INTEGRATED SCIENTIFIC SERVICES
A MEDTECH COMPANY

# MEDICAL GRADE BOOTLOADER FOR ARM MICROCONTROLLER

**Confidentiality – Authenticity – Integrity – Security**

**Firmware updates in the field are a challenge for many active medical devices. Based on several implementations for demanding applications in medical technology, we have developed a generic bootloader core that can be flexibly embedded in medical device firmware and allows secure distribution and installation of such firmware updates.**
**By using the ISS Bootloader, you can save a lot of development time in a project and add value in the area of secure remote updates.**

## Summary

The ISS Bootloader is part of a platform for the secure distribution of software and firmware on medical devices. The Bootloader implements the functionality for installation, updates and protection of the firmware against modification and reverse engineering. Each time the device is started, the Bootloader performs an integrity check of the installed firmware. With accordingly designed hardware, the patient risk caused by age-related, environmental and malicious modification of the firmware can be minimized.
The Bootloader is primarily used in systems consisting of a host computer (PC) and one or more microcontrollers. The Bootloader and the respective application firmware implement a proprietary RPC interface via which they can be controlled by an application software on the host computer.

### Functions and Properties

The Bootloader has been developed to meet the following requirements:
- Installation of the firmware during production
- Update of the firmware and the Bootloader in the field (In-Application Programming)
- Integrity and authenticity check of the firmware at startup
- Protection of the device against the installation of unauthorized firmware
- Protection against the installation of firmware on unauthorized devices
- Protection against modification of the firmware or the firmware update
- Protection against reverse engineering of the firmware or the firmware update
- Resistance to errors / preventing that the device becomes useless or at least ensuring the recoverability of the device in case of failed updates (e.g., in the event of a power failure)
- Resistance to attacks
- Limiting the damage of a successful attack (e.g., reading of keys from the microcontroller)

Optional:
- Downgrade prevention – Preventing installation of older firmware versions
- Device-specific firmware – The firmware is linked to the hardware of a device (e.g., via the unique identifier of the microcontroller)
- Production keys – Ensuring that devices or components which, e.g., have “fallen from the back of a lorry” on their way from the electronics manufacturer to the assembler cannot be used

### Level of Development

ISS offers a finished, licensable bootloader solution as well as customer-specific adapations and developments as a service.

### Verification and Integration Verification

ISS AG is ISO 13485 certified and the Bootloader has been developed, documented and verified in accordance with IEC 62304.
For integration into a customer product, ISS supplies the integration plans and protocols, which are adapted by the customer or ISS to the respective product.

**Your contact**

Andreas Müller
Head of Software Development

T  +41 32 513 67 83
andreas.mueller@iss-ag.ch

INTEGRATED SCIENTIFIC SERVICES
A MEDTECH COMPANY

## Technical Details

**Supported Hardware Platforms**
At the beginning, models of the STM32F4xx (Cortex-M4) family by STMicroelectronics will be supported. Support will be extended to other controller families from the same manufacturer as needed. The plan is to support microcontroller families based on ARMv7-M, ARMv7e-M and possibly ARMv6-M architectures providing the required hardware resources (RAM, ROM). Activities are currently underway to support Texas Instruments C2000 ™ microcontrollers.

**Hardware Requirements**
On-Chip Flash Memory: four independent regions are needed in flash memory; the regions must be erasable and writable independently of one another, and provide sufficient space for the Bootloader components and application firmware.
These are at least:

| | |
|---|---|
| Region 1 | 4kB (Pre-Bootloader) |
| Region 2 | 100kB (Bootloader primary) or 16…32kB for variants with adapted cryptography |
| Region 3 | 100kB (Bootloader secondary) or 16…32kB for variants with adapted cryptography |
| Region 4 | Given by application firmware |
| On-Chip SRAM | 32kB RAM and more (depending on functionality) |

**Bootloader Update**
In addition to the application firmware, the Bootloader and the cryptographic keys can also be safely updated in the field.
In order for the device to continue to be usable, or at least recoverable, in the event of a failed update, there are two copies of the Bootloader on the target system. A Bootloader can only overwrite the other copy, which guarantees a fallback in case of an error.

**Asymmetric cryptography**
By using asymmetric cryptography to encrypt and decrypt the firmware, as well as generating and verifying digital signatures, keys extracted by an attacker from one device cannot be used directly for further attacks on other devices or the overall system. A key which, e.g., for the verification of a signature is in the memory of the device, cannot be used to generate valid signatures.

**Interfaces**
The Bootloader functions can be used via a proprietary RPC protocol (Remote Procedure Call). The same protocol can also be used in the application firmware, which allows an optimal integration of the Bootloader and the application. Software libraries for C, C # and C ++ are available on request.

The RPC protocol can be used across most of the available electronic interfaces and can be configured for different topologies:
- Point to Point – One host (Master) and one microcontroller (Slave): Master → Slave 1
- Daisy Chain – One host (Master) and several microcontrollers connected in series (Slaves): Master → Slave 1 → Slave 2 →...
- Star – One host (Master) with multiple physical point-to-point connections
- Network – One host (Master) with several logical point-to-point connections, e.g., via an existing TCP/IP network, an I2C or SPI bus with several addressable slaves.

**Limitations**
Since in most cases the Bootloader, just like the application firmware itself, is stored in the normal program memory, it is recommended to additionally protect it against malicious modification (e.g. via JTAG) by means of hardware measures. Most microcontrollers already include different possibilities.