NXT Development Tutorial
Part 1: leJOS Basics

NTHU NMSL

Shu-Ting Wang

leJOS?

- leJOS NXJ is a open-source Java programming environment for the LEGO Mindstroms NXT.
- It supports some java utilities and mainstream third-party sensors
- It supports IDE development with Eclipse and Netbeans

Outline

- Environment Setup
  - Prerequisites
  - Flushing firmware
  - Eclipse plugin
- Hello World
- Understand Components
  - Buttons
  - LCD
  - Sensors
  - Motors

Outline

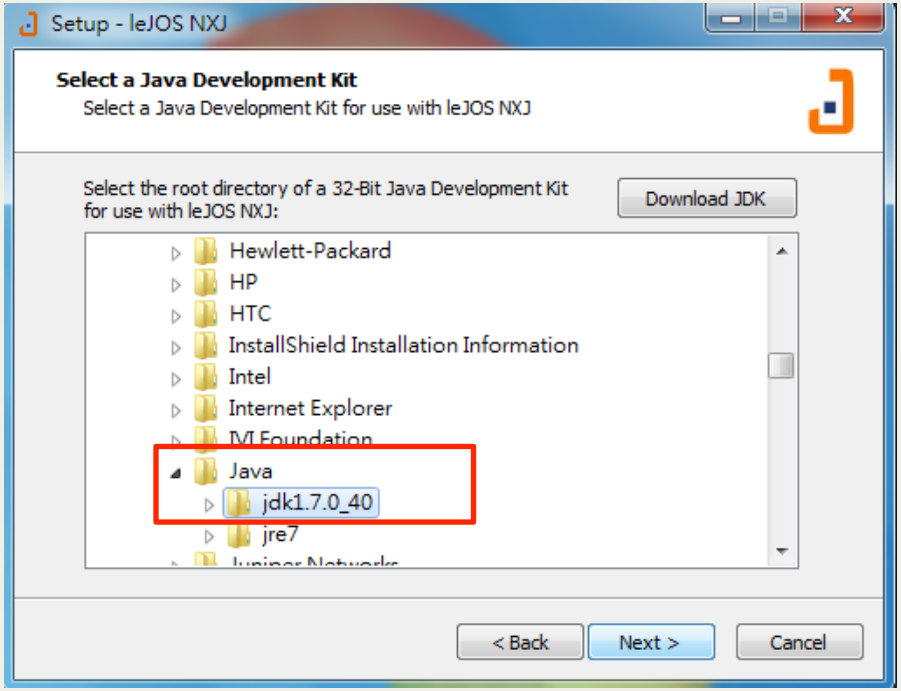
- Environment Setup
  - Prerequisites
  - Flushing firmware
  - Eclipse plugin
- Hello World
- Understand Components
  - Buttons
  - LCD
  - Sensors
  - Motors

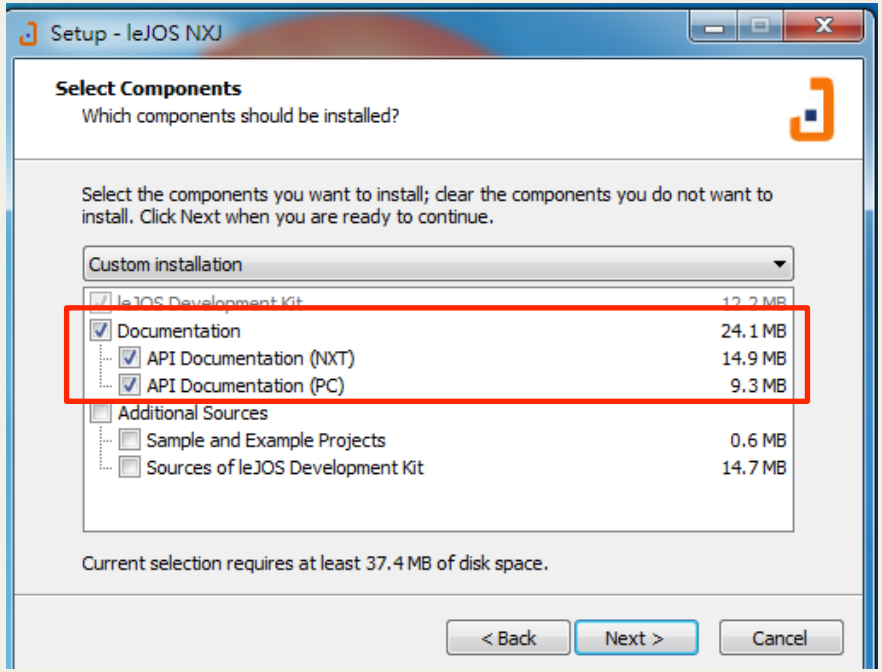
Prerequisites

- **32 bits** JDK and **32 bits** IDE
- NXT programming tool
  - To install the driver and update firmware
  - Download Link

Flushing firmware(1/3)

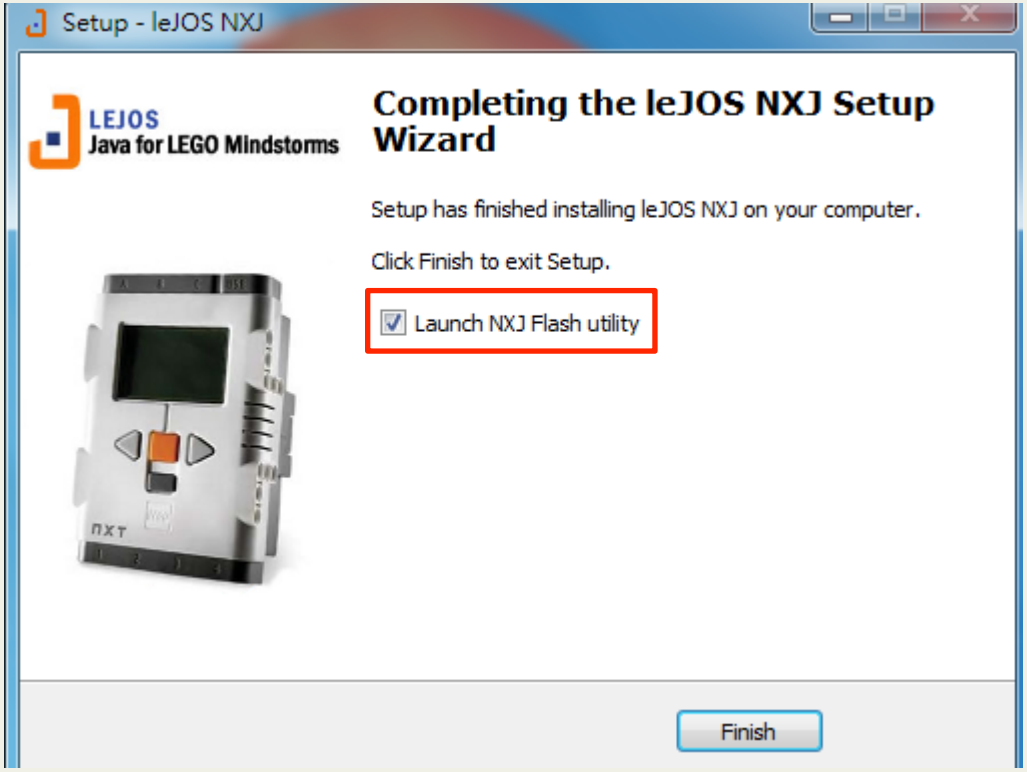
- First, download the latest leJOS and install
- Make Sure you select the 32bits JDK .
- Documents can be useful





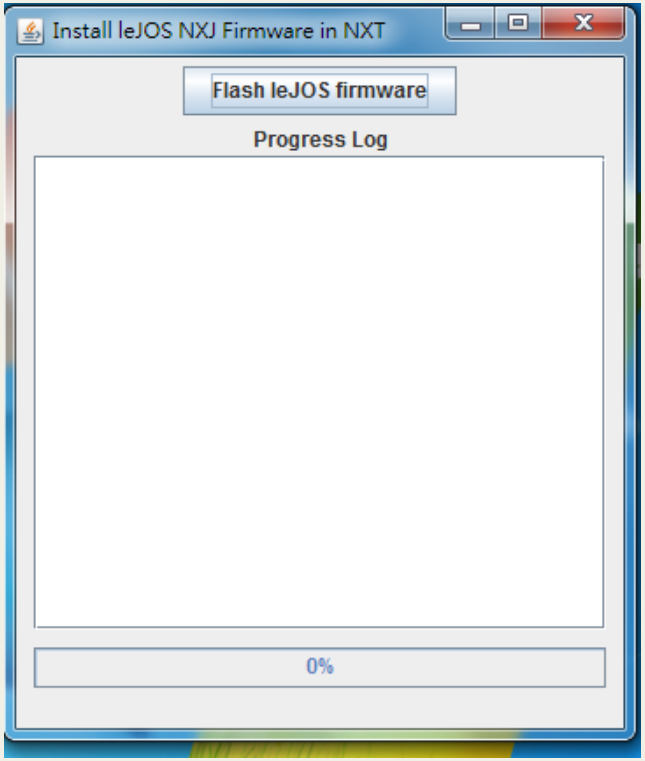
Flushing firmware(2/3)

- Click the check box and ready to flush.



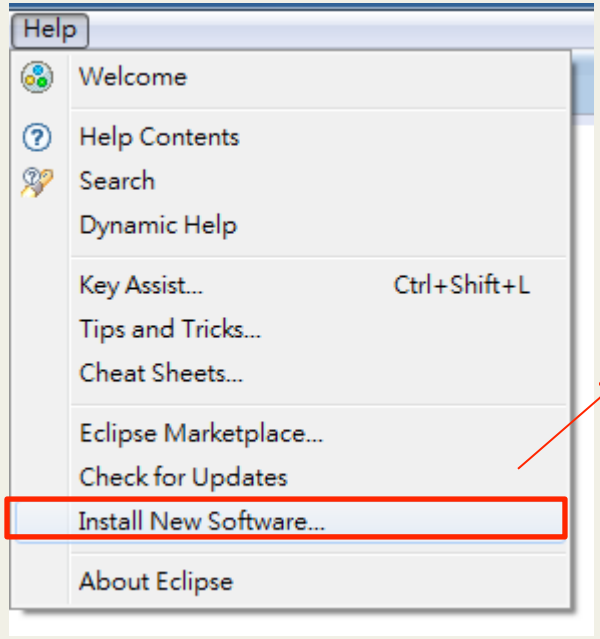
Flushing firmware(3/3)

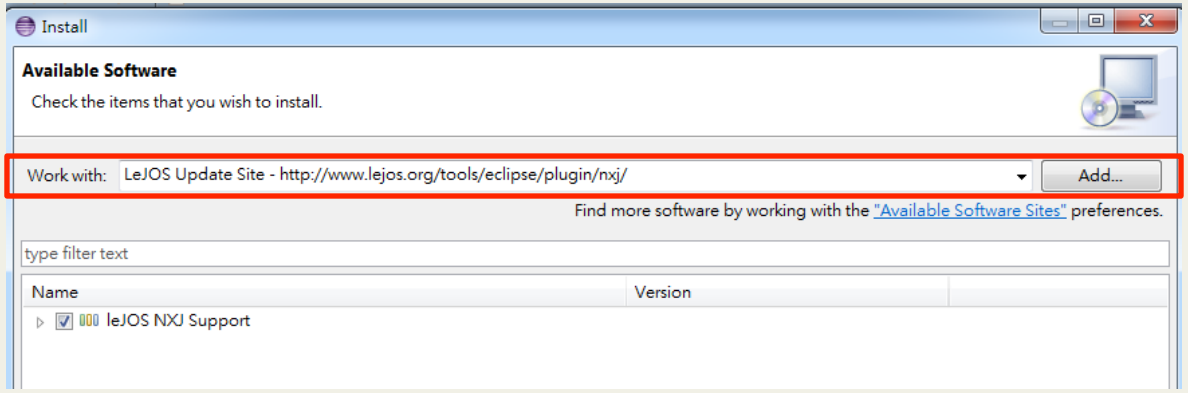
- Flush leJOS firmware
- Don’t Worry, the original firmware can be recovered



Eclipse plugin

- Type"http://www.lejos.org/tools/eclipse/plugin/nxj/" in the “Work With” row
- Press OK in security warning pop out window





Outline

- Environment Setup
  - Prerequisites
  - Flushing firmware
  - Eclipse plugin
- Hello World
- Understand Components
  - Buttons
  - LCD
  - Sensors
  - Motors

Hello World 1.0

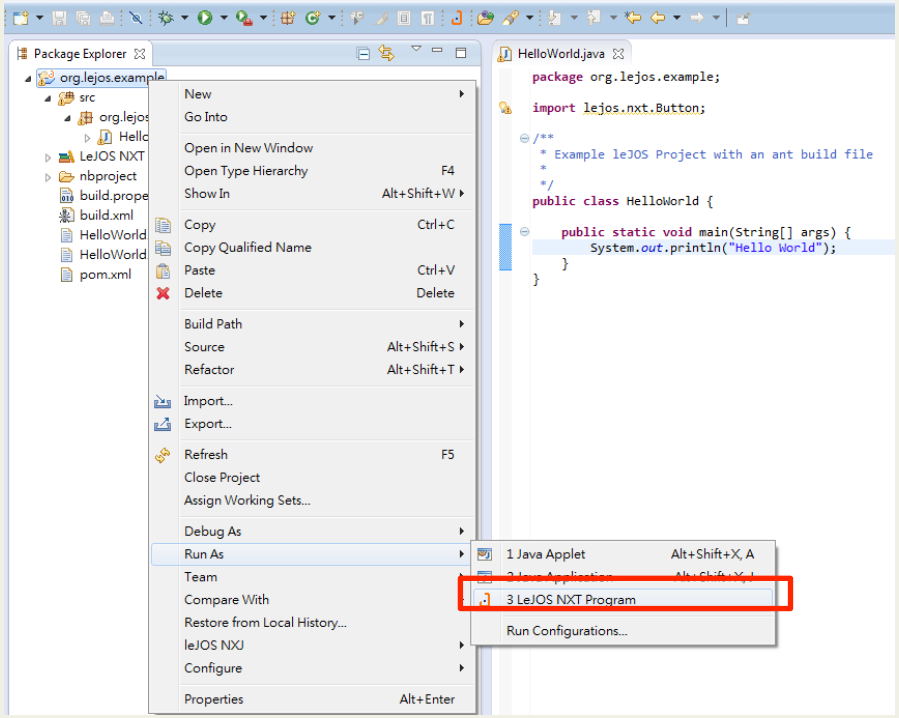
- Let's try "Hello World" on NXT !

```
public class HelloWorld {
        public static void main(String[] args) {
                System.out.println("Hello World");
        }
}
```

Wait! Is this Hello World for NXT?

Hello World 1.0

- Sure. leJOS makes you do programming on NXT very similar to you did on Java
- Launch Hello World 1.0



Hello World 1.1

- But I cannot see anything?
- Let's change something, and run again

```
public class HelloWorld {
        public static void main(String[] args) {
                System.out.println("Hello World");
                Button.waitForAnyPress();
        }
}
```

Outline

- Environment Setup
  - Prerequisites
  - Flushing firmware
  - Eclipse plugin
- Hello World
- Understand Components
  - Buttons
  - LCD
  - Sensors
  - Motors

LCD

- Control the LCD by calling draw functions

```
public class HelloLCD {
    public static void main(String[] args) {
        LCD.drawString(Bluetooth.getLocalAddress(), 0, 0);
        LCD.clear();
        LCD.drawString("Free RAM:", 0, 0);
        LCD.drawInt((int)System.getRuntime().freeMemory(),
        6, 9, 0);
        Button.waitForAnyPress();
    }
}
```

Button

- Try the following to test the four buttons

```
import lejos.nxt.*;
public class ButtonPresses {
    public static void main(String[] args) throws Exception {
        while(!Button.ESCAPE.isDown()) {
            LCD.clear();
            if (Button.ENTER.isPressed())
                LCD.drawString("ENTER", 0, 0);
            if (Button.ESCAPE.isPressed())
                LCD.drawString("ESCAPE", 0, 0);
            if (Button.LEFT.isPressed())
                LCD.drawString("LEFT", 0, 0);
            if (Button.RIGHT.isPressed())
                LCD.drawString("RIGHT", 0, 0);
        }
    }
}
```

Sensors

- NXT is equipped the following sensors:
  - Light 
  - Sound 
  - Touch 
  - Ultrasonic 
- There are 4 ports for sensors, ex: S1-S4, and 3 ports for motors, ex: A, B, and C.

Ultrasonic Sensor

- Get the distance from your hand

```
public static void main(String[] args) throws Exception {
        UltrasonicSensor Ultrasonic = new UltrasonicSensor(SensorPort.S1);
        // UltrasonicSensor uses port 1 on NXT
        while(!Button.ESCAPE.isDown()) {
                LCD.clear(3,0,0);
                LCD.drawInt(Ultrasonic.getDistance(),3,0,0);
        }
 }
```

Motors

- What can motors do?
  - Forward/ Backward
  - Float freely
  - Stop
  - Rotate
  - Get or Set status: setPower, getSpeed … etc.

Hands-on Practice

- **What the program should do:**
  1. Display "Program 1 " in the top line of the LCD.
  2. Wait for a button to be pressed
  3. Run motor A in the forward direction.
  4. Display " FORWARD " in the top line.
  5. Wait until a button is pressed.
  6. Run the motor backward.
  7. Display " BACKWARD " in next line.
  8. Wait until a button is pressed.
  9. Stop the motor.

Useful Classes

Class	Method name	Notes
NXTRegulatedMotor, e.g. Motor.A	forward()	Start the motor rotating forward
	backward()	Start rotating backward
	stop()	Stop quickly
Button	waitForPress()	Wait till any button is pressed
LCD	drawString(String str, int x, int y)	Draw a string at position x in row y

Coding Time

Reference

- leJOS official tutorial
- leJOS API documentation