**DESCRIÇÃO DO CIRCUITO - PROJETO FINAL**

**PROJETO**

Neste circuito, terá vários componentes como: sensor de luminosidade, temperatura e de movimento. Juntos, eles mostrarão as informações por meio de uma tela LCD.

**MATERIAIS**

* 2 Protoboards
* 1 Buzzer
* TMP36
* 36 jumpers
* Fotorresistor
* 2 leds
* LCD 16 x 2
* PIR
* 4 Resistores

**MONTAGEM**

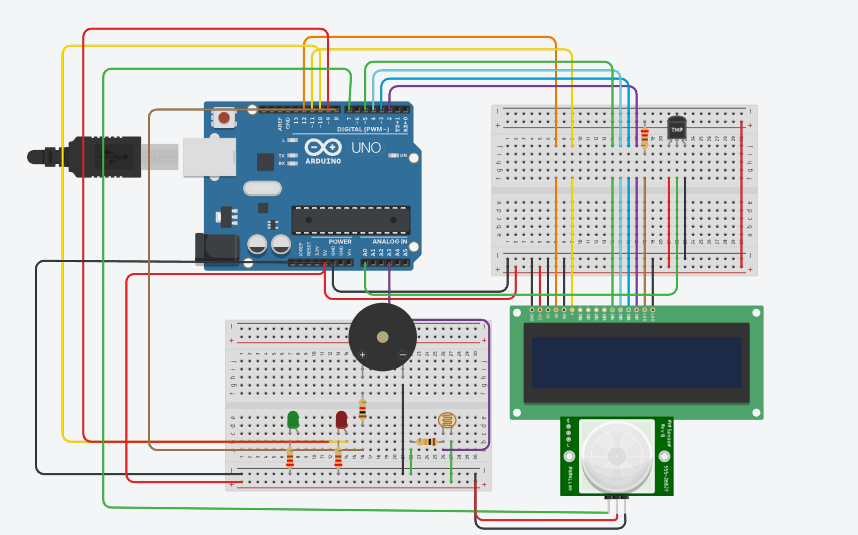
Para se montar o circuito, mostrado na Figura 1, é preciso, primeiramente, conectar os pinos do monitor LCD no Arduino, sendo estas conexões: os pinos 1, 5 e 16 em entradas GND; o pino 2 numa entrada de +5V; e os pinos 4, 6, 11, 12, 13 e 14 nas entradas de controle 12, 11, 5, 4, 3 e 2.

Depois, deve-se conectar os LEDs verde e vermelho no Arduino com o auxílio de jumpers, com os ânodos nos pinos de controle 9 e 10, respectivamente, e os cátodos no GND, intermediados por resistores.

O cátodo do buzzer é conectado no pino negativo por meio de um jumper na protoboard, já seu anodo é conectado por meio de um resistor e um jumper no pino 8 do Arudíno.

Por fim, devem ser conectados os pinos dos sensores no Arduino, sendo: o sensor de movimento com o pino da esquerda num pino de controle (neste circuito, o 7), o central em +5V e o da direita em GND; o sensor de luminosidade com o pino da direita também em +5V, e o da esquerda conectado tanto no GND, intermediado por um resistor, quanto num pino analógico de controle (no caso, o pino A3); e o sensor de temperatura com o pino da esquerda no +5V, o central num pino analógico de controle (no caso, o A0), e o da direita no GND.

**Figura 1 -** Representação do Circuito no Tinkercad



**DESCRIÇÃO DO CÓDIGO**

Como pode ser visto na Figura 2, o código se inicia com a inclusão dos comandos da biblioteca LiquidCrystal, que são relativos ao LCD. Depois, os pinos de controle do LCD são declarados como os pinos 12, 11, 5, 4, 3 e 2, respectivamente, e são criadas quatro variáveis inteiras: “SensorTemp”, com o valor de 0; “PIR”, com o valor de 7, já que este é o número do pino de controle do sensor de movimento); “foto”, com o valor de 0 e “buzzer” com o valor de 8.

Já na função setup, como pode ser visto também na Figura 2, os pinos de controle 10, 9 e 6, são definidos como saída, assim como o buzzer. Já o da variável “PIR” é declarado como entrada. Após isto, o LED é inicializado com as dimensões 16x2, seu cursor é definido para a posição 0x1 e ali é impresso a linha de cima da tabela, que mostra as unidades de medida.

Depois, na função loop, são criadas mais duas variáveis inteiras, sendo elas "SensorTensão", que recebe o valor lido pelo sensor de temperatura, e “acionamento”, que recebe o valor lido pelo sensor de movimento, além de mais duas variáveis float, sendo elas “Tensão” e "celsius", que servem para converter o valor lido pelo sensor de temperatura para graus Celsius. Após isto, o ponteiro do monitor LCD vai para a posição 0x1 para imprimir o valor convertido, e para a posição 9x1 para imprimir o valor lido pelo sensor de luminosidade. Por fim, são colocadas quatro estruturas if else: na primeira, ele ativa o LED vermelho se o índice de luminosidade detectado pelo fotoresistor for menor ou igual a 256, e o desativa caso contrário; na segunda, ele ativa o LED verde se o índice de luminosidade detectado pelo fotoresistor for maior que 768, e o desativa caso contrário; e no terceiro ele move o cursor do monitor LCD para imprimir “Alguém!” se o movimento for detectado, ou para imprimir “Ninguém!” Caso contrário.

A quarta estrutura if/else foi criada especialmente para o funcionamento correto do buzzer, caso sensor PIR seja acionado por conta de algum movimento, o buzzer irá apitar, quando o movimento parar, ele se silenciará.

**Figuras 2 e 3 -** Código do Circuito

