

Instructions for FBWwinAppDemo

FBWwinAppDemo is a Visual Studio 2005 solution. It was developed to illustrate the programming techniques described in the book *A Fly-By-Wire Architecture for Multi-Threaded Windows Apps*, by Will Warner. The book is available on Amazon.com. The solution converts effortlessly into VS2008 and VS2010.

The FBWwinAppDemo Visual Studio solution contains two projects. One project—FBWwinAppInfrastructure—produces a DLL that furnishes the fly-by-wire infrastructure. The other project—RobotController—includes the forms and additional classes that make use of the infrastructure to achieve the functionality required of the robot controller.

You can build the executable and run it (...\\WinAppDemo\\bin\\FBWwinAppDemo.exe), or run the program in the Visual Studio debugger (double click on FBWwinAppDemo.sln to load the solution into Visual Studio).

Either way you run it, the first thing you see is the form that serves as the control panel of the robot controller (Figure 1).

Once the controller is started, text boxes arrayed across the top of the form will display the intensity of radiation detected by the eight IR sensors. Below these, the form will indicate the direction of the IR it is sensing—is it coming mostly from the left-front, right-front, or straight ahead?

Along the left side, groups of controls will display the status of the left wheels, front and rear. A similar arrangement of controls on the right will display the status of the right wheels.

The progress bar at the bottom will show progress of self-testing and setup-file downloading.

A click on the *Close* button will close the program.

A click the *Debug* button will launch the debug screen.

The buttons down the middle initiate self-test, setup (configuration), and the starting and stopping of active control of the robot.

Getting Started

If there were an actual robot connected to your PC, the robot controller program—*Moth Control*—would communicate with it and control it. Because there is no real robot, however, you have to begin by launching a form that will simulate the robot. That form responds to the robot controller's commands as a real robot would.

Figure 1: Main form of the robot control program. This form serves as the control panel for the robot controller.

The screenshot shows a software window titled "MOTH Control". It contains several sections for controlling a robot:

- IR Readings:** A section at the top with eight input fields labeled IR-L4, IR-L3, IR-L2, IR-L1, IR-R1, IR-R2, IR-R3, and IR-R4. Below these fields is a label "Direction of IR: Unknown".
- Left Front Wheel:** A section with "Self-test:" and "Speed:" input fields, a "Self-test" button, and a "Setup" button.
- Right Front Wheel:** A section with "Self-test:" and "Speed:" input fields, a "Self-test" button, and a "Setup" button.
- Left Rear Wheel:** A section with "Self-test:" and "Speed:" input fields, a "Self-test" button, and a "Setup" button.
- Right Rear Wheel:** A section with "Self-test:" and "Speed:" input fields, a "Self-test" button, and a "Setup" button.
- Start:** A central button labeled "Start".
- Status Bar:** At the bottom, there is a "Close" button, a progress bar labeled "Idle" showing 0%, and a "Debug" button.

To launch the robot simulator form, first launch the debug form by clicking the *Debug* button on the main form. On the *Debug Support* form that appears (Figure 2), click *Show Additional Debug* and you will see the *Simulate Moth* form (Figure 3).

Once the that form is showing, you can return to the main form and begin the steps to self-test, setup, and start control of the (simulated) robot. The *Start* button on the control panel becomes enabled only after the robot has passed self-test and been configured.

Figure 2: *Debug Support* form. From this form you launch the simulated robot, view message traffic, and launch forms that monitor the state and activity of Engines (threads that respond to messages) within the program.

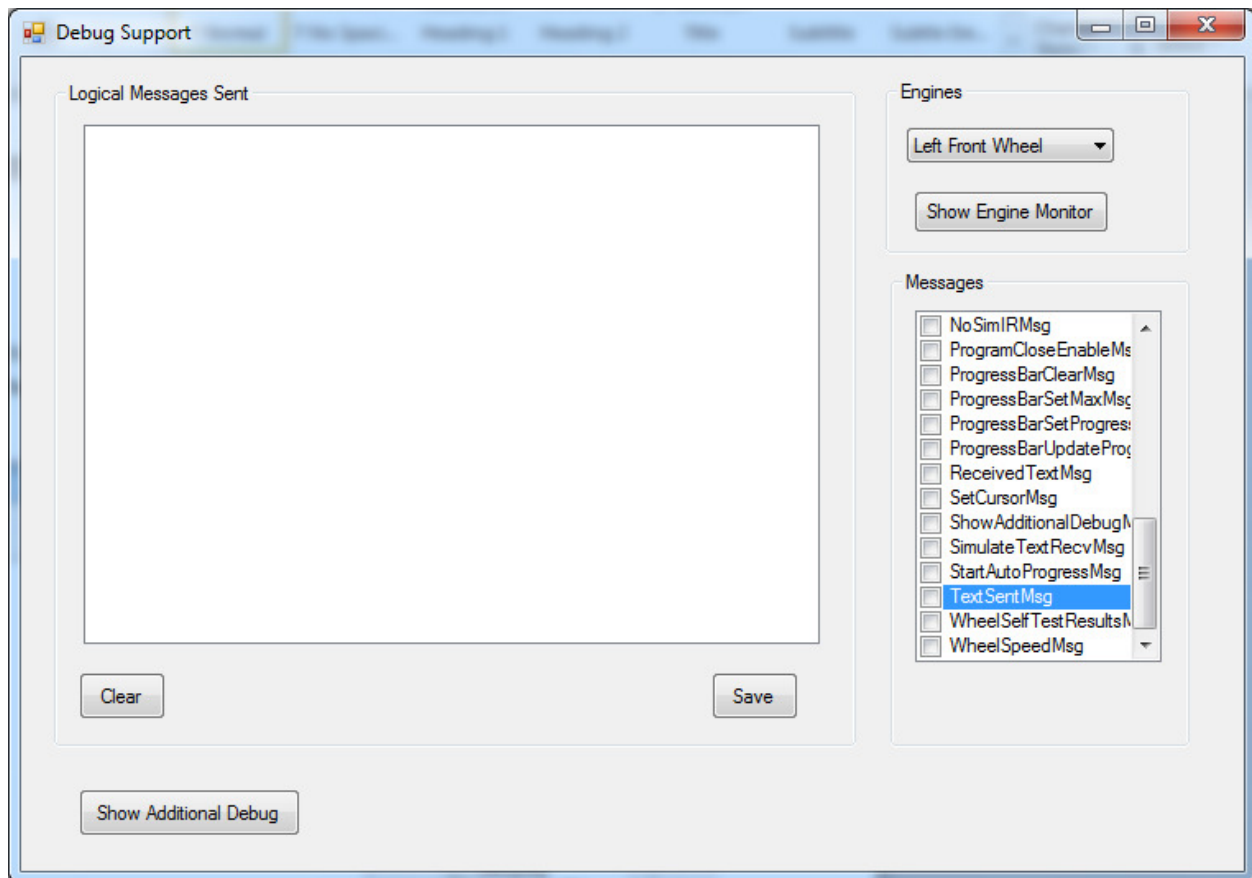


Figure 3: Robot simulation form, *Simulate MOTH*. This is its initial appearance before robot control has started. The form must be open for the robot control program to complete the self-test and setup operations, because code in this form responds to commands as a real robot would.

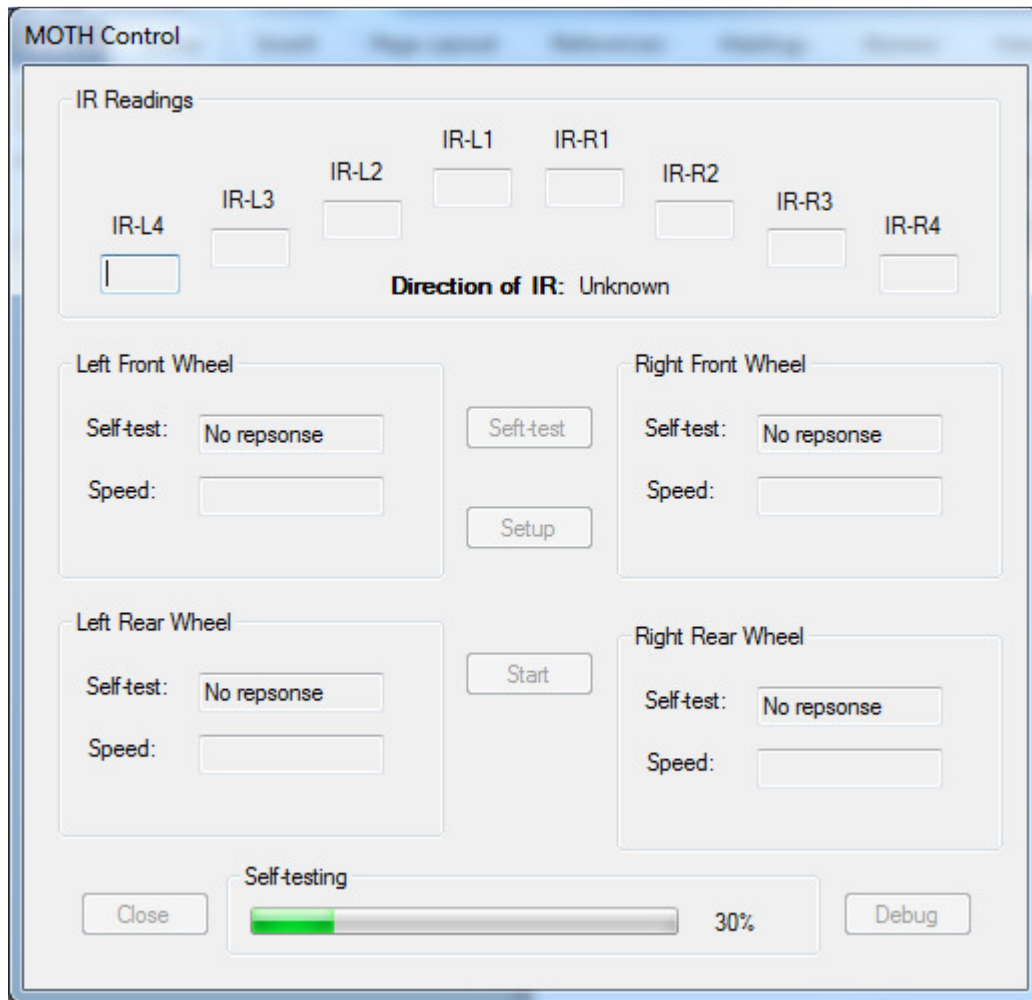
The screenshot shows a window titled "Simulate MOTH" with a standard Windows-style title bar. Inside the window, there is a large, light gray rectangular area. In the top right corner of this area, there is a small blue button labeled "No Inv". Below this button, there is a vertical scrollbar with a stick figure icon at the top. In the lower-left portion of the window, there is a smaller, gray-bordered box titled "Simulated MOTH". This box contains a form with several input fields arranged in two columns. The left column has four fields labeled LR, LF, RR, and RF. The right column has eight fields labeled IRL4, IRL3, IRL2, IRL1, IRR1, IRR2, IRR3, and IRR4. All fields are currently empty.

LR	LF	IRL4
		IRL3
		IRL2
		IRL1
		IRR1
		IRR2
		IRR3
RR	RF	IRR4

Self-test

When you click the *Self-test* button, the program sends self-test commands to all the wheel controllers. Figure 4 depicts the appearance of the control panel while self-testing is in progress.

Figure 4: Robot control panel while self-testing underway.



Setup (Configuration)

After self-testing, the next step is to configure control of the wheels by sending the wheel controllers the contents of a setup file. Click on *Setup* and the program displays a dialog for selecting a setup file used to configure the wheel controllers (Figure 5). Choose *MothSetup.txt* and click OK.

The dialog will close and the program will download the contents of the file you selected to the four wheel controllers, one by one. Figure 6 shows the appearance of the control panel while downloading to the left-rear wheel controller. Figure 7 shows the appearance of the simulated robot after the left-

front wheel controller has been configured, and while the left-rear wheeler controller is in the process of being configured.

Start

When configuration of all wheel controllers completes, the control panel will enable its *Start* button (Figure 8). Click it to begin operation. Once operation has begun, the start-button is relabeled *Stop*. While actively controlling the robot, the *Close* button is disabled; it is re-enabled when you stop active control of the robot by clicking the *Stop* button.

Figure 9 depicts the control panel after starting but before the simulated robot form has begun to simulate infrared radiation (IR). The control panel shows 0 IR and direction none.

Go to the *Simulated Moth* form. Click on Irv to start the simulation of IR. Irv turns red and the robot avatar begins walking toward Irv (Figure 10). Use the slide bar to move Irv to another location. The robot avatar will change course to follow Irv. Figure 11 shows the robot homing in on Irv after you have stopped moving him.

Figure 5: Dialog box for picking a setup file for configuring the wheel controllers.

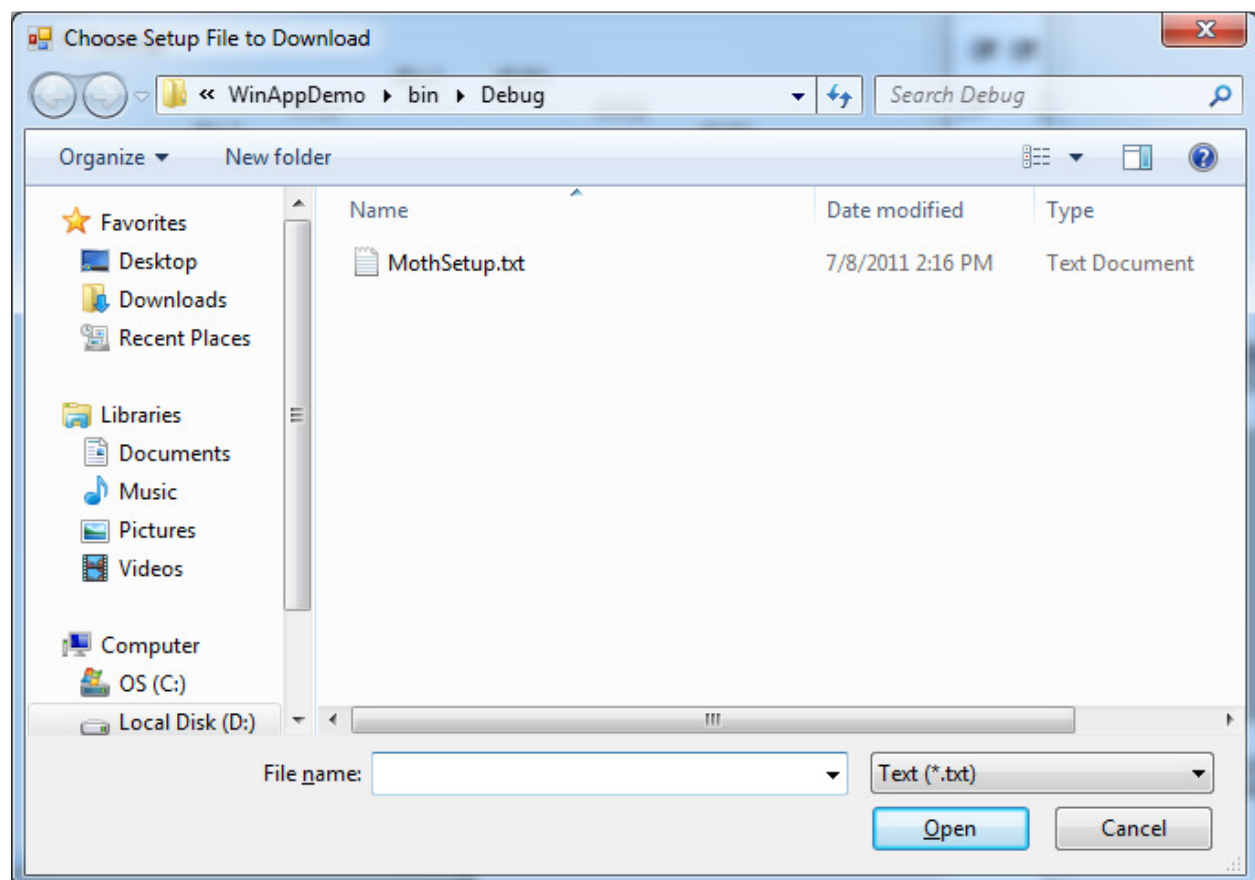


Figure 6: Appearance of the robot control panel while downloading configuration to the left-rear wheel controller.

The **MOTH Control** window displays the following components:

- IR Readings:** A grid of input fields for IR-L4, IR-L3, IR-L2, IR-L1, IR-R1, IR-R2, IR-R3, and IR-R4. Below this grid, the text "Direction of IR: Unknown" is displayed.
- Left Front Wheel:** Includes "Self-test: OK" and "Speed:" input fields, with "Self-test" and "Setup" buttons.
- Right Front Wheel:** Includes "Self-test: OK" and "Speed:" input fields, with a "Self-test" button.
- Left Rear Wheel:** Includes "Self-test: OK" and "Speed:" input fields, with a "Start" button.
- Right Rear Wheel:** Includes "Self-test: OK" and "Speed:" input fields.
- Bottom Section:** A "Close" button, a progress bar labeled "Downloading to LR wheel" at 90%, and a "Debug" button.

Figure 7: Appearance of the *Simulate MOTH* form while downloading configuration to the left-rear wheel controller, after having downloaded configuration to the left-front wheel controller.

The **Simulate MOTH** window displays a simulated robot environment with the following elements:

- Top Right:** A "No Inv" button and a stick figure icon.
- Simulated MOTH Panel:**
 - LR:** PARAM4=100
 - LF:** PARAM9=Low
 - IR Sensors:** IRL4, IRL3, IRL2, IRL1, IRR1, IRR2, IRR3, and IRR4, each with an associated input field.
 - RR:** Input field
 - RF:** Input field

Figure 8: The control panel form, after self-test and setup, ready to start controlling the robot.

The image shows a software window titled "MOTH Control". It contains several sections for controlling a robot:

- IR Readings:** A section at the top with eight input fields labeled IR-L4, IR-L3, IR-L2, IR-L1, IR-R1, IR-R2, IR-R3, and IR-R4. Below these fields is the text "Direction of IR: Unknown".
- Left Front Wheel:** A section on the left with "Self-test:" (showing "OK") and "Speed:" (empty) fields, and a "Self-test" button.
- Right Front Wheel:** A section on the right with "Self-test:" (showing "OK") and "Speed:" (empty) fields, and a "Self-test" button.
- Left Rear Wheel:** A section on the left with "Self-test:" (showing "OK") and "Speed:" (empty) fields, and a "Self-test" button.
- Right Rear Wheel:** A section on the right with "Self-test:" (showing "OK") and "Speed:" (empty) fields, and a "Self-test" button.
- Start Button:** A central button labeled "Start".
- Setup Button:** A button labeled "Setup" located between the front wheel sections.
- Close Button:** A button labeled "Close" at the bottom left.
- Idle Slider:** A slider bar at the bottom with the label "Idle" and a value of "0%".
- Debug Button:** A button labeled "Debug" at the bottom right.

Figure 9: Control panel after control started, but before IR simulation has begun.

The screenshot displays the 'MOTH Control' application window. At the top, a menu bar includes 'File', 'View', 'Help', 'About', 'Settings', 'Tools', and 'Exit'. The main interface is divided into several sections:

- IR Readings:** A section containing eight numerical input fields for IR sensors: IR-L4 (0), IR-L3 (0), IR-L2 (0), IR-L1 (0), IR-R1 (0), IR-R2 (0), IR-R3 (0), and IR-R4 (0). Below these fields, the text 'Direction of IR: None' is displayed.
- Left Front Wheel:** A control panel with 'Self-test' (set to 'OK') and 'Speed' (set to 'Stop') dropdown menus, and 'Self-test' and 'Setup' buttons.
- Right Front Wheel:** A control panel with 'Self-test' (set to 'OK') and 'Speed' (set to 'Stop') dropdown menus, and a 'Self-test' button.
- Left Rear Wheel:** A control panel with 'Self-test' (set to 'OK') and 'Speed' (set to 'Stop') dropdown menus, and a 'Stop' button.
- Right Rear Wheel:** A control panel with 'Self-test' (set to 'OK') and 'Speed' (set to 'Stop') dropdown menus.
- Idle:** A horizontal slider at the bottom, currently positioned at 0%, with 'Close' and 'Debug' buttons on either side.

Figure 10: The robot avatar tracking toward Irv, the source of simulated infrared radiation.

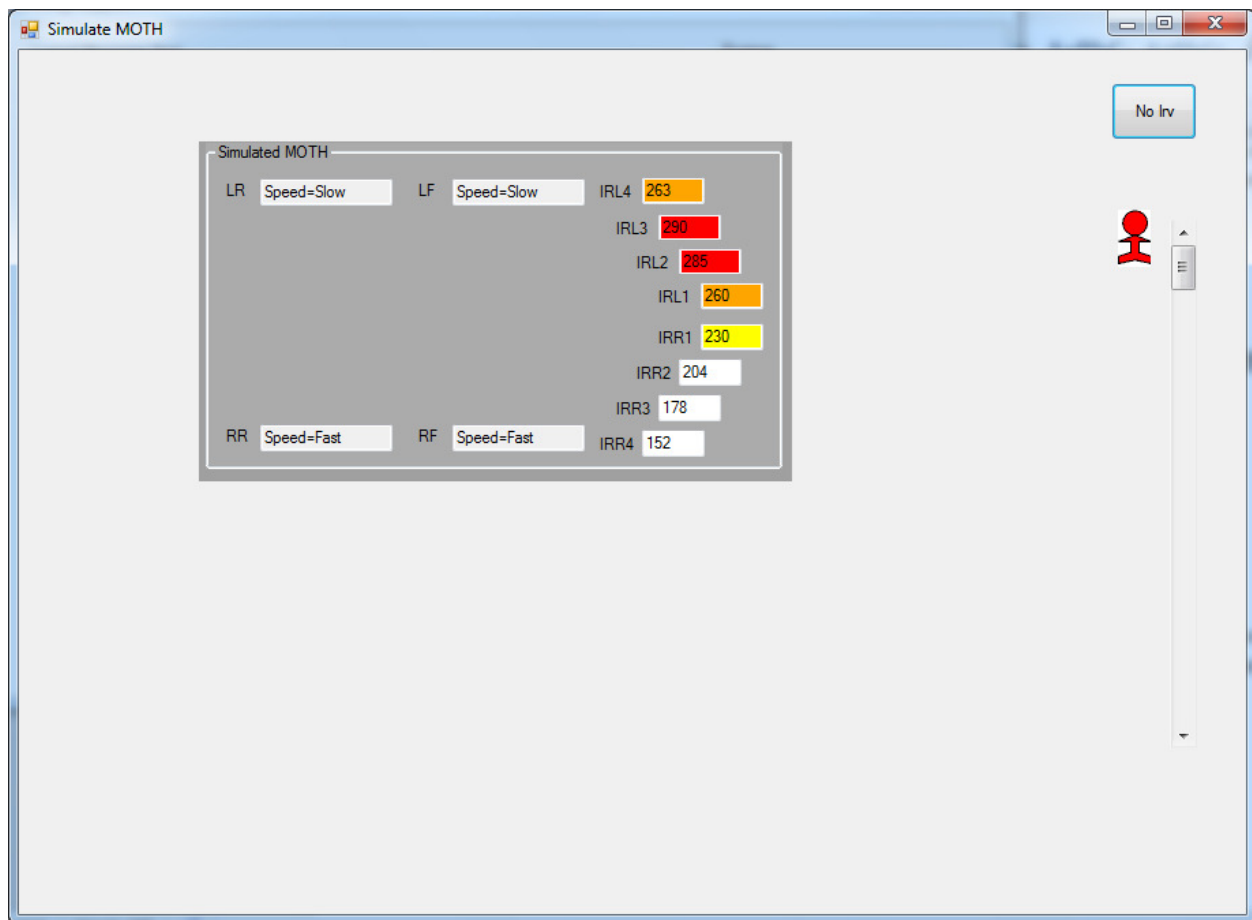
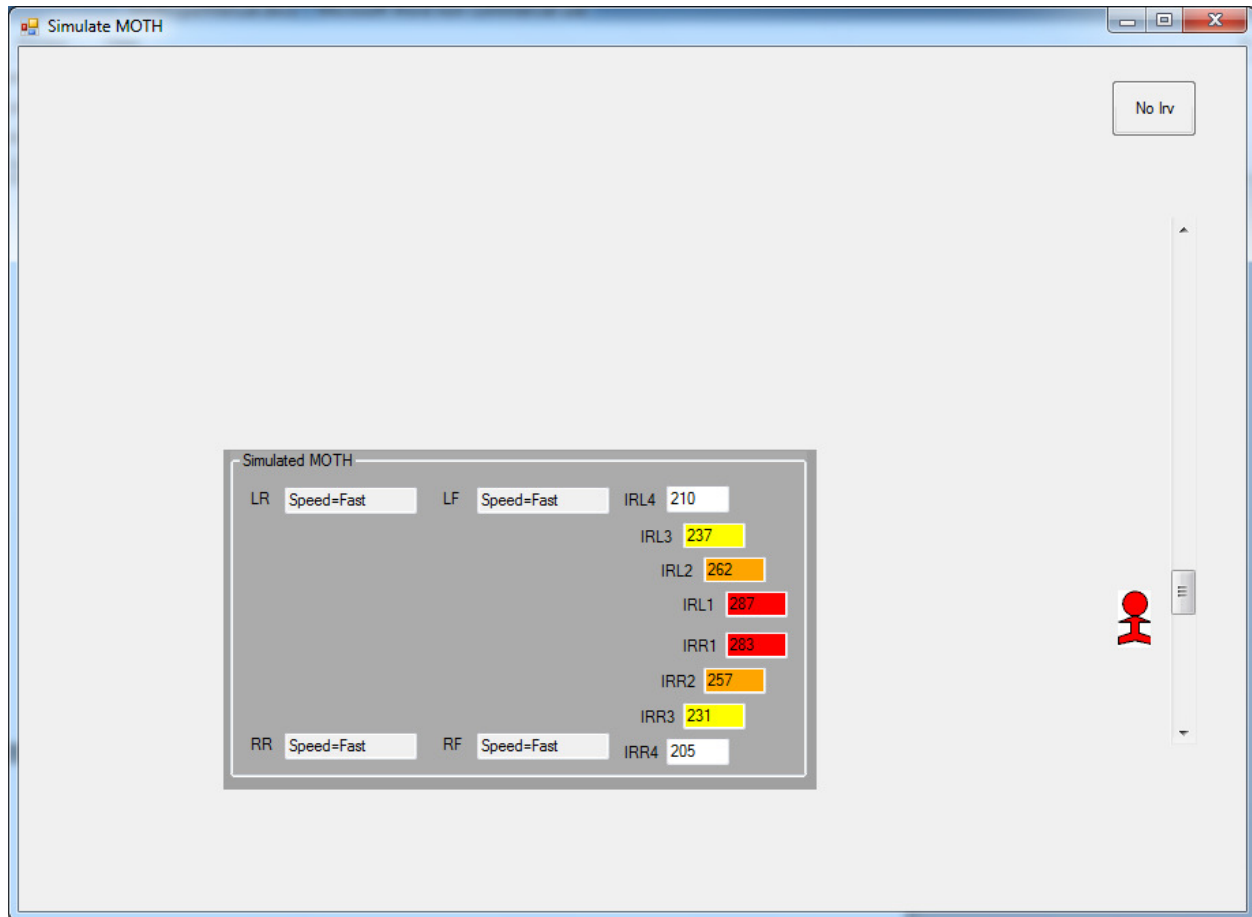


Figure 11: The robot homing in on Irv.



Debug Support

The *Debug Support* form (Figure 12) can aid in debugging. That form functions as a “sniffer” on the logical bus. On the right, the form lists the names of all the logical messages defined in the program. A programmer can select any number of the messages names. Each time a selected message appears on the bus, the form displays it and its payload.

The *Debug Support* form also enables a programmer to see what’s going on with engines and jobs. The form presents a list of all the engines defined by the program (top right). Select one, say, Left Front Wheel, and click the *Show Engine Monitor* button. The form in Figure 13 will appear. The form displays the status of the engine and the job it is running, if any. The one in Figure 13 shows that the wheel engine for the left front wheel executed a Self-Test Wheel job, the job was successful, and the engine became idle.

Figure 12: Debug Support form showing communication with wheel controllers during download of setup files.

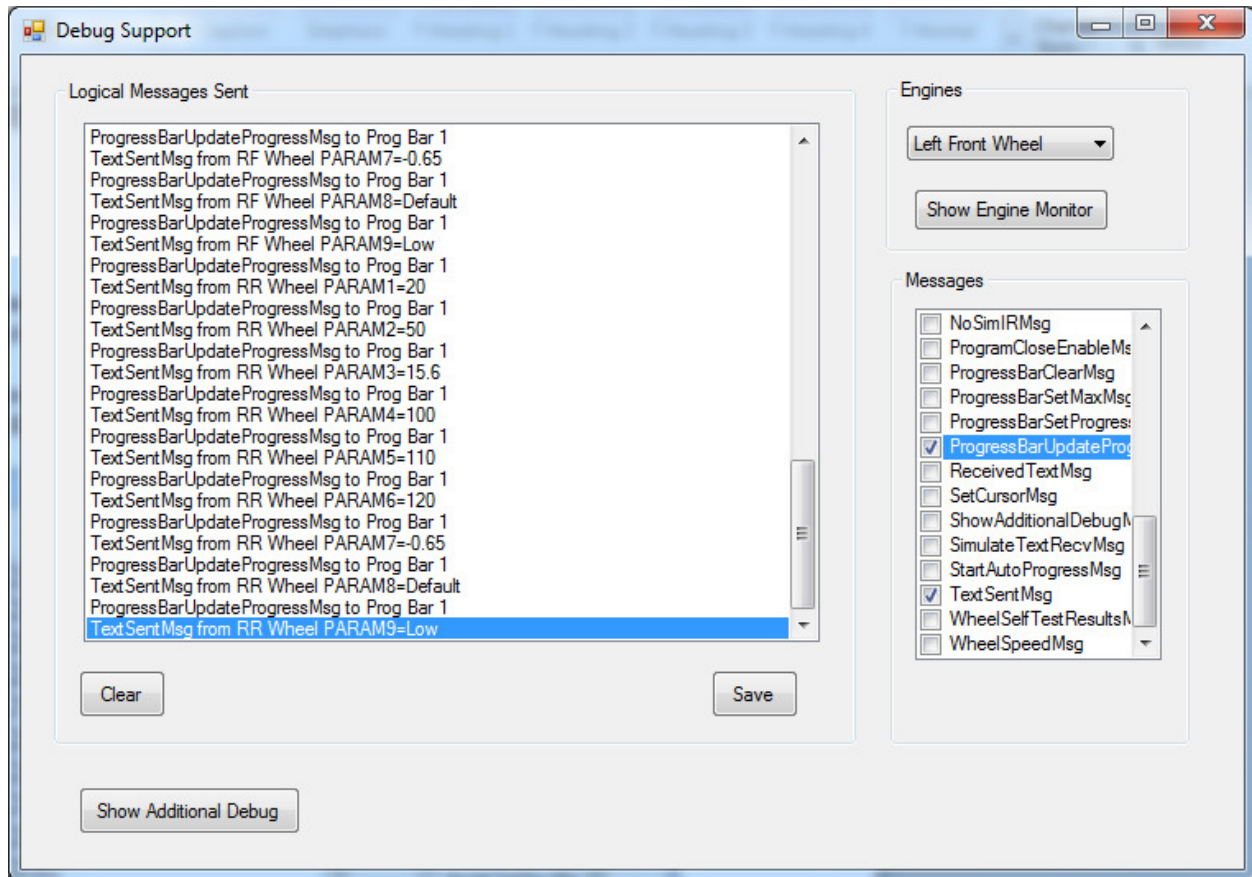


Figure 13: Engine Monitor showing status of the wheel engine for the left front wheel during self-testing on that wheel.

