



# 라즈베리 파이를 활용한 RC Car 모델

이지혜 | 이정환 | 성현기



## Contents

### 01 개요

목표 설정

데이터 흐름과 설계

역할 분담 | 단계

### 02 구현

카메라 활용

모터 제어

센서 조정 및 원격 핀제어

### 03 시연

### 04 질의응답

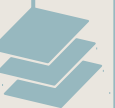


# 01 | 개요

목표

데이터 흐름과 설계

역할 분담 | 단계



# 01 | 개요

목표

데이터 흐름과 설계

역할 분담 | 일정

1

라즈베리 파이를 통한 RC 카 모터 제어

2

스트리밍 기능을 통해 원격으로 주행할 수 있도록 환경 조성

3

주행기능과 주차기능을 분리하여 실행

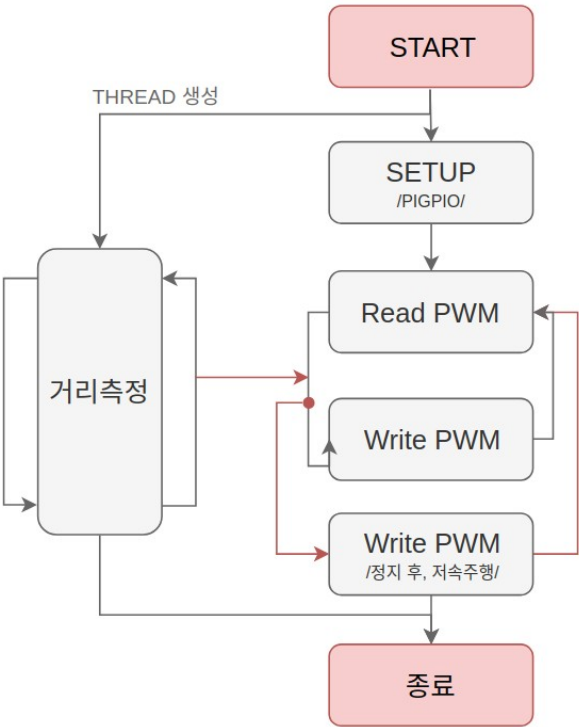
4

그 외 편의 기능

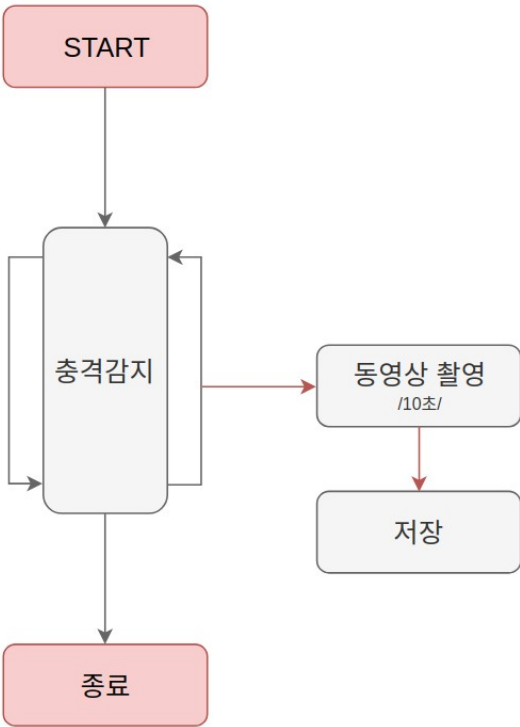
# 01 | 개요

목표      데이터 흐름과 설계      역할 분담 | 일정

## 주행 기능



## 주차 기능



## 웹 스트리밍

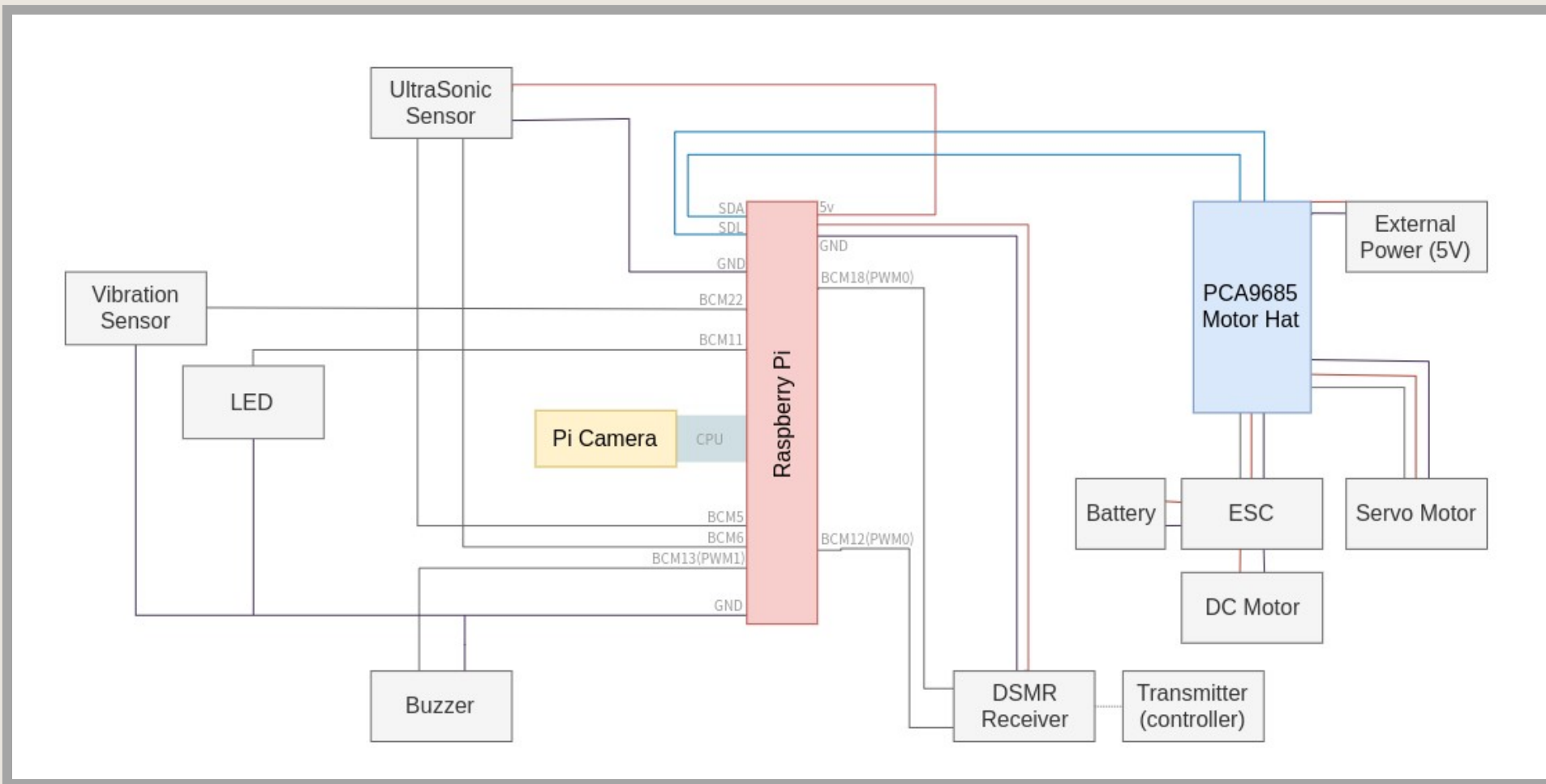


# 01 | 개요

목표

데이터 흐름과 설계

역할 분담 | 일정



# 01 | 개요

목표

데이터 흐름과 설계

역할 분담 | 단계

## 모터 제어 / 이지혜

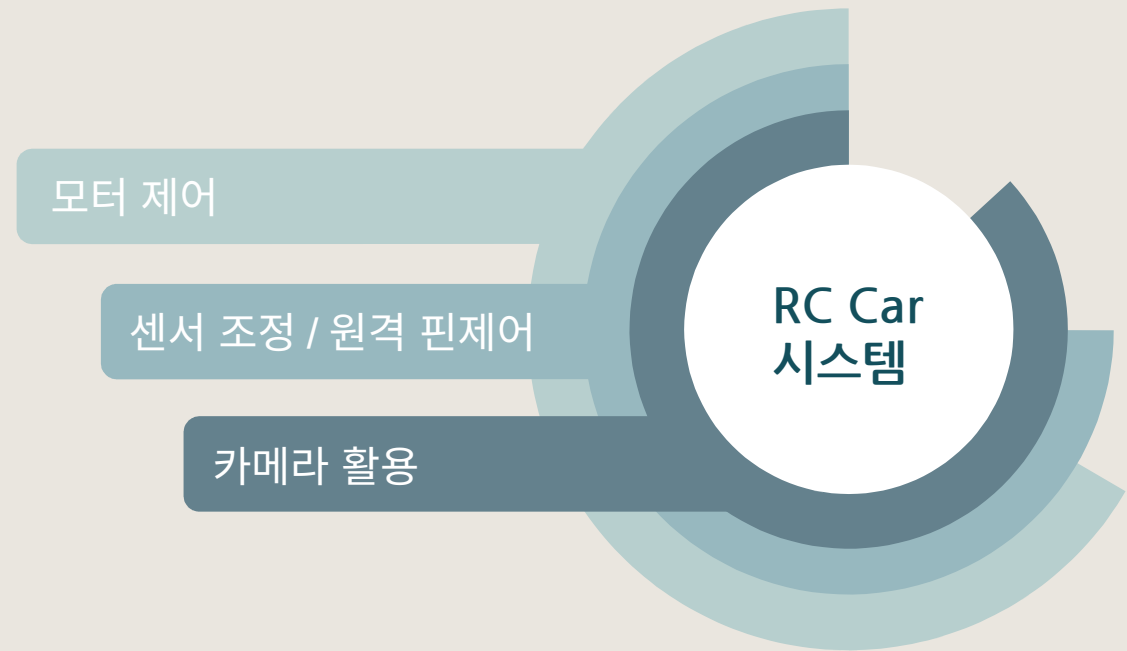
- 주행
- 주행 중 위험대응

## 센서 조정 및 원격 핀 제어 / 이정환

- 거리 측정 및 충격 감지
- 센서 조정 및 테스트

## 카메라 활용 / 성현기

- 영상 스트리밍
- 주차 중 충격대응

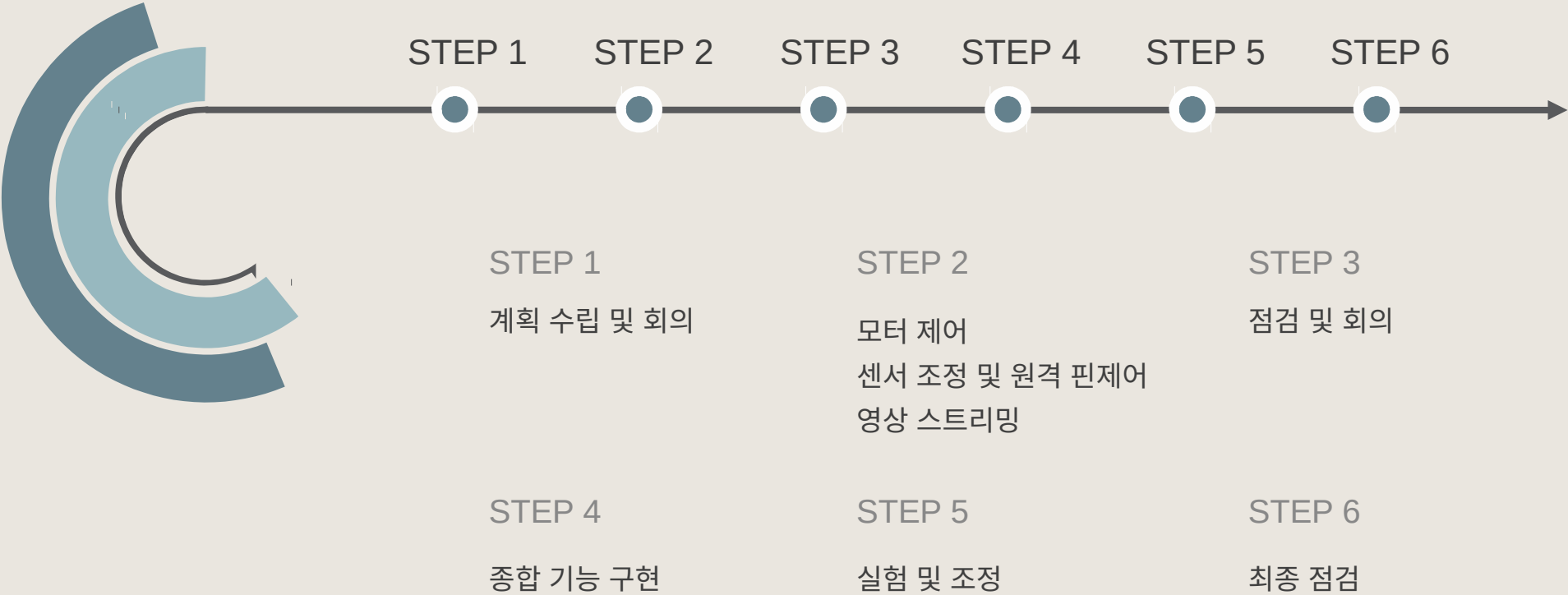


# 01 | 개요

목표

데이터 흐름과 설계

역할 분담 | 단계





## 02 | 구현

카메라 활용

모터 제어

센서 조정 및 원격 핀제어



## 02 | 구현

카메라 활용

모터 제어

센서 조정 및 원격 핀제어

### 실시간 동영상 스트리밍

Motion  
library

Logitech webcam

반응 속도가 느려 현실적으로  
사용 불가



Mjpg  
library

Pi camera

반응 속도 측면에서는 어느 정도 개선  
되었으나 wifi 거리 의존도가 높음



Guvvview  
application

FPV camera

반응 속도나 거리 측면에서 매우  
좋으나 종종 끊김 현상 발생



## 02 | 구현

카메라 활용

모터 제어

센서 조정 및 원격 핀제어

### 블랙박스 동영상 촬영

#### Pi camera



Raspivid API  
활용

Shell script  
작성

충격 감지 코드  
연계

```
#include<stdio.h>
#include<wiringPi.h>
#include<stdlib.h>
#include<time.h>

#define VIV 3
#define BUZZER 23

char* timeToString(struct tm *t);

int main() {
    if (wiringPiSetup() == -1)
        return 1;
    pinMode(VIV, INPUT);      //진동감지센서 INPUT
    pinMode(BUZZER, OUTPUT);  //부저 OUTPUT

    struct tm *t;
    time_t timer;

    //printf("%s\n", timeToString(t));

    while (1) {
        int a = digitalRead(VIV);    // a == 0 진동 o, a == 1 진동 x

        printf("%d\n", a);
        digitalWrite(BUZZER, 0);    // 평소에는 BUZZER을 LOW시켜서 소리가

        if (a == 0) {               // 진동이 올리면

            digitalWrite(BUZZER, 1);    // HIGH시켜서 소리가 나게 함

            timer = time(NULL);    // 현재 시각을 초 단위로 얻기
            t = localtime(&timer); // 초 단위의 시간을 분리하여 구조체에 넣기
            char bash[40];
            sprintf(bash, "%s %s", "sh black.sh", timeToString(t));
            printf("%s\n", timeToString(t));
            printf("%s\n", bash);
            system(bash);           // 카메라촬영하는 쉘 스크립트 실행.
            delay(100);
        }
        delay(10);
    }
}

char* timeToString(struct tm *t) {
    static char s[20];

    sprintf(s, "%04d-%02d-%02d %02d-%02d-%02d", t->tm_year + 1900,
            t->tm_mon + 1, t->tm_mday, t->tm_hour, t->tm_min, t->tm_sec);

    return s;
}
```

## 02 | 구현

## 카메라 활용

## 모터 제어

## 센서 조정 및 원격 핀제어

## 블랙박스 동영상 촬영 / 저장 ( 날짜 포함 )

```
1
1
1
1
1
1
1
1
1
1
0
2019-10-18 16-42-28
sh black.sh 2019-10-18 16-42-28
```

```
pi@raspberrypi188:~/Desktop $ ls
PCA9685.py          buzzer              video2019-10-18_12-52-04.h264
PCA9685.pyc         rc_receiver         video2019-10-18_12-52-58.h264
PCAtest            rc_receiver.c      video2019-10-18_16-42-28.h264
RCcar_RunningMode  read_PWM.py        vidout2019-10-17_17-09-45.mp4
black.sh           read_PWM.pyc       vidout2019-10-17_17-11-44.mp4
blackbox          vidcod.sh
```

```
pi@raspberrypi188: ~/Desktop
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
#!/bin/sh

echo `raspivid -o video$1_$2.h264 -vf -t 10000`

4.0-1
모두
```

```
pi@raspberrypi188: ~/Desktop
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
#!/bin/sh

echo `ffmpeg -f h264 -i video$1.h264 -vcodec copy vidout$1.mp4`

echo `scp vidout$1.mp4 shk@192.168.10.160:/home/shk/다운로드/과제/프로젝트`

~
~
~
~
~
~
~
~
~
~
```

8,19-26 모두

## 02 | 구현

카메라 활용

모터 제어

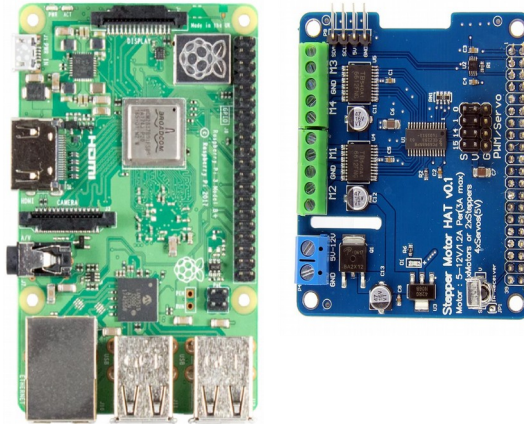
센서 조정 및 원격 핀제어

### 부품 목록 ( 하드웨어 )

Controller/DX5C  
&  
Receiver/SR415



Raspberry Pi 3  
&  
Motor HAT



ESC/Electronic Speed Controller  
DC Motor  
SERVO Motor  
Battery



### pigpio

- GPIO 를 제어하기 위한 라즈베리 파이 라이브러리
- PWM/servo pulse 생성
- GPIO interrupt callback
- simple interface: thread, I2C, SPI...

### pigpio Daemon

- pipe 와 socket interface 의 명령을 받아 백그라운드에서 실행



### Read PWM

- GPIO 에 인터럽트가 발생할 때 마다 , 현재 LEVEL 과 TICK 전달
- LEVEL (HIGH / LOW) -> 시간 측정 (microseconds)
- 폭 (pulse width) 빈도 (frequency)

### Write PWM

- I2C( 확장보드 Motor HAT / PCA9685)

### 전방 장애물 충돌 방지

- 초음파 센서를 통해 전방 거리측정
- 스레드를 사용하여 실시간으로 반영
- EVENT 발생 시 , 정지 ( 경고등 )
- 단시간 저속 주행으로 장애물을 회피 ( 경고등 \_ 노랑 )



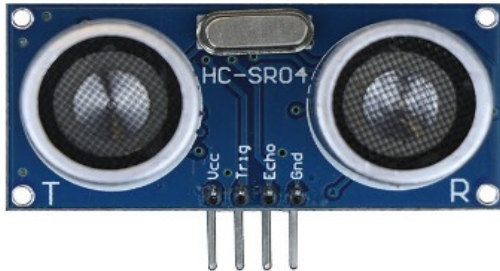
## 02 | 구현

카메라 활용

모터 제어

센서 조정 및 원격 핀제어

### 초음파 센서 / HC-SR04 & 기타 부품



### 오류 발생

Error 1  
비정상 값 출력

Error 2  
불규칙적인 거리

```
111.000000cm
120.000000cm
99.000000cm
91.000000cm
108.000000cm
107.000000cm
83.000000cm
108.000000cm
88.000000cm
108.000000cm
86.000000cm
```

### 정상 작동

Solution 1  
부품 교체

Solution 2  
전압 변경 (3V -> 5V)

```
103.000000cm
104.000000cm
102.000000cm
103.000000cm
105.000000cm
104.000000cm
103.000000cm
103.000000cm
104.000000cm
103.000000cm
```

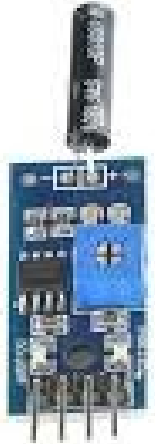
## 02 | 구현

카메라 활용

모터 제어

센서 조정 및 원격 핀제어

### 초음파 센서 /SW-1801P



1  
1  
1  
1  
1  
1  
1  
1  
1  
0  
1  
1  
1  
1  
1  
1  
1  
1  
1



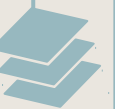
모바일과 라즈베리 파이 ( 하드웨어 ) 를  
연동을 하기 위한 라이브러리

인터넷 (WiFi) 을 통해 통신

목표 : raspberry pi 의 데이터를 앱에서 처리  
Blynk 자체 함수의 문제로 생각됨

```
TX : 5000287
RX : 6973748
34025.000000cm
-----6973845      34025
-----6973881      34025
TX : 5000287
RX : 6973911
34028.000000cm
-----6974007      34028
-----6974038      34028
TX : 5000287
RX : 6974069
34030.000000cm
-----6974163      34030
-----6974196      34030
TX : 5000287
RX : 6974227
34033.000000cm
-----6974321      34033
-----6974356      34033
TX : 5000287
RX : 6974385
34036.000000cm
-----6974478      34036
-----6974510      34036
TX : 5000287
RX : 6974541
34038.000000cm
-----6974664      34038
-----6974698      34038
TX : 5000287
RX : 6974730
34042.000000cm
-----6974824      34042
-----6974857      34042
TX : 5000287
RX : 6974889
34044.000000cm
-----6974986      34044
-----6975018      34044
```

## 03 | 시연





시연

## 04 | 질의응답





질의 응답



THANK YOU