



어린이를 위한 교육용 알고리즘 장난감 블록

Educational algorithm toy blocks for children

종합설계 설계서

2014150018 서준원 (지도교수 서대영)
2012151002 권혁준 (지도교수 한익주)
2012151029 오우석 (지도교수 한익주)

Know Answer.

차례

1. 종합설계 개요
2. 관련 연구 및 사례
3. 시스템 수행 시나리오
4. 시스템 구성도
5. 시스템 모듈 상세 설계
6. 개발 환경 및 개발 방법
7. 데모 환경 설계
8. 업무 분담
9. 종합설계 수행일정
10. 필요기술 및 참고문헌
11. Q & A

지난 발표에서의 지적 사항

- ① 어느 정도 수준의 알고리즘 구현이 가능한지 명시가 필요하다.
- ② 콘텐츠는 어떤 콘텐츠인지 명시가 필요하다.
- ③ 콘텐츠 다양성을 설계해야할 필요가 있다.

지적 사항에 대한 답변

- ① 5~7세의 아이들이 너무 어렵지 않게 느껴 흥미를 잃지 않도록 콘텐츠가 구성되어야 하므로 상하좌우 이동이나 회전 같은 간단한 수행의 알고리즘부터 비교적 어려운 Loop 알고리즘 등을 이용하여 문제를 해결하도록 한다.
- ② 블록마다 고유의 기능이 담겨져 있어서 그 기능에 따라 콘텐츠의 스테이지에 맞게 블록을 올바른 순서로 두어 클리어하는 방식
- ③ 콘텐츠 다양성의 한계를 해소하고자 레벨 디자인을 통한 학습 곡선 완화

종합설계 개요

개발 배경

개발 목표

개발 효과

어째서 SW교육이 필요한가?

19세기 이전
농경사회



19 - 20세기
1, 2, 3차
산업혁명 시대

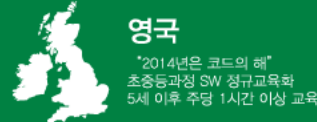


세상의 산업구조가
어떻게 변화되어가는가?

지금 해외에선
무슨일이 일어나고 있는가?



중국, 일본
연간 각 55,70시간 이상
SW 의무교육



영국

"2014년은 코드의 해"
초중등과정 SW 정규교육화
5세 이후 주당 1시간 이상 교육



미국

"하루 한시간씩 코딩하라"
코드교육 캠페인 품
초중등과정 SW 정규교육화

한국또한...

2018년 초중등 SW교육 의무화!

2016 초 5 - 6 대상 <SW 기초 소양교육> 개편

교육용 도구를 이용한 놀이중심의
알고리즘, 프로그래밍 체험 교육!

교육부

미래사회 경쟁력 강화 인식

"여러분의 아이들에게
SW교육이 필요할까요?"

학부모 61%
잘은 모르지만..
필요하다!

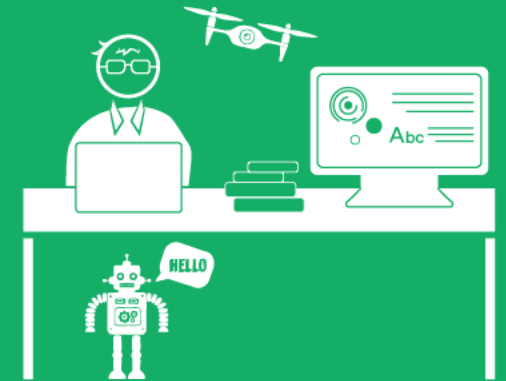
출처 : KBS 뉴스 기사 「러먼폴이기도 알고리즘, 아이들 영어 말고 코드 교육」
NFC 통신 기사 「초등학부모 61%, SW교육 의무화..잘 모르지만 필요하다」
교육부 「2015년 초중고 교육과정 개정 보도자료」

21세기

4차 산업혁명_ 산업 융복합 시대

Digital Natives

기술자, 개발자, 디자이너 간의 장벽이 사라진다.



디터 제체 _ 메르세데스 벤츠 다임러 CEO

"이제 자동차는 가솔린이 아닌 소프트웨어로 달립니다."

버락 오바마 _ 미 전 대통령

"코딩은 당신의 미래일 뿐 아니라 조국이 미래이기도 합니다.
앱을 다운받아 놀지말고, 앱을 만들 수 있어야합니다."

마크 저커버그 _ 페이스북 CEO

"코드를 만들 줄 안다는 것은,
무언가를 만들어낼 수 있는 힘을 갖게 된다는 뜻이고,
아무도 당신을 막을 수 없다는 뜻입니다."

어린이가 알고리즘에 대해
쉽게 학습하도록 하려면
무엇이 필요할까!

종합설계 개요

개발 배경

개발 목표

개발 효과

어린이를 위한 소프트웨어 교육 장난감 !

소프트웨어 교육의 첫 수용자
차세대 디지털 세대의 주체

어린이

: 어린이가 소프트웨어 개념, 즉
알고리즘을 쉽게 익힐 수 있는 장난감을 만들자 !

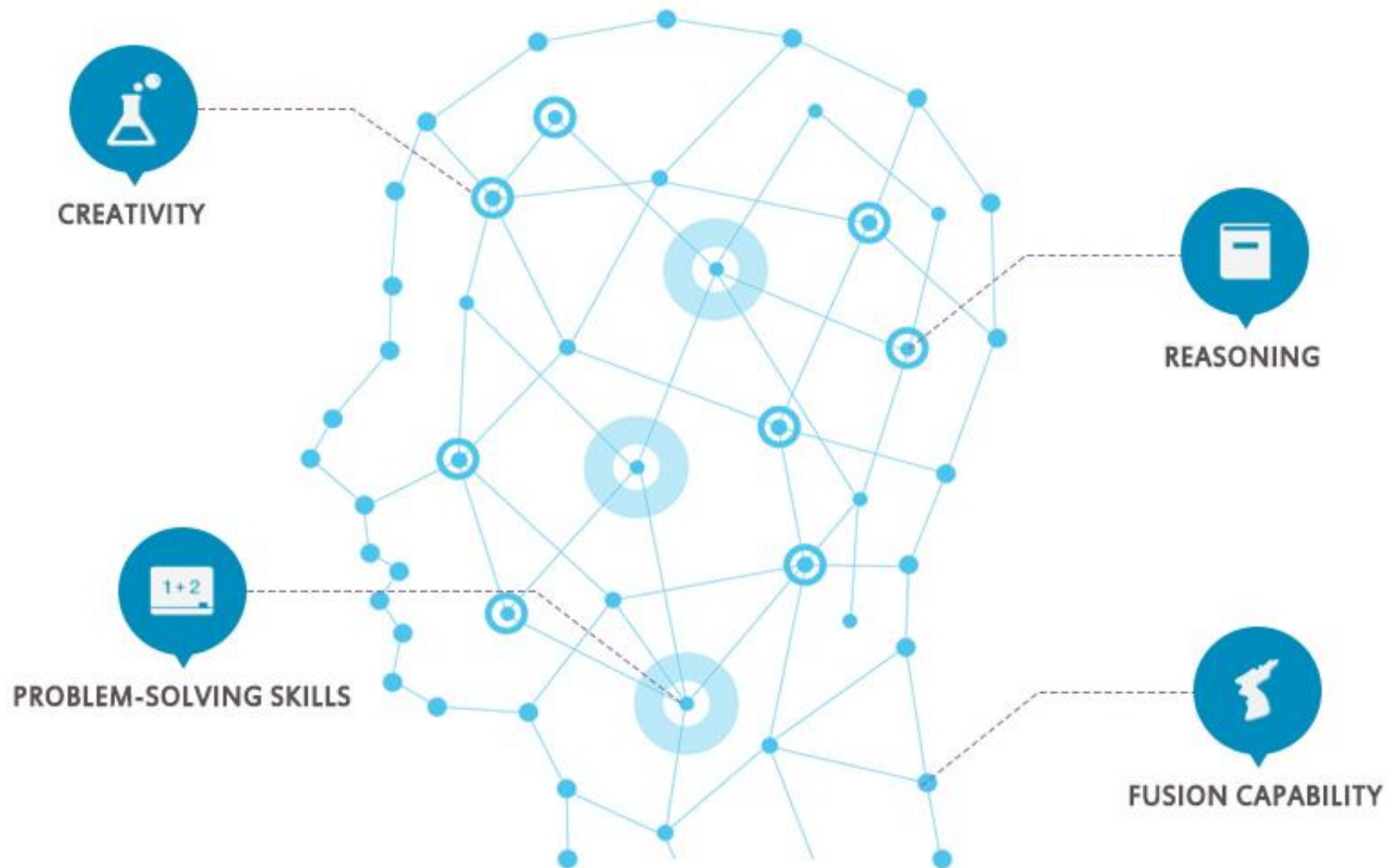


종합설계 개요

개발 배경

개발 목표

개발 효과



관련연구 및 사례

기존 사례

로봇형 제품

Tangible Programming

기존의
교육방식은
어떠한가?

FIRST

블록형 코딩 프로그램
스크래치 등장

SCRATCH



메사추세츠 공대 media labs 개발
2007년 개발, 2009년 국내 도입

현재까지 어린이 소프트웨어 교구로
가장 많이 쓰이고 있는 툴

THEN

블록형 코딩 프로그램
다양화

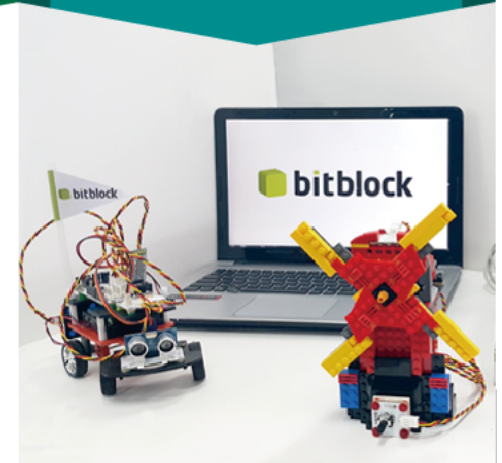


더 넓은 사용연령층을 위하여
더 쉬운 사용성을 가진
스크래치 주니어 출시

스크래치의 오픈소스를 필두로
국산 블록형 코딩 툴
엔트리 출시

AND NOW

코딩 교육용
로봇 장난감 등장



오픈소스 <아두이노>와
<블록형 코드>를 결합한
쉬운 로봇 제어 툴 등장

사운드, 이미지 중심의
디스플레이 교육을 넘어
아이들의 흥미를 돋구는 방향으로

관련연구 및 사례

유형 1_ 로봇형 제품

주로 한국 중소기업에서 최근 뜨겁게 출시되고 있는 로봇 형태의 코딩 교육 장난감.

자칫 딱딱할수 있는 알고리즘의 수학적 개념을 로봇 제어 개념과 접목시켜 친근하게 다가갈 수 있도록 한 제품들이 이목을 끌고있다

기존 사례



로봇형 제품



Tangible Programming



하지만

- 1) 기존에 존재하던 아두이노 + 스크래치의 교구를 외형 디자인만 변화시켜 접근하고자 한 제품들이다!
- 2) 때문에 교육의 난이도, 접근성, 콘텐츠 등 UD 측면에서의 디자인이 고려되지 않고 있다!
- 3) 액추에이터 (구동부품)이 많아질수록 제품 단가가 상승한다!

관련연구 및 사례

유형 2_ Tangible Programming

최근 외국에서는 유아동층을 타겟팅하여
실제 블록을 이용한 **Tangible Programming** 개념이 확산되고 있다.

문제 해결의 순서 개념을 시각적으로 보여주는데 집중한다.

기존 사례

Fisher-Price Code-A-Pillar

Code와 Catapillar (애벌레)의 합성어로,
애벌레의 몸체를 구성하는 각 블록에
방향 명령을 입력해놓으면
그 순서에 맞추어 움직이는 장난감



Google Project : Blocks

구글에서 개발중인 제품으로,
블록간의 다양한 조합을 통해
로봇을 움직이거나
회로 설계를 할 수 있도록 한다.

로봇형 제품

PRIMO Cubetto

상하좌우를 나타내는 블록을
판 위에 순서대로 배치하면
그 순서에 따라
작은 로봇이 움직인다

말을 손으로 움직이는 대신
알고리즘 판을 이용하여 원격조종하는,
보드게임에 로봇조종을 더한 장난감이다



OSMO Game : Coding

디스플레이와 퍼즐간의
상호작용을 이용하는
여러 게임 중 하나로
(Coding)을 선보였다.

컨트롤러가 아닌
블록의 조합을 통해
캐릭터를 움직여 임무를 완수한다.

Tangible Programming



실제 블록의 조합 순서에 따라

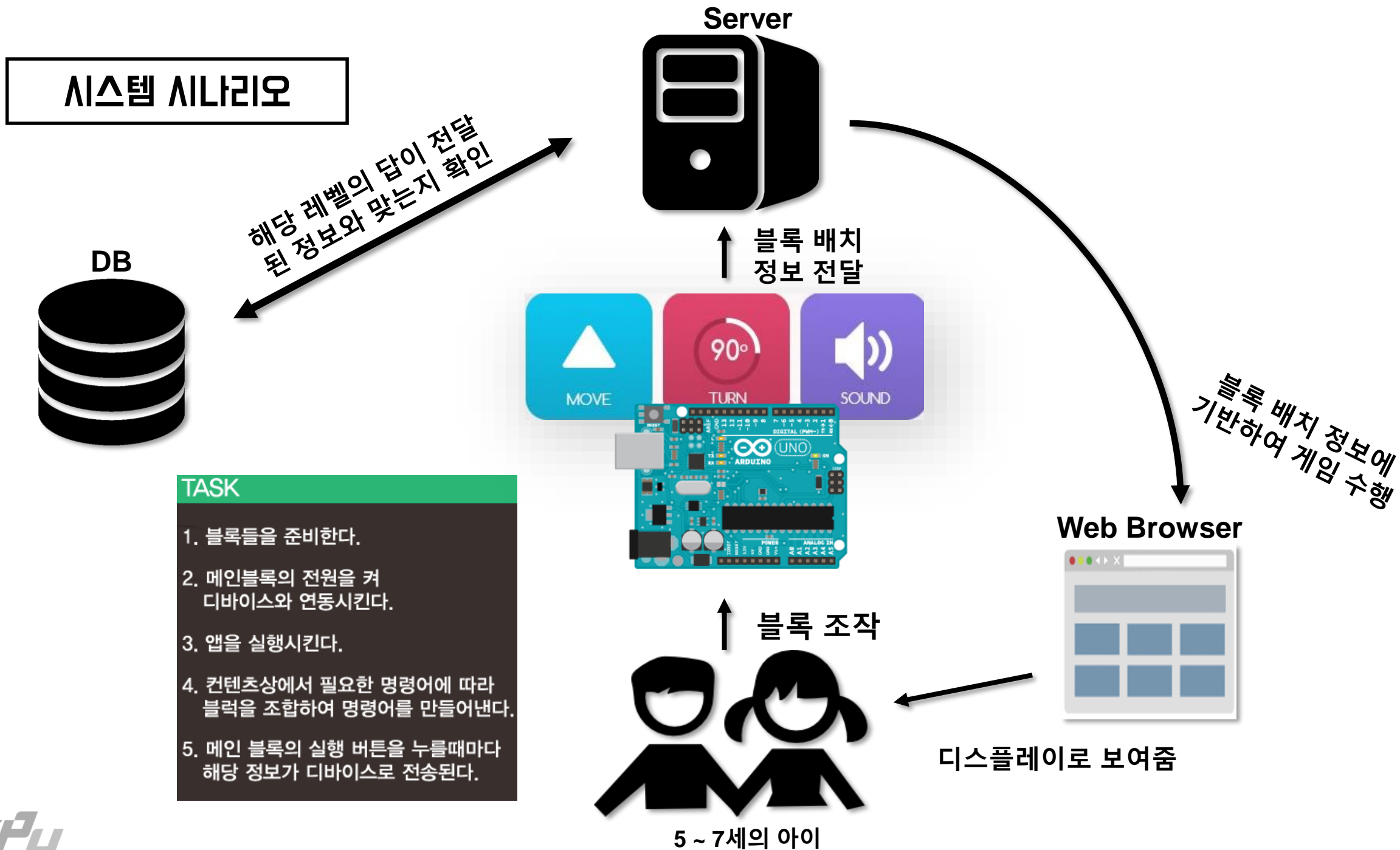


로봇이나 디스플레이 속 캐릭터를 움직이는 방식!

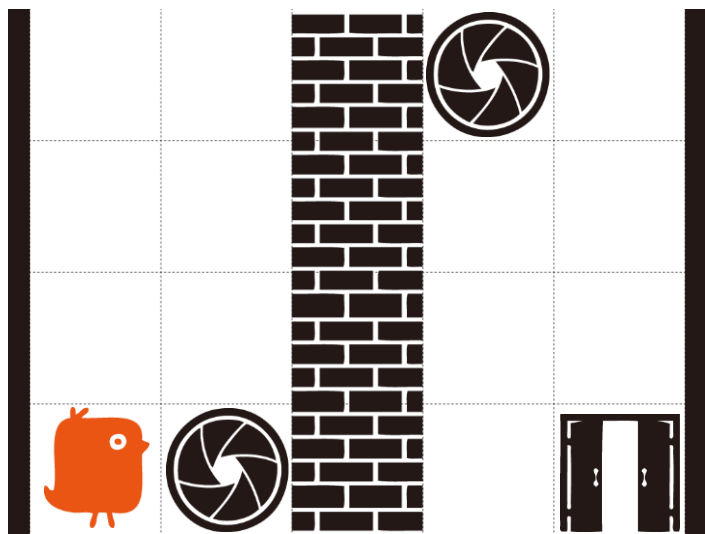
하지만

- 1) 프로그램을 짜고 제어하는데에는 훨씬 많은 수의 명령어가 필요한데, **블록의 수는 제한되어있다!**
- 2) 다수의 블록을 꺼내고, 정리하는데에 **추가적인 어려움이 따른다!**
- 3) 제공될 수 있는 콘텐츠에 **한계가 있다!**

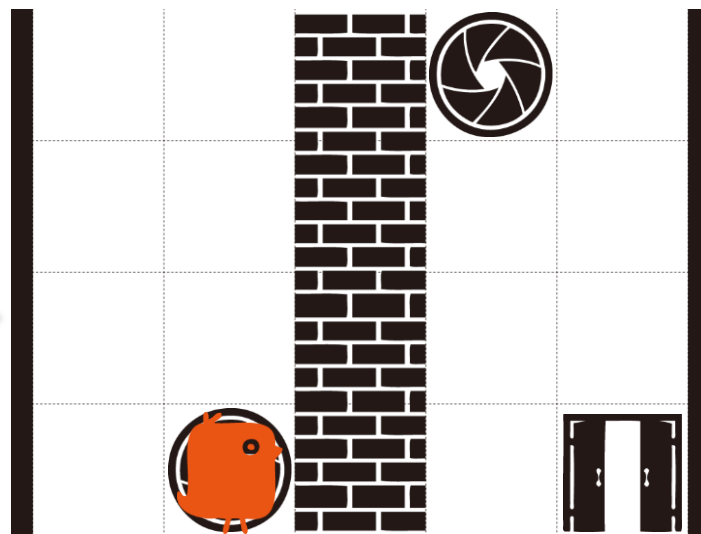
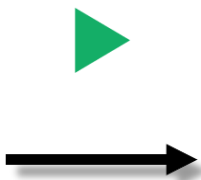
시스템 시나리오



시스템 시나리오



오른쪽 !



오른쪽 ! 점프 ! 왼쪽 !



+

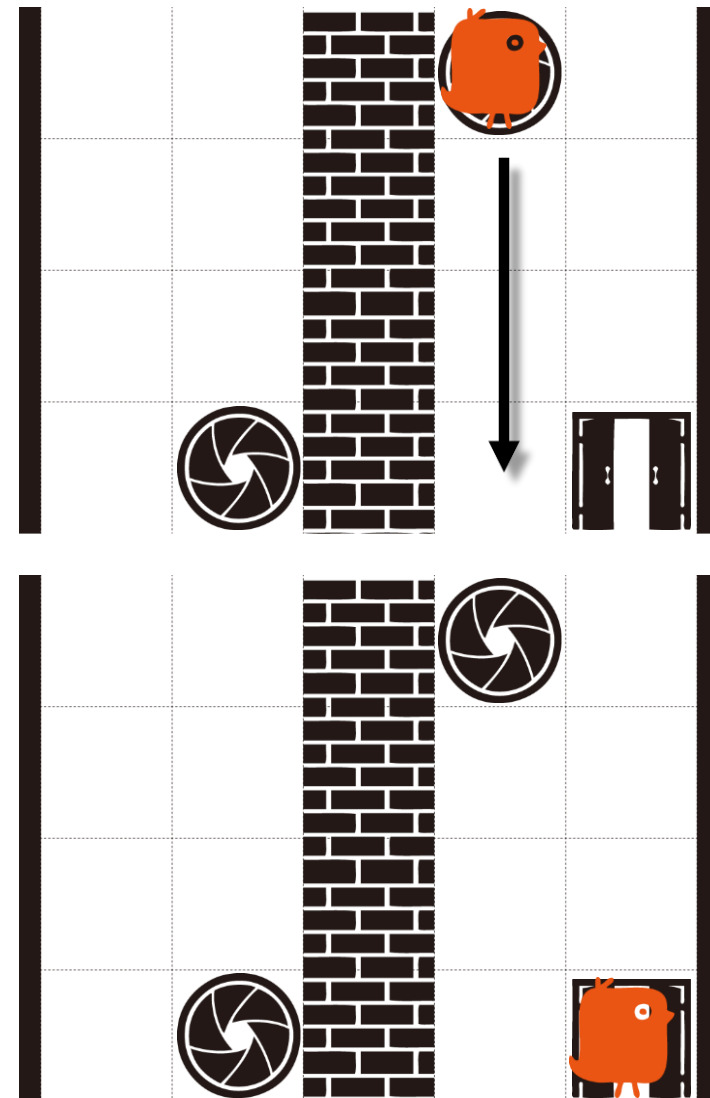
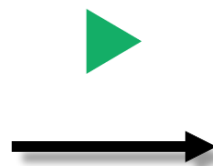
어린이 수준에 맞춘
알고리즘 구현!!

왼쪽 회전 ! 오른쪽 회전 !



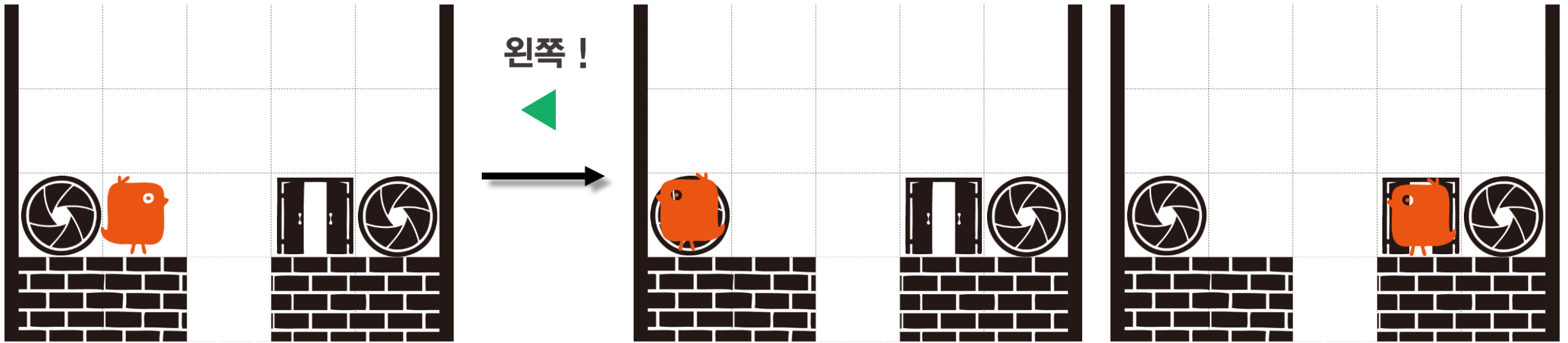
블록마다 고유 기능이 담겨있어 배치된 순서의 블록에
따라서 해당 스테이지를 클리어하는 방식

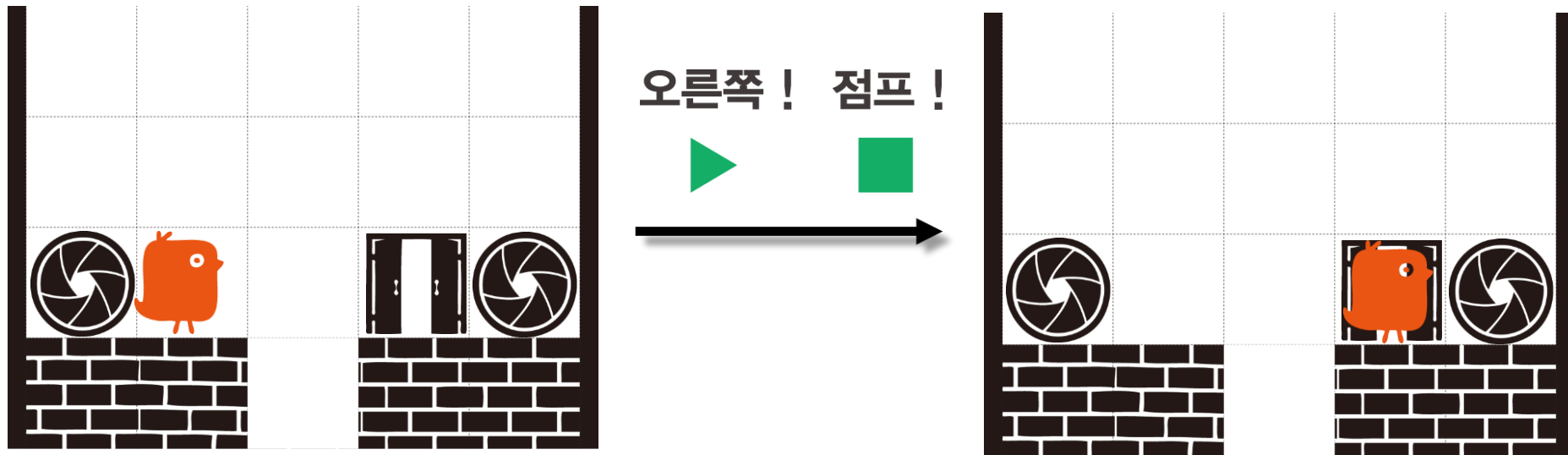
오른쪽 !



시스템 시나리오

컨텐츠 다양성의 한계를 해소하고자 레벨 디자인을 통한 학습곡선 완화





컨텐츠 기획

구현 블록

이동블록
전진하기 / 후진하기 /
회전하기 (90°) / 회전하기(45°)

반복블록
LOOP{ / }

행동블록
줄기

01

튜토리얼
게임 진행방식 소개

02

미로찾기
사용블록 : 전후+회전

03

미로찾기
사용블록 : 전후+회전+반복

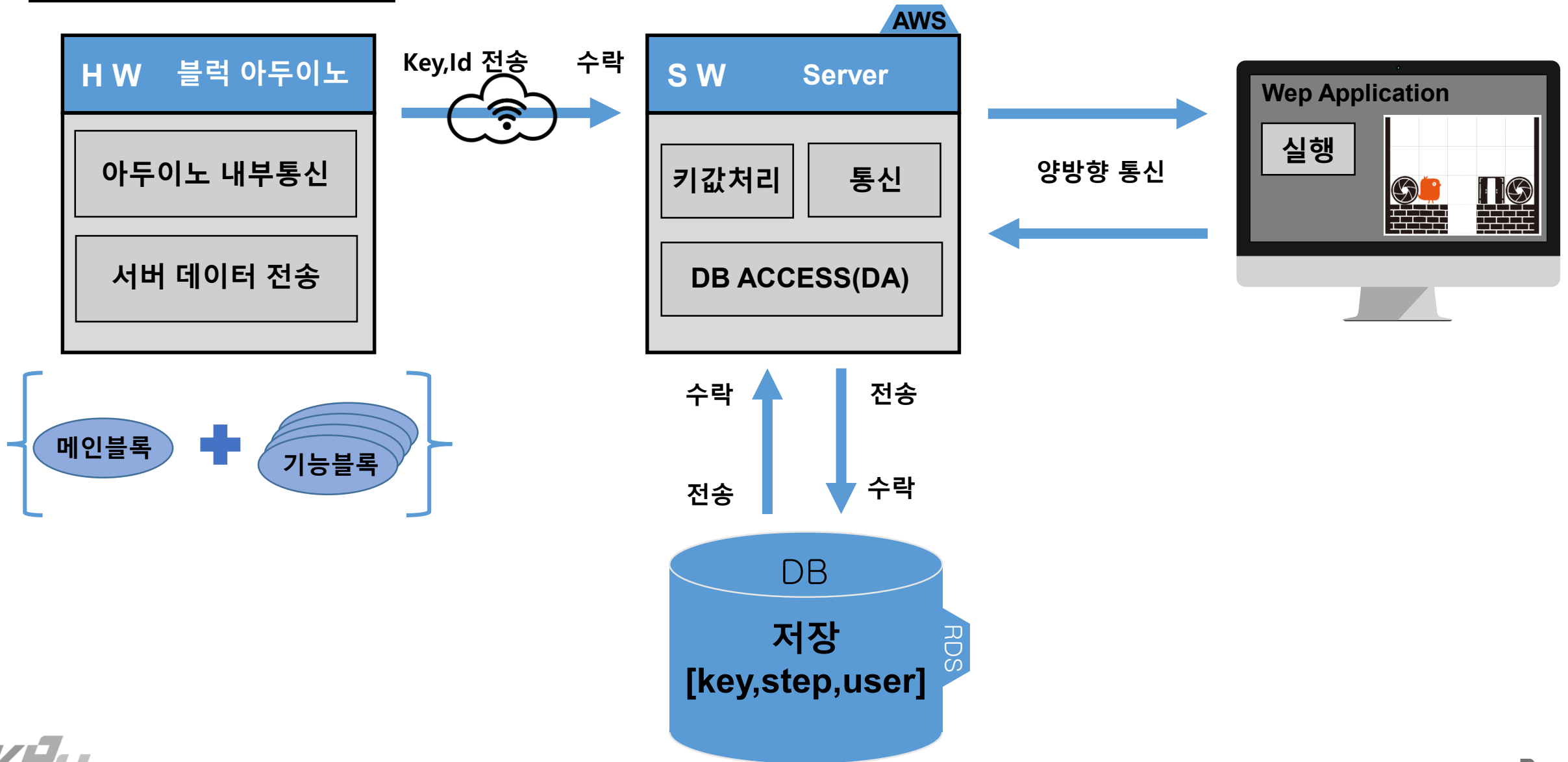
04

길찾으며 행동하기_미로찾기(HARD)
사용블록 : 전후+회전+반복+행동 (줄기)

05

그림그리기_도형 그리기
사용블록 : 전후 (거리) +회전 (각도)+반복+행동(줄기)

시스템 구성도



기능

- > 통신 모듈이 아두이노로부터 수신한 데이터를 구분자를 이용하여 구분 (StringTokenizer 이용)
- > 구분한 데이터를 순서에 맞게 사용할 수 있도록 문자열 배열 **리스트**에 저장

다루는 데이터

- > rcvData: 아직 구분자로 구분이 안된 데이터

예) data1**K**data2**K**data3**K**data4..... 인 문자열 형식(K구분자로 구분)

- > keyValues: 구분자로 구분된 데이터
- > TOKENIZER: final로 선언할 구분자

수행시작 시점

- > 통신 모듈이 아두이노에서 데이터를 수신한 다음 수행

데이터 별 자료형	
ArrayList<String>	KeyValues
String	rcvData

API

ClassifyKeyValues()	
형식	ArrayList<String> ClassifyKeyValues(String rcvData, char tokenizer)
리턴값	구분된 문자열들의 배열리스트
설명	문자열 rcvData를 구분자 tokenizer로 구분하여 배열리스트에 각각 삽입한 후, 배열리스트를 반환
예시	ClassifyKeyValues(rcvData, TOKENIZER);

기능

- > 데이터베이스에 접속
- > 데이터베이스에 원하는 요청에 따른 결과값 반환 및 수정

다루는 데이터

- > databaseInfo: 데이터베이스에 관련된 정보를 저장
- > rcvData: 아직 구분자로 구분이 안된 데이터

예) data1**K**data2**K**data3**K**data4..... 인 문자열 형식(K구분자로 구분)

수행시작 시점

- > 통신 모듈이 아두이노에서 데이터를 수신한 다음 수행

데이터 별 자료형

DB_INFO	databaseInfo
String	rcvData

데이터 구조 DB_INFO

String	DATABASE_URL
String	DATABASE_DRIVER
String	DATABASE_ID
String	DATABASE_PW

API

CheckID()	
형식	boolean CheckID()
리턴값	true나 false
설명	아두이노 별 고유 ID 값 확인
예시	If (CheckID()) { ... }

GetKey()	
형식	String GetKey ()
리턴값	String
설명	사용자가 입력한 답을 데이터베이스에서 가져오기
예시	GetKey();

GetSolutionKey()	
형식	String GetSolutionKey (int step)
리턴값	String
설명	해당 스테이지의 답을 데이터베이스에서 가져옴
예시	GetSolutionKey(step);

GetStep()	
형식	int GetStep (String id)
리턴값	스테이지 단계
설명	아두이노의 고유 ID 값으로 사용자를 식별 후 어떤 스테이지를 진행하고있는지 확인
예시	int step = GetStep(id);

UpdateStep()	
형식	void UpdateStep (String id)
리턴값	x
설명	아두이노의 고유 ID 값으로 사용자를 식별 후 스테이지를 갱신
예시	UpdateStep(id);

UpdateKey()	
형식	void UpdateKey (String key)
리턴값	x
설명	사용자가 입력한 답을 데이터베이스에 등록
예시	UpdateKey(key);

기능

- > 아두이노로부터 HTTP GET 방식으로 데이터를 수신
- > 키 값 처리 모듈이 처리한 데이터를 클라이언트로 전송
- > 클라이언트의 요청 처리

다루는 데이터

- > request: 클라이언트나 아두이노로부터의 요청
- > response: 클라이언트로의 응답

수행시작 시점

- > 아두이노에서 데이터를 서버로 송신한 다음 수행

고려사항

- > express framework의 API 사용



기능

- > 블록 아두이노(슬레이브)의 계산 또는 고정된 데이터를 I2C 통신을 이용하여 메인 아두이노(마스터)에 데이터 전송
- > 각 아두이노 블록으로부터 키값을 도킹 스테이션의 지정된 순서대로 받는다.(마스터)
- > 순서대로 받은 값을 구분자 K 로 두어 하나의 문자열로 저장(마스터)
- > 하나의 문자열로 된 값이 정상적인 논리인지 판별하고, 그에 대한 결과값에 대한 시각적인 피드백 수행(마스터)
(정상시 초록불, 비정상시 빨간불)

아두이노간 데이터 통신방식

- > Inter Integrated Circuit (I2C)

아두이노간 데이터 통신방법

- > 1:N (메인 아두이노 : 블록 아두이노들)

다루는 데이터

- > keyValues : 각 아두이노의 키 값
예) data1 : UPUP / data2 : DOWN / data3 : RIGH
- > AllKeyValue : 연결된 모든 블록 아두이노의 값을 구분자로 저장
예) "data1Kdata2Kdata3Kdata4....." 인 문자열 형식(K구분자로 구분)

수행시작 시점

-> 메인 아두이노 블록에 수행 이벤트가 들어왔을 경우 시행

적용 알고리즘(마스터)

-> 스택 자료구조를 이용하여 알맞은 논리로 되어있는지 검사에 이용(CheckAllValueData() 이용)

적용 알고리즘(통신 시)

-> I2C 수신 알고리즘

- 아두이노간 통신 알고리즘

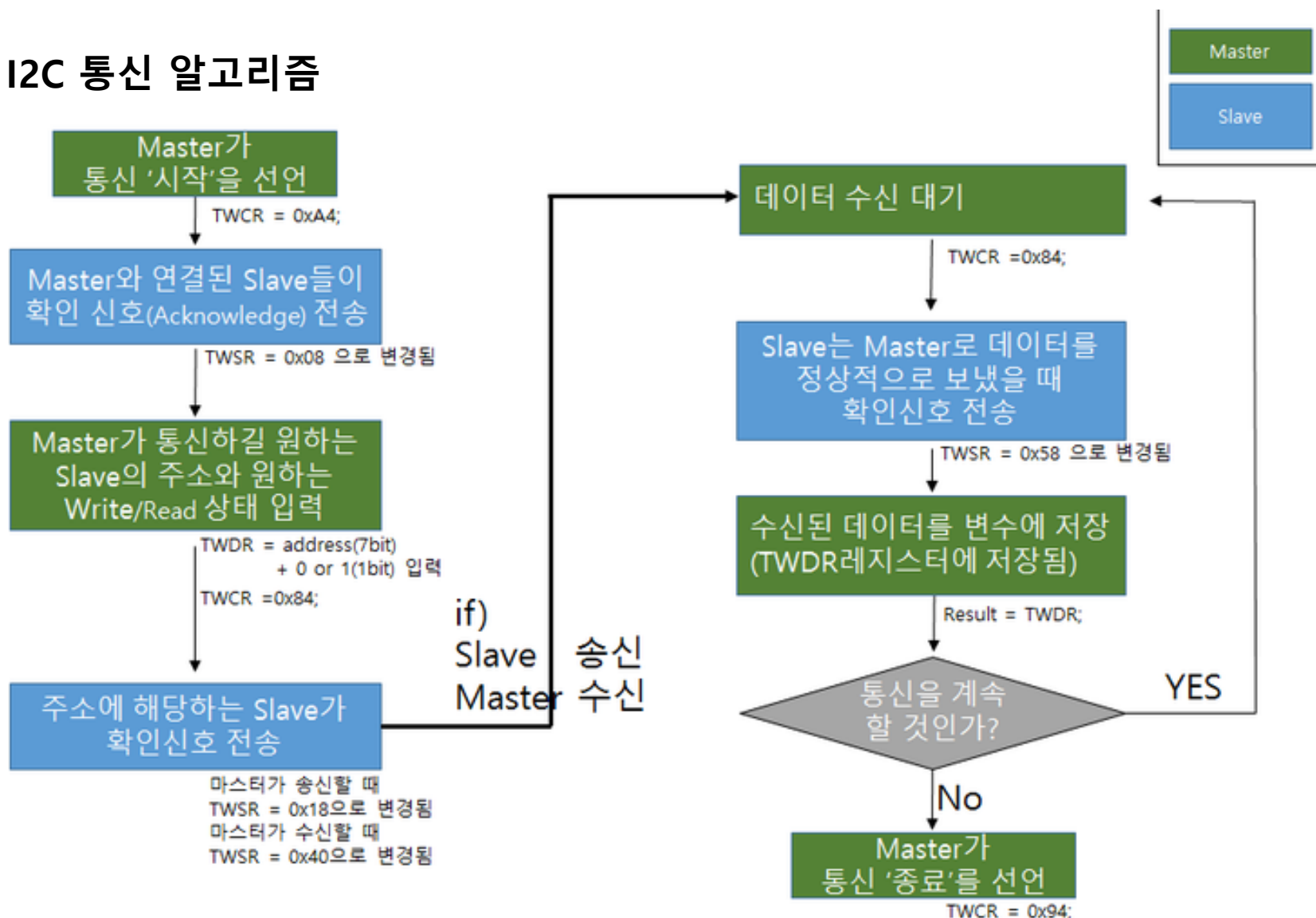
API

CheckAllValueData()	
형식	int CheckAllValueData(char[100] _AllkeyValue)
리턴값	검증 성공 시 1, 실패 시 0
설명	AllkeyValue 문자열 내에 순서도에 맞는 구성으로 되어있는지 판별 한다.
예시	CheckAllValueData(AllKeyValue[100]);

데이터 별 자료형	
char[]	KeyValues
String	AllKeyValue

CombineValueData()	
형식	char[100] CombineValueData(char[4] data1, char[4] data2, char[4] data3)
리턴값	char[100]
설명	data1,data2...의 값을 구분자를 두어 하나의 문자열로 만든다.
예시	CombineValueData(data1,data2,data3,data4.....);

I2C 통신 알고리즘



기능

-> ESP8266(와이파이 모듈) 을 통해 서버에 데이터값을 전송한다.

사용하는 통신규약

-> hypertext transfer protocol (http)

전송 방식

-> Get method

전송 데이터

-> 데이터 전송 시 값은 String 주소 형태로 전송

-> 예) http://hostname:port/?key=dataValue

다루는 데이터

-> AllKeyValue : 연결된 모든 블록 아두이노의 값을 구분자로 저장된 문자열데이터

수행시작 시점

-> 메인 아두이노 블록에 수행 이벤트값이 들어왔을 경우 KeyValue_Get_Module 실행 후 시행

데이터 별 자료형

char[4]	KeyValue
char[100]	AllKeyValue

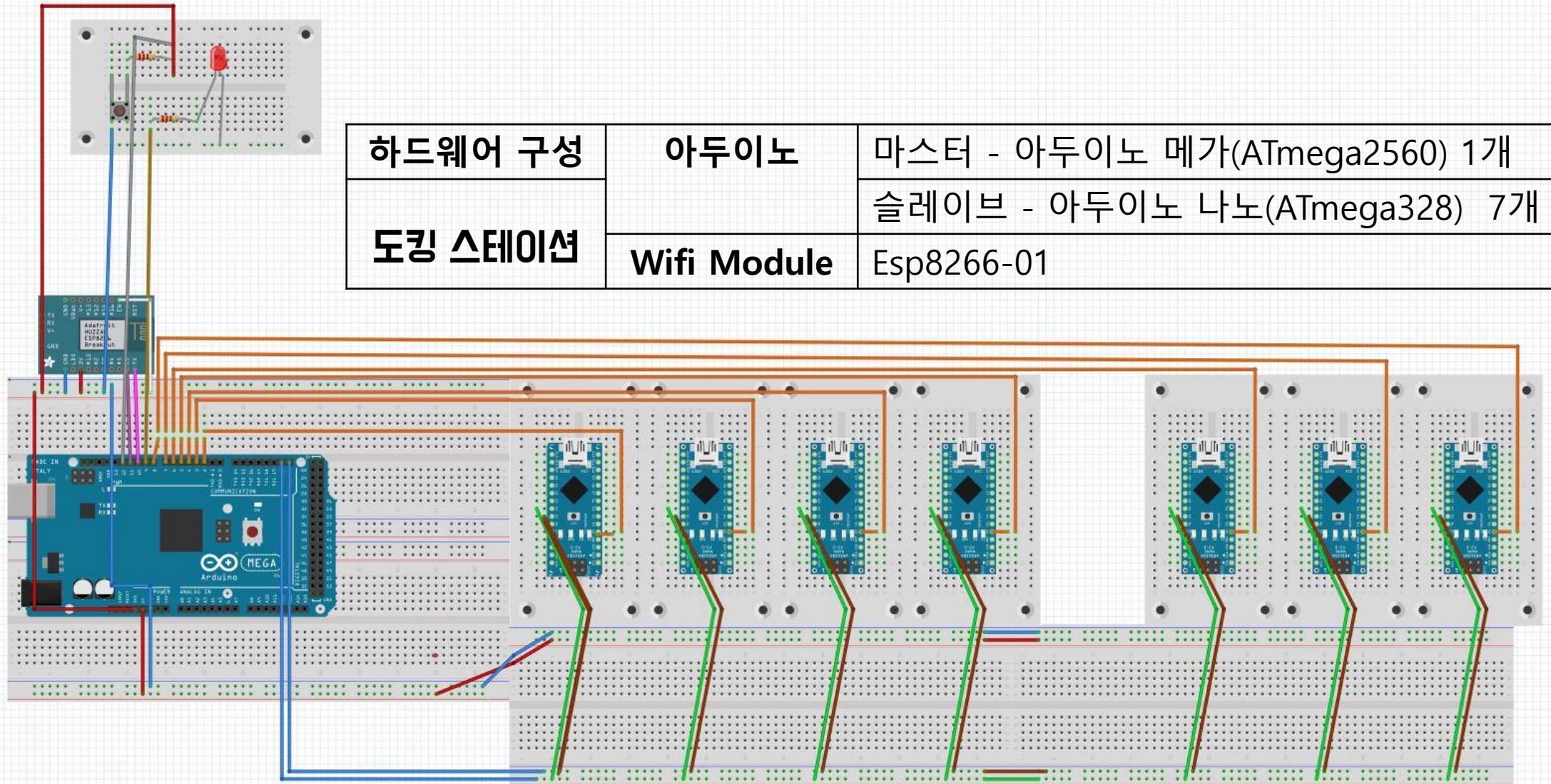
A P I

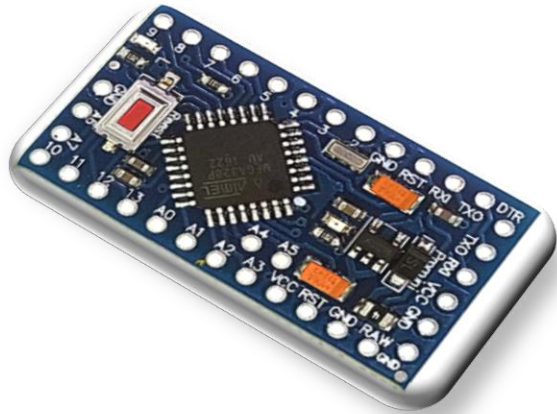
SendAllValueData()

형식	int SendAllValueData(char[100] _AllkeyValue)
리턴값	전송 성공 시 1, 실패 시 0
설명	AllkeyValue 값을 http 통신을 통해 서버에 전송
예시	SendAllValueData(AllKeyValue[100]);

시스템 모듈 상세설계

회로도(HW)





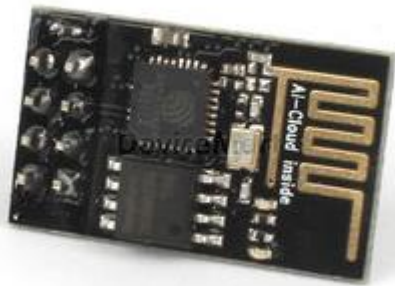
Arduino Mega328

마이크로 컨트롤러	ATmega328
작동 전압	5V
정격 전압	7~12V
허용 전압	6~20V
디지털 입출력핀	14개 (이중 6개는 PWM을 지원함)
아날로그 입력핀	6개
DC 입출력핀의 출력 전류	20 mA
DC 3.3v일 때 출력 전류	50 mA
플래시 메모리	32 KB (이중 0.5 KB는 부트로더가 사용)
SRAM	2 KB
EEPROM	1 KB
클럭 속도	16 MHz



Arduino Mega2560

마이크로 컨트롤러	ATmega2560
작동 전압	5V
정격 전압	7~12V
허용 전압	6~20V
디지털 입출력핀	54개 (이중 15개는 PWM을 지원함)
아날로그 입력핀	16개
DC 입출력핀의 출력 전류	40 mA
DC 3.3v일 때 출력 전류	50 mA
플래시 메모리	256 KB (이중 8 KB는 부트로더가 사용)
SRAM	8 KB
EEPROM	4 KB
클럭 속도	16 MHz



ESP8266-01

S802.11 b/g/n

Wi-Fi Direct(P2P), soft-Ap

TCP/IP 프로토콜 내장

SDIO 2.0, SPI, UART

32핀 QFN 패키지

RF 스위치, balun, 24dBm PA, DCXO, PMU 내장

RISC 프로세서, 메모리, 외부 메모리 인터페이스 내장

MAC/baseband 프로세서 내장

고기능성 오디오 어플리케이션과 인터페이스를 위한 I2S

모든 내부 전원 공급을 위한 low-dropout linear regulator 내장

적절한 spurious-free clock generation 아키텍처

WEP, TKIP, AES, WAPI engine 내장

개발 환경

졸업작품 GitHub 주소

- https://github.com/kwonmory/KPU_KnowAnswer.git

팀원별 GitHub ID

팀장 : 김혁준

ID : kwonmory

팀원 : 서준원

ID : junwon1994



Home

kwonmory edited this page just now · 1 revision

한국산업기술대학교
KOREA POLYTECHNIC UNIVERSITY

KOREA POLYTECHNIC UNIVERSITY

COMPUTER ENGINEERING

Graduation Work GITHUB

Member

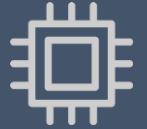
- Seo junwon (12)
- Kwon Hyukjun (12)

Start date 2016-12-01

Report

- Date of Proposals & Plan - 16.12.22
- The Proposal release date - 16.12.26

개발 방법



Hardware

아두이노를 이용하여 서버에 필요한 데이터값 전송하는 하드웨어 구현
ESP8266 모듈을 통해 데이터값을 통신
ARDUINO IDE 1.8.1 로 펌웨어 실시



Server

Amazon Web Service의 EC2를 이용
Node.js 서버 구축



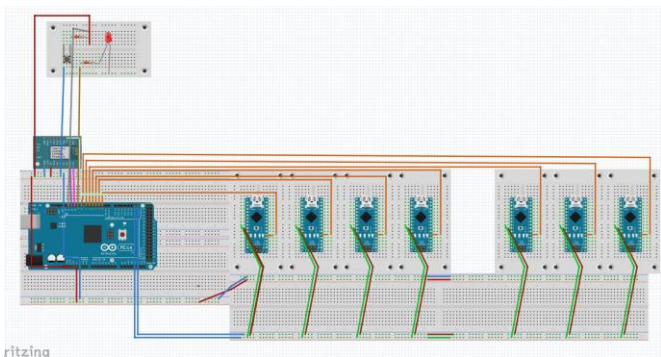
Database

Amazon Web Service의 RDS 이용
MySQL 데이터베이스 구축



Web Application

Node.js, Express, Typescript 으로 개발
Polyfill.js를 이용하여 크로스 브라우징 가능하게 함
웹 기반으로 어플리케이션을 개발함으로써 모든 플랫폼에서 이용 가능하게 함



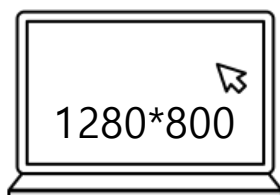
도킹 스테이션 1대



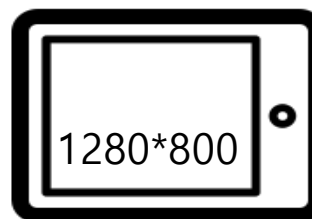
순서 블록 00개



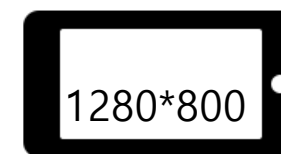
와이파이



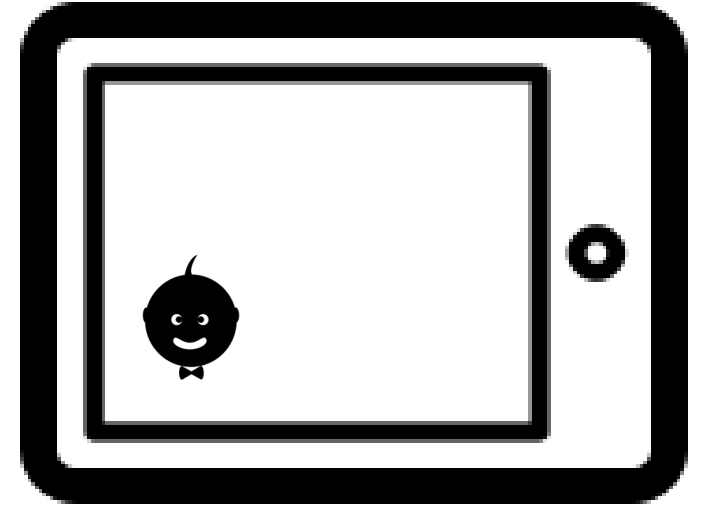
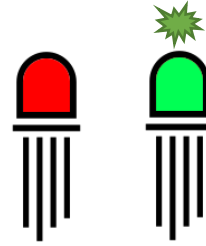
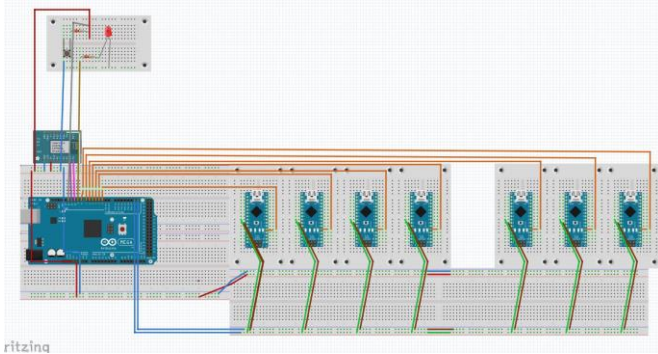
OR



OR



디스플레이 1대(노트북,태블릿,핸드폰 중 택1)





업무 분담

	자료수집	설계	구현	테스트
서준원	ESP8266 Typescript Node.js 웹 구조(UX,UI 포함) 어린이대상 콘텐츠	Web 구조설계 웹서버 구축 콘텐츠 로직	컨텐츠 구현 목업 디자인 출력	데이터 정확성 테스트 최종 테스트
김혁진	ESP8266 Typescript Node.js AWS (서버 & DB) 선행기술조사	ESP8266 설계 Web 통신부분 설계 DB 설계 콘텐츠 로직	컨텐츠 구현 목업 디자인 조립	단위 테스트 컨텐츠 테스트
공통	컨텐츠 개발.유지보수			

종합설계 수행일정

추진사항	12월	1월	2월	3월	4월	5월	6월	7~9월
조사 및 역할 배정	■ ■ ■							
요구사항 정의 및 분석		■ ■ ■						
시스템 설계 및 상세설계			■ ■ ■					
구현 및 프로토타입 제작				■ ■ ■				
통합 테스트					■ ■ ■			
산업기술대전 준비					■ ■ ■	■ ■ ■		
문서화 및 발표						■ ■ ■	■ ■ ■	
최종보고서 작성								■ ■ ■

필요기술 및 참고문헌

필요기술	참고문헌
Node.js	https://nodejs.org/ko/docs/
Express	http://www.expressjs.com
Arduino	www.Arduino.cc
Typescript	https://www.typescriptlang.org/docs/tutorial.html
jQuery	www.jquery.com
AWS (Amazon Web Service)	https://aws.amazon.com/ko/getting-started/?nc2=h_l2_cc

Q & A

별첨자료

지도학생	지도교수	서명란
2014150018 서준원	서대영교수님	
2012151002 김혁준	한익주교수님	