Introduction to Robotics

Chapter 2

Objectives

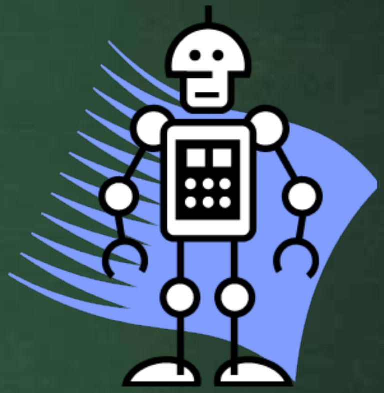
- Understand how robotics fits in to computer science
- Understand some typical uses of robots today and the types of problems addressed in robotics.
- Understand the PropBot v2.0 robot that you will be using in the lab
- Understand the types of problems addressed in robotics

What's in Here ?

- Introduction
  - What is a robot ?
  - Where are They Used ?
- Programming Strategies
  - Approaches
  - Challenges
  - Uncertainty
- PropBot 2.0
  - Original Boe-Bot Kit
  - Its History and Changes
  - Add-On Sensors
  - Power Requirements

Introduction

What is a Robot ?



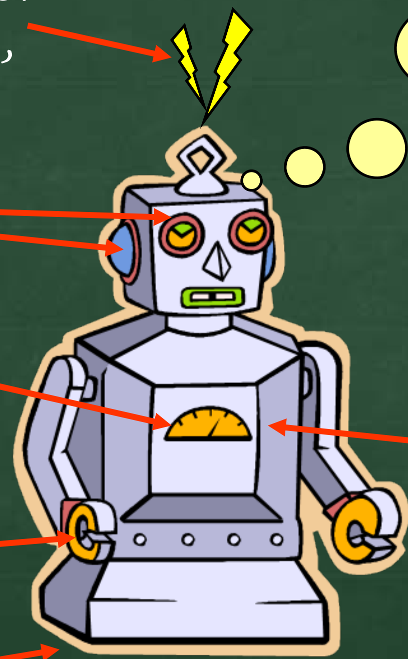
- The definition of a robot can vary greatly.
- Simply put, a *robot* is a device that can move and react to sensory input.
- *Robotics* is the science or study of the technology associated with the design, fabrication, theory, and application of robots.
- Their design can involve many specialized areas:
  - Mechanical Engineering
  - Electrical Engineering
  - Computer Science
  - Cognitive Science (A.I.)
  - Chemistry (nanotechnology)

Robot Components

Wireless Link
to central station, other robots, GPS, etc…

"Brain(s)"
processor, algorithms, strategies, etc…

Sensors
vision, sound, touch, gauges, etc…

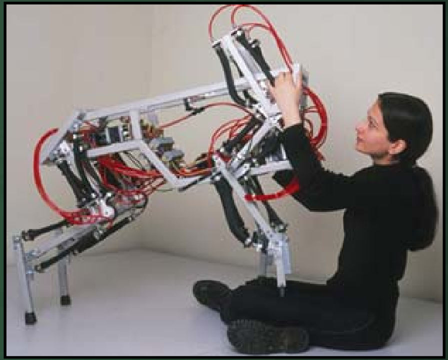


Body
appearance, metal, plastic, "bells & whistles", etc…

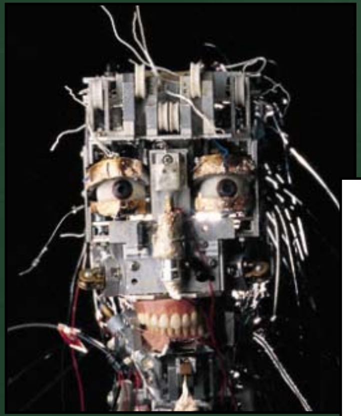
Actuators
motors, hydraulics, pneumatics (air), etc…

Robots – Mechanical Engineering

- Typically studies:
  - designing robot shape & its mechanics
  - issues: efficiency, payload limit, materials
  - walking, climbing, flexible bending
  - biomimetics (mimicking real life design)



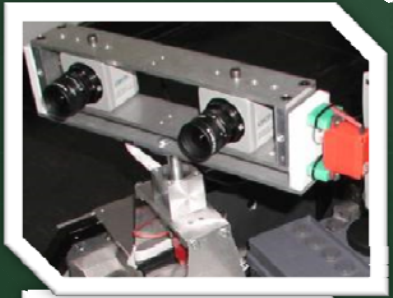






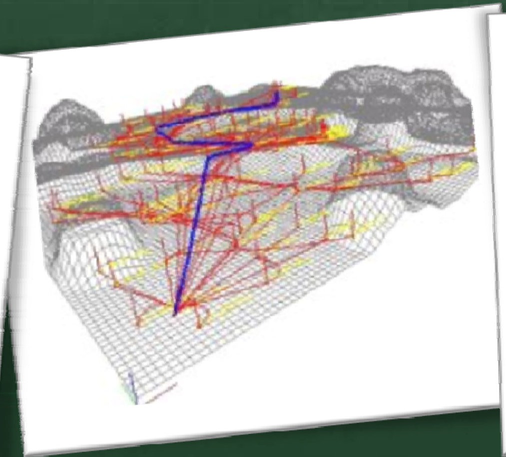
Robots – Electrical Engineering

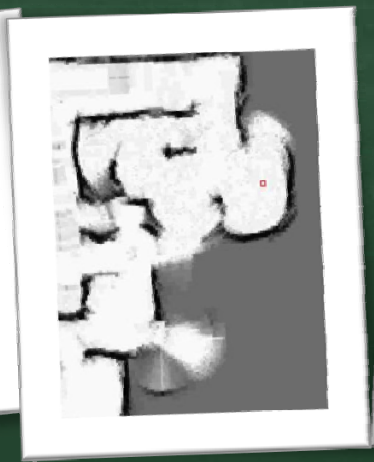
- Typically studies:
  - efficiency issues (power/battery requirements)
  - sensor and actuator (e.g., motor) design
  - wireless communications
  - board design and computer interfacing

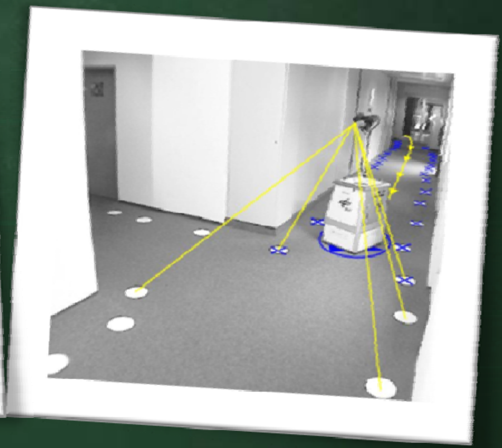


Robots – Computer Science

- Typically studies:
  - deciding what to do with the robot
  - navigation, motion planning, behaviors
  - machine vision, 3D scene reconstruction
  - cooperation and learning strategies

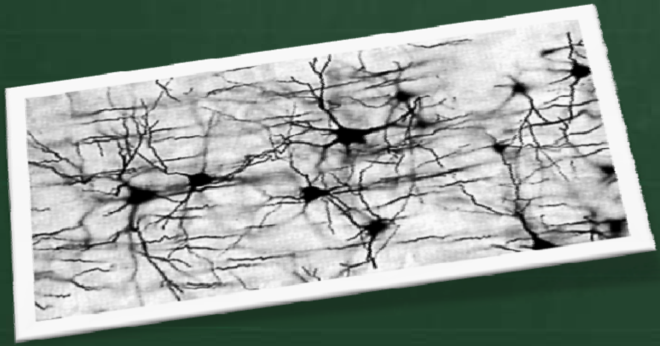






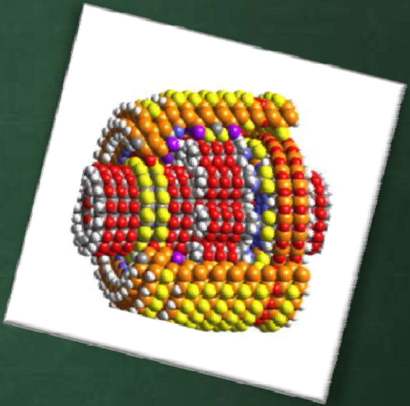
Robots – Cognitive Science

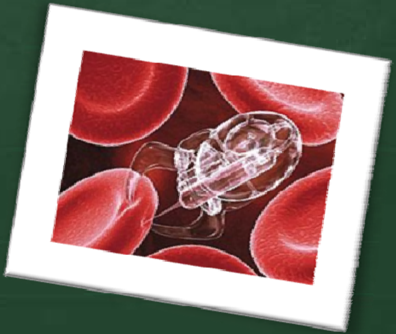
- Typically studies:
  - Artificial Intelligence
  - humanoids
  - connectionism (neural networks)
  - language processing
  - learning and Memory



Robots – Chemistry

- Typically studies:
  - nano-sized robots for nano-applications
  - chemical engineering to produce motors
  - etc…





Where are they used ?

- Robots are now widely used in:
  - factories to perform high-precision jobs

    (e.g., welding, painting, riveting)

  - dangerous locations for humans

    (e.g., cleaning toxic wastes or defusing bombs)

  - home applications

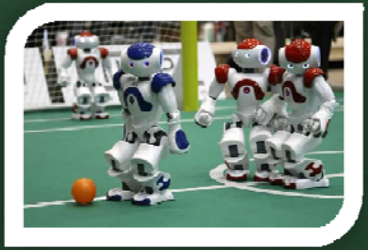
    (e.g., vacuum cleaner, lawn mowing)

  - entertainment

    (e.g., AIBO)

  - competitions

    (e.g., robocup, F.I.R.S.T.)



The Robotics Market

- There are different markets for robotics:

For more information, see
www.roboticstrends.com

Robotics

Industrial Robotics

Articulated robot arms
(e.g., manufacturing, spot welding, painting, assembly)

Service Robotics

Mobile, interactive, autonomous devices

Professional Service Robotics

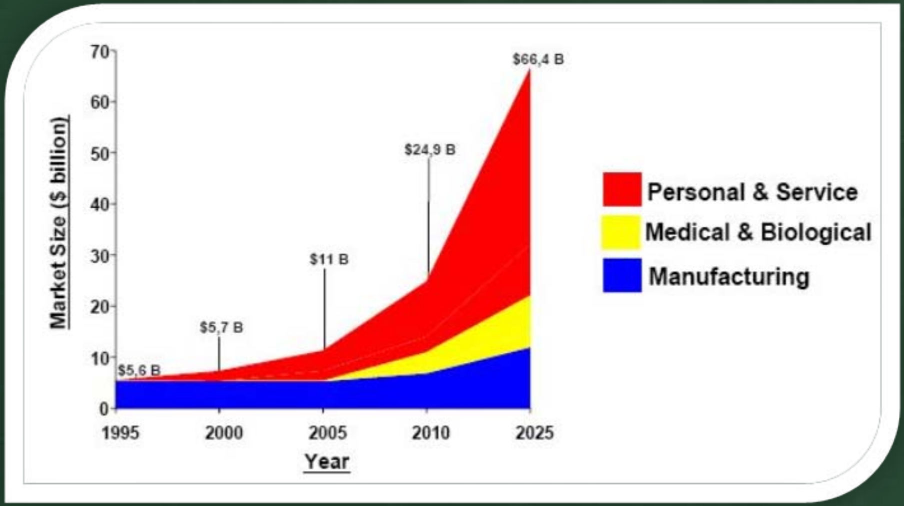
(e.g., industrial cleaning, equipment maintenance / servicing, data acquisition, medical/healthcare, rehabilitation, prosthetic/orthotic devices, mars exploration, inspection, security, construction, demolition, delivery, unmanned ground / underwater / aerial vehicles, rescue, fire fighting, de-mining, surveillance, public safety etc..)

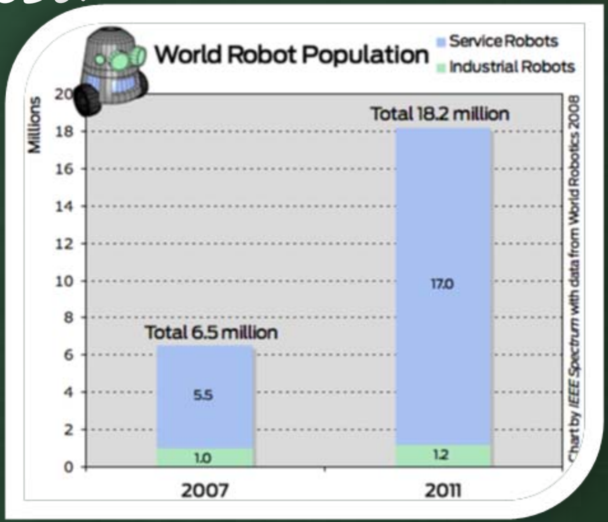
Personal/Domestic/Consumer Service Robotics

Education/Hobbyist/Entertainment – LEGO mindstorms
Home Care – vacuum cleaners, lawn cutting, security
Smart Toys – RobotSapien, Pleo, Aibo
Home Assistance – elderly assistance

The Robotics Market

- Trends are showing that the robotics market is expanding in many areas and projections indicate a dramatic increase in both funding and interest:



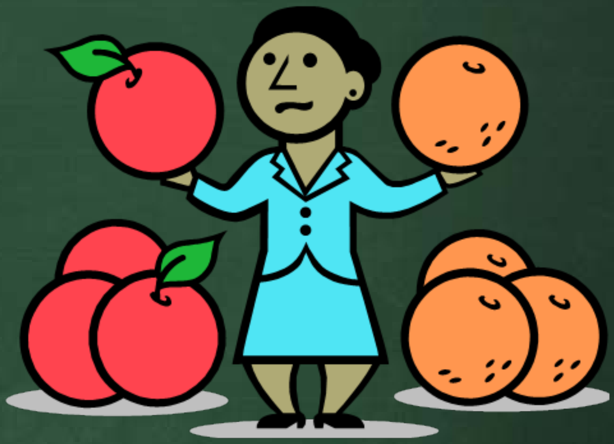


- Many robotics-related jobs will arise, in engineering, computer science, sales & marketing, accounting & operations, business development, public relations, etc..

Programming Strategies

Computer vs. Robotic Programs

- **Computer Programs:**
  - designed to compute an answer
  - data usually valid when available
  - predictable program flow
  - foreseen errors easily handled



- **Robotic Programs:**
  - designed to react to achieve goals, not "an answer"
  - sensors often produce invalid data (or data missing)
  - unpredictable situations due to dynamic environment (e.g., unforeseen obstacles, wrong or missing sensor data, communication outages, hardware failure, etc..)
  - program must degrade gracefully in difficult situations

Robot Programming – Approaches

- There are two fundamental approaches:
  - **Top-down (Classical) approach:**
    - Central processing for overall robot actions.
    - Start with high-level plan of action
    - Hierarchical/sequential processing structure
    - Well-defined functionality
  - **Bottom-up (Behavior-Based) approach:**
    - Independent low-level behaviors
    - Learn basic functionality and build upon them
    - Flat/parallel processing structure
    - More complex tasks accomplished through emergent behavior

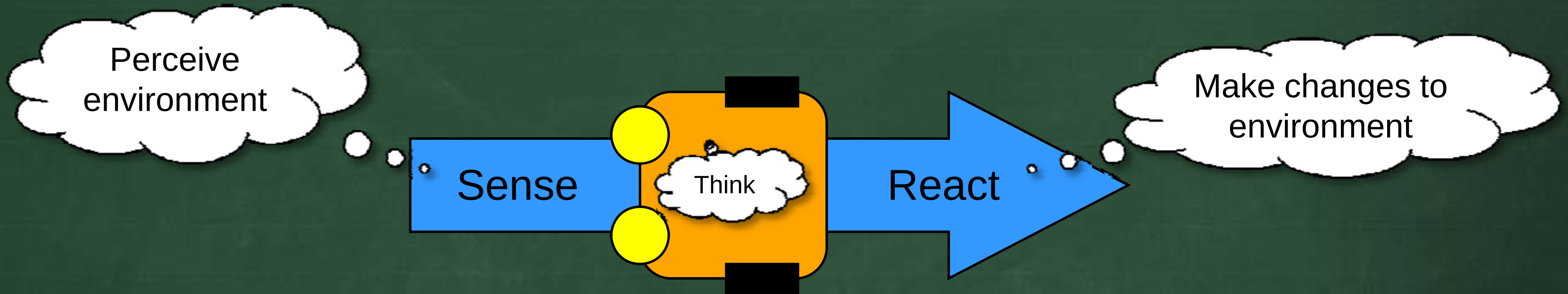
Robot Programming – Compare

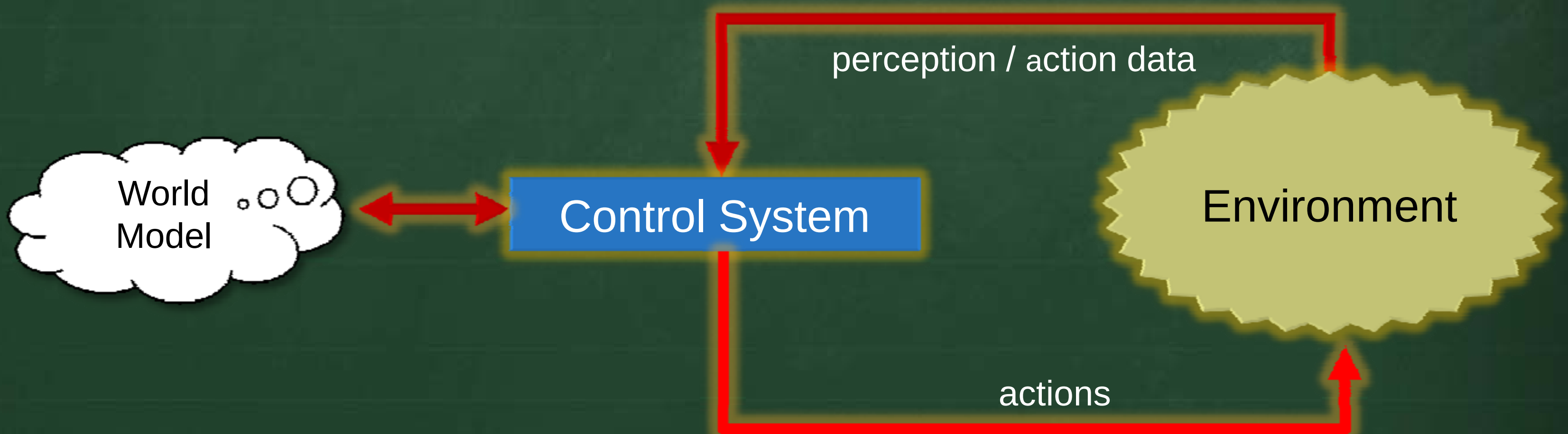
- **Top-down approach:**
  - \+ Easy to assess and re-plan overall goal strategy
  - – Usually complex and fragile code
  - – Hard to handle unforeseen problems
  - – Adding functionality requires more computational power
  - – Slow response time.

- **Bottom-up approach:**
  - \+ Quick response time
  - \+ More robust
  - \+ Simpler to code
  - \+ Easy to handle unforeseen problems
  - – Hard to achieve high-level goals

Robot Processing

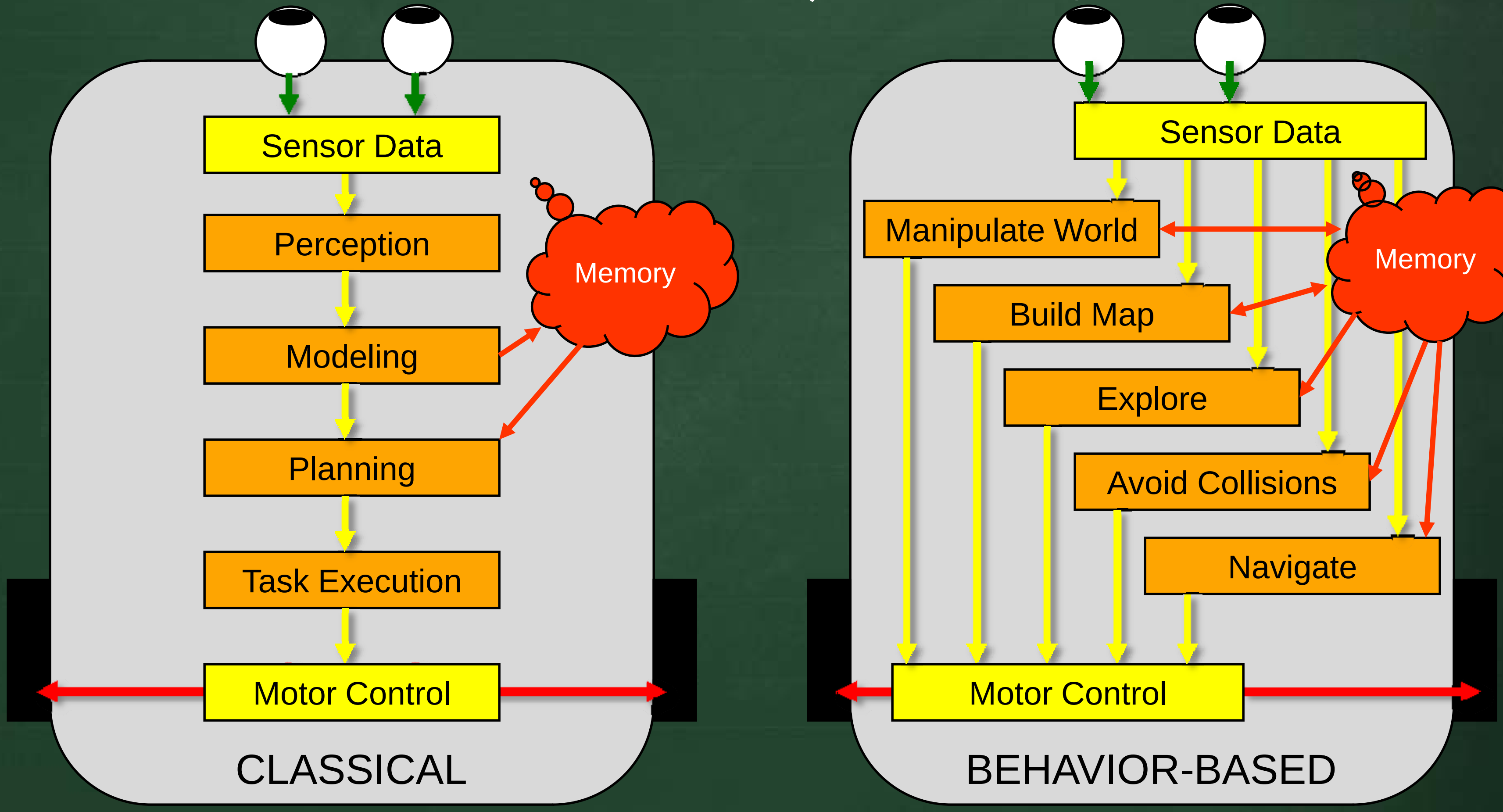
- Here is the basic "flow of control" commonly used:





Robot Processing

- Classical vs. Behavior-Based processing:



Challenges

- Physical/Mechanical/Electrical Issues:
  - sensors are prone to errors and bad readings
  - sensor input requires lots of processing power
  - actuators drain batteries, not small/powerful enough

- Knowledge Representation & Retrieval:
  - difficult to represent real world in robot's memory
  - difficult to "sift though" mounds of data to pick out relevant information
  - real world changes, requires robots to be adaptive

Uncertainty

- There is an enormous amount of uncertainty in a robot's environment.
- Various physical factors contribute to uncertainty:
  - Environment is unpredictable and highly dynamic
  - Sensors are limited:
    - range and resolution limited physically
    - subject to noise
    - can (and will) break
  - Actuators can be unpredictable
    - noise, wear-and-tear, mechanical failure

Uncertainty

- Other factors contribute to uncertainty:
  - Internal models of the environment are approximate.
  - Algorithms are approximate in order to be real-time.

- So what's the big deal ?
- Robots are sometimes forced to act possibly without sufficient information from sensors and internal models.
- They cannot make the right decisions with absolute certainty

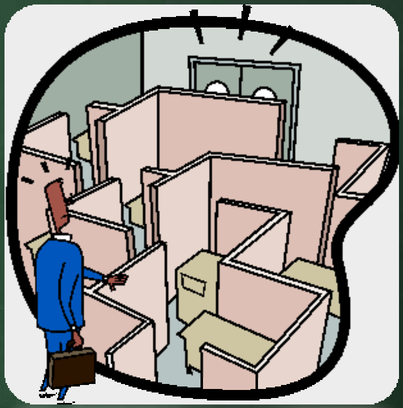
Probabilistic Robotics

- *Probabilistic Robotics*:
  - is a relatively new approach in robotics
  - addresses uncertainty in robot perception and action
  - represents uncertainty explicitly using math:

    Instead of maintaining a "best guess", information is represented by probability distributions over a whole space of guesses.
  - degrades gracefully in the presence of uncertainty
  - outperforms other techniques in many situations

Probabilistic Robotics

- The internal models are integrated with sensor data making it more robust.
- Allows it to scale better to more complex environments.



- The only working solution for mapping and localization in large environments.
- Limitations:
  - Computational complexity (can be less efficient)
  - Need to approximate

What will WE do ?

- We will learn how to:
  - Write small, condensed programs in SPIN
  - Program simple behaviors in a robot
  - Estimate a robot's position as it moves
  - Use various sensors and cope with their inefficiencies
  - Extract map features and create maps
  - Analyze and merge sensor data
  - Understand simple 3D vision
  - Navigate in a 2D environment
  - Understand the pain (yet joy) of working with a real robot

What we will NOT do ?

- We will not learn:
  - How to design and build robots
  - How to program different types of robots
  - Artificial Intelligence
  - Team robotics strategies
  - Complicated machine vision and 3D reconstruction
  - Genetic Algorithms and other forms of learning
  - How to perform particular tasks.

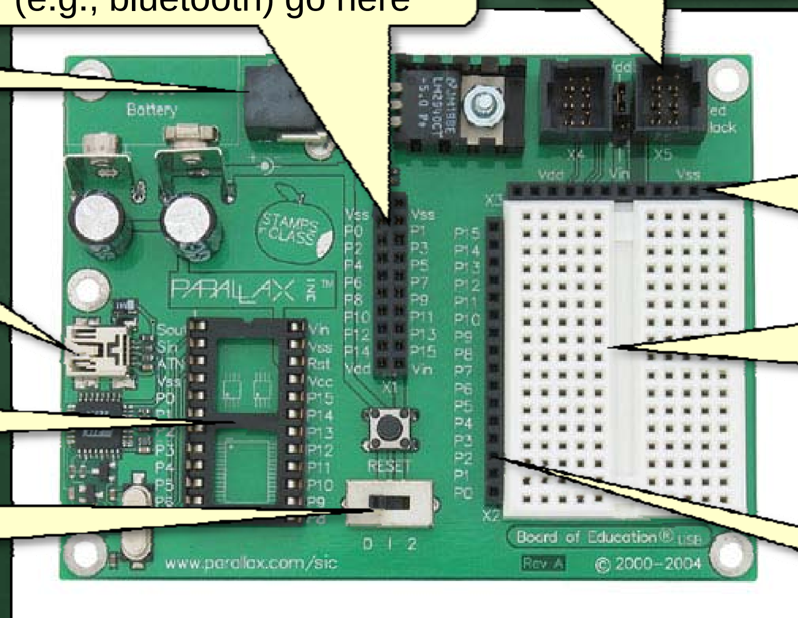
- There is a lot that we are not doing … but it is important to know the basics.

  You can always do an honours project if you have an interesting idea.

The PropBot v2.0

The Board of Education

- Robots for this course were initially based on the Boe-Bot kit:



Power supply or batteries

USB connector to PC
- upload code
- debug

Microprocessor goes here

Power switch:
- 0 = off
- 1 = on without servos
- 2 = on with servos

Application Modules (e.g., bluetooth) go here

Servos (motors) connect here

Power for components:
- Vdd = +5v
- Vss = Gnd
- Vin (we won't use)

Breadboard for connecting circuits (e.g., sensors)

Connections to microprocessor

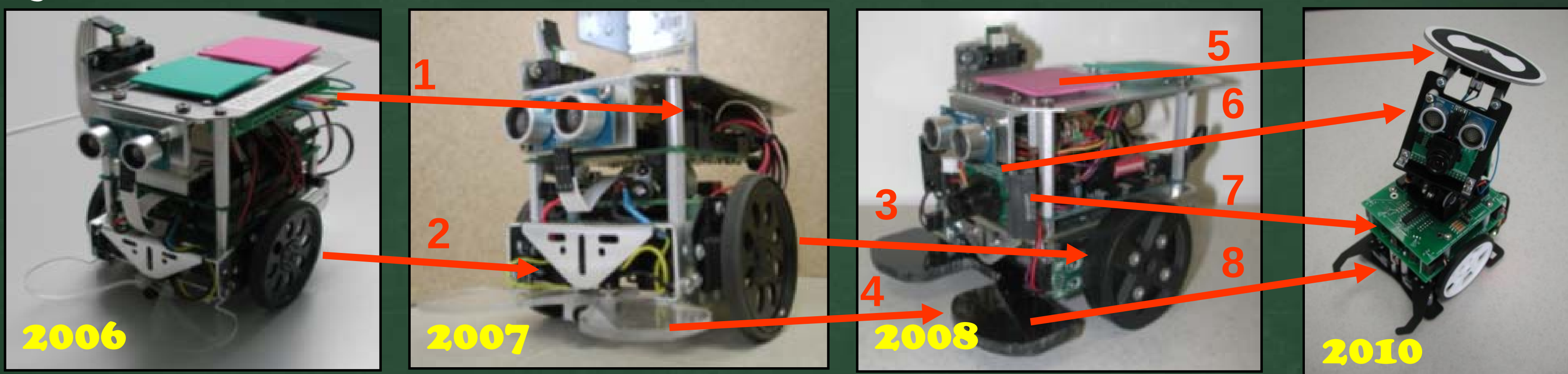
Stamp Microprocessors

- Various "Stamp" microprocessors fit onto BOE:
  - *vary in terms of speed, memory, current usage etc…*

Stamp Microprocessor								
Operating Speed	20Mhz	20Mhz	50Mhz	20Mhz	8Mhz	32Mhz	25Mhz	? Mhz
Execution Speed (instructions per sec.)	~4k	~4k	~10k	~12k	~6k	~19k	~8.5k	83k
RAM (bytes)	32	32	32	38	38	38	32k	400
ScratchPad RAM (bytes)	N/A	64	64	128	128	128	N/A	N/A
EEPROM (Program Size)	2k bytes	8 x 2k bytes	8 x 2k bytes	8 x 2k bytes	16 x 2k bytes	8 x 2k bytes	32k bytes	32k bytes
Current (run)	3mA	25mA	60mA	40mA	15mA	55mA	80mA	~80mA
Current (sleep)	50uA	200uA	500uA	350uA	36uA	450uA	N/A	N/A
PBASIC commands	42	45	45	61	61	63	JAVA	BasicX

Our Robot's History

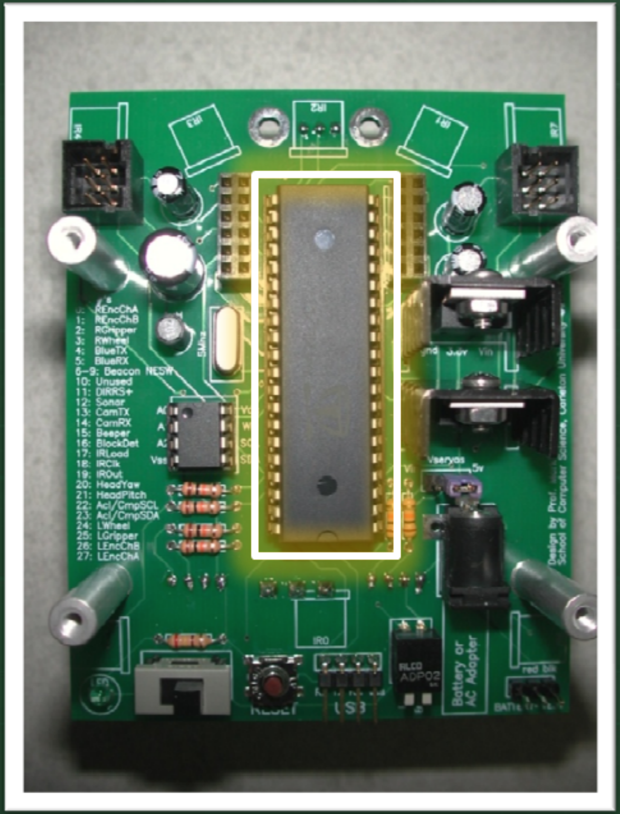
- The robot for this course has been improved over the years …

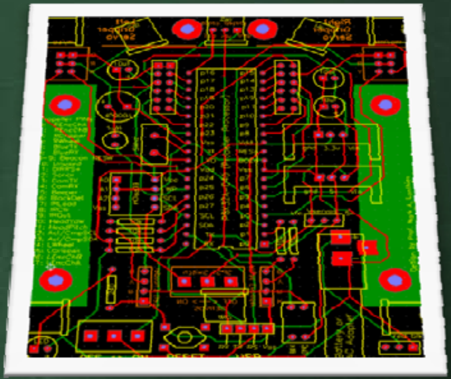


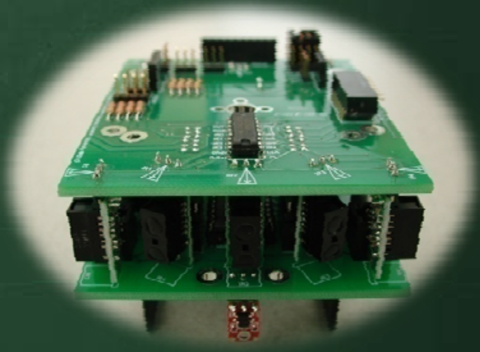
1. Removed user-defined sensor setup … now hardwired to avoid mistakes
2. Switched to battery packs … less battery waste
3. Thicker wheels with real tires … less prone to wear
4. Block guide thicker and wider … hold blocks better when turning
5. Pan/Tilt head, smaller body
6. Smaller/improved tracking tags
7. More IR sensors and other improvements
8. Grippers!

The Changes

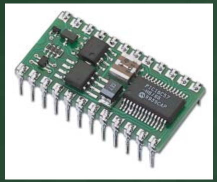
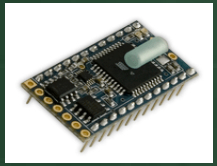
- New to 2011, completely re-designed main board running **Parallax Propeller Processor**:





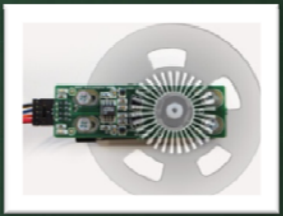


Why Change Microprocessors ?

	The Good	The Bad
Basic Stamp 2 	+ simple to use + very nice IDE + came with BOE kit (no extra cost)	– slow – no floating point numbers – only 16 I/O lines – only 2k of program space – only 24 bytes for variables
BasicX 	+ faster than BS2 + has full floating point math + 32k of program space + 400 bytes for variables	– very poor IDE – difficulties with Serial I/O – still only 16 I/O lines
Propeller 	+ true multi-tasking with 8 processors + has floating point + 64k of program/variable space + nice IDE with Spin language + 32 I/O lines (i.e., more sensors)	– weird floating point math – serial I/O in assembly code

Additional Components

- We also added the following to our robot:
  - WheelWatcher1 **Encoders**
  - **Continuous Rotation Servos**
  - **Wheels** with encoder stickers
  - QTR-1RC **Reflectance Sensor**
  - 8 GP2Y0D810(05)Z0F **IR Sensors**
  - Ping))) **Ultrasonic** Range Finder
  - DIRRS+ **IR Ranging** System
  - CMUcam1 **Camera** Module
  - Micro Pan Tilt System
  - 2 Pico Servos
  - Parallax **EasyBluetooth** Module





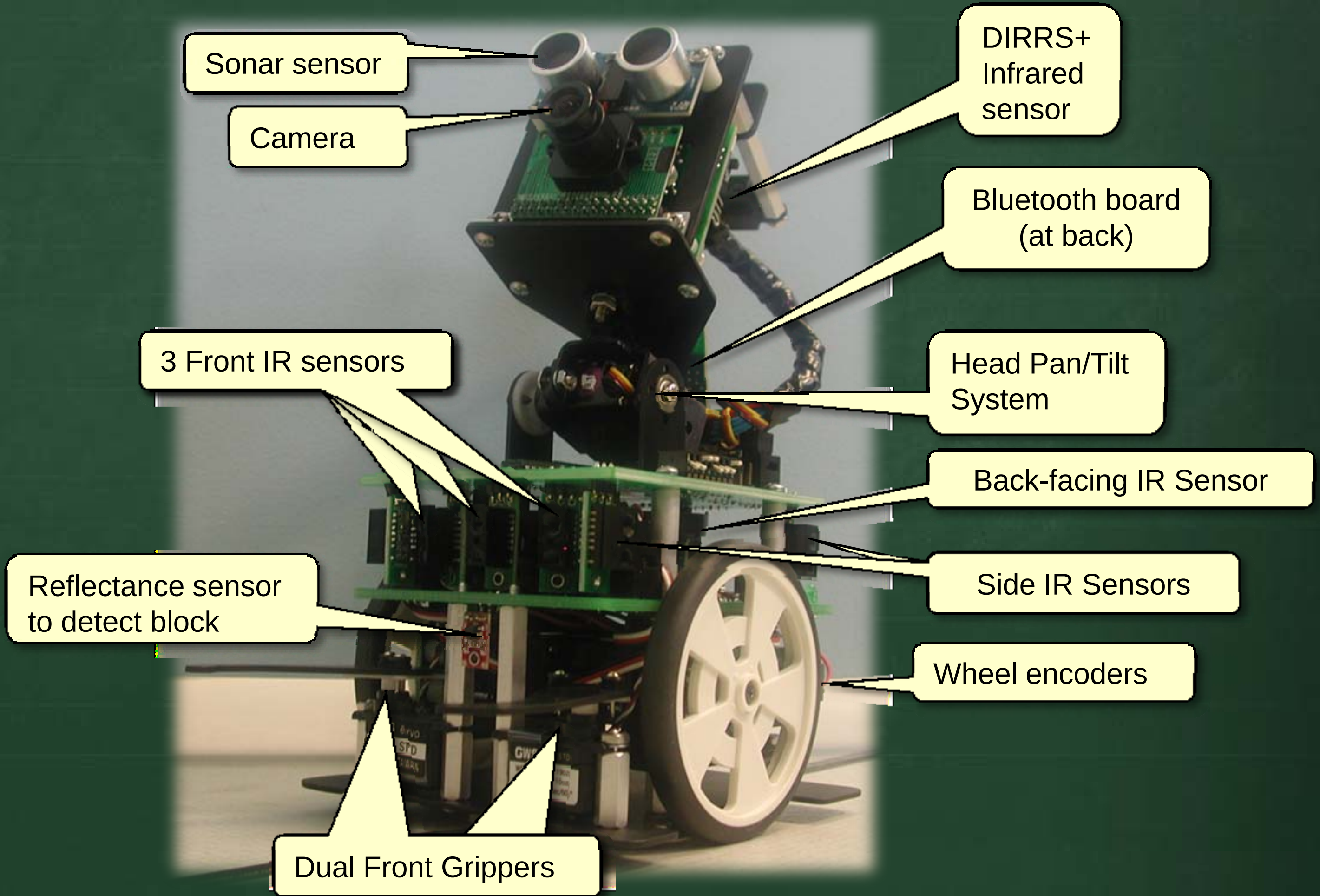






The Completed PropBot 2.0

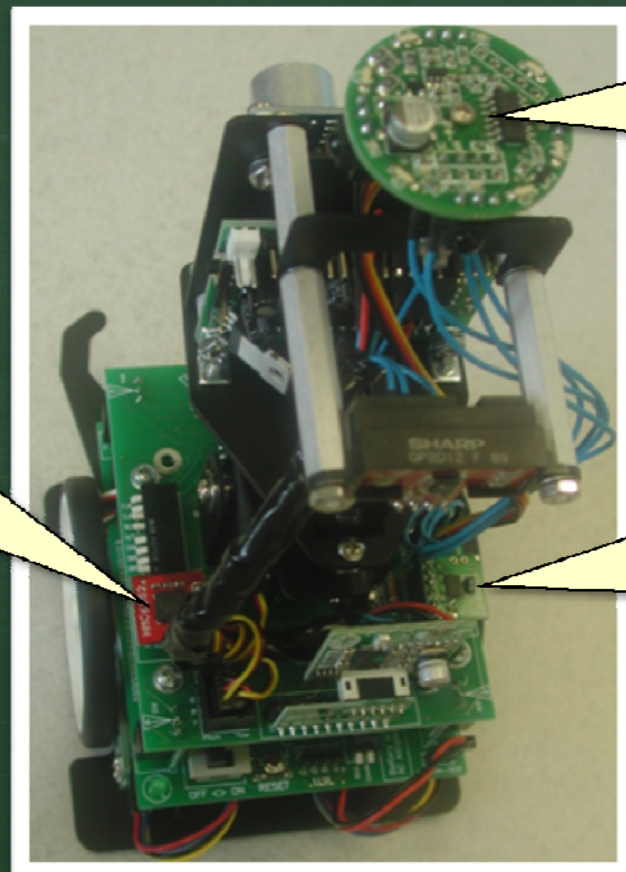
- The completed robot as will be used in the labs:



PropBot Options

- Optional **Accelerometer**, **Compass** and **IR Beacons** may be installed:

HMC6352
Compass
Module



Pololu
IR Beacon
Transceiver

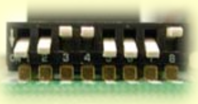
LIS302DL
3-axis
Accelerometer

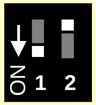
The Sensor Switches

- Robot has an **8-position dip-switch** on the top board and a **2-position switch** at the back of the bottom board:
  - enables/disables power to various sensors
  - always turn off power to sensors not being used to save battery power

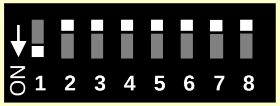
Switches are hard to flip. Use the tip of a pen to flip them up or down.

ON = down, OFF = up

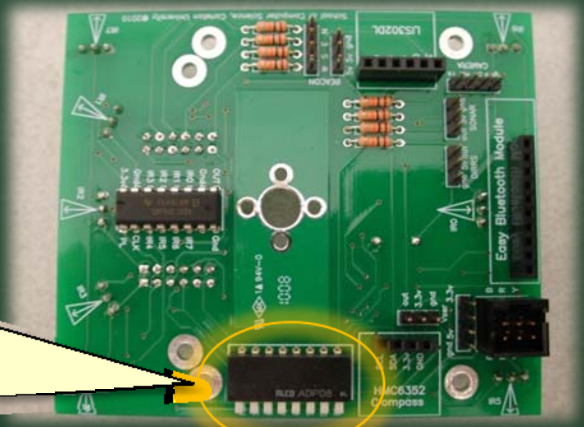




1. **Block Detector**
2. **Wheel Encoders**

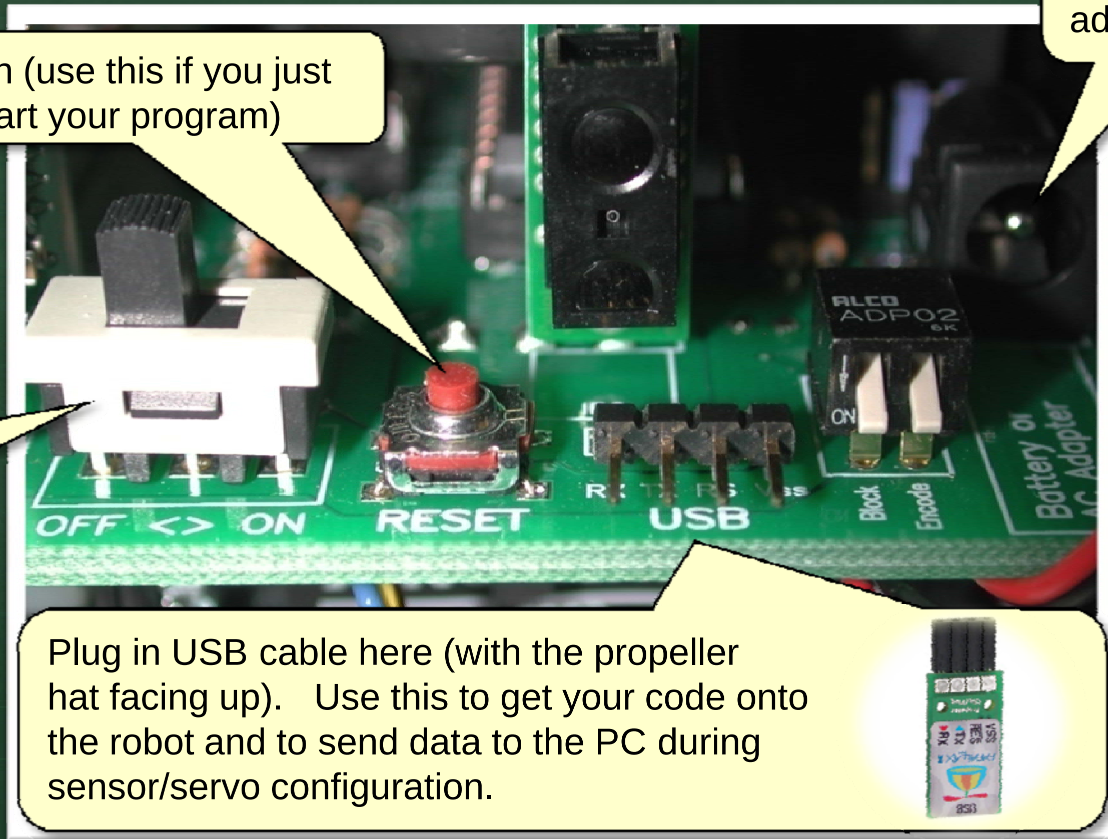


1. **Front & Back IR**
2. **Side IR Sensors**
3. **Bluetooth**
4. **Accelerometer & Compass**
5. **Camera**
6. **Sonar**
7. **DIRRS+**
8. **Beacon**



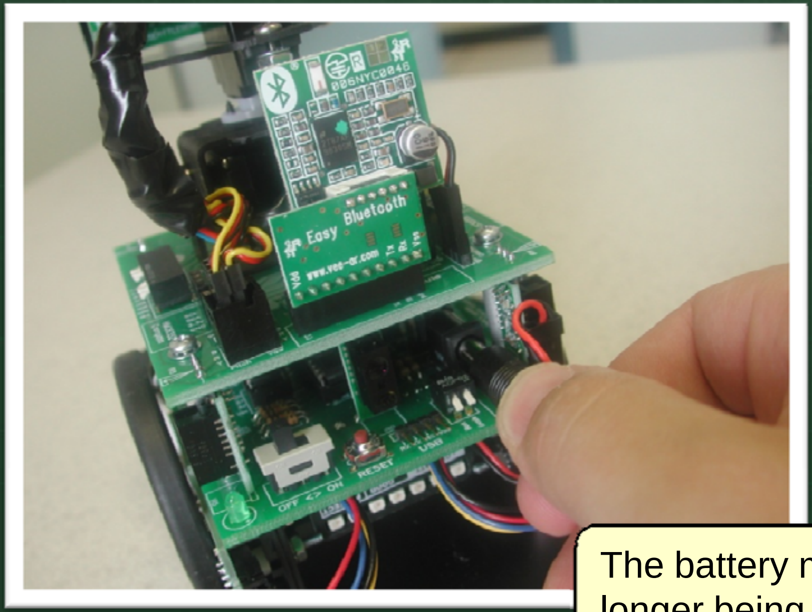
Robot Power

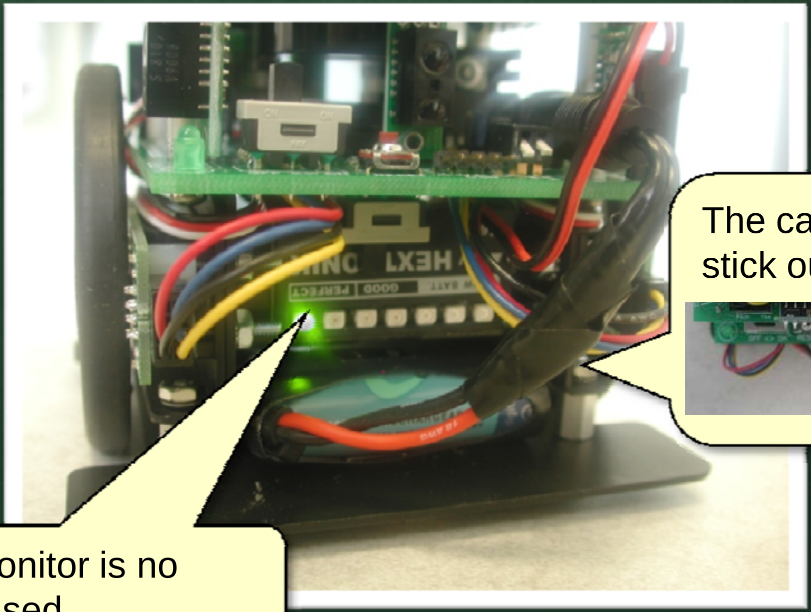
- The back of the robot contains important connections:



The Battery Packs

- Robot uses 7.2v rechargeable battery packs
  - There are also power adapters available for testing the robot while beside you at the desk.
- Plug in battery packs and slide under frame





The battery monitor is no longer being used.

The cables will stick out a bit.

Battery Pack Charging

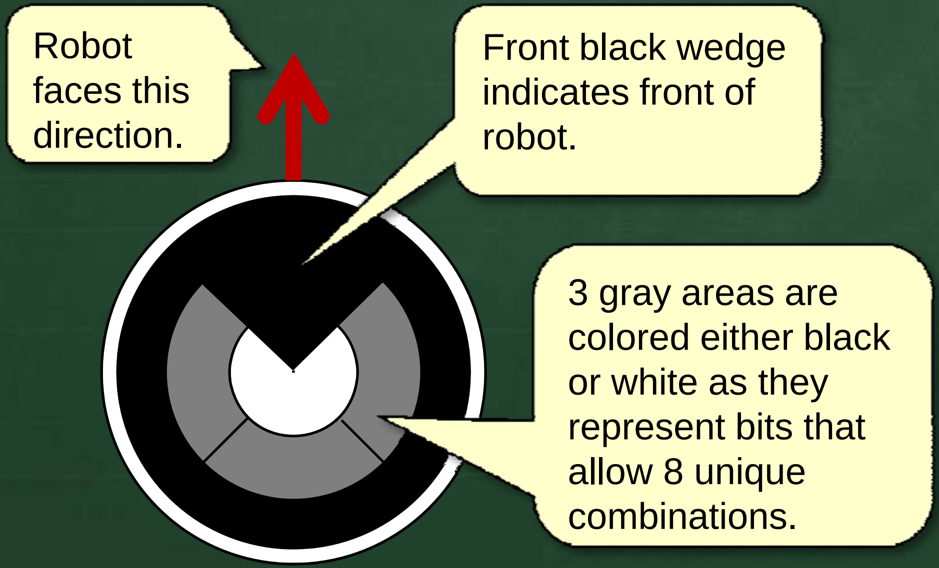
- To recharge a battery pack …
  - remove and unplug it from the robot
  - plug it into the charger
  - press the Start button on the charger
  - wait for 3 beeps, then remove the battery pack

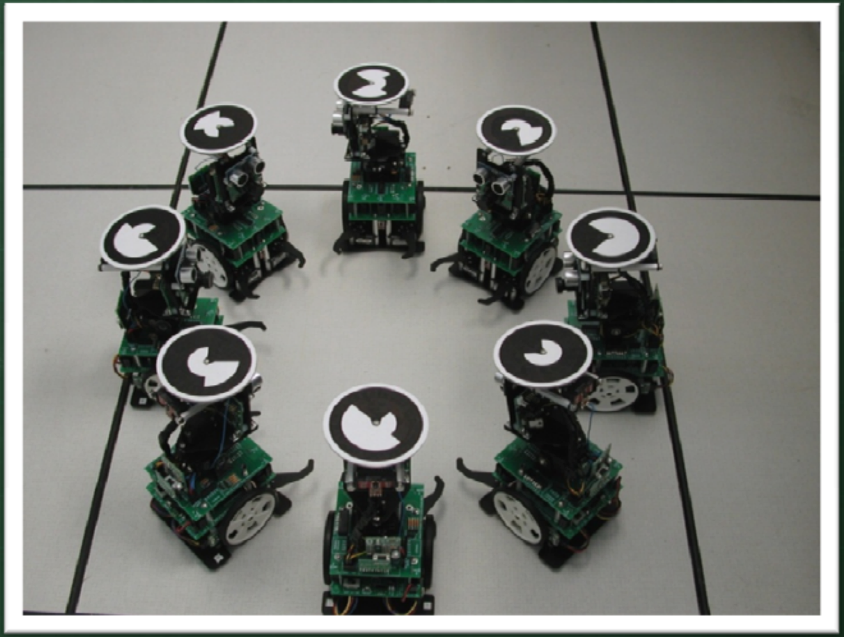
Note:

Sometimes, a battery pack won't fully charge (i.e., the bars don't make it all the way to the left on the display). When this happens, let the pack cool off and then charge it again.

The PropBot Tags

- The PropBots have a circular disk at the top with a unique tracking tag:
  - Over time, tag may rotate a little and need to be adjusted.
  - Largest triangular wedge portion of tag must face forward





Summary

- You should now understand:
  - The areas of study involved with robotics and how we, as computer scientists, fit-in.
  - Some of the main uses and issues in robotics
  - The difference between behavior-based programming and the classical approach
  - The components of the PropBot that will be used in the lab as well as some available options.