

# Atuadores

- Saídas
- Som
- Exibição
- Motor

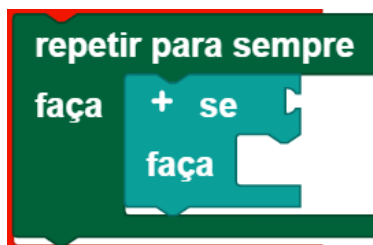
# Saídas

## O início de cada bloco de tempo

O **bloco de tempo** oferece a possibilidade de executar um programa se uma condição for satisfeita. Portanto, ele funciona de maneira semelhante a uma distinção de caso, mas não é executado apenas uma vez, e sim sempre que a condição for atendida durante todo o curso do programa. O **início de cada bloco de tempo**:



É uma abreviatura para a seguinte construção:



Você pode usar todas as condições desta categoria **ao iniciar cada bloco de tempo** da categoria de saídas.

**Observação:** A seção do programa dentro do **iniciar de cada bloco de tempo** deve ser mantida curta e não conter chamadas de bloqueio ou loops sem fim para que esta parte do programa possa ser processada rapidamente.

## LEDs



### Definir

Com os blocos **Definir LED ...** e **Definir brilho do LED ...**, é possível ligar e desligar o LED ou definir seu brilho para um determinado valor (de 0 a 512).

### Lembrar

Com o bloco **obter brilho do LED**, o brilho de um LED pode ser acessado e posteriormente processado como um valor.

### Consultar

Com os blocos **o LED está ...** e **O brilho do LED está ...**, você pode usar a atividade ou o brilho de um LED como condição. No exemplo, o brilho do LED está definido para 512, a menos que já tenha esse brilho.



## Motores

O símbolo nos blocos do motor representa todos os motores que não são codificadores ou servo motores.

### Definir

Com o bloco **definido a velocidade do motor para [] ...**, você pode definir a velocidade de um motor para um determinado valor (de 0 a 512).

### Lembrar

Com o bloco **obter velocidade do motor**, a velocidade de um motor pode ser acessada e posteriormente processada como um valor.

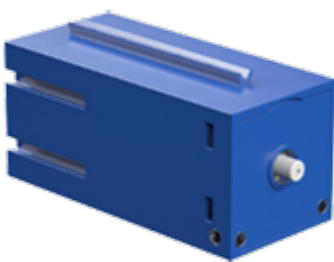
### Consultar

Com os blocos **motor em execução** e **a velocidade do motor está ...**, você pode usar a atividade ou a velocidade de um motor como condição.

### Parar

Com o bloco **parar motor ...**, é possível parar um motor.

## Compressor



### Definir

Com o bloco **definir compressor []**, você pode ligar ou desligar o compressor.


### Consultar

Com o bloco **o compressor está []**, pode-se usar a atividade de um compressor como condição.


## Válvula solenoide



## Definir

Com o bloco **definir válvula solenoide**  , você pode ligar ou desligar a válvula solenoide. Aqui, “ligado” significa que a válvula está aberta, e “desligado”, que a válvula está fechada.

## Consultar

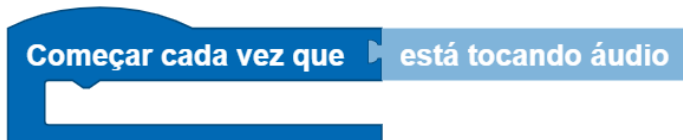
Com o bloco **a válvula solenoide está**  , pode-se usar a atividade de uma válvula solenoide como condição.

# Som

O controlador TXT 4.0 possui um alto-falante embutido e, portanto, oferece a possibilidade de reproduzir sons.

## O início de cada bloco de tempo

O **bloco de tempo** oferece a possibilidade de executar um programa se uma condição for satisfeita. Portanto, ele funciona de maneira semelhante a uma distinção de caso, mas não é executado apenas uma vez, e sim sempre que a condição for atendida durante todo o curso do programa. O **início de cada bloco de tempo**:



É uma abreviatura para a seguinte construção:



Você pode configurar todas as condições da categoria som em **iniciar cada bloco de tempo** a este nível.

**Observação:** A seção do programa dentro do **iniciar de cada bloco de tempo** deve ser mantida curta e não conter chamadas de bloqueio ou loops sem fim para que esta parte do programa possa ser processada rapidamente.

## Reproduzir

### Arquivos de áudio pré-instalados

Com o bloco a seguir, você pode reproduzir um dos 29 sons pré-instalados. O arquivo de áudio desejado pode ser selecionado no menu suspenso (triângulo pequeno). Também é possível reproduzir o som em um loop contínuo. Para tanto, você deve marcar a caixa atrás do símbolo de loop.



### Arquivos de áudio próprios

Caso queira tocar seu próprio som, você pode usar o bloco



. Para incorporar seu próprio som no bloco, você deve:

1. Estar conectado ao controlador
2. Digite o endereço IP do controlador no navegador (o IP que também foi usado para se conectar ao controlador deve ser selecionado aqui)
3. Digite USER: ft, PASSWORD: fischertechnik na página aberta
4. Abra a pasta de sons e use o sinal de mais para carregar o arquivo de áudio desejado no controlador (importante: o arquivo de áudio deve estar no formato wav)
5. Digite no endereço "./nome do arquivo.wav" no bloco de codificação ROBO Pro

Também aqui existe a opção de reproduzir o som em loop contínuo.

## Consultar

Para saber se um arquivo de áudio está sendo reproduzido, o bloco **reproduzir som** é usado . Isso pode ser usado como uma condição no programa.

## Parar

Para interromper um som, basta usar o bloco **parar reprodução de som** no programa.


# Exibição

Com os blocos da categoria exibição, a tela do controlador TXT 4.0 pode ser desenhada e tornada utilizável. Isso é feito em duas etapas:

1. Configurar, isto é
  - Abra um novo arquivo na categoria Display por meio do símbolo de página com o sinal de mais no canto superior esquerdo
  - Arraste os elementos desejados para a área da grade (representa a parte configurável da tela)
  - adaptar as especificações, se necessário.
2. Programar, isto é
  - No programa principal, usar os blocos da categoria Exibição para programar o efeito da interação com a exibição.

## Blocos

### Consulta de evento

O bloco **E vento**  abre o valor de retorno de um elemento. Este bloco só pode ser utilizado nos programas de eventos. Nestes programas de eventos, o bloco refere-se automaticamente ao evento em cujo programa é utilizado. O tipo adequado para o valor de retorno pode ser selecionado através do menu suspenso (triângulo pequeno):



### Campo de rotulagem

Com o elemento campo de rotulagem, você pode colocar um texto na tela. O símbolo no configurador de exibição é a etiqueta. Se você arrastar este símbolo para a área da grade, uma janela será aberta à direita. Aqui, em inspetor,

- o tamanho do campo de rotulagem em pixels,
- a posição do campo de rotulagem em pixels (o canto superior esquerdo do campo de texto está no ponto especificado),
- o nome do campo de rotulagem e
- o conteúdo do campo de rotulagem (este texto é mostrado quando a exibição é iniciada)

são definidos.

Com o bloco **definir texto do campo de rotulagem ...** o texto exibido pode ser alterado no decorrer do programa.

## Entradas

O elemento **Entrada** permite que os(as) usuários(as) insiram texto por meio do controlador. O símbolo correspondente no configurador de exibição é o caractere "T". Se você arrastar este símbolo para a área da grade, uma janela será aberta à direita. Aqui, em inspetor,

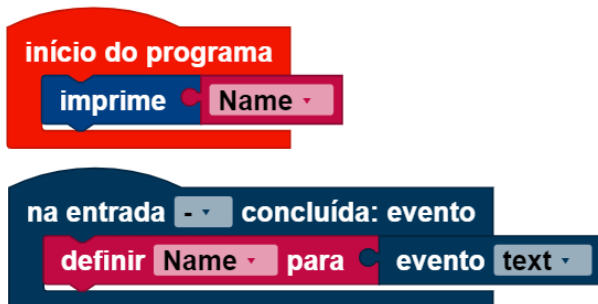
- o tamanho do campo de entrada em pixels,
- a posição do campo de entrada em pixels (o canto superior esquerdo do campo de entrada está no ponto especificado),
- o nome do campo de entrada e
- o conteúdo do campo de entrada (este texto é mostrado quando a exibição é iniciada)

são definidos.

Com o bloco **definir texto do campo de entrada...** o texto exibido pode ser alterado no decorrer do programa.

## Programa de entrada

O programa de entrada é executado quando uma entrada é concluída. Ele é escrito separadamente do programa principal. As variáveis funcionam globalmente em ambos os programas. O programa de entrada é executado no bloco **quando a entrada é concluída**. O bloco **Evento []** é definido no programa de entrada como "texto". Neste exemplo, a variável **Nome** é definida para o texto digitado, que é então utilizado no programa principal para alterar o texto digitado para produzir:



## Instrumento de medição

A função do instrumento de medição pode exibir valores (nenhum valor menor que 1). O símbolo correspondente no configurador de exibição é a escala. Se você arrastar este símbolo para a área da grade, uma janela será aberta à direita. Aqui, em inspetor,

- o tamanho do instrumento de medição em pixels,
- a posição do instrumento de medição em pixels (o canto superior esquerdo do medidor está no ponto especificado),
- o nome do instrumento de medição,
- a orientação do instrumento de medição
- a faixa de valores representada pelo instrumento de medição e
- o valor do instrumento de medição mostrado quando o display é iniciado

são definidos.

Com o bloco **definir valor do instrumento de medição em ...** o instrumento de medição pode ser definido para o valor inserido. Este valor deve estar na faixa de valores definida anteriormente. Se o valor estiver fora da faixa de valores, um dos limites da faixa de valor é exibido, dependendo se o valor é muito grande ou muito pequeno.

## Exibição de status



O indicador de status exibe alguma atividade. Dependendo do estado, acende ("ativo") ou não acende ("inativo"). O símbolo no configurador de tela é um diodo brilhante. Se você arrastar este símbolo para a área da grade, uma janela será aberta à direita. Aqui, em inspetor,

- o tamanho da exibição de status em pixels,
- a posição da exibição de status em pixels (o canto superior esquerdo dos indicadores de status está no ponto especificado),
- o nome da exibição de status,
- a cor da exibição de status e
- se a exibição de status deve estar ativa ou inativa no início,

são definidos.

Com o bloco **definir exibição de status como ativa** [], a exibição de status pode ser habilitada ou desabilitada. No menu suspenso (triângulo pequeno), você pode escolher se a exibição de status deve ser definida como ativa ou inativa.

## Controle deslizante

O controle deslizante retorna valores dependendo de sua posição. A posição pode ser alterada pelo usuário por meio da tela sensível ao toque. O valor pode ser recuperado através do bloco **Evento** [] assim que o controle deslizante estiver em repouso. O valor recuperado é um número decimal. Se você quiser que o valor do controle deslizante seja um número inteiro, você deve usar o bloco **redondo**. O símbolo correspondente para o controle deslizante é a linha com o círculo. Se você arrastar este símbolo para a área da grade, uma janela será aberta à direita. Aqui, em inspetor,

- o tamanho do controle deslizante em pixels,
- a posição do controle deslizante em pixels (o canto superior esquerdo do controle deslizante está no ponto especificado),
- o nome do controle deslizante,
- a atividade do controle deslizante,
- a orientação do controle deslizante,
- o intervalo de valores coberto pelo controle deslizante e
- o valor em que o controlador está quando começa a exibição

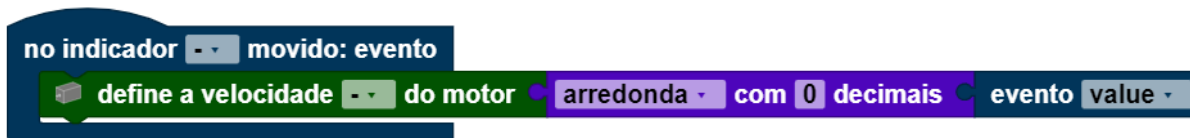
são definidos.

com o bloco **Definir valor do controle deslizante ...** você pode mover o controle deslizante para um valor diferente.

Com **definir controle deslizante como ativado** [], você pode alterar a atividade por meio do menu suspenso (triângulo pequeno).

## Programa do controle deslizante

O programa do controle deslizante é executado após o controle deslizante ser movido. Ele é escrito separadamente do programa principal. As variáveis funcionam globalmente em ambos os programas. O programa do controle deslizante é executado no bloco **quando o controle deslizante se move**. O bloco **Evento** [] é definido quanto ao valor no programa do controle deslizante. Neste exemplo, a velocidade do motor é controlada usando-se o controle deslizante. O valor do controle deslizante deve ser arredondado porque o motor só aceita números inteiros como velocidade:



## Botão

O botão é um campo rotulado que pode ser pressionado. Se você pressionar o botão, o programa do botão será executado assim que for liberado novamente. O símbolo associado ao botão é o quadrado com o rótulo "OK". Se você arrastar este símbolo para a área da grade, uma janela será aberta à direita. Aqui, em inspetor,

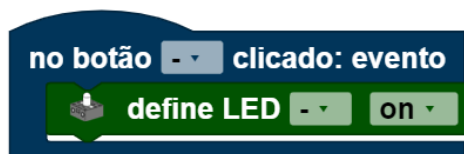
- o tamanho do botão em pixels,
- a posição do botão em pixels (o canto superior esquerdo do botão está no ponto especificado),
- o nome do botão,
- o texto no botão e
- a atividade do botão

são definidos.

Com o bloco **definir controle deslizante como ativado** [], você pode alterar a atividade por meio do menu suspenso (triângulo pequeno).

## Programa de botões

O programa de botões é executado assim que o botão não estiver mais pressionado. Ele é escrito separadamente do programa principal. As variáveis funcionam globalmente em ambos os programas. O programa de botões é executado no bloco **quando o botão é clicado**. O bloco **Evento** [] pode programar botões não usados, pois o botão não tem valor de retorno. Neste exemplo, o LED é ativado quando o botão é pressionado.



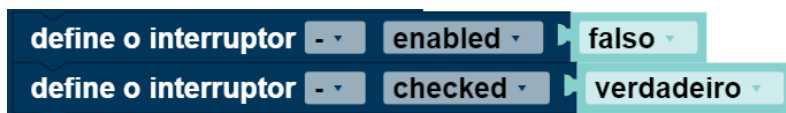
## Interruptor

O interruptor pode ocupar duas posições e está sempre exatamente em uma dessas duas posições. Dependendo da posição, ele retorna **verdadeiro** ou **falso**. O símbolo correspondente para o interruptor é o oval com a ponta. Se você arrastar este símbolo para a área da grade, uma janela será aberta à direita. Aqui, em inspetor,

- o tamanho do interruptor em pixels,
- a posição do interruptor em pixels (o canto superior esquerdo do interruptor está no ponto especificado),
- o nome do interruptor,
- o texto próximo ao interruptor,
- a atividade do interruptor e
- o estado em que o interruptor deve estar quando o programa começa

podem ser ajustados.

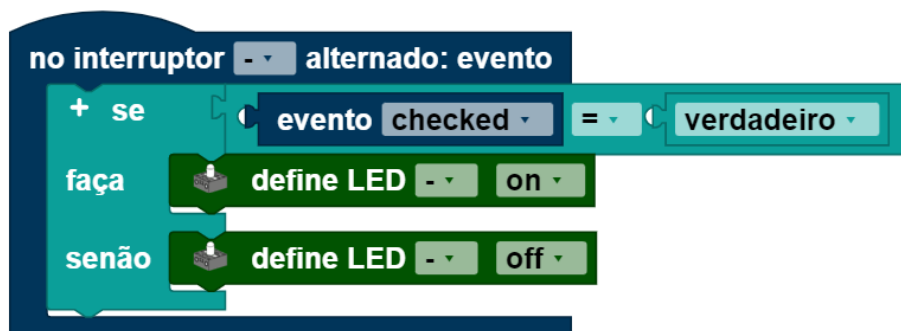
O bloco



executa duas funções. Você pode definir a atividade (selecionar ativado no menu suspenso) ou o estado (marcad no menu suspenso) como **verdadeiro** ou **falso**.

## Programa do interruptor

O programa do interruptor é executado sempre que o interruptor é acionado. Ele é escrito separadamente do programa principal. As variáveis funcionam globalmente em ambos os programas. O programa do interruptor é executado no bloco **quando o interruptor é desativado**. O bloco **Evento []** é definido no programa de ativação como "marcado", ele retorna **verdadeiro** ou **falso**. Este programa de exemplo liga o LED quando a chave é acionada, caso contrário, o LED é desligado:



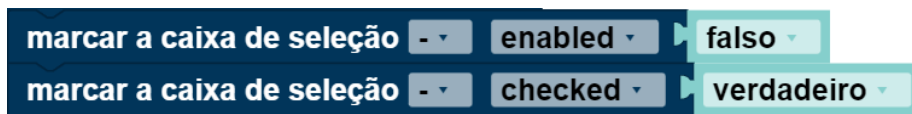
## Caixa de seleção

A caixa de seleção pode ter dois estados e está sempre exatamente em um desses dois estados. Dependendo da posição, ela retorna **verdadeiro** ou **falso**. O símbolo da caixa de seleção é o quadrado com a marca de seleção. Se você arrastar este símbolo para a área da grade, uma janela será aberta à direita. Aqui, em inspetor,


- o tamanho da caixa de seleção em pixels,
- a posição da caixa de seleção em pixels (o canto superior esquerdo da caixa de seleção está no ponto especificado),
- o nome da caixa de seleção,
- o texto próximo à caixa de seleção,
- a atividade da caixa de seleção e
- o estado em que a caixa de seleção deve estar quando o programa começa

são definidos.

O bloco seguinte executa duas funções. O menu suspenso (triângulo pequeno) pode ser selecionado para se escolher qual usar. Você pode definir a atividade (selecionar ativado no menu suspenso) ou o estado (marcad no menu suspenso) como **verdadeiro** ou **falso**.



## Programa de caixa de seleção

O programa da caixa de seleção é executado sempre que a caixa de seleção é pressionada. Ele é escrito separadamente do programa principal. As variáveis funcionam em ambos os programas. O programa do interruptor é executado no bloco **quando a caixa de seleção é desativada**. O bloco **Evento**  é definido no programa de ativação como "marcado", ele retorna **verdadeiro** ou **falso**. Este programa de exemplo liga o LED quando a caixa de seleção é acionada, caso contrário, o LED é desligado:



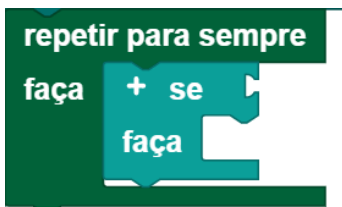
# Motor

## O início de cada bloco de tempo

O **bloco de tempo** oferece a possibilidade de executar um programa se uma condição for satisfeita. Portanto, ele funciona de maneira semelhante a uma distinção de caso, mas não é executado apenas uma vez, e sim sempre que a condição for atendida durante todo o curso do programa. O **início de cada bloco de tempo**:



É uma abreviatura para a seguinte construção:



Você pode configurar todas as condições da categoria motor em **iniciar cada bloco de tempo** a este nível.

**Observação:** A seção do programa dentro do **iniciar de cada bloco de tempo** deve ser mantida curta e não conter chamadas de bloqueio ou loops sem fim para que esta parte do programa possa ser processada rapidamente.

## Motor

O símbolo nos blocos do motor representa todos os motores que não são codificadores ou servo motores.

### Definir

Com o bloco **definido a velocidade do motor para [] ...**, você pode definir a velocidade de um motor para um determinado valor (de 0 a 512). A direção de rotação pode ser selecionada no menu suspenso (triângulo pequeno).

### Lembrar

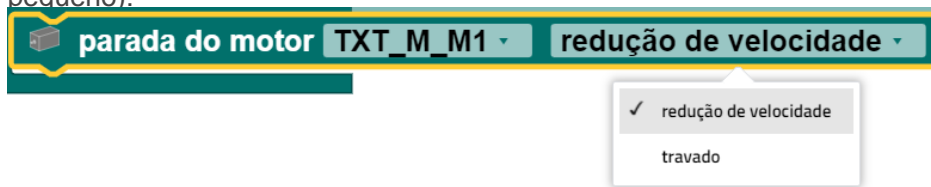
Com o bloco **obter velocidade do motor**, a velocidade de um motor pode ser acessada e posteriormente processada como um valor.

### Consultar

Com os blocos **motor em execução** e **a velocidade do motor está ...**, você pode usar a atividade ou a velocidade de um motor como condição.

### Parar

Com o bloco **parar motor** [], é possível parar um motor. O bloco **parar motor** [] oferece as opções de parar um motor diretamente ou por inércia. A opção desejada pode ser selecionada por meio do menu suspenso (triângulo pequeno):



## Servo motor



### Definir

Com o bloco **Definir posição a ...**, você pode definir a posição de um servo motor a um determinado valor (de 0 a 512). 0 e 512 são os valores para a deflexão máxima à direita e à esquerda. Com o valor 256, o servo motor está, correspondentemente, no meio.

### Lembrar

A posição de um servo motor pode ser acessada com **acessar posição** e posteriormente processada como um valor.

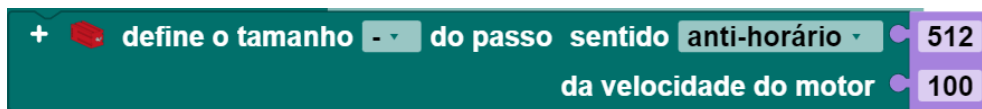
## Motor codificador



O motor codificador tem as mesmas funções de um motor normal, mas também oferece a opção de contar as rotações e controlar vários motores de maneira síncrona. Uma revolução é dividida em ~64 etapas.

### Definir

Com o bloco

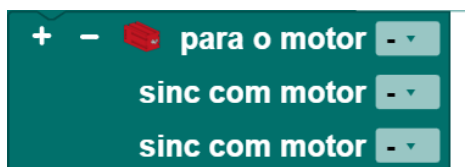


you can define the speed of a motor for a determined value (from 0-512). The direction of rotation can be selected in the menu (small triangle). Besides this, you can insert the number of steps that the motor must traverse. In this example, the motor rotates 100 steps, or in other words, one revolution and one third. As can be seen in the example, this block has a plus sign, with the help of which several motors can be controlled in a synchronized manner. It is possible to synchronize motors in the master or in an extension; a general synchronization, for example, between the motors of the master and an extension, is not possible.

**Observação: Chamadas de sincronização em rápida sucessão, como aquelas possibilitadas por um loop (veja o exemplo), podem prejudicar a sincronicidade ou até mesmo causar sua perda completa.**

## Parar

With the **stop motor ...** block it is possible for a motor. If you want to stop several motors at the same time, you can add up to three motors using the addition button on the left of the block.



## Consultar

The **reached position** block is used to use the position reached as a condition. Here, position means the final position of an encoder motor after the step is completed.