100% Open Source Development on ARM Cortex M

Mike Anderson

Chief Scientist

The PTR Group, Inc.

http://ThePTRGroup.com

mike@theptrgroup.com

Who is The PTR Group?

- The PTR Group was founded in 2000
- We are involved in multiple areas of work:
  - Robotics (NASA Space arm)
  - Flight software (over 35 satellites on orbit)
  - Offensive and defensive cyber operations
    - I'll leave this to your imagination ☺
  - Embedded software ports to RTOS/Linux/bare metal
  - IoT systems architecture and deployment

Who am I?

- Over 39 years in the embedded space
- Developed part of VxWorks™
- Instructor for Linux/Android internals
- Mentor for FRC #116 FIRST Robotics Team
- Frequent speaker at:
  - Embedded Linux Conference
  - Embedded Systems Conference
  - CIA Emerging Technology Conference
  - And more…

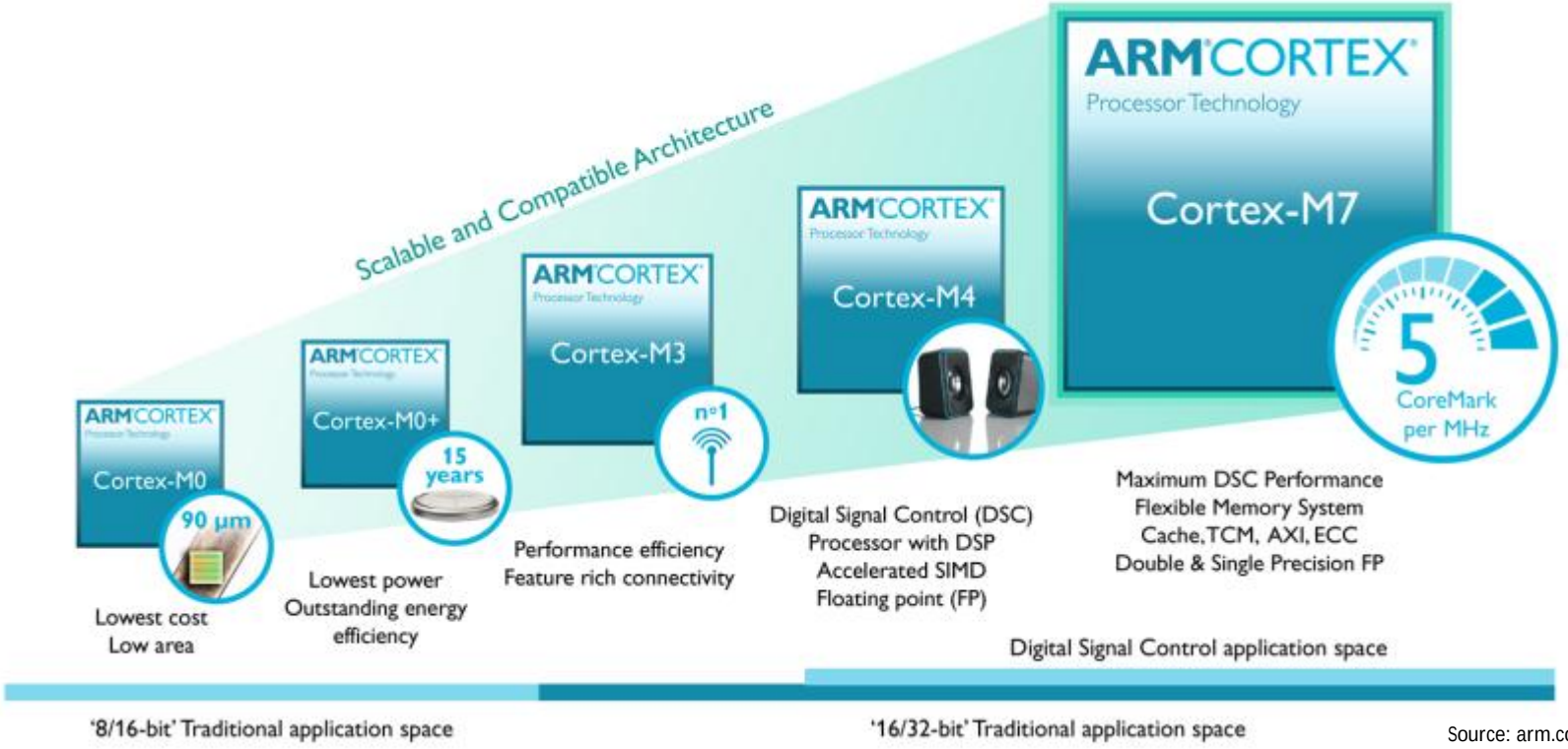
What We'll Talk About…

- Intro to the ARM Cortex-M series and why it plays in the IoT
- Why the focus on open-source solutions?
- What is required in a complete solution?
- Different compilers to do the trick
- IDEs
- Debugging solutions
- Summary

Why the Focus on the ARM Cortex-M?

- ARM's processor cores are steadily increasing in both scope and performance
  - Cortex-A application processors
  - Cortex-R real-time processors
  - Cortex-M energy-efficient microcontrollers
- The Cortex-M series encompasses processors from the M0/M0+ to the M7
  - 32-bit platforms
    - Reduces the hassles and memory restrictions of 8- and 16-bit platforms
  - Over 175 ARM licensees producing silicon
    - Lots of options to choose from in a multitude of sizes and form factors
  - Large support from RTOS and open-source communities for operating systems and ecosystem
    - VxWorks©, ThreadX©, mC/OS©, FreeRTOS, and more

The Cortex-M Family



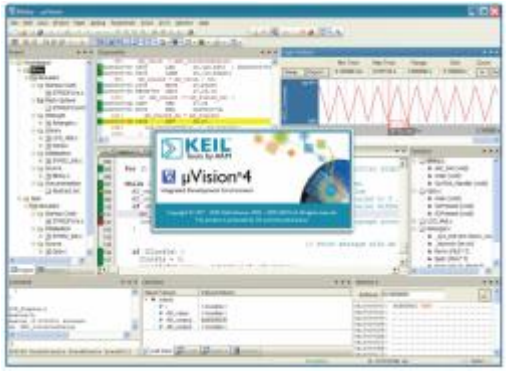
Source: arm.com

How does the Cortex-M Support IoT?

- The Cortex-M series enables a broad selection of embedded devices that need low power consumption
  - Wearables, activity trackers, smart metering, audio headphones, sensor applications, etc.
- Availability of DSP in higher end M4 and M7 cores expands capability
  - Low-cost infotainment, body electronics, audio processing, etc.
- Connectivity that's needed
  - Wireless solutions that include IEEE 804.15.4, Bluetooth Smart, Wi-Fi, etc.

Example Closed-Source Development

- There are a number of well-respected, closed-source development environments that support the Cortex-M
  - Keil, IAR, Crossworks for ARM, ARM DS-5 (non-community version), Green Hills MULTI and many more
- Most of these try to be a complete solution for code development and debugging
  - IDE, Integrated source control, debugger/JTAG interface
- Most use a yearly subscription model
  - Each seat is several thousand $$$ and must be renewed yearly to keep tools from shutting down
- All support Windows, some support OS/X, few support Linux



Source: keil.com

Why Focus on Open-Source?

- There is a tremendous wave known as the “Maker Movement”
  - Innovation coming from individuals and small groups that are creating new applications that will transform our lives
    - E.g., 3D printing, drones/robotics, new sensor platforms
- The makers typically operate on shoestring budgets
  - Their projects are typically labors of love first, potential products second
- Makers want to be able to share their ideas with others without licensing restrictions or large financial barriers
- Open-source hardware availability
  - Arduino, Raspberry Pi, BeagleBone Black, and many more
- Even large corporations are transitioning to open-source
  - Linux, Android, Tizen, etc.

What is Needed for a Complete Solution?

- System development requires many phases
  - Which ones apply depends on your development methodology
    - Agile, Spiral, Waterfall, ad hoc, etc.
- Tools to support all phases of the development cycle
- In general, we need a way to design, edit, compile, debug, deploy and maintain the code
  - Source control systems, code profiling, JTAGs and more are all involved

Getting Started

- Well, that depends on how far you want to take the open-source approach
- Novena open-source laptop platform
  - Open-source hardware, NDA-free data sheets, documentation for the PCBs and more
    - Quad-core ARM iMX-6 with Ubuntu or Debian Linux
- However, most of us will use traditional commercial hardware
  - x86-based development hardware

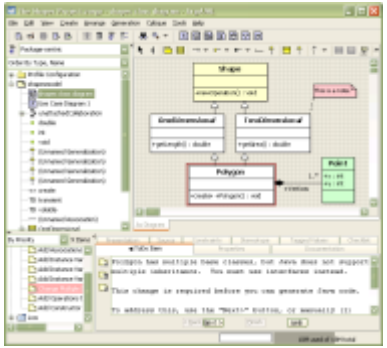
Source: croudsupply.com

Operating System Choices

- At this point, Linux is running on over a billion devices
  - They call them Android phones ;-)
- However, Android and iOS are not known as development platforms
- This leaves Linux, Windows, OS/X and the BSD variants
  - Some development environments can support any of these
- However, we'll limit our discussion to Linux, Windows and OS/X due to time constraints

Starting at the Design

- Tools that support modeling and software architecture are still scarce
  - UML tools are available, but they are not as far along as their commercial counterparts
- ArgoUML
  - Java based, cross-platform including BSD
- StarUML
  - Linux, Windows, OS/X



Source: argouml.tigris.org

Source: staruml.io

Source Code Control

- The two primary players in the open-source community are svn and git
  - Clients for all of the major O/S platforms
- svn and git have fundamentally different approaches
  - svn uses a centralized repository that users check in and out code
- git uses a distributed concept where each user has a private copy of the repository and does their development locally with local branches
  - Periodically synchronize with the global repository
  - Designed for distributed development by geographically disparate developers
  - Has worked well for Linux, Android and other large open-source projects

Compilation

- Open-source tools that support compilation, assembly/disassembly, linking and language front ends boil down to two primary options
  - LLVM/Clang
  - GCC/MinGW
- LLVM is not an acronym – it's the name of the project
  - Started as a University of Illinois research project
  - Available for all major O/S platforms
  - UIUC BSD-style license
  - Native C/C++/Objective-C compiler
    - Support for C++11
  - Compiles 3x faster than GCC in some benchmarks
  - Integrated Clang Static Analyzer
  - LLDB debugger
  - An up-and-coming toolchain with a growing following

Source: llvm.org

GCC

- The GNU Compiler Collection is not one toolchain, but a collection of different language frontends and a code generator backend
  - C/C++11, Objective-C, Ada, Fortran, Java and Go
- Frequently tied with binutils and glibC
- Supports debugging, code profiling, code coverage
- Originally written as the compiler for the GNU operating system
- Runs on all major O/S platforms and supports code generation for nearly all major CPU architectures
- Code is licensed under GPL
  - Exceptions allow the development of commercial applications with GCC
- Arguably, the most popular compiler toolchain on the planet

Source: gcc.gnu.org

Editors/IDEs

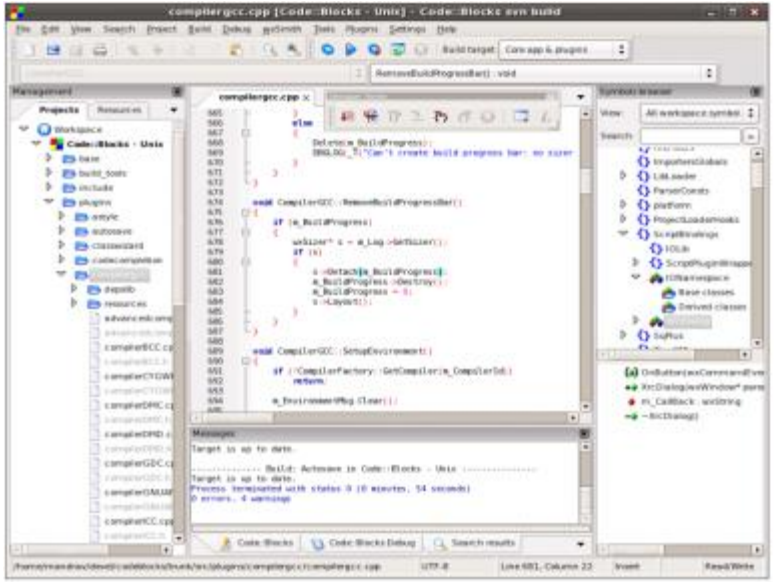
- IDEs/Editors are a religious topic
  - Many developers prefer simple editors like vi or Emacs and command-line building via Makefiles
    - Younger developers consider us dinosaurs ☺

Source: typesofeverything.com

- For those who want pretty GUIs, there are a number of options
  - On Windows, some commercial environments integrate closed-source MS Visual Studio with GCC
    - Atmel Studio is one example
  - Fortunately, that's not the only option

Major Open-Source IDEs

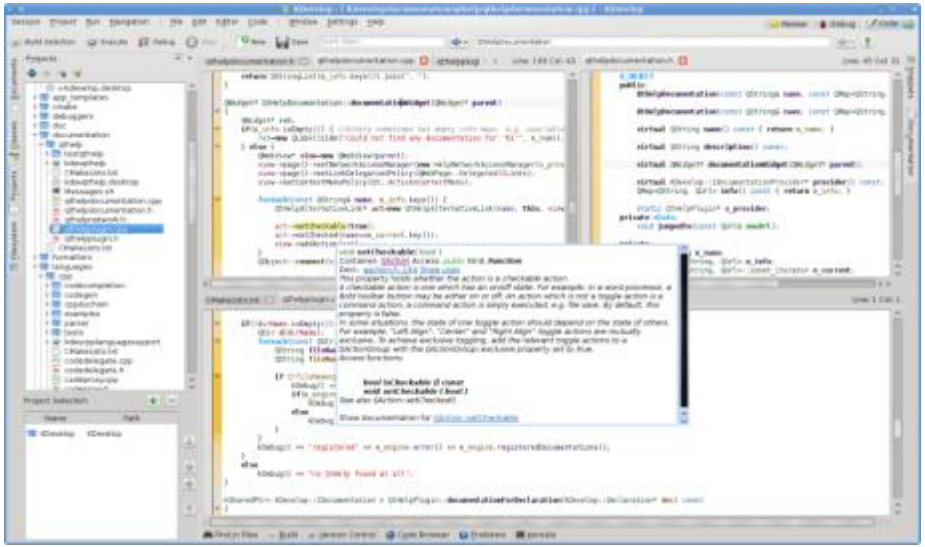
- Eclipse CDT (EPL) – http://www.eclipse.org
  - Runs on all major platforms
  - Integration with ARM GCC/Clang via plug-ins
- Code::Blocks (GPLv3) – http://www.codeblocks.org
  - Runs on all major platforms
  - Supports 7 different compilers including GCC/Clang
- Arduino IDE (GPL) – http://arduino.cc
  - Runs on all major platforms
  - Directly supports GCC, Clang is possible



Source: codeblocks.org

Major Open-Source IDEs #2

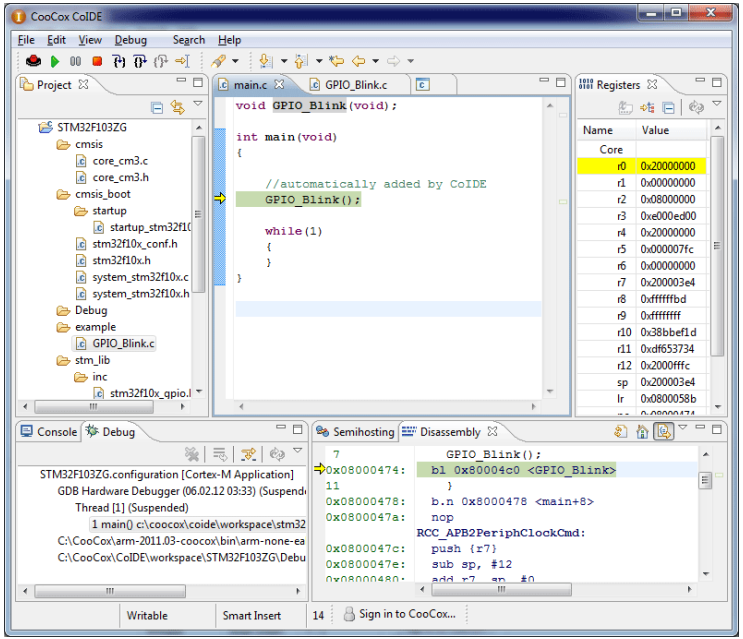
- KDevelop (GPL) – http://www.kdevelop.org
  - Linux, OS/X and new Windows support
  - Requires QT widgets
  - Support for GCC/Clang
- NetBeans (Dual CDDL/GPL) – http://www.netbeans.org
  - Support for all major platforms
  - Support for GCC/Clang
- Ultimate++ (BSD License) – http://www.ultimatepp.org
  - Windows and Linux
  - Support for GCC/Clang



Source: kdevelopment.org

Major Open-Source IDEs #3

- CooCox (BSD) – http://www.coocox.org
  - Windows only
  - Well-developed ecosystem with many support tools
  - Started by Embest Info. Tech, Co. Shenzhen, China
  - Supports GCC
- CodeLite (GPL) – http://www.codelite.org
  - Supports all major platforms
  - Support for GCC/Clang



Source: segger.com

The Reality for IDEs

- In reality, all of the IDEs basically look alike
  - Nothing really new since 1997
- Which one you pick is a matter of personal preference
  - Some are more resource hungry and require more screen real estate
- IDEs in Java are more portable but tend to be slower than ones written in C/C++
  - Still, all of the major O/S variants are supported with almost any of them

Example: Eclipse CDT and GCC

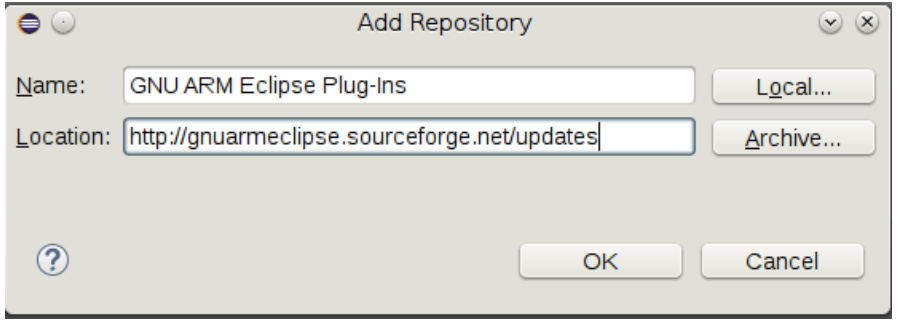
- Integrated with Eclipse 4.4 (Luna) CDT release

Source: gnuarmeclipse.livius.net

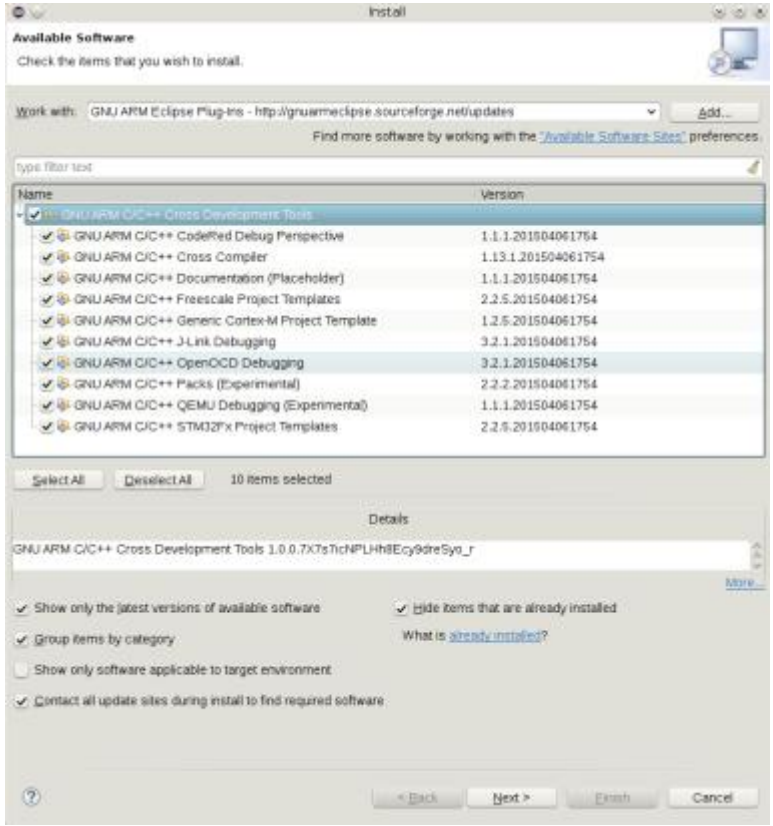
- Support for GCC 4.9 ARM compiler
- Integration with both SEGGER J-Link and OpenOCD JTAG interfaces among others
- Templates for several ARM Cortex-M variants including STMicro STM32F(01234)x and Freescale KLxx

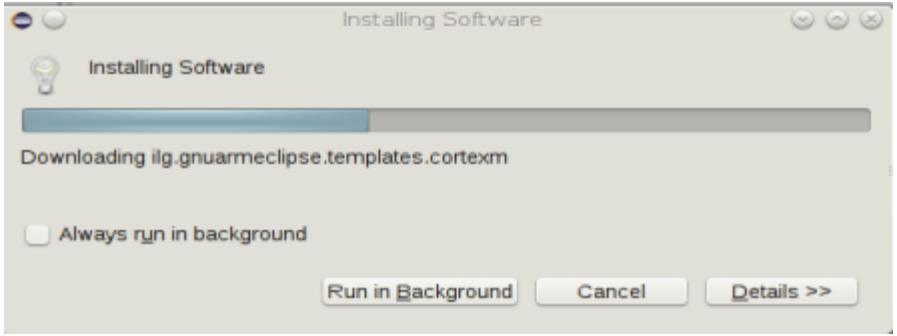
Configure Eclipse for ARM and GCC

- First download and extract the Eclipse Mars.2 CDT release
  - http://www.eclipse.org/downloads/
- Download and extract the latest GCC ARM toolchain
  - https://launchpad.net/gcc-arm-embedded
- Start Eclipse and open Help->Install New Software and add the ARM Eclipse plug-in site

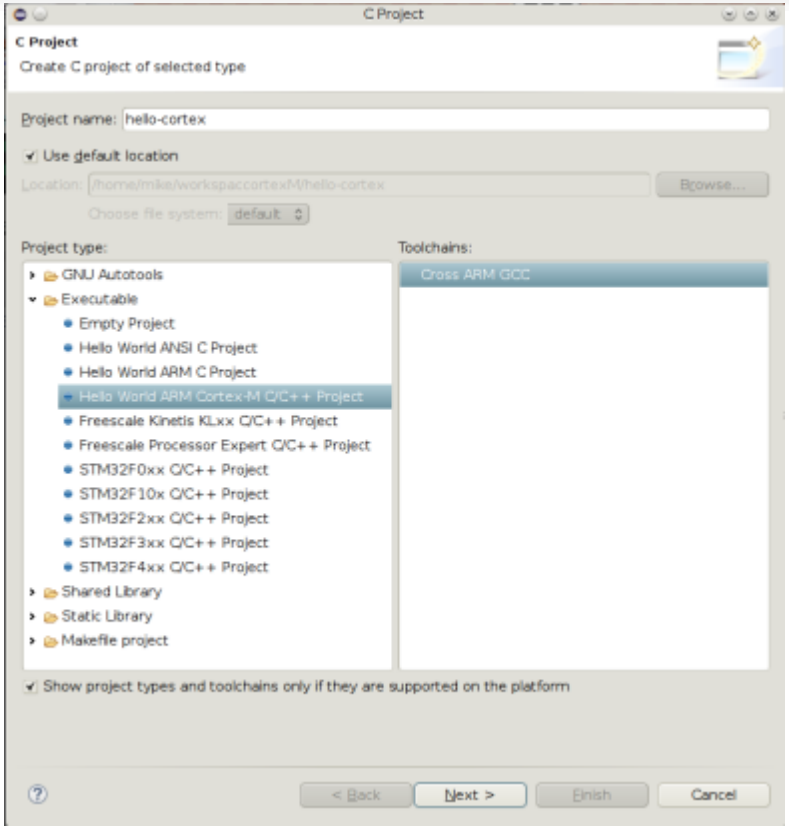


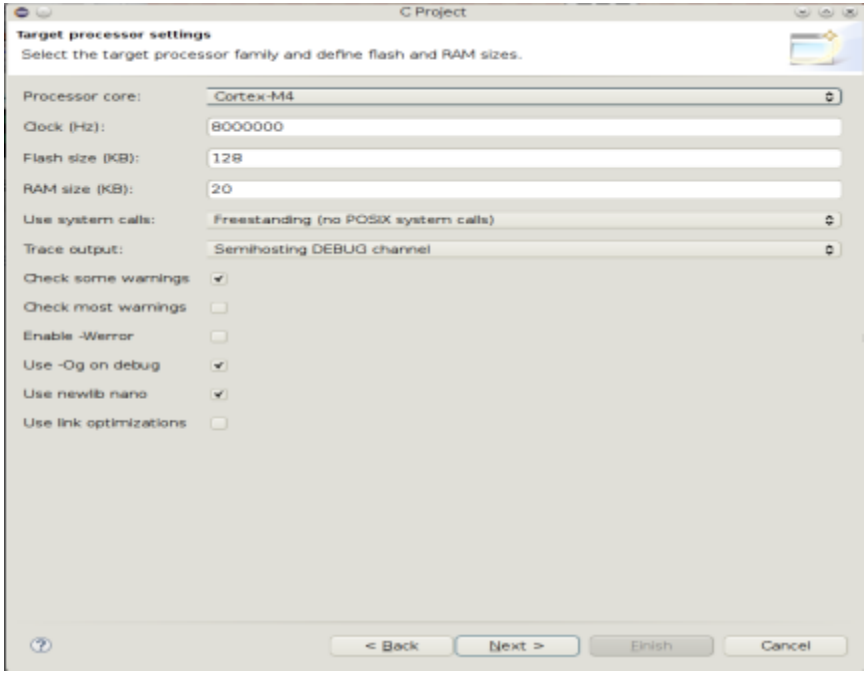
Select Plug-Ins



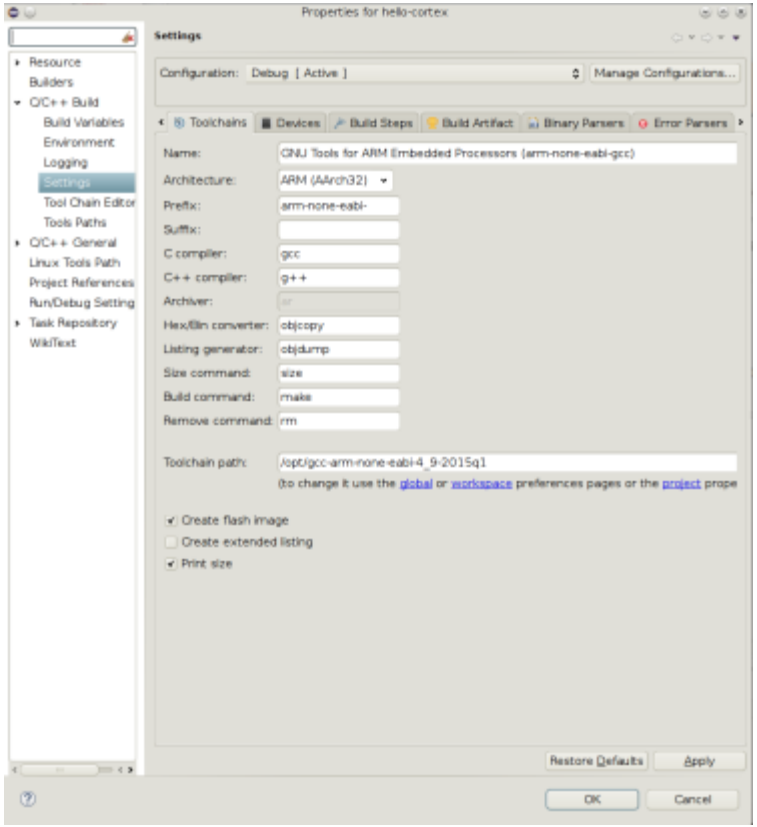


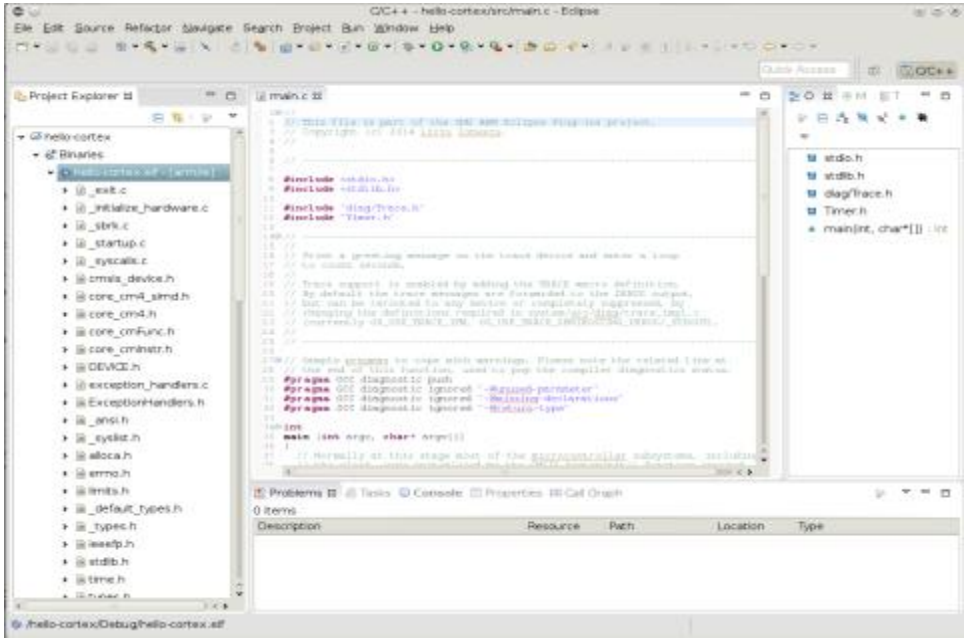
Create a Project





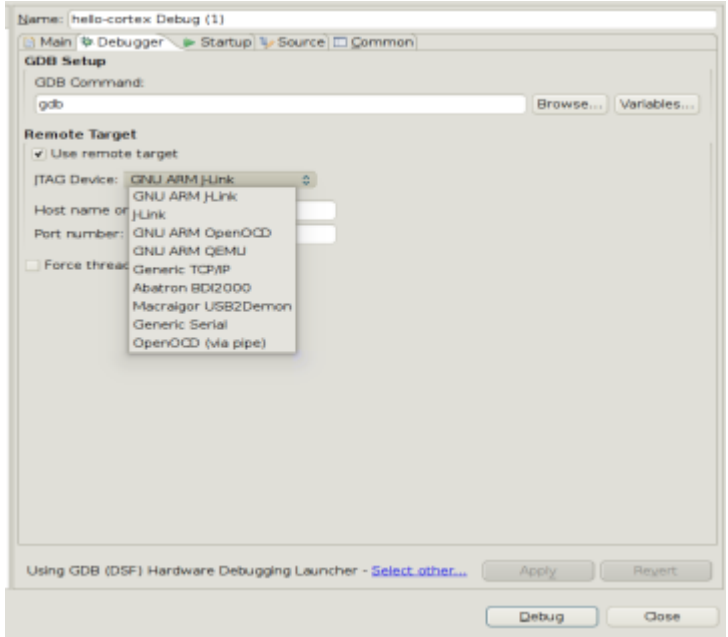
Set Toolchain Path and Build





Debugger Options

- Now that we have a binary, how do we debug it?
- The Eclipse debugging plug-in supports several different debugger options
  - OpenOCD
  - SEGGER J-Link
  - Abatron BDI-2000
  - Macraigor USB2Daemon
  - More…
- Some JTAG debuggers are open-source hardware
  - BusPirate, etc.
  - But, commercial, OpenOCD-compatible USB debuggers can be had for < $100



Summary

- The ARM Cortex-M series processors will likely see a lot of use in the IoT
  - SWaP considerations and wireless connectivity options
- There are several open-source development tools
  - The problem is “Which one do I pick?”
  - Since they are open-source, try several to find the one you like most
- Either GCC or Clang/LLVM can generate the code
  - ARM works with the open-source community to optimize the compilers to make them comparable to commercial offerings
- Your search engine is your friend
  - Many have blazed the trails ahead of you, so follow their lead