

보고서 / 품의서 / 제안서

작성자	하권용, 서동준
작성 일자	2014. 10. 30
분 류	<input checked="" type="checkbox"/> 기술 <input type="checkbox"/> 제안서 <input type="checkbox"/> 의견제출 <input type="checkbox"/> 구매요청 <input type="checkbox"/> 기타 <input checked="" type="checkbox"/>
제 목	SNC app 기술문서

SNC App 기술문서

본 문서는 파인 디지털의 SNC APP의 개선사항을 기록한 기술문서이다.

<Revision History>

	버전	내용	날짜	담당자
1	v1.0	초안	2014.10.29	하권용,서동준
2	v1.1	내용추가	2014.10.30	하권용,서동준
3	V1.2	내용추가	2014.10.30	하권용,서동준

<목차>

1. SNC app 기능.....	3P
1.1 사용자 등록.....	3P
1.2 블랙박스과 통신.....	3P
1.3 데이터 분석.....	3P
1.4 서버와 통신.....	3P
2. SNC app 주요 개선사항.....	4P
2.1 SD 카드 WIFI Search – Hotspot 전환 로직 재설계.....	4P
2.2 WIFI SD Card와의 통신 기능 재설계/구현.....	4P
2.3 서버관제 기능 재설계/구현.....	5P
2.4 보관함 기능 구현.....	5P
2.5 버그 수정 / UI 수정.....	5P
3. SNC app 구조.....	6P
3.1 클래스 목록 및 주요 기능.....	7P
3.2 SharedPreferences Table.....	8P
3.3 Service 부분 전체 구조도.....	9P
3.4 WIFI – Hotspot 관련 구조 및 설명.....	10P
3.5 File I/O와 서버관제 관련 구조 및 설명.....	12P
4. 향후 개선사항.....	16P

1. SNC app 기능

1.1 사용자 등록

사용자는 블랙박스의 시리얼/차량번호/연락처를 입력하여 app에 등록한다.

1.2 블랙박스와의 통신(app<->WIFISD)

app은 WIFI Direct를 이용하여 블랙박스의 SD카드에 접속하고 로그를 분석 영상파일을 다운로드 하여 로컬에 저장한다. (SD카드 Time split - 매 분당 0~20초는 블랙박스가 사용, 22~58초는 app이 사용하도록 설계되어 있음)

1.3 데이터 분석

app은 블랙박스로부터 받아온 로그 파일을 분석하여 사고(이벤트)일 경우 서버와 통신한다.

1.4 서버와 통신(app<->server)

app은 블랙박스로부터 받아온 로그 파일을 분석하여 사고(이벤트)일 경우, 서버와 소켓통신을 통해 사고를 접수하고, FTP를 이용하여 사고 영상파일을 업로드 한다.

2. SNC app 주요 개선사항

2.1 SD 카드 WIFI Search – Hotspot 전환 로직 재설계/구현

2.1.1 파일 I/O 관련 기능들을 분리 및 불필요한 핸들러를 삭제하고, WIFI를 Scan하여 WIFISD에 연결하는 동작을 수행하도록 재설계/구현하였다. 그리고, WIFISD에 연결되면 Hotspot 모드로 변경 되도록 기능을 추가하였다.

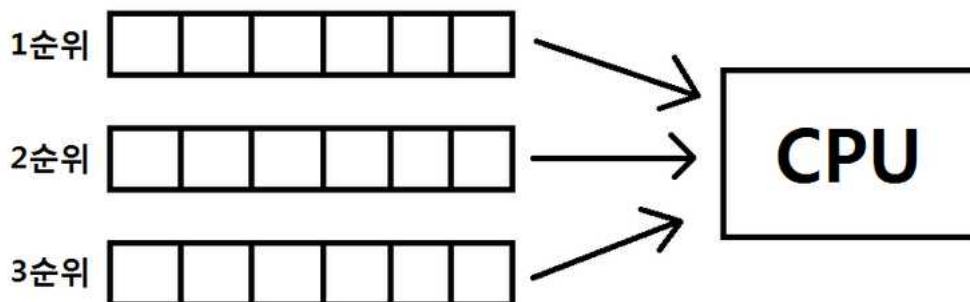
2.2 WIFI SD Card와의 통신 기능 재설계/구현

2.2.1 WIFI SD Card와 통신(파일 I/O 관련) 기능들을 WIFI Service로부터 분리해 dependency를 낮추고 스케줄러를 사용하여 작업들(로그파일 다운로드 / 영상 파일 다운로드 / 서버관제)이 꼬이지 않고 유기적으로 실행되도록 설계/구현하였다.

2.2.2 스케줄러 구현 시 고려했던 사항

- App에서 WIFI SD Card를 사용할 수 있는 시간이 제한적이다. (22초~58초)
 - 상기 사항으로 인하여 time deadline을 맞추기 위해 로그파일 다운로드와 영상파일 다운로드를 분리해야 했다.
- Android에서 AsyncTask 작업을 동시에 하나 이상 수행할 경우 수행시간과 성능을 보장(예측)할 수 없다.
 - 상기 사항으로 인하여 스케줄러는 크게 로그파일 다운로드 / 영상파일 다운로드 / 서버관제 3가지로 나누어 동시에 하나의 작업만을 수행하도록 구현하였다. 스케줄러 동작은 다단계 큐와 비슷한 방식으로 동작한다. (단, SNC app에 선점기능은 없다.) 우선순위는 v1.2.3버전을 기준으로 서버관제/서비스에 의한 영상파일 다운로드/사용자에 의한 영상파일 다운로드 / 로그파일 다운로드 순서이다.

1순위가 다 비어야 2순위가 실행되고 1,2순위가 다 비어야 3순위가 실행된다.



만약 2순위가 실행되는 중에 1순위준비큐에 프로세스가 등록되면 바로 선점된다.

이로 인한 '기아 현상' 이 발생할 수 있다.

<그림1. 다단계 큐 스케줄러 동작방식>

2.3 서버관제 기능 재설계/구현

2.3.1 서버관제 관련 기능들을 WIFI Service와 SocketTCPController로부터 분리해 dependency를 낮추고 동작 로직을 재설계하여 delay 없이 실행되도록 설계/구현 하였다.

2.4 보관함 기능 구현

2.4.1 메인 화면의 '이벤트영상'을 클릭하면 사용할 수 있는 보관함 기능들(사용자 다운로드, 영상 재생, 파일 삭제)을 구현하였다.

2.5 버그 수정 / UI 수정

2.5.1 App 동작 시 존재하던 각종 버그와 구현되지 않았던 UI를 구현/추가 하였다.

3. SNC app 구조

3.1 클래스 목록 및 주요 기능

패키지	클래스 이름	설명	기능
activity	BaseActivity	activity 인터페이스	
	BaseFragmentActivity	Fragment activity 인터페이스	
	IntroActivity	app 실행 시 스플래시 화면	
	MapViewActivity	지도를 보여주는 화면	
	MenuMainActivity	메인 화면	상단과 하단에 메뉴 구성 Fragment 관리
	ProgramInfoActivity	설정 - 프로그램 정보 화면	
	RegisterBBXActivity	블랙박스 시리얼 / 차량번호 등록화면	시리얼/차량번호 저장
	RegisterContactsActivity	연락처 등록화면	연락처 저장
	TermOfUseActivity	이용약관 화면	이용약관 출력
	TermOfUseDetailActivity	이용약관 상세내용 출력화면	이용약관 출력
	TermOfUseViewActivity	설정 - 이용약관 보기 화면	
	UpdateFirmWareActivity	업데이트 화면	
	VideoPlay	영상 재생화면	영상 재생
broadcast	BootReciever	리시버	안드로이드 부팅 시 Service 실행
dataset	ChuldongDataSet	출동정보가 들어가는 구조체 class	
	DataSetPromy	Promy 정보가 들어가는 구조체 class	
	DataSet	Log 파싱 결과가 들어가는 구조체 class	
fragment	BaseFragment	Fragment 인터페이스	
	FragmentASandInstall	하단메뉴 - 인포 - a/s	
	FragmentDefault	상단 - 홈(와이파이 신호가 깜박거리는)	
	FragmentEventVideo	하단메뉴 - 이벤트 영상	
	FragmentInfo	하단메뉴 - 인포	
	FragmentMobilization	하단메뉴 - 긴급출동	
	FragmentPromyWorld	하단메뉴 - 인포 - a/s - 프로모월드 찾기	
	FragmentSetting	상단메뉴 - 설정	
	FragmentUserGuide	하단메뉴 - 사용자가이드	

	FragmentWebView	하단메뉴 - 인포 - 스마트앤케어+	
serviceRun	WIFISearch		Wifi 검색 / 접속
	RunDeviceSearch		Hotspot 접속기기 검색
services	WIFIService	Service	WIFI 셋팅
	FileControlService	Service	블랙박스로부터 영상리스트, 로그리스트, 로그파일, 영상 파일 다운로드 로그파일 파싱/필요 시 서버관제 시작
controller	ControllerHotspot		Hotspot 사용
	ControllerNotification		Notification 사용
	ControllerWiFi		Wifi 사용
Server	ServerConnection	AsyncTask	서버와 소켓통신 / 영상 파일 업로드
	PacketFormat		Packet data format 설정
utils	BackPressCloseHandler		Back 버튼 리스너
	CarNameUtil		차량번호 파서
	CustomEventListAdapter		이벤트 영상 리스트 구성
	CustomPromyListAdapter		프로미 월드 리스트 구성
	DaumMapRender		지도 이미지 렌더링
	ChangeConnectMode	WIFISD에 Hotspot모드변경 커맨드 보냄	WIFI연결 후 tm파일 업로드, 모드변경 커맨드 보냄, 폰 hotspot 켜
	LogfileControl		로그 파일 생성 / 로그 기록

3.2 SharedPreference Table

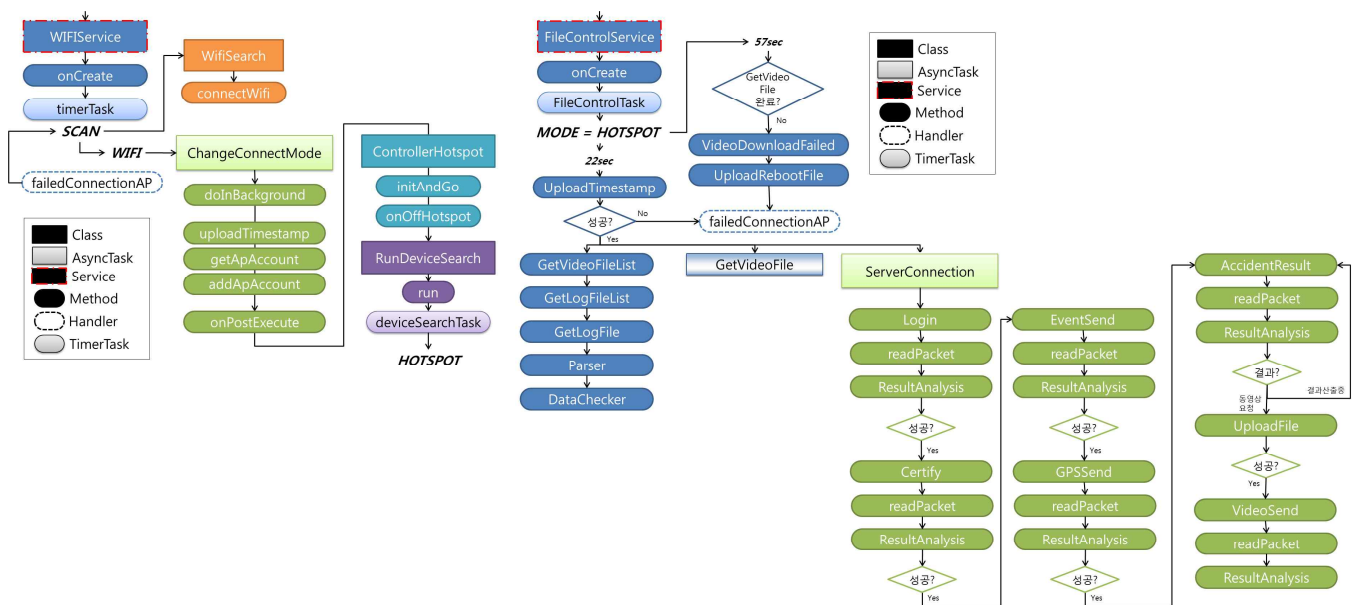
SNC Application 내부에서 사용하는 SharedPreference Table이다. SharedPreference는 간단한 String 정도의 데이터를 전역변수처럼 이용하고 싶을 때 사용한다.

Directory	Key	Value	삽입시점	삭제시점
pref	termOfUse	yes/no	이용약관 동의 시	
pref	bbxRegister	yes/no	블랙박스 등록 시	
pref	bbxserial	블랙박스 시리얼	블랙박스 등록 시	
pref	carnumber	차량번호	블랙박스 등록 시	
pref	contatcRegister	yes/no	연락처 등록 시	
pref	contact0	연락처1	연락처 등록 시	
pref	contact1	연락처2	연락처 등록 시	
pref	contact2	연락처3	연락처 등록 시	
pref	ap_mode	scan/wifi/hotspot	Scan - AP 검색 시 Wifi - hotspot 전환 시 Hotspot - SD Card가 붙었을 시	WIFIService 종료 시
pref	firmware	로그파일의 펌웨어 정보	첫 로그파일 다운로드 시	
pref	rms	(Default)0.3	(테스트용) rms threshold 값	
pref	rmsserver	(Default)1.0	(테스트용) rms 서버전송 배율 값	

<표1. SharedPreference Table>

3.3 Service 부분 전체 구조도

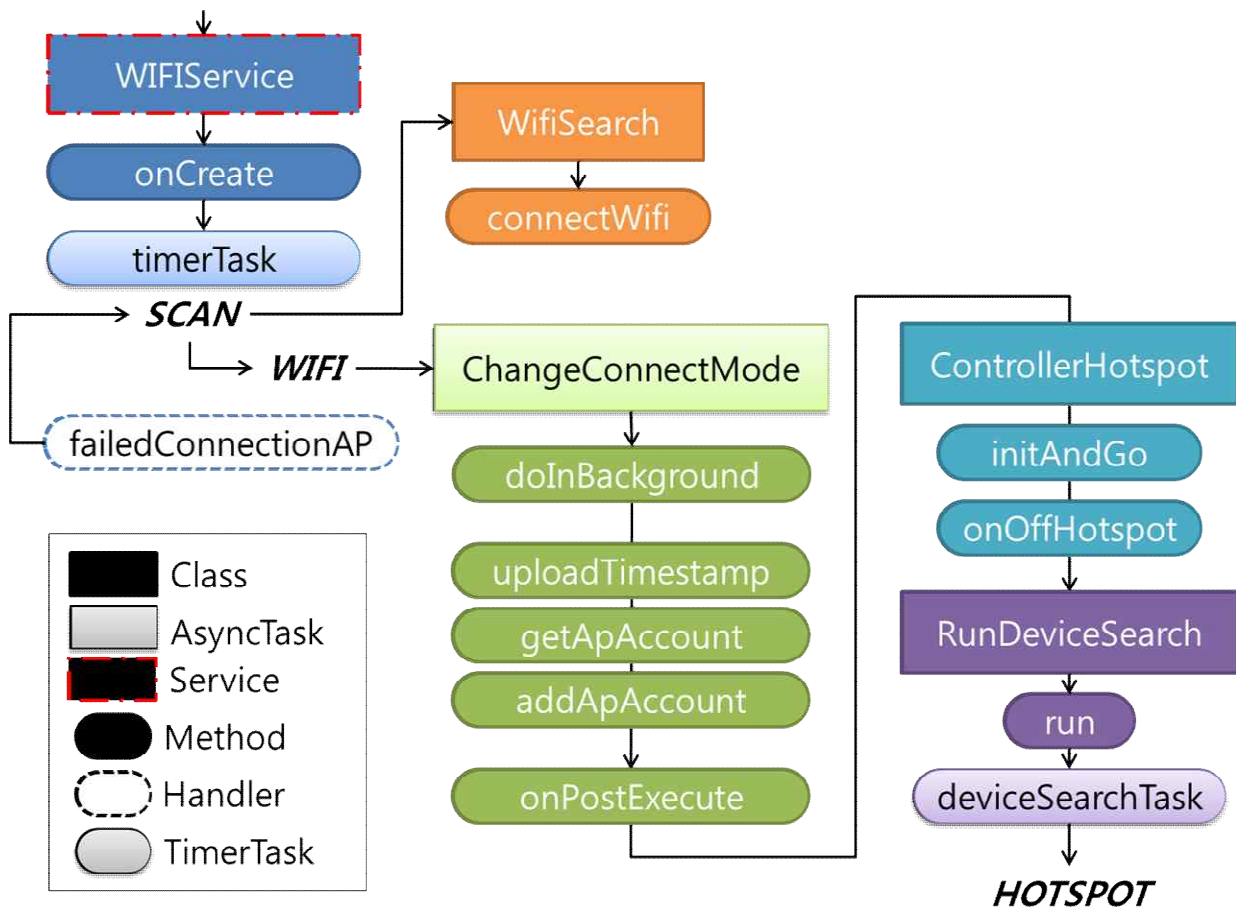
주로 개선된 Service부분의 전체적인 구조도 이다. 자세한 설명은 3.4, 3.5에서 다룬다.



<그림2. Service 부분 구조도>

3.4 WIFI – Hotspot 관련 구조 및 설명

3.4.1 [WIFIService]



<그림3. WIFI – Hotspot 관련 구조도>

- WIFIService의 실행 : 부팅 시 또는 앱 실행 시
- WIFIService의 종료 : 설정 – 서비스 종료

- ap_mode에 따른 동작

scan : 서비스 시작 시 또는 연결실패 시에 scan모드가 되며, wifi를 스캔하여 WIFISD를 찾아 연결을 시도한다. WIFISD에 wifi로 연결되면 wifi모드로 전환하고 ChangeConnectMode를 호출하여 hotspot으로 연결을 시도한다.

wifi : WIFISD에 wifi로 연결되고 나서부터 hotspot으로 연결을 시도할 때까지 wifi모드이며, WIFIService에서는 아무런 동작도 하지 않는다.

hotspot : hotspot에 WIFISD가 붙은 이 후의 모드로, WIFIService에서는 아무런 동작도 하지 않는다.

- 동작 흐름 설명

WIFIService가 실행되면 scan모드가 되고 3초 주기로 timerTask가 동작한다. scan모드에서 WifiSearch클래스를 통해 WIFISD에 연결되었는지 검사하여 연결되었다면 wifi모드로 전환하고 ChangeConnectMode를 실행한다.

ChangeConnectMode는 AsyncTask로 doInBackground에서 tm파일을 업로드하고 SD카드에서 핫스팟 계정을 읽어 들인다. tm파일 업로드에 실패하면 scan모드로 돌아간다. SD카드에 핫스팟 계정이 설정되어 있지 않다면, "smartncare"를 아이디로, 등록한 블랙박스의 시리얼 번호를 패스워드로 사용하도록 SD카드에 설정한다. 그 후에 SD카드에 핫스팟 전환 명령을 전송한다. doInBackground에서 핫스팟 전환 메시지 전송까지 한 후 onPostExecute에서 ControllerHotspot클래스의 함수를 사용하여 폰의 핫스팟 옵션을 설정하고 켜다. 그리고, RunDeviceSearch의 run함수를 호출하여 deviceSearchTask 타이머를 동작시켜 1분동안 SD카드가 핫스팟으로 연결되기를 기다린다. arp테이블에 연결된 SD카드의 정보가 생성되면 간단한 mac주소 검사와 ping을 체크하여 정상적으로 연결이 되었는지 확인한 후에 hotspot모드로 전환된다. 1분안에 연결이 되지 않거나 핫스팟을 끄면 scan모드로 돌아간다.

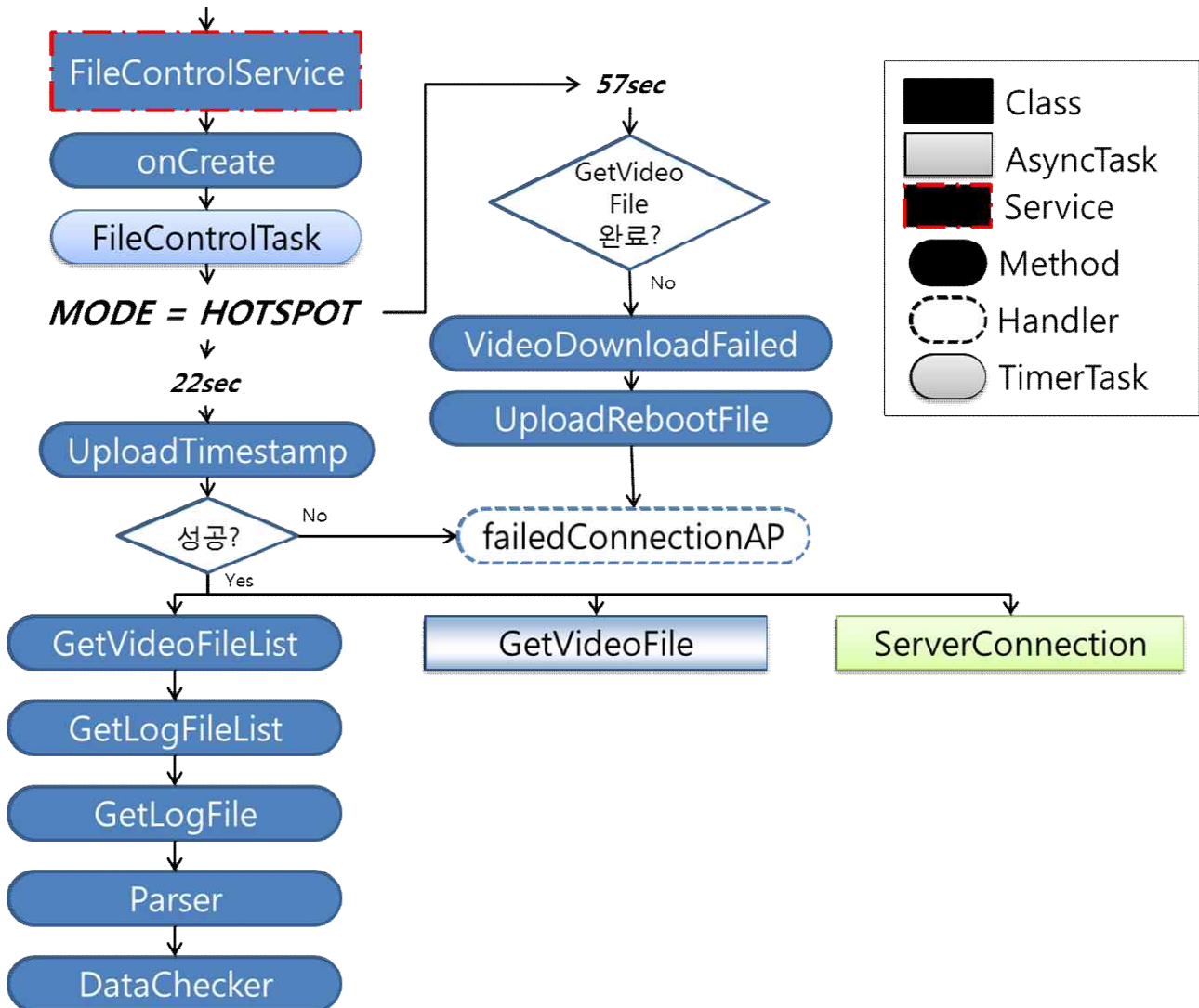
scan모드로 돌아갈 때는 failConnectionAP핸들러를 호출하도록 하였다. failConnectionAP핸들러에는 다시 scan모드로 동작하는데 필요한 설정들을 구현하였다.

3.5 File I/O와 서버관제 관련 구조 및 설명

3.5.1 [FileControlService]

android service 클래스이며 SD Card<->app 사이의 파일다운/업로드 관련 기능을 수행한다.

● 구조도



<그림4. File I/O와 서버관제 관련 구조 및 설명>

FileControlService 는 HOTSPOT모드(WIFI SD Card가 휴대폰 테더링에 접속 완료된 상태)에서만 동작하며 파일 업/다운로드 기능을 수행한다. 매분 22초에 timestamp파일을 SD Card로 업로드 하여 성공 스케줄러를 통해 다음동작을 수행하게 되고 실패 시 ap검색 모드로 전환한다. 스케줄러는 3가지 작업 중 하나(비디오파일리스트-로그파일리스트-로그파일 다운로드 / 영상파일 다운로드 / 서버관제)를 선택하여 수행하게 된다. (스케줄러를 사용한 이유는 4P 2.2.2 참조)

- 기능설명

FileControlTask() – 서비스에서 지속적으로 동작하는 Timer로써 초단위로 시간을 검사하여 동작을 수행한다. Ap_mode가 "hotspot" 상태일 때에만 동작한다. (ap_mode 변경은 WiFiService에서 수행된다.) 매분 22초에 timestamp파일을 업로드 하여 파일 업로드 성공 시 3가지 동작 중 하나(GetVideoFileList / GetVideoFile / ServerConnection)를 선택해 수행한다.

GetVideoFileList() – SD CARD에서 비디오 파일 리스트를 불러와 저장한다. (약 1sec 소요)

GetLogFileList() – SD CARD에서 로그 파일 리스트를 불러와 저장한다. (약 1sec 소요)

GetLogFile() – SD CARD에서 로그 파일을 다운로드 받는다. (로그파일당 0.5~1sec 소요)

Parser() – SD CARD에서 다운받은 로그 파일을 줄(line) 단위로 파싱하여 저장한다.

DataChecker() – 파싱된 로그데이터를 검사하여 사고존재여부를 판단하고 사고가 존재할 경우 사고에 해당하는 영상 파일 다운로드를 예약한다.

GetVideoFile() – SD CARD에서 영상 파일을 다운로드 받는다. Queue에 영상파일 다운로드 작업이 들어 있을 경우에만 수행된다. 20~30초 소요

ServerConnection – 서버관제 AsyncTask를 시작한다.

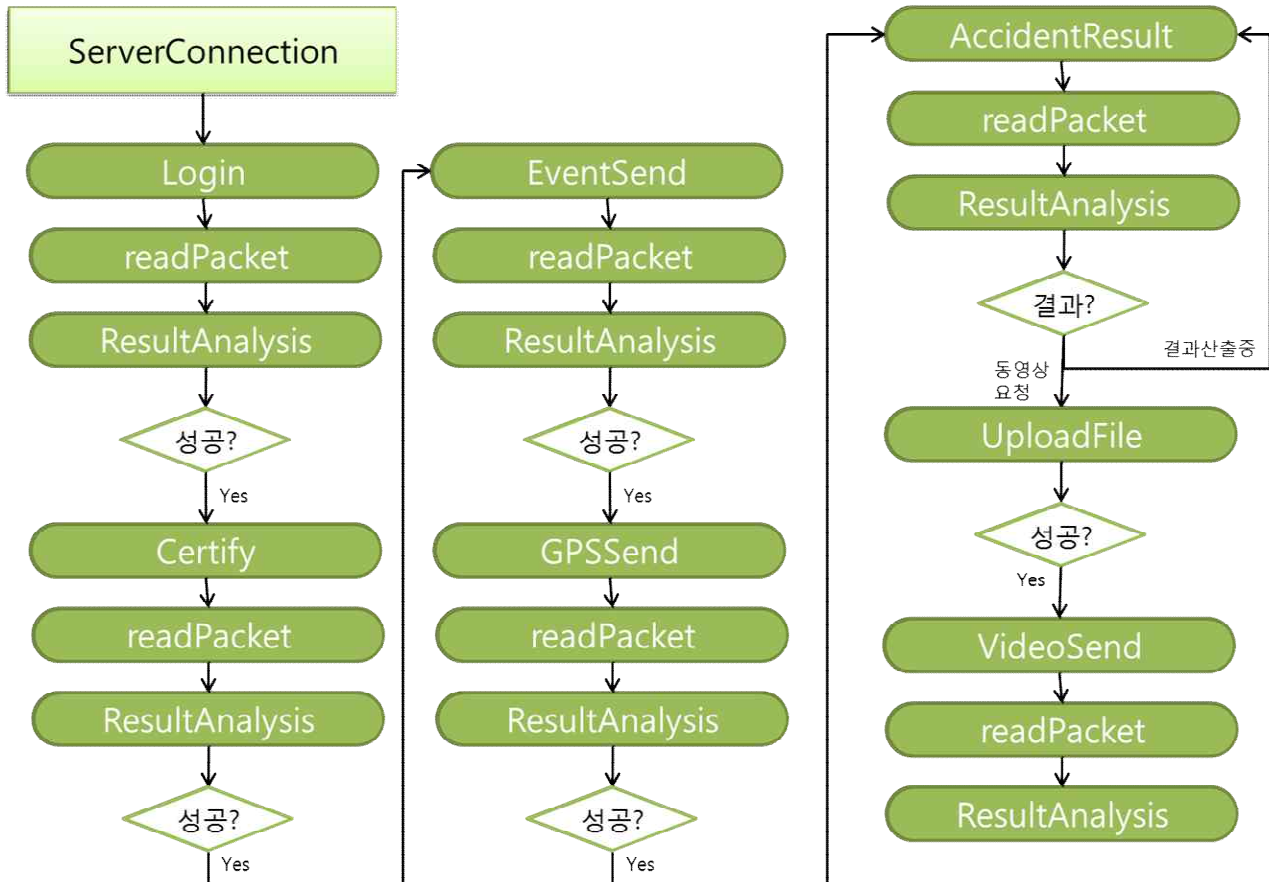
UploadTimeStamp() - yyyy_MM_dd_HH_mm_ss.tm 파일을 업로드 한다.

VideoDownloadFailed() – WIFI SD Card의 전송속도가 느려지는 현상을 처리하기 위한 예외코드로써, 영상 파일 다운로드가 57초에 완료되지 않았을 경우 실행된다. 사용자에게 의한 다운로드 라면 사용자 다운로드 dialog를 없애는 동작을 수행, 다운로드중인 파일을 삭제하고 reboot.tm파일을 업로드, app을 ap search모드로 전환한다.

3.5.2 [ServerConnection]

AsyncTask 클래스이며 소켓을 통한 서버통신, FTP를 통한 영상 파일 업로드 기능을 수행한다.

● 구조도



<그림5. 서버관제 구조도>

ServerConnection은 순서상 로그파일 파싱 후 사고를 확인, 영상파일을 다운받은 후에 실행되며 서버와의 소켓통신을 통해 사고를 접수하고 영상파일을 올리는 기능을 수행한다. 사고결과가 '결과 산출 중' 으로 리턴될 경우 5회 retry하는 기능이 추가되었다. (v1.2.2)

- 기능 설명

Login() – Login body를 생성하고 Packet화(PacketFormat 클래스 이용)하여 소켓을 통해 서버로 전송한다. 소켓 전송 결과는 readPacket()을 사용하여 읽는다.

Certify() – Certify body를 생성하고 Packet화(PacketFormat 클래스 이용)하여 소켓을 통해 서버로 전송한다. 소켓 전송 결과는 readPacket()을 사용하여 읽는다.

EventSend() – EventSend body를 생성하고 Packet화(PacketFormat 클래스 이용)하여 소켓을 통해 서버로 전송한다. 소켓 전송 결과는 readPacket()을 사용하여 읽는다.

GPSSend() – GPSSend body를 생성하고 Packet화(PacketFormat 클래스 이용)하여 소켓을 통해 서버로 전송한다. 소켓 전송 결과는 readPacket()을 사용하여 읽는다.

AccidentResult() – AccidentResult body를 생성하고 Packet화(PacketFormat 클래스 이용)하여 소켓을 통해 서버로 전송한다. 소켓 전송 결과는 readPacket()을 사용하여 읽는다.

VideoSend() – VideoSend body를 생성하고 Packet화(PacketFormat 클래스 이용)하여 소켓을 통해 서버로 전송한다. 소켓 전송 결과는 readPacket()을 사용하여 읽는다.

UploadFile() – 사고 영상 파일을 FTP를 사용하여 업로드 한다. 파일명을 변경하여 업로드 한다. 파일 업로드 중에는 해당 사용자가 보관함을 통해 재생/삭제를 할 수 없도록 예외처리를 하였다.

readPacket() – 받은 패킷을 읽어 파싱한다. 파싱결과는 ResultAnalysis()로 전달한다.

ResultAnalysis() – 받은 패킷 내용 중 응답코드를 분석하여 응답코드 결과에 따라 다음 작업을 수행한다.

4. 향후 개선사항

4.1 사고관제 로직 변경 기능 추가?

블랙박스의 사고 발생 시점부터 서버관제가 완료되기까지의 시간이 오래 걸린다는 의견이 있어 로직을 수정하려는 시도가 있었으나 일정상 구현되지는 않았다.

현재 로직 : 블랙박스로부터 로그다운 - 파싱 - 사고확인 - 사고영상 다운 - 서버관제 - 사고영상업로드

변경 시 : 블랙박스로부터 로그다운 - 파싱 - 사고확인 - 서버관제 - 사고영상 요청 시 사고영상 다운 - 서버관제 - 사고영상업로드

변경 시 예상되는 이슈

1. 1차적으로 사용자 다운로드 기능에 영향을 주게 된다. 서버 관제 프로세스 도중에 사고영상 다운로드가 끼어들면서 사용자 다운로드가 언제 시작/완료될지 모르게 됨.
2. 현재는 WIFI SD Card 통신부분과 서버관제 부분이 분리되어 있는데 서버관제 도중 wifi sd card와 통신을 해야 하므로 상호 모듈간에 dependency가 생겨 로직변경과 예외처리를 추가해야 하며 사이드 effect도 테스트해야 함.