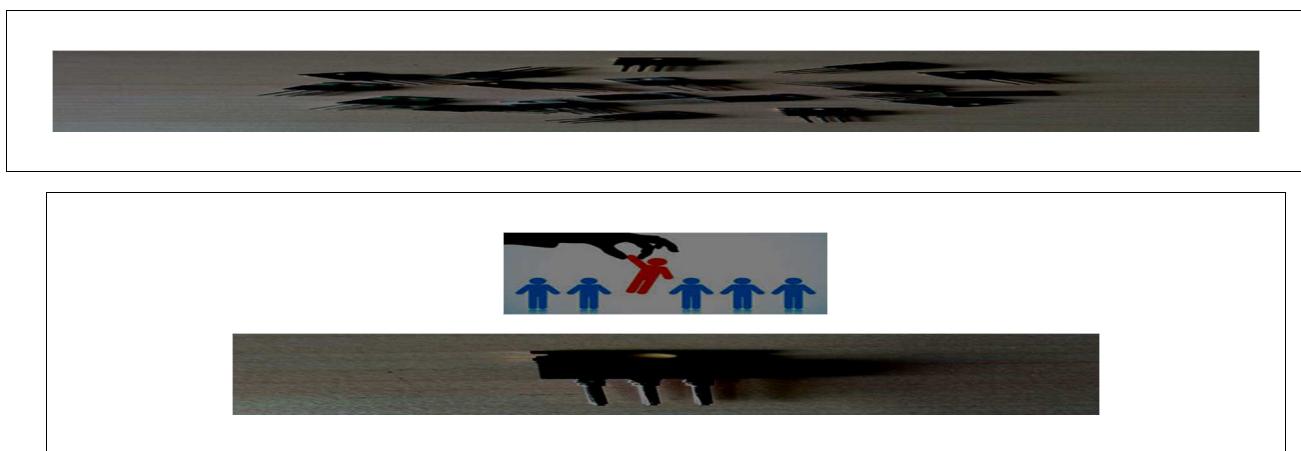
By taking some margin, I select 220uF electrolytic capacitor.

The equations used for calculation of inductor and capacitor is taken from a article [LC Selection Guide for the DC-DC Synchronous Buck Converter](#)



LC selection.PDF 889 KB

## Step 8: MOSFET SELECTION



The vital component of a buck converter is MOSFET. Choosing a right MOSFET from the variety of it available in the market is quite challenging task.

These are few basic parameters for selecting right MOSFET.

**1. Voltage Rating :** Vds of MOSFET should be greater than 20% or more than the rated voltage.

**2. Current Rating:** Ids of MOSFET should be greater than 20% or more than the rated current.

**3.ON Resistance (Rds on) :** Select a MOSFET with low ON Resistance (Ron)

**4.Conduction Loss :** It depends on Rds(ON) and duty cycle.Keep the conduction loss minimum.

**5.Switching Loss:** Switching loss occurs during the transition phase. It depends on switching frequency,voltage ,current etc.Try to keep it minimum.

These are few links where you can get more information on selecting the right MOSFET.

## 1.MOSFET selection for Buck Converter

### 2.A simple guide to selecting power MOSFETs

In our design the maximum voltage is solar panel open circuit voltage(Voc) which is nearly 21 to 25V and maximum load current is 5A.

I have chosen IRFZ44N MOSFET. The Vds and Ids value have enough margin as well as it has low Rds(On) value.

You can check the other parameters of IRFZ44N from the [data sheet](#)



IRFZ44N DATA SHEET.pdf 104 KB

## Step 9: MOSFET DRIVER



Why we need a gate driver ?

A Mosfet driver allows a low current digital output signal from a Microcontroller to drive the gate of a Mosfet. A 5 volt digital signal can switch a high voltage mosfet using the driver. A MOSFET has a gate capacitance that you need to charge so that the MOSFET can turn on and discharge it to switch off, the more current you can provide to the gate the faster you switching on/off the mosfet, that is why you use a driver.

For more details you can read about [MOSFET Basics](#)

For this design I am using a IR2104 Half Bridge driver. The IC takes the incoming PWM signal from the micro controller, and then drives two outputs for a High and a Low Side MOSFET.

### How to use it ?

From the data sheet I have taken the image shown above.

#### Input :

First we have to provide power to the gate driver. It is give on Vcc (pin-1) and its value is in between 10-20V as per data sheet.

The high frequency PWM signal from Arduino goes to IN (pin-2). The shut down control signal from the Arduino is connected on SD ( pin 3).

#### Output :

The 2 output PWM signals are generated from HI and LO pin. This gives the user the

opportunity to fine tune the dead-band switching of the MOSFETs.

### Charge Pump Circuit :

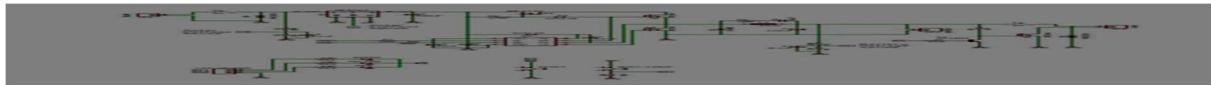
The capacitor connected between VB and VS along with the diode form the charge pump. This circuit doubles the input voltage so the high switch can be driven on. However this bootstrap circuit only works when the MOSFETs are switching.

The data sheet of IR2104 is attached [here](#)



IR2104 data sheet.pdf 137 KB

## Step 10: SCHEMATIC AND WORKING



The input power connector to the solar panels is the screw terminal JP1 and JP2 is the output screw terminal connector to the battery. The third connector JP3 is connection for the load.

F1 and F2 are the 5A safety fuses.

The buck converter is made up of the synchronous MOSFET switches Q2 and Q3 and the energy storage devices inductor L1 and capacitors C1 and C2. The inductor smooths the switching current and along with C2 it smooths the output voltage. Capacitor C8 and R6 are a snubber network, used to cut down on the ringing of the inductor voltage generated by the switching current in the inductor.

The third MOSFET Q1 is added to allow the system to block the battery power from flowing back into the solar panels at night. In my earlier charge controller, this is done by a diode in the power path. As all diodes have a voltage drop a MOSFET is much more efficient. Q1 turns on when Q2 is on through D1. R1 drains the voltage off the gate of Q1 so it turns off when Q2 turns off.

The diode D3 (UF4007) is an ultra fast diode that will start conducting current before Q3 turns on. It is supposed to make the converter more efficient.

The IC IR2104 is a half bridge MOSFET gate driver. It drives the high and the low side MOSFETs using the PWM signal from the arduino (Pin -D9). The IR2104 can also be shut down with the control signal (low on pin -D8) from the Arduino on pin 3. D2 and C7 are part of the bootstrap circuit that generates the high side gate drive voltage for Q1 and Q2. The software keeps track of the PWM duty cycle and never allows 100% or always on. It caps the PWM duty cycle at 99.9% to keep the charge pump working.

There are two voltage divider circuits (R1,R2 and R3,R4) to measure the solar panel and battery voltages. The output from the dividers are feeds the voltage signal to Analog pin-0 and Analog pin-2. The ceramic capacitors C3 and C4 are used to remove high frequency spikes.

The mosfet Q4 is used to control the load. The driver for this mosfet is consists of a transistor and resistors R9 ,R10.

The diode D4 and D5 are TVS diodes used for over voltage protection from solar panel and load side.

The current sensor ACS712 sense the current from the solar panel and feeds to the Arduino analog pin-1.

The 3 LEDs are connected to the digital pins of the microcontroller and serve as an output interface to display the charging state.

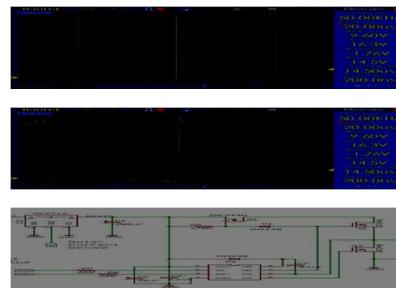
Reset switch is helpful if the code gets stuck.

The back light switch is to control the back light of LCD display.



[mppt controller\\_schema.pdf](#) 31 KB

## Step 11: Test the gate driver and MOSFETs Switching



Hey I think I have talked a lot on the theory. So lets do some practical.

As I have told earlier the heart of the MPPT charge controller is Buck Converter. As per me if your buck converter circuit work perfectly. You can do the rest thing easily. So first lets test the Mosfets switching and the driver.

Before soldering ,I request to do it on a bread board.I have blown lot of MOSFETs during my testing. So be careful during the connection.

Connect the everything as per schematic given above. Now you can omit the TVS diode, current sensor and voltage divider.

After connecting everything test the resistance between the input rail. It should be several KOhm. If you get resistance below 1K then recheck the circuit connection.

Upload the test sketch to the Arduino. The code in the form of text file is attached below.

Then connect the scope in between the source of Q1 and GND.

The result should be a PWM with frequency 50KHz.

The waveform obtained during my testing are shown above.

If everything goes right then proceed to complete the bulk converter circuit.( i.e adding inductor and capacitor)



test\_code.txt 843 bytes

## Step 12: Test the Buck Converter



[Show All 8 Items](#)

In the previous steps we have calculated the inductor and capacitor rating. Now it is time to use and test it.

Add the 33uH inductor and 100uF input and 220uF output electrolytic capacitor as per schematic. You can also use 0.1uF ceramic capacitors parallel with input and output capacitors. It will give better results. But it is not mandatory.

Then make the snubber circuit by using a 0.1uF ceramic capacitor and 200ohm resistor.

Again check the resistance in between the input rail. It should be of the order of K ohm.

Now give power to the input rail and Arduino.

Connect the probe of your scope in between the output capacitor.

The result is shown above .The out put should be a steady DC.

$$V_{out} = \text{Duty Cycle} \times V_{in}$$

For example if i give 50% duty cycle to a 12 input supply, the output should be 6V in the scope.

After confirmed that everything working fine,now we can add the blocking mosfet Q1.It is used to block reverse power from battery to the solar panel during night.

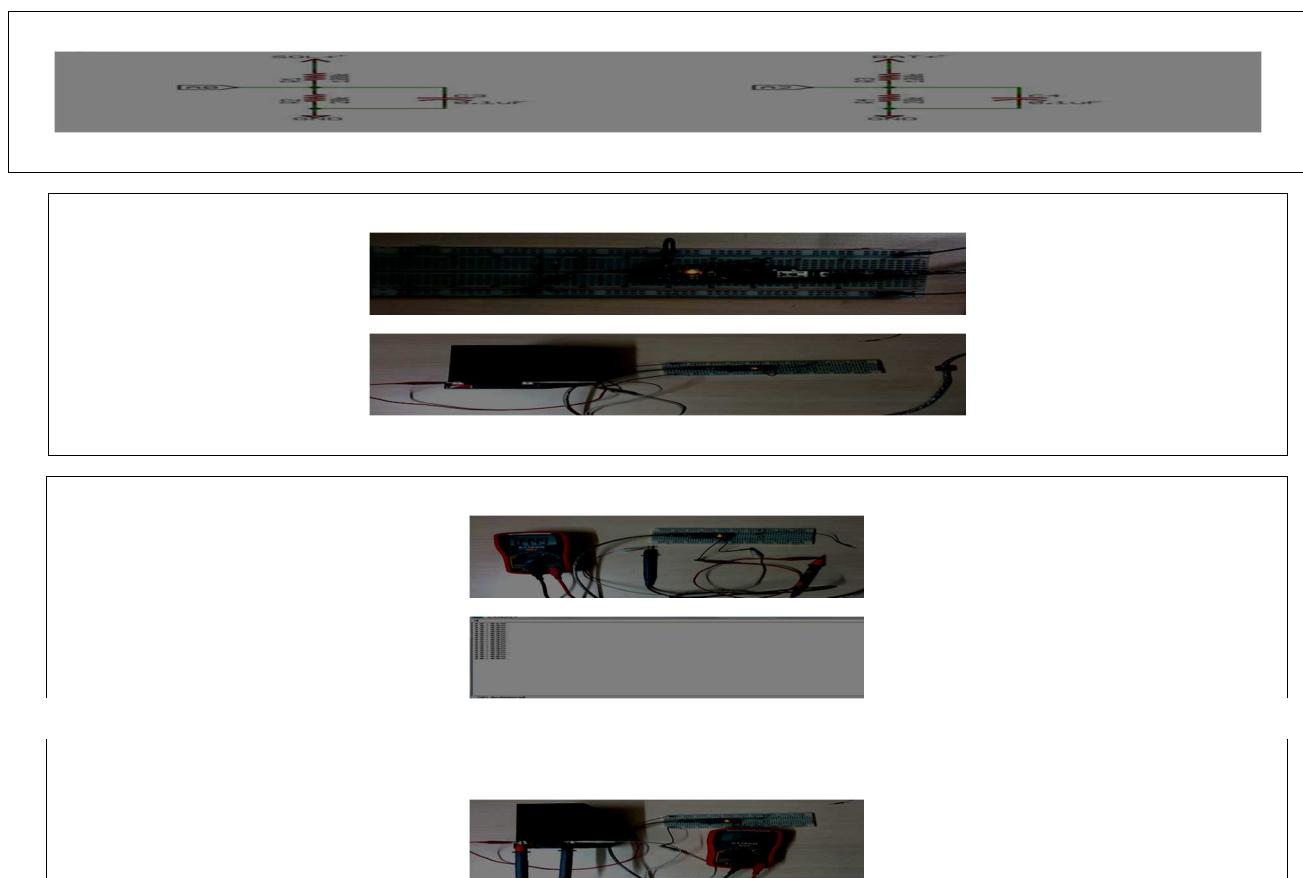
Add the third mosfet Q3 as per schematic.Then place the 470k resistance and diode IN4148.

Again check the output it should be same.

At last put the scope in between the gate of Q1 and Gnd.

Do you know ? you have done the most critical part of this project.

## Step 13: VOLTAGE MEASUREMENT



## Show All 7 Items

### Voltage Measurement :

As you may well know, Arduino's analog inputs can be used to measure DC voltage between 0 and 5V (when using the standard 5V analog reference voltage) and this range can be increased by using two resistors to create a voltage divider. The voltage divider decreases the voltage being measured to within the range of the Arduino analog inputs. We can use this to measure the solar panel and battery voltages.

For a voltage divider circuit

$$V_{out} = R_2/(R_1+R_2) \times V_{in}$$

$$V_{in} = (R_1+R_2)/R_2 \times V_{out}$$

The analogRead() function reads the voltage and converts it to a number between 0 and 1023

### Example code :

```
// read the input on analog pin 0 ( You can use any pin from A0 to A5)  
int Value = analogRead(A0);  
Serial.println(value);
```

The bove code gives an ADC value in between 0 to 1023

### Calibration :

We're going to read output value with one of the analog inputs of Arduino and its analogRead() function. That function outputs a value between 0 (0V in input) and 1023 (5V input) that is 0,0049V for each increment (As  $5/1024 = 0.0049V$ )

$$V_{in} = V_{out} * (R_1+R_2)/R_2 ; R_1=100k \text{ and } R_2=20k$$

$$V_{in} = \text{ADC count} * \mathbf{0.0049} * (\mathbf{120/20}) \text{ Volt} // \text{Highlighted part is Scale factor}$$

**Note :** This leads us to believe that a reading of 1023 corresponds to an input voltage of exactly 5.000 volts.

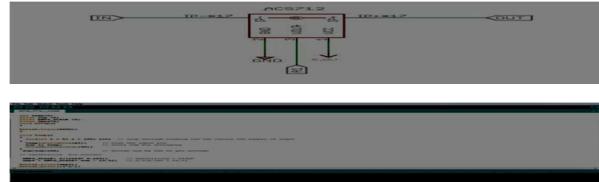
*In practical you may not get 5V always from the arduino pin 5V .So during calibration first measure the voltage between the 5v and GND pins of arduino by using a multimeter, and use  $IADC = \text{measured voltage}/1024$  instead of  $5/1024$*

Check your voltage sensor by a test code attached bellow



**Voltage\_mesurement\_code.ino** 784 bytes

## Step 14: CURRENT MEASUREMENT



For current measurement I used a Hall Effect current sensor ACS 712 (5A).

The ACS712 sensor read the current value and convert it into a relevant voltage value, The value that links the two measurements is sensitivity. You can find it on the [datasheet](#).

#### As per data sheet for a ACS 712 (5A) model :

1. Sensitivity is 185mV/A.
2. The sensor can measure positive and negative currents (range -5A...5A),
3. Power supply is 5V
4. Middle sensing voltage is 2.5V when no current.

#### Calibration:

Value =  $(5/1024) * \text{analog read value}$

// If you are not getting 5V from arduino 5V pin then, value =  $( V_{\text{measured}} / 1024 ) * \text{analog read value}$

//  $V_{\text{measured}}$  is the voltage in between Arduino pin 5V and GND. You can measure it by a multimeter.

But as per data sheets offset is 2.5V (When current zero you will get 2.5V from the sensor's output)

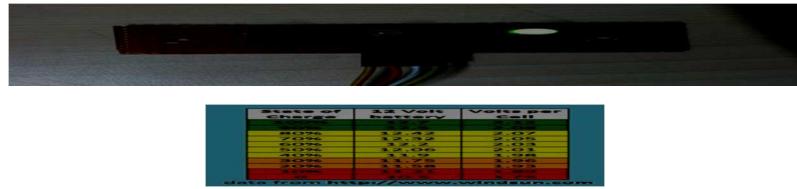
Current in amp =  $(\text{value} - 2.5) / 0.185$

Test it by a sample code for ACS712 attached bellow.

 **ACS\_712\_code.ino** 791 bytes

### Step 15: LCD Display and LED Indication





#### LCD display :

A 20X4 char LCD is used for monitoring solar panel, battery and load parameters. For simplicity a I2C LCD display is chosen. It needs only 4 wires to interface with the arduino. In my earlier design the LCD was consuming lot of power. The main cause was LCD back light. So I add a push switch to control the back light. By default the back light will be in off condition. If the user press the switch then it will on for 15 secs and again goes off.

Vcc--> 5V , GND-->GND, SDA-->A4 and SCL-->A5

Column-1 : Solar panel voltage, Current and Power

Column-2 : Battery Voltage, Charger state and SOC

column-3 : PWM duty cycle and load status

For testing the LCD download the test code attached bellow.

You download the library from [LiquidCrystal\\_I2C](#) .

#### LED indication :

Red ,Green and Yellow leds are used to indicate the battery voltage level.

Low Voltage -- > Red led

Normal Voltage --> Green Led

Fully Charged --> Yellow Led



[LCD\\_code\\_test.ino](#) 2 KB

## Step 16: HARDWARE AND SOLDERING



Before soldering you should clear about the Power and Control Signal. Do not mix up between them. Otherwise you will fry everything.

#### Power Signal :

1.Solar panel -> Fuse -> Current sensor -> Mosfets Q1,Q2 ,Q3 -> Inductor -> Battery.

2.Battery -> Fuse -> Load -> Mosfet Q4

#### Control Signals :

1.Signal from the different Sensors to Arduino

2. Signals from the Arduino to the Mosfet drivers,LED,LCD

3. Signal between the Arduino and ESP8266

I used red and black thick wires ( 0.5 to 0.75 sq mm) for power and ground connections respectively.

All the colored thin wires are for control signals.

**Tips:** Print the PDF format Schematics before soldering.Keep it in front of you during soldering for reference.

### Step 17: Drill Holes for mounting



First hold the prototype board by a vice.

Then drill 4 holes (3mm) at the 4 corners of the prototype board.

### Step 18: Add the Input and out put terminals :





First solder the three screw terminals for solar panel,battery and load connection.

The left one is for solar panel,middle one is for battery and the right one is for load connection.

## Step 19: Add the Fuse Holders



On the extreme left and right solder the two fuse holders.( One in the solar panel side and other on the load side)

Then connect the left terminal of the solar screw terminal with one leg of the fuse holder.

## Step 20: Solder the MOSFETS and Input Capacitor





Solder all the 4 MOSFETs with equally spaced on the top of the prototype board.(Leave some space to putting the heat sinks)

Then add the input 100uF capacitor.I left some space in between the fuse holder and Capacitor for installing the current sensor later.

Solder connecting wires as follows :

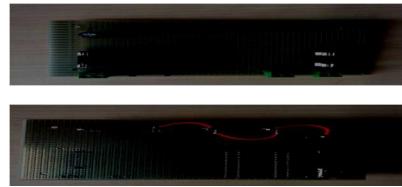
Between positive terminal of input capacitor(C) and source of mosfet Q1.

Between drains of mosfet Q1 and Q2.

Then in between source of Q2 and drain of Q3.

## Step 21: Mounting the Arduino Nano





#### Show All 7 Items

First cut two rows of female and male header pin with 15 pins in each.I used a diagonal nipper to cut the headers.

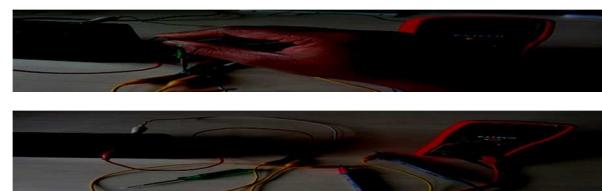
Then solder the male header pins.Be sure the distance between the two rails fits the arduino nano.

Leave two rows on each side of the female header and then solder the two male headers.

Then short the corresponding male and female pins.Though I forgot this during my soldering.

The female headers is used to mount the Arduino nano and male headers are used for external connection with the Arduino.

## Step 22: Make The Power supply



## Show All 11 Items

To run the Arduino ,different sensors,LED,LCD and the wifi module( ESP8266 ) we need power.

Except ESP8266 module all the others can be run by 5V power supply.The ES8266 module need power not more than 3.7V. It is recommended to run it on 3.3V. Though Arduino Nano have 3.3V pin but it can not provide sufficient power ( around 200mA to 300mA) to run the ESP8266 module.So we need a separate 3.3V power supply which can provide at least 300mA current.

### **5V Power Supply :**

In my previous version I used a LM7805 linear voltage regulator to step down the battery voltage to 5V for the power supply.But it produces a lot of heat during its working.So I used a high efficient buck converter in this design.

Adjust the output voltage of buck converter :

First connect the battery on the input terminal of the buck converter and adjust the potentiometer to get 5V out put.

See the above picture.

Cut 4 pcs of male header with 2pins in each.Solder the headers as per the holes given in the converter.

Place the converter on the above 4 header pin and solder on the top.Be sure the input side is toward the battery screw terminal.

Add the output capacitor(C2) near to the battery screw terminal.The positive terminal of the capacitor should be on the left.

Then connect the input of the buck converter to the battery screw terminal and output to the 5V and GND pin of the Arduino Nano.At this stage you can check it.Place the Arduino nano on the header pin and connect the 12V battery to the screw terminal.If everything is correct then Arduino power led should glow.

Finally add two rows of male header pins to the side of Arduino 5V and GND pin for external connection.

### **3.3V Power Supply :**

I am planning to use a voltage regulator AMS1117 to step down from 5V to 3.3V.

Solder the voltage regulator first, then add two 10uF capacitors. One on the input and other on the output side.

See the above schematic.

## **Step 23: Solder the Mosfet Driver circuit**



#### Show All 9 Items

First solder the 8 pins DIP socket just above the arduino header pins.

Add 10uF capacitor and and a 0.1uF capacitor in between the pin-1 and pin-4.

Solder the diode (D2) in between pin -1 and 8.The diode cathode should be connect to the pin-8.

Solder the capacitor (C7) in between pin-8 and pin-6.

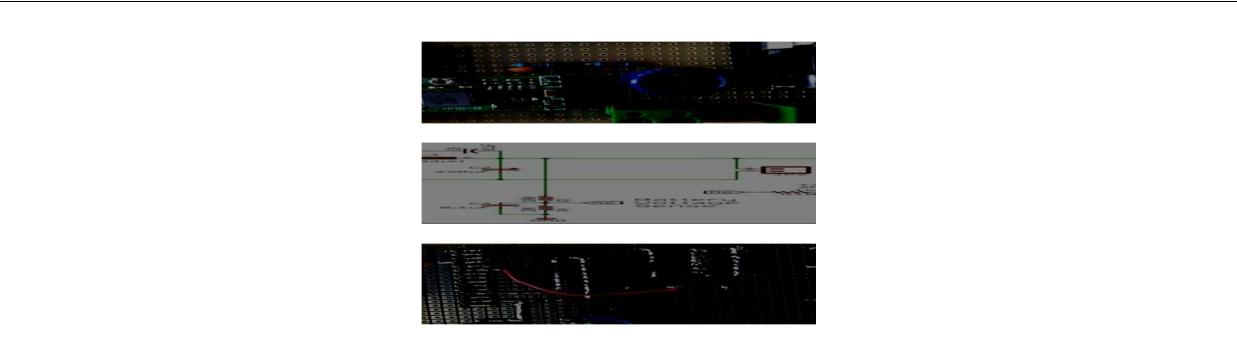
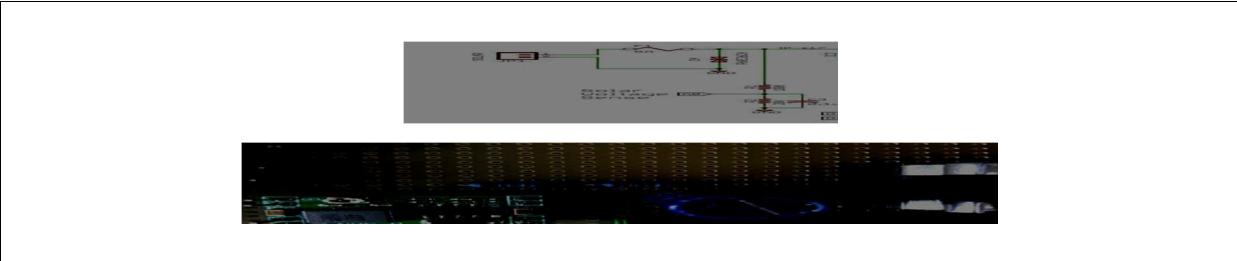
Solder two 200ohm resistors ( R7 and R8) just side to the pin-2 and pin-3.

Solder one 470K resistor (R1) near to the mosfet Q1 and a diode (D1) in between gates of mosfets Q1 and Q2.The diode cathode connects to the gate of Q1.

After this complete the circuit by soldering wires as per the schematics.

### Step 24: Solder the Voltage Sensors





#### Show All 7 Items

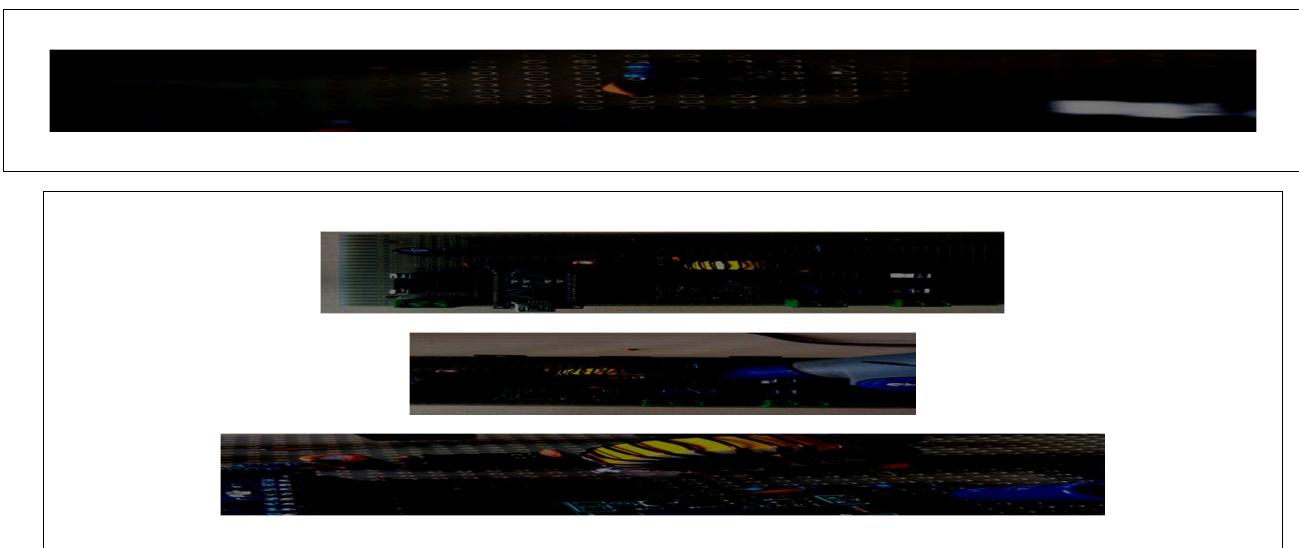
Solder solar panel voltage divider near to the fuse and battery voltage divider near to the output capacitor.

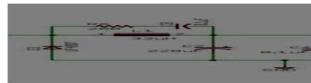
Then solder two ceramic capacitors ( C3 and C4) across the 20k resistors.

Then solder a wire between middle point of the solar panel side voltage divider and arduino pin A0.

Finally solder a wire between middle point of the battery side voltage divider and arduino pin A2.

### Step 25: Solder The Inductor and Snubber Circuit





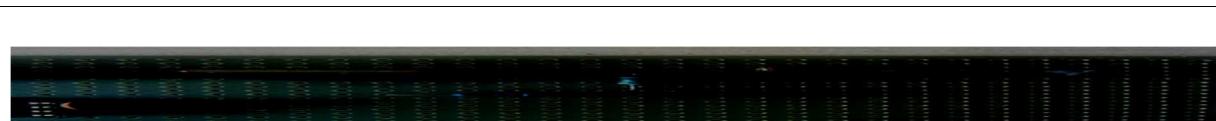
First solder the resistor (R6) and capacitor (C8) in series just above the output capacitor( C2).

Then solder the inductor parallel to it.

Inductor is the heavier component in the entire circuit.To sit it firmly, apply glue at the base.

Then solder the ultra fast diode (D3) .

## Step 26: Solder the Load Mosfet (Q4) Driver

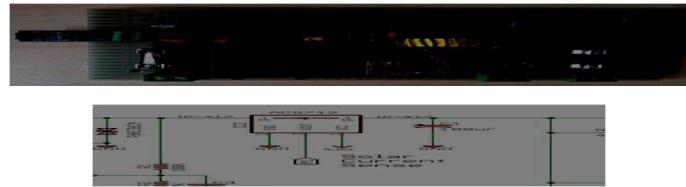


Solder the 2N2222 transistor near the gate of the mosfet (Q4).

Then add a 10k resistor (R9) near to the collector and a 1k resistor( R10) near to the base.

Then connect the points as per schematic.

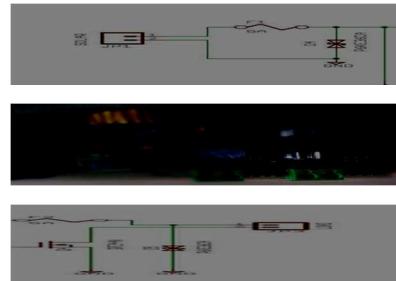
## Step 27: Adding The Current Sensor



Solder two thick wire in between the solar panel side fuse and capacitor (C1).

Then screw the wire in to the ACS712 screw terminal.

## Step 28: Solder the TVS diodes

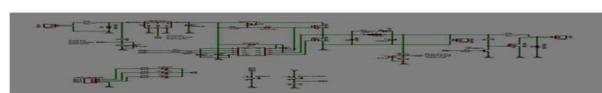
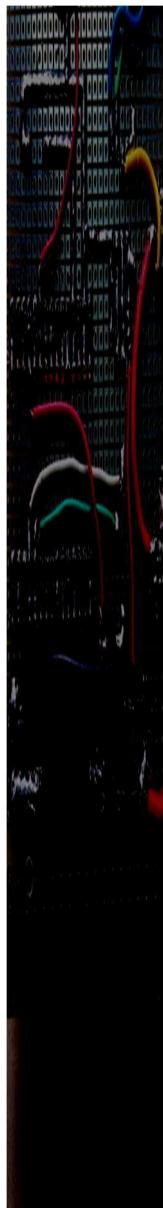


I do not have spare TVS diode. So I solder it later. You can solder it earlier also.

One TVS diodes, D4 near the connector JP1 and D5 near the connector JP3.

**Note :** I am using bidirectional TVS diode. So no polarity mark is there.

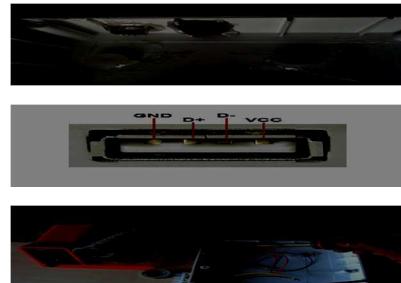
## Step 29: Connect the GND



After soldering all the components, connect all the grounds (GND) shown in the schematic.

I am using thick black wires.

## Step 30: Make the USB Charging Circuit



### Show All 9 Items

The buck converter used for power supply can deliver maximum current 3A. So the power supply have sufficient margin for charging the USB gadgets.

#### Make the Circuit :

Solder the male JST connector near to the buck converter and connect two pins with positive ( 5V )and negative

( GND ) out of the converter.See the picture.

Insert the USB port and switch in to the slots made earlier.Then apply hot glue surround them.

Solder the red wire (+ ve ) of the JST connector to one terminal of the switch.Then solder a small red wire between another terminal of switch and USB Vcc terminal.Finally solder the black wire (-ve ) of the JST connector to the USB GND.

For USB pin out see the above picture.

You can make this step earlier also.

## Step 31: Make the Wifi Module Circuit



First cut 2 female header with 4pins in each.

The solder it side by side near the load side fuse holder.

Complete the circuit as per schematic.

Be careful about when you solder this module. Voltage more than 3.7 V kill this module as it operates at 3.3 V .

Even the serial lines should not exceed this voltage.I am planning to use a 3.3 V regulator ( AMS1117 ) to power this module. A voltage divider circuit is used to drop the arduino Tx ( 5V ) to ESP8266 3.3 V ( RX ).

### Setting up the ESP8266 :

The first thing you want to do with ESP8266 is to establish communication.You can see this [example project](#) for setting up the ESP8266.Then connect it to your WiFi router.

Hey now you are ready to upload your data to the web.

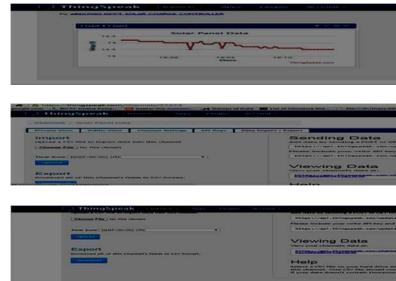
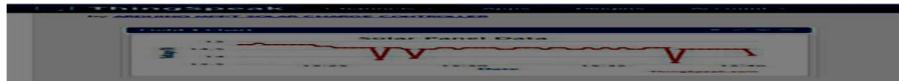
**Note:** I have not completed this part.Once completed,I will post the details along with code.

You can see the following projects to get some idea to use ESP8266 for data uploading to web.

<http://www.instructables.com/id/ESP8266-Wifi-Tempe...>

<http://www.element14.com/community/groups/internet...>

## Step 32: WiFi Data logging and Scientific Exploration



As the solar panel are installed at remote location, monitoring systems parameter is vital for us. This gives me the idea to add the data logging feature to my controller.

The WiFi module( ESP8266 ) automatically uploads live power generation, voltage, Current data to the Web( <https://thingspeak.com/> ). Then the web application graph and tabulate data in live. You can download the feeds from the website in the form of a Xcel sheet. Then explore these data for further analysis.

Go to Data Import/Export and then click on Download. See the above pics.

If you are app developer, then develop a apps for Android, iPhone and Windows Mobile to see these useful data. If you make please share me. I am not a developer.

### Scientific Exploration of Data :

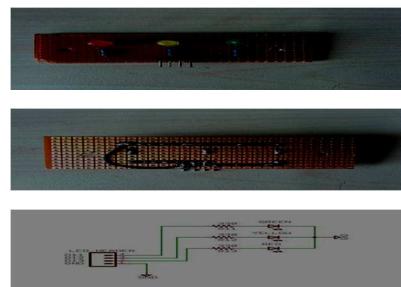
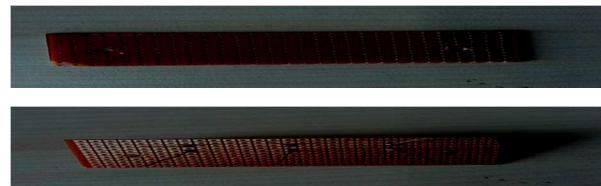
Science can be defined as the practice of observing the natural world, and trying to make objective sense of it by uncovering facts or cause-and-effect relationships.

By data logging, you can collect solar photo voltaic (PV) data that otherwise would be invisible to your senses. These are fun, interesting, and practical things to do- but doing them by themselves is not doing science. Suppose you'd like to learn more from what you uncover. Maybe you'd like to measure PV system parameters and atmospheric conditions ( temperature and solar irradiance ) over days or weeks, and then search similar solar installations within a geographical area. Compare daily data with another system, past date or postcode in different parts of your neighborhood, county, or state.



feeds.csv 21 KB

## Step 33: Make The LED Panel



Take a small size rectangular prototype board and drill holes at both end for mounting on the enclosure.

Solder the Leds with equally spaced.

Then solder the 330 ohm resistors (R11,R12 and R13) and 4pin male headers.

Finally complete the circuit as per schematics.

## Step 34: Make the Back light and Reset Switch





Take 5 female -female jumper wires and cut one side headers in all.

Insert heat shrink tube in all jumper wires.

#### **Reset Switch :**

Solder two jumper wires directly to the two pin of the push switch.

#### **Back Light Switch :**

Solder two jumper wires to the two pins of the switch.

Solder a 10k resistor to any one pin of the switch.

Then solder a jumper wire to the other end of the resistor.

Finally cover the joints with heat shrink tube and apply hot air.

### **Step 35: Prepare the Enclosure**



[Show All 12 Items](#)

I used a 6" x 8" plastic enclosure.

Mark the LCD,USB and Switch sizes .Then cut out the rectangular portion by using a dremel. Finally finish the edges by a hobby knife.

Then mark the mounting holes position for LCD,LED panel,Switches and External screw terminal by a pencil.

Drill holes at all the marked position.

**Note :** The holes size for LED is 5mm ,switches are 7mm and all other are 3mm.

## Step 36: Make the external connection Terminal



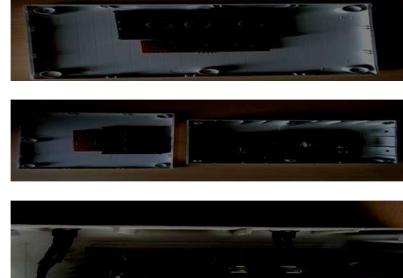
The external connector is used for outside access of all the 3 screw terminals in the controller board.

Mark the hole positions for mounting and 6 wires.

Then screw the wires in all the terminals.Use different color to distinguish between positive and negative terminal.

## Step 37: Mount Everything





To mount the controller board I used 4 plastic bases.Screw the main board over the base.

Mount the LCD and Led panel by screw and bolts.

Then mount the two switches.

## Step 38: Connect all the panel and switches



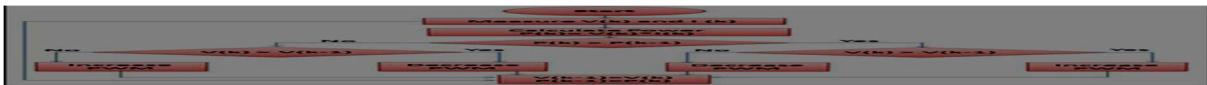
After mounting everything connect the panels,switches and external connector.

Use female-female jumper wires for connecting the panels.

Refer schematics for connection.

Finally box up the enclosure.

## Step 39: Software and Algorithm



The Maximum Power Tracker uses an iterative approach to finding this constantly changing MPP. This iterative method is called Perterb and Observe or hill climbing algorithm. To achieve MPPT, the controller adjusts the voltage by a small amount from the solar panel and measures power, if the power increases, further adjustments in the direction are tried until power no longer increases.

The voltage to the solar panel is increased initially, if the output power increase, the voltage is continually increased until the output power starts decreasing. Once the output power starts decreasing, the voltage to the solar panel decreased until maximum power is reached. This process is continued until the MPPT is attained. This result is an oscillation of the output power around the MPP.

The full software is attached bellow.



[MPPT\\_Code.ino](#) 23 KB

## Step 40: Conclusion

I have tried my best to make this instructable. Till now I am learning more on MPPT. So if I have done any mistakes please forgive me and raise a comments. I will rectify it as soon as possible.

I love getting feedback on my projects! The earlier version charge controllers has received a ton of feedback, and many users have posted pictures of their build. If you follow this Instructable and make your own controller, please share pictures and videos.

At last, I would like to give very special thanks to **timnolan**. As I have learned and used several things from his design.

For more updates and new projects subscribe me.

Thank you so much for reading my instructable.