Implementation of an Arduino Simulator to Support Circuit Design

Tatsumi Nishida[*], Tetsuro Katayama[*], Yoshihiro Kita[†],
Hisaaki Yamaba[*], Kentaro Aburada[*], and Naonobu Okazaki[*]
[*]University of Miyazaki, 1-1 Gakuen-kibanadai nishi, Miyazaki, 889-2192 Japan
[†]Tokyo University of Technology, 1404-1 Katakuramachi, Hachioji City, Tokyo 192-0982, Japan
E-mail: nishida@earth.cs.miyazaki-u.ac.jp, kat@cs.miyazaki-u.ac.jp, kitayshr@stf.teu.ac.jp,
yamaba@cs.miyazaki-u.ac.jp, aburada@cs.miyazaki-u.ac.jp, oka@cs.miyazaki-u.ac.jp

Abstract

Embedded technologies are used everywhere. One of the educational materials to learn the embedded technologies is Arduino. When a beginner designs circuit, it is possible to design the circuit that damages Arduino itself or the modules on it. Therefore, this research implements an Arduino simulator that runs on iPad for supporting to design a circuit. In this simulator, the circuit on iPad is analyzed to detect the value of voltage on the circuit and the risk that Arduino itself or the modules are damaged.

Keywords: embedded technologies, Arduino, simulator, iPad

1. Introduction

Embedded technologies are used everywhere and are indispensable in our daily life.[1] For example, there are cars, car navigation systems, air conditioners, televisions, and so on. As the demand for embedded software increases, its development is also diversifying. One of the educational materials to learn the embedded technologies is Arduino. Arduino is used around the world as learning kits.[2,3] We can learn embedded technologies by designing a circuit and controlling the circuit by programming.

The learning kits that summarize sensors and jumper wires are also on the market, and Arduino has the advantage that even a beginner who have never designed a circuit can easily develop it. However, when a beginner designs a circuit, it is possible to design the circuit that damages Arduino itself or the modules on it.

Therefore, this research implements an Arduino simulator ADVIS (Arduino VIrtual Simulator) that runs on iPad for supporting to design a circuit. In ADVIS, the circuit on ADVIS is analyzed to detect the value of voltage on the circuit and the risk that Arduino itself or the modules are damaged. These features can support to design a circuit on Arduino.

2. Arduino Uno

Arduino has various kinds such as Arduino Uno, and Arduino Leonardo, and so on. ADVIS developed in this paper simulates Arduino Uno.

Fig.1 shows an overview of Arduino Uno. Arduino Uno is a digital input / output device equipped with 8 bits microcomputer. Arduino Uno itself has 32KB flash memory. Therefore, even if it is turned off, the program can be saved in it. Arduino Uno flows electric current of 20 mA of current from each output pin. Based on the above specifications, the value of the electric current flowing from each output pin of ADVIS is 20 mA.

In Arduino, depending on the circuit designed by a user, it is possible to design the circuit that damages Arduino itself on it. Specifically, there are the followings.

- A risk of overvoltage when output pins are connected

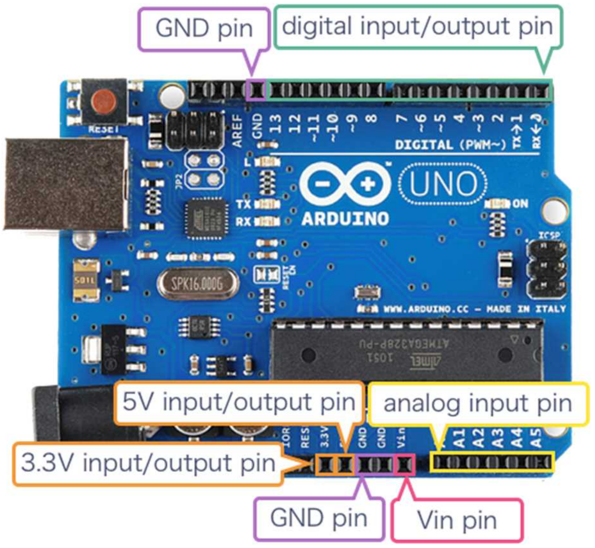

Fig. 1. An overview of Arduino Uno

- A risk of damaging the blocking diode in the Arduino itself when the voltage exceeding 3V is sent to the input pin
- A risk of overcurrent if the input pin receives an electric current exceeding 200mA

In the implementation of ADVIS in this paper, only the first and second risks as above can be detected.

3. Arduino simulator ADVIS

The requirements of ADVIS is as follows.

- iPad Pro 12.9 (2nd generation)
- iOS 10+

Fig.2 shows an overview of ADVIS. ADVIS consists of the following three parts.

- Circuit design part
- Module selection part
- Menu part

3.1. *Circuit design part*

In the circuit design part, a circuit is designed using jumper wires, LEDs, and resistors.

3.2. *Module Selection part*

In the module selection part, a module to be handled in for the circuit design part is selected. ADVIS can handle jumper wires, LEDs, and resistors only. The LED has a resistance value of 100Ω, and when a voltage of 3 V or more flows, the LED lights up in green. The resistor has a resistance value of 150Ω.

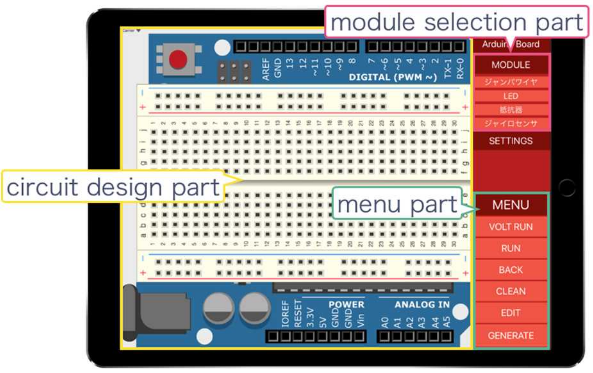

Fig. 2. An overview of ADVIS.

3.3. *Menu part*

In the menu part, it is implemented the "CLEAR" function, the "BACK" function, the "RUN" function, and the "VOLT RUN" function.

Using the "Clear" function removes all modules that are placed in the circuit design part and returns to the initial state.

Using the "BACK" function returns the circuit design part to its previous state. Here, immediately after using the "CLEAR" function, you cannot use the "BACK" function.

Using the "RUN" function, the designed circuit is executed to detect the risk of module breakage. Fig.3 shows an example of the “RUN” function (Here, the image is edited for easy-to-read). When the risk of a module breakage is detected, highlight the input pin of the module in red.

Using the "VOLT RUN" function displays the value of voltage on the designed circuit. Fig.4 shows an example of the “RUN” function (Here, the image is edited for easy-to-read).

3.4. *Function of ADVIS*

In ADVIS, in addition to the risk of damaging the Arduino itself described in Section2, the risk of damaging the module can be detected. It’s target module is only LED. LED is damaged if it receives voltage more than 3V.

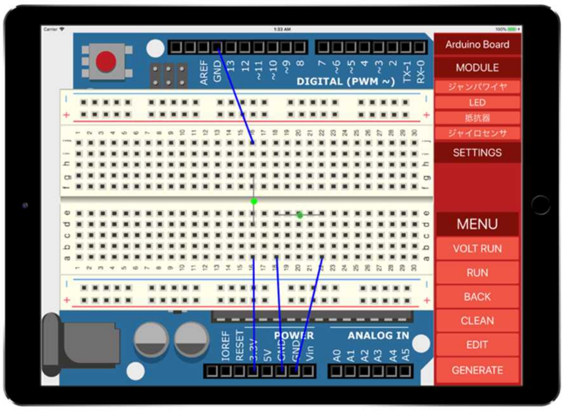

Fig. 3. Example of the "RUN" function

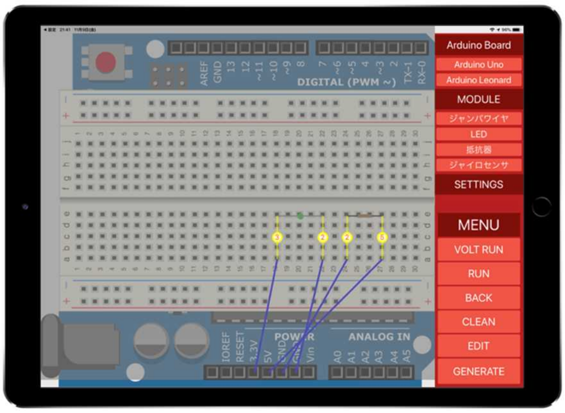

Fig. 4. Example of the "VOLT RUN" function

4. Verification if ADVIS

It is confirmed that ADVIS can simulates correctly the designed circuit. The followings are verified.

- Displaying of the value of voltage on the circuit
- Detecting a risk of Arduino itself being damaged
- Detecting a risk of module being damaged

4.1. *Displaying of value of voltage on the circuit*

To display the value of voltage on the circuit, it can be displayed by using the "VOLT RUN" function. As an example, connect 5V pin and Vin pin to the breadboard respectively. Fig.5 shows the result that the "VOLT RUN" function is executed.

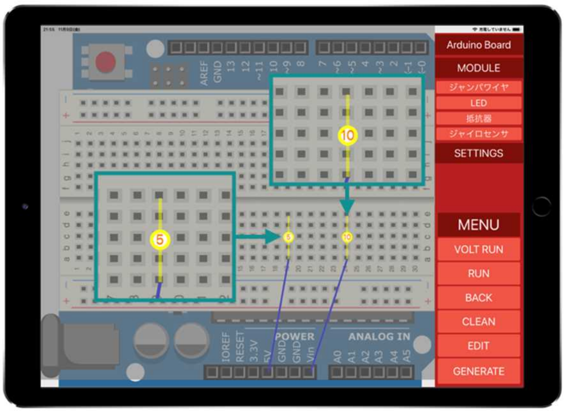

Fig. 5. Displaying the value of voltage on the circuit

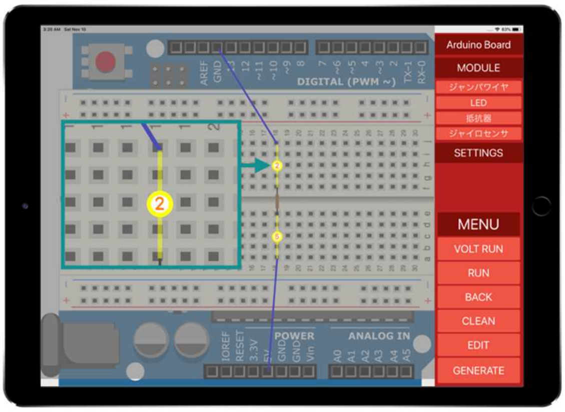

Fig. 6. Displaying of value of voltage using resistor

From Fig.5, 5V is displayed on the circuit connected to the 5V pin, and 10V is displayed on the circuit connected to the Vin pin.

Next, the resistor and the 5V pin to the breadboard are connected. Fig.6 shows the result that the "VOLT RUN" function is executed.

From Fig.6, 5V is displayed on the circuit, and 2V, which is calculated from Ohm's law ($V = 0.02 \times 150$), is displayed on the circuit.

Therefore, it can be confirmed that ADVIS displays the value of voltage.

4.2. *Detecting a risk of Arduino itself being damaged*

In order to detect where there is a risk of corruption of Arduino itself, it can be displayed by using the "VOLT RUN" function. As an example, connect 5 V pin and Vin

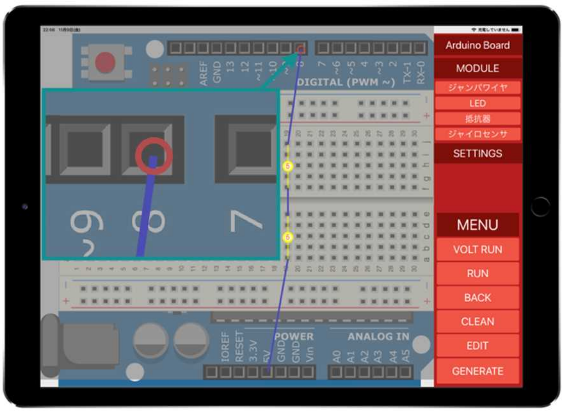

Fig. 7. Detecting a risk of Arduino itself being damaged

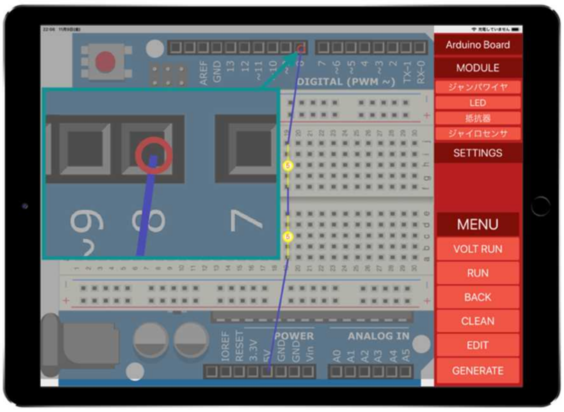

Fig. 8. Detecting a risk of the module being damaged

pin to input / output pins. In this circuit because the voltage exceeding 3V is followed to the input pin, there is a risk that the blocking diode of the Arduino itself will be damaged. Fig.7 shows the result that the "VOLT RUN" function is executed.

From Fig.7, ADVIS can display the Arduino input / output pins in red when the risk that the Arduino itself is damaged is detected.

4.3. *Detecting a risk of module being damaged*

In order to detect where there is a risk of corruption of modules, it can be displayed by using the "RUN" function. As an example, connect the LED to the 5V pin. In this circuit, there is a risk of damaging the LED because it receives the value of voltage higher than 3V. Fig.8 shows the result that the "RUN" function is executed.

From Fig.8, ADVIS can display the input pin in red when the risk that the module is damaged is detected.

5. Conclusion

This research has implemented an Arduino simulator ADVIS (Arduino VIrtual Simulator) that runs on iPad for supporting to design a circuit. The implemented ADVIS has the following functions.

- Designing a circuit
- Visualizing the value of voltage in the circuit
- Detecting a risk of Arduino itself being damaged
- Detecting a risk of module being damaged

Consequently, ADVIS can support to design a circuit. Future issues are as follows.

- Save function of the designed circuit
- Circuit control by programming
- A function to automatically generate a program based on the designed circuit
- Implementation of resistors with various values and other modules

References

1. Masayuki Hirayama: Software Engineering for Embedded System, Current State of Embedded Software Development, IPSJ Magazine, Vol. 45, No. 7, pp. 677-681, 2004 (in Japanese).
2. P. Jamieson, "Arduino for teaching embedded systems. are computer scientists and engineering educators missing the boat?", *Proc. FECS*, pp. 289-294, 2010.
3. Mohammed El-Abd, A Review of Embedded Systems Education in the Arduino Age: Lessons Learned and Future Directions, International Journal of Engineering Pedagogy, Vol. 7, No.2, pp.79-93, 2017.