

Dongbu HerkuleX Servo Library

(DRS-0101, DRS-0102, DRS-0201, DRS-0401, DRS-0402, DRS-0601, DRS-0602)

for Arduino – ver 0.2 (2014.03.28)

➔ 시작하기 앞서 알아두기

아두이노(Arduino) 보드와 허큘렉스(HerkuleX) 서보 모터를 연결하기 전에 다음 사항을 확인하세요.

(1) 허큘렉스(HerkuleX) 서보 모터 모델

허큘렉스 서보 모터의 모델에 따라 동작전압이 다릅니다. DRS-010X와 DRS-020X 모델은 7.4V에서 동작하며, DRS-040X와 DRS-060X 모델은 12V에서 동작합니다.

해당 모델에 맞는 동작 전압을 인가하십시오. (모터 파손의 원인이 됩니다.)

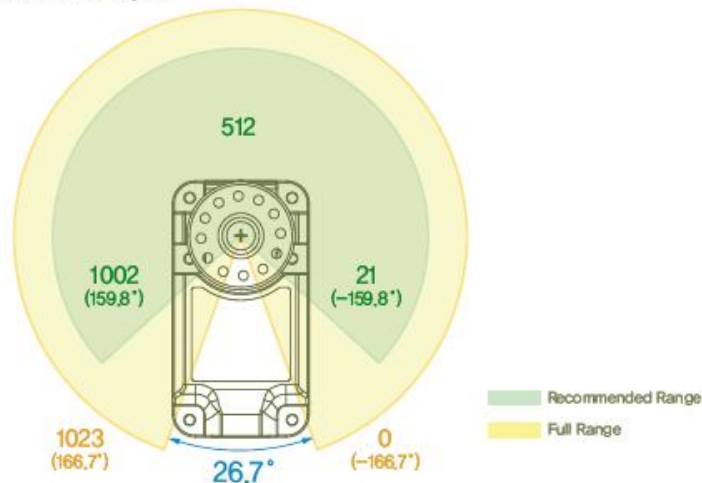
	포텐셜 미터 버전				엔코더 버전		
	DRS-0101	DRS-0201	DRS-0401	DRS-0601	DRS-0102	DRS-0402	DRS-0602
동작전압	7.4V	7.4V	12V	12V	7.4V	12V	12V

(2) 이동 범위와 토크

허큘렉스 서보 모터의 이동 범위는 다음과 같습니다. (참고로, DRS-0X01의 X는 서보 모터 토크의 세기를 나타냅니다.) 각 모델 별 상세내역은 아래 그림을 참조하십시오.

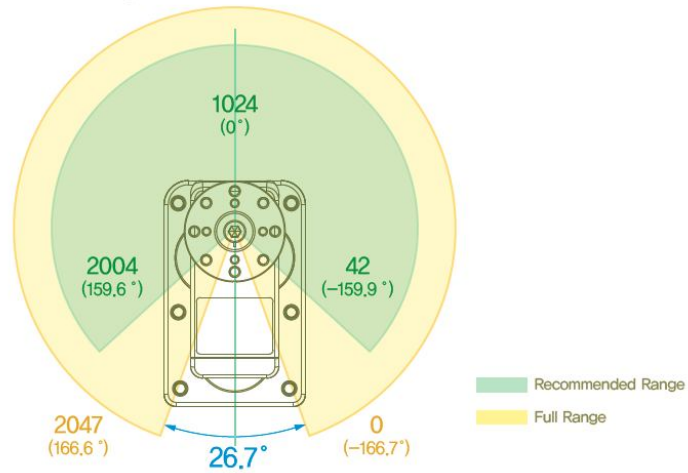
	포텐셜 미터 버전				엔코더 버전		
	DRS-0101	DRS-0201	DRS-0401	DRS-0601	DRS-0102	DRS-0402	DRS-0602
이동범위	0~1023	0~1023	0~2047	0~2047	0~32767	0~32767	0~32767
제어범위	320도	320도	320도	320도	900도	900도	900도
무한회전	-1024 ~ 1024						

■ 각도(Degree) = Position Raw Data X 0.325



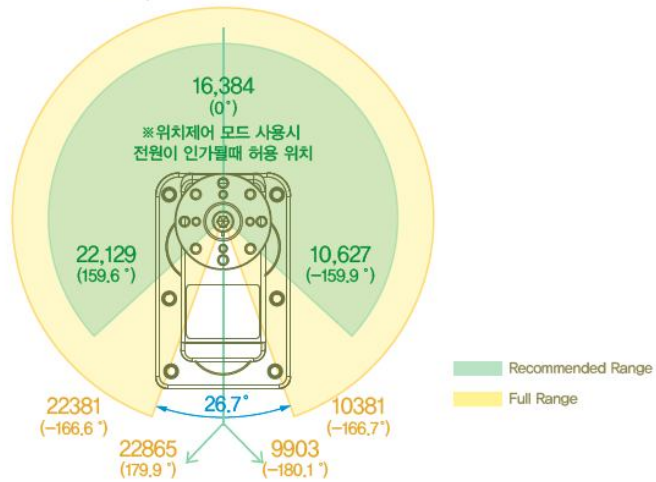
<DRS-0101, DRS-0201>

- 각도(Degree) = Position Raw Data X 0,163



<DRS-0401, DRS-0601>

- 각도(Degree) = (Position Raw Data-9903)X0,02778



<DRS-0X02 엔코더 모델>

➔ 아두이노(Arduino)와 허큘렉스(HerkuleX) 서보 모터 연결

허큘렉스를 제어하기 위하여 다양한 모델의 아두이노 보드를 사용할 수 있습니다. 다만, 아두이노 보드의 모델에 따라 다음 사항을 유의하시기 바랍니다.

(1) Arduino Uno의 경우 HerkuleX 서보 모터의 통신 속도를 57600으로 설정하세요.

그 밖의 보드에서는 115200에서 원활히 동작합니다.

	Uno	Mega	Due
권장 속도(Baudrate)	57600	115200	115200

* 허큘렉스 서보 모터의 공장 초기화 통신 속도(Baudrate)는 115200입니다. 라이브러리에 포함된 HerkuleX_Change_Baudrate 예제를 활용해서 서보 모터의 통신 속도를 설정할 수 있습니다.

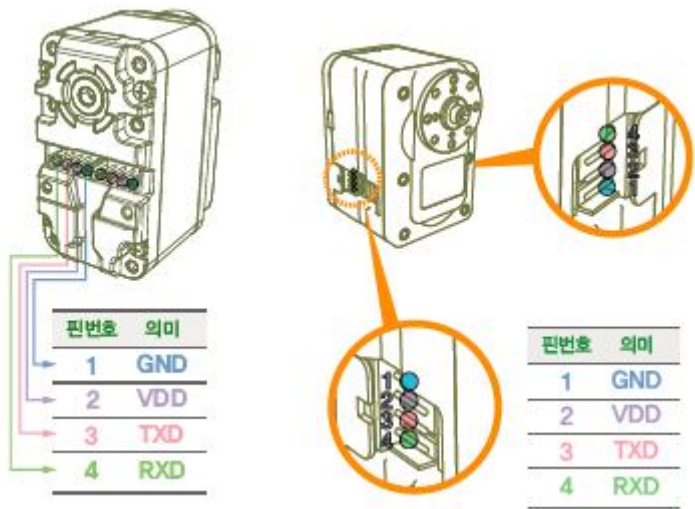
(라이브러리 폴더 내 WexampleWHerkuleX_Change_BaudrateW 참조)

(2) **Arduino Due** 는 ARM 기반 보드로서 I/O 통신의 전압 레벨이 3.3V 입니다. 허큘렉스(HerkuleX) 서보 모터의 I/O 통신 레벨이 5V 입니다. 5V I/O 핀을 보드에 바로 연결하면 Due 보드의 데미지를 입힐 수 있습니다. (특히 모터의 TX 핀을 보드의 RX 핀에 연결한 경우)

* 별도의 레벨 시프트 회로 구성이 필요할 수 있습니다

	Uno	Mega	Due
I/O 핀 전압	5V	5V	3.3V

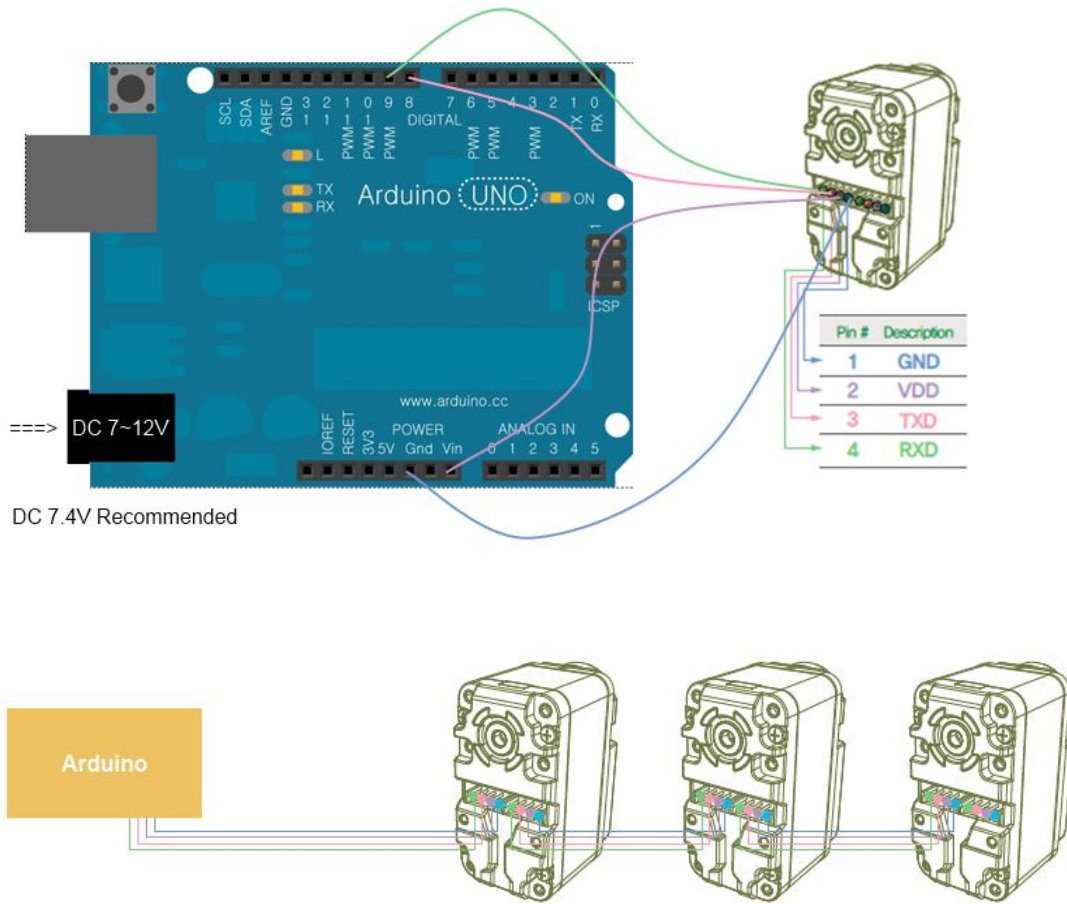
(3) 모델 별 서보 모터의 핀맵



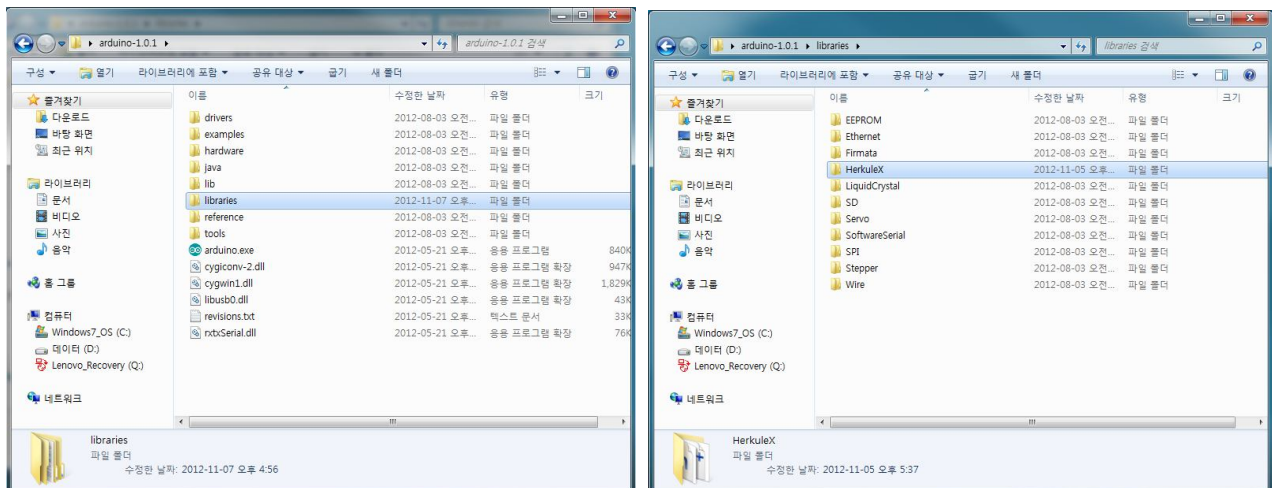
<DRS-0101, DRS-0201>

<DRS-040X, DRS-060X>

(4) Aruidno Uno와 DRS-0101의 연결 예제



(5) 허큘렉스 라이브러리(HerkuleX.zip) 파일의 압축을 풀고 아두이노 폴더\libraries\에 복사합니다.



➔ 아두이노 라이브러리 (Arduino Library)

(1) 상수

- LED 관련 상수

HERKULEX_LED_RED	- RED LED
HERKULEX_LED_GREEN	- GREEN LED
HERKULEX_LED_BLUE	- BLUE LED

- 모터 상태 관련 상수

HERKULEX_STATUS_OK	- 정상
HERKULEX_ERROR_INPUT_VOLTAGE	- 모터 입력 전압 에러
HERKULEX_ERROR_POS_LIMIT	- 모터 Pos 에러
HERKULEX_ERROR_TEMPERATURE_LIMIT	- 모터 온도 에러
HERKULEX_ERROR_INVALID_PKT	- 패킷 이상 에러
HERKULEX_ERROR_OVERLOAD	- 오버로드 에러
HERKULEX_ERROR_DRIVER_FAULT	- 드라이버 에러
HERKULEX_ERROR_EEPROM_DISTORT	- EEPROM 레지스터 에러

- Broadcast ID

HERKULEX_BROADCAST_ID	- 0xFE(254) 모든 모터에 지령 전달
-----------------------	--------------------------

(2) 함수

- 시작

```
void beginSoftwareSerial(long baudrate, uint8_t rx, uint8_t tx);  
void beginSerial(long baudrate);  
void beginSerial1(long baudrate);  
void beginSerial2(long baudrate);  
void beginSerial3(long baudrate);
```

- Torque ON, OFF

```
void torqueOn(uint8_t id);  
void torqueOff(uint8_t id);
```

- 무한회전

```
void turn(uint8_t id, int16_t pwmValue, uint8_t playtime = 0x30, uint8_t led = 0x00);  
int16_t getTurnSpeed(uint8_t id);
```

- 위치제어

```
void movePos(uint8_t id, uint16_t pos, uint8_t playtime = 0x30, uint8_t led = 0x00);  
uint16_t getPos(uint8_t id);
```

- 모터 상태 체크 및 에러 Clear

```
void init();  
void clear(uint8_t id);  
byte getStatus(uint8_t id);
```

- 엔코더 버전 (DRS-0X02) 함수

```
void compensateEncoderPos(uint8_t id);  
void resetEncoder(uint8_t id);
```

- 모터 ID 및 Baudrate 변경

```
void changeID(uint8_t old_id, uint8_t new_id);  
void changeBaudrate(uint8_t id, int baudrate);
```

(3) 함수 상세

void beginSoftwareSerial(long baudrate, uint8_t rx, uint8_t tx);	
설명	SoftwareSerial을 이용하여 HerkuleX를 시작한다 SoftwareSerial은 8Bit 프로세서만 동작한다 (ARM기반 32비트 프로세서가 장착 된 Arduino Due에서 동작안함)
인자	* baudrate (recommended) Arduino Uno -57600, Arduino Mega - 115200 * rx - Arduino board RX (HerkuleX servo TX pin) * tx - Arduino board TX (HerkuleX servo RX pin)
사용 예	HerkuleX.beginSoftwareSerial(57600);

void beginSerial(long baudrate);	
설명	Serial을 이용하여 HerkuleX를 시작한다 Arduino 보드의 0,1번 핀 보통, Serial은 PC와 연결을 위해 사용되므로, 이 함수는 거의 사용하지 않는다
인자	* baudrate - baudrate
사용 예	HerkuleX.beginSerial(115200);

void beginSerial1(long baudrate);
void beginSerial2(long baudrate);
void beginSerial3(long baudrate);

설명	Serial1~3을 이용하여 HerkuleX를 시작한다 Arduino Mega, Due에서만 이용 가능하다. HerkuleX와 연결시 Arduino 보드의 RX, TX핀을 각각 HerkuleX의 TX핀과 RX핀에 연결한다.
인자	* baudrate - 115200(recommended)
사용 예	HerkuleX.beginSerial1(115200); HerkuleX.beginSerial2(115200); HerkuleX.beginSerial3(115200);

void torqueOn(uint8_t id);	
설명	지정된 ID의 모터를 Torque ON한다. Torque On이 되어있어야만, 모터가 동작한다. 모터 ID를 0xFE(Broadcast ID)로 지정하면, 모든 모터에 Torque ON한다.
인자	* id - HerkuleX Servo ID (0~253 Broadcast ID는 254)
사용 예	HerkuleX.torqueOn(253);

void torqueOff(uint8_t id);	
설명	지정된 ID의 모터를 Torque OFF한다. Torque OFF되면, 모터가 동작하지 않는다. 모터 ID를 0xFE(Broadcast ID)로 지정하면, 모든 모터에 Torque OFF한다.
인자	* id - HerkuleX Servo ID (0~253 Broadcast ID는 254)
사용 예	HerkuleX.torqueOff(253);

void turn(uint8_t id, int16_t pwmValue, uint8_t playtime = 0x30, uint8_t led = 0x00);	
설명	모터를 무한회전으로 동작한다.
인자	* id - HerkuleX Servo ID (0~253 Broadcast ID는 254) * pwmValue - 회전속도 (-1023 ~ 1023) -값이면 시계방향 회전 CW(Clock Wise) +값이면 반시계방향 회전 CCW(Counter Clock Wise) * playtime - 회전속도에 도달하는 시간이다 playtime값에 11.2ms를 곱한 시간으로 동작한다. 1: 11.2ms, 2: 22.4ms ... 10: 112ms 값을 지정하지 않으면 0x30으로 동작 (16*3*11.2ms = 537.6ms) * led - 모터 동작 시 LED를 제어한다 값을 지정하지 않으면 LED를 켜지 않는다.
사용 예	HerkuleX.turn(253, 500, 10, HERKULEX_LED_BLUE); HerkuleX.turn(254, -500, 10, HERKULEX_LED_RED HERKULEX_LED_GREEN);

	HerkuleX.turn(1, -300)
--	------------------------

int16_t getTurnSpeed(uint8_t id);	
설명	지정된 ID에 해당하는 모터의 현재 회전속도를 반환한다.
인자	* id - HerkuleX Servo ID (0~253)
사용 예	HerkuleX.getTurnSpeed(253);
리턴 값	현재 모터의 회전 속도를 반환한다.

void movePos(uint8_t id, uint16_t pos, uint8_t playtime = 0x30, uint8_t led = 0x00);	
설명	모터를 대상 위치로 이동한다.
인자	* id - HerkuleX Servo ID (0~253 Broadcast ID는 254) * pos - 대상 위치 (0~1023 512는 정중앙) * playtime - 대상 위치에 도달하는 시간이다 playtime값에 11.2ms를 곱한 시간으로 동작한다. 1: 11.2ms, 2: 22.4ms ... 10: 112ms 값을 지정하지 않으면 0x30으로 동작 (16*3*11.2ms = 537.6ms) * led - 모터 동작 시 LED를 제어한다. 값을 지정하지 않으면 LED를 켜지 않는다.
사용 예	HerkuleX.movePos(253, 235, 50, HERKULEX_LED_BLUE); HerkuleX.movePos (254, 768, 100, HERKULEX_LED_RED HERKULEX_LED_GREEN); HerkuleX.movePos(1, 512)

int16_t getPos(uint8_t id);	
설명	지정된 ID에 해당하는 모터의 현재 위치를 반환한다.
인자	* id - HerkuleX Servo ID (0~253)
사용 예	HerkuleX.getPos(253);
리턴 값	현재 모터의 위치를 반환한다.

void clear(uint8_t id);	
설명	지정된 ID에 해당하는 모터의 에러를 Clear한다 -> 모터 에러 상태에서는 RED LED가 깜빡인다.
인자	* id - HerkuleX Servo ID (0~253 Broadcast ID는 254)
사용 예	HerkuleX.clear(253); HerkuleX.clear(254);

byte getStatus(uint8_t id);	
설명	지정된 ID에 해당하는 모터의 현재 상태를 반환한다.

인자	* <code>id</code> - HerkuleX Servo ID (0~253)
사용 예	<code>HerkuleX.getStatus(253);</code>
리턴 값	<p>현재 모터의 상태를 반환한다.</p> <pre> static byte HERKULEX_STATUS_OK = 0x00; static byte HERKULEX_ERROR_INPUT_VOLTAGE = 0x01; static byte HERKULEX_ERROR_POS_LIMIT = 0x02; static byte HERKULEX_ERROR_TEMPERATURE_LIMIT = 0x04; static byte HERKULEX_ERROR_INVALID_PKT = 0x08; static byte HERKULEX_ERROR_OVERLOAD = 0x10; static byte HERKULEX_ERROR_DRIVER_FAULT = 0x20; static byte HERKULEX_ERROR_EEPREG_DISTORT = 0x40; </pre>

<code>void changeID(uint8_t old_id, uint8_t new_id);</code>	
설명	서보 모터의 ID를 변경한다
인자	<p>* <code>old_id</code> - 현재 ID (0~253)</p> <p>* <code>new_id</code> - 변경할 ID (0~253)</p>
사용 예	<code>HerkuleX.changeID(253, 1);</code>

<code>void changeBaudrate(uint8_t id, int baudrate);</code>	
설명	모터의 통신속도(Baudrate)를 변경한다.
인자	<p>* <code>id</code> - HerkuleX Servo ID (0~253)</p> <p>* <code>baudrate</code> - 57600, 115200 (두 가지 속도만 지원)</p>
사용 예	<code>HerkuleX.changeBaudrate(253, 57600);</code>