

# SNAP Release Notes for 2.5.6

17 February 2015

## Table of Contents

---

Introduction .....	1
How to use this document .....	1
Numbering of SNAP Firmware Releases .....	2
The “synchronized releases” period: Versions 2.0.0 - 2.4.9 .....	2
The “transitional” period: Versions 2.4.10 - 2.4.11 .....	2
The “independent releases” period: Versions 2.4.11 and up .....	2
Recommended Firmware Versions .....	3
Per-Platform Timelines .....	4
ATMEL ATmega128RFA1 Family .....	4
Synapse RF200 .....	5
Synapse SS200 .....	6
Synapse SM200 .....	7
Synapse RF200P81 .....	8
Synapse RF266 .....	9
ATmega128RFR2 .....	10
ATmega1284RFR2 .....	11
Synapse SM220 .....	12
Synapse RF220 .....	13
Synapse RF100 .....	14
Freescale MC1321x Chip .....	14
Composite Timeline - Summary .....	15
2.4.35 .....	15
2.4.36 .....	15
2.4.37 .....	15
2.5.0 .....	15
2.5.1 .....	15
2.5.2 .....	15
2.5.3 .....	16
2.5.3 .....	16
2.5.4 .....	16
2.5.5 .....	16
2.5.6 .....	16
Composite Timeline - Detailed .....	17
A few words about platforms .....	17
Version 2.4.35 .....	17
Version 2.4.36 .....	18
Version 2.4.37 .....	19

Version 2.5.0 .....	20
Version 2.5.1 .....	21
Version 2.5.2 .....	21
Version 2.5.3 .....	22
Version 2.5.4 .....	22
Version 2.5.5 .....	22
Version 2.5.6 .....	23
Installation and Upgrade Notes .....	23
Known Problems and Workarounds .....	26
Not all demo scripts work on all platforms.....	26
Not all SNAP engines work on all demonstration boards.....	26
Related Documents.....	26

## Document Revision History

---

Previous Version	Change	Page
10 October 2014	Initial draft version of 2.5.3 Release Notes	
15 October 2014	Cleanup of various WORD formatting issues	throughout
20 October 2014	Removing reviewer comments and final pagination	throughout
17 February 2015	Updating for the 2.5.6 release	throughout

## Introduction

---

This document describes the cumulative changes made to the SNAP Firmware since a previous formal release, version 2.4.34.

SNAP version 2.5.3 for ATMEL was released on 7 October 2014.

QA for SNAP version 2.5.3 for Freescale 9S08 platforms was aborted when an issue was found in the reliability of SNAPpy script upload. This issue was fixed and a candidate SNAP version 2.5.4 for Freescale 9S08 platforms entered QA.

QA for SNAP version 2.5.4 for Freescale was aborted when an issue with Packet Serial **affecting all SNAP Platforms** was discovered. The issue was fixed and candidate SNAP version 2.5.5 for ATMEL and Freescale 9S08 platforms entered QA.

SNAP 2.5.5 failed QA due to a single issue found with AES-128 over Packet Serial being non-functional. SNAP 2.5.6 release candidates were quickly created and QA testing resumed.

SNAP version 2.5.6 for ATMEL and Freescale 9S08 platforms was released on 17 February 2015.

For information regarding earlier releases of the SNAP Firmware (for example the SNAP 2.4 series), refer to the corresponding earlier release note documents.

## How to use this document

---

To better understand the logic behind the numbering of the different versions of SNAP, read the section **Numbering of SNAP Firmware Releases**. To quickly determine if you should upgrade the firmware on your particular SNAP nodes, look at the table in the **Recommended Firmware Versions** section of this document.

To better understand what software changes have been made for your particular platform, look for that platform's entry in the **Per-Platform Timelines** section.

For a high-level view of what has changed across all versions, look in the **Composite Timeline - Summary** section of this document.

For a more detailed composite timeline, look in the **Composite Timeline - Detailed** section of this document.

# Numbering of SNAP Firmware Releases

---

The naming conventions used in numbering SNAP releases have changed over the years. This section explains the changes.

## The “synchronized releases” period: Versions 2.0.0 - 2.4.9

SNAP releases up through version 2.4.9 were done in “batches”. If a new version was released, it was released on all SNAP platforms at the same time. This required *all* SNAP ports to be retested anytime *any one* of them was changed, and delayed release of some platforms while others “caught up”.

## The “transitional” period: Versions 2.4.10 - 2.4.11

There was a transitional period during which we attempted to keep the version numbers in lockstep, but the sheer number of new hardware platforms made this impractical. A few version numbers were released during this period (on different platforms) where you could not use the version number alone to determine “which code was newer” (between platforms).

## The “independent releases” period: Versions 2.4.11 and up

Starting with version 2.4.11, we instituted a new policy of advancing the SNAP revision number whenever a new version was released on any platform. This restores the strict ordering (even across platforms), but it means that there are *many* version numbers that do not exist for a given platform. For example, as of this writing, the latest version of SNAP for the Synapse RF100 is version 2.4.19, and the last released build before that was 2.4.9 – there is no 2.4.10, 2.4.11, 2.4.12, (etc.) for the RF100. Keep all of this in mind as you look through the following quick reference tables.

**NOTE** – SNAP version 2.5.6 is an exception to the pattern, in that we were able to do simultaneous releases on both the ATMEL and Freescale 9S08 platforms at the same time.

## Recommended Firmware Versions

The table below details the recommended firmware version for each SNAP capable hardware platform.

Platform	Recommended Version
Synapse RF100	2.5.6
Synapse RF200	2.5.6
Synapse SS200 (SNAP Stick 200)	2.5.6
Synapse SM200	2.5.6
Synapse RF200P81 (SM200 on a carrier board)	2.5.6
Synapse SM220	2.5.6
Synapse RF220	2.5.6
Synapse RF266	2.5.6
Synapse RF300	2.4.31
Synapse SM300	2.4.31
Synapse SM301	2.4.31
Synapse RF301	2.4.31
Synapse SM700	2.4.33
Freescale MC1321x	2.5.6
California Eastern Labs ZIC2410P1	2.4.13
California Eastern Labs ZIC2410P2	2.4.13
ATMEL ATmega128RFA1	2.5.6
ATMEL ATmega128RFR2	2.5.6 is recommended but is unverified
ATMEL ATmega1284RFR2	2.5.6 is recommended but is unverified
Silicon Laboratories Si1000 (900 MHz)	2.4.31
Silicon Laboratories Si1002 (868 MHz)	2.4.31
Freescale MC1322x	2.4.33
ST Microelectronics STM32W108xB	2.4.28

## Per-Platform Timelines

---

Each of the following tables details the development timeline for a particular SNAP hardware platform. SNAP versions prior to 2.4.35 are not shown, and if the first version listed for a particular platform is higher than 2.4.35, then that version represents the first version of SNAP to run on that hardware platform.

**NOTE** – Separate tables for every platform have been provided as a convenience (you can look up the *exact* platform you are interested in). It is true that in the current version of this document, all of the presented platforms are in sync, and so the tables are all identical. We expect that could change over time, and are setting up the document format to make it easy to update individual platforms in the future.

### ATMEL ATmega128RFA1 Family

Version	Comments
2.4.35	Link to Summary: <b>2.4.35</b> Link to Details: <b>Version 2.4.35</b>
2.4.36	Link to Summary: <b>2.4.36</b> Link to Details: <b>Version 2.4.36</b>
2.4.37	Link to Summary: <b>2.4.37</b> Link to Details: <b>Version 2.4.37</b>
2.5.0	Link to Summary: <b>2.5.0</b> Link to Details: <b>Version 2.5.0</b>
2.5.1	Link to Summary: <b>2.5.1</b> Link to Details: <b>Version 2.5.1</b>
2.5.2	Link to Summary: <b>2.5.2</b> Link to Details: <b>Version 2.5.2</b>
2.5.3	Link to Summary: <b>2.5.3</b> Link to Details: <b>Version 2.5.3</b>
2.5.6	Link to Summary: <b>2.5.6</b> Link to Details: <b>Version 2.5.6</b> Recommended Release



## Synapse RF200

Version	Comments
2.4.35	Link to Summary: <b>2.4.35</b> Link to Details: <b>Version 2.4.35</b>
2.4.36	Link to Summary: <b>2.4.36</b> Link to Details: <b>Version 2.4.36</b>
2.4.37	Link to Summary: <b>2.4.37</b> Link to Details: <b>Version 2.4.37</b>
2.5.0	Link to Summary: <b>2.5.0</b> Link to Details: <b>Version 2.5.0</b>
2.5.1	Link to Summary: <b>2.5.1</b> Link to Details: <b>Version 2.5.1</b>
2.5.2	Link to Summary: <b>2.5.2</b> Link to Details: <b>Version 2.5.2</b>
2.5.3	Link to Summary: <b>2.5.3</b> Link to Details: <b>Version 2.5.3</b>
2.5.6	Link to Summary: <b>2.5.6</b> Link to Details: <b>Version 2.5.6</b> Recommended Release

## Synapse SS200

Version	Comments
2.4.35	Link to Summary: <b>2.4.35</b> Link to Details: <b>Version 2.4.35</b>
2.4.36	Link to Summary: <b>2.4.36</b> Link to Details: <b>Version 2.4.36</b>
2.4.37	Link to Summary: <b>2.4.37</b> Link to Details: <b>Version 2.4.37</b>
2.5.0	Link to Summary: <b>2.5.0</b> Link to Details: <b>Version 2.5.0</b>
2.5.1	Link to Summary: <b>2.5.1</b> Link to Details: <b>Version 2.5.1</b>
2.5.2	Link to Summary: <b>2.5.2</b> Link to Details: <b>Version 2.5.2</b>
2.5.3	Link to Summary: <b>2.5.3</b> Link to Details: <b>Version 2.5.3</b>
2.5.6	Link to Summary: <b>2.5.6</b> Link to Details: <b>Version 2.5.6</b> Recommended Release

## Synapse SM200

Version	Comments
2.4.35	Link to Summary: <b>2.4.35</b> Link to Details: <b>Version 2.4.35</b>
2.4.36	Link to Summary: <b>2.4.36</b> Link to Details: <b>Version 2.4.36</b>
2.4.37	Link to Summary: <b>2.4.37</b> Link to Details: <b>Version 2.4.37</b>
2.5.0	Link to Summary: <b>2.5.0</b> Link to Details: <b>Version 2.5.0</b>
2.5.1	Link to Summary: <b>2.5.1</b> Link to Details: <b>Version 2.5.1</b>
2.5.2	Link to Summary: <b>2.5.2</b> Link to Details: <b>Version 2.5.2</b>
2.5.3	Link to Summary: <b>2.5.3</b> Link to Details: <b>Version 2.5.3</b>
2.5.6	Link to Summary: <b>2.5.6</b> Link to Details: <b>Version 2.5.6</b> Recommended Release

## Synapse RF200P81

Version	Comments
2.4.35	Link to Summary: <b>2.4.35</b> Link to Details: <b>Version 2.4.35</b>
2.4.36	Link to Summary: <b>2.4.36</b> Link to Details: <b>Version 2.4.36</b>
2.4.37	Link to Summary: <b>2.4.37</b> Link to Details: <b>Version 2.4.37</b>
2.5.0	Link to Summary: <b>2.5.0</b> Link to Details: <b>Version 2.5.0</b>
2.5.1	Link to Summary: <b>2.5.1</b> Link to Details: <b>Version 2.5.1</b>
2.5.2	Link to Summary: <b>2.5.2</b> Link to Details: <b>Version 2.5.2</b>
2.5.3	Link to Summary: <b>2.5.3</b> Link to Details: <b>Version 2.5.3</b>
2.5.6	Link to Summary: <b>2.5.6</b> Link to Details: <b>Version 2.5.6</b> Recommended Release

## Synapse RF266

Version	Comments
2.4.35	Link to Summary: <b>2.4.35</b> Link to Details: <b>Version 2.4.35</b>
2.4.36	Link to Summary: <b>2.4.36</b> Link to Details: <b>Version 2.4.36</b>
2.4.37	Link to Summary: <b>2.4.37</b> Link to Details: <b>Version 2.4.37</b>
2.5.0	Link to Summary: <b>2.5.0</b> Link to Details: <b>Version 2.5.0</b>
2.5.1	Link to Summary: <b>2.5.1</b> Link to Details: <b>Version 2.5.1</b>
2.5.2	Link to Summary: <b>2.5.2</b> Link to Details: <b>Version 2.5.2</b>
2.5.3	Link to Summary: <b>2.5.3</b> Link to Details: <b>Version 2.5.3</b>
2.5.6	Link to Summary: <b>2.5.6</b> Link to Details: <b>Version 2.5.6</b> Recommended Release

## ATmega128RFR2

Version	Comments
2.4.35	Link to Summary: <b>2.4.35</b> Link to Details: <b>Version 2.4.35</b>
2.4.36	Link to Summary: <b>2.4.36</b> Link to Details: <b>Version 2.4.36</b>
2.4.37	Link to Summary: <b>2.4.37</b> Link to Details: <b>Version 2.4.37</b>
2.5.0	Link to Summary: <b>2.5.0</b> Link to Details: <b>Version 2.5.0</b>
2.5.1	Link to Summary: <b>2.5.1</b> Link to Details: <b>Version 2.5.1</b>
2.5.2	Link to Summary: <b>2.5.2</b> Link to Details: <b>Version 2.5.2</b>
2.5.3	Link to Summary: <b>2.5.3</b> Link to Details: <b>Version 2.5.3</b>
2.5.6 – untested experimental build	Link to Summary: <b>2.5.6</b> Link to Details: <b>Version 2.5.6</b> Recommended Release

## ATmega1284RFR2

Version	Comments
2.4.35	Link to Summary: <b>2.4.35</b> Link to Details: <b>Version 2.4.35</b>
2.4.36	Link to Summary: <b>2.4.36</b> Link to Details: <b>Version 2.4.36</b>
2.4.37	Link to Summary: <b>2.4.37</b> Link to Details: <b>Version 2.4.37</b>
2.5.0	Link to Summary: <b>2.5.0</b> Link to Details: <b>Version 2.5.0</b>
2.5.1	Link to Summary: <b>2.5.1</b> Link to Details: <b>Version 2.5.1</b>
2.5.2	Link to Summary: <b>2.5.2</b> Link to Details: <b>Version 2.5.2</b>
2.5.3	Link to Summary: <b>2.5.3</b> Link to Details: <b>Version 2.5.3</b>
2.5.6 – untested experimental build	Link to Summary: <b>2.5.6</b> Link to Details: <b>Version 2.5.6</b> Recommended Release

## Synapse SM220

Version	Comments
2.4.35	Link to Summary: <b>2.4.35</b> Link to Details: <b>Version 2.4.35</b>
2.4.36	Link to Summary: <b>2.4.36</b> Link to Details: <b>Version 2.4.36</b>
2.4.37	Link to Summary: <b>2.4.37</b> Link to Details: <b>Version 2.4.37</b>
2.5.0	Link to Summary: <b>2.5.0</b> Link to Details: <b>Version 2.5.0</b>
2.5.1	Link to Summary: <b>2.5.1</b> Link to Details: <b>Version 2.5.1</b>
2.5.2	Link to Summary: <b>2.5.2</b> Link to Details: <b>Version 2.5.2</b>
2.5.3	Link to Summary: <b>2.5.3</b> Link to Details: <b>Version 2.5.3</b>
2.5.6	Link to Summary: <b>2.5.6</b> Link to Details: <b>Version 2.5.6</b> Recommended Release



## Synapse RF220

Version	Comments
2.4.35	Link to Summary: <b>2.4.35</b> Link to Details: <b>Version 2.4.35</b>
2.4.36	Link to Summary: <b>2.4.36</b> Link to Details: <b>Version 2.4.36</b>
2.4.37	Link to Summary: <b>2.4.37</b> Link to Details: <b>Version 2.4.37</b>
2.5.0	Link to Summary: <b>2.5.0</b> Link to Details: <b>Version 2.5.0</b>
2.5.1	Link to Summary: <b>2.5.1</b> Link to Details: <b>Version 2.5.1</b>
2.5.2	Link to Summary: <b>2.5.2</b> Link to Details: <b>Version 2.5.2</b>
2.5.3	Link to Summary: <b>2.5.3</b> Link to Details: <b>Version 2.5.3</b>
2.5.6	Link to Summary: <b>2.5.6</b> Link to Details: <b>Version 2.5.6</b> Recommended Release

**NOTE** – The RF220 is actually a SM220 mounted on a thru-hole carrier board, and so the RF220 software is *identical* to the SM220 build.

## Synapse RF100

Version	Comments
2.5.3, 2.5.4, 2.5.5	Did not pass QA - <b>DO NOT USE!</b>
2.5.6	Link to Summary: <b>2.5.6</b> Link to Details: <b>Version 2.5.6</b> Recommended Release

## Freescal MC1321x Chip

Version	Comments
2.5.3, 2.5.4, 2.5.5	Did not pass QA - <b>DO NOT USE!</b>
2.5.6	Link to Summary: <b>2.5.6</b> Link to Details: <b>Version 2.5.6</b> Recommended Release

## Composite Timeline - Summary

The following table provides a high level summary of the different builds since version 2.4.34, including what platform they were for, and the primary reason(s) for their creation.

Version	Platform(s)	Release Driver
<b>2.4.35</b> (unreleased Alpha)	ATmega128RFR2 ATmega1284RFR2 (smaller IC) RFR2-based SM220 <b>Prototypes</b>	An unreleased version that added initial support for two new ATMEL chips and the prototype hardware that later became the SM220 module. Sleep enhancements and overall reduction in current consumption were the remaining areas of focus.
<b>2.4.36</b> (unreleased Beta)	ATmega128RFR2 ATmega1284RFR2 (smaller IC) RFR2-based SM220 <b>Prototypes</b>	An unreleased version that continued the integration work for the two new ATMEL chips and the new prototype module that later became the SM220. Leveraging special hardware features <i>unique to the ATMEL chips</i> was the focus.
<b>2.4.37</b> Released to <b>Manufacturing only</b> for Pilot Builds	ATmega128RFR2 ATmega1284RFR2 (smaller IC) RFA1-based SM220	Stabilization and subsequent <b>Pilot Builds</b> of the new SM220 modules was the primary driver for this release. Changes were either to support the SM220 or were new SNAP features intended for the upcoming 2.5 series of firmware.
<b>2.5.0</b> Release candidate that did not receive any QA testing	All ATMEL platforms	The remaining SNAP 2.5 features interrupted by the need to cut a 2.4.37 release (for Manufacturing) were completed, and the first formal 2.5 Release Candidate was built but <b>never entered QA</b>
<b>2.5.1</b> Release candidate that did not pass QA testing	All ATMEL platforms	With SNAP 2.5.0 testing delayed by QA testing of other products, we decided to add in one more feature and start formal testing with version 2.5.1
<b>2.5.2</b> Release candidate that did not pass QA testing	All ATMEL platforms	One issue was found during sleep() testing, requiring a fix and rebuild (thus bumping the version).

(table continues next page...)

Version	Platform(s)	Release Driver
<b>2.5.3</b> Passed all QA tests, released.	All ATMEL platforms	txPwr() levels were adjusted for “worldwide” use. (The previous reductions were only for the FCC and IC regions.)
<b>2.5.3</b> Release candidate that did not pass QA testing	Freescale ‘9S08 platforms	Did not pass QA, not released
<b>2.5.4</b> Release candidate that did not pass QA testing	Freescale ‘9S08 platforms	An issue with SNAPpy script upload was corrected. Did not pass QA, not released
<b>2.5.5</b> Release candidate that did not pass QA testing	All ATMEL platforms and All Freescale ‘9S08 platforms	An issue with the Packet Serial protocol used in full-duplex scenarios was corrected. A potential “string buffer leak” when SNAPpy functions were invoked <i>with too many parameters</i> was also corrected. Did not pass QA, not released (see below).
<b>2.5.6</b> Passed all QA tests, released.	All ATMEL platforms and All Freescale ‘9S08 platforms	An issue with Packet Serial protocol <i>used in combination with AES-128</i> was found during the 2.5.5 QA testing and corrected.

## Composite Timeline - Detailed

---

This section distills the entire source code change log from version 2.4.34 through 2.5.6, and tries to place those changes in context. For a higher-level summary of the changes between 2.4.34 and 2.5.6, refer to the previous section of this document.

### A few words about platforms

In the following change logs, changes affecting specific platforms are prefaced by a (platform). Be aware that a change for a chip also affects any modules or boards that are based on that chip. So, the following shorthand is used:

ATmega128RFA1 also implies RF200, SS200, RF266, SM200, SM220 and RF220

Freescale 9S08 implies RF100 and MC1321x

### Version 2.4.35

Most of the development efforts between version 2.4.34 and 2.4.35 were driven by porting SNAP to new hardware platforms based on newer ATMEL chips. There were some fixes and enhancements made around the sleep() function.

#### Changes affecting all platforms

None – all changes were specific to the ATMEL-based modules and chips.

#### Changes affecting specific platforms (ATMEL)

**NOTE** – the original ATMEL versions of SNAP were written for the ATMEL ATmega128RFA1.

**New Chip Platform** – A new build of SNAP firmware was created for the ATMEL ATmega128RFR2. This chip has an enhanced radio relative to its predecessor the 'RFA1, but turned out to not be 100% compatible (thus requiring its own SNAP firmware – you couldn't just load ATmega128RFA1 code into it). As a quick example, the hardware random number generator was no longer usable due to other radio enhancements.

**New Chip Platform** – A new build of SNAP firmware was created for the ATMEL ATmega1284RFR2. This chip also has the enhanced radio, but comes in a much smaller package (at the cost of fewer I/O pins brought out).

**New Module Platform** – A new build of SNAP firmware was also created for a prototype module based on the 'RFR2. This module would eventually become the SM220, but by the time it was released it had been changed over to the 'RFA1 chip.

Overall current consumption was reduced slightly by removing the initialization of some unused hardware. The “go to sleep” and “wake back up” code paths were also optimized for speed.

It was discovered that the sleep() function could wake up early due to the internal MAC Symbol Counter rolling over, as well as the internal 1 millisecond clock interrupt occurring. Both of these issues were corrected, so that the unit would remain asleep for the requested duration.

As part of the above sleep() work, the sleep software was recalibrated for higher accuracy (the test case for this work was a 12 hour sleep duration).

The `getLq()` built-in was changed to return a “snapshotted” value taken at the time the radio packet was received, instead of returning a “live” reading. This brings the ATMEL platforms in line with the rest of the SNAP platforms.

## Version 2.4.36

Version 2.4.35 put SNAP on two new chips (the ATmega128RFR2 and the ATmega1284RFR2) but it did not fully leverage the new features of these chips, nor did it address all the incompatibilities between them and their ATmega128RFA1 predecessor. This was the bulk of the work in version 2.4.36.

The SM220 prototypes continued to evolve, and required software support as well.

The “moveable I<sup>2</sup>C” feature was also back-ported into ATMEL SNAP from the STM32W108xB version of SNAP where this feature was first introduced.

### Changes affecting all platforms

None – all changes were specific to the ATMEL-based modules and chips.

### Changes affecting specific platforms (ATMEL)

Version 2.4.36 replaced the original software-based implementation of AES-128 with one that utilized the internal “crypto engine” of the ATmega128xxx processors. This enhancement applied to all of the ATMEL chips.

It was discovered that the internal FLASH of the ‘RFR2 chips was not 100% compatible with the FLASH of the original ‘RFA1 chip. This required the FLASH “write” routines to be re-written, resulting in the ‘RFR2 chips gaining their own unique Boot Loader.

The SM220 was the first SNAP Module to boast two onboard antennas – a “meandering F” and a “U.FL” connector. Support for *software controlled antenna selection* was added in this version (refer to NV Parameter 64).

The `txPwr()` levels were reduced from “maximum supported by the chip” to levels estimated to be FCC (and IC) compliant (in preparation for sending modules to the FCC test lab).

The ‘RFR2 chips boasted an internal feature ATMEL dubbed “SRT – Smart Radio Technology.” This was supposed to enable a 5 milliamp “radio receive” mode. A lot of effort went into this, but we were unable to get the chips to reliably enter and stay in this mode. (You will likely see some current savings on a ‘RFR2 but not as much as we had hoped).

The “moveable I<sup>2</sup>C” feature was back-ported from the STM32W108xB version of SNAP. Now if you need to connect an I<sup>2</sup>C peripheral to a different pair of pins, just specify the alternate SCL and SDA pins in the `i2cInit()` function call.

**NOTE** – these two new parameters are *optional*, you do not have to change your existing scripts unless you want to leverage this new capability. Calling the `i2cInit()` function *without* the new optional parameters causes the original pin assignments to be used.

## Version 2.4.37

Stabilization and subsequent Pilot Builds of the new SM220 modules was the primary driver for this release. In parallel, addition of new features intended to be part of the SNAP 2.5 series of firmware had already begun: PACKET\_CRC, I2C\_RESTART and STDIN “line mode” enhancements (described below).

### Changes affecting all platforms

**New Feature:** PACKET\_CRC – similar to the previous RPC\_CRC feature introduced in SNAP 2.4.19, the PACKET\_CRC feature added an additional software CRC to the radio packets. This was added to address issues with “packet storms” seen out in the field. Like RPC\_CRC, enabling PACKET\_CRC costs you two bytes of packet space (the additional CRC takes up two bytes). Here is how PACKET\_CRC differs from RPC\_CRC:

Aspect	PACKET_CRC	RPC_CRC
Enabled by Feature Bit... (look at NV #11)	0x0400	0x0100
Applies to...	All packet types, including RPC packets	RPC packets only (both unicast and multicast)
Calculated from...	The entire packet, including the header	The packet payload only
Applied to packets sent or received...	Over the radio only	Radio and Serial (both)

**NOTE** – you can enable **both** CRCs if you wish, but this will cost you 4 bytes of packet space total.

**New Feature:** I<sup>2</sup>C\_RESTART – Prior to version 2.4.37, SNAP could only work with devices that used the “I<sup>2</sup>C\_START, I<sup>2</sup>C\_STOP, I<sup>2</sup>C\_START, I<sup>2</sup>C\_STOP” hardware handshake sequence for back-to-back commands (for example, an i2cWrite() to specify the data to read, followed by an i2cRead() to capture that data). Some I<sup>2</sup>C devices instead use a “I<sup>2</sup>C\_START, I<sup>2</sup>C\_RESTART, I<sup>2</sup>C\_STOP” hardware handshake sequence. SNAP version 2.4.37 introduces an optional trailing parameter to the i2cWrite() function.

When the optional parameter is provided and its value is True, SNAP will end the i2cWrite() command such that the beginning of an I<sup>2</sup>C\_RESTART is created. The following i2cRead() will complete the I<sup>2</sup>C\_RESTART (instead of generating an I<sup>2</sup>C\_START).

When the optional parameter is omitted, or is provided but its value is False, then the normal I<sup>2</sup>C\_STOP sequence is generated.

This enhancement allows SNAP to work with a wider range of I<sup>2</sup>C device.

**New Feature** – SNAP has always had a “line mode” for HOOK\_STDIN, but if you received too many characters before the receipt of a Carriage Return or Line Feed character, the system would print an error message and discard the data. Now *even if you have specified “line mode”* which technically means “don’t send the data until you get a CR or LF” the system will push what it has received so far *if the buffer fills up*. This makes the feature more useful.

To support this new behavior, an new getStat() option has been added, getStat(18). By calling this function, your SNAPpy script can check and see why the HOOK\_STDIN handler has been called.

For more details, refer to the **SNAP Reference Manual**.

**Bug Fix:** Comparison of SNAPpy integers (signed 16-bit) was improved. Prior to this version, a comparison like “20000 > -12768” would return False instead of True due to 16-bit wrap-around. This

has been corrected. Note that this might require changes to your existing SNAPpy scripts if you were relying on the previous (incorrect) behavior.

**Bug Fix:** The Manufacturing Date was not being preserved through a Factory Default (fixed)

**Bug Fix:** In an exhaustive review of the SNAPpy Virtual Machine, numerous “dynamic string leaks” were identified and corrected.

## Changes affecting specific platforms (ATMEL)

(All ATMEL) – The hardware inside the ATMEL radios includes “trim capacitors” that can be selectively enabled. You can now specify an alternate radio trim setting via an NV Parameter (#63 – NV\_ALT\_RADIO\_TRIM\_ID) to take advantage of this hardware capability. Most customers will never need to use this, but if for some reason your units are running higher in frequency then you can change this NV Parameter from its default value of 0 to enable 1-15 steps of additional capacitance (which will lower the radio frequency).

**New Feature:** Sleep mode 2 (sleep(2, ticks)) added. This new sleep mode uses the “MAC Symbol Counter” inside the radio as a timebase, and provides finer-grained sleep durations.

**NOTE** – the hardware does not have the ability to adjust the frequency in the other direction. (You cannot use the internal trim to raise the frequency.)

(SM220) – After the FCC testing was completed, the txPwr() levels were adjusted to the levels specified by the test results. This included a reduction in power on channels 0-14, and channel 15 had to be disabled for broadcast completely.

(RF266) – The default Feature Bits for the RF266 were changed from “enable both UARTs” (this is the default used by all other ATMEL-based platforms) to “enable UART1 only”. This was done because the first UART (UART0) is not brought out to any of the RF266 pins.

## Version 2.5.0

Several of the features originally intended for SNAP 2.5 were first seen in the *unreleased* SNAP versions 2.4.35 through 2.4.37. The remaining planned features were added (or in some cases, *enabled* since they were present in the previous versions but disabled before shipping due to lack of formal testing) and a formal 2.5.0 Release Candidate was created and submitted for full QA testing. A few miscellaneous improvements were made along the way.

## Changes affecting all platforms

**New Feature:** CPU\_IDLE – Behind the scenes, SNAP was constantly checking the radio and serial ports looking for incoming data to be processed. Starting in this version, if SNAP has checked all of the possible sources of incoming data and *found nothing to be processed*, it will use the CPU’s built-in “idle” capability to wait for the next interrupt.

This reduces the power consumption of SNAP nodes that are not processing a lot of traffic, which can increase battery life. Note that if your SNAP node is being kept busy (for example, your application sends a lot of radio and/or serial traffic) then you will not see much benefit from this enhancement.

**New Feature:** type() built-in added to SNAPpy. The ability to tell (at run-time) if a variable was (for example) a String versus an Integer was added to the SNAPpy Virtual Machine. As a quick example of where this can come in handy, the loadNvParam() function can reload a previously saved value, but until now there was no easy way to verify it’s TYPE.



**NOTE** – You could do “is None” and “is not None” checks before (and you still can) but the type() function is much more versatile.

**New Feature:** SNAP now allows you to use different multicast “packet forwarding” settings on the serial port versus the radio.

In previous versions of SNAP, NV Parameter 6 controlled which multicast groups were forwarded on both the radio and the serial port. You can still choose to do that, but now there is an additional NV Parameter 78 that *when set* gives the serial port its own settings, and means that the multicast group bitmask in NV Parameter 6 apply only to packets forwarded over the radio. This allows you to do things like “only forward group 0x0002” packets over the radio, and only forward group 0x0004 over the serial port”. To have NV Parameter 6 control both serial and radio forwarding, let NV Parameter 78 to None.

**Enhancement:** Now if you call initUart() on a UART that is being used by the Packet Serial feature, the Packet Serial state machine gets re-initialized too. (There were users who were changing their serial port baud rates on the fly and getting poor results.)

**Enhancement:** Robustness of the NV Parameters storage area when performing “page swaps” in the presence of power outages or system resets was improved.

## Changes affecting specific platforms (ATMEL)

This was the first version to include a “SNIFFER Firmware” build for the ATmega1284RFR2.

### Version 2.5.1

Testing of other Synapse products had delayed the start of the SNAP 2.5.0 QA testing, and the recent addition of the type() function (see version 2.5.0) had called our attention to some differences in SNAPpy’s handling of “is” and “is not” clauses compared to “desktop” Python.

Since testing of 2.5.0 had not even started, we decided to go ahead and address these issues and submit a 2.5.1 candidate to QA.

## Changes affecting all platforms

**Enhancement:** The behavior of the “is” clause in SNAPpy was changed to more closely match what full “desktop” Python does.

## Changes affecting specific platforms

None

### Version 2.5.2

In 2.5.1 QA testing an issue was noticed with sleep() mode. This was corrected in 2.5.2.

We also added another build of firmware to the set.

## Changes affecting all platforms

None – all changes were specific to the ATMEL-based modules and chips.

## Changes affecting specific platforms

It was noticed that the chip was sometimes pausing for about 10 milliseconds at a 5 milliamp current draw before fully entering sleep mode. Since many SNAP applications are battery powered, this short period of higher power consumption was removed.

A version of the “DMX” variant of SNAP for the SM220 module was created and added to the set of firmware images.

### Version 2.5.3

In version 2.4.37 we had set the txPwr() levels for the SM220 correctly for the FCC and IC regions, but had not reduced the txPwr() settings for “worldwide” usage.

*The SM220 has more transmit power than its predecessor the SM200 and so cannot re-use the same internal txPwr() settings.*

## Changes affecting all platforms

None – all changes were specific to the ATMEL-based modules and chips.

## Changes affecting specific platforms

SM220 – reduced txPwr() levels for “worldwide” from 4 to 2 (this platform only).

**NOTE** – a version SNAP 2.5.3 for Freescale 9S08 platforms entered QA but was not released (see SNAP 2.5.4)

### Version 2.5.4

An issue was corrected regarding SNAPpy script upload on the Freescale ‘9S08 platforms only.

## Changes affecting all platforms

None – all changes were specific to the Freescale ‘9S08-based modules and chips.

## Changes affecting specific platforms (Freescale 9S08)

RF100 and MC1321x – Corrected issue with SNAPpy script upload.

**NOTE** – this version was not released due to a Packet Serial bug being found in all versions of SNAP. See versions 2.5.5 and 2.5.6 instead.

### Version 2.5.5

While 2.5.4 was in QA testing a problem was found with the Packet Serial protocol. After that issue was fixed, a small SNAPpy Virtual Machine improvement was made.

## Changes affecting all platforms

Testing in the field revealed a software race condition if packets were being sent and received over the Packet Serial interface with a particular (overlapping) timing. The bulk of the 2.5.5 work was correcting this issue.

It was also found that calling SNAPpy functions with too many parameters could in certain situations lead to string buffer leaks. This was also corrected.

## Changes affecting specific platforms

None – all changes in the version applied to all platforms.

### Version 2.5.6

While 2.5.5 was in QA testing a problem was found with the Packet Serial protocol *when used in combination with AES-128*.

## Changes affecting all platforms

QA testing of SNAP 2.5.5 revealed that AES-128 over Packet Serial had been accidentally broken (AES-128 over the radio was fine, and non-AES-128 traffic over the serial interface was fine). This was quickly corrected, and SNAP 2.5.5 testing morphed into SNAP 2.5.6 testing.

## Changes affecting specific platforms

None – all changes in the version applied to all platforms.

**NOTE** – This is the first release in the 2.5 series to support Freescale 9S08 platforms.

## Installation and Upgrade Notes

---

You must first upgrade to the latest version of Portal. For details on how to do this, refer to the Portal Reference Manual or the Portal Release Notes.

When you install the latest version of Portal, you will automatically have the latest versions of the SNAP firmware available on your PC. You can then use Portal to load the new firmware into each of your existing SNAP nodes.

**NOTE** – there are now quite a few SNAP platforms. Be sure to load the correct firmware for your hardware.

**NOTE** – Portal now only includes firmware images for Synapse modules (e.g., RF200, SM220, etc). Contact Synapse Customer Support to request SNAP images for “chips” (e.g., ATmega128RFA1, etc.).

In the following table, modules are listed first, followed by chips.

Platform	Configuration	Firmware
RF100 24-pin through-hole mounted SNAP Engine	Standard	RF100_Snap V2.5.6.sfi
	AES128	RF100_AES128_SnapV2.5.6.sfi
	Debug	RF100_debug_SnapV2.5.6.sfi
	Debug AES128	RF100_AES128_debug_SnapV2.5.6.sfi
Freescal MC1321x chip	Standard	MC1321x_Snap V2.5.6.sfi
	AES128	MC1321x_AES128_SnapV2.5.6.sfi
	Debug	MC1321x_debug_SnapV2.5.6.sfi
	Debug AES128	MC1321x_AES128_debug_SnapV2.5.6.sfi
RF200 24-pin through-hole mounted SNAP Engine	Standard	RF200_Snap V2.5.6.sfi
	AES128	RF200_AES128_SnapV2.5.6.sfi
	Debug	RF200_debug_SnapV2.5.6.sfi
	Debug AES128	RF200_AES128_debug_SnapV2.5.6.sfi
SS200 USB bridge device, looks similar to a thumb drive	Standard	SS200_Snap V2.5.6.sfi
	AES128	SS200_AES128_SnapV2.5.6.sfi
	Debug	SS200_debug_SnapV2.5.6.sfi
	Debug AES128	SS200_AES128_debug_SnapV2.5.6.sfi
SM200 64-pad surface-mount module	Standard	SM200_SnapV2.5.6.sfi
	AES128	SM200_AES128_SnapV2.5.6.sfi
	Debug	SM200_debug_SnapV2.5.6.sfi
	Debug AES128	SM200_AES128_debug_SnapV2.5.6.sfi
RF200P81 24-pin module using an SM200 on a carrier board	Standard	RF200P81_SnapV2.5.6.sfi
	AES128	RF200P81_AES128_SnapV2.5.6.sfi
	Debug	RF200P81_debug_SnapV2.5.6.sfi
	Debug AES128	RF200P81_AES128_debug_SnapV2.5.6.sfi
SM220 64-pad surface-mount module with improved transmit power	Standard	SM220_SnapV2.5.6.sfi
	AES128	SM220_AES128_SnapV2.5.6.sfi
	Debug	SM220_debug_SnapV2.5.6.sfi
	Debug AES128	SM220_AES128_debug_SnapV2.5.6.sfi

(table continues next page...)

Platform	Configuration	Firmware
RF220 24-pin module using an SM220 on a carrier board	Standard	RF220P81_SnapV2.5.6.sfi
	AES128	RF220P81_AES128_SnapV2.5.6.sfi
	Debug	RF220P81_debug_SnapV2.5.6.sfi
	Debug AES128	RF220P81_AES128_debug_SnapV2.5.6.sfi
ATMEL ATmega128RFA1 chip build	Standard	ATmega128RFA1_SnapV2.5.6.sfi
	AES128	ATmega128RFA1_AES128_SnapV2.5.6.sfi
	Debug	ATmega128RFA1_debug_SnapV2.5.6.sfi
	Debug AES128	ATmega128RFA1_AES128_debug_SnapV2.5.6.sfi
ATMEL ATmega128RFR2 chip build	Standard	ATmega128RFR2_SnapV2.5.6.sfi
	AES128	ATmega128RFR2_AES128_SnapV2.5.6.sfi
	Debug	ATmega128RFR2_debug_SnapV2.5.6.sfi
	Debug AES128	ATmega128RFR2_AES128_debug_SnapV2.5.6.sfi
ATMEL ATmega1284RFR2 chip build	Standard	ATmega1284RFR2_SnapV2.5.6.sfi
	AES128	ATmega1284RFR2_AES128_SnapV2.5.6.sfi
	Debug	ATmega1284RFR2_debug_SnapV2.5.6.sfi
	Debug AES128	ATmega1284RFR2_AES128_debug_SnapV2.5.6.sfi

## Known Problems and Workarounds

---

### Not all demo scripts work on all platforms

**Problem:** Many of the existing “demo” and “helper” scripts predate the newer SNAP ports, and will not work on those platforms without modification.

**Work-around:** Be sure to refer to the appropriate section of the **SNAP Reference Manual** for your particular hardware platform, and watch for differences relative to the original Synapse RF100 Nodes. Also, read the **SNAP Users Guide** for an understanding of the platform variable and the lists of constants that Portal can import based on that variable.

### Not all SNAP engines work on all demonstration boards

**Problem:** The RF200, RF300, and SNAP Engines based on the ZIC2410 are not compatible with the original Synapse SN111 End Device Demonstration Board. There is a supply voltage conflict between the relay drivers on that board and the initial pin states of those engines, preventing the engines from fully “booting up.”

**Work-around:** Look on the support forum at <http://forums.synapse-wireless.com/> for application note “Updating SN111 Boards for Use with Various SNAP Engine”. It provides details on how to modify your SN111 End Device Demonstration board to coexist with these SNAP Engines.

**NOTE** – “next-gen” SN111 boards do not have an onboard relay, and so this issue does not apply to them. Any SNAP Engine can be run in (can be powered by) these newer SN111 boards, although there can still be hardware modifications required for certain SNAP Engines (such as the Si100x-based ones) to access all SN111 features (for example, the seven-segment display).

## Related Documents

---

- SNAP Reference Manual
- SNAP User Guide
- SNAP Primer
- Portal Reference Manual
- Portal Release Notes
- SNAP Connect Python Package Manual
- SNAP Connect Release Notes
- SNAP Connect E10 Gateway User Manual
- SNAP Connect E15 Gateway User Manual
- SNAP Connect E20 Gateway User Manual