



Embedded Software Design Techniques™

P/N: BE15-6020

Bootloader Design for MCU's Workshop

Page 1 of 5

Course Overview:

The ability to update firmware in the field or over-the-air is a critical component in every embedded system. The complexity and the skills necessary to successfully develop the capability to update firmware in the field are often under-estimated and result in buggy updates or bricked systems. The Bootloader Design for MCU's Workshop walks engineers through every stage of the bootloader design process and provides hands-on examples to accelerate the engineers' abilities and allow them to walk away with skills and knowledge to implement a bootloader on any MCU with any communication interface.

Who Should Attend?

Attendees are engineers who are interested in or will be designing a bootloader for a production product. The course covers the fundamentals and advanced topics in bootloader design. The course is appropriate for engineers with little experience in the area or experienced engineers looking for new techniques and skills. Attendees should understand the fundamentals of the C programming language. Advanced concepts are reviewed as they apply.

Key Take-a-ways:

- Hands-on experience developing a boot-loader
- Bootloader implementation models and how to select the correct one
- Bootloader design, implementation and testing from concept through production
- Design and implementation of a bootloader protocol
- Example code and test hardware platform
- Key concepts related to security, robustness and fast updates
- Troubleshooting and debug techniques
- Best practices for bootloaders
- Techniques to port the boot-loader to other peripherals

Cost:

Single Early Bird: \$750 USD

Single Regular: \$1000 USD

Group Rates: Contact jacob@beningo.com

Course Dates and Locations:

2016 January 15th, Troy Michigan

Contact jacob@beningo.com for additional dates and locations

Contact Information:



Jacob Beningo

jacob@beningo.com

P.O. Box 400
Linden, Mi 48451

Sessions Overview:

Session 1 – Overview of Bootloaders

- Introduction to bootloaders
- Bootloader models
- Bootloader components
- Bootloader system elements
- Features and requirements
- Memory Maps
- Linker file fundamentals
- Software architecture techniques

Session 2 - Bootloader Protocol and GUI Design

- Bootloader protocol design
- Bootloader commands
- Restart strategies
- Partitioning memory
- File formats
- Application record generation
- GUI fundamentals
- Basic implementation model

Session 3 – Bootloader Implementation Part 1

- Bootloader linker file updates
- Branching code development
 - Decision making process
 - Verifying checksums
 - Backdoors
- Robust design techniques
- Bootloader organization
- Understanding time and space requirements
- Bootloader protocol implementation

Session 4 – Bootloader Implementation Part 2

- Vector table relocation
- Record assembly and writing
- Error handling
- Locking down the system

- Testing a bootloader system
- Debugging techniques

Session 5 – Advanced Techniques

- Handling multiple application spaces
- Updating the bootloader
- Encrypting and Decrypting an image
- Difference generators
- Locating code blocks
- Production deployment

Contact Information:



Jacob Beningo
jacob@beningo.com
P.O. Box 400
Linden, Mi 48451

Exercises Overview:

Exercise 1 – Overview of Bootloaders

- Select a bootloader model
- Define the requirements
- Design boot-loader
- Application Project setup
- Application Building blocks
 - Communication Driver
 - LED Driver
 - Task Scheduler
 - Flash Driver
- Create basic test application

Exercise 2 – Bootloader Protocol and GUI Design

- Bootloader hooks into application code
 - Bootloader commands
 - Reset strategies
- Application record generation
- GUI project setup
- GUI design and command setup
- Bootloader project setup

Exercise 3 – Bootloader Implementation Part 1

- Bootloader linker file setup
- Branching code development
- Bootloader protocol implementation
- GUI updates and integration

Exercise 4 – Bootloader Implementation Part 2

- Final boot-loader commands
- Vector table relocation
- Record assembly and writing
- Error handling
- Locking down the system
- Full bootloader test
- Debugging techniques

Lecturer Background



Jacob Beningo, CSDP. Jacob Beningo is a Certified Software Development Professional (CSDP), chair of the IEEE Consultants Affinity Group, an independent consultant and lecturer who specializes in the design of embedded software for resource constrained and low energy devices. He has successfully completed projects across a number of industries including automotive, defense, medical and space. He enjoys developing and teaching real-time and software development techniques using the latest methods and tools. He blogs for EDN.com about embedded system design techniques and challenges. He is an avid tweeter, a tip and trick guru, a homebrew connoisseur and a fan of pineapple! Jacob holds Bachelor's degrees in Electrical Engineering, Physics and Mathematics from Central Michigan University and a Master's degree in Space Systems Engineering from the University of Michigan. He can be reached at jacob@beningo.com

Additional resources, templates and Jacob's monthly embedded software newsletter can be found at www.beningo.com . Check out his workshops at <http://www.beningo.com/training/>

Click the social media link below to follow Jacob and get more tips and tricks:



Blogs can be found at the following sites by clicking the image:





Testimonials

“Jacob Benigo conducted a 5-session hands-on class on the fundamentals of microcontrollers for EETimes University, that was sponsored by STMicroelectronics. While the “fundamentals” approach was familiar, we added the unique hands-on wrinkle and Jacob proved to be an excellent instructor. He adapted seminar material we supplied and patiently walked the on-line class through lessons while contributed to their learning about the valuable features of the provided STM32 microcontroller and Discovery Kit. His success in leading the class was amply demonstrated in the outstanding participant engagement during the sessions and in their comments and feedback afterwards.”

-- Michael Markowitz, Director Technical Media Relations, STMicroelectronics