

1 マイコンボードの概要(端子(ピン)と LED とリセットスイッチ)

マイコンボードには USB mini B の端子、AC アダプタをつなぐ端子と、他の回路や電子部品とつなぐための横に並んだ端子の列が 4 つあります。端子の列はそのまま挿せるシールドと呼ばれる基板（市販されています。自作もできます。）を挿したり、ジャンプワイヤーとよばれる電線を挿してほかの回路や電子部品とつなぎます。端子の列の各端子の役割は基板上に印刷されています。各端子はマイコンのピン（足）につながっているため、ピンとよぶことがあり、以下でもそう呼びます。

マイコンボードには **4 つの LED** があります。PWR とあるのは電源が入っていると点灯する LED です。USB 端子で PC とつなぐと点灯します。TX, RX と表記のある LED は PC とマイコンが通信をしているときに点滅する LED です。**AREF の近くにある LED** はプログラムから点灯/消灯できる LED です。

マイコンボード上にあるスイッチは **リセットスイッチ** です。マイコンのプログラムを電源を入れたときと同じように最初から実行するときに使います。

表 1 DIGITAL と表示のある側のピンの機能とプログラム上での番号

| 基板上の表記 | 機能 | プログラム上での番号 / 表記 |
|--------|-----------------------|-----------------|
| 0R | デジタル入力/デジタル出力/シリアルポート | 0 |
| 1T | デジタル入力/デジタル出力/シリアルポート | 1 |
| 2 | デジタル入力/デジタル出力 | 2 |
| 3 | デジタル入力/デジタル出力/PWM | 3 |
| 4 | デジタル入力/デジタル出力 | 4 |
| 5 | デジタル入力/デジタル出力/PWM | 5 |
| 6 | デジタル入力/デジタル出力/PWM | 6 |
| 7 | デジタル入力/デジタル出力 | 7 |
| 8 | デジタル入力/デジタル出力 | 8 |
| 9 | デジタル入力/デジタル出力/PWM | 9 |
| 10 | デジタル入力/デジタル出力/PWM | 10 |
| 11 | デジタル入力/デジタル出力/PWM | 11 |
| 12 | デジタル入力/デジタル出力 | 12 |
| 13 | デジタル入力/デジタル出力 | 13 |
| GND | グラウンド(電源のマイナス側) | なし |
| AREF | A/D 変換の基準電圧 | なし |

表 2 POWER, ANALOG IN と表示のある側のピンの機能とプログラム上での番号

| 基板上の表記 | 機能 | プログラム上での番号 /表記方法 |
|--------|---------------------------|---------------------|
| RESET | マイコンのリセットピン | なし |
| 3V3 | 3.3V 電源 | なし |
| 5V | 5V 電源 | なし |
| Gnd | グラウンド (電源のマイナス側) | なし |
| Gnd | グラウンド (電源のマイナス側) | なし |
| Vin | USB の 5V または AC アダプタのプラス側 | なし |
| 0 | デジタル入力/デジタル出力/アナログ入力 | A0 |
| 1 | デジタル入力/デジタル出力/アナログ入力 | A1 |
| 2 | デジタル入力/デジタル出力/アナログ入力 | A2 |
| 3 | デジタル入力/デジタル出力/アナログ入力 | A3 |
| 4 | デジタル入力/デジタル出力/アナログ入力 | A4 |
| 5 | デジタル入力/デジタル出力/アナログ入力 | A5 |

各ピンの基板上での表記と機能、プログラム上での番号は表 1 と表 2 のようになっています。表から分かるようにデジタル出力に使えるピンには 0~13 番のピンと ANALOG IN(アナログ入力)にある 0~5 の合計 19 本あります。0 番 1 番とアナログ入力のピンは少し注意があるので最初は 2 番から 13 番のピンを使うとよいでしょう。また 13 番のピンは基板上の LED(AREF の近くの LED)につながっています。

2 ピンの機能の切り替え (pinMode)

マイコンの電源が入った直後やリセットした直後には各ピンはデジタル入力またはアナログ入力になっています (つないだ回路がマイコンに信号を出力する回路で、マイコン側も出力になっていると信号の衝突が起きてマイコン又は回路を壊す危険があるため)。デジタル出力に切り替えるにはプログラムで

```
pinMode(<ピン番号>, OUTPUT);
```

デジタル入力に切り替えるにはプログラムで

```
pinMode(<ピン番号>, INPUT);
```

と書きます。デジタル入力で内蔵プルアップを有効にするには

```
pinMode(<ピン番号>, INPUT_PULLUP);
```

と書きます。<ピン番号>のところに直接数値を書いたり、変数や計算式を書きます。2～13ピンでデジタル入力を使う場合と、アナログ入力ピン A0～A5 でアナログ入力を使う場合は `pinMode` を書かなくて構いません。

例1) `pinMode(2, OUTPUT);` // 2 番ピンを出力モードにする

例2) `pinMode(i, OUTPUT);` // 変数 `i` の値が指すピンを出力モードにする

例3) `for (int i=2; i<=13; i++) { // 2 番ピンから 13 番ピンを出力モードにする
 pinMode (i, OUTPUT);
}`

例4) `for (int i=0; i<10; i++) { // 2 番ピンから 10 個(11 番ピンまで)を出力モードにする
 pinMode (i+2, OUTPUT);
}`

3 デジタル入力

デジタル入力を使うとマイコンはピンの電圧が高い(HIGH または 1 で表す)か低い(LOW または 0 で表す)を知ることができます。おおよそ電源電圧である 5V の半分(2.5V)より高いと HIGH、低いと LOW となります。デジタル入力を使うには `digitalRead` 関数を使います。2 番ピンの電圧を知って電圧が低いときに基板上の 13 番ピンにつながった LED を点灯するプログラムはプログラムリスト 1 のようになります。ジャンプワイヤの片側を 2 番ピンに挿し、反対側をグラウンド(GND または Gnd)あるいは電源(5V)に挿して LED が点灯/消灯の様子を確認してみましょう。またデジタル入力ピンになにもつながないと LOW になるか HIGH になるかは電気的なノイズによって決まります。ジャンプワイヤの片側を 2 番ピンに挿したまま、反対側を手で触ると LOW になるときも HIGH になるときもあることが分かります。

プログラムリスト 1 入力で LED が点滅するプログラム

```
void setup() {  
    pinMode(13, OUTPUT);  
}  
void loop() {  
    if (digitalRead(2) == LOW) {  
        digitalWrite(13, HIGH);  
    } else {  
        digitalWrite(13, LOW);  
    }  
}
```

4 デジタル出力

デジタル出力を使うとマイコンから電圧を出力することができます。電圧の状態は LOW（または 0）と HIGH（または 1）で表します。正確には少し違いますが図 1 のように LOW に設定すると電源のグラウンド側（マイナス側）に HIGH にするとプラス側（このマイコンボードでは電源が 5V なので 5V）にスイッチが切り替わるとして下さい。つまりデジタル出力はプログラムから制御できる電源になります。デジタル出力を電源として使う場合は、マイコン内のスイッチや配線が扱える電流に制限があるため、1 ピンあたり 10mA までにしておくのが無難です。そのため LED を点灯することはできますが、模型用のモータを直接つなぐことはできません。

LED を点灯する回路(図 2)を例に考えてみましょう。図 2 のように抵抗と LED と電源とスイッチをつなぐとスイッチで LED の点灯/消灯ができます。図 2 では電源に手動のスイッチが付いていますが、これをマイコンのピンに置き換えます(図 3)。図 3 ではマイコンの 2 番ピンに抵抗と LED をつないでいます。電源のグラウンドと GND の端子はマイコンボード内でつながっていますので、プログラムからスイッチを HIGH 側につなぐと LED に電流が流れて点灯し、LOW 側につなぐと消灯します。プログラムでは

```
pinMode(2, HIGH);
```

または

```
pinMode(2, LOW);
```

と書きます。マイコンの 13 番ピンにはボード上で図 3 のように LED がつながっているので配線することなくプログラムの動作を確かめられます。また図 3 の回路は図 4 のように書きます(別の表記をすることもあります)。

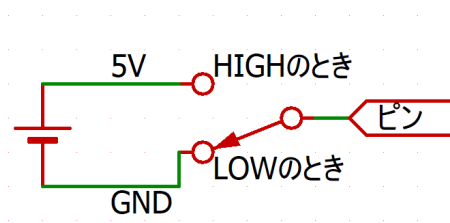


図 1 デジタル出力の概念図

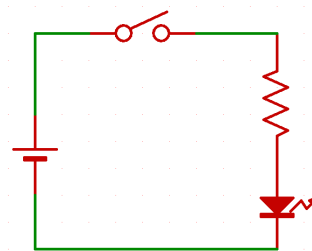


図 2 LED をスイッチで点滅する回路

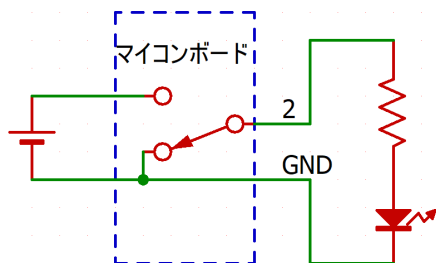


図3 マイコンの2番ピンにLEDつなぎ点灯制御する回路（概念図）

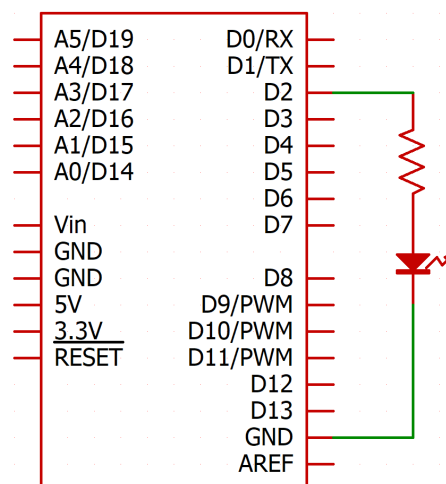


図4 マイコンの2番ピンにLEDをつなぎ点灯制御する回路

4.1 デジタル出力の初期設定 (pinMode)

デジタルの2番から13番のピンを使うときには、まず

```
pinMode(<式>, OUTPUT);
```

とすることでピンを出力モードにします。式のところには数字を書いても構いませんし、変数や式を使っても構いません。

4.2 プログラムからデジタル出力を制御する (digitalWrite)

デジタル出力の値を変えるには

```
digitalWrite(<式>, HIGH);
```

又は

```
digitalWrite(<式>, LOW);
```

と書きます。一度 digitalWrite の文を実行すると次に実行するまでピンの値はそのままです。式のところには数字を書いても構いませんし、変数や式を使っても構いません。

例5) digitalWrite(2, HIGH); // 2番ピンを HIGH にする

例6) digitalWrite(i, LOW); // 変数 i の値が指すピンを LOW にする

```
例7) for (int i=2; i<=13; i++) { // 2番ピンから13番ピンを HIGH にする
    digitalWrite (i, HIGH);
}
```

```
例8) for (int i=0; i<10; i++) { // 2番ピンから10個(11番ピンまで)を HIGH にする
    digitalWrite (i+2, LOW);
}
```

4.3 マイコンボード上の LED を点滅するプログラム

マイコンボード上の LED(13 番につながっている)を 1 秒点灯、1 秒消灯することを繰り返すプログラムはプログラムリスト 2 のように書けます。このプログラムは Arduino IDE のファイルメニューから「スケッチ例」→「01 BASICS」→「Blink」をクリックして開く例と同じ動作をします。

Blink スケッチ(プログラムリスト 3 は Blink スケッチからの抜粋)では最初に led という名前の値を変更できない変数を宣言して 13 を代入しています。プログラム内で led と書くと変数の値が 13 なので直接 13 を書いたのと同じことになります。わざわざ宣言しているのは、13 ではなく led と書くことで意味をわかりやすくし、必要なら 1 カ所を書き換えれば別の数字にできるようにするためです。たとえば基板上的 LED ではなく 2 番ピンにつないだ LED を点滅したくなったときにプログラムリスト 2 では 3 カ所書き換える必要がありますが、プログラムリスト 3 では 1 カ所だけ書き換えればよいのでミスを防げます。

プログラムリスト 2 基板上的 LED を点滅
するプログラム(1)

```
void setup() {  
    pinMode(13, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(13, HIGH);  
    delay(1000);  
    digitalWrite(13, LOW);  
    delay(1000);  
}
```

プログラムリスト 3 基板上的 LED を点滅
するプログラム(2)

```
const int led = 13;  
void setup() {  
    pinMode(led, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(led, HIGH);  
    delay(1000);  
    digitalWrite(led, LOW);  
    delay(1000);  
}
```

4.4 ブレッドボード上のLEDを点滅する

図5の回路を実際に作ってみましょう。配線するときはマイコンボードにつながった **USB ケーブルをPCから抜いておきます**。ブレッドボードとジャンプワイヤ2本、抵抗(1kΩ程度、300Ω～4.7kΩ程度ならよい)、LED(色は何でもよい)を用意して配線します。ブレッドボードの穴は縦5つがつながっています。図通りの位置に配線する必要はありませんが、LEDの足とジャンプワイヤ、LEDの足と抵抗の足、抵抗の足とジャンプワイヤが同じ縦5つのところに挿します。またLEDには向きがあります。2番ピン側(抵抗側)にLEDの足の長い側を挿します。プログラムリスト2または3を書き換えて2番ピンを初期化して、digitalWriteでHIGH/LOWを切り替えるようにしてみましょう。

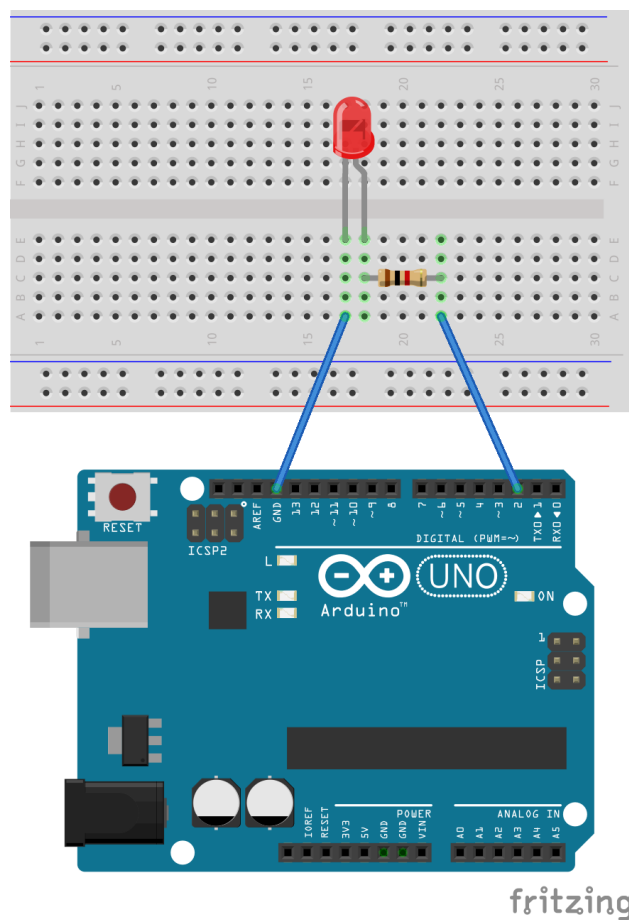


図5 マイコンの2番ピンにLEDをつなぐ回路の実体配線図

4.5 マイコンボードとブレッドボードのLEDを点滅させる

マイコンボード上(13番ピン)とブレッドボード上(2番ピン)のLEDを点滅させるにはどうしたらいいでしょうか。図5の配線のまま、プログラムを変えてみます。プログラムリスト4をみてください。2番ピン用と13番ピン用のpinModeとdigitalWriteの文が並んでいます。プログラムは十分に速く動くので、2つの文は順番に実行されますがLEDは同時に点滅して見えます。細かいタイミングを気にする必要がなければ、2番と13番のどちらの文が先にあっても大丈夫です。

ではプログラムリスト5のプログラムはどのような動作をするでしょうか。2番ピンがHIGHのとき13番ピンをLOWに、2番ピンがLOWのとき13番ピンをHIGHにするプログラムです。このプログラムの場合には2つのLEDが交互に点滅します。

プログラムリスト4 マイコンボードとブレッドボードのLEDを同時に点滅させる(1)

```
void setup() {  
    pinMode(2, OUTPUT);  
    pinMode(13, OUTPUT);  
}  
void loop() {  
    digitalWrite(2, HIGH);  
    digitalWrite(13, HIGH);  
    delay(1000);  
    digitalWrite(2, LOW);  
    digitalWrite(13, LOW);  
    delay(1000);  
}
```

プログラムリスト5 マイコンボードとブレッドボードのLEDを同時に点滅させる(2)

```
void setup() {  
    pinMode(2, OUTPUT);  
    pinMode(13, OUTPUT);  
}  
void loop() {  
    digitalWrite(2, HIGH);  
    digitalWrite(13, LOW);  
    delay(1000);  
    digitalWrite(2, LOW);  
    digitalWrite(13, HIGH);  
    delay(1000);  
}
```


4.6 もっと多くのLEDを点滅させる

もっと多くのLEDを点滅させるにはどうしたらいいでしょうか。プログラムリスト4, 5のようにpinModeの文とdigitalWriteの文を増やせばよいことが分かります。それでは配線の方はどうでしょうか。図6をみてください。図6はLEDを2番ピンから6番ピンにそれぞれつないだ配線例です。LEDと抵抗が5組並んでいます。ブレッドボードの端(図の上下)は横にすべての穴があがっています。これを電源(ここではグラウンド)用に使くと配線が分かりやすくなります。図では下のマイナスの列にグラウンド(GND)をつなぎ、その列からLEDにつないでいます。順番に点灯する例をプログラムリスト6に示します。点灯するLEDがだんだん増えていくようにするにはどうしたらいいでしょうか。考えてみて下さい。

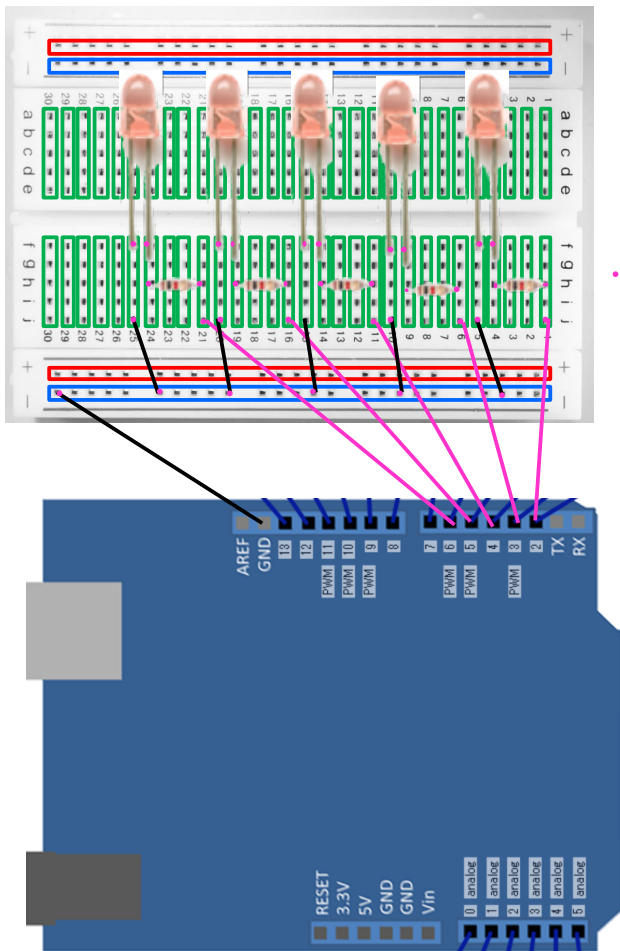


図6 マイコンの2番ピンから6番ピンにLEDをつなぐ回路の実体配線図

プログラムリスト6 5個のLEDを順番に点滅させる

```
void setup() {  
    for (int i=2; i<6; i++) {  
        pinMode(i, OUTPUT);  
    }  
}  
  
void loop() {  
    digitalWrite(2, HIGH);  
    delay(500);  
    digitalWrite(2, LOW);  
    digitalWrite(3, HIGH);  
    delay(500);  
    digitalWrite(3, LOW);  
    digitalWrite(4, HIGH);  
    delay(500);  
    digitalWrite(4, LOW);  
    digitalWrite(5, HIGH);  
    delay(500);  
    digitalWrite(5, LOW);  
    digitalWrite(6, HIGH);  
    delay(500);  
    digitalWrite(6, LOW);  
}
```

5 圧電スピーカーから音を出す

圧電スピーカー（圧電素子とよばれることもある）という部品を使うと音を出せます。圧電スピーカーには2つ端子があるので、図7のように片側をグラウンドに片側をデジタル出力のできるピン（図7では8番ピン）につなぎます。プログラムリスト7のように書くと電源が入るかリセットすると440Hzの音が0.5秒(500ミリ秒)でます。toneで音を出し始めることを命令します。

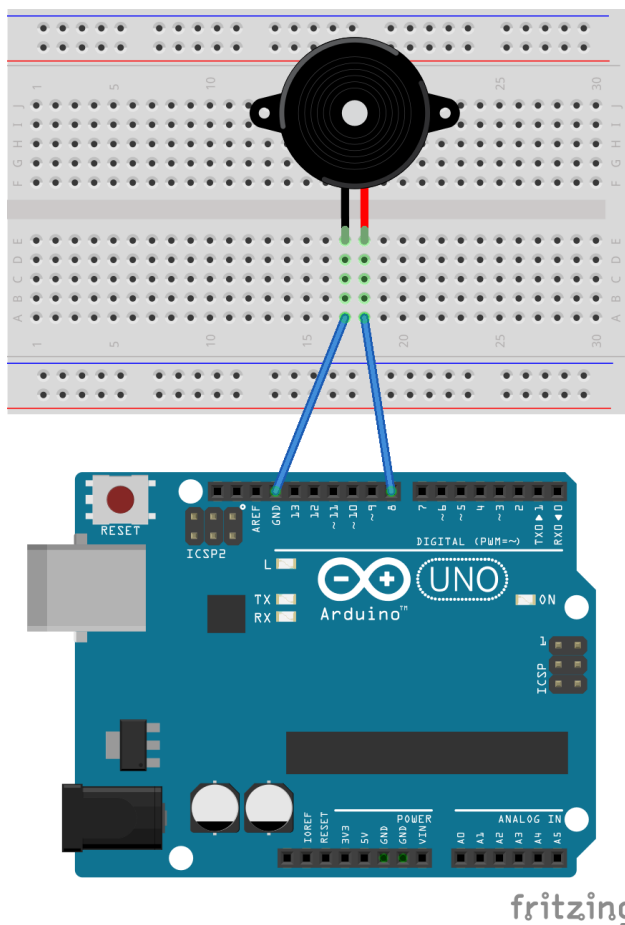
```
tone(<音を出すピンの番号>, <音の高さ>);
```

と書きます。プログラムリスト7では8ピンから440Hzの音を出し始めます。そしてdelayで500ミリ秒待ちnoToneで音を止めます。

```
noTone(<音を出しているピンの番号>);
```

と書きます。noToneがないと音がずっとでたままです。

音階と音の周波数の関係は表3のようになっています。プログラムリスト8のように書くとドレミファソをずっと繰り返し出します。音を止めたいときは圧電スピーカーの配線を抜くか、USBケーブルを抜いて電源を切ります。休符のように無音を作りたいときは音を出さずにdelayで少し待ちます。また音と音の間を少し開けると歯切れがよく聞こえます。工夫してみてください。



プログラムリスト7 音を出すプログラム

```
void setup()
{
    tone(8, 440);
    delay(500);
    noTone(8);
}

void loop()
{
}
```

図7 圧電スピーカーをマイコンボードにつなぐ

LEDなどと圧電スピーカを一緒に使うときには、LEDと圧電スピーカのピンが同じにならないように決めます。例えばLEDを2番から6番ピンにして圧電スピーカを8番ピンにします。圧電スピーカを2番ピンにしてLEDを3番から6番ピンにしても構いません。tone/noToneはデジタル出力のピンならどれでも使えます。

表3 音階と周波数の関係

| 音階 | 周波数[Hz] |
|----|---------|
| ド | 523 |
| レ | 587 |
| ミ | 659 |
| ファ | 698 |
| ソ | 784 |
| ラ | 880 |
| シ | 988 |
| ド | 1046 |

プログラムリスト 8 ドレミファソを繰り返すプログラム

```

void setup()
{
}
void loop()
{
    tone(8, 523);
    delay(500);
    noTone(8);

    tone(8, 587);
    delay(500);
    noTone(8);

    tone(8, 659);
    delay(500);
    noTone(8);

    tone(8, 698);
    delay(500);
    noTone(8);

    tone(8, 784);
    delay(500);
    noTone(8);

    delay(1000);
}

```