Ball robot

This model uses a camera to detect a coloured target, and has a mechanism to shoot a ball to hit the target.

Build the model using the building instructions and wire the sensors and actuators as described in the wiring diagram.

The “Ball robot” model is divided into 5 programming tasks:

omniwheels_x4_ball_simple.ft	Function – Load ball and fire using the servo
omniwheels_x4_ball_gui.ft	Travel sideways, load the ball and shoot using buttons on the display of the TXT 4.0 Controller.
omniwheels_x4_target_detect.ft	Detect targets (red, yellow or green panels), align the robot to the target. The colour is selected using the buttons. The robot searches for the selected target
omniwheels_x4_ball_target_detect.ft	Red, yellow and green panels are detected and a shot is made at each – one after the other.
ommniwheels_x4_ball_voice_control.ft	Remote control program with voice input – voice control of the travel commands and shots

Before you start programming the individual tasks, start a brief introduction to the new component – the servo motor.

There is a servo motor, or a servo for short, integrated into the model. It consists of a DC motor, a gear and control electronics.
The connection cable consists of three lines. These are the Vcc power supply (red), the GND ground (brown) and the signal line PWM (yellow).

The servos are controlled via pulse width modulation “PWM”.

The width of the pulse is used to control the angle at which a servo motor should be set. One pulse period is 20 ms. At a pulse width of 1 ms, the motor turns to 0°. At a pulse width of 1.5 ms it turns to 90°, and at a pulse width of 2 ms it turns to an angle of 180°.

This function is used to move the servo lever to specific positions.

A total of three servos can be connected to the TXT 4.0 Controller.

Important: When installing the servo, you must follow the steps exactly as described in the building instructions. When the servo is connected to servo output S1, the servo will move to the centre position when the Controller is supplied with power. Then you can install the servo lever so that it is in the middle of the swivel range. The zero positions for different servos can deviate slightly. You can compensate for this via programming. Further details are provided in the program description.

Controller configuration

First, start the “ROBO Pro Coding” program and assign a name to the project:

“omniwheel_x4_ball_simple.ft”

Complete the Controller configuration.

You will need the 4 encoder motors and the associated counter for the tasks. You will also need a servo motor.

Configure the Controller accordingly.

<u>Task 1</u>

Then, switch to the main program. This first program can be used to test the shooting mechanism. Create a function named “fire”. First, the shooting lever of the servo is pushed in front of the catapult strut. Then the program waits two seconds before turning the shooting lever to the “110” position. This tensions the catapult strut and builds up energy. A ball also falls in front of the strut from the magazine at the same time. The lever is turned until the shooting lever triggers.
This shoots the ball forward out of the catapult. Servo position “230” is used as a block for balls still in the magazine.

The position of the servo can be adjusted to define the exact zero position and achieve optimal results for shooting.

The main program is very simple, and consists of only the program start and a loop command. The function block “fire” is inserted into it. The robot should run through the loop once first.

Save the program under Project, – export to your computer and test it out.

If everything worked correctly, you can turn to task 2.

Task 2

Load the last project, since you can take some data from it for the new program (such as the “fire” function).

Save it under the new name **“omniwheels_x4_ball_gui”** using Project – export.

Since the ball robot should be controlled via the TXT 4.0 display, you need to add to the project. To do so, activate “New file” and then “Display”.

Confirm this by pressing “CREATE”.

Next, activate the display configuration.

Define the following displays and buttons on the display window.
The ball robot is moved sideways using the first button, the TXT button “sideways”. Place the TXT button on the display, then use the inspector to enter the following details:
Identity - name: “txt_button_sideways” –
Text name: “sideways”
Activated: “yes”.

The second button, “rotation” turns the ball robot. ID-Name “txt_button_rotation” - Text name “rotation“ - Activated “yes”.

The third button “FIRE” fires the ball. ID-Name “txt_button_fire” - Text name “fire” - Activated “yes”.

“Fire ball” is a check box, and ticking the box releases the shot. ID-Name “txt_checkbox” - Text name “fire ball” - Clicked “no”.

Set the width at 100 and height at 40 for all variables in the “Inspector”. The buttons are now defined and you can start the main program.

First, start with the function to travel sideways.

Drag the command “Define do something” into the program screen from “Functions”. Replace “do something” with “move_sideways”. Then add the command “repeat x times” from the “Loops” area. Change the rat of repetition to “3”.

The commands to start the 4 encoder motors are integrated into the repetition loop. All motors turn “synchronously” at the full speed of “512” and an increment of “200” until the pulse counter on motor M1 has reached the value. Then the motors will stop.

Before you complete the “move_sideways” function, you need to create a query for the control box on the display. To do so, drag the command “if control box ...” from the “Displays” group into the program screen. Since the check box has already been defined with a name, it is shown automatically.

Next, create a variable “ball_fire”. To do so, select “Create variable” from the “Variables” group.

Insert the command “set – to” from the “Variables” block. Click the arrow and select the entry “ball_fire” from the selection window.

The command is completed by the command “Event checked” from the “Displays” block.

Now you can continue working on the “move_sideways” function. Add the command “+ if – then” from the “Logic” group and the command “ball_fire” from the “Variables” group.

In the “then” area. Insert the command “fire” from the “Functions” group.

The “move_sideways” function is now complete, and you can create the “move_rotate” function.

To do so, you simply need to copy the “move_sideways” function and change some parameters. Right click on the command block to be copied. A context window will appear with the “Copy” command.

Activate it and insert the copied function on the screen. Change the designation “move_sideways” to “move_rotation” (click the name and overwrite the text). Change the values for synchronous running of the motors and the direction of rotation. Change the increment to “100”, the direction of rotation for M3 to right and M4 to left. All other values will remain unchanged.

Before you define the last function, you must first program the program start. To do so, delete the “Loops” area.

Create four more variables “pos_f”, “i”, “pos_load” and “pos_f”.

Add the command “set x to” 4 times from the “Variables” group and change the name by clicking the arrow.

Now you need to fill the empty docking points. First, set the variable “pos_load” to the value “110”. This is the angular value at which the servo will turn. Use the command “one number” from the “Maths” group to do so. Change the value to “110”.

Then add the command “is the total of two numbers” twice from the same group. Fill the first empty space with the variables “pos_load”, the second with the command “one number”, then set its value to “120” or “250”.

In the last variable block, insert the command “true” from the “Logic” group and change it to “false”.

Then add the command “repeat continuously” from the “Loops” group under these blocks. Then the program will run in an endless loop.

You need the next two parts of the program to query the buttons “txt_button_sideways” and “txt_button_rotation. The two commands are available in the “Display” group. Drag the command “if button clicked: event” to the screen twice and select the relevant entry (click the arrow) in the context window.

Add the relevant functions to be executed when the button is pressed from the “Functions” group.
Now the last function - “define load” is still required, which will load the shooting mechanism from the magazine from the balls. Simply use the function shown here.

This function is required to query the button “txt_button_fire”. Once you have created the “fire” function, you need to integrate it into the query for the button. When the button is pressed, 3 shots should be fired. The program shown here shows the query.

Now all parts of the program are complete. Load the balls into the ball holder and start the program. If everything works correctly, save it on your computer.

Task 3

The next program you will create is used for the following task: The goal is for the robot to find red, yellow or green panels. Which colour it should find is defined on the TXT 4.0 display. Once the robot has found the colour, it aligns itself to the goal.

You don't have to write this program yourself. Load the program **"omniwheels_x4_targetdetect_easy"**. It is found under

"Menu" "Examples"

Beispiel

"New"

Click the folder and confirm by pressing "NEXT".

In the next step, select the required building set "Hightech". Browse by clicking the right arrow.

Then confirm by pressing "NEXT".

If you press the selection arrow again, all available model programs are displayed to you in a row. Confirm with "NEXT" until you come to the required program.

Bypass the following information by clicking "CREATE".

ROBO Pro Coding will open the selected program and display it to you on the screen.

There will certainly be some parts of the program you are not familiar with yet. We will explain these to you.
Let's look at the camera function here.

To do so, open "Camera function" in the project window.

The photo function is set over the entire camera field, so that the camera's entire area is used to take photos.

Look at the Controller configuration as well. The camera is added to USB1_1 here.

Next, open the “lib_move_sync” window. Here, you will find all the subprograms you need to move the vehicle. The program for travelling forwards is shown here.

The main program is used to define the variables and start the ball robot, as well as to continuously repeat the travel and shooting function.

For instance, the number of balls is defined by setting the variable “number_balls” to 3.

Look at another subprogram.

If the coloured area “yellow” is detected, the position should be determined. The blue block is used to show certain information to you on the console.

Before looking at the model with voice input, start the program

“omniwheels_x4_target_balldetect_advanced”

If you look at the program screen, you will see that the “Display configuration” is activated in the project window. If you click the entry with your mouse, the following window will open.

Use the “TXTSwitch” switches to select the individual colours that a shot should be fired at on the TXT 4.0 Controller display. The selected colour is shown visually on the display.

Task 4

The main program is essentially the same as the previous program, although it contains the query for the display switch.

The button “Start” is queried in the first block.

In the second block (also applies to all of the other buttons), the switching status is queried and displayed visually.

The information on the settings is then adopted into the analysis program. Simply test out your model and think about what is being executed in the program. Set up the 3 targets beside one another; the robot will search for the coloured areas, move to the right position and then shoot at the targets with the selected colours.

Note: If the camera does not detect the coloured areas, this may be due to the lighting conditions in the room. In this case, open the camera configuration in the ROBO Pro Coding program, select the image detection element used there, and adjust the colour value that is not detected correctly in the inspector.

Task 5

To complete our work with the “Ball robot” model, we will use the voice control.

The program is available under

“omniwheels_x4_ball_voice_control”

New project – Example – ROBOTICS Hightech – **omniwheels_x4_ball_voice_control**

To use the voice control, you will first need to load an app to your smartphone (Android or iOS). You can download the app from an app store at

fischertechnik Voice Control

and install it.

Important: The TXT 4.0 Controller must be switched on and connected to the same WiFi router as the smartphone/tablet on which the Voice Control app is used. Set up a WiFi connection on the TXT 4.0 Controller: see Basics on the home page or the operating manual for the TXT 4.0 Controller.

The ROBO Pro Coding program must be launched on the TXT Controller.
Then start or open the app.

The first time you open it, change the working language. To do so, activate the “Settings” button.

Select your language here.
Now, the voice control app needs to be connected to the Controller. To do so, select the “Connect” button.

You need to enter the IP address of the Controller and its API key in the connection dialogue.

The IP address is found on the Controller display under

“Info” - “WiFi” - IP and the address, such as 192.168.0.xx

The API key is found under “Settings” - API key, such as h3TZTf.

Once both values have been entered, you can activate “Connect”.

The voice input window will appear. After you activate the microphone symbol, you can give a command. The detected text is shown on the display. Then you can either wait until the command is automatically transmitted, or press the microphone symbol again to send it immediately.

The voice input is sent to the Controller and processed there by the ROBO Pro Coding program.

The terms received by the Voice Control app that should trigger a certain action are defined in the ROBO Pro Coding program.

Now, test out your model using voice control.

Information The following voice commands are used in the ball robot program to trigger specific actions:

Action	Voice command
Load ball from magazine	Load
Shoot ball	Ball, shoot ball, fire, shoot
Travel commands	Forward, back, left, right, turn Small turn left Small turn right
Greeting signal output	Hello

We hope you enjoy controlling the ball robot with voice control!