Robot “Brain”

SS-3406 Introduction to Robotics

RECAP

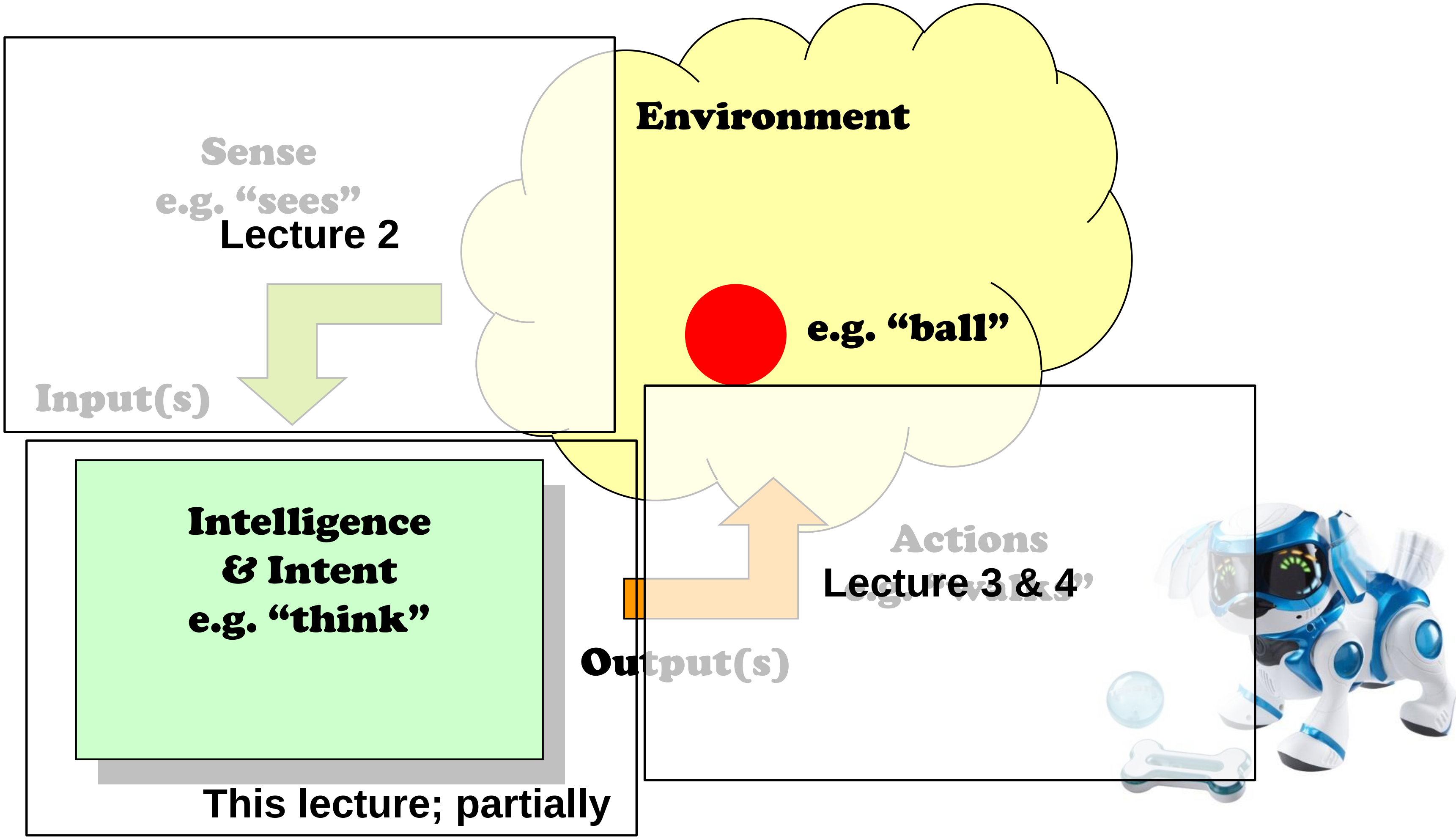
Summary of Prev Lecture

- We were looking at the robot actions.
- Recall that we had looked into robot sensing in earlier lectures.
- We discussed robot actions in two main sub-topics:
  - The hardware involved: actuators and effectors.
  - The actions: locomotion and manipulation.
  - In both types of actions, they involve movement or motion.
- We looked at a few concepts in robot motion:
  - Degree of freedom, Holonomic
  - Kinematic, Dynamics, Trajectory
- On locomotion:
  - Gait, stability
  - We are preferring wheels
- On manipulation:
  - Forward and inverse kinematics
  - We shy away from the dynamics

Today's Menu

- On the robot "brain"
- Microcontrollers
- Robot control concepts
- Robot programming: intro

Properties of a Robot



Two Components of Intelligence

- Learn
  - Gain new knowledge and skills.
- Think
  - Use existing knowledge and skills.
  - E.g. sense, plan, act.
- Today: only the “thinking” part.
  - What does the sensor data mean?
  - What does the current state mean?
  - What should I do now?

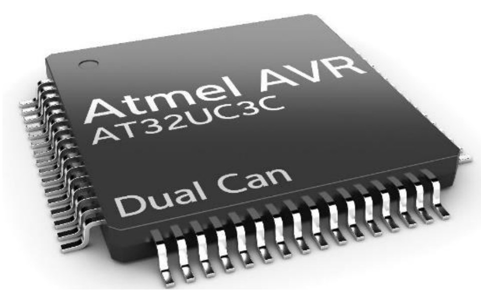
Two Components of Intelligence

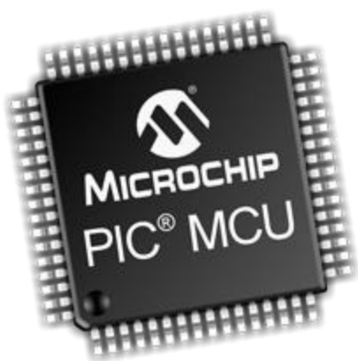
- Learn
  - Gain new knowledge and skills.
- Think
  - Use existing knowledge and skills.
  - E.g. sense, plan, act.
- Today: only the “thinking” part.
  - What does the sensor data mean?
  - What does the current state mean?
  - What should I do now?

THE HARDWARE: MICROCONTROLLERS

“Brain”: Microcontrollers

- A microcontroller, sometimes called microcontroller unit (MCU), is a **computer** on a single integrated circuit (IC) chip, with features:
  - Incorporate most computer functions in a **single chip**.
  - Access to **input and output** (IOs) is essential.
  - Programmed to perform **specific task(s)**.
  - **Embedded** (hence small) in appliances, toys, devices, vehicles, rob0ts.





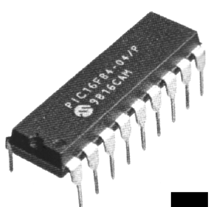
Applications of Microcontrollers

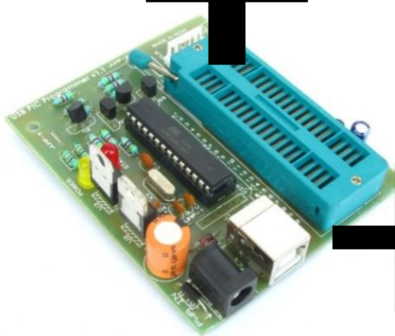
Microcontrollers for Robots

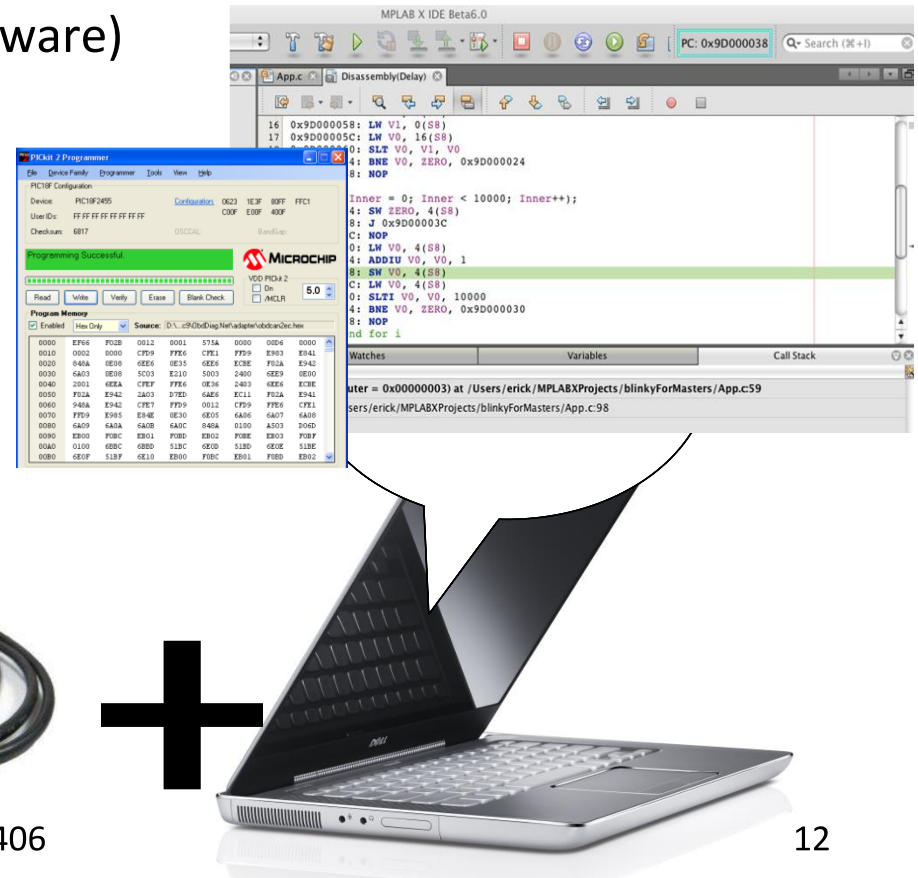
- Popular:
  - Microchip Technology **PIC** (e.g. PIC16, PIC24, etc)
  - Atmel **AVR** (e.g. Atmega, ATtiny, etc)
  - **ARM** technology (e.g. ARM Cortex-A series)
- What to choose?
  - Accessibility: availability, expertise
  - Price: level of complexity required
  - Characteristics, features
    - Available IOs, devices (e.g. timer)
    - Storage capability
    - Processing speed, energy consumption

Programming a Microcontroller

- Three things required:
  - Microcontroller
  - Programming Environment (Software)
  - Programmer (Hardware)



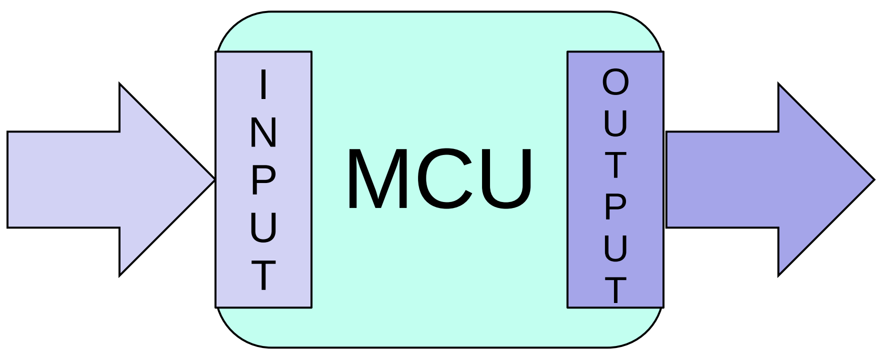




Microcontroller IO Ports

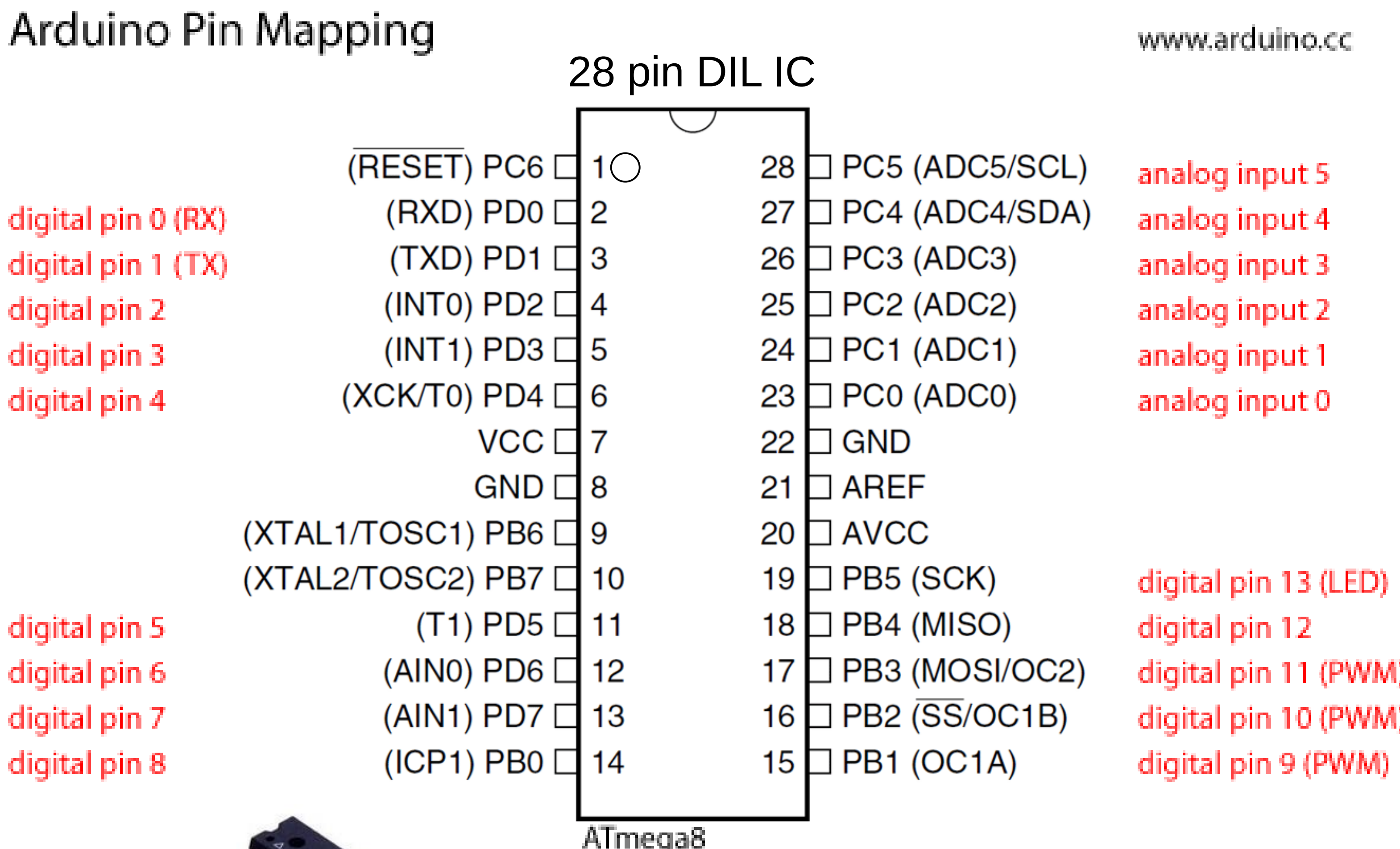
- The inputs and outputs of an MCU are called **port,** and each of them is identified by its **port number** or **pin number**.
  - IOs can be **analog** or **digital**. Digital IOs are easy to use: 1 or 0.
  - An IO **port** often has a number of IO **pins**. In this case, each **port** usually has an **alphabetic** identifier, while each **pin** has an identifier **number**.

Inputs: sensors' data, e.g. switches, are fed in through the **input ports/pins**.



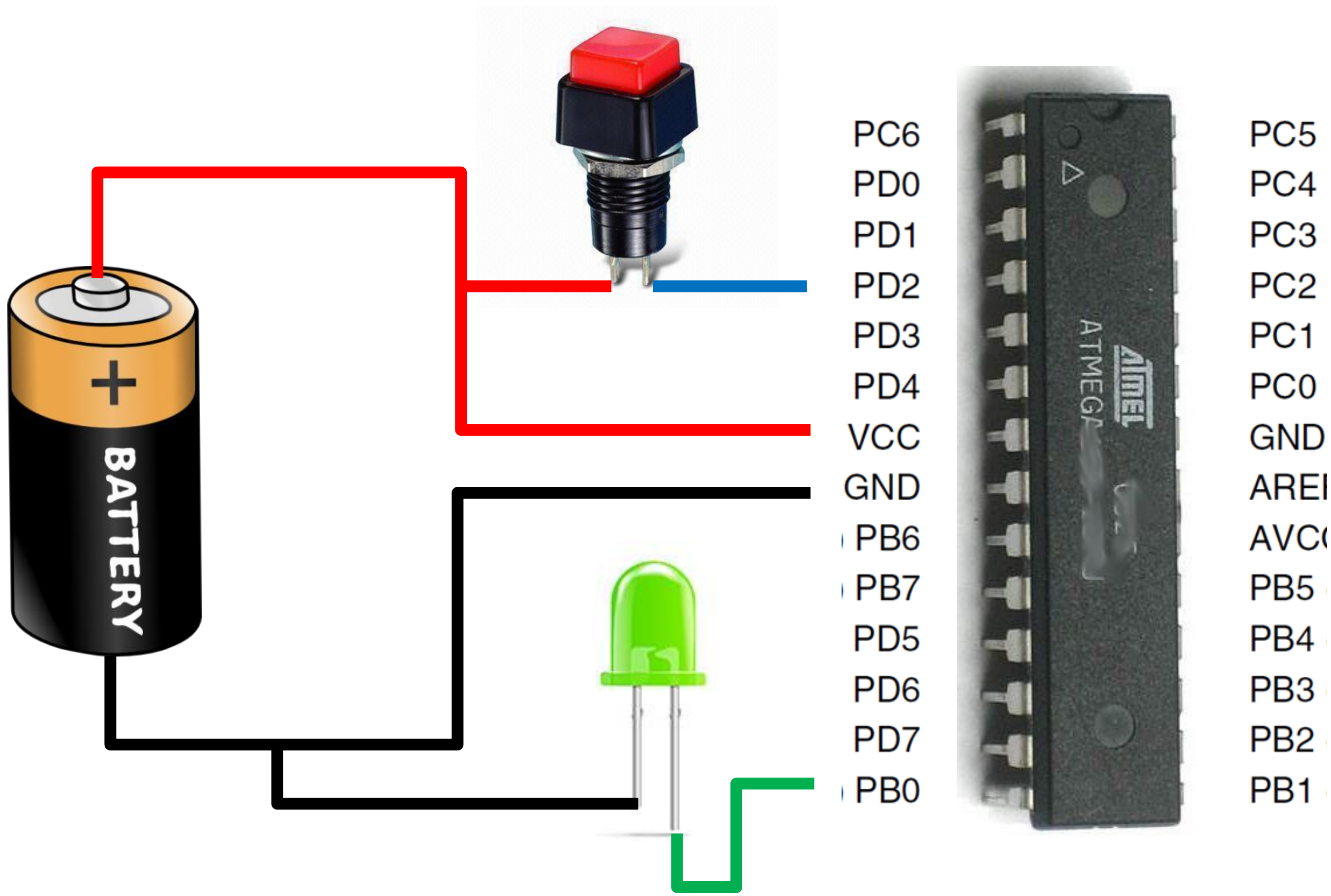
Outputs: signals to activate the actuators, e.g. turn on a light, are sent out through the **output ports/pins**.

Atmel ATmega8 Pin Layout



E.g.
PB1 = Pin 1 of Port B; Pin PB1 or B1.
PC4 = Pin 4 of Port C; Pin PC4 or C4.
PD0 = Pin 0 of Port D; Pin PD0 or D0.

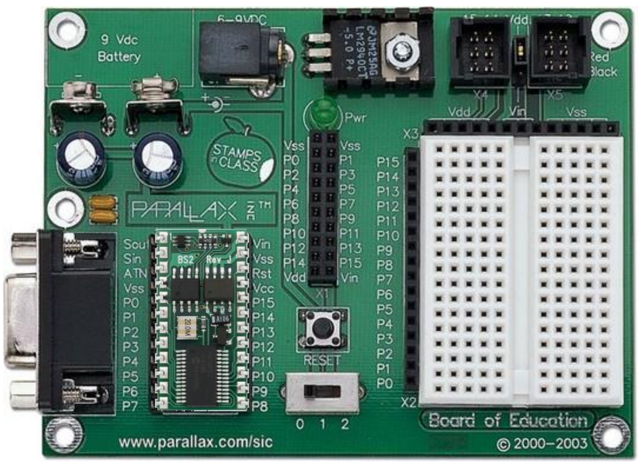
Simple Connection

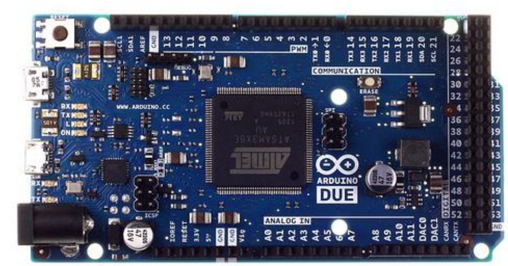


(Note: This circuit has been simplified. A few more components are required to form a practical circuit.)

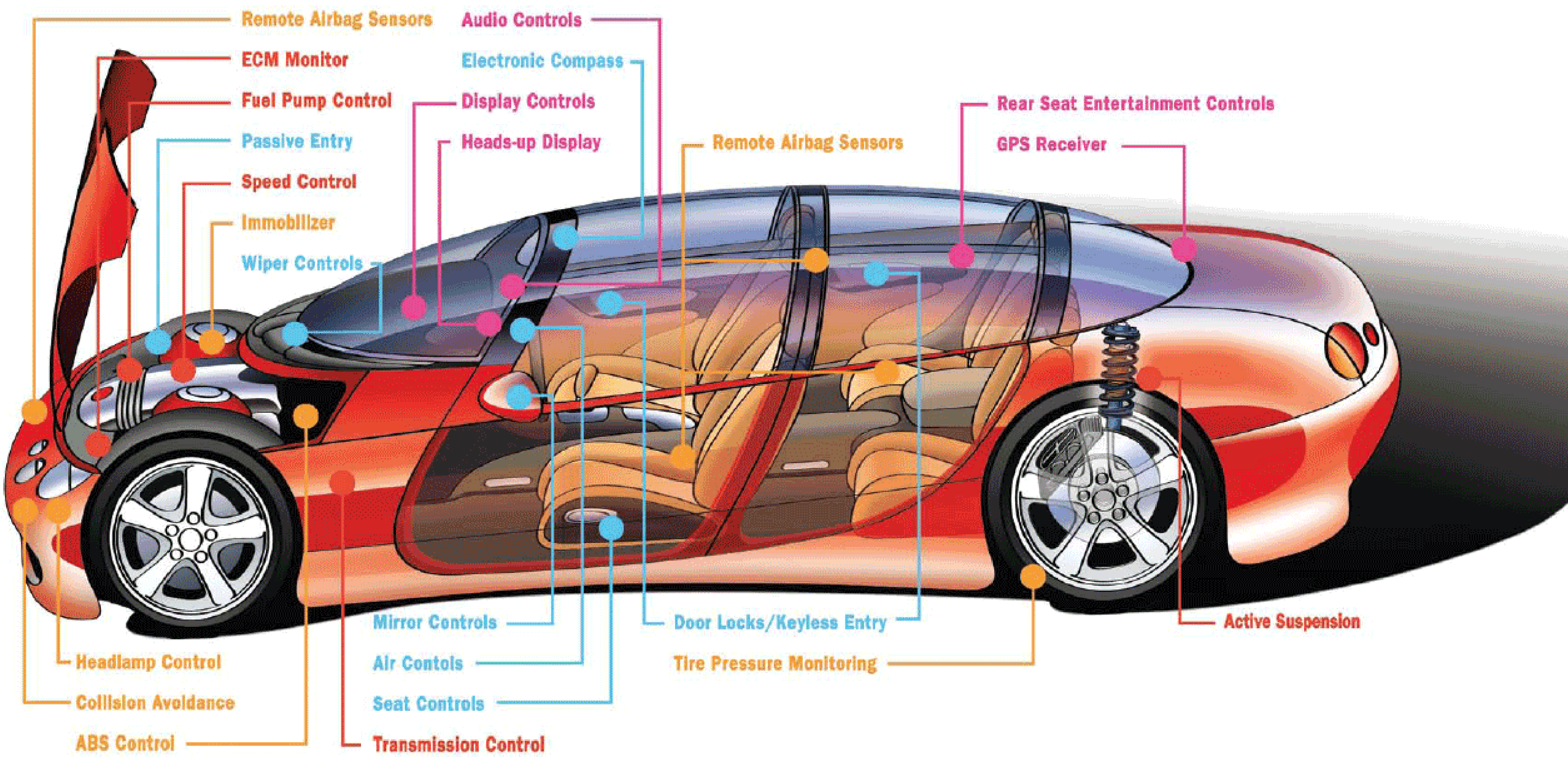
Microcontroller Boards

- Microcontrollers are often produced on a development board that includes IO and programming circuitry, e.g.
  - BASIC Stamp (PIC-based)
  - Arduino Uno (AVR-based)
  - Arduino Due (ARM-based)





Many MCUs in One System



Today's vehicles feature many more MCUs, which control numerous functions. *(Courtesy of Microchip Technology)*

ROBOT CONTROL CONCEPTS

Why is Robot Control Hard?

- High-level, complex goals
  - Impossible to "formulate" solution.
  - Chasing the cat versus solving an equation.
- Dynamic (changing) environment
- Robot has dynamic constraints of its own (don't fall over)
- Sensor noise and uncertainty
- Unexpected events (collisions, dropped objects, etc.)

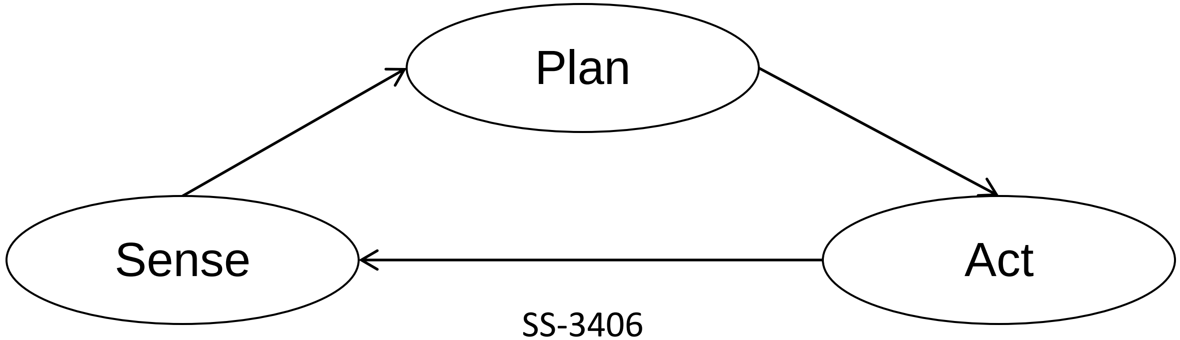
Ref: Coste-Maniere and Simmons (ICRA 2000)

Robot Control Architectures

- Deliberative (also called hierarchical)
  - Sense-Plan-Act
  - "Top-down" approach: from goals, break into subtasks.
- Reactive
  - Sense-Act
  - May not think.
- Hybrid
  - Deliberative at high level; reactive at low level.
- Behavioral
  - Multiple "hybrid" (or other architectures in above) modules.
  - "Bottom-up" approach: compose high level behaviors from lots of independent low-level behavior modules.

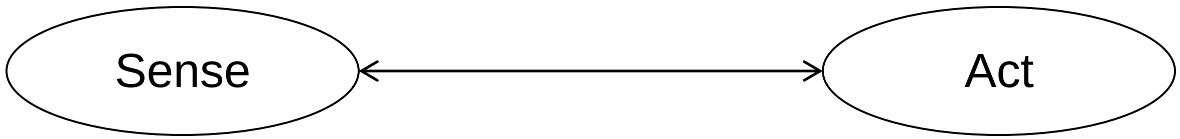
1. Deliberative Control

- Think hard, then act: **Sense-Plan-Act** (SPA)
  - All the sensing data tends to be gathered into one global world model (state).
  - Look into the future (plan) before deciding how to act, i.e. evaluate all possible actions to **avoid bad actions**.
  - Take time, i.e. **slow**. Thinking too long can be dangerous, e.g. fall at a cliff.
- Useful in tasks that can take time to plan ahead, e.g. playing chess, determine the route to a destination, when stealing.



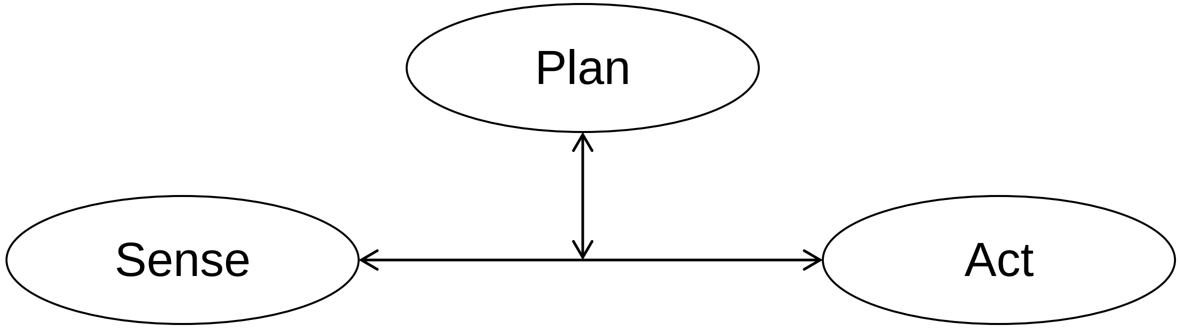
2. Reactive Control

- Don't think, react: **Sense-Act**.
  - Many animals are largely reactive.
  - Tight couple between sensory inputs and effector outputs.
  - Respond very quickly, i.e. **fast**.
  - Do not look into the past or the future.
  - No ability to learn over time, **cannot deal with complex situations**.
- Use in tasks that require very quick action, e.g. when touched a hot surface, when someone tab your shoulder while you are stealing.



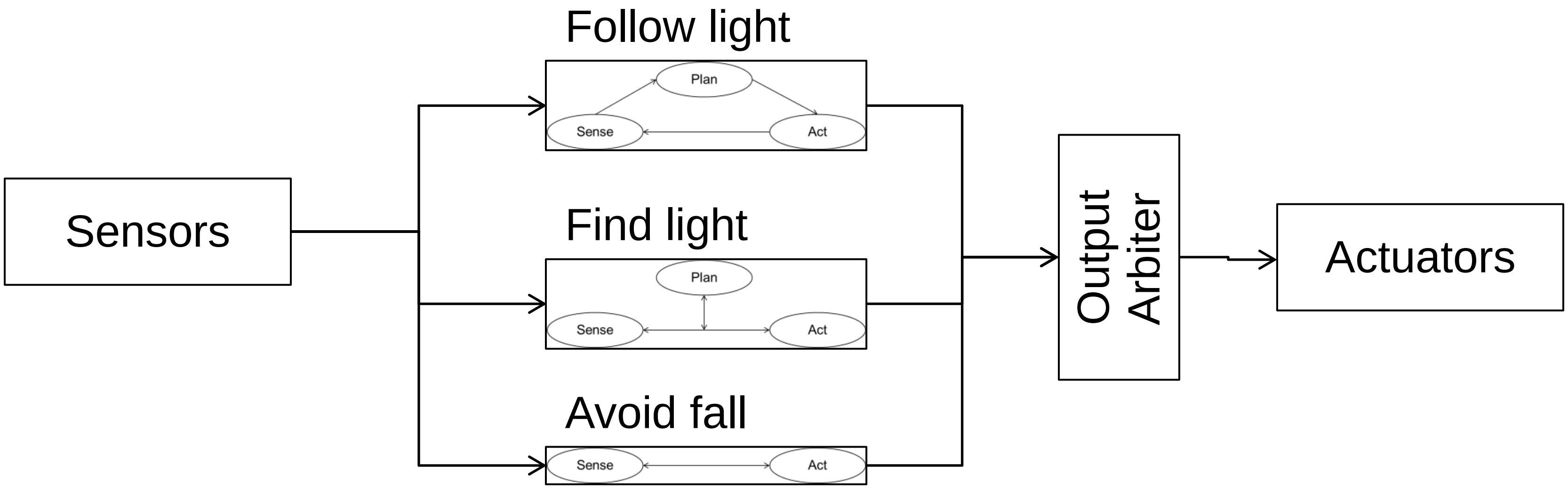
3. Hybrid Control

- Think and act independently, in parallel.
  - Combine the best of both Reactive and Deliberative control.
  - **Different parts** of the "brain": plan some tasks (e.g. the route to distance), while react quickly in some tasks (e.g. avoid fall).
  - Sensors' data are routed to reactive and deliberative parts of the brain.



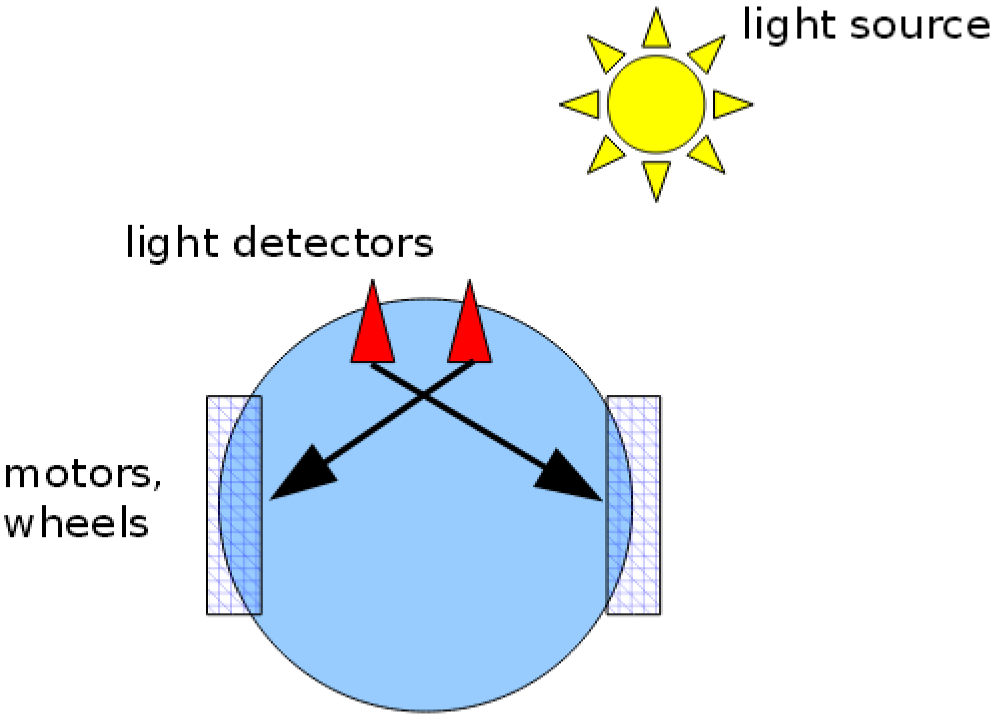
4. Behavior-based Control

- Overall behavior is composed from a network of primitive behaviors.
  - Examples of primitive behaviors: avoid fall, avoid obstacle, follow light, find light.
- Each behavior module is a control unit based on either deliberative, reactive or hybrid architecture.



Which control architecture?

- Light chasing robot
  - Moves towards the light source.
  - Light detectors' output drives the wheels.



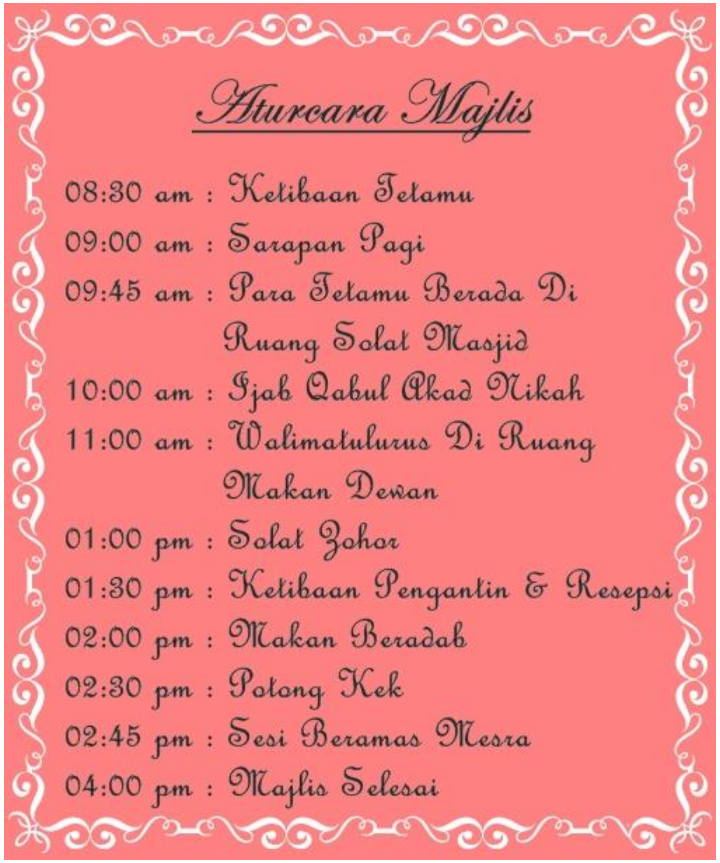
Implementing the robot control in the MCU …

THE SOFTWARE: ROBOT PROGRAMMING

Programming?!

Program

- A sequence of events to follow.



Aturcara Majlis

08:30 am : Ketibaan Tetamu
09:00 am : Sarapan Pagi
09:45 am : Para Tetamu Berada Di Ruang Solat Masjid
10:00 am : Ijab Qabul Akad Nikah
11:00 am : Walimatulurus Di Ruang Makan Dewan
01:00 pm : Solat Zohor
01:30 pm : Ketibaan Pengantin & Resepsi
02:00 pm : Makan Beradab
02:30 pm : Potong Kek
02:45 pm : Sesi Beramas Mesra
04:00 pm : Majlis Selesai

Event Programme

9.00-9:30 *Arrival and Registration*

9:30 Welcome - Councillor Ken Meeson, Leader Solihull MBC

Health and the Local Economy
Melanie Mills, Chief Executive, SEWM
Meet our CCG Social Value Champions

9.45 **Social Value - Outlining the Opportunity**
Chris White MP for Warwick and Leamington, Author of the Public Services (Social Value) Act

10:30 **Q & A Panel - Policy in Action**

11:00 *Coffee and networking*

11:30 **Social Value in Local Authority Procurement**
Geoff Walker, CEO, Sandwell Community Caring Trust

12:00 **Social Value in Health Commissioning**
Arden CSU and The Young Foundation - Healthy Living Network
Wendy Lane, Mary Parkes & Sonya Johnson

12:30 **Social Value and Public Service Transformation**
Rashid Bhayat, CEO, Positive Youth Foundation
Deborah Harrold, CEO, Agewell UK

1:00 **Social Value and Evidencing Health Outcomes**
Building Better Health Partnerships
Mark Ellerby, Director, Cloudberry
Suzanne Callen, CEO, Summit House

1:30-2:30 **Social Food Fair**
Put Your Money Where Your Mouth Is Campaign

2:30-4:00 **Masterclasses**

Networking till Close

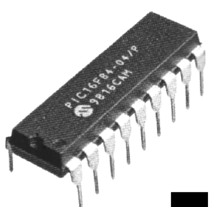
Robot Program

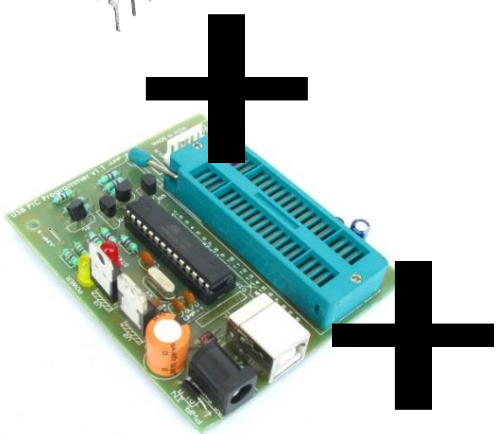
- Remember MCUs are computer?
  - So, a robot program is a computer program
- **Computer program** is a **sequence of instructions** for a computer (MCU) to follow.
  - Note instructions lead to events.

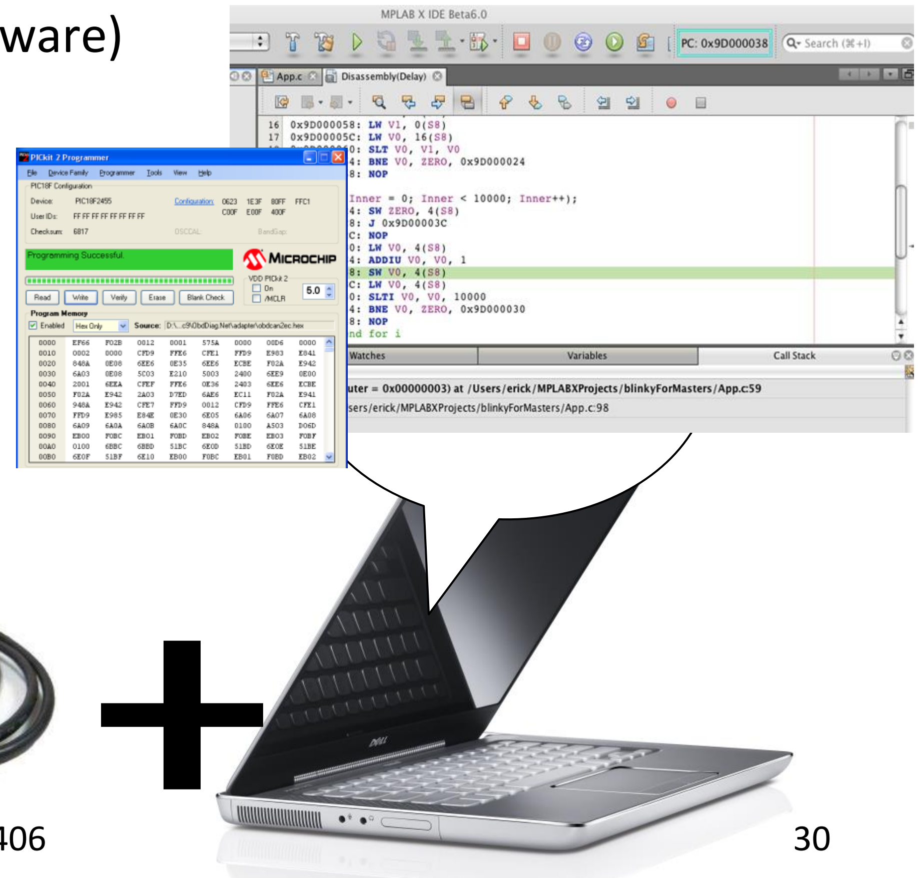
```
task main()
{
  bMotorReflected[port2]=1;
  while(true)
  {
    if(SensorValue(bumper)==0)
    {
      motor[port3]=127;
      motor[port2]=127;
    }

    else
    {
      motor[port3]=127;
      motor[port2]=-127;
      wait1Msec(1500);
    }
  }
}
```

Programming a Microcontroller

- Three things required:
  - Microcontroller
  - Programming Environment (Software)
  - Programmer (Hardware)



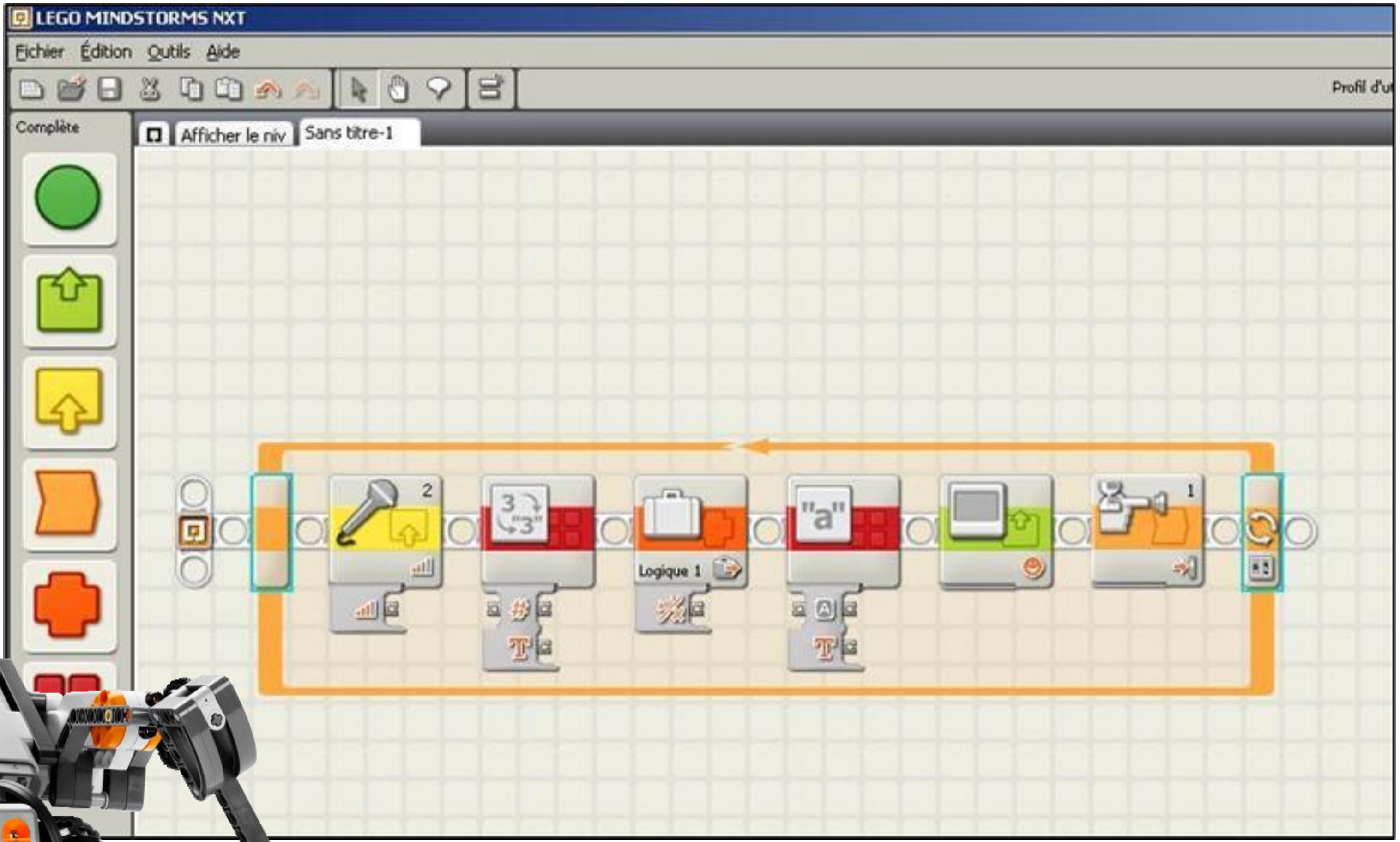




Programming Environment

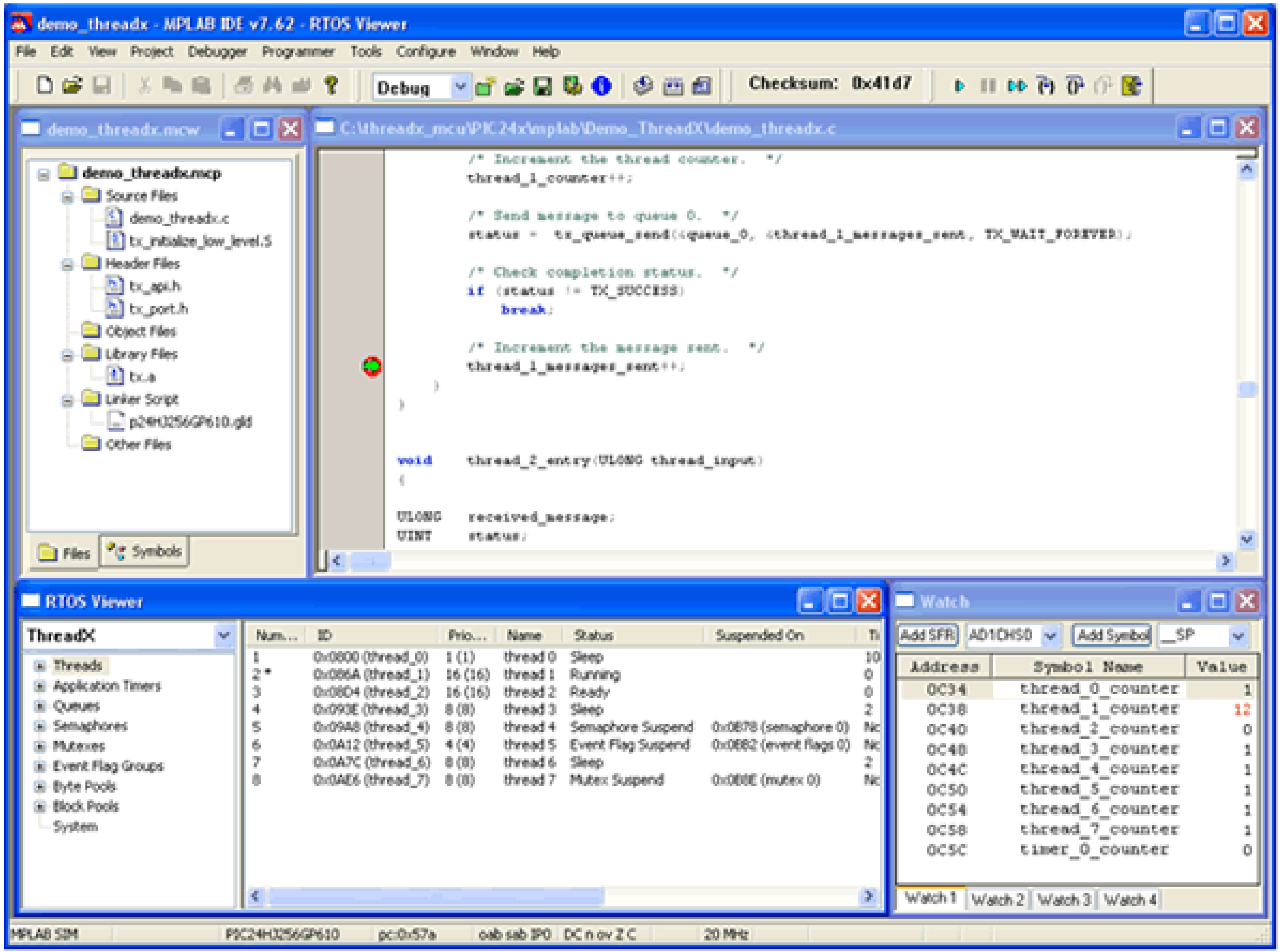
- Integrated Development Environment (IDE)
  - Software to write/develop your program.
    - IDE to write program versus MS Word to write letter.
  - Usually has:
    - Source code editor
    - Build automation tools (compiler, download to MCU)
    - Debugger (find errors)
- Two variants:
  - Visual Programming
    - Create the program by moving programming, building blocks, or code nodes to create flowcharts or structure diagrams.
  - Source Code Programming
    - Write instructions in the program based on specific language.

IDE: Visual Programming



E.g. NXT-G for Lego Mindstorms

IDE: Source Code Programming



E.g. MPLab for PIC MCUs

Programming Languages

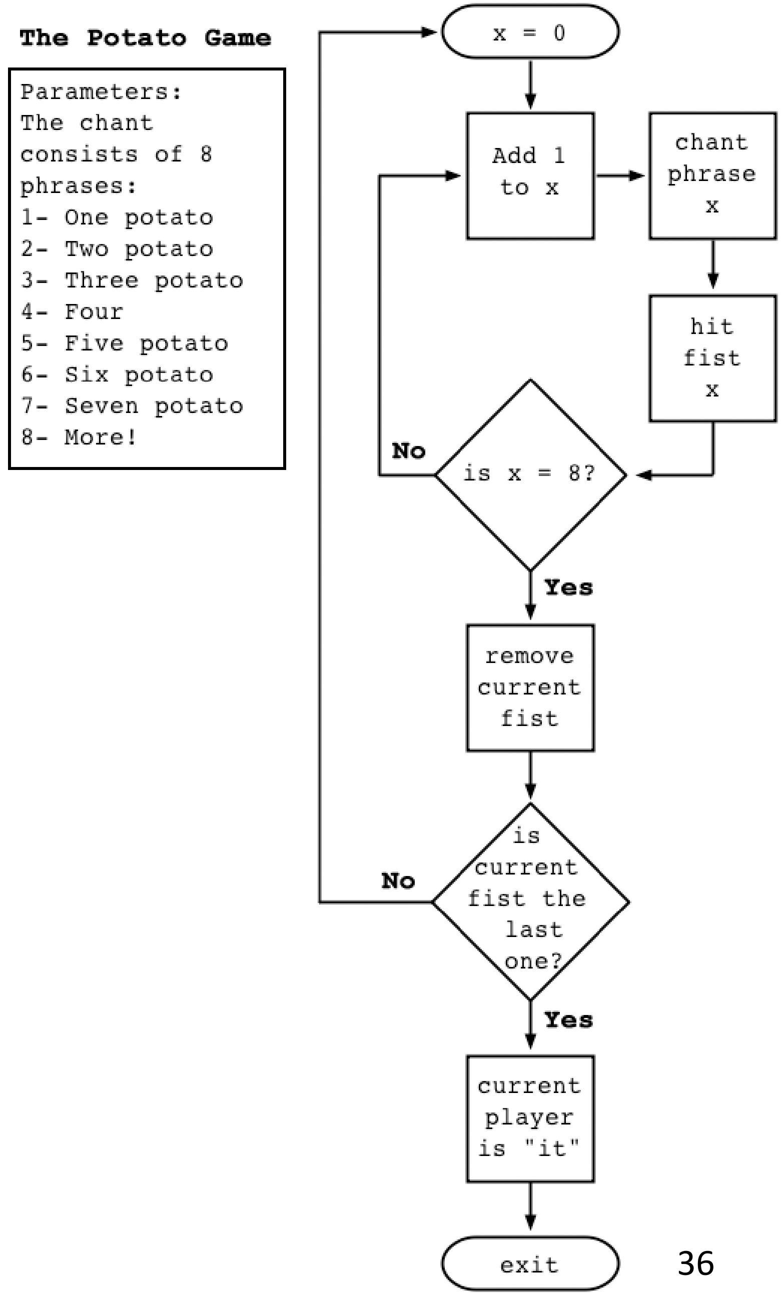
- Different MCU understand instructions in different languages.
  - Use the correct language to give instructions, i.e. write program, to the MCU.
- Some MCUs have IDE that allow the use of a generic high-level programming language.
  - The “compiler” in the IDE “translate” the generic language into one that the MCU understands.
  - E.g. mikroC PRO for PIC
  - Note: generic high-level programming languages e.g. C, C++, C#, Java, Python.

Programming in Five Steps

1. What? What exactly do you want to program?
   – E.g. Robot to follow a white line.
2. How? Design the program.
   – Determine program logic (flow).
   – Design details using flowchart and/or pseudocode.
3. Write it. Code the program.
   – Know the language, know the IDE.
4. Test the program. Debugging.
5. Document and maintain.

Flowchart

- Graphical illustration (diagram) of the program logic, i.e. flow of the sequence of events or instructions.



Pseudocode

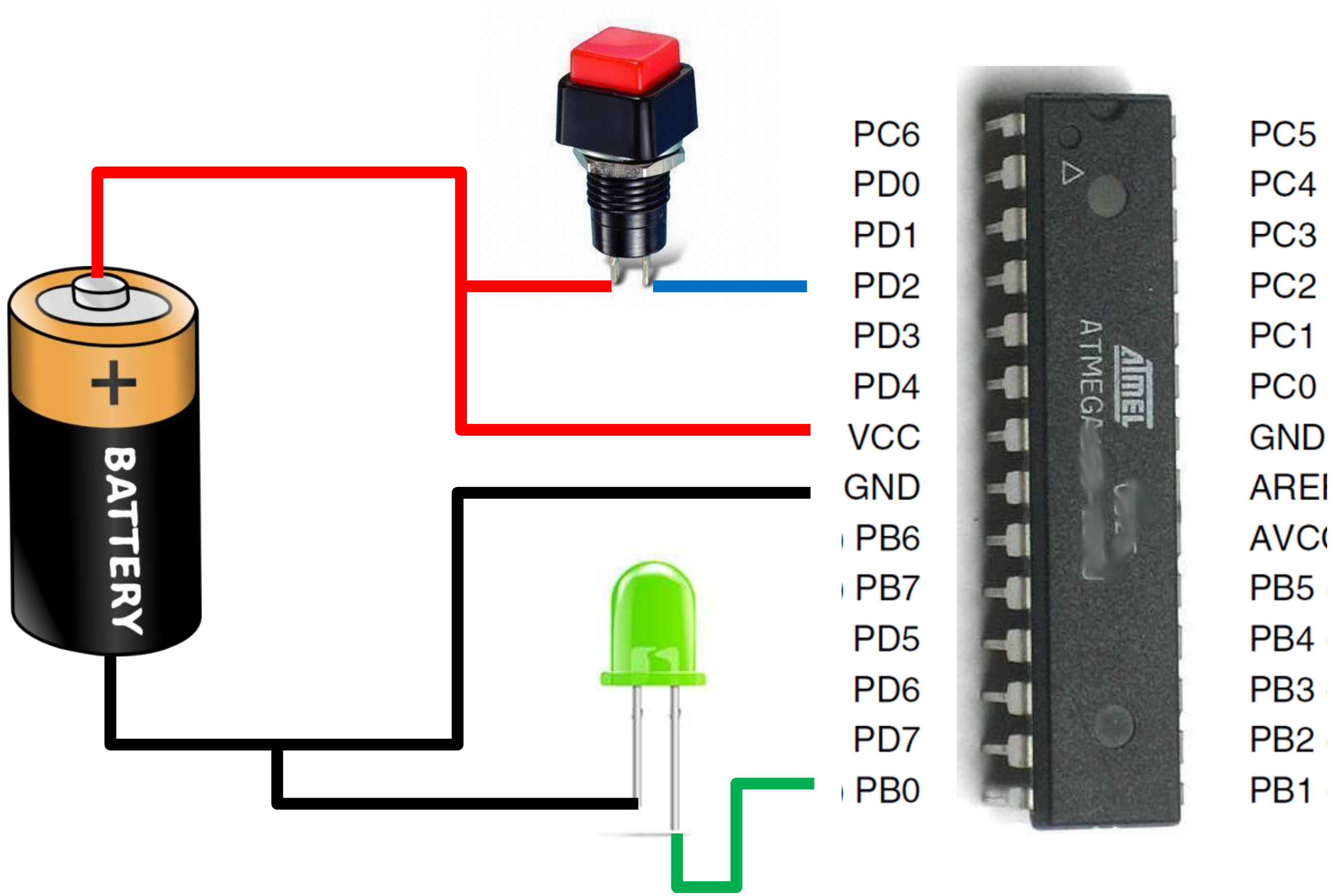
- Normal language (our language) statements to describe the program logic, i.e. the flow of the sequence of events or instructions.
- Translates our thinking to program.

```
To play "One Potato, Two Potato":
Gather all players in a circle
Players put both fists in the circle
Choose a player to be the counter
The counter begins chanting
He repeats until one fist is left:
[
The counter repeats 8 times:
  [Hit one fist
   If 1-3 or 5-7 say count + "potato"
   If count is 4 say "Four!"
   If count is 8:
    [Say "More!":
     Current fist is taken out
     Restart chant on next fist]
   If count ≠ 8 add 1 to count]
if there is only one fist left:
   that player is "it"
] End
```

Programming Concepts

- **Sequence** of instructions
  - Get the order correct.
- Program **flow control**
  - **Conditional structure**: do certain things based on a true or false, yes or no decision.
  - **Looping structure**: a list of instructions to do more than once.
- **Program structure** – language specific
  - The template: different sections of the source code.
- Program **syntax** – language specific
  - Instructions, symbols and statements in the source code.
- How to deal with the **data**: **variables**, **constant**, **data structures**.(e.g. array).

Pseudocode: Flash LED



(Note: This circuit has been simplified. A few more components are required to form a practical circuit.)

Pseudocodes: Flash LED

- High Level:

```
1. Loop:
   1. If Red switch is pressed and released.
      1. Flash LED
```

- At a lower level:

```
1. Loop:
   1. If PD2 receives a 1 and then 0
      1. Send timed string of 0 and 1 to PB0
```

Reading List

- Introduction to Microcontrollers on Society of Robots.
  - http://www.societyofrobots.com/microcontroller_tutorial.shtml
- Microcontrollers and Robotics.
  - http://home.roboticlab.eu/en/microcontrollers

To Do List

- For the simple circuit shown on Slide 14/38, write pseudocode at a lower level to perform the following:
  - Turn on the LED when the switch is pressed and let it turn on when the switch is released.
  - Toggle the LED on/off when the switch is pressed and released.
- Do the preparation work described in the slides in Programming Part I available on the Moodle under Practical section.
  - Download and extract the relevant files.

Summary

- **Intelligence**: **think** and **learn**. We looked briefly on the thinking.
- Brain of a robot: **microcontroller** (MCU).
- MCU are: **IC** + **board**. MCU have: **IO pins**.
- We learned briefly how sensors are **connected** to input pins and actuators are connected to output pins.
- A system, e.g. robot, may have more than one MCU.
- Robot **control**: **deliberative**, **reactive**, **hybrid**, **behavior-based**.
- **Programming**: MCU + IDE + Computer.
- Start with pseudocode: translating our thinking to programming.