

## Image Scanner!

<b>Subject Area(s)</b>	Science & technology
<b>Associated Unit</b>	None
<b>Associated Lesson</b>	None
<b>Activity Title</b>	Image Scanner!
<b>Header</b>	Insert image 1 here, right justified to wrap

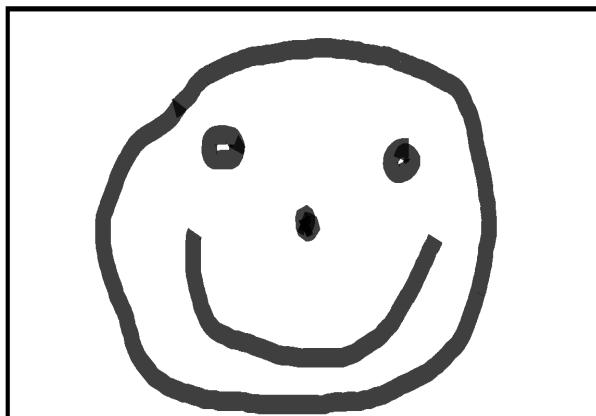
### Image 1

**ADA Description:** Students constructing an image scanner using the NXT Mindstorms

**Caption:** Students constructing an image scanner

**Image file name:** image\_scanner1.png

**Source/Rights:** Copyright 2010 Pavel Khazron.  
Used with permission.



<b>Grade Level</b>	6-8
<b>Activity Dependency</b>	None
<b>Time Required</b>	50 minutes
<b>Group Size</b>	3
<b>Expendable Cost per Group</b>	US\$0

## **Summary**

In this activity, students use the Lego Mindstorms Education kit to build a simple digital image scanner consisting of an NXT Mindstorms intelligent brick and an NXT light sensor. Students create a simple image of their choosing on supplied square ruled paper, and use the scanner to read in their images into a file. Using the Processing programming environment, students process the scanned file and display it on a computer screen.

## **Engineering Connection**

Images are all around us, and are important in the way they relate to our sense of sight. An area of engineering that studies images is called image processing. Image processing engineers study ways of capturing images on media such as film, and come up with different approaches to process them depending on the task at hand. For example, a given image that lacks color may be processed in a computer to become more vibrant. An image out of focus may be manipulated so it appears sharper. In general, image processing finds applications in art, medicine, and security, to name a few. Image processing is also used by artists, for example in making collages.

Insert Image 2 here, centered



**Image 2**

**ADA Description:** A sample digital image of peppers

**Caption:** A digital image of peppers

**Image file name:** image\_scanner2.png

**Source/Rights:** none.

## **Engineering Category**

Category 3: Engineering analysis or partial design

## **Keywords**

Image, light sensor, sense of sight, programming

## **Educational Standards**

- New York state intermediate-level science: Standard 1: T1.4b, T1.5a
- New York state technology: engineering design, information systems, computer technology

## **Pre-Requisite Knowledge**

Ability to open, modify, and run a programming sketch written in Processing.

## **Learning Objectives**

After this activity, students should be able to:

- Understand how digital images use pixels
- Understand operation of a light sensor
- Explain how images are formed by imaging instruments like digital cameras

## **Materials List**

Each group needs:

- One Lego Mindstorms NXT brick, one NXT light sensor, one medium cable, and one USB programming cable (all from kit 9797)
- One laptop or desktop computer with Lego Mindstorms NXT software v2.0, and the Processing 1.0.8 programming environment
- Printouts of supplied grid paper
- No. 2 pencils

## **Introduction/Motivation**

Our ability to see has enabled us to interact with our surroundings, engage in moment-to-moment decision making, create and maintain relationships, enjoy nature, and many other things. Obviously, our sense of sight is very important. Often, cameras are used to take pictures of a given scene. At other times, writing and illustrations such as on documents or in books are photographed or scanned to make digital images for later viewing or archival. Digital images have recently become abundant because they are simpler to make and store than images captured on film or paper, and are also easier to enhance. You may have had a chance to enhance a digital image (conduct a discussion on digital image editing - has anyone taken a picture with a digital camera that came out too dark or fuzzy, and how were they able to improve it using an image editing program supplied with the camera?).

A digital image consists of a collection of pixels. A pixel (or picture element) can be thought of as a dot with a certain color. A rectangular arrangement of pixels, with each pixel properly

colored, is basically the definition of a digital image. Our eyes are normally not able to perceive each pixel individually but only a collection of pixels. This is because each pixel is very tiny and there are very many of them in a given area. For example, an image may have 40000 pixels per square inch. The number of pixels in a given area is called image resolution. While high resolution is great for images that contain lots of detail, such as photographs of nature, not all images are of high resolution. Think of the icons on your computer, each of which is in fact a small image (typically 32 by 32 pixels). In this activity we will be working with images of size 16 by 16 pixels.

### Vocabulary / Definitions

Word	Definition
Digital image	An image that consists of pixels and can be stored digitally on a computer.
Pixel	A building block of digital images, literally a “picture element”.
Resolution	Number of pixels in a given area.
Light sensor	An electronic instrument that can be used to measure the amount of light reflected from a surface.
Sketch	A name for a computer program written in the language Processing

### Procedure

#### Before the Activity

- Make copies of the attachment **Image\_Scanner\_worksheet.pdf**.

Insert Image 3 here, centered



**Image 3**

**ADA Description:** A digital image scanner constructed using the NXT Mindstorms kit

**Caption:** A digital image scanner

**Image file name:** image\_scanner3.png

**Source/Rights:** Copyright 2010 Pavel Khazron. Used with permission.

## With the Students

1. Construct an image scanner shown in Image 3.
2. Open the NXT Mindstorms program **scanner.rbt** on your computer. Attach the image scanner to the computer using the USB programming cable. Download your program to the NXT brick using the **download** button in the lower right of the screen, as shown in Image 4.
3. Open the Processing programming environment. Processing is a kind of programming language. In Processing, a program is called a sketch. Open the sketch called **Image Scanner.pde**. This sketch should be located inside a folder called **Image\_Scanner**. Run the sketch. Observe the 16 by 16 image that appears. The image is low resolution version of the image called Trevor. Compare it to the high resolution in the image file called **trevor\_original.png**.
4. Run your **Image Scanner.pde** sketch, but this time, find and change the line **reader = createReader("trevor.txt");** to **reader = createReader("house.txt");**, and similarly for **smile.txt**. Compare your observations to the high resolution versions of these images in files **house\_original.png** and **smile\_original.png**. All the text files should be located inside the folder **Image\_Scanner**.
5. Now, create your own image! On the worksheet **Image\_Scanner\_worksheet**, create an image by penciling in appropriate squares (pixels). Be creative!
6. When you've created your image, scan it. To do this, run the program **scanner.rbt** on the NXT brick. Position the light sensor above the top left pixel, and press the orange button. Move the light sensor one pixel to the right, and press the orange button again. Continue scanning the pixels on the first line. When done, continue scanning the remaining lines from left to right. Each time you press the orange button on the NXT brick, a number reflecting the pixel's value is stored into a file called **scanner.txt**.
7. Now click on the **NXT window**, and under the **Memory** tab, verify that log file **scanner.txt** has been created (this should be under the heading **Other**). Select the file **scanner.txt** and click **Upload**. Choose the **Image\_Scanner** folder, and click **OK**. Your text file containing pixel values should now be located in the folder **Image\_Scanner**.
8. In Processing, run your sketch, but replace your **createReader** line with **reader = createReader("scanner.txt");**. If all went well, you should see the image you've created pixel by pixel on paper displayed on your computer screen!

Insert Image 4 here, centered

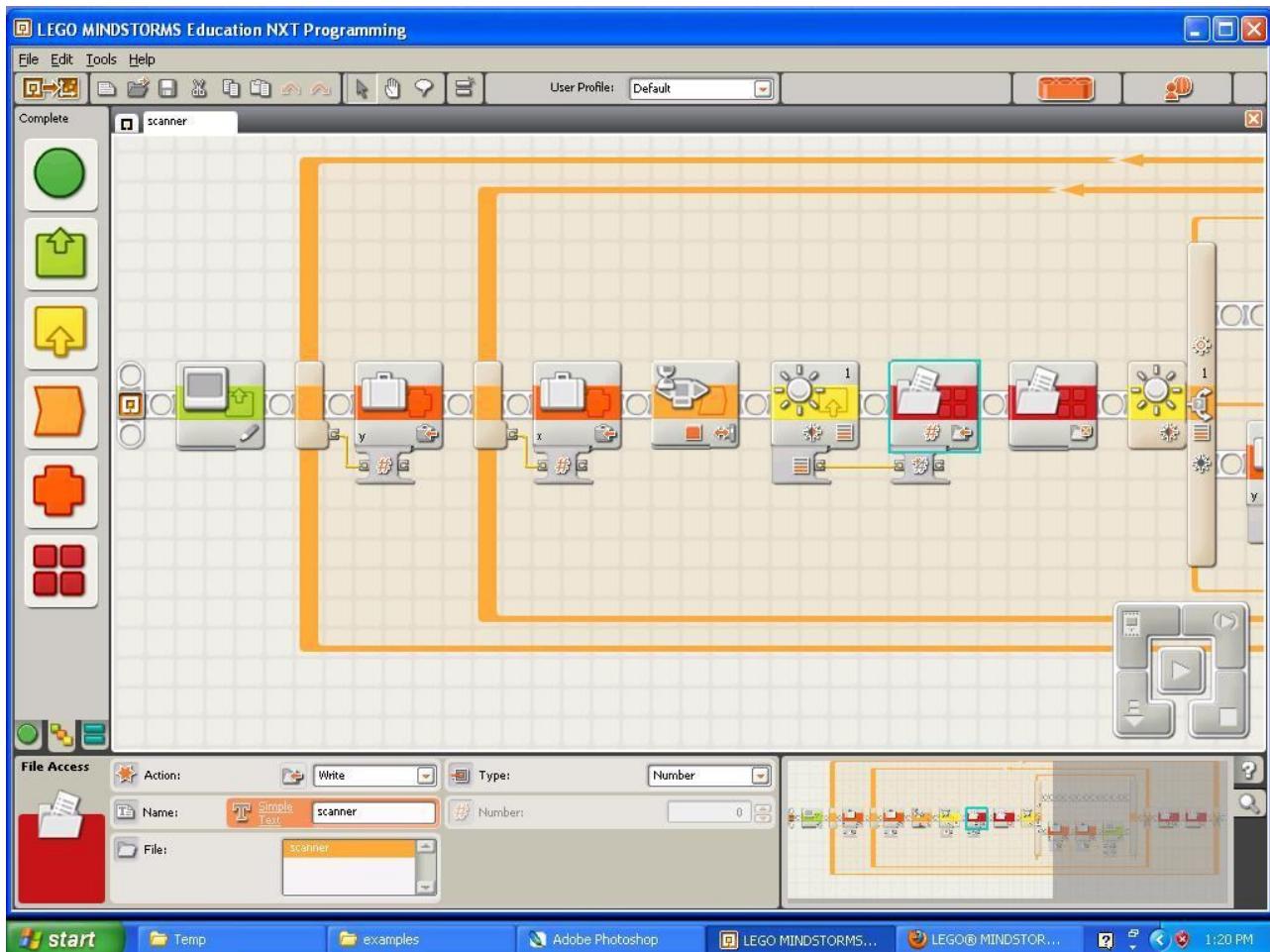


Image 4

**ADA Description:** A part of the NXT program that reads pixel values into a file

**Caption:** The first panel of the NXT program for the image scanner

**Image file name:** image\_scanner4.png

**Source/Rights:** Copyright 2010 Pavel Khazron. Used with permission.

Insert Image 5 here, centered

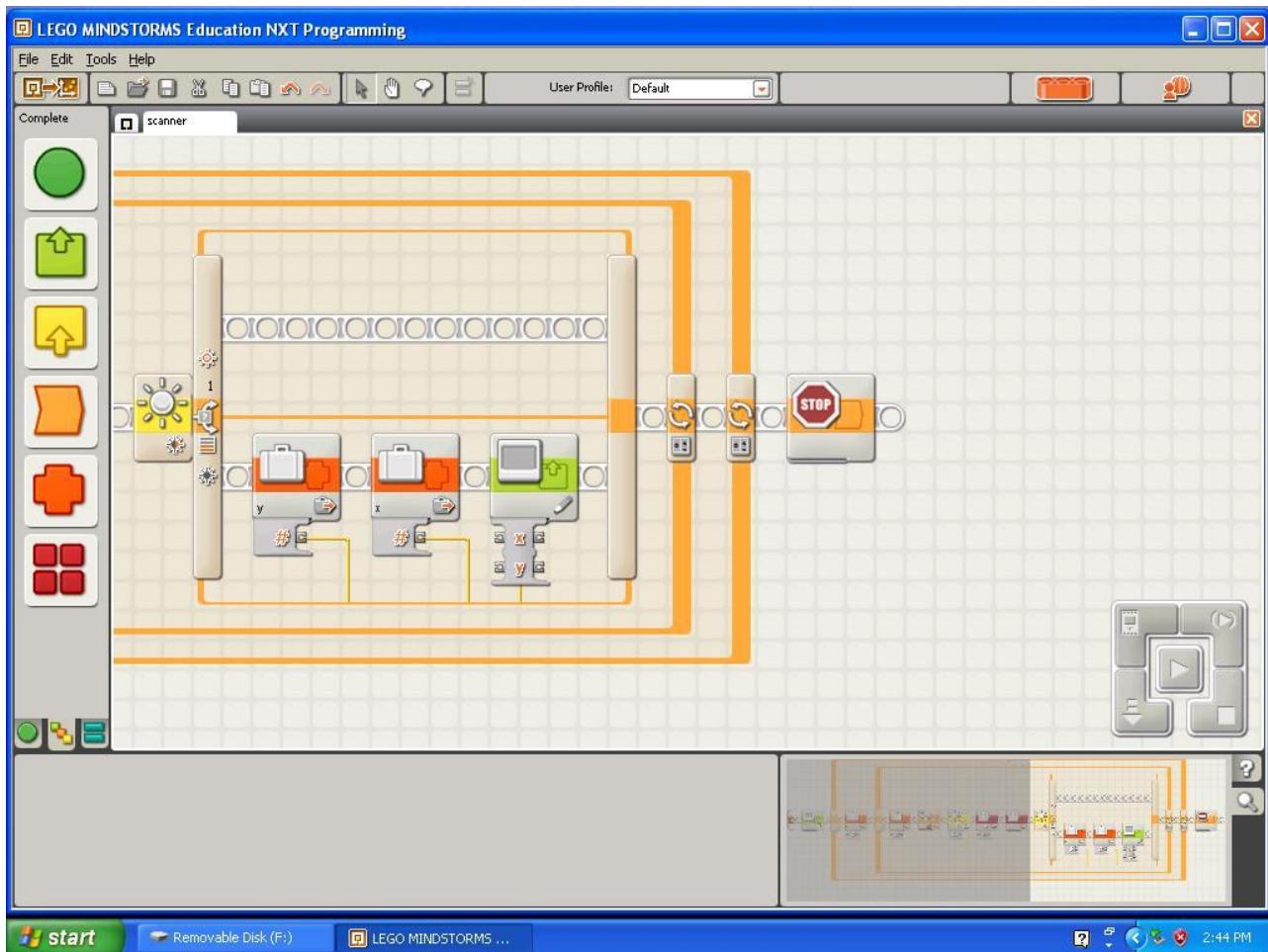


Image 5

**ADA Description:** A part of the NXT program that displays each pixel on the screen of the NXT brick

**Caption:** The second panel of the NXT program for the image scanner

**Image file name:** image\_scanner5.png

**Source/Rights:** Copyright 2010 Pavel Khazron. Used with permission.

## Attachments

Image\_Scanner\_worksheet.pdf

scanner.rbt

scanner.txt

trevor.txt

house.txt

smile.txt

trevor\_original.png

house\_original.png

smile\_original.png

## Troubleshooting Tips

Sometimes it is necessary to manually format the scanned image file **scanner.txt** in cases where characters other than numbers appear. Refer to the attached file for a sample.

## Investigating Questions

### Assessment

#### Pre-Activity Assessment

Sharing Experiences: Lead a class discussion on images in general. Photography will likely be the most popular topic. Discuss digital photography, and image editing programs that are supplied with most digital cameras or those that are readily available (see references).

#### Activity Embedded Assessment

Analysis: Encourage comparison among the different teams. Have students share the image created with other groups.

#### Post-Activity Assessment

What Have We Learned?: Conduct a discussion with the same theme as in the pre-activity assessment.

### Activity Extensions

Have an image editing project as part of the activity. Have students design a public service announcement, and advertisement, or school logo using image editing software (see references).

### Activity Scaling

None

### Additional Multimedia Support

None

### References

Processing programming environment. Accessed February 10, 2010. <http://processing.org/>.

Free image editing software. Accessed February 10, 2010. <http://www.getpaint.net/>.

A collection of royalty free images. Accessed February 10, 2010. <http://www.images.com/>.

Open source 3D content creation suite. Accessed February 10, 2010. <http://www.blender.org/>.

**Redirect URL** <http://gk12.poly.edu/amps-cbri/>

**Owner** Pavel Khazron

**Contributors** Pavel Khazron

**Copyright** Copyright © 2010 by Polytechnic Institute of NYU. The development of this activity was supported by Project AMPS under a GK-12 Fellows grant 0807286 from the National Science Foundation.

**Version:** January, 2010