Name: \_\_\_\_\_\_\_\_\_\_ Class: \_\_\_ Date: \_\_\_\_\_\_\_\_

# Solution sheet task 3:

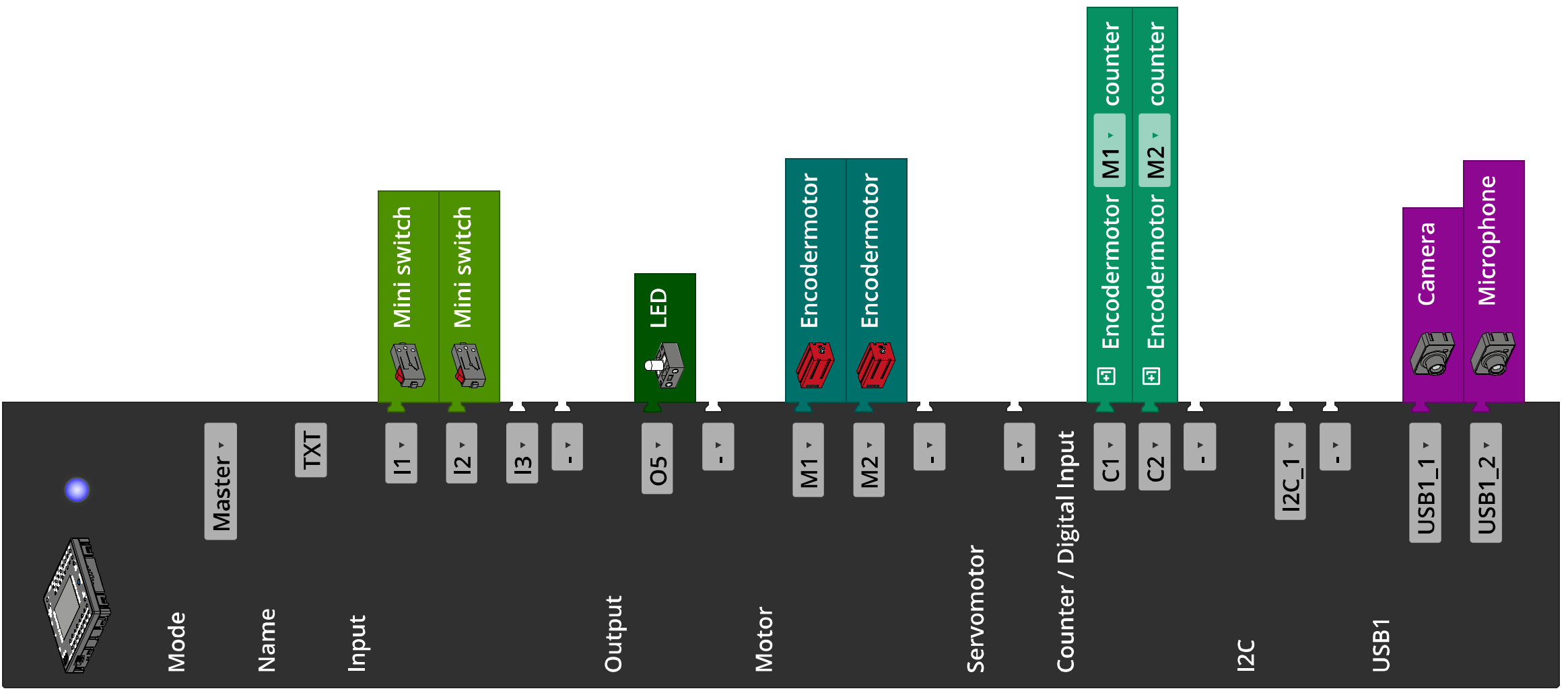
# Alarm system

## Construction task

See building instructions.

## Programming tasks

Configuring the sensors and actuators:



The “Voice Control” app (for iOS or Android) is required to solve experimental task 2. The app must be connected to the internet for voice recognition, and connected to the Controller (via Bluetooth or WiFi).

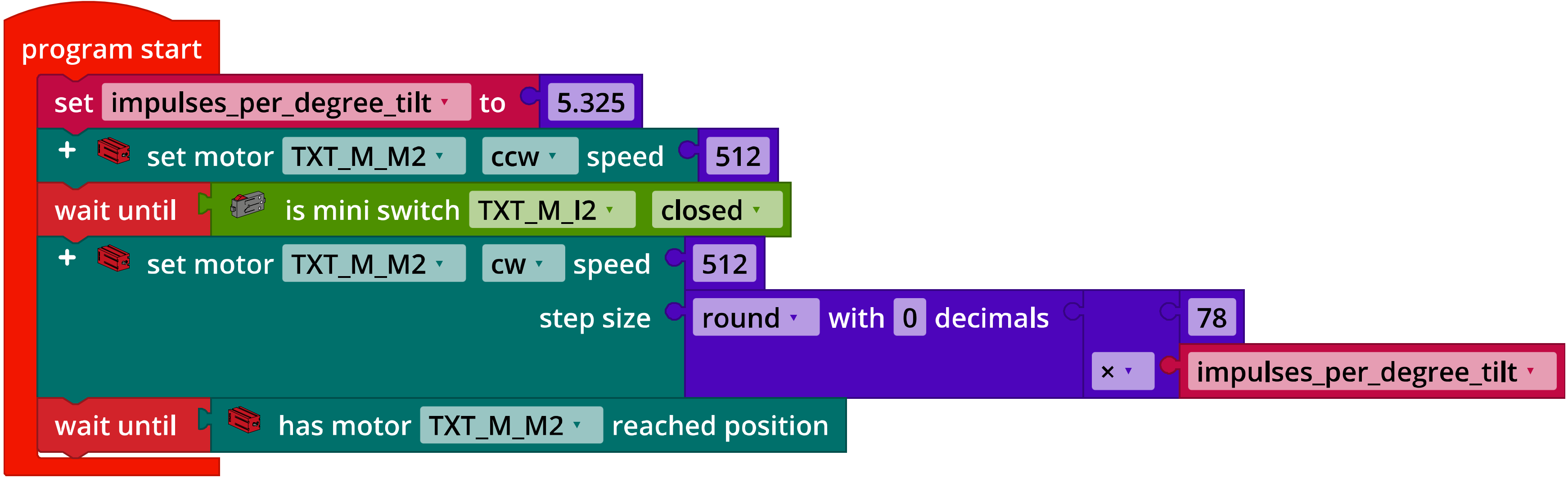
**1. Home position**

1a. Each rotation of the motor axis corresponds to one rotation of the worm, and each rotation of the worm turns the Z30 by one tooth. Therefore, 30/4 rotations of the motor axis are required to turn the Z30 by 90°. This corresponds (at 63.9 encoder pulses per axial rotation)

Encoder pulses (or 5.325 pulses per angle degree).

In reality, the end position switch is not triggered exactly after one quarter turn; the camera must be turned only around 78°. This corresponds to encoder pulses.

Program (example):



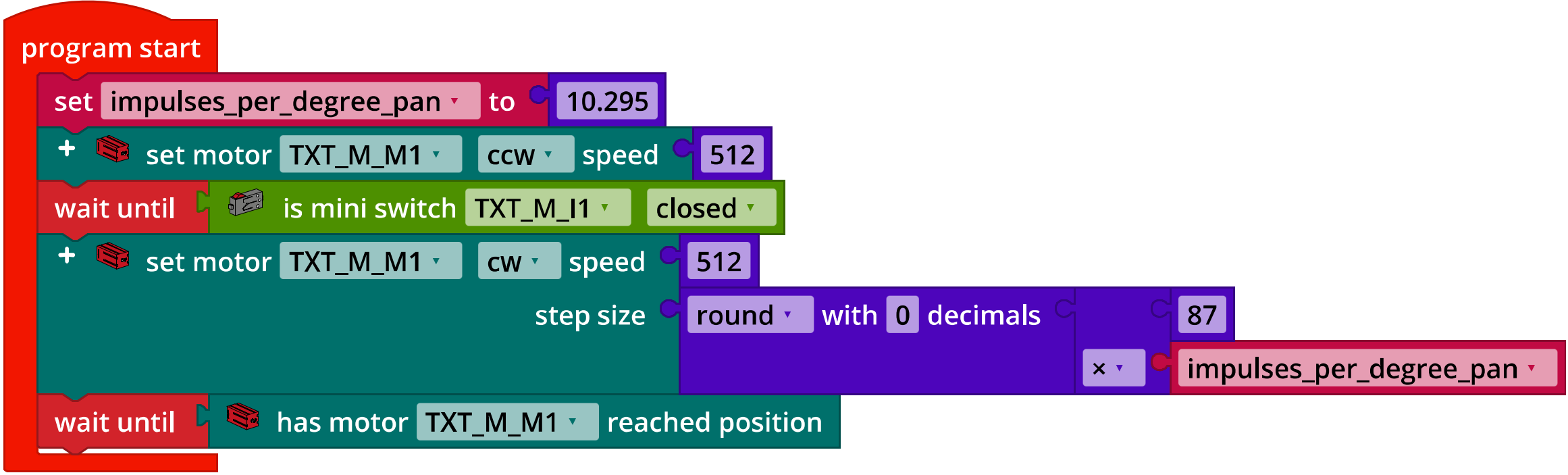
*IoT\_Init\_Camera\_Tilt.ft*

1b. The slewing ring has 58 teeth; a 90° turn, therefore, requires 58/4 rotations of the motor axis. This corresponds to

Encoder pulses (or 10.295 pulses per angle degree).

Here as well, the end position switch is not triggered only after a 90° turn, it is triggered already at approx. 3°; the camera must be turned by only around 87°. This corresponds to encoder pulses.

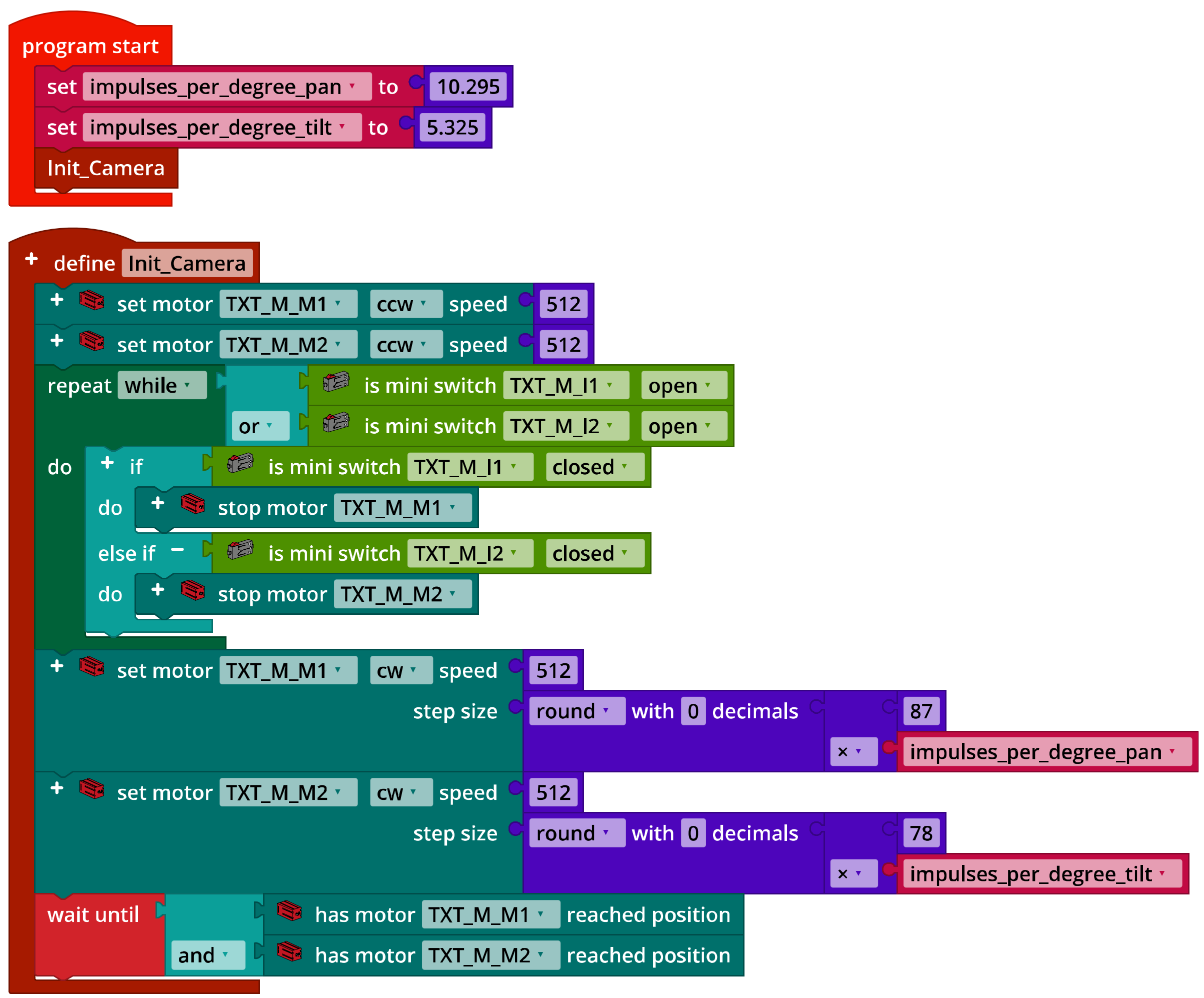
Program (example):



*IoT\_Init\_Camera\_Pan.ft*

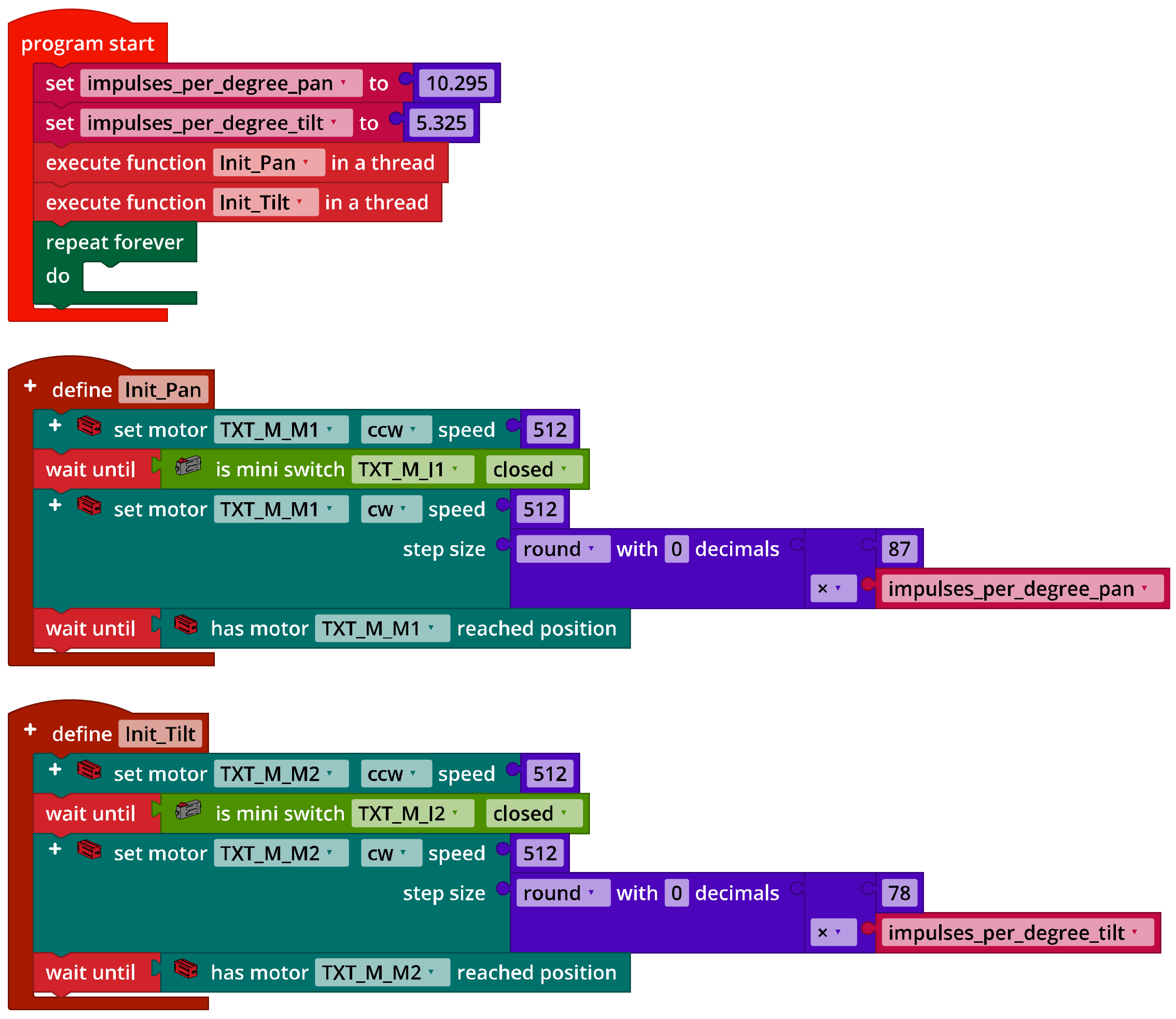
1c. When combining both programs, the two movements can be conducted in parallel, thereby significantly shortening the initialisation time. There are two potential solutions for this: Execute the two functions as parallel threads, or integrating them into one function.

Program solution method A – one function (example):



*IoT\_Init\_Camera\_A.ft*

Program solution method B – parallel threads (example):



*IoT\_Init\_Camera\_B.ft*

**2. Camera surveillance**

Program (example):

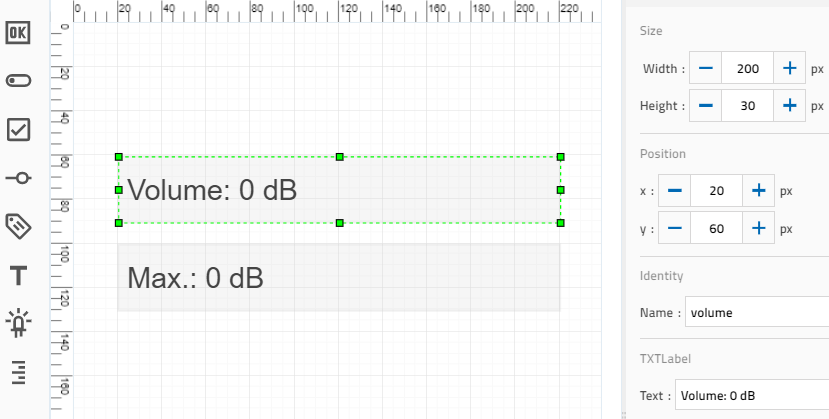
Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

*IoT\_Surveillance\_Camera.ft*

**3. Noise activation**

3a. Configuration of the TXT display (example):

Display configuration

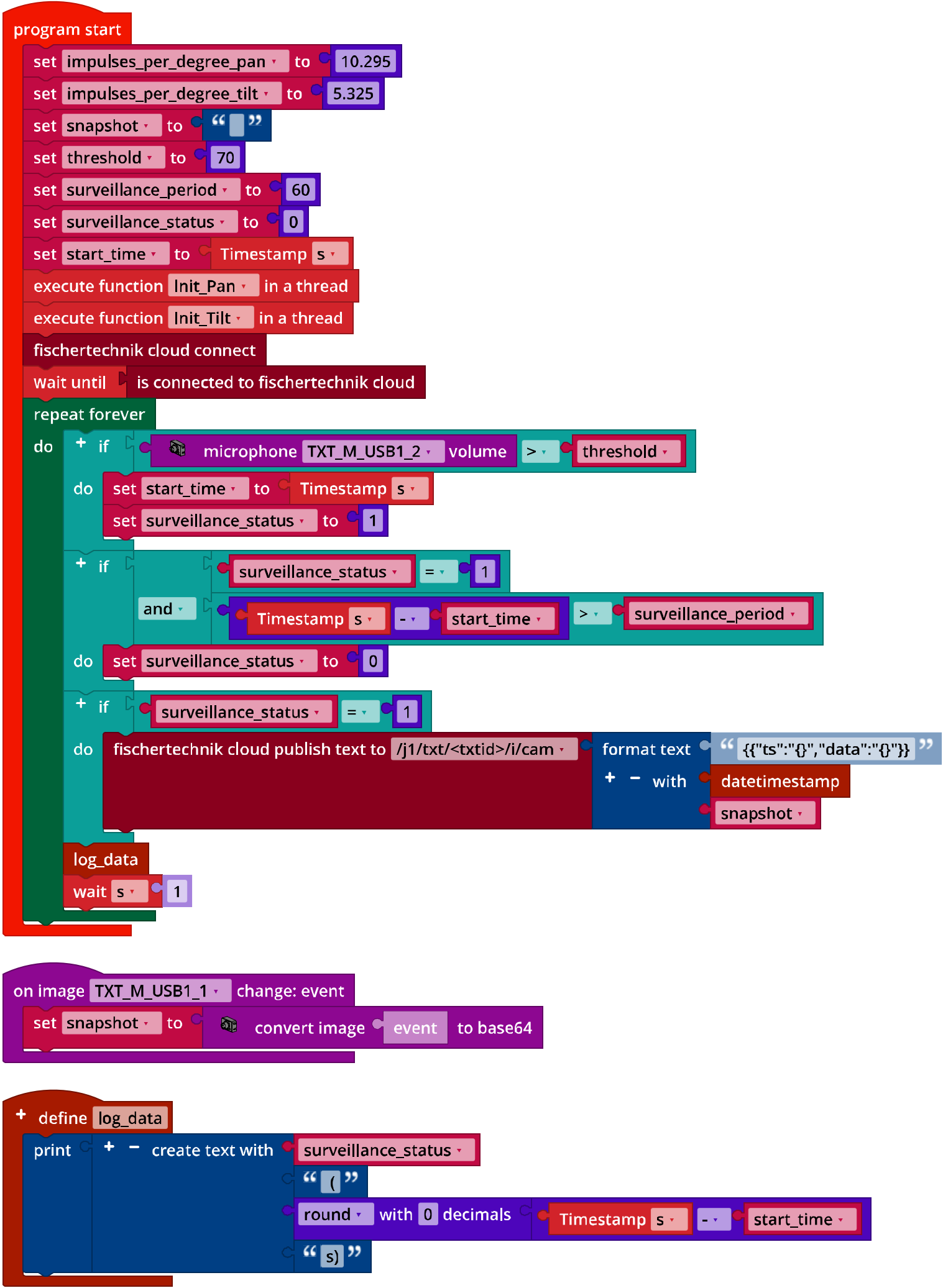
Program (example):

Ein Bild, das Text, Screenshot, Monitor, Bildschirm enthält.

Automatisch generierte Beschreibung

*IoT\_Microphone\_Level.ft*

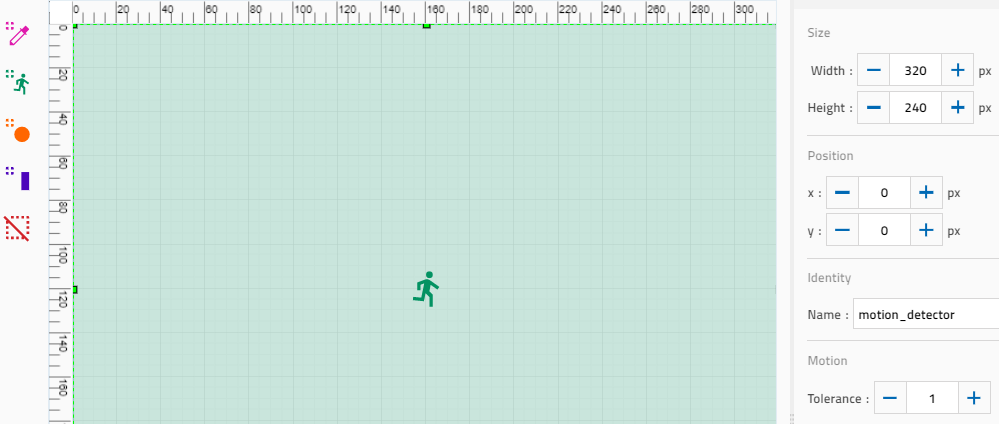
3b. Program with volume threshold 70 dB (example):



*IoT\_Surveillance\_Camera\_Noise\_Detection.ft*

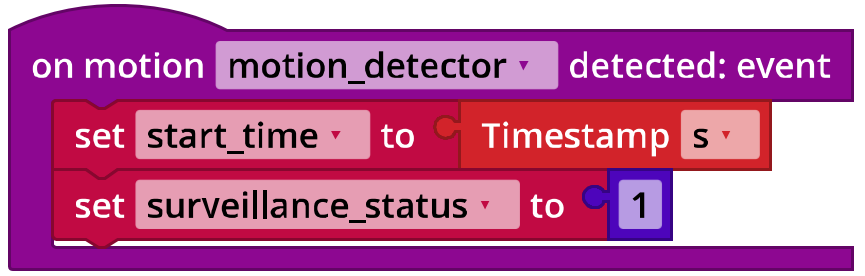
**4. Motion detection**

Motion detection is defined over the entire camera window. The “tolerance” can be used to set the sensitivity of detection to a value of 0 to 1 in the inspector:



Configuring motion detection

4a. Program excerpt (example):



*IoT\_Surveillance\_Camera\_Motion\_Detection.ft*

If a motion is detected, this is saved in a flag (“surveillance\_status”), along with the time of detection.

4b. The “flash” function is started as a thread at the beginning of the main program:

Ein Bild, das Text, Screenshot, Schild enthält.

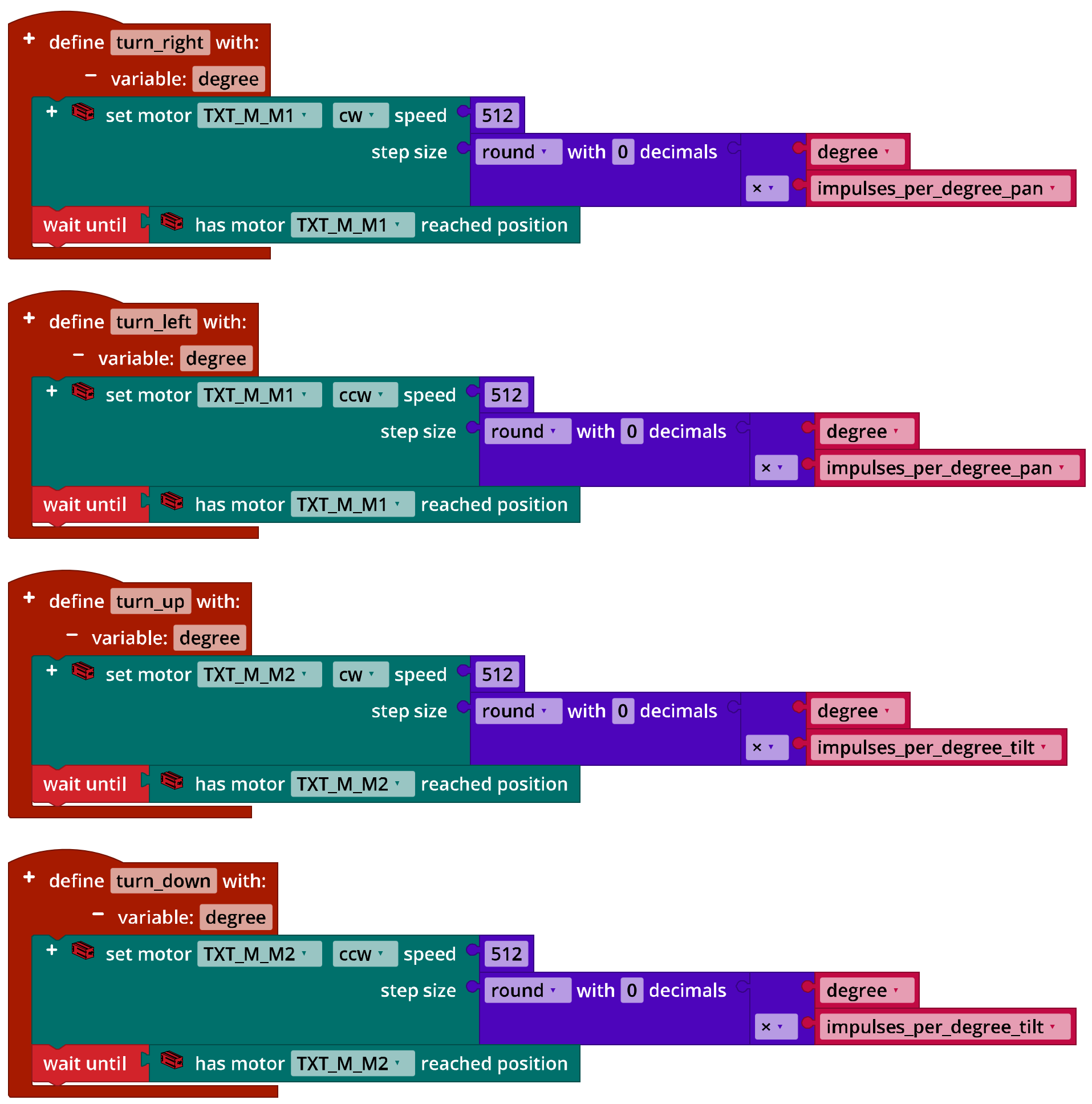
Automatisch generierte Beschreibung

*IoT\_Surveillance\_Camera\_Motion\_Detection.ft*

## Experimental tasks

**1. Voice control**

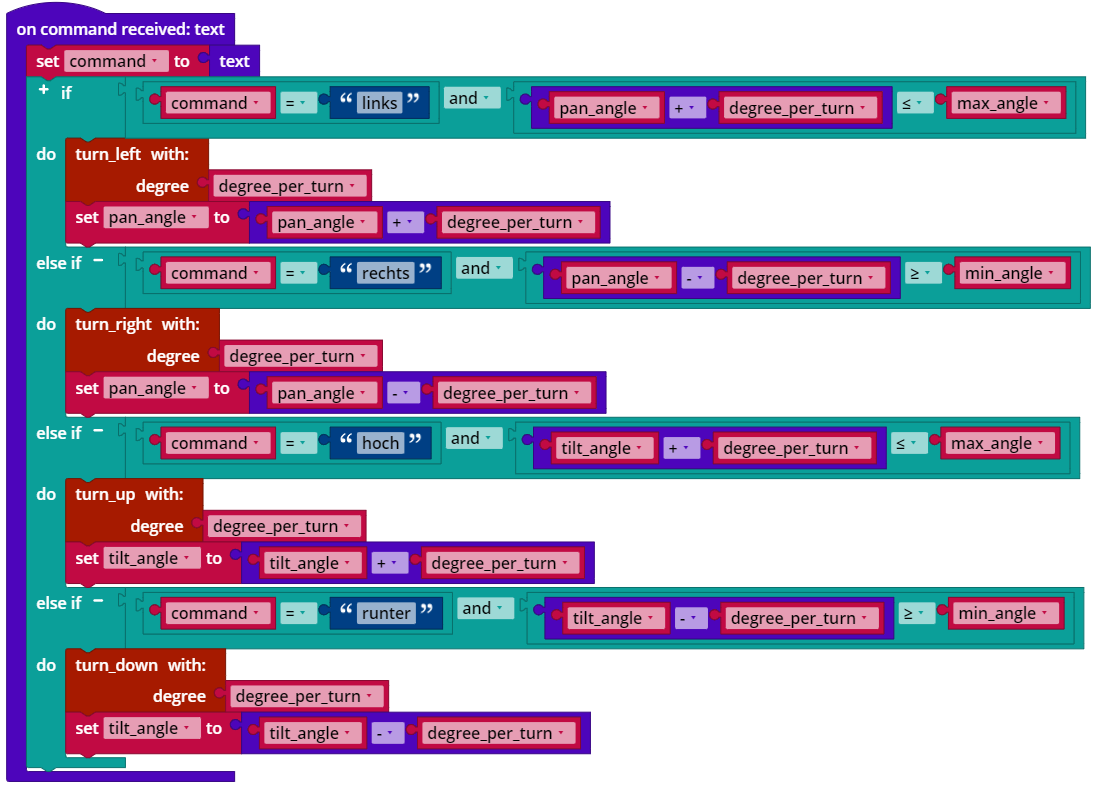
1a. Control functions (example):



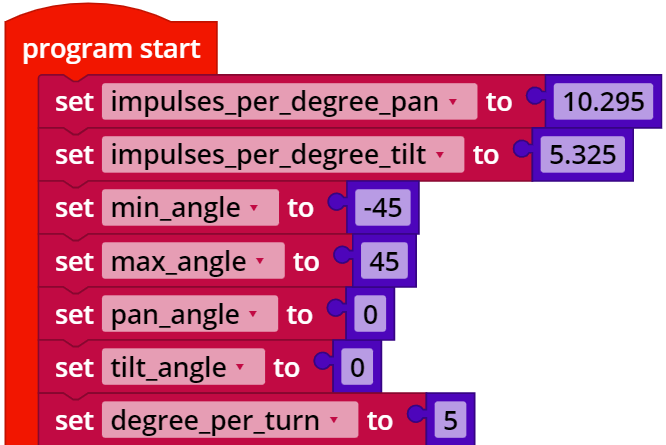
*IoT\_Camera\_Motion\_Control.ft*

The easiest way to prevent the camera from turning too far is by limiting the rotational angle in both the horizontal and vertical direction to a total of ±90° each. If you set both deflection angles to 0° after initialising home position, then the program is easy to check.

1b. Program excerpt: Camera control (example):

*IoT\_Surveillance\_Camera\_Voice\_Control.ft*

The following variables are initialised at the beginning of the main program:



*IoT\_Surveillance\_Camera\_Voice\_Control.ft*

The maximum and minimum deflection angles should be adjusted to the design if necessary (cabling range).

**2. Controlling the camera using the cloud dashboard**

The control knobs on the dashboard deliver the commands “stop”, “home”, “relmove\_left”, “relmove\_right”, relmove\_up”, and “relmove\_down”. The “relmove” commands correspond to the four control functions from experimental task 1. If a value “degree” (2, 5, 10, or 20°) is transmitted by the IoT server, the rotational angle specified in the program must be adjusted. The “home” command moves the camera to the home position; a function should be added for the “stop” command that stops the two motors.

The query of the control command can be taken from the solution to experimental task 1.

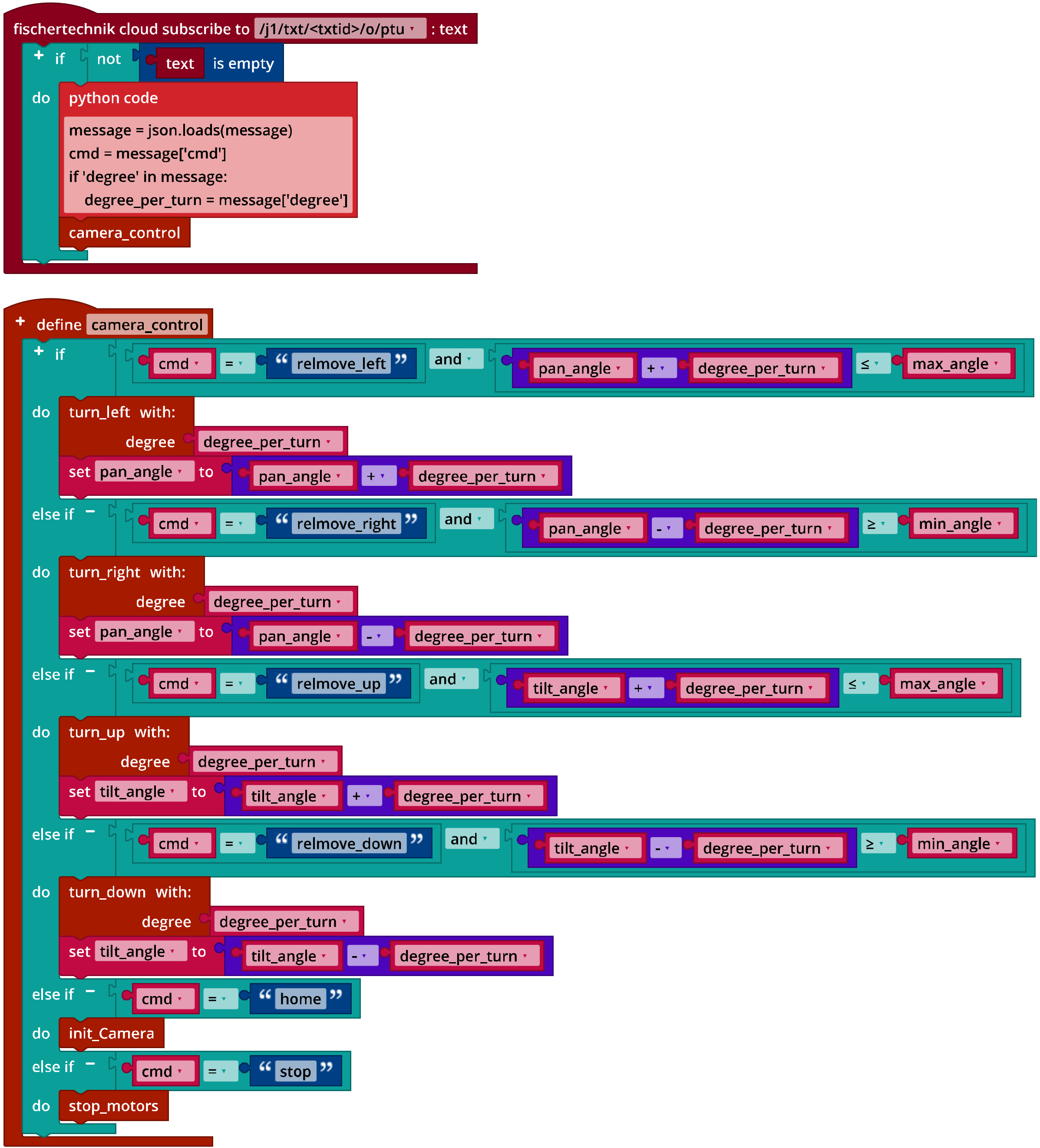
“stop\_motors” function (example):

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

*IoT\_Surveillance\_Camera\_Dashboard\_Control.ft*

Program excerpt: Camera control (example):



*IoT\_Surveillance\_Camera\_Dashboard\_Control.ft*

Annex

# Task 3: Alarm system

## Required materials

* PC for program development, local or via web interface.
* USB cable or BLE or WiFi connection for transmitting the program to the TXT4.0.
* fischertechnik “Voice Control“ app
* Auxiliary program “IoT\_Test\_Dashboard\_Control.ft”
* Account in the fischertechnik cloud ([www.fischertechnik-cloud.com](http://www.fischertechnik-cloud.com))