SmartHBA 2100 and SmartRAID 3100 Software/Firmware Release Notes

Released / August 2021

© 2021 Microchip Technology Inc.

Revision History : August 2021

Revision	Revision Date	Details of Change
29	August 2021	SR 2.6.2 with firmware version 4.11 B0
28	April 2021	SR 2.6.1.1 with VMware driver version 4054.2.118.
27	March 2021	SR 2.6.1 with VMware driver version 4054.1.103.
26	February 2021	SR 2.6 Production Release
25	October 2020	SR 2.5.4 Production Release
24	August 2020	SR 2.5.2.2 Production Release with Firmware 3.00
23	March 2020	SR 2.5.2 Production Release with Firmware 2.93
22	March 2020	SR 2.5 Production Release with Firmware 2.66
21	February 2020	SR 2.5.2 Production Release
20	October 2019	SR 2.5 Production Release
19	September 2019	Updated for SR 2.4.8.1 (fw v2.31 Build 0)
18	August 2019	Updated for SR 2.4.8
17	January 2019	SR2.4 Production Release
16	June 2018	SR2.3 Production Release
15	June 2018	Updated for RC Release
14	October 2017	Update supported OSs
13	October 13, 2017	First Production Release
1-12	June 2016-July 2017	Pre-Production Releases.

Table of Contents

1 About This Release

The development release described in this document includes firmware, OS drivers, tools, and host management software for the SmartHBA 2100/SmartRAID 3100 controller solutions from Microchip.

1.1 Release Identification

The firmware, software, and driver versions for this release are shown in the following table.

Table 1-1 • Release Summary

Solutions Release	2.6.2
Package Release Date	August 5, 2021
Firmware Version	4.11 B0[1,2] (basecode 06.06.005.000)
UEFI Version	1.3.14.5
Legacy BIOS	1.3.14.2
Driver Versions[3]	Windows SmartPQI: • Windows 2012/2016/2019: 1010.6.0.1025 • Windows 8/8.1/10: 1010.6.0.1025 Linux SmartPQI: • RHEL 6/7/8: 2.1.12-055 • SLES 12/15: 2.1.12-055 • Ubuntu 16/18/20: 2.1.12-055 • Debian 9/10: 2.1.12-055 • CentOS 6/7/8: 2.1.12-055 • Oracle Linux 7/8: 2.1.12-055 • Citrix XenServer 7/8: 2.1.12-055 VMware SmartPQI: • VMware 6.5/6.7/7.0: 4150.0.119 FreeBSD/Solaris SmartPQI: • FreeBSD 11/12: 4130.0.1008 • Solaris 11: 4120.0.1005
Management Software (arcconf, maxView™, Event Monitor, BootUSB)	B24308

Note:

1. Downgrading to 1.04 B0 or older builds from this release or prior 1.29 releases may cause the board to not boot or have supercap errors due to an incompatibility in SEEPROMs between this release and prior releases. Refer to the section " *Updating the Controller Firmware* " to downgrade an existing board.
2. If the firmware running on the board is older than 0.01 B594, existing data in the logical volumes must be backed up if it needs to be used after the upgrade. After the upgrade from firmware prior to 0.01 B594, the logical volumes will need to be recreated.
3. Only run the driver on firmware 0.01 build 500 or later.

1.2 Components and Documents Included in this Release

Download the firmware, drivers, host management software, and supporting documentation for your SmartHBA 2100/SmartRAID 3100 controller SmartHBA 2100/SmartRAID 3100 controller and SmartRAID 3100 and SmartRAID 3100 controller solutions from the Microchip Web site at *https://start.adaptec.com*

1.3 Files Included in this Release

This release consists of the files listed in the following tables:

Firmware Files

Table 1-2 • Firmware Files

Component	Description	Pre-Assembly Use	Post-Assembly Use
SmartFWx100.bin	Programmable NOR Flash File Use to program NOR Flash for boards that are already running firmware.		X

Table 1-3 • Firmware Programming Tools

Tool	Description	Executable
Arcconf romupdate	The command allows to upgrade/downgrade the firmware and BIOS image to the controller.	Refer to *Table 1-7 • Host Management Utilities* on page 4
maxView firmware upgrade wizard	The firmware upgrade wizard allows to upgrade/downgrade the firmware and BIOS image to one or more controller(s) of same model in the system.	Refer to *Table 1-7 • Host Management Utilities* on page 4

Driver Files

Table 1-4 • Windows Storport Miniport SmartPQI Drivers

Package	Drivers	Binary	Version
2012	Server 2019 Server 2016 and Windows 10 Server 2012 SP1, R2 SP1 and Windows 8.1, 8	SmartPqi.sys	x64
		SmartPqi.inf	x64
		Smartpqi.cat	x64

Table 1-5 • Linux SmartPQI Drivers

Drivers	Version
Red Hat Enterprise Linux 8.4, 8.3, 8.2, 8.1, 8.0, 7.9, 7.8, 7.7, 7.6	x64
CentOS 8.3, 8.2, 8.1, 8.0, 7.9, 7.8, 7.7, 7.6	x64
SuSE Linux Enterprise Server 12[1], SP5, SP4, SP3	x64
SuSE Linux Enterprise Server 15 SP3, SP2, SP1[1]	x64
Oracle Linux 7.6 with UEK5u2 (4.14.35)	x64
Oracle Linux 7.9, 7.8, 7.7 UEK 6U1	x64
Oracle Linux 8.2, 8.1 UEK6	x64

Drivers	Version
Oracle Linux 8.3, 8.2 UEK6U1	x64
Ubuntu 21.04	x64
Ubuntu 20.04.2, 20.04.1, 20.04	x64
Ubuntu 18.04.5, 18.04.4, 18.04	x64
Ubuntu 16.04.5	x64
Debian 10.05	x64
Debian 9.13	x64
Citrix xenServer 8.2, 8.1, 8.0, 7.6	x64
Fedora 33 (inbox only)	x64

Note: 1. To mitigate against the Spectre Variant 2 vulnerability, the RHEL 6u9/RHEL7u4/RHEL7u5 and SLES11 SP3 and higher drivers have been compiled to avoid the usage of indirect jumps. This method is known as "Retpoline".

Table 1-6 • FreeBSD, Solaris, and VMware SmartPQI Drivers

Drivers	Version
FreeBSD 13, 12.2, 11.4	x64
Solaris 11.4	x64
VMware 6.7 U3/U2/U1, 6.5 U3/U2/U1	x64
VMware 7.0 U2/U1	x64

Host Management Software

Table 1-7 • Host Management Utilities

Description	OS	Executable
ARCCONF Command Line Utility	Windows x64 Linux x64 VMware 6.5 and above XenServer FreeBSD x64 Solaris x86	See the Arcconf download package for the OS-applicable installation executable.
ARCCONF for UEFI		Included as part of the firmware downloadable image.
maxView Storage Manager	Windows x64 Linux x64 VMware EXSi 6.5 and above XenServer	See the maxView Storage Manager download package for the OS-applicable installation executable.

Description	OS	Executable
maxView vSphere Plugin	VMware 6.5 and above	See the VMware maxView Storage Manager download package for the OS-applicable installation executable.
Boot USB (offline or pre-boot) for ARCC-ONF and maxView Storage Manager	Linux x64	See the maxView BootUSB download package for the .iso file.

2 What is New?

2.1 Features

The following table lists features supported for this release.

Table 2-8 • Feature Summary

Feature		Supported in this Release	Future Release
UEFI Driver, Boot Support		X	
Legacy Boot Support		X	
Dynamic Power Management		X	
SMR Drive Support	Enumeration, Unrestricted Command Flow-Through	X	
	SATL Translation for HA/HM SMR Management	X	
	Identify All Drive Types	X	
Driver Support	Windows	X	
	Linux	X	
	VMware	X	
	FreeBSD	X	
	Solaris	X	
	OS certification	X	
Out of Band interface selection support of MCTP or PBSI		X	
Flash Support		X	
MCTP BMC Management		X	
Configurable Big Block Cache Bypass		X	
Green Backup Support for SmartRAID		X	
4Kn Support in RAID		X	

2.2 Fixes

2.2.1 Firmware Fixes

2.2.1.1 Fixes and Enhancements for Firmware Release 4.11 B0

This release includes the following fixes and enhancements:

- Enabled persistent firmware logging capability (Serial Output Buffer) across cold resets for controllers that support data preservation and added support to provide the persistent event logs to host software applications (for example, via ADU report or savesupportarchive.)
- The following error handling improvements were added for fault tolerant logical drive READ operations to a Predictive Failure (PF) drive encountering Unrecovered Read Error (URE) or timeouts.
  - READ operations to the logical drive are regenerated from redundant data instead of using PF drive. SSD IOBypass is temporarily disabled on the logical drive.
  - Error recovery logic was improved to avoid READ retries for URE on PF drives.
  - Predictive Spare Rebuild (PSR) was improved to perform regenerative rebuild to the spare drive instead of using the URE/timeout exhibiting PF drives as source for copying data to the spare drive.
  - If there are no more UREs/timeouts for the last 30 minutes, the PF drive will be used as usual with SSD IOBypass enabled and PSR rebuild will copy data from the PF drive to the spare drive.
- Enabled persistent firmware logs (Serial Output Buffer) across cold resets for controllers that support data preservation.
- Fixed an issue where drives are still discovered when the firmware reports a "Memory Self-Test Error" message.
  - Root Cause: During Power ON, the controller will run an initialization thread and discovery thread concurrently that causes internal memory usage to increase. If a memory self-test error occurs, the attached devices are still discovered when they should not have been discovered.
  - Fix: Firmware will run the initialization thread first so the memory self-tests are completed before discovery is run.
  - Risk: Low
- Fixed a problem where LUN resets were observed when host I/O and a logical drive rebuild is in progress due to a physical request resource contention.
  - Root Cause: Due a physical request resource contention between the parity mapper firmware responsible for host IOs and the rebuild priority code which is responsible for quiescing the host IOs based on priority, host IOs are submitted at a lower rate than expected, resulting in slower responses back to OS driver leading to a LUN Reset.
  - Fix: Cancel the rebuild quiesce host I/O logic for the rebuild iteration during the physical request resource contention situation, so host IOs waiting on the queue can be submitted to avoid LUN Resets.
  - Risk: Low
- Fixed a TLB exception lockup issue when multiple Out-of-Band MCTP requests were sent to the firmware at the same time.
  - Root Cause: TLB Exception/NULL pointer exception occurs in the firmware when it receives asynchronous MCTP requests at the same time in a session when a previous MCTP request has not been processed fully. Due to this, the firmware gets into a timing sensitive situation where one of the threads in the firmware is setting up the packetized MCTP responses by accessing the OOB session memory buffer which was just freed up by another thread responsible for processing MCTP requests. This is because the firmware handles one MCTP request in a session in the synchronous manner, if it receives another request from the same session before completing the existing request, it deletes the old session context and starts processing the new request.
  - Fix: To gracefully handle this situation, the firmware will use spinlock while accessing the OOB session from different threads.
  - Risk: Low
- Fixed an issue where SSD IOBypass is not enabled on logical drives created on the same array in certain scenarios.
  - Root Cause: When there are multiple logical drives in the same array and if any operation involving logical drive transformation occurs, then SSD IOBypass gets disabled. When the transformation operation finishes, IOBypass is not getting enabled again due to incorrect lookup logic while

parsing the logical drives. A similar issue was also found when the OS driver was disabled and reloaded again.
  - Fix: Lookup logic while parsing through the logical drives within the same array was corrected to enable SSD IOBypass on all eligible logical drives.
  - Risk: Low

- Fixed a problem where WRITE and READ fails on a RAID6 volume where it could have completed successfully in certain error recovery scenarios.
  - Root Cause: In the following use cases, the firmware could have completed the I/O operation successfully but it does not allow the regeneration of RAID6 read requests that failed with re-mappable errors.

    1. Writing a single column with one drive returning remappable read error without valid URE in sense information and the URE sector cannot be found. When there was host read waiting on a continuously failing internal flush write requests, it resulted in OS driver initiating the LUN reset and waiting for its completion indefinitely.

       In another situation, when requests to flush to the disks were failing continuously during boot-time, firmware was not completing the pre-boot components' requests on time, resulting in a 0x1E30 lockup since the maximum number of synchronous host requests were exhausted.
    2. Reading the data strip from failed drive with bad block (URE) on another drive in the same row.
    3. When the WRITE is mapped into a failed drive and bad block (URE) on another drive in the same row, the current implementation is propagating URE into P and Q drives, write is returned successfully.

  - Fix: Allow the RAID6 code to regenerate data which completes the above scenarios. In the third scenario, the bad block could also be cleared if the regenerated data is written successfully on the bad block (URE).
  - Risk: High

- Fixed an issue where rebuilding the volume was reported in Media Exchange (MEx) state after reboot.
  - Root Cause: The RAID metadata writes are not attempted for retry when it is failed with CHECK CONDITION status. This resulted in the metadata read from the drive becoming invalid during next boot, the drive is marked as a REPLACEMENT type. Therefore, the firmware fails the volume with MEx status once the REPLACEMENT count exceeds the fault-tolerance of the RAID volume.
  - Fix: Retry the failed RAID metadata writes and once retry limit is exhausted, fail the drive.
  - Risk: low

- Fixed a problem where events were not getting logged in certain scenarios.
  - Root Cause: The firmware has a duplicate event check using a few fields of the event record. If the host tools are not consuming events and new events are of same class type (for example, several array creation and deletion with no tools consuming events), the duplicate event check is not adequate to determine uniqueness and assumes that those are duplicate events.
  - Fix: Log all events including same event class types except the physical drive class type.
  - Risk: Low

- Fixed an issue where a Windows memory dump process was taking longer than 30 minutes.
  - Root Cause: As part of the Windows memory dump process, the smartPQI driver triggers a soft-reset to the controller firmware. After the soft-reset the controller firmware was not correctly enabling the logical drive write cache which resulted in the memory dump process taking more than 30 minutes when it was expected to complete in a few minutes.
  - Fix: Properly enable the logical drive write cache after the controller soft-reset was triggered.
  - Risk: Low

- Fixed a problem where FAULT LED was not turning on for the drives failing during device discovery behind the UBM backplane.
  - Root Cause: The firmware was not setting the internal flag which is referred while turning on the FAULT LED when a drive is failed during spinup/discovery sequence behind the UBM backplane.
  - Fix: Set the internal flag for the failed drives behind the UBM backplane.
  - Risk: Low
- Fixed an issue where LUN reset for SEP device completes with incorrect LUN status to the host.
  - Root Cause: When processing the LUN reset for SEP devices, the firmware incorrectly detects the SEP device as RAID device and completes the LUN Reset with the error status back to the host driver.
  - Fix: Allow the LUN Reset command to be sent to the SEP device to complete.
  - Risk: Low
- Fixed a problem where the host WRITE I/O failed on the maxCache logical drive with UREs present on the primary logical drive.
  - Root Cause: During a maxCache write operation, previously existing data in the smart cache logical drive needed to be flushed to the primary logical drive. However the write retry count was not initialized correctly during the cache flush operation so the write operations to the primary logical drive failed when UREs were encountered. Since the previously cached data could not be written to the primary logical drive, the new write request to the maxCache logical drive failed.
  - Fix: Initialize the retry count of WRITE requests to the primary logical drive along with resetting status when the READ from primary logical drive fails. This will result in WRITE logical request retried properly if there are errors during parity generation.
  - Risk: Low
- Fixed an issue where the controller may be non-responsive after drive failure with SSD IOBypass enabled under a high queue depth workload.
  - Root Cause: The logic to update RAID metadata on the remaining drives was taking long enough to falsely cause the controller to appear hung due to the large amount of I/O error recovery. If the I/O error recovery loop was processing multiple IO's for the same device, it would get into a non-preemptable loop where the controller heartbeat would not be updated in a timely manner.
  - Fix: The non-preemptable condition was eliminated from the error recovery flow to allow regular heartbeat updates to occur.
  - Risk: Low
- Fixed an issue where an SMP discover request can be sent to an invalid PHY index during expander hot-add.
  - Root Cause: Some expanders do not properly report any affiliation with an enclosure, example via EnclosureLogicalIdentifier, and thus firmware uses additional checks such as PHY information to determine if two expanders are JBOD "I/O module" mirrors. These checks assumed the number of PHYs would be equal, but if the number of PHYs on the expander is less than the expander it is being compared to, the SMP request that is generated will be to an invalid PHY index.
  - Fix: Added a check prior to generating these requests that the number of phys in the expanders is the same.
  - Risk: Low
- Added a workaround to a potential controller hang where a JBOD SES target becomes unresponsive during power cycle
  - Root Cause: Power cycle testing with third-party external JBOD revealed an issue where that device's SES processor can be in a state where it will accept an inbound SSP request, but will never respond to it. Because this device can also take several minutes to become ready and because it can only accept one command at a time, the resulting user experience appears as

though the controller has become hung if the host attempts to access that target after it's initially been discovered.
    - Fix: Firmware will now ubiquitously apply a timeout value to requests to SES targets regardless of whether the host has provided a timeout. This can at least allow the I/O stack to abort and recover from the attempted I/O rather than waiting for the device to respond.
    - Risk: Low. (Ubiquitous application of a timeout is generally problematic for SCSI when it is applied in the middle of the I/O stack, but SES historically is not exposed to the longer run-time SCSI requests that this behavior might conflict with, for example, tape rewind or obsoleted 'seek' requests.)

- Fixed a potential controller hang issue when hot-add and hot-removal activity is occurring during controller boot up.
    - Root Cause: During boot, the hotplug handling context incorrect initialization of a loop and missing event post-processing logic caused the hotplug handling context to become stuck waiting for hot-removal handling to execute, which had already occurred.
    - Fix: The loop initialization was corrected and appropriate post-processing of events was moved such that it now applies all event types.
    - Risk: Low. (Exposure is only to configurations with unstable hardware links in which a device is repeatedly linking up and then dropping the link while the controller is booting.)

- Fixed an issue where SSD IOBypass is not re-enabled if there are multiple volumes in the same array.
    - Root Cause: When there are multiple volumes in the same array, a logic error was resulting in the last volume in the set being the only one to have SSD IOBypass enabled during initial driver loading. In some cases, a subsequent update to IOBypass status could cause all volumes to be appropriately enabled.
    - Fix: The logic errors in the SSD IOBypass state change flow were simplified and updated to ensure all of the volumes in an array are appropriately updated.
    - Risk: Low

- Fixed an issue where the serial output buffer could return an incorrect data size if the entire log has been filled and wrapped around.
    - Root Cause: The code calculating the buffer size incorrectly assumed the buffer would not be full.
    - Fix: Corrected the logic to properly compute the buffer size to account for the wrap-around case.
    - Risk: Low

- Fixed an issue where controller background processes may be sluggish or appear hung if there are conditions causing RAID metadata to be updated repeatedly in a short period of time.
    - Root Cause: The RAID metadata update may be deferred to the background context to allow the I/O or Error handling paths to continue to progress. The background process was using a do-while loop to repeat the metadata update in case a second event had occurred while the previous was being saved. If events continued to cause metadata updates to be prompted, this code would stay in this loop.
    - Fix: Added a logic to break out of the loop and service other background operations if the metadata update has occurred three times in a row. The deferred metadata update will still take place the next time the background process is scheduled to run.
    - Risk: Low

- Fixed a lock up when there is no available logical requests.
    - Root Cause: There are quite a number of logical requests consumed but not dispatched in coalesce split logic.
    - Fix: Speed up dispatching of logical requests in coalesce logic.
    - Risk: Low

- Fixed an issue where controller would lock up after executing arcconf Save Support Archive.
  - Root Cause: The firmware was accessing an invalid address while transferring the controller logs to ARCCONF. The invalid address was due to a calculation error in the firmware, which didn’t factor the end of the memory region correctly.
  - Fix: The firmware calculates the end of the buffer correctly and also ensure the buffer wrap around conditions are taken into account.
  - Risk: Low
- Fixed an issue with retrieving and saving the UBM Host Facing Connector (HFC) ID.
  - Root Cause: HFC ID is not set up correctly when sending out the UBM command.
  - Fix: HFC is saved and proper HFC index is used when sending out the UBM command.
  - Risk: Low
- Fixed an issue of "Unknown" form factor reported for the SCSI block device.
  - Root Cause: Firmware incorrectly read 8-bits for the "nominal form factor" byte from SCSI Block Device Characteristics VPD page (0xB1) that is used for drive form factor identification.
  - Fix: Read only the lower 4-bits that are the "nominal drive format" information from SCSI Block Device Characteristics VPD page (0xB1).
  - Risk: Low
- Fixed a potential LUN reset occurring on the controller's virtual SEP device due to a timeout for the Notification On Event (0x40) command.
  - Root Cause: While processing a Notification On Event command and there is no event to be notified to the host, firmware will put this request in one of the event consumer entries with a timeout value set to 5 minutes and monitor it in the background task. If there is no event and the timeout value is expired for the Notification On Event command, firmware will trigger an event log so the event command can be completed to the host. In this case, the event log cannot add this pending consumer event into the queue to be completed which results in the timeout and LUN Reset to the controller's virtual SEP device.
  - Fix: Firmware will make sure the Notification On Event command completes and posts back to the host within the timeout value.
  - Risk: Low
- Fixed a potential controller 0x1A91 lockup during a workload to a device configured in a RAID volume which is in process of failing.
  - Root Cause: A drive encountering persistent timeouts is being marked as failed. At the same time the drive is marked as failed there is a race condition between hotplug context and other RAID contexts where some of the contexts may still be attempting to use the device, which slows down the drive failure and I/O retry and error handling processes.
  - Fix: Added a flag to indicate earlier that the device state is "failing" that can be set immediately on entry into the drive failure routine such that other contexts checking device status will immediately understand the device is not usable. This flag is cleared upon the device reaching the terminal "failed" status after the failure information has been saved.
  - Risk: Low
- Fixed an sequential I/O performance issue.
  - Root Cause: When internal controller memory is consumed with many coalesced IO's that have not completed, then new incoming IO's are not coalesced.
  - Fix: When internal controller memory is consumed wait for a short amount of time, then retry coalescing the new IOs to increase performance.
  - Risk: Low

2.2.2 UEFI Fixes

Note: Microsoft signed and secure boot is supported**.**

2.2.2.1 Fixes and Enhancements for UEFI Driver 1.3.14.5/Legacy BIOS 1.3.14.2

This release includes the following legacy BIOS fixes and enhancements:

- Fixed an issue where the legacy option ROM displays command error during boot.
  - Root Cause: Wrong parsing of device enumeration data causes a command error due to a command sent to a non-existing device.
  - Fix: Corrected device enumeration data parsing.
  - Risk: Low
- Fixed an issue legacy utility Ctrl-A hangs after entering **Create Array** menu.
  - Root Cause: Out of bound memory access due to mismatch in command buffer format for Sense Feature data.
  - Fix: Updated command buffer structure as per latest specification for Sense Feature data command.
  - Risk: Low

This release includes the following UEFI fixes and enhancements:

- Improved HII to dispaly more accurate drive sizes by including more digits after a decimal point.
- Added HII Save Support Archive option under Administration menu to capture controller persistent event logs in both raw and decoded format.
- Added driver message string for driver health protocol and new device states when encryption is enabled on SED devices whose encryption is not managed by the controller.
- Fixed an issue where there is no option to ignore and proceed when controller health status is set to configuration required state.
  - Root Cause: When driver health is set to configuration required state, there is user action required to resolve the issue. System may not proceed with boot until the status is changed from configuration required to healthy state. Additional option is required for the cases when user want to ignore the configuration required health state and proceed with boot.
  - Fix: Added option in driver health HII form to ignore the driver health status which overrides the configuration required state to healthy.
  - Risk: Low
- Fixed an issue where OS filesystem is not available after making configuration changes in HII.
  - Root Cause: Configuration changes in HII re-installs block I/O protocol of all handles. The change in handle does not synchronize with the filesystem drivers and causes issues in loading existing OS.
  - Fix: Install or uninstall Block IO protocol only for the handles which are affected by the configuration change.
  - Risk: Low
- Fixed an issue where no warning help text is displayed if cache backup power source is absent when creating a logical drive for the first time.
  - Root Cause: Help text construction for the Acceleration method option does not consider the initial state of no battery write cache.
  - Fix: Consider the initial state of no battery write cache while generating help text for the Acceleration method option.
  - Risk: Low
- Fixed an issue where firmware version is listed as null string in the firmware information page.
  - Root Cause: Command to get firmware version failed due to insufficient buffer size.
  - Fix: Allocated a larger buffer to correctly get the firmware version.

    - Risk: Low
- Fixed an issue where the HII identify disk operation does not provide information on the duration left.
    - Root Cause: No information provided on duration left for identify disk operation.
    - Fix: Added identify disk menu to show remaining duration in seconds if there is an existing operation in progress.
    - Risk: Low
- Fixed an issue where UEFI Self Certification Tests SCT fails for Component name2 protocol.
    - Root Cause: GetControllerName of Component name2 protocol does not validate input language. SCT fails when incorrect language is provided as input.
    - Fix: Supported language validation added for GetControllerName of Component name2 protocol.
    - Risk: Low
- Fixed an issue where error code is not within Independent Hardware Vendor (IHV) range for the Driver Health protocol.
    - Root Cause: Error code returned for Driver Health protocol is not in specified IHV range as mentioned in UEFI specification.
    - Fix: Error code assigned from the permitted IHV range as per UEFI specification.
    - Risk: Low
- Fixed an issue where the complete controller name is not provided in driver health messages.
    - Root Cause: Only short controller name is provided in driver health messages.
    - Fix: Dynamically construct complete controller name and provide it in driver health messages.
    - Risk: Low
- Fixed an issue where failed HBA drives are not shown in HII.
    - Root Cause: Failed HBA devices are not displayed in HII and driver health messages.
    - Fix: Populate and provide available information on failed devices in HII and driver health messages.
    - Risk: Low
- Fixed an issue where error was observed while selecting edit maxCache Logical drive option when maxCache logical drive is in Degraded mode.
    - Root Cause: Mismatch in possible options and set value for "Modify Cache write policy".
    - Fix: Updated the config variable with proper values to avoid mismatch against possible options.
    - Risk: Low
- Fixed an issue with port discovery protocol changes does not provide the status to inform user that reboot is required.
    - Root Cause: Port discovery protocol operation status only shows if it is success or failed.
    - Fix: Added reboot required message in final status of port discovery protocol settings.
    - Risk: Low

2.2.3 Driver Fixes

2.2.3.1 Fixes and Enhancements for Linux Driver Build 2.1.12-055

This release provides the following fixes and enhancements.

- Fixed an issue where duplicate device nodes for Ultrium tape drive and medium changer are being created.
    - Root Cause: The Ultrium tape drive is a multi-LUN SCSI target. It presents a LUN for the tape drive and a second LUN for the medium changer. The controller firmware lists both LUNs in the RPL results. As a result, the SmartPQI driver exposes both devices to the OS. Then, the OS does its normal device discovery via the SCSI REPORT LUNS command, which causes it to re-discover both devices a second time, which results in the duplicate device nodes. This broken

behavior was masked by an earlier SmartPQI bug that caused the OS to skip its device discover for this type of device. This issue was fixed by a recent change to SmartPQI to report more accurate information about SAS initiator and target port protocols.
    - Fix: When the OS re-discovers the two LUNs for the tape drive and medium changer, the driver recognizes that they have already been reported and blocks the OS from adding them a second time.
    - Risk: Low
- Fixed an issue where in some situations when the driver takes the controller offline, a kernel crash can occur.
    - Root Cause: While taking controller offline, it is possible for the driver to fail IOs which have already been completed by the OS, causing a kernel crash.
    - Fix: If the device has been marked offline by the OS, do not fail IOs pertaining to that device because IOs may have been previously completed.
    - Risk: Low
- Fixed an issue where OS boot may fail during logical volume rebuild.
    - Root Cause: The driver was reading a controller register too soon after writing to that register.
    - Fix: Moved the delay in the register polling loop to the beginning of the loop to ensure there is always a delay between writing the register and reading it.
    - Risk: Low
- Fixed an issue where using sysfs to temporarily remove a device does not work.
    - Root Cause: Defining `slave_destroy` causes SCSI mid-layer to call into the `slave_destroy` to remove the device from the SCSI table. The `slave_destroy` is not complete.
    - Fix: Remove `slave_destroy`.
    - Risk: Low
- Fixed an issue where during system hibernation, driver frees all the irqs, disables MSIx interrupts and requests legacy INTx interrupt. When driver invokes `request_irq()`, OS returns—EINVAL.

  For example, smartpqi 0000:b3:00.0: irq 191 init failed with error -22 genirq: Flags mismatch irq 34. 00000080 (SmartPQI) vs. 00000000 (i40e-0000:1a:00.0:misc).
    - Root Cause: The first argument `irq` in `request_irq` is not correct.
    - Fix: If the Interrupt mode is being set to INTx, use PCI device's `irq` as first parameter to `request_irq()`.
    - Risk: Low
- Due to a change in the SCSI mid-layer, some Linux distributions may take a long time to come up if the system is rebooted while a hard disk(s) is being sanitized. This has been observed on RHEL 7.9/RHEL8.3 and SLES 15SP2.
    - Root Cause: During boot-up, some OSes appear to hang when there are one or more disks undergoing sanitize. According to *SCSI SBC4 specification section 4.11.2 Commands* allowed during sanitize, some SCSI commands are permitted, but read/write operations are not. When the OS attempts to read the disk partition table a `CHECK CONDITION ASC 0x04 ASCQ 0x1b` is returned which causes the OS to retry the read until sanitize has completed. This can take hours.
    - Fix: Add in a Test Unit Ready to HBA disks and do not present them to the OS if `0x02/0x04/0x1b` (sanitize in progress) is returned.
    - Risk: Low
- Fixed an issue with request leakage, performance drop, and system crash.
    - Root Cause: The issue happens in a max configuration where heavy I/O load is exercised with occasional LUN resets on the exposed devices. While failing queued IOs in the TMF path, there was a request leak and hence stale entries in request pool with reference count being non-zero. In the shutdown path, there is a `BUG_ON` to catch stuck I/O either in the firmware or in the driver. The unfreed stale request caused system crash. If the above situation keeps occurring then the I/O request pool keeps leaking and there could be a significant performance drop.

- Fix: The driver now frees the leaked request properly in the TMF path while failing outstanding requests.
- Risk: Low

2.2.3.2 Fixes and Enhancements for FreeBSD Driver Build 4130.0.1008

This release provides the following enhancements and fixes:

- Fixed an issue of OS booting into Single-user mode in FreeBSD13.0 when it is installed on SmartPQI controller.
  - Root Cause: The OS was not booting because of the drive order changes which is due to logical volume creation/deletion.
  - Fix: Updated `README.txt` document—Modify the `/etc/fstab` with specified boot drive if the OS is booting in Single-user mode.
  - Risk: Low
- Fixed an issue when drives are added/removed/offline, HBA devices, and controllers are displayed as a default RAID 0 value.
  - Root Cause: There is no check for physical devices or controllers before printing display info.
  - Fix: Modify the messaging so that it prints differently based on physical devices and controllers to identify them accordingly.
  - Risk: Low
- Fixed an issue where uninitialized CCB structure causes undefined behavior when it is shared with the CAM layer.
  - Root Cause: CCB is being used without clearing stack values.
  - Fix: Clear CCB before it is used.
  - Risk: Low
- Fixed an issue in which the driver is disabling drives if it detects the controller going offline but there is no information that logs the controller lockup code when it is offline.
  - Root Cause: The driver is not displaying the controller lockup code.
  - Fix: Display the lockup code in driver logs. Also, the timer handler is disabled when the controller is offline to prevent a system crash in the event of delay during post memory deletion.
  - Risk: Low
- Fixed an issue where a kernel panic is occasionally encountered in function `pqisrc_build_scsi_cmd_raidbypass` during a test that repeatedly cycles IOBypass feature on and off for RAID volumes used for VM virtual disks, undergoing I/O loading.
  - Root Cause: Device's RAID map pointer is null for a short time while disabling the IOBypass.
  - Fix: Check for NULL RAID map and return `PQI_STATUS_FAILURE` if a null is found, redirecting the request to the RAID path.
  - Risk: Low

2.2.3.3 Fixes and Enhancements for Solaris Driver Build 4120.0.1005

This release provides the following enhancements and fixes:

- Fixed an issue when drives are added/removed/offline, HBA devices, and controllers are displayed as a default RAID 0 value.
  - Root Cause: There is no check for physical devices or controllers before printing display information.
  - Fix: Modify the messaging so that it prints differently based on physical devices and controllers to identify them accordingly.
  - Risk: Low

2.2.3.4 Fixes and Enhancements for Windows Build 1010.6.0.1025

This release provides the following enhancements and fixes:

- Fixed an issue where IOBypass request is failing when controller is not expecting a multi-column write request via the IOBypass path.
  - Root Cause: Driver was incorrectly assuming all 2 drive RAID 1 writes are single-column and incorrectly allowing the request to be submitted via IOBypass path, thus causing controller to fail the IOBypass request. The driver was not checking any write request transfers sizes to see if it was a Multi-column write. Thus the request was incorrectly submitted via the IOBypass path.
  - Fix: Fixed driver logic to always check all request transfers sizes to see if they qualify as a single-column write and if not, reject the requests as candidates for the IOBypass path. The 1 MB sequential write to two drive—128K strip RAID 1 volume is now rejected as a candidate for IOBypass path and thus submitted via RAID path.
  - Risk: Low
- Fixed an issue where the OS would possibly fail to boot.
  - Root Cause: Driver can fail to load because writing to Administrator Queue Configuration Function Register and then reading the register without delay can give erroneous stale status.
  - Fix: Added a 1 ms=1000 µs delay after writing to Administrator Queue Configuration Function register, but before polling the status.
  - Risk: Low

2.2.3.5 Fixes and Enhancements for VMware Driver Build 4150.0.119

This release provides the following enhancements and fixes:

- Fixed an issue where device quiesce process is missing a key step to make sure any in-flight interrupts are processed before the call to disable interrupts on the device.
  - Root Cause: An OS API for synchronizing and flushing interrupts was not being called.
  - Fix: Add call to the OS API for flushing any pending interrupts.
  - Risk: Low
- Fixed an issue where ESXi server PSOD due to page fault in TMF handler.
  - Root Cause: During "virtual reset" TMF, driver iterates through IO structures and will issue aborts for all pending IOs. While framing an abort request, the driver uses the device structure pointer from the IO structure. If IO associated with the IO structure completes in parallal, the device structure pointer might reset to NULL, which will result in a page fault.
  - Fix: Use device structure pointer given by the OS TMF handler.
  - Risk: Low
- Fixed an issue where ESXi server PSOD during boot.
  - Root Cause: An inquiry command to one of the drives is timing out and the OS issues a TMF abort. During TMF completion, the driver will print the TMF status. Internally this uses the driver private structure which was not set when framing the TMF request.
  - Fix: Set driver private structure pointer when framing TMF request.
  - Risk: Low
- Fixed an issue where ESXi server issues NMI and PSOD due to lack of heartbeat.
  - Root Cause: Driver acquires a lock to get a slot on the inbound queue. All cores might end up in using the same inbound queue if the number of SCSI completion worlds are less than the number of cores. In most cases, the number of scsi completion worlds are the same as number of sockets and most servers have 1 to 2 sockets. This might cause lock congestion as many threads will be trying to acquire the same lock. Driver uses a custom lock that does a tight busy wait if the lock is not available. This will cause the IO submission thread to hold the CPU core

and the ESXi heartbeat thread might not get a chance to run for a long time. This will result in ESXi issuing a NMI and PSOD the server.
    - Fix: Use the spinlock in submission path.
    - Risk: High
- Fixed an issue when drives are added/removed/offline, HBA devices and controllers are displayed as a default RAID 0 value.
    - Root Cause: There is no check for physical devices or controllers before printing display info.
    - Fix: Modify the messaging so that it prints differently based on physical devices and controllers to identify them accordingly.
    - Risk: Low
- Fixed an issue where ESXi 7.0 u2 system hangs during driver load and PSOD was observed after a long wait time.
    - Root Cause: Driver uses an infinite timeout for sending internal commands related to event configuration during driver initialization stage.
    - Fix: Added timeout for sending internal commands related to event configuration during driver init stage.
    - Risk: Low
- Fixed an issue where after installing ESXi 7.0 U2 on server with 64-Core Processor, ESXi PSOD was observed with stack trace pointing to interrupt acknowledgment function of SmartPQI driver.
    - Root Cause: SmartPQI driver creates maximum of 64 outbound queues. Queues are created based on number of cores/scsi completion worlds and MSIX availability. Driver will create a maximum of 64 queues and 64 handlers which are registered. Driver handler data array size was 63 instead of 64 and resulting in PSOD.
    - Fix: Corrected the handler data array size to be 64.
    - Risk: Low
- Fixed an issue where a PSOD is occasionally encountered in function `pqisrc_build_scsi_cmd_raidbypass` during a test that repeatedly cycles IObypass feature on and off for RAID volumes used for virtual machine disks, undergoing IO loading.
    - Root Cause: Device's raid map pointer is null for a short time while disabling the IOBypass.
    - Fix: Check for NULL raid map and return PQI_STATUS_FAILURE if a null is found, redirecting the request to RAID path.
    - Risk: Low
- Fixed an issue where driver produces many abort messages when device resets are occurring. During device reset testing, this may produce thousands of log lines in a short amount of time.
    - Root Cause: All IO requests pending to a device will be aborted by an incoming device reset request. For devices capable of high queue depths, this could be tens or hundreds of individual abort requests, per device Reset.
    - Fix: Change logging level for this type of message from WARN to INFO, so that it is only printed when someone purposefully changes the driver's logging level to do debug or analysis.
    - Risk: Low
- Fixed an issue where PSOD is detected during array creation/deletion and followed by driver unload.
    - Root Cause: SmartPQI driver maintains a linked list of hot-removed devices. Whenever a new device is present, driver checks whether that device is already present in the remove_device_list, and if it is present, driver moves that device from remove_device_list to actual device list. Entries in the remove_device_list will be reviewed in fixed time interval and list will be updated by removing device which has been in that list for more than 20 minutes (to handle vSAN hotplug test).

      During driver unload, driver checks for any devices present in the list and does the cleanup (free the device memory). PSOD stack trace indicates an invalid device memory freeing during this cleanup.
    - Fix: Remove entry from the device list whenever the device memory is freed up.

  - Risk: Medium
- Fixed an issue where PSOD is encountered when `pqisrc_send_aio_tmf` is called from task management handlers.
  - Root Cause: Code to use new IOBypass abort task IU referred to device pointer of abort request rather than dev pointer of request that is being aborted, to check for RAID type.
  - Fix: Replace pointer with aborted request's dev pointer.
  - Risk: Low

2.2.4 Management Software Fixes

2.2.4.1 Fixes and Enhancements for Arcconf/maxView Build B24308

This release includes the following fixes and enhancements for arcconf/maxView:

Important:

Microchip strongly recommends the users of maxView prior to 2.06 version to update to the latest version of the tools to avoid a security vulnerability that has since been resolved.

- RAID level name changed from 'RAID1 ADM' to 'RAID1 Triple' in maxView/arcconf.
- Support to add firmware event log buffer as part of Support Archive.
- Added passive SED support.
- Support for controller to report failed physical devices in the configuration.
- Fixed an issue where Remote arcconf has OpenSSL security vulnerabilities.
  - Root Cause: Remote arcconf uses older version of open source library OpenSSL which had security vulnerabilities.
  - Fix: Added changes to Remote arcconf by adding the latest version of OpenSSL library that had addressed the security vulnerabilities.
  - Risk: Low
- Fixed an issue where maxView does not display the configuration properly when a physical device has a model name with quotation marks ["] in it.
  - Root Cause: Having quotation marks [“] in the physical device model name has corrupted the JSON format of the configuration making maxView unable to display it properly.
  - Fix: Added changes to JSON configuration creation to address characters such as [“].
  - Risk: Low
- Fixed an issue in maxView to move the array using combination of unassigned and array member devices.
  - Root Cause: maxView was listing only the unassigned physical device for move array operation using the same drive type.
  - Fix: Added changes to list with all the unassigned physical devices along with array members for the move array operation.
  - Risk: Low
- Fixed an issue in maxView where move logical device option is disabled when trying to move from SATA to SATA SSD device type.
  - Root Cause: Move logical device operation in maxView is blocked for non-SSD logical device to devices of type SSD.
  - Fix: Added changes to support both SSD and non-SSD devices during move logical device operation in maxView.
  - Risk: Low

2.3 Limitations

2.3.1 Firmware Limitations

2.3.1.1 Limitations for Firmware Release 4.11 B0

This release includes the following firmware limitations:

- A firmware update causes the UART log buffer (Serial Output Buffer) to be reinitialized, since the DDR gets reinitialized.
  - Workaround: None
- SATA drives attached to a non-Microchip expander may get into a failed state when upgrading the controller firmware from previous releases to this release due to the expander not clearing STP affiliation.
  - Workaround: Power cycle the expanders to clear the STP affiliation.
- A rare corner-case scenario where controller may hang during expander firmware update on multi-level expander/SEP device topology along with I/Os.
  - Workaround: After the enclosure firmware update, avoid enclosure Reset. It is recommended to download the new firmware and perform manual power cycle. This issue is intermittent and can cause a hang, a reboot is known to fix such instances.

    Note:

    This issue was mostly seen when using Linux OS.
- Controller cache will not be converted into 100% read cache, if any backup power source cable error, charge or charge timeout error occurs when expansion or transformation task is active.
  - Workaround: None
- Physical drive firmware update will not succeed when initiated through Out-Of-Band MCTP host transport.
  - Workaround: Update physical drive firmware through In-Band methods using OS tools.
- maxCache reconfiguration operations, such as changing from writeback to writethrough or maxCache cache delete, may not complete successfully.
  - Workaround: Reboot controller and reconfigure maxCache cache, as required.

2.3.1.2 Limitations for Firmware Release 1.32 Build 0

- Firmware release 1.32b0 may become unresponsive while attempting to flash firmware or execute other RAID logical volume operations.
  - Description: Refer to entry "Fixed an issue where firmware may become unresponsive while attempting to flash firmware or execute other RAID logical volume operations" in the Firmware fixes section.
  - A fix for this issue is available in the 1.60 B0 firmware release. If a firmware flash failure is occurring, try the following workarounds:
    - Workaround: If there are no target devices (expanders or drives) attached to the controller, attach a target device to the controller and try the host management operation again.
    - Workaround: If the system is operating using UEFI, the HII tool can be used to flash the firmware to this release as outlined in the *Microchip SmartIOC 2100/SmartROC 3100 Installation and User's Guide (ESC-2170577),* appendix entry “Updating the SmartIOC 2100/SmartROC 3100 Controller Firmware”.
    - Workaround: If there are target devices attached to the controller and this issue occurs or none of the workarounds can be used, contact Microchip Support.

2.3.2 UEFI Limitations

2.3.2.1 Limitations for UEFI Build 1.3.14.5/Legacy BIOS Build 1.3.14.2

There are no known limitations for this release.

2.3.3 Driver Limitations

2.3.3.1 Limitations for Linux Driver Build 2.1.12-055

This release has the following Linux limitations:

- On AMD/RHEL 7.9 systems, the system might panic due to the a bug in the IOMMU module. For details, see *https://lore.kernel.org/linux-iommu/20191018093830.GA26328@suse.de/t/*
  - Workaround: Disable the IOMMU setting option in BIOS.

2.3.3.2 Limitations for Windows Driver Build 1010.6.0.1025

There are no known limitations for this release.

2.3.3.3 Limitations for FreeBSD Driver Build 4130.0.1008

There are no known limitations for this release.

2.3.3.4 Limitations for Solaris Driver Build 4120.0.1005

There are no known limitations for this release.

2.3.3.5 Limitations for VMware Driver Build 4150.0.119

There are no known limitations for this release.

2.3.4 Management Software Limitations

2.3.4.1 Limitations for Arcconf/maxView Build B24308

There are no known limitations for this release.

2.3.5 Hardware Limitations

This release includes the following hardware limitations:

- Two Wire Interface (TWI) address conflicts can cause system DDR memory to not be discovered.
  - Description: The SmartRAID 3100 and SmartHBA 2100 boards include two TWI targets on the host-facing SMBUS interface with the following slave addresses:
    - 0xA0 – Field Replaceable Unit (FRU) SEEPROM
    - 0xDE – PBSI (default)

      According to the JEDEC specification, the default TWI addresses for the DDR SPD is 0xA0-0xAE (the spec uses 7 bit addressing which is 0x50-0x57). On platform system board designs with SMBUS wiring that has both PCIe slots and DDR slots shared on the same TWI bus, the TWI devices for the DDR and Smart controller are exposed to address conflicts which can result in the system memory not being discovered. The Smart controller PBSI interface defaults to a value of 0xDE (0x6F in 7-bit addressing) and is not a problem unless

it is changed to an address that conflicts with the JEDEC defined values. The Smart controller FRU SEEPROM is hardwired to 0xA0.

- Workaround: None available. If this issue is encountered, contact your Microchip support engineer to determine the next steps for your system.
- Performance with workaround: Not applicable
- Performance without workaround: Not applicable

3 Updating the Controller Firmware

This section describes how to update the board's firmware components to the latest release.

3.1 Updating the Controller Firmware

This procedure describes how to prepare your board to be programmed with the latest firmware.

Note:

1. If the running firmware is older than 1.98 and a transformation is in progress, complete the transformation before proceeding with the following steps to upgrade the firmware.
2. Complete these procedures exactly as described for proper functionality. If you do not follow all of the steps correctly, you could encounter unusual runtime behavior.

Flashing the board to the latest firmware:

This section describes how to update all the firmware components on Adaptec controller boards to the latest release.

If the controller is currently running 1.60 b0 firmware or newer, follow these steps:

1. **Mandatory:** Flash the target with the provided " SmartFWx100.bin" image with arcconf/maxView software.
2. **Mandatory:** Use the OS shutdown/restart operation to gracefully reboot the system to complete the firmware update process.

   Note:

   After completing the firmware update, if the firmware version is still showing the prior version, retry the firmware update steps.

If the controller is currently running 1.32 b0 firmware, follow these steps:

1. **Mandatory:** Flash the target with the provided “SmartFWx100.bin" image with arcconf/maxView software.
   - If the arcconf/maxView software becomes unresponsive or hangs then power cycle the system to recover and refer to firmware limitation section *Limitations for Firmware Release 1.32 Build 0* on page 19.
2. **Mandatory:** If flashing completes, use the OS shutdown/restart operation to gracefully reboot the system to complete the firmware update process.

   Note:

   After completing the firmware update, if the firmware version is still showing the prior version, retry the firmware update steps.

If the controller is currently running 1.04 b0 firmware, follow these steps:

1. **Mandatory:** Flash the controller with the provided “SmartFWx100_ v1.29_b314.bin" image with arcconf/maxView software.
2. **Mandatory:** Reboot the system to refresh all components**.**
3. **Mandatory**: Flash the target with the provided " SmartFWx100.bin" image with arcconf/maxView software.
4. **Mandatory**: Use the OS shutdown/restart operation to gracefully reboot the system to complete the firmware update process.

At this point, the controller would be updated and would be ready to use. Install the SmartPQI driver and the latest version of the Arcconf/maxView management utility to monitor and configure the controller.

Note: Downgrading firmware could lead to unexpected behavior due to an incompatibility in SEEPROMs between this release and the prior release.

4 Installing the Drivers

See the "Microchip Adaptec® SmartRAID 3100 Series and SmartHBA 2100 Series Host Bus Adapters Installation and User's Guide (ESC-2171547)" for complete driver installation instructions.

5 The Microchip Web Site

Microchip provides online support via our web site at *http://www.microchip.com/*. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user guides and hardware support documents, latest software releases, and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, and Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors, and factory representatives

5.1 Customer Change Notification Service

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions, or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at *http://www.microchip.com/*. Under “Support”, click on “Customer Change Notification” and follow the registration instructions.

5.2 Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: *http://www.microchip.com/support*

5.3 Microchip Devices Code Protection Feature

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

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip’s Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be

a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

5.4 Legal Notice

Information contained in this publication regarding device applications and the like 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. 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 ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. 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.

5.5 Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, AVR, AVR logo, AVR Freaks, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, Heldo, JukeBlox, KeeLoq, Kleer, LANCheck, LINK MD, maXStylus, maXTouch, MediaLB, megaAVR, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, Prochip Designer, QTouch, SAM-BA, SpyNIC, SST, SST Logo, SuperFlash, tinyAVR, UNI/O, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, EtherSynch, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and Quiet-Wire are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, INICnet, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, memBrain, Mindi, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2021, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

5.6 Quality Management System Certified by DNV

ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and

analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

5.7 Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office	**Australia - Sydney**	**India - Bangalore**	**Austria - Wels**
2355 West Chandler Blvd.	Tel: 61-2-9868-6733	Tel: 91-80-3090-4444	Tel: 43-7242-2244-39
Chandler, AZ 85224-6199	**China - Beijing**	**India - New Delhi**	Fax: 43-7242-2244-393
Tel: 480-792-7200	Tel: 86-10-8569-7000	Tel: 91-11-4160-8631	**Denmark - Copenhagen**
Fax: 480-792-7277	**China - Chengdu**	**India - Pune**	Tel: 45-4450-2828
Technical Support:	Tel: 86-28-8665-5511	Tel: 91-20-4121-0141	Fax: 45-4485-2829
http://www.microchip.com/	**China - Chongqing**	**Japan - Osaka**	**Finland - Espoo**
support	Tel: 86-23-8980-9588	Tel: 81-6-6152-7160	Tel: 358-9-4520-820
Web Address:	**China - Dongguan**	**Japan - Tokyo**	**France - Paris**
www.microchip.com	Tel: 86-769-8702-9880	Tel: 81-3-6880- 3770	Tel: 33-1-69-53-63-20
Atlanta	**China - Guangzhou**	**Korea - Daegu**	Fax: 33-1-69-30-90-79
Duluth, GA	Tel: 86-20-8755-8029	Tel: 82-53-744-4301	**Germany - Garching**
Tel: 678-957-9614	**China - Hangzhou**	**Korea - Seoul**	Tel: 49-8931-9700
Fax: 678-957-1455	Tel: 86-571-8792-8115	Tel: 82-2-554-7200	**Germany - Haan**
Austin, TX	**China - Hong Kong SAR**	**Malaysia - Kuala Lumpur**	Tel: 49-2129-3766400
Tel: 512-257-3370	Tel: 852-2943-5100	Tel: 60-3-7651-7906	**Germany - Heilbronn**
Boston	**China - Nanjing**	**Malaysia - Penang**	Tel: 49-7131-67-3636
Westborough, MA	Tel: 86-25-8473-2460	Tel: 60-4-227-8870	**Germany - Karlsruhe**
Tel: 774-760-0087	**China - Qingdao**	**Philippines - Manila**	Tel: 49-721-625370
Fax: 774-760-0088	Tel: 86-532-8502-7355	Tel: 63-2-634-9065	**Germany - Munich**
Chicago	**China - Shanghai**	**Singapore**	Tel: 49-89-627-144-0
Itasca, IL	Tel: 86-21-3326-8000	Tel: 65-6334-8870	Fax: 49-89-627-144-44
Tel: 630-285-0071	**China - Shenyang**	**Taiwan - Hsin Chu**	**Germany - Rosenheim**
Fax: 630-285-0075	Tel: 86-24-2334-2829	Tel: 886-3-577-8366	Tel: 49-8031-354-560
Dallas	**China - Shenzhen**	**Taiwan - Kaohsiung**	**Israel - Ra'anana**
Addison, TX	Tel: 86-755-8864-2200	Tel: 886-7-213-7830	Tel: 972-9-744-7705
Tel: 972-818-7423	**China - Suzhou**	**Taiwan - Taipei**	**Italy - Milan**
Fax: 972-818-2924	Tel: 86-186-6233-1526	Tel: 886-2-2508-8600	Tel: 39-0331-742611
Detroit	**China - Wuhan**	**Thailand - Bangkok**	Fax: 39-0331-466781
Novi, MI	Tel: 86-27-5980-5300	Tel: 66-2-694-1351	**Italy - Padova**
Tel: 248-848-4000	**China - Xian**	**Vietnam - Ho Chi Minh**	Tel: 39-049-7625286
Houston, TX	Tel: 86-29-8833-7252	Tel: 84-28-5448-2100	**Netherlands - Drunen**
Tel: 281-894-5983	**China - Xiamen**		Tel: 31-416-690399
Indianapolis	Tel: 86-592-2388138		Fax: 31-416-690340
Noblesville, IN	**China - Zhuhai**		**Norway - Trondheim**
Tel: 317-773-8323	Tel: 86-756-3210040		Tel: 47-72884388
Fax: 317-773-5453			**Poland - Warsaw**
Tel: 317-536-2380			Tel: 48-22-3325737
Los Angeles			**Romania - Bucharest**
Mission Viejo, CA			Tel: 40-21-407-87-50
Tel: 949-462-9523			**Spain - Madrid**

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Fax: 949-462-9608			Tel: 34-91-708-08-90
Tel: 951-273-7800			Fax: 34-91-708-08-91
Raleigh, NC			**Sweden - Gothenberg**
Tel: 919-844-7510			Tel: 46-31-704-60-40
New York, NY			Sweden - Stockholm
Tel: 631-435-6000			Tel: 46-8-5090-4654
San Jose, CA			**UK - Wokingham**
Tel: 408-735-9110			Tel: 44-118-921-5800
Tel: 408-436-4270			Fax: 44-118-921-5820
Canada - Toronto			
Tel: 905-695-1980			
Fax: 905-695-2078			