

AULA 3  
INTERNET DAS COISAS

INTERAÇÃO ENTRE MICROCONTROLADORES

IoT - Internet of Things

Modelix Robotics

# Índice

1.0. Introdução .....	2
2.0. Terceiro projeto - Central de comunicação.....	4
2.0.1. <i>O que você precisa para criar este projeto?</i> .....	4
2.0.2. <i>Introdução à novos conceitos e programação do “enviador”</i> .....	4
2.0.3. <i>Programação do “rebedor”</i> .....	7
2.0.4. <i>Conectando os microcontroladores</i> .....	7

## 1.0. Introdução

Agora que sabemos como enviar para a internet tanto um sinal pelo Dashboard como pelo microcontrolador, está na hora de juntar os dois conceitos aprendidos para uma aplicação real - o Alarme de luz.

está na hora de criar nossa primeira aplicação real - o Alarme de Luz. Para criar este projeto, teremos que aprender um novo conceito: como programar a interação virtual entre dois microcontroladores.

## 2.0. Terceiro Projeto - Alarme de Luz

Neste projeto, iremos criar um alarme que avisa quando uma luz acende em um ambiente que você não está. O projeto pode ser útil para economizar energia, assim como uma medida de segurança.

### 2.0.1. O que você precisa ter para criar este projeto?

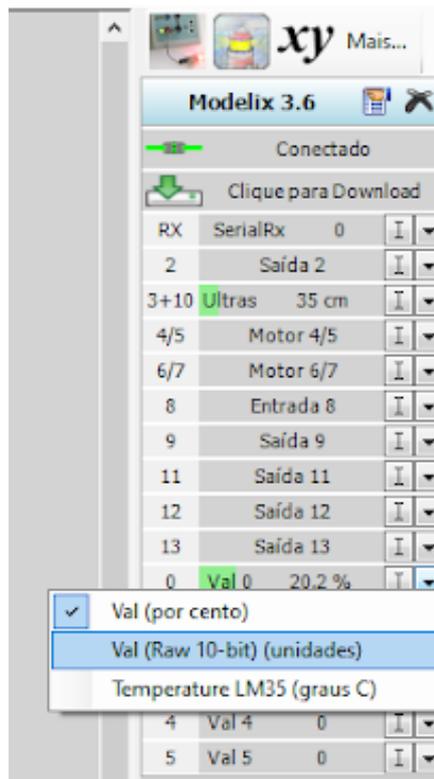
1. Dois computadores conectados à Internet. É necessário que o sistema operacional seja Windows.
2. Dois microcontroladores com cabos USB para conexão com os computadores.
3. Um ou mais sensores ou atuadores (LED, som, motor). Note que alguns microcontroladores já possuem tais embutidos.
4. Um software para programar o microcontrolador, podendo ser um IDE (Ambiente de Desenvolvimento Integrado) de C++ genérico ou um software educacional de Fluxograma como o Modelix System Starter (recomendado). Para ter mais informações e baixar o programa, acesse <https://www.modelix.com.br/software-de-programacao-robotica-1>.
5. Cadastro de usuário na Plataforma IoT da Modelix, assim como a realização do download do programa "broker" (não requer instalação) **em ambos os computadores**. Para acessar ambos, entre em <https://www.modelix.com.br/c%C3%B3pia-modelix-system-simulador>.
6. Ter realizado a Aula 2 do Curso Modelix IoT.

Agora mãos à obra!

## 2.0.2. Introdução à novos conceitos e programação do “enviador”

Para a comunicação entre microcontroladores ocorrer, é necessário que um microcontrolador emita um sinal que o segundo microcontrolador entenda. No ambiente IoT, isto é possível através da emissão de um código ASCII, que aprendemos sobre nas aulas anteriores.

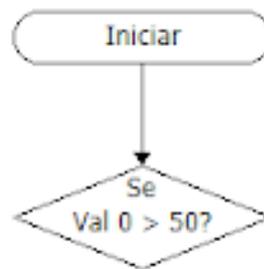
1. Para iniciarmos, conecte o microcontrolador ao computador e então ao software Modelix System Starter. Após a conexão, mude a “saída 0” para SerialRx.
2. Este microcontrolador será o “enviador” do sinal. Logo, sua função é enviar um sinal ao reconhecer uma mudança na luminosidade.
3. Como maneira de reconhecer a luminosidade, precisamos configurar o sensor, este que já está embutido nos microcontroladores Modelix.
4. Na aba de saídas e entradas, este está localizado na entrada analógica Val 0. Clique na pequena seta da linha Val 0, e mude de Val (por cento) para Val (Raw 10-bit)(unidades).



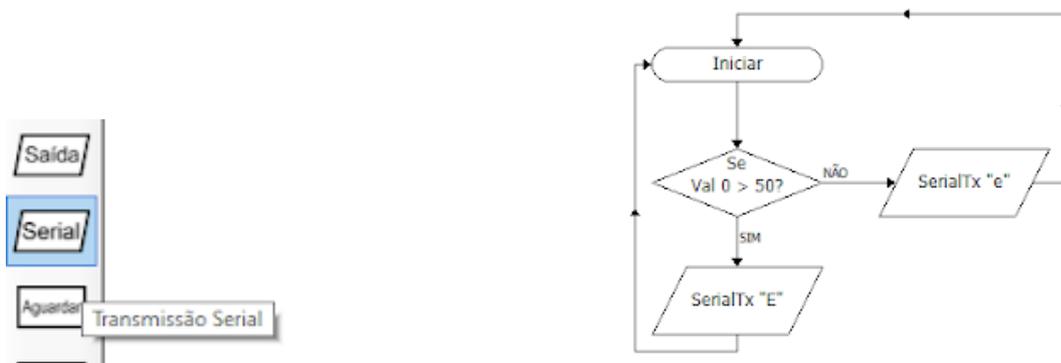
5. Perceba que agora Val 0 apresenta um valor numérico. Este valor é relativo a luminosidade captada pelo sensor, não possuindo unidade. Tente apagar e acender a luz, e observe o padrão de mudança no número. Você verá que, sem luz, o valor cai para uma faixa 10-50, enquanto com luz se mantém perto de 200-300. Estes valores não precisam ser exatos, mas

obtenha uma referência numérica de luminosidade para os próximos passos (por exemplo: se o valor está acima de 50, eu sei que a luz está acesa).

- Inicie um programa, criando um bloco Iniciar seguido de um bloco de decisão. O bloco decidirá o que acontecerá baseado em dados coletados, sendo neste caso a luminosidade. Assim sendo, configure para que este se refira à faixa de valores onde a luz estará acesa.



- Agora precisamos configurar para que, caso seja detectado alta luminosidade, um sinal seja enviado pelo controlador. E caso não seja, outro sinal deve ser enviado. Insira dois blocos de Transmissão Serial ao redor do bloco de decisão, com valores de E (maiúsculo) assim como e (minúsculo). Complete o programa conforme a imagem abaixo.



- Por mais que programamos o envio da mensagem "E" e "e", vamos lembrar que este valor é um código ASCII, contendo um valor numérico equivalente. Neste caso, será o seguinte:

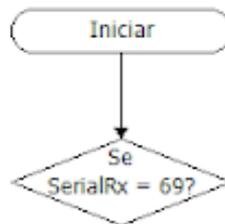
Letra	E	e
Valor Numérico	69	101

É interessante lembrar que poderíamos usar qualquer letra dentro do código ASCII (visto na Aula 1). Somente seria necessário saber o valor numérico equivalente da letra ou símbolo escolhido.

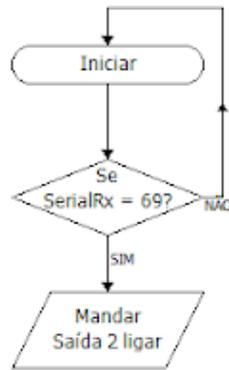
9. Terminamos de programar o “enviador”! Faça o download do programa para o microcontrolador, desconecte-o do computador e repita o passo 1 desta secção com o outro controlador.

### 2.0.3. Programação do "rebedor"

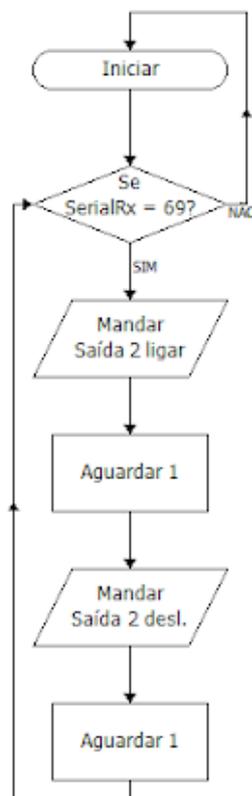
1. Diferente do primeiro controlador, este terá a função de interpretar o sinal gerado pelo outro que está correndo pelo servidor (através da tecnologia IoT). Assim sendo, o primeiro passo é programar tal interpretação.
2. Inicie um novo programa, colocando um bloco Iniciar seguido de um bloco de decisão abaixo. Programaremos este bloco para reagir se SerialRx = 69 (que é equivalente a “E”).



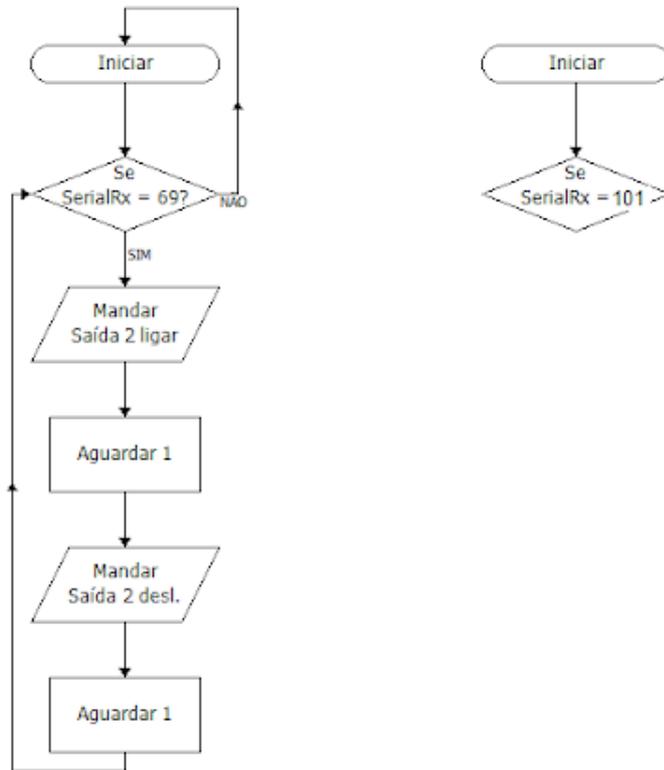
3. Lembrando da programação do “enviador”, sabemos que se o sensor perceber que a luz está acesa, ele enviará o sinal E (69). Então vamos criar uma saída para o que deve acontecer caso a luz esteja acesa. Coloque um bloco de saída que liga a saída 2, que é um BIP. Se preferir, pode renomear a saída para BIP, conforme aprendemos nas aulas anteriores.



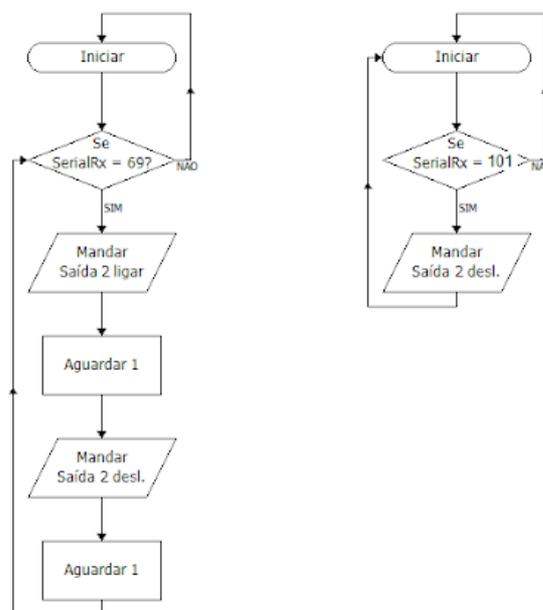
4. Agora, vamos fazer um ajuste no BIP. Na programação atual, ele irá tocar continuamente. Iremos programar para que ele seja oscilante, como um alarme real.



5. Por fim, precisamos programar para que o nosso "receptor" interprete mais uma opção de sinal enviado pelo "enviador". Este sinal (que configuramos nos passos 7-8 da seção 2.0.1), é o "e", equivalente ao valor numérico 101. No mesmo arquivo, realizaremos o mesmo procedimento da programação que fizemos para a luz acesa, inserindo um bloco iniciar seguido de um bloco de decisão.



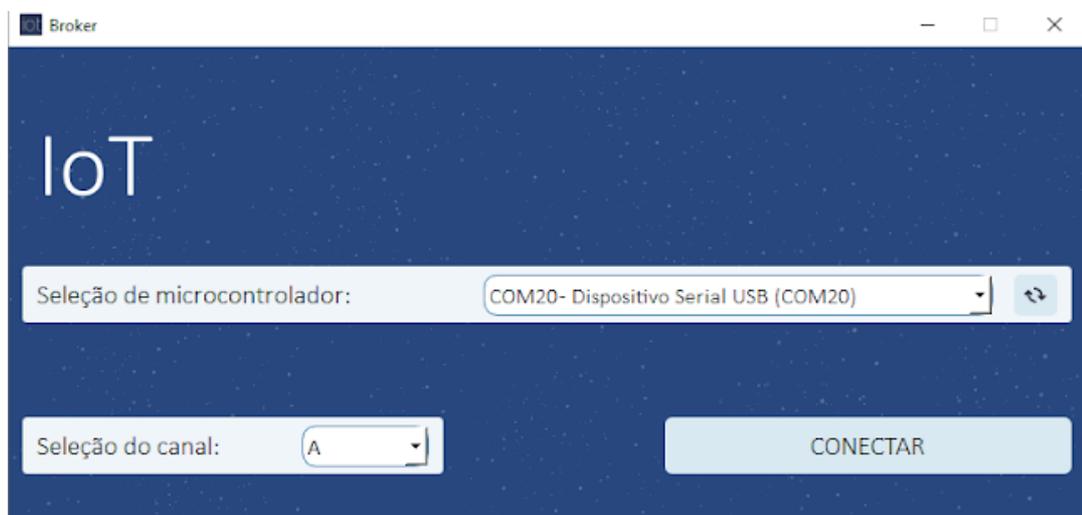
6. Agora vamos lembrar o que o sinal 101, ou “e”, significa. De acordo com o programa do “enviador”, este sinal é enviado quando Val 0 é menor que 50, ou seja, quando a luz está apagada. Assim, iremos configurar para que o alarme permaneça desligado caso isto aconteça.



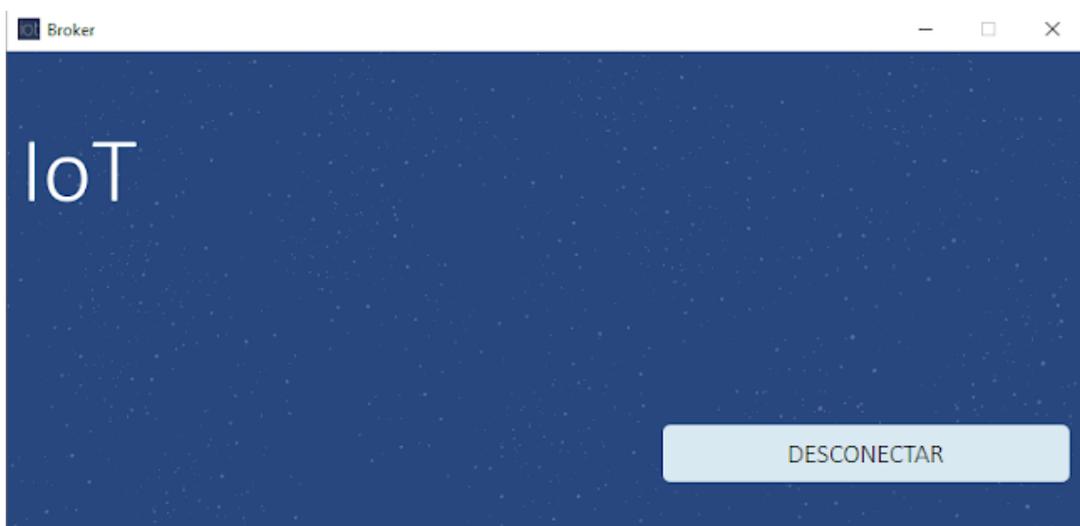
7. Pronto! Nosso programa está pronto. Basta realizar o download para o microcontrolador, e fechar o Software.

#### 2.0.4. Conectando os microcontroladores

1. Tendo os microcontroladores prontos para uso, só basta conectá-los pelo Modelix Iot Broker. Vá até o computador onde o “enviador”, que também podemos chamar de detector de luz, estará conectado.
2. Abra o aplicativo Modelix Iot Broker. Caso não o tenha, realize o download pelo link <https://www.modelix.com.br/c%C3%B3pia-modelix-system-simulador>
3. Com todos os Softwares Modelix fechados, abra o programa. Inicie a conexão escolhendo o microcontrolador e clique em “Conectar”.



4. Quando os campos de preenchimento desaparecerem, o “enviador” já estará funcional.



5. Agora só nos resta instalar o alarme no local onde queremos ser avisados, em qualquer lugar do mundo que tenha internet!
6. No computador onde será conectado o alarme, repita o procedimento dos passos 2-3. É **muito importante** que você utilize o mesmo usuário e senha inseridos no computador do “enviador”. Caso contrário, o sistema não funcionará.
7. Quando os campos de preenchimento desaparecerem, o "recebedor" estará funcional.
8. Seu alarme está pronto! Agora, quando a luz for acesa no ambiente que está o “enviador”, um alarme tocará no “recebedor”. O projeto pode ser útil para um ambiente onde a luz estará quase sempre apagada. Você será avisado se alguém estiver lá e acender a luz, assim como se alguém esquecer a luz acesa!