YNQ:
A Portable SMB Solution for Embedded Systems

A Visuality Systems White Paper

visualitynq.com

Table of Contents

Embedded Market Overview **4**
Challenges **5**
No portable & standard SMB solution 5
Vulnerabilities and limited resources 6
Long product life cycles 6
Real-time constraints 6
Open source alternatives 6
Other file transfer protocols 6
Limited portability 7
SMB1 risks 7
The Solution **7**
Why SMB? 7
Visuality Systems YNQ: Comprehensive SMB Client and Server 7
Use Cases **8**
Scan to Folder 8
Automotive Manufacturing Floor 9
SMB3 shield for WinCE systems 9
YNQ Architecture **9**
Porting 10
Integration 10
Features **12**
QUIC Add-on **13**
Compliance and Connectivity **13**
Summary **14**

Embedded Market Overview

The embedded systems market is worth hundreds of billions of US Dollars, and is expected to grow by over 6% in a decade until 2032 (source: Global Market Insights).

Embedded software is a critical component of an overarching embedded system architecture for devices that runs on low footprint, and low RAM, Real Time Operating Systems (RTOS). From home appliances to on-board aircraft networks, robotics to medical equipment, automotive to smart watches, ATMs to printers, there are many types of embedded software applications running on different hardware, each with its own size, shape and custom requirements.

There is something, however, that unites most of them, and that is the need for network connectivity. This is because, nowadays, devices are not isolated anymore. For instance, they may need to save their jobs on a remote server or may want to expose their files to the outer world.

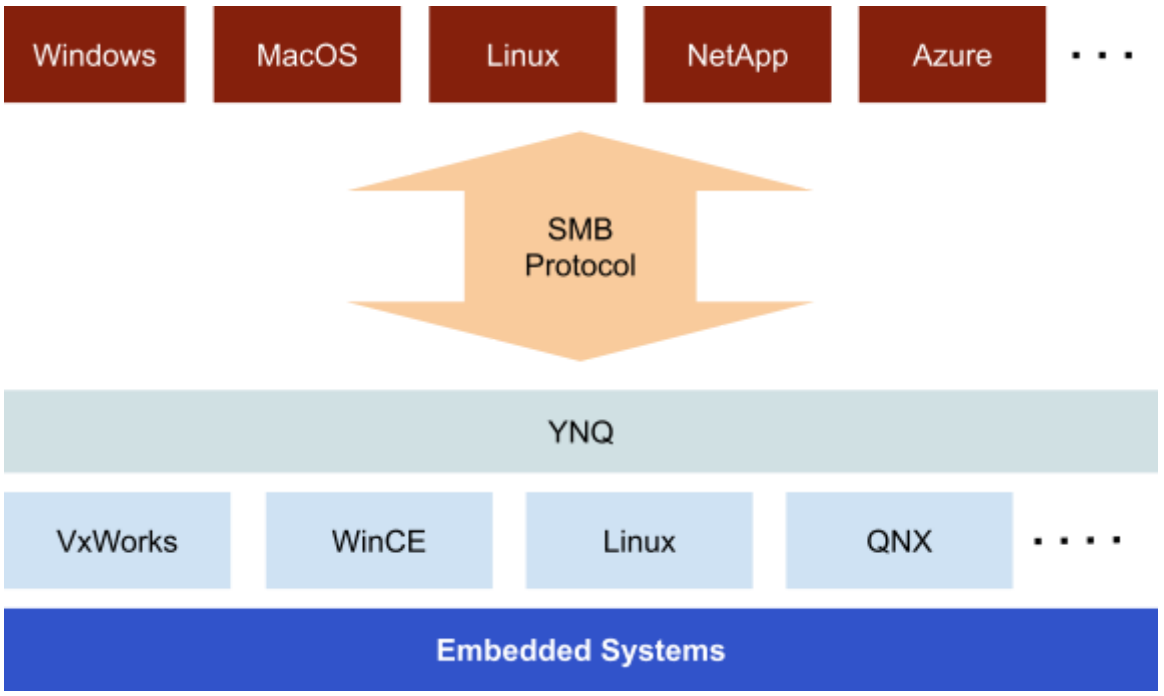

Figure 1: YNQ possible connectivity

Remote file access and print services are just two of the most common embedded software use cases. The field continues to expand as technology evolves and new applications emerge. Relevant industries and applications include:

- MFPs (multifunctional printers) - scan & share over a distributed file system.
- Telecommunications - storage and flash drives connected to routers and other network devices.

- Aerospace and defense - avionics systems, UAVs, missile guidance systems, radars, GPS navigation through remote maps.
- Automotive - engine management, anti-lock braking systems (ABS), infotainment, advanced driver-assistance systems (ADAS).
- Smart factory & industry 4.0 - manufacturing automation, robots & machinery control, NC & CNC, process monitoring, operations optimization.
- Medical devices - monitor & assist patients through imaging equipment, pacemakers, insulin and infusion pumps.
- IoT & consumer electronics - smartphones, tablets, smart TVs, smart home devices, digital cameras, and wearable devices.
- Energy and utilities - smart grid infrastructure, renewable energy systems, SCADA monitoring & control of utility networks, HMI.
- Financial systems - banking terminals, ATMs, point-of-sale (POS) systems, and electronic payment devices.
- Transportation and logistics - fleet management systems, GPS navigation, toll collection and logistics optimization.
- Gaming consoles - set-top boxes, media players, and other entertainment devices.
- Homeland Security (HLS) - access control, surveillance cameras, alarm systems, biometric identification devices.
- Scientific instruments - spectroscopes, microscopes, telescopes, data acquisition systems.

Since a large percentage of devices are communicating with back-end Windows systems, the Server Message Block (SMB) protocol, which is the default standard in Microsoft-based systems, is typically used for this remote connectivity. SMB has been widely adopted in heterogeneous environments involving Unix/Linux, MacOS/iOS, Android and RTOS like VxWorks, QNX, ThreadX, Integrity, Itron and more.

Challenges

No portable & standard SMB solution

Embedded systems are highly diverse and typically lack a native SMB solution. Samba open source is available on Linux and Unix, but it has limited support, if at all, and licensing requirements that force users to share their code changes with the community. This is hardly acceptable in a business competitive and dynamic environment. The complex nature of the SMB protocol and the specific constraints

of RTOS require either in-depth knowledge and experience or a seasoned partner that is able to guarantee quick and prompt support when needed.

Vulnerabilities and limited resources

Embedded systems and ordinary computers encounter similar challenges and security threats like malware, unauthorized access, and data breaches. However, the challenge is bigger for embedded systems because of their limited processing power, memory, and storage. These limitations make it difficult to implement robust protecting measures, encryption algorithms, and secure communication protocols against potential vulnerabilities required to safeguard data.

Long product life cycles

Many embedded systems are in use for extended periods, sometimes spanning decades. Keeping the systems updated over time can be an issue, as security vulnerabilities may emerge and need to be addressed through software upgrades or patches. Embedded systems operate often in isolated or constrained environments, therefore updating the firmware or software on these systems isn't always easy.

Real-time constraints

Certain embedded systems, such as those used in critical infrastructure or automotive applications, have strict real-time requirements. Implementing changes without compromising real-time performance can be problematic.

Open source alternatives

Communities of developers collaborating to write code are very common and contribute to building a healthy ecosystem. However, open source licensing terms require that modifications to the codebase are available to the public, something that a business could hardly accept because it could jeopardize its competitiveness. Another drawback is that support is not ruled by any SLA therefore it can be lacking and, in RTOS environments, even nonexistent. Finally, even if open source can be modular, addressing all the needs might have a cost in terms of an oversized footprint that is not suitable for embedded systems with limited resources.

Other file transfer protocols

File transfer solutions like FTP, HTTP, WebDAV and NFS, must read/write entire files, thus limiting performance compared to SMB. SMB supports file and printer sharing, as well as various operations like file read/write, directory browsing, and file access control from remote, even without actually transferring the file.

Limited portability

Complete portability of firmware developed for embedded systems often requires significant effort and customization, depending on the complexity of the firmware and the differences between the target platforms.

SMB1 risks

The WannaCry attack exploited a vulnerability in the SMB1 protocol, resulting in a widespread ransomware campaign. The incident highlighted the need for timely patching of vulnerabilities and cybersecurity measures. Nevertheless, even if SMB1 is no longer supported by Microsoft, it is still being used by a significant number of embedded systems.

The Solution

Why SMB?

The SMB protocol, formally referred to as CIFS, is a file and printer sharing protocol that serves as the basis for Microsoft's Distributed File System implementation. Contrary to FTP and HTTP, the SMB protocol allows not only copying an entire file, but also grants access to files over the network. File editing, for example, can be executed over SMB without a change in its location. The latest dialect - SMB 3.1.1 - enjoys the highest levels of built-in security, including pre-authentication and encryption for all file operations.

While desktop computers and servers such as Windows and Macintosh natively benefit from SMB connectivity, the situation in the embedded world is more complicated. A device may be developed on top of Linux/Unix, iOS or any RTOS that lacks native SMB support such as VxWorks, ThreadX, Integrity, Itron, or Windows CE and the like.

Visuality Systems YNQ: Comprehensive SMB Client and Server

YNQ is a portable SMB Server and Client solution developed following the Agile methodology. YNQ can be used with, and integrated into, any operating system (OS), device, CPU or compiler according to Microsoft specifications across the whole SMB protocol evolution - from SMB1 to SMB3.1.1, the latest version that includes critical, built-in connectivity and file encryption requirements.

YNQ has 3 modularity levels:

- **High: API/Protocol level (or Frontend) –** API, server, client, NetBIOS
- **Medium: Service level –** Authentication, common, network

- **Low: OS level (or Backend) –** System, user defined, driver

Each level utilizes the levels beneath: the API level utilizes both the Service level and OS level, whilst the Service level utilizes the OS level.

YNQ includes four separate products:

- **Standalone Client** – full SMB client functionality
- **Corporate Client** – full SMB client functionality + ability to register the machine to the corporate Active Directory
- **Standalone Server** – full SMB server functionality
- **Corporate Server** – full SMB server functionality + ability to register the server to the corporate Active Directory + pass-through authentication

Use Cases

Scan to Folder

The YNQ Client operates in over 340 million high-end printers globally and is a de facto SMB solution for MFPs (Multi-Functional Printers), granting to end users a seamless and secure way of saving scanned documents.

When YNQ runs on an MFP, it can save scanned documents directly into a network folder. The Visuality Systems' SMB functionality grants the scanner the ability to browse the network, locate available computers or servers, view shared folders and deliver documents to an accessible destination in the desired format, quickly, reliably, conveniently and securely. Since YNQ fully supports SMB3, scan jobs are securely transferred under end-to-end encryption.

Figure 2: Scan to folder

Automotive Manufacturing Floor

Automotive manufacturers have integrated Visuality Systems' YNQ into their custom factory test equipment. YNQ facilitates efficient data acquisition, allowing car makers to collect and analyze large volumes of data generated during the tests.

The configuration data that drives the tests is updated via SMB by an automated controller running YNQ. The data is then transferred securely encrypted via SMB to a data warehouse for data mining. The YNQ integration runs fast, consistently and reliably, while accelerating decision-making and enhancing manufacturing efficiency.

Figure 3: Automotive manufacturing automation

SMB3 shield for WinCE systems

Some legacy devices and human-machine interface (HMI) systems are still based on RTOS (like Windows Embedded Compact) that natively can support only the vulnerable SMB1 dialect. Even if the system is not connected to a network, many users expect vendors to upgrade these devices to SMB3 in order to guarantee end-to-end encryption. YNQ can be seamlessly added to the lower levels of the RTOS (e.g., WinCE) to enable SMB3 transparently and without affecting the application, thus prolonging the life of the embedded system by many years going forward.

YNQ Architecture

YNQ is a portable SMB software library, written in C language, that can be integrated virtually into any platform. There are two levels of YNQ adaptation - Porting and Integration.

Porting

The Porting process occurs when YNQ must be used on a new Operating System (OS), CPU or any other platform. Porting YNQ involves implementing its low layer by means of the most common programming interfaces (e.g., POSIX). This process is seamless and requires minimum effort. Porting is a prerequisite of Integration.

Integration

Integration of YNQ is provided for specific platforms, including Linux/UNIX, VxWorks, Nucleus, iOS, Integrity, Android, Windows, Windows CE and QNX with off-the-shelf solutions readily available. By configuring a few parameters, YNQ can be seamlessly incorporated into a new embedded system on these platforms.

YNQ architecture is illustrated in Figure 4. The components shown in blue (API/Protocol and Service) are fully portable, while those in green may be modified during either Porting or Integration.

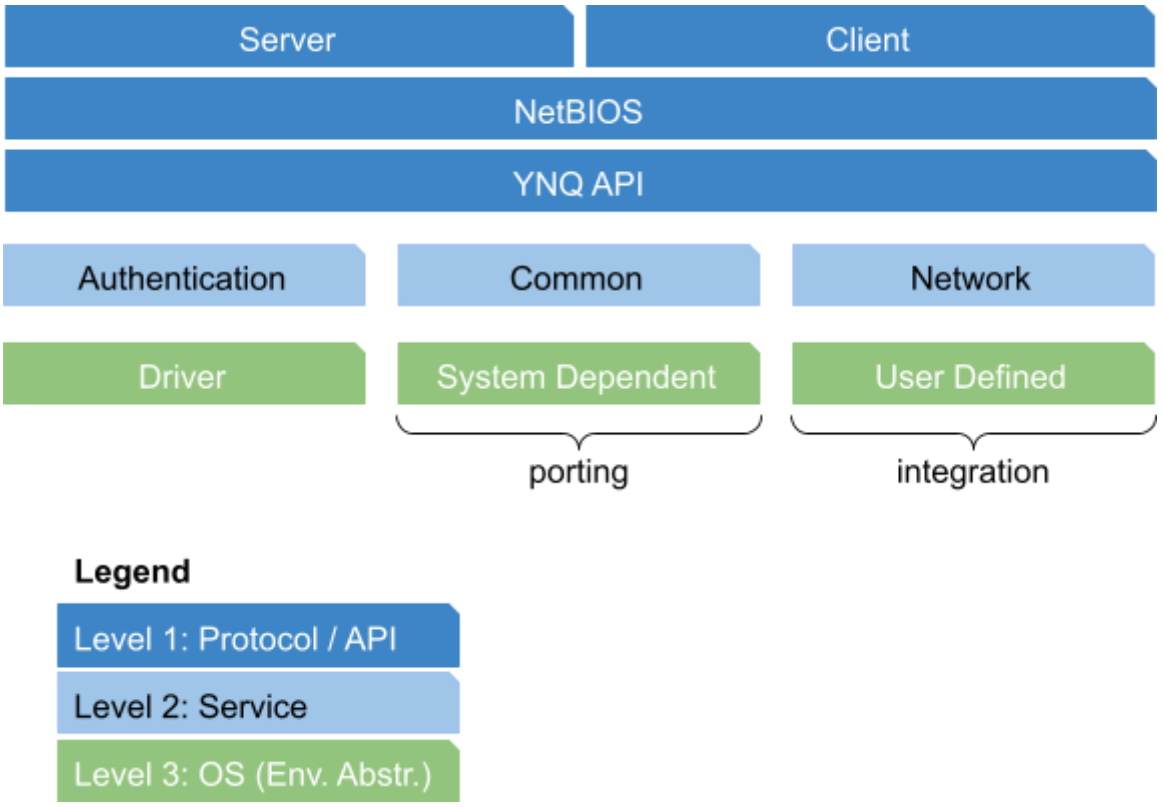

Figure 4: YNQ Architecture Layers

The YNQ Level 1 here is the central component of the entire architecture. It is responsible for the SMB Server and SMB Client functionality. The Level 3 (Environment abstraction component) maps an abstract system API on the exact operating system calls. There is a distinction between Project-Dependent, User-Defined (UD) and System-Dependent (SY) layers. With reference to the

difference between Porting and Integration, SY corresponds to Porting, while UD corresponds to Integration.

Level Number	Hierarchy	Name	Position	Description	Role / Modules / Functionality	Adaptation
1	High	API/Protocol	Frontend	Core code	API, Server, Client, NetBIOS Daemon	OS independent
2	Medium	Service		Service code	Authentication, Common, Network	
3	Low	OS (Environment Abstraction)	Backend	Sample implementation	System Dependent (SY), User Defined (UD), FS Driver	Porting (SY), integration (UD)

From a functional perspective, YNQ may be seen as an SMB Server, SMB Client and NetBIOS Daemon. Since the SMB Server is an application, it can be easily deployed, configured and run. The SMB Client is a software library available through its API. To benefit from the SMB Client, a YNQ customer should develop an application (or a set of applications). The SMB Corporate Server also uses SMB Client to achieve Domain Authentication (also called Active Directory Authentication). Another component is the File System (FS) Driver. This feature is system dependent, and the Driver is currently available on VxWorks, WinCE, QNX and Linux/UNIX (through FUSE). The Driver option allows developing client applications on top of the native API instead of Client API. The NetBIOS Daemon component is shared between the Server and Client to provide NetBIOS services, mostly name resolution.

The YNQ architecture is designed for the embedded world , therefore it supports the following features:

- **Pre-allocated memory:** YNQ uses fixed-size tables, which are either allocated statically or pre-allocated on startup. It implies some restrictions - like the maximum number of connections and open files - that perfectly fit the intrinsic constraints of an embedded system in terms of limited resources.
- **Multi-threading:** This feature applies to the SMB Client and ensures a fully thread-safe library. YNQ's SMB Server is single-threaded to guarantee a stable and reliable behavior.
- **Zero-copy:** YNQ's SMB Server prevents the payload copy of read and write operations.

Features

The YNQ product features are summarized in the following table.

Feature	YNQ Server	YNQ Client
Multi-dialect support from SMB1 to SMB 3.1.1	x	x
Authentication - Active Directory integration (or Domain Authentication)	x (Corporate)	
Authentication - local users	x	
Authentication - NTLM	x	x
Authentication - Kerberos		x
Message signing (AES-based for SMB3)	x	x
Encryption (AES-128-GCM, AES-128-CCM)	x	x
Encryption (AES-256-GCM, AES-256-CCM)	x	x
Optional ACL integration	x	
RPC over SMB - Basic – SRVSVC	x	x
RPC over SMB - Basic – WKSSVS, WINREG	x	
RPC over SMB - Authentication – NetLogon	x (Corporate)	x (Corporate)
RPC over SMB - Printing – SPOOLSS	x	
RPC over TCP - SPOOLSS	x	
RPC over TCP - NetLogon	x (Corporate)	
RPC over SMB - LSA, SAMR	x (Corporate)	x (Corporate)
IPv4 and IPv6 support	x	x
Calls - File data ops, meta-data, run-time fine-tuning		x
Synchronous and asynchronous reads and writes		x
Host to IP Resolution via DNS & LLMNR	x	x
Host to IP Resolution via NetBIOS & WINS	x	x
Network discovery - domain/server/share enumeration (including WS-Discovery)		x
Multi-threading		x
Durability		x
LDAP		x (Corporate)
SMB over QUIC with transport-level encryption (add-on)		x
Distributed File System (DFS)		x
Directory crawling		x
Large read/write packets to boost performance		x
Server side copy		x
FS driver		x
Symbolic links	x	x

QUIC Add-on

The QUIC Add-on, based on Microsoft MsQuic, provides YNQ clients with enhanced security, increased connectivity performance, high reliability, and improved user experience.

- **Enhanced security** is achieved through end-to-end encryption of data packets throughout the connection, eliminating the need for a VPN. The handshake process utilizes TLS 1.3 authentication.
- QUIC improves **connectivity performance** due to significantly lower latency. This is made possible by requiring only a single handshake to establish the connection. Additionally, a session ticket allows for faster connection restoration, when revisiting sites, compared to TCP.
- **High reliability** is ensured through a smart packet header mechanism that facilitates recovery in the event of a high loss-rate. The mechanism selectively resends only the packets that require reliable delivery. By utilizing a single session for simultaneous data streams, head-of-line blocking is avoided.
- QUIC enhances the **user experience** by assigning its own ID independent of IP and port. This feature enables seamless switching between cellular, Wi-Fi, or wired networks and effectively handles IP or port changes without causing disruptions.

Compliance and Connectivity

YNQ implements the most recent dialect of the SMB1 specification (NT LM 0.12), and all dialects of SMB2/SMB3 up to 3.1.1. Applications built with YNQ can enable connections from any SMB client running on Windows, Apple Macintosh, jNQ, Samba and more.

YNQ will always negotiate the highest SMB dialect supported between client and server. The following table shows what's the most recent SMB dialect supported by each Windows version.

Windows	Windows 11 Server 2022	Windows 10 Server 2019	Windows 10 Server 2016	Windows 8.1 Server 2012 R2	Windows 8 Server 2012	Windows 7 Server 2008 R2	Windows Vista Server 2008	Older
SMB Version	3.1.1	3.1.1	3.1.1	3.0.2	3	2.1	2	1

Summary

Current embedded systems may still lack an optimized SMB solution. It could be due to an RTOS with no available SMB solution, an old Windows machine with an outdated SMB version or a Linux/UNIX developer that refrains from utilizing open source Samba due to its support pattern, large footprint or licensing terms. The embedded market therefore needs an SMB solution which is reliable, effective, portable (to any Operating System), and which carries lower resource consumption.

With a multi-year, established experience in the SMB market, Visuality Systems offers its Embedded SMB solution, YNQ™, which is portable and can be integrated into any environment, thus bringing SMB client and server capabilities to any embedded device under a commercial license.

YNQ is available for integration as a source code.

Having an SMB protocol challenge?

Drop an email to info@visualitynq.com