

N.lock

2019.06.13

한국외국어대학교

정보통신공학과

1팀

201502766 이호찬

201503183 조동철

201701954 신주호

201700124 이산가비두샤

머리말

본 문서는 아두이노, 라즈베리파이와 안드로이드의 소켓통신을 이용한 노트북 도난을 방지하는 자동 잠금 시스템을 기술한다. 사용자는 노트북과 일정거리 이상 떨어지게 되면 노트북이 자동으로 잠기고 누군가 노트북을 도난하려는 시도를 하면 어플을 통해 사진으로 알림을 받을 수 있다.

목 차

1. 개요	9
1.1 목적	9
1.2 용어 및 약어	9
2. 제품 소개	11
2.1 제품 개요	11
2.2 제품 필요성	12
3. 시스템 구성 및 설계	14
3.1 전체 시스템 구성도	14
3.2 세부 시스템 구성도	14
3.3 소프트웨어 및 하드웨어	14
4. 요구사항 및 제한사항	14
5. 시스템 시나리오	15
5.1 시나리오	15
5.2 시나리오 어플 수행 순서	15
6. 기대효과	16
7. 프로젝트 세부일정	17
8. 참고 문헌	18
8.1 관련 문서	18
8.2 참고 사이트	18
9. [부록]N.lock_SourceCode	19

1. 개요

1.1 목적

카페, 도서관, 독서실에서 등 노트북은 우리가 생활하는 모든 곳에 함께해왔다. 하지만 오래전부터 노트북 도난 사건은 꾸준히 발생해왔다. 그래서 이 시스템은 잠시 자리를 비우는 사이에 노트북이 도난 당하는 것을 방지할 수 있는 자동 안전 잠금 시스템이다. 다른 사람이 자동으로 잠금이 걸린 노트북을 도난 하려는 시도를 하는 등의 행동을 한다면 캠으로 사진이 찍혀서 시스템 사용자의 어플로 알림이 가서 보다 쉽게 범인을 잡거나 안심하고 자리를 비울 수 있게 될 것이다.

1.2 용어 및 약어

용어 및 약어	풀이	비고
iBeacon	iBeacon은 애플에서 출시한 비콘표준이다. ios, 안드로이드 OS에서 모두 활용이 가능하고 ios에서 직접 비콘 신호를 처리하기 때문에 어플리케이션 백그라운드에서 실행할 필요가 없다. iBeacon에서 전송하는 패킷은 UUID(16bit string of numbers), Major(4 digits), Minor(4 digits)이 있다.	일반적으로 한 곳에 대량의 비콘을 설치할 때 모두 동일한 UUID를 이용하도록 하고 major와 minor값을 설정하여 각각 구분된 영역 또는 공간을 설정한다. UUID: UUID만으로 영역을 설정할 경우 전 지점을 하나의 영역으로 지정가능
아이 BLE4.0 (Bluetooth Low Energy)	블루투스는 휴대폰, 노트북, 이어폰 등의 휴대기기를 서로 연결해 정보를 교환하는 근거리 무선 기술 표준을 뜻한다. 주로 10미터 안팎의 초단거리에서 저전력 무선 연결이 필요할 때 쓰인다. 기존의 bluetooth와 가장 큰차이는 훨씬 적은 전력을 이용하여 classic과 비슷한 수준의 무선통신을 할 수 있다.	
AWS	아마존닷컴에서 개발한 클라우드 컴퓨팅 플랫폼이다. AWS는 사전 계약금이나 장기 약정 없이 저렴한 총량, 과금제 방식으로 운영된다. 확장형 글로벌 인프라를 구축 및 관리하고, 더 저렴한 요금의 형태로 사용자에게 비용 절감 혜택을 고스란히 돌려준다. 즉시 새로운 앱을 배포하고, 수요를 기준으로 축소할 수도 있다AWS는 언어 및 운영 체제에 구애받지 않는 플랫폼이다. 따라서 사용자의 비즈니스에 가장 적합한 개발 플랫폼 또는 프로그래밍 모델을 선택할 수 있다	
RSSI	수신 신호 강도 (Received Signal Strength Indicator) 단말에 수신되는 모든 power의 총 크기, 신호 세기에 인접한 채널의 간섭, 열잡음이 포함, TX Power는 수신 신호의 세기 RSSI 값과 더불어 거리를 측정해 자신의 위치를 추정하는데 사용된다.	거리를 구하는 공식은 다음을 이용하였다. $RSSI = -10\log d + TxPower$ $d = 10^{((TxPower - RSSI) / (10 * n))}$

2. 제품 소개

2.1 제품 개요

‘N.lock제품은 Notebook lock의 줄임말로 자동 안전 잠금 장치를 뜻하는 상품명이다. 어플을 통해 알람을 받음으로써 도난을 방지할 수 있는 사용자가 안심할 수 있는 서비스를 제공한다.

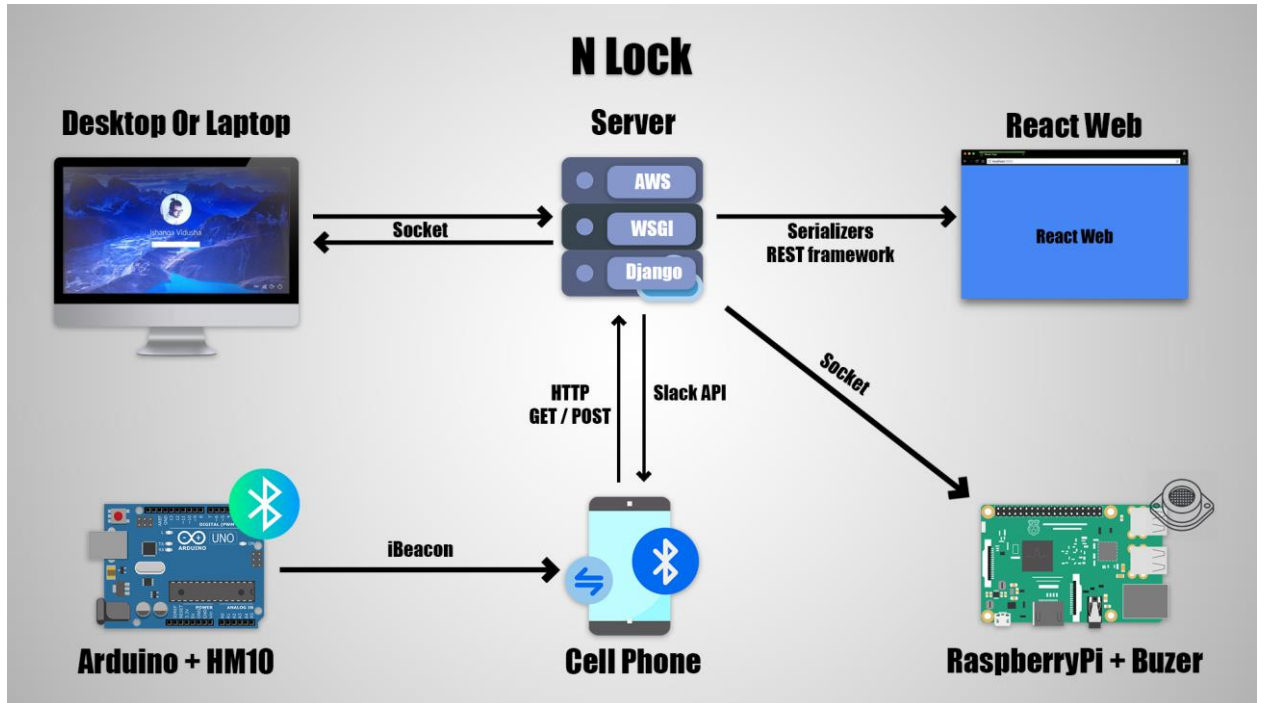
2.2 제품 필요성



위 사진은 노트북 도난 사건에 대한 기사들이다. 도서관, 카페, 독서실 등 일상생활에서 우리는 노트북과 함께한다. 노트북 도난 사건은 보통 우리가 잠깐 화장실을 가는 등 자리를 비우는 상황에 발생한다. 노트북이 이미 도난 당한 후에 되찾는 방법은 매우 복잡할 뿐 아니라 개인이 직접 하기 어렵기 때문에, 도난을 방지하는 것이 해답이라 생각하여 이런 자동 잠금 장치 서비스를 제공하고자 하였다.

3. 시스템 구성 및 설계

3.1 전체 시스템 구성도



3.2 세부 시스템 작동 순서

- 1) **iBeacon 작동**: 아두이노에 연결되어 있는 hm10의 iBeacon을 활성화 시켜서 iBeacon에서 제공되는 정보를 핸드폰으로 수신한다.
- 2) **핸드폰에서 서버로 iBeacon 정보 수신**: HM10(블루투스4.0 모듈)에서 iBeacon 정보를 핸드폰에서 수신하고 POST 형식으로 서버로 송신한다. iBeacon에서 차례대로 <Beacon MAC Address, iBeacon UDID, iBeacon Major Number, iBeacon Minor Number, TX Power at 1m, RSSI> 정보가 송신된다.
- 3) **iBeacon 정보를 거리 값으로 환산**: $d = 10^{(-RSSI + A/10n)}$ 공식을 기반으로 정확한 거리 값을 산출한다.
- 4) **윈도우 자동 잠금, 해제**: 사용자가 노트북으로부터 일정거리 떨어지면 자동으로 윈도우 잠금 상태가 되고 일정거리 가까워지면 자동으로 잠금이 풀린다.
- 5) **노트북에서 비정상 사용 감지**: 노트북 잠금 상태에서 키입력이 있을 때, 서버에 신호를 보내 사용자가 노트북으로부터 일정거리 이상 떨어졌는지를 확인하고, 노트북에서 거리가 일정거리 이상 떨어졌으면 서버로 알림을 전송한다.
- 6) **서버에서 핸드폰에 알림 전송**: iBeacon으로부터 산출된 거리 값이 일정 구간 이상일때, 서버에서 핸드폰에 Slack으로 알림을 Push한다. 알림에는 이미지, 시간, 정보가

포함된다.

7) 서버에서 노트북으로 알림 전송: IBeacon으로부터 산출된 거리 값이 일정 구간 이상일때, 서버에서 노트북으로 정보를 보내서 노트북 윈도우가 자동으로 잠금상태가 된다.

8) 서버에서 라즈베리파이로 알림 전송: 알림이 전송되면 버저를 울린다.

```
[12/Jun/2019 12:34:10] "POST /rssi/ HTTP/1.1" 200 2
{'beacons': [{'beaconAddress': 'DE:0E:3D:65:6F:87', 'beaconType': 'eddystone_uid', 'distance': 5.170522152558649, 'eddystoneUidData': {'instanceId': '0xde0e3d656f87', 'namespaceId': '0xeddd1e5defa99'}, 'hashCode': 700165574, 'isBlocked': False, 'lastMinuteSeen': 26005714, 'lastSeen': 1560342847660, 'manufacturer': 65194, 'rssi': -96, 'telemetryData': {'batteryMilliVolts': 2610, 'pduCount': 4749048, 'temperature': 23.75, 'uptimeSeconds': 41800560, 'version': 0}, 'txPower': -68}, {'beaconAddress': 'CA:B3:08:6A:EE:D3', 'beaconType': 'ibeacon', 'distance': 3.729799488391219, 'hashCode': -888544150, 'ibeaconData': {'major': '61139', 'minor': '2154', 'uuid': 'b9407f30-f5f8-466e-aff9-25556b57fe6d'}, 'isBlocked': False, 'lastMinuteSeen': 26005714, 'lastSeen': 1560342849791, 'manufacturer': 76, 'rssi': -99, 'txPower': -74}, {'beaconAddress': 'D6:01:91:66:A6:0B', 'beaconType': 'ibeacon', 'distance': 6.659927365338656, 'hashCode': -645536256, 'ibeaconData': {'major': '42507', 'minor': '37222', 'uuid': 'b9407f30-f5f8-466e-aff9-25556b57fe6d'}, 'isBlocked': False, 'lastMinuteSeen': 26005714, 'lastSeen': 1560342849794, 'manufacturer': 76, 'rssi': -97, 'txPower': -66}], 'reader': 'hohocha'}
<QuerySet [(<Post_rssi: {'beacons': [{'beaconAddress': 'DE:0E:3D:65:6F:87', 'beaconType': 'eddystone_uid', 'distance': 5.170522152558649, 'eddystoneUidData': {'instanceId': '0xde0e3d656f87', 'namespaceId': '0xeddd1e5defa99'}, 'hashCode': 700165574, 'isBlocked': False, 'lastMinuteSeen': 26005714, 'lastSeen': 1560342847660, 'manufacturer': 65194, 'rssi': -96, 'telemetryData': {'batteryMilliVolts': 2610, 'pduCount': 4749048, 'temperature': 23.75, 'uptimeSeconds': 41800560, 'version': 0}, 'txPower': -68}, {'beaconAddress': 'CA:B3:08:6A:EE:D3', 'beaconType': 'ibeacon', 'distance': 3.729799488391219, 'hashCode': -888544150, 'ibeaconData': {'major': '61139', 'minor': '2154', 'uuid': 'b9407f30-f5f8-466e-aff9-25556b57fe6d'}, 'isBlocked': False, 'lastMinuteSeen': 26005714, 'lastSeen': 1560342849791, 'manufacturer': 76, 'rssi': -99, 'txPower': -74}, {'beaconAddress': 'D6:01:91:66:A6:0B', 'beaconType': 'ibeacon', 'distance': 6.659927365338656, 'hashCode': -645536256, 'ibeaconData': {'major': '42507', 'minor': '37222', 'uuid': 'b9407f30-f5f8-466e-aff9-25556b57fe6d'}, 'isBlocked': False, 'lastMinuteSeen': 26005714, 'lastSeen': 1560342849794, 'manufacturer': 76, 'rssi': -97, 'txPower': -66}], 'reader': 'hohocha'})]>
```

사진 0) django 서버에 전송받은 전체 ibeacon의 신호

Home > Post_Rssi > Post_rssis

Select post_rssi to change ADD POST_RSSI +

Action: 0 of 1 selected

☐ POST_RSSI

☐ {'beacons': [{'beaconAddress': 'DE:0E:3D:65:6F:87', 'beaconType': 'eddystone_uid', 'distance': 5.170522152558649, 'eddystoneUidData': {'instanceId': '0xde0e3d656f87', 'namespaceId': '0xeddd1e5defa99'}, 'hashCode': 700165574, 'isBlocked': False, 'lastMinuteSeen': 26005714, 'lastSeen': 1560342847660, 'manufacturer': 65194, 'rssi': -96, 'telemetryData': {'batteryMilliVolts': 2610, 'pduCount': 4749048, 'temperature': 23.75, 'uptimeSeconds': 41800560, 'version': 0}, 'txPower': -68}, {'beaconAddress': 'CA:B3:08:6A:EE:D3', 'beaconType': 'ibeacon', 'distance': 3.729799488391219, 'hashCode': -888544150, 'ibeaconData': {'major': '61139', 'minor': '2154', 'uuid': 'b9407f30-f5f8-466e-aff9-25556b57fe6d'}, 'isBlocked': False, 'lastMinuteSeen': 26005714, 'lastSeen': 1560342849791, 'manufacturer': 76, 'rssi': -99, 'txPower': -74}, {'beaconAddress': 'D6:01:91:66:A6:0B', 'beaconType': 'ibeacon', 'distance': 6.659927365338656, 'hashCode': -645536256, 'ibeaconData': {'major': '42507', 'minor': '37222', 'uuid': 'b9407f30-f5f8-466e-aff9-25556b57fe6d'}, 'isBlocked': False, 'lastMinuteSeen': 26005714, 'lastSeen': 1560342849794, 'manufacturer': 76, 'rssi': -97, 'txPower': -66}], 'reader': 'hohocha'}

1 post_rssi

사진 1) django 서버에 저장되는 전체 ibeacon의 신호

사진 1에서 보듯 데이터가 정제되지 않고 보내지기 때문에 mac주소와 uuid값으로 비교를 해서 필요한 자료를 필터링해서 이용한다.

Select post_rssi to change ADD POST_RSSI +

Action: 0 of 1 selected

☐ POST_RSSI

☐ {'beacons': [{'beaconAddress': '18:62:E4:3D:F7:00', 'beaconType': 'ibeacon', 'distance': 1.0432682273472875, 'hashCode': 921005105, 'ibeaconData': {'major': '4660', 'minor': '64001', 'uuid': '74278bda-b644-4520-8f0c-720eaf059935'}, 'isBlocked': False, 'lastMinuteSeen': 26005727, 'lastSeen': 1560343637989, 'manufacturer': 76, 'rssi': -59, 'txPower': -59}], 'reader': 'hohocha'}

1 post_rssi

사진 2) 사진1에서의 정제되지 않은 신호를 mac주소와 uuid값을 이용해 필터링 한 후의 데이터


```

C:\Users\Ishanga\Desktop\NP_lastproject_2019\client server (master -> origin)
λ python server.py
Server starts: ('192.168.0.27', 5555)
Client connecting: ('192.168.0.27', 51660)
('192.168.0.27', 51660):{'method': 'POST', 'deviceid': 'Windows_10', 'msgid': 1, 'data': {'lock': True, 'fLock': True, 'image': 0}}
Windows is locked!
Logon failed detected!
{"status": "OK", "deviceid": "Windows_10", "msgid": 1, "activate": false}
('192.168.0.27', 51660):{'method': 'POST', 'deviceid': 'Windows_10', 'msgid': 2, 'data': {'lock': True, 'fLock': True, 'image': 0}}
Windows is locked!
Logon failed detected!
{"status": "OK", "deviceid": "Windows_10", "msgid": 2, "activate": false}
('192.168.0.27', 51660):{'method': 'POST', 'deviceid': 'Windows_10', 'msgid': 3, 'data': {'lock': True, 'fLock': True, 'image': 0}}
Windows is locked!
Logon failed detected!
{"status": "OK", "deviceid": "Windows_10", "msgid": 3, "activate": false}
('192.168.0.27', 51660):{'method': 'POST', 'deviceid': 'Windows_10', 'msgid': 4, 'data': {'lock': True, 'fLock': True, 'image': 0}}
Windows is locked!
Logon failed detected!
{"status": "OK", "deviceid": "Windows_10", "msgid": 4, "activate": false}
('192.168.0.27', 51660):{'method': 'POST', 'deviceid': 'Windows_10', 'msgid': 5, 'data': {'lock': True, 'fLock': True, 'image': 0}}
Windows is locked!
Logon failed detected!
{"status": "OK", "deviceid": "Windows_10", "msgid": 5, "activate": false}

C:\Users\Ishanga\Desktop\NP_lastproject_2019\client server (master -> origin)
λ python client.py
{'lock': False, 'fLock': False, 'image': 0}
4798
4672
4799
5379
5382
4625
Login Failed Detected!
4672
[ WARN:0] terminating async callback
4634
Windows Locked!
  
```

사진 3) 노트북과 서버 간의 통신

3.3 흐름도

- 1-1) 일정 거리 이상으로 사용자가 멀어지지 않을 경우 계속 거리를 측정한다.
- 1-2) 일정 거리 이상으로 사용자가 멀어지게 되면 노트북으로 신호를 준다.
1. 1-2에서의 일정거리 이상으로 사용자가 멀어지게 되면 보내는 신호를 받으면 노트북으로 신호를 준다.
 - 2-1) 다른 사람이 노트북을 건드리지 않을 경우 사용자와 노트북 사이의 거리를 계속 측정한다.
 - 2-2) 다른 사람이 노트북을 건드릴 경우 노트북에서 서버로 신호를 보낸다. 노트북에서 카메라로 사람을 찍고 django서버로 신호를 보낸다.

2. django에서 데이터를 받은 후 buzzer를 울리고 slack으로 알림을 준다.
3. web화면에 관리자 창으로 상황을 띄워준다.
4. 사용자가 다시 가까워지면 노트북의 lock이 풀린다.
3에서 4번과정으로 넘어갈 때 서버에서 web으로 가는 과정에서 restful framework으로 serializefh 보낸다.web에서는 url로 받아(fetch) 화면에 보여준다.

3.4 소프트웨어 및 하드웨어

3.4.1 Django



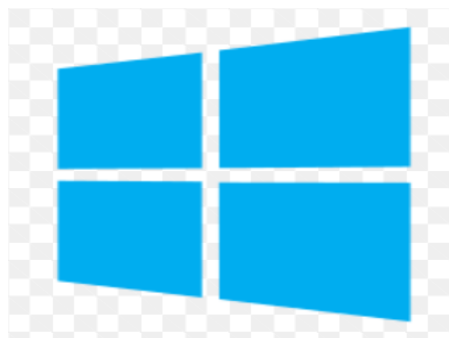
Django는 파이썬으로 만들어진 무료 오픈소스 웹 어플리케이션 프레임워크이다.

3.4.2 AWS



AWS는 Amazon Web Service의 약자로 클라우드 기반 서비스를 말한다.

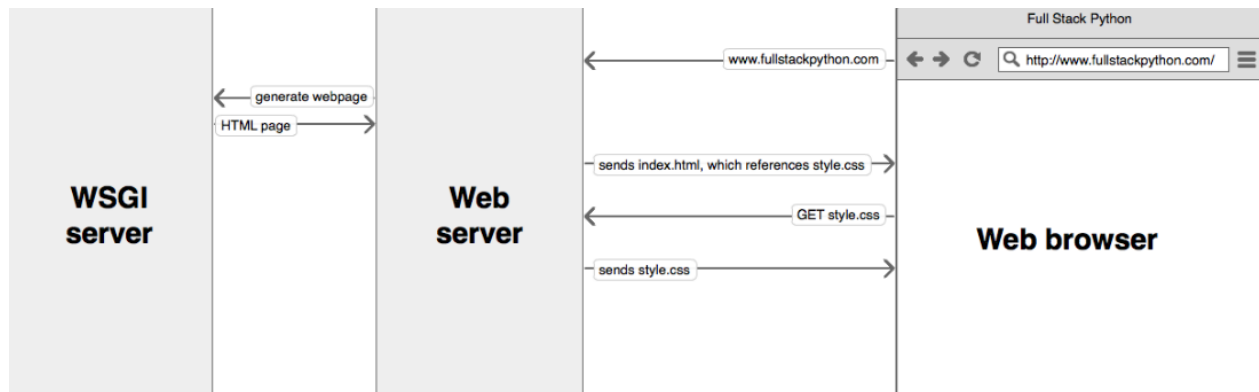
3.4.3 Window System Command Event Log



Window system command event log는 마이크로소프트 윈도 운영 체제들이 사용하는 API이다. C/C++ 프로그램에서 직접 운영 체제와 상호 작용할 수 있도록

만들어졌으며, 그보다 더 낮은 수준의 제어는 Ntdll.dll을 사용한 낮은 수준의 DLL로 가능하다.

3.4.4 WSGI



WSGI(Web Server Gateway Interface)는 웹서버와 파이썬을 사용한 웹 어플리케이션 개발환경 간의 인터페이스에 대한 규칙이다.

3.4.5 POSTMAN



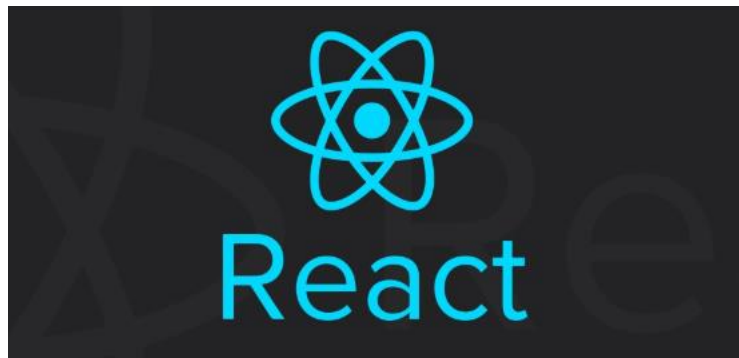
Postman은 개발한 API를 테스트하고, 테스트 결과를 공유하여 API 개발의 생산성을 높여주는 플랫폼입니다.

3.4.6 Slack



Slack은 2013년 8월에 출시된 클라우드 기반 협업 툴이다.

3.4.7 React



React는 facebook에서 개발한 오픈 소스 자바스크립트 UI 라이브러리이다.

3.4.8 iBeacon



ibeacon은 iOS7과 함께 도입한 기술로 블루투스 저전력 기술(BLE: Bluetooth Low Energy)을 활용한 근거리 데이터통신기술이다.

3.4.9 Buzzer



전자석과 진동편을 조합해서 신호음(信號音)을 발생시키거나, 직류전원으로부터 교류를 발생시키는 장치이다.

3.4.10 AWS



HM - 10 모듈은 블루투스를 이용하여 다양한 기기와 통신 할 수 있게 도와주는 모듈이다. 블루투스란 휴대폰, 노트북 등의 휴대기기를 서로 연결해 정보를 교환하는 근거리 무선 기술 표준을 말한다. 이번 프로젝트에서는 ibeacon모드로 전환하여 이용한다. Ibeacon이란 실내/외에서 수신기와의 거리를 5cm의 오차범위로 파악 할 수 있고 작은 데이터도 보낼 수 있는 방식이다.

3.4.2 Arduino



마이크로 컨트롤러를 내장한 기기 제어용 기판으로 회로기판에서 다양한 센서나 부품 등 디바이스를 연결할 수 있다. 본 프로젝트에서는 사용자와 사용자의 노트북 사이에서 ibeacon을 이용해 RSSI값을 거리로 환산해 알려주는 역할을 수행시키기 위해 제어하는 역할을 한다.

4. 요구사항 및 제한사항

본 장에서는 bluetooth 4.0의 ibeacon모드를 이용한 실시간으로 사용자와 노트북간의 거리 파악에서의 요구사항과 요구사항 구현에서의 한계점과 제한사항에 대해 기술한다.

4.1 사용자 인터페이스 기능

사용자 인터페이스

1. 아두이노를 노트북 주변에 둔다.
2. 사용자가 핸드폰으로 실시간으로 사용자와 노트북간의 거리 데이터를 파악할 수 있다.
3. Web을 이용해서 노트북이 사용자 이외의 사람에 의해 작동될 경우 해당 사람에 대한 사진을 받아볼 수도 있다.
4. 별 다른 특이사항이 없을 경우 다시 도착하여 노트북을 계속 이용하면 된다.

사용자 인터페이스 요구사항	
데이터 형식	텍스트, 이미지 등
사용 가능 기능	노트북과 사용자 간 거리 파악 및 식별되지 않은 사용자의 사진 정보 파악
사용 불가능 기능	노트북이 도난 당할 경우 노트북과 사용자 간의 방향정보

4.2 시스템 구현의 제한사항

-노트북과 사용자 간의 방향정보

노트북과 사용자 간의 거리 정보는 파악할 수 있지만 방향에 대한 정보를 파악 할 수 없었다.

5. 참고문헌

- 라즈베리파이 무선랜 접속 <https://webnautes.tistory.com/903>
- BLE 통신 및 라즈베리파이_ HM10 연결
<http://blog.naver.com/PostView.nhn?blogId=juke45ef&logNo=220834141429>
- 연결 구성도
<https://prgmr.tistory.com/2>
<https://blog.naver.com/zeta0807/221179622856>
- 라즈베리파이에서 블루투스 이용, 앱까지 연동
http://blog.daum.net/_blog/BlogTypeView.do?blogId=0ZPt7&articleno=2640&categoryId=7®dt=20180506215514
- 아두이노와 라즈베리파이 연결
<https://kocoafab.cc/tutorial/view/305>
<http://www.3demp.com/community/boardDetails.php?cbID=232>
- 소켓 통신
<https://webnautes.tistory.com/1110> (WeeESP8266)
<https://m.blog.naver.com/PostView.nhn?blogId=roboholic84&logNo=221139049600&proxyReferer=http%3A%2F%2Fwww.google.com%2Furl%3Fsa%3Dt%26rct%3Dj%26q%3D%26esrc%3Ds%26source%3Dweb%26cd%3D1%26ved%3D2ahUKEwicp9Xlzs7iAhXEbwKHaKACmQQFjAAegQIARAB%26url%3Dhttp%253A%252F%252Fm.blog.naver.com%252Froboholic84%252F221139049600%26usg%3DAOvVaw0J5xWx7mOOtIMQlqHYonJ5>
- 아두이노 블루투스 rssi
http://makeshare.org/bbs/board.php?bo_table=question&wr_id=1621
- hm 10으로 ibeacon 만들기
<https://bbangpan.tistory.com/17>
- 거리 측정
<https://github.com/Vinayrraj/Android-iBeacon-Demo/blob/master/android-ibeacon-service/src/com/radiusnetworks/ibeacon/IBeacon.java>
- RSSI, 거리공식 <https://blog.ghatdev.xyz/7> [Koding]
- 라즈베리파이- 안드로이드 연결 : <https://webnautes.tistory.com/995>
- 라즈베리파이- PC 연결: <https://webnautes.tistory.com/1137>
- 라즈베리파이- 아두이노 연결: <https://webnautes.tistory.com/979>
- 문제 해결
- 1. 라즈베리파이 코드52 해결
<https://notemylife.tistory.com/29>
- 2. 라즈베리파이 코드10 해결
<https://webnautes.tistory.com/902>
- 3.라즈베리파이에서 아두이노 깔때 dpkg 문제

<https://m.blog.naver.com/PostView.nhn?blogId=duwnsr&logNo=220463418670&proxyReferer=http%3A%2F%2Fwww.google.com%2Furl%3Fsa%3Dt%26rct%3Dj%26q%3D%26esrc%3Ds%26source%3Dweb%26cd%3D3%26ved%3D2ahUKEwi3zfeuxsziAhW1JaYKHfauA8wQFjACegQIARAB%26url%3Dhttp%253A%252F%252Fm.blog.naver.com%252Fduwnsr%252F220463418670%26usg%3DAOvVaw2oBPIdcoVLNhogDuRKFd6U>

6. [부록]