



# CSCI 1106 Lecture 12

Introduction to Robotics



## Announcements

- Today's Topics
  - Overview of Robotics Module
  - What is Robotics?
  - Anatomy of a Robot
  - The Sense-Decide-Act Cycle
  - Introduction to the Aseba Studio



## The Robotics Module

### Topics

- Overview of Robotics
- Hardware
  - Sensors
  - Actuators
- Software
  - State Transition Diagrams
  - Event Based Architecture
  - Dealing with Failure
  - Planning
  - Debugging
  - Programming Techniques

### To Do List

- Six tutorials:
  - Introducing the Thymio II
  - Modeling sensors
  - Modeling actuators
  - Modeling the real world
  - Recovering from faults
  - Programming Techniques
- Robot Olympics Project
  - Design three programs to compete in the Robot Olympics
  - Implement the programs
  - Compete in the Robot Olympics
  - Write a technical report



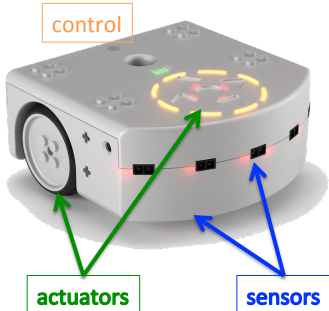
## What is Robotics?

- From the Thrun, Burgard, and Fox  
“*Robotics*Robotics is the science of perceiving and manipulating the physical world through computer-controlled devices.”

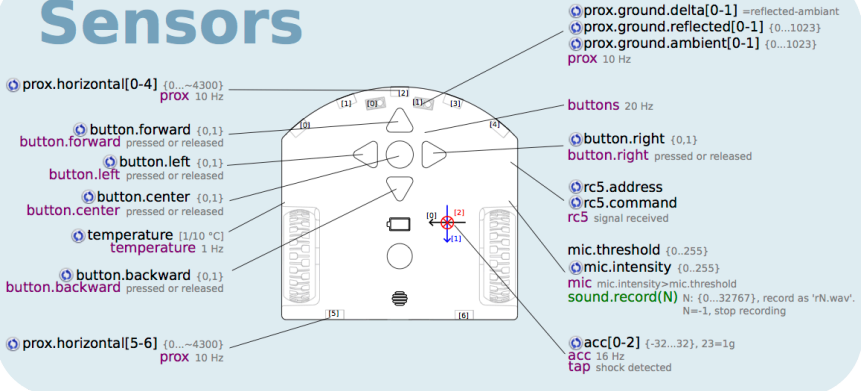


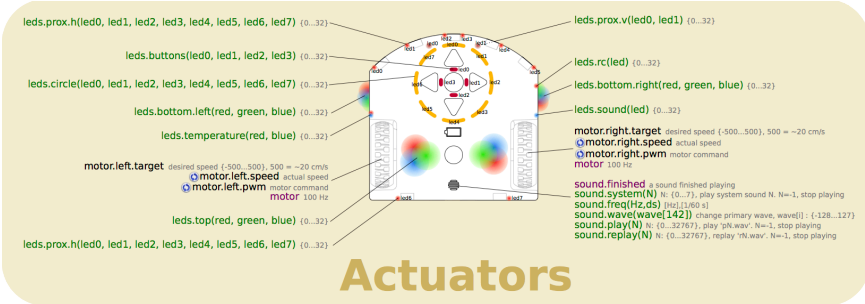
# Anatomy of a Robot

- Components:
  - Sensors
  - Controller
  - Actuators

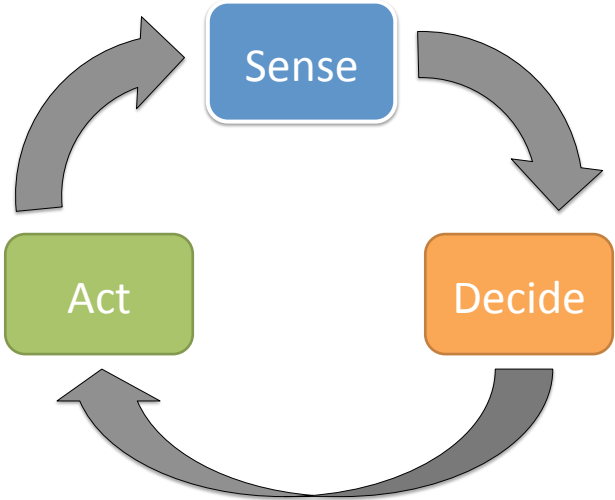


## Sensors





# The Sense-Decide-Act Framework



## Programming in Aseba

AG

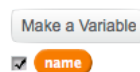
- Programs are text-based
- Key Ideas:
  - Everything is done by event handlers
  - A robot is a sprite
  - The world is the stage
- Observation this is similar to game design!

## Scratch vs Aseba

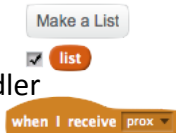
AG

### Scratch

- Variables



- Event Handler



- Conditional



### Aseba

- Variables

```
var name
var list[]
```

- Event Handler

```
onevent prox
```

- Conditional

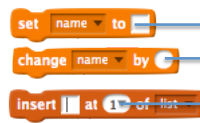
```
if then
end
```



## Scratch vs Aseba

### Scratch

- Variable/List Assignment



- Expressions



- Motion



### Aseba

- Variable/List Assignment

```
name =
name +=
list[ ] =
```

- Expressions

```
( name + 5 ) * list[2]
```

- Motion

```
motor.left.target =
motor.right.target =
```



## The Four Parts of an Aseba Program

- Variable declarations
  - Begin with the **var** keyword
- Initialization code
  - Anything except declarations
- Subroutines
  - Begin with the **sub** keyword
- Event handlers
  - Begin with the **onevent** keyword

Make a Variable

name

Make a List

list

when clicked

when I receive prox

## AG

# A Sample Program

```

var speed = 100

motor.left.target = 0
motor.right.target = 0

onevent button.forward
  motor.left.target = speed
  motor.right.target = speed

onevent button.backward
  motor.left.target = 0
  motor.right.target = 0

onevent button.left
  motor.left.target = -speed
  motor.right.target = speed

onevent button.right
  motor.left.target = speed
  motor.right.target = -speed

```

Key Idea: Actuators are controlled by setting variables that represent them

## AG

# Sensors and Actuators in Aseba

- Key Idea: All sensors and actuators are accessed via predefined variables, e.g.,
  - to control motors, assign values to motor variables
 

```

motor.left.target = 100
motor.right.target = 100

```
  - to check if an object is close, read proximity variable
 

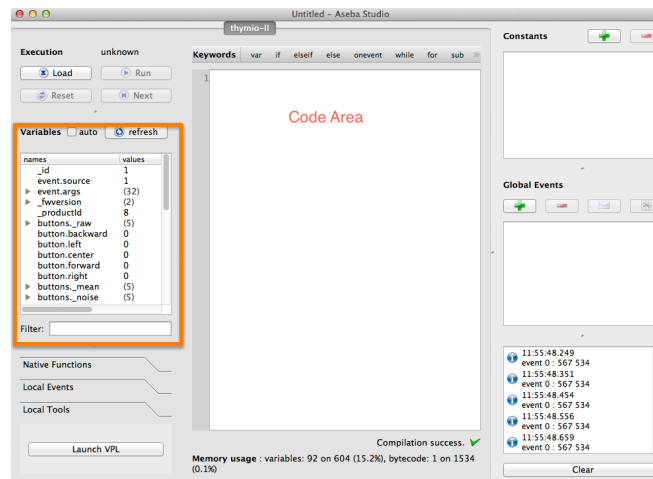
```

if prox.horizontal[2] > 1000 then
  ...
end

```
- Where are all the predefined variables listed?
- When do we check variables?

# Aseba Studio

AG

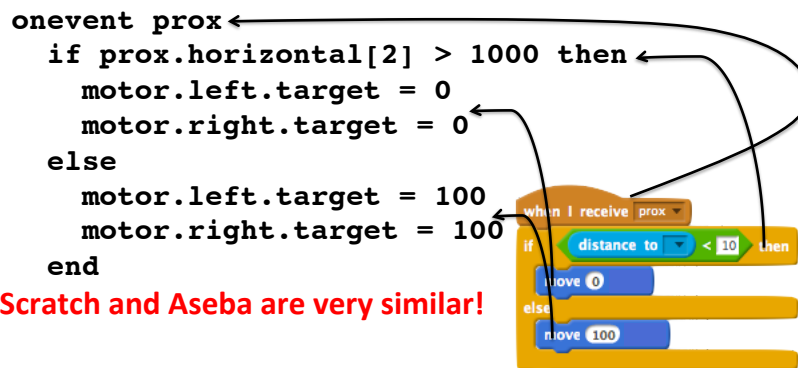


<https://aseba.wikidot.com/en:thymioapi>

## When do We Check the Sensors?

AG

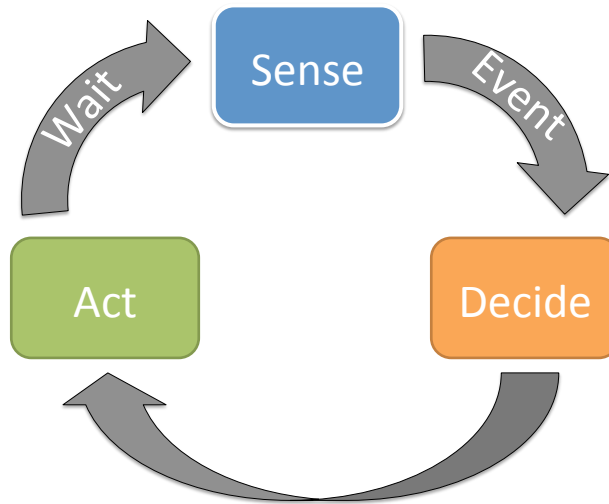
- Key Idea: Sensors generate events. Event handlers check sensors. E.g.,
  - Proximity (prox) sensors generate 10 events per second





## Event Driven Framework (Wait) Sense (Event)-Decide-Act

AG



## Last Example

AG

```

onevent prox
  if prox.horizontal[2] > 1000 then
    motor.left.target = 0
    motor.right.target = 0
  elseif prox.horizontal[4] > 1000 then
    motor.left.target = -100
    motor.right.target = 100
  elseif prox.horizontal[0] > 1000 then
    motor.left.target = 100
    motor.right.target = -100
  else
    motor.left.target = 100
    motor.right.target = 100
  end

```

