Interfacing iRobot Create with LabVIEW, LabVIEW Real-Time, and LabVIEW Embedded Module for ARM Microcontrollers

Introduction

The iRobot® Create® [1] is a popular consumer robotics platform based on the Roomba® autonomous vacuum cleaner. The robot is popular because of its sensors (four infrared "cliff" sensors, an infrared wall sensor, two bump sensors, three wheel-drop sensors), and its simple UART serial interface [2]. Combined with National Instruments hardware or software, the iRobot Create is often used as an educational robotics platform [3].

Supported Targets

The simple serial interface to the iRobot Create allows the robot to be easily controlled from a desktop computer or an embedded controller. Targets compatible and tested include:

- LabVIEW (Windows / Mac / Linux)
- LabVIEW Real-Time 2011
- LabVIEW Embedded Module for ARM® Microcontrollers 2011

LabVIEW desktop and LabVIEW RT use NI-VISA to communicate to the iRobot via UART serial. LabVIEW Embedded for ARM Microcontrollers uses the Serial Compatability Palette.

Requirements

- iRobot Create (available from http://irobot.com)
- LabVIEW 2011
- NI-VISA 5.0.1 or later
- JKI VI Package Manager
- [optional] LabVIEW Real-Time 2011 (for embedded control using NI sbRIO or compactRIO)
- [optional] LabVIEW Embedded Module for ARM Microcontrollers 2011 (for embedded control using an ARM-based microcontroller)

Implementation

The iRobot Create transmits sensor packets and receives commands via UART serial interface. The communication interface is defined by the iRobot Create Open Interface (OI) [2], and has been implemented in its entirety here.

A simple way to communicate with the iRobot is to dedicate one loop to reading from the serial port, and another to writing. This works well with sensor streams, which after being enabled, need only be read from the serial port, allowing another loop to issue commands. Note that the iRobot Create will inaccurately report distance and angle when sensors are streamed; review the example code in the project for more detail.

Desktop Control of the iRobot Create

The iRobot Create ships with a serial cable that plugs into the 9-pin serial port on the iRobot and a standard DB-9 RS-232 serial connector on a desktop or laptop computer. Using LabVIEW and NI-VISA, commands are exchanged with the robot. You may also use the Rx/Tx pins on the DB-25 cargo bay connector and a USB serial adapter. Configure your application to use the COM port number associated with the serial port to which it is connected.

The robot may be controlled remotely by using the Element Direct® Bluetooth Adapter Module (BAM) for iRobot Create, or a third-party wireless UART devices such as Bluetooth® (we tested with the BlueSMiRF® Gold [4]) or XBee®.

Download the "iRobot Create for Desktop and Real-Time Targets" package below to get started.

Autonomous Control of the iRobot Create using sbRIO or cRIO

The onboard RS-232 port on a cRIO or sbRIO can be connected to the iRobot Create cargo bay connector. Using LabVIEW and NI-VISA, commands are exchanged with the robot.

Download the "iRobot Create for Desktop and Real-Time Targets" package below to get started.

Learn how to Connect iRobot Create to an NI single-board RIO (sbRIO).

Autonomous Control of the iRobot Create using an ARM-based microcontroller

LabVIEW Embedded Module for ARM Microcontrollers installs a Serial Compatability palette with VIs that provide a VISA-like serial interface to UART ports on an ARM-based microcontroller. The desktop project has been modified (trivially) to use the Serial Compatability VIs, but all remaining code remains the same.

Download the "iRobot Create for ARM Targets" package below to get started

Learn how to Connect iRobot Create to an LM3S8962 Microcontroller.

Sample Code

The simplest example polls sensors and displays them on the front panel (**Figure 1**). Note that clusters may not properly display on the front panel when debugging on ARM targets. The Poll Sensors VI will block until a sensor packet has been received.

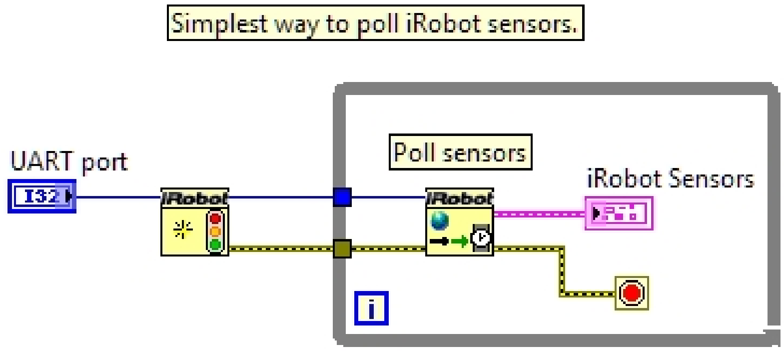

Figure 1: VI to initialize communication to the robot and poll its sensors.

The iRobot Create can be configured to periodically transmit sensor packets. The following code (**Figure 2**) initializes communication to the robot, begins a sensor stream, and processes the data as it arrives.

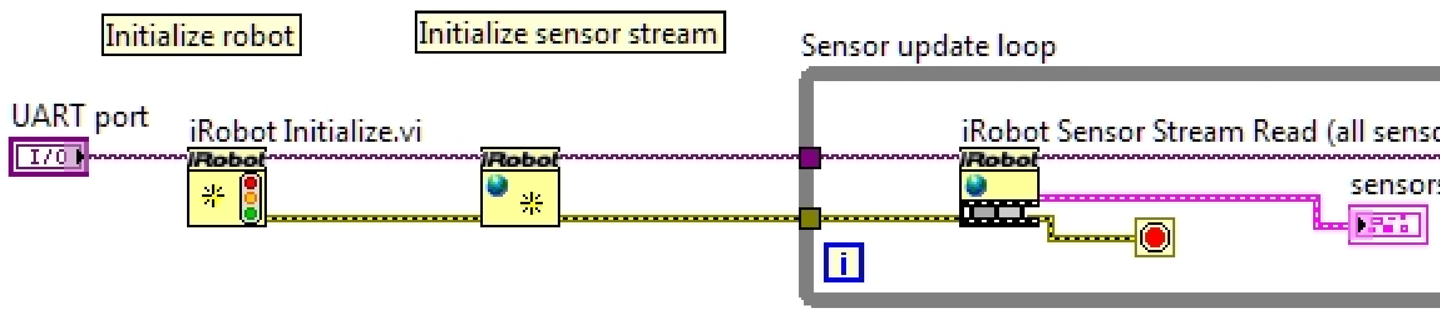

Figure 2: Vi to initialize communication to the robot, begin a sensor stream, and process incoming data.

A simple demo application uses a sequence to drive the robot in a square pattern by driving a fixed distance, turning 90 degrees, and repeating. This is included in the Demo folder of the project; a screenshot of the top-level VI is shown below in **Figures 3a-b**:

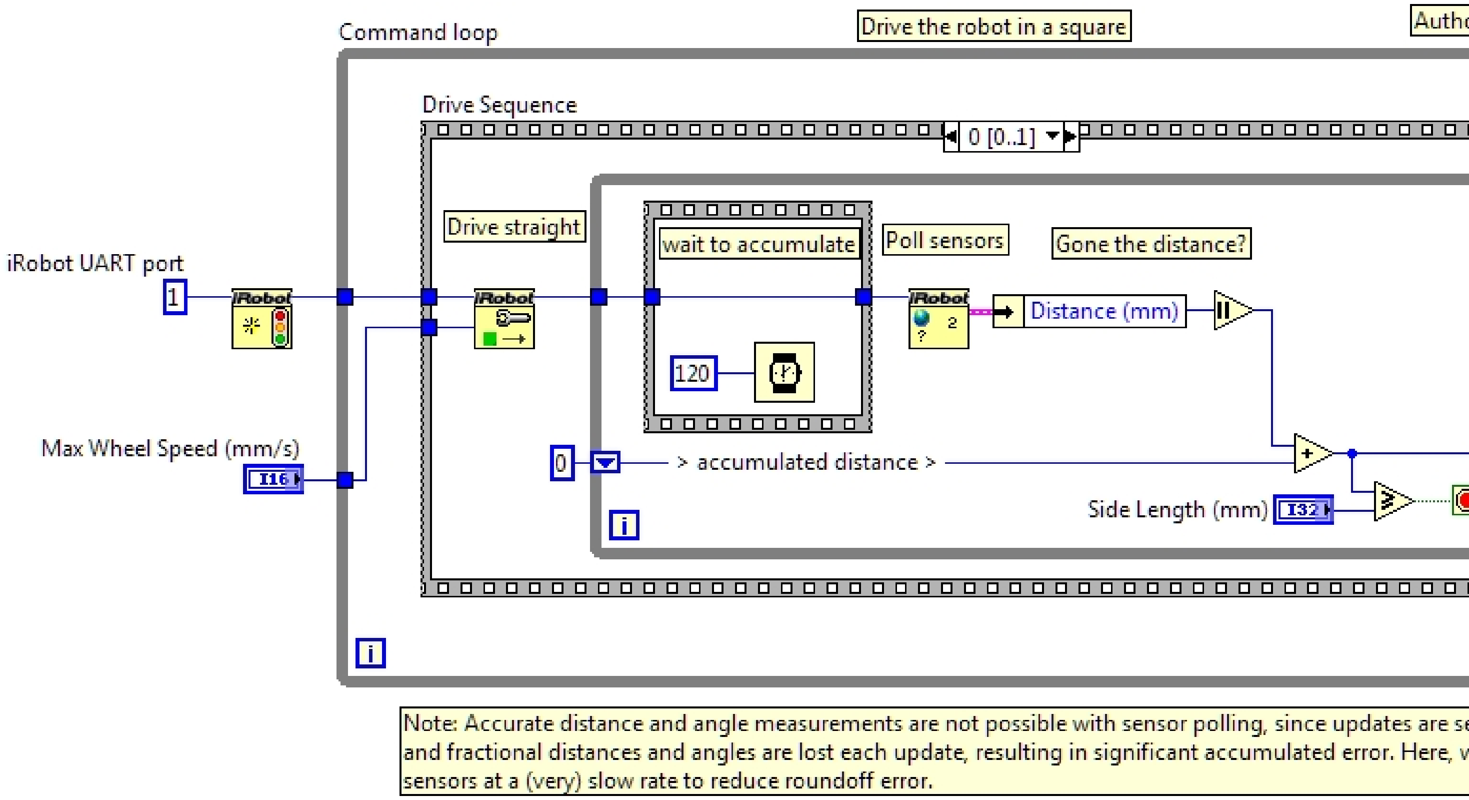

Figure 3a: VI to drive an iRobot Create in a square. The first element of the sequence, which instructs the robot to drive straight for a fixed distance, is shown.

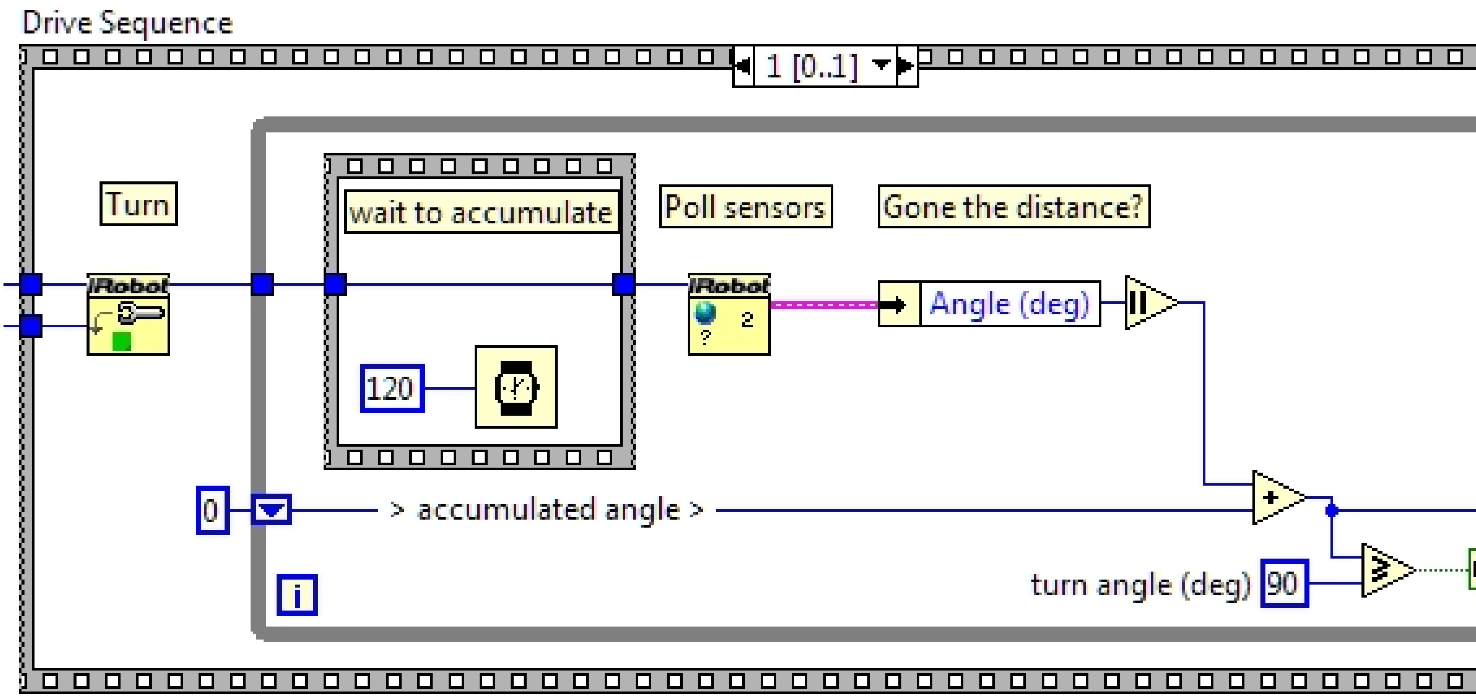

Figure 3b: Second element of the command sequence, which instructs the robot to turn through 90 degrees.

The application begins by initializing the serial port that will interface with the iRobot Create, and establishing a connection with the robot. The robot is instructed to drive until its reported distance is greater than the Side Length (mm) control, after which it will turn until the robot reports it has rotated through 90 degrees. The sequence restarts and the robot will continue to drive the shape of a square.

A second application demonstrates how to stream sensors from the iRobot and display them on the front panel, and how to use the front panel to drive or turn. A screenshot of the Sensor Stream Demo with Concurrency VI is shown below in **Figure 4**:

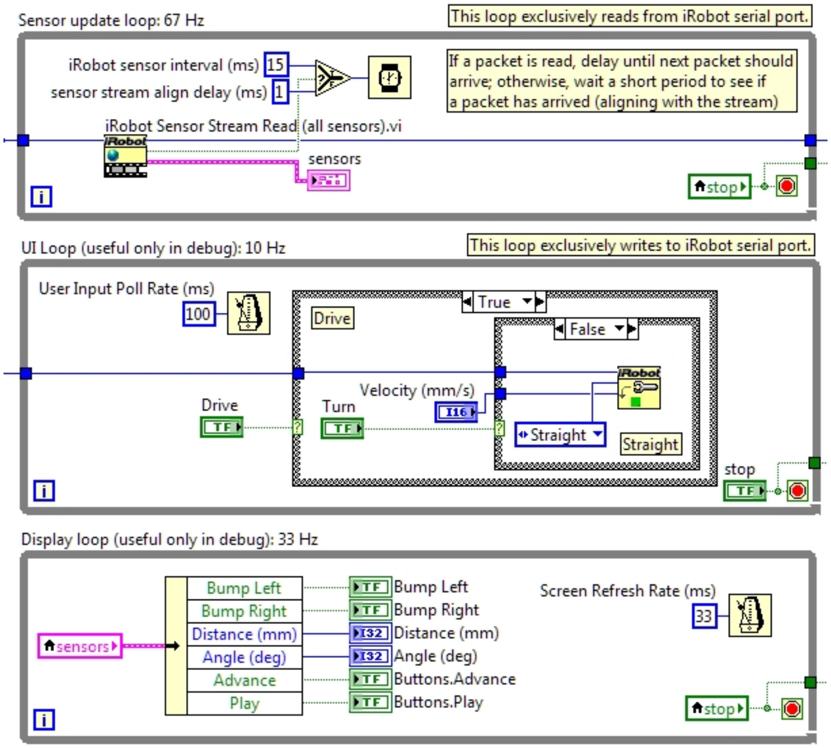

Figure 4: VI to read streaming sensor data from the iRobot Create, and to instruct the robot to drive or turn.

The initialization frame on the left opens the connection to the iRobot and begins a stream of sensor data. The sensor update loop (top) reads and processes the serial communication from the iRobot Create, updating a local variable when new sensor data arrives. The UI Loop (middle) reads controls from the front panel (which is visible when running on a desktop, or when connected to an RT host, or debugging an ARM target) in order to instruct the robot to drive, turn, or stop. The display loop (bottom) updates front panel indicators with the most recent sensor data. Finally, when all loops have terminated, the robot is instructed to stop.

Installing the Package

Download and install the JKI VI Package Manager, which we use to put all iRobot VIs and palettes in the correct location. Then download the appropriate package for your target.

Once installed, you will see the iRobot Create menu in your LabVIEW palette.

Case Study

The iRobot Create, together with LabVIEW Embedded Module for ARM Microcontrollers with ARM Cortex-M3 hardware, is used by the University of California at Berkeley in introductory courses in robotics and embedded systems. The embedded controller - a Texas Instruments LM3S8962 microcontroller - is programmed by LabVIEW to accomplish various tasks.
One interesting example is the Cal Climber, a transformation of the iRobot Create into an autonomous robot that can climb hills using feedback from an accelerometer, avoiding cliffs and obstacles along the way.

The video below features a demonstration of the Cal Climber.

Versions

1.0 (2010-08-10): Initial release.

1.1 (2010-08-18): Minor bug fixes (sensor polling requests wrong packet), more demos, more documentation.

1.2 (2010-08-23): Added missing Byte Array Checksum VI.

1.3 (2011-07-25): Moved to VI Package Manager.

1.4 (2011-08-16): Added sbRIO content.

References

[1] iRobot, *iRobot Create Owner's Guide*, iRobot Inc., January 2006. Available: http://www.irobot.com/filelibrary/create/Create%20Manual_Final.pdf. [Accessed August 10th 2010].

[2] iRobot, *iRobot Create Open Interface (OI) Specification*, Version 2, iRobot, Inc., January 2006. Available: http://www.irobot.com/filelibrary/create/Create%20Open%20Interface_v2.pdf. [Accessed August 10th 2010].

[3] J. Brettle and J. Jensen, "UC Berkeley Teaches Embedded Systems with NI LabVIEW Embedded Module for ARM Microcontrollers," *National Instruments*, April 2009. Available: http://sine.ni.com/cs/app/doc/p/id/cs-12246. [Accessed August 10th 2010].

[4] SparkFun Electronics, "BlueSMiRF Gold" RN-v1 datasheet. February 2008.