

2022

전기전자심화설계 및 소프트웨어실습 프로젝트

[ 프로젝트명: Adalm Pluto 를 활용한 인생네컷 ]

2022. 12

2022. 12. 12

6 조

(학번: 201710851 성명: 박중선 )

(학번: 201810845 성명: 박종혁 )

## 1. 프로젝트명

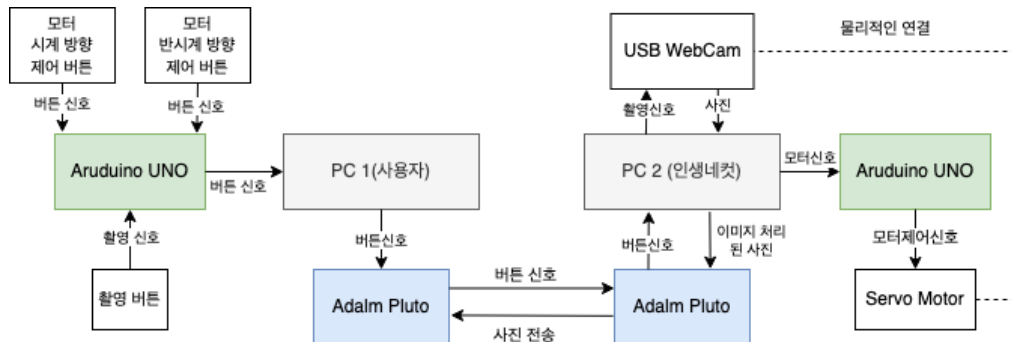
- Adalm Pluto 를 활용한 인생네컷

## 2. 프로젝트의 목표 및 해결하고자 하는 사항

- ADALM PLUTO 와 Arduino 를 이용한 무선 카메라 제어  
카메라의 각도 조절과 카메라 촬영 명령을 버튼으로 제어하고, 이를 Adalm Pluto 를 통해 송수신하기 위해 신호를 encoding 및 양자화한다. 아두이노 보드 및 webcam 을 MATLAB 에서 제공하는 툴박스를 활용해 제어한다.
- 디지털 변조를 이용하여 더욱 선명한 이미지의 전송 및 각 디지털 변조 기법에 따른 퍼포먼스 비교  
수업 시간 동안 다뤄본 여러 디지털 변조 방식에 따라 퍼포먼스의 비교 및 분석 그리고 더욱 선명한 이미지 전송을 위한 변조방법을 찾는다.
- 이미지 프로세싱을 통해, 이미지 보정 및 품질 향상  
이미지 수집과정에서 MATLAB 에서 제공하는 Image Processing Toolbox 을 사용하여, 디지털 이미지를 목적에 맞게 재설정하고 수집한다.

### 3. 프로젝트 수행 내용

#### - 프로젝트의 전체적인 구성도



#### - SW 주요 내용

##### ▶ User 측 아두이노 디지털 인풋에 따른 신호 전송

본 프로젝트에서 'User'측 신호 전송 관련 SW 기능은 다음과 같다. 매트랩 프로그래밍을 통해 푸시버튼에 연결된 아두이노 Digital input 을 읽고, 각각의 푸시버튼 입력에 따라 서로 다른 주파수의 정현파를 생성한 후 전송한다.

버튼이 눌리지 않은 상태에서는 고정으로 100KHz 의 정현파를 전송하고, 버튼이 눌리면, 연결된 아두이노의 디지털입력을 읽어서, 각 버튼에 할당된 주파수의 정현파를 전송한다. 이 과정은 촬영 버튼이 4 번이 입력될 때까지 반복하여, '인생네켓' 측에서 4 번의 사진촬영을 실행하게 한다.

```
% State (flag) %
IDLE = true; CAPTURE = false; UP = false; DOWN = false;
% Freq. %
idle = 100e3; selfi = 110e3; up = 120e3; down = 130e3;
% No. of Captures %
i = 1;
% Sine wave %
fs = 10e6; % sampling freq.
sw = dsp.SineWave;
sw_setting(sw, idle, 1000);

% Transmit %
fc = 830e6; % 반송파 주파수
tx = sdrtx('Pluto');
tx.CenterFrequency = fc;
tx.BasebandSampleRate = fs;
tx.Gain = 0;
txdata = sw();
```

- 초기 세팅
- 각각의 버튼 입력에 따라 제어하기 위한 state를 초기화한다.
- IDLE 상태일때 항상 전송하는 정현파를 정의해준다.
- 반송파 주파수를 830e6으로 설정한다.

```

while (i ~= 5)
    transmitRepeat(tx,txdata);

    % Arduino Button input %
    x = readDigitalPin(a, 'D2');
    y = readDigitalPin(a, 'D6');
    z = readDigitalPin(a, 'D9');

    % State setting %
    if x
        CAPTURE = true;
        IDLE = false;
    elseif y
        UP = true;
        IDLE = false;
    elseif z
        DOWN = true;
        IDLE = false;
    end

    if CAPTURE && ~IDLE
        release(sw);
        sw = dsp.SineWave;
        sw_setting(sw, selfi, 1000);
        txdata = sw();
        transmitRepeat(tx,txdata);
        pause(1);
        CAPTURE = false;
        IDLE = true;
        i = i + 1;
    end

    if UP && ~IDLE
        release(sw);
        sw = dsp.SineWave;
        sw_setting(sw, up, 1000);
        txdata = sw();
        transmitRepeat(tx,txdata);
        pause(1);
        UP = false;
        IDLE = true;
    end

    if DOWN && ~IDLE
        release(sw);
        sw = dsp.SineWave;
        sw_setting(sw, down, 1000);
        txdata = sw();
        transmitRepeat(tx,txdata);
        pause(1);
        DOWN = false;
        IDLE = true;
    end
end

```

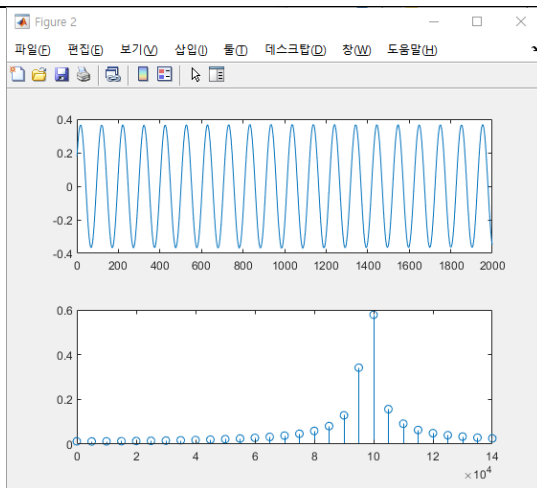
- 디지털 핀 D2, D6, D9 을 변수 x, y, z에 할당
- if ~elseif문을 활용하여 푸쉬 버튼 입력에 따라 3가지로 state 변화
- 3가지 state에 따라 해당되는 명령 수행
  - a) CAPTURE
    - 정현파의 주파수를 110KHz으로 바꿔주고, pause 함수를 활용해 1초 동안만 해당 정현파를 전송한다.
    - i 를 증가시켜 사진을 한번 촬영 횟수를 기록한다.
  - b) UP
    - 정현파의 주파수를 120KHz으로 바꿔주고, 1초간 전송한다.
  - c) DOWN
    - 정현파의 주파수를 130KHz으로 바꿔주고, 1초간 전송한다.

▶ '인생네켓'측 신호 수신 및 webcam, 모터 제어

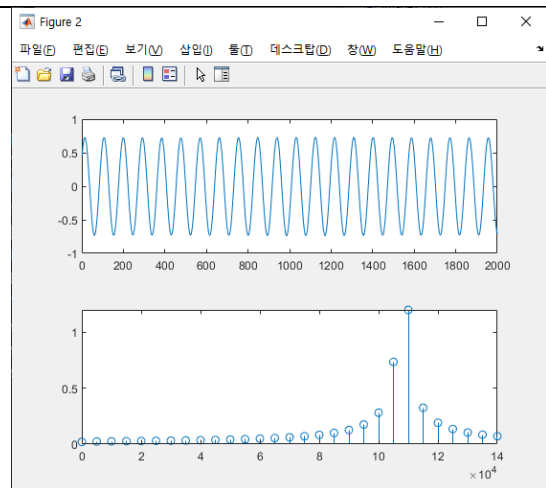
'User'측에서 보낸 신호가 수신되면, 신호를 FFT 변환하고, 가장 크기가 큰 신호의 인덱스를 센싱하여, 수신한 신호의 주파수에 따라 스텝모터 및 webcam 동작을 제어한다.

```
%% Receive settings %%
rx = sdrx('Pluto');
rx.CenterFrequency = fc;
rx.BasebandSampleRate = fs;
rx.OutputDataType = 'double';
rx.SamplesPerFrame = 1000*2;    % txdata.SamplesPerFrame의 2배
rx.GainSource = 'Manual';
rx.Gain = 25;

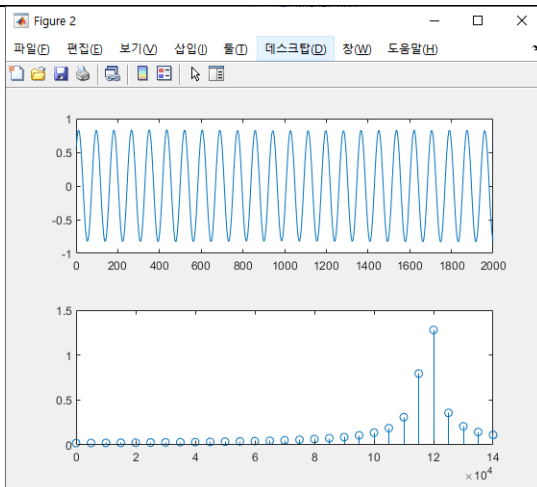
while (i ~= 5) % 사진 4번 찍으면 종료
    rxdata = capture(rx, rx.SamplesPerFrame, 'Samples');
```



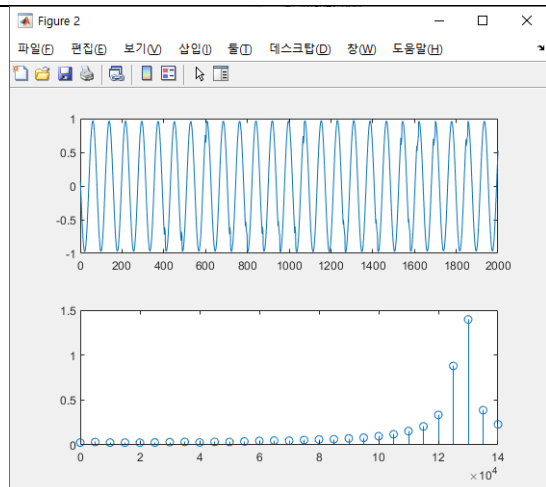
<default 신호 수신 : 100 KHz>



<capture 신호 수신 : 110 KHz>



<각도 up 신호 수신 : 120 KHz >



<각도 down 신호 수신 : 130 KHz >

```

% FFT %
N = rx.SamplesPerFrame;    % samples per frame
n = -N/2 : N/2-1;
f = fs * n/N;
Ft = fftshift(fft(rxdata));

nexttile
stem(f, 2*abs(Ft)/N)
xlim([0 140e3])

len = length(Ft);
Ftt = Ft((len/2 + 2 : len), 1);
[Max, index] = max(Ftt);
index = index + len/2 + 1;
freq = f(index);    % rxdata의 주파수
fprintf("rxdata freq : %d\n", freq)
pause(1)

```

```

switch freq
case selfi
    image = snapshot(mycam);
    file_name = sprintf('Image%d.png', i);
    imwrite(image, file_name, 'png');
    i = i + 1;
    %pause(2);
    %imshow(my_image);
case up
    for angle = 0.5+0.07*(theta-3) : 0.01 : 0.5+0.07*(theta-2) % 약 10 degrees up
        writePosition(s, angle);
    end
    theta = theta+1;
case down
    for angle = 0.5+0.07*(theta-3) : -0.01 : 0.5+0.07*(theta-4) % 약 10 degrees down
        writePosition(s, angle);
    end
    theta = theta-1;
otherwise
    continue;
end

```

수신된 신호에 따라 촬영, 카메라각도 조절을 하기 위한 모터의 각도와 방향을 제어한다. Up, Down 버튼의 신호가 수신되면, 지정된 각도로부터 10° 씩 각도를 움직인다. 촬영 신호가 4 번이 수신되면 반복문을 탈출하여, 이미지 처리 및 전송 과정에 들어간다.

## ▶ '인생네컷'측 이미지 처리

4 번의 사진촬영이 완료되면, 이미지의 배경, 대비 조정, 조도 조정 을 거친 후 4 개의 이미지를 하나로 묶어 이미지 파일을 저장한다.

```
% background generate %
s0 = [70 2040]; bg1 = repmat(permute(uint8([13 97 62])),[1 3 2]),s0); % uint8
s0 = [1080 60]; bg2_1 = repmat(permute(uint8([13 97 62])),[1 3 2]),s0);
s0 = [40 2040]; bg3 = repmat(permute(uint8([13 97 62])),[1 3 2]),s0);
s0 = [1080 2040]; bg_last = repmat(permute(uint8([13 97 62])),[1 3 2]),s0);

% Image read %
img1 = imread('Image1.png'); img2 = imread('Image2.png'); img3 = imread('Image3.png'); img4 = imread('Image4.png');

% Image connect %
tmp1 = horzcat(bg2_1, img1, bg2_1);
tmp2 = horzcat(bg2_1, img2, bg2_1);
tmp3 = horzcat(bg2_1, img3, bg2_1);
tmp4 = horzcat(bg2_1, img4, bg2_1);

result_img = vertcat(bg1, tmp1, bg3, tmp2, bg3, tmp3, bg3, tmp4, bg_last);

% Image save %
ImName = "Life4cut_ori.png";
ImFileOut = fullfile(pwd, ImName);
imwrite(result_img, ImFileOut);
```

```
%% 이미지 흑백으로 변환 및 흑백 대비 조정 %%
img1_gray = imadjust(im2gray(img1),[0.1 0.8],[]); img2_gray = imadjust(im2gray(img2),[0.1 0.8],[]);
img3_gray = imadjust(im2gray(img3),[0.1 0.8],[]); img4_gray = imadjust(im2gray(img4),[0.1 0.8],[]);
bg1_gray = im2gray(bg1); bg2_1_gray = im2gray(bg2_1); bg3_gray = im2gray(bg3); bg_last_gray = im2gray(bg_last);

% Image connect %
tmp1 = horzcat(bg2_1_gray, img1_gray, bg2_1_gray);
tmp2 = horzcat(bg2_1_gray, img2_gray, bg2_1_gray);
tmp3 = horzcat(bg2_1_gray, img3_gray, bg2_1_gray);
tmp4 = horzcat(bg2_1_gray, img4_gray, bg2_1_gray);

result_img_gray = vertcat(bg1_gray, tmp1, bg3_gray, tmp2, bg3_gray, tmp3, bg3_gray, tmp4, bg_last_gray);

% Image save %
ImName = "Life4cut_gray.png";
ImFileOut = fullfile(pwd, ImName);
imwrite(result_img_gray, ImFileOut);
```

```

%% 저조도 영상 향상 %%
img_array = {img1, img2, img3, img4};
img_imp = {0,0,0,0};
for ii = 1 : 4
    tmp1 = imcomplement(img_array{ii});
    tmp2 = imreducehaze(tmp1, 'Method','approx','ContrastEnhancement','boost');
    img_imp{ii} = imcomplement(tmp2);
end

% Image connect %
tmp1 = horzcat(bg2_1, img_imp{1}, bg2_1);
tmp2 = horzcat(bg2_1, img_imp{2}, bg2_1);
tmp3 = horzcat(bg2_1, img_imp{3}, bg2_1);
tmp4 = horzcat(bg2_1, img_imp{4}, bg2_1);
result_img_imp = vertcat(bg1, tmp1, bg3, tmp2, bg3, tmp3, bg3, tmp4, bg_last);

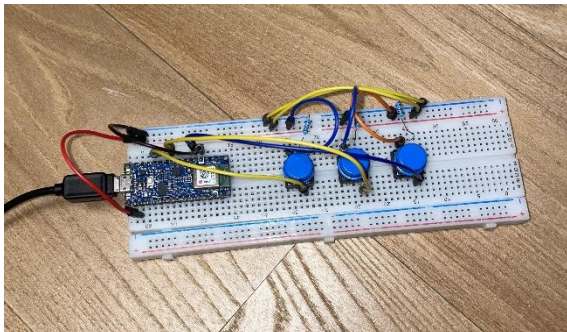
% Image save %
ImName = "Life4cut_imp.png";
ImFileOut = fullfile(pwd, ImName);
imwrite(result_img_imp, ImFileOut);

```

## ▶ 이미지 파일 전송 및 수신

이미지 파일 송신 및 수신 관련 코드는, 이전에 과제에서 다뤘던 'Pluto Image Transmission and Receive' 예제를 활용했다. Scale 과 Gain, 변조방식 등을 바꿔가며, 여러 지표들을 비교하고 확인했고, 가장 선명하고 좋은 퍼포먼스를 보이는 경우를 골랐다.

## - HW 주요 구성



\* Arduino nano, push button 3 개(pull down)



\* Arduino Uno R3, Webcam, servo motor



#### 4. 프로젝트 결과

##### ▶ 전체적인 구성



##### ▶ 3 가지 버전의 인생네컷 제작



\* 인생네컷 -원본

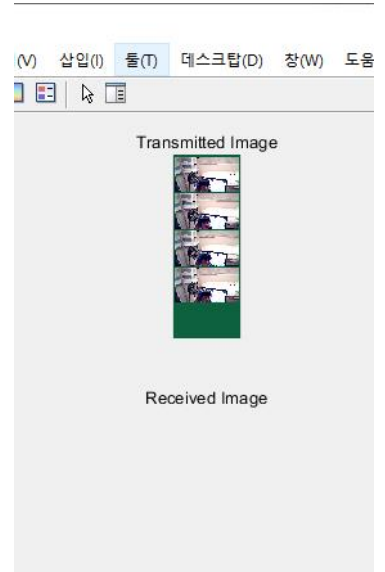
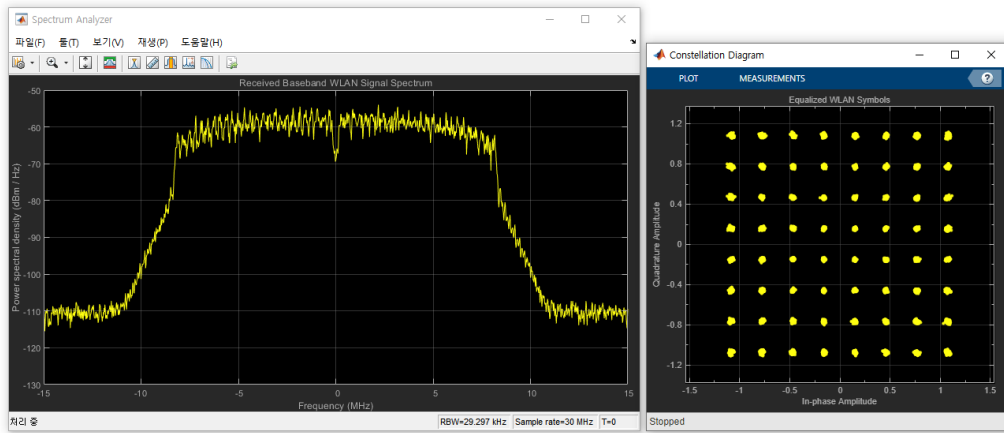


\* 인생네컷 - 저조도 영상향상

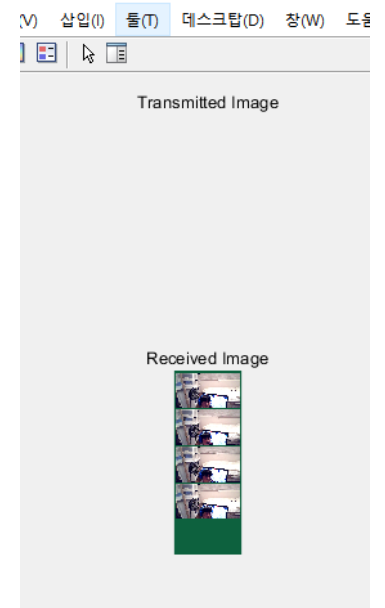
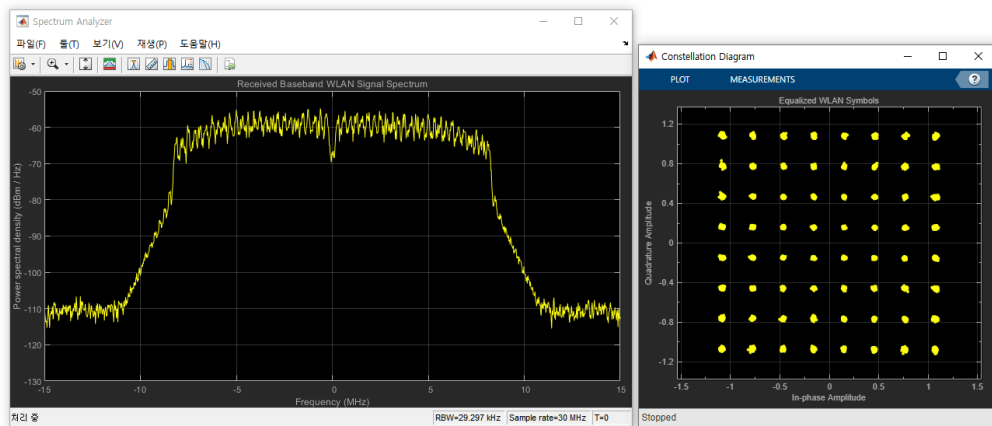


\* 인생네컷 - 흑백 처리

## ▶ 인생네컷 장치에서 User 로 이미지 파일 송신



## ▶ User 이미지 파일 수신



## 5. 프로젝트 결과 활용 방안

기존의 '인생네컷'은 좁은 부스안에서 고정된 카메라로 사진을 촬영하는 경우가 대다수이다. SNS 가 발달하고, 피드에 업로드를 위한, 일명 '감성사진'에 대한 욕구가 늘어나고 있는 MZ 세대들을 타겟팅하여, 고객들에게 '감성사진'을 얻기 용이한 더 넓은 공간에서 무선으로 카메라를 찍는 위치와, 각도에 대해 커스터마이징 기능을 제공하고자 한다.

### - 무선 촬영

RF 을 활용한 무선 촬영이므로 선이 전혀 보이지 않아, 남이 찍어주는 듯한 자연스러운 연출이 가능하다.

### - OpenCV 기능을 접목하여 발전 가능

Person detection 과 Tracking 기능 등 Computer Vision 기능을 추가하여, 자동 보안 시스템, 스마트 홈캠 같이 다양한 분야에도 활용 가능하다.

## 6. 프로젝트 수행하며 느낀 점 및 개선방안

### - 박중선\_201710851

프로젝트를 진행하며, 많은 오류와 시행착오가 있었다. 수신한 신호에 따라, 다른 동작을 제어해야 하는데, 특히 신호의 송수신 과정에서 원하지 않는 성분의 신호가 크게 뜨거나, 노이즈가 끼서 제대로 동작하지 않는 경우가 많았다. 적지 않은 시간을 할애하고, 고민을 통해, 문제를 해결하는 과정에서 배운 점도 많았고, 전공에 더욱 관심과 흥미를 가지게 되는 경험이 되었다. 실시간 무선 보안 시스템을 구현하고자 했던 첫 아이디어를 실현하지 못한 것이 아쉬웠지만, 이번 경험을 바탕으로 다음에는 더욱 원활한 프로젝트 진행을 할 수 있을 것이다.

- 박종혁\_201810845

처음에 어떤 방식으로 명령을 전달할지 결정하는 과정에서 어려움이 있었지만, 우리 프로젝트가 정확한 숫자 값 전달보다는, 올바른 타이밍에 명령을 전달하는 것이 주목적이라는 것을 깨닫고 나서는 수월하게 진행할 수 있었다. 또, 촬영한 이미지를 후 처리하는 부분에 있어서 더 다양한 선택지가 있었지만, 시간 관계상 2 가지 밖에 선택하지 못해 아쉬움이 있었다. 처음에는 막막했지만, 그 동안 실습했던 내용을 차근 차근 떠올리며 생각을 정리하다 보니 원활하게 프로젝트를 진행할 수 있었다.

## 7. 프로젝트 BoM (Bill of Material)

No.	상품명	가격	수량
1	Arduino Uno R3	31000	1
2	Arduino nano 33 ble	45100	1
3	서보 모터 SG90	2400	1
4	HD WEB CAMERA	39000	1

## 8. 프로젝트 수행시 업무 분담

- 박종선\_201710851 : 사전 자료 조사, 아두이노 제어 관련 매트랩 코드 작성, 보고서 초안 작성, 이미지 프로세싱 관련 매트랩 코드 작성, 발표 영상 제작 , 통합 코드 작성
- 박종혁\_201810845 : 사전 자료 조사, 부품 주문, 하드웨어 외형 설계, 디지털 변조 및 무선 통신 관련 코드 작성, 보고서 최종안 작성, 발표 영상 제작, 통합 코드 작성

### [ 참고문헌 ]

<https://kr.mathworks.com/help/images/image-enhancement-and-restoration.html>