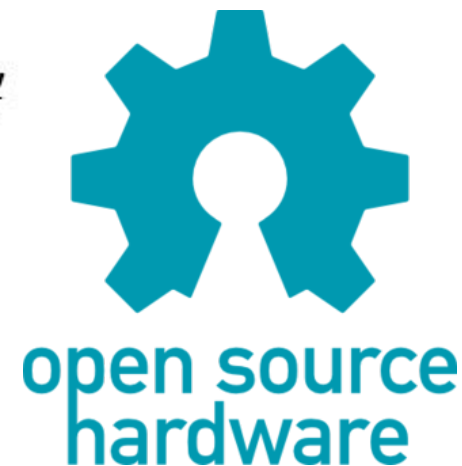


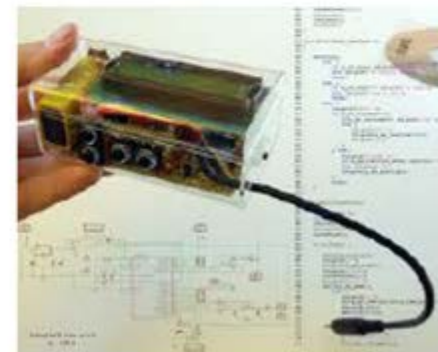


Smart Phone 개발자 관점에서 본
Arduino,
RaspberryPi,
ROS의 구조 분석
그리고
3D 프린터 활용



www.kelp.or.kr
cafe.naver.net/openrt

박 철
e2g1234@naver.com



스마트폰 개발자 입장과 FW, HW 개발자의 입장



- ✓ 스마트폰 개발자
 - 자바 위주의 언어
 - C언어에 대한 막연한 두려움 존재
 - SDK에만 익숙하고 NDK, PDK에는 막연한 두려움 존재
 - DIY시스템의 주류인 Arduino, RaspberryPi, ROS의 전반적 흐름 이해
 - 오픈 소스 기반 공동 작업 및 결과물 도출의 방법 제시

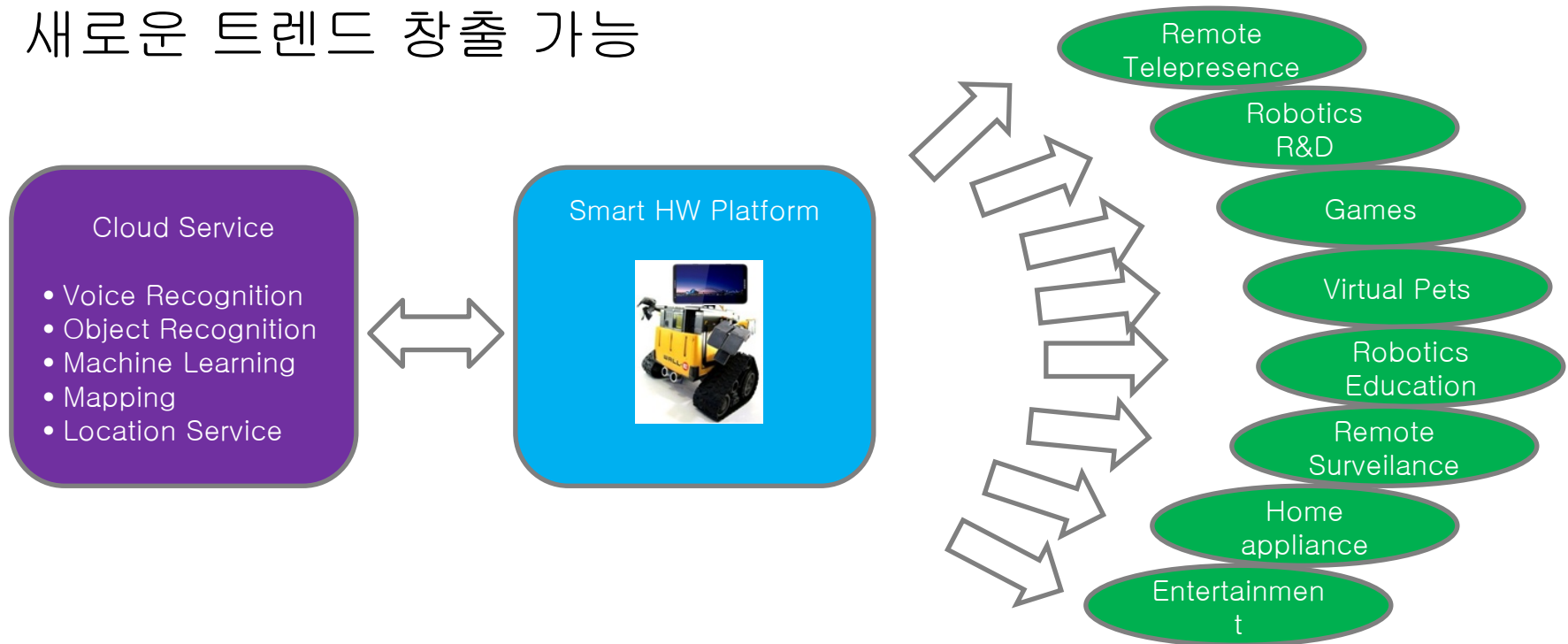
- ✓ HW, FW 개발자
 - 지금까지와 다른 흐름 이해
 - 개발자로서 자신의 포지션 재정립
 - 스마트폰 개발자와 공동작업 룰 모델 정립
 - DIY 형식으로 스스로 embedded system을 이용하여 제작

- ✓ 융합형 Embedded System은 이미 진행 중이며 결과물 도출 단계

Smart Phone 응용 HW의 활용 범위



- ✓ BT/WIFI/NFC를 통하여 HW 제어 가능.
- ✓ Smart Phone의 막강한 기능을 HW에 쉽게 적용
- ✓ 기존의 복잡한 HW 시스템을 단순화 한다.
- ✓ 임베디드 시스템에서 구현하기 어려운 UI를 쉽게 구현할 수 있다.
- ✓ 새로운 트렌드 창출 가능



[Why] 로봇·오디오·노트북... 모두 스마트폰으로 通한다

http://news.chosun.com/site/data/html_dir/2012/11/09/2012110901250.html



현재까지 구현된 스마트를 이용한 로봇은 단순한 바퀴 구동형 로봇으로 도킹 시스템 형식으로 되어 있어서 스마트폰의 기능을 로봇 플랫폼이 제한을 하고 있는 형식이다.

로봇 플랫폼의 구조를 개선해서 스마트폰의 기능을 전반적으로 활용할 수 있는 구조로 변환하여야 한다.

Smart Phone :



스마트폰 연동

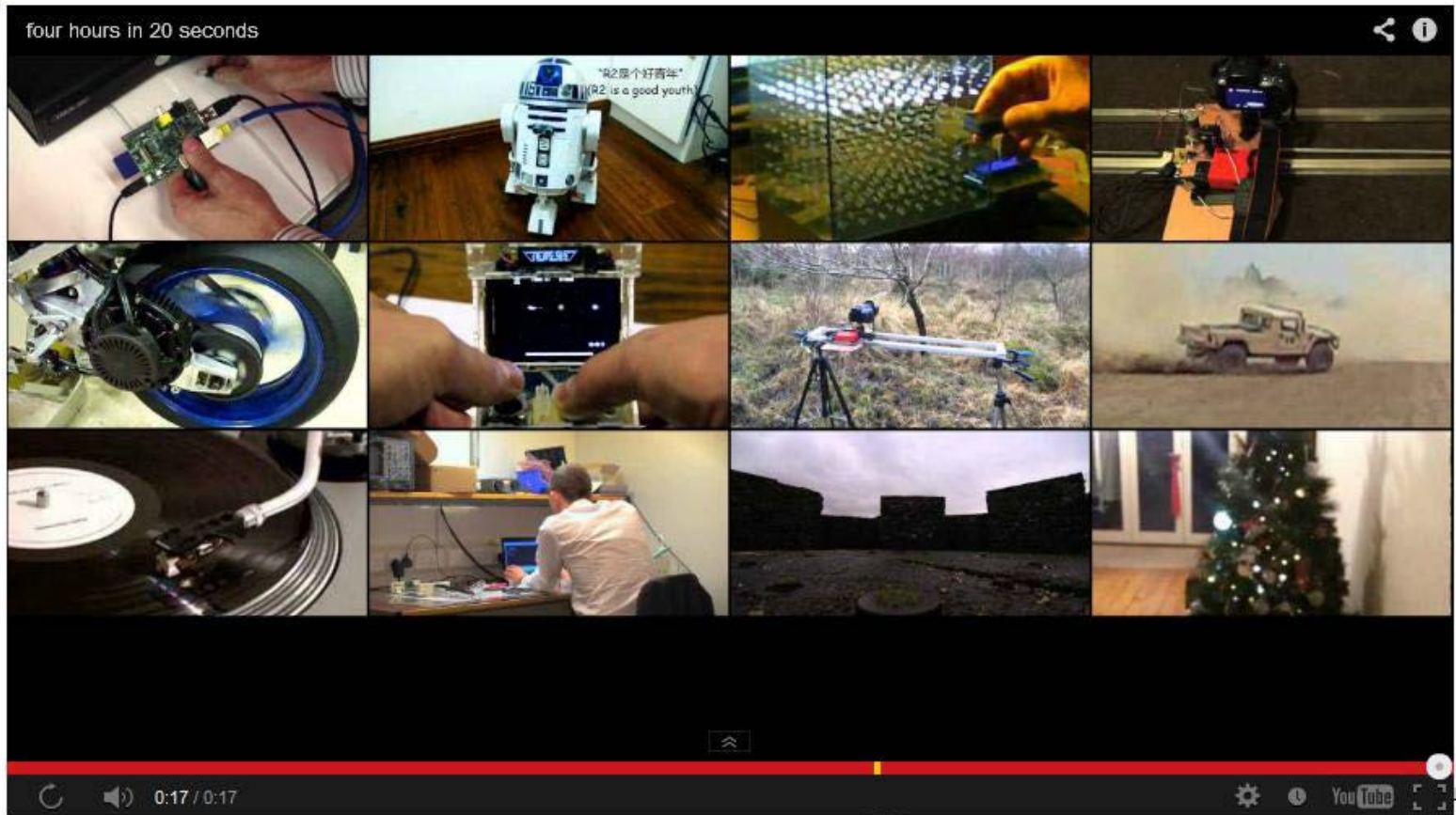
- BT,WIFI,NFC 인터페이스 기반으로 스마트폰과 연동
- 스마트폰의 막강한 기능 사용

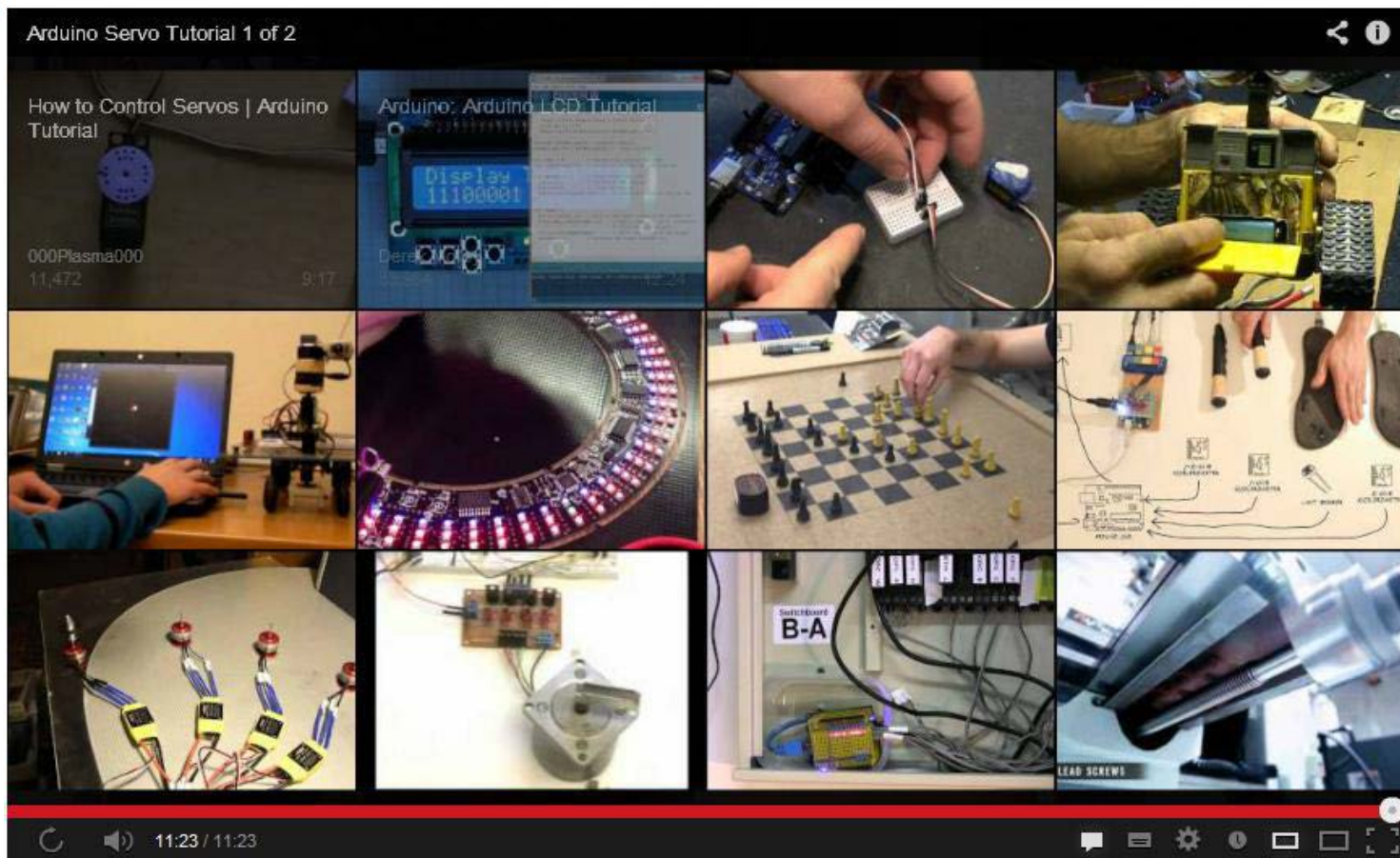
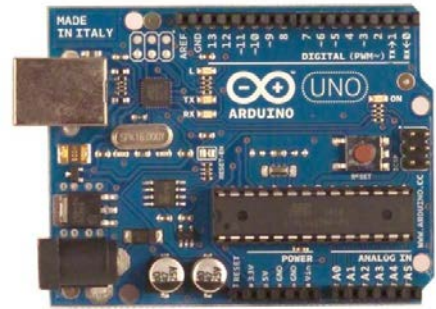


Raspberry Pi



라즈베리파이 : 임베디드 리눅스 기반 저가의 싱글 보드 PC







PR2 Beta Program

Nao Robot

awesomo2001
3,047,113

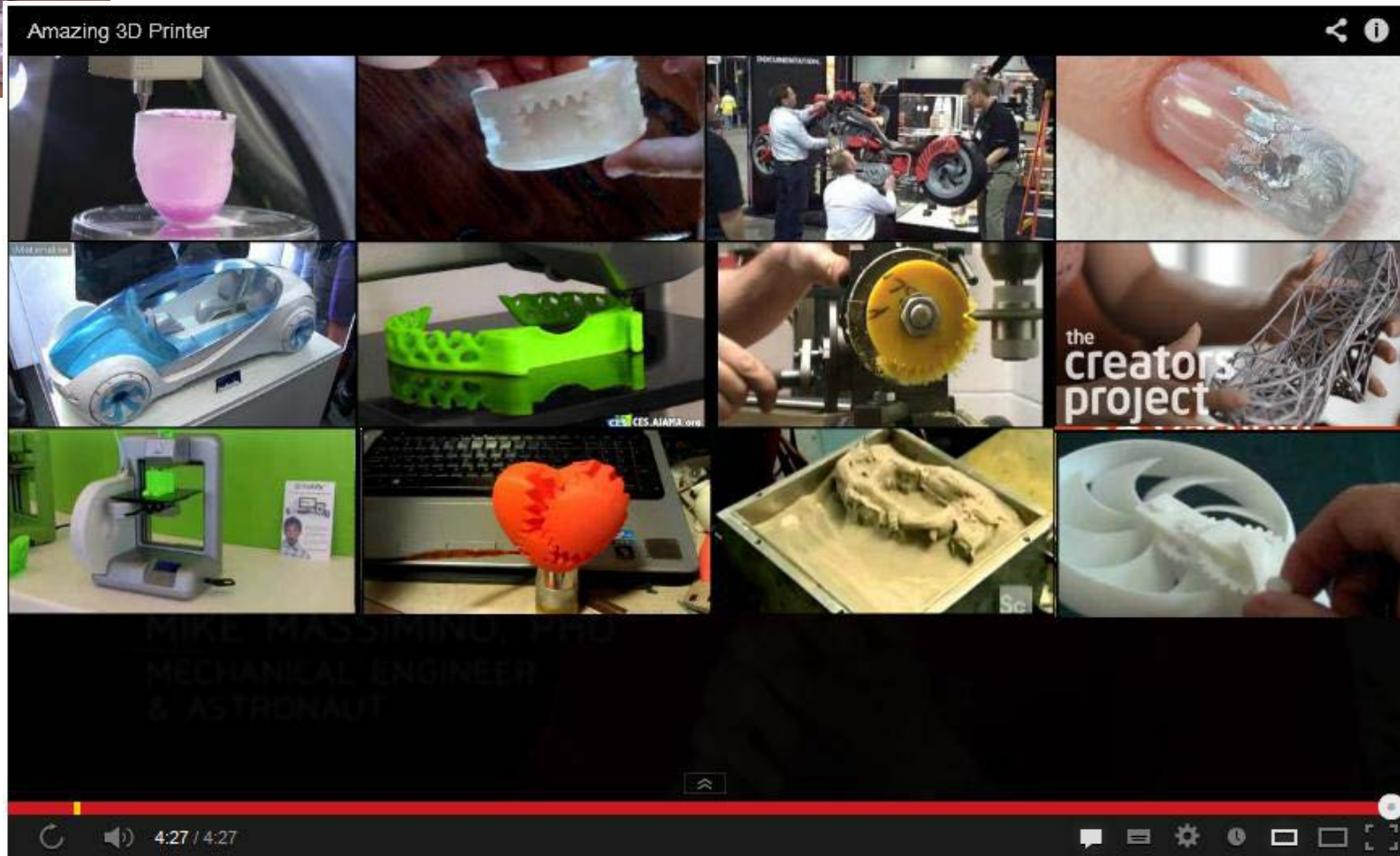
2:53

4:00 / 4:00

3D Printer



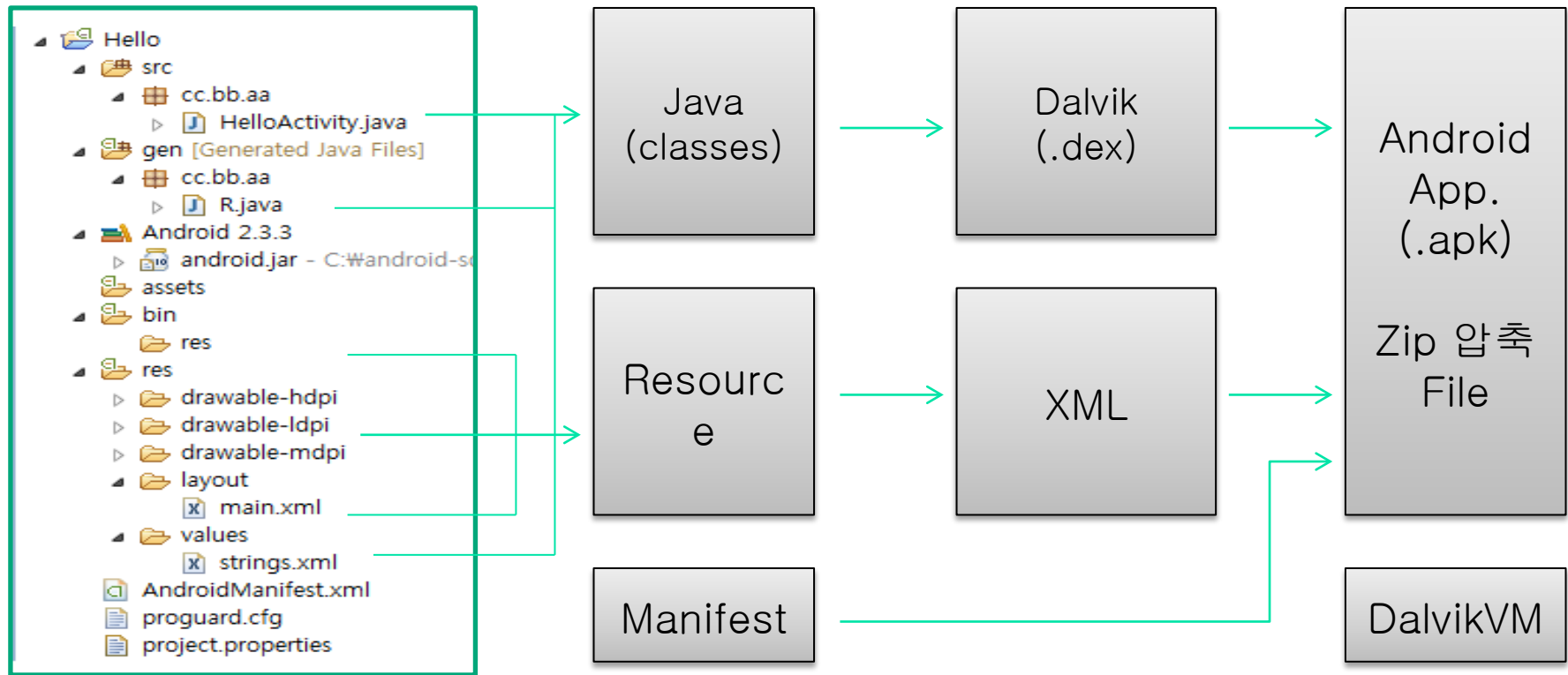
- 목적하는 결과물의 기구 파트를 직접제작
- 완성도 높은 결과물을 도출함



스마트폰에 추가해야 하는 것



기존 스마트폰기능에서 BT, WIFI, NFC 기능만 추가 되면 기존 Application 환경과 동일하다.



개발자와 일반인의 시각 차



Open Source/HW 기반의 HW,FW,스마트폰 개발자 일반인이 참여하는
Open Project를 통한 결과물 공유

소통: 서로를 이해하자

개발자	일반인
직업	취미
벌거 없네	대단하다.
아무생각 없다	재미있다
제품	DIY

응용레벨 - 아이디어

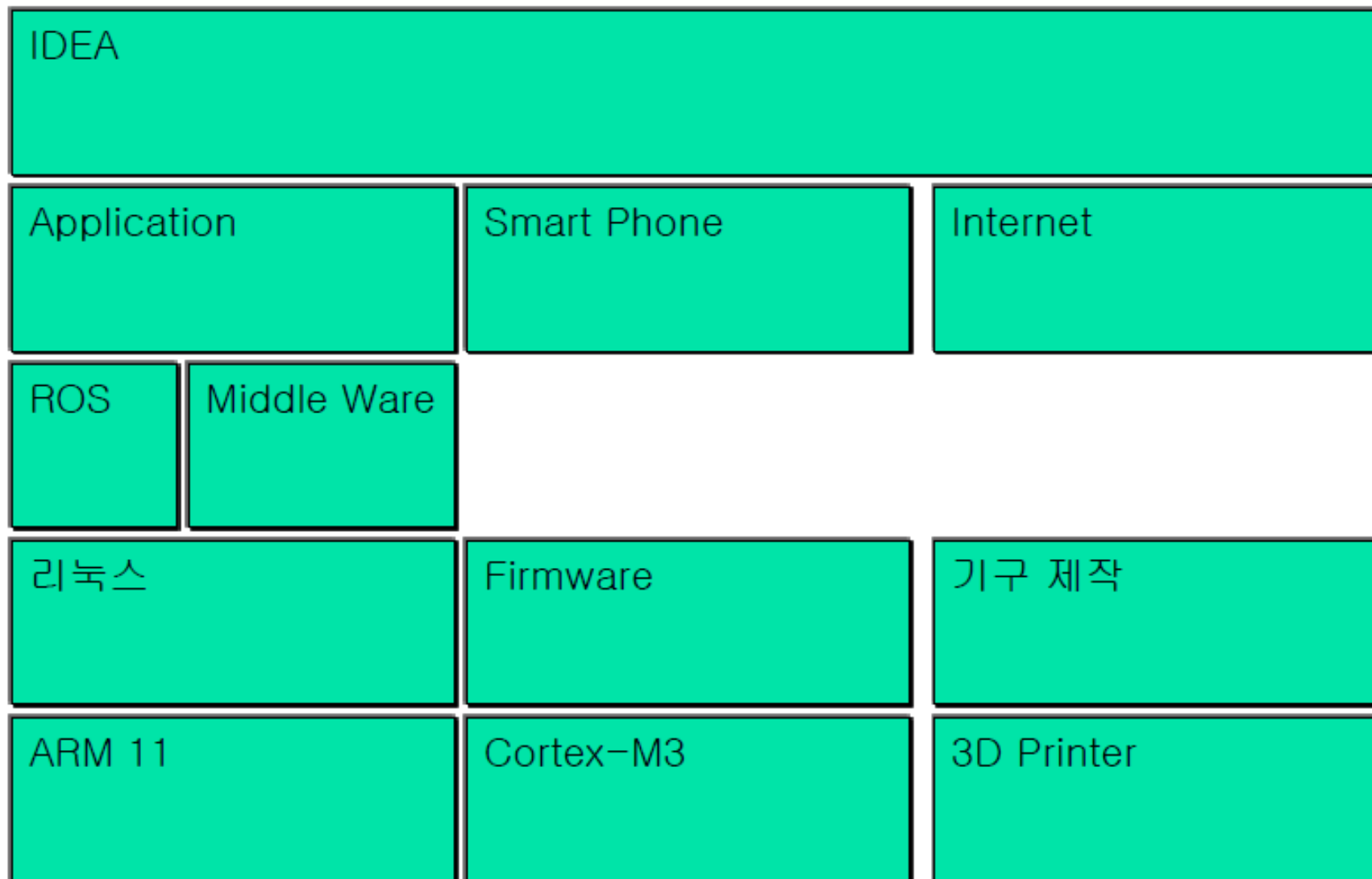
개발 레벨 - 기술

전문화 -> 일반화
Mass -> 다양화
구입 -> 제작

Open Source
Open HW
Open Project

새로운 시장 형성
단순 DIY 부터 고차원 DIY
국내 초보 수준/외국 매우 활발

전반적인 Structure



아두이노를 예로 한 추상화의 효과



아두이노

- 보드의 Pin map으로 기술됨 (Pin Map으로 추상화)
- 핀의 속성에 따른 API를 이해해야 함
- 아두이노 IDE(통합환경) 도입됨
- Pin 속성에 맞는 API를 사용

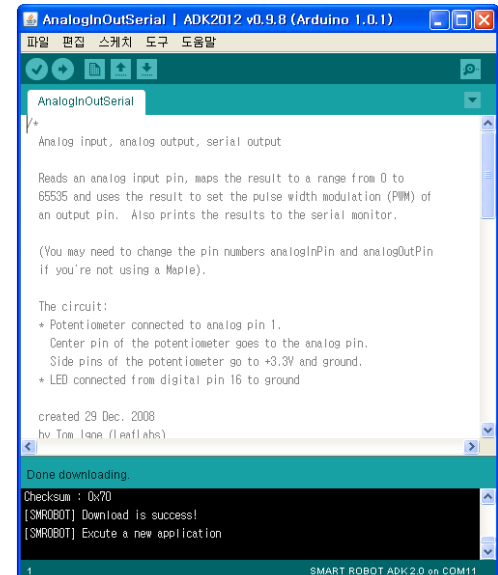
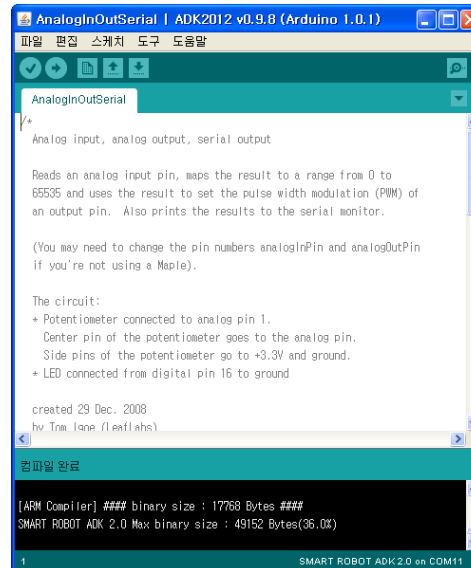
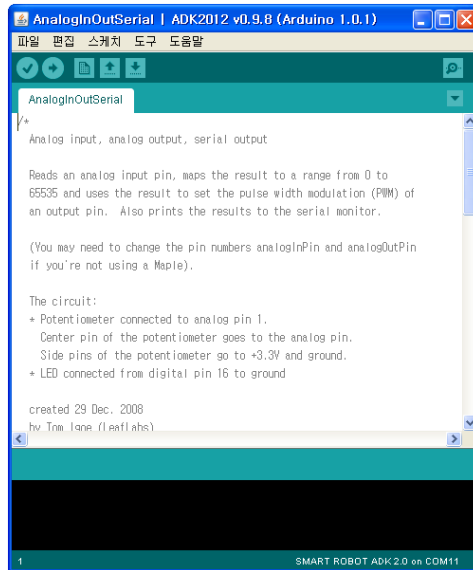
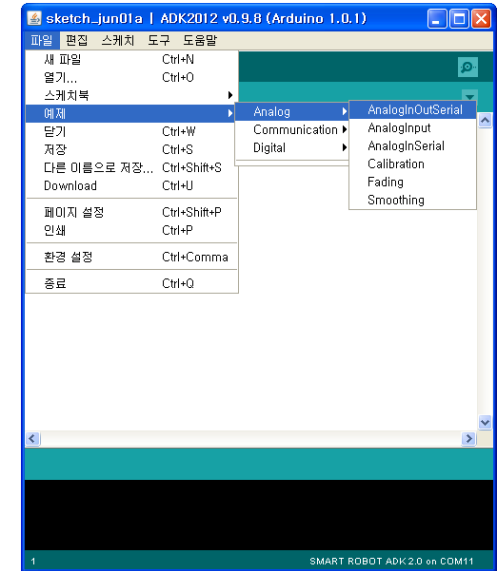
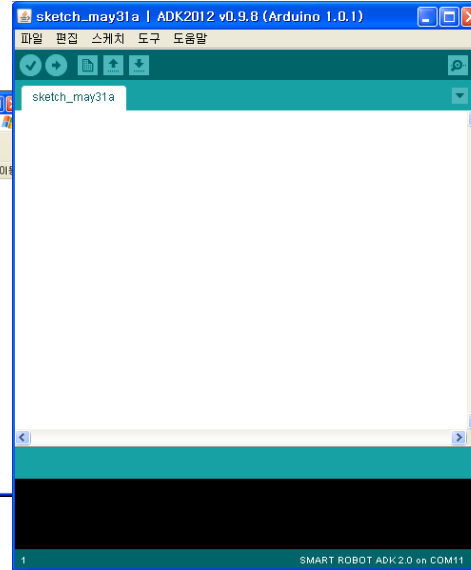
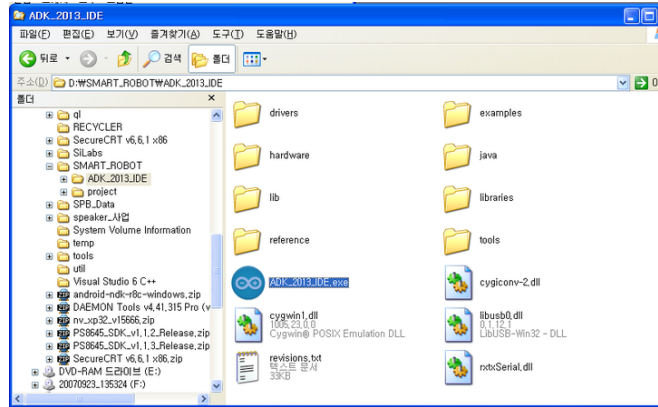
C언어

- 주변장치 Memory map으로 기술됨(Memory Map으로 추상화)
- 주변장치의 특성을 이해해야 함
- 컴파일러가 도입됨
- 컴파일러가 c언어를 Assembly언어로 변환

Assembler

- Register level에서 기술됨(심볼로 추상화)
- Micom 내부구조를 이해해야 함
- Assembler 도입됨
- Assembly 언어를 기계어로 변환

아두이노 컴파일 및 다운로드 과정



예제 및 실행 결과

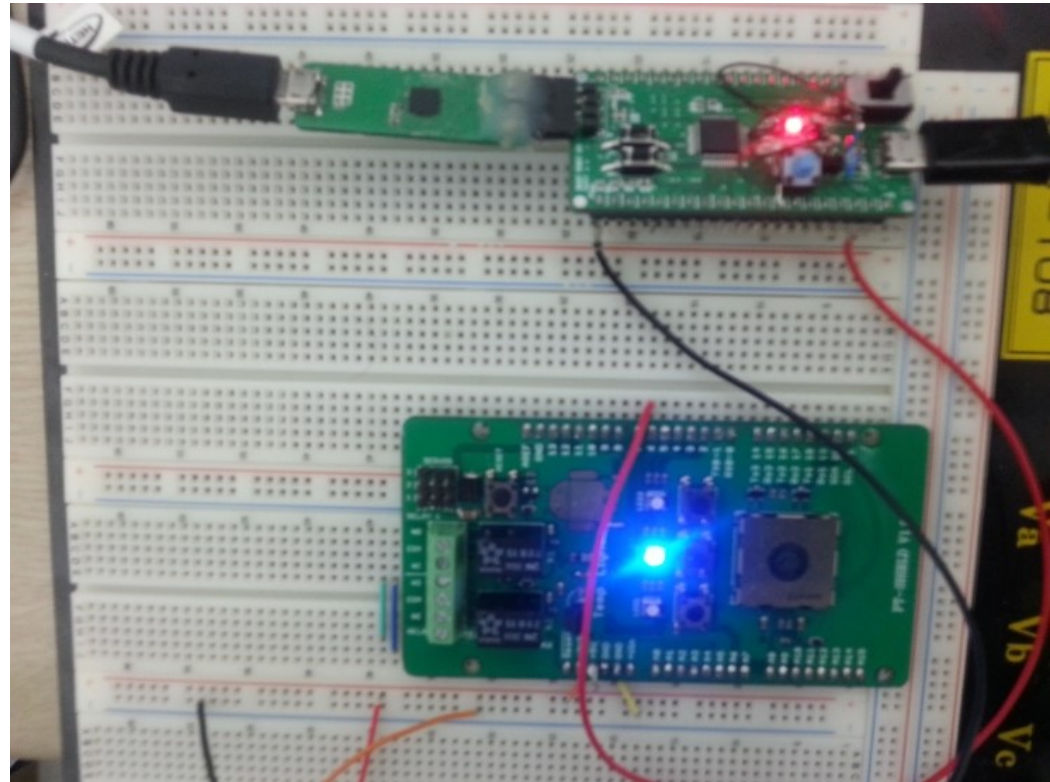


```
// These variables will change:
int sensorValue = 0;    // value read from the pot
int outputValue = 0;    // value output to the PWM

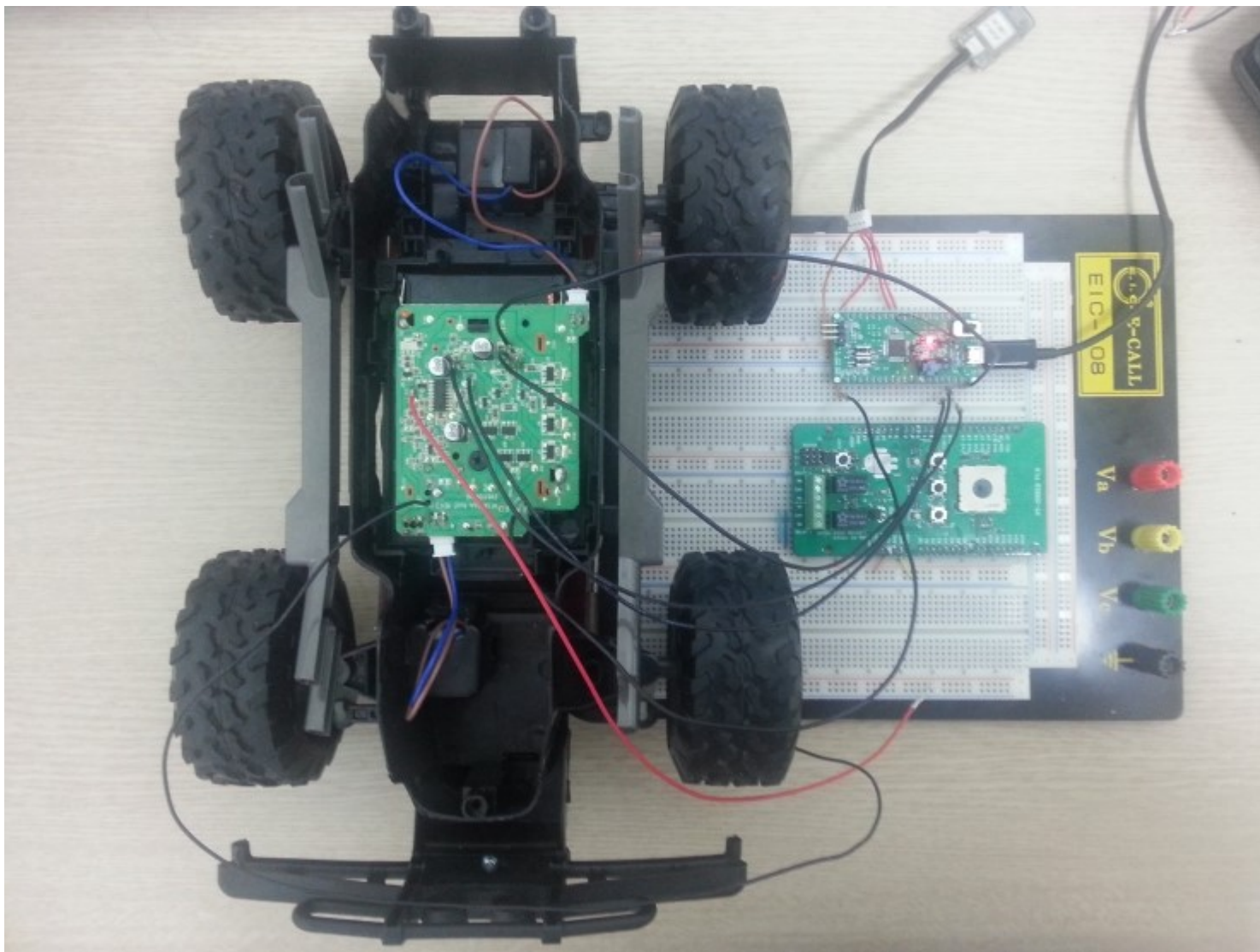
void setup() {
  // Configure the ADC pin
  pinMode(1, INPUT_ANALOG);
  // Configure LED pin
  pinMode(6, PWM);
}

void loop() {
  // read the analog in value:
  sensorValue = analogRead(analogInPin);
  // map it to the range of the analog out:
  outputValue = map(sensorValue, 0, 1023, 0, 65535);

  // change the analog out value:
  pwmWrite(pwmOutPin, outputValue);
  // print the results to the serial monitor:
  SerialUSB.print("sensor = ");
  SerialUSB.print(sensorValue);
  SerialUSB.print("\n\t output = ");
  SerialUSB.println(outputValue);
}
```



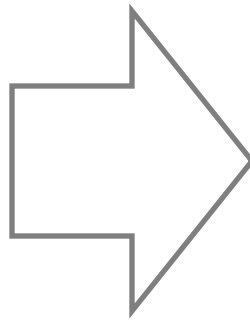
RC카 Test



RC카 제작 과정



Wall-E 개조 스마트폰 결합로봇



스마트 Wall-E project

Wall-E 및 아두이노를 선택이유



짧은 개발 기간

- 어떤 플랫폼을 적용할 것인가에 대한 고민이 있었음
- 스마트폰이 머리 처럼 사용 되며 자유가 있어야 스마트폰의 기능을 최대한 활용가능
- 2월 중순까지 기능 대비 구현 가능한 platform
- 이미 구현된 예제가 있으므로 시행착오가 적음

완성도 높은 개발 환경 및 어플리케이션

- 초보자도 쉽게 구현가능한 개발환경
- 자율 주행, 음성 액션, 얼굴 트래킹, 칼라 트래킹 등 완성도 있는 어플리케이션 구현
- 향후 구현을 고려해야 할 기능이 이미 구현되어 있음

기존 완구 플랫폼 사용

- Wall-E, RoboPhilo, RoboQuad, iRobot Roomba Hack 등 참고할 만한 플랫폼이 많음
- 완구를 변형 HW 플랫폼 이식에 대한 예제가 많음

스마트 Wall-E 구현기능



WALL-E 플랫폼 사용

- 일반 완구 Wall-E에 플랫폼 적용
- BT를 기본 개발환경으로 사용하여 향후 스마트폰으로 제어가능

자율 주행 및 댄스

- 구동부용 서버 모터, 초음파 센서, 머리 및 팔 구동용 모터를 이용하여 자율주행기능이 구현되어 있음
- 음성 액션과 연결하여 음성 명령에 따라 방향전환, 댄스, 영상이용 트래킹 기능 구동가능

음성 액션 기반 트래킹 기능

- 음성 인식을 기반으로 다양한 명령을 통하여 제어가능
- 얼굴인식, 칼라인식, 특정 패턴인식 등 영상 인식과 구동부와 연동을 통한 응용기능 구현가능
- 음성 액션 기반 얼굴 트래킹, 볼 트래킹, 특정 패턴인식 등의 어플리케이션이 구현되어있음

거북이 라즈베리파이 스마트폰



Test2 : TurtleBot2 + raspberypi + xtion + ROS

Image View

/jopeni2_camera/rgb/image_raw

/jopeni2_camera/depth/image_raw

Forward/back arrows : linear velocity tncr/decr.
Right/left arrows : angular velocity tncr/decr.
Spacebar : reset linear/angular velocities.
d : disable motors.
e : enable motors.
N : quit.

```
INFO [1367565263.285907699]: KeyOp: angular velocity incremented [0.33]
INFO [1367565267.231024933]: KeyOp: angular velocity incremented [0.66]
INFO [1367565268.558466269]: KeyOp: reset linear/angular velocities.
INFO [1367565269.334811056]: KeyOp: linear velocity incremented [0.05[0.33]
INFO [1367565270.383868469]: KeyOp: angular velocity incremented [0.05[0.33]
INFO [1367565271.839531778]: KeyOp: linear velocity incremented [0.1[0.33]
INFO [1367565273.007172706]: KeyOp: angular velocity decremented [0.1[0.33]
INFO [1367565274.886547677]: KeyOp: reset linear/angular velocities.
INFO [1367565275.698616019]: KeyOp: angular velocity decremented [0.33]
```

```
ROS_MASTER_URI=http://192.168.4.250:11311

core service [/rostop] found
process[kobuki-1]: started with pid [2714]
process[mobile_base-2]: started with pid [2715]
process[diagnostic_aggregator-3]: started with pid [2720]

NODES
 /
   diagnostic_aggregator (diagnostic_aggregator/aggregator_node)
   kobuki (nodelet/nodelet)
   mobile_base (nodelet/nodelet)

[ WARN ] [1367565249.111409233]: Using default parameters for RGB camera calibration.
[ INFO ] [1367565249.125449973]: using default calibration URL
[ INFO ] [1367565249.134244809]: camera calibration URL: file:///home/pi/.ros/camera_info/depth.yaml
[ WARN ] [1367565249.138140202]: unable to open camera calibration file [/home/pi/.ros/camera_info/depth.yaml]
[ WARN ] [1367565249.150137302]: Camera calibration file /home/pi/.ros/camera_info/depth.yaml not found.
[ WARN ] [1367565249.165314229]: Using default parameters for IR camera calibration.
```

0:19 / 0:35

YouTube

아두이노, 라즈베리파이, ROS, 3D 프린터 스마트폰



마무리

- ✓ 취미 생활 DIY로써 즐거움 제공
- ✓ 자녀 들과의 공통의 취미생활 제공
- ✓ 다양한 분야의 사람들과의 소통의 기회 제공
- ✓ 새로운 비즈니스 모델의 가능성



감사합니다.

e2g1234@naver.com