

.NET Micro Framework

Software Development Kit Version 4.2

TECHNICAL DATA SHEET

September 2011

.NET MICRO FRAMEWORK FEATURES AND CAPABILITIES

Targeted to Small Devices	Both the CLR and the managed libraries provided in the SDK are carefully designed and crafted to work on small, inexpensive devices.
Programming Language	C#. Other programming languages are not supported. Supports v1.1. CLI ECMA standard, with the exception of the native pointer types and relative conversions. Because generics were introduced in CLI v2.0, the .NET Micro Framework does not support them.
Libraries	Implementations of several standard .NET namespaces including classes for threading, containers, reflection, TCP/IP, HTTP(S), SSL, Cryptographic primitives, IO, XML parsing, Resources and Text. Also implements a number of non-standard namespaces designed specifically for embedded programmers to use on small, resource-constrained hardware. These namespaces include classes for hardware, low-level communication protocols, cryptography, graphics, touch panels, and Web Services.
Execution Model	CLR runs natively on hardware so an operating system is not required. However, the framework can be ported so that it runs as a process on top of an operating system. Supports managed multi-threading.
Debugging	Supports the standard Visual Studio debugging tools for managed applications such as breakpoints, watches, stepping through code, and so forth. Debugging transport can be UART/USART, USB, or TCP/IP, depending on the hardware.
Connectivity	UART/USART, I ² C, SPI, USB, support for adding 802.11
Networking and Web Services	TCP/IP, HTTP, SSL, DPWS (Device Profile for Web Services, i.e. SOAP on HTTP). Wi-Fi configuration for most common 802.11 parameters supported. Time Synchronization is supported though SNTP.
Cryptography	Provides standard encryption and decryption techniques. Also enables you to extend the cryptographic capabilities for customized cryptography by using PKCS #11 tokens.
(Remote) Firmware Update	Supports remote firmware updates, assembly updates and other OEM

	defined updates (such as key provisioning). The framework is extensible to allow the OEM to determine the best solution for their device.
Globalization	Supports standard .NET globalization techniques.
File System	Block storage API and file system API enable your applications to store information on block storage devices such as SD cards. The SDK provides a FAT-compatible file for both FAT32 and FAT16 systems. However, you can also create one of your own.
Graphics and LCD	<p>Can handle graphics natively or from the managed application. Managed applications use the Windows Presentation Foundation model for dynamic layout and event dispatching. Native-level bitmaps are based on and include the basic primitives for drawing ellipsis and rectangles in a customizable source code format. Supports using bitmaps as resources in the TinyCLRBitmap (an internal format), BMP, JPG and GIF formats. The following bitmap types are supported.</p> <ul style="list-style-type: none"> • 24bpp RGB, • 32bpp RGB, • 48bpp RGB, • 16bpp RGB (565 format) • 15bpp RGB (555) format; • 1bpp monochrome <p>All color bitmap formats are silently converted to 16bpp RGB (565) at build time for the managed application. 1bpp monochrome bitmaps are compressed at build-time. When loading bitmap data at run-time:</p> <ul style="list-style-type: none"> • Only 16bpp (565 format) and 24bpp RGB formats are supported by BitmapImageType.Bmp. • The only supported mechanism for loading 1bpp monochrome bitmaps is embedding them as bitmap resources at compile time. • Bitmap size is unlimited.
Touch Screen	Standardized interface for hardware that uses a touch screen and stylus, including gestures and multi-touch support.
Graphical User Interfaces (GUIs)	Offers a subset of the Windows Presentation Foundation (WFP) classes.
Managed Drivers	Managed C# applications can communicate with peripheral devices through standard connections such as GP I/O, SPI, I ² C, OneWire, Serial, USB and so forth.
Versioning	Managed applications can query the HAL, runtime, and assembly version number to ensure that the application is installed on the correct version of the firmware.
Emulation	Powerful hardware emulator that you can enhance and extend to suit your needs when creating embedded systems for hardware devices. Sample emulator provided.
Real-Time Hardware Access	No real-time hardware access provided. Nevertheless, you can port it as a process on top of a real-time operating system and thereby access real-time features.

HARDWARE REQUIREMENTS AND AVAILABILITY

SUPPORTED PROCESSORS

Currently the framework runs on these 32-bit processor cores.

- ARM v4,
- ARM Thumb
- Thumb-2
- Blackfin

- SH2/2A

DEVELOPMENT BOARDS

The porting kit provides sample ports available for the development boards listed below. These sample ports include the source code.

- Atmel SAM9261-EK development board (ARM core AT91SAM9261)
- Atmel SAM9RL64-EK development board (ARM core AT91SAMRL64)
- Atmel SAM7X-EK development board (ARM core AT91SAM7X512)
- Phytex phyCORE-ARM7/LPC2294 (Rapid Development Kit PCM023, ARM core LPC2294 by NXP)
- Embedded Artists EA-LPC2478 development board (ARM LPC2478 by NXP)
- Crossbow MOTE2 platform from Crossbow (for Marvell PXA271 processor)
- Freescale i.MXS development kit (Freescale MC9328MXS)board for i.MXS processor
- Renesas SH7264 RSK development board (SH2A core)
- Renesas SH7216 RSK development board (SH2A core)
- Renesas SH7619 EVB development board (SH2 core)

Please see the PK distribution and documentation for an up-to-date list.

MEMORY REQUIREMENTS

The memory requirements for the .NET Micro Framework depend entirely on which features you use. In its smallest form, the framework requires 64 Kb of RAM and 256 Kb of flash memory.

SUPPORTED TOOLSETS

To write a managed C# applications that run your hardware, you can use Visual Studio. The framework is completely compatible with both the commercial version and Visual C# Express Edition. Therefore, if you would like to evaluate the .NET Micro Framework at no cost, you can download and install both Visual C# Express Edition and the Platform SDK for free.

The .NET Micro Framework is fully integrated into Visual Studio so it enables you to use the normal development and debugging features that you commonly use for writing other types of applications. For example, it enables you to set breakpoints, watch variables change, step through code, and so forth.

SUPPORT AND COMMUNITY

For more information on development hardware, support issues, the .NET Micro Framework development community, and more, see <http://www.microsoft.com/netmf/hardware/default.aspx> and <http://www.netmf.com> for community development.