



CSCI 1108

Introduction to Experimental Robotics

Intro to ASEBA

Classic Robotics themes

Actuators and movements:

Kinematics –basic movement geometry

without taking mass and forces into account

Differential movements - change in position

Dynamics – differential motions and movement mechanics

Sensor and object recognition:

Computer Vision

Localization:

Kalman filtering, SLAM, etc

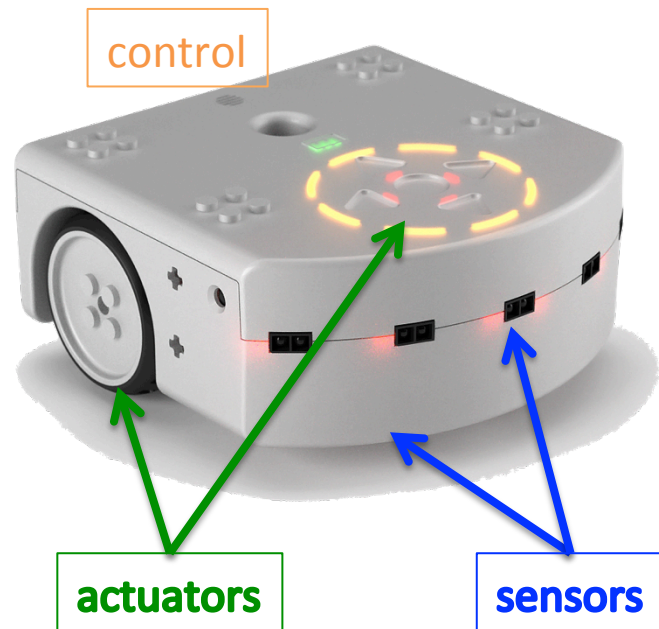
Motion planning

A*, tangent bug, obstacle avoidance, etc

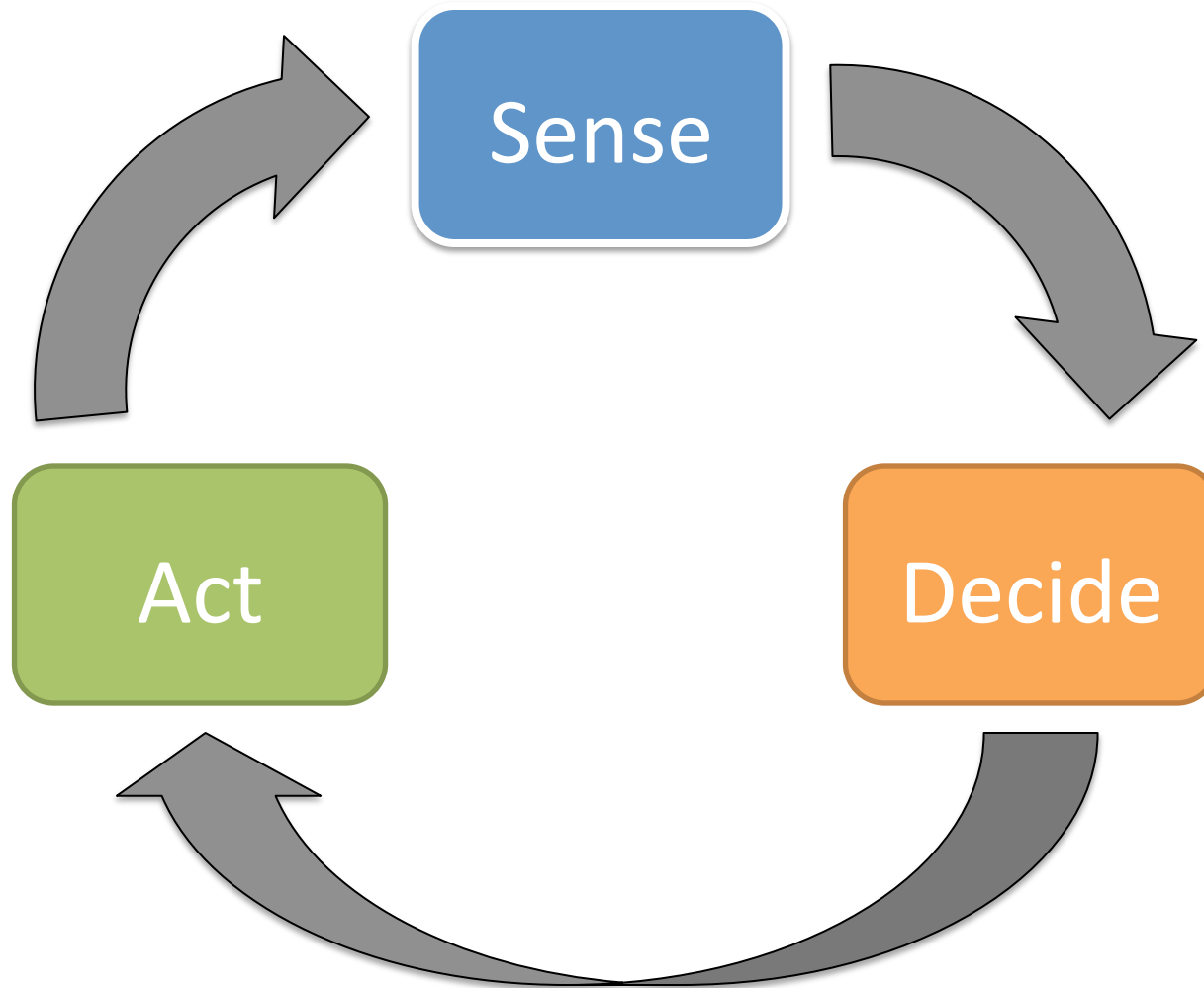


Anatomy of a Robot

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



The Sense-Decide-Act Framework



Controller:

A controller decides what action to take based on input from sensors. Our task is to write a control program for the Thymio II.

This is done in a special programming language called ASEBA

Programming in Aseba

- Programs are text-based
- Programming environment called Aseba Studio
- Key Ideas: Event-based programming
 - Events are triggered by sensors
 - Events are handled by event handlers for which we write the code: **onevent** ...
 - Common programming model for interactive programs (e.g. www, computer interface, etc)

Basic Aseba

- Variables

```
var name
```

```
var list[]
```

- Event Handler

```
onevent prox
```

- Conditional

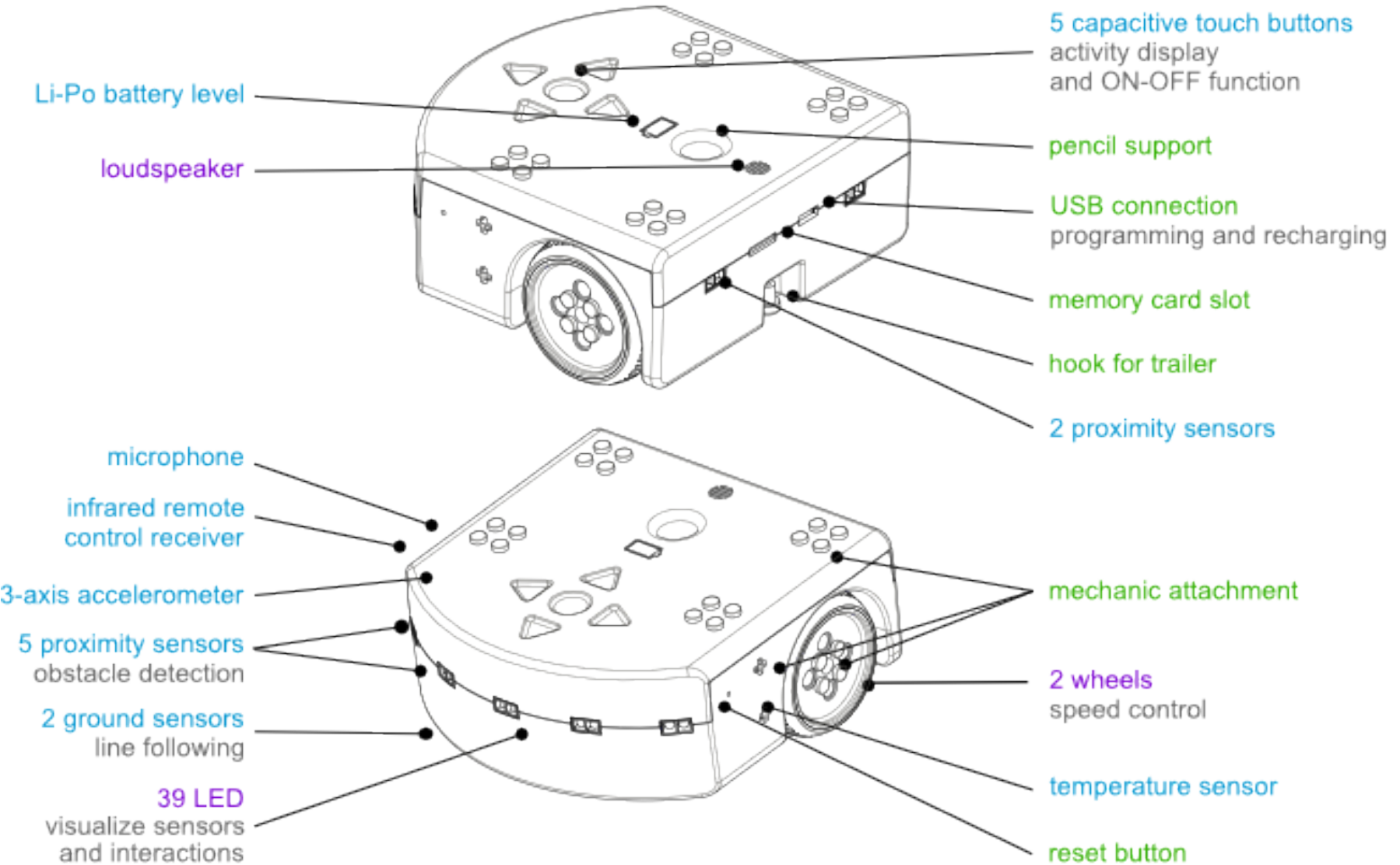
```
if      then
```

```
end
```

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 and Actuators



actuators

sensors

support

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

The screenshot displays the Aseba Studio interface for a thymio-II robot. The main window is titled "Untitled - Aseba Studio" and contains a "Code Area" with the text "Code Area" in red. The interface is divided into several panels:

- Execution:** Includes buttons for "Load", "Run", "Reset", and "Next". The status is "unknown".
- Variables:** A panel with a "refresh" button and a table of variables. The table is highlighted with an orange border.
- Keywords:** A list of keywords: var, if, elseif, else, onevent, while, for, sub.
- Constants:** A panel with a "+" button and a "-" button.
- Global Events:** A panel with a "+" button, a "-" button, and a "Clear" button. It displays a list of events with timestamps and IDs.
- Native Functions, Local Events, Local Tools:** Panels on the left side of the interface.
- Launch VPL:** A button at the bottom left.
- Compilation success:** A green checkmark and the text "Compilation success." at the bottom center.
- Memory usage:** Text at the bottom center: "Memory usage : variables: 92 on 604 (15.2%), bytecode: 1 on 1534 (0.1%)".

names	values
_id	1
event.source	1
▶ event.args	(32)
▶ _fwversion	(2)
▶ _productId	8
▶ buttons._raw	(5)
button.backward	0
button.left	0
button.center	0
button.forward	0
button.right	0
▶ buttons._mean	(5)
▶ buttons._noise	(5)

Global Events:

- 11:55:48.249 event 0 : 567 534
- 11:55:48.351 event 0 : 567 534
- 11:55:48.454 event 0 : 567 534
- 11:55:48.556 event 0 : 567 534
- 11:55:48.659 event 0 : 567 534

<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
- Example: 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
```

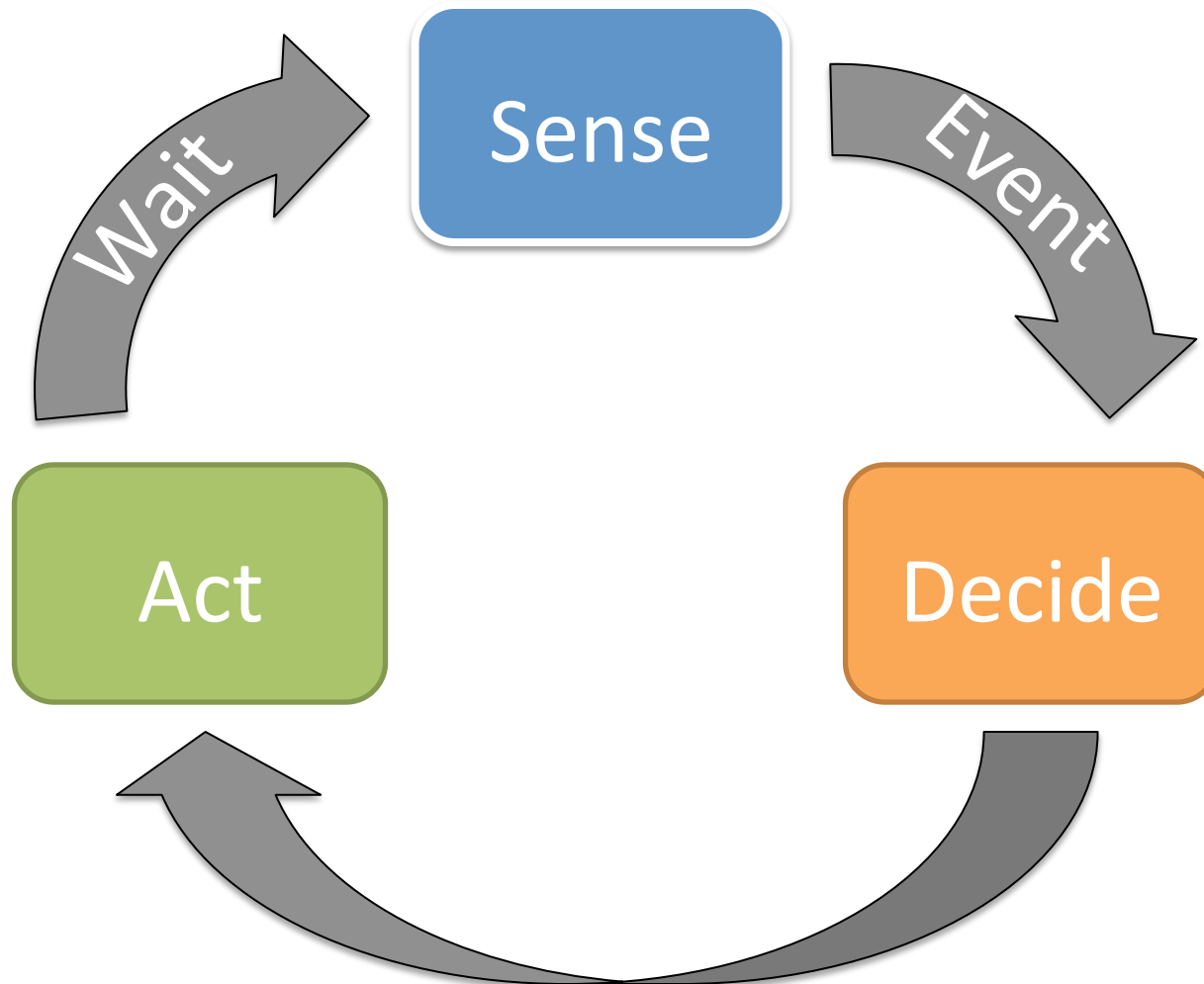
```
    motor.left.target = 100
```

```
    motor.right.target = 100
```

```
end
```

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
```

