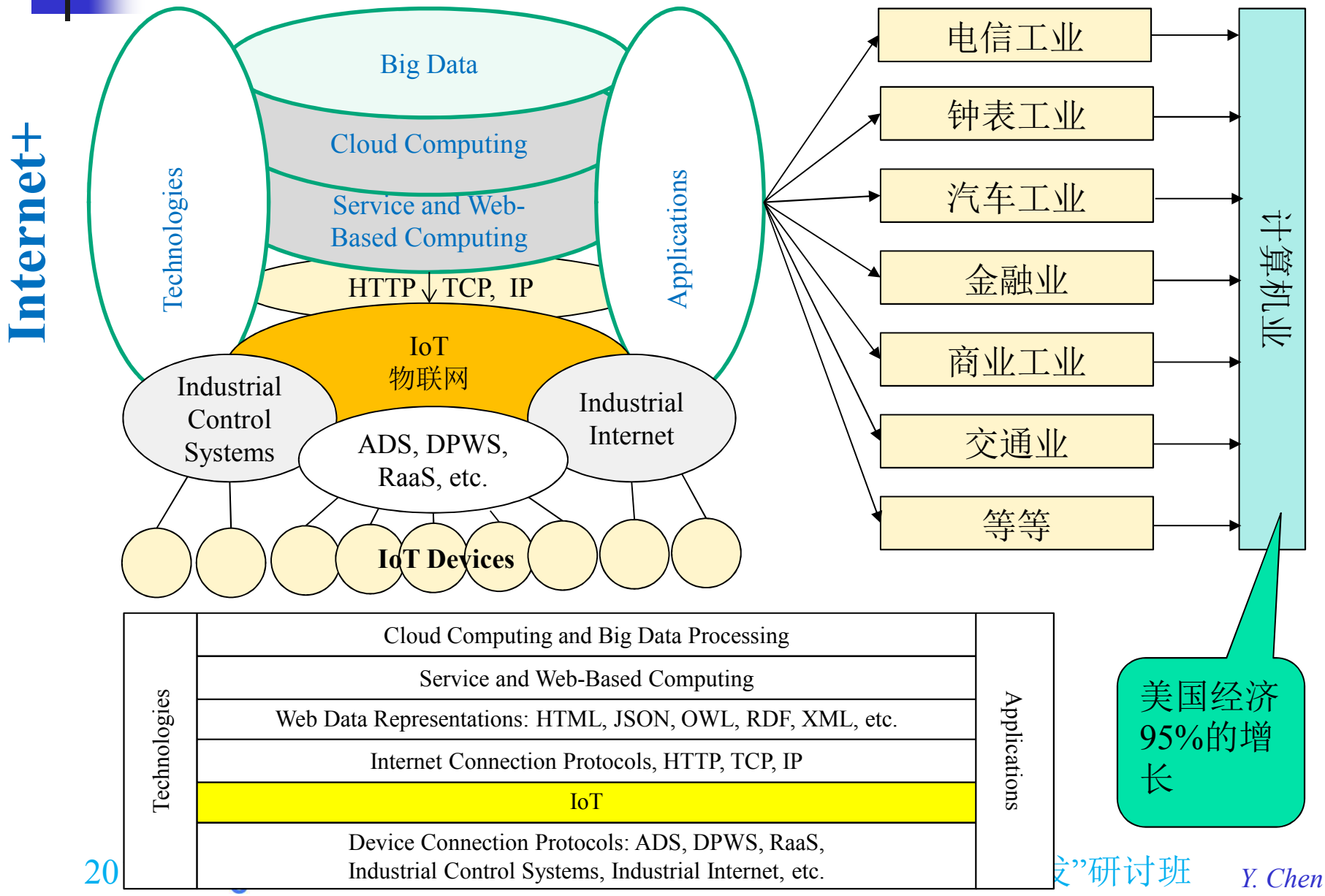
物联网和机器人导论
可视化编程实验环境

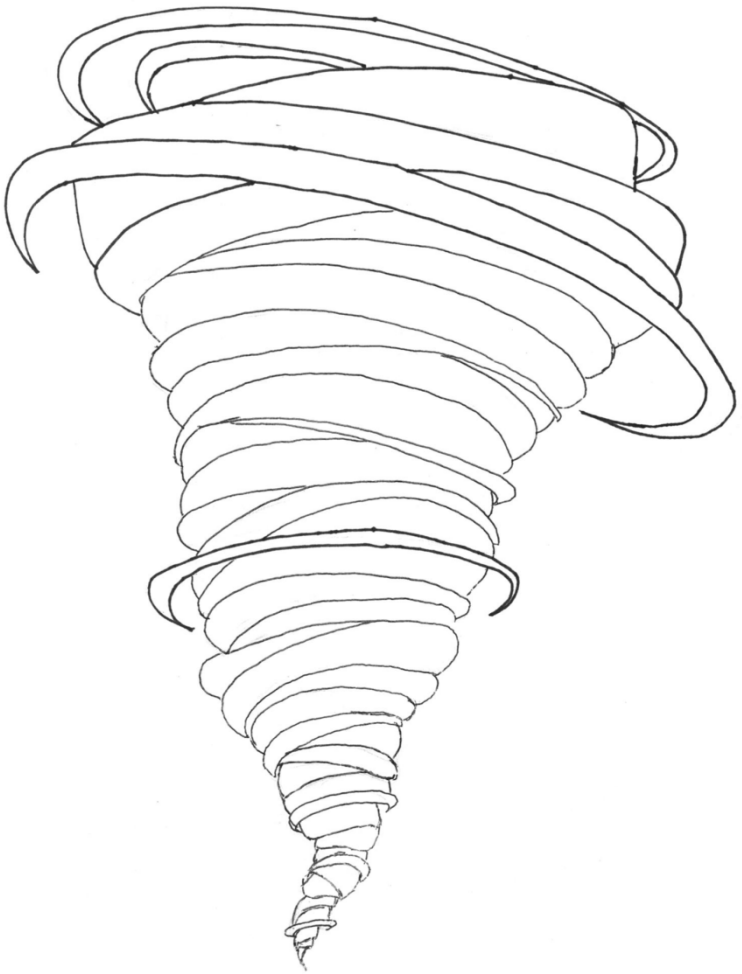
Introduction to IoT and Robotics, based on Visual Programming Experiments

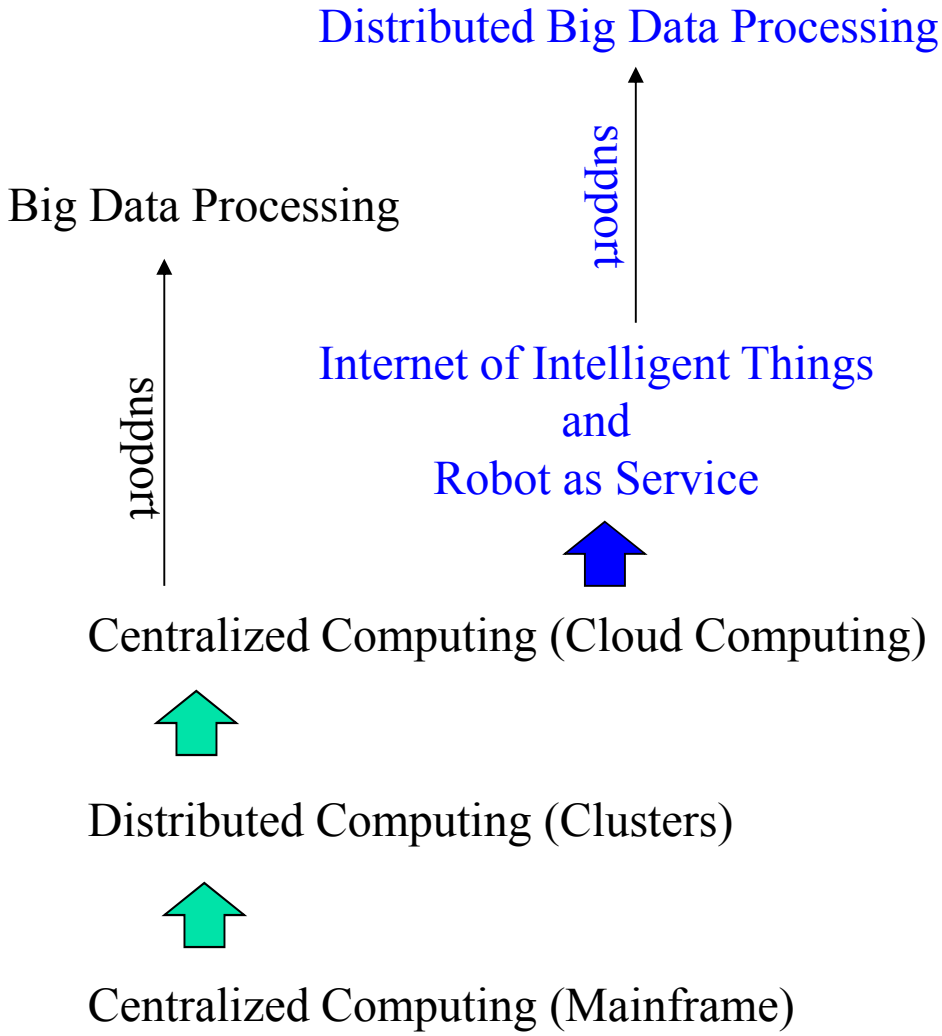
Yinong Chen
Arizona State University, U.S.A.

The Big Things behind Internet+



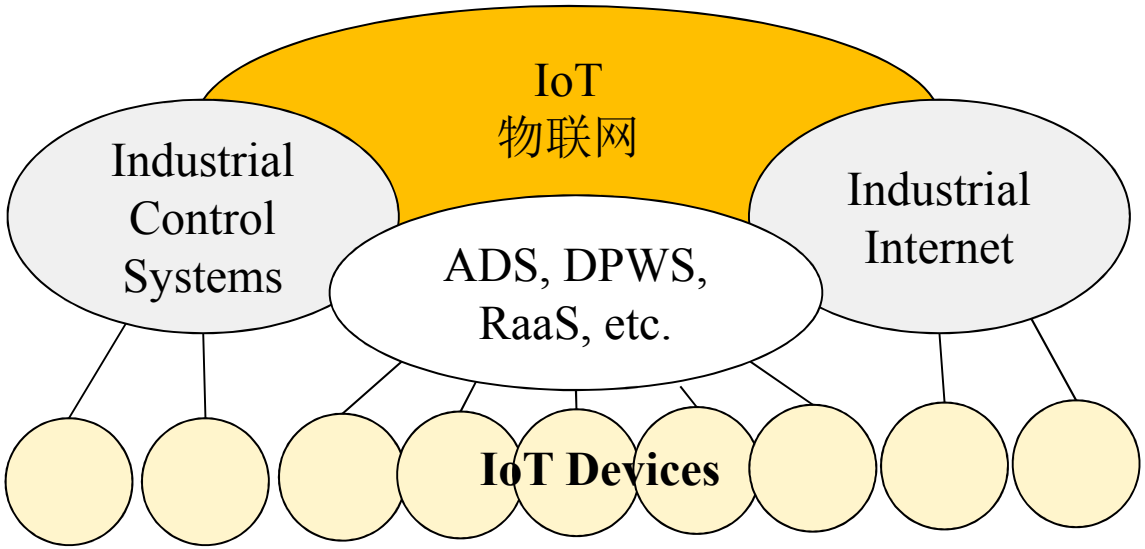
Spiral Model of Computing System Development



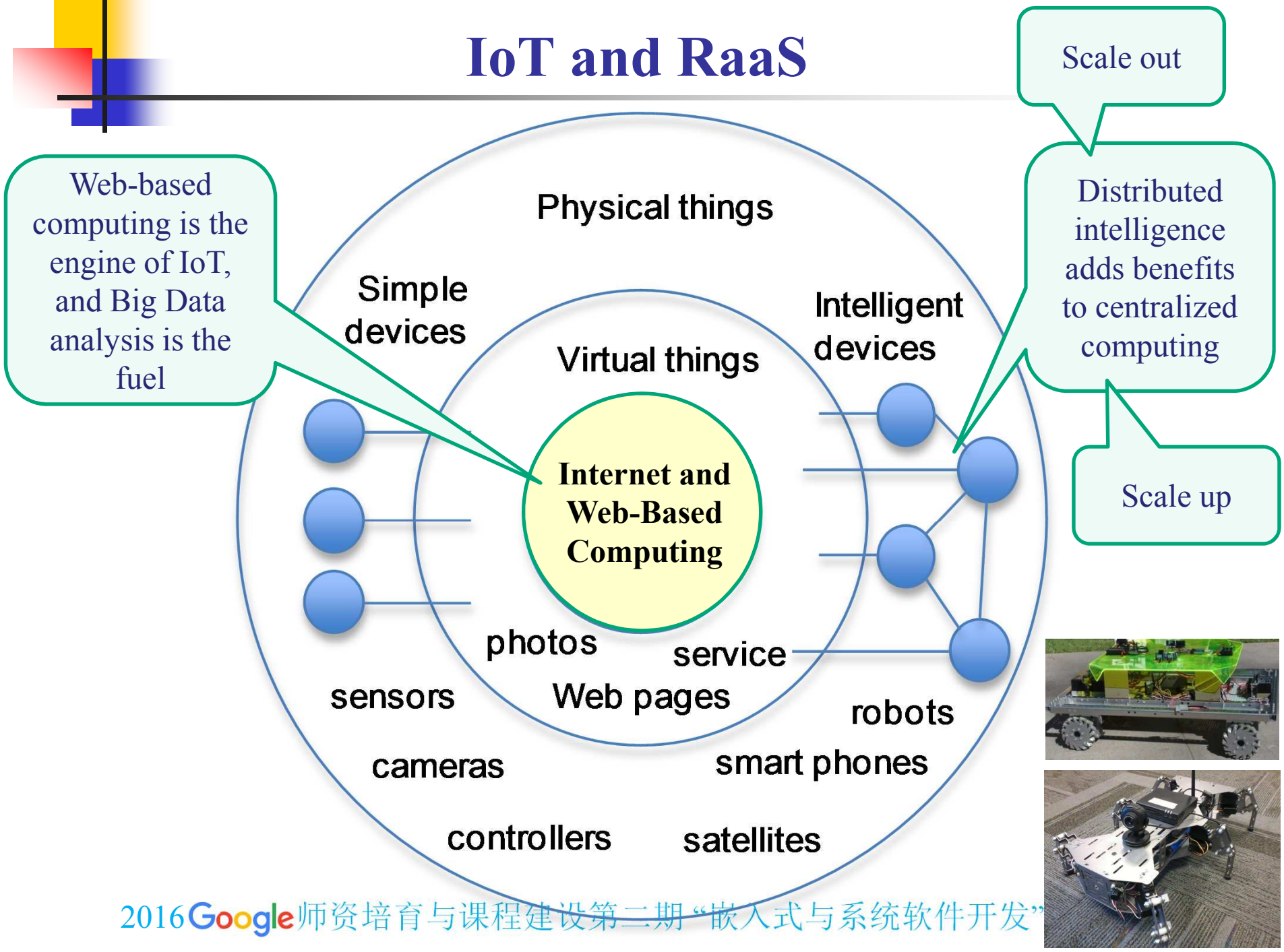


Lecture Outline

- **Internet of Things and Robot as a Service**
  - Device as a Service
  - Robot as a Service (RaaS)
  - My RaaS software and hardware for Computer Science Education



IoT and RaaS


Scale out
Web-based computing is the engine of IoT, and Big Data analysis is the fuel
Distributed intelligence adds benefits to centralized computing
Scale up
Physical things
Simple devices
Virtual things
Intelligent devices
Internet and Web-Based Computing
photos
service
sensors
Web pages
robots
cameras
smart phones
controllers
satellites

15B IoT Devices are the main Source of Big Data

Device-Generated Data

Big Data

Human-Generated Data

Cloud Computing

My Definition of RaaS

Robot as a Service

From Wikipedia, the free encyclopedia

Robot as a Service (or RaaS) is a cloud computing unit that facilities the seamless integration of robot and embedded devices into Web and cloud computing environment. In terms of Service-Oriented Architecture (SOA), a RaaS unit includes services for performing functionality, a service directory for discovery and publishing, and service clients for user's direct access.[1][2] The current RaaS implementation facilitates SOAP and RESTful communications between RaaS units and the other cloud computing units. Hardware support and standards are available to support RaaS implementation. Devices Profile for Web Services (DPWS) defines implementation constraints to enable secure Web Service messaging, discovery, description, and eventing on resource-constrained devices between Web services and devices. RaaS can be considered a unit of Internet of Things (IoT), Internet of Intelligent Things (IoIT) that deal with intelligent devices that have adequate computing capacity,[3] Cyber-physical system (CPS) that is a combination of a large computational and communication core and physical elements that can interact with the physical world,[4] and Autonomous decentralized System (ADS) whose components are designed to operate in a loosely coupled manner and data are shared through a content-oriented protocol [5] [6]

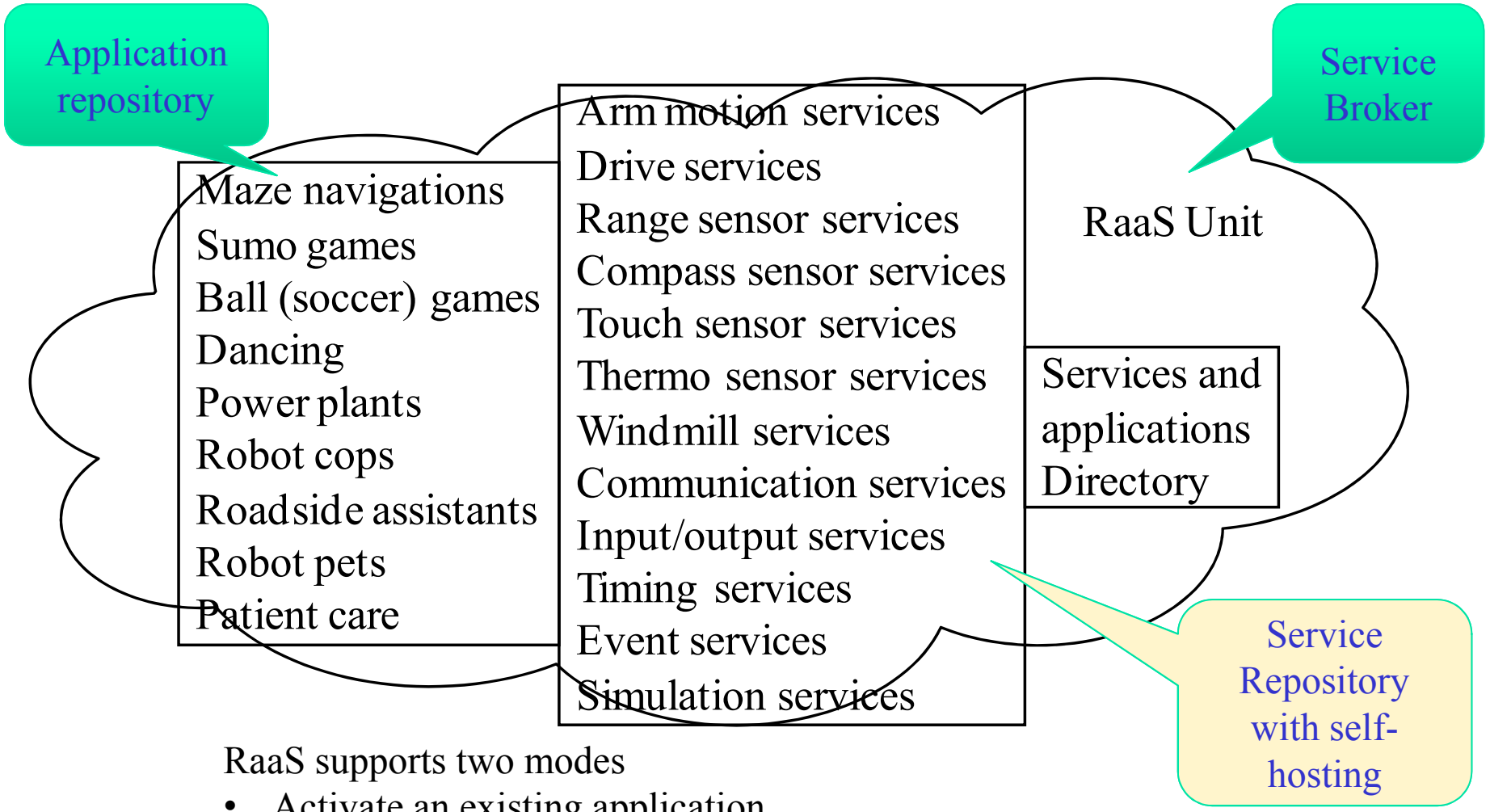
What is RaaS in Simple Words

RaaS (Robot as a Service) is

- an Embedded Intelligent System
- an Internet of Intelligent Thing
- a Cyber-Physical System
- an Autonomous Decentralized System
- a Service in Web and in Cloud Computing
- a Mobile Computing System
- a Real-Time System

Robot as a Service Architecture

Textbook Chapter 9 Section 9.3



RaaS supports two modes

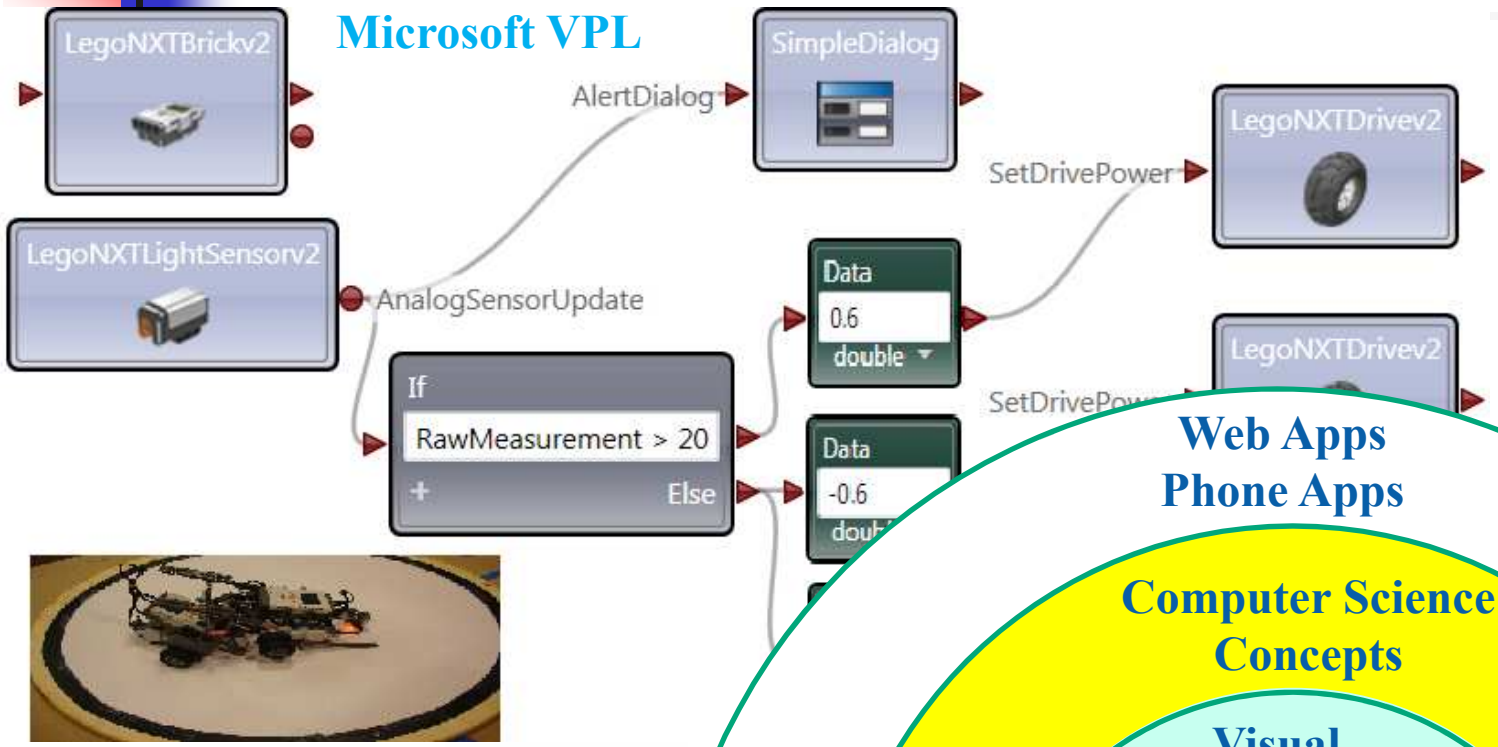
- Activate an existing application
- Compose a new Web application, which using the services

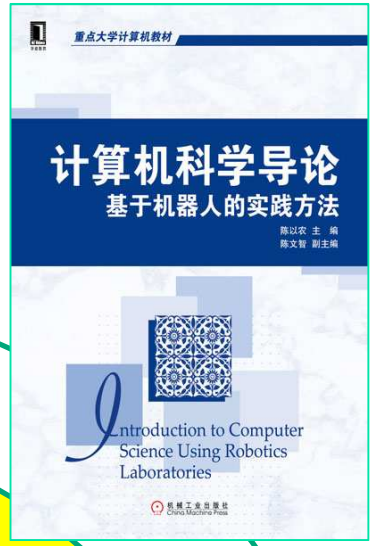
My RaaS Platforms and Implementations

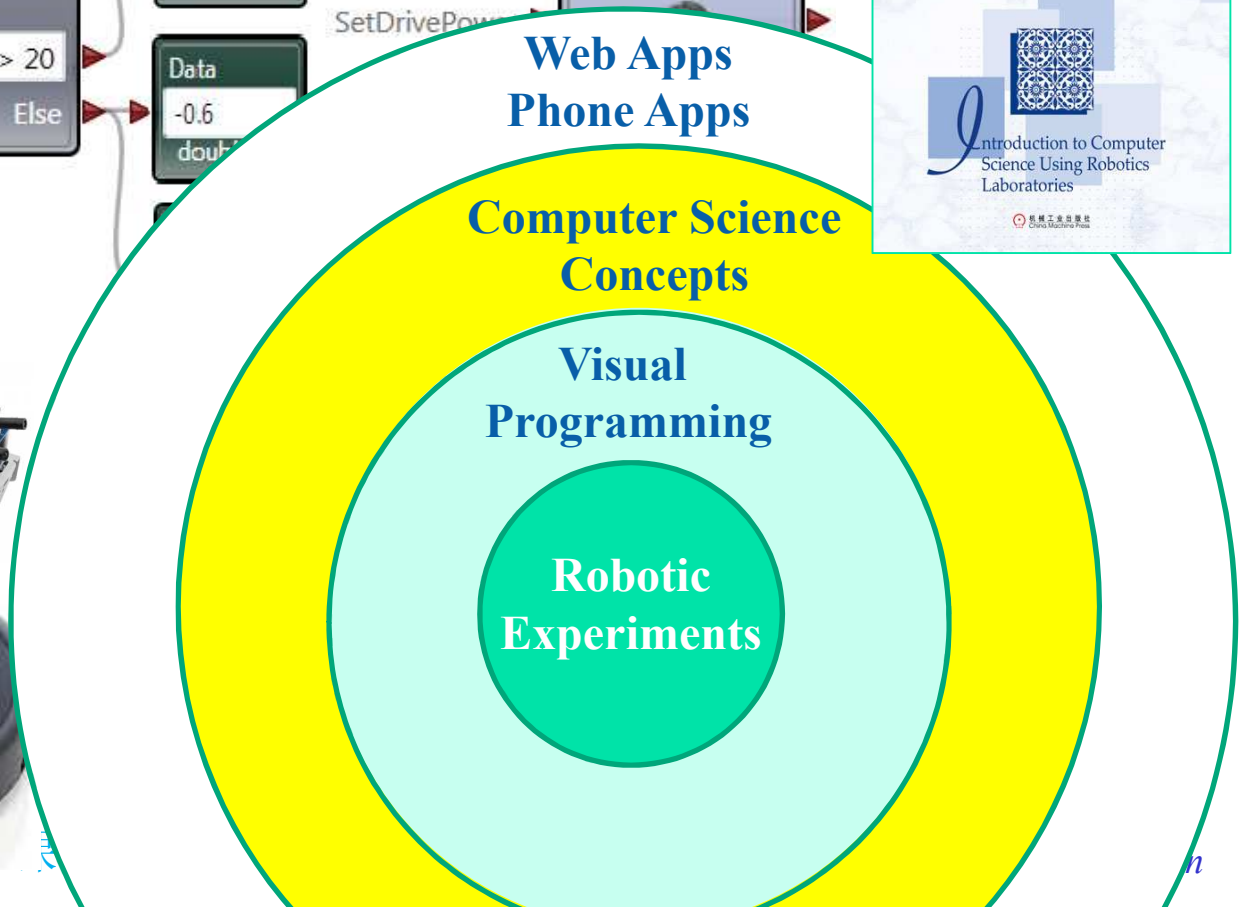
RaaS (Robot as a Service) is a cloud and Internet of Thing unit, consisting of Web Accessing Interface and Web-enabled mobile hardware

- Version 1, 2012 RaaS uses Intel Atom-based processor , with Parallax Hex Crawler Hardware
- Version 2, 2013, Cornell Cup, Using Web Programmable Interface
- Version 3, 2014 Intel Cup Shanghai, using Galileo and BayTrail,
- Version 4, 2015, Using Edison boards to build multiple robots. We are developing a new ASU Visual Programming Language, similar to Microsoft VPL, which can control RaaS over Internet

Before ASU IoT and RaaS





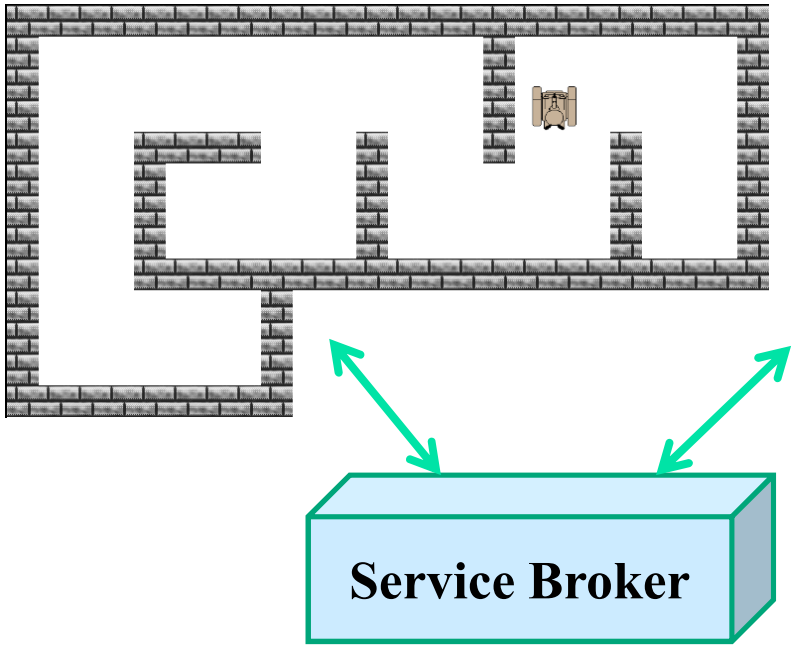


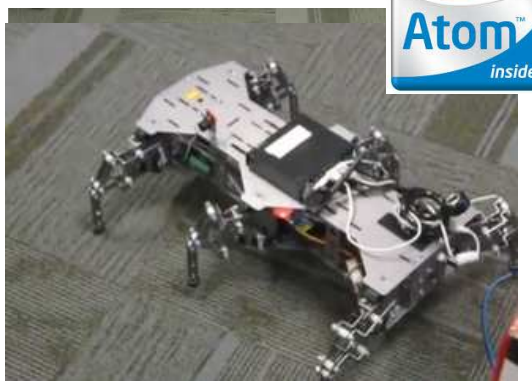
2016 G

ASU IoT and RaaS Version 1, 2012

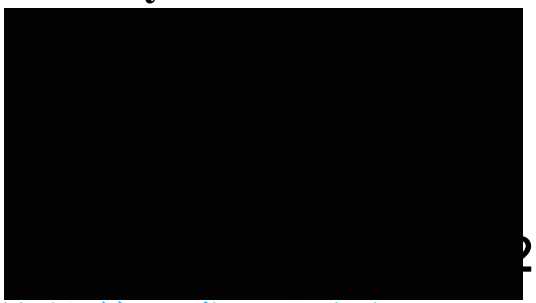
- It is based on Intel Atom processor and Parallax Hex Crawler robot framework
- The first robot that is a full RaaS unit and is controllable over the Web: http://venus.eas.asu.edu/WSRepository/RaaS/main/

Web Simulation Environment

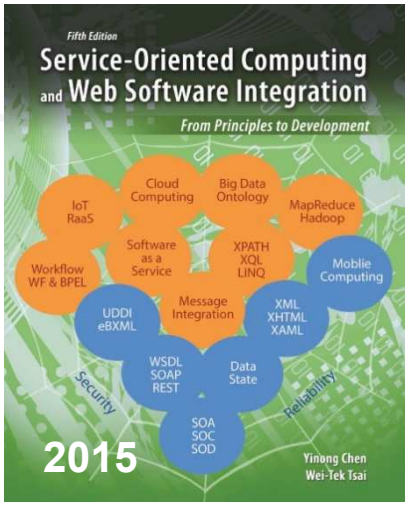




Physical Robot

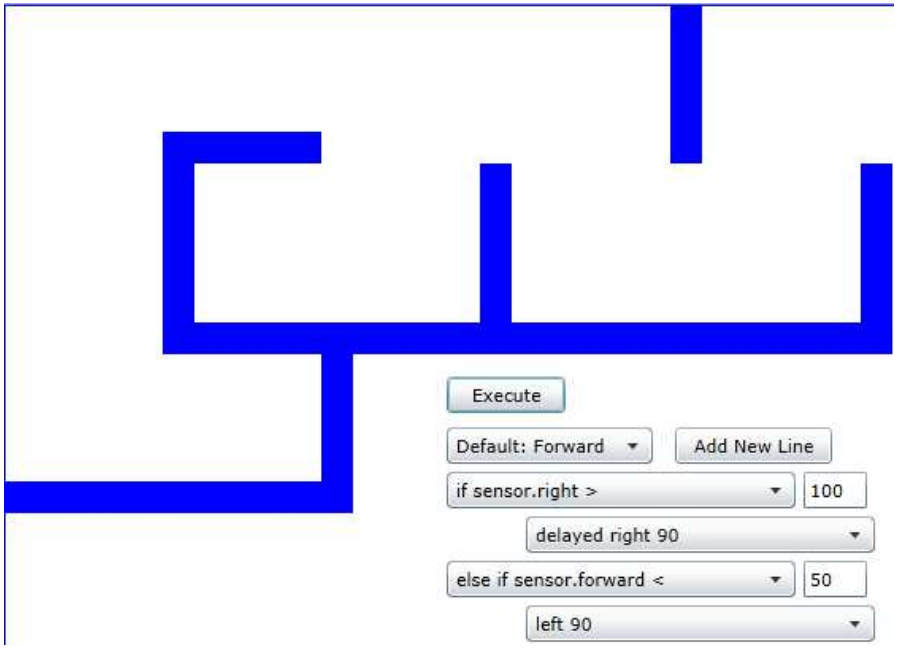


ASU IoT and RaaS Version 2, 2013



- **Cornell Cup Participant 2013**
- **The RaaS unit that can be programmed and controllable over the Web: http://venus.eas.asu.edu/WSRepository/eRobotic/**

Web Simulation & Programming Environment



Physical Robot

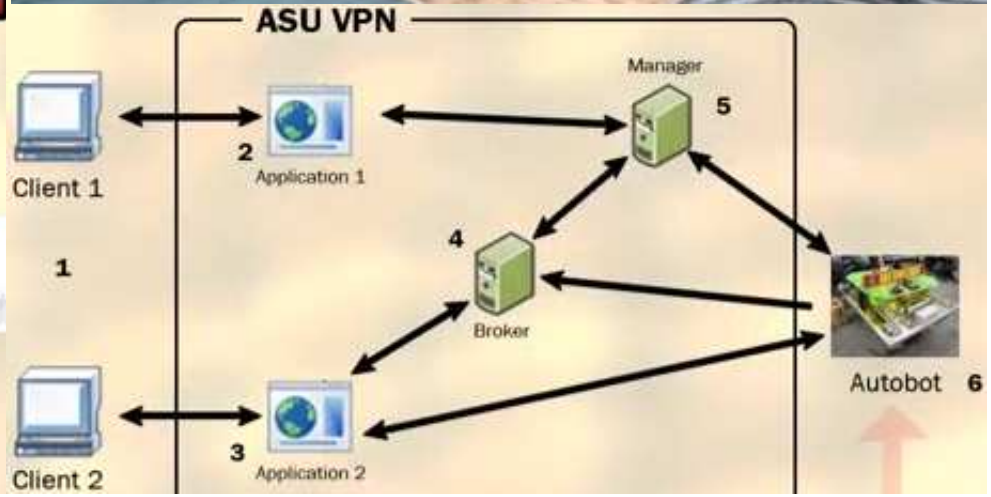
2013 Intel SINO-US
Intelligent Embedded System
Academic Workshop

August 21st-22nd,2013
Sichuan Province ,China

Intel-Based Robot

intel Education

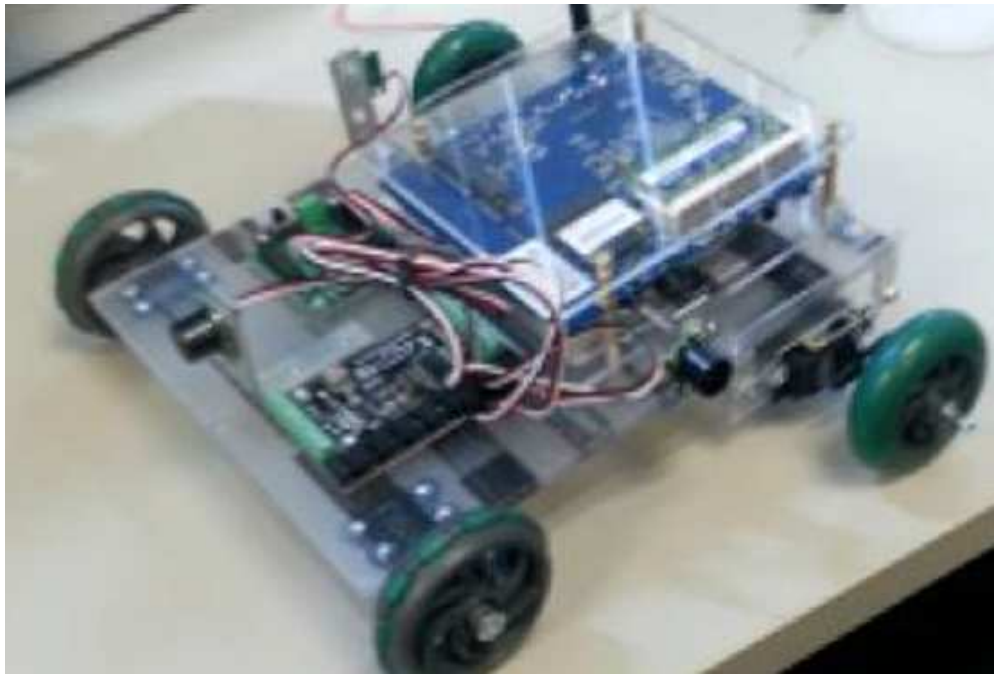
Cornell Cup: http://venus.eas.asu.edu/WSRepository/eRobotic/video2013.wmv



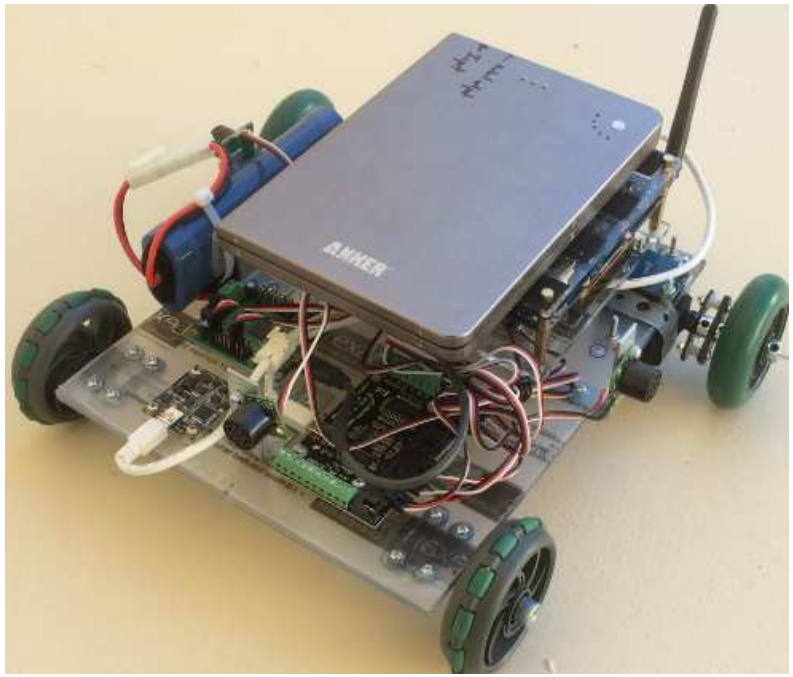
ASU IoT and RaaS Version 3 in 2014

- RaaS Hardware
  - Small robot better suitable for being used in large class.
    It is based on Quark and Atom;
  - The robot is an IoT unit;
- Software
  - The programming language is no longer Microsoft VPL. It is a Web-based graphic language – Made in ASU!
  - Program can control both simulated robot and physical robot.

ASU IoT and RaaS Version 3 2014 Hardware

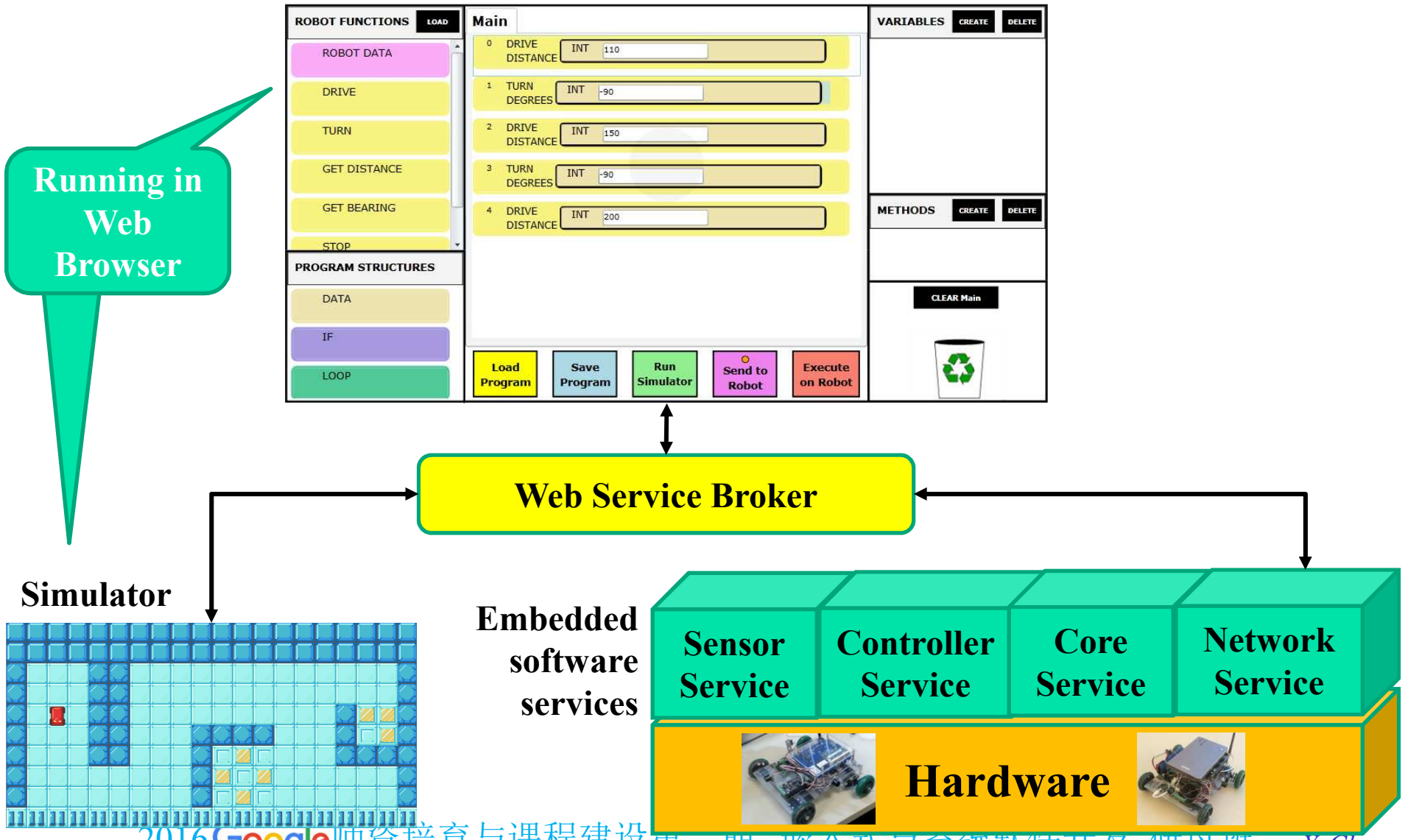


Galileo-based Version



Bay-Trail-based Version

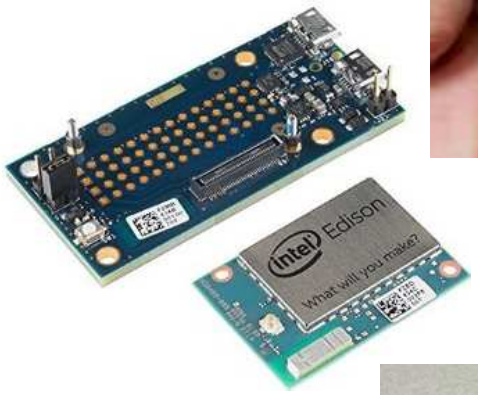
ASU IoT and RaaS Version 2014 System



ASU IoT and RaaS Version 4 in 2015

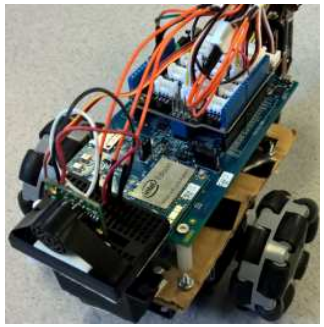
➢ RaaS Hardware

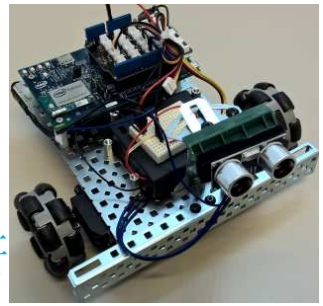
- The robot will be even smaller. It is based on Edison or Curie;
- Multiple robots work together



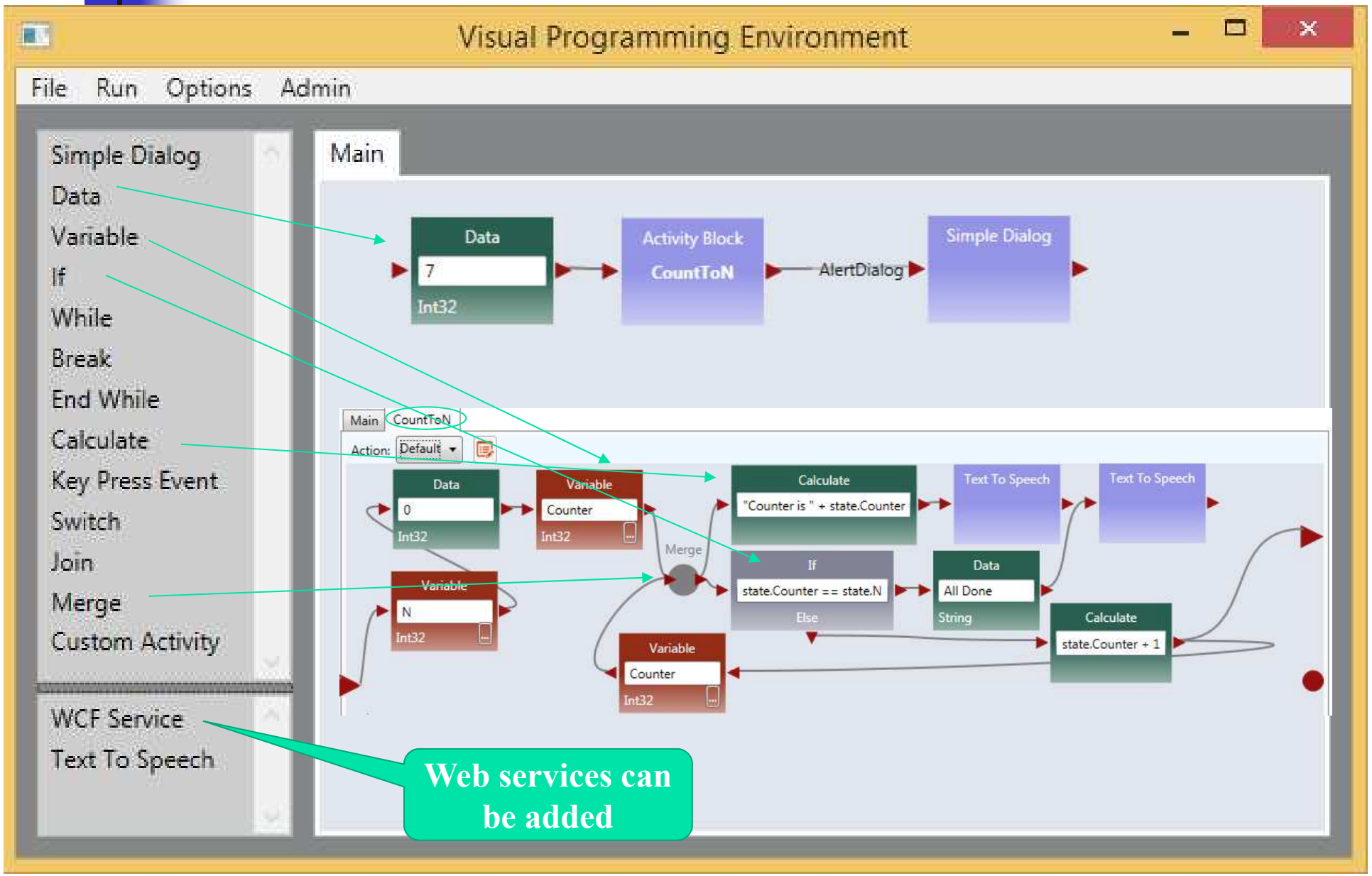
➢ Software

- A new software **ASU VIPLE**
- It does what Microsoft VPL does
- It is a RaaS unit and connect to Internet
- It connects our Intel-based robots





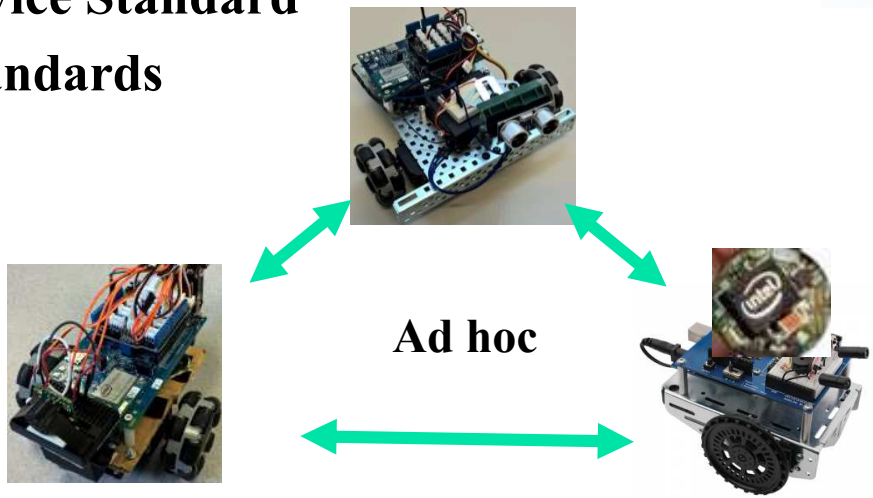
New ASU RaaS Software -- ASU VIPLE



ASU IoT and RaaS Version 4 Hardware

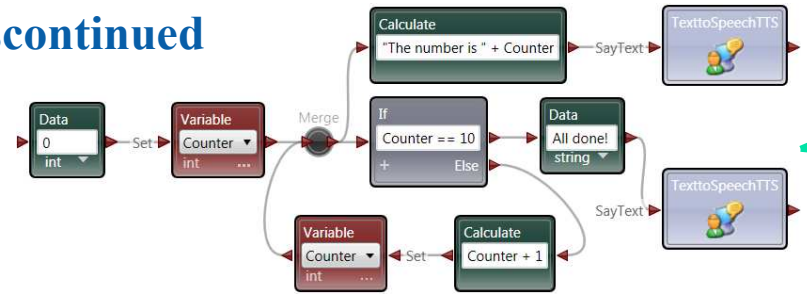
- **RaaS Hardware**
  - **Multiple robots collaboration**
  - **Ad hoc local communication**
  - **Infrastructure-based communication**
  - **Interoperability and standards**
    - **Web Service Standard**
    - **Other standards**

Infrastructure-based communication



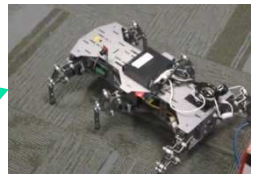
ASU IoT and RaaS Series in 2012 - 2015

Microsoft VPL
Discontinued



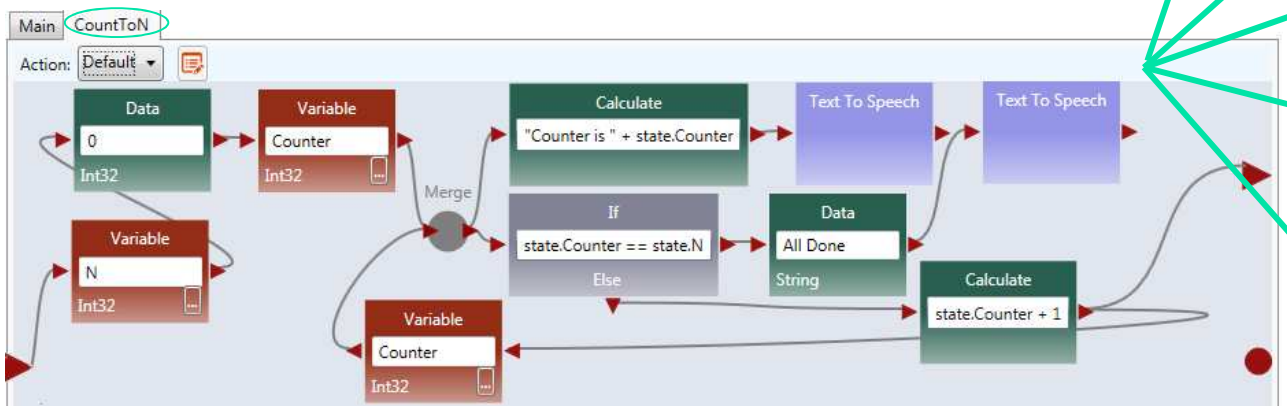
A Series of RaaS Robots

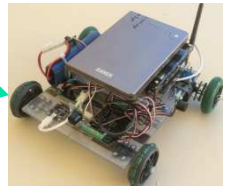
Largest



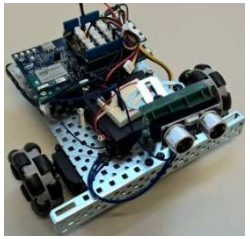
Large

ASU VPL, making use of Microsoft VPL Skills



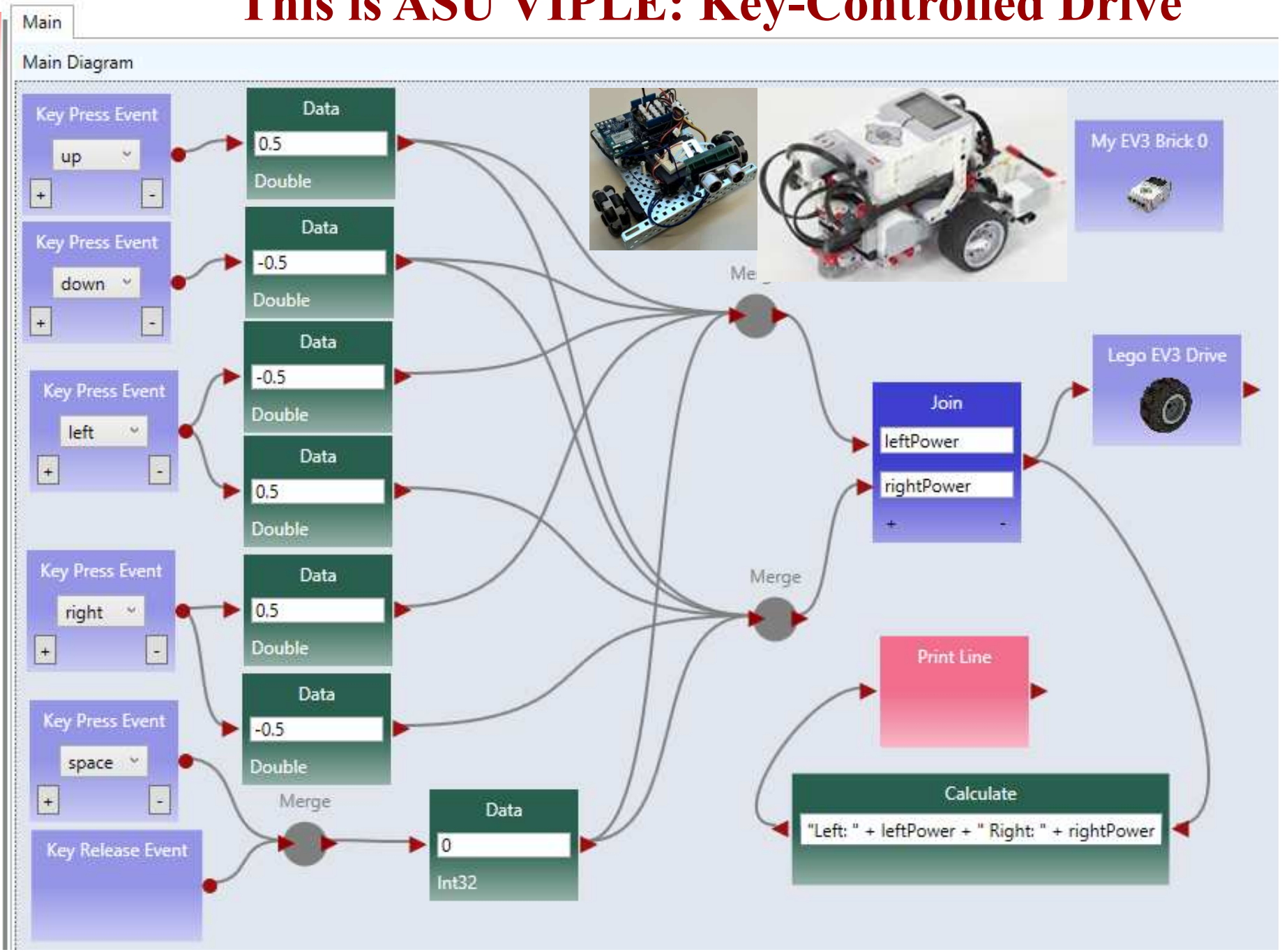


Small

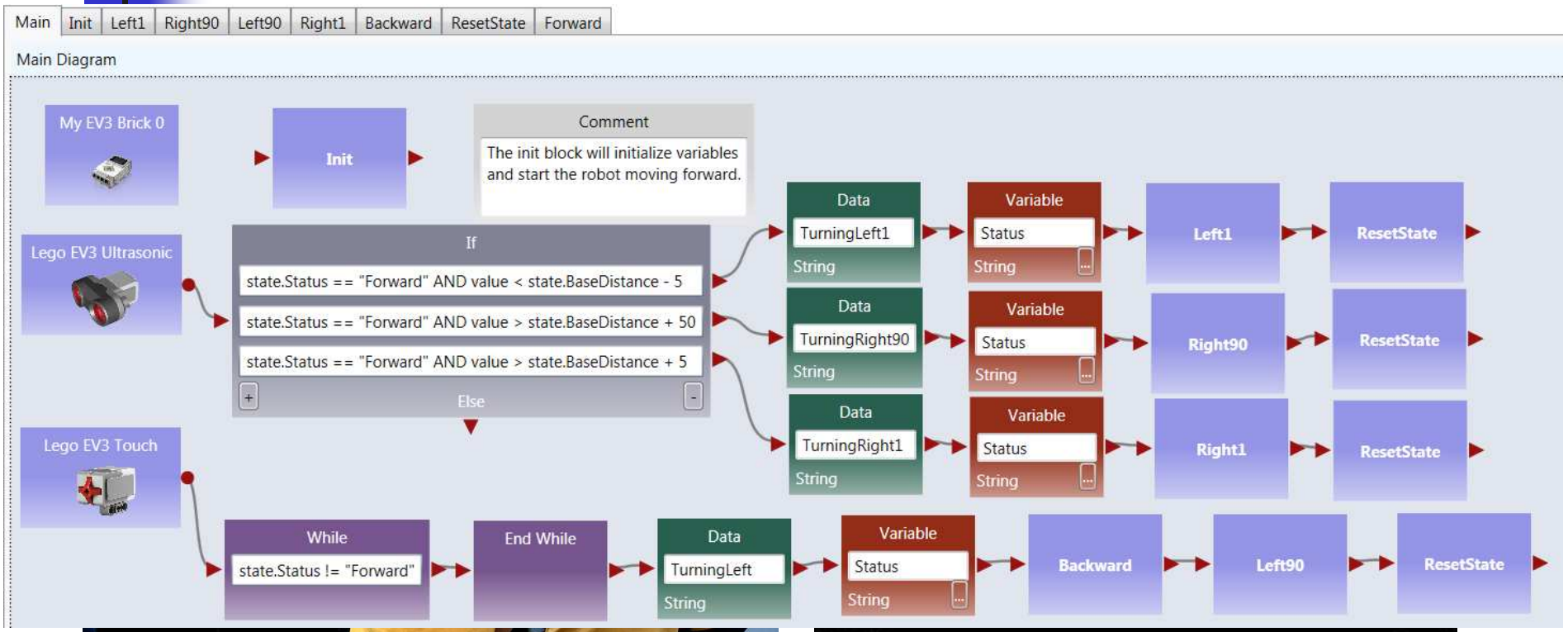


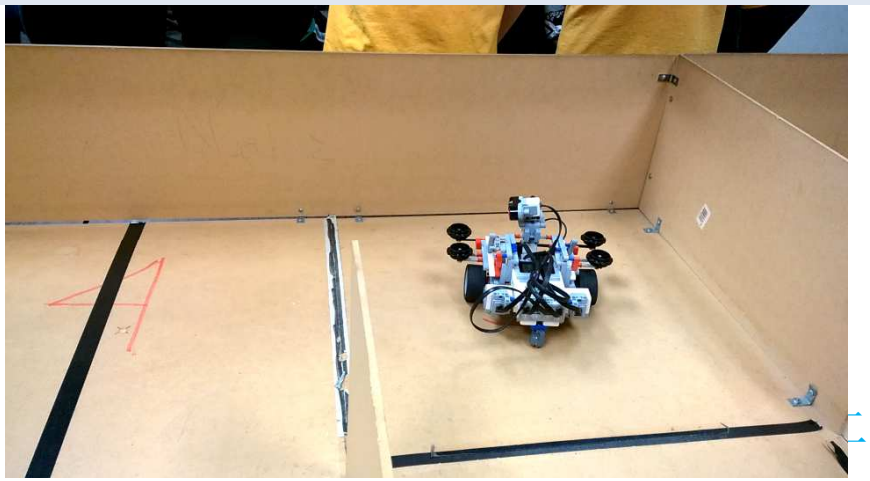
Smallest

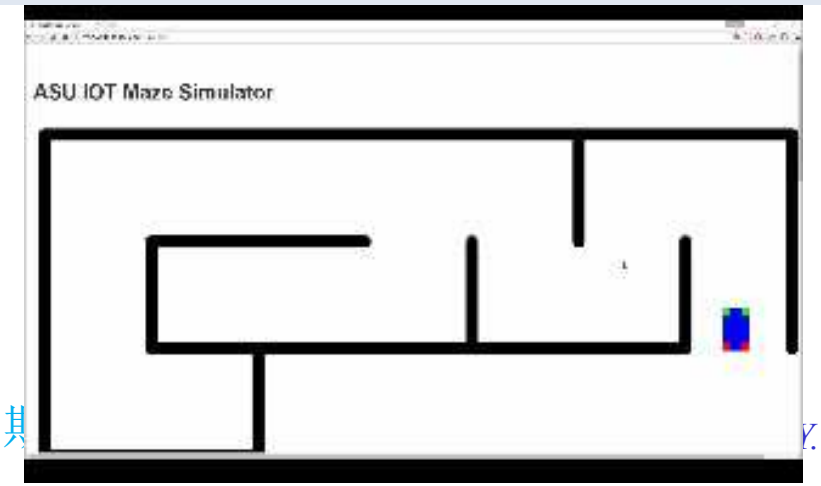
This is ASU VIPLE: Key-Controlled Drive



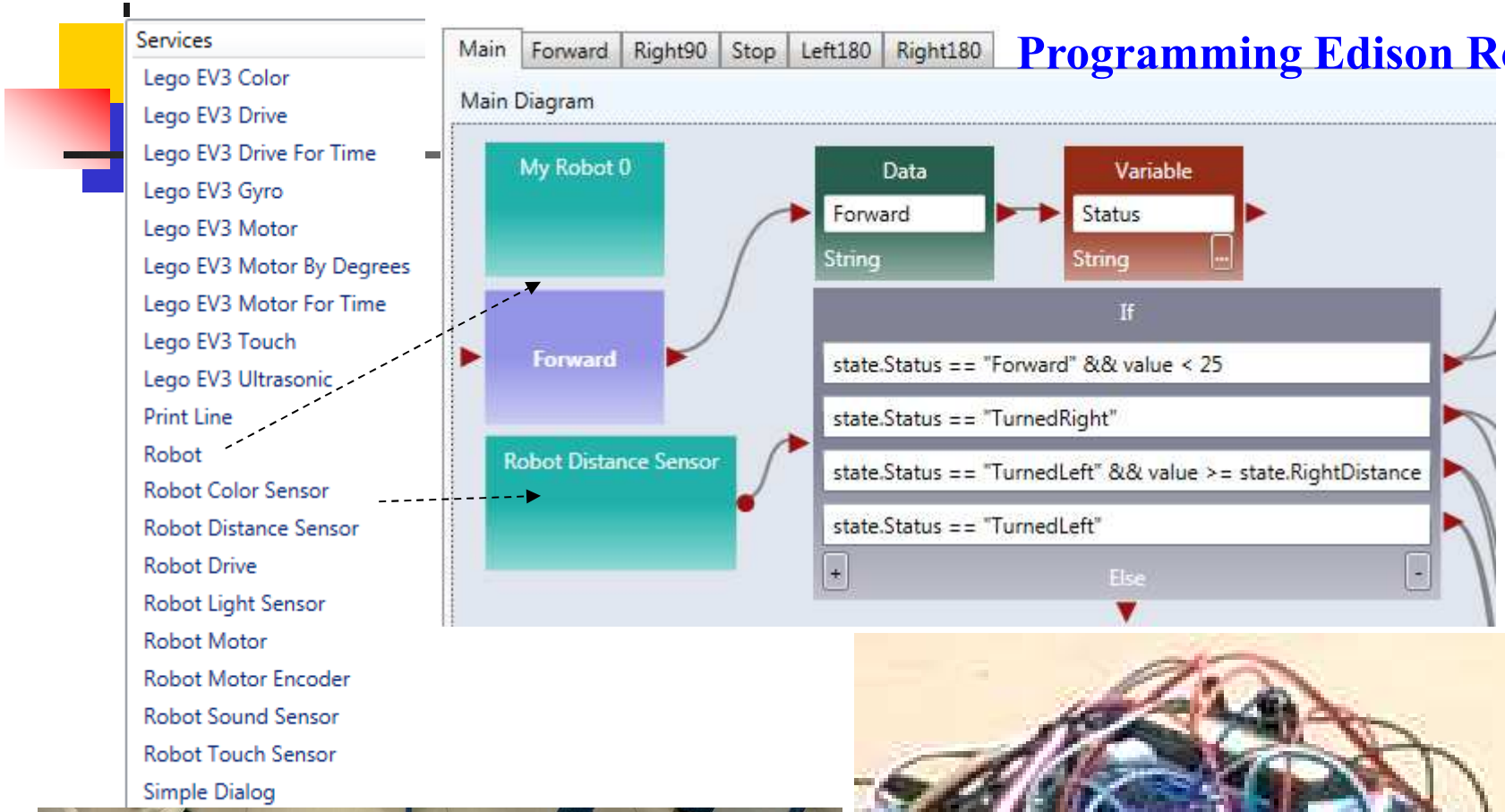
EV3 Robot and Web Simulated Robot

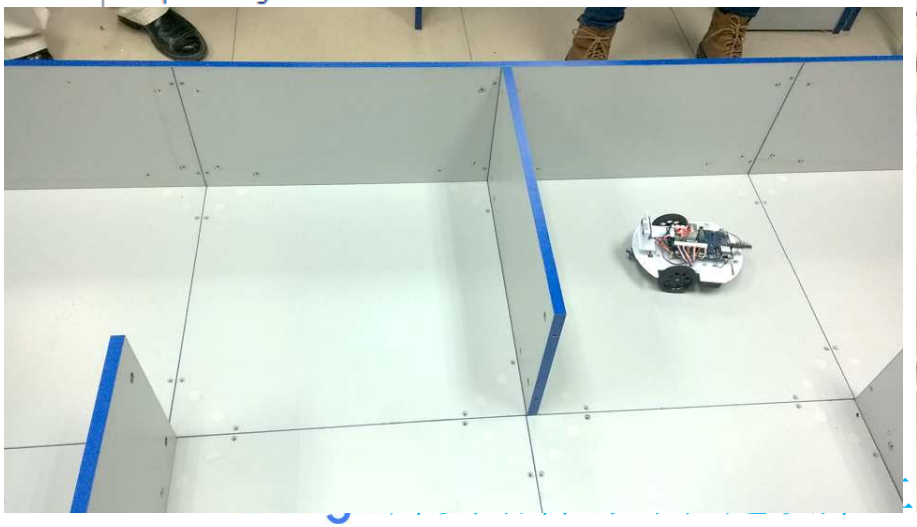


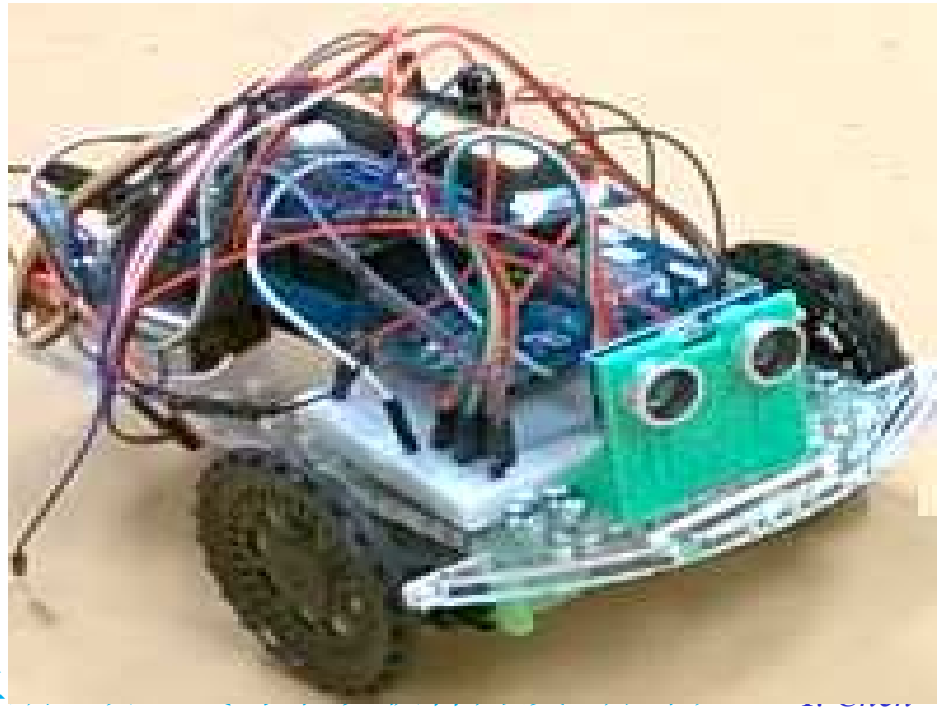




Programming Edison Robot



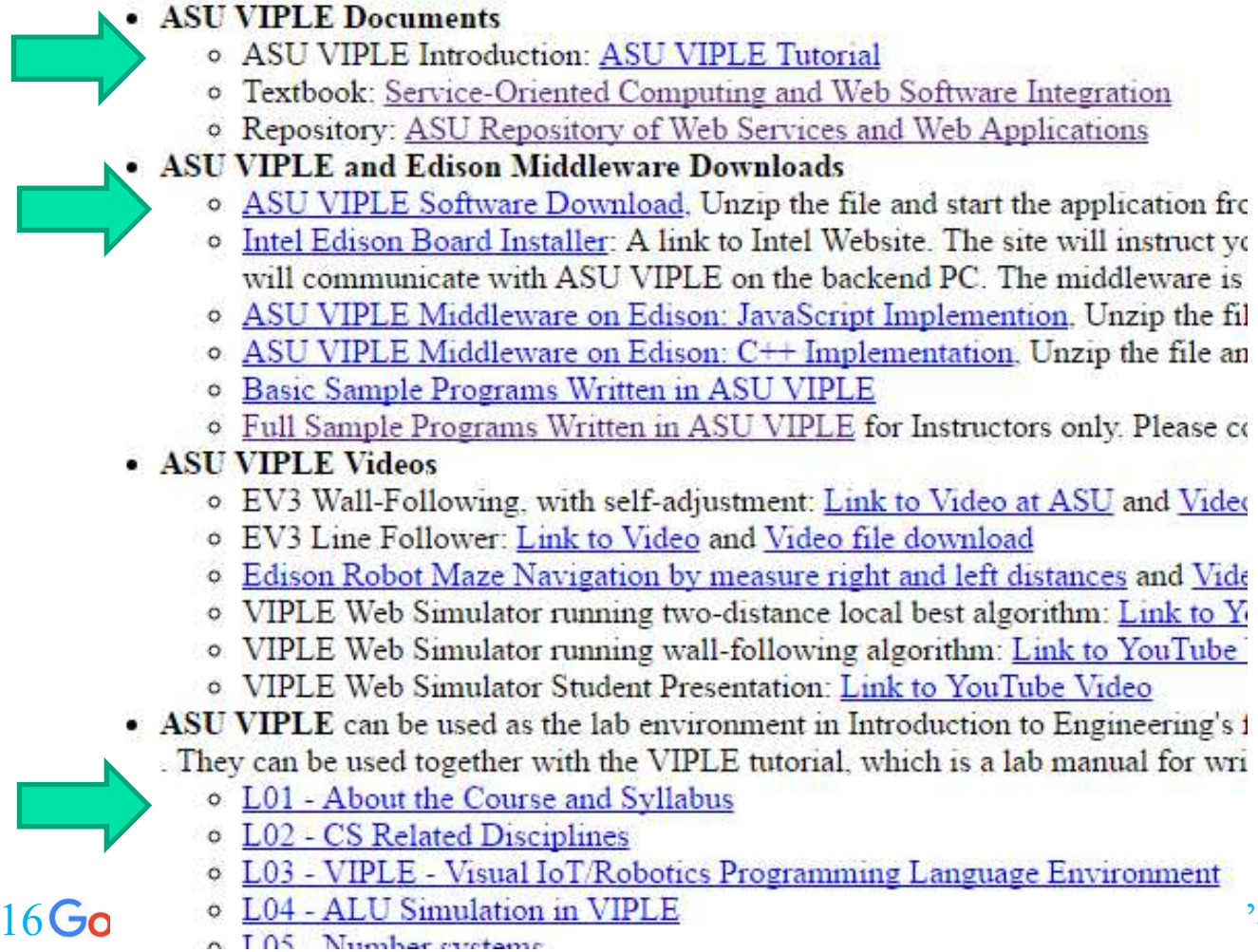




ASU VIPLE Download for Intel and EV3 Robots

- **Document and Software Download Link:**

  http://neptune.fulton.ad.asu.edu/VIPLE/



- **ASU VIPLE Documents**
  - ASU VIPLE Introduction: ASU VIPLE Tutorial
  - Textbook: Service-Oriented Computing and Web Software Integration
  - Repository: ASU Repository of Web Services and Web Applications
- **ASU VIPLE and Edison Middleware Downloads**
  - ASU VIPLE Software Download. Unzip the file and start the application fro
  - Intel Edison Board Installer: A link to Intel Website. The site will instruct yo
    will communicate with ASU VIPLE on the backend PC. The middleware is
  - ASU VIPLE Middleware on Edison: JavaScript Implemention. Unzip the fil
  - ASU VIPLE Middleware on Edison: C++ Implementation. Unzip the file an
  - Basic Sample Programs Written in ASU VIPLE
  - Full Sample Programs Written in ASU VIPLE for Instructors only. Please co
- **ASU VIPLE Videos**
  - EV3 Wall-Following, with self-adjustment: Link to Video at ASU and Video
  - EV3 Line Follower: Link to Video and Video file download
  - Edison Robot Maze Navigation by measure right and left distances and Vide
  - VIPLE Web Simulator running two-distance local best algorithm: Link to Yo
  - VIPLE Web Simulator running wall-following algorithm: Link to YouTube
  - VIPLE Web Simulator Student Presentation: Link to YouTube Video
- **ASU VIPLE** can be used as the lab environment in Introduction to Engineering's f
  . They can be used together with the VIPLE tutorial, which is a lab manual for wri
  - L01 - About the Course and Syllabus
  - L02 - CS Related Disciplines
  - L03 - VIPLE - Visual IoT/Robotics Programming Language Environment
  - L04 - ALU Simulation in VIPLE
  - L05 - Number systems

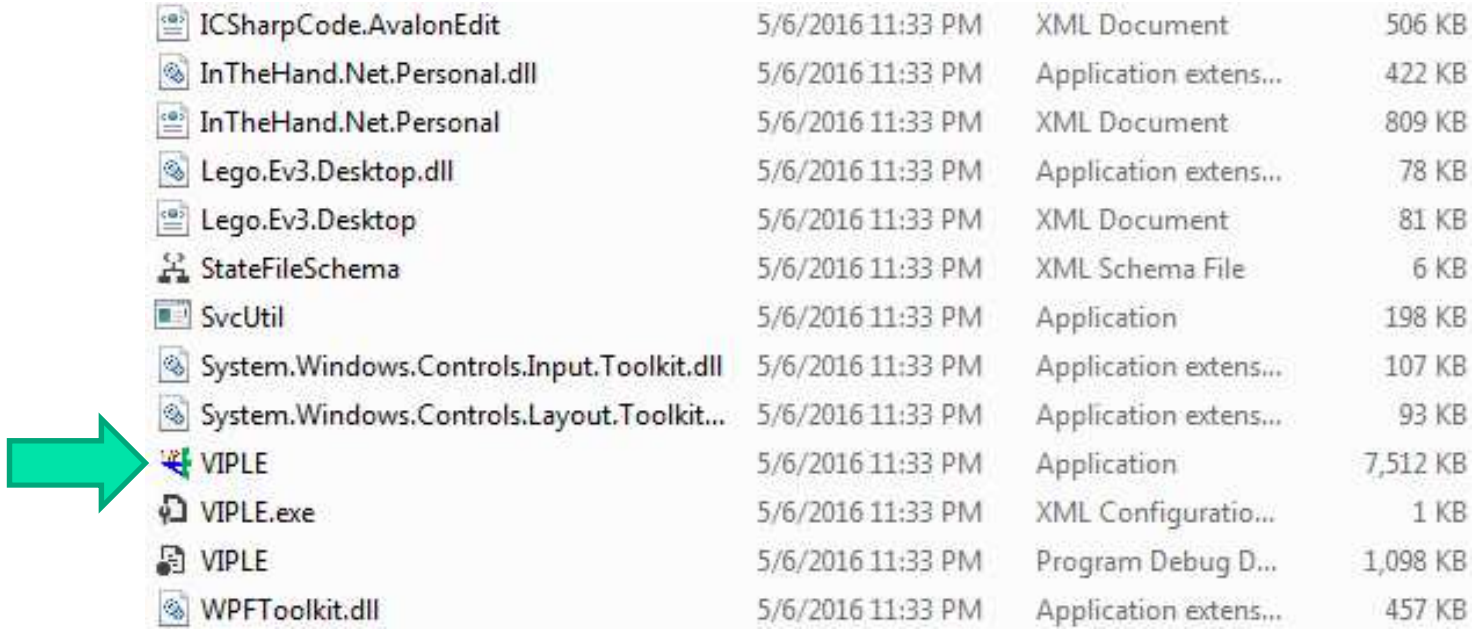
ASU VIPLE Download for Intel and EV3 Robots

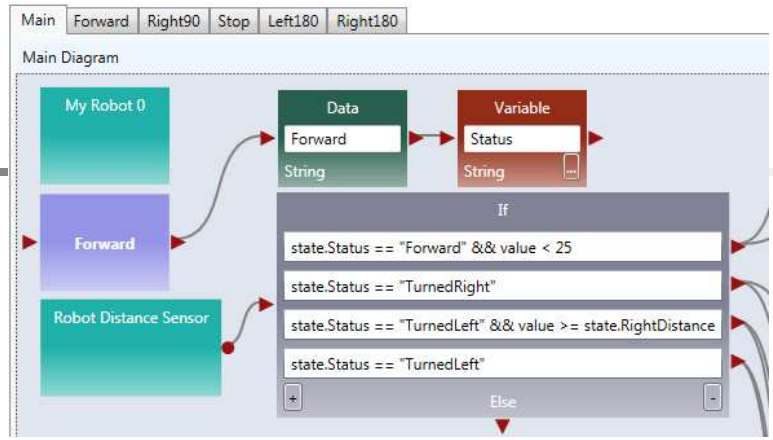
- **Document and Download Link:**

  http://neptune.fulton.ad.asu.edu/VIPLE/

- **After download ASU VIPLE,open the zip file:**

  1. Unzip and open the folder, start the application from the file: VIPLE

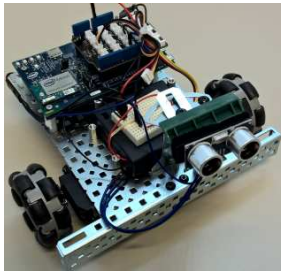


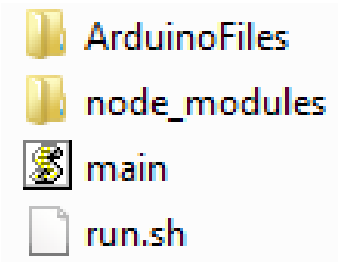


ASU VIPLE Program on PC

```
ROBOT OUTPUT

name: string (touch, distance, sound, light, color, motorEncoder)
id: int
value: For touch sensor, value will be an int (0 = not pressed and 1 = pressed).
       For other sensors, value will be a double

{"sensors": [{"name":"touch", "id":0, "value":0},{"name":"distance", "id":0, "value":12.8}]}


ROBOT INPUT

servoId: int

servoSpeed: double between -1 and 1
 - negative values represent a backwards motion

{"servos": [{"servoId":3, “servoSpeed”:0.5},{"servoId":5, “servoSpeed”:-0.5}]}
```

JSON Object is a Web service data standard for communication





Code Running on Edison