

Embedded System Design Techniques™

Bootloader Design for MCUs

Session 4: Bootloader Implementation

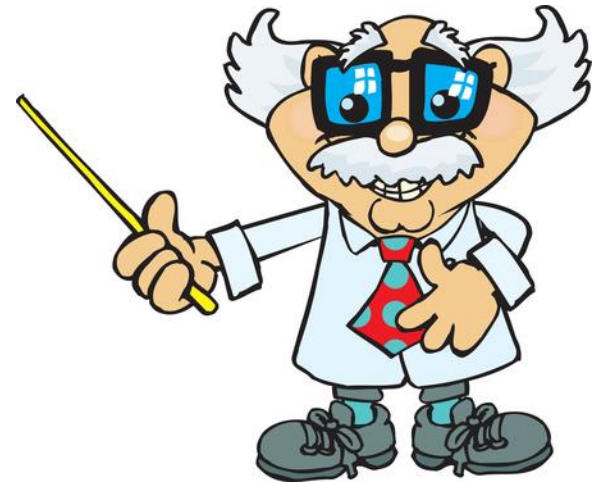
January 28th, 2016
Jacob Beningo, CSDP

Course Overview

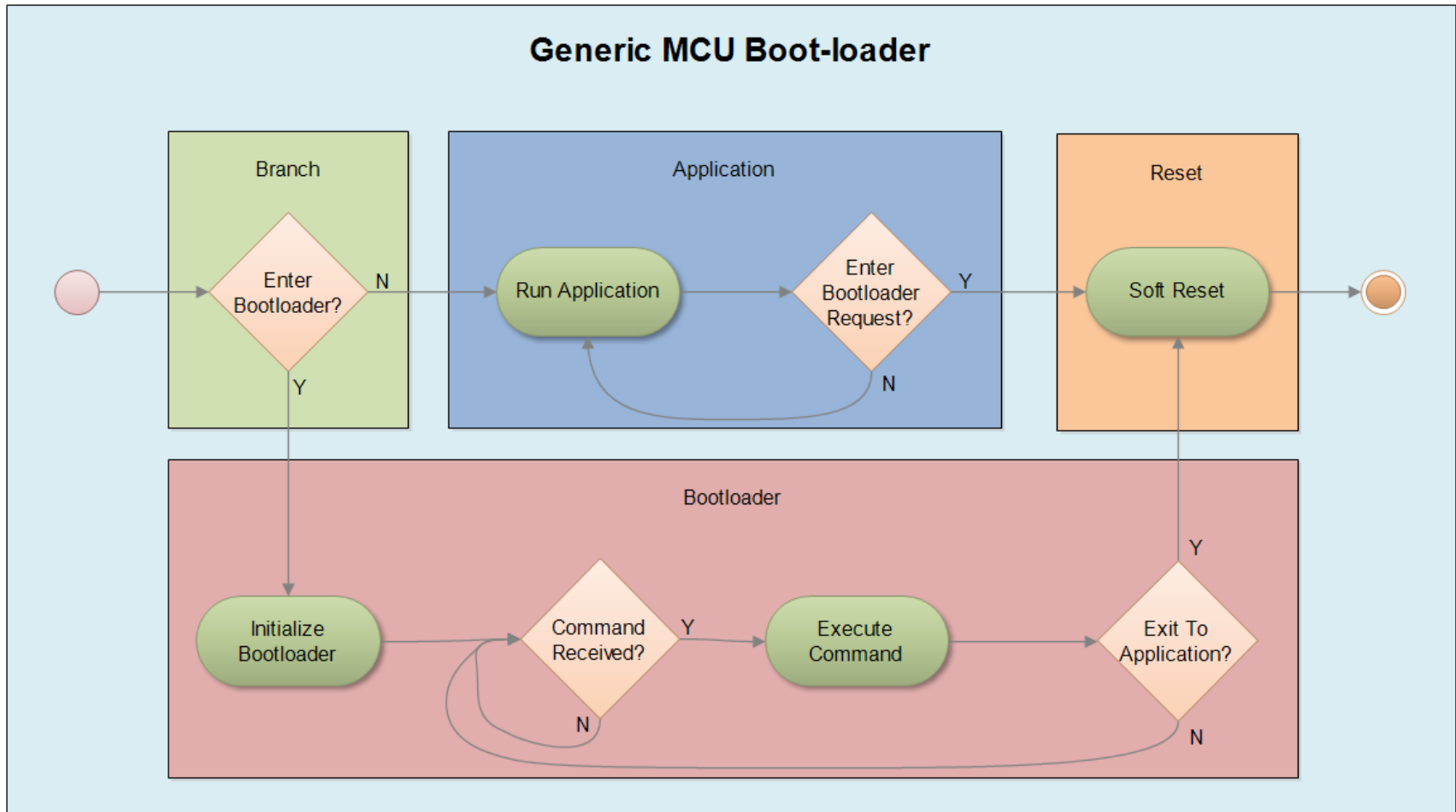
- Bootloader Models and Concepts
- Protocol Design
- Setting up a Test Application
- **Bootloader Implementation**
- Troubleshooting Techniques

Session Overview

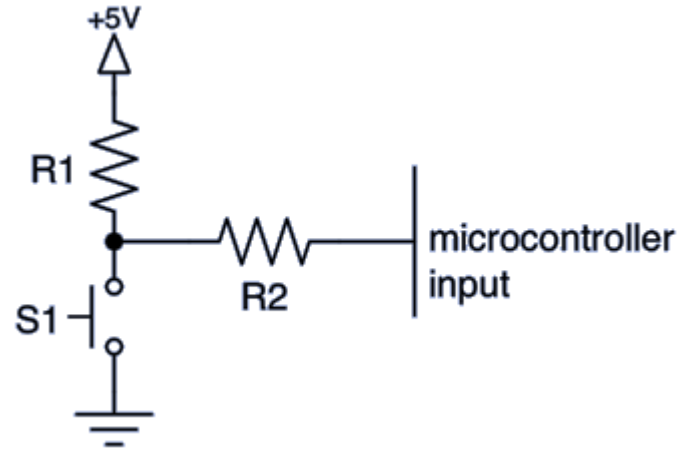
- Bootloader System Review
- Start-up Branching
- Startup Checks
- Assembling the Image
- Resetting the Image



Bootloader System Review



Startup Branching



Advantages	Disadvantages
Code can be in assembly	Susceptibility to start-up noise
Branch is executed quickly	Dedicated GPIO
Very simple implementation	Accidental bootloader entry

Startup Branching

Example:

Assembly branch code

```
brclr $0259, $01, GoBoot      ; if PP0 == 0 then start the boot-loader
                               ; if PP0 == 1 then start the application

ldd  AppResetVect            ; Load the Application Reset Vector
ldx  AppResetVect

jmp  0,x                      ; jump to the application

GoBoot:
lds  #StackTop
jmp  main ;
```

Presented by:

Startup Branching

What is the potential flaw with the previous branch code?



Startup Branching

Example:

Checking the reset vector

```
brclr $0259, $01, GoBoot    ; if PPO == 0 then start the boot-loader
                             ; if PPO == 1 then start the application
ldd  AppResetVect          ; Load the Application Reset Vector
cpd  #$ffff                ; Compare it to 0xFFFF
beq  _GoBoot               ; if the application reset vector is not
                             ; available then start the bootloader
ldx  AppResetVect
jmp  0,x                   ; jump to the application

_GoBoot:
lds  #StackTop
jmp  main                   ; Continue Boot-loader startup
```

Presented by:

Startup Branching

- Example “Advanced” Check

```
ldd AppResetVect      ; Load the Application Reset Vector
cpd #$ffff            ; Compare it to 0xFFFF
beq _GoBoot           ; if the application reset vector is not
                     ; available then start the bootloader

ldd EepromProgStatus  ; Read the programmed status byte from eeprom
cpd #'B'              ; Compare it to 'B' for boot-load
beq _GoBoot           ; if Status == 'B' for Boot-loader then jump to
                     ; boot-loader, otherwise continue to the application

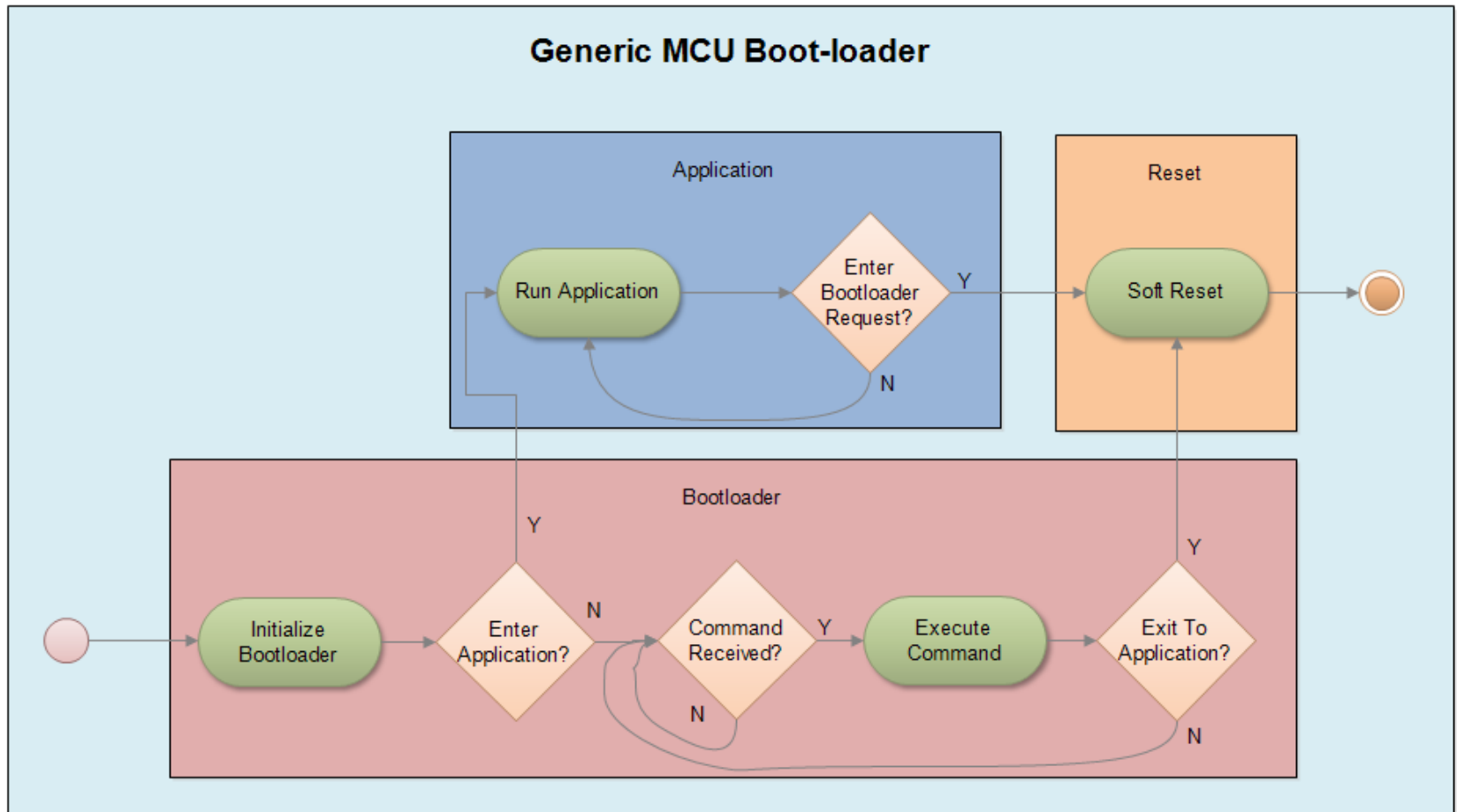
ldx AppResetVect
jmp 0,x               ; jump to the application

_GoBoot:
lds #StackTop
jmp main              ; Continue Boot-loader startup
```

Presented by:

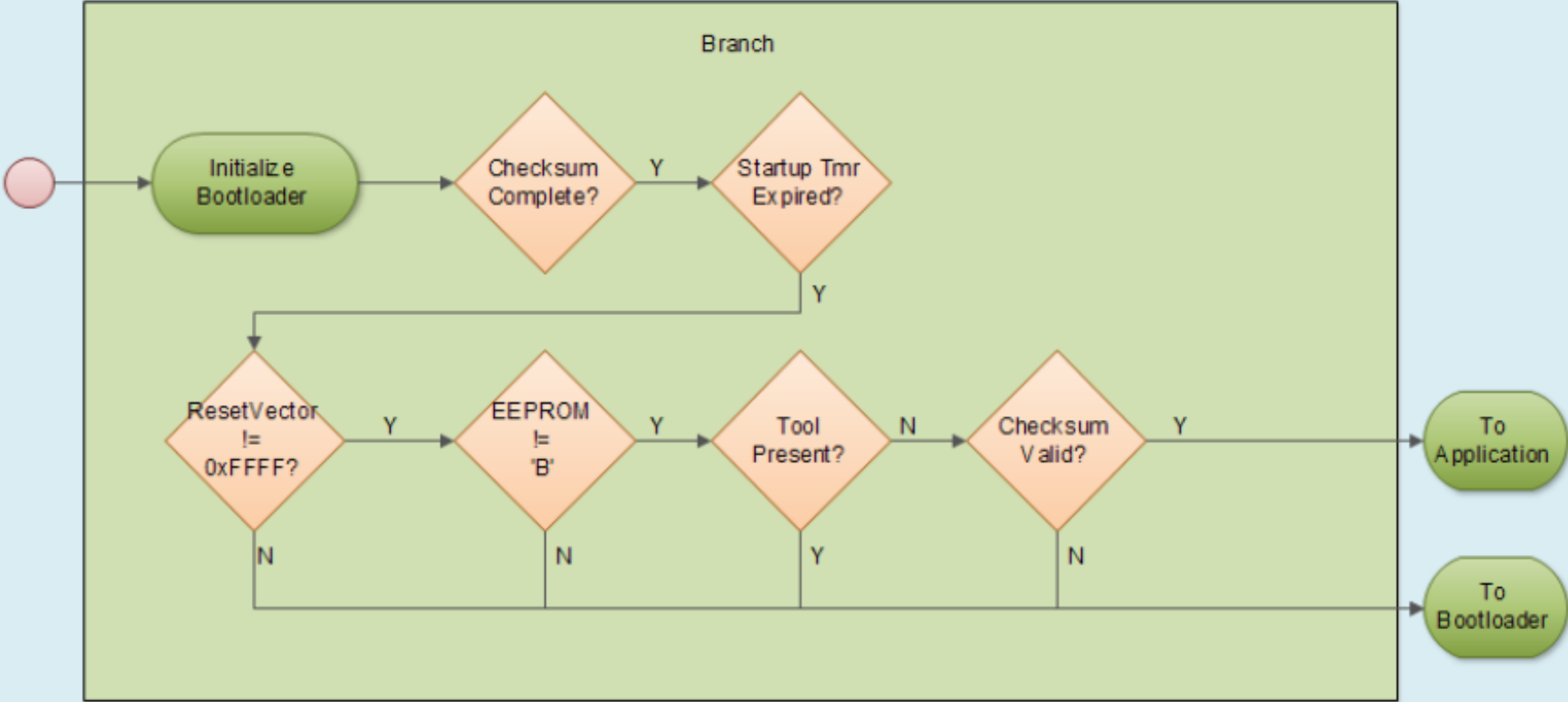
Startup Branching

- Integrating the branch code and the bootloader



Startup Checks

Generic MCU Boot-loader Branch Code



Startup Checks

- Branch Code in C

```
if((Checksum_Complete == TRUE) && (StartUpTmr == EXPIRED))
{
    if((*ResetVector != 0xFFFF) && /* Does app reset vector exist? */
        (Status != 'B') && /* EEPROM status set? */
        (Boot_ToolPresent != TRUE) && /* Tool present? */
        (Checksum_Valid != FALSE)) /* Checksum valid? */
    {
        App_LoadImage();
    }
    else
    {
        Boot_LoadImage();
    }
}
```

Assembling the Image

- Requirements
 - Command driven vs image driven
 - Commands
 - Lock/Unlock Flash
 - Read/Write Configuration
 - Image/Record Data
 - Switch to Application
 - Image Driven
 - Continuously loops through image
 - Completely Autonomous

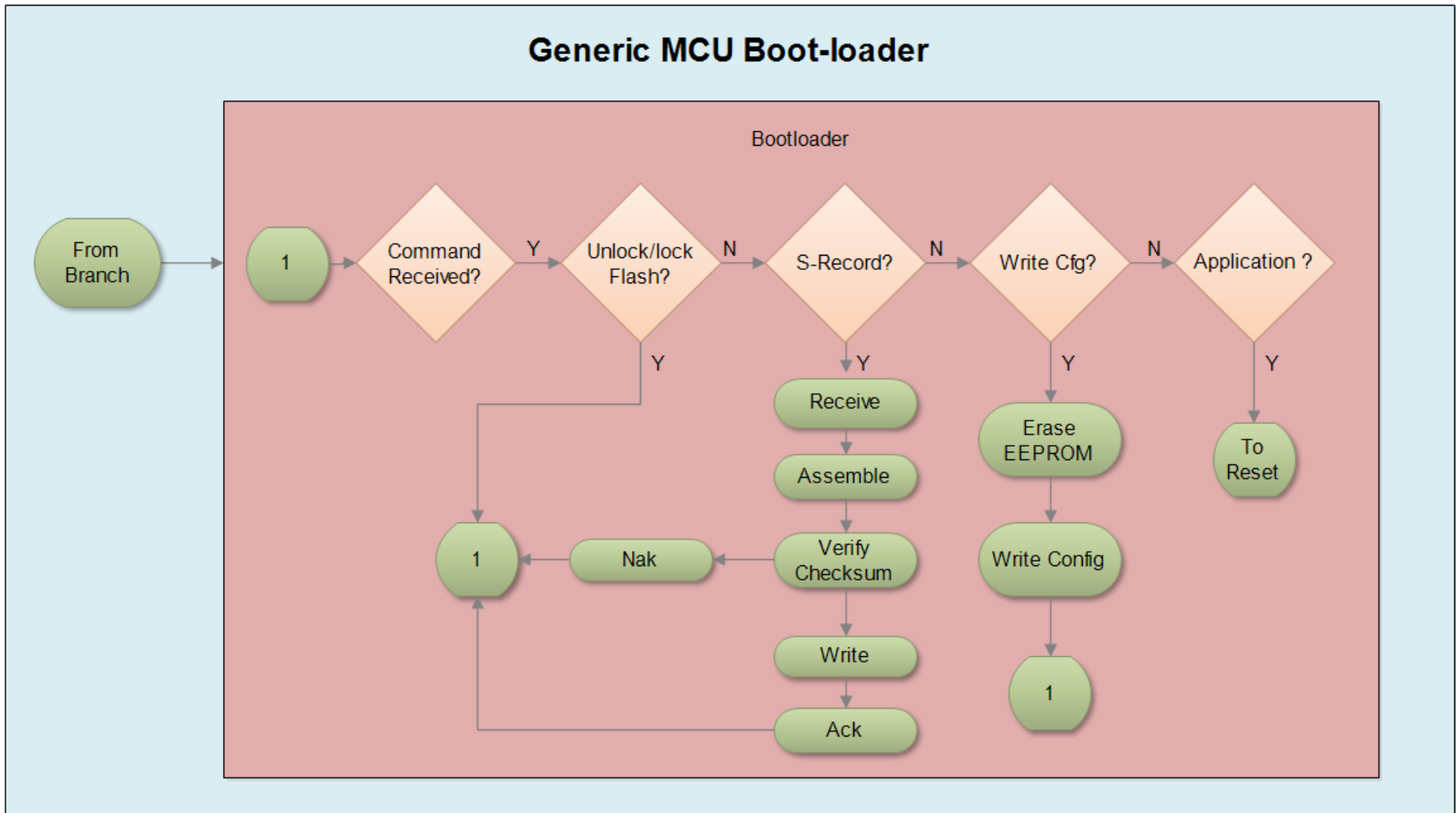


Assembling the Image

- A block of image data is usually larger than can be directly communicated
- Memory region broken up into separate packets
- Packets need to be reassembled and validity checked
- Steps
 - Receive image packets
 - Reassemble into image block
 - Verify Checksum
 - Write
 - Acknowledge
- Repeat until completed



Assembling the Image



Resetting the System

- How to reset the system
 - Watchdog timer
 - Infinite loop
 - Illegal write to register
 - Soft reset command
 - Manual software reset
 - Notify user to power cycle



```
void Wdt_Reset(void)
{
    /* Enter an invalid key to force reset */
    SWT.SR.R = 0x0000FFFF;
}
```

Presented by:

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 Benigo
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/company/JacobBenigo)

EDN : [Embedded Basics](#)

ARM Connected Community

Newsletters

- [Embedded Bytes](#)



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

Training

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