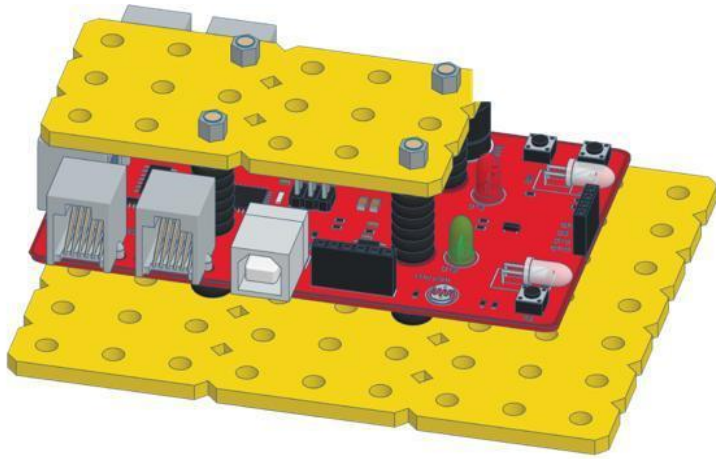


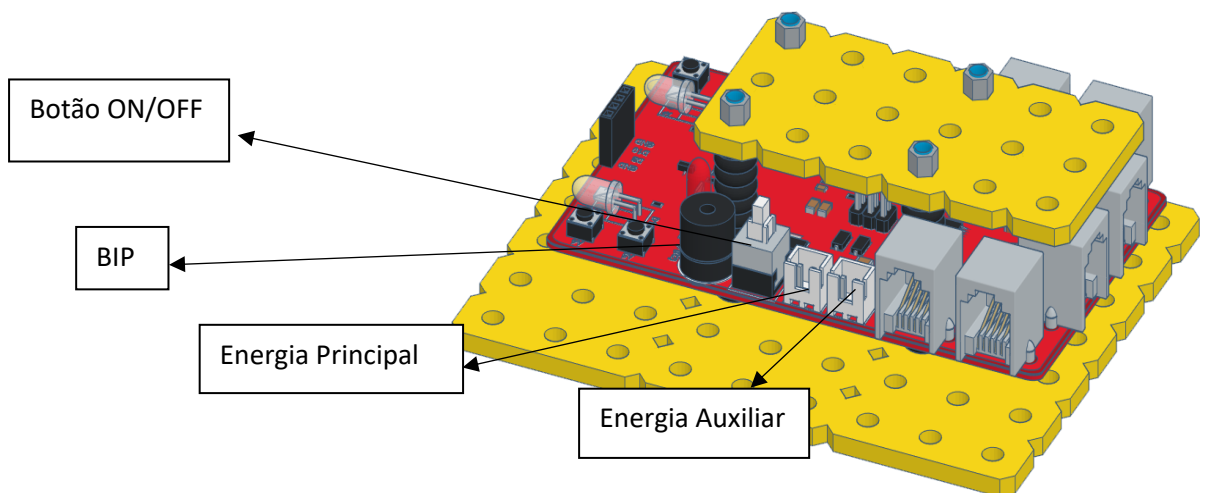
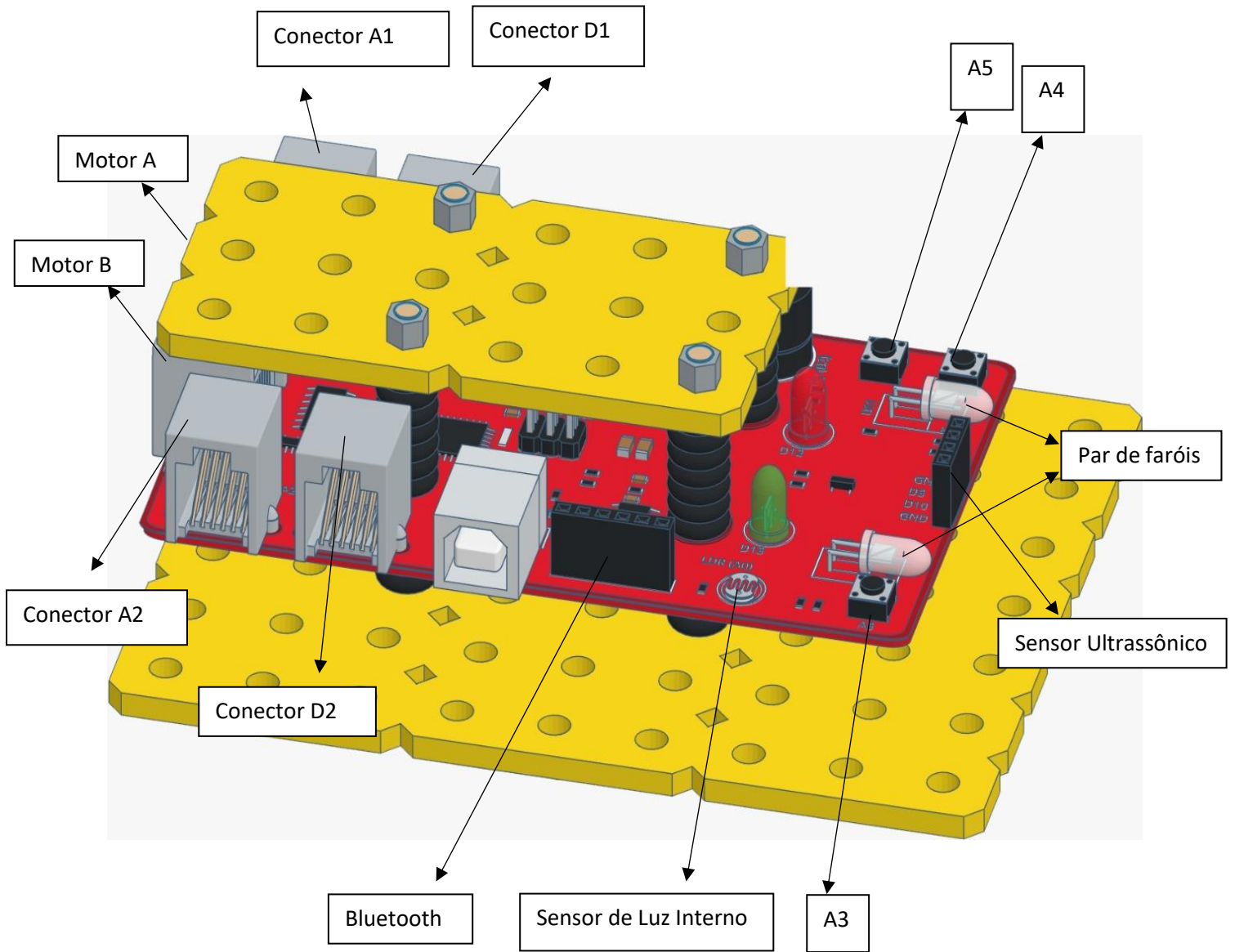
Guia de Uso do Aplicativo KALCODE



Sumário

1. Microcontrolador FUND1B RJ	3
1.1 Conexões no Microcontrolador FUND1B RJ	4
1.2 Sensores e Sinais	5
2. Execução dos Comandos	5
3. Conexão com o Bluetooth	7
4. Menu de Opções	8
4.1. Modo programação	8
4.2. Nova rotina	9
4.3. Guardar rotina	9
4.4. Carregar rotina	9
4.5. Modo controle remoto	9
4.6. Modo offline	9
5. Aplicações	10
5.1. Modo Programação	10
5.1.1. Comandos Básicos	10
5.1.1.1. Atuadores	11
5.1.1.2. Pausar	12
5.1.1.3. Recomeça	13
5.1.1.4. Motores	14
5.1.1.5. Marcador	14
5.1.1.6. Se	15
5.1.1.7. Expressão	19
5.1.2. Demais Comandos	21
5.1.2.1. Atuador	21
5.1.2.2. Motor	21
5.1.2.3. Ir para	22
5.1.2.4. Como deletar um bloco de comando	23
5.1.2.5. Falar	24
5.1.2.6. Mensagem	25
5.1.2.7. Interruptor	27
5.1.2.8. Gatilho	28
5.1.2.9. Ultrassônico	31
5.1.2.10. PWM	35
5.2. Modo controle remoto	36
6. Gabarito	37

1. Microcontrolador FUND1B RJ



1.1. Conexões no Microcontrolador FUND1B RJ

Os sensores externos são conectados nos conectores A1 e/ou A2.

Os atuadores externos são conectados nos conectores D1 e/ou D2.

Na Energia Auxiliar deve ser conectado pilhas para a alimentação dos motores, caso estes estejam conectados no projeto atual.

Na Energia Principal as pilhas são conectadas para alimentar o microcontrolador como um todo. Alternativamente pode-se usar cabo USB com o computador ou notebook .

Componentes externos são os que podem ser conectados ao microcontrolados e os componentes internos estão integrados no microcontrolador.

Os atuadores deste kit são:

- Leds internos (Verde, Vermelho e Par de Faróis);
- Bip interno;
- Led externo Vermelho;
- Led externo Amarelo;
- Led externo Verde;
- Par de Motores;
- Bip externo.

Os sensores são:

- Sensor externo de Obstáculo infravermelho;
- Sensor externo de Luz;
- Sensor interno de Luz;
- Sensor de Ímã externo;
- Sensor de Toque externo;
- Sensor de Toque interno (A3, A4, A5);
- Sensor Ultrassônico.

1.2. Sensores e Sinais

Há dois tipos de sinais que são enviados pelos sensores – o analógico e o digital.

No kit há dois sensores digitais: o Sensor de Toque (botão), Sensor de Ímã e os botões sobre o FUND1B RJ A3, A4 e A5

O sinal digital possui dois estados: ligado ou desligado. Para este aplicativo considere que o sinal desligado é representado por uma faixa de valores de 0 até 50. O sinal ligado é representado por uma faixa de valores superior a 50 até 100. Desta forma, ao acionar um sensor digital, um valor maior que 50 será enviado (sinal ligado) e ao desativá-lo um valor menor que 50 será enviado (sinal desligado).

Os sensores analógicos são: Sensor de Luz, Sensor de Obstáculo, Ultrassônico e o Sensor de Luz sobre o FUND1B RJ.

O sinal analógico possui uma faixa de estados de 0 a 100, pois a informação é quantificada. Por exemplo: um Sensor de Luz quantifica o quanto de luz está sendo incidida e quanto maior o valor mostrado, maior é a luminosidade sobre o sensor.


Nos itens Se, Interruptor e Gatilho serão dados exemplos.

2. Execução dos Comandos

A programação deste aplicativo é feita através de bloquinhos de comandos e cada um deles representa um comando específico. Abaixo encontra-se um programa:

Comandos :



Assim que o Play  for pressionado o processador executa um bloco de cada vez, desta forma os comandos serão executados linha a linha. Neste guia será usado o termo *leitura do programa* para fazer referência a qual bloco o processamento se encontra em dado instante.

Execução do programa acima:

O bloquinho em que a leitura do programa se encontra é marcado com a cor azul claro, portanto inicialmente o farol é ligado através do primeiro bloquinho:

Comandos :

The image shows three stacked command blocks. The top block is light blue and contains an actuator icon, the text 'Atuador', a dropdown menu with 'farol', and a dropdown menu with 'Ligar'. The middle block is dark blue and contains a pause icon, the text 'Pausar', a text input field with '2', and the text 'segundos'. The bottom block is pink and contains an actuator icon, the text 'Atuador', a dropdown menu with 'farol', and a dropdown menu with 'Desligar'.



A leitura do programa avança para o segundo bloco, que deixa passar dois segundos sem que qualquer outra ação seja executada:

Comandos :

The image shows three stacked command blocks. The top block is pink and contains an actuator icon, the text 'Atuador', a dropdown menu with 'farol', and a dropdown menu with 'Ligar'. The middle block is light blue and contains a pause icon, the text 'Pausar', a text input field with '2', and the text 'segundos'. The bottom block is pink and contains an actuator icon, the text 'Atuador', a dropdown menu with 'farol', and a dropdown menu with 'Desligar'.



Após a ação acima ser executada, a leitura do programa segue para a próxima linha ou bloco de comando, desligando o farol que estava aceso.

Comandos :

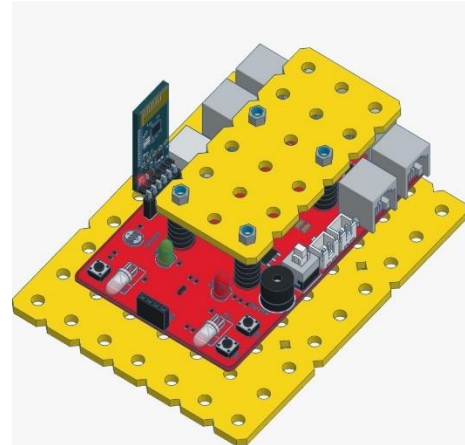
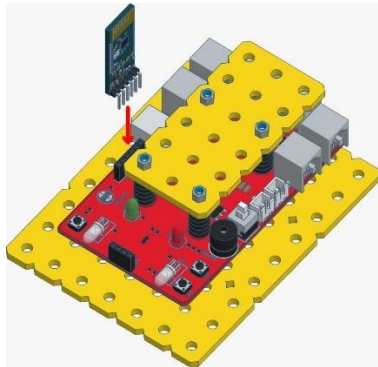
The image shows three stacked command blocks. The top block is pink and contains an actuator icon, the text 'Atuador', a dropdown menu with 'farol', and a dropdown menu with 'Ligar'. The middle block is dark blue and contains a pause icon, the text 'Pausar', a text input field with '2', and the text 'segundos'. The bottom block is light blue and contains an actuator icon, the text 'Atuador', a dropdown menu with 'farol', and a dropdown menu with 'Desligar'.



3. Conexão com o Bluetooth

Com o Bluetooth já conectado no microcontrolador FUND1B RJ abra as configurações do Bluetooth do celular, ative-o e pareie-o com o MODELIX BLUETOOTH que aparecerá nas opções possíveis para conexão.

Em seguida abra o aplicativo KALCODE.



Verifique se no bloco abaixo demarcado pela cor verde há o nome do dispositivo que foi pareado com o celular. Neste caso, o celular foi pareado com o dispositivo que se chama MODELIX BLUETOOTH e este nome está escrito no bloco. Caso não seja o mesmo nome, clique neste bloco para selecionar o dispositivo correto.

Por fim clique em **Conectar**.



4. Menu de opções

Para acessar ao menu, arraste o dedo partindo do canto esquerdo da tela para a direita.

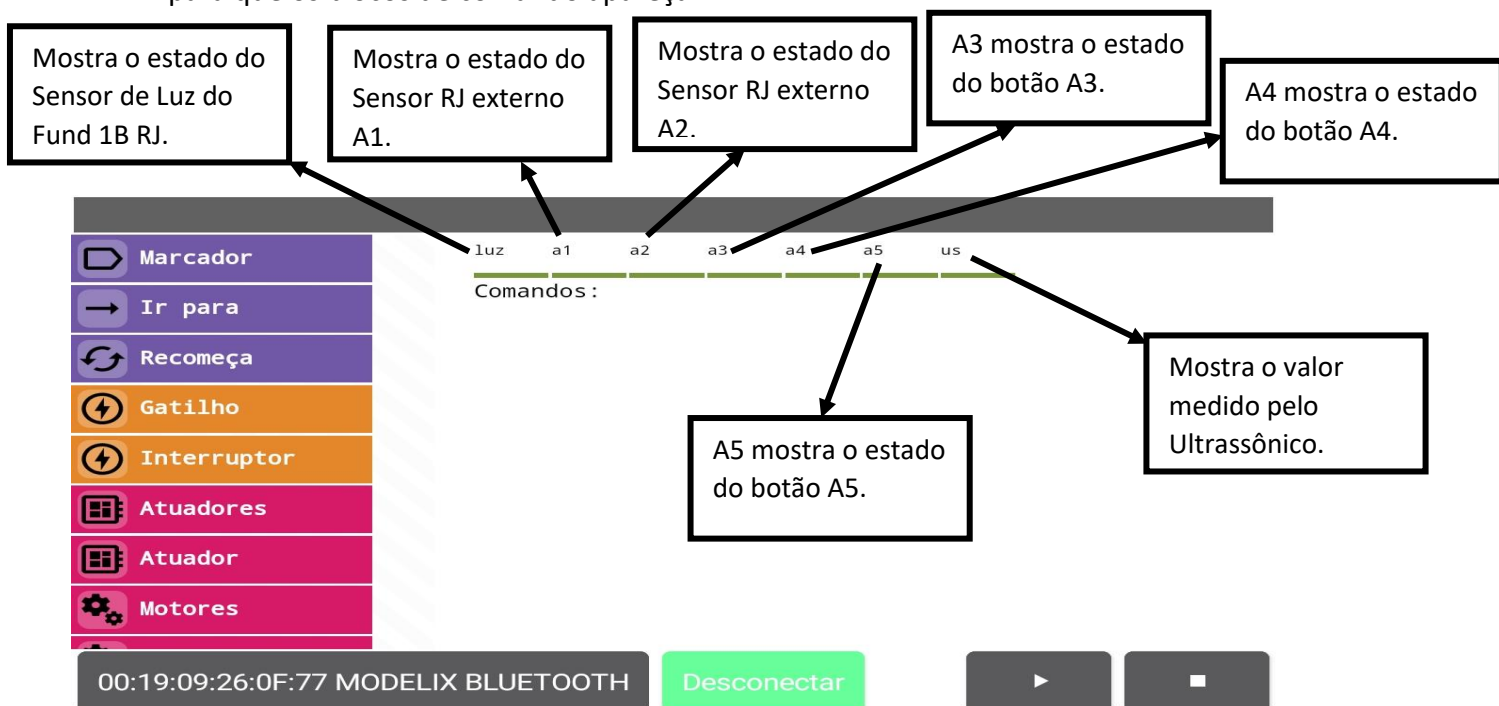
Há cinco listagens: Nova rotina, Carregar rotina, Guardar rotina, Modo programação, Modo controle e Modo offline.



4.1. Modo programação

Consiste na criação de um programa por meio de blocos. O programa que está sendo feito no momento permanece neste modo.

Antes de iniciar a programação é necessário clicar em **Conectar**, caso ainda não o fez, para que os blocos de comando apareçam.



A leitura dos valores dos sensores começa a ser feita após o Play.

4.2. Nova rotina

É usada para criar um novo programa, desta forma a interface será completamente limpa.

4.3. Guardar rotina

É usada para salvar o programa. O nome do arquivo tem o formato da data e hora atual, para alterá-lo clique no texto e digite o nome desejado. Por fim, clique em “Salvar”.

4.4. Carregar rotina

Onde se encontram todos programas salvos. Selecione o programa e clique em “Carregar” para abri-lo.

4.5. Modo controle remoto

Nenhuma programação é realizada. A interface do celular é utilizada como um meio para acionar ou controlar um mecanismo ou dispositivo à distância.

4.6. Modo offline

A lista dos blocos de comandos e suas utilizações apareceram com o clique em **Conectar**, entretanto é possível fazer isso sem se conectar ao microcontrolador utilizando apenas o aplicativo isoladamente. Para isso, assim que o aplicativo for iniciado, vá ao menu de opções e selecione o **Modo offline**. É possível construir as rotinas, salvá-las e carrega-las para possíveis alterações.

5. Aplicações

Serão demonstrados usos básicos dos dois modos.

Após escolher algum modo ou até mesmo ao criar uma nova rotina, no canto inferior da tela aparecerão as opções:



O símbolo azul permite que o aplicativo se conecte ao Bluetooth do microcontrolador.

Para executar a programação dê "Play" e para interrompe-la dê "Stop".

"6DF56F5 5D5SD" se refere ao dispositivo Bluetooth conectado ao microcontrolador.

5.1. Modo programação

Após ter clicado em "Conectar" aparecerá uma lista de bloquinhos de comandos com uma série de opções. Para deslizar para baixo ou cima mantenha o dedo à direita minimamente distante dos bloquinhos e deslize.

Para usar o bloquinho, pouse o dedo sobre o mesmo, aguarde frações de segundos e arraste-o para a área de comandos.

5.1.1. Comandos Básicos

Considera-se que de todos os comandos sete são fundamentais para a programação neste aplicativo: **atuadores**, **pausar**, **recomeçar**, **motores**, **marcador**, **se** e **expressão**.

Nesta etapa cada um deles será abordado com exemplos básicos.

5.1.1.1. Atuadores

Por meio deste bloquinho é possível ligar ou desligar atuadores externos e/ou internos simultaneamente.

Atuadores

- farol — se selecionado, os dois LEDs brancos acendem.
- verde — se selecionado, o LED externo verde acende.
- vermelho — se selecionado, o LED externo vermelho acende.
- bip — se selecionado, o BIP externo é acionado.
- 1 — aciona o atuador externo conectado em D1.
- 2 — aciona o atuador externo conectado em D2.

Exemplo:

Conecte um LED no Conector D1.

Comandos :

Atuadores

- farol
- verde
- vermelho
- bip
- 1
- 2

→ LEDs brancos e LED em D1 são acesos.

Atuadores

- farol
- verde
- vermelho
- bip
- 1
- 2

→ LEDs brancos e LED em D1 são apagados. (basta desmarcar o que estava marcado)

Os dois LEDs brancos e o LED em D1 serão acesos e imediatamente serão desligados.

5.1.1.2. Pausar

Este bloquinho impede que a leitura do programa continue por uma quantidade de segundos determinada pelo usuário. Este conceito é igual ao comando “aguardar” ou “delay” em outras linguagens de programação.

Alterando o exemplo anterior:

Comandos :

The diagram illustrates the state of actuators and a pause block in a program. It consists of three panels:

- Top Panel (Pink):** Titled "Atuadores", it contains a list of actuators with checkboxes: farol, verde, vermelho, bip, 1, and 2. An arrow points from this panel to the text: "LEDs brancos e LED em D1 são acesos."
- Middle Panel (Blue):** Titled "Pausar", it contains a clock icon, a text input field with the value "1", and the label "segundos".
- Bottom Panel (Pink):** Titled "Atuadores", it contains a list of actuators with checkboxes: farol, verde, vermelho, bip, 1, and 2. An arrow points from this panel to the text: "LEDs brancos e LED em D1 são apagados. (basta desmarcar o que estava marcado)".

Os LEDs brancos e o LED conectado em D1 são acionados. Em seguida a leitura do programa avança para o bloco Pausar de um segundo, isto significa que o programa permanecerá parado por um segundo neste bloco. Passado um segundo, a leitura do programa avança e os componentes são desligados.

5.1.1.3. Recomeça

É um bloquinho de repetição, sempre que a leitura do programa passar por ele o programa será direcionado para a primeira linha do programa e toda sequência abaixo dela será executada novamente.

Exemplo:

No programa abaixo, o farol, LED verde e LED vermelho permanecerão acesos por 1 segundo, em seguida permanecerão apagados por 1 segundo e assim que a leitura do programa chegar no bloco Recomeça o programa será repetido.

Comandos :

The image shows a vertical stack of five programming blocks. The first block is pink and titled 'Atuadores', containing a list of actuators with checkboxes: 'farol~' (checked), 'verde' (checked), 'vermelho' (checked), 'bip' (unchecked), '1' (unchecked), and '2~' (unchecked). The second block is cyan and titled 'Pausar', with a timer set to '1 segundos'. The third block is pink and titled 'Atuadores', with the same list of actuators as the first block, but all checkboxes are unchecked. The fourth block is cyan and titled 'Pausar', with a timer set to '1 segundos'. The fifth block is purple and titled 'Recomeça', representing a loop command.

O bloco Recomeça permite que esta sequência não seja executada apenas uma vez, mas infinitas ou até o momento em que o Stop for selecionado.

5.1.1.4. Motores

Para usar os motores é necessário colocar pilhas na caixa de pilhas e conectá-la na Energia Auxiliar, localizado ao lado do Conector D1. O botão ON/OFF deve ser pressionado para que ocorra a distribuição de energia da pilha para o motor.

Os dois motores são controlados de forma a desligá-los ou ligá-los. Também é possível controlar sua velocidade de giro arrastando o cursor preto para a direita, aumentando-a ou para a esquerda, reduzindo-a.

Exemplo:

O motor A será acionado no sentido de giro horário e o B no sentido de giro anti-horário, ambos com a velocidade média. Após dois segundos serão desligados.

Comandos :



5.1.1.5. Marcador

Usado para referenciar determinado trecho de código. É possível utilizar dez marcadores.

5.1.1.6. Se

Este bloquinho é responsável pela tomada de decisão do programa. Se uma dada condição for verdadeira, seja esta sobre variáveis ou sensores, o programa trilhará um determinado caminho, caso contrário tomará o outro caminho. Esta tomada de decisão ocorre no momento em que a leitura do programa estiver no próprio bloco Se (decisão).

As opções de preenchimento são: “...” ou os Marcadores de 1 a 10.

Se for utilizado “...”, o programa avança para a linha seguinte e se for usado o marcador o programa avança para o trecho referenciado.

Comandos :




Em “Se” é digitado uma expressão que se deseja verificar relacionada a variáveis ou sensores.

Este bloco também trabalha com múltiplas expressões lógicas com o uso dos operadores lógicos.

E	&&
OU	
NEGAÇÃO	!=

É aconselhável que a expressão a ser digitada seja escrita com caracteres minúsculos e que não tenha nenhum espaço entre elas, entretanto as possibilidades a seguir também são possíveis:



Exemplo com sensor digital:

Se o botão a5 é pressionado a leitura do programa avança para a sequência de comandos referenciada pelo marcador 1, caso contrário a leitura do programa avança para a sequência de comandos referenciada pelo marcador 2.

Desta forma, tem-se que ao manter o botão A5 pressionado o LED vermelho, verde e farol permanecem acesos, pois o sinal enviado é maior que 50. Ao soltar o botão os atuadores desligam e o sinal enviado é menor que 50.

O bloco Recomeça tem por função retornar a leitura do programa para a primeira linha de comando, verificando se o botão A5 continua apertado ou não.

Comandos :

The image shows a vertical stack of programming blocks:

- Se** (blue): Condition `a5 > 50`. **Vai para:** 1, **Se não:** 2.
- Marcador** (purple): 1.
- Atuadores** (pink):
 - farol~
 - verde
 - vermelho
 - bip
 - 1
 - 2~
- Recomeça** (purple): Refresh icon.
- Marcador** (purple): 2.
- Atuadores** (pink):
 - farol~
 - verde
 - vermelho
 - bip
 - 1
 - 2~
- Recomeça** (purple): Refresh icon.

Exemplo com sensor analógico:

Inicialmente o farol e LED vermelho são ligados e apagados. Em seguida ocorre a verificação se o Sensor de Luz sobre o microcontrolador capta um valor maior que 75. Se o valor não for maior que 75 o programa retorna para o marcador 1 e executará toda sequência novamente, portanto enquanto for menor que 75 os atuadores piscarão a cada 0,5 segundo. Por outro lado, se o valor for maior, o programa segue o comando seguinte, responsável por acionar o LED verde e o bip por 1 segundo e por fim o programa é finalizado.

Comandos:

The image shows a vertical sequence of programming blocks in a software interface:

- Marcador:** A purple block with a house icon and a dropdown menu set to '1'.
- Atuadores:** A pink block with a grid icon and a list of actuators: farol~, verde, vermelho, bip, 1, and 2~.
- Pausar:** A teal block with a clock icon, a text input '0.5', and the label 'segundos'.
- Atuadores:** A pink block with a grid icon and a list of actuators: farol~, verde, vermelho, bip, 1, and 2~.
- Pausar:** A teal block with a clock icon, a text input '0.5', and the label 'segundos'.
- Se:** A teal block with a conditional icon, a text input 'luz>75', and a question mark. Below it, 'Vai para:' is followed by a dropdown menu, and 'Se não:' is followed by a dropdown menu set to '1'.
- Atuadores:** A pink block with a grid icon and a list of actuators: farol~, verde, vermelho, bip, 1, and 2~.
- Pausar:** A teal block with a clock icon, a text input '1', and the label 'segundos'.

Exemplo com o operador lógico E (&&):

Neste programa será usado os dois botões A4 e A5 e ambos devem ser apertados simultaneamente. Desta forma os LEDs verde e vermelho serão ligados e depois de dois segundos serão desligados.

“a4 > 50 && a5 > 50” é o mesmo que “botão A4 E botão A5 ligados”.

Comandos :

The image shows a vertical stack of command blocks in a programming environment. The blocks are as follows:

- Marcador 1**: A purple block with a house icon and a dropdown menu showing '1'.
- Pausar 0.5 segundos**: A teal block with a clock icon, a text input field containing '0.5', and the word 'segundos'.
- Se a4>50 && a5>50**: A teal block with a logic icon, a text input field containing 'a4>50 && a5>50', and a question mark. Below it, 'Vai para:' has a dropdown with '2' and 'Se não:' has a dropdown with '1'.
- Marcador 2**: A purple block with a house icon and a dropdown menu showing '2'.
- Atuadores**: A pink block with a grid icon and a list of actuators: 'farol~', 'verde' (checked), 'vermelho' (checked), 'bip', '1', and '2~'.
- Pausar 2 segundos**: A teal block with a clock icon, a text input field containing '2', and the word 'segundos'.
- Atuadores**: A pink block with a grid icon and a list of actuators: 'farol~', 'verde', 'vermelho', 'bip', '1', and '2~'.
- Recomeça**: A purple block with a circular arrow icon.

DICA:

Para sensores digitais a verificação é feita comparando se o valor enviado pelo sensor é maior ou menor que 50.

Para sensores analógicos a verificação é feita com qualquer faixa de valores de 0 a 100.

5.1.1.7. Expressão

Usado para declarar variáveis e expressões em gerais, como cálculos e atribuir resultados a uma variável. Estas devem ter valores numéricos e não aceitam strings (texto).

Exemplo:

O farol, LEDs verde e vermelho devem piscar três vezes. Por fim, o farol permanecerá aceso por 5 segundos e o programa será finalizado.

Comandos :

The image shows a vertical stack of Scratch-style command blocks:

- Expressão** (blue): `x=0`
- Marcador** (purple): `1`
- Se** (blue): `x==3` with a question mark. Below it, `Vai para: 3` and `Se não: 2`.
- Marcador** (purple): `2`
- Pausar** (blue): `0.5` segundos
- Atuadores** (pink): farol~, verde, vermelho, bip, 1, 2~
- Pausar** (blue): `0.5` segundos

A variável contadora é iniciada com o valor.

Se o valor da variável é 3, o farol será aceso por 5 segundos, se não os atuadores piscarão.

-
-
-

Continua na próxima página

The image shows a Scratch script with the following blocks:

- Atuadores** (Actuators):
 - farol~
 - verde
 - vermelho
 - bip
 - 1
 - 2~
- Expressão** (Expression):
 - $x=x+1$
- Ir para** (Go to):
 - 1
- Marcador** (Marker):
 - 3
- Atuadores** (Actuators):
 - farol~
 - verde
 - vermelho
 - bip
 - 1
 - 2~
- Pausar** (Pause):
 - 5 segundos

Após cada piscar, o valor de x será somado de 1.

5.1.2. Demais Comandos

5.1.2.1. Atuador

Através deste bloquinho é possível controlar apenas um atuador isoladamente.

Exemplo:

Comandos :



São utilizados quatro blocos de Atuador, e cada comando está sendo executado linha a linha, uma vez de cada, diferentemente do bloco Atuadores em que todos são acionados simultaneamente.

5.1.2.2. Motor

Usado para controlar apenas um dos dois motores.

A diferença entre Motores e Motor é análoga à diferença de Atuadores e Atuador. Para cada situação problema um deles pode ser uma melhor solução.

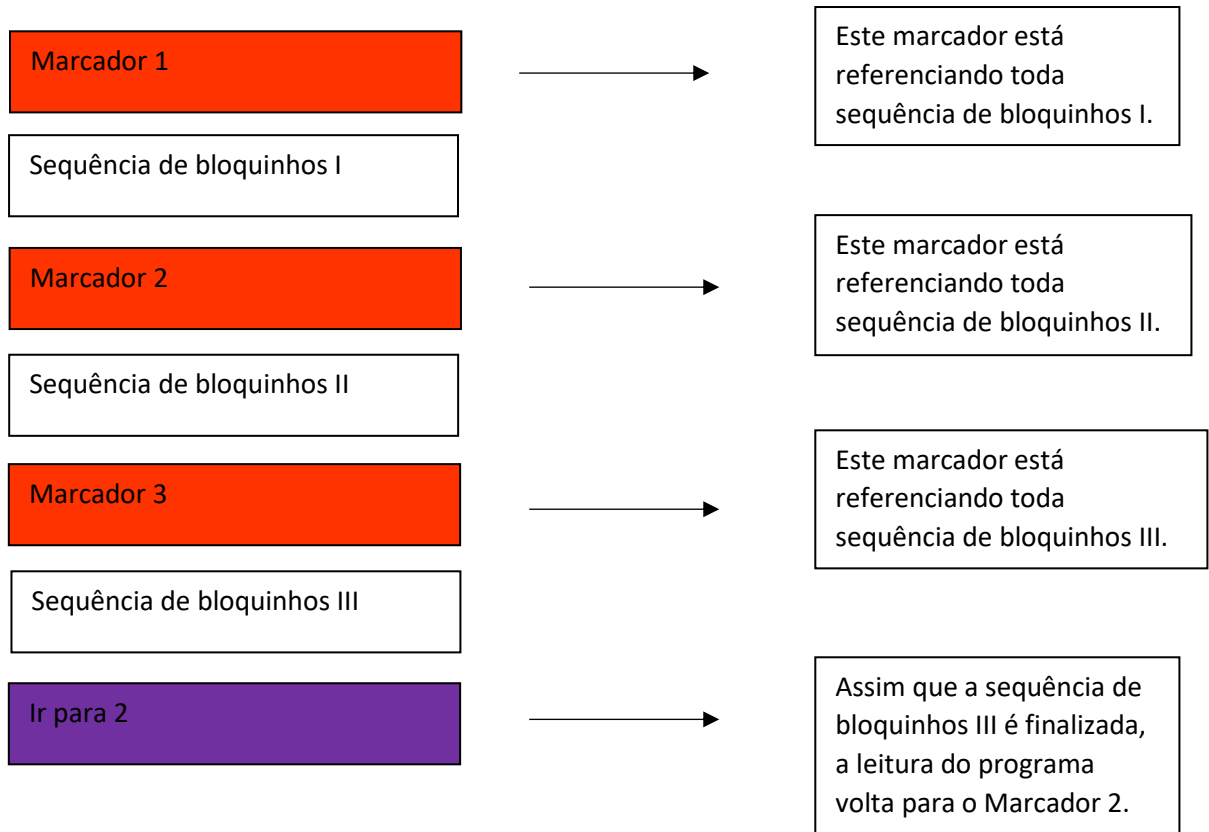
Comandos :



5.1.2.3. Ir para

É utilizado para que a leitura do programa pule ou retorne comandos. Note que seu espaço deve ser preenchido por algum número e este se refere ao trecho em que a leitura do programa deve ir. O trecho é referenciado por um Marcador.

Exemplo estrutural:



Segue um exemplo na próxima página.

Exemplo prático:

Comandos:



The image shows a vertical stack of five colored command blocks from a programming environment:

- Marcador** (purple): A dropdown menu set to '1'.
- Motores** (pink): Two motor controls. Motor A is set to 'Horário' and Motor B is set to 'Anti-Horário'. Each has a slider bar.
- Pausar** (teal): A pause icon and a text input field containing '2 segundos'.
- Motores** (pink): Two motor controls. Both Motor A and Motor B are set to 'Desligar'. Each has a slider bar.
- Ir para** (purple): An arrow icon and a dropdown menu set to '1'.

O início do programa foi referenciado por um marcador. No momento em que a leitura do programa indicar o bloco Ir para 1 toda a sequência abaixo do marcador 1 será executada novamente.

5.1.2.4. Como deletar um bloco de comando

Para deletar qualquer bloquinho basta segurá-lo e em seguida arraste-o para a esquerda e solte o dedo.

5.1.2.5. Falar

Bloco usado para reproduzir uma fala digitada, que será reproduzida pelo dispositivo de som do celular.

A fala digitada deve obedecer às regras gramaticais, tais como vírgulas, pontos de exclamação, ortografia, acentuação de modo que a voz produzida soe o mais natural possível. Valores numéricos e de variáveis também são pronunciados.

Escreva uma frase correta e outra errada, execute e veja a diferença.

Exemplo:

A variável x é definida com o valor de nove. A cada um segundo será executado uma fala diferente.

Para o valor da variável ser reproduzido use o formato “@variável”.

Comandos :

The image shows a vertical list of nine blue command blocks. Each block contains an icon and text. The icons are: a sigma symbol for 'Expressão', a speaker for 'Falar', and a clock for 'Pausar'. The text in the blocks is as follows:

- Expressão: $x=9$
- Falar: O programa será iniciado.
- Pausar: 1 segundos
- Falar: O programa sera iniciado.
- Pausar: 1 segundos
- Falar: 9 segundos.
- Pausar: 1 segundos
- Falar: @x segundos.
- Falar: Tempo de @x segundos

5.1.2.6. Mensagem

O texto digitado é exibido na barra superior da tela, como mostra a imagem do exemplo abaixo.

A exibição do valor das variáveis é feita da mesma forma – “@variável”.

Exemplo:

A mensagem “Programa de contagem” será exibida na barra superior por dois segundos. Em seguida a variável x foi definida com o valor zero. Aparecerá na barra superior “X = 0”, pois zero é o valor da variável. Após apertar o botão A3 o valor de x será atualizado para um. Na tela será exibido “X = 1”. O programa retorna para o Marcador 1 e aguarda até que o botão A3 seja novamente pressionado para que o valor de x seja atualizado para dois e assim ser exibido na tela.

Em resumo, cada vez que o botão A3 é pressionado, o valor de x aumenta em uma unidade e é exibido na tela.

The screenshot shows a programming environment with the following components:

- Header:** "Programa de contagem" in yellow text on a dark grey background.
- Variables:** A row of variables: luz (72), a1 (72), a2 (0), a3 (0), a4 (0), a5 (0), and us (empty).
- Comandos:** A list of commands:
 - Mensagem:** "Programa de contagem"
 - Pausar:** "2 segundos"
 - Expressão:** "x=0"
 - Mensagem:** "X = @x"
 - Marcador:** "1" (dropdown)
 - Interruptor:** "a3" (dropdown)
 - Expressão:** "x=x+1"
 - Mensagem:** "X = @x"
 - Ir para:** "1" (dropdown)

Exemplo com uma variável no bloco Expressão e o bloco Se com operador lógico E (&&):

A variável x foi inicializada com o valor zero. Após um segundo, seu valor será incrementado de um, passando a valer um, e este valor será exibido na tela com a frase "X É 1". Para o programa ir para o Marcador 2 é necessário que o botão A4 E o botão A5 sejam pressionados E que o valor de x seja maior que zero, estas três condições devem ocorrer simultaneamente. Caso não ocorra, o programa retorna para o Marcador 1 e o valor de x terá outro incremento, tendo assim um novo valor e este aparecerá na tela.

(Neste exercício, o valor de x sempre será maior que 0 quando a leitura do programa estiver no bloco Se).

The image shows a Scratch script with the following blocks:

- Monitor:** A display area at the top showing "X É 1" in yellow text. Below it, a row of monitors for variables: luz (65), a1 (65), a2 (65), a3 (0), a4 (0), a5 (0), and us (empty).
- Comandos:** A section containing several blocks:
 - Expressão:** $x=0$
 - Marcador:** 1
 - Pausar:** 1 segundos
 - Expressão:** $x=x+1$
 - Mensagem:** X É @x
 - Se:** $a4>50 \&\& a5>50 \&\& x>0$. Below the condition, "Vai para:" is set to 2 and "Se não:" is set to 1.
 - Marcador:** 2
 - Atuador:** farol (dropdown) and Ligar (dropdown)
 - Pausar:** 3 segundos
 - Atuador:** farol (dropdown) and Desligar (dropdown)
 - Recomeça:** A block with a circular arrow icon.

5.1.2.7. Interruptor

Este bloco é usado para sensores. Assim que a leitura do programa chegar neste bloco, esta permanecerá travada, aguardando que o sensor selecionado seja ativado ou que seu estado seja alterado para que a leitura do programa avance para o próximo bloquinho.

Exemplo com sensor digital:

Conecte um Sensor de Toque (botão) no Conector A2.

Comandos :



O farol iniciará ligado. O programa permanecerá travado em Interruptor e avançará se o estado inicial do botão mudar, ou seja, se o botão for pressionado. O farol permanecerá desligado por 5 segundos e então o programa reiniciará.

Exemplo com sensor analógico:

Tente o mesmo exemplo mudando de “a2” para “luz” (este é o Sensor de Luz on-board). Nesta situação para o farol desligar será necessária uma das duas situações abaixo:

- Se seu valor medido em “luz” na barra superior normalmente for maior que 50 devido às condições do ambiente – então o valor deverá se alterar para um número inferior a 50 e depois deverá aumentar novamente para um número maior que 50 e então será enviado um sinal para a ação ser tomada. (Ao dar Play, inicie o programa com a lanterna do celular apontada para o sensor, assim que a leitura do programa chegar em Interruptor coloque o dedo no sensor para o valor medido diminuir. Por fim tire o dedo para aumentar o valor. Após isso o farol será desligado)
- Se seu valor medido em “luz” normalmente for menor que 50 – o valor deverá se alterar para um número maior que 50 para a ação ser tomada. (Ao dar Play, inicie o programa com o dedo no sensor, assim que a leitura do programa chegar em Interruptor tire o dedo e aponte a lanterna do celular sobre o sensor para o valor aumentar. O farol será desligado.)

5.1.2.8. Gatilho

Este bloquinho também é usado para verificar o estado do sensor, entretanto a sua verificação ocorre a todo instante de tempo, independentemente de onde estiver a leitura atual do programa.

O programa principal estará em execução e ao ativar o sensor, a leitura do programa principal será interrompida e direcionada imediatamente para outro trecho com outras sequências de comandos específicas que foi determinado pelo usuário caso o sensor tenha seu estado alterado.

Para usar este bloquinho a estrutura do programa deve respeitar as seguintes condições:

Sequência de blocos do programa principal; (estará em execução até o momento em que o estado do sensor for alterado)

Bloco Recomeça ou Ir Para;

Bloco Gatilho; (se for acionado, a sequência abaixo é executada)

Sequência que deve ser executada caso o sensor tenha seu estado alterado;

Bloco Recomeça.

Exemplo com o bloco Recomeça:

Conecte um LED ou um BIP no Conector D1 e um Sensor de Obstáculo no Conector A2.

Comandos:

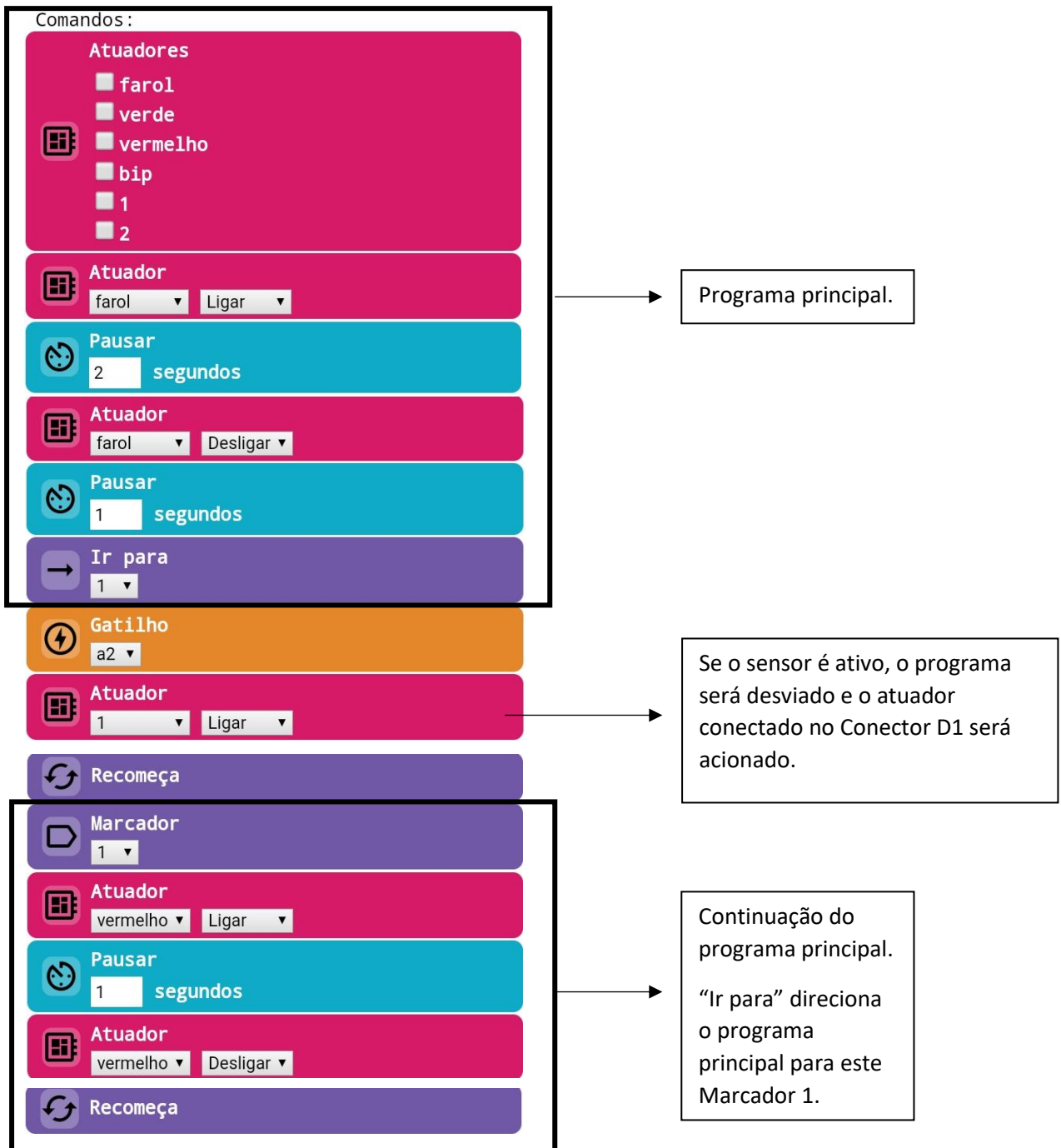


Programa Principal

Se o sensor em A2 é ativo, o programa será desviado e o atuador conectado em Conector D1 será acionado.

O farol alternará entre ligado e desligado a cada dois segundos. Ao acionar o sensor conectado em A2 o gatilho será acionado e o atuador conectado em D1 é ativo. Por fim o programa é reiniciado.

Exemplo com o bloco Ir para:



Os atuadores iniciam desligados. O farol irá acender e apagar, a leitura do programa será encaminhada para o marcador 1, onde o led vermelho acende e apaga. Após essas sequências o programa recomeça. Caso o gatilho é acionado, a leitura do programa é encaminhada para acionar o conector D1 (para ficar mais tempo acionado coloque um bloco Pausar entre o Atuador 1 e o bloco Recomeça) e em seguida o programa é reiniciado.

A mesma explicação para os sensores digitais e analógicos feita no item anterior continuam válidas.

5.1.2.9. Ultrassônico

O sensor ultrassônico capta distância de objetos de 0 a 100 cm.

Exemplo:

Se o sensor captar um objeto com uma distância inferior a 14 cm o farol permanecerá aceso e se o objeto se afastar ou estiver numa distância superior a 14 cm o LED vermelho será aceso (o farol será apagado) e permanecerá assim enquanto esta condição for verdadeira.

Comandos:



Marcador
1

Se
us<14
Vai para: 2 Se não: 3

Marcador
2

Atuadores
 farol-
 verde
 vermelho
 bip
 1
 2-

Ir para
1

Marcador
3

Atuadores
 farol-
 verde
 vermelho
 bip
 1
 2-

Ir para
1

Para utilizar o **ultrassônico como Interruptor**:

* Se o valor medido é maior que 15 é necessário fazer com o que valor se altere para um número menor que 15.

* Se o valor medido é menor que 15 é necessário fazer com que o valor seja maior que 15 e em seguida volte a ser menor que 15.

Exemplo:

O LED vermelho ficará aceso até que o ultrassônico seja ativado como Interruptor (caso o valor é maior que 15 coloque a mão bem próxima do sensor, caso contrário afaste o sensor do objeto e aproxime algo do sensor novamente) para que o LED seja desligado. Ao acionar o botão A3 do microcontrolador os LEDs verde e vermelho serão acesos. Se A4 for pressionado em seguida, estes LEDs serão desligados e o farol será aceso com o menor brilho possível e chegará até a sua máxima intensidade de 100. Terá seu brilho incrementado de 1 a 1, iniciando do 0 e chegando até 100. Assim que chegou no brilho máximo de 100, o farol permanecerá aceso por 5 segundos para então ser desligado. Por fim o programa recomeçará.

Comandos :

Atuador
vermelho ▾ Ligar ▾

Interruptor
us ▾

Atuador
vermelho ▾ Desligar ▾

Interruptor
a3 ▾

Atuadores

- farol~
- verde
- vermelho
- bip
- 1
- 2~

Interruptor
a4 ▾

Atuadores

- farol~
- verde
- vermelho
- bip
- 1
- 2~

Expressão
X=0

Marcador
1 ▾

(continua na próxima página)

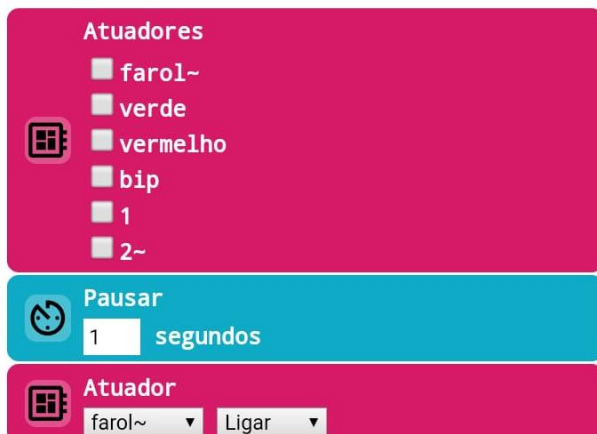


Para utilizar o **ultrassônico como Gatilho** é necessário que para ativar o Gatilho a distância seja menor que 15 cm.

Exemplo:

O farol permanece piscando, mas se o ultrassônico for ativado como Gatilho, o programa será desviado fazendo com que o farol e os LEDs permaneçam acesos por três segundos.

Comandos:



(continua na outra página)



5.1.2.10. PWM

Para gerar valores analógicos nas saídas é usado uma técnica chamada de PWM (modulação por largura de pulso). Desta forma, saídas digitais que possuíam apenas dois estados - desligado e ligado -, agora têm a possibilidade de se definir o quanto de intensidade pode estar atuando. Agora é possível controlar se o LED vai brilhar mais ou menos.

Somente o farol e o conector D2 possuem o controle por PWM.

Exemplo:

Se o sensor de luz detectar uma luminosidade inferior a 35, o farol terá sua intensidade máxima, mas se o sensor captar uma luminosidade maior ou igual a 35, o farol terá uma intensidade de 25%.

Comandos:



(continua na próxima página)

Marcador
 3

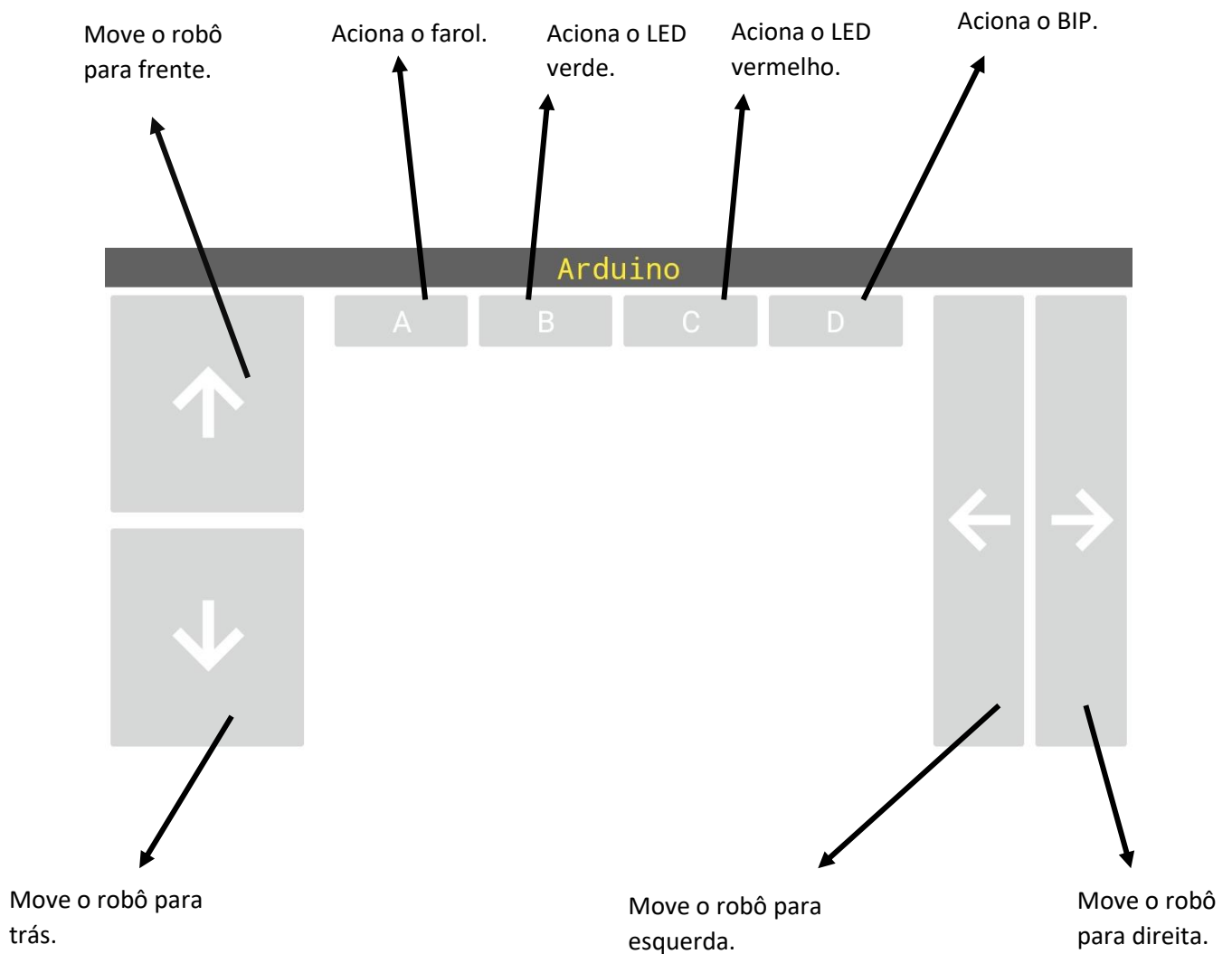
Se
 Luz >= 35 ?
 Vai para: ... Se não: 1

PWM
 farol ~ 25 por cento

Recomeça

5.2. Modo controle remoto

Usado para controlar motores, LEDs e BIP on-boards.



6. Gabarito

O gabarito é para ser usado como um método manuscrito com o objetivo de fixar os conceitos de cada bloquinho e a lógica de programação. Ao invés de se programar diretamente no celular, a programação será feita a mão no gabarito.

Seu modelo se encontra na última página.

“Comandos” representa cada bloco. “Parâmetros” são os itens a serem preenchidos de acordo com o solicitado pelo comando.

Comandos e Parâmetros na sequência abaixo estará no formato Comando(parâmetros):

Marcador (número do marcador; exemplo: 1)

Gatilho (sensor a ser utilizado; exemplo: a1)

Interruptor (sensor a ser utilizado; exemplo: a1)

Ir para (número do marcador de destino; exemplo: 1)

Recomeça (vazio)

Atuadores (atuadores ligados são colocados nos parâmetros; exemplo: farol, bip, ...)

Atuador (atuador, estado; exemplo: farol, ligado)

Motores (estado motor A, velocidade motor A, estado motor B, velocidade motor B)

Motor (motor, estado do motor, velocidade do motor; exemplo: A, ligado, 100)

Se (decisão) (expressão, marcador vai para, marcador se não; exemplo: a1 = 1, 1, 2)

Expressão(expressão; exemplo: $x = x + 1$)

Falar (“Texto”; exemplo: “Olá, tudo bem?”)

Mensagem (“Texto”; exemplo: “O valor é de @x”)

Pausar (tempo; exemplo: 8)

PWM(atuador, porcentagem; exemplo: “farol, 88%”)

Exemplo de uso:

Comandos/Parâmetros

Atuador	Farol, ligar
Marcador	2
Pausar	0,5
Se	A5>50, 1, 2
Marcador	1
Motores	horário, 50, anti-h, 75
Atuadores	Led vermelho, bip
Pausar	3
Atuadores	
Motores	desligar, 50, desligar, 50

Este programa inicia com o farol ligado. A cada 0,5 segundo será verificado se o botão A5 é pressionado (se não for pressionado o programa retorna para o Marcador 2). Se for pressionado, a leitura do programa é direcionada para o Marcador 1 e esta seguirá os blocos em sequência – o motor A terá 50% de rendimento no sentido horário e o motor B terá 75% de rendimento no sentido anti-horário. Em seguida o Led Vermelho e Bip serão acionados com um tempo de três segundos. O penúltimo bloco faz com que o farol, o LED vermelho e o BIP sejam desligados. Por fim, os motores também o são.

Na próxima página há o mesmo programa na versão para celular.

Comandos:

Atuador
farol ▼ Ligar ▼

Marcador
2 ▼

Pausar
0.5 segundos

Se
a5>50 ?
Vai para: 1 ▼ Se não: 2 ▼

Marcador
1 ▼

Motores
A: Horário ▼
B: Anti-Horário ▼

Atuadores
 farol
 verde
 vermelho
 bip
 1
 2

Pausar
3 segundos

Atuadores
 farol
 verde
 vermelho
 bip
 1
 2

Motores
A: Desligar ▼
B: Desligar ▼

Recomeça

