

	TESINA FINALE DI ROBOTICA	Classe 5RA
Data 31/05/24	Alunni: Villani Nicolò, De Giuli Filippo	A.S.: 2023/24

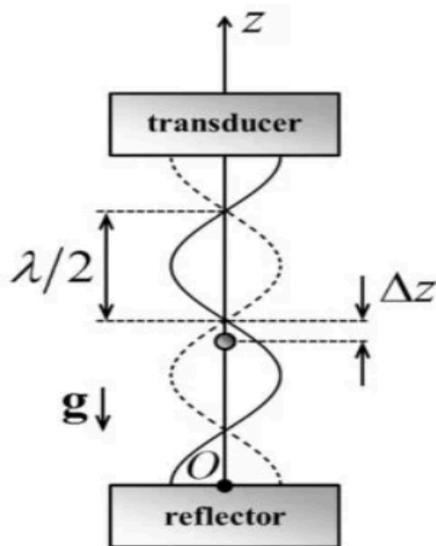
OGGETTO:	LEVITATORE AD ULTRASUONI
-----------------	---------------------------------

Introduzione
<p>L'obiettivo primario di questo progetto è quello di esaminare l'applicazione pratica dei principi fisici che stanno alla base del levitatore ad ultrasuoni, mentre si tiene sempre presente l'importanza di minimizzare i costi senza sacrificare la qualità e l'efficacia del risultato finale.</p> <p>All'interno di questa tesina, esploreremo in dettaglio il funzionamento del levitatore ad ultrasuoni, analizzando le sue applicazioni pratiche e il processo di progettazione che lo rende accessibile anche per progetti con risorse limitate. Abbiamo scelto di concentrare il nostro progetto su questo principio fisico perché riteniamo che sia molto efficiente, sostenibile, e soprattutto di forte impatto visivo.</p> <p style="text-align: center;">PUNTI DI FORZA DI QUESTO PROGETTO:</p> <ul style="list-style-type: none"> ● Facile da realizzare ● Bassi costi di realizzazione ● Impatto visivo non indifferente e di facile intuizione, che lo rende ideale per illustrare questo fenomeno scientifico

CHE COS'È LA LEVITAZIONE AD ULTRASUONI?

La levitazione ad ultrasuoni è un fenomeno fisico che sfrutta onde sonore ad alta frequenza per contrastare la forza di gravità e sollevare oggetti nel vuoto. Questo processo, che sembra quasi magico, è il risultato di una complessa interazione tra le onde sonore emesse e l'oggetto da far levitare.

Quando l'onda di ultrasuoni attraversa l'aria, l'onda sonora controbilancia la forza di gravità, con il risultato che gli oggetti possono librarsi senza sostegno nell'aria. Per far sì che questo avvenga è necessario avere un'onda stazionaria. Un'onda stazionaria si forma quando due onde identiche provenienti da direzioni opposte si sovrappongono l'una all'altra. Pertanto, in una configurazione di levitazione acustica si utilizza un trasduttore a ultrasuoni per creare onde di pressione longitudinali e un riflettore sull'altro lato riflette le onde, in modo che le onde identiche provenienti da entrambi i lati possano sovrapporsi e formare un'onda stazionaria.

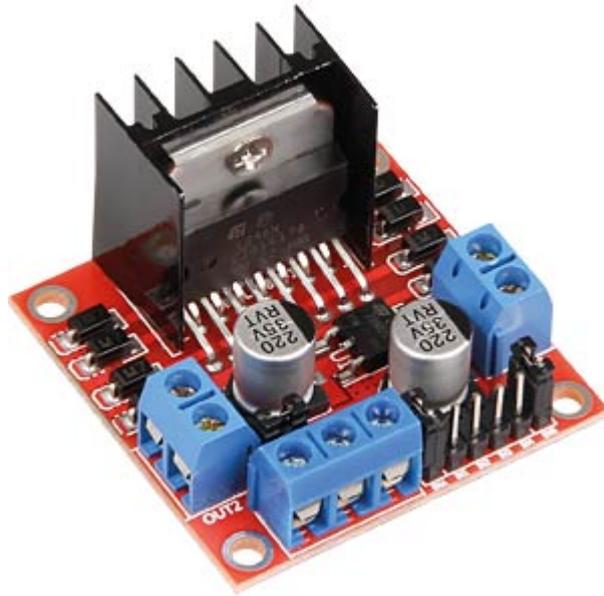


L'onda di pressione longitudinale degli ultrasuoni intensi permette la levitazione senza contatto con l'aria. Le onde stazionarie hanno **nodi** definiti. Un nodo è l'area di minima pressione, mentre un **anti-nodo** è definito come l'area di massima pressione. I nodi di un'onda stazionaria sono al centro della levitazione acustica

Questa tecnologia al giorno d'oggi trova applicazioni in moltissimi ambiti, che andremo a vedere nel dettaglio in seguito, tra cui la distruzione/estrazione delle cellule dalle alghe, il trattamento dei liquidi, e la manipolazione o il posizionamento di piccoli strati di materiale senza contatto.

COMPONENTI ELETTRONICI

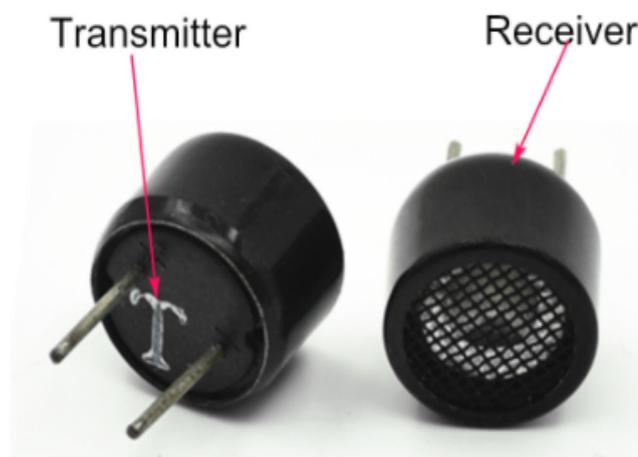
- **L297N:** L'L297N è un circuito integrato (IC) prodotto da STMicroelectronics, progettato specificamente per il controllo dei motori passo-passo e dei motori in corrente continua (DC). Nel nostro progetto viene utilizzato per il controllo della tensione nel circuito.



- **ARDUINO UNO R3:** L'Arduino UNO R3 è una scheda microcontrollore basata sul microcontrollore ATmega328P.



- **TRASMETTITORE AD ULTRASUONI:** Un trasmettitore ad ultrasuoni è un dispositivo che converte segnali elettrici in onde sonore ad alta frequenza al di là della capacità uditiva umana, generalmente oltre i 20 kHz. Il trasmettitore ad ultrasuoni inizia con la generazione di un segnale elettrico. Questo segnale può essere una forma d'onda sinusoidale o un segnale modulato in frequenza, a seconda delle esigenze dell'applicazione. Il segnale elettrico viene quindi inviato a un trasduttore piezoelettrico. Questo trasduttore converte l'energia elettrica in energia meccanica, creando onde sonore ad alta frequenza. Una volta attivato dal segnale elettrico, il trasduttore emette le onde sonore ad ultrasuoni nell'ambiente circostante. Queste onde viaggiano attraverso il mezzo di propagazione, come ad esempio l'aria o l'acqua, fino al destinatario o all'oggetto bersaglio. Le onde sonore ad ultrasuoni si propagano attraverso il mezzo e possono essere rilevate da un ricevitore posizionato nella vicinanza del trasmettitore. Questo ricevitore è in grado di captare le onde sonore e convertirle nuovamente in segnali elettrici rilevabili e interpretabili dall'elettronica di controllo o dall'utente finale.
- **RICEVITORE AD ULTRASUONI:** Un ricevitore ad ultrasuoni è un dispositivo che rileva e converte le onde sonore ad alta frequenza (ultrasuoni) in segnali elettrici interpretabili da un circuito elettronico o da un sistema di elaborazione dati. Il ricevitore ad ultrasuoni è costituito da un trasduttore piezoelettrico che funziona in modo opposto rispetto al trasmettitore. Quando le onde sonore ad ultrasuoni colpiscono il trasduttore, questo ne converte l'energia meccanica in segnali elettrici. Il trasduttore piezoelettrico produce una tensione elettrica proporzionale alla pressione delle onde sonore ad ultrasuoni che lo colpiscono. Questa tensione generata è quindi amplificata e processata da circuiti elettronici interni al ricevitore. I segnali elettrici prodotti dal trasduttore sono spesso deboli e possono essere influenzati da rumori ambientali. Pertanto, i circuiti interni al ricevitore filtrano il segnale per eliminare il rumore e lo amplificano per renderlo più robusto e interpretabile.



- **CONNETTORE PER L'ALIMENTAZIONE:** Serve per alimentare il circuito, si connette direttamente all'L297N e a uno switch.



- **SWITCH:** serve per aprire e chiudere il circuito e di conseguenza stabilire quando il levitatore sarà attivo o spento.



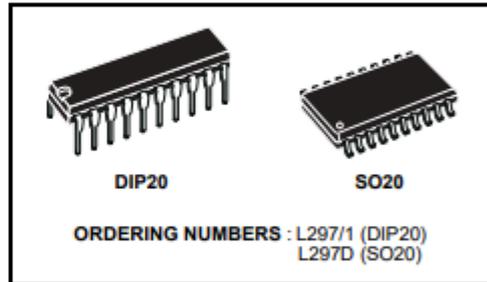
L297N



L297

STEPPER MOTOR CONTROLLERS

- NORMAL/WAVE DRIVE
- HALF/FULL STEP MODES
- CLOCKWISE/ANTICLOCKWISE DIRECTION
- SWITCHMODE LOAD CURRENT REGULATION
- PROGRAMMABLE LOAD CURRENT
- FEW EXTERNAL COMPONENTS
- RESET INPUT & HOME OUTPUT
- ENABLE INPUT



DESCRIPTION

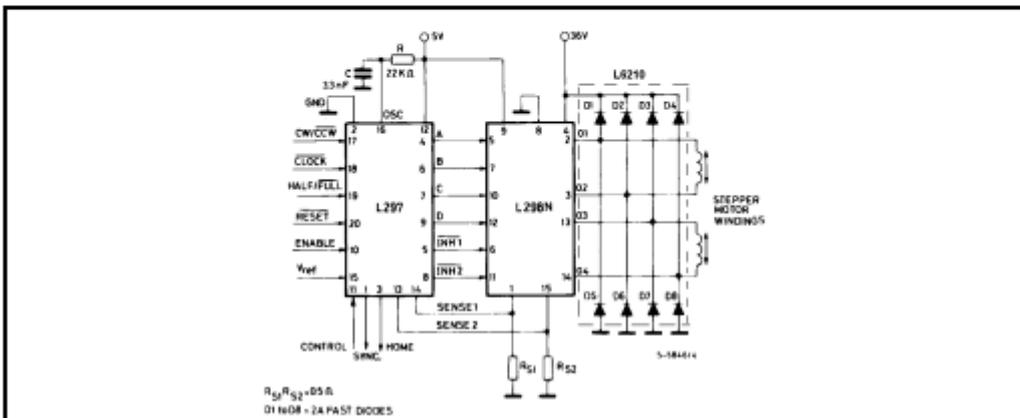
The L297 Stepper Motor Controller IC generates four phase drive signals for two phase bipolar and four phase unipolar step motors in microcomputer-controlled applications. The motor can be driven in half step, normal and wave drive modes and on-chip PWM chopper circuits permit switch-mode control of the current in the windings. A feature of

this device is that it requires only clock, direction and mode input signals. Since the phase are generated internally the burden on the microprocessor, and the programmer, is greatly reduced. Mounted in DIP20 and SO20 packages, the L297 can be used with monolithic bridge drives such as the L298N or L293E, or with discrete transistors and darlington.

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_s	Supply voltage	10	V
V_i	Input signals	7	V
P_{tot}	Total power dissipation ($T_{amb} = 70^{\circ}C$)	1	W
T_{stg}, T_j	Storage and junction temperature	-40 to + 150	$^{\circ}C$

TWO PHASE BIPOLAR STEPPER MOTOR CONTROL CIRCUIT



PIN FUNCTIONS - L297/1 - L297D

N°	NAME	FUNCTION
1	SYNC	Output of the on-chip chopper oscillator. The SYNC connections of all L297s to be synchronized are connected together and the oscillator components are omitted on all but one. If an external clock source is used it is injected at this terminal.
2	GND	Ground connection.
3	HOME	Open collector output that indicates when the L297 is in its initial state (ABCD = 0101). The transistor is open when this signal is active.
4	A	Motor phase A drive signal for power stage.
5	$\overline{\text{INH1}}$	Active low inhibit control for driver stage of A and B phases. When a bipolar bridge is used this signal can be used to ensure fast decay of load current when a winding is de-energized. Also used by chopper to regulate load current if CONTROL input is low.
6	B	Motor phase B drive signal for power stage.
7	C	Motor phase C drive signal for power stage.
8	$\overline{\text{INH2}}$	Active low inhibit control for drive stages of C and D phases. Same functions as INH1.
9	D	Motor phase D drive signal for power stage.
10	ENABLE	Chip enable input. When low (inactive) INH1, INH2, A, B, C and D are brought low.
11	CONTROL	Control input that defines action of chopper. When low chopper acts on INH1 and INH2; when high chopper acts on phase lines ABCD.
12	V_s	5V supply input.
13	SENS ₂	Input for load current sense voltage from power stages of phases C and D.
14	SENS ₁	Input for load current sense voltage from power stages of phases A and B.
15	V_{ref}	Reference voltage for chopper circuit. A voltage applied to this pin determines the peak load current.
16	OSC	An RC network (R to V_{CC} , C to ground) connected to this terminal determines the chopper rate. This terminal is connected to ground on all but one device in synchronized multi - L297 configurations. $f \approx 1/0.69 RC$
17	$\overline{\text{CW/CCW}}$	Clockwise/counterclockwise direction control input. Physical direction of motor rotation also depends on connection of windings. Synchronized internally therefore direction can be changed at any time.
18	$\overline{\text{CLOCK}}$	Step clock. An active low pulse on this input advances the motor one increment. The step occurs on the rising edge of this signal.

L297

PIN FUNCTIONS - L297/1 - L297D (continued)

N°	NAME	FUNCTION
19	$\overline{\text{HALF/FULL}}$	Half/full step select input. When high selects half step operation, when low selects full step operation. One-phase-on full step mode is obtained by selecting FULL when the L297's translator is at an even-numbered state. Two-phase-on full step mode is set by selecting FULL when the translator is at an odd numbered position. (The home position is designate state 1).
20	$\overline{\text{RESET}}$	Reset input. An active low pulse on this input restores the translator to the home position (state 1, ABCD = 0101).

THERMAL DATA

Symbol	Parameter		DIP20	SO20	Unit
$R_{th-j-amb}$	Thermal resistance junction-ambient	max	80	100	°C/W

CIRCUIT OPERATION

The L297 is intended for use with a dual bridge driver, quad darlington array or discrete power devices in step motor driving applications. It receives step clock, direction and mode signals from the systems controller (usually a microcomputer chip) and generates control signals for the power stage.

The principal functions are a translator, which generates the motor phase sequences, and a dual PWM chopper circuit which regulates the current in the motor windings. The translator generates three different sequences, selected by the HALF/FULL input. These are normal (two phases energised), wave drive (one phase energised) and half-step (alternately one phase energised/two phases energised). Two inhibit signals are also generated by the L297 in half step and wave drive modes. These signals, which connect directly to the L298's enable inputs, are intended to speed current decay when a winding is de-energised. When the L297 is used to drive a unipolar motor the chopper acts on these lines.

An input called CONTROL determines whether the chopper will act on the phase lines ABCD or the inhibit lines INH1 and INH2. When the phase lines

are chopped the non-active phase line of each pair (AB or CD) is activated (rather than interrupting the line then active). In L297 + L298 configurations this technique reduces dissipation in the load current sense resistors.

A common on-chip oscillator drives the dual chopper. It supplies pulses at the chopper rate which set the two flip-flops FF1 and FF2. When the current in a winding reaches the programmed peak value the voltage across the sense resistor (connected to one of the sense inputs SENS₁ or SENS₂) equals V_{ref} and the corresponding comparator resets its flip flop, interrupting the drive current until the next oscillator pulse arrives. The peak current for both windings is programmed by a voltage divider on the V_{ref} input.

Ground noise problems in multiple configurations can be avoided by synchronising the chopper oscillators. This is done by connecting all the SYNC pins together, mounting the oscillator RC network on one device only and grounding the OSC pin on all other devices.



Product Reference Manual
SKU: A000066



Description

The Arduino UNO R3 is the perfect board to get familiar with electronics and coding. This versatile development board is equipped with the well-known ATmega328P and the ATmega 16U2 Processor. This board will give you a great first experience within the world of Arduino.

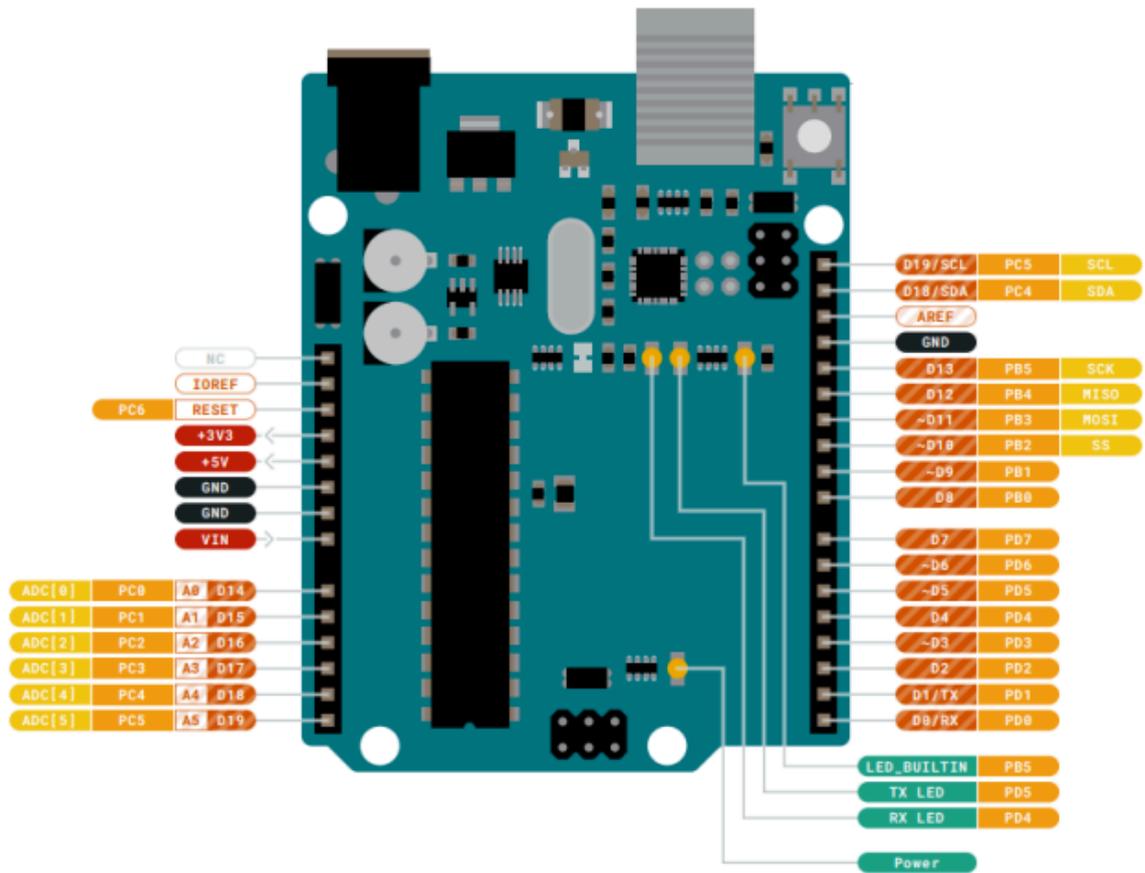
Target areas:

Maker, introduction, industries

Features

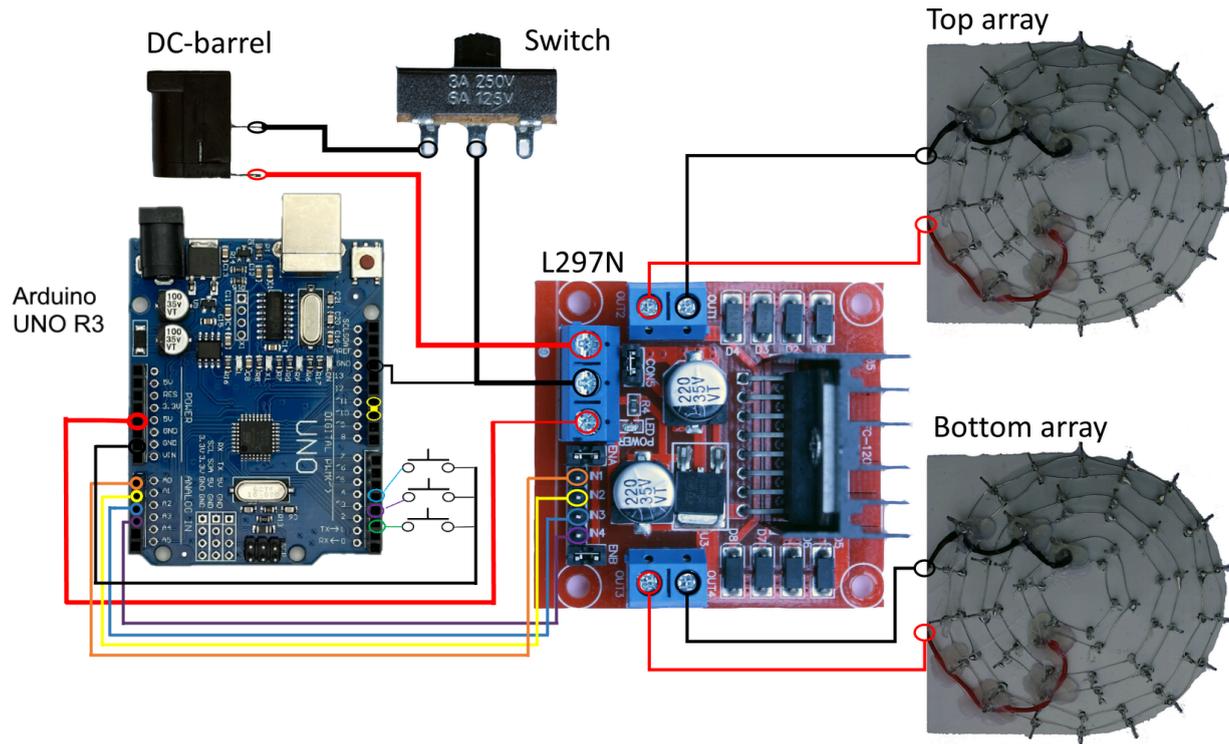
- **ATMega328P Processor**
 - **Memory**
 - AVR CPU at up to 16 MHz
 - 32KB Flash
 - 2KB SRAM
 - 1KB EEPROM
 - **Security**
 - Power On Reset (POR)
 - Brown Out Detection (BOD)
 - **Peripherals**
 - 2x 8-bit Timer/Counter with a dedicated period register and compare channels
 - 1x 16-bit Timer/Counter with a dedicated period register, input capture and compare channels
 - 1x USART with fractional baud rate generator and start-of-frame detection
 - 1x controller/peripheral Serial Peripheral Interface (SPI)
 - 1x Dual mode controller/peripheral I2C
 - 1x Analog Comparator (AC) with a scalable reference input
 - Watchdog Timer with separate on-chip oscillator
 - Six PWM channels
 - Interrupt and wake-up on pin change
- **ATMega16U2 Processor**
 - 8-bit AVR® RISC-based microcontroller
- **Memory**
 - 16 KB ISP Flash
 - 512B EEPROM
 - 512B SRAM
 - debugWIRE interface for on-chip debugging and programming
- **Power**
 - 2.7-5.5 volts

5 Connector Pinouts

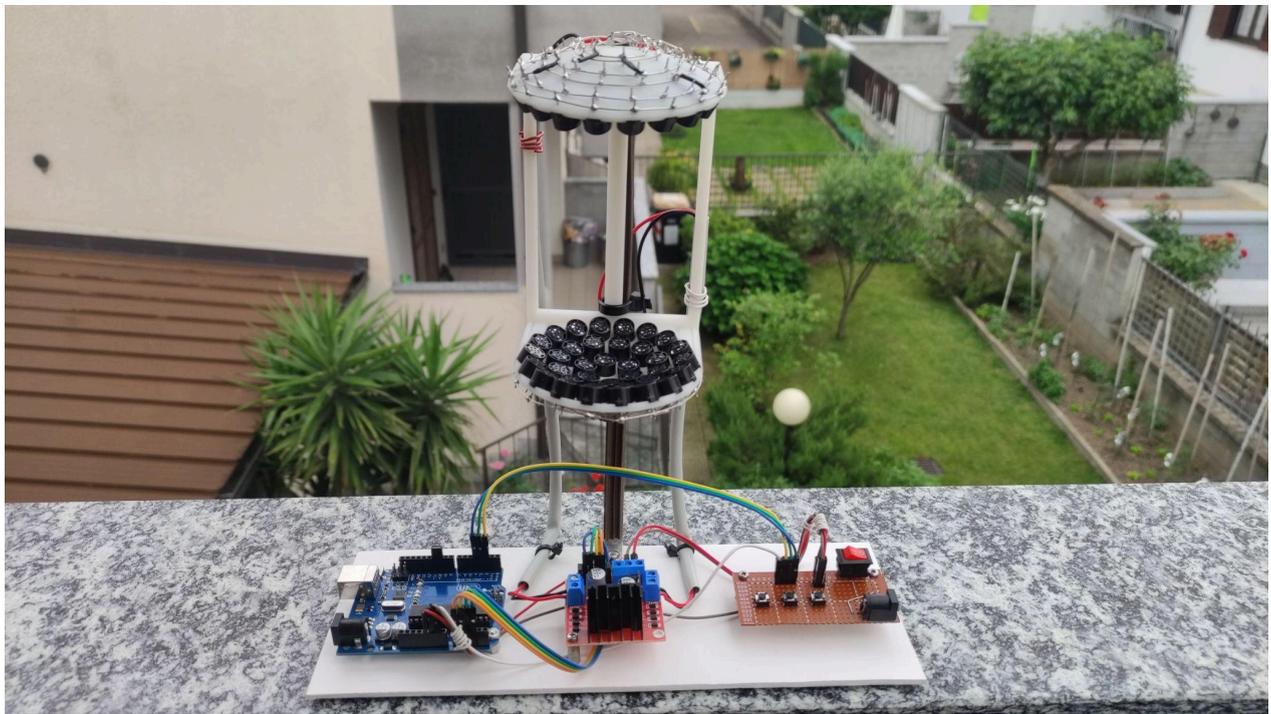


Pinout

SCHEMA ELETTRICO

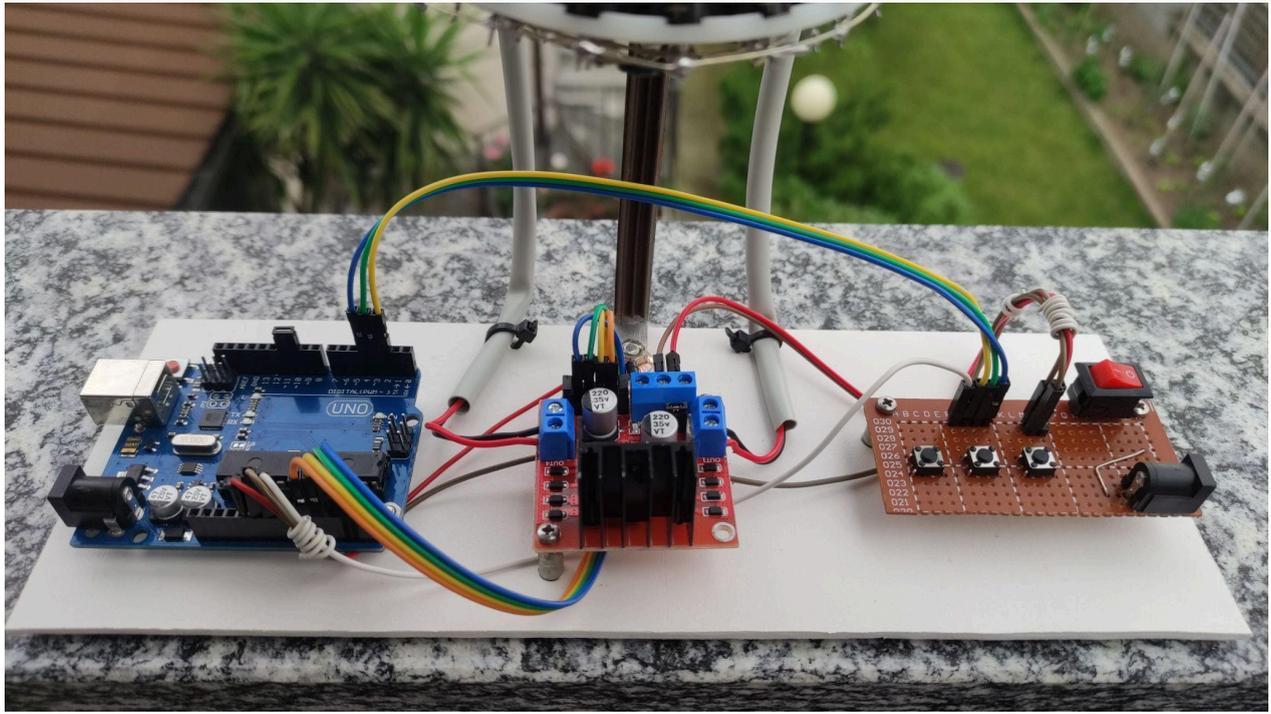


PROGETTO REALIZZATO





PROGETTO REALIZZATO



CODICE SU SCHEDA ARDUINO UNO R3

```
#include <avr/sleep.h>
#include <avr/power.h>

#define N_PORTS 1
#define N_DIVS 24

#define WAIT_LOT(a) __asm__ __volatile__ ("nop"); __asm__ __volatile__ ("nop"); __asm__
__volatile__ ("nop"); __asm__ __volatile__ ("nop"); __asm__ __volatile__ ("nop"); __asm__
__volatile__ ("nop"); __asm__ __volatile__ ("nop"); __asm__ __volatile__ ("nop"); __asm__
__volatile__ ("nop"); __asm__ __volatile__ ("nop"); __asm__ __volatile__ ("nop"); __asm__
__volatile__ ("nop"); __asm__ __volatile__ ("nop"); __asm__ __volatile__ ("nop");
#define WAIT_MID(a) __asm__ __volatile__ ("nop"); __asm__ __volatile__ ("nop"); __asm__
__volatile__ ("nop"); __asm__ __volatile__ ("nop"); __asm__ __volatile__ ("nop"); __asm__
__volatile__ ("nop"); __asm__ __volatile__ ("nop"); __asm__ __volatile__ ("nop"); __asm__
__volatile__ ("nop"); __asm__ __volatile__ ("nop"); __asm__ __volatile__ ("nop"); __asm__
__volatile__ ("nop"); __asm__ __volatile__ ("nop");
#define WAIT_LIT(a) __asm__ __volatile__ ("nop"); __asm__ __volatile__ ("nop"); __asm__
__volatile__ ("nop"); __asm__ __volatile__ ("nop"); __asm__ __volatile__ ("nop"); __asm__
__volatile__ ("nop"); __asm__ __volatile__ ("nop"); __asm__ __volatile__ ("nop"); __asm__
__volatile__ ("nop");

#define OUTPUT_WAVE(pointer, d) PORTC = pointer[d*N_PORTS + 0]

#define N_BUTTONS 6
//0.5 secondi
#define STEP_SIZE 1
#define BUTTON_SENS 2500
#define N_FRAMES 24
```

CODICE SU SCHEDA ARDUINO UNO R3

```
static byte frame = 0;
static byte animation[N_FRAMES][N_DIVS] =
{{0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa},
{0x9,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x6,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa},
{0x9,0x9,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x6,0x6,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa},
{0x9,0x9,0x9,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x6,0x6,0x6,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa},
{0x9,0x9,0x9,0x9,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x6,0x6,0x6,0x6,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa},
{0x9,0x9,0x9,0x9,0x9,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x6,0x6,0x6,0x6,0x6,0xa,0xa,0xa,0xa,0xa,0xa,0xa},
{0x9,0x9,0x9,0x9,0x9,0x9,0x5,0x5,0x5,0x5,0x5,0x5,0x6,0x6,0x6,0x6,0x6,0x6,0xa,0xa,0xa,0xa,0xa,0xa},
{0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x5,0x5,0x5,0x5,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0xa,0xa,0xa,0xa,0xa,0xa},
{0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x5,0x5,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0xa,0xa,0xa,0xa,0xa},
{0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x5,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0xa,0xa,0xa},
{0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x5,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0xa,0xa},
{0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x5,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0xa},
{0x5,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0xa,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6},
{0x5,0x5,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0xa,0xa,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6},
{0x5,0x5,0x5,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0x9,0xa,0xa,0xa,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6,0x6},
{0x5,0x5,0x5,0x5,0x5,0x9,0x9,0x9,0x9,0x9,0x9,0xa,0xa,0xa,0xa,0xa,0x6,0x6,0x6,0x6,0x6,0x6,0x6},
{0x5,0x5,0x5,0x5,0x5,0x5,0x9,0x9,0x9,0x9,0x9,0xa,0xa,0xa,0xa,0xa,0xa,0x6,0x6,0x6,0x6,0x6,0x6},
{0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x9,0x9,0x9,0x9,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0x6,0x6,0x6,0x6,0x6},
{0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x5,0x9,0x9,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0xa,0x6,0x6},
```


CODICE SU SCHEDA ARDUINO UNO R3

```
OUTPUT_WAVE(emittingPointer, 3); buttonPressed[1] = buttonsPort & 0b00001000;
WAIT_MID();
OUTPUT_WAVE(emittingPointer, 4); buttonPressed[2] = buttonsPort & 0b00010000;
WAIT_MID();
OUTPUT_WAVE(emittingPointer, 5); buttonPressed[3] = buttonsPort & 0b00100000;
WAIT_MID();
OUTPUT_WAVE(emittingPointer, 6); buttonPressed[4] = buttonsPort & 0b01000000;
WAIT_MID();
OUTPUT_WAVE(emittingPointer, 7); buttonPressed[5] = buttonsPort & 0b10000000;
WAIT_MID();
OUTPUT_WAVE(emittingPointer, 8); WAIT_LOT();
OUTPUT_WAVE(emittingPointer, 9); WAIT_LOT();
OUTPUT_WAVE(emittingPointer, 10); WAIT_LOT();
OUTPUT_WAVE(emittingPointer, 11); WAIT_LOT();
OUTPUT_WAVE(emittingPointer, 12); WAIT_LOT();
OUTPUT_WAVE(emittingPointer, 13); WAIT_LOT();
OUTPUT_WAVE(emittingPointer, 14); WAIT_LOT();
OUTPUT_WAVE(emittingPointer, 15); WAIT_LOT();
OUTPUT_WAVE(emittingPointer, 16); WAIT_LOT();
OUTPUT_WAVE(emittingPointer, 17); WAIT_LOT();
OUTPUT_WAVE(emittingPointer, 18); WAIT_LOT();
OUTPUT_WAVE(emittingPointer, 19); WAIT_LOT();
OUTPUT_WAVE(emittingPointer, 20); WAIT_LOT();
OUTPUT_WAVE(emittingPointer, 21); WAIT_LOT();
OUTPUT_WAVE(emittingPointer, 22); WAIT_LOT();
OUTPUT_WAVE(emittingPointer, 23);
```

```
if( anyButtonPressed ){
  ++buttonCounter;
  if (buttonCounter > BUTTON_SENS){
    buttonCounter = 0;
```

```
    if (! buttonPressed[0] ) {
      if( frame < STEP_SIZE ) {
        frame = N_FRAMES-1;
      }else{
        frame-=STEP_SIZE;
      }
    }
  }
```

```
    else if (! buttonPressed[1] ) {
```

```
+++++
```

```
    if ( frame >= N_FRAMES-STEP_SIZE ) {
      frame = 0;
    }else {
      frame+=STEP_SIZE;
    }
  }
```

```
    }else if (! buttonPressed[2] ) {
      frame = 0;
    }
  }
```

CODICE SU SCHEDA ARDUINO UNO R3

```
    emittingPointer = & animation[frame][0];
  }
}else {
  buttonCounter = 0;
}

goto LOOP;

}

void loop(){}
```

APPLICAZIONI DELLA LEVITAZIONE AD ULTRASUONI

In questa sezione esploreremo nel dettaglio le numerose applicazioni della levitazione ad ultrasuoni, suddividendole in categorie a breve e lungo termine. Analizzeremo come questa tecnologia sta già rivoluzionando alcuni settori e come potrebbe trasformare radicalmente altri in futuro. Dalla manipolazione di materiali delicati alla creazione di nuovi mezzi di trasporto, dalle innovazioni mediche alle tecnologie per la ricerca spaziale, la levitazione ad ultrasuoni si presenta come una delle tecnologie più promettenti del nostro tempo.

Manipolazione di Materiali Sensibili

- **Chimica e Biotecnologia:** La levitazione ad ultrasuoni può essere utilizzata per manipolare materiali sensibili senza contatto fisico, evitando contaminazioni. Questo è particolarmente utile nella manipolazione di campioni biologici o chimici delicati.
- **Fabbricazione di Cristalli:** Permette di far crescere cristalli senza contatto, migliorando la qualità e la purezza dei cristalli per applicazioni elettroniche o ottiche.

Ricerca e Sperimentazione

- **Microgravità:** La levitazione acustica può simulare condizioni di microgravità per esperimenti scientifici sulla Terra, facilitando la ricerca in ambito spaziale.
- **Analisi dei Materiali:** Utilizzata per studiare le proprietà fisiche e chimiche dei materiali in assenza di contatto con superfici solide.

Assemblaggio di Microcomponenti

- **Industria Elettronica:** Facilita l'assemblaggio di microcomponenti senza contatto, riducendo il rischio di danni e migliorando la precisione del posizionamento.

APPLICAZIONI FUTURE

Medicina

- Chirurgia Non Invasiva: Utilizzo di onde ultrasoniche per levitare e manovrare strumenti chirurgici all'interno del corpo senza incisioni, riducendo il trauma e accelerando i tempi di recupero.
- Terapia Fisica: Sviluppo di dispositivi per la levitazione di parti del corpo per terapie fisiche avanzate, migliorando la riabilitazione.

Robotica e Automazione

- Robot di Precisione: I robot dotati di tecnologia di levitazione acustica potrebbero manipolare oggetti con estrema precisione, utili in ambito medico, industriale e di ricerca.

Tecnologia Consumer

- Dispositivi Portatili: Sviluppo di gadget e dispositivi portatili che utilizzano levitazione acustica per funzionalità innovative, come speaker fluttuanti o strumenti di gioco avanzati.

TRENI A LEVITAZIONE AD ULTRASUONI

Treni a Levitazione Acustica: I treni a levitazione acustica potrebbero rappresentare una rivoluzione nel trasporto pubblico, offrendo un'alternativa ai treni a levitazione magnetica (maglev). Questi treni sfrutterebbero onde sonore ad alta intensità per mantenere il treno sollevato sopra i binari, riducendo l'attrito e migliorando l'efficienza energetica. Un grosso vantaggio è rappresentato dai costi inferiori: sarebbero infatti potenzialmente meno costosi dei treni maglev, poiché non richiedono costosi elettro magneti superconduttori e sistemi di raffreddamento complessi. Questa tecnologia non è ancora applicabile oggi giorno, tuttavia gli scienziati stanno lavorando a questo progetto per garantire in futuro questo tipo di trasporto.

