

# NexusMesh-X



## Built-in features

---

**Robot Control Protocol (RCP)**

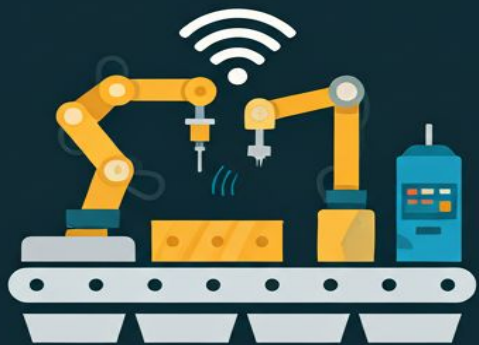
**Kernel module for check Wi-Fi health**

**Real-time monitoring tool of Wi-Fi status**

*Author: Park Jeong-Woo*  
*E-mail: [gjw1702@gmail.com](mailto:gjw1702@gmail.com)*  
*Tel: +82 10-8946-5291*

Intro

# What is the NexusMesh-X?



NexusMesh-X

NexusMesh-X는 스마트 팩토리 환경에서의 원활한 로봇 제어 및 통신을 위해 개발된 **Wi-Fi 게이트웨이(AP) 소프트웨어**로, 로봇과 컨베이어 시스템 간의 안정적이고 효율적인 Mesh 네트워크 통신을 지원합니다.

본 소프트웨어는 IEEE 802.11ac 표준을 따르는 **Wi-Fi5**를 기반으로 설계되었으며, 자체 프로토콜인 **RCP(Robot Control Protocol)**가 적용되어 **비동기 통신** 기반의 네트워크 성능 최적화와 통신 제어 기능을 제공합니다.



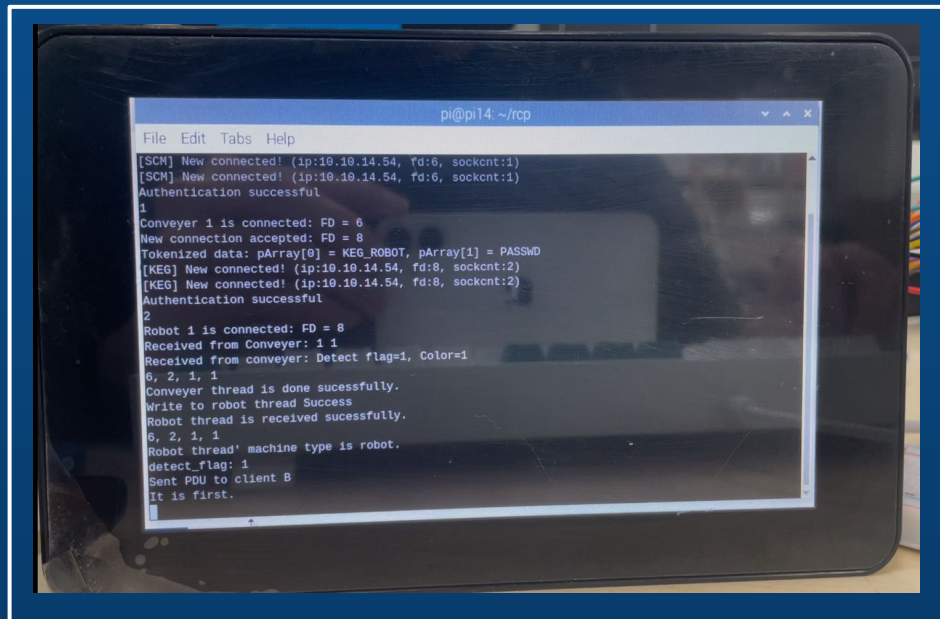
Learn More



Link: [https://github.com/pjw1702/Project\\_SmartFactory/tree/main/gateway](https://github.com/pjw1702/Project_SmartFactory/tree/main/gateway)

## Feature 1

# Communication channel



Link: [https://youtu.be/\\_WCIA4oQqF0](https://youtu.be/_WCIA4oQqF0)

Link: <https://youtube.com/shorts/IDN-9bgD8J8>



## Component

컨베이어와 로봇과의 통신 채널을 생성하기 위한 네트워크  
데몬

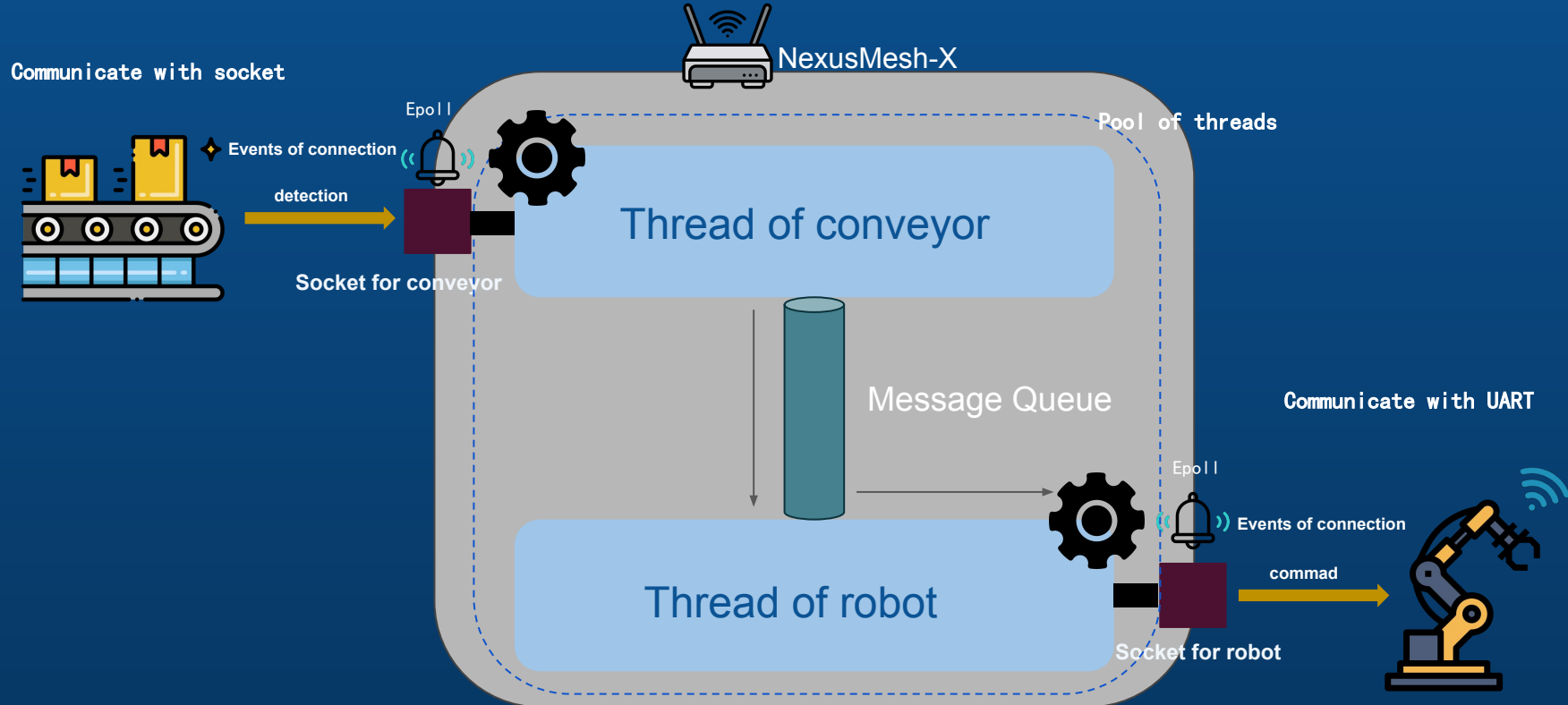


## Support things

- 컨베이어와 로봇을 원격으로 제어하기 위한 프로토콜(RCP) 탑재
- Epoll 시스템을 통한 이벤트 기반 비동기 클라이언트 연결 지원
- POSIX 메시지 큐를 통한 Publisher-Subscriber 모델 기반 통신
- POSIX 스레드를 통한 N:N 멀티스레딩 기반의 채널 생성
- UART 통신을 통해 STM32 기반의 클라이언트 하드웨어 제어

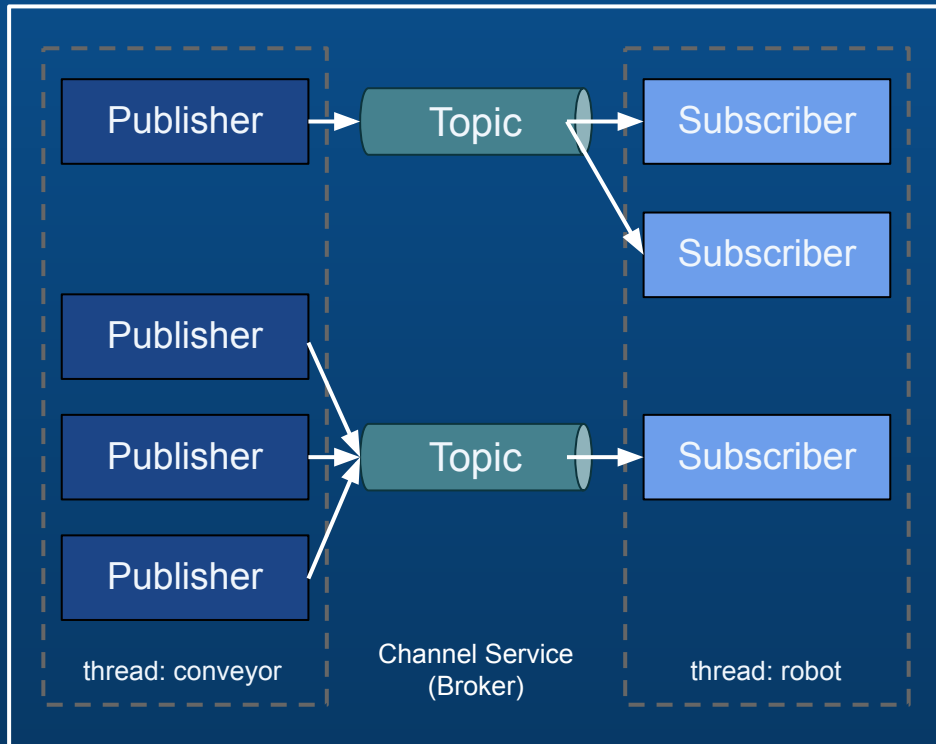
## Feature 1

# Communication channel



## Feature 1

# Communication channel



## Concepts

NexusMesh-X는 메시지 기반의 데이터를 송수신하며, Pub-Sub 패턴 메시지 큐 기반의 통신 채널을 브로커로써 지원합니다.

채널의 Publisher는 컨베이어이며, Subscriber는 로봇이 되어 각 객체가 스레드로써 실행됩니다.



## Specification

```
// Subscriber 스레드 풀 공간 할당
pthread_t conveyer_thread_pool[THREAD_POOL_SIZE];
// Publisher 스레드 풀 공간 할당
pthread_t robot_thread_pool[THREAD_POOL_SIZE];
...
// Subscriber 스레드 생성
pthread_create(&conveyer_thread_pool[conveyer_id], NULL,
conveyer_handler, &fd_data.conveyer_fd[conveyer_id]);
// Publisher 스레드 생성
pthread_create(&robot_thread_pool[robot_id], NULL, robot_handler,
&fd_data.robot_fd[robot_id]);
// 채널 토막 생성
mqd_t mq = mq_open(TOPIC_NAME, O_RDWR | O_NONBLOCK);
```

## Feature 1

# Communication channel



## structures of message

type of message(2bytes)	version of protocol(1bytes)
Length of palyoads(4bytes)	code of response(2bytes)
Checksum(2bytes)	
Timestamp(8bytes)	

## RCP(Robot Control Protocol) header



## Concepts

RCP의 메시지 헤더 구조는 프로토콜 버전, 메시지 타입, 메시지 응답 코드, 페이로드의 전체 길이, 체크섬, 메시지를 전송한 시간(타임스탬프) 값을 필드로 갖도록 설계되어 있습니다.



## Specification

```
/* 응답 코드 및 메시지 타입은 rcp.h 파일에 매크로 상수로 정의되어  
있음 */
```

```
typedef struct __rcp_header__ {  
    char proto_ver[8];  
    char msg_type[16];  
    uint16_t err;  
    uint16_t p_size;  
    uint16_t check;  
    uint64_t timestamp;  
    uint32_t d_len;  
} Rcphdr;
```

## Feature 1

# Communication channel



## structures of message

yaw (8bytes)	pitch (8bytes)	roll (8bytes)
x_pos (8bytes)	y_pos (8bytes)	z_pos (8bytes)
flag of detect (1bytes)		flag of color (1bytes)
id of conveyer (1bytes)		id of robot (1bytes)
command (20bytes)		

## RCP(Robot Control Protocol) payload



## Concepts

RCP의 메시지 페이로드 구조는 로봇 팔이 움직인 공간의 x,y,z 좌표(pos)와 각 좌표로 회전한 각도(yaw, pitch, roll), 컨베이어로 부터의 탐지 및 색깔 플래그와 id, 명령 데이터를 필드로 갖도록 설계되어 있습니다.



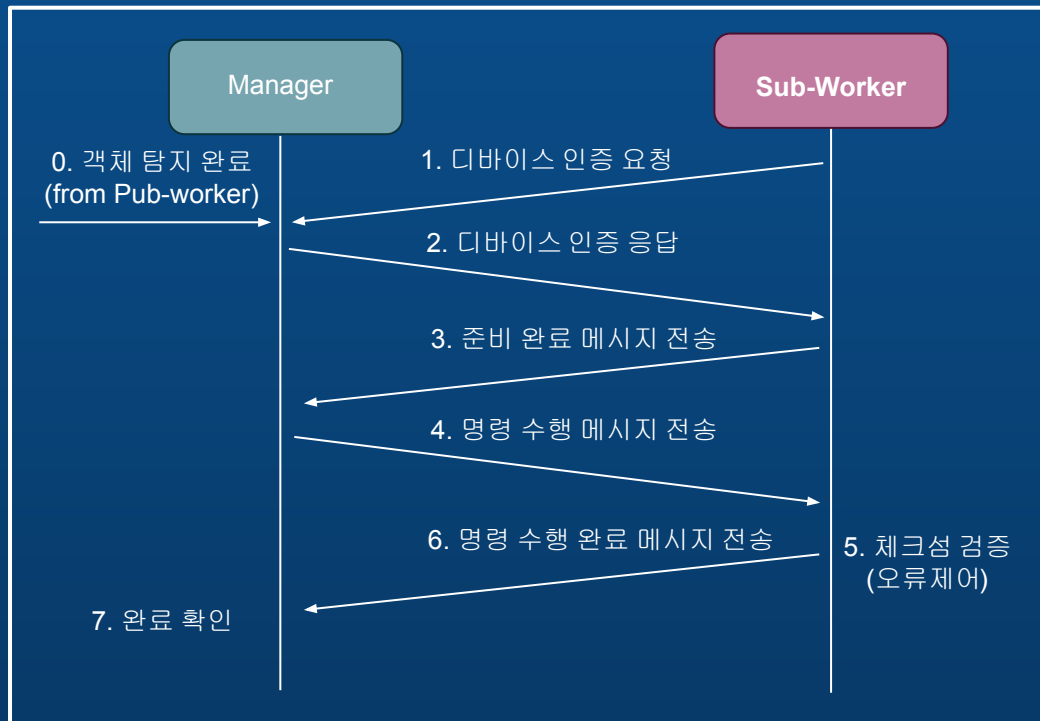
## Specification

```
/* 명령 데이터 및 플래그 값은 rcp.h 파일에 매크로 상수로 정의되어  
있음 */
```

```
typedef struct __rcp_payload__ {  
    float yaw;  
    float pitch;  
    float roll;  
    float x_pos;  
    float y_pos;  
    float z_pos;  
    uint8_t detect_flag;  
    uint8_t color;  
    char data[20];  
} RcdData;
```

## Feature 1

# Communication channel



## Concepts

NexusMesh-X의 RCP는 통신에 필요한 인증제어, 오류제어 흐름제어를 각 단계에 따라 수행하도록 설계되어 있습니다.

채널의 클라이언트는 Worker, 채널 공급자는 Manager라는 이름으로 오브젝트가 설계되어 있습니다.



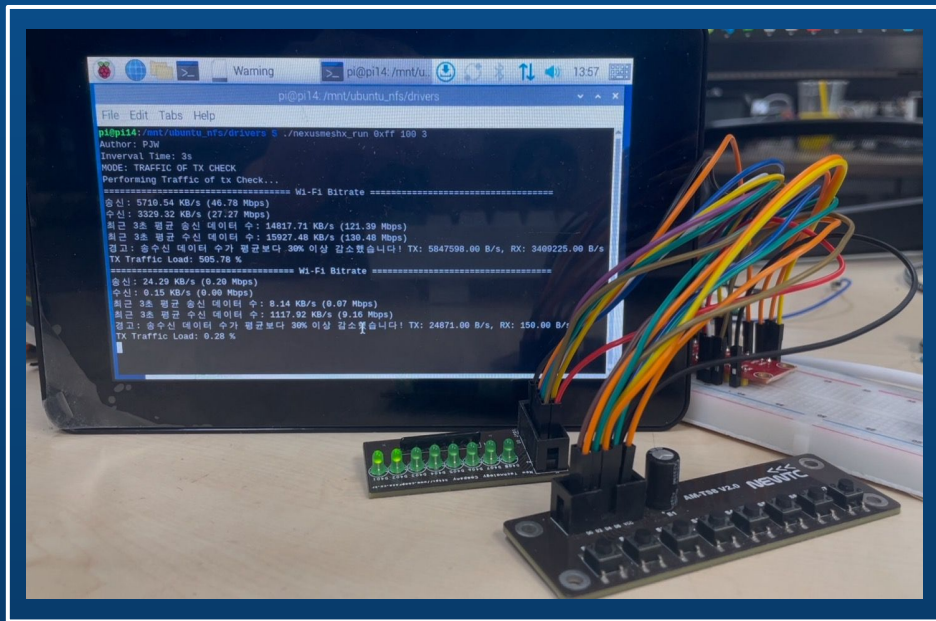
## Specification

```
// Subscriber 스레드 풀 공간 할당
pthread_t conveyer_thread_pool[THREAD_POOL_SIZE];
// Publisher 스레드 풀 공간 할당
pthread_t robot_thread_pool[THREAD_POOL_SIZE];
...
// Subscriber 스레드 생성
pthread_create(&conveyer_thread_pool[conveyer_id], NULL,
conveyer_handler, &fd_data.conveyer_fd[conveyer_id]);
// Publisher 스레드 생성
pthread_create(&robot_thread_pool[robot_id], NULL, robot_handler,
&fd_data.robot_fd[robot_id]);
// 채널 토픽 생성
mqd_t mq = mq_open(TOPIC_NAME, O_RDWR | O_NONBLOCK);
```



## Feature 2

# Kernel module for check Wi-Fi health



Link: <https://youtu.be/THtCkkHFF7w>



## Component

스마트 팩토리용 게이트웨이를 사용 시, 해당 게이트웨이의 Wi-Fi 네트워크 상태를 LED로 점검하기 위한 커널 모듈 및 드라이버입니다.

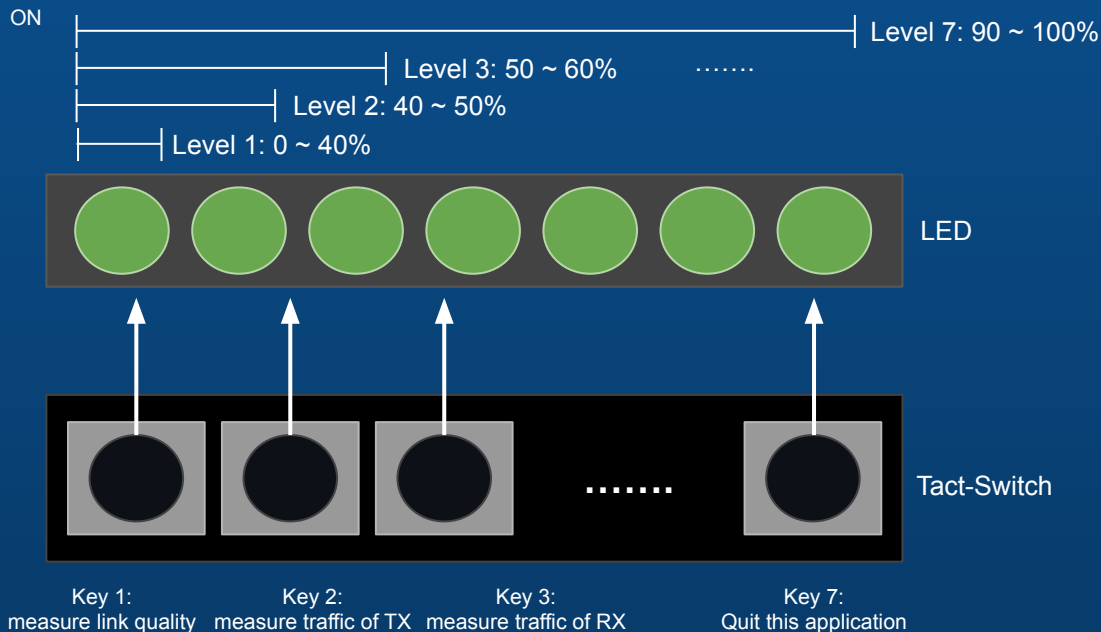


## Support things

- Wi-Fi의 신호 품질을 LED 점등 갯수를 통해 확인
- Wi-Fi 대역폭에 대한 트래픽 상태를 LED 점등 갯수를 통해 확인
- 커널 타이머 실행을 통해 1초 기준의 정확한 시간으로 계산
- 원하는 타이머 시간 및 시간 간격(Interval-Time) 설정 가능
- Netlink, Generic-Netlink 기반의 IPC 수행과 nl80211 시스템 콜 라이브러리를 통한 정확한 Wi-Fi 정보 값 추출
- 차후 버전인 v1.1에서는 cfg80211 라이브러리 기반으로 커널 레벨에서 정보 값을 얻을 수 있도록 개발 진행 중

## Feature 2

# Kernel module for check Wi-Fi health



## Manual

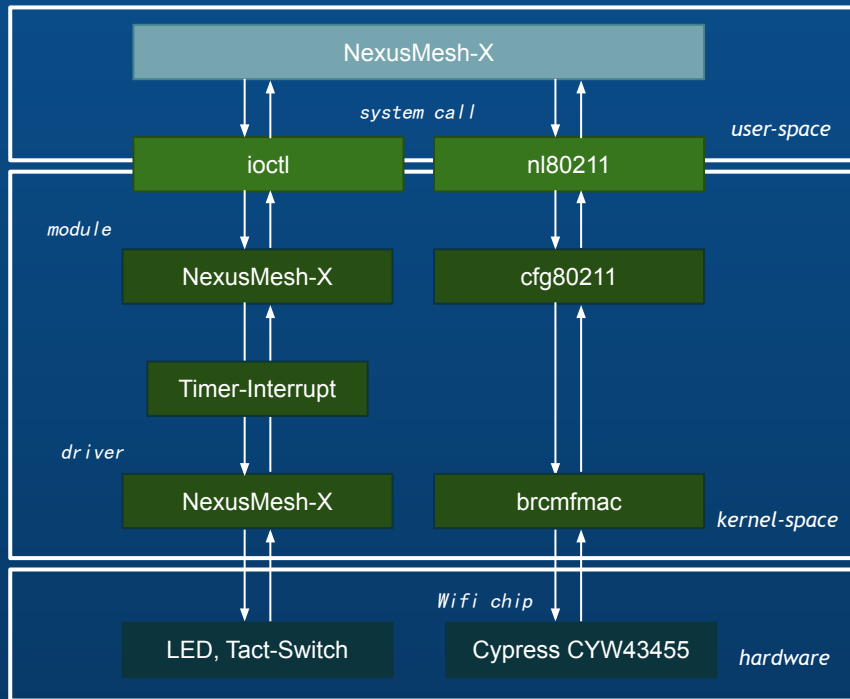
NexusMesh-X는 LED 점등을 통해 현재 Wi-Fi의 네트워크 품질 상태를 점검할 수 있도록 지원합니다.

연결된 Tact-Switch의 1번은 신호의 품질, 2번과 3번은 각각 TX와 RX 대역폭에 대한 트래픽 상태를 실시간으로 확인하실 수 있습니다.

파라미터 값으로 커널 타이머 값과 측정 시간 간격 값을 각각 전달함으로써, 원하는 측정 시간으로 자유롭게 동작시키실 수 있습니다.

## Feature 2

# Kernel module for check Wi-Fi health



## Concepts

NexusMesh-X는 현재 버전에서는 `nl80211` 시스템 콜 API를 통해 Wi-Fi에 대한 정보 메시지를 전달받습니다.

차후 버전인 v1.1에서는 `cfg80211` 커널 라이브러리를 통해, 커널 레벨에서 Wi-Fi 데이터를 제어할 예정입니다.



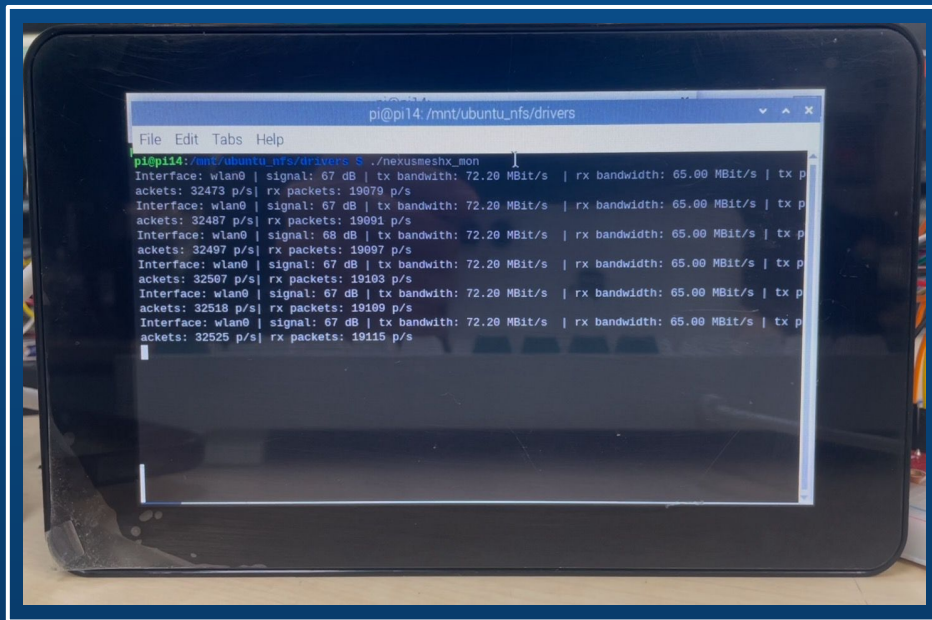
## Specification

```
// nl80211에 대한 Netlink 메시지 파싱을 위한 정책 정의
static struct nla_policy stats_policy[NL80211_STA_INFO_MAX + 1] = {
    ...
    [NL80211_STA_INFO_TX_BITRATE] = { .type = NLA_NESTED },
    [NL80211_STA_INFO_RX_BITRATE] = { .type = NLA_NESTED },
    ...
}; // 대역폭 속성과 같은 경우 하위 중첩 관계를 가지므로, 추가 정책 정의 필요

// nl80211에 대한 Netlink 메시지의 속성 파싱
nla_parse_nested(wifi_rate_info, NL80211_RATE_INFO_MAX,
    wifi_sta_info[NL80211_STA_INFO_TX_BITRATE], rate_policy);
...
bitrate = nla_get_u16(wifi_rate_info[NL80211_RATE_INFO_BITRATE])
```

## Feature 3

# Real-time monitoring tool of Wi-Fi status



Link: <https://youtu.be/pCD0AjM-aF0>



## Component

스마트 팩토리용 게이트웨이를 사용 시, 해당 게이트웨이의 WiFi 네트워크 상태를 실시간으로 모니터링하기 위한 터미널 툴

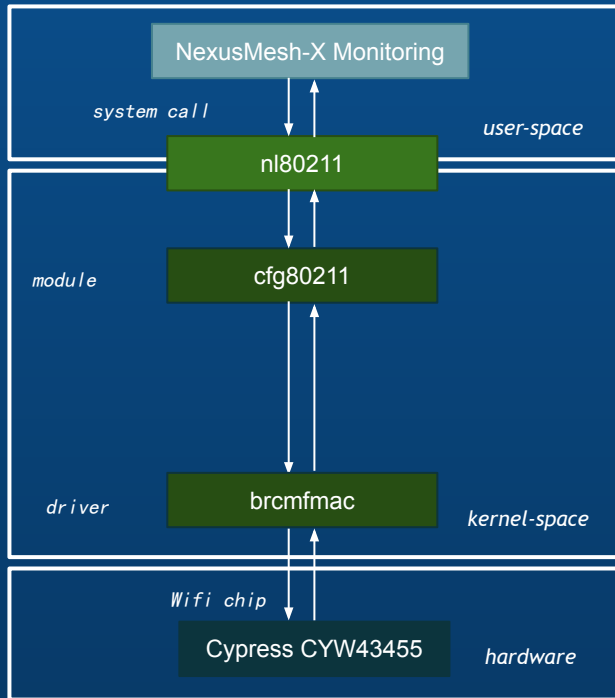


## Support things

- Wi-Fi 네트워크 인터페이스 명 확인
- Wi-Fi 네트워크의 신호 강도 확인
- Wi-Fi 네트워크의 현재 송신 및 수신 대역폭 확인
- Wi-Fi 네트워크의 현재 송신 및 수신 패킷 수 확인
- 차후 버전인 v1.1에서는 cfg80211 라이브러리 기반으로 커널 레벨에서 정보 값을 얻을 수 있도록 개발 진행 중

## Feature 3

# Real-time monitoring tool of Wi-Fi status



## Concepts

NexusMesh-X는 현재 버전에서는 nl80211 시스템 콜 API를 통해 Wi-Fi에 대한 정보 메시지를 전달받습니다.

차후 버전인 v1.1에서는 cfg80211 커널 라이브러리를 통해, 커널 레벨에서 Wi-Fi 데이터를 제어할 예정입니다.



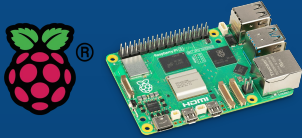
## Specification

```
// nl80211에 대한 Netlink 메시지 파싱을 위한 정책 정의
static struct nla_policy stats_policy[NL80211_STA_INFO_MAX + 1] = {
    ...
    [NL80211_STA_INFO_TX_BITRATE] = { .type = NLA_NESTED },
    [NL80211_STA_INFO_RX_BITRATE] = { .type = NLA_NESTED },
    ...
}; // 대역폭 속성과 같은 경우 하위 종첩 관계를 가지므로, 추가 정책 정의 필요

// nl80211에 대한 Netlink 메시지의 속성 파싱
nla_parse_nested(wifi_rate_info, NL80211_RATE_INFO_MAX,
    wifi_sta_info[NL80211_STA_INFO_TX_BITRATE], rate_policy);
...
bitrate = nla_get_u16(wifi_rate_info[NL80211_RATE_INFO_BITRATE])
```

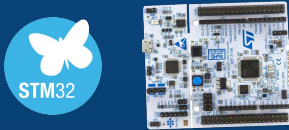
# Used Technologies

## Gateway



- MCU: Raspberry Pi 4b
- OS: Raspberry Pi OS

## Robot



- MCU: STM32F103RB
- IDE: CubeIDE

## Conveyor



- MCU: Arduino Uno R3
- IDE: Arduino IDE

## Programming



- Language: C
- Library: Posix, Netlink, Generic-Netlink, nl80211
- Communicate: Socket, UART

## Test



- OS: Ubuntu 22.04
- Computer: VirtualBox VM

# Contact me

---

본 문서를 읽어주셔서 감사합니다.

차후에도 더욱 더 많은 노력을 기울여 더 발전된 네트워크 소프트웨어를 개발하도록 하겠습니다.



## My developer blog

---



Link: <https://velog.io/@wkftkqslek/series>



## My address

---

E-mail: [qjw1702@gmail.com](mailto:qjw1702@gmail.com)

Tel: +82 10-8946-5291