

캡스톤디자인 I 계획서

제 목	국문	블록 코딩을 활용한 하드웨어 제어 프레임워크 개발							
	영문	Development of hardware control framework using Block Coding							
프로젝트 목표 (500자 내외)	<p>최근 스마트팜, 스마트팩토리 등 사물인터넷 기술의 발달로 하드웨어 제어의 중요성이 대두되었다. 산업현장에서도 IoT 기술을 상용화하는 단계에 있으나 하드웨어를 제어하는 것은 여전히 쉽지 않다. 그래서 하드웨어에 쉽게 접근하며 생산성을 향상시키는 것을 목적으로 하드웨어 제어 프레임워크를 개발하고자 한다. 본 프로젝트에서는 프로그래밍을 잘 알지 못하는 사람들도 쉬운 하드웨어 제어가 가능하도록 그래픽 환경에서 프로그래밍이 가능한 Blockly를 선택했다. Google의 오픈 소스 라이브러리인 Blockly는 텍스트 프로그래밍 보다 직관적이고 사용하기 쉽다는 장점으로 Scratch 등 교육 분야에서 활용되고 있다. 본 프로젝트에서는 사용자의 블록을 저장하여 재사용성을 향상시키고, 저장된 블록의 데이터를 바탕으로 사용자에게 블록을 추천해주는 시스템도 개발하는 것을 목표로 한다.</p>								
프로젝트 내용	<p>Front-end에서는 블록 코딩이 가능한 UI를 제공하고, Back-end에서는 Front-end, 하드웨어와 통신하며 요구하는 기능을 수행한다. Blockly를 이용하여 제어하는 하드웨어는 시각적으로 확인이 쉬운 드론을 우선적으로 선택했다. 해당 프로젝트는 사용성 측면에서 긍정적인 효과를 기대할 수 있을지 확인하기 위해 알파 테스트를 진행할 계획이다. 이 테스트를 통해 실제 사용자의 요구사항을 분석하여 부족한 기능을 추가하고, 사용자가 저장한 블록 데이터를 바탕으로 블록 추천 시스템을 도입한다. 나아가 시뮬레이션 엔진을 접목시켜 그래픽 환경에서 다양한 하드웨어를 제어할 수 있도록 구현한다.</p> <ol style="list-style-type: none"> 1. 블록 코딩을 위한 웹 페이지 구축 2. 하드웨어, 소프트웨어 제어를 위한 블록 커스터마이징 3. Front-end와 하드웨어의 연동을 위한 Back-end 개발 4. 알파테스트 후 사용자 요구사항에 따른 추가 기능 구현 5. 블록 추천 시스템 개발 6. 시뮬레이션 엔진 접목 								
중심어(국문)	블록리	사물인터넷	요구사항	블록 추천 시스템					
Keywords (english)	Blockly	Internet of Things	Requirements	Block recommendation system					
멘토	소속	.	이름	.					
팀 구성원	학년/ 반	학 번	이 름	연락처(전화번호/이메일)					
	4/H1	20191743	이민주	010-6308-7346 / 20191743@edu.hanbat.ac.kr					
	4/H2	20191775	백아름	010-8402-3036 / dkfma1458@naver.com					
	4/H2	20191777	송민지	010-3249-0332 / alswl5436@naver.com					
<p>컴퓨터공학과 캡스톤디자인 관리규정과 모든 지시사항을 준수하면서 본 캡스톤디자인을 성실히 수행하고자 아래와 같이 계획서를 제출합니다.</p> <p style="text-align: center;">2022 년 03 월 11 일</p> <p style="text-align: right;">책 임 자 : (인) 희망 지도교수 : 최창범 교수님</p>									

1. 캡스톤디자인의 배경 및 필요성

1) 배경 및 필요성

최근 사물인터넷(Internet of Things, IoT, 이하 IoT) 기술의 발전으로 IoT 기술은 우리 생활 가까이 위치하게 되었다. 하지만 네트워크 기능이 없는 장치를 IoT화하기 위해서는 장치를 제어하기 위한 지식을 익히고 프로그래밍을 통해 플랫폼을 만들어야하는 어려움이 있다. “전기화재 예측 및 예방을 위한 IoT 플랫폼 시스템”이라는 논문에서는 해당 센서 및 시스템을 제작하기 위한 지식이 필요한데, 이를 쉽게 접근하고자 프로그래밍 라이브러리를 이용하여 관련 프레임워크를 제작하였다. 이처럼 쉽게 접근하고자 해도 결국 프로그래밍 기술이 필요하기 때문에 프로그래밍을 접해보지 않은 사람에게는 IoT를 만들고 제어하는 것은 큰 어려움이 있을 수밖에 없다.

Blockly는 블록 프로그래밍 라이브러리로 python과 java 등의 텍스트 언어 보다 쉽고 직관적으로 그래픽 환경에서 코딩할 수 있으며 다양하게 응용될 수 있다. “오픈소스 Blockly를 이용한 모바일용 피지컬 컴퓨팅 개발환경 구축”이라는 논문에서 그 장점을 활용한 사례를 볼 수 있다. 프로그래밍을 처음 시작하는 학생들을 대상으로 하기 때문에 블록 코딩이 가능한 Blockly를 사용하였으며, 하드웨어 장치인 kamibot을 제어하기 위한 블록을 생성하고 학생들은 블록 코딩을 통해 kamibot을 제어할 수 있다. 우리는 이러한 Blockly의 장점을 고려하여 하드웨어 또는 소프트웨어를 제어하는 수단으로 Blockly를 선택하였다.

스마트팩토리, 스마트팜 등 IoT 기술이 발전함에 따라 편의성도 증대 되었지만, 기존의 비싼 장비를 교체하기가 쉽지 않다는 문제점이 존재한다. 이를 해결하기 위해 라즈베리파이를 이용하여 네트워크 기능이 있는 장치로 만들고 여기에 우리가 만든 프레임워크를 결합시켜 비교적 적은 비용으로 IoT 장치가 될 수 있도록 하면 이러한 문제점을 해결할 수 있을 것이다. 또한 사용자 입장에서는 텍스트 기반의 프로그래밍을 모르더라도 블록 코딩을 통해 쉽게 장치를 제어할 수 있어 사용성 또한 높일 수 있을 것이다.

2) 프로젝트의 구체적인 필요성

하드웨어 또는 소프트웨어를 제어하기 위해서는 전송할 데이터의 종류와 전송 방식, 전송 빈도 등을 고려하여 프로그래밍 하는 과정이 필요하다. 예를 들면, tello 드론을 프로그래밍으로 제어한다고 했을 때, take off, up, land 등을 수행하기 위해 사용자는 프로그래밍을 통해 tello가 알아들을 수 있는 명령어로 변환하고 해당 명령어를 tello에게 전송하는 작업이 필요하다. 프로그래밍이 가능한 사용자는 큰 어려움이 없겠지만, 프로그래밍을 처음 접해보는 사람에게는 매우 어려운 일이 될 것이다. 이러한 과정을 감추고 사용자는 블록 코딩만을 통해 하드웨어를 제어할 수 있다면 그 사용성은 더욱 증가할 것이다. 또한 네트워크 기능이 없는 장치에 라즈베리파이 같은 장비를 결합시킨 후 우리가 만든 프레임워크를 이용한다면 그 장치는 IoT화 될 수 있다. 기존의 기술을 비용 등의 문제로 활용할 수 없었던 경우도, 비교적 적은 비용으로 IoT 기술을 사용할 수 있게 되는 것이다. 이러한 장점을 살려 스마트팩토리나 같은 실제 산업현장에 접목 시킨다면 긍정적인 효과도 기대해볼 수 있을 것이다. 또한 교육 측면에서도 학생들이 쉽게 하드웨어를 제어해볼 수 있는 환경을 만들 수 있다는 장점이 있다.

2. 캡스톤디자인 목표 및 비전

본 프로젝트에서는 블록 코딩 환경을 제공해주는 blockly를 이용하여 소프트웨어 또는 하드웨어를 쉽게 제어할 수 있는 프레임워크를 개발하고자 한다. 특히 하드웨어의 경우, 네트워크가 연결되어 있지 않은 장치라도 이 프레임워크를 결합시키게 되면 사물인터넷처럼 쓰는 것이 가능해지도록 한다. 이렇게 사물인터넷화 된 장치를 사용자는 블록 코딩을 통해 제어할 수 있게 된다. pc와 장치 간에 어떤 데이터를 어떻게, 얼마나 자주 주고받을지 등의 문제는 이 프레임워크로 대체하고, 사용자들은 블록 코딩을 통해 보다 쉽게 장치를 제어할 수 있는 것이다. 또한 블록을 저장하고 불러오는 기능을 통해 블록 재사용성을 높여 사용성을 증대시킬 것이다. 더 나아가 상황에 맞는 블록을 추천하는 서비스를 만들어 사용자들은 더욱 쉽게 블록 코딩이 가능해질 것이다. 프로젝트를 진행하는데 있어서의 비전은 좋은 시스템을 개발하는 것뿐만 아니라 실제 산업 현장에서 요구되는 실무 능력과 팀워크 능력을 향상시키는 것이다. 이를 위해 사용자들의 요구사항을 반영하여 프로젝트를 보완하는 것과 팀원들 사이의 원활한 의사소통에 집중할 것이다.

3. 캡스톤디자인 내용

1) 요구사항 목록

유형	내용	개수	비고
기능 요구사항 (Functional Requirements)	기능 요구사항(SFR) (System Function Requirement)	7	
비기능 요구사항 (Non-Functional Requirements)	성능 요구사항(PER) (Performance Requirement)	2	
	인터페이스 요구사항(SIR) (System Interface Requirement)	6	
	테스트 요구사항(TER) (Test Requirement)	3	
	제약사항(COR) (Constraint Requirement)	8	
합 계		26	

2) 상세 요구사항

가. 기능 요구사항 (System Function Requirement)

요구사항 명	고유번호	요구사항 정의 및 세부내용	비고
블록 커스터마이징	SFR-01	① Blockly의 기본 블록에 추가적으로 하드웨어 제어에 필요한 블록을 커스터마이징한다. ② 블록 카테고리 별로 색상을 설정하여 구분이 쉽게 한다. ③ 작업 공간 확대, 축소 기능	
사용자 블록 저장	SFR-02	① 결합된 블록을 Local에 저장, 삭제, 불러오기 기능 ② 결합된 블록을 json 등 외부 파일로 저장하는 기능	
블록 추천 시스템	SFR-03	사용자가 사용한 블록의 유사성을 분석해서 블록을 추천하는 기능	
하드웨어 제어	SFR-04	① DJI Tello Drone 제어 기능 ② 라즈베리파이 제어 기능 ③ 아두이노 제어 기능	
하드웨어1. 드론	SFR-05	① 추가된 블록은 take off, land, flip, forward, up, down, rotate, wait 등이며 각 블록은 드론 제어를 위한 동작을 수행한다. ② UDP 통신을 이용해 드론에 명령어를 보내고 조종한다. ③ 드론에서 정보를 획득하고 활용하는 기능	
하드웨어1. 드론 안전성 준수	SFR-06	① 비상 상황 발생 시 자동으로 드론이 착륙하는 등 안정적인 시스템 운영을 위한 기능 ② 클라이언트 하나 당 연결 가능한 드론의 개수는 임의로 1개로 제한 ③ 드론의 배터리가 일정 값 이하라면 자동으로 착륙하는 기능	
소프트웨어 제어	SFR-07	시뮬레이션 엔진과 결합하여 소프트웨어 제어가 가능하도록 기능을 확장한다.	

나. 성능 요구사항 (Performance Requirement)

요구사항 명	고유번호	요구사항 정의 및 세부내용	비고
처리 속도 및 시간	PER-01	① 하드웨어의 종류에 따라 처리 속도의 제한을 다르게 설정한다. ② 드론의 경우에는 응답이 오는 시간을 최대 15초로 제한한다.	
느린 작업에 대한 사전 경고	PER-02	일정 시간 이후에도 하드웨어에서 응답이 없다면 사용자에게 팝업 메시지로 알린다.	

다. 인터페이스 요구사항 (System Interface Requirement)

요구사항 명	고유번호	요구사항 정의 및 세부내용	비고
UI 표준 준수	SIR-01	UI 표준에 근거한 일관성 있는 사용자 인터페이스를 설계한다. 시스템의 추후 확장성을 고려하여 카테고리 별로 구조화한다.	
직관성	SIR-02	조작방법이 편리하고 인터페이스의 각 버튼 부분이 어떤 기능을 수행하는지 알아보기 쉽게 설계한다.	
사용자 인터페이스	SIR-03	① 사용자의 이용이 편리하도록 웹 기반으로 구축한다. ② 시스템의 모든 기능은 웹 브라우저를 통하여 작동되도록 한다.	
데이터 전송	SIR-04	① GET 방식으로 하드웨어 혹은 소프트웨어와의 연결 여부를 확인한다. ② POST 방식으로 Json 형태의 데이터를 전송한다.	
블록 데이터	SIR-05	워크스페이스에 존재하는 블록들을 모두 Json 형태로 전송한다.	
서버와의 통신	SIR-06	웹 페이지에서 서버에게 데이터를 요청하고, 하드웨어를 통해 수집한 데이터는 웹 페이지에서 확인이 가능하도록 설계한다.	

라. 테스트 요구사항 (Test Requirement)

요구사항 명	고유번호	요구사항 정의 및 세부내용	비고
장비 성능 테스트	TER-01	① 연결 예정인 드론의 배터리 성능을 파악한다. ② 하드웨어의 정상적인 동작 여부를 테스트한다.	
단위 테스트	TER-02	일부 선행되어 개발되는 부분은 일정에 맞춰 계획 수립 후 부분적으로 테스트를 진행한다.	
인수 테스트	TER-03	블록 코딩으로 제어가 가능한 하드웨어 중 시각적으로 확인이 쉬운 드론을 선택해서 테스트를 진행한다. 프로그래밍 경험이 없거나 적은 사람들을 대상으로 해당 프로젝트의 알파 테스트를 진행한다. 사용자의 추가 요구사항을 수집하고 해당 내용을 바탕으로 기능을 추가하는 등 프로젝트를 보완한다.	

마. 제약사항(Constraint Requirement)

요구사항 명	고유번호	요구사항 정의 및 세부내용	비고
시스템 기술	COR-01	Google의 Blockly open source를 이용한다. Blockly는 웹 및 모바일 애플리케이션에 블록 형태의 시각적 코드 편집기를 추가하는 라이브러리이다.	
항공안전법 (드론 관련)	COR-02	① 비행장치 신고 의무: 중량에 상관없이 모든 드론은 관할 지방항공청에 신고하며, 기체 신고필증을 교부 받아야 한다. 간편하게 드론 원스탑 민원포털서비스를 이용하여 신고하는 것도 방법이다. 드론 소유자는 신고번호가 잘 보일 수 있도록 드론 기체에 적절한 방법으로 표기하여야 한다. ② 조종자 준수 사항: 단순 취미용 드론(무인비행장치)이라도 모든 조종자가 준수해야 할 안전수칙을 항공안전법에 정하고 있고 조종자는 이를 지켜야 한다. 조종자 준수 사항은 비행장치의 무게나 용도와 관계없이 무인비행장치를 조종하는 모든 사람에게 적용된다.	*드론 원스탑 민원 포털 서비스 https://drone.onestop.go.kr
인터넷 연결	COR-03	컴퓨터는 연결 대상 하드웨어의 AP 혹은 유선으로 연결하여 통신을 준비한다.	
로컬 호스트	COR-04	로컬 호스트인 http://127.0.0.1:5000/ 에 접속해서 웹 서버를 확인한다.	
컴퓨터 연결	COR-05	제어하고 싶은 하드웨어 혹은 소프트웨어는 같은 컴퓨터에 연결을 해야 한다.	

블록 코딩	COR-06	워크스페이스에서 작성한 블록은 모두 하나로 연결되어야 한다. 사용하지 않는 블록은 워크스페이스의 우측 하단에 위치한 쓰레기통으로 드래그 후 워크스페이스에서 제거한다.	
AI 학습용 블록 데이터 수집	COR-07	블록 자동 추천 시스템을 위해서 학습할 데이터는 사용자가 저장한 블록의 Json 데이터를 수집한다.	
제어 장비 리스트화	COR-08	블록 코딩으로 제어 할 장비의 종류에 따라 장비와의 연결 방법이 달라지므로 각 장비별로 리스트화 해서 정리한다.	

4. 캡스톤디자인 추진전략 및 방법

1) 캡스톤 디자인 목표 달성을 위한 추진전략, 수행방법 및 추진절차

(1) 캡스톤디자인에 대한 이해

[캡스톤디자인 추진 시 예상 문제점 및 준비방안]

	예상 문제점	준비방안
개발 측면	① 블록 코드를 각각의 소프트웨어 또는 하드웨어에 맞게 변환하는 작업이 필요 ② Blockly 오픈소스에 존재하지 않는 블록이 추가적으로 필요 ③ 블록 자동 추천 시스템을 만들기 위한 블록 데이터 확보의 문제 ④ 장비에서 보내는 제어 메시지 등을 수신해야 한다.	① 블록 코드를 서버에서 json형태로 받은 후, 장치에 맞는 명령어로 변환 ② 필요한 블록을 Blockly developer tools를 이용해 커스터마이징 한다. ③ 알파테스트 진행 시 사용한 블록을 저장하도록 하고 그 데이터를 수집한다. ④ 시뮬레이션 엔진을 서버와 결합하여 장비에서 보내는 관련 신호를 수신한다.
사용자 측면	① 네트워크 연결이 불가능한 장치는 사용이 어려움 ② 프로그래밍을 아예 접해보지 않은 경우, 블록 코딩도 어려울 수 있음 ③ 사용자가 원하는 블록이 없을 수 있음 ④ 원하는 빈도수만큼 메시지를 받는 기능이 필요	① 라즈베리파이 등을 장치와 결합해 네트워크를 사용할 수 있도록 한다. ② 블록 기능에 대한 매뉴얼을 제공한다. ③ 유지보수를 통해 개선 ④ 시뮬레이션 엔진 관련 블록을 커스터마이징 하여 사용자가 원하는 빈도수만큼 메시지를 얻을 수 있게 한다.

(2) 캡스톤디자인 경험

인공지능 프로젝트를 통해 습득했던 python 프로그래밍 실력을 바탕으로 python 웹 프레임워크인 Flask를 사용해 서버를 구현한다. 또한 팀 프로젝트 진행에서 가장 중요한 것은 팀원들 사이의 원활한 소통임을 인지하여 매주 자신의 작업물 정리 및 공유 활동을 진행한다. 이 활동을 통해 다른 팀원이 맡은 개발 내용도 확실하게 숙지하도록 한다. 프로젝트 결과의 완성도에만 치중하지 않고 팀원 모두가 함께 연구하고 발전하는 것을 목표로 삼는다.

(3) 프로젝트 관리체계 수립

① 프로젝트 통합 관리:

프로젝트 진행 상황, 태스크 관리, 코드 병합 등 프로젝트 전체를 관리한다.

② 일정 및 범위 관리:

매주 해야 할 태스크를 세부적으로 정리하고 분담한 후, 수행 기간을 공지한다.

③ 코드 관리:

코드의 복잡성, 오류 등을 검사해서 간결하고 깔끔한 코드가 유지 되도록 관리한다.

④ 태스크 정리 및 공유:

자신이 수행한 태스크, 질문 등을 정리해서 매주 공유한다.

(4) 수행방법

블록 프로그래밍을 위한 웹 페이지는 Javascript를 사용하여 구현한다. front-end의 블록을 하드웨어 혹은 소프트웨어와 연결시키기 위한 back-end는 python 웹 프레임워크인 Flask를 사용한다. 서버를 이용해 제어할 수 있는 하드웨어 영역은 드론, 라즈베리파이 등 점차 확장해 나갈 것이며, 소프트웨어는 시뮬레이션 엔진을 이용한 챗봇 등을 계획하고 있다. 또한 시스템이 잘 동작하는지, 사용성이 좋은지 등을 알아보기 위해 학생들을 대상으로 알파테스트를 진행하며, 테스트를 위한 하드웨어는 드론으로 한다. 그 이유는 코드의 동작을 시각적으로 확인하기 좋고 학생들에게 드론을 체험할 수 있는 기회를 제공할 수 있기 때문이다. 그리고 테스트 후 설문조사를 통해 피드백을 얻어 프로젝트를 보완한다. 또한 사용자들이 사용했던 블록 파일을 모아 블록 추천 시스템을 위한 데이터로 사용할 것이다.

2) 캡스톤디자인 목표 달성을 위한 팀 구성 체계 및 역할

(1) 프로젝트 관리 역할

- ① 프로젝트 통합 관리 : 백아름
- ② 일정 및 범위 관리 : 이민주
- ③ 코드 관리 : 송민지
- ④ 태스크 정리 및 공유 : 공동

(2) 개발 관련 역할

- ① front-end : 공동
- ② back-end : 공동

5. 참고문서

- [1] 오주연 외 2명, 「BSQL : Blockly를 활용한 SQL 학습용 블록 코딩 환경 개발」, 한국컴퓨터교육학회, 2020년
- [2] 조상, 「Blockly webc 프로그래밍 융합 학습시스템」, 한국융합학회논문지, 2015년
- [3] 조은주, 문미경, 「오픈소스 Blockly를 활용한 모바일용 피지컬 컴퓨팅 개발환경 구축」, 한국차세대컴퓨팅학회, 2017년
- [4] 양승의 외 3명, 「전기화재 예측 및 예방을 위한 IoT센서-게이트웨이-서버 플랫폼 개발」, 한국정보통신학회, 2021년