



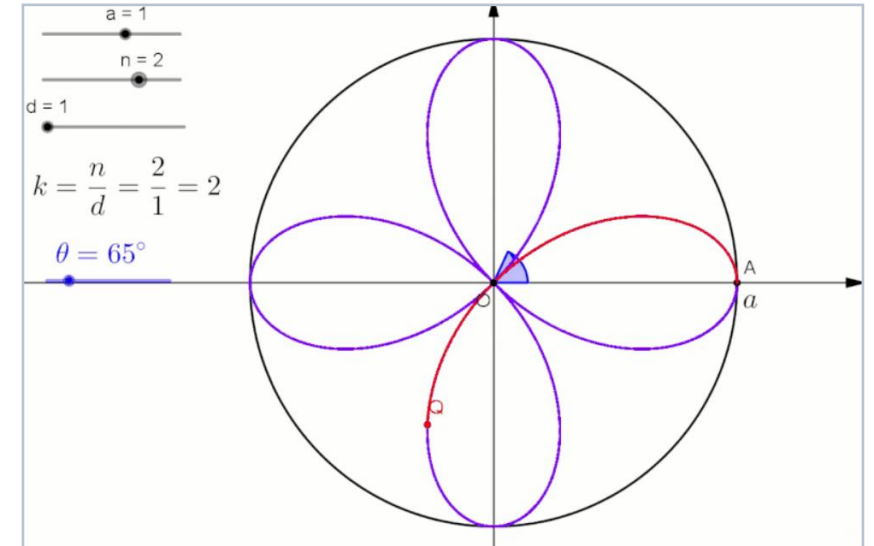
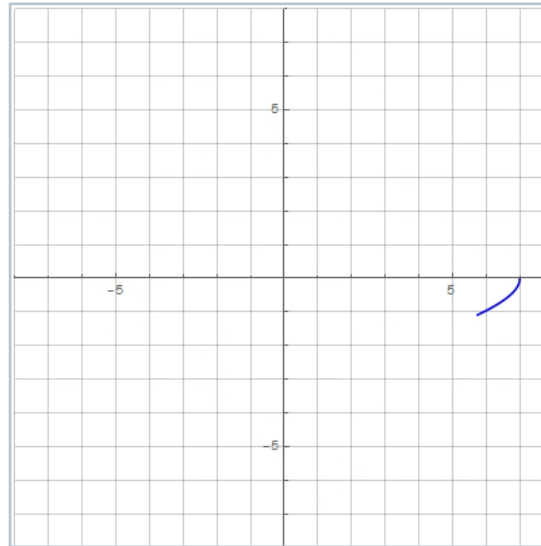
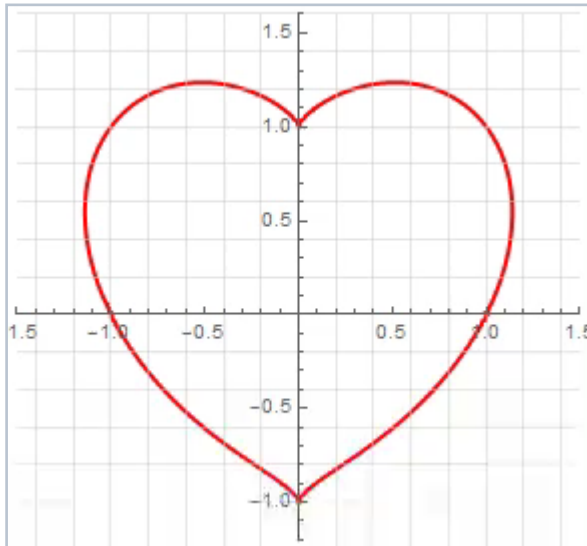
System Design

TP2 Software Quality Evaluation

Goal



SW에서 파라미터 설정으로
HW를 제어해보는 경험



Evaluation table

필수 평가

도형이 이송 가능 영역 내 어디든 이송 가능

도형이 정상적으로 완성된 후 이송 시작

이송 중 현재 궤적 표시

이송 중 새 포인트 및 이송명령 금지

이송 종료 후 포인트 및 궤적 제거

추가 개발 항목

프로그램 시작 시 사용자 정보 입력

프로그램 동작 완료 후 로그파일 작성

시스템 ON/OFF시 영점 위치

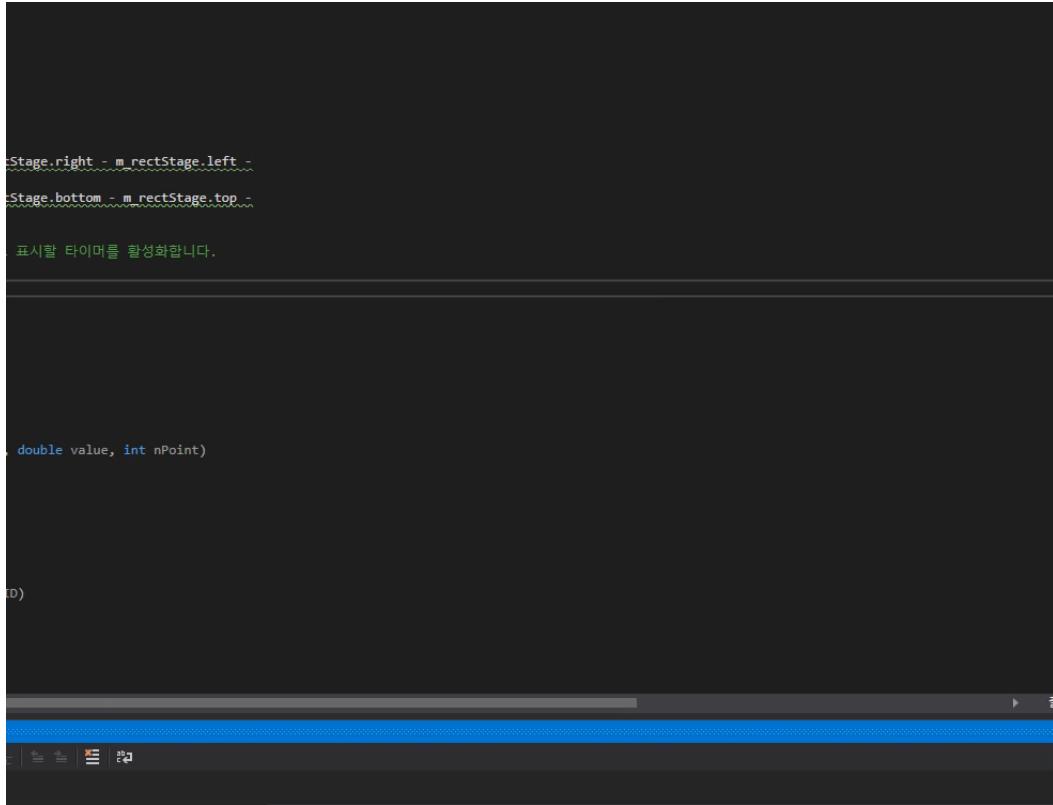
도형 파라미터 설정 기능(크기, 개수, 동작 속도)

도형 크기 변화에 따른 이송 영역 제한

자율 평가

- | | |
|----|----------------------------------|
| 1 | 한 도형 동작 시 다른 도형 동작 불가 |
| 2 | 이송된 결과(PTP, CP, LEAF) 로그파일 구분 저장 |
| 3 | 결과 로그 오류 없이 저장 |
| 4 | 사용자 이름 로그파일 저장 |
| 5 | 시스템 시작 시 원점 이동 |
| 6 | 시스템 종료 시 원점 이동 |
| 7 | 앞의 개수 설정 (50 ~ 150) |
| 8 | 앞의 크기 설정 |
| 9 | 앞 동작 속도 설정(저속, 고속) |
| 10 | 앞의 크기에 따라 제한 범위 유동적 변화 |
| 11 | 앞 동작 중 크기 변경 방지 |
| 12 | 앞 동작 중 개수 변경 방지 |
| 13 | 앞 동작 중 속도 변경 방지 |
| 14 | 임의 동작에 따른 피드백 알림 |
| 15 | 코드 최적화 |

Operation Video



[필수 평가항목]

이송 가능 영역 이송

도형 완성 후 이송

궤적 표시

이송 중 중복 명령 금지

이송 종료 후 화면 초기화

[추가 개발 항목]

프로그램 시작 시 사용자 정보 입력

로그파일 작성

시스템 ON/OFF시 영점 이동

도형 파라미터 설정 기능

크기 변화에 따른 이송 영역 제한

[평가 지표]

한 도형 동작 시 다른 도형 동작 방지

이송된 결과 로그파일 구분 저장

결과 로그 오류 없이 저장

사용자 이름 로그파일 저장

시스템 시작 시 원점 이동

시스템 종료 시 원점 이동

앞의 개수 설정(50 ~ 150)

앞의 크기 설정

앞의 동작 속도 설정(저속, 고속)

앞의 크기에 따라 제한범위 알림

앞 동작 중 크기 변경 방지

앞 동작 중 개수 변경 방지

앞 동작 중 속도 변경 방지

Transfer Process Solution

```
void CMy20220401HMYD1g::draw_rose(CPoint point_x, int sePose)
{
    CPoint point_val;
    int SliderPosY;
    point_val = point_x;
    SliderPosY = sePose;
    if (rose_click_flag == true) { ... }
    else { ... }
    r_point_x = (int)rose_point_X;
    r_point_y = (int)rose_point_Y;
    return;
}
```

장미(잎)을 그려주는 함수

이유 : 잎의 개수, 크기에 따라 작성해야 하는데 각 조건문에 따라 반복적인 프로그래밍을 방지하기 위해서.

draw_rose()

특징 :

1. LbuttonDown으로 부터 받은 클릭 좌표와 실제 위치를 파라미터로 받는다.
2. 계산이 완료된 후 좌표를 메모장에다 저장해준다.

Transfer Process Solution

```
void CMy20220401HMYDlg::OnLButtonDown(UINT nFlags, CPoint point)
{
    if (Set_Cursor == 1)
    {
        AfxMessageBox("표현할 수 없는 위치입니다.");
    }
    else if (Set_Cursor == 2)
    {
        if (!btn_ptp && !btn_cp && !btn_rose) { ... }
        else if (btn_ptp) { ... }
        else if (btn_cp) { ... }
        else if (btn_rose) { ... }
    }
    CDialogEx::OnLButtonDown(nFlags, point);
}

if (leaf_size_num <= 150 && leaf_size_num >= 50)
{
    if (point.x >= rect2.left + leaf_size_num + 5 && point.x <= rect2.right - leaf_size_num)
    {
        if (point.y <= rect2.bottom - leaf_size_num + 5 && point.y >= rect2.top + leaf_size_num + 5)
        {
            //AfxMessageBox("여기");
            draw_rose(point, nSliderEndPosY);
        }
        else
        {
            AfxMessageBox("영역 밖 입니다.");
        }
    }
    else
    {
        AfxMessageBox("영역 밖 입니다.");
    }
}
```

이송 영역 제한 구현

1. Set_Cursor를 통해 클릭영역을 구분
2. btn_ptp, btn_cp, btn_rose 각종 flag를 라디오버튼과 연계하여 그리고자 하는 도형의 크기에 따라 영역을 제한한다.
3. 장미(잎)을 그릴 경우 선택한 사이즈에 가중치 값을 더해주어 구동 영역을 확실하게 정했다.

Solution

극좌표 → 직교좌표 적용

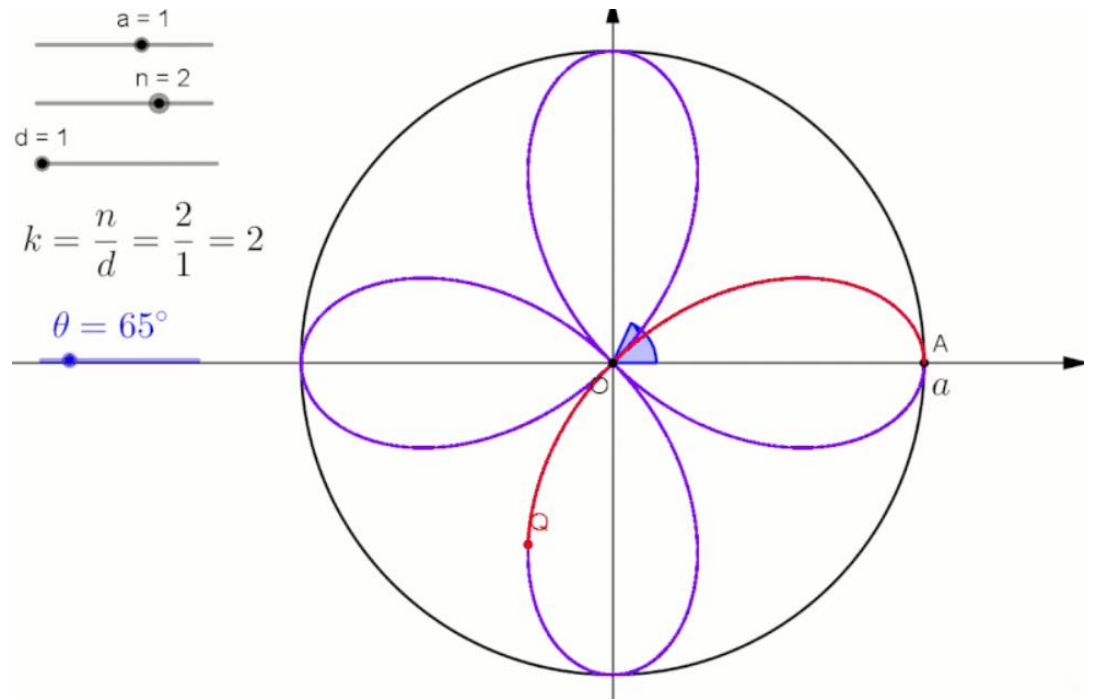
$$r = a \cos k\theta$$

$$\begin{cases} X = r \cos \theta \\ Y = r \sin \theta \end{cases}$$

$$\begin{aligned} \therefore X &= \text{click_point} + \cos(k * \theta) * \cos(\theta), \\ Y &= \text{click_point} + \cos(k * \theta) * \sin(\theta) \end{aligned}$$

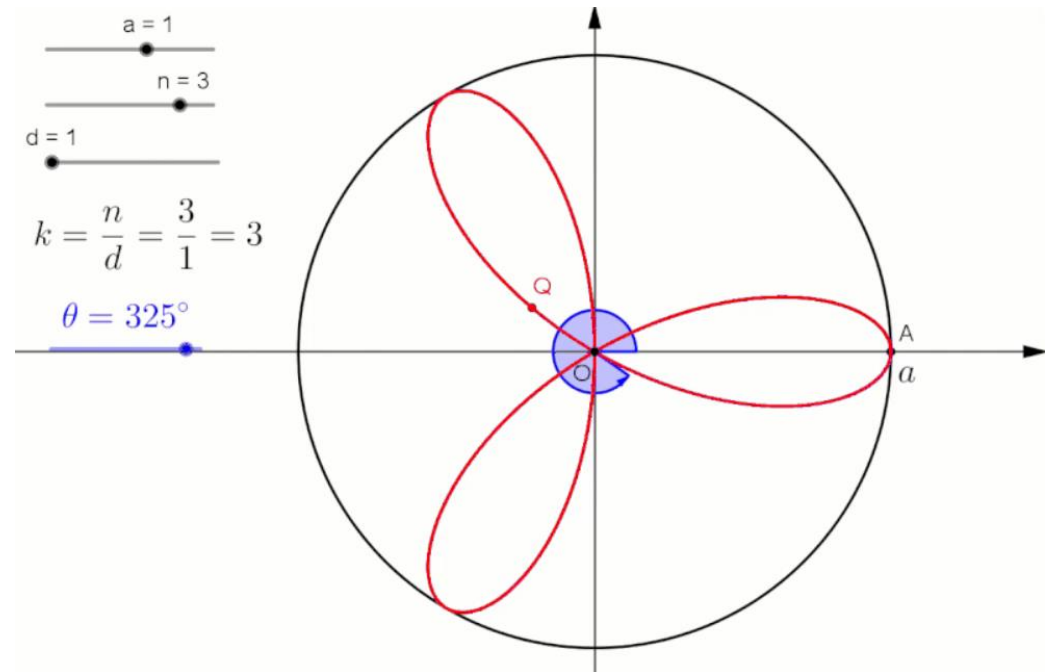
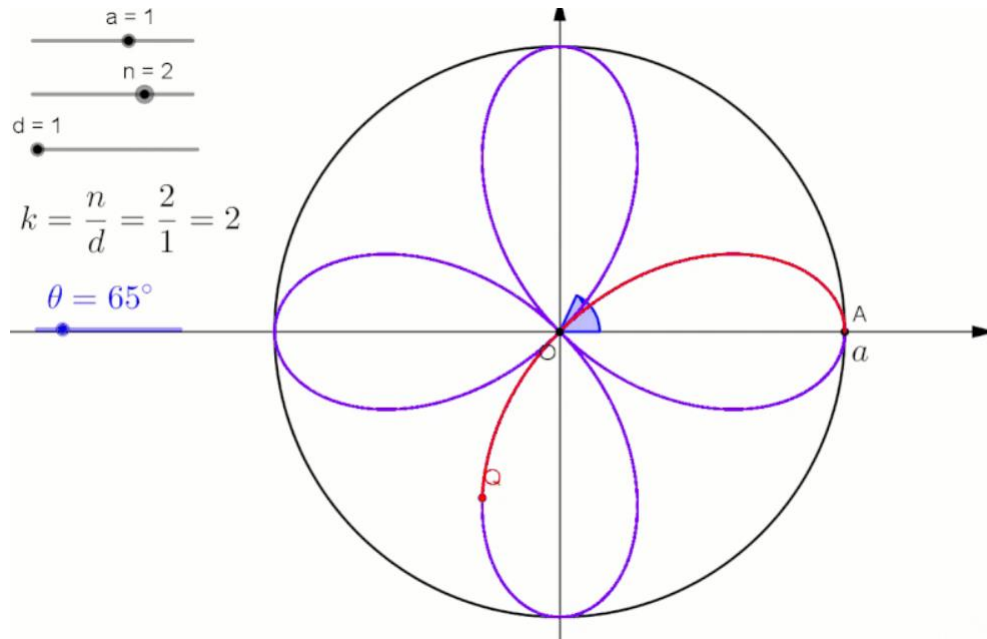
```
for (double i = 0; i < THETA_2; i += 0.01)
{
    dPos[0] = (rose_point_X + leaf_size_num * cos(leaf_cnt * i) * cos(i)) * m_posRevisionX;
    dPos[1] = (rose_point_Y + leaf_size_num * cos(leaf_cnt * i) * sin(i)) * m_posRevisionY;
}
```

꽃잎을 구현하기 위해
매개변수방정식을 구하여
극좌표에서 직교좌표로 변
환을 한다.



Solution

꽃잎의 개수에 따라 다른 범위 설정



```
if (leaf_cnt % 2 == 1) //홀수 -> 3,5,7장
{
    for (double i = 0; i < THETA_2; i += 0.01) { ... }
}
else if (leaf_cnt % 2 == 0) //짝수 -> 4, 8장
{
    for (double i = 0; i < THETA; i += 0.01) { ... }
}
```


Solution

다중 if문

If(꽃잎의 개수)

~

else

If(꽃의 사이즈)

~

else

If(클릭 영역)

~

else

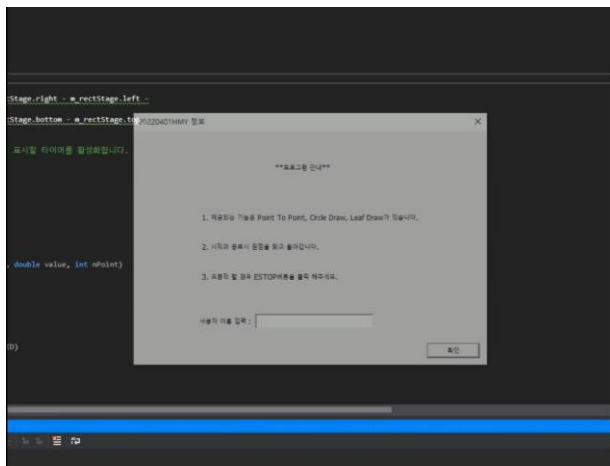
```
if (leaf_size_num<=150&&leaf_size_num>=50)    //Leaf Size 범위
{
    if (point.x >= rect2.left + leaf_size_num+5 && point.x <= rect2.right - leaf_size_num) //클릭 범위 제한 if문
    {
        if (point.y <= rect2.bottom - leaf_size_num+5 && point.y >= rect2.top + leaf_size_num+5)
        {
            draw_rose(point, nSliderEndPosY);
        }
    }
}

void CMy20220401HMYDlg::draw_rose(CPoint point_x, int sePose)
{
    CPoint point_val;
    int SliderPosY;
    point_val = point_x;
    SliderPosY = sePose;
    if (rose_click_flag == true){ ... }
    else
    {
        AfxMessageBox("꽃잎의 갯수를 클릭해주세요!");
    }
    r_point_x = (int)rose_point_X;
    r_point_y = (int)rose_point_Y;
    return;
}
```

꽃잎에 대한 크기, 개수와 영역제한에 대한 설정들로 if문이 많음.
따라서 중복되는 프로그래밍을 막기 위해 반복되는 알고리즘은 함수로 구현.

Add-Develop Solution

1. 사용자 정보 입력
2. System On/Off시 영점 이동



CAboutDlg를 이용해 초반 사용자 이름 입력 다이얼로그를 만들었고, 로그인 기능이 아닌 누가 이 기기를 사용했는지 알기 위함.

System On인 경우 Initstate에서 동작하도록 설정
-> 서보on -> HomeSearch

System Off인 경우 Onclose 함수에서 조건을 통해 동작
-> Okbutton? -> HomeSearch

```
void CAboutDlg::OnBnClickedOk()
{
    CString str;    //Edit Ctrl 에서 문자열을 가져오기 위한 과정
    GetDlgItemText(IDC_USER_NAME_INPUT, str);
    //유저 이름을 입력하지 않으면 다음 단계로 넘어가지 않는다.
    if (str.IsEmpty()==TRUE) { ... }
    else { ... }
}

BOOL CMy20220401HMYDlg::InitState()
{
    AxmSignalServoOn(m_lAxisNoY, 1);
    AxmSignalServoOn(m_lAxisNoX, 1);

    OnBnClickedBtnXHomeSearch();
    OnBnClickedBtnYHomeSearch();
    return TRUE;
}

void CMy20220401HMYDlg::OnClose()
{
    // TODO: 여기에 메시지 처리기 코드를 추가 및/또는 기본값을 호출합니다.
    // "종료 하시겠습니까?" 확인 팝업 창을 출력하여 종료 여부를 한 번 더 확인합니다.
    if (AfxMessageBox("종료 하시겠습니까?", MB_OKCANCEL) == IDCANCEL)
    {
        return;
    }
    else
    {
        OnBnClickedBtnXHomeSearch();
        OnBnClickedBtnYHomeSearch();
        AfxMessageBox("안녕히가세요");
    }
    CDialogEx::OnClose();
}
```

3. 도형 파라미터 입력



```
void CMY20220401HMYDlg::OnCbSelchangeCombo2()
{
    //잎의 개수를 받아오는 함수
    int nSel = m_combo_control.GetCurSel();
    m_combo_control.GetLBText(nSel, leaf_cnt_str);
    inputsize.GetWindowTextA(leaf_size);
    if (leaf_cnt_str == "3")
    {
        GetDlgItem(IDC_EDIT_SIZE_INPUT)->EnableWindow(TRUE);
        leaf_cnt = 3;
        AfxMessageBox("3 LEAF");
    }
    else if (leaf_cnt_str == "4") { ... }
    else if (leaf_cnt_str == "5") { ... }
    else if (leaf_cnt_str == "7")
    {
        leaf_cnt = 7;
        GetDlgItem(IDC_EDIT_SIZE_INPUT)->EnableWindow(TRUE);
        AfxMessageBox("7 LEAF");
    }
    else if (leaf_cnt_str == "8") { ... }
}
```

```
//저속, 고속 버튼에 대한 코드
void CMY20220401HMYDlg::OnBnClickedRadio3()
{
    fast_mode = true;
    slow_mode = false;
    SetDlgItemInt(IDC_EDT_X_INPUT_VEL, 70);
    SetDlgItemInt(IDC_EDT_Y_INPUT_VEL, 70);
}

void CMY20220401HMYDlg::OnBnClickedRadio01d()
{
    fast_mode = false;
    slow_mode = true;
    SetDlgItemInt(IDC_EDT_X_INPUT_VEL, 10);
    SetDlgItemInt(IDC_EDT_Y_INPUT_VEL, 10);
}
```

```
void CMY20220401HMYDlg::OnBnClickedButton1()
{
    GetDlgItemText(IDC_EDIT_SIZE_INPUT, leaf_size);
    leaf_size_num = _ttoi(leaf_size);
    if (leaf_size_num > 150)
    {
        AfxMessageBox("입력범위 초과 ! 다시 입력하세요");
        rose_click_flag = false;
    }
    else if (leaf_size_num < 50) { ... }
    else
    {
        AfxMessageBox("입력버튼 눌림");
        rose_click_flag = true;
    }
}
```

ComboBox에서 잎의 개수를 받는다.
: 클릭한 칸의 문자를 검출해 변수에 저장됨

저속 : 10, 고속 : 70, 파라미터를 변경했을 때 진동이 적고 무리가 없다 판단하여 설정.


입력한 잎 사이즈가 50~150 밖 이라면 동작하지 않고 경고창이 뜨게 된다.

4. 로그 파일 저장

```
void CMy20220401HMYD1g::log_file_write(int t)
if (btn_ptp && !btn_cp && !btn_rose)
{
    if (file.Open(_T("ptp_log_file.txt"), CFile::modeCreate | CFile::modeNoTruncate | CFile::modeWrite, NULL))
    {
        file.SeekToEnd();
        file.WriteString(sTime + "\n");
    }
}
else if (!btn_ptp && btn_cp && !btn_rose)
{
    if (file.Open(_T("cp_log_file.txt"), CFile::modeCreate | CFile::modeNoTruncate | CFile::modeWrite, NULL))
    {
        file.SeekToEnd();
        file.WriteString(sTime + "\n");
    }
}
else if (!btn_ptp && !btn_cp && btn_rose)
{
    if (file.Open(_T("leaf_log_file.txt"), CFile::modeCreate | CFile::modeNoTruncate | CFile::modeWrite,
    {
        file.SeekToEnd();
        file.WriteString(sTime + "\n");
    }
}
file.Close();
```

동작 시간 : Thread 동작 시간

```
startT = clock();
while (1) { ... }
endT = clock();
time_val = (int)((endT - startT) / CLOCKS_PER_SEC);
line_thread->log_file_write(time_val);
```



leaf_log_file.txt	20
ptp_log_file.txt	20
ReadMe.txt	20
resource.h	20
stdafx.cpp	20
stdafx.h	20
targetver.h	2022-04-01 오후...
user_log.txt	2022-05-02 오후...

user_log.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

2022년 5월 2일 17시 39분 48초 이름 : 김영준2017130010

leaf_log_file.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

2022년 5월 2일 17시 40분 39초 #클릭 좌표 : (202, 216) # 동작 시간 : 19초
 2022년 5월 2일 17시 41분 0초 #클릭 좌표 : (248, 200) # 동작 시간 : 7초
 2022년 5월 2일 17시 41분 34초 #클릭 좌표 : (145, 212) # 동작 시간 : 24초

Windows (CRLF) Ln 1, Col 1 100%

5. 동작 영역 제한



```

if (point.x >= rect2.left + leaf_size_num + 5 && point.x <= rect2.right - leaf_size_num)
{
    if (point.y <= rect2.bottom - leaf_size_num + 5 && point.y >= rect2.top + leaf_size_num + 5)
    {
        draw_rose(point, nSliderEndPosY);
    }
    else
    {
        AfxMessageBox("영역 밖 입니다.");
    }
}
else
{
    AfxMessageBox("영역 밖 입니다.");
}

```

[클릭된 포인트의 좌표 + 잎의 크기]

Solution

이송 중 다른 동작 불가



```
void CMy20220401HMYDlg::button_disable()  
{  
    GetDlgItem(IDC_RADIO1)->EnableWindow(FALSE);  
    GetDlgItem(IDC_RADIO_PTP)->EnableWindow(FALSE);  
    GetDlgItem(IDC_RADIO_CP)->EnableWindow(FALSE);  
    GetDlgItem(IDC_EDIT_SIZE_INPUT)->EnableWindow(FALSE);  
    GetDlgItem(IDC_RADIO_OLD)->EnableWindow(FALSE);  
    GetDlgItem(IDC_RADIOI3)->EnableWindow(FALSE);  
}  
  
void CMy20220401HMYDlg::button_enable()  
{  
    GetDlgItem(IDC_RADIO1)->EnableWindow(TRUE);  
    GetDlgItem(IDC_RADIO_PTP)->EnableWindow(TRUE);  
    GetDlgItem(IDC_RADIO_CP)->EnableWindow(TRUE);  
    GetDlgItem(IDC_RADIO_OLD)->EnableWindow(TRUE);  
    GetDlgItem(IDC_RADIOI3)->EnableWindow(TRUE);  
}
```

동작하는 경우 위 함수를 통해 모든 버튼을 비활성화 하여 다른 동작을 못하도록 막는다.

Supplement Point

신형 - 구형 스위칭 옵션



```
// Limit Level을 LOW로 설정합니다.  
dwRetCode = AxmSignalSetLimit(m_lAxisNoX, EMERGENCY_STOP, LOW, LOW);  
dwRetCode = AxmSignalSetLimit(m_lAxisNoY, EMERGENCY_STOP, LOW, LOW);  
  
// Unit Per Pulse를 10:1048576으로 설정합니다.  
dwRetCode = AxmMotSetMoveUnitPerPulse(m_lAxisNoX, 10, 1048576);  
dwRetCode = AxmMotSetMoveUnitPerPulse(m_lAxisNoY, 10, 1048576);
```

```
#define DEFAULT_UNIT 10
```

신형 구형 고를 수 있다면
호환성이 좋은 프로그램이 될 수 있었다.

감사합니다.
