

『거동이 불편한 고령환자를 위한
스마트침대시스템 개발』

Manual

교과목	설계및프로젝트 심화 II
담당교수	강병익, 조용석 교수님
조명	2조
학과	의료IT공학과
팀장	15615002 강현욱
팀원	15615013 김준혁 15615042 이건호
연락처	lego7407@naver.com
제출일	2019.09.23.(월)
문서버전	1.0.0



Manual

Manual Table ▾

Manual Number	Document Version	담당자	Tags	Last Edited Time
1.1_시스템_동작	Ver 1.0.0	현욱 강 건호 이 준혁 김	Start_Program	9월 22, 2019 4:14 오후
1.2_시스템_사양	Ver 1.0.0	현욱 강 건호 이 준혁 김	System Specification	9월 22, 2019 2:50 오후
1.3_시스템_BlockDiagram	Ver 1.0.0	현욱 강 건호 이 준혁 김	System BlockDiagram	9월 22, 2019 4:22 오후
1.4_시스템_FlowChart	Ver 1.0.0	현욱 강 건호 이 준혁 김	System FlowChart	9월 22, 2019 3:05 오후
1.5_시스템_학술자료분석	OPEN	현욱 강 건호 이 준혁 김	System Thesis Magazine	9월 22, 2019 3:12 오후
1.6_시스템_시연영상	Ver 1.0.0	현욱 강 건호 이 준혁 김	영상 Testing	9월 22, 2019 5:16 오후
2.1_HX711	Ver 1.0.0	건호 이	HX711	9월 22, 2019 2:08 오후
2.1.1_HX711_AD Converter	Ver 1.0.0	현욱 강	HX711 ADC	9월 22, 2019 4:20 오후
2.1.2_HX711_AMP	Ver 1.0.0	건호 이	HX711 AMP	9월 22, 2019 4:27 오후
2.2_Atemega328p	Ver 1.0.0	건호 이	Atemega328p	9월 22, 2019 2:17 오후
2.2.1_Atemega328p_Firmware	Ver 1.0.0	현욱 강 건호 이	Atemega328p Firmware	9월 22, 2019 4:30 오후
2.3_Loadcell	Ver 1.0.0	건호 이	Loadcells	9월 22, 2019 2:29 오후
2.3.1_Loadcell_spec	Ver 1.0.0	건호 이	Loadcells Spec	9월 22, 2019 4:35 오후
2.3.2_Loadcell_종류별차이	Ver 1.0.0	건호 이	Loadcells 종류별차이	9월 22, 2019 5:12 오후
2.4_의료용침대_규격	Ver 1.0.0	현욱 강	의료용침대 규격	9월 22, 2019 4:36 오후
2.5_RaspberryPi_Camera	Ver 1.0.0	현욱 강	RaspberryPi Camera	9월 22, 2019 2:48 오후
3.1_PC프로그램	Ver 1.0.0	준혁 김	C# Windows Summary	9월 22, 2019 4:13 오후
3.1.1_PC프로그램_DFD	Ver 1.0.0	준혁 김	C# Windows DFD	9월 22, 2019 4:15 오후
3.1.2_PC프로그램_데이터수집	Ver 1.0.0	준혁 김	C# Windows DataRead	9월 22, 2019 3:22 오후
3.1.3_PC프로그램_Tare	Ver 1.0.0	준혁 김	C# Windows Tare	9월 22, 2019 3:32 오후
3.1.4_PC프로그램_UnitTare	Ver 1.0.0	준혁 김	C# Windows UnitTare	9월 22, 2019 3:33 오후
3.1.5_PC프로그램_Measure	Ver 1.0.0	준혁 김	C# Windows Measure	9월 22, 2019 3:33 오후
3.1.6_PC프로그램_영상출력	Ver 1.0.0	준혁 김	C# Windows Monitoring	9월 22, 2019 4:21 오후
3.1.7_PC프로그램_Loadcells_Chart	Ver 1.0.0	준혁 김	C# Windows Loadcells	9월 22, 2019 5:16 오후
3.1.8_PC프로그램_Weight_Chart	Ver 1.0.0	준혁 김	C# Windows Weight	9월 22, 2019 5:16 오후
3.1.9_PC프로그램_COP_Chart	Ver 1.0.0	준혁 김	C# Windows COP	9월 22, 2019 5:16 오후
3.2_침대촬영	Ver 1.0.0	현욱 강	침대촬영	9월 22, 2019 3:45 오후
3.2.1_침대촬영_라즈베리파이카메라연결	Ver 1.0.0	현욱 강	침대촬영 Camera	9월 22, 2019 5:13 오후
3.3_Database	Ver 1.0.0	현욱 강	Database	9월 22, 2019 5:03 오후
3.3.1_Database MySql	Ver 1.0.0	현욱 강	Database MySQL	9월 22, 2019 4:34 오후
3.3.2_Database_Workbench	Ver 1.0.0	현욱 강	Database MySQL Workbench	9월 22, 2019 4:37 오후
3.3.3_Database_phpMyAdmin	Ver 1.0.0	현욱 강	Database phpMyAdmin	9월 22, 2019 4:33 오후
3.3.4_Database_TCP_데이터전송형식	Ver 1.0.0	현욱 강	Database TCP	9월 22, 2019 5:13 오후
3.3.5_Database_Camera연동	Ver 1.0.0	현욱 강	Database Camera	9월 22, 2019 4:10 오후

1.1_시스템_동작

Created Time	@Sep 18, 2019 3:56 PM
Document Version	Ver 1.0.0
Last Edited By	👤 현욱 강
Last Edited Time	@Sep 22, 2019 4:14 PM
Tags	Start_Program
담당자	👤 현욱 강  건호 이  준혁 김

시스템

동작

1. 큰 깡통에 전원케이블(AC220V→DC12V) 연결
2. 네 개의 로드셀 케이블을 큰 깡통에 장착 - 1번:UpLeft, 2번:DownLeft, 3번:DownRight, 4번:UpRight
3. 시리얼통신을 위한 미니깡통을 큰 깡통의 out에 삽입하고, 케이블을 PC에 아두이노 포트연결하듯이 삽입
4. PC프로그램을 실행하고 시리얼설정 창에서 포트설정함 - 기본적으로 포트만 설정하고 나머지는 안 건들임
5. 침대가 있는 주변환경이 바뀌었다면 Tare와 Unit Tare를 해야만 해당 사항은 3.1_PC프로그램 참고
6. 그렇지 않다면 버튼을 눌러 원하는 모드로 변경

영상참조

1.1_시스템_동작준비1.mp4

1.1_시스템_동작준비2.mp4

UL = 1번

전원 = 12V

DL = 2번

OUT = USB연결 PORT

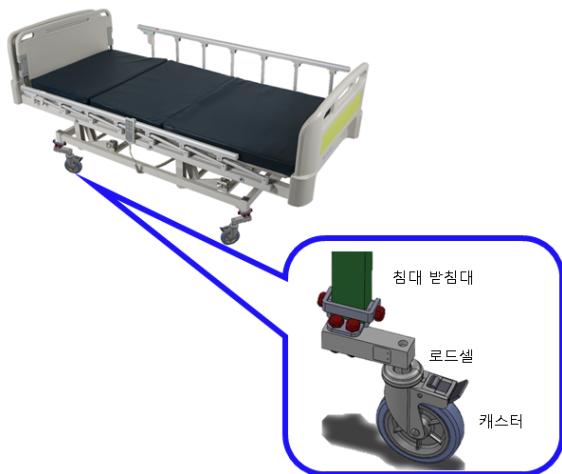
DR = 3번

- 각 로드셀은 침대에 연결

UR = 4번

- 바퀴로 높이 조절

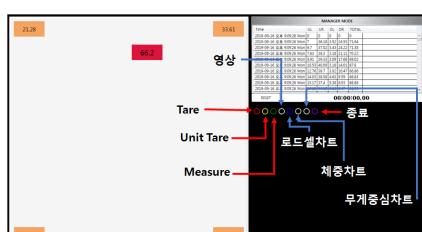
시스템 HW



4개의 로드셀로 무게중심과 침대 위치를 측정하여 거동이 불편한 고령환자를 대상으로 낙상알림 및 욕창방지를 위한 모니터링 시스템

침대 바퀴에 로드셀을 장착하여 침대에서의 사용자 체중 및 무게중심, 자세, 낙상 등의 분석을 위한 기초데이터를 제공

시스템 SW



무게값, 체중, 무게중심을 사용하여 차트를 그리거나 무게중심 캔버스 기능 제공

DB와 영상을 위해 라즈베리파이를 별도로 사용하며 PC프로그램에 실시간으로 영상을 전송

1.2_시스템_사양

Created Time	@Sep 19, 2019 10:56 PM
Document Version	Ver 1.0.0
Last Edited By	👨‍💻 현욱 강
Last Edited Time	@Sep 22, 2019 2:50 PM
Tags	Specification System
담당자	👨‍💻 현욱 강 HH 건호 이 준혁 김

시스템

사양

Bed System

시스템 구성	번호	Tool	언어	비고
<u>MCU (ATmega328P)</u>	1	Atmel Studio	C++	센서와 연결
<u>모니터 프로그램</u>	2	Visual Studio	C#	시각적 확인
<u>라즈베리 파이</u>	3	리눅스 OS	Python	카메라 제어
<u>스트리밍서버</u>	4	Mjpg-Streamer	-	CCTV 실행 시, 영상 실행
<u>DB</u>	5	MySQL	sql	환자 정보 저장 및 사진, 영상 저장

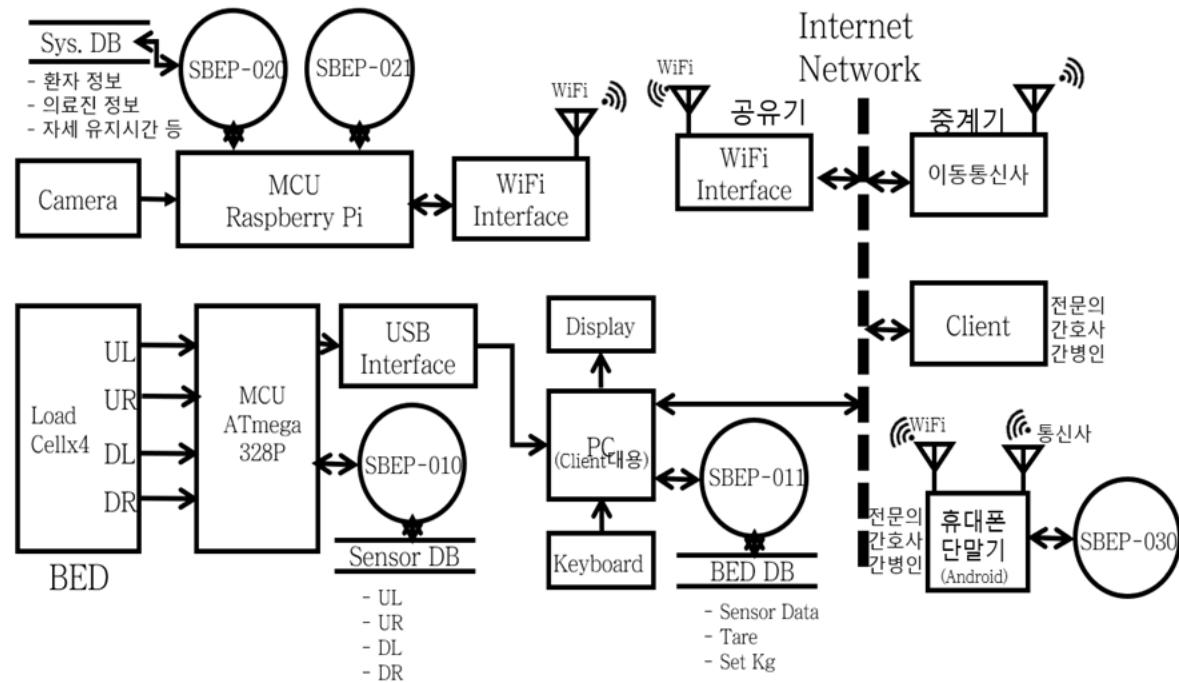
1.3_시스템_BlockDiagram

Created Time	@Sep 21, 2019 4:42 PM
Document Version	Ver 1.0.0
Last Edited By	 현욱 강
Last Edited Time	@Sep 22, 2019 4:22 PM
Tags	BlockDiagram System
담당자	 현욱 강  건호 이  준혁 김

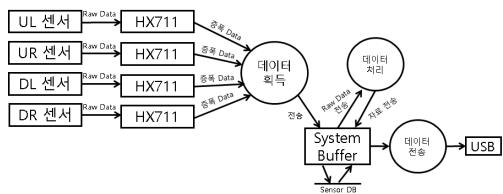
시스템

BlockDiagram

SBEP



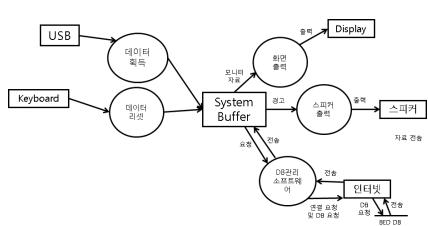
SBEP-010 Level2



SBEP-010은 로드셀의 데이터를 처리함에 목적이 있음 침대의 꼭지점마다 설치된 4개의 로드셀의 값을 받아들이고, 무게 amp센서인 HX711로 값을 증폭시킴

이를 통해 데이터를 획득하고 시스템 버퍼를 사용하여 들어오는 데이터 값을 하나의 블록으로 만들어 USB를 통해 전송함

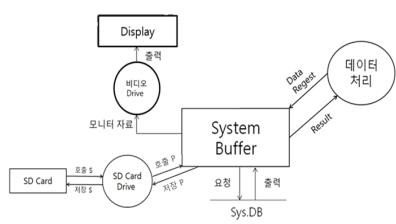
SBEP-011 Level2



SBEP-011은 받아들인 데이터를 DB에 저장 및 피씨 출력에 목적이 있음

USB로부터 받아들인 데이터를 피씨 출력 및 인터넷과 연결되어 DB 저장이 있음

SBEP-020 Level2

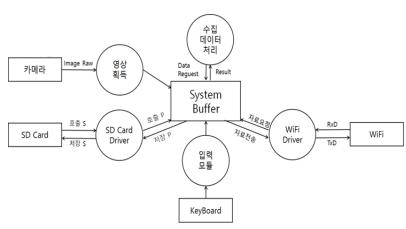


SBEP-020은 영상 출력이 목적

저장공간은 SD카드를 사용하여 늘렸고 시스템 관리 화면을 보기 위해 디스플레이 화면을 통해 관리 화면 출력

또한 시스템 버퍼는 언제든지 시스템 데이터베이스로 데이터를 요청과 출력 가능

SBEP-021 Level2



SBEP-021은 영상 전송이 목적

카메라를 통해 얻은 영상 정보를 와이파이를 통해 클라이언트로 전송

SD 카드를 통해 저장 공간을 확보할 수 있고 키보드를 통해 입력제어 가능

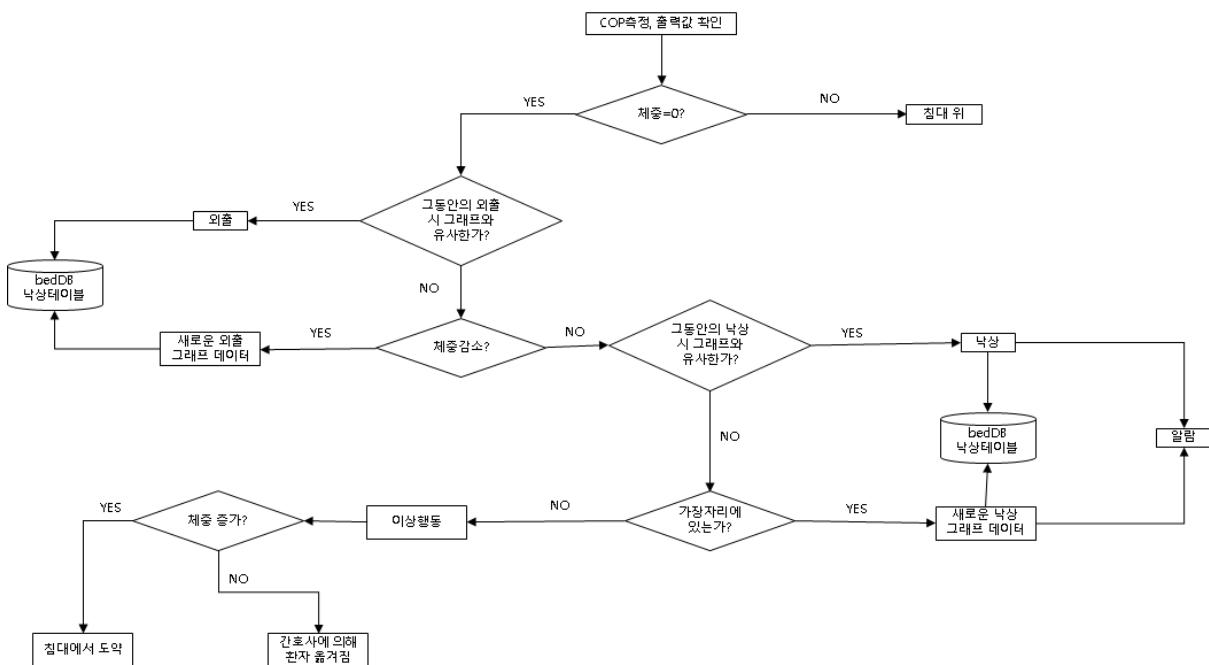
1.4_시스템_FlowChart

Created Time	@Sep 22, 2019 1:22 PM
Document Version	Ver 1.0.0
Last Edited By	준혁 김
Last Edited Time	@Sep 22, 2019 3:05 PM
Tags	FlowChart System
담당자	현욱 강 건호 이 준혁 김

시스템

FlowChart

낙상 알고리즘



1. 낙상 파형 분석

1) 낙상 직전(T_0):

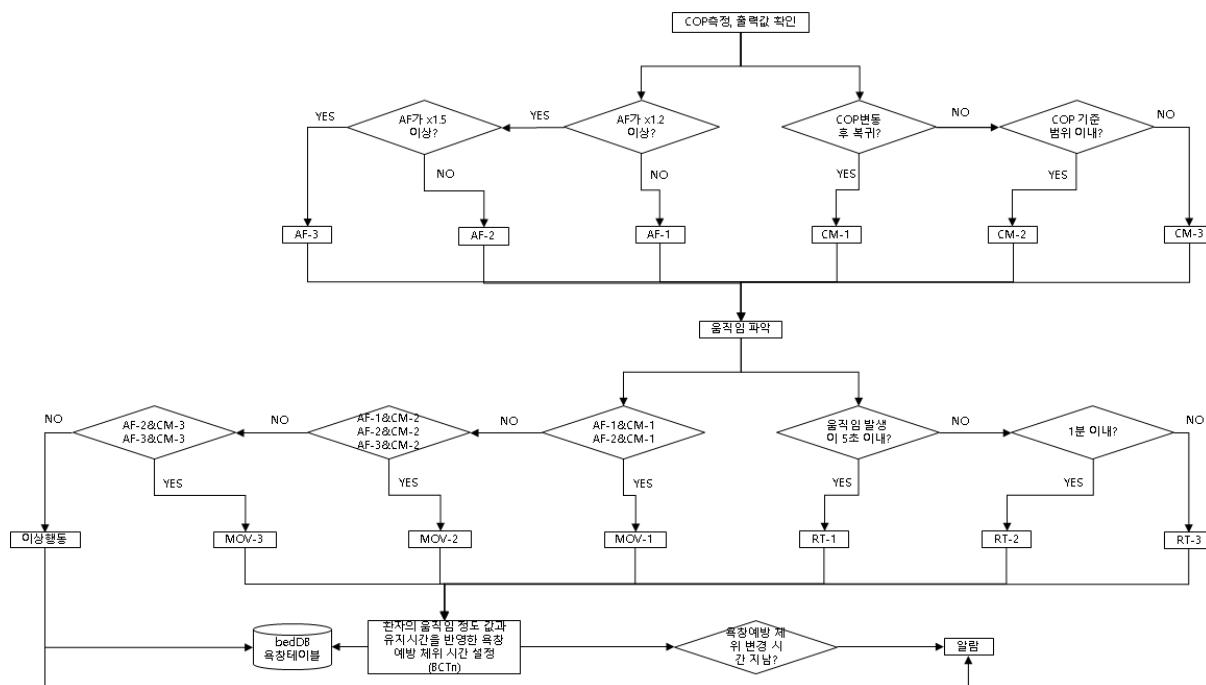
- 침대 가장자리로 COP가 이동
- 누운 범위가 존재
- 2) 낙상 직후(T_1):
- COP가 나타나지 않음
 - > 일어났을 때와 구분이 필요
- 압력센서 출력파형이 급감함
 - > 일어났다면 손으로 짓고 일어나기 때문에 무게가 증가 후 떨어짐
- 압력센서 출력파형의 크기가 크게 발생

2. 외출 파형 분석

1) 침대 밖 외출 직전(T_0):

- 침대 가장자리로 COP가 이동
- 앉아 있는 범위가 존재
- 빨은 땅에 지지하고 있기 때문에 체중 감소 발생
- 2) 침대 밖 외출 직후(T_1):
- COP가 나타나지 않음
- 체중 증가 발생
- 낙상시와 출력 파형이 다름을 예상

욕창 알고리즘



1. 육창 알고리즘

- 1) 압력센서 출력파형 실시간 측정 및 변동 확인
- 2) COP 측정 및 변동 확인
- 3) 실시간 육창예방 움직임 정도(Movement Level) 측정
- 4) 환자의 움직임 정도에 따른 유지시간(Retention Time) 측정
- 5) 환자의 움직임 정도 값과 유지시간을 반영한 육창예방 체워변경시간 설정
- 6) 육창예방 체워변경시간에 따른 알람 발생

2. 육창예방 움직임 및 유지시간

- 1) 육창예방 움직임 정도(Movement Level) 분류
 - 1단계(Mov-1): 육창예방에 기여하지 못한 움직임이 발생
 - 2단계(Mov-2): **육창예방에 어느 정도 기여하는 움직임**이 발생
 - 3단계(Mov-3): 육창예방에 충분히 기여하는 움직임이 발생
- 2) 육창예방에 기여하는 움직임
 - 압력센서 출력파형의 급격한 변화와 함께 COP 변화 발생으로 2가지 조건을 활용
 - ① 압력센서 출력 파형 변동 일정크기 이상 변화
 - ② 환자 COP 변동(일정한 길이 이상)

욕장 파형 분석

- 1) 압력센서 출력파형(크기/주파수) 실시간 측정
- 2) COP 측정 및 변동(어깨 폭/2 기준으로 함) 확인
 - 1단계(CM-1) : COP 작은 변화가 있으나 제자리로 바로 돌아옴
 - 2단계(CM-2) : COP 변동 조금(기준 이내 범위) 있음
 - 3단계(CM-3) : COP 변동 충분히(기준 범위 또는 이상) 있음
- 3) 욕장예방 움직임 정도 설정
 - 1단계(Mov-1) : (AF-1 & CM-1), (AF-2 & CM-1)
 - 2단계(Mov-2) : (AF-1 & CM-2), (AF-2 & CM-2), (AF-3 & CM-2)
 - 3단계(Mov-3) : (AF-2 & CM-3), (AF-3 & CM-3)
- 4) 유지시간
 - 움직임 발생 후 유지시간
- 5) 욕장예방 체위변경시간(Body Change Time, BCT_n) 설정
 - BCT_n = 2T - ((Mov) × (RT)),

	AF-1	AF-2	AF-3
CM-1	Mov-1	Mov-1	무게중심은 고정 출력파형이 크게 변함 = 별작의심 (참고)
CM-2	Mov-2	Mov-2	Mov-2
CM-3	무게중심이 크게 변하면서 출력파형은 변하지 않는다. = 침대 위를 조심스럽게 이동한다.	Mov-3	Mov-3

AF : 압력센서 출력파형
 CM : COP 측정 및 변동 (어깨 폭/2 기준) 확인
 Mov : 욕장예방 움직임 정도 설정
 RT : 유지시간
 BCT_n = 2T - ((Mov) × (RT)), 여기서 T는 60분

1.5_시스템_학술자료분석

Created Time	@Sep 21, 2019 4:52 PM
Document Version	Ver 1.0.0
Last Edited By	준혁 준혁 김
Last Edited Time	@Sep 22, 2019 3:12 PM
Tags	Magazine System Thesis
담당자	현욱 강HH 건호 이준혁 준혁 김

시스템

학술자료분석

학술자료분석 + Add a View			
▲ 자료명	■ 형태	■ 관련분야	■ 내용
고령친화산업 시장 동향(한국보건산업진흥원)	매거진	#고령화 #시장	고령친화산업 시장규모 및 전망
낙상방지장치(개명대)	특허	#낙상 #기네트	서비스에 연결된 낙상 측정 장치를 모니터링카페트센서를 사용
노인의 수면 장애(순천향대 의과대학)	논문	#노인 #수면	노화에 따른 수면의 변화수면 장애의 위험 요인수면 장애의 평가수면장애
병설 모니터링 시스템에 관한 연구 (창원대학교)	논문	#모니터링 #병설 #낙상 #압력패드	기존의 시스템 - 가속도 센서, 카메라, 포트다이오드등작 감지 시스템 설계 및 구현압력패드 사용
병원침대용 체중 측정기 (삼성캐스터)	특허	#캐스터 #로드셀#브라켓	브라켓을 사용하여 캐스터와 로드셀을 고정
요양보호사 직무교육 교재(국민건강보험)	매거진	#요양 #육상 #노인	요양보호사가 알아야되는 노인의 인권, 육상 관리, 재가급여전자관리시스템 작성
소비자 수용도 분석을 통한 품케어 침대 디자인 컨셉 개발(KIIT)	논문	#침대 #디자인 #품케어	고령자의 트렌드 수용도를 분석하여 기능성에 중점을 두어 프레임, LED등을 이용해 디자인 설계
영상분석을 통하여 환자의 낙상 위험을 판단하는 장치가 포함된 침대(동의대)	특허	#영상 #낙상 #침대	침대에 환자를 모니터링 카메라를 설치하여 낙상 가능성을 판단하는 감시모듈
육상 예방 침대(한태희)	특허	#육상 #침대	침대 프레임을 이용해 육상 예방을 할 수 있게 작동
요양침대용 자동사세전환 및 낙상모니터링 시스템에 관한 연구	논문	#침대 #낙상 #모니터링	각도 측정으로 낙상 모니터링을 통하여 자동 침대 각도 조절
침대 우게 감지 기능을 갖는 환자 자세 주정 시스템(가온소프트)	특허	#침대 #우게#로드셀	로드셀을 이용한 침대 우게 감지로 환자 자세 주정 시스템
환자 낙상예방 모니터링 장치(경남대)	특허	#낙상 #모니터링	임베디드 보드를 활용하여 OpenCV 프로그램으로 데이터 처리하여 환자의 움직임을 주른
Rondish	제품	#침대 #낙상	낙상 관련 제품
SmartCareBedsfor Elderly Patients with Impaired Mobility	논문	#침대 #로드셀 #움직임	로드셀과 FSR센서를 이용하여 낙상 패턴 예측
65세 이상 노인의 주막 및 낙상 일원환자의 역학적 특성	매거진	#노인 #낙상	2004년 ~ 2013년 퇴원손상증조사 결과를 바탕으로 노인 주막 행동 및 결과 분석
u-Home 환경에서 멀티센서 기반 u-Care System 구현	논문	#스마트홈#노인	집에서 노인 모니터링 시스템
육창간호 실무지침서	교재	#육창 #간호	육창의 정의, 육창 예방, 관리 지침서
요양병원 치매 환자의 낙상실태와 낙상위험요인 분석	매거진	#치매 #낙상	치매환자에 대한 낙상 실태 및 위험요인 분석
일 종합병원 입원 환자의 낙상실태와 낙상위험요인 분석	매거진	#입원환자 #낙상	입원환자에 대한 낙상 실태 및 위험요인 분석
중소병원 입원 환자의 낙상실태와 낙상위험요인 분석	매거진	#입원환자 #낙상	입원환자에 대한 낙상 실태 및 위험요인 분석
입원 환자의 낙상 사고에 대한 간호사의 경험	매거진	#간호사#낙상	부담감, 책임, 어려움, 지원 등을 작성예방 전략이 부족
노약자를 위한 침대의 매트릭스 관한 연구	논문	#노인 #침대#매트릭스	매트릭스 사용에 대한 영향 분석
매트리스에서의 수면형태에 따른 체압분포 비교	논문	#수면 #체압#매트릭스	매트리스에서 수면시체압 변화 분석
수면장애 모니터링을 위한 수면감지 시스템	논문	#수면#모니터링	수면 장애에 대한 모니터링 시스템 제안
수면형태 감지 시스템에 관한 연구	논문	#수면	수면 감지 시스템 제안
주막 및 낙상 손상환자의 손상발생장소별 퇴원환자수: 성별	예 셰	#낙상 #퇴원환자	http://kosis.kr/statHtml/statHtml.do?orgId=1170tb1Id=DT_11760NP339conn_path=I3
체중분포측정을 통한 노인용 침대활동 모니터링 시스템 개발(정상현)	논문	#노인 #체중 #침대	체중 분포 가선상을 활용한 알고리즘침대 하중 측정을 위한 치구
육창예방과 관리를 위한 통합적 육창 간호 프로그램의 효과(kci)	매거진	#육창 #간호	육창 예방 및 관리 프로그램의 효과
의료용침대시장(한국개발특구진흥재단)	매거진	#침대 #시장	2017년과 비교하여 2022년의 침대 시장 분석
자동차화된 대기능 통합 전동 침대 개발에 대한 연구(JIIBO)	논문	#침대 #전동	높이조절, 등관상승, 무릎상승, 훌트, 좌우회전 등 환자가 편하게 침대 움직임
정상 수면의 생리(HMR)	매거진	#수면	인간의 정상적인 수면의 데이터 분석
증환자실의 육창 예방 중재 프로그램의 효과	매거진	#육창 #증환자실	어떠한 프로그램이 육차예방에 도움이 되었는지 분석
증환자실 환자의 육창 관련 경계압력 예측요인(경상대병원)	매거진	#증환자실 #육창 #압력	육창위험요인이 무엇인지 압력에 따라 분석



분석 결과

1. 논문이 많다
 - 개인적인 의견이지만 검증된 결과를 토대로 증명
 2. 매거진이 많다
 - 증명된 기관에서 추출된 다량의 데이터를 분석



키워드 분석 결과

- 침대, 낙상
 - 노인
 - 욕창
 - 압력, 수면



분석 결과

1. 대상 : 노인
 2. 도구 : 침대
 3. 목적 : 낙상, 육창

1.6_시스템_시연영상

Document Version	Ver 1.0.0
Last Edited By	👩 현욱 강
Last Edited Time	@Sep 22, 2019 5:16 PM
Tags	Testing 영상
담당자	👩 현욱 강  건호 이  준혁 김

시연 영상

영상

▼ Unit

0_Unit_방법.mp4 참조

▼ 공통

▼ 누웠을 때의 체중 및 중심 확인

1. 무게중심

1.1_체크_무게중심_정확도.mp4 참조

▼ 좌측 및 우측으로 이동시 움직임 변화

1. 무게중심

1.2_체크_좌우측_무게중심.mp4 참조

2. 차트

1.3_체크_좌우측_차트.mp4 참조

▼ 낙상

▼ 좌측낙상

1. 무게중심
- 3.1_낙상_좌측_무게중심.mp4 참조
2. 차트
- 3.2_낙상_좌측_LR차트.mp4 참조

▼ 외출

- ▼ 외출 청년버전
1. 무게중심
 - 2.1_외출_청년버전_무게중심.mp4 참조
 2. 무게차트
 - 2.2_외출_청년버전_무게차트.mp4 참조

▼ 우측낙상

1. 무게중심
- 3.3_낙상_우측_무게중심.mp4 참조
2. 차트
- 3.4_낙상_우측_LR차트.mp4 참조

- ▼ 외출 노인버전
1. 무게중심
 - 2.3_외출_노인버전_무게중심.mp4 참조
 2. 무게차트
 - 2.4_외출_노인버전_무게차트.mp4 참조

▼ 욕창

▼ 완전 정면

1. 무게중심
- 4.1_욕창_정면.mp4 참조

▼ 다양한 자세

- ▼ 불가사리형 정면
1. 무게중심
 - 5.2_불가사리형_정면.mp4 참조

▼ 똑바로 좌측면

1. 무게중심
- 4.2_욕창_좌측_정면.mp4 참조

▼ 태아형 좌측 우측면

1. 무게중심
- 5.1_태아형_좌측_우측.mp4 참조

▼ 똑바로 우측면

1. 무게중심

4.3_욕창_우측_정면.mp4
참조

▼ 작은 움직임 좌우측면

1. 무게중심

4.4_욕창_작은움직임_좌우
측_무게중심.mp4.mp4 참
조

2. LR차트

4.5_욕창_작은움직임_좌우
측_LR차트.mp4 참조

영상분석목록

Name	Tags	Upload Date
<u>0 Unit 방법</u>	unit	@Sep 21, 2019
<u>1.1 체크 무게중심 정확도</u>	무게중심 정확도 체크	@Sep 21, 2019
<u>1.2 체크 좌우측 무게중심</u>	무게중심 좌우측 체크	@Sep 21, 2019
<u>1.3 체크 좌우측 차트</u>	좌우측 차트 체크	@Sep 21, 2019
<u>2.1 외출 청년버전 무게중심</u>	무게중심 외출 청년	@Sep 21, 2019
<u>2.2 외출 청년버전 무게차트</u>	무게차트 외출 청년	@Sep 21, 2019
<u>2.3 외출 노인버전 무게중심</u>	노인 무게중심 외출	@Sep 21, 2019
<u>2.4 외출 노인버전 무게차트</u>	노인 무게차트 외출	@Sep 21, 2019
<u>3.1 낙상 좌측 무게중심</u>	낙상 무게중심 좌측	@Sep 21, 2019
<u>3.2 낙상 좌측 LR차트</u>	LR차트 낙상 좌측	@Sep 21, 2019

Name	Tags	Upload Date
<u>3.3 낙상 우측 무게중심</u>	낙상 무게중심 우측	@Sep 21, 2019
<u>3.4 낙상 우측 LR차트</u>	LR차트 낙상 우측	@Sep 21, 2019
<u>4.1 육창 정면</u>	육창 정면	@Sep 21, 2019
<u>4.2 육창 좌측 정면</u>	육창 정면 좌측	@Sep 21, 2019
<u>4.3 육창 우측 정면</u>	육창 우측 정면	@Sep 21, 2019
<u>4.4 육창 작은움직임 좌우측 무게 중심</u>	무게중심 육창 작은움직임 좌우측	@Sep 21, 2019
<u>4.5 육창 작은움직임 좌우측 LR 차트</u>	LR차트 육창 작은움직임 좌우측	@Sep 21, 2019
<u>5.1 태아형 좌측 우측</u>	좌우측 태아형	@Sep 21, 2019
<u>5.2 불가사리형 정면</u>	불가사리형 정면	@Sep 21, 2019

2.1_HX711

Created Time	@Sep 21, 2019 10:32 PM
Document Version	Ver 1.0.0
Last Edited By	 현욱 강
Last Edited Time	@Sep 22, 2019 2:08 PM
Tags	HX711
담당자	 건호 이

HX711

로드셀과 PC 중간에서 위치함

로드셀로부터 받은 전압값을 증폭시키고 아날로그 데이터로 들어오는 로드셀의 전압값을 디지털로 변환시킴

세부구성

- AD Convert → 2.1.1
- AMP → 2.1.2

2.1.1_HX711_AD Converter

Created Time	@Sep 21, 2019 3:20 PM
Document Version	Ver 1.0.0
Last Edited By	HH 건호 이
Last Edited Time	@Sep 22, 2019 4:19 PM
Tags	ADC HX711
담당자	현욱 강

HX711

AD Convert

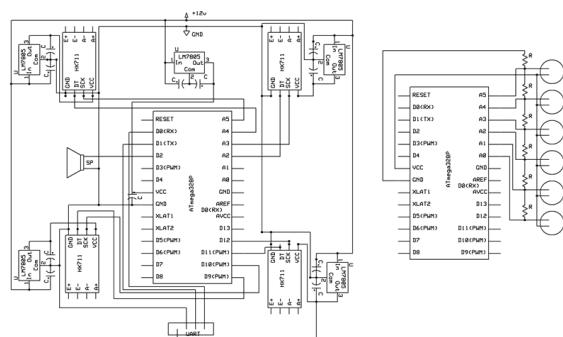
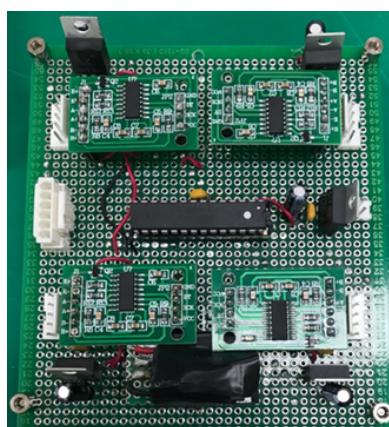
HX711

HX711은 계량용으로 설계된 정밀 24비트 아날로그 디지털 컨버터(이하 ADC)임
입력 멀티플렉서는 low-noise에 대해 증폭을 위해 Channel A 또는 Channel B를 선택함

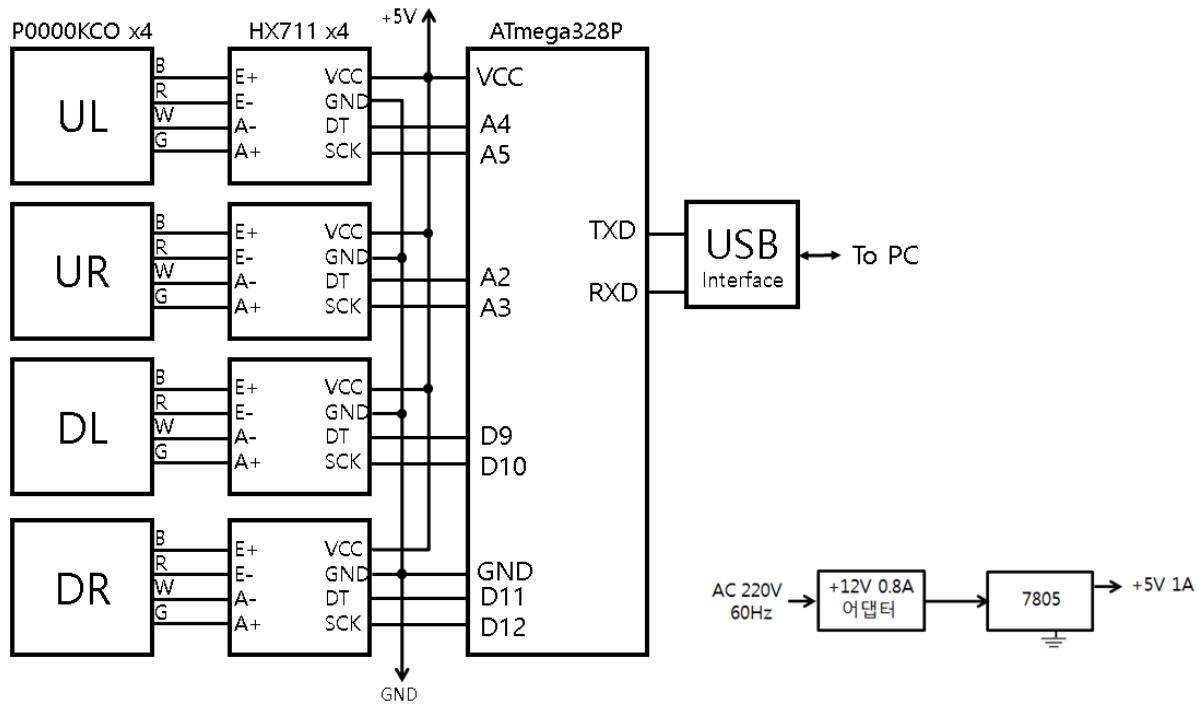
Channel A는 프로그래을 통해 128 또는 64의 이득으로 풀 스케일 차동 입력에 각각 20mV, 40mV가 입력되어야 함

만약 5V 전원이 공급되면 Channel B의 고정 이득은 32

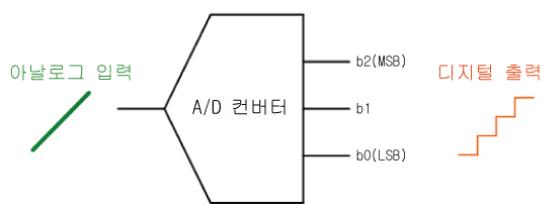
HX711은 핀을 통해 Control 가능



세부블럭도



AD Convert 개념

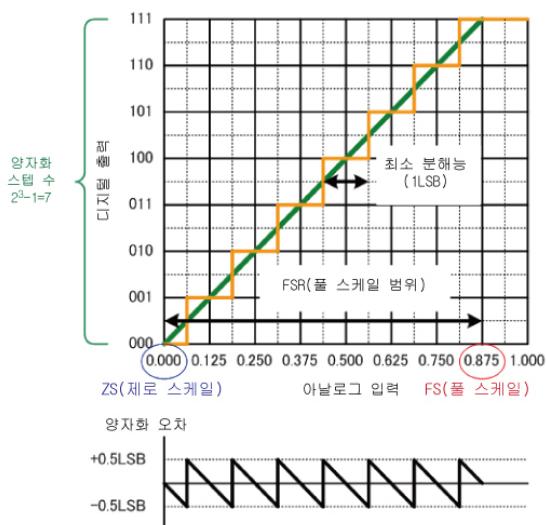


A/D 컨버터는 아날로그 신호의 진폭을 이산적인 주기로 추출하여, 부호로 표시된 디지털 신호로 변환함

A/D 변환된 디지털 신호의 bit 수를 분해능 (이 경우, 3bit), 최상위 bit를 MSB (Most Significant Bit), 최하위 bit를 LSB (Least Significant Bit)라 함

그래프는 아날로그 신호 (입력)와 디지털 신호 (출력)의 관계를 나타내었음

디지털 신호의 차이로서 판별 가능한 아날로그 신호의 최소 진폭이 최소 분해능 ($=1\text{LSB}$)이며, 아날로그 신호와



디지털 신호 사이에서 발생하는 오차를 양자화 오차라 함

또한, 최초의 디지털 신호 변화점 ($000 \rightarrow 001$)의 0.5LSB 이하를 제로 스케일, 최후의 디지털 신호 변화점 ($110 \rightarrow 111$)의 0.5LSB 이상을 풀 스케일, 제로 스케일에서 풀 스케일의 간격을 풀 스케일 범위라고 함

Analog to Digital Converter

아날로그 신호를 디지털 값으로 변환

microcontroller를 이용하여 임베디드 시스템을 구축할 때 Timer 만큼이나 중요한 기능임 - DAC보단 ADC를 많이 사용함

예를 들어 어떠한 임베디드 시스템을 만들어낼 경우 그 시스템 내에서 처리되는 모든 데이터와 연산 등은 모두 디지털화($0, 1$)가 되어 처리됨

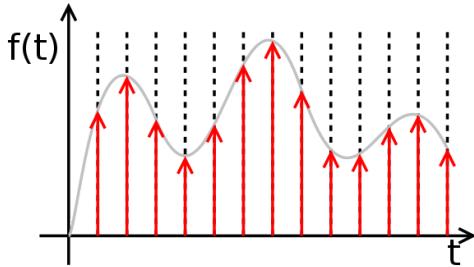
실제적인 환경은 아날로그적(연속으로 변함)이지만 데이터 처리는 디지털이기 때문에 ADC가 필요

표본화(Sampling)

표본화는 아날로그 신호를 일정한 시간 간격을 가지고 해당 시간 간격마다 아날로그의 신호를 채취(시간에 따라 연속적으로 변화하는 아날로그 신호들 중에서 특정한 매 시간 간격마다 '그' 아날로그 신호만을 취득)를하는 것을 의미

아날로그 신호를 처리하기위서는 처리 속도가 매우 빨라야하지만 MCU에서의 한계 (1초를 쪼개면 0.1초 , 그것을 또 쪼개면 0.01초 ... 결국 시간이 쪼개도 무한이기 때문에 처리의 속도 한계가 발생)가 발생함

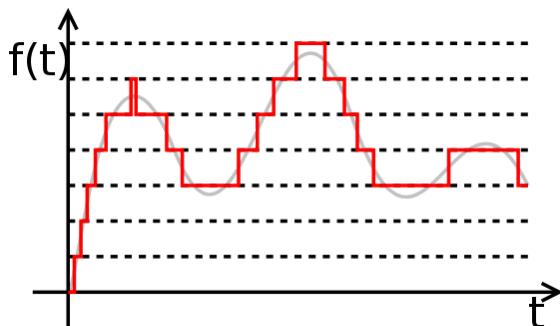
따라서, 아날로그 신호를 표본화 함



' 샤논의 표본화 정리 '

아날로그 신호를 일정한 시간 간격으로 나눈 후 그 시간 간격에 해당하는 아날로그 신호를 취득한 그래프

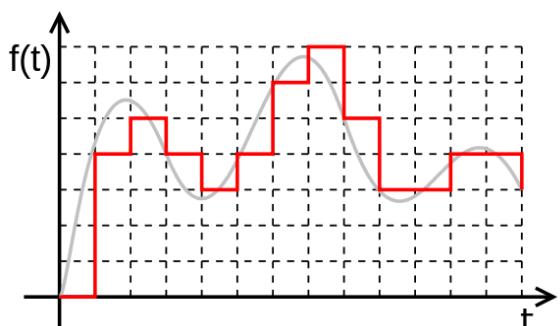
⇒ 일정한 시간 간격으로 아날로그 신호를 취득할 때 표본화를 수행할 아날로그 신호의 주파수의 2배의 속도로 표본화를 수행하여 얻는다면 그 표본화된 데이터를 바탕으로 원래 신호를 거의 완벽하게 재생 가능



' 양자화-Quantization '

표본화의 과정으로 아날로그 신호를 일정한 접압 레벨의 구간을 나눈 영역에 강제적으로 대응시키는 과정

⇒ 전압 레벨 구간을 촘촘하게 나누면 나눌수록 처리할 데이터의 양은 많아지지만 원래의 아날로그 신호에 매우 근사하게 접근할 수 있기 때문에 해상도 (Resolution)가 중요



' 부호화-Encoding '

표본화와 양자화가 상호적으로 이뤄진 후에 얻어진 데이터를 가지고 여기에 2진수의 코드를 부여

ex) 해상도가 4-bit라면
0000, 0001, 0010 ~ 1110, 1111까지 16개의 코드를 부여할 수 있음

위 표본화→양자화→부호화의 순으로 아날로그 신호를 처리하면 Transistor의 On/Off만으로 모든 데이터를 처리하는 디지털시스템에서도 실제 환경인 아날로그 데이터를 디지털 영역으로 가져와 처리가 가능

참고문헌

[1] HX711_datasheet, AVIA SEMICONDUCTOR

https://www.mouser.com/ds/2/813/hx711_english-1022875.pdf

[2] Analog to Digital Converter(ADC)란 무엇인가?,
ansdbtls4067, 2016.11.28

::::: 땡쓰 전자연구소 :::: : 네이버 블로그

<https://blog.naver.com/ansdbtls4067/220873048333>

2.1.2_HX711_AMP

Created Time	@Sep 19, 2019 10:35 PM
Document Version	Ver 1.0.0
Last Edited By	HH 건호 이
Last Edited Time	@Sep 22, 2019 4:27 PM
Tags	AMP HX711
담당자	HH 건호 이

HX711

AMP

도입

HX711은 로드셀과 PC사이에서 데이터 증폭 및 ADC역할을 담당함

필요성

로드셀로부터 받아오는 데이터는 전압값임

스트레인 게이지에 가해지는 압력에 의해 바뀌는 전압값을 읽는데, 그 전압값이 너무 작아서 무게간 구별이 어려움

따라서 전압값의 차이를 크게 하여 획득한 데이터간의 구별을 쉽게 하기 위한 것을 Programmable Gain Amplifier(PGA)라 함

PGA

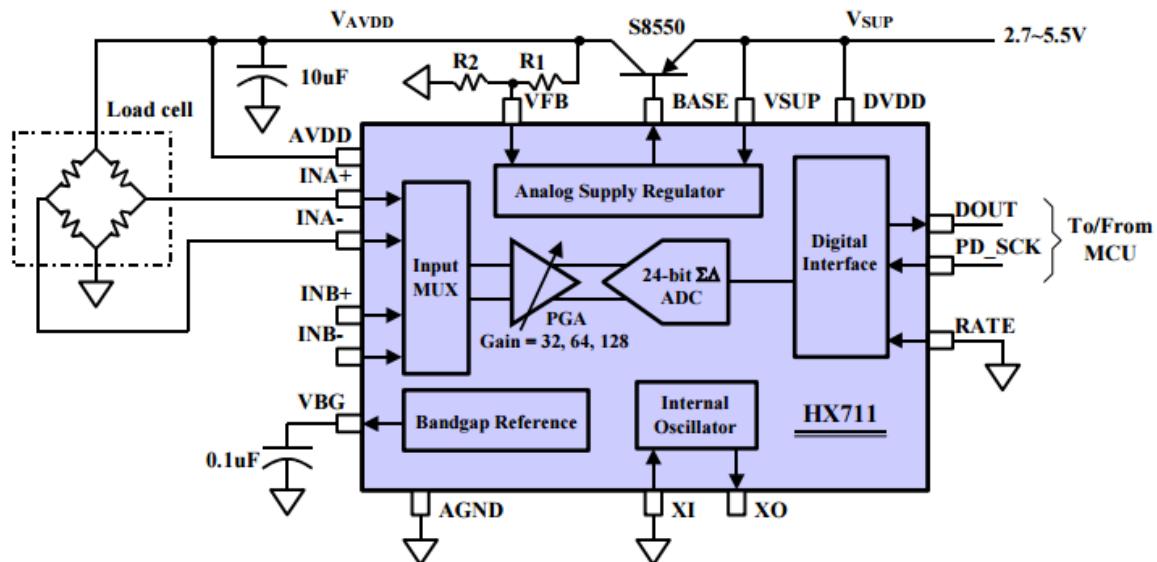


Fig. 1 Typical weigh scale application block diagram

위의 사진은 HX711의 블록다이어그램이며 PGA 영역을 주목

Input MUX(Input Multiplexer 로드셀로부터 여러 개의 데이터를 받아들이는 역할을 수행)에 로드셀로부터 데이터가 들어오면 PGA는 32, 64, 128bit 중 하나를 선택하여 증폭

증폭된 데이터는 ADC로 이어져 디지털값이 됨

32, 64, 128bit 선택에 따른 PGA의 차이점

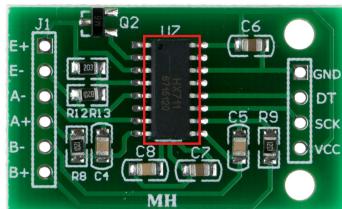
받아들이는 bit가 커지면 단순히 받아들이는 데이터 값이 커짐을 의미함

PGA의 변화는 없고 전체적 속도의 차이는 생길 수 있음

32, 64, 128bit는 받아들이는 데이터에 따라 알맞게 설정하면 됨

4byte를 받아들이는 이번 프로젝트에서는 32bit를 사용하였음

Input MUX로 들어오는 로드셀 데이터와 SOP-16L Package



HX711에서 데이터를 받아들이고 이를 전송하는 역할을 하는 IC임

위의 사진 중 붉은 사각형이 안에 있는 장치임

SOP-16L Package 중 SOP은 Small out-line package의 의미로 표면 실장형(핀이 장치에 바로 납땜할 수 있도록 나온 IC) 패키지 중 하나임

PGA에 사용되는 핀은 로드셀의 데이터를 받아오는 7, 8, 9, 10번 핀임

자세한 내용은 아래의 핀맵을 참고

Regulator Power	VSUP	1	*	16	DVDD	Digital Power
Regulator Control Output	BASE	2		15	RATE	Output Data Rate Control Input
Analog Power	AVDD	3		14	XI	Crystal I/O and External Clock Input
Regulator Control Input	VFB	4		13	XO	Crystal I/O
Analog Ground	AGND	5		12	DOUT	Serial Data Output
Reference Bypass	VBG	6		11	PD_SCK	Power Down and Serial Clock Input
Ch. A Negative Input	INNA	7		10	INPB	Ch. B Positive Input
Ch. A Positive Input	INPA	8		9	INNB	Ch. B Negative Input

SOP-16L Package

Pin #	Name	Function	Description
1	VSUP	Power	Regulator supply: 2.7 ~ 5.5V
2	BASE	Analog Output	Regulator control output (NC when not used)
3	AVDD	Power	Analog supply: 2.6 ~ 5.5V
4	VFB	Analog Input	Regulator control input (connect to AGND when not used)
5	AGND	Ground	Analog Ground
6	VBG	Analog Output	Reference bypass output
7	INA-	Analog Input	Channel A negative input
8	INA+	Analog Input	Channel A positive input
9	INB-	Analog Input	Channel B negative input
10	INB+	Analog Input	Channel B positive input
11	PD_SCK	Digital Input	Power down control (high active) and serial clock input
12	DOUT	Digital Output	Serial data output
13	XO	Digital I/O	Crystal I/O (NC when not used)
14	XI	Digital Input	Crystal I/O or external clock input, 0: use on-chip oscillator
15	RATE	Digital Input	Output data rate control, 0: 10Hz; 1: 80Hz
16	DVDD	Power	Digital supply: 2.6 ~ 5.5V

Table 1 Pin Description

첨부된 데이터시트를 참고(p2)하면 다음과 같은 핀 배치도를 볼 수 있음

Input MUX로 들어오는 로드셀 데이터는 INA-, INA+, INB-, INB+ 총 4가지 있음 즉, A채널과 B채널을 가지고 있으며 A채널로 들어가는 '센서출력신호A+'(INA+)와 '센서출력신호A-'(INA-)가 있으며 B채널로 들어가는 '센서출력신호B+'(INB+)와 '센서출력신호B-'(INB-)가 있음

따라서 로드셀의 데이터시트를 확인하고 로드셀 데이터 핀을 알맞은 위치에 연결하면 됨

A채널과 B채널을 사용하여 한 HX711센서에 두 개의 로드셀이 가능함 하지만 두 데이터가 합쳐진 상태로 출력됨 코딩을 통해 마스터에서 명령을 내려 채널 A와 채널 B의 값을 각각 받을 순 있음 하지만 그 과정의 복잡함과 저렴한 HX711센서의 가격을 고려했을 때 로드셀 1개에 HX711센서 1개 사용을 권장함

2.2_Atemega328p

Created Time	@Sep 21, 2019 10:36 PM
Document Version	Ver 1.0.0
Last Edited By	HH 건호 이
Last Edited Time	@Sep 22, 2019 2:17 PM
Tags	Atemega328p
담당자	HH 건호 이

Atemega328p

로드셀의 데이터를 변환한 HX711 컨트롤을 담당하고 있음

4개의 HX711센서로부터 순차적으로 값을 받아오고, Check sum을 이용한 통신 오류 확인

PC로부터 요구사항이 들어왔을 때 인터럽트를 실행하여 모은 데이터를 전송

세부구성

- Atemega328p_Firmware → 2.2.1

2.2.1_Atemega328p_Firmware

Created Time	@Sep 21, 2019 4:00 PM
Document Version	Ver 1.0.0
Last Edited By	HH 건호 이
Last Edited Time	@Sep 22, 2019 4:30 PM
Tags	Atemega328p Firmware
담당자	娃 현욱 강 HH 건호 이

Atemega328p

Firmware

Loadcell_Atmega328P_Code 폴더에는 다음과 같이 있음



▼ Loadcell 폴더를 보면 Master 폴더와 Slave 폴더가 있음

- Mater는 깡통내부 중앙에 위치한 칩이고 Slave는 각각의 모서리에 있는 4개의 칩 있음
- Slave는 프로그램이 실행되면 로드셀 각각의 값을 받아옴
- Master는 계속해서 Slave의 값을 UL,DL,DR,UR 순 블록단위로 받아옴
- PC에서 요청을하면 인터럽트를 발생해서 Master의 버퍼블록이 전달

Code Review

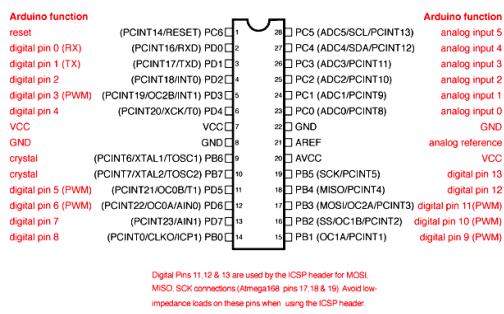
사용하지 않는 코딩

- xADC.cpp
- xI2CMaster.cpp
- xI2CSlave.cpp
- xLCD.cpp

xHX711.cpp

- 사용함
- 데이터를 읽어오는 역할을 담당

ATMega328P and Arduino Uno Pin Mapping



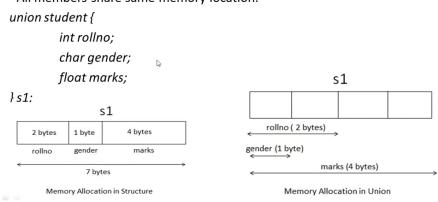
- 보통 포트는 8bit가 한 세트
- 14번 핀은 B0
- B0를 사용하려면 B포트를 전부 선택하여 그 중에 하나의 값을 읽어야 함
- 즉, B가 8비트이기 때문에 0부터 7까지 있음
- 기존의 14번핀을 읽는 방법
 - 8비트인 B를 전부 읽음
 - 그 중에 가장 작은 0만을 도출
- xPort의 핀 읽는 방법
 - 핀 하나의 포트를 만들었음
 - 따라서 핀 번호를 쓰면 그 핀을 읽을 수 있음
- Atmega328p에 맞춰둬서 128은 안 됨
- 입출력시 DDR을 바꿔줘야 하는데 이를 다 처리해줌

xUnion.h

Union 설명

• Union

- Collections of dissimilar types of data.
- All members share same memory location.



- 버퍼를 10으로 한 이유는 여유있게 사용하기 위함

ex) 체크썸같은걸 붙이면 데이터를 더 많이 보내줘야하는 경우가 생기니까.

- 데이터를 보낼때는 항상 버퍼를 보냄
- "size of 자료형" 을 입력하면 그 만큼의 버퍼를 보냄

```
// Union.h
```

```
typedef union {
    float f;
    uint8_t u8;
    uint16_t u16;
    uint32_t u32;
```

```
    uint8_t buffer[10];
} xUnion;
```

xUART.cpp

- 시리얼 통신을 위해 사용

put

```
// xUART.h

static void put(char x);
static void put(const char *str);

static void put(int8_t x);
static void put(uint8_t x);

static void put(int16_t x);
static void put(uint16_t x);

static void put(int32_t x);
static void put(uint32_t x);

static void put(float x);
static void put(double x);

static void putln();
static void putBin(uint8_t x);
static void putHex(uint8_t x);
```

- 받아들인 데이터를 무조건 문자로 바꿔서 보내줌
- 화면에 바로 보이기 위함
- put이 데이터를 보낼 땐 문자열로 보내게 되니까 받은쪽에선 다시 숫자로 바꿔야함.
- 이런 현상을 막기위해 바로 데이터를 보내 줄 수 있는 transmitBuffer를 만듦

transmitBuffer

```
//xUART.cpp

void xUART::transmitBuffer(uint8_t *buff, int16_t len)
{
    if(!isInitialized) return;
    for(int16_t i=0;i<len; i++) {
        transmitByte(buff[i]);
    }
}
```

- 실제로 데이터를 보낼 땐 transmitBuffer를 많이 씀

xUART.h

- 아트메가와 PC통신 사이에 사용되는 레지스터

```
// xUART.h

class xUART
{
    static bool isInitialized;
    static xDoubleStruct get_Double_Struct(uint8_t intLen, uint8_t fracLen);

public:

    xUART();
    xUART(bool receiveInterrupt);
    static void init(bool receiveInterrupt);

    static uint8_t isBufferReady();
    static uint8_t isDataReady();

    static void transmitByte(uint8_t x);
    static uint8_t receiveByte();

    static void transmitBuffer(uint8_t *buff, int16_t len);
    static void receiveBuffer(uint8_t *buff, int16_t len);

    static void transmitData(xUnion xdu, uint8_t len);
    static xUnion receiveData(uint8_t len);

    static void put(char x);
    static void put(const char *str);

    static void put(int8_t x);
    static void put(uint8_t x);

    static void put(int16_t x);
    static void put(uint16_t x);

    static void put(int32_t x);
    static void put(uint32_t x);

    static void put(float x);
    static void put(double x);

    static void println();
    static void putBin(uint8_t x);
    static void putHex(uint8_t x);

    static char getChar();
    static void getStr(char *str, uint16_t maxLength);
    static int16_t getInt16();
    static int32_t getInt32();
    static float getFloat();

};

};
```

- 아트메가의 레지스터
- 아두이노는 이런 레지스터가 자동으로 처리되는데 아트메가는 그렇지 않음으로 통신을 위해 생성
- 아두이노는 이런 것들이 있는 함수들인데 아트메가는 그런 기능이 없으므로 만들어 줌
- AVR프로그램이 아님

Loadcell_SPI_Master.cpp

- 깡통에서 중간에 있는 아트메가칩을 말함
- 슬레이브를 4개를 가지고 있음

인터럽트 함수

```
// Loadcell_SPI_Master.cpp  
  
xUART serial(true);
```

- 인터럽트를 하겠다는 의미
 1. 시리얼 포트에서 PC가 데이터를 보내라고 하면 보내야함
 2. PC가 데이터를 보내라고하면 저장장치가 없는 마스터는 보내야하는 데이터가 모일때 까지 대기만 하고 다른 일을 하지 못함
 3. 그래서 마스터는 일을 하고 있다가 PC가 데이터를 보내라고 하면 인터럽트 처리를 하며 데이터를 보냄
 4. 이 인터럽트 처리가 가능하게 하려면 true가 필요함

```
// Loadcell_SPI_Master.cpp  
  
ISR(USART_RX_vect)  
{  
    udr = UDR0;  
    if(udr==0xF0) out12Volt.put(HIGH);  
    else if(udr==0xFF) out12Volt.put(LOW);  
    else sendData();  
}
```

- 데이터가 오면 자동으로 실행되는 인터럽트 함수

▼ 아두이노 들어있는 깡통 설명

- 옆으로 빠지는 부분에서 외부와 연결할 수 있는 전원이 있음
- 딱 두 선만 사용
- 외부에서 깡통에 12V를 넣어주면 레귤레이터들이 9V 5V로 맞춰줘서 사용
- 그래서 F0가 들어오면 불을 켜고 FF가 들어오면 불을 끔
- 그 외의것은 전부 데이터임

sendData()

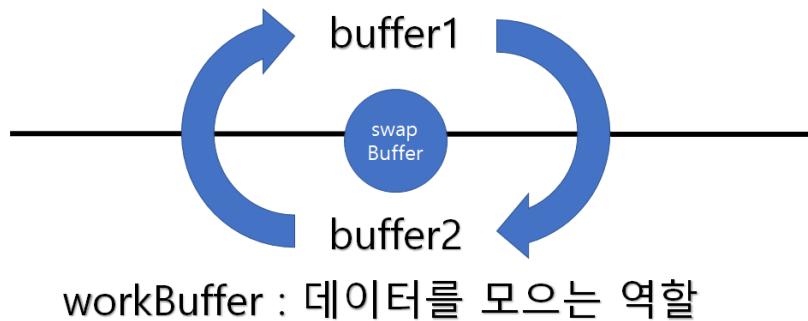
```
// Loadcell_SPI_Master.cpp  
  
uint8_t buffer1[17]; //17: 4bytes x 4loadcell + 1checksum byte  
uint8_t buffer2[17];  
volatile uint8_t *dataBuffer = buffer1;
```

```
volatile uint8_t *workBuffer = buffer2;
volatile uint8_t udr;
```

- 유니온 속성을 사용하지 않고 16바이트 + 체크섬을 합쳐 17바이트 사용
- volatile
 - 최적화를 포기하기 위한 키워드

dataBuffer와 workBuffer

dataBuffer : 데이터를 PC로 보내는 역할



workBuffer : 데이터를 모으는 역할

```
// Loadcell_SPI_Master.cpp

void sendData()
{
    serial.transmitBuffer((uint8_t *)dataBuffer, 16);
}
```

- 데이터를 보내는 건 dataBuffer가 담당함

```
// Loadcell_SPI_Master.cpp

void swapBuffer()
{
    volatile uint8_t *temp = dataBuffer;
    dataBuffer = workBuffer;
    workBuffer = temp;
}
```

- 스왑버퍼때만 데이터가 겹치지 않도록 해야함

체크섬

```
// Loadcell_SPI_Master.cpp

uint8_t getChecksum(uint8_t *buffer, uint8_t len)
{
    uint16_t sum = 0;
    for(uint8_t i=0; i<len; i++) {
```

```

        sum += buffer[i];
        if (sum & 0xFF00) sum = (sum & 0x00FF) + 1;
    }

    return (uint8_t)(sum);
}

```

- 4바이트의 데이터를 읽어서 NOT 연산함
- 데이터 통신간에 체크섬이 항상 맨 뒤에 오지 않음
- 따라서 그 때 원래의 데이터 + NOT 연산한 데이터를 더해주면 FF 즉, 10이 됨
- 10이 라면 정상으로 판단

데이터 수집

```

//Loadcell_SPI_Master.cpp

if(F_CPU == 8000000UL) clock_prescale_set(clock_div_1);

```

- 이게 없으면 하드웨어 클럭은 100만인데 소프트웨어 클럭은 800만인줄 알고 처리속도가 8 배 느려짐

```

//Loadcell_SPI_Master.cpp

uint8_t bufferIndex = 0;

for(uint8_t i=0; i<NUM_SLAVE; i++)
{
    slave[i].put(HIGH);
}

clearBuffer(buffer1, 17);
clearBuffer(buffer2, 17);

out12Volt.put(LOW);

sei();

```

- 슬레이브가 4개라 따로따로 받아야 함
 1. 전부 데이터를 받지 않는 상태인 HIGH를 해줌
 2. 버퍼 클리어 해줌
 3. 12V 출력된는거 꺼줌
 4. sei()를 사용하여 인터럽트 발생시킴

```

//Loadcell_SPI_Slave.cpp

while(1)
{
    for(uint8_t i=0; i<NUM_SLAVE; i++)
    {

```

```

    spi328.selectSlave(slave[i]);

    bufferIndex = i * 4;

    //read data from slave inside critical section protected by critical section algorithm.
    spi328.readBufferMuteEx((uint8_t *) (workBuffer+bufferIndex), 4);

    //spi328.blockCount is increased when critical section is blocked by slave.
    //by checking this count you can measure competition between master and slave.
    spi328.blockCount = 0;

}

//for

swapBuffer();
_delay_us(500);

}

//while

```

- 슬레이브 4개를 돌림
- 슬레이브 하나를 골라서 데이터를 받아들임
- 그 데이터로 bufferIndex를 만듬
- 0.5ms 대기
 - 이는 메뉴얼에 따라 결정
- bufferIndex
 - 4개의 데이터를 합쳐주는 작업

x328Util.h

CPU 클럭

```

// x328Util.h

#define F_CPU 8000000UL

```

- 아트메가의 클럭 스피드가 800만까지 되는데 100만으로 줄여서 나옴
- 이는 배터리를 아끼기 위함
- 우리는 외부 전원을 계속 사용하기 때문에 800만으로 올렸음
- UL : Unsigned Long
- 프로그램이 알아서 컴버전을 하지만 이를 빨리 하기 위해서 UL을 작성

슬레이브와의 통신

```

// x328Util.h

#define HIGH 1
#define LOW 0

```

- 슬레이브가 4개임
- HIGH = 연결 X
- LOW = 연결 O
- 따라서 전부 HIGH로 두었다가 필요한 슬레이브 하나만 LOW로 만들어 통신

SPI handler

- 마스터가 데이터를 달라고 할 때 슬레이브와 데이터 교환을 통해 바꿈

328이 있는 것 vs 없는 것

- 328이 있는 건 328의 핀 번호와 직접적으로 연관된 것
- 없는 것은 범용

참고문헌

[1] [c/cpp] volatile 과 사용 예, memOry, 2017.07.08

[c/cpp] volatile 과 사용 예

<https://m.blog.naver.com/PostView.nhn?blogId=kyed203&logNo=220053808209&proxyReferer=https%3A%2F%2Fwww.google.com%2F>

2.3_Loadcell

Created Time	@Sep 21, 2019 10:41 PM
Document Version	Ver 1.0.0
Last Edited By	준혁 김
Last Edited Time	@Sep 22, 2019 2:29 PM
Tags	Loadcells
담당자	건호 이

Loadcell

무게를 실질적으로 받아들이는 역할

다양한 로드셀 중에서 프로젝트에 사용된 빔형과 S자형 로드셀에 대해 주로 알아보고 두 가지 형태의 차이점등에 대해 설명

세부구성

- Spec → 2.3.1
- 종류별차이 → 2.3.2

2.3.1_Loadcell_spec

Created Time	@Sep 21, 2019 5:06 PM
Document Version	Ver 1.0.0
Last Edited By	HH 건호 이
Last Edited Time	@Sep 22, 2019 4:34 PM
Tags	Loadcells Spec
담당자	HH 건호 이

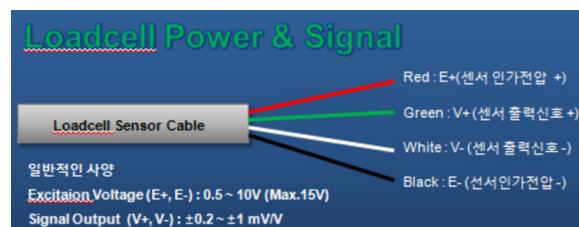
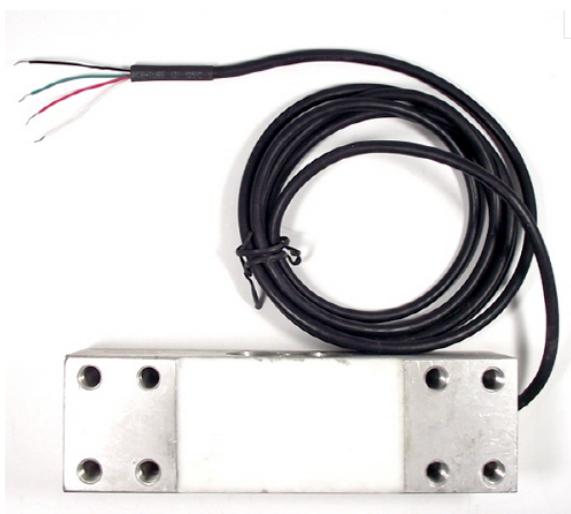
Loadcell

spec

도입

이번장에선 인수인계받은 P0293과 추후 문제로 인하여 교체한 SBA-500L, 두 모델에 대해 알아보도록 함

P0293



▼ 배선 설명

- 빨강선 : VCC
- 검정선 : GND
- 초록선 : + 신호
- 흰색선 : - 신호

▼ 주요기능

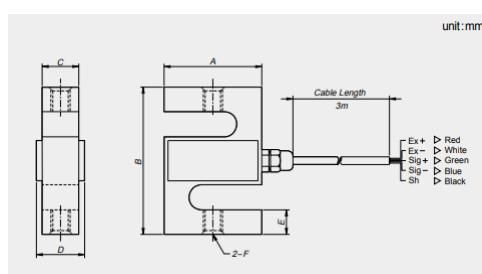
- 부하 정격 하중 : 60kg, 100kg, 150kg, 200kg, 300kg, 500kg, 800kg
- 절연 저항 임피던스 : $\geq 3000\Omega$
- 정격 출력 : $2 \pm 0.002mV/V$
- 안전 과부하 : 150 % FS
- 종합 오류 총 오류 : ± 0.02 %의 FS
- Repeatability의 Repeatability : ± 0.01 % FS
- 작동 온도 범위 작동 온도 범위 : (-30 - +70) °C
- 크리프 (30 분) 크리프 (30 분) : ± 0.02 %의 FS/30min
- 온도 보상 범위 보상 온도 범위 : (0-60) °C
- 추천 작동 전압 권장 : 5V - 12V (AC 또는 DC)
- 출력 온도 효과 : 출력 ± 0.02 % FS/10K
- 최대 작동 전압 최대 여기 15V (AC 또는 DC)
- 제로 온도 효과 : 제로에서 ± 0.02 %의 FS/10K
- 제로 밸런스 : $\pm 1\%$ FS
- 재질 : 알루미늄 합금 소재
- 입력 임피던스 : $400 \pm 10\Omega$
- 보호 건설 : IP67/IP68
- 출력 임피던스 : $351 \pm 2\Omega$
- 참조 표준 테이션 : GB/T7551-1997/OIML R60
- 연결배선 모드 : Red(전원 +), Black(전원 -), Green(신호 +), White(신호 -)
- 제품크기 : 150 x 44 x 33mm
- 가격은 50,000원 선으로 로드셀 기준 굉장히 저렴한 편의 제품
- 제조국과 OIML(International Organization of Legal Metrology 정부간 국제기구)의 인증마크가 없어 제품에 대한 신뢰도가 떨어짐

SBA-500



Capacity(R.L)	kgf	25, 50, 100, 200, 500, 1K, 2K, 3K, 5K
정격 출력(R.O.) 제로 렌즈	mV / V mV / V	3.0 ± 0.3 0.0 ± 0.03
정확도 등급	-	C3
비직선성	% R.O.	≤ 0.02
허스테리시스	% R.O.	≤ 0.02
컴바인드 에러	% R.O.	≤ 0.02
반복성	% R.O.	≤ 0.01
크립(30분)	% R.O.	≤ 0.017
크립 복귀(30분)	% R.O.	≤ 0.017
분해도	-	≤ 1/5000
한 눈의 값	mV / V	0.0006
온도 영향	- 영점 - 출력	≤ 0.028 ≤ 0.015
인가전압	- 적정 인가전압 - 최대 인가전압	V V
저항	- 일리저항 - 출력저항 - 절연저항	Ω Ω MΩ
온도 보상 범위	℃	-10 to +40
허용온도 범위	℃	-30 to +80
재질 및 표면처리	-	스틸
케이블 시양	-	Ø 5.4 x 4P x 3m (Urethane)
허용 과부하	% R.L.	150% R.L.

- 가격은 100,000원 선으로 로드셀 기준 중저가형의 제품
- OIML의 인증을 받아 C3의 정확도 등급을 획득함
- 제품의 오차율(컴바인드 에러)은 0.02%정도로 환자의 움직임을 파악하기 위한 이번 프로젝트에서 높은 수준의 신뢰도임을 알 수 있음



▼ 배선 설명

- 밸브선 : VCC
- 흰색선 : GND
- 초록선 : + 신호
- 파랑선 : - 신호
- 검정선 : 쉴드 → 의미없음

결론

- 개선 후 상대적으로 신뢰도 있는 값을 받아들일 수 있는 로드셀을 사용함을 알 수 있음

2.3.2_LoadCell_종류별차이

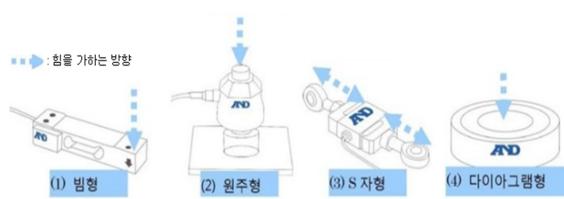
Created Time	@Sep 21, 2019 9:04 PM
Document Version	Ver 1.0.0
Last Edited By	HH 건호 이
Last Edited Time	@Sep 22, 2019 4:35 PM
Tags	Loadcells 종류별차이
담당자	HH 건호 이

LoadCell

종류별 차이

도입

로드셀에는 다음과 같이 4가지의 종류가 있음



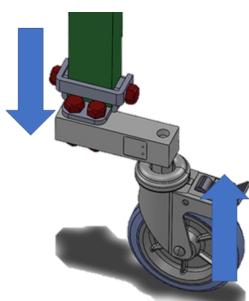
이번 프로젝트에선 빔형과 S형을 분석

원주형은 호버, 탱크 등 큰 부피와 무게가 큰 물체들에 어울리며 안정성이 좋음

또한 크기도 다른 로드셀에 비해 커서 산업용으로 주로 사용됨

다이아그램형은 크기가 작아 저울 혹은 범용으로 사용됨

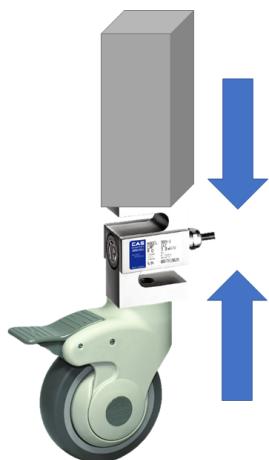
빔형



프로젝트에서 사용된 빔형 로드셀

고정된 축(바퀴쪽)과 힘을 가하는 축(침대프레임 쪽)이 서로 달라 로드셀 중간의 스트레인 게이지를 구부려 저항을 발생, 변하는 전압값을 측정

S형



S자형 로드셀의 측정 방식은 빔형 로드셀의 기본 원리를 가지고 있음 빔형 로드셀 위에 'ㄱ'자 프레임, 아래엔 'ㄴ'자 프레임을 덧댄 형태

따라서 빔형 로드셀과 마찬가지로 중간에 있는 스트레인 게이지에 하중을 주어 변화된 전압을 측정하여 무게의 변화를 알 수 있게 됨

빔형 VS S형

빔형 로드셀 사용 시 2가지의 큰 문제가 발생함

1. 침대 위치에 따라 변하는 사람 몸무게
2. 빔형 로드셀 특성상 달라지는 무게축으로 인한 불안정성

1번의 문제점은 로드셀을 4개 사용할 때 나타나는 오류로써 빔형 로드셀의 문제점이 아님

따라서 2번의 문제점을 해결하고자 무게 축이 같은 S형 로드셀을 사용하기로 함

2.4_의료용침대_규격

Created Time	@Sep 17, 2019 6:41 PM
Document Version	Ver 1.0.0
Last Edited By	HH 건호 이
Last Edited Time	@Sep 22, 2019 4:36 PM
Tags	규격 의료용침대
담당자	현욱 강

의료용침대

규격

- 적용범위 : 의료기관에서 환자의 진료를 위하여 사용하는 성인용 침대에 대하여 규정
- 구성 : 본체, 사이드레일, 매트리스 등으로 구성

수동식의 규격

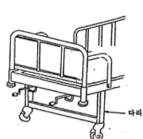
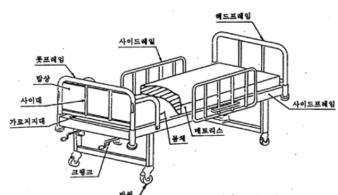


그림 2

표 2 바퀴 또는 다리 지지부의 구분 및 호칭

호칭 또는 치수		약 호
바 퀴	지름 75mm	S
	지름 100mm	M
	지름 125mm	L
침대 다리 지지부		N

본체 및 사이드레일 치수

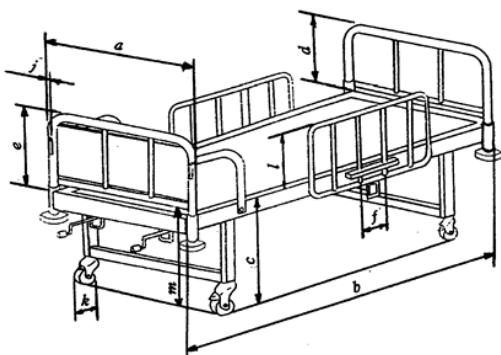


그림 5 본체 각 부분의 치수(크랭크 침대)

표 6 본체 및 사이드레일의 치수

단위 : mm

항 목	호칭치수	실제치수와 허용범위	양 호	비 고
본체의 나비	850	850±20	85	그림 3의 a
	900	900±20	90	그림 5의 a
	1000	1000±20	100	
본체의 길이	2100	2100±20	A	그림 3의 b
	2200	2200±20	B	그림 5의 b
	2300	2300±20	C	
종체의 높이(바닥에서 종체윗면까지)	300	300±20	3	
	400	400±20	4	
	500	500±20	5	
종체의 높이(종체의 뒷면에서 종체윗면까지)	550	550±20	5.5	그림 3의 c
	600	600±20	6	그림 5의 c
	최저 400 ₀ ²⁵	-	-	
율 형	최고 600 ₀ ²⁵	-	-	
	500±10	-	-	그림 3의 d 그림 5의 d
헤드프레임의 높이 (종체의 뒷면에서 헤드프레임의 위쪽까지)	-	400±10	-	그림 3의 e
풋프레임의 높이(평침대)	-4	500±10	-	그림 5의 e
풋프레임의 높이(크랭크침대) (종체의 뒷면에서 풋프레임의 위쪽까지)	-	400±0.5 190±0.5	-	그림 3의 f 그림 5의 f
사이드레일 연결구멍 간격(평침대)	-	20±0.3	-	그림 4의 g
사이드레일 연결구멍의 지름(평침대)	-	18±0.3	-	그림 6의 g
사이드레일 연결구멍의 높이(평침대)	-	50 이상	-	그림 4의 h 그림 6의 h
사이드레일 연결구멍의 높이(크랭크침대)	-	28~35	-	그림 4의 i
사이드레일 저지대의 두께(평침대)	-	25~35	-	그림 6의 i
嘲대 부착구멍의 지름(평침대)	-	13.5±0.5	-	그림 3의 j
嘲대 부착구멍의 지름(크랭크침대)	-	16±0.5	-	그림 5의 j
바퀴의 호칭치수(⁹)	75(⁸) 100 125	- - -	S M L	그림 3의 k 그림 5의 k
사이드레일의 높이(사이드레일의 뒷면에서 사이드레일의 위쪽까지)	-	350 이상	-	그림 3의 l 그림 5의 l
바닥에서 헤드프레임 및 풋프레임의 가로저지대 아랫면까지의 높이	-	350 이상	-	그림 3의 m 그림 5의 m

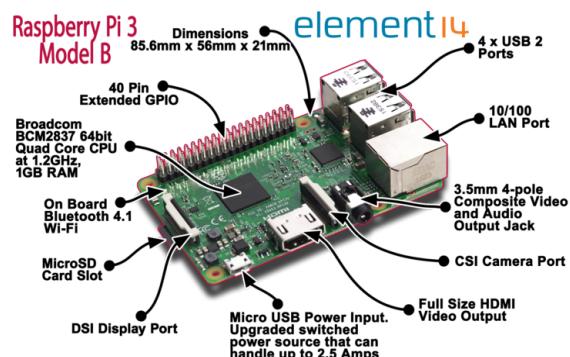
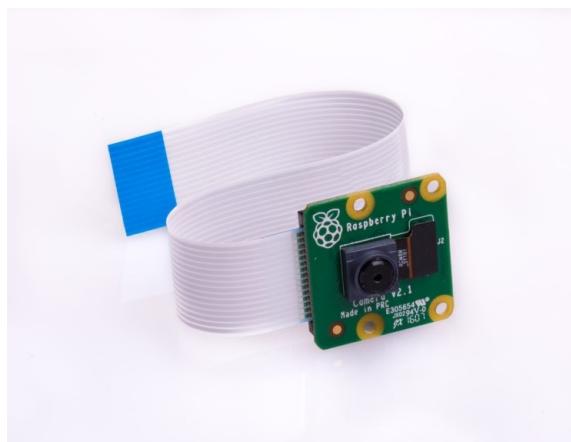
참고문헌

[1] 의료기기 기준규격, 식품의약품안전처 고시 제2017-16호(2017. 3. 8, 개정)

2.5_RaspberryPi_Camera

Created Time	@Sep 18, 2019 8:05 PM
Document Version	Ver 1.0.0
Last Edited By	HH 건호 이
Last Edited Time	@Sep 22, 2019 2:48 PM
Tags	Camera RaspberryPi
담당자	현욱 강

RaspberryPi Camera



라즈베리파이를 보면 카메라 모듈을 연결할 수 있는 단자가 있음

→ CSI(Camera Serial Interface) 포트

비슷하게 DSI(Display Serial Interface) 포트도 존재

→ 라즈베리파이 디스플레이 연결할 때 사용

카메라 모듈을 사용하려면 라즈베리파이 CSI 포트에 카메라 모듈 연결과정

* 연결하기에 앞서 주의사항

- 카메라 모듈을 연결할 때에는 라즈베리파이의 전원 연결을 끊고 연결

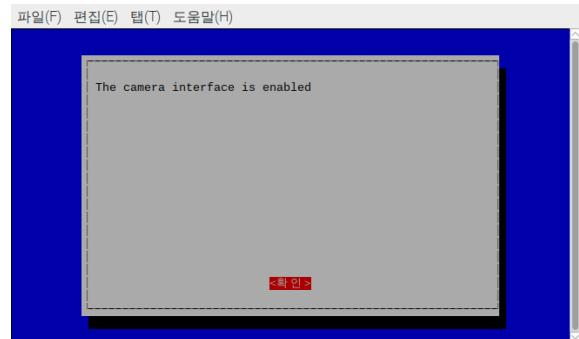
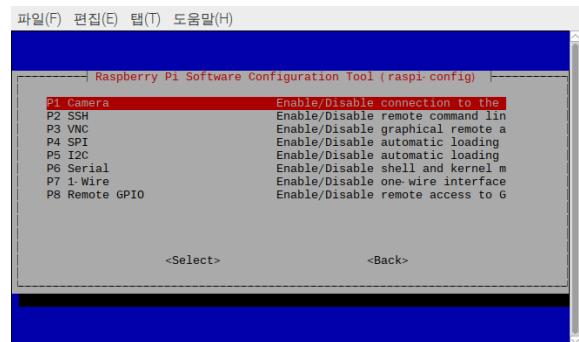
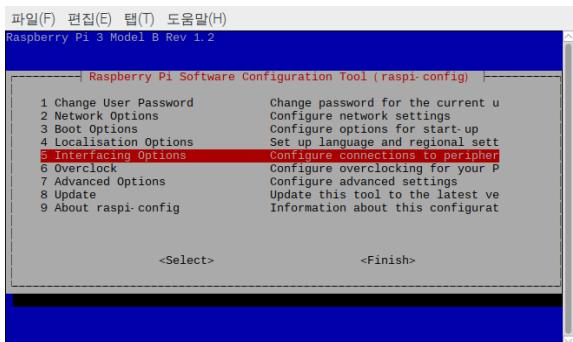


카메라 모듈의 끝을 보면 흰부분과 파란색 부분이 있는데

흰부분을 HDMI 단자 방향으로 하여
라즈베리파이에 연결

카메라 모듈 연결이 끝났으면, 라즈베리파이를 켜서 카메라 설정 필요

```
sudo raspi-config
```



다시 부팅이 됨

모듈을 사용 방법

```
raspistill -o test.jpg
```

test.jpg로 현재 디렉토리에 저장

- 옵션
 - 상하반전 : -vf
 - 좌우반전 : -hf

```
raspivid -o vid.h264
```

5초 정도 촬영

파이썬을 이용해 촬영

```
sudo nano camera.py
```

```
from picamera import PiCamera

from time import time

camera = PiCamera()

camera.start_preview()

sleep(5)

camera.capture('/home/pi/capture.jpg')

camera.stop_preview()
```

위 코드에서 `start_preview()` 부분이 실행되면 모니터에 카메라 시점이 출력

`sleep(5)` 코드에 의해 5초 쉬고, (모니터에 카메라 시점이 출력되는 중)

`capture('/home/pi/capture.jpg')` 코드에 의해 괄호 안의 디렉토리에
`capture.jpg`라는 파일로 사진이 저장

`stop_preview()` 부분이 실행되면서 카메라가 시점이 종료

* `sleep()`

카메라 모듈에 `capture()` 명령을 주기 전에 `sleep()` 명령을 입력한 이유는,
카메라가 촬영을 하기 위해 약간의 준비시간이 필요하기 때문

* `import`의 방법

`import`의 방법에는 크게 두가지 방식이 있음

[`import 모듈명`] 방법과 [`from 모듈명 import 메소드`] 방법이 있음

첫번째 방법은 모듈 전체를 불러오는 것이고, 두번째 방법은 모듈의 특정 부분만
불러옴

큰틀에서 보면 첫번째 방법이 두번째 방법 또한 포함하기 때문에 편해 보이지만,
사용방법이 다름

첫번째 방법으로 `import`를 하면 메소드를 사용할 때 모듈의 이름을 붙여주어야함

두번째 방법으로 `import`를 하면 메소드의 이름만 사용하면 됨

위의 코드를 예시로 들면

첫번째 방법일 때

```
camera = picamera.PiCamera()
```

두번째 방법일 때

```
camera = PiCamera()
```

3.1_PC프로그램

Created Time	@Sep 18, 2019 12:15 AM
Document Version	Ver 1.0.0
Last Edited By	준혁 김
Last Edited Time	@Sep 22, 2019 4:13 PM
Tags	C# Summary Windows
담당자	준혁 김

PC프로그램

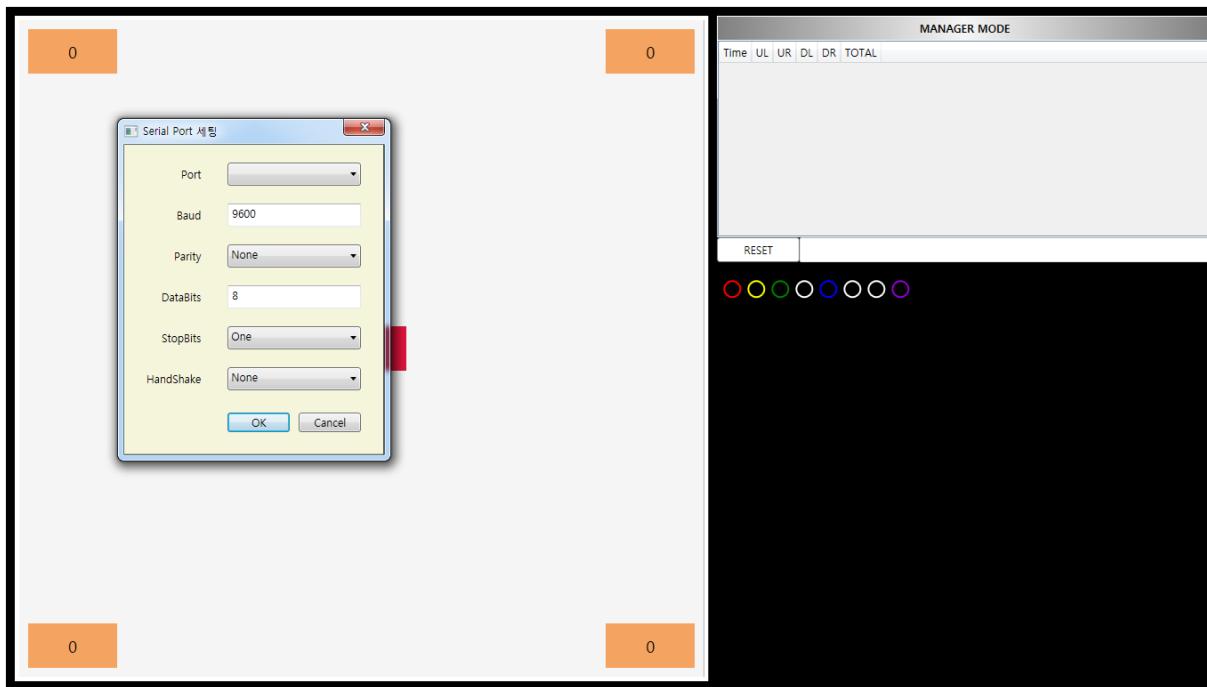
본 프로그램 설명은 필요없는 기능은 삭제하고, 낙상 및 욕창 판단을 위해 여러 차트 기능을 추가한 분석용 프로그램 'fallbed_ver1.1.0'에 대한 설명임을 명시함

실행환경

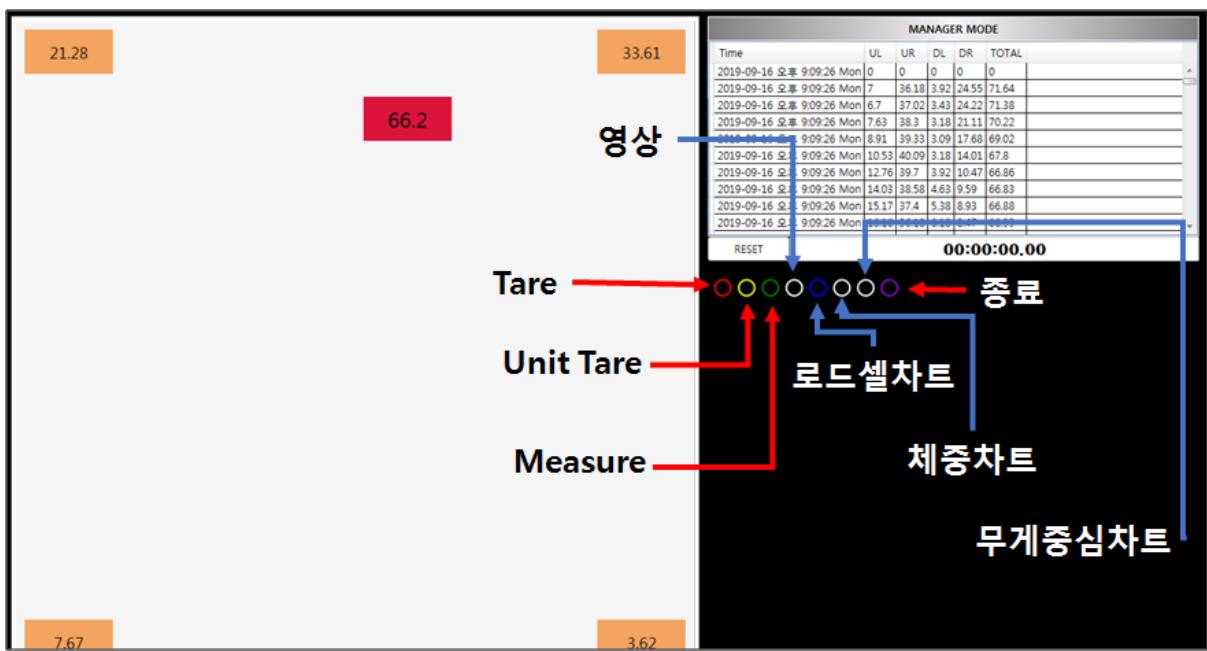
- IDE : Visual Studio 2019
- .NET Framework Version : 4.7.2
- 프로그램에 사용된 패키지는 SPRS(프로젝트 폴더)\bin\debug 또는 release 폴더 내에 dll파일로 존재함

기능구성

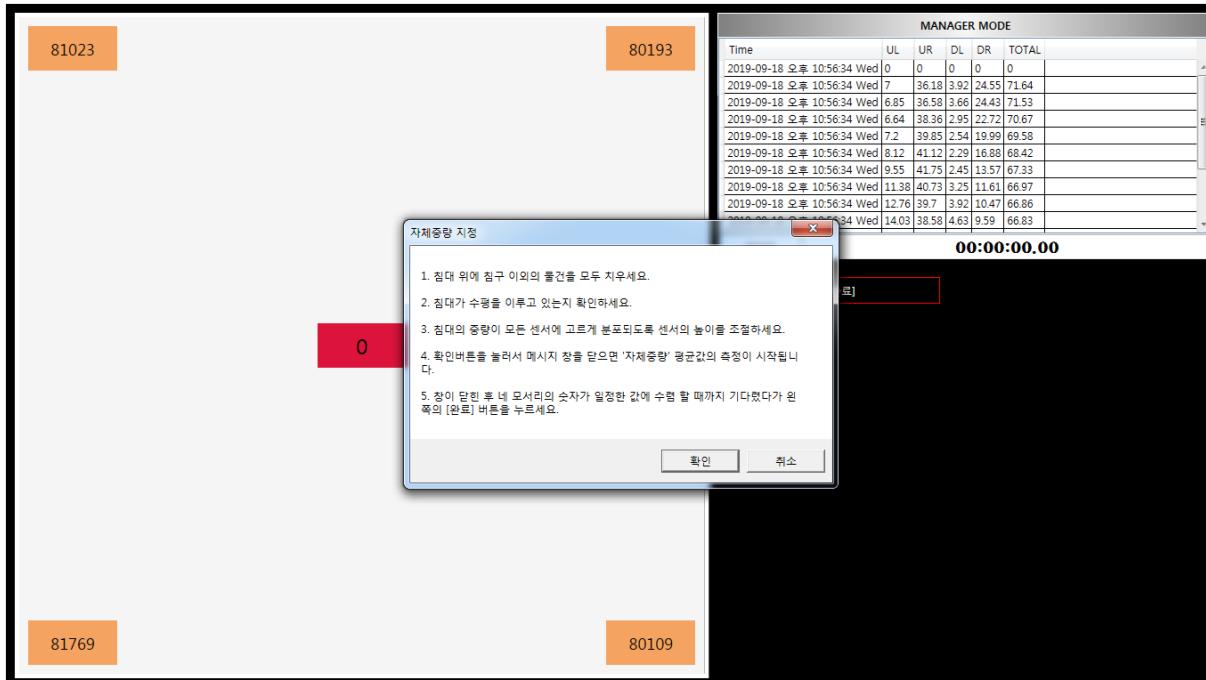
- 프로그램 시작 시 시리얼통신 설정창이 출력됨, 보통 Port항목만 설정함



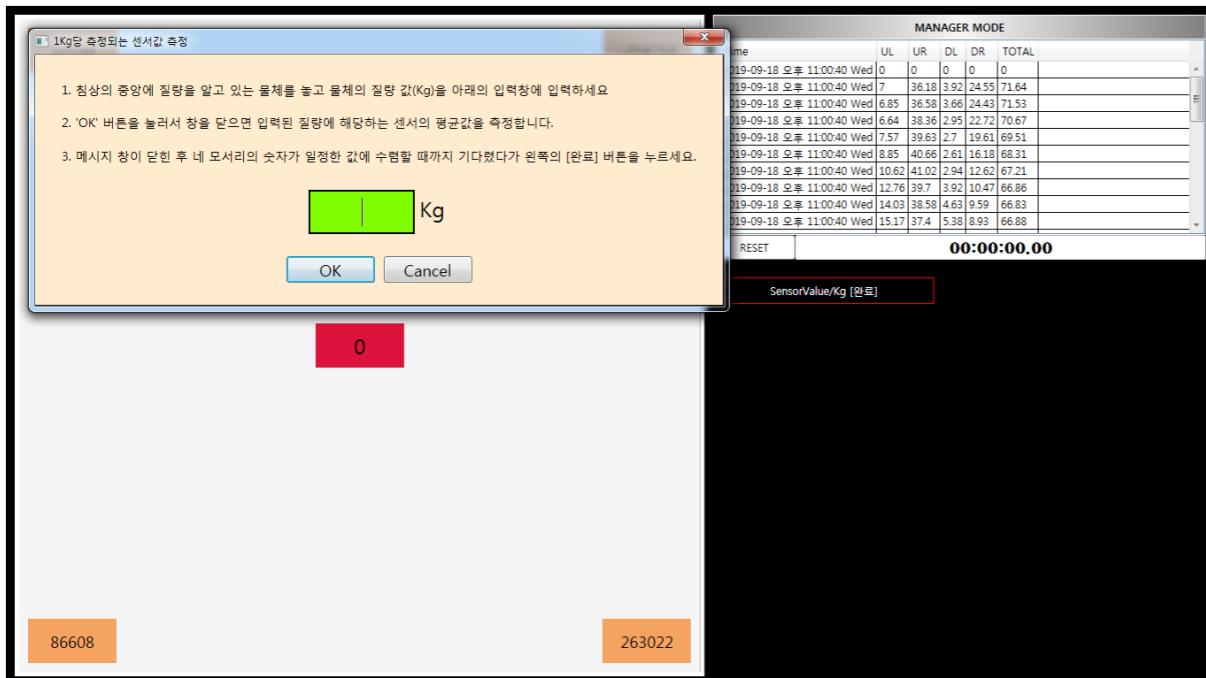
- 메인화면 : 기본적으로 네 개의 로드셀 값과 체중, 무게중심이 표시됨
또한 우측에는 데이터를 테이블 형태로 출력하고, 모드설정을 위한 버튼이 있음



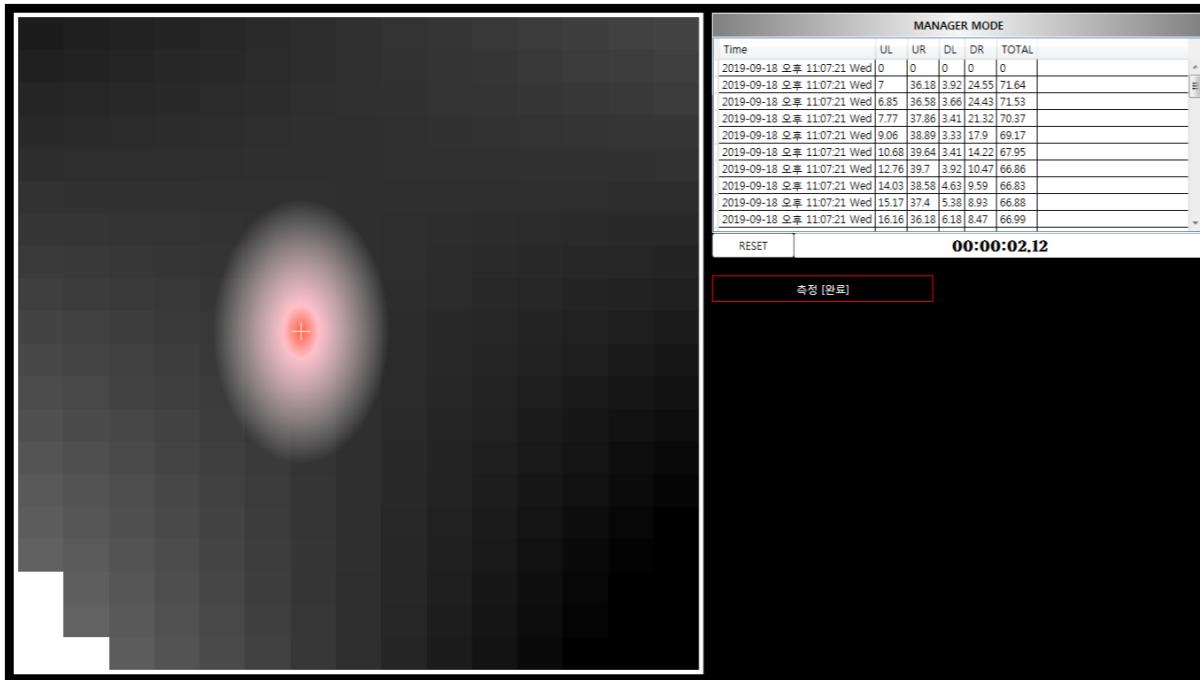
- Tare모드 : 침대자체중량을 측정함, 침대 위에 아무것도 올려놓지 말 것



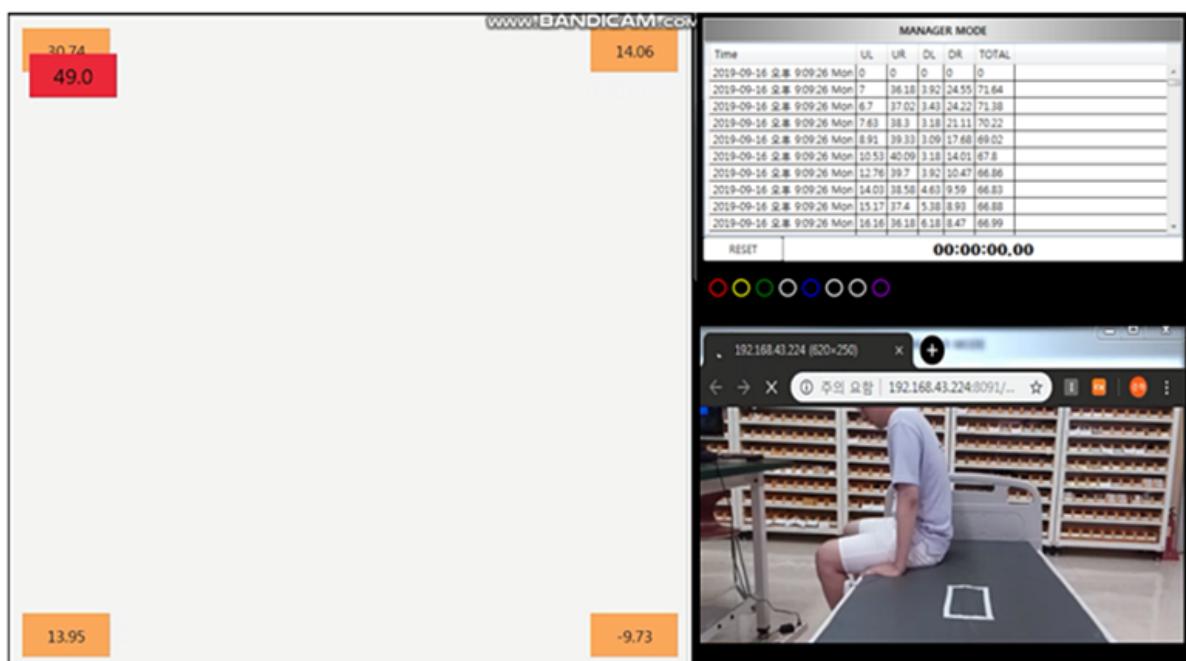
- Unit Tare모드 : 로드셀 센서는 가해진 하중에 대해 전압값을 출력하므로 무게를 알고 있는 무게추를 따로 측정하여 kg변환을 하기 위함



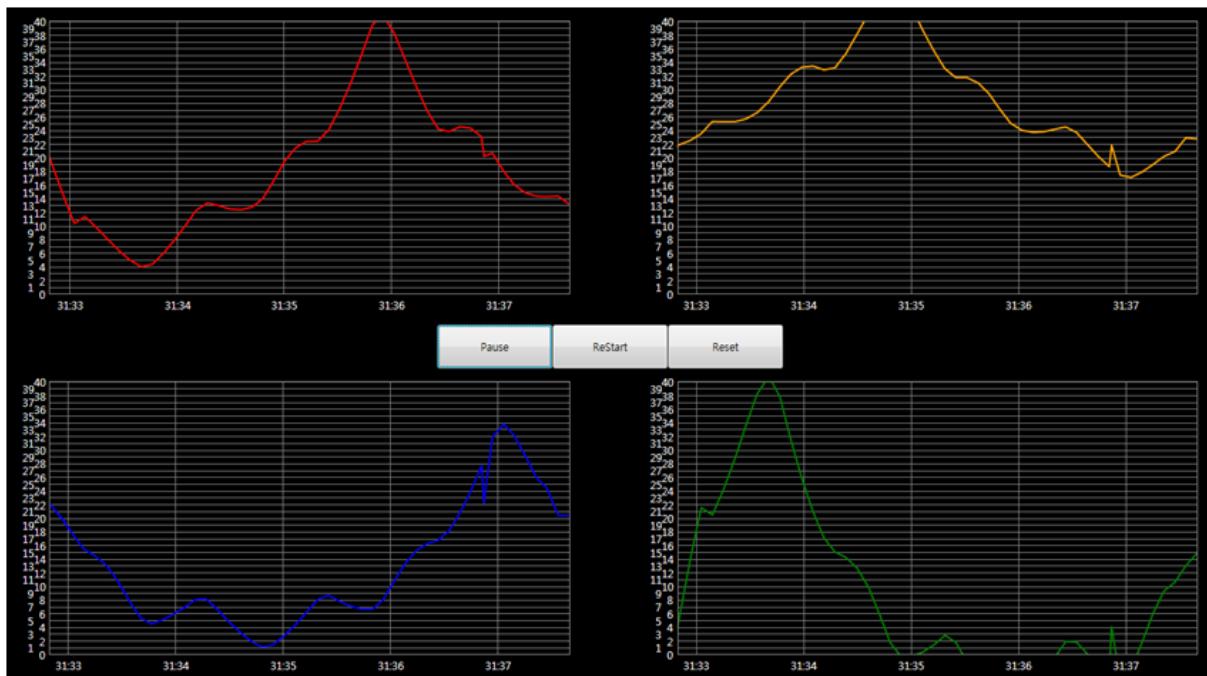
- Measure모드 : 환자의 대략적인 자세를 알기 위한 모드



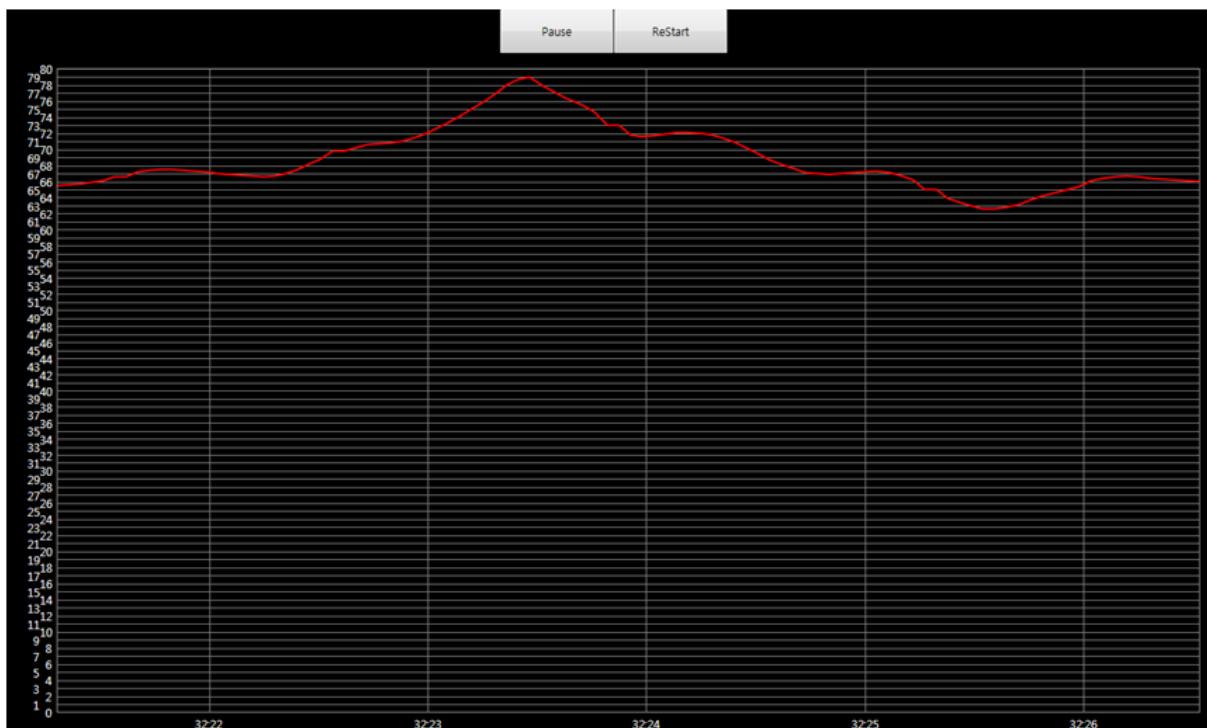
- 영상출력모드 : 라즈베리파이에 연결된 카메라를 통해 침대 위의 환자를 실시간 모니터링함, 카메라 영상은 `mjpg-streamer`를 통해 웹에 스트리밍되는데 UI 내에 삽입하기 어려워서 웹을 출력하는 것으로 대신함



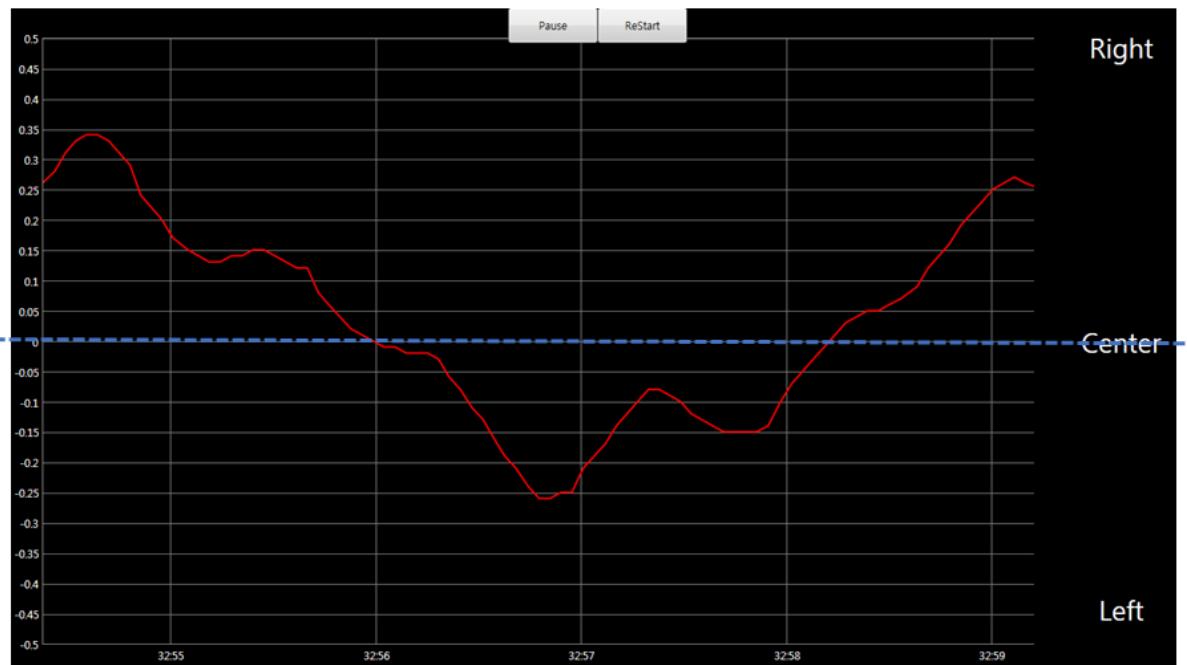
- 로드셀차트모드 : 네 개의 로드셀 값에 대한 차트, 정지\재시작\초기화 가능



- 체중차트모드 : 오직 체중의 변화를 분석하기 위한 차트, 정지\재시작 가능



- 무게중심차트모드 : 무게중심 변화를 좌우에 한해 분석하기 위한 차트, 정지 \재시작 가능



3.1.1_PC프로그램_DFD

Created Time	@Sep 21, 2019 4:58 PM
Document Version	Ver 1.0.0
Last Edited By	준혁 김
Last Edited Time	@Sep 22, 2019 4:15 PM
Tags	C# DFD Windows
담당자	준혁 김

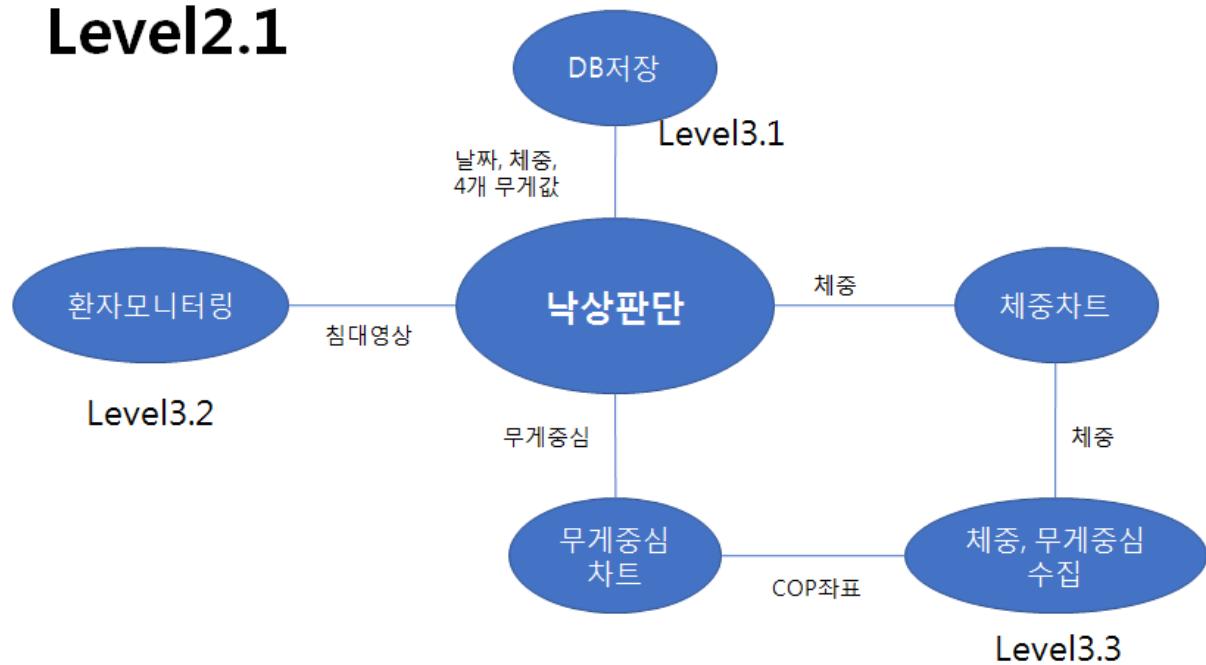
PC프로그램

DFD - 기능별 레벨링

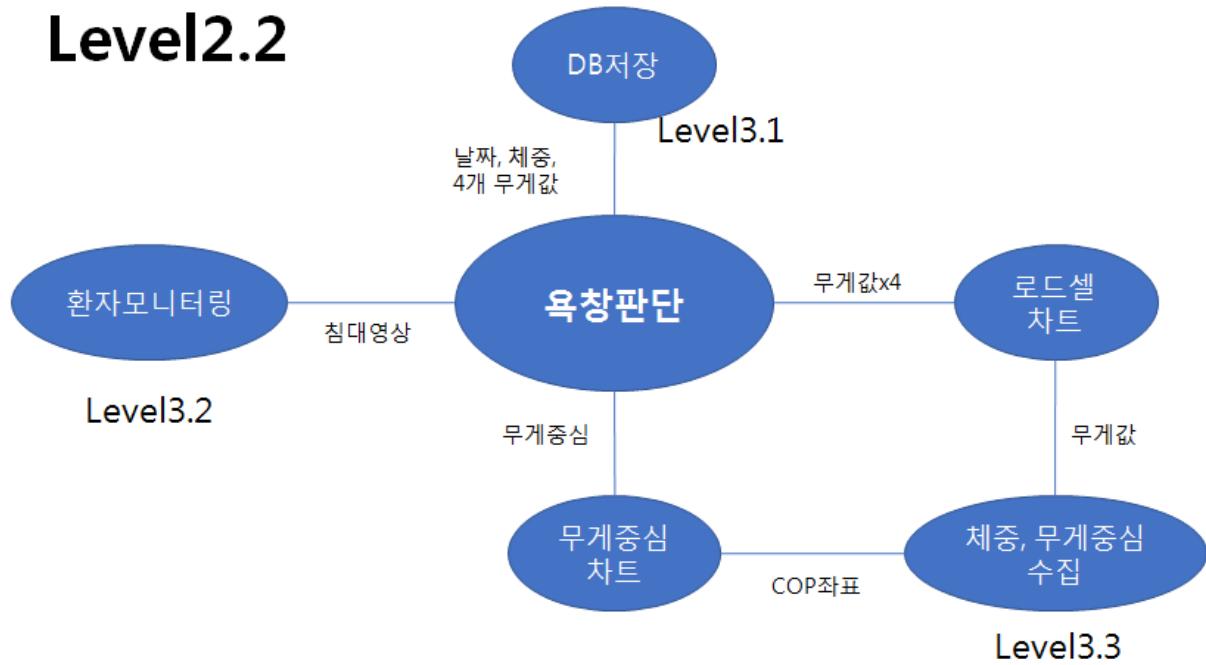
Level1



Level2.1



Level2.2



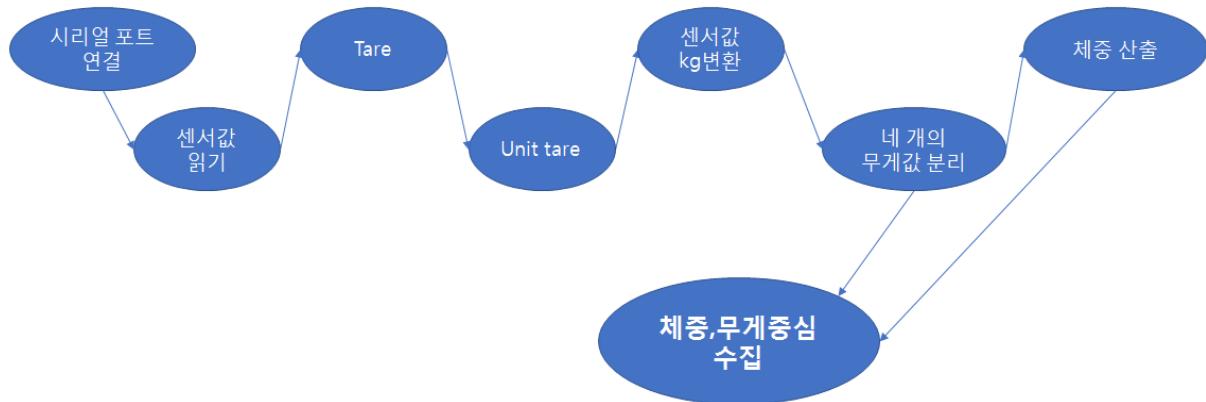
Level3.1



Level3.2



Level3.3



3.1.2_PC프로그램_데이터수집

Created Time	@Sep 21, 2019 6:16 PM
Document Version	Ver 1.0.0
Last Edited By	준혁 김
Last Edited Time	@Sep 22, 2019 3:22 PM
Tags	C# DataRead Windows
담당자	준혁 김

PC프로그램

데이터수집

```
SPRS_Main.cs
// 데이터를 가져오는 함수
void dataReader(object obj)
{
    // SerialPort 객체 serial을 obj로 초기화 -> 포트연결 위함
    // byte[1]형 변수 temp에 CMD_DATA 저장 -> 데이터를 보내라고 아트메가에 요청하기 위함
    // 포트연결이 되어있지 않으면 랜덤값을 가져오는 testData() 호출

    // 포트연결이 되어있다면 temp를 매개변수로 serial.write() 호출하여 데이터 요청
    // Mutex.WaitOne() 호출 -> 쓰레드를 소유하여 자원접근에 대한 충돌방지
    // 받은 데이터는 인스턴스변수 buffer에 복사 -> 데이터를 무게로 변환하여 사용
    // Mutex 해제
}
```

- Mutex

System.Threading 네임스페이스
프로세스 간 동기화에 사용
WaitOne() : 현재 쓰레드 차단, 호출 쓰레드 소유
ReleaseMutex() : Mutex 해제

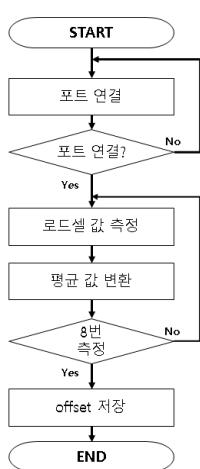
3.1.3_PC프로그램_Tare

Created Time	@Sep 18, 2019 10:28 PM
Document Version	Ver 1.0.0
Last Edited By	준혁 김
Last Edited Time	@Sep 22, 2019 3:32 PM
Tags	C# Tare Windows
담당자	준혁 김

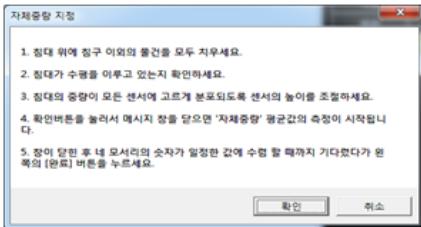
PC프로그램

Tare

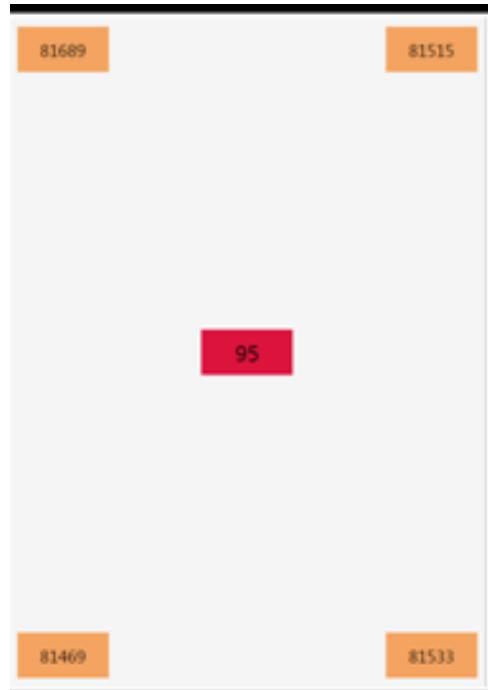
- 침대자체중량을 측정하는 모드
- Tare버튼을 눌러 현재 프로그램 실행모드를 Tare모드로 지정하면 동작함



```
SPRS_main.cs
// Tare : 침대자체중량을 측정함
void getTareMass(double[] data)
{
    // 현재모드가 Tare_Info모드이면
    // 캔버스는 네 개의 무게값 라벨에 data 출력(로우데이터)
    // 현재모드가 Tare모드이면
    // samplingCount 증가, Tare시작
    // Tare가 시작되면 네 개의 무게값 라벨의 값이 조정되어 변화함
    // 변화가 적어지면 '완료'버튼으로 중지시켜도 됨
}
```



Tare_Info모드 : Tare버튼 눌렀을 때, 알림창 나오는 모드



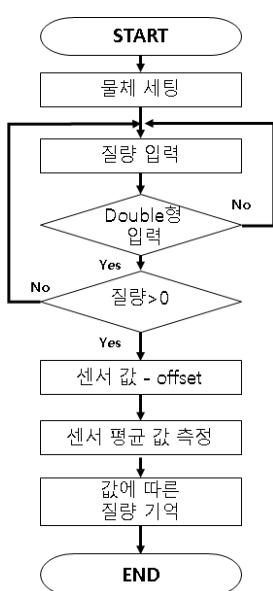
3.1.4_PC프로그램_UnitTare

Created Time	@Sep 18, 2019 10:43 PM
Document Version	Ver 1.0.0
Last Edited By	준혁 준혁 김
Last Edited Time	@Sep 22, 2019 3:33 PM
Tags	C# UnitTare Windows
담당자	준혁 준혁 김

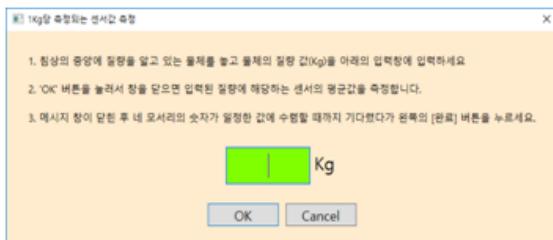
PC프로그램

UnitTare

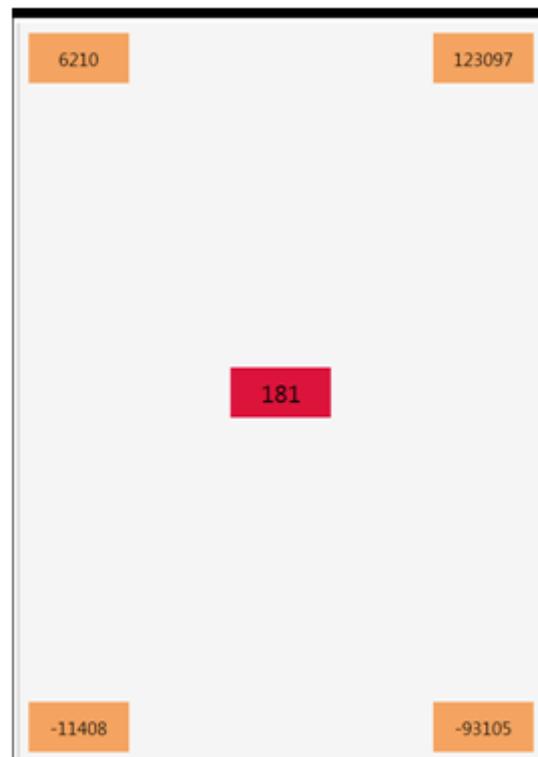
- kg당 로드셀 값을 알기 위한 모드
- UnitTare버튼을 눌러 현재 프로그램 실행모드를 UnitTare모드로 지정하면 동작함
- 무게를 알고 있는 무게추를 침대 위에 올려 측정 시작



```
SPRS_main.cs
// 로드셀값을 kg변환하기 위해 무게기준단위를 구하기 위한 함수
void getUnitMass(double[] data)
{
    // data에서 tare값(offset)을 뺀 - 침대자체중량을 기준으로 하기 위함
    // 현재모드가 Unit_Info모드이면
    // 캔버스는 네 개의 무게값 라벨에 data 출력(로우데이터)
    // 현재모드가 Unit모드이면
    // samplingCount 증가, Unit tare시작
    // Unit tare가 시작되면 네 개의 무게값 라벨의 값이 조정되어 변화함
    // 변화가 적어지면 '완료'버튼으로 중지시켜도 됨
}
```



Unit_Info모드 : 기준 무게추의 무게 입력



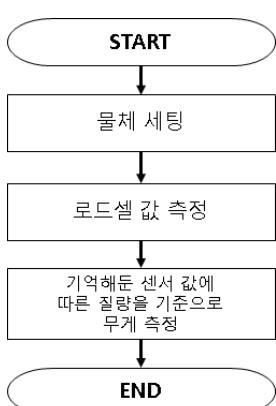
3.1.5_PC프로그램_Measure

Created Time	@Sep 21, 2019 5:45 PM
Document Version	Ver 1.0.0
Last Edited By	현욱 강
Last Edited Time	@Sep 22, 2019 3:33 PM
Tags	C# Measure Windows
담당자	준혁 김

PC프로그램

Measure

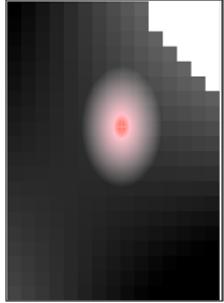
- 무게중심을 파악하기 위한 모드
- Measure버튼을 눌러 현재 프로그램 실행모드를 Measure모드로 지정하면 동작함



```
SPRS_main.cs
// kg단위의 무게값과 무게중심을 파악하는 함수
void getCenterOfMass(double[] data)
{
    // Unit Tare가 되어있지 않으면 return
    // data에서 tare값을 빼고 unit값으로 나눔으로써 kg단위의 무게값 산출
    // 체중을 저장하는 변수 totalMass에 data들 더함
    // 체중이 1.5미만으로 판단되면 data를 0으로 바꿈

    // 현재모드가 IDLE이면 data를 UI의 라벨에 출력
    // 체중이 최소체중기준(1.0)보다 크면 낙상플래그를 True로 설정
    // (현재 환자가 침대에 있음을 의미)
    // 무게중심 좌표 계산
    // 현재 무게중심 좌표를 copX, copY에 저장

    // Mutex를 통해 현재 쓰레드를 차단하고 호출 쓰레드를 소유함
    // 시간변화를 판단하는 TimeChanged()를 호출하여 일정시간 경과하였으면
    // 인스턴수변수 saveCOP에 현재 무게중심 좌표 저장
    // Mutex 해제
    // calmass2(data1) 호출하여 무게중심 이외의 부분을 검게함
}
```



```
//--질량 중심 계산-----  
//scaleBoardLeft = -(scaleBoardWidth / 2.0);  
//scaleBoardRight = scaleBoardWidth / 2.0;  
//scaleBoardTop = scaleBoardHeight / 2.0;  
//scaleBoardBottom = -(scaleBoardHeight / 2.0);  
//data[0] : UpperLeft  
//data[1] : DownLeft  
//data[2] : DownRight  
//data[3] : UpperRight  
  
copX = ((data[0] + data[2]) * scaleBoardLeft  
        + (data[1] + data[3]) * scaleBoardRight) / totalMass;  
  
copY = ((data[0] + data[1]) * scaleBoardTop  
        + (data[2] + data[3]) * scaleBoardBottom) / totalMass;
```

$$x kg = \frac{Sensor - Tare}{Unit}$$

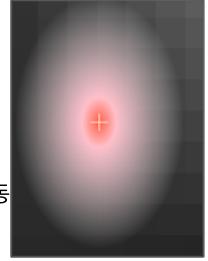
Sensor: 로드셀 센서값,

Tare: 초기 tare값(침대 자체중량),

Unit: 무게추을 올렸을 때 센서값

SPRS_main.cs

```
// 무게중심 위치를 판단하고 그리는 함수  
void copSeeker()  
{  
    // Point 객체 target 생성  
    // Mutex를 통해 현재 쓰레드를 차단하고 호출 쓰레드를 소유함->무게중심 위치파악 프로세스 동  
    // target의 위치좌표 저장  
    // 캔버스에 무게중심을 그리기 위한 인스턴스변수 cursorPos에 target 대입  
    // moveCursor() 함수 호출 - 캔버스에 무게중심을 그리는 함수  
    // Mutex 해제  
}
```



3.1.6_PC프로그램_영상출력

Created Time	@Sep 21, 2019 6:17 PM
Document Version	Ver 1.0.0
Last Edited By	준혁 김
Last Edited Time	@Sep 22, 2019 4:21 PM
Tags	C# Monitoring Windows
담당자	준혁 김

PC프로그램

영상출력

- 라즈베리파이에 장착된 카메라가 촬영하는 영상을 받아와 출력하는 모드
- 영상이 있는 크롬 브라우저를 출력함 - 프로젝트 폴더 안에 있는 'alwaysontop-mrjob' 프로그램을 실행하여 크롬 창이 항상 맨위에 있게 설정할 것

```
void cameraBtn_Click(object sender, MouseButtonEventArgs args)
{
    // 새 Process 생성
    Process process = new Process();

    // Process에서 실행할 프로그램의 파일 이름 지정
    process.StartInfo.FileName = "chrome.exe";

    // 실행파일에 전달할 요소입력
    process.StartInfo.Arguments = "라즈베리파이주소/?action=stream";

    // 출력되는 윈도우창 스타일 지정
    process.StartInfo.WindowStyle = ProcessWindowStyle.Normal;
    process.Start();
}
```

3.1.7_PC프로그램 _Loadcells_Chart

Created Time	@Sep 21, 2019 5:46 PM
Document Version	Ver 1.0.0
Last Edited By	준혁 준혁 김
Last Edited Time	@Sep 22, 2019 3:34 PM
Tags	C# Loadcells Windows
담당자	준혁 준혁 김

PC프로그램

로드셀차트

- 네 개의 무게값을 이용해 차트를 그리는 모드
- 로드셀차트 버튼을 눌러 실행
- WPFToolkit패키지를 사용하여 실시간 차트를 그림
- 실시간차트를 그리는 또 다른 패키지인 LiveChart와의 차이점은 LiveChart는 자원을 많이 소모에 여러 개의 실시간차트를 그리면 렉이 걸려 본 프로그램에는 적합하지 않음

```
OnlyCharts.xaml.cs
// 네 개의 무게값 차트를 그리는 클래스
class OnlyCharts : Window
{
    // 멤버변수-----
    // ObservableCollection<ChartData>[4]형 chartData 선언 -> 네 개의 차트를 그리는데 사용
    // ChartData[4]형 objChartData 선언 -> 차트에 값을 넣기 위한 변수
    // EventWaitHandle형 waitHandle 선언 -> 차트 초기화, 일시정지, 시작을 조절하기 위함

    OnlyCharts() // 생성자
    {
        // 차트의 크기를 조절하는 부분
        // 차트를 그리기 위한 초기설정(데이터바인딩, 최소값 지정 등)
        // StartChartDataSimulation 쓰레드 시작 -> 차트를 그리는 쓰레드
    }

    // 메소드-----
    void StartChartDataSimulation()
    {
```

```
// Dispatcher.Invoke 호출하여 차트 갱신
// -> UI를 수정하는 것은 메인쓰레드만 가능하므로 Dispatcher.Invoke를 통해 메인쓰레드가 UI를 수정하도록 함
// 처음 50개의 데이터는 수집 후, x축값이 1초 간격으로 계속 갱신됨
}
}
```

```
OnlyCharts.xaml.cs
// 차트를 그리기 위한 차트데이터 클래스
class ChartData : INotifyPropertyChanged
{
    // 속성-----
    // DateTime Name
    // get _Name 반환
    // set _Name 갱신, OnPropertyChanged("Name") 호출로 속성값 갱신

    // double Value
    // get _Value 반환
    // set _Value 갱신, OnPropertyChanged("Value") 호출로 속성값 갱신

    // 메소드-----
    void OnPropertyChanged(string propertyName)
    {
        // PropertyChanged 이벤트 발생 시 속성값 갱신
    }
}
```

3.1.8_PC프로그램_Weight_Chart

Created Time	@Sep 21, 2019 9:08 PM
Document Version	Ver 1.0.0
Last Edited By	준혁 김
Last Edited Time	@Sep 22, 2019 3:34 PM
Tags	C# Weight Windows
담당자	준혁 김

PC프로그램

체중차트

- 네 개의 무게값의 합인 체중을 사용하여 차트를 그리는 모드
- 체중차트 버튼을 눌러 실행

```
WeightChart.xaml.cs
// 체중차트를 그리는 클래스
class WeightChart : Window
{
    // 멤버변수-----
    // ObservableCollection<ChartData>[4]형 chartData 선언 -> 네 개의 차트를 그리는데 사용
    // ChartData[4]형 objChartData 선언 -> 차트에 값을 넣기 위한 변수
    // EventWaitHandle형 waitHandle 선언 -> 차트 초기화, 일시정지, 시작을 조절하기 위함

    WeightChart() // 생성자
    {
        // 차트의 크기를 조절하는 부분
        // 차트를 그리기 위한 초기설정(데이터바인딩, 최소값 지정 등)
        // StartChartDataSimulation 쓰레드 시작 -> 차트를 그리는 쓰레드
    }

    // 메소드-----
    void StartChartDataSimulation()
    {
        // Dispatcher.Invoke 호출하여 차트 생성
        // -> UI를 수정하는 것은 메인쓰레드만 가능하므로 Dispatcher.Invoke를 통해 메인쓰레드가 UI를 수정하도록 함
        // 처음 50개의 데이터는 수집 후, x축값이 1초 간격으로 계속 갱신됨
    }
}
```

```
WeightChart.xaml.cs
// 차트를 그리기 위한 차트데이터 클래스
class WeightChartData : INotifyPropertyChanged
{
    // 속성-----
    // DateTime Name
    // get _Name 반환
    // set _Name 갱신, OnPropertyChanged("Name") 호출로 속성값 갱신

    // double Value
    // get _Value 반환
    // set _Value 갱신, OnPropertyChanged("Value") 호출로 속성값 갱신

    // 메소드-----
    void OnPropertyChanged(string propertyName)
    {
        // PropertyChanged 이벤트 발생 시 속성값 갱신
    }
}
```

3.1.9_PC프로그램_COP_Chart

Created Time	@Sep 21, 2019 9:08 PM
Document Version	Ver 1.0.0
Last Edited By	준혁 김
Last Edited Time	@Sep 22, 2019 3:35 PM
Tags	C# COP Windows
담당자	준혁 김

PC프로그램

무게중심차트

- 무게중심 좌표를 이용해 차트를 그리는 모드
- 무게중심차트 버튼을 눌러 실행
- 0을 기준으로 +가 오른쪽, -가 왼쪽에 대상이 위치함을 의미

```
LRChart.xaml.cs
// 무게중심차트를 그리는 클래스
class LRChart : Window
{
    // 멤버변수-----
    // ObservableCollection<ChartData>[4]형 chartData 선언 -> 네 개의 차트를 그리는데 사용
    // ChartData[4]형 objChartData 선언 -> 차트에 값을 넣기 위한 변수
    // EventWaitHandle형 waitHandle 선언 -> 차트 초기화, 일시정지, 시작을 조절하기 위함

    LRChart() // 생성자
    {
        // 차트의 크기를 조절하는 부분
        // 차트를 그리기 위한 초기설정(데이터바인딩, 최소값 지정 등)
        // StartChartDataSimulation 쓰레드 시작 -> 차트를 그리는 쓰레드
    }

    // 메소드-----
    void StartChartDataSimulation()
    {
        // Dispatcher.Invoke 호출하여 차트 갱신 - 단 이 클래스는 무게중심의 x좌표를 가져옴
        // -> UI를 수정하는 것은 메인쓰레드만 가능하므로 Dispatcher.Invoke를 통해 메인쓰레드가 UI를 수정하도록 함
        // 처음 50개의 데이터는 수집 후, x축값이 1초 간격으로 계속 갱신됨
    }
}
```

```
LRChart.xaml.cs
// 차트를 그리기 위한 차트데이터 클래스
class LRChartData : INotifyPropertyChanged
{
    // 속성-----
    // DateTime Name
    // get _Name 반환
    // set _Name 갱신, OnPropertyChanged("Name") 호출로 속성값 갱신

    // double Value
    // get _Value 반환
    // set _Value 갱신, OnPropertyChanged("Value") 호출로 속성값 갱신

    // 메소드-----
    void OnPropertyChanged(string propertyName)
    {
        // PropertyChanged 이벤트 발생 시 속성값 갱신
    }
}
```

3.2_침대촬영

Created Time	@Sep 21, 2019 11:11 PM
Document Version	Ver 1.0.0
Last Edited By	 현욱 강
Last Edited Time	@Sep 22, 2019 3:45 PM
Tags	침대촬영
담당자	 현욱 강

침대촬영

라즈베리파이 카메라를 연결시켜 움직임이 발생시에 이미지를 불러오는 기능에 대해 서술함

세부구성

- 라즈베리파이카메라연결 → 3.2.1

3.2.1_침대촬영_라즈베리파이_카메라_연결

Created Time	@Sep 18, 2019 9:54 PM
Document Version	Ver 1.0.0
Last Edited By	这家伙 姜
Last Edited Time	@Sep 22, 2019 3:42 PM
Tags	Camera 침대촬영
담당자	这家伙 姜

침대촬영

라즈베리파이카메라 연결

NoIR 카메라

움직임 감지 시 영상을 저장하기 위해 CSI카메라 모듈 NoIR을 사용

NoIR은 스틸 사진뿐 아니라 고화질의 비디오 영상을 얻는데 사용 가능

카메라 모듈은 5M픽셀의 해상도를 지원하며, 스틸 사진을 비롯해 1080픽셀 기반의 30프레임, 720 픽셀 기반의 60프레임, 그리고 VGA90비디오 모드를 지원

CSI 인터페이스에 리본 케이블을 통해 연결되는 카메라 모듈에 대한 접근은 MMAL과 V4L API를 통해 이루어짐

스트리밍

```
sudo apt-get update  
sudo apt-get upgrade  
업데이트부터 하자
```

웹스트리밍을 위한 동영상 스트리밍 패키지를 받고 실행

```
1. mkdir mjpg  
2. cd mjpg  
3. ~/mjpg $ git clone https://github.com/jacksonilam/mjpg-streamer.git  
생성한 폴더 디렉토리 내부로 들어가 mjpg_streamer의 git 파일을 받기  
4. ~/mjpg $ cd mjpg-streamer  
5. ~/mjpg/mjpg-streamer $ cd mjpg_streamer-experimental  
받은 파일의 디렉토리 내부로 이동  
6. ~/mjpg/mjpg-streamer/mjpg_streamer-experimental $ sudo apt-get install cmake
```

```
~/mjpg/mjpg-streamer/mjpg_streamer-experimental $ sudo apt-get install libjpeg-dev  
(~/mjpg/mjpg-streamer/mjpg_streamer-experimental $ sudo apt-get install python-imaging)  
스트리밍에 필요한 패키지를 설치  
7. ~/mjpg/mjpg-streamer/mjpg_streamer-experimental $ make CMAKE_BUILD_TYPE=Debug  
8. ~/mjpg/mjpg-streamer/mjpg_streamer-experimental $ sudo make install mjpg-streamer  
컴파일 생성
```

스트리밍을 위한 코드 작성

```
cd home #을 작성하면 home 디렉토리로 이동  
vim mjpg.sh #vim을 이용해 작성  
#입력은 i를 누르면 됨
```

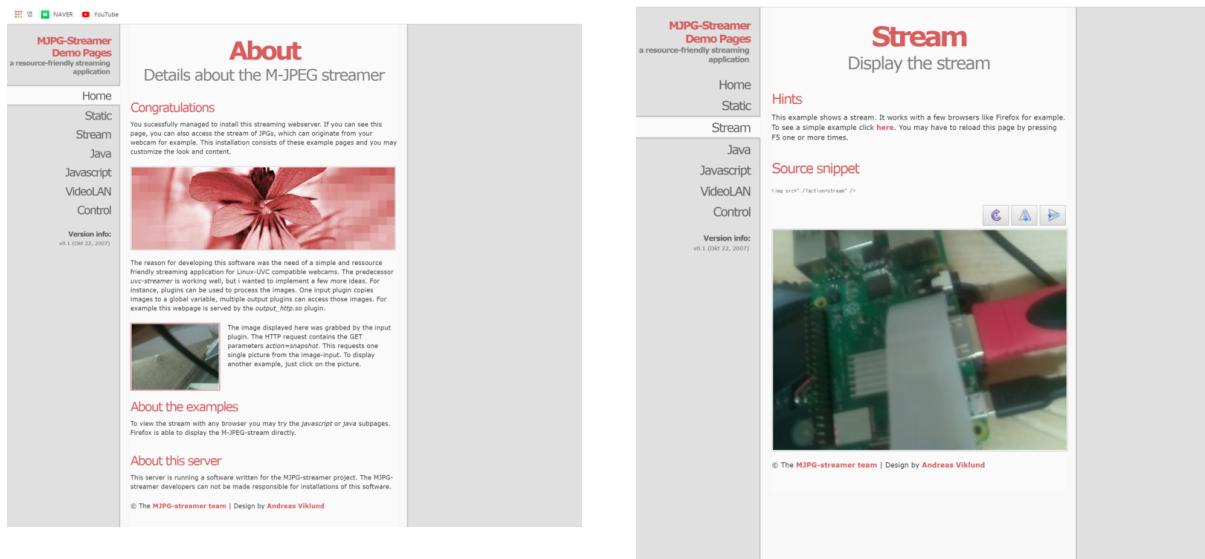
```
#mjpg.sh  
export STREAMER_PATH=$HOME/mjpg/mjpg-streamer/mjpg-streamer-experimental  
export LD_LIBRARY_PATH=$STREAMER_PATH  
$STREAMER_PATH/mjpg_streamer -i"input_raspicam.so -d 200" -o"output_http.so -w $STREAMER_PATH/www"
```

다 적었으면 ESC를 눌러 명령어 입력으로 돌아와서

```
:wq mjpg.sh #저장하고 나가기
```

이제 터미널 창에 입력

```
sh mjpg.sh #스트리밍시작! 이걸 해줘야지 웹에서 스트리밍 가능  
주소는 http://라즈베리파이아이피:8080 임  
  
라즈베리파이아이피를 모르면 터미널창에  
ifconfig  
라고 치면 나옴
```



참고

주소창에 아이피:포트번호/?action=stream으로 접속하면 스트리밍 화면만 출력됨

스트리밍을 중단은 Ctrl + Z

다시 스트리밍을 하기위해서는 이전의 프로세스를 꺼야함

```
kill -9 processNum
```

스트리밍 서버



일반 서버와 스트리밍 서버의 차이점을 나타내는 그림임

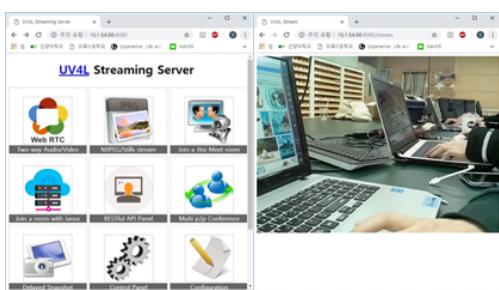
스트리밍이란 실시간 영상 압축 복원 기술을 기반으로 파일 전체가 다운로드 될 때까지 기다리지 않고 일정량의 데이터만으로 바로 동영상을 보여주는 기술

스트리밍 서버는 몇 백 메가(M)씩 하는 큰 동영상 파일을 한꺼번에 다운로드 시켜 주는 것이 아니라 부분적으로 조각을 내어 차례로 다운로드 시킨 후, 다운로드 받은 부분을 먼저 재생해줌

따라서 시청자는 전체 파일이 다운로드 완료될 때까지 기다리지 않고 바로 동영상 화면을 시청할 수 있음

20분짜리 동영상이 100M 정도 한다고 할 때, 이를 다 다운로드 받을 때까지 기다릴 필요 없이 우선 조금만 다운로드 받고 보면서 나머지를 받을 수 있음

참고 - UV4L (fallbed ver1.0.0에서 사용)



리눅스의 V4L(Video4Linux)란 리눅스에서 비디오를 캡처하고 출력하기 위한 API 및 디바이스 드라이버 프레임워크로 리눅스 커널에 위치하며 USB 웹캠, TV튜너와 같은 많은 디바이스들을 지원

UV4L(UserSpace Video4Linux)는 단순한 프레임워크로서 v4l과 호환이 되는 실제 또는 가상의 비디오 장치들을 위한 사용자 영역 드라이버를 제공

UV4L모듈 및 UV4L용 라즈베리파이 카메라 드라이버를 설치하기 위해서 Linprojects의 저장소를 APT 리스트에 추가해야 한다. 먼저 해당 저장소로부터 PGP키를 다운로드 받고, APT 키 리스트에 추가

실제 0.35초 정도의 딜레이 밖에 없어 가장 CCTV에 적합하다 판단하여 사용함

3.3_Database

Created Time	@Sep 21, 2019 5:12 PM
Document Version	Ver 1.0.0
Last Edited By	준혁 김
Last Edited Time	@Sep 22, 2019 4:03 PM
Tags	Database
담당자	현욱 강

Database

Database에 관한 것으로 MySql로 table, column 생성등의 기본 명령어와 data를 다루기 위한 Workbench 설치 및 사용법, phpMyAdmin으로 관리를 서술

프로그램과 연동되어 db가 저장되는 방식과 카메라의 사진이 출력되는 부분을 담당

세부구성

- MySql → 3.2.1
- Workbench → 3.2.2
- phpMyAdmin → 3.2.3
- TCP 데이터 전송 형식 → 3.2.4
- Camera연동 → 3.2.5

3.3.1_Database_MySql

Created Time	@Sep 18, 2019 4:01 PM
Document Version	Ver 1.0.0
Last Edited By	준혁 김
Last Edited Time	@Sep 22, 2019 4:34 PM
Tags	Database MySQL
담당자	현욱 강

Database

MySQL

데이터베이스 정의

데이터베이스는 이제 모든 사람들의 생활에서 없어서는 안될 매우 중요한 부분이 되었음

데이터베이스가 없다면 우리가 하고 있는 많은 일들은 매우 지루하고, 불편한 일들이 될것임

은행과 대학, 도서관은 이러한 데이터베이스 시스템에 크게 의존하고 있는 기관의 예이며, 인터넷의 서치 엔진, 온라인 쇼핑몰또한 데이터베이스를 이용하여 구축됨

데이터베이스는 하나의 컴퓨터가 데이터베이스 서버와 개방적인 상호 관계를 할 수 있도록 도와주는 도구 이자 인터페이스의 역할을 함

T.c.X. DataKonsultAB에 의해 개발된 MySQL 서버는 현재 시장에서 급속하게 성장하고 있는 SQL(Structured Query Language) 데이터베이스 서버임

MySQL은 <http://www.mysql.com>에서 다운로드 받을 수 있음

다른 데이터베이스 서버에서는 보기도문 배열된 옵션과 유용한 특성을 가진 DBMS를 제공함

MySQL을 사용하여 명확한 어플리케이션을 개발하고 싶다면 MySQL's licensing section(http://www.mysql.com/Manual_chapter/manual_Licensing_and_Support.html)을 참고 요망

MySQL 능력

- 동시 사용자의 수가 무제한적인 처리 능력
- 50,000,000+ record를 처리할 수 있는 용량
- 매우 빠른 명령 수행 능력
- T.c.X 는 일반 대중에게 공개용 데이터베이스인 MySQL을 무료로 제공함

telnet 접속하기

MySQL은 일반적으로 telnet을 통해 접속

(참고로 Easyterm은 <http://www.arachnoid.com>에서 구할 수 있음)

ISP에서 제공된 username(ID)과 password가 필요함

MySQL 연결하기

1. CMD에 명령어 입력

```
mysql -u root -p mysql
Syntax : mysql -h hostname -u username -p password database
실행파일 | 호스트 | db유저 | 패스워드 | 사용할 데이터베이스이름
```

2. 패스워드 입력

3. 데이터베이스 선택

```
mysql> use '사용할 데이터베이스'
```

데이터베이스와 테이블

데이터베이스는 좀 더 복잡한 데이터 구조의 계층에 지나지 않음

MySQL에서는 정보의 묶음(또는 레코드)을 테이블이라 함

레코드들은 데이터 타입이라고 알려진, 사용자에 의해 조작될 수 있는 아주 작은 크기의 개체들로 구성됨
데이터타입들은 하나 또는 그 이상이 모여 레코드를 구성하며, 데이터베이스의 부분을 이루는 테이블은 레코드의 집합으로 구성됨

-----Database < Table < Record < Datatype-----

데이터 타입은 사용자가 어떠한 프로젝트를 수행하기 위해 가장 적합한 테이블을 만들 때 여러 가지의 형식과 크기로 나타낼 수 있음

MySQL 데이터 타입

▼ CHAR

CHAR 데이터 타입은 고정된 길이의 문자열을 나타내는데 사용됨

하나의 CHAR 문자열은 1-255 자 범위의 문자를 저장할 수 있음

```
car_model CHAR(10);
```

▼ VARCHAR

VARCHAR 데이터 타입은 가변적인 길이의 문자열을 저장하므로 CHAR 보다는 좀더 융통성 있는 데이터 타입이다.

VARCHAR 문자열은 1-255 자 범위의 문자를 저장할 수 있다.(CHAR 는 포함된 데이터의 크기에는 관계없이 이미 지정된 가변적인 전체의 길이를 저장하는 반면에 VARCHAR 는 오직 들어가는 데이터의 양만을 저장하므로 데이터베이스 파일의 크기를 줄일 수 있다.)

```
car_model VARCHAR(10);
```

▼ INT

INT 데이터타입은 -2147483648에서 2147483647 사이의 정수를 저장함

"unsigned" 옵션과 함께 0부터 4294967295 범위의 정수를 나타낼 수 있음

```
light_years INT;
Valid integer: '-24567'. Invalid integer: '3000000000'.

light_years INT unsigned;
Valid integer: '3000000000'. Invalid integer: '-24567'.
```

▼ FLOAT

FLOAT는 다소 정확한 숫자의 표시가 필요할 때 사용됨
작은 양의 소수점 숫자를 나타냄

```
rainfall FLOAT (4,2);
```

▼ DATE

날짜와 관련된 정보 저장
디플트 형식은 'YYYY-MM-DD'이며, '0000-00-00'에서 '9999-12-31'까지의 범위를 가짐
MySQL은 이 기사에서는 다루어지기에 날짜형식의 조절과 조작 명령어를 제공함
MySQL 매뉴얼에 이 기능에 대해서 자세하게 다루어져 있으므로 참고하기 바란다.

```
the_date DATE;
```

▼ TEXT

text 와 blob 데이터 타입은 255 - 65535 자의 문자열을 저장할 때 사용
이것은 여러분이 읽은 기사와 같은 것을 저장하기에 유용
그러나 VARCHAR 와 CHAR처럼 딱 잘라 비교할 수는 없음
단지 BLOB 와 TEXT 사이에 차이점이 있다면 BLOB은 변하기 쉬운경우에 비유할 수 있고, TEXT는
영향을 받지 않는 무감각한 경우에 비유할 수 있음

▼ SET

지정된 값으로부터 어떤 주어진 값을 선택하는, 정해진 문자열의 데이터 타입으로 그것은하나의 값이
될 수도 있고 여러개의 값을 가질 수도 있음
64개의 값까지 지정할 수 있음

```
transport SET ("truck", "wagon") NOT NULL;

위의 예로부터 수송기관(transport)은 다음의 값을 가질 수 있음
-----
"""
"truck"
>wagon"
"truck", "wagon"
-----
```

▼ ENUM

SET 데이터 타입과 비슷한 특징을 갖는 정해진 문자열의 데이터 타입이지만 선택할 수 있는값이 하나로만 정해져 있다는 점이 다름
한 공간의 바이트만을 가지므로 테이블내의시간과 공간을 절약할 수 있음

```
transport ENUM ("truck", "wagon") NOT NULL;
```

레코드

데이터들은 관련된 것들끼리 묶여서 디스크에 저장되는데 이러한 데이터의 묶음을 레코드라고 함
레코드는 데이터저장의 기본단위이며, 하나의 레코드는 서로 다른 형태의 값(데이터타입)을 가질 수 있음
아래에 나열된 이름, 주소, 전화번호는 각각 세개의 부분으로 구성된 레코드로 저장할 수 있으며 이러한 레코드는 데이터가 저장되는 기본단위로서 하나의 테이블을 구성함

이름		주소		전화번호
이순신		한국		007-0000
홍길동		북한		008-0909

테이블

우리는 데이터베이스 상에서 어떠한 명령어들을 수행하기 전에, 먼저 데이터가 저장될 수 있는 테이블을 만들어야 함

```
-----  
mysql> CREATE TABLE test (  
> name VARCHAR (15),  
> email VARCHAR (25),  
> phone_number INT,  
> ID INT NOT NULL AUTO_INCREMENT,  
> PRIMARY KEY (ID));  
-----  
테이블이 성공적으로 만들어지면 다음과 같은 화면이 출력  
  
-----  
Query OK, 0 rows affected (0.10 sec)  
mysql>  
-----  
  
여러분의 데이터베이스에 첫 번째 테이블이 생성  
주의(1) : 서로 다른 두 개의 테이블은 같은 이름을 가질 수 없음  
주의(2) : 각각의 데이터공간은 종종 컬럼(column)이라는 개체의 속성으로 나타냄
```

- 칼럼특징
 - 하나의 이름이 숫자로만 구성될 수 없음
 - 하나의 이름은 숫자로 시작할 수 있음
 - 하나의 이름은 64자까지 가능

테이블 옵션

▼ 기본키(Primary Key)

다른 속성들로부터 하나의 레코드를 유일하게 식별하는데 사용되어지며, 두 개의 레코드는 똑같은 기본 키를 가질 수 없음

이것은 두개의 레코드가 중복되어지는 실수를 없애려고 할 때 아주 적합하게 사용

▼ Auto_Increment

이 함수(function)를 가진 컬럼(column)은 레코드에 삽입될 때 자동적으로 한 개씩 값이 (previous + 1) 증가

`data`가 '`NULL`' 값 일 때도 컬럼(column)에 자동적으로 한 개씩값이 증가하여 삽입

▼ NOT NULL

`NULL`값(값이 없는)을 절대로 허용하지 않는 컬럼(column)을 의미

테이블과 관련된 명령어

- `Show Tables` : 데이터베이스에 현재 존재하고 있는 모든 테이블들을 나열
- `Show Columns` : 지정된 테이블과 관련된 컬럼과 컬럼의 정보를 보여줌

데이터베이스 다루기

1. test 테이블 만들기

`CREATE`를 사용

```
-----  
mysql> CREATE TABLE test (  
> name VARCHAR (15),  
> email VARCHAR (25),  
> phone_number INT,  
> ID INT NOT NULL AUTO_INCREMENT,  
> PRIMARY KEY (ID));  
-----
```

2. 테이블 입력하기

`INSERT`를 사용

```
-----  
mysql> INSERT INTO test VALUES  
mysql> ('Bugs Bunny', 'carrots@devshed.com',  
mysql> 5554321, NULL);  
-----  
성공적으로 입력이 되면 다음과 같은 메시지가 나타남  
-----  
Result, assuming the command was correctly entered:  
  
Query OK, 1 row affected (0.02 sec)  
mysql>  
-----
```

- `VARCHAR` 데이터타입은 단일인용문으로 둘러싸여 있음
- 모든 문자열 타입의 데이터타입은(`char`, `varchar`, `text`, `blob` 등)단일 인용문으로 둘러싸여야 하며 그렇지 않으면 에러가 발생
- `phone number`와 같은 `INT` 타입의 데이터타입은 단일인용문이 필요하지 않음
- `NULL`은 아직정해지지않은 값으로 값이 없을 수도 있지만 `NOT NULL`은 반드시 값을 입력
- `AUTO_INCREMENT`와 사용된 레코드는 값이 `NULL`이면 1을 할당하고, 값이 있으면 이전값에 +1씩 자동적으로 부여
- 한 가지 주의할 점은 데이터를 입력할때 반드시 레코드의 갯수만큼의 데이터를 입력

3. 테이블정보 검색

`SELECT`를 사용

```

-----
mysql> SELECT * FROM test
mysql> WHERE (name = "Bugs Bunny");
-----
결과:
+-----+-----+-----+
| name | email | phone_number | ID |
+-----+-----+-----+
| Bugs Bunny | carrot@devshed.com | 5554321 | 1 |
+-----+-----+-----+

```

같은 이름이지만 서로 다른 email와 phone number를 가진 데이터를 입력

```

+-----+-----+-----+
| name | email | phone_number | ID |
+-----+-----+-----+
| Bugs Bunny | carrot@devshed.com | 5554321 | 1 |
| Bugs Bunny | pepper@devshed.com | 5554331 | 2 |
| Bugs Bunny | celery@devshed.com | 5554351 | 4 |
| Bugs Bunny | lettuce@devshed.com | 5554341 | 3 |
+-----+-----+-----+

```

4. 삭제하기

DELETE를 사용

```

-----
mysql> DELETE FROM test
mysql> WHERE (name = "Bugs Bunny");
-----
결과 : 이름이 Bugs Bunny인 레코드를 모두 삭제하므로 테이블에는 데이터가 없음

-----
mysql> DELETE FROM test
mysql> WHERE (phone_number = 5554321);
-----
결과 : phone_number 가 5554321인 레코드를 삭제
+-----+-----+-----+
| name | email | phone_number | ID |
+-----+-----+-----+
| Bugs Bunny | pepper@devshed.com | 5554331 | 2 |
| Bugs Bunny | celery@devshed.com | 5554351 | 4 |
| Bugs Bunny | lettuce@devshed.com | 5554341 | 3 |
+-----+-----+-----+

```

5. 변경하기

UPDATE를 사용

```

-----
mysql> UPDATE test SET name = 'Daffy Duck'
mysql> WHERE name = "Bugs Bunny";
-----
+-----+-----+-----+
| name | email | phone_number | ID |
+-----+-----+-----+
| Daffy Duck | pepper@devshed.com | 5554331 | 2 |
| Daffy Duck | celery@devshed.com | 5554351 | 4 |
| Daffy Duck | lettuce@devshed.com | 5554341 | 3 |
+-----+-----+-----+

```

6. 조건에 맞는 특정한 열만 검색

```
-----  
mysql> SELECT name FROM test WHERE  
mysql> (name = "Bugs Bunny");  
-----  
결과 :  
+-----+  
| name |  
+-----+  
| Bugs Bunny |  
+-----+
```

테이블 변경하기

MySQL의 가장 중요한 기능중의 하나가 이미 만들어진 테이블을 변경하는 것

ALTER 문을 사용하여 다음과 같은 것을 수행

- 새로운 열 추가하기
- 열의 속성 바꾸기
- 열 삭제하기
- 테이블 이름 바꾸기

```
Example: 테이블 이름 변경하기(Rename the table)  
-----  
mysql> ALTER table test RENAME mytest;  
-----  
  
Example: 새로운 열 추가하기(Add a column)  
-----  
mysql> ALTER table mytest ADD birthday DATE;  
-----  
  
Example: 열의 속성변경하기(Modify a column)  
-----  
mysql> ALTER table mytest CHANGE  
mysql> name newname VARCHAR (25);  
-----  
  
Example: 열 삭제하기(Delete a column)  
-----  
mysql> ALTER table mytest DROP newname;  
-----  
  
MySQL 에서 ALTER 사용시 컬럼 위치 지정하기  
-----  
  
MySQL 에서 ALTER 명령을 이용해 테이블의 스키마를 변경할 때  
컬럼 위치를 지정 할 수 있음  
아래 예는 email 이라는 컬럼 뒤에 attachment 라는 컬럼을 추가하는 SQL  
> alter table {TABLE_NAME} add attachment char(64) DEFAULT '' NOT NULL AFTER email;
```

3.3.2_Database_Workbench

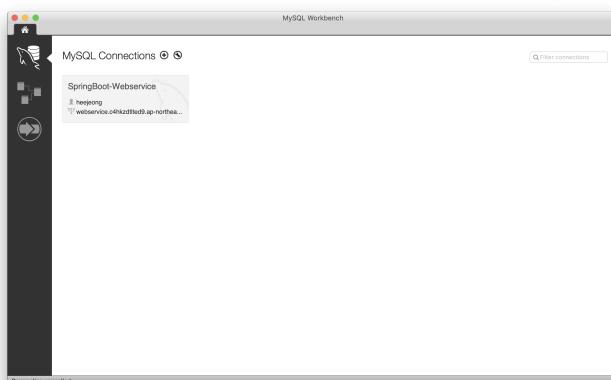
Created Time	@Sep 18, 2019 4:45 PM
Document Version	Ver 1.0.0
Last Edited By	👤 현욱 강
Last Edited Time	@Sep 22, 2019 4:37 PM
Tags	Database MySQL Workbench
담당자	👤 현욱 강

Database

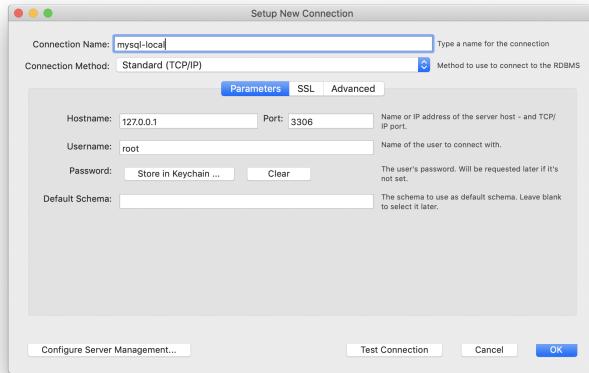
Workbench

1. Workbench 방법

- MySQL Workbench에서 새로운 연결 생성
MySQL Workbench 실행 > + 클릭



- 원하는 Connection Name('mysql-local')을 입력



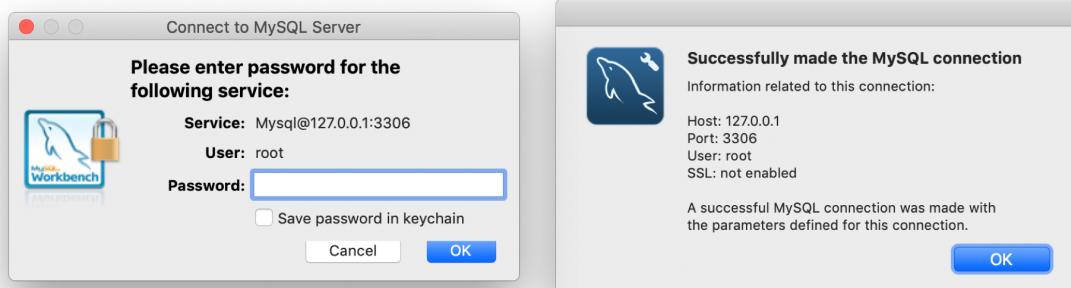
- 로컬에서 실행 중인 MySQL Server를 사용하는 경우

hostname: 127.0.0.1 또는 localhost

mysql 기본 포트: 3306

3. 아래의 Test Connection 클릭

MySQL을 설치할 때 지정한 root password를 입력

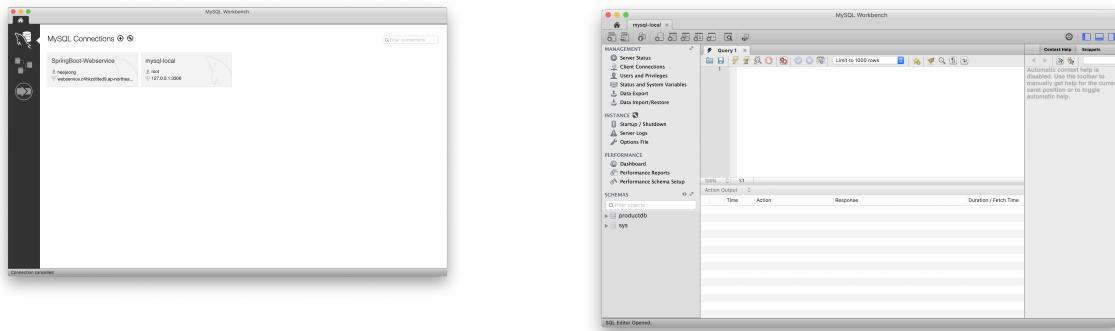


만약 connection을 실패할 경우, mysql server가 켜져 있는지 확인하거나 password가 맞는지 확인

mysql server 실행 및 root password 재설정은 MySQL 깨알 팁 모음을 참고하자!

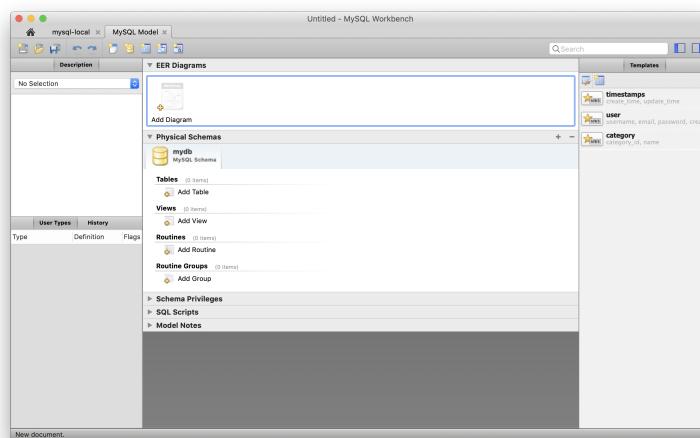
2. 실행 중인 MySQL과 연결

MySQL Connections > 위에서 지정한 Connection Name('mysql-local') 클릭 > root password 입력



3. File > New Model (MacOS: Cmd + N)

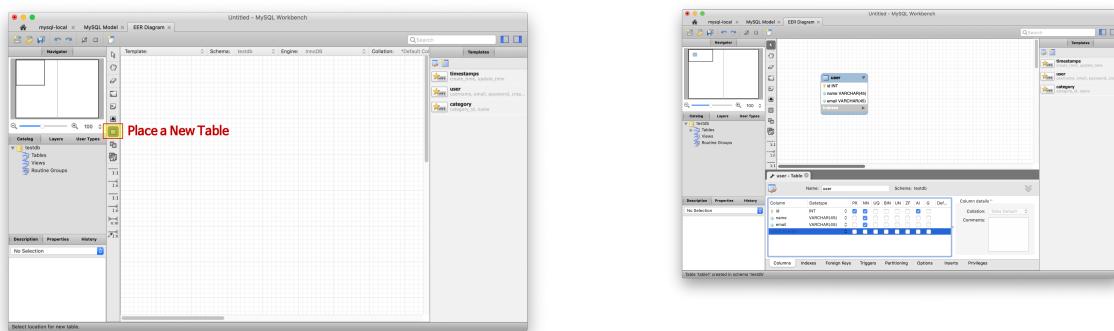
mydb 더블 클릭 - 원하는 Schema Name(' testdb')을 입력



4. 자신이 설계한 Table을 DB에 생성

[방법1] Add Diagram 이용

Add Diagram 클릭 > Place a New table 클릭



Ex)

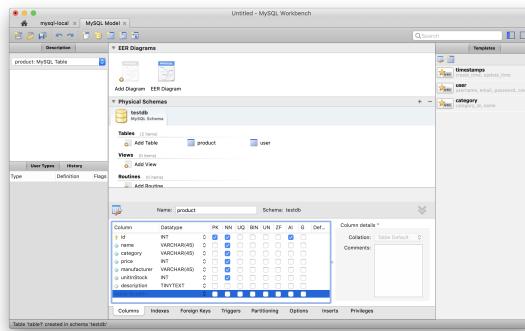
table name을 “user”로 설정

AI: auto incremental, 자동으로 1씩 증가하도록 하는 것

id, 이름, 이메일 등록

[방법2] Add Table 이용

Add Table 클릭



Ex)

table name을 “product”(Ex. 쇼핑몰의 상품)로 설정

id, 상품명, 카테고리(분류), 가격, 제조사, 잔고, 설명을 등록

참고 자신이 설계한 Model 저장

File > save model as > 원하는 Name(해당 포스트에서는 ‘testdb’)을 입력

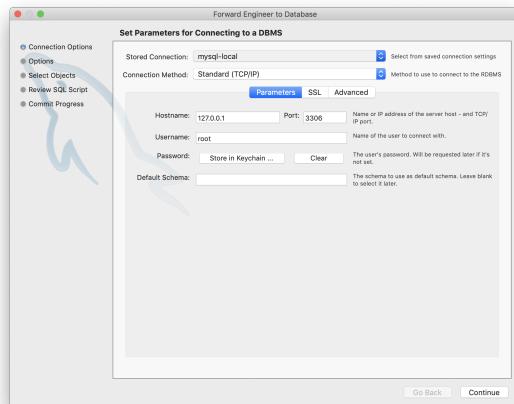
자신이 설계한 내용을 불러옴

5. 자신의 설계한 Model을 실제 DB에 반영

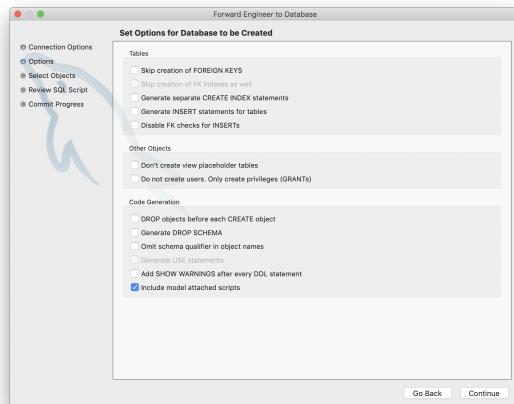
Database > Forward Engineer 클릭

Forward Engineer: 자신의 설계한 Model을 실제 DB에 반영하는 것

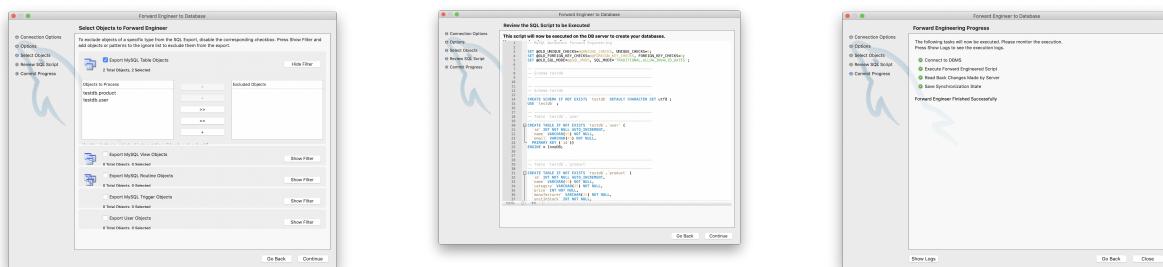
Reverse Engineer: DB의 내용을 가지고 Model을 만드는 것



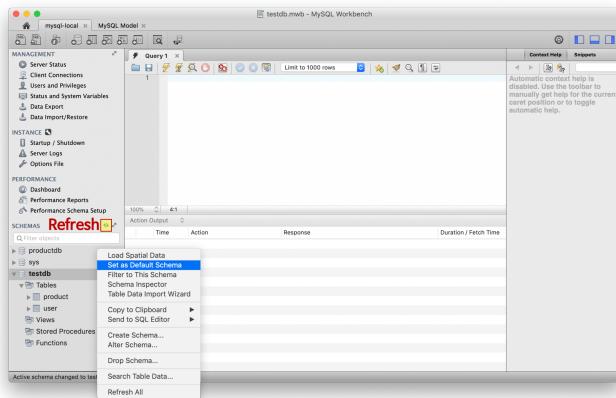
Stored Connection: 자신이 설정한 Connection Name('mysql-local') 선택



MySQL을 설치할 때 지정한 root password를 입력



6. 실제 DB에 반영되었는지 조회



1. MySQL Model은 닫고 Local instance로 돌아옴

2. refresh

3. 해당 Schema를 기본으로 설정

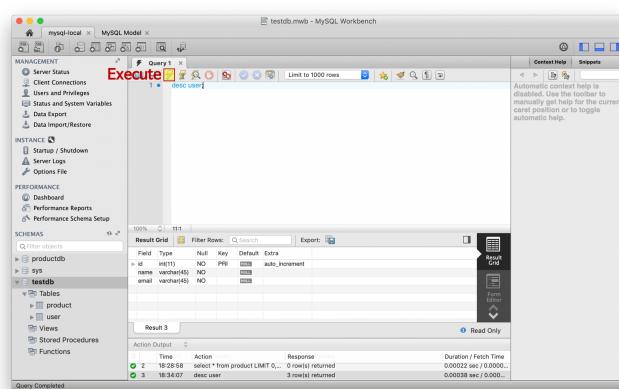
eStore 우클릭 - Set as Default Schema

해당 Schema가 진하게 표시된 것이 확인됨

7. 기본 SQL 문법

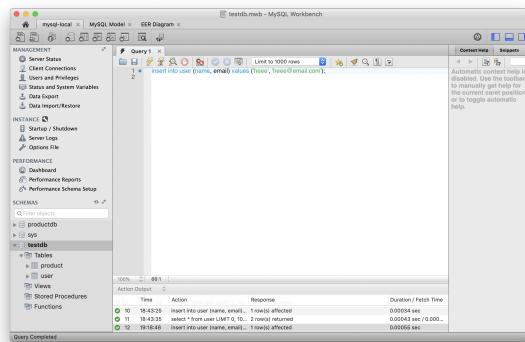
1. 'desc table이름'을 통해 설계한 columns의 내용 확인

Ex) desc user; 입력 후 실행 (번개모양)



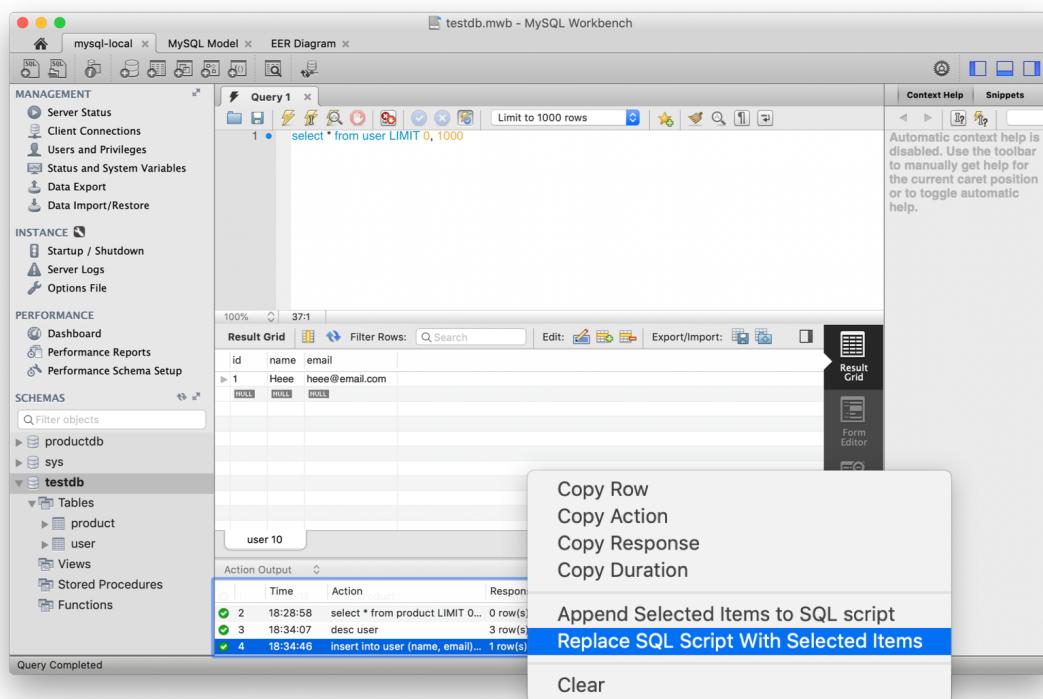
2. 'insert into table이름 (column1, column2, column3) values (value1, value2, value3);'으로 table에 data 삽입

Ex) `insert into user (name, email) values ('heee', 'heeee@email.com');`

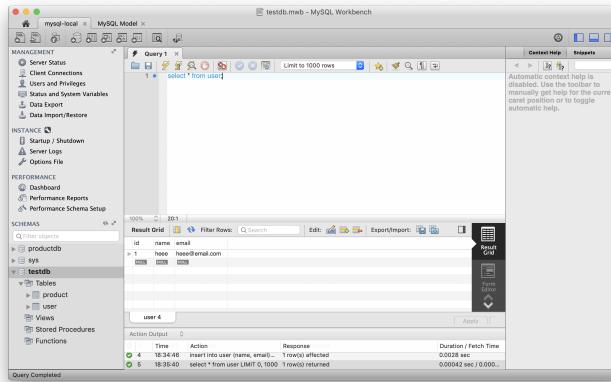


- 참고 명령어 반복

- 아래의 Action Output의 명령어 우클릭 > ‘Replace SQL Script With Selected items’ 클릭



3. ‘`select * from table이름`’으로 table에 들어간 columns 확인
Ex) `select * from user;` 입력 후 columns 확인



8. Result Grid에서 추가/수정한 내용을 DB에 반영하기 위해서는 우측 하단에 Apply를 선택

heee -> Heee 변경 후 Apply

9. DB 백업

- 자신이 설계한 Model 저장하기
- Table 안의 Data 백업하기
- DB 전체 백업하기
- DB 내의 모든 Table과 그 안의 Data

3.3.3_Database_phpMyAdmin

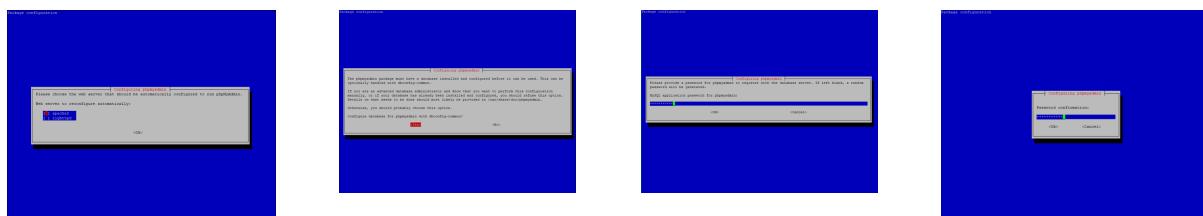
Created Time	@Sep 19, 2019 12:57 AM
Document Version	Ver 1.0.0
Last Edited By	这家伙 강
Last Edited Time	@Sep 22, 2019 4:33 PM
Tags	Database phpMyAdmin
담당자	这家伙 강

Database

phpMyadmin

라즈베리파이에 phpMyAdmin 설치

```
$ sudo apt-get install phpmyadmin
```



차례대로 Yes, MariaDB 비밀번호, phpMyAdmin에 쓸 비밀번호 입력

phpmyadmin 설정 셋업

설치가 완료되면 apache2.conf 파일 맨 마지막에 다음의 문장을 추가

```
# /etc/apache2/apache2.conf 파일  
Include /etc/phpmyadmin/apache.conf
```

심볼릭 링크를 활용하여 주소창에서 접속할 수 있게 설정

```
$ sudo ln -s /usr/share/phpmyadmin/ /var/www/html
```

주소창에 입력

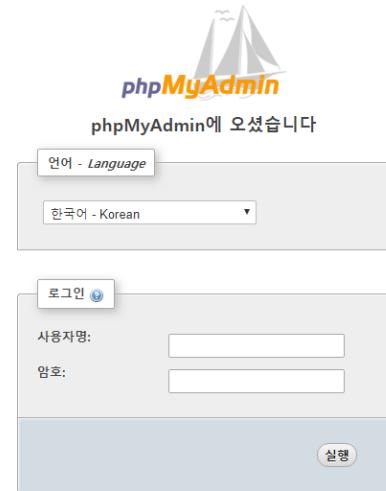
→ 라즈베리파이 IP주소/phpmyadmin

만약 심볼릭 링크로 하지 않는다면

```
sudo service apache2 restart
```

주소는

http://127.0.0.1/phpmyadmin
으로 접속

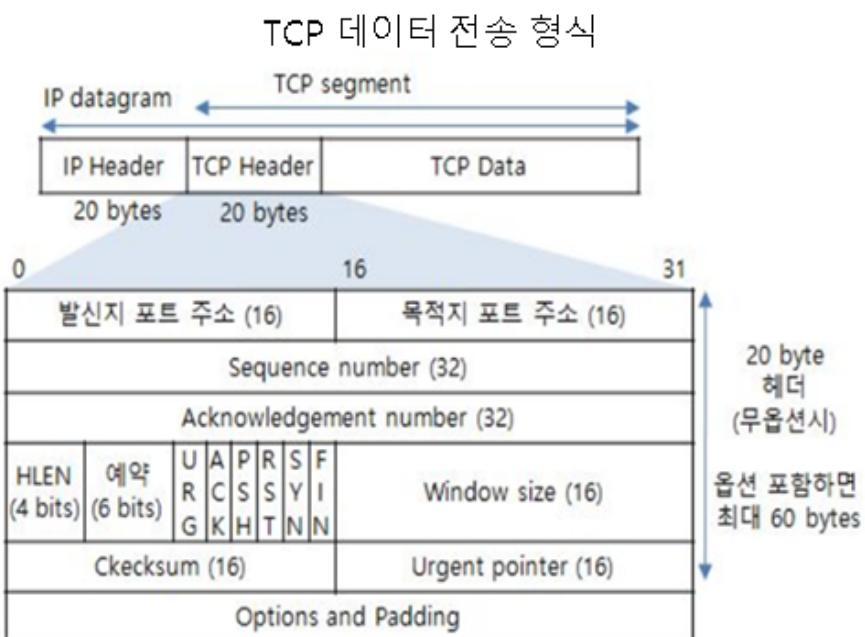


3.3.4 Database_TCP_데이터_전송형식

Created Time	@Sep 18, 2019 9:17 PM
Document Version	Ver 1.0.0
Last Edited By	👤 현욱 강
Last Edited Time	@Sep 22, 2019 4:31 PM
Tags	Database TCP
담당자	👤 현욱 강

Database

TCP 데이터 전송 형식



- 목적지 포트주소는 종류에 따라 처음 세그먼트를 전송하는 측에서 사용하는 목적지 포트가 정해져 있음
 - Sequence Number(순서번호 32비트)는 TCP 세그먼트의 순서번호를 표시
 - TCP 각 세그먼트의 첫번째 바이트에 부여되는 번호는 32 비트이므로 최대 4기가 바이트 크기를 송신
 - acknowledgement Number는 상대방이 보낸 세그먼트를 잘 받았다는 것을 알려주기 위한 번호
 - 헤더 길이 필드는 4 바이트 단위로 표시
 - TCP 헤더 길이는 최소 20바이트 총 60바이트 이하의 Flag bits (URG, ACK, PSH, RST, SYN, FIN)는 TCP 세그먼트 전달과 관련되어 TCP 회선 및 데이터 관리 제어 기능을 하는 플래그로 흐름제어, 연결설정, 연결종료, 연결리셋, 데이터 전송모드
 - 윈도우 크기는 흐름제어를 위해 사용하는 16비트 필드로 TCP 흐름제어를 위해 통신의 상대편에게 자신의 버퍼 여유용량 크기를 지속적으로 통보
 - Checksum는 헤더와 데이터의 에러를 확인
 - Urgent PointerTCP 세그먼트에 포함된 긴급 데이터의 마지막 바이트에 대한 일련번호
-

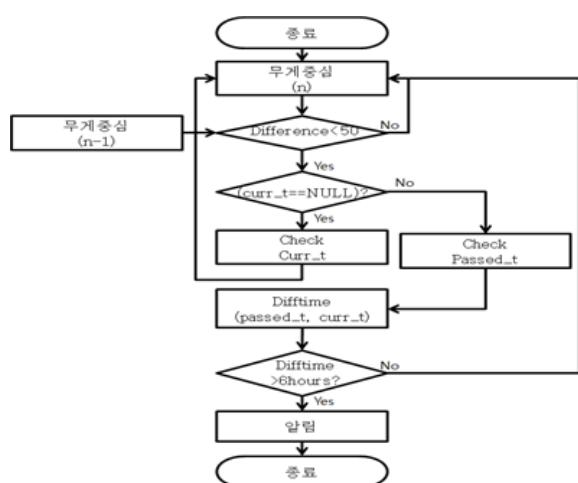
3.3.5_Database_Camera연동

Created Time	@Sep 18, 2019 3:36 PM
Document Version	Ver 1.0.0
Last Edited By	긱스 강
Last Edited Time	@Sep 22, 2019 4:10 PM
Tags	Camera Database
담당자	긱스 강

Database

Camera연동

CCTV 구현 순서도



CCTV구현 순서도로 Difference는 이전중심($n-1$)과 현재중심(n)을 비교하여 차이가 50이상 나는 무게중심으로 Difference가 적으면, 움직임의 변화가 별로 없는 것이므로, 움직임이 감지되지 않는 것으로 봄

`difftime(curr_t, passed_t)`은 움직임의 변화가 감지되지 않은 때로부터의 경과 시간으로 curr_t는 화면에 변화가 없기 시작한 시간이고 passed_t는 curr_t이 기록된 이후 계속해서 화면 변화가 발생하지 않는 시간임

MOVE 판단

```
Boolean PatientMove()
{
    double tmpX = 0.0, tmpY = 0.0;
    double copX, copY;
    double x, y;
    double root;

    copX = (((data1[0] + data1[2]) * scaleBoardLeft + (data1[1] + data1[3]) * scaleBoardRight) / massTotal);
    copY = (((data1[0] + data1[1]) * scaleBoardTop + (data1[2] + data1[3]) * scaleBoardBottom) / massTotal);

    //x축
    if (copX > 0) //양수이면
    {
        copX += scaleBoardWidth / 2.0;
    }
    else //음수이면
    {
        copX *= (-1);
    }
}
```

```

    }

    //y축
    if (copY > 0)//양수이면
    {
        copY += scaleBoardHeight / 2.0;
    }
    else //음수이면
    {
        copY *= (-1);
    }

    //시간이 흐른 후 그 점
    if (TimeChanged()) //시간 지나고
    {
        if (tmpX != copX || tmpY != copY) //위치가 같지 않다면
        {
            tmpX = copX;
            tmpY = copY;
        }

        x = tmpX - copX;
        y = tmpY - copY;
        root = Math.Sqrt((x * x) + (y * y));

        if (root > 1.5)
        {
            return false;
        }
    }
    return true;
}

```

```

Boolean PatientMove2()
{
    double dx = saveCOP.X - COP.X;
    double dy = saveCOP.Y - COP.Y;
    double distance = Math.Sqrt(dx * dx + dy * dy);

    if (distance > 0.5) return true;
    return false;
}

// 50cm 이상 움직인다면 PatientMove2가 true가 되어 db가 저장됨

```

DB LIST

```

User newUser = new User()
{
    Time = (string)dateTimeString.Content,
    DR = Math.Round(data1[3], 2),
    DL = Math.Round(data1[2], 2),
    UR = Math.Round(data1[1], 2),
    UL = Math.Round(data1[0], 2),
    TOTAL = Math.Round(massTotal, 2)    // 체중
};

```

DB INSERT

```

APM_DB db = new APM_DB("라즈베리파이주소", "DB이름");
string queruyStr;

queruyStr = string.Format("INSERT INTO sensor VALUES('{0}', '{1}', '{2}', '{3}', '{4}')",

```

```
DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss"),
Math.Round(data1[0], 2), Math.Round(data1[1], 2),
Math.Round(data1[2], 2), Math.Round(data1[3], 2));

db.query(qeruyStr);
```

DB SELECT

```
string queryStr = string.Format("select * from sensor where ul='{0}', ur='{1}',
dl='{2}', dr='{3}'", data1[0], data1[1], data1[2], data1[3]);
Console.WriteLine(data1[1]);

APM_DB db = new APM_DB("라즈베리파이주소", "DB이름");

string result = db.query(queryStr);
```