FSE598 前沿计算技术

模块 1　计算思维
单元 2　工作流与可视化编程
第 5 讲　迷宫和自动驾驶

本讲座的英文版内容基于教材：
Y. Chen, G. De Luca Service-Oriented Computing and System Integration: Software, IoT, Big Data, and AI as Services, 8th edition, Kendall Hunt Publishing, 2022.
https://www.public.asu.edu/~ychen10/book/socsi.html

本讲提要

学习

- 自动迷宫导航
- 迷宫导航算法
- 迷宫导航编程
- 自动驾驶传感器/控制器
- 自动驾驶编程

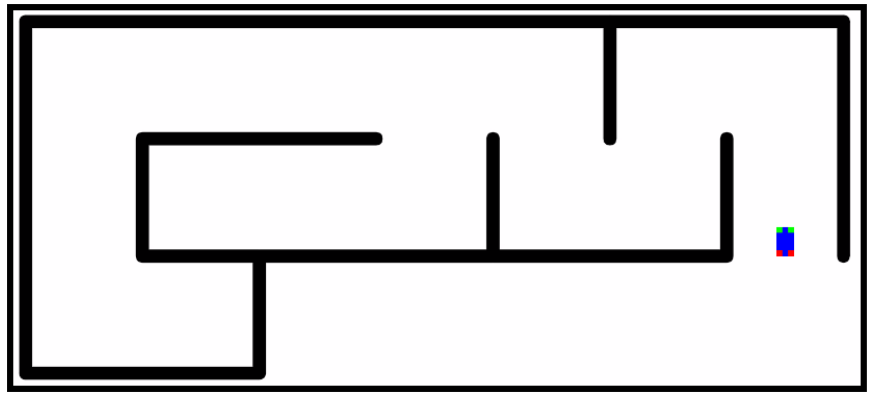
迷宫导航

一个人或一个**自主机器人**必须在迷宫中导航。
导航问题是：从内部解决迷宫。

导航算法逐渐接收输入，而不是一次性接收。它必须根据这个部分输入做出决定。

ASU Web 2D 模拟器：绕右墙

http://venus.sod.asu.edu/VIPLE/Web2DSimulator/indexcn.html



Add a New Line | Remove a Line

Default: Forward

1. If Right sensor > 100 pixels

Then: Delayed Turn Right by 90 pixels

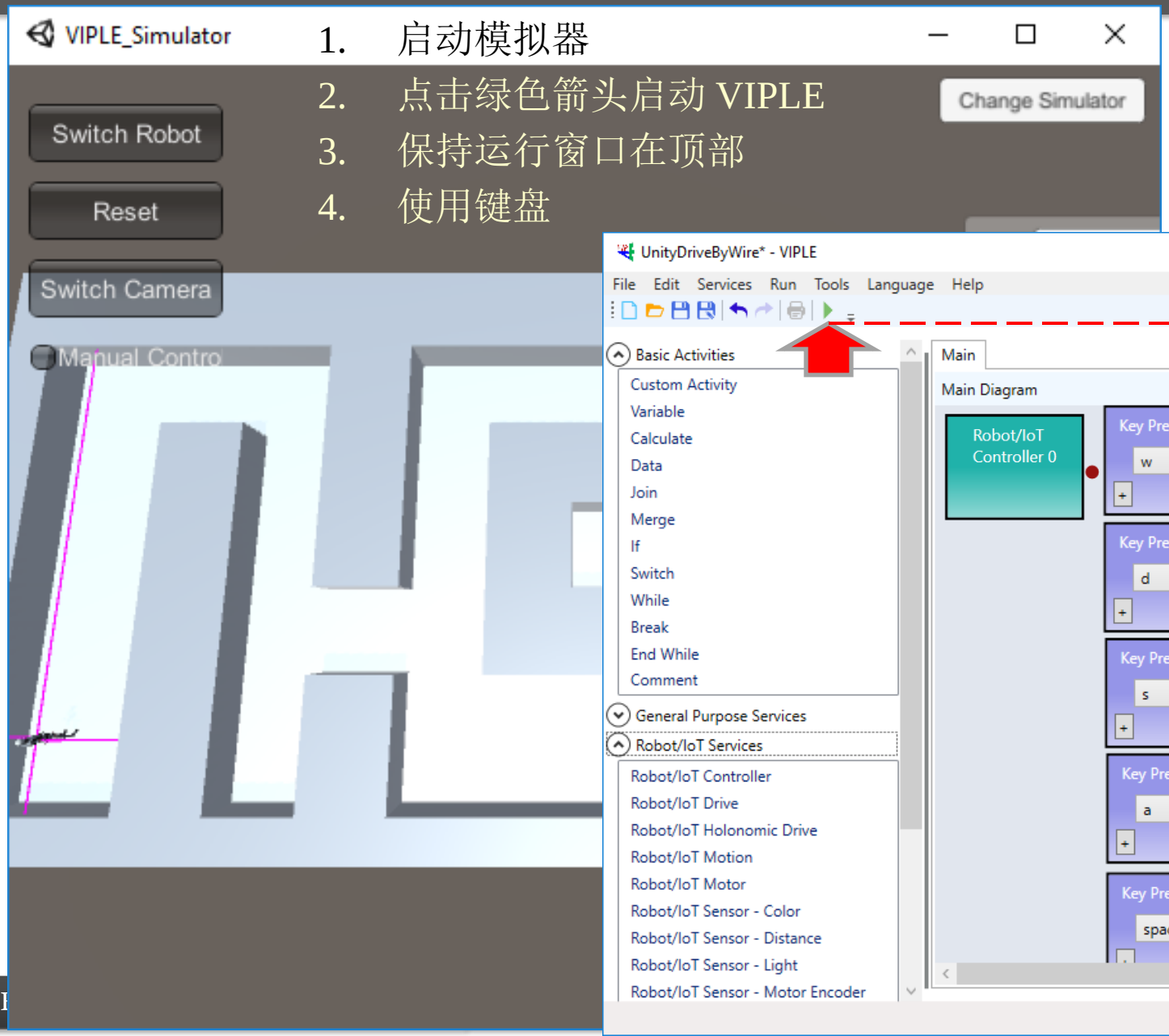
2. Else if Forward sensor <= 50 pixels

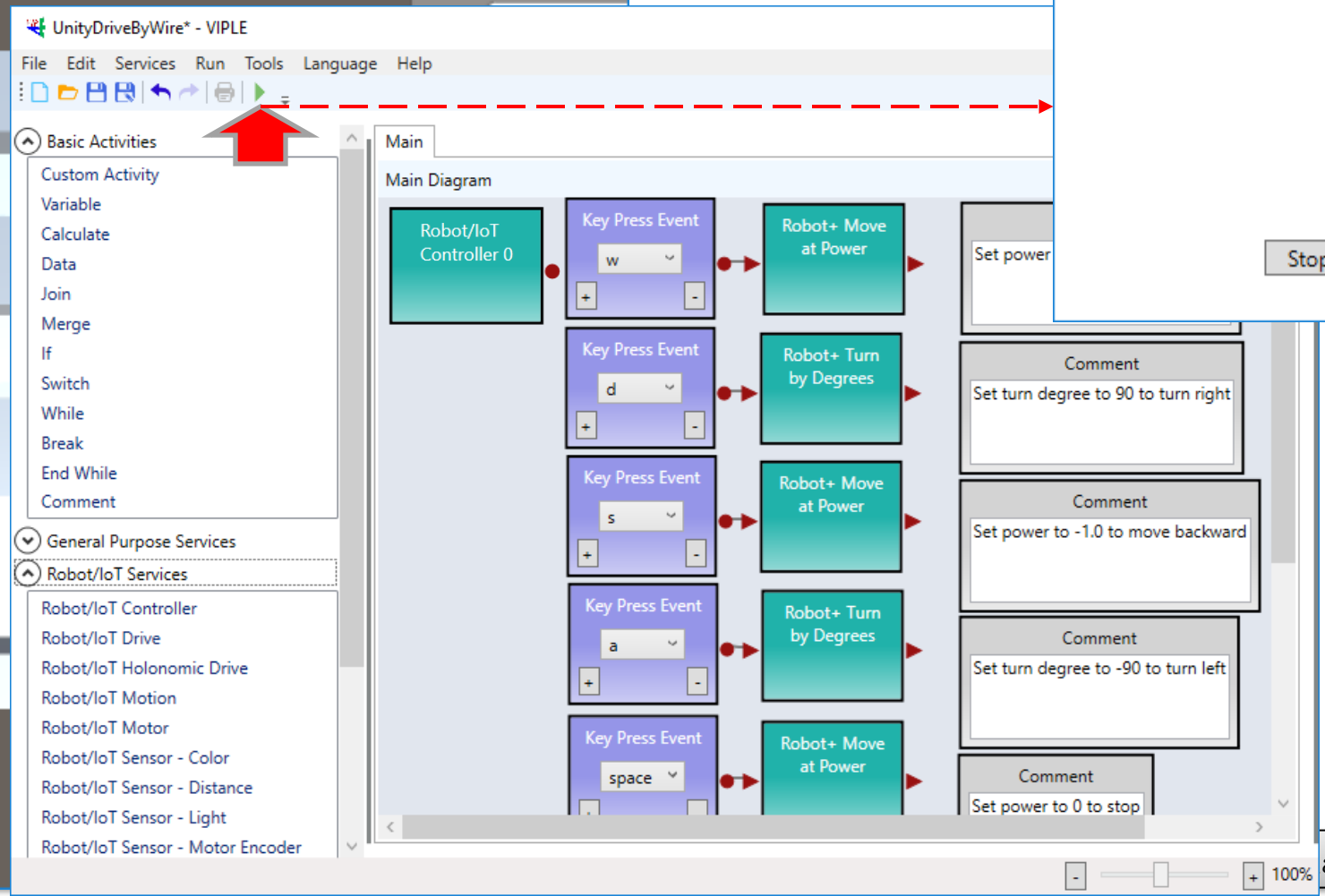
Then: Turn Left

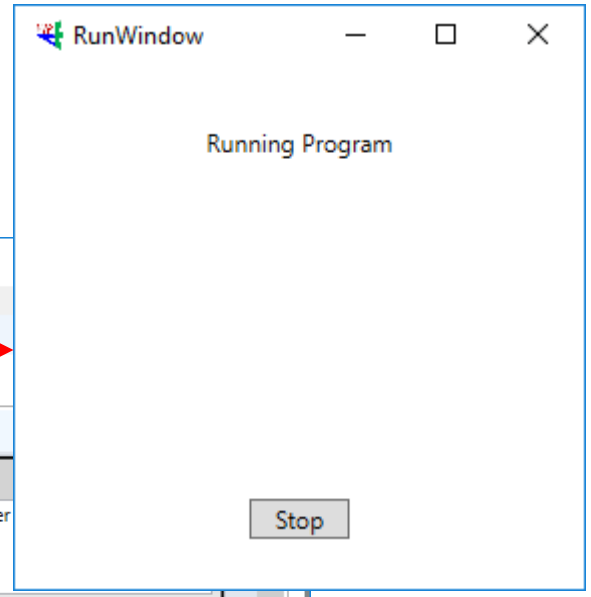
Run

启动和运行 Unity 模拟器

1. 启动模拟器
2. 点击绿色箭头启动 VIPLE
3. 保持运行窗口在顶部
4. 使用键盘

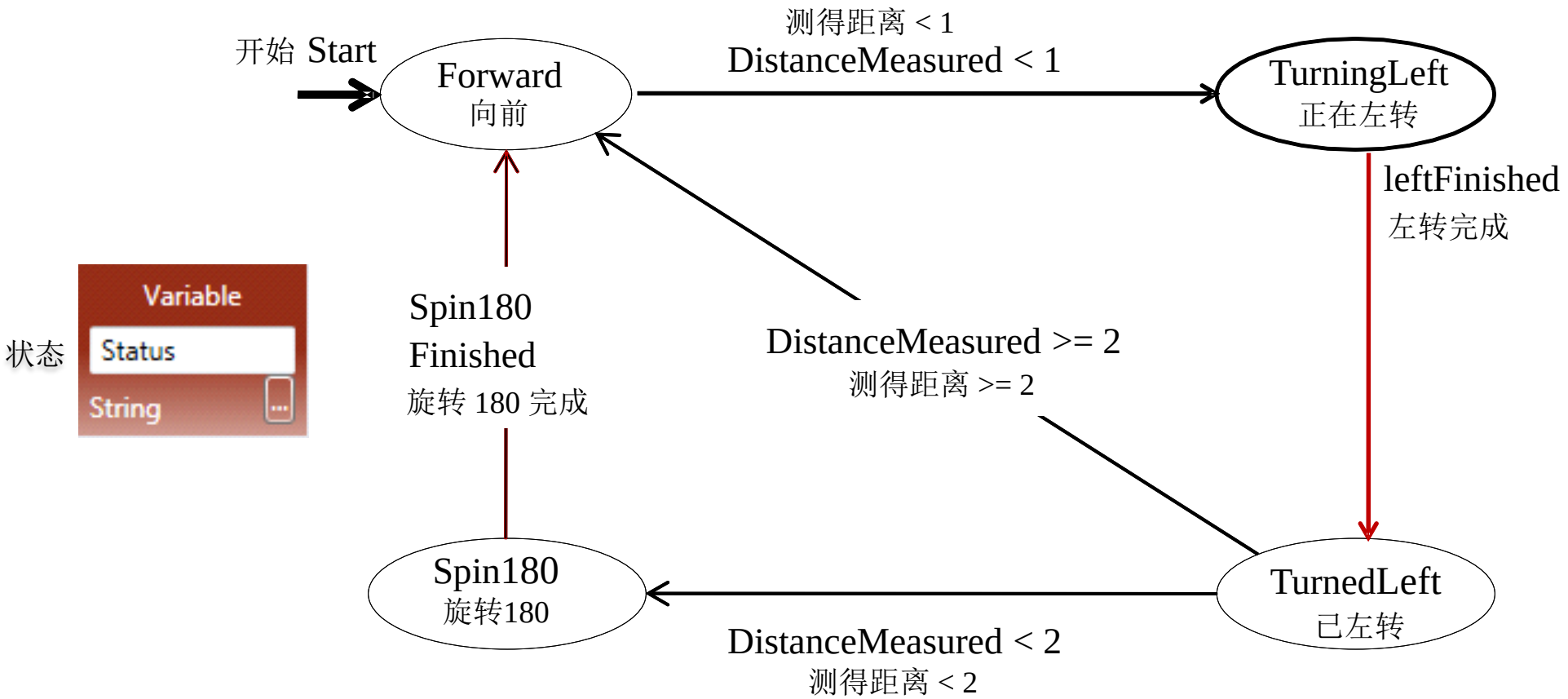






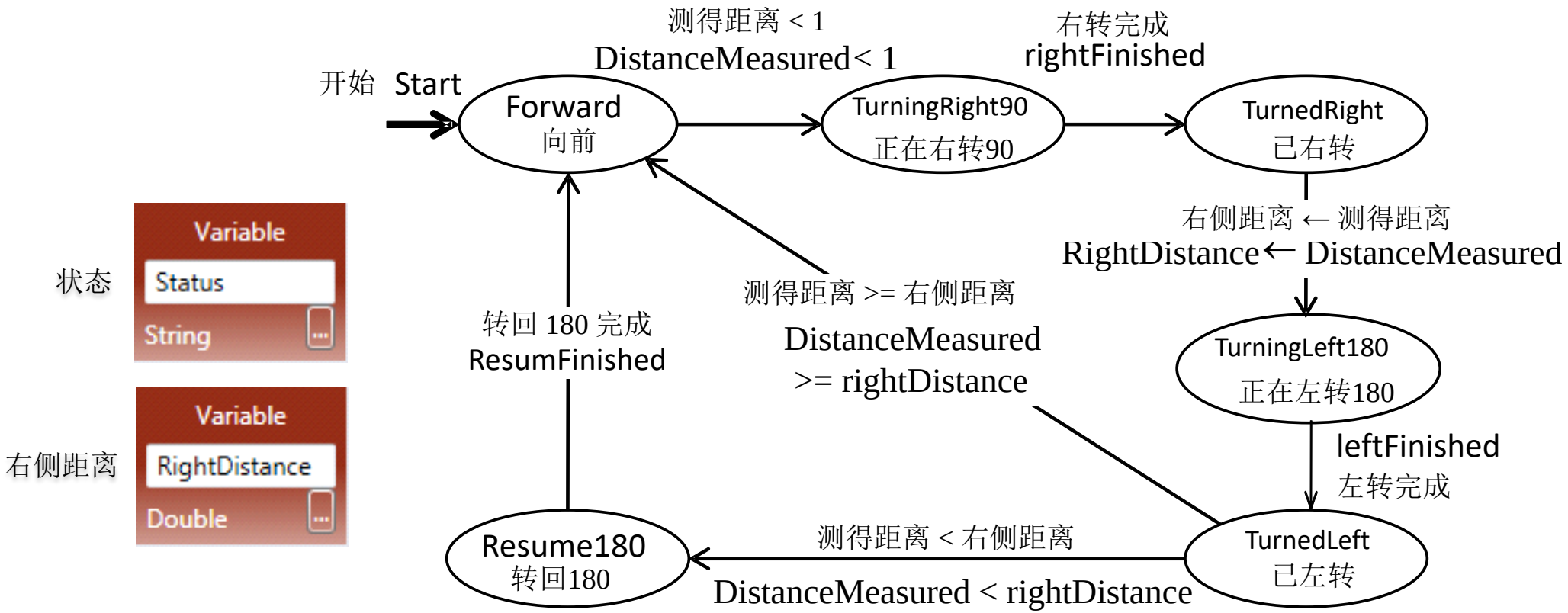
迷宫导航算法

- 贪心算法：采用第一个可行的解决方案



启发式迷宫导航算法

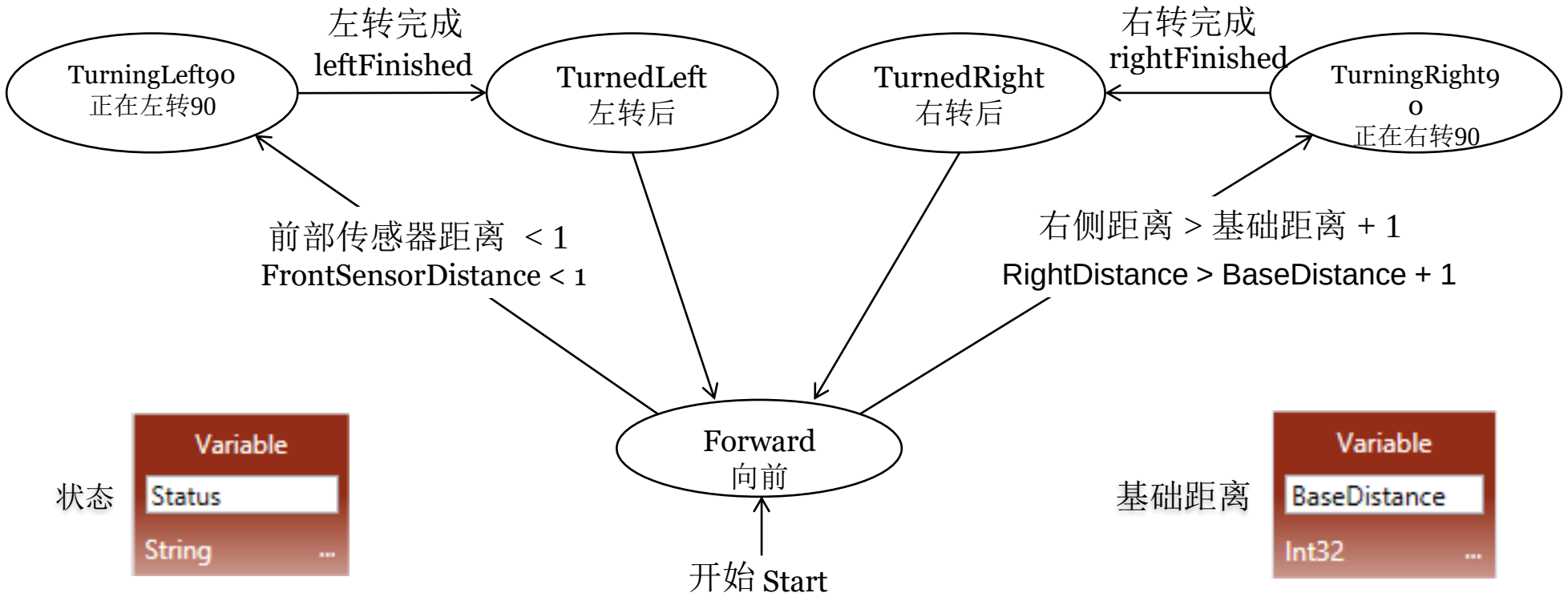
- 局部最佳：测量两个距离并选择距离较长的方向



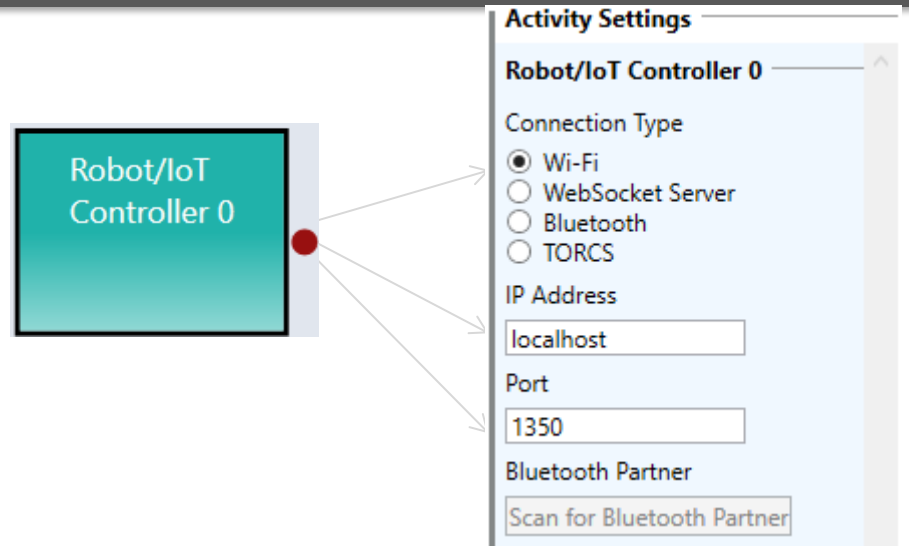
简单的绕墙迷宫导航算法

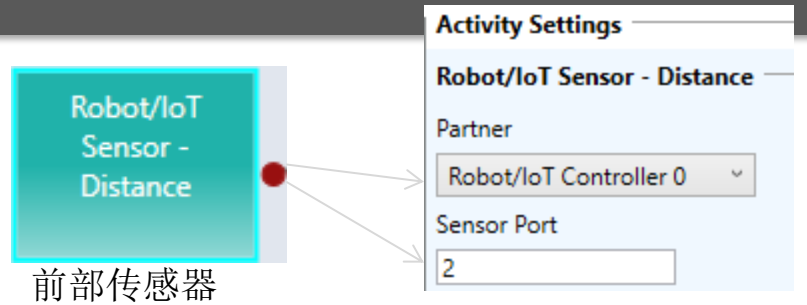
绕右墙：

- 前移；
- 如果右侧有开口则右转；
- 如果前方没路则左转。
- 它假设前进和转弯是准确的。

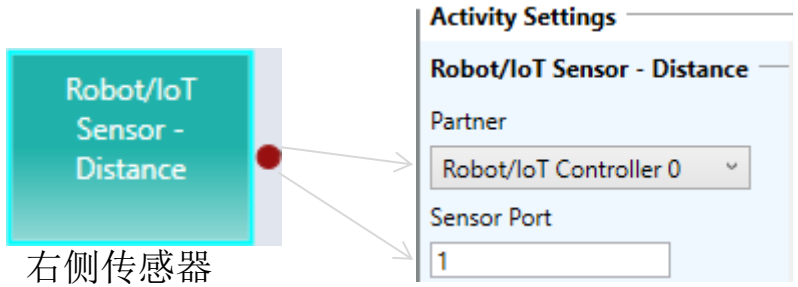


迷宫导航的 Unity 模拟器配置

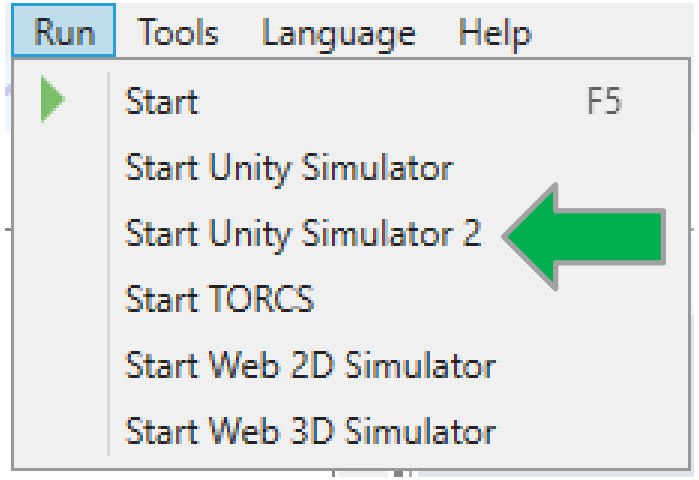




前部传感器



右侧传感器

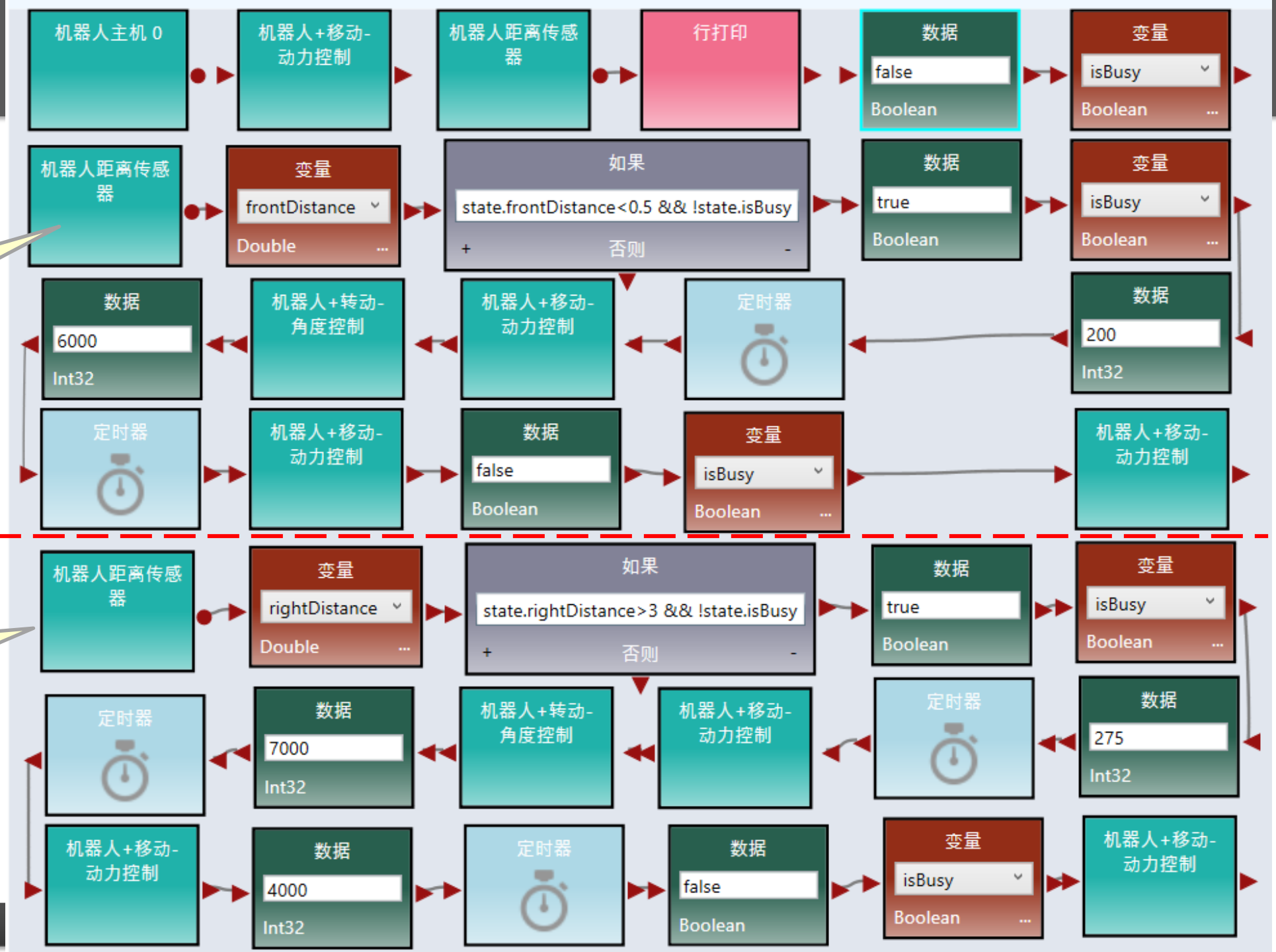


模拟机器人在前面和两侧都有距离传感器。

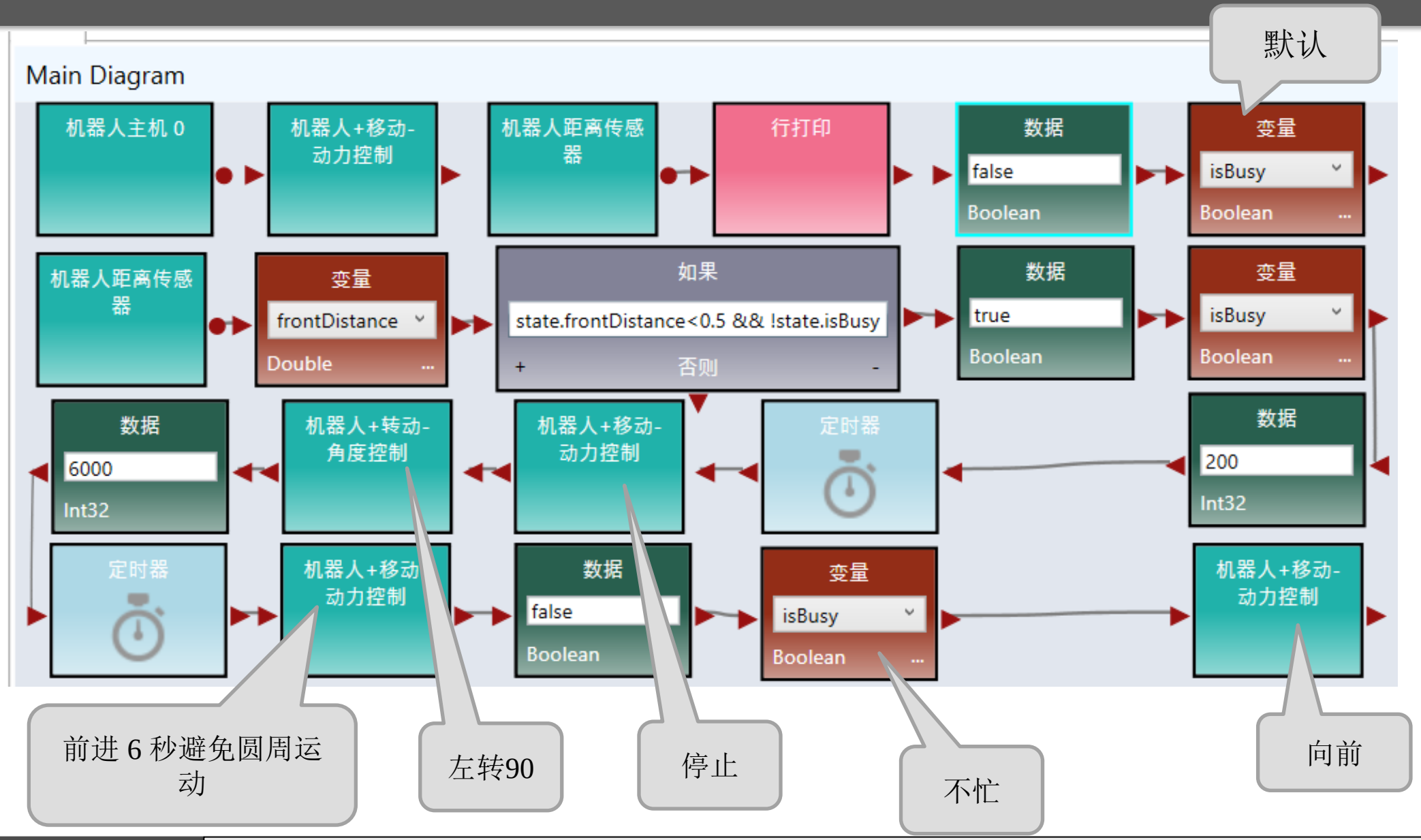
Unity 模拟器中的绕墙

处理前方距离传感器

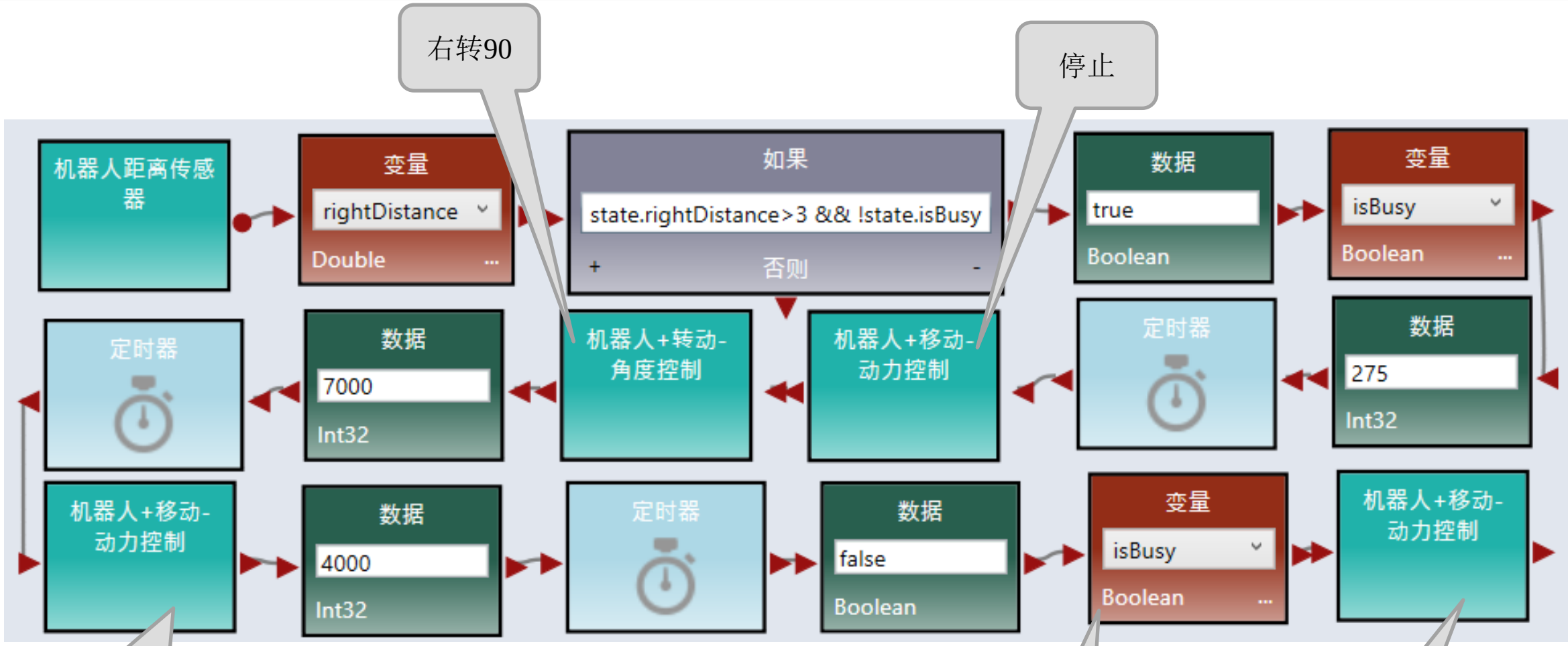
处理右方距离传感器



处理前方距离传感器（第 1 部分）

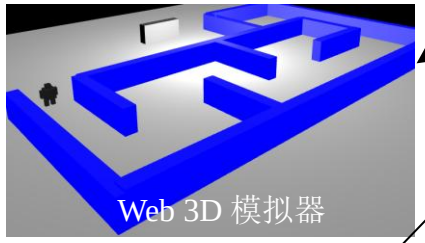


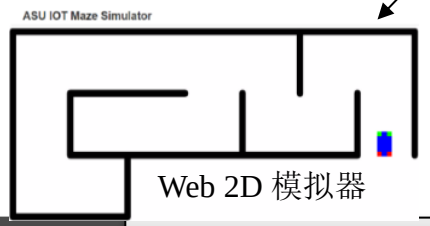
处理右方距离传感器（第 2 部分）



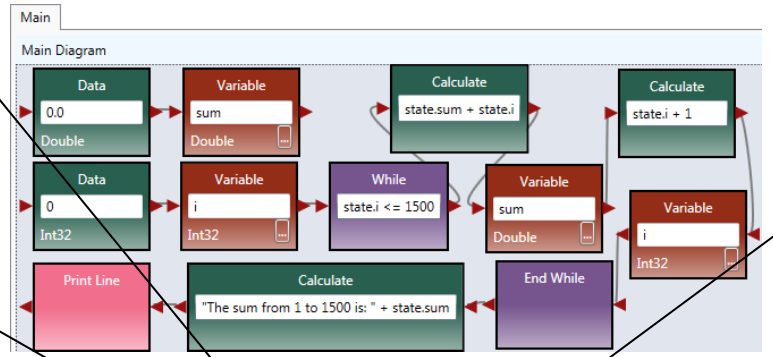
VIPLE 支持的平台

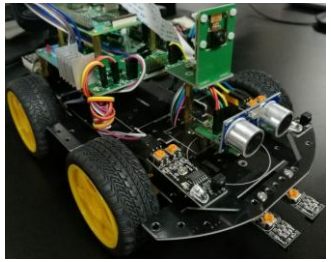
自动驾驶赛车编程



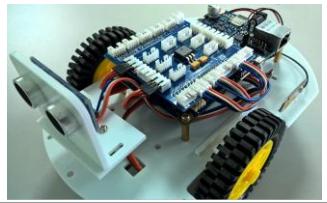


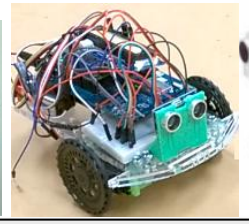
http://venus.sod.asu.edu/VIPLE/

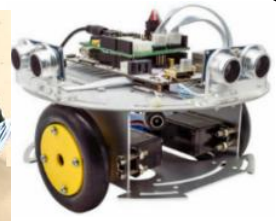




Raspberry Pi 机器人







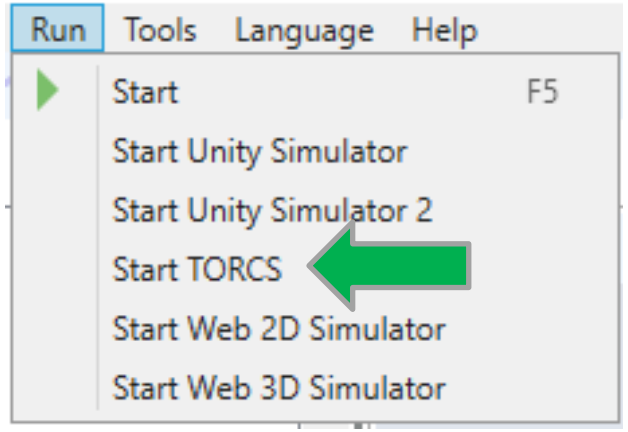
Lego EV3

自动驾驶赛车编程

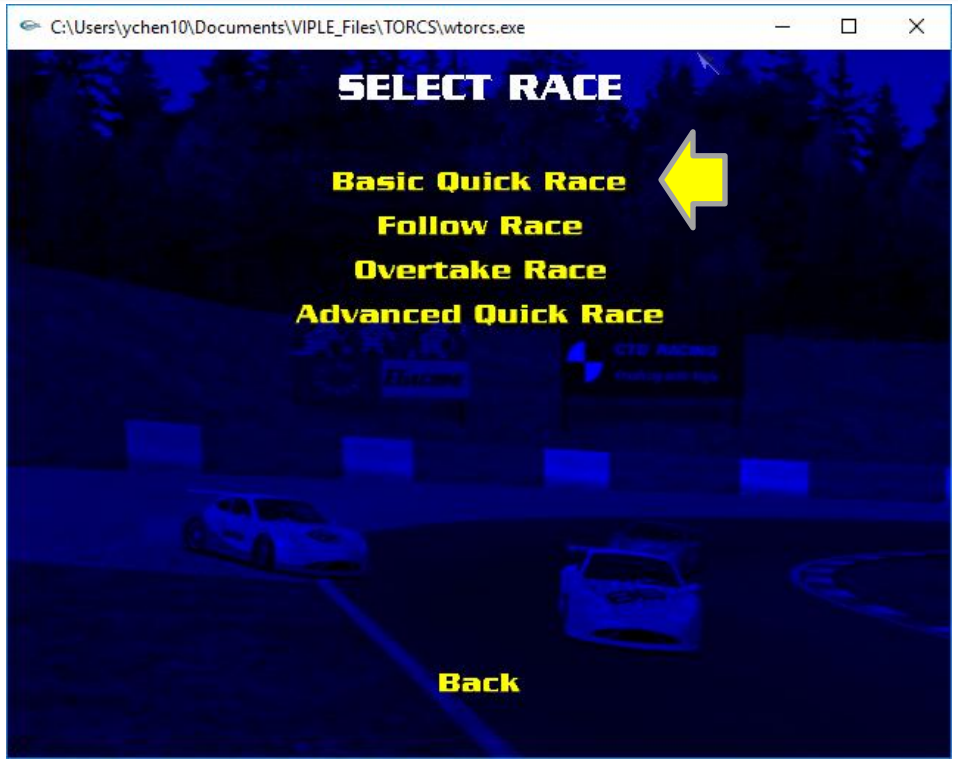
http://torcs.sourceforge.net/

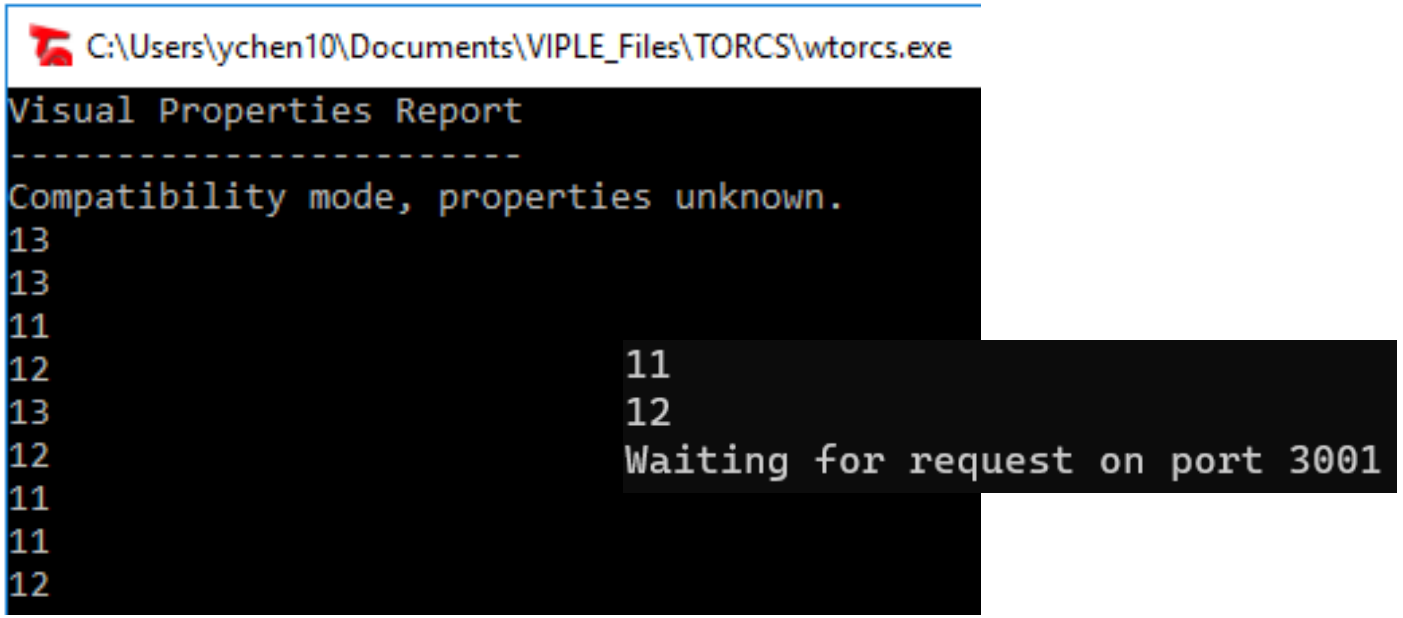
(1) 编写 VIPLE 代码。

(2) 启动 TORCS



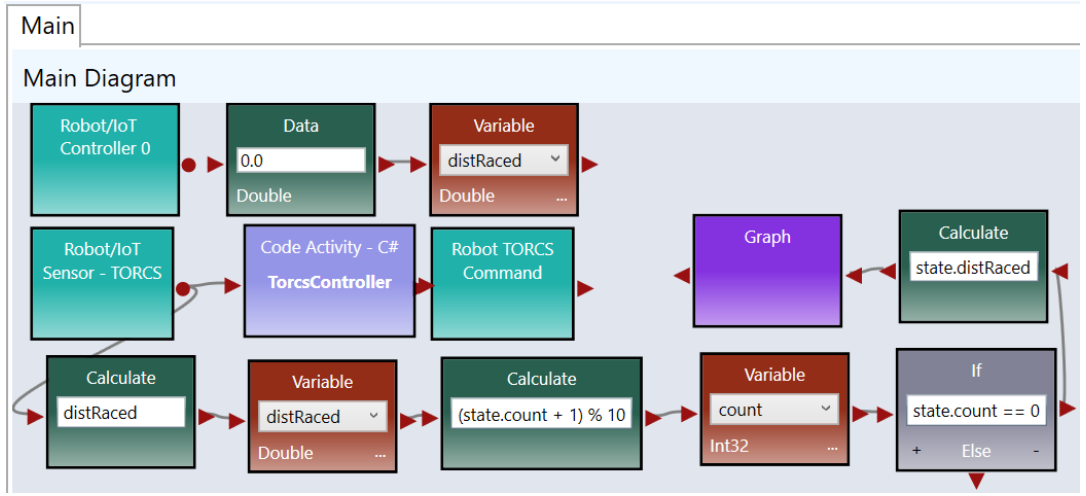
(3) 选 Race
(4) 选 Basic Quick Race
(5) 选 New Race

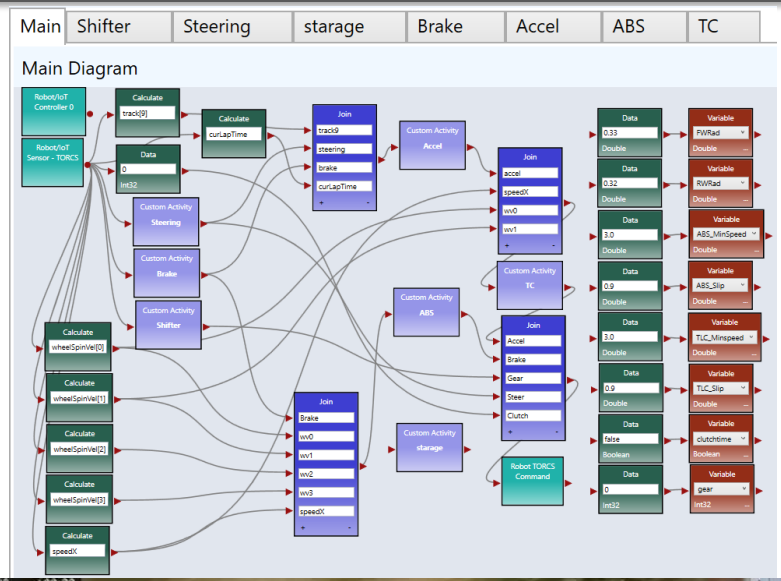


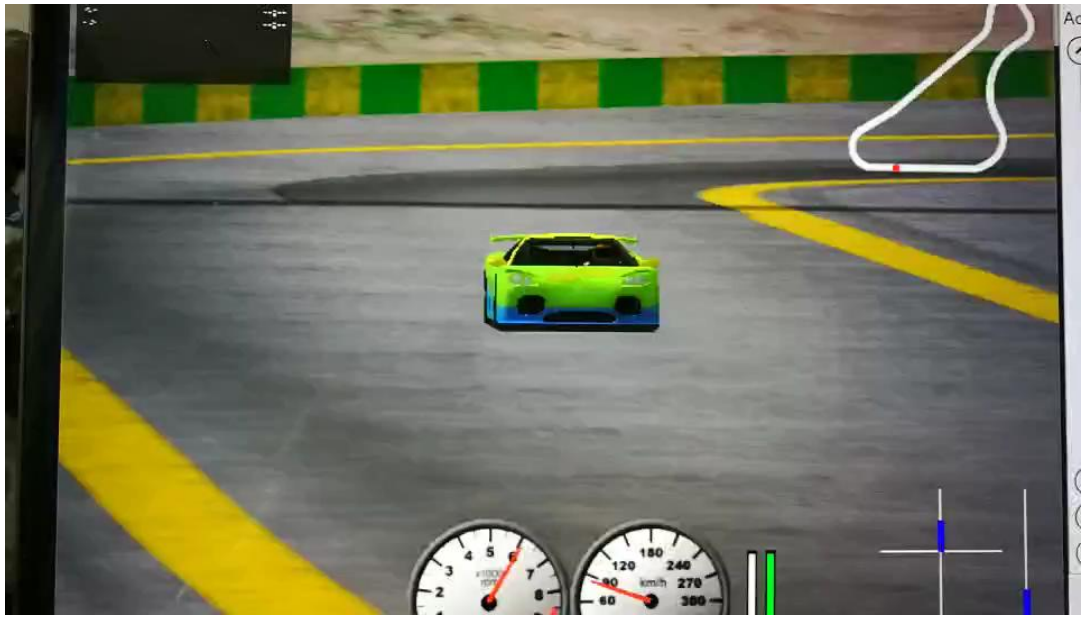


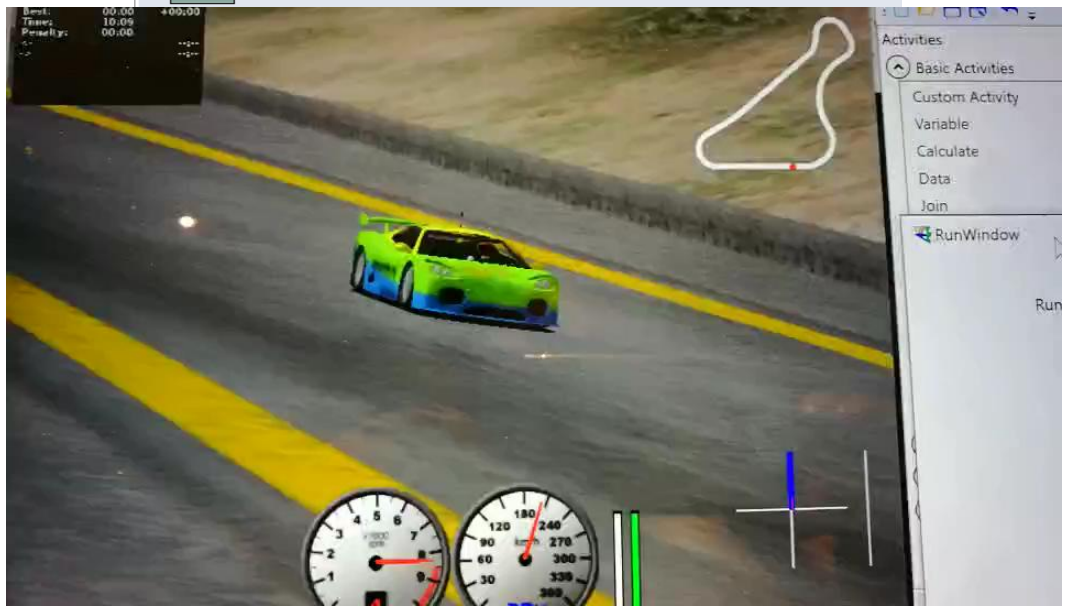
(6) 当仿真器显示:
Waiting for request on port 3001
(7) 启动 VIPLE 代码。

自动驾驶：基础和高级

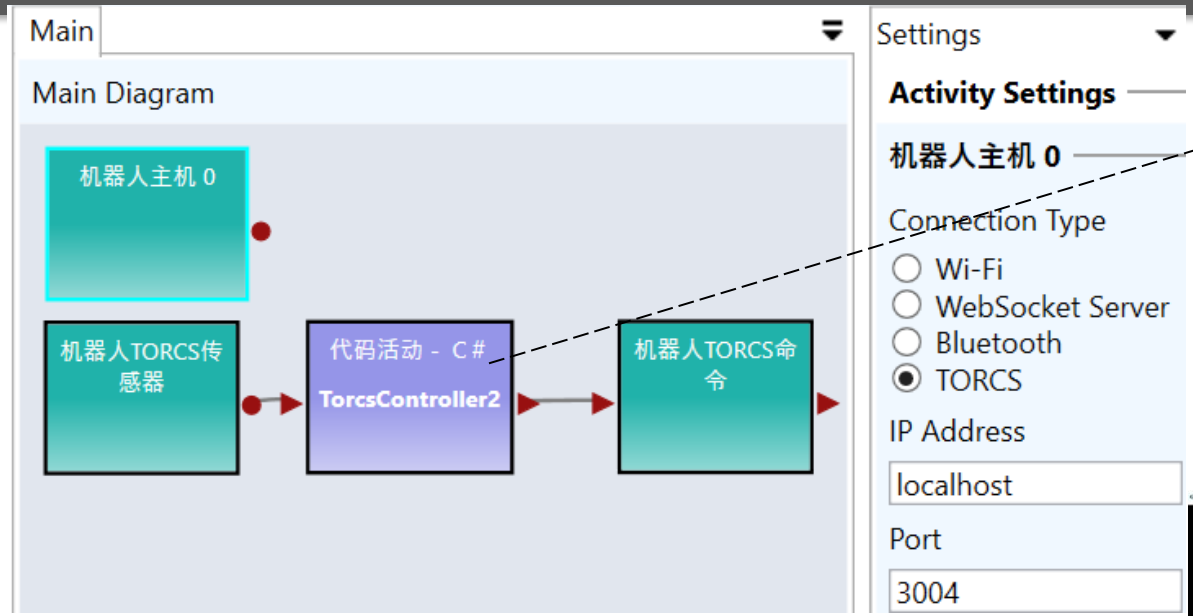

Main
Main Diagram
Robot/IoT Controller 0
Data
0.0
Double
Variable
distRaced
Double
Robot/IoT Sensor - TORCS
Code Activity - C#
TorcsController
Robot TORCS Command
Graph
Calculate
state.distRaced
Calculate
distRaced
Variable
distRaced
Double
Calculate
(state.count + 1) % 10
Variable
count
Int32
If
state.count == 0
+
Else
-


Main
Shifter
Steering
starage
Brake
Accel
ABS
TC
Main Diagram




Activities
Basic Activities
Custom Activity
Variable
Calculate
Data
Join
RunWindow

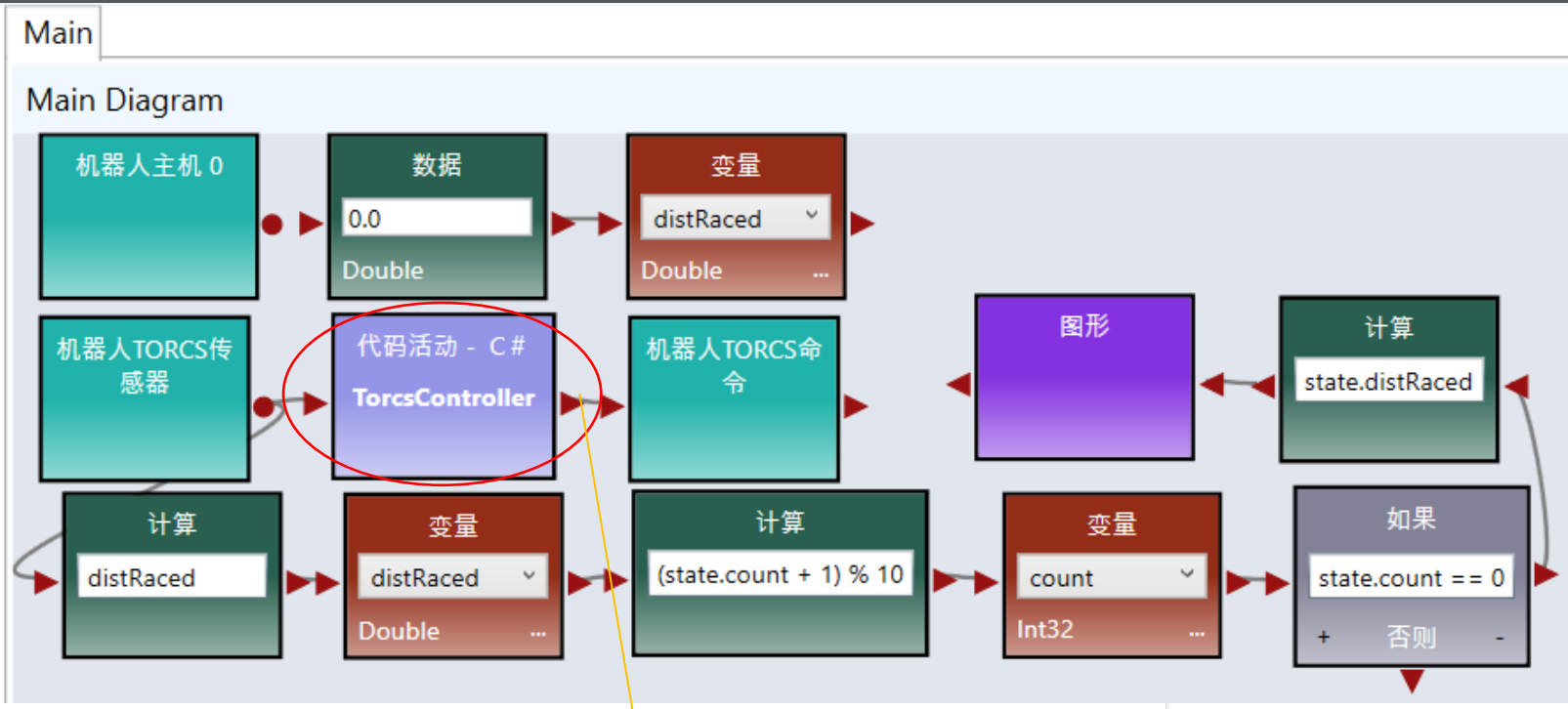
自动驾驶：两车竞赛



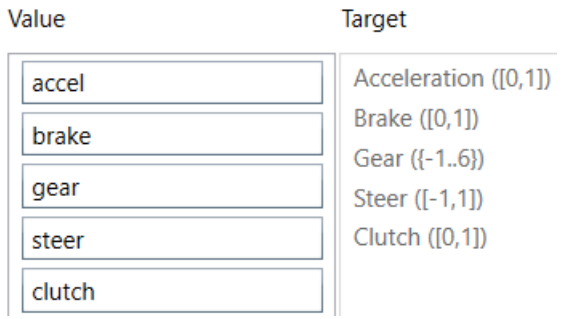

```
if(GetValue("racePos") == 2)
{
    // keeps car centered
    steer = GetValue("angle") - GetValue("trackPos") * 0.5;

    if(GetValue("speedX") < 50) // gear 1
    {
        gear = 1;
    }
    else if(GetValue("speedX") > 60 && GetValue("speedX") < 118)
    {
        gear = 2;
    }
    else if(GetValue("speedX") > 123)// && GetValue("speedX") < 155)
    {
```

基本程序中的主函数



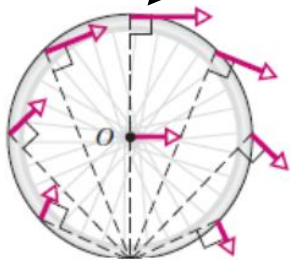

```
public override void Execute() {
    // 值作为对象传入，需要拆箱.
    var inputDictionary = Input as Dictionary<string, object>;
    // 返回字典允许我们按名称引用值.
    var toReturn = new Dictionary<string, double>() {
      {“accel”, 0.3}, // key-value pair 关键字-值对
      {"brake", 0.0},
      {"gear", 2},
      // 需要将这些值转换为双精度值，因为它们实际如此.
      {"steer"  (double)inputDictionary["angle"] - (double)inputDictionary "trackPos"  * 0.5}
      {"clutch", 0.2}
    };
    Output = toReturn;
  }
```



高级编程：TORCS 赛车仿真器中的传感器

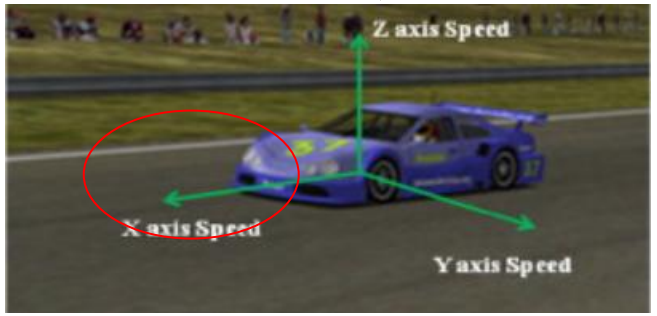
传感器 1：物理信息
(a)速度
(velocity)

车轮产生的速度



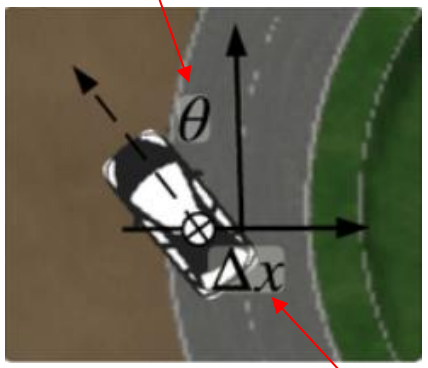
wv1
wv2
wv3
Wv4
rmp

车身速度



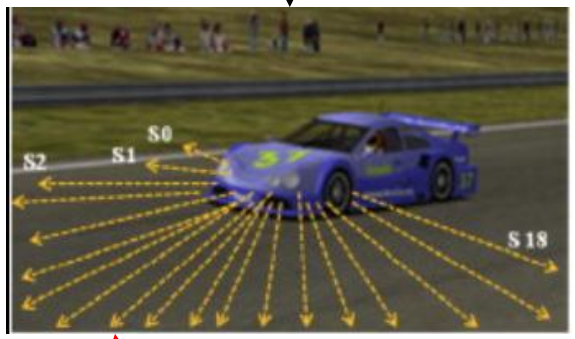
speedX
speedY
speedZ

(b) 位置和方向
(角度 angle)



位置
trackPos

传感器 2:
(c) 激光测距



S9 前方
中间
track[9]

可控制设备：
Brake, gear, clutch,
steer, accel

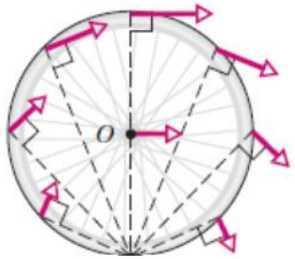
传感器 3:
(d) 前方摄像头

可以用着图像识别和机器学习

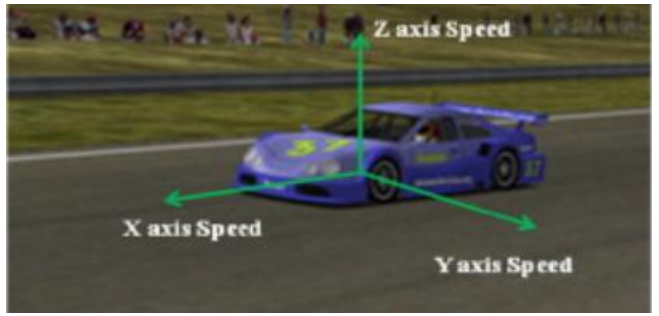
https://xed.ch/help/torcs.html

防抱死制动系统（ABS）和牵引力控制（TC）概念

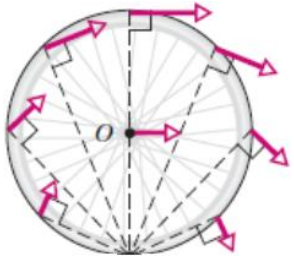
车轮产生的速度　　车身速度



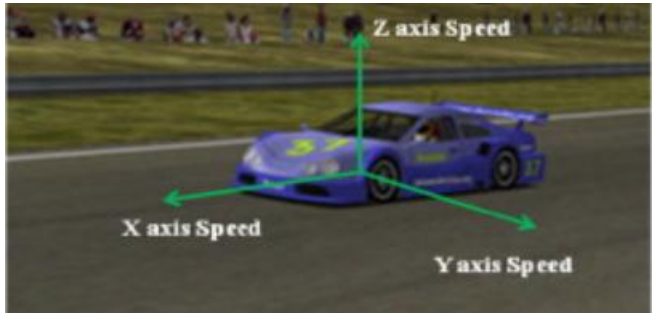
=



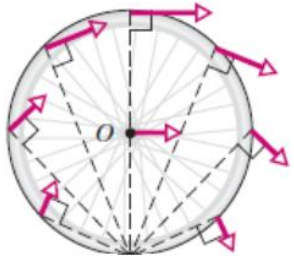
正常驾驶 ABS 和 TC 不参与



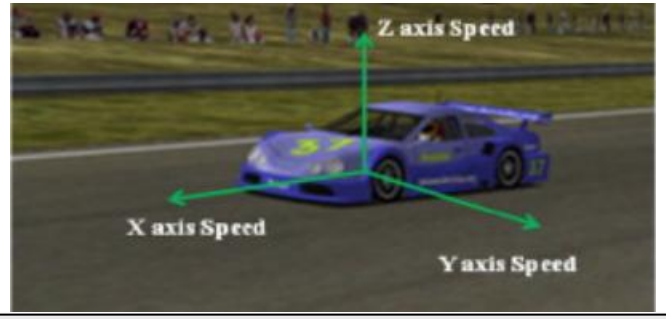
<



ABS 减少刹车力度

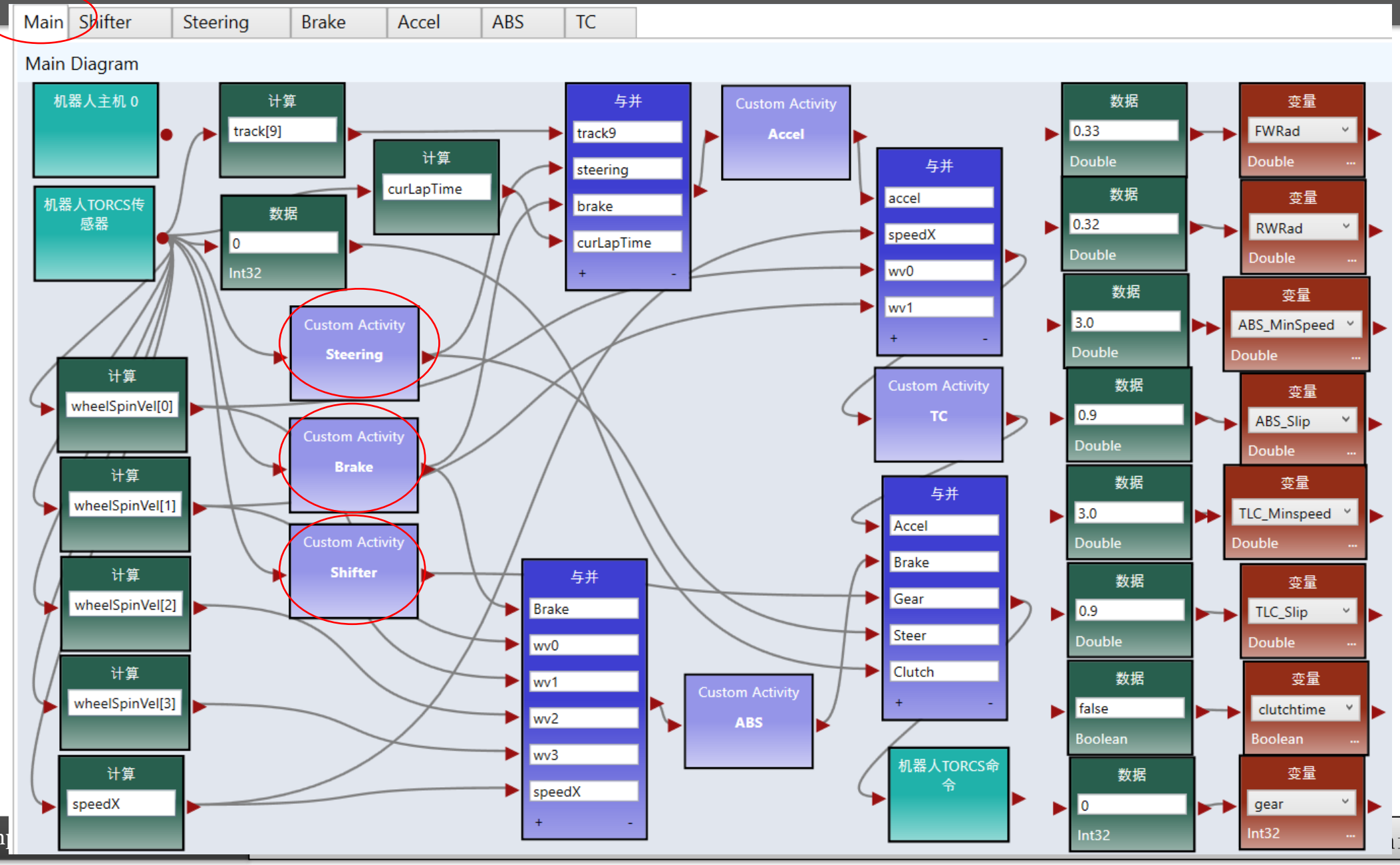


>

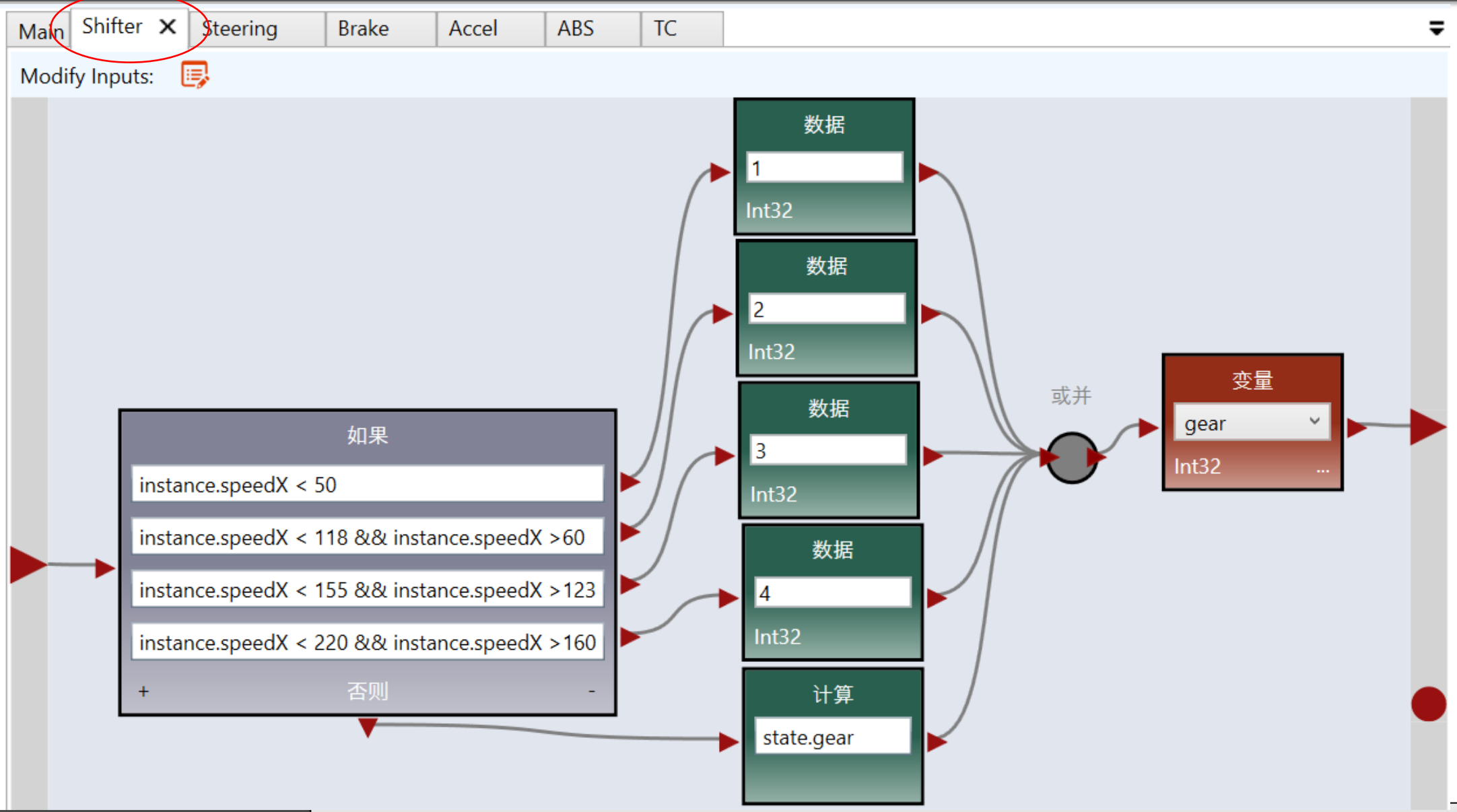


TC 减少加油量

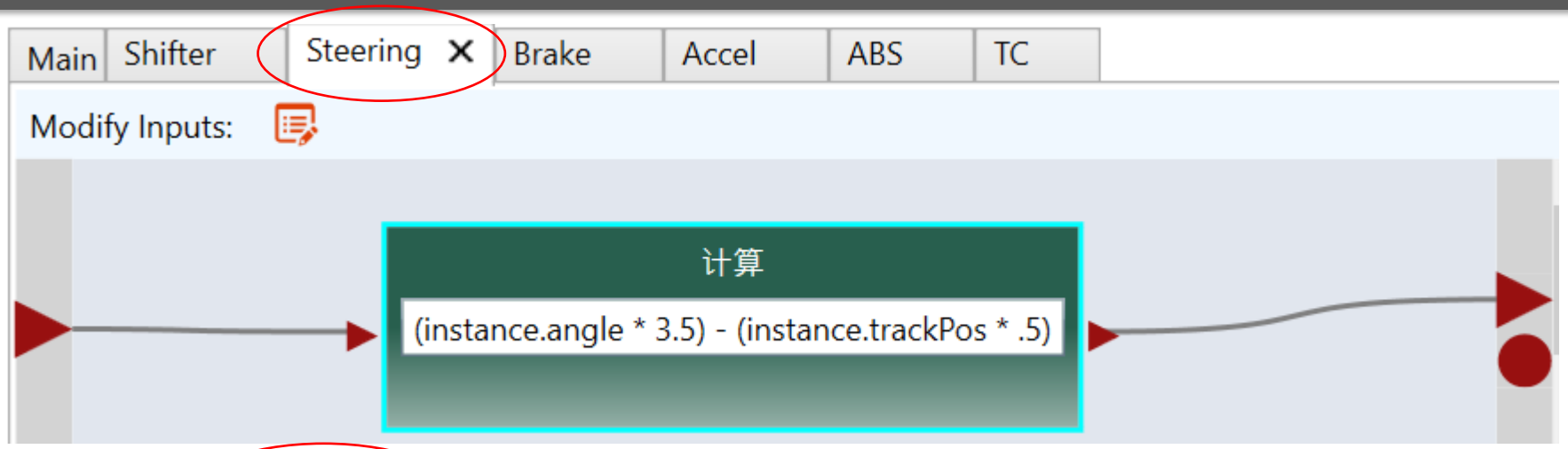
高级程序中的主函数

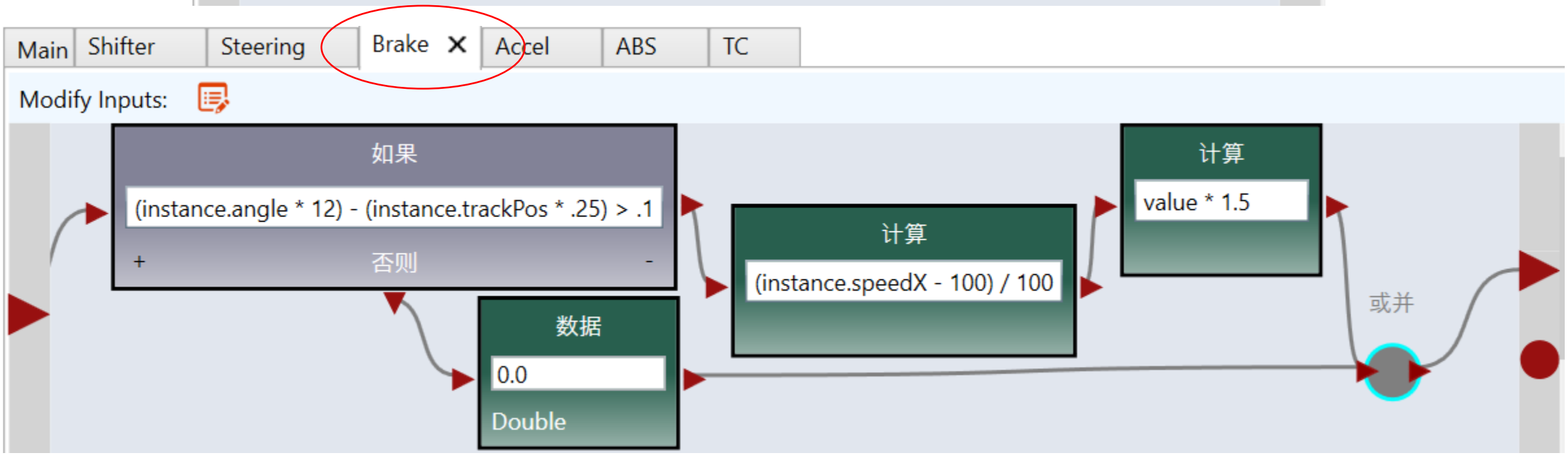

Main
Shifter
Steering
Brake
Accel
ABS
TC
Main Diagram
机器人主机 0
机器人TORCS传感器
计算
track[9]
数据
0
Int32
计算
curLapTime
与并
track9
steering
brake
curLapTime
Custom Activity
Accel
与并
accel
speedX
wv0
wv1
数据
0.33
Double
变量
FWRad
Double
数据
0.32
Double
变量
RWRad
Double
数据
3.0
Double
变量
ABS_MinSpeed
Double
Custom Activity
Steering
Custom Activity
Brake
Custom Activity
Shifter
计算
wheelSpinVel[0]
计算
wheelSpinVel[1]
计算
wheelSpinVel[2]
计算
wheelSpinVel[3]
计算
speedX
Custom Activity
TC
数据
0.9
Double
变量
ABS_Slip
Double
与并
Accel
Brake
Gear
Steer
Clutch
数据
3.0
Double
变量
TLC_Minspeed
Double
数据
0.9
Double
变量
TLC_Slip
Double
与并
Brake
wv0
wv1
wv2
wv3
speedX
Custom Activity
ABS
机器人TORCS命令
数据
false
Boolean
变量
clutchtime
Boolean
数据
0
Int32
变量
gear
Int32

换档活动

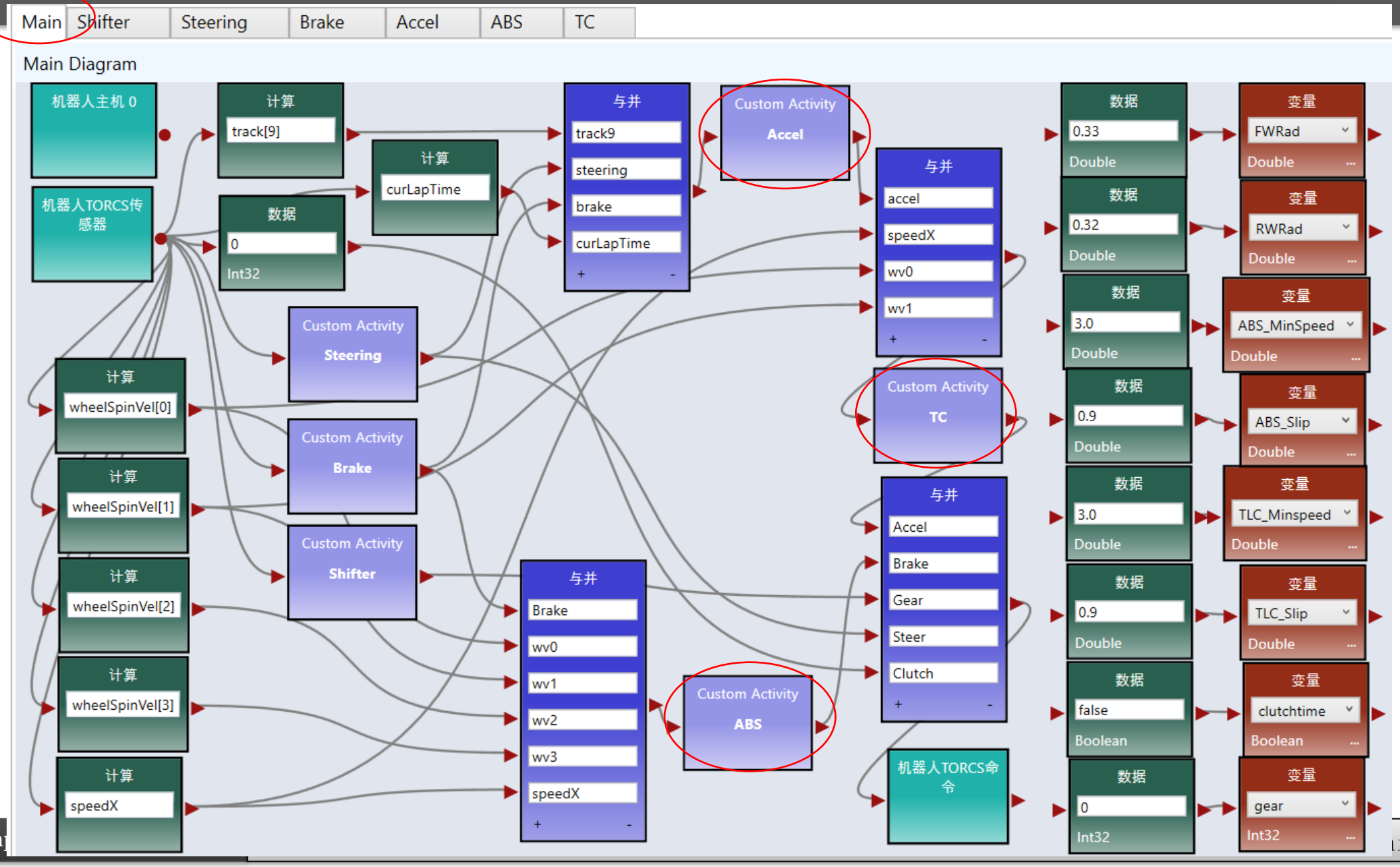

Main
Shifter
Steering
Brake
Accel
ABS
TC
Modify Inputs:
数据
1
Int32
数据
2
Int32
数据
3
Int32
数据
4
Int32
计算
state.gear
或并
变量
gear
Int32
如果
instance.speedX < 50
instance.speedX < 118 && instance.speedX >60
instance.speedX < 155 && instance.speedX >123
instance.speedX < 220 && instance.speedX >160
+
否则
-

制动活动和转向活动

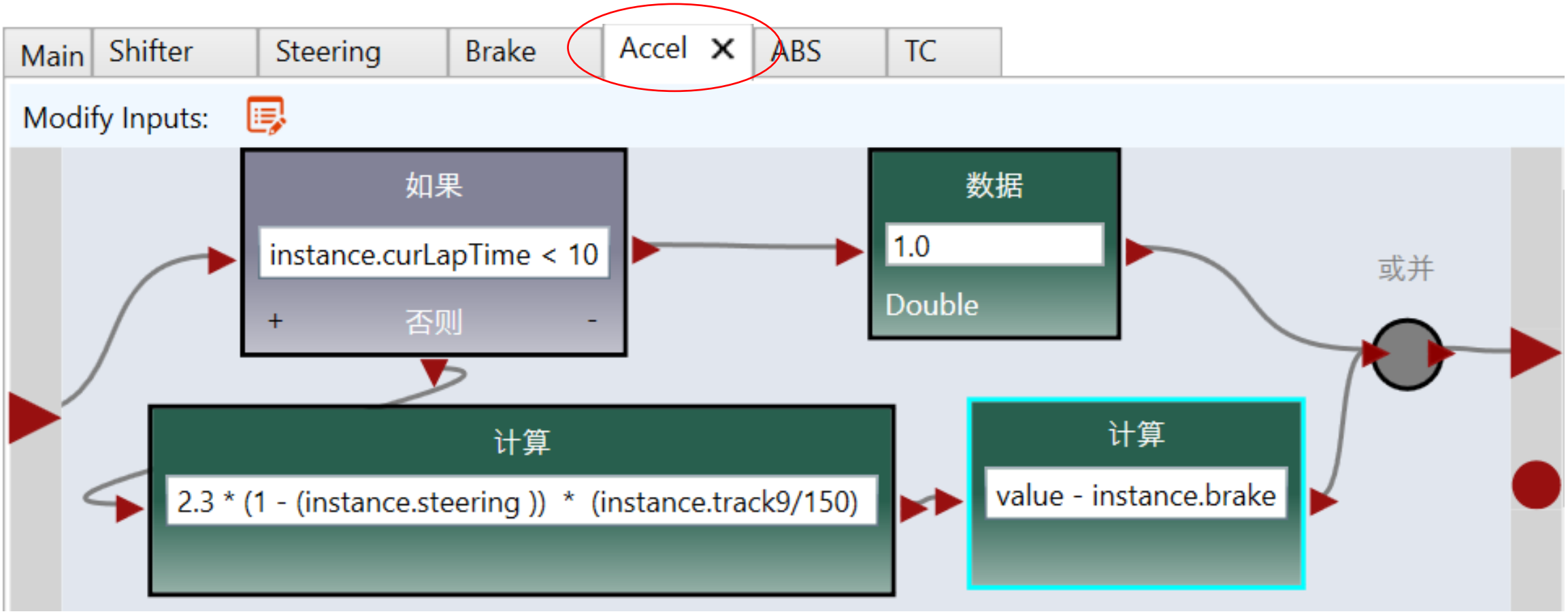

Main
Shifter
Steering
Brake
Accel
ABS
TC
Modify Inputs:
计算
(instance.angle * 3.5) - (instance.trackPos * .5)


Main
Shifter
Steering
Brake
Accel
ABS
TC
Modify Inputs:
如果
(instance.angle * 12) - (instance.trackPos * .25) > .1
+
否则
-
计算
(instance.speedX - 100) / 100
计算
value * 1.5
数据
0.0
Double
或并

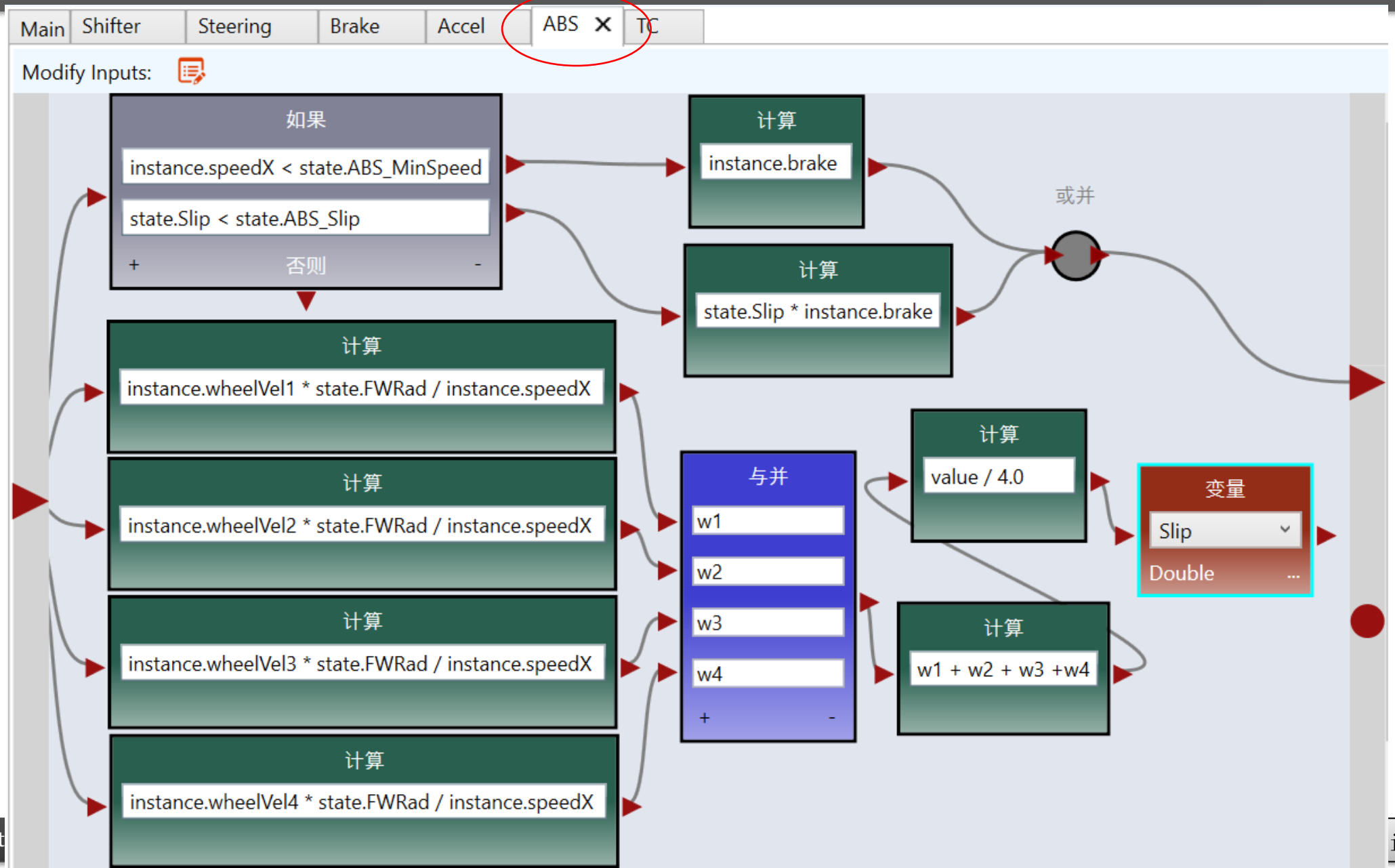
高级程序中的主函数


Main
Shifter
Steering
Brake
Accel
ABS
TC
Main Diagram
机器人主机 0
机器人TORCS传感器
计算
track[9]
数据
0
Int32
计算
curLapTime
与并
track9
steering
brake
curLapTime
Custom Activity
Accel
Custom Activity
Steering
Custom Activity
Brake
Custom Activity
Shifter
计算
wheelSpinVel[0]
计算
wheelSpinVel[1]
计算
wheelSpinVel[2]
计算
wheelSpinVel[3]
计算
speedX
与并
Brake
wv0
wv1
wv2
wv3
speedX
Custom Activity
ABS
与并
accel
speedX
wv0
wv1
Custom Activity
TC
与并
Accel
Brake
Gear
Steer
Clutch
机器人TORCS命令
数据
0.33
Double
数据
0.32
Double
数据
3.0
Double
数据
0.9
Double
数据
3.0
Double
数据
0.9
Double
数据
false
Boolean
数据
0
Int32
变量
FWRad
Double
变量
RWRad
Double
变量
ABS_MinSpeed
Double
变量
ABS_Slip
Double
变量
TLC_Minspeed
Double
变量
TLC_Slip
Double
变量
clutchtime
Boolean
变量
gear
Int32

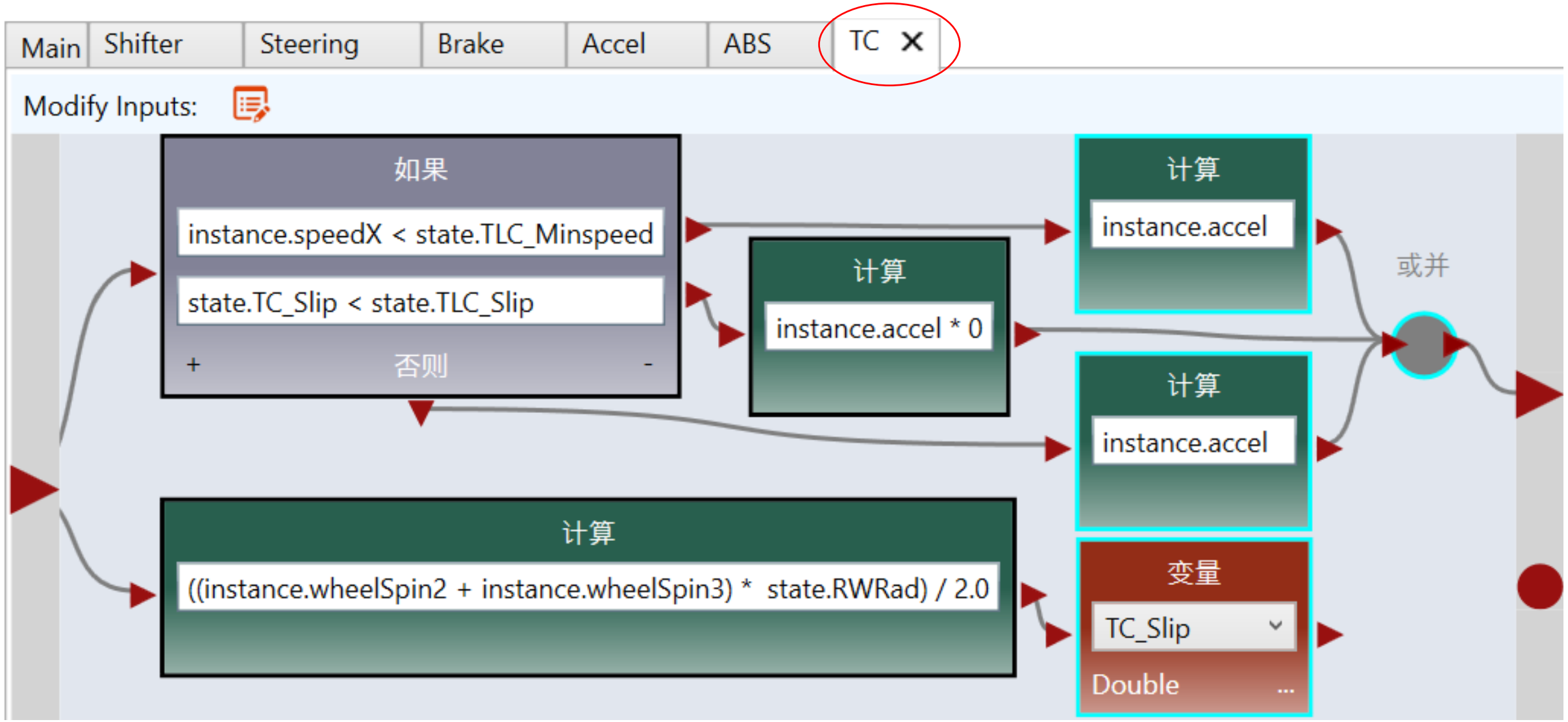
加速活动


Main
Shifter
Steering
Brake
Accel
ABS
TC
Modify Inputs:
如果
instance.curLapTime < 10
+
否则
-
数据
1.0
Double
或并
计算
2.3 * (1 - (instance.steering)) * (instance.track9/150)
计算
value - instance.brake

制动防抱死系统(ABS)活动



牵引力控制 (TC) 活动


Main
Shifter
Steering
Brake
Accel
ABS
TC
Modify Inputs:
如果
instance.speedX < state.TLC_Minspeed
state.TC_Slip < state.TLC_Slip
+
否则
-
计算
instance.accel * 0
计算
instance.accel
计算
instance.accel
或并
计算
((instance.wheelSpin2 + instance.wheelSpin3) * state.RWRad) / 2.0
变量
TC_Slip
Double
...