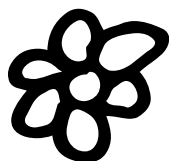


# LA MÁQUINA DE KLAUSNER



## MANUAL DEL USUARIO

RONI BANDINI

## IMPORTANTE

La presente guía es otorgada a efectos informativos.

Existe la posibilidad de que ciertas funciones o especificaciones del dispositivo se vean modificadas desde la fecha de publicación. El inventor no se hace responsable por el uso, la interpretación o la exactitud de la información presentada. El software se otorga bajo licencia Creative Commons Attribution. Ante cualquier duda consultar a Roni Bandini @RoniBandini

.....

# INTRODUCCIÓN

En La Máquina del Sonido, Roald Dahl cuenta la historia de un inventor que fabrica una máquina para escuchar a las plantas:

"Se colocó los auriculares y conectó la máquina. Durante un momento escuchó el débil y familiar zumbido; después levantó el hacha, tomó impulso con las piernas abiertas, y la clavó con tanta fuerza como le fue posible en la base del tronco del árbol. La hoja penetró profundamente en la madera y se quedó allí. En el momento mismo del impacto, a través de los auriculares oyó un ruido extraordinario. Era un ruido nuevo, distinto – un bronco, inarmónico e intenso ruido, un sonido sordo, grave, quejumbroso; no corto y rápido como el de las rosas, sino prolongado durante casi un minuto, más fuerte en el instante en que clavó el hacha, y debilitándose

gradualmente hasta desaparecer."

La Máquina de Klausner utiliza la ficción de Roald Dahl como disparador. Para la fabricación del dispositivo fueron empleados circuitos electrónicos, código de programación y modelado 3D.

El carácter performático, arbitrario y lúdico de la Máquina de Klausner, no va en perjuicio de ciertas revelaciones que suceden con el uso. A partir de un punto, parece surgir la conciencia de que ninguna forma de vida permanece en silencio.

## MATERIALES

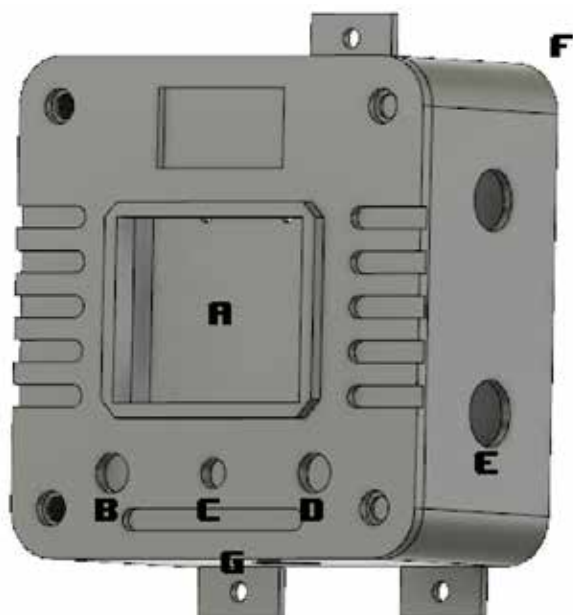
- Placa Arduino Nano
- Sensor de humedad en tierra FC28
- Un panel de vúmetros analógicos

- Un parlante de 4 ohms
- Una placa DFPlayer
- Un botón
- Un switch
- Un led
- Un potenciómetro
- Una tarjeta microSD
- Un gabinete a medida

## **PARTES**

- A. Panel de sensibilidad y lectura de humedad
- B. Switch de encendido

- C. Led de operación
- D. Regulador de sensibilidad
- E. Ficha de alimentación 5V
- F. Sensor de humedad en tierra
- G. Botón de medición



# OPERACIÓN

Tras encender el switch la máquina realizará un control interno, para lo cual encenderá el led de operación, moverá las agujas del vúmetro analógico y presentará un alerta de audio. Finalmente quedará lista para su utilización.

Para “escuchar” plantas es preciso clavar el sensor de humedad en tierra, regular la sensibilidad y presionar el botón de medición.

# CIRCUITOS ELECTRÓNICOS

- El Sensor de humedad a 3.3v, Ground y A0
- DFPlayer 3.3v y GND a alimentación del Arduino. SPK1 y SPK2 al parlante de 4ohms, RX al pin 10 y TX al pin 11.

- Los vúmetros analógicos a GND y D5 y GND y D6
- Led al D9 y Ground
- Potenciómetro a 3.3v, GND y D11.
- El botón a Ground y D2
- El switch se conecta interrumpiendo el paso de 5V al pin VIN.

## TROUBLESHOOTING

A. La máquina no enciende: verifique que ambos extremos del cable de alimentación se encuentren bien conectados

B. La máquina no emite sonidos: verifique que la tarjeta microSD se encuentre bien insertada en la placa DFPlayer que se encuentra adentro del gabinete. Verifique también los contactos del speaker.

C. Las mediciones no son correspondientes



con los sonidos: realice nuevamente el procedimiento de calibración detallada en la sección de operación de la máquina.

## CÓDIGO DE PROGRAMACIÓN

```
/*  
Klausner, The Sound Machine 2  
Based on Roald Dahl short story  
Creative Commons Attribution License  
Roni Bandini, October 2020, Buenos Aires, Argentina  
Twitter/Instagram @RoniBandini  
*/  
  
#include "SoftwareSerial.h"  
  
// Mp3 player rx and tx  
SoftwareSerial mySerial(10, 11);  
  
# define Start_Byte 0x7E  
# define Version_Byte 0xFF  
# define Command_Length 0x06  
# define End_Byte 0xEF  
# define Acknowledge 0x00  
# define ACTIVATED LOW  
  
int sensorPin = 7;  
int vumeterPin1 = 5;  
int vumeterPin2 = 6;
```

```
int potPin = 1;
int butPin = 2;
int ledPin = 9;

int humidValue=0;
long randNumber;
int sensValue=0;

// init array
int arrayCounter=0;
int arrayUsed[25] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

void setup(){

    digitalWrite(ledPin , HIGH);

    Serial.begin(9600);
    Serial.println("Klausner started");

    // nothing being played, just init
    playFirst();

    // Play welcome msg
    execute_CMD(0x0F,0x01,0x1F);

    // humidity sensor
    pinMode(sensorPin, INPUT);
    // button
    pinMode(butPin, INPUT_PULLUP);
    // led
    pinMode(sensorPin, OUTPUT);
```

```
// mp3 player
mySerial.begin(9600);

// test vumeters
int myV=0;
while (myV<20)
{
  analogWrite(vumeterPin1, myV);
  analogWrite(vumeterPin2, myV);
  delay(50);
  myV=myV+1;
}

analogWrite(vumeterPin1, 0);
analogWrite(vumeterPin2, 0);

delay (1000);

sensValue = map(analogRead(potPin), 0, 1023, 0, 20);

Serial.print("Sensibility: ");
Serial.print(sensValue);
Serial.println("%");

// Write sensibility into vumeter 1
analogWrite(vumeterPin1, sensValue);

}

void loop(){

  int butValue = digitalRead(butPin);
```

```
if (butValue==0){

    // button pressed

    digitalWrite(ledPin , LOW);

    humidValue = map(analogRead(sensorPin),
0, 1023, 100, 0);

    Serial.print("Humidity: ");
    Serial.print(humidValue);
    Serial.println("%");

    int humidValueVumeter =
map(analogRead(sensorPin), 0, 1023, 20, 0);

    // Write humid into vumeter 2
    analogWrite(vumeterPin2, humidValueVumeter);

    float sensFactor =
map(analogRead(potPin), 0, 1023, 0, 1);
    Serial.print("Sens factor: ");
    Serial.print(sensFactor);

    if (humidValue*sensFactor<40){
        Serial.println("Scream:");

        int mySound=selectSound(1, 15);

        switch (mySound) {
            case 1:
                execute_CMD(0x0F,0x01,0x01);
```

```
break;
case 2:
    execute_CMD(0x0F,0x01,0x02);
    break;
case 3:
    execute_CMD(0x0F,0x01,0x03);
    break;
case 4:
    execute_CMD(0x0F,0x01,0x04);
    break;
case 5:
    execute_CMD(0x0F,0x01,0x05);
    break;
case 6:
    execute_CMD(0x0F,0x01,0x06);
    break;
case 7:
    execute_CMD(0x0F,0x01,0x07);
    break;
case 8:
    execute_CMD(0x0F,0x01,0x08);
    break;
case 9:
    execute_CMD(0x0F,0x01,0x09);
    break;
case 10:
    execute_CMD(0x0F,0x01,0x0A);
    break;
case 11:
    execute_CMD(0x0F,0x01,0x0B);
    break;
case 12:
    execute_CMD(0x0F,0x01,0x0C);
    break;
```

```
    case 13:
        execute_CMD(0x0F,0x01,0x0D);
        break;
    case 14:
        execute_CMD(0x0F,0x01,0x0E);
        break;
    case 15:
        execute_CMD(0x0F,0x01,0x0F);
        break;
}
```

```
}
```

```
    if (humidValue*sensFactor>39 and
        humidValue*sensFactor<60){
        Serial.println("Middle:");
        int mySound=selectSound(16, 18);
        switch (mySound) {
            case 16:
                execute_CMD(0x0F,0x01,0x10);
                break;
            case 17:
                execute_CMD(0x0F,0x01,0x11);
                break;
            case 18:
                execute_CMD(0x0F,0x01,0x12);
                break;
        }
    }
}
```

```
if (humidValue*sensFactor>61){
    Serial.println("Happy:");
    int mySound=selectSound(19, 24);
```

```

Serial.println("My Sound is " + String(mySound));
switch (mySound) {
  case 19:
    execute_CMD(0x0F,0x01,0x13);
    break;
  case 20:
    execute_CMD(0x0F,0x01,0x14);
    break;
  case 21:
    execute_CMD(0x0F,0x01,0x15);
    break;
  case 22:
    execute_CMD(0x0F,0x01,0x16);
    break;
  case 23:
    execute_CMD(0x0F,0x01,0x17);
    break;
  case 24:
    execute_CMD(0x0F,0x01,0x18);
    break;
  case 25:
    execute_CMD(0x0F,0x01,0x19);
    break;
  case 26:
    execute_CMD(0x0F,0x01,0x1A);
    break;
  case 27:
    execute_CMD(0x0F,0x01,0x1B);
    break;
}
} // humid>61

delay(2500);

```

```
digitalWrite(ledPin , HIGH);

} // but pressed

delay(250);

// reset vumeter 2
analogWrite(vumeterPin2, 0);

sensValue = map(analogRead(potPin), 0, 1023, 0, 20);
Serial.print("Sensibility: ");
Serial.print(sensValue);
Serial.println("%");

// Write sensibility into vumeter 1
analogWrite(vumeterPin1, sensValue);

}

void playFirst()
{
  execute_CMD(0x3F, 0, 0);
  delay(500);
  setVolume(25);
  delay(500);
}

void pause()
{
  execute_CMD(0x0E,0,0);
  delay(500);
}
```



```
}
```

```
void play()
{
    execute_CMD(0x0D,0,1);
    delay(500);
}
```

```
void playNext()
{
    execute_CMD(0x01,0,1);
    delay(500);
}
```

```
void playPrevious()
{
    execute_CMD(0x02,0,1);
    delay(500);
}
```

```
void setVolume(int volume)
{
    execute_CMD(0x06, 0, volume); //
    Set the volume (0x00~0x30)
    delay(2000);
}
```

```
void execute_CMD(byte CMD, byte Par1, byte Par2)
// Execute the command and parameters
{
    // Calculate the checksum (2 bytes)
    word checksum = -(Version_Byte + Command_
```

```

Length + CMD + Acknowledge + Par1 + Par2);
    // Build the command line
    byte Command_line[10] = { Start_Byte, Version_
Byte, Command_Length, CMD, Acknowledge,
    Par1, Par2, highByte(checksum),
lowByte(checksum), End_Byte};
    //Send the command line to the module
    for (byte k=0; k<10; k++)
    {
        mySerial.write( Command_line[k]);
    }
}

int selectSound(int lowerLimit, int upperLimit)
{
    Serial.println("Selecting sound from
:"+String(lowerLimit)+ " to " + String(upperLimit));

    // check if all were assigned
    int oneEmpty=0;
    for (byte i = lowerLimit; i < upperLimit; i = i + 1) {
        if (arrayUsed[i]==0){
            oneEmpty=1;
        }
    }
} // for

if (oneEmpty==0){
    // no empty spots, clean all
    Serial.println("Cleaning array");
    for (byte i = lowerLimit; i < upperLimit; i = i + 1) {
        arrayUsed[i]=0;
    }
} // if

```

```
int foundSpot=0;

while (foundSpot==0) {

    randNumber = random(lowerLimit, upperLimit);
    //Serial.println("Trying "+ String(randNumber));
    if (arrayUsed[randNumber]==0) {
        arrayUsed[randNumber]=1;
        //Serial.println("OK");
        foundSpot=1;
    }

}

return randNumber;

} // end loop
```

## AUDIOS

Los archivos de audio deben colocarse dentro de la carpeta 01/ y sus nombres de archivo deben partir desde 001.mp3 en adelante. Desde el código de programación, los comandos que llaman a los archivos son ejecutados con referencias hexadecimales. Por lo tanto, para llamar al archivo 016.mp3 se utiliza 0x10

```
execute_CMD(0x0F,0x01,0x10);
```

Y para llamar al 010.mp3 0x0A

```
execute_CMD(0x0F,0x01,0x0A);
```

## OTROS PROYECTOS

La máquina de Klausner es un invento más de un proyecto de investigación sobre la combinación de tecnología y literatura, entre los cuales se encuentran:

- Rayelomatic: la máquina para leer Rayuela
- Borgy: el androide de Jorge Luis Borges
- Expendedor de literatura: expendedor de ficciones cortas en tickets
- Mind Poetry: generación de poesía con lectura de ondas cerebrales

- Booksound: generación de música utilizando páginas de novelas

## CONTACTO

Roni Bandini

Twitter @RoniBandini

Instagram @RoniBandini

<https://medium.com/@Bandini>

.....