Oregon Robotics Tournament and Outreach Program

Programming Techniques Workshop for Mindstorms® EV3

2019

Opening doors to the worlds of science and technology for Oregon’s youth
Instructor Contacts

Terry Alexander  
taalexand@gmail.com  
(503) 970-9243

Ken Cone  
kencone@gmail.com  
(503) 415-1465

Don Nelson  
don-nelson@ortop.org  
(503) 550-8028

Jim Ryan  
jrryan57@gmail.com  
(503)-803-0388

Roger Swanson  
swanson@hevanet.com  
(503) 297-1824
Loridee Wetzel
loridee-wetzel@ortop.org
503-486-7622, ext. 102
Workshop Goals

- The earlier Workshops have introduced very basic programming to get you started with your robots
- In this Workshop we delve more deeply into just the programming
  - Introduce techniques that we think you are likely to use sometime in your coaching careers
  - Discuss the Why as well as the How
Agenda

- Tour the EV3 software to review all the available EV3 blocks
- Distinguish those with higher utility from those with lower utility
- Hands on exercises to try out the new, higher utility blocks
Caveat about this Workshop

- Don’t expect to go right back to your teams and teach them all this new stuff you have learned.
- Look for opportunities when the youngsters are ready to pick up a new fundamental technique.
- Our goal is to prepare you better so you are ready when a new teaching opportunity arises.
- And remember: “The kids do the work!”
Review EV3 Blocks You’ve Used

- **Action Blocks – Move Steering**
  - Has multiple modes
  - Available parameters vary with mode
  - Match ports to robot

- **Flow Control Blocks – Wait Blocks**
  - One for each sensor
  - Compare and Change options
  - Program flow of control stays on block until condition of block is met
  - Match ports to robot
Move Steering Blocks

Which motors are controlled

Mode
Steering
Speed/Power
Number of Rotations
Brake or Coast
Wait Blocks

Wait For Reflected Light Sensor

Compare Value
- Comparison Operator
- Comparison Value

Change Value
- Change Direction
- Change Amount

Light Sensor Port
EV3 Block Review

- Review the EV3 Software to look at all programming blocks that are available
EV3 Block Summary

High Utility Blocks
- Motor and Move
- Wait
- Loop
- Switch
- Sensor Blocks
- Data Blocks (variables)
- My Blocks

Low Utility Blocks
- Brick Status Light
- Loop Interrupt
- Array
- Round
- Text
- Random
- Advanced

Useful for Debugging
- Sound
- Display

You should know already
Will cover tonight
Let’s Start with Variables

- How would you write a program to move a certain number of inches straight ahead?
Convert Inches to Revolutions

- I = inches you want to move
- D = diameter of wheels (these small wheels are about 2 1/4”)
- C = circumference of wheels
  = \pi \times D = 3.14 \times D
- R = revolutions
  = \frac{I}{C}

How do we get these calculations into a program?
Variables

- A variable is a container to hold a value
- Each variable has a name
- Each variable has a type
  - Numeric
  - Logic
  - Text
- Each variable can be
  - Written – put a value into the variable
  - Read – retrieve the value from the variable
Variable Blocks

Add the variable name here

Note the different kind of parameter boxes for each variable block above
Variable Blocks

Add the variable name here

- Write Numeric
- Read Numeric
- Write Logic
- Read Logic
- Write Text
- Read Text

Inputs:
- Numeric
- Logic
- Text

Outputs:
- Numeric
- Logic
- Text
Variable to Control Motor Speed

- Let’s write a program together to use a variable to control the motor speed in a Move Steering block
Use Variable to Control Speed

- Set the initial value of the Speed variable
- Use a Data Wire to Connect the Speed variable to the speed parameter of the Move Steering block. Note: Output of Speed variable connects to input of Move block, and the numeric types match.
You Try It

- Write a program to go straight ahead a number of inches that you save in a variable in your program
Throughout this workshop an answer is given in the slide set so you have something to review when you get home. Don’t look ahead until you have tried the exercise yourself. 😊
Move Forward a Distance in Inches

Set the distance to travel

Set wheel diameter

Set the value of pi as a constant

Multiply pi times the wheel diameter and put the result in the variable Circum

Divide the distance by the circumference and use the result to set the number of revolutions in the Move Steering Block
Let’s Try Something New

- What if I wanted the robot to stop if the touch sensor is pressed OR if the light sensor has detected a black line?
Steps to Develop a Program

- Understand what is to be accomplished
- Write down a list of tasks describing how you would solve the problem
- Break the tasks up into simpler subtasks
- Program the subtasks, try them, and debug them
- Put the subtasks together for the final program
How Would You Approach This?

- Write down a list of steps you would take to get your robot to stop when the touch sensor is pressed OR the light sensor detects a black line.

- Do this thought process without thinking about the programming ramifications.
One List of Steps

- Start the motors to get the robot moving.
- Check if the touch sensor has been pressed.
- If it has, stop the robot.
- If not, check if a black line has been detected.
- If so, stop the robot.
- Otherwise go back and repeat the above steps
Loop Blocks

Name of loop

This output value provides the count of iterations of the loop
Switch Blocks – If and Case Statements
Stop on Touch or Light Sensor

- You try it.
- Write a program so that your robot will stop if the light sensor detects a black line OR if the touch sensor is pressed.
Note that the robot will stop if either the touch sensor is pressed or a black line is detected, but did you notice that the loop is still executing? What might we do to exit the loop so that we can continue with something else after the robot stops?
Use a Logic Variable to Control the Loop

Define a logic variable “StopLoop” and initialize it to False. Use “StopLoop” to control the loop. When either event happens, set “StopLoop” to true, and control will exit the loop.

Note the Sound block after the loop to show that the program does indeed exit the loop. Sound blocks are good debugging aids.
Sound Block

Play a file from a list of files provided by the EV3 software

Play a tone as specified by the frequency

Play a note as specified by the note of the scale

Volume

Duration

Mode
• 0 – Play the sound once, and the program waits until the sound is finished
• 1 – Play the sound once, and the program continues immediately
• 2 – Repeat the sound until another Sound Block is executed, and the program continues immediately
Homework Exercise

- Use the Loop Interrupt block to do the same thing as the last exercise
Loop Interrupt Solution

[Diagram of a loop interrupt solution with details about starting motors, testing touch sensors, and logic for handling sensor inputs]
Use math on the logic outputs of the Sensor Blocks to produce an OR result that controls the loop.
Display Blocks

- A Display Block is another useful debugging aid
- Two modes to display a text value
  - Pixel mode
    - 178 pixels left and right
    - 128 pixels up and down
  - Grid mode – easier for text
    - 22 columns of 8 pixels each
    - 12 rows of 10 pixels each
    - Small characters are 1 row and 1 column
    - Large characters are 2 rows and 2 columns
Display Blocks – Grid Mode

- Text to be displayed
- Click on field to choose wired

Text supplied on a wire
Erase screen before display
Column to start display
Row to start display
Black or white text
Text size
  0 – small font
  1 – small bold font
  2 – large font

Pixel mode is very similar
Display a Number

- Write a program to display a number
- Helpful if you want to know what number your program is using at a particular point
Program to Display a Number

Wired option

Set the value to display into the variable Number

Display the variable Number. Use the wired option and connect the output of the Read Variable block to the input text value in the Display block.

Wait 3 seconds so you have time to read the value on the screen.
Subroutines or My Blocks

What if you find that you are using the same sequence of blocks in many places in your program?

Examples:
- 90 degree spin turn
- Go forward to stop on a black line

Define that sequence as a My Block and save the effort of duplicating blocks and save memory at the same time
Building a My Block

- Select the blocks from the program
- Choose My Block Builder from the Tools Menu
- Give the My Block a name, fill in the description, and choose an icon (optional)
Move in a Square

- A “primitive” way to move in a square
- How big is this program?
  - Download the program
  - Use Memory Browser in the Tools Menu
  - Size is 832B
- Demonstrate memory management in the software
Move in a Square with My Blocks

- Same program with My Blocks is 610B
- Note that if I double click on a My Block, it opens the code for inspection or editing.
Managing My Blocks

- All the My Blocks in a project are available in the My Blocks area of the Programming Palette
- They can also be found in the Project Properties section of the EV3 software
Parameters in My Blocks

- My Blocks can be defined to have both input and output parameters
- Before you hit the Finish button in the My Block builder, click the + button to add a parameter and click the Parameter Setup tab
  - Type in a Parameter Name
  - Choose Input or Output
  - Choose Number, Text, or Logic
  - Give it a Default Value (optional)
  - Choose the Parameter Style
You Build a My Block

- Use the program to display a number to build a My Block that has one input parameter and will display the value of the input on the EV3 display screen.
Display Number My Block

Display the value of the input parameter. Use the wired option and connect the input parameter to the input text value in the Display block.

Wait 3 seconds so you have time to read the value on the screen.
Debugging Example

- Go back to your most simple Stop on Black Program
- Use your display number My Block to display the light value when the robot stops.
- Add a Move Block to move ahead a small amount (20 degrees)
- Now use your display number My Block to display the light value.
- What did you notice?
But First, Sensor Blocks

- The last new block we’ll cover in the workshop.
- We have used Wait Blocks associated with sensors (the Orange blocks)
- There are other sensor blocks that let you read what a sensor is doing at a particular place in your program (the Yellow blocks)
Reflected Light Sensor Block

- Measure Reflected Light
- Compare Reflected Light

Value of light measured
Logic value of comparison
Display Reflected Light

- Now use your My Block to display the reflected light seen by your robot as described on the Debugging Example slide
Using the Display Number My Block

Note that two blocks have been added at each point where we want to see the light sensor value.
What Did You Notice in this Last Exercise?

- The light sensor value when the robot first stops will meet the trigger value constraint.
- The value 20 degrees further, however, can be quite different.
- If the light sensor senses right on the border between black and white, the value will be less than white but greater than black – it might look like green.
Other Debugging Ideas

- If you don’t know where the robot is in the program, use a Sound Block to just play a tone to see if the robot has gotten to a particular point.

- What value does my variable have? “Print” the value to the EV3 screen to see the value.

- Take a subtask out of a more complex larger program and debug it separately rather than trying to debug it in the full more complex environment.
Class Exercises – Summary 1

- Drive a specified number of inches
  - Variables
    - Define a variable
    - Write a value into a variable
    - Read a value from a variable
  - Do math on variables
  - Use data wires to connect the output of a variable or other programming block to the input of another programming block or variable
Class Exercises – Summary 2

- Stop if the touch sensor is touched OR if the light sensor detects a dark area
  - Loop blocks
  - Switch block
  - Logic controlled loop block
  - Loop interrupt block

- Debugging aids
  - Sound block
  - Display block
Class Exercises – Summary 3

- Display a number on the EV3 screen – My Block with parameter
- Display the reflected light value when sensor first encounters a darker area and again after moving 20 degrees further
  - Sensor block
  - Use of the display a number My Block
More Exercises for the High Achievers or To Do at Home

- Write a program that will stop only on a green line using reflected light sensor. (An example is attached at the end of the slide set.)
- Convert the first exercise to a My Block with a parameter giving the distance to drive in inches
- Use distance sensor in CanDo challenge
- Use color sensor to do the stop on green only exercise
Program to Stop on Green

- We did the following light sensor exercises in Workshop 1.
  - Stop on black or green
  - Stop only on black
- Obvious follow-up is to stop only on green
- What is the list of tasks that you would use?
Program to Stop on Green

- One possible list of tasks
  - Go forward and stop on green or black (We did this program in Workshop 1. No need to reinvent the wheel.)
  - Decide if robot has stopped on green or black
  - If green, stop the program
  - If black, continue the program to the next green or black block
Note this addition, which was not in our steps. I discovered the need for it in the debugging process.