

# Relé controlado por time usando arduino.

---

## Objetivo

Controlar cargas de corrente alternada (de até 10A) por time (tempo). Ou seja, O projeto final terá a função de um temporizador e ao mesmo tempo, que tenha um custo de fabricação barato.

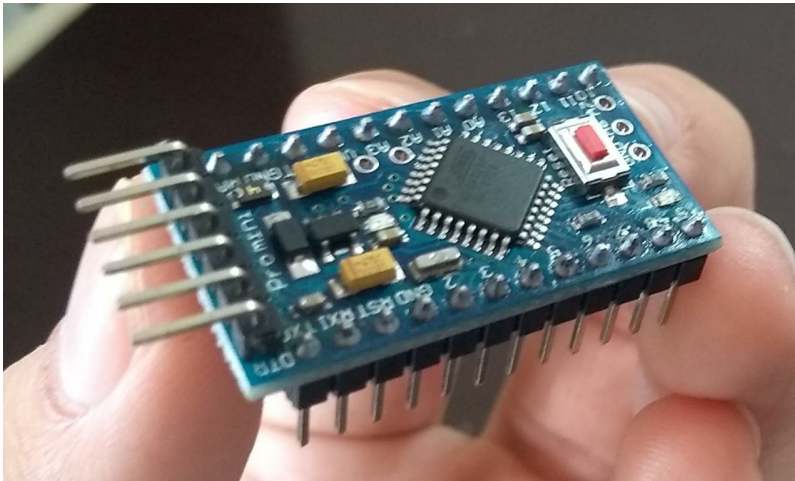
## Motivação

Esse projeto nasceu como uma solução de um problema prático. Onde temos um açude (um "buraco" cheio d'água) onde é feito a criação de peixes e um pequeno rio. O rio está abaixo do açude de forma que a água não vá de forma natural até o açude. O circuito que irá controlar a bomba d'água terá que ligar a bomba, aguardar algum tempo e depois desliga-lo por certo tempo e essa atividade será cíclica.

## Materiais usados

### 1. Arduino pro mini

- O uso de um arduino pro mini em vez de um nano ou uno é justamente por causa do preço. O arduino pro mini custa em média R\$13.
- Esse aqui, eu mesmo que soldei mais no projeto eu usei outro que não tinha sido soldado.



### 2. Relé

- No meu caso, eu usei um que é ativado com 5V. A maioria que se encontra no mercado são de 12V ativação.



### 3. Fonte com saída 5V

- o Essa eu retirei de um carregador de celular



### 4. Led e resistor

- o Usei Led amarelo e um resistor de 320 ohms para indicar que o relé está ligado.

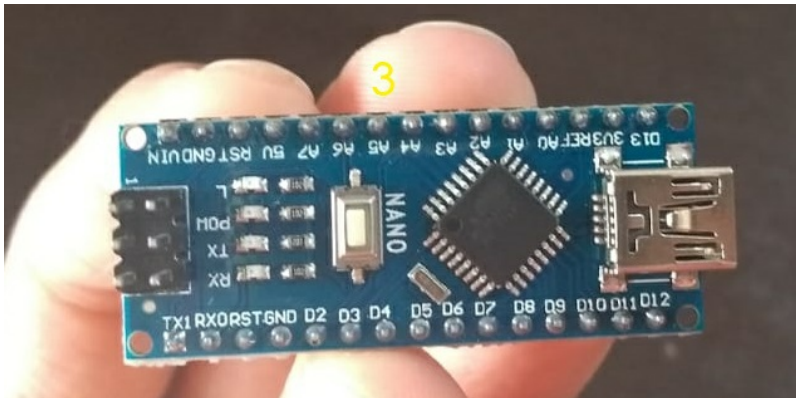
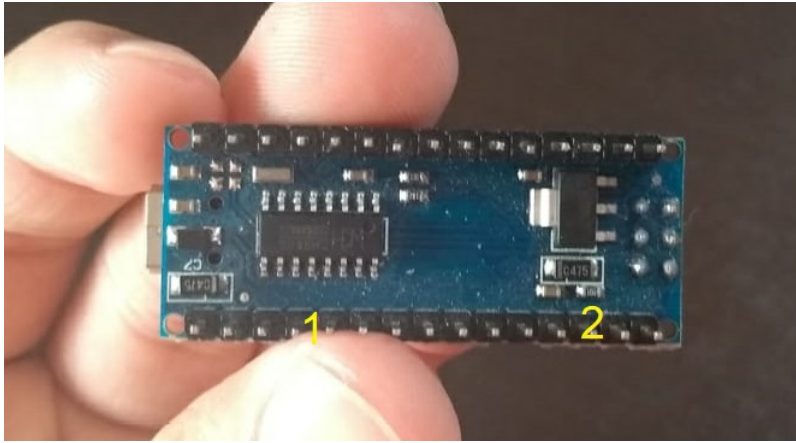
Produto	Preço em R\$
Arduino pro mini	13
Relé	3
Fonte 5V	10
Led e resistor	<0.20
total:	R\$26.2

## Começando o projeto ...

---

### Oque é arduino pro mini e como programa-lo?

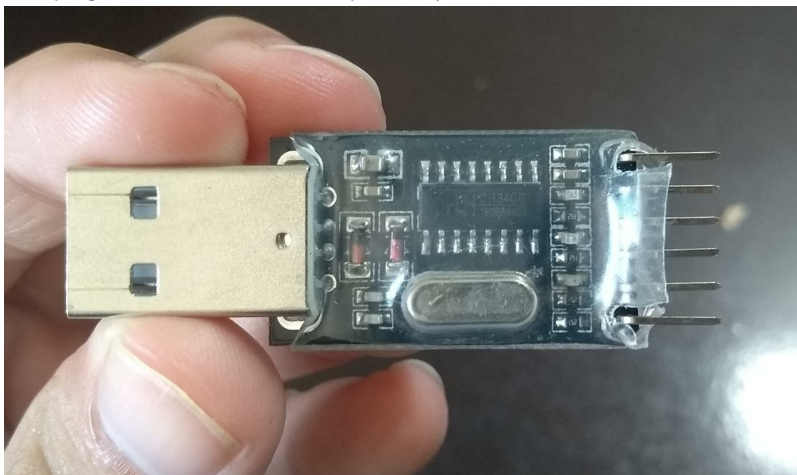
O arduino pro mini é "1/3" arduino nano. Ele só tem o micro controlador em si (o ATmega328P de 5V). Portanto não tem conversor serial e regulador de tensão.



Onde:

- 1 é conversor serial.
- 2 é regulador de tensão.
- 3 é micro controlador.

Para programa-lo vamos usar essa pecinha (também chamada de conversor serial):



Se você repara, verá que ela tem o mesmo chip serial do Arduino nano.

Fazendo a ligação. Atenção pois o Tx do conversor é o Rx do Arduino e vice-versa :).

Conversor serial	Arduino
Vcc 5V	Vcc
GND	GND
Tx	Rx
Rx	Tx

Para fins de teste, vamos carregar o código blink (aquele que pisca o led da placa Arduino) modificado.

```
int cont = 1;

void setup()
{
  Serial.begin(115200);

  pinMode(LED_BUILTIN, OUTPUT);
}

void loop()
{
  digitalWrite(LED_BUILTIN, HIGH);
  delay(100);
  digitalWrite(LED_BUILTIN, LOW);
  delay(100);

  digitalWrite(LED_BUILTIN, HIGH);
  delay(100);
  digitalWrite(LED_BUILTIN, LOW);
  delay(50);

  digitalWrite(LED_BUILTIN, HIGH);
  delay(50);
  digitalWrite(LED_BUILTIN, LOW);
  delay(500);

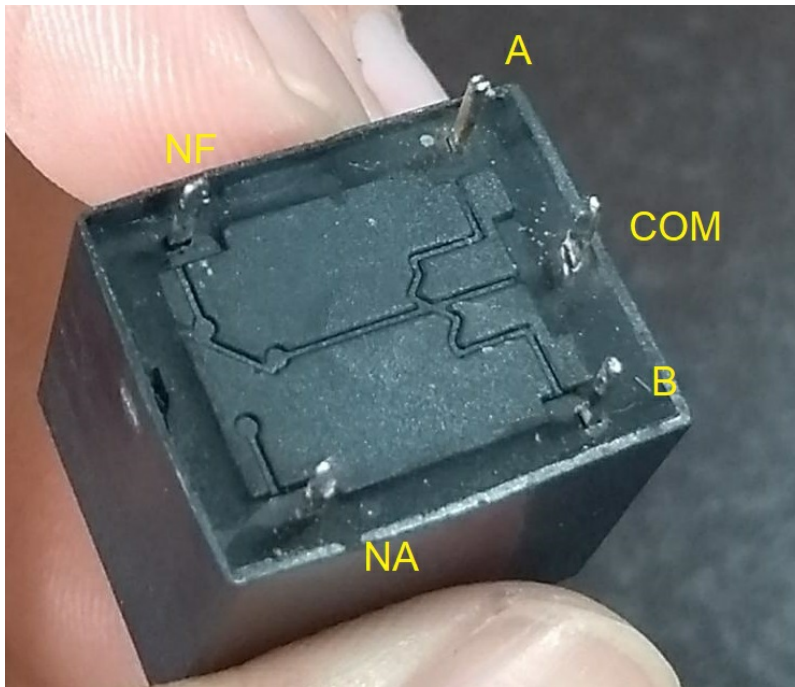
  Serial.print("I'm alive hahahaha");
  Serial.print("  cont: ");
  Serial.println(cont);
  cont++;
}
```

Obs: no caso desse conversor serial, tem que apertar o botão resete do arduino assim que o código terminar de copilar para poder carregar o código.

Se tudo deu certo. O led da placa ficará piscando e imprimindo no monitor serial "I'm alive hahahaha cont: " mais a variavel cont que vai se somando.

## Como usar o relé?

---



Onde:

- A e B é os pontos onde vamos energizar.
- COM é o ponto da carga em comum. Onde vamos ligar o fio fase 127VAC.
- NA é normalmente aberto. Se os pontos A e B não tiver energia não terá continuidade entre pontos NA e COM.
- NF é normalmente fechado. Se os pontos A e b não tiver energia terá continuidade entre os pontos NF e COM.

Olhando na foto. Energizando os pontos A e B, o ponto normalmente aberto (NA) vai ficar fechado.

Obs: Dependendo do relé, tem que montar um circuito para tirar ruído. Bom, eu ainda estou estudando sobre isso...

## Juntando tudo

---

Ok. O correto fazer uma placa (PCB) e juntar todos os componentes. Mas... eu não tenho acesso a CNC (ainda está no papel a construção de uma). Então, vou juntar em uma placa genérica e soldar alguns pontos.

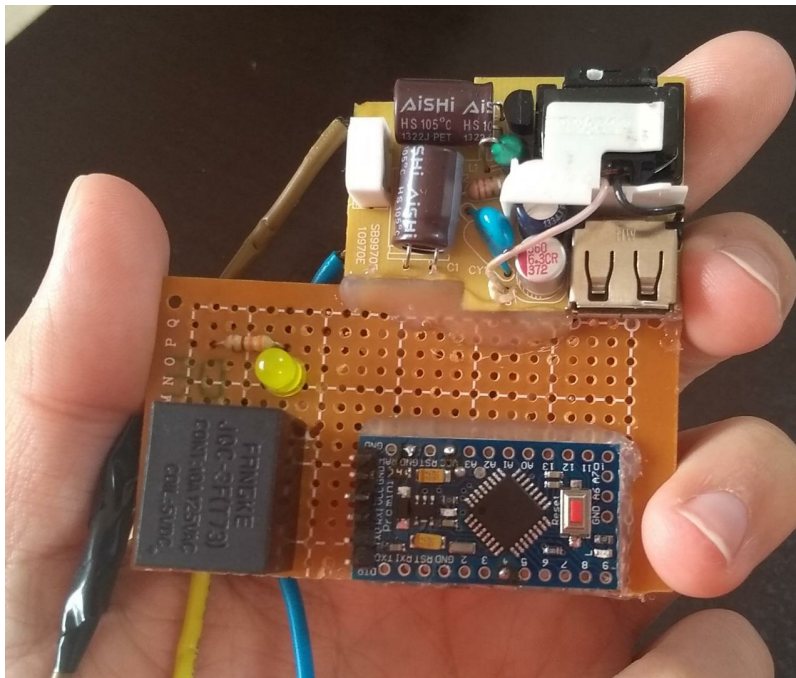
Fazendo essa ligação:

- Usando Fritzing <3

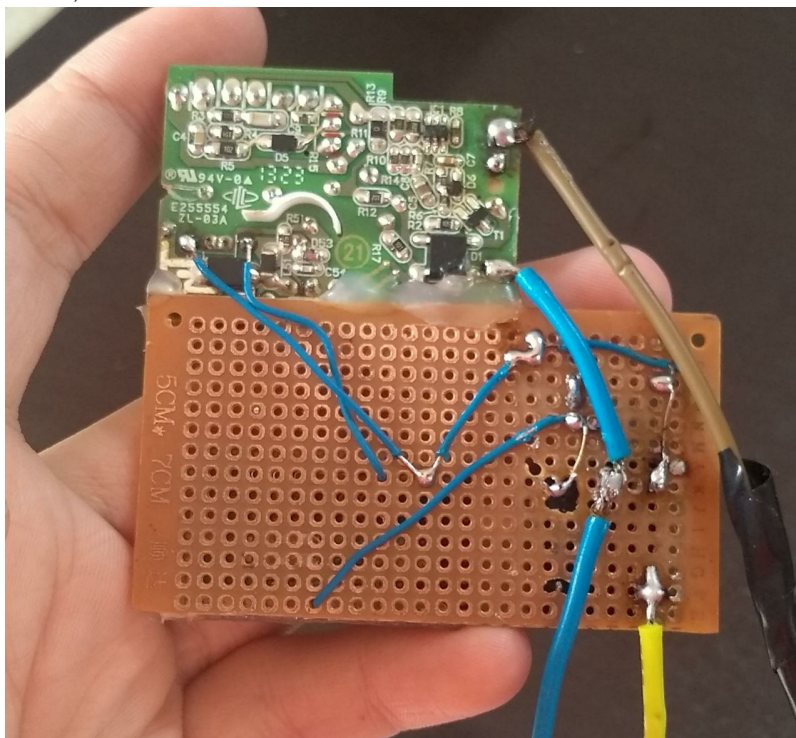
Na prática ficou assim:

- Frente :)





- Costa :)



## Código

O código não tem segredo. É mandar:

```
//1h * 60min/1h => 60min*60s/1min => 3600s * 1000ms/1s => 3 600 000ms

digitalWrite(4, HIGH);
delay(1800000) //30min
digitalWrite(4, LOW);
delay(3600000) //1h
```

Obs: o tempo de delay não foi testado :/

## Packages (itens adicionais)

---

Tendo o principal que é o hardware (arduino). Todo outro processo de otimização como:

- Gerenciar da irrigação das plantas.
- Monitoramento do nível do açude.
- Uma mini estação meteorológica para monitorar temperatura, índice de chuva e vento.

Todos esses processos tornar-se possíveis.

## Todo projetos esta disponível no Github

[https://github.com/wesley-cantarino/Sensor\\_Nivel\\_Acude](https://github.com/wesley-cantarino/Sensor_Nivel_Acude)