Cyber Physical System

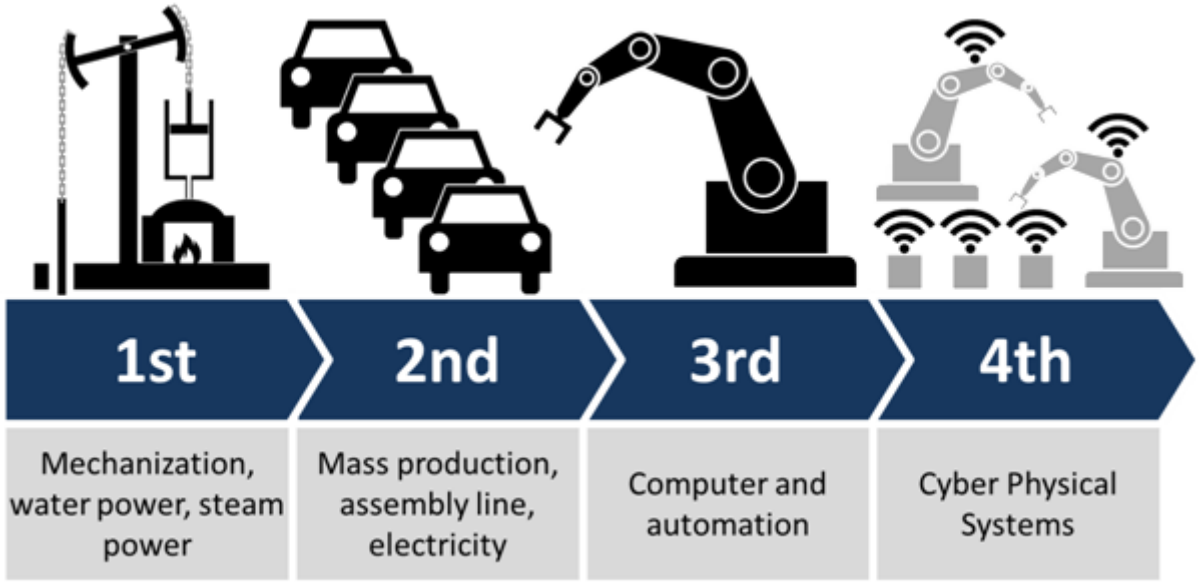
전북대학교
이성현

목차

I. CPS란?

CPS

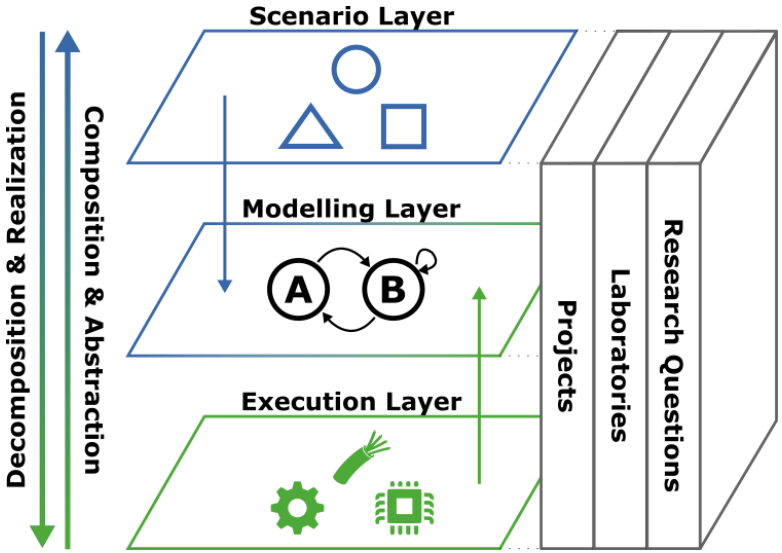
- CPS (Cyber Physical System)
  - 프로그래밍으로 만들어진 가상 세계와 물리적인 실제의 세계를 통합하는 시스템
  - 가상 공간의 컴퓨터가 네트워크를 통해서 실제의 물리 환경을 제어하는 기술
  - IoT와 밀접한 관련이 있음
  - 제4차 산업혁명을 이끌어가기 위한 기술 중 하나



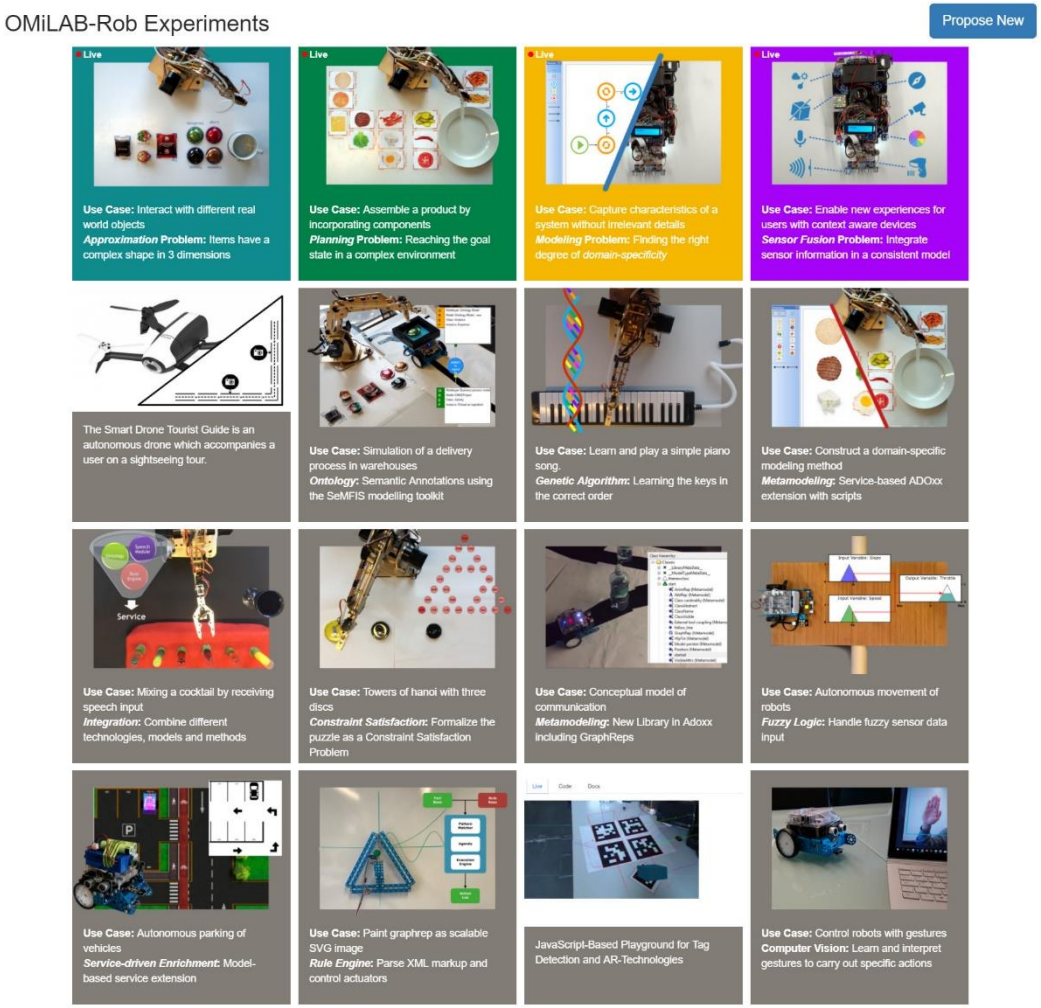
II. OMiPOB

OMiPOB

- OMiPOB (OMiLAB Physical Objects)
  - OMiLAB에서 주관하는 Project
    - Conceptual model을 CPS와 통합하기 위한 목적
    - 모델링을 통해 CPS를 설계 및 구현하는 것이 목적임
  - 3-Layer
    - Scenario Layer
    - Modeling Layer
    - Execution Layer
  - 여러 실험들이 진행되고 있음
    - http://austria.omilab.org/psm/omirob



OMiPOB


OMiLAB-Rob Experiments
Propose New
Live
Use Case: Interact with different real world objects
Approximation Problem: Items have a complex shape in 3 dimensions
Live
Use Case: Assemble a product by incorporating components
Planning Problem: Reaching the goal state in a complex environment
Live
Use Case: Capture characteristics of a system without irrelevant details
Modeling Problem: Finding the right degree of domain-specificity
Live
Use Case: Enable new experiences for users with context aware devices
Sensor Fusion Problem: Integrate sensor information in a consistent model
The Smart Drone Tourist Guide is an autonomous drone which accompanies a user on a sightseeing tour.
Use Case: Simulation of a delivery process in warehouses
Ontology: Semantic Annotations using the SeMFIS modelling toolkit
Use Case: Learn and play a simple piano song.
Genetic Algorithm: Learning the keys in the correct order
Use Case: Construct a domain-specific modeling method
Metamodeling: Service-based ADOxx extension with scripts
Service
Use Case: Mixing a cocktail by receiving speech input
Integration: Combine different technologies, models and methods
Use Case: Towers of hanoi with three discs
Constraint Satisfaction: Formalize the puzzle as a Constraint Satisfaction Problem
Use Case: Conceptual model of communication
Metamodeling: New Library in Adoxx including GraphReps
Use Case: Autonomous movement of robots
Fuzzy Logic: Handle fuzzy sensor data input
Use Case: Autonomous parking of vehicles
Service-driven Enrichment: Model-based service extension
Use Case: Paint graphrep as scalable SVG image
Rule Engine: Parse XML markup and control actuators
JavaScript-Based Playground for Tag Detection and AR-Technologies
Use Case: Control robots with gestures
Computer Vision: Learn and interpret gestures to carry out specific actions

III. CPS 구성

CPS 구성

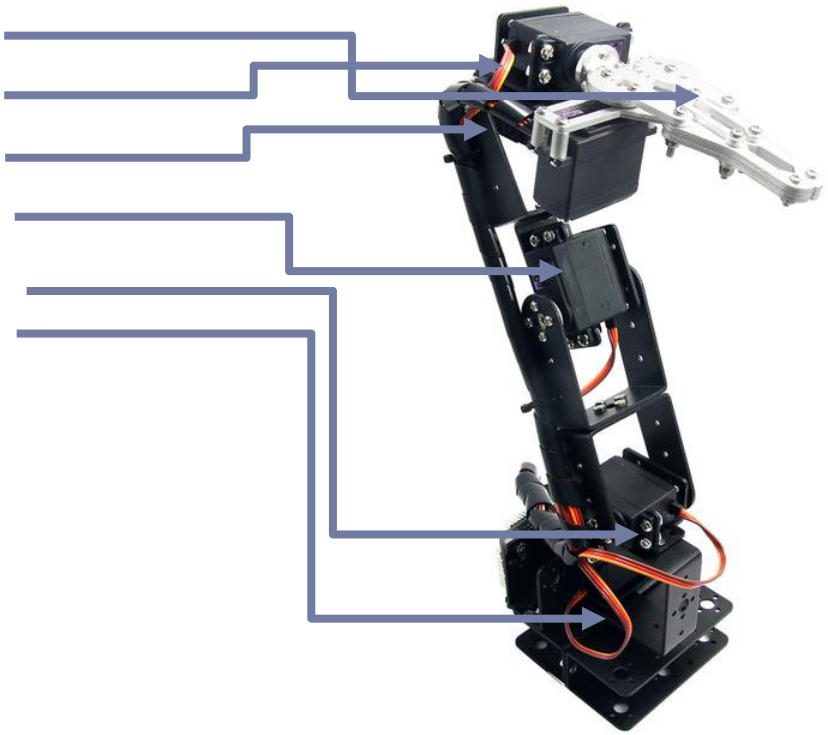
I. HW 제원

II. SW 정보 및 통신 방식

III. CPS library with ADOxx

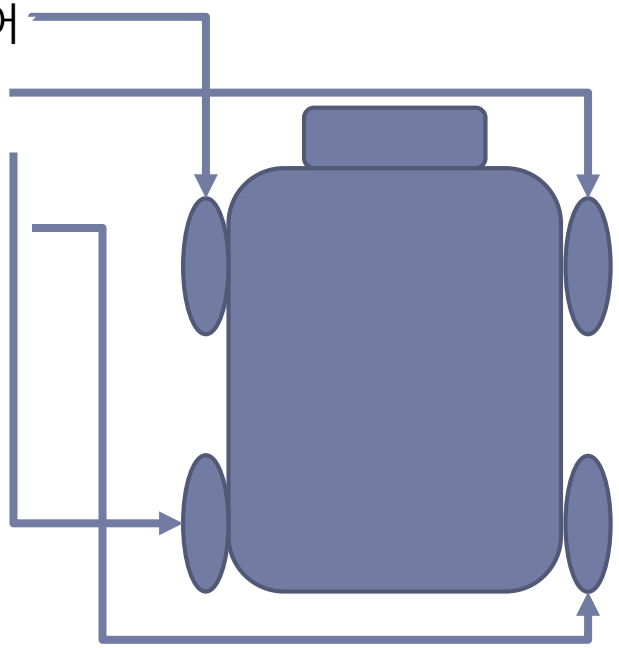
Class with Attribute

- RobotArm의 하위 Class
  - Motor
    - Motor Number (열거형)
      - M1: 집게 제어 모터
      - M2: 집게 회전 모터
      - M3: 팔 움직임 제어 모터
      - M4: 팔 움직임 제어 모터
      - M5: 팔 움직임 제어 모터
      - M6: 팔 움직임 제어 모터
    - Degree (정수)



Class with Attribute

- Automobile의 하위 Class
  - Automobile Motor
    - M1 (열거형): {HIGH, LOW} 좌상단 움직임 제어
    - M2 (열거형): {HIGH, LOW} 우상단 모터 제어
    - M3 (열거형): {HIGH, LOW} 좌하단 모터 제어
    - M4 (열거형): {HIGH, LOW} 우하단 모터 제어
    - Execution Time (ms) (실수): 실행 시간
  - Trace Line
    - Mode (열거형): {On, Off} 라인 트레이서 Mode 설정
    - Execution Time (ms) (실수): 실행 시간



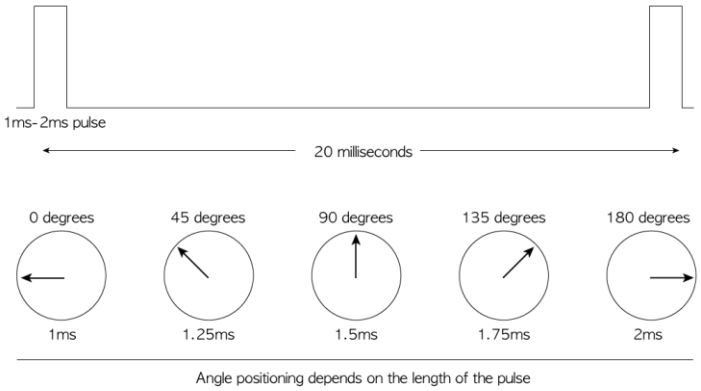
HW 제원

- Arduino
  - 프로그래밍을 쉽게 할 수 있도록 돕는 통합개발환경
  - 오픈소스 기반
- Arduino UNO
  - Arduino board 중 가장 범용적으로 쓰임
  - Micro Controller Unit (MCU)
    - CPU와 그 주변 장치를 하나의 칩으로 만든 반도체
  - 다수의 센서 및 통신 모듈을 연결하여 프로젝트를 구현할 수 있도록 도와줌

Fig. 2 우노 R3의 기본 부품 및 핀 정보

HW 제원

- 서보모터
  - 회전반경이 정해짐: 0~270도의 회전각을 가짐
  - 회전각도는 PWM(펄스폭변조) 방식으로 제어됨

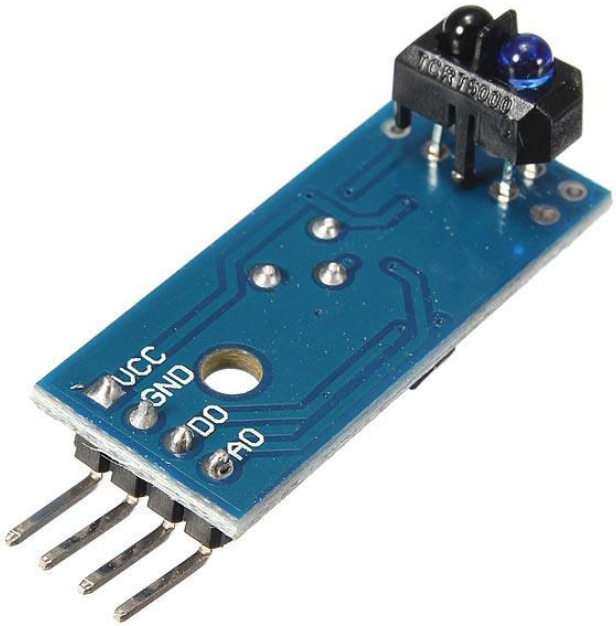


HW 제원

- ESP-8266
  - Wi-Fi 네트워킹을 지원하는 통신모듈
  - 저렴한 단가로 인해 IoT 구현 시 자주 쓰이는 모듈

HW 제원

- TCRT5000
  - 적외선 발사 센서
  - 적외선을 발사하고 반사되어 돌아오는 적외선을 검출하는 센서
    - 거리와 컬러 검출 가능

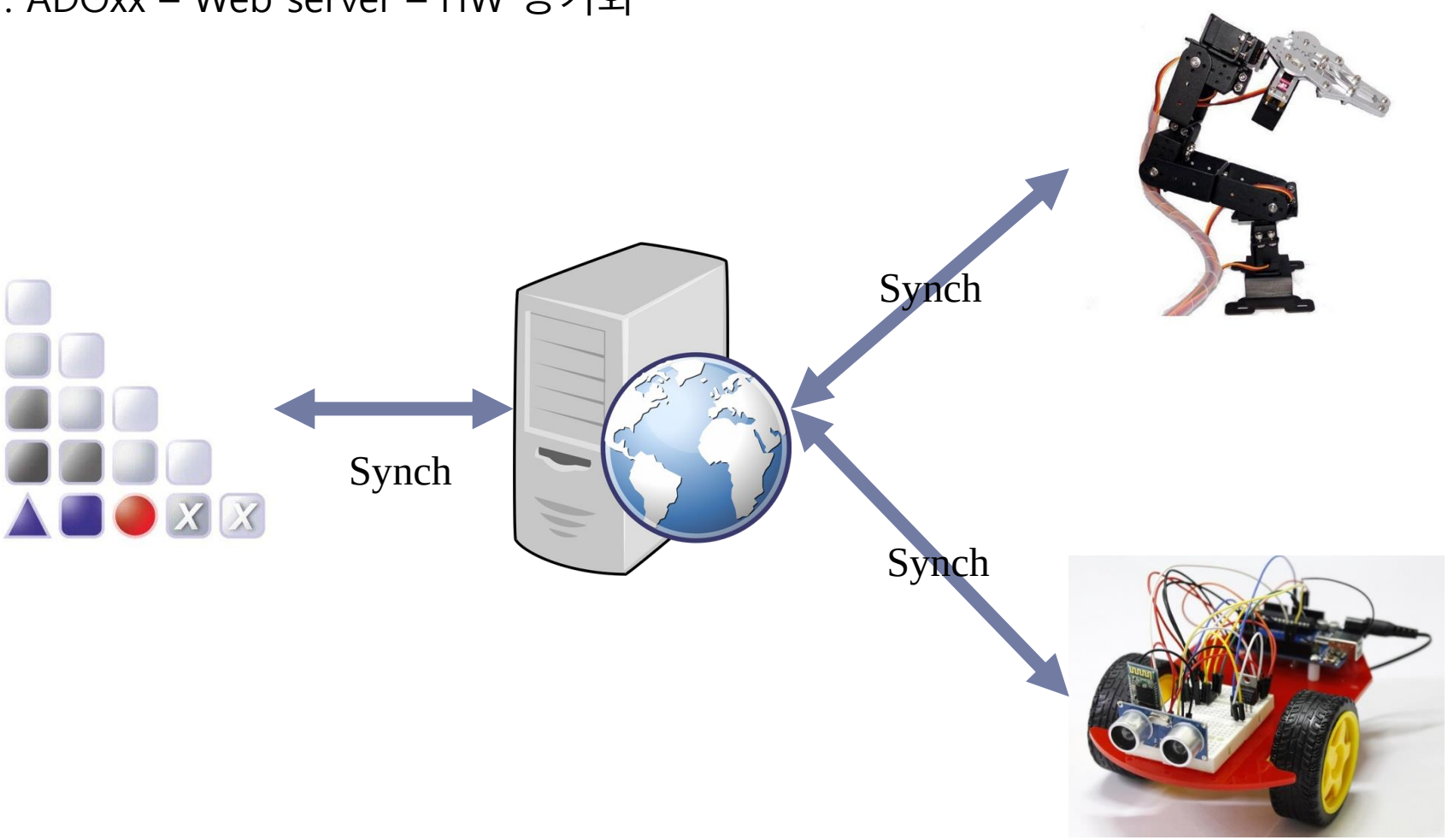


HW 제원

- HC-SR04
  - 초음파센서
  - 초음파를 감지하여 원하는 값을 되돌려줌

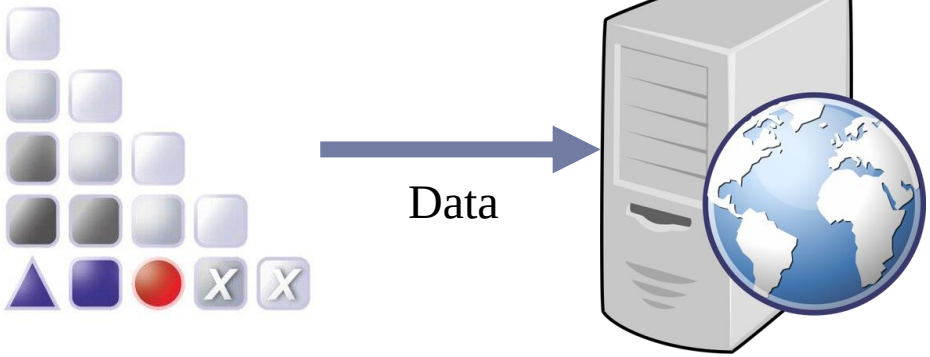
CPS 환경 및 구동 방식

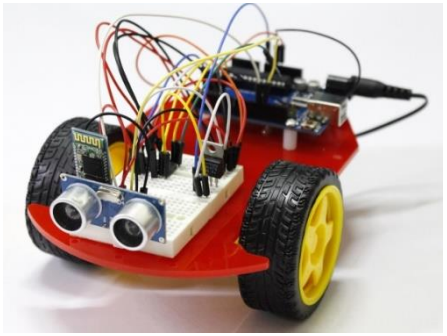
- 1. ADOxx – Web server – HW 동기화



CPS 환경 및 구동 방식

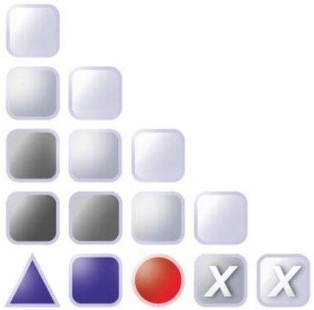
- 2. ADOxx – Web server 데이터 전송

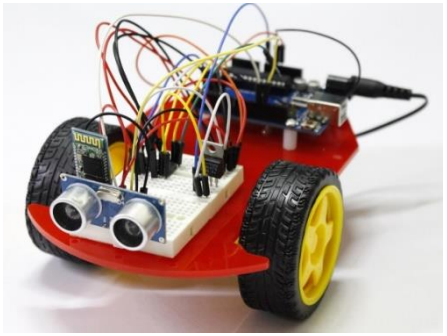




CPS 환경 및 구동 방식

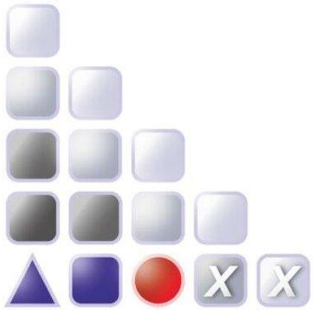
- 3. Web server - HW에 데이터 전송



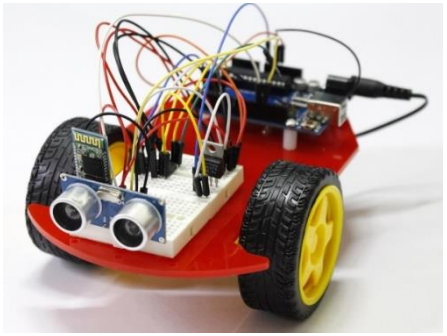


CPS 환경 및 구동 방식

- 4. HW – Web server 결과 데이터 전송

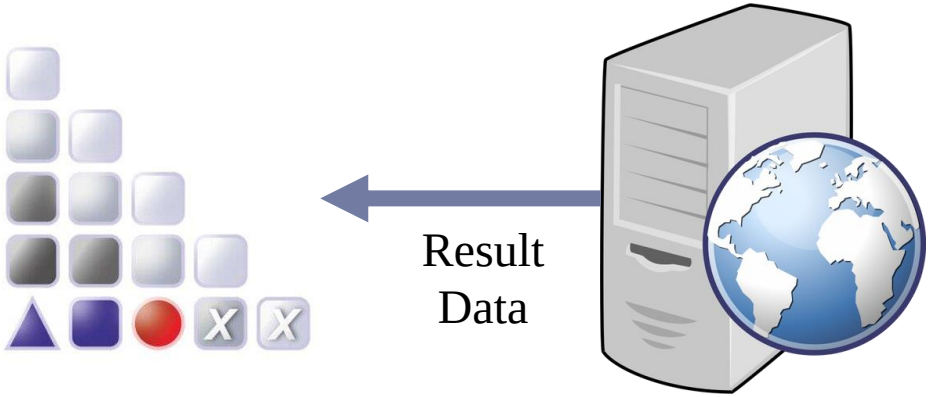


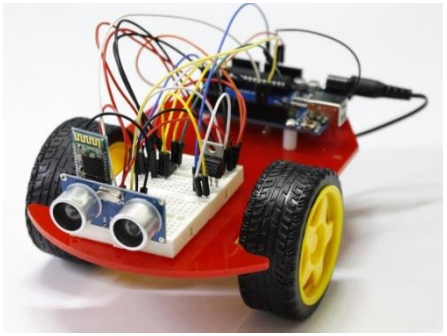
Result
Data



CPS 환경 및 구동 방식

- 5. Web server - ADOxx 결과 데이터 전송



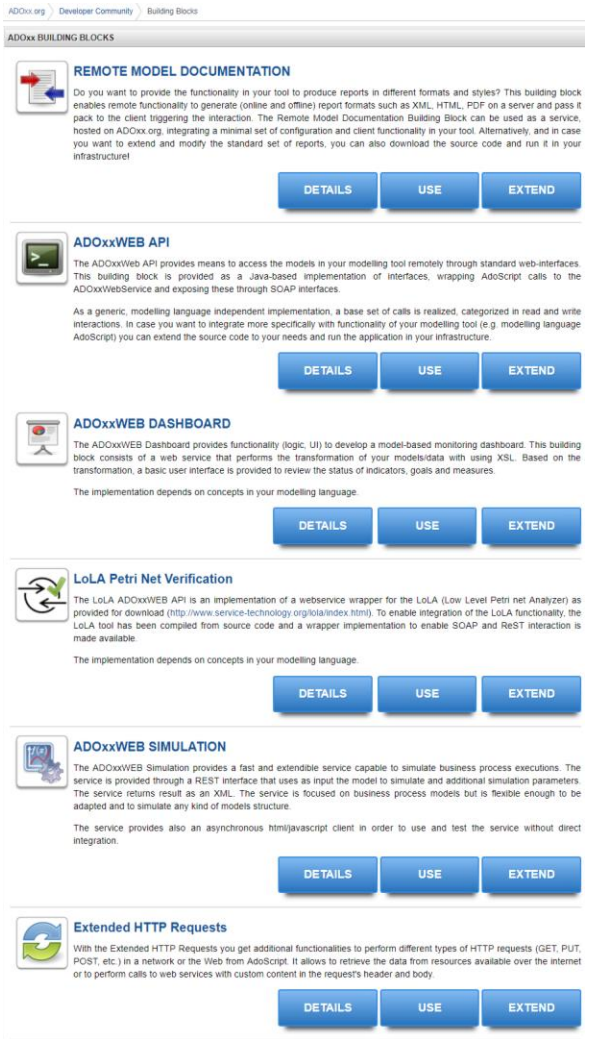


IV. Building Block

Building Block

- Building Block이란?
  - ADOxx의 기본 기능을 확장하는 개념
  - 외부 모듈을 ADOxx와 결합하여 ADOxx의 기능을 확장함
- Building Block을 공유하는 개발자들의 모임이 있음
  - ADOxx Community에서 제공하는 기본 Building Block
    - Remote Model Documentation
    - ADOxxWEB API
    - ADOxxWEB DASHBOARD
    - ....
  - 자신이 만든 Building Block을 공유 가능
  - https://www.adoxx.org/live/community

Building Block



V. Building Block for CPS

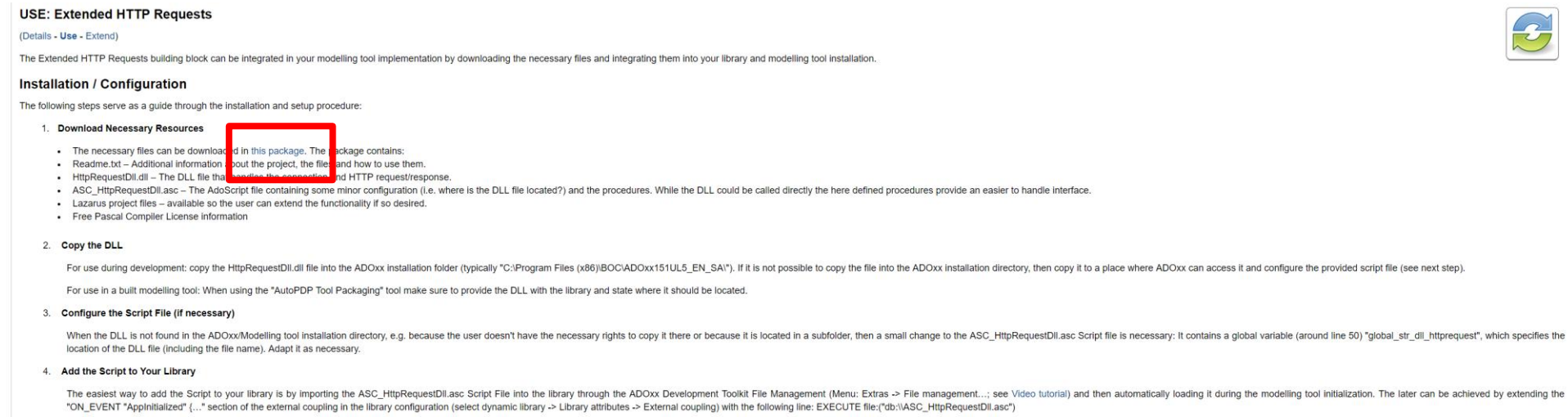
Extended HTTP Requests

- ADOxx에서 제공하는 기본 함수는 HW와 직접통신에 제약이 있음
  - ADOxx 홈페이지에서 제공하는 Building block의 설치가 필요함

- Extended HTTP Requests
  - 기존의 HTTP_REQUEST 사용 시, 보낼 수 없던 값이 있음
  - Extended HTTP Requests를 사용하면 추가 값을 보낼 수 있음
  - 총 6가지의 Procedure를 제공함

Extended HTTP Requests

- 설치 방법
  - 1. 아래의 주소에 들어가 Package 다운
    - https://www.adoxx.org/live/extended-http-requests-use



USE: Extended HTTP Requests

(Details - **Use** - Extend)

The Extended HTTP Requests building block can be integrated in your modelling tool implementation by downloading the necessary files and integrating them into your library and modelling tool installation.

Installation / Configuration

The following steps serve as a guide through the installation and setup procedure:

1. **Download Necessary Resources**
   - The necessary files can be downloaded in this package. The package contains:
   - Readme.txt – Additional information about the project, the files and how to use them.
   - HttpRequestDll.dll – The DLL file that handles the connection and HTTP request/response.
   - ASC_HttpRequestDll.asc – The AdoScript file containing some minor configuration (i.e. where is the DLL file located?) and the procedures. While the DLL could be called directly the here defined procedures provide an easier to handle interface.
   - Lazarus project files – available so the user can extend the functionality if so desired.
   - Free Pascal Compiler License information
2. **Copy the DLL**

   For use during development: copy the HttpRequestDll.dll file into the ADOxx installation folder (typically "C:\Program Files (x86)\BOC\ADOxx151UL5_EN_SA\"). If it is not possible to copy the file into the ADOxx installation directory, then copy it to a place where ADOxx can access it and configure the provided script file (see next step).

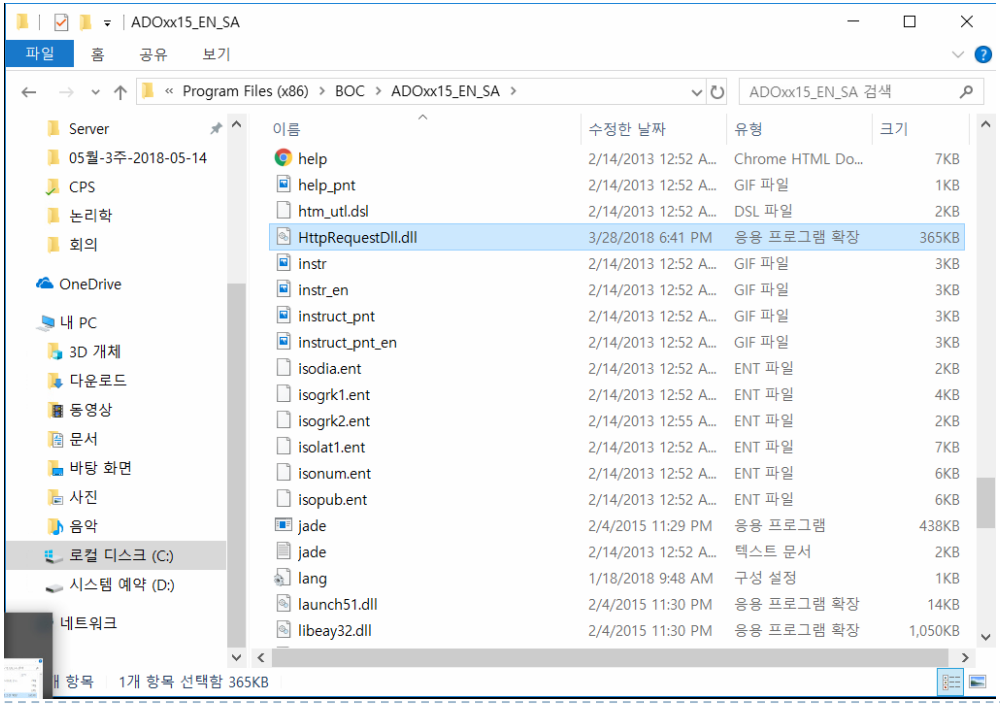
   For use in a built modelling tool: When using the "AutoPDP Tool Packaging" tool make sure to provide the DLL with the library and state where it should be located.
3. **Configure the Script File (if necessary)**

   When the DLL is not found in the ADOxx/Modelling tool installation directory, e.g. because the user doesn't have the necessary rights to copy it there or because it is located in a subfolder, then a small change to the ASC_HttpRequestDll.asc Script file is necessary: It contains a global variable (around line 50) "global_str_dll_httprequest", which specifies the location of the DLL file (including the file name). Adapt it as necessary.
4. **Add the Script to Your Library**

   The easiest way to add the Script to your library is by importing the ASC_HttpRequestDll.asc Script File into the library through the ADOxx Development Toolkit File Management (Menu: Extras -> File management...; see Video tutorial) and then automatically loading it during the modelling tool initialization. The later can be achieved by extending the "ON_EVENT "AppInitialized" {..." section of the external coupling in the library configuration (select dynamic library -> Library attributes -> External coupling) with the following line: EXECUTE file:("db:\\ASC_HttpRequestDll.asc")

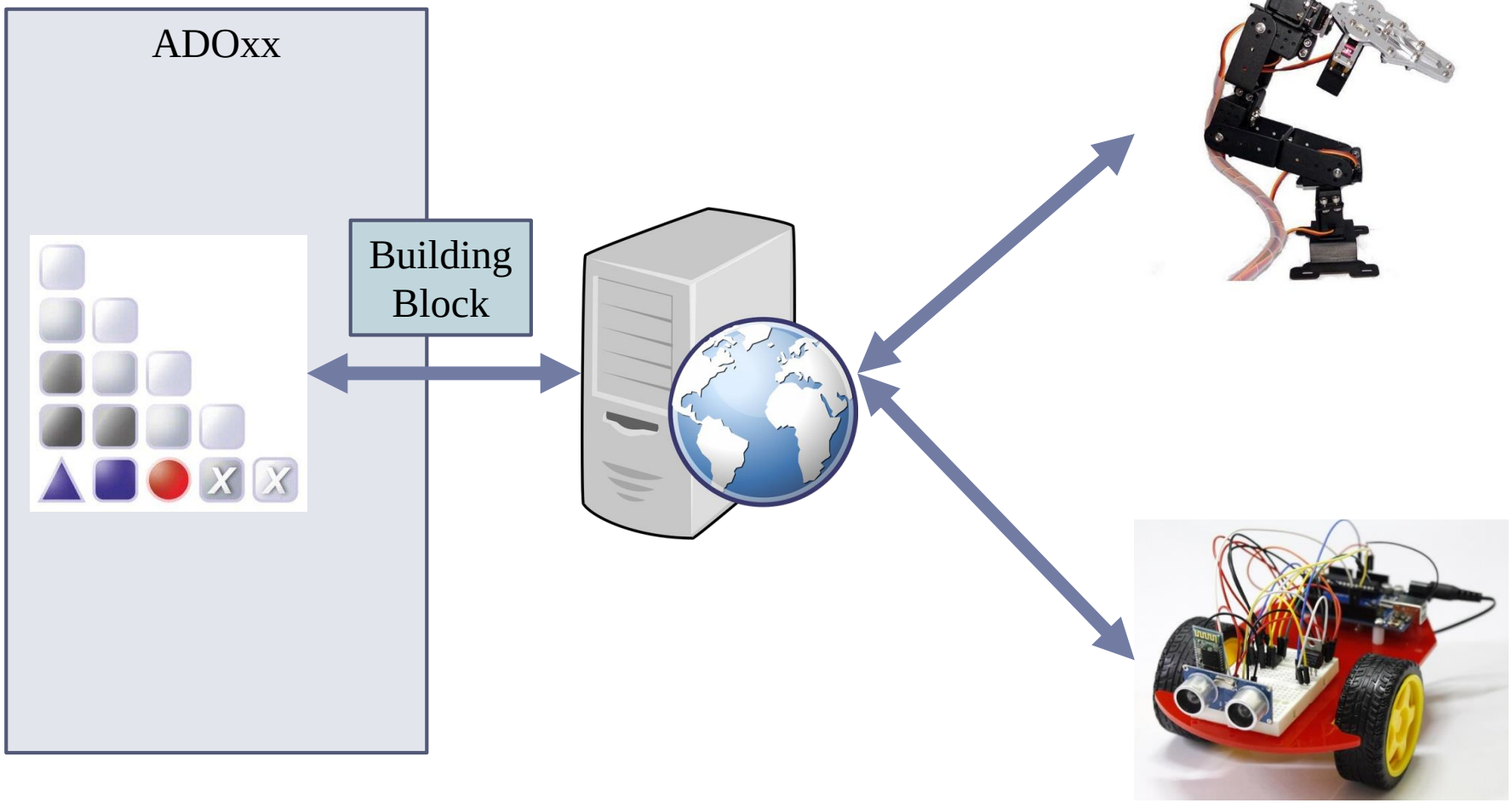
Extended HTTP Requests

- 설치 방법
  - 2. 압축을 풀면 다음과 같은 파일이 있음
    - HttpRequestDll.dll
  - 3. HttpRequestDll.dll 파일을 ADOxx 설치경로에 복사
    - 기본경로: C:₩Program Files (x86)₩BOC₩ADOxx151UL5_EN_SA



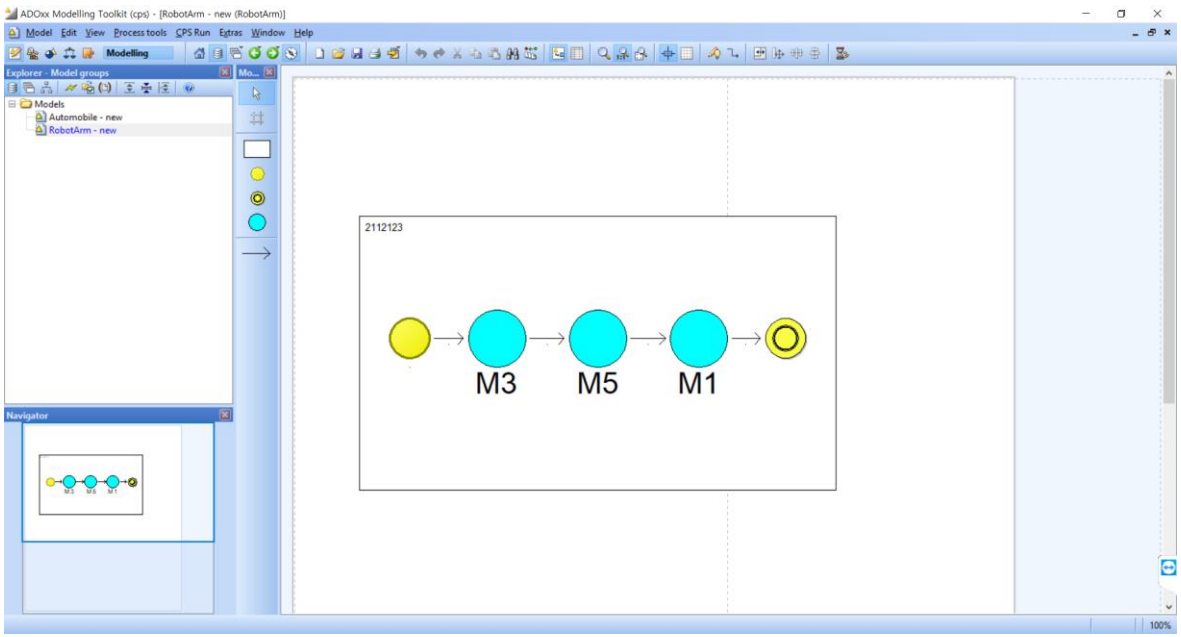
Building Block for CPS

- 1. Building Block의 영역



Building Block for CPS

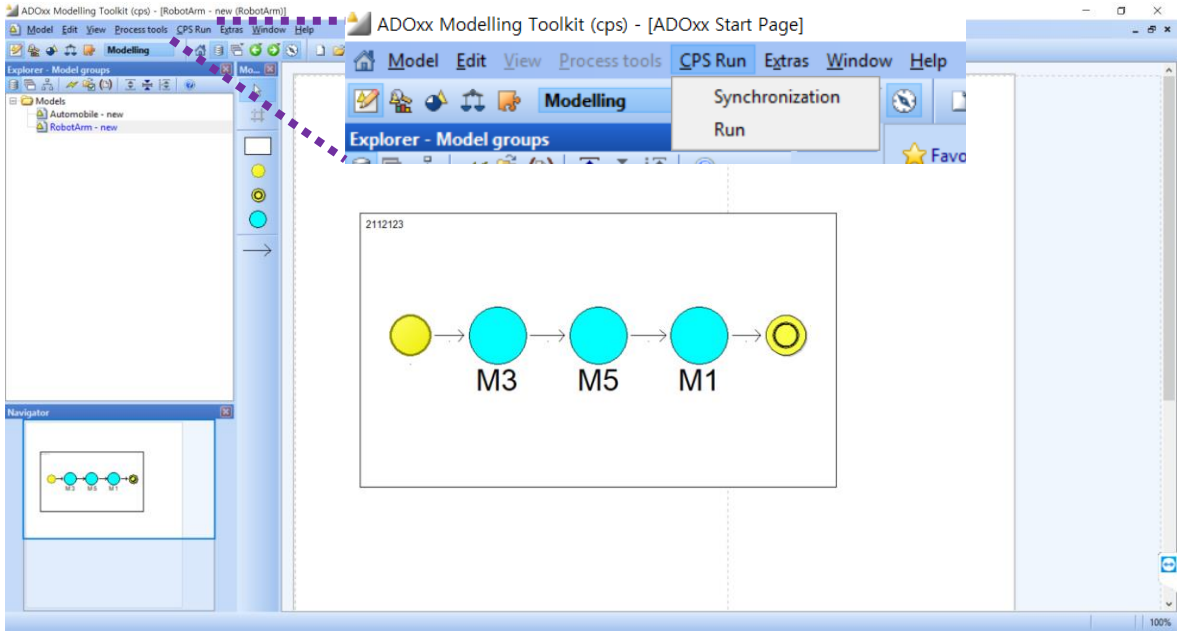
- ADOxx – Web server 데이터 전송 과정
  - 1. ADOxx의 Model에서 데이터 수집 및 정제



OMLAB KOREA

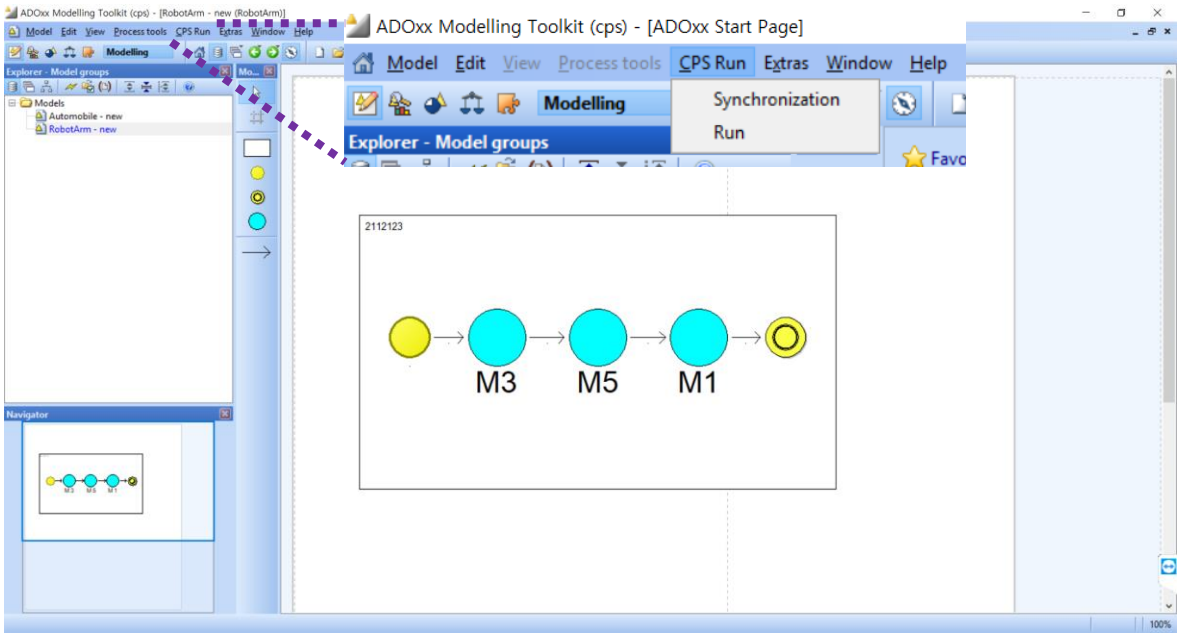
Building Block for CPS

- ADOxx – Web server 데이터 전송 과정
  - 2. Web server와 동기화
    - 동기화가 완료되면: “Complete” 메시지 받음
    - 동기화가 왼료되지 못하면: “Incomplete” 메시지 받음



Building Block for CPS

- ADOxx – Web server 데이터 전송 과정
  - 3. Web server에 데이터 전송 (txt파일)
    - 데이터 구조: JSON 형식



Building Block for CPS

- Web server 데이터 전송 과정
  - 1. HW - Web server 동기화
  - 2. HW – Web server 동기화가 완료되면 Session을 사용하여 연결 상태 유지
  - 3. ADOxx로부터 동기화 요청 받은 후 동기화
  - 4. Session 사용하여 ADOxx – Web server 연결 상태 유지
  - 5. ADOxx로부터 데이터 받음 (txt파일(JSON형식))
  - 6. Web server에 데이터 저장
  - 7. HW에 데이터 전송 (JSON 형식)
  - 8. Web server 예제 코드: server.js

Building Block for CPS

- Web server 파일 구성
  - Package.json: 헤더파일을 불러오는 역할
  - Passport.js: 로그인과 관련된 코드
  - Route.js: Routing과 관련된 코드
  - Server.js: 서버 구성과 관련된 코드
  - Socket.js: 웹소켓과 관련된 코드

CPS Library 및 Source code 다운

- 아래의 파일을 다운로드 후 실행
  - CPS Library.abl
  - Automobile-2.asc
    - 자동차 실행파일
  - Robotarm-3.asc
    - 기계팔 실행파일
- 실행 방법
  - Modeling toolkit에서 Shell window를 통해 실행할 것
  - 실행 전에, 실행할 모델을 활성화 시킨 후 사용해야 함

VI. HW 설정

HW 준비

- HW 준비 시 주의사항
  - 주의사항
    - 1. HW 전원을 켠 후, HW의 led가 켜진 것을 확인 후 작동시킬 것
      - 통신 모듈의 server mode가 설정됐을 때, led가 켜짐
      - 만약 led가 켜지지 않을 경우 전원을 완전히 제거 후, 다시 시작하기 바람
    - 2. 기계팔의 경우, 통신 도중에 모터들이 조금씩 움직이는 현상이 있음
      - HW 상의 문제이므로 통신에 지장을 주지 않음
      - 해당 문제점 수정예정
    - 3. 자동차의 경우, 동작 명령 시 HW가 재시작되는 경우가 있음
      - 배터리의 전류 부족으로 인해, Arduino가 리셋되는 경우임
      - 배터리를 완전히 충전 후 사용하면 문제 해결됨
      - 또는 간단한 테스팅 시, Arduino의 USB 포트를 직접 컴퓨터에 연결하여 전류를 공급해주면 문제 해결됨

참고자료

- http://www.hardcopyworld.com/gnuboard5/bbs/board.php?bo_table=lecture_esp&wr_id=1
- https://kocoafab.cc/tutorial/view/354
- https://blog.naver.com/icbanq/220891355006
- http://mechasolutionwiki.com/index.php?title=%EC%95%84%EB%91%90%EC%9D%B4%EB%85%B8%EB%9E%80%3F