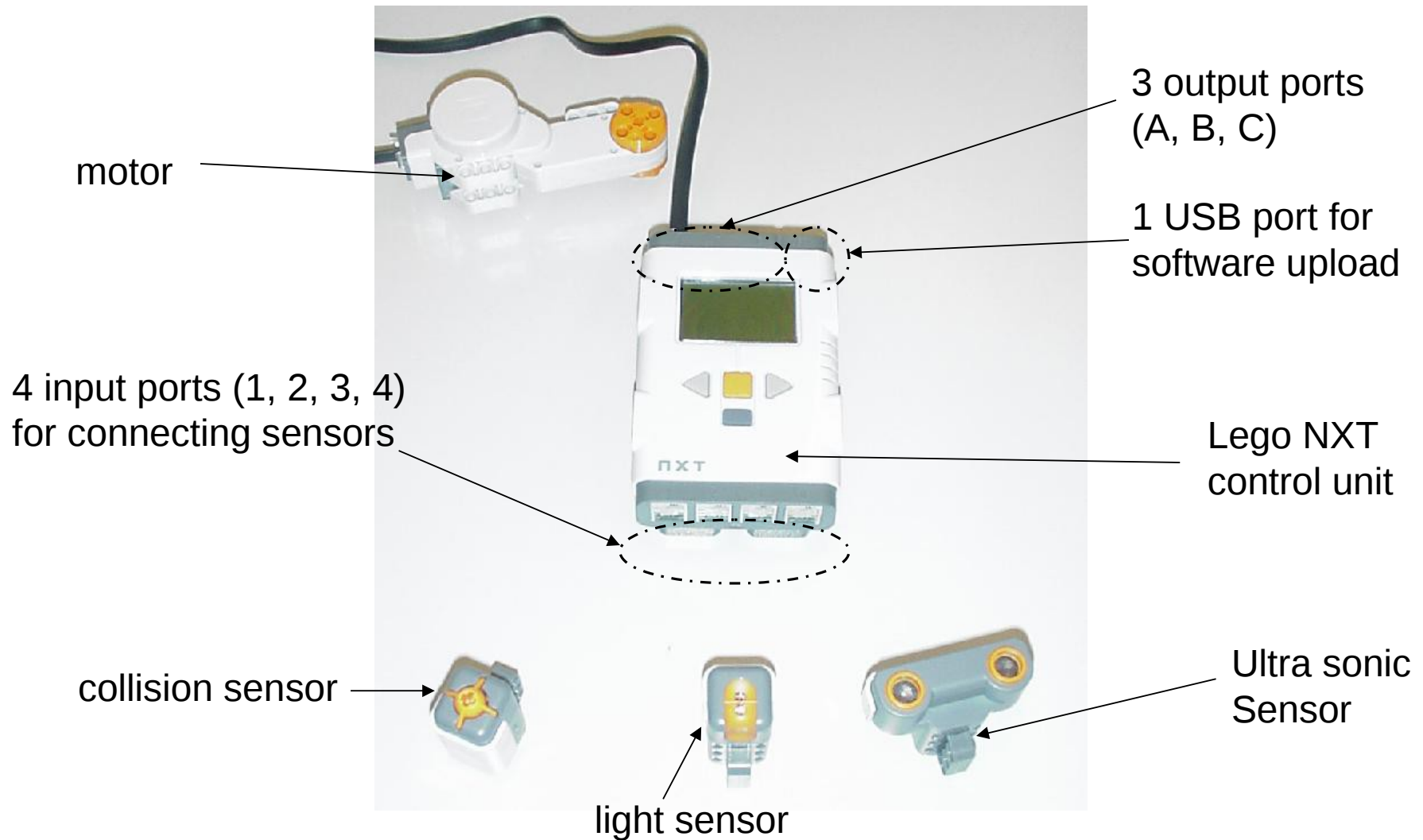


LEGO mindstorm robots

Peter Marwedel
Informatik 12
TU Dortmund
Germany



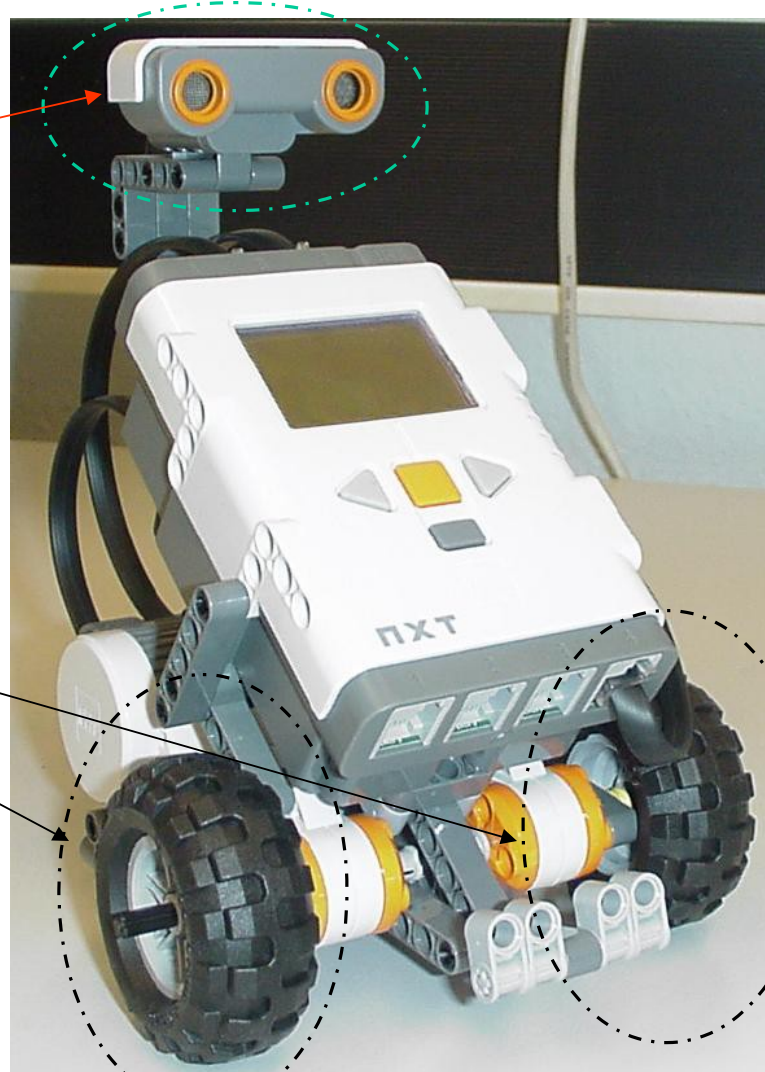
Lego Mindstorm® components



Basic robot for lab

1 ultra sonic sensor

2 independently
controlled
wheels

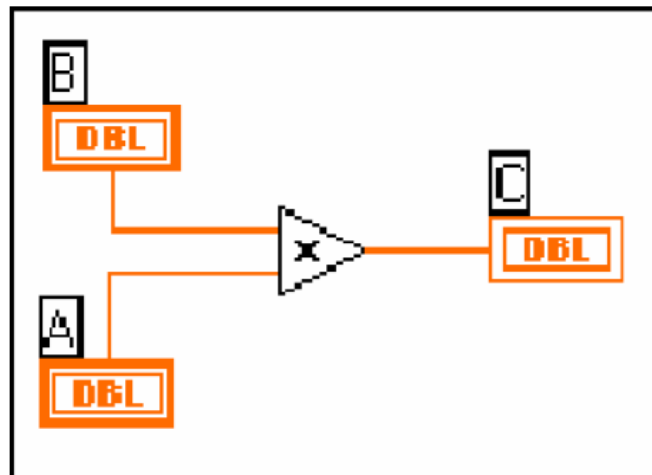


The basic robot will be extended by additional sensors and actuators during the labs

Data flow programming using LabVIEW

- LabVIEW programs = graphs
- Specification of operations and dependences
- can be executed in arbitrary sequence as long as data dependences are met
- we don't care about the precise sequential code needed for each of the nodes.

Example:

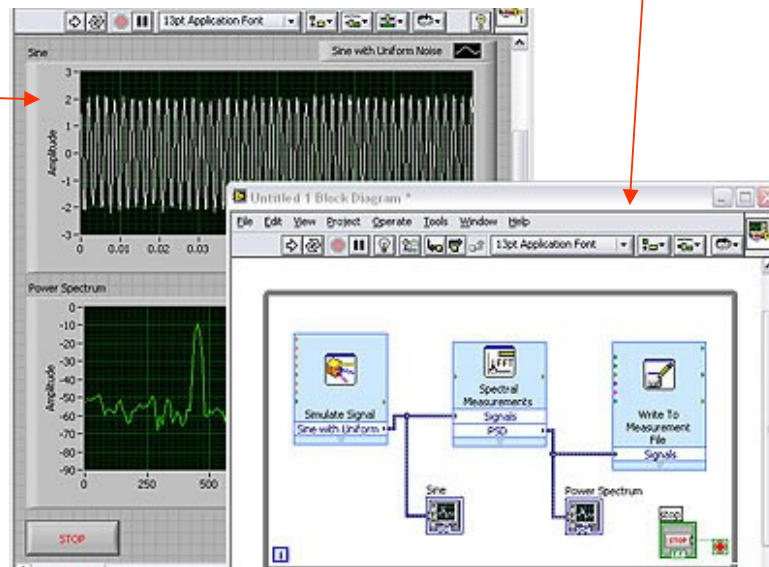


Virtual instruments

VI = virtual instrument

VIs represented in 2 windows:

- Front panel: user interface
- Block diagram: functionality of the system



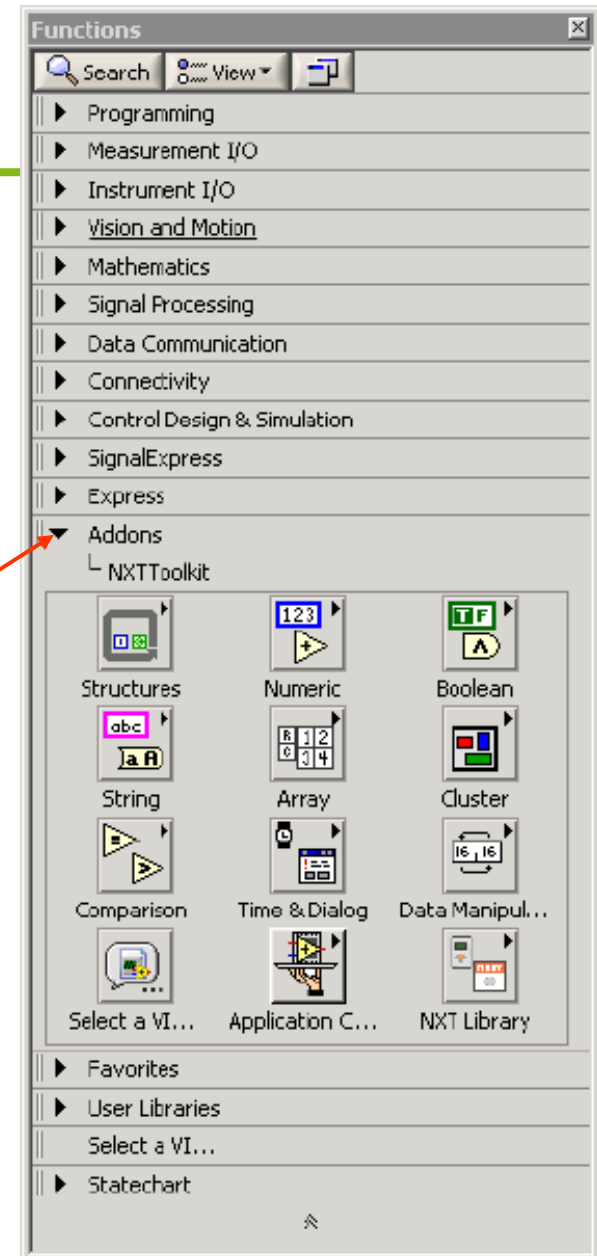
LabVIEW NXT tool kit

Front panel irrelevant, since robots have no “user interface“

Mindstorm programming:

- functions menu contains all required features as addons
- Use **only** these addons!
The other features are not available for the mindstorms

Introduction of most relevant NXT features



Input (1)

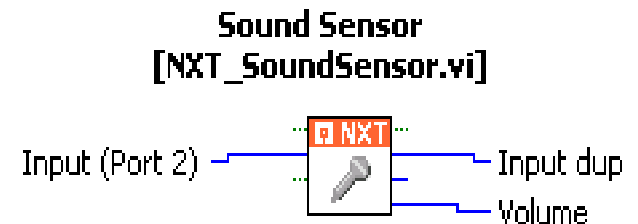
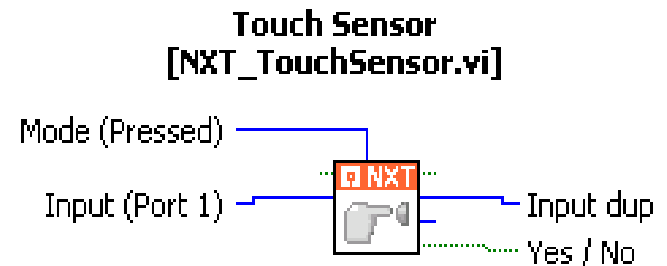
Menu: NXTToolkit => NXT Library =>
Input => Touch Sensor / Sound Sensor

Touch sensor

- reads in sensor from designated input port
- Mode: pressed/released
- Output: yes/no

Sound sensor

- reads in sensor from designated input port
- Output: volume

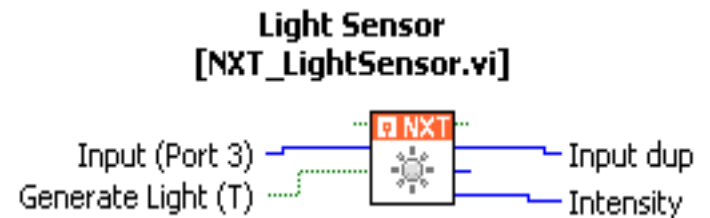


Input (2)

Menu: NXTToolkit => NXT Library => Input => Light Sensor / Ultrasonic Sensor

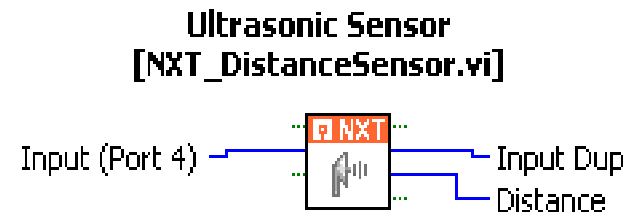
Light sensor

- reads in sensor from designated input port
- sensor must be switched on (Generate Light => True)
- Output: intensity

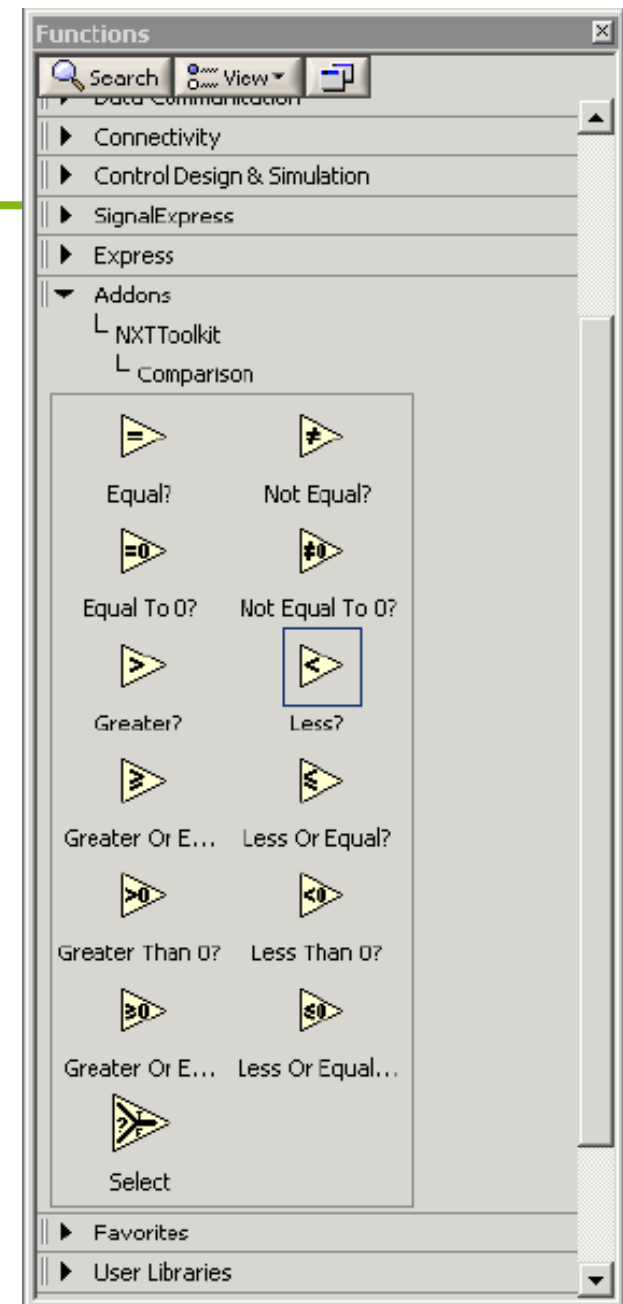


Ultra sonic sensor:

- reads sensor from designated input port
- Output: distance



Comparison



Menu: NXTToolkit => Comparison

- Essentially self-explaining
- Result: Boolean
- Exception: Select

$\approx 2 \rightarrow 1$ Mux

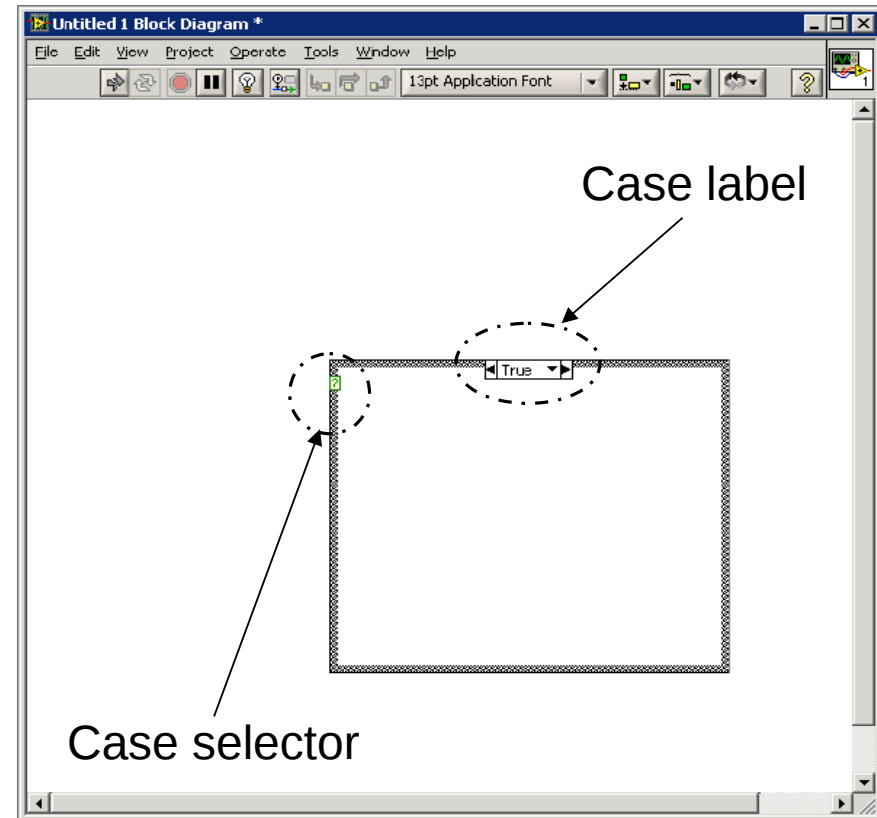
Case-dependent data flow

(some control elements present in LabView)

Menu: NXTToolkit => Structures => Case Structure

Move from Functions-Palette into editing area using drag & drop

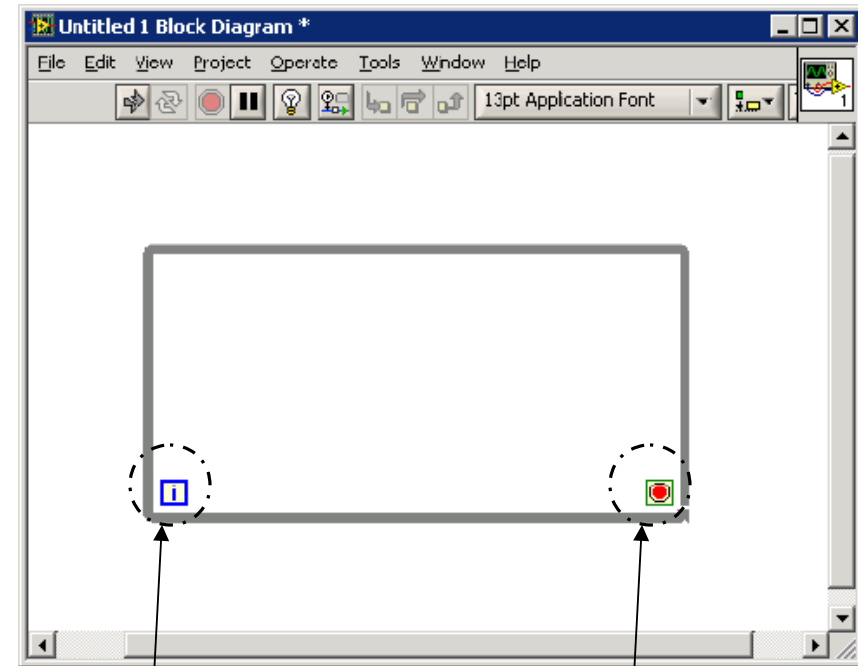
- Consists of several sub diagrams, only one of which can be active
- Click on the arrows next to the case label to display a particular sub diagram.
- The case selector serves as the input to the case structure;
Possible data types: Bool, String, Integer.
- Action to be performed designated by additional elements within the case structure
- Right click opens context menu



Data flow loops

Menu: NXTToolkit => Structures => While Loop

- Sub diagram will be repeated until Boolean condition is true
- Condition is represented by conditional terminal
- Right allows selecting whether iterations will stop or continue if condition is true
- The iteration terminal includes the number of the actually executed iteration



Iteration terminal

Conditional terminal

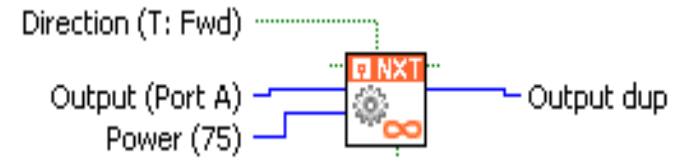
Output (1)

Menu: NXTToolkit => NXT Library =>
Output => Motor Unlimited / Sync Unlimited

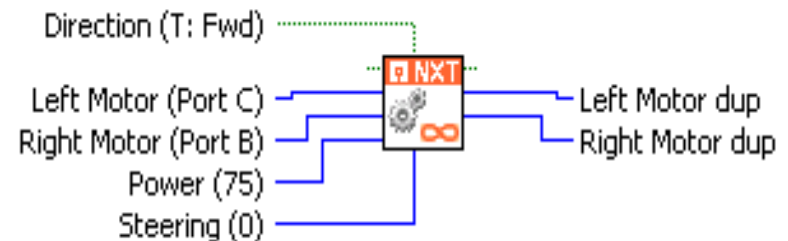
Motor control (unlimited)

- controlling one motor
 - Designate output port
 - Direction (true = forward)
 - Velocity
- controlling both motors
 - Designate output ports
 - Direction
 - Velocity
 - Relative speed (steering):
-100 (left) ... 100 (right)

Motor Unlimited
[NXT_MotorUnlimited.vi]



Sync Unlimited
[NXT_SyncUnlimited.vi]



Output (2)

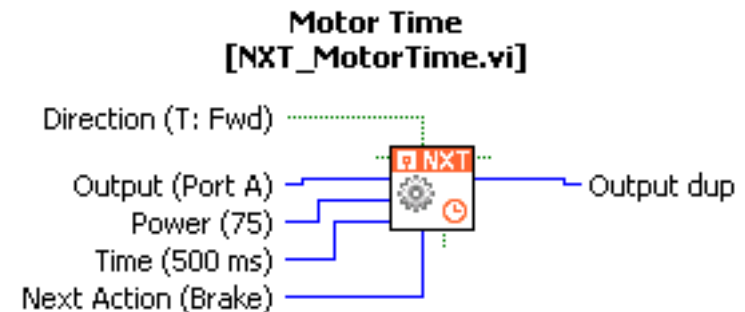
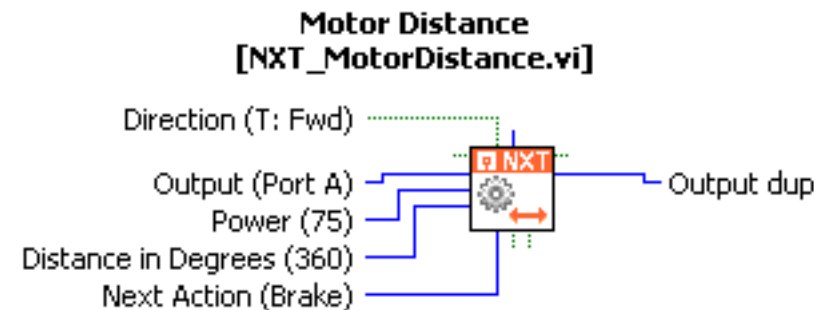
Menu: NXTToolkit => NXT Library =>
Output => Motor Distance / Motor Time

Motor control (distance)

- controlling one motor
 - Designate output port
 - Direction
 - Velocity
 - Distance in degrees
 - Follow-up action
(braking, free running)

Motor control (time)

- controlling one motor
 - Distance indicated via time
 - Otherwise, same as above



Output (3)

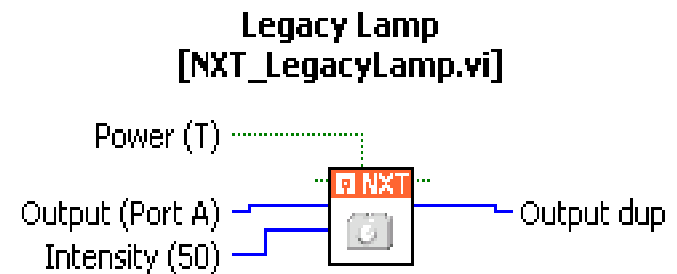
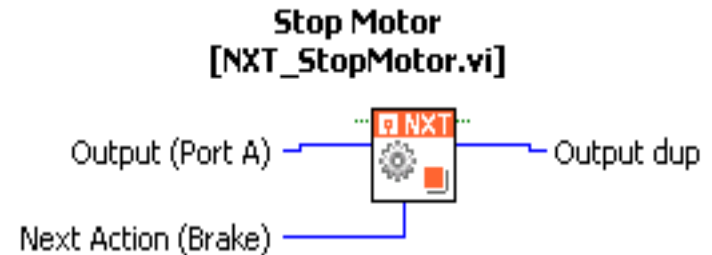
Menu: NXTToolkit => NXT Library => Output =>
Stop Motor / Legacy Lamp

Motor control (stop)

- controlling one motor
 - Designate port
 - Stop motor thrust
- there is a variant controlling both motors

Lamp

- Designate output port
- indicate intensity
- switch on explicitly



Display

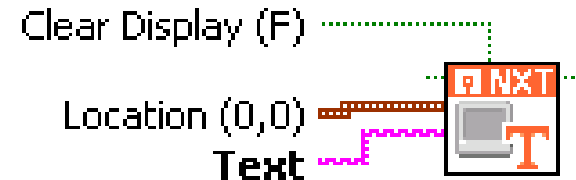
Menu: NXTToolkit => NXT Library =>
Display => Display Text

Display

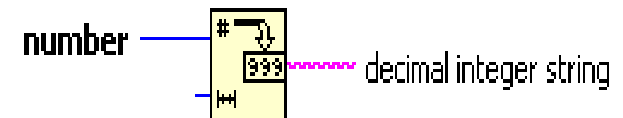
- display line of text on robot's display
- displays only strings
- numbers must be converted into strings:

*NXTToolkit => String =>
String/Number Conversion
=> Number to Decimal*

Display Text [NXT_DisplayText.vi]



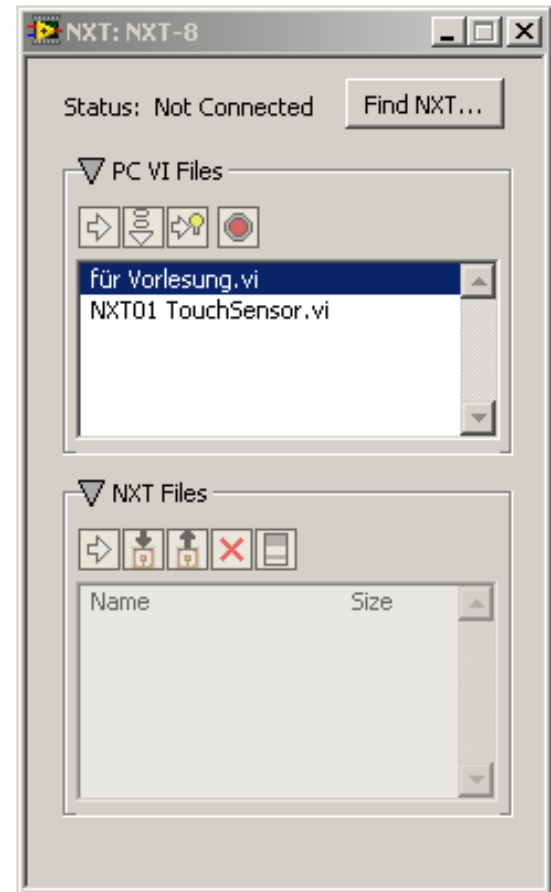
Number To Decimal String



Downloading software

Menu: Tools => NXT Module => NXT Terminal

- Download either via USB or Bluetooth
- Terminal window
 - Find robot
 - compile + download
 - upper window: files on PC
 - lower window: files on robot



Small example

Goal: robot moves forward until collision sensor detects a collision

