

종합설계 프로젝트 수행 보고서

프로젝트명	센서로 Spindle 진단이 가능한 smart 공작기계
팀번호	S3-10
문서제목	수행계획서() 2차발표 중간보고서() 3차발표 중간보고서() 최종결과보고서(O)

2020.12.04

팀원 : 한세은 (팀장)
김채영 (팀원)

지도교수 : 서대영 교수 (인)
지도교수 : 최진구 교수 (인)

문서 수정 내역

작성일	대표작성자	버 전 (Revision)	수정내용	
2020.01.22	한세은(팀장)	1.0	수행계획서	최초작성
2020.04.09	김채영	2.0	시험시나리오	
2020.05.02	김채영	2.1	상세설계	
2020.06.27	한세은	3.0	테스트	
2020.07.30	한세은	3.1	데모	
2020.08.30	한세은,김채영	4.0	서론~본론 수정	
2020.10.29	한세은,김채영	4.1	본론 수정	
2020.11.29	한세은	5.0	결론 작성	최종본

문서 구성

진행단계	프로젝트 계획서 발표	중간발표1 (2월)	중간발표2 (4월)	학기말발표 (6월)	최종발표 (10월)
기본양식	계획서 양식	계획서 양식	계획서 양식	계획서 양식	계획서 양식
포함되는 내용	I. 서론 (1~6) II. 본론 (1~3) 참고자료	I. 서론 (1~6) II. 본론 (1~4) 참고자료	I. 서론 (1~6) II. 본론 (1~5) 참고자료	I. 서론 (1~6) II. 본론 (1~7) 참고자료	I II III

이 문서는 한국산업기술대학교 컴퓨터공학부의 “종합설계”교과목에서
프로젝트“센서로 Spindle 진단이 가능한 smart 공작기계”을 수행하는
(S3-10, 한세은, 김채영)들이 작성한 것으로 사용하기 위해서는
팀원들의 허락이 필요합니다.

목 차

I. 서론

1. 작품선정 배경 및 필요성	4
2. 기존 연구/기술동향 분석	
3. 개발 목표	5
4. 팀 역할 분담	6
5. 개발 일정	
6. 개발 환경	

II. 본론

1. 개발 내용	7
2. 문제 및 해결방안	
3. 시험시나리오	
4. 상세 설계	8
5. Prototype 구현	30
6. 시험/ 테스트 결과	34
7. Coding & DEMO	35

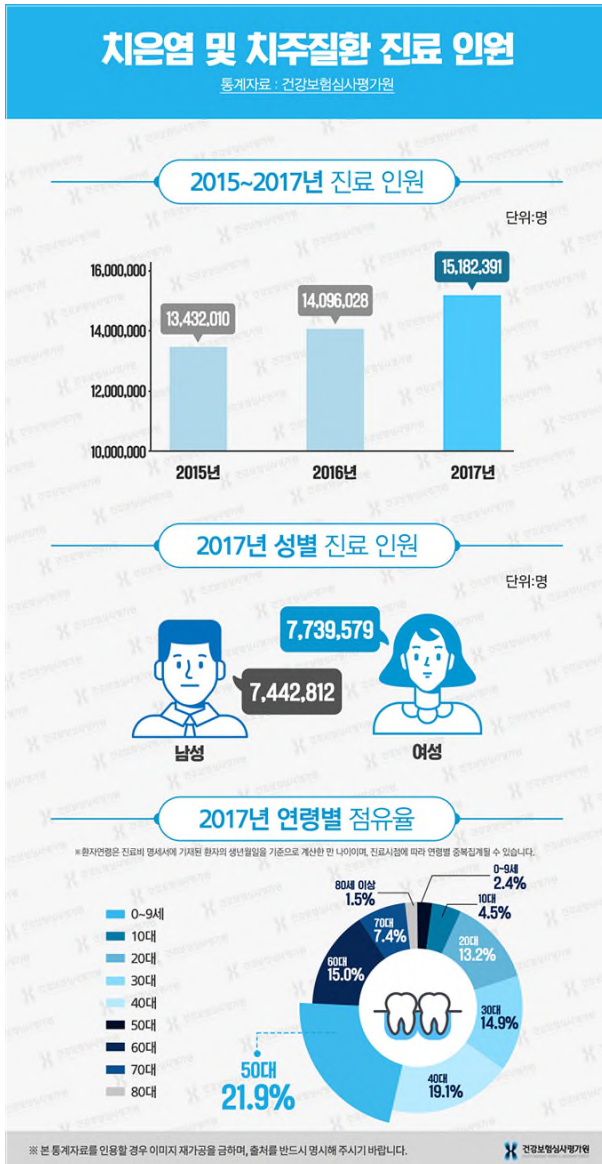
III. 결론

1. 연구 결과	37
2. 작품제작 소요재료 목록	38

참고자료	39
------------	----

I. 서론

1. 작품선정 배경 및 필요성



최근 치주질환 환자의 수가 지속적으로 증가함에 따라 치아 가공 산업 성장의 필요성이 대두되고 있다. 건강보험심사평가원에 따르면 치주질환으로 진료를 받은 인원이 2015년도에는 1343만 명, 2016년도에는 1409만 명, 2017년에는 1518만 명으로 지속적으로 증가해 왔음을 확인할 수 있다. 다빈도 질병 통계에서는 급성기관지염 다음으로 외래 진료 환자의 수가 가장 많았다. 증가하는 환자를 위한 효율적인 치료를 위해 치가공기 또한 스마트화되어야 할 필요하다.

또한 2014년도부터 제조업 혁신 3.0 전략이 추진되면서 공장의 스마트화, 융합 신산업 창출을 통한 선도형 전략으로 경쟁우위를 확보하기 위해 힘써왔다. 이로 인해 기계장비 부분에서도 ICT/IoT를 이용한 스마트화 기술을 접목하려는 노력이 계속되고 있다.

위 4차 산업을 이루기 위하여 CNC 기계장비의 경우 아래 3단계의 연구 개발과정이 필요하다.

(1) 작업 상황 sensing, Data communication 이 가능한 스피들 연구, 개발

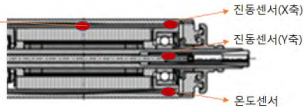

(2) 이 스피들을 활용한 CNC 기계장비의 완성 (장비 하드웨어) (3) CNC 기계장비 운전 시스템의 스마트화 (장비 소프트웨어)

- 스피들의 공구(엔드밀,드릴)가 마모, 파손되거나, 공구의 진동이 발생되면 가공품 불량 발생된다. 이러한 점을 보완하기 위해 센서를 스피들에 부착하고 센서 정보를 이용하기 위한 모니터링 시스템이 필요하다.

2. 기존 연구/기술동향 분석

현대위아의 NC장비인 아이트룰, 두산인프라코어의 Smart-i system 등은 장비 콘트롤을 일부 IT 기술이 접목되는 제품을 출시하고 있다.

이러한 대기업의 Smart 제품화도 역시 스피들 실제 상황을 모니터링 하는 기능은 부족, 운전 기능을 스마트화(개인 PC data화)하는 정도에 머무르고 있다.

구분	주요 내용	
본 연구과제	<ul style="list-style-type: none"> • 작업중인 스피들의 실제 온도, 진동 전류 및 rpm 센싱 	
현대위아(아이트룰), 두산인프라코어 (Smart-I)	<ul style="list-style-type: none"> • 기계장비의 운전 Data 스마트화 • 실제 온도, 진동 및 rpm을 센싱하는 기능 없음 	

3. 개발 목표

스피들은 운영되는 그 순간의 온도, 진동, 실제 rpm 및 전류측정 data를 monitoring 하여 장비/스피들의 상태확인 및 실시간 관리 그리고 Operator의 개인 tablet/Smart phone에서 monitoring 가능하게 한다.

4. 팀 역할 분담

이름	역할 분담
김채영	<ul style="list-style-type: none"> - 데이터 가공 모니터링 어플리케이션 개발 - 아날로그 입출력 전송 - 서버 구축
한세은	<ul style="list-style-type: none"> - 센서 신호 모니터링 프로그램 작성 - 데이터 전송 통신 모듈 - DB 와 서버 연동 - 랩뷰와 서버 연동

5. 개발 일정

	내용	12월	1월	2월	3월	4월	5월	6월	7월
1	스핀들 선정								
2	스핀들 해석 및 센서부착								
3	데이터 모니터링 UI 구현								
4	App 개발 및 연동								
5	스핀들 시운행 및 보완								
6	최종 보고서 작성 및 발표								

6. 개발 환경

OS	Window 10
Language	Java, PHP
DataBase	Mysql
Sensor	온도 센서, rpm 센서, gap 센서
Application	Android Studio
Board	NI USB-6000

II. 본론

1. 개발 내용

- 스피들이 설치된 밀링, 드릴링 기계에 접목하여 가공정보 모니터링 및 진단에 적용
- 각 센서의 특성을 고려한 샘플링타임에 대한 연구를 진행
- 센서신호들을 예측하여 normal, emergency, warning 상태 측정
- 센서 데이터에 에러가 발생하면 바로 신호를 보내주는 실시간 처리 기능
 - 안드로이드 스튜디오를 활용하여 운전자 개인 tablet/smart phone에서 모니터링 가능한 어플리케이션(안드로이드, ios 버전) 개발
- 데이터를 그래프 형식을 받아오기 위하여 오픈소스 라이브러리 활용
- 데이터는 내장 데이터베이스 말고 외부 데이터베이스 프로그램(Mysql)사용
- 가공 현상에 따른 센서의 신호특성에 대한 연구진행

2. 문제 및 해결방안

스핀들의 공구(엔드밀,드릴)가 마모, 파손되거나, 공구의 진동이 발생되면 가공품 불량 발생된다. 기계에 센서를 부착하여 데이터들을 측정하여 세가지 상태들로 분류한다.

1. normal 2. warning 3. emergency 단계로 표현하여 첫 번째 상태일 경우 기계에 아무런 이상상태가 발생하지 않은 것으로 간주하여 지속적으로 동작시킨다. 두 번째 상태일 경우 스피들의 공구에 문제가 발생하여 가공품 불량 발생될 수 있으니 기계의 동작을 중지시키고 상태를 확인하여 해결한다. 마지막 세 번째 상태인 emergency는 위험 단계로 기계의 심각한 문제가 발생하였으므로, 두 번째와 마찬가지로 동작을 중지시키고 모니터링 시스템 상으로 emergency 상태 이상을 여러번 표현해준다.

3. 시험시나리오

스핀들 기계에 다섯가지 센서(가속도, 온도, 전류, rpm, gap)를 부착한다. 데이터를 통신하기 위하여 백엔드 서버를 띄운다. 서버에 Mysql을 활용하여 데이터베이스를 구축하여 서버와 데이터베이스를 연동한다. 연동하기 위하여 통신 프로토콜은 tcp/ip를 사용할 예정이다. Mysql에 넣을 데이터들은 크게 여섯가지로 온도 센서(베어링 온도, 코일 온도), 진동 센서(런아웃 변화량, 외부 충격량, 파손 감지), rpm 센서(부하 변동량, 가공 성능), 전류 센서(작업 부하량, 절삭력, 공구 마모)와 관리자과 회원이다. 진동 센서와 rpm 센서, 전류 센서에 대한 데이터들을 측정하여 기계의 상태를 normal, warning, emergency 세가지 상태로 분류 하며 operator가 기계의 특정 문제가 생기면 바로 알 수 있도록 알려준다. 또한, 모니터링 프로그램은 이동성이 좋고 데이터 값들을 그래프 형식으로 나타내며 operator가 gui를 잘 다룰수 있도록 간단하게 구성할 수 있는 Android Studio를 이용해 모바일 어플리케이션을 개발한다.

4. 상세설계

● 데이터베이스 Spindle

1) table operator : operator 정보들이 담긴 테이블

userID	userPassword	userName	userPhone	userEmail
--------	--------------	----------	-----------	-----------

2) table temperature : 온도 센서의 데이터 값들이 담긴 테이블

value

3) table rpm : rpm 센서의 데이터 값들이 담긴 테이블

value

4) table gapx : gap_x 센서의 데이터 값들이 담긴 테이블

value

5) table gapy : gap_y 센서의 데이터 값들이 담긴 테이블

value

● 서버 php

1) UserLogin.php

user가 로그인하여 id값과 pw값을 받아 테이블에 담긴 튜플들을 비교하여 로그인 성공 여부 판단한다.

2) userRegister.php

안드로이드 앱으로부터 id, pw, name, phone, email값들을 받아 insert 쿼리문을 실행하여 회원가입을 실행한다.

4) SensorData.php

테이블에 저장되어있는 temperature 데이터값들을 안드로이드 앱으로 가져와 출력해준다.

5) gapxData.php

테이블에 저장되어있는 gap x축 데이터값들을 안드로이드 앱으로 가져와 출력해준다.

6) gapyData.php

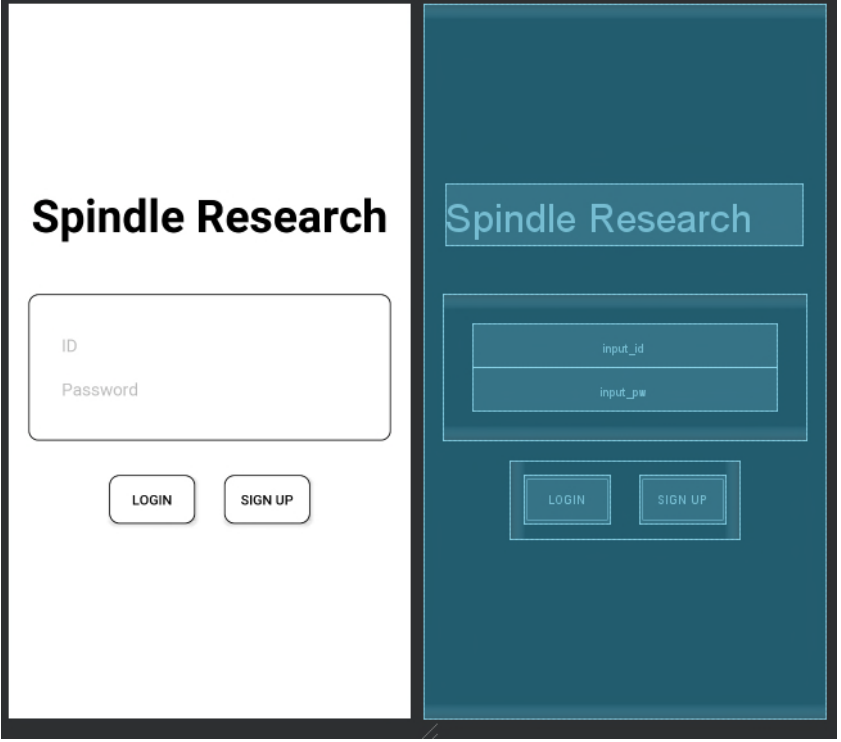
테이블에 저장되어있는 gap y축 데이터값들을 안드로이드 앱으로 가져와 출력해준다.

7) rpmData.php

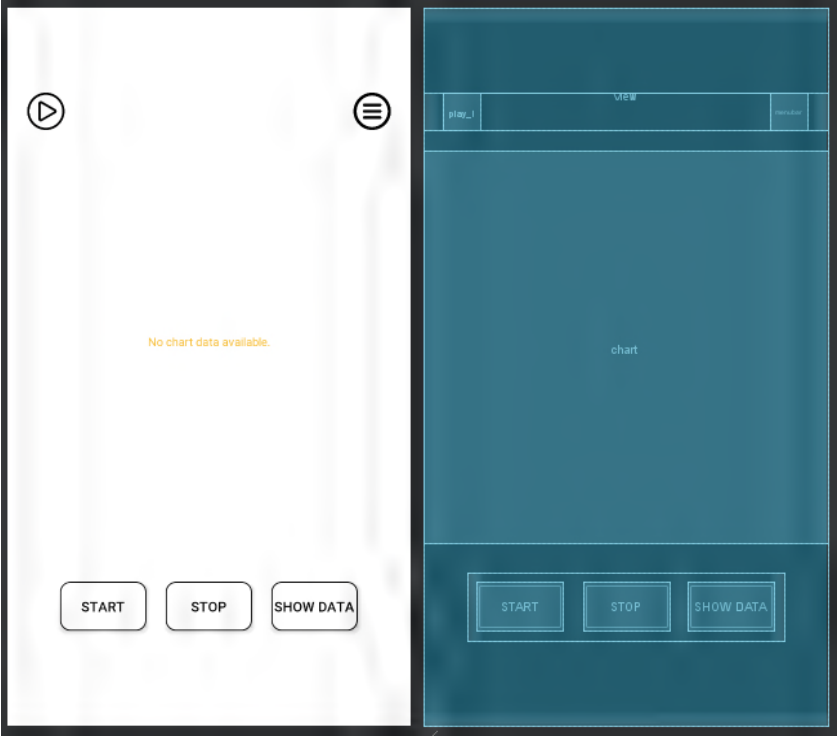
테이블에 저장되어있는 rpm 데이터값들을 안드로이드 앱으로 가져와 출력해준다.

● layout 구성

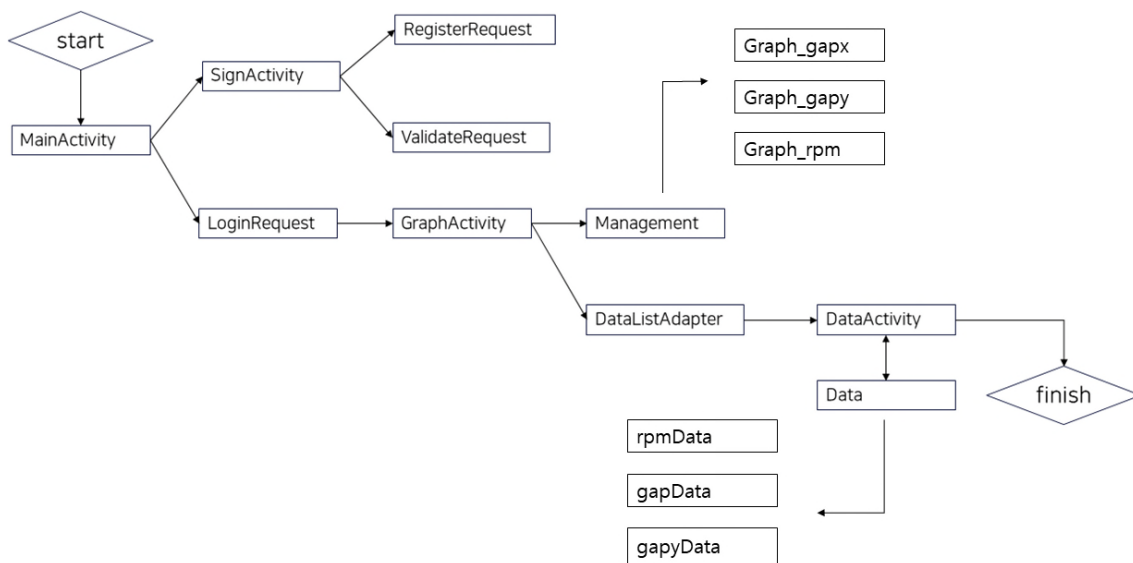
- activity_main.xml
- activity_sign.xml
- activity_data.xml
- activity_graph.xml
- activity_graph_gapx.xml
- activity_graph_gapy.xml
- activity_graph_rpm.xml
- activity_management.xml

layout name	detail
activity_main.xml	

layout name	detail
activity_sign.xml	
activity_data.xml	

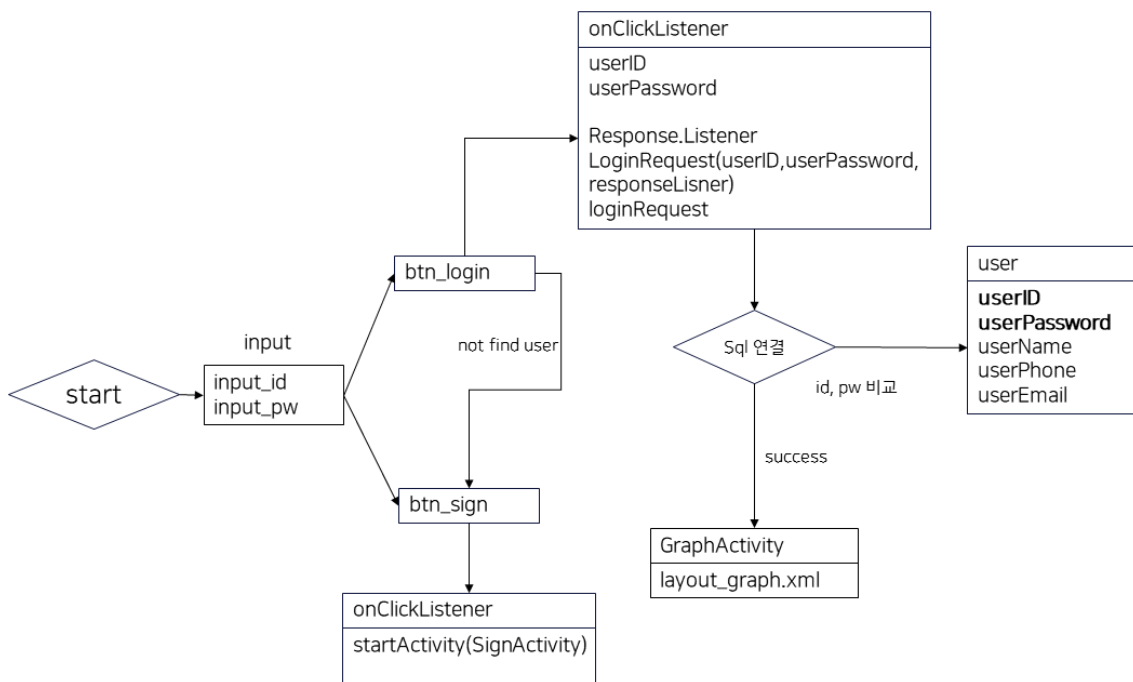
layout name	detail
activity_graph.xml	

● Activity 흐름

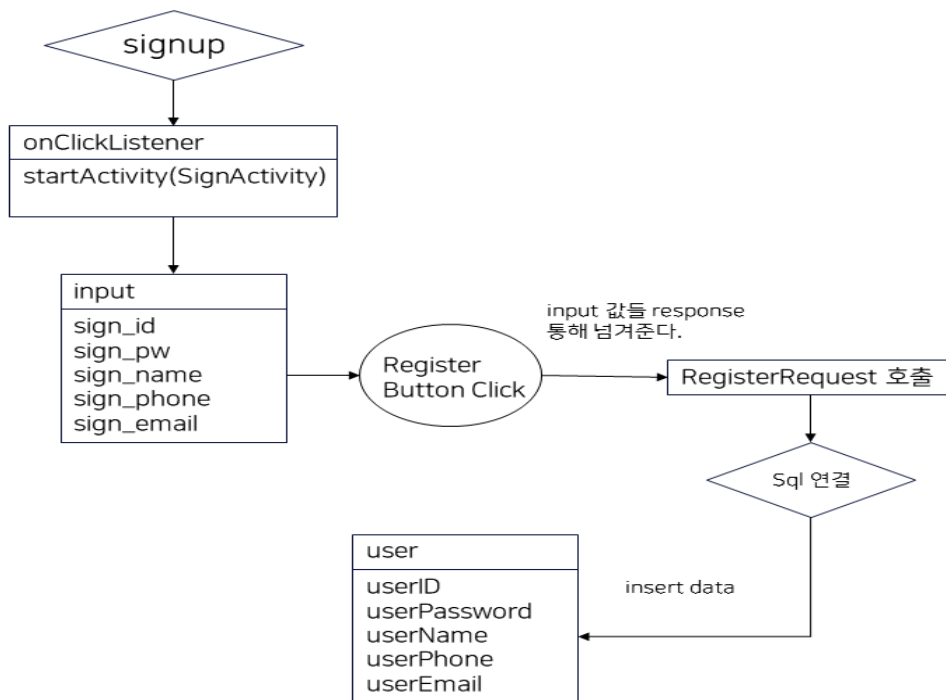


● 상세 Activity

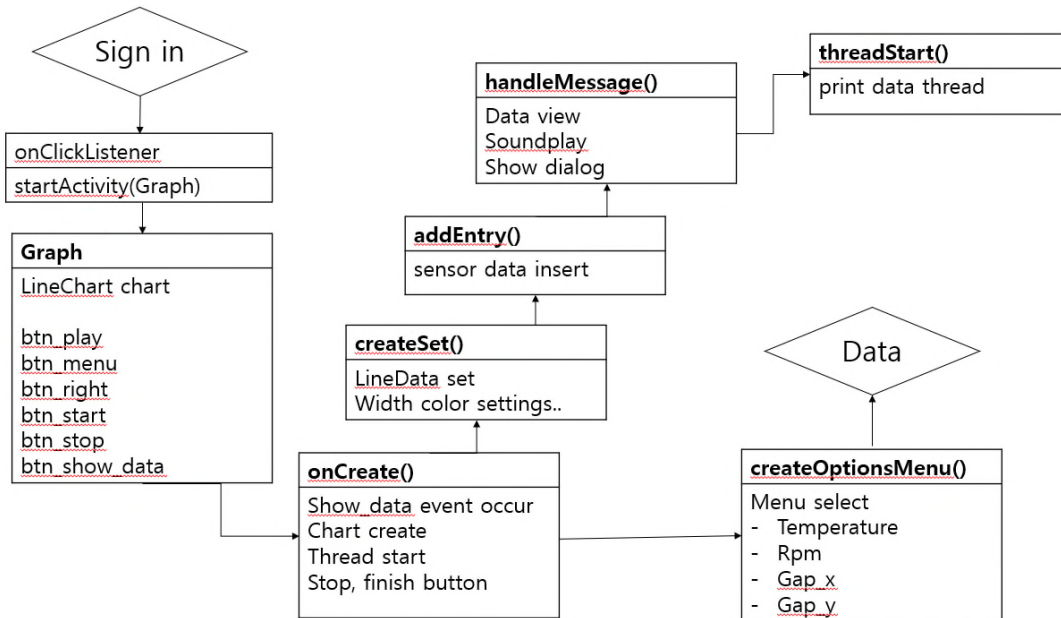
1) main - login



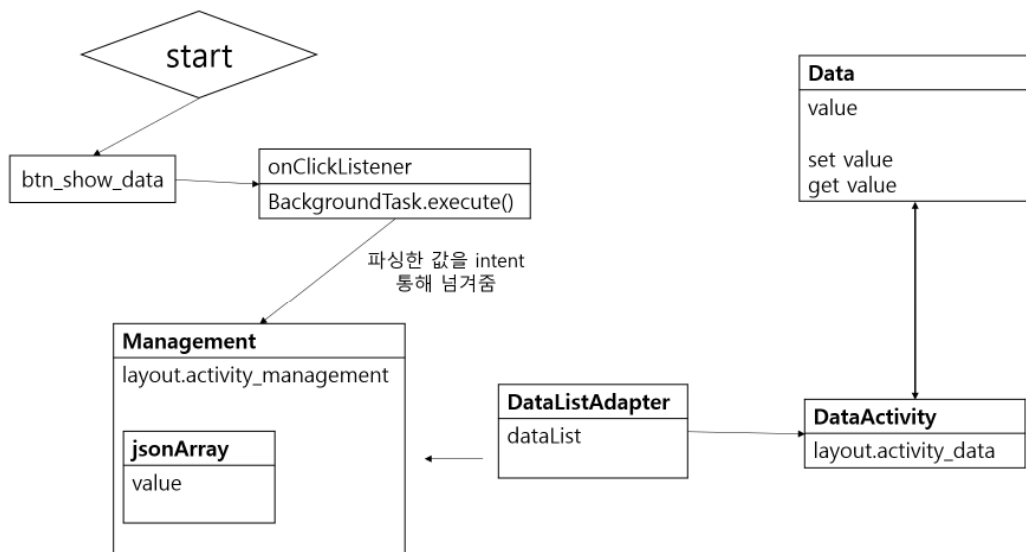
2) main - sign



3) graph



4) data



● 액티비티 시나리오

1) 회원가입

- 1-1) 홈 화면에서 sign in 버튼을 통해 액티비티 접근 가능
- 1-2) 아이디, 비밀번호, 이름, 전화번호, 이메일을 입력 받아 데이터 값들을 하나의 튜플로 구성하여 회원 데이터베이스에 insert 하여 회원가입
- 1-3) 서버 통신은 Volley 라이브러리를 사용
- 1-4) 회원가입 형식 중 빈칸이 있을 경우 회원가입이 불가능하도록 예외처리
- 1-5) 회원가입이 성공적으로 완료되면 액티비티 종료 후 홈 화면으로 돌아감

2) 로그인

- 2-1) 아이디와 비밀번호를 입력한 후 sign in 버튼을 통해 로그인
- 2-2) 서버 통신을 통해 데이터베이스의 데이터와 비교하여 올바르지 않은 계정 정보가 입력 되었을 경우 “계정을 다시 확인하세요” 다이얼로그 메시지 출력
- 2-3) 올바르게 입력된 경우 “로그인에 성공했습니다” 다이얼로그 메시지 출력 후 액티비티 전환

3) 센서 데이터

- 3-1) 온도 센서, rpm 센서, gap 센서 그래프를 메뉴를 통해 전환시키며 실시간 데이터 값들을 확인 가능
- 3-2) 데이터들의 그래프는 MPAndroid 라이브러리를 사용하여 꺾은선 그래프로 출력
- 3-3) stop 버튼은 그래프의 실시간 갱신을 중지시키고 정적인 데이터 값들을 보여줌
- 3-4) start 버튼은 갱신이 중지되었던 그래프를 다시 실시간 데이터를 동적으로 출력하도록 함
- 3-5) 알림 스위치를 on했을 때 temperature 값이 39°C 이상으로 상승하면 operator에게 warning 신호를 push 알림으로 알려줌
- 3-6) temperature 값이 44°C 이상으로 상승했을 때는 10초 동안 operator에게 dangerous 신호를 주면서 기계를 멈추도록 권고

4) show data

- 4-1) 센서 데이터를 분석·가공한 데이터로 출력
- 4-2) 분석된 데이터를 통해 센서의 상태를 normal, warning, dangerous의 세 가지 상태로 출력

● 중요 코드(어플리케이션)

code	<pre> registerButton.setOnClickListener((view) → { Intent registerIntent = new Intent(packageContext MainActivity.this, SignActivity.class); MainActivity.this.startActivity(registerIntent); }); LoginRequest loginRequest = new LoginRequest(userID, userPassword, responseLisner); RequestQueue queue = Volley.newRequestQueue(context MainActivity.this); queue.add(loginRequest); </pre>
detail	<p>MainActivity에서 registerButton을 누르면 SignActivity로 넘어가는 코드 부분과</p> <p>로그인 버튼을 누르면 id와 password의 값들을 넘겨주어 LoginRequest로 가 로그인을 해준다.</p>

code	<pre> public RegisterRequest(String userID, String userPassword, String userName, String userPhone, String userEmail, Response.Listener<String> listener){ super(Method.POST, URL, listener, errorListener: null); // 해당 URL에 POST방식으로 파마미터들을 전송할 parameters = new HashMap<>(); parameters.put("userID", userID); parameters.put("userPassword", userPassword); parameters.put("userName", userName); parameters.put("userPhone", userPhone); parameters.put("userEmail", userEmail); } </pre>
detail	<p>RegisterRequest에서 id,password,name,phone,email을 서버에 post방식으로 전송한다.</p>

code	<pre> Response.Listener<String> responseListener = (response) → { try{ JSONObject jsonResponse = new JSONObject(response); boolean success = jsonResponse.getBoolean(name: "success"); if(success){//사용할 수 있는 아이디라면 AlertDialog.Builder builder = new AlertDialog.Builder(context: SignActivity.this); dialog = builder.setMessage("Register Your ID") .setPositiveButton(text: "OK", listener: null) .create(); dialog.show(); finish();//액티비티를 종료시킴(회원등록 창을 닫음) }else{//사용할 수 없는 아이디라면 AlertDialog.Builder builder = new AlertDialog.Builder(context: SignActivity.this); dialog = builder.setMessage("Register fail") .setNegativeButton(text: "OK", listener: null) .create(); dialog.show(); } } catch(Exception e){ e.printStackTrace(); } };//Response.Listener 완료 //Volley 라이브러리를 이용해서 실제 서버와 통신을 구현하는 부분 RegisterRequest registerRequest = new RegisterRequest(userID, userPassword, userName, userPhone, userEmail, responseListener); RequestQueue queue = Volley.newRequestQueue(context: SignActivity.this); queue.add(registerRequest); </pre>
detail	<p>SignActivity에서 회원가입을 시행하는 코드와 서버와 통신하는 코드이다.</p>

code	<pre> @Override public View getView(int i, View View, ViewGroup viewGroup) { View v=View.inflate(context,R.layout.activity_data, root null); TextView ave_temp=(TextView)v.findViewById(R.id.ave_temp); TextView max_temp=(TextView)v.findViewById(R.id.max_temp); TextView dis_temp=(TextView)v.findViewById(R.id.dis_temp); TextView jud_temp=(TextView)v.findViewById(R.id.jud_temp); ave_temp.setText(dataList.get(i).getAve_temp()); max_temp.setText(dataList.get(i).getMax_temp()); dis_temp.setText(dataList.get(i).getDis_temp()); jud_temp.setText(dataList.get(i).getJud_temp()); //v.setTag(dataList.get(i).getAve_temp()); return v; } </pre>
detail	<p>DataListAdapter.java에서 센서 데이터값들을 activity_data에 뿌려주는 역할을 한다.</p>

code	<pre> <com.github.mikephil.charting.charts.LineChart android:id="@+id/chart" android:layout_width="match_parent" android:layout_height="400dp"/> chart.getXAxis().setDrawGridLines(true); chart.getXAxis().setDrawAxisLine(false); chart.getXAxis().setEnabled(true); chart.getXAxis().setDrawGridLines(false); end Legend l = chart.getLegend(); l.setEnabled(true); l.setFormSize(10f); // set the size of the legend forms/shapes l.setTextSize(12f); l.setTextColor(Color.WHITE); YAxis leftAxis = chart.getAxisLeft(); leftAxis.setEnabled(true); leftAxis.setTextColor(getResources().getColor(R.color.colorAccent)); leftAxis.setDrawGridLines(true); leftAxis.setGridColor(getResources().getColor(R.color.colorAccent)); YAxis rightAxis = chart.getAxisRight(); rightAxis.setEnabled(false); </pre>
detail	<p>그래프를 생성한 layout부분과 그래프의 x축, y축을 설정한 코드이다.</p>

code	<pre> private final Handler handler = handleMessage(msg) → { if(msg.what == 0) { if(mArrayList.size() == i) { handler.removeMessages(what: 0); } else { double a = Double.parseDouble(mArrayList.get(i++)); addEntry(a); if (a >= 27) { sign++; if(sign >= 5) { value_fa = 0; soundplay(); showdiag(); } }else { sign = 0; } } if(data_count > 20) { chart.moveToX(xValue: data_count - 15); } } }; </pre>
detail	<p>쓰레드 핸들러 부분코드로 임의 상수로 쓰레드를 종료 또는 실행한다. 그리고 데이터가 일정 기준치를 넘어서면 경고음과 알림을 띄워준다. 그래프가 나오는 속도를 조정해준다.</p>

code	<pre> void soundplay(){ if(timer!=null){ timer.cancel(); timer.purge(); timer=null;} if (mp != null && mp.isPlaying()) { mp.stop(); mp.release(); mp = null; } mp = MediaPlayer.create(getApplicationContext(), R.raw.alram); mp.setOnPreparedListener(new MediaPlayer.OnPreparedListener() { @Override public void onPrepared(MediaPlayer mp) { mp.start(); } }); timer= new Timer(); timer.schedule(() -> { mp.stop(); }, delay: 80000); return; } </pre>
detail	<p>데이터 일정 기준치 넘어서면 울리는 알람 코드. 8초후에 종료하도록 설정</p>

code	<pre> class MyThread extends Thread{ Handler myhandler = handler; @Override public void run() { while(value_fa == 1){ myhandler.sendMessage(what: 0); try { Thread.sleep(millis: 100); }catch (InterruptedException e){ e.printStackTrace(); } } } } private void threadStart(){ final MyThread thread=new MyThread(); thread.setDaemon(true); thread.start(); } </pre>
detail	<p>쓰레드 본체 부분. value_fa 상수로 실행 또는 종료</p>

code	<pre> //그래프 통신 class GraphgroundTask extends AsyncTask<Void, Void, String> { String target; @Override protected void onPreExecute() { //List.php은 파싱으로 가져올 웹페이지 target = "http://13.59.96.134/tempData.php"; } @Override protected String doInBackground(Void... voids) {...} @Override protected void onProgressUpdate(Void... values) { super.onProgressUpdate(values); } @Override protected void onPostExecute(String result) { mJsonString=result; showGraph(); } } </pre>
detail	<p>데이터를 서버로 받아와 그래프에 출력하는 코드</p>

code	<pre> void showGraph(){ try{ JSONObject jsonObject=new JSONObject(mJsonString); JSONArray jsonArray=jsonObject.getJSONArray(name: "response"); int count=0; String value; while(count<jsonArray.length()){ JSONObject object=jsonArray.getJSONObject(count); value=object.getString(name: "value"); mArrayList.add(value); count++; } threadStart(); }catch (Exception e){ e.printStackTrace(); } } </pre>
detail	<p>액티비티가 넘어갈 때 그래프를 보여주는 코드</p>

code	<pre> void showdiag(){ AlertDialog.Builder builder= new AlertDialog.Builder(context this); builder.setTitle("Dangerous!").setMessage("Check and Stop machine!"); builder.setPositiveButton(text: "Yes", new DialogInterface.OnClickListener() { @Override public void onClick(DialogInterface dialog, int which) { Toast.makeText(getApplicationContext(), text: "Yes..",Toast.LENGTH_SHORT).show(); } }); builder.setNegativeButton(text: "No", new DialogInterface.OnClickListener() { @Override public void onClick(DialogInterface dialog, int which) { Toast.makeText(getApplicationContext(), text: "No..",Toast.LENGTH_SHORT).show(); } }); AlertDialog alertDialog = builder.create(); alertDialog.show(); } </pre>
detail	<p>데이터가 일정 기준치를 넘어서면 경고 메시지를 띄워주는 코드이다.</p>

code	<pre> @Override public boolean onOptionsItemSelected(@NonNull MenuItem item) { Intent intent; switch (item.getItemId()){ case R.id.temp: intent=new Intent(packageContext: GraphActivity.this,GraphActivity.class); startActivity(intent); return true; case R.id.rpm: intent=new Intent(packageContext: GraphActivity.this,Graph_rpm.class); startActivity(intent); return true; case R.id.prox: intent=new Intent(packageContext: GraphActivity.this,Graph_proximity.class); startActivity(intent); return true; case R.id.gap: intent=new Intent(packageContext: GraphActivity.this, Graph_gap.class); startActivity(intent); return true; default: return super.onOptionsItemSelected(item); } } </pre>
detail	<p>다른 센서 데이터를 선택하는 옵션 메뉴 코드 부분이다. 액티비티를 넘겨주면서 실행시킨다.</p>

code	<pre> Intent intent = getIntent(); listView = (ListView) findViewById(R.id.listView); dataList = new ArrayList<Data>(); adapter = new DataListAdapter(getApplicationContext(),dataList); listView.setAdapter(adapter); try{...}catch(Exception e){ e.printStackTrace(); } </pre>
detail	데이터를 서버와 인텐트를 통해 받아와 출력해주는 코드

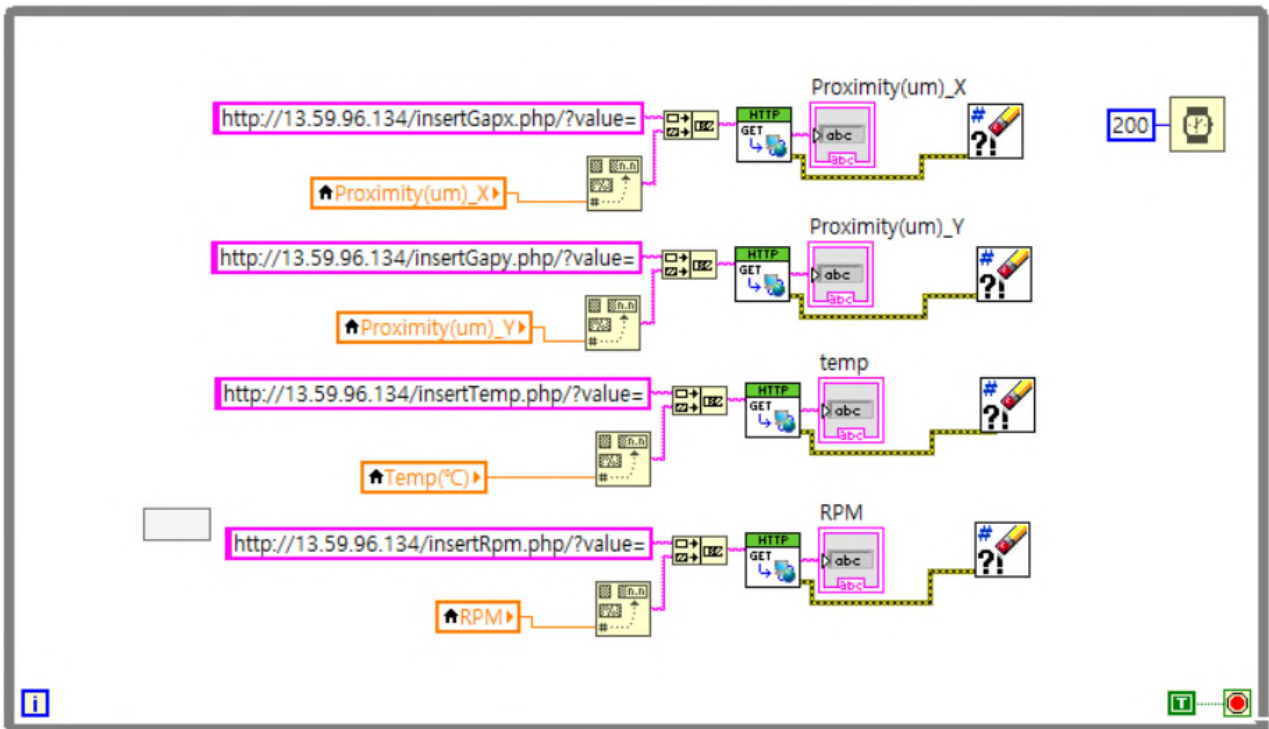
● 중요 코드(서버)

code	<pre> <?php \$con=mysqli_connect('13.59.96.134','test','1234','mysql','3306'); \$userID=\$_POST["userID"]; \$userPassword=\$_POST["userPassword"]; \$stmt=mysqli_prepare(\$con,"select userID, userPassword from operator where userID = ? and userPassword = ?"); mysqli_stmt_bind_param(\$stmt,"ss",\$userID,\$userPassword); mysqli_stmt_execute(\$stmt); mysqli_stmt_store_result(\$stmt); mysqli_stmt_bind_result(\$stmt,\$userID,\$userPassword); \$response=array(); \$response["success"]=false; while(mysqli_stmt_fetch(\$stmt)){ \$response["success"]=true; \$response["userID"]=\$userID; } echo json_encode(\$response); ?> </pre>
detail	userLogin.php 코드로 어플리케이션에서 입력받은 데이터들을 테이블에 있는 값들과 비교한다.

code	<pre> <?php \$con = mysqli_connect('13.59.96.134','test','1234','mysql','3306'); \$userID = \$_POST["userID"]; \$userPassword = \$_POST["userPassword"]; \$userName = \$_POST["userName"]; \$userPhone = \$_POST["userPhone"]; \$userEmail = \$_POST["userEmail"]; \$statement = mysqli_prepare(\$con, "INSERT INTO operator VALUES (?, ?, ?, ?, ?)"); mysqli_stmt_bind_param(\$statement, "sssss", \$userID, \$userPassword, \$userName, \$userPhone, \$userEmail); mysqli_stmt_execute(\$statement); \$response = array(); \$response["success"] = true; echo json_encode(\$response); ?> </pre>
detail	<p>userRegister.php로 어플리케이션에서 입력받은 값들을 데이터베이스에 삽입하여 회원가입한다.</p>

code	<pre> <?php \$con=mysqli_connect('13.59.96.134','test','1234','mysql','3306'); \$result=mysqli_query(\$con,"select * from gap_x"); \$response=array(); /*while(\$row=mysqli_fetch_array(\$result)){ array_push(\$response,array("ave_prox"=>\$row[0],"max_prox"=>\$row[1], "dis_prox"=>\$row[2],"jud_prox"=>\$row[3])); }*/ while(\$row=mysqli_fetch_array(\$result)){ array_push(\$response,array("value"=>\$row[0])); } echo json_encode(array("response"=>\$response)); mysqli_close(\$con); ?> </pre>
detail	<p>showGapX.php, showGapY.php, showRpm.php, showData.php와 동일하며, 센서 데이터들을 어플리케이션에 보내주는 코드</p>

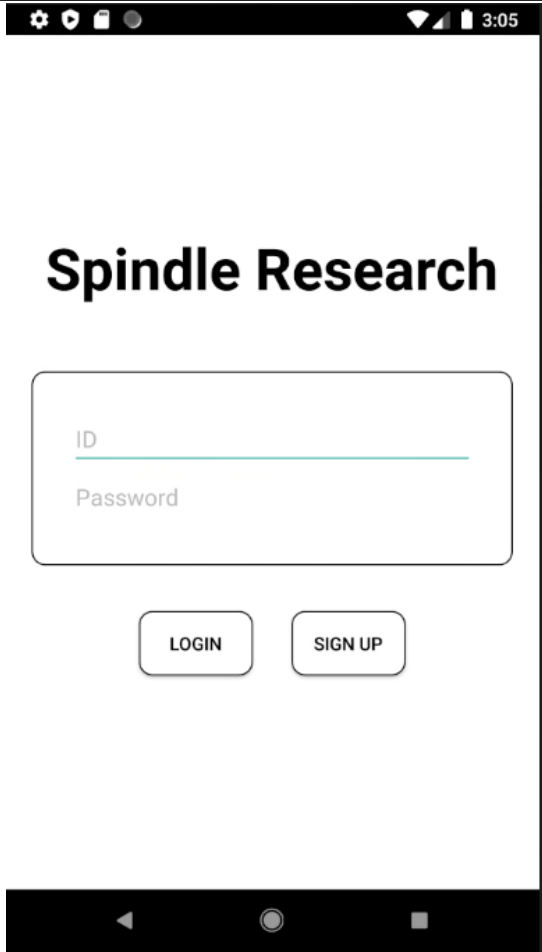
● 중요 코드 (랩뷰)

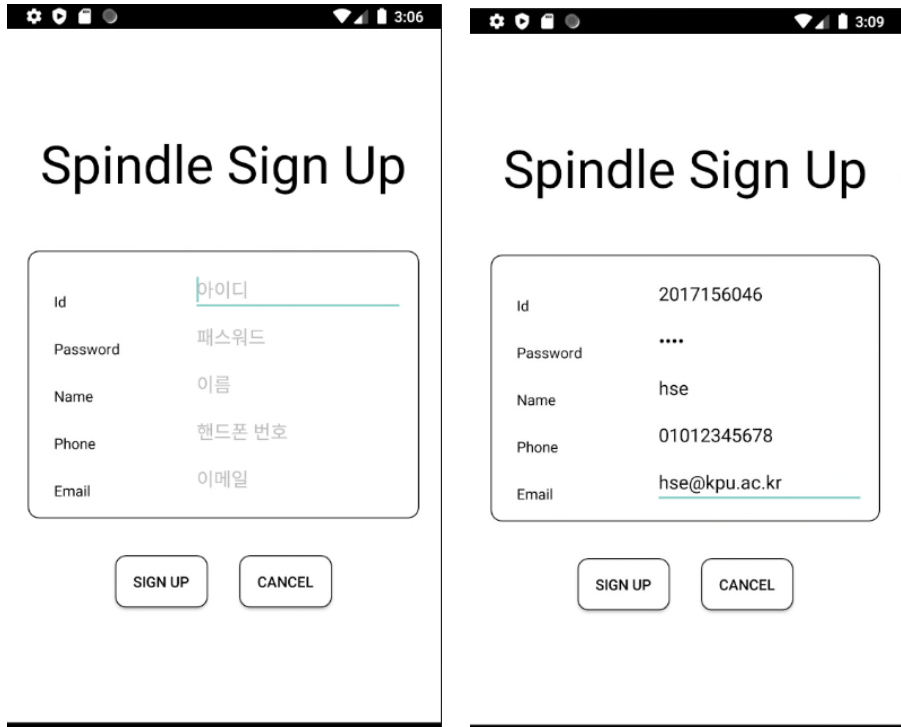
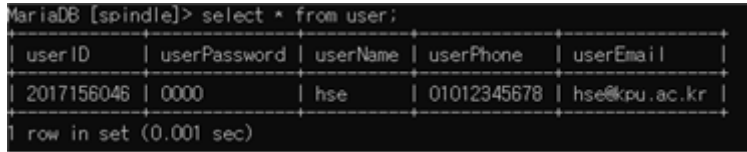


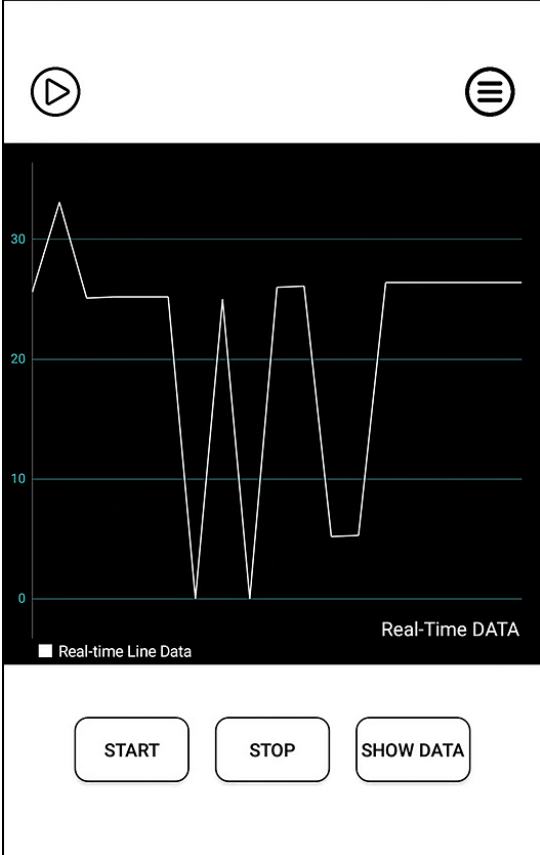
- 랩뷰 블록 다이어그램 부분으로 기계가 작동하는 동안 센서 데이터 값들을 서버에 넣어주는 과정이다. 우측 상단에 보면 시계 모양으로 된 블록을 보면 타이머를 설정하여 약 2초 간격으로 데이터들을 삽입하여 준다.

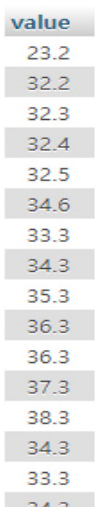
- 중간 블록들을 살펴보면 데이터 값들을 실수로 받아와 http 통신으로 get방식으로 데이터 통신을 한다. 서버 php 주소로 value 값들을 (센서 데이터 값)을 전송한다.

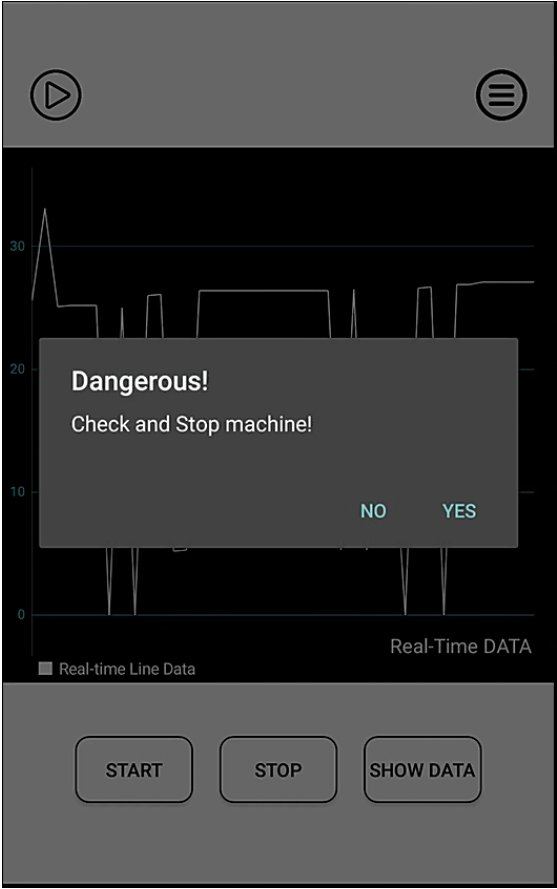
5. prototype

Action	Explanation
Main	 <p>- 어플리케이션 첫 화면으로 아이디와 비밀번호를 입력받아 로그인 또는 SIGN UP 버튼을 누를 수 있다.</p>
USER DATABASE ASE	<pre>MariaDB [spindle]> select * from user; Empty set (0.001 sec)</pre> <p>- 생성한 데이터베이스 테이블 user가 비어있다.</p>


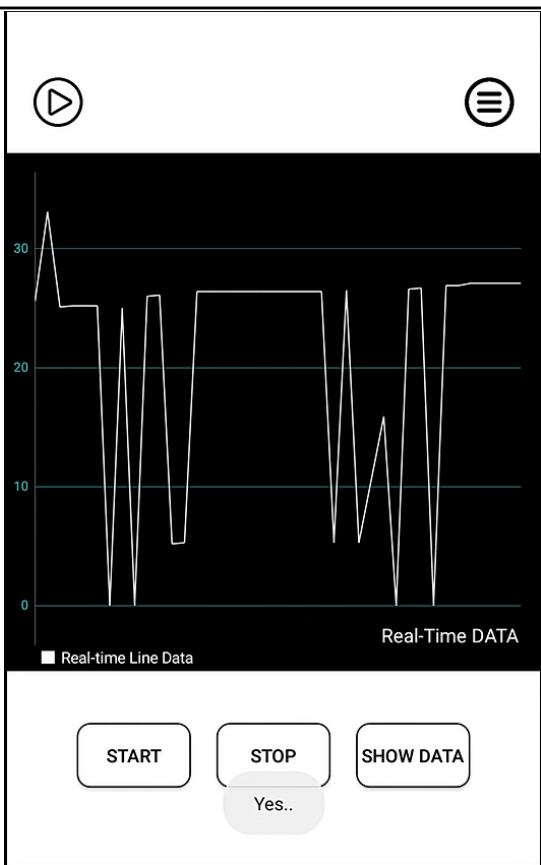
Action	Explanation
Sign	 <p>- 아이디 비밀번호 등 정보들을 입력하여 회원가입 버튼을 누른다.</p>
USER DATABASE	 <p>- 데이터베이스에 회원정보가 들어온 것을 확인 가능 하다.</p>

Action	Explanation
Graph	 <p>- 로그인하면 넘어가는 graph 액티비티로 상단엔 그래프 view와 하단엔 버튼 세개가 있다. show data를 누르면 상세 데이터값이 나오는 액티비티로 넘어간다.</p>

Action	Explanation
Data	 <p>- 센서 데이터가 서버로 받아와 저장되어있는 것을 확인할 수 있다.</p>

Action	Explanation
alert	 <p>- 데이터가 일정 기준치를 넘으면 dangerous 메시지가 발생하면서 기계를 확인하라고 요구한다.</p>

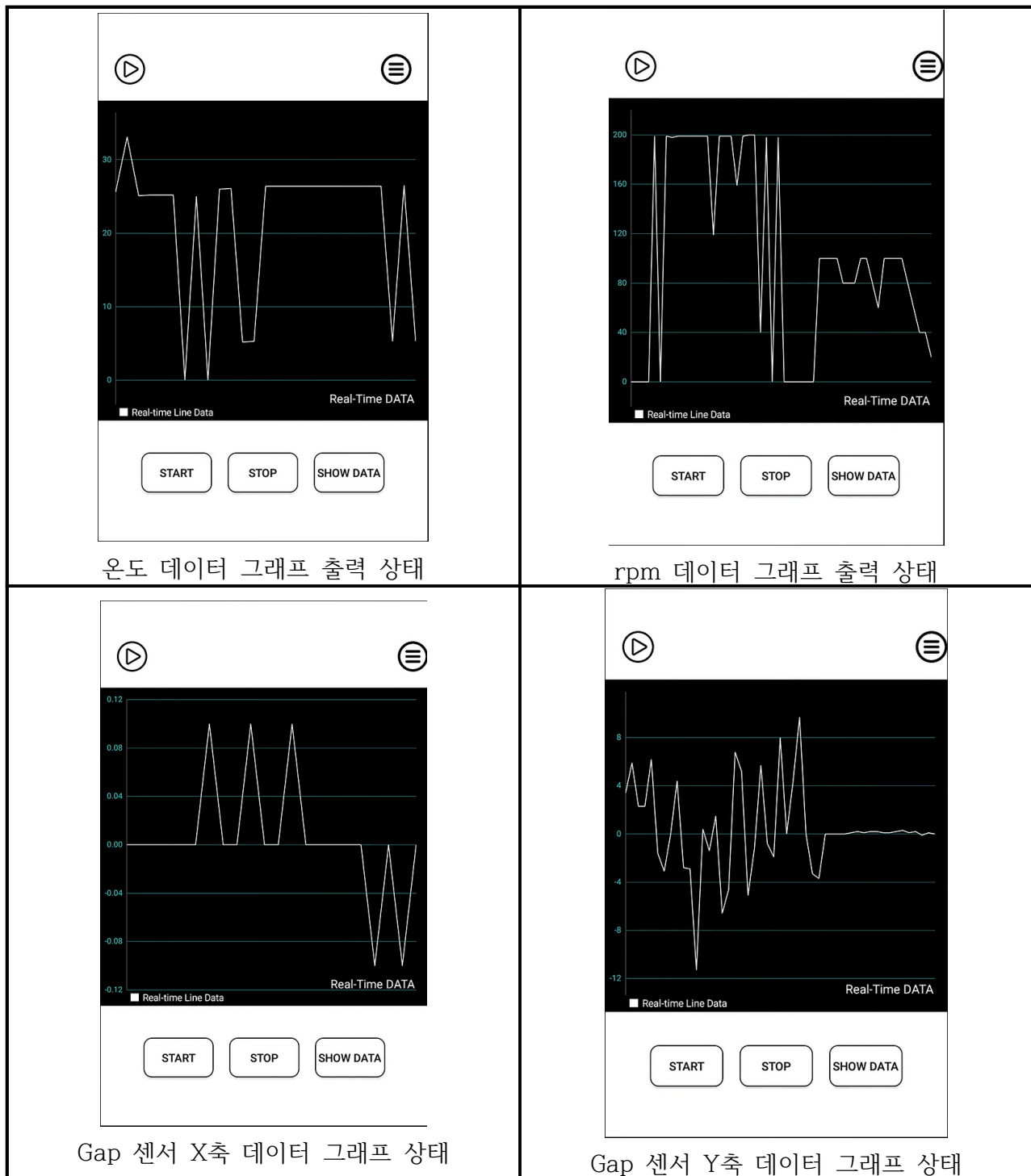
6. 시험/테스트 결과

Test	Result
spindle	
mobile	
result	<p>랩뷰 프로그램으로 데이터 난수를 출력하고 랩뷰와 local mysql 연결하였다.</p>

7. Coding / Demo

Action	Explanation
spindle	<div data-bbox="384 306 1058 1200" data-label="Image"> </div> <p data-bbox="384 1211 1359 1397">- 센서 설치 완료된 스펀들 기계로 하드웨어 장착상태로 센서 데이터 입력이 가능하다. 따라서 기계를 작동시키면 센서 데이터 값들이 랩뷰 패널로 들어와 보여줌과 동시에 데이터들을 서버로 전송하여 어플리케이션으로 보여준다.</p>

- 어플리케이션 데모



III. 결론

1. 연구 결과 및 기대효과

스마트팩토리(ICT 장비)는 정부의 주요 정책 중 하나로 큰 흐름을 이루고 있다. 기존의 개방형 CNC 제어기 기반의 공작기계 스마트화 기술을 고가의 시스템 구축비용, 낮은 확장성, 개별 CNC 컨트롤러(화낙, 지멘스 등)에 종속적이므로 기업이 기보유한 공작기계시스템에 대한 적용이 어렵고, 재활용성이 낮아 일반적인 스마트 공작기계시스템 구성에 어려운 상황이다. 그리하여 CNC 기계장비 산업이 발전되어 있으며 시장의 요구(생산 시스템 효율화, 무인 설비 운전)이 충분히 형성이 가능하다.

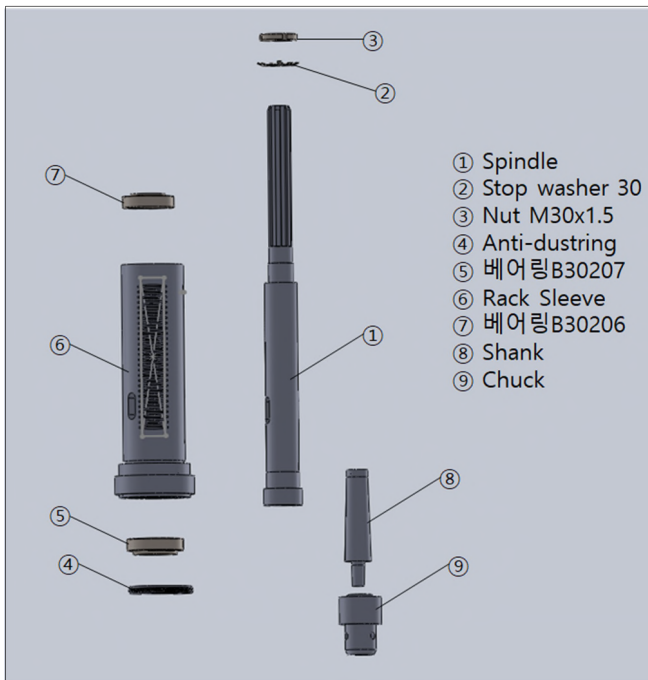
Smart Spindle은 생산 현장의 개념적인 효율화뿐만 아니라 실제 인건비를 낮추고 생산 효율을 높이는 실질적 시스템이다. 장비 상황을 파악하기위해 많은 인력이 24시간 현장 관리하는 현실과 가공이 정밀화되며 이에 대응하는 장비 가격이 상승하여 장비의 안정적 운영(상태 파악, 관리) 필요성이 증대 되면서 장비의 상황을 모니터링하여 효율적인 시스템 운영으로 생산성 향상을 이룰수 있다.

스핀들이 부착되어 있는 장비에 센서(온도, rpm, gap)가 설치 완료 후 LabView 프로그램으로 즉각적으로 볼 수 있는 창과 블록 다이어그램으로 센서 값이 프로그램에 들어오면 그래프와 창으로 확인이 가능하도록 구현하였다. 또한, 어플리케이션과 동일한 데이터를 확인이 가능하도록 LabView와 어플리케이션이 연동이 되어야 하는데 4-Layer(LabView-Server-Database-App)로 구축하였다. 블록 다이어그램에서 센서 데이터값을 문자열로 받아오게 처리하였다. http 프로토콜을 활용하여 데이터 값을 GET방식으로 전송하였으며, 미리 구축한 PHP 서버를 통해 데이터베이스에 저장하도록 설계하였다. 0.1초마다 출력되는 데이터값을 전부 다 저장하면 대량의 데이터로 무리가 생길 것을 대비하여 타이머를 설정하여 2초마다 서버로 보내게 하였으며, 어플리케이션에서도 동일하게 2초마다 출력하게 하였다. 랩뷰와 어플리케이션 동일하게 알람 또는 경고 신호는 일정 온도 기준치가 10초동안 유지되면 실행하도록 하였고, 어플리케이션에서 위험, 노멀 상태 구분은 하지않고 위험 상태만 경고하도록 하였다. 각 센서별로 데이터들을 확인하기 위해 그래프화면을 분류하였다.

향후 이 작품에 스마트화가 적용이 된다면 PC, APP 모니터링을 하지 않아도 장비의 프로그램이 스스로 판단하여 운전 또는 정지가 가능하여 인건비 절감 및 효율적 생산이 가능하다고 볼 수 있다. 또한 4차 산업혁명으로 스마트 팩토리 산업화가 증대되면서 공작기계 기술 측면이 HW / SW 적으로 발전할 수 있다고 기대한다.

2. 작품제작 소요재료 목록

1)



2)

센서	센서정보로부터 얻을 가공정보	효과
온도	베어링 온도, 코일(로터, 스테이터) 과열 감지	내구성
진동	런아웃의 변화, 외부 충격, 공구 파손 등 감지	교체시기, 정밀성
rpm	부하 변동,가공성능 등 감지	운전조건, 작업성
gap	축의 변형량 감지, 가공력 예측	정밀성

3) ASW EC2 Ubuntu

4) Database Mysql

참고문헌

- [1] Hwang Young-sik. Development of CNC Five-Axis High-Speed Processing System for Artificial Teeth and Implant Abutment. 2010
- [2] Amazon EC2. <https://aws.amazon.com/ko/ec2/>
- [3] USB-6000. <https://www.ni.com/en-us/support/model.usb-6000.html>
- [4] Chun In-guk. Android programming that is easily described in pictures. 2018
- [5] Welling Luke, Thomson Laura. PHP and MySQL Web development. 3rd ed. 2005
- [6] Eun Jong-min, Oh Jae-gon and Kim Jung-jun. Club management system based on Apache web server and Android app. , 18(2), 141-147. 2018.
- [7] Kwak Doo-young, Intermediate LabView, Lab View Education Center, 2016
- [8] Son Hye-young, LabView's Essence: Rescue, Seoul: Infinity Books, 2019
- [9] LabView to MySQL. <https://www.youtube.com/watch?v=EJQtEVddeWQ>