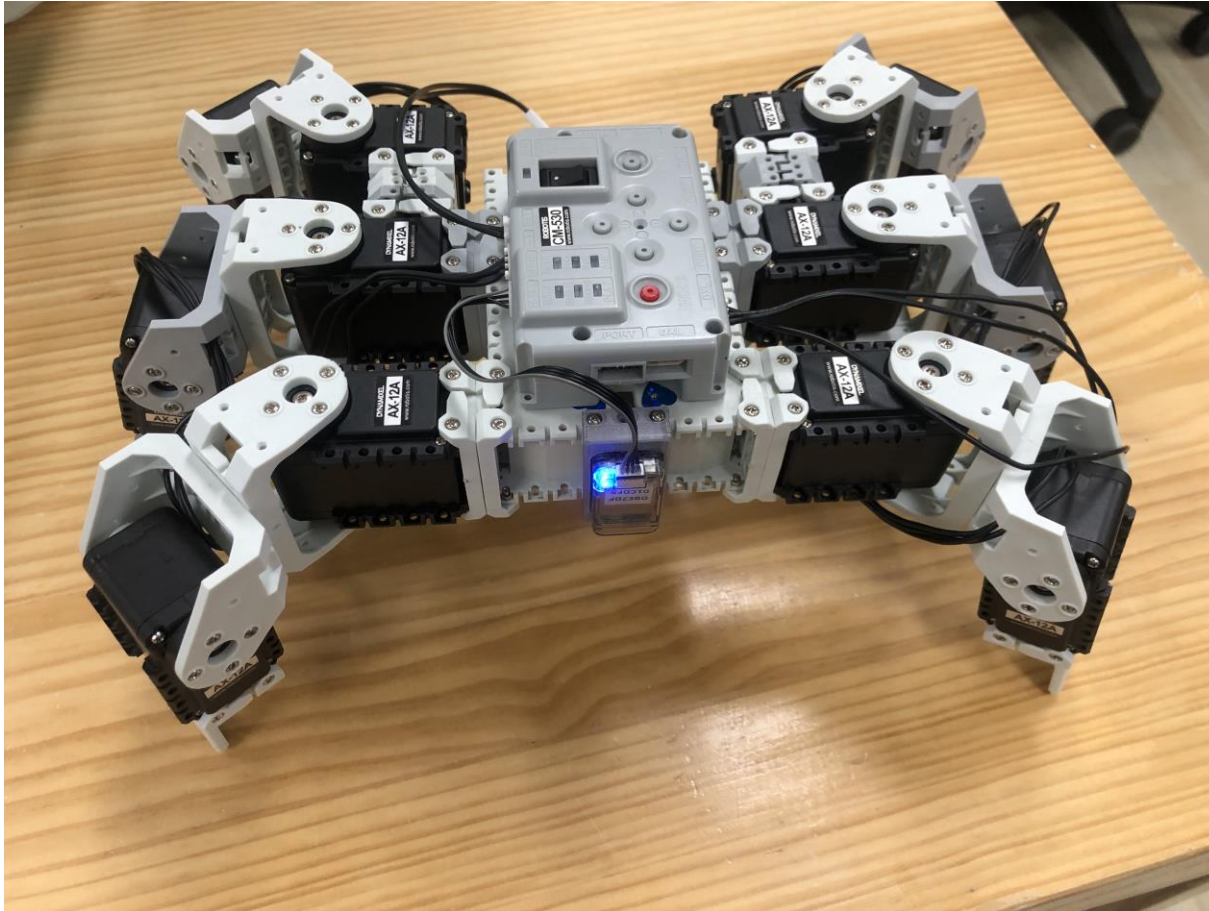


Project 3

Task

최종 목표

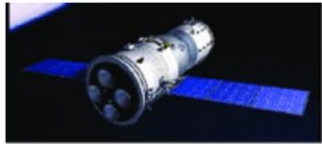
- 육족 로봇으로 모터를 제어하고,
컴퓨터를 통한 블루투스 통신으로 실시간 로봇제어
- 제스처 인식을 통한 실시간 로봇제어



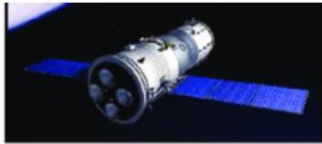
로봇
블루투스
dongle

USB dongle

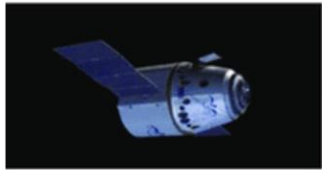
Optical Flow



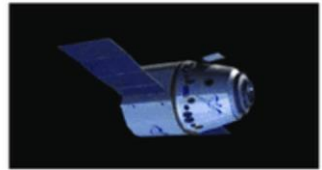
Frame 57_Satellite-1



Frame 59_Satellite-1



Frame 65_Satellite-2



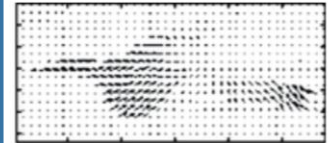
Frame 67_Satellite-2



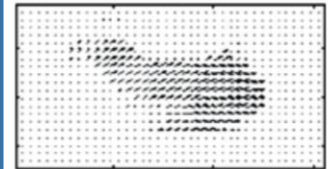
Optical flow map



Optical flow map



Flow vector



Flow vector

- 옵티컬 플로우는 연속된 이미지 내 광류 변화량을 이용하여 움직임 정보를 추정하는 알고리즘.
- 현재 t 시간의 화소가 $t+1$ 시간에 어떤 화소와 매칭이 되는지 찾는 광류 추정 알고리즘을 통해 광류 벡터를 도출하고 움직임을 추정.

Optical Flow - 적용예시



명령 창

Up
Up
Up
Up
Up

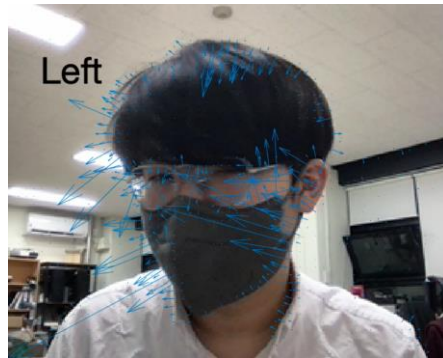
fx Up



명령 창

Down
Down
Down
Down
Down

fx Down



명령 창

Left
Left
Left
Left
Left

fx Left



명령 창

Right
Right
Right
Right
Right

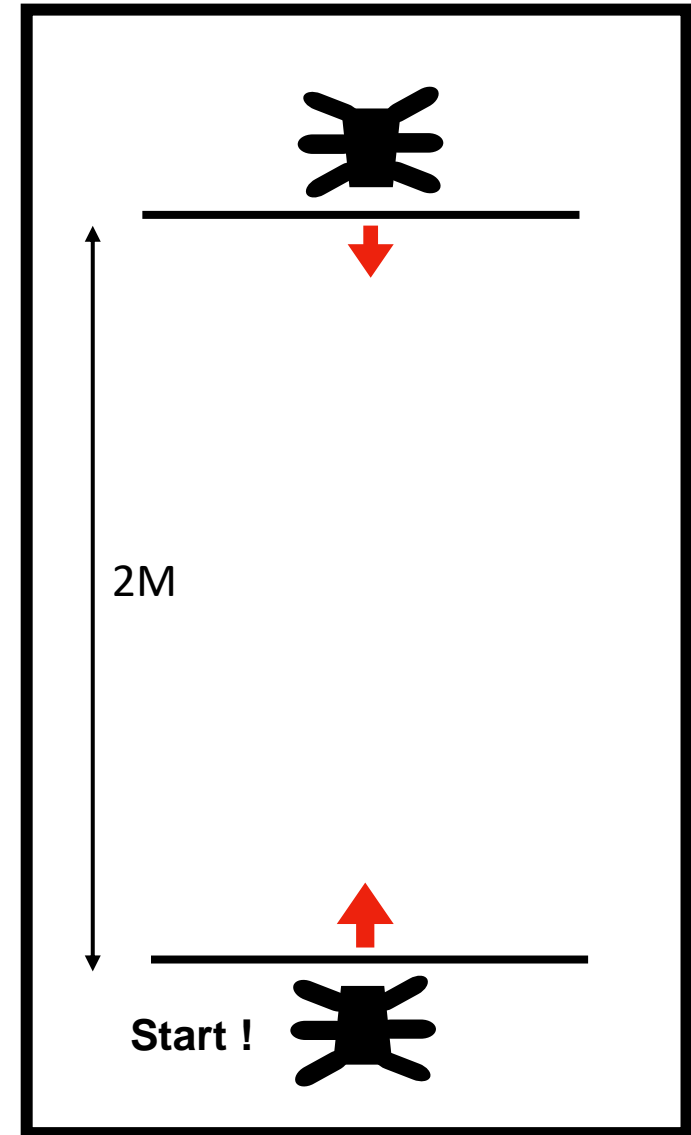
fx Right

- 옵티컬 플로우 벡터값을 활용하여 방향 제스처를 인식할 수 있음.
- 위 예시는 옵티컬 벡터값을 활용한 얼굴 제스처 인식 방향 결과를 창에 텍스트로 나타내고, 명령창에 결과값을 출력함.

Task 1

목표

- 6족 로봇의 달리기 모션을 구현하여, 2M 거리의 구간을 왕복하는 것으로, 왕복지점의 선을 넘어 갔다가 다시 돌아온다.
- 최대한 빠른 시간 내에 2M 거리의 구간을 왕복하는 모션 제어를 구상하고 구현한다.
- 로봇의 발걸음 수를 count하여, 2m 지점을 파악한 뒤 출발점으로 복귀한다. 또는 통신과 matlab을 통해. 학생이 보고 2m 지점에서 돌아올 수 있도록 원격 조종한다.

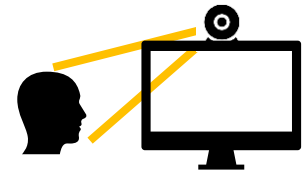


Task 1

평가기준

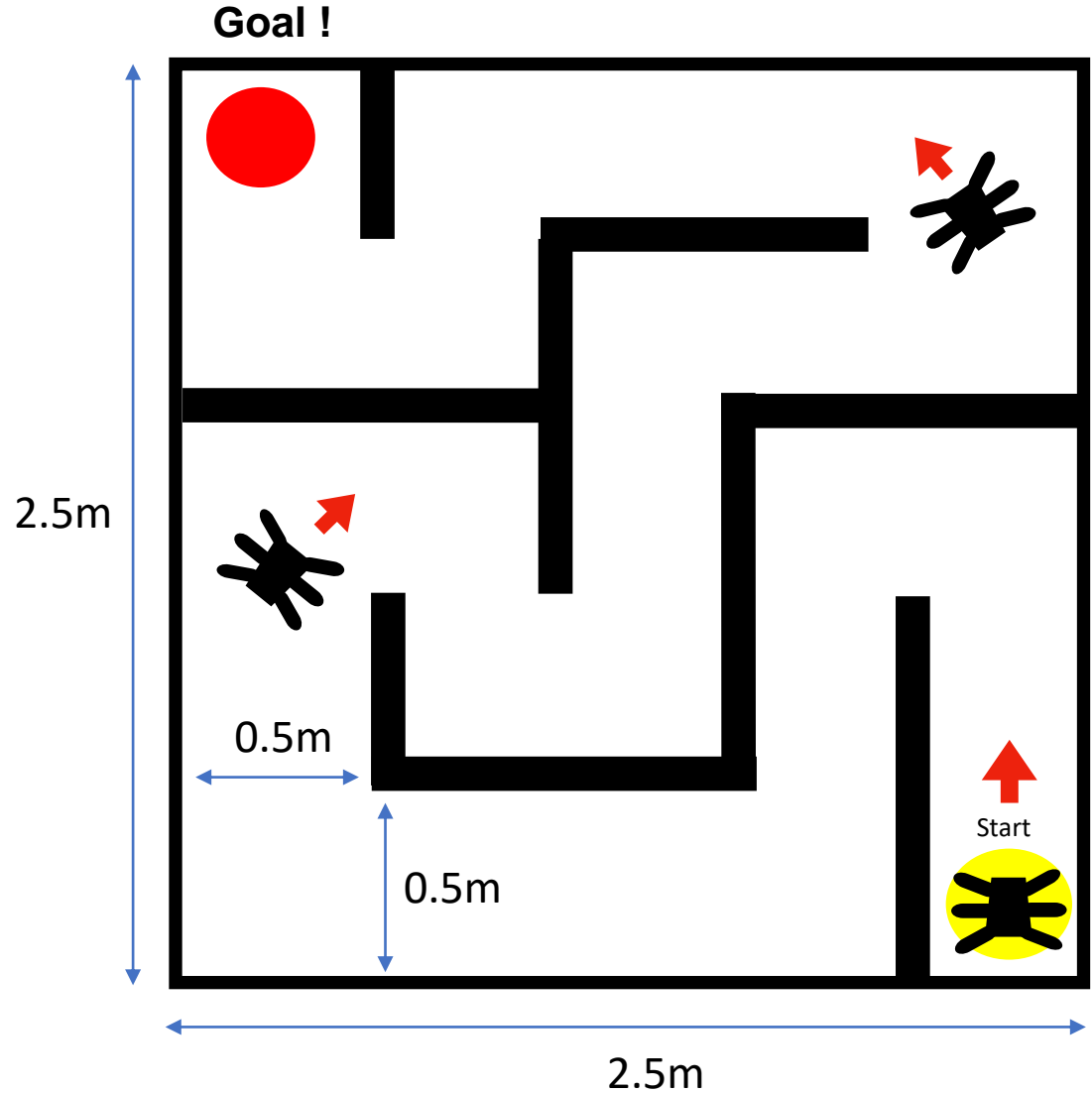
1. 로봇이 2미터 지점 선을 넘은 뒤, 다시 출발선 안으로 돌아왔는가?
2. 얼마나 빠르게 올 수 있는가?
3. 얼마나 효율적으로 로봇이 움직이는가?
4. 적절한 제어 코드는 작성하였는가?
 - 리턴 제어 코드
5. 창의성 (+ 점수)

Task 2



목표

- 모션 제어를 통해 장애물 피하며 목적지에 도달한다.
- 제스처 인식을 통해 로봇에 방향 제어 명령을 내리면서 장애물을 피해 도착지점에 도달하고자 한다.
- 주어진 미로를 잘 빠져나가 도착 지점에 도달하는 것이 목표이다.



Task 2

평가기준

1. 미로의 라인을 벗어나지 않고 골 지점에 도달했는가?
2. 얼마나 빠르게 올 수 있는가?
3. 얼마나 효율적으로 로봇이 움직이는가?
4. 적절한 제어 코드는 작성하였는가?
 - 회전, 제스처 인식, 통신 프로토콜
4. 창의성 (+ 점수)

평가 방법

모든 Task 들은 조교와 함께 아레나에서 결과를 확인한다.

Task 1

2번의 시도가 주어지고 아레나에서 로봇의 왕복시간을 측정한다.
2번의 시도의 평균 시간을 Task1의 소요시간이라고 간주한다.

Task 2

동영상 촬영과 함께 로봇이 미로를 잘 빠져나가는 지 확인한다.

Task 자료

- 로봇 제어 예시 C 코드, matlab를 제공할 예정

```
dxl_initialize( 0, 1 );
bt_initialize(0);
USART_Configuration(USART_PC, 57600);
mDelay(2000);
int tog = 0;

GPIO_ResetBits(PORT_LED_POWER, PIN_LED_POWER);
//GPIO_SetBits(PORT_LED_POWER, PIN_LED_POWER);
while(1)
{
    if(tog){
        GPIO_ResetBits(PORT_LED_POWER, PIN_LED_POWER);
        dxl_write_word( 3, P_GOAL_POSITION_L, 412 );
        dxl_write_word( 4, P_GOAL_POSITION_L, 612 );
        dxl_write_word( 7, P_GOAL_POSITION_L, 512 );
        dxl_write_word( 8, P_GOAL_POSITION_L, 512 );
        dxl_write_word( 11, P_GOAL_POSITION_L, 612 );
        dxl_write_word( 12, P_GOAL_POSITION_L, 412 );
        tog = 0;
    }else{
        GPIO_SetBits(PORT_LED_POWER, PIN_LED_POWER);
        dxl_write_word( 3, P_GOAL_POSITION_L, 512 );
        dxl_write_word( 4, P_GOAL_POSITION_L, 512 );
        dxl_write_word( 5, P_GOAL_POSITION_L, 412 );
        dxl_write_word( 6, P_GOAL_POSITION_L, 412 );
        dxl_write_word( 7, P_GOAL_POSITION_L, 512 );
        dxl_write_word( 8, P_GOAL_POSITION_L, 512 );
        dxl_write_word( 9, P_GOAL_POSITION_L, 412 );
        dxl_write_word( 10, P_GOAL_POSITION_L, 412 );
        dxl_write_word( 11, P_GOAL_POSITION_L, 512 );
        dxl_write_word( 12, P_GOAL_POSITION_L, 512 );
        dxl_write_word( 13, P_GOAL_POSITION_L, 412 );
        dxl_write_word( 14, P_GOAL_POSITION_L, 412 );
        tog = 1;
    }
    mDelay(2000);
}
return 0;
```

제어기 임베디드 C 코드

```
1  clc; clear all; close all;
2  webcamlist
3  cam = webcam('Webcam');
4  cam.Resolution = '1280x720';
5
6  preview(cam)
7
8  delete(instrfindall)
9  %%
10 s = serial('COM4');
11 set(s, 'BaudRate', 9600)
12 fopen(s);
13
14 input = fscanf(s);
15 % ....
16 % ....
17 % ....
18 % ....
19 fprintf(s, "%c", output);
20
```

제어기 블루투스 - 시리얼 통신 Matlab 코드

예시 코드

- 예시에서는 Horn-Shunk(HS) 옵티컬 플로우 알고리즘 적용.
- 간단하게 옵티컬 플로우 계산하는 과정까지에 대한 예시 코드.

```
% 혼-성크 옵티컬 플로우 알고리즘 이용
opticFlow = opticalFlowHS;

count = 0;

nFrames = 0;
while 1

    % 이미지 프레임 획득
    frameRGB = vidDevice();

    % 프레임으로부터 옵티컬 플로우 계산
    flow = estimateFlow(opticFlow, rgb2gray(frameRGB));

    % 연산량으로 인해 10프레임마다 화면 출력
    if rem(count,10) == 0
        imshow(frameRGB)
        hold on
        plot(flow, 'DecimationFactor', [5 5], 'ScaleFactor', 275)

        %%
        %% 화면 텍스트 표시 관련 코드 내용
        %%

    end

    %%
    %% 옵티컬 플로우 활용 제스처 방향 인식 코드 내용
    %%

    % 카운트 관련 변수
    count = count + 1;
    nFrames = nFrames + 1;

    hold off
end
```