

Embedded System Design Techniques™

Bootloader Design for MCUs

Session 1: Bootloader Models and Concepts

January 25th, 2016
Jacob Beningo, CSDP

Course Overview

- **Bootloader Models and Concepts**
- Protocol Design
- Setting up a Test Application
- Bootloader Implementation
- Troubleshooting Techniques



Jacob Beningo
Principal Consultant



P.O. Box 400
Linden, Michigan 48451

www.beningo.com

E : jacob@beningo.com

T : 810-844-1522

Twitter : [Jacob_Benigo](https://twitter.com/Jacob_Benigo)

f : [Benigo Engineering](https://www.facebook.com/BenigoEngineering)

in : [JacobBenigo](https://www.linkedin.com/in/JacobBenigo)

EDN : [Embedded Basics](#)

ARM Connected Community

Newsletters

- [Embedded Bytes](#)



<http://bit.ly/1BAHYXm>

Training

- [MicroPython](#)
- [Bootloaders](#)
- [Low Power Design](#)
- [Resume Workshop](#)
- [Embedded Techniques](#)

Courses Overview

CEC 2013 – 2015

Fundamentals of Embedded Software (2013)

Mastering the Software Design Cycle (2014)

Python for Embedded Systems(2014)

Software Architecture Design (2014)

Baremetal C (2015)

Mastering the ARM Cortex-M Processor (2015)

Writing Portable and Robust Firmware in C (2015)

Design Patterns and the Internet (2015)

CEC 2016

Bootloader Design for MCUs
January 2016

Topic TBD
May 2016

Side Topics 2016

Real-Time Software using Micro Python

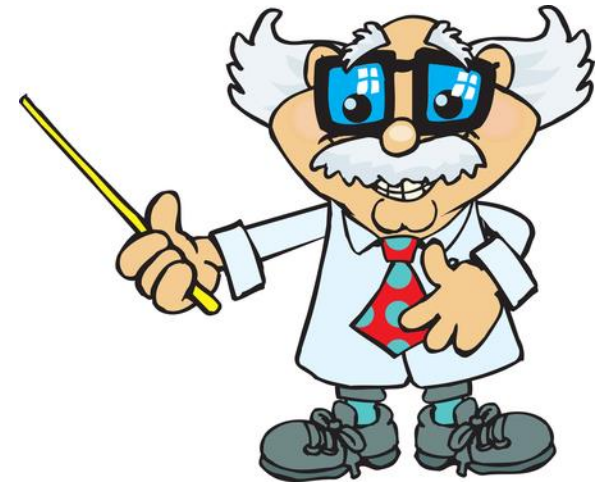
PROMO-PYTHON

Embedded System Design Techniques Series

<http://bit.ly/1BAHYXm>

Session Overview

- Introduction to Bootloaders
- Where bootloaders come from
- Bootloader Models
- Requirements
- The Bootloader System
- Bootloader Behavior



Where do bootloaders come from?

1



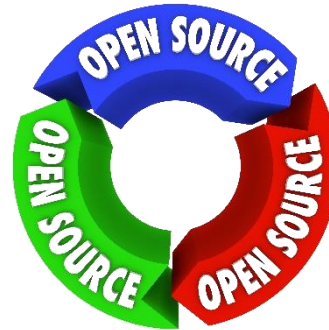
Silicon Vendors

2



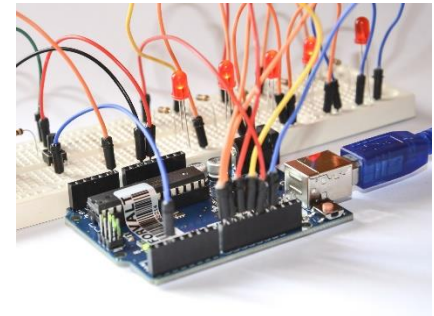
3rd Party

3



Open Source

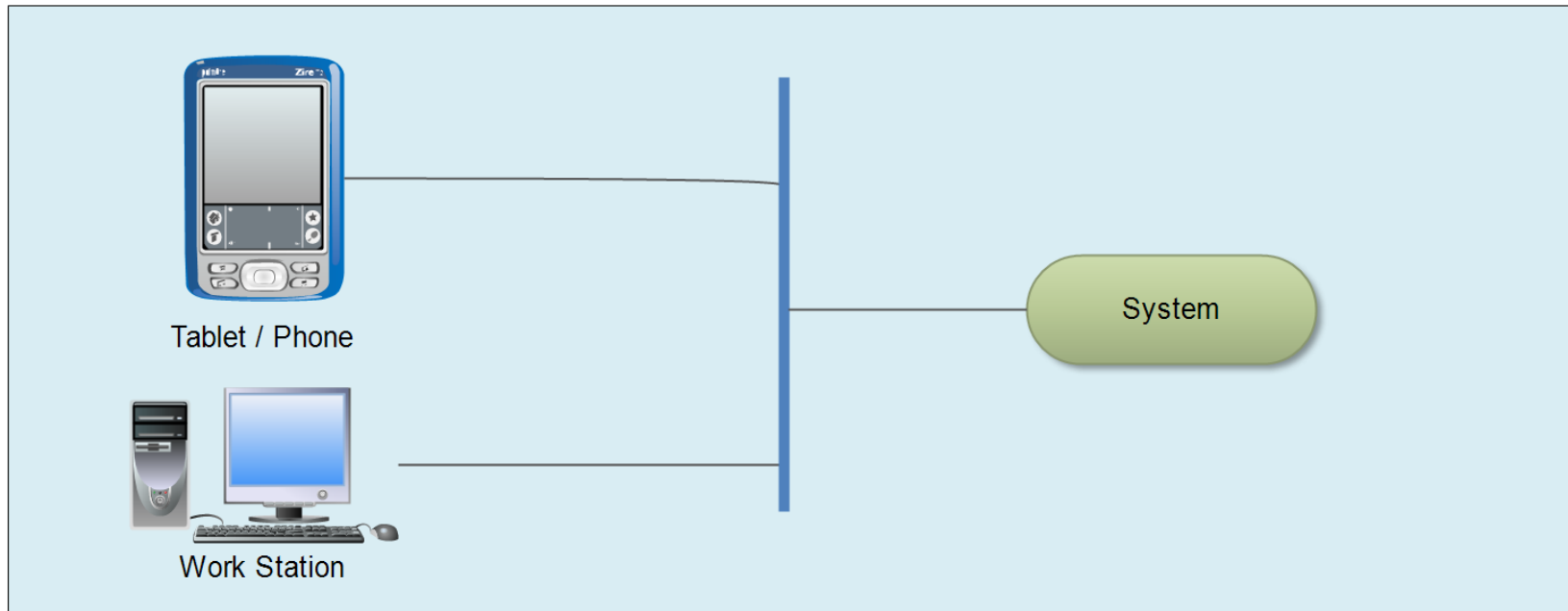
4



Roll your own

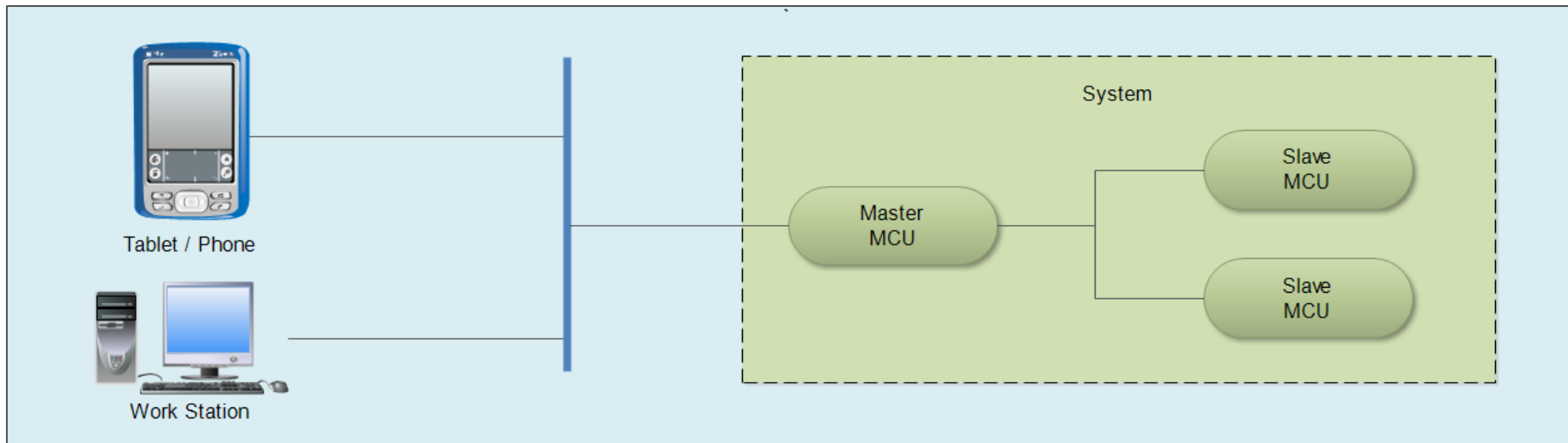
Bootloader Models

- Single MCU System (Traditional / Most Common)
- Flashing Method
 - Laptop / Workstation
 - Tablet or mobile device
 - USB Flash System



Bootloader Models

- Multi MCU System (Not as common or traditional)
- Flashing Method
 - Laptop / Workstation
 - Tablet or mobile device
 - USB Flash System
- Master MCU
 - Can be updated itself
 - Passes new application to slave devices and acts as the flash tool



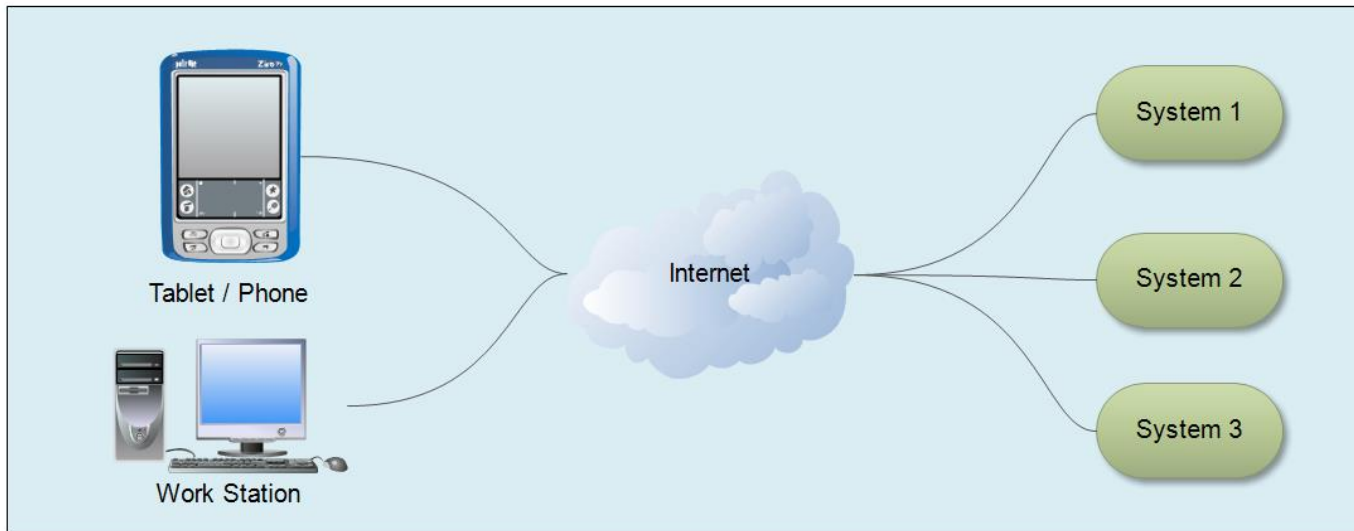
Bootloader Models

- MCU System

- Single MCU Devices
- Multi MCU Devices
- Systems are internet enabled
- Physical Separation from imaging tool

- Flashing Method

- Internet Connected Devices
 - Tablets
 - Phones
 - Computers



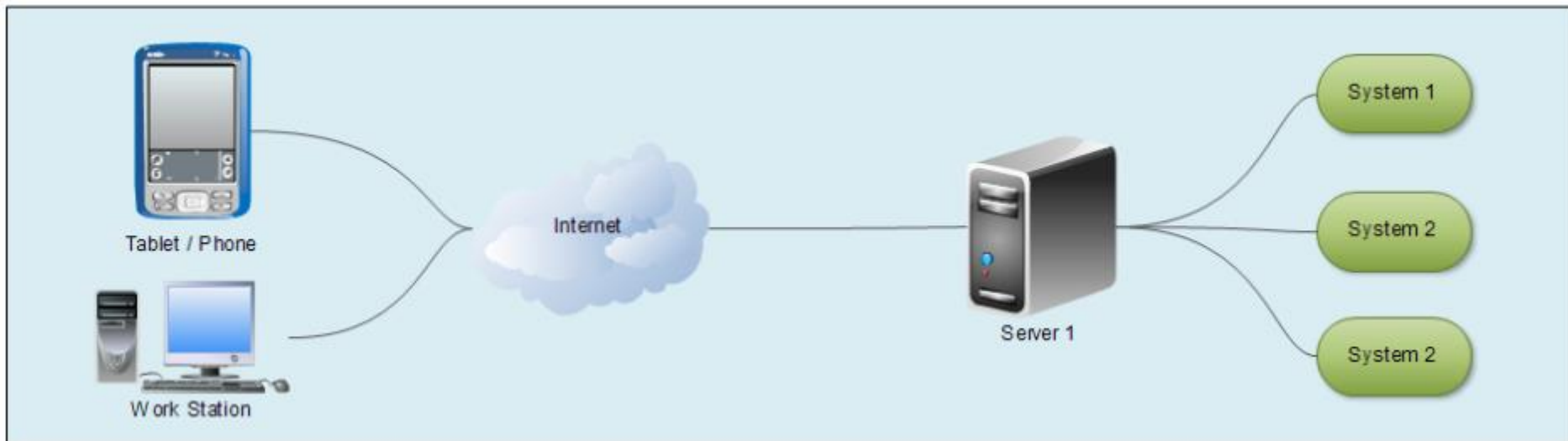
Bootloader Models

- MCU System

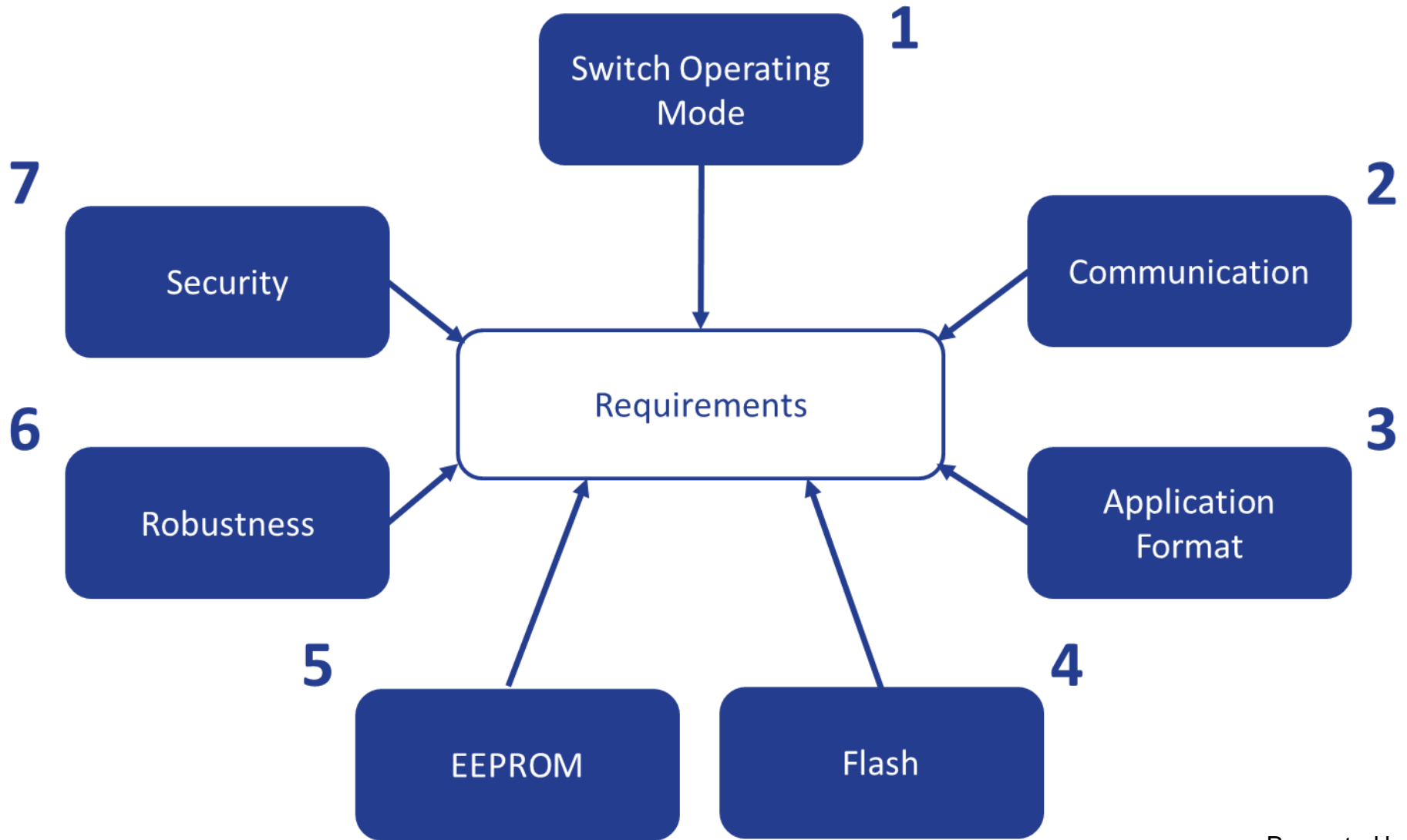
- Single MCU Devices
- Multi MCU Devices
- Systems are not internet enabled
- Physical Separation from imaging tool

- Flashing Method

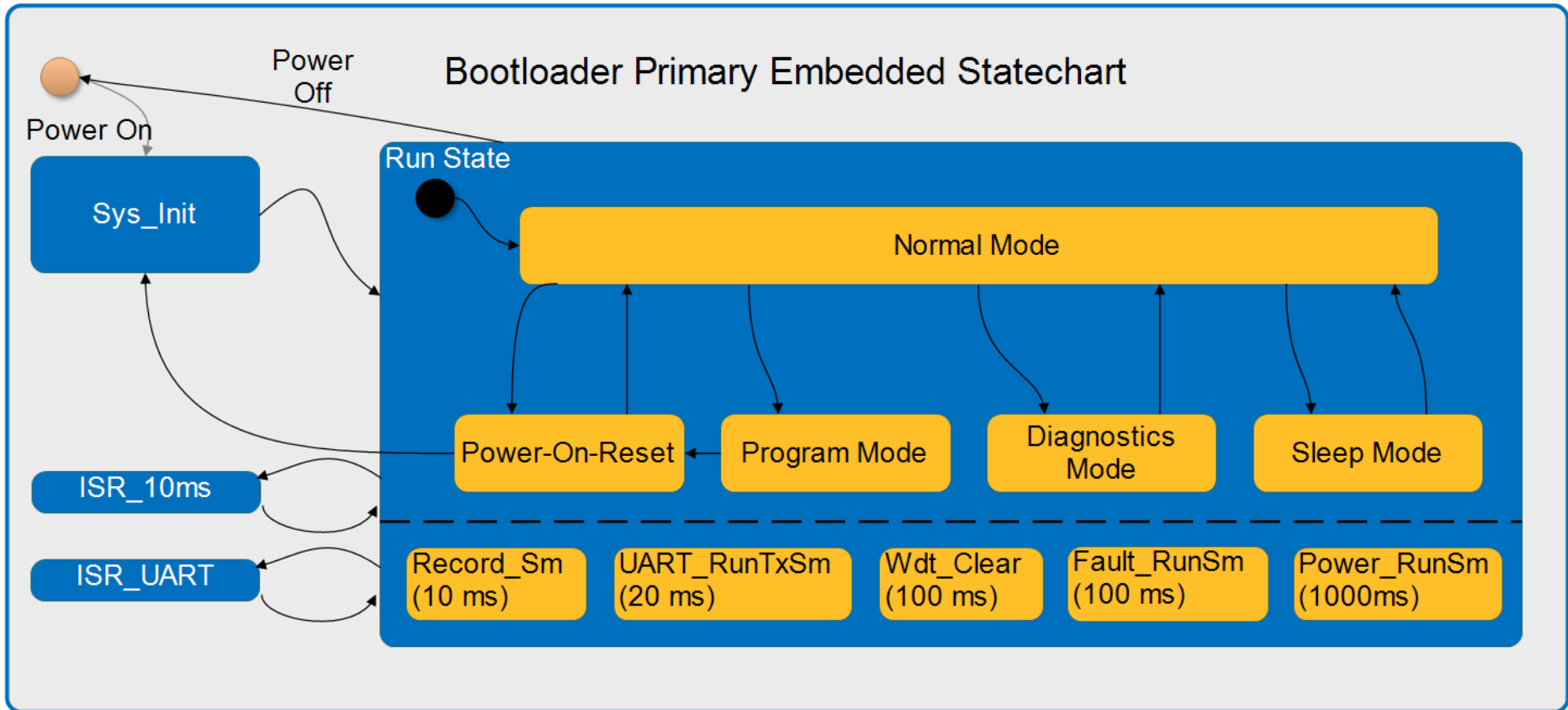
- Internet Connected Devices
 - Tablets
 - Phones
 - Computers



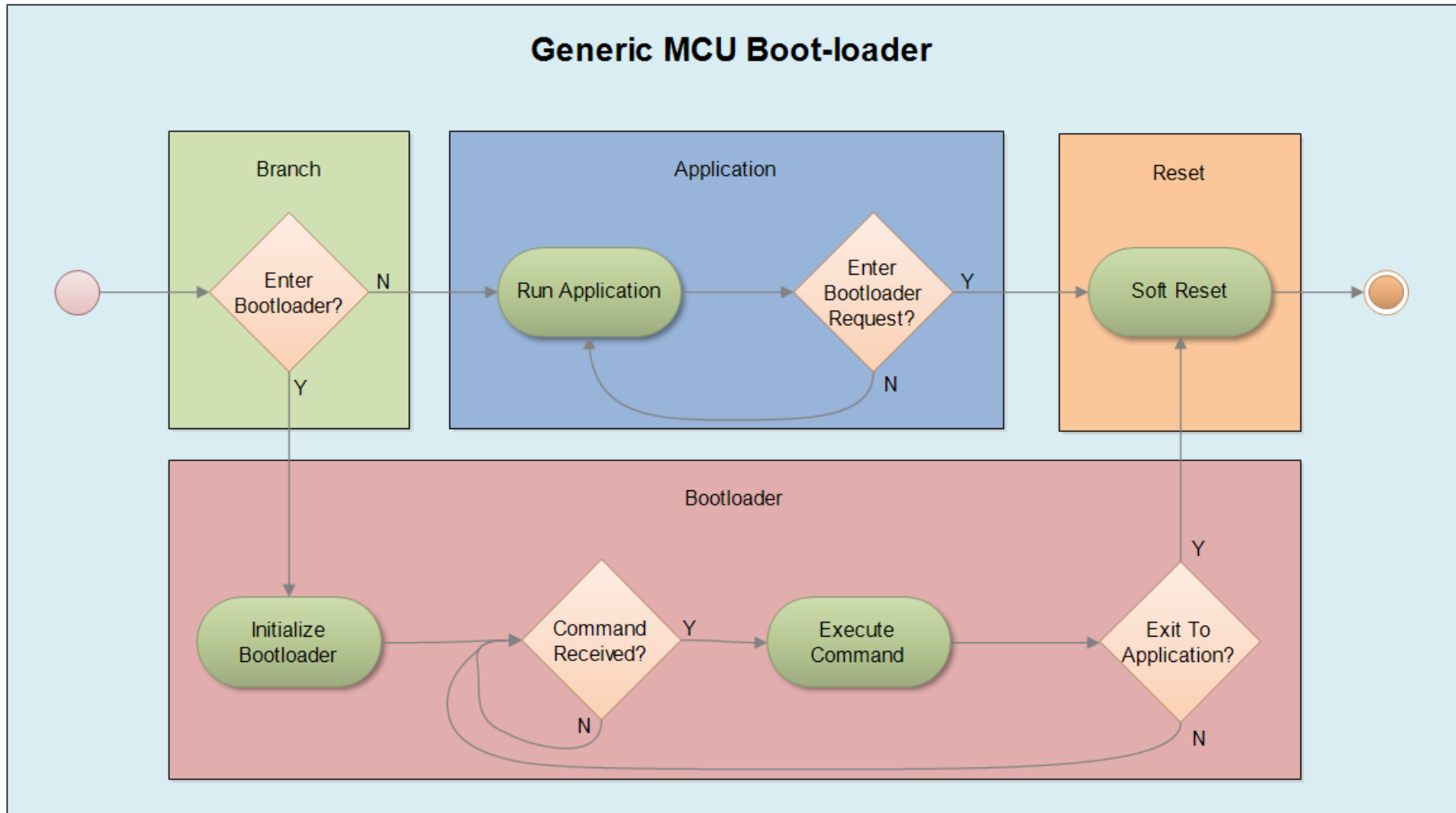
Requirements



The Bootloader System



The Bootloader System



Bootloader Behavior

There are two behavioral models for a boot-loader

- 1) Boot-loading process is completely automated and self-contained within the system
 - boot-loader would automatically detect the new firmware and manage its own flashing process
 - Commands from an external source would not be required to successfully carry out the boot-loading process
 - Ex. SD card boot-loader, web based automated updated, etc

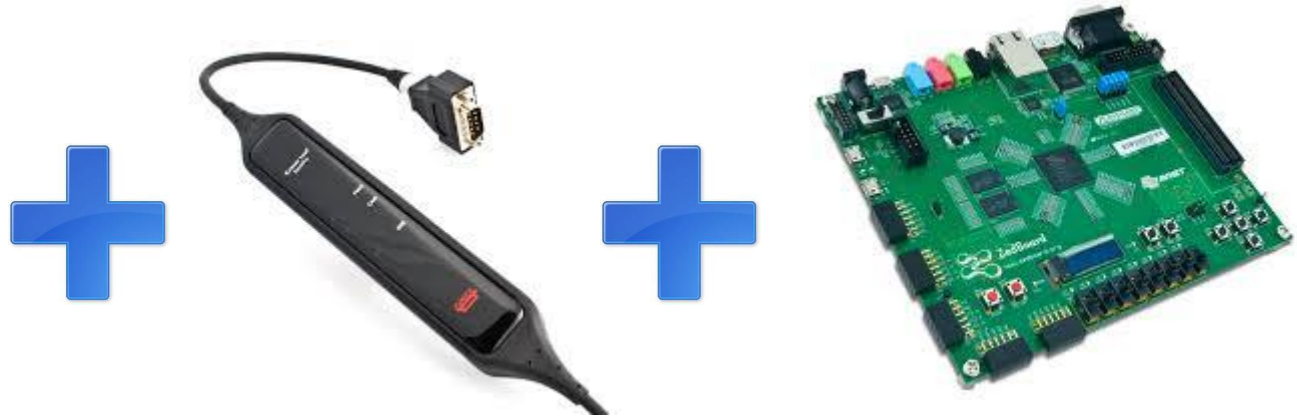


Bootloader Behavior

There are two behavioral models for a boot-loader

2) Boot-loading process is not self-contained but initializes into an idle state and awaits instructions from an outside source

- a pc based software application commands the boot-loader into the different states necessary to flash a new image onto the system
- Ex. CAN, UART, USB boot-loader



Additional Resources

- Download Course Material for
 - Updated C Doxygen Templates (Sept 2015)
 - Example source code
 - Bootloader White Paper
 - Templates
- Microcontroller API Standard
- EDN Embedded Basics Articles
- Embedded Bytes Newsletter



From www.beningo.com under

- Blog and Articles > Software Techniques > CEC Bootloader Design for MCUs



Jacob Beningo
Principal Consultant



P.O. Box 400
Linden, Michigan 48451

www.benigo.com

E : jacob@benigo.com

T : 810-844-1522

Twitter : [Jacob_Benigo](https://twitter.com/Jacob_Benigo)

f : [Benigo Engineering](https://www.facebook.com/BenigoEngineering)

in : [JacobBenigo](https://www.linkedin.com/in/JacobBenigo)

EDN : [Embedded Basics](#)

ARM Connected Community

Newsletters

- [Embedded Bytes](#)



<http://bit.ly/1BAHYXm>

Training

- [MicroPython](#)
- [Bootloaders](#)
- [Low Power Design](#)
- [Resume Workshop](#)
- [Embedded Techniques](#)