Here's something I missed hearing about until I attended a session on programming ARM microcontrollers at National Instruments' recent "NI Week" conference in Austin, Texas. The company's LabVIEW graphical-development software lets engineers create embedded systems and then directly download compiled code to a variety of ARM processors. Engineers might know LabVIEW as a software tool that speeds the development of test, measurement, data-acquisition, and control applications. And although the software has provided FPGA-programming capabilities for some time, the ARM programming direction came as a surprise. Maybe I hadn't read all NI's press releases.

Why ARM? The ARM MCU cores have found homes in many semiconductor device families from companies such as STMicroelectronics, Atmel, Luminary Micro, and NXP. So it seems like a good deal for LabVIEW to link with a such a wide variety of vendors and chips.

NI breaks the supported ARM families into Tier-1 and Tier-2 devices and defines them thus:

**Tier 1 devices are validated microcontrollers that provide an out-of-the-box programming experience. In addition to basic programming and mathematics, tier 1 devices also provide LabVIEW palettes for communications, analog, and digital I/O. Tier 2 devices provide processor support, but no specific palettes are provided, so this requires you to develop and integrate drivers.**

The Tier-2 "palettes" generally refer to I/O device drivers, interrupt control, and cycle-accurate simulations. You can find more information about specific devices, families, and supported dev kits at [www.ni.com/arm/](http://www.ni.com/arm/). This Web page links to other information that includes several tutorials that cover interrupts, debugging, optimization, and CAN-bus applications, for example.

LabVIEW serves as a graphical front end to the Keil uVision software-development tools for ARM processors. ARM (the company) owns Keil, so you can expect the Keil compilers to work with code for almost every ARM processor. For a complete list of the processors the Keil tools support, visit: [www.keil.com/rl-arm/chips.asp](http://www.keil.com/rl-arm/chips.asp).

In essence, engineers use LabVIEW to create their system and then assign operations to MCU resources such as serial ports and analog-to-digital converters (ADCs). The intermediate code—probably C or C++—flows into the Keil tools that then run through its processes to create the assembly-language code for a target MCU.
You can use LabVIEW to program an ARM ARM Cortex-M3 MCU such as the Luminary Micro LM3S8962 on this dev kit, which includes a CAN interface, SD-Card interface, OLED graphics display, Ethernet port and analog I/O.

After compiling their LabVIEW programs, developers use a Keil ULINK2 USB-JTAG adaptor that connects the host PC to the target dev kit or prototype. You can download a trial version of the software, formally named "LabVIEW Embedded Module for ARM Microcontrollers 1.1 - Windows Vista x64/Vista x86/XP," but ensure you have the prerequisite hardware and software already installed: A desktop computer with Windows Vista/XP, the RealView Microcontroller Development Kit including Keil µVision3, LabVIEW 8.6 (or later) with embedded support, and a Keil ULINK2 USB-JTAG adaptor (if you want to program hardware).

If you lack these "components," you can purchase a complete package--less the PC--from NI for about $US 9000. You also can purchase a state-chart option for LabVIEW for an additional $US 1600.

If you program ARM MCUs and want to concentrate on solving high-level problems rather than deal with all the intricacies of C/C++ code, the LabVIEW-ARM tools deserve a look.