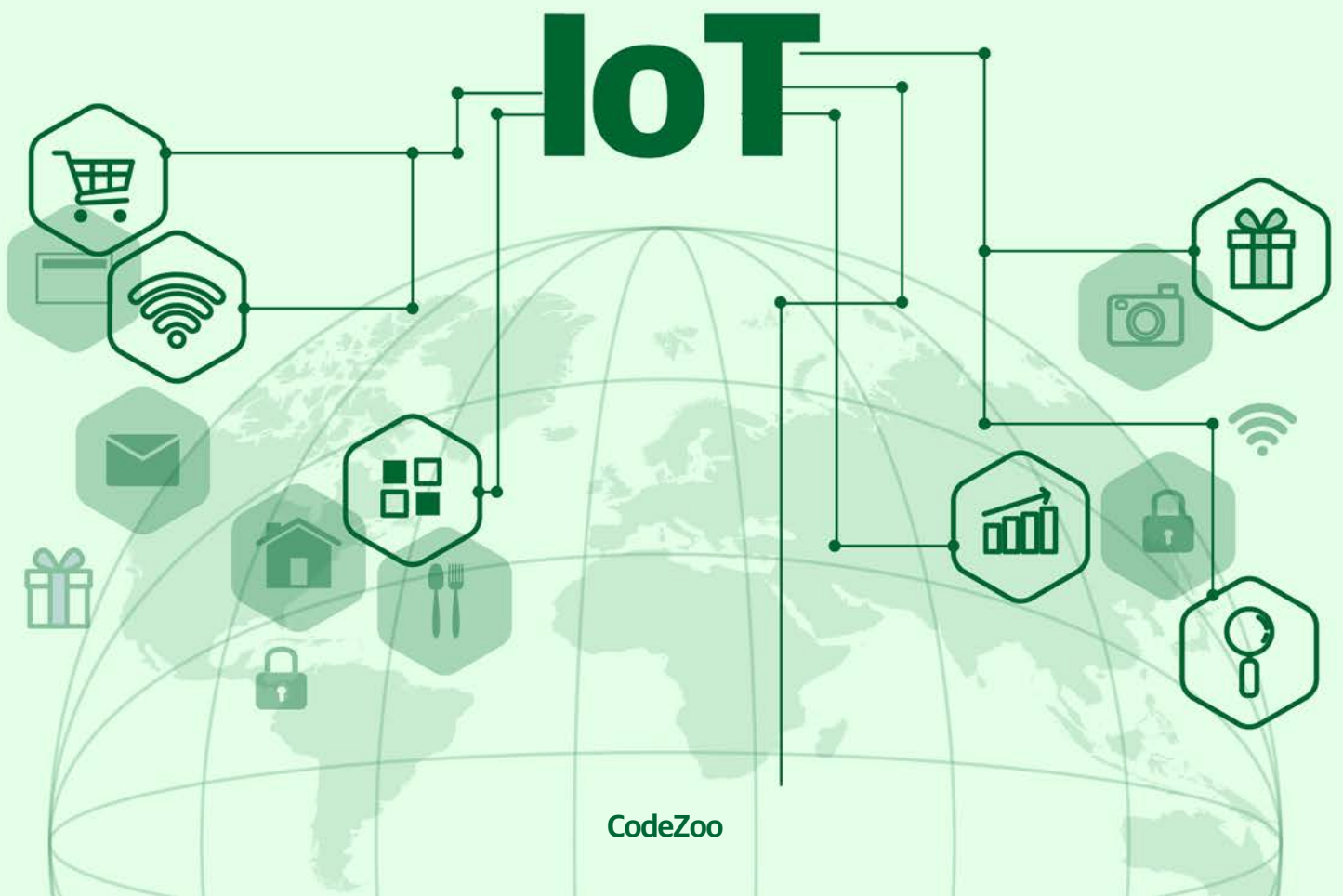


CodeZoo

NB-IoT

라즈베리파이 교재

IoT



1.

LG U+ NB-IoT 서비스 소개 자료

NB-IoT 라즈베리파이 교재

1

LPWA망의 이해 및 NB-IoT와 LoRa 비교

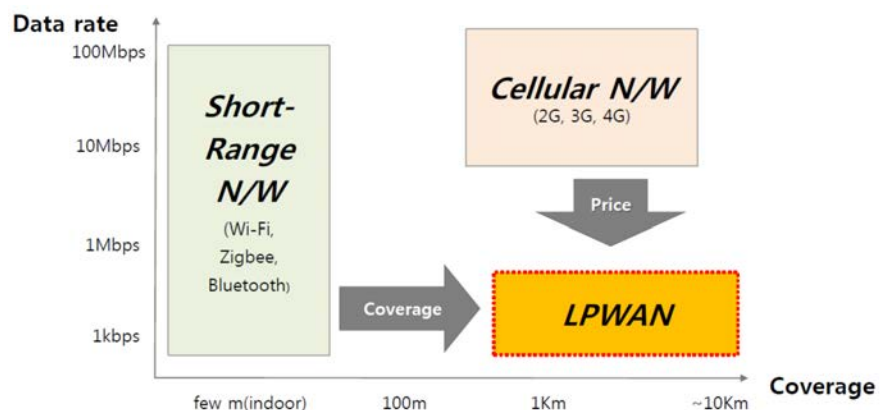
LPWA

: IoT 전용망 필요성

폭발적으로 증가하는 IoT 회선 수에 대응하기 위해, 데이터 전송 속도가 낮고 전력 소모량이 적으며 넓은 지역을 커버할 수 있는 LPWA* 기술이 필요

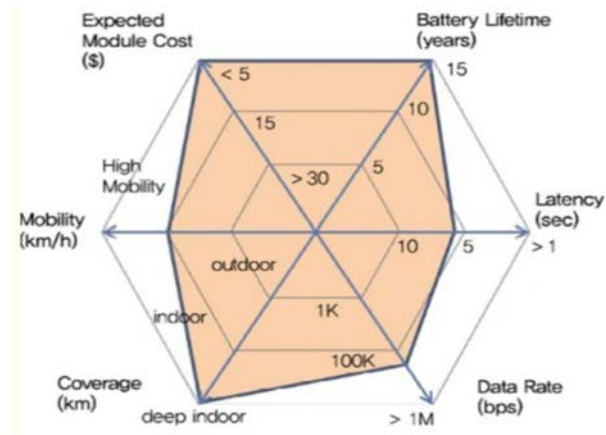
* LPWA : Low Power Wide Area

LPWA의 특징



- ① 넓은 커버리지
- ② 낮은 가격
- ③ 낮은 소비전력

LPWA의 핵심 요구사항



요구사항

저전력 소모 / Long battery life

저가 단말기 공급 / Low cost chipset

낮은 구축 비용 / Low cost Network

안정적 커버리지 / Wide area connectivity

대규모 단말기 접속

비고

10년

기기 당 \$5 이하

HW 추가 보다는 간단한 SW 업그레이드

빌딩 내, 지하, 외곽지역 등

동시 접속/수용 (10 만+ 디바이스 per cell)

LPWA

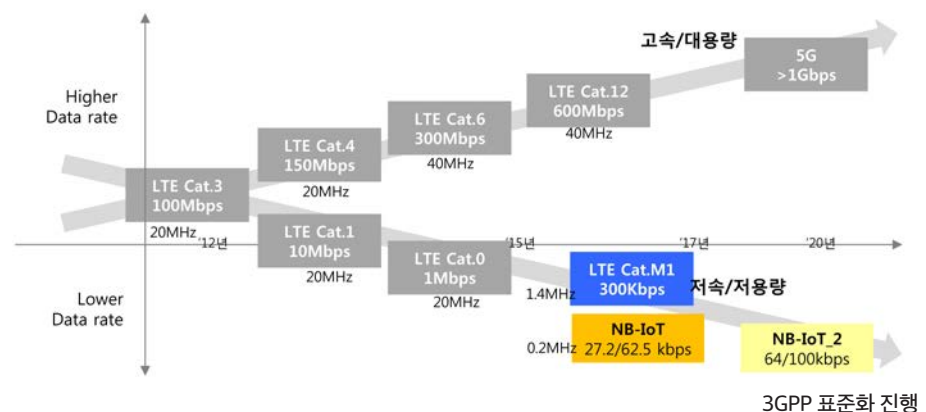
: 표준 기술의 진화 방향

무선 통신망 기술은 고속/대용량과 저속/소용량의 양방향으로 동시에 기술 발전이 진행되고 있음

▶ 고속/대용량 : 멀티미디어 서비스 → 4G, 5G

▶ 저속/소용량 : IoT 서비스 → LPWA (NB-IoT)

NB-IoT (Narrow Band - IoT) 특징



- 국제 이동통신 표준에서 정의한 저전력광역(LPWA : Low Power Wide Area)* 통신을 지원하는 사물인터넷 기술
- 기존 LTE 망과 호환성 보장, 면허대역 주파수 사용으로 안정적인 통신 환경 제공
- 저전력(수명 10년), 저속(단순 Thing 정보 획득 우위), 저가격(보급/확산 용이)

LPWA

: 비 면허 대역 통신의 종류 및 진화

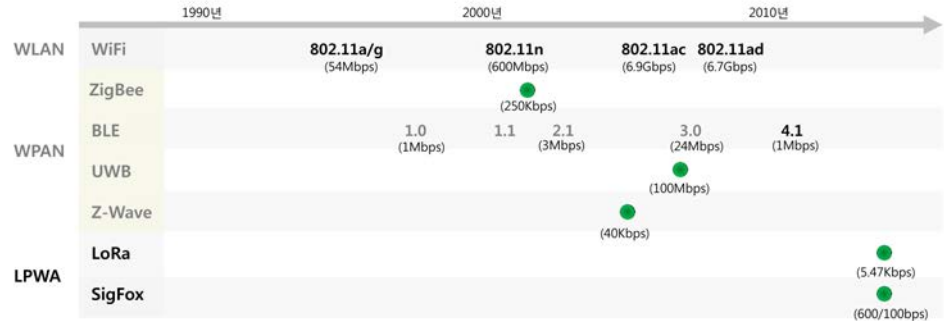
비면허 대역은 같은 주파수 대역 내에 여러 방식의 통신을 같이 사용하기 때문에 통신 기기 간 간섭 경감을 위하여 출력이 제한되어 단거리 통신 위주였으나, 최근 LoRa와 sigfox의 개발로 장거리용 통신 기술이 등장함

*WLAN : Wireless Local Area Network

*WPAN : Wireless Personal Area Network

*LPWA : Low Power Wide Area

NB-IoT (Narrow Band - IoT) 특징



LPWA

: 기술 비교

구분	LTE-MTC ^{주1}		NB-IoT	LoRa	SigFox
	Cat.1	Cat.M1			
표준화 일정	Rel-8 ('07)	Rel-13 ('16.3)	Rel-13 ('16.6)	비표준 (LoRa Alliance)	비표준 (ETSI)
사용 대역폭	20 MHz	1.4 MHz	200 kHz	125 kHz	100 kHz
최대 속도 (DL/UL)	10/5 Mbps	1/1 Mbps	27.2/62.5 kbps	5.47 kbps	600/100 bps
커버리지 (MCL ^{주2})	140.7 dB (20W)	155.7 dB (20W)	158.9 dB (6W)	157 dB (200mW)	154 dB (200mW)
서비스 지원 거리 ^{주3}	도심	0.7 km	1.8 km	2.1 km	1.8 km
	외곽	~4 km	~11 km	~13 km	~12 km
모뎀 복잡도 (Cat1 대비)	100%	25%	20%	~20%	~20%
수용 단말 수 (셀 당)	-	-	2,000~15,000	200~4,000	-
단말 배터리 수명 ^{주4}	~1년	~10년	~10년	~10년	~10년
망 구축 여부	기존 LTE망에서 지원	기존 LTE망에서 지원	기존 LTE망에서 지원	신규 구축 필요	신규 구축 필요
IoT 서비스별 지원여부	웨어러블	○	△ (지연시간)	△ (지연시간)	△ (지연시간)
	의료진단	○	△ (지연시간)	△ (지연시간)	△ (지연시간)
	센서	X (단말 가격)	△ (단말 가격)	○	○
	모니터링	X (단말 가격)	△ (단말 가격)	○	○
	트래킹	○	△	△	△
	검침	X (커버리지)	△ (단말 가격)	○	○
	주차	○	○	○	○
기기제어	△ (배터리 수명)	△ (배터리 수명)	△ (배터리 수명)	△ (배터리 수명)	△ (배터리 수명)

주1) MTC : Machine Type Communication(=M2M)

주2) MCL : Maximum Coupling Loss

주3) Hata model for urban areas(+20dB offset)/open areas 기준

주4) 2xAA/1.5V/3000mAh 배터리로 하루에 한 번 트래픽 전송 가정 (PSM: Power Save Mode 기준)

NB-IoT vs LoRa 비교

NB-IoT는 비표준 대역 통신 기술을 사용하는 LoRa 대비

- ① 실시간 상태정보 조회 ② 역제어 ③ FOTA(Firmware Over The Air) 지원
- ④ 보안성 측면에서 우수

차이점

1 기지국 구축

- 『NB-IoT』는 “기존 LTE기지국/중계기”를 사용하여, “현재 LTE커버리지 = IoT 커버리지” 이나,
- 『LoRa』는 “기지국/중계기(DCU)”를 신규/추가설치한 후에, “IoT 커버리지”를 확보(커버리지 비균질)해야 해서 품질 문제, 투자 비용, 구축 기간 의 문제가 발생할 수 있습니다.

2 확장, 유지보수

- 『NB-IoT』 방식은 국제 무선표준(3GPP) 적용으로 “서비스 확장성” 및 “단가 인하소지”가 큼니다.
- 『LoRa』 방식은 “단일 LoRa Chip제조사”인 미국SEMTECH社에서 주도하는 “Alliance회원사간 표준”으로한번 설치 후 최소 8~10년간 유지보수되어야 할 IoT의 기술지원 지속Risk가 있습니다.
- 향후, 서비스확장을 고려 시 LoRa 대비 5배 이상 속도가 빠른 NB-IoT가 유리합니다.

* FOTA : Firmware Over the Air
(SW 원격 업데이트 기능)

비교표

구분		NB-IoT	LoRa
서비스 품질	커버리지	현재 LTE기지국 활용	기지국 추가 구축 필요
	신호 간섭	검증된 면허대역	비면허 대역 (무선마이크/RFID 중첩)
	양방향(2way), 통신지원	가능	제한적
	속도(Kbps) (다운로드/업로드)	27.2/62.5 kbps	5.47kbps
보안	보안 신뢰성	복제 불가 (USIM)	개발 검토 중
	기업 고객전용망 서비스	가능	TBD
확장 가능성	국제 표준	3GPP	Alliance
	원격 업데이트(FOTA*)	지원	TBD

LPWA 보안기술의 차이

NB-IoT가 보안측면에서 우수한 평가 획득 (2017. 5.2, GSMA 보안성 비교 백서 중)

LPWA 기술별 보안 수준 평가 결과표

	LTE-M	NB-IoT	EC-GSM-HoT	LoRaWAN	Sigfox
Smart Pallet	Good	Good *	Adequate	Good	Poor
Smart Agriculture	Good	Good	Good	Adequate	Adequate
Smart Street Lighting	Adequate	Good *	Adequate	Adequate *	Adequate
Water Metering	Adequate *	Good *	Adequate *	Adequate	Poor
Domestic Smoke Detectors	Good	Good	Good	Adequate	Adequate

GSMA, 세부 항목별 평가 결과

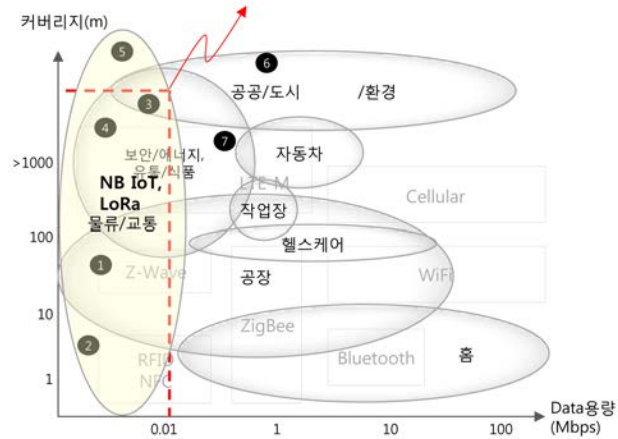
항목	NB-IoT	LoRaWAN	Sigfox
최대속도			
- 다운로드	250kbps	50kbps	600kbps
- 업로드	250kbps	50kbps	100kbps
인증방식	UICC/eUICC	기기/유지	기기
데이터 완전성	선택적	제한적	가변적
전송 신뢰성 확보 기능	O	X	X
업데이트 가능성(기기)	가능	제한적	X
네트워크 모니터링/필터링	O	제한적	모니터링only
알고리즘 교섭(선택)	O	X	X
보안침입 저항력	O	선택적	O
인증된 장비 사용	필수	선택적	필수

Global Carrier NB-IoT 도입 계획

	NB-IoT		LoRa
	도입 발표	시범서비스제공	
미주지역			
유럽 지역		   	   
아시아 지역	 	  	 
오세아니아	 		
아프리카			

NB-IoT 적용 서비스 구분

NW 통신기술별 적용 가능 서비스(예시적)



서비스 예시

- ① 수도, 전기, 가스 등 Utility 분야 원격 검침
- ② 공장 이동체(장비, 부품 등) 위치관제 및 예지 보전
- ③ 물류센터 화물추적 및 에너지효율(가로등), 보안
- ④ 공사장 환경 오염 측정 및 대기 질 관리
- ⑤ 식품원재료 신선도 관리(팩/박스 단위), 진열/재고관리
- ⑥ 팔레트/화물 유통 추적, 자산 관리 및 동산 담보
- ⑦ 도시간 전력량 분석 및 관제, 산림/화재 감시
- ⑧ 식자재 원산지 추적

2

NB-IoT 솔루션

LG U+ NB-IoT 솔루션

NB-IoT*)망을 통해 GPS(실외 지역)와 셀기반(실내) 대인/대물의 위치상태를 추적 가능한 서비스입니다.

* Narrow Band - Internet Of Things: 기존 LTE 망의 주파수 대역을 활용하여 소용량 데이터를 저속으로 전송하는 사물인터넷 표준기술



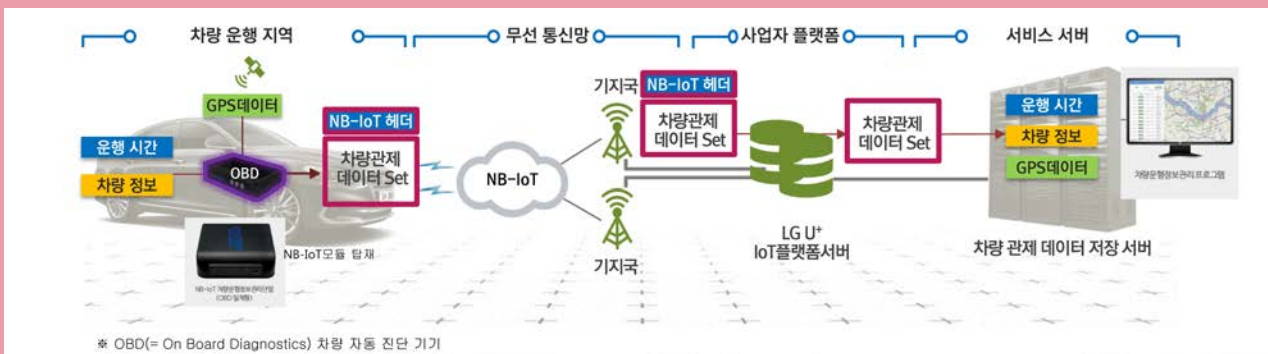
LG U+ NB-IoT 솔루션 개발 현황



LG U+ NB-IoT 솔루션 개발 현황_현대/기아차 개발 건

" NB-IoT 통신을 이용한 시험 차량 관제 서비스 제공 "

- 목적 : 현대차 그룹 자동차 연구소 시험 차량 및 업무용 차 운행 정보 관리
- 추출 데이터 : 차량운행 기록 거리, 운행 시간, 고장코드, 차량 위치 등 제공 (100Byte이내)
- 운영방안 : 주기적 차량 정보(위치+OBD정보) 전달, 서버에서 차량정보 요청 (일 수회 수준)
 - 위치정보 수집 : 이동 시 1분 단위로 수집하여 5분 이상 주기로 전송(전송 주기 변경 가능)
 - 차량 정보 : 운행 중 10분 단위로 차량 상태 정보 전송 (시동 OFF 시 1시간 주기로 전송)
 - 지정된 이벤트 발생시 비 주기 전송 가능



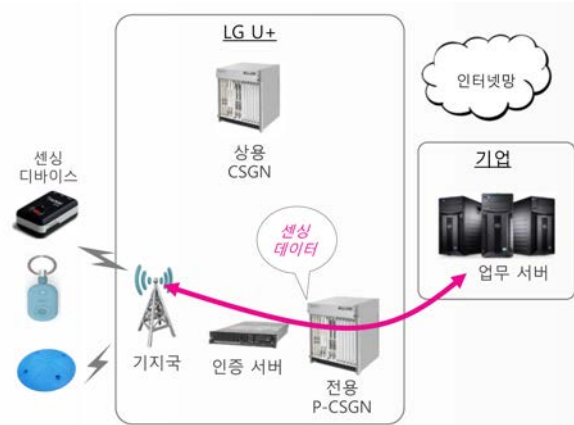
3

NB-IoT 고객전용망

NB-IoT 고객전용망

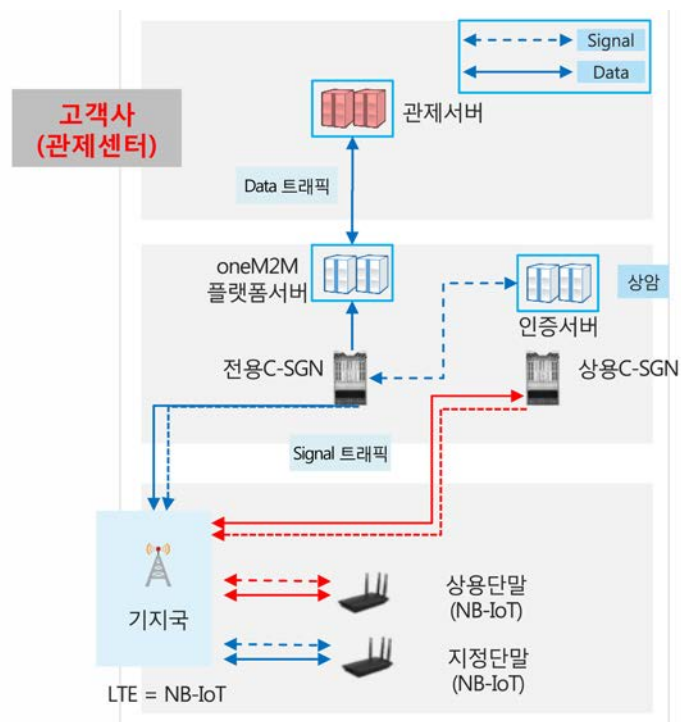
고객사가 지정한 단말만 사용할 수 있도록 무선 전용망을 구성 상용망 분리로 “보안성”을 강화하고 “업무 확장성” 제공

P-NB IoT망 (센서,상태관리)



- 기존 P-LTE망 내에서 P-C-SGN추가를 통한 전용 NB-IoT 망 제공 가능 (IoT 단말 제어를 위한 Signaling만 처리)
- 망 분리로 보안성 확보,해킹 및 데이터 유출 차단

NB-IoT망 시그널링 및 데이터 흐름



2.

CodeZoo NB-IoT 개발보드 소개

NB-IoT 라즈베리파이 교재

NB-IoT 개발보드 제작을 마치고 외부에 소개할 때 가장 많이 들었던 질문은 NB-IoT 통신이 무엇인지와 와이파이, 블루투스와의 차이를 알려달라는 내용이었습니다.

그때 설명드렸던 내용을 인용하면 아래와 같습니다.

- ① NB-IoT는 LTE 전화기와 같은 LTE망을 이용해서 통신 합니다.
- ② LTE 전화기로 통화가 가능한 곳이라면, NB-IoT를 이용해서 통신이 가능합니다. 인터넷 전용선, 무선공유기와 같은 통신 설비가 필요 없습니다. NB-IoT 제품과 배터리만 있으면 됩니다.
- ③ 전화기와 마찬가지로 유심(USIM)을 사용 합니다. 유심이 없으면 통신을 할 수 없습니다.

개발자 또는 메이커분들에게는 아래와 같은 내용을 추가로 전달 했습니다.

- ① NB-IoT는 내부에서 통신소켓(Socket)을 열어서 UDP/TCP 통신이 가능합니다.
- ② 따라서 UDP/TCP 서버를 열고 접속 가능한 아이피만 제공하면 NB-IoT와 연결해서 인터넷 통신이 가능합니다.

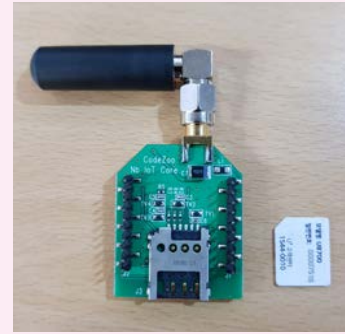
먼저 CodeZoo NB-IoT 개발보드의 하드웨어를 소개 합니다.

NB-IoT 개발보드는 크게 3부분으로 구성 됩니다.

① NB-IoT 코어모듈



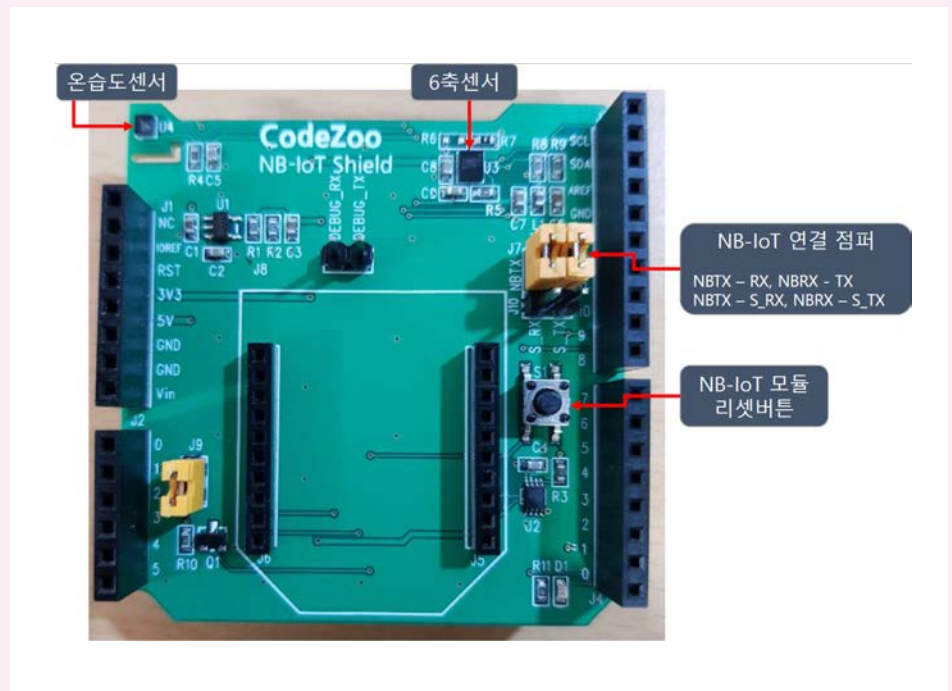
코어모듈 전면부



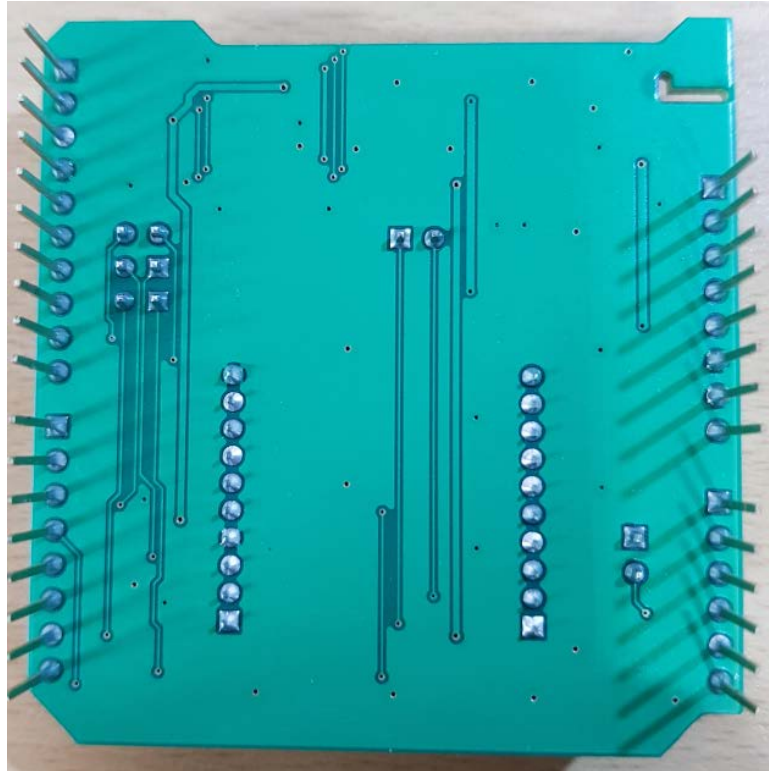
코어모듈 후면부

코어모듈은 Shield 연결보드와 Hat 연결보드에 공통으로 사용할 수 있는 Xbee 폼 팩터로 만들었습니다. 전면부에는 NB-IoT 통신모듈이 장착되어 있고, 후면부에는 유심카드 슬롯이 있습니다. 코어모듈 상단에는 LTE 신호를 수신하기 위한 안테나가 장착되어 있습니다.

② NB-IoT Shield 보드



Shield 보드 전면부



Shield 보드 후면부

шил드(Shield) 보드는 위 사진과 같이 온습도 센서(HTS221)와 6축센서(LSM6DS3)가 장착되어 있고 NB-IoT코어모듈과 연결해서 사용 합니다. 실드 확장핀이 있는 아두이노, ARM MbedOS 보드에 연결해서 사용할 수 있습니다. 테스트는 Arduino UNO 보드와 ARM MbedOS B-L475E-IOT01A1 보드로 진행 했습니다.

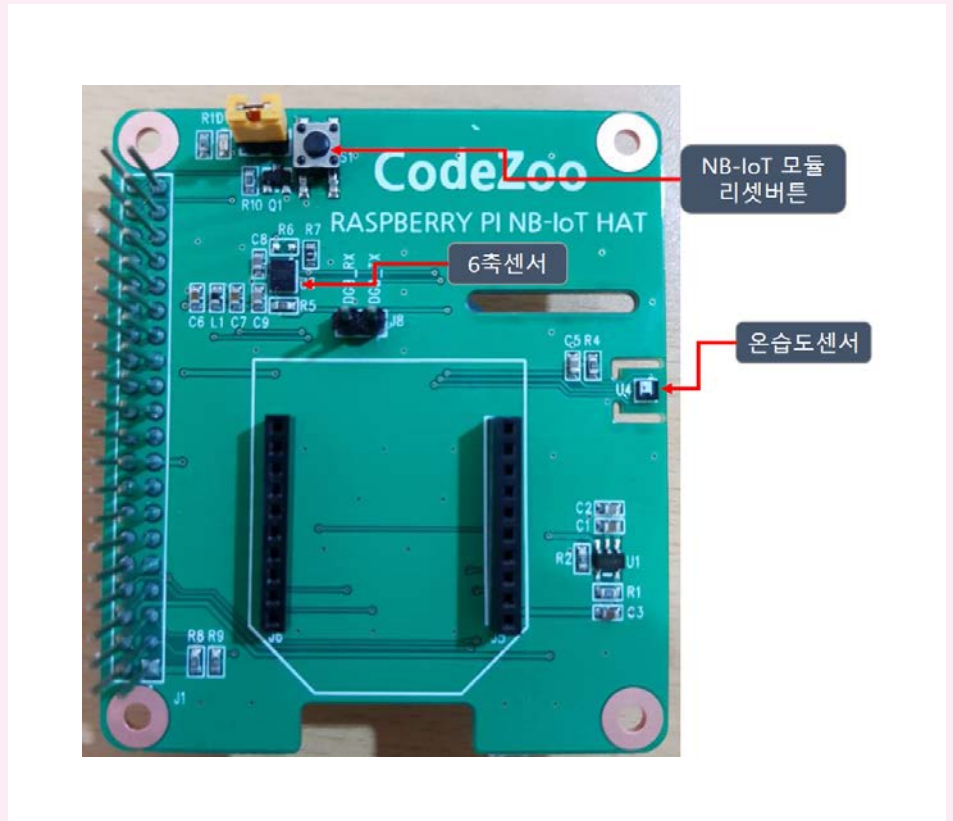


NB-IoT 실드 + 아두이노 우노보드

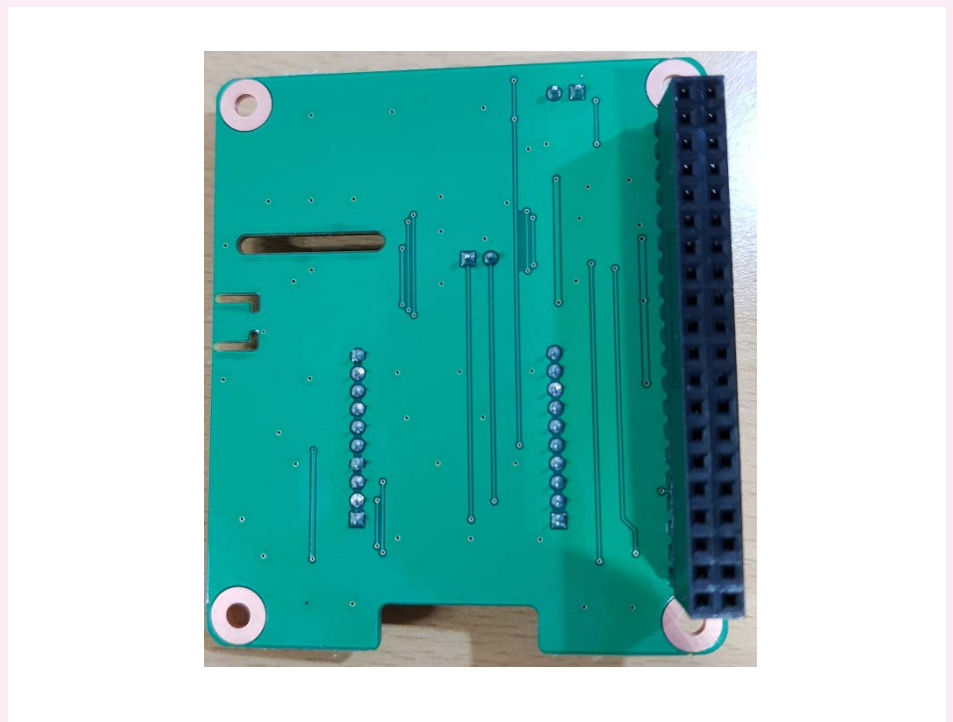


NB-IoT 실드 + B-L475E-IOT01A1 보드

③ 라즈베리파이 HAT 보드



Hat 보드 전면부



Hat 보드 후면부

햇(Hat) 보드는 쉴드 보드와 마찬가지로 온습도 센서(HTS221)와 6축센서 (LSM6DS3)가 장착되어 있고 NB-IoT코어모듈과 연결해서 사용 합니다. 라즈베리 파이 보드와 연결은 40핀 확장핀을 사용 합니다. 테스트는 Raspberry Pi 3(Model B)에서 진행 했습니다.



NB-IoT 햇 + 라즈베리파이 3 Model B

NB-IoT 모듈(SERCOMM TPB23) 스펙

CodeZoo NB-IoT보드의 통신을 담당하는 핵심 모듈 TPB23의 스펙은 아래와 같습니다.



Form Factor	<ul style="list-style-type: none"> 20x24mm, LGA 94-pad
Band	<ul style="list-style-type: none"> Multiple Band (3, 5, 8, 20...)
Protocol	<ul style="list-style-type: none"> 3GPP Release 13/Release 14
Interface	<ul style="list-style-type: none"> UART, I2C, GPIO, USIM, ADC, Antenna
Feature	<ul style="list-style-type: none"> PSM/DRX/eDRX IP/UDP/TCP/COAP/LWM2M DTLS/DTLS+ Cell Reselect DNS over UDP HTTP over TCP Non-IP

TPB23(<https://www.sercomm.com/contpage.aspx?langid=1&type=prod3&L1id=2&L2id=1&L3id=97&Prodid=701>)

3. NB-IoT 라즈베리파이에서 사용하기

NB-IoT 라즈베리파이 교재

① 라즈베리파이 설치 : 아래 주소를 참고하시기 바랍니다.

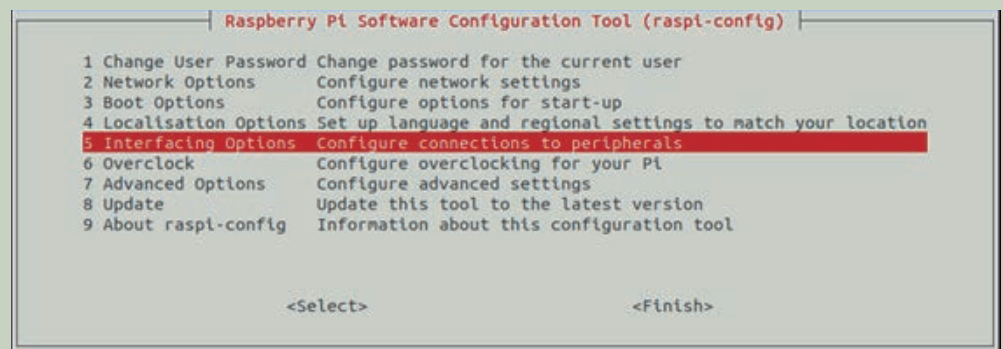
<https://www.raspberrypi.org/documentation/installation/installing-images/>

② I2C 설정하기 (※)I2C는 CodeZoo NB-IoT 보드에 내장된 센서(온습도, 6축)에 사용됩니다.

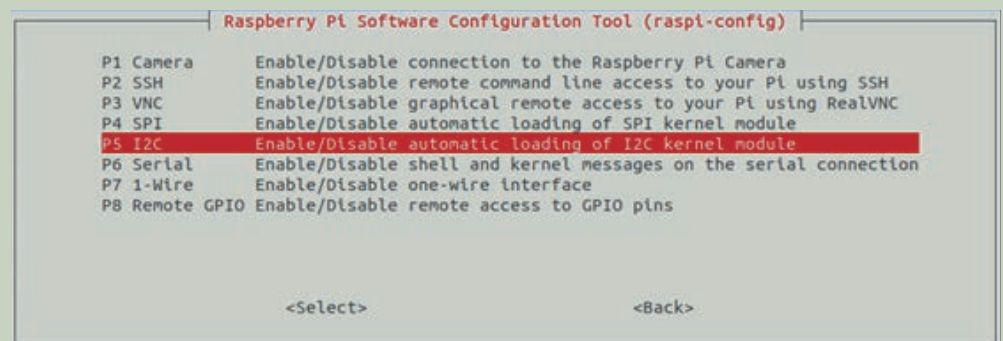
라즈베리파이 터미널에서

```
$ sudo raspi-config
```

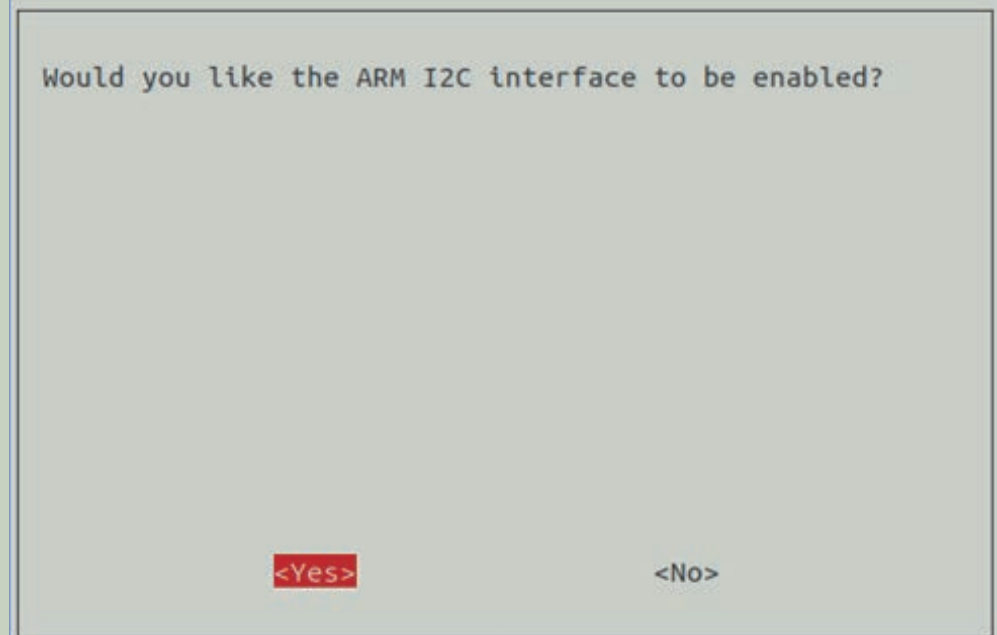
5. Interfacing Options 선택



P5 I2C 선택



Yes 선택



Would you like the ARM I2C interface to be enabled?

<Yes> <No>

OK 선택



The ARM I2C interface is enabled

<Ok>

"ESC" 나 "Finish"로 나감
라즈베리파이 다시 부팅

③ I2CTools 사용하기

라즈베리파이 터미널에서 다음 명령의 i2c tool설치 합니다

```
$ sudo apt-get install i2c-tools
```

라즈베리파이에 I2C장치를 연결하고 다음 명령으로 연결된 장치 검색 합니다

```
$ i2cdetect -y 1
```

```
pi@raspberrypi:~ $ i2cdetect -y 1
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- 5f
60:  -- -- -- -- -- -- -- -- 6a -- -- -- -- -- --
70:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
pi@raspberrypi:~ $
```

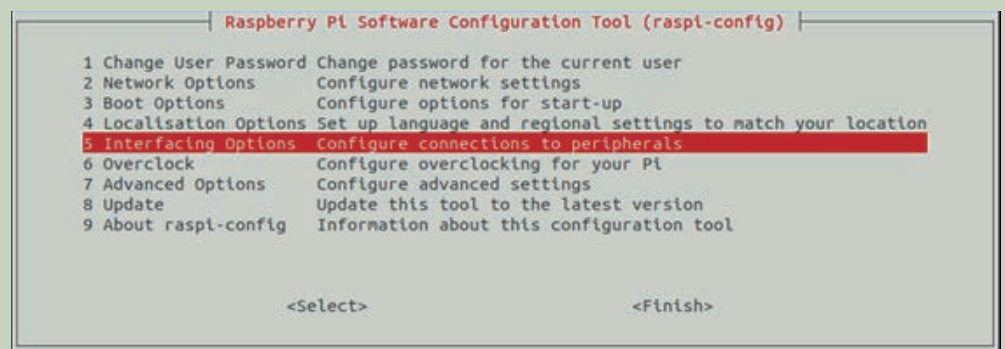
위와 같이 연결된 장치의 I2C address가 출력됨

④ 시리얼 포트 설정

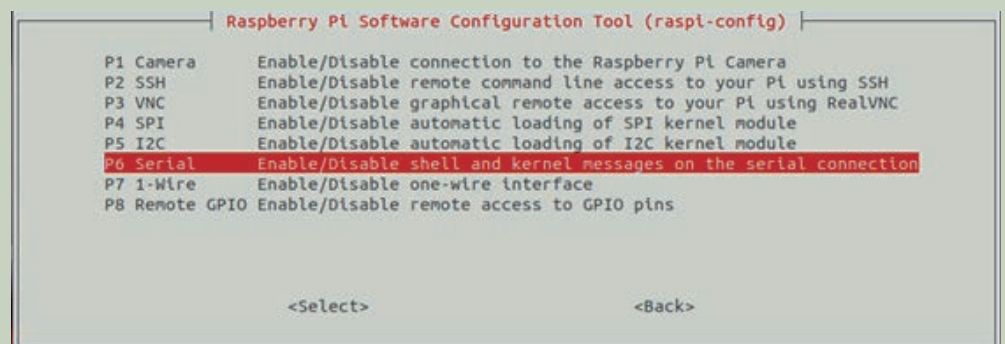
라즈베리파이 터미널에서

```
$ sudo raspi-config
```

5. Interfacing Options 선택



P6 Serial 선택



No 선택

Would you like a login shell to be accessible over serial?

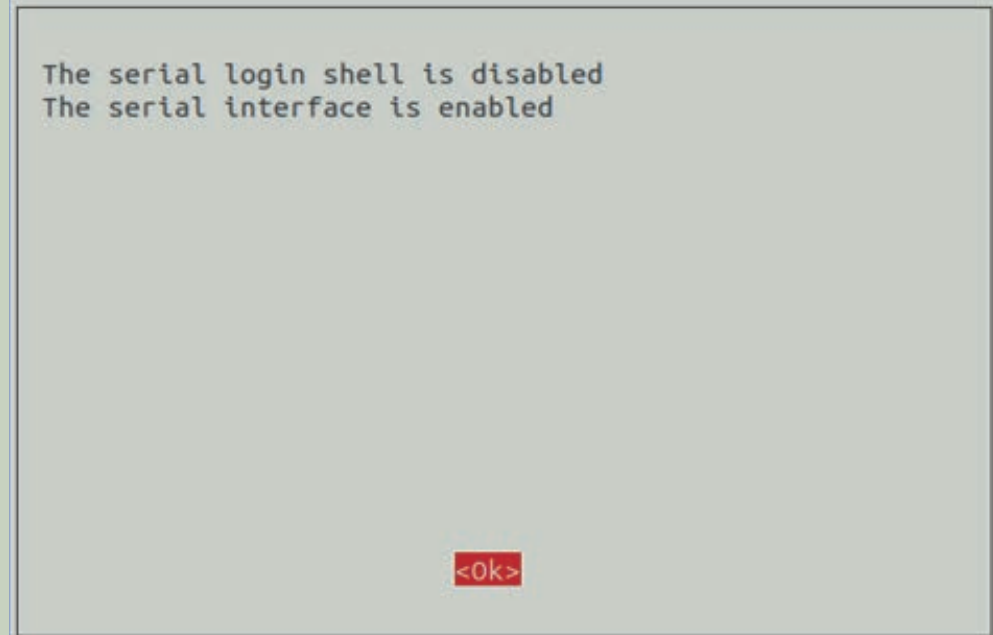
<Yes> **<No>**

Yes 선택

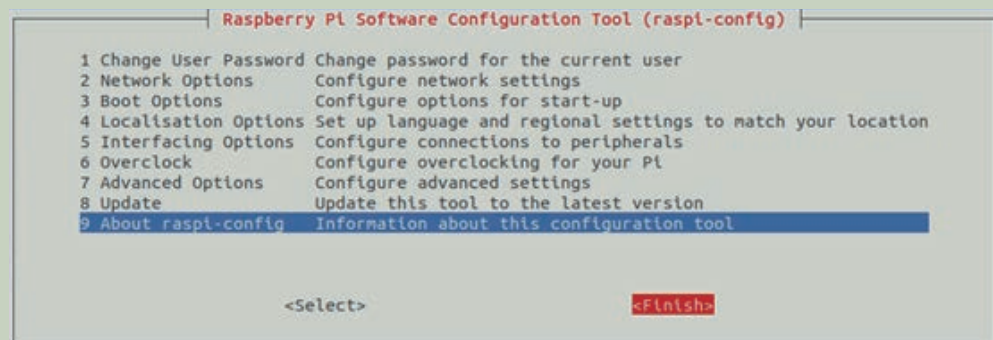
Would you like the serial port hardware to be enabled?

<Yes> <No>

OK 선택

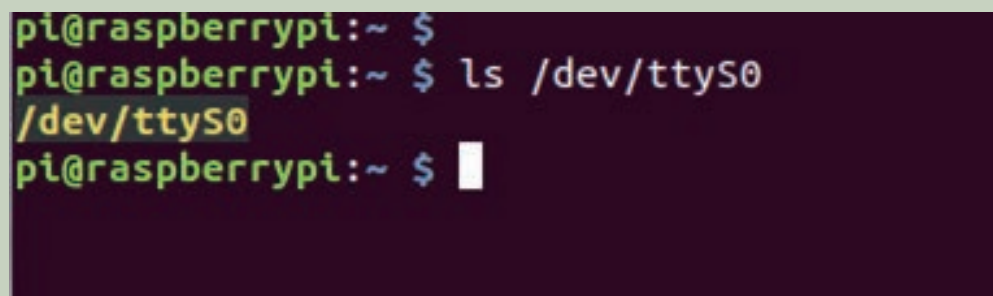


"ESC" 나 "Finish"로 나감



라즈베리파이 다시 부팅 합니다

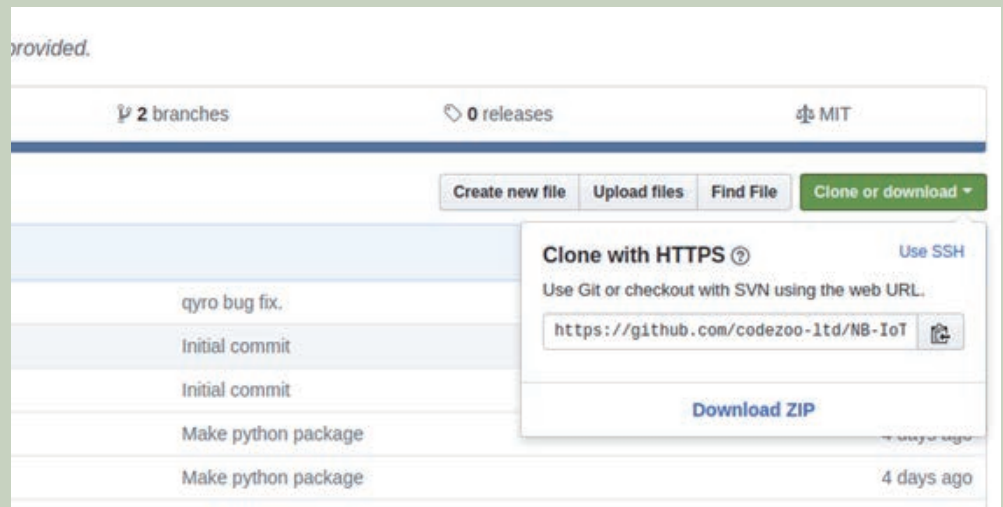
부팅후 "/dev/ttyS0" 파일이 생성 됩니다.



⑤ codezoo NB-IoT raspberrypi 소스코드 설치하기

https://github.com/codezoo-ltd/NB-IoT_Raspberrypi

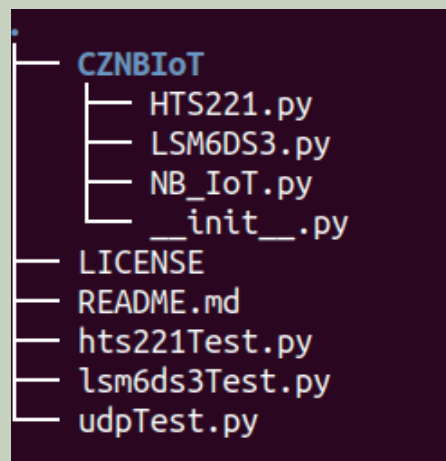
Download Zip 을 선택 합니다.



또는 라즈베리 터미널에서

```
$ git clone https://github.com/codezoo-ltd/NB-IoT_Raspberrypi.git
```

⑥ NB-IoT Raspberrypi 파일 디렉토리 구조 및 코드 실행하기



CZNBIoT(CodeZoo NB-IoT package)

HTS221.py 온/습도 센서 드라이버

LSM6DS3.py 6축 센서 드라이버

NB_IoT.py NB-IoT 드라이버

- hts221Test.py: 온/습도 센서 테스트 코드

- lsm6ds3Test.py: 6축 센서 테스트 코드

- udpTest.py: NB-IoT 테스트 코드

테스트 코드 실행하기

```
$ python3 hts221Test.py
```

```
$ python3 lsm6ds3Test.py
```

```
$ python3 udpTest.py
```

NB-IoT

class NB_IoT.NB-IoT

`__init__(serialPort='/dev/ttyS0', baudrate=9600, resetPinNum=4)`

serialPort: 시리얼 포트

baudrate: 시리얼 속도, NB-IoT 모뎀은 9600만 지원

resetPinNum: NB-IoT 모뎀의 리셋 핀 번호

`resetModem()`: NB-IoT 모뎀 리셋, 60초 이상 소요

`getResetPinNum()`: NB-IoT 모뎀의 리셋 핀 번호 반환

`getIMEI()`: IMEI값 반환

`getFirmwreInfo()`: NB-IoT 모뎀 F/W 정보 반환

`getHardwareInfo()`: NB-IoT 모뎀 H/W 정보 반환

`isAttachNetwork()`: NB-IoT 모뎀이 기지국과 연결되어 있는지 확인

True: 연결되어 있음

False: 연결되어 있지 않음

`attachNetwork(connect=True)`: NB-IoT 모뎀이 기지국과 연결 또는 연결 해제

connect : True 기지국 연결

False 기지국 연결 해제

return: True 기지국 연결 또는 연결해제 성공

False 기지국 연결 또는 연결해제 실패

`sendATCmd(command, cmd_response, timeout = None)`: NB-IoT 모뎀에 AT명령어 전송

command: 전송할 AT 명령어

cmd_response: 전송한 AT명령어에 대한 NB-IoT 모뎀의 리턴값

timeout: AT명령어 전송 후 NB-IoT 모뎀의 리턴값까지 기다리는 시간 (기본값 3초)

`openUDPSocket(port=10)`: udp socket 생성

port: NB-IoT 모뎀내 udp socket에 사용할 port번호 (기본값 10)

return: 생성한 socket

-1 socket 생성 실패

`closeUDPSocket(mySocket)`: 생성한 udb socket을 종료

`setIPAddress(ip)`: 데이터를 전송할 udp server의 ip

`setPortNum(port)`: 데이터를 전송할 udp server의 port

`sendUDPData(mySocket, data, ip_address=None, ip_port=None)`: server에 데이터 전송

mySocket: 생성한 udp socket

data: 전송할 데이터

ip_address: 데이터를 전송할 udp server의 ip 지정하지 않으면 setIPAddress()함수에서 지정한 ip 사용

ip_port: 데이터를 전송할 udp server의 port 지정하지 않으면 setPortNum()함수에서 지정한 port 사용

`receiveUDPData(mySocket, rev_length=256, rev_timeOut=3)`: udp server로 부터 데이터 수신

mySocket: 생성한 udp socket

rev_length: 수신 받은 데이터를 한번에 읽을 최대 크기 (기본값 256)

rev_timeout: udp server로 부터 데이터 수신을 위한 대기 시간 (기본값 3)

return: 리스트 자료형으로 데이터 반환

```
[
    'udp server ip', 'udp port number',
    'NB-IoT 모뎀에 수신된 데이터에서 읽은 데이터 길이',
    '읽은 데이터',
    'NB-IoT 모뎀에서 수신된 데이터중 읽고 남은 데이터의 길이'
]
```

HTS221

```
class HTS221.HTS221
getDeviceID(): 디바이스 ID 반환 (hts221 id: 0xBC)
getHumidity(): 습도 반환
getCTemperature(): 온도 반환 (섭씨)
getFTemperature(): 온도 반환 (화씨)
```

LSM6DS3

```
class LSM6DS3.LSM6DS3
getDeviceID(): 디바이스 ID 반환 (lsm6ds3 id: 0x69)
getXAxes(): 가속도 값 반환
    튜플 자료형으로 (x축, y축, z축)
getGAxes(): 자이로스코프 값 반환
    튜플 자료형으로 (x축, y축, z축)
```


“

감사의 글

CodeZoo NB-IoT 보드(코어 1종, 쉘드보드 1종, 햇보드 1종) 제작은 서울산업진흥원 산하 메이커스페이스 지캠프의 연결형 보드 플랫폼 개발사업으로 진행되었습니다. 국내에서도 해외에 판매되고 있는 오픈소스 기반의 NB-IoT Connectivity 모뎀보드를 만들고자 하는 여러 관계자분들의 뜻을 모아 사업이 진행되었으며 전세계 개발자들이 가장 많이 사용하고 있는 아두이노, 라즈베리파이, ARM MbedOS에서 각각 동작할 수 있도록 하드웨어 플랫폼과 라이브러리와 예제코드를 개발하였습니다. 해당 리소스는 github를 통해 제공되며 국내 개발자들이 짧은 시간내 NB-IoT 제품 프로토타입을 개발하는데 큰 도움을 줄 수 있을거라 생각 합니다.

사업을 기획하고 물심양면으로 도와주신 지캠프 관계자 분들과 NB-IoT 통신을 위한 핵심모듈 및 자료제공, 통신테스트를 지원해 주신 LGU+ 기업IoT 사업담당자분들에게 진심으로 감사 드립니다.

”