

# Graphic programming design based on embedded AI processor

Li lin<sup>1</sup>, Yin Shangxia<sup>1</sup>, Li Kai<sup>1</sup>

<sup>1</sup>Beijing Institute of Technology, ZHUHAI

**Abstract:** This paper designs a system platform for graphical programming. The system is based on the open source software platform developed for the second time, which can realize the layer-by-layer conversion of diagrams and source codes, allowing users to deeply understand the logical relationship of programming, and programming and the host can be downloaded. The user can complete the drag-and-drop procedure, the icon shows the basic elements of the function, and the attribute box can modify each variable and automatically build it. After the user builds a graphics program with this system, click the run button to generate Python code and save the project.

## 1. Introduction

With the development of the world, the topic of "intelligence" has become quite popular. For the intelligent industry, the first reaction is robots. Robotics is a modern automation tool that integrates ergonomics, industrial engineering and other complex technologies in most disciplines, such as mechanical principles, control engineering, circuit science, computer science, and sensor manufacturing. Since robots need to implement functions and complete tasks through high-level programming languages, it is difficult for people who do not have a programming foundation to understand abstract programming languages.

Since robots need to implement functions and complete tasks through high-level programming languages, it is difficult for people who do not have a programming foundation to understand abstract programming languages. In order to realize robot programming like a building, a new programming method-graphic design is gradually derived.

In the field of open source design, Arduino designed a graphic design software of Arduino block constructed by graphic blocks. Scratch is a children's programming tool designed and developed by the Massachusetts Institute of Technology (MIT). Under the guidance of the software, a 6-year-old child can complete the program according to the system's pre-settings. Nanjing Ziguang Robot developed a graphic design platform for robot teaching for lower grade students. The software uses a programming method for selecting the direction of the object. The graphical programming software developed by DJI has a graphical programming interface and human-computer interaction functions.

In many graphics design software at present, the type

and number of modules are fixed and cannot be expanded. The fixed module type limits the function of the graphic programming logic and cannot exceed the existing scope. In view of the existing problems of the above-mentioned graphical programming, this article will compare the robot design, module operation design, module layout design and optimization of existing graphical design software, and provide simple and convenient module operation and excellent interface layout logo.

## 2. Visual programming platform

The platform is developed by Google and is based on Blockly. On this basis, the programming language code is converted and displayed simultaneously through visualization and coding. Learners can provide more help. The platform uses Blockly as a library to create and define visualization modules. Based on the basic knowledge of the combination of transaction networks and control boards, the platform designs the interaction of sensors and liquid processors and other visual functions to achieve visual programming control. Graphical programming contains many functional modules. The platform combines requirements analysis and design principles, and designs its functional modules in accordance with the overall framework design pattern. It is mainly divided into three parts: graphic design module, language module and hardware connection module. Graphical programming modules are divided into nine categories in total, namely network, control, logic, operation, text, variables, arrays, functions, and vision. The graphical programming system interface design is shown in Figure 1.

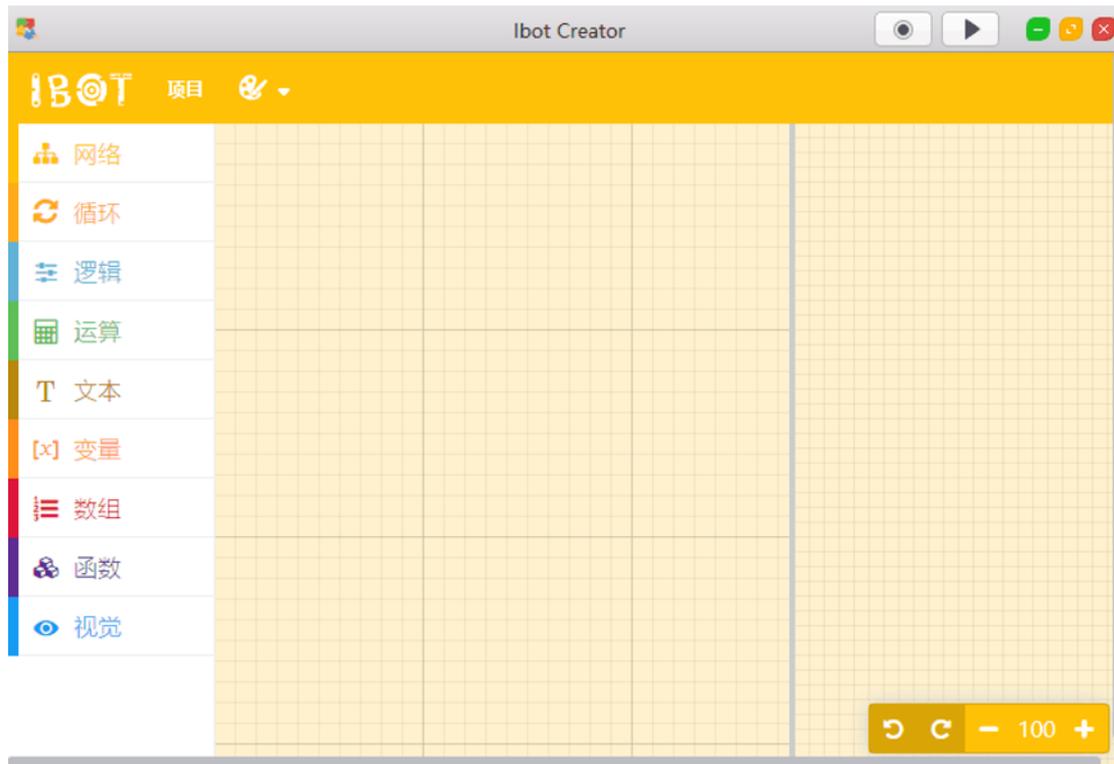


Fig.1 Interface design

### 3. Modular design

The modular design can achieve the high cohesion and low binding characteristics of the system, and can be combined with this closely related content. Each module has the same attributes in a software module. They are located at the vertex position, length, width, and inverse port. Therefore, it can not only be related to common module connections and attributes, but also inherit characteristics from four module groups. But in addition to common attributes, each category defines different parameters, mouse clicks, and movement events. The special complexity between the interlaced module and the switch module will make a special definition for this module. Modular design can reduce the complexity of the system and make the software easier to reuse.

#### 3.1. Structural classification of modules

The module program consists of multiple connected graphic elements. In order to meet the various program requirements combined with the graphical requirements of process control, execution, and sensor types, the module elements are divided into three structures, which are divided into loop structure, condition structure, and single structure.

##### (1) Single structure

As shown in Figure 2, A single structure is a channel, which can only start in one direction and end in the other direction, such as LED on and off and delay.



Fig.2 Single structure

##### (2) Loop structure

The module of the cyclic structure is a requirement for providing the cyclic function in the program design, as shown in Figure 3. The first of the cyclic structure is the start module, and the latter is the end module. It can be considered that the cyclic structure is a special structure of a single structure. In order to facilitate the operation of the user, the two single structures are combined into a special fixed pattern, and the two user's own loops are inserted into the two content that needs to be circulated. The content makes it easy for users to use.



Fig.3 Loop structure

##### (3) Condition structure

The conditional structure is a module that implements branch selection in the process of programming. In the conditional structure, the former is a conditional judgment module with one input and two outputs (true and false), and the other is an end module, as shown in Figure 4. Conditional judgment is carried out according

to the parameter setting of the conditional judgment module. If it is true, the executed content will be inserted after output, otherwise it will be inserted after false output.

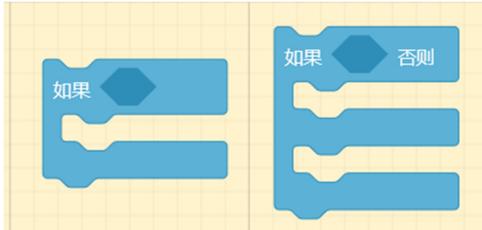


Fig.4 Condition structure

### 3.2.Module packaging

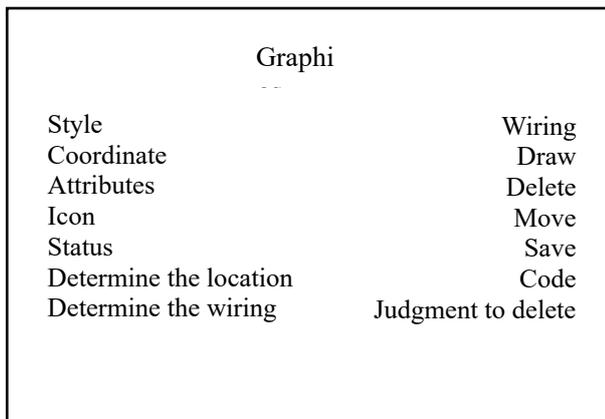


Fig.5 Module package structure diagram

The package of the module should be able to fully describe all the attributes and parameters of a common graph. Therefore, in order to meet the various needs of the user design, it should be combined. For example, when the user performs operations, such as dragging, moving, building, removing, etc., the graphics and rows are encapsulated into building blocks, including the name, type, location, and status. , Attributes, icons, etc., as shown in Figure 5.

## 4. Graphical programming and code generation

The user can design the graphics program through the operation of the mouse, including movement, construction, removal and parameters. There are three operations on the mouse: left mouse button and right mouse button. When clicking the mouse, it mainly controls the current click position. When you put down the mouse, you need to perform tasks such as element generation and element connection. When you move the mouse, you need to perform the task of element generation and element connection, and move.

### 4.1.Graphical attribute design

Attributes will be mainly used in the response attribute window, that is, the display and composition of image attributes, each variable and return value representing the function, characteristics and the characteristic requirements of the program, etc. The property window interacts with the property variables in the chart, reads and saves property values. With the special graphic design matching the property window and the regular function as the goal, it is designed as a general property window with expandable functions. As shown in Figure 3.10, it is the parameter adjustment of a camera. The schematic diagram before and after the parameter setting of the attribute is shown in Figure 6.



Fig.6 Graphical parameter adjustment

Click the mouse to pop up the attribute box, and take down the value of each variable in the current image, set a variable in the corresponding attribute box in each variable option, and then transfer each variable to each variable in the image. In the characteristics, to represent each variable of the image.

### 4.2.Code generation technology

After the graphic programming is completed, the automatic code generation mechanism set by the system converts the graphic language to python language code. The automatic code creation mechanism mainly uses a tree structure. The drawing software abstracts it into a tree structure, and each picture will be counted as a tree node. Tree browsing will be used to explore image programs to realize automatic generation technology.

A tree is a set of nodes, a hierarchical structure, with only one node, the node has no pre-line nodes, and the node has no subsequent nodes. According to the characteristics of the graphic program structure, the intermediate factor of the linear structure can only be a pioneer and a follow-up factor, which cannot satisfy the conditional structure of program design, so it is abstracted into a tree structure diagram. In actual applications, subscriptions start from the root node, and nodes of different nodes can be accessed in sequence. The flow chart is shown in Figure 7.

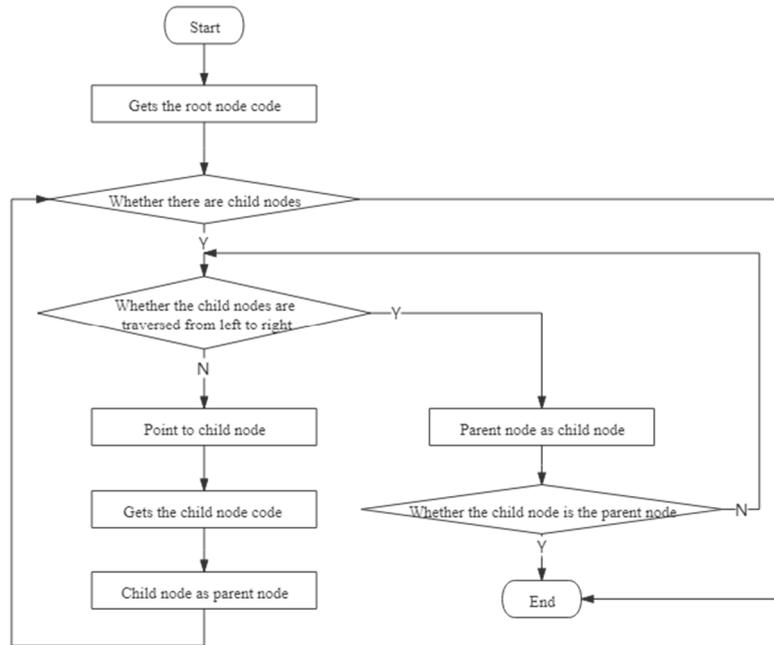


Fig.7 Code generation

Take the initialization of face mask recognition as an example, as shown in Figure 8. Complete the graphical program of face mask recognition initialization. According to the graphic splicing relationship, the graphic program is abstracted into a tree structure.

Get the root node A, start the pre-order traversal of the tree from A, A>B>C>D>E>F>G, divide the hierarchical nodes according to the structure of the tree, extract the graph code, and get the python language after integration Code.



Fig.8 Automatic code generation diagram

## 5. Summary

This paper develops a graphic design system, and realizes a programming platform based on graphic design, graphic classification, and graphic basic packaging. Based on graphic design, complete the related functions of graphic programming, including creation, movement, splicing, deletion and attributes. The file will be used to construct the image program, and the file conversion interface is designed to realize the two-way conversion between the image program and the file. This paper designed the interactive interface of the graphic library, completed the design of the dynamic image library.

## References

1. Khamphroo M , Kwankeo N , Kaemarungsi K , et al. MicroPython-based educational mobile robot for computer coding learning[C]// 2017 8th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES). IEEE, 2017.
2. Piccirillo J. The art and science of selecting robot motors[J]. Unpublished draft paper. Accessed: March, 2009, 1: 2013.
3. Howard A G , Zhu M , Chen B , et al. MobileNets:

Efficient Convolutional Neural Networks for Mobile Vision Applications[J]. 2017.

4. J. Redmon, S. Divvala , R. Girshick, et al. You Only Look Once: Unified, Real-Time Object Detection[C]. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016
5. J. Redmon, A. Farhadi. YOLO9000: Better, Faster, Stronger[C]. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, 6517-6525