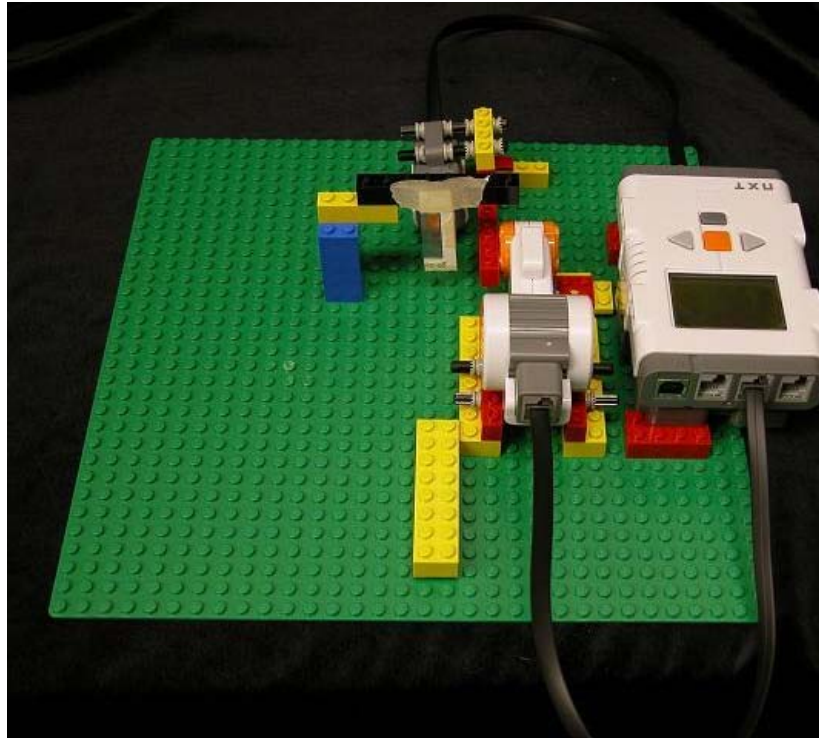# Modeling X-Ray Diffraction with the LEGO® NXT

Josiah D. Miller, Mike R. Hoehn, Richard  B. Villarreal, Dean J. Campbell
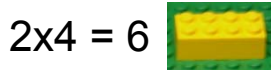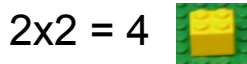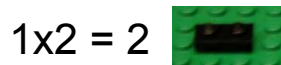*Bradley University*
**2011**

The goal of this project is to adapt the basic principles of operation of a powder X-ray diffractometer and construct a simpler and safer diffractometer for students to use built from LEGO® bricks.  The theory behind both the LEGO® model and X-ray diffractometers are almost the same.  X-ray diffractometers are used to measure the spacing between planes of atoms using the Bragg equation. X-rays are necessary for the X-ray diffraction experiment because they have enough energy to easily penetrate materials and they have a sufficiently small wavelength to be diffracted by the small spacing between planes of atoms. The LEGO® model uses visible laser light, which is more safe than the X-rays used in the powder X-ray diffractometer. The LEGO® model uses a NXT module connected to a motor to rotate an arm on which a light sensor is mounted.  After construction of the LEGO® model and programming of the NXT module, its laser can be used to find the spacing between the rows of a diffraction grating.  To do this, laser light is shined through the diffraction grating and the sensor arm is rotated through the path of light diffracted from the grating.  The data is extracted from the NXT and Microsoft® Office Excel® is used to graph light intensity as a function of sensor angle.  The angle between the light intensity peaks in the graph can be used in the Fraunhofer equation to find the spacing between the rows of the grating.
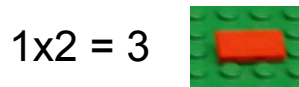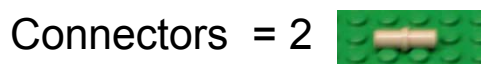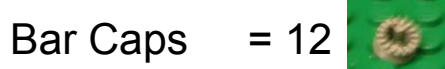
1

# Parts List

**Bricks:**

1x4 = 2 

1x8 = 1 

2x2 = 4 

2x4 = 6 

2x8 = 1 

**Plates:**

1x2 = 2 

1x4 = 2 

2x4 = 3 

**Flat Tiles:**

1x2 = 3 

**Bricks, with holes:**

1x2 

1x6 

**Cross axles:**

m6 = 2 

m8 = 2 

**Miscellaneous:**

Bar Caps     = 12 

Connectors  = 2 
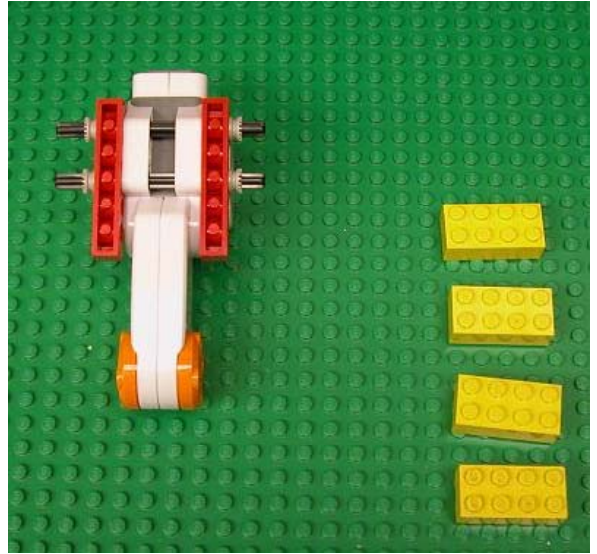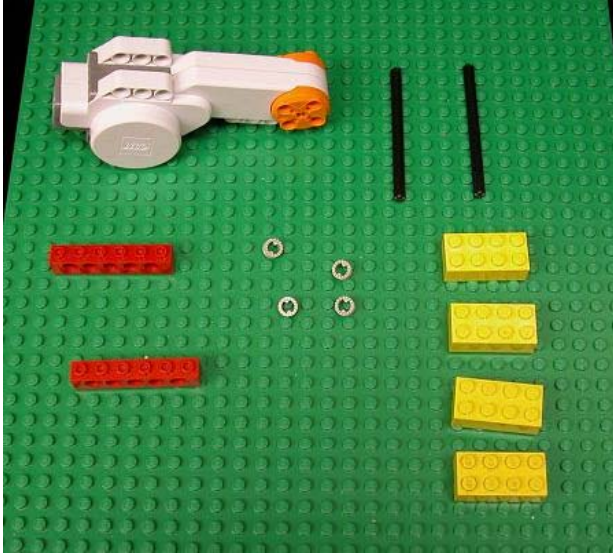
**Note**: A full Mindstorms® NXT kit (#8527) is required for this instrument
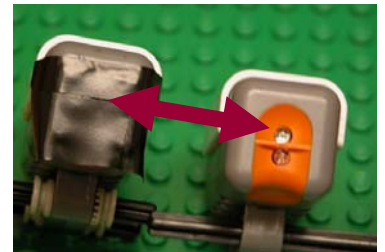
---

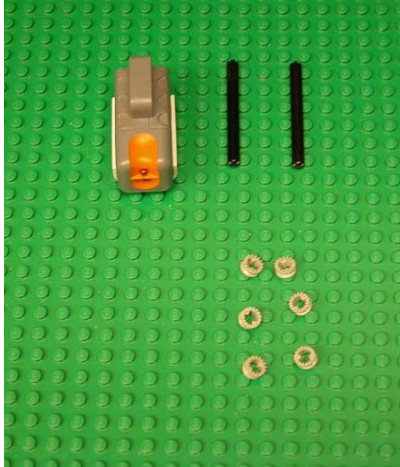# Non-LEGO® Parts

Electrical Tape

Diffraction Grating

Laser

---
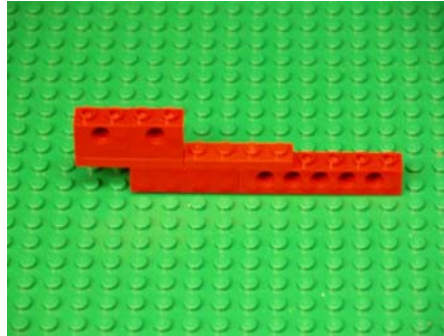
# Motor Mount Construction

# Grating Mount Construction







**Note**: Make sure the diffraction grating is aligned so the grooves are parallel to the base and perpendicular to the axis of rotation.

# Arm and Sensor Construction

**Note**: The slit between the pieces of electrical tape is positioned in front of the top sensor and reduces the overall light intensity reaching the sensor.
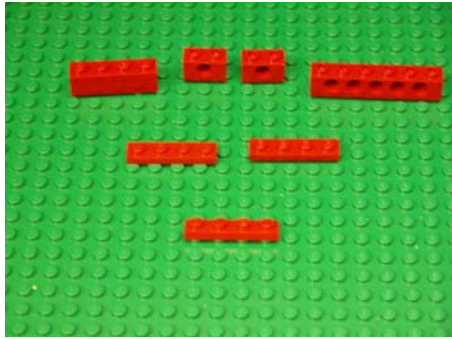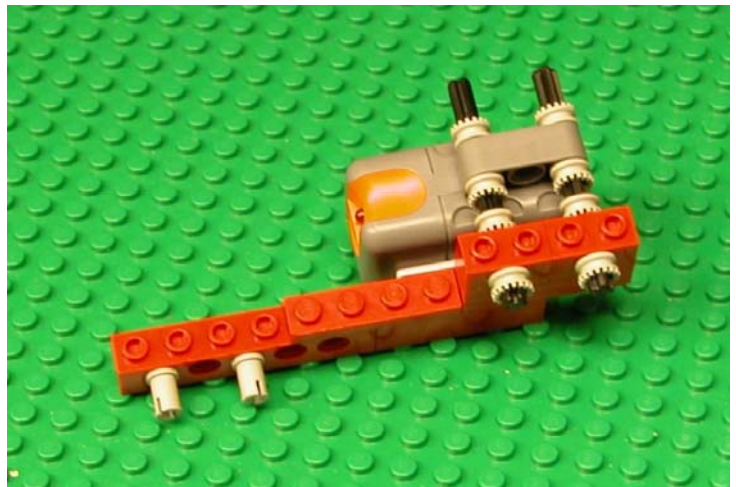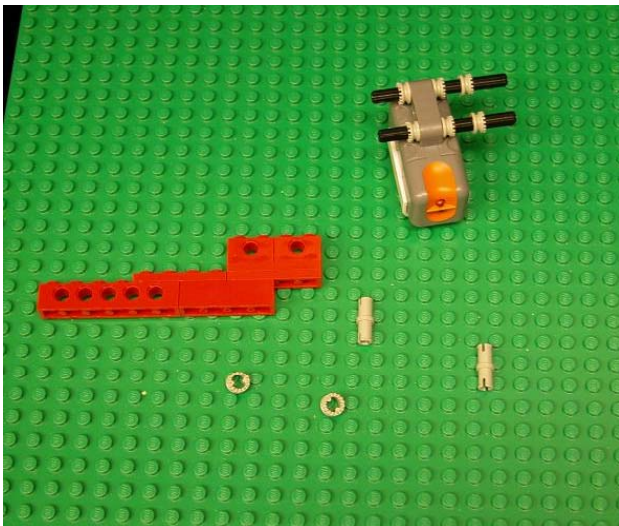
# Sensor Support Construction
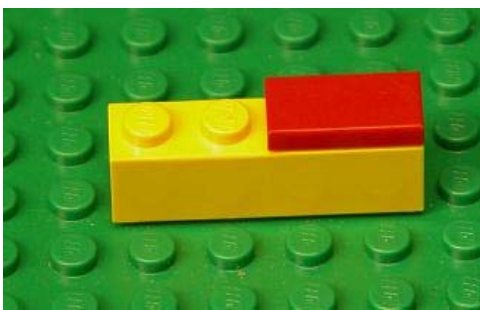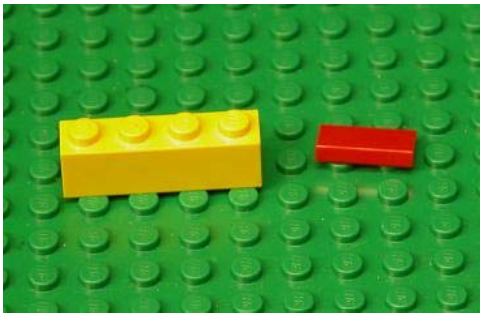




# Swivel Support Construction
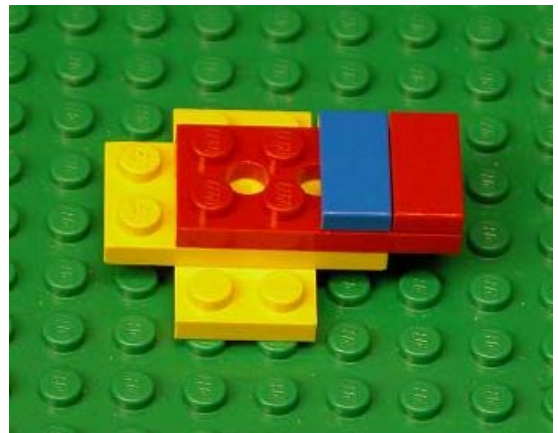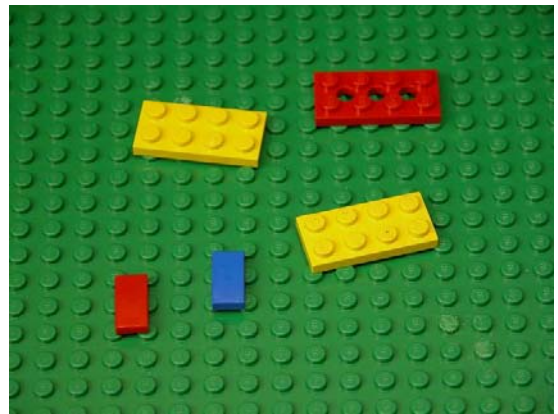




# Laser Mount Construction
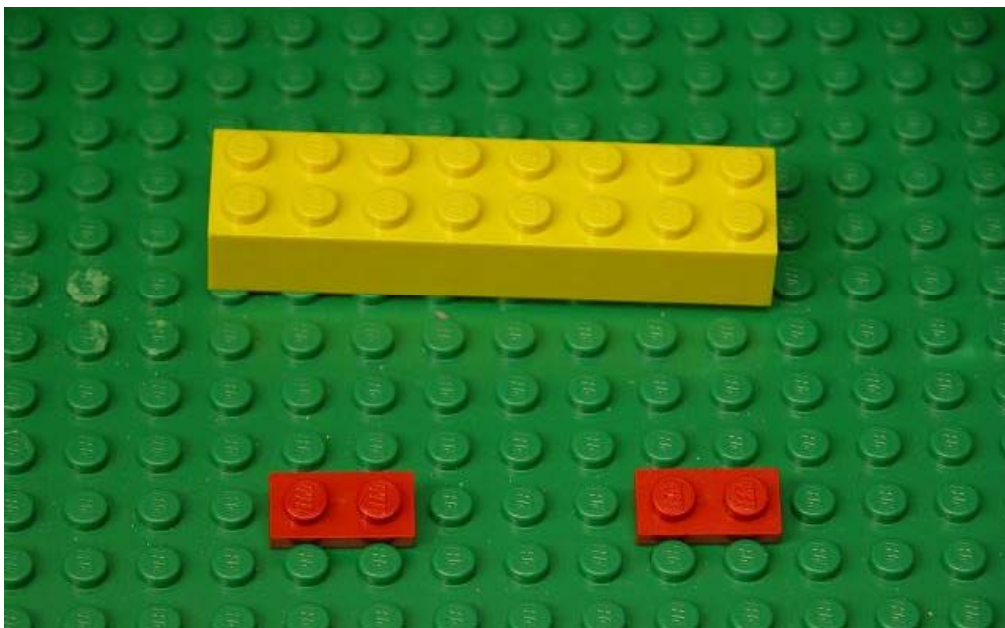
# Support and Motor Placement
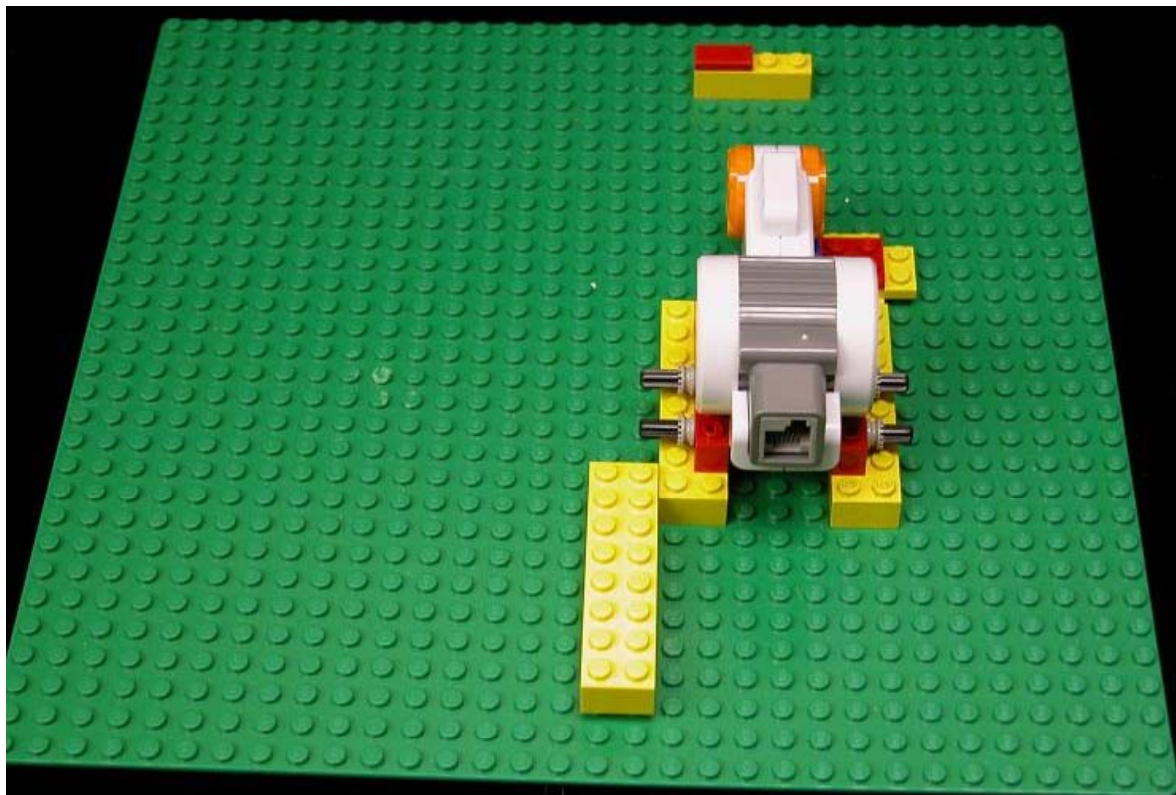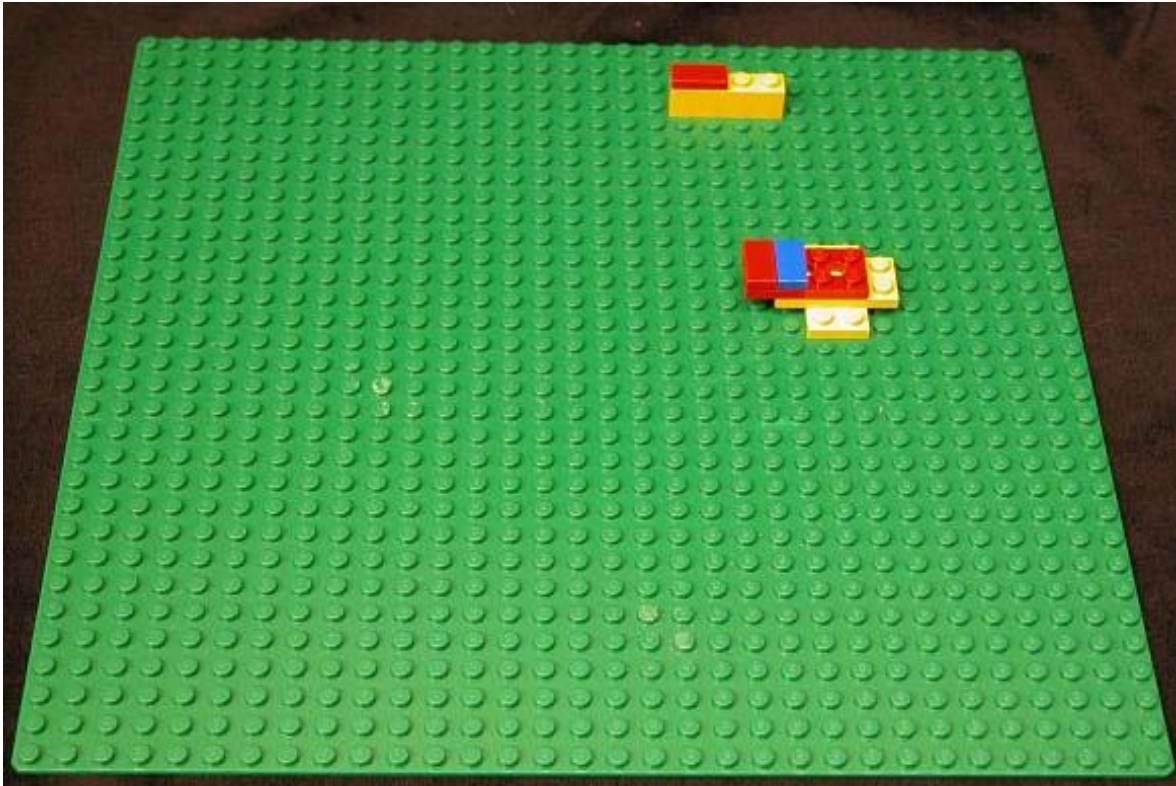
# Arm and Grating Placement
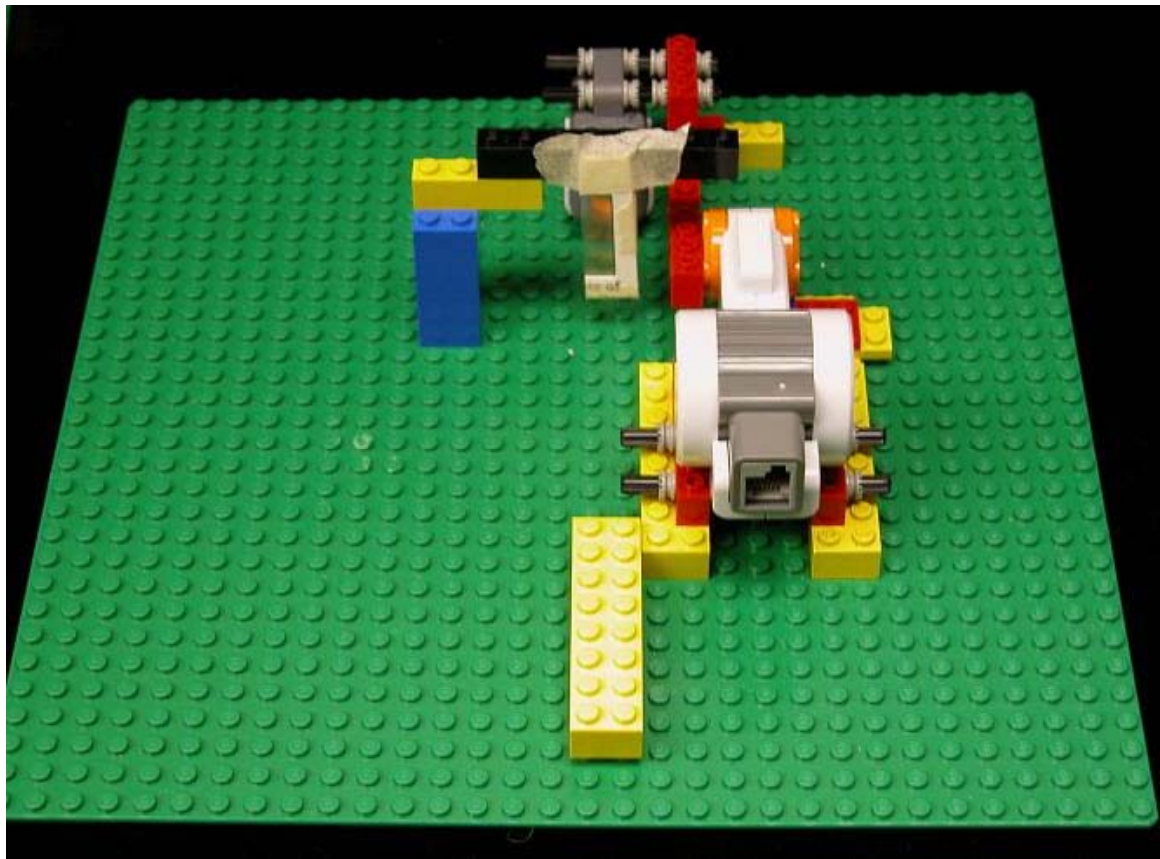
# NXT  and Laser Placement
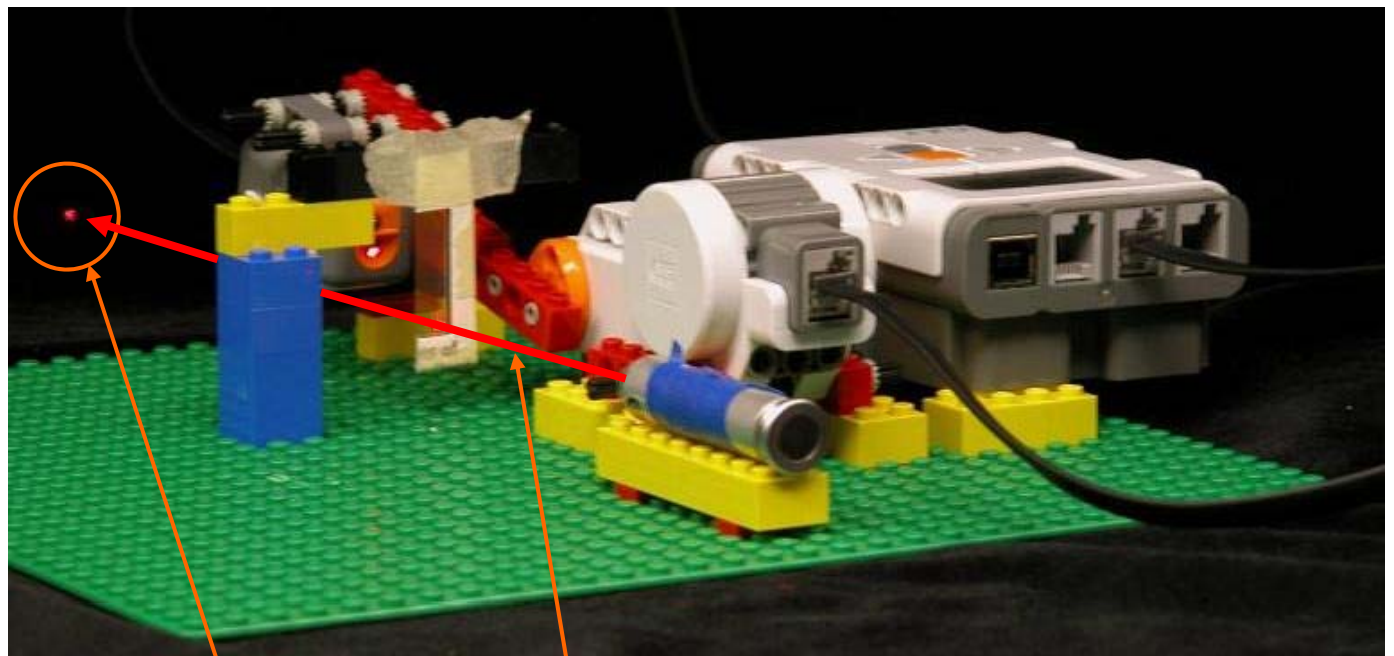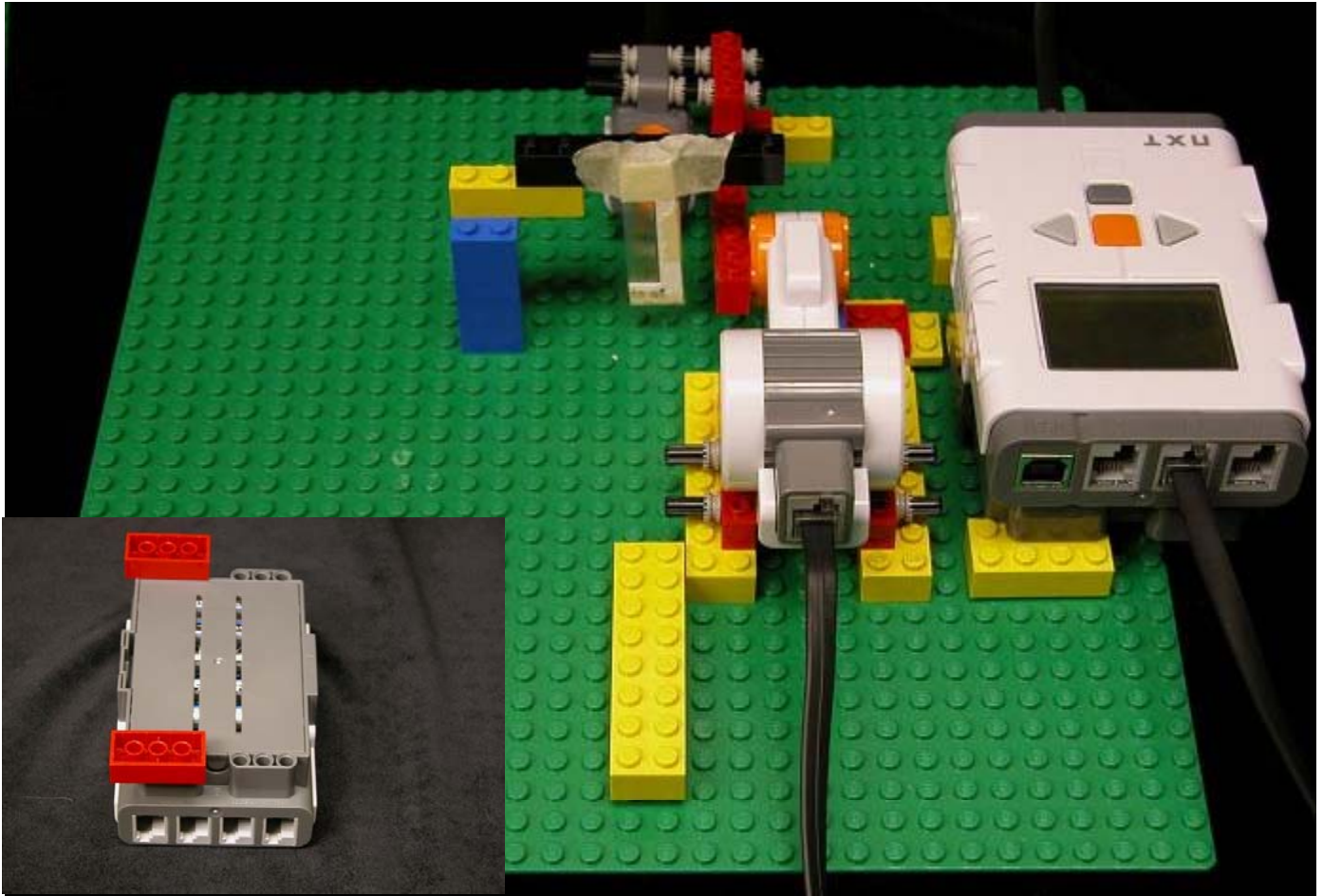


Beam path

Caution: Avoid direct eye exposure to laser beam.

9

# Incident and Diffracted Beam Path

## LEGO Laser Diffractometer

First order light

Detector

Source

$\phi$

Zeroth order light

Incident laser beam

## Powder X-ray Diffractometer

Incident X-ray beam

$\theta$

$\theta$

Source

Sample Stage

Detector
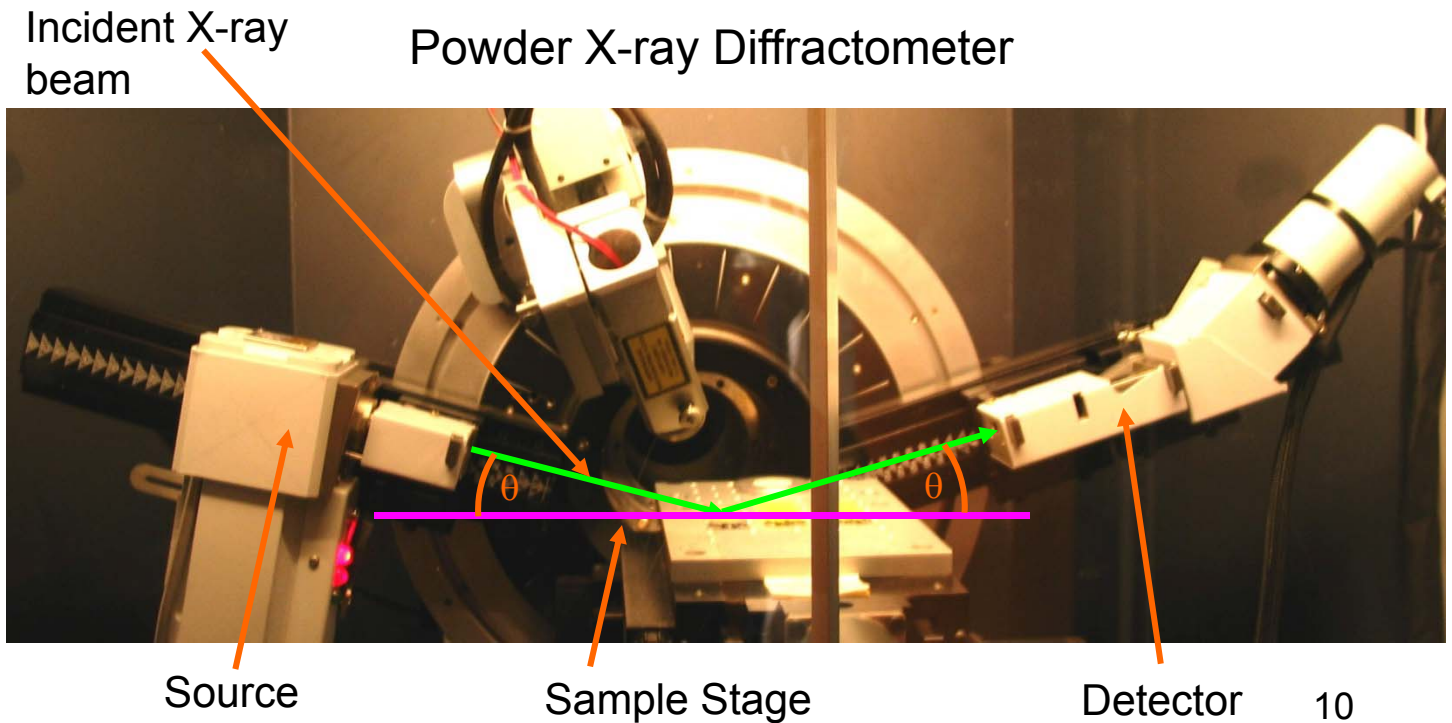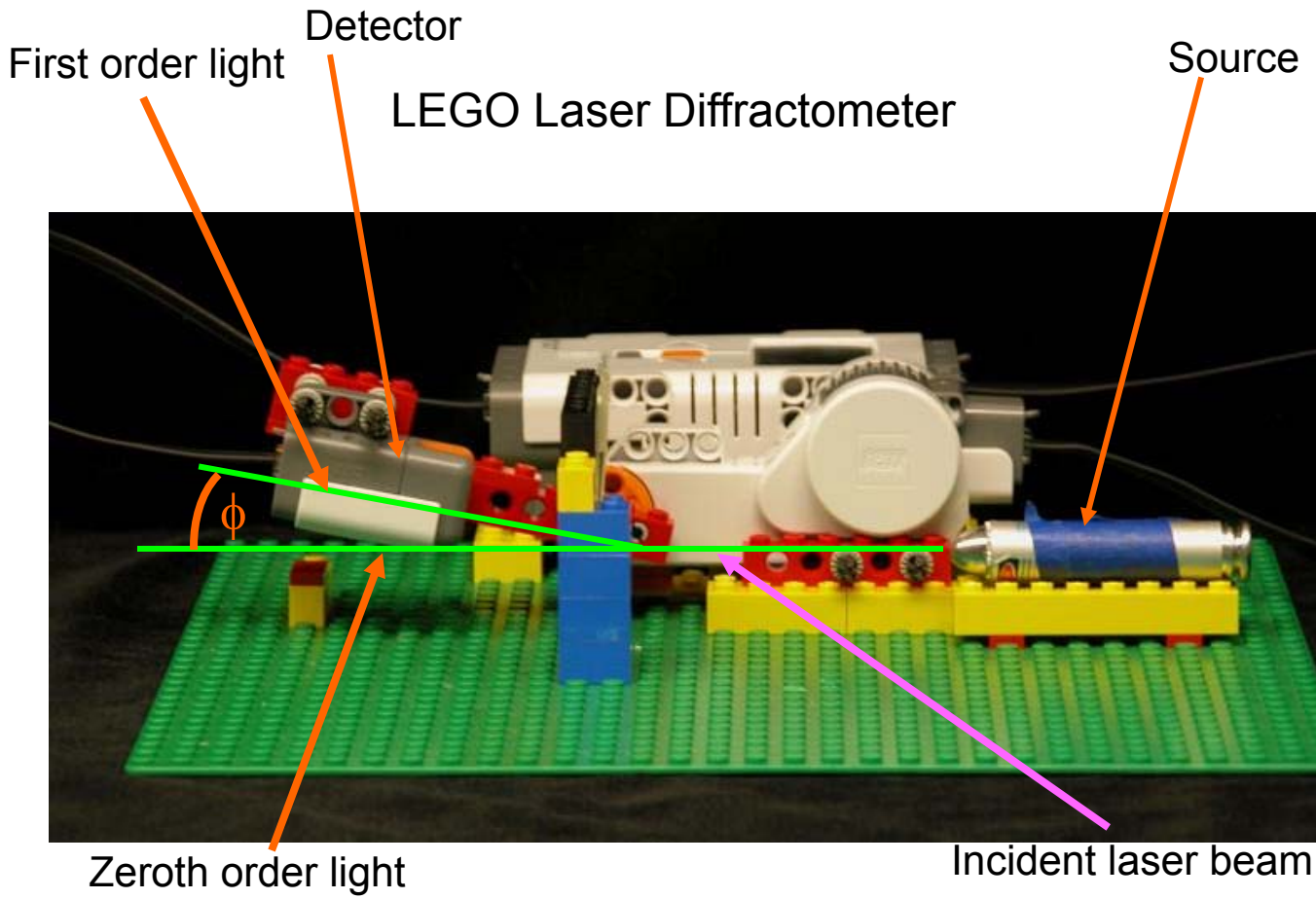
# Programming
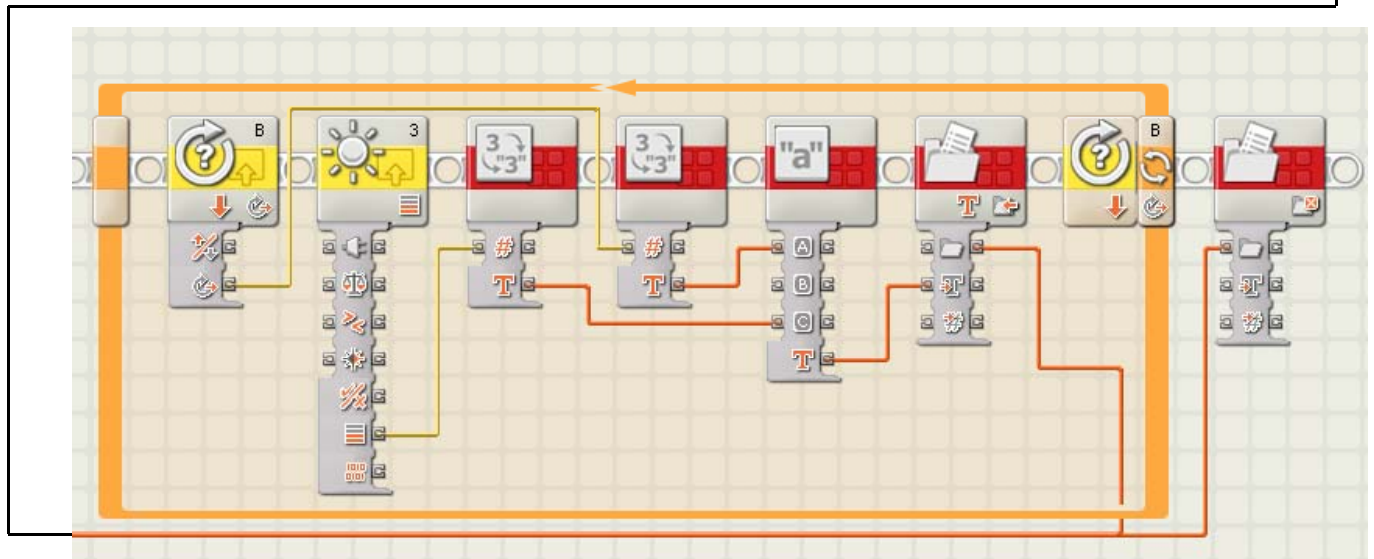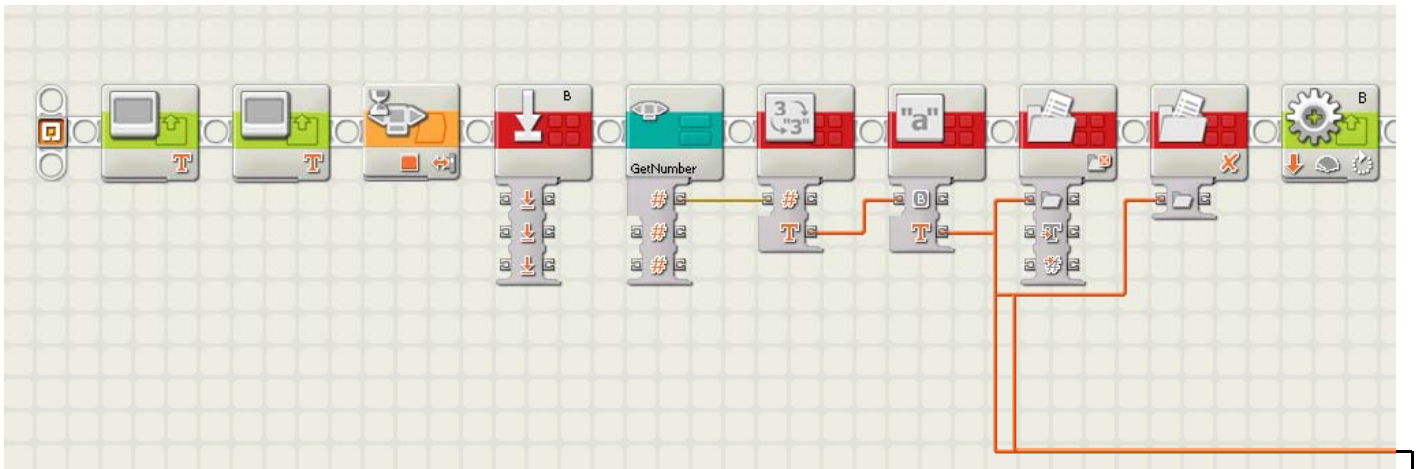


**Note**: Make sure that all wires are connected properly between blocks!
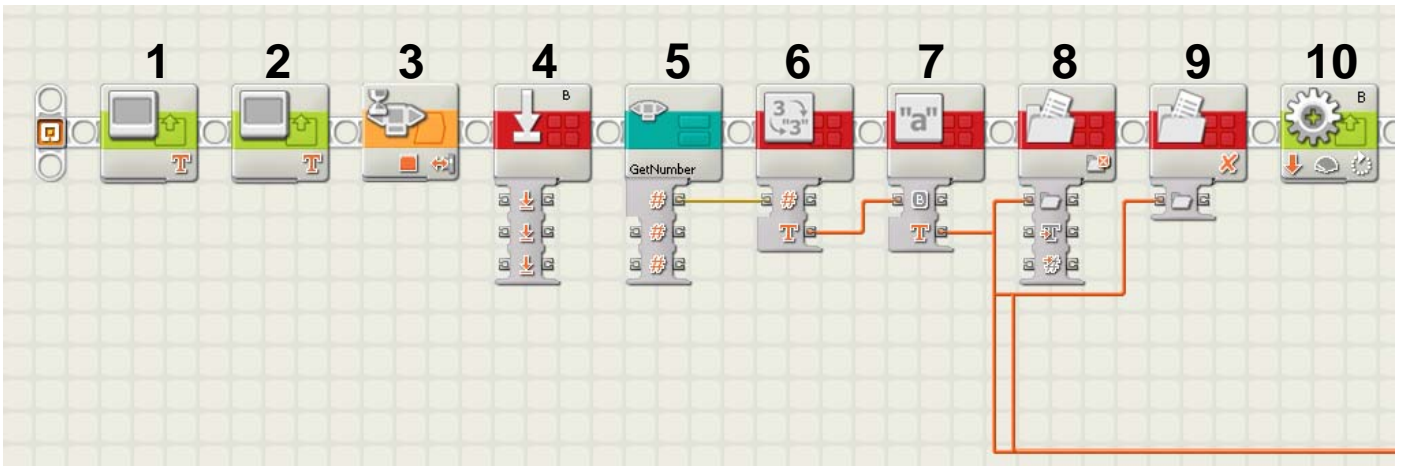
1: Display, in text box type "select file #"

2: Display, in text box type "<R> to start"

3: Wait, NXT buttons

4: Rest Motor, designate which port the motor is plugged into NXT

5: Get Number, subroutine (see page 14)

6: Number to Text

7: Text

8: File Access

       Action: Close

9: File Access

       Action: Delete

10: Motor

       Power: 12  (for fully charged batteries; batteries with lower charge require higher power settings)

       Degrees: 50

       Direction: down

       Action: Constant

12

       Make sure to designate which NXT port that the motor is plugged into (e.g. A, B, C)
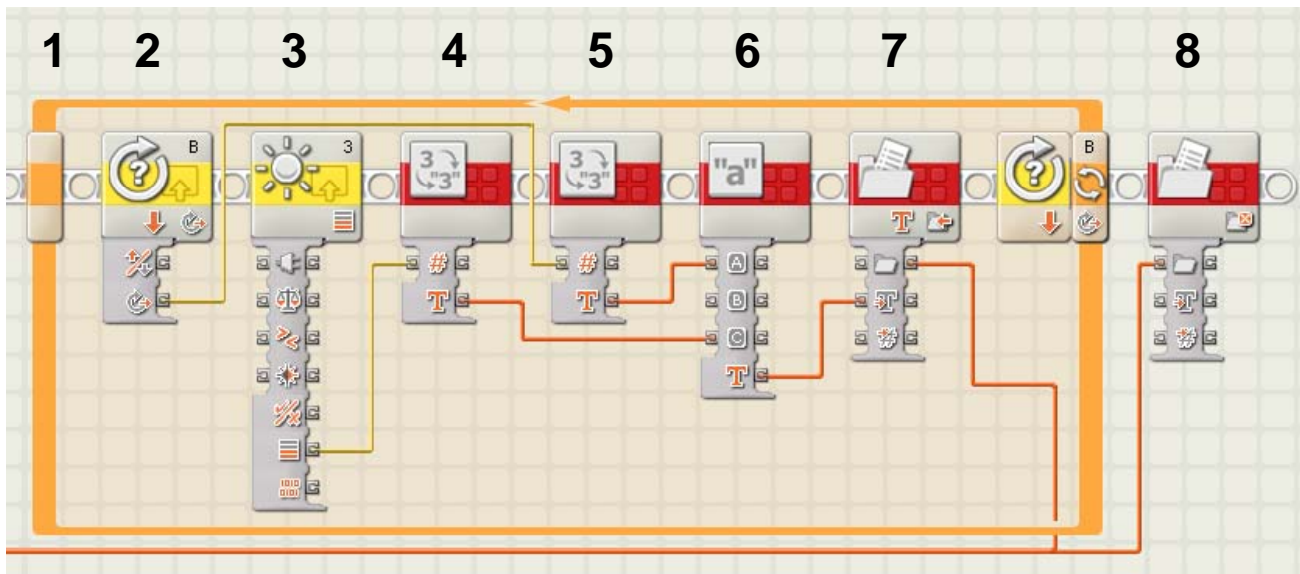
1: Loop, select rotation sensor

2: Rotation Sensor, designate which port the motor is

plugged into

3: Light Sensor, designate which port the sensor is plugged

into, also to have the light side selected

4: Number to Text

5: Number to Text

6: Text

7: File Access

Action: Type

Type: Text

8: File Access

Action: Close

# GetNumber Subroutine Procedure

*The GetNumber block will be the last block created in the code needed for the diffractometer. After **all** other blocks are created, use the following instructions.*

- Create two number variables by clicking "Define Variables" under the edit menu.
  - Name them "_cursor" and "_result"
- Between the 4th and 5th blocks of the code, create 5 variable blocks
  - Block 1 = Click "Number 1" in the list and set action to Read
  - Block 2 = Click "Number 1" in the list and set action to Read
  - Block 3 = Click "_result" in the list and set action to Write
  - Block 4 = Click "_cursor" in the list and set action to Write
  - Block 5 = Click "_result" in the list and set action to Read
- Make a connection from the left input of Block 3 to the input of Block 1
- Make a connection from the left input of Block 4 to the input of Block 2
- Make a connection between Block 5 and the left number input on the Number-to-Text block immediately to the right
- Hold shift and click Block 3, Block 4, and Block 5 (referring to the newly created variable blocks)
- Click "Make A New My Block" under the edit menu.
- Name the block "GetNumber", click next, pick any picture you want, then hit finish.
- The newly created custom block will be in place of Blocks 3, 4 and 5
- Delete the two variable blocks immediately before the newly created block
- Double click on the GetNumber Block
- Rename the value plugs by double clicking on the name above the plugs, and name them the following.
  - Rename the plug connected to the first variable block as "seed"
  - Rename the plug connected to the second variable block as "curLoc"
  - Rename the plug connected to the third variable block as "Result"
- To complete the code for the "GetNumber" block, follow the following GetNumber diagrams and create the remaining blocks between the second and third variable blocks.
- After the GetNumber block is completed, you must delete, then re-add the number block to the original code in order to apply the changes you have made.

Here's a MyBlock that prompts the user to input a number using a simple menu system on the LCD screen. It depends on another smaller MyBlock that just waits for a keypress, returning a -1, 0, or +1 depending on which key is pressed.

brdavis@iusb.edu  11-Oct-2006

Record the initial value of the number (the "seed" value) and the initial cursor location (examples: 9 highlights the "9", while 12 starts the cursor on the "+")

Display the header text…

…and display the current value of the variable…

…& the menu text.

Use the cursor variable to calculate where exactly on the screen to draw the "cursor" circle, with a 6 pixel offset between each character and a 7 pixel fixed offset at the beginning (due to where the menu string has been drawn)

Retrieve a keypress and beep

GetKey

Get Number, Subroutine

We're done; wire out the result

Result
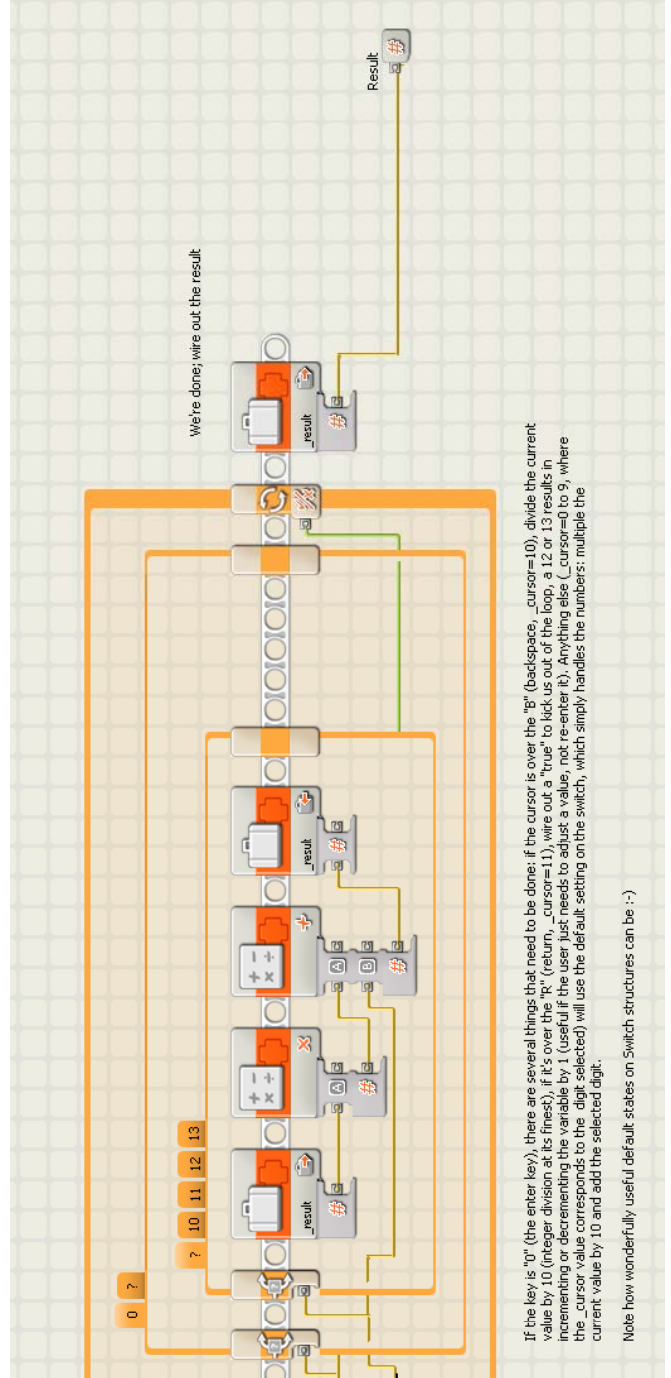
If the key is "0" (the enter key), there are several things that need to be done: if the cursor is over the "B" (backspace, _cursor=10), divide the current value by 10 (integer division at its finest); if it's over the "R" (return, _cursor=11), wire out a "true" to kick us out of the loop, a 12 or 13 results in incrementing or decrementing the variable by 1 (useful if the user just needs to adjust a value, not re-enter it). Anything else (_cursor=0 to 9, where the _cursor value corresponds to the digit selected) will use the default setting on the switch, which simply handles the numbers: multiple the current value by 10 and add the selected digit.

Note how wonderfully useful default states on Switch structures can be :-)

seed

curLoc

_cursor

result

_result

? 10 11 12 13

0 ?

15

# Get Key, subroutine



Keypress

Pass the value out of the MyBlock.

Check to see if _key is greater than -5; if so, flag true which will kick out of the loop.

Etc.

If the center button has been pressed, set _key = 0, etc.

Now, loop until the variable _key contains a valid return (in other words, one of the buttons has been recorded as pressed)

If the left-arrow is pressed, set _key = -1, and wait for the key to be released

Set the returned key to a value of -5 (i.e., invalid)

brdavis@iusb.edu    11-Oct-2006

16
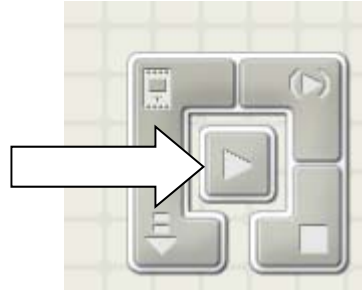
# Downloading Program to NXT from a Computer

Press indicated button to download the program to the NXT



## Running the Program from the NXT

Press the orange square button. Then select a number for file name and press the orange button again.  Now select "R" and press the orange button to run.

Note: Keep hands away from the sensor arm because getting the arm caught could damage the motor.

## How to Upload NXT Data to a Computer



- Press indicated button

- Select Memory Tab

- Choose Other

- Highlight and Upload file number you assigned on the NXT

- Save file (Notepad document) to desired location

17

- Open notepad document
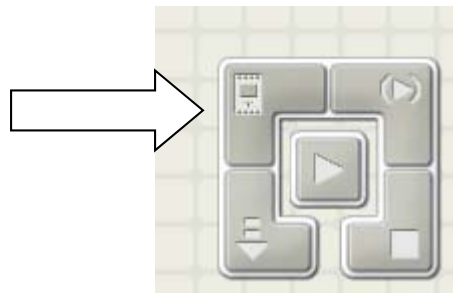
# Data Analysis

-Data will appear in a notepad document in the form of "angle, intensity"

-Copying and pasting these numbers into an Microsoft Excel will NOT work.  Instead, open the notepad document from within Excel

  -Make sure to have "All files" selected in the "Types of files" pull down menu in the "Open" window

  -Click "Open" and a window will appear called "Text Import Wizard"

  -Make sure the "Delimited" radio button is selected, and click "Next"

  -Check the "Comma" box and the window should demonstrate the numbers separated into two columns, and click Finish

-With your data in two separate columns, they can be represented in a graph to help you calculate your angle of diffraction
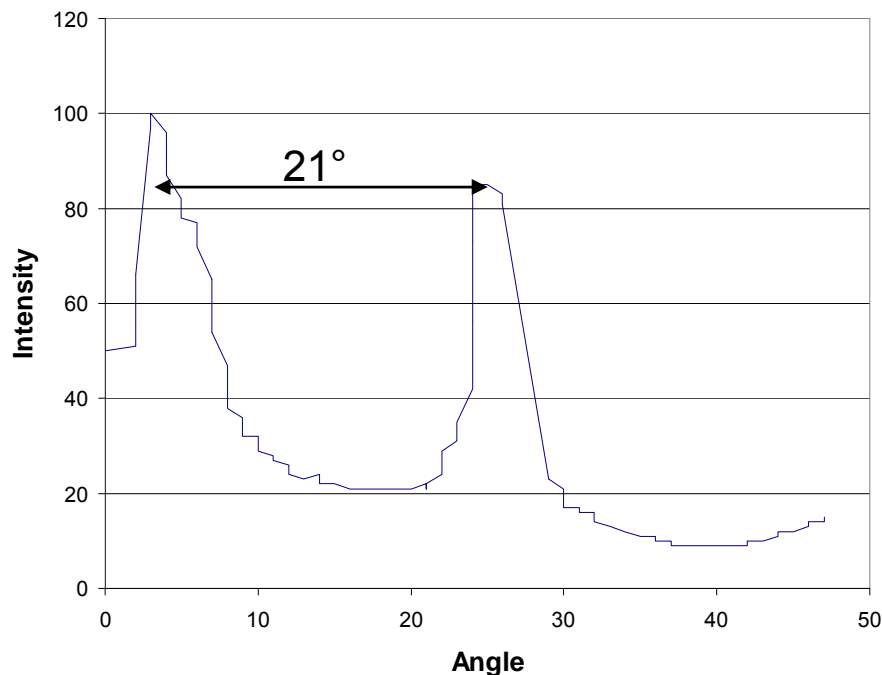


- By starting the sensor below the zeroth order beam, your results will indicate a peak for the beam path and a peak for the first order of diffraction.
- The difference between these two numbers is the angle Φ.

# Calculations

The Fraunhofer equation can be used to calculate the row repeat distance in the diffraction grating:

$$d \times \sin \Phi = n \times \lambda$$

where:    d = row spacing

Φ = angle of diffraction

n = order of diffraction

λ = wavelength of light source

# Sample Calculation

Φ = 21°

n = 1

λ = 655 nm

d = ?
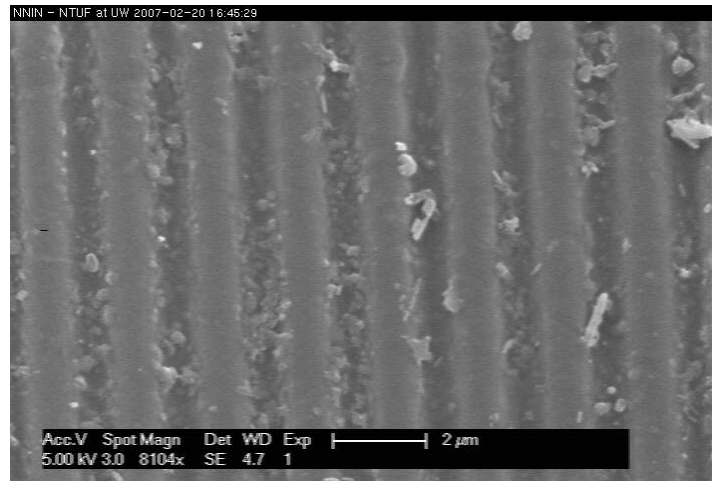


d[sin(21°)] = 1(655 nm)

d = 1827 nm

Diffraction grating spacing measured by scanning electron microscopy is approximately 1.8 μm