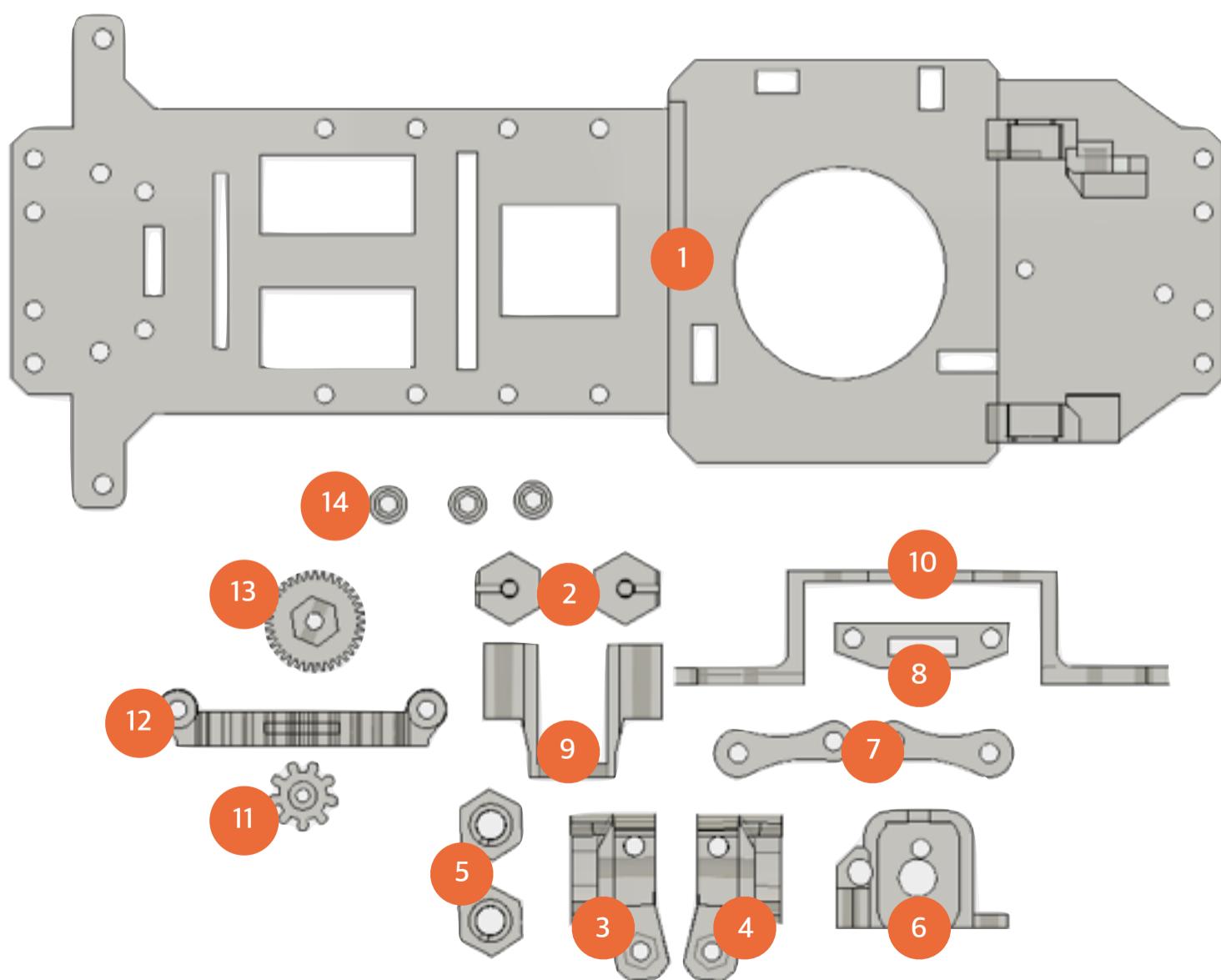


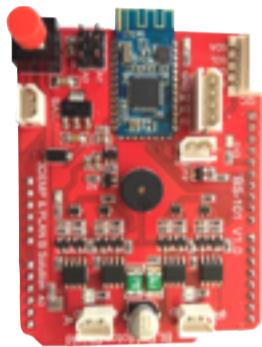
EMP-C201 PARTS LIST (2WD)

3D PRINTING PARTS				
	NO	NAME	QTY	PRINTER STEEING
Main Frame	1	mainFrame.stl	1	Layer Height : 0.3mm Fill Density : 100%
Rc Parts	2	FrontWheel_Hub.stl	2	
	3	L_Steering.stl	1	
	4	R_Steering.stl	1	
	5	RearWheel_Hub.stl	2	
	6	GearBox.stl	1	Layer Height : 0.3mm Fill Density : 100%
	7	Bredge.stl	2	
	8	ServoGuide_A.stl	1	
	9	ServoGuide_B.stl	1	
	10	SteeringBar.stl	1	
	11	ServoGear.stl	1	
Gear Parts	12	SteeringGear.stl	1	Layer Height : 0.2mm Fill Density : 100%
	13	LockGear.stl	1	Platform adhesion Type : Raft
Shaft Guide	14	ShaftGuide.stl	3	Layer Height : 0.1mm Fill Density : 100%



HARDWARE PARTS

NO	NAME	QTY	NO	NAME	QTY
1	EMP MOTOR SHIELD	1	10	BOLT (M3 X 40mm)	2
2	ARDUINO UNO	1	11	BOLT (M3 X 30mm)	4
3	SERVO MOTOR	1	12	BOLT (M3 X 10mm)	16
4	DC MOTOR	1	13	LOCKNUT (M3)	17
5	BATTERY HOLDER	1	14	SHAFT LOCKNUT	1
6	BATTERY (18650 3.7V)	2	15	GEAR SHAFT (M3 X 20mm)	1
7	1:10 65mm Tire	4	16	WHEEL SHAFT (M3 X 100mm)	1
8	PINION GEAR	1	17	BEARING (688zz)	2
9	GEAR 12-36	2	18	BEARING (MR115)	2



1 EMP MOTOR SHIELD



2 ARDUINO UNO (Clone)



3 SERVO MOTOR



4 DC MOTOR



5 BATTERY HOLDER



6 BATTERY (18650)



7 1:10 65mm Tire



16 WHEEL SHAFT

8 PINION GEAR



9 GEAR 12-36

15 GEAR SHAFT

12 BOLT
10mm x 16



18 BEARING(MR115) x 2

17 BEARING(688zz) x 2

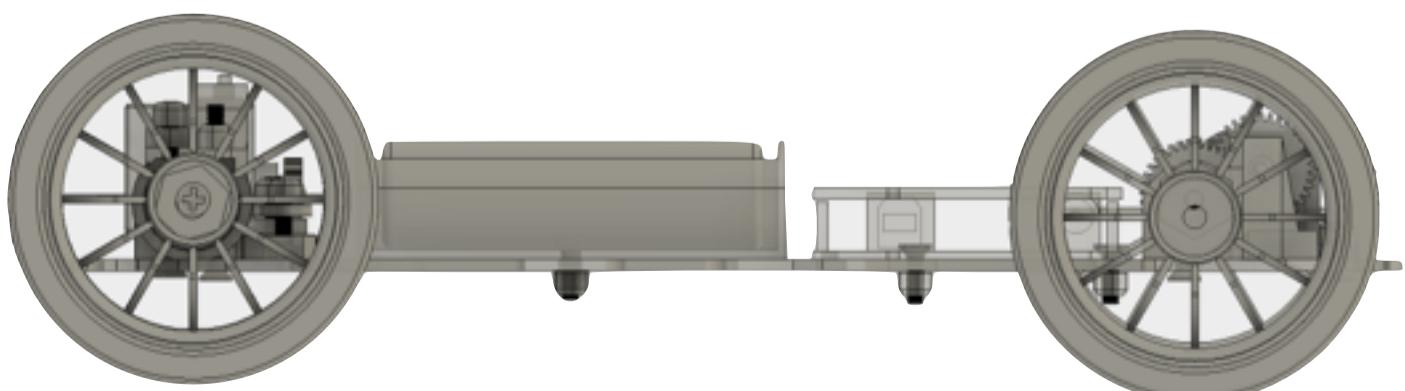
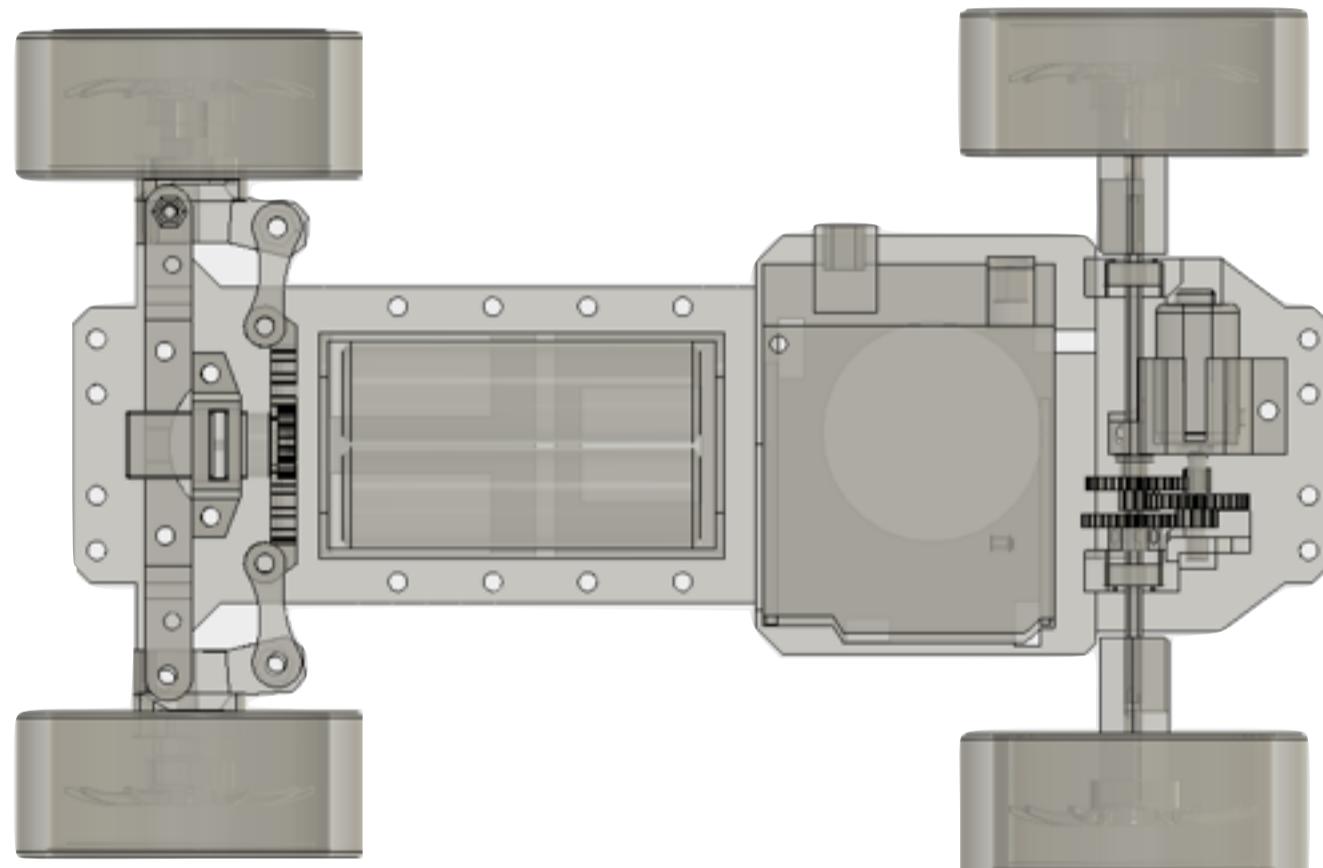
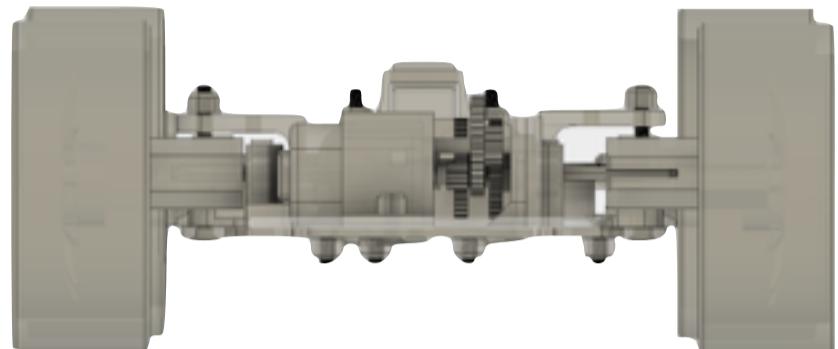
13 LOCKNUT
x 17

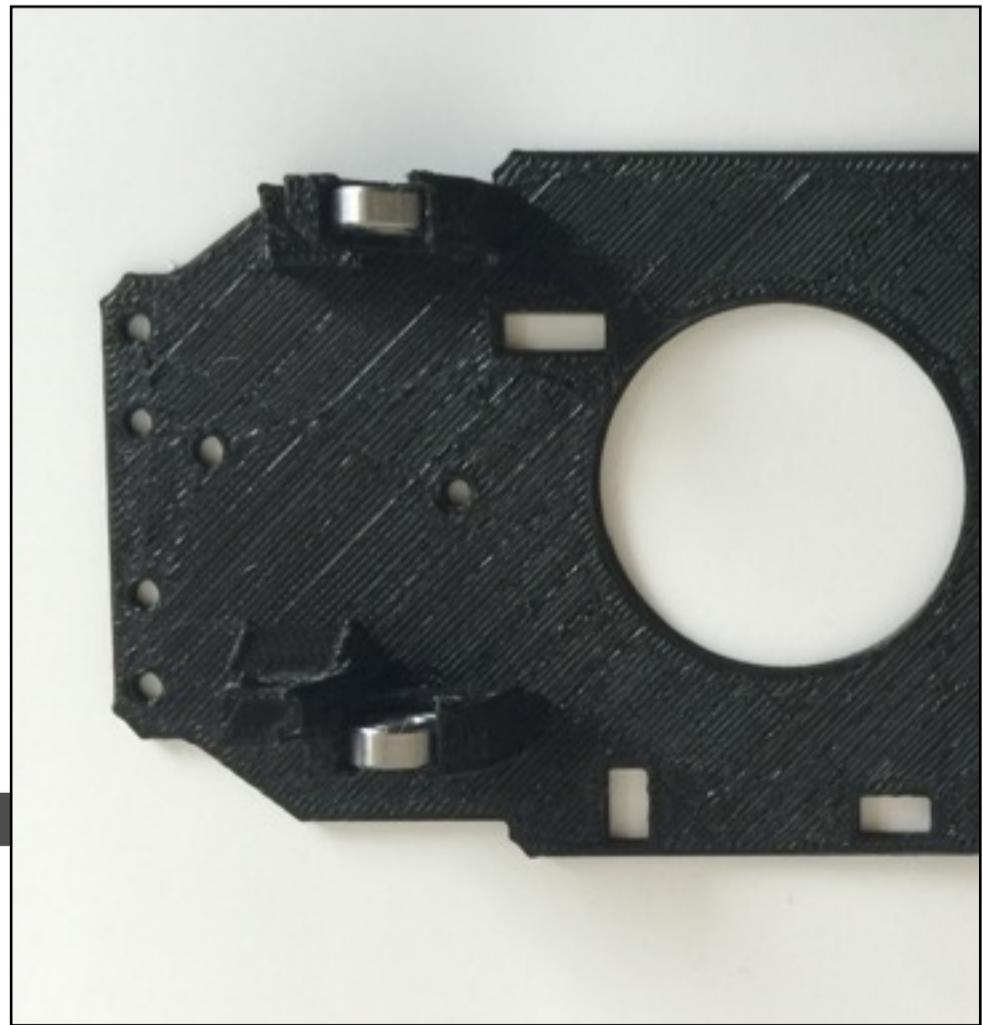
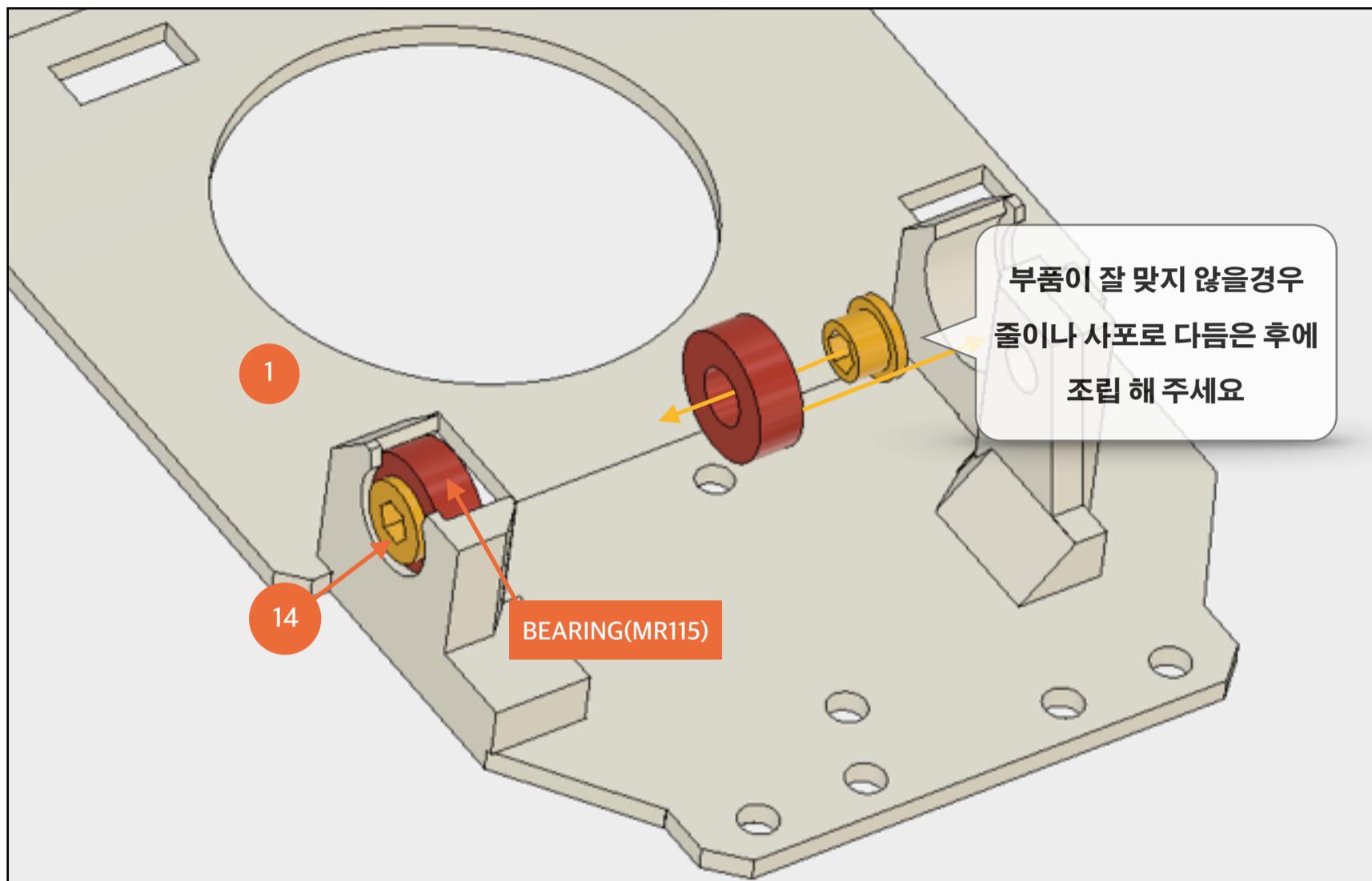
14 SHAFT
LOCKNUT

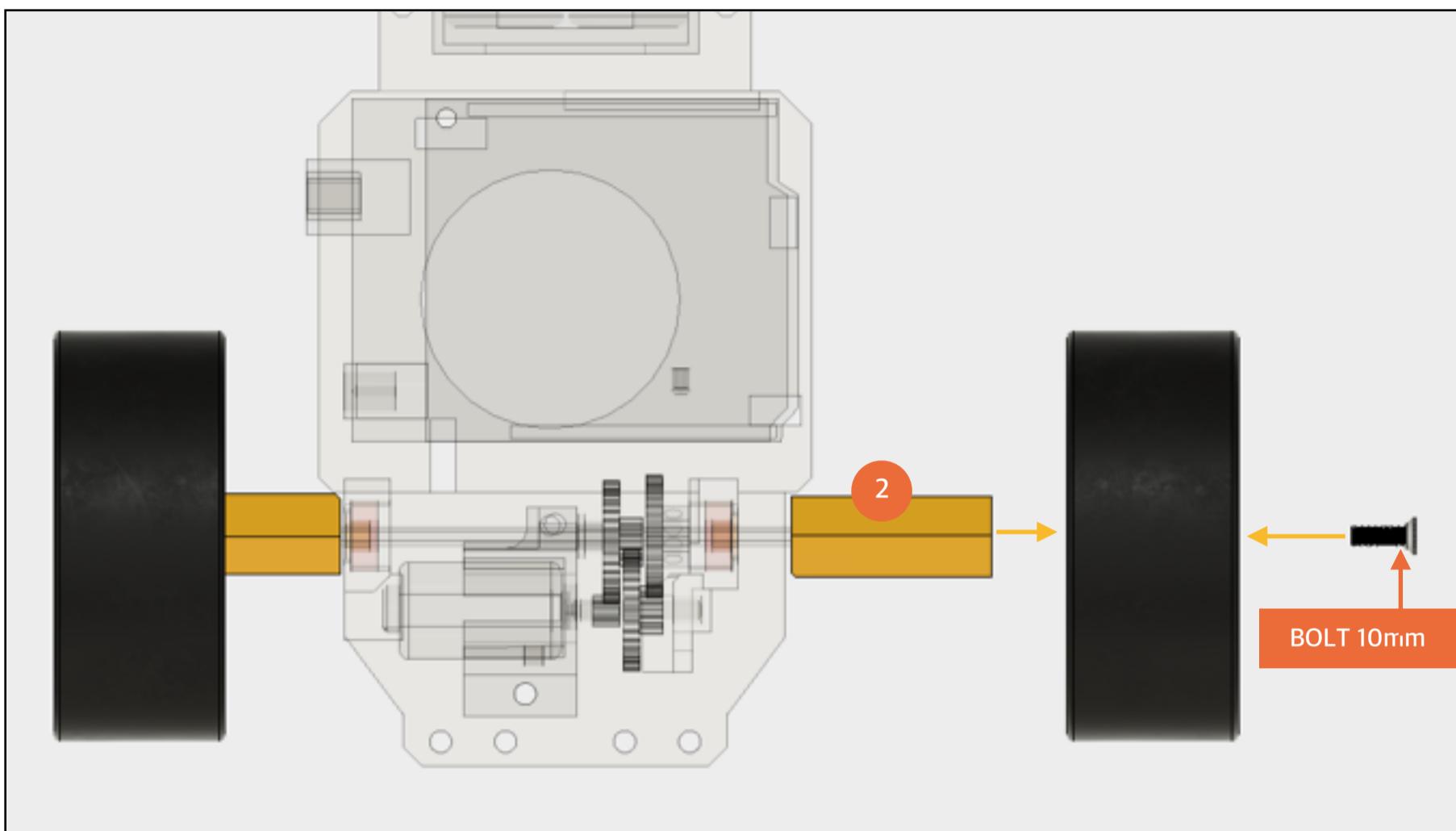
11 BOLT
30mm x 4

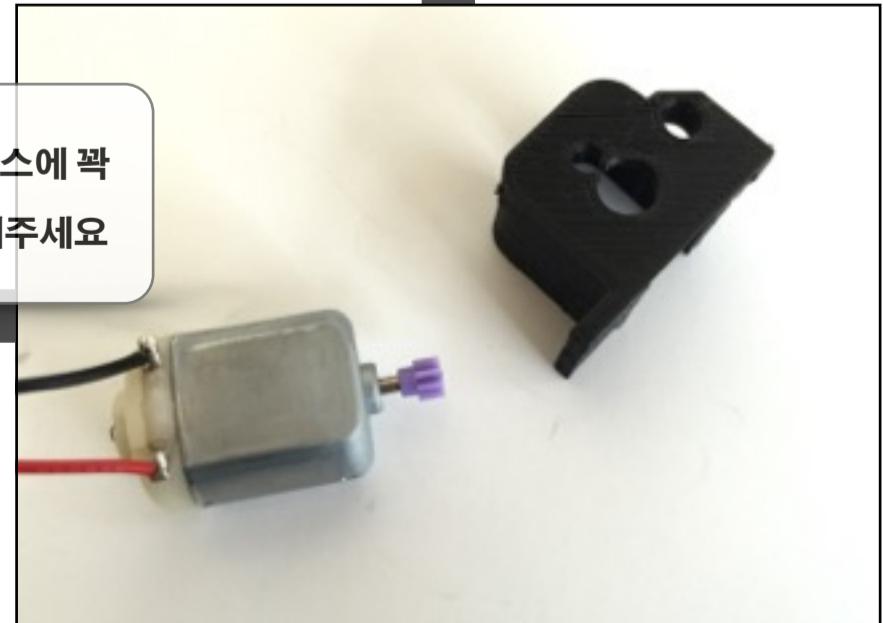
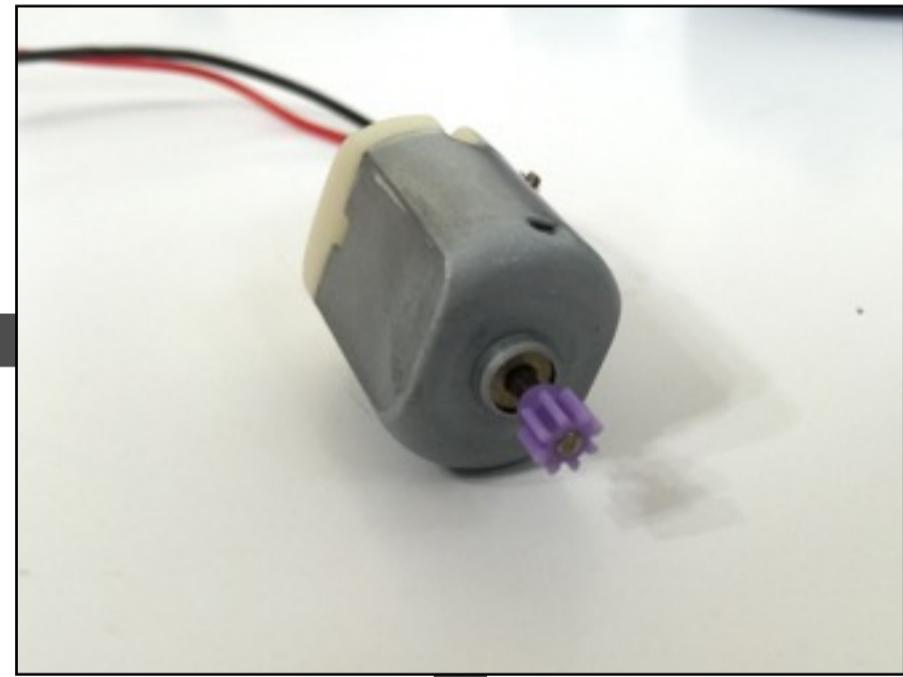
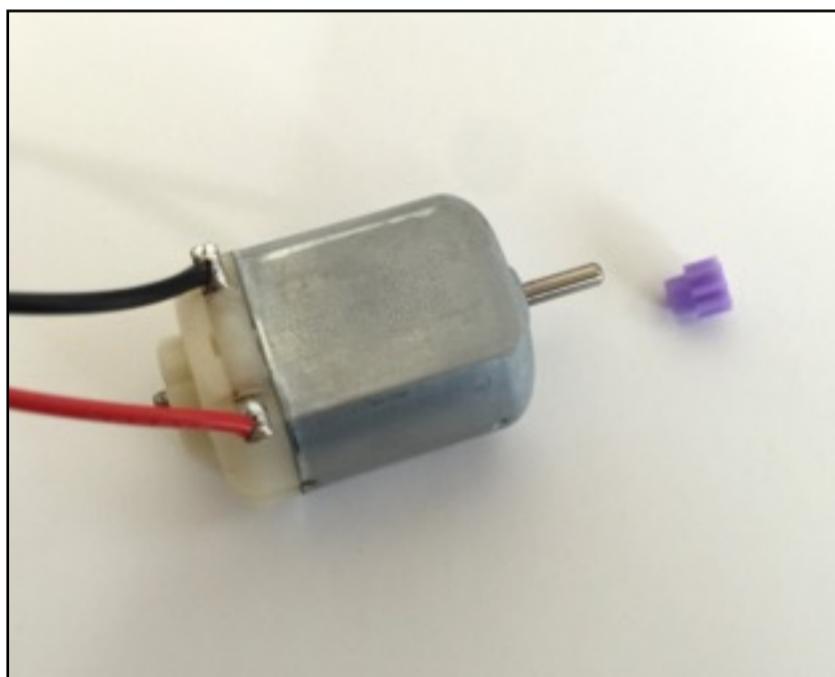
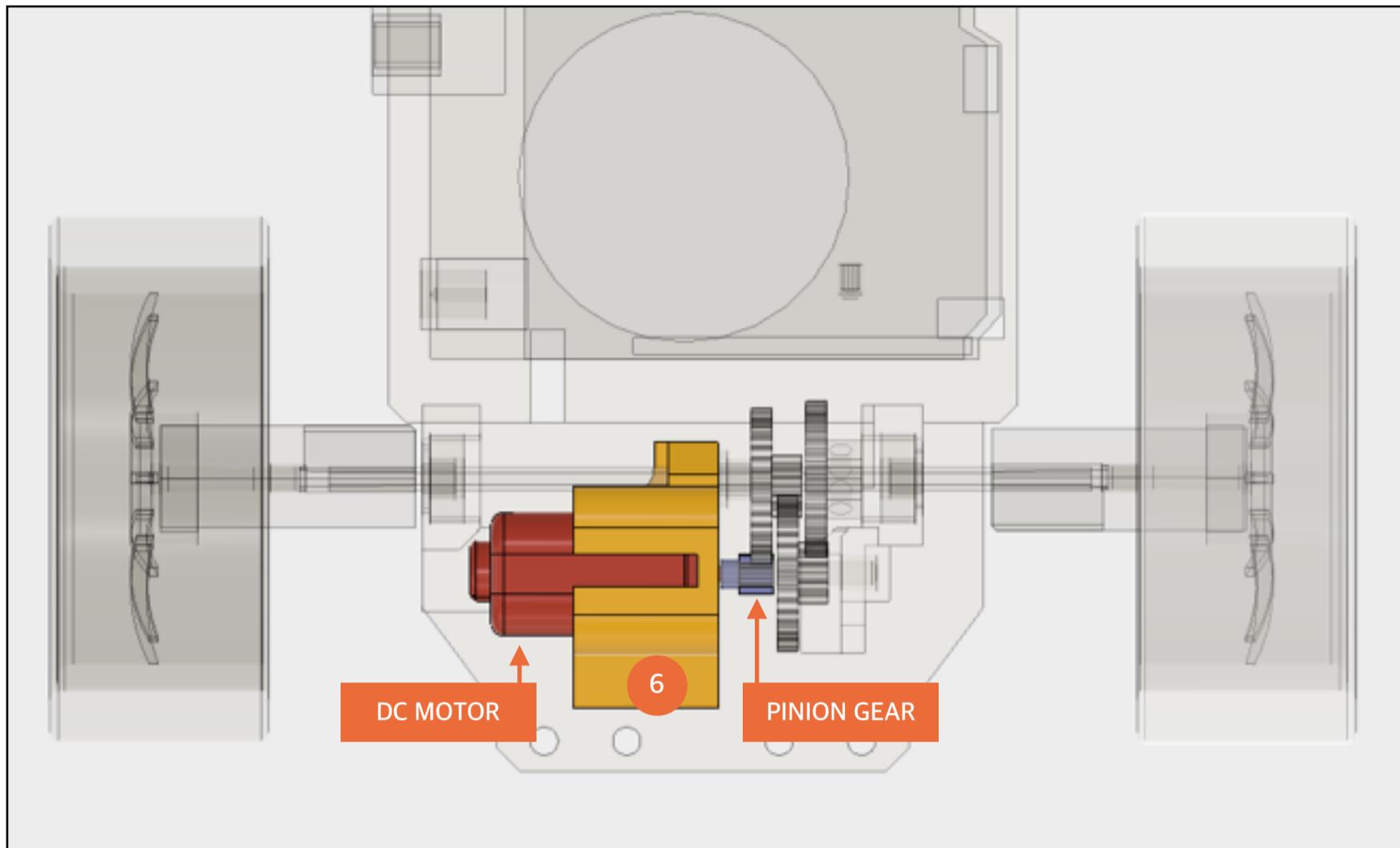
10 BOLT
40mm x 2

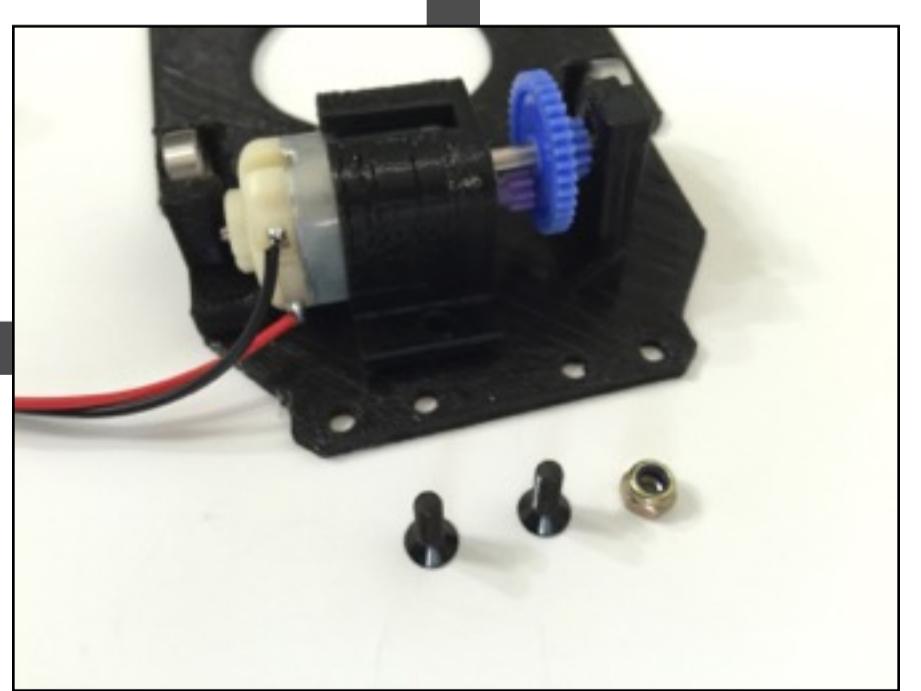
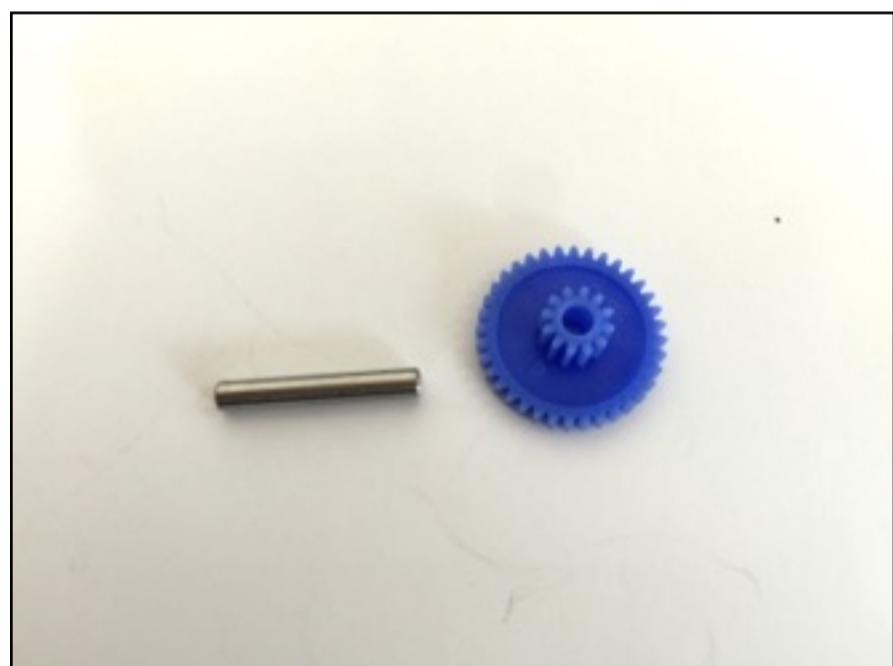
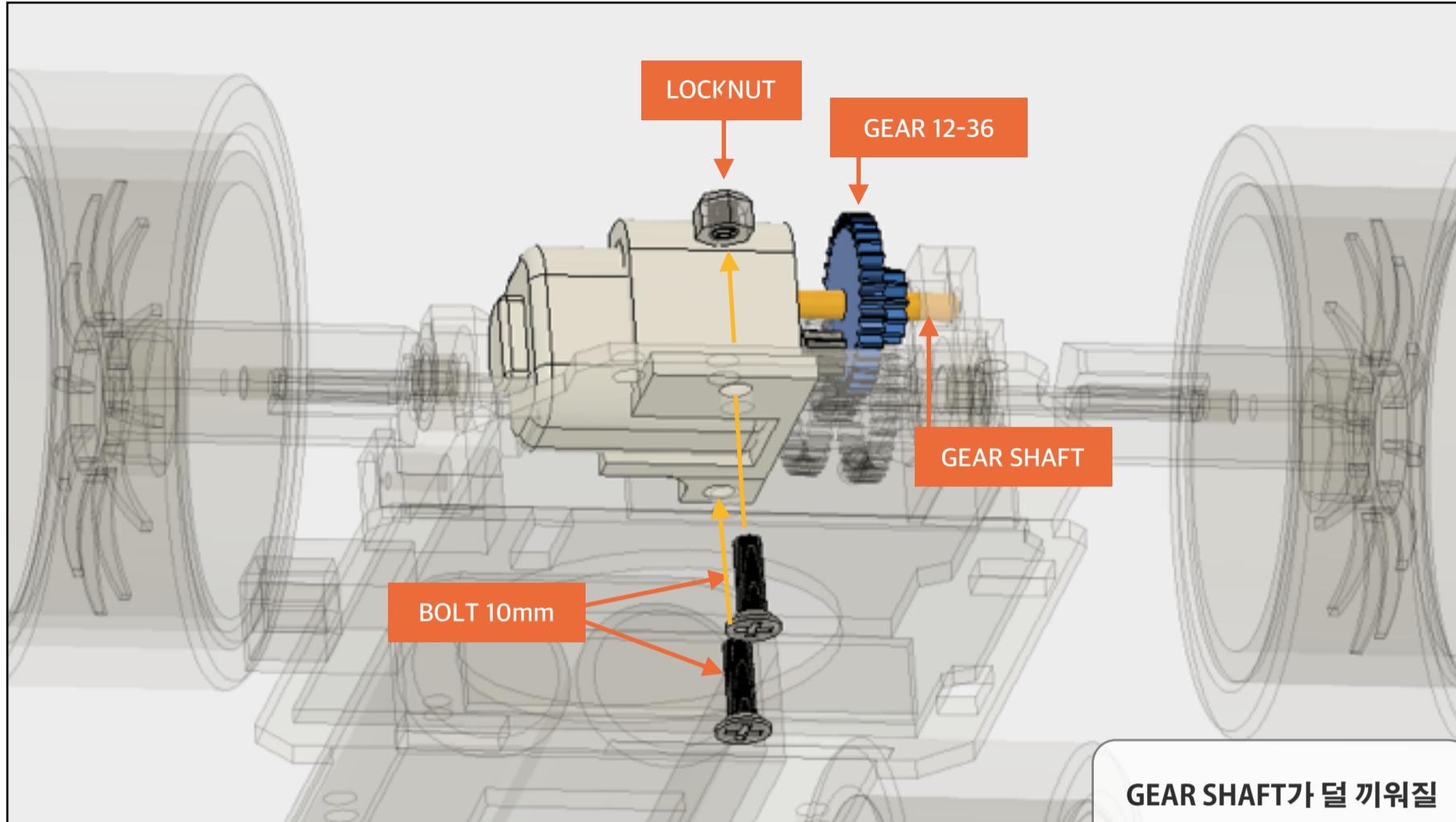
STEP 1 - RC CAR 조립하기

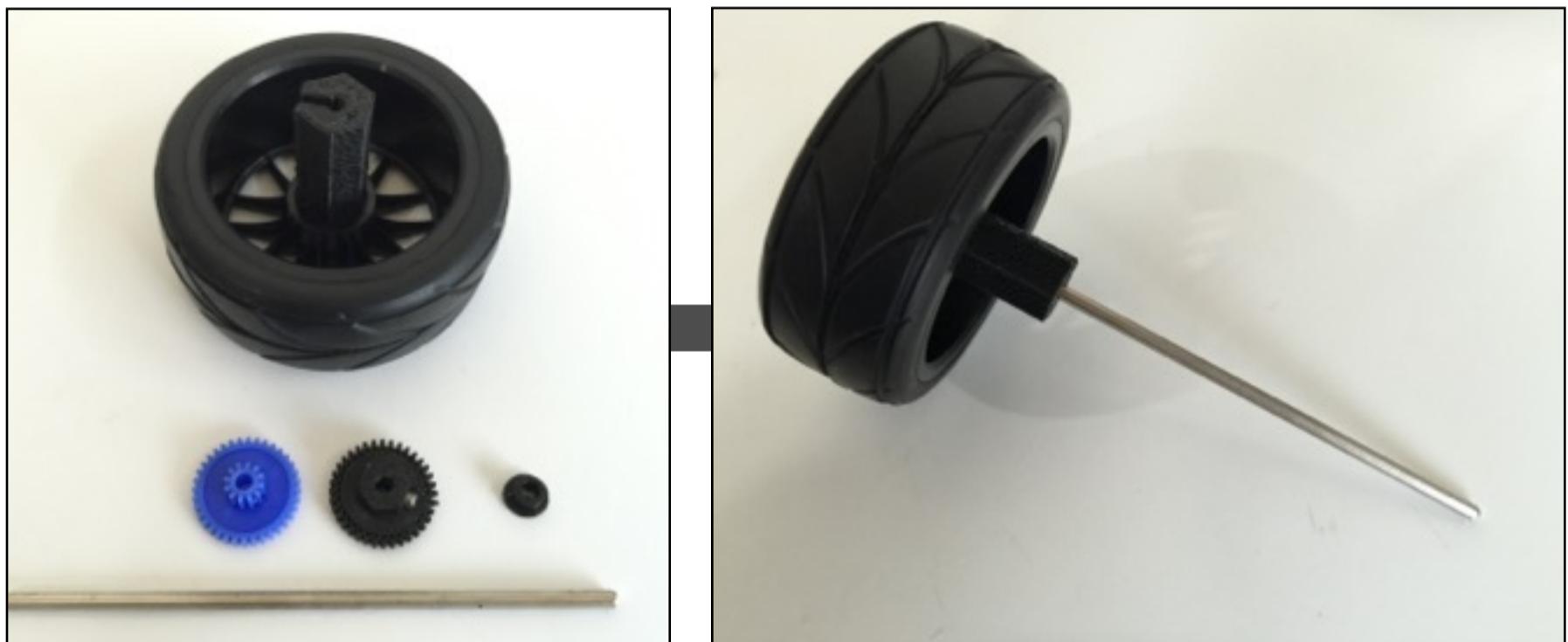
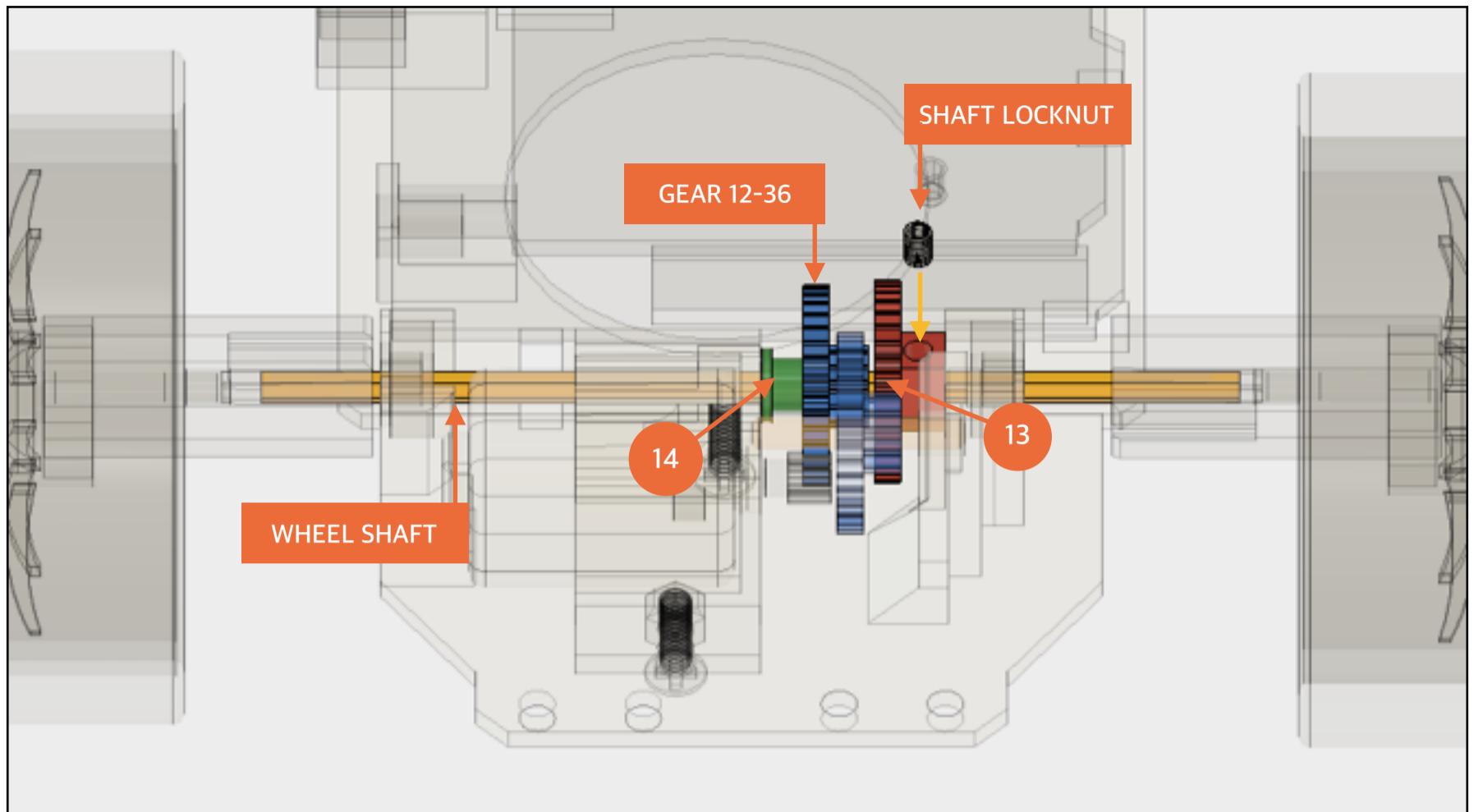


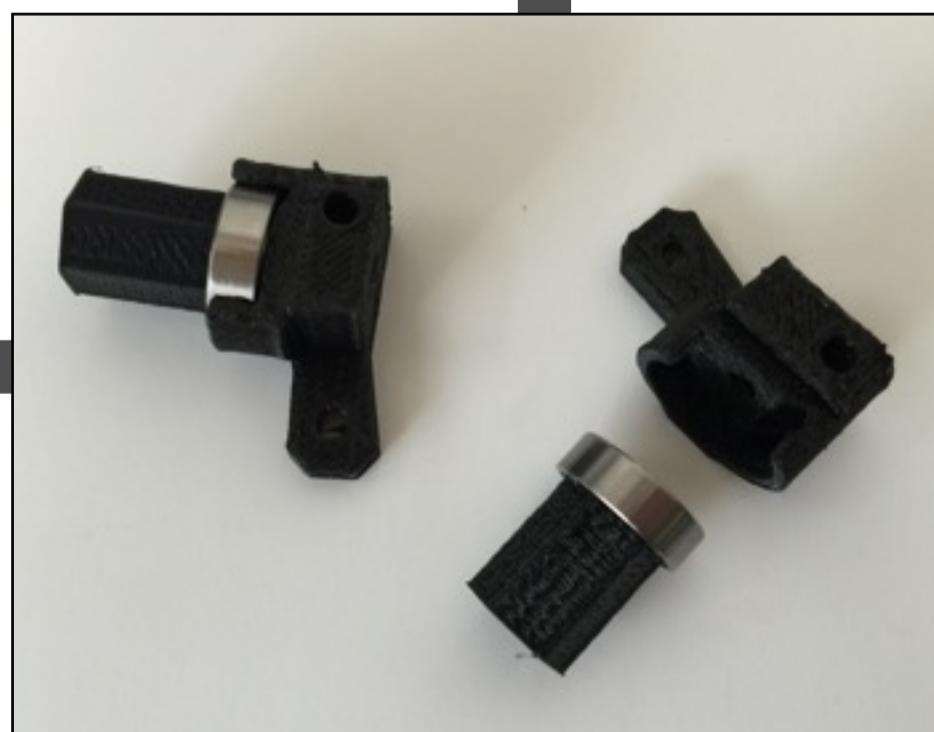
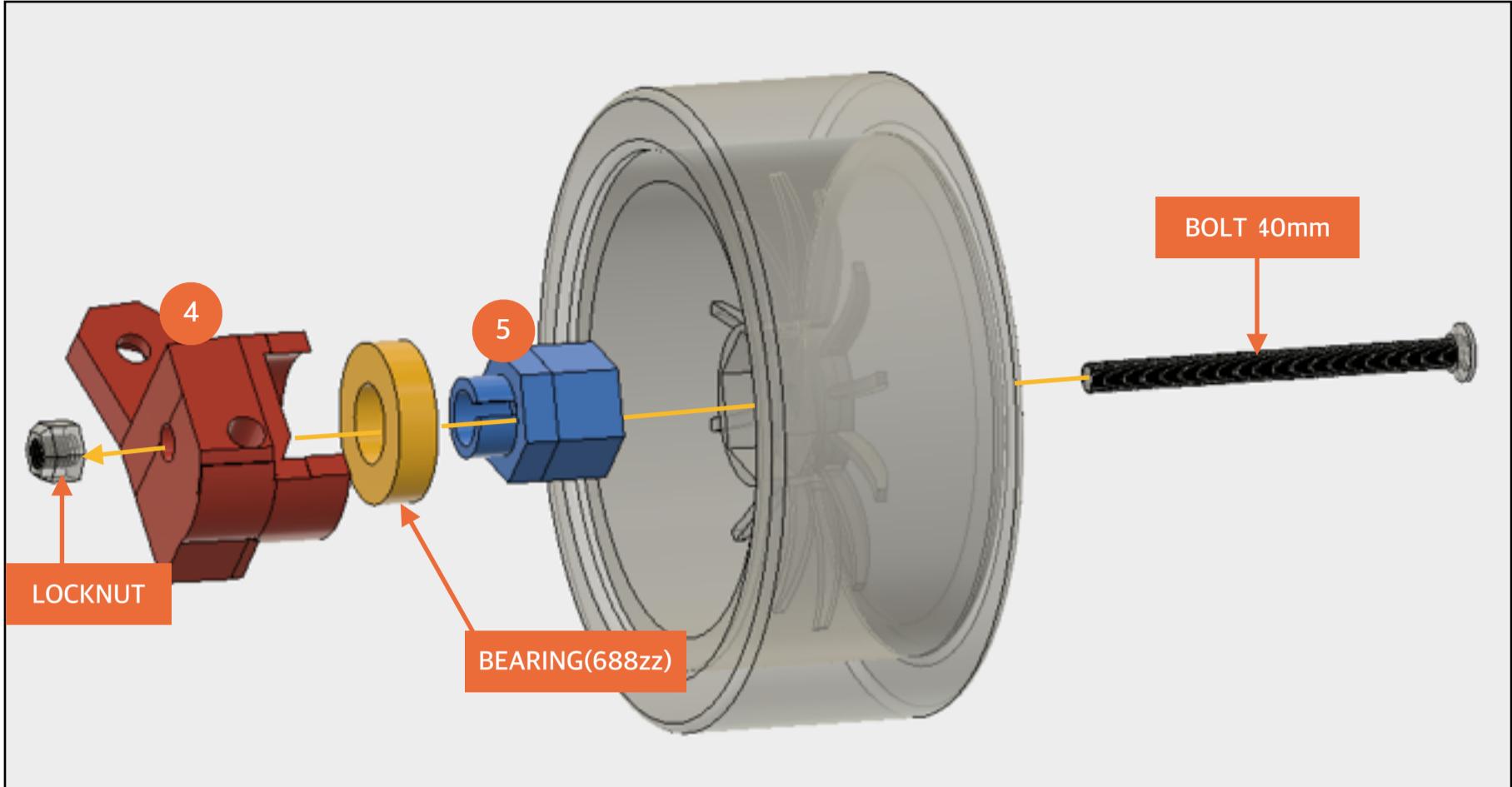


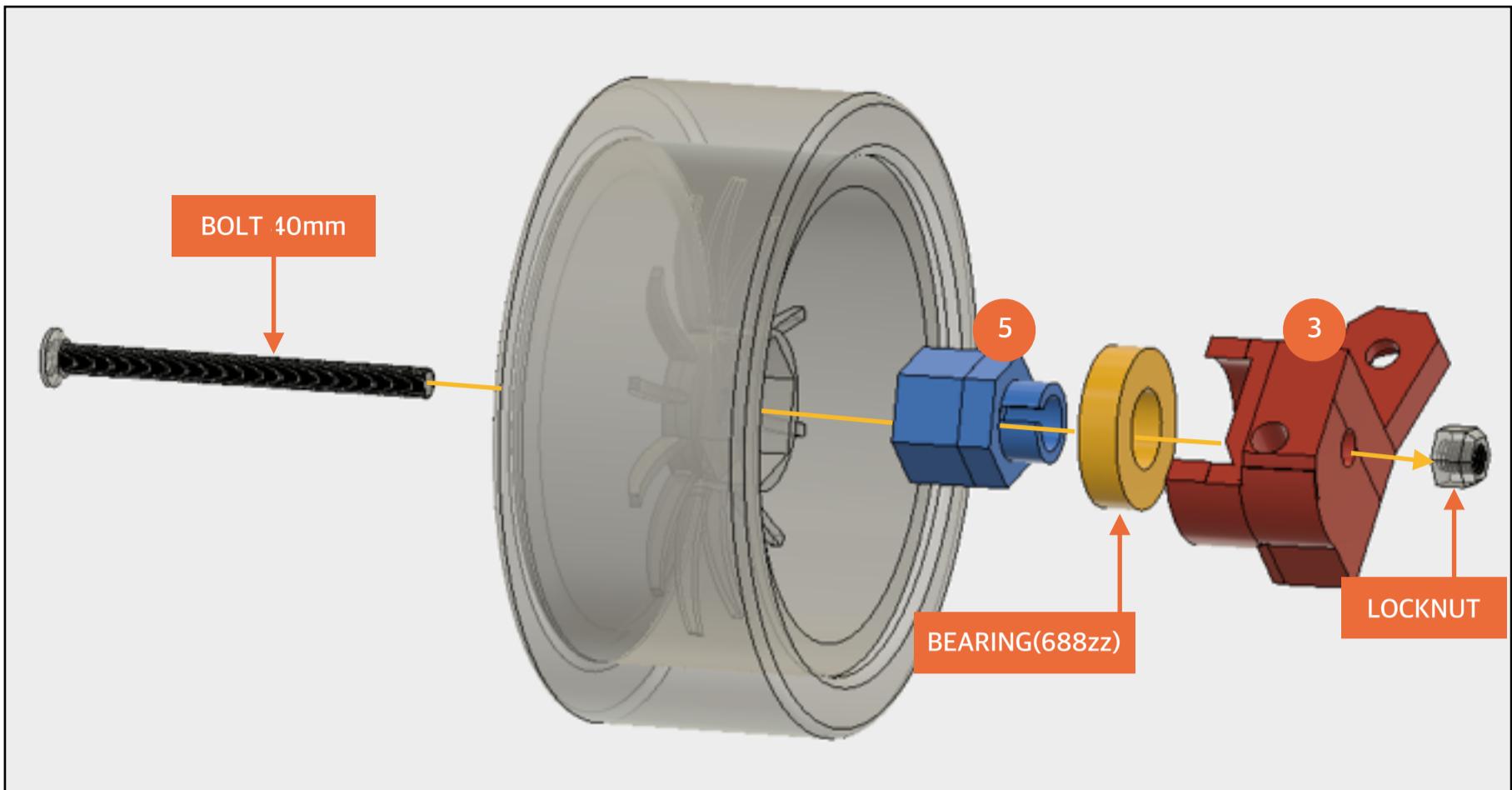


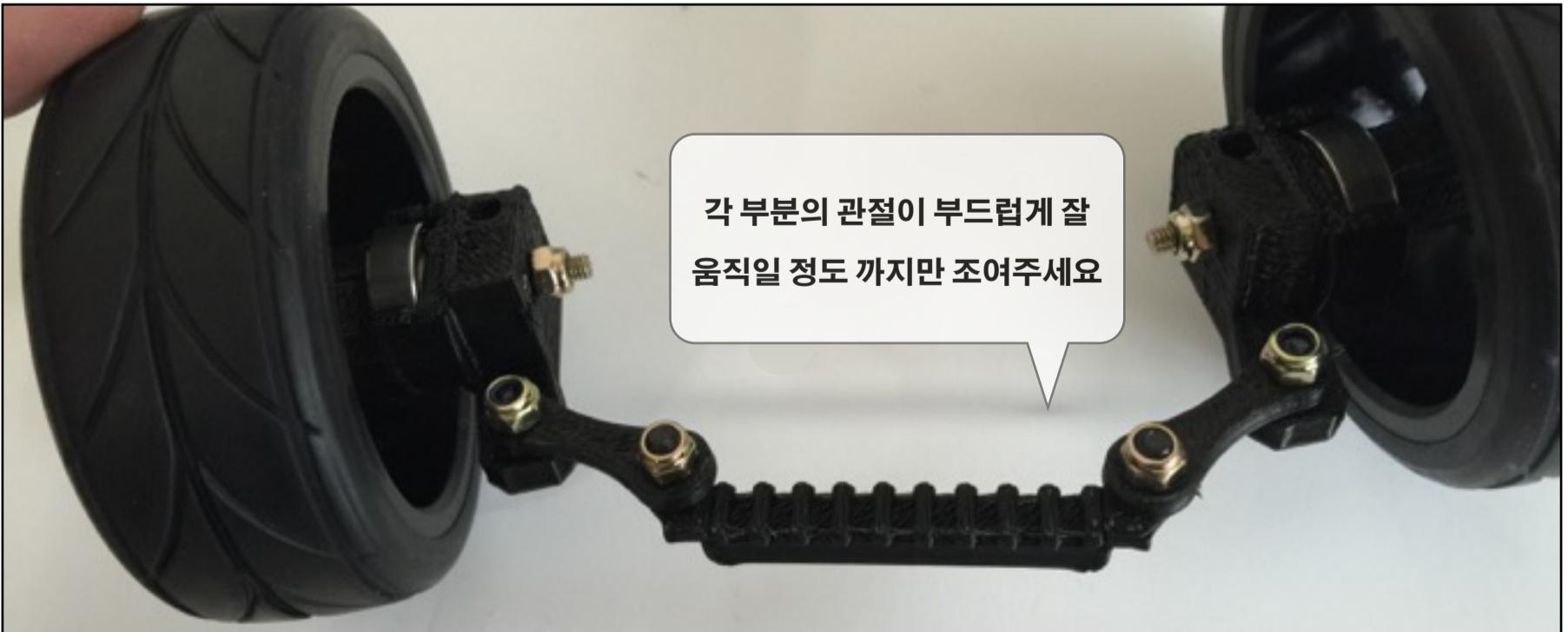
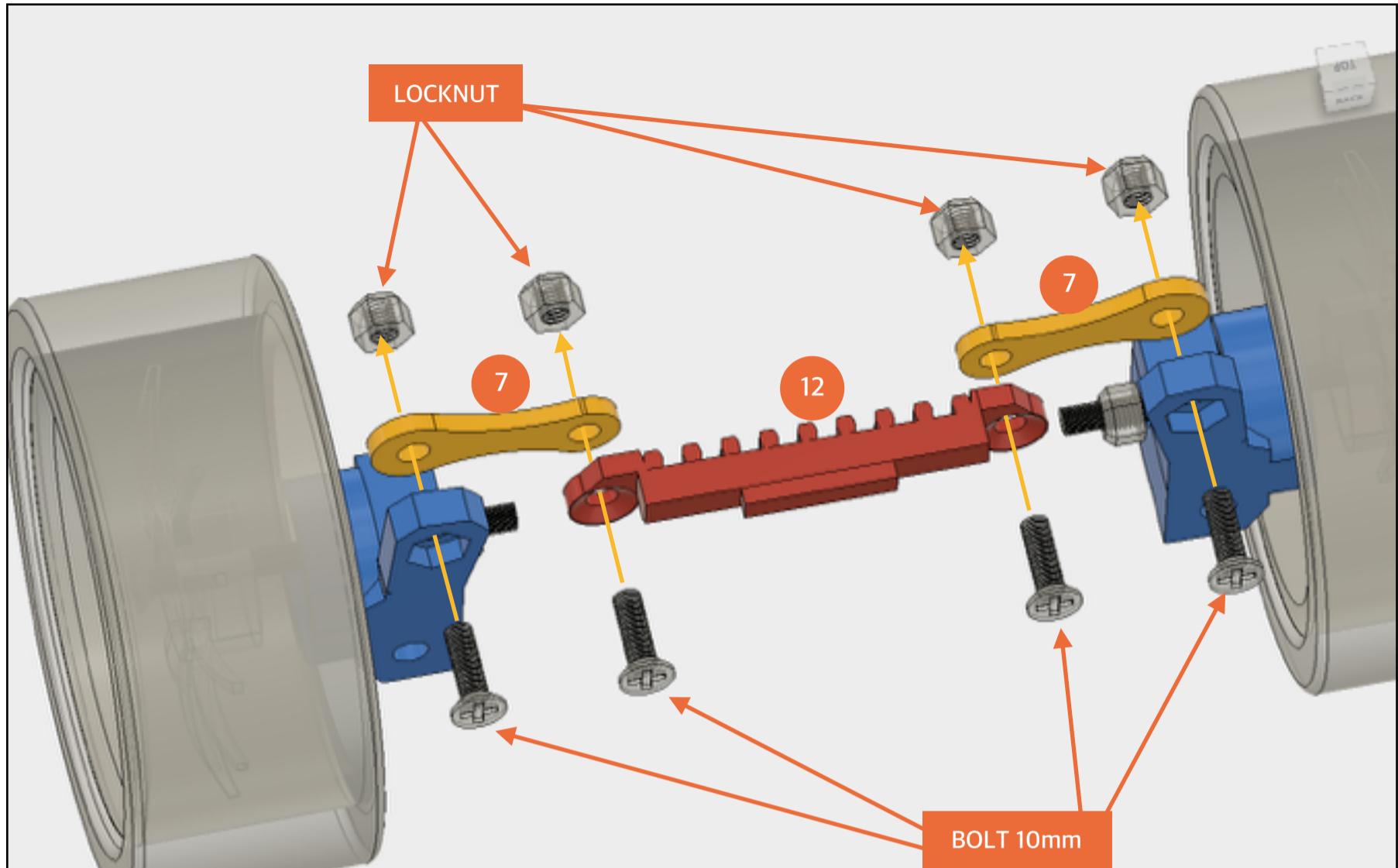


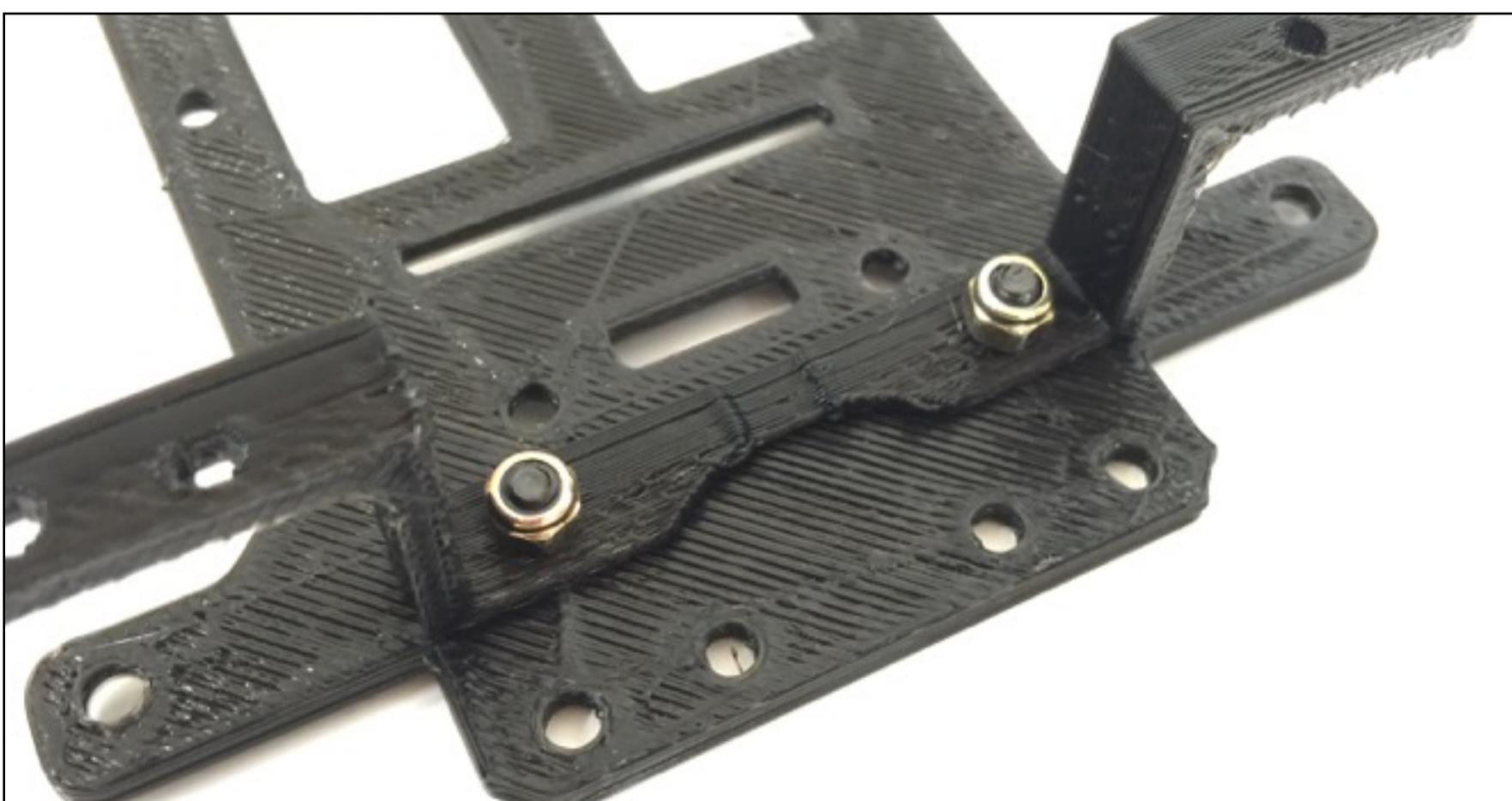
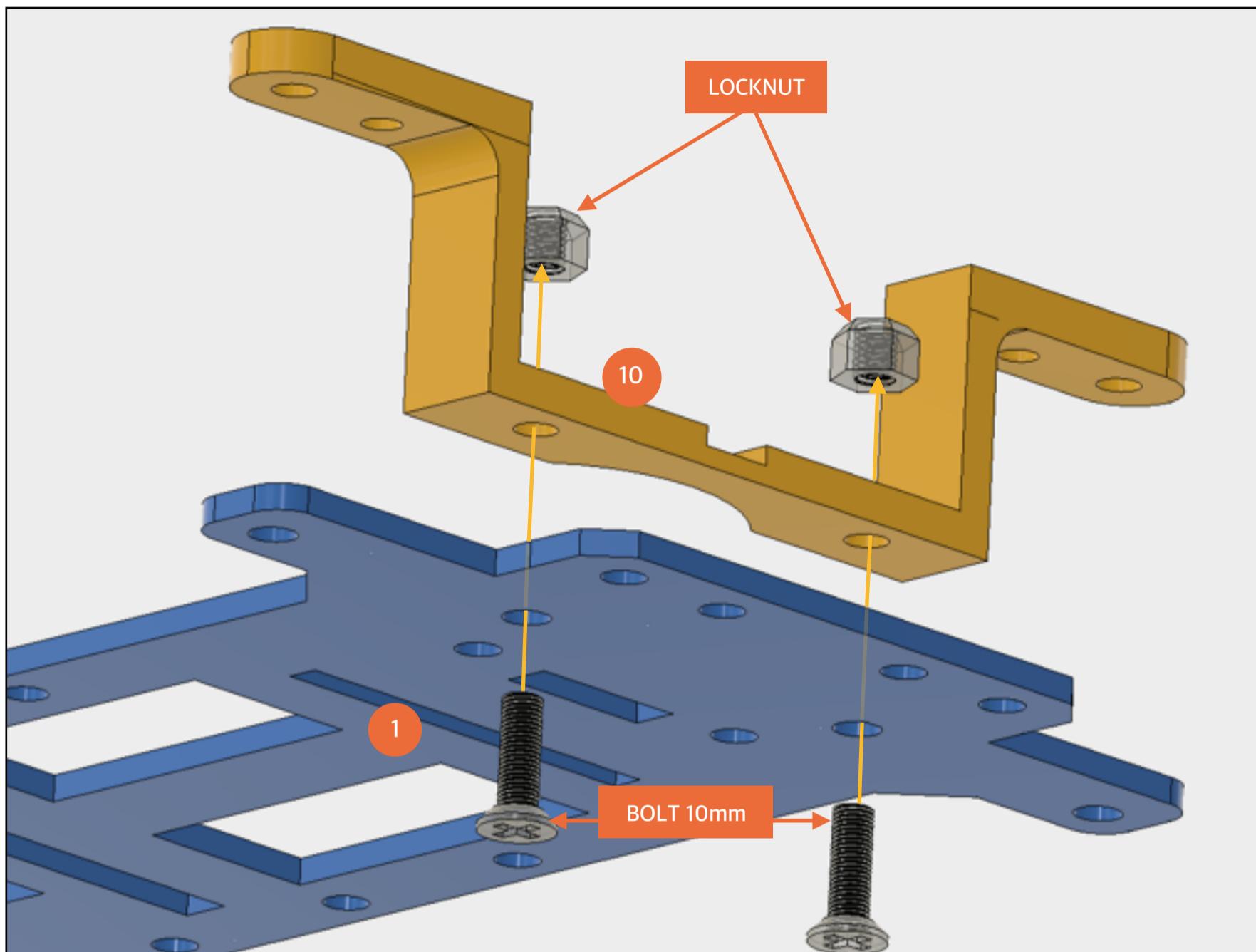


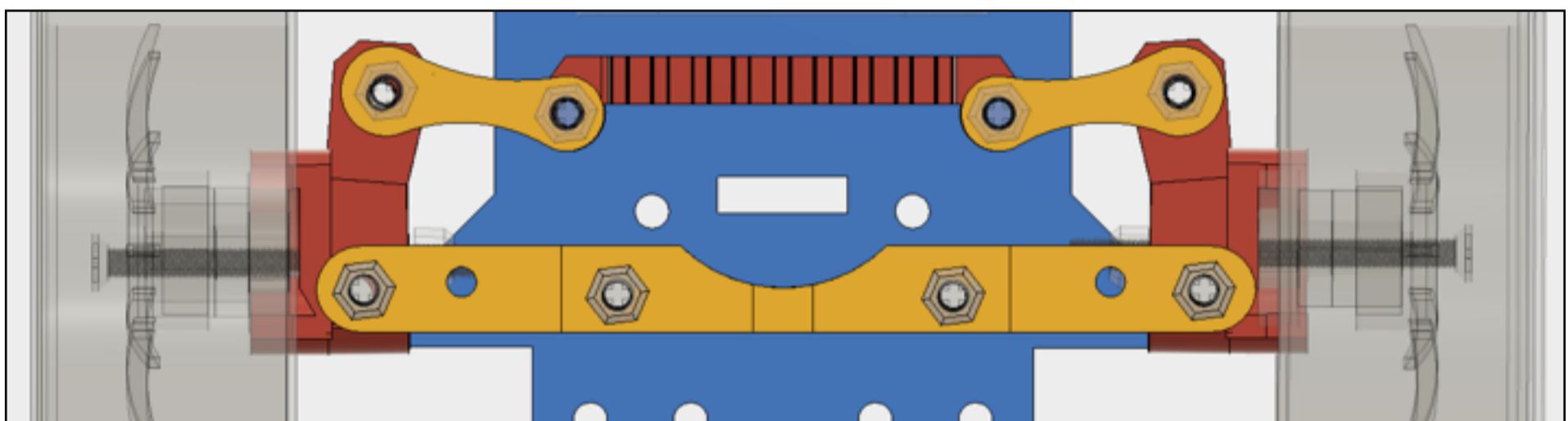
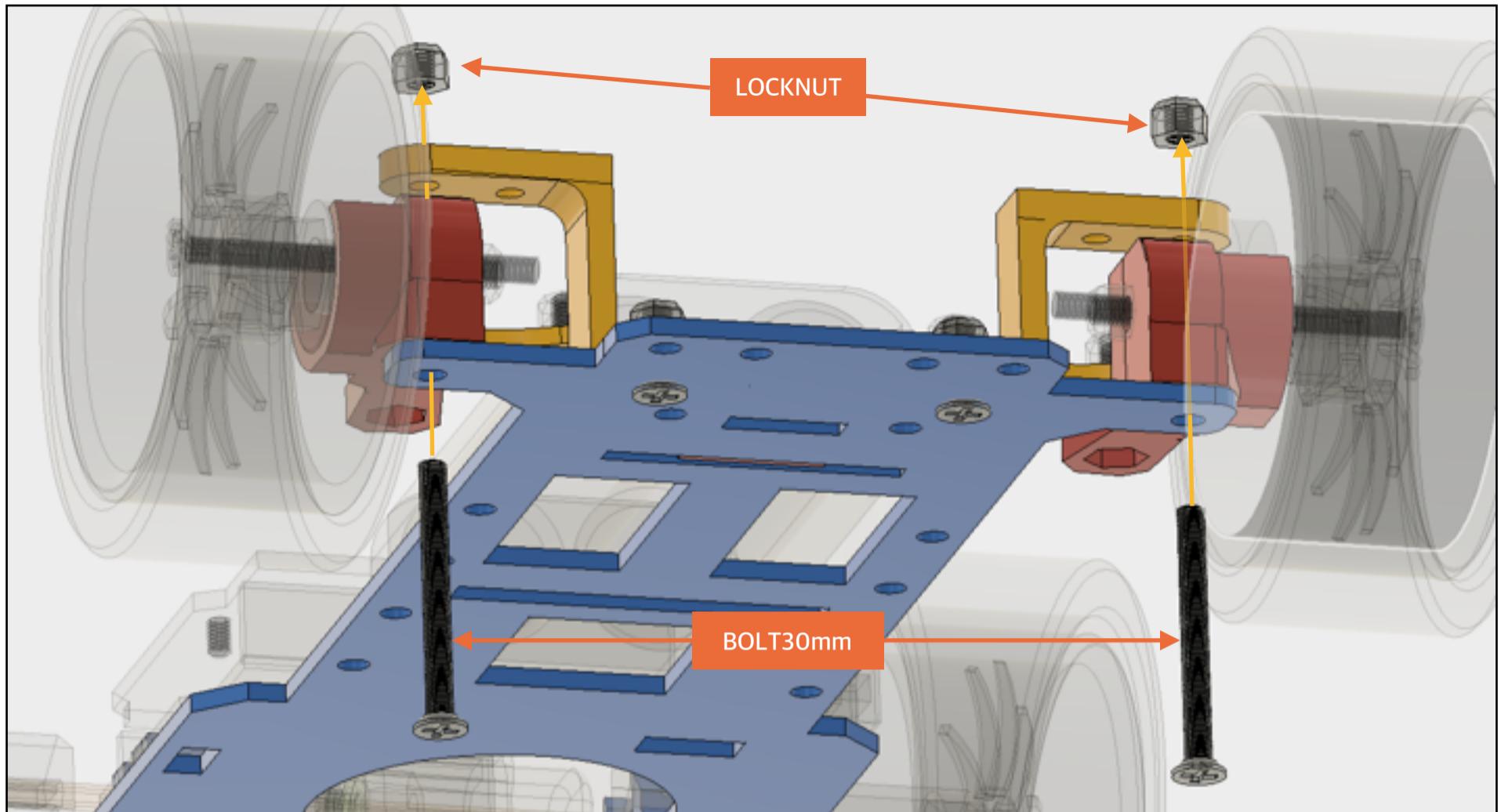


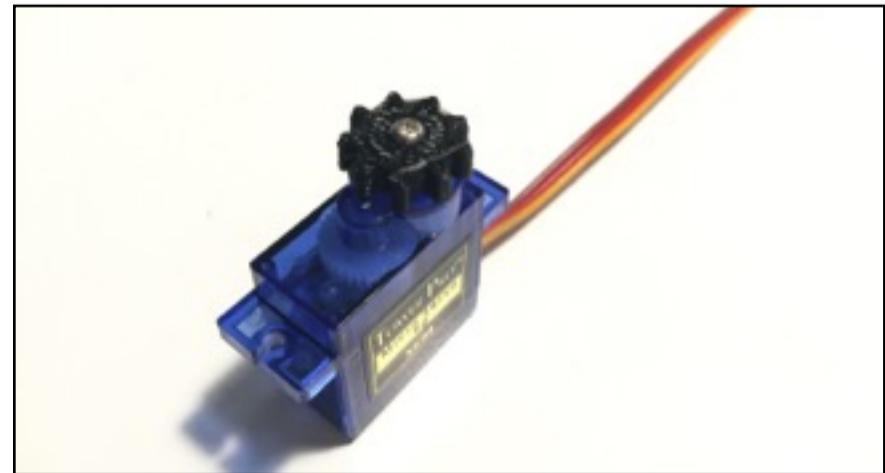
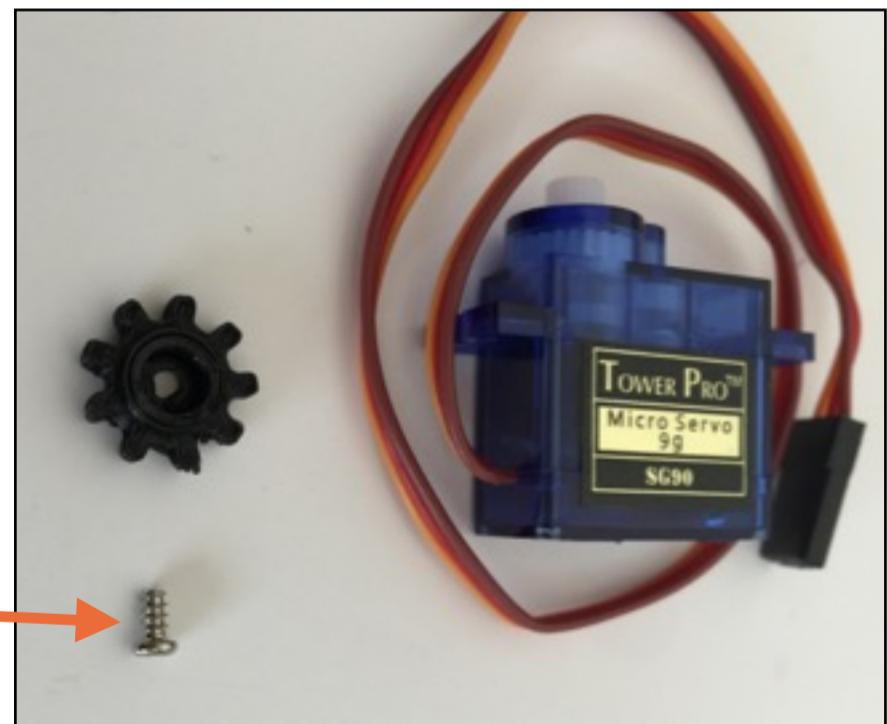
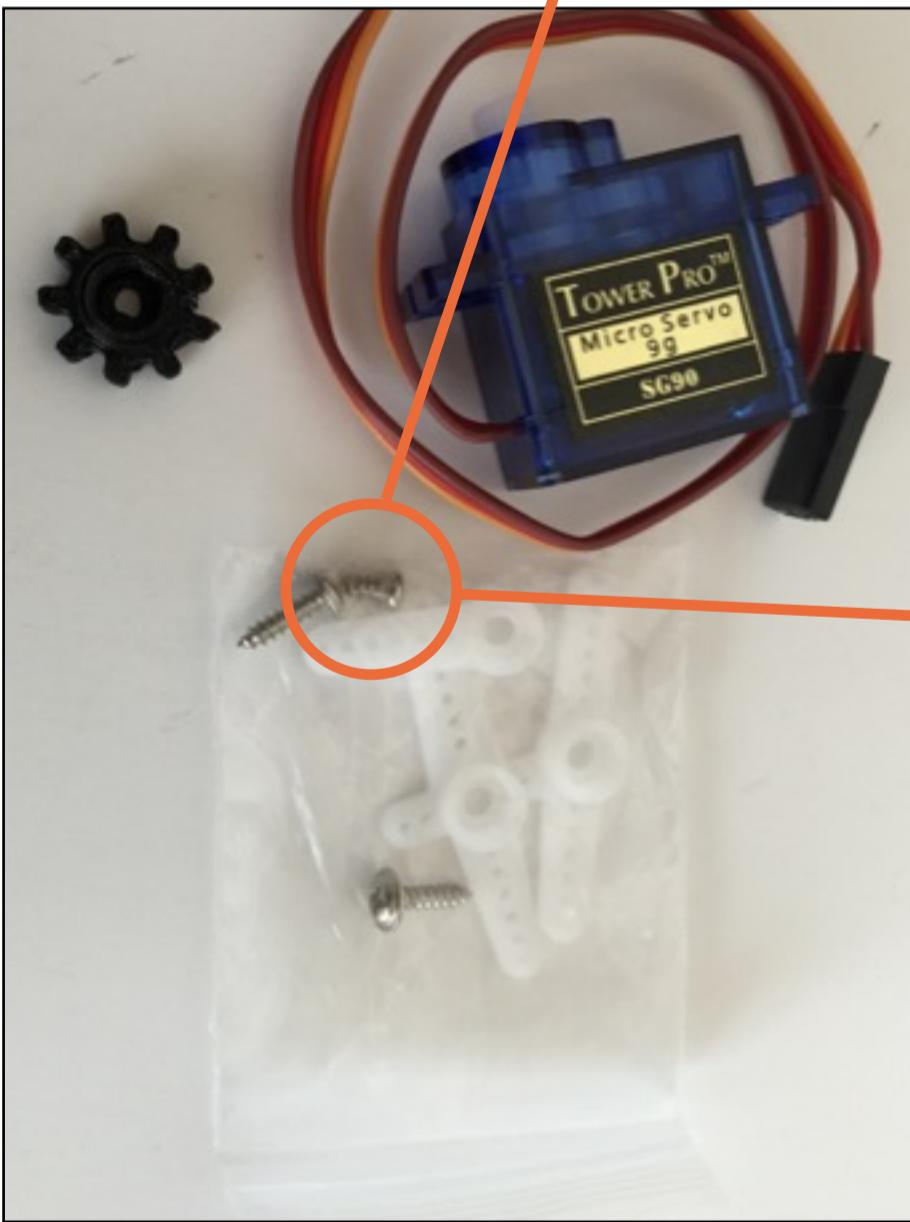
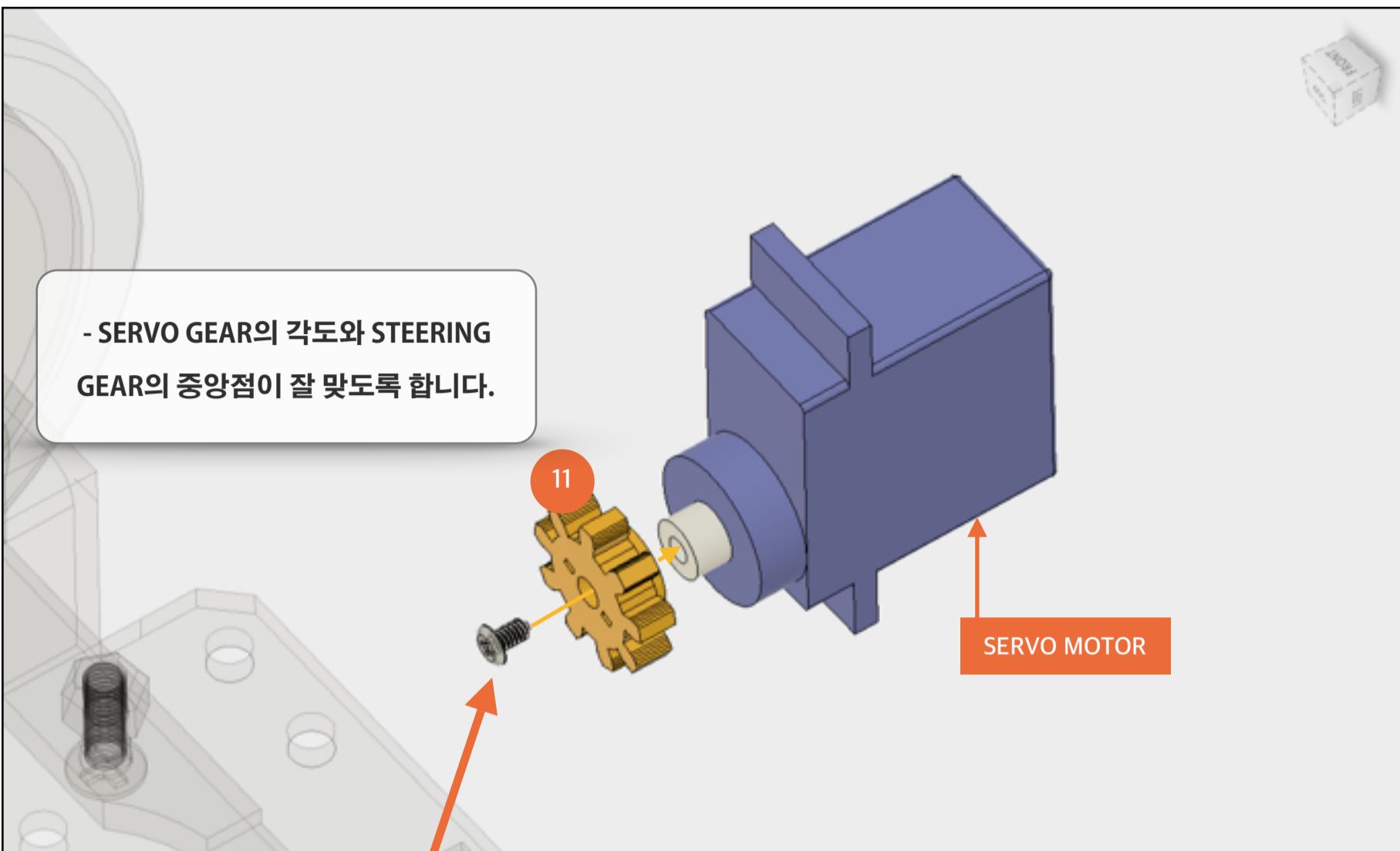


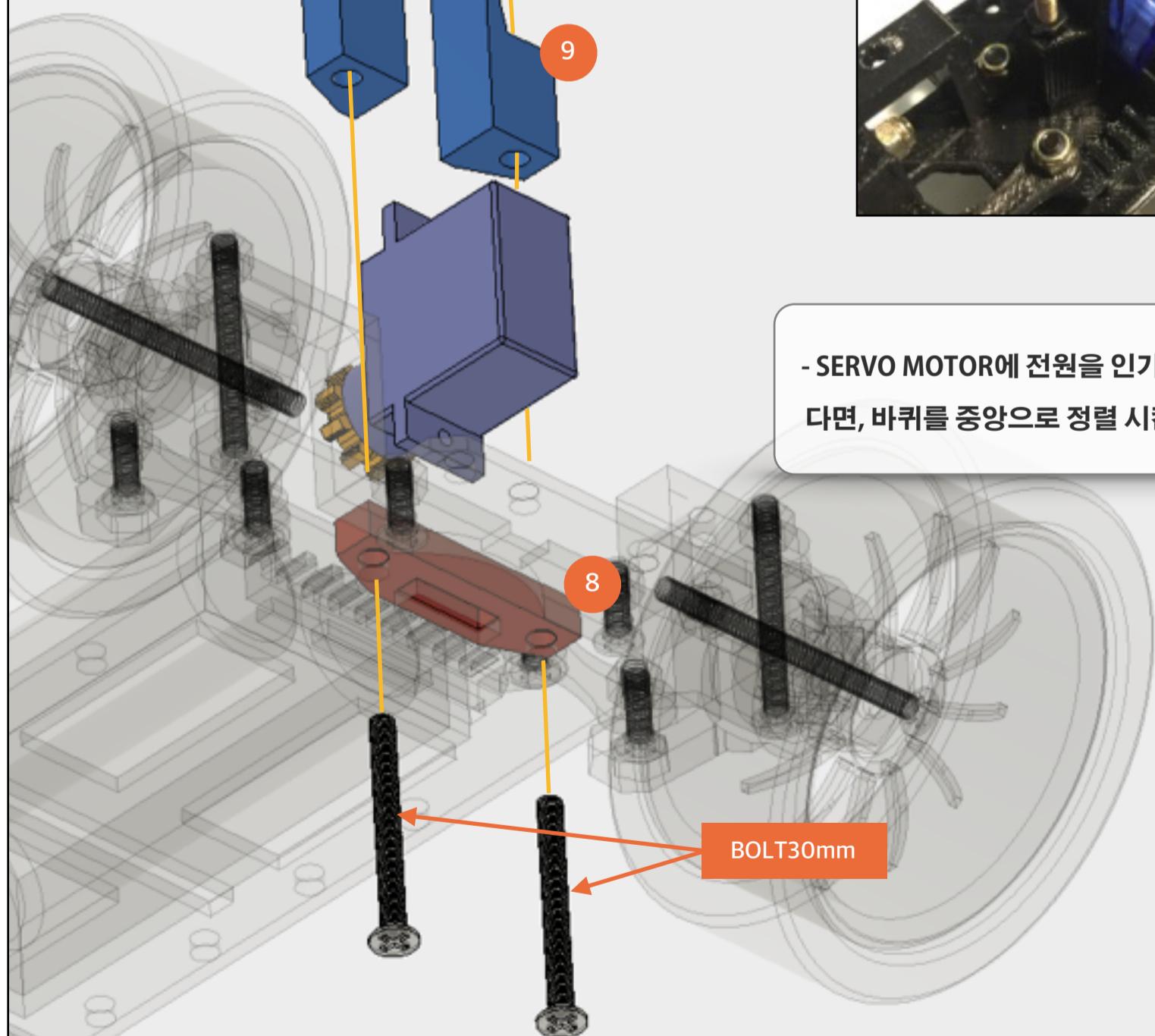
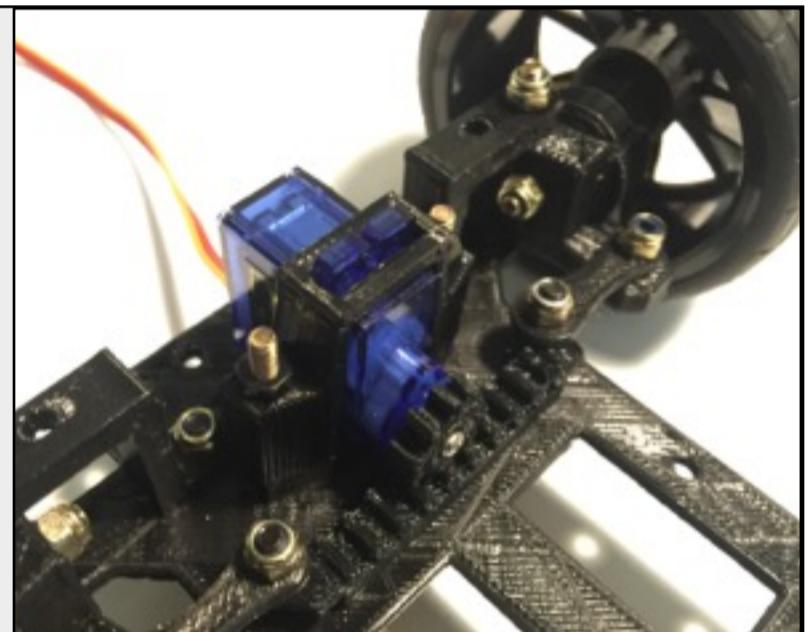
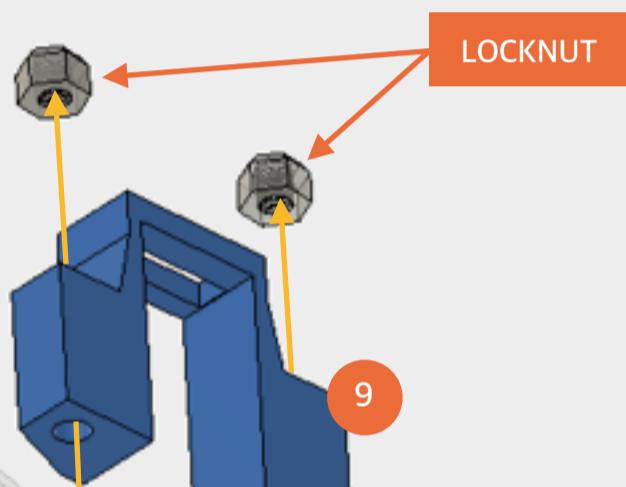
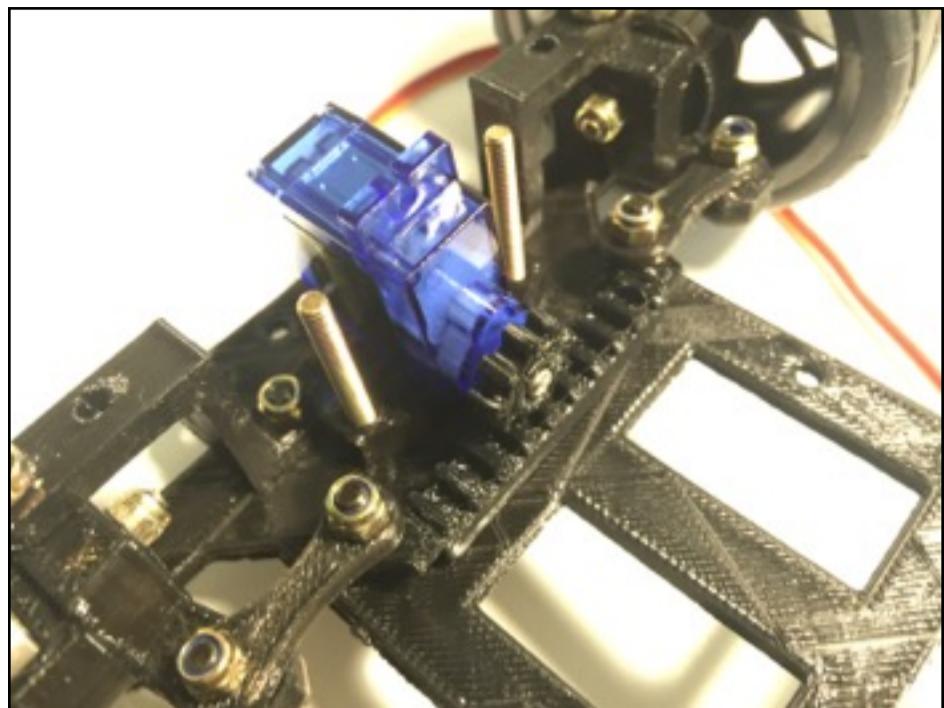
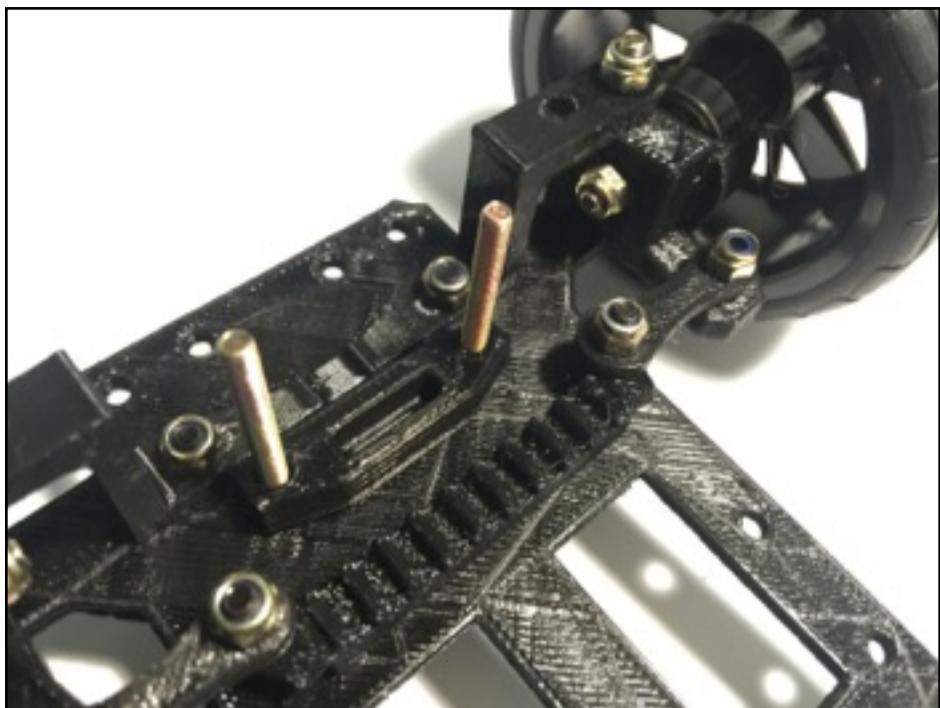






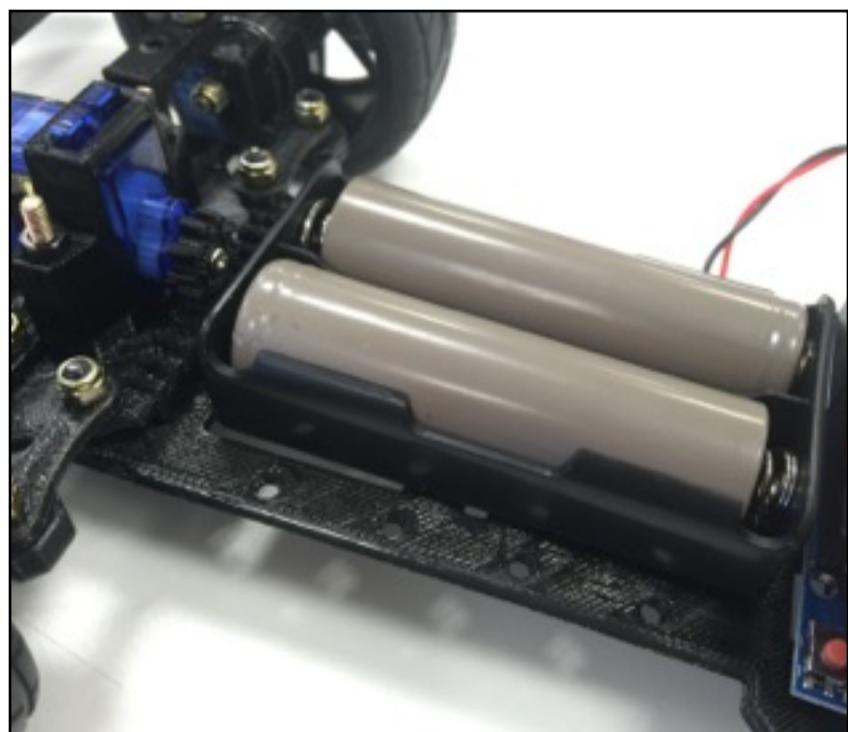
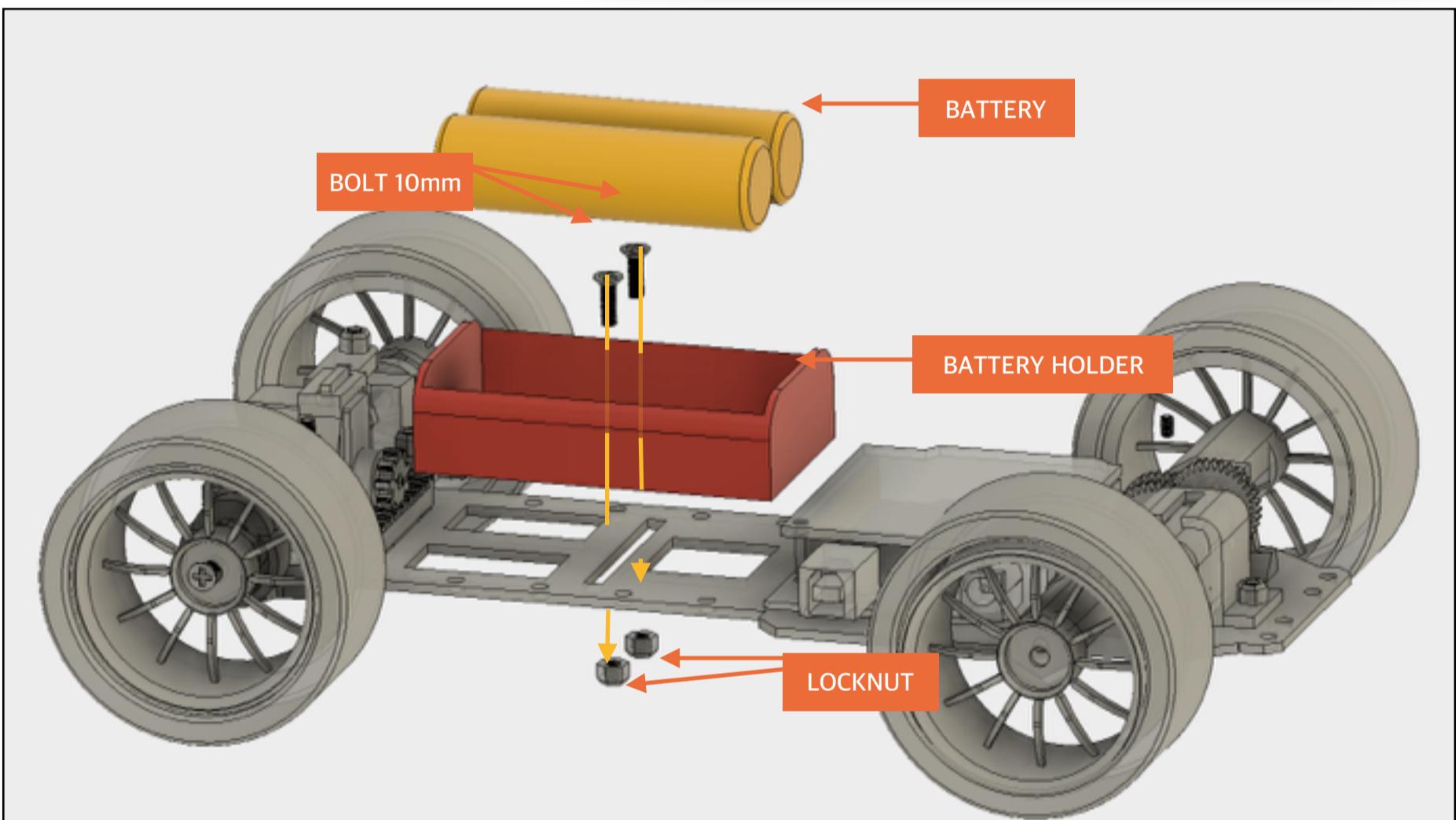


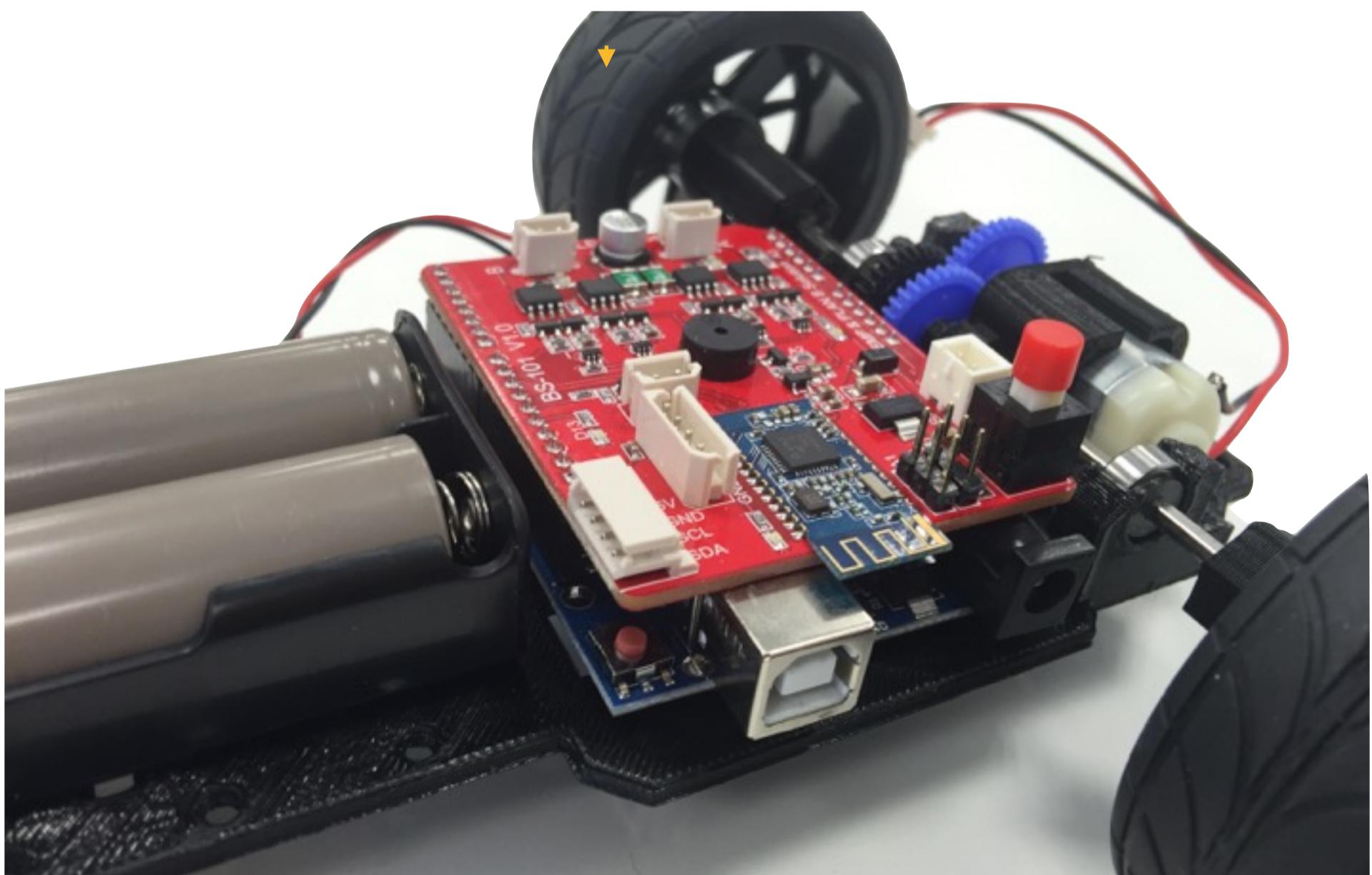
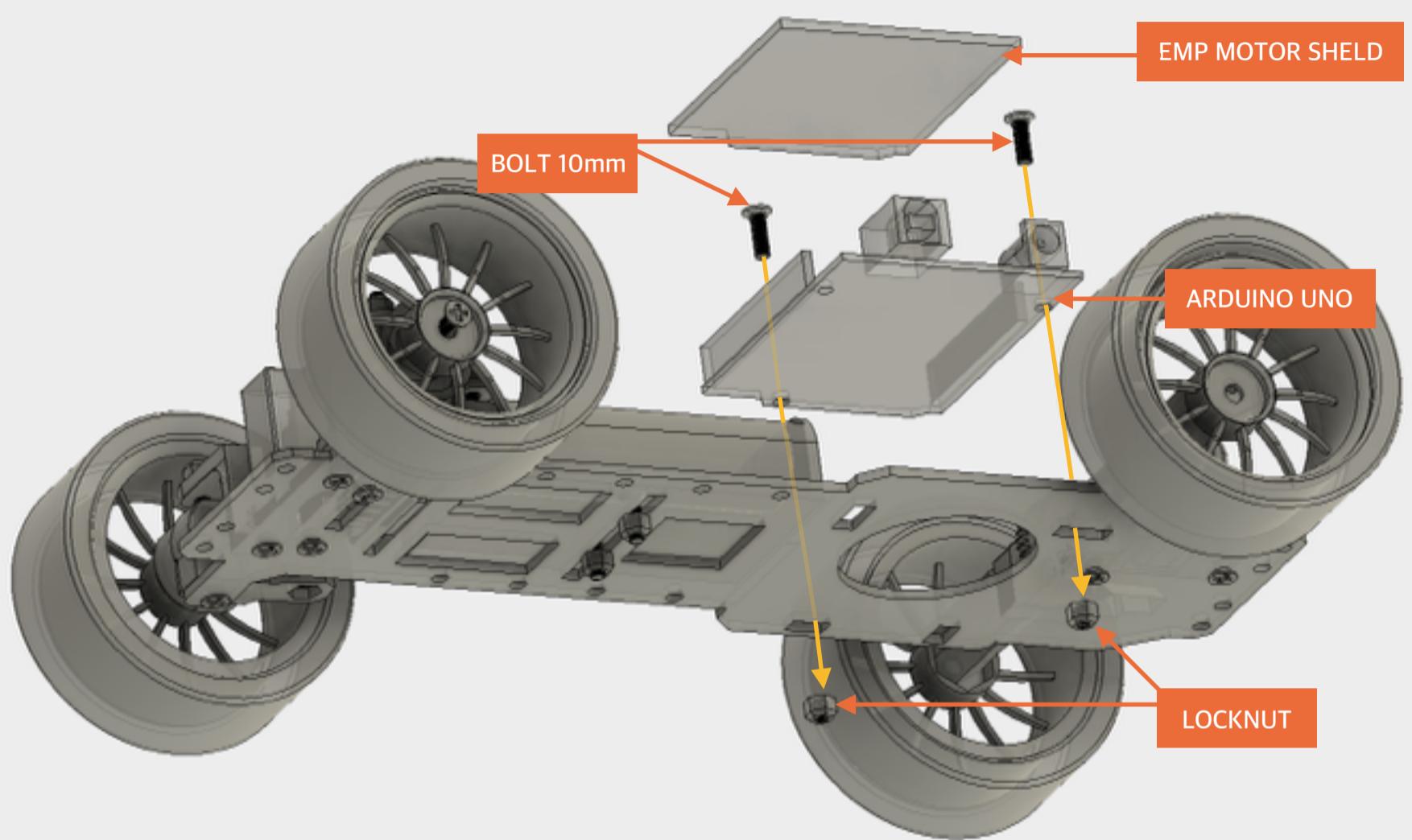






- 배터리를 홀더에 넣을때 +, - 를 잘 구분해서 넣어 주어야 합니다.
- 잘못 연결해서 연기가 날경우에는 당황하지 않고 배터리를 다시 빼줍니다.

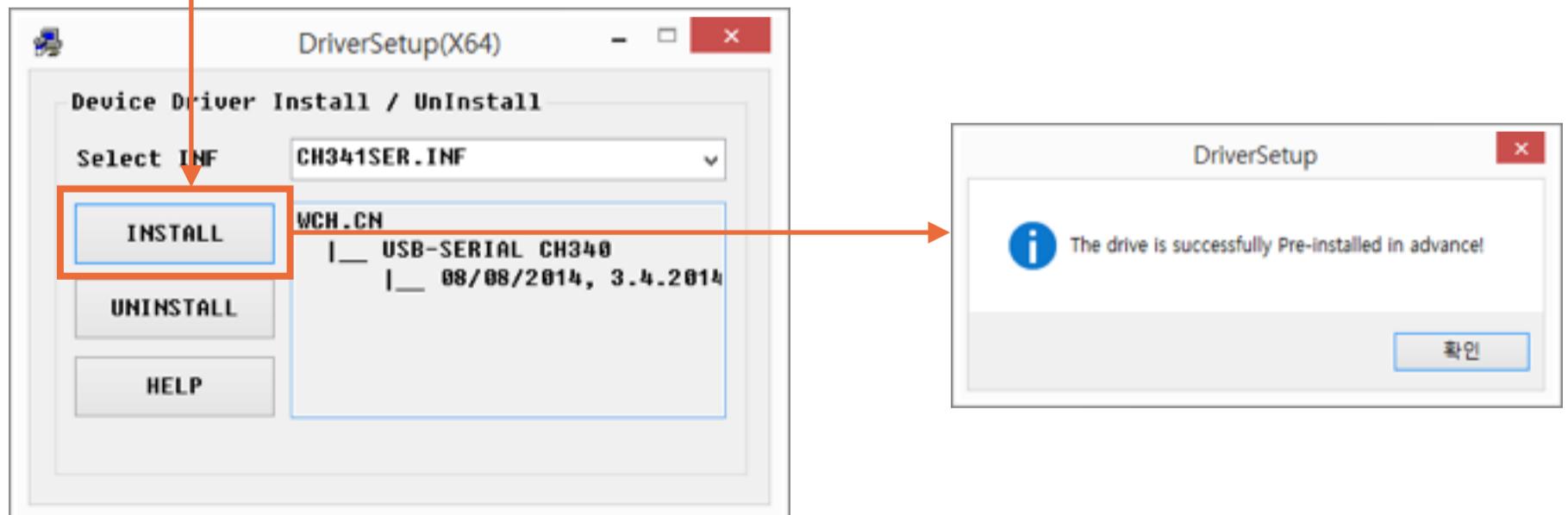
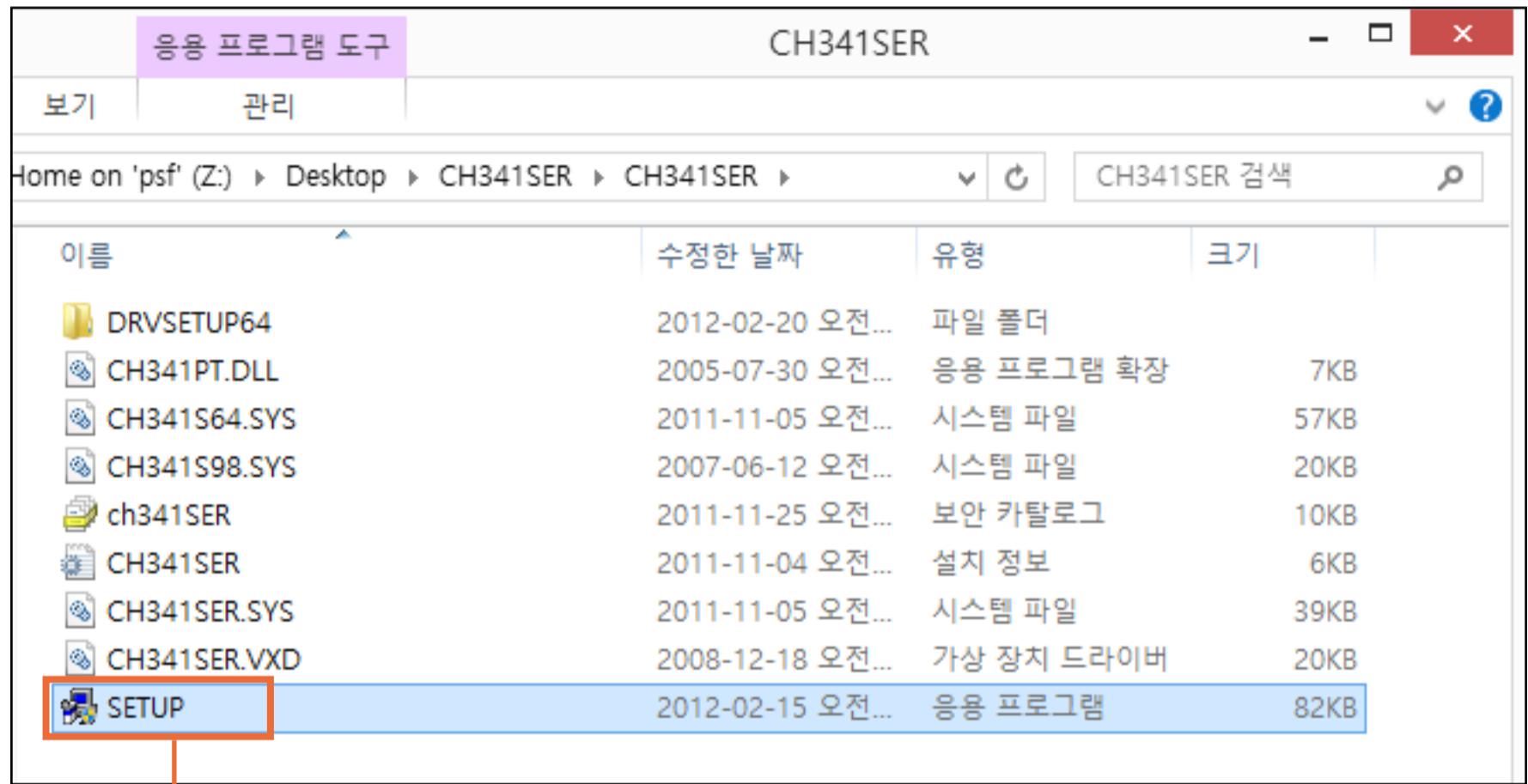




STEP 2 - 아두이노 세팅

1. 아두이노 드라이버 설치

- 아두이노를 컴퓨터에 연결하기 위해서 CH341SER 드라이버를 3DEMP 홈페이지에서 다운받은 후 설치한다.
- 아두이노 정품은 별도의 세팅이 필요 없지만 CH340계열의 시리얼 어댑터를 사용하는 아두이노 복제품은 아래와 같은 추가 세팅이 필요하다.



2. 아두이노 스케치 프로그램 설치

- 아두이노 홈페이지(www.arduino.cc)에서 아두이노 소프트웨어를 다운받아 설치한다.

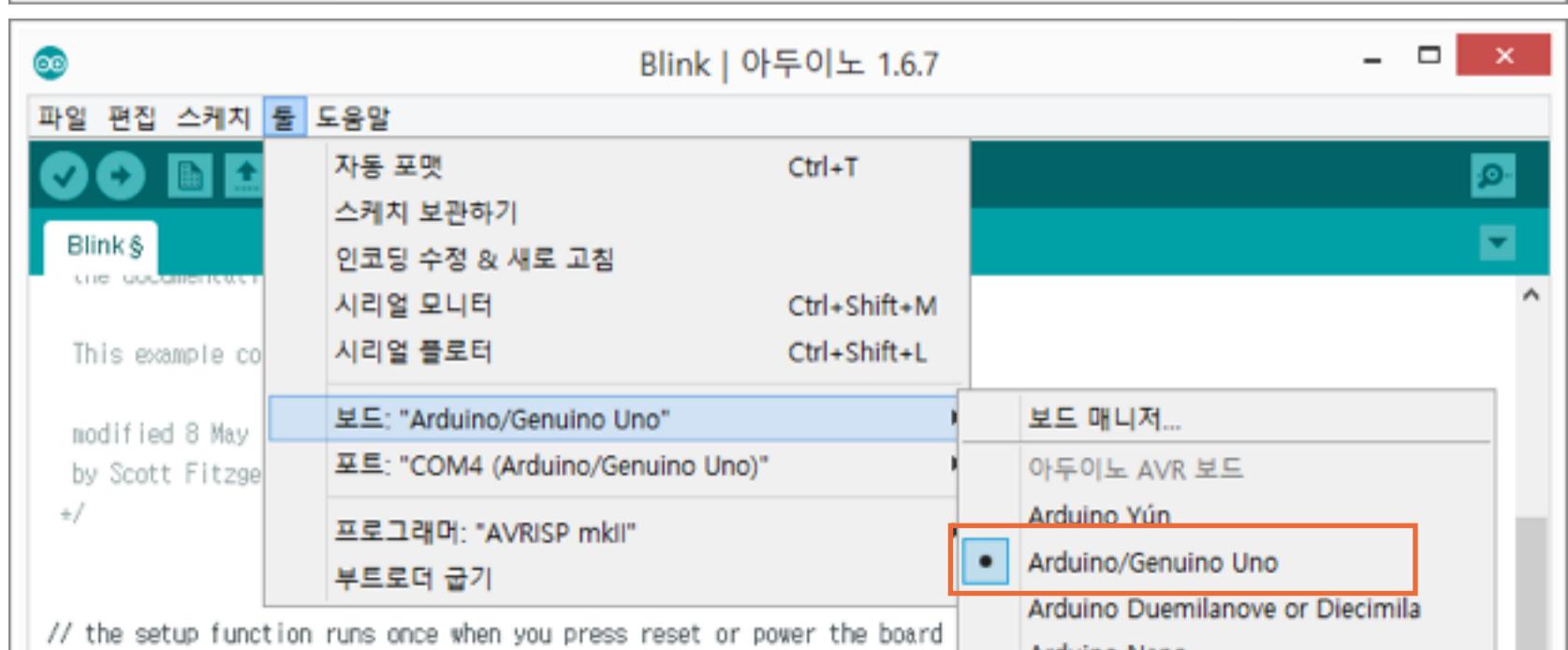
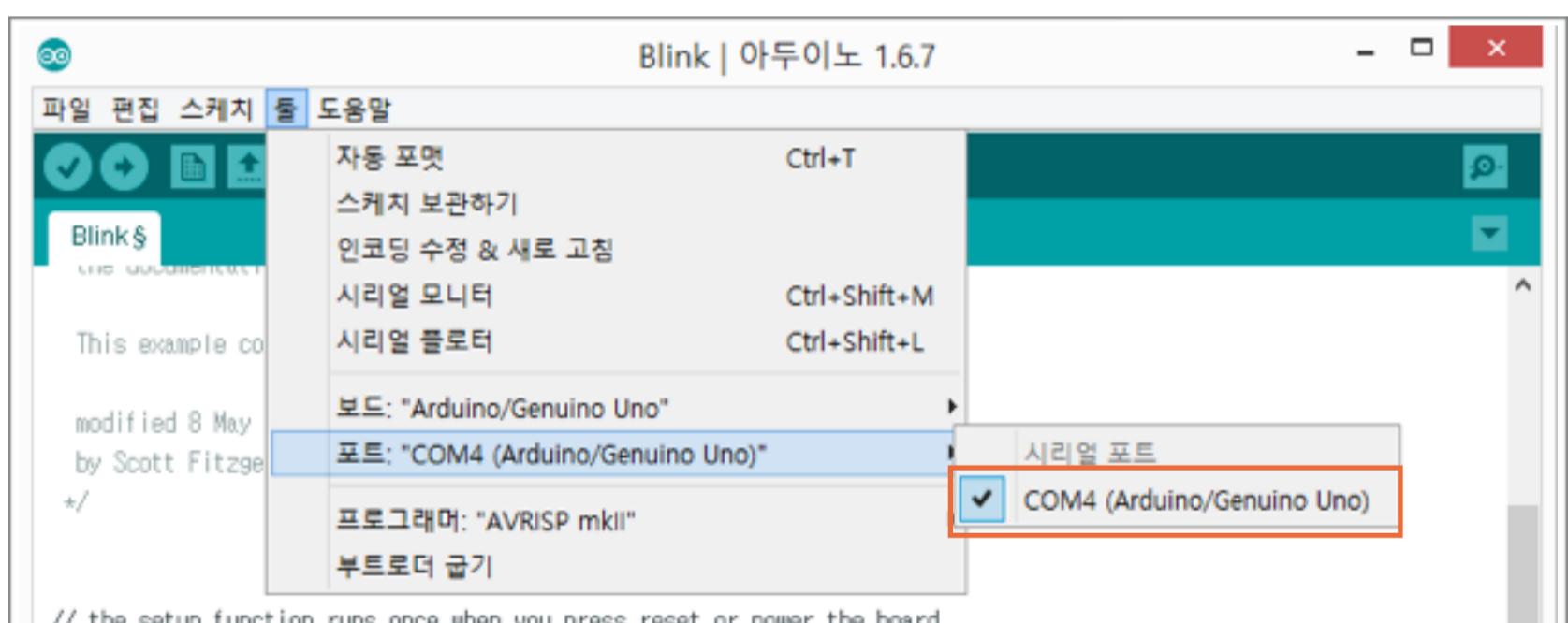
The screenshot shows the 'Download the Arduino Software' page. At the top, there's a large image of the Arduino logo (a circle with a minus and plus sign) and the text 'ARDUINO 1.6.7'. Below it, a paragraph describes the software as open-source and suitable for Windows, Mac OS X, and Linux. To the right, there are download links for Windows (Windows Installer and ZIP file), Mac OS X (10.7 Lion or newer), and Linux (32-bit and 64-bit). Below these are links for 'Release Notes', 'Source Code', and 'Checksums'. Underneath, there are two sections: 'ARDUINO SOFTWARE HOURLY BUILDS' (last updated 5 January 2016) and 'ARDUINO 1.0.6 / 1.5.x / 1.6.x PREVIOUS RELEASES' (last updated 5 January 2016).

The screenshot shows a Windows file explorer window titled 'Arduino'. The path 'C:\Program Files (x86)\Arduino' is selected. The contents of the 'Arduino' folder are listed in a table:

이름	수정한 날짜	유형	크기
dist	2015-06-05 오후...	파일 폴더	
drivers	2015-06-05 오후...	파일 폴더	
examples	2015-06-05 오후...	파일 폴더	
hardware	2015-06-05 오후...	파일 폴더	
java	2015-06-05 오후...	파일 폴더	
lib	2015-06-05 오후...	파일 폴더	
libraries	2015-06-05 오후...	파일 폴더	
reference	2015-06-05 오후...	파일 폴더	
tools	2015-06-05 오후...	파일 폴더	
arduino	2015-05-07 오전...	응용 프로그램	392KB
arduino.l4j	2015-05-07 오전...	구성 설정	1KB
arduino_debug	2015-05-07 오전...	응용 프로그램	389KB
arduino_debug.l4j	2015-05-07 오전...	구성 설정	1KB
libusb0.dll	2015-05-07 오전...	응용 프로그램 확장	43KB
msvcp100.dll	2015-05-07 오전...	응용 프로그램 확장	412KB
msvcr100.dll	2015-05-07 오전...	응용 프로그램 확장	753KB
revisions	2015-05-07 오전...	텍스트 문서	63KB
uninstall	2015-06-05 오후...	응용 프로그램	402KB

3. 아두이노 세팅

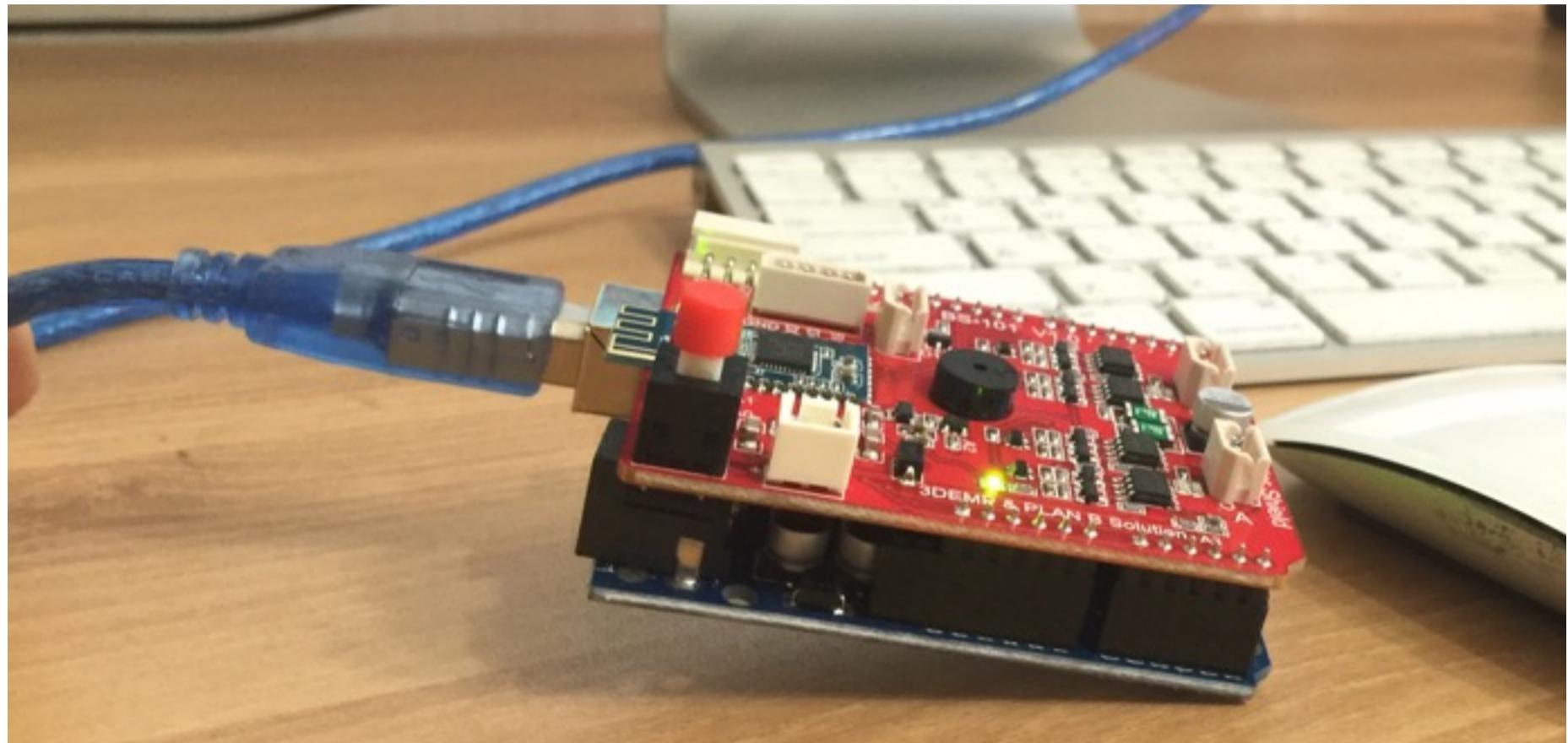
- 아두이노를 컴퓨터와 연결한다.
- 아두이노 소프트웨어를 실행 시킨후 아래와 같이 시리얼 포트와 보드 세팅을 해준다.



4. 블루투스 세팅

- 아두이노 우노 보드 위에 EMP 모터쉴드를 결합한 후에 블루투스 세팅 코드를 업로드 한다.
- 콘솔창을 열어서 텍스트 박스에 'AT'를 친후 엔터를 친다.
- 아래에 OK라는 응답이 나오면 블루투스 연결 완료.
- **블루투스 이름 세팅 AT+NAME**** (예:AT+NAME3demp)**

AT+NAME는 블루투스 이름을 세팅하는 명령으로 콘솔창의 텍스트 박스에 대문자로 AT+NAME라고 쓴 후에 띄어쓰지 않고 원하는 블루투스명을 정한후 엔터를 친다.(영문12자 이내)



5. Bluetooth Setting Code

```
#include <SoftwareSerial.h>
SoftwareSerial BT_Serial(10,11);

void setup(){
    Serial.begin(9600);
    BT_Serial.begin(9600);
}

void loop(){
    if(BT_Serial.available())
        Serial.write(BT_Serial.read());
    if(Serial.available())
        BT_Serial.write(Serial.read());
}
```

EMP - C201 소스코드

```
#include <SoftwareSerial.h> //블루투스 라이브러리 호출
#include <Servo.h> //서보 라이브러리 호출

Servo servo;
SoftwareSerial BTSerial(10, 11); //블루투스 통신에 사용 할 RX핀과 TX핀을 선언한다.

// 아두이노 핀설정
const int SERVO_PIN = A0; //서보모터 각도 제어용 핀

const int DM_SPEED_PIN = 5; // 스피트 관련 핀
const int DM_DRIVE_PIN = 3; // 전진 핀
const int DM_BACK_PIN = 2; // 후진 핀

const int SERVO_RANGE = 120; // 서보모터 최대각도(총 움직일 수 있는 각도)
const int SERVO_MAX_STEP = 6; // 서보모터 좌우 총 최대단계값
const int SERVO_ONE_STEP_VAL = SERVO_RANGE / SERVO_MAX_STEP;

const int MOTOR_MAX_VAL = 255; // 최고속도 255까지
const int MOTOR_RANGE_VAL = 20; // 전후진 스텝 1당 변동값
const int MOTOR_MIN_VAL = 100; // 전후진 최저속도

int curSteeringValue = 90; // 현재 각도
int curSpeedVal = 0; // 현재 속도값(최초 0)
int curDirection = 1; // 현재 방향(1-전진, 0-후진)

void setup() {
    BTSerial.begin(9600); //블루투스 시리얼 통신속도 선언
    Serial.begin(9600); //시리얼모니터 통신속도 선언

    servo.attach(SERVO_PIN); // 서보모터 핀 설정

    pinMode(DM_SPEED_PIN, OUTPUT);
    pinMode(DM_DRIVE_PIN, OUTPUT);
    pinMode(DM_BACK_PIN, OUTPUT);
    servo.write(curSteeringValue); // 서보모터 중립으로 초기화
}

void loop() {
```

```

char cmd = getCommand();
if (cmd != '\0') {
    setRC(cmd);
}
}

char getCommand() {
    char cmd;
    if (BTSerial.available()) { //블루투스에서 넘어온 값이 있으면
        cmd = (char)BTSerial.read(); //읽어들인 문자(한글자)를 data에 저장
    }

    return cmd;
}

void setRC(char cmd) {
    if (cmd == 'L') {
        if (curSteeringValue > 90 - (SERVO_RANGE / 2)) { // 현재각도가 30도보다 클 때만 움직임
            curSteeringValue -= SERVO_ONE_STEP_VAL;
            servo.write(curSteeringValue);
        }
    } else if (cmd == 'R') {
        if (curSteeringValue < 90 + (SERVO_RANGE / 2)) { // 현재각도가 150보다 작을 때만 움직임
            curSteeringValue += SERVO_ONE_STEP_VAL;
            servo.write(curSteeringValue);
        }
    } else {
        if (cmd == '+') {
            if (curSpeedVal == 0) { // 현재값이 0이면 최저속도로 전진
                curDirection = 1;
                curSpeedVal = MOTOR_MIN_VAL;
            } else if (curDirection == 0 && curSpeedVal <= MOTOR_MIN_VAL) {
// 현재값이 최저속도 이하면서 후진중이면 정지
                curSpeedVal = 0;
            } else { // 기타
                if (curDirection == 1) { // 전진이면 MOTOR_RANGE_VAL만큼 +
                    curSpeedVal += MOTOR_RANGE_VAL;
                } else { // 후진이면 MOTOR_RANGE_VAL만큼 -
                    curSpeedVal -= MOTOR_RANGE_VAL;
                }
            }
        }
    }
}

} else if (cmd == '-') {
    if (curSpeedVal == 0) { // 현재값이 0이면 최저속도로 후진
        curDirection = 0;
        curSpeedVal = MOTOR_MIN_VAL;
    } else if (curDirection == 1 && curSpeedVal <= MOTOR_MIN_VAL) {

```

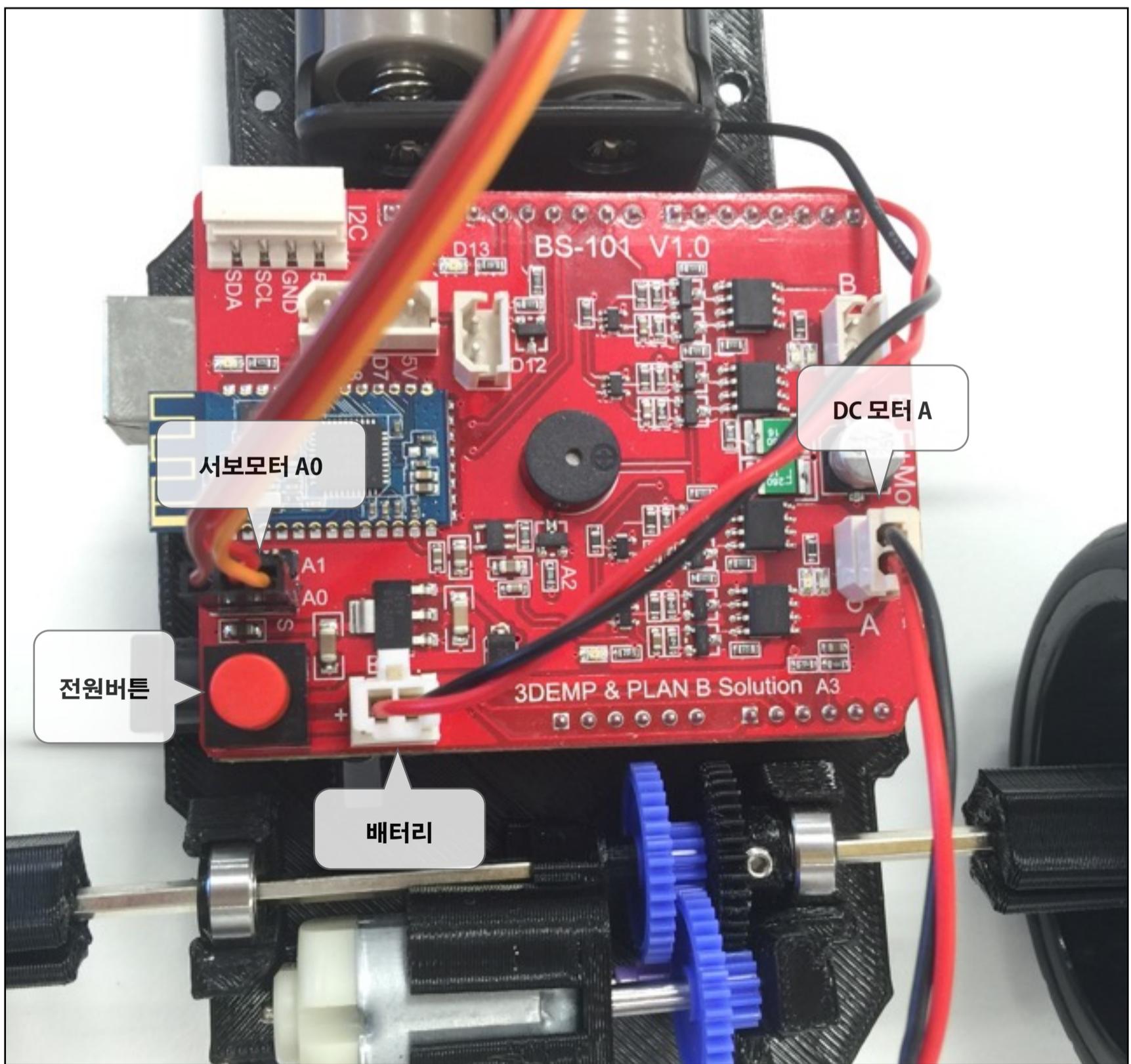
```
// 현재값이 최저속도 이하면서 전진중이면 정지
    curSpeedVal = 0;
} else { // 기타
    if (curDirection == 1) { // 전진이면 MOTOR_RANGE_VAL만큼 -
        curSpeedVal -= MOTOR_RANGE_VAL;
    } else { // 후진이면 MOTOR_RANGE_VAL만큼 +
        curSpeedVal += MOTOR_RANGE_VAL;
    }
}
} else if (cmd == 'S') { // Boost 버튼 - 최대 속도로 올림
    curSpeedVal = MOTOR_MAX_VAL;
    curDirection = 1; // 부스트 기능은 전진으로만 사용
} else if (cmd == 'B') {
    curSpeedVal = 0;
}

if (curSpeedVal > MOTOR_MAX_VAL) { // 최대 속도 이하
    curSpeedVal = MOTOR_MAX_VAL;
}

Serial.println(curSpeedVal);
if (curSpeedVal == 0) { // 정지상태일때
    digitalWrite(DM_DRIVE_PIN, LOW);
    digitalWrite(DM_BACK_PIN, LOW);
    analogWrite(DM_SPEED_PIN, curSpeedVal);
} else {
    if (curDirection == 1) {
        digitalWrite(DM_DRIVE_PIN, HIGH);
        digitalWrite(DM_BACK_PIN, LOW);
    } else {
        digitalWrite(DM_DRIVE_PIN, LOW);
        digitalWrite(DM_BACK_PIN, HIGH);
    }
    analogWrite(DM_SPEED_PIN, curSpeedVal);
}
}
```

6. 핀 연결

- 아두이노에 EMP-C201소스코드를 업로드 한다.
- 아래 이미지와 같이 서보모터, DC모터, 배터리핀을 연결한다.
- 서보모터를 연결할 때는 노란색 선이 안쪽으로 오도록 해서 연결한다.
- 블루색 버튼을 눌러서 전원을 켠다.
- 전원을 켠 후에 바퀴의 중립이 틀어질 경우 서보 가이드 부분을 분해해서 바퀴의 방향을 중립으로 한 후에 다시 조립한다.



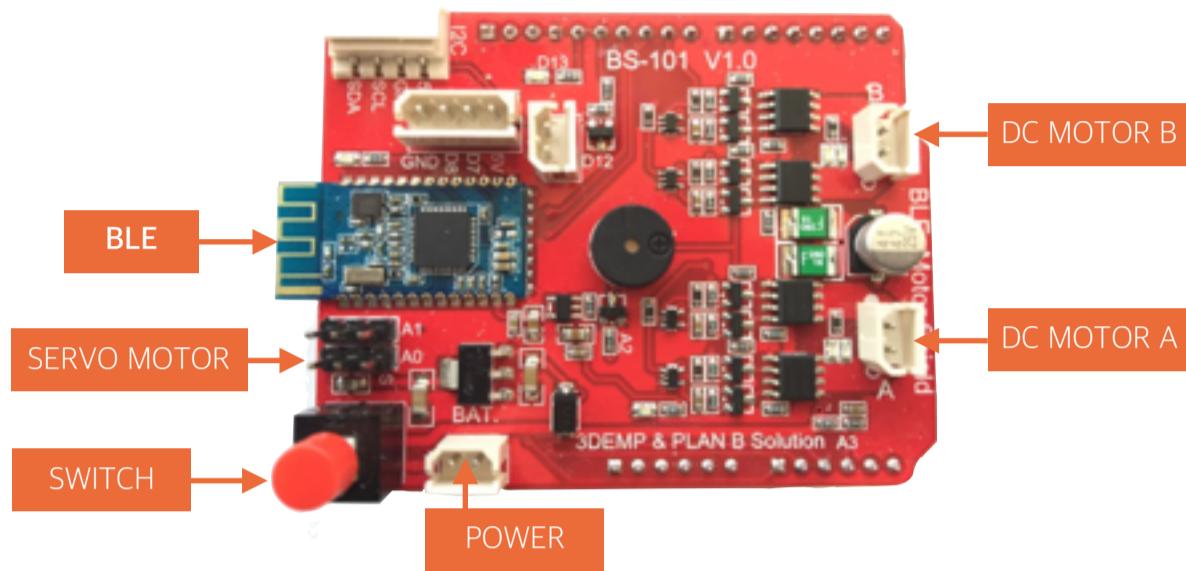
7. 컨트롤 앱 설치 및 세팅

- 앱 스토어에서 3demp라고 검색한후 컨트롤 앱을 스마트폰에 다운 받는다.
- 앱을 실행시킨후 중앙의 블루투스 버튼을 클릭 한다
- SCAN버튼을 클릭해서 내가 이름을 정한 블루투스를 찾아 선택한다.
- 메인화면의 블루투스 마크가 ON상태로 된것을 확인한 후에 가장 왼쪽의 RC CAR컨트롤러 버튼을 누른다.
- 오른쪽 상단의 설정 버튼을 눌러 설정화면으로 들어간후 SAVE버튼을 눌러 키 값을 저장한다.
- 컨트롤 화면으로 돌아와서 RC CAR를 조종해본다.





Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7 ~ 12V
Input Voltage (Limits)	6 ~ 20V
Digital I/O Pins	20
Analog Input Pins	6
DC Current per I/O Pin	40mA
DC Current for 3.3V Pin	50mA
Flash Memory (ATmega328)	32KB
SRAM	2KB
EEPROM	1KB
Clock Speed	12 MHZ
USB Chip	CH340



Bluetooth RX	D10
Bluetooth TX	D11
DC Motor A Speed	D5
DC Motor A Drive	D2
DC Motor A Back	D3
DC Motor B Speed	D6
DC Motor B Drive	D4
DC Motor B Back	D9
Ultrasonic Sensor Echo	D7
Ultrasonic SensorTrig	D8
LED Test	D13
Buzzer	A2
Character LCD	A4, A5
Servo Motor1	A0
Servo Motor2	A1
LED Light	D12
Battery Level	A3