**<PROJECT 스펙보고서>**

**스마트 도어**

**팀원**

**2015104184 신은섭**

**2014104124이규태**

**교수님**

**김재홍**

Table of Contents

Abstract

Introduction

Background Study

Goal/Problem & Requirements

Approach

Project Architecture

Architecture Diagram

Architecture Description

Implementation Spec

Input/Output Interface

Inter Module Communication Interface

Modules

Current Status

Future Work

Division & Assignment of Work

Schedule

[Appendix] Detailed Implementation Spec

# **Abstract**

**문의 잠금을 풀때 비밀번호를 처야 해서 불편했던 순간이 누구나 한번 쯤은 있을 것이다. 손대지 않고 잠금을 풀기 위해 블루투스와 스마트폰 어 플리케이션을 이용한 ‘스마트 도어’ 시스템을 소개한다.**

# **Introduction**

매일 들어가는 현관문, 번호키인 현관문을 열 때 번호조차 치기 귀찮을 때가 있다. 또, 두손 가득 장을 보고 돌아오면 현관문 비밀번호를 치기 위해 짐을 내려 놓아야 한다. 이런 불편함을 해결하기 위해 가까이만 가면 잠금이 자동으로 해제되는 스마트 도어 시스템을 제안한다.

# **Background Study**

## **관련 접근방법/기술 장단점 분석**

1. **스마트 폰을 이용한 사용자의 위치 탐색**

|  |  |  |  |
| --- | --- | --- | --- |
| **기술** | **설명** | **장점** | **단점** |
| **GPS** | **GPS위성의 정보를 받아 위치를 알려주는 시스템.** | **구현이 간단하다.** | **실내에서 잘 동작하지 않는다.**  **사용자가 GPS 기능을 켜놓아야 한다.** |
| **Bluetooth Beacon** | **Bluetooth Beacon에서 송신하는 신호를 받아 근처에 오면 정보를 알려주는 시스템.** | **실내, 근거리에서 GPS보다 오차가 작다.** | **먼 거리에서 오차가 크다. 사용자가 Bluetooth기능을 켜놓아야 한다.** |
| **고주파 근거리 탐지** | **고주파 소리를 내고 있으면 근처에오면 위치정보 서비스를 제공하는 시스템.** | **근거리에서 오차가 적다.**  **사용자가 다른 기능을 켜지 않아도 된다.** | **먼 거리에서 오차가 크다.**  **사용자가 마이크 접근을 허용해야 한다.** |

1. **라즈베리 파이와 스마트폰과의 무선통신**

**라즈베리 파이 3는 Bluetooth와 Wi-Fi를 내장하고 있기 때문에 두 가지 중에 한가지 방법을 선택하여 통신이 가능하다.**

## **프로젝트 개발환경**

|  |  |
| --- | --- |
| **IoT Board** | **Raspberry pi 3 b** |
| **IoT OS** | **Raspbian Stretch** |
| **Android OS** | **Android 8.0.0** |
| **Development Language** | **Python** |
| **IDE** | **VS Code** |

# **Goal/Problem & Requirements**

**등록된 휴대폰을 들고 문 근처로 가면 자동으로 잠금이 해제되는 문**

# **Approach**

**문에 부착된 장치가 블루투스 비콘 신호를 보내고 있다.**

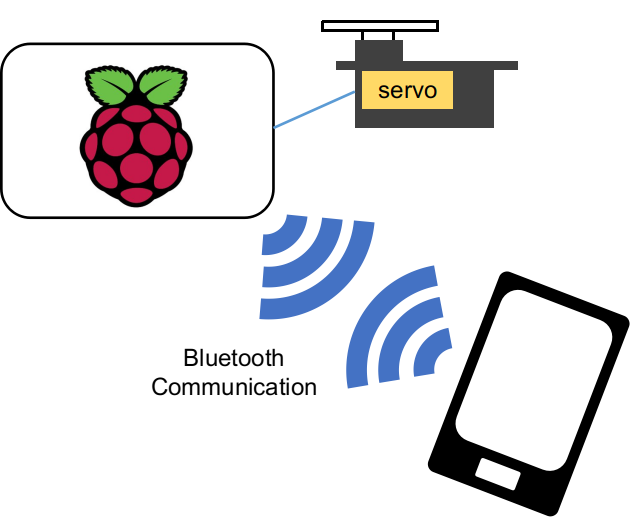
**어플리케이션이 주기적으로 신호를 감지 하고 있다.**

**신호 감지 중에 비콘 신호가 들어오면, 블루투스로 열림 신호를 전송한다.**

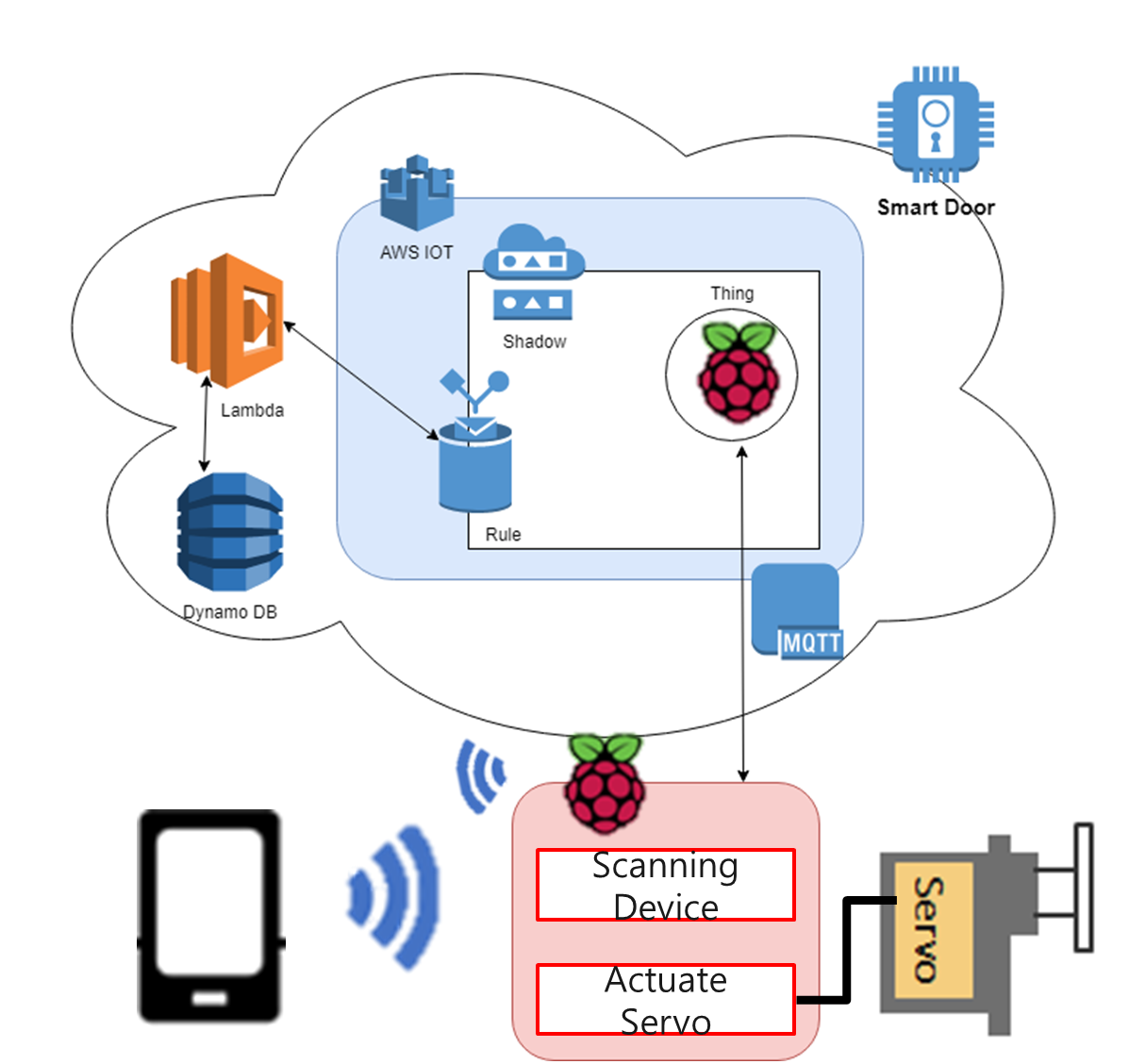
**신호를 받은 장치가 적합한 사용자인지 인증 후에 문을 연다.**

# **Project Architecture**

## **Architecture Diagram**



## **Architecture Description**

****

# **Basic Spec**

1. Raspberry pi
2. **Bluetooth Beacon**
3. **Control Servo Motor**
4. **Recive ‘Open’ message and open door**
5. **Application**
6. **Scan Bluetooth beacon and Sent ‘Open’ message**

# **Current Status**

1. **Door Thing ID Update**

**Smart Door Thing은 보안을 위해 일정 시간마다 ID를 업데이트 합니다. ID는 aws IoT Shadow를 통해 관리됩니다.**

|  |
| --- |
| **{**  **"reported": {**  **"state": "Active",**  **"doorid": "P789G8GP"**  **}**  **}** |

**기본적인 동작 상태의 Shadow는 위와 같습니다. Door Thing이 일정 주기로 ID를 요청합니다. 다음과 같은 상태로 바뀝니다.**

|  |
| --- |
| **{**  **"desired": {**  **"state": "RequireNewID",**  **},**  **"reported": {**  **"state": "RequireNewID",**  **"doorid": "P789G8GP"**  **}**  **}** |

**Desired의 State가 “RequestNewID”로 바뀌면 AWS IoT Role Engine이 다음과 같은 SQL로 상태변화를 감지합니다.**

|  |
| --- |
| SELECT \*, topic(3) as thingName  FROM '$aws/things/+/shadow/update'  WHERE state.desired.state = 'RequestNewID' |

**Role Engine이 상태변화를 감지하면 Lambda의 GenerateNewID 함수를 실행하여 IoT Shadow를 다음과 같이 업데이트 합니다.**

|  |
| --- |
| **{**  **"desired": {**  **"state": "Active",**  **"doorid": "CRWI5SCT"**  **},**  **"reported": {**  **"state": "Active",**  **"doorid": "P789G8GP"**  **}**  **}** |

**최종적으로 Door Thing이 이를 Shadow에 반영하여 ID가 업데이트 됩니다.**

|  |
| --- |
| **{**  **"reported": {**  **"state": "Active",**  **"doorid": "CRWI5SCT"**  **}**  **}** |

**모든 변경사항 로그는 AWS Cloud Watch에 기록됩니다.**

1. **Bluetooth와 Smart Phone 연결**

**라즈베리 파이가 주기적으로 신호를 보내 페어링 할 디바이스를 검색한다. 라즈베리 파이에 저장된 MAC과 스마트폰의 MAC이 동일할 경우 IOT를 작동시켜 문을 연다.**

**2-1 지속적인 연결을 시도하는 코드(5초 간격)**

|  |
| --- |
| **while True :**  **if target\_address != "None" :**  **target\_address = "None"**  **user = connect()**  **sleep(3)** |

**2-2 연결 대상을 페어링 하고, 페어링 완료 후 해당 기기의 정보와 라즈베리 파이에 저장된 정보를 확인한다.**

|  |
| --- |
| **def connect() :**  **nearby\_devices = discover\_devices()**  **#scanning for target device**  **for device\_id in nearby\_devices:**  **print(lookup\_name(device\_id))**  **if target\_name == lookup\_name(device\_id):**  **target\_address = device\_id**  **break**  **if target\_address is not None :**  **try:**  **sock.connect((target\_address, port))**  **print(target\_address)**  **#if check\_user(target\_address) == True :**  **# check\_signal(sock)**  **sock.close()**  **return True**  **except btcommon.BluetoothError as err:**  **print("An error occurred : %s " % err)**  **return False** |

# **Division & Assignment of Work**

|  |  |
| --- | --- |
| 항목 | 담당자 |
| Bluetooth Beacon | 이규태 |
| Control Servo Moter | 신은섭 |
| Recive ‘Open’ message and open door | 신은섭 |
| Scan Bluetooth beacon and Sent Open message | 이규태 |

# **Schedule**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 내용 | 9월 | | | 10월 | | | | 11월 | | | | 12월 | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| Bluetooth Beacon |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Control Servo Moter |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Issue New Thing ID |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Recive ‘Open’ message and open door |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Scan Bluetooth beacon and Sent Open message |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 버그 수정 및  데모 테스트 |  |  |  |  |  |  |  |  |  |  |  |  |  |