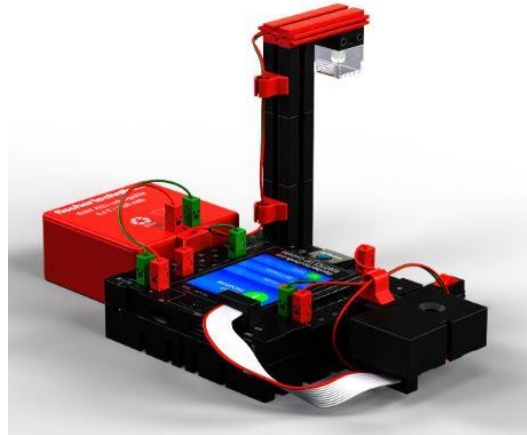




Twilight switch



You will use this model to learn about the fundamentals of programming and about the elements used in the model, in particular the gesture sensor.

The following sensors and actuators are used in the model:

Gesture sensor	LED (actuator)
	

An explanation of the components is provided on the introductory page.

The “Dimmer switch” model is divided into 2 programming tasks:

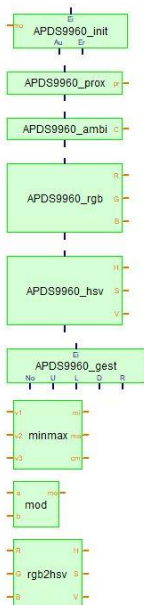
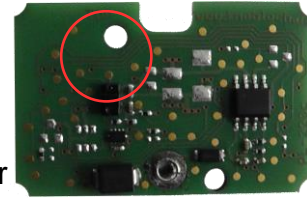
Task 1 Twilight_switch_1.rpp	<u>Programming level 3</u> The LED will switch on or off at a specified level of darkness.
Task 2 Twilight_switch_2.rpp	<u>Programming level 3</u> The LED will switch on or off at a specified level of darkness. When the LED is switched on, it should flash in a 1 second interval.

Task 1

Before you start programming, here is some information on the

“**Gesture sensor**” module.

Actually, the APDS-9960 sensor is a small, rectangular black electronic module with two lenticular openings which is installed on the electronic circuit board with other components.



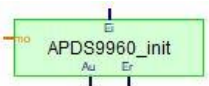
Why is this module used in fischertechnik models? It combines distance, movement, colour and light sensors in a very compact space.

There are pre-programmed subroutines under ROBO Pro for this module, which you can integrate into your control program as needed.

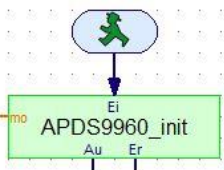
Start ROBO Pro and open the folder library – I2C - “RGB-gesture-sensor-APDS9960” on the left side of the screen under the element groups.

All available program blocks are shown in the program elements area.

You must initialise the sensor each time the program is started. To do so, use the command block “APDS9960_init”.

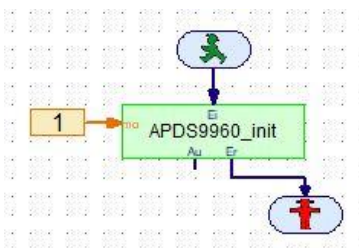


Start with task 1 - with the “Start” command block, of course, followed by “APDS9960_init”.



0

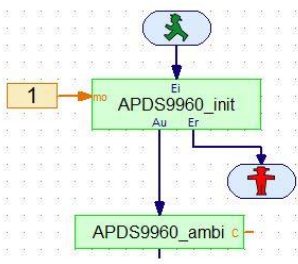
Then define the outputs. If you are programming the model as a twilight switch in your program, then you must enter a variable value for mode on the orange input “mo”. In your case, enter a “1” for the function “light enabled”.



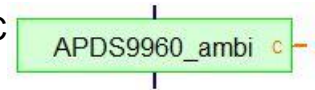
To do so, drag the command “Constants” from the element group “Variable, Timer, ...”. Change the value “0” to “1”.

Drag the “End” command to the command output “Er” for “Error”.

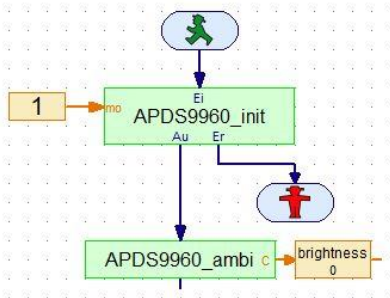
You will then need to define the output “Au”.



You will need the module “APDS9960_ambi” from the I2C programs to do so. Drag it into your program.



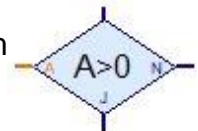
This block has two docking sites, the block output and the data output “c”. There is a variable measured value for brightness at the data output “c”, which is evaluated later in the program.



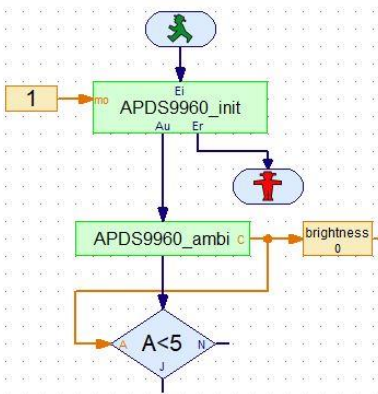
Connect a “variable” from the “Variable, Timer, ...” command block at this output. Change the name of the variable to “brightness”. Switch from “whole numbers” to “floating decimal”.

You must insert a query for the measured value at the output of the “APDS9960_ambi”.

Do so using the “Branch (with data input)” command from the element group “Branch, wait, ...”.



Now, connect the orange output “A” with the input for the variables. Change the value “A>0” to “A<5” and to the floating comma 48bit (decimal points) (in the module dialogue field). You can experiment with this value a little to test out the switching behaviour of the LED.

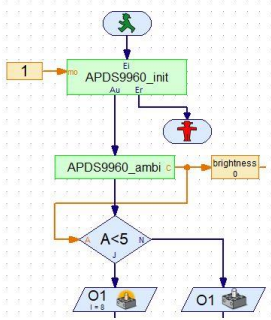


In the next programming step, you must define the two outputs for the query.

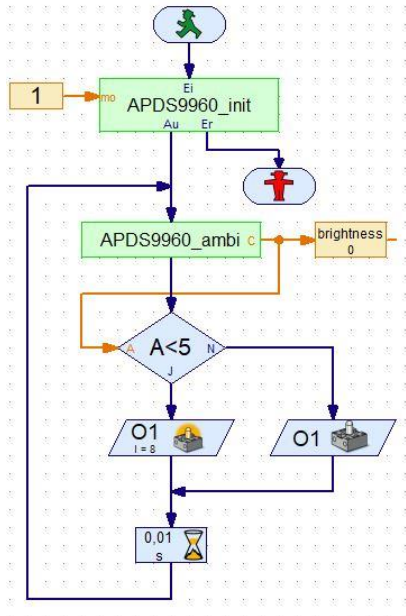
- If the value for the query is “A<5” (if A is less than 5), then the LED should switch on
- If the value for the query is “A> 5” (if A is greater than 5), then the LED should switch off



Drag the “Lamp output” module to your program twice from the program elements - basic elements, and connect its inputs to the relevant outputs of the query.



Before you complete the program with a loop, insert a “wait element” after the two light outputs, and change the wait time to “0.01 s”. This means that a brightness measurement is conducted after each 0.01 second.



Now, test out the program and save it on your computer if no errors occur.

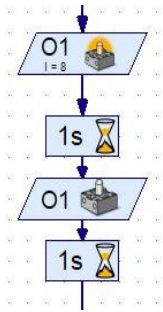
Do so using “File” - “Save as...” - select the storage location - assign a file name - end with “Save”.

“Twilight_switch_1”

Important: The LED may flash during the test. This may be because of scattered ambient light. In this case, just try again in a relatively dark room.

Task 2

For the 2nd task, modify your program so that the LED flashes when it is in light mode. To do so, you will need to integrate the LED into the sequence, insert a wait time of 1 second, and switch off the LED again.



Test out the program and save it under the name

“Twilight_switch_2.rpp”

on your computer.

Then disassemble the model and build the next model.