

## 128duino : ATmega128을 위한 아두이노 플랫폼의 확장

최훈<sup>1</sup> · 허경용<sup>2\*</sup>

### 128duino : An Extension of the Arduino Platform for ATmega128

Hun Choi<sup>1</sup> · Gyeongyong Heo<sup>2\*</sup>

<sup>1</sup>Professor, Department of Electronic Engineering, Dong-Eui University, Busan, 47340 Korea

<sup>2\*</sup>Associate Professor, Department of Electronic Engineering, Dong-Eui University, Busan, 47340 Korea

#### 요 약

아두이노는 비전공자를 위한 마이크로컨트롤러 플랫폼의 하나로 학습용으로 널리 사용되고 있다. 아두이노 보드에는 여러 종류의 AVR 시리즈 마이크로컨트롤러가 사용되지만, ATmega128은 사용하지 않는다. ATmega128은 아두이노 보드에 사용되는 ATmega328이나 ATmega2560과 비교했을 때 확장성이 우수하고 가격 경쟁력이 높아 지금도 많이 사용되고 있으므로, ATmega128을 아두이노 환경에서 사용할 수 있도록 함으로써 기존 하드웨어의 활용성을 높일 수 있음은 물론 아두이노 플랫폼 역시 활용할 수 있는 장점이 있다. 이 논문에서는 ATmega128과 아두이노 플랫폼의 장점을 결합할 수 있도록 ATmega128을 기반으로 하는 아두이노 호환 보드 설계와 이를 활용하는 방법을 제시한다. 이러한 아두이노 플랫폼의 확장은 기존 하드웨어를 활용하면서 아두이노 플랫폼의 장점을 사용할 수 있어 다양한 마이크로컨트롤러 관련 교육에서 학습 효과를 높일 수 있을 것으로 기대된다.

#### ABSTRACT

Arduino is a microcontroller platform for non-IT major students, and is widely used as a learning tool. Several AVR series microcontrollers are used in Arduino boards, but ATmega128 is not used. ATmega128 is widely used because of its high expandability and competitive price compared to ATmega328 and ATmega2560 used in Arduino boards. Therefore, by allowing ATmega128 to be used in an Arduino environment, the usability of existing hardware and the Arduino platform can be improved. In this paper, proposed are an Arduino-compatible board design based on ATmega128 and ways to use the ATmega128-based board. As the strengths of the Arduino platform can be used while utilizing existing hardware in the proposed extension, it is expected that the proposed one can be used in various microcontroller-related education and enhance the learning efficiency.

**키워드** : ATmega128, 아두이노, 마이크로컨트롤러, 확장 플랫폼

**Keywords** : ATmega128, Arduino, Microcontroller, Extensible platform

Received 22 July 2020, Revised 23 July 2020, Accepted 3 August 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.2020.24.10.1369>

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/licenses/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

아두이노(Arduino)는 이탈리아의 이브레아에서 예술가와 디자이너가 쉽고 간단하게 주변 환경과 상호작용할 수 있는 전자 장치를 만들 수 있도록 하자는 목적으로 시작된 마이크로컨트롤러 플랫폼이다[1]. 2005년 처음 발표된 이후 아두이노는 쉽고 간단한 사용법으로 수많은 참여자를 끌어들이며 독자적인 생태계를 구축함으로써 오픈 소스 프로젝트 중 가장 많은 참여자를 가진 마이크로컨트롤러 관련 프로젝트로 자리잡고 있다.

아두이노 플랫폼은 크게 마이크로컨트롤러를 사용하여 만들어진 하드웨어인 아두이노 보드와 아두이노 보드를 위한 프로그램(또는 펌웨어)을 작성하는 소프트웨어 개발환경으로 구성된다. 아두이노 보드에는 다양한 마이크로컨트롤러가 사용된다. 애초 아두이노 보드는 비전공자들이 쉽게 사용할 수 있도록 8비트 AVR 시리즈 마이크로컨트롤러를 사용하여 만들어졌지만, 고성능 마이크로컨트롤러에 대한 수요의 증가에 따라 32비트의 ARM 시리즈 마이크로컨트롤러를 사용하는 아두이노 보드 역시 출시되고 있다. 하지만 주로 사용되는 아두이노 보드는 AVR 시리즈 마이크로컨트롤러를 사용하여 만들어진 보드로 ATmega328을 사용한 아두이노 우노(UNO)와 ATmega2560을 사용한 아두이노 메가2560이 대표적이다. 아두이노는 초중고등학교는 물론 대학에서도 프로그래밍 교육, 피지컬 컴퓨팅, 사물인터넷 등의 다양한 주제를 위한 학습 도구로 널리 사용되고 있으며[2-3] 아두이노를 사용한 교육의 효율성과 유용성이 여러 사례 연구를 통해 입증되었다[4-5]. 이에 따라 아두이노 플랫폼을 확장하여 적용 분야를 넓히려는 시도 역시 이루어지고 있다[6-7].

아두이노 이전에 이들 주제를 위해 흔히 사용되었던 마이크로컨트롤러는 역시 AVR 시리즈 마이크로컨트롤러의 하나인 ATmega128로 많은 교육 기관에서 지금도 ATmega128을 사용한 교육이 이루어지고 있다. ATmega128을 사용한 교육에서는 아두이노 환경에서는 다루기 어려운 하드웨어 수준에서의 마이크로컨트롤러 교육(이하 ‘하드웨어 교육’)이 주로 이루어진다면, 아두이노를 사용한 교육에서는 추상화된 하드웨어를 사용하여 알고리즘 개발 및 시스템 개발 관련 교육(이하 ‘소프트웨어 교육’)이 주로 이루어지고 있다. 이처럼 기존의 마이크로컨트롤러 개발 환경과 아두이노 환경은

사용하는 목적에 차이가 있으므로 함께 사용하는 것이 바람직하지만, 사용하는 마이크로컨트롤러의 차이로 인해 서로 다른 하드웨어를 준비해야 하고 연관된 교육에서 서로 다른 마이크로컨트롤러를 사용함에 따라 학습 효율의 저하가 발생할 수 있다. 아두이노 보드에 사용된 ATmega328/2560과 비교했을 때 ATmega128은 기본적인 기능과 성능 면에서 동일하다. 또한 기존에 많은 교육 기관에서 ATmega128 기반의 하드웨어를 구비하고 있으므로 ATmega128을 아두이노 환경에서 사용할 수 있다면 하나의 마이크로컨트롤러를 사용하여 하드웨어와 소프트웨어 교육이 모두 이루어질 수 있어 여러 가지 하드웨어가 필요하고 학습의 효율이 저하되는 문제를 줄일 수 있다.

이 논문에서는 ATmega128을 아두이노 환경에서 사용할 수 있도록 아두이노 플랫폼을 확장하는 방법을 제시한다. ATmega128을 위해 아두이노 플랫폼을 확장하기 위해서는 기존 아두이노 보드와 호환될 수 있도록 새로운 아두이노 보드의 디자인이 필요하며 이 논문에서는 ATmega128을 사용한 아두이노 호환 보드인 128duino 보드 디자인을 제안한다. 128duino 보드 이외에도 ATmega128 마이크로컨트롤러를 아두이노 플랫폼에서 사용하기 위해서는 아두이노 함수와 라이브러리의 수정이 필요하지만, 이 논문에서는 MegaCore[8]를 사용한다. MegaCore는 공식 아두이노 보드에 사용되지 않은 AVR 시리즈 마이크로컨트롤러를 아두이노 플랫폼에서 사용할 수 있도록 해주는 오픈 소스 프로젝트의 하나다. 제안하는 128duino 보드와 MegaCore를 사용하면 교육용으로 효율성이 입증된 ATmega128을 사용하여 하드웨어 교육 및 소프트웨어 교육이 함께 이루어질 수 있으므로 비용 절감은 물론 학습 효과를 높일 수 있다.

이 논문의 구성은 다음과 같다. 먼저 2장에서는 ATmega128을 아두이노 보드에 사용된 ATmega328/2560과 비교하고 기존 아두이노 보드와 호환되는 128duino 보드 디자인을 제안한다. 3장에서는 제안하는 128duino 보드를 위한 프로그램을 작성하는 방법을 설명한다. 하드웨어 교육은 레지스터를 사용하는 방법, 소프트웨어 교육은 추상화된 아두이노 함수와 라이브러리를 사용하는 방법으로 이들 모두가 아두이노 프로그램을 통해 가능하다. 4장에서는 제안하는 128duino의 활용 방안을 설명하며, 결론 및 향후 연구 방향은 5장에서 언급한다.

## II. 128duino 보드 디자인

아두이노 보드에서의 핵심은 사용된 마이크로컨트롤러라고 할 수 있다. 다양한 아두이노 보드가 판매되고 있지만, 많이 사용되는 보드에는 첫 번째 정식 보드이면서 ATmega328 마이크로컨트롤러를 사용한 아두이노 우노와 아두이노 우노의 확장성을 보완하기 위해 ATmega2560 마이크로컨트롤러를 사용한 아두이노 메가2560이 있다. 아두이노 우노에 사용된 ATmega328은 20개의 범용 입출력(GPIO : General Purpose Input Output) 핀을 사용할 수 있지만 20개의 GPIO 핀으로 여러 주변장치를 연결하기는 쉽지 않다. 따라서 아두이노 메가2560은 아두이노 우노의 확장성을 보완하여 ATmega2560의 86개 GPIO 핀 중 70개 핀을 사용할 수 있도록 해준다. 아두이노 메가2560은 많은 수의 GPIO 핀을 사용할 수 있지만 사용할 수 있는 모든 핀을 사용하지 않으며, 70개 핀은 그 수가 많아 아두이노 기반 프로젝트에서 70개 핀을 모두 사용하는 경우는 거의 없다.

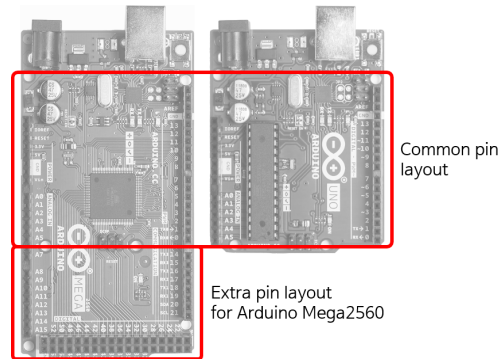
**Table. 1** Comparison of microcontrollers for Arduino boards

	ATmega328	ATmega128	ATmega2560
# of uC pins	28	64	100
Clock(MHz)	16 (External Clock)		
Operational voltage(V)	5		
Program memory(KB)	32	128	256
SRAM(KB)	2	4	8
EEPROM(KB)	1	4	4
# of data pins	20/20	53/53	70/86
# of ADC channels	6	8	16
# of PWM channels	6	7	15
# of UART ports	1	2	4

표 1은 ATmega328/128/2560 마이크로컨트롤러를 비교한 것이다. 표 1에서 알 수 있듯이 ATmega328/128/2560은 기본적인 성능과 기능에서는 차이가 없지만, GPIO 핀 수의 차이에 의해 사용할 수 있는 핀 수와 통신 포트 수에서 차이가 있다. 또한, 프로그램을 위한 메모리의 크기도 차이가 있다. 아두이노 우노의 20개 핀은 충분하지 않지만, 아두이노 메가2560의 70개 핀은 모두 활용하는 경우가 드물어 ATmega128의 53개

GPIO 핀을 아두이노 환경에서 모두 사용할 수 있다면 아두이노 메가2560에 비해 가격 경쟁력이 있고 아두이노 우노의 낮은 확장성을 보완할 수 있어 기존 아두이노 보드의 대안으로 사용할 수 있다. 또한, 기존에 보급된 보드를 활용할 수 있다는 점도 장점이다.

아두이노 보드의 특징 중 하나는 GPIO 핀 배치가 표준화되어 있다는 점이다. 그림 1은 아두이노 우노와 아두이노 메가2560 보드를 비교한 것으로, 사용할 수 있는 핀 수의 차이는 있지만, 아두이노 우노의 20개 핀과 아두이노 메가2560의 처음 20개 핀은 같은 위치에 배치되어 있음을 알 수 있다. 이러한 표준화된 핀 배치는 여러 아두이노 보드에서 공통으로 사용할 수 있는 확장 보드인 쉴드(shield)를 구현할 수 있도록 해준다.



**Fig. 1** Comparison of pin layouts between Arduino UNO and Arduino Mega2560

MegaCore에서는 ATmega128의 53개의 GPIO 핀을 모두 아두이노 환경에서 사용할 수 있도록 해준다. 128duino 보드 디자인에서 53개 GPIO 핀 중 처음 20개 핀 배치는 아두이노 우노와 호환되어야 하며, 나머지 33개 핀 역시 아두이노 메가2560과 호환성을 고려하여야 한다. 그림 2는 아두이노 우노 및 아두이노 메가2560과의 핀 호환성을 고려하여 이 논문에서 제안하는 128duino 보드의 디자인을 나타낸 것이다. 128duino 보드에서는 0번부터 52번까지 53개 핀 모두를 사용할 수 있으며, 아두이노 핀 번호는 MegaCore[8]의 핀 번호를 따랐다. 53개 핀 중 0번부터 13번까지의 디지털 데이터 입출력 핀과 A0번부터 A5번까지의 아날로그 출력 핀 등 20개 핀은 아두이노 우노 및 아두이노 메가2560의 핀 배치와 동일하게 배치하여 쉴드를 사용할 수 있도록 하였다. 나머지 33개 핀은 표준화되어 있지 않지만 아두이노 메가

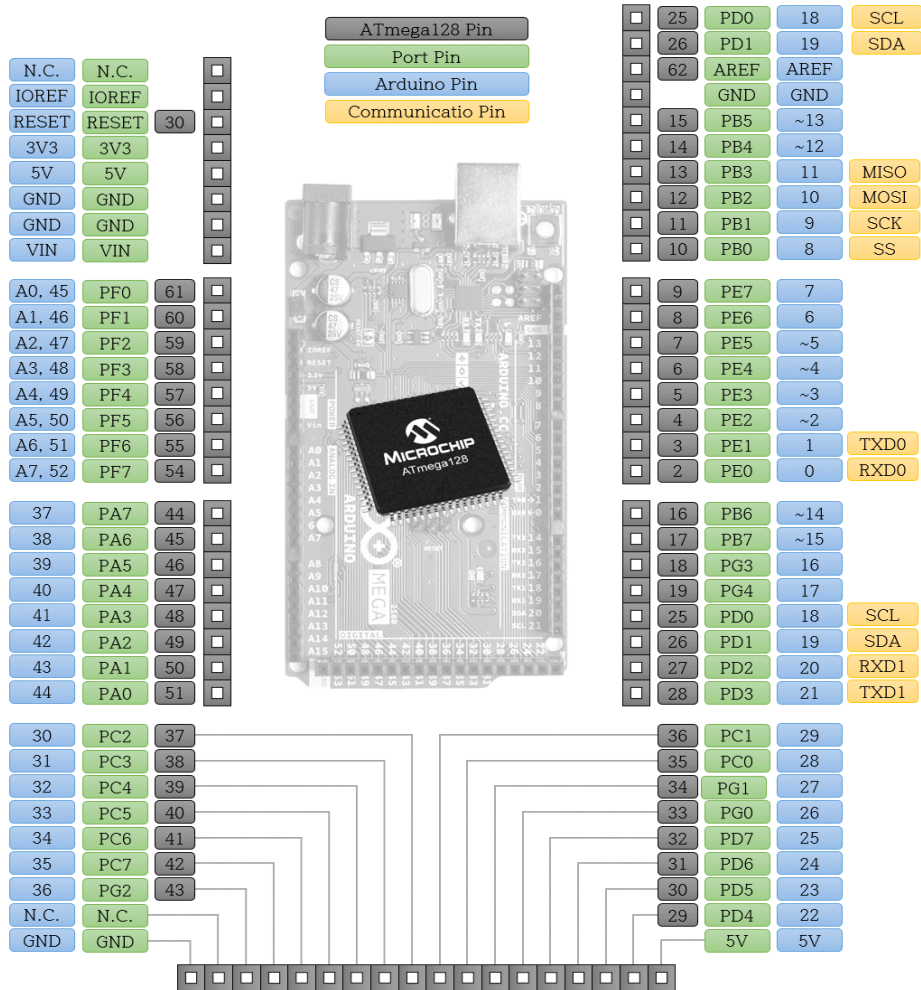


Fig. 2 128duino design

560의 핀 배치를 참고하여 배치하였다. 128duino 보드는 아래쪽의 2열 핀 헤더에서 한 열을 제거한 것을 제외에는 아두이노 메가2560과 같은 크기를 가진다.

### III. 128duino 프로그래밍

아두이로 보드를 위한 프로그램은 C 언어를 통해 레지스터를 조작하는 ‘AVR 스타일’과 C/C++ 언어를 통해 아두이노의 함수와 라이브러리를 사용하는 ‘아두이노 스타일’의 두 가지를 사용할 수 있으며 128duino 보드 역시 마찬가지다. AVR 스타일이 마이크로컨트롤러

의 구조를 포함한 하드웨어 중심의 교육에서 주로 사용되는 전통적으로 프로그래밍 방식이라면, 아두이노 스타일은 하드웨어 관련 내용을 추상화하여 기초 교육이나 시스템 구현에서 주로 사용되는 소프트웨어 중심의 프로그래밍 방식이라는 차이가 있다. AVR 스타일 프로그래밍을 위해서는 마이크로칩에서 제공하는 Atmel Studio와 같은 IDE가 주로 사용된다면, 아두이노 스타일 프로그래밍에서는 아두이노 프로그램이 주로 사용된다는 점도 차이점이다. 하지만 아두이노 스타일에서 사용하는 아두이노 함수와 라이브러리는 AVR 스타일 코드를 추상화한 것으로 동일한 틀체인을 사용하여 컴파일 할 수 있으므로 아두이노 프로그램에서도 AVR 스

타일 코드를 컴파일 할 수 있다. 따라서 128duino 보드를 위한 코드는 AVR 스타일과 아두이노 스타일 모두 아두이노 프로그램을 기본으로 한다. 그림 3은 128duino 보드를 위한 코드의 계층적 구조를 나타낸 것으로 아두이노 프로그램의 WinAVR 툴체인을 통해 AVR 스타일과 아두이노 스타일 코드를 작성할 수 있다.

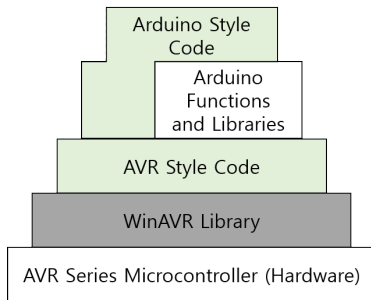


Fig. 3 Hierarchical structure of code components for 128duino

```
#define F_CPU 16000000L
#include <avr/io.h>
#include <util/delay.h>

int main(void) {
    DDRB = 0x20; // set pin mode as output

    while(1) {
        PORTB = 0x20; // turn on LED
        _delay_ms(1000);
        PORTB = 0x00; // turn off LED
        _delay_ms(1000);
    }

    return 1;
}
```

Fig. 4 Blink code - AVR style

그림 4는 128duino를 위한 AVR 스타일 코드의 예로 포트 B의 5번 핀(PB5)에 연결된 LED를 1초 간격으로 점멸하는 코드다. 그림 5는 그림 4의 코드와 같은 기능을 하는 아두이노 스타일 코드의 예다. MegaCore에서 ATmega128의 PB5 핀은 아두이노 13번 핀으로 정의되어 있다. 아두이노 13번 핀의 LED 점멸은 아두이노의 기본 예제 중 하나로 그림 5의 코드는 아두이노 우노나 아두이노 메가2560에서 사용하는 코드와 같다.

```
void setup() {
    pinMode(13, OUTPUT); // set pin mode
}

void loop() {
    digitalWrite(13, HIGH); // turn on LED
    delay(1000);
    digitalWrite(13, LOW); // turn off LED
    delay(1000);
}
```

Fig. 5 Blink code - Arduino style

#### IV. 128duino의 활용

128duino의 가장 큰 장점은 동일한 하드웨어를 다양한 수준과 다양한 주제의 교육에 사용할 수 있다는 점이다.

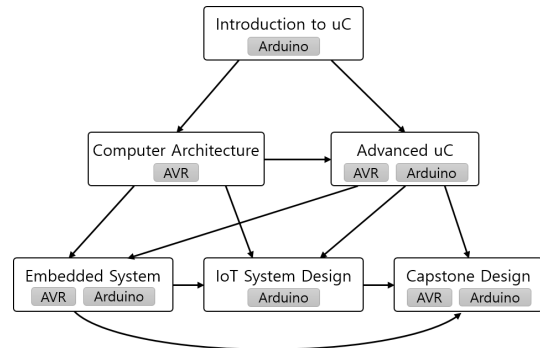


Fig. 6 Curriculum structure using 128duino

아두이노 스타일 프로그래밍은 마이크로컨트롤러 기초 교육과 알고리즘 위주의 시스템 구현에 주로 사용되고 있으며, AVR 스타일 프로그래밍은 마이크로컨트롤러 구조 및 하드웨어 디자인에서 주로 사용된다. 이처럼 다양한 주제와 다양한 수준의 교육을 하나의 하드웨어를 통해 다룰 수 있다는 것은 일관성을 유지하여 학습 효율을 높일 수 있음을 의미한다. 그림 6은 마이크로컨트롤러를 사용할 수 있는 교과목들의 흐름과 연관성을 나타낸 예로 모든 교과목에서 128duino를 사용할 수 있으며 이것이 제안하는 128duino 보드의 목적이자 가장 큰 장점이라 할 수 있다.

128duino를 사용하기 위해서는 아두이노와 핀 배치 호환성이 있는 그림 2의 보드를 사용하는 것이 좋지만,

ATmega128 기반의 모든 보드를 사용할 수 있다는 점 역시 128duino의 장점이라 할 수 있다. 특히 아두이노 이전에 ATmega128 기반의 하드웨어를 사용했던 상황에서는 ATmega128에서 아두이노로 이어지는 고리를 만들 수 있다는 점을 무시할 수 없다. 한국의 상황에 맞는 아두이노 기반 교육 도구의 개발이 시도되고 있지만 기존 아두이노 보드로는 어려움이 있는 상황에서 128duino는 그 대안이 될 수 있을 것이다[9-10].

## V. 결 론

최근 아두이노의 보급에 따라 다양한 교육 현장에서 아두이노를 교육 도구로 사용하고 있다. 하지만 아두이노 환경을 사용하기 위해서는 이전에 보급된 ATmega128 기반 하드웨어가 아닌 다른 하드웨어가 필요하다는 점 그리고 주로 사용되는 아두이노 보드가 다양한 주제의 교육용으로 사용되기에는 적합하지 않다는 점 등의 문제가 있다. 이를 해결하기 위한 방법의 하나가 ATmega128을 아두이노 환경에서 사용할 수 있도록 아두이노 플랫폼을 확장하는 것이며, 이를 위해 이 논문에서는 아두이노 보드와 호환되는 ATmega128 기반의 128duino 보드 디자인을 제안하고 이를 활용하여 실제 교육에 사용할 수 있는 예를 제시하였다. ATmega128은 성능과 기능 면에서 아두이노 보드에 사용된 다른 마이크로컨트롤러를 대체할 수 있으며, 아두이노 이전에 널리 보급되어 있어 쉽게 찾아볼 수 있다는 점에서 이 논문에서 제안하는 128duino는 국내 상황에 적합한 선택이 될 수 있다. 현재 제안한 보드를 제작 중에 있으며, 제작된 보드를 실제 교육 현장에 적용하고 피드백에 따라 보드 업데이트에 반영할 계획이다.

## ACKNOWLEDGEMENT

This Work was supported by Dong-eui University Foundation Grant(2019).

## REFERENCES

- [ 1 ] Arduino [Internet]. Available: <https://www.arduino.cc/>.
- [ 2 ] P. Plaza, E. Sancristobal, G. Fernandez, M. Castro, and C. Perez, "Collaborative robotic educational tool based on programmable logic and Arduino," in *Proceedings of 2016 Technologies Applied to Electronics Teaching*, Seville, Spain, pp. 1-8, 2016.
- [ 3 ] 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.
- [ 4 ] J. C. Martinez-Santos, O. Acevedo-Patino, and S. H. Conteras-Ortiz, "Influence of Arduino on the Development of Advanced Microcontrollers Courses," *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 12, no. 4, pp. 208-217, Nov. 2017.
- [ 5 ] 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.
- [ 6 ] G. Y. Heo and J. W. Jung, "Arduino Compatible Modular Kit Design for Educational Purpose," *Journal of the Korea Institute of Information and Communication Engineering*, vol. 22, no. 10, pp. 1371-1378, Jan. 2018.
- [ 7 ] G. Y. 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.
- [ 8 ] MegaCore [Internet]. Available: <https://github.com/MCUdude/MegaCore/>.
- [ 9 ] A. Malizia, T. Turchi, and K. A. Olsen, "Block-oriented programming with tangibles: An engaging way to learn computational thinking skills," in *Proceedings of 2017 IEEE Blocks and Beyond Workshop*, Raleigh, NC, USA, 2017.
- [ 10 ] H. Y. Eom and K. H. Lee, "Design of Embodiment-based Programming Education using Arduino for Middle School Students," *The Journal of the Convergence on Culture Technology*, vol. 6, no. 1, pp. 471-476, Jan. 2020.



**최훈(Hun Choi)**

충북대학교 전자공학과 공학석사 (2001)  
충북대학교 전자공학과 공학박사 (2006)  
한국표준과학연구원 Post Doc(2006~2008)  
동의대학교 전자공학과 교수 (2008~현재)  
※관심분야 : 적응신호처리, 계측신호처리, 디지털신호처리 응용



**허경용(Gyeongyong Heo)**

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