Lesson 8 Programming Control Magician

Scene lead-in

In order to let others know about your own ideas, information exchange is required. Human beings communicate with each other using language; the bees tell the specific direction of the nectar to their fellows through dance; and many birds transmit information through their chirps. However, people need to give instructions to the computer via programming. The computer cannot understand human language: so how can we communicate with the computer under this status?

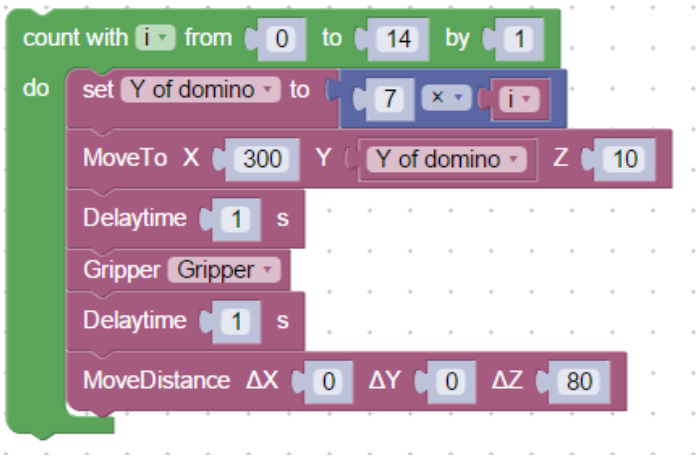
The emergence of programming languages builds a communication bridge between people and computers. The programming languages run the course from machine language, assembly language and advanced language. Sometimes, we also call programming languages “computer languages”.

The machine language means the compilations of all the instructions of the computer. After the invention of the computer, people sent the instruction sequences consisting of "1" and "0" to the computer for implementation. At this time, the procedures are binary documents.

Machine language is complex, so people have improved it, and used a few simple English letters, symbol strings to replace binary strings. For example, people use "ADD" to represent the addition, so people can easily understand the procedures, and change and maintain the procedures easily. The resulting language is called assembly language.

After this painful experience of communicating with computers, people are aware that a language which is close to mathematics or human language and is independent of the computer hardware shall be designed. In 1954, the first advanced language - FORTRAN was born, followed by more advanced languages, and FORTRAN, BASIC, C, C++, VC, VB, JAVA, etc. are applied widely.

Although the advanced languages are easier to understand than the original computer language, they also involve grammar and are difficult for the new learner. In this context, graphical programming software emerges at the right moment. Blockly and Scalch are representatives of the fully visual programming languages with well-defined programming modules, where users just drag the program modules directly to the editing area when used, and combine the simple functions as building blocks to complete the programming, without entering code.

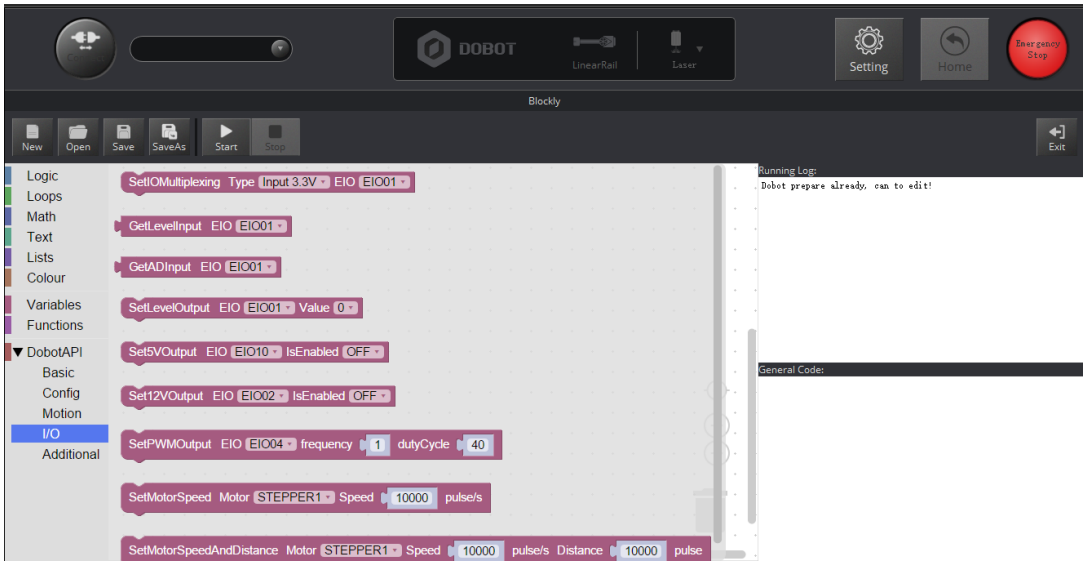


Which graphic programming software do you know? Let's talk about it.

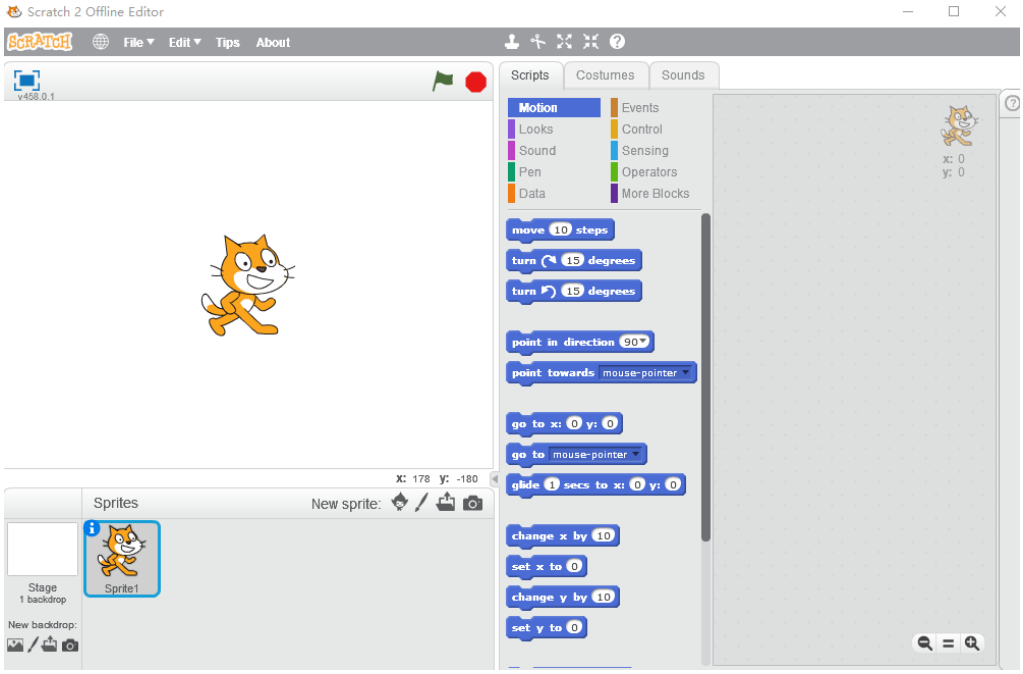
Dobot Blockly is based on the Google Blockly platform, which integrates Dobot API into the platform to facilitate control of the robotic arm. With the help of Dobot Blockly, we can easily control the robotic arm to achieve various functions. Next, let's get to know the interface of Dobot Blockly!

Open the DobotStudio software and click the Blockly module to enter the graphical programming interface.

There are Common Tool Button area, Function Modules (can be selected) and Editing area (drag the program module required by us to this side) on the top, left and middle respectively. The right side is divided into two parts: running log and code area. The running log shows the running status of the current program. In the code area, you can see how a program module built by us is represented in a high-level language.

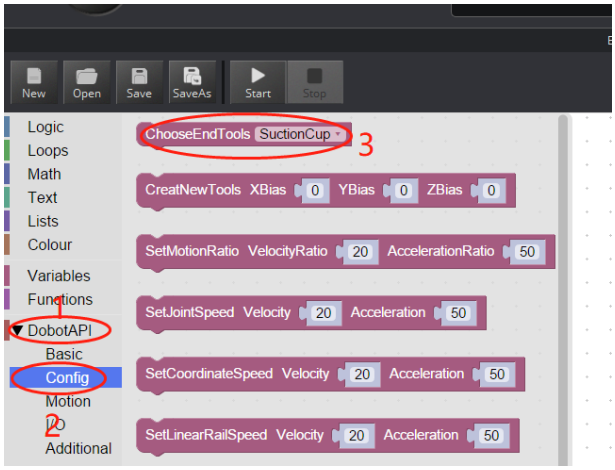
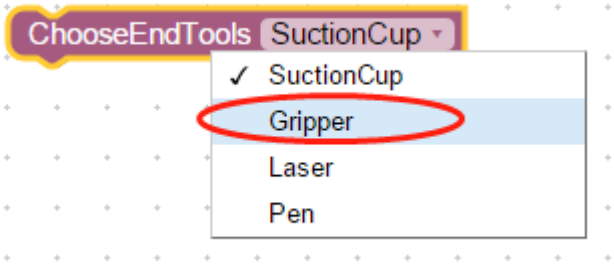
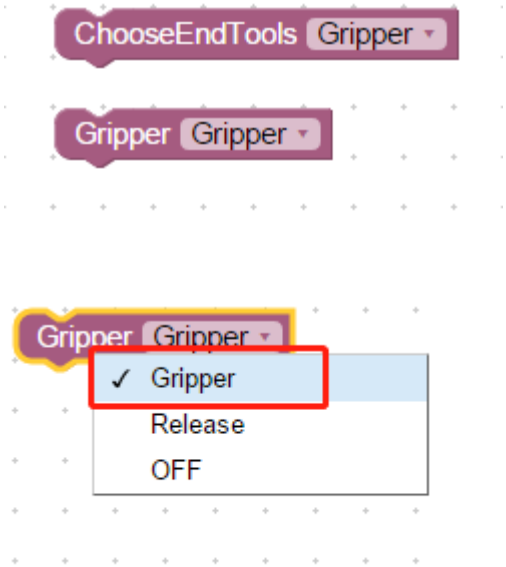


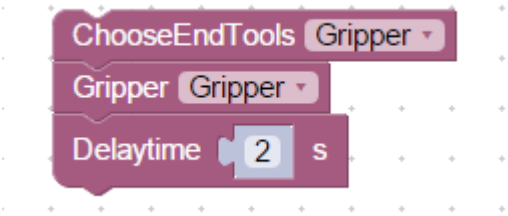
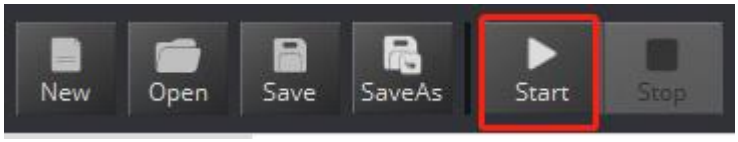
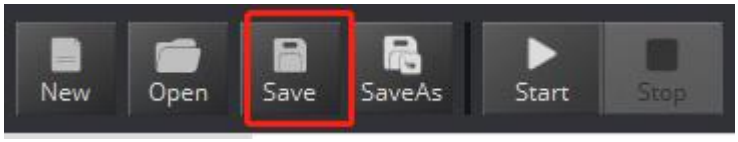
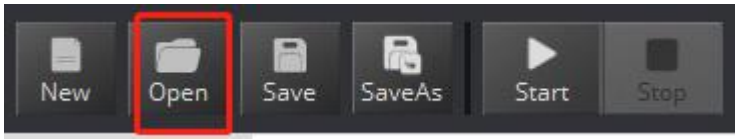
We can compare the Blockly main interface window shown in the figure above with the scratch software interface that we have learned previously and observe and analyze the differences.



1 Understanding of the Dobot Blockly interface

Click New in the "Common Tool Button area", and we will try to control the opening and closing of the gripper through programming.

Steps	Description of screenshots
1 Click the Dobot API module, select "Configuration", click "Select the Clamp", and drag the module to the editing area.	
2 Select the "Gripper" as the clamp from the drop-down list in the editing area.	
3 Click the Dobot API, select the "Gripper" module in the motion classification, and select "Open" from the drop-down list.	

4 Connect "Select the Clamp Gripper" and "Gripper Opening" modules, to complete the initial setting of the program.	
5 Click Start and run program.	
6 Click Save and save program. (It's a good habit to save the program so that next time we can open the document to continue writing the unfinished program or optimize the completed program.)	 

We just learned how to control the opening and closing of the gripper by writing a program. You can now try to control the suction cup by programming.

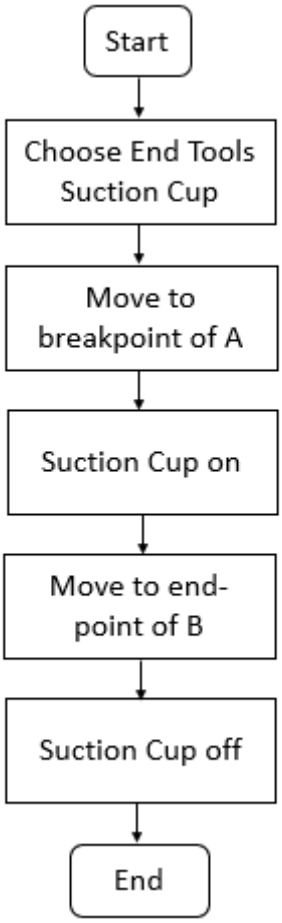
Creative design

Project practice: little expert in handling

Project requirements: The object in area A is transferred to area B, without being out of the frame or pressing the line.

Project score: The winner is the first to accurately place all the building blocks.

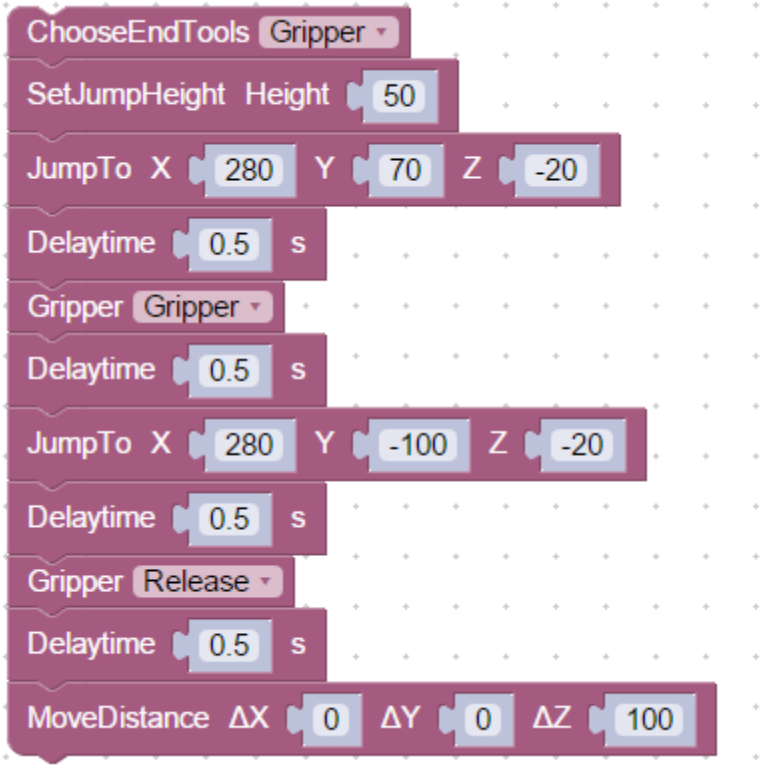
1 Expression of the action to be completed by the robotic arm with the flow chart



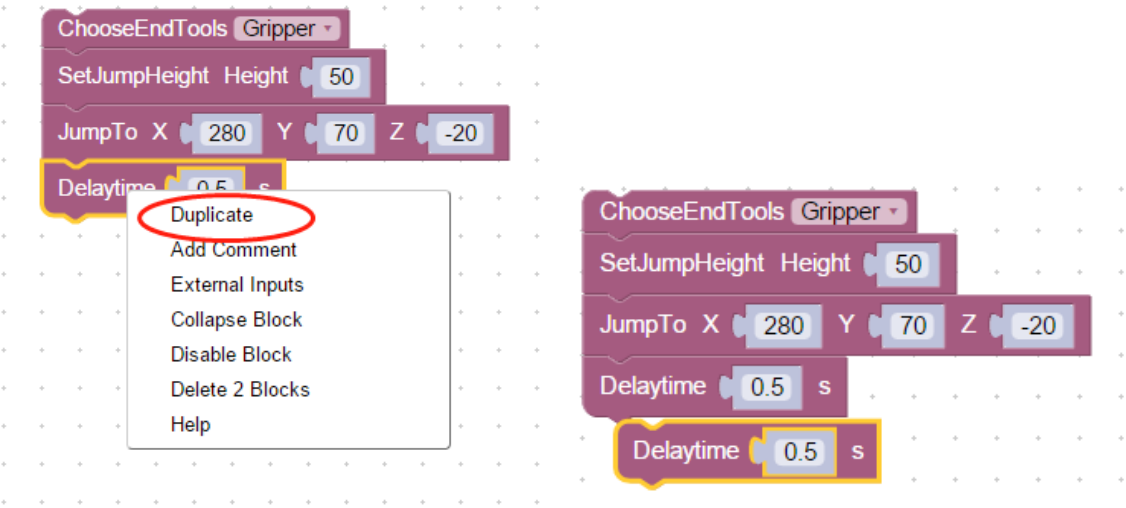
2 Setup of program modules

Based on the flow chart that has been built, find out the modules as required by us from the program modules and drag them to the editing area for assembly.

"Delay for 0.5 s" module appears 3 times in a row. If we drag it out from the program module area every time we use it, will it be very troublesome? Is there any way to facilitate subsequent operations?



When the same module is used several times in the program, direct copy is available. To be specific, right-click on the module, select "Copy" from the pop-up menu, copy the same module, and drag the module to connect to the program.



Knowledge sharing

1 How do the two program modules fit together? You can exchange your experiences with each other.

2 There are four icons in the lower right corner of the Dobot Blockly Program Editing area. Have you found out which functions they have?

Comprehensive evaluation

Sum up the learning contents of the class, and evaluate it.

Evaluation method	**Evaluation phase**	**Evaluation contents**	**Evaluation criteria**		
			3 Excellent	**2 Good**	**1 General**
Self-evaluation	**Scene creation lead-in**	• Understand programming language development history			
	Knowledgeability	• Be able to write the program of handling independently			
	Ability to solve problems	• Be able to perform programming to complete handling tasks according to the task requirements			
	Engagement	• Participate in class discussion and express ideas actively			
	Team collaboration	• Be able to cooperate with others.			

Nous contacter !

0 820 820 081
Service 0,12 € / min
+ prix appel

Technologie Services
Zi du Gavé
42330 Saint-Galmier
Fax : 04 77 94 55 14
Mail : contact@technologieservices.fr

Inscrivez-vous !

Visitez notre

Site internet

www.technologieservices.fr

Un espace dédié

- Écoles primaires
- Technologie collège
- Lycées technologiques
- Lycées professionnels
- Arts plastiques
- Matériel d'équipement
- Matériel audiovisuel et bureautique

Et toujours la possibilité d'établir des devis, de commander et de suivre vos commandes/factures en ligne.