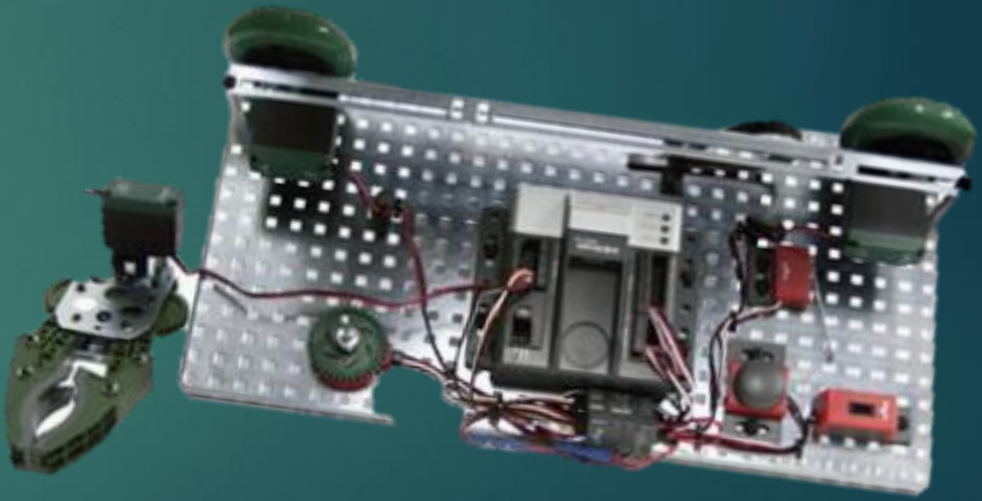
GTT Test Bed

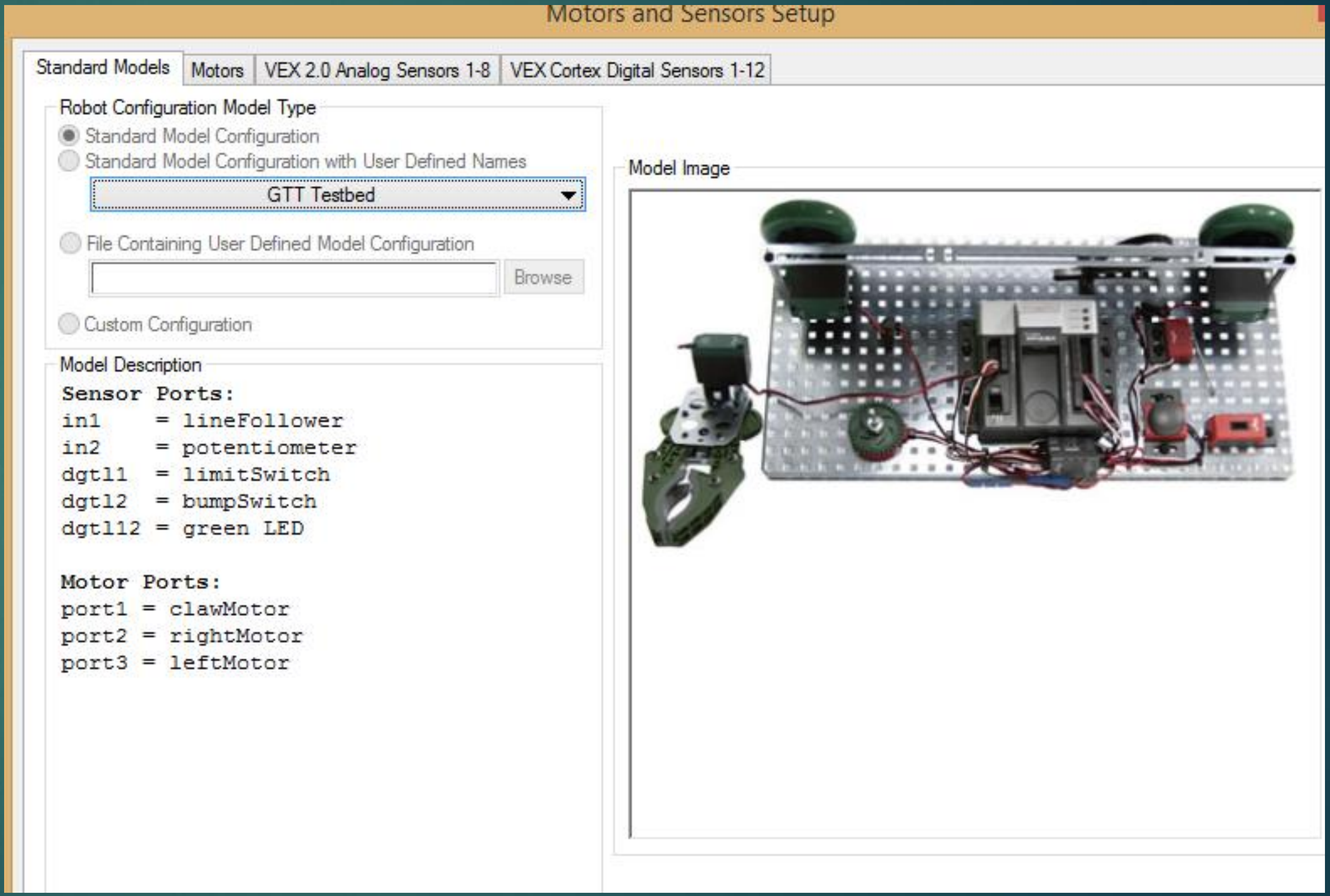
SIMPLE STEPS TO ROBOTC PROGRAMMING



Test Bed Motor and Sensor Set up

- in1 = lineFollower
- in2 = potentiometer
- dgtl1 = limitSwitch
- dgtl2 = bumpSwitch
- dgtl12 = green LED

- **Motor Ports:**
- port1 = clawMotor
- port2 = rightMotor
- port3 = leftMotor


Motors and Sensors Setup
Standard Models
Motors
VEX 2.0 Analog Sensors 1-8
VEX Cortex Digital Sensors 1-12
Robot Configuration Model Type
Browse
Model Image

● Standard Model Configuration
○ Standard Model Configuration with User Defined Names
GTT Testbed
○ File Containing User Defined Model Configuration
○ Custom Configuration

Model Description

```
Sensor Ports:
in1    = lineFollower
in2    = potentiometer
dgtl1  = limitSwitch
dgtl2  = bumpSwitch
dgtl12 = green LED

Motor Ports:
port1 = clawMotor
port2 = rightMotor
port3 = leftMotor
```

Start coding

- Using the PLTWTemplate.c
  - Create a program that runs a motor at ½ speed for 5 seconds, then stops for 5 seconds, then runs in reverse at ½ speed for 5 seconds, then stops
  - Fill in the heading information
  - Copy and Paste this into the Task Description
  - Type in Pseudocode
  - Create code in "task main ()"
- Once you have Compiled the program and fixed any errors-
  - Save the file as "halfSpeedmotor" in your H:/ drive
- Download the code to the Cortex and test it

Continue the Code

- Using the PLTWTemplate.c
  - Create a program that runs a motor at ½ speed when a bumper switch is pressed, then stops when a bumper switch is pressed, then runs in reverse at ½ speed when a bumper switch is pressed, then stops when a bumper switch is pressed
  - Fill in the heading information
  - Copy and Paste this into the Task Description
  - Type in Pseudocode
  - Create code in "task main ()"
  - Refer to the "halfSpeedmotor" code you created in the last activity (some of that code can be copy and pasted into your new program
- Once you have Compiled the program and fixed any errors-
  - Save the file as "motorWithbumper" in your H:/ drive
- Download the code to the Cortex and test it

Continue the Code

- Open your file named "motorWithbumper.c"
  - Add to the program
    - Make the LED flash on 1 second intervals 5 times simultaneously with the start of the right motor when the bumper switch is pressed the first time, then also flashes on 1 second intervals 5 times when the bumper switch is pressed the 3rd time to restart the right motor in reverse
  - Copy and Paste this onto the end of the Task Description
  - Amend the Pseudocode
  - Edit the code in "task main ()"
- Once you have Compiled the program and fixed any errors-
  - Save the file as "motorLEDflash" in your H:/ drive
- Download the code to the Cortex and test it

Continue the Code

- Open your file named “motorLEDflash.c”
  - Add to the program
    - Make the Left Motor turn on and run at ½ speed forward simultaneously with the final flash of the LED after the bump switch is pressed the first time. Make it stop with the press of the bump switch along with the Right Motor. Then make it start again, in reverse, at ½ speed simultaneously with the final flash of the LED after the bump switch is pressed the third time. Finally, make the Left Motor stop with the final press of the bump switch along with the Right Motor.
  - Copy and Paste this onto the end of the Task Description
  - Amend the Pseudocode
  - Edit the code in “task main ()”
- Once you have Compiled the program and fixed any errors-
  - Save the file as “dualmotorsLEDflash” in your H:/ drive
- Download the code to the Cortex and test it