

[비전인식기반 모듈형 인공지능 로봇 콘텐츠 개발]

6주차

우송대학교 IT센터
응용로봇SW 전임교수 : 황동하
010-2512-6818
mrt2020@daum.net

과 제 명	비전인식기반 모듈형 인공지능 로봇 콘텐츠 개발
<p>1. 목표</p> <ul style="list-style-type: none"> - 대부분의 인공지능 활용 교육 프로그램이 SW 개발에 치중되어 있어 교육생들의 흥미가 떨어짐 - 로봇과 IoT를 접목한 AI 교육은 실생활 문제를 해결하는데 보다 더 적합하고 교육생들의 만족도도 높은 편임 - 그러나, 로봇 구매비용이 수백~수천만원에 이르고 설치 환경도 까다로워 도입이 쉽지 않은 상황 - 따라서, 한 종류의 모터로 다양한 형태의 로봇 조립이 (바퀴형과 관절형) 가능하고 오픈소스 기반의 아두이노, Pixy 카메라 등을 활용하여 프로그램 확장성이 있는 인공지능로봇 콘텐츠 개발이 필요함 <p>2. 개발내용 및 결과</p> <ul style="list-style-type: none"> ◦ 개발 내용 <ul style="list-style-type: none"> - Pixy2 카메라 접속 모듈 장치 - pixy2의 인식 결과를 SPI통신을 이용하여 전송 받아 콘텐츠를 구현하는 제어보드 ICM-PA01하드웨어의 제작 - 제어보드는 콘텐츠의 처리 결과를 로봇으로 전달하도록 함 - Pixy2와 ICM-PA01간의 결속을 담당하는 브라켓 구조물 제작 ◦ 개발 결과 <ul style="list-style-type: none"> - ICM-PA01을 이용한 로봇 3종 대회(달리기, 장애물 회피, 마라톤) 용 샘플 코드 1종 작성 및 설명서 배포 - ICM-PA01 모듈을 이용한 로봇 제어 강좌 무료 개설 시험 운영 - 강좌를 통해 로봇 3종 대회에 참가 유도 	



[그림] 개발 결과 로봇 3종 경기 대회 개최

3. 과제결과물

- ICM 모듈 6개
- ICM 모듈기반 로봇 3종 대회 대비용 소스 코드 3종

4. 사업성과

- 기술적 성과
 - 로봇의 베이스 제작의 어려움으로 응용 로봇 S/W와 콘텐츠 개발이 뒤처지게 되는 문젯점을 해결 할 수 있는 Secondary Development 모듈을 개발함.
 - 향후 다기능 시각용 카메라 모듈과 음성 인식, 출력 모듈을 활용하여 AI의 품질을 높일 수 있는 방법을 제시 함.
- 경제적 성과
 - 고가의 AI 비전 모델을 대체 할 수 있어 교육 및 취미 활동을 통한 인공지능 로봇 관련 인력의 폭을 넓히는 계기가 됨.
 - 교육 현장의 인공지능 로봇 수요에 대처 할 수 있음.

1. 목표 및 추진배경

가. 목표

- 1) 기존의 휴머노이드 로봇 또는 바퀴구동형 로봇 등에 부착하여 활용 가능한 인공지능 기반의 모듈을 개발 하고자 함.
- 2) 아두이노 기반의 마이크로 프로세서를 이용하여 제어 할 수 있도록 함.

나. 추진 배경

- 1) 기존 휴머노이드 로봇 또는 바퀴구동형 로봇은 단순 리모콘형 이거나 센서에 의존하는 형태가 많음. 리모콘으로 조종하는 형태나 센서에만 의존 하는 로봇 학습 및 활용에 제한이 있음. 이에 시기반의 시각 인공지능을 탑재한 디바이스의 개발 및 활용을 통해 다양한 방식의 자율형 서비스 로봇으로 활용 할 수 있는 콘텐츠 제작 및 교육이 가능이 가능하게 하도록 함.

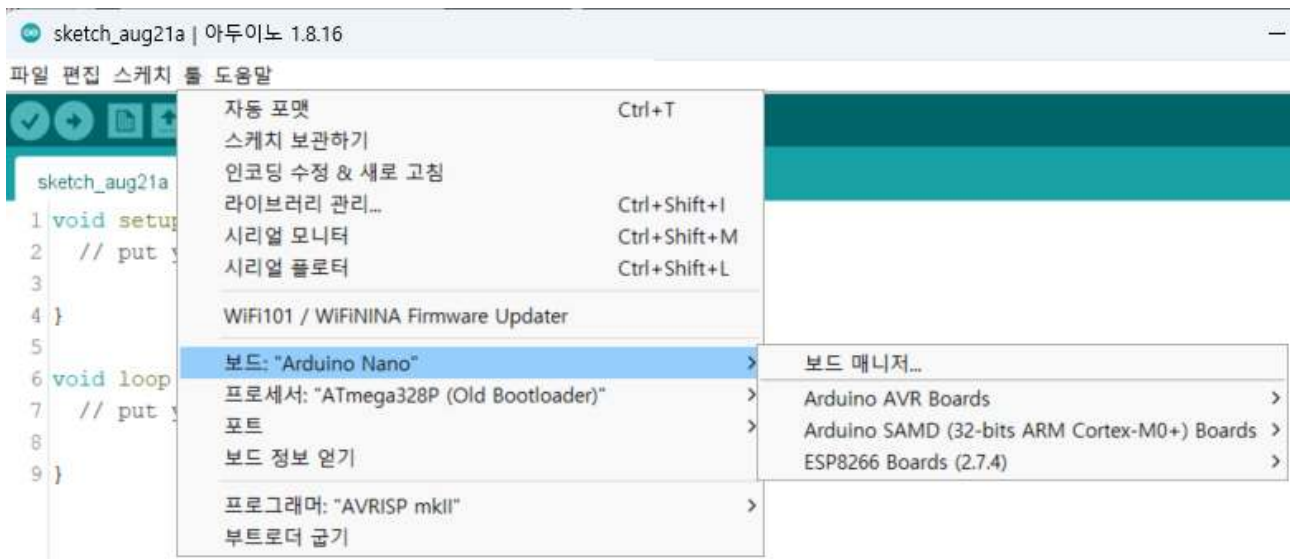
다. 산업체와 대학의 산학 공동 진행 필요성

- 1) 개발하는 디바이스 보드로 아두이노 코딩 실습이 가능한 Arduino Nano를 채용하여 교육용으로 활용이 가능하도록 하고자 함.
- 2) 실제 사용하게 될 학생들을 프로젝트에 참여시켜 적응 정도를 파악 하고자 함.
- 3) 실제 학생들이 제작한 응용 SW를 통해 개발 결과물의 확장 가능성을 가늠해 볼 수 있음.

2. 개발 내용 및 결과

가. ICM-PA01의 주요 기능

- 1) 보드 및 프로세서 정의
 - 보드 : 아두이노 호환 ATMEGA NANO
 - 프로세서 : ATmega328P(Pld Bootloader)
 - 프로그래머 : AVTISP mkII



2) 기능

- 아두이노 IDE를 사용 할 수 있음.
- PIXY2 카메라 모듈의 기능을 모두 사용 할 수 있음.

색상 지정 기능	지정 색상 물체 추적 기능	벡터기능을 이용한 라인추적

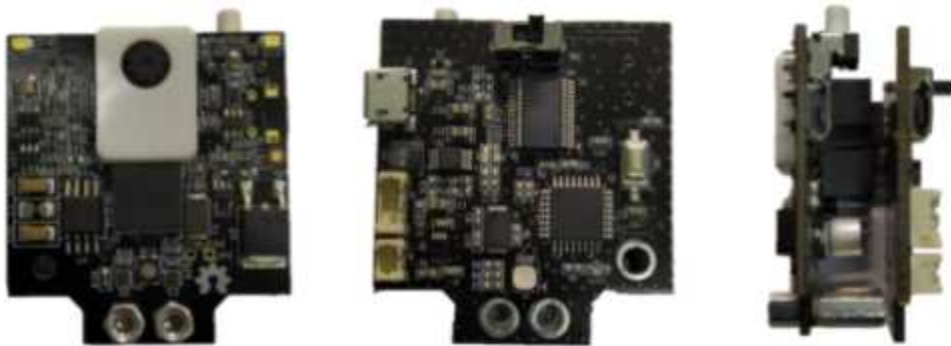
- PIXY2 카메라 모듈의 Detection 내용을 SPI 통신을 통해 아두이노에 전송 할 수 있음.

3) Spec

MCU	ATMEGA328P
Operating Voltage	3.3V, 5V
Input Voltage	3.3V, 5V
Digital I/O pin	6+SPI+UART
DC Current per I/O pin	40mA
Flash Memory	32K of which 0.5KB used by bootloader
SRAM	2KB
EEPROM	1KB
Clock Speed	16MHz

4) ICM-PA01 하드웨어 구성

- 카메라 모듈(앞)과 아두이노호환 제어보드(뒤) 두 보드로 구성 되어 있음.



나. 제작 과정

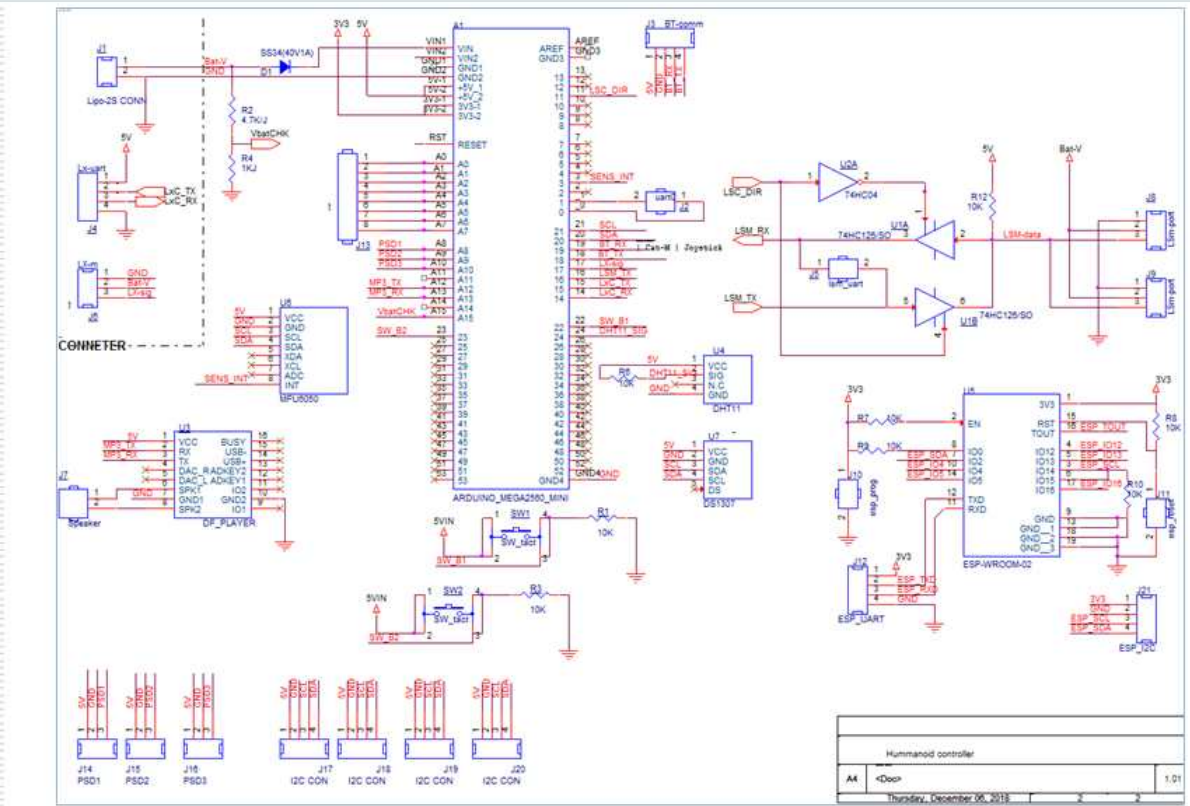
- 1) 팀 셋업 및 설계를 위한 외부 전문가 미팅 및 개발 회의



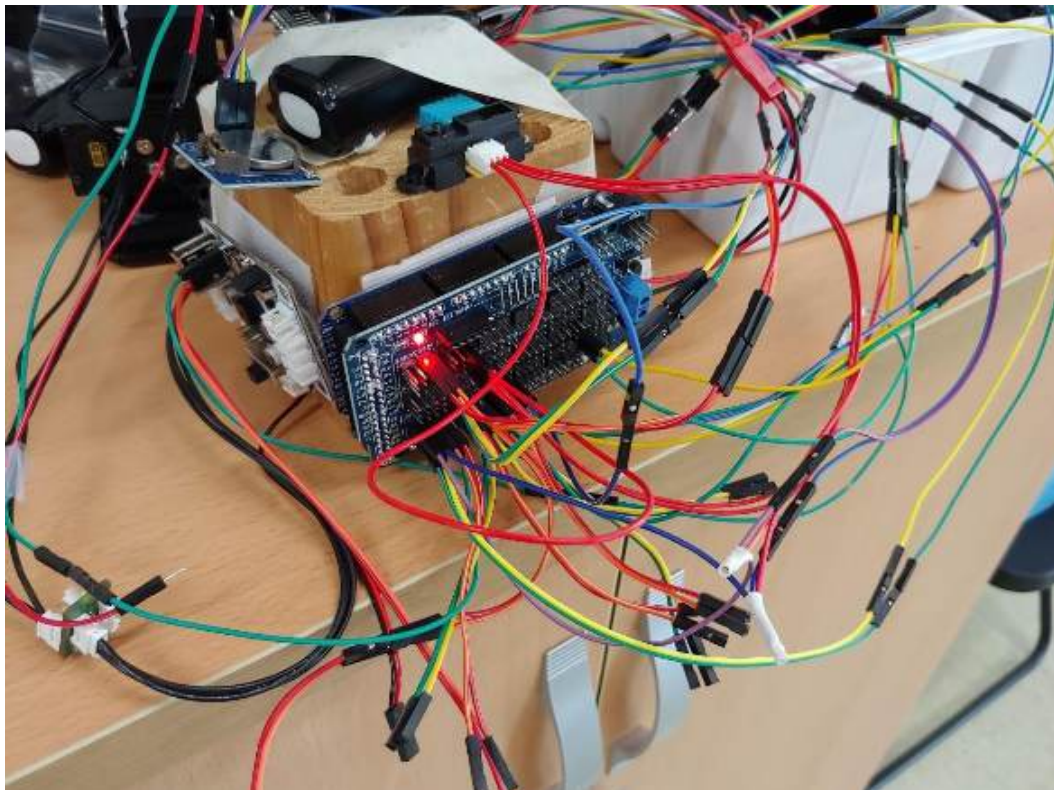
[그림] 팀 구성 및 제작 회의

- 2) ICM-PA01 컨트롤러 제작을 위한 PCB 회로도 작성 및 회로구성 테스트

Layer 3-1: 컨트롤러 회로도



[그림] ICM-PA01용 컨트롤러 회로도면



[그림] 동작 시험을 위한 회로 구성과 테스트

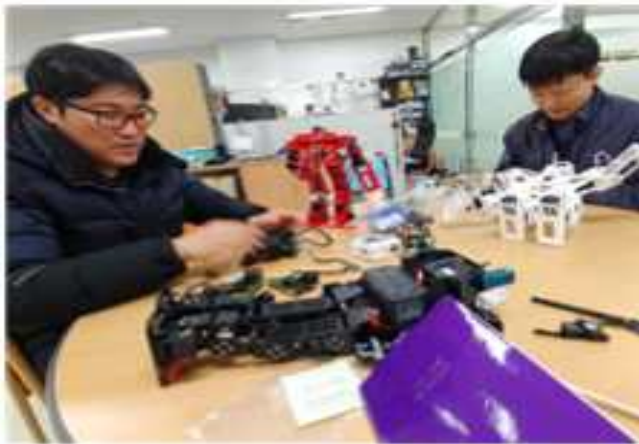
ICM-PA01 제품 용도와 특성에 관한 관련 회의를 통해 SPEC을 결정하였음.

ICM-PA01 컨트롤러를 제작 하기 위한 PCB 제작 회로도를 작성 함.

ICM-PA01 컨트롤러의 모든 기능을 구현 가능한지 사전 점검을 위해 회로를 구성하고 기능을 구현 함.

다. 3D디자인(모형 제작)

1) 3D 프린터를 이용한 휴머노이드 제작



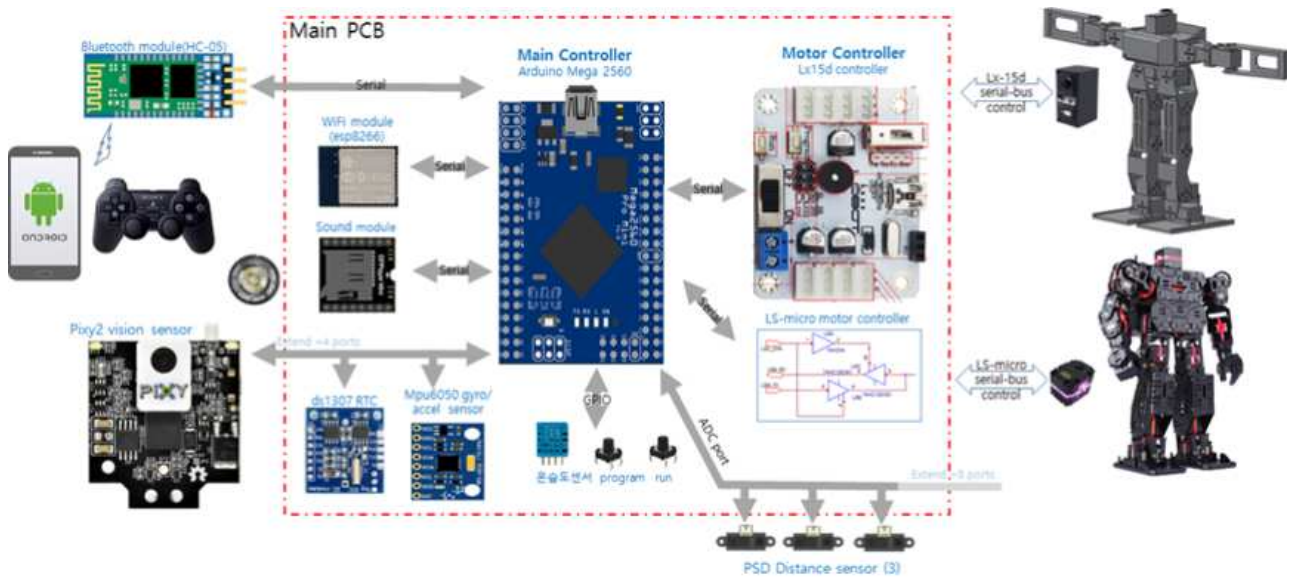
[그림] 3D 프린팅한 휴머노이드 모형(움직이지 않음)

로봇에 장착하기 위해 움직이지 않는 3D모형을 제작 휴머노이드 로봇과의

결착성 등을 검증.

라. 참여 학생들을 위한 멘토링과 콘텐츠 제작을 위한 교육 내용

- 1) 실제 동작중인 휴머노이드 로봇에 적용 하기 위하여 로봇의 프로토콜과 동작 번호를 알려주고 로봇에 명령어를 전달하여 동작 할 수 있도록 지도함.



[그림] 실제 동작중인 로봇에의 응용 예시도

2) 프로토콜

1. 공통 조건

- 통신 조건에 다른 지연 시간을 고려하여 전송 속도를 정함.

CONTROLLER - LSM			HOST to CTRL (Statement)			Packet Total		
Baudrate	bps	115200	PacketSize	byte	12	PacketSize	byte	24
1ByteT	us	86.81	1PacketT	us	1041.67	1PacketT	us	2083.33
			HOST from CTRL (Response)					
			PacketSize	byte	12			
			1PacketT	us	1041.67			
CONTROLLER - LSM			CTRL to LSM (Instruction)			Packet Total		
Baudrate	bps	1000000	PacketSize	byte	8	PacketSize	byte	16
1ByteT	us	10.00	1PacketT	us	80.00	1PacketT	us	160.00
			CTRL from LSM (Response)			NumOfLSM	EA	18
			PacketSize	byte	8	1FrameT	ms	2.88
			1PacketT	us	80.00	1FramFeq	Hz	347.22222

- 각 명령을 전달 하기 위한 최소의 시간을 고려한 전송 속도를 확정함.

command	4						EA	18	
CTRL Protocol	:	8	x	EA			=	144	FIXED
Host Protocol	:	command	x	EA	+	12	=	84	ADJ

2. 데이터 체인 방식 서보 모터 전송 규약

- LS868 데이터체인 방식 서보 모터 제어를 위한 프로토콜을 이해하고 실제 명령어를 전달하여 개별적으로 또는 그룹으로 동작 하는 방법을 습득하도록 함.
- 프로토콜을 8Byte 구조로 이루어져 있으며 아래와 같음.

Header	Prduct ID	Module ID	Instruction	Parameter1	Parameter2	Parameter3	Checksum
Header	0xFF						
Prduct ID	Controller		0x10	+ Model Version (0x00 ~ 0x1F)			
	Display		0x30				
	Comm		0x50	+ Model Version (0x00 ~ 0x3F)			
	Sensor		0x70				
	Actuator		0XB0				
Module ID	Each ID		0~253	0x00 ~ 0xFD			
	BroadCast ID		254	0xFE			
Instruction	DATA GET		0x00	+ Instruction Code (0x00 ~ 0x3F)			
	DATA SET		0x40				
	ACTION		0x80				
	MSB 2bit : OpCode			LSB 6bit : InstructionCode			
Parameter1~3	Depends on Instruction						
Checksum	(ModuleID + Instruction + Prameter1 + Parameter2 + Parameter3)						

- Product ID중 Actuator는 다양한 서보모터를 지원하고 있습니다.

Product ID	LSM-micro	0xB0
	LSM-mini	0xC0
	LSM-middle	0xD0
	LSM-mega	0xE0

사용예정인 모터는 LSM-micro 버전입니다.

- Module ID 는 0x00부터 0xFD까지 총 253개의 모터를 제어 할 수 있습니다.
전체 모터에 대한 브로드캐스트 명령어 전달은 0xFE를 사용할 수 있습니다.

- 컨트롤러와 각 서보 모터간의 명령어(Instruction Code)에 대한 상세 내용을 이해합니다.

DATAGET	0x00	+ Instruction Code (0x00 ~ 0x3F)
DATASET	0x40	
ACTION	0x80	
MSB2bit : OpCode		LSB6bit : InstCode (0~63)

명령어는 상위(MSB) 2Bit를 이용하여 GET / SET / ACTION을 정함.

다름 하위(LSB) 6Bit를 이용하여 실제 명령어를 전달 합니다.

DATA GET_Instruction	OpCode	InstCode	Packet				TXD Direction
			Instruction	Parameter1	Parameter2	Parameter3	
Get Module ID	0x00	1	0x01	0	0	0	CTRL → LSM
			0x01	0	Module ID (Note.1)		LSM → CTRL
Get Now Position		2	0x02	0	0	0	CTRL → LSM
			0x02	0	Now Position (Note.2)		LSM → CTRL
Get Target Position		3	0x03	0	0	0	CTRL → LSM
			0x03	0	Target Position (Note.2)		LSM → CTRL
Get Now Current		4	0x04	0	0	0	CTRL → LSM
			0x04	0	Now Current [mA] (Note.3)		LSM → CTRL
Get Limit Current		5	0x05	0	0	0	CTRL → LSM
			0x05	0	Limit Current [mA] (Note.3)		LSM → CTRL
Get NowTemperature		6	0x06	0	0	0	CTRL → LSM
			0x06	0	Now Temperature [°C] (Note.4)		LSM → CTRL
Get LimitTemperature		7	0x07	0	0	0	CTRL → LSM
			0x07	0	Limit Temperature [°C] (Note.4)		LSM → CTRL
Get Gain P		8	0x08	0	0	0	CTRL → LSM
			0x08	0	Gain P (Note.5)		LSM → CTRL
Get Gain I		9	0x09	0	0	0	CTRL → LSM
			0x09	0	Gain I (Note.5)		LSM → CTRL
Get Gain D		10	0x0A	0	0	0	CTRL → LSM
			0x0A	0	Gain D (Note.5)		LSM → CTRL
Get Control Margin		11	0x0B	0	0	0	CTRL → LSM
			0x0B	0	Control Margin (Note.6)		LSM → CTRL
Get ForwardControl Value		12	0x0C	0	0	0	CTRL → LSM
			0x0C	0	Forward Control Value (Note.7)		LSM → CTRL
Get Voltage		13	0x0D	0	0	0	CTRL → LSM
			0x0D	0	Voltage [mV] (Note.8)		LSM → CTRL
Get Boundary Max		14	0x0E	0	0	0	CTRL → LSM
			0x0E	0	Boundary Max (Note.9)		LSM → CTRL
Get Boundary Min		15	0x0F	0	0	0	CTRL → LSM
			0x0F	0	Boundary Min (Note.9)		LSM → CTRL
Get LED Value		16	0x10	0	0	0	CTRL → LSM
			0x10	Value R (Note.10)	Value G (Note.10)	Value B (Note.10)	LSM → CTRL
Get Model Version		17	0x11	0	0	0	CTRL → LSM
			0x11	0	Model Version (Note.11)		LSM → CTRL
Get FW Version		18	0x12	0	0	0	CTRL → LSM
			0x12	0	FW Version (Note.12)		LSM → CTRL
Get Comm Baudrate		19	0x13	0	0	0	CTRL → LSM
			0x13	Comm Baudrate [BPS] (Note.13)			LSM → CTRL
Get ZeroCompValue		20	0x14	0	0	0	CTRL → LSM
			0x14	0	ZeroCompValue (Note.14)		LSM → CTRL

[그림] 데이터 읽기 명령어 전송과 응답과정 - 양방향 통신

DATA SET_Instruction	Opcode	InstCode	Packet				TXD Direction	
			Instruction	Parameter1	Parameter2	Parameter3		
Set Factory Reset	0x40	0	0x40	0x53	0x46	0x52	CTRL → LSM	
Set Module ID		1	0x41	x	New ID (Note.1)			
Set Sync Target Position		2	0x42	Sync Tq/Spd	SyncTarget Position (Note.2)			
Set Sync LED Control Value		3	0x43	Value R (Note.10)	Value G (Note.10)	Value B (Note.10)		
Set Limit Current		4	0x44	x	Limit Current [mA] (Note.3)			
Set Limit Temperature		5	0x45	x	Temperature [°C] (Note.4)			
Set Gain P		6	0x46	x	Gain P (Note.5)			
Set Gain I		7	0x47	x	Gain I (Note.5)			
Set Gain D		8	0x48	x	Gain D (Note.5)			
Set Control Margin		9	0x49	x	Control Margin (Note.6)			
Set ForwardControl Value		10	0x4A	x	Forward Control Value (Note.7)			
Set Boundary Max		11	0x4B	x	Boundary Max (Note.9)			
Set Boundary Min		12	0x4C	x	Boundary Min (Note.9)			
Set Comm Baudrate		13	0x4D	Comm Baudrate [BPS] (Note.13)				
Set ZeroCompValue		14	0x4E	x	ZeroCompValue (Note.14)			

[그림] 단방향 데이터 셋팅 명령어

ACTION_Instruction	Opcode	InstCode	Packet				TXD Direction
			Instruction	Parameter1	Parameter2	Parameter3	
EEP PROG Config Data	0x80	1	0x81	1	2	3	CTRL → LSM
Action Move T_Position		2	0x82	Torque	Target Position (Note.2)		
Action Move S_Position		3	0x83	Speed	Target Position (Note.2)		
Action LED Control		4	0x84	Value R (Note.10)	Value G (Note.10)	Value B (Note.10)	
Action Sync Move Position		5	0x85	1:Torq/2:Spd	1	1	
Action Sync LED Control		6	0x86	1	1	1	
Action Wheel Mode		7	0x87	1:CW/2:CCW	Wheel Speed [%] (Note.15)		
Action Active Stop		8	0x88	x	x	x	
Action Passive Stop		9	0x89	x	x	x	

[그림] 단방향 Action Instruction

No	Item	Unit	Range	Code	Product Default	Factory Reset Value
1	ID	-	0~253	2byte U_Int		0
2	Position	-	-500 ~ 500	2byte Int	Center Positin 0	
3	Current	[mA]	0~1000	2byte U_Int	Limit Current 1000	Limit Current 1000
4	Temperature	[°C]	0~80	2byte U_Int	Limit Temperature 60	Limit Temperature 60
5	Gain PID	-	Each 0~254	2byte U_Int	P:40 / I:0 / D:30	P:40 / I:0 / D:30
6	Control Margin	-	0 ~ 1000	2byte U_Int	0	0
7	Forward Control Value	-	0 ~ 1000	2byte U_Int	40	40
8	Voltage	[mV]	-	2byte U_Int	-	-
9	Boundary Min/Max	-	-500 ~ 500	2byte Int	Min:-500 / Max:500	Min:-500 / Max:500
10	LED RGB	-	0~250	2byte U_Int	-	-
11	Model Version	-	-	2byte U_Int	0x80	0x80
12	FW Version	-	-	2byte U_Int	-	-
13	Baudrate	[BPS]	2400~1000000	3byte U_Int	1000000	1000000
14	ZeroCompValue	-	-12 ~ 12	2byte Int	-	0
15	Wheel Speed	[%]	0 ~ 100	2byte U_Int	-	-

[그림] 각 파라메타의 값 범위

3. 아두이노와 통신을 위한 UART 통신 규약

Packet Size : 12 Byte ~ (12+[N]) Byte			Header 4 byte	Destination Address 2 byte	Source Address 2 byte	Statement 2 byte	Para Length 1 byte	Parameters Length [N] byte	Checksum 1 byte
Header	Header1	0xFF		Destination Address	Controller	Addr H Addr L	Each Addr 1~65535	BroadCast 0	
	Header2	0xFF							
	Header3	0x4C		Source Address	Host	Addr H Addr L			
	Header4	0x53							
Statement	Statement1	Type		Length Para	Para Length Parameters	Length [N] [N] byte	※ Depends on Statement		
	Statement2	Code							
Checksum			= SrcAddr H + SrcAddr L + Statement1 + Statement2 + Length (+ Para1 + Para... + ParaN)						

[그림] 통신 규약 UART

- 통신 규약은 115200bpx의 속도를 갖는 12Byte ~ (12 + n) Byte의 길이로 정의 되었음.
- Header는 반드시 지켜야 함.
- Destination Address와 Source Address는 0x00으로 채워도 무방함.
- Statement는 두가지로 구성됨

Statement	Statement1	Type
	Statement2	Code

- Para Length는 실제 파라메타의 개수를 기록함.

Length	Para Length	Length [N]	※Depends on Statement
Para	Parameters	[N] byte	

- Checksum은 아래 표에 기술된 각 필드의 값을 모두 Or 연산한 값.

Checksum	= SrcAddr H + SrcAddr L + Statement1 + Statement2 + Length (+ Para1 + Para... + ParaN)
----------	--

- State의 구성 요소는 **Type**와 Instruction **Code**로 2Byte로 구성 되어 있음.

Type	DATA GET	0x10
	DATA SET	0x20
	EXECUTION	0x30
	Files Management	0x40

[그림] UART통신 프로토콜 Type 유형

- Data Get Code의 명령어 처리 순서

< DATA GET > Process	Direction \ Packet		Header	Dest Addr	Src Addr	Statement	Length	Parameter N	Checksum
	HOST	LSC (Command)	Header	LSC Addr	HOST Addr	Statement	ParaLength	Para	Checksum
	LSC	HOST (Response)	Header	HOST Addr	LSC Addr	Statement	RespParaLength	RespPara	Checksum

[그림] Data Get Code의 명령어 처리 Process

– Data get Code 의 상세 명령어

DATA GET	Type	Code	Para Length	Para1	Para2	Para3	Para4	Para5	Para6	Para7	Para8	Para9	...
Get LSC Version	0x10	1	0										
		3	0	HW Version	FW Major	FW Minor							
Get BAT Voltage		2	0	BAT VoltageH	BAT VoltageL								
Get Module ID (only 1EA Connected)		3	0										
		1	1	LSM ID									
		1	1	Last ID									
Get Nbr of LSM(Connected)		4	1	NumberOfLSM									
		1	1	LSM ID	1st	...	LSM ID (N-1)th	LSM ID (N)th					
Get LSMs NowPosition		5	3	ID 1st	ID 1st PosH	ID 1st PosL	...	ID (N)th	ID (N)th PosH	ID (N)th PosL			
		1	1	LSM ID	1st	...	LSM ID (N-1)th	LSM ID (N)th					
Get LSMs NowVoltage		6	3	ID 1st	ID 1st VoltH	ID 1st VoltL	...	ID (N)th	ID (N)th VoltH	ID (N)th VoltL			
		1	1	LSM ID	1st	...	LSM ID (N-1)th	LSM ID (N)th					
Get LSMs NowCurrent		7	3	ID 1st	ID 1st NowCurH	ID 1st NowCurL	...	ID (N)th	ID (N)th NowCurH	ID (N)th NowCurL			
		1	1	LSM ID	1st	...	LSM ID (N-1)th	LSM ID (N)th					
Get LSMs LimitCurrent		8	3	ID 1st	ID 1st LimCurH	ID 1st LimCurL	...	ID (N)th	ID (N)th LimCurH	ID (N)th LimCurL			
		1	1	LSM ID	1st	...	LSM ID (N-1)th	LSM ID (N)th					
Get LSMs NowTemperature		9	2	ID 1st	ID 1st NowTemp	...	ID (N)th	ID (N)th NowTemp					
		1	1	LSM ID	1st	...	LSM ID (N-1)th	LSM ID (N)th					
Get LSMs LimtTemperature		10	2	ID 1st	ID 1st LimTemp	...	ID (N)th	ID (N)th LimTemp					
		1	1	LSM ID	1st	...	LSM ID (N-1)th	LSM ID (N)th					
Get LSMs GainP		11	2	ID 1st	ID 1st GainP	...	ID (N)th	ID (N)th GainP					
		1	1	LSM ID	1st	...	LSM ID (N-1)th	LSM ID (N)th					
Get LSMs GainI		12	2	ID 1st	ID 1st GainI	...	ID (N)th	ID (N)th GainI					
		1	1	LSM ID	1st	...	LSM ID (N-1)th	LSM ID (N)th					
Get LSMs GainD		13	2	ID 1st	ID 1st GainD	...	ID (N)th	ID (N)th GainD					
		1	1	LSM ID	1st	...	LSM ID (N-1)th	LSM ID (N)th					
Get LSMs ControlMargin		14	3	ID 1st	ID 1st CM H	ID 1st CM L	...	ID (N)th	ID (N)th CM H	ID (N)th CM L			
		1	1	LSM ID	1st	...	LSM ID (N-1)th	LSM ID (N)th					
Get LSMs FwdControl Value		15	3	ID 1st	ID 1st FCV H	ID 1st FCV L	...	ID (N)th	ID (N)th FCV H	ID (N)th FCV L			
		1	1	LSM ID	1st	...	LSM ID (N-1)th	LSM ID (N)th					
Get LSMs BoundaryMax		16	3	ID 1st	ID 1st BNDMX H	ID 1st BNDMX L	...	ID (N)th	ID (N)th BNDMX H	ID (N)th BNDMX L			
		1	1	LSM ID	1st	...	LSM ID (N-1)th	LSM ID (N)th					
Get LSMs BoundaryMin		17	3	ID 1st	ID 1st BNDMN H	ID 1st BNDMN L	...	ID (N)th	ID (N)th BNDMN H	ID (N)th BNDMN L			
		1	1	LSM ID	1st	...	LSM ID (N-1)th	LSM ID (N)th					
Get LSMs LED Value		18	4	ID 1st	ID 1st LED R	ID 1st LED G	ID 1st LED B	...	ID (N)th	ID (N)th LED R	ID (N)th LED G	ID (N)th LED B	
		1	1	LSM ID	1st	...	LSM ID (N-1)th	LSM ID (N)th					
Get LSMs ModelVersion		19	3	ID 1st	ID 1st MV H	ID 1st MV L	...	ID (N)th	ID (N)th MV H	ID (N)th MV L			
		1	1	LSM ID	1st	...	LSM ID (N-1)th	LSM ID (N)th					
Get LSMs FW Version		20	3	ID 1st	ID 1st FV H	ID 1st FV L	...	ID (N)th	ID (N)th FV H	ID (N)th FV L			
		1	1	LSM ID	1st	...	LSM ID (N-1)th	LSM ID (N)th					
Get LSMs ZeroCompValue		21	3	ID 1st	ID 1st ZCMP H	ID 1st ZCMP L	...	ID (N)th	ID (N)th ZCMP H	ID (N)th ZCMP L			
		1	1	LSM ID	1st	...	LSM ID (N-1)th	LSM ID (N)th					
Get Sync Info		22	2	임시데이터 인덱스	로스트번호								

– Data set Code의 상세 명령어

DATA SET	Type	Code	Para Length	Para1	Para2	Para3	Para4	Para5	Para6	Para7	Para8	Para9	...
Set LSC Version	0x20	1	3	HW Version	FW Major	FW Minor							
Set LSC Address		2	2	Address H	Address L								
Set LSM Module ID		3	2	LSM ID	new LSM ID								
Set Sync Target Position		4	4	ID 1st SyncT/S	ID 1st SyncPosH	ID 1st SyncPosL	...	ID (N)th	ID (N)th SyncT/S	ID (N)th SyncPosH	ID (N)th SyncPosL		
Set Sync LED Control Value		5	4	ID 1st LED R	ID 1st LED G	ID 1st LED B	...	ID (N)th	ID (N)th LED R	ID (N)th LED G	ID (N)th LED B		
Set LSM Limit Current		6	3	ID 1st	ID 1st LimCurH	ID 1st LimCurL	...	ID (N)th	ID (N)th LimCurH	ID (N)th LimCurL			
Set LSM Limit Temperature		7	2	ID 1st	ID 1st LimTemp	...	ID (N)th	ID (N)th LimTemp					
Set LSM GainP		8	2	ID 1st	ID 1st GainP	...	ID (N)th	ID (N)th GainP					
Set LSM GainI		9	2	ID 1st	ID 1st GainI	...	ID (N)th	ID (N)th GainI					
Set LSM GainD		10	2	ID 1st	ID 1st GainD	...	ID (N)th	ID (N)th GainD					
Set LSM ForwardControl Margin		11	3	ID 1st	ID 1st CM H	ID 1st CM L	...	ID (N)th	ID (N)th CM H	ID (N)th CM L			
Set LSM Boundary Max		12	3	ID 1st	ID 1st BNDMX H	ID 1st BNDMX L	...	ID (N)th	ID (N)th BNDMX H	ID (N)th BNDMX L			
Set LSM Boundary Min		13	3	ID 1st	ID 1st BNDMN H	ID 1st BNDMN L	...	ID (N)th	ID (N)th BNDMN H	ID (N)th BNDMN L			
Set LSM Comm Baudrate		14	4	ID 1st	ID 1st BPS H	ID 1st BPS M	ID 1st BPS L	...	ID (N)th	ID (N)th BPS H	ID (N)th BPS M	ID (N)th BPS L	
Set LSM ZeroComp Value		15	3	ID 1st	ID 1st ZCMP H	ID 1st ZCMP L	...	ID (N)th	ID (N)th ZCMP H	ID (N)th ZCMP L			
Set LSM Factory Reset		16	4	LSM ID	0x53	0x4E	0x52						
Set LSS Module ID		17	3	Product ID	LSS ID	new LSS ID							

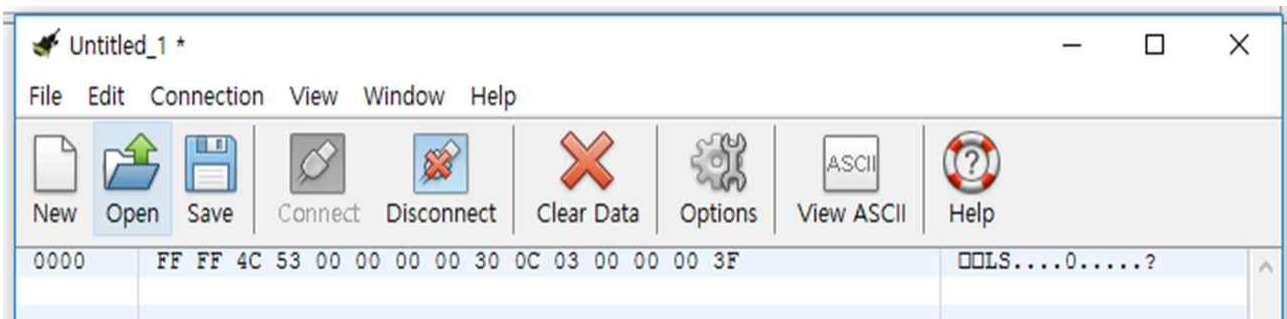
– Data Execution Code의 상세 명령어

EXECUTION	Type	Code	Para Length	Para1	Para2	Para3	Para4	Para5	Para6	Para7	Para8	Para9	...
Exe LSM PWR ON/OFF	0x30	1	1	0:OFF / 1:ON									
Exe LSM Configure Save		2	[N]LSMs	LSM ID: 1st	---	LSM ID: 2ndth							
Exe LSM PosMove Torq		3	[N]LSMs x 4	LSM ID: 1st	ID: 1st Torque	ID: 1st Pos	ID: 1st Pos	---	LSM ID: 2ndth	ID: 2ndth Torque	ID: 2ndth Pos	ID: 2ndth Pos	
Exe LSM PosMove Speed		4	[N]LSMs x 4	LSM ID: 1st	ID: 1st Speed	ID: 1st Pos	ID: 1st Pos	---	LSM ID: 2ndth	ID: 2ndth Speed	ID: 2ndth Pos	ID: 2ndth Pos	
Exe LSM LED Control		5	[N]LSMs x 4	LSM ID: 1st	ID: 1st LED R	ID: 1st LED G	ID: 1st LED B	---	LSM ID: 2ndth	ID: 2ndth LED R	ID: 2ndth LED G	ID: 2ndth LED B	
Exe LSM Sync PosMove		6	[N]LSMs x 2	LSM ID: 1st	ID: 1st SyncMode	---	LSM ID: 2ndth	ID: 2ndth SyncMode					
Exe LSM Sync LED Control		7	[N]LSMs	LSM ID: 1st	---	LSM ID: 2ndth							
Exe LSM Integration Control		8	[N]LSMs x 7	LSM ID: 1st	ID: 1st Torque	ID: 1st Pos	ID: 1st Pos	ID: 1st LED R	ID: 1st LED G	ID: 1st LED B	---		
Exe LSM Wheel Mode		9	[N]LSMs x 3	LSM ID: 1st	ID: 1st WheelMode	ID: 1st WheelMode	---	LSM ID: 2ndth	ID: 2ndth WheelMode	ID: 2ndth WheelMode			
Exe LSM ActiveStop		10	[N]LSMs	LSM ID: 1st	---	LSM ID: 2ndth							
Exe LSM PassiveStop		11	[N]LSMs	LSM ID: 1st	---	LSM ID: 2ndth							
Exe Motion		12	3	MotionNumber	Start delay [ms]	Speed: 50~500%							
Exe Process		13	1	0:OFF / 1:ON									
Exe Remocon		14	4	mode	0:OFF / 1:ON	L: data	R: data						
Exe Motion info		15	2	MotionNumber	PosNumber								

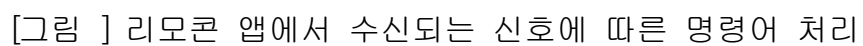
- 사용 예시

Exe Motion	0xFF	0x4C	0x53	0x00	0x00	0x00	0x00	0x30	0x0C	0x03	MotionNumber	0x00	0x00	Checksum	
LED control	: ID[3] LSM Red LED On														
	0xFF	0xFF	0x4C	0x53	0x00	0x00	0x00	0x30	0x05	0x04	0x03	0x05	0x00	0x00	Checksum
Position control	: ID[2] LSM Position '-100' with torque 80%														
	0xFF	0xFF	0x4C	0x53	0x00	0x00	0x00	0x30	0x03	0x04	0x02	0x50	0xFF	0x9C	Checksum
											id 2	torq 80	position -100		

- 컨트롤러에 저장된 3번 모션을 실행 하는 명령어 완성본



[그림] 통신 내용을 캡처



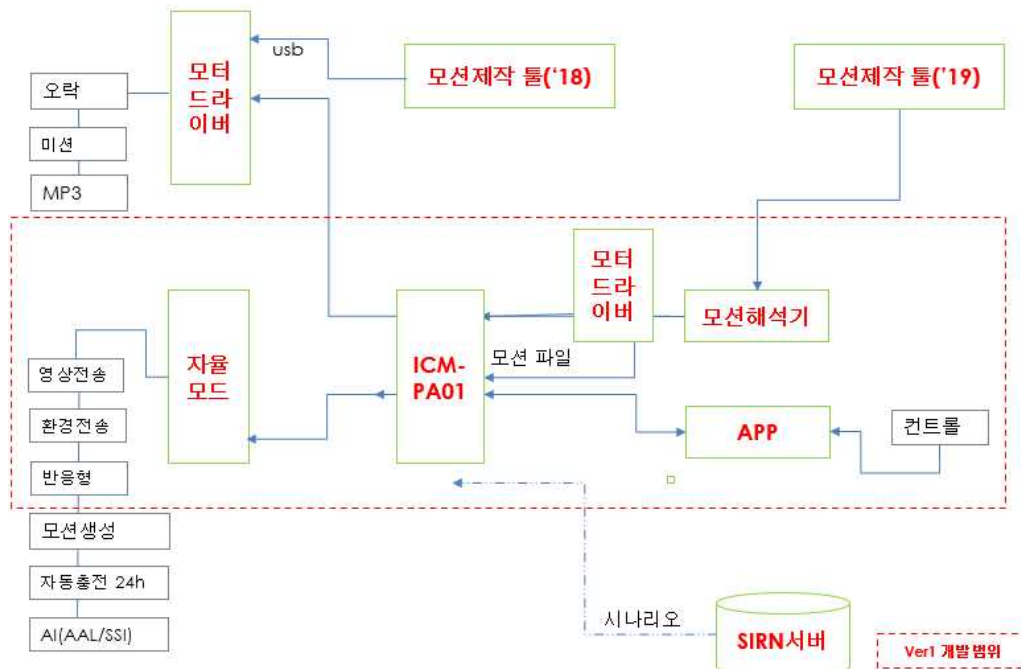
- | Management | Type | Code | Para Length | Para1 | Para2 | Para3 | Para4 | Para5 | Para6 | Para7 | Para8 | Para9 |
|------------|------|------|-------------|-----------|-----------|------------|------------|-----------|-----------|-------|-------|-------|
| PROCESS | | 1 | 6 | PlatformH | PlatformL | FileSize 1 | FileSize 2 | FileSize3 | FileSize4 | | | |

[illegible]

1.	HOST >> LSC (Command)																Checksum		
	Header				Dest Addr		Source Addr		Statement		ParaLength	Para						Checksum	
	0xFF	0xFF	0x4C	0x53	LSC AddrH	LSC AddrL	Host AddrH	Host AddrL	0x40	1	6	PlatformH	PlatformL	FileSize 1 (MSB)	FileSize 2	FileSize3	FileSize4 (LSB)	Checksum	
2.	LSC >> HOST (Response)																Checksum		
	Header				Source Addr		Dest Addr		Statement		ParaLength	Para						Checksum	
	0xFF	0xFF	0x4C	0x53	Host AddrH	Host AddrL	LSC AddrH	LSC AddrL	0x40	1	6	STATUS 1	STATUS 2	FileSize 1 (MSB)	FileSize 2	FileSize3	FileSize4 (LSB)	Checksum	
3.	HOST >> LSC (File Transfer : Max 64 byte)																		
	Filedata[1]	...	Filedata[64]																
4.	LSC >> HOST (Response : Received byte CountNbr)																		
	64 (0x40)																		
5.	HOST >> LSC (File Transfer : Max 64 byte)																		
	Filedata[65]	...	Filedata[129]																
6.	LSC >> HOST (Response : Received byte CountNbr)																		
	64 (0x40)																		
7.	HOST >> LSC (File Transfer : Max 64 byte)																		
	Filedata[130]	...	Filedata[140]																
8.	LSC >> HOST (Response : Received byte CountNbr)																		
	10 (0x0A)																		

[그림] 모션 데이터 파일 전송 규약의 프로세스

4. 종합 서비스 설계



[그림] 콘텐츠 제작을 위한 종합 서비스 개념도

- 로봇의 모션 테이블

기존에 판매 중인 휴머노이드 로봇은 컨트롤러에 기본적으로 저장된 모션 테이블을 갖고 있다.

앞서 살펴본 프로토콜에 의해서 전송된 모션 번호에 따라 로봇은 정해진 모션을 순차적으로 실행 하도록 되어 있다.

Motion Table						
MRT R&D Center				(ver. 2019.04.30)		
Motion	Mode		Motion		Voice Command	Remark
Index No.	No	Name	Des1	Des2		
1	M0	Basic	Ready		Ready	
2			F	ST	Go Forward	
3				Loop		
4				End		
5			Left		Go Left	
6			Right		Go Right	
7			Turn	L	Turn Left	
8			Turn	R	Turn Right	
9			B	ST	Go Backward	
10				End		
11				Loop		
12			L	Forward	Left Forward	
13			R	Forward	Right Forward	
14			Getup	F	Getup forward	
15			Getup	B	Getup backward	
16			Losw1		lose	
17			Win1		champion	
18			Hi		Hi	
19			Bow1		bow	
20			Tumble	F	Tumbling forward	
21			Tumble	B	Tumbling backward	

[그림] 로봇 제조사의 컨트롤러에 기본 내장된 모션 21개

– ICM-PA01의 기본 예제 제작

아래 코드는 PIXY2 카메라에서 지정한 Sig1번 색상을 가진 물체를 감지하였을 때 아두이노에서 어떻게 반응하는가를 알아 보는 기본적인 소스 코드이다.

```
#include <Pixy2.h>

// This is the main Pixy object
Pixy2 pixy;

void setup()
{
  Serial.begin(115200);
  Serial.print("Starting...\n");

  pixy.init();
}

void loop()
{
  int i;
  // grab blocks!
  pixy.ccc.getBlocks();

  // If there are detect blocks, print them!
  if (pixy.ccc.numBlocks)
  {
    Serial.print("Detected ");
    Serial.println(pixy.ccc.numBlocks);
    for (i=0; i<1; i++) // 여러개의 Block 정보 가운데 1번째에 해당하는 내용만 사용함.
    {
      Serial.print("  block ");
      Serial.print(i);
      Serial.print(": ");
      pixy.ccc.blocks[i].print();
    }
    pixy.setLamp(255,0); // 어떠한 물체라도 찾아 냈을 경우 위쪽 좌우의 2개의 램프를 점등
  }
  else
  {
    pixy.setLamp(0,0); // 위쪽 좌우의 2개의 램프를 소등
  }
  delay(100); // 수시로 점등/소등이 되지 않도록 하기 위해서 약간의 딜레이를 준다.
}
```

- ICM-PA01의 방향 감지와 방향 전환용 코드

```
#include <Pixy2.h>
Pixy2 pixy;
void setup()
{
    Serial.begin(115200);
    Serial.print("Starting...\n");
    pixy.init();
}
void loop()
{
    pixy.ccc.getBlocks();
    if (pixy.ccc.numBlocks)
    {
        Serial.print("Detected ");
        Serial.println(pixy.ccc.numBlocks);
        for (int i=0; i<1; i++) // 여러개의 Block 정보 가운데 1번째에 해당하는 내용만 사용함.
        {
            Serial.print("  block ");
            Serial.print(i);
            Serial.print(": ");
            pixy.ccc.blocks[i].print();
        }
        pixy.setLamp(255,0); // 어떠한 물체라도 찾아 냈을 경우 위쪽 좌우의 2개의 램프를 점등
        // 카메라의 전체 화소는 320 x 240 임
        // 카메라의 중심은 160임
        // 물체의 중심이 100 보다 작으면 왼쪽
        // 물체의 중심이 100 ~ 220에 있으면 중앙에 있다고 봄
        // 물체의 중심이 220보다 크면 오른쪽에 위치 하고 있다고 봄
        if (pixy.ccc.blocks[0].m_x < 100) // 감지 물체의 중심이 카메라의 중앙 보다 왼쪽에 위치
        {
            // 왼쪽으로 턴
        }
        else if (pixy.ccc.blocks[0].m_x > 220) // 감지 물체의 중심이 카메라의 중앙 보다 오른쪽에 위치
        {
            // 오른쪽으로 턴
        }
        else
        {
            // 앞으로 전진
        }
    }
    else
    {
        pixy.setLamp(0,0); // 위쪽 좌우의 2개의 램프를 소등
    }
    delay(100); // 수시로 점등/소등이 되지 않도록 하기 위해서 약간의 딜레이를 준다.
}
```


- ICM-PA01의 로봇 제어 전체 코드

```
#include <Pixy2.h>
Pixy2 pixy;
// 로봇 모션을 정의한 테이블 번호와 동일함.
enum {
    STOP          = 0,
    M0_Ready       = 1,
    M0_F_ST        = 2,
    M0_F_Loop       = 3,
    M0_F_End       = 4,
    M0_Left        = 5,
    M0_Right       = 6,
    M0_Trun_L      = 7,
    M0_Turn_R      = 8,
    M0_B_ST        = 9,
    M0_B_End       = 10,
    M0_B_Loop      = 11,
    M0_L_Forward   = 12,
    M0_R_Forward   = 13,
    M0_Getup_F     = 14,
    M0_Getup_B     = 15,
    M0_Losw1       = 16,
    M0_Win1        = 17,
    M0_Hi          = 18,
    M0_Bow1        = 19,
    M0_Tumble_F    = 20,
    M0_Tumble_B    = 21,
};

void initialize(void)
{
    //Serial.begin(115200);
    //LED pin setting
    pinMode(3, OUTPUT);
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);
    pinMode(7, OUTPUT);

    //RGB LED initialize
    digitalWrite(3, HIGH);    //Blue(Active low)
    digitalWrite(5, HIGH);    //Green(Active low)
    digitalWrite(6, HIGH);    //Red(Active low)

    //LED initialize
    digitalWrite(7, HIGH);    //Active high

    delay(1000);              //Waiting for init Line Core M
}

void setup()
{
    Serial.begin(115200);
    Serial.print("Starting...\n");
    initialize(); // ICM-PA01 보드의 램프를 사용하기 위하여 초기화 함.
    pixy.init();
}
```

```

// 로봇 제어와 ICM-PA01의 RGB LED를 제어 하기 위한 함수
void sendData(unsigned char data, unsigned int r, unsigned int g, unsigned int b)
{
    if (data >= M0_Ready && data <= M6_Getup_B) Serial.write(data);
    analogWrite(3, 255-b);      //Blue(Active low)
    analogWrite(5, 255-g);      //Green(Active low)
    analogWrite(6, 255-r);      //Red(Active low)
}
void loop()
{
    int i;
    pixy.ccc.getBlocks();
    if (pixy.ccc.numBlocks)
    {
        for (i=0; i<1; i++)
        {
            // Block's x is left
            if(pixy.ccc.blocks[i].m_x < 80){
                if (beforeCenter){
                    beforeCenter = 0;
                    sendData(M0_F_End); delay(500);
                }
                sendData(M0_Trun_L);
                atFirst = 1;
            }
            // Block's x is right
            else if(236 < pixy.ccc.blocks[i].m_x){
                if (beforeCenter){
                    beforeCenter = 0;
                    sendData(M0_F_End); delay(500);
                }
                sendData(M0_Turn_R);
                atFirst = 1;
            }
            // Block's x is centre
            else if(80 <= pixy.ccc.blocks[i].m_x <= 236){
                beforeCenter = 1;
                if (atFirst){
                    sendData(M0_F_ST); delay(300);
                    atFirst = 0;
                }
                sendData(M0_F_Loop); delay(300);
            }
        }
    }
}

```

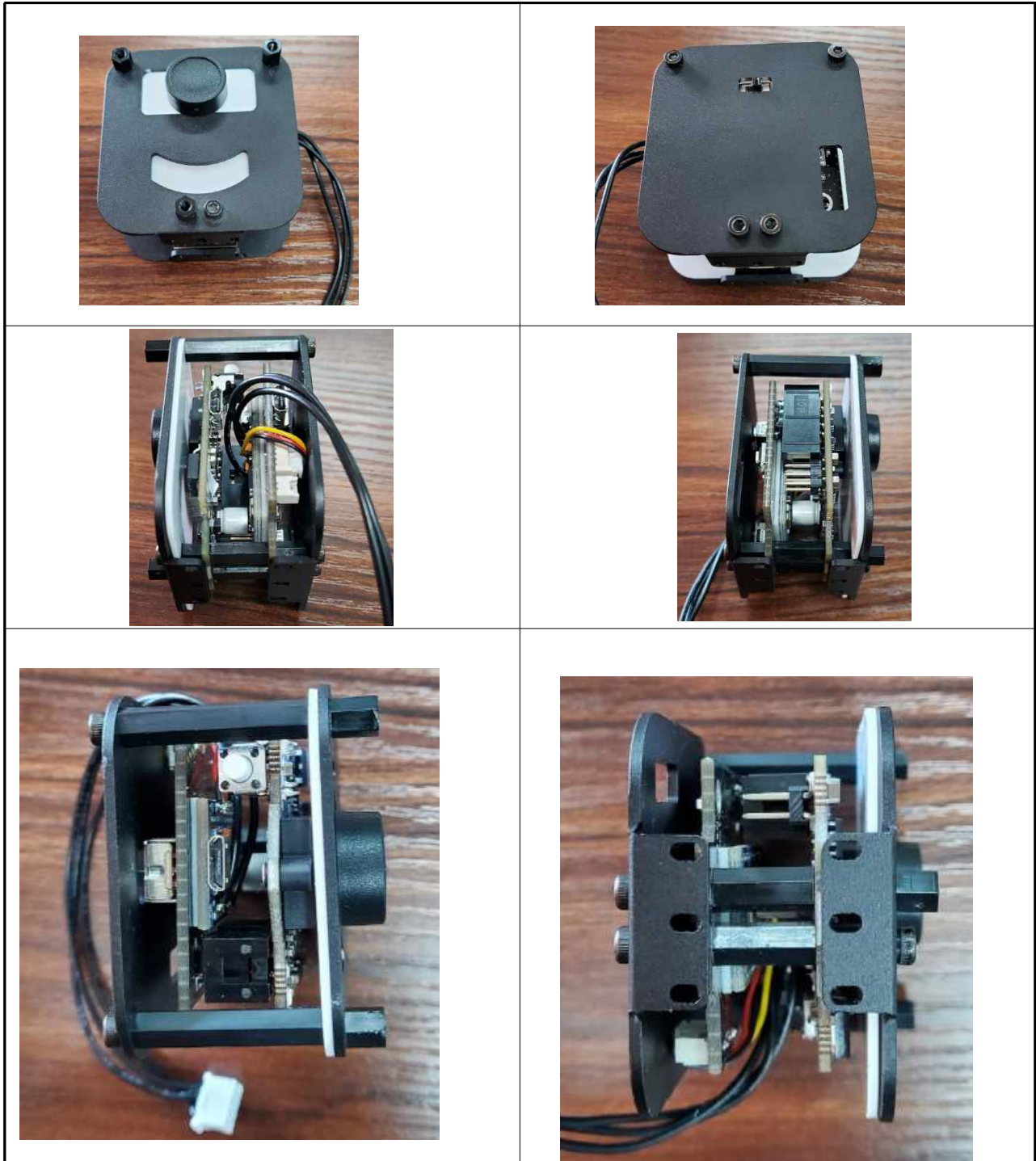
마. 소결

학습한 Protocol을 교육하고 만들어진 ICM-PA01모듈을 이용하여 실제 로봇을 움직이도록 하는 체험을 통해 참여 학생들을 대상으로 과제의 결과물을 활용한 다양한 실험을 할 수 있도록 하여 콘텐츠 개발과 대회 참가등의 성과를 낼 수 있도록 함.

3. 과제 결과물

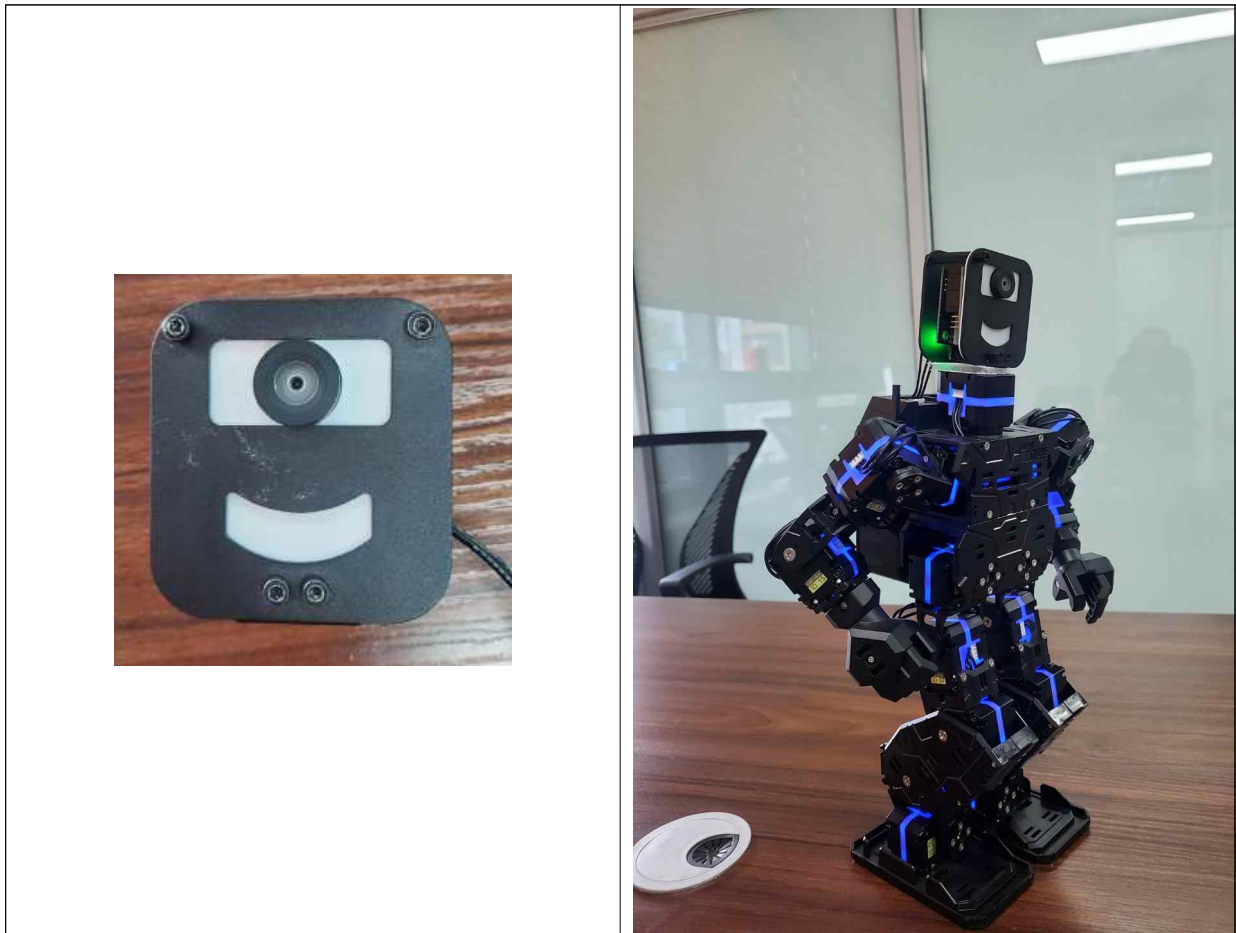
가. 시제품

- 제작 기획을 하면서 기존의 휴머노이드 로봇과 바퀴 굴림형 로봇등에서 활용 할 수 있는 독립적인 컨트롤러기능을 하는 인공지능 시각 처리 모듈을 제작함.



[그림] ICM-PA01 최종 결과물 상세

- 결과물은 제작의도와 도면에 맞게 제작 되었으며 실제 휴머노이드 로봇에 장착하여 테스트를 하였음.



[그림] 왼쪽 최종 결과물과 휴머노이드와 결합 사용 예

2) 제품의 용도와 특성

◦ 용도 :

- 단순 리모콘형 이거나 센서에 의존하는 휴머노이드 로봇 또는 바퀴 굴림 로봇에 AI기반의 시각 인공지능을 탑재한 디바이스 ICM을 만들어 추가함으로서 자율형 서비스 로봇으로 활용 할 수 있는 콘텐츠 제작이 가능.
- 아두이노 코딩 실습이 가능한 Arduino Nano를 채용하여 교육용으로 활용 할 수 있음.
- 국제 로봇 3종대회(달리기, 장애물 회피, 마라톤)와 같은 로봇 챌린지에 출전 할 수 있는 샘플 코드를 제공하여 참가의 기회를 제공.

◦ 특성 :

- 2개의 PCB를 연결하는 브라켓 구조물을 추가 하여 완성도를 높였음.
- 로봇의 머리 부분과 ICM-PA01의 접결은 4pin 커넥터로 처리하여 완성도 높임

