DSSP-HAL
Operation Manual
(TETRTA-DS III PRO™)

Version 1.1

2011. 03.

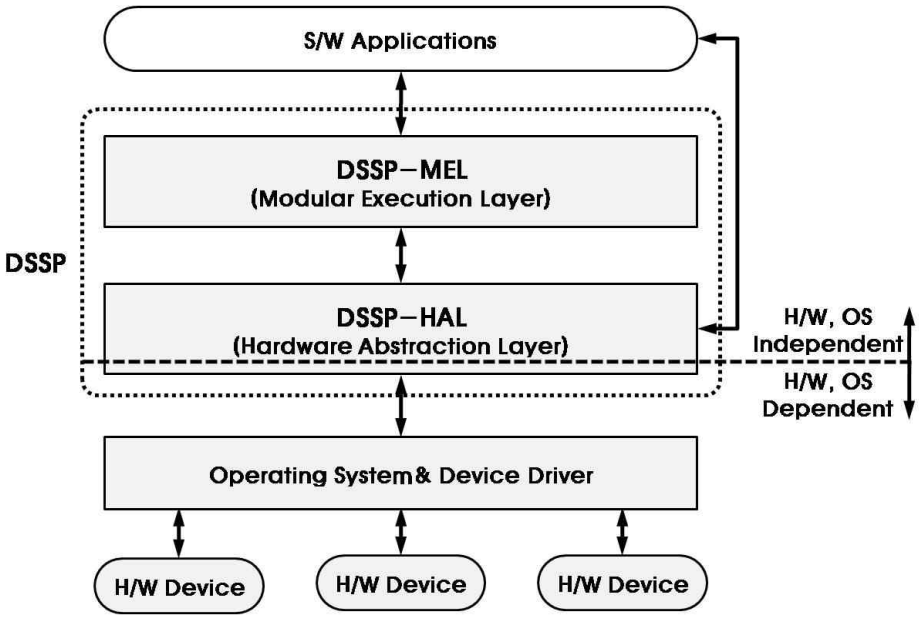
dASAROBOT

목 차 (Table of Contents)

Chapter 1. DSSP-HAL 개요 (DSSP-HAL Overview)

1-1. DSSP란? (What' s DSSP?)

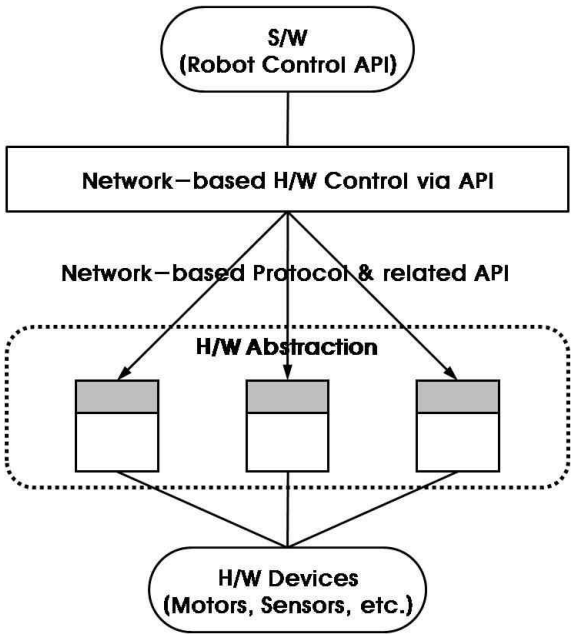
'DSSP(DaSarobot Software Platform)' 는 이동로봇의 자율주행 기술 개발을 위해 ㈜다사로봇에서 개발한 네트워크 기반 모듈화된 소프트웨어 플랫폼(Network-based Modularized Software Platform)을 말합니다. 그림 1-1은 DSSP의 소프트웨어 아키텍처(Architecture)를 나타내고 있습니다. 그림 1-1에 나타난 바와 같이 '하드웨어(Hardware)' 및 '운영체제(Operating System)' 에 비의존적인(Independent) 특성을 지닌 응용 프로그램의 개발을 지원하기 위해 DSSP는 'DSSP-HAL' 과 'DSSP-MEL' 의 2개의 층(Layer)으로 구성되어 있습니다. 'DSSP-HAL' 은 이동로봇에 장착된 다양한 장치-하드웨어-들을 접근할 수 있는 하드웨어 및 운영체제에 비의존적인 특성을 지닌 XML 형태의 통신규약(Protocol)을 가진 네트워크 기반의 추상화된 API층이며, 'DSSP-MEL' 은 DSSP-HAL API들의 조합으로 이동로봇의 자율주행 응용프로그램(Application)의 개발을 용이하게 하는 운영체제에 비의존적인 '파이선(Python)' 기반의 모듈화된 실행층(일종의 프로세스)입니다. DSSP-MEL에는 이동로봇의 자율주행 기술을 구성하는 '장애물 감지/회피(Obstacle Detection/Avoidance)' , '경로 계획(Path Planning)' , '위치인식(Localization)' , 그리고 '맵 생성(Map Building)' , 등과 같은 기술들을 구현하는 다양한 모듈들이 계층적(Hierarchical) 구조로 이루어져 있습니다. DSSP 및 DSSP-MEL에 대한 세부적인 기술 사항은 당사 홈페이지 및 기술지원센터를 통해 정보를 얻으실 수 있습니다.



<그림 1-1> DSSP Architecture

1-2. DSSP-HAL이란? (What' s DSSP-HAL?)

'DSSP-HAL(DaSarobot Software Platform — Hardware Abstraction Layer)' 은 ㈜다사로봇의 이동로봇 플랫폼인 'TETRA-DS III™' 에 기본적으로 장착되어 있거나, 혹은 옵션품으로 플랫폼에 장착할 수 있는 모든 하드웨어-장치-들을 통합된 형태로 간단히 접근(Access) 및 제어(Control)할 수 있도록 XML 형태의 통신 규약으로 이루어진 네트워크 기반(Network-based) 하드웨어 추상화 계층을 말합니다. 그림 1-2는 DSSP-HAL의 개념적인 구성도를 나타내고 있습니다.



〈그림 1-2〉 DSSP-HAL Concept

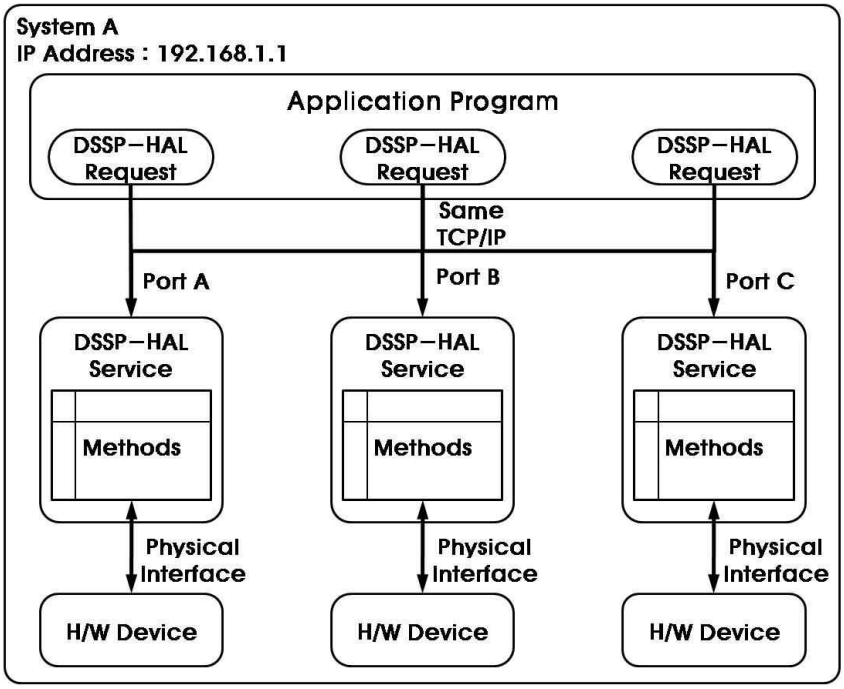
그림 1-2에 나타난 바와 같이 DSSP-HAL은 로봇에 적용되는 다양하고 복잡한 하드웨어들을 좀 더 쉽게 사용할 수 있도록 추상화하는 것을 그 목적으로 합니다. 따라서, DSSP-HAL은 쉽게 이해하고 사용할 수 있는 간단 명료한 데이터 구조를 지원하며, 네트워크를 기반으로 구성되어 네트워크를 통한 분산 처리 구조로 하드웨어 계층을 추상화합니다. 또한, 네트워크를 통해 주고 받는 통신 규약은 시스템에 의존적이지 않은 XML(eXtended Mark-up Language)로 구성되어 있어, 다양한 시스템을 기반으로 하는 이동로봇의 하드웨어 계층을 구성하는 데 적합합니다. DSSP-HAL의 XML기반 통신 규약은 2장의 'DSSP-HAL 구조' 설명부분을 참조하시기 바랍니다.

1-3. DSSP-HAL API 구현 (DSSP-HAL API Implementation)

DSSP-HAL은 소켓(Socket) 인터페이스를 통해 TCP/IP 프로토콜(Protocol)을 기반으로 네트워크 상에서 동작합니다. 그림 1-3과 그림 1-4는 다양한 사용 환경(네트워크 환경)에서 구현되는 DSSP-HAL API의 통신 형태를 표현하고 있습니다. 그림 1-3과 그림 1-4에 나타난 바와 같이 DSSP-HAL은 전형적인 네트워크 기반 응용의 성격을 띠며, 네트워크를 중심으로 서버 역할을 하며 '서비스를 제공하는 부분(DSSP-HAL Service)', 클라이언트 역할을 하며 '서비스를 요청하는 부분(DSSP-HAL Request)'으로 크게 나눌 수 있습니다. 서비스를 제공하는 부분은 로봇의 복잡한 하드웨어와 연동하여 그 기능을 좀 더 쉽게 사용할 수 있도록 'RPC(Remote Procedure Call)'의 메소드(Method) 형태로 하드웨어를 추상화합니다. 또한, 서비스를 사용하기 위해 주고 받는 데이터는 DSSP-HAL에서 제공하는 간단하고 유연한 데이터 구조를 사용해 표준화되고 단순화되어, 결과적으로 로봇의 복잡한 하드웨어는 DSSP-HAL의 기능을 통해 네트워크 상에서 원격으로 간단히 메소드들을 호출하는 방법으로 제어할 수 있도록 추상화됩니다. DSSP-HAL은 TCP/IP 프로토콜을 기반으로 동작하므로, TCP/IP 프로토콜을 지원하는 어떠한 시스템에서도 구현이 가능합니다.

1-3-1. 동일 시스템상에서의 API 구현 (API Implementation in same System)

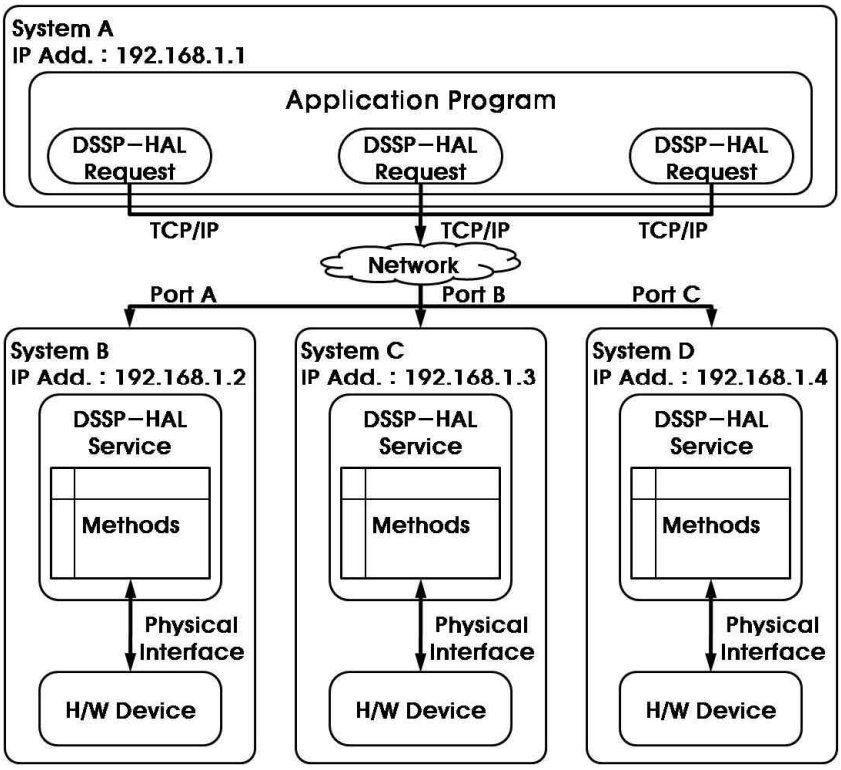
그림 1-3에 나타나 있는 바와 같이 동일 네트워크 시스템상 – 동일한 IP 주소를 가지는 상황 - 에 있는 하드웨어로의 접근 및 하드웨어의 제어를 위해서, DSSP-HAL 호출(Request)은 포트(Port) 구분만을 통해서 장치 – 하드웨어(Hardware) - 로의 접근 및 제어가 가능한 특정 DSSP-HAL Service를 호출하게 됩니다. 특정 포트로 할당된 DSSP-HAL 서비스가 이러한 DSSP-HAL 호출 명령을 받게 되면, DSSP-HAL 서비스가 지원하는 메소드를 통해 하드웨어가 지원하는 물리적인 인터페이스를 통해 하드웨어로 접근하여 그 정보를 받거나 하드웨어를 제어하게 됩니다.



<그림 1-3> DSSP-HAL API Implementation in the same system

1-3-2. 서로 다른 시스템상에서의 API 구현 (API Implementation among Systems)

그림 1-4에 나타나 있는 바와 같이 서로 다른 네트워크 시스템상 – 서로 다른 IP 주소를 가지는 상황 - 에 있는 하드웨어로의 접근 및 하드웨어의 제어를 위해서, DSSP-HAL 호출(Request)은 포트(Port), 뿐만 아니라 IP 주소의 구분을 통해서 장치 – 하드웨어(Hardware) - 로의 접근 및 제어가 가능한 특정 DSSP-HAL Service를 호출하게 됩니다. 특정 IP 주소로 설정된 시스템 내의 특정 포트로 할당된 DSSP-HAL 서비스가 이러한 DSSP-HAL 호출 명령을 받게 되면, DSSP-HAL 서비스가 지원하는 메소드를 통해 하드웨어가 지원하는 물리적인 인터페이스를 통해 하드웨어로 접근하여 그 정보를 받거나 하드웨어를 제어하게 됩니다.



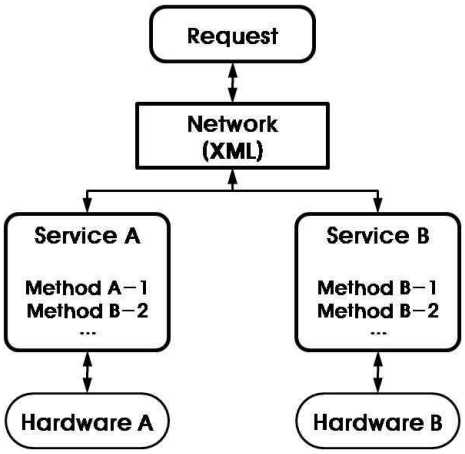
<그림 I-4> DSSP-HAL API Implementation among systems

DSSP-HAL은 TETAA-DS III™에 내장되어 있는 DSCP를 구성하는 주 제어기에 기본적으로 설치되어 있으며, 각각의 하드웨어를 추상화하는 다수의 서비스들로 이루어져 있습니다. 각각의 DSSP-HAL 서비스에 대한 자세한 사항은 다음 장을 참조하시기 바랍니다.

Chapter 2. DSSP-HAL 구조 (DSSP-HAL Structure)

2-1. DSSP-HAL 통신 규약 (DSSP-HAL Protocol)

DSSP-HAL은 네트워크 기반(Network-based)의 하드웨어 추상화 계층이며, DSSP-HAL의 통신 규약(Protocol)은 XML(eXtended Mark-up Language) 기반으로 설계되어 있어 사용하는 운영체제(Operating System)에 비의존적인 특성을 지닙니다. 따라서, DSSP-HAL은 리눅스(Linux)와 윈도우(Windows) 시스템 모두를 지원하며, 윈도우 시스템 환경의 경우에는 winsock2를 사용합니다. 그림 2-1은 DSSP-HAL의 구조를 나타내고 있습니다.

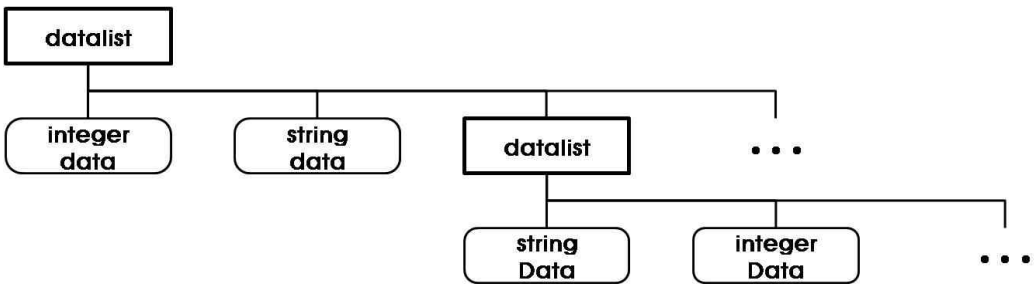


<그림 2-1> DSSP-HAL Structure

그림 2-1에 나타난 바와 같이 모든 서비스들은 서비스가 구현할 수 있는 기능을 나타내는 다양한 '메소드(Method)'들을 가지고 있으며, 각각의 메소드에 의해 요청하는 부분과(Request)과 서비스를 제공하는 부분(Service) 사이에 로봇 제어를 위해 사용되는 실질적인 데이터를 XML 형태의 구조체로 주고 받게 됩니다.

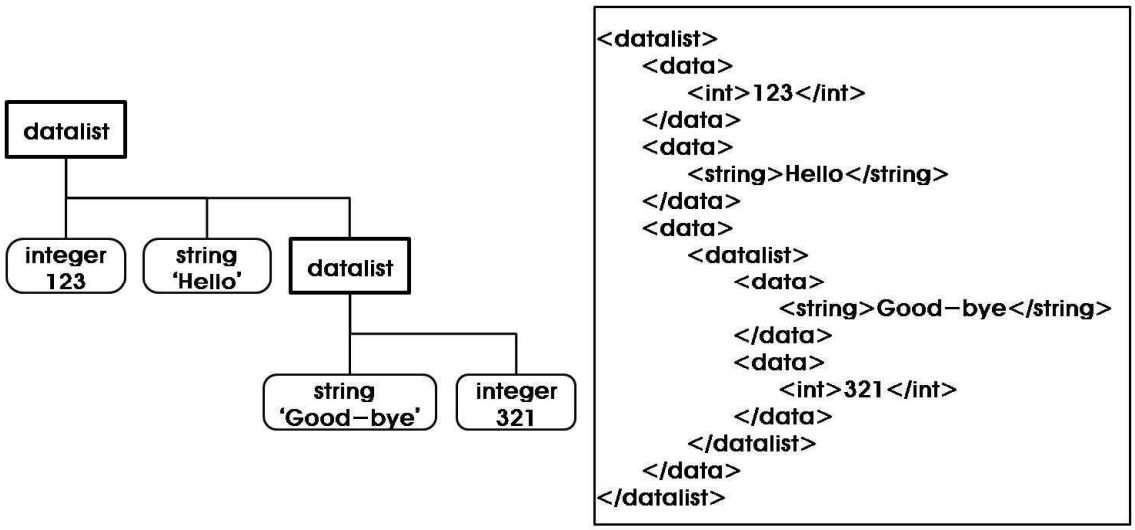
2-2. DSSP-HAL 데이터 (DSSP-HAL Data)

DSSP-HAL은 '정수형(Integer type) 데이터', '문자열형(String type) 데이터', '데이터 목록형(Datalist type) 데이터' 의 3가지 데이터 형을 지원합니다. 그림 2-2에 나타난 바와 같이 데이터 목록형은 정수형, 문자열, 데이터 목록형의 데이터를 구성요소로 가질 수 있습니다. 이러한 데이터를 통해 DSSP-HAL 요청(Request) 부분과 DSSP-HAL 서비스(Service) 부분 사이에 정보 전달이 이루어지게 됩니다.



<그림 2-2> Example of DSSP-HAL Data Structure

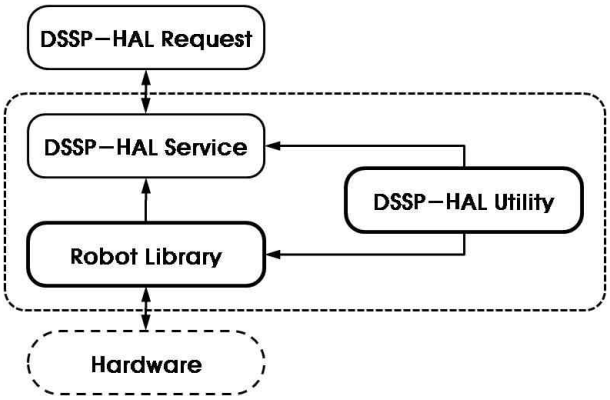
그림 2-2에 나타나 있는 바와 같이 DSSP-HAL 서비스를 통해 제공되는 데이터는 지원하는 데이터 형의 다양한 조합으로 구성이 가능하기 때문에, 다양한 하드웨어들의 정보 전달이 가능합니다. 또한, DSSP-HAL 서비스를 통해 제공되는 데이터는 그림 2-3에 나타난 바와 같이 운영체제에 비의존적인 XML 형태의 구조체로 구성되어 있습니다.



<그림 2-3> XML-based Struct Code of DSSP-HAL Data

2-3. DSSP-HAL 요청과 서비스 (DSSP-HAL Request and Service)

DSSP-HAL은 네트워크를 중심으로 하드웨어와 연동하여 그 기능을 좀 더 쉽게 사용할 수 있도록 서비스를 제공하는 부분(DSSP-HAL Service)과 서비스를 요청하는 부분(DSSP-HAL Request)으로 나눌 수 있습니다. 네트워크 서버의 형태로 하드웨어 관련 서비스를 제공하는 개체를 'DSSP-HAL 서비스(DSSP-HAL Service)'라 통칭합니다. 따라서, DSSP-HAL은 일반적인 네트워크 서버의 형태로 그러한 서비스가 제공되며, 그 서비스를 요청하는 쪽은 네트워크 클라이언트라 볼 수 있습니다. 그림 2-4는 이동로봇의 제어 시, DSSP-HAL의 통신 구조를 나타내고 있습니다.

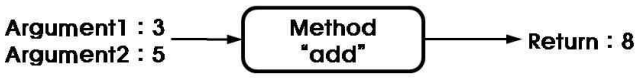


<그림 2-4> DSSP-HAL Request & Service Structure

그림 2-4에 나타나 있는 '로봇 라이브러리(Robot Library)'는 DSSP-HAL 서비스의 메소드를 통해 사용할 수 있는 UART 통신 상의 프로토콜로 정의된 (제어하려고 하는) 실제 로봇에 장착되어 있는 장치 - 하드웨어 - 들의 라이브러리 집합을 의미합니다. 예를 들면, 당사에서 판매하고 있는 이동로봇 플랫폼인 'TETAA-DS III™'의 'TETAA Library'는 '로봇 라이브러리'의 일종입니다.

또한, 그림 2-4의 'DSSP-HAL 유틸리티(DSSP-HAL Utility)'는 소켓(Socket) 통신, UART 통신, 세마포어(Semaphore), 등 '로봇 라이브러리'와 'DSSP-HAL 서비스'에서 기본적으로 사용할 수 있도록 정의된 함수들의 집합을 의미합니다.

DSSP-HAL의 데이터 구조에서 설명한 바와 같이, DSSP-HAL의 통신 규약은 운영체제에 비의존적인(Independent) XML 기반의 구조체로 설계되어 있습니다. 아래 그림 2-5는 두 인자를 더하여 그 값을 반환하는 'add'라는 메소드(Method)를 호출하는 경우의 호출(Request)과 서비스(Service)의 XML 기반의 구조체의 예제를 나타내고 있습니다. 다시 말해, DSSP-HAL 서비스의 모든 메소드들은 그림 2-5에 나타난 바와 같은 XML 기반의 구조체로 설계되어 있어 운영체제에 비의존적인 특성을 지니고 있습니다.




```
<method_call>
    <method_name>add</method_name>
        <method_datalist_arg>
            <datalist>
                <data>
                    <int>3</int>
                </data>
                <data>
                    <int>4</int>
                </data>
            </datalist>
        </method_datalist_arg>
</method_call>
```

(a) Request XML-based Struct Code

```
<method_response>
    <method_datalist_ret>
        <datalist>
            <data>
                <int>8</int>
            </data>
        </datalist>
    </method_datalist_ret>
</method_response>
```

(b) Service XML-based Struct Code

<그림 2-5> XML-based Struct Code of DSSP-HAL Request & Service

2-4. DSSP-HAL 서비스 메소드 (DSSP-HAL Service Method)

DSSP-HAL을 사용하는 경우, 특정 하드웨어로의 접근 및 하드웨어의 제어는 DSSP-HAL 서비스 내에 등록된 특정 메소드의 호출을 통해 가능합니다. 그림 2-7은 당사의 이동로봇 플랫폼인 'TETAA-DS III™' 의 구동 모터를 제어하는 기능을 수행하는 DSSP-HAL 서비스인 'Drive 서비스' 내에 등록된 메소드 중의 하나인 'method_velocity_control' 의 코드를 나타내고 있습니다.

```
dsphal_method_return_t method_velocity_control(
                    dsphal_datalist_t *datalist_arg, dsphal_datalist_t **datalist_ret)
{
        tetra_drive_module_t tetra_drive_module;

        int left_vel;
        int right_vel;
        int semaphore;

        if (!datalist_arg)
                goto ret_err_3;
        if ((semaphore = tetra_semaphore_create(SEMKEY_DRIVE_MODULE, 1)) < 0)
                goto ret_err_3;
        tetra_semaphore_wait(semaphore);
        if (tetra_drive_module_open(&tetra_drive_module, DEVICE_DRIVE_MODULE)) ❶
                goto ret_err_2;
        dsphal_decompose_root_datalist(datalist_arg, "[{i}{i}]", &left_vel, &right_vel); ❷
        if (tetra_drive_module_vel_ctrl(&tetra_drive_module, left_vel, right_vel))
                goto ret_err_1;
        tetra_drive_module_close(&tetra_drive_module);
        tetra_semaphore_signal(semaphore);
        tetra_semaphore_close(semaphore);

        return DSPHAL_METHOD_RETURN_OK;
ret_err_1:
        tetra_drive_module_close(&tetra_drive_module);
ret_err_2:
        tetra_semaphore_signal(semaphore);
        tetra_semaphore_close(semaphore);
ret_err_3:
        return DSPHAL_METHOD_RETURN_ERR;
}
```

<그림 2-7> Example of DSSP-HAL Method

그림 2-7에 나타나 있는 바와 같이 DSSP-HAL 서비스 내의 각각의 메소드들은 (그림 2-4에 나타나 있는) '로봇 라이브러리(Robot Library)'

인 'TETAA Library' 와 'DSSP-HAL Utility' 에 정의되어 있는 함수들을 호출하는 형태로 구성됩니다. 그림 2-7에 나타나 있는 코드 중 ① 번 라인 내의 tetra_drive_module_open( )은 '로봇 라이브러리(Robot Library)' 함수를, ②번 라인 내의 dsphal_decompose_root_datalist( )' 는 'DSSP-HAL Utility' 함수를 나타냅니다.

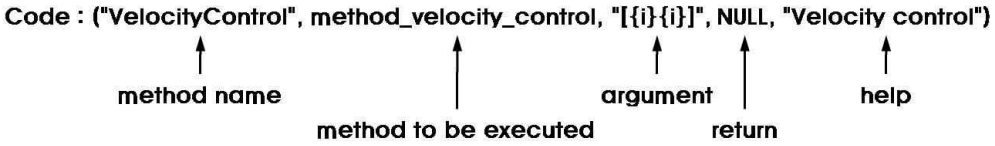
DSSP-HAL 서비스 (Service) 내의 각각의 메소드는 DSSP-HAL 서비스 main에서 등록되어 요청(Request) 명령이 호출되면 해당 메소드를 실행한 후 대기하는 형태로 구현되어 있어, 네트워크 상의 서버(Server)의 역할과 같은 기능을 수행합니다. 그림 2-8은 당사 이동로봇 플랫폼인 'TETAA-DS III™' 에 장착된 구동 모터를 제어하는 기능을 수행하는 DSSP-HAL 서비스인 'Drive 서비스' 의 main 함수 코드의 일부를 나타내고 있습니다.

```
int main(int argc, char **argv)
{
    dsphal_tcp_server_t *tcp_server;
    dsphal_tcp_server_t *new_tcp_server;
    dsphal_service_t *service;
    dsphal_registry_t *registry;
    dsphal_initialize();
    if (initialize())
        return -1;
    registry = dsphal_registry_create();
    dsphal_introspection_register(registry);
    dsphal_registry_add_method(registry, dsphal_method_create(
        "VelocityControl", method_velocity_control, "[{i}{i}]", NULL, "Velocity
control"));
    ......
    service = dsphal_service_create(registry);
    tcp_server = dsphal_tcp_server_create(PORT_DRIVE);
    while (1) {
        new_tcp_server = dsphal_tcp_server_accept(tcp_server);
        dsphal_service(new_tcp_server, service);
        dsphal_tcp_server_destroy(new_tcp_server);
    }
    return -1;
}
```

<그림 2-8> Example of DSSP-HAL Service Main

그림 2-8에 나타나 있는 메소드 생성 함수 내의 인자들의 설명은 아래와 그림 2-9에 나타나 있는 바와 같습니다. 그림 2-9에 나타나 있는 인자(argument) 또는 리턴(return) 값은 "[{?}]" 형태로 정의되며, 물음표(?)에 'i' 가 있는 경우에는 정수(integer)형 데이터를, 's' 가 있는 경우에는 문자(string)형 데이터를 의미합니다. 또한, {?}의 개수는 인자나 리턴값의 개수를 의미합니다. 그림 2-9에 나타난 바와 같이 'VelocityControl' 이라는 메소드는 2개의 정수형 데이터를 인자(argument)로 가지며 리턴값(return)은 없음을 의미합니다.

그림 2-9는 DSSP-HAL 서비스에서 메소드를 추가 및 등록하는 'DSSP-HAL Utility' 함수인 'dsphal_registry_add_method' 의 인자에 대한 설명을 나타내고 있습니다.



<그림 2-9> Example of DSSP-HAL Service Code

그림 2-10은 당사의 이동로봇 플랫폼인 'TETAA-DS III™' 의 좌/우 바퀴의 속도를 좌측 100 mm/s, 우측 100 mm/s로 구동시키는 제어 명령을 수행할 경우, DSSP-HAL로 구현한 요청(Request) 구문의 예제 코드를 나타내고 있습니다.

```
#include "dsphal.h"  ❶

#define IP_ADDR_DMP "192.168.51.10"  ❷
#define PORT_DRIVE 50010  ❸

int main(void)
{
    dsphal_tcp_client_t *tcp_client;
    dsphal_datalist_t *datalist_arg;

    tcp_client = dsphal_tcp_client_create(IP_ADDR_DMP, PORT_DRIVE);
    dsphal_tcp_client_connect(tcp_client);
    datalist_arg = dsphal_build_root_datalist("[{i}{i}]", 100, 100);  ❹
    dsphal_request_method_call(tcp_client, "VelocityControl", datalist_arg);  ❺
    dsphal_tcp_client_destroy(tcp_client);

    return 0;
}
```

<그림 2-10> Example of DSSP-HAL Request

그림 2-10에 나타나 있는 요청 구문의 코드 내부의 라인별 설명은 아래와 같습니다.

라인 ① : DSSP-HAL이 정의되어 있는 헤더(Header)를 Include합니다.

라인 ② : DSSP-HAL 서비스가 실행되는 주 제어보드의 IP 주소를 정의합니다.

라인 ③ : 각각의 DSSP-HAL Service의 포트 번호를 정의합니다.

라인 ④ : 메소드에서 필요로 하는 인자(argument)를 구성합니다.

라인 ⑤ : 'VelocityControl' 이라는 메소드를 호출합니다.

Chapter 3. DSSP-HAL Service 로 의 접근 (Access DSSP-HAL Service)

이번 장에서는 DSSP-HAL의 핵심 중 하나인 특정 하드 웨어로 의 접근 및 하드 웨어의 제어 기능을 수행하는 DSSP-HAL 서비스로 의 접근 방법을 설명합니다.

3-1. DSSP-HAL 서비스 구조체 (Struct code of DSSP-HAL Service)

3-1-1. dsphal_tcp_client_t

Struct	typedef struct dsphal_tcp_client_t { char *server_ip_addr; int server_port; int socket_fd; } dsphal_tcp_client_t;	
Description	접속 할 DSSP-HAL 서비스 의 TCP 정보 저장 dsphal_tcp_client_create( ) 함수 의 리턴 값	
Argument	server_ip_addr	DSSP-HAL 서비스 IP(v4) 주소
	server_port	DSSP-HAL 서비스 Port 번호
	socket_fd	소켓의 파일 디스크립터
Example Code	dsphal_tcp_client_t *tcp_client; (포인터로 정의)	

3-1-2. dsphal_datalist_t

Struct	typedef struct dsphal_datalist_t { size_t size; dsphal_data_t *head; dsphal_data_t *tail; } dsphal_datalist_t;	
Description	Argument(입력) 또는 Return(출력) 받을 데이터 리스트 dsphal_request_method_call( ) 함수 의 리턴 값	
Argument	size	datalist의 크기
	*head	datalist의 시작 포인터
	*tail	datalist의 끝 포인터
Example Code	dsphal_datalist_t * datalist_arg; (포인터로 정의)	

3-2. DSSP-HAL 서비스 함수 (Functions of DSSP-HAL Service)

3-2-1. dsphal_tcp_client_t *dsphal_tcp_client_create(char *server_ip_addr, int server_port)

Function Name	dsphal_tcp_client_t *dsphal_tcp_client_create(char *server_ip_addr,

	int server_port)	
Argument	server_ip_addr	DSSP-HAL 서비스 IP(v4) 주소
	server_port	DSSP-HAL 서비스 Port 번호
Description	접속 할 DSSP-HAL 서비스의 IP 및 Port 정보 입력	
Return	OK(address)	입력된 TCP 정보의 주소 값
	NULL	실패
Example Code	#define IP_ADDR_DSSMP "192.168.51.10" #define PORT_BUMPER 50011 dsphal_tcp_client_t *tcp_client; tcp_client = dsphal_tcp_client_create(IP_ADDR_DSSMP, PORT_BUMPER);	

3-2-2. int dsphal_tcp_client_connect(dsphal_tcp_client_t *dsphal_tcp_client)

Function Name	int dsphal_tcp_client_connect(dsphal_tcp_client_t *dsphal_tcp_client)	
Argument	dsphal_tcp_client	접속 할 DSSP-HAL 서비스의 TCP 정보
Description	접속 할 DSSP-HAL 서비스의 TCP에 연결	
Return	OK (0)	연결 성공
	False(-1)	연결 실패

Example Code	#define IP_ADDR_DSSMP "192.168.51.10"
	#define PORT_BUMPER 50011
	
	dsphal_tcp_client_t *tcp_client;
	int connect_state;
	
	tcp_client = dsphal_tcp_client_create(IP_ADDR_DSSMP, PORT_BUMPER);
	connect_state = dsphal_tcp_client_connect(tcp_client);	

3-2-3. int dsphal_tcp_client_destroy(dsphal_tcp_client_t *dsphal_tcp_client)

Function Name	void dsphal_tcp_client_destroy(dsphal_tcp_client_t *dsphal_tcp_client)	
Argument	dsphal_tcp_client	접속하고 있는 DSSP-HAL 서비스의 TCP 정보
Description	접속하고 있는 DSSP-HAL 서비스와의 연결 해제	
Return	None	
Example Code	#define IP_ADDR_DSSMP "192.168.51.10" #define PORT_BUMPER 50011 int connect_state; dsphal_tcp_client_t *tcp_client;	

	tcp_client = dsphal_tcp_client_create(IP_ADDR_DSSMP, PORT_BUMPER);
	connect_state = dsphal_tcp_client_connect(tcp_client);
	… … .
	dsphal_tcp_client_destroy (tcp_client);

3-2-4. dsphal_datalist_t *dsphal_build_root_datalist(char *format, data1, data2 …)

Function Name	dsphal_datalist_t *dsphal_build_root_datalist(char *format, …)	
Argument	Format	DSSP-HAL 데이터 구조 포멧 * variable length argument
Description	DSSP-HAL 서비스에 보낼 데이터 리스트 생성	
Return	dsphal_datalist_t	데이터 리스트 주소 값

Example Code	dsphal_datalist_t *datalist_arg;
	int lvel, rvel;
	lvel = 50;
	rvel = 50;
	
	datalist_arg = dsphal_build_root_datalist("[{i}{i}]", lvel, rvel);	

※ char *format 데이터 형식

DSSP-HAL 에서 정의된 모든 데이터의 형식은 정수형(integer), 문자열형(string)으로 구성되어 있습니다. 위에서 나타난 바와 같이, char 형식의 *format 변수는 '['로 시작해 ']' 로 끝나게 되어 있으며, 그 사이에는 정수형(integer)일 경우에는 그 개수만큼 '{i}' 가 들어가게 되고, 문자열형(string)일 경우에는 그 개수만큼 '{s}' 가 들어가도록 설계되어 있습니다.

3-2-5. void dsphal_datalist_destroy(dsphal_datalist_t *dsphal_datalist)

Function Name	void dsphal_datalist_destroy(dsphal_datalist_t *dsphal_datalist)	
Argument	dsphal_datalist	데이터 리스트 주소 값
Description	데이터 리스트가 저장된 메모리 해제	
Return	None	
Example Code	dsphal_datalist_t *datalist_arg; … … if (datalist_arg) dsphal_datalist_destroy(datalist_arg);	

3-2-6. dsphal_datalist *dsphal_request_method_call(dsphal_tcp_client_t *dsphal_tcp_client, char *method_name, dsphal_datalist_T *dsphal_datalist_arg)

Function Name	dsphal_datalist_t *dsphal_request_method_call(dsphal_tcp_client_t *dsphal_tcp_client, char *method_name, dsphal_datalist_t *dsphal_datalist_arg)	
Argument	dsphal_tcp_client	접속 할 DSSP-HAL 서비스 의 TCP 정보
	method_name	호출할 메소드 명
	dsphal_datalist_arg	데이터 리스트의 Argument
Description	DSSP-HAL 서비스에 메소드 콜(Call) 요청	
Return	None	반환값이 없는 경우
	dsphal_datalist_t	반환값이 있는 경우, 데이터 리스트 주소 값

Example Code	**A. 데이터 리스트 argument가 있는 경우**
	#define IP_ADDR_TETRA "192.168.51.10"
	#define PORT_DRIVE 50010
	
	dsphal_tcp_client_t *tcp_client;
	dsphal_datalist_t *datalist_arg;
	dsphal_datalist_t *datalist_ret;
	int lvel
	int rvel;
	lvel = 50;
	rvel = 50;
	
	tcp_client = dsphal_tcp_client_create(IP_ADDR_TETRA, PORT_DRIVE);
	dsphal_tcp_client_connect(tcp_client);
	datalist_arg = dsphal_build_root_datalist("[ii]", lvel, rvel);
	datalist_ret = dsphal_request_method_call(
	tcp_client, "VelocityControl", datalist_arg);
	
	B. 데이터 리스트 argument가 없는 경우
	dsphal_tcp_client_t *tcp_client;
	dsphal_datalist_t *datalist_ret;
	int i;
	
	tcp_client = dsphal_tcp_client_create(IP_ADDR_TETRA, PORT_BUMPER);
	dsphal_tcp_client_connect(tcp_client);
	datalist_ret = dsphal_request_method_call(
	tcp_client, "SetBumperDirMode", NULL);	

3-2-7. int dsphal_decompose_root_datalist(dsphal_datalist_t *dsphal_datalist, char *format, data1, data2 …)

Function Name	int dsphal_decompose_root_datalist(dsphal_datalist_t *dsphal_datalist, char *format,

	data1, data2, ..)	
Argument	dsphal_datalist	분해될 데이터 리스트 포인터
	format	DSSP-HAL 데이터 구조 포멧 * variable-length argument
Description	dsphal_request_method_call( ) 함수 에서 요청한 데이터 분해	
Return	OK(0)	성공
	FAIL(-1)	실패

Example Code	#define IP_ADDR_TETRA "192.168.51.10"
	#define PORT_BUMPER 50011
	
	dsphal_tcp_client_t *tcp_client;
	dsphal_datalist_t *datalist_ret;
	int i;
	int bumper_val[8];
	
	tcp_client = dsphal_tcp_client_create(IP_ADDR_TETRA, PORT_BUMPER);
	dsphal_tcp_client_connect(tcp_client);
	datalist_ret = dsphal_request_method_call(
	tcp_client, " ReadBumperArray ", NULL);
	if (datalist_ret) {
	dsphal_decompose_root_datalist(datalist_ret, "[{i}{i}{i}{i}{i}{i}{i}{i}]",
	&bumper_val[0],
	&bumper_val[1],
	&bumper_val[2],
	&bumper_val[3],
	&bumper_val[4],
	&bumper_val[5],
	&bumper_val[6],
	&bumper_val[7]);
	dsphal_datalist_destroy(datalist_ret);
	}
	dsphal_tcp_client_destroy(tcp_client);	

3-3. DSSP-HAL 요청 예제 (Example of DSSP-HAL Request)

네트워크 상에서 DSSP-HAL 내의 메소드 콜(Method call)을 할 경우, 인자(Argument)와 리턴 값(Return)의 유무 및 리턴 값의 형태에 따른 DSSP-HAL 요청(Request)의 사용 예제는 아래와 같습니다.

3-3-1. 인자 및 리턴 값이 없는 경우 (No Argument & Return)

DSSP-HAL 서비스 내의 메소드 호출 시, 인자와 리턴 값이 없는 경우의 DSSP-HAL 요청 구문은 아래 코드와 같습니다.

```
#include "dsphal.h"

#define IP_ADDR_TETRA "192.168.51.10"
```

```
#define PORT_POWER 50017

int main(void)
{
	dsphal_tcp_client_t *tcp_client;

	tcp_client = dsphal_tcp_client_create(IP_ADDR_TETRA, PORT_POWER);
	dsphal_tcp_client_connect(tcp_client);
	dsphal_request_method_call(tcp_client, "PowerDownDriveModule", NULL);
	dsphal_tcp_client_destroy(tcp_client);

	return 0;
}
```

3-3-2. 인자만 있는 경우 (No Return)

DSSP-HAL 서비스 내의 메소드 호출 시, 인자만 있고 리턴 값이 없는 경우의 DSSP-HAL 요청 구문은 아래 코드와 같습니다.

```
#include  "dsphal.h"

#define IP_ADDR_TETRA "192.168.51.10"
#define PORT_DRIVE 50010

int main(void)
{
	dsphal_tcp_client_t *tcp_client;
	dsphal_datalist_t *datalist_arg;
	int lvel;
	int rvel;
	lvel = 100;
	rvel = 100;

	tcp_client = dsphal_tcp_client_create(IP_ADDR_TETRA, PORT_DRIVE);
	dsphal_tcp_client_connect(tcp_client);
	datalist_arg = dsphal_build_root_datalist("[{i}{i}]", lvel, rvel);
	dsphal_request_method_call(tcp_client, "VelocityControl", datalist_arg);
	if (datalist_arg)
	dsphal_datalist_destroy(datalist_arg);
	dsphal_tcp_client_destroy(tcp_client);

	return 0;
}
```

3-3-3. 리턴 값만 있는 경우 (No Argument)

DSSP-HAL 서비스 내의 메소드 호출 시, 인자는 없고 리턴 값만이 있는 경우의 DSSP-HAL 요청 구문은 아래 코드와 같습니다.

```
#include <stdio.h>
#include  "dsphal.h"
```

```
#define IP_ADDR_TETRA "192.168.51.10"
#define PORT_DRIVE 50010

int main(void)
{
	dsphal_tcp_client_t *tcp_client;
	dsphal_datalist_t *datalist_ret;
	int l_encoder;
	int r_encoder

	tcp_client = dsphal_tcp_client_create(IP_ADDR_TETRA, PORT_DRIVE);
	dsphal_tcp_client_connect(tcp_client);
	datalist_ret = dsphal_request_method_call(tcp_client, "ReadEncoder", NULL);
	if (datalist_ret) {
	    dsphal_decompose_root_datalist(datalist_ret, "[{i}{i}]", &l_encoder, &r_encoder);
	    dsphal_datalist_destroy(datalist_ret);
	    printf("Encoder : %d, %d\n", l_encoder, r_encoder);
	}

	dsphal_tcp_client_destroy(tcp_client);

	return 0;
}
```

3-3-4. 인자와 리턴 값이 모두 있는 경우 (Argument & Return)

DSSP-HAL 서비스 내의 메소드 호출 시, 인자는 없고 리턴 값만이 있는 경우의 DSSP-HAL 요청 구문은 아래 코드와 같습니다. 앞에서 설명했던 경우와 다른 점은, 레이저 스캐너(Laser Rangefinder)와 같이 반환되는 데이터 리스트(datalist)의 인자가 매우 많을 경우, 코드상에서 '{i}' 를 반복해서 입력하게 되므로 DSSP-HAL 요청 구문이 비효율적이 됩니다. 따라서, 아래 코드는 효율적인 방법으로 데이터 리스트(datalist)를 파싱(Parsing)하는 DSSP-HAL 요청 구문의 사용 예제를 나타내고 있습니다.

```
#include <stdio.h>
#include  "dsphal.h"

#define IP_ADDR_TETRA "192.168.51.10"
#define PORT_RANGEFINDER 50014

int main(void)
{
	dsphal_tcp_client_t *tcp_client;
	dsphal_datalist_t *datalist_arg;
	dsphal_datalist_t *datalist_ret;
	dsphal_data_t *data; //데이터 리스트 내부의 데이터를 받아올 포인터
	int requested_resolution;
	int i;
	int rangefinder_data[1000];
```

```
        requested_resolution = 500;
        tcp_client = dsphal_tcp_client_create(IP_ADDR_TETRA, PORT_RANGEFINDER);
        dsphal_tcp_client_connect(*tcp_client);
        datalist_arg = dsphal_build_root_datalist("[{i}]", requested_resolution);
        datalist_ret = dsphal_request_method_call(tcp_client, "ReadRangeArray", datalist_arg);
        if (datalist_arg)
            dsphal_datalist_destroy(datalist_arg);
        if (datalist_ret) {
            /*데이터 리스트의 시작 포인터를 얻음 */
            data = dsphal_datalist_get_head(datalist_ret);
            for (i = 0; i < requested_resolution; i++) {
                    /*데이터 리스트로부터 정수 값을 하나 parcing*/
                    rangefinder_data[i] = dsphal_data_int_get(dsphal_data_get_data(data));
                    /*다음 데이터의 포인터를 얻음 */
                    data = dsphal_data_get_next(data);
            }
            dsphal_datalist_destroy(datalist_ret);
        }
        for (i = 0; i < requested_resolution; i++) {
            printf("%d\n", rangefinder_ data[i]);
        }
        dsphal_tcp_client_destroy(tcp_client);

        return 0;
}
```

3-4. DSSP-HAL Service 포트 구성 (Port Configuration of DSSP-HAL Service)

당사의 이동로봇 플랫폼인 'TETRA-DS III™' 의 기본품 및 옵션품에 대한 'DSSP-HAL 서비스' 의 포트 구성은 각각 표 3-1과 3-2에 나타나 있습니다. 아래의 표에 나타나 있는 DSSP-HAL 서비스 항목은 TETRA-DS III™에 장착되어 있는 기본 모듈 및 옵션품 구성에 따라 사용할 수 있는 DSSP-HAL 서비스의 항목이 추가 또는 변경될 수 있습니다. 본 매뉴얼에서는 당사 이동로봇 플랫폼인 'TETRA-DS III™' 의 기본품에 대한 DSSP-HAL 서비스의 내용만을 포함하고 있으며, 추가적인 옵션품에 대한 DSSP-HAL 서비스 자료는 당사 고객지원센터를 통해 별도로 제공 받으시기 바랍니다.

<표 3-1> DSSP-HAL Service Port Configuration of default TETRA-DS III™

DSSP-HAL Service Name	Assigned TCP Port	Related Device
drive	50010	구동 모터
bumper	50011	충돌 감지용 범퍼 센서
sensors	50012	초음파 센서
power	50017	전원 B/D
emergency	50022	비상정지 버튼

<표 3-2> DSSP-HAL Service Port Configuration of Option Parts for TETRA-DS III™

DSSP-HAL Service Name	Assigned TCP Port	Related Device
rangefinder	50014	레이저 스캐너
cam	50015	카메라

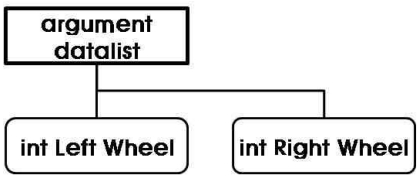
lps	50016	위치인식 센서(LPS)
gyro	50019	자이로스코프 센서
pantilt	50020	팬-틸트 모듈
exio	50024	외부 입출력 장치
stargazer	50028	위치인식 센서(StarGazer)

추가적인 장치의 사용을 원하시는 경우에는 당사 고객지원센터를 통해 해당 장치의 DSSP-HAL 서비스의 개발 지원을 받으시기 바랍니다.

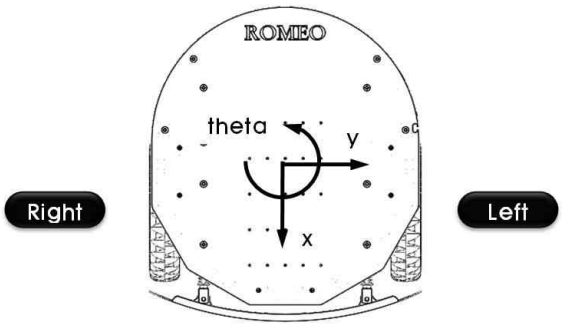
Chapter 4. DSSP-HAL 구동 서비스 (DSSP-HAL drive Service)

TCP 50010 포트를 통해 제공되는 구동 모터 관련 DSSP-HAL 서비스인 'drive' 서비스에 등록된 메소드들은 아래에 나타나 있는 바와 같습니다. TCP/IP 통신을 통해 각각의 메소드를 호출함으로써, 구동 모터의 속도를 제어하거나, 구동 모터에 장착된 엔코더의 정보, 등을 얻을 수 있습니다. 추가적인 메소드의 개발관련 사항은 당사 고객 지원센터로 문의하시기 바랍니다.

Method Name	VelocityControl	
Argument	datalist	TETAA-DS III™의 좌/우 바퀴의 속도 값을 나타내는 정수형 Left Wheel, Right Wheel 형태의 데이터. (그림 4-1 및 4-2 참조) 단위는 mm/sec.
Description	TETAA-DS III™에 장착된 좌/우 2개의 구동모터의 속도를 제어함.	
Return	None	Null
Example Code	Sample Code 참조	

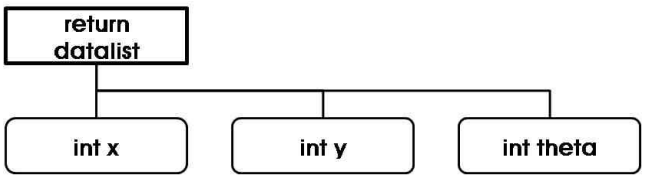


<그림 4-1> argument datalist of VelocityControl Method



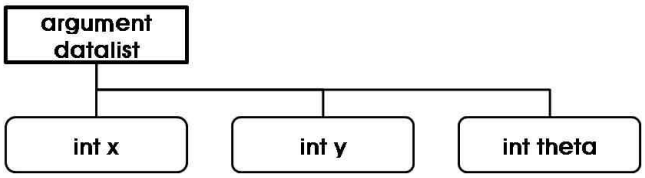
<그림 4-2> Drive Motor Configuration and Coordinates System

Method Name	ReadPosition	
Argument	None	Null
Description	Dead-Reckoning에 의한 TETAA-DS III™인 로봇 중심의 위치 값을 얻음.	
Return	datalist	정수형 x, y, theta 형태의 로봇 중심의 위치 데이터. (그림 4-3 참조) x, y의 단위는 mm, theta의 단위는 0.1degree.
Example Code	Sample Code 참조	



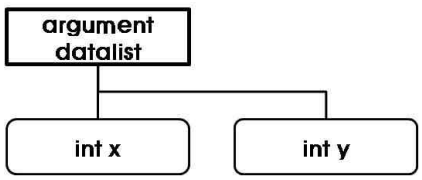
<그림 4-3> return datalist of ReadPosition Method

Method Name	ChangePosition	
Argument	datalist	갱신하고자 하는 정수형 x, y, theta 형태의 데이터. (그림 4-4 참조) x, y의 단위는 mm, theta의 단위는 0.1degree.
Description	TETRA-DS III™인 로봇 중심의 위치 값을 갱신함.	
Return	None	Null
Example Code	Sample Code 참조	



<그림 4-4> argument datalist of ChangePosition Method

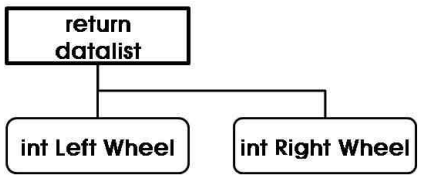
Method Name	ChangePosition2	
Argument	datalist	갱신하고자 하는 정수형 x, y 형태의 데이터. (그림 4-5 참조) x, y의 단위는 mm.
Description	TETRA-DS III™인 로봇 중심의 위치 값을 갱신함	
Return	None	Null
Example Code	Sample Code 참조	



<그림 4-5> argument datalist of ChangePosition2 Method

Method Name	ReadEncoder	
Argument	None	Null
Description	TETRA-DS III™인 로봇에 장착된 좌/우 구동 모터의 엔코더 정보를 얻음.	
Return	datalist	정수형 Left Wheel, Right Wheel 형태의 데이터. (그림 4-6 참조)

Example Code	Sample Code 참조



<그림 4-6> return datalist of ReadEncoder Method

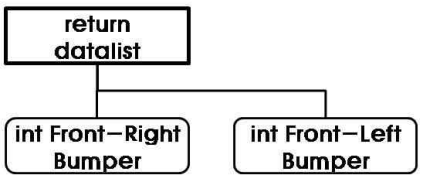
Method Name	ServoOn	
Argument	None	
Description	TETRA-DS III™인 로봇에 내장된 구동모터의 드라이버를 ON함.	
Return	None	
Example Code	Sample Code 참조	

Method Name	ServoOff	
Argument	None	Null
Description	TETRA-DS III™인 로봇에 내장된 구동모터의 드라이버를 Off함.	
Return	None	Null
Example Code	Sample Code 참조	

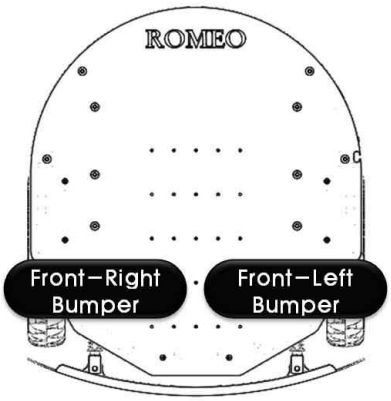
Chapter 5. DSSP-HAL 범퍼 서비스 (DSSP-HAL bumper Service)

TCP 50011 포트를 통해 제공되는 충돌 감지용 범퍼 센서 관련 DSSP-HAL 서비스인 'bumper' 서비스에 등록된 메소드들은 아래에 나타나 있는 바와 같습니다. TCP/IP 통신을 통해 각각의 메소드를 호출함으로써, 범퍼 센서의 정보, 등을 얻을 수 있습니다. 추가적인 메소드의 개발관련 사항은 당사 고객지원센터로 문의하시기 바랍니다.

Method Name	ReadBumperArray	
Argument	None	Null
Description	TETRA-DS III™인 로봇에 장착된 전방 범퍼 센서의 (충돌 감지 여부에 대한) 정보를 얻음.	
Return	datalist	정수형 Front-Left Bumper, Front-Right Bumper 형태의 데이터. (그림 5-1 및 5-2 참조) 충돌 : 1, 非충돌 : 0
Example Code	Sample Code 참조	



<그림 5-1> return datalist of ReadBumperArray Method



<그림 5-2> Bumper Sensor Configuration

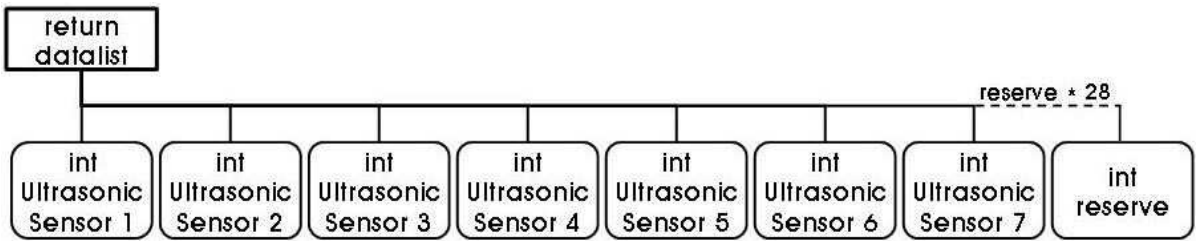
Method Name	SetBumperDirMode	
Argument	None	
Description	범퍼 센서의 작동 모드를 'Enable Mode'로 설정함. 충돌 감지용 범퍼 센서가 충돌을 감지하면 구동 모터의 동작을 멈춤.	
Return	None	Null
Example Code	Sample Code 참조	

Method Name	SetBumperOffMode	
Argument	None	Null
Description	범퍼 센서의 작동 모드를 'Disable Mode' 로 설정함. 충돌 감지용 범퍼 센서가 충돌을 감지하더라도 구동 모터는 계속 동작함.	
Return	None	Null
Example Code	Sample Code 참조	

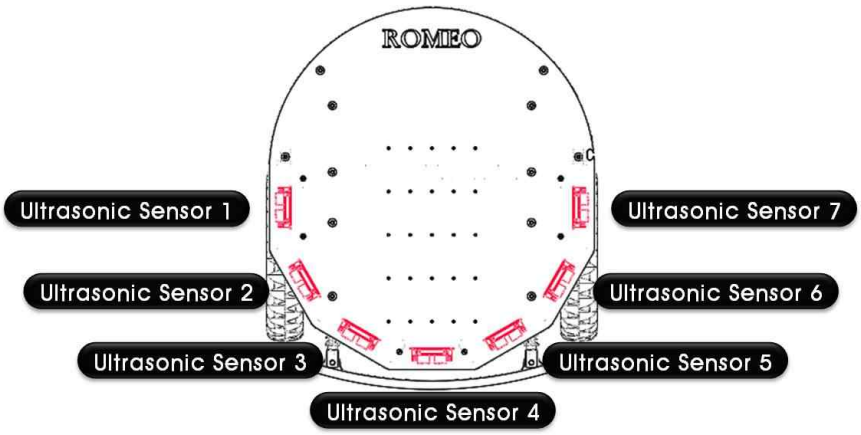
Chapter 6. DSSP-HAL 초음파 센서 서비스 (DSSP-HAL usonic_sensor Service)

TCP 50012 포트를 통해 제공되는 초음파 센서 관련 DSSP-HAL 서비스인 'usonic_sensor' 서비스에 등록된 메소드들은 아래에 나타나 있는 바와 같습니다. TCP/IP 통신을 통해 각각의 메소드를 호출함으로써, 초음파 센서의 정보, 등을 얻을 수 있습니다. 추가적인 메소드의 개발관련 사항은 당사 고객 지원센터로 문의하시기 바랍니다.

Method Name	ReadSensors	
Argument	None	Null
Description	TETAA-DS III™에 장착된 7 개의 초음파 센서에 대한 정보를 얻음. 특이사항으로 28개의 Reserve 데이터를 가짐.	
Return	datalist	감지된 장애물과의 거리를 나타내는 7 개의 초음파 센서의 정수형 데이터. (그림 6-1 및 6-2 참조) 단위는 cm. Reserved data 28개
Example Code	Sample Code 참조	



<그림 6-1> return datalist of ReadUltraSonicSensorArray Method



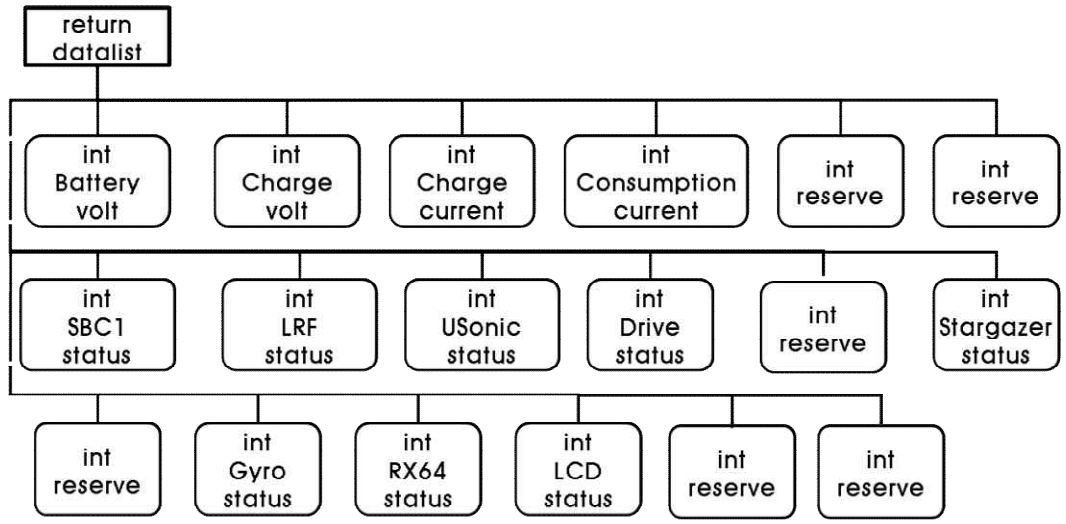
<그림 6-2> Ultrasonic Sensors Configuration

Method Name	SetMaxDistance	
Argument	datalist	1 : 1.5m, 2 : 2.0m, 3 : 2.5m, 4 : 3.0m, 5 : 3.5m, 6 : 4.0m , 7 : 4.5m, 8 : 5.0m, 9 : 5.5m (기본 값은 3.0m)
Description	Integer 형태의 1개 변수로 입력되는 변수의 값에 따라 초음파 센서가 측정하는 최대거리가 변경됨. 기본 값은 4번(3,0m)로 되어있음 .	
Return	NULL	NULL
Example Code	Sample Code 참조	

Chapter 7. DSSP-HAL 전원 서비스 (DSSP-HAL power Service)

TCP 50017 포트를 통해 제공되는 전원 시스템 관련 DSSP-HAL 서비스인 'power' 서비스에 등록된 메소드들은 아래에 나타나 있는 바와 같습니다. TCP/IP 통신을 통해 각각의 메소드를 호출함으로써, 배터리의 정보, 소비 전류, 장치들의 전원 상태 정보, 등을 얻을 수 있습니다. 추가적인 메소드의 개발관련 사항은 당사 고객 지원센터로 문의하시기 바랍니다.

Method Name	ReadPowerStatus	
Argument	None	Null
Description	시스템의 전원 관련 정보를 얻음. 첫번째 배터리 정보와 9,10번째 센서보드, 구동보드의 전원상태를 나타냄 1일경우 on, 0일경우 off 상태를 나타냄.	
Return	datalist	배터리 전압 데이터와 17개의 reserved data. (그림 8-1 참조)
Example Code	Sample Code 참조	

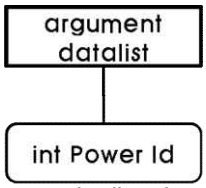


〈그림 8-1〉 return datalist of ReadPowerStatus Method

Method Name	PowerUp	
Argument	datalist	TETAA-DS III[TM]의 전원 관리 보드의 세부 항목을 나타내는 정수형 Power Id 형태의 데이터. (그림 8-2 및 표 8-1 참조)
Description	전원 관리 보드의 세부 항목의 전원 활성화	
Return	None	Null
Example Code	Sample Code 참조	

Method Name	PowerDown	
Argument	datalist	TETAA-DS III[TM]의 전원 관리 보드의 세부 항목을 나타내는 정수형 Power Id 형태의 데이터. (그

		림 8-2 및 표 8-1 참조)
Description	전원 관리 보드의 세부 항목의 전원 비활성화	
Return	None	Null
Example Code	Sample Code 참조	



<그림8-2> argument datalist of PowerUp Method

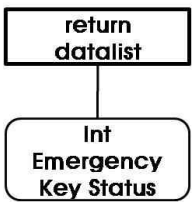
<표 8-1> argument datalist of Power Id list

Power Id	Name	Volt	Description
1	Drive	-	구동부 전원제어 (기본 시스템 사양)
2	Ultrasonic	-	초음파 센서 보드 전원 제어
3	LRF	5V	레이저 스캐너 전원 제어
4	SBC1	24~27V	SBC 전원 제어 (배터리 전압 사용)
5	Gyro	5V	Gyro 전원 제어
6	StarGazer	5V, 12V	StarGazer용 두 개의 전원 단자 동시 제어
7	Spare1	12V	여분의 12V (max. 3A) 전원 제어
8	Spare3	12V	여분의 12V (max. 3A) 전원 제어
9	Spare2	12V	여분의 12V (max. 3A) 전원 제어
10	LCD	12V	LCD 전원 제어
11	AX64	7.4V	AX64 전원 제어
12	Spare4	12V	여분의 12V (max. 3A) 전원 제어

Chapter 8. DSSP-HAL 비상정지 서비스 (DSSP-HAL emergency Service)

TCP 50022 포트를 통해 제공되는 비상정지 버튼 관련 DSSP-HAL 서비스인 'emergency' 서비스에 등록된 메소드들은 아래에 나타나 있는 바와 같습니다. TCP/IP 통신을 통해 각각의 메소드를 호출함으로써, 비상정지 버튼의 정보, 등을 얻을 수 있습니다. 추가적인 메소드의 개발관련 사항은 당사 고객지원센터로 문의하시기 바랍니다.

Method Name	ReadEmergencyKey	
Argument	None	Null
Description	비상정지 버튼의 상태 값을 얻음. 비상정지 키가 눌려지면, 메소드 호출과 관계없이 로봇에 장착된 구동 모터의 제어는 모두 차단됨.	
Return	datalist	비상정지 버튼의 상태를 나타내는 정수형 데이터. (그림 8-1 참조) 버튼 On : 1, 버튼 Off : 0
Example Code	Sample Code 참조	



<그림 9-1> return datalist of ReadEmergencyKey Method

※ 위에 설명되지 않은 옵션품의 DSSP-HAL 메소드에 대한 설명은 옵션품의 종류에 따라 매뉴얼이 별도로 제공됩니다.