CycloneTCP is a dual IPv4/IPv6 stack dedicated to embedded applications. CycloneTCP conforms to RFC standards and offers seamless interoperability with existing TCP/IP systems. By supporting IPv6, CycloneTCP eases deployment of next-generation Internet. The stack is distributed as a full ANSI C and highly maintainable source code.

HTTP	HTTP/2	MQTT	MQTT-SN	CoAP	FTP	7 - Application
SMTP	SNTP	DNS	NetBIOS	SNMPv3	TFTP	
WebSocket		mDNS	DNS-SD	DHCP	DHCPv6	
Socket						5 - Session
TCP		UDP			RAW	4 - Transport
IPv4			IPv6			3 - Network
ARP	Auto-IP		NDP	SLAAC		
ICMP	IGMPv2		ICMPv6	MLDv1		
Ethernet	Wi-Fi	PPP	USB/RNDIS	G3-PLC		2 - Data Link

Main Features

- Dual stack (IPv4 and/or IPv6)
- Built-in support for multiple network interfaces
- Flexible memory footprint (built-time configuration to embed only the necessary features)
- Configurable memory model (static memory pool or heap memory allocation)
- Portable architecture (no processor dependencies)
- Straightforward port to any RTOS
- Highly maintainable source code
- Debugging and trace functionality to ease development and integration
- BSD style socket API
- Blocking/non-blocking socket operation and event-driven functions (select and poll)
- Efficient data transfer through zero copy
- Well-crafted TCP module with selective acknowledgement (SACK) and congestion control
- Raw socket interface
- IP fragmentation and reassembly support
- Support for virtual interfaces (multiple MAC addresses per physical interface)
- Support for multi-homed hosts (multiple IPv4 addresses per interface)
- Ethernet port multiplication using VLAN tagging (SMSC switches) or tail tagging (Micrel switches)
- VLAN support (802.1Q and 802.1ad)
- USB Device RNDIS class driver (for STM32 microcontrollers)

© 2010-2026 Oryx Embedded

Supported Protocols

- LLDP agent compliant with 801.1AB-2005 (TX-only, RX-only and TX/RX modes supported
- DHCP client and server
- Auto-IP (dynamic configuration of IPv4 link-local addresses)
- DHCPv6 client and relay agent
- SLAAC (IPv6 stateless address autoconfiguration)
- Multicast source filtering (IGMPv3 host and MLDv2 node)
- DNS client
- NetBIOS client and responder
- LLMNR client and responder
- mDNS client and responder
- DNS-SD responder (DNS-based service discovery)
- FTP / FTPS client and server (implicit TLS and explicit TLS modes supported)
- TFTP client and server
- HTTP / HTTPS client
- HTTP / HTTPS server with SSI, CGI scripting and WebSocket support
- HTTP/2 client (including HPACK compression, server push and https scheme)
- SMTP client
- MQTT v3.1.1 client (TCP, TLS, WebSocket and secure WebSocket transport layers supported)
- MQTT-SN client (UDP and DTLS transport layers supported)
- CoAP client (DTLS-secured CoAP, Observe and Block-Wise Transfers supported)
- CoAP server (DTLS-secured CoAP supported)
- SNMP agent (SNMPv1, SNMPv2c and SNMPv3 supported)
- Remote management of SNMP users and access rights (SNMP-USM-MIB and SNMP-VACM-MIB)
- Standard MIBs: MIB-II, IF-MIB, IP-MIB, TCP-MIB, UDP-MIB, SNMPv2-MIB, LLDP-MIB
- SNTP client
- NTS client (Network Time Security)
- Echo server
- Modbus/TCP client and server (Modbus/TCP security supported)
- Syslog client
- WebSocket client and server (WebSocket connections tunneled over SSL/TLS supported)
- PPP (Point-to-Point Protocol)

Data Link Layer (PPP)

- RFC 1332: The PPP Internet Protocol Control Protocol (IPCP)
- RFC 1334: PPP Authentication Protocols
- RFC 1661: The Point-to-Point Protocol (PPP)
- RFC 1662: PPP in HDLC-like Framing
- RFC 1994: PPP Challenge Handshake Authentication Protocol (CHAP)
- RFC 2472: IP Version 6 over PPP

Network Layer (IPv4)

- RFC 791: Internet Protocol Specification
- RFC 792: Internet Control Message Protocol Specification
- RFC 815: IP Datagram Reassembly Algorithms
- RFC 826: Ethernet Address Resolution Protocol
- RFC 1112: Host Extensions for IP Multicasting
- RFC 1122: Requirements for Internet Hosts - Communication Layers
- RFC 2113: IP Router Alert Option
- RFC 2236: Internet Group Management Protocol, Version 2
- RFC 3376: Internet Group Management Protocol, Version 3
- RFC 3927: Dynamic Configuration of IPv4 Link-Local Addresses
- RFC 4541: Considerations for IGMP and MLD Snooping Switches
- RFC 5227: IPv4 Address Conflict Detection
- RFC 9776: Internet Group Management Protocol, Version 3

Network Layer (IPv6)

- RFC 2460: Internet Protocol, Version 6 (IPv6) Specification
- RFC 2464: Transmission of IPv6 Packets over Ethernet Networks
- RFC 2710: Multicast Listener Discovery (MLD) for IPv6
- RFC 3484: Default Address Selection for Internet Protocol version 6 (IPv6)
- RFC 3493: Basic Socket Interface Extensions for IPv6
- RFC 3590: Source Address Selection for MLD Protocol
- RFC 3678: Socket Interface Extensions for Multicast Source Filters
- RFC 3810: Multicast Listener Discovery Version 2 (MLDv2) for IPv6
- RFC 4291: IP Version 6 Addressing Architecture
- RFC 4294: IPv6 Node Requirements
- RFC 4443: Internet Control Message Protocol Version 6 (ICMPv6) Specification
- RFC 4861: Neighbor Discovery for IP version 6 (IPv6)
- RFC 4862: IPv6 Stateless Address Autoconfiguration
- RFC 6106: IPv6 Router Advertisement Options for DNS Configuration
- RFC 9777: Multicast Listener Discovery Version 2 (MLDv2) for IPv6

Transport Layer

- RFC 768: User Datagram Protocol
- RFC 793: Transmission Control Protocol
- RFC 2018: TCP Selective Acknowledgment Options
- RFC 5681: TCP Congestion Control
- RFC 6298: Computing TCP's Retransmission Timer
- RFC 6528: Defending against Sequence Number Attacks
- RFC 9293: Transmission Control Protocol (TCP)

Application Layer

- RFC 959: File Transfer Protocol (FTP)
- RFC 1035: Domain Names – Implementation and Specification
- RFC 1157: A Simple Network Management Protocol (SNMP)
- RFC 1213: Management Information Base for Network Management of TCP/IP-based internets (MIB-II)
- RFC 1350: The TFTP Protocol (Revision 2)
- RFC 1769: Simple Network Time Protocol (SNTP)
- RFC 1905: Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)
- RFC 1945: Hypertext Transfer Protocol - HTTP/1.0
- RFC 2131: Dynamic Host Configuration Protocol
- RFC 2132: DHCP Options and BOOTP Vendor Extensions
- RFC 2616: Hypertext Transfer Protocol - HTTP/1.1
- RFC 2617: HTTP Authentication: Basic and Digest Access Authentication
- RFC 2818: HTTP Over TLS
- RFC 2863: The Interfaces Group MIB
- RFC 3164: The BSD syslog Protocol
- RFC 3207: SMTP Service Extension for Secure SMTP over Transport Layer Security
- RFC 3315: Dynamic Host Configuration Protocol for IPv6 (DHCPv6)
- RFC 3410: Introduction and Applicability Statements for Internet Standard Management Framework
- RFC 3411: An Architecture for Describing SNMP Management Frameworks
- RFC 3412: Message Processing and Dispatching for the SNMP
- RFC 3413: Simple Network Management Protocol (SNMP) Applications
- RFC 3414: User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)
- RFC 3415: View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)
- RFC 3418: Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)
- RFC 3584: Coexistence between Version 1, Version 2, and Version 3 of SNMP Framework
- RFC 3646: DNS Configuration options for DHCPv6
- RFC 3826: AES Cipher Algorithm in the SNMP User-based Security Model
- RFC 4022: MIB for the Transmission Control Protocol (TCP)
- RFC 4113: MIB for the User Datagram Protocol (UDP)
- RFC 4293: MIB for the Internet Protocol (IP)
- RFC 4795: Link-local Multicast Name Resolution (LLMNR)
- RFC 4954: SMTP Service Extension for Authentication
- RFC 5321: Simple Mail Transfer Protocol
- RFC 6455: The WebSocket Protocol
- RFC 6528: Defending against Sequence Number Attacks
- RFC 6762: Multicast DNS
- RFC 6763: DNS-Based Service Discovery
- RFC 7252: The Constrained Application Protocol (CoAP)
- RFC 7540: Hypertext Transfer Protocol Version 2 (HTTP/2)
- RFC 7541: HPACK Header Compression for HTTP/2
- RFC 7641: Observing Resources in the Constrained Application Protocol (CoAP)
- RFC 7860: HMAC-SHA-2 Authentication Protocols in the User-based Security Model
- RFC 7959: Block-Wise Transfers in the Constrained Application Protocol (CoAP)
- RFC 8915: Network Time Security for the Network Time Protocol

IEEE

- IEEE Std 802.1AB-2005: IEEE Standard for Local and metropolitan area networks - Station and Media Access Control Connectivity Discovery

Supported Processors

- ARM Cortex-M3
- ARM Cortex-M4
- ARM Cortex-M7
- ARM Cortex-M33
- ARM Cortex-M55
- ARM Cortex-M85
- ARM Cortex-R4
- ARM Cortex-A5
- ARM Cortex-A7
- ARM Cortex-A8
- ARM Cortex-A9
- ARM Cortex-A55
- RISC-V
- MIPS M4K
- MIPS microAptiv / M-Class
- Infineon TriCore AURIX
- PowerPC e200
- Coldfire V2
- RX600
- AVR32
- Xtensa LX6

Supported Operating Systems

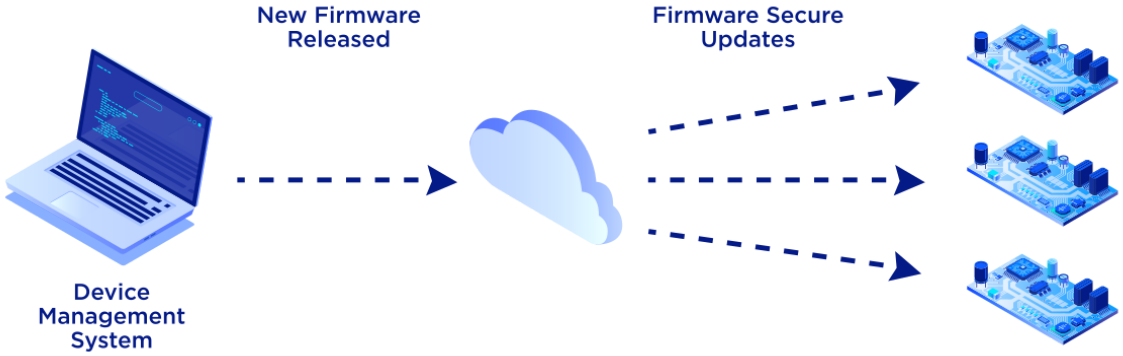
- Amazon FreeRTOS
- SafeRTOS
- ChibiOS/RT
- CMSIS-RTOS
- CMSIS-RTOS2
- CMX-RTX
- Keil RTXv4 and RTXv5
- Micrium µC/OS-II and µC/OS-III
- Eclipse ThreadX
- PX5 RTOS
- Segger embOS
- TI-RTOS (SYS/BIOS)
- Zephyr RTOS
- Bare Metal programming (without RTOS)

Supported Compilers / Toolchains

Toolchain / IDE	Compiler
Makefile	GCC
AC6 System Workbench for STM32 (SW4STM32)	GCC
Atollic TrueSTUDIO	GCC
Espressif ESP-IDF	GCC
HighTec Toolset for TriCore	GCC
IAR Embedded Workbench	EWARM, EWRX
Infineon DAVE	GCC
Keil MDK-ARM	ARM Compiler v5, ARM Compiler v6 (CLANG)
Microchip Studio (Atmel Studio)	GCC
Microchip MPLAB X	GCC, XC32
Microsoft Visual Studio	MSVC
NXP MCUXpresso	GCC
NXP S32 Design Studio (S32DS)	GCC
Renesas e2Studio	GCC, CC-RX
Segger Embedded Studio	GCC
ST STM32CubeIDE	GCC
Tasking VX-Toolset	VX-Toolset for TriCore

CycloneBOOT

CycloneBOOT is a secure firmware update solution targeting 32-bit microcontrollers. It provides a reliable and secure method for booting and updating the firmware of your device. Tailored to work with a variety of ARM Cortex-M based microcontrollers, CycloneBOOT ensures a seamless boot process every time.



Main Features

CycloneBOOT includes multiple security measures to protect against external threats and unauthorized access. It features an advanced verification process that can be enabled to check the integrity of firmware update images before processing. It can also handle encrypted firmware update images and optionally supports authentication or digital signatures to verify incoming updates. Additionally, boot-time application firmware verification using RSA or ECDSA signatures for Secure Boot can be activated as needed.

CycloneBOOT offers versatile support for various memory partitioning configurations. It accommodates different MCU internal flashes, whether used with or without external flash. It can also enable In-Application Programming (IAP) with dual-bank flash MCUs. This flexibility allows the boot process to be tailored to different scenarios depending on the desired levels of security and reliability.

CycloneBOOT includes fallback and anti-rollback support to ensure that your device is always able to boot, even in the event of a failure. The fallback feature allows user to revert to a previous firmware if the latest firmware contains bugs or serious issues. The anti-rollback feature prevents unauthorized downgrades of the current firmware, ensuring that only latest versions of the firmware are used. This helps to protect against potential vulnerabilities that may exist in older firmware versions.

CycloneBOOT is protocol agnostic, allowing firmware updates to be performed using various communication channels such as Ethernet, USB, UART, Wi-Fi, Cellular Modem, etc. It features a simple and intuitive interface, making it easy to integrate alongside your existing firmware and your favorite protocol.

© 2010-2026 Oryx Embedded

Detailed Feature List

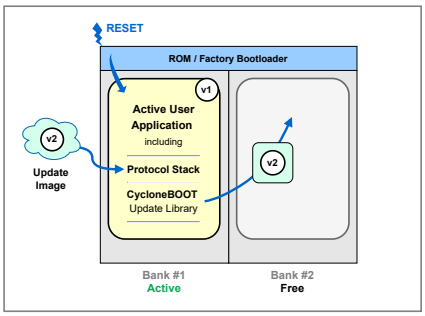
- Secure firmware update solution for 32-bit MCUs (ARM Cortex-M)
- Support for various MCU internal flashes and external flashes
- Support for In-Application Programming (IAP) when using MCUs with dual-bank flash capabilities
- Update image verification using MD5/CRC32/SHA-1/SHA-2 integrity checks, HMAC authentication, or RSA/ECDSA signatures
- Support for encrypted update images using AES-CBC
- Boot-time application firmware verification at every startup using CRC32/SHA-1/SHA-2 integrity checks, or RSA/ECDSA signatures for Secure Boot
- Anti-rollback support (prevents installing a previous firmware version)
- Fallback support (restores previous firmware version if needed)
- Can be integrated in client or server operation
- Can run alongside a RTOS or in Bare Metal

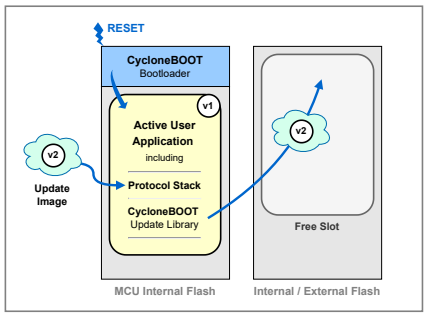
Modular Architecture

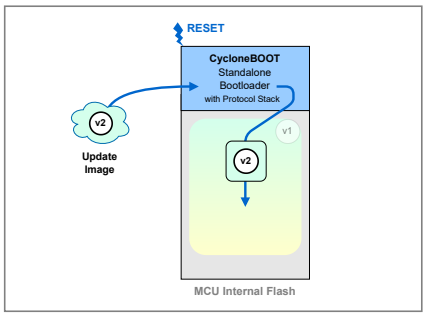
CycloneBOOT solution provides modular architecture whose components can be enabled independently or integrated together, depending on the desired update scenario:

- **CycloneBOOT Update Library** integrated within the user application, can process incoming update images, including reception, validation, and installation or storage. It can also enforce anti-rollback protection.
- **CycloneBOOT Bootloader** can install update images and supports advanced features such as application firmware verification at every startup (Secure Boot), fallback mechanisms, and external flash management. The optional multi-stage approach provides an immutable first-stage bootloader that enables the second stage to be updated.
- **CycloneBOOT Standalone Bootloader** manages the entire firmware update process, including reception, validation, and installation. In this case, the bootloader also includes a predefined protocol.

We can help you compare these features and various update scenarios, and provide a custom demo tailored to your needs. As always with ORYX, the full source code is available for evaluation!

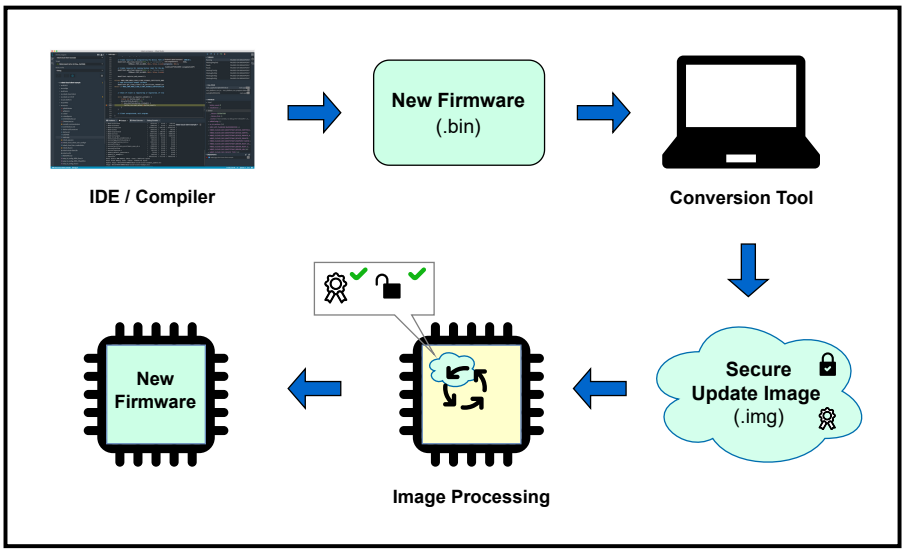






ImageBuilder Tool

ImageBuilder is a cross-platform CLI utility (Windows and Linux) for building secure firmware update images, with support for encryption, integrity tags, authentication tags, and signatures. It can also generate signature keys.



Easy to Use with TCP/IP Protocols

With our experience on TCP/IP protocols we can provide you with a ready-to-use Ethernet Bootloader by bundling CycloneBOOT with CycloneTCP (TCP/IP stack), CycloneSSL (TLS library) and CycloneSSH (SSH library). You could for example fetch the new firmware image over Internet (LAN, Wi-Fi, Cellular Modem) using protocols like:

- TFTP / FTP / FTPS
- HTTP / HTTPS
- MQTT / MQTTS
- SFTP / SCP ...

Supported Microcontrollers

- STM32L4
- STM32F4
- STM32F7
- STM32H7
- STM32U5
- STM32H5
- ATSAME54

Supported Toolchains / Compilers

Toolchain / IDE	Compiler
CMake	GCC
Makefile	GCC
IAR Embedded Workbench	EWARM
Keil MDK-ARM	ARM Compiler v5, ARM Compiler v6 (CLANG)
Microchip Studio	GCC
ST STM32CubeIDE	GCC