

SmartX-mini 센터를 활용한 스마트 도어 서비스 실험

이준기, 김승룡, 송지원, 김종원
 광주과학기술원 정보통신공학부
 e-mail : {jglee, srkim, jwsong, jongwon}@nm.gist.ac.kr

Experimenting Smart Door Service employing SmartX-mini Center

Jungi Lee, Seungryong Kim, Jiwon Song, JongWon Kim
 School of information and communications, Gwangju Institute of Science and Technology

요 약

사물인터넷에 대한 연구가 활발해지면서 많은 분야에 접목이 시작되면서, 스마트 홈 자동화에 대한 관심도 새롭게 확대되고 있다. 본 논문에서는, SmartX-mini 로 호칭하는 소형 IoT-Cloud 테스트베드를 활용하여, 스마트 홈 자동화의 일례로 집 내부 문들의 정보를 실시간으로 확인하는 스마트 도어 서비스를 빠르고 효율적으로 시험적으로 구현해 본다.

1. 서론

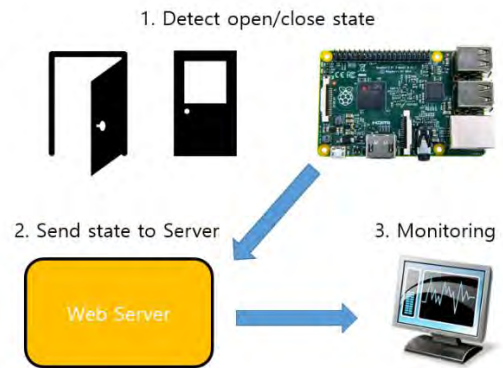
사물인터넷(IoT: Internet of Things) 개념이 발전함에 따라, 스마트 홈 자동화(automation)에 대한 연구가 새롭게 진행 중이다[1]. IoT 서비스에서, IoT 단말은 정보의 입출력을 담당하고 중앙의 서버가 전체적으로 IoT 서비스를 총괄하여 관리하게 된다. 한꺼번에 여러 IoT 단말 장치들을 관리하고 제어하기 위해서는 중앙의 서버에 이들을 관제하기에 충분한 초융합형(컴퓨터/스토리지/네트워킹) 자원의 동적인 할당이 필요하다. 클라우드 컴퓨팅(Cloud computing) 환경은 소요되는 융합형 자원을 동적으로 제공하면서 적당한 수준으로 유지할 수 있게 한다.

상기한 IoT 와 Cloud 가 접목되는 IoT-Cloud 서비스의 실증을 위해 요구되는 자원 소요를 충족시키기에 용이한 테스트베드용 시험환경을 위한 초융합형 박스형 자원 집합(resource pool)을 SmartX Box 라고 호칭하고 있다[2]. 본 논문에서는 Raspberry Pi 2 와 같은 저사양의 기기를 가지고 구축한 SmartX 시험환경, 즉 SmartX-mini 시험환경의 활용을 다루고자 한다. SmartX-mini 시험환경에서는 μ -Box 라 불리는 여러 IoT 단말들을 연결해서 IoT-Cloud 테스트베드를 구축하고, 구현하고자 하는 IoT-Cloud 서비스를 쉽고 빠르게 실증해 본다.

특히 본 논문에서는 홈 안전(safety)에 관련된 부분에 집중하면서 IoT 단말들과 중앙 서버를 이용해 많은 수의 집 내부 문들을 관리하는 서비스를 구현해 본다. SmartX-mini 에 μ -Box 로 구성된 IoT 단말들을 시험적으로 활용하여 홈 자동화를 위한 서비스를 구현해보고, 다가오는 IoT-Cloud 환경에 있어서 SmartX-mini 의 활용 가능성을 검증해본다.

2. Smart Door Management 서비스

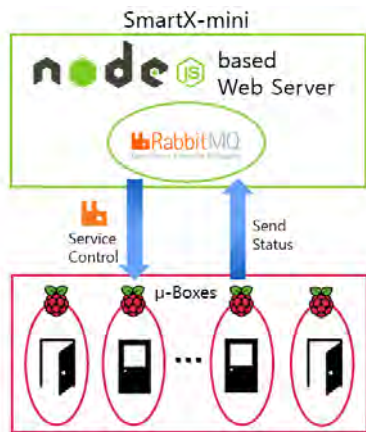
SmartX-mini 와 IoT 단말을 활용해 문의 개폐 여부를 확인할 수 있는 서비스를 구현하고, 이를 Smart Door Management 서비스라 칭한다. <그림 1>은 Smart Door Management 서비스의 전반적인 구도를 나타낸다. SmartX-mini 에서는 웹 서버를 통해 Smart Door Management 서비스를 제어하는 중심 서버의 역할을 한다. IoT 단말 부분에서는 센서를 이용해 문의 개폐 여부를 확인 후 웹 서버로 전송한다. 그리고 사용자는 웹 서버를 통해 실시간으로 문의 개폐 여부를 확인할 수 있다.



(그림 1) Smart Door Management 서비스

Smart Door Management 서비스는 SmartX-mini 를 활용하여 효율적인 서비스의 관리 및 운용을 할 수 있다. 여러 대의 IoT 단말 장치를 하나의 중앙 서버에서 관리하는 데 있어 SmartX-mini 를 통한 컴퓨팅 자원의 적절한 할당으로 IoT 단말들에 대한 효과적인 관리가 가능하다.

3. 설계 및 구현



(그림 2) Smart Door Management 시스템 구성도

<그림 2>는 Smart Door Management 서비스의 전체 시스템 구성도이다. Smart Door Management 서비스는 SmartX-mini 와 각 문에 연결된 μ -Box 를 활용해서 서비스 환경을 구성하였다. 이번 논문에 사용된 SmartX-mini 는 총 6 대의 Intel NUC 을 사용해 구성하였다. NUC 기기에는 Ubuntu 14.04 LTS 를 사용하였으며 SmartX-mini 의 컨트롤러의 역할을 하는 NUC 은 4 세대, 나머지 5 대의 NUC 은 3 세대 모델을 사용하였다. 자세한 성능은 <표 1>에 확인할 수 있다.

(표 1). Intel NUC 성능

Model	CPU	SSD	Networking
D54250WYKH	Core i5-4250U	512GB	Gigabit Ethernet
DC53427HYE	Core i5-3427U	128GB	Gigabit Ethernet

SmartX-mini 에서는 IoT 단말 역할을 맡은 μ -Box 들을 관리하기 위해 Node.js 기반의 웹 서버를 활용한다. Node.js 는 Google 의 V8 엔진 기반의 이벤트 처리 프레임워크이다. Node.js 는 Non-blocking I/O 와 단일 스레드 방식을 통해 빠른 처리성을 보인다는 장점을 지니고 있다[3]. 본 논문에서는 Node.js 기반 웹 서버에 Google Analytics[4]의 API 를 이용해 문의 개폐 여부를 확인할 수 있도록 구현하였다[5].

IoT 단말의 역할을 하는 μ -Box 는 저렴한 가격과 가격 대비 높은 성능을 보이는 Raspberry Pi 2 모델을 사용한다(<표 2>). Raspberry Pi 2 에는 Raspbian OS 를 설치하여 사용하였다. 구현한 서비스에서는 Raspberry Pi 2 를 통해서 문의 잠금 여부를 확인하기 위해 두 센서가 일정 거리 안에 있을 때 인식을 하는 마그네틱 센서를 활용한다.

(표 2). Raspberry Pi 2 성능

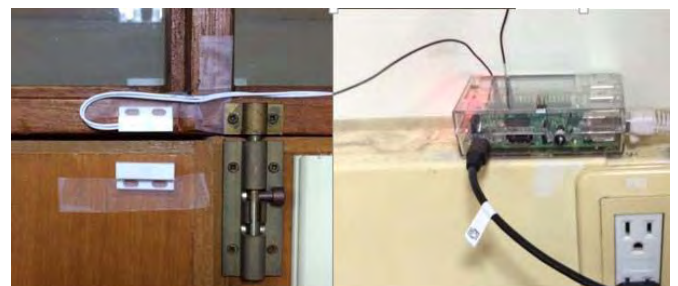
Model	CPU	GPU	Networking
-------	-----	-----	------------

Raspberry Pi 2	900MHz ARM Cortex-A7 Quad core	Broadcom VideoCore IV	10/100Mbit/s Ethernet
----------------	--------------------------------	-----------------------	-----------------------

Smart Door Management 서비스의 관리는 Node.js 기반 웹 서버가 담당하며, 각 IoT 단말의 제어는 RabbitMQ[6]를 통해 이루어진다. 즉 Node.js 기반 웹 서버상에서 사용자가 RabbitMQ 를 통해 μ -Box 에 명령을 보내어 서비스의 시작과 종단을 지시할 수 있다.

<그림 3>은 SmartX-mini 와 1 대의 Raspberry Pi 2 만을 사용하여 Smart Door Management 서비스를 구현한 모습이다. SmartX-mini 와 연결된 Raspberry Pi 2 에 마그네틱 센서를 연결하여 μ -Box 를 구성한 모습이다.

<그림 4>는 Smart Door Management 서비스의 구동 결과이다. μ -Box 에 연결된 문이 열리고 닫힐 때마다 문의 현재 상태를 Google Analytics 서버에 전송하고, 그 결과를 사용자가 웹상에서 확인할 수 있다.



(그림 3) Smart Door Management 구현 모습

```

pi@raspberrypi ~ $ sudo python doorlock.py
Door Open
Door Close
Door Open
Door Close
    
```



(그림 4) Smart Door Management 구동 결과

4. 결론 및 향후 계획

본 논문에서는 SmartX-mini 와 μ -Box 를 활용해서 서비스를 구현해 보았다. 이를 통해 SmartX-mini 를 활용한 IoT 서비스의 운용 가능성을 확인할 수 있었다.

구축한 서비스는 SmartX 에 하나의 μ -Box 만을 연결하여 실험하였지만, 추후 여러 대의 μ -Box 를 SmartX-mini 와 연결해 서비스 범위를 확장하거나, 웹 UI 를 통해 여러 문의 정보를 실시간으로 확인할 수 있으며, Pi 의 적외선 카메라 모듈 등을 이용해 추가적인 기능들을 적용해 웹 애플리케이션의 기능이 확장된 스마트 도어 관리 서비스를 운용할 수 있을 것으로 기대된다. 향후에는 나아가 보다 강화된 기능을 가진 서비스를 위해 여러 IoT 단말들의 정보를 수집해 로그를 남기고 방문자들의 사진 정보를 저장해 놓는 웹 서버를 구축할 예정이다.

감사의 글

본 논문은 2014 년 미래창조과학부의 재원으로 SW 융합기술고도화 사업의 지원을 받아 수행된 연구임 (S1004-14-1045).

참고문헌

- [1] Best smart home devices of 2015, <http://www.cnet.com/topics/smart-home/best-smart-home-devices/>.
- [2] 김남곤, 김종원, “효율적인 자원 활용을 위한 SmartX 노드 가상화 적용,” KCC 2013, 2013 년.
- [3] Node.js, <https://nodejs.org>.
- [4] Google Analytics, <http://www.google.com/analytics/>.
- [5] Nico Miceli, “Physical Analytics Part 1: Tracking Your Home with Google Analytics,” <http://nicomiceli.com/tracking-your-home-with-google-analytics/>.
- [6] RabbitMQ, <https://www.rabbitmq.com>.