

Este documento irá apresentar de como preparar o Fone Bluetooth para este projeto. Os fones na maior taxa de chance serão diferentes do que será apresentado aqui, mas dará uma ideia de como preparar ele, pois terá que desmontar o fone.

Quando estiver desmontando o fone, observe onde é a alimentação da bateria, a saída de áudio e onde fica o botão de Power.

1. Alimentação de Bateria.
2. Saída de áudio.
3. Botão Power.

Importância dos itens acima:

1. Alguns fones funcionam sendo alimentados diretamente pelo o Arduino, isso é bom para não se preocupar com a autonomia da bateria, mas outros já não funcionam, assim terá que manter a bateria, ou usar outra bateria com maior capacidade de energia, recomendamos as recarregáveis. Precisa ter um cuidado, medir a tensão da bateria original do fone para colocar outra com mesma tensão ou se a alimentação for direta do Arduino usar a porta correta. No caso desde projeto usamos uma bateria externa de 3,8V recarregável, 3,8V porque era a tensão da bateria original.
2. A saída de áudio será conectada no pino A0 do Arduino, é nesta conexão que poderemos ler a frequência da onda Quadrada feita pelo celular enviado para o fone. Terá duas conexões, e juntando elas é o que faz ligar e desligar.
3. Isto é opcional, se usar os terminais do botão você poderá fazer com que o Arduino consiga ligar ou desligar o fone automaticamente usando código, no sistema contém um transistor BC547 e um resistor de base de 1,2k ohms para conseguir realizar esta função.

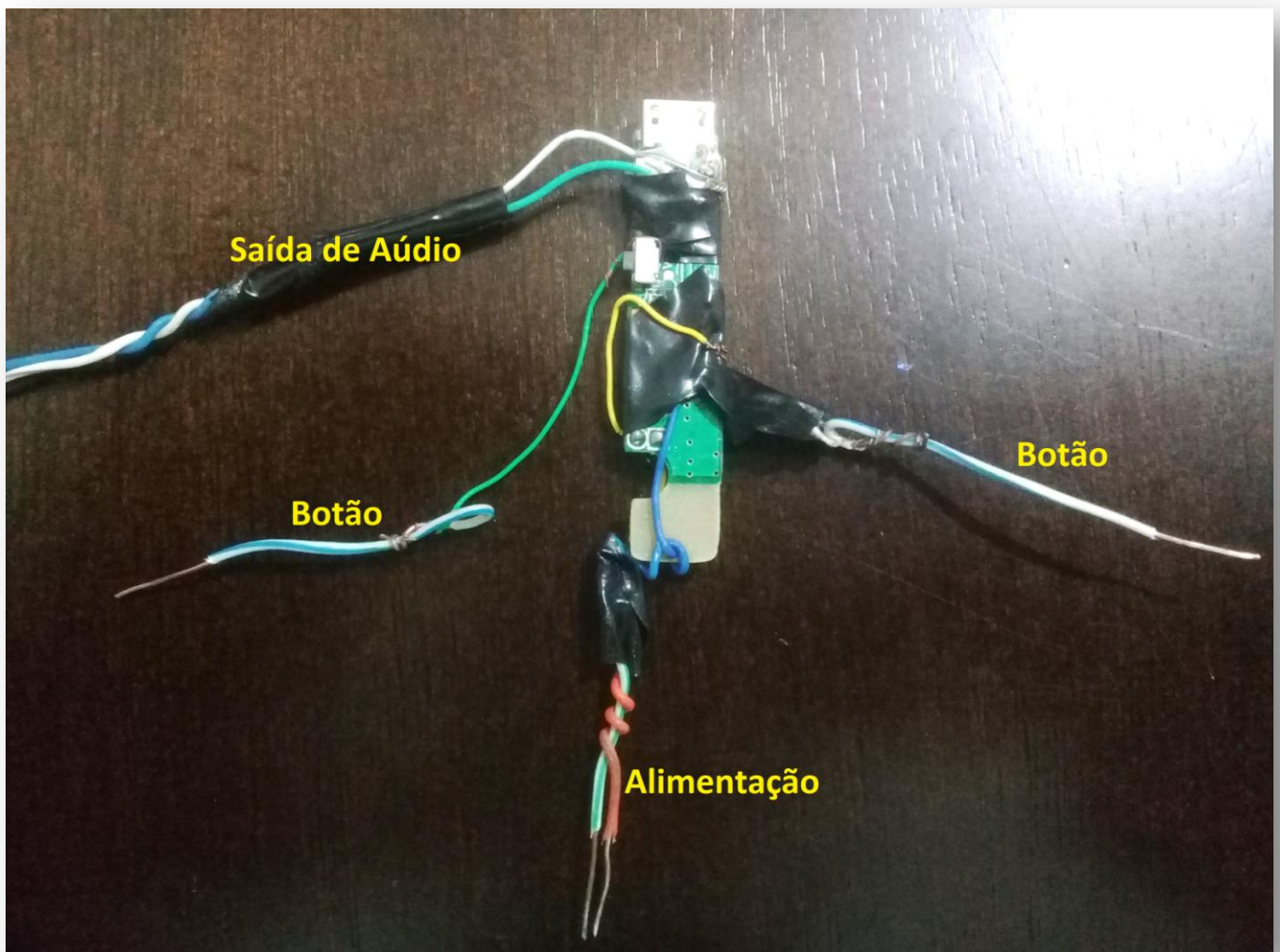
O fone usado foi do seguinte modelo:



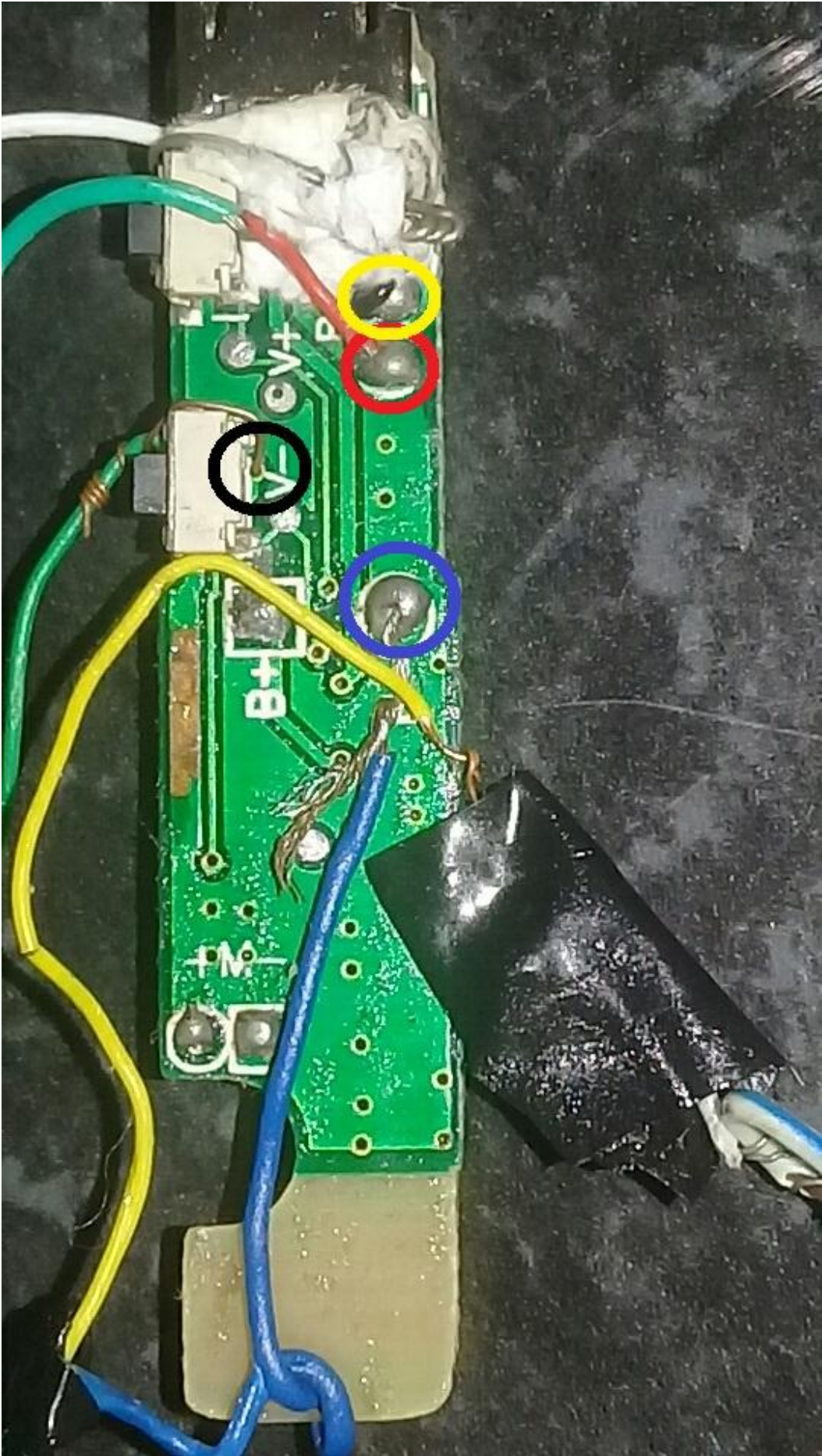
Iremos mostrar agora como ficou o fone em sua primeira montagem:

Na primeira montagem usamos fios finos para fazer as conexões, no total foram seis, saída de áudio (2), botão (2) e alimentação (2).

É preciso observar o sistema do fone para saber onde iram ser feito os ligamentos.



As seguintes imagens mostra onde foi feita estas conexões:









No final da montagem, foi tudo isolado e usado pequenos fios rígidos e posicionado de uma forma que a montagem no protoboard ficasse fácil:

