Designing a Robust IIoT to SCADA Gateway

Class 4: Programming the R-IN Protocol Engine

October 25, 2018

Charles J. Lord, PE
President, Consultant, Trainer
Blue Ridge Advanced Design and Automation

This Week's Agenda

10/22 The Challenges of IIoT and Industrial Ethernet

10/23 Introduction to the RZ/N1

10/24 Many Protocols, One Abstraction - GOAL

10/25 Programming the R-IN Protocol Engine

10/26 Writing and Testing Our Application

DesignNews

Blue Ridge Advanced Design and Automation
Asheville, North Carolina

CEC CONTINUING EDUCATION CENTER

Digi-Key ELECTRONICS

This Week's Agenda

10/22 The Challenges of IIoT and Industrial Ethernet

10/23 Introduction to the RZ/N1

10/24 Many Protocols, One Abstraction - GOAL

10/25 Programming the R-IN Protocol Engine

10/26 Writing and Testing Our Application

Presented by:

DesignNews

Blue Ridge Advanced Design and Automation
Asheville, North Carolina

CEC CONTINUING EDUCATION CENTER

Digi-Key ELECTRONICS

Developing for the RZ/N1D

- For the next two days, we will look at some example programs and what setup is needed to write, load, and debug code on the three processors in the Renesas RZ/N1D

- Our setup will be the minimum needed, without the protocol analyzers and other test equipment that would be needed for proper development

Question1 – What IDEs do you use for ARM?

DesignNews

Blue Ridge Advanced Design and Automation
Asheville, North Carolina

4

CEC CONTINUING EDUCATION CENTER

Digi-Key ELECTRONICS

Programming the R-IN

ARM Cortex-M3 (with special HW accel)

- Windows (7/10) computer running
  IAR Embedded Workbench for ARM 8.22+

- GOAL OSAL (included in CONNECT-IT)

- Renesas Drivers

- µITRON

- IAR iJet JTAG

DesignNews

Blue Ridge Advanced Design and Automation
Asheville, North Carolina

CEC CONTINUING EDUCATION CENTER

Digi-Key ELECTRONICS

Programming the Dual A7s

Dual ARM Cortex-A7

- Linux computer (Ubuntu 16+ recommended) running linaro-6.3.1-2017.02 GCC-based cross-compiler system for ARM

- Although we will concentrate on the dual A7 application side tomorrow, we need to run some demos on the application side to test our R-IN implimentations

Development Environments

Device	RZ/N1D		
CPU	Cortex-A7		Cortex-M3 (R-IN engine)
OS	Linux or ThreadX		uITRON (HWRTOS)
IDE	Linux	ThreadX	IAR EWARM
		IAR SYSTEMS	IAR SYSTEMS
Bootloader	U-Boot*		

- We will use Linux for the A7s for our demos

- They can also run ThreadX or VxWorks

- Choices can also depend on your license for IAR

Presented by:

DesignNews

Blue Ridge Advanced Design and Automation
Asheville, North Carolina

CEC CONTINUING EDUCATION CENTER

Digi-Key ELECTRONICS

Bring up the Quick Start

CONNECT IT! - ETHERNET RZ/N1D
Quick Start Guide

RZ Family RZ/N1 Series

List of reference documents

Document name
User's Manual: System Introduction, Multiplexing, Electrical and Mechanical Information
User's Manual: System Control and Peripheral
User's Manual: Peripherals
User's Manual: R-IN Engine and Ethernet Peripherals
User's Manual: Generic Open Abstraction Layer
RZ/N1D Development Board Schematic
RZ/N1D Development Board Setup Notes
RZ/N1 U-Boot User Manual
RZ/N1 Linux User Manual

Presented by:

DesignNews

Blue Ridge Advanced Design and Automation
Asheville, North Carolina

CEC CONTINUING EDUCATION CENTER

Digi-Key ELECTRONICS

Your Windows Computer Will Need

Presented by:

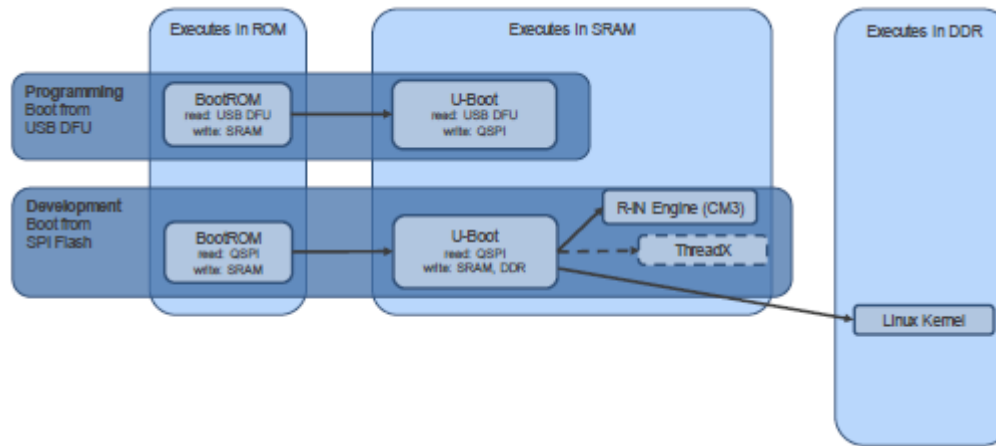# U-boot Is Also Needed

Presented by:

Now We Bring up IAR EWARM

GOAL Projects**	Several example projects showing the different functionalities for the switch management and GOAL	Cortex M3	\GOAL\goal\projects\00410_goal\
Protocol Stack Projects***	Several Protocol Stack slave application examples, which are running I/O communication between a PLC and RZ/N1D	Cortex M3	\GOAL\goal\projects\goal_co_lib\ \GOAL\goal\projects\goal_ecat\ \GOAL\goal\projects\goal_eip_lib\ \GOAL\goal\projects\goal_epl_lib\ \GOAL\goal\projects\goal_mbs\ \GOAL\goal\projects\goal_pnio_lib\

Boot Sequence and Binaries

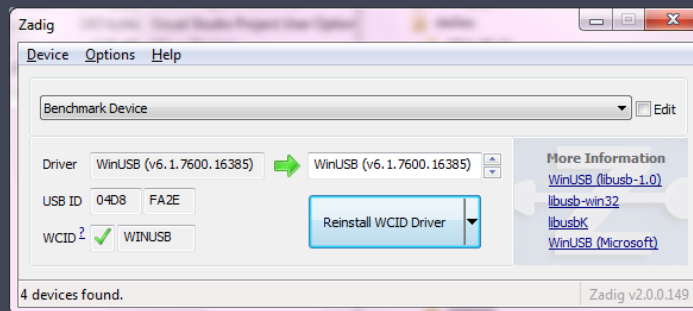Linux and U-Boot binaries are to be found in the solution kit under:

U-Boot for RZ/N1D	YCONNECT-IT-RZN_V1.X\Software\U-Boot-and-Linux\u-boot\binaries\u-boot-rzn1d400-db.bin.spkg
Linux Kernel	YCONNECT-IT-RZN_V1.X\Software\U-Boot-and-Linux\kernel\binaries\uImage
Device Tree Blob	YCONNECT-IT-RZN_V1.X\Software\U-Boot-and-Linux\kernel\binaries\uImage-rzn1d400-db.dtb

Presented by:

Blue Ridge Advanced Design and Automation
Asheville, North Carolina

Another Handy Tool

Presented by:

Our Programming Connections

SW1
SW Dip Switch

J2
Debug Connector

CN9
USB Function

CN10
- FTDI serial-over-USB
- FTDI JTAG-over-USB
- **Power Connector**

SW4
Soft Reset

SW5
Boot Mode

CN6
Power via USB
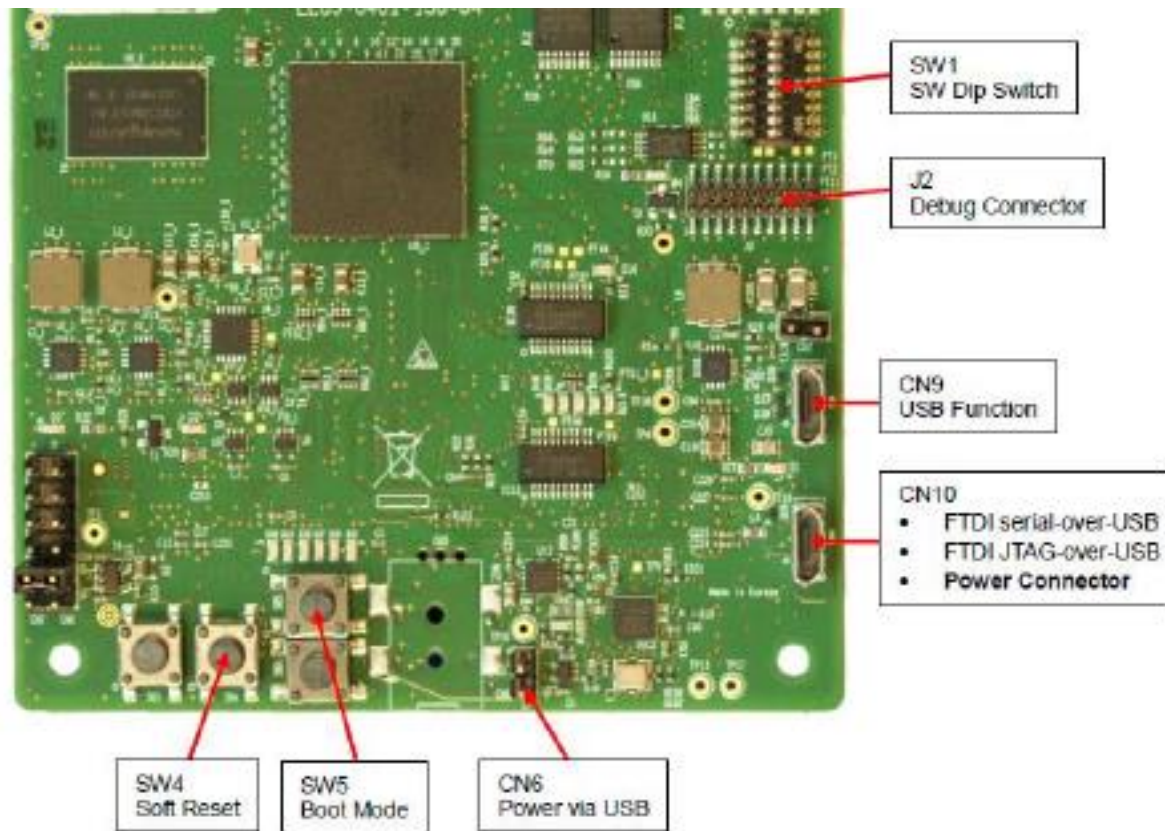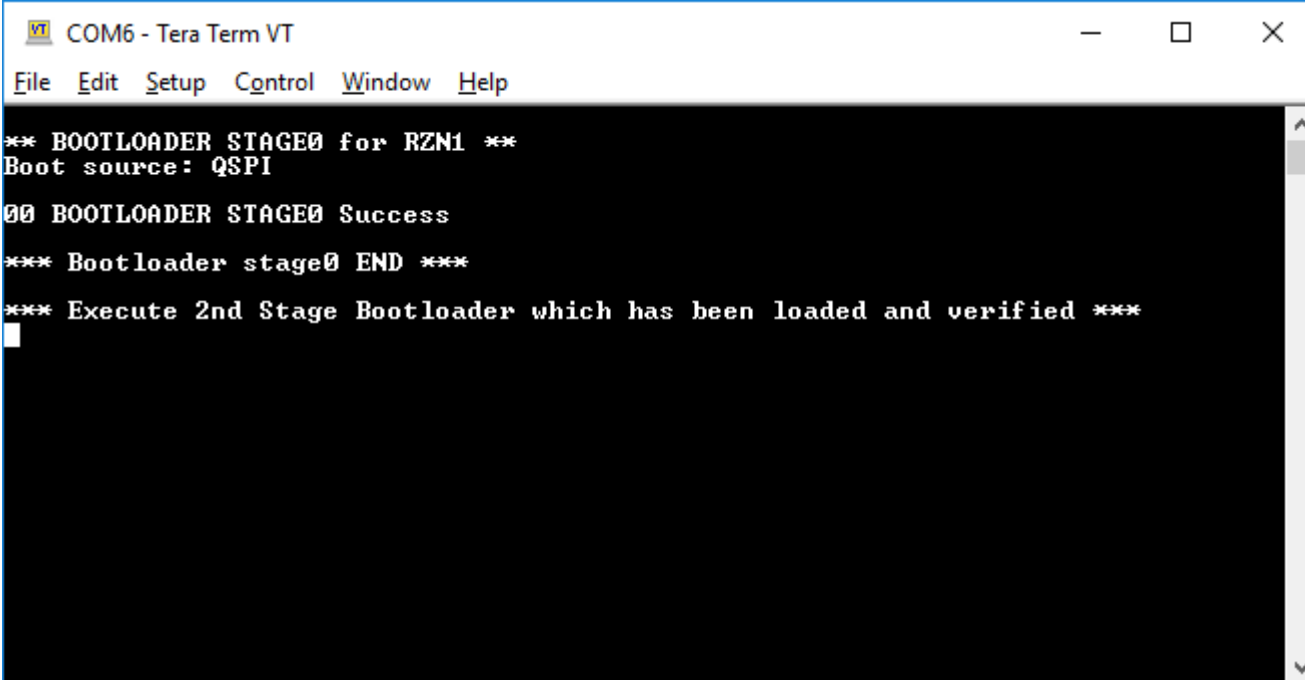
Question 2 – What Does FTDI stand for – and who started the company?

Presented by:

Design News

Blue Ridge Advanced Design and Automation
Asheville, North Carolina

14

CEC CONTINUING EDUCATION CENTER

Digi-Key ELECTRONICS

Putting it together

- Following the manual, we loaded DFU on our PC

- Then we run Zadig on our PC (once to register the software-over-USB)

- Connect to the "CN10 USB port"

- Once we set up our new serial port

- we installed U-Boot on the RZ/N1D

We are now ready to run the bootloader

Presented by:

Installing Linux

- For today, we will just load the pre-compiled Linux that is provided and set up the environment. The distribution is based on Yocto which is popular for embedded systems

- These steps are in the quick-start manual pages 17-20
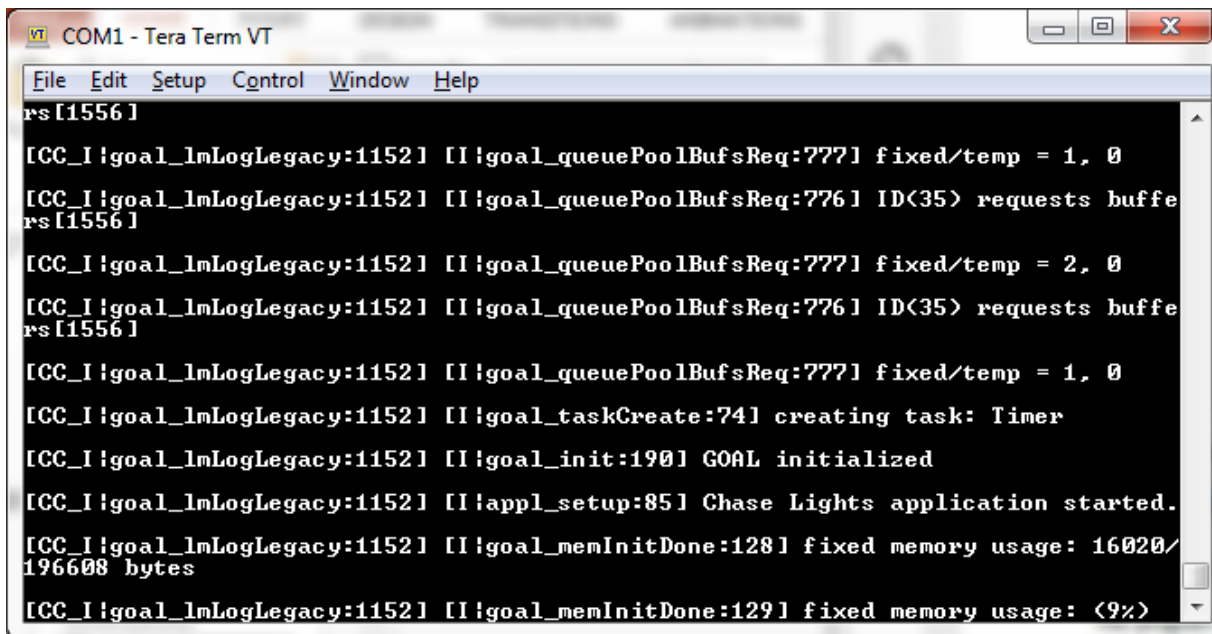
Presented by:

Our Demo Code

- We will start with the general demo to test out our setup

- This is found at *Software\GOAL\goal\projects\00410_goal\ chase_lights\iar\7_70\rzn1d_demo_board\ rzn1d_demo_board_eb.eww*

- NOTE – is you don't have the expansion board use …rzn1d_demo_board.eww

- Load this workspace in IAR (8.22 or later)

Presented by:

Now we need to set the environment

- In the U-Boot console type

  => setenv cm3 "sf probe&&mw 0x04000004 1&&rzn1_start_cm3"

  => saveenv

  => run cm3

- Press enter.

- Now connect your i-Jet to the JTAG connector, make and debug the project.

Your Green LEDs Should "Chase"

- The green LEDs will chase back and forth and you should see the following in your serial terminal window

Now We Are Ready

- We can now look at the different fieldbus drivers and stacks in GOAL, but for that we need to be able to program the A7s to run our test application

- Tomorrow, we look at how we program the application processor in Linux and run some real code!

Question 3 – Experience with Linux? Embedded? What Distos?

Presented by:

This Week's Agenda

10/22 The Challenges of IIoT and Industrial Ethernet

10/23 Introduction to the RZ/N1

10/24 Many Protocols, One Abstraction - GOAL

10/25 Programming the R-IN Protocol Engine

10/26 Writing and Testing Our Application

Presented by:

DesignNews

Blue Ridge Advanced Design and Automation
Asheville, North Carolina

CEC CONTINUING EDUCATION CENTER

Digi-Key ELECTRONICS

Please stick around as I answer your questions!

- Please give me a moment to scroll back through the chat window to find your questions

- I will stay on chat as long as it takes to answer!

- I am available to answer simple questions or to consult (or offer in-house training for your company)
  c.j.lord@ieee.org
  http://www.blueridgetechnc.com
  http://www.linkedin.com/in/charleslord
  Twitter: @charleslord
  https://www.github.com/bradatraining

Presented by:

DesignNews   Blue Ridge Advanced Design and Automation
Asheville, North Carolina   CEC CONTINUING EDUCATION CENTER   Digi-Key ELECTRONICS