**IEEE P802.15**

**Wireless Personal Area Networks**

|  |  |
| --- | --- |
| Project | IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs) |
| Title | HRP UWB SRDEV PPDU Text Contribution |
| Date Submitted | [5 July, 2018] |
| Source | Frank Leong (NXP Semiconductors)Wolfgang Küchler (NXP Semiconductors)Thomas Baier (NXP Semiconductors)Brima Ibrahim (NXP Semiconductors)Jochen Hammerschmidt (Apple)Ayman Naguib (Apple)Thomas Reisinger (Continental)Daniel Knobloch (BMW) |  |
| Re: | [If this is a proposed revision, cite the original document.][If this is a response to a Call for Contributions, cite the name and date of the Call for Contributions to which this document responds, as well as the relevant item number in the Call for Contributions.][Note: Contributions that are not responsive to this section of the template, and contributions which do not address the topic under which they are submitted, may be refused or consigned to the “General Contributions” area.] |
| Abstract | [HRP UWB SRDEV PPDU text contribution to 802.15.4z] |
| Purpose | [Describe in detail elements of HRP UWB PHY and MAC for 802.15.4z] |
| Notice | This document has been prepared to assist the IEEE P802.15. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein. |
| Release | The contributor acknowledges and accepts that this contribution becomes the property of IEEE and may be made publicly available by P802.15. |

#  Introduction

Present HRP ranging-capable devices (RDEVs) use the content of the SHR to estimate time of flight. Due to the periodicity of the SHR, it may be vulnerable to attack.

This document describes interoperable secure ranging devices (SRDEVs) featuring enhanced immunity to attack. The main enhancement is the inclusion of a Scrambled Timestamp Sequence (STS) in the basic HRP PPDU format.

# Secure Ranging PPDU Formats

An SRDEV shall support the basic HRP frame format outlined in [1] §16.2.

In addition, SRDEVs shall support the transmission and reception of frames, optimized for Secure Ranging, consisting of:

* “Synchronization Header” with SFD (Start-of-Frame Delimiter)
* “Scrambled Timestamp Sequence”, changing for every frame (according to a Key and a Seed, to be defined on higher layer)
* Fixed DATA (PHR+PSDU) part

Three Secure Ranging PPDU formats shall be supported, the difference between the formats being the location of the STS field and the existence of a PHR and PHY Payload field:



Figure 1: Secure Ranging Frame Format A (mandatory, avoiding STS timing dependency on payload length)



Figure 2: Secure Ranging Frame Format B (optional, backward compatible in case of 62.4 MHz Mean PRF and legacy PHR data rate)



Figure 3: Secure Ranging Frame Format C (Optional)

Secure ranging frame format C is intended for certain use cases where the participants in the secure ranging exchange are known to each other such that information about source and/or destination are implicit in the knowledge of what STS is used for transmission and reception between the connected devices, respectively.

# Mean PRF

A nominal mean PRF of at least 62.4 MHz shall be used for all packets during a ranging session. Nominal mean PRFs lower than 62.4 MHz are not required for SRDEVs.

It is mandatory for SRDEVs to support nominal mean PRF 62.4 MHz (referred to as “PRF64” in this document) as well as nominal mean PRF 124.8 MHz (referred to as “PRF128” in this document).

# RF Requirements

## Operating Frequency for Secure Ranging

It is mandatory for SRDEVs to support CH5 and CH9 (i.e., 6489.6 MHz and 7987.2 MHz, both in high band).

Channel selection may be based upon regional regulatory conditions.

Support of all other channels in [1] is optional.

# PRF64 PHY

## Synchronization Header (SHR)

### SYNC

The SYNC duration in number of periodic symbols () shall be configurable and cover the following values: 32, 64, 128, 256, 1024, 2048, and 4096. shall be static or known a priori via signaling at higher protocol layers. Support for of 64 is mandatory for SRDEVs.

The SYNC sequences for PRF64 are as specified in [1] §16.2.5.1.

### SFD

To improve the performance of devices with coherent demodulators, the binary sequences listed in Table 1 can optionally be used as SFD. The selection of the SFD pattern is either static or signaled via higher layers.

|  |  |  |
| --- | --- | --- |
| ID | Pattern | Nsym |
| 0 | --+- | 4 |
| 1 | ----++-+ | 8 |
| 2 | ---+-+--+--+++-+ | 16 |
| 3 | -----+---+++--+--+-+-+--+--++--- | 32 |
| 4 | -----+---+--+----+-++---+-+-+-------++---++-+-++--+-----++++--+- | 64 |

Table 1: Binary codes for SFD (Optional)

## PHR and PSDU

### PHR

It is mandatory for SRDEVs to support transmission and reception of PRF64 PHRs 850 kbit/s data rate.

To allow for reduced PSDU durations, given with very short Payloads, the PHR for ranging messages can optionally have the PHR field coded at the same data rate as the payload, i.e., 6.8 and 27 Mbit/s PHRs are allowed for payloads also coded at the same data rate. Note that 850 kbit/s for both payload and PHR is already available per [1].

The signaling of the use of same rate PHRs shall be performed at the MAC layer, i.e., no auto detection of the PHR rate is required.

### PSDU

It is mandatory for an SRDEV to support PRF64 at 6.8 Mbit/s data rate, while 110 kbit/s, 850 kbit/s, and 27.2 Mbit/s are optional.

### Modulation and Coding

Modulation and coding of PHR and PSDU for PRF64 are as specified in [1] §16.2-§16.3.

## Scrambled Timestamp Sequence

### Deterministic Random Bit Generator

The NIST, i.e., [2] §10.2.1, based Deterministic Random Bit Generator (DRBG) shall be used to generate the Scrambled Timestamp Sequence (STS). The length of the timestamp sequence shall be configurable.



Figure 4: DRBG for STS generation

The ith output bit of the DRBG is denoted by Ci. Internally, the DRBG uses a block size of 128 bits. The jth generated block is denoted by Bj, where j = floor(i / 128). The mapping from blocks to DRBG output bits follows Bj = {Cj\*128, Cj\*128 + 1, … Cj\*128 + 127}.

An example of the first 256 generated bits of the DRBG can be found in Section 9.1.

### Sequence Length and Gaps

The STS Building Block of the STS field in the frame is shown below:



Figure 5: STS Building Block

A segment of active (non-zero) pulsing at the nominal mean PRF is encapsulated by guard intervals, or gaps, of length Lgap in counts of chips. Lgap shall be 512, a guard duration of, roughly, 1 µs. All chips during each gap shall carry no signal energies at each respective transmitter.

The STS active segment length is specified in number of chips as follows:

All devices shall support , while other values of are optional.

Several STS Active intervals and Gaps can be combined to form the overall STS. The following diagram shows an example of two concatenated Building Blocks:



Figure 6: STS consisting of two STS Building Blocks

A single Gap of length 512 chips shall be inserted between successive STS Active intervals A and B, and Gaps shall also be inserted before and after the first and last STS Active segment. During the gap between successive STS Active intervals, the STS code shall be paused, and be resumed again after the gap.

Any number of STS Building Blocks can be concatenated, always with a single Gap of length 512 chips between consecutive STS Active intervals, up to 8 STS Building Blocks. The number of STS Building Blocks to be concatenated shall be static or made available a priori via upper layer signaling. All devices shall support use of 1, 2, 3, and 4 STS Building Blocks, while support of 5, 6, 7, and 8 is optional.

The segmented approach allows consistency checks to improve the security and/or to exploit a stronger overall processing gain for the correlation.

### STS, PRF64, Hopped Mode (Mandatory)

In addition to the main purpose of the STS, i.e., ranging with coherernt receivers featuring enhanced immunity to attack, this mode allows use of non-coherent receivers, specifically to identify the STS according to the transmitted hopping bits.

Two consecutive values of the DRBG sequence Ai,i+1 are mapped directly to the STS code Ci,i+1 using Table 2, i.e., bit Ai sets hopping position and bit Ai+1 sets binary value.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

(i = 0,2,4,…)

Table 2: DRBG mapping for PRF64 STS, Hopped Mode

The resulting sequence Ci is then spread using δL = 4.

### STS, PRF64, Data Symbol Structure Based Mode (Optional)

In this mode, the conventional payload mapper is used to map DRBG bits to UWB pulses. Specifically, following the Data Symbol Structure nomenclature of Table 16.3 in HRP UWB PHY of [1], the number of hop bursts is chosen to be 2 (), and the number of chips per burst is 1 (), for a total number of chips per symbol of . DRBG bits are directly passed to the symbol mapper with no encoding, where the DRBG and the PN sequence together determine the pulse position and pulse polarity (see BPM as described in [1], §16.2.2 and §16.3). Figure 7 depicts the symbol mapping mechanism for the STS sequence. Note that number-of-chip quantities and are related to the symbol duration , BPM half-symbol duration , and chip duration by and .



Figure 7: Data Symbol Structure Based STS signal generation

For the -th STS pulse, … every 2 binary bits from the DRBG engine, namely and along with the LFSR scrambling bit are used to define the ternary amplitudes of the chips within the -th STS symbol as

wherein location of the one and only non-zero pulse per per symbol is defined as

The result combination of possible input/output values is shown in Table 3. Each possible scenario has exactly one non-zero pulse. Note that no additional spreading takes place after this mapping.

|  |  |
| --- | --- |
|  |  |
|  | +0000000 |
|  | -0000000 |
|  | 0000+000 |
|  | 0000-000 |
|  | 0-000000 |
|  |  0+000000 |
|  | 00000-00 |
|  | 00000+00 |

Table 3: DRBG mapping for PRF64 STS, Data Symbol Structure Based Mode

### Message-Encoding STS (Optional)

To further improve the level of security of the STS for first-path extraction and ranging purposes, the following optional Message-Encoding STS mechanism is provisioned (see Figure 8).

An additional processing step is introduced after the STS pulse mapping. In this step, consecutive blocks of chips will be polarity flipped as a function of consecutive bits from a message E = e0, e1, … The *i*’th *B*[*i*] block of Ne chips from the pulse mapper is associated with bit ei from the message. If bit ei has value 0, block *B*[*i*] remains unchanged. If bit ei has value 1, all chips in block *B*[*i*] are polarity reversed (pulse polarity +1 becomes -1, pulse polarity -1 becomes +1, pulse polarity 0 remains unchanged). Note that chips in the Gaps are skipped in this process, so that no message bit ei is associated with any Gaps; the Message-Encoding procedure only applies during the “STS Active” intervals.



Figure 8: Message Encoding STS

Ne, defined in units of chip counts, can be any value defined by:

 Ne = 512 \* 2*l*,

where can be any number *l* = 0,1,2,3,4,5. Both Ne and *l* are pre-negotiated between communicating devices at higher communication layers during connection setup.

The message E can optionally be used for advanced authentication schemes. The contents of message E are outside of the scope of this specification.

# PRF128 PHY

## Synchronization Header (SHR)

### SYNC

The SYNC duration in number of periodic symbols () shall be configurable and cover the following values: 32, 64, 128, 256, 1024, 2048, and 4096. shall be static or known a priori via signaling at higher protocol layers. Support for of 64 is mandatory for SRDEVs.

For all PRF128 frame formats, a length-127 binary code shall be used to construct the SYNC symbols.

One of the binary sequences listed in Table 4 shall be used. The selection of the code is either static or signaled via higher layers.

All mappings of PRF128 SYNC codes to channel numbers are allowed, although some considerations may apply that could lead an implementer to avoid specific code-channel combinations, for example:

* Within channels, use of sets of codes with good cross-correlation may reduce the probability of receivers synchronizing to unwanted transmissions, thereby improving multi-user performance.
* Aliasing effects may cause receivers to synchronize to transmissions on adjacent channels, which may be avoided by ensuring sufficiently low cross-correlation between codes selected for use in adjacent channels (center frequencies).

|  |
| --- |
| Length-127 |
| ID | Code |
| 0 | -+++++++-+-+-+--++--+++-+++-+--+-++---++-++++-++-+-++-++--+--+---+++----+-+++++--+-+-+++--++-+---+--++++---+-+----++-----+----- |
| 1 | -+--+--++-+--++++-+++----+++++++---+++-++---+-+--+-+++++-+-+-+----+-++-++++--+++--+-+-++--++-----++-++-+-+++-+---++--+---+----- |
| 2 | -++--++-++---+++--+++-+-+++----+--++-----+-+-+-++-+--+--+-+--++++--+---++-+-+----+++++++-+++-++-++++-+---+-++--+-+++++---+----- |
| 3 | -++-+++++-----+-++----+----+++-+---++---+--+++--+-+--++-+--+-++++-+-+++-+++---++++--++--+--+---+-+-+-++-++--+++++++-++-+-+----- |
| 4 | -++----+--+--+++++++-+++++---+++-+-+-+--+-+-++++-+--++--+++--++-+-++---+---+-+++-++--+----++++--+-++-+++-----+-+---++-++-+----- |
| 5 | -++++-++--++---+--+--+++--+++++--+-----+---++-+-+-+--++-++-+--+-+----+-++----++--+-+++++++-+-++-+++-++++---+++-+---+-+-+++----- |
| 6 | -+---+--++---+-+++-+-++-++-----++--++-+-+--+++--++++-++-+----+-+-+-+++++-+--+-+---++-+++---+++++++----+++-++++--+-++--+--+----- |
| 7 | -+-++-----+++-+----+--+++---++-+--+--+-+++-++-+++--++-++--+-+-++-+-+++++-++++----++---+---+-+--++--++++-+-+-+---+++++++--+----- |
| 8 | -+++-++-+++++--+++++++-+--+--+-+-+++-+-+-+--+++---++--+-++--++---+-++++-+++--+---+----++-+-++-+---++++-----+--++-++----+-+----- |
| 9 | -+--+++++++---+-+-+-++++--++--+-+---+---++----++++-+++++-+-++-+-+--++-++--+++-++-+++-+--+--+-++---+++--+----+-+++-----++-+----- |
| 10 | -+-----++----+-+---++++--+---+-++--+++-+-+--+++++-+----+++---+--+--++-++-+-++-++++-++---++-+--+-+++-+++--++--+-+-+-+++++++----- |
| 11 | -+++-+-+---+-+++---++++-+++-++-+-+++++++-+--++----++-+----+-+--+-++-++--+-+-+-++---+-----+--+++++--+++--+--+---++--++-++++----- |
| 12 | -+-+----++-++--+-----++++---+-++-+-++----+---+--+++-++++-+---++--++-+--++---+++--+-+-+-+++-+-+--+--+-+++++++--+++++-++-+++----- |
| 13 | -+-+-++-+++++++--++-++-+-+-+---+--+--++--++++---+++-+++-+-++++-+--+-++--+-+--+++--+---++---+-+++----+----++-+-----+++++-++----- |
| 14 | -++--+-+---+++----+++++---+-++-++--++---++-+--++-+++++++-+++--++++-+-----+-+-++++--+--+---+----+--+++-++-+-+-+--+-+++-+-++----- |
| 15 | -+---+++++-+--++-+---+-++++-++-+++-+++++++----+-+-++---+--++++--+-+--+--+-++-+-+-+-----++--+----+++-+-+++--+++---++-++--++----- |
| 16 | -++-+-+++-+--+-+-+-++-+++--+----+---+--+--++++-+-+-----+-++++--+++-+++++++-++--+-++---++--++-++-+---+++++----+++---+-+--++----- |
| 17 | -+-++-++---+-+-----+++-++-+--++++----+--++-+++-+---+---++-+-++--+++--++--+-++++-+-+--+-+-+-+++---+++++-+++++++--+--+----++----- |

Table 4: PRF128 SYNC codes

### SFD

To maximize the performance of devices with coherent demodulators, the SFD pattern shall be selected from the binary patterns listed in Table 5, support for all listed patterns being mandatory for SRDEVs operating in PRF128. The selection of the SFD pattern is either static or signaled via higher layers. SFD ID #1 (length 8) is the default.

|  |  |  |
| --- | --- | --- |
| ID | Pattern | Nsym |
| 0 | --+- | 4 |
| 1 | ----++-+ | 8 |
| 2 | ---+-+--+--+++-+ | 16 |
| 3 | -----+---+++--+--+-+-+--+--++--- | 32 |
| 4 | -----+---+--+----+-++---+-+-+-------++---++-+-++--+-----++++--+- | 64 |

Table 5: Binary codes for SFD (Mandatory for PRF128)

## PHR and PSDU

### PHR

For PRF128, a the PHR strucure shown in Figure 9 shall be used.

|  |  |  |
| --- | --- | --- |
| Bits: 0 | 1-10 | 11 |
| Reserved | Frame Length | Ranging |

Figure 9: PHR structure for PRF128

The Reserved field shall be set to zero. The value “1” is reserved for future use.

The Frame Length field shall be an unsigned integer number that indicates the number of octets in the PSDU field. The Frame Length field shall be passed to the modulator MSB-first.

The Ranging field shall be set to one if the current frame is an RFRAME and shall be set to zero otherwise.

For PRF128, the data rate of the PHR always equals the data rate of the PSDU, being 7.8 Mbit/s.

### PSDU

For PRF128, the data rate of the PSDU shall always be 7.8 Mbit/s.

### Data Modulation

PHR and PSDU data are modulated using BPSK at a mean PRF of 124.8 MHz.

The symbols are divided into two sub-symbols. The linear feedback shift register (LFSR) defined in [1] §16.3.2 shall be used as scrambler. Time Hopping cannot not be used as there are no hopping positions available. During a burst, all pulses shall be separated by one guard chip.

The following formula describes the modulation:

Here:

 is the sub-symbol index

, are the two parity bits generated by the FEC for every data bit

 is the chip period
Nppb is the number of pulses (active chips) per burst

 is the pulse shape

 is the scrambling sequence

The following figure shows how the symbol is formed of two sub-symbols.



Figure 10: Data modulation scheme for PRF128

### Data Encoding

For all 124.8 MHz mean PRF frame formats, PHR and PSDU encoding shall be achieved using a convolutional code of constraint length 7, with the generator polynomials G=(133,171). Use of this coding scheme results in a data rate of 7.8 Mbit/s.

The following figure shows an illustration of the encoder.



Figure 11: Coding scheme for PRF128

After encoding the PHR the encoder shall be forced to the all-zero state by appending 6 tailbits to the PHR bits.

At the end of the PSDU, the encoder shall be forced to the all-zero state by appending 6 tailbits to the PSDU bits.

## Scrambled Timestamp Sequence

### Deterministic Random Bit Generator

For PRF128, the DRBG is identical to the PRF64 DRBG described in Section 5.3.1.

### Sequence Length and Gaps

For PRF128, the STS building block structure is identical to the PRF64 structure described in Section 5.3.2.

### STS, PRF128 Mode (Mandatory)

This mode provides the highest entropy/time density that can be realized without changing the peak PRF of the SHR as defined for 62.4 MHz nominal mean PRF in [1].

The DRBG sequence Ai is mapped directly to the STS code Ci using the equation:

,

The resulting sequence Ci is then spread using δL = 4.

# MAC

This is a partial list of capability fields that need to be included in the MAC to support the changes described above:

* Specifying the STS frame format: 0 = No STS (mandatory), 1 = STS before PHR (mandatory), 2 = STS after PSDU (optional), 3 = STS-Only (optional)
* Specifying the STS building block configuration parameters
* Specifying the NIST DRBG Key and Seed initial value
* Specifying the capability and configuration of PHR being at specified data rate 6.8 Mbit/s or 27 Mbit/s (optional)

In addition to these changes, there are also changes required in order to provide efficient support for secure ranging for use cases where one device needs to simultaneously range with multiple nodes. While this can be done by sequential secure ranging exchanges with each one of those nodes, this will increase traffic considerably and will lead to increased power/energy consumption and increased latency. Hence, these use cases will benefit from the definition of an optimized multicast/broadcast secure ranging mode. In this contribution, we outline a possible approach for multicast/broadcast secure ranging and the required MAC changes in order to support them.

We make the following distinction between multicast and broadcast ranging. In multicast ranging, we assume that the target set of SRDEVs are known to the originating SRDEV, i.e., that each target SRDEV is known to the originating SRDEV by a specific ID, this ID being used to receive the STS. In broadcast ranging, we assume that the target SRDEVs are not known beforehand to the originating SRDEV. However, both the originating SRDEV and target SRDEVs share a common key that can be used to receive the STS.

The proposed multicast/broadcast secure ranging scheme uses a virtual slot based ranging approach. In this approach, the originating SRDEV sends a multicast/broadcast ranging request to the intended devices. Each target device sends its own response in a virtual slot that is relative to the ranging request. These virtual slots are not determined beforehand and do not require any global synchronization. Specifically,

1. Originating SRDEV sends a pre-poll packet **PP**. In this packet, it will advertise the parameters needed by the target SRDEV to complete the ranging exchange, specifically, the number of virtual slots in one ranging exchange during which it expects to receive responses from target SRDEVs, the duration of each virtual slot, and other information needed by the target SRDEV to decrypt the STS.
2. The pre-poll packet will send a first poll ranging packet **P1** to the target SRDEV.
3. The order in which the target SRDEVs will send their responses **R1**,**R2**, …, **RN** can be determined beforehand. How the order is determined is beyond the scope of this standard. However, this order is can be optionally communicated between the originating SRDEV and the target SRDEVs in the pre-poll packet.
4. The originating SRDEV will, optionally, a second ranging poll message **P2** to the target SRDEVs.
5. The target SRDEV, will use the same order to send their time stamps packets **T1**,**T2**, …, **TN**.



Figure 12: Scheduled Response Ranging

1. Alternatively, the target devices contend for these virtual slots in sending their response packets as well as the time stamp packets.



Figure 13: Response Contention Based Ranging

1. The originating SRDEV will, optionally, send a timestamps packet **TO** to the target SRDEVs that contains all of its timestamps. This will allow target SRDEVs to have their own secure ranging estimates for the target SRDEV.
2. The originating SRDEV can optionally repeat the ranging exchange. In this case, the originating SRDEV will advertise the number of rounds the ranging exchange will be repeated and the index of the current round in each pre-poll packet.



Figure 14: Sequence of ranging cycles

1. Note that the pre-poll **PP** can be merged into the first poll ranging packet **P1**.
2. Note that the pre-poll **PP** can be merged into the first poll ranging packet **P1**.
3. For each ranging round, the beginning and the end of each virtual slot at each of the target SRDEVs, is defined relative to the time the first packet in the ranging cycle (whether the ranging cycle starts with a pre-poll packet **PP** or a first poll ranging packet **P1**) is received.

In order to support this proposed optimized multicast/broadcast ranging, a number of fields need to be included in the MAC. Specifically:

* Specify the multicast/broadcast ranging mode: 0=single node, 1=multicast, 2=broadcast
* Number of ranging rounds and number of virtual slots in each round
* Duration of virtual slot
* Ranging round counter in the current exchange
* If multicast ranging, specify whether target SRDEVs are scheduled or whether they will contend.
* If multicast ranging and target SRDEVs are scheduled, specify virtual slot assignment (this could be part of the upper layer protocols).

# References

1. IEEE Standard for Low-Rate Wireless Networks (IEEE Std 802.15.4™-2015).
2. NIST SP 800-90A Rev. 1 (2015).

# Appendix

## Example data for STS construction

### DRBG data

This example illustrates the first two blocks generated by the DRBG described in Section 6.3.1 using following initialization:

key = 0x14148674D1D336AAF86050A814EB220F

data = 0x362EEB34C44FA8FBD37EC3CA1F9A3DE4

b = 0x00000000

DRBG Blocks:

B(0) = 0x7AA6F63EF917AE47115EB6FE3B5A5791

B(1) = 0x41DA0C7503566357EBF38B2C12BB3E92

C(0:255) = 0111101010100110111101100011111011111001000101111010111001000111
0001000101011110101101101111111000111011010110100101011110010001
0100000111011010000011000111010100000011010101100110001101010111
1110101111110011100010110010110000010010101110110011111010010010

### Example PRF64 Hopped STS

This example illustrates how bits C(0:255) in Section 9.1.1 are mapped to pulses, as described in Section 5.3.3. Symbols A(i) are then spread by δL = 4 chips.

A(0:255) =
-00-0+0+0+0+-00+0-0--00++00-0-0+0-0-0+-0+0-0-00-0+0+0-0+-0+0-00-
+0-0+0-0-0-00-0+0+0--00+0-0-0-0++00-0+0--0-00+0+-0-0-00-0+-0+0-0
-0+0+0-00--00+0++0+00-+0-00--0-0+0+0+00--0-0-00+-00++00--0-0-00-
0-0+0+0-0-0-+00-0++00+0-+00+0-+0+0-0+00+0+0-0+0-+00-0-0+0+-0+00+

### Example PRF64 Data Symbol Structure Based STS

This example illustrates how bits C(0:255) in Section 9.1.1 and PN bits S(0:127) are mapped to the first 128 pulses, as described in Section 5.3.4. Each A(i) corresponds to a single chip. PN bits are obtained by initializing the LFSR to S(-15:-1) = 111000101101101 as in [1], Table 16-10.

S(0:127) =
0010011101101110110100110110011011101011010101100111101111110101
0001100000111110010100001000010111100011000111000100101001001001

A(0:1023) =
-00000000000-00000000-000000+0000000+00000000-000+00000000000-00
0000-00000000+000+0000000000+0000-00000000000+0000000+000000+000
00000+0000000+000000+0000+000000+0000000-00000000+00000000000+00
0000+00000000-0000000+000000+000-00000000-0000000+0000000000-000
0-0000000+0000000-000000-00000000+000000-000000000000+0000000-00
0000+00000000+00-000000000000-000000-00000000+0000000+000000+000
+000000000000+0000000-0000000+000+000000-000000000000-0000000-00
0+0000000+0000000+00000000000+000000+0000+000000+00000000+000000
-0000000+0000000+00000000+00000000000+00-00000000000+0000000+000
+0000000+000000000000+000-0000000+00000000000+000+000000-0000000
+00000000-000000+000000000000+00-0000000-0000000-00000000000+000
0+0000000000+000+00000000000-000-00000000+000000-000000000000+00
00000+0000000-0000000-000000-0000000-0000000-0000-00000000000+00
0000+000+00000000000+00000000+000-00000000000-000000-000+0000000
+00000000+000000+00000000000+00000000-000000-00000000-000000-000
+000000000000+000000-0000000+00000000-00-0000000+000000000000-00

### Example PRF128 STS

This example illustrates how bits C(0:255) in Section 9.1.1 are mapped to the first 256 pulses of the STS, as described in Section 6.3.3. Symbols A(i) are then spread by δL = 4 chips.

A(0:255) =
+----+-+-+-++--+----+--+++-----+-----++-+++-+----+-+---++-+++---
+++-+++-+-+----+-+--+--+-------+++---+--+-+--+-++-+-+----++-+++-
+-+++++---+--+-+++++--+++---+-+-++++++--+-+-+--++--+++--+-+-+---
---+-+------++---+++-+--++-+--+++++-++-+-+---+--++-----+-++-++-+