Camera man

With this model, you will learn some initial fundamentals of programming using the camera.

If you have built the model using the building instructions, then you have already noticed one all new component - the camera.

The camera is a particularly versatile sensor. The image resolution is 1 Megapixel. The images the camera takes can be transmitted to a PC and displayed on the screen. In addition, the TXT 4.0 Controller can process the images and use them to detect movements, colours and tracks so that the user can control the robotic model. However, it is also possible to process the data delivered by the camera using the ROBO Pro Coding software. The camera can be focused by turning the mount.

Task 1	Programming level 2 The goal is to detect movements within the camera's image. If a movement is detected, both an acoustic and a visual signal should be output to the TXT 4.0 Controller.	Hyperlink to Task 1
Task 2	Programming level 2 3 different colours (red, yellow, green) should be detected (coloured plates) The colour detected should be displayed on the TXT 4.0 Controller.	Hyperlink to task 2

Aufgabe1

Start ROBO Pro Coding

Start ROBO Pro Coding and select a template for your new project.

Since you need both the camera and the display, activate the selection:

Kamera + Anzeige

and confirm by pressing : NÄCHSTE

You also need to assign the project a name, such as: **cameraman_alarm**

Controller configuration

Here as well, as with the function model you need to define the connected actuators and sensors. Select the “**Controller configuration**” window and drag the controller symbol from the “**Controller**” block into the work area.

Only the element “**Camera**” needs to be integrated from the “**USB**” group.

The controller is then configured.

There are two other entries in the project window:

Here as well, you need to complete the different entries and adjustments.

First, click on the camera configuration.

The following window will appear.

5 detectors are shown to the right of the window. These are:

Colour detector

Motion detector

Ball detector

Line detector

Blocked area

If you need one of these detectors, drag it to the coordinate window using your mouse (see image). The entry “Inspector” is in the right corner above the code window. Activate the entry and a window will appear where you can define the detector in more detail.
You can change the size, location, name and contrast here.
If you want to delete the window, you have to mark it (green frame points), then press the right mouse button and “Delete” in the selection window.

For the first task, you will need a window where a movement can be detected. To do so, drag “Movement detector” into the coordinate window.

If a movement is detected later in this window, additional commands can be used to trigger an action, for instance to output a warning signal.

In the second task, your goal is to detect a colour. You will need to use the colour detector to do so.

In both tasks, you will need the display of the TXT 4.0 Controllers.

To configure it, activate the button

Different elements are available here on the left side to design the display view. These are:

TXTButton - confirmation button

TXTSwitch - slide switch

TXTCheckBox - control field

TXTSlider - slider

TXTLabel - text

TXTInput - text input

Status indicator

Gauge - measurement device display

Drag the status indicator from the selections to the screen.

You can use the green points to enlarge the status display to fill the format.

The entry “inspector” is at the left top corner of the screen. If you access it, you can also change additional settings here.

You need to switch from “active” to “not active” in the “status indicator” area by removing the tick.

Important: You do not have to switch on the camera using a separate command. If it has been defined, then it is activated automatically.

Now, you can start programming.

First, insert a wait command into the program start from the “Util” block with a time specification of “3” seconds. The purpose of the wait time is only to have a short waiting time after transmission to the model until the following commands are executed.

Then, add a command from the “Display” group.

If a status indicator has been created in the system configuration, it will appear in the “set status indicator” command, and can be selected there. You only need to change the logic output from “true” to “false”. This will switch off the indicator on the display.

To ensure the program is executed continuously, add the command “continuously repeat – do” from the “loops” group

The first part of the program is then complete, and you can turn to the actual query and analysis program itself.

According to the definition of the camera, the “motion_detector” will be queried.

You will find the command “If movement --- detected” for this purpose in the “USB” group. Drag the command under the program start.

Here as well, the entry “motion_detector” will be entered automatically.

What should happen when the camera detects a movement?

The status indicator on the display should be activated. Do so by setting the status indicator to “true”. Integrate this command.

This will look as follows in the program:

Next, stop the sound output. The command “stop sound output” is available in the “sound” group. Add the command to the program.

Then, activate a sound. To do so, insert the command “xxx play file” from the “sound” group. When you press the button, a list of different sounds appears. Simply select a suitable sound.

You can define the duration of the sound using the command “wait until” from the “Util” group and the command “play sound” from the “sound” group.

Now, you only need the command “set status indicator – false” to turn the indicator back off again. Simply copy it by right clicking on the previously defined command and moving it to the last line of the program. Change the value “true” to “false” here.

The program is now complete, and you can test it out.

To do so, use your mouse to click on the command “Start program” in the right top blue bar.

The program will be transmitted to your TXT 4.0 Controller and started.

If you look at the display, the non-active status indicator is shown there.

If the camera detects a motion, the display will switch to the status “true” and a sound will be played over the speaker on the TXT 4.0 Controller.

The program runs in an endless loop. If you want to stop it, click the command “Stop” in the blue bar.

Task 2

In the second task, the goal is to detect different colours and show them on the display. You can use the coloured plates from the building set as coloured surfaces.

The controller configuration is the same as for the first task.

You need to make a couple of changes to both the camera and display configuration.

First, call up the camera configuration. Replace the motion detector with a colour detector.

You can also change two settings in the inspector. Change the size to “60/60”

and the position to “130/90”. You can also enter your own values and experiment a little here.

Then, switch to the display configuration.

Instead of one display, you will need 3 displays for this task. Delete the one and add 3 status indicators as shown in the image.

You can set the size in the inspector:
Size: 60/60 and the respective position: 20/40, 90/40 and 160/40. Switch the status indicator to “not active”.

Use a unique name for the specific colour

Identity: For green: txt_status_indicator_green
For yellow: txt_status_indicator_yellow
For red: txt_status_indicator_red

Now, change the specific colour for the display of the status indicator. To do so, click the coloured area and select the appropriate colour, either using the slider or by using the following values

Green: #61996a, Yellow: # b88124, Red: #85131d

Conclude the entry or colour change with “Save”.

You can take the first part of the programming from task 1. Now, insert two additional “set status indicators” and change “active” to “false”.

Now, you can build the program used to determine the colour. To do so, you need the command “If colour --- detected” from the “USB” block.

Then, insert a command you are already familiar with “if --- then – otherwise” from the “Logic” group.

Since you need to query three colours, you will need it three times. Click the + sign twice.

The query is expanded as shown in the image.

Add the query after the individual colours in the docking points. The command “if colour – event = colour” is found in the “USB” group

Copy the command to the other two docking points. Change the colour to yellow and red there. The colour codes are provided in the display configuration.

Now, the status indicators for the individual colours need to be defined. To do so, you need the command “set status indicator ...” from the “Display” block. Insert the command at the free points in the program.

Insert it once and then copy it to the other points.

Also change the colour tolerance to 0.7 here for green and red as well as 0.9 for yellow. This will cause the detection to work very reliably.

You need to switch the status indicators for each detected colour on or off. To do so, you need two “set status indicator” commands in each query. Simply copy them to the relevant points.

First, change the display (arrow and selection) and then the “active setting” for the specific colour.

Green: false, false, true
Yellow: false, true, false
Red: true, false, false

Otherwise: false, false, false

The program is now complete and you can start it. The associated indicator is activated on the display of the TXT 4.0 Controller, based on which colour is detected.

Add the “Hexadecimal indicator” and “RGB indicator” for the detected colour to the program.

The system of **hexadecimal colour definition** is found in many areas of computer-supported design. In this system, a colour is represented by three hexadecimal numbers in a row, each of which stands for one colour in the RGB colour spectrum.

RGB is an abbreviation for the percentages of Primary colours red, green and blue in the blended colour.

Grundfarben

Farbe	Hex-Wert	R	G	B	Komplementär zu
Schwarz	#000000	0	0	0	Weiß
Grau	#888888	136	136	136	#777777
Weiß	#FFFFFF	255	255	255	Schwarz
Rot	#FF0000	255	0	0	Cyan
Dunkelrot	#880000	136	0	0	#77FFFF
Grün	#00FF00	0	255	0	Magenta
Dunkelgrün	#008800	0	136	0	#FF77FF
Blau	#0000FF	0	0	255	Gelb
Dunkelblau	#000088	0	0	136	#FFFF77

The hexadecimal colour definition is commonly used in a six-digit form, which means that it is a sequence of three, two digit hexadecimal numbers written according to the scheme of: #RRGGBB, which makes it possible to define different states for each Object (= 256) – from 0 for a colour that is completely “switched off” to 255 (FF) for 100 % colour saturation. The additive colour mixture is used to mix the colours, the code #FFFF00 results in the colour yellow, because red is mixed with green.

Add the window with a text label and a text window where something is displayed to the display configuration.

Position both elements under the colour display. You can use the following coordinates: Open the inspector to do so.

Size: 40/40, Position: 20/150, Identity Name: text_label, TXTLabel Text: RGB
Size 110/40, Position: 110/150, Identity Name: txt_label_RGB, TXTLabel Text: #

The addition to the indicator configuration is now complete, and you can add a program line to your program.

Add the command “Set label field --- Text – abc“ from the “Display” block after the “if otherwise” command.

Instead of ABC, add the command “high colour event as RGB” from the “USB” block. The RGB value for the detected colour is shown on the display.

The program is now complete, and you can test it using your model.

Then, disassemble the model and turn to the next task.