Brooklyn College
The City University of New York

Exploring Robotics (CISC 1003)
LEGO MINDSTROMS NXT

UNIT A Lab, part 1 : Robot Construction

BEFORE YOU BEGIN:

Make sure you have all of the following materials before you start the lab:

- Lego Mindstorm NXT robot kit
- Instructions for assembling the robot

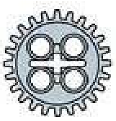
INSTRUCTIONS:

1. Assemble the robot:

- Follow the building instructions (up to page 23) and construct the robot. Some of the parts you will use include:

NXT

motor


gear

connector cables

touch sensor

light sensor

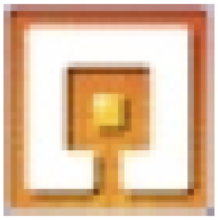
sound sensor

ultrasonic sensor

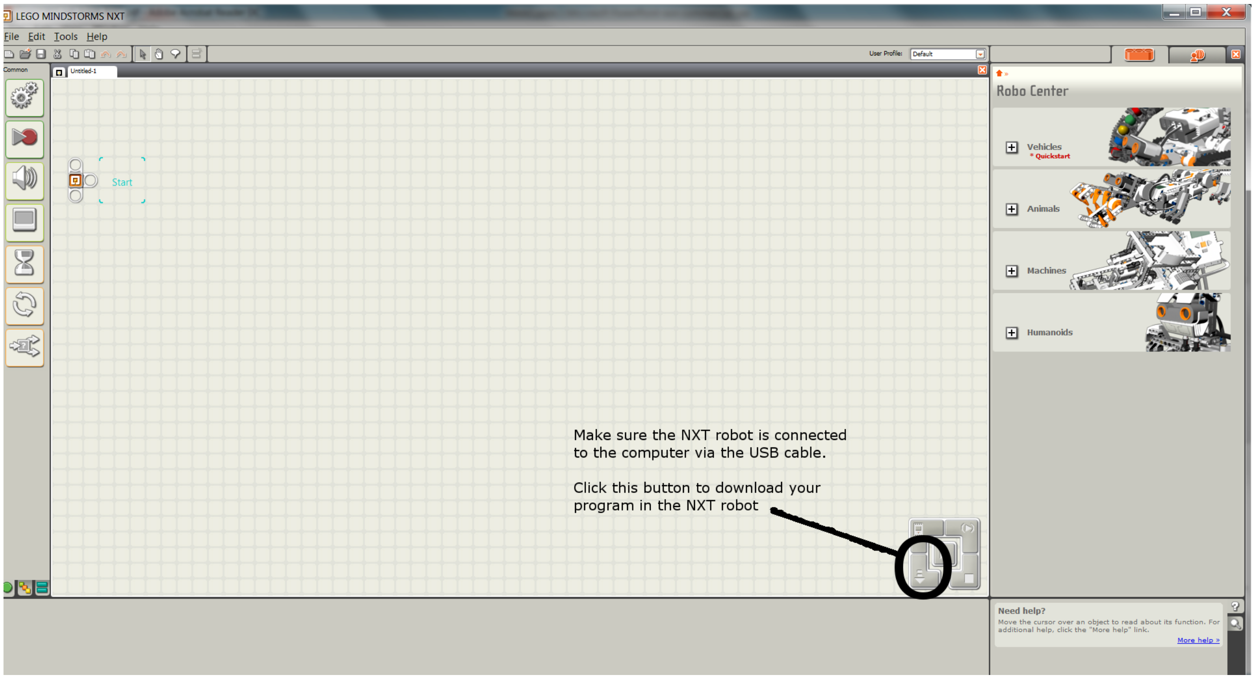
2. Download the test program:

The NXT is a **microprocessor**, a small computer which controls what the robot does. The NXT needs a **program** that contains **instructions** for controlling the robot. We will use an application called **MINDSTORMS NXT 2.0** to program your robot. **MINDSTORMS NXT 2.0** runs on a PC or a Mac and includes an environment in which you can compose programs for the robot. A program must be **downloaded** — transferred from the PC or Mac to the NXT.

A. Write a test program. For this step, you need to open **MINDSTORMS NXT 2.0** on your computer. The program icon looks like this:



B. Create a new program:



Need help?
Move the cursor over an object to read about its function. For additional help, click the "More help" link.
More help »

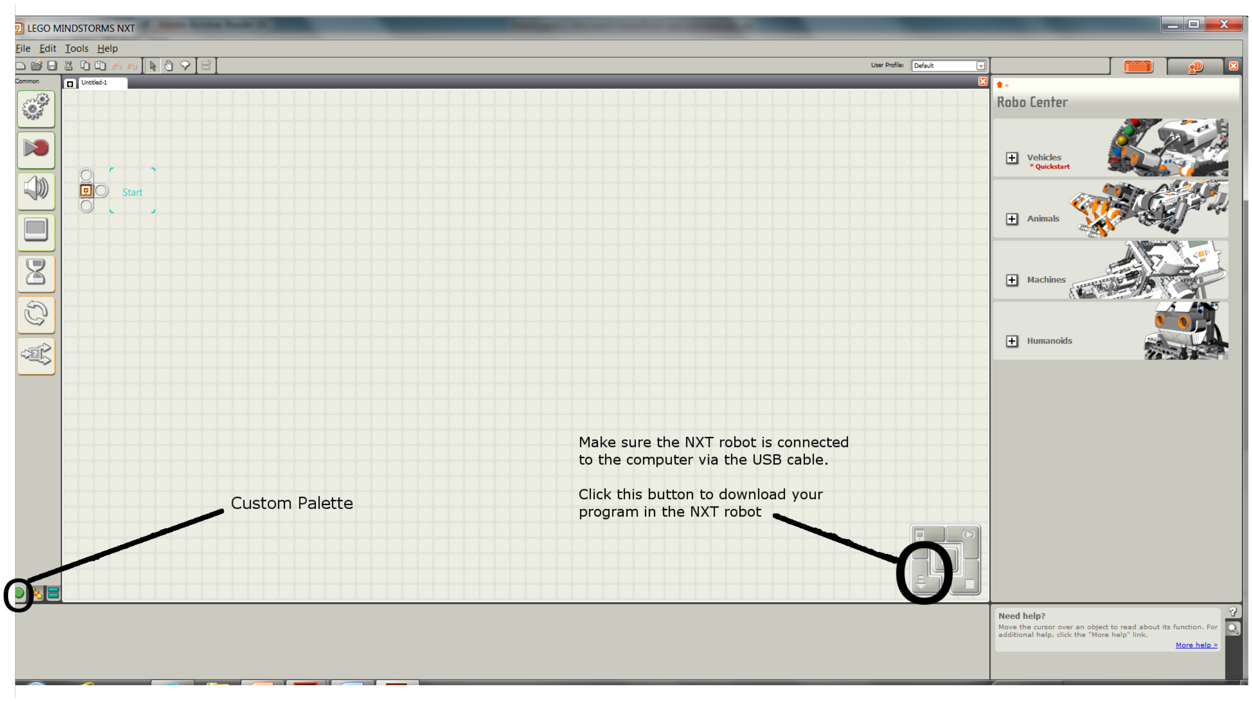
Make sure the NXT robot is connected to the computer via the USB cable.

Click this button to download your program in the NXT robot

C. The **USB Cable** facilitates downloading. It is connected to both the computer’s and NXT’s USB ports. Click on the **download arrow** to send the program to the robot.

When the download is finished, then the NXT will play a tune.

D. Go to **Common Palette** and follow the programming instructions (NXT user guide) for **Your First Program (**play sound**).**



Make sure the NXT robot is connected to the computer via the USB cable.

Click this button to download your program in the NXT robot

3. Run the test program

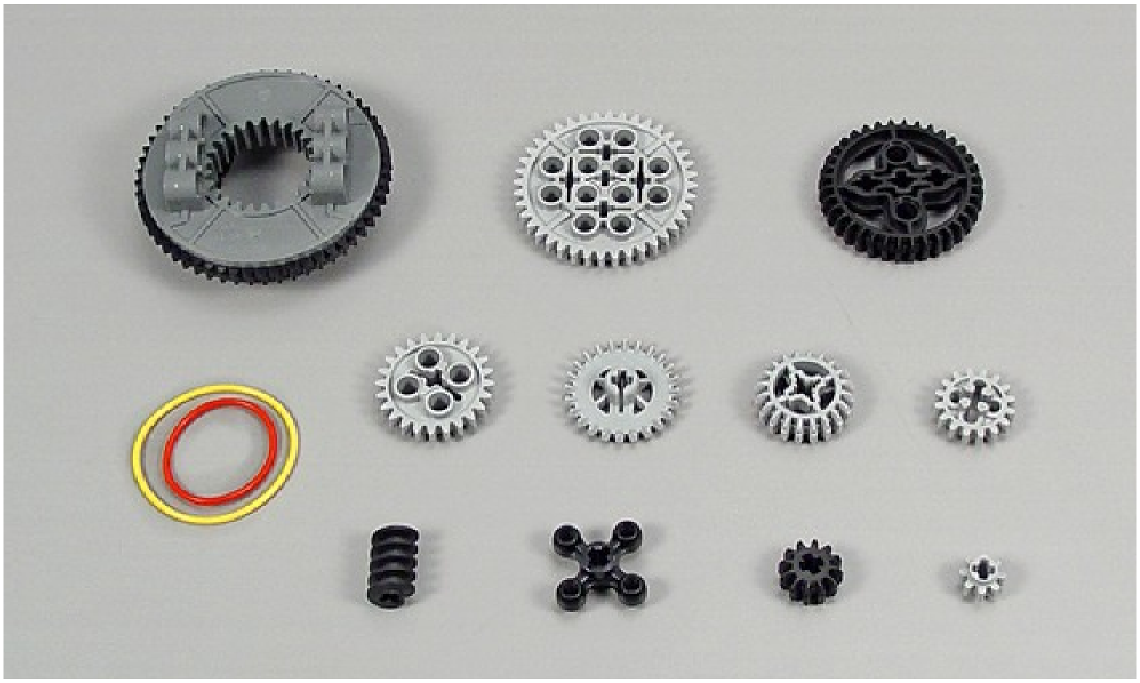
- After the program downloads successfully, put your robot on the floor and turn it on by pressing the orange button.
- When the NXT is on, use arrow buttons to select **My Files**! **Software Files**! *your-program*
- What did your robot do? Write your answer below. Be precise!

4. Gearing Exercise:

- You can use **gears** to control the robot's speed - the way in which power is transferred to the robot's wheels from the motor. This is called using **gear ratios**. Your robot kit has several gears, each of different sizes. The size is measured by the number of **teeth** on the gear. How many different gears do you have and what sizes are they?

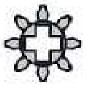
____

____

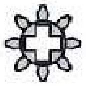


Top row (from left to right): Performance gear, 40 tooth gear, 36 tooth gear
Middle row: 24 and 36 mm belts, 24 tooth gear, 24 tooth crown gear, 20 tooth double bevel gear, 16 tooth gear.
Bottom row: Worm gear, 4-tooth gear, 12-tooth double bevel gear, 8 tooth gear.

- When two gears are meshed together, if they have the same number of teeth, then every time one gear goes around, the other does too:

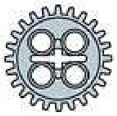


8-tooth gear

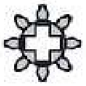


8-tooth gear

However, if one gear is larger than another, then it will rotate more slowly than the smaller gear:



24-tooth gear



8-tooth gear

In the example above, how many times does the 8-tooth gear rotate for each rotation of the 24-tooth gear?

____

In the example above, how many times does the 24-tooth gear rotate for each rotation of the 8-tooth gear?

____

- Using gears to make your robot go slower is called **gearing down**. Using gears to make your robot go faster is called **gearing up**. If the gear configuration of the robot consists of 2 gears, a 24-tooth attached to the motor and a 8-tooth attached to the wheel, is it gearing up or down?

____

- If the gear configuration of the robot consists of 2 gears, a 8-tooth attached to the motor and a 24-tooth attached to the wheel, is it gearing up or down?

____

Vocabulary:

- NXTbrick
- gear
- motor
- NXT development environment (NXT-2.0)
- USB cable
- sensor
- touch sensor
- light sensor
- sound sensor
- ultrasonic sensor
- connector cable
- beam
- tire

Brooklyn College The City University of New York

Exploring Robotics (CISC 1003)
LEGO MINDSTROMS NXT

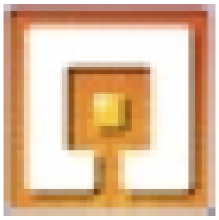
UNIT A Lab, part 2 : Introduction to NXT-2.0 and Differential Drive Control

vocabulary

- wait icon
- touch sensor
- sound sensor
- light sensor
- ultrasonic sensor
- syntax error
- logical error
- wire

Instructions

1. start up the **NXT-2.0** software
    - Find the **NXT-2.0** icon on your computer and double-click on it to start it up.



- Create a new program file for each excercise and save it frequently, on the computer or to a USB flash drive.

- Open the ’Common Palette’.

Definitions

Algorithm

An algorithm is a step-by-step instructions or a set of rules to be followed to perform an operation. It is an essential aspect of programming and a necessity to design whole and/or parts of computer programs. You can think of it as a *recipe* for writing a program.

Example: Program the robot to go in a square.

Algorithm:

```
Repeat 4 times:
Go forward
Make a 90 degree turn
```

In the above example, the algorithm doesn’t specify how the robot should go forward or how long, also doesn’t tell which way to turn, however it is easy to see that if this algorithm is followed correctly, the robot will go in a square.

Syntax error

Syntax error is a type of programming error, which occurs if your program doesn’t comform to the rules of the language. It is the result of a *bad code*.

Logic error

Logic error is a type of programming error, which occurs when the program doesn’t perform the expected operation due to wrong expression of the ideas. These types of errors are harder to fix than syntax errors, since they conform with the language rules therefore the system cannot provide any feedback as to where an error of this type might be in the program. The best way to aviod these types of errors is to design the algorithm before actually writing the program.

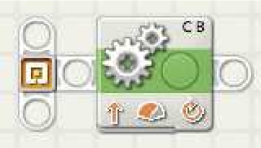
Differential drive control

Differential drive refers to controlling a robot using two independent motors. If both motors have the same speed and direction, the robot moves forward or reverse. All other cases causes robot to turn with a speed and direction, as a result of the difference between the motor speeds (hence the term, ’differential’).

Exercises

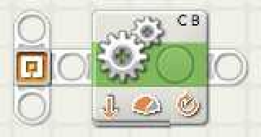
Answer the questions and demonstrate each program to your instructor.

- **Drive Forward:**



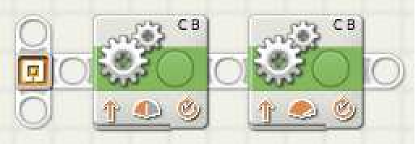
What will the above program do? How can you change the speed?

2. **Drive Reverse**:



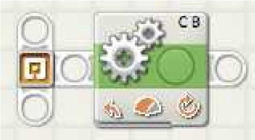
What will the above program do? Modify the program to backup without stopping.

3. **Accelerate:**



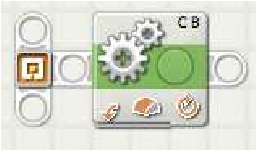
What will the above program do?

4. **Curve Turn** :



What will the above program do? Do the same using one motor only.

5. **Point Turn**:



What will the above program do? How long (in seconds) does it take for the robot to make a 90 degrees turn?

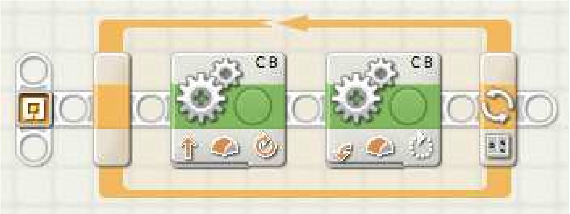
Loops:

6. Loops are structures that allow your program to repeat specific tasks.

7. All loops have the same basic structure; a set of commands that you want the program to repeat and a condition that tells when to stop. Some loops run for a specific number of times. These are the types of loops that we will cover in this lab. Other types of loops run until a certain condition occurs, for example, go forward until you hit a wall. We will discuss these types of loops later.

From the **User Guide** look for the *Loop Block* icon for further information.

- **Drive in Square:**



What will the above program do? How would you change the program to make a bigger square?

Consider the effects the effects of some additional factors:

1. How do you think different wheels will affect the robot’s ability to turn? Does it matter?

2. Does the surface on which the robot is turning matter?

Brooklyn College
The City University of New York

Exploring Robotics (CISC 1003)
LEGO MINDSTROMS NXT
© 2017

UNIT B Lab, part 1: Robot Gears and Program Control

vocabulary

- gear ratio
- rotational speed
- revolutions per minute (rpm)
- angular velocity
- torque
- idler gear
- robot control

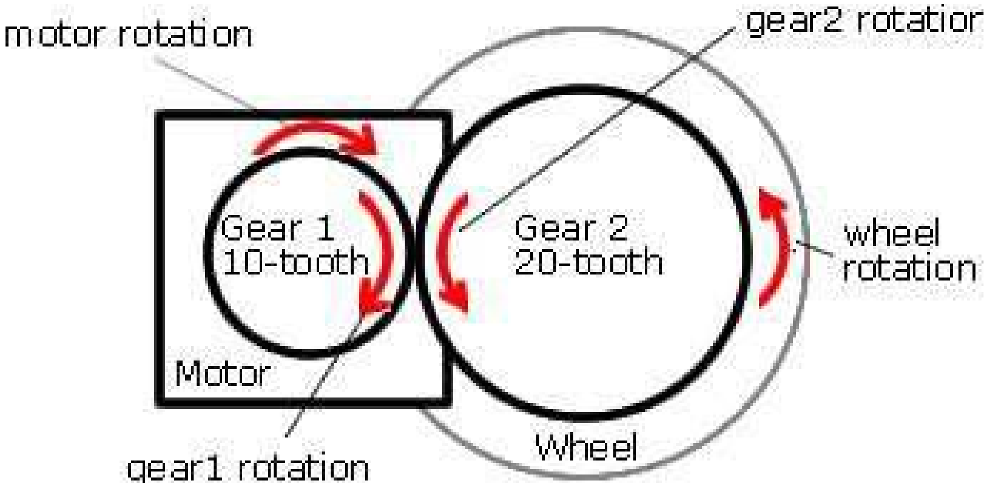
INSTRUCTIONS:

Assemble the robot arm with mallet

- Follow the building instructions (between pages 50-61) and construct the arm with mallet.

Gears:

In our previous lab sessions, we have used gears for reversing the direction and changing the speed of rotation of the wheel. Consider the gear configuration below:



In the image above, there are 2 gears: 10-tooth and 20-tooth. Gear 1 is connected to a motor, and Gear 2 is connected to a wheel. The gear connected to the motor is called the *drive* gear. Assume the motor is spinning clockwise at 100 rpm (revolutions per minute). How many times does the wheel rotate in a minute and in which direction (clockwise or counterclockwise)?

Answer:

- For each revolution of the motor, the drive gear (Gear 1, 10-tooth) will rotate once in the same direction as the motor (i.e., clockwise in this case).

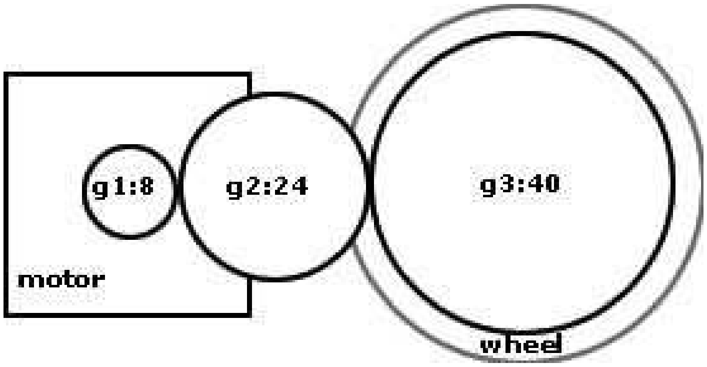
- In each revolution, Gear 1 will touch 10 of the teeth in Gear 2 (each tooth in Gear 1 will interlock with a tooth in Gear 2 and force Gear 2 to turn). Thus, Gear 2 will turn half a rotation in the opposite direction (i.e., counterclockwise in this case). The *gear ratio* is defined as the ratio of the gear being driven (in this case, Gear 2) to the gear doing the driving (in this case, Gear 1). In our example, this ratio is “Gear 2” to “Gear 1”, or 2 : 1 (read “two to one”), meaning that for every two revolutions of the drive gear (Gear 1), the interlocking gear (Gear 2) will turn one revolution. This can also be stated as saying that for one rotation of Gear 1, Gear 2 will complete 1/2 a rotation.

- Since Gear 2 is attached to the wheel, the wheel will also turn 1/2 a rotation for each rotation of the motor, counterclockwise in this case.

- The motor spins 100 revolutions per minute. How many revolutions does the wheel rotate in a minute? We can find the answer by multiplying the number of revolutions the motor completes in a minute by the wheel revolution rate that we found above (i.e., 1/2). Therefore the answer is :
1/2 * 100 = 50 revolutions, counterclockwise.

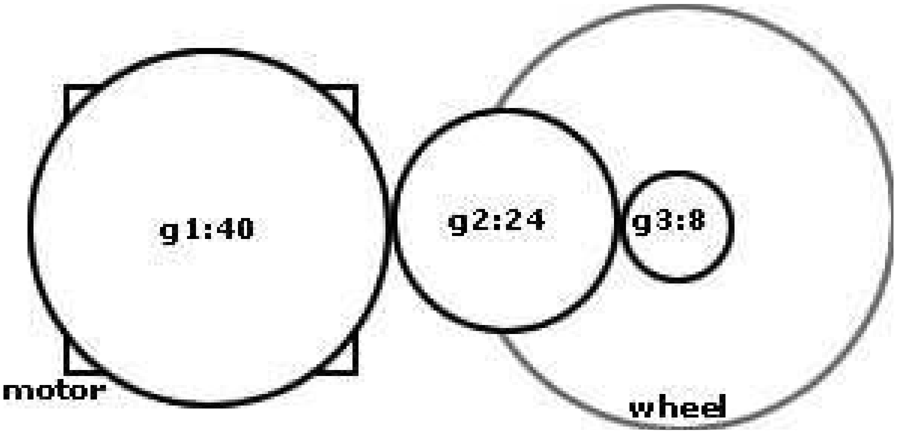
For each configuration below, determine how many times each gear rotates in a minute and in which direction (clockwise or counterclockwise). Assume that the motor is spinning clockwise at 200 rpm.

1.



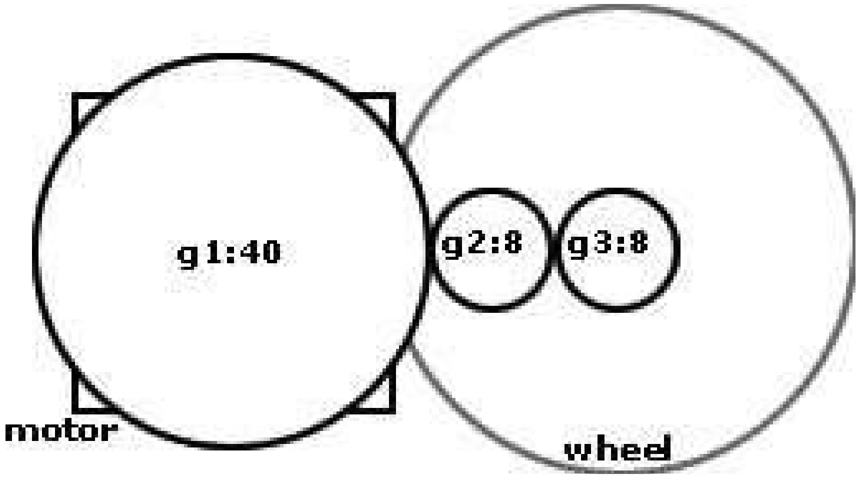
Answer:

2.



Answer:

3.



Answer:

____

____

____

Rotation control

You need the robot arm with mallet for this section. If you haven’t built it yet, follow the instructions at the beginning of the lab. Your motor, attached to port A, should be connected to a 36-tooth gear, which is meshed with a 12-tooth gear, connected to the mallet.

1. Program your robot to move motor A forward, for one full rotation. How many times the mallet rotated?

   ____

2. Program your robot to move motor A forward, for 180 degrees. How many times the mallet rotated?

   ____

3. Switch the position of 12-tooth gear with the 36-tooth gear and re-run the program in question. How many times the mallet rotated?

   ____

4. After making the switch in question 3, re-run the program in question 2. How many times the mallet rotated?

   ____

Brooklyn College
The City University of New York

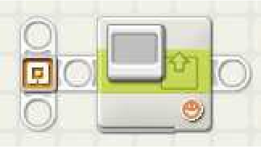
Exploring Robotics (CISC 1003)
LEGO MINDSTROMS NXT
© 2017

UNIT B Lab, part 2: Get to know the robot!

Exercises from the common palette:

Before you start the exercises, read the following definitions and instructions. Answer the questions and demonstrate each program to your instructor.

- **Use Display:**



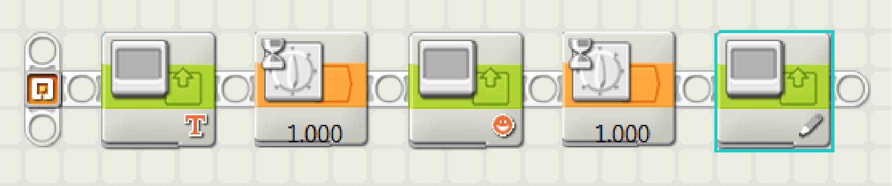
What will the above program do? Try to display your name on the screen.

____

____

____

Create a program that displays your name, a smily face and a line:



What will the above program do? Try to display all three items on the screen.

____

____

____

Multitasking

– Sometimes you may want to have your robot do two things at once. This is called **multi-tasking**. For example, if you were programming your robot to play the game of soccer, then you would want the robot to be able to look for the soccer ball and at the same time, you would want it to look out for obstacles.

– We human beings are superb at **multi-tasking**! You can *listen* to your favorite music while *walking* down the street and *eating* candy all at the same time. In fact, we are always multi-tasking — hearing and seeing and feeling and breathing all at once.

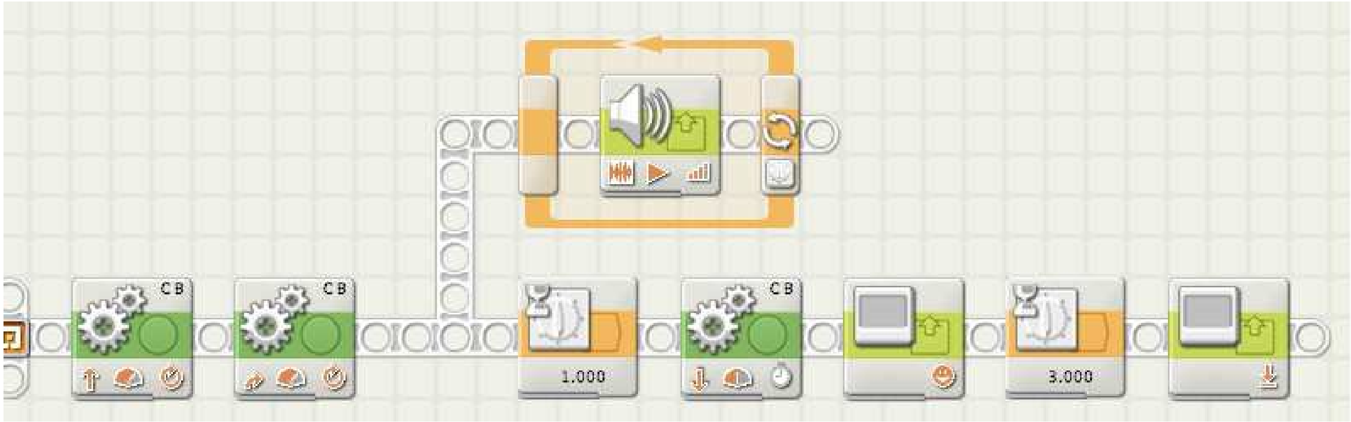
– Your computer is also **multi-tasking** by having multiple windows open at once. You can browse in the internet (e.g. using Mozilla) and play music (e.g. running iTunes) at the same time.

– The NXT also has the ability to multi-task, meaning it can execute more than one task at a time.

– Multi-tasking in programming refers to running parts of the program in parallel. This means they are executed simultaneously. Although multi-tasking has advantages, if several of these tasks try to control the same resource *hardware conflicts* occur. One task commanding output B (left wheel) to go forward and another commanding it to go backward is an example to this situation. This will become more of an issue as our programs get more complex. You need to think about possible hardware conflicts when you write a multi-tasking program. make sure that multiple tasks do not try to control the same hardware at the same time.

The exercise below is introducing multitasking in NXT:

- Parking Bay:



What will the above program do? Why the sound icon separated from the main branch of the program?

____

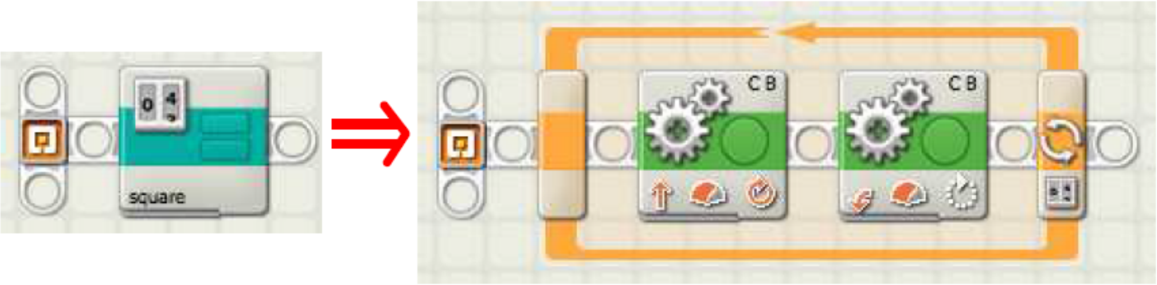
____

____

Functions, Subroutines, Blocks

In programming, a common practice is to isolate certain parts of the program that are repeated often. These isolated parts, depending on the programming language, are referred as *functions, subroutines, methods, etc. the* NXT software refers to them as *blocks*. The main idea is to define blocks once and call them in the program whenever needed. As an example, think of a 'backup block' which makes the robot move backwards for 1 second and turns 90º. Once this backup behavior is defined, it can be called from anywhere in the program. With this approach, the program becomes simpler and more manageable, once a block is defined correctly. Following exercise shows how to define blocks in the NXT software.

- **Programming block:**



What will the above program do? What are the advantages of using blocks?

____

____

____

Brooklyn College The City University of New York

Exploring Robotics (CISC 1003)
LEGO MINDSTROMS NXT
© 2017

UNIT C Lab: Introduction to NXT Sensors

vocabulary

- wait icon
- touch sensor
- sound sensor
- light sensor
- ultrasonic sensor

Instructions:

1. Adding Sensors

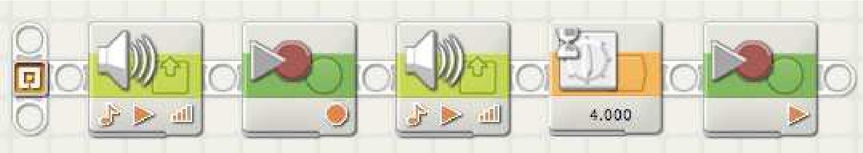
Follow the building instructions to add to your robot:

- touch sensor (pg. 40 - 45)
- sound sensor (pg. 24 - 27)
- light sensor (pg. 32 - 39)
- ultrasonic sensor (pg. 28 - 31)

Exercises from the common palette:

Before you start the exercises, read the following definitions and instructions. Answer the questions and demonstrate each program to your instructor.
For the exercise below, refer to **NXT-2.0 Software Icon Reference** for *Record/Play* icon for further information.

- **Action Replay:**



What will the above program do? What does the program record?

____

____

____

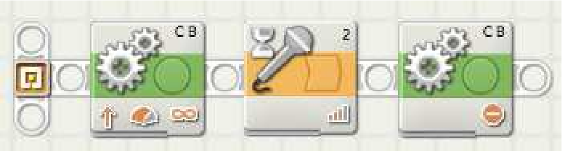
For the exercise below, refer to **NXT Software Icon Reference** for *Wait* icon for further information.

Wait Icons
So far, the wait icons that you used were for a specific duration. This lab introduces different ways of using wait icons, particularly to wait for an event to happen. The following exercises will show how to use a wait icon for a particular type of sensor.

Sound Sensor

- Sound sensor is used for detecting the level of noise in the environment. It does not particularly tell the source of the noise or where it's coming from, only the sound level in decibels (dB).
- **To view the Ultrasonic sensor's value:**
  Turn on the NXT. On the main menu select **VIEW** by pressing the right arrow button until the selection is visible. Next, select 'Sound dB' option. Finally, select the port that the sound sensor is plugged into. The displayed number is the percentage of the noise level that the sensor is capable of detecting.

- **Detect Sound** :



What will the above program do? What is the average percentage of the maximum detectable noise level (90dB) in the room?

____

____

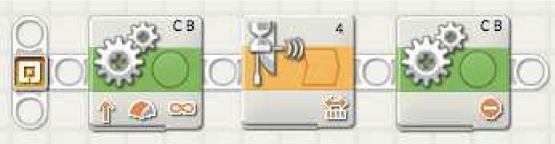
____

Ultrasonic Sensor

- Ultrasonic sensor is a range sensor that measures the distance between the sensor and the closest object, using ultrasound waves. It does this by emitting an ultrasound signal and measuring the time the waves travel to and back from the object.

- **To view the Ultrasonic sensor's value:**
  Turn on the NXT. On the main menu select **VIEW** by pressing the right arrow button until the selection is visible. Next, select either 'Ultrasonic inch' or 'Ultrasonic cm' options. Finally, select the port that the ultrasonic sensor is plugged into. The displayed number is the distance between the sensor and the nearest objects in inches or cm, based upon your previous selection.

- **Detect Distance**:



What will the above program do? Try to make the robot stop as close to an obstacle as possible.

____

____

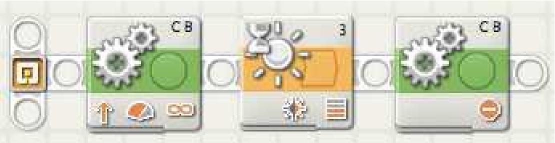
____

Light Sensor

- The light sensor has a transmitter and a receiver. It transmits infrared light (also called "IR", e.g. your TV remote control uses IR to talk to your TV), which bounces off objects and then returns in the direction of the receiver. The receiver records a value indicating how much light it read, which basically tells you about the brightness of the object that the light sensor is pointing at. The light sensor produces a value between 0 and 100, where 100 means very bright and 0 means very dark.

- **To view the light sensor's value:**
  Turn on the NXT. On the main menu select **VIEW** by pressing the right arrow button until the selection is visible. Next, select either 'Reflected light' or 'Ambient light' options. Finally, select the port that the light sensor is plugged into. The displayed number is a percentage, 0 represents the darkest and 100 the brightest values.

- **Detect Dark Line** :



What will the above program do?

____

____

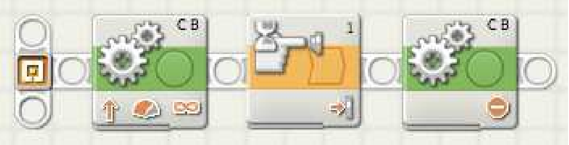
____

Touch Sensor:

- The touch sensor is a simple switch which has two states; *on* or *off*. By building a bumper around this sensor, the robot can detect better if it gets in contact with an obstacle.

- **To view the touch sensor's value:**
  Turn on the NXT. On the main menu select **VIEW** by pressing the right arrow button until the
  selection is visible. Next, select 'Touch' option. Finally, select the port that the touch sensor is plugged into. The displayed number is '1' for contact, '0' for no contact.

- **Detect Touch** :



What will the above program do?

____

____

____