

Challenge 06: IDE App





rexroth

ctrlX CORE

F01

Let's Start!

10

Let's Start!

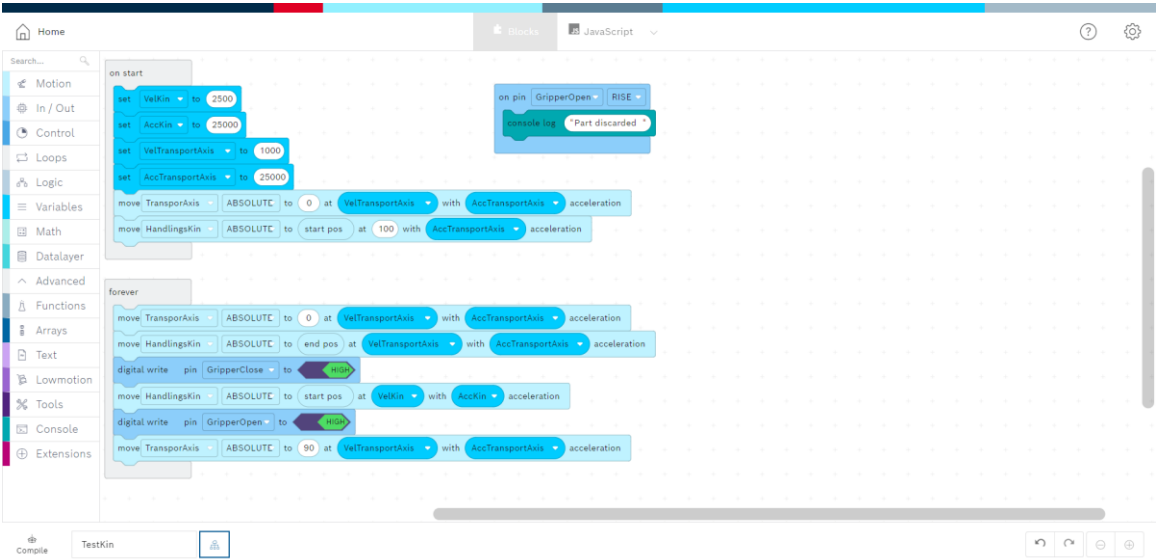
Inform

General information

Integrated Development Environment is referred to as **IDE**. It is a piece of software that gives programmers all the tools and functionality they need to write, edit, test, and debug code. IDEs provide a cohesive and unified environment that is intended to expedite the development process.

ctrlX AUTOMATION – IDE App (Visual Coding)

By using **Visual Coding**, the programmer does not have to deal with the specifics of the programming language, such as syntax, but intuitively uses predefined program elements, e.g. for variable handling, program loops, control instructions or motion commands. By assembling the building blocks via drag & drop and their parameterization, the program flow is successively created. The actual program code in Python or JavaScript is created automatically in the background.

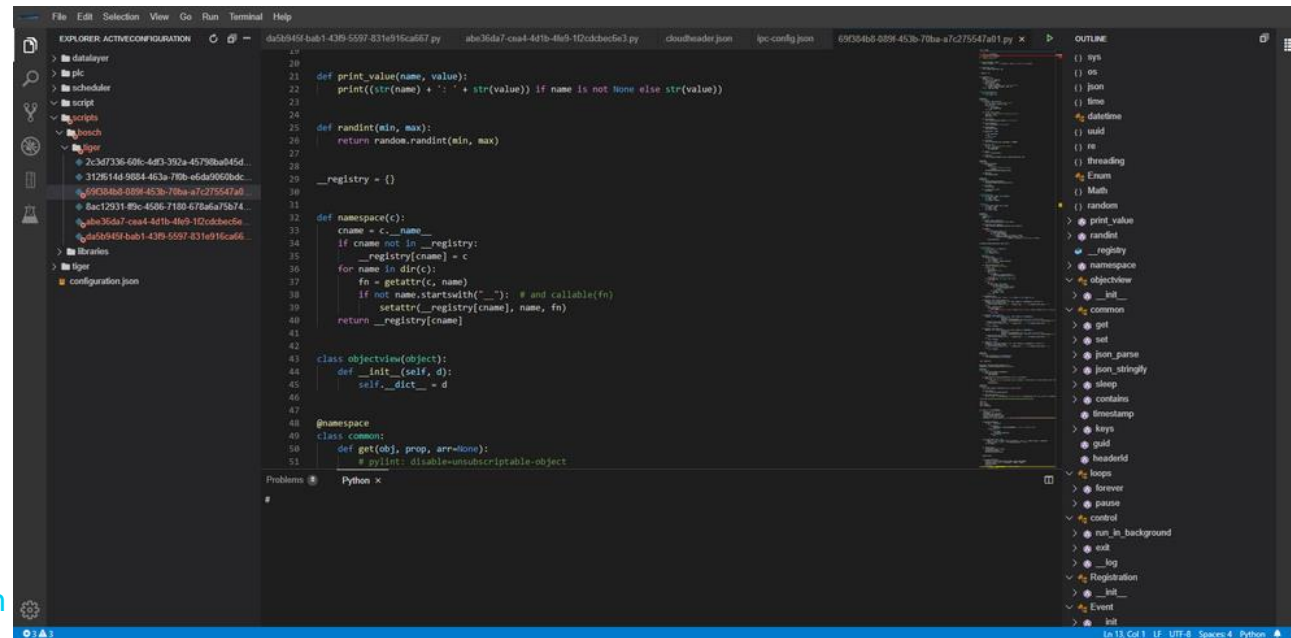


Let's Start!

Introduction to

ctrlX AUTOMATION – IDE App (Textual Coding)

Textual Coding represents a web-based development environment (code editor, console, debugging, etc.) with native ctrlX CORE connectivity. It can be used to create Python scripts or edit files in the active solution.



Information about the ctrlX IDE App can

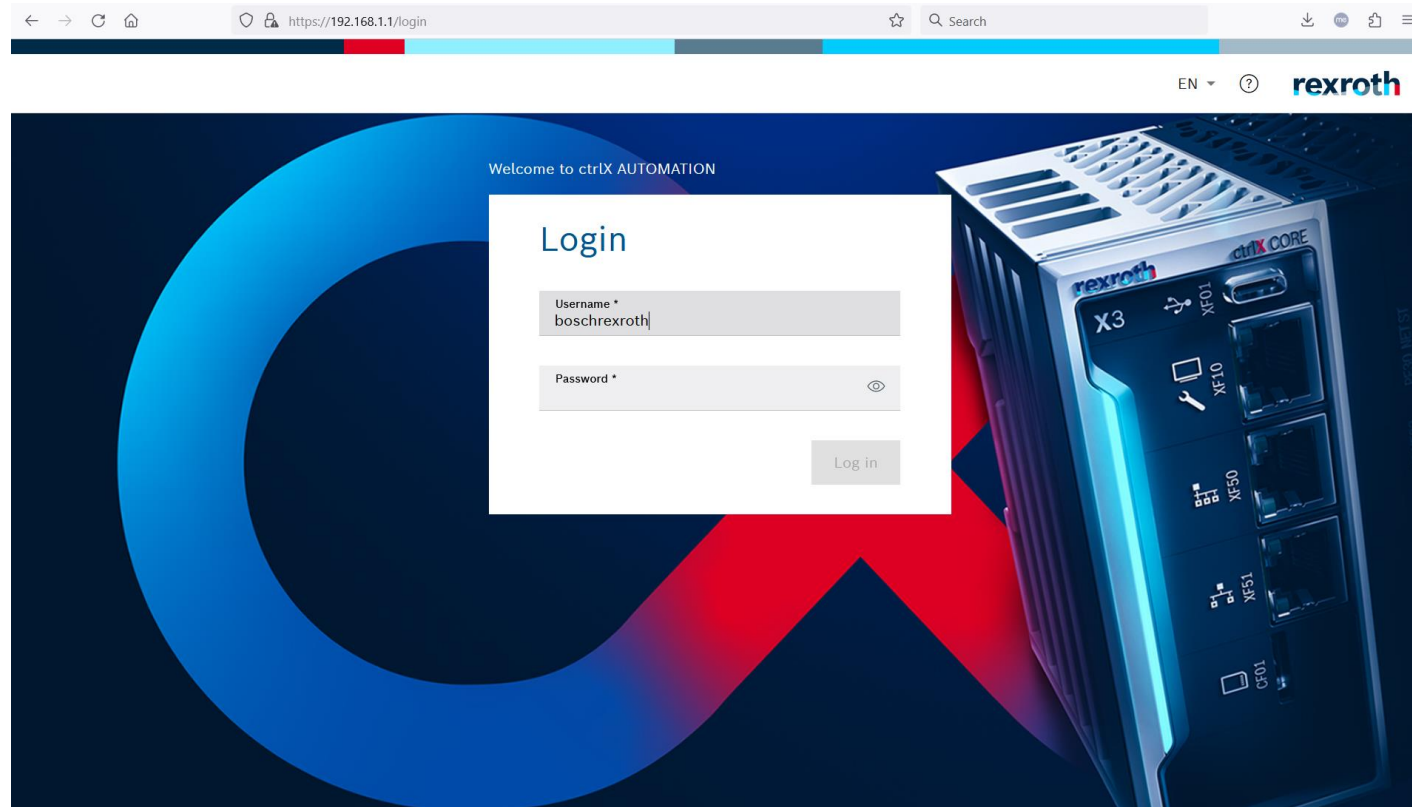
- [ctrlX IDE App | ctrlX AUTOMATION Community](#)
- [ctrlX IDE App | Application Manual](#)

Challenge 06: ctrlX IDE App | Setting up Visual Coding

Steps

1. Login into the ctrlX web-based user interface. Enter the Login details (Username: boschrexroth, Password: B0schrexroth).

ctrlX Web-Based User Interface – Login Page



Challenge 06: ctrlX IDE App | Setting up Visual Coding

Steps

2. Open your ctrlX CORE Web Interface, click on IDE, then click on 'Visual Coding'.

ctrlX Web-Based User Interface – Home Page

1. Click on the IDE App

ctrlX-CORE

ctrlX CORE

Home

Service Indicator

DeviceBridge

Diagnostics

EtherCAT Master

IoT Dashboard

IDE

InfluxDB

Modbus TCP

Motion

Node-RED

OPC UA

PLC

PROFINET device

Remote Agent

Home

Operating

Show more about ctrlX CORE

EtherCAT

Configure and analyze your EtherCAT network

Name: ethercatmaster
State: OP
Message: Running

Status I/O Engineering Configuration

Axes

Axes not configured

Configure axes

Axis profiles

Axis profiles not configured

Configure axis profiles

Kinematics

Kinematics not configured

Configure kinematics

OPC UA Client

Manage OPC UA Client

Client connections: 0

Configuration/ Status

OPC UA Server

Manage OPC UA Server

Status: RUNNING
Start time: 10/27/2023, 10:02:46 AM
Sessions 0
connected:

Configuration/ Status
Certificate Configuration

2. Click on Visual Coding

IDE

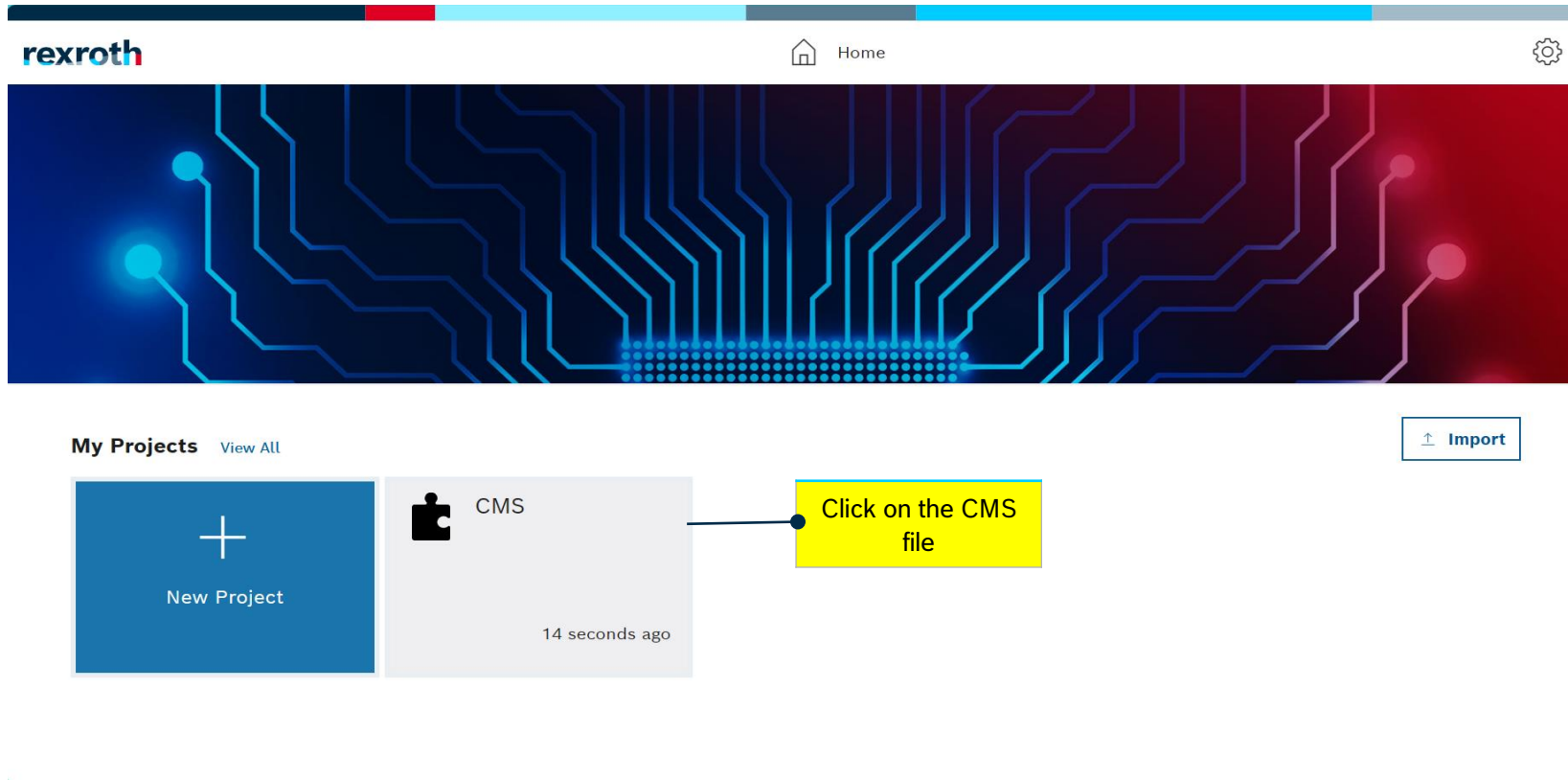
Textual Coding

Visual Coding

Challenge 06: ctrlX IDE App | Setting up Visual Coding Steps

3. On the same tab, the Visual Coding home page will pop up. Then, please click on CMS (Cartesian Multi-axis Systems) file.

ctrlX IDE App – Visual Coding Home Page



Challenge 06: ctrlX IDE App | Setting up Visual Coding

Steps

4. Here is the Visual Coding Interface, the space where you will do blockly programming.

ctrlX IDE App – Visual Coding Interface

The screenshot displays the ctrlX IDE App's Visual Coding Interface. It features a top header bar with tabs for 'Home', 'Blocks', and 'JavaScript'. A left sidebar contains a search bar and a categorized toolbox with groups like Motion, In / Out, Control, Loops, Logic, Variables, Math, Datalayer, and Advanced. The main workspace is a large grid for block programming, currently showing an 'on start' block. A bottom footer bar includes a 'Compile' button, a 'Test' button, and a 'Start' button. Four numbered callouts provide details: 1. Header (Editor Switching), 2. Footer (Compile and Start buttons), 3. Side Navigation (Toolbox groups), and 4. Code Editor (Block dropping area).

1. Header

- Editor Switching for changing the block programming <-> python <-> Javascript

2. Footer

- Compile Button to compile the project
- Start Button to execute the created project

3. Side Navigation

- The toolbox at the side navigation is divided into several group
- Each group contains several blocks for graphical programming

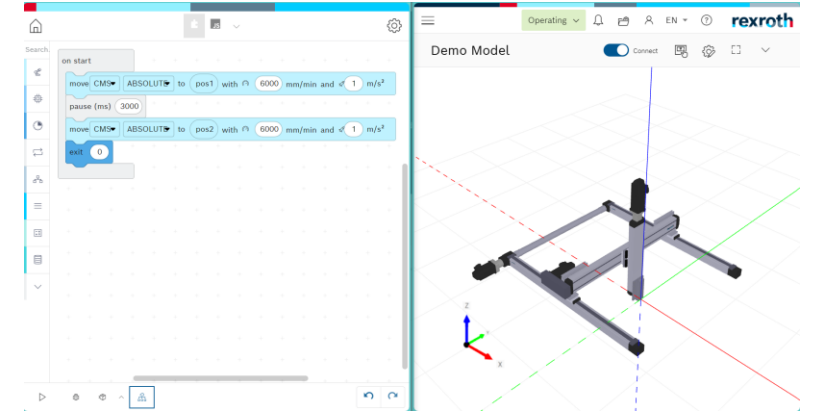
4. Code Editor

- The space where you drop your blocks from side navigation and do visual coding

Challenge 06: IDE App

Task 01

Challenge 06: ctrlX IDE App | Task 01



Description

After you have configured the CMS by adding some axes in Motion App Challenge, you want the flexibility of controlling the CMS by using a “Low-Code” language such as the “Block” programming. You still have not configured your Axis on Motion group inside the Visual Coding. Your task is to make the program running on position 1 and position 2 without stop.

Position 1:

Axis_X = 400mm, Axis_Y = -400mm, Axis_Z = 0mm

Position 2:

Axis_X = 300mm, Axis_Y = 200mm, Axis_Z = 50mm

Task

This task will test your understanding on controlling your cartesian robot which is CMS using Visual Coding

Challenge 06: ctrlX IDE App | Task 01

Safety instructions for the project exercise

In order to ensure the operational capability and to identify the possible hazards of machines and systems, the safety regulations must be observed before and during the order execution.

The ctrlX CORE may only be operated in technically perfect condition. The intended use, performance data and operating conditions may not be changed. No protective devices/components may be deactivated.



In case of emergency, failure or other irregularities:

- Before connecting or disconnecting any electrical components, ensure that the power to the ctrlX CORE unit and associated equipment is turned off.

Challenge 06: ctrlX IDE App | Task 01

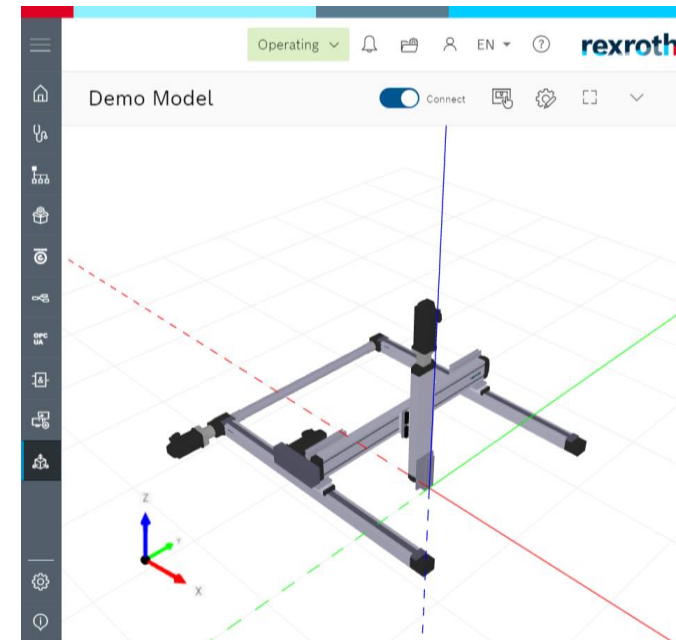
Steps

5. After you click on the CMS file, it already include with a template programming without any configuration yet.

Solve the issues in a Block Programming

- The block programming has errors when the red line outline show on the block.
- You need to move the CMS (which is the Demo Model) after configured the motion group. Now, please click on Motion.

Click on Motion



Challenge 06: ctrlX IDE App | Task 01

Steps

6. Here you can see the blocks inside the Motion group, now please click on Configure.

The screenshot displays the ctrlX IDE interface. On the left, a sidebar contains a search bar and a list of categories: Motion, In / Out, Control, Loops, Logic, Variables, Math, Datalayer, and Advanced. The 'Motion' category is selected, and a yellow callout box with the text 'Click on Configure' points to the 'Configure' button within the Motion block library. The main workspace shows a 'Motion' block with various configuration options, including 'ABSOLUTE' mode, 'MAIN_AXIS_X' position, and '0' values. A 'Configure' button is visible on the right side of the block. At the bottom of the interface, there are buttons for 'Compile', 'CMS', and a 'Run' button.

Challenge 06: ctrlX IDE App | Task 01

Steps

7. From this page, you can search your previous Motion axes configuration.

Home

Blocks

JavaScript

?

⚙

Go back

Axes

To manage project specific configurations, an individual axes setup can be created. The axes can be included by creating new ones (plus-button) or via look up (magnifier-button). Save your configuration for activation.

0 items

↺

🔍

+

Id	Address	Type	Limits	Axis Params	Parent	Factory	Actions
----	---------	------	--------	-------------	--------	---------	---------

Kins

To manage project specific configurations, an individual kinematic setup can be created. The kinematic can be included by creating new ones (plus-button) or via look up (magnifier-button). Save your configuration for activation.

0 items

↺

🔍

+

Id	Address	Axes	Limits	Units	Actions
----	---------	------	--------	-------	---------

Points

Points are named position values. You can either define a single value or multiple values for one name.

Click on Search Icon

Challenge 06: ctrlX IDE App | Task 01

Steps

8. Click on the axes and import all axes which are Axis_X, Axis_Y and Axis_Z.

Import from target

Axes
None

Import ✓

Cancel ✕

Click on Axes

Import from target

Axes
Multiple values selected

☒ Axis_X

☒ Axis_Y

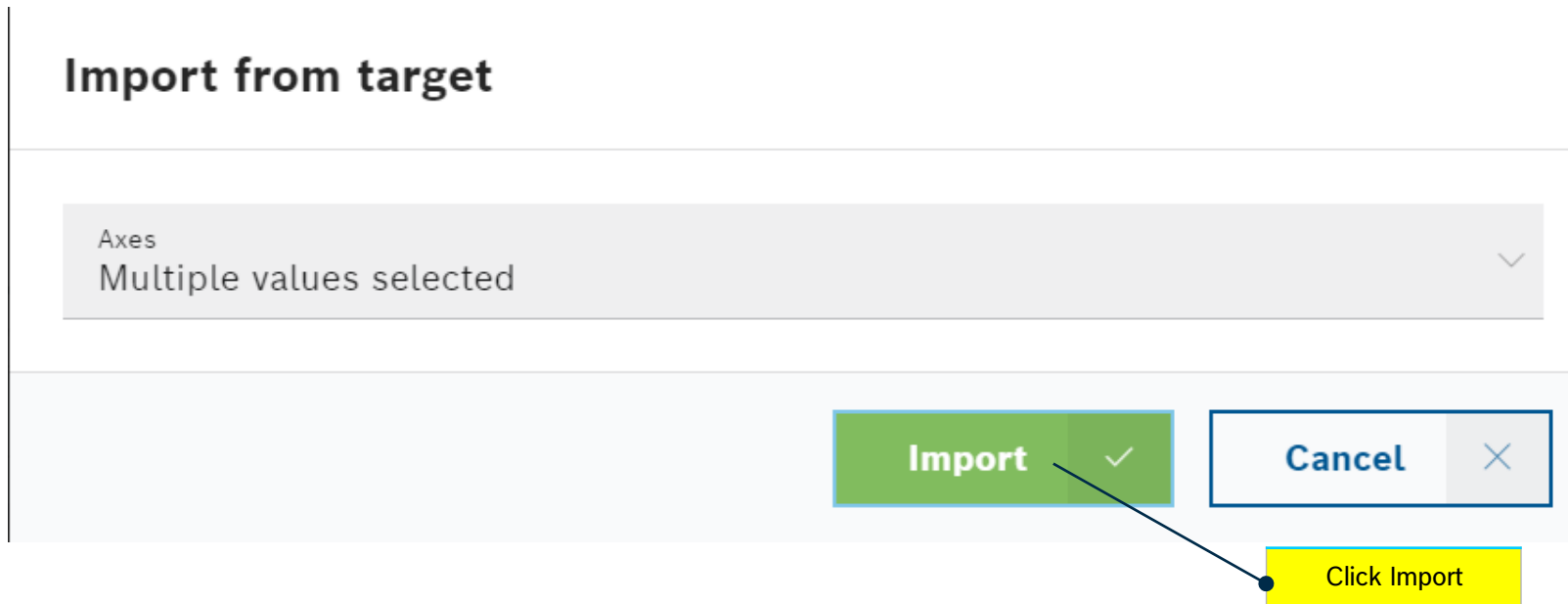
☒ Axis_Z

Choose all axes that have been configured in Motion Challenge

Challenge 06: ctrlX IDE App | Task 01

Steps







9. Then, please click Import.



Challenge 06: ctrlX IDE App | Task 01

Steps




10. You can see your previous axes that have been imported, now let's search for our CMS which we can do it in Kinematics (Kins). Click on Search icon.

Id	Address	Type	Limits	Axis Params	Parent	Factory	Actions
Axi s_X	Axis_X	0	{ "position": { "min": -1000, "max": 1000, "unit": "mm"}, "velocity": { "min": -6000, "max": 6000, "unit": "mm/min"}, "acceleration": { "min": -100, "max": 100, "unit": "m/s^2"}, "torque": { "min": 0, "max": 100, "unit": "N"} }	{ "x": 0, "y": 0, "z": 0, "x r": 0, "y": 0, "z": 0 }	undefin ed	undefine d	 
Axi s_Y	Axis_Y	0	{ "position": { "min": -1000, "max": 1000, "unit": "mm"}, "velocity": { "min": -6000, "max": 6000, "unit": "mm/min"}, "acceleration": { "min": -100, "max": 100, "unit": "m/s^2"}, "torque": { "min": 0, "max": 100, "unit": "N"} }	{ "x": 0, "y": 0, "z": 0, "x r": 0, "y": 0, "z": 0 }	undefin ed	undefine d	 
Axi s_Z	Axis_Z	0	{ "position": { "min": -1000, "max": 1000, "unit": "mm"}, "velocity": { "min": -6000, "max": 6000, "unit": "mm/min"}, "acceleration": { "min": -100, "max": 100, "unit": "m/s^2"}, "torque": { "min": 0, "max": 100, "unit": "N"} }	{ "x": 0, "y": 0, "z": 0, "x r": 0, "y": 0, "z": 0 }	undefin ed	undefine d	 

Kins

To manage project specific configurations, an individual kinematic setup can be created. The kinematic can be included by creating new ones (plus-button) or via look up (magnifier-button). Save your configuration for activation.

0 items


  

Id	Address	Axes	Limits	Units	Actions
----	---------	------	--------	-------	---------

Points

Points are named position values. You can either define a single value or multiple values for one name.

0 items



Id	Reference	Positions	Actions
----	-----------	-----------	---------

Click on Search Icon

Challenge 06: ctrlX IDE App | Task 01

Steps

11. Choose the CMS and click Import.

The image displays two sequential screenshots of the 'Import from target' dialog in the ctrlX IDE App.

Left Screenshot: The dialog is titled 'Import from target'. It features a dropdown menu labeled 'Kins CMS'. The dropdown is expanded, showing a list with a checked checkbox and the text 'CMS'. A yellow callout box labeled 'Choose CMS' points to this checkbox. At the bottom of the dialog are two buttons: 'Import' (green) and 'Cancel' (blue).

Right Screenshot: The dialog is in the same state, but the 'Import' button is now highlighted. A yellow callout box labeled 'Click Import' points to this button.

Challenge 06: ctrlX IDE App | Task 01

Steps

12. From here, you can see your previous Kins which is CMS that has been imported. Now, lets create our Position 1. Please click on ‘+’.

Axi s_Z	Axis_Z	0	{ "position": { "min": -1000, "max": 1000, "unit": "mm" }, "velocity": { "min": -6000, "max": 6000, "unit": "mm/min" }, "acceleration": { "min": -100, "max": 100, "unit": "m/s^2" }, "torque": { "min": 0, "max": 100, "unit": "N" } }	{ "x": 0, "y": 0, "z": 0, "x r": 0, "y r": 0, "z r": 0 }	undefin ed	undefine d		
------------	--------	---	---	---	---------------	---------------	--	--

Kins

To manage project specific configurations, an individual kinematic setup can be created. The kinematic can be included by creating new ones (plus-button) or via look up (magnifier-button). Save your configuration for activation.

1 items

Id	Address	Axes	Limits	Units	Actions	
CM S	CMS	[{"axis": "Axis_X", "direction": 0, "meaning": 0}, {"axis": "Axis_Y", "direction": 0, "meaning": 1}, {"axis": "Axis_Z", "direction": 0, "meaning": 2}]	{ "velocity": { "min": 0, "max": 6000, "unit": "mm/min" }, "acceleration": { "min": 0, "max": 100, "unit": "m/s^2" } }	{ "position": "mm", "t orque": "Nm" }		

Points

Points are named position values. You can either define a single value or multiple values for one name.

0 items

Id	Reference	Positions	Actions
----	-----------	-----------	---------

Save

Edit Settings As text

Click on +

Challenge 06: ctrlX IDE App | Task 01

Steps

13. By following the instruction vertically, please type 'Pos1' as your id, 'CMS' as your reference, set MAIN_AXIS_X to -400mm, MAIN_AXIS_Y to -400mm and MAIN_AXIS_Z to 0mm. Then, click 'Apply'.

Id

Pos1

Reference

CMS

Positions

Define a single position in the Point array.

3 items

+

Value	Meaning	Actions
-400	0	<div><div></div><div></div></div>
-400	1	<div><div></div><div></div></div>
0	2	<div><div></div><div></div></div>

Apply

Cancel

For id, create the first position as Pos1

Choose CMS for your Reference

Set your MAIN_AXIS_X to -400mm

Set your MAIN_AXIS_Y to -400mm

Set your MAIN_AXIS_Z to 0mm

20

ctrlX AUTOMATION Two Steps Ahead | February 2021
© Bosch Rexroth AG 2021. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.

rexroth
A Bosch Company

Challenge 06: ctrlX IDE App | Task 01

Steps

14. After finish add axes, CMS as our Kinematics and created Position 1, please click on save.

Axi s_Z	Axis_Z	0	{ "position": { "min": -1000, "max": 1000, "unit": "mm" }, "velocity": { "min": -6000, "max": 6000, "unit": "mm/min" }, "acceleration": { "min": -100, "max": 100, "unit": "m/s^2" }, "torque": { "min": 0, "max": 100, "unit": "Nm" } }	{ "x": 0, "y": 0, "z": 0, "x r": 0, "y": 0, "z": 0 }	undefi ned	undefi ned		
------------	--------	---	--	---	---------------	---------------	--	--

Kins

To manage project specific configurations, an individual kinematic setup can be created. The kinematic can be included by creating new ones (plus-button) or via look up (magnifier-button). Save your configuration for activation.

1 items

Id	Address	Axes	Limits	Units	Actions	
CM S	CMS	{ "axis": "Axis_X", "direction": 0, "meaning": 0 }, { "axis": "Axis_Y", "direction": 0, "meaning": 1 }, { "axis": "Axis_Z", "direction": 0, "meaning": 2 }	{ "velocity": { "min": 0, "max": 6000, "unit": "mm/min" }, "acceleration": { "min": 0, "max": 100, "unit": "m/s^2" } }	{ "position": "mm", "torque": "Nm" }		

Points

Points are named position values. You can either define a single value or multiple values for one name.

1 items

Id	Reference	Positions	Actions	
Pos1	CMS	{ "meaning": 0, "value": -400 }, { "meaning": 1, "value": -400 }, { "meaning": 2, "value": 0 }		

Save

Edit Settings As text

Click on Save

Challenge 06: ctrlX IDE App | Task 01

Steps

15. Please click on  to run the initialization for motion.

Run initialization (motion)

Extension specific initialization routine should be executed after the configuration has been changed. Press 'Run' to start target initialization now...

Run



Click on Run

Challenge 06: ctrlX IDE App | Task 01

Steps

16. Please wait until the initialization finished, then click on Close.

Initialization (motion)

 **INITIALIZATION FINISHED**

- ✓ Step 4: initializing axes...
- ✓ Step 5: initializing kinematics...
- ✓ Step 6: booting hardware profiles...
- ✓ Step 7: switching to running mode...
- ✓ Step 8: persisting solution configuration...

Close

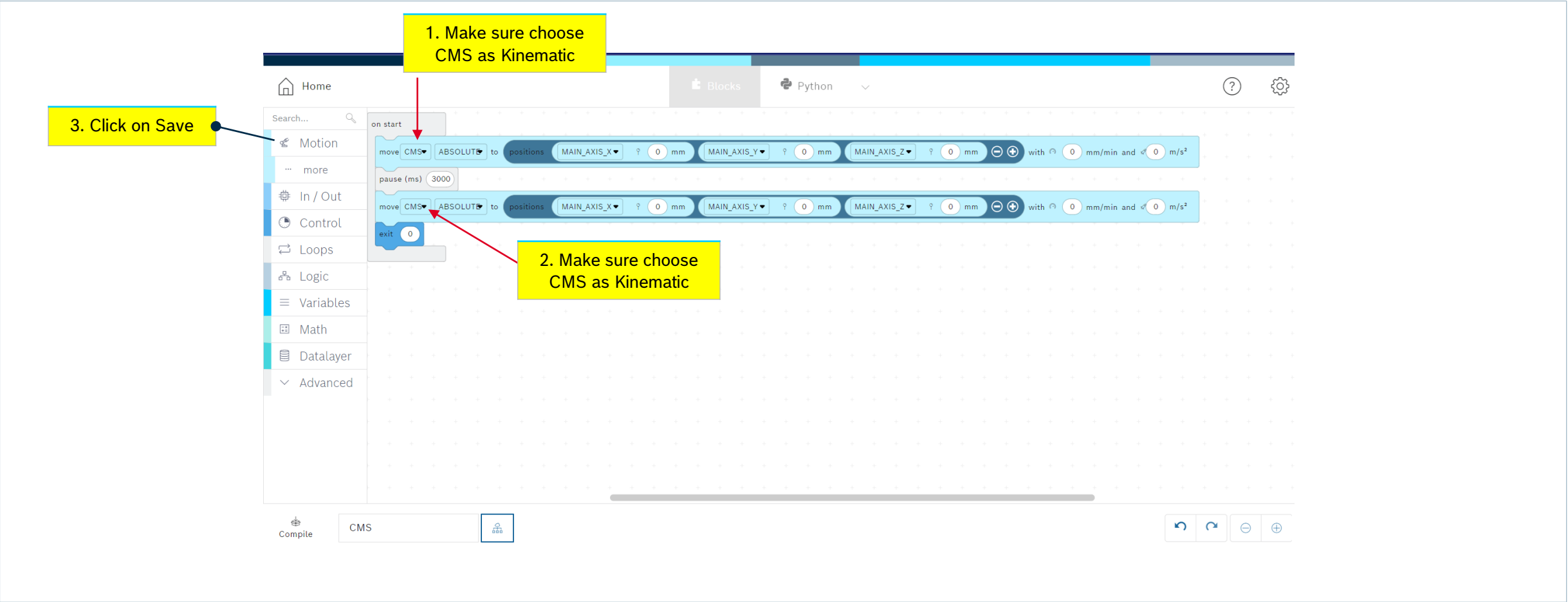


Click on Close

Challenge 06: ctrlX IDE App | Task 01

Steps

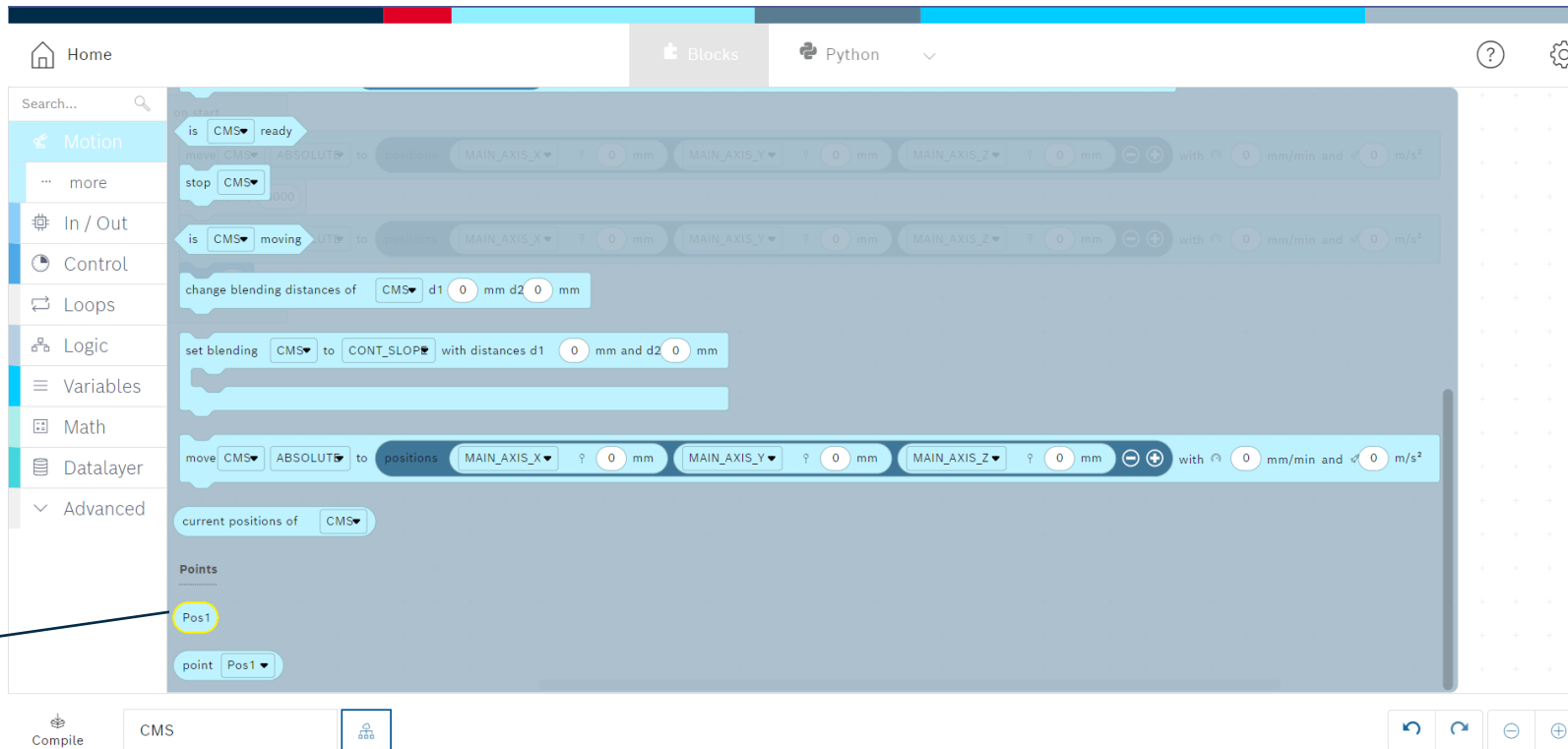
17. Now, you will be back to your Visual Coding Interface, please make sure you choose CMS as your Kinematic. Then click on Motion to add your Position 1.



Challenge 06: ctrlX IDE App | Task 01

Steps

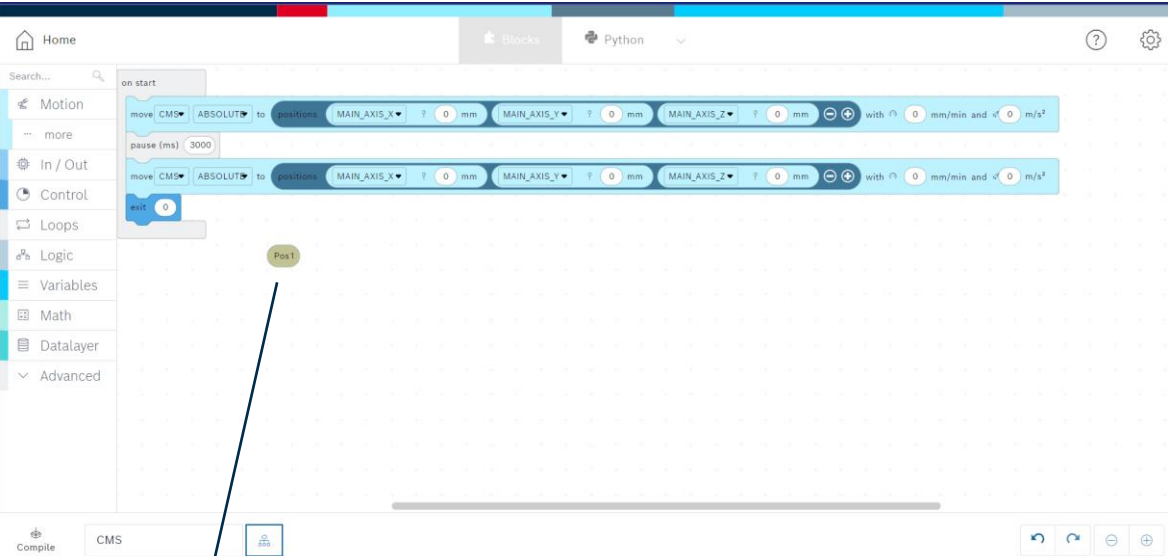
18. Scroll until you can find the Pos1, then please drag it to the Code Editor.



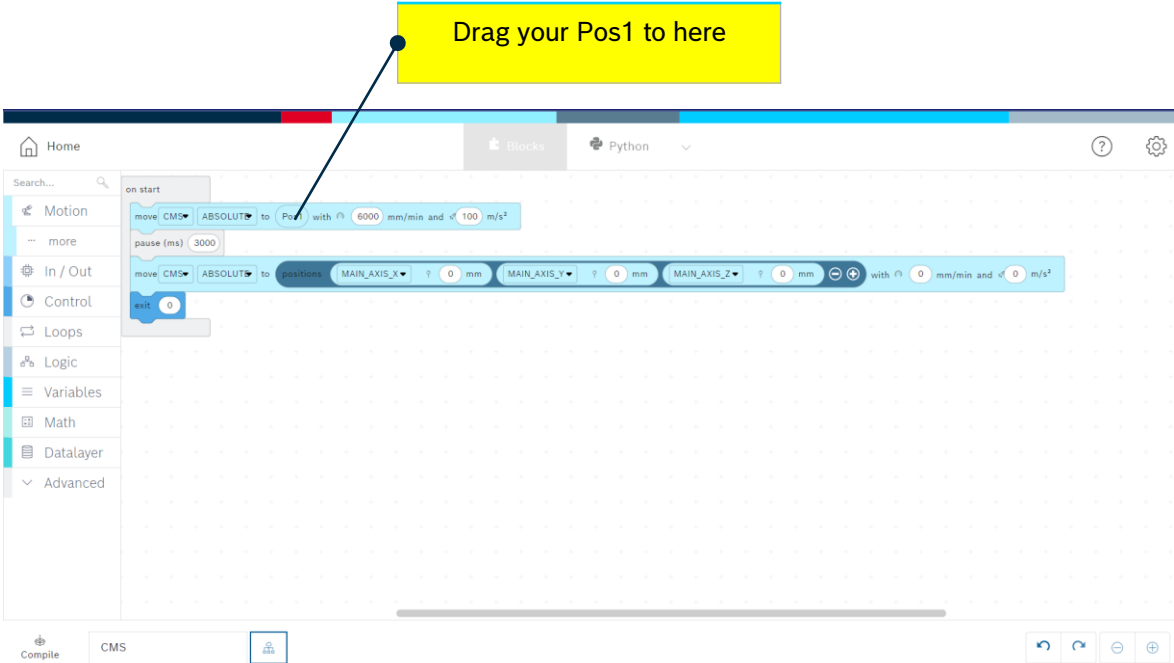
Challenge 06: ctrlX IDE App | Task 01

Steps

19. Please drag your Pos1 to the Move block.



Pos1



Drag your Pos1 to here

Challenge 06: ctrlX IDE App | Task 01

Steps

20. Please set the distance, velocity and acceleration of Pos1 and Pos2 correctly.

The screenshot displays the ctrlX IDE interface with a motion program. The program consists of the following blocks:

- on start**
- move** CMS ABSOLUTE to Pos1 with 6000 mm/min and 100 m/s²
- pause** (ms) 3000
- move** CMS ABSOLUTE to positions MAIN_AXIS_X 300 mm MAIN_AXIS_Y 200 mm MAIN_AXIS_Z 50 mm with 6000 mm/min and 100 m/s²
- exit** 0

Annotations with red arrows point to specific values in the program:

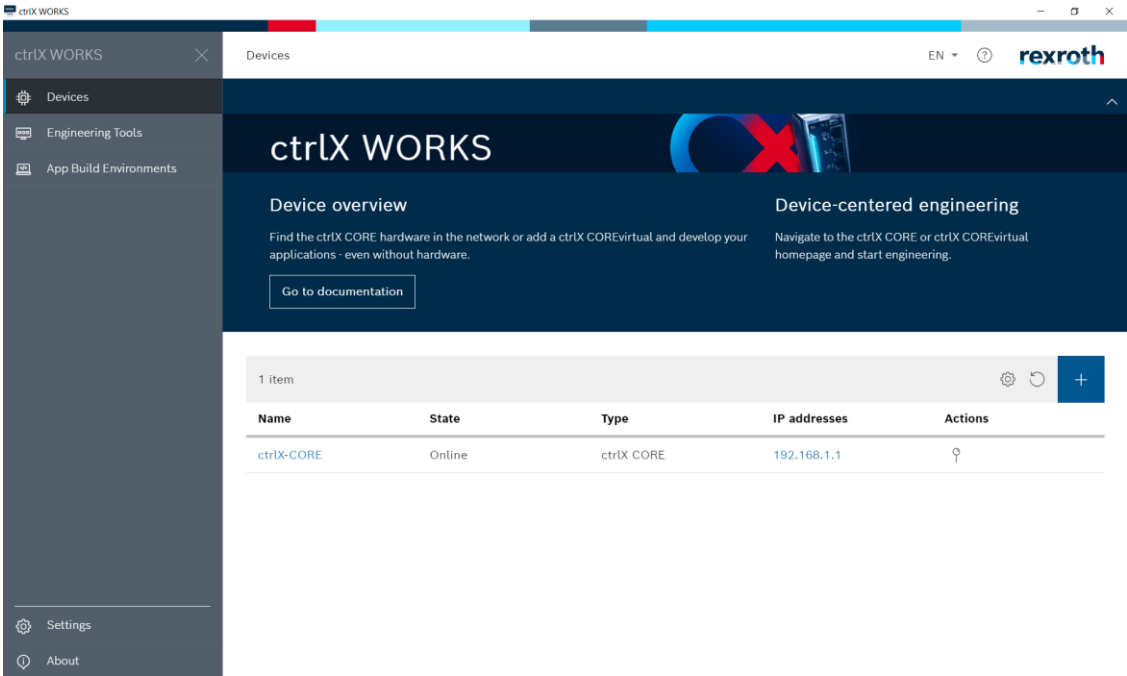
- Pos1, Velocity: 6000 mm/min** points to the velocity value in the first move block.
- Pos1, Acceleration: 100 m/s²** points to the acceleration value in the first move block.
- 300mm** points to the X-axis position value in the second move block.
- 200mm** points to the Y-axis position value in the second move block.
- 50mm** points to the Z-axis position value in the second move block.
- Pos2, Velocity: 6000 mm/min** points to the velocity value in the second move block.
- Pos2, Acceleration: 6000 m/s²** points to the acceleration value in the second move block.

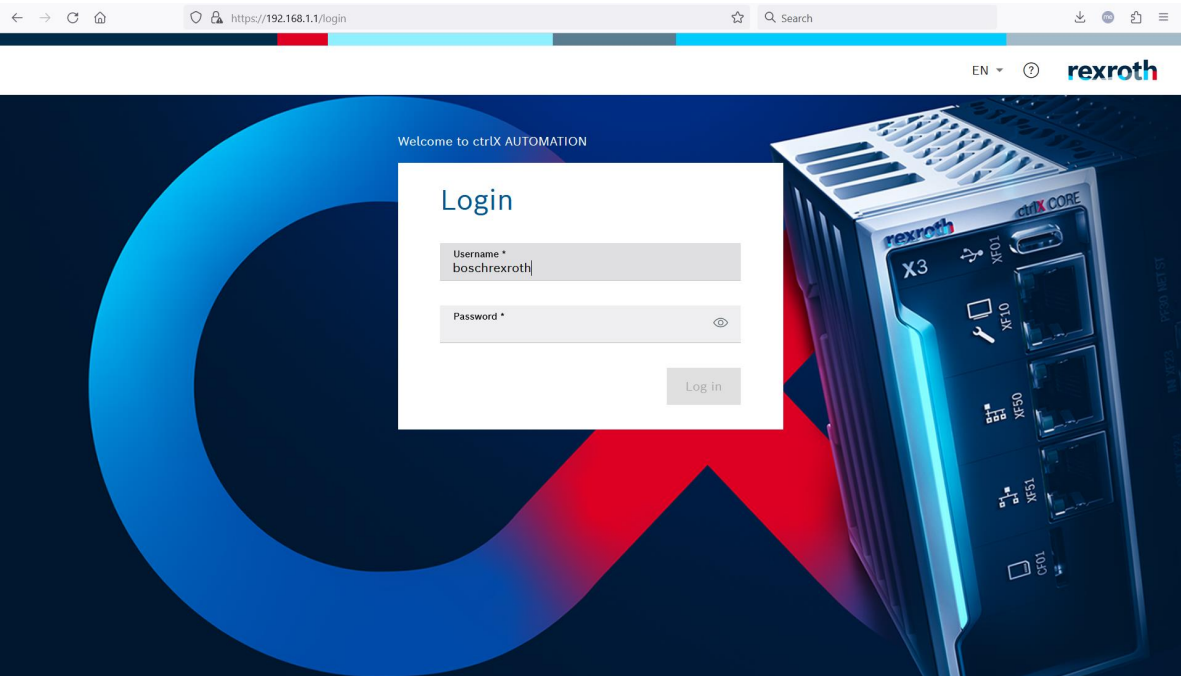
A yellow box at the bottom left contains the text "Click on Compile" with an arrow pointing to the "Compile" button in the bottom toolbar.

Challenge 06: ctrlX IDE App | Task 01

Steps

21. Now, we want to open 3D Viewer, please go back to ctrlX WORKS and log in again on a new tab.

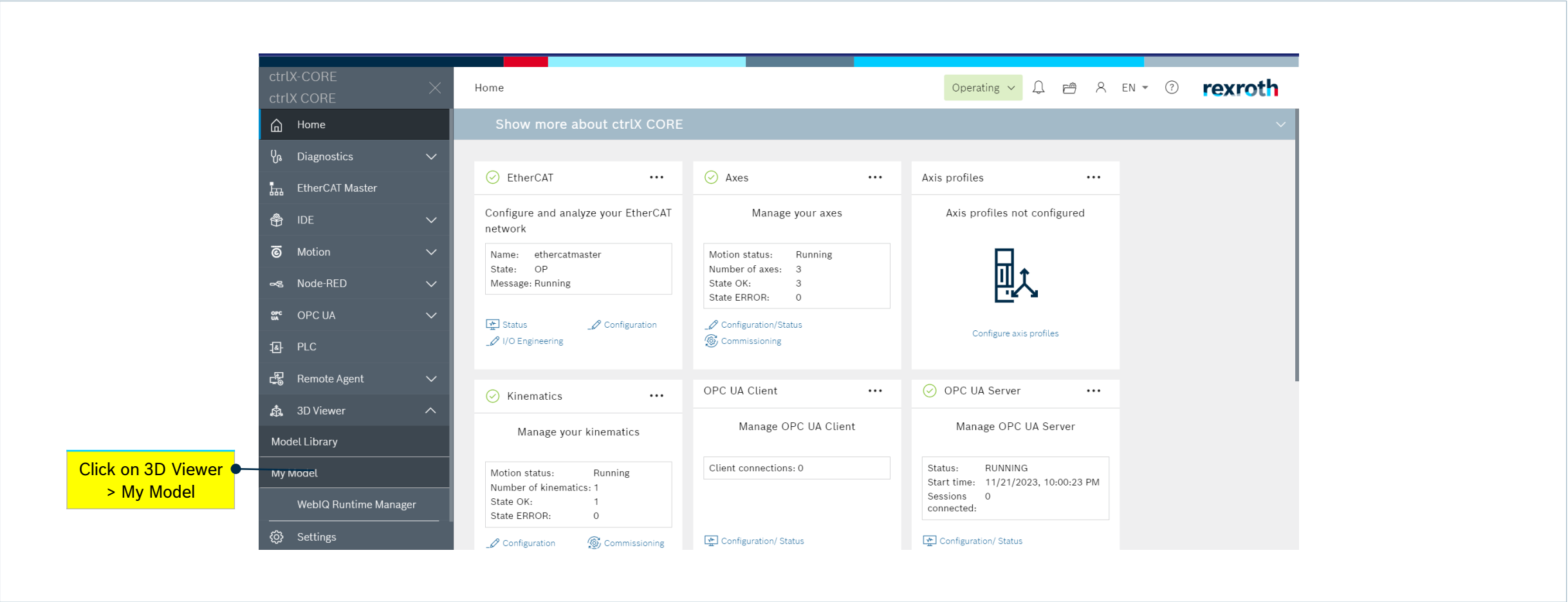




Challenge 06: ctrlX IDE App | Task 01

Steps

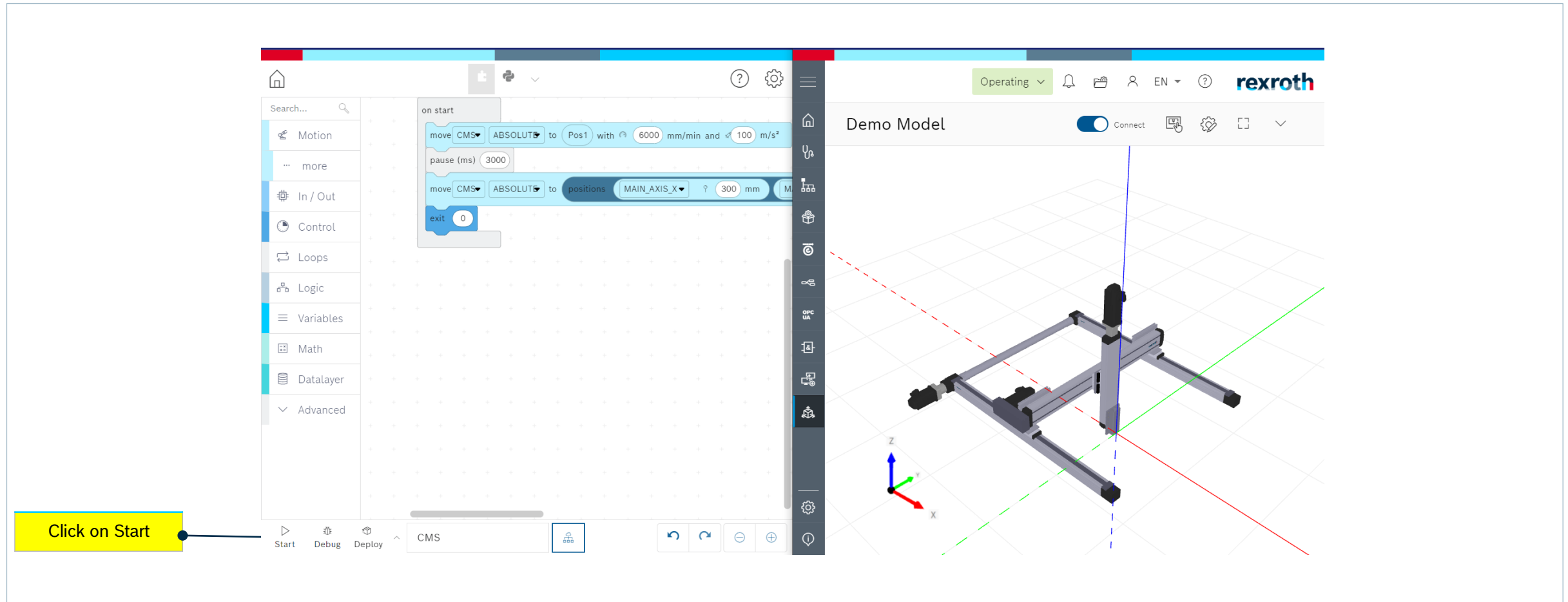
22. Click on 3D Viewer → My Model.



Challenge 06: ctrlX IDE App | Task 01

Steps

23. Please split the screen like this. On the left side is our Visual Coding and on the right side is our CMS from 3D Viewer. Then, click on Start Icon.



Challenge 06: ctrlX IDE App | Task 01

Steps

24. For Task 1, you need to solve the issues that exists in “**CMS**” in the IDE App – Visual Coding to make it work. Follow the steps below:

Do it yourself

- Configure a point for Position 2 as Pos2
- Please do a loop for the whole visual coding, hence it will not stop.

Challenge 06: ctrlX IDE App | Task 01

Steps

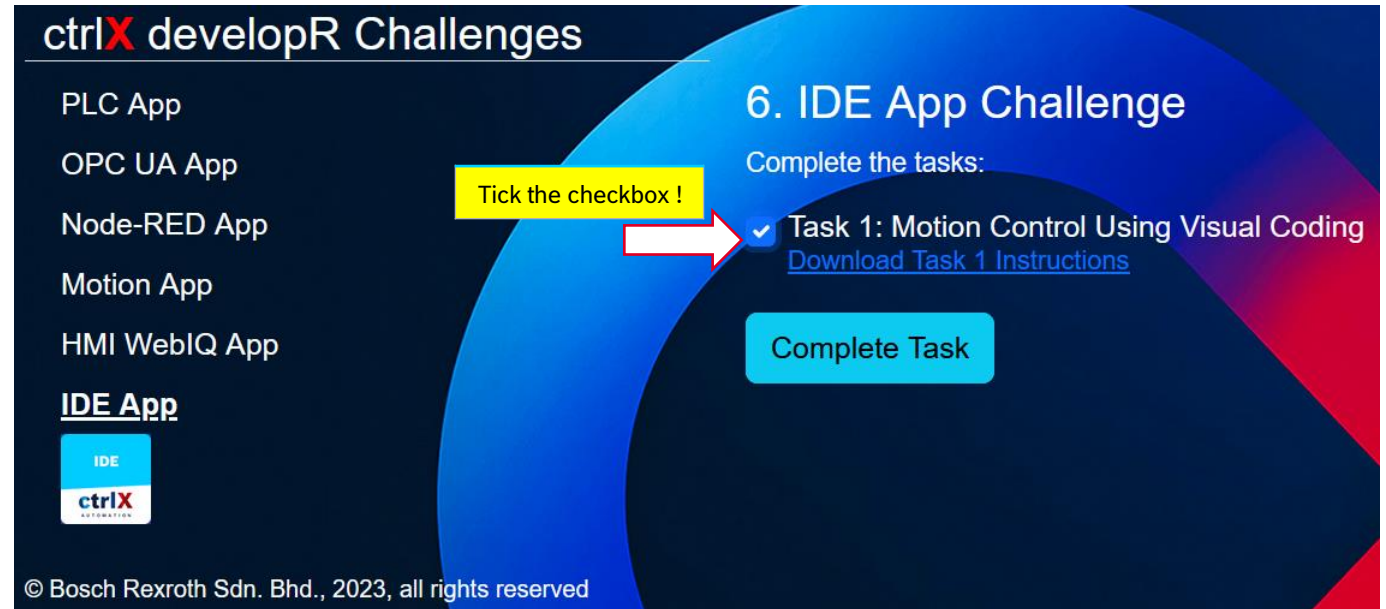
Once you have completed Task 1, follow the steps below.

How to complete Task 1 IDE App

- You can test your solution against the Task description
- Once it satisfies the requirements, confirm that you have completed the task by informing the available instructor for verification
- In the ctrlX developR challenge [website](#), under the IDE App challenge section, tick [✓] the Task 1 checkbox

Congratulations, you've completed the task!

Follow the next step to complete the challenge!



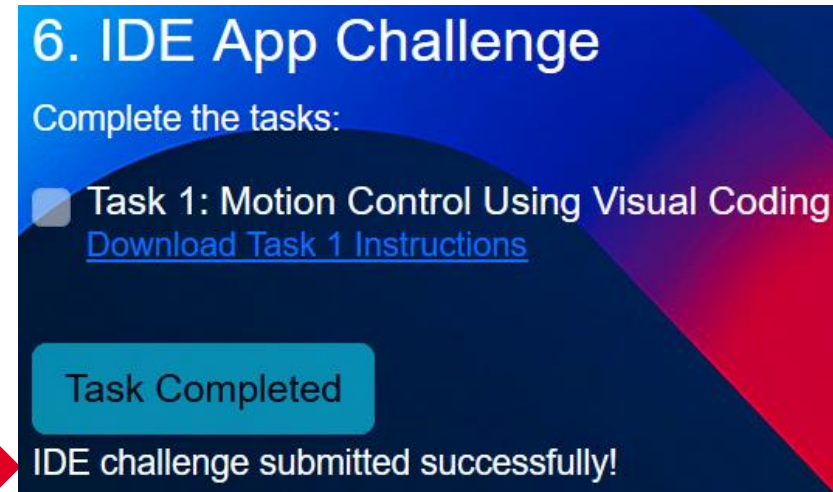
Challenge 06: ctrlX IDE App | Task 01

Steps

Once you have completed Task 1, follow the steps below.

How to complete the IDE App Challenge

- Finally, click on the “**Complete Task**” button



- Once pressed, the button text will change to “Task Completed” and you will be notified with a message that the challenge has been successfully submitted.
- By pressing the “Complete Task” button, the duration it takes for the team to complete the challenge will be automatically submitted.
- Submission only can be done once per challenge.

Congratulations, you’ve successfully completed the IDE App challenge! Wunderbar!

Challenge 06: IDE App

Congratulations!