



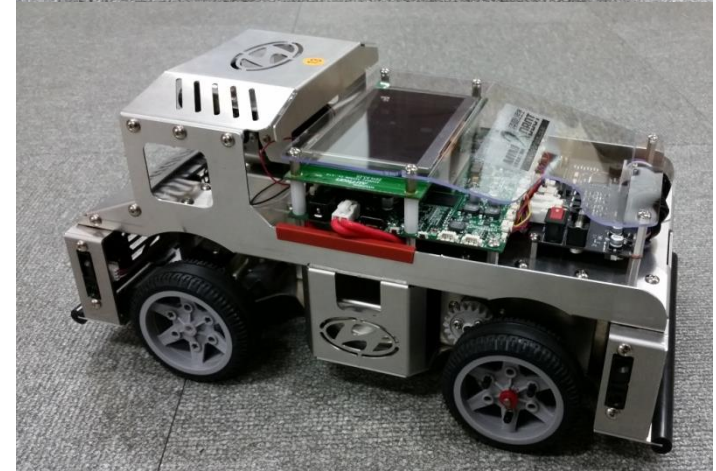
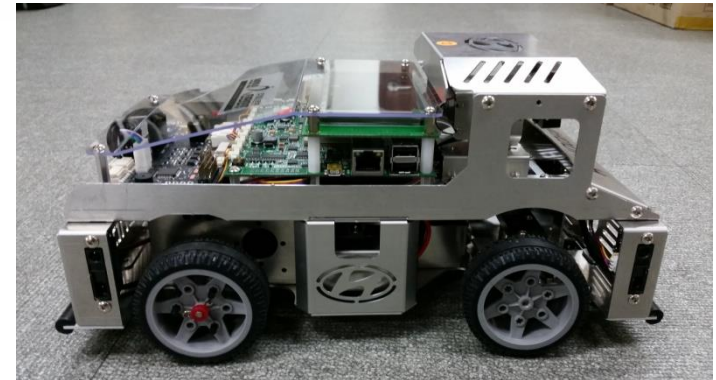
[주]미니로봇

2017년 무인자동차



1. 2017지능형 자동차

- 임베디드보드 제작처: 현대 오토론
- 자동차 바디 및 제어보드 제작처: (주)미니로봇





항목	규격 및 내용	비고
크기	330 * 190 * 160 mm	
무게	약 2300g	
최고속도	1000 mm/sec	
구동 모터	DC서보 모터 엔코더 (12V-12W)	
임베디드 보드	NVIDIA Tegra3 (ARM Cortex-A9 Quadcore 900MHz)	DDR3 2GB, NOR Flash 64MB, eMMC 8GB
카메라	CMOS 1.3M Pixel (1280 x 1024)	
디버깅	Ethernet (10/100) 및 mini USB	
LCD	4.3" (해상도 480x272)	
바퀴분해능	195step/1회전	
서보 모터 - 3EA (바퀴 조향 및 카메라 틸트)	6V. 9kg(180도)	카메라 상하좌우, 조향
회전 반지름	약 300mm	
거리 분해 능	1mm	거리 분해능
거리센서	적외선 거리센서 - 6SET (4cm~30cm)	GP2Y0A41SK0F
적외선센서	라인감지용 7조	오토 튜닝
바퀴지름	65mm	
충전기	ULTRAMAT AC/DC EQ	
전압 / 배터리	리튬이온 4CELL 14.8V - 3500mA	방전율 5A
통신#1	RS232C: 19200BPS(고정)	
충전 및 동작 시간	약180분 충전, 약 2시간30분사용	
전조등, 방향등, 정지등		

2017지능형 자동차



소프트웨어 개발 환경

임베디드 프로그램 개발환경: Linux OS
NVIDIA Tegra3
(ARM Cortex-A9 Quadcore 900MHz)

자동차 제어 보드:
UART 통신 제어 (19200bps)



↔
SPI 또는
UART 통신



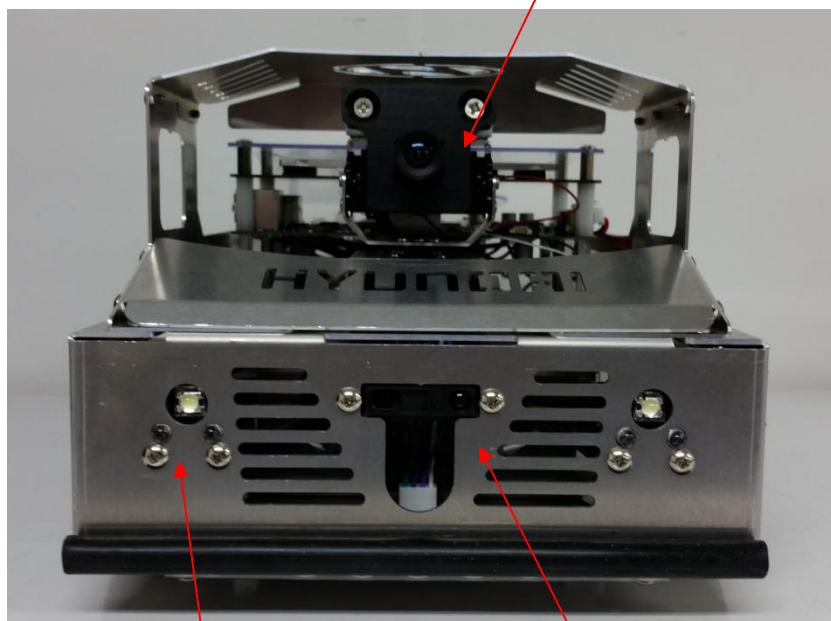
2017지능형 자동차

앞 뒤면 부분 명칭



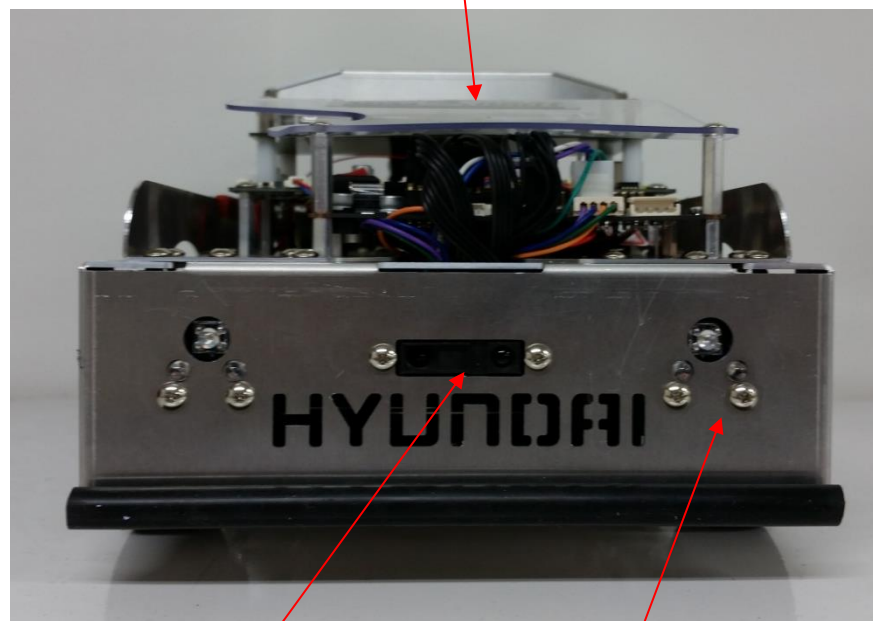
카메라

480x272 TFT LCD



- 전조등
- 좌우 측 감박이

전방 거리 센서



후방 거리 센서

- 좌우 측 감박이
- 정지등

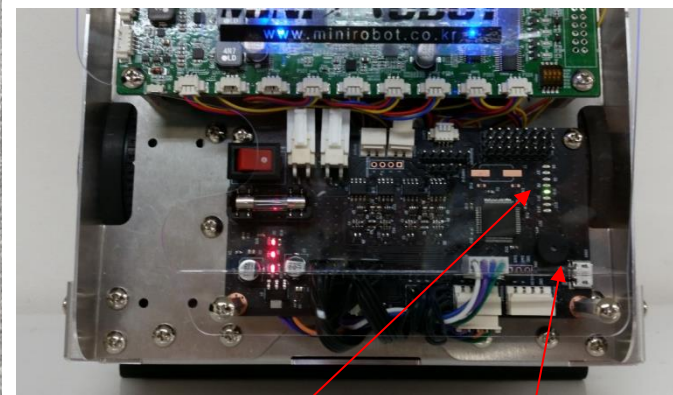
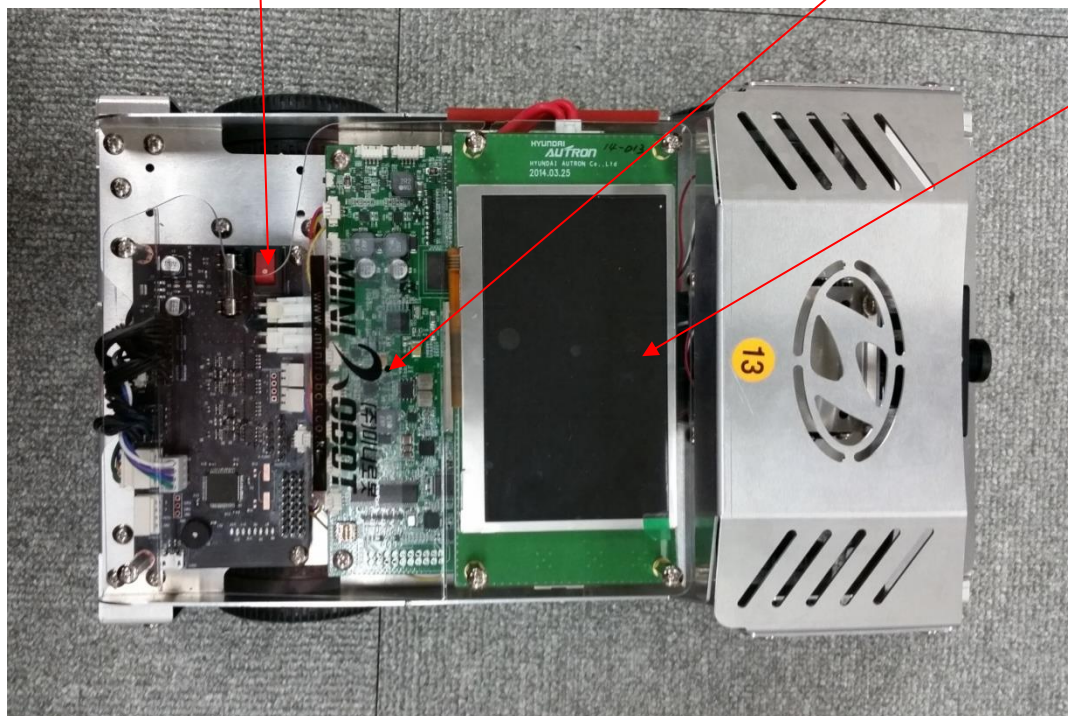
2017지능형 자동차

평면 부분 명칭

자동차
전원스위치

임베디드 보드
NVIDIA Tegra3
(ARM Cortex-A9 Quadcore 900MHz)

480x272 TFT LCD



배터리 체커

부저



2017지능형 자동차

측면 부분 명칭



배터리 충전단자

임베디드 개발 포트

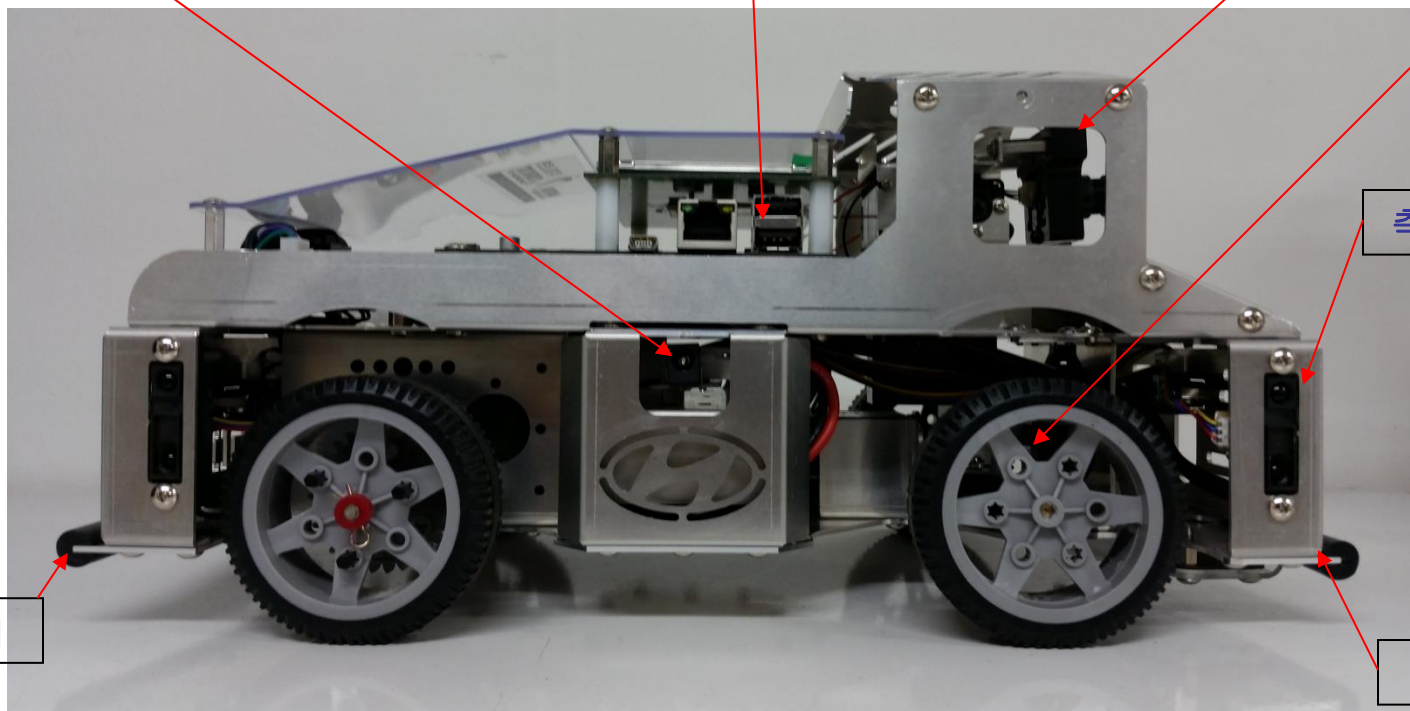
카메라 상하좌우 제어

조향 제어

측면 거리 감지

실리콘 범퍼

실리콘 범퍼



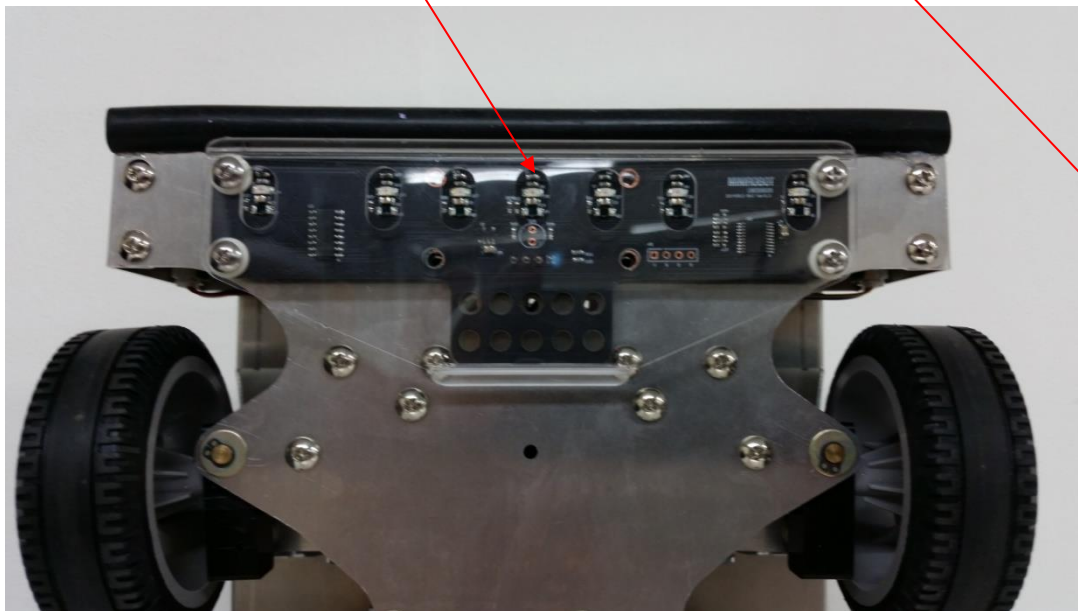
2017지능형 자동차

하부 면 부분 명칭



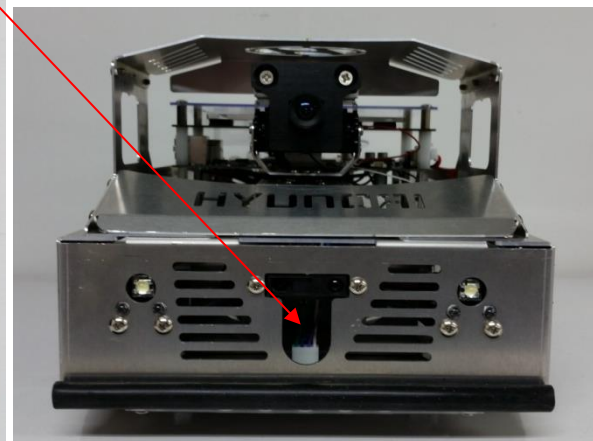
하부 적외선 라인감지센서 7조

라인감지센서 Auto 튜닝 버튼



오토튜닝 방법:

- 1.자동차를 감지할 라인 위에 놓는다.
- 2.전면 부 오토튜닝 버튼을 한번 누른다.
- 3.자동차를 라인 위에서 앞뒤로 5초 정도 굴린다.
- 4.시간이 지나면 라인 감지가 오토 튜닝 된다.



2017지능형 자동차



충전기: 리튬이온 충전지 4CELL

사용방법: 컨넥터 배터리 연결 시 충전 중 빨간색, 완충 녹색



제어 코드표



명령코드	의미	읽기/쓰기	Byte 개수	BIT 형식	제어 상태	최대,최소값	
0x90	SpeedControlOnOff	R/W	1	부호없는 8bit Byte	1 = 제어, 0 = 단순 PWM	0~1	
0x91	DesireSpeed	R/W	2	부호있는 16bit Short	(encoder tic 값) * 10 / s	(-500)~500	읽혀지는값은 현재 속도, 거리제어중에는 거리가 남아있어야 동작.
0x92	SpeedPID Proportional	R/W	1	부호없는 8bit Byte	speed 비례 제어값	1~50	기본값 10
0x93	SpeedPID Integral	R/W	1	부호없는 8bit Byte	speed 적분 제어값	1~50	기본값 10
0x94	SpeedPID Differential	R/W	1	부호없는 8bit Byte	speed 미분 제어값	1~50	기본값 10
0x96	PositionControlOnOff	R/W	1	부호없는 8bit Byte	1 = 제어, 0 = 거리 제어없이 속도 제어	0~1	속도 값이 있어야 제어가 가능합니다.
0x97	Desire EncoderCount	R/W	4	부호있는 32bit Long	encoder tic	(-2147483648)~2147483647	값을 써넣으면 그 값 만큼 이동, 읽혀지는값은 남아있는 거리 값
0x98	PositionProportionPoint	R/W	1	부호없는 8bit Byte	EncoderCounter 비례 제어값	1~50	
0xA0	FrontARearLightControl	R/W	1	부호없는 8bit Byte	0 = AllOff, 1 = 전조등On, 2 = 후미등On, 3 = AllOn	0~3	
0xA1	RightALeftFlickerControl	R/W	1	부호없는 8bit Byte	0 = AllOff, 1 = 우측On, 2 = 좌측On, 3 = AllOn	0~3	
0xA2	sound	R/W	1	부호없는 8bit Byte	(부저음길이값) * 0.01s	0~255	
0xA3	SteeringServoControl	R/W	2	부호있는 16bit Short	(각도제어값) 1500값 = 0도, 1값 = 0.1도	1000~2000	
0xA5	CameraXServoControl	R/W	2	부호있는 16bit Short	각도제어 1500값 = 0도, 1값 = 0.1도	600~2400	
0xA7	CameraYServoControl	R/W	2	부호있는 16bit Short	각도제어 1500값 = 0도, 1값 = 0.1도	1200~1800	
0xB0	EncoderCounter	R/W	4	부호있는 32bit long	encoder tic	(-2147483648)~2147483647	현재 기록중인 엔코더값을 읽어옵니다. 값을 써넣는것으로 초기화
0xB1	LineSensor	R	1	부호없는 8bit Byte	1~7 bit	0~255	bit 단위로 검은색 1 흰색 0 입니다. * 라인센서는 센서의 버튼을 눌른 후 모든 센서가 흰색과 검은색을 지나가게 하여 초기화를 해야 동작을 시작합니다.