Combination Curriculum of Robotics and Mobile Phone in Primary Education Level with Graphical Programming Environment

Chi-Hung Tseng
Research and Develop Department,
CAVE Education
Taipei, Taiwan, ROC
e-mail: nissin@cavedu.com

Gi-Ming Liu
Department of Mechanical Engineering,
Tamkang University
Taipei , Taiwan, ROC
e-mail: 498500098@s98.tku.edu.tw

Wen-You Lu
Department of Mechanical Engineering,
Southern Taiwan University
Tainan, Taiwan, ROC
e-mail: 49812011@mail.stut.edu.tw

***Abstract*—Robots have proved effective in support of play-based teaching activities, especially at the primary education level. With the rapid improvement of the smart phone, we found many useful functions which robots can benefit from smart phones. For instance, robot can know where it is and where it is going from the information of the smart phone's GPS sensor; it can also realize its motion through the orientation sensor; Moreover, with the help of the large touch screen on the phone, user can easily view various kind of robot's information on the screen; Finally, robot can take advantage of the camera as its eyes for object tracking. All these features can be implemented with Android devices and LEGO NXT robots. At the same time, we realized that it is not practical to teach users become a skilled Java programmer, which is the official programming language of Android development. Thanks for App Inventor provided by Google, it is a graphical programming environment on the base of Google Application Engine. User with relatively no programming skills can easily build up there smart phone apps in the first run, therefore App Inventor gains worldwide received from elementary to high school teachers and non-computer-science related college student who intend to design smart phone applications. This topic is going to briefly discuss about the combination of robots and mobile phone applications in primary Education Level and as well how graphical programming environment is employed to improve students' learning performance.**

Keywords: primary education; robotics; Android; App Inventor; graphical programming;

I. INTRODUCTION

With the rapid development of intelligent mobile devices, we are now in a world that changing even faster, the meaning of the word "programming" is also shifting from traditional algorithm to now so-called fast prototyping, which means user takes advantage from existed SDK(software develop kit) and then use ready-to-use functions.

This topic is going to briefly discuss about the combination of robots and mobile phone applications in primary Education Level and as well how graphical programming environment, App Inventor, is employed to improve students' learning performance. The authors of this topic had established App Inventor TW website [1] and published a book for general accessibility.

Many topics [2][3] had illustrated that robotics curriculum is widely adapted to develop student's programming, mechanism and team working skills in a problem-based learning course design. O the other hand, graphical programming environment had been proved that it is a suitable aid to improve programming and logical thinking abilities of primary level students[4]-[6]. Here are the main advantages of graphical programming for beginners:

A. Easy to understand

When user dragged visual components on the screen, he/she has already decided the sequence of how the program is executed. In Fig 1, it is quite obviously that BC motors will keep rotating (means robot going straight) until it bumps into something then stop.

B. Visual and straight forward

When passing value or argument in traditional text-based programming language; it is not that easy for student to realize the concept that argument is "flowing" from one module to another. Take LabVIEW for example, the output of the sine block will directly passing to the Waveform chart module to show the value, meanwhile there is a green light to point out whether the sine value is greater than 0.8 or not (Fig 2).

However, graphical programming has several withdraws that it is hard to manage code (Fig 3.) and lack of executive efficiency.

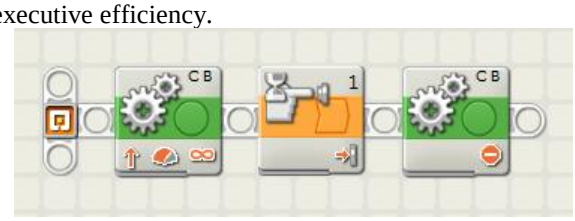

Fig. 1. Graphical programming is easy to understand

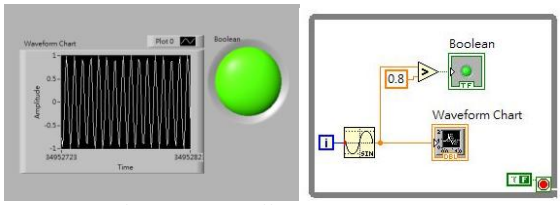

Fig. 2. Data flow between different modules.

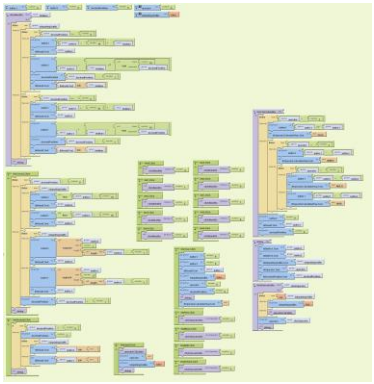

Fig. 3. Graphical codes become hard to read when getting larger.

II. App Inventor Graphical Programming Environment

App Inventor was first announced as a small project of Google Lab in late 2010, and had transferred to MIT Mobile Learning Center for public use under the spirit of open source. Anyone who can't wait to start playing can log into MIT App Inventor website [6] where they can access worldwide App Inventor users sharing their project and ideas.

App Inventor is a graphical and online programming environment, which can let users to "build up" their Android apps. You can develop applications for Android phones and tablets using a web browser, the App Inventor server will periodically store your project, you can log into App Inventor server then continue your work anytime and anywhere with a computer.

A. How we get our first app with App Inventor?

According to Fig. 4, App Inventor has two main components which let users build their apps sequentially:

- Designer: This is a webpage where users select the components for their app and adjust attributes of each component.
- Blocks Editor: This is a java window where users put the program blocks together to decide how the components should behave. The way to assemble programs blocks is visually and quite straightforward, just fitting pieces together like playing puzzles.

Fig. 4. App Inventor System structure, including Designer and Blocks Editor. Source:[http://www.appinventor.mit.edu/]

App Inventor provides a real-time adjustment function while users doing any modification on their app, which is quite different from traditional programming environment. In old days, programs need to be recompiled and downloaded again after any modification. When finished, users can package their app and produce a stand-alone application to install.

Users can upload their own app to Android devices. Don' worry if you don't have one, you can build your apps using the Android emulator (Fig. 5), this is software that runs on your computer and behaves just like the phone.

B. Take a glance of App Inventor components

App Inventor can be considered as an entry level Android development environment, in truth it is. App Inventor take good care of programming beginners therefore some technical terms had been modified in the purpose of user-friendliness. For instance, array in App Inventor is called "list" and all kinds of numeric data is treated as "number", which is not the situation in other text-based programming language; we usually have data types of integer, float and double in C and Java. Below is the brief introduction of App Inventor components:

- Basic: Most-used components such as button, label, input field and images.
- Media: components to play music/sound and video files. And camera component to take picture in your app.
- Animation: Movable components on the screen, you can control them by dragging or tilting the phone.
- Social: Telephony-related functions, such as phone-calling, texting and get contact list.
- Sensors: most-seen sensor on Android devices, such as accelerometer, orientation sensor and GPS.
- Screen Arrangement: components let users to arrange the layout of their app.
- LEGO® MINDSTORMS® : users can control Lego NXT robots through Bluetooth communication [7].
- Other Stuff: many interesting components are listed in here. For example, barcode scanner, Text to speech, speech recognizer, web database for online data storage and web components.

Fig. 5. Android emulator.

- Not ready for prime time: experimental or unstable components, will move to other categories later.

Users who are willing to learn can download detailed documents of these components on App Inventor TW website.

III. Combination of Android Devices and Robots

Students start playing their robot from setting motors' movement, such as "turn clock-wisely for 5 seconds". Then combine various kinds of sensors, so the commands become "Go straight until something is bumped then stop". Therefore it is always an important issue for robot that "Where am I?" and "Where am I going to?" which implies positioning and navigation. With comparison with our human beings, robot has to know the precise distance and azimuth between its destination otherwise it is unable to arrive there. This situation requires robot to be equipped with certain sensor such as GPS chip and orientation sensor. Luckily enough, you can solve this problem by one single smart phone.

A. What do we have on an Android device?

1) Powerful processing: Android devices nowadays are equipped with dual-core even quad-core processors, which is much faster than five-years-ago PCs. Therefore robot with an Android phone as its controller can process faster and handle larger amount of data.

2) Large touch screen: generally almost all Android devices have large touch screen, let users can do more things than traditional cell phones. On the other hand, large screen can display more information (pictures, video, etc). Fig. 6 is a demonstration of displaying sensor value of a Lego robot on the screen.

3) Sensors and camera: The most popular sensors on Android devices are accelerometer, orientation and GPS. These sensors are quite suitable for robots to perform navigation-related tasks. Furthermore, high-resolution camera on the phone can be appropriate vision solution for robot.

4) Internet connection: Robot can do even more when it was connected to the cloud. For example, robot may become a data gathering platform while send data back to remote server (Fig 6). On the other hand, users can remote control their robot through Wi-Fi or 3G telephone network. This is especially useful in some case like disaster field where is too danger for human to enter.

Fig 6. Display robot's sensor value on screen.

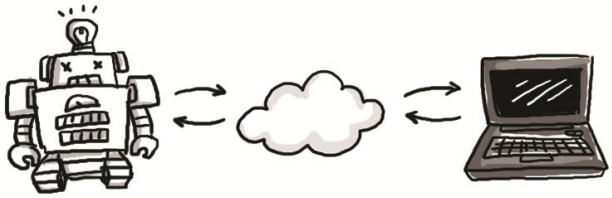

Fig 7. Duplex communication between computer and robot via a cloud. Source:[http://www.appinventor.tw]

B. Lego Mindstorms NXT Intelligent Bricks

Nowadays, Lego robot set bad become an important teaching aid worldwide[8][9]. They consist of a set of robot components that can be used to construct robots that have light, touch, ultrasonic and sound sensors. The components can also be used to construct robots with three types of motion driven by an electric motor. Lego Company provides students with all the basic building blocks for construction of robots. Students can play and experiment with the Mindstorms NXT components, create robots of their own design, assemble them quickly and try them out to see whether the robots will perform as the designers had intended. Countless designs are possible [10][11]. For example, Figure 6 depicts a project requiring a student to design a robot that detects and reacts to the brightness of a light source.

C. What's more after these two are integrated?

In the past, it was a serious issue to remote-control a robot with a hand-held device. This is a master-slave structure that the robot (slave) will continue receiving commands from hand-held devices (master) to determine what to do. First of all, you have to maintain two source codes, one for master and one for slave. Once any adjustment is needed, you have to modify all of them. What is ever worse that programming environment of master and slave devices are different in most cases, indicating that you have to be familiar with tow kind of programming languages. That's why control robot is not easy in old days.

Lego robots support a special format of command" NXT Direct Command", which can let robot directly receive byte array then execute correct movements with a running programming. This is really helpful that users can concentrate on one programming language (For Android, it is Java) instead of taking care of various kind of robot languages. This is not seen on other robot platform.

Fig. 8. Robot with the ability to detect light

We summarize some interesting projects after robot is integrated with a smart phone.

1) South-Pointing Chariot: The 3-axis orientation sensor on a Android device can be used as a direction guidance for robot. Robot can tell its included angle with target by orientation sensor value. More detailed, robot should turn right while the current heading (e.g. 150 degrees) is less than the target heading (e.g. 180 degrees). The simple if/else condition can be further extended to proportional control method, which is an extremely important topic in machinery and engineering related departments.

2) Object tracking/face recognition and surveillance: Don't for the high-resolution camera on your Android device. Robot can become more powerful after some visual computing library such as OpenCV is added. Base on this functionality, robot will be able to handle complex visual processing issues, such as solving Rubik's cube, object tracking and even face recognition.

On the other hand, through TCP/UDP protocol, an Android device can send back its camera images as video streaming back to remote PC through Wi-Fi or 3G telephone networks. Compared with an expensive wireless IP-camera, it should be a relatively cost-effective solution to adapt Android device as video input.

3) Google Map Route tracking robot: The most obvious advantage of Android is the perfect integration of various kinds of Google services. Therefore it is possible that a robot can follow the Google Map route? The author had written an Android app to parse the route returned from Google map and locate each corner's coordinates.

Fig. 9. South-Pointing Chariot by orientation sensor.

Fig. 10. A robot equipped with an Android phone as video input.

First, user will decide the start and end point of the route for robot to travel. Second, send these tow point to Google Map, and then Google Map will return a route. The app can calculate how long to travel and where to make a right/left turn from each two points. Finally the distance and heading will be transferred to motor encoder value for robot to read. This is a substantial demonstration that robot can use Android device as a direction guidance.

IV. Community for Future Teachers in Taichung

Began from September, 2011, Community for Future Teachers was raised for teachers of information & computer and living technologies. This community tried to help teachers to develop courses with the assistance of mobile devices. One of the authors had been invited as the tutor of this community, using App Inventor to develop mobile applications such as internet connection, drawing board and simple Google Map search. Teachers had also realized with mobile devices, there will be more flexibility on their curriculum design and daily routine works. For instance, a teacher from National Chung-Hsing Senior High School had developed an application to scan students' exam papers, automatically calculate the average score and generate a bar chart. Finally, the app will fill out an email form if a student is going to be flunked in this subject.

(a) (b)

Fig. 11. (a) Decide the start/end point for robot to travel.(b) Parse route to get the distance and turning angle.

Fig 12. One of the teachers control robot with a Android tablet, Community for Future Teachers, 2011.

A. How to use App Inventor in Science course?

After one semester's training, teachers of this community had highly agreed that App Inventor is suitable as teaching aids in information & computer and living technologies courses, as listed below:

1) Basic programming concepts: With App Inventor's graphical environment, students can quickly establish the basic programming concept and skill without being trapped in frustrating debugging works.

2) Interactive design: With sensors of the Android device, it is easy to build up an interactive project. For example, users can shake their phone to simulate a bubble tea shaker. Students will learn how to get accelerometer value and further calculate it.

3) Robot and other embedded systems: With Bluetooth components, App Inventor can communicate with many other devices. Not only Lego robots, such as Arduino and Basic Stamp microcontrollers.

In October 2011, there was a robot competition event held by Vanung University. Unlike traditionally autonomous robot, this event required participants to use a handheld device to control their robot to travel through an obstacle course. On the spot, the most popular solution is App Inventor with Lego robots, and it is a positive phenomenon that not only students from information or electrical engineering department but multimedia and design related department had joined this event. Because robot needs more the programming skills, including mechanism, material, human factors engineering and industrial design. This situation can be considered as a clarification that students with no programming background could also demonstrate beyond-average abilities on robotics or other related issues.

Fig. 13. A student practiced to make his robot get on a slope, Vanung University, 2011.

Fig. 14. A student controlled his robot to travel through a maze, National Yunlin University of Science & Technology, 2011.

V. Publication and Website

For more accessibility for general users, the author of this topics had published a tutorial book "Android App So Easy – App Inventor" in 2012[13]. This book is written for anyone who would to give a try of mobile programming but lack with coding skills. On the other hand, App Inventor TW website was established at 2011, and keep updating and gathering interesting apps and topics. Please join us on App Inventor TW website.

Besides the website, the authors had went outdoors and held more than twenty App Inventor workshops nationwide from elementary schools to colleges in 2011.

VI. Conclusion

Robotics curriculum is an appropriate practice for student to learn many skills, including programming, mechanism and problem-solving abilities. But robotics is not very easy for beginners because it has so much to catch up in a short time. Meanwhile, robot can do more after connected with an Android device with the function of visual computing, motion detection and internet connection. Instead of Java programming, these can be easily implemented by App Inventor.

We can consider App Inventor as an appropriate tool in information education and robotics curriculum as listed below: 1. Fun!! 2. Lower the entry age down to elementary school students. 3. Easy to understand and less programming skills required. 4. More motivation and willing to learn. 5. The easiest solution to control a robot with a mobile device so far.

On the other hand, App Inventor is somehow limited that the codes are hard to manage when getting larger and it is not easy to implement complex control structures, therefore users cannot implement a large scale application by App Inventor. As a matter of fact, App Inventor can be considered as a middleware for children and software beginners to practice; and for advanced users to build up their prototype in a very short time.

As mentioned in the previous paragraph, the authors had founded App Inventor TW website to integrate population and resource of mobile learning and information education. Since open on October 2011, App Inventor TW has daily traffic of more than 200 visits, which means more than 6,000 visits monthly.

The authors had tried to provide some advices in this topic for teachers and student who want to get into robotics and mobile programming.

References

[1] "App Inventor Learning Protal," 2012. [Online]. Available: http://www.appinventor.mit.edu/.

[2] M. Resnick, and S. Ocko, "LEGO/Logo: Learning through and about design," in Constructionism, I. Harel, and S. Papert, Ed. Norwood, NJ: Ablex Publishing, 1991.

[3] B. Denis, and S. Hubert, "Collaborative learning in an educational robotics environment," Computers in human behavior, 17(5-6), 465-480, 2001.

[4] K. Lau, H. K. Tang, B. Erwin, and P. Ptrovic, "Creative learning in school with LEGO programmable robotics products," presented at the ASEE/IEEE Front. Educ. Conf., San Juan, Puerto Rico, 1999.

[5] G. Jones, C. Langrall, C. Thorton, and S. Nesbit, “Elementary students’ access to powerful mathematical ideas,” In L. English (Ed.) Handbook of international research in mathematics education. (pp. 113-141). NJ: Lawrence Erlbaum Associates, 2002.
[6] F. G. Martin, “Circuits to control: Learning engineering by designing LEGO robots,” MIT Press, 1994.
[7] “MIT App Inventor,” Massachusetts Institute of Technology, 2012. [Online]. Available: http://www.appinventor.mit.edu.
[8] [Text in Chinese] Y. X. Lin, and C. H. Tseng, “Android / NXT : Control Robot by Smart phone,” Fullon Culture, 2011.
[9] C. H. Tseng. “A Study of Developing the Technological Curiosity Inventory and Its Application to LEGO Hands-on Learning,” Master thesis, National Taipei University of Education, Taipei, 2007.
[10] B. Erwin, M. Cyr, and C. Rogers, “LEGO engineer and ROBOLAB: Teaching engineering with LabVIEW from kin-dergarten to graduate school,” *Int. J. Eng. Educ.*, vol. 16, pp. 1–12, 2000.
[11] Lemons, G., Carberry, A., Swan, C., Jarvin, L. and Rogers C. The benefits of model building in teaching engineering design. Design Studies, 31, 288-309. 2010.
[12] E. Cejka, C. Rogers, and M. Portsmore, “Kindergarten Robotics: Using Robotics to Motivate Math, Science, and Engineering Literacy in Elementary School,” International Journal of Engineering Education, 22(4), 711-722, 2006.
[13] [Text in Chinese] C. H. Tseng, I. T. Tsai, K. C. Huang, W. M. Lai, W. Y. Lu and S. L. Wei, “Android App So Easy – App Inventor,” Fullon Culture, 2012.