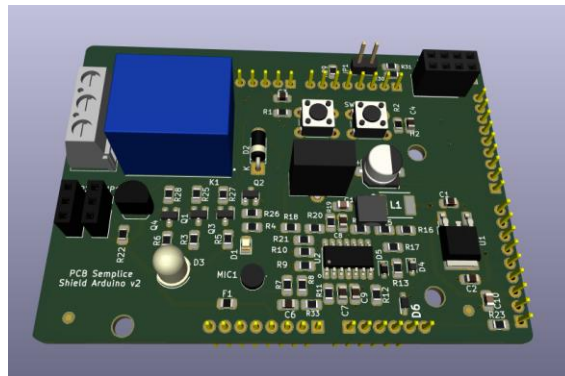


Documentazione Shield Arduino Uno v2



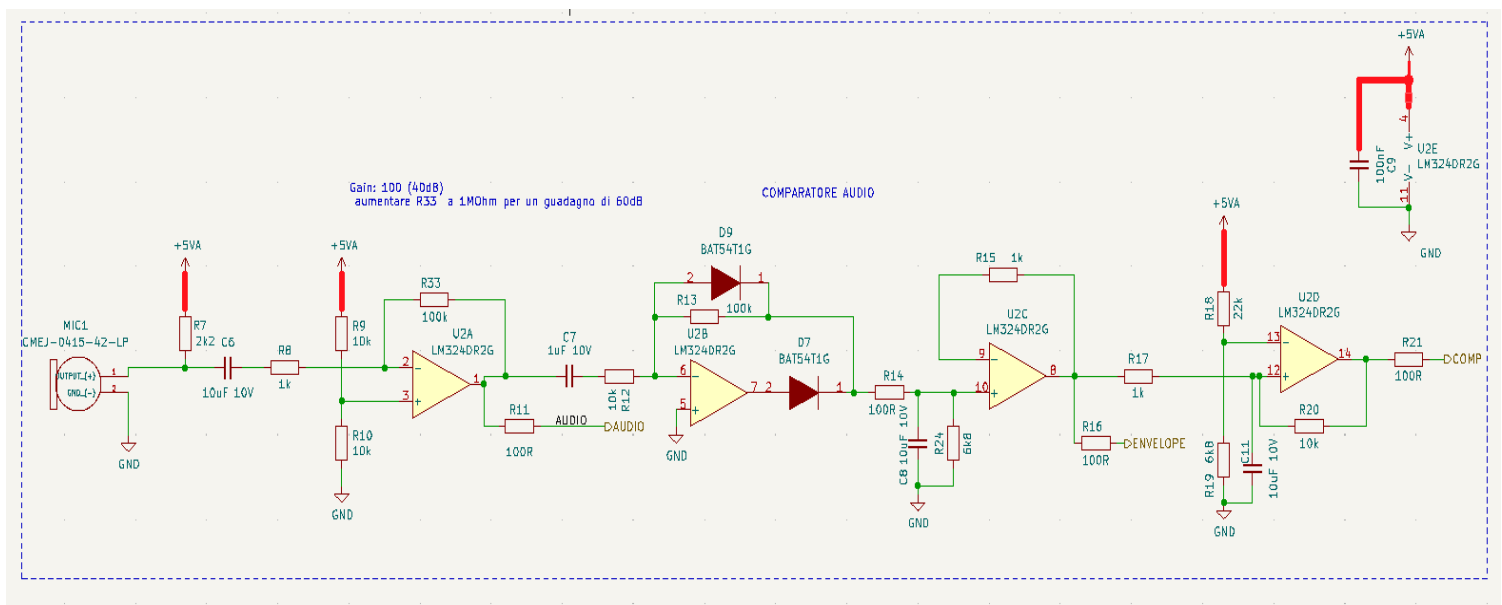
Descrizione generale

La scheda ospita queste sezioni

- E' presente uno stadio che implementa un interruttore audio
- La scheda contiene un trimmer collegato a un pin analogico della scheda Arduino
- Sono presenti due pulsanti
- La scheda contiene un LED RGB da 5mm
- E' presente un relè da 10 A
- La scheda è predisposta per ospitare il modulo Wi-Fi ESP8266
- E' presente un regolatore lineare che genera una tensione di 3.3V con una corrente massima di 1 A
- Sono presenti due header maschio che permettono allo shield di essere collegato a una breadboard

Interruttore audio

Lo stadio si basa sull'amplificatore operazionale a 4 canali LM324



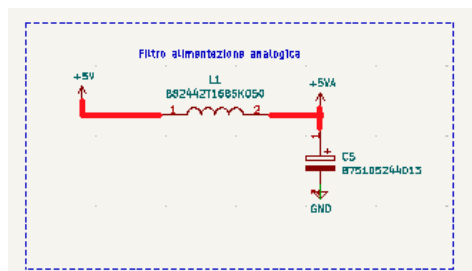
La prima sezione (U2A) dell'operazionale realizza un amplificatore invertente. Il segnale del microfono viene amplificato e il guadagno è determinato sostanzialmente dal resistore R33

L'uscita viene inviata sia a un canale analogico della scheda Arduino Uno, sia al secondo stadio (U2B) che realizza un raddrizzatore a singola semionda (half wave rectifier).

L'uscita del raddrizzatore viene inviata al terzo stadio (U2C) che costituisce un buffer con un filtro passa-basso.

L'uscita di quest'ultimo stadio viene inviata sia a un pin analogico della scheda Arduino Uno, sia al quarto stadio (U2D) che è un trigger di Schmitt. L'uscita (COMP) di quest'ultimo stadio viene inviata a un pin digitale della Arduino Uno.

Per evitare commutazioni spurie, la tensione di alimentazione (+5VA) dell'operazionale è ottenuta dalla tensione +5V attraverso un filtro costituito dall'induttore L1 e dal condensatore C5



Lo sketch caricato nel microcontrollore AtMEGA328P della scheda Arduino Uno, attiva un relè quando viene pronunciato per due volte il suono "a". Il relè viene disattivato quando viene pronunciato tre volte il suono "a"

```
const byte ledPin = 12;
const byte rele = 4;
const byte interruptPin = 2;
volatile byte state = LOW;
unsigned char count = 0;
unsigned char ON = 0;
unsigned char OFF = 0;
unsigned long timeComparison = 0;
unsigned char alreadyON = 0;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(interruptPin, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(interruptPin), checkComparator, RISING);
  Serial.begin(9600);
}

void loop() {
  Serial.println(count);
  if (ON==1 && alreadyON==0)
  {
    digitalWrite(ledPin, HIGH);
    digitalWrite(rele, HIGH);
    alreadyON=1;
    count=0;
    Serial.println("ON");
  }
}
```

```

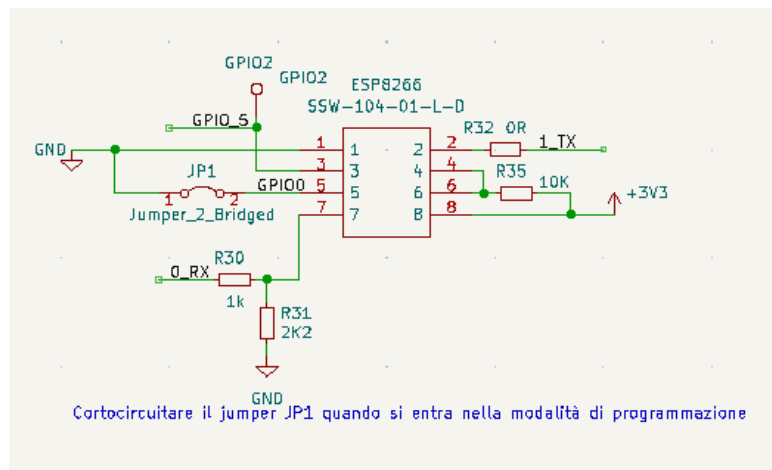
if (OFF==1){
  digitalWrite(ledPin, LOW);
  digitalWrite(rele, LOW);
  Serial.println("OFF");
  alreadyON =0;
  count=0;
  OFF=0;
}
}

void checkComparator() {
  count++;
  switch (count){
    case 1:
      timeComparison=millis();
      break;
    case 2:
      timeComparison=(millis())-timeComparison;
      if (timeComparison <2000)
      {
        ON=1;
        OFF=0;
        timeComparison=millis();
      }
      else{
        count=0;
      }
      break;
    case 3:
      timeComparison=(millis())-timeComparison;
      if (timeComparison <2000)
      {
        ON=0;
        OFF=1;
        count =0;
      }
      else{
        count=0;
      }
      break;
    default:
      count=0;
      break;
  }
}

```

Predisposizione per collegare un modulo ESP8266

E' possibile interfacciarsi con un modulo Wi-Fi ESP8266 montando il componente in corrispondenza del connettore J4. Orientare il modulo in modo che sia tutto all'interno dello shield.



Prima di programmare il modulo, caricare lo sketch qui sotto nel microcontrollore ATMEGA328P della scheda Arduino Uno in modo che non si creino conflitti a livello di comunicazione seriale (pin 0 e pin 1 della scheda Arduino UNO) Lo sketch permette, inoltre di comandare il relè in base allo stato del pin GPIO5.

```
const int EspOutput = 5;    // ESP8266 output
const int rele = 4;        // the number of the LED pin
const int led= 13;
void setup() {
  pinMode(EspOutput, INPUT);
  pinMode(rele, OUTPUT);
  pinMode(led, OUTPUT);
  // put your setup code here, to run once:
}

void loop() {
  if (digitalRead(EspOutput))
  {
    delay(500);
    if (digitalRead(EspOutput))
    {
      digitalWrite(rele,HIGH);
      digitalWrite(led,HIGH);
    }
    else
    {
      digitalWrite(led,LOW);
      digitalWrite(rele,LOW);
    }
  }
}
```

La resistenza R29 permette di selezionare se collegare un'uscita digitale del modulo al pin GPIO5 della scheda Arduino Uno. In questo caso il pin della scheda Arduino dovrà essere configurato come ingresso.

Per l'impostazione del modulo, fare riferimento al video (E' in inglese)

<https://www.youtube.com/watch?v=AvSCAxbqvE>.

Prima di programmare il modulo Wi-Fi, togliere alimentazione allo shield e inserire un jumper in corrispondenza del connettore JP1. A questo punto fornire nuovamente l'alimentazione e programmare il microcontrollore usando lo sketch "Blynk_NodeMCU_ESP8266" selezionando come board "Generic ESP8266 module". E' fondamentale sostituire i parametri BLYNK_TEMPLATE_ID e BLYNK_DEVICE_NAME con i valori forniti quando si crea l'applicazione attraverso il portale <https://blynk.cloud/>