Institute of Space Technology

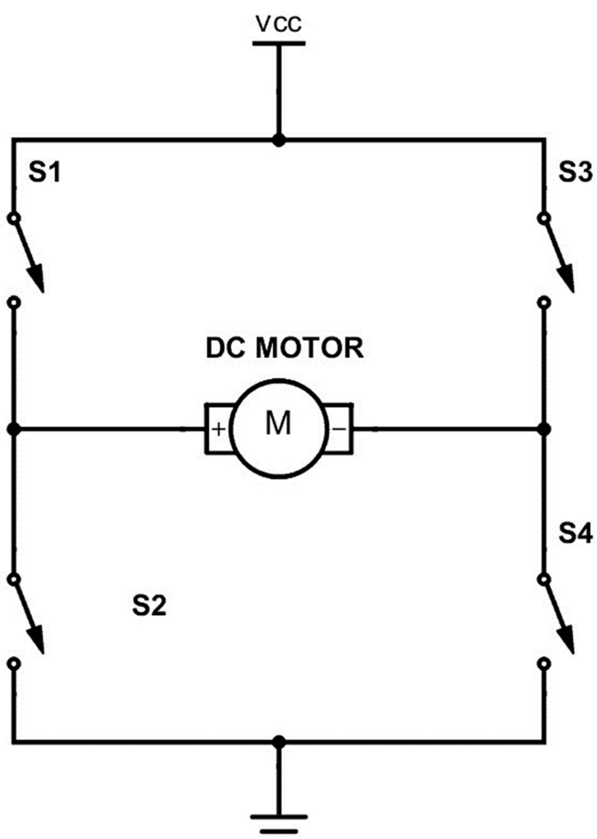
ECD Lab Project
Square Wave Inverter (400W)

Submitted By:
Fahad Qalbi

A **DC to AC square wave inverter** is an electronic device that converts direct current (DC) from a source such as a battery or solar panel to alternating current (AC). The H-bridge configuration is often used to create this AC output. While there are various types of inverters such as pure sine wave inverters and modified sine wave inverters, the simplest form is the square wave inverter.

H-bridge Configuration

The H-bridge is a configuration of four switches, usually MOSFETs or IGBTs, arranged in the shape of the letter "H." These switches are used to control the direction of current flow through a load, which in this case, is typically a transformer or the inductive load that requires AC power.



How it Works

In a DC to AC square wave inverter using an H-bridge:

1. The two switches on one diagonal of the H-bridge are closed, allowing current to flow from the DC source through the load in one direction. This creates a positive voltage across the load.

2. After a short period, these switches are opened, and the two switches on the opposite diagonal are closed. This causes current to flow in the opposite direction, creating a negative voltage across the load.

3. This rapid switching between positive and negative voltages effectively generates a square wave AC output.

The rate at which these switches are toggled determines the frequency of the AC output. For example, in many applications, the switches are toggled at 50 or 60 times per second (50Hz or 60Hz) to mimic the frequency of the AC power available from the mains supply.

Applications and Limitations

Square wave inverters are simple and inexpensive to build, making them suitable for basic applications such as running motors or simple resistive loads. However, the square wave output is not ideal for sensitive electronics or devices that require a smooth sine wave input, as the abrupt changes in voltage can cause harmonic distortion, increased heat, and potentially damage to sensitive components. For these applications, more sophisticated pure sine wave inverters are used.

In summary, a DC to AC square wave inverter using an H-bridge configuration is a fundamental device for converting DC power to AC. While it has its limitations due to the nature of the square wave output, it is a foundational building block in power electronics.

GATE DRIVER IC (IR2112):

A gate driver IC is used to control and drive the gate of a power electronic device such as a MOSFET or IGBT. These power devices are often used in high-power applications such as motor drives, inverters, and power converters. The gate driver IC serves as an interface between a low-power control signal and the high-power switching device.

Here's how a gate driver IC typically works:

Level Shifting:

Often, the control signals (such as PWM signals) are at low voltage levels, like 3.3V or 5V, which are not sufficient to properly turn on the power devices, especially in high voltage applications. The gate driver IC shifts these low voltage control signals to the appropriate levels needed for driving the gates of the power devices. For example, a MOSFET may require 10V or 15V at the gate to be properly turned on.

Current Boosting:

The gate of a MOSFET or an IGBT behaves like a capacitor. To turn the device on and off quickly, a large current is needed to charge and discharge this capacitance. The gate driver IC can supply this higher current, allowing for fast switching times. This is essential in reducing switching losses in power electronics applications.

Protection:

Many gate driver ICs have built-in protection features such as under-voltage lockout, over-current protection, and desaturation detection. These features help protect the power device from conditions that could cause failure or damage.

Isolation:

In some applications, electrical isolation between the control side and the power side is necessary for safety and noise immunity. Some gate driver ICs provide galvanic isolation, often using transformers or optocouplers, to ensure that no direct electrical connection exists between the input and output.

Driving Complementary Devices:

In applications like H-bridge circuits, two power devices need to be driven in a complementary fashion (when one is on, the other must be off). Gate driver ICs often have two outputs designed to drive a pair of devices in this manner. This can also include dead-time insertion, where there is a small delay between turning one device off and turning the other on, to prevent shoot-through current.

Providing Bootstrap Voltage:

In some configurations like high-side switching, the gate voltage needs to be higher than the supply voltage. Gate driver ICs often have bootstrap circuits which use a small capacitor to provide a higher voltage for the gate drive.

Overall, gate driver ICs are crucial components in power electronics, enabling efficient and safe operation of switching devices in a wide range of applications.

Bootstrap voltage is a technique used in high-side gate driver circuits to provide a voltage that is higher than the supply voltage, in order to properly turn on a high-side MOSFET or IGBT. This is particularly important in configurations such as H-bridges, where the high-side switch needs a gate voltage higher than the supply voltage to be fully turned on.

Here's how the bootstrap voltage technique typically works:

Components:

The basic components involved in creating a bootstrap voltage are a bootstrap diode, a bootstrap capacitor, and a gate driver IC.

Charging Phase:

When the low-side switch (MOSFET or IGBT) is turned on, current flows through the bootstrap diode and charges the bootstrap capacitor. The voltage across the capacitor is approximately equal to the supply voltage minus the diode's forward voltage drop.

Driving Phase:

When the high-side switch needs to be turned on, the low-side switch is turned off. The gate driver uses the voltage stored in the bootstrap capacitor to provide the necessary voltage to the gate of the high-side switch. Because the source of the high-side switch is also rising (it is connected to the switched node), the gate voltage needs to be higher than the supply voltage to ensure that the gate-source voltage (Vgs) is sufficient to fully turn on the switch.

Refresh Phase:

As the high-side switch remains on, the bootstrap capacitor will slowly discharge due to the gate's capacitance and any leakage currents. Therefore, it is important to periodically turn off the high-side switch and turn on the low-side switch to recharge the bootstrap capacitor. This is naturally done in applications like PWM motor control, where the switches are being toggled at a high frequency.

Bootstrap Resistor:

Often, a resistor is used in series with the bootstrap diode to limit the inrush current when charging the bootstrap capacitor.

The bootstrap technique is a simple and efficient way to provide the necessary gate drive voltage for high-side switches without the need for an additional high-voltage power supply. However, it's important to ensure that the switching frequency is high enough to keep the bootstrap capacitor charged, and the duty cycle should not be 100%, as some off-time is needed for recharging the capacitor.

Procedure:

First to design the H-bridge we require the values of input voltages and how much power application maximum we want. With these we can measure the peak current a MOSFET has to deal with. The calculations done for my project are given below:

Date:

Square
400 watt Inverter

↳ Since Given 12V DC.

↳ at output peak to peak
max can be = $24V_{pp}$

↳ $V_{rms} = \frac{24}{2\sqrt{2}} = 8.485\ V$

↳ $P = VI$
$400 = 8.48V \times I$
↳ $I = 47.1698$ Amps RMS

↳ $I_{peak} = 66.708$ Amps

So, we have to chose ~~a~~
mosfets that can hanndle 66.708 peak
current & V_{peak} of 12 Volts.

Now that we know the value of current and voltage, we can choose a suitable MOSFET for our application.

The MOSFET we chose is IRL2203N.

A SHORT DISCRIPTION about IRL2203N:

The IRL2203N is an N-channel MOSFET (Metal Oxide Semiconductor Field-Effect Transistor) produced by Infineon Technologies (formerly International Rectifier). It is designed for fast switching and high current handling, making it suitable for various applications including power conversion, motor control, and general-purpose switching.

Here are some key features and specifications of the IRL2203N:

1. N-Channel:

Being an N-Channel MOSFET, it is turned on (conducts) when a positive voltage is applied to its gate relative to its source.

2. Logic Level Gate Drive:

The "L" in IRL indicates that this MOSFET can be driven by logic-level voltages. This means it can be fully turned on with a relatively low gate-source voltage (e.g., 5V), making it compatible with microcontrollers and other logic devices.

1. **High Current Handling:**

   The IRL2203N is capable of handling a continuous drain current of around 116A with proper heatsinking, making it suitable for high-current applications.

2. **Low On-Resistance:**

   It has a low on-state resistance (Rds(on)) of around 7 mΩ (milliohms) at a gate-source voltage of 10V, which means it has very little voltage drop across it when it is turned on and allows it to operate efficiently with less heat generation.

3. **Fast Switching:**

   The IRL2203N is designed for fast switching speeds, which is beneficial in applications such as switching power supplies and Class D amplifiers.

4. **Voltage Rating:**

   It has a drain-to-source breakdown voltage (Vdss) of 30V, which means it can handle voltages up to 30V between its drain and source.

5. **TO-220 Package:**

   The IRL2203N typically comes in a TO-220 package, which is a widely used package type for power semiconductors. This package has good heat dissipation characteristics and is suitable for through-hole mounting.

The IRL2203N MOSFET is a versatile component that is used in a wide range of applications where efficient high-current switching is required.

Short Description Of IR2112:

The IR2112 is a high voltage, high-speed gate driver IC manufactured by Infineon Technologies (formerly International Rectifier). It is designed to drive both N-channel MOSFETs and IGBTs in various power electronics applications.

Here are some key features and specifications of the IR2112:

1. **Dual Outputs:**
   The IR2112 has dual outputs, one for driving a high-side MOSFET or IGBT and another for driving a low-side MOSFET or IGBT. This makes it ideal for half-bridge configurations.

2. **Bootstrap Operation:**
   It supports bootstrap operation, allowing it to drive the high-side switch without the need for an additional isolated power supply. This is achieved through an integrated bootstrap diode and an external capacitor.

3. **High Voltage Capability:**
   The IR2112 can handle high voltage levels on the VS pin (up to 600V in transient conditions), allowing it to be used in high voltage applications.

4. **Protection Features:**
   It includes under-voltage lockout (UVLO) protection for both the high-side and low-side drivers. This feature ensures that the MOSFETs or IGBTs are not driven unless the supply voltage is above a certain threshold.

5. **Fast Switching Speeds:**
   The IR2112 is designed for high-speed operation, with fast rise and fall times, making it suitable for high-frequency switching applications such as SMPS (Switch Mode Power Supplies) and Class D amplifiers.

6. **Logic Input Compatibility:**
   The input pins (HIN and LIN) are compatible with standard logic levels, making it easy to interface with microcontrollers or other logic devices.

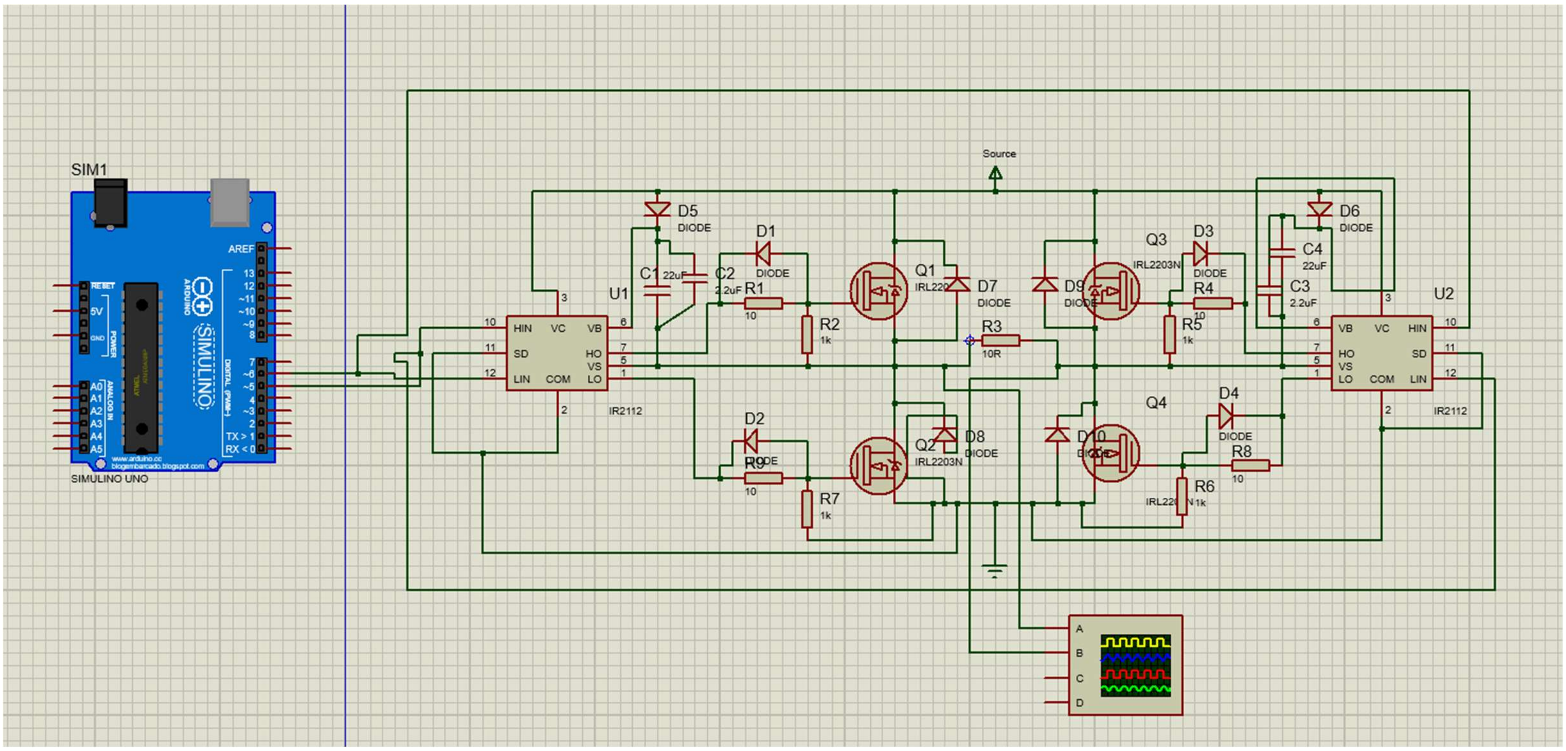
7. **Independent Inputs:**
   The IR2112 allows for independent control of the high-side and low-side drivers through separate input pins, giving more flexibility in control strategies, such as PWM signals with adjustable dead times.

8. **Shutdown Pin:**
   It includes a shutdown pin (SD) which, when driven high, turns off both the high-side and low-side outputs. This can be used for protection or soft-start purposes.

9. **Compact Package Options:**
   The IR2112 is available in compact packages such as DIP and SOIC, making it suitable for space-constrained applications.

In summary, the IR2112 is a versatile high-voltage gate driver IC widely used in motor control, inverters, and various other power electronics applications where efficient and fast switching of MOSFETs and IGBTs is required.

The Schematic and Perf Board:



The components used:

1. Gate Driver IC IR2112 *2
2. Boot Strap Capacitors 22uF *2
3. 10 Ohms Gate Resistance *2
4. 1k gate discharge resistance *4
5. IRL220N *4
6. Shotkey Diodes U4007 to avoid voltage fluctuations *10
7. Audrino UNO to generate PWM control Signal
8. 47uF capacitance to remove any noice from DC supply
9. 5V and 12V voltage Regulators

Code for PWM control signal for gate driver IC with a dead:

```
#define HIGH_SIDE_A 5  // High-side MOSFET on side A & Low-side B
#define HIGH_SIDE_B 6  // High-side MOSFET on side B & Low-side A

#define FREQUENCY 1 // 60 Hz
#define DUTY_CYCLE 50 // 50% duty cycle

unsigned long period;
unsigned long highStateDuration;
unsigned long lowStateDuration;
unsigned long lastChangeTime;
bool isHighState = true;

void setup() {
  pinMode(HIGH_SIDE_A, OUTPUT);
  pinMode(HIGH_SIDE_B, OUTPUT);


  period = 1000000 / FREQUENCY; // Period in microseconds
  highStateDuration = (period * DUTY_CYCLE) / 100; // High state duration
  lowStateDuration = highStateDuration; // Low state duration is equal to high
state duration
}

void loop() {
  unsigned long currentTime = micros();

  if (isHighState && currentTime - lastChangeTime >= highStateDuration) {
    // Switch to Low State
    digitalWrite(HIGH_SIDE_A, LOW);
    delayMicroseconds((period * 10) / 100); // 10% of period as dead time
    digitalWrite(HIGH_SIDE_B, HIGH);
    lastChangeTime = currentTime;
    isHighState = false;
  } else if (!isHighState && currentTime - lastChangeTime >= lowStateDuration) {
    // Switch to High State
    digitalWrite(HIGH_SIDE_B, LOW);
    delayMicroseconds((period * 10) / 100); // 10% of period as dead time
    digitalWrite(HIGH_SIDE_A, HIGH);
    lastChangeTime = currentTime;
    isHighState = true;
  }
}
```

This Arduino code is controls four MOSFETs connected to an electronic circuit. MOSFETs are like switches that can be used to control power in a circuit. The code switches them on and off in a particular pattern.

1. The code starts by defining names for pins to which the MOSFETs are connected. HIGH_SIDE_A and HIGH_SIDE_B are connected to pin numbers 5 and 6 on the Arduino respectively.

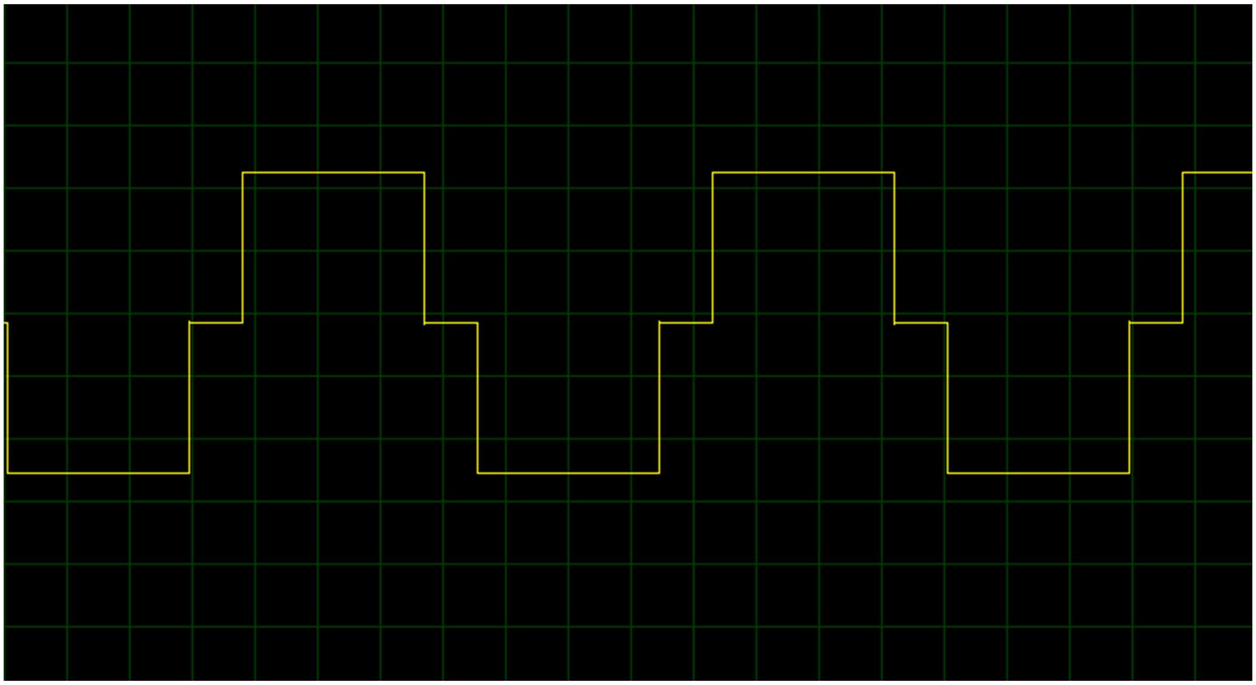
2. It also defines two constants, FREQUENCY and DUTY_CYCLE. FREQUENCY is set to 60, meaning that the switching will happen 60 times per second (60 Hz). DUTY_CYCLE is set to 50, meaning that each MOSFET will be turned on for 50% of the time.

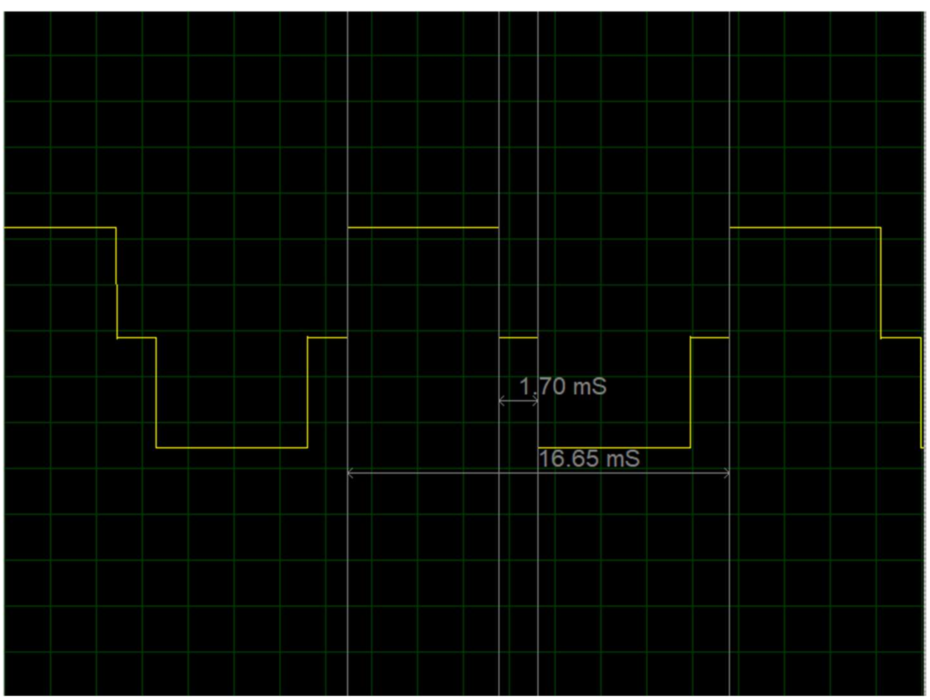
3. In the setup() function:
    a. The code sets the pins (5 and 6) as OUTPUTs, meaning that these pins will send out signals to the connected MOSFETs.
    b. It calculates the period, highStateDuration, and lowStateDuration in microseconds. The period is the total time for one on-and-off cycle. highStateDuration is the time for which the MOSFET stays in the ON state, and lowStateDuration is the time for which it stays in the OFF state (equal to highStateDuration in this case).

4. In the loop() function:
    a. The code constantly keeps track of the current time.
    b. It checks if it's time to change the state of the MOSFETs (based on highStateDuration and lowStateDuration).
    c. When it's time to switch from high to low state:
        i. It turns off HIGH_SIDE_A,
        ii. Waits for a short delay (10% of the period) called dead time to prevent both MOSFETs being ON at the same time,
        iii. And turns on HIGH_SIDE_B.
    d. When it's time to switch from low to high state:
        iv. It turns off HIGH_SIDE_B,
        v. Waits for the dead time,
        vi. And turns on HIGH_SIDE_A.

In summary, this code is used for controlling two MOSFETs by alternately turning them on and off at a frequency of 60Hz with a 50% duty cycle, and with a short dead time between the switching to ensure that they are never on at the same time.

Simulation Results:





Unfortunately, our hardware implementation didn't work properly as we might have damaged our **mosfets** due to excessive heat of soldering iron.