Solar Nano grids using PV panels and LFP Battery bank

Introduction

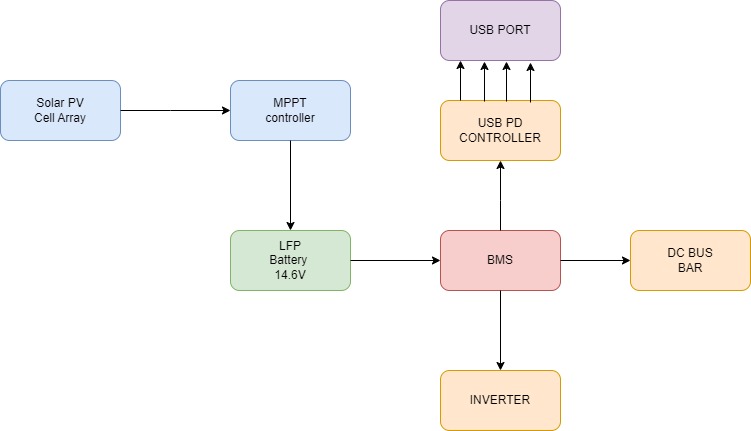
Solar nano grids are small, self-contained energy systems designed to generate, store, and distribute solar power to local areas, typically on a neighborhood or community scale. These grids harness solar energy through photovoltaic panels and store it in battery systems, making electricity available even in remote or off-grid locations. Unlike traditional centralized power grids, solar nano grids are decentralized and modular, offering a flexible solution to areas with unreliable or no grid access.

Their small scale makes them ideal for rural electrification, supporting small businesses, households, schools, and medical facilities, where reliable power can significantly impact quality of life and economic growth. In addition to being sustainable, solar nano grids are often cost-effective and environmentally friendly, as they reduce reliance on fossil fuels and minimize transmission losses. Moreover, they are resilient and can continue to operate independently during large-scale grid failures, making them suitable for disaster-prone regions or areas where climate adaptation is a priority. As such, solar nano grids play a critical role in advancing renewable energy access and supporting the global transition toward sustainable energy systems.

In this project a miniature solar nano grid is created with PV cell array and a LFP battery bank system which is ideal for domestic application.

**Architecture**

The Architecture of the Solar nano grids have two main potential components they are Solar PV cell array and a battery bank. The solar PV cells are connected in such a way that 20wpp is achieved at full sunlight. A maximum power point tracker is used to extract the maximum energy out of solar PV cell even at low intensity. The Lithium ferro phosphate battery with 3.2V nominal voltage is connected in series and parallel combination to achieve 14.6V which is a standard for normal Low powered inverters. The loads are connected to LFP battery bus. The detailed architecture is shown in the below figure.



High level architecture

The above block diagram shows the high level architecture for the solar nano grid concept. The important key clocks are explained below.

1. Solar PV cells.
2. MPPT controller.
3. LFP Battery bank.
4. Battery management system.
5. USB PD controller
6. DC to AC Inverter

Each of the above blocks are explained below.

1. **Solar PV array.**

Solar photovoltaic (PV) cells are devices that convert sunlight directly into electricity using the photovoltaic effect. They are made primarily from semiconductor materials, typically silicon, which have unique properties that allow them to generate an electric current when exposed to sunlight.There are four types of solar cells available in the market they are

· **Monocrystalline** Silicon Cells: Made from a single, continuous crystal structure, these cells have high efficiency and durability, though they are generally more expensive to produce.

· **Polycrystalline** Silicon Cells: Composed of multiple crystal structures, these are less efficient than monocrystalline cells but more affordable and widely used.

**· Thin-Film Solar Cells:** Made by depositing layers of photovoltaic material (such as cadmium telluride or amorphous silicon) onto a substrate, these cells are lightweight, flexible, and can be used in applications where conventional silicon panels aren’t practical.

· **Perovskite and Organic Cells:** Newer technologies using organic compounds or perovskite structures are emerging, promising potentially lower production costs and greater versatility, though they are still in the development stages.

The solar cells are connected in series and parallel as shown in the below figure.The nominal voltage of the solar cell is mentioned from the following image.

In our project we are using 1000 W/m^2 cells so the maximum power point for these cells are 18V output at peak time. Maximum current flow from the PV cells are 1.04A.

1. MPPT Controller.

A Maximum Power Point Tracking (MPPT) controller is an electronic device used in solar power systems to optimize the power output from solar panels. MPPT controllers maximize the efficiency of energy capture and transfer from solar panels to batteries or the grid by continuously adjusting to find the ideal operating point. This process allows the system to extract the maximum power available from the solar panels under varying conditions like changes in sunlight, shading, temperature, or panel voltage.

Solar panels are nonlinear power sources, meaning their voltage and current characteristics change constantly with varying environmental conditions. Without an MPPT controller, a solar panel may not consistently operate at its optimal power point, leading to power losses. The benefits of using an MPPT controller include:

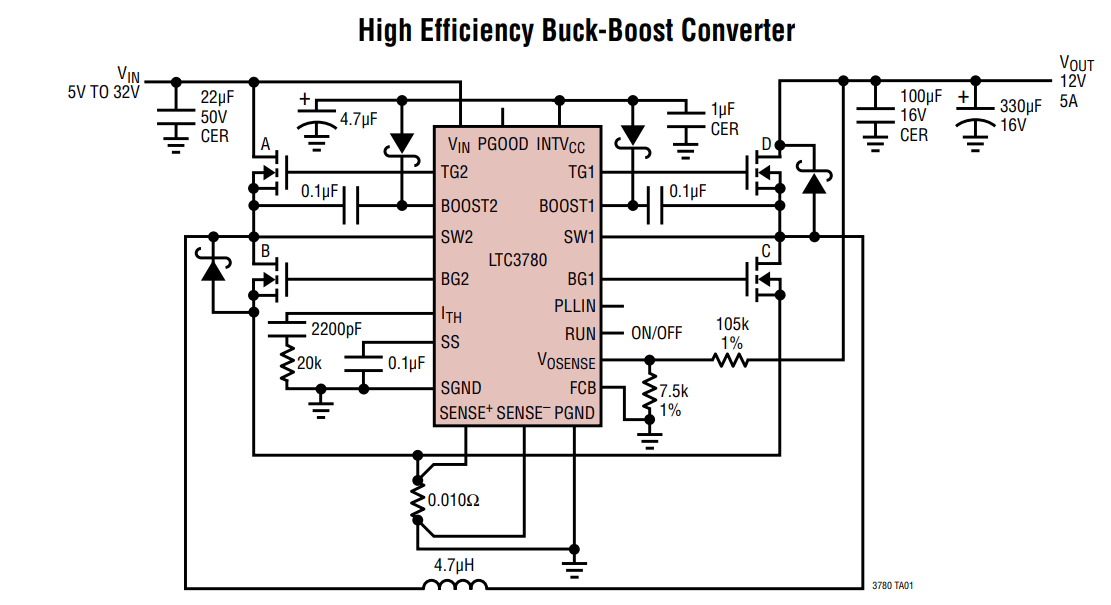
· **Increased Efficiency**: MPPT controllers can increase energy capture by 20-30% compared to conventional charge controllers. This is particularly beneficial in cloudy weather, winter conditions, or when panels operate at high voltages.

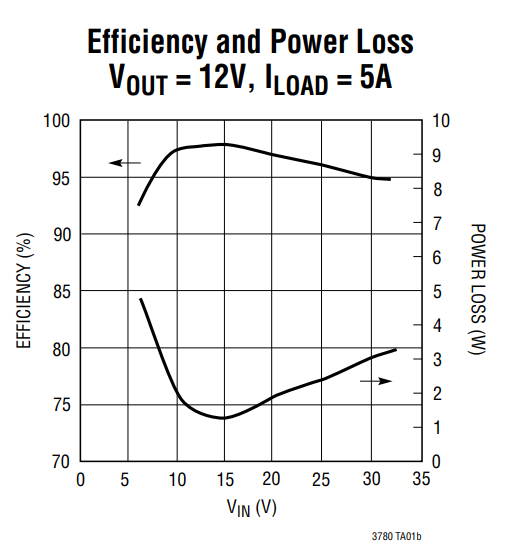
· **Effective Battery Charging**: MPPT controllers optimize charging for batteries, especially in systems with a large difference between panel voltage and battery voltage. This is particularly important for systems with batteries that need precise charging conditions to extend lifespan and prevent damage.

· **Compatibility with High-Voltage Panels**: Some solar panels produce higher voltages than battery banks or load requirements. MPPT controllers step down the voltage while increasing the current to efficiently match the load requirements, allowing the use of high-voltage solar arrays without wasting energy.

· **Better Performance in Variable Conditions**: An MPPT controller dynamically adjusts to changes in sunlight, temperature, and other factors to keep the solar panel operating at the MPP, thereby improving overall system performance and reliability.

With the above reference a buck boost controller is designed to provide proper voltage from the solar cell.



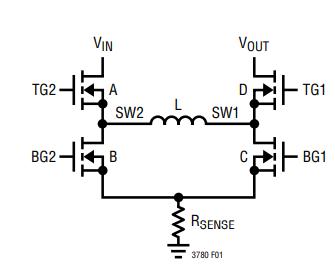


Operation :

The LTC3780 is a current mode controller that provides an output voltage above, equal to or below the input voltage. The LTC proprietary topology and control architecture employs a current-sensing resistor in buck or boost modes. The sensed inductor current is controlled by the voltage on the ITH pin, which is the output of the amplifier EA. The VOSENSE pin receives the voltage feedback signal, which is compared to the internal reference voltage by the EA.

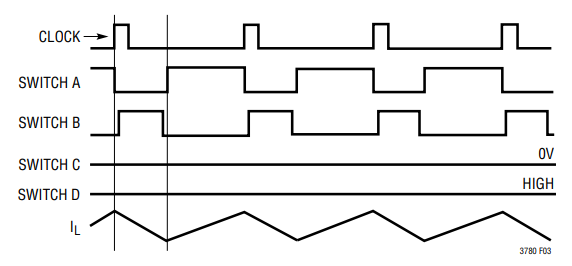
The top MOSFET drivers are biased from floating booststrap capacitors CA and CB (Figure 11), which are normally recharged through an external diode when the top MOSFET is turned off. Schottky diodes across the synchronous switch D and synchronous switch B are not required, but provide a lower drop during the dead time. The addition of the Schottky diodes will typically improve peak efficiency by 1% to 2% at 400kHz.

The main control loop is shut down by pulling the RUN pin low. When the RUN pin voltage is higher than 1.5V, an internal 1.2µA current source charges soft-start capacitor CSS at the SS pin. The ITH voltage is then clamped to the SS voltage while CSS is slowly charged during start-up. This “soft-start” clamping prevents abrupt current from being drawn from the input power supply.

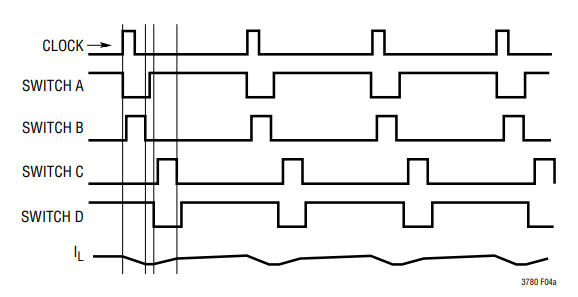


Simplified Schematics

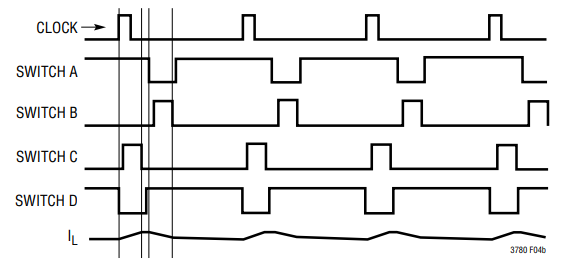
The switching characteristics for different device is shown below.



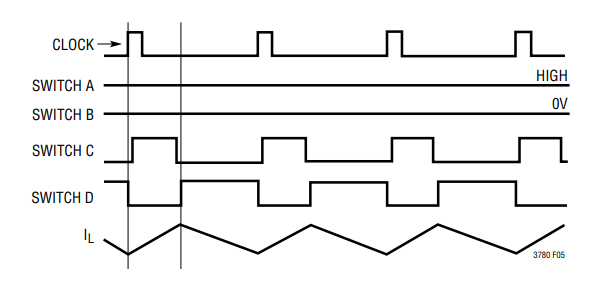
Buck Mode ( Vin > Vout)



Buck-Boost Mode (VIN ≥ VOUT)



Buck-Boost Mode (VIN ≤ VOUT)



Boost Mode (VIN < VOUT)

The duty cycle of switch C decreases until the minimum duty cycle of the converter in boost mode reaches DMIN\_BOOST, given by:

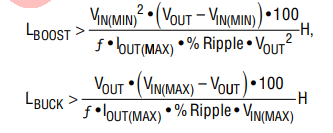
DMIN\_BOOST = DBUCK-BOOST

where DBUCK-BOOST is the duty cycle of the buck-boost switch range:

DBUCK-BOOST = (200ns • f) • 100%

F=400Khz

The operating frequency and inductor selection are interrelated in that higher operating frequencies allow the use of smaller inductor and capacitor values. The inductor value has a direct effect on ripple current. The inductor current ripple ∆IL is typically set to 20% to 40% of the maximum inductor current at boost mode VIN(MIN). For a given ripple the inductance terms in continuous mode are as follows:



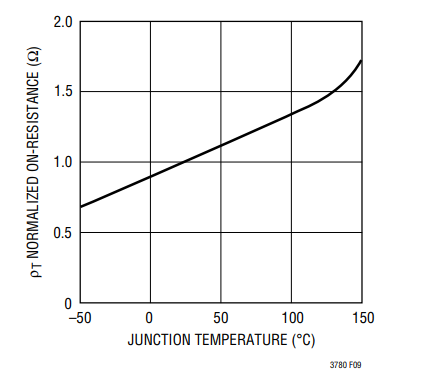
where: f is operating frequency, Hz % Ripple is allowable inductor current ripple, % VIN(MIN) is minimum input voltage, V VIN(MAX) is maximum input voltage, V VOUT is output voltage, V IOUT(MAX) is maximum output load current For high efficiency, choose an inductor with low core loss, such as ferrite and molypermalloy (from Magnetics, Inc.). Also, the inductor should have low DC resistance to reduce the I2R losses, and must be able to handle the peak inductor current without saturation. To minimize radiated noise, use a toroid, pot core or shielded bobbin inductor.approximatly 4.7uH Inductor is selected.

Power MOSFET Selection and Efficiency Considerations:

The drive voltage is set by the 6V INTVCC supply. Consequently, logic-level threshold MOSFETs must be used in LTC3780 applications. If the input voltage is expected to drop below 5V, then the sub-logic threshold MOSFETs should be considered.



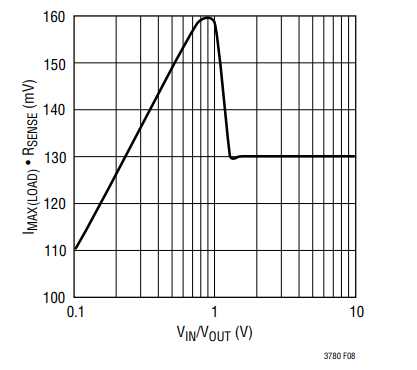
where ρT is a normalization factor (unity at 25°C) accounting for the significant variation in on-resistance with temperature, typically about 0.4%/°C as shown in Figure 9. For a maximum junction temperature of 125°C, using a value ρT = 1.5 is reasonable.



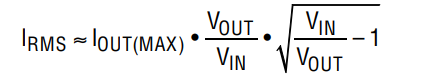
Temperature Vs normalized RDS ON is shown in above graph.

CIN and COUT Selection:

In boost mode, input current is continuous. In buck mode, input current is discontinuous. In buck mode, the selection of input capacitor CIN is driven by the need to filter the input square wave current. Use a low ESR capacitor sized.

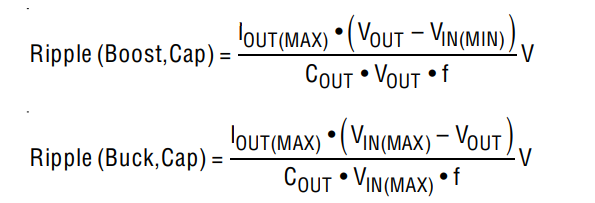


Load current vs Vin and Out is shown in above figure, To suppress the inrush current spike the capacitor must be sized as per the below equaltion.



This formula has a maximum at VIN = 2VOUT, where IRMS = IOUT(MAX)/2. This simple worst-case condition is commonly used for design because even significant deviations do not offer much relief. Note that ripple current ratings from capacitor manufacturers are often based on only 2000 hours of life which makes it advisable to derate the capacitor.

In boost mode, the discontinuous current shifts from the input to the output, so COUT must be capable of reducing the output voltage ripple. The effects of ESR (equivalent series resistance) and the bulk capacitance must be considered when choosing the right capacitor for a given output ripple voltage. The steady ripple due to charging and discharging the bulk capacitance is given by:



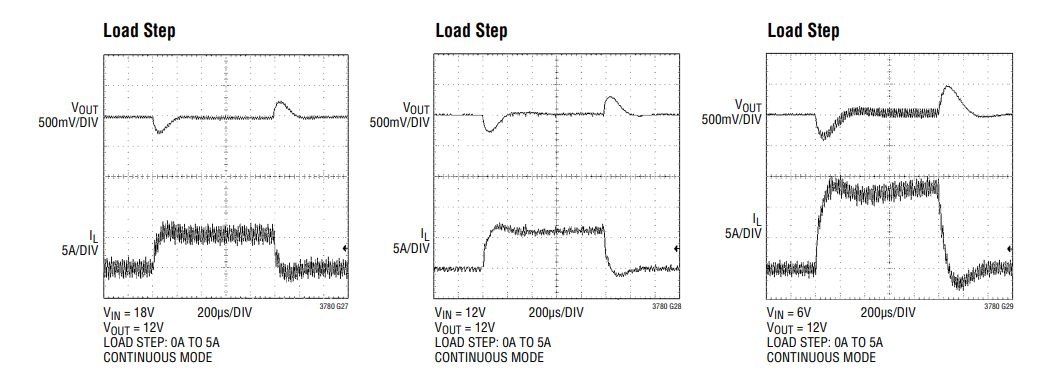
where COUT is the output filter capacitor. The steady ripple due to the voltage drop across the ESR is given by:

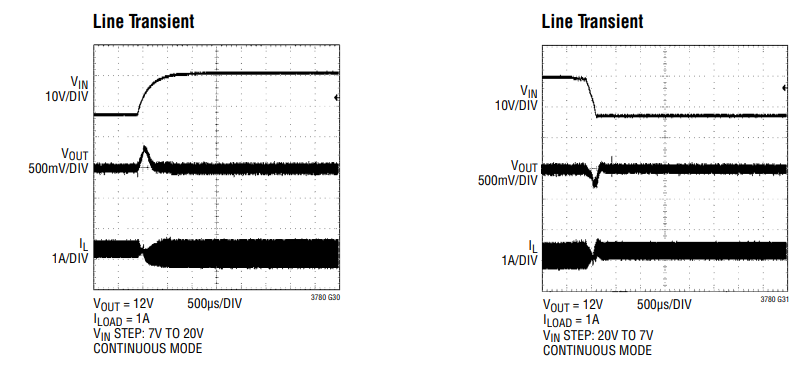
∆VBOOST,ESR = IL(MAX,BOOST) • ESR

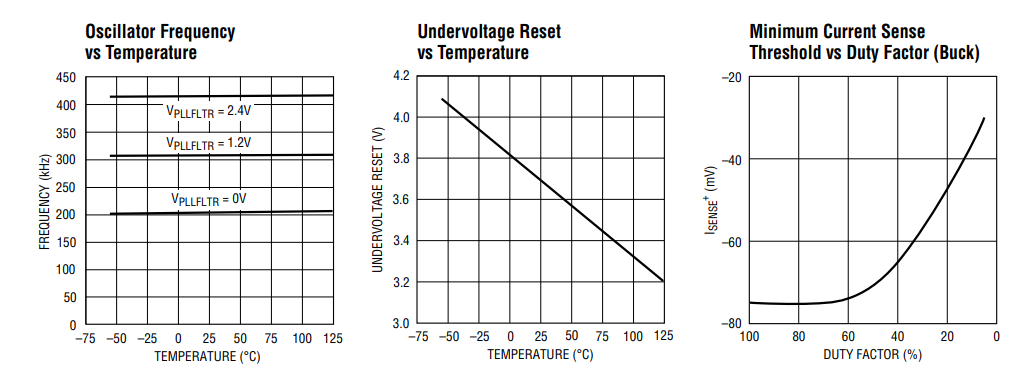
∆VBUCK,ESR = IL(MAX,BUCK) • ESR

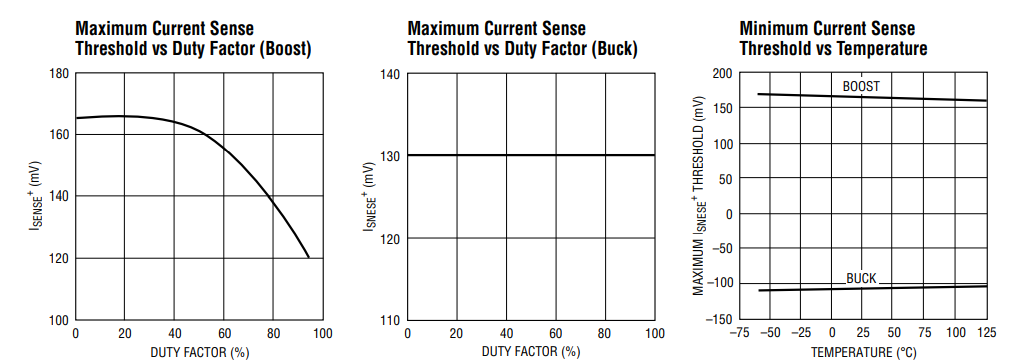
Multiple capacitors placed in parallel may be needed to meet the ESR and RMS current handling requirements. Dry tantalum, special polymer, aluminum electrolytic and ceramic capacitors are all available in surface mount packages. Ceramic capacitors have excellent low ESR characteristics but can have a high voltage coefficient. Capacitors are now available with low ESR and high ripple current ratings, such as OS-CON and POSCAP.

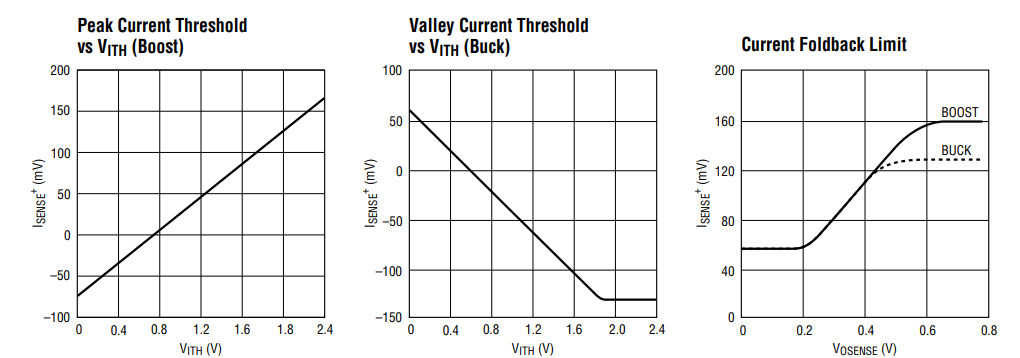
The performance graph of the buck boost converter is shown below:











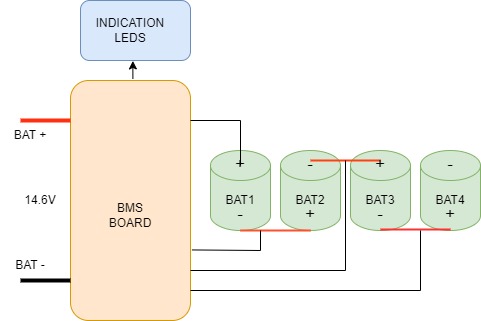
1. Battery Management system.

**BMS (Battery Management System) board** is a control circuit designed for managing and protecting a 4-cell LFP battery pack. "4S" refers to the battery configuration of four cells in series, which typically yields a nominal voltage of 12.8V (3.2V per cell). The BMS board monitors and balances each cell's voltage, ensuring they stay within safe limits to prevent overcharging, over-discharging, and over current. Additionally, it provides temperature protection and sometimes communication features, optimizing the battery’s performance, lifespan, and safety.

The specification of the BMS are as below:

|  |  |
| --- | --- |
| **Single cell battery overcharge protect voltage** | 3.61-3.69 V |
| **Single cell battery overcharge protect recover voltage** | 3.5-3.6V |
| **Overcharge protect delay** | 1.5-2.5S |
| **Overcharge protect Current** | 100A |
| **Over-discharge protect Current** | 100A |
| **Disconnect protect** | Yes |
| **Shortage protect** | Yes |
| **Charging curren** | 60A |
| **Voltage alance accuracy** | 3.355-3.455 V |

Battery connection:



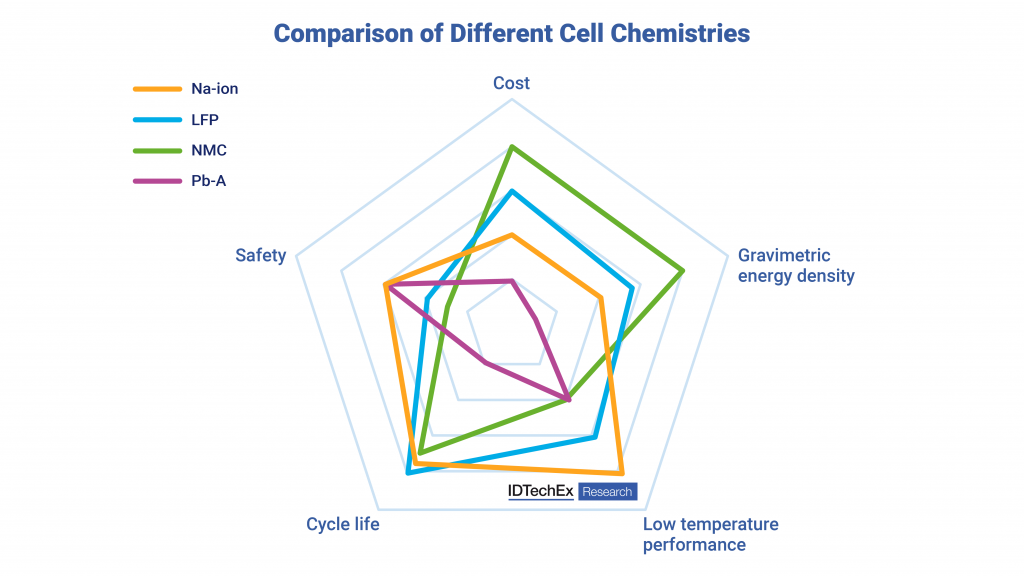
1. LFP Battery bank.

### ****Lithium Iron Phosphate (LFP) batteries**** are a type of lithium-ion battery that uses lithium iron phosphate (LiFePO₄) as the cathode material. LFP batteries are known for their stability, long cycle life, and enhanced safety compared to other lithium-ion chemistries. They are widely used in applications that require reliable power storage, such as renewable energy systems, electric vehicles, and backup power.Advantages of LFP Batteries

1. **Safety and Stability**: LFP batteries are more stable and have a lower risk of overheating or thermal runaway, making them one of the safest lithium-ion options.
2. **Long Cycle Life**: These batteries can typically handle 2,000 to 5,000 charge-discharge cycles, far exceeding other lithium-ion chemistries, which generally last 500-1,500 cycles.
3. **High Temperature Tolerance**: LFP batteries perform well in higher temperature environments without degrading as quickly as other lithium-ion batteries.
4. **Environmental Friendliness**: They contain no cobalt, which is environmentally harmful to mine, and are often considered more sustainable.
5. **Consistent Power Output**: LFP batteries maintain stable voltage during discharge, providing steady power output throughout their cycle.

Overall, LFP batteries are ideal for applications prioritizing safety, longevity, and sustainability, even if they sacrifice some energy density and cold-weather performance.

Specification of single LFP cell:



Each cell capacity around 6 Ah 3.2V nominal so 4 cells in series and 6 sets in parallel to achieve 14.6V 36Ah battery pack. The combination is technically called as 4S 6P combination.

The Charge rate of the cells are 1 C charge and 2 C discharge. Which means it can supply 14.6V 6A current means

6X6X14.6 x 1 = 525.6 Watts for charging.

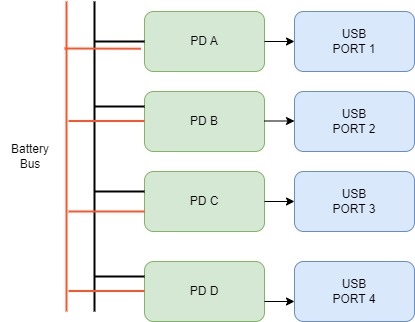
6X6X14.6 x 2 = 1051.2 Watts for Discharging.

The maximum load which can be connected to this battery bank is 1Kw for long life operation.

1. **USB PD controller**

sw3518s fast charging module dc6-32v usb pd3.0 pps mobile phone fast charging board step down module 4 way at best price. This DC6-32V USB PD3.0 fast charging module supports a wide range of devices. Designe to deliver efficient power, it also ensures quick charging with precision.

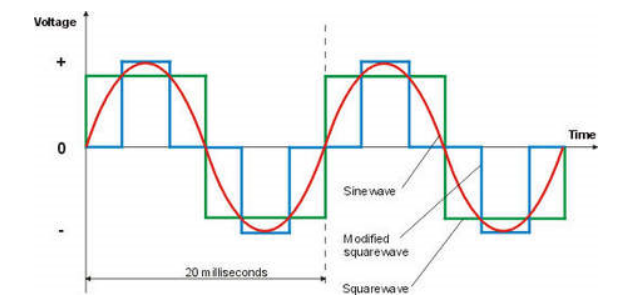
This SW3518S Mobile Phone Fast Charging Board offer four way charging capability & enhance its use. It also support usb PD3.0 (PPS) for efficient and rapid charging. With an input range of DC 6-32V, it is compatible with numerous power source. This fast charging board is perfect for both mobile device and other usb powered gadget.



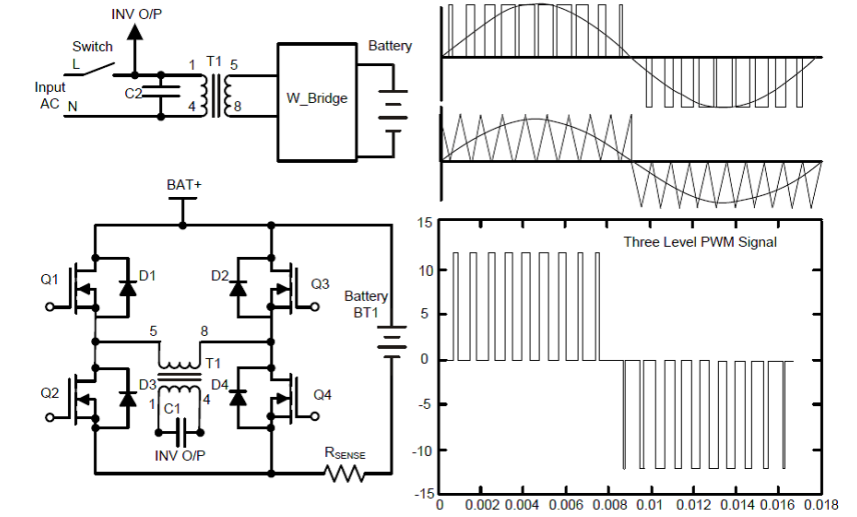
1. **Inverter Block**

Power inverter is a device that converts electrical power from DC form to AC form using electronic circuits. It is typical application is to convert battery voltage into conventional household AC voltage allowing you to use electronic devices when an AC power is not available. There are basically three kinds of Inverter out of which, the first set of inverters made, which are now obsolete, produced a Square Wave signal at the output.

The Modified Square Wave also known as the Modified Sine Wave Inverter produces square waves with some dead spots between positive and negative half-cycles at the output. The cleanest utility supply like power source is provided by Pure Sine Wave inverters. The present Inverter market is going through a shift from traditional Modified Sine Wave Inverter to Pure Sine Wave inverters because of the benefits that these inverters offer.



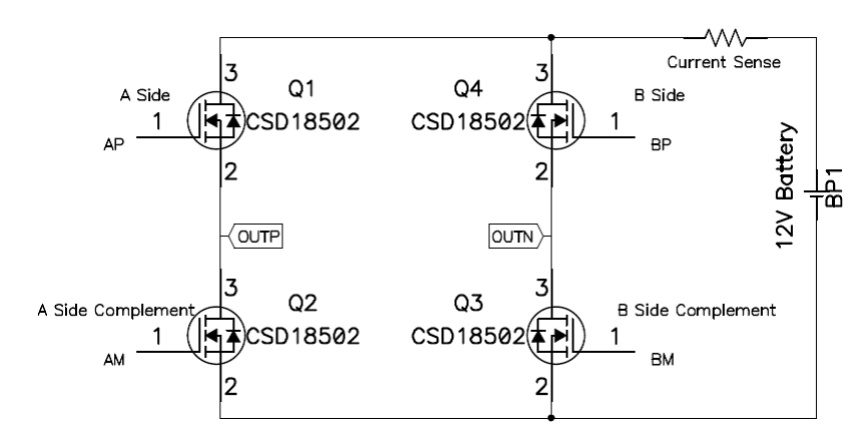
The Stepped sinewave is created using a special algorithm using 8051 controller. Which controls the output current and monitors the input voltage.The High level schematics is shown below.



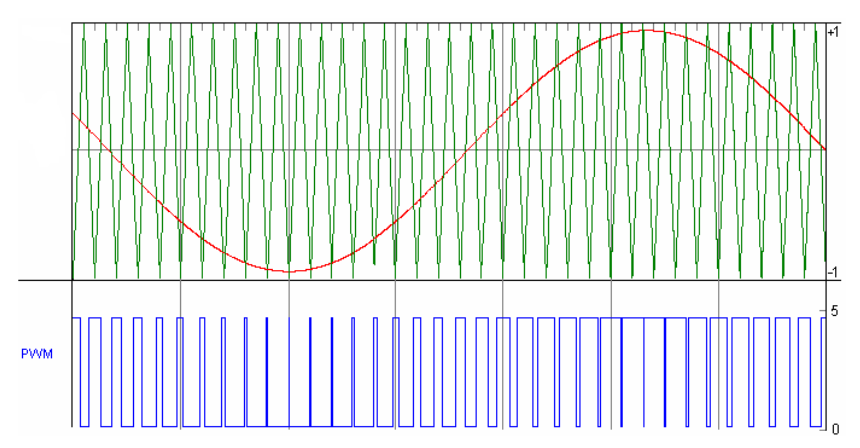
High level block Diagram for Stepped Sine Inverter

To understand the functioning of an Inverter, the user must understand the switching requirement of the four drives of the MOSFETs in H Bridge both in Inverter as well as Mains mode.

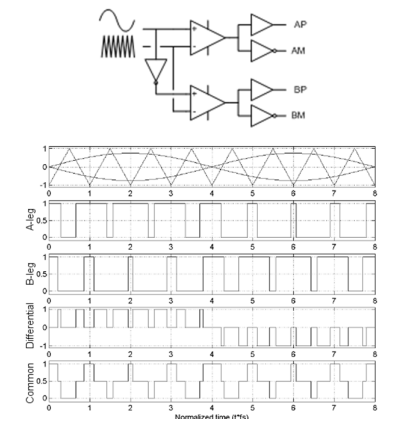
1. **Inverter Mode** The Switching Wave Form in an Inverter is very simple to understand and generate.



On the A Side MOSFET of the H Bridge, the PWM is generated by modulating the Sine Wave with high frequency (6 KHz to 20 KHz) Square wave in such a way that the positive peak of the Sine Wave is represented by maximum duty cycle and the negative peak by the minimum duty Cycle as shown in below figure.

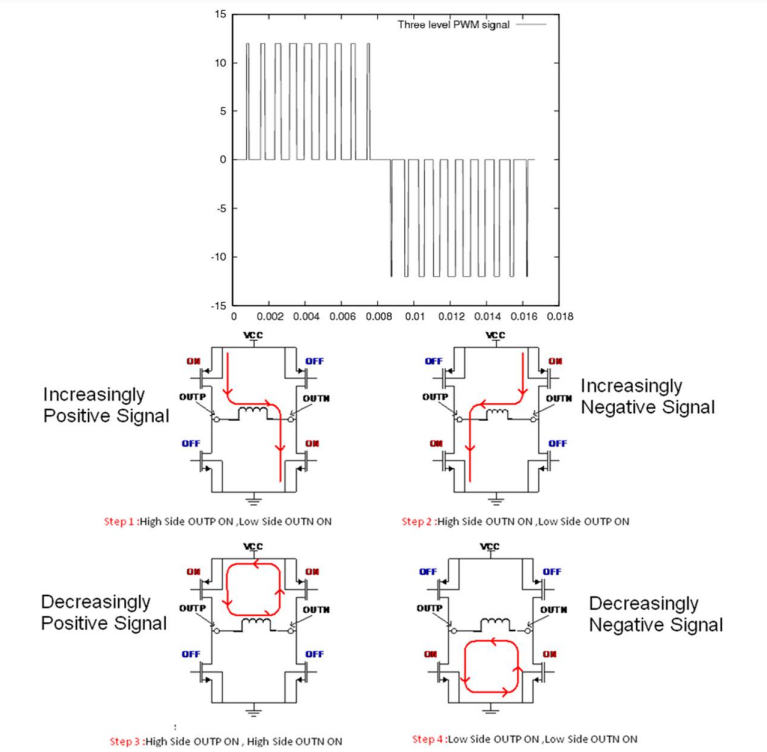
**Modulation of Sine Wave With Higher Frequency PWM Signals**

Now on the B Side, just phase shift this Sine Wave by 180 degree and generate the PWM in a similar Way as mentioned above. The following simple hardware implementation of the PWM generation will make the design more clear.



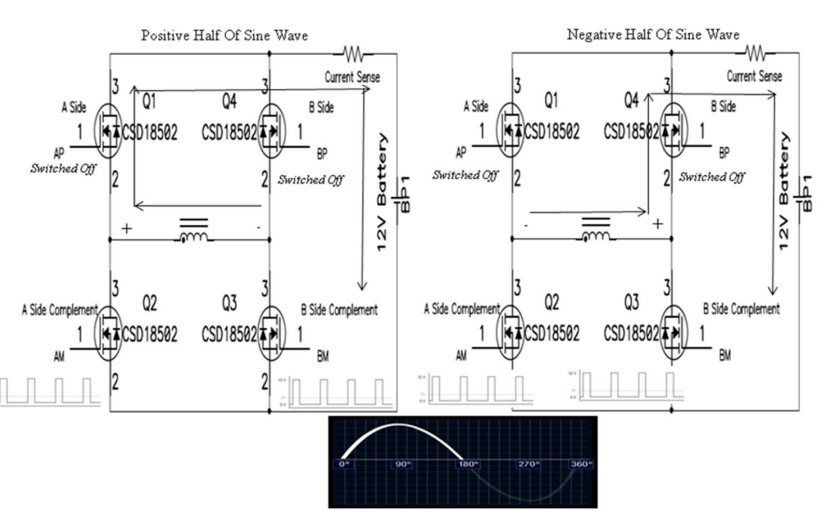
**Waveform Generation in Inverter Mode**

A side complementary or the AM signal is obtained by just inverting the A side or AP waveform and the same goes for B Side complementary or BM waveform.The differential signal seen across the OUTP and OUTN will be a Trilevel PWM Signal as mentioned in below images.



**Trilevel PWM Signal During the Inverter Mode for Pure Sine Wave Generation**

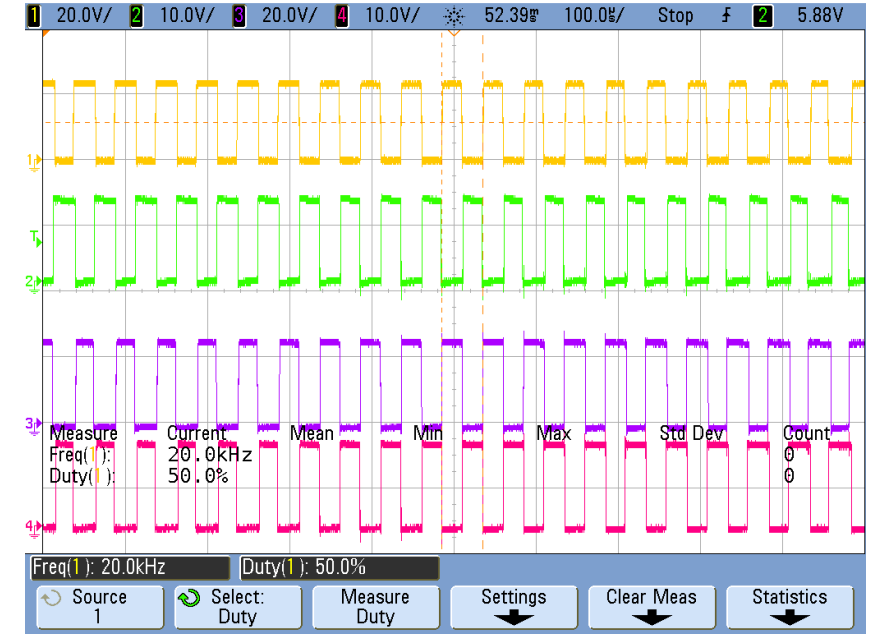
1. **Mains Mode:** In the mains mode, both the high-side MOSFETs ie A side as well B side is switched off and both the low-side MOSFETs are switched with the similar PWM waveform where the duty cycle of lower side PWM signals determine the charging current.



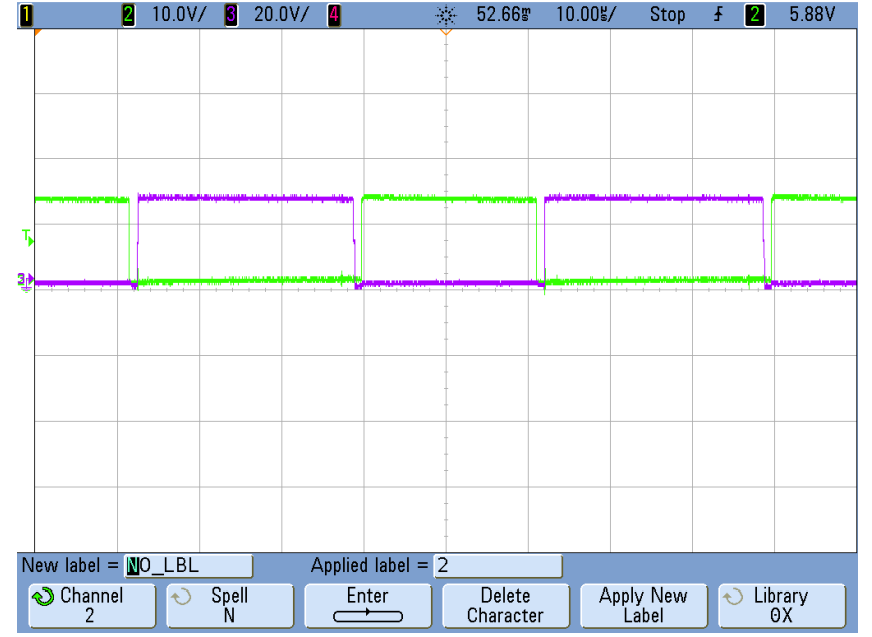
When the lower switches are turned on at the same time, there is a boosted voltage, that appear across the primary leakage inductance of transformer connected to the H–Bridge, by the Ldi/dt effect and this energy is use to charge the battery through the body diodes of the high-side MOSFETs. Also each of the high-side MOSFET’s body diode will conduct in the each half of the Sine Wave.

When the mains mode is sensed, firstly all the MOSFETs are switched off and the Relay between the Ac input and the Inverter output is connected. After this, the Lower FETs are tuned on with PWM of small duty Cycle (5 to 10 percent) and the high-side MOSFETS are switched off. Now the voltage across the current sense is measured by controller and if the corresponding current is less or more than required by charging algorithm than the duty cycle is altered correspondingly ie duty cycle is increased if more charging current is required and decreased if the charging current reduction is desired.

The pure sine wave is tapped out from the transformer and given to AC outlets, There are there AC outlets are provided in the circuit.

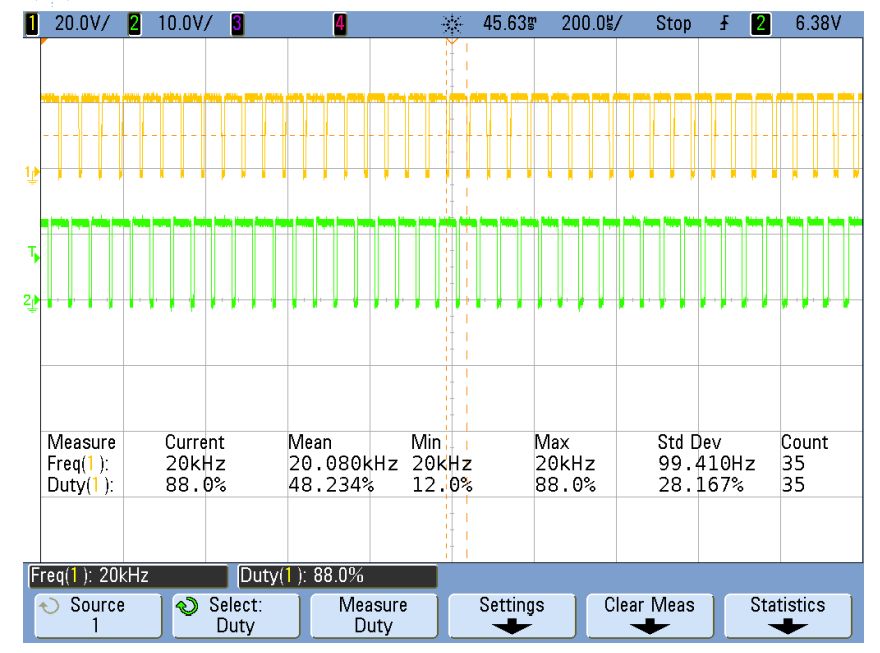


**Inverted Waveform (HOA-LOA and HOB-LOB) at the Gates of MOSFETS.**



**Dead Band between Complementary HOB and LOB Pair**

**PWM switching at the gates of the MOSFETs at no load (inverter mode) with 12-V battery input:**



In the LF inverter, the battery voltage is first chopped with the full bridge (using high-frequency PWM, generally 3 kHz to 20 kHz) to an AC waveform. The iron core transformer then boosts the 12-V chopped waveform to 220-V RMS output waveform at 50 Hz. At the output of the transformer, a capacitor helps filters the waveform to make a clean 50-Hz AC Sine Wave. Although this inversion method is widespread today, the iron core transformer is quite bulky and increases the cost of the overall solution. The LF inverters use SM72295 – a highly integrated gate driver with two high-side, current-sensing amplifiers – AMC1100 for AC mains current sensing.

The next advancement in inversion technology is the use of high frequency inverters, or HF inverters. This technology involves more processing complexity but can significantly increase overall system efficiency and eliminate the use of the bulky iron core transformer. In a high-frequency inverter, the battery voltage is converted to an intermediate high DC voltage before it’s converted to an AC waveform using Pulse Width modulation. Here, we demonstrate an example implementation using current fed push-pull topology.

During the inverter stage (when the AC mains is not present), the push-pull stage converts the battery voltage to a high voltage in the range of 350 V, which is then chopped with the help of full bridge PWM driven by a microcontroller to form 220-V AC. During the charging stage, the AC main is rectified using the body diodes in the full bridge stage and the second stage acts as half bridge. Thus, bidirectional control is achieved without the need of any additional components.

Appendinx:

MOSFET - Metal Oxide Semiconductor Field Effect Transistor

LFP - Lithium Ferro Phosphate

MPPT - Maximum power point tracking

PV - Photo voltaic

LC - incuctor and capacitor

LPF - Low pass Filter

AC - Alternating current

DC - Direct current

RMS - Root Mean Square.

HF - High Frequency

LF - Low Frequency

PWM - Pulse width modulation

FET - Field effect Transistor

FM - Frequency Modulation

AM - Amplitude Modulation

USB - Universal Serial Bus

PD - Power Delivery

**PHASE – 2**

**Phase 2: System Upgrade - LCD Display Integration for Battery Monitoring**

**Overview:**

In Phase 2 of the solar nano-grid project, a significant upgrade was implemented to improve user accessibility and system transparency — the integration of an LCD display module to monitor the battery bank voltage in real-time. This enhancement provides immediate visual feedback on the system's state of charge, helping users better manage energy usage and optimize load distribution.

**Objective:**

The primary objective of this phase is to introduce a cost-effective and user-friendly interface that displays the live battery voltage. This empowers users with critical data to avoid battery over-discharge, plan power usage effectively, and maintain battery health over the long term.

**Hardware Upgrade:**

A 16x2 character LCD based on the Hitachi HD44780 driver was selected for its simplicity, widespread availability, and compatibility with most microcontrollers. It is connected via a 4-bit or I2C interface to the system's microcontroller (such as Arduino or STM32), which continuously reads the battery voltage using the onboard ADC and displays the value on the LCD screen.

**Key Features:**

* **Live Battery Voltage Display** (e.g., "Battery Voltage: 13.8V")
* **Low Voltage Warning** (e.g., blinking text or alert below 11.5V)
* **Backlight Control** for better visibility
* **Compact & Low Power Consumption.**

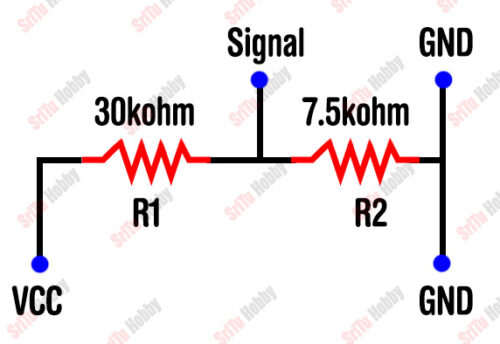
**Working Principle:**

The voltage at the battery output terminal is read using a voltage divider circuit to step down the voltage to a range readable by the ADC of the microcontroller (usually 0–5V). The microcontroller then processes this analog value, converts it into a voltage level, and updates the LCD with the measured battery voltage every few seconds.

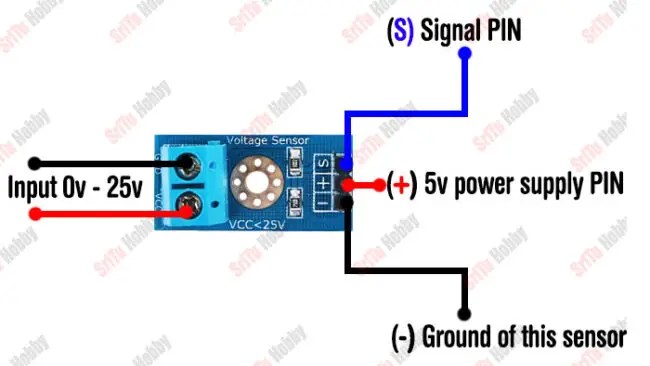
**How the voltage sensor module works with Arduino?**

**How does the voltage sensor work?**

This sensor is made using a voltage divider circuit. Also, this voltage divider circuit includes two resistors of 30kohm and 7.5kohm. When a potential is applied to this circuit, we can get the voltage difference in the voltage divider point via the signal pin. So, the voltage can be calculated by getting it to the Arduino board through an analog pin. The voltage divider circuit is as follows.



**The PIN structure of this sensor**

****

**how to connect this sensor to the Arduino UNO board**

* Arduino UNO board x 1
* Voltage sensor x 1
* LCD display x 1
* I2C module x 1
* Jumper wire

**Step 1**

Firstly, identify these components.

**Arduino UNO board**



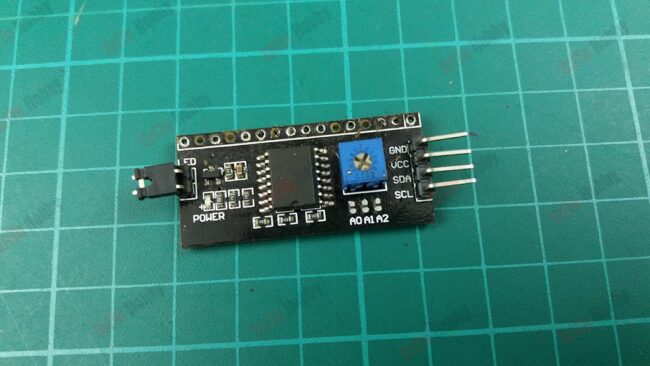
**Voltage sensor**



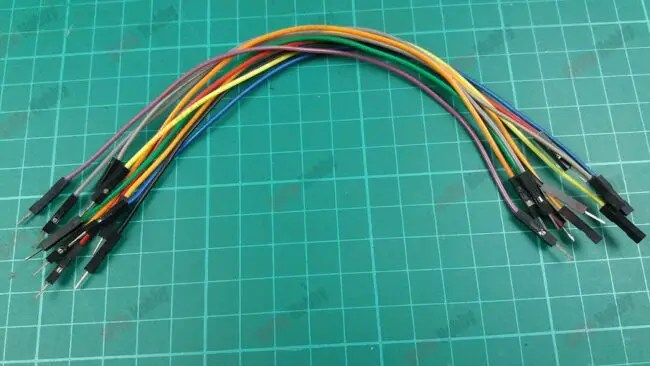
**LCD screen**



**I2C module**

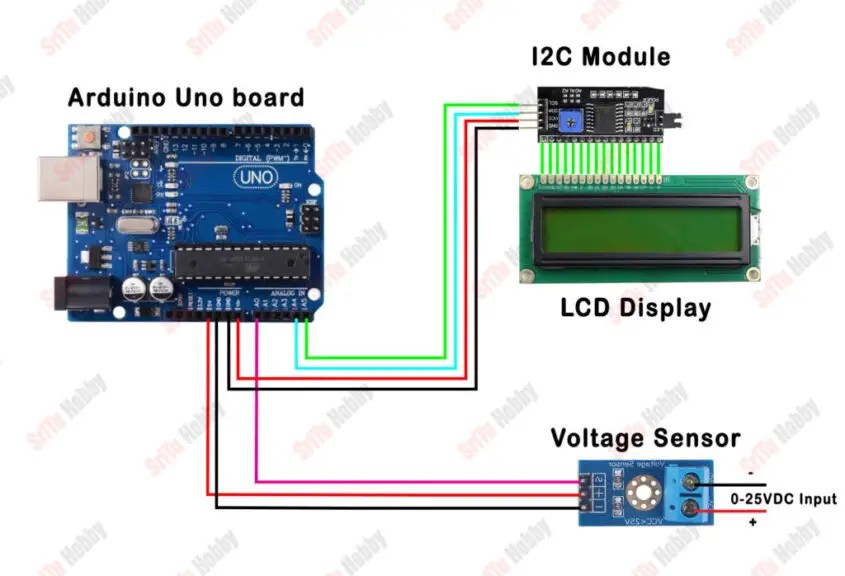


**Jumper wires**



**Step 2**

Secondly, connect these components. To do this, use the circuit diagram below.



**Step 3**

Thirdly, let’s create the program for this project.

* I2C library
* The complete program of this project

Code for Arduino and I2C to display voltage in display:

|  |
| --- |
| #include <LiquidCrystal\_I2C.h>  LiquidCrystal\_I2C lcd(0x27, 16, 4);  #define Sensor A0  float vOUT = 0.0;  float vIN = 0.0;  float R1 = 30000.0;  float R2 = 7500.0;  void setup() {  Serial.begin(9600);  lcd.init();  lcd.backlight();  }  void loop() {  int value = analogRead(Sensor);  vOUT = (value \* 5.0) / 1024.0;  vIN = vOUT / (R2 / (R1 + R2));  lcd.setCursor(0, 0);  lcd.print("Voltage :");  lcd.print(vIN);  lcd.print("v ");  Serial.print("Voltage : ");  Serial.println(vIN);  } |

**Code explanation**

Firstly, the I2C library is included and creates an object for the library. Also, it includes the I2C address and size of the LCD

|  |
| --- |
| #include <LiquidCrystal\_I2C.h>  LiquidCrystal\_I2C lcd(0x27, 16, 4); |

Secondly, the sensor pin is defined. Also, four variables have been created to help the program.

|  |
| --- |
| #define Sensor A0  float vOUT = 0.0;  float vIN = 0.0;  float R1 = 30000.0;  float R2 = 7500.0; |

In the setup function, the serial monitor and LCD are enabled.

|  |
| --- |
| void setup() {  Serial.begin(9600);  lcd.init();  lcd.backlight();  } |

In the loop function, the sensor values are taken and the voltage is calculated using the formula. These R1 and R2 resistors are in the voltage divider circuit. The voltage divider circuit is described above.

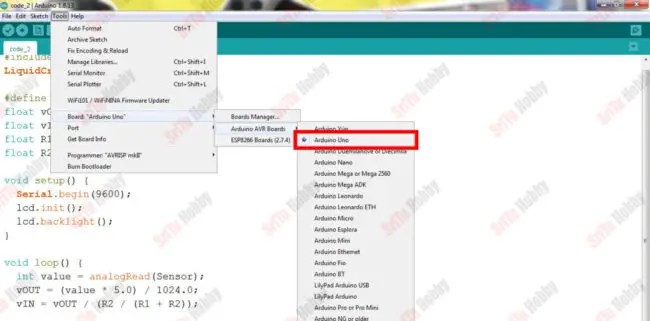
|  |
| --- |
| void loop() {  int value = analogRead(Sensor);  vOUT = (value \* 5.0) / 1024.0;  vIN = vOUT / (R2 / (R1 + R2)); |

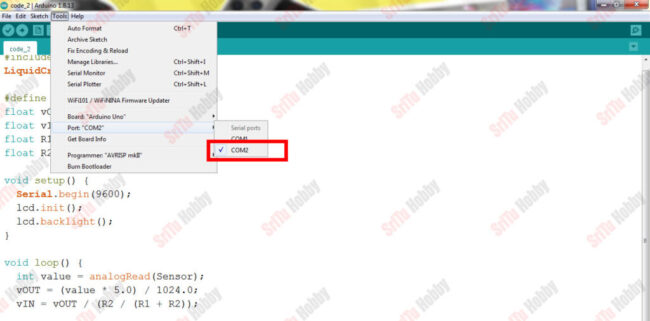
This code prints the voltage values on the serial monitor and LCD.

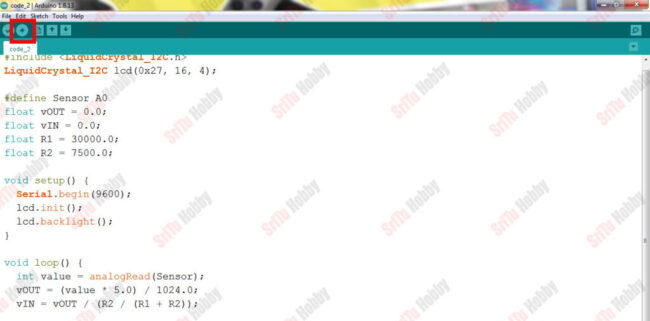
|  |
| --- |
| lcd.setCursor(0, 0);  lcd.print("Voltage :");  lcd.print(vIN);  lcd.print("v ");  Serial.print("Voltage : ");  Serial.println(vIN);  } |

**Step 4**

OK, now select the board and port, Afterward, upload this code to the Arduino board.





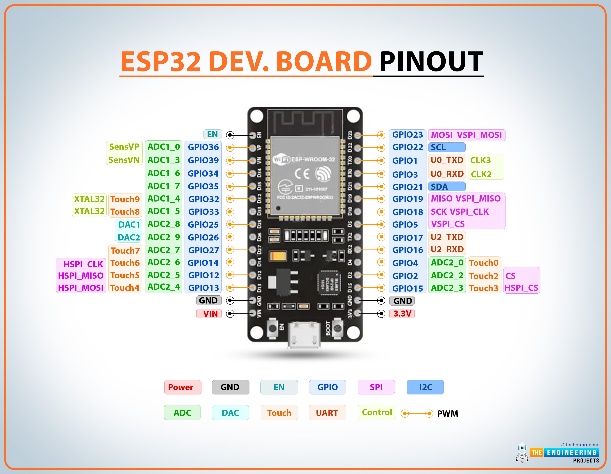


Give voltage sources from 0v to 25v. For this project, we can also use the following program.

Code for I2C to config with LCD board.

|  |
| --- |
| #include <LiquidCrystal\_I2C.h>  #define Sensorpin A0  LiquidCrystal\_I2C lcd(0x27, 16, 4);  void setup() {  Serial.begin(9600);  lcd.init();  lcd.backlight();  }  void loop() {  int value = analogRead(Sensorpin);  double voltage = map(value, 0, 1024, 0, 2500);  voltage /= 100;  lcd.setCursor(1, 0);  lcd.print("Voltage :");  lcd.print(voltage);  lcd.print("v "); Serial.print("Voltage : ");  Serial.println(voltage);  } |

**ESP32 Wi-Fi Light Control System Using Relay**



**ESP32 Microcontroller — Full Details**

The ESP32 is a powerful, low-cost, low-power system-on-chip (SoC) microcontroller designed by Espressif Systems. It has built-in Wi-Fi and Bluetooth and is ideal for IoT (Internet of Things) applications.

**Key Features of WIFI controller**

* **Wi-Fi**: 802.11 b/g/n up to 150 Mbps
* **Bluetooth**: v4.2 BR/EDR and BLE
* **Processor**: Dual-core 32-bit Xtensa® LX6, up to 240 MHz
* **Memory**: 520 KB SRAM, external Flash (commonly 4MB or more)

**Common Peripherals**

| **Type** | **Quantity & Notes** |
| --- | --- |
| GPIO | Up to 34 general purpose I/O pins |
| ADC | 18 channels, 12-bit resolution |
| DAC | 2 channels (8-bit) |
| PWM | Available on almost all GPIOs |
| UART | 3 hardware serial ports |
| SPI | 4 SPI interfaces |
| I2C | 2 I2C interfaces |
| CAN | Yes (not on all models) |
| Touch Sensors | 10 capacitive touch sensors |
| Hall Sensor | Built-in magnetic sensor |

**Voltage & Power**

* Operating Voltage: 3.0V–3.3V
* I/O Voltage: 3.3V logic (⚠ 5V on GPIO may damage it!)
* Power via: Micro-USB or 3.3V to 3V3 pin
* Deep sleep current: ~10µA

**Programming**

* Arduino IDE
* ESP-IDF
* PlatformIO
* MicroPython or Lua

**Development Boards**

* MH-ET LIVE ESP32 DevKit
* DOIT ESP32 DevKit v1
* NodeMCU-32S
* Adafruit HUZZAH32

**Typical Pinout Overview**

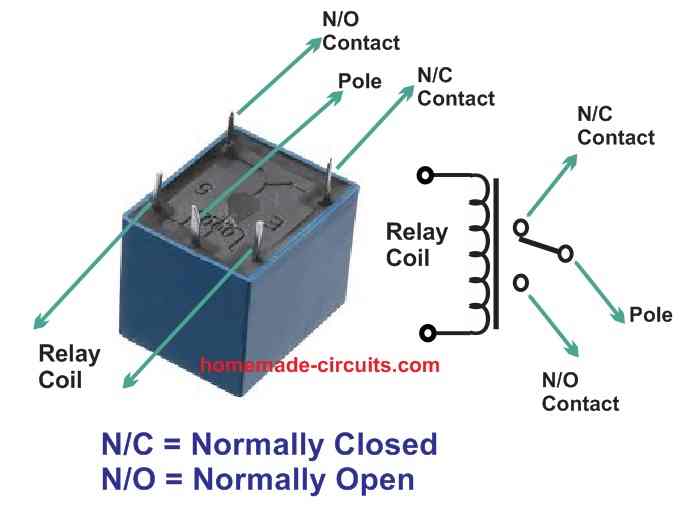
* 3V3, GND, EN, IO0, IO2-IO15: Common GPIOs
* IO34–IO39: Input only

Relay Image



**What Is a Relay**

A relay is an electrically operated switch. It lets a low-voltage device like ESP32 control a high-voltage circuit (e.g., 230V AC light).



**KYOTA KT-603 Relay Module (or similar) or relay module**

Includes:

* Mechanical relay (blue cube)
* Input: IN, VCC, GND
* Output: COM, NO, NC

**Relay Specifications**

* Coil Voltage: 3V DC
* Control Current: ~70–100mA
* Trigger Level: Active LOW
* Switching Capacity: 10A at 120V AC, 7A at 240V AC
* Type: SPDT (Single Pole Double Throw)
* Isolation: Optocoupler (in some versions)

**Input Pins Description**

| **Pin** | **Description** |
| --- | --- |
| IN | Signal pin (to ESP32 GPIO) |
| VCC | Power pin (3.3V or 5V) |
| GND | Connect to ESP32 GND |

**Powering the ESP32 and Relay**

* ESP32: via USB or regulated 5V/3.3V
* Relay: via 5V (from USB, battery, adapter)
* Important: **Common GND** for both modules

**ESP32 Creates a Wi-Fi Network (Access Point Mode)**

WiFi.softAP(ssid, password);

* Creates: ESP32\_Light\_Control
* Password: 12345678

**ESP32 Acts as Web Server**

Wi-Fi Server server (80);

server.begin();

* Access via browser: 192.168.4.1

**Web Interface with Buttons**

* ✅ ON button sends GET /on
* ❌ OFF button sends GET /off

**ESP32 Handles Request and Controls Relay**

if (request.indexOf("GET /on") >= 0)

* LOW = Relay ON
* HIGH = Relay OFF

**Relay Real World Switching**

* COM → Power Source
* NO → Light
* When relay ON: COM & NO connect → Light turns ON

**Complete Arduino Code Explanation**

#include <WiFi.h>

const char\* ssid = "ESP32\_Light\_Control";

const char\* password = "12345678";

const int relayPin = 26; // GPIO26

WiFiServer server(80);

void setup() {

Serial.begin(115200);

pinMode(relayPin, OUTPUT);

digitalWrite(relayPin, HIGH); // Relay OFF initially (active LOW)

WiFi.softAP(ssid, password);

Serial.println("✅ Access Point started");

Serial.print("📶 IP address: ");

Serial.println(WiFi.softAPIP());

server.begin();

}

void loop() {

WiFiClient client = server.available();

if (client) {

Serial.println("🔗 New Client connected");

String request = "";

while (client.connected()) {

if (client.available()) {

char c = client.read();

request += c;

if (c == '\n') {

if (request.indexOf("GET /on") >= 0) {

digitalWrite(relayPin, LOW); // Relay ON

Serial.println("💡 Light ON");

}

if (request.indexOf("GET /off") >= 0) {

digitalWrite(relayPin, HIGH); // Relay OFF

Serial.println("💡 Light OFF");

}

client.println("HTTP/1.1 200 OK");

client.println("Content-type:text/html\n");

client.println("<!DOCTYPE html><html><head><title>ESP32 Light Control</title></head><body>");

client.println("<h2>💡 ESP32 Relay Control</h2>");

client.println("<a href=\"/on\"><button style='padding:30px;font-size:24px;background:green;color:white;'>ON</button></a>");

client.println("<a href=\"/off\"><button style='padding:30px;font-size:24px;background:red;color:white;'>OFF</button></a>");

client.println("</body></html>");

break;

}

}

}

client.stop();

Serial.println("❌ Client disconnected");

}

}

**System Summary Flow**

1. Power ESP32 → Wi-Fi network created
2. Connect phone to ESP32 Wi-Fi
3. Open browser → 192.168.4.1
4. Tap ON/OFF
5. ESP32 receives command → switches GPIO
6. Relay turns light ON/OFF

**Diagram: ESP32, Relay, and Bulb Control using Wi-Fi]**

