

# 설치

## 요구사항

- CMake 3.1 이상
- Python 3
- Xenomai
- build\_essential

## 설치 명령어

```
git clone --recursive https://git.kangwon.ac.kr/iic/indurop.git
cd indurop
mkdir build
cd build
cmake ..
sudo cmake --build . --target install
```

git clone ... 은 소스 코드 다운로드를 위한 명령어로 별도의 방법을 이용하여 받는다면 생략해도 된다. (현재 git 저장소 비공개로 별도의 방법을 이용)

경로 설정을 위해 아래의 명령어를 추가 수행한다.

```
echo export PATH=$PATH:/usr/local/InduRoP/bin >> ~/.bash_profile
source ~/.bash_profile
```

## 설치 위치

InduRoP 은 설치 명령어 방법대로 설치하였다면 기본 /usr/local/InduRoP 디렉토리에 설치 된다.

# 모듈 작성 방법

공유 변수를 사용하는 동적 라이브러리형 모듈을 작성하기 위해서는 다음과 같은 3가지 파일이 필요하다.

- CMakeLists.txt : 빌드를 위한 프로젝트 구성 파일
- example.glb : 공유 변수 명세 파일
- example.cpp : 모듈의 소스코드

위 3개 파일에 대한 작성 방법은 아래에서 설명한다.

## example.glb

int32\_t 타입을 갖는 counter 공유 변수를 선언하기 위해서 아래와 같이 glb 파일을 작성할 수 있다.

```
namespace ex;
int32_t counter;
```

namespace ex; 는 공유 변수의 이름공간을 선언한다. int32\_t counter; 는 공유 변수를 선언한다.

추가 적인 공유 변수를 선언하고 싶을 경우 해당 파일에 추가 작성하면 된다. 만약 구조체 변수를 선언하고 싶을 경우 아래와 같은 방식으로 선언할 수 있다.

```
struct {
    int a, b, c;
} composed;
```

glb 파일은 최종적으로 h와 cpp 파일로 변환되어 C++ 코드에서 사용 할 수 있다. 생성되는 h 파일은 glb 파일의 확장자를 제거한 이름과 동일하게 생성된다.

## example.cpp

example.glb 에서 선언한 공유 변수인 counter 를 증가 시키면서 출력하는 모듈의 소스 코드는 아래와 같다.

```
#include <iostream>
#include <example.h>

extern "C" void run() {
```

```
std::cout << ex::counter << std::endl;
ex::counter = ex::counter + 1;
}
```

#include <example.h> 는 전처리를 통해 example.glb 파일이 C++ 언어로 변환된 헤더를 추가한다. example.glb는 example.h 로 변환되기 때문에 example.h 를 추가한다.

미들웨어는 모듈내의 run() 함수를 주기적으로 호출한다. 따라서 모듈의 소스 코드에서 extern "C" void run() 와 같이 함수를 작성하여 주기적인 동작을 수행할 수 있다.

example.glb 에서 선언한 counter 공유 변수는 이름공간이 ex 이다. 따라서 해당 공유 변수에 접근하기 위해서는 std::cout << ex::counter << std::endl; 와 같이 이름공간을 명시하여 해당 공유 변수에 접근할 수 있다.

operator++()와 operator--() 같은 연산은 현재 지원하지 않기 때문에 ex::counter = ex::counter + 1; 같은 방법으로 증가 연산을 이용할 수 있다.

## CMakeLists.txt

```
cmake_minimum_required(VERSION 3.1)

find_package(InduRoP CONFIG REQUIRED)

add_library(example SHARED example.cpp)
add_shared_variables(module1 FILES example.glb)
```

## 모듈 빌드

위 3개의 파일을 작성 한후 아래와 같은 명령어를 통해 모듈을 빌드 할 수 있다.

```
mkdir build
cd build
cmake .. && cmake --build .
```

리눅스에서는 빌드가 완료 되면 libexample.so 파일이 build 디렉토리에 생성된다.

## 모듈 실행

모듈 실행을 위해서는 해당 모듈의 실행 주기 및 타입에 대한 내용이 작성 되어 있는 modules.xml 이 필요하다.

### modules.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <module>
    <filename>libexample.so</filename>
    <moduletype>thread</moduletype>
    <operationtype>periodic</operationtype>
    <period>1000000000</period>
  </module>
</root>
```

위 modules.xml 에서 filename 은 모듈의 파일 경로를 입력한다. 이 때 파일 경로는 절대 경로를 입력한다. period는 모듈을 동작 시킬 주기로 nano sec 단위로 입력한다.

새로운 모듈을 추가하고 싶을 경우 module 항목을 추가 작성하면 된다.

### 실행 명령어

작성이 완료된 후 modules.xml 이 있는 디렉토리로 이동하여 아래와 같은 명령어를 수행 함으로써 모듈을 실행할 수 있다.

```
middleware
```

또는

```
/usr/local/InduRoP/bin/middleware
```