K'NEX Education®

TEACHER'S GUIDE

ROBOTICS

Building System

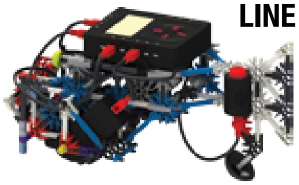
79100/79484

V01 8-2016

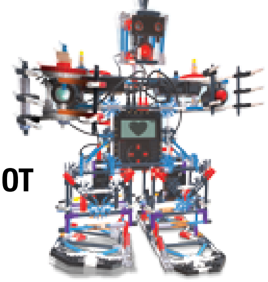
K'NEX Education®

TEACHER'S GUIDE

ROBOTICS
Building System

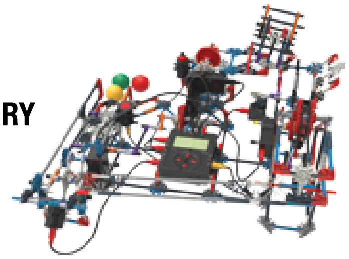


LINE TRACKER



BI-PED ROBOT

SEARCH & RESCUE ROBOT



BALL FACTORY

TABLE OF CONTENTS:

COMMON CORE STATE STANDARDS 3

NEXT GENERATION SCIENCE STANDARDS 4

INTRODUCTION 10

LESSON 1
Motor Control – Line Tracker 11

LESSON 2
Line Tracker 22

LESSON 3
Bi-Ped Robot 31

LESSON 4
Search & Rescue Robot 45

LESSON 5
Ball Factory 53

⚠ **WARNING:** CHOKING HAZARD – Small parts.
Not for children under 3 years.

A NOTE ABOUT SAFETY:

Safety is of primary concern in science and technology classrooms. It is recommended that you develop a set of rules that governs the safe, proper use of K'NEX in your classroom. Caution students to keep hands, face, hair and clothing away from all moving parts.

Alignments of the K'NEX Education Robotics Building System Set and Teacher's Guide to The Common Core State Standards – Mathematics

Mathematical practices - associated with mathematics at all grade levels

1. Make sense of problems and persevere in solving them.
2. Reason abstractly and quantitatively.
3. Construct viable arguments and critique the reasoning of others.
4. Model with mathematics.
5. Use appropriate tools strategically.
6. Attend to precision.
7. Look for and make use of structure.
8. Look for and express regularity in repeated reasoning.

GRADE 5

Operations and Algebraic Thinking

- Write and interpret numerical expressions.
- Analyze patterns and relationships.

Number and Operations in Base Ten

- Perform operations with multi-digit whole numbers and with decimals to hundredths.

Measurement and Data

- Convert like measurement units within a given measurement system.
- Represent and interpret data.

MATHEMATICS GRADE 6

In Grade 6, instructional time should focus on four critical areas:

- Connecting ratio and rate to whole number multiplication and division and using concepts of ratio and rate to solve problems.
- Writing, interpreting, and using expressions and equations.
- Developing understanding of statistical thinking.

GRADE 6

Ratios and Proportional Relationships

- Understand ratio concepts and use ratio reasoning to solve problems.

The Number System

- Compute fluently with multi-digit numbers and find common factors and multiples.

Expressions and Equations

- Apply and extend previous understandings of arithmetic to algebraic expressions.
- Reason about and solve one-variable equations.
- Represent and analyze quantitative relationships between dependent and independent variables.

Statistics and Probability

- Develop understanding of statistical variability.

MATHEMATICS GRADE 7

In Grade 7, instructional time should focus on four critical areas:

- Developing understanding of operations with rational numbers and working with expressions and linear equations.

GRADE 7

The Number System

- Apply and extend previous understandings of operations with fractions to add, subtract, multiply, and divide rational numbers.

GRADE 8

Expressions and Equations

- Work with radicals and integer exponents.
- Understand the connections between proportional.

MATHEMATICS HIGH SCHOOL

Number and Quantity

The Real Number System

- Extend the properties of exponents to rational exponents.
- Use properties of rational and irrational numbers.

Algebra

Creating Equations

- Create equations that describe numbers or relationships.

Reasoning with Equations and Inequalities

- Understand solving equations as a process of reasoning and explain the reasoning.
- Solve equations and inequalities in one variable.

Grades 3 – 5 Engineering Design

3-5-ETS1-2 Engineering Design

Students who demonstrate understanding can:

- **3-5-ETS1-2** Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem.

The performance expectation above was developed using the following elements from the NRC document A Framework for K- 12 Science Education:

Science and Engineering Practices	Disciplinary Core Ideas	Crosscutting Concepts
Constructing Explanations and Designing Solutions Constructing explanations and designing solutions in 3–5 builds on K–2 experiences and progresses to the use of evidence in constructing explanations that specify variables that describe and predict phenomena and in designing multiple solutions to design problems. • Generate and compare multiple solutions to a problem based on how well they meet the criteria and constraints of the design problem.	**ETS1.B: Developing Possible Solutions** • Research on a problem should be carried out before beginning to design a solution. Testing a solution involves investigating how well it performs under a range of likely conditions. • At whatever stage, communicating with peers about proposed solutions is an important part of the design process, and shared ideas can lead to improved designs.	**Influence of Science, Engineering, and Technology on Society and the Natural World** • Engineers improve existing technologies or develop new ones to increase their benefits, decrease known risks, and meet societal demands.

3-5-ETS1-3 Engineering Design

Students who demonstrate understanding can:

- **3-5-ETS1-3** Plan and carry out fair tests in which variables are controlled and failure points are considered to identify aspects of a model or prototype that can be improved.

The performance expectation above was developed using the following elements from the NRC document A Framework for K- 12 Science Education:

Science and Engineering Practices	Disciplinary Core Ideas	Crosscutting Concepts
Planning and Carrying Out Investigations Planning and carrying out investigations to answer questions or test solutions to problems in 3–5 builds on K–2 experiences and progresses to include investigations that control variables and provide evidence to support explanations or design solutions. • Plan and conduct an investigation collaboratively to produce data to serve as the basis for evidence, using fair tests in which variables are controlled and the number of trials considered.	**ETS1.B: Developing Possible Solutions** • Tests are often designed to identify failure points or difficulties, which suggest the elements of the design that need to be improved. **ETS1.C: Optimizing the Design Solution** • Different solutions need to be tested in order to determine which of them best solves the problem, given the criteria and the constraints.	

Grades 6 – 8

MS-PS2-2 Motion and Stability: Forces and Interactions

Students who demonstrate understanding can:

- **MS-PS2-2** Plan an investigation to provide evidence that the change in an object's motion depends on the sum of the forces on the object and the mass of the object.

The performance expectation above was developed using the following elements from the NRC document A Framework for K- 12 Science Education:

Science and Engineering Practices	Disciplinary Core Ideas	Crosscutting Concepts
Planning and Carrying Out Investigations Planning and carrying out investigations to answer questions or test solutions to problems in 6–8 builds on K–5 experiences and progresses to include investigations that use multiple variables and provide evidence to support explanations or design solutions. • Plan an investigation individually and collaboratively, and in the design: identify independent and dependent variables and controls, what tools are needed to do the gathering, how measurements will be recorded, and how many data are needed to support a claim. - *Connections to Nature of Science* **Scientific Knowledge is Based on Empirical Evidence** • Science knowledge is based upon logical and conceptual connections between evidence and explanations.	**PS2.A: Forces and Motion** • The motion of an object is determined by the sum of the forces acting on it; if the total force on the object is not zero, its motion will change. The greater the mass of the object, the greater the force needed to achieve the same change in motion. For any given object, a larger force causes a larger change in motion. • All positions of objects and the directions of forces and motions must be described in an arbitrarily chosen reference frame and arbitrarily chosen units of size. In order to share information with other people, these choices must also be shared.	**Stability and Change** • Explanations of stability and change in natural or designed systems can be constructed by examining the changes over time and forces at different scales.

MS-PS2-3 Motion and Stability: Forces and Interactions

Students who demonstrate understanding can:

- **MS-PS2-3** Ask questions about data to determine the factors that affect the strength of electric and magnetic forces.

The performance expectation above was developed using the following elements from the NRC document A Framework for K- 12 Science Education:

Science and Engineering Practices	Disciplinary Core Ideas	Crosscutting Concepts
Asking Questions and Defining Problems Asking questions and defining problems in grades 6–8 builds from grades K–5 experiences and progresses to specifying relationships between variables, and clarifying arguments and models. • Ask questions that can be investigated within the scope of the classroom, outdoor environment, and museums and other public facilities with available resources and, when appropriate, frame a hypothesis based on observations and scientific principles.	**PS2.B: Types of Interactions** • Electric and magnetic (electromagnetic) forces can be attractive or repulsive, and their sizes depend on the magnitudes of the charges, currents, or magnetic strengths involved and on the distances between the interacting objects.	**Cause and Effect** • Cause and effect relationships may be used to predict phenomena in natural or designed systems.

MS-PS3-5 Energy

Students who demonstrate understanding can:

- **MS-PS3-5** Construct, use, and present arguments to support the claim that when the kinetic energy of an object changes, energy is transferred to or from the object.

The performance expectation above was developed using the following elements from the NRC document A Framework for K- 12 Science Education:

Science and Engineering Practices	Disciplinary Core Ideas	Crosscutting Concepts
Engaging in Argument from Evidence Engaging in argument from evidence in 6–8 builds on K–5 experiences and progresses to constructing a convincing argument that supports or refutes claims for either explanations or solutions about the natural and designed worlds. • Construct, use, and present oral and written arguments supported by empirical evidence and scientific reasoning to support or refute an explanation or a model for a phenomenon. - ***Connections to Nature of Science*** **Scientific Knowledge is Based on Empirical Evidence** • Science knowledge is based upon logical and conceptual connections between evidence and explanations.	**PS3.B: Conservation of Energy and Energy Transfer** • When the motion energy of an object changes, there is inevitably some other change in energy at the same time.	**Energy and Matter** • Energy may take different forms (e.g. energy in fields, thermal energy, energy of motion).

MS-PS4-3 Waves and Their Applications in Technologies for Information Transfer

Students who demonstrate understanding can:

- **MS-PS4-3** Integrate qualitative scientific and technical information to support the claim that digitized signals are a more reliable way to encode and transmit information than analog signals.

The performance expectation above was developed using the following elements from the NRC document A Framework for K- 12 Science Education:

Science and Engineering Practices	Disciplinary Core Ideas	Crosscutting Concepts
Obtaining, Evaluating, and Communicating Information Obtaining, evaluating, and communicating information in 6-8 builds on K-5 and progresses to evaluating the merit and validity of ideas and methods. • Integrate qualitative scientific and technical information in written text with that contained in media and visual displays to clarify claims and findings.	**PS4.C: Information Technologies and Instrumentation** • Digitized signals (sent as wave pulses) are a more reliable way to encode and transmit information.	**Structure and Function** • Structures can be designed to serve particular functions. - ***Connections to Engineering, Technology, and Applications of Science*** **Influence of Science, Engineering, and Technology on Society and the Natural World** • Technologies extend the measurement, exploration, modeling, and computational capacity of scientific investigations.

MS-ETS1-2 Engineering Design

Students who demonstrate understanding can:

- **MS-ETS1-2** Evaluate competing design solutions using a systematic process to determine how well they meet the criteria and constraints of the problem.

The performance expectation above was developed using the following elements from the NRC document A Framework for K- 12 Science Education:

Science and Engineering Practices	Disciplinary Core Ideas	Crosscutting Concepts
Engaging in Argument from Evidence Engaging in argument from evidence in 6–8 builds on K–5 experiences and progresses to constructing a convincing argument that supports or refutes claims for either explanations or solutions about the natural and designed world. • Evaluate competing design solutions based on jointly developed and agreed-upon design criteria.	**ETS1.B: Developing Possible Solutions** • There are systematic processes for evaluating solutions with respect to how well they meet the criteria and constraints of a problem.	

MS-ETS1-3 Engineering Design

Students who demonstrate understanding can:

- **MS-ETS1-3** Analyze data from tests to determine similarities and differences among several design solutions to identify the best characteristics of each that can be combined into a new solution to better meet the criteria for success.

The performance expectation above was developed using the following elements from the NRC document A Framework for K- 12 Science Education:

Science and Engineering Practices	Disciplinary Core Ideas	Crosscutting Concepts
Analyzing and Interpreting Data Analyzing data in 6–8 builds on K–5 experiences and progresses to extending quantitative analysis to investigations, distinguishing between correlation and causation, and basic statistical techniques of data and error analysis. • Analyze and interpret data to determine similarities and differences in findings.	**ETS1.B: Developing Possible Solutions** • There are systematic processes for evaluating solutions with respect to how well they meet the criteria and constraints of a problem. • Sometimes parts of different solutions can be combined to create a solution that is better than any of its predecessors. **ETS1.C: Optimizing the Design Solution** • Although one design may not perform the best across all tests, identifying the characteristics of the design that performed the best in each test can provide useful information for the redesign process—that is, some of those characteristics may be incorporated into the new design.	

MS-ETS1-4 Engineering Design

Students who demonstrate understanding can:

- **MS-ETS1-4** Develop a model to generate data for iterative testing and modification of a proposed object, tool, or process such that an optimal design can be achieved.

The performance expectation above was developed using the following elements from the NRC document A Framework for K- 12 Science Education:

Science and Engineering Practices	Disciplinary Core Ideas	Crosscutting Concepts
Developing and Using Models Modeling in 6–8 builds on K–5 experiences and progresses to developing, using, and revising models to describe, test, and predict more abstract phenomena and design systems. • Develop a model to generate data to test ideas about designed systems, including those representing inputs and outputs.	**ETS1.B: Developing Possible Solutions** • A solution needs to be tested, and then modified on the basis of the test results, in order to improve it. • Models of all kinds are important for testing solutions. **ETS1.C: Optimizing the Design Solution** • The iterative process of testing the most promising solutions and modifying what is proposed on the basis of the test results leads to greater refinement and ultimately to an optimal solution.	

Grade 9

HS-PS2-1 Motion and Stability: Forces and Interactions

Students who demonstrate understanding can:

- **MS-PS2-1** Analyze data to support the claim that Newton's second law of motion describes the mathematical relationship among the net force on a macroscopic object, its mass, and its acceleration.

The performance expectation above was developed using the following elements from the NRC document A Framework for K- 12 Science Education:

Science and Engineering Practices	Disciplinary Core Ideas	Crosscutting Concepts
Analyzing and Interpreting Data Analyzing data in 9–12 builds on K–8 and progresses to introducing more detailed statistical analysis, the comparison of data sets for consistency, and the use of models to generate and analyze data. • Analyze data using tools, technologies, and/or models (e.g., computational, mathematical) in order to make valid and reliable scientific claims or determine an optimal design solution.	**PS2.A: Forces and Motion** • Newton's second law accurately predicts changes in the motion of macroscopic objects.	**Cause and Effect** • Empirical evidence is required to differentiate between cause and correlation and make claims about specific causes and effects.

HS-PS3-3 Energy

Students who demonstrate understanding can:

- **HS-PS3-3** Design, build, and refine a device that works within given constraints to convert one form of energy into another form of energy.

The performance expectation above was developed using the following elements from the NRC document A Framework for K- 12 Science Education:

Science and Engineering Practices	Disciplinary Core Ideas	Crosscutting Concepts
Constructing Explanations and Designing Solutions Constructing explanations and designing solutions in 9–12 builds on K–8 experiences and progresses to explanations and designs that are supported by multiple and independent student-generated sources of evidence consistent with scientific ideas, principles, and theories. • Design, evaluate, and/or refine a solution to a complex real-world problem, based on scientific knowledge, student-generated sources of evidence, prioritized criteria, and tradeoff considerations.	**PS3.A: Definitions of Energy** • At the macroscopic scale, energy manifests itself in multiple ways, such as in motion, sound, light, and thermal energy. **PS3.D: Energy in Chemical Processes** • Although energy cannot be destroyed, it can be converted to less useful forms—for example, to thermal energy in the surrounding environment.	**Energy and Matter** • Changes of energy and matter in a system can be described in terms of energy and matter flows into, out of, and within that system.

Introduction to K'NEX ROBOTICS

Welcome to the wonderful world of robotics with the K'NEX Robotics Set designed to meet the needs of students in a high interest STEM program. Most will agree that robotics is the first thing that comes to mind when discussions turn to STEM. This set will provide you or you and your students with exciting, large, fully functioning K'NEX models to build. In addition, the set includes easy to use software and custom designed electronics that will enable students to explore programming from the simple to the complex. As countries around the world seek to provide computer education and coding opportunities for all students, K'NEX has developed this comprehensive robotics set to help you and your students to meet this challenging goal.

A comprehensive Teacher's Guide, a Hardware User's Guide, and a Software User's Guide are provided to enhance your students' understanding of programming logic, structure and design. The software has been designed around a "Click and Drop" system using programmable output, process, and decision symbols. As the symbols are sequenced, they produce flowcharts to control any of the nine models in the K'NEX Robotics Set. Additionally, the flowcharts can be viewed in text form with a click of the mouse. The set includes: 4 motors, an LED, Distance Sensor, Color/Light Sensor, Push Button Sensor, cables/wiring, 825 K'NEX pieces, Hardware User's Guide, Software User's Guide, and the K'NEX Robotics Control Box *(with battery power option)*.

The Teacher's Guide:

The Teacher's Guide includes five lessons which use four of the nine models that can be constructed from the set. The lessons progress from simple to complex programming as they introduce students to: the model's mechanical systems, motor operation, sensor operation and input, programming structure and design, and feedback systems. Initial lessons introduce basic programming strategies and include challenge activities to pique students' interest and desire to know and understand more! Later lessons enable students to explore the programming of machines with multiple motors, multiple sensor inputs, and output devices. These later lessons will encourage complex computational thinking and enable students to reach higher levels of programming excellence. Once students have completed the 5 lessons in the Teacher's Guide, they can move on to the other models in the K'NEX Robotics Set and let their imaginations guide their explorations as they expand their programming design skills in conjunction with the valuable information found in the K'NEX Hardware and Software User's Guides. Each lesson is aligned to standards (NGSS, ITEEA, NSES, NCTM, and Common Core).

Each lesson provides information for the teacher, instructional suggestions, objectives, materials list, and appropriate programming examples. Each lesson also provides Student Instruction Sheets and Student Response Sheets. The Student Instruction Sheets outline the lesson for the students so they can tackle the particular project or challenge on their own or with a minimum of supervision. The initial lessons will require more support for the students if they are new to programming. The models highlighted in the lessons are the Line Tracker *(used in two lessons)*, Bi-Ped Robot, Search & Rescue Robot, and the Ball Factory. The Student Response Sheets provide charts and questions for the students to complete as they work through each lesson. These sheets enable students to keep a record of their work and they are excellent for the collection of assessment data.

Now it is time for you and your students to begin a very excellent adventure in robotics.

LESSON 1:

Motor Control – Line Tracker

Robotics technology plays a very important role in almost every aspect of today's life from small toy vehicles to drivable toys, electric cars and even to rockets that carry people and satellites into space. The K'NEX Education Robotics Set will introduce students to their first K'NEX hands-on robotics experience. Students will build model robotic systems that they program to move and respond to information gathered from the environment. This first lesson will introduce students to the concepts related to motors and their potential when used in robotic systems. This is a project-based learning opportunity that allows students working in collaborative groups to explore, investigate, and experiment. Students will follow instructions provided in the K'NEX Education Robotics Instruction Booklet to build the Line Tracker model that they will use during this lesson.

Objectives

Students will be able to:

- Work in collaborative teams.
- Build models from instructions in the Robotics Instruction Booklet.
- Design and write programs using flowcharts to control their model.
- Design techniques and/or experiments to test the performance of their model.
- Experience real world applications of STEM concepts.

Materials

Each group will need:

- Built Line Tracker model *(page 64 in the instructions book)*
- Metric tape and/or meter stick
- Graph paper
- Open area in the classroom or hallway for the model to operate
- K'NEX Robotics User's Guide
- PC and K'NEX Robotics Programming Software
- Student Instruction Sheet A - Lesson 1
- Copies of Student Response Sheets 1 & 2 - Lesson 1

Teacher's Notes

- Allow time for students to explore the K'NEX pieces *(connectors, rods, wheels, etc.)* before they begin building the model *(students may need assistance building the Line Tracker model).*
- Allow time for students to explore the K'NEX Robotics Building System User's Guide before they begin using the K'NEX Robotics Programming Software.
- It is recommended to build the Line Tracker model first, with all motors and sensors connected to the Control Box, before trying to write programs.
- When the Control Box is connected (via USB cable) to the K'NEX Robotics Programming Software, the software should recognize all the sensors, motors, and the ports where they are connected into the Control Box.
- Before programming, have the students click on each motor icon in the software to operate the motors in a "simulation" mode. This will help them discover which motor icon is the "left" motor and which one is the "right" motor.

Introductory Robotics Lesson

This lesson is designed for educators teaching an introductory robotics class. Students will measure the distance the Line Tracker travels when they change the speed of the motor using the K'NEX Robotics Programming Software.

Lesson 1 includes:

- Line Tracker Building Instructions – step-by-step plans from the Robotics Instruction Booklet.
- Student Instruction Sheet A to allow students to write their first program which will be used as they collect, present, and analyze data.
- Student Response Sheet 1. Students will keep records and results of their experiments on this page.
- Optional challenge activities provide additional programming opportunities to strengthen student understanding and for student groups who finish early. Student Response Sheet 2 is provided for students to keep records of their programming strategies.
- Clean up time.
- Optional end-of-class discussion activity *(at teacher's discretion)*.

Exploring the K'NEX Robotics Programming Software and Building the Line Tracker Model

Pre-Lesson:

- Identify student groups of 2 - 3.
- Build the Line Tracker Model from the Robotics Instruction Booklet provided.
- Provide groups a computer with K'NEX Robotics Programming Software and K'NEX Robotics Building System User's Guide. The K'NEX Robotics Programming Software and the K'NEX Robotics Building System User's Guide can be downloaded on as any computers as needed for instruction.
- Instruct students to explore the User's Guide to familiarize themselves with the K'NEX Robotics Programming Software. *NOTE: Based on your student's experience level, you may need to provide some level of direct instruction to familiarize them with K'NEX Robotics Programming Software.*
- Specific motor and sensor port connections must be known to correctly program the model. After connecting the USB cable to the Control Box and clicking the "Connect" button in the top menu bar, you should see the icons pictured in Figure 1 in the I/O panel.

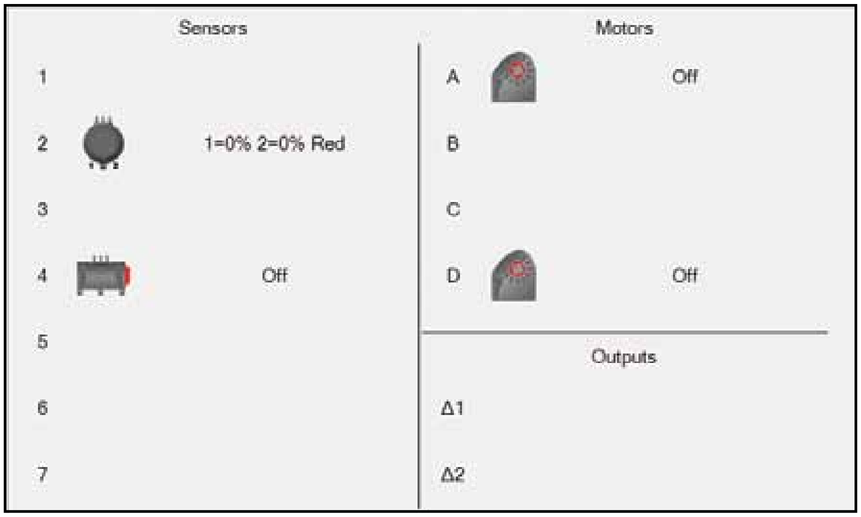


Figure 1: *Light Sensor: Port 2, Push Button Switch: Port 4, Right Motor: Port D, Left Motor: Port A*

- The motors in the Line Tracker model are positioned so that when the motors are programmed to run forward, the model will also move forward.
- Encourage students to design sample programs that include motor commands.

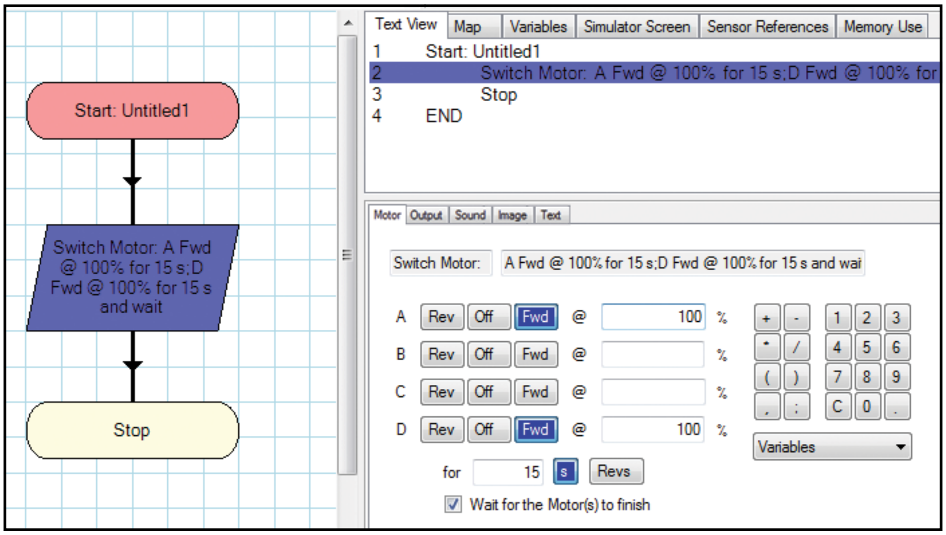
Lesson Process:

- Provide students with copies of Student Instruction Sheet A. These instructions walk students though the process of writing and running their first program.
- Review the instructions to ensure that the students understand the information that is provided to make their programming experience more successful.
- Distribute Student Response Sheet 1 and review the directions along with the charting and graphing requirements of the activity. Also, introduce Student Response Sheet 2 if the students will continue on to complete Challenge Activities.
- Student Instruction Sheet A directs students as they write their first flowchart program to operate their model.
- The figures and text below provide you with solutions to the programs that the students are required to design as they complete Student Response Sheets 1 & 2.
- Provide metric tape/meter sticks and graph paper for students.

Sample solutions to the programs students will need to complete as they work on both Student Response Sheets

Students will use their first program *(outlined on Student Instruction Sheet A)* as a template to develop the second, third, and fourth programs required to complete Student Response Sheet 1.

The Flowchart sample solution for the students' second program



```
Start: Untitled1
        Switch Motor: A Fwd @ 100% for 15 s;D Fwd @ 100% for
        Stop
END
```

Switch Motor: A Fwd @ 100% for 15 s;D Fwd @ 100% for 15 s and wai

A Fwd @ 100 %

B @ %

C @ %

D Fwd @ 100 %

for 15 s

☑ Wait for the Motor(s) to finish

Figure 2: *Line Tracker driving forward for 15 seconds with 50% power.*

The Flowchart sample solutions for the students' third and fourth programs

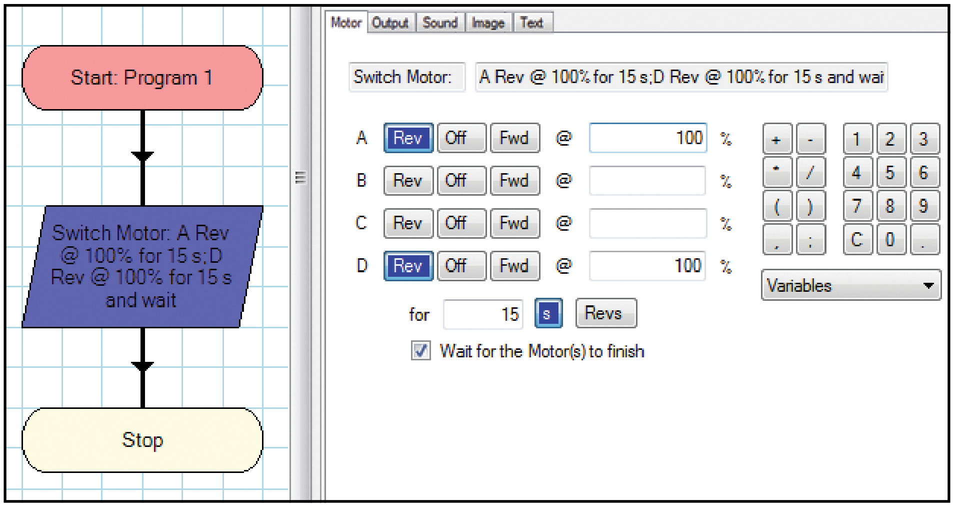
The flowcharts for the third and fourth programs will be identical to the flowchart in Figure 2 except the percentages will be 25% and 75% respectively.

Solutions to Student Response Sheet 2 Challenges

STUDENT CHALLENGE: Program the model to move backwards

Teacher's Note: Distribute Student Response Sheet 2 and remind students that they are to brainstorm and draw the flowchart for their proposed programs before using the computer. Ask them to explain their logic orally as you move about the classroom.

The first challenge is to write and test a flowchart for the Line Tracker to move backwards. Once the flowchart is initiated, the students will set motor A to 'Rev' *(reverse)* and select a power level. They will also set motor D to 'Rev' *(reverse)* and select the same power level they chose for motor A. Students then must set the time duration, in seconds, for the motors *(Figure 3)*.



Start: Program 1

Switch Motor: A Rev @ 100% for 15 s;D Rev @ 100% for 15 s and wait

Stop

Switch Motor: A Rev @ 100% for 15 s;D Rev @ 100% for 15 s and wai

A Rev Off Fwd @ 100 %

B Rev Off Fwd @ %

C Rev Off Fwd @ %

D Rev Off Fwd @ 100 %

for 15 s Revs

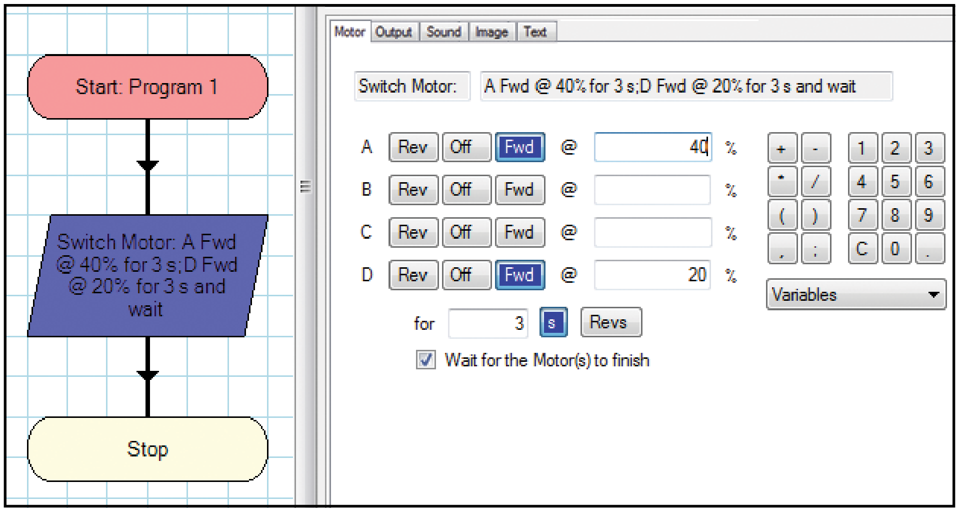
☑ Wait for the Motor(s) to finish

Figure 3: *Line Tracker moving backwards for 15 seconds with 100% power.*

STUDENT CHALLENGE: Turn the Line Tracker to the right

Teacher's Note: Have the students explore different motor controls to make a 90° right turn.

- **Arc Turn Right –** Both motors turn the same direction, but the right motor is powered less than the left. This arrangement will cause the model to turn in an arc to the right. Challenge students to explore what happens to the size of the turn when larger or smaller power differences are used. *NOTE: Adjustments to the duration (in seconds) need to be made to keep the right turn approximately 90°, and not have the model turn 180° or spin in a circle. Figure 4 shows an example of the Arc Turn Right.*



Switch Motor: A Fwd @ 40% for 3 s;D Fwd @ 20% for 3 s and wait

A Fwd @ 40 %

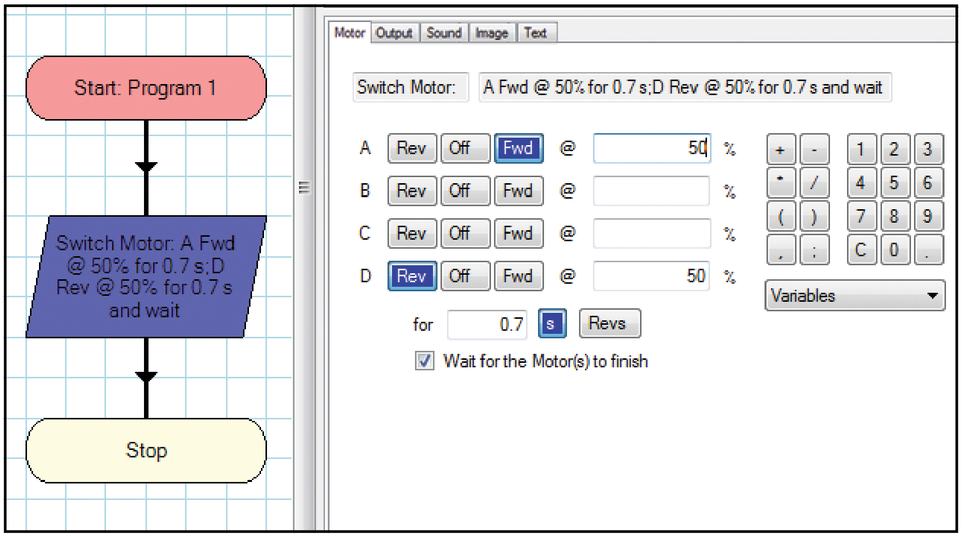
D Fwd @ 20 %

for 3 s

☑ Wait for the Motor(s) to finish

Figure 4: *Arc Turn Right.*

- **Pivot Turn Right –** Right motor spins in the opposite direction to the left motor causing the model to pivot in place as it turns to the right *(See Figure 5)*.



Switch Motor: A Fwd @ 50% for 0.7 s;D Rev @ 50% for 0.7 s and wait

A Fwd @ 50 %

D Rev @ 50 %

for 0.7 s

☑ Wait for the Motor(s) to finish

Figure 5: *Pivot Turn Right.*

STUDENT CHALLENGE: Turn the Line Tracker to the left

Teacher's Note: To program the model to turn left, students will use information similar to the data in Figures 4 & 5 except they will reverse the motor directions and power settings for motors A and D.

LESSON 1 – Student Instruction Sheet A

Programming and Investigating the Motion of the Line Tracker

- Gather your model, USB cable, computer, a meter stick or metric tape, and a sheet of graph paper.
- After receiving your teacher's approval, start the programming process to meet the requirements of the activity on Student Response Sheet 1.
- You will be using two motors, and the K'NEX Robotics Control Box to operate the Line Tracker.

Introduction to Motors

Motors are used in almost all robotic projects. Anything that moves has a motor of some form. The two motors used will control the motion of the Line Tracker. The motors need power that is provided through the K'NEX Robotics Control Box. The motor ports (A, B, C, D) are on the top side of the Control Box. Letter names for each motor port are on the front face above the LCD screen – see Figure 6.

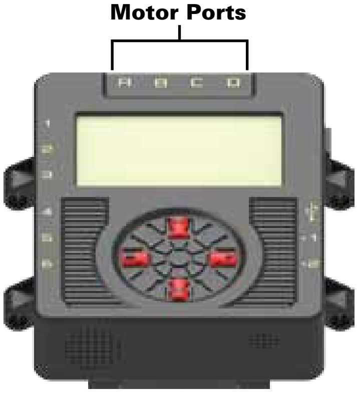
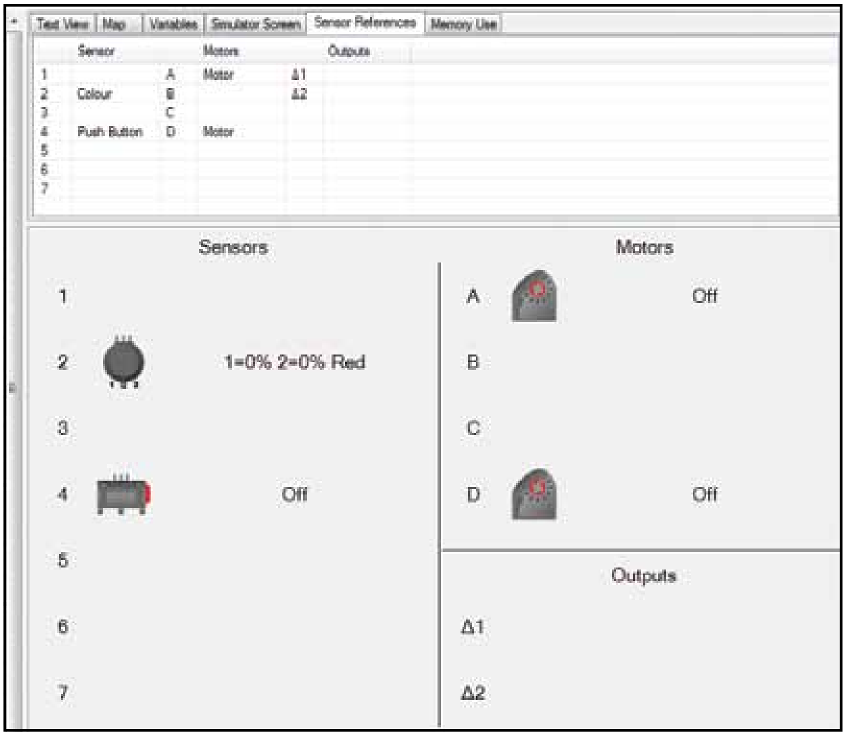


Figure 6: *K'NEX Robotics Control Box - Front Face*

Connecting the Control Box to the Programming Software

After the Line Tracker is built, ensure that the Control Box and the two motors are attached and inserted in the model as per the Line Tracker instructions. Launch the K'NEX Robotics Programming Software and select "USB connection". Turn the power switch on the Control Box to "ON." Attach the USB cable between the Control Box and your computer. Find and click on the 'Connect' icon in the control toolbar at the top of the screen. Specific motor and sensor port locations must be known to correctly program the model. When the software is connected to the Control Box of an assembled model, it will automatically populate the motor and sensor ports that are connected. After connecting the USB cable to the Control Box and clicking the "Connect" button on the top menu bar, you should see the icons pictured in Figure 7. Note that the Color/Light Sensor and Push Button Switch are on the model and visible in the software screen, but will not be used until Lesson 2. Refer to the troubleshooting guide if you are having difficulty making a connection between the box and your computer.



	Sensor		Motors	Outputs
1		A	Motor	Δ1
2	Colour	B		Δ2
3		C		
4	Push Button	D	Motor	
5				
6				
7				

Sensors

1

2 1=0% 2=0% Red

3

4 Off

5

6

7

Motors

A Off

B

C

D Off

Outputs

Δ1

Δ2

Figure 7: *Software screen with correct control box connections.*

PROGRAMMING

First Program: Moving the Line Tracker Forward

To write your first flowchart program, select the required icons one at a time from the symbol toolbar on the left of the screen and place them in the edit area. Select and place an "Output" and a "Stop" icon as shown in Figure 8. The start icon is provided by default on every new workspace for programming. You may choose to name your program by clicking on the Start Icon and typing a name for the program, i.e., "Forward", in the space that opens at the bottom right of the screen. The name will appear as "Start: Forward" in the start oval once you type the name and press enter on the keyboard or click the "OK" button on the screen.

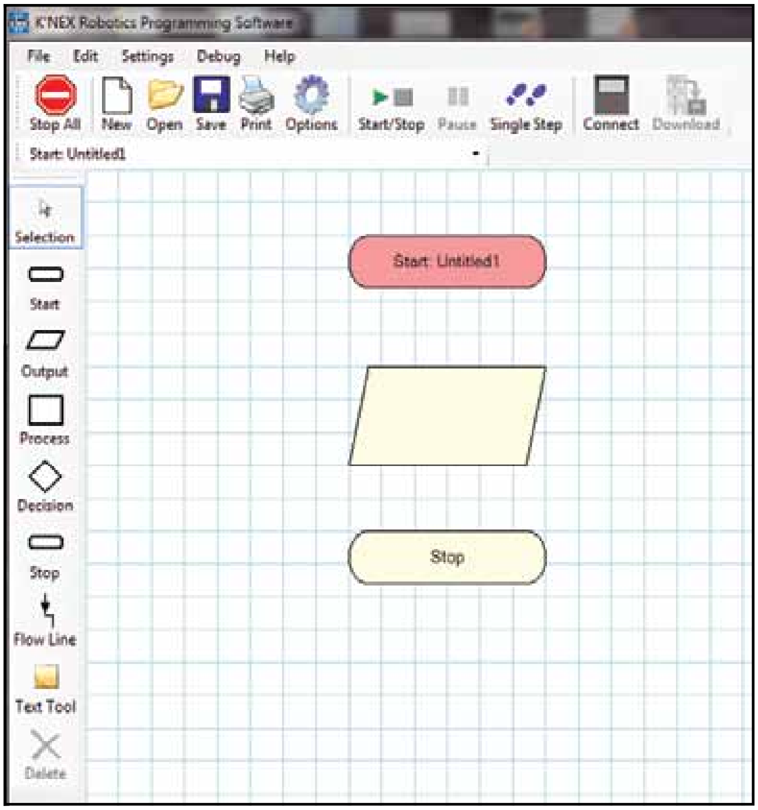


Figure 8: Added "Output" and "Stop" program icons.

Now, connect the blocks using the flow line icon. First, select the "Flow Line" icon from the symbol toolbar. Second, click on the "Start" oval in the edit area, and then click on the "Output" symbol just below it. The "Start" and "Output" program items should now be connected with a flow line. Repeat the process to connect the "Output" with "Stop." If your flowchart looks like Figure 9, you are ready for the next step.

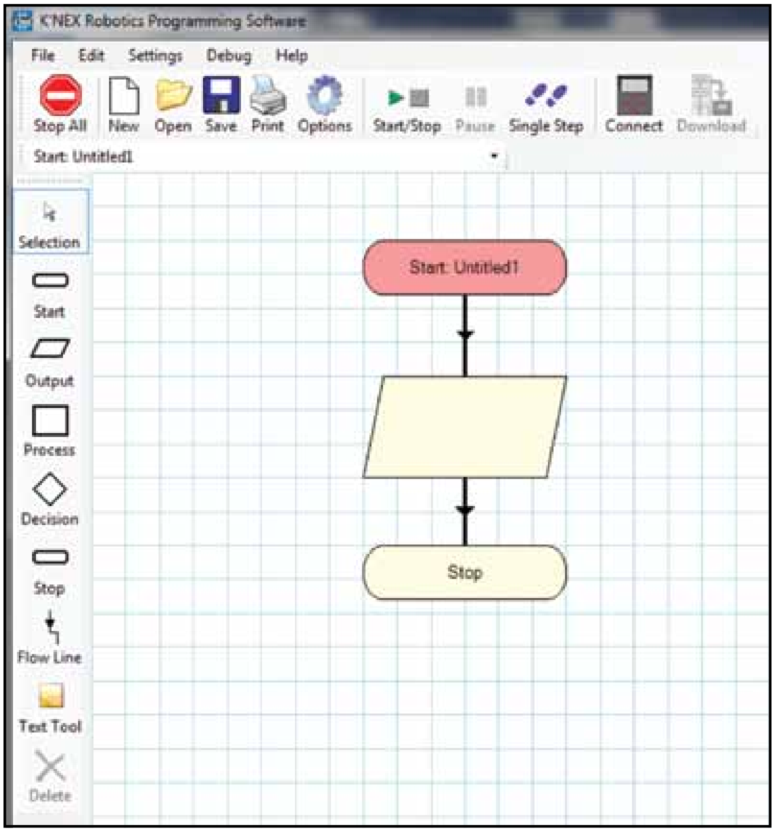
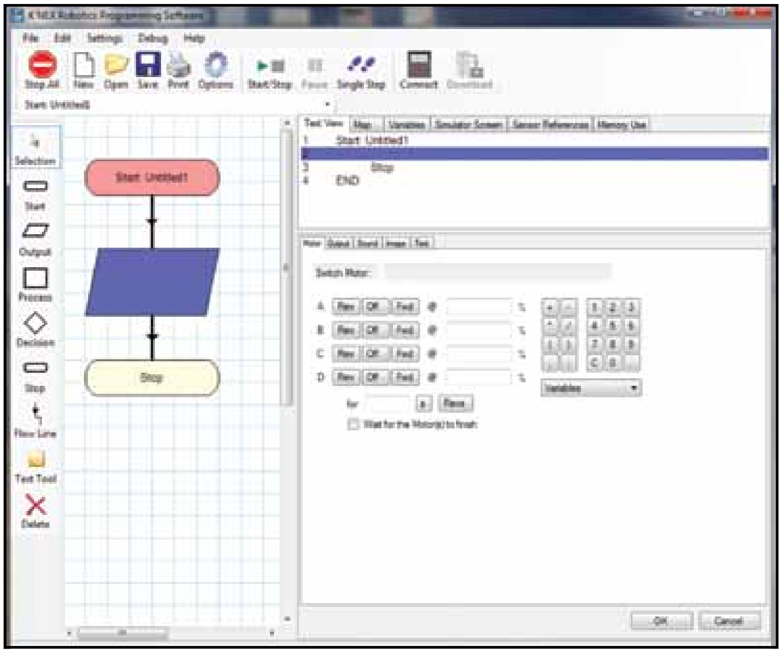


Figure 9: Added flow lines to program.

Click the selection tool from the symbol toolbar and select the "Output" symbol *(parallelogram shaped box)* in your program. The shape will highlight and the lower right window *(Input/ Output panel)* will change. You will see six tabs in this new window. Click on the "motor" tab and motor programming options will appear, see Figure 10.

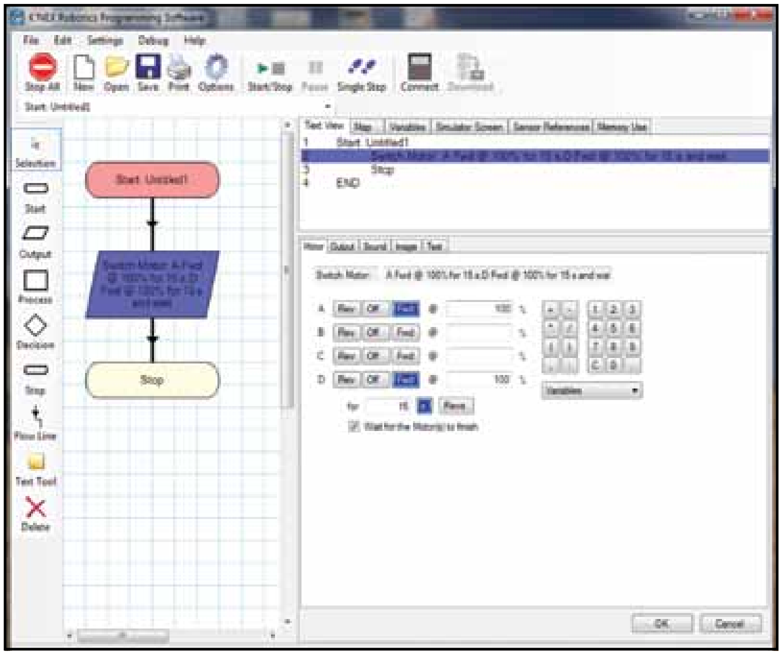

K'NEX Robotics Programming Software
File Edit Settings Debug Help
Stop All New Open Save Print Options Start/Stop Pause Single Step Connect Download
Start: Untitled1
Selection Start Output Process Decision Stop Flow Line Text Tool Delete
Start Untitled1
Stop
Text View Map Variables Simulator Screen Sensor References Memory Use
Motor Output Sound Image Text
Switch Motor:
A Rev Off Fwd @ %
B Rev Off Fwd @ %
C Rev Off Fwd @ %
D Rev Off Fwd @ %
for s Pause
Variables
OK Cancel

```
1    Start  Untitled1
2
3           Stop
4    END
```

☐ Wait for the Motor(s) to finish

Figure 10: *Switch Motor Program Option*

To program the motors correctly you need to know the position of the motors in the Line Tracker model and the ports the motors are using on the Control Box. For this model the right side motor is plugged into port D and the left side motor is plugged into port A. You will need to program the motors to run forward in your first program. To answer the question, "How far does the Line Tracker travel when you program the model to move forward at 100% motor power for 15 seconds," you will need to enter motor parameters as shown in Figure 11. Be sure to select the "wait for motor to finish" box so that the motor runs for 15 seconds. Otherwise the motor will only run for a fraction of a second before the program comes to a stop. You also need to click "OK" to save your entered motor parimeters.


K'NEX Robotics Programming Software
File Edit Settings Debug Help
Stop All New Open Save Print Options Start/Stop Pause Single Step Connect Download
Start: Untitled1
Selection Start Output Process Decision Stop Flow Line Text Tool Delete
Start Untitled1
Switch Motor A Fwd @ 100% for 15 s,D Fwd @ 100% for 15 s and wait
Stop
Text View Map Variables Simulator Screen Sensor References Memory Use
Motor Output Sound Image Text
Rev Off Fwd
Pause
Variables
OK Cancel

```
1    Start  Untitled1
2           Switch Motor: A Fwd @ 100% for 15 s,D Fwd @ 100% for 15 s and wait
3           Stop
4    END
```

Switch Motor: A Fwd @ 100% for 15 s,D Fwd @ 100% for 15 s and wait

A Fwd @ 100 %

D Fwd @ 100 %

for 15 s

☑ Wait for the Motor(s) to finish

Figure 11: *Line Tracker Program – moving forward for 15 seconds with 100% power*

If your program looks like Figure 11, then you are ready to download it to the K'NEX Robotics Control Box. Click the "Download" icon on the control toolbar near the top of the screen. In the pop-up window, select a location for your program by clicking one of the numbered rows *(the selected space will turn blue)*. Click "Download Program" and a "Control Program Name" pop-up window will appear. Enter the name of your program in the space provided and click the "OK" button. Your program has been downloaded to the Control Box. The program name will be shown in parentheses after the file name at the top of the program window. Click the red "X" on the pop-up window to exit. You can now run the program from the computer using the "Start/Stop" button from the control toolbar. To run the program remotely from the Control Box, click the "Disconnect" button from the control toolbar and unplug the USB cable. Select the program you downloaded from the menu screens on the Control Box, set the model at the starting point and run the program to see how far the model travels in 15 seconds. At this point you have a program to help you complete Student Response Sheet 1.

LESSON 1

Student Response Sheet 1

Name ____ **Group #** ________

How far does the Line Tracker travel when you program the model to move forward at 100% motor power for 15 seconds? How far does the model move forward at 50% motor power for 15 seconds?

Distance traveled at 100% motor power in 15 seconds: ________________

Distance traveled at 50% motor power in 15 seconds: ________________

Briefly describe what this information tells you about the performance of the Line Tracker model.

Insert the results from the previous activity into the chart below and write additional programs to collect enough data to complete the chart.

Motor Power	Distance Travelled	Comments
25%		
50%		
75%		
100%		

Identify the independent variable shown on your data chart.

Identify the dependent variable.

Graph your results. *(Attach your graph to this page)*

Analyze your graph and explain what it shows about the relationship between motor power and distance traveled by the model.

LESSON 1

Student Response Sheet 2

Name ____ **Group #** __________

Diagram a flowchart below, that will allow the Line Tracker to move backwards for 15 seconds.

Program the model to move backwards and test its operation.

Did the Line Tracker move as you expected? Explain!

LESSON 1

Student Response Sheet 2

(continued)

Name ____ **Group #** ________

Diagram a flowchart below, that turns the Line Tracker to the right.

Diagram a second flowchart to turn left.

Program the Line Tracker to complete the turns and test your programs.

Did the Line Tracker turn in each case as you expected? Explain!

LESSON 2: Line Tracker

The Line Tracker will be the first interactive robot students build that requires motors and a sensor to work together. In the previous lesson *(Motor Control – Line Tracker)* we introduced the concept of the motor, its performance and motion *(forward and reverse)*. In Lesson 2, we will add another feature where the Line Tracker moves using motors and has a sensor that acts like an eye to check where it is going. The Line Tracker will have a feedback sensory system that interacts with the motors.

The Color/Light Sensor included in this feedback sensory system helps to determine *(or continuously check)* if the sensor is above a white or black area. In simple terms, the sensor *(The K'NEX sensor is actually made up of a single unit that includes two small sensors mounted next to each other)* acts like our eye and sends signals back to the Control Box which then sends commands to the Line Tracker to go forward or turn left or right.

Objectives

Students will be able to:

- Work in collaborative teams.
- Build the Line Tracker model or use the model from Lesson 1.
- Design and write a main program that includes procedures to enable the Line Tracker to follow a black line around a track drawn on the K'NEX Color Mat.
- Experience and practice using a feedback sensory system.

Materials

Each group will need:

- Built Line Tracker model *(Page 64 in the instructions book)*
- K'NEX Color Mat for Color/Light Sensor activities
- Open area in the classroom or hallway for the model to operate
- PC and K'NEX Robotics Programming Software
- K'NEX Robotics User's Guide
- Student Instruction Sheet A – Lesson 2
- Copies of Student Response Sheets 1, 2, & 3 - Lesson 2

Teacher's Notes

- Identify student groups of 2 - 3.
- Allow the students to explore the K'NEX Education materials and give them time to review the K'NEX Robotics User's Guide.
- Require that students show their progress before they begin to program their model.
- The Line Tracker model should be programmed to operate at low motor speeds so that there is enough time for the microcontroller to process the information and direct the motors to operate the Line Tracker smoothly. Once the robot follows a line successfully, ask the students to increase the speed of the Line Tracker to explore the limits of the system. Students may also need to experiment with the light values used to optimize performance.
- It is recommended that the supplied transformer be used to power the models while writing programs to conserve battery power for remote use. The Control Box will only draw power from the transformer when it is connected; when the transformer is disconnected from the Control Box it reverts immediately to battery power. The transformer will not charge batteries installed in the battery compartment.

The task for students is to:

- Explore the logic flow of a program that can power the Line Tracker to follow a black line.
- Program the model to actually follow a black line on the K'NEX Color Mat.
- Troubleshoot and redesign the program to ensure the model moves smoothly in their classroom environment.
- Complete one or more challenge activities to improve the operation of their model.

Explore a Program that can power the Line Tracker along a Black Line

Provide students with a copy of the flowchart shown in Figure 1.

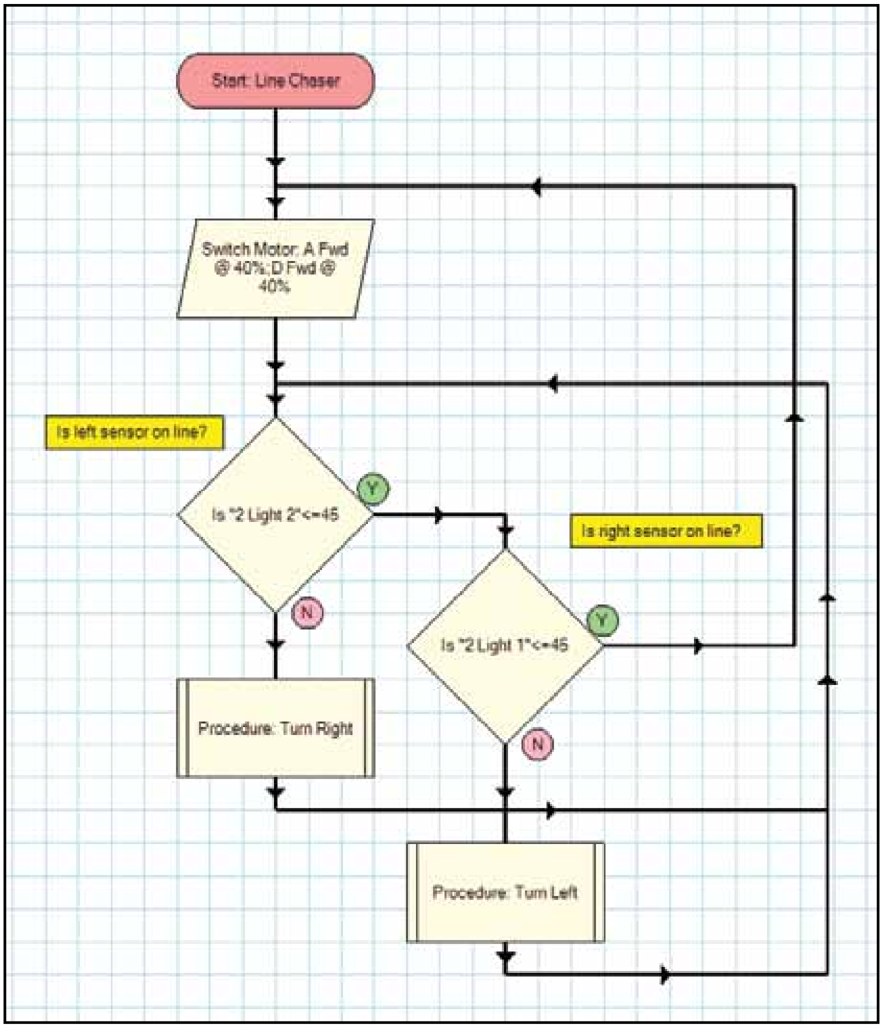


Figure 1: *Line Tracker Main Program Flowchart*

Ask students to review and explore the logic used in the flowchart. The K'NEX Robotics User's Guide will enable them to explore the various Decision and Output symbols used in the program. Ask them to explain in a written paragraph or list format how the program will operate to ensure that the Line Tracker actually stays on the line. Review the responses in a large or small group setting to ensure that students are able to follow the logic displayed by the flowchart.

As shown in the main program flowchart, the students will produce separate procedures that will power the Line Tracker along a black line on the K'NEX Color Mat. The directions to complete the program are provided on Student Instruction Sheet A – Lesson 2.

LESSON 2 – Student Instruction Sheet A

The following activities will introduce you to the use of sensors in robotic systems and will allow you to program the Line Tracker robot to respond to sensor input.

Requirements:

Complete Student Response Sheets 1, 2, and 3 as you work on this activity to keep a record of your work.

Introduction to Sensors

Sensors are an important part of any robotic design as they act like a feedback system. They provide information to the microcontroller, which acts like our brain to make decisions and send information to the motors that power the model. In this lesson, you will use the Color/Light Sensor to measure digital light values. Before programming, the operation of the Color/Light Sensor should be evaluated to determine the digital values the sensor identifies for black and white areas of the K'NEX Color Mat.

Before you begin:

- Review the building instructions for the Line Tracker.
- Read and review the K'NEX Robotics Programming Software User's Guide as it pertains to Decision and Output programming symbols.
- Read the Color/Light Sensor description provided in the K'NEX Robotics Programming Software User's Guide.

Steps:

1. Build the Line Tracker. Add the Control Box, Color/Light Sensor, Push Button Sensor and motors as shown in the Building Instructions.
2. **DO NOT** forget to power the Control Box. K'NEX recommends using the supplied transformer if possible during programming to conserve battery power.
3. Open the K'NEX Robotics Programming Software.
4. Click the 'connect' icon on the K'NEX Robotics Programming software to connect the software with the Control Box.
5. You will see Color/Light Sensor readings on line 2 under the "Sensors" heading in the I/O panel on the lower right side of the screen. Line 2 corresponds to Port 2 on the Control Box where the sensor is connected.
6. Place the model so that the Color/Light Sensor is on a white section of the mat.
7. Collect sensor readings at three white areas on the mat, enter them on Student Response Sheet 1 and average the data.
8. Place the model so that the Color/Light Sensor is on the black line on the mat.
9. Collect sensor readings at three black areas on the mat, enter them on Student Response Sheet 1 and average the data.
10. The average data values calculated on Student Response Sheet 1 will help decide a value that can be used to separate readings that indicate the sensor is above a white area or a black area.
11. After our testing at K'NEX, we determined that a sensor reading less than or equal to 45 indicates black and any value greater than 45 indicates white. Your values may vary from these results, please select your value as per the readings you receive from the sensor.
12. Once you have the digital values for the sensor, turn the model off, set it aside and start the logic design activity on Student Response Sheet 2.
13. Move on to the programming section for further instructions.

PROGRAMMING

Program Logic:

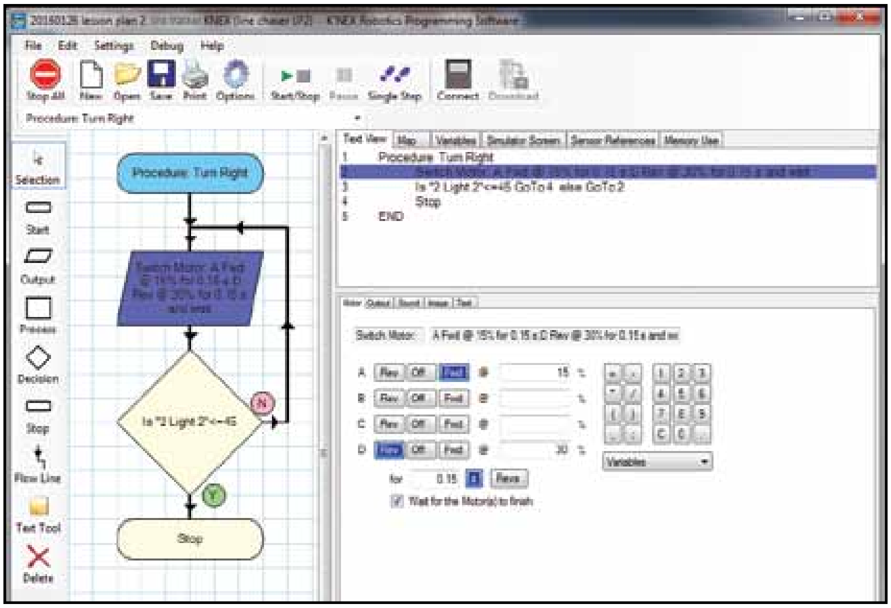
- The Line Tracker needs to know how to turn left, turn right, or go forward if it is to successfully follow the black line. Create separate procedures for each direction; LEFT and RIGHT.
- The Color/Light Sensor is actually made up of two smaller sensors mounted a small distance apart inside a single housing. These smaller sensors are labeled “light 1” and “light 2” in the software. As positioned in the Line Tracker model, sensor “light 1” is on the right side and sensor “light 2” is on the left side.
- For this lesson the goal is to follow the black line by keeping both sensors over the line.
- Based on our sensor readings at K’NEX, if sensor “light 1” has a reading of 60 and sensor “light 2” has a reading of 25, the Line Tracker is partially off the right side of the black line. The program will have to turn the model to the left to return the Line Tracker to the line.
- If sensor “light 1” has a reading of 25 and sensor “light 2” has a reading of 60 the model is partially off the left side of the black line. The Line Tracker will have to turn right to return to the black line.

How to create a procedure?

You should see a Start oval at the top of the edit area. Click on the Start and enter the procedure name “Turn Right” in the data box provided in the lower right portion of the screen. You can toggle between a “Start” and “Procedure” by clicking the labeled tabs just above the data box. Click the procedure tab and click OK. “Procedure: Turn Right” will appear in the oval and you are now ready to create the Turn Right procedure. *NOTE: Start ovals will be red, Procedure ovals will be blue.*

PROCEDURE: “Turn Right”

The Line Tracker must turn slowly in order to avoid overshooting the black line. The procedure logic for “Turn Right” is shown in Figure 2. The example suggests switching motor A forward at 15% power while at the same time motor D is powered in reverse at 30% power for 0.15 seconds. The “wait for the motor(s) to finish” box is checked so the motors run for the time specified. To see if the model has been rotated back onto the line, a Decision symbol is added to check the input from the Color/Light Sensor. The statement in the Decision symbol asks “Is 2 Light 2<= 45”. This tells the Control Box to check the value of Sensor “light 2”, (the left side) of the Color/Light Sensor plugged into port 2. If the sensor value is greater than 45 the sensor is still off the left side of the line. The model has not turned far enough and the right turn will be repeated. If the sensor value is less than or equal to 45, *(black)* the model is back on the line and the procedure exits back to the main program. *NOTE: It is best to always add a Stop to the end of each of your procedures.*



```
1    Procedure  Turn Right
2         Switch Motor: A Fwd @ 15% for 0.15 s D Rev @ 30% for 0.15 s and wait
3         Is "2 Light 2"<=45 GoTo 4  else GoTo 2
4         Stop
5    END
```

Switch Motor: A Fwd @ 15% for 0.15 s D Rev @ 30% for 0.15 s and wa

A	Rev	Off	Fwd	@	15	%
B	Rev	Off	Fwd	@		%
C	Rev	Off	Fwd	@		%
D	Rev	Off	Fwd	@	30	%

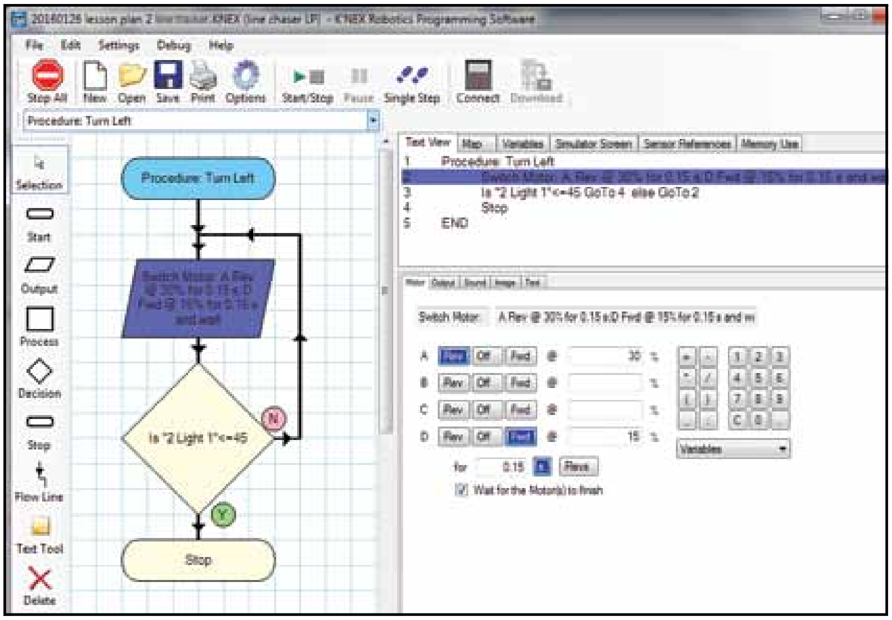
for 0.15 s Revs

☑ Wait for the Motor(s) to finish

Figure 2: *Turn Right.*

PROCEDURE: "Turn Left"

If you successfully followed the logic to write the "Turn Right" procedure, you should have no problem programming the "Turn Left" procedure. Click on the Start icon from the symbol toolbar to get a new Start oval for the Turn Left procedure. You will notice immediately that the "Turn Right" procedure has "disappeared" into the background. You can toggle between procedures and starts by selecting the desired option from the pull down menu below the control toolbar. You can also click on different procedures and starts from the map view in the upper right side of the screen. If the map tab is selected, click on the desired procedureor start to activate it. Create the procedure for a left turn. Refer to Figure 3 if necessary.



Procedure: Turn Left

```
1      Procedure: Turn Left
2            Switch Motor: A Rev @ 30% for 0.15 s,D Fwd @ 15% for 0.15 s and wa
3            Is "2 Light 1"<=45 GoTo 4  else GoTo 2
4            Stop
5      END
```

Switch Motor: A Rev @ 30% for 0.15 s,D Fwd @ 15% for 0.15 s and wi

A Rev @ 30 %
D Fwd @ 15 %
for 0.15 s

☑ Wait for the Motor(s) to finish

Figure 3: *Turn Left.*

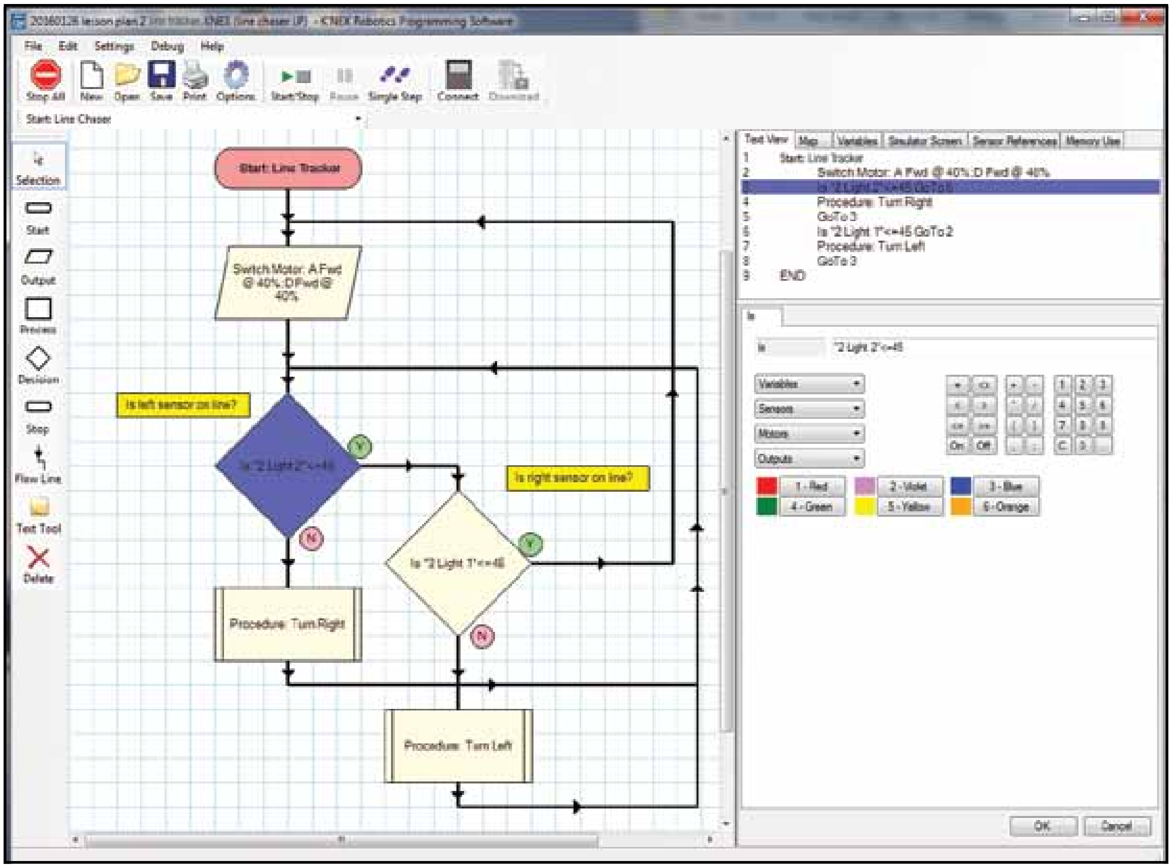
Main Program:

Click the "Start" icon from the Symbol toolbar on the left of the screen to start a new flowchart for the main program. Click on the "Start" oval in the edit area and enter an appropriate name for the main program. Be sure to select the Start Tab rather than the Procedure Tab when entering a name for the program.

When you operate the Line Tracker on the K'NEX Color Mat, you will start the model with the Color/Light Sensor centered on the black line. This means the first step in your program should be to make the model move forward. Place an Output Symbol under the Start in the edit area.
Based on our experience at K'NEX, a good starting speed for the Line Tracker is 40% motor power.
NOTE: See Figure 4 Main Program, if you are unsure of how to complete this task.

You are now ready to complete your program using the procedures you prepared earlier. The program will need to constantly check sensors 1 & 2 to be sure the model remains on the black line. To make that happen, you will need to add some decision symbols and loops to your program. The decision symbols will guide the program based on the sensor readings and the loops will allow the program to check the sensors over and over again.

A copy of the Line Tracker program developed at K'NEX is provided in Figure 4. The program has been provided to allow you to get your model operational. Test your model, troubleshoot its performance, and make changes to your procedures and program as necessary to improve the model's operation. Record any changes you make and the reason for those changes on Student Response Sheet 2 – Lesson 2.



```
1   Start: Line Tracker
2       Switch Motor: A Fwd @ 40% :D Fwd @ 40%
3       Is "2 Light 2"<=45 GoTo 6
4       Procedure: Turn Right
5       GoTo 3
6       Is "2 Light 1"<=45 GoTo 2
7       Procedure: Turn Left
8       GoTo 3
9   END
```

Is "2 Light 2"<=45

Figure 4: *Main Program.*

Challenge: Refer to Student Response Sheet 3 – Lesson 2 for a challenge activity for your group to complete.

LESSON 2

Student Response Sheet 1

Name ____ Group # __________

Measure the Color/Light Sensor digital readings for the black portion of the mat.

Color/Light Sensor Readings	Sensor 1 – Black Color	Sensor 2 – Black Color
Trial 1		
Trial 2		
Trial 3		
Final Average		

Explain your results and describe how this information will help as you program the Line Tracker to follow the black line.

Calculate the Color/Light sensor digital readings for the white portion of the mat.

Color/Light Sensor Readings	Sensor 1 – White Color	Sensor 2 – White Color
Trial 1		
Trial 2		
Trial 3		
Final Average		

Explain your results and describe how this information will help as you program the Line Tracker to follow the black line.

LESSON 2

Student Response Sheet 2

Name ____ **Group #** _________

Describe any changes that you made to the initial program provided by K'NEX to make the Line Tracker operate more efficiently in your classroom or setting.

What did you notice about the performance of the Line Tracker that led you to decide to make adjustments to the program?

List any changes you made to the program? Describe the logic behind those changes.

Draw or attach copies of the flowcharts for the changes you made to the program.

LESSON 2

Student Response Sheet 3

Name ____ Group # ________

Challenge – In the Main program shown in Figure 4, flow lines that emanate from the Turn Right and Turn Left procedures loop back to the first Decision symbol. In contrast, the flow line from the second Decision symbol loops back to the Output symbol.

Discuss the logic involved in this arrangement with your group and explain why this strategy is necessary in writing below.

LESSON 3:
Bi-Ped Robot

The Bi-Ped Robot model used in this lesson will help you program using strategies that are similar to how humans process information for simple tasks. In previous lessons (Motor Control and Line Tracking with the Line Tracker model), the interaction of sensors and motors were explored and practiced. In Lesson 3, the feedback system and communication between the Control Box, motors and sensors are used in more advanced ways to program the Bi-Ped Robot. The Robot has three sensors: Distance, Color/Light and Push Button. It also has four motors: one motor to rotate the head, two motors to power the legs and one motor to open/close the hand to grab or drop an object.
The Distance Sensor can measure the distance the robot is from an object, the Color/Light Sensor will determine if something of a certain color *(yellow)* is in the robot's hand, the Push Button is used as the operator determines when to grab an object *(ball)* that is not yellow.

The programming structure in this lesson will introduce a method called "procedure inside a procedure". This modular programming structure will expose students to a real time programming experience where even a simple task like grabbing a ball needs a structural programming flow to operate correctly and smoothly. This advanced level of logic is critical to student learning at an early age so they can investigate and practice modularizing the codes/procedures they use to optimize other programming strategies.

Objectives

Students will be able to:

- Work in collaborative teams.
- Design and write programs including embedded procedures.
- Create multi-layer procedures *(procedure inside a procedure)*.
- Experience real world applications of STEM concepts.
- Develop a logic strategy and convert that logic into a structured program.

Materials

Each group will need:

- Bi-Ped Robot model *(page 27 in the instructions book)*
- Open area in the classroom or hallway for the robot to walk
- PC and K'NEX Robotics Programming Software
- K'NEX Robotics User's Guide
- Copies of Student Instruction Sheet A - Lesson 3
- Copies of Student Response Sheets 1, 2, & 3 - Lesson 3

Teacher's Notes

- Identify student groups of 2 - 3.
- Allow students to explore the K'NEX Education materials and give them time to review the K'NEX Robotics User's Guide.
- Students may need help building the robot.
- Provide students with a screen shot that shows the completed main program.
- Complete the structured portion of the lesson with students.

- Request that students describe their thought processes and model building progress before they begin further programming with their model.
- Students may need help as they program.
- Ask students to verify the direction the motors turn so that the robot will walk forward.

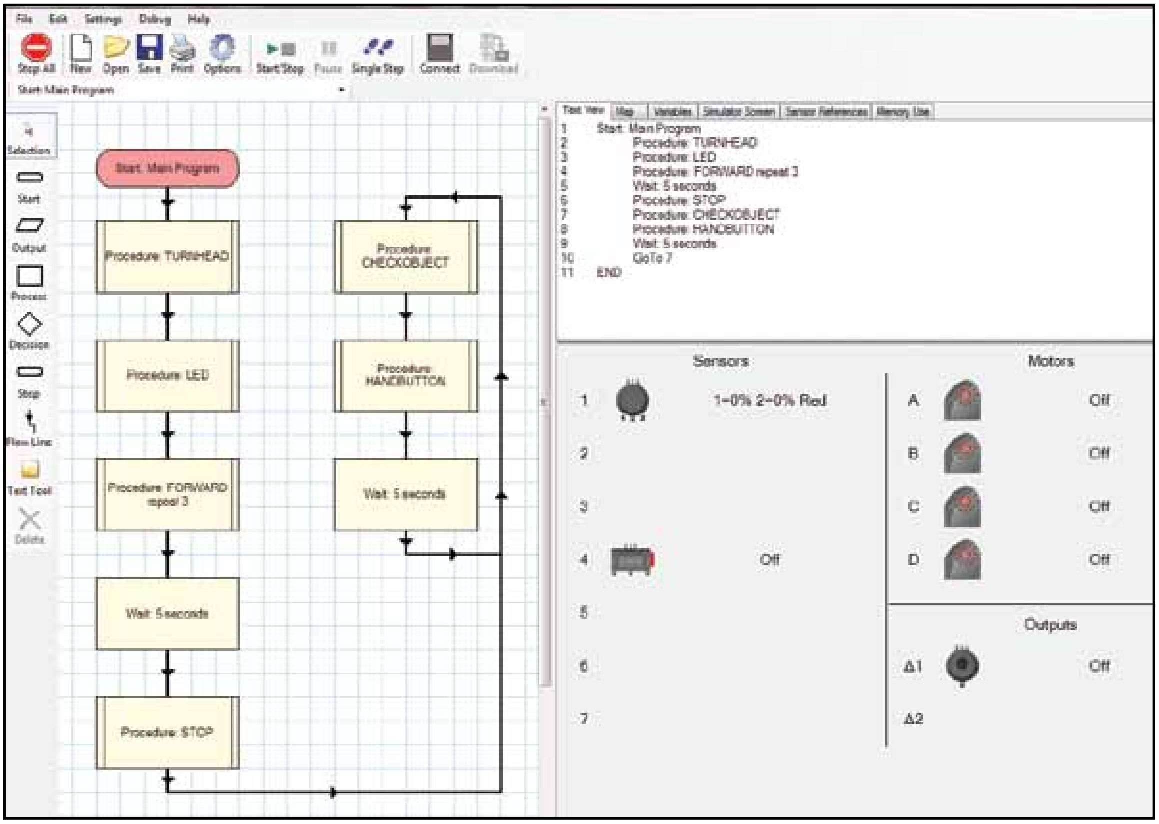
The students will learn to write separate procedures within procedures that they can incorporate into a main program in order to power the Bi-Ped Robot to look around, move forward, grab a ball, and finally drop that ball. The structured part of this lesson introduces the main program and allows you to guide students as they write the beginning steps of their programs. You can then turn the programming responsibility over to the students beginning with the first procedure within a procedure as outlined on Student Instruction Sheet A – Lesson 3.

The following few pages include instructions for the procedures and process symbols that make up the structured portion of the lesson along with screen shots you can use as you assist the students as they begin their programming. For more detail on how to generate procedures, review the K'NEX Robotics User's Guide. If you would like students to take responsibility for programming the model independently, copy the screen shots on the next few pages to direct their efforts. They can then move on to Student Instruction Sheet A – Lesson 3 to complete their programs.

Main Program:

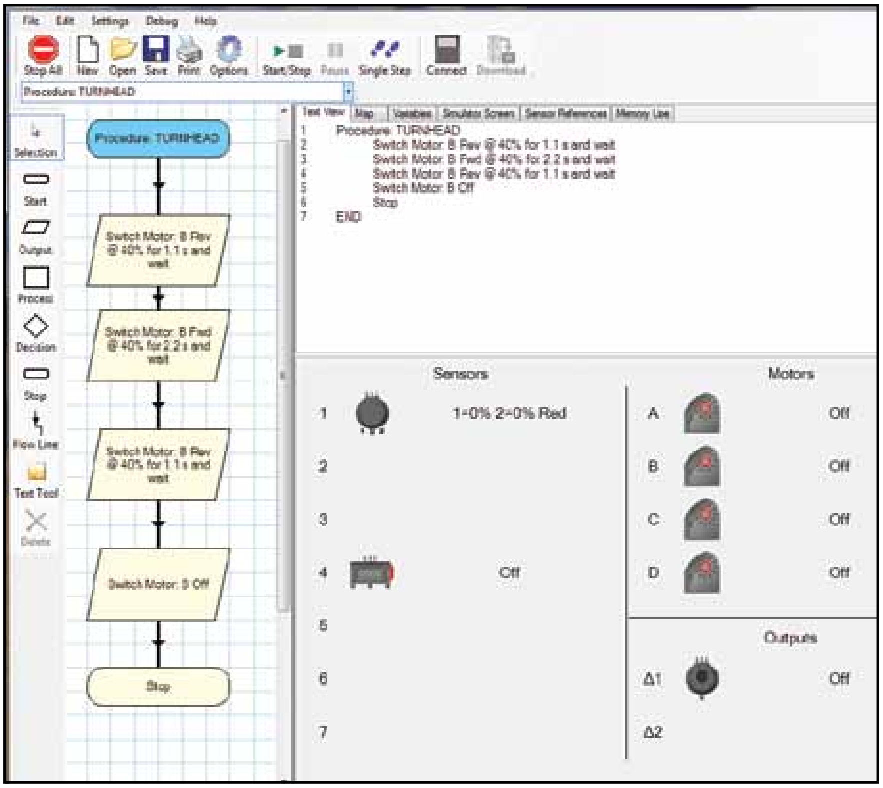
The main program shows the logic that could be used to program the Bi-Ped Robot to grab a ball *(Figure 1)*. A step-by-step method is described as follows:

1. The first block *(procedure)* in the flowchart as shown below in Figure 1 is "TURNHEAD." The procedure "TURNHEAD" instructs the Robot to move its head once left, once right, and back to straight ahead. The logic flow of the procedure "TURNHEAD" can be seen in Figure 2.



```
1     Start: Main Program
2          Procedure: TURNHEAD
3          Procedure: LED
4          Procedure: FORWARD repeat 3
5          Wait: 5 seconds
6          Procedure: STOP
7          Procedure: CHECKOBJECT
8          Procedure: HANDBUTTON
9          Wait: 5 seconds
10         GoTo 7
11    END
```

Figure 1: *Main Program.*

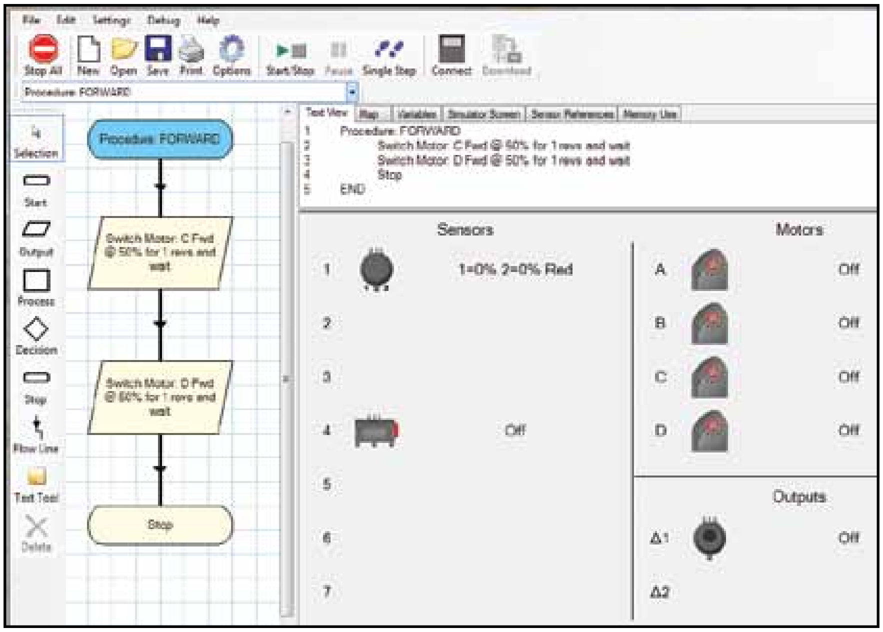


```
Procedure: TURNHEAD
        Switch Motor: B Rev @ 40% for 1.1 s and wait
        Switch Motor: B Fwd @ 40% for 2.2 s and wait
        Switch Motor: B Rev @ 40% for 1.1 s and wait
        Switch Motor: B Off
        Stop
END
```

***Figure 2:** "TURNHEAD" Procedure.*

To complete the "TURNHEAD" program as shown in Figure 2 complete the following steps.

- To turn the Robot's head, motor B turns at 40% speed in reverse *(that command turns the head to left)* for 1.1 seconds. Be sure to click the "Wait for motor(s) to finish" box when you set the motor percentage and the time for the motor to run.
- Next, motor B turns forward at 40% power for 2.2 seconds and directs the program to wait until the motor is finished *(the head turns right until the Robot is facing right)*.
- The final command turns motor B in reverse at 40% for another 1.1 seconds to bring the head back to its original position.
- *NOTE: The motor speed needs to be slow enough for an observer to see the Robot turn its head as smoothly as possible. Test different motor speeds and time delays until your Robot moves its head smoothly. Do not forget to turn motor B off at the end of the procedure.*



```
Procedure: FORWARD
        Switch Motor: C Fwd @ 50% for 1 revs and wait
        Switch Motor: D Fwd @ 50% for 1 revs and wait
        Stop
END
```

***Figure 3:** "FORWARD" Procedure.*

The "FORWARD" procedure enables the robot to walk forward for several steps.

- From Figure 1 you can see that the program logic flows from the "TURNHEAD" procedure to the "FORWARD" procedure. Motors C and D are connected to the two legs of the Robot. Figure 3 shows the flowchart for the Robot walking. The motors (C and D) are programmed to move forward at 45% speed. As you follow the logic in "FORWARD" you will notice motor C completes a revolution and then motor D completes a revolution. This process is repeated three times *(When you designate the process symbol for "FORWARD" as a procedure you will name it "Forward" and place a "3" in the 'repeat' space to the right of the title)*. The main program as seen in Figure 1 makes the Robot stop after a total of three repeats of the procedure. If you want the Robot to walk for more time, increase the number of repeats within the procedure.

To continue to the procedure inside a procedure, refer to Student Instruction Sheet A - Lesson 3. Steps 1 – 3 addressed the first section of the main program. The directions on the Student Instruction Sheets will allow students to develop the logic to complete the main program.

Student Instruction Sheet A – Lesson 3

The following activities will introduce you to the use of the Push Button and Color/Light Sensor in a robotic system and will allow you to program the Bi-Ped Robot to respond to sensor inputs. The sensor inputs will result in signals being sent to the motors to perform tasks. The main task assigned to the Bi-Ped Robot is to grab a ball and let go after a couple of seconds. Challenge activities are also included for you to attempt.

Before you begin:

- Review the building instructions for the Bi-Ped Robot
- Read and review the K'NEX Robotics Programming Software User's Guide as it refers to programs and procedures.

Steps:

1. Build the Bi-Ped Robot per the K'NEX Building Instructions.
2. **DO NOT** forget to power the Control Box and to switch it on.
3. Open the K'NEX Robotics Programming Software.
4. Click the 'connect' icon on the K'NEX Robotics Programming Software.
5. Move on to the programming section below.

PROGRAMMING

Previously you began a program to direct the Robot to move forward, grab a yellow ball and release it. The following steps will enable you to complete that program and provide some challenges to test your programming skills.

Grab and Throw Logic

The simple task of holding a ball and throwing is a process that requires a series of multiple signals in our brains. Our brain sends signals to our hands when the ball comes close to us to grab and throw it back. Isn't this how we play catch? Keep this in mind as you develop a logical sequence of actions to train and program the Robot to complete a similar action.

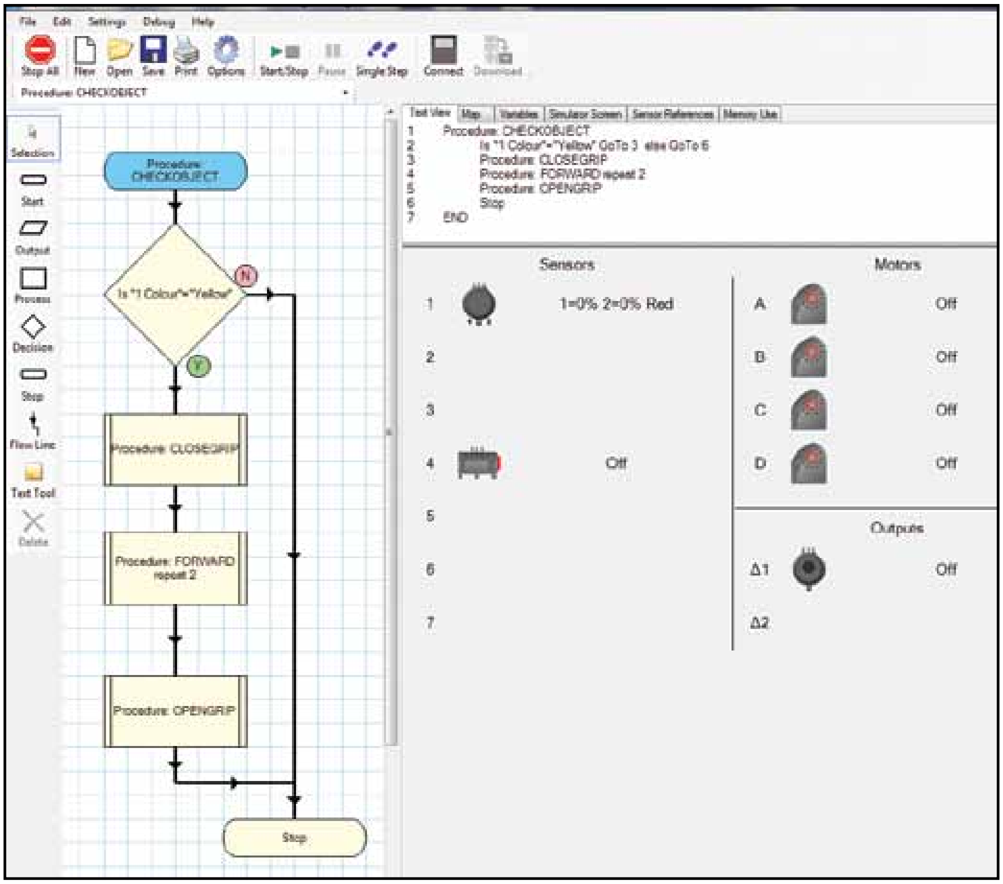
You will use an 'if - then' conditional statement as you program. For example, 'if' the ball is close, 'then' grab it. The same logic will enable you to create a procedure inside a procedure.

Procedure Inside A Procedure

When we want to hold something and drop it repeatedly, our brain processes these repetitive tasks and performs them quickly. You will program procedures within procedures to allow the Robot to complete repetitive tasks.

Program the Bi-Ped Robot to:

- Check the color of a ball placed in its hand with the Color/Light Sensor.
- "If" a yellow ball has been placed in its hand, "then" the hand will close and the Robot will take a few steps forward before dropping the ball.
- "If" the ball is not yellow, "then" the program will check to see whether or not the "Push Button" is pressed.
- "If" the "Push Button" is pressed, "then" the Robot will grab the ball for a short time before releasing it.
- "If" the "Push Button" is not pressed, "then" the program will repeat these steps over and over.



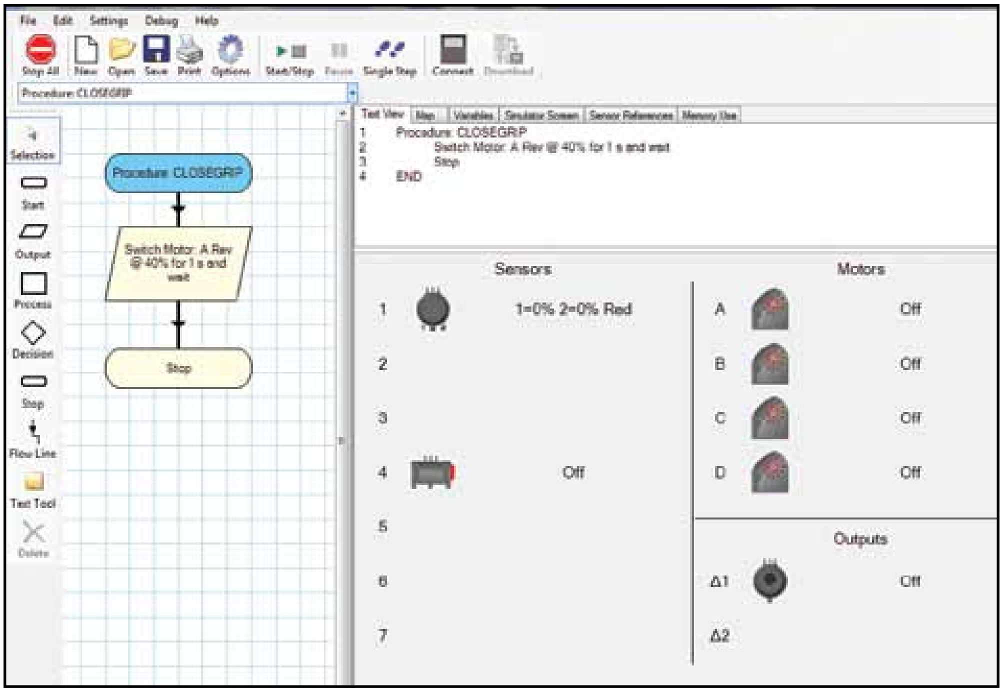
```
1   Procedure: CHECKOBJECT
2        Is "1 Colour"="Yellow" GoTo 3  else GoTo 6
3        Procedure: CLOSEGRIP
4        Procedure: FORWARD repeat 2
5        Procedure: OPENGRIP
6        Stop
7   END
```

Figure 4a: *Procedure "CHECKOBJECT".*

Figure 1 *(main program)* shows a flow logic where the block with the procedure "CHECKOBJECT" is activated. Figure 4a shows the details of the flowchart logic for the "CHECKOBJECT" procedure.

- As seen in Figure 4a, the Color/Light Sensor controls the function of the hand grip. *(Later when you run the program, you will place a yellow ball in front of the Color/Light Sensor.)* "If" the sensor identifies the color yellow, "then" it means the yellow K'NEX ball is ready to be grabbed and the procedure "CLOSEGRIP" is activated.
- When you place a "Decision" symbol in the program and click on it, you need to enter information in the input/output panel at the bottom right of the screen for the Color/Light Sensor to respond to the color yellow. From the "Sensor" pull down menu, select "1 Light 1," click "=" and click "yellow." The "Decision" symbol will send the program in one direction if the sensor identifies yellow and in another direction for any other color.
- Figure 4b shows the logic of the "CLOSEGRIP" procedure that closes the Robot's hand to grab the ball. Motor A turns forward at 40% for 1 second to grip the ball.
- After the ball has been grabbed, the Figure 4a logic shows that the procedure "FORWARD" is activated and the Robot walks forward for a few steps. *(This is the same "Forward" procedure you used earlier except the number of repeats has been changed to 2.)* When the Robot stops walking the procedure "OPENGRIP" is activated and the ball is released.
- Figure 4c shows the "OPENGRIP" logic flow where the motor A is in reverse drive mode at 40% power for 1 second causing the robot to drop the ball.

Figure 1 *(main program)* has another block showing the procedure "HANDBUTTON". When this procedure is placed in the same loop as the "CHECKOBJECT" procedure, the program will check the condition of the Color/Light Sensor to see if it is reading yellow and then check the Push Button to see if it is depressed. Figure 5 shows the logic for the procedure "HANDBUTTON". The Figure 5 logic strategy checks the condition of the pushbutton. If it is pressed, the procedure "OPENCLOSEGRIP" procedure *(Figure 6)* is activated. This procedure will open and close the hand three times.



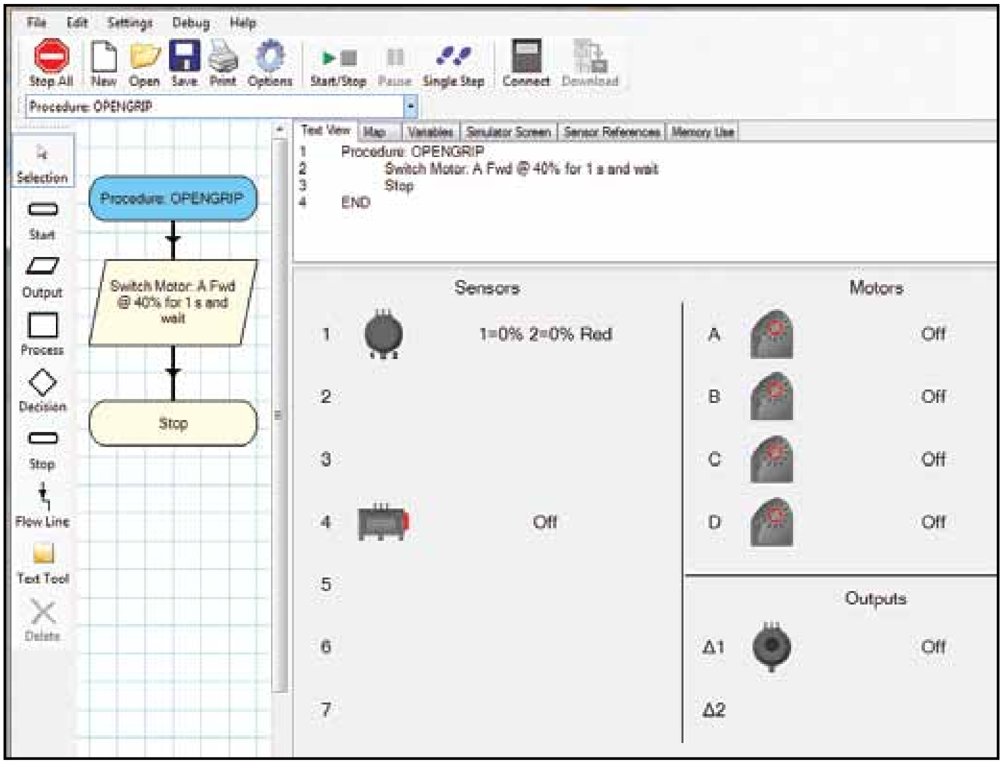
```
Procedure: CLOSEGRIP
        Switch Motor: A Rev @ 40% for 1 s and wait
        Stop
END
```

Sensors
1 1=0% 2=0% Red
2
3
4 Off
5
6
7

Motors
A Off
B Off
C Off
D Off

Outputs
Δ1 Off
Δ2

Figure 4b: *Procedure "CLOSEGRIP".*



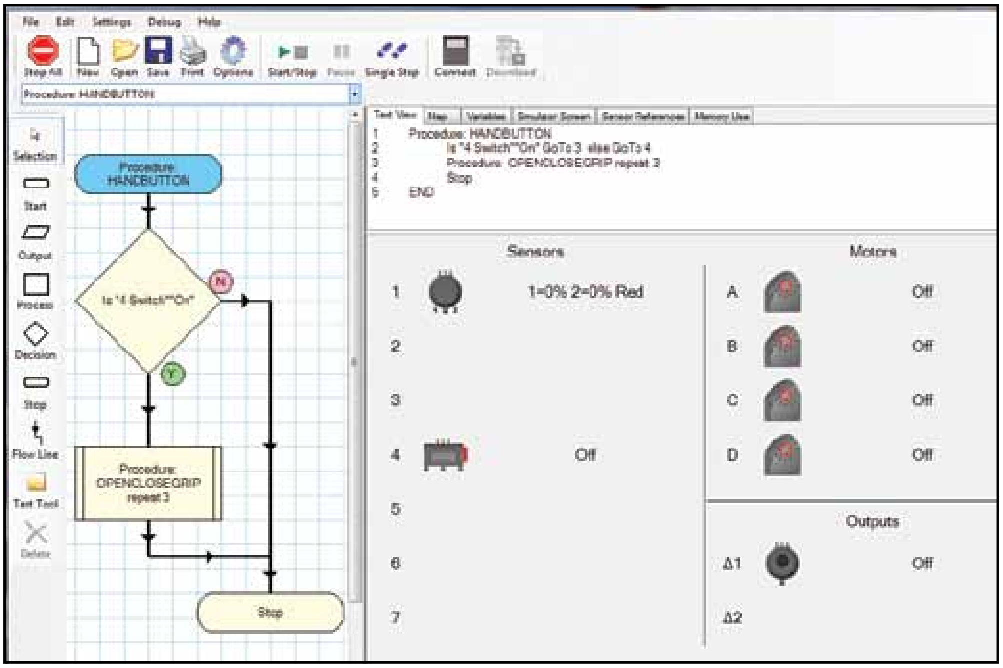
```
Procedure: OPENGRIP
        Switch Motor: A Fwd @ 40% for 1 s and wait
        Stop
END
```

Sensors
1 1=0% 2=0% Red
2
3
4 Off
5
6
7

Motors
A Off
B Off
C Off
D Off

Outputs
Δ1 Off
Δ2

Figure 4c: *Procedure "OPENGRIP".*



```
Procedure: HANDBUTTON
        Is "4 Switch""On" GoTo 3  else GoTo 4
        Procedure: OPENCLOSEGRIP repeat 3
        Stop
END
```

1 1=0% 2=0% Red

4 Off

A Off

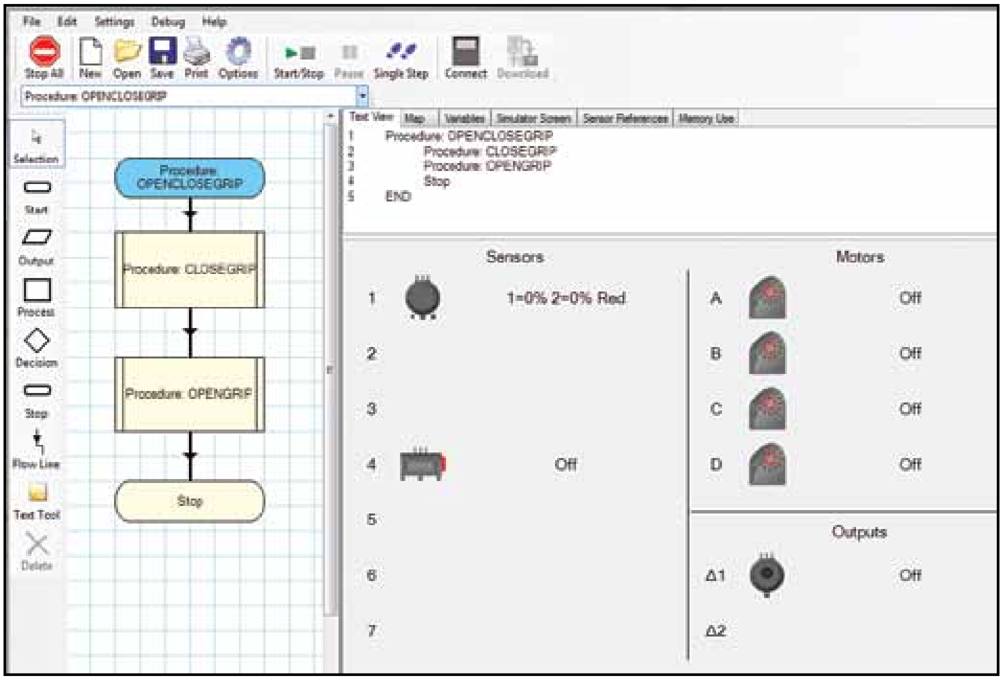
B Off

C Off

D Off

Δ1 Off

Figure 5: *Procedure "HANDBUTTON".*



```
Procedure: OPENCLOSEGRIP
        Procedure: CLOSEGRIP
        Procedure: OPENGRIP
        Stop
END
```

1 1=0% 2=0% Red

4 Off

A Off

B Off

C Off

D Off

Δ1 Off

Figure 6: *Procedure "OPENCLOSEGRIP".*

If you have designed your Main Program and procedures carefully and identified appropriate values for your sensors, the Bi-Ped Robot should have operated as you expected. Make every effort to refine your program to optimize the actions of the Robot.

When you are satisfied with the operation of your model, move on to the challenges below or allow your imagination to guide you as you program your model to complete tasks of your own design.

Your **challenge** is to:

CHALLENGE 1

Design and write a program and procedures using the Distance Sensor to stop the model when the Bi-Ped Robot is a specified distance from an object or a wall. For this activity, refer to the average digital value for the Distance Sensor from Student Response Sheet 1 before creating a procedure for STOP WHEN YOU SEE A WALL.

When you have designed your program, enter it on Response Sheet 1 or attach printed copies of the program and procedures to Response Sheet 1.

Test your model, troubleshoot its performance, and make changes to your procedures and program as necessary to improve the model's operation.

CHALLENGE 2

Design and write a Main Program and procedures that utilizes each of these sensors and inputs; Color/Light Sensor, Push Button Sensor, and Distance Sensor to complete a task.

When you have designed your programs, enter them on Response Sheet 2 or attach printed copies of the program and procedures to Response Sheet 2.

Test your model, troubleshoot its performance, and make changes to your procedures and program as necessary to improve the model's operation.

CHALLENGE 3

Redesign one of your main programs to incorporate the LED that is included in the head of the model.

When you have designed your programs, enter them on Response Sheet 2 or attach printed copies of the program and procedures to Response Sheet 2.

Test your model, troubleshoot its performance, and make changes to your procedures and program as necessary to improve the model's operation.

LESSON 3

Student Response Sheet 1

STOP THERE'S A WALL!

Name ____ **Group #** __________

Describe the information that a flowchart design needs to contain in a procedure to check for an obstacle.

Write a flowchart design for the obstacle detection procedures.

LESSON 3

Student Response Sheet 1

(continued)

Name ____ **Group #** ________

Draw a complete main program flowchart design for the Bi-Ped Robot that incorporates your newly designed procedure(s).

LESSON 3

Student Response Sheet 2

COLOR/LIGHT SENSOR PUSH BUTTON & DISTANCE SENSOR

Name ____ **Group #** __________

CHALLENGE – Design and write a program and procedures that utilizes each of these sensors and inputs (Color/Light Sensor, Push Button & Distance Sensor) to complete a task.

Describe the information that a flowchart design needs to contain in a procedure to complete the challenge above.

LESSON 3

Student Response Sheet 2

(continued)

Name ____ **Group #** __________

Write a flowchart design for the procedure(s) that will meet the challenge.

Draw a complete main program flowchart design for the Bi-Ped Robot that incorporates your newly designed procedures.

LESSON 3

Student Response Sheet 3

INCORPORATE AN LED

Name ____ **Group #** __________

Describe the information that a flowchart design needs to contain in a procedure to include the LED.

Write a flowchart design for the procedure that includes an LED.

Draw a complete main program flowchart design for the Bi-Ped Robot that incorporates your newly designed procedure(s).

LESSON 4:
Search & Rescue Robot

The Search & Rescue Robot is another interactive K'NEX robot. Students build and program its motors and sensors to interact with each other. In previous lessons the concept of motors, their performance and motion was introduced. In this lesson, the robot uses four motors: moves using two motors, moves its claws up and down using a third motor and grabs a ball and stand using a fourth motor. The robot has sensors that can send feedback signals to the controller which in turn will activate motors on the model. The Search & Rescue Robot is designed to travel to an object's location to pick up a ball on a stand, carry the ball and stand to a new location, and move away. This requires knowledge of different types of sensors to control the movement of the robot and the pick/grab action used to gather a ball. Students will begin by using the push button sensor to gather data that will direct the robot to close the claws to pick up and later release the ball and stand. The robot also includes a distance sensor that can be used to control its drive motors.

Objectives

Students will be able to:

- Work in collaborative teams.
- Build the Search & Rescue Robot.
- Design and write a looped program, create procedures and call the procedures within the main program.
- Gain experience working with and programming a push button and distance sensor.
- Learn and practice strategies to calibrate motors and sensors.

Materials

Each group will need:

- Built Search & Rescue Robot model *(Go to www.knex.com/robotics for instructions)*
- PC and K'NEX Robotics Programming Software
- K'NEX Robotics User's Guide
- Student Instruction Sheet A – Lesson 4
- Copies of Student Response Sheets 1, 2, & 3 – Lesson 4

Teacher's Notes

- Identify student groups of 2 - 3.
- Allow the students to explore the K'NEX Education materials and provide time to review the K'NEX Robotics User's Guide.
- Request that students show their progress before they begin to program their model.
- Students may need help as they program their models.

A partial, main program flowchart template is provided to help students organize their thought processes as they develop the procedures and symbols to complete the program.

PROGRAMMING

The Search & Rescue Robot is designed to search for a ball on a stand, grab it in its claws and carry it to a new destination. At its destination, it will release the ball and stand and move away. These tasks require that the Robot perform the following functions:

- **Drive forward and reverse:** it will need two motors to drive forward, drive backward, and turn right or left.
- **Lift up and down**: it will need one motor to move the claws up and down.
- **Grab**: one motor can both open and close the claws.
- **Sensors**: the digital readings from a sensor sends input information to the controller that directs the motors to carry out the Robot's motions.

Main Program:

Creating a main program was described in Lesson 1 and creating a procedure was described in Lesson 2. For more information, read the K'NEX Robotics User's Guide.

When the Robot was constructed, a Push Button Sensor was placed at the base of the claws. The push button will be pushed if it comes in contact with a ball and stand. The main program must include a loop that allows the Control Box to keep the model moving forward until the push button is pressed. The loop is initiated with a Decision symbol and a Flow Line that returns the program flow back to a point between the Open and Forward procedures. Once the Robot has begun to move forward, it will continue until the push button is pressed by, in this case, the ball and stand.

Once the push button is pressed, the procedure 'CLOSE' is activated and the claws close to grab the ball. After it grabs the object, the procedure 'UP' is initiated to lift it up the claws and 'LEFT' turns the robot to the left. 'FORWARD2' drives the robot forward for a specific distance. The robot carries the ball to its destination and drops it (procedure 'DOWN' and 'OPEN'). The Robot then drives away from the ball moving backwards (procedure 'BACKWARD'). Also, you will note the main program includes procedures to allow the LED to shine green when the ball is grabbed and shine red when the ball is released.

As this is the fourth lesson, students should be able to program the various procedures that are required to complete the main program.

Student Instruction Sheet A – Lesson 4

As you create the following procedures, you will be introduced to the use of a push button as a sensor. This information and the procedures you design will allow you to program the Search & Rescue Robot to respond to sensor input. Sensor input sent to the controller will activate the various motors.

The task assigned to the Search & Rescue Robot is to approach a ball and stand, grab them, carry them to another location, release them and back away. When the ball and stand are in the claws of the Robot, a green LED is lit, when the ball and stand are released, the LED is lit.

Requirements:

Complete Student Response Sheets 1, 2, & 3 as you work on this activity to keep a record of your work.

Before you begin:

- Review the building instructions for the Search & Rescue Robot.
- Read and review the K'NEX Robotics User's Guide. Pay particular attention to creating a procedure within a program and how to include the procedure in the main program.

Steps:

- Build the Search & Rescue Robot per the K'NEX Building Instructions.
- **DO NOT** forget to power the Control Box.
- Open the K'NEX Robotics Programming Software.
- Click the 'Connect' icon on the top menu bar to connect the computer and Control Box.
- **Ensure that the claws on your Robot are in the up and closed position.**
- **Place a ball on one of the stands you constructed and place it at some distance in front of the Search & Rescue Robot.**
- **Test the function of the motors, sensors, and outputs on your model. Enter the information requested on Student Response Sheet 1 and complete the rest of the sheet before you begin programming.**

Your task is to write a program to operate the Search & Rescue Robot. Use the template shown in Figure 1 to begin the main program. Name the main program 'Main – Rescue Lesson' and place blank Process, Decision, and Output symbols in place as shown. Additional symbols required to finish the program should be placed to the right side of the grid to keep the main program to a single screen. You will be prompted to add additional symbols when necessary.

The main program will include many procedure symbols that you must create in order to complete the main program. Some of these procedures will be easy and some will be more difficult. A few complete procedures will be shown in Figures 2, 3, and 4. Add flow lines to your main program as you complete the procedures and symbols that will make up the main program. The running times and revolutions of the motors in these procedures are based on sample programs prepared by K'NEX. Your particular running times and revolutions of the motors may vary from these printed values.

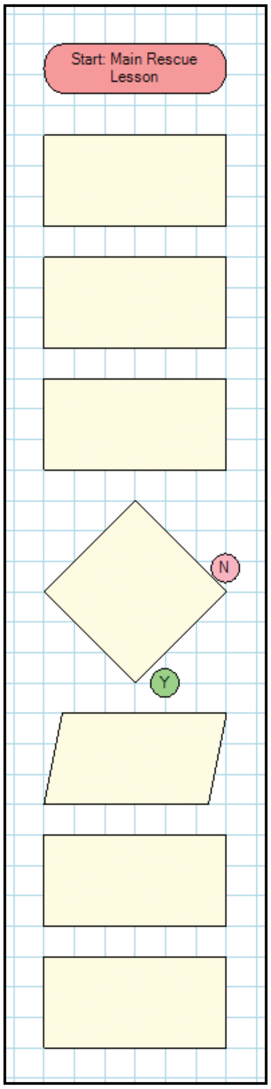


Figure 1: *Partial Template for the 'Main – Rescue Lesson' Program.*

Program the following procedures to complete the flowchart that will enable your model to operate. Refer to Student Response Sheet 2 as you program to answer questions related to specific procedures and programming strategies.

Procedure Down:

(This procedure will appear in the first Process symbol on the Figure 1 template. Succeeding procedures will follow in order.)

This procedure moves the Robot arms/claws down. Name the procedure and program it to slowly move the claws to their down position. You will need to determine which motor raises and lowers the claw, the direction to turn the motor, a power percentage for the motor, and the number of revolutions the motor will turn. *(HINT: the motor will turn less than one revolution and you will need to have the procedure 'Wait' until the motor stops.)*

Procedure Open:

This procedure opens the Robot claws. *(HINT: Remember that the motor percentage and the number of revolutions may vary slightly from those in the Output symbol.)* See Figure 2.

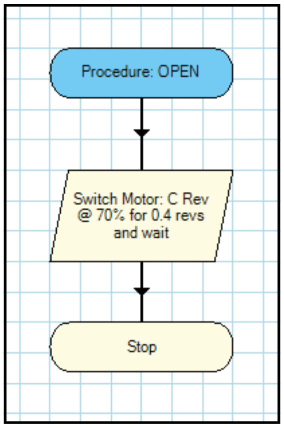


Figure 2:
Procedure Open.

Procedure Forward:

This procedure will power both motors in the same direction to move the Robot slowly forward. Do not include any of the following in the Process symbol inside this procedure: "s" or "Rev" or "Wait."

Decision Symbol:

Program the symbol to determine if the pushbutton is on. The tricky part is placing the Flow Lines for this symbol.

Output Symbol:

Program the symbol to turn the LED on green. *(HINT: Check to see where the LED is plugged in.)*

Procedure Close:

Program the procedure to close the claws. *(HINT: Can you reverse the information you programmed in the 'Open' procedure?)*

Procedure Up:

This procedure raises the Robot claws. See Figure 3.

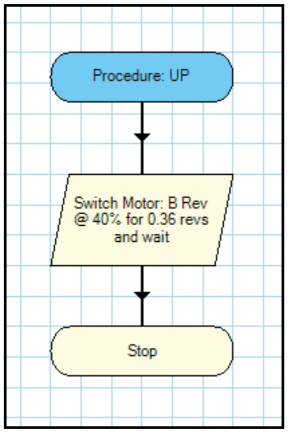


Figure 3:
Procedure Up.

Procedure Left:

This procedure turns the Robot to the left. Notice the motors only turn only 0.54 revolutions to complete the turn and the motors move at slightly different speed settings. See Figure 4.

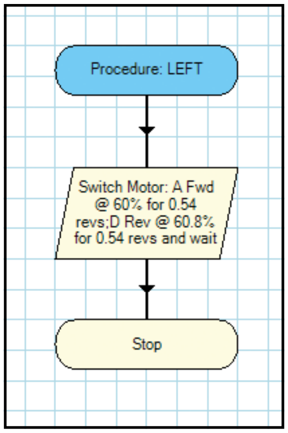

Procedure: LEFT
Switch Motor: A Fwd @ 60% for 0.54 revs;D Rev @ 60.8% for 0.54 revs and wait
Stop

Figure 4:
Procedure Left.

Procedure Forward 2:

This procedure moves the Robot forward for just 1.3 revolutions and waits for the motors to complete the revolutions. Check your previous 'Forward' procedure to determine the power settings for the motors to move in a straight line.

Procedure Down:

You programmed this procedure earlier. Place your Process Symbol and call the 'Down' Procedure again.

Procedure Open:

You programmed this procedure earlier. Just place your Process symbol and call the 'Open' procedure again.

Output Symbol:

Program the symbol to turn the red LED on.

Procedure Backward:

This procedure moves the Robot backwards for two revolutions of the motors and waits until the motors have completed their task.

Output Symbol:

Turn the red LED off.

If you have all of these procedures in place, you should be able to complete the main program and test your Robot.

CHALLENGE ACTIVITY:

Your challenge is to design and write a main program and procedures using a Distance Sensor for the Search & Rescue Robot. The Robot must pick up a ball with its stand and drop it in a new location. *(You may also use the push button and LED in your main program.)*

When you have designed your program and procedures, enter them on Response Sheet 3.

Test your model, troubleshoot its performance, and make changes to your procedures and main program as necessary to improve the Robot's operation and performance.

SUPER CHALLENGE ACTIVITY:

Your challenge is to program the Search & Rescue Robot to pick up and move two or even three balls and stands with a single main program. Good Luck!

LESSON 4

Student Response Sheet 1

Name ____ **Group #** _________

List the location of the various motors, sensors, and outputs.
Also list what each does to support your program.

MOTORS:	**ACTION: What does each item do?**
Motor A	
Motor B	
Motor C	
Motor D	
SENSORS:	
Push Button	
Distance	
OUTPUT:	
LED	

Explain how the information on the chart above will be of assistance as you program your model.

LESSON 4

Student Response Sheet 2

Name ____ **Group #** __________

Procedure Down:

Why is it important to move the claws down slowly?

Procedure Forward:

The Process symbol for this procedure only includes two motor designations and power percentages for each. There is no motor running time, number of revolutions, or wait for the motor(s) to finish commands included in the Process symbol. Why is this strategy necessary for the operation of this procedure?

Decision Symbol:

Draw a diagram to show where you placed the Flow Lines for this symbol. When you tested the program did the Robot respond to the Push Button as you expected? If not, what changes did you have to make to the Decision symbol or Flow Lines to correct the situation?

LESSON 4

Student Response Sheet 3

Name ____ **Group #** _________

Design and write a main program and procedures that utilize the Distance Sensor on the Search & Rescue Robot. The Robot must pick up a ball with its stand and drop it in a new location. *(You may also use the Push Button and LED in your main program.)*

Describe how your main program operates in paragraph form. Include information about how you utilized the Distance Sensor to assist the Robot in completing its task. Also, describe any problems you encountered during the programming and/or testing of the Robot.
How did you correct these problems?

Print a copy of your main program and procedures for this challenge and attach them to this Student Response Sheet.

LESSON 5:
Ball Factory

Robotic systems are of many types; some are small, some are drivable, but some are also huge as we often see in industrial settings. A simple example is the Ball Factory. The Ball Factory will give students a perspective of how factory conveyor systems function as part of a quality control process in an industrial setting. Students will gain experience as they build a system to lift, pick up a ball, sense the color of the ball, decide if the ball needs to be eliminated from the system and much more. Previous lessons performed communication tasks between the motors and sensors. Here students will continue to do the same with more checks as the balls move through four separate stations that make up the Ball Factory.

The motion of balls from one station to the other with quality control checks along the way requires an interactive program where processes depend on previous actions of the system. Programming the Ball Factory system will prepare students to take on challenging logic development activities and communicate with multiple sensors and motors in a complex program. The feedback sensor system feature of this lesson is more robust than previous lessons. It continuously checks sensor readings to determine if the ball traveling on the conveyor is the correct color (yellow: in this lesson) and if it is not, the ball is removed from the system. In simple terms the sensors act like eyes and send check signals back to the Control Box, which in turn sends commands to the motors to operate the Ball Factory.

Objectives

Students will be able to:

- Work in collaborative teams.
- Build the Ball Factory following the Building Instructions.
- Explore the various mechanical functions of the Ball Factory such as lift, pick and drop, rotational and translational motion.
- Design and write a loop program, create procedures and call the procedures into the main (flowchart) program.
- Gain advanced experience using a feedback sensor system.

Materials

Each group will need:

- Built Ball Factory model *(Go to www.knex.com/robotics for instructions)*
- A red, green and yellow K'NEX ball
- Table/counter space to operate and store the model
- PC Computer and K'NEX Robotics Programming Software
- K'NEX Robotics User's Guide
- Student Instruction Sheet A – Lesson 5
- Copies of Student Response Sheets 1, 2, & 3 – Lesson 5

Teacher's Notes

- Identify student groups of 2 - 3.
- Allow the students to explore the K'NEX Education materials and give them time to read the sections of the K'NEX Robotics Programming Software User's Guide related to procedures and incorporating those procedures within a main program.
- Provide time for students to build the Ball Factory.
- Request that students show their progress before they begin to program their model.
- Ensure students test their motors to determine the directions they will move when programmed.
- Instruct students to begin their programming.

By this point students should have developed programming skills that will allow them to produce their own Main Program and Procedures for the Ball Factory.

Ball Factory Concept

The feedback system the students program to communicate between the Control Box and sensors will allow balls to travel smoothly and be sorted as necessary based on their color.

Station #1:

A ball is placed at the bottom of the chain drive escalator. The Distance Sensor identifies that a ball is present and the Control Box starts the motor on the chain drive to lift the ball to a ramp that carries the ball down to the next station.

Station #2:

The ball drops into position near a Color/Light Sensor. The sensor determines the color of the ball and transmits that information to the Control Box.

OPTION 1: If the ball is yellow, a motor turns a wheel that pushes the ball forward along another ramp to an area where it can be picked up.

OPTION 2: If the ball is not yellow, a trap door opens to cause the ball to drop out of the Ball Factory as it is pushed forward.

Station #3

A mechanical arm swings a claw over the top of the yellow ball, drops to capture the ball and carries the ball to a delivery area that allows the ball to travel through the Ball Factory again.

Four motors have been included to power the different systems on the Ball Factory. The Control Box will receive data from three different sensors and use that information to direct the motors. Students will also add an LED to the Ball Factory. When all of these devices are programmed, the Ball Factory will operate to move a yellow ball continuously through the system. Balls that are not yellow will be rejected and dropped out of the Ball Factory.

Student Instruction Sheet A – Lesson 5

The following activities will introduce you to the use of multiple sensors in robotic systems and will allow you to program the Ball Factory. The program you design will move different color K'NEX balls one at a time through the Ball Factory using four motors. The motors will be directed by the Control Box based on sensor data that is collected as the ball moves from one location to another. You will be successful when a yellow ball moves continuously through the Ball Factory and red or green balls are rejected and removed from the Factory.

Requirements:

Complete Student Response Sheets 1 & 2 as you work on this activity to keep a record of your work.

Before you begin:

- Review the building instructions for the Ball Factory before building.
- Read and review the K'NEX Robotics User's Guide. Pay particular attention to creating a "procedure" within a program and how to include the procedure in the main program.
- Review the operation and programming of the Color/Light Sensor, Distance Sensor and Push Button Sensor.

Steps:

1. Build the Ball Factory as shown in the Building Instructions.
2. Collect a red, green and yellow K'NEX ball.
3. Power the Control Box.
4. Determine the action of each of the motors, sensors and outputs and record that information on Student Response Sheet 1.
5. Complete Student Response Sheet 2 as you complete this lesson.
6. Move on to the programming section below for further instructions.

PROGRAMMING

ACTION 1:

To operate the Ball Factory, place balls one at a time at the bottom of the chain drive escalator. There is a Distance Sensor at this point that will be used to start the balls moving. Your first task is to prepare a procedure that will identify that there is a ball at the base of the chain drive. If so, activate a motor to power the chain drive long enough to raise the ball to the ramp that runs down one side of the Ball Factory. Check the operation of your procedure several times to troubleshoot its operation. Also, ensure that the procedure places the ball so it will roll down the ramp. Add an LED that indicates the Ball Factory is operating.

ACTION 2:

When the ball reaches the end of the ramp, it will drop into a "V" shaped space formed by pairs of white rods on a wheel. The Color/Light Sensor shines its light on the ball as it determines its color. Add a check to the program to determine if the ball is yellow. *(NOTE: a Decision Symbol will allow you to continue now and return to the symbol later to program what will happen if the ball is not yellow.)*

LESSON 5 – Student Instruction Sheet A (continued)

ACTION 3:

Write a procedure that will advance the ball in the system if the ball is yellow. The procedure will operate a motor driven wheel to push the ball down another ramp which leads to a catch area that will stop the ball. Continue by programming the motor that controls the mechanical arm to swing over the ball, drop down and grasp the ball with its claw. Notice that the mechanical arm presses a pushbutton as it comes down on the ball.

ACTION 4:

Program a check (Decision symbol) to determine if the push button is pressed. If so, direct the motor controlling the mechanical arm to rotate the arm through a 180 degree flip. This will return the yellow ball to the start of the loop, where it will begin moving through the Ball Factory again.

ACTION 5:

Return to the Decision Block that checked to see if the ball in the system was yellow. Write a procedure that will reject red and green balls that may pass through the Ball Factory. If the ball is red or green, program the motor which operates the trapdoor in the ramp to open the door. The door should be opened before the wheel that advances the ball out of the Color/Light Sensor is activated. The LED color should be changed during this process to indicate a ball has been rejected from the Ball Factory. The Main program should restart at this time.

ACTION 6:

Ensure that your Flow Lines are organized so the program is able to run continuously.

LESSON 5

Student Response Sheet 1

Name ____ Group # ________

List the location of the various motors, sensors, and outputs.
Also list what each does to support your program.

MOTORS:	ACTION: What does each item do?
Motor A	
Motor B	
Motor C	
Motor D	
SENSORS:	
Push Button	
Distance	
OUTPUT:	
LED	

Explain how the information on the chart above will be of assistance as you program your model.

LESSON 5

Student Response Sheet 2

Name ____ **Group #** __________

You programmed 6 different actions *(and possibly more)* to complete this lesson.
What problems did you encounter as you programmed each of these actions?
Indicate how you were able to overcome these issues.

ACTION 1:

ACTION 2:

LESSON 5

Student Response Sheet 2

(continued)

Name ____ Group # __________

ACTION 3:

ACTION 4:

LESSON 5

Student Response Sheet 2

(continued)

Name ____ Group # __________

ACTION 5:

ACTION 6: