Preliminary Design of Interactive Visual Mobile Programming on Educational Robot ADROIT V1

Adnan Rachmat Anom Besari, Sritrusta Sukaridhoto, Iwan Kurnianto Wibowo,
M. Herwindra Berlian, M.A. Wildan Akbar, Yohanes Yohanie F.P., Aldi Bayu K.I.

EEPIS Robotics Research Center (ER2C), Politeknik Elektronika Negeri Surabaya
anom@pens.ac.id, dhoto@pens.ac.id, eone@pens.ac.id

***Abstract*—The development of robotics have a great potential to be integrated with educational system. Robotics become more common in elementary and high school students. However, the absence of a module which integrates hardware and software makes harder for them to learn robotic. To help students learning robotics by using an easy to build robot with a dedicated interactive Graphical User Interface (GUI) to program it, we propose a module that integrates the simplicity of modular robot with feasible and appealing mobile visual IDE programming apps. A modular robot which allows student to build the robot without have to build it from the scratch, and Mobile visual IDE is an android based apps which allow user to program robot in GUI with a drag and drop icons followed by property editor. The icons represent robot commands such as moving and interacting with objects. This module also support advanced robotics features such as line follower, wall follower and free movement. The evaluation results demonstrate the visual appearance, ease of operation, the level of enjoyment, and benefit considered quite good by the respondents.**

Keywords: Robotics, Modular Robot, Education, Visual Programming, Android

I. INTRODUCTION

Robot development becomes a popular topic in educational system. To give a deeper understanding of robot, many researcher and robot vendor develop an educational robot. Educational robot made for introducing robotics and its function in educational purpose. Some of educational robots made for elementary and high school students. Generally, an educational robot will have a separated software and hardware [1][2][3]. Most educational robots are programmed using the conventional way which requires knowledge of how to code. In addition, most educational robots commonly use a fixed mechanic not a modular robot thus makes it harder for the student to build them. To address these problems, Politeknik Elektronika Negeri Surabaya launch another educational robot which named as ADROIT V1.

ADROIT V1 is a low-cost modular educational robot platform which has various features and modular design in mind. The overall parts categorized into three main building block, mechanical building block, electronics building block, and programming building block [5]. This robot needs a support apps to work together. The apps must be user friendly and support all features of modular robot.

In this paper we propose a new novell apps for ADROIT V1. By using android based computer tablets, this apps will improve the mobility and flexibility to program educational robot. Children no need to bring notebook or personal computer to play around with the robot, they just bring the computer tablet which dimension and weight are smaller than personal computer or notebook. This app is not using Google Blockly interface like dash and dot's programmer apps [5]. This apps use block icons which each of them represent source code of robot abilities. Each block supported by different property editor to change the block parameter and constant. There are three main abilities of robot such as freely move, move as line follower and wall follower. Other additional ability such as delay, print text, make a sound, gripper to move object also included in this app. The user just choose the block icons and drag it into worksheet and then combine it with another blocks. The user is able to edit the constants, select the robot mode, or use the default setting in property editor. The data from each block in worksheet extracted and assembled into a packet data and then upload it into the robot use Bluetooth as media with an asynchronous serial communication.

Fig. 1. Educational Robot ADROIT V1.

This apps will improve the usability of ADROIT robot. Because not only works with modular robot, the user will explore the programming side for each ability of the ADROIT robot.

II. PREVIOUS WORKS

The Lego Mindstorms NXT contains software and hardware to build robots. The robot shapes is very flexible

because it using Lego bricks which can freely combined by the children. The robot is equipped with some sensor such as color sensor and sound, and also can drive motors equipped with encoders [8]. The software programmer is desktop based with a drag and drop features. It provided blocks which represent the robot ability supported with a property editor. It also has a mobile app to control robot movement manually.

Dash and Dot robot from wonder workshop contains a mobile apps for two robots dash and dot. There are four kind of mobile apps, Go, Path, Blockly and Xylo. Go is android and iOS based mobile apps which interact with both robots, freely move robot, change robot's LED color and play a sound. Path is an iOS based mobile apps which only works with dash. It can makes Dash moving by drawing path on the screen. Xylo is an iOS based mobile apps which works with dash and xylophone as add ons. This apps is to compose a simple music. Blockly is iOS based mobile apps to program Dash and Dot by snapping together puzzle pieces of code. Blockly is for ages 8 and up, but the others Xylo, Go and Path is for ages 5 and above. Squeak [9] and Microsoft Robotic Developer Studio [10] are also programing language that can be use to program a robot. But those applications cannot be use on tablet.

Prior version of ADROIT V1, PENS-ER v1.0 [4] contains a robot with a desktop based of visual programmer apps. The robot has a line sensor and ultrasonic sensor. It also has a gripper to move an object. The desktop based programmer apps is similar with Lego NXT which use a drag and drop features and property editor. The difference is in the block features. There is a line follower mode and gripper control to move an object for PENS-ER v1.0.

III. System Design

This module built from two architectures, Middleware of the ADROITV1 and its mobile apps architecture. Middleware implementation of this robot is the process of obtaining the instruction written High-level programming language to be translated into machine codes. Middleware is contained in the program in which the microcontroller includes all functions of the robot. Middleware is implemented here is the program contained in the microcontroller ATmega16 which include all the functions on the robot. Robot has the ability to read sensors and actuators move. Block diagram will illustrates some ability middleware system on the robot, including:

- Function to activate the line sensor.
- Function to activate the proximity sensor.
- Function to activate the left motor.
- Function to activate the right motor.
- Functionality to enable the gripper.
- Functionality to enable the LCD.

The above functions can be set with the values that have been determined, thus minimizing the error that occurred in the higher level programming. Block diagram of the middleware architecture design of educational robot has shown in Figure 2.

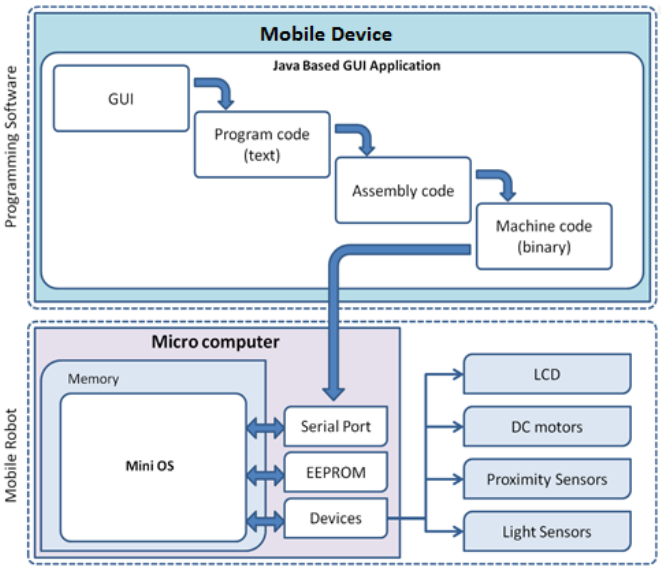


Fig. 2. Middleware Architecture Design.

The middleware architecture is built with several layers with a specific function on each, thus related to one another. Instruction data which comes through the serial port will be stored in the memory and translated into a bundle of tasks to be executed by the robot. After the task is given, the process is continued by giving instructions to each device's driver. In this phase, user can give a simple instruction to each device's driver because the driver is a simplified device accessing function. The system workflow has be shown in Figure 3.

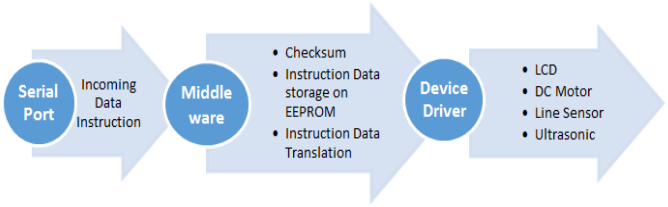


Fig. 3. System Workflow.

There is a mechanism on serial port block that contains the entire instruction data which sent by the software in form of a compilation of previously encoded data. Hence there will be a decoding process of the obtained instruction in the operating system. The method which is being used both in decoding and encoding has been agreed upon in hardware and software, so that both can understand the form of the sent information. Each information data which is being sent is a compilation of function codes and parameter, each of it has similar structure which also has function and parameter. However the number of program notation code can have different numbers. For example : free movement notation, alarm notation, line follower notation, wall follower notation, gripper notation, and delay notation. Program notation code for each block is :

```
<function_code><parameter-1><parameter-2><n-parameter>
```

For the application architecture consist of some component, it divided into user interface (UI) components, user interface action manager, event handler, drag and drop service, compiler and program files. All of them are saved in memory. Software architecture of this app can be seen in Figure 4.

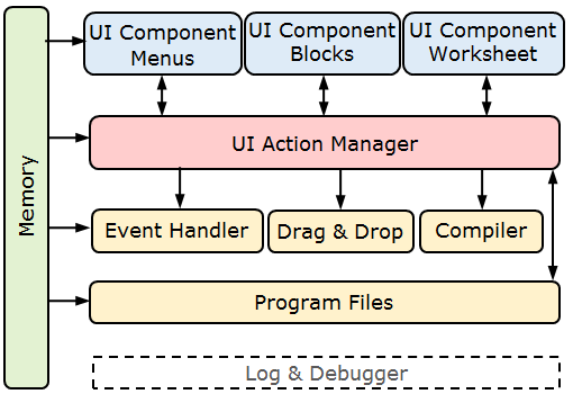


Fig. 4. Application Architecture Design.

- **Worksheet** is an area to drop the blocks. Blocks dropped on this area called active blocks. From active blocks compiler extract the data and convert to hexadecimal data protocols.
- **Property editor** allow user to edit parameters value for each block. Each block has a different properties editor. From properties editor user can choose mode from robot movement, change speed value, and many options depends from block type.
- **User interface action manager** manage all interaction from UI component and give response to user interaction. It manages program file, property editor, drag and drop service, compile data from each active blocks until send the data to robot by Bluetooth.
- **Event handler** manage the interaction between user and user interface. For example when the user press the button and button give a response.
- **Drag and drop service** allow the user to move blocks depends on user's touch coordinates on screen. When user touch the block icon, action manager will call this features to start moving the block. User can move block from menu to worksheet, switch position between blocks, or drag to trash icon to delete the blocks.
- **Compiler** extracts data from each block on worksheet, convert it to hexadecimals data protocols. And then assembles to be one packet data before send data to robot. Each icons has different data type and data length. All variable value will be converted to a hexadecimal code. The result will associated with our protocols. The protocols will use some header and followed by parameters.

```
<header> <parameter-1> <parameter-2> <parameter-n>
```

The compiler will generate that protocols code by extracting each active blocks on the worksheet. Each block has different header and parameters and will be combined to be one full packet data. Below is the header and parameters mapping for robot movement data protocols.

- **The hexadecimal program** code generates by the compiler send to the robot via Bluetooth with asynchronous serial communication. Robot receive that code and translate it to move the actuator according to the value and data header. Data header used to difference for each robot ability.
- **Program files** save user data when they start to create a new file, delete file, open file, or edit file. Memory is the place for save all application data and programming configuration

Headers and parameters was build according to robot's basic function. There are some additional headers and parameters if we give additional features to the robot. The implementation of code above for example when the robot moving forward in line follower mode with speed=255 and the duration = 5 seconds. The compiler will generates code below:

- Function header : moving forward line follower mode based of time (header code = a0)
- First parameter : time 5 seconds (convert to hex = 05)
- Second parameter : speed 255 (convert to hex = ff)
- Program code : a005ff

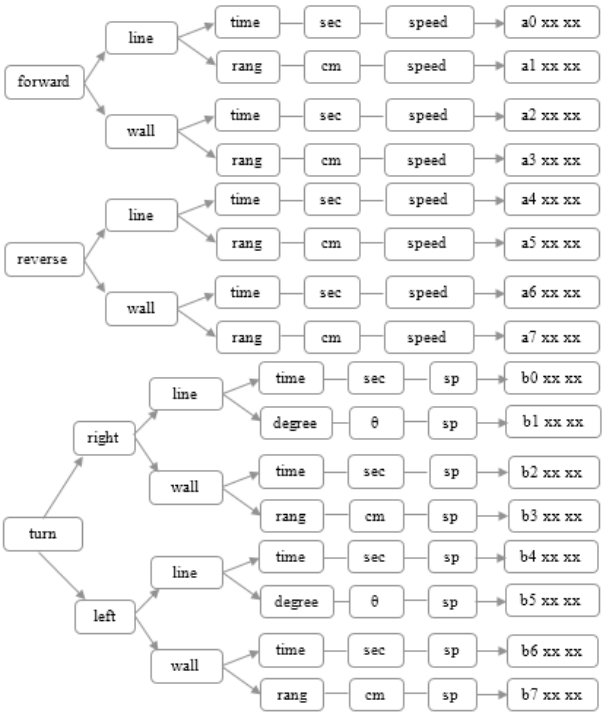


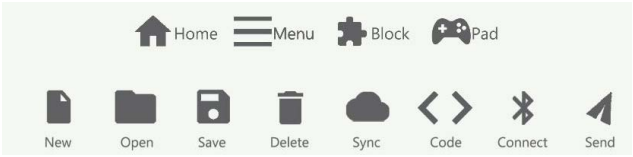
Figure 5. Headers and parameters mapping example

The description for each header and parameters has been shown in Table 1.

TABLE 1. MAPPING CODE DESCRIPTION

Move	Mode	Base	Code	Parameters	length (bytes)
Forward	Line follower	time	a0	2	3
		range	a1	2	3
	Wall follower	time	a2	2	3
		range	a3	2	3
Reverse	Line follower	time	a4	2	3
		range	a5	2	3
	Wall follower	time	a6	2	3
		Range	a7	2	3
Turn Left	Line follower	time	b0	2	3
		degree	b1	2	3
	Wall follower	time	b2	2	3
		degree	b3	2	3
Turn Right	Line follower	time	b4	2	3
		degree	b5	2	3
	Wall follower	time	b6	2	3
		degree	b7	2	3

The user interface component consist of all component that visible by user. It is contain menus, blocks, and worksheet. Menus contain regular menu to make a new file, open file, save file, delete file, information, bluetooth pair, and upload program to robot. Each block has a different features and function like shown in Figure 6.



(a)

(b)

Fig. 6. Design and visualization of menu (a) and blocks (b)

Below is the list of blocks and its function :

- **Direction** can control robot to move in 4 direction, forward, backward, left, and right.
- **Delay** makes the robot not to do anything in certain duration in second.
- Gripper can control the gripper robot to take or to release object
- **Sound** can turn on the buzzer and make some different sound types.
- **Display Text** can display message on LCD text
- **Line Follower** is a simple mode to program a line follower robot.
- **Wall Follower** is a simple mode to program a wall follower robot.

There are different blocks on the menu. Each block has different property editor which located on the bottom screen. The next figure shows the interaction between blocks and user. To start programming user just drag the blocks from blocks menu to drop zone on worksheet. Another version from this app shown by figure 16 for omnidirectional robot. This version has different type of blocks and feature for path mapping. The main GUI of this apps shown on Figure 7.

Fig. 7. Visual mobile programming preview

This apps has been tested on Samsung Galaxy Tab 4 8". The general specification of the device use Android v4.4.2 Kitkat as the operating system, Qualcomm APQ8026 chipset, 1,2 GHz Quad-core CPU, 1.5 GB RAM, 16 GB internal memory, 800 x 1280 pixels resolution (189 ppi pixel density) and V4.0, A2DP Bluetooth. The result is robot move according to blocks combination. Serial communication between robot and apps worked well, and it could mapping perfectly. We can see the robot's path and its error because the robot also send coordinates data for each coordinates path. The overall result is apps ran well without any trouble or lags because it consumed only 2,03 Mb in memory. All UI components worked very well including drag and drop service.

IV. SURVAY AND DISCUSSION

On the next stage of this educational robot programmer research is the evaluation for the next version development. Previously survay has been done for user of the educational robot programmer software, in terms of visual appearance, ease of operation, benefits and the level of excitement. Survay has been done by testing for multiple users. Users are asked to

assemble a mechanical robot, install sensors and actuators, put the hardware, and start programming. They were given a case study for some problems, then see if they can resolve the problem within the time. Once everything is completed, they were asked to fill out questionnaires a response to software programmers who have been tested. This response is the subject of the evaluation of the author to redevelop the software programmers that have been made to suit the wishes of the user. Here is presented a graph the results of questionnaires conducted on a sample of 50 respondents primary school students in the age range of 7 to 12 years old in SDIT Nurul Islam Krembung Sidoarjo as shown in Figure 8.

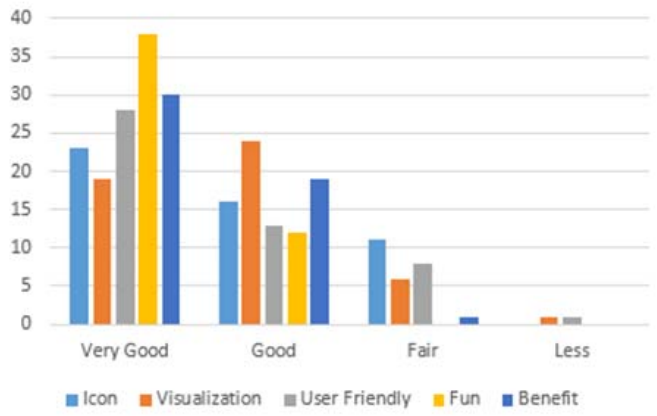


Fig. 8. The result of survay from 50 students from 7-12 years old in SDIT Nurul Islam Krembung Sidoarjo by using questionnaires.

The evaluation results demonstrate the visual appearance, ease of operation, the level of enjoyment, and benefit considered quite good by the respondents. However, there are several points to note are:

- Display visually robot is made more interesting for students.
- Operation of the hardware system needs to be tailored to the abilities of students, because not all students have the same basic electronics.
- The programming language is still considered too difficult by the students, so it requires a programming medium corresponding to the students language.

V. Conclusion and Future Works

This paper propose the preliminary design of interactive visual mobile programming on educational robot ADROIT V1. As an integrated educational robotics module that employs the easiness of modular design feature in ADROITV1, and equipped with interactive mobile programmer application that even student with no programming knowledge can use. This app has features such as user friendly, drag and drop service, a flexible variable value to change robot parameters. For the future works we will develop the application to be a multi-platform programmer apps, which not exclusive to ADROIT V1. The application will be able to be employed in various or general educational robot modules. Advanced robot features will be added with more complex menu in property editor. Programming level menu also will be added for novice, intermediate, and expert. We will give a direct manual control menu with a joystick visualization. So not only for student user, but we hope a wide range of user to play with programming robotics. This study tried to open access to multiple agencies, both government and the private sector as in order to build synergy for the development of educational robot. The form of cooperation include preparation robotics training curriculum and the design of competition for elementary, junior high and high school.

Acknowledgment

This research was supported by Local Research Grant on Politeknik Elektronika Negeri Surabaya (PENS) / Engineering in Electronic Engineering Polytechnic Institute of Surabaya (EEPIS) from 2015-2016.

References

[1] J.S. Patel, D.R. Patel, “Visual Programming Language for Educational Robot”. Journal of Information, Knowledge and Research in Electronics and Communication Engineering Vol.02, Issue-02. 2013.

[2] A. Felicia, S. Sharif, “A Review on Educational Robotics as Assistive Tools for Learning Mathematics and Science”. International Journal of Computer Science Trends and Technology (IJCST) vol. 02 issue 2 , 2014.

[3] F. Mondada, M. Bonani, X. Raemy, X.J. Pugh, C.Cianci , A. Klaptocz, A. Martinoli, “The e-puck, a robot designed for education in engineering”. Proceedings of The 9th Conference on Autonomous Robot Systems and Competitions Vol. 1, No. 1, pp. 59-65.

[4] J.Kharisma, A.R.A. Besari, F.Ardilla. “Perangkat Lunak Pemrograman Berbasis GUI Pada Prototipe Robot Edukasi”. Jurnal PA Politeknik Elektronika Negeri Surabaya 2013 Vol. 2.

[5] E.H. Binugroho, A.S. Khalilullah, D. Pramadihanto, R.W. Sudibyo. “ADROIT V1: Design of Low-cost Modular Educational Robot Platform”. The 16th International Electronics Symposium Recent Trends on Engineering Technology 2014, pp. A-6.

[6] EH Binugroho, ES Ningrum, DK Basuki, ARA Besari, “Design of Curriculum Matrix for Robotics Education Derived from Bloom's Taxonomy and Educational Curriculum of 2013”, IPTEK Journal of Proceedings Series, 2014.

[7] Wonder Workshop a home for Das and dot, robots that help kids to learn a code, http://www.makewonder.com.

[8] Lego Mindstorm, http://mindstorm.lego.com.

[9] Squeak Smalltalk, http://www.squeak.org.

[10] Microsoft MSDN Robotics, http://msdn.microsoft.com/robotics.