JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지 Vol. 25, No. 1: 96~101, Jan. 2021

AVR 기반 아두이노 호환 보드를 위한 통합 프로그래머

허경용[1*] · 류대우[2]

Unified Programmer for AVR-Based Arduino-Compatible Boards

Gyeongyong Heo[1*] · Daewoo Ryu[2]

[1*]Associate Professor, Department of Electronic Engineering, Dong-Eui University, Busan, 47340 Korea
[2]CEO, NEWTC Corp., Seoul, 06698 Korea

요 약

아두이노는 오픈 소스 마이크로컨트롤러 프로젝트 중 하나로 AVR 마이크로컨트롤러를 사용한 아두이노 보드가 흔히 사용된다. AVR 마이크로컨트롤러에서는 일반적으로 ISP(In System Programming) 방식 업로드를 사용하지만, 아두이노에서는 부트로더를 통한 시리얼 방식 업로드를 기본으로 하며 이를 위해 전용 마이크로컨트롤러를 보드에 내장하고 있다. 아두이노에서 ISP 방식 업로드를 사용하기 위해서는 별도의 전용 업로드 장치가 필요하지만, 아두이노 보드에는 포함되어 있지 않다. 이 논문에서는 ISP 방식 업로드와 시리얼 방식 업로드를 하나의 전용 마이크로컨트롤러를 통해 처리할 수 있는 통합 프로그래머를 제안하고, 통합 프로그래머를 사용하여 ISP 방식과 시리얼 방식 업로드가 가능함을 보인다. 이외에도 제안하는 통합 프로그래머는 USB-시리얼 변환장치로 동작하여 컴퓨터와의 시리얼 통신을 지원하며 아두이노의 부트로더를 굽는 것도 가능하다. 통합 프로그래머의 모든 동작은 실험 결과를 통해 확인할 수 있다.

ABSTRACT

Arduino is one of the open source microcontroller projects, and Arduino boards using AVR microcontrollers are commonly used. In general, AVR microcontrollers use ISP (In System Programming) upload, but in Arduino, serial upload through a bootloader is basically used, and a dedicated microcontroller is built into the board for this purpose. In order to use the ISP upload in Arduino, a dedicated upload device is required, which is not included in the Arduino board. In this paper, we propose a unified programmer that can handle ISP upload and serial upload through one dedicated microcontroller, and show that ISP and serial upload are possible using the unified programmer. In addition, the proposed unified programmer works as a USB-serial converter, supports serial communication with a computer, and it is also possible to burn the Arduino bootloader. All operations of the unified programmer can be confirmed through the experimental results.

키워드 : 아두이노, ISP 프로그래머, 시리얼 업로더, 통합 프로그래머, ATmega128

Keywords : Arduino, ISP programmer, Serial uploader, Unified programmer, ATmega128

Received 4 November 2020, **Revised** 11 November 2020, **Accepted** 15 November 2020

*** Corresponding Author** Gyeongyong Heo (E-mail:hgycap@deu.ac.kr, Tel:+82-51-890-1675)
Associate Professor, Department of Electronic Engineering, Dong-eui University, Busan, 47340 Korea

Open Access http://doi.org/10.6109/jkiice.2021.25.1.96 print ISSN: 2234-4772 online ISSN: 2288-4165

ⓒThis is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(http://creativecommons.org/li-censes/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright Ⓒ The Korea Institute of Information and Communication Engineering.

Ⅰ. 서 론

아두이노는 예술가와 디자이너들이 쉽고 간단하게 상호작용이 가능한 전자 장치를 만들 수 있도록 하자는 목적으로 시작된 마이크로컨트롤러 플랫폼의 하나다[1]. 2005년 발표된 이후 아두이노는 쉽고 간단한 사용법으로 다양한 분야의 자발적인 참여를 통해 독자적인 생태계 구축에 성공하였으며 오픈 소스 프로젝트 중 가장 영향력 있는 프로젝트의 하나로 자리 잡고 있다.

아두이노는 8비트의 AVR 시리즈 마이크로컨트롤러를 사용한 보드가 처음 출시되었고, 높은 연산 능력과 다양한 무선 연결의 필요성이 증가함에 따라 32비트의 ARM Cortex-M 시리즈 마이크로컨트롤러를 사용한 보드도 출시되었다[1]. 그 중 8비트 AVR 시리즈 마이크로컨트롤러를 사용한 아두이노 보드는 쉽고 간단한 보드라는 아두이노의 철학과 일치하는 보드로 초중고등학교는 물론 대학에서도 하드웨어와 소프트웨어를 위한 기초 교육 도구로 널리 사용되고 있다[2-4]. 이 논문에서 대상으로 하는 보드 역시 8비트 AVR 시리즈 마이크로컨트롤러를 사용하는 보드다.

아두이노를 포함하여 마이크로컨트롤러 보드는 교차 개발 환경에서 마이크로컨트롤러를 위한 프로그램 또는 펌웨어(firmware)를 개발한다. 교차 개발을 위해서는 개발 시스템에서 동작하면서 목적 시스템을 위한 기계어 파일을 생성할 수 있는 교차 컴파일러가 필요하다. 또한, 만들어진 기계어 파일을 개발 시스템에서 목적 시스템으로 옮겨 설치하기 위해 프로그래머(programmer) 또는 업로더(uploader)를 사용한다. AVR 시리즈 마이크로컨트롤러를 사용하는 8비트 아두이노 보드는 교차 컴파일러로 자유 소프트웨어의 하나인 GCC(GNU Compiler Collection)를 사용한다[5].

AVR 시리즈 마이크로컨트롤러를 위한 프로그래머는 전용 하드웨어 장치가 사용된다. 디버깅까지 가능한 JTAG(Joint Test Action Group) 역시 사용할 수 있지만, SPI(Serial Peripheral Interface)를 사용하는 ISP(In System Programming) 방식이 주로 사용된다. 하지만 아두이노에서는 UART(Universal Asynchronous Receiver/Transmitter) 시리얼 인터페이스와 부트로더(bootloader)를 사용한 시리얼 방식을 기본으로 하고 있다. UART 시리얼 인터페이스를 사용하면 컴퓨터와의 시리얼 통신을 위해 대부분의 마이크로컨트롤러 보드에 포함되어 있는 하드웨어를 사용할 수 있어 추가 하드웨어가 필요하지 않은 것은 장점이다. 하지만, 마이크로컨트롤러의 프로그래밍 메모리 일부가 부트로더를 위해 사용되고 초기 구동 속도가 느린 점 등은 단점이라 할 수 있다. 또한 시리얼 방식 업로더로는 부트로더를 구울 수 없는 점 등 시리얼 방식은 ISP 방식을 완전히 대체할 수 없으므로 아두이노 환경에서도 ISP 방식 프로그래머를 완전히 배재할 수는 없다.

대부분의 아두이노 보드에는 시리얼 방식 업로드를 위한 전용 칩이 내장되어 있지만, ISP 방식을 지원하기 위한 하드웨어는 포함하고 있지 않다. 따라서 ISP 방식 업로드를 위해 별도의 전용 장치를 사용해야 하며 시리얼 방식과 ISP 방식 전환을 위해 설정을 변경해야 하는 불편함이 있다.

이 논문에서는 서로 다른 통신 인터페이스를 사용하는 시리얼 방식과 ISP 방식 업로드 기능을 하나의 전용 칩과 펌웨어만을 사용하여 구현한 통합 프로그래머(unified programmer)를 제안한다. 제안하는 통합 프로그래머는 컴퓨터와 USB 인터페이스를 통해 연결되며 프로그램을 업로드 할 때 업로드 방식에 따라 다르게 보내지는 초기 데이터에 따라 마이크로컨트롤러의 UART 또는 SPI 인터페이스로 연결하고 프로그램을 업로드 하는 기능을 기본으로 한다. 또한, 프로그램 업로드에 사용되지 않는 경우 통합 프로그래머는 USB와 UART 인터페이스 변환 기능을 수행할 수 있으므로 추가 하드웨어 없이 컴퓨터와 마이크로컨트롤러 사이의 시리얼 통신 역시 가능하게 해준다. 이처럼 제안하는 통합 프로그래머는 AVR 시리즈 마이크로컨트롤러에서 흔히 사용되는 업로드 방식을 모두 지원하면서 USB-UART 변환 기능까지도 지원하며 이 모든 기능을 하나의 칩으로 구현하므로 각 기능을 사용하기 위해 서로 다른 케이블을 연결할 필요가 없으므로 사용이 간편하고, 별도의 전용 장치가 필요하지 않으므로 가격면에서도 아두이노를 포함하여 다른 보드와 비교했을 때 장점이 될 것이다.

이 논문의 구성은 다음과 같다. 먼저 2장에서는 AVR 시리즈 마이크로컨트롤러에서 사용되는 업로드 방식들을 통해 통합 프로그래머의 필요성을 보인다. 3장에서는 제안하는 통합 프로그래머의 하드웨어 및 펌웨어 기능에 대해 설명하고, 4장에서는 ATmega128 마이크로컨트롤러와 통합 프로그래머를 사용하여 구현한 보드가 아두이노 IDE(Integrated Development Environment)

에서 정상적으로 동작하는 것을 보인다. 결론 및 향후 연구 방향은 5장에서 언급한다.

Ⅱ. AVR을 위한 업로드 장치

마이크로컨트롤러를 위한 프로그램은 흔히 펌웨어라고 한다. 펌웨어는 프로그래밍 언어를 사용하여 작성한다는 점에서 소프트웨어에 속하지만, 전통적으로 하드웨어를 사용하여 구현하는 기능을 대신한다는 점, 그리고 설치된 펌웨어의 교체가 빈번하지 않다는 점에서 하드웨어의 특징 역시 가지고 있다. 개발 시스템에서 만들어진 프로그램은 마이크로컨트롤러의 프로그램 메모리에 옮겨 설치한 후 실행될 수 있으며 이 과정에서 프로그래머 또는 업로더가 필요하다.

AVR 시리즈 마이크로컨트롤러에서 전통적으로 사용하는 업로드 방식은 ISP 방식으로 SPI 인터페이스를 통해 프로그램 메모리에 직접 프로그램을 설치하는 방식이다. 반면 아두이노에서는 UART 인터페이스를 사용하는 시리얼 방식 업로드를 기본으로 한다. 시리얼 방식 업로드를 위해서는 전용 하드웨어 이외에 부트로더가 필요하다. 시리얼 방식에서 UART 인터페이스를 통해 전달된 프로그램은 부트로더를 거쳐 프로그램 메모리에 설치된다.

시리얼 방식 업로드는 별도의 부트로더가 필요하고, 부트로더 설치를 위해 프로그램 메모리의 일부를 소비하며, 마이크로컨트롤러 부팅 과정에서 부트로더가 먼저 실행되어 구동 지연이 발생하는 등의 단점이 있다. 하지만 시리얼 방식 업로드에 사용되는 하드웨어는 개발 시스템의 USB 인터페이스와 마이크로컨트롤러의 UART 인터페이스 사이 상호 변환 기능을 기본으로 하며 아두이노 보드를 포함하여 많은 마이크로컨트롤러 보드가 USB-UART 변환 하드웨어를 포함하고 있으므로 추가 하드웨어 없이 펌웨어 추가만으로 프로그램 업로드에 사용할 수 있다는 점은 장점이 아닐 수 없다. 또한 프로그램 업로드에 사용하지 않을 때는 개발 시스템과의 시리얼 통신을 위해 사용할 수 있다. 프로그램 업로드와 시리얼 통신은 동시에 사용되지 않으므로 하나의 하드웨어를 두 가지 목적으로 사용할 수 있다. 그림 1은 ISP 방식과 시리얼 방식 업로드 과정을 비교하여 나타낸 것이며, 표 1은 두 가지 업로드 방식을 비교한 것이다.

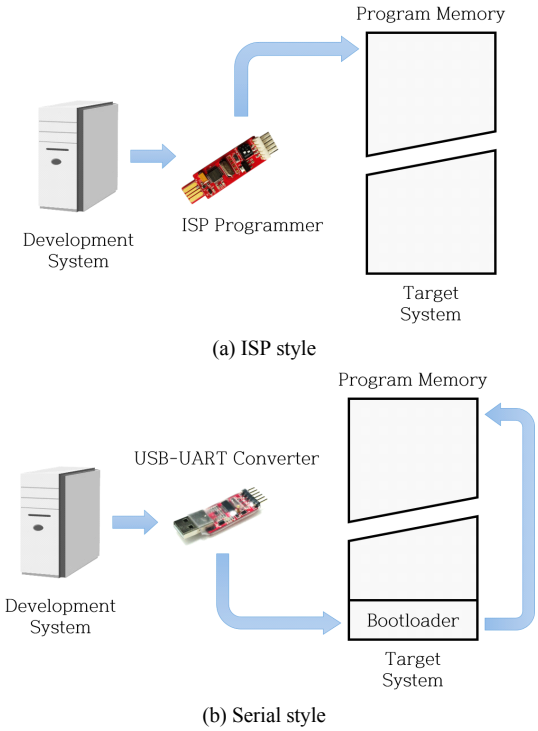


Fig. 1 Program upload process

Table. 1 Comparison of program upload styles

	ISP	Serial
Interface	SPI	UART
Hardware	ISP programmer	USB-UART converter
Using bootloader	×	○
Bootloader burning	○	×
Main purpose in Arduino	Bootloader burning	Program upload

부트로더를 굽기 위해서는 ISP 방식을 사용해야 하고, ISP 방식을 사용하면 프로그램 메모리 전부를 사용자 프로그램을 위한 공간으로 사용할 수 있으며, 프로그램이 손상되었을 때 한 종류의 프로그램 설치만으로 복구가 가능한 점 등은 ISP 방식의 장점이다. 따라서 아두이노에서 시리얼 방식을 기본으로 하지만, ISP 방식 역시 빼놓을 수 없다. 하지만 아두이노 및 호환 보드에 ISP 방식 프로그래머가 포함된 예를 찾아볼 수 없는 것은 낮은 활용도와 하드웨어 추가에 따른 부담 때문이다. 따라서 이 논문에서는 하드웨어 추가 없이 시리얼 방식과

ISP 방식을 모두 사용할 수 있는 통합 프로그래머를 제안한다.

Ⅲ. 통합 프로그래머의 기능

이 논문에서 제안하는 통합 프로그래머는 AVR 시리즈 마이크로컨트롤러 중 USB 기능이 없는 마이크로컨트롤러를 사용한 아두이노 보드를 대상으로 한다. 대상 보드는 시리얼 방식과 ISP 방식 업로드를 위해 별도의 하드웨어를 사용하고 있으므로 이를 하나의 전용 칩과 펌웨어를 통해 사용할 수 있도록 하는 것이 통합 프로그래머의 목표다. 아두이노의 시리얼 방식 업로드에 사용되는 전용 장치가 개발 시스템과의 시리얼 통신에도 사용될 수 있는 것은 통합 프로그래머 역시 마찬가지다. 즉, 통합 프로그래머는 ISP 방식 업로드, 시리얼 방식 업로드 그리고 시리얼 통신의 3가지 기능을 하나의 전용 장치를 통해 구현하는 것을 목표로 한다. 부트로더를 굽는 기능은 ISP 방식 업로드 기능에 포함되므로 별도의 기능으로 정의하지 않았다.

Table. 2 Comparison of the functions in the unified programmer

	ISP upload	Serial upload	Serial comm.
Target System Interface	USB		
uC Interface	SPI	UART	UART
Using RESET	○		×
Additional function	Bootloader burning	-	-

표 2는 통합 프로그래머에서 지원하는 3가지 기능을 비교한 것이다. 2가지 업로드 방법은 리셋 신호를 사용한다는 점에서 시리얼 통신 기능과 구별된다. 또한 각각의 업로드 방법은 사용하는 인터페이스의 차이에 따라 개발 시스템에서 보내는 초기 정보에서 차이가 있으므로 구별이 가능하다.

그림 2는 통합 프로그래머에 사용된 펌웨어에서 개발 시스템의 신호에 따라 마이크로컨트롤러의 각 인터페이스로 연결하고 데이터를 전달하는 흐름도를 나타낸 것이다. 아두이노에서 개발 시스템으로 데이터를 보내는 경우는 UART 인터페이스를 통한 시리얼 통신이 유일하므로 별도로 나타내지 않았다.

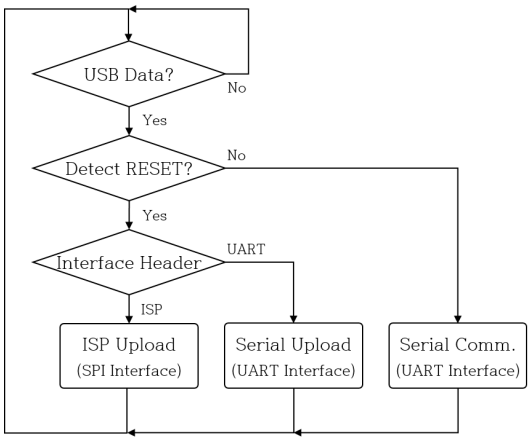
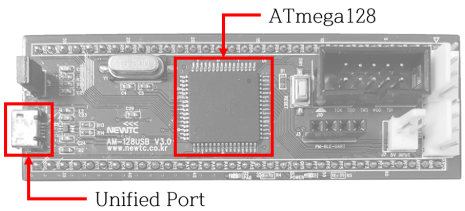


Fig. 2 Function discrimination process

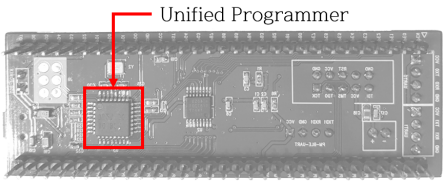
Ⅳ. 보드 구현 및 실험 결과

제안하는 통합 프로그래머는 르네사스(Renesas) 사의 마이크로컨트롤러를 사용하여 구현하였다. 마이크로컨트롤러의 펌웨어에서 그림 2의 절차에 따라 자동으로 기능을 구별하여 처리하므로 별도의 기능 전환을 위한 부품은 필요하지 않으며 대상 마이크로컨트롤러 보드와 개발 시스템은 USB 케이블만 연결하면 된다.

그림 3은 구현한 통합 프로그래머를 ATmega128 마이크로컨트롤러를 사용한 보드에 적용한 예로 프로그램 업로드와 시리얼 통신을 위해 통합 포트(unified port)를 가지고 있다. ATmega128은 공식 아두이노 보드에 사용되는 마이크로컨트롤러는 아니지만 MegaCore 프로젝트[6]를 통해 아두이노 호환 보드로 사용할 수 있다[7]. 보드 매니저를 통해 ATmega128을 위한 MegaCore 파일을 설치하고, 통합 프로그래머를 stk500v2 프로토콜을 사용하는 ISP 방식 업로더로 아두이노에 표 3과 같이 등록하면 통합 프로그래머를 사용할 준비는 끝난다.



(a) Front



(b) Back

Fig. 3 ATmega128 board using the unified programmer

Table. 3 Programmer information for the unified programmer

UP.name=Unified Programmer UP.communication=serial UP.protocol=stk500v2 UP.program.protocol=stk500v2 UP.program.tool=avrdude UP.program.extra_params=-P{serial.port}

통합 프로그래머를 사용한 보드를 컴퓨터에 연결하면 일반(generic) USB-시리얼 변환장치로 인식되어 시리얼 포트로 나타난다. 인식된 시리얼 포트는 시리얼 방식 업로드, ISP 방식 업로드, 시리얼 통신 등 세 가지 동작에서 모두 사용하는 포트다. 기존 아두이노 환경에서는 원하는 동작에 따라 포트를 포함하여 설정의 변경이 필요했던 점과 비교하면 통합 프로그래머를 사용할 때는 기능에 따라 설정을 변경할 필요가 없다는 점이 장점이다. 통합 프로그래머를 사용한 아두이노 호환 ATmega128 보드의 아두이노 환경 설정은 표 4와 같다.

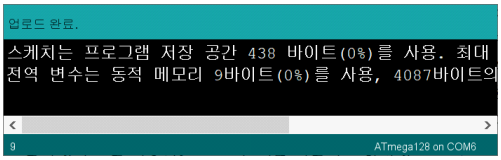
Table. 4 Arduino settings for the ATmega128-based board with the unified programmer

Menu	Select	Explanation
Tool→Board	ATmega128	from MegaCore
Tool→Port	COM* (Unified port)	computer dependent
Tool→ Programmer	Unified Programmer	from Table 3

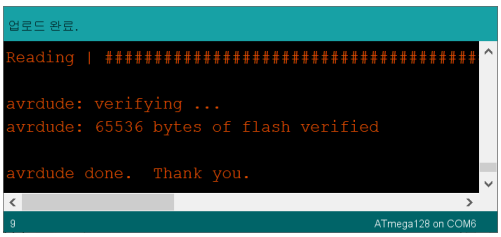
이후 사용 방법은 아두이노 보드의 사용과 차이가 없다. 그림 4는 부트로 굽기와 프로그램 업로드 결과를 나타낸 것으로, 각각의 과정에서 사용하는 명령은 아두이노 보드의 경우와 같다.

통합 프로그래머는 저렴하고 간단하면서도 다양한 기능을 사용할 수 있는 보드를 제작할 수 있도록 해준다는 점이 장점 중 하나다. 통합 프로그래머는 아두이노 환경은 물론 전통적인 AVR 환경에서도 사용할 수 있으므로 아두이노 호환을 넘어 이종 플랫폼 간의 융합 및 한국형 오픈 소스 플랫폼으로 아두이노의 대안이 될 수 있을 것이다[8-9].

(a) Bootloader burning



(b) Serial upload



(c) ISP upload

Fig. 4 Bootloader burning and program upload results

Ⅴ. 결 론

AVR 시리즈 마이크로컨트롤러를 사용하는 아두이노에서는 시리얼 방식과 ISP 방식의 업로드가 흔히 사용된다. 아두이노에서는 시리얼 방식을 기본으로 하지만 ISP 방식으로만 가능한 작업이 있어 두 가지 방법이 모두 사용되고 있다. 따라서 아두이노에서는 2개의 전용 업로드 장치가 필요하며, 서로 다른 설정으로 사용해야 하는 불편함이 있다.

이 논문에서는 두 가지 업로드 방식을 하나의 전용 마이크로컨트롤러와 펌웨어를 통해 구현하는 방법을 제안하고 이를 ATmega128을 사용한 아두이노 호환 보드로 구현하여 통합 프로그래머의 동작을 확인하였다. 통합 프로그래머를 사용함으로써 하나의 통합 포트를 통해 설정이나 케이블 연결의 변경 없이 필요한 모든 작업

을 수행할 수 있어 사용이 편리하고, 별도의 외부 장치를 필요로 하지 않으므로 AVR 및 아두이노 환경 구축을 위한 비용 역시 절감할 수 있다는 점이 장점이라 할 수 있다. 향후 아두이노 환경을 중심으로 통합 프로그래머의 호환성 검사를 진행할 예정이며, 아두이노에 사용된 AVR 시리즈 마이크로컨트롤러에 통합 프로그래머를 적용한 보드 제작 및 검증을 계획하고 있다. 또한, 프로그래머가 포함되어 있지 않은 소형 보드를 위한 외부 프로그래머의 개발도 고려하고 있다.

ACKNOWLEDGEMENT

This Work was supported by Dong-eui University Foundation Grant (202003600001).

REFERENCES

[1] Arduino [Internet]. Available: https://www.arduino.cc/.

[2] M. Novak, J. Kalova, and J. Pech, “Use of the Arduino Platform in Teaching Programming,” in *Proceedings of 2018 IV International Conference on Information Technologies in Engineering Education*, Moscow, Russia, pp. 1-4, 2018.

[3] A. Bashir, M. Alhammadi, M. Moath Awawdeh, and T. Faisal, “Effectiveness of using Arduino platform for the hybrid engineering education learning model,” in *Proceedings of 2019 Advances in Science and Engineering Technology International Conferences*, Dubai, United Arab Emirates, pp. 1-6, 2019.

[4] G. Heo, “Implementation of an Arduino Compatible Modular Kit for Educational Purpose,” *Journal of the Korea Institute of Information and Communication Engineering*, vol. 23, no. 5, pp. 547-554, May. 2019.

[5] GCC, the GNU Compiler Collection [Internet]. Available: https://gcc.gnu.org/.

[6] MegaCore [Internet]. Available: https://github.com/MCUdude/MegaCore/.

[7] G. Heo, “128duino : An Extension of the Arduino Platform for ATmega128,” *Journal of the Korea Institute of Information and Communication Engineering*, vol. 24, no. 10, pp. 1396-1375, Oct. 2020.

[8] E. F. Moreno and J. Pastor-Mendoza, “Synergy LEGO Mindstorms – Arduino: Taking advantage of both platforms,” in *Proceedings of Technologies Applied to Electronics Teaching Conference*, La Laguna, Spain, pp. 1-6, 2018.

[9] J. Pech and M. Novak, “Use Arduino and Micro:bit as Teaching Platform for the Education Programming and Electronics on the STEM Basis,” in *Proceedings of International Conference on Information Technologies in Engineering Education*, Moscow, Russia, pp. 1-4, 2020.

허경용(Gyeongyong Heo)

연세대학교 전자공학과 공학석사 (1996)
University of Florida 컴퓨터공학과 공학박사 (2009)
동의대학교 전자공학과 교수 (2012~현재)
※관심분야 : 인공지능, 패턴인식, IoT 시스템

류대우(Daewoo Ryu)

한양대학교 전자통신전파공학과 공학석사 (2000)
LG전자 연구소 (1997~2003)
㈜뉴티씨 대표이사 (2005~현재)
※관심분야 : 노이즈 최소화 설계, 계측제어, IoT시스템, 인공지능, 마이크로프로세서