Overview

- Introductions
- Overview
- Autonomous GNC
- **Lego® Mindstorms Intro**
- Lego® Mindstorms Challenge

2:00 – 3:15 – Robotics Laboratory Tutorial (Jade)

•Hardware components, ports, servo motors, sensors
  - Get to know your NXT (30 min.)

•Software development environment: BricxCC (10 min.)

•Programming: NXC (35 min.)

3:15 – 3:30 – Break

3:30 – 5:30 Laboratory Continued (All)

- Challenge 1: Building a Tri-Bot (30 minutes)
- Challenge 2: Use sonar for obstacle detection and avoidance (20 min.)
- Challenge 3: Color sensor calibration (10 min.)
- Challenge 4: Color line following (30 min.)

Lego® Mindstorms: Introduction

- A line of Lego sets combining
  - Microcontroller (programmable brick)
  - Motors, sensors, and Lego parts
- Invented by MIT Media Lab
- First released in 1998 as Robotics Invention System
  - 2 motors, 2 touch sensors, 1 light sensor

- Lego Mindstorm NXT, 2006
  - 3 servo motors, 4 sensors (touch, light, sound, ultrasonic)
- Lego Mindstorm NXT 2.0, 2009
  - 3 servo motors, 4 sensors (2xtouch, color, ultrasonic)
- Software:
  - GUI-based programming software (NI LabVIEW as engine)
  - Third-party languages
    - leJOS: Java
    - RobotC
    - Interactive C
    - Not Quite C (NQC)
    - Not Exactly C (NXC)
    - ……..

Lego® Mindstorms NXT2.0 Hardware

- Introductions
- Overview
- Autonomous GNC
- **Lego® Mindstorms Intro**
  - **Hardware**
  - Software
  - Programming
- Lego® Mindstorms Challenge

NXT Intelligent Brick

Servo motors with built-in rotation sensors (3)

Rechargeable lithium battery

Touch

Sound

Ultrasonic

Color

Light

Connector cables

NXT Software

- USB cable
- Charger
- Lego parts
- Wheels
- Gears
- Lamps
- ……

Lego® Mindstorms NXT: Additional Components

- Introductions
- Overview
- Autonomous GNC
- **Lego® Mindstorms Intro**
  - **Hardware**
  - Software
  - Programming
- Lego® Mindstorms Challenge

Acceleration

Compass

Gyro

IR sensor

RFID

Temperature sensor

Bluetooth Dongle

Getting to Know Your NXT

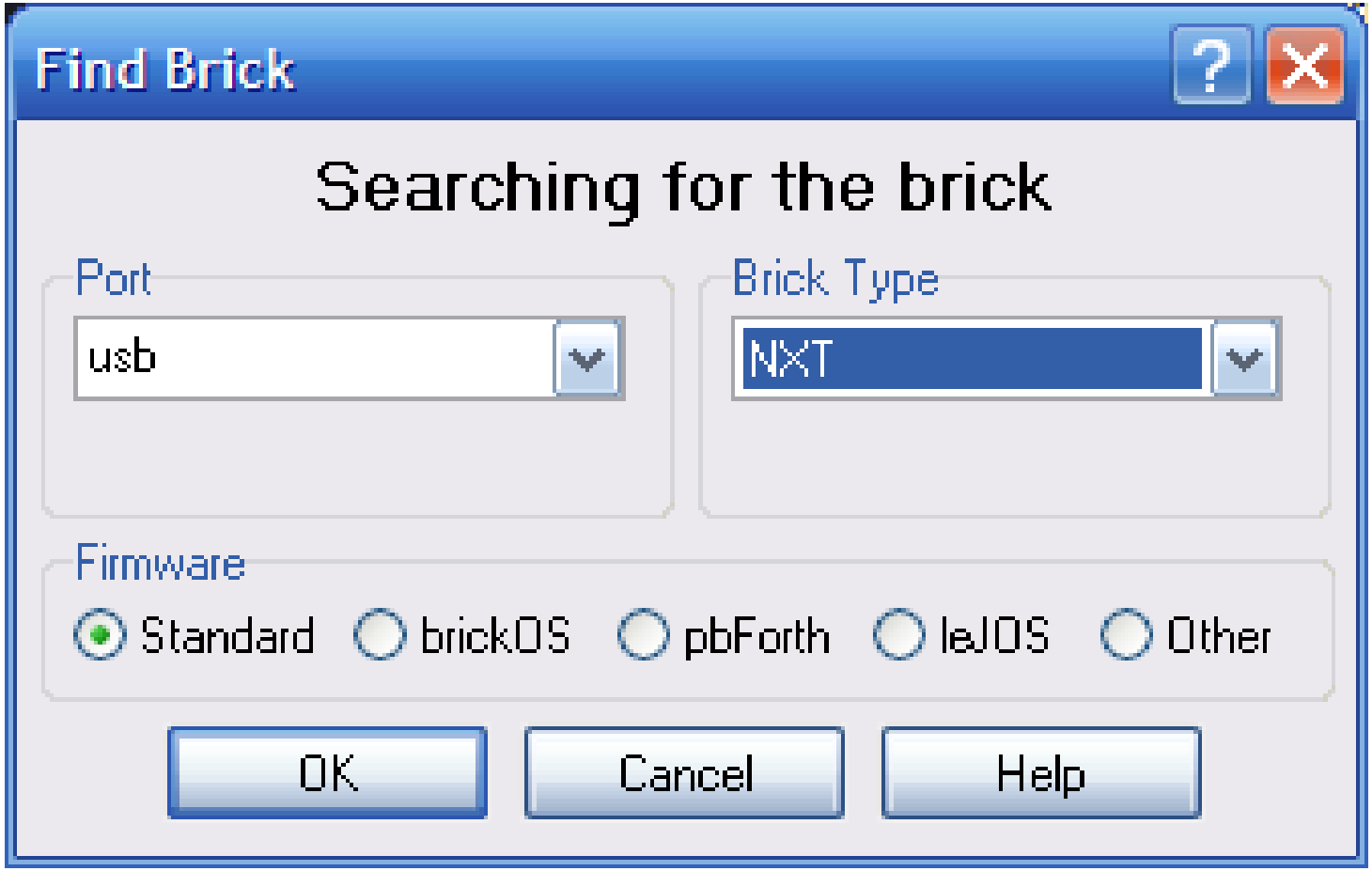
- Get to know you NXT robot (30 minutes):
  Follow instructions in LEGO Mindstorms User Guide p16-35 to learn the following features of the NXT:
  - Sensor inputs and motor ports (outputs)
  - On-screen icons and indicators
  - Menu items
  - NXT on-board programs
  - Test sensors and motors
  - View sensor inputs
  - Change settings

Software Development Environment

- Introductions
- Overview
- Autonomous GNC
- **Lego® Mindstorms Intro**
  - Hardware
  - **Software**
  - Programming
- Lego® Mindstorms Challenge

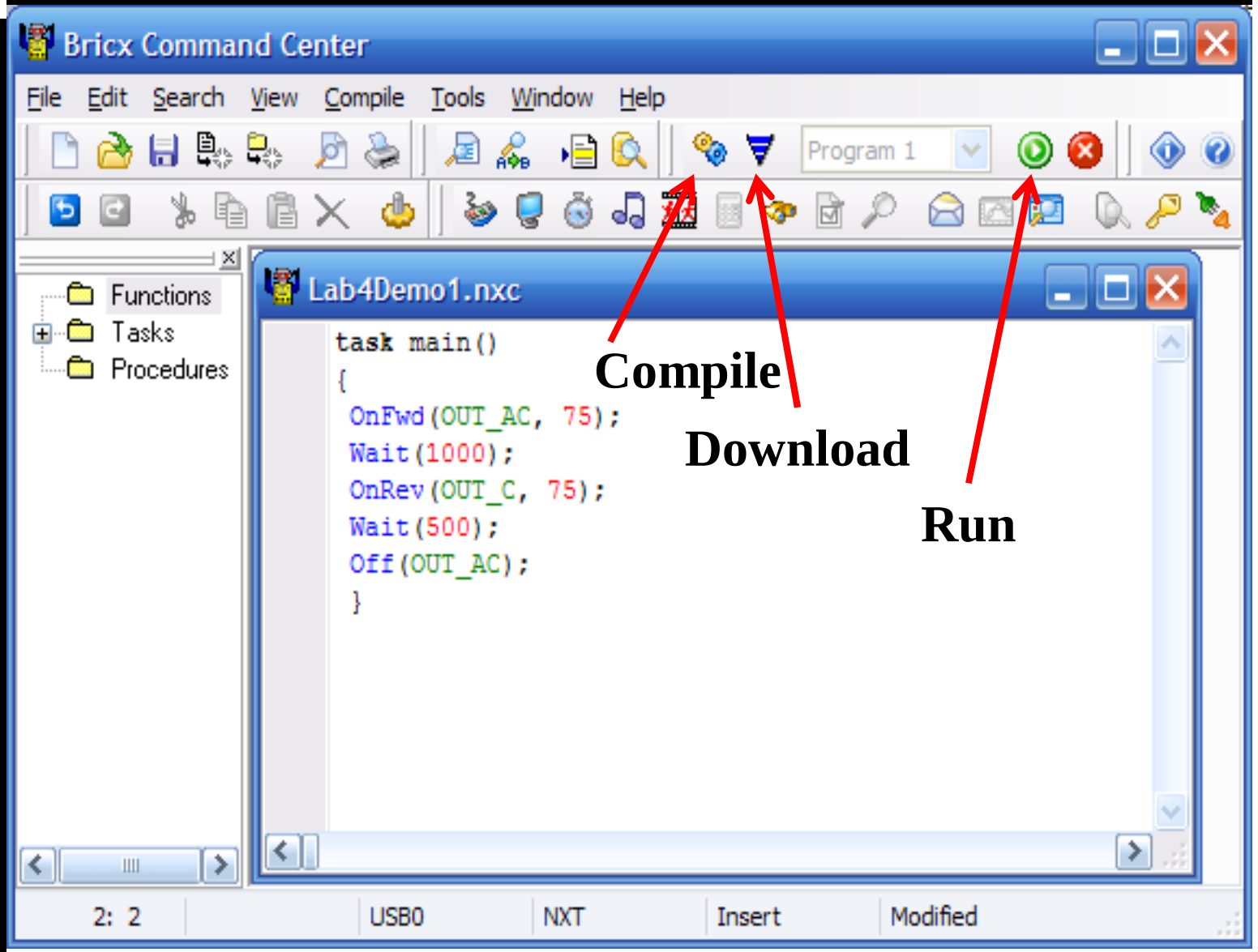
Download BricxCC: http://bricxcc.sourceforge.net/

Test release version



Software Development Environment

- Introductions
- Overview
- Autonomous GNC
- **Lego® Mindstorms Intro**
  - Hardware
  - **Software**
  - Programming
- Lego® Mindstorms Challenge



Edit→Preferences→Compiler→Switches: -v=128

Lego NXT Programming in NXC

- Introductions
- Overview
- Autonomous GNC
- **Lego® Mindstorms Intro**
  - Hardware
  - Software
  - **Programming**
- Lego® Mindstorms Challenge

1. Basic concepts:
   - tasks, main, API, constants, comments
2. Variables:
   - int, boolean, ….
   - Array, struct
3. Sensors:
   - Set sensors, read sensor values
   - Low speed sensors
4. Flow structures:
   - while
   - do…while
   - if…else
   - until…
5. Flow chart

A Simple Example Program

- Introductions
- Overview
- Autonomous GNC
- **Lego® Mindstorms Intro**
  - Hardware
  - Software
  - **Programming**
- Lego® Mindstorms Challenge


```
task main()
{
   OnFwd(OUT_A,  75);
   OnFwd(OUT_C,  75);
   Wait(4000);
   OnRev(OUT_AC,  75);
   Wait(4000);
   Off(OUT_AC);
}
```

NXC programs consist of tasks
Each program has to have a main task
The computer executes the main task

A task consists of statements
Each statement must end with ;
Braces include all statements belong to a task

Can you guess what these statements do?

onFwd, OnRev, Wait, Off
are NXT built-in functions (part of NXT API)

A More Sophisticated Version

Define constants
(values cannot be changed in the program)

```
#define MOVE_TIME  1000
#define TURN_TIME   360

task main()
{
  OnFwd(OUT_AC, 75);
  Wait(MOVE_TIME);    //Both motors drive with 75% power
  OnRev(OUT_C, 75);   //Wait time unit is ms
  Wait(TURN_TIME);    //Reverse motor C with 75% power
  Off(OUT_AC);
}
```

Comments
Computer ignores statements following //

Using Variables

Variable declaration:
Variable type
Variable name

Variable values can be defined, used, and changed in the program

Same as:
move_time = move_time +200;

```
#define TURN_TIME    360

int move_time;

task main()
{
  move_time = 200;
  repeat(50)
  {
    OnFwd(OUT_AC, 75);
    Wait(move_time);
    OnRev(OUT_C, 75);
    Wait(TURN_TIME);
    move_time += 200;
  }
  Off(OUT_AC);
}
```

Repeat the statement inside { } 50 times

Working with Arrays

- Introductions
- Overview
- Autonomous GNC
- **Lego® Mindstorms Intro**
  - Hardware
  - Software
  - **Programming**
- Lego® Mindstorms Challenge


```
int aaa;
int bbb,ccc;
int values[];

task main()
{
  aaa = 10;
  bbb = 20 * 5;
  ccc = bbb;
  ccc /= aaa;
  ccc -= 5;
  aaa = 10 * (ccc + 3); //
  ArrayInit(values, 0, 10);
  values[0] = aaa;
  values[1] = bbb;
  values[2] = aaa*bbb;
  values[3] = ccc;
}
```

Array declaration

Array initialization

Array element value assignment

Note: 1st array element index is 0

Working with NXT Sensors

- Ways to let NXT know which sensor you are using:
  - SetSensorTouch(IN_#);
  - SetSensorLight(IN_#);
  - SetSensorSound(IN_#);
  - **SetSensorLowspeed(IN_#);** *//For ultrasonic sensor*
  - **SetSensorColorFull(IN_#);** *//Color sensor*
    - SetSensorColorRed(IN_#);
    - SetSensorColorGreen(IN_#);
    - SetSensorColorBlue(IN_#);
- Ways to read sensor values:
  - SENSOR_port#; // For touch, light, sound
    eg: SENSOR_1;
  - **SensorUS(IN_port#);** // For ultrasonic
    eg: SensorUS(IN_1);
  - **ReadSensorColorRaw(IN_#, int_array);** //For color
    eg: ReadSensorColorRaw(IN_1, rgb[4]);

Color Sensor Representation &Characterization

1. Many formats to digitize color: RGB, CMYK, HSV
2. Example of a 24-bit RGB representation:
   8 bits to represent amount of red, green and blue in a color
   (0,0,0) = Black
   (255,255,255) = White
   (255,0,0) = Red
   (0,255,0) = ?

```
task main()
{   SetSensorColorFull(IN_1);
    int rgb[4], csr;
    ReadSensorColorRaw(IN_1, rgb);
    csr = rgb[INPUT_RED]+rgb[INPUT_GREEN]+rgb[INPUT_BLUE];
    NumOut (0, LCD_LINE1, rgb[INPUT_RED]);
    NumOut (0, LCD_LINE2, rgb[INPUT_GREEN]);
    NumOut (0, LCD_LINE3, rgb[INPUT_BLUE]);
    NumOut (0, LCD_LINE4, csr);
}
```

Displays the value of variable
on designated location of the NXT screen

Control Structure: while loop

```
while  (condition)      // condition outcome can be true or false
{  statements;          // Statements are executed if condition is true
}
```

```
task main()
{   SetSensorTouch(IN_1);
    while  (SENSOR_1 == 0)
    {       OnFwd(OUT_AB, 75);
            Wait(1000);
    }
}
```

Relational operators:

==	equal to
<	smaller than
<=	smaller than or equal to
>	larger than
>=	larger than or equal to
!=	not equal to

```
task main()
{   SetSensorTouch(IN_1);
    int counter = 0;
    while  (SENSOR_1 == 0 && counter < 10)
    {       OnFwd(OUT_AB, 75);
            Wait(1000);
            counter = counter +1;
    }
}
```

Logical operators:
&& (and) || (or)

Control Structure: do … while loop

```
task main()
{    int counter = 0;
     do
     {          OnFwd(OUT_AB, 75);
                Wait(1000);
                counter = counter +1;
     } while   (counter < 10)
}
```

What is the difference between these two programs?

```
task main()
{    int counter = 0;
     while   (counter < 10)
     {          OnFwd(OUT_AB, 75);
                Wait(1000);
                counter = counter +1;
     }
}
```

Control Structure: until ….

- Introductions
- Overview
- Autonomous GNC
- **Lego® Mindstorms Intro**
  - Hardware
  - Software
  - **Programming**
- Lego® Mindstorms Challenge


```
task main()
{    SetSensorTouch(IN_1);
     OnFwd(OUT_AB, 75);
     until (SENSOR_1 == 1);
     Off (OUT_AB);
}
```

Program will wait here until the condition is true

Control Structure: if … else …

- Introductions
- Overview
- Autonomous GNC
- **Lego® Mindstorms Intro**
  - Hardware
  - Software
  - **Programming**
- Lego® Mindstorms Challenge

```
task main()
{   SetSensorTouch(IN_1);
    while  (true)
    {
        if  (SENSOR_1 == 0)
        {       OnFwd(OUT_AB, 75);
                Wait (1000);
        }
        else
        {       OnRev(OUT_AB, 75);
                Wait (1000);
        }
    }
}
```

Condition

true

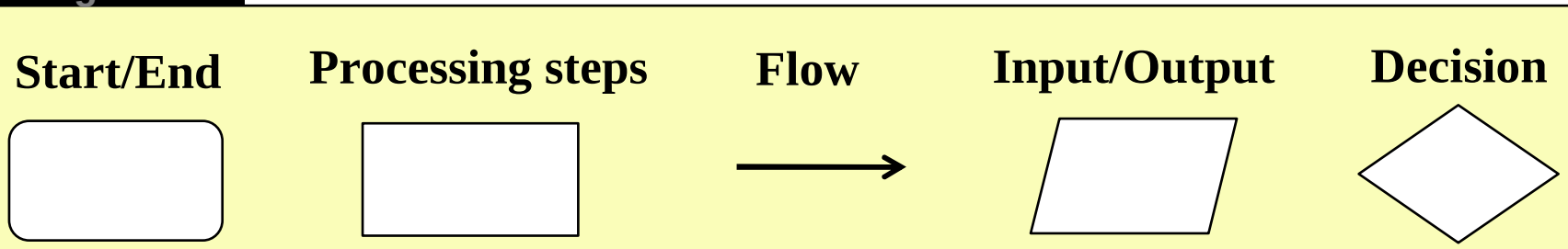
false

What Is A Flowchart?

- Introductions
- Overview
- Autonomous GNC
- **Lego® Mindstorms Intro**
  - Hardware
  - Software
  - **Programming**
- Lego® Mindstorms Challenge

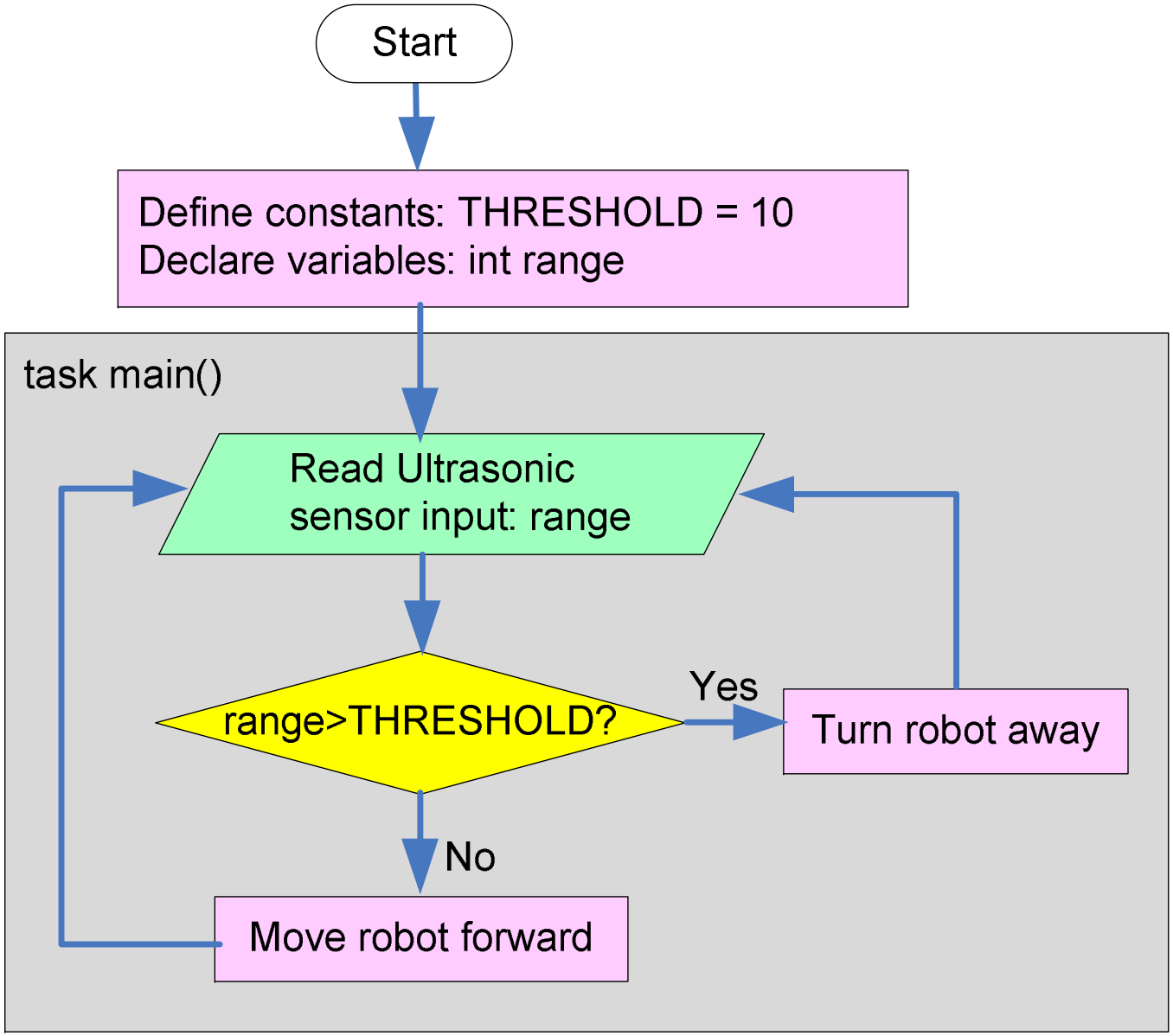
- A diagram that represents an algorithm or process, showing the steps, order of execution, and data flow.
- Gives step-by-step solution to solving a problem.
- Used in many fields to analyze/design/document/manage process & programs.

Basic flowchart symbols



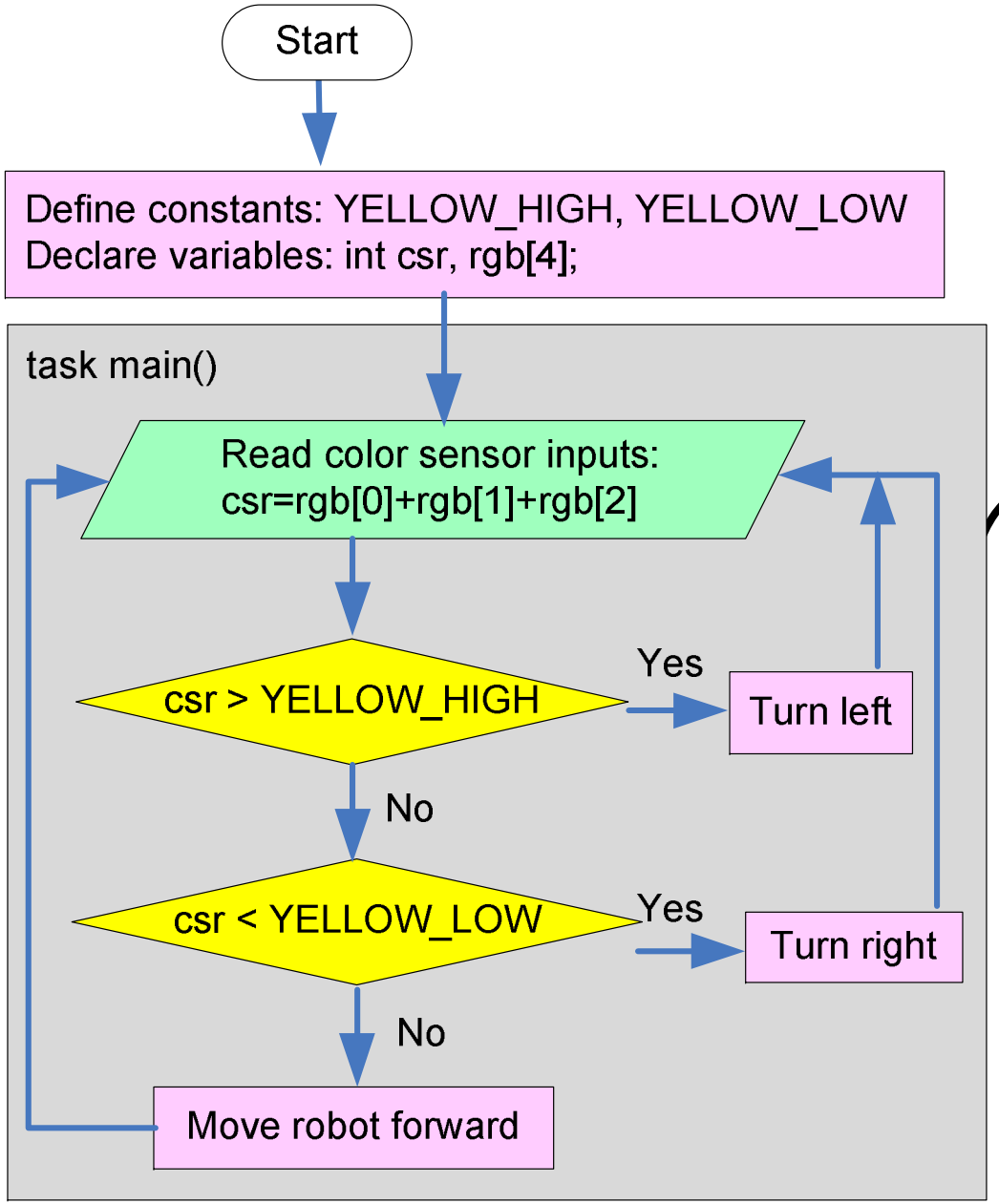
An Example Flow Chart: Obstacle Avoidance

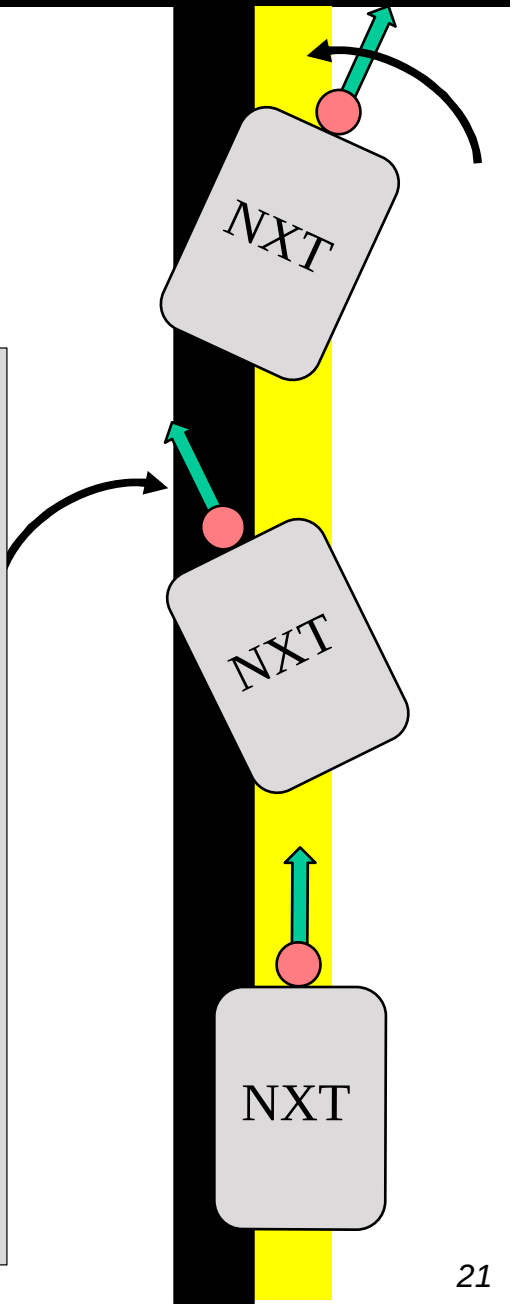
- Introductions
- Overview
- Autonomous GNC
- **Lego® Mindstorms Intro**
  - Hardware
  - Software
  - **Programming**
- Lego® Mindstorms Challenge



An Example Flow Chart: Line Following

- Introductions
- Overview
- Autonomous GNC
- **Lego® Mindstorms Intro**
  - Hardware
  - Software
  - **Programming**
- Lego® Mindstorms Challenge





Challenge 1: Build A Tri-Pot

- Introductions
- Overview
- Autonomous GNC
- Lego® Mindstorms Intro
  - Hardware
  - Software
  - Programming
- **Lego® Mindstorms Challenge**

Follow the 14-step instructions at:
http://www.nxtprograms.com/NXT2/3-motor_chassis/steps.html
to build Dave Parker's 3 Motor Chassis (30 minutes)

Programming Challenges

- Obstacle avoidance
  - Follow the flow chart shown in slide #20, use an ultrasonic sensor to sense obstacles and avoid them
- Color sensor calibration
  - Follow the example shown in slide #14 to determine a safe upper value (YELLOW_HIGH) and lower csr value (YELLOW_LOW) for the yellow color tape provided to you.
- Color line follower
  - Follow the flow chart shown in slide #21 and yellow color upper and lower limit determined above to program a yellow line follower

Additional References

- Introductions
- Overview
- Autonomous GNC
- Lego® Mindstorms Intro
  - Hardware
  - Software
  - Programming
- **Lego® Mindstorms Challenge**

- An easy to follow tutorial at http://bricxcc.sourceforge.net/nbc/nxcdoc/NXC_tutorial.pdf
- A comprehensive NXC guide at: http://bricxcc.sourceforge.net/nbc/nxcdoc/NXC_Guide.pdf.