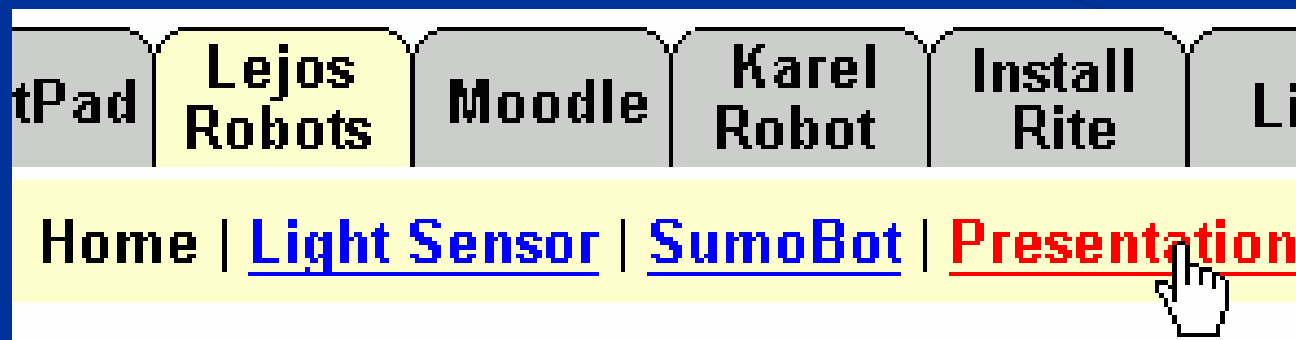


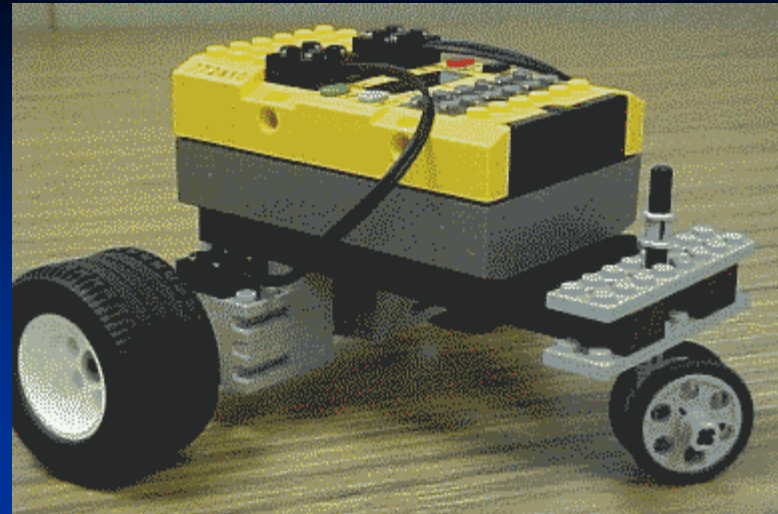
Programming Lego Mindstorms with Lejos

Presentation Handouts

- This PowerPoint
- Other PowerPoints, code samples, handouts, photos, video, etc.
- <http://www.edhsonline.org/other/ap/>
- Click Lejos Robots → Presentation



About Lejos



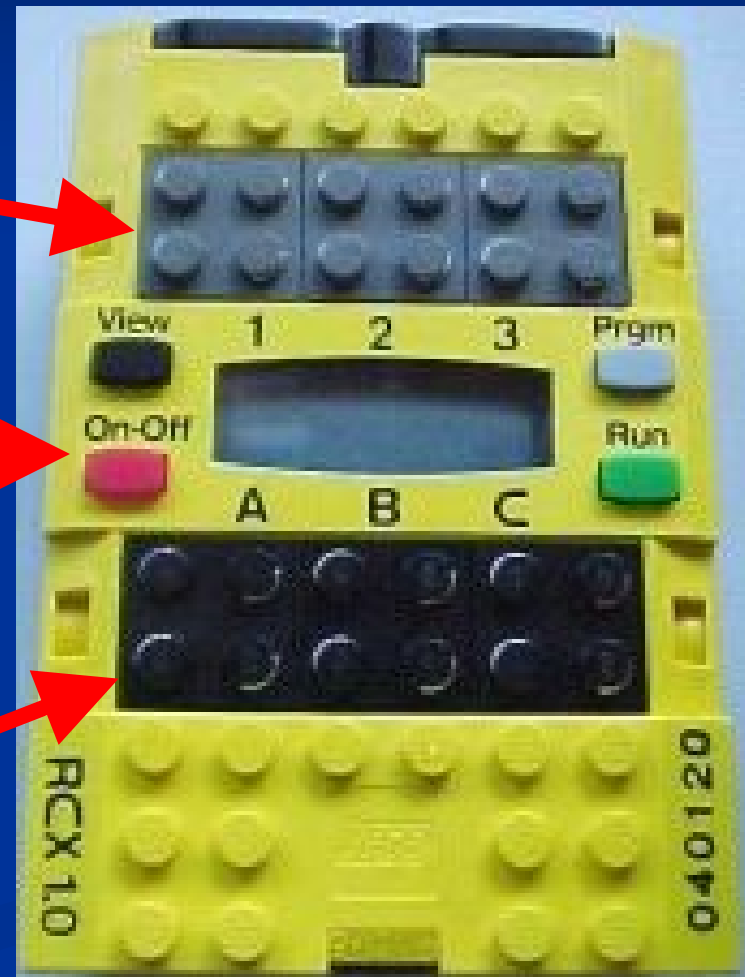
- Lejos is open source Java software for Lego Mindstorm Robots
- It specifically replaces the “RCX Brick” firmware with a tiny Java Virtual Machine
- Visit <http://lejos.sourceforge.net/> for code, tutorials, downloads, etc.

The RCX Brick

Sensor Inputs

Controls Buttons

Motor Outputs



The RCX Brick

- Two motors come with each kit and are controlled by the RCX brick programs...



The RCX Brick

- Mindstorm kits come with two touch sensors and one light sensor



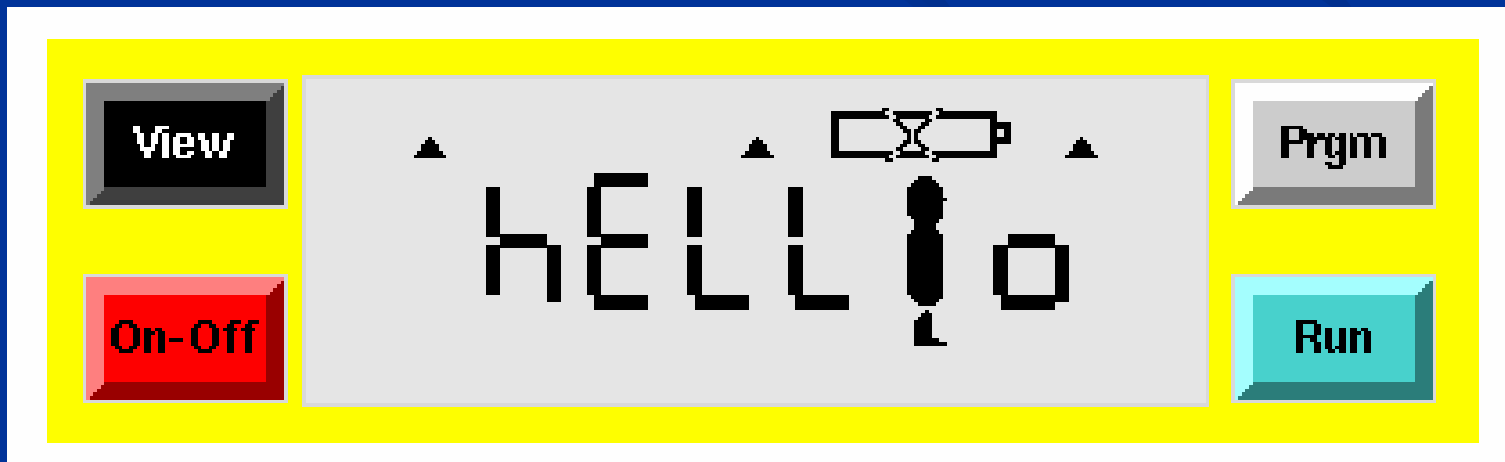
The RCX Brick

- The IR tower sends programs from your Java IDE to the RCX brick



The RCX Brick

- The LCD display on the RCX Brick allows for program execution and selection.
- Four programs can be stored on the brick at once.



Basic Process

- (1) Overwrite Mindstorm firmware with Lejos
- (2) `import josx.platform.rcx.*;`
- (3) Compile, Link, and Download programs to the robot
- (4) Run the robot programs

Some Code Samples

- Hello World on LCD Display
- Note static LCD object

```
public class LejosHelloWorld.java
{
    public static void main (String[]
    {
        LCD.clear();
        TextLCD.print ("hello");
        Thread.sleep(2000);
        TextLCD.print ("world");
        Thread.sleep(2000);
    }
}
```

Some Code Samples

■ Touch Sensor stopping motors

```
public class ExampleTouch
{
    public static void main (String[] args) throws Exception
    {
        Sensor.S1.setTypeAndMode (1, 0x20);
        Sensor.S1.activate();

        Motor.A.forward();
        Motor.C.forward();

        WaitFor.push(Sensor.S1);

        Motor.A.stop();
        Motor.C.stop();
    }
}
```

Some Code Samples

- LCD showing real time light sensor readings

```
public class ExampleLight
{
    public static void main (String[] args) throws Exception
    {
        //Uses sensor integer values instead of constants
        Sensor.S2.setTypeAndMode (3, 0x80);
        Sensor.S2.activate();

        LCD.clear();

        while ( true )
        {
            LCD.showNumber (Sensor.S2.readValue());
            Thread.sleep(50);
        }
    }
}
```

Behaviors & Arbitrators

- How does Lejos implement multiple behaviors?
- How does Lejos decide which behavior has priority?
- Answer: **Arbitrator objects** and the **Behavior interface!!**

Behaviors & Arbitrators

BUMPER CAR ROBOT:

- Normally, we want to **drive forward**
- If we hit a wall, **wall avoidance** should have priority over driving forward.

Behaviors & Arbitrators

Lower level “drive forward” behavior

```
public class DriveForward implements Behavior
{
    public boolean takeControl()
    {
        return true; //takes over when possible
    }

    public void suppress()
    {
        Motor.A.stop();
        Motor.C.stop();
    }

    public void action()
    {
        Motor.A.forward();
        Motor.C.forward();
    }
}
```

Behaviors & Arbitrators

Higher level “hit wall” behavior

```
public class HitWall implements Behavior
{
    public boolean takeControl()
    {
        return Sensor.S2.readBooleanValue();
    }

    public void suppress()
    {
        Motor.A.stop();
        Motor.C.stop();
    }

    public void action()
    {
        Motor.A.backward();
        Motor.C.backward();
        try{Thread.sleep(1000);}catch(Exception e) {}
        Motor.A.stop();
        try{Thread.sleep(300);}catch(Exception e) {}
        Motor.C.stop();
    }
}
```

Behaviors & Arbitrators

- The all powerful **Arbitrator**...
- Higher array position behaviors have priority. In this case the **b2** behavior!!

```
public class BumperCar
{
    public static void main(String [] args)
    {
        Behavior b1 = new DriveForward();
        Behavior b2 = new HitWall();
        Behavior [] bArray = {b1, b2};

        Arbitrator arby = new Arbitrator(bArray);

        arby.start();
    }
}
```

Behaviors & Arbitrators

SUMO WRESTLER ROBOT:

- **Staying** inside the ring is most important
- **Fighting** the enemy is second in importance
- **Searching** for the enemy is third in importance

Behaviors & Arbitrators

SUMO WRESTLER ROBOT:



Behaviors & Arbitrators

SUMO WRESTLER ROBOT:



U++ pushing **KingBot** out of the ring.

Behaviors & Arbitrators

SUMO WRESTLER ROBOT:

2004-2005 SumoBot Competition

June 8th – June 9th, 2005

RESULTS

Place	Team	Win	Loss	Draw	PTS
1st	U++	3	1	8	14
2nd	KingBot	4	3	5	13
3rd	Quit Now... We Won	2	1	8	12
4th	Team Grievous	0	2	10	10

Day 1 Winner: (TIE), U++, Quit Now... We Won

Day 2 Winner, KingBot

Most Impressive Win Streak, KingBot, 4 Wins

Links

- (1) Lejos Homepage:
<http://lejos.sourceforge.net/>
- (2) Parts, etc
<http://www.legoeducationstore.com/>
- (3) Buy at Amazon.com (temporary link?)
<http://tinyurl.com/4wu2f>
- (4) “Columbia University” Lejos Curriculum:
<http://www.sci.brooklyn.cuny.edu/~sklar/er/curriculum/lejos.html>
- (5) SumoBot Rules
<http://www.24tooth.com/SumobotRules.html>

Other Issues

- Check in / Check out boxes to students. Keep track of parts.
- Motors can burn out. Lego VERY GOOD (but slow) about replacing motors. Buy **extra motors**.
- I used a four foot diameter table top to create my Dohyo (Home Depot)

Let's Give It A Try

