Journal of The Korea Convergence Society
Vol. 12. No. 10, pp. 45-53, 2021
I-SSN 2233-4890 / e-ISSN 2713-6353
DOI : https://doi.org/10.15207/JKCS.2021.12.10.045

A Study on the Virtual Remote Input-Output Model for IoT Simulation Learning

Hyeon-Ho Seo[1], Jae-Woong Kim[2*], Dong-Hyun Kim[3], Seong-Hyun Park[4]
[1]Ph.D. Student, Dept. of Computer Engineering, Kongju National University
[2]Professor, Dept. of Computer Science & Engineering, Kongju National University
[3]Professor, Dept. of IT Artificial Intelligence, Korea Nazarene University
[4]Ph.D., Dept. of Computer Engineering, Kongju National University

IoT 시뮬레이션 학습을 위한 가상 리모트 입출력 모델에 관한 연구

서현호[1], 김재웅[2*], 김동현[3], 박성현[4]
[1]공주대학교 컴퓨터공학과 박사과정, [2]공주대학교 컴퓨터공학부 교수, [3]나사렛대학교 IT인공지능학부 교수, [4]공주대학교 컴퓨터공학과 박사

Abstract In our technology-driven world, various methods for teaching in an educational venue or in a simulated environment have been suggested especially for computer and coding education. In particular, IoT related education has been made possible owing to the industrial developments that have occurred in various fields since the Fourth Industrial Revolution. The proposed model allows various IoT systems to be indirectly built; it provides an inexpensive learning method by applying a simulation system in a 3D environment. The model is implemented on Virtual Remote IO based on the Arduino platform, thereby reducing the cost of building an education system. In addition various education-related content can be provided to learners through such an indirectly developed system. Test code was written to check the consistency of an operation between the real system and the virtual system.

Key Words : Virtual simulation, Physical computing, Internet of Things, Arduino platform, Educational contents

요 약 교육 장소에서 실제 수업하거나, 시뮬레이션 환경에서 교육하는 방법에 대한 방향이 제시되고 있다. 4차 산업혁명 이후에 다양한 분야의 산업 발전이 이루어지고 있고, 특히 IoT와 관련된 교육이 실행되고 있는 실정이다. 제안 모델은 3D 환경에서의 시뮬레이션 시스템을 응용하여 큰 비용 없이 다양한 IoT 시스템을 간접적으로 구축하여 교육할 수 있는 모델로서, 아두이노 플랫폼을 기반으로 가상 리모트 IO를 구현하였으며, 이를 통하여 교육을 위한 시스템 구축비용의 경감과 시스템을 간접적으로 구축하여 학습할 수 있는 모델이다. 또한 교육과 관련된 콘텐츠들을 다양하게 실습할 수 있다. 테스트 코드를 작성하여 실제 시스템과 가상의 시스템과의 동작 일치성을 확인하였다.

주제어 : 가상 시뮬레이션, 피지컬 컴퓨팅, 사물 인터넷, 아두이노 플랫폼, 교육 콘텐츠

*Corresponding Author : Jae-Woong Kim(jykim@kongju.ac.kr)
Received September 6, 2021 Revised September 14, 2021
Accepted October 20, 2021 Published October 28, 2021

1. Introduction

In a technology-driven world, where questions arise regarding the objectives of engineering education, proposals on how to effectively teach in an educational venue or in a simulation environment are highly desirable [1,2]. A virtual learning environment is a representative educational method that can facilitate class courses, the provision of educational materials, and communication; it is also widely used in higher education curriculums [3]. Recently, in educational institutions such as schools, computer-based educational programs are gradually expanding to replace and supplement traditional educational methods. Computer-based virtual simulations are used in various ways to facilitate engineering education [4]. In particular, software education has had an extremely positive effect on creativity, computational thinking, mathematical thinking, logical thinking, problem solving ability, and development of a competitive spirit [5,6]. Unlike other subjects, software education depends on the ability of the learners to write and create programs on their own [7]. Hence, software education through an online learning environment is effective because the learners can immediately execute and check their programming results and learn by themselves through feedback [8,9]. Therefore, it is necessary to develop various content related to software education that enables self-directed learning online. The Internet of Things (IoT), which has emerged since the Fourth Industrial Revolution, is not only a key element in developing smart factories at manufacturing sites by combining with existing embedded systems, it is also widely used as a home IoT system and for do-it-yourself (DIY) hobbies [10,11]. Following this trend, courses using IoT systems are being established in coding education for children and non-professionals [12]. However, such courses are either based on hardware for practical training or simulations to check simple operations in a two-dimensional (2D) environment; the training situation is assumed in advance in most cases, decreasing the learner's understanding and interest. Therefore, a model that can indirectly build and educate various IoT systems at low cost by applying a simulation system in a three-dimensional (3D) environment is proposed in this study. The proposed model is expected to reduce the cost of building a system for education and provide an environment in which interesting educational content can be experienced in various ways.

2. Related Works

2.1 Arduino Platform

Arduino is an open-source platform configured to make electronic circuits and programming easier to achieve using a micro controller unit (MCU) [13]. Because it is well modularized, Arduino is easily connected to peripheral devices, such as various types of sensors and motors. It is widely used in an IoT education environment as it provides a library to easily set up and use complex register settings that normally require hardware knowledge. In particular, a USB (universal serial bus) is installed in the circuit by default, making it easy to upload the program; 5V of power is supplied, allowing a simple system to be run without a separate power supply [14]. The platform also provides UART, I2C, and SPI, which are built-in communication drivers, and various wired/wireless communication modules and libraries supporting them, allowing non-experts to easily implement the system. In addition, research on and development of its automation products, IoT, and wearable applications are perpetually expanding.

2.2 Physical Computing

Physical computing is a method for constructing a physical system capable of achieving an interaction between humans and computers [15], connecting the physical real world and the virtual world of a computer. It uses expanded input and output devices and information; the input devices can be extended to heat, light, and sound sensors, and the output devices can be used by extending various actuators such as motors and light-emitting devices (LEDs). Physical computing is capable of providing a learning environment in which learners can quickly and easily express their ideas through many different studies. Therefore, it is being used as an environment for learning in many fields such as the robotics, interactive art, environmental sensing, and scientific experimentation [16]. In addition, although physical computing has developed differently from computer science, it can be used to increase the creativity of learners in computer science education [17]. When non-specialists learn the programming process, they generally need to study many complex concepts within a limited time and thus, experience significant difficulty owing to a lack of mathematical background and interactive media content, which can provide rapid feedback [18]. Physical computing devices are in an ideal position to support the development of engineering and computing knowledge in learners owing to their ability to match computing knowledge and abstract algorithms with the real world [19].

3. Proposed Model

3.1 System Configuration

Fig. 1 shows a system configuration diagram for building an IoT device simulation environment; the configuration is composed of three main parts: the IoT device, the real IoT module, and the IoT simulator.

First is the IoT device; it is a system constructed using Arduino as a microcontroller controlling the entire IoT system and a power board to drive it. Users can control the digital or analog signals from sensors or actuators that are actually wired through the GPIO. In addition, the device communicates with the virtual remote IO through the UART library provided similarly to the GPIO function calls and transmits the input/output signals to the virtual IoT module. Second is the real IoT system; it is a peripheral device that is actually connected to an IoT device that transmits a module or external interrupt signal requiring a separate hardware test of the GPIO. Third is the IoT simulator installed on a PC; it is composed of a virtual remote IO and an IoT module. IoT modules are virtual devices implemented in 3D space where the signals generated are converted into serial communication signals through the remote IO.

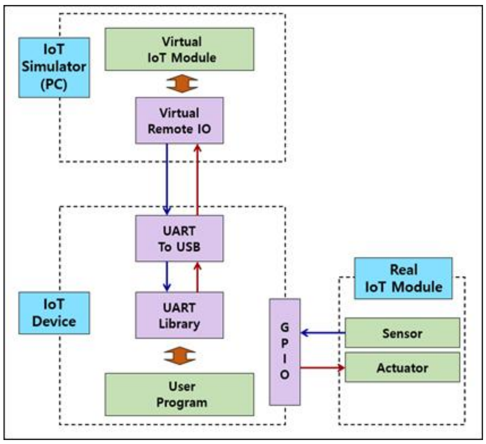

Fig. 1. System Configuration Diagram

3.2 System Process

Fig. 2 shows a system process for the IoT simulation, which consists of six steps.

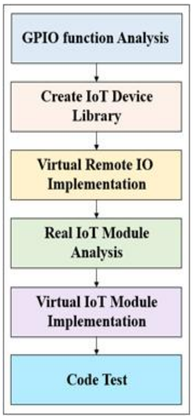

Fig. 2. System Process

First, the prototype of the input/output function of the GPIO suitable for the IoT device to be used needs to be analyzed. This allows the environment that controls the actual hardware and the environment used in the simulation to be as similar as possible for the user. Second, a library suitable for the language to be used in the IoT device needs to be created. In this study, a library was created in the form of a header file in accordance with C/C++, which is the development environment of Arduino. Third, a virtual remote IO module for connecting with the created library through a communication device needs to be implemented. The remote IO transmits the signal of the virtual IoT module implemented through a 3D simulation to the IoT device. Fourth, the physical/electrical characteristics of the peripheral devices used in accordance with the IoT system to be implemented need to be understood. Fifth, the analyzed I/O devices must be implemented in a virtual space. In this case, the educational effect can be increased by implementing sensors and actuators with various characteristics. Finally, a test code needs to be written to check the operational consistency between the real and virtual systems.

3.3 Virtual GPIO Function Library

Following Fig. 3 is a message processing routine.

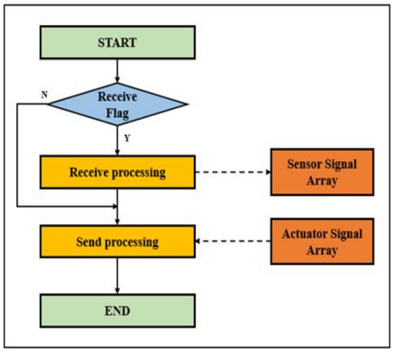

Fig. 3. Message Processing Routine

Recently, hardware related to IoT devices has implemented digitalRead, digitalWrite, analogRead, and analogWrite functions that exchange external I/O signals through the GPIO based on Arduino UNO R3, which is widely used in IoT education. However, in this study, a RemoteIO class was created and implemented as a lower method for using the existing GPIO control function at the same time. In addition, the function for an OPEN communication was configured in the form of Remote.begin (int baudrate) by imitating the existing UART OPEN function, Serial.begin (int baudrate), so that the system configuration is not confusing to learners, and the environment is the same as the existing user environment. Once the UART device is prepared using the Remote.begin() function, a message is processed periodically inside, as shown in Fig. 3] by using an MCU interrupt. The data of these sent and received messages are processed using an array declared for buffering; the GPIO function reads/writes the data in this buffer to the user. The data in the buffer consist of an array of bytes, and the size of the array is determined based on the number of inputs/outputs to be used in the system. The index number of the array corresponds to the address of the buffer, and each byte of data of the address is mapped to a virtual digital pin with 8 bits. Each digital pin mapped has a serial pin number; read/write data can be accessed through a bit mask technique. Fig. 4 shows the operation algorithm of digitalWrite among the GPIO functions.

```
CALL digitalWrite Function
INPUT 'Pin Number'              //type is integer
INPUT 'Write Data'              // type is 'true' or 'false'
Address = Pin Number / 8
Bit Number = Pin Number % 8
Mask Data = 0x01 << Bit Number
If Write Data is true
    Buffer[Address] = Buffer[Address] | Mask Data
If Write Data is false
    Bit inversion of Mask Data
    Buffer[Address] = Buffer[Address] & Mask Data
```

Fig. 4. digitalWrite Pseudo Code

3.4 IoT Simulator

Fig. 5 shows a block diagram of the IoT simulator. It is composed of a virtual IoT module that implements an IoT peripheral device on a computer and a remote IO to transmit the electrical signal of the virtual IoT module to the virtual GPIO function of the IoT device. Both processes exchange data using the shared memory of the inter-process communication (IPC).

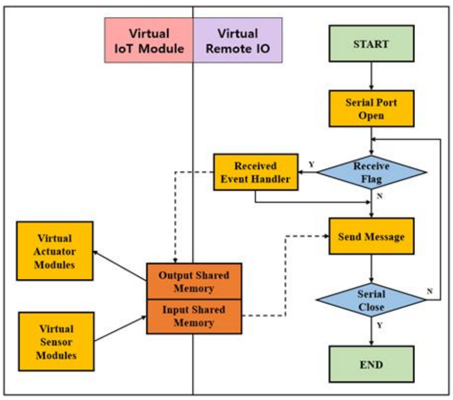

Fig. 5. IoT Simulator Block Diagram

Fig. 6 shows a class diagram indicating the relationship between the virtual sensor module and the input shared memory.

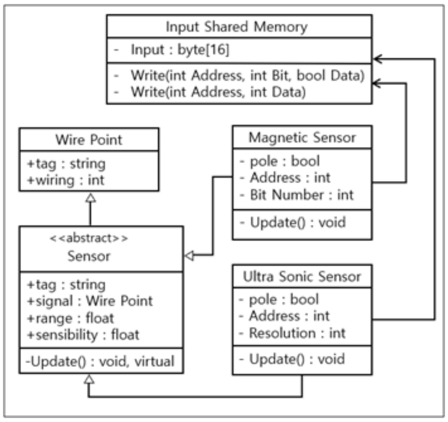

Fig. 6. IoT Simulator Class Diagram

4. Results and Discussion

4.1 Experimental Methods

The automatic guided vehicle (AGV) system to be applied during the project learning was configured as shown in Fig. 7 for the experiment conducted in this study. For ease of operation, a switch signal is placed at the top of the screen, and mode change functions such as "no line," "track driving," and "cross-track" have been added to practice various tracks.

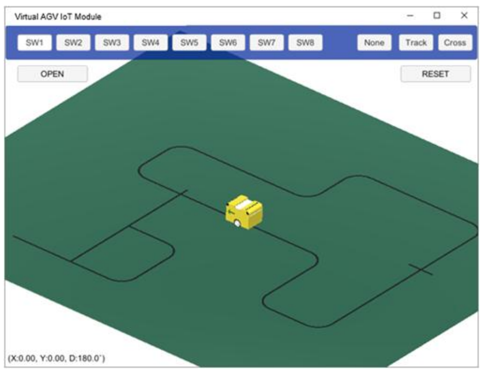

Fig. 7. Virtual AGV IoT Module

In addition, the virtual remote IO used to transmit the I/O signals of the virtual AGV IoT module is configured as shown in Fig. 8. The screen composition consists of the Arduino connection method, the basic communication setting of the COM port and baud rate, and a monitoring screen for convenient signal debugging for the learner.

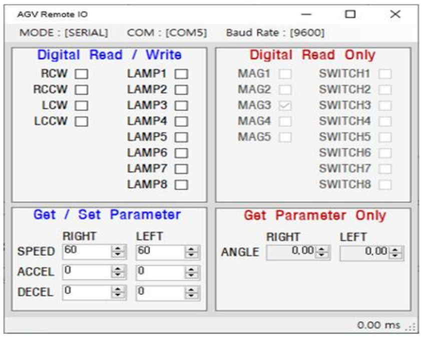

Fig. 8. Virtual AGV Remote IO

From the center, the left screen is a writable area and corresponds to the output signal sent from the Arduino to the virtual IoT module; the right screen is a readable area where the input signal of the virtual IoT module is sent to the Arduino. In addition, the digital and analog signals are grouped by placing the digital signals at the top and the analog signals at the bottom. The AGV system uses a line tracer method that is widely applied in the basic stages of IoT. To this end, it consists of two motors for driving the AGV, an encoder to detect the rotation angle of the motor, a magnetic sensor to detect the line, and a switch and a lamp to check the basic I/O. Hence, a message frame with a total data size exchange of 14 bytes was constructed Table 1.

Table 1. Total data size of IO signals used

Data Type	Data Size	Number of Data	Total Data Size
Digital Input	1 bit	13	2 bytes
Digital Output	1 bit	12	2 bytes
Analog Input	2 bytes	2	4 bytes
Analog Output	1 byte	6	6 bytes
Sending and receiving data size			14 bytes

The line tracer of the comparison target of the experiment was implemented as shown in Fig. 9 using the same configuration as the AGV virtual system.

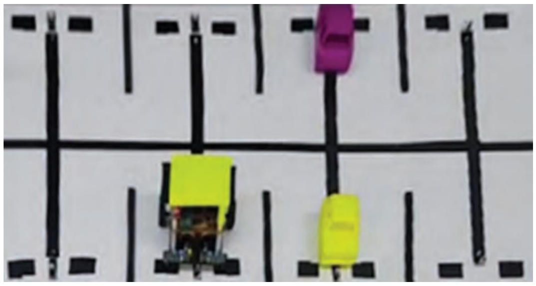

Fig. 9. AGV Real System

4.2 Comparison Analysis

The problem arising from the difference between the operational principle of the actual stepping motor and that of the simulator was solved by installing a motor driver. The communication speed of the simulation was 19,200 bps, which is the most stable speed for testing in the Arduino environment; the tests were conducted without using the delay function because doing so affects the loop period. For the experimental method, the real and virtual AGVs were controlled based on the firmware written using the same algorithm; the scan time of the Loop() function and the update period of the I/O signals were measured and compared. As shown in Table 2, the scan time of the Loop() function did not show a significant difference, although the actual input/output update period was over 5000-times different on average. However, this is a physical limitation resulting from the difference between the method of directly controlling the actual electrical signal with the register and the way the signals are collected and exchanged through communication. In fact, there were no cases in which the AGV deviated from the line in the motion analysis between the two systems; there was a slight difference in responsiveness when returning to the original position when the AGV deviated from the line.

Table 2. Comparison of reactivity differences

Real System	Loop Scan Time	1 ~ 2 ms
	Signal Update Cycle	5 ~ 10 us
Virtual System	Loop Scan Time	2 ~ 3 ms
	Signal Update Cycle	34 ms

5. Conclusion

The various objectives in the field of education related to engineering are a subject of intense discussion, and research regarding the direction of actual teaching at an educational venue or in a simulation environment is ongoing. In this situation, a virtual learning environment is being widely used as a representative educational method. Since the advent of the Fourth Industrial Revolution, developments have been made in various fields. In particular, composing smart factories in industrial sites where products are manufactured by combining existing systems with IoT architecture has become vital. To this end, coding education and educational courses related to IoT are ongoing, but most of the existing education is based on hardware or consists of simulations that can be used to check simple operations in 2D; in most cases, the education is applied under the assumption of necessary circumstances. Therefore, the interest and understanding of learners may decrease. Accordingly, many educational systems have switched from offline to online environments for various reasons in recent years. However, most educational systems that still require practical training apply group education owing to spatial or cost limitations. To overcome these limitations and build an environment in which online education is feasible, research on combining various simulation technologies and AR/VR is ongoing. In this study, a system that can apply IoT through simulation techniques is proposed to meet these societal trends. By implementing online content using a simulator similar to an actual IoT device, a practice environment that can arouse interest in learning is expected to be built. Research into the development of educational content of various techniques that can be used in systems requiring such control should be continued because the MCU used in IoT is optimized for real-time control.

REFERENCES

[1] L. Gil, E. Blanco, & J. M. Auli. (2000). The virtual laboratory concept applied to strain measurements. *European Journal of Engineering Education, 25(3),* 243-251.
DOI : 10.1080/030437900438676

[2] J. V. Nickerson, J. E. Corter, S. K. Esche & C. Chassapis. (2007). A model for evaluating the effectiveness of remote engineering laboratories and simulations in education. *Computers & Education, 49(3),* 708-725.
DOI : 10.1016/j.compedu.2005.11.019

[3] T. J. McGill & V. J. Hobbs. (2008). How students and instructors using a virtual learning environment perceive the fit between technology and task. *Journal of Computer Assisted Learning, 24(3),* 191-202.
DOI : 10.1111/j.1365-2729.2007.00253.x

[4] W. J. Shyr. (2010). Enhancement of PLC programming learning based on a virtual laboratory. *World Transactions on Engineering and Technology Education, 8(2),* 196-202.

[5] P. J. Rich, N. Bly & K. R. Leatham. (2014). Beyond cognitive increase: Investigating the influence of computer programming on perception and application of mathematical skills. *Journal of Computers in Mathematics and Science Teaching, 33(1),* 103-128.

[6] P. J. Rich, K. R. Leatham & G. A. Wright. (2013). Convergent cognition. *Instructional Science, 41(2),* 431-453.
DOI : 10.1007/s11251-012-9240-7

[7] S. Y. Lye & J. H. L. Koh. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12?. *Computers in Human Behavior, 41,* 51-61.
DOI : 10.1016/j.chb.2014.09.012

[8] B. S. Xia & E. Liitiäinen. (2017). Student performance in computing education: an empirical analysis of online learning in programming education environments. *European Journal of Engineering Education, 42(6),* 1025-1037.
DOI : 10.1080/03043797.2016.1250066

[9] P. E. Robinson & J. Carroll. (2017, April). An online learning platform for teaching, learning, and assessment of programming. *In 2017 IEEE Global Engineering Education Conference (EDUCON),* 547-556.
DOI : 10.1109/EDUCON.2017.7942900

[10] N. Shariat Zadeh, T. Lundholm, L. Lindberg & G. ranzén Sivard. 2016). Integration of digital factory with smart factory based on Internet of Things. *In 26th CIRP Design Conference, 2016, KTH Royal Institute of Technology Stockholm, Sweden, 15 June 2016 through 17 June 2016, Elsevier, 50,* 512-517.
DOI 10.1016/j.procir.2016.05.050

[11] K. J. Singh & D. S. Kapoor. (2017). Create Your Own Internet of Things: A survey of IoT platforms. *IEEE Consumer Electronics Magazine, 6(2),* 57-68.
DOI : 10.1109/MCE.2016.2640718

[12] H. Ning & S. Hu. (2012). Technology classification, industry, and education for Future Internet of Things. *International Journal of Communication Systems, 25(9),* 1230-1241.
DOI : 10.1002/dac.2373

[13] M. Banzi. (2009). *Getting Started with Arduino: Make.* O'Reilly Media, Inc.

[14] Y. A. Badamasi. (2014, September). The working principle of an Arduino. *In 2014 11th international conference on electronics, computer and computation (ICECCO), IEEE,* 1-4.
DOI : 10.1109/ICECCO.2014.6997578

[15] D. O'Sullivan & T. Igoe. (2004). *Physical computing: sensing and controlling the physical world with computers.* Course Technology Press.

[16] P. Blikstein. (2013, June). Gears of our childhood: constructionist toolkits, robotics, and physicalcomputing, past and future. *In Proceedings of the 12th international conference on interaction design and children,* 173-182.
DOI : 10.1145/2485760.2485786

[17] M. Przybylla & R. Romeike. (2014). Physical Computing and Its Scope--Towards a Constructionist *Computer Science Curriculum with Physical Computing. Informatics in Education, 13(2),* 241-254.

[18] M. Rizvi, T. Humphries, D. Major, M. Jones & H. Lauzun. (2011). A CS0 course using Scratch. *Journal of Computing Sciences in Colleges, 26(3),* 19-27.

[19] A. Vasilchenko, M. Venn Wycherley, M. Dyson & F. R. Abegão. (2020, January). Engaging Science and Engineering Students in Computing Education through Learner-Created Videos and Physical Computing Tools. *In Proceedings of the 4th Conference on Computing Education Practice, 2020,* 1-4.
DOI : 10.1145/3372356.3372372

서 현 호(Hyeon-Ho Seo) [정회원]

· 2020년 2월 : 공주대학교 IT융합학과(공학석사)
· 2020년 3월 ~ 현재 : 공주대학교 컴퓨터공학과(박사과정)
· 관심분야 : PLC, IoT, 시뮬레이션, 스마트 팩토리
· E-Mail : shhmetis@naver.com

김 재 웅(Jae-Woong Kim) [종신회원]

· 1983년 2월 : 중앙대학교 전자계산학과(공학사)
· 1988년 2월 : 중앙대학교 대학원 컴퓨터공학과(공학석사)
· 2000년 2월 : 대전대학교 대학원 컴퓨터공학과(공학박사)
· 1992년 8월 ~ 현재 : 공주대학교 컴퓨터공학부 교수
· 관심분야 : 소프트웨어공학, 빅데이터, 멀티미디어공학
· E-Mail : jykim@kongju.ac.kr

김 동 현(Dong-Hyun Kim) [정회원]

· 1986년 2월 : 중앙대학교 전기공학과(공학사)
· 2005년 2월 : 공주대학교 컴퓨터멀티미디어공학과(공학석사)
· 2010년 2월 : 공주대학교 컴퓨터공학과(공학박사)
· 2021년 2월 ~ 현재 : 나사렛대학교 IT인공지능학부 교수
· 관심분야 : 인공지능, 영상처리, 지식관리, 시뮬레이션
· E-Mail : dhkim@kornu.ac.kr

박 성 현(Seong-Hyun Park) [정회원]

· 2012년 2월 : 충남대학교 예술대학 음학학과(성악전공)
· 2017년 2월 : 공주대학교 IT공학전공(공학석사)
· 2020년 2월 : 공주대학교 컴퓨터공학과 박사
· 관심분야 : 인공지능, 컴퓨터음악, 빅데이터
· E-Mail : giornopark@daum.net