Programming with Excel

Workshop for FLL Coaches

Tony Ayad

June 2012

Outline

- What is programming with Excel and Why use it
- The Command Interpreter
- Standard Commands list
- How to create your own commands?
- Bonus:
  - Proportional Line Follower
  - the Master Program

Programming with Excel

- What?
  - The NXT can read simple text files
  - The text file can only contain a single value on a line
  - Excel is used to create commands on a single line to move, turn, etc.
  - The Excel commands are converted to text files and downloaded into the NXT
  - The NXT reads the text files and executes the commands by a program that is created by your team

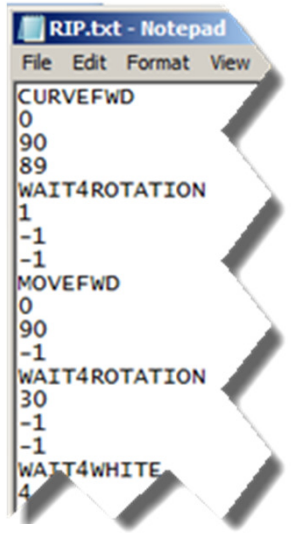
- Why?
  - Downloading programs takes seconds
  - Programs can be very long without crashing PC
  - Each step can be documented
  - Different version of the same program can be created
  - Work on same program from different computers
  - Something new and different for advanced team to keep them challenged…

Using Excel - Explained

Step 1: create program in Excel

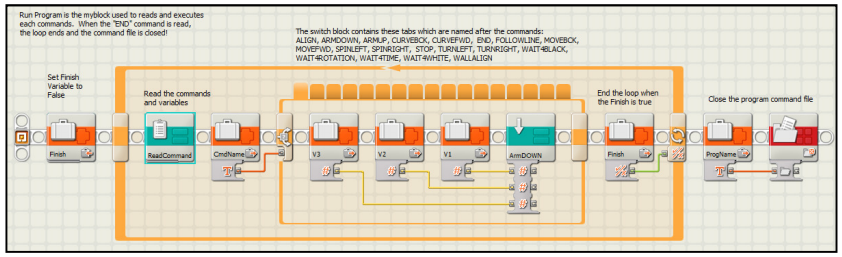
1	RIP				Build Program
2	OutPutFileName	C:\Users\lokomets\Documents\LEGO Creations\MINDSTORMS Projects\Profiles\Default\RIP.txt			
3	Command	Variable 1	Variable 2	Variable 3	Description
4		Get the blue rat first			
5	CurveFWD	0	90	89	Adjust the path toward the gap to left or right
6	Wait4Rotation	1			for a number of inches
7	MoveFWD	0	90		move straight
8	Wait4Rotation	30			Get to the area before the black line the runs int
9	Wait4White	4	75		wait for the line leading to the sink
	4Bl		50		bl le th

Step 2: Create the text file by running the “Build program” macro



Step 3: download program
Text file into NXT

Step 4: run your **command interpreter** program
That reads and executes commands in the text file



What you need to do on the NXT

- Teams need to design, develop and create their own commands (move, turn, spin, etc.)
- Teams need to create their own NXT "**Command interpreter**" program to read the text file and run each of the commands.
- An example interpreter program will be overviewed in this workshop

NOTE: the Excel is just a "black box" that creates text files.

EXCEL – Design your own command in "***Commands***" worksheet

Commands **MUST** be described in the Commands worksheet before they can be used in programs

	A	B	C	D	E
1	**Command**	**Variable 1 Description**	**Variable 2 Description**	**Variable 3 Description**	**Command Explanation**
2	**ALIGN**	Not Required	Not Required	Not Required	Align robot with black line
3	**ARMDOWN**	Duration Degrees	Power (0 - 100)	Not Required	Drop the arm a number of degrees.
4	**ARMUP**	Duration Degrees	Power (0 - 100)	Not Required	Raise the arm a number of degrees.
5	**CURVEBCK**	Duration (Inches); 0 for unlimited	Left Motor Power (0-100)	Right Motor Power (0-100)	Arc Backwards for a number of inches
	CURVEFWD	Duration (Inches); 0	Left Motor Power (0-	Right Motor	Arc Forward for a number of inches

Command name:
- Describes the action
- 12 characters in length
- No spaces
- This name **MUST** correspond with each tab name in the NXT-G Command Interpreter myblock

Variables:
- Each command can have up to 3 variables
- The description entered here will appear as column heading in program files
- Remember: only numeric

Describe the action being performed.

Remember: teams control what goes in the ***Commands*** worksheet

EXCEL - Program Worksheet Explanation

Exact location and name of the file to be created

Command Button that creates the text file

	A	B	C	D	E
1	**RIP**				*Build Program*
2	OutPutFileName	C:\Users\lolcomets\Documents\LEGO Creations\MINDSTORMS Projects\Profiles\Default\RIP.txt			
3	Command	Duration in Degrees (0 for	Power (0 - 100)	Not Required	Use the MOVE block to move forward. Leave inches to 0 for unlimited
4		**RIP stands for Rat, Ice Cream and Pizza**			
5		**Get the blue rat first**			
6	MoveFWD	0	100		Move toward the gap between the sink and fence
7	Wait4Rotation	2	1100		
8	Wait4White	4	75		wait for the line leading to the sink

Command name is entered in this column. When the command line is blank, the entire line is ignored. If the command name not found in the ***Commands*** worksheet, it will be highlighted in red.
Note: Last command in your program must be "**END**"

Variable are entered in those columns.
ONLY numeric values are allowed

Describe the action being performed. This is just for documentation.

Column headings will change as you scroll through the rows.
The titles are obtained from the ***Commands*** worksheet

EXCEL – how to

Create a new Command:

1. Go to ***Commands*** worksheet
2. Enter command name in column 1 (use upper case)
3. Describe the variables in columns 2, 3, and 4
4. Provide a description of the command.
5. Make sure that the command interpreter has a new tab to accommodate the new command.

Create New Program:

Options 1 –

- Press and hold an existing program tab and click and drag the tab to a new position (you'll see a "+" sign appear)

Option 2 –

- Right Mouse Click on the program tab you wish to copy; a popup menu will appear (exhibit 1);
- click on the "Move or Copy" option;
- the "Move or Copy" window will appear (exhibit 2);
- click to check the "Create a Copy" button,
- click on "(move to end)",
- then click on the "OK" button.

Exhibit 1

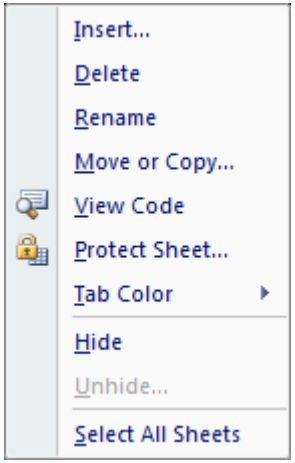
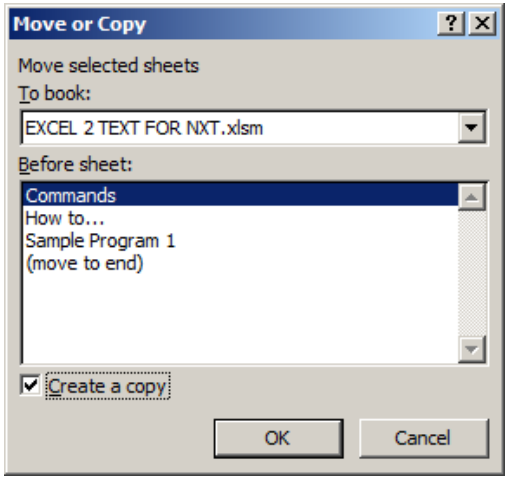


Exhibit 2



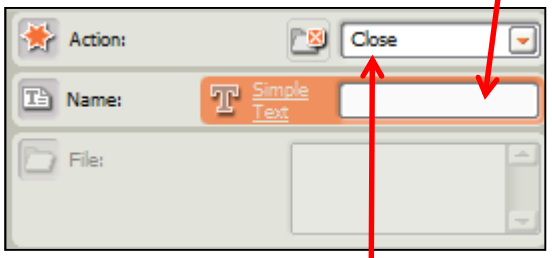
What is needed on the NXT - General Command

- The following is a sample list of Actions / MyBlocks / Commands that need to be programmed into the NXT
  - MoveFWD and MoveBCK – use the MOVE block
  - TurnLeft, TurnRight – make a turn using one MOTOR block
  - SpinLeft, SpinRight – turn using both motors that are moving in opposite direction
  - ArmUP, ArmDown – Raise or lower the arm motor
  - Wait4Dark – detects when light sensor value is less than specified
  - Wait4Bright – detects when light sensor value is more than specified
  - Wait4time – wait a number of seconds
  - Wait4Rotation – wait until rotation sensor
  - LineFollower – follow the line
  - WallSquare – bump and align against a stationary object (e.g., wall)
  - Stop – stops both motors

The **FILE ACCESS** Block – details

The **FILE ACCESS** block is used to work with files on the on the NXT

Control Panel



Name: file name goes here. MAKE SURE IT IS A TEXT FILE CREATED WITH PROGRAM such as “Notepad”.

Action:s
- Read = read one line
- Write = write one line to the end of the file
- Close: close the file
- Delete = Delete the file from NXT

INPUTS:

File Name

Text to write

Number to write

True if error is encountered

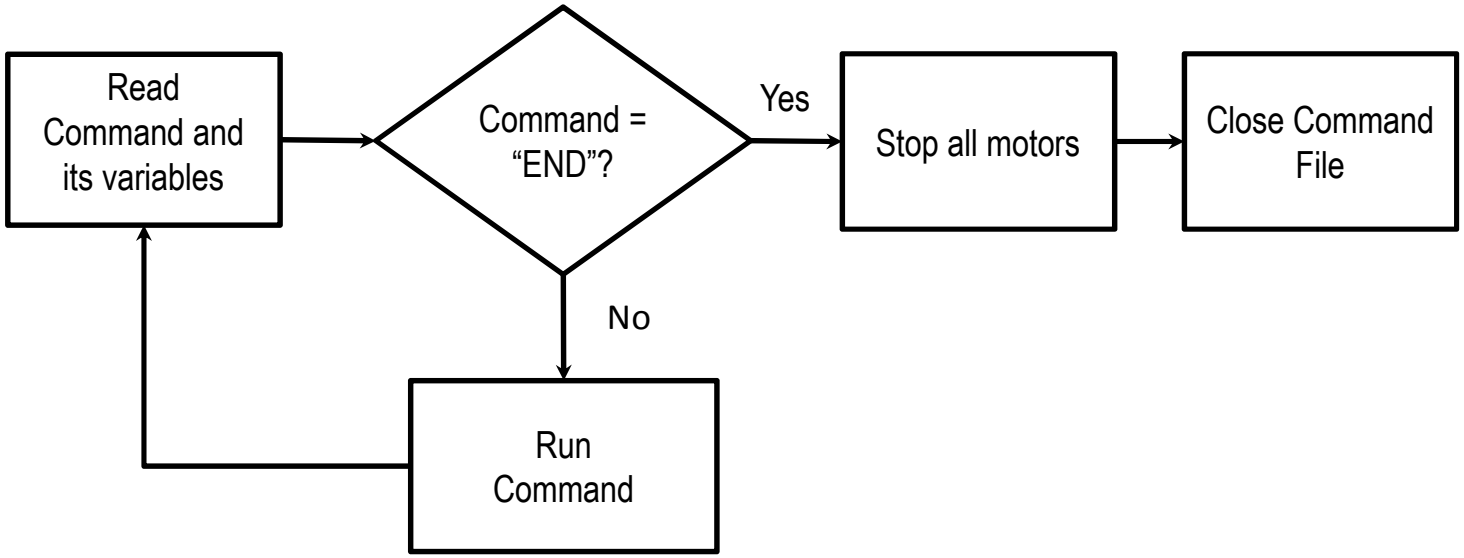
Text value (READ action) – maximum length of text value is 12

Numeric value (Read action) – can be decimal value

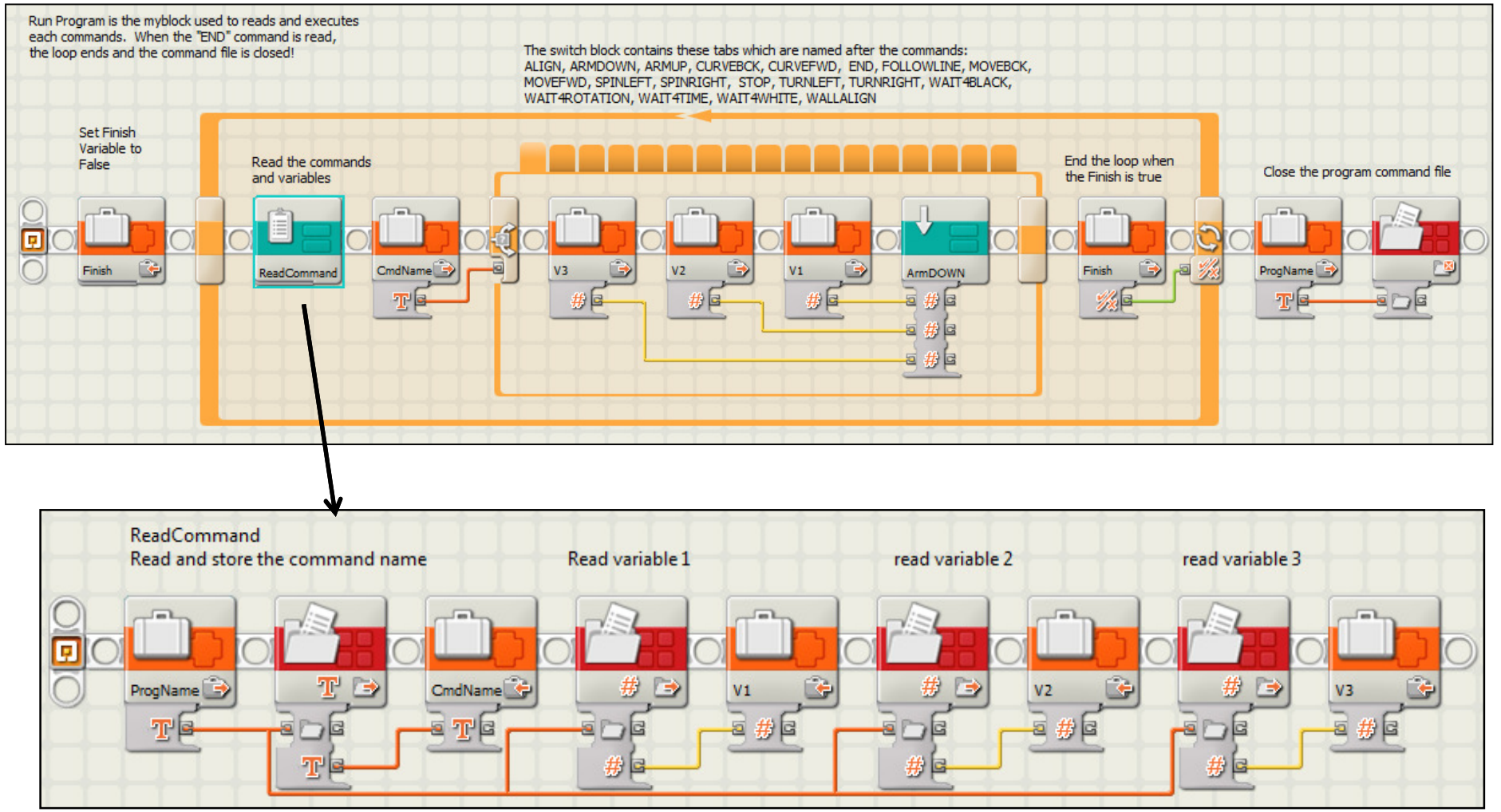
What are the type of Errors?
- Read when end of file is reached
- Read from number port and value is text
- Write when NXT is out of memory
- Close and the file doesn’t exist
- File name is not found on the nxt

Command Interpreter Flow Chart

- Note: example program will be developed during this workshop

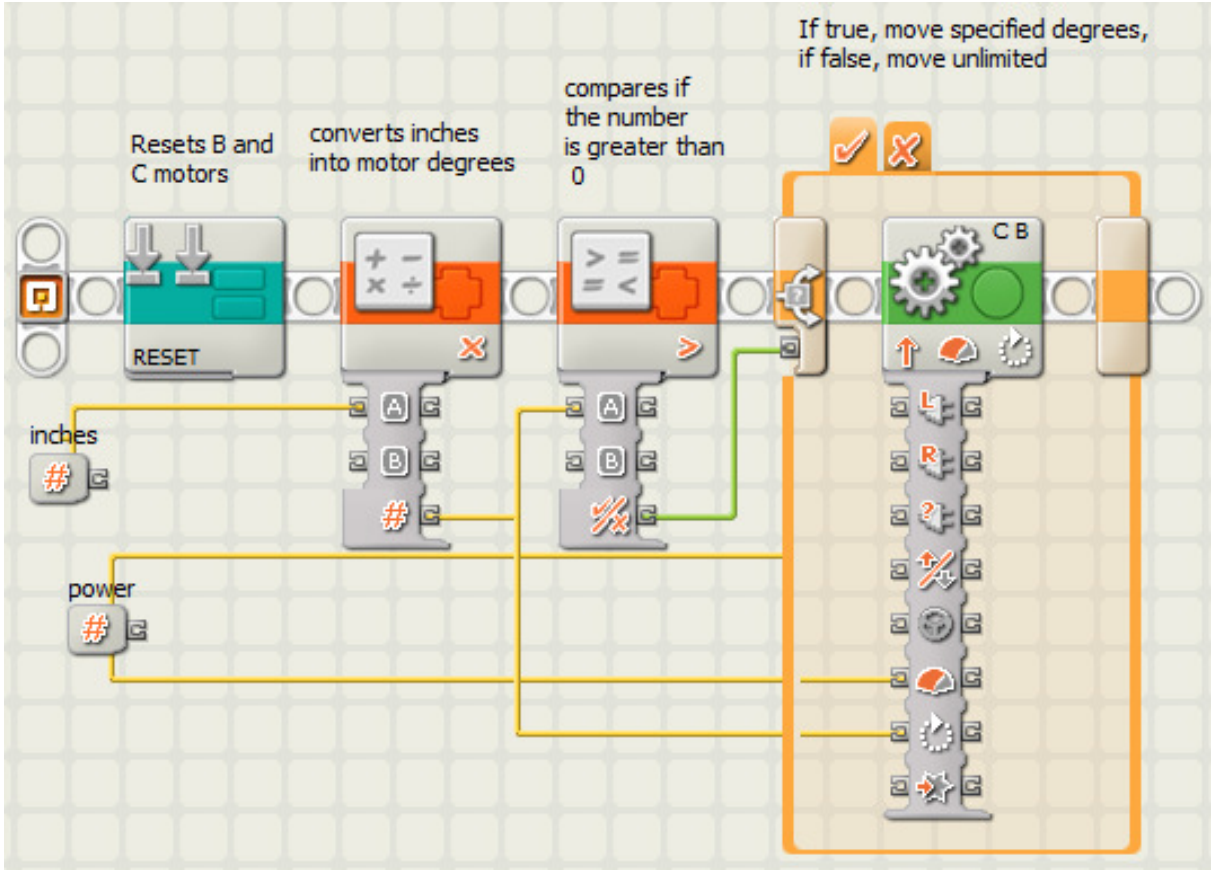


Command Interpreter Program (My Block)



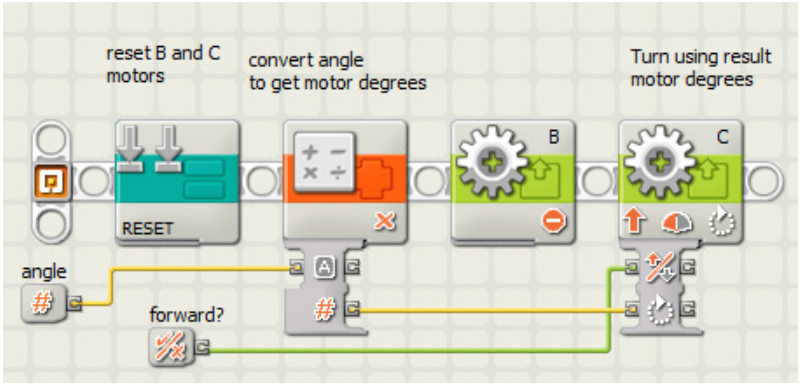
Move Forward (MoveFWD) and Move Back (MoveBCK)

1. The standard MOVE block is used with these features
2. The duration input is Inches which is converted to motor degrees
3. To accommodate unlimited move, the Inches variable would be set to zero. Using unlimited, the robot can move and do other action such as raising or lowering the arm, waiting for black line, etc.



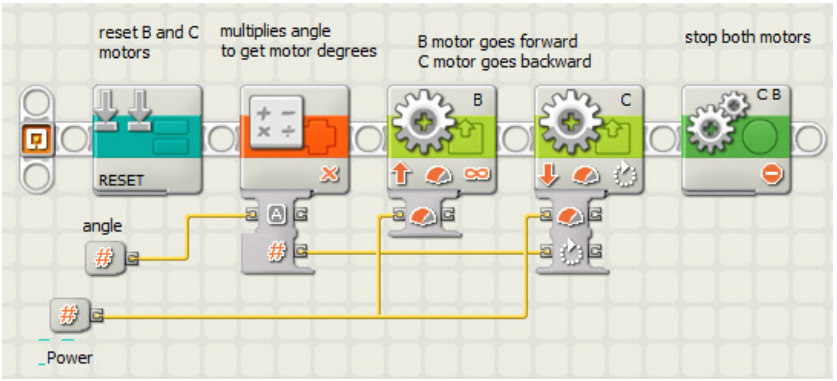
Turn Left / Right

1. Turn angle is converted into motor degrees using the relationship between turn circle and wheel circumference
2. Direction allows the motor to move forward or back



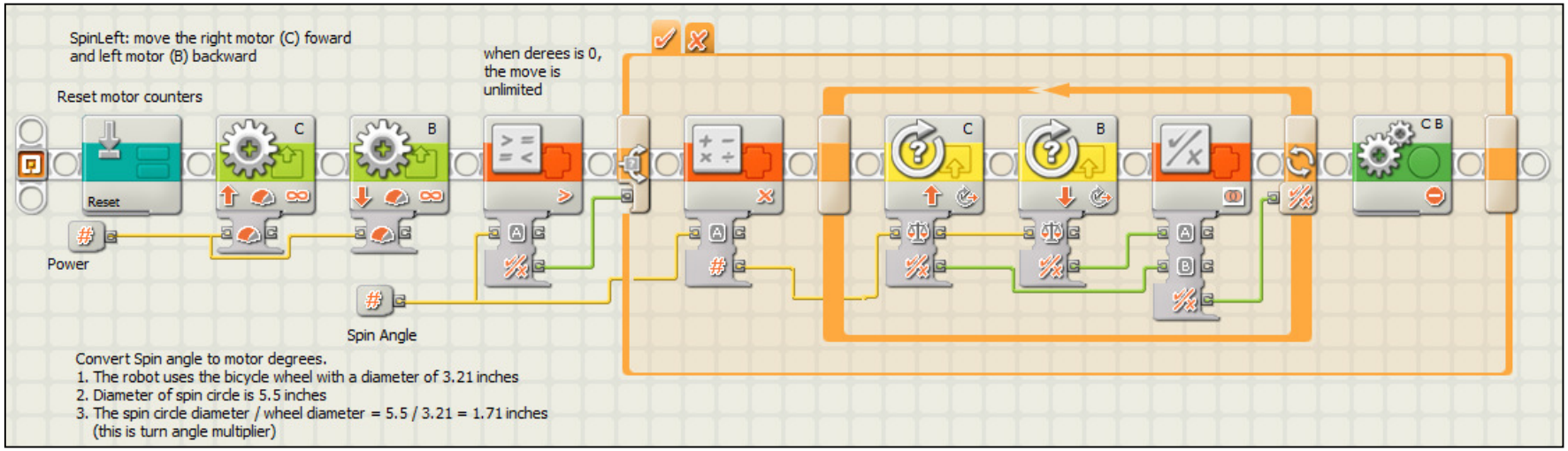
Spin Right / Left

1. Spin angle is converted into motor degrees by multiplying the angle by relationship between spin circle and wheel circumference
2. When the Spin Angle is zero, the spin is unlimited



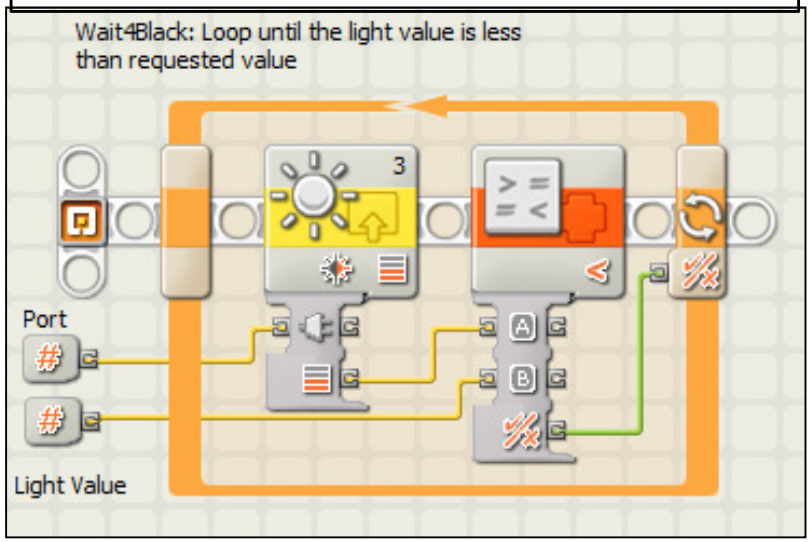
Spin Left / Right

1. Spin angle is converted into motor degrees by multiplying the angle by 1.71
2. When the Spin Angle is zero, the spin is unlimited

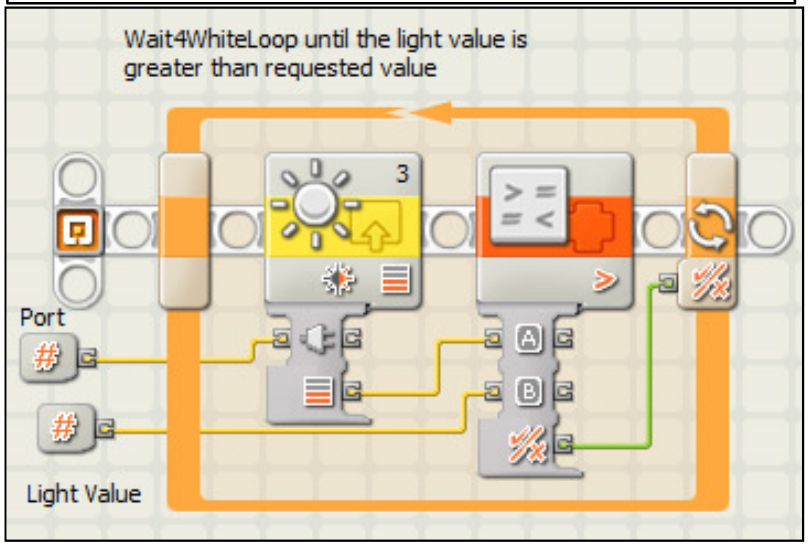


Wait2Black, Wait4White, Wait4Rotation

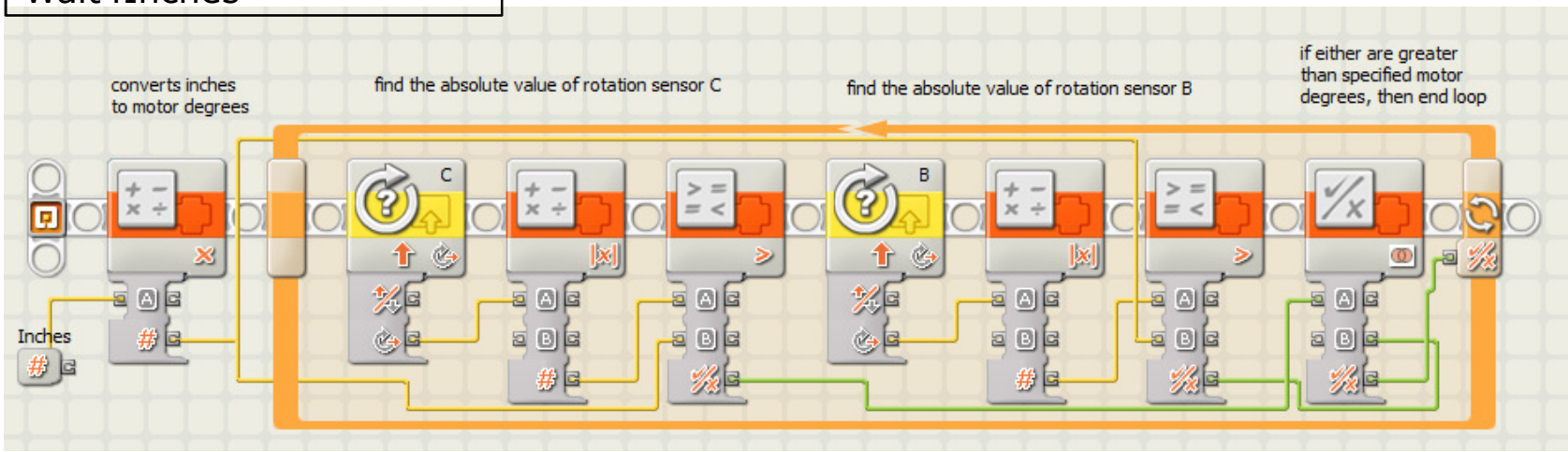
Wait4Black



Wait4White

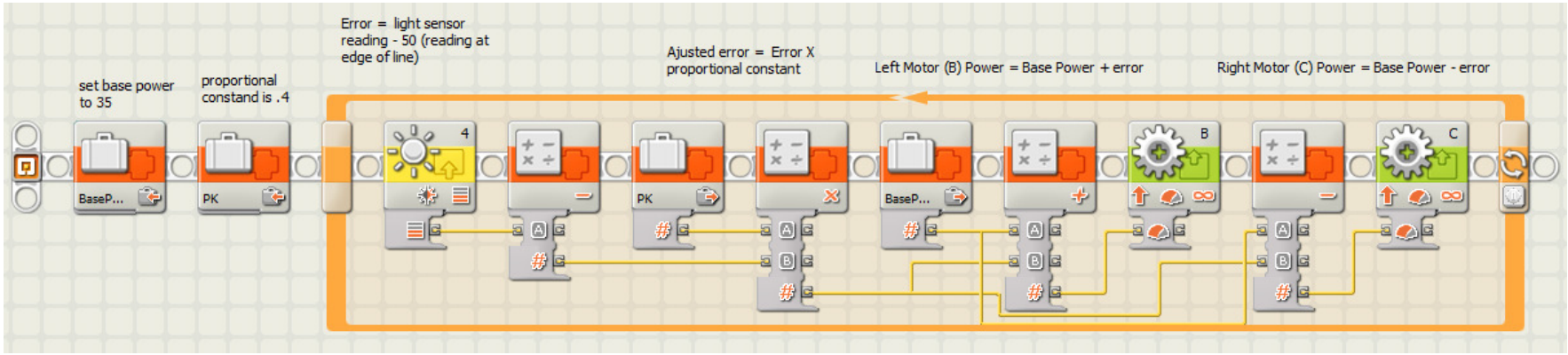


Wait4Inches

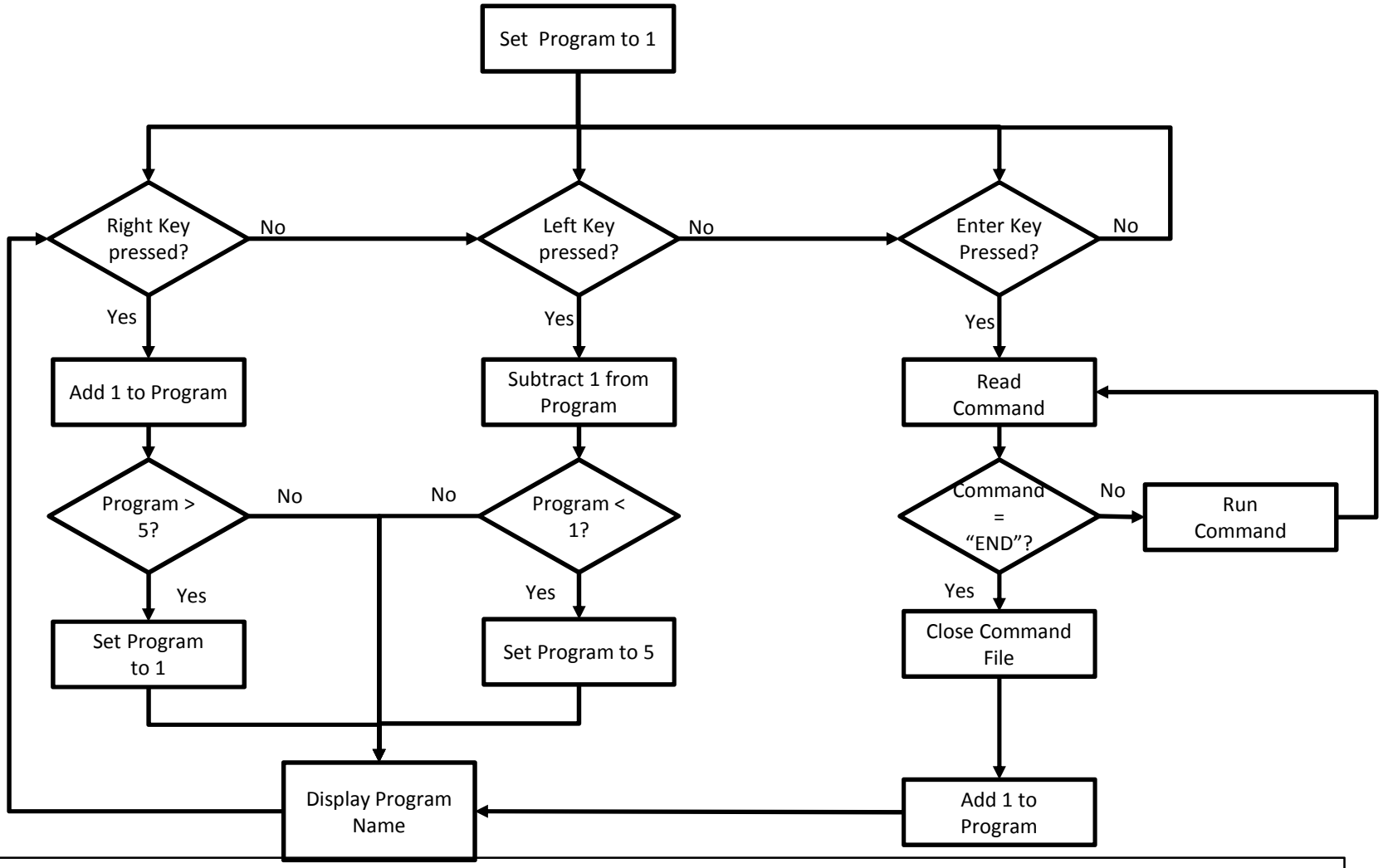


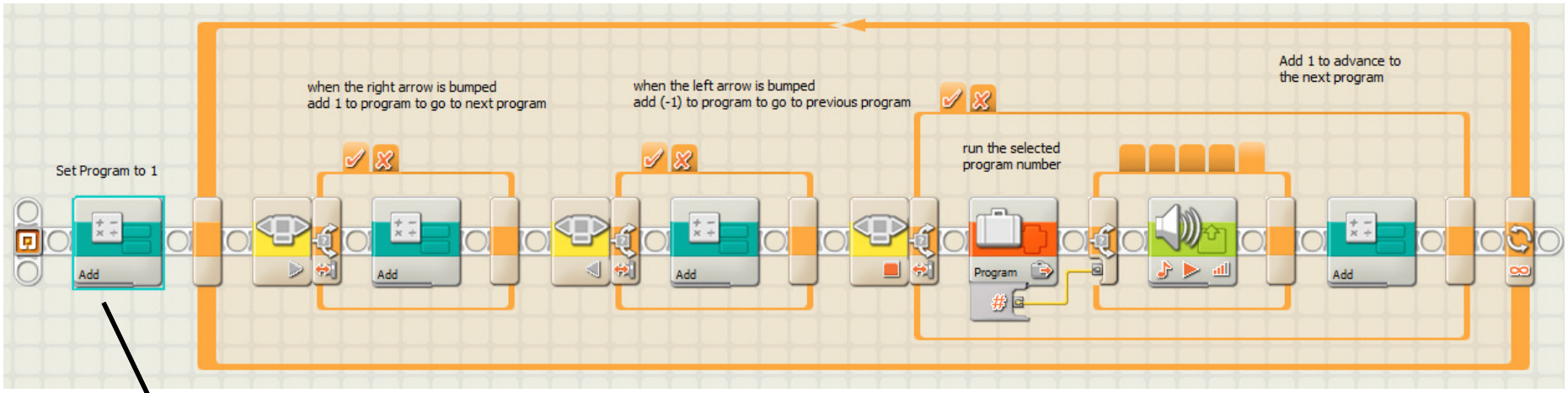
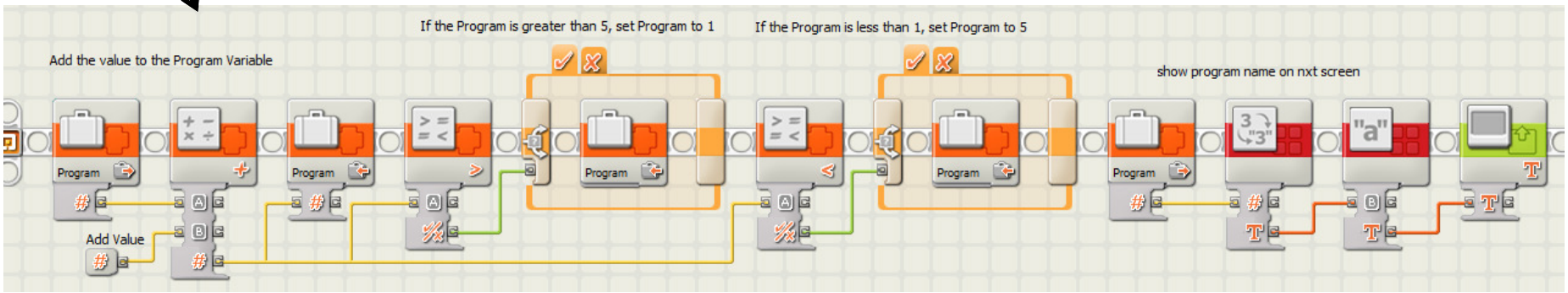
Proportional Line Follower – to avoid zig-zagging

- Use a single light sensor to follow the line where the turn varies and is proportional to how far away the robot is from the line.
- Assumptions:
  - At the edge of the line, the light sensor reading will be 50 which is the **ideal** reading
  - The **Base power** for both motors is 35
  - The difference between what the light sensor sees and **ideal** reading is the **Error**
  - The error is adjusted down using a Proportional Constant (Kp) of .4
  - Left edge of the line is followed
- Example 1: light sensor reading is 90 (robot drifted too far to the left of the line)
  - Error = light sensor reading – ideal reading = 90 – 50 = 40
  - Adjusted Error = Error X Proportional Constant = 40 X .4 = 16
  - Left Motor Power = base power + error = 35 + 16 = 51
  - Right Motor Power = base power – error = 35 – 16 = 19



Master Program Flowchart



The master program is used to automatically advance to the next program to minimize the required scrolling through to find the next program that needs to be run.

Master Program