

Laboratório 6

SENSORES E ATUADORES

REALIZADO POR:

RODRIGO COELHO (A50251)

TATIANA DAMAYA (A50299)

ESTRUTURA

1. MEDIDOR DE DISTÂNCIAS (SONAR)

Protótipo: void distance(int pinTrig, int pinEcho, int rate=10)

2. IMPRESSORA PARA CONSOLA (SÓ VALORES NUMÉRICOS, FORMATO CSV, UM VALOR POR COLUNA):

Protótipo: void printCSV(int rate=2, float val1, float val2, ...,)

3. RECETOR NUMEROS INTEIROS (PORTA SERIAL)

Protótipo: void receiveInt()

4. TOCAR MELODIA (PIEZO)

Protótipo: void melodia(int duration, int nota1, int nota2, ...);

5. DETETOR TOQUE (BOTÃO)

Protótipo: void toque(int pinBotao)

6. CONTADOR (BOTÃO)

Protótipo: void contador(int pin)

7. GERADOR SINAIS (ONDA SINUSOIDAL (S), OU TRIANGULAR (T), OU QUADRADA(Q))

Protótipo: void gerador(int rate, int período, char tipoOnda)

1. Medidor de distâncias (sonar)

PROTÓTIPO: VOID
DISTANCE(INT PINTRIG, INT
PINECHO, INT RATE=10)

```
void distance(int pinTrig, int pinEcho, int rate) {  
    // Período de 1 segundo em micros por o rate desejado  
    unsigned int T = 1000000/rate;  
    static unsigned long t0 = micros();  
    static const int TRIGSTART = 0, TRIGSTOP = 1, ECHOSTART = 2, ECHOSTOP = 3;  
    static int state = TRIGSTART;  
    switch(state){  
        // Lançar o sinal do trigger  
        case TRIGSTART:  
            digitalWrite(pinTrig, HIGH);  
            t0 = micros();  
            state = TRIGSTOP;  
            break;  
        // Parar o sinal do trigger  
        case TRIGSTOP:  
            if(micros() - t0 > T) {  
                digitalWrite(pinTrig, LOW);  
                state = ECHOSTART;  
            }  
            break;  
        // Ligar o canal echo  
        case ECHOSTART:  
            if(digitalRead(pinEcho) == HIGH) {  
                t0 = micros();  
                state = ECHOSTOP;  
            }  
            break;  
        // Receber o valor do sinal lançado do trigger  
        case ECHOSTOP:  
            if(digitalRead(pinEcho) == LOW) {  
                int d = (micros() - t0)/58.0;  
                Serial.println(d);  
                state = TRIGSTART;  
            }  
            break;  
    }  
}
```

2. Impressora para consola (só valores numéricos, formato CSV, um valor por coluna):

2. IMPRESSORA PARA CONSOLA
(SÓ VALORES NUMÉRICOS,
FORMATO CSV, UM VALOR POR
COLUNA):

```
void printCSV(int rate, float val1, float val2) {
    unsigned int T = 1000 / rate;
    static unsigned long t0 = millis();
    static const int WAIT = 0, SEND = 1;
    static int state = WAIT;
    switch (state) {
        // Espera pelo os valores pedidos
        case WAIT:
            if(millis() - t0 > T) {
                state = SEND;
                t0 = millis();
            }
            break;
        // Introduce na consola os valores pedidos
        case SEND:
            Serial.println((String)val1 + "\t" + (String)val2);
            state = WAIT;
            break;
    }
}
```


3. Recetor numeros inteiros (porta serial)

PROTÓTIPO: VOID
RECEIVEINT()

```
static int val = 0;
void receiveInt() {
    static const int WAIT = 0, READ = 1;
    static int state = WAIT;

    switch (state) {
        //ESPERAR PELO INPUT NA CONSOLA
        case WAIT:
            if (Serial.available() > 0) {
                state = READ;
            }
            break;
        //ATRIBUIR VALOR DA CONSOLA
        case READ:
            val = Serial.parseInt();
            Serial.println("Escolheu a " + (String)val + "ª oitava.");
            state = WAIT;
            break;
    }
}
```

4. Tocar melodia (piezo)

PROTÓTIPO: VOID
MELODIA(INT DURATION, INT
NOTA1, INT NOTA2, ...);

```
void melodia(int duration, int nota1, int nota2) {  
    unsigned int T = duration;  
    static unsigned long t0 = millis();  
    static const int NOTA1 = 0, NOTA2 = 1;  
    static int state = NOTA1;  
    switch (state) {  
        // Iniciar o som da nota1  
        case NOTA1:  
            tone(PIEZO, nota1);  
            if(millis() - t0 > T) {  
                // Parar o som da nota1  
                noTone(PIEZO);  
                state = NOTA2;  
                t0 = millis();  
            }  
            break;  
        // Iniciar o som da nota2  
        case NOTA2:  
            tone(PIEZO, nota2);  
            if(millis() - t0 > T) {  
                // Parar o som da nota2  
                noTone(PIEZO);  
                state = NOTA1;  
                t0 = millis();  
            }  
            break;  
    }  
}
```

5. Detetor toque (botão)

PROTÓTIPO: VOID TOQUE(INT PINBOTAO)

```
void toque(int pinBotao) {  
    int buttonPressed = 0;  
    static const int START = 0, STOP = 1;  
    static int state = START;  
  
    switch (state) {  
        case START:  
            buttonPressed = digitalRead(pinBotao);  
            if(buttonPressed == LOW) {  
                state = STOP;  
                Serial.println("STOP");  
            }  
            break;  
        case STOP:  
            buttonPressed = digitalRead(pinBotao);  
            if(buttonPressed == HIGH) {  
                state = START;  
                Serial.println("START");  
            }  
            break;  
    }  
}
```

6. Contador (botão)

PROTÓTIPO: VOID
CONTADOR(INT PIN)

```
// Para que a contagem não seja afetada pela chamada da função
int count = 0;
void contador(int pin) {
    static const int UP = 0, DOWN = 1;
    static int state = UP;
    switch (state) {
        // Quando o botão não é premido
        case UP:
            Serial.println("UP");
            if(digitalRead(pin) == 1) {
                state = DOWN;
            }
            break;
        // Quando o botão é premido
        case DOWN:
            if(digitalRead(pin) == 0) {
                state = UP;
            }
            // Incrementa o valor pelo o tempo que é premido
            count++;
            Serial.println(count);
            break;
    }
}
```


7. Gerador sinais (onda sinusoidal (s), ou triangular (t), ou quadrada(q))

PROTÓTIPO: VOID
GERADOR(INT RATE, INT
PERÍODO, CHAR TIPOONDA)

```
void gerador(int rate, int periodo, char tipoOnda) {
    unsigned int T = 1000 / rate;
    static unsigned long t1 = millis(), t0 = 0
    static int state;
    static const int q = 0, s = 1, t = 2;
    //ESCOLHER OPÇÃO DO TIPO DE ONDA
    if (tipoOnda == 'q') {
        state = q;
    } else if (tipoOnda == 's') {
        state = s;
    } else if (tipoOnda == 't'){
        state = t;
    }

    switch (state) {
        // Opção quadrada
        case q:
            static const int UPPER = 0, DOWN = 1;
            static int bound = UPPER;
            switch (bound) {
                case UPPER:
                    Serial.println(10);
                    if(t1 - t0 >= T/2) {
                        t0 = t1;
                        state = DOWN;
                    }
                    break;
                case DOWN:
                    Serial.println(0);
                    if(t1 - t0 >= T) {
                        t0 = t1;
                        state = UPPER;
                    }
                    break;
            }
        // Opção Sinusoidal
        case s:
            Serial.println(1.0 * sin(2.0*3.14*millis()));
            break;
        // Opção triangular
        case t:
            const int ASC = 0, DESC = 1;
            static int state = ASC;
            switch (state) {
                // Para quando a função está a ascender
                case ASC:
                    t1 = millis();
                    amp = (2*(10.0/T)) * (t1- t0);
                    Serial.println(amp);
                    if(t1 - t0 >= T/2) {
                        t0 = t1;
                        state = DESC;
                    }
                    break;
                // Para quando a função está a descender
                case DESC:
                    t1 = millis();
                    amp = (-2*(10.0/T)) * (t1- t0) + 10;
                    Serial.println(amp);
                    if(t1 - t0 >= T/2) {
                        t0 = t1;
                        state = ASC;
                    }
                    break;
            }
    }
}
```

PROJETO ESCOLHIDO

Fazer um piano, através do Arduino, com a opção da escolha de oitava pretendida, pelo utilizador.

Output Serial Monitor ✕

Message (Enter to send message to 'Arduino Uno' on '/dev/cu.usbmodem14201')

Escolha a sua oitava:

Escolheu a 5ª oitava.

Dó

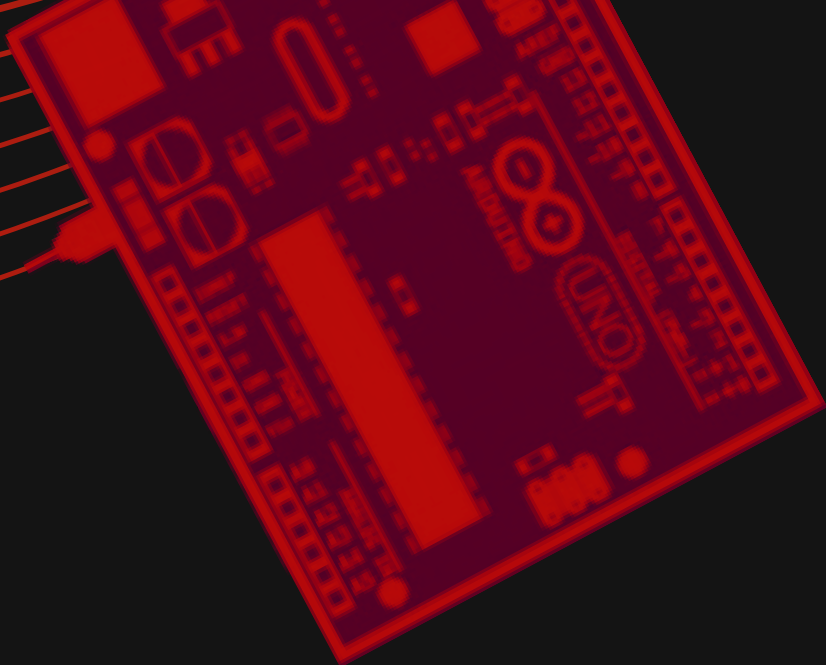
Ré

Mi

Ré

Dó

PROJETO ESCOLHIDO

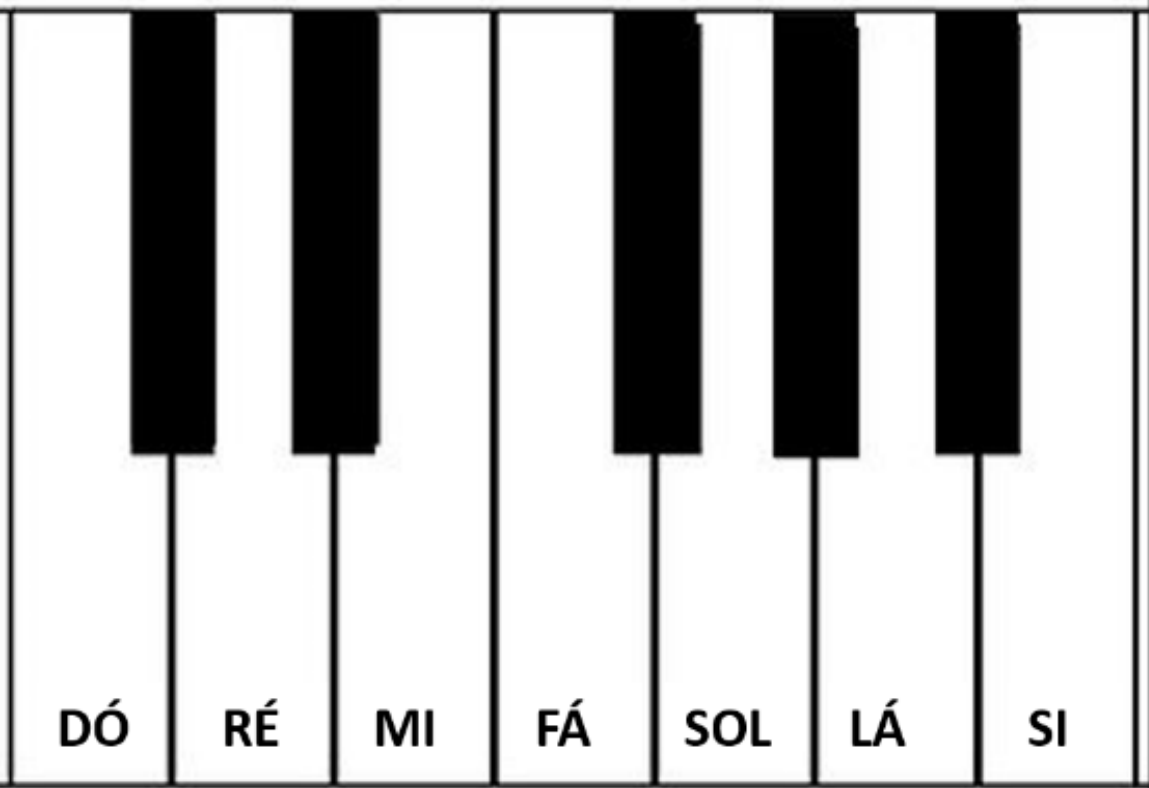


Oitavas e Notas

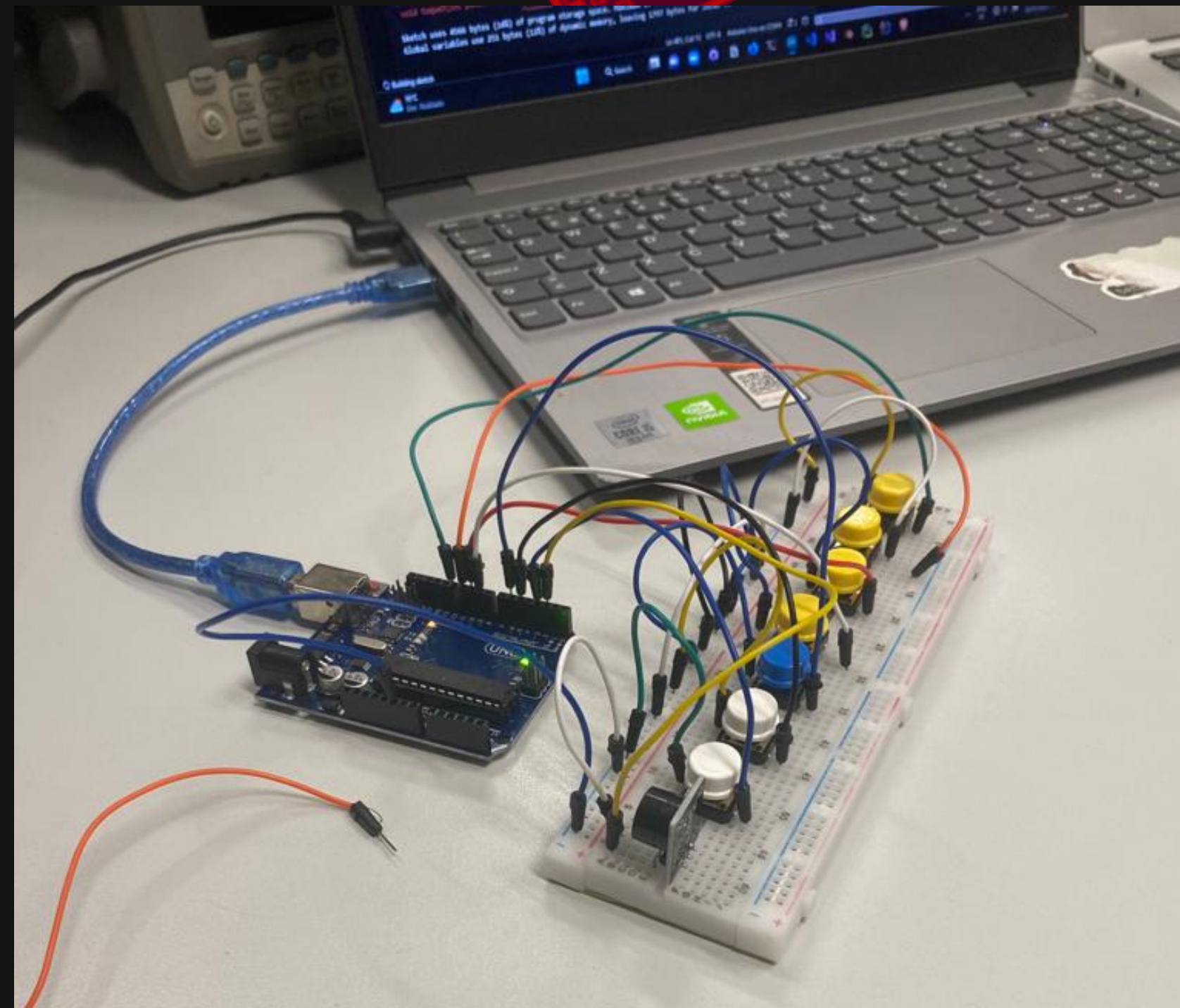
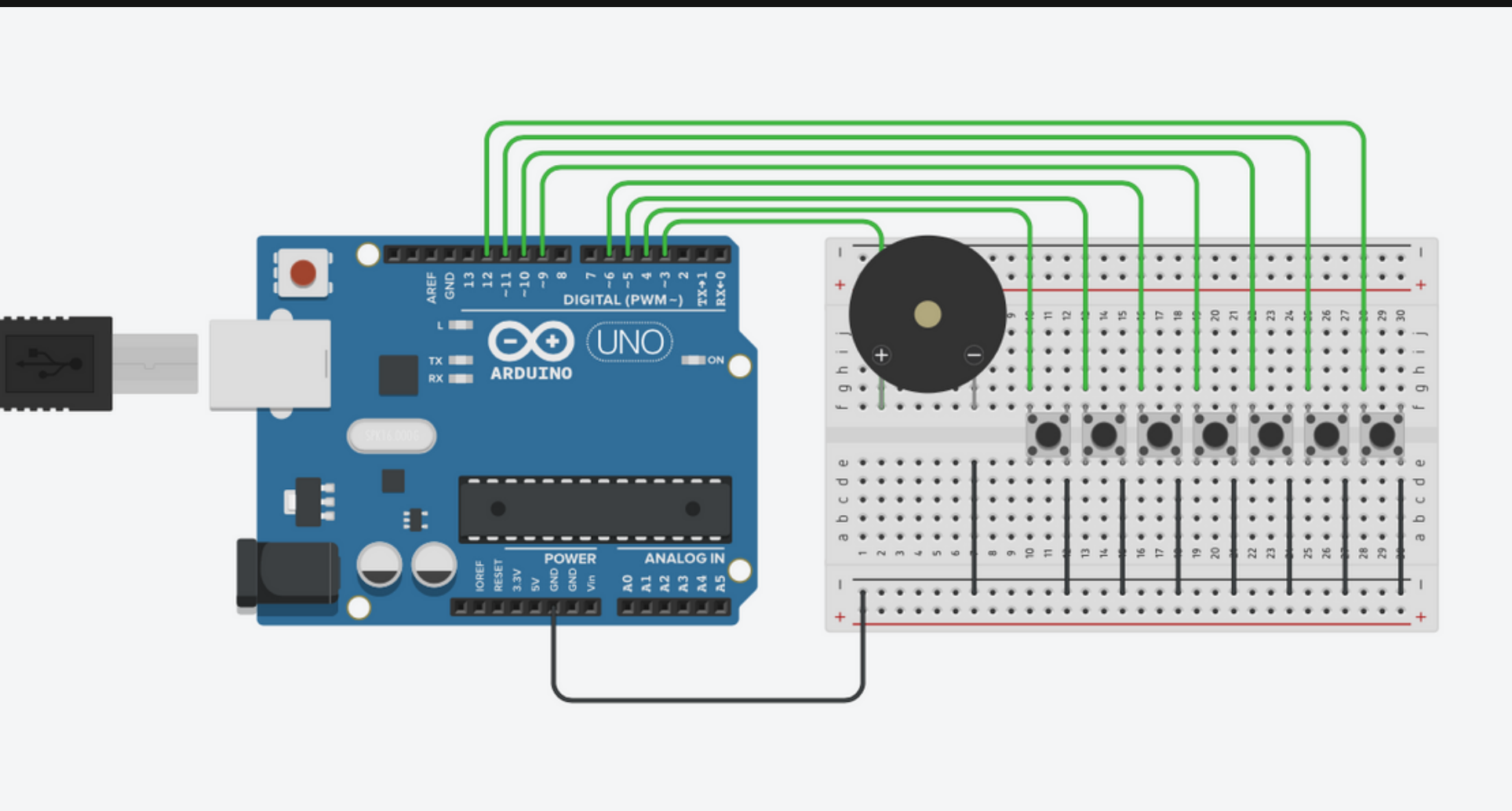
Cada nota musical não tem apenas uma frequência, mas um conjunto de frequências derivado do timbre de cada instrumento. Porém há uma frequência associada à nota, a frequência fundamental (ou básica) é o som mais grave de todas componentes de uma nota.

Frequências fundamentais das notas musicais (Hz)

Oitava	Dó	Ré	Mi	Fá	Sol	Lá	Si
Zero	16,351	18,354	20,601	21,827	24,499	27,500	30,362
Um	32,703	36,708	41,203	43,654	48,999	55,000	60,725
Dois	65,406	73,416	82,407	87,307	97,999	110,00	121,45
Três	130,81	146,83	164,81	174,61	196,00	220,00	242,90
Quatro	261,63	293,66	329,63	349,23	391,99	440,00	485,80
Cinco	523,25	587,33	659,26	698,46	783,99	880,00	971,60
Seis	1046,5	1174,7	1318,5	1396,9	1568,0	1760,0	1943,2
Sete	2093,0	2349,3	2637,0	2793,8	3136,0	3520,0	3886,4
Oito	4186,0	4698,6	5274,0	5587,7	6271,9	7040,0	7772,8
Nove	8372,0	9397,3	10548,0	11175,0	12544,0	14080,0	15546,0



PROJETO ESCOLHIDO





ESTRUTURA



1. RECETOR NUMEROS INTEIROS (PORTA SERIAL)

Protótipo: void receiveInt()

2. DETETOR TOQUE (BOTÃO)

Protótipo: void toque(int pinBotao)

3. PIANO

Protótipo: void piano()

1. Recetor numeros inteiros (porta serial)

PROTÓTIPO: VOID
RECEIVEINT()

```
int val = 0;
void receiveInt() {
    static const int WAIT = 0, READ = 1;
    static int state = WAIT;
    switch (state) {
        // Enquanto espera por um valor da consola
        case WAIT:
            if (Serial.available() > 0) {
                state = READ;
            }
            break;
        // Ler o valor da consola
        case READ:
            val = Serial.parseInt();
            Serial.println(val);
            state = WAIT;
            break;
    }
}
```

2. Detetor toque (botão)

PROTÓTIPO: VOID TOQUE N
(INT PINBOTAO)

```
void toque1(int pinBotao, int nota) {  
    int buttonPressed = 0;  
    static const int WAIT = 0, PLAY = 1;  
    static int state = WAIT;  
  
    switch (state) {  
  
        case WAIT:  
            buttonPressed = digitalRead(pinBotao);  
            if(buttonPressed == LOW) {  
                state = PLAY;  
            }  
            break;  
  
        case PLAY:  
            buttonPressed = digitalRead(pinBotao);  
            tone(PIEZ0, nota);  
            if(buttonPressed == HIGH) {  
                Serial.println("Dó");  
                state = WAIT;  
                noTone(PIEZ0);  
            }  
            break;  
    }  
}
```

3. piano()

PIANO

```
// Circuito combinado
receiveInt();
switch (val) {
  case 1:
    toque1(BTN1, 32);
    toque2(BTN2, 36);
    toque3(BTN3, 41);
    toque4(BTN4, 43);
    toque5(BTN5, 48);
    toque6(BTN6, 55);
    toque7(BTN7, 60);
    break;
  case 2:
    toque1(BTN1, 65);
    toque2(BTN2, 73);
    toque3(BTN3, 82);
    toque4(BTN4, 87);
    toque5(BTN5, 97);
    toque6(BTN6, 110);
    toque7(BTN7, 121);
    break;
}
```

```
case 3:
  toque1(BTN1, 130);
  toque2(BTN2, 146);
  toque3(BTN3, 164);
  toque4(BTN4, 174);
  toque5(BTN5, 196);
  toque6(BTN6, 220);
  toque7(BTN7, 242);
  break;
case 4:
  toque1(BTN1, 261);
  toque2(BTN2, 293);
  toque3(BTN3, 329);
  toque4(BTN4, 349);
  toque5(BTN5, 391);
  toque6(BTN6, 440);
  toque7(BTN7, 485);
  break;
```

```
case 5:
  toque1(BTN1, 523);
  toque2(BTN2, 587);
  toque3(BTN3, 659);
  toque4(BTN4, 698);
  toque5(BTN5, 783);
  toque6(BTN6, 880);
  toque7(BTN7, 971);
  break;
case 6:
  toque1(BTN1, 1046);
  toque2(BTN2, 1174);
  toque3(BTN3, 1318);
  toque4(BTN4, 1396);
  toque5(BTN5, 1568);
  toque6(BTN6, 1760);
  toque7(BTN7, 1943);
  break;
```

```
case 7:
  toque1(BTN1, 2093);
  toque2(BTN2, 2349);
  toque3(BTN3, 2637);
  toque4(BTN4, 2793);
  toque5(BTN5, 3136);
  toque6(BTN6, 3520);
  toque7(BTN7, 3886);
  break;
case 8:
  toque1(BTN1, 4186);
  toque2(BTN2, 4698);
  toque3(BTN3, 5274);
  toque4(BTN4, 5587);
  toque5(BTN5, 6271);
  toque6(BTN6, 7040);
  toque7(BTN7, 7772);
  break;
case 9:
  toque1(BTN1, 8372);
  toque2(BTN2, 9397);
  toque3(BTN3, 10548);
  toque4(BTN4, 11175);
  toque5(BTN5, 12544);
  toque6(BTN6, 14080);
  toque7(BTN7, 15546);
  break;
```