

# VIPLE

Visual IoT and Robotics,  
Programming Language Environment

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

可视化物联网/机器人程序语言环境

Yinong Chen

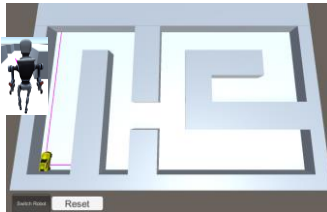
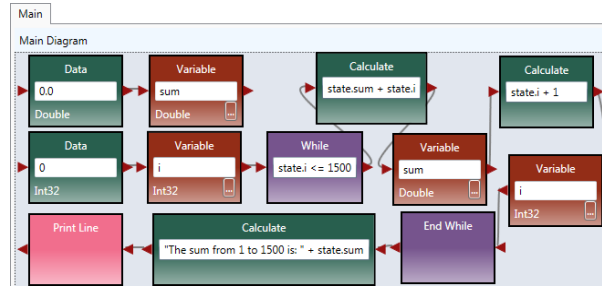
Arizona State University, U.S.A.

**IoT & Robotics Education  
Laboratory**

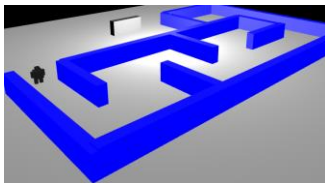
# VIPLE Supported Platforms

## 可视化物联网/机器人程序语言环境支持平台

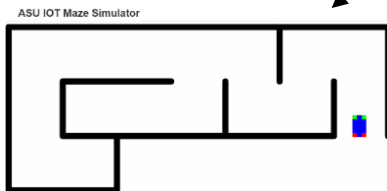
<http://neptune.fulton.ad.asu.edu/VIPLE/>



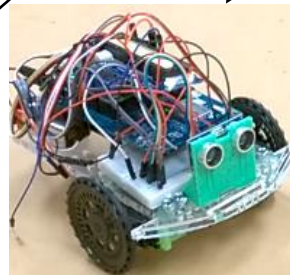
Unity Simulator



3D Web Simulator



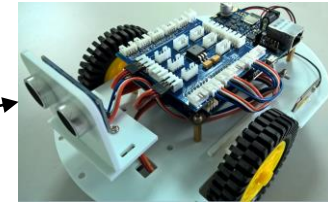
2D Web Simulator



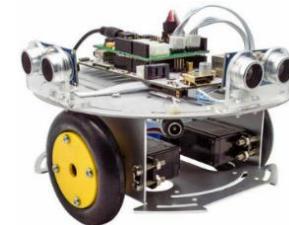
Intel Edison Robot



Minnow and Curie Robot



Intel Galileo Robot



ARM pcduino Robot



Lego EV3

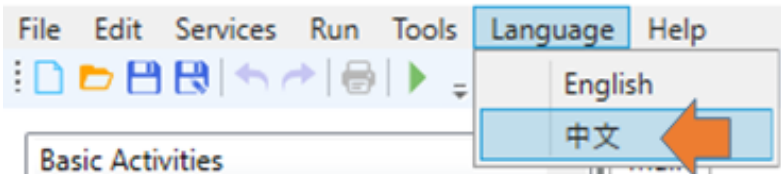


## VIPLE Features 特征

- General-purpose control flow programming  
通用控制程序设计
- Service-oriented computing, supporting RESTful and WSDL  
面向服务的计算，支持RESTful和WSDL标准
- Parallel / multithreading programming, with underlying threads safety (synchronization)  
并行/多线程编程，底层线程安全（同步）
- Event-driven programming, with built-in and custom events  
事件驱动编程，内置和定制事件
- Workflow and visual programming 工作流与可视化编程
- IoT and Robotics programming 物联网与机器人编程

# VIPLE Basic Activity 基本活动

Unnamed - VIPLE



基本活动

自定义活动

变量

计算

数据

与并

或并

如果

开关

条件循环

退出

结束循环

注释

→ **Activity:** 用于创建一新模块。

→ **Variable:** 支持基本的类型(Int32, Double, String, Boolean, etc.)。

→ **Calculate:** 计算程序语言的一般表达式的值。

→ **Data:** 导入程序语言中的常数值。

→ **Join:** 与并将等待所有线程或输入数据的到来才能进入下一步。所有数据可以通过成员函数"."来访问。

→ **Merge:** 或并只需等待第一个线程或输入数据的到来就能进入下一步。

→ **If:** 条件选择。允许多个条件顺序选择。

→ **Switch:** 根据输入值与所给值的匹配决定下一步的路径。

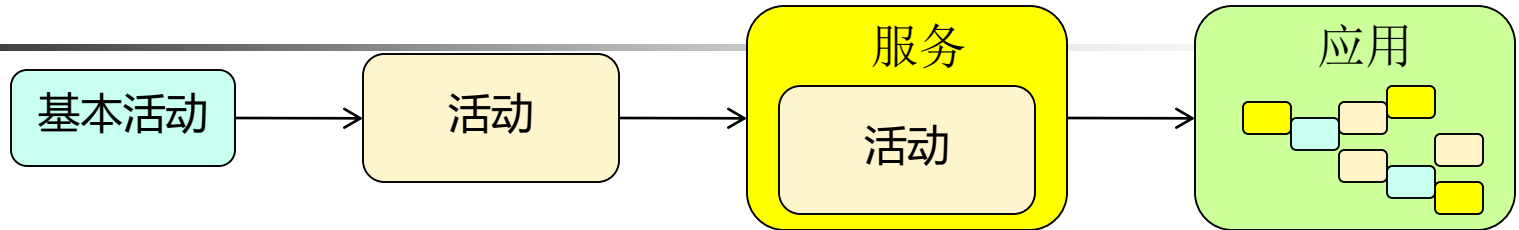
→ **While:** 循环开始点;

→ **Break:** 退出循环

→ **End While:** 循环的结束点。自动跳到循环开始点。

# Basic Activities 基本活动 (继续)

5



## 基本活动

自定义活动

变量

计算

数据

与并

或并

如果

开关

条件循环

退出

结束循环

注释

- ❖ Basic Activity and Activity are building blocks of a diagram or flowchart. 基本活动和活动是流程图的构建块。
- ❖ Activity can be wrapped into a service 活动可以包装成一个服务
- ❖ Data transfer between activities are through 活动之间的数据传输通过
  - global variables 全局变量
  - parameter passing 参数传递

■ Construct a composite activity (block or module) 构建复合活动 (块或模块)

# VIPLE is Based on $\pi$ -Calculus $\pi$ -代数

6

$\pi$ -calculus construct	$\pi$ -calculus construct in computational thinking diagram	VIPLE activities matching the diagram	Underlying CS concepts / Computational Thinking
<b>Sequential Process</b> $S ::= new\ a(A B).0$ $A ::= \tau_A.\bar{a}(x).0$ $B ::= a(x).\tau_B.\bar{b}(y).0$		Activity Block Variable Calculate Data	Procedural process
<b>Split</b> $n \geq 2$ $S ::= new\ a1a2 \dots an(A B1 B2  \dots  Bn)$ $A ::= \tau_A.\bar{a1}(x).\bar{a2}(x).\dots.\bar{an}(x)$ $Bn ::= an(x).\tau_{Bn}.\bar{bn}(y)$		All VIPLE activities support split or fan-out 	Creating parallel process
<b>Join</b> $n \geq 1$ $S ::= new\ a1a2 \dots an(A1 A2  \dots  An B)$ $An ::= \tau_{An}.\bar{an}(x)$ $B ::= a1(x1).a2(x2).\dots.an(xn).\bar{b}(y)$			Wait for all parallel threads to join, e.g., parameter passing;
<b>Merge</b> $n \geq 1$ $S ::= (A1 A2  \dots  An B)$ $An ::= \tau_{An}.\bar{a}(x)B ::= !a(x).\bar{b}(y)$			Pick the first result that arrives among the parallel threads
<b>Choice (if)</b> $n \geq 2$ $S ::= new\ a1a2 \dots an(A B1 B2  \dots  Bn)$ $A ::= if\ bexpr1\ then\ \bar{a1}(x)\ else$ $\quad if\ bexpr2\ then\ \bar{a2}(x)$ $\quad el\ \square\ if \dots\ else\ if\ bexpr(-1)$ $\quad\quad then\ a(n-1)(x)$ $\quad\quad else\ \bar{an}(x) Bn ::= an(x).\tau_{Bn}.\bar{bn}(y)$			Conditional statement; nested statement; Boolean operators
<b>Loop (While ... End While)</b> $S ::= A B C$ $A ::= d(y).$ $\quad if\ bexpr\ then\ \bar{a}(x)\ else\ \bar{c}(y)$ $B ::= a(x).\tau_B.\bar{b}(y)C ::= b(y).\bar{d}(y)$			Iteration statement; conditional Statement;

# VIPLE Services 服务列表

## 通用计算和 事件服务

源代码活动

自定义事件

按键事件

释键事件

行打印

随机

RESTful服务

简单的对话

文字转语音

定时器

## 通用机器人 服务

机器人主机

机器人彩色传感器

机器人距离传感器

机器人驱动器

机器人完整协调驱动

机器人光传感器

机器人运动

机器人电机

机器人电机编码器

机器人声音传感器

机器人触觉传感器

机器人+移动-动力控制

机器人+转动-角度控制

## 乐高机器人 服务

乐高主机EV3

乐高EV3彩色

乐高EV3驱动器

乐高EV3驱动器-时间控制

乐高EV3陀螺

乐高EV3舵机

乐高EV3电机-转角控制

LEGO EV3电机-时间控制

乐高EV3按下触摸

乐高EV3释放触摸

乐高EV3超声

# VIPLE Programming: Output 输出

## 基本活动

### 基本活动

自定义活动

变量

计算

数据

与并

或并

如果

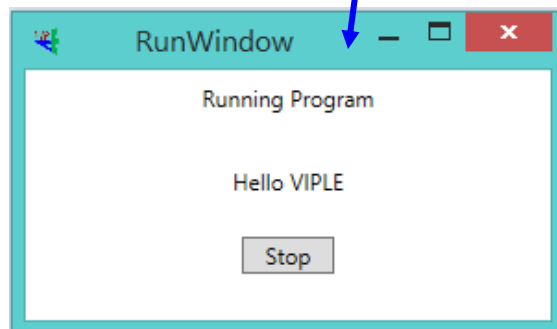
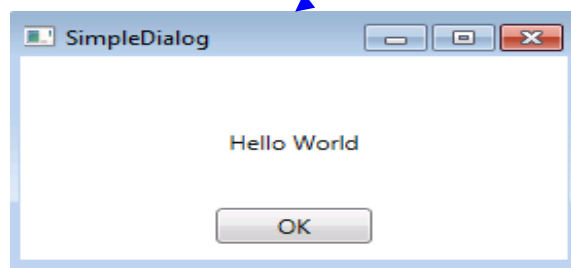
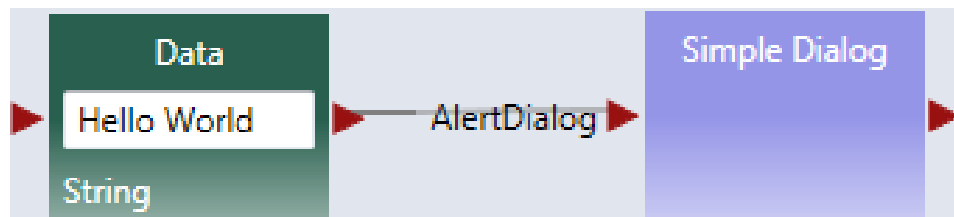
开关

条件循环

退出

结束循环

注释



## 通用计算和事件服务

源代码活动

自定义事件

按键事件

释键事件

行打印

随机

RESTful服务

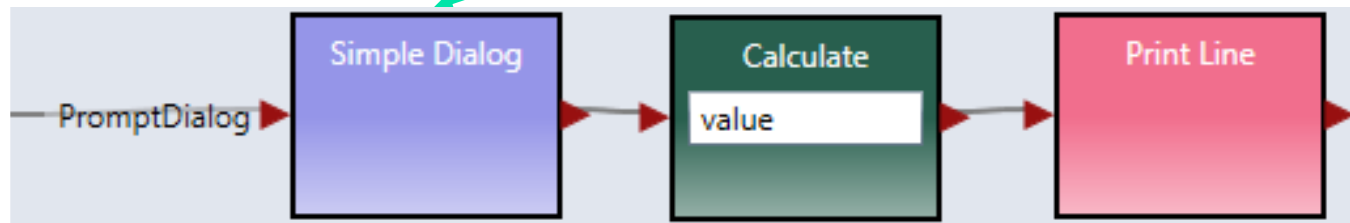
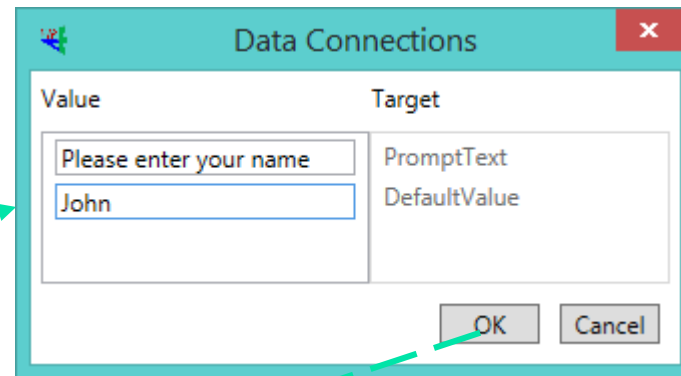
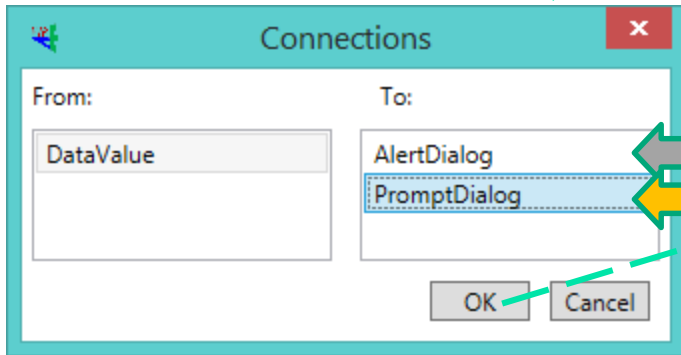
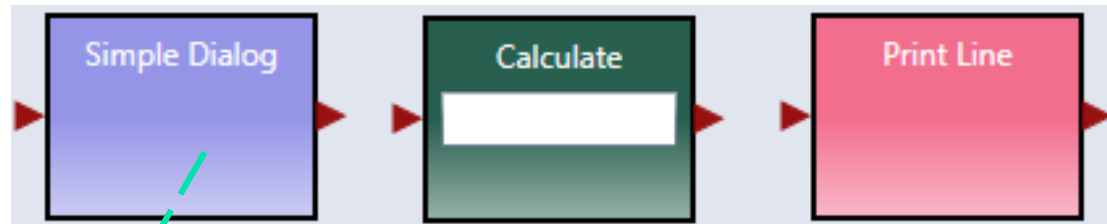
简单的对话

文字转语音

定时器



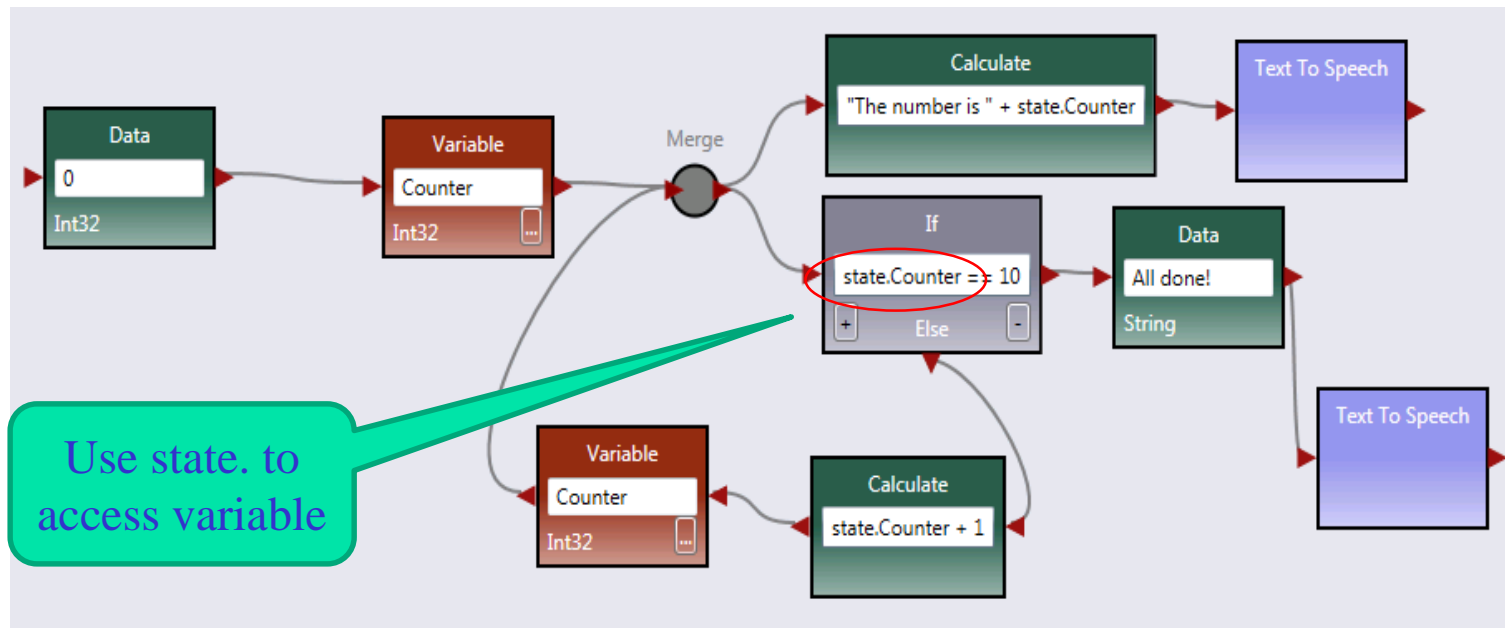
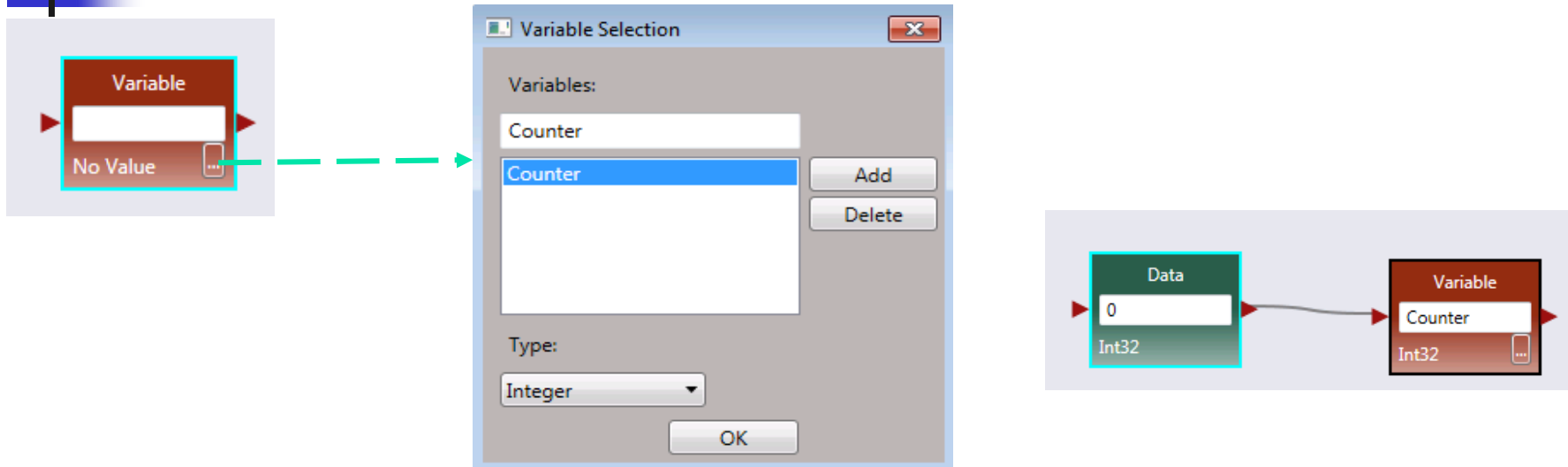
# VIPLE Programming: Input 输入



# VIPLE Programming: Variable and Loop

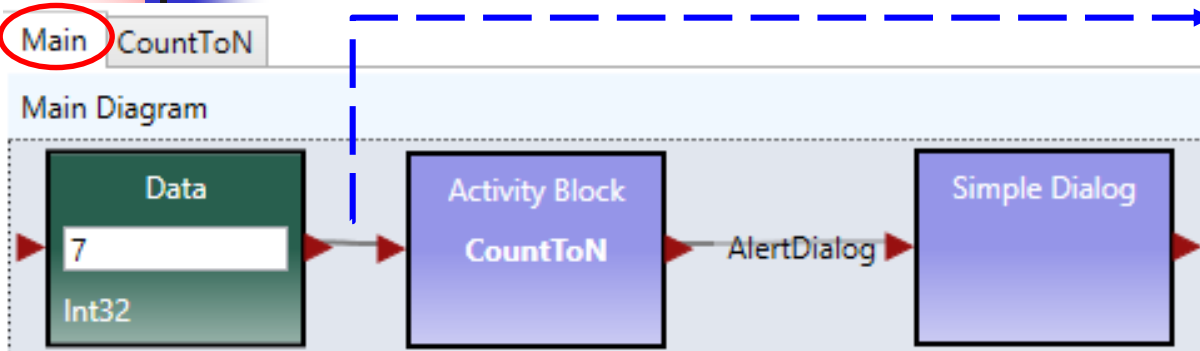
10

## VIPLE编程：变量和循环



# Activity and Parameter Passing 活动和参数

11



Data Connections

Value	Target
value	Limit

OK Cancel

Input Variable Definition

Input Variables:

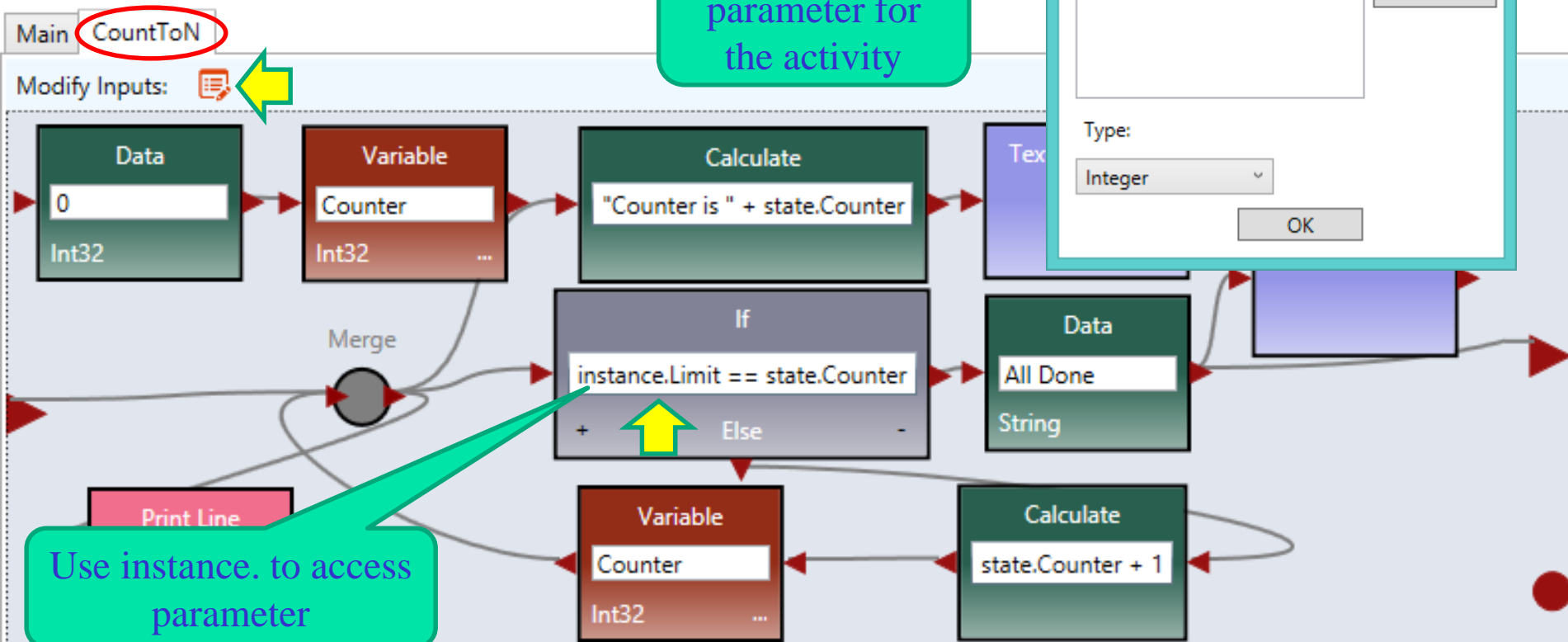
- Limit

Add Delete

Type: Integer

OK

Add a parameter for the activity



Use instance. to access parameter

# ASU VIPLE Download Site 下载网站


Download Link: <http://neptune.fulton.ad.asu.edu/VIPLE/>



## ASU VIPLE Tutorials and 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 Software and Edison Middleware Free Downloads

-  [ASU VIPLE Standard Edition Installer](#).
- [Microsoft .Net Framework 4.5 Download](#). To run VIPLE, .Net Framework 4.5 or later is required.
- [Intel Edison Board Installer](#): A link to Intel Website. The site will instruct you to install Linux OS Middleware to run. ASU VIPLE middleware will communicate with ASU VIPLE on the backend sensors and motors for your robot.
- [ASU VIPLE Middleware on Edison: JavaScript Implementation](#), Unzip the file and copy all the folders
- [ASU VIPLE Middleware on Edison: C++ Implementation](#), Unzip the file and copy all the folders :
- [Basic Sample Programs Written in ASU VIPLE](#)
- [Full Sample Programs Written in ASU VIPLE](#) for Instructors only. Please contact Dr. Chen at yinc

## ASU VIPLE Videos

- EV3 Wall-Following, with self-adjustment: [Link to Video at ASU](#) and [Video file download](#) and [on](#)
- EV3 Line Follower: [Link to Video](#) and [Video file download](#)
- [Edison Robot Maze Navigation by measure right and left distances](#) and [Video file download](#) and [o](#)