AN3576

Software for Microchip Ethernet PHYs

Author: *John Haechten*
Microchip Technology Inc.

1.0 INTRODUCTION

This application note details the software required for Microchip Ethernet PHYs with part numbers beginning with “LAN,” “VSC,” and “KSZ.” These PHY groups are referred to in this document as Microchip PHYs.

This document addresses the stand-alone PHY software, the Microchip Ethernet PHY API (MEPA). Specifically, this application note is not needed for LAN, VSC, or KSZ PHYs included in a switch architecture or reference design, and whose design is using the MEPA software package included in the Switch Application MESA software package (such as, WebStaX, SMBStaX, IStaX, or CEServices). The MEPA software APIs within MESA have been integrated and tested within the switch SDK using a switch reference design.

The PHYs covered are connected to an embedded processor, FPGA, or Ethernet switch from a third-party vendor. This document also provides instructions on how to find and download the available PHY software from GitHub.

The specific Microchip PHY part numbers covered by this document are in Table 1, Table 2, and Table 3. Not all parts are included as some Microchip PHYs may not require any software. However, most Microchip PHYs (10M, 100M, 1G, 10G, and 25G) require software to control the PHY. This software consists of a known sequence of register access, which has been tested and verified to provide desired results.

Multiple options are available, including:

- U-Boot Driver
- Linux® Kernel Driver (open source)
- User Space APIs:
  - VSC6802: Legacy PHY API (API 4.67.05 is the latest release as of this writing, VSC PHY Maintenance only)
  - VSC6803: MESA API (MESA 2025.06 is the latest as of this writing, LAN/VSC/KSZ)
  - MEPA API (MEPA 2025.06 is the latest release as of this writing, covers LAN/VSC/KSZ PHYs)

Among the options, User Space APIs provide the most complete feature coverage and control for the Microchip PHYs. In some cases, both the Linux Kernel Driver and User Space API are available. In such instances, the User Space API is expected to have complete control of the PHY. Therefore, the Linux Driver should be disabled.

The use of a U-Boot Driver is often the desired way to control the Microchip PHY during initial boot sequences. In this phase of the boot cycle, Diagnostic Power-On Self Test (POST) may be run, and the Microchip PHY may be configured to receive the necessary system initialization code to continue the boot process.

The Linux Kernel Driver is also often used to control the Microchip PHY. However, this method can only be used with the Linux operating system (OS). The drivers have been released (and upstreamed into Linux Kernel.org). Linux drivers typically support common capabilities seen across many available PHYs, regardless of vendor. Therefore, the full functionality of a particular PHY, like any of the Microchip PHYs, may not be provided in the driver if that functionality is not common or supported.

Note 1: Not all VSC PHYs are included in the Linux Kernel Driver. For the Microchip PHYs that are included, not all PHY features may be supported by the Linux Kernel Driver, including advanced features, such as 1588 or MACsec.

2: When using the Linux Kernel Driver to control the PHY, the User Space API for the Microchip PHY cannot be used as this creates conflicts in the control plane. Conversely, when the User Space API for the Microchip PHY is being used, the Linux Driver support must be excluded for the same reasons.

The User Space API is OS agnostic and may be used with any OS. The User Space API comprises standard C code functions for specified Microchip PHYs.

Most Microchip PHYs are supported in the User Space API. There are few cases of PHYs being supported by the Linux Kernel Driver and not by the User Space API. For the VSC PHYs, there is an exception for VSC8530/31 and VSC8540/41, which are only supported by the Linux Driver.

A user should base the software on the available driver or User Space API instead of attempting to create a custom register sequence. Many of the Microchip PHYs have an embedded microcontroller core. Some register sequences include loading commands into the microcontroller core, which runs autonomously within the Microchip PHY and requires a handshake mechanism to communicate with the outside Application/API/Driver. Therefore, scripting of register accesses without checking the transaction status is not recommended.

The scripting of register access sequences may not always be an appropriate option, and the handshake is handled by the driver or User Space API. In addition, the register sequences provided in the driver or User Space API have been verified to operate correctly for the functionality provided without undesired side effects. It is recommended to use the driver as is or use it as the starting point and to adapt it to a specific environment in case designers prefer to expose and control additional features through the driver.

1.1 Sections

This document covers the following topics:

- Section 2.0, Software Availability for Each Microchip PHY
- Section 3.0, U-Boot Driver Solutions
- Section 4.0, Linux Kernel Driver Solutions
- Section 5.0, User Space API Solutions

1.2 References

Consult the following documents and design resources when using this application note:

- MEPA Release Notes: https://github.com/microchip-ung/sw-mepa/blob/v2025.06/mepa/docs/release-notes/mepa-2025.06.adoc
- MEPA: Getting Started: https://github.com/microchip-ung/sw-mepa/blob/master/mepa/docs/getting-started.adoc
- GitHub Software Repository for MEPA: https://github.com/microchip-ung/sw-mepa
- MESA: https://www.microchip.com/en-us/software-library/vsc6803
- GitHub Software Repository for MESA: https://github.com/microchip-ung/mesa
- Legacy VSC API: https://www.microchip.com/en-us/software-library/vsc6802
- GitHub Software Repo for Legacy VSC PHY API: https://github.com/microchip-ung/unified_api_4x

2.0 SOFTWARE AVAILABILITY FOR EACH MICROCHIP PHY

2.1 Software Availability for VSC PHY

Table 1 indicates the software available for each VSC device.

TABLE 1: VSC PHY SOFTWARE AVAILABILITY

PHY	Advanced Features	API	Linux Driver	Linux® Release (mscc.c or mscc/mscc_main.c)	U-Boot Driver	Product Page
VSC8211	No	Yes	Y*	—	Y*	https://www.microchip.com/wwwproducts/en/VSC8211
VSC8221	No	Yes	Y*	—	Y*	https://www.microchip.com/wwwproducts/en/VSC8221
VSC8501	No	Yes	Y	6.4.2	No	https://www.microchip.com/wwwproducts/en/VSC8501
VSC8502	No	Yes	Y	5.10.189	No	https://www.microchip.com/wwwproducts/en/VSC8502
VSC8514	No	Yes	Y	5.2	Y*	https://www.microchip.com/wwwproducts/en/VSC8514
VSC8512	No	Yes	No	—	No	https://www.microchip.com/wwwproducts/en/VSC8512
VSC8530	No	No	Y	4.10	Y	https://www.microchip.com/wwwproducts/en/VSC8530
VSC8531	No	No	Y	4.10	Y	https://www.microchip.com/wwwproducts/en/VSC8531
VSC8540	No	No	Y	4.10	Y	https://www.microchip.com/wwwproducts/en/VSC8540
VSC8541	No	No	Y	4.10	Y	https://www.microchip.com/wwwproducts/en/VSC8541
VSC8541RT	No	No	Y	4.10	Y	https://www.microchip.com/wwwproducts/en/VSC8541RT
VSC8504	No	Yes	Y	5.5	No	https://www.microchip.com/wwwproducts/en/VSC8504
VSC8552	No	Yes	Y	5.5	No	https://www.microchip.com/wwwproducts/en/VSC8552
VSC8572	Yes	Yes	Y**	5.5	No	https://www.microchip.com/wwwproducts/en/VSC8572
VSC8574	Yes	Yes	Y	4.2	Y**	https://www.microchip.com/wwwproducts/en/VSC8574
VSC8562	Yes	Yes	Y	5.5	Y*	https://www.microchip.com/wwwproducts/en/VSC8562
VSC8564	Yes	Yes	Y	5.5	Y*	https://www.microchip.com/wwwproducts/en/VSC8564
VSC8575	Yes	Yes	Y	5.5	No	https://www.microchip.com/wwwproducts/en/VSC8575
VSC8582	Yes	Yes	Y	5.5	No	https://www.microchip.com/wwwproducts/en/VSC8582
VSC8584	Yes	Yes	Y	4.2	Y**	https://www.microchip.com/wwwproducts/en/VSC8584
VSC8489	Yes	Yes	No	—	No	https://www.microchip.com/wwwproducts/en/VSC8489
VSC8490	Yes	Yes	No	—	No	https://www.microchip.com/wwwproducts/en/VSC8490
VSC8491	Yes	Yes	No	—	No	https://www.microchip.com/wwwproducts/en/VSC8491
VSC8254	Yes	Yes	No	—	No	https://www.microchip.com/wwwproducts/en/VSC8254
VSC8256	Yes	Yes	No	—	No	https://www.microchip.com/wwwproducts/en/VSC8256
VSC8257	Yes	Yes	No	—	No	https://www.microchip.com/wwwproducts/en/VSC8257
VSC8258	Yes	Yes	No	—	No	https://www.microchip.com/wwwproducts/en/VSC8258

Note 1: **Y** = Support in the `mscc.c or mscc/mscc_main.c` file only (implemented by Microchip).
Y* = Support in the `vitesse.c` file only (implemented by Freescale)
Y** = Support in `mscc.c or mscc/mscc_main.c` and `vitesse.c` files. User must only enable one driver. The `mscc.c or mscc/mscc_main.c` option is recommended.
Adv Feature = Advanced Feature Set in PHY = MACsec, 1588, or both

2: The functionality in the file `vitesse.c` will be deprecated if there is similar functionality in `mscc.c or mscc/mscc_main.c`.

3: The `mscc.c` file was transitioned to `mscc/mscc_main.c` for Linux Releases 5.7 and later.

2.2 Software Availability for LAN PHY

Table 2 indicates the software available for each LAN device.

TABLE 2: LAN PHY SOFTWARE AVAILABILITY

PHY	MEPA API Sup	Linux Driver	Linux File	U-Boot	U-Boot File	Product Page
LAN8814	2021.12		micrel.c	N/A		https://www.microchip.com/en-us/product/LAN8814
LAN8804	2021.12		micrel.c	N/A		https://www.microchip.com/en-us/product/LAN8804
LAN8841	2022.12		micrel.c	N/A		https://www.microchip.com/en-us/product/LAN8841
LAN8263	2025.06	N/A	N/A	N/A		N/A
LAN8268	2025.06	N/A	N/A	N/A		N/A
LAN8024	2025.06	N/A	N/A	N/A		N/A
LAN8043	2025.06	N/A	N/A	N/A		N/A
LAN8044	2025.06	N/A	N/A	N/A		N/A
LAN83C185	—	—	smsc.c	N/A		https://www.microchip.com/en-us/product/LAN83C185
LAN8770	2023.6	—	microchip_t1.c	—	smsc.c	https://www.microchip.com/en-us/product/LAN8770
LAN937X	2023.6	—	microchip_t1.c	—	smsc.c	https://www.microchip.com/en-us/product/LAN9370
LAN867X	2023.6	—	microchip_t1s.c	—	smsc.c	https://www.microchip.com/en-us/product/LAN8670
LAN865X	2023.6	—	microchip_t1s.c	—	smsc.c	https://www.microchip.com/en-us/product/LAN8650
LAN8700	—	—	smsc.c	—	smsc.c	https://www.microchip.com/en-us/product/LAN8700
LAN8710A	—	—	smsc.c	—	smsc.c	https://www.microchip.com/en-us/product/LAN8710A
LAN8720A	—	—	smsc.c	—	smsc.c	https://www.microchip.com/en-us/product/LAN8720A
LAN8740A	—	—	smsc.c	—	smsc.c	https://www.microchip.com/en-us/product/LAN8740A
LAN8741A	—	—	N/A	—	smsc.c	https://www.microchip.com/en-us/product/LAN8741A
LAN8742A	—	—	smsc.c	—	smsc.c	https://www.microchip.com/en-us/product/LAN8742A
LAN8870	2024.9	—	microchip_t1.c	—	smsc.c	https://www.microchip.com/en-us/product/LAN8870

2.3 Software Availability for KSZ PHY

Table 3 indicates the software available for each LAN device.

TABLE 3: KSZ PHY SOFTWARE AVAILABILITY

PHY	API	Linux Driver	Linux File	U-Boot	U-Boot File	Product Page
KSZ9021	—	—	micrel.c	—	micrel_ksz90x1.c	https://www.microchip.com/en-us/product/KSZ9021
KSZ9031	2021.12	—	micrel.c	—	micrel_ksz90x1.c	https://www.microchip.com/en-us/product/KSZ9031
KSZ9131	—	—	micrel.c	—	micrel_ksz90x1.c	https://www.microchip.com/en-us/product/KSZ9131
KSZ8001	—	—	micrel.c	—	—	https://www.microchip.com/en-us/product/KSZ8001
KSZ8721	—	—	micrel.c	—	—	https://www.microchip.com/en-us/product/ksz8721
KSZ8737	—	—	micrel.c	—	—	—
KSZ8021	—	—	micrel.c	—	micrel_ksz8xxx.c	—
KSZ8031	—	—	micrel.c	—	micrel_ksz8xxx.c	—
KSZ8041	—	—	micrel.c	—	micrel_ksz8xxx.c	https://www.microchip.com/en-us/product/KSZ8041
KSZ8041RNLI	—	—	micrel.c	—	micrel_ksz8xxx.c	https://www.microchip.com/en-us/product/KSZ8041
KSZ8051	—	—	micrel.c	—	micrel_ksz8xxx.c	https://www.microchip.com/en-us/product/KSZ8051
KSZ8061	—	—	micrel.c	—	micrel_ksz8xxx.c	https://www.microchip.com/en-us/product/KSZ8061
KSZ8081	—	—	micrel.c	—	micrel_ksz8xxx.c	https://www.microchip.com/en-us/product/KSZ8081
KSZ8873MLL	—	—	micrel.c	—	—	https://www.microchip.com/en-us/product/KSZ8873
KSZ886X	—	—	micrel.c	—	micrel_ksz8xxx.c	https://www.microchip.com/en-us/product/KSZ8864
KSZ9477	—	—	micrel.c	—	—	https://www.microchip.com/en-us/product/KSZ9477
KSZ8795	—	—	micrel.c	—	—	https://www.microchip.com/en-us/product/KSZ8795

3.0 U-BOOT DRIVER SOLUTIONS

A driver in U-Boot for some of the Microchip PHYs is available (U-Boot v2025.01 or later). See https://www.denx.de/wiki/U-Boot/WebHome.

Refer to Table 1, Table 2, and Table 3 to determine which files are needed to support any given PHY.

Using version v2025.01 as an example, the:

- MCHP/VSC driver is located at u-boot-2025.01/drivers/net/phy/mscc.c.
- Freescale driver is located at u-boot-2025.01/drivers/net/phy/vitesse.c.
- Micrel KSZ8xxx driver is located at u-boot-2025.01/drivers/net/phy/micrel_ksz8xxx.c.
- Micrel KSZ90x1 driver is located at u-boot-2025.01/drivers/net/phy/micrel_ksz90x1.c.
- SMSC LAN8700/10/40/41/42 and LAN911x drivers are located at u-boot/blob/master/drivers/net/phy/smsc.c.

3.1 Enabling U-Boot Driver

The U-Boot driver can be configured to include or exclude drivers for the intended support. Designers are advised to familiarize themselves with the KConfig in U-Boot to configure the compilation to be included in driver support for the PHYs on their particular boards. Refer to https://docs.u-boot.org/en/latest/develop/kconfig.html for more information.

Note the following guidelines:

For the VSC PHYs, make sure that the CONFIG_PHY_MSCC definition is defined in the build, so the `mscc.c` file gets built and pulled into the build. When included, a `mscc.o` is generated.

For the SMSC PHYs, make sure that the CONFIG_PHY_SMSC definition is defined in the build, so the `smsc.c` file gets built and pulled into the build. When included, a `smsc.o` is generated.

For the MICREL_KSZ8XXX PHYs, make sure that the CONFIG_PHY_MICREL_KSZ8XXX definition is defined in the build, so the `micrel_ksz8xxx.c` file gets built and pulled into the build. When included, a `micrel_ksz8xxx.o` is generated.

For the MICREL_KSZ90X1 PHYs, make sure that the CONFIG_PHY_MICREL_KSZ90X1 definition is defined in the build, so the `micrel_ksz90X1.c` file gets built and pulled into the build. When included, a `micrel_ksz90x1.o` is generated.

For more information or additional documentation about U-Boot, go to https://www.denx.de/wiki/U-Boot/WebHome.

Note:	In the event of any duplication, U-Boot drivers in `vitesse.c` and in `mscc.c`, those drivers in `vitesse.c` shall be deprecated.

4.0 LINUX KERNEL DRIVER SOLUTIONS

Linux Kernel Drivers are available for many of the PHYs. At minimum, the Linux Kernel Driver supports the basic Microchip PHY functionality. Advanced feature functionality (1588 and MACsec) may also be available for some devices. If the MIT-licensed Linux driver exists, it is released via open source.

Download the latest driver from Linux Kernel Archives at https://www.kernel.org/.

- HTTP: https://www.kernel.org/pub/
- GIT: https://git.kernel.org/

Any Linux Kernel Driver version after version 4.10 will have all contributions for VSC8530/31 and VSC8540/41 families. Support for other Microchip Ethernet PHYs continues to be added over time.

It is recommended to download a tarball from one of the later stable versions of Linux. (See 6.16.1, which is the latest stable version available at this time.)

In the past, Linux Kernel Drivers were developed by customers or third parties that contributed to the open source community. Those drivers are located in the `vitesse.c` file and are mostly for older PHY generations.

Using version 5.6.14 as an example, the:

- Microchip-developed drivers are located at `linux-5.6.14\drivers\net\phy\mscc.c`
- Freescale-developed drivers are located at `linux-5.6.14\drivers\net\phy\vitesse.c`

Note that prior to the Linux 5.7 release, the Microchip-developed drivers are available and located in `mscc.c`. There may be duplicated drivers developed by third parties that are deprecated. Therefore, when building the Kernel, the build configuration should be updated to exclude the deprecated drivers, as they should not be included.

With the Linux 5.7 release, the directory structure changed slightly and the `mscc.c` file was moved as support for more PHYs. In addition, more PHY functionality was added. The updated location is: linux-5.7\drivers\net\phy\mscc\mscc_-main.c.

Using version 6.16.5 as an example:

- Microchip-developed VSC drivers (VSC) are located at https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/tree/drivers/net/phy/mscc/mscc_main.c?h=v6.16.5
- Microchip-developed micrel drivers (LAN/KSZ) are located at https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/tree/drivers/net/phy/micrel.c?h=v6.16.5
- Microchip-developed smsc drivers (LAN) are located at https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/tree/drivers/net/phy/smsc.c?h=v6.16.5.
- Freescale-developed drivers (VSC legacy) are located at https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/tree/drivers/net/phy/vitesse.c?h=v6.16.5.

4.1 Enabling Linux Kernel Driver

The Linux Kernel Documentation can be found at: https://docs.kernel.org.

The Linux Kernel must be downloaded and then using the Kernel Build System. (See: https://docs.kernel.org/kbuild/index.html) the Kernel must be configured for the environment/specific board. The PHY Driver modules must be added to the Kernel configuration in order to be compiled and included in the kernel. The Config Keywords are shown below:

- For VSC PHYs, refer to Table 1. Ensure that the CONFIG_MICROSEMI_PHY definition is defined in the build, so the `mscc.c` file gets built and pulled into the build. When included, an `mscc.o` is generated.
- For LAN/KSZ PHYs, refer to Table 2, and Table 3. Ensure that the CONFIG_MICREL_PHY definition is defined in the build, so the `micrel.c` file gets built and pulled into the build. When included, a `micrel.o` is generated.
- For LAN88xx PHYs, refer to Table 2. Ensure that the CONFIG_MICROCHIP_PHY definition is defined in the build, so the `microchip.c` file gets built and pulled into the build. When included, a `microchip.o` is generated.
- For LAN T1 PHYs, refer to Table 2. Ensure that the CONFIG_MICROCHIP_T1_PHY definition is defined in the build, so the `microchip_t1.c` file gets built and pulled into the build. When included, a `microchip_t1.o` is generated.
- For LAN T1s PHYs, see Table 2. Ensure that the CONFIG_MICROCHIP_T1S_PHY definition is defined in the build, so the `microchip_t1s.c` file gets built and pulled into the build. When included, a `microchip_t1s.o` is generated.

5.0 USER SPACE API SOLUTIONS

5.1 User Space API Software Packages

The Microchip Ethernet PHY API is a software API package that is compiled with or linked to the user application. The API uses accessor functions (MDIO/SPI) to manage, control, and configure the PHY via register programming.

There are three types of User Space APIs:

1. PHY API (API 4.67.05 is the latest release as of this writing, VSC PHY Maintenance only). Sample application code is located in the `appl/vtss_appl_cu_phy.c` file.
2. MESA API (MESA 2025.06 is the latest as of this writing, LAN/VSC/KSZ). For releases prior to 2025.12, App Note 4028 applies and the sample application code is located in the `phy_demo_appl/appl/vtss_appl_vsc_phy_1g.c` file.
3. MEPA API (MEPA 2025.06 is the latest release as of this writing, covers LAN/VSC/KSZ PHY. Please see the associated `.adoc` documentation with the release.

5.2 MEPA API (Applicable to New PHY Designs and the Latest Revisions of the Legacy PHY Designs)

The MEPA API is a stand-alone PHY API recommended for all designs which do not require integration with the IStax application. This API get synchronized/integrated with the MEPA software within the MESA API, but the latest releases may not have been fully integrated into MESA at the time of the SW-MEPA release. Therefore, users need to read the release notes if there are any questions and concerns.

The VSC PHY versions are distinguished using the suffix in their part numbers. The latest versions of the legacy IEEE-1588-capable VSC PHYs have been updated to address the Out-of-Sync (OOS) issue in the IEEE-1588 feature and thus do not require any workaround. These versions are listed below:

- VSC8572-02
- VSC8572-05
- VSC8574-02
- VSC8574-05
- VSC8254-01
- VSC8256-01
- VSC8257-01
- VSC8258-01
- VSC8489-02
- VSC8489-17
- VSC8490-17
- VSC8491-17
- LAN8814
- LAN8804
- LAN8770
- LAN8870
- LAN8871
- LAN8872
- LAN8840
- LAN8841
- KSZ9031
- Intel Corp. PHY chips supported by PHY API library: GPY_API_v2.7.1.1
- Aquantia supported by PHY API libraries 2.6.1 and 2.6.2. Examples are:
  - AQR407
  - AQR409
  - AQR411
  - AQR412

5.2.1 MEPA API (UNDER DEVELOPMENT, FREQUENT NEW RELEASES, AND NEW PRODUCT SUPPORT)

The MEPA API is available on GitHub. Currently, the GitHub repository for the MEPA API contains the latest support for Microchip PHYs. Among the MEPA API releases available are:

- MEPA-2024.09
- MEPA-2024.12
- Quarterly releases starting with MEPA-2024.09 and continuing:
  - MEPA-2024.12
  - MEPA-2025.03
  - MEPA-2025.06
- Future Releases

The repository can be found at https://github.com/microchip-ung/sw-mepa. The release notes indicate the versions of MEPA and the latest versions can be found at: https://github.com/microchip-ung/sw-mepa/tree/master/mepa/docs/release-notes.

The latest MEPA API version is the default version at any given time. To obtain earlier versions, go to the **Tags** tab and choose the associated commit for the earlier version of the code. To find a specific release, download the release and artifacts for that release on GitHub. For example, for the MEPA-2022.12 release, go to https://github.com/microchip-ung/sw-mepa/releases/tag/v2024.12.

Under Assets, the Source Code file indicates: `sw-mepa-2024.12.tar.gz`. The file contains all the artifacts required to build the released code. The mepa/docs folder contains all the important reference materials for the product.

- Example: mepa-v2024.12/sw-mepa-2024.12/mepa/docs
  - Readme
  - Introduction
  - getting-started
  - mepa_instantiation
  - basic
  - linkup_config
  - timestamp
  - macsec
  - Others

5.3 VSC6803: Microsemi Ethernet Switch API or MESA API

The VSC6803 API has been integrated with the IStax Application and is recommended for all designs which use the IStax application. The MEPA version within the VSC6803 MESA API does get synchronized or integrated into the SW-MEPA repository, but the latest releases of SW-MEPA API may not have been integrated into MESA at the time of release. Therefore, users need to read the release notes for the MEPA code within the MESA Repository.

5.3.1 VSC6803 – MESA API

The product pages can be found on the Microchip website at https://www.microchip.com/wwwproducts/en/VSC6803.

The **Documents** tab contains all the important reference materials for the product. The VSC6803 MESA API is available on GitHub. Currently, the GitHub repository for MESA API contains the following MESAAPI releases: v2019.06, v2019.09, v2019.12, v2020.03……. v2024.12.

The MESA repository can be found at https://github.com/microchip-ung/mesa.

The release notes within the MESA repository indicate the version of MEPA which has been integrated with MESA code and tested with the IStax application. The latest versions can be found at:

https://github.com/microchip-ung/mesa/tree/master/mepa/docs/release-notes.

The latest MESA API version is the default version at any given time. For obtaining earlier versions, go to the **Tags** tab and choose the associated commit for the earlier version of the code. To obtain a specific release, download the release and artifacts for that release on GitHub.For example, for the v2024.12 release, go to: https://github.com/microchip-ung/mesa/releases/tag/v2024.12.

Under Assets, search the file, `mesa-v2024.12.tar.gz`. The file contains all the artifacts required to build the released code.

5.4 VSC PHY API (Only Applicable to VSC Legacy Designs)

The VSC PHY API is only applicable to designs that use legacy versions of 1588 VSC PHYs, and those that use the 1588 Timestamping feature in those PHYs. In the past, the VSC PHY API had a software workaround that was needed for legacy versions of 1588 within legacy VSC PHYs. The issue (1588 OOS Recovery) was fixed in the hardware in 2017/2018, and the latest revisions of the 1588 within the VSC PHYs do not require the workaround as it is fixed in the silicon.

This OOS Recovery workaround was not ported to the MESA API or to the MEPA API.

VSC PHY versions are distinguished using the suffix in their part numbers. Some part numbers do not have a suffix. The legacy versions of 1588 VSC PHYs that require the use of PHY API when using the 1588 feature are:

- VSC8572-01
- VSC8572-04
- VSC8574-01
- VSC8574-04
- VSC8254 (no suffix)
- VSC8256 (no suffix)
- VSC8257 (no suffix)
- VSC8258 (no suffix)
- VSC8489 (no suffix)
- VSC8489-10
- VSC8489-11
- VSC8489-13
- VSC8489-14
- VSC8489-15
- VSC8490-10
- VSC8490-11
- VSC8490-13
- VSC8490-14
- VSC8491-10
- VSC8491-11
- VSC8491-13
- VSC8491-14

Note:	When working with the VSC8489-02, VSC8489-17, VSC8490-17, VSC8254-01, VSC8257-01, or VSC8258-01 in an application that uses the 1588 feature, do not use VSC PHY API. Use either the MESA API integrated with IStax or use the latest MEPA API instead.

5.4.1 VSC PHY API (MAINTENANCE ONLY, NO NEW RELEASES)

The product pages can be found on the Microchip website at https://www.microchip.com/wwwproducts/en/VSC6802.

The **Documents** tab contains all the important reference materials for the product. The current and latest PHY API release for VSC6802 is 4.67.05. This code line is in maintenance mode and is not expected to change. This version of the code includes 1588 OOS Recovery for earlier legacy versions of the VSC PHYs.

The PHY API is available on GitHub. Currently, the GitHub repository for PHY API contains the following PHY API release: 4.67.05. The repository is located at: https://github.com/microchip-ung/unified_api_4x.

The latest/last PHY API version is the default version, and this is the only version currently available. As this code line is in Maintenance mode, the latest/last released version is API 4.67.05 which was released Sept 21, 2018.

The release notes (`release_history.txt`) can be found at: https://github.com/microchip-ung/unified_api_4x/tree/master/vtss_api.

APPENDIX A: REVISION HISTORY

TABLE A-1: REVISION HISTORY

Revision Level & Date	Section/Figure/Entry	Correction
DS00003576E (10-02-25)	Document title	Updated document title to include other Microchip PHY products.
	Section 1.0, Introduction	Updated the section to show the expanded coverage of the application note's new version, which now includes Microchip KSZ and LAN PHYs.
	Section 2.0, Software Availability for Each Microchip PHY	Added tables and subsections to include Microchip KSZ and LAN PHY parts.
DS00003576D (03-20-25)	Section 5.0, User Space API Solutions	Updated section.
	Section 5.2, MEPA API (Applicable to New PHY Designs and the Latest Revisions of the Legacy PHY Designs)	New section
	All	Made minor text and formatting changes.
DS00003576C (09-29-23)	Table 1	Corrected Linux® driver support for VSC8501/VSC8502 based on KB "VSC8502/VSC8501 - Linux Support."
	Section 1.0, Introduction	Changed "whereKSZ" to "KSZ."
DS00003576B (07-22-22)	VSC6803 - ETH API/MESA API	Added file and content information on GitHub.
	All	Made minor text and formatting changes.
DS00003576A (08-03-20)	Initial release	

Microchip Information

Trademarks

The “Microchip” name and logo, the “M” logo, and other names, logos, and brands are registered and unregistered trademarks of Microchip Technology Incorporated or its affiliates and/or subsidiaries in the United States and/or other countries (“Microchip Trademarks”). Information regarding Microchip Trademarks can be found at https://www.microchip.com/en-us/about/legal-information/microchip-trademarks.

ISBN: 979-8-3371-2101-7

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.