CSCI 1106
Lecture 11

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
  - Event Based Architecture
  - Dealing with Failure
  - Planning
  - Debugging
  - Programming Techniques
- Mindware
  - State Transition Diagrams
  - Motion model

To Do List

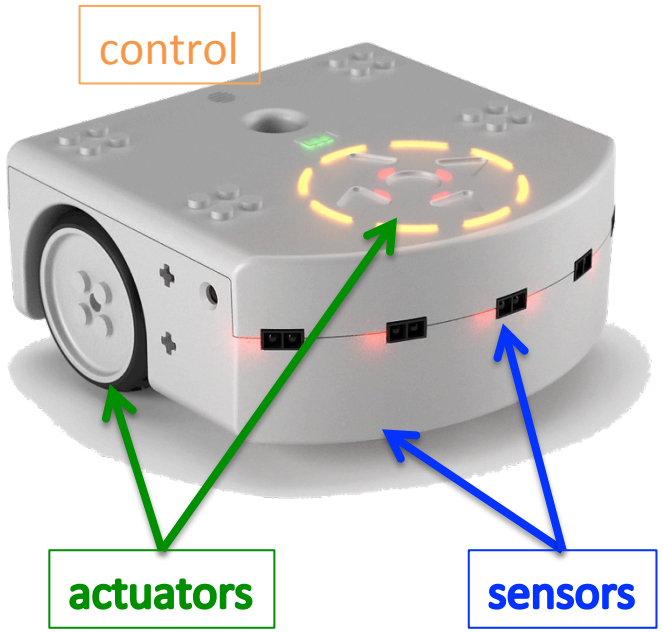
- Six tutorials:
  - Introducing the Thymio II
  - Modeling sensors
  - Modeling actuators
  - Modeling the real world
  - Recovering from faults
  - Programming Techniques
- Robot Olympics Project
  - Design three programs to compete in the Robot Olympics
    - Marathon, hurdles, and curling
  - Implement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

What is Robotics?

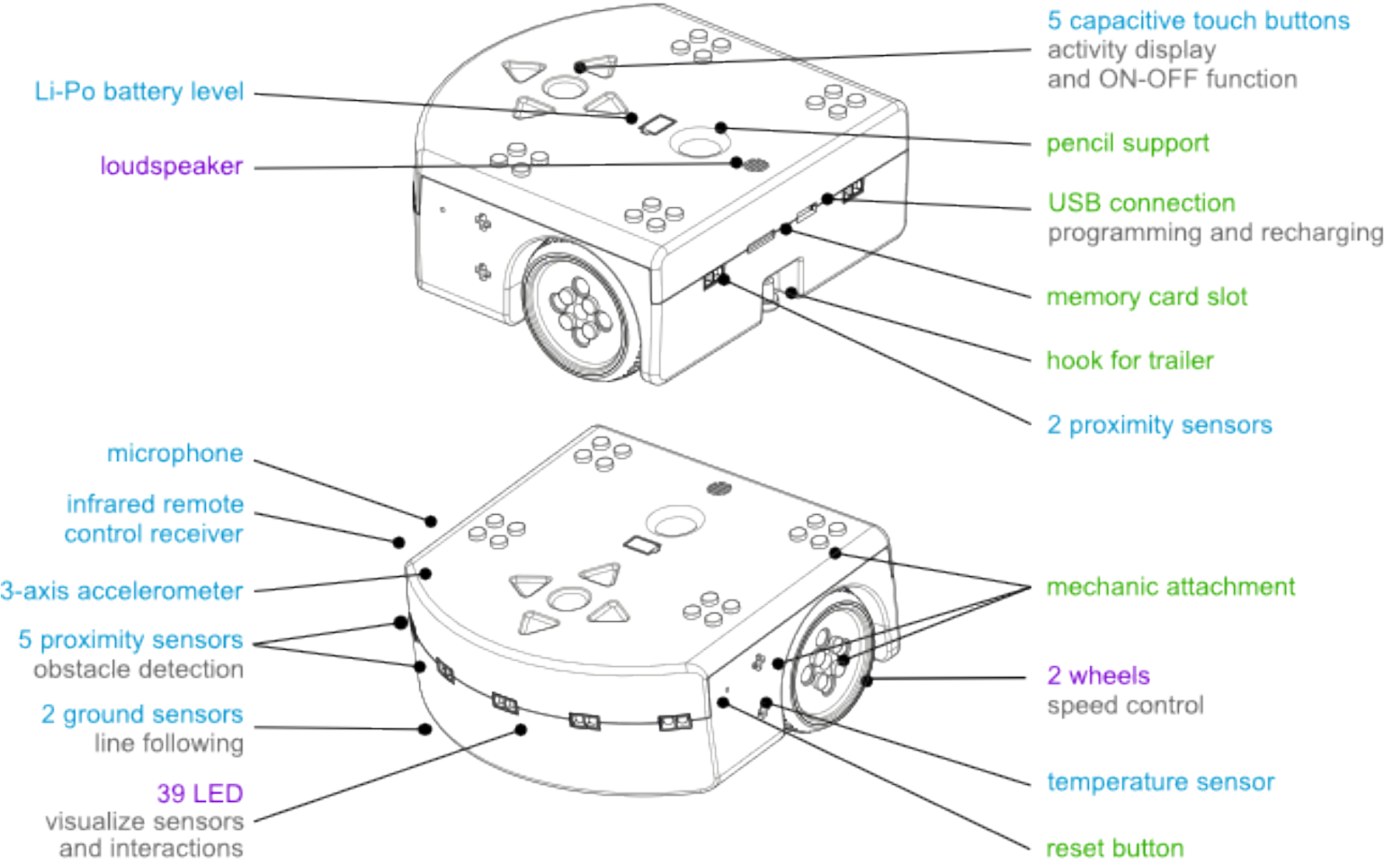
- From the Thrun, Burgard, and Fox
  "*Robotics* is the science of perceiving and manipulating the physical world through computer-controlled devices."
- A robot is composed of
  - Hardware: the machine
  - Software:
    the program that controls the machine
    (Robotics middleware like ROS, SLAM, CV, etc)
- Robotics includes both aspects.

Anatomy of a Robot

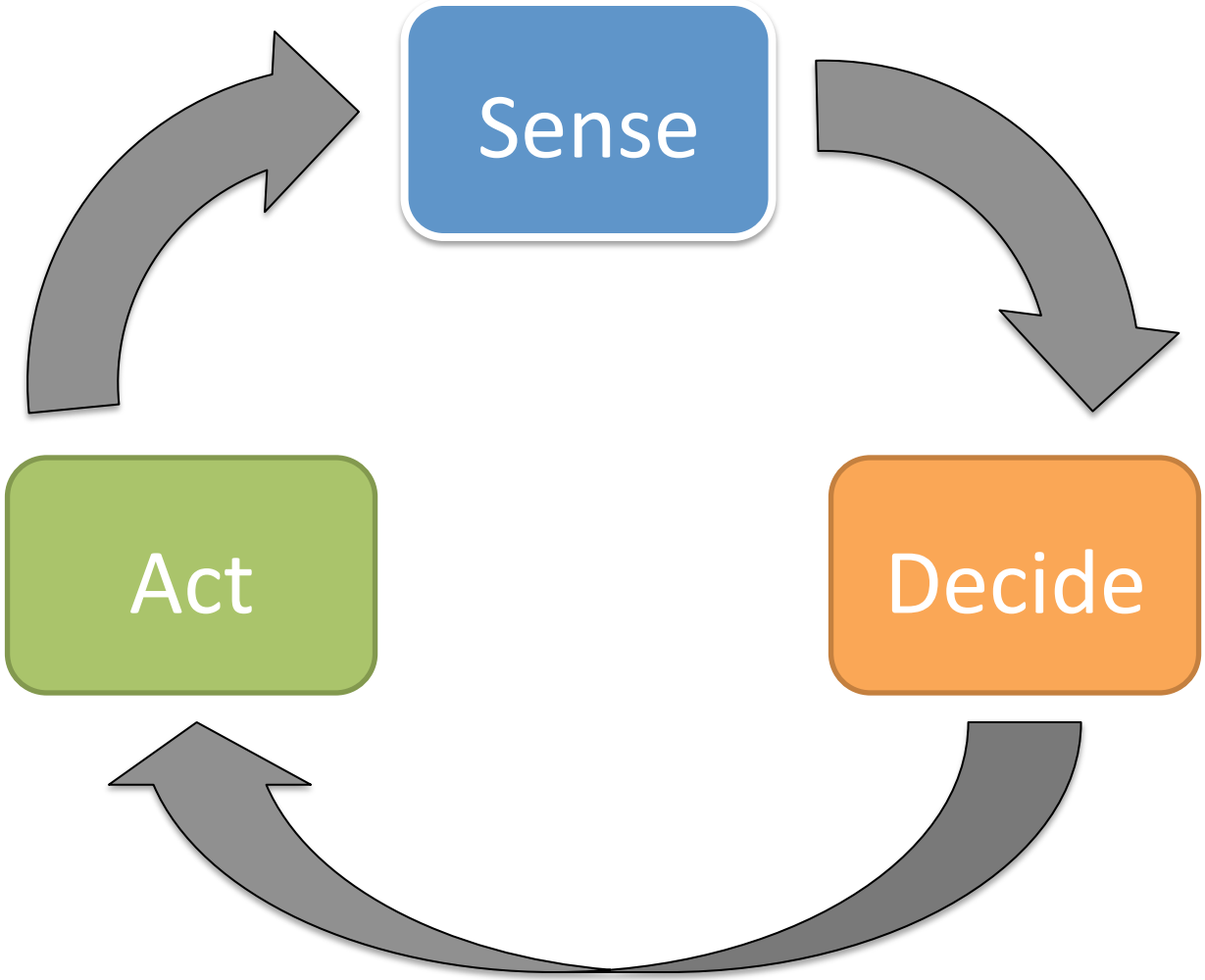
- Thymio II robot
  - https://aseba.wikidot.com
- Components:
  - Sensors
  - Controller
  - Actuators



Sensors and Actuators



The Sense-Decide-Act Framework



Subsumption architecture

Rodney Brooks

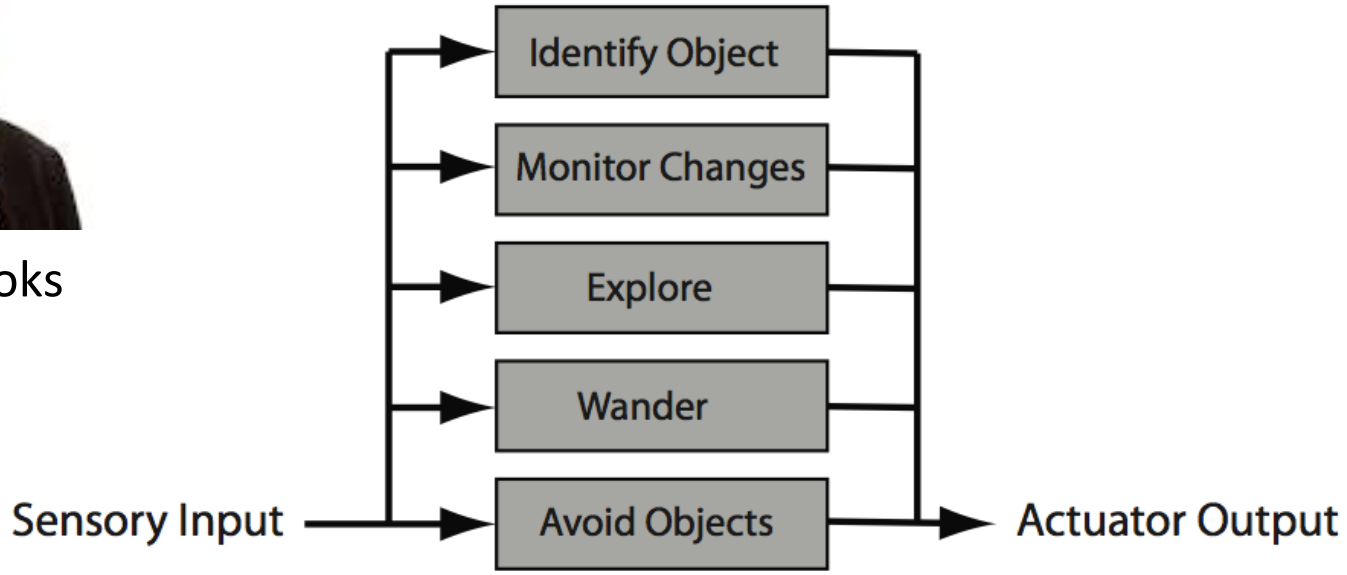


Fig. 2.1 An example of an subsumption architecture. From Maja J. Mataric, *The Robotocs Primer, MIT Press 2007*

What is the Robot Doing?

- What is being sensed?
- What is being decided?
- What action(s) result?

- -> state estmation !!!!

https://www.youtube.com/watch?v=ASoCJTYgYB0

https://www.youtube.com/watch?v=6wK0Ld13US8

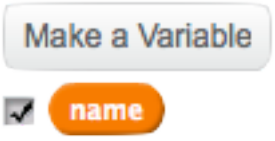
Programming in Aseba

- 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

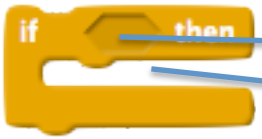
Scratch

- 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

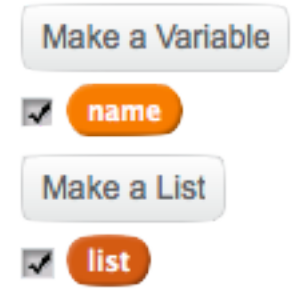
```
( name + 5 ) * list[2]
```

- Motion

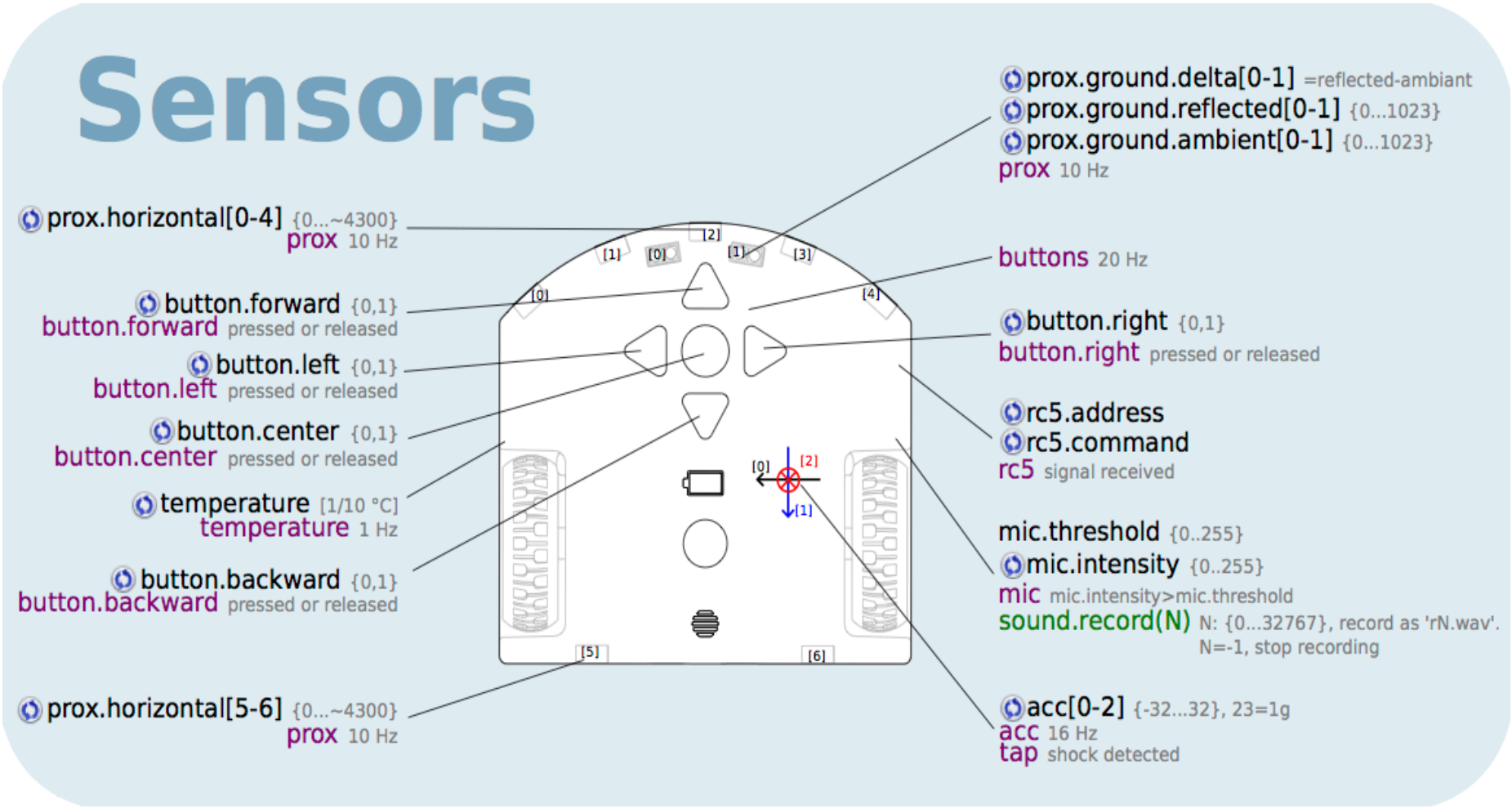
```
motor.left.target =
motor.right.target =
```

The Four Parts of an Aseba Program

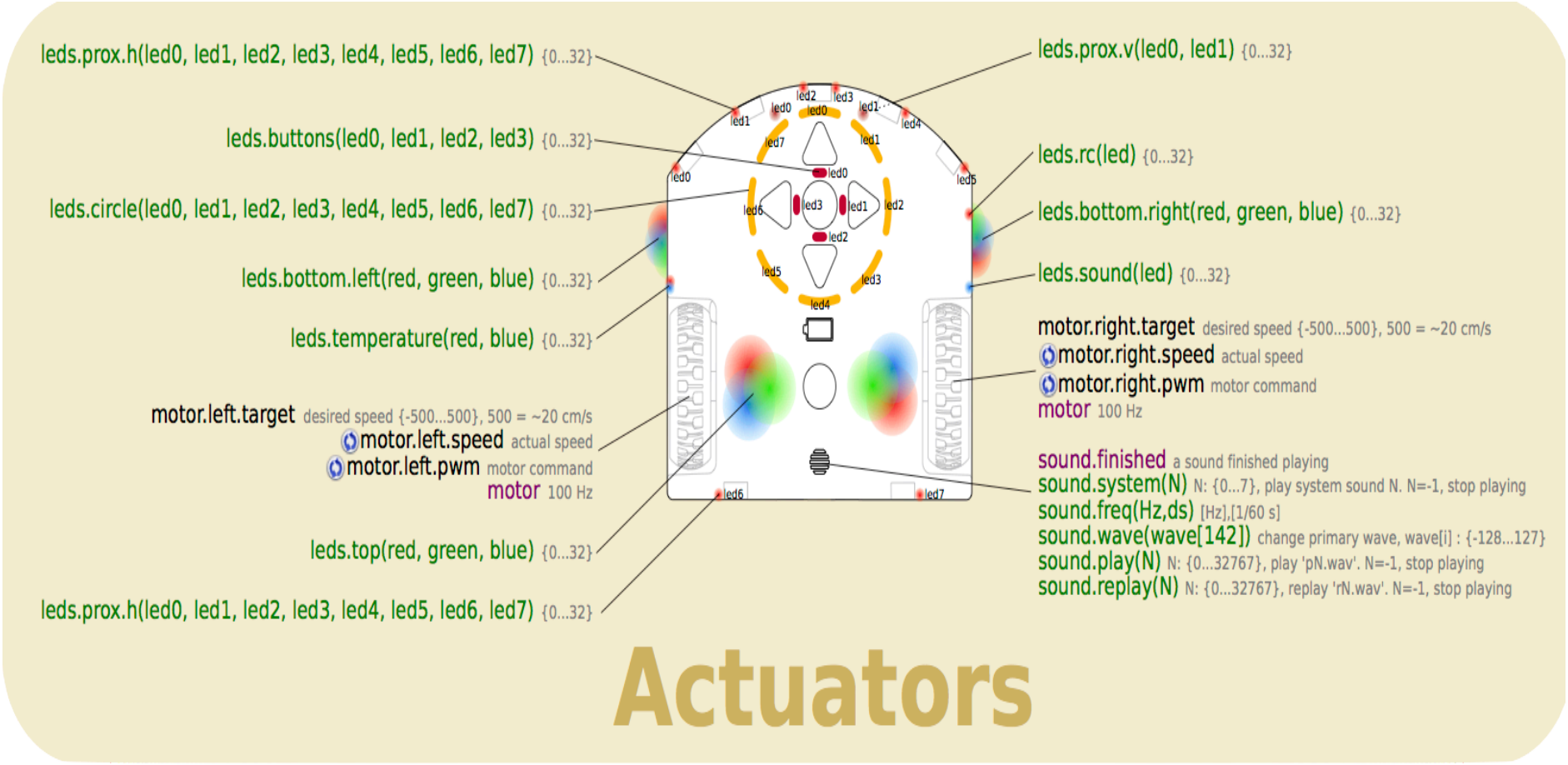
- Variable declarations
  - Begin with the **`var`** keyword
- Initialization code
  - Anything except declarations
- Event handlers
  - Begin with the **`onevent`** keyword
- Subroutines
  - Begin with the **`sub`** keyword



Sensors

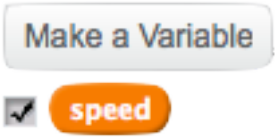

Sensors
prox.horizontal[0-4] {0...~4300}
prox 10 Hz
button.forward {0,1}
button.forward pressed or released
button.left {0,1}
button.left pressed or released
button.center {0,1}
button.center pressed or released
temperature [1/10 °C]
temperature 1 Hz
button.backward {0,1}
button.backward pressed or released
prox.horizontal[5-6] {0...~4300}
prox 10 Hz
prox.ground.delta[0-1] =reflected-ambiant
prox.ground.reflected[0-1] {0...1023}
prox.ground.ambient[0-1] {0...1023}
prox 10 Hz
buttons 20 Hz
button.right {0,1}
button.right pressed or released
rc5.address
rc5.command
rc5 signal received
mic.threshold {0..255}
mic.intensity {0..255}
mic mic.intensity>mic.threshold
sound.record(N) N: {0...32767}, record as 'rN.wav'.
N=-1, stop recording
acc[0-2] {-32...32}, 23=1g
acc 16 Hz
tap shock detected

Actuators


leds.prox.h(led0, led1, led2, led3, led4, led5, led6, led7) {0...32}
leds.buttons(led0, led1, led2, led3) {0...32}
leds.circle(led0, led1, led2, led3, led4, led5, led6, led7) {0...32}
leds.bottom.left(red, green, blue) {0...32}
leds.temperature(red, blue) {0...32}
motor.left.target desired speed {-500...500}, 500 = ~20 cm/s
motor.left.speed actual speed
motor.left.pwm motor command
motor 100 Hz
leds.top(red, green, blue) {0...32}
leds.prox.h(led0, led1, led2, led3, led4, led5, led6, led7) {0...32}
leds.prox.v(led0, led1) {0...32}
leds.rc(led) {0...32}
leds.bottom.right(red, green, blue) {0...32}
leds.sound(led) {0...32}
motor.right.target desired speed {-500...500}, 500 = ~20 cm/s
motor.right.speed actual speed
motor.right.pwm motor command
motor 100 Hz
sound.finished a sound finished playing
sound.system(N) N: {0...7}, play system sound N. N=-1, stop playing
sound.freq(Hz,ds) [Hz],[1/60 s]
sound.wave(wave[142]) change primary wave, wave[i] : {-128...127}
sound.play(N) N: {0...32767}, play 'pN.wav'. N=-1, stop playing
sound.replay(N) N: {0...32767}, replay 'rN.wav'. N=-1, stop playing
Actuators

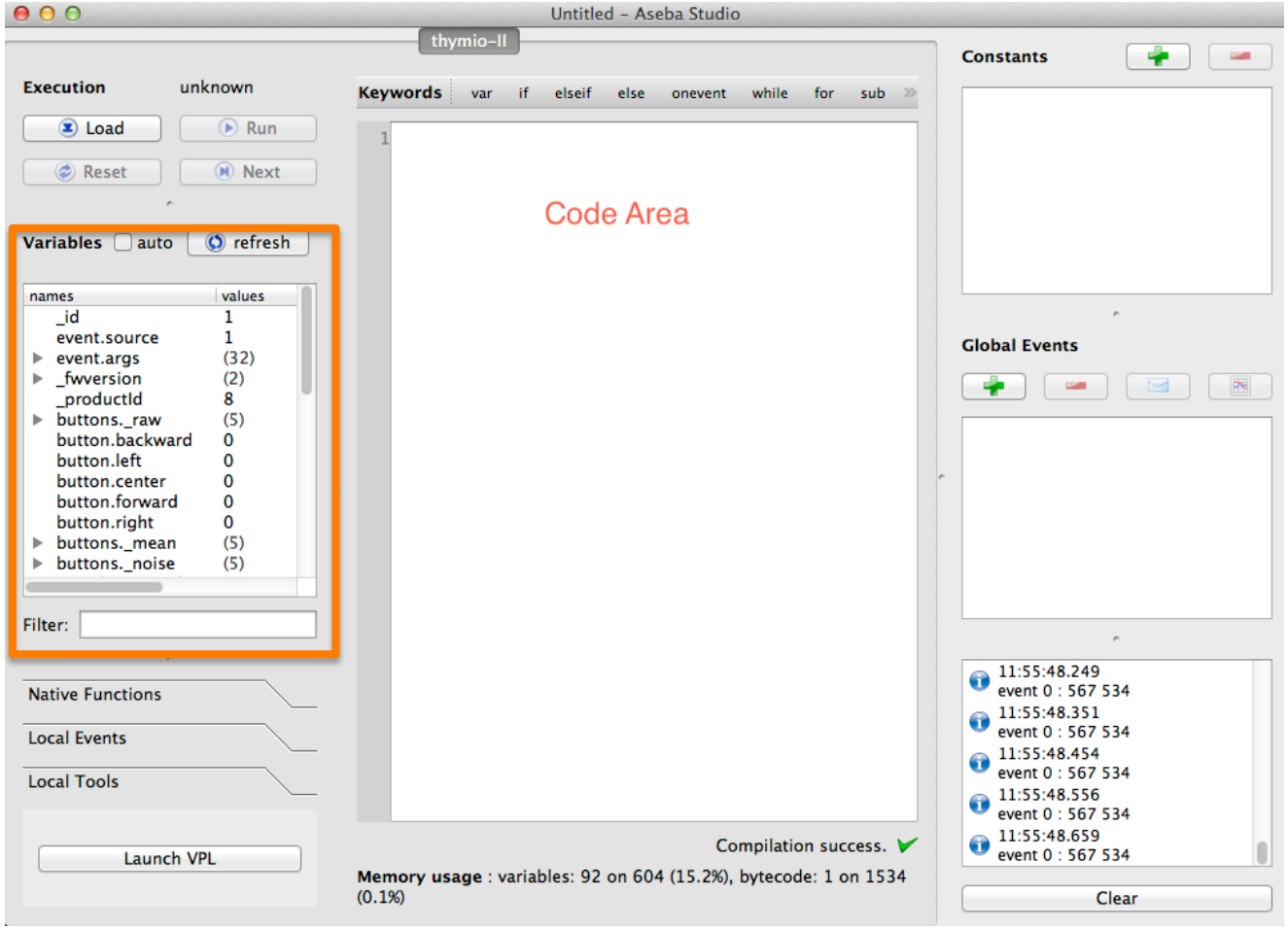
A Sample Program

```
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

Aseba Studio



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

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?

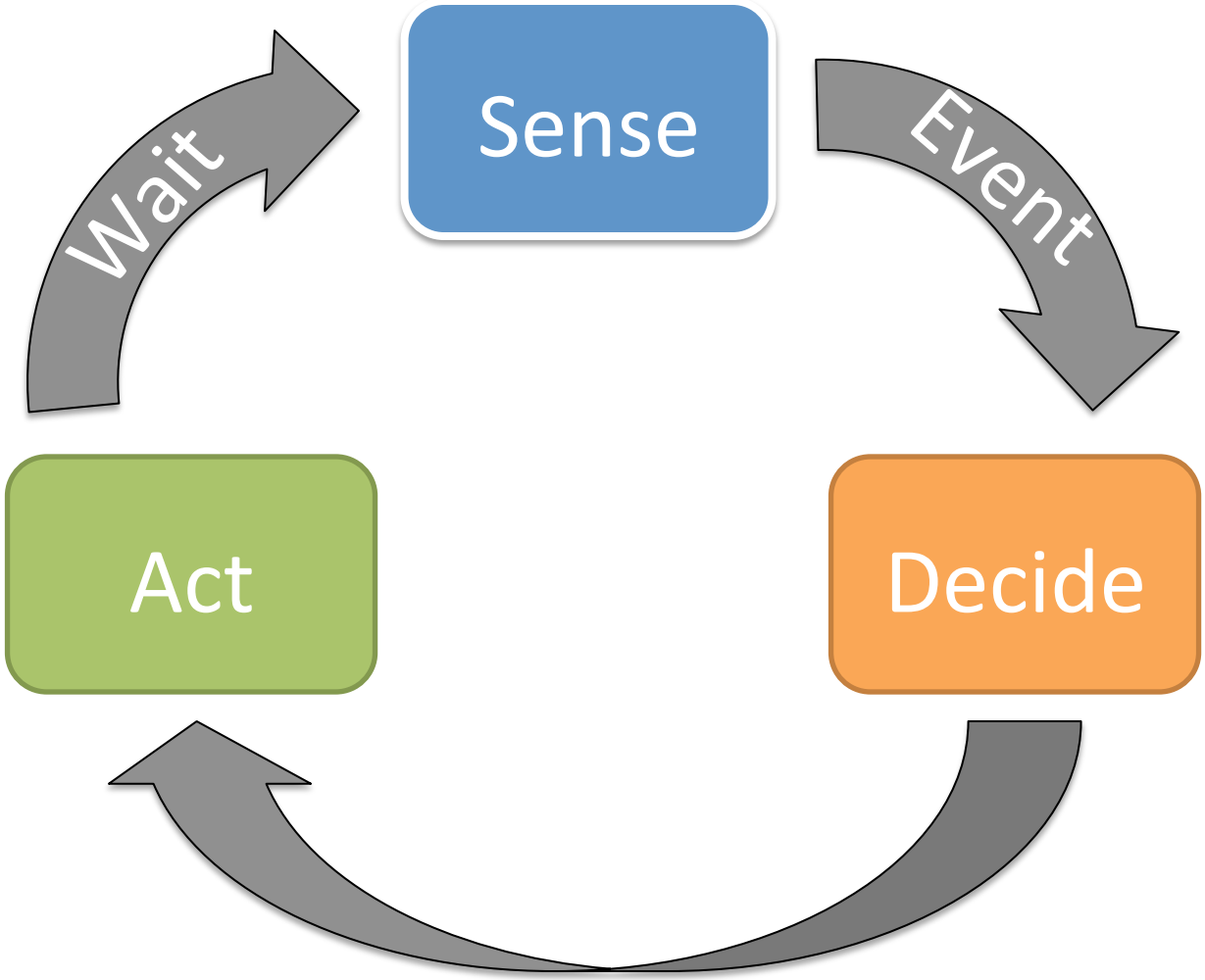
When do We Check the Sensors?

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

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

- **Scratch and Aseba are very similar!**

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



Last Example

```
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
```