

DU-도전학기 결과보고서

성 명		학 번	
단과대학	정보통신대학	학과(전공)	전자제어공학 (복수전공:AI응용전공으로학점신청)
도전학기 과제명	[국문] 음성 제어를 통한 방법 드론 [영문] Anti-crime drones with voice control		
지도교수 의견	상기 학생은 열정을 갖추어 본 사업을 잘 진행 하였습니다.		

1. 도전 과제의 목표

작품제작에 있어서, 하드웨어부분에는 드론을 제어하여 호버링, 상승, 하강 등 드론이 가져야할 기능 구현하는 것이 1차 목표입니다. 2차 목표로는 소프트웨어적으로 조종기 없이 드론과 통신하는 안드로이드 스튜디오 기반 애플리케이션 제작입니다. 끝으로, 제작한 애플리케이션을 활용하여 음성인식을 통해 드론을 제어하는 것이 최종 목표입니다.

완성된 작품을 통해 소속된 동아리 회원들에게 세미나를 통해 아이템을 설명하고, 구현 영상을 제작하여 youtube에 업로드 할 것입니다. 교내외 공모전을 통해 작품으로 참가하고, 음성인식 관련 드론 기술 특허를 출원하는 것이 최종목표입니다.

2. 도전 과제 내용



- 드론이 조종기 조작없이 소프트웨어로 정의하여 자율 주행이 가능한 어플 개발
- 비행체 안정화 기술을 사용하여 스스로 호버링이 가능한 드론 제작
- 안드로이드 스튜디오 학습
- 소속 동아리 회원들에게 작품 세미나 발표
- ICT 및 드론관련 교내외 공모전 참가
- 카카오 오븐 등을 활용한 애플리케이션 프로토타입 제작

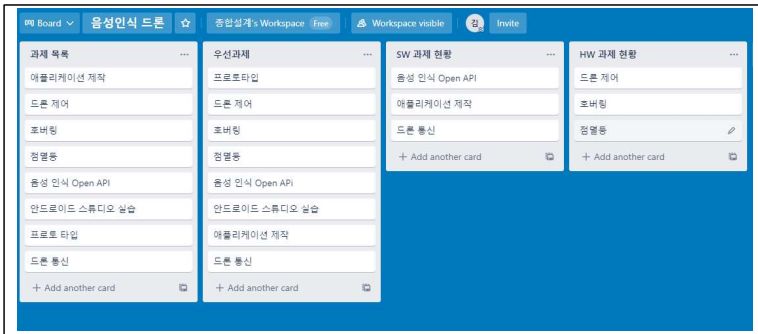
- Trello를 통한 일정관리 및 프로젝트 과제 관리
- 구현 영상 제작 및 업로드

3. 도전 과제의 성과

□ 도전 과제의 진행 완료 상황

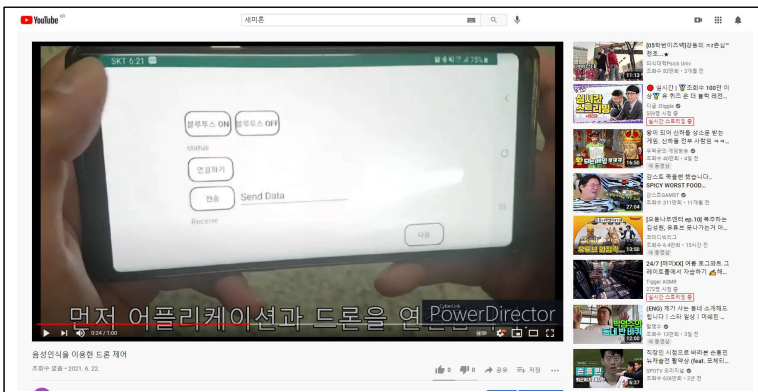
주차	활동 목표	활동 내용	진행 상황
1주차	작품 설계	실시간 음성 관련 학습 및 관련 소프트웨어 설계	100 %
2주차	작품 설계	드론관련 학습, 제작 및 하드웨어 설계	100 %
3주차	작품 설계	안드로이드 스튜디오를 소프트웨어 학습 및 소프트웨어 설계	100 %
4주차	작품 제작	안드로이드 스튜디오를 활용한 음성인식 소프트웨어 제작	100 %
5주차	작품 제작	안드로이드 스튜디오를 활용한 음성인식 소프트웨어 제작	100 %
6주차	작품 제작	드론의 호버링, 상승, 하강 등 기능 도출	100 %
7주차	작품 제작	안드로이드 스튜디오기반의 애플리케이션과 드론의 통신	100 %
8주차	작품 제작	애플리케이션을 이용한 드론제어 설계 및 제작	100 %
9주차	작품 제작	애플리케이션을 이용한 드론제어 설계 및 제작	100 %
10주차	작품 제작	음성인식을 통한 드론제어 설계 및 제작	100 %
11주차	작품 제작	음성인식을 통한 드론제어 설계 및 제작	100 %
12주차	작품 완성	최종적인 음성인식 애플리케이션을 이용한 드론 제작	100 %
13주차	작품 테스트	소스코드 및 하드웨어 오류 분석	100 %
14주차	작품 테스트	소스코드 및 하드웨어 오류 분석	100 %
15주차	특허 출원	소스코드 및 하드웨어 오류 분석	100 %
16주차	결과보고서 및 영상 제작	완성된 결과물을 통한 결과보고서 작성 및 구현 영상 제작	100 %

□ trello를 이용한 과제 목록 정리



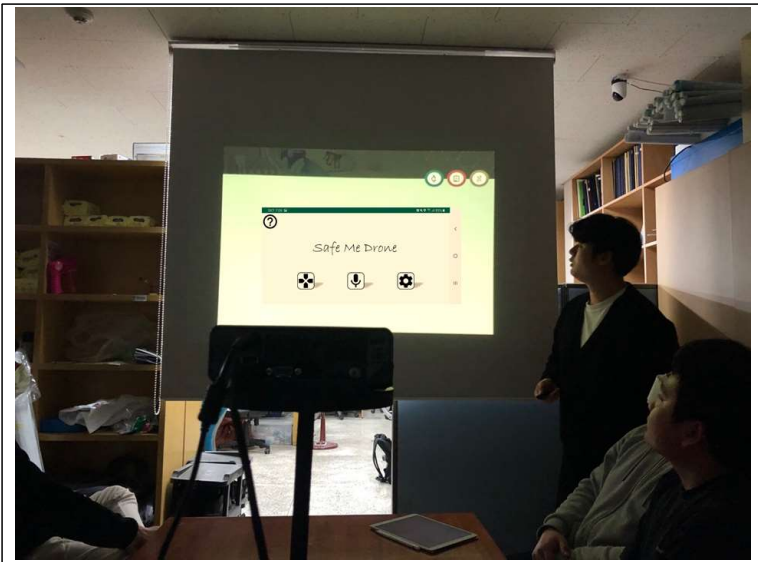
- 과제 목록을 정리하는 trello를 이용하였다.
- <https://trello.com/b/RpwYy33C/%EC%9D%8C%EC%84%B1%EC%9D%B8%EC%8B%9D-%EB%93%9C%EB%A1%A0>

□ 작품소개 유튜브 영상 업로드



- 작품소개 영상을 직접 제작하여 개인 유튜브에 업로드 하였다

□ 작품관련 세미나 발표



- 작품 관련 세미나발표를 학과동아리에서 하였다.
- 교외 대회 참가

치안현장 아이디어 신청서(소형드론 활용 해결방안)

성명 / 연락처	성명 : 김 동 우 / 연락처 : 010-5653-8527
소속	□ 임원인 □ 경찰관
아이디어 명	Safe.Me.Drone (세이프존)
주제	음성 인식 드론을 활용한 범죄예방 방안 아이디어

□ 개발 배경

■ CCTV 설치의 빈번한 연관성

범용형 CCTV 설치 및 연계 현황

범용형 CCTV 설치 현황

- 총 415대 CCTV 설치되어 있는 서울 중구, 범죄율 3.3% 증가
- CCTV 설치 여부 범용형 발생 및 미수 58% 통계 결과
- 이면 인식 범용드론은 범용 설치 및 사용 범의 커다란 적용 유리

■ 범용 드론의 필요성

드론 활용도 증가

범용 순찰에 사용되는 드론

- 소프트웨어 기술 개발도 인한 드론 활용도 상승
- CCTV 설치여부 구별 운용 강화 필요
- 음성 인식 드론을 통한 범용 범용 범죄 예방 가능성 기대

□ 개요

■ 아이디어 소개

음성 인식 드론 프로토타입 교안

범용 서비스 모델 사용된 범용 드론

- 차량의 움직임을 인식 하고 자율 주행 가능
- 초해상도 카메라 드론 촬영, 주요 범용을 촬영 또는 원격 제어 사용
- 드론 외부 프레임에 RED LED, BLUE LED 등 유용한 조명도 탑재
- 비행제어 및 영상 전송 가능

발명아이디어 설명서

발명주제	<input type="checkbox"/> 주제 1 (스마트시티 도시와 관련된 발명품) <input type="checkbox"/> 주제 2 (에너지 절약할 수 있는 발명품) <input checked="" type="checkbox"/> 주제 3 (재난, 자연재해 대비, 기타 안전을 위한 발명품) <input type="checkbox"/> 기타
발명의 명칭	음성 인식 드론을 이용한 범죄 예방 방법 (The Crime Prevention Method Using Voice Recognition Drone)

[I] 발명의 개요

1. 발명의 동기

가. 종래기술 및 문제점

□ 드론 (한국 등록특허 제 10-2016-0075502)

대표 도면

컴퓨팅 환경 블록도

■ 문제기술

드론은 사람이 탑승하지 않고 원격 조정에 의해 비행하거나 지정된 경로를 따라 자율적으로 비행하는 비행체로서, 주로 군사적 용도로 활용되어 왔으나, 최근 운송 분야, 보안 분야 등 다양한 분야에서 활용되고 있으며, 개인적인 용도로 이용된다. 로터의 개수 및 형태에 따라 단일 로터형, 동축 반결형, 워드 플러 등으로 구분되고 해당 기술은 4개의 로터로 이루어진 쿼드콥터를 사용한다. 4개의 로터중 2개의 로터는 시계방향으로 나머지 2개의 로터는 반 시계 방향으로 회전하고 이때, 대각선 방향의 로터는 동일한 방향으로 회전한다.

■ 문제점

드론이 진행하고자 하는 방향에 있는 로터 회전 속도를 줄이고, 반대 방향에 있는 로터의 회전 속도를 높이면 드론이 진행하고자 하는 방향으로 기울어지면서 기울어진 상태로 이동하게 되는데 이는 운반 및 기타 내장 장비 부파에 어려움 점이 있다.

- 치안현장 아이디어 경진대회에 참가하였다.
- 동아리 세미나를 통해 음성인식 드론을 방법에 활용하여 경진대회에 참가하였다.
- 달서구 발명 아이디어 경진대회에 참가하였다.

4. 자기 평가

○ 예상 투입 시간 변경 (9시간 → 10시간)

- 초기 계획으로 주차마다 3시간 3회 진행 하여 9시간의 계획이 프로그래밍 언어 및 실무 능력이 부족하여 언어 정보 및 학습을 위한 시간 투입을 증가 시켰다. 최종적으로 주차마다 10시간 투입하여 프로젝트를 진행 완료 하였다.

5. 최종 결과물

□ 하드웨어 주요 기능

기능	설명	작품실물사진
호버링 드론 (구매H/W)	제자리에서 정지 비행의 기능과 코딩하여 여러 기능을 추가할 수 있는 드론 구매 컨트롤러를 만지지 않아도 제자리에서 정지하여 비행	
LED 점멸등	방법 드론의 시각적 효과를 낼 수 있게 드론의 외부 프레임에 RED LED와 BLUE LED를 이용하여 점멸등을 제작	
블루투스를 이용한 스마트 폰과의 통신	드론과 스마트폰의 어플리케이션 연동을 위한 블루투스 통신	

□ 제작한 어플리케이션 UI

	
<p><#1 아이콘 디자인></p>	<p><#2 어플리케이션 아이콘></p>
	
<p><#3 메인 화면></p>	<p><#4 도움말></p>
	
<p><#5 음성화면></p>	<p><#6 GPS 화면></p>
	
<p><#7 수동 조작 화면></p>	

□ 소스코드

```
import com.example.safemedrone_v1.R;
import com.zerokol.views.joystickView.JoystickView;
import com.zerokol.views.joystickView.JoystickView.OnJoystickMoveListener;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.TextView;

public class JoystickActivity extends Activity {
    private TextView angleTextView;
    private TextView powerTextView;
    private TextView directionTextView;           조이스틱 관련 필드정의
    // Importing also other views
    private JoystickView joystick;

    private Button gps_btn;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_joystick);

        angleTextView = (TextView) findViewById(R.id.angleTextView);
        powerTextView = (TextView) findViewById(R.id.powerTextView);
        directionTextView = (TextView) findViewById(R.id.directionTextVie
        //Referencing also other views
        joystick = (JoystickView) findViewById(R.id.joystickView);
    }
}
```

XML파일로부터 ID를 불러 온다

```
gps_btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(JoystickActivity.this, GpsActivity.class);
        startActivity(intent);
    }
});
```

GPS 아이콘 실행 시, GPS 메인
화면으로 이동

```
//Event listener that always returns the variation of the angle in degrees, motion power in percentage and direction of movement
joystick.setOnJoystickMoveListener(new OnJoystickMoveListener() {
```

```
    @Override
    public void onValueChanged(int angle, int power, int direction) {
        // TODO Auto-generated method stub
        angleTextView.setText(" " + String.valueOf(angle) + "");
        powerTextView.setText(" " + String.valueOf(power) + "%");
        switch (direction) {
            case JoystickView.FRONT:
                directionTextView.setText(R.string.front_lab);
                break;
            case JoystickView.FRONT_RIGHT:
                directionTextView.setText(R.string.front_right_lab);
                break;
            case JoystickView.RIGHT:
                directionTextView.setText(R.string.right_lab);
                break;
            case JoystickView.RIGHT_BOTTOM:
                directionTextView.setText(R.string.right_bottom_lab);
                break;
            case JoystickView.BOTTOM:
                directionTextView.setText(R.string.bottom_lab);
                break;
        }
    }
});
```

조이스틱 조작 시, 조이스틱의
각도, 힘, 방향의 값을 받아옴


```
import android.Manifest;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Build;
import android.os.Bundle;
import android.os.Environment;
import android.os.Handler;
import android.os.Message;
```

외부 실행 파일에 있는 함수 및
라이브러리 참조

```
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
```

```
import com.example.safemedrone_v1.utils.AudioWriterPCM;
import com.naver.speech.clientapi.SpeechRecognitionResult;
```

```
import java.lang.ref.WeakReference;
import java.util.List;
```

```
public class SpeechActivity extends Activity {
```

```
    private static final String TAG = SpeechActivity.class.getSimpleName();
    private static final String CLIENT_ID = "r3zft8vc3r";
```

NAVER API의 클라이언트 ID를
정보 입력

```

private TextView txtResult;
private Button btnStart;
private String mResult;
private Button gps_btn2;

private AudioWriterPCM writer;

// 음성인식 메시지 처리
private void handleMessage(Message msg) {
    switch (msg.what) {
        case R.id.clientReady:
            // Now an user can speak.
            txtResult.setText("Connected");
            writer = new AudioWriterPCM(
                Environment.getExternalStorageDirectory().getAbsolutePath() + "/NaverSpeechTest");
            writer.open("Test");
            break;
        //음성인식 진행될때
        case R.id.audioRecording:
            writer.write((short[]) msg.obj);
            break;

        case R.id.partialResult:
            // 처리 되는중 결과 받는 경우
            mResult = (String) (msg.obj);
            txtResult.setText(mResult);
            break;
            //최종인식 완료되면 결과 보여줌
        case R.id.finalResult:
            // Extract obj property typed with String array.
            // The first element is recognition result for speech.
            SpeechRecognitionResult speechRecognitionResult = (SpeechRecognitionResult) msg.obj;
            List<String> results = speechRecognitionResult.getResults();
            StringBuilder strBuf = new StringBuilder();
            for(String result : results) {

```

```

public void onClick(View v) {
    if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
        int permissionResult = checkSelfPermission(Manifest.permission.RECORD_AUDIO);
        안드로이드 버전 체크

        if(permissionResult == PackageManager.PERMISSION_DENIED) {
            if(shouldShowRequestPermissionRationale(Manifest.permission.RECORD_AUDIO)) {
                AlertDialog.Builder dialog = new AlertDialog.Builder(SpeechActivity.this);
                dialog.setTitle("권한이 필요합니다.");
                휴대폰 내부 음성 권한 자동 허용

                .setMessage("이 기능을 사용하기 위해서는 권한이 필요합니다. 계속하시겠습니까?")
                .setPositiveButton("네", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                            requestPermissions(new String[]{Manifest.permission.RECORD_AUDIO}, 1000);
                            휴대폰 내부 음성 권한 확인
                        }
                    }
                })
                .setNegativeButton("아니오", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        Toast.makeText(SpeechActivity.this, "기능을 취소했습니다.", Toast.LENGTH_SHORT).show();
                    }
                })
                .create()
                .show();
            }
            else{
                requestPermissions(new String[]{Manifest.permission.RECORD_AUDIO},1000);
                최초 휴대폰 내부 음성 권한
                요청하는 경우
            }
        }
        else {
            if(naverRecognizer.getSpeechRecognizer().isRunning()) {
                최초 휴대폰 내부 음성 권한이 허용된 경우
            }
        }
    }
}

```

```

@Override
protected void onStart() {
    super.onStart();
    // 음성인식 서버 초기화
    naverRecognizer.getSpeechRecognizer().initialize();
}

@Override
protected void onResume() {
    super.onResume();

    mResult = "";
    txtResult.setText("");
    btnStart.setText(R.string.str_start);
    btnStart.setEnabled(true);
}

@Override
protected void onStop() {
    super.onStop();
    // 음성인식 서버 종료
    naverRecognizer.getSpeechRecognizer().release();
}

// 메시지 처리하는 핸들러 정의
static class RecognitionHandler extends Handler {
    private final WeakReference<SpeechActivity> mActivity;

    RecognitionHandler(SpeechActivity activity) {
        mActivity = new WeakReference<SpeechActivity>(activity);
    }
}

```