

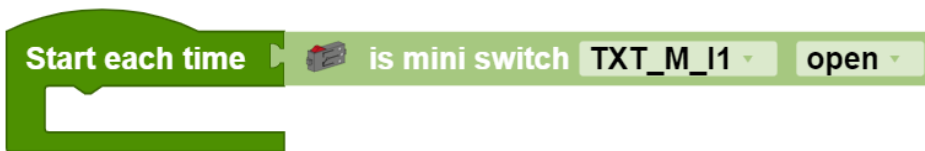
# Sensors

- Input
- Counter
- I2C
- USB

# Input

## The Start each time block

The **Start each time block** offers the option of running a program when a condition is fulfilled. Therefore, it works similar to a case distinction, but runs not only one time, but rather every time the condition is fulfilled during the entire course of the program. The **Start each time block**:



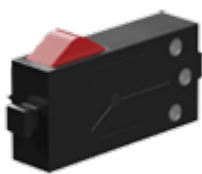
Is an abbreviation for the following construct:



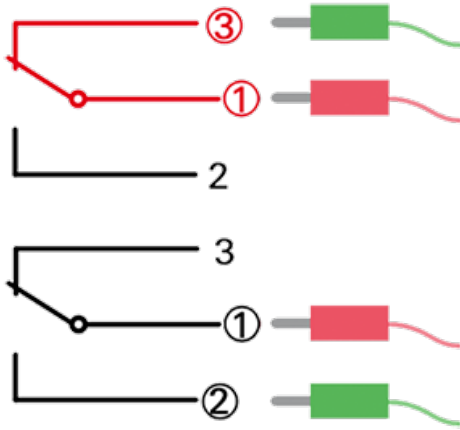
You can insert all conditions from this category into the inputs category in the **Start each time block**.

**Note:** The program section in the **Start each time block** should be kept short, and should not contain any blocking calls or endless loops, so that this part of the program can be processed quickly.

## Button



The button is a digital sensor that differentiates “current flowing” from “no current flowing.” Whether current is flowing, therefore, depends both on the wiring and on whether the button is pushed. You can use the button in two ways:



As a “normally open contact”:  
contacts 1 and 3 are connected.

Button pressed: Current is flowing.  
pressed: No current is flowing

Button not

As a “normally closed contact”:  
Contacts 1 and 2 are connected.

Button pressed: No current is flowing.  
pressed: Current is flowing.

Button not

## Call

You can use **get mini button status** to obtain more information on whether current is flowing through the button or not. If current is flowing, it will return **1**, or **0** if no current is flowing.

## Query

To query whether the button is in a certain status, use the **is button ...** block. You can use the drop down menu (small triangle) to select which status to query. This block can be used as a condition.

## Ultrasonic sensor



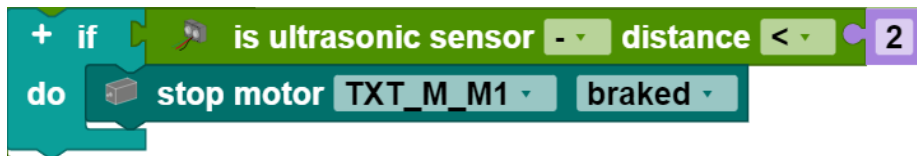
The ultrasonic sensor is used to measure distances.

## Call

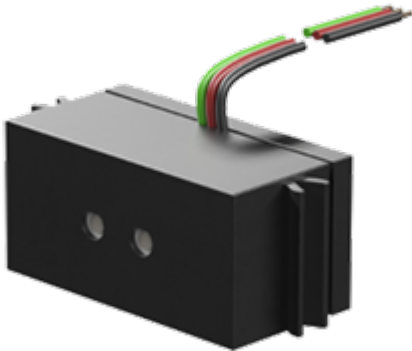
You can use **get ultrasonic sensor distance** to obtain information on how far the sensor is from the closest object. The distance is stated in cm.

## Query

To query whether the sensor is a certain distance from the closest object, use the **is ultrasonic sensor distance ...** block. You can use the drop down menu (small triangle) to select how the measured distance should be compared to an entered value (<, ?, =, ?, ?, >) . The comparison value is entered in the number field at the end of the block. This block can be used as a condition. In the example, the motor is stopped when the measured distance is less than 2 cm.



## Color sensor



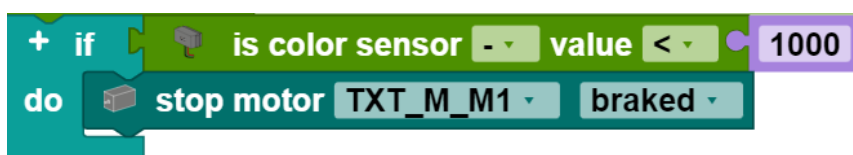
The color sensor transmits red light, and measures how much of the light is reflected back. The color sensor returns values between 0 and 2000 depending on how strong the reflection is. It is well-suited for detecting previously calibrated colors.

## Call

You can use **get color sensor value** to obtain information on how strongly a surface reflects the light.

## Query

To query whether the sensor is in front of a certain color, use the **is color sensor value ...** block. You can use the drop down menu (small triangle) to select how the measured color value should be compared to the entered value (<, ?, =, ?, ?, >) . The comparison value is entered in the number field at the end of the block. This block can be used as a condition. In the example, the motor is stopped when the measured color value is less than 1000.



## IR track sensor



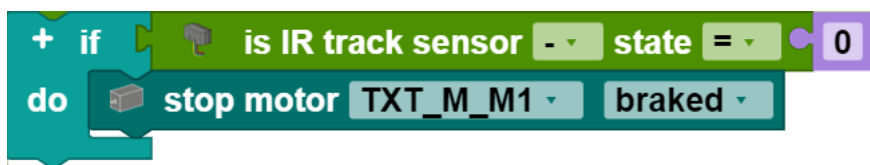
The infrared track sensor is a digital sensor used for detecting a black track on a white background, and works at a distance of 5 - 30 mm between the sensor and the background.

## Call

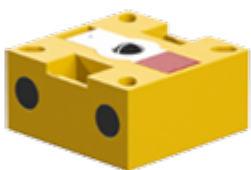
**Get IR track sensor status** will return 0 if the sensor does not detect a track. If the sensor detects a track, it will return 1

## Query

To query whether the IR track sensor detects a track, compare the current track status with 0 or 1. To do so, use the block **is IR track sensor status** []. You can use the drop down menu (small triangle) to select how the track status should be compared to the entered value (<, ?, =, ?, ?, >) . The comparison value is entered in the number field at the end of the block. This block can be used as a condition. In the example, the motor is stopped when the track status is 0



## Phototransistor



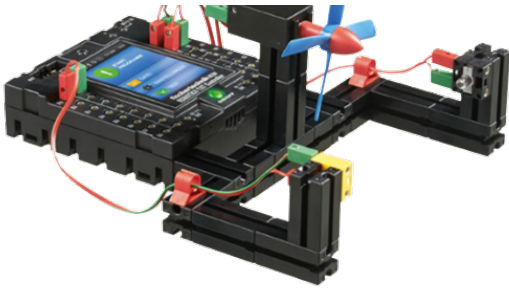
The phototransistor is a digital sensor that differentiates between light and dark.

## Call

**Get phototransistor status** will return 0 if the sensor does not detect any light. If the sensor detects sufficient light, it will return 1.

## Query

To query whether the phototransistor detects light or dark, compare the brightness status with 0 or 1. To do so, use the block **is phototransistor status** []. You can use the drop down menu (small triangle) to select whether to query light or dark.



One possible use for a phototransistor is in a light barrier, as in this

model.

## Photoresistor



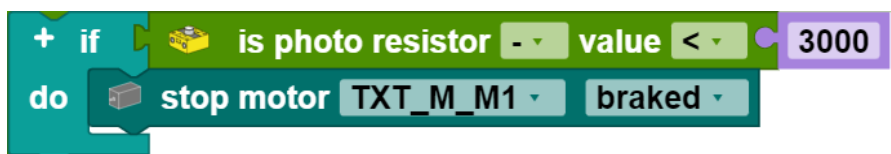
the resistance of the photoresistor drops when it is exposed to more brightness. The value output by the photoresistor, therefore, is a measurement of brightness.

## Call

**Get photoresistor value** provides information on how bright it is. The smaller the output value, the brighter it is.

## Query

To query whether the photoresistor measures a certain brightness value, use the block **is photoresistor value []** ... You can use the drop down menu (small triangle) to select how the brightness value should be compared to the entered value (<, ?, =, ?, ?, >). The comparison value is entered in the number field at the end of the block. This block can be used as a condition. In the example, the motor is stopped when the brightness value is less than 3000.




## NTC resistor




The NTC resistor is a non-binary temperature sensor. Its electrical resistance drops when the temperature increases, and is therefore a measurement for the temperature.

## Call

**Get NTC resistance**  delivers either a resistance value or the temperature calculated from it. You can select what should be delivered using the drop down menu (small triangle).

## Query

To query whether the NTC resistor measures a certain value, use the block **is NTC resistance**  ... You can use the drop down menu (small triangle) to select what should be compared, and with what comparative operator. The comparison value is entered in the number field at the end of the block. This block can be used as a condition. In the example, the motor is stopped when the temperature is less than 20.



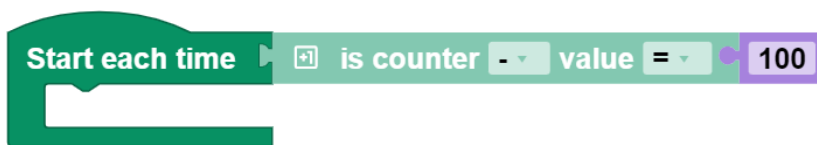
# Counter



The encoder motor can be used as a counter. The number of revolutions is counted, not only when it is turning itself as a motor, but also when it is mechanically driven by an external force.

## The Start each time block

The **Start each time block** offers the option of running a program when a condition is fulfilled. Therefore, it works similar to a case distinction, but runs not only one time, but rather every time the condition is fulfilled during the entire course of the program. The **Start each time block**:



Is an abbreviation for the following construct:



You can insert all conditions from this category into the counter category in the **Start each time** block.

**Note:** The program section in the **Start each time** block should be kept short, and should not contain any blocking calls or endless loops, so that this part of the program can be processed quickly.

## Call

Use **Get counter value** to obtain the value counted by the counter.

## Query

To query whether the counter has counted a certain value, use this





the drill hole. You can use the drop down menu (small triangle) to select how the counted value should be compared to the entered value (<, ?, =, ?, ?, >) .

## Reset

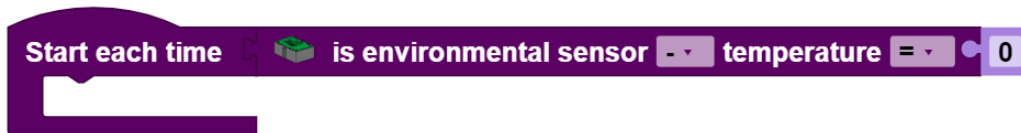
The counter starts over at 0 when the **Reset counter block** is executed.

# I2C

The I2C sensors described in this section are connected to the TXT 4.0 Controller using a suitable ribbon cable.

## The Start each time block

The **Start each time block** offers the option of running a program when a condition is fulfilled. Therefore, it works similar to a case distinction, but runs not only one time, but rather every time the condition is fulfilled during the entire course of the program. The **Start each time block**:



Is an abbreviation for the following construct:



You can insert all conditions from this category into the I2C category in the **Start each time** block.

**Note:** The program section in the **Start each time** block should be kept short, and should not contain any blocking calls or endless loops, so that this part of the program can be processed quickly.

## Combi sensor



The combi sensor combines the three functions of an acceleration sensor, gyroscope, and compass sensor in one component.

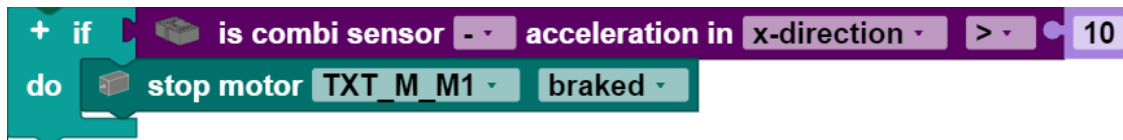
## Acceleration sensor

## Call

Use **get combi sensor acceleration in []** to obtain the acceleration in one spatial direction. You can select the desired spatial direction using the drop down menu (small triangle). The acceleration is indicated in g.

## Query

Use the **combi sensor acceleration in [] [] ...** block to query whether you are measuring a certain acceleration. You can use the drop down menu (small triangle) to select how the acceleration should be compared to the entered value (<, ?, =, ?, ?, >) and what spatial direction should be queried. The comparison value is entered in the number field at the end of the block. This block can be used as a condition. In the example, the motor is stopped when the acceleration in the x direction is greater than 10.



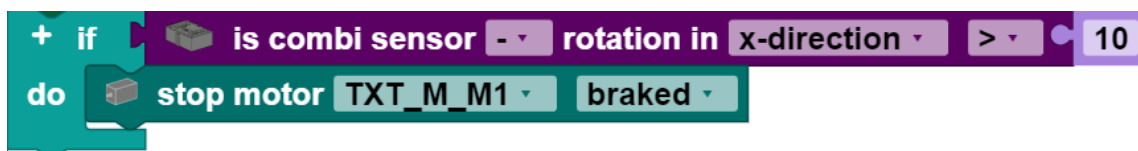
## Gyroscope

### Call

Use **get combi sensor rotation in []** to obtain the rotation in one spatial direction. You can select the desired spatial direction using the drop down menu (small triangle). The rotation is indicated in °/s.

### Query

Use the **combi sensor rotation in [] [] ...** block to query whether you are measuring a certain angular speed. You can use the drop down menu (small triangle) to select how the rotation should be compared to the entered value (<, ?, =, ?, ?, >) and what spatial direction should be queried. The comparison value is entered in the number field at the end of the block. This block can be used as a condition. In the example, the motor is stopped when the rotation in the x direction is greater than 10.



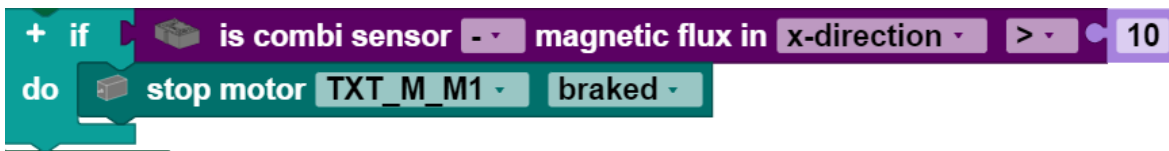
## Compass sensor

### Call

Use **get combi sensor magnetic flux in []** to obtain the magnetic flux in one spatial direction. You can select the desired spatial direction using the drop down menu (small triangle). The magnetic flux is indicated in ?T.

### Query

Use the **combi sensor magnetic flux in [] [] ...** block to query whether you are measuring a certain magnetic flux. You can use the drop down menu (small triangle) to select how the magnetic flux should be compared to the entered value (<, ?, =, ?, ?, >) and what spatial direction should be queried. The comparison value is entered in the number field at the end of the block. This block can be used as a condition. In the example, the motor is stopped when the magnetic flux in the x direction is greater than 10.



# Environmental sensor



The environmental sensor combines the four functions of an air quality sensor, humidity sensor, barometer, and thermometer in one component.

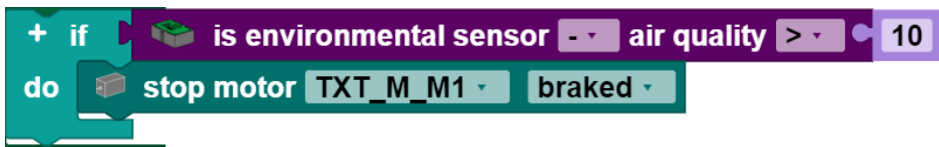
## Air quality sensor

### Call

You can use the **get environmental sensor air quality as []** block to measure the air quality. You can use the drop down menu (small triangle) to select whether the air quality should be returned as a numerical value (from 0 to 500) or as a text.

### Query

Use the **environmental sensor air quality [] ...** block to query whether you are measuring a certain air quality. You can use the drop down menu (small triangle) to select how the air quality should be compared to the entered value (<, ?, =, ?, ?, >) . The comparison value is entered in the number field at the end of the block. This block can be used as a condition. In the example, the motor is stopped when the air quality is greater than 10.



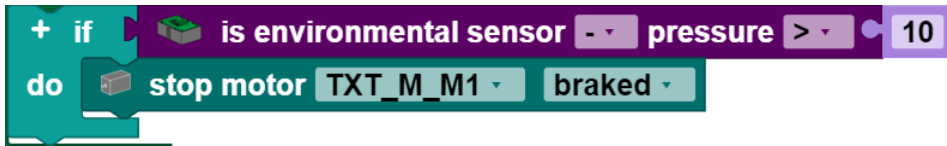
## Barometer

### Call

You can use the **get environmental sensor air pressure block** to measure the air pressure.

### Query

Use the **environmental sensor air pressure [] ...** block to query whether you are measuring a certain air pressure. You can use the drop down menu (small triangle) to select how the air pressure should be compared to the entered value (<, ?, =, ?, ?, >) . The comparison value is entered in the number field at the end of the block. This block can be used as a condition. In the example, the motor is stopped when the air pressure is greater than 10.



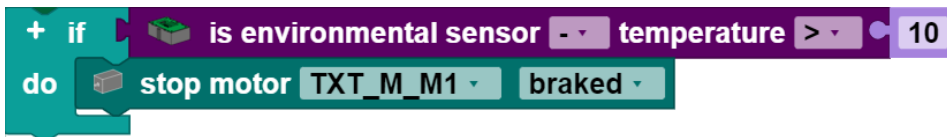
## Thermometer

### Call

You can use the **get environmental sensor temperature** block to measure the temperature.

### Query

Use the **environmental sensor temperature** ☐ ... ☐ ... block to query whether you are measuring a certain temperature. You can use the drop down menu (small triangle) to select how the temperature should be compared to the entered value (<, ?, =, ?, ?, >) . The comparison value is entered in the number field at the end of the block. This block can be used as a condition. In the example, the motor is stopped when the temperature is greater than 10.



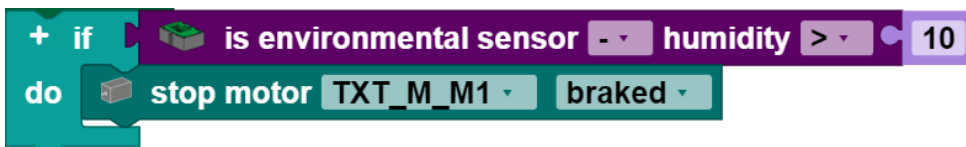
## Humidity sensor

### Call

You can use the **get environmental sensor humidity** block to measure the humidity.

### Query

Use the **environmental sensor humidity** ☐ ... ☐ ... block to query whether you are measuring a certain humidity. You can use the drop down menu (small triangle) to select how the humidity should be compared to the entered value (<, ?, =, ?, ?, >) . The comparison value is entered in the number field at the end of the block. This block can be used as a condition. In the example, the motor is stopped when the humidity is greater than 10.



# USB

You can use the USB port to connect the camera with integrated microphone. The camera and microphone are considered separately here. To use the camera functions, you must first configure them in the camera configurator. [This section](#) explains how to access the camera configurator.

## Camera



The camera can be used, in particular, as a motion detector, a color detector, a ball detector, and a line detector.

### Motion detector

To use the camera as a motion detector, you must drag the little man symbol into the screened area in the camera configurator; then, a window will open at the right side where you can define, under inspector

- the pixel area that will be checked for movement,
- the position of this area (the top left corner of the area is at the point indicated),
- the name of the motion detector, and
- the tolerance

### Motion detector program

The motion detector program runs when a motion is detected. It is written separately from the main program. Variables work globally across both programs. The input program runs in the block **when motion detected**.

### Color detector

To use the camera as a color detector, you must drag the pipette symbol into the screened area in the camera configurator; then, a window will open at the right side where you can define, under inspector

- the pixel area that will be checked for colors,
- the position of this area (the top left corner of the area is at the point indicated),
- the name of the color detector, and
- the contrast

### Call

You can use **get color as []** to obtain the detected colors in a hexadecimal or RGB format. You can set the format using the drop down menu (small triangle).

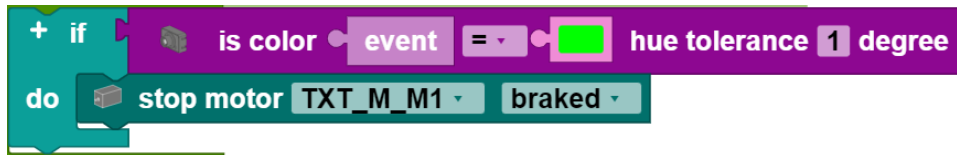
### Query

To query whether the detector detects a color, use the **is color detected** block. This block can be used as a condition.

To query whether the counter has detected a certain color, use this



the drill hole. You can use the block to compare the detected color with the one that was entered. You can use the drop down menu (small triangle) to select whether the set color should be the same as or different from the filmed color. This block can be used as a condition. In the example, the motor is stopped when the color detected is red.



## Color detector program

The color detector program runs when a color is detected. It is written separately from the main program. Variables work globally across both programs. The input program runs in the block **if color detected**.

## Ball detector

To use the camera as a ball detector, you must drag the Ball symbol into the screened area in the camera configurator; then, a window will open at the right side where you can define, under inspector

- the pixel area that will be checked for balls,
- the position of this area (the top left corner of the area is at the point indicated),
- the name of the ball detector,
- the range in which the diameter of the balls is located,
- the range of the x axis,
- the color of the ball, and
- the color tolerance

## Call

Use **get [] ball** to obtain the x position, y position, radius or diameter of the ball.

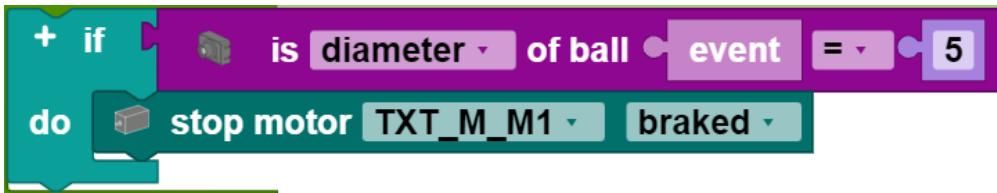
## Query

To query whether the detector detects a ball, use the **if ball detected** block. This block can be used as a condition.

To query whether the detector detects a ball with a certain x position, y position, radius, or diameter, use this



the drill hole. You can use the block to compare the specifications for the detected ball with an entered value. You can use the drop down menu (small triangle) to select what should be compared, and with what comparative operator. This block can be used as a condition. In the example, the motor is stopped when the diameter of the detected ball is 5.



## Ball detector program

The ball detector program runs when a color is detected. It is written separately from the main program. Variables work globally across both programs. The input program runs in the block **if ball detected**.

## Line detector

To use the camera as a line detector, you must drag the symbol with the dots on a line into the screened area in the camera configurator; then, a window will open at the right side where you can define, under inspector

- the pixel area that will be checked for lines,
- the position of this area (the top left corner of the area is at the point indicated),
- the name of the line detector,
- the number of lines to be detected, and
- the range in which the width of the line(s) lies

## Call

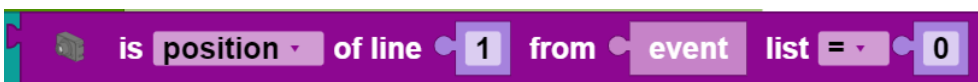
Use **get [] the line []** to obtain the position or width of one of a maximum of five lines.

Use **get color of the line [] as []** to obtain the color of a line output in hexadecimal or RGB format. You can set the format using the drop down menu (small triangle).

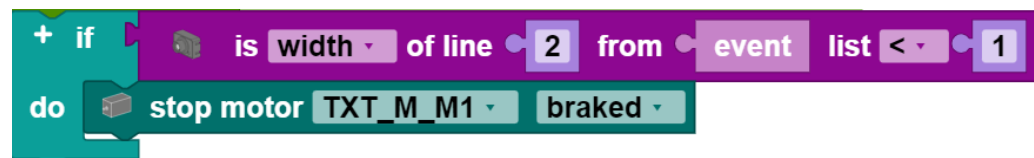
## Query

To query whether the detector detects a line, use the **if line detected** block. This block can be used as a condition.

To query whether the detector detects a line with a certain position or width use this



the drill hole. You can use the block to compare the specifications for the detected line(s) with an entered value. You can use the drop down menu (small triangle) to select what should be compared, and with what comparative operator. This block can be used as a condition. In the example, the motor is stopped when the width of the detected line is less than 2.



To query whether the detector detects a line with a certain color, use this





the drill hole. You can use the block to compare the detected line color with the one that was entered. You can use the drop down menu (small triangle) to select whether the set color should be the same as or different from the filmed color. This block can be used as a condition.

## Line detector program

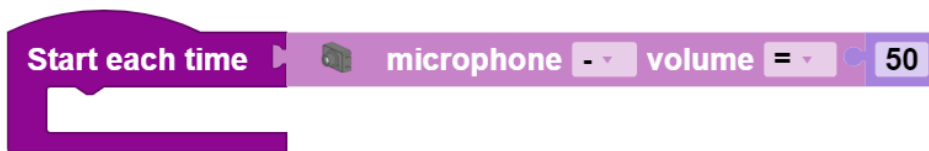
The line detector program runs when one or more lines have been detected. It is written separately from the main program. Variables work globally across both programs. The input program runs in the block **if line detected**.

# Microphone

The microphone integrated in the camera can be used as a volume detector.

## The Start each time block

The **Start each time block** offers the option of running a program when a condition is fulfilled. Therefore, it works similar to a case distinction, but runs not only one time, but rather every time the condition is fulfilled during the entire course of the program. The **Start each time block**:



Is an abbreviation for the following construct:



You can insert all conditions from this category into the microphone category in the **Start each time** block.

**Note:** The program section in the **Start each time** block should be kept short, and should not contain any blocking calls or endless loops, so that this part of the program can be processed quickly.

## Volume detector

### Call

**Microphone volume** provides the volume in decibels.

### Query

To query whether the volume detector detects a certain volume level, use this



the drill hole. You can use the block to compare the detected volume with the one that was entered. You can use the drop down Menu (small triangle) to select what comparative operator should be used for the comparison. This block can be used as a condition.