



Centro de Informática
Grupo de Engenharia da Computação



Introdução a Programação Programando com Intel Galileo Experimento 1

Objetivos

- Desenvolver um circuito com LED e botão utilizando uma protoboard;
- Implementar código utilizando o condicional IF com IDE Intel-Arduino para controlar o circuito desenvolvido.

Material Utilizado

- 1 Intel Galileo Gen 2;
- 1 Protoboard;
- 5 Fios de conexão;
- 1 Botão Tátil;
- 1 resistor de 10k Ω ;
- 1 resistor de 300 Ω ;
- 1 LED.

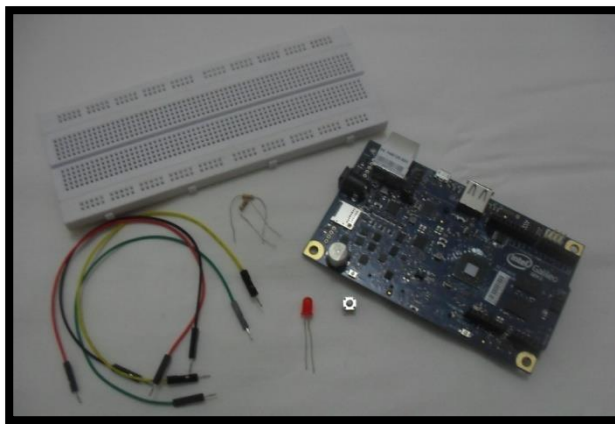


Figura 1 – Material Utilizado

Introdução

Protoboard

Para o desenvolvimento das aulas práticas, utilizaremos a *protoboard*, que é uma matriz de contatos que permite o desenvolvimento de circuitos sem necessitar de soldagem. Ela pode ser visualizada na imagem abaixo:

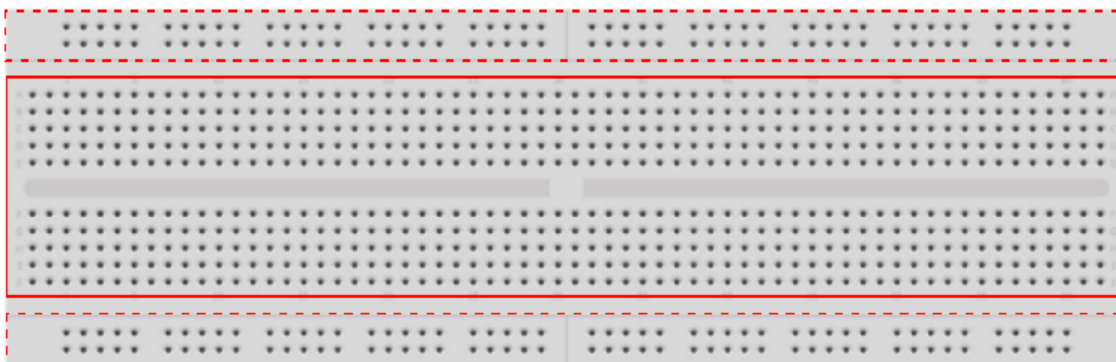


Figura 2 – Imagem de uma Protoboard

Como pode ser observada na Figura 2, a protoboard nada mais é do que uma placa de plástico, repleta de pequenos furos com ligações internas, onde você poderá fazer as ligações elétricas. Os furos nas extremidades superior e inferior, onde estão os retângulos tracejados, são ligados entre si na horizontal, enquanto que as fileiras de furos do meio, onde está o retângulo cheio, são ligadas na vertical. Veja:



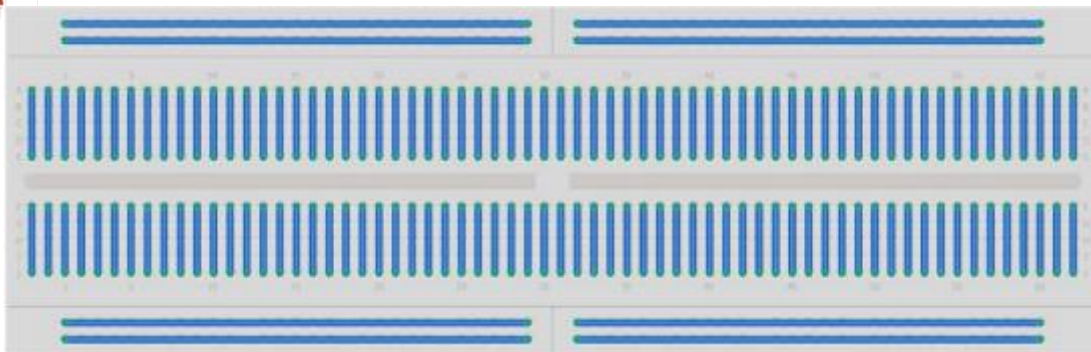


Figura 3 – Conexões da Protoboard

Observe que onde os furos estão ligados a tensão (U) é a mesma.

Vamos relembrar um pouquinho das aulas de física do ensino médio, observem o circuito esquemático da Figura 4 (lado esquerdo) pode ser obtida usando uma protoboard na Figura 4 (lado direito).

Circuito em Série

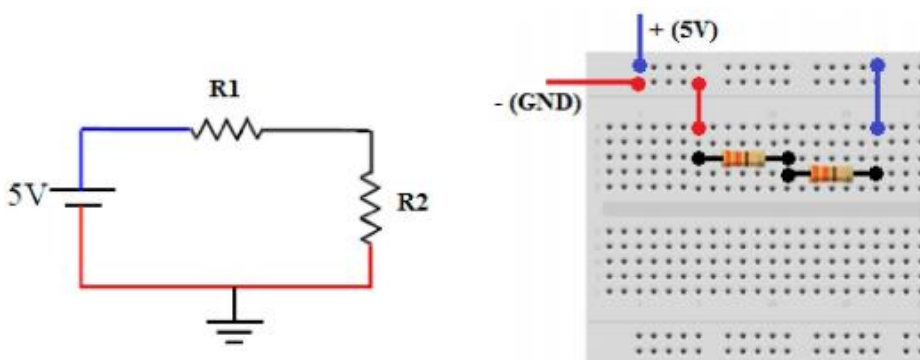


Figura 4 – Circuito em Série na Protoboard

Observe que a ligação em série entre os resistores R1 e R2 foi obtida na protoboard colocando uma extremidade do resistor R1 com a outra extremidade do resistor R2 na mesma coluna.

Circuito em Paralelo

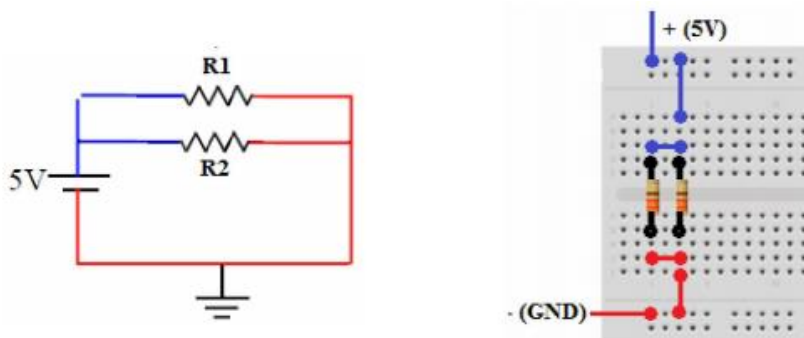


Figura 5 – Circuito em Paralelo na Protoboard



Complementando um circuito em série, temos o circuito em paralelo. Na Figura 5 (lado esquerdo), temos os resistores R1 e R2 em paralelo no diagrama esquemático. Já no lado direito da Figura 5, temos a protoboard onde as extremidades correspondentes dos resistores R1 e R2 foram colocadas nas mesmas linhas (superior e inferior).

Já que entendemos como fazer as conexões na protoboard, vamos entender como funciona o Resistor.

Resistor

O resistor é um componente elétrico que tem como característica transformar eletricidade em calor. Essa característica é fundamental para a implementação circuitos, pois com ele podemos reduzir a tensão que chega nos componentes, evitando assim que eles queimem, por exemplo.

Cada resistor tem uma capacidade de resistência e para diferenciá-los a indústria utiliza um código de cores. Veja a Tabela 7:

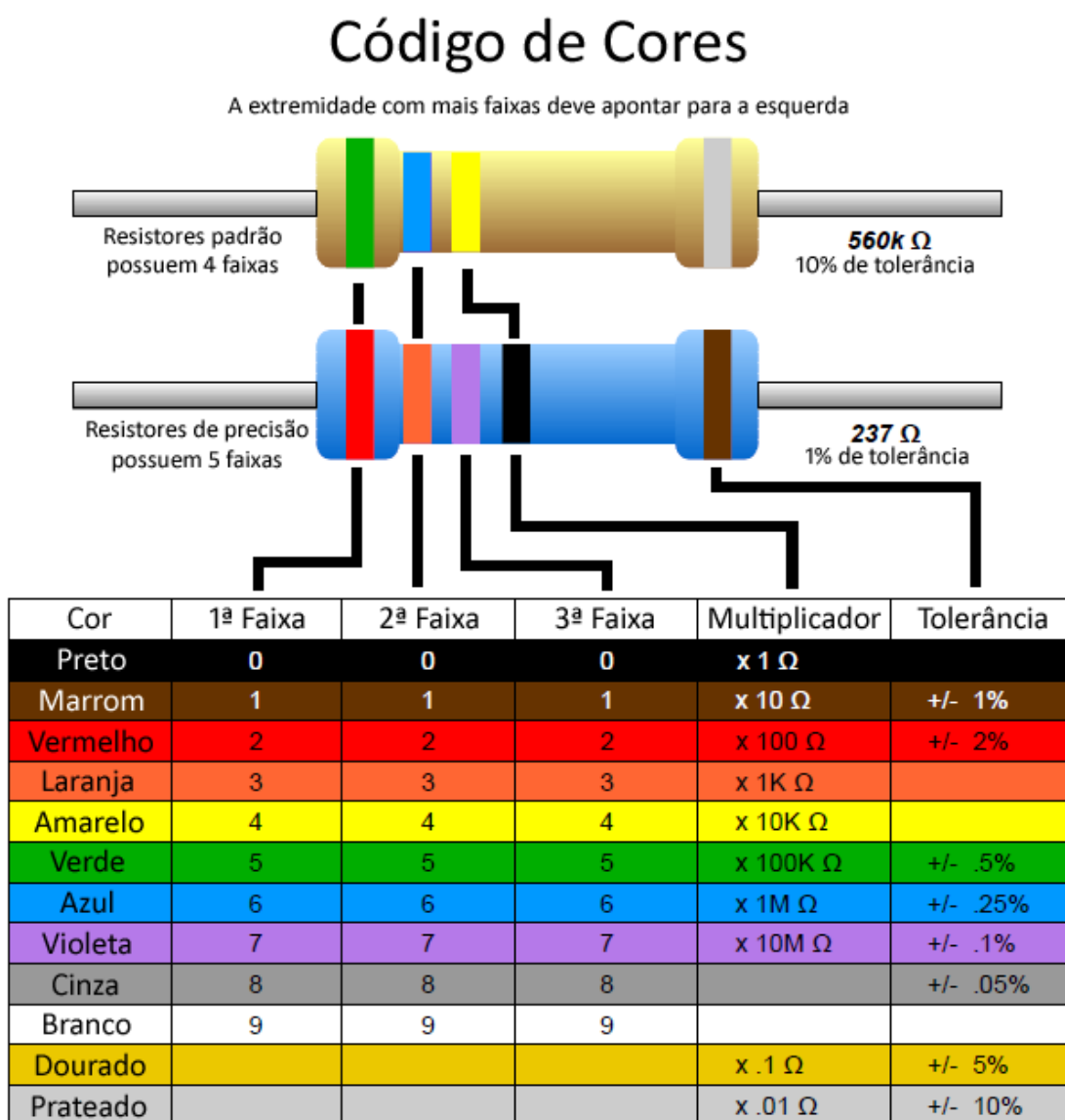


Figura 6 – Tabela de Cores dos Resistores



Para utilizar esta tabela é necessário diferenciar quem é a primeira cor e quem é a última cor de linha do resistor. A primeira linha de cor é aquela onde tem mais linhas de cores próximas. Na parte superior da Figura 7, temos 2 resistores. O primeiro tem as cores verde, azul e amarelo e portanto a sua primeira cor é verde. Logo sua última cor é a cinza. O outro resistor tem a primeira cor como vermelho (tendo as cores próximas laranja, violeta e preta) e a última cor como marrom.

Com estas cores definidas, podemos calcular o valor do resistor. Mas como?!

Simples! A última faixa corresponde à tolerância do resistor, a penúltima faixa corresponde ao fator multiplicador e as demais linhas formam o valor número do resistor. Veja o exemplo:



Primeira faixa: Laranja
Segunda faixa: Azul
Terceira faixa: Vermelho
Quarta faixa: Dourado

Agora olhando na tabela, veremos:

Primeira faixa com laranja: 3

Segunda faixa com Azul: 6

Terceira faixa com Vermelho: 100Ω

Quarta faixa com Dourado: +/- 5%

Agora vamos as contas!!!

Valor do resistor (juntar os valores antes do fator de multiplicação): 36

Multiplicando pelo fator: $36 * 100\Omega = 3600\Omega$

Fator de tolerância: $3600\Omega +/- 5\%$

LED

Os LEDs (Light Emitting Diode – diodo emissor de luz) são diodos capazes de emitir luz, como uma lâmpada, mas de baixo consumo elétrico. Eles possuem várias cores (verde, vermelho, amarelo, azul, laranja e outros). Cada um destes componentes tem uma voltagem específica de funcionamento. Alguns destes valores podem ser visualizados na tabela abaixo:

Cor do LED	Voltagem	Amperagem
Vermelho	1.6V	20mA
Amarelo	2.1V	20mA
Verde	2.1V	20mA
Azul	4.5V	20mA
Laranja	2.2V	20mA

Tabela 1 – Valores de Voltagem e Amperagem correspondentes às cores do LED

Outra característica muito importante do LED é a sua estrutura, pois ele é formado por um lado negativo e outro positivo, como apresenta a imagem abaixo:



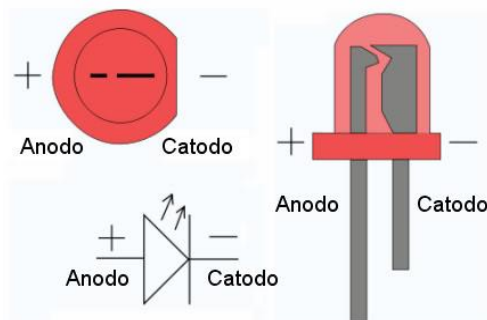


Figura 7 – Representação LED

Observe que o LED não é um círculo completo. Ele possui um lado que é abaulado e neste há o conector menor, que corresponde ao catodo (polo negativo). O outro lado é redondo e tem o conector maior, que corresponde ao anodo (polo positivo). Esta definição é importante, pois para o LED emitir luz é necessário que a corrente elétrica passe do polo positivo para o negativo, ou seja, que a corrente passe do anodo para o catodo, caso contrário o LED não emitirá luz.

OBS: No caso dos LEDs, existia a polaridade positiva (anodo) e polaridade negativa (catodo), mas no caso dos resistores isto não existe. De qualquer forma que o resistor for colocado no circuito, ele funcionará da mesma forma.

Botão Táctil

Este botão também é conhecido como push button ou botões de pressão, pois precisam estar pressionados para mudarem o estado do interruptor para aberto ou fechado. O funcionamento deste componente pode ser visualizado na Figura 8:

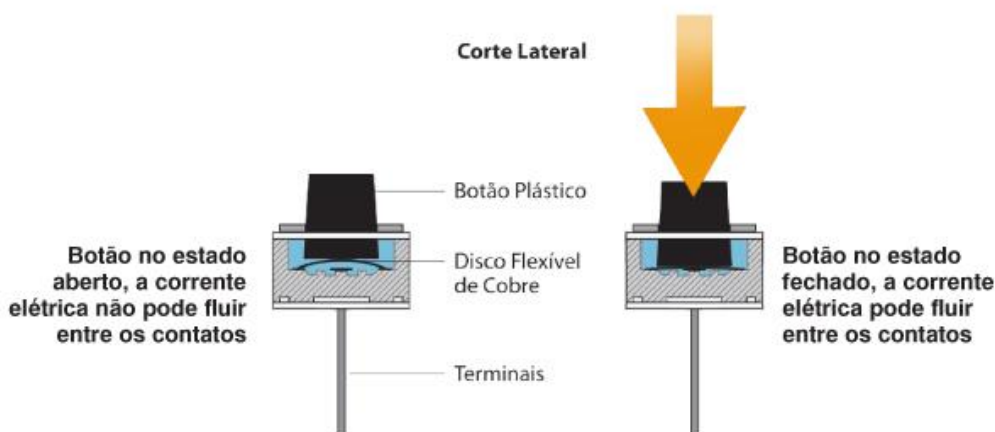


Figura 8 – Funcionamento do botão táctil

Dizemos que o estado do interruptor está fechado quando a corrente pode fluir de um contato do botão para o outro. Quando os contatos estão separados, dizemos que o estado do botão é aberto, e a corrente não pode fluir entre os contatos.



Estes botões também são chamados de interruptores tendenciosos ou momentâneos, porque depois de pressionados, eles retornam ao estado de origem (aberto ou fechado).

Existem dois tipos de botões de pressão, classificados quanto ao seu estado:

NO (abreviação de Normally Open) – este interruptor momentâneo fica normalmente aberto (desligado), mas se você pressionar e segurar o botão, o interruptor fecha (liga). Ao soltar o botão, o interruptor abre novamente. Utilizado em teclados de computadores, calculadoras, etc.

NC (abreviação de Normally Closed) – esse interruptor momentâneo fica normalmente fechado (ligado), mas se você pressionar e segurar o botão, o interruptor abre (desliga). Ao soltar o botão, o interruptor fecha novamente. Utilizado na iluminação interna das geladeiras, veículos, etc.

O botão que estamos utilizando no projeto é o botão com 4 pinos como apresentado na Figura 9:



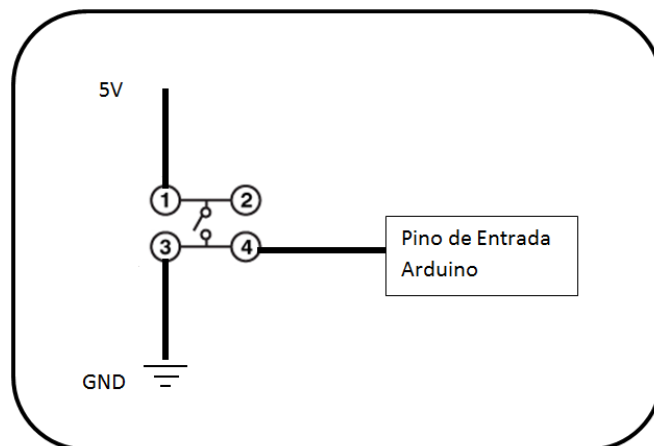
Figura 9 – Botão Tátil com 4 pinos

Observe que ele funciona como uma porta. Mas por que ter 4 pinos?

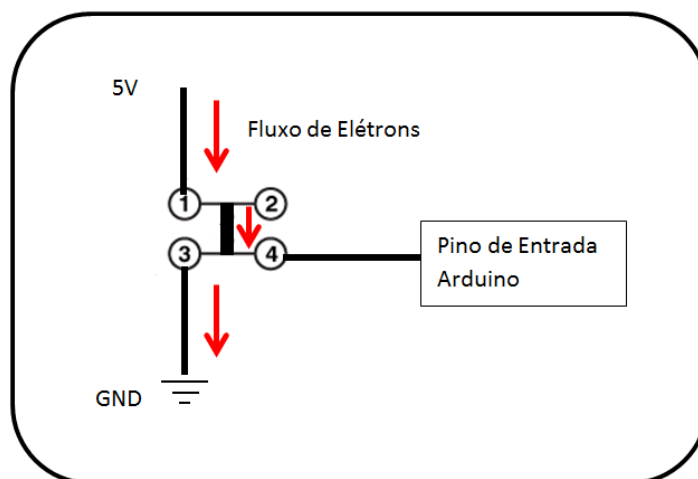
Os pinos que estão do mesmo lado (pinos 1 e 3 ou pinos 2 e 4) são utilizados para passagem de corrente, um dos terminais é conectado ao VCC e o outro terminal ao terra.

Quando estamos utilizando este botão no Galileo, usamos 3 pinos, pois dois pinos são responsáveis por fazer a passagem de corrente elétrica e um pino envia a informação do estado do botão para o Galileo, como mostra as imagens a e b abaixo:





a)



b)

Figura 10 – Funcionamento Botão Táctil

Observe que na Figura 10 a) o botão (representado pelos pontos 1,2,3 e 4) não está pressionado e assim não há corrente passando pelo botão e a leitura do pino no Galileo indica que a tensão V é zero. Já na Figura 10 b) é possível verificar que com o botão pressionado, a corrente atravessa o botão vai diretamente para o pino de terra e assim a leitura do pino no Galileo indica que a tensão V é diferente de zero.

Agora já entendemos os resistores, LEDs , protoboard e o botão táctil, fica só uma dúvida que é: quem vai alimentar o sistema?

Essa é fácil! A fonte!

Fonte

A fonte é o circuito responsável por gerar a energia necessária ao sistema para fazê-lo funcionar, ou seja, ela é quem gera a diferença de potencial elétrico para que exista uma corrente elétrica passando pelos componentes. No caso dos nossos



experimentos, a fonte será o próprio Galileo, pois ele possui uma fonte de 5V e o terra, fazendo assim uma diferença de potencial de 5V.

Curiosidade: Neste projeto foram escolhidos dois resistores para formar o projeto. Mas por que os escolhemos?!

Vamos ao primeiro resistor de 10kΩ que está ligado ao LED vermelho.

Pela tabela de tensão (Tabela 1), o LED só suporta 1,6V e uma corrente de 20mA, então é necessário utilizar um resistor para reduzir a tensão que chega ao LED, pois a fonte que o Galileo gera é de 5V. Vamos usar a lei de Ohm para encontrar o valor do resistor para esta redução:

$$V = RI$$

$$R = V/I$$

$$R = (5-1,6)/0,02$$

$$R = 3,4/0,02$$

$$R = 170\Omega$$

Então, como no material que acompanha o Galileo, não temos um resistor de 170Ω, escolhemos um resistor com valor aproximado que foi o resistor de 200Ω.

Já para o botão táctil o uso do resistor foi para garantir que a corrente que chegue ao pino de GND do Galileo seja a menor possível. Para garantir isto, a corrente passando pelo botão quando ele for apertado deve ser zero ou próximo de zero. Por isso o uso de um resistor com o valor tão grande (10kΩ = 10000Ω).

$$V = RI$$

$$I = 5/10000 = 0,0005A$$

Corrente bem baixa e assim circuito seguro!

Agora que vimos um pouquinho de todos os componentes que vamos usar, iremos estudar um pouco de programação.

Condicional IF

A função condicional IF serve como uma forma de fazer com que o código execute ou não uma instrução dependendo de uma condição. Por exemplo, imagine que você tem que escolher que roupa usar e deve considerar a possibilidade de estar frio. Caso faça frio, você coloca um casaco. Assim em linguagem de programação fica:

```
if (estiverFrio == 1){  
    useCasaco = 1;  
}
```

Garantia de estar quentinho no inverno e usando somente um if!!!

Para tornar o nosso trabalho mais divertido, vamos fazer com que o Galileo acenda um LED quando for apertado o botão. Mas para conseguir isto, precisamos seguir este diagrama:



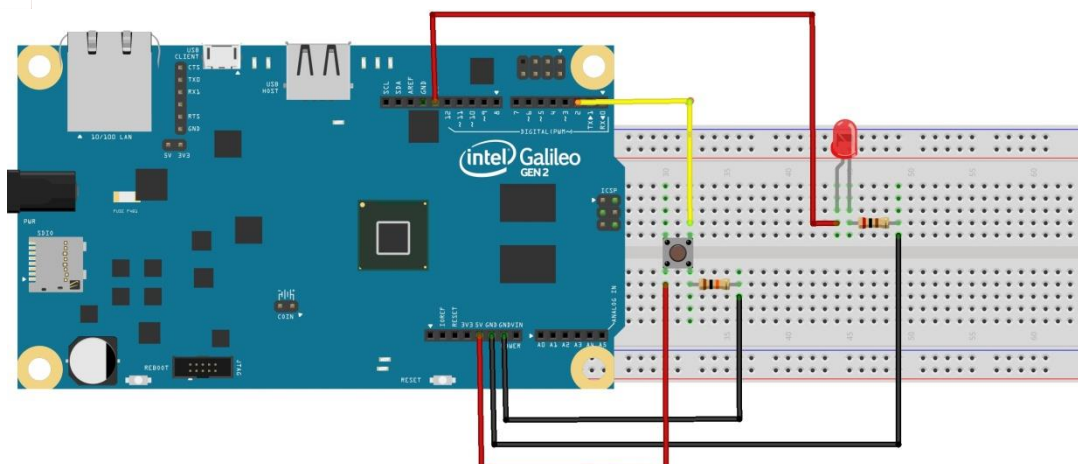


Figura 11 – Diagrama do Experimento 1

Vamos fazê-lo passo a passo. Vamos lá:

- 1- Encaixe o botão, o LED e os resistores respeitando as linhas e colunas onde devem ficar cada conector dos componentes;

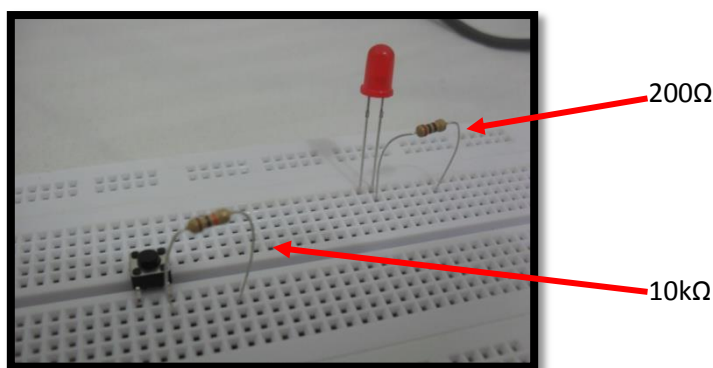


Figura 12 – Botão, LED e Resistores na Protoboard

Observe que o conector do botão está na mesma coluna que o conector do resistor de 10KΩ e o conector do LED está na mesma coluna que o conector do resistor de 200Ω.

- 2- Conectar os dois fios terra (usei os fios de cor azul) aos pinos GND do Galileo.





Figura 13 – Dois fios de conexão ligados ao GND (terra) do Galileo

- 3- Conectar os fios terra na protoboard no terminal do resistor de 10k Ω e do resistor de 200 Ω ;

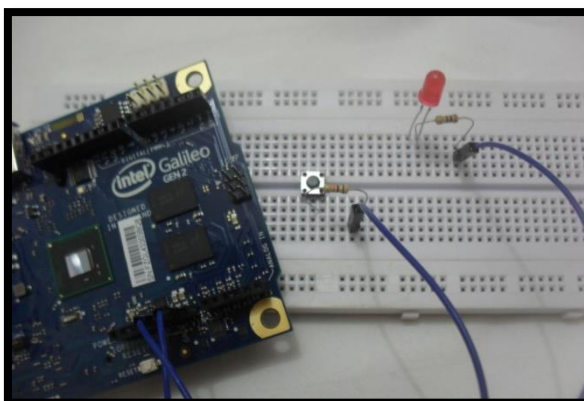


Figura 14 – Ligação fios GND com os terminais dos resistores

- 4- Conectar um dos fios vermelhos ao pino de 5V do Galileo;

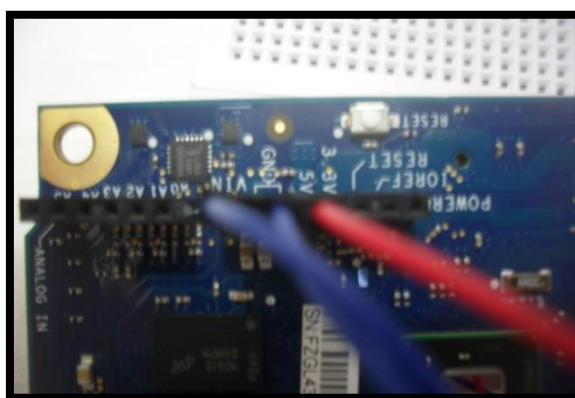


Figura 15 – Ligação fio conector ao pino de 5V do Galileo

- 5- Ligar o fio vermelho, que está ligado ao pino de 5V, ao terminal do botão;



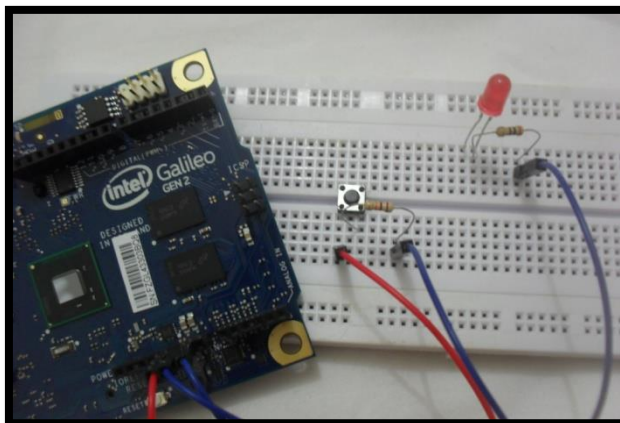


Figura 16 – Ligação fio 5V ao terminal do botão

- 6- Conectar o fio amarelo ao pino digital 2 do Galileo;



Figura 17 – Ligação do fio amarelo ao pino digital 2 do Galileo

- 7- Ligar o fio amarelo, que está conectado ao pino digital 2 do Galileo, ao terminal do botão;

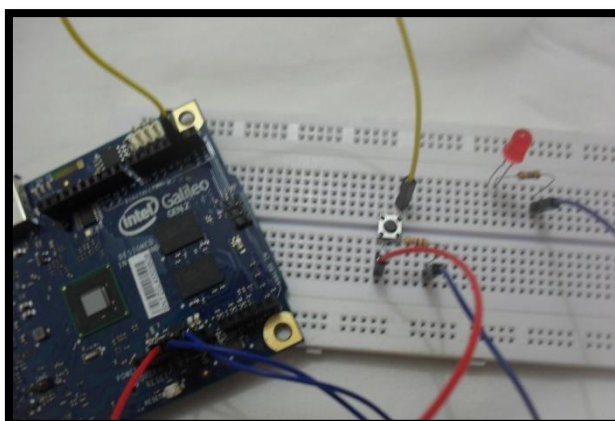


Figura 18 – Conexão do fio amarelo ao terminal do botão

- 8- Conectar o fio vermelho, que restou, ao pino 13 do Galileo;



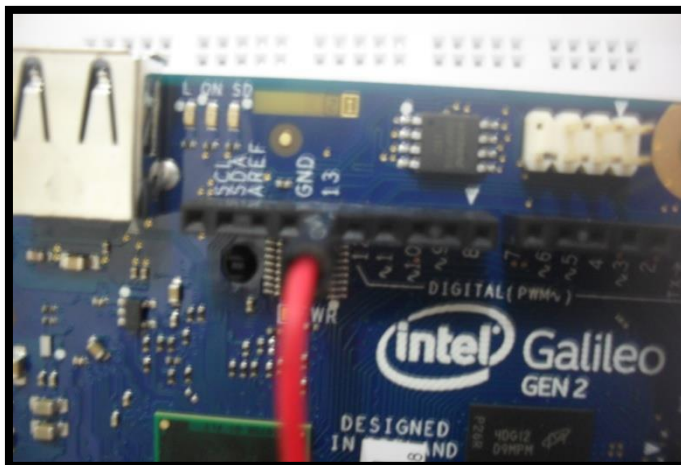


Figura 19 – Conexão fio vermelho ao pino 13 do Galileo

- 9- Ligar o fio vermelho, que está conectado ao pino 13 do Galileo, ao terminal do LED;

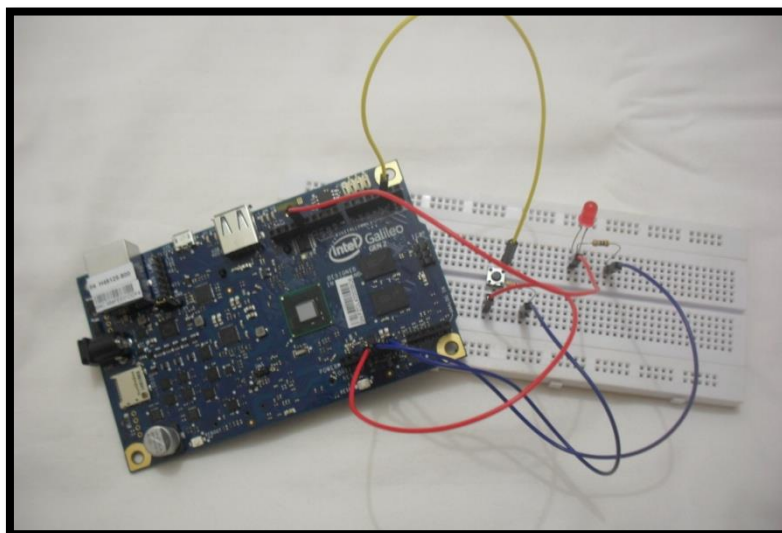


Figura 20 – Conexão fio vermelho ao terminal do LED e Sistema na versão final

Os componentes já estão todos conectados e prontos para serem utilizados. Agora, finalmente, vamos à programação do microcontrolador do Galileo!

Passo a passo:

- 1- Abra o software do Galileo. Para isto, utilize o menu iniciar do Windows;



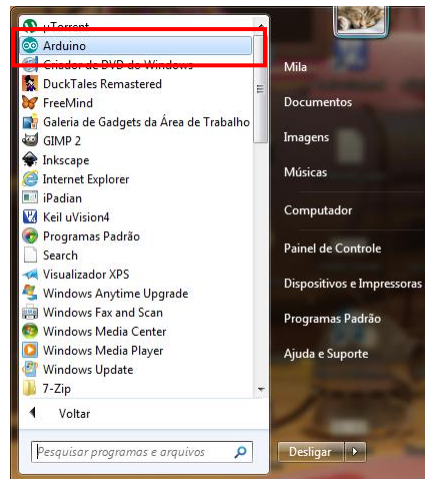


Figura 21 – Abrindo o software Galileo

2- Quando abrir o software, aparecerá a seguinte tela:

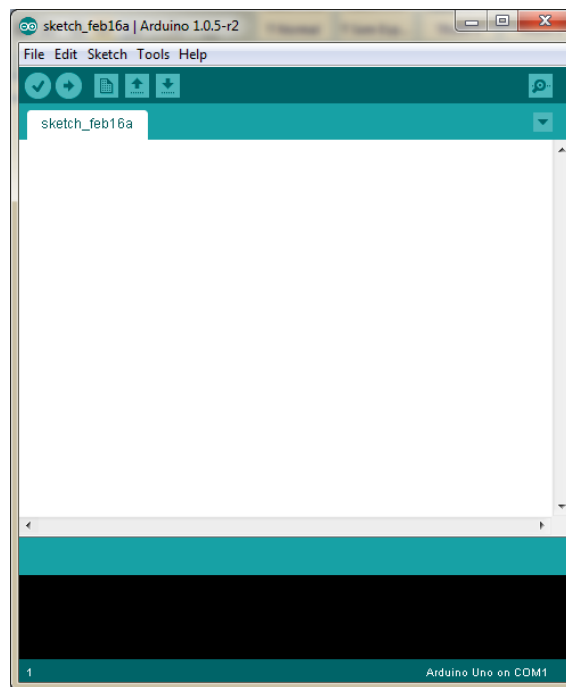


Figura 22 – Tela inicial do software Galileo

3- Agora vamos programar, copiando o código da Figura 20 para esta tela (Figura 19):




```

/* Função if */
void setup() //primeira função a ser chamada no início do programa
{
    Serial.begin(9600); //fixa a taxa de comunicação em 9600 bps
    pinMode(2, INPUT); //o pino 2 é selecionado para receber informações
    pinMode(13, OUTPUT); //o pino 13 é selecionado para transmitir
    informações
}
void loop() //todas as funções embarcadas vão ser repetidamente executadas
{
    int botao; //declarando variável botão como tipo inteiro
    botao = digitalRead(2); //a variável botão guarda o estado lógico do pino 2
    Serial.println(2); //envia para a porta serial o estado lógico do pino 2

    if(botao == 1){ //teste condicional do botão pressionado
        digitalWrite(13, HIGH);
        //caso for verdadeiro, é enviado um sinal lógico alto para o pino
        13
    }
}
}

```

Figura 23 – Código que faz o LED ligar quando o botão é pressionado

- 4- Agora que nosso código já está no software do Galileo, é hora de salvar o nosso trabalho. Clique em *File* e depois em *Save*.

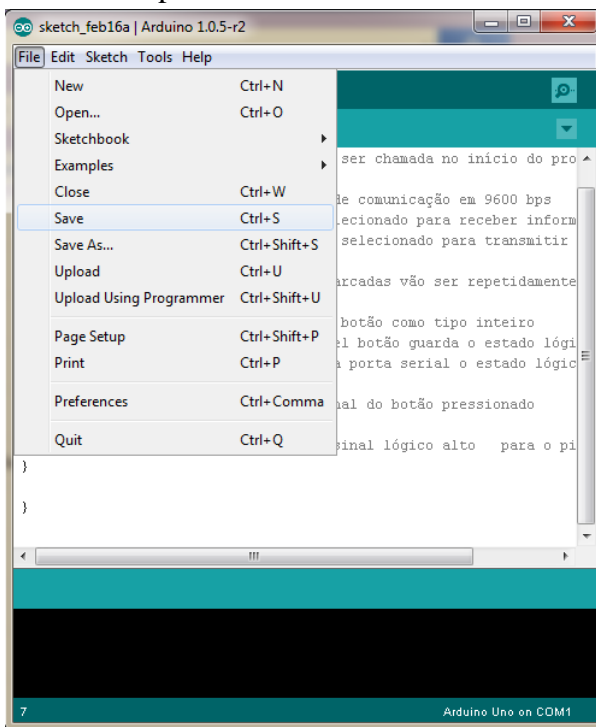


Figura 24 – Salvando trabalho no software do Galileo



- 5- Escolha um local no computador para salvar e um nome para o seu software e depois clique em salvar.

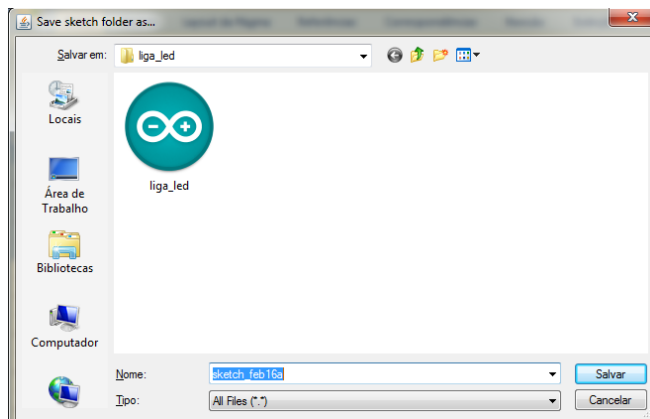


Figura 25 – Salvar trabalho software Galileo

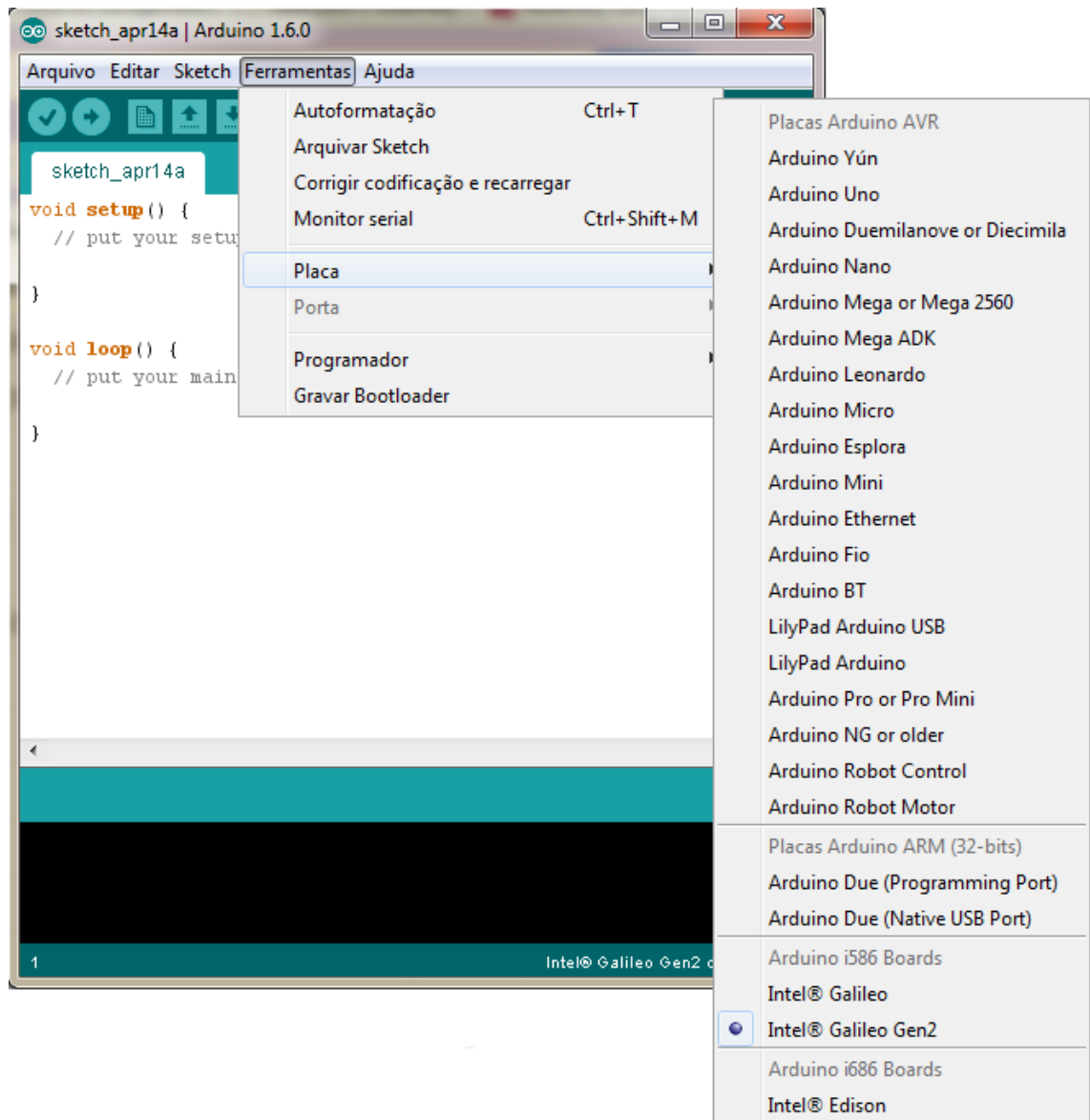
- 6- Depois de salvo, vamos carregar e executar o nosso projeto. Para isto clique em Upload.



Figura 26 – Fazendo o download do software para o Galileo

- 7- Lembre-se de selecionar a placa Galileo Gen 2. Veja:





- 8- E agora? O que aconteceu? O que o IF fez?
- 9- Vamos agora colocar um temporizador dentro do IF para ele contar 2 segundos e depois desligar o LED.
- 10- Será que conseguimos colocar o LED para desligar mais rápido e mais devagar?

