Guideline for unique identification of tires based on the pertinent ISO Standards, using RAIN RFID tags with GS1 SGTIN encodings

RAIN RFID Alliance Guideline

V.1.0
April 2021

Document Summary

Document Item	Current Value
Document Name	Guideline for unique identification of tires based on the pertinent ISO Standards, using RAIN RFID tags with GS1 SGTIN encodings
Document Date	April 2021
Document Version	1.00
Document Issue	26 April 2021
Document Status	Published
Document Description	

Contributors

Name	Organisation
Claude TETELIN	GS1 Global Office – Guideline Project Leader
Jordi BAETA	Avery Dennison RBIS
Thomas BECHERER	Continental
Shreyans BENGANI	Ecartes
Chris BROWN	Printronix Auto ID
Riccardo GIOVANNOTTI	Bridgestone
Dong LANFEI	Mesnac
John FENKANYN	Goodyear
Marc FLEDERUS	FineLine Technologies
Philippe LALLEMENT	Michelin
Hiroshi OTSUKA	Air.Ocn.Ne
Yoshiaki OTSUKA	JATMA
Goetz PFEIFFERLING	Syo.io Limited
Neil PURVES	Bridgestone
Peter RAMIREZ	Michelin
Shunsuke TSURIMOTO	JATMA

Table of Contents

1. Introduction

The tire industry has recently contributed to the publication of four ISO standards defining the core specifications of RFID implementation for tire traceability. One key specification is the adoption of the RAIN technology for the RFID chip communication protocol. A second one is the adoption of a unique item identifier based on the GS1 SGTIN-96 coding.

This document production has been suggested as an extension of this ISO work and provides recommendations and guidelines for an adequate implementation of the SGTIN-96 coding by the tire manufacturers.

Section 2 of this document provides a general overview of purpose and application for unique identification and tagging of tires.

Section 3 summarizes the ISO 20910 and ISO 20912 requirements and explain how these standards are linked to other international standards, especially the GS1 air interface protocol and tag data encodings.

Section 4 provides a step-by-step process to create and encode unique GS1 SGTIN-96 identifiers into RFID tags. These steps are described in more detail in sections 5 to 9.

Section 9 provides a glossary of terms used throughout this document.

2. Purpose and application for unique ID and tagging of tires

Like many industries, tire identification and its associated product information requirements are facing multiple stakeholders' needs through a complex product life cycle.

Typical stakeholders and sample information requirements these stakeholders might face are:

- Tire manufacturers (eg: manufacturing process, inventory, logistics, …)
- Vehicle makers / OEM (eg: part conformance, part traceability, …)
- Tire dealers (eg: product information, order to sale, inventory, claim management, …)
- End users (eg: product information, customer experience, …)
- Government agencies (import/export regulations, recall management, …)
- Tire retreaders (eg: casing ownership, casing history management, …)
- Vehicle fleets (eg: tire asset management, …)
- Tire recycling (eg: product/content identification, …)

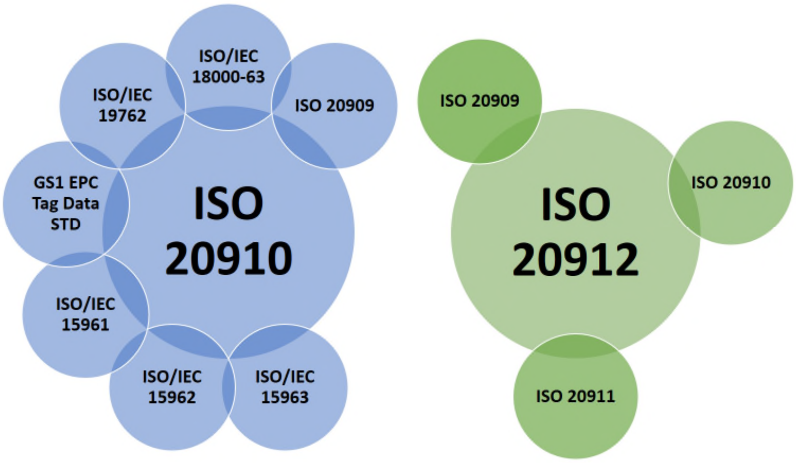
The concept of a Unique Item Identifier (UII; also commonly referred to as an "EPC number") is a key enabler at every stage of the product life cycle where a specific tire can be uniquely identified without knowing upfront in which country or for which stakeholder the tire is going to be used.

It can also enable access to product information services platforms where every stakeholder can be provided with fine-tuned sets of information depending on their requirements.

It therefore becomes critical that every RFID enabled tire must be uniquely and correctly identified through its SGTIN-96 code to ensure proper operations downstream after tire manufacture.

3. ISO 20910 and ISO 20912 requirements

The Figure below shows the relation between ISO 20910 and ISO 20912 standards and other ISO/IEC and GS1 standards.



NOTE: *ISO 20912 is shown for completeness. ISO 20910 is the core standard discussed in this document.*

3.1. ISO 20910 Coding for radio frequency identification (RFID) tire tags

This standard specifies the terms and definitions, general requirements, and data structure for coding radio frequency identification (RFID) tire tags.

It describes tire RFID data construction best practices, processes and methods for tire identification, verification, traceability, and product characteristics.

Minimum requirements for passive RFID tags are reported and are not intended to limit the development of the technology in the future.

An RFID tag in tire applications follows strict rules so the data format can be used globally within the tire and transportation industries.

This standard:

— specifies the air interface standards required between the RFID interrogator (also known as a reader) and RFID tag;

— specifies the use of industry recognized GS1 coding;

— clarifies the role of the company prefix;

— specifies the semantics and data syntax to be used in the data construction for Memory Bank 01 ("MB01," also commonly known as the "EPC" or "UII" memory bank);

— provides a unique item identifier for traceability (known as the EPC or UII number, stored in MB01);

— provides information on the use of the EPC/UII in MB01 as the link to database infrastructures;

— specifies the option for user memory bank (MB11).

This standard is based on the requirements in the following:

— technology: only passive ultra-high frequency (UHF) tags (RAIN tags) are used;

— global interface: the air interface protocol is ISO/IEC 18000-63, which is equivalent to GS1 EPC Gen2 V2.0.1.

A unique item identifier (EPC/UII number) is not the same as a part number, a DOT tire identification number (TIN), a CCC code, a branded ID, or other names for tires. Foremost in making an ideal EPC/UII is to know who made the part. Because of this, the EPC/UII is often referred to as the "birth record". A constraint with internally derived numbering schemes is that they are only unique to the domain owner; they may have duplicates within the "world", and rarely communicate intelligently who the owner actually is (outside of their domain). To clearly identify who, there must be a globally recognized register of companies. That is where GS1 plays a key role.

The international role of GS1 is to register companies and issue assigned company prefixes, in much the same way that cell phones have assigned company IDs as provided by international telecommunications agencies.

3.2. ISO 20912 Conformance test methods for RFID enabled tires

This document refers to RFID tags in the RAIN RFID band from 860 to 930 MHz.

Defines the test methods for validating the conformance of RFID enabled tires with the minimum reading distance specifications given in ISO 20909.

Indeed, different RFID tag attachment solutions can be considered as per ISO 20911 and the tests reported in this document allow the tire manufacturer to evaluate whether the selected option is suitable or not to grant the RFID enabled tire minimum transmission performance level.

During the development of this document, attention was paid on the key parameters influencing the test results. However, other parameters remain non-specified, so a testing lab can use those parameters at their discretion to perform the test.

The use cases have been simplified by considering a standalone/unmounted tire and describing a test set-up that can be used throughout the tire's lifetime and/or the tire's supply chain.

Additional use cases and more precise, detailed, and traceable testing methodologies may be added in future revisions of this document as RFID technology and its adoption moves forward.

The specifications in this document are not intended to limit any additional verification.

The two presented methodologies give comparable test results only when the same radio frequency and energy power parameters are used.

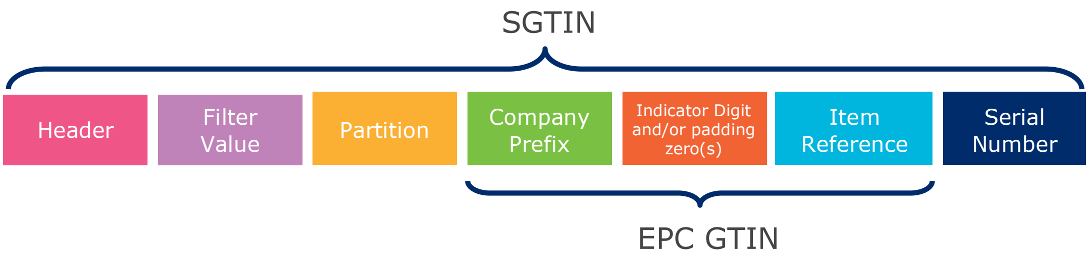
When conformity decisions are being made among various stakeholders, objective comparisons can be performed using the results obtained applying this document.

This standard is not intended to set any requirement on mass production quality control, nor on the frequency for testing.

4. Getting Started - Operating Manual

This section summarises the different steps a tire manufacturer must go through when the decision to provide RFID enabled tires to their customers has been taken. Details on these steps are provided throughout the other sections later in the document.

The overall structure and the different components of SGTIN (Serialized Global Trade Item Number) are described in the following figure (see section 5.2).

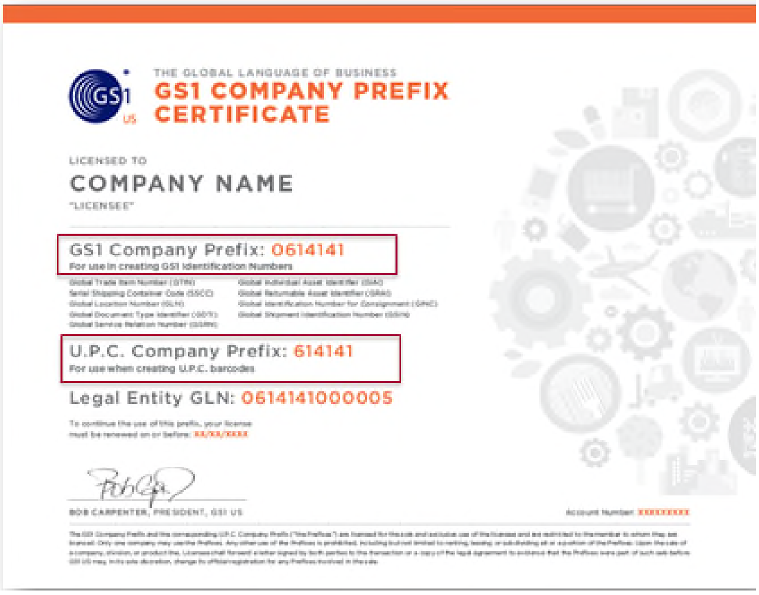


4.1. Step 1 – Choosing the right Company Prefix

To create an SGTIN, the tire manufacturer shall have a valid U.P.C. Company Prefix (only allocated by GS1 US) or a GS1 Company Prefix (GCP) (see section 5.3.1). If tire manufacturers have no assigned Company Prefix, they need to apply to one of the GS1 Member Organisations.

__NOTE:__ It is good practice to apply for a Company Prefix to a GS1 Member Organisation where the tire manufacturer's Headquarters are based.

If the tire manufacturer owns multiple valid Company Prefixes, they need to decide what prefix to use for their SGTIN assignment. The figure below shows an example of a GS1 Company Prefix certificate issued by a GS1 Member Organisation to a licensee.



THE GLOBAL LANGUAGE OF BUSINESS

GS1 COMPANY PREFIX CERTIFICATE

LICENSED TO

COMPANY NAME

"LICENSEE"

GS1 Company Prefix: 0614141

For use in creating GS1 Identification Numbers

Global Trade Item Number (GTIN)
Serial Shipping Container Code (SSCC)
Global Location Number (GLN)
Global Document Type Identifier (GDTI)
Global Service Relation Number (GSRN)
Global Individual Asset Identifier (GIAI)
Global Returnable Asset Identifier (GRAI)
Global Identification Number for Consignment (GINC)
Global Shipment Identification Number (GSIN)

U.P.C. Company Prefix: 614141

For use when creating U.P.C. barcodes

Legal Entity GLN: 0614141000005

To continue the use of this prefix, your license must be renewed on or before: XX/XX/XXXX

BOB CARPENTER, PRESIDENT, GS1 US

Account Number: XXXXXXXXX

The GS1 Company Prefix and the corresponding U.P.C. Company Prefix ("the Prefixes") are licensed for the sole and exclusive use of the licensee and are restricted to the member to whom they are licensed. Only one company may use the Prefixes. Any other use of the Prefixes is prohibited, including but not limited to renting, leasing, or subdividing all or a portion of the Prefixes. Upon the sale of a company, division, or product line, Licensee shall forward a letter signed by both parties to the transaction or a copy of the legal agreement to evidence that the Prefixes were part of such sale before GS1 US may, in its sole discretion, change its official registration for any Prefixes involved in the sale.

See section 5.3.1 for more information about a GS1 and U.P.C. Company Prefixes.

For more information "About the GS1 Company Prefix" watch this short video: *https://www.youtube.com/watch?v=Aft2kdhrq5E*

4.2. Step 2 – Defining zero padded Item Reference number

Every single type of tire shall have a unique Item Reference assigned by the tire manufacturer. The length of the Item Reference that can be assigned to a type of tire depends on the type and length of the Company Prefix (see section 5.3).

When using GS1 Company Prefix, one padding zero must be added and the overall length of {Company Prefix; Padding zero; Item Reference} shall be 13 digits. Therefore, if N is the length of the GS1 Company Prefix, the length of Item Reference is 13-(N+1).

When using U.P.C. Company Prefix, two padding zeros must be added and the overall length of {Company Prefix; Padding zeros; Item Reference} shall be 13 digits. Therefore, if M is the length of the U.P.C. Company prefix, the length of Item Reference is 13-(M+2).

The tire manufacturer shall define a way to assign Item Reference numbers to tire types at corporate level to comply with the GS1 principle that: 1 Item Reference=1 product type and 1 product type = 1 Item Reference.

4.3. Step 3 – Converting Company Prefix and zero padded Item Reference number to binary strings

Following the length of Company Prefix, translate the digital values of both Company Prefix and zero padded Item Reference number into binary strings. The length of each binary string is defined in Table 1.

Table 1 Lengths of Company Prefix and zero padded Item Reference number

Company Prefix length		Zero padded Item Reference number length	
Digits	Bits	Digits	Bits
12	40	1	4
11	37	2	7
10	34	3	10
9	30	4	14
8	27	5	17
7	24	6	20
6	20	7	24

Sum is always 13-digit long

Sum is always 44-bit long

4.4. Step 4 – Defining Partition Value

The Partition Value is a 3-bit long binary string which indicates the length of the Company Prefix (see section 5.6).

The Partition value is derived from the Company Prefix length (see Table 2).

Table 2 Relation between Partition Value and Company Prefix length

Partition Value	Company Prefix length
Bits	Digits
000	12
001	11
010	10
011	9
100	8
101	7
110	6

4.5. Step 5 – Defining Header and Filter Value

The Header and Filter Value are defined as per GS1 Tag Data Standard. The Header is an 8-bit long binary string that indicates which type of EPC identifier is used. For SGTIN-96, the Header shall be $(0011\ 0000)_2$.

The Filter Value is a 3-bit long binary string that indicates the type of handling unit of the product. For unique identification of tires, the Filter Value shall be $(000)_2$.

4.6. Step 6 – Serial Number

Every single tire must be identified by a unique Serialized GTIN (SGTIN). Tire manufacturers must define a serialisation process at a corporate level to fit business needs and to avoid duplication of SGTIN numbers.

Section 6 gives examples of serialisation processes.

The Serial Number of the SGTIN-96 is a 38-bit long binary string (see section 5.2). Therefore, the Serial Number shall be a number between 0 and 274,877,906,943 (2^{38}-1)

4.7. Step 7 – Creating SGTIN-96 binary string

Based on the previous steps, identify the values and length of the different components: Header, Filter Value, Partition Value, Company Prefix, zero padded Item Reference and Serial Number.

All the previous binary strings must be concatenated in one single 96-bit long string as per the Table 3 below and encoded in the EPC memory of the RAIN tag starting at address 20h.

Table 3 Components of SGTIN-96

Component	Header	Filter Value	Partition	Company Prefix	Zero padded Item Reference	Serial
Length (bits)	8	3	3	X	44-X	38
Value	0011 0000	000	As per Company prefix length	As per Company Prefix value	As per tire manufacturer allocation	As per tire manufacturer allocation

4.8. Example 1: GS1 Company Prefix

4.8.1. Step 1 - Choosing the right Company Prefix

Your Company Prefix is a GS1 Company Prefix and its value is 9524141 which is 7-digits long.

4.8.2. Step 2 – Defining zero padded Item Reference

Since you are using a 7-digit GS1 Company Prefix, the Item Reference number shall be 5-digit long (6 digits including 1 padding zero).

Assume that you allocate the Item Reference 00002 (5-digit long) to the type of tire you want to identify.

4.8.3. Step 3 – Converting Company Prefix and zero padded Item Reference in binary strings

As per Table 1, you must translate the GS1 Company Prefix 9524141 into a 24-bit long binary string:

$(1001\ 0001\ 0101\ 0011\ 0101\ 1101)_2$

The next step is to add a leading zero to the Item Reference Number (00002) and to convert this value: 0 00002 into a 20-bit long binary string:

$(0000\ 0000\ 0000\ 0000\ 0010)_2$

4.8.4. Step 4 – Defining Partition Value

As per Table 2, since the GS1 Company Prefix is 7-digit long, the Partition value is $(101)_2$.

4.8.5. Step 5 – Defining Header and Filter Value

The Header and Filter values shall be $(0011\ 0000)_2$ and $(000)_2$ respectively.

4.8.6. Step 6 – Serial Number

Assume that you want to uniquely identify the 648th instance of such a tire. The Serial Number is 647. The digital representation of the Serial Number must be translated into a 38-bit long binary string:

$(00\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0010\ 1000\ 0111)_2$

4.8.7. Step 7 – Creating SGTIN-96 binary string

The binary string of this SGTIN-96 will be the concatenation of the preceding values as per Table 3:

$(0011\ 0000\ 0001\ 0110\ 0100\ 0101\ 0100\ 1101\ 0111\ 0100\ 0000\ 0000\ 0000\ 0000\ 1000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0010\ 1000\ 0111)_2$

This can be represented in hexadecimal by:

(3 0 1 6 4 5 4 D 7 4 0 0 0 0 8 0 0 0 0 0 0 2 8 7)h

4.9. Example 2: U.P.C. Company Prefix

4.9.1. Step 1: Choosing the right Company Prefix

Your Company Prefix is a U.P.C. Company Prefix and its value is 952156 which is 6-digit long.

4.9.2. Step 2 – Defining zero padded Item Reference

Since you are using a 6-digit long U.P.C. Company Prefix, the Item Reference number shall be 5-digit long (7 digits including two padding zeros).

Assume that you allocate the Item Reference 00004 (5-digit long) to the type of tire you want to identify.

4.9.3. Step 3 – Converting Company Prefix and zero padded Item Reference in binary strings

As per Table 1, you must translate the U.P.C. Company Prefix 952156 into a 20-bit long binary string:

$(1110\ 1000\ 0111\ 0101\ 1100)_2$

The next step is to add two leading zeros to the Item Reference Number (00004) and to convert this value: 00 00004 into a 24-bit long binary string:

$(0000\ 0000\ 0000\ 0000\ 0000\ 0100)_2$

4.9.4. Step 4 – Defining Partition Value

As per Table 2, since the U.P.C. Company Prefix is 6-digit long, the Partition value is $(110)_2$.

4.9.5. Step 5 – Defining Header and Filter Value

The Header and Filter values shall be $(0011\ 0000)_2$ and $(000)_2$ respectively.

4.9.6. Step 6 – Serial Number

Assume that you want to uniquely identify the 4330th instance of such a tire. The Serial Number is 4329. The digital representation of the Serial Number must be translated into a 38-bit long binary string:

$(00\ 0000\ 0000\ 0000\ 0000\ 0000\ 0001\ 0000\ 1110\ 1001)_2$

4.9.7. Step 7 – Creating SGTIN-96 binary string

The binary string of this SGTIN-96 will be the concatenation of the preceding values as per Table 3:

$(0011\ 0000\ 0001\ 1011\ 1010\ 0001\ 1101\ 0111\ 0000\ 0000\ 0000\ 0000\ 0000\ 0001\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0001\ 0000\ 1110\ 1001)_2$

This can be represented in hexadecimal by:

(3 0 1 B A 1 D 7 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 E 9)h

4.10. Final step – Commissioning Tire tag

To be compliant with GS1 Tag Data Standard and ISO 20910, the following steps are required before letting the tire tag leave the tire manufacturer's facilities.

- Verify that the Length bits of the PC Word (bits 10h-14h) are set to 00110_2
- Verify that Toggle bit of the PC Word (bit 17h) is set to 0_2.
- Verify that the 8 Attribute bits of the PC Word (bits 18h-1Fh) are set to 00000000_2.
- Verify that SGTIN-96 has been well encoded (bits 20h-79h)
- Permalock the EPC memory
- Verify that the 32-bit Kill password is set to 0 (bits 00h-1Fh of RESERVED memory bank)
- Permalock the Kill password

Once all the previous steps have been fulfilled, the tire can leave the factory and live its life.

5. Creating Serialized GTIN (SGTIN)

5.1. RAIN RFID requires uniqueness of identifiers

One of the main advantages of RAIN RFID (UHF passive RFID) technology is about making high read rate and accurate inventories. The air interface protocol has been developed to achieve this goal, but this has also been made possible because RAIN tags can be read without line of sight.

Therefore, when doing inventory of products (tires) in the field of view of a reader (ie., that can be read by an RFID reader), each must have a tag with a unique EPC/UII number so it can be differentiated from the others. Without such unique identifiers, RAIN readers would be unable to count the total number of products.

RAIN RFID can also be implemented because the business process requires unique identification of products and can be based on the desire to have more efficient processes like stock management, supply chain visibility, product recalls, and much more. Local regulations may also be a factor that determines implementation.

Whether technical, business, or regulatory driven, RAIN tags that are attached to products shall carry unique identifiers in the EPC/UII memory.

As per ISO 20910, tag encoding for tires shall be implemented using GS1 SGTIN-96 format. SGTIN-96 is one of the GS1 keys and identifiers that are defined in GS1 Tag Data Standard. This standard is available here: *https://www.gs1.org/standards/epc-rfid/tds*

5.2. Structure of the SGTIN-96

SGTIN-96 is based on the GS1 GTIN (Global trade Item Number) identification key (see section Annex A.A.2). GTINs are widely used for EAN or U.P.C. barcodes (see sections Annex A.A.5 and Annex A.A.6) and are used to identify types of products. For example, two identical 1,5 litres bottles of water will have the same GTIN. But a 0,75 litre bottle of the same brand will have a different GTIN.

There are four different types of GTINs: GTIN-8, GTIN-12, GTIN-13 and GTIN-14. For tire identification, only GTIN-12 and GTIN-13 must be considered.

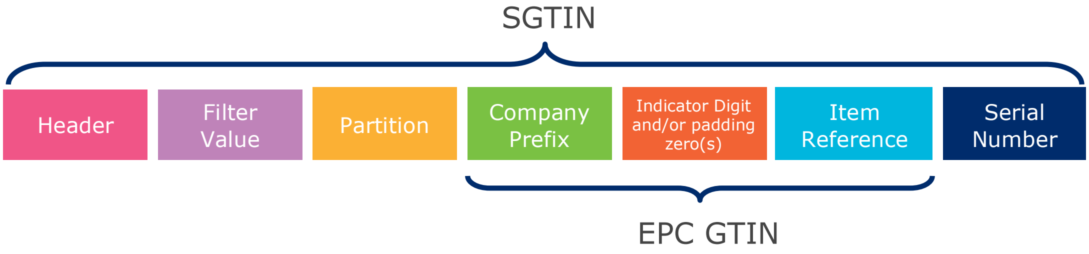
GTIN-12 are 12-digits long and are based on U.P.C. Company Prefix (only allocated by GS1 US). GTIN-13 are 13-digits long and are based on GS1 Company Prefix allocated by GS1 Member Organizations or GS1 Global Office (see section Annex A).

The main GTIN-12 and GTIN-13 components are:

- The Company Prefix (see section 5.3.1)
- The Item Reference (see section 5.3.2)
- A Check Digit

> ***NOTE:*** *the Check Digit is derived from the value of the other components and added at the end of the GTIN (see section Annex A). When creating SGTIN (to be encoded in RFID tag), this Check Digit is simply ignored.*

To create SGTINs, we need to add some additional information as described in the Figure below and especially a Serial Number that will serve to uniquely identify each instance of a tire type.



> ***NOTE:*** *Indicator Digit is only used for GTIN-14. As the creation of SGTIN-96 numbers for tires is based only on GTIN-12 and GTIN-13 constructs, Indicator Digit will no longer be considered in the rest of the document.*

5.3. Company Prefix, Item Reference and padding zero(s)

When creating SGTIN-96 to be encoded in RAIN RFID tags, the length of the concatenation of Company Prefix and padding zero(s) and Item Refence shall always be 13-digit long.

To ensure compatibility between GTINs and SGTINs representations, padding zero(s) must be added in front of the Item Reference to achieve this.

When using a U.P.C. Company Prefix, two padding zeros must be added in front of the Item Reference. When using a GS1 Company Prefix, one single padding zero must be added in front of the Item Reference.

The lengths of Company Prefix and zero padded Item Reference are summarized in Table 4. Table 4 Lengths of Company Prefix and zero padded Item Reference number

Company Prefix length		Zero padded Item Reference number length	
Digits	Bits	Digits	Bits
12	40	1	4
11	37	2	7

Company Prefix length		Zero padded Item Reference number length	
10	34	3	10
9	30	4	14
8	27	5	17
7	24	6	20
6	20	7	24

Sum is always 13-digit long

Sum is always 44-bit long

5.3.1. Company Prefix

A Company Prefix is a unique string of four to twelve digits used to issue GS1 Identification Keys. The Company Prefix is issued by a GS1 Member Organisation or by GS1 Global Office.

To create a GS1 Identification Key like an SGTIN, a company shall license, at least, one Company Prefix. As the use of the GS1 system has grown, some companies have exhausted their prefix capacity and needed to acquire additional Company Prefixes. So, a Company Prefix does not identify a company; a Company Prefix is associated with a company.

Two kinds of Company Prefixes can be assigned to a Company: a GS1 Company Prefix or a U.P.C. Company Prefix (only assigned by GS1 US). The first one is used to create EAN barcodes, the second one is used to create U.P.C. barcodes (see section Annex A).

When you want to create GS1 SGTIN, you can choose whatever Company Prefix you licensed. There are no rules other than the ones described in section 5.3.2 (i.e., all the tires that belong to a given tire type shall be identified by the same GTIN).

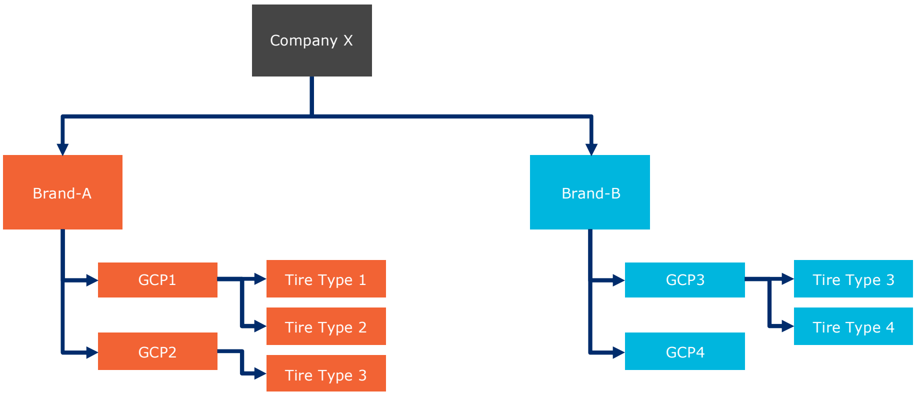
Example

A company X manufactures tires under 2 different brands, Brand-A and Brand-B. Brand-A operates mainly in Europe, Middle East, and Africa. Brand-B in North America, and Asia. Brand-A has licensed Company Prefix1 in GS1 Germany and Company Prefix2 in GS1 Italy. Brand-B has licensed Company Prefix3 in GS1 US and Company Prefix4 in GS1 Japan.

If the company X decides to manufacture a new tire (Tire Type 1) under Brand-A, it can choose to assign SGTINs based on either Company Prefix1 or Company Prefix2. Once the Company Prefix has been chosen, it cannot be changed. If another kind of tire (Tire Type 2) is designed and manufactured by the same Brand-A, different SGTINs must be assigned to these tires. These SGTINs can be based on the same Company Prefix than the previous tire model or not. The choice of the Company Prefix does not impact the country in which the tires can be distributed and sold.

> ***NOTE:*** *Company X can decide to have only one Company Prefix that will be used by the two brands, but this could create issues in case the Company X decided to sell one of these brands to another company.*

This example is summarised in the picture below. Note that even if the same Tire Type 3 is manufactured by both Brand-A and Brand-B, it shall be identified using two different Company Prefixes since the brands are different.



5.3.2. Item Reference Numbers and GTIN management rules

The two fundamental rules for GTIN allocation are:

1 tire type = 1 GTIN

1 GTIN = 1 tire type

When making decisions about tire identification, it is important to understand the differences between a NEW tire and changes to an existing tire type.

New tire types are those which do not currently exist in a brand owner's tire offering and are new to the marketplace. A new tire type should be considered different from the existing tire type offerings. The GTIN Management Rules require that if a tire type is new, it should **always** be assigned a different GTIN to accurately distinguish the new tire type from any tire type that is currently available.

Changes to existing tire type are considered "replacement tire type" (the previous version will no longer exist once the replacement tire type has flowed through). The GTIN Management Rules define when a change to certain attributes of an existing tire type requires a new GTIN.

For more information about GTIN allocation rules, refer to the GS1 standards: *https://www.gs1.org/1/gtinrules//en/*

Therefore, a new GTIN will **always** be created for a new tire type and any change in an existing tire type **may** impact its GTIN use.

Brand owner (Tire Manufacturer) shall allocate a unique Item Reference number to each tire type. The length of the Item Reference number depends on:

- The type of Company Prefix (U.P.C. or GS1) (see section 5.3.1)
- The length of the Company Prefix

GS1 recommends a sequential allocation of Item Reference Numbers as a best practice. Whatever the internal reference and the time at which the tire type has been made available to the market, Item Reference shall be only made of numeric digits and should be allocated sequentially.

Table 5 is an example of assignments to an existing list of tire types based on a 7-digit long GS1 Company Prefix. The Item Reference number shall therefore be 5-digit long.

Table 5 Example of sequential allocation based on 7-digit long GS1 Company Prefix

Product	Product type	Internal reference*	GS1 Company Prefix	Item Reference number
Motorcycle tire	160/70 B7	HFT-23/std12	9528765	00000
Truck tire	11R22.5	TRX-11/hyb1	9528765	00001

Product	Product type	Internal reference*	GS1 Company Prefix	Item Reference number
Personal car	195/55 R16	PRS-16/veh1	9528765	00002
Personal car	195/55 R18	PRS-18/veh3	9528765	00003
Personal car	195/55 R20	PRS-20/veh2	9528765	00004
Motorcycle tire	160/70 B8	HFT-24/std6	9528765	00005
Personal car	195/55 R15	PRS-15/veh2	9528765	00006
*Internal reference provided in this Table are purely fictional, any resemblance to existing or former tire references are accidental.				

Table 6 is another example of assignments to the same existing list of tire types in Table 5 but based on an 8-digit long U.P.C. Company Prefix. The Item Reference number shall therefore be 3-digit long.

Table 6 Example of sequential allocation based on 8-digit long U.P.C. Company Prefix

Product	Product type	Internal reference*	U.P.C. Company Prefix	Item Reference number
Motorcycle tire	160/70 B7	HFT-23/std12	95287654	000
Truck tire	11R22.5	TRX-11/hyb1	95287654	001
Personal car	195/55 R16	PRS-16/veh1	95287654	002
Personal car	195/55 R18	PRS-18/veh3	95287654	003
Personal car	195/55 R20	PRS-20/veh2	95287654	004
Motorcycle tire	160/70 B8	HFT-24/std6	95287654	005
Personal car	195/55 R15	PRS-15/veh2	95287654	006
*Internal reference provided in this Table are purely fictional, any resemblance to existing or former tire references are accidental.				

5.3.3. Padding zero(s)

Whatever the Company Prefix type (U.P.C. or GS1) and whatever the length of both the Company Prefix and Item Reference number, the segment to be encoded in SGTIN-96 shall be 13-digit long. Padding zero(s) must be added between Company Prefix and Item Reference number to achieve this number of digits.

To identify tires with SGTIN-96, the following rules apply:

When using GS1 Company Prefix, one padding zero must be added and the overall length of {Company Prefix; Padding zero; Item Reference} shall be 13 digits. Therefore, if N is the length of the GS1 Company Prefix, the length of Item Reference is 13-(N+1).

When using U.P.C. Company Prefix, two padding zeros must be added and the overall length of {Company Prefix; Padding zeros; Item Reference} shall be 13 digits. Therefore, if M is the length of the U.P.C. Company prefix, the length of Item Reference is 13-(M+2).

5.3.4. Example using a GS1 Company Prefix (GCP)

Assume you have a 7-digit long GS1 Company prefix which is 9524141. Since it is a GS1 Company Prefix, the Item Reference number is 5-digits long and we must add one padding zero.

Assume that the Item Reference number is 12345 (which should mean that it the 12346th type of tire that you are manufacturing and selling).

Since the GCP is 7-digit long, it must be encoded into a 24-bit long binary string and the zero padded Item Reference number must be encoded in a 20-bit long binary string (see Table 4).

The GCP is interpreted as an integer value (9524141) which is easily translated into a binary format. Therefore, $(1001\ 0001\ 0101\ 0011\ 1010\ 1101)_2$ are the 24 bits to encode as the GS1 Company Prefix in the RFID tag (see Table 8).

The following bits to encode in the RFID tags correspond to the zero padded Item Reference number. Therefore, you must encode 012345 as a 20-bit long binary string (see Table 4).

$(0000\ 0011\ 0000\ 0011\ 1001)_2$ are the 20 bits to encode as the zero padded Item Reference number in the RFID tag (see Table 8).

5.3.5. Example using a U.P.C. Company Prefix

Assume you have an 8-digit long U.P.C. Company Prefix which is 95241412. Since it is a U.P.C. Company Prefix, the Item Reference number is 3-digits long and we must add two padding zeros.

Assume that the Item Reference number is 123 (which should mean that it the 124th type of tire that you are manufacturing and selling).

Since the U.P.C. Company prefix is 8-digit long, it must be encoded into a 27-bit long binary string and the zero padded Item Reference number must be encoded in a 17-bit long binary string (see Table 4).

The U.P.C. Company Prefix is interpreted as an integer value (95241412) which is easily translated into a binary format. Therefore, $(101\ 1010\ 1101\ 0100\ 0100\ 1100\ 0100)_2$ are the 27 bits to encode as the U.P.C. Company Prefix in the RFID tag (see Table 8).

The following bits to encode in the RFID tags correspond to the zero padded Item Reference number. Therefore, you must encode 00123 as a 17-bit long binary string (see Table 4).

$(0\ 0000\ 0000\ 0111\ 1011)_2$ are the 17 bits to encode as the zero padded Item Reference number in the RFID tag (see Table 8).

5.4. Header

The general structure of an EPC Binary Encoding as used on a tag is as a string of bits (i.e., a binary representation), consisting of a fixed length, 8-bit, Header followed by a series of fields whose overall length, structure, and function are determined by the header value.

SGTIN-96 is one of the available GS1 Tag Data Standard coding schemes. Its Header value is defined by the 8 following bits: $(0011\ 0000)_2$. These are the bit values that must be encoded in the RFID tag. This Header can also be represented as a hexadecimal string (30)h or by its equivalent decimal value (48).

5.5. Filter value

The filter value is additional control information that is included in the EPC memory bank of an RFID tag. The intended use of the filter value is to allow an RFID reader to select or deselect the tags corresponding to certain physical objects, to make it easier to read the desired tags in an environment where there may be other tags present in the environment.

Filter values are available for most EPC types. There is a different set of standardised filter values associated with each type of EPC, as specified by GS1 Tag Data Standard.

For SGTIN, the Filter value consists of a fixed 3-bit length binary string.

Filter Value $(000)_2$ is recommended for unique tire identification.

5.6. Partition Value

Since the SGTIN is made of a Company Prefix and zero padded Item Reference numbers, and since the Company Prefix may vary in length, the Partition Value indicates the number of bits for encoding the Company Prefix and the number of bits for encoding the zero padded Item Reference number.

The Partition Value is encoded within a 3-bit long binary string but only 7 different values can be used since there is only 7 different lengths of Company Prefixes.

The Table 7 describes the values and meanings of the Partition Value.

Table 7 SGTIN Partition Table

Partition Value		Company Prefix length		Zero padded Item Reference length	
Decimal	Bits	Bits	Digits	Bits	Digits
0	000	40	12	4	1
1	001	37	11	7	2
2	010	34	10	10	3
3	011	30	9	14	4
4	100	27	8	17	5
5	101	24	7	20	6
6	110	20	6	24	7

5.7. Serial number

The Serial Number is used to uniquely identify every instance of a tire. When using SGTIN-96 format, the serial number is encoded into a 38-bit long binary string. Serial numbers are therefore numeric values between 0 and 2^{38}-1 which is 274,877,906,943.

The translation and encoding are simple. You take your serial number as a decimal value and translate it as a 38-bit long binary string.

Example of serial number

Assume your serial number is 45 which could correspond to the 46th instance of a given GTIN (type of tire). The binary equivalent representation of 45 is $(10\ 1101)_2$. Therefore, you will encode $(00\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0010\ 1101)_2$ as the last 38 bits of the SGTIN-96.

5.8. Summary of SGTIN components

The lengths and values of the different SGTIN-96 components are described in Table 8.

Table 8 Structure of SGTIN-96

Data Structure of SGTIN-96			
EPC SGTIN-96	Field length (bits)	Value (decimal)	Comment

Header	8	48	Number for SGTIN-96
Filter value	3	0	**0** = All Others→ used for **individual tires**
Partition	3	0 to 6	GS1 allocated Company Prefix (CP) length 0 = CP of 12 digits 1 = CP of 11 digits 2 = CP of 10 digits 3 = CP of 9 digits 4 = CP of 8 digits 5 = CP of 7 digits 6 = CP of 6 digits
Company Prefix	20 to 40	N/A	GS1 allocated Company Prefix
Zero padded Item reference	24 to 4	N/A	Depends on the number of digits in Company Prefix. The total number of digits (Company Prefix + zero padded Item reference) shall be 13. Value to be defined and assigned by the tire manufacturer (see section 5.3.2)
Serial number	38	0 to 2^{38}	Up to **12 digits** with the highest allowable value of 274,877,906,943 since serial number is encoded with 38 bits within SGTIN-96. **Value to be defined by the tire manufacturers.** Companies must implement measures to **ensure serial number uniqueness** to allow for **unambiguous identification of each individual tire**.

NOTE: *Padding zero(s) for tire identification, shall be either 00 (for U.P.C. Company Prefix) or 0 (for GS1 Company Prefix).*

NOTE: *By definition, the total length of the binary encoding of Company Prefix and zero padded Item Reference number shall always be 13 digits (44 bits), whatever the length of the Company Prefix.*

5.9. Full SGTIN-96 Example

The company McIntire has been allocated a 7-digit GS1 Company Prefix (9528765) by GS1 for the creation of GS1 identifiers. After having created 7 different types of tires, this company is manufacturing an eighth model and has allocated the Item reference (00007) to this tire type.

NOTE: *Item Reference number 0 has been allocated to the first model of tire.*

To comply with ISO 20909, McIntire wants to identify each single tire of this model using RFID with SGTIN-96 format.

This company would like to know what the 96 bits are to be encoded in EPC memory of the tag for the tire that is allocated Serial Number 569.

First, as they are using SGTIN-96, the first 8 bits of the EPC memory represent the Header for SGTIN-96. As per GS1 TDS, these bits are (0011 0000)$_2$.

The next 3 bits represent the Filter Value and, following this guideline, they must set these 3 bits to (000)$_2$. Therefore, the next 3 bits of the EPC memory are (000)$_2$.

The value of the Partition needs to be defined next. Since their GCP is 7-digits long, the Partition Value is 5 and the next 3 bits to be encoded in the EPC memory are (101)$_2$.

Next, the decimal value of the GCP must be converted into a 24-bit long binary string. These 24 bits are (1001 0001 0110 0101 1011 1101)$_2$.

Since McIntire is using a GS1 Company prefix, we need to add a leading zero to the Item Reference number. The zero padded Item Reference Number value (0 00007) must be converted into a 20-bit long binary string. These 20 bits are (0000 0000 0000 0000 0111)$_2$.

Finally, the serial number 569 is converted into a 38-bit long binary string which is (00 0000 0000 0000 0000 0000 0000 0010 0011 1001)$_2$.

The binary values of all these components are summarised in the Table 9.

Table 9 Components of SGTIN-96

Component	Header	Filter Value	Partition	GCP	Zero padded Item Reference number	Serial
Length (bits)	8	3	3	24	20	38
Value	0011 0000	000	101	1001 0001 0110 0101 1011 1101	0000 0000 0000 0000 0111	00 0000 0000 0000 0000 0000 0000 0010 0011 1001

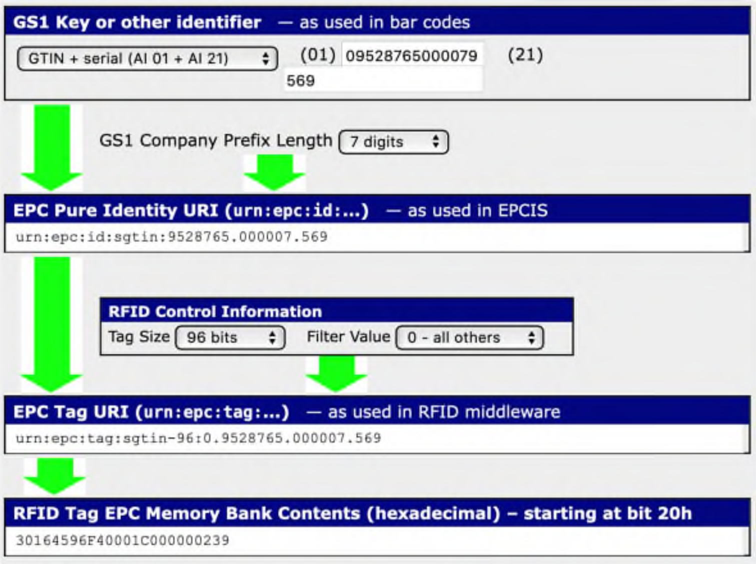
All these bits can be grouped and are usually represented as a string of hexadecimal characters (see Table 10).

Table 10 Binary and Hexadecimal representations of SGTIN-96

String of 96 bits:
0011 0000 0001 0110 0100 0101 1001 0110 1111 0100 0000 0000 0000 0001 1100 0000 0000 0000 0000 0000 0000 0010 0011 1001
String of 24 hexadecimal characters
3 0 1 6 4 5 9 6 F 4 0 0 0 1 C 0 0 0 0 0 0 2 3 9

To help create this binary string, use the GS1 tools available at *https://www.gs1.org/services/epc-encoderdecoder*

The Figure below shows the conversion made with the EPC Encoder/Decoder tool.



GS1 Key or other identifier — as used in bar codes

GTIN + serial (AI 01 + AI 21) (01) 09528765000079 (21) 569

GS1 Company Prefix Length: 7 digits

EPC Pure Identity URI (urn:epc:id:...) — as used in EPCIS

urn:epc:id:sgtin:9528765.000007.569

RFID Control Information

Tag Size: 96 bits Filter Value: 0 - all others

EPC Tag URI (urn:epc:tag:...) — as used in RFID middleware

urn:epc:tag:sgtin-96:0.9528765.000007.569

RFID Tag EPC Memory Bank Contents (hexadecimal) – starting at bit 20h

30164596F40001C000000239

6. Serialisation processes

6.1. Introduction

This section focuses on the importance of a strategy for managing the assignment of unique serial numbers to tires. The tires are mainly intended to be serialized "at source"; i.e., by the tire manufacturer so the RFID enabled tires are uniquely identified prior to leaving the tire manufacturers' facilities. There may be some limited need to assign serial numbers further downstream in the supply chain, e.g., in an exceptional situation where the source tag is missing.

In principle, assigning unique serial numbers is straightforward. For example, using a simple incrementing counter 1, 2, 3, ..., for each GTIN to assign the next unused serial number to each instance of that GTIN as it is manufactured. However, this becomes more complicated when serial numbers for the same product may be assigned in more than one physical location. Examples of situations in which this occurs are:

- The same product is manufactured on multiple manufacturing lines in the same building
- The same product is manufactured in different manufacturing plants
- One or more contract manufacturers are used
- Tire tags (including patches and stickers) are affixed to tires by one or more 3rd party service bureaus
- Pre-programmed tire tags (including patches and stickers) are obtained from one or more tag service providers and affixed to the products later
- Downstream supply chain parties (e.g., OEM, retailers, retreaders) need to assign new serial numbers in exception situations, e.g., where the original tag is missing, or a tire returned by a customer with the label removed is to be restocked.

6.2. Creating a Top-level Serialisation Plan

It is best practice for all tire manufacturers to create a top-level plan to define what serialisation methods are to be used. In creating a top-level plan, the tire manufacturer should understand the following requirements:

- What types of tires are to be serialized? What are their GTINs?
- What is the expected volume of each GTIN that will be manufactured over the life of the GTIN? This helps to assess how many serial numbers will be needed over time.
- Where will serialisation take place? In the brand owner's own manufacturing facility, in 3rd parties contracted by the brand owner (contract manufacturers, serviced bureaus, etc), by other supply chain parties?
- How many different internal facilities and/or 3rd parties will be used?
- What IT capabilities are available, or can be made available, to manage serialisation?
- How are the answers to the above expected to change over time?

These answers to these questions will help determine the various serialisation methods available.

The conclusion from analysing the available approaches is often that more than one approach needs to be supported. Sometimes, the tire manufacturer sees a need to use two approaches simultaneously. Other times, only one method is to be used at any given time, but the tire manufacturer wants flexibility to change to a different method in the future. This means that a top-level plan will generally include a high-level allocation of the full 38-bit serial number space, so that these different choices can be accommodated.

Regardless of the methods chosen in the top-level plan, the tire manufacturer is still responsible for quality control of serialisation, including ensuring that the correct GTIN and serial number according to the serialisation plan is encoded into the tag, that the tag is readable, and that verification is performed to ensure serial numbers are not duplicated.

6.3. Business scenarios

This sub-section illustrates the two commonly occurring business process scenarios for serialisation of tires. It illustrates different approaches for managing serial numbers to ensure uniqueness and shows how they may be applied to each of the business scenarios.

6.3.1. Tire manufacturing on a single line

The simplest business scenario for serialising tires is when a tire manufacturer tags a product on a single manufacturing line. Although this single manufacturing line is the only place where serial numbers for the given product are assigned, a process is needed to ensure uniqueness of serial numbers. The software that runs the manufacturing line has complete control over what serial numbers are issued; and so, on its own, it can employ a method to ensure uniqueness of serial numbers.

6.3.2. Tire manufacturing on multiple lines

A more complex business scenario is where the same type of tire (same GTIN) is manufactured and tagged on more than one manufacturing line. This includes having several manufacturing lines within the same plant, or manufacturing lines that are geographically separated. In this case, the challenge is to ensure that one manufacturing line does not use the same serial number that a different manufacturing line has already used for the same type of tire.

6.4. Examples of serialisation management

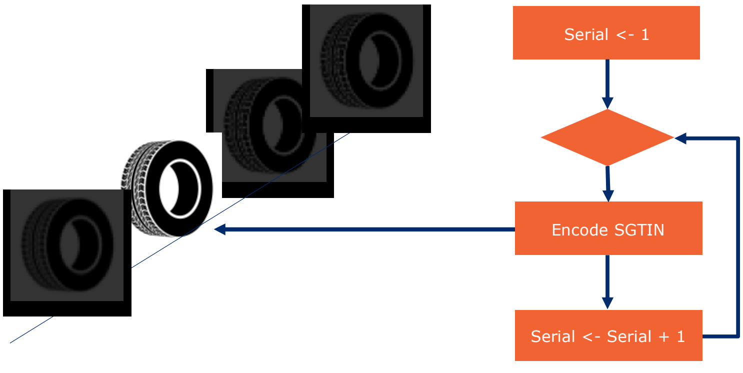
This section provides 2 examples for managing serialisation. The first one can be used when tires are manufactured on a single production line. The second one could be preferred in situations where the same type of tire is manufactured in multiple locations.

These descriptions are based on the use of SGTIN-96 so that the serial numbers are all-numeric, with no leading zeros, so that the serial number is a decimal numeral in the range [0; 2^{38}-1]. That is, the lowest-numbered serial number is 0, and the highest-numbered serial number is 274877906943. To be encoded in RAIN tags, the decimal value of the serial number has to be converted into an equivalent 38-bit long binary string (see section 5.7).

6.4.1. Example #1: sequential serialisation on a single production line

The method in this example is called “IT-based” because it relies upon the information systems of a tagging party to manage the allocation of serial numbers. The IT systems are used to keep track of which serial numbers have been allocated and which have not. The software performing this function can exist at a variety of levels within an enterprise’s IT architecture, from being embedded directly in a printer or other manufacturing device, to being a function of a Manufacturing Execution System (MES), to being a corporate-wide enterprise software function. Regardless, they all rely upon stored information that keeps track of which serial number to allocate next, and so it is important that these systems are properly secured and backed up.

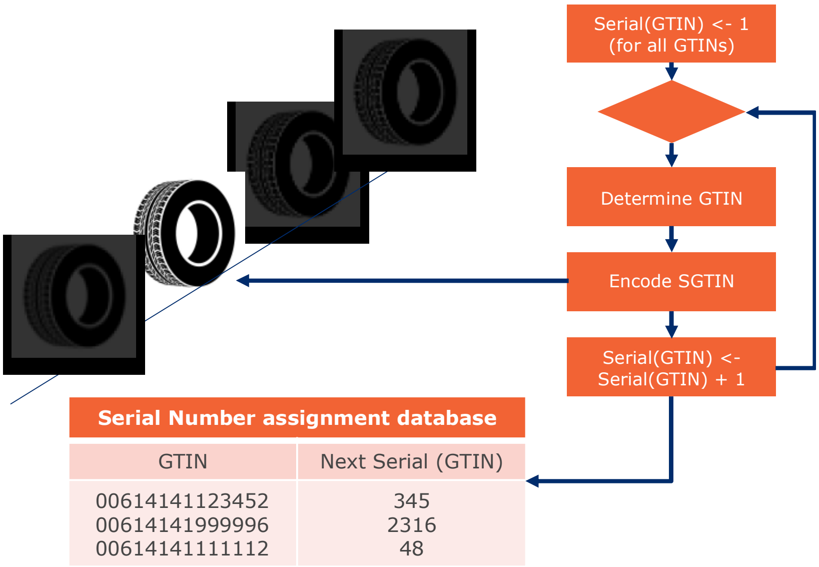
The basic building block for an IT-based serialisation method is the allocation of serial numbers on a single manufacturing or tagging line. The simplest method is to have a counter which allocates serial numbers sequentially one at a time, as illustrated below:



In this example, the first instance of the tire receives serial number 1, the second receives serial number 2, and so on. If the full 38-bit serial number available in the SGTIN-96 RFID tag is consumed in this way, there is capacity to serialize 2^{38} = 274,877,906,944 instances of a tire (GTIN) without duplications.

The software responsible for sequential serialisation need only keep track of a single number; namely, the next available serial number in the sequence or the last number that has been allocated. This is critical information to ensure that serial numbers are not duplicated.

The serial number only must be unique within a given GTIN, and so when there are multiple types of tire (multiple GTINs) there is a "next number" for each GTIN. When a single software system is responsible for serialising multiple GTINs, the implementation looks something like the following.



Serial Number assignment database

GTIN	Next Serial (GTIN)
00614141123452	345
00614141999996	2316
00614141111112	48

The serial number assignment database keeps track of the next available serial number for each GTIN being manufactured. As in the previous case, it is imperative that this information is carefully backed up.

6.4.2. Example #2: static allocation of serial number ranges to multiple production lines

When there are two or more manufacturing lines tagging the same type of tire (same GTIN), they may each use sequential serialisation as described above, but additional care must be taken to ensure that those lines do not issue the same serial number. In the examples that follow, "Line #1", "Line #2", and "Line #3" will be used generically to illustrate three different manufacturing lines located in either the same or separated factories.

A straightforward way to avoid duplication is to give each manufacturing line a separate set of serial numbers to use for each different GTIN. In this example of a "static allocation" approach, the entire range of possible serial numbers for a GTIN is divided into large blocks, each block assigned to a manufacturing line, and each manufacturing line allocates serial numbers for that GTIN within its specified block. There are many ways that dividing the serial number range into blocks can be done. In this example we propose to make use of contiguous ranges in decimal. This is described in Table 11.

Table 11 Static allocation of serial number ranges to multiple production lines

Manufacturing Line	Minimum Serial Number	Maximum Serial Number
Line #1	0	19999999

Manufacturing Line	Minimum Serial Number	Maximum Serial Number
Line #2	20000000	39999999
Line #3	40000000	59999999

In this example, each manufacturing line is given a range of 20 million serial numbers to use. For example, Manufacturing Line #2 is free to assign any serial number, provided the number is greater than or equal to 20000000 and less than 40000000. If Manufacturing Line #2 is using sequential serialisation, it simply initializes its counter to 20000000. It should also check to make sure the upper limit is not exceeded, though normally in static allocation the size of the range is more than the number of tires that could be possibly manufactured on any given line.

This scheme is simple to understand, gives flexibility to add more manufacturing lines (because the entire serial number range has not been fully allocated), and the human-readable form of the serial number makes it easy to recognize which range was used.

There are many other ways to divide the entire range of serial numbers into different sets for different manufacturing lines. Tire manufacturers can design to suit their manufacturing process.

Since static allocation requires careful forward planning in devising a suitable plan, companies should consider other ways that they can dynamically allocate serial numbers to multiple production lines.

6.4.3. Cautionary notes

Even if the two previous serialisation management processes are easy to understand, it is not unusual to have implementation issues. This is often due to a poor or missing top-level serialisation plan (see section 6.2).

Here are some examples of possible issues.

Mistake #1

A tire manufacturer decides to implement a sequential serialisation plan for a production line that manufactures a type of tire. To address an increase in demand for such tires, the manufacturer decides to set up a new production line that will use the same serialisation method without consideration that the first one already allocates serial numbers. Therefore, all the serial numbers will be duplicated.

Solution: never implement a new serialisation plan without understanding existing ones.

Mistake #2

A tire manufacturer decides to implement a sequential serialisation plan for four existing production lines that manufacture a specific type of tire. It is decided to follow the example given in section 6.4.2 and divide the full serial number range into four equal sub-ranges as described in Table 12.

Table 12 Example of static allocation of serial number ranges to multiple production lines

Manufacturing Line	Minimum Serial Number	Maximum Serial Number
Line #1	0	68 719 476 735
Line #2	68 719 476 736	137 433 953 471
Line #3	137 433 953 472	206 158 430 207
Line #4	206 158 430 208	274 877 906 943

This allocation does not consider the capacity of each individual line. It assumes each line will manufacture the same number of tires. Furthermore, in case a fifth production line is implemented, the entire serialisation plan will need updating.

Solution: since the total number of serial numbers is far above the number of manufactured tires, the serial number range that is allocated to a production line shall be defined according to the expected number of tires that will be produced. A range of serial numbers shall also be reserved for future use.

Mistake #3

A tire manufacturer, who manufactures different tire types on multiple production lines, decides to make use of a simple sequential serialisation plan starting with serial number 0 and incrementing this serial number by 1 whatever the tire type (i.e., whatever the GTIN) and whatever the production line. Therefore, the full range of available serial numbers is divided by the number of tire types. Furthermore, this requires a very robust central server / database that allows all production lines to connect in real time and ensures serial numbers are allocated and recorded to prevent duplication.

Solution: it is recommended to allocate either a static or dynamic range to each production line to reduce the dependency to one single server.

6.5. A few words about TID-based serialisation

The two serialisation examples presented in Section 6.4 are based on the brand owner controlling its own serial number assignment through information systems it deploys. An alternative approach makes use of an RFID tag hardware feature called the Tag Identifier (TID). Because this method relies upon a hardware feature of the RFID tag, it is called "chip-based" serialisation.

The TID is a special identifier encoded in the dedicated memory bank of the RFID chip that holds information about the RFID tag itself, as opposed to information about the object to which the tag is affixed. All RFID tags include information in the TID that identifies the maker/designer of the chip and its product type.

Many RFID tags also include additional information in the TID. One of these additional pieces of information is a serial number that is assigned by the manufacturer of the RFID chip – unique among all RFID chips of the same maker/designer and model. This serial number is referred to as the TID serial number.

The TID serial number is different from the serial number that is part of the SGTIN. The SGTIN consists of a GTIN that identifies the type of tire and a serial number assigned by the brand owner to identify a specific instance of that tire. The serial number in the SGTIN is unique within a given GTIN. The TID serial number, in contrast, is assigned by the RFID chip manufacturer before the chip is affixed to a product (indeed, before the chip is even made into an RFID tag). The chip manufacturer has no idea to what product the chip will eventually be affixed, and so the TID serial number has nothing to do with the SGTIN. The chip manufacturer simply changes the serial number for each chip it makes. If the chip maker/designer and model information from the TID is combined with the TID serial number, the result is a number that is different for every RFID tag manufactured by anybody.

Deriving all 38 bits of the SGTIN serial number from the TID presents difficulties especially when different chip manufacturers are involved. We therefore encourage the tire manufacturers who want to leverage from TID to derive the SGTIN serial numbers to clearly define the serialisation process and to discuss with the chip manufacturers about their own TID serialisation process.

7. Different EPC syntaxes

7.1. Introduction

A GS1 identification key such as an SGTIN may exist in many different representations, some specific to a data carrier, some independent of data carrier.

In the GS1 system, a clear distinction is made between data and a data carrier. GS1 standards provide normative definitions of various data elements and these definitions hold true regardless of the form that data takes or how data is transmitted. A "data carrier" as used in GS1 standards is a means of physically affixing data to a physical object so that the data can be captured without human data entry. In the GS1 system, data carriers include a variety of types of barcodes and radio frequency identification (RFID) tags.

A principle of GS1 standards is that data elements are defined in a data carrier neutral way so that their semantics is the same regardless of what data carrier is used to affix them to a physical object (similarly when outside of a physical data carrier, such as in an electronic message). However, there are many different data carriers, each optimised for a particular set of physical and performance constraints arising in the real world. Standards for each data carrier therefore define a carrier-specific representation of carrier-neutral data elements, allowing the data to be encoded in a manner compatible with the physical constraints of the carrier.

7.2. Identifier Syntaxes: Plain, GS1 element string, EPC URI, GS1 Digital Link

When a GS1 identification key or other identifier is used in an information system, it is represented using a specific syntax. The syntax that is used may depend on the medium in which the identifier exists; for example, an XML message is text-oriented, while the memory of an RFID tag is binary-oriented.

GS1 standards provide four different syntaxes for identifiers that support progressively broader application contexts:

- **Plain:** This syntax is just the GS1 identification key with no additional characters or syntactic features.
- **GS1 element string:** This syntax consists of a short (2-4 character) "application identifier" that indicates what type of GS1 identification key follows, followed by the key itself. This allows one type of GS1 identification key to be distinguished from another. Related to the GS1 element string is the "concatenated element string", in which two or more AI-value pairs are concatenated into a single string (with delimiters, if needed). This provides a syntax for SGTIN which is defined as the concatenation of a GTIN and a Serial Number.
- **Electronic Product Code (EPC) URI:** This syntax is a Uniform Resource Identifier (URI), specifically a Uniform Resource Name (URN) beginning with urn:epc:id:... and the remainder having a syntax defined by the GS1 EPC Tag Data Standard. This provides a syntax for any key that identifies a specific physical or digital object.
- **GS1 Digital Link URI:** The GS1 Digital Link URI provides a syntax for expressing GS1 Identification Keys, Key qualifiers, and data attributes in a format that can be used on the Web in an intuitive manner (via a straightforward Web request) to enable direct access to relevant information and services about products, assets, locations, etc.

While any given GS1 identification key may be represented in more than one of the above four syntaxes, its meaning is always the same regardless of syntax. The Table 13 illustrates a GS1 SGTIN in each of the four syntaxes:

Table 13 Different SGTIN syntaxes

Representation	Example	Comment
Plain	09528765000079 6789	The GTIN and serial number are here shown separately
GS1 element string	(01)09528765000079(21)6789	The Application Identifier (01) precedes the 14-digit GTIN, and (21) precedes the serial number[1]
EPC URI	urn:epc:id:sgtin:9528765.000007.6789	In the Pure Identity EPC URI, the digits of the GTIN comprising the GS1 Company Prefix are separated from the others, the indicator digit or leading zero is moved to immediately precede the Item Reference, and the check digit is omitted. See section Annex A.

[1] Application Identifiers are defined in the GS1 General Specifications

Representation	Example	Comment
Digital Link	https://id.gs1.org/01/09528765000079/21/6789	GS1 Digital Link URI on the id.gs1.org domain

7.3. EPC syntaxes

The Electronic Product Code (EPC) was the creation of the MIT Auto-ID Center, a consortium of over 120 global corporations and university labs. EPC identifiers were designed to uniquely identify each item manufactured, as opposed to just the manufacturer and class of products. After being managed by EPCglobal, Inc., a subsidiary of GS1, EPCs are now directly managed by GS1 Global Office. The specifications for the EPC identifiers can be found in the GS1 Tag Data Standard, which is an open standard, freely available for anyone to download. *https://www.gs1.org/standards/epc-rfid/tds*

The canonical representation of an EPC is a URI, namely the "pure-identity URI" representation that is intended for use when referring to a specific physical object in communications about EPCs among information systems and business application software.

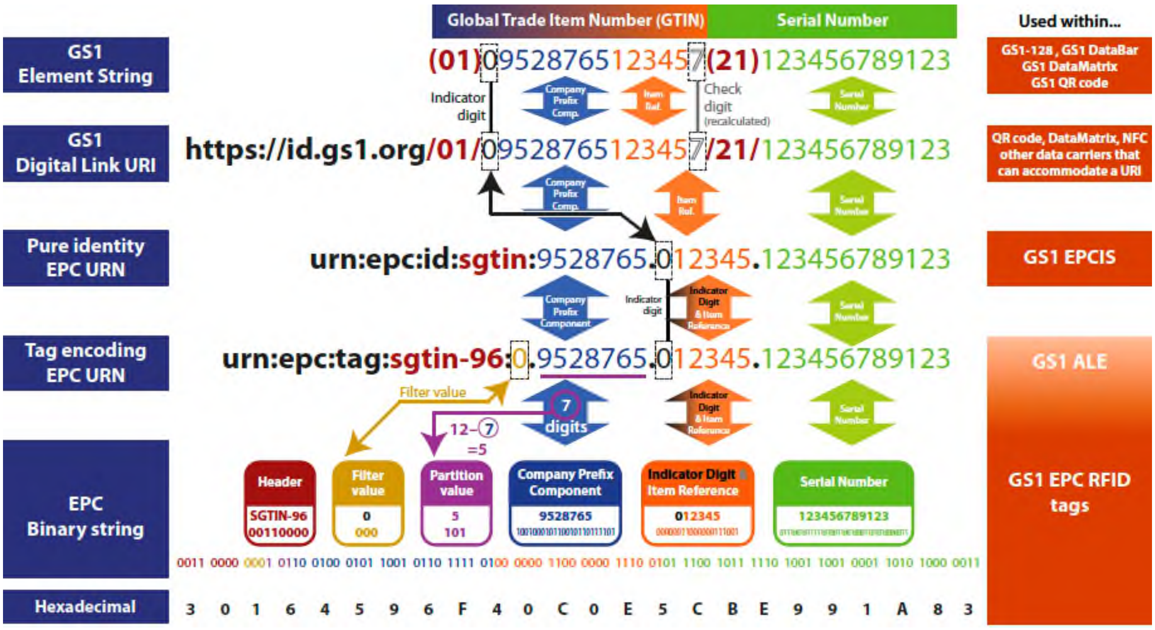
The GS1 Tag Data Standard also defines additional representations of an EPC identifier, such as the tag-encoding URI format and a compact binary format suitable for storing an EPC identifier efficiently within RFID tags (for which the low-cost passive RFID tags typically have limited memory capacity available for the EPC/UII memory bank). The GS1 Tag Data Standard also defines the encoding and decoding rules to allow conversion between these representations.

The Table 14 gives an example of EPC syntaxes of a given SGTIN, starting from a GS1 element string.

Table 14 Different EPC syntaxes

Representation	Example	Comment
GS1 element string	(01)09528765000079(21)569	The GTIN shall always be represented as a 14-digit long decimal string. GS1 defines the Serial Number (AI 21) as a 20 alphanumeric string. When using SGTIN-96 EPC format, the Serial Number shall only be numeric with a value between 0 and 274,877,906,943
EPC Pure Identity URI	urn:epc:id:sgtin:9528765.000007.569	In the Pure Identity EPC URI, the digits of the GTIN comprising the GS1 Company Prefix are separated from the others, the indicator digit or leading zero is moved to immediately precede the Item Reference, and the check digit is omitted. See Section Annex A. To build this Pure Identity URI, the length of the GS1 GCP shall be known. In this example, the GCP length is 7 digits
EPC Tag URI	urn:epc:tag:sgtin-96:0.9528765.000007.569	To create an EPC Tag URI, additional information shall be known as the length of the binary string to encode in the RAIN tag (here 96-bit) and the filter value (here 0).
EPC binary (hexadecimal)	30164596F400001C000000239	This represent the 96-bit long binary string that must be encoded into the EPC memory bank of the RAIN tag. Since EPCs are not the only unique identifiers that can be encoded in RAIN tags, additional information must be encoded in the Protocol Control (PC) word of the EPC memory of the RAIN tag.

The figure below shows the different translation steps between GS1 element string, Digital Link and EPCs representations.



Global Trade Item Number (GTIN)
Serial Number
Used within...
GS1 Element String
(01)09528765123457(21)123456789123
GS1-128 , GS1 DataBar GS1 DataMatrix GS1 QR code
Indicator digit
Company Prefix Comp.
Item Ref.
Check digit (recalculated)
Serial Number
GS1 Digital Link URI
https://id.gs1.org/01/09528765123457/21/123456789123
QR code, DataMatrix, NFC other data carriers that can accommodate a URI
Pure identity EPC URN
urn:epc:id:sgtin:9528765.012345.123456789123
GS1 EPCIS
Company Prefix Component
Indicator digit
Indicator Digit & Item Reference
Tag encoding EPC URN
urn:epc:tag:sgtin-96:0.9528765.012345.123456789123
GS1 ALE
Filter value
7 digits
12–7 =5
EPC Binary string
Header SGTIN-96 00110000
Filter value 0 000
Partition value 5 101
Company Prefix Component 9528765 100100010110010110111101
Indicator Digit & Item Reference 012345 00000011000000111001
Serial Number 123456789123
GS1 EPC RFID tags
0011 0000 0001 0110 0100 0101 1001 0110 1111 0100 0000 1100 0000 1110 0101 1100 1011 1110 1001 1001 0001 1010 1000 0011
Hexadecimal
3 0 1 6 4 5 9 6 F 4 0 C 0 E 5 C B E 9 9 1 A 8 3

8. Encoding a RAIN tag

Once unique SGTIN-96 identifiers have been created and translated into 96-bit long binary strings, they must be encoded in the RAIN tire tags. Due to air interface protocol considerations and memory organization of RAIN tags, some additional information must be encoded. Furthermore, for data security purposes, or when a stakeholder would like to make use of the optional User Memory bank, a few precautions must be taken.

8.1. Structure / content of a RAIN tag

This document is based on the use of the GS1 Gen2 (or ISO/IEC 18000-63) air interface protocol that specifies physical data transfer between RAIN RFID tags and readers with commands for controlling the readers relative to the tags. This air interface protocol covers mandatory and optional functionalities which allow:

- Filtering tags
- Implementing high speed inventories
- Accessing data on tags (read/write commands)
- Protecting encoded data
- Protecting privacy
- Authenticating tags

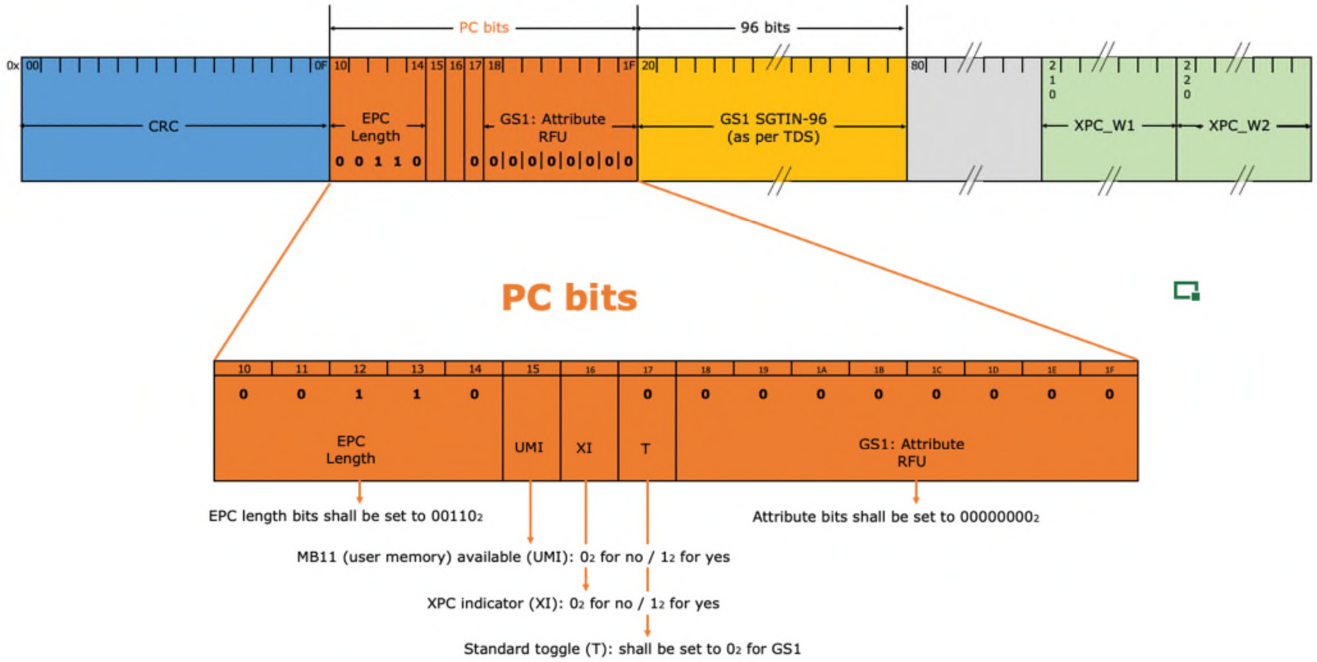
To achieve these goals, the overall memory of RFID chips has been segmented into 4 different memory banks that are described in the figure below.

8.2. EPC/UII memory bank

The EPC Memory Bank (01) contains the binary string representing the SGTIN-96 together with two other required words (a word is a 16-bit long binary string); the Stored-CRC and the StoredPC.

There are also two optional words, the extended protocol control (XPC) word1 and word2. When the tag implements one or both XPC words, these XPC word(s) are stored at address beginning at 210h.

This is summarised in the figure below.


PC bits
96 bits
0x 00
0F
10
14
15
16
17
18
1F
20
80
210
220
CRC
EPC Length
0 0 1 1 0
GS1: Attribute RFU
0 0 0 0 0 0 0 0 0
GS1 SGTIN-96 (as per TDS)
XPC_W1
XPC_W2
PC bits
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
0 0 1 1 0
0
0 0 0 0 0 0 0 0
EPC Length
UMI
XI
T
GS1: Attribute RFU
EPC length bits shall be set to 00110₂
Attribute bits shall be set to 00000000₂
MB11 (user memory) available (UMI): 0₂ for no / 1₂ for yes
XPC indicator (XI): 0₂ for no / 1₂ for yes
Standard toggle (T): shall be set to 0₂ for GS1

The Tag shall implement the StoredCRC by first calculating a CRC-16 over the StoredPC and the EPC specified by the length (L) bits in the StoredPC, and then storing the thus-computed StoredCRC into EPC memory 00h to 0Fh. The Tag shall calculate the StoredCRC on word boundaries, shall deassert all Tag-computed StoredPC bit values (XI and UMI if Tag-computed) when performing the calculation, and shall omit XPC_W1 and XPC_W2 from the calculation.

If an Interrogator attempts to write to EPC memory 00h – 0Fh then the Tag shall not execute the write and instead treat the command's parameters as unsupported.

The StoredPC bits and values shall be as follows:

L (EPC length field, bits 10h – 14h): Bits 10h – 14h are written by an Interrogator and specify the length of the EPC that a Tag backscatters in response to an ACK, in words. When encoding an SGTIN-96, the value of the L bits shall be 00110_2.

UMI (User-memory indicator, bit 15h): Bit 15h may be fixed by the Tag manufacturer or computed by the Tag depending on the status of the User Memory bank.

XI (XPC_W1 indicator, bit 16h): If a Tag does not implement XPC_W1 then bit 16h shall be fixed at 0_2 by the Tag manufacturer. If a Tag implements XPC_W1 then a Tag shall compute XI both at powerup and upon changing any bits of XPC_W1.

T (numbering system identifier toggle, bit 17h): When bit 17h is 0_2 then the application is referred to as a GS1 EPCglobal Application and PC bits 18h – 1Fh shall be as defined in this protocol. When bit 17h is 1_2 then the application is referred to as a non-GS1 EPCglobal Application and bits 18h – 1Fh shall be as defined in ISO/IEC 15961.

As per ISO 20910, SGTIN-96, which is a GS1 EPCglobal encoding scheme, is the format of unique identifier for tires. Therefore, T shall be set to 0_2.

Attribute bits RFU or AFI (Reserved for Future Use or Application Family Identifier, bits 18h – 1Fh): Since T=0_2 then the Tag manufacturer (if the bits are not writeable) or an Interrogator (if the bits are writeable) shall set these bits to 00h.

8.2.1. Encoding the SGTIN-96

The binary format of the SGTIN-96, as defined in section 5.2, shall be encoded in the EPC memory bank starting at address 20h.

The L bits shall be set to 00110_2 to indicate that the EPC length is 6 words long (96 bits)

The T bit shall be set to 0_2 to indicate that GS1 numbering system is used.

The attribute bits shall be set to 00h as per GS1 Gen2 Air interface protocol.

All other bits of the EPC memory bank are either encoded by the chip manufacturer or computed by the RFID chip upon conditions that are described in GS1 Gen2 Air interface protocol.

8.2.2. Permalock and tag commissioning

Write-protecting the EPC memory bank of RAIN RFID Tags

An RFID chip implements a set of Lock Action bits for every memory bank. Each memory bank's Lock status is described by two bits. One for the pwd-write (or pwd-read/write) and a second one for permalock.

Take the example of the EPC memory bank (MB01). If locked (using a non-zero access password) or if permalocked, you can read it, but you cannot write it. Table 15 summarizes the meaning of the two lock status bits.

Table 15 Lock status bits of a RAIN tag

	Pwd-write=0_2	Pwd-write=1_2
Permalock=0_2	You can read and write the EPC memory and you can change the protection status. This is not recommended.	You can read the EPC memory but to write it you must know the access password. This means that the EPC memory is password write protected. You can also change the protection status. This is the preferred way to reversibly lock the EPC memory.
Permalock=1_2	Under this condition the EPC memory will be always writeable (and readable), and this status cannot ever be changed. There is no way to write-protect the EPC memory.	You can read the EPC memory, but you can never write it whatever the state. You will never be able to change the protection status. This is the preferred way to permalock the EPC memory.

As per ISO 20910, once the EPC memory has been encoded and the content verified, it shall be permalocked to be protected from any over-writing.

Disabling Kill function

The Kill command allows an Interrogator to permanently disable a Tag. To kill a Tag, an Interrogator shall follow a dedicated kill procedure for which the Kill password (32 bits stored in the Reserved memory bank) must be a non-zero value.

As per ISO 20910, the kill function shall be disabled before the tire leaves the tire manufacturer facilities. Therefore, the Kill password must be set to 00000000h and permalocked.

> ***NOTE:*** *Permalocking the Kill password makes it unreadable and unwritable forever.*

Access password

The Access password is another 32-bit password stored in the Reserved Memory bank. The Access password allows to transition a tag from Open to Secured state.

For example, if you want to reversibly write-protect the content of the User memory, you can make use of the Lock command with Password-write bit set to 1_2 and the Permalock bit set to 0_2. Therefore, those who know the Access password will be able to over-write the content of the User memory bank, others not.

Of course, the Access password itself can be read/write protected (either reversibly or permanently) using the corresponding Lock command. As per ISO 20910, the fact that the Access password is readable or not (whatever its value) means that there might be additional useful information about the tire encoded in the User memory bank and elsewhere in a database.

8.3. User memory bank

User memory bank allows user data storage. If a Tag implements User memory, it may partition the User memory into one or more files. If the Tag implements a single file, that file is File_0.

NOTE: *At the date of writing this document, there is no RFID chip supporting multiple files.*

In case a stakeholder would like to make use of this memory bank, we encourage them to comply with open standards. There are two standardized ways to encode additional data into the User memory bank:

- GS1 EPCglobal Application. In that case, the file encoding shall be as defined in the GS1 EPC Tag Data Standard.
- non-GS1 EPCglobal Application. In that case, the file encoding shall be as defined in ISO/IEC 15961 and 15962.

Any other proprietary encoding is highly discouraged.

8.4. TID and Reserved memory banks

TID Memory bank

TID memory locations 00h to 07h shall contain either an E0h or E2h ISO/IEC 15963 class-identifier value. The Tag manufacturer assigns the class identifier (E0h or E2h), for which ISO/IEC 15963 defines the registration authority. The class-identifier does not specify the Application. TID memory locations above 07h shall be defined according to the registration authority defined by this class-identifier value and shall contain, at a minimum, sufficient information for an Interrogator to uniquely identify the custom commands and/or optional features that a Tag supports. TID memory may also contain Tag- and manufacturer-specific data (for example, a Tag serial number).

If the class identifier is E0h, TID memory locations 08h to 0Fh contain an 8-bit manufacturer identifier, TID memory locations 10h to 3Fh contain a 48-bit Tag serial number (assigned by the Tag manufacturer), the composite 64-bit TID (i.e., TID memory 00h to 3Fh) is unique among all classes of Tags defined in ISO/IEC 15963, and TID memory is permalocked at the time of manufacture.

If the class identifier is E2h, the registration authority is GS1 and the TID memory above 07h is be configured as follows:

- 08h: XTID (X) indicator (whether a Tag implements an XTID)
- 09h: Security (S) indicator (whether a Tag supports the Authenticate and/or Challenge commands)
- 0Ah: File (F) indicator (whether a Tag supports the FileOpen command)
- 0Bh to 13h: A 9-bit Tag mask-designer identifier (MDID). The list of MDID can be downloaded here: *https://www.gs1.org/epcglobal/standards/mdid*
- 14h to 1Fh: A Tag-manufacturer-defined 12-bit Tag model number
- Above 1Fh: As defined in the GS1 EPC Tag Data Standard (see also section 6.5)

If the class identifier is E2h, TID memory locations 00h to 1Fh are permalocked at time of manufacture. If the Tag implements an XTID then the entire XTID is also permalocked at time of manufacture.

Reserved memory bank

Reserved memory contains the kill and/or access passwords if passwords are implemented on a Tag. If a Tag does not implement the kill and/or access password(s), the Tag shall logically operate as though it has zero-valued password(s) that are permanently read/write locked, and the corresponding physical memory locations in Reserved memory need not exist.

The kill password is a 32-bit value stored in Reserved memory 00h to 1Fh, MSB first. The default (unprogrammed) value is zero. A Tag that does not implement a kill password behaves as though it has a zero-valued kill password that is permanently read/write locked. A Tag shall not execute a password-based *Kill* command if its kill password is zero. An Interrogator may use a nonzero kill password in a password-based *Kill* command sequence to kill a Tag and render it nonresponsive thereafter (see section 8.2.2).

The access password is a 32-bit value stored in Reserved memory 20h to 3Fh, MSB first. The default (unprogrammed) value is zero. A Tag that does not implement an access password behaves as though it has a zero-valued access password that is permanently read/write locked. A Tag with a zero-valued access password transitions from the **acknowledged** state to the **secured** state upon commencing access, without first entering the **open** state. A Tag with a nonzero-valued access password transitions from the **acknowledged** state to the **open** state upon commencing access; an Interrogator may then use the access password in an *Access* command sequence to transition the Tag from the **open** to the **secured** state.

9. Glossary

Term	Definition
38-bit serial number	A numerical serial number encoded as a 38-bit long binary string; minimum value is à and maximum value is 274 877 906 943
8-bit	An 8-bit long binary string also defined as a byte
access password	A 32-bit long password encoded in the RESERVED memory bank of a RAIN tag to enable transition from open to secured state
acknowledged state	State of the RAIN tag after receiving an ACK command with correct RN16 from the interrogator
AES-128	Advanced Encryption Standard is a symmetric encryption algorithm having a 128-bit long cryptographic key
AFI	Application Family Identifier is a 8-bit long binary string to serve as a method for classifying radio frequency identification by application, enabling a single air interface protocol to be used across several applications
air interface protocol	The rules that govern how tags and readers communicate
allocation class identifier	An 8-bit long binary string defined in ISO/IEC 15963 to identify the data structure of unique RFID chip identifier
allocation table	A repository of serial number values (or ranges of values) that are allocated to a dedicated user/encoder
API	Application Programming Interface is a computing interface that defines interactions between multiple software intermediaries
application identifier	Application Identifier (AI) is a code defined by GS1 to uniquely identify different type of data like GS1 identifiers or general data like a date or a location
AuthComm	AuthComm is an optional Gen2v2 air interface protocol command that allows authenticated communications from Reader to Tag by encapsulating another command and typically also a MAC in the AuthComm's message field
backscatter	Backscatter (or backscattering) is the reflection of waves back to the direction from which they came. It isthe way by which passive tags can respond to reader's commands
bits	binary digit represents a logical state with two possible values. These values are represented as either "1"or"0"
CCC code	China Compulsory Certification

Term	Definition
CD	See Check Digit
Check Digit	A check digit is a form of redundancy check used for error detection on identification numbers
chip-based serialization	A check digit is a form of redundancy check used for error detection on identification numbers
class identifier	See allocation class identifier
command-response	Sequence of command (reader) and response (tag) defined in the state-machine based air interface protocol
company prefix	See GS1 Company Prefix
Contiguous Ranges in Decimal	Ranges of decimal values that are contiguous (example: [10-15] and [16-50])
CRC-16	A 16-bit long cyclic redundancy check (CRC) is an error-detecting code commonly used in digital networks and storage devices to detect accidental changes to raw data
data syntax	Data syntax defines the structure or order of the different elements (or data fields). Semantics is the meaning of these elements.
Downstream Exception Serialization	Process that consists of enabling downstream parties to allocate serial number for tire tagging on behalf of the GCP owner
downstream parties	Downstream parties are companies that operate at the demand side of the supply chain toward the final assembly unit, retailer and consumer
Dynamic Allocation	A serialization process that consists of allocating serial numbers on demand.
EAN	The European Article Number (also known as International Article Number or EAN) is a standard describing a barcode symbology and numbering system used in global trade to identify a specific retail product type, in a specific packaging configuration, from a specific manufacturer.
EDI	Electronic Data Interchange: an interactive electronic system that enables a supplier and a customer to communicate easily
Encode	Action of writing binary data at a specific and dedicated memory address of a microelectronic chip
EPC Binary	String of bits that represents all or part of the EPC
EPC memory	See EPC memory bank
EPC Memory Bank	Dedicated memory addresses of an RFID chip in which the EPC (and other information) is encoded
EPC syntaxes	Different ways to represent EPC to be used or encoded in data carriers
EPC Tag URI	EPC syntax for RFID tag encoding
EPC URI	EPC Universal Resource Identifier is a general EPC syntax as defined by W3C
exception tagging	Operation that consists of enabling the tagging of products by third parties on behalf of the manufacturer of the product
Filter value	A binary string that defines the general purpose of the tagged product
GCP	See GS1 Company Prefix
Gen2V2	Nickname of the EPC UHF Gen2 Air Interface Protocol version 2
Grain-128A	Grain 128a is stream cipher that allows symmetric key encryption based on 128-bit long keys

Term	Definition
GS1	Global Standards 1 (GS1) is an international standards organization with member bodies in more than 110 countries worldwide
GS1 company prefix	A numeric code allocated by GS1 to their members to allow them creating GS1 identifiers
GS1 DataBar	A GS1 standardized symbology that consists of aggregation of multiple linear barcodes
GS1 DataMatrix barcode	A GS1 standardized symbology that allows encoding of GS1 identifiers, application identifiers and data in a 2D-barcode
GS1 Digital Link URI	A GS1 standardized URI syntax for representation of GS1 identifiers and application identifiers
GS1 element string	A GS1 standardized representation of GS1 identifiers and application identifiers as a simple string of data
GS1 EPC Gen2 V2.0.1	See Gen2v2
GS1 Identification Key	A unique, non-significant number, standardized by GS1, that can be used internationally for identification throughout the supply chain
GS1 Tag Data Standard	GS1 standard that defines the way GS1 identification keys, application identifiers, and data can be encoded in RFID tags
GS1 Tag Data Standard coding schemes	Codification system defined in GS1 Tag Data Standard
GS1-128	A GS1 standardized symbology that allows encoding of GS1 identifiers, application identifiers, and data in a Code-128 barcode
GTIN	Global Trade Item Number is a GS1 identification key
Header	An 8-bit long binary string that identify the EPC scheme encoded in an RFID tag
hexadecimal	A base 16 representation of numbers
Identifier Syntaxes	Different ways to represent GS1 identifiers to be used or encoded in data carriers
Indicator	Number that indicates the packaging level of an item identified by a GTIN
Interrogator	See RFID interrogator
Item Reference Number	Number that is allocated by the manufacturer of a product to identify it using a GTIN
ITF-14	ITF-14 is the GS1 implementation of an Interleaved 2 of 5 (ITF) barcode to encode a GTIN. ITF-14 symbols are generally used on packaging levels of a product, such as a case box of 24 cans of soup. The ITF-14 will always encode 14 digits
Kill command	Command defined is Gen2v2 air interface and which allows to permanently deactivate an RFID tag
L and U bits	Specific bits of PC and XPC_W1 words defined in Gen2v2 air interface protocol
L bits	Specific bits of PC word defined in Gen2v2 air interface protocol. L bits give information about the length of the encoded EPC
LGTIN EPC	A compound GS1 identifier formed from the combination of GTIN and Batch/Lot identifier.
line code	A code allocated to a specific manufacturing or production line to manage serial numbers

Term	Definition
Low-level RFID	Software to manage all parameters of Gen2v2 commands
MB01	Memory Bank 01, see EPC Memory Bank
MB11	Memory Bank 11, also known as User memory bank, that serves for addition data encoding in RFID tags
MDID	Mask Designer Identifier is a unique number assigned by GS1 to RFID chip manufacturers so they can create Unique Tag Identifiers
memory bank	Dedicated memory addresses of an RFID chip in which the different data must be encoded
MES	Manufacturing Execution System are computerized systems used in manufacturing to track and document the transformation of raw materials to finished products
MSB	Most Significant Bit is the bit in a multiple-bit binary number with the largest value
non- volatile memory	Non-volatile memory (NVM) or non-volatile storage is a type of memory that can retain stored information even after power is removed
open state	State of the RAIN tag after being inventoried when Access Password is non-zero
Partition Value	Number that indicates the length of the Company Prefix in SGTIN EPC scheme
passive ultra high frequency	Type of RFID technology that makes use of backscattering in the 860-960 MHz frequency band
permalock	Form of the Gen2v2 Lock command that permanently locks an entire memory bank
Plain	Syntax that simply concatenates GS1 identification key and other optional data without separators
pure-identity URI	EPC syntax of GS1 identifiers, application identifiers, and data in a carrier agnostic format
pwd-read/write	State of memory bank which is protected by a password against reading and writing
pwd-write	State of memory bank which is protected by a password against writing
RAIN RFID Tag	Brand name for a passive UHF tag that follows the ISO/IEC 18000-63 or GS1 UHF Gen 2 standards
reader	See RFID interrogator
reading distance	Distance between the antenna of a reader and a tag at which the tag can be read
Reserved Memory bank	Dedicated memory addresses of an RFID chip in which the Kill and Access passwords are stored
RFID	Radio Frequency Identification
RFID chips	Microelectronic device that supports Gen2v2 air interface protocol
RFID interrogator	Electronic devices that can energize and communicate with RFID tags
RFU	Reserved for Future Use
SecureComm	Gen2v2 protocol command that allows encryption of communications from reader to tags by encapsulating another, encrypted command in the SecureComm's message field

Term	Definition
secured state	State of the RAIN tag after being inventoried when Access Password is zero or when it has been sent by the interrogator to the tag
Security (S) indicator	TID memory bank bit that indicates whether a Tag supports the optional Gen2v2 Authenticate and/or Challenge commands
Select command	Gen2v2 protocol command that can be broadcast to tags to filter out those that will not be part of the inventory round
SGTIN	Serialized Global Trade Item Number
SGTIN EPC	EPC format of the SGTIN
SGTIN-96 format	EPC format of the SGTIN that is encoded in a 96-bit long binary string
state-machine	A device which can be in one of a set number of stable conditions depending on its previous condition and on the present values of its inputs.
Static Allocation	Serialization process based on fixed allocation of ranges of serial numbers to a given production line
Stored-CRC	A 16-bit long cyclic redundancy check (CRC) that is computed by the RFID chip on the entire EPC content and stored in the EPC memory bank
StoredPC	Gen2v2 air interface protocol control word that contains additional information about information encoded in the EPC memory
T bit	Toggle bit that defined the numbering system in use in the RFID tag (GS1 or ISO)
Tag Data Standard	See GS1 Tag Data Standard
Tag mask-designer identifier (MDID)	Identifier allocated by GS1 to RFID chip manufacturers to create unique TID
TDS	See GS1 Tag Data Standard
TID	Tag Identifier that uniquely identifies an RFID chip
TID Memory bank	Dedicated memory addresses of an RFID chip in which the TID must be encoded
trade item	Product that is intended to be sold
U.P.C. Company Prefix	A numeric code allocated by GS1 US to their members to allow them creating GS1 identifiers
UII	Unique Item Identifier
UMI	User memory Indicator
UPC-A	Universal Product Code type A
UPC-E	Universal Product Code type E
URI	Uniform Resource Identifier as defined by W3C
user memory bank	Dedicated memory addresses of an RFID chip in which the additional information must be encoded
XI	XPC_W1 indicator
XPC_W1	Gen2v2 air interface protocol control word that contains additional information about the features that the RFID tag supports or not
XPC_W2	Protocol Control word as defined in Gen2v2 air interface protocol that contains additional information about sensor measurements
XTID	Extended TID

Annex A. GS1 Global Trade Item Number (GTIN)

A.1. What is a Global Trade Item Number?

One of the main building blocks of the GS1 System, the Global Trade Item Number (GTIN), is the globally unique GS1 Identification Key used to identify "trade items". It should be noted that GTINs do not uniquely identify instances of a trade item. Serialized GTINs (SGTINs) are used to uniquely identify items within a GTIN. SGTINs are explained and discussed in detail in previous sections (see section 5.2).

What is a trade item? A Trade Item is any product or service that may be priced, ordered, or invoiced at any point in the supply chain. The term trade item is applicable to individual items as well as all other packaging configurations offered for sale (e.g., two-pack; case; pallet; etc.). Each different trade item must be identified by a different GTIN. For example, a tire manufacturer which has been assigned a Company Prefix, would use different GTINs to identify the different types of tires (see section 5.3.2).

GTINs are generated from a Company Prefix (GS1 Company Prefix (GCP), assigned by GS1 to a company or U.P.C. Company Prefix, assigned by GS1 US only) and a unique number per type of trade item (not per each instance of a trade item, which is accomplished by using an SGTIN (see section 5)). This unique number per type of trade item is assigned by the company.

A.2. GTIN Structure

Each GTIN is a numerical string comprised of distinct segments. GTIN segments include:

Indicator Digit: this digit is only used for GTIN-14. Values between 1 and 8 indicates a packaging level and the value 9 is used for variable measure item.

GS1 Company Prefix (GCP): A globally unique number issued to a company by a GS1 Member Organization to serve as the foundation for generating GS1 identifiers (e.g., GTINs). GS1 Company Prefixes are assigned in varying lengths depending on the company's needs.

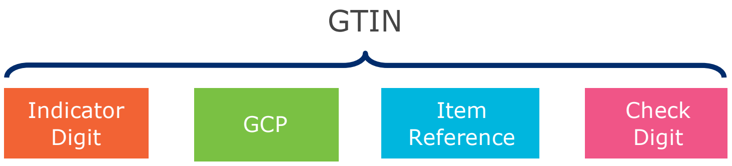
> ***NOTE****: to create and use a GTIN-12 a U.P.C. Company Prefix is required. The difference between a GS1 Company Prefix and a U.P.C. Company Prefix is that the GS1 Company Prefix has a zero added to the beginning of the number. Use the U.P.C. Company Prefix to assign GTINs and create U.P.C. barcodes for products that are sold at a store or online. Use the GS1 Company Prefix for all other GS1 identifiers*

Item Reference: A number assigned by the user to identify a trade item. The Item Reference varies in length as a function of the GS1 Company Prefix length (see section 5.3.2).

Check Digit: A one-digit number calculated from the preceding digits of the GTIN used to assure data integrity. GS1 provides a *check digit calculator* to automatically calculate Check Digits for you.

> ***NOTE:*** *The Global Trade Item Number (GTIN) Management Standard is designed to help industry make consistent decisions about the unique identification of trade items in open supply chains helping manufacturers and brand owners who are responsible for GTIN assignment to their products -* https://www.gs1.org/1/gtinrules//en/

The different GTIN components are summarized in the Figure below.



A.3. Where are GTINs used?

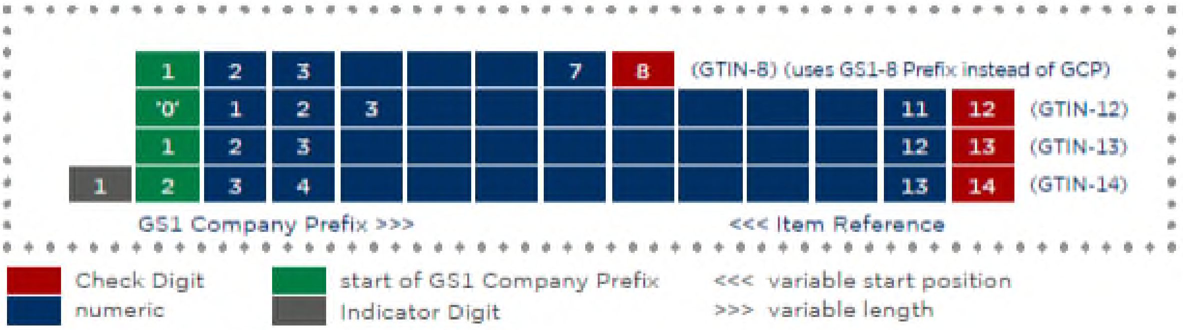
GTINs are used anywhere that a product or service needs to be identified -- including the Internet, business transactions, IT systems, and the physical product itself. GS1 Standards define how to format and structure the GTIN in a wide variety of applications so the same GTIN can be used to identify the product for any of those needs:

- **Physical products**: GTINs can be encoded into barcodes and Radio Frequency Identification (RFID) tags and affixed to products to facilitate accurate identification as products move through the supply chain.
- **E-commerce:** GTINs can be used in E-commerce business transactions (e.g., Purchase Order, Advance Ship Notice, Invoice, etc.) to support Order-to-Cash business processes. GTINs are an integral data element of most Electronic Data Interchange (EDI) transactions to reduce errors and support machine-to-machine processing.
- **Internet applications**: GTINs can be presented as a Uniform Resource Name (URN), a subset of Uniform Resource Identifier (URI), to support Internet-based applications and data sharing. The GS1 Electronic Product Code (EPC) standards include two URN data formats for GTIN. These EPC URNs provide a standardised data format for GTIN that is consistent with the standardised data format of the Internet to support Internet-based data sharing as well as "virtual" products (i.e., downloadable books and music) and "virtual" points of sale where there is no barcode reader and no EPC scanner.
    - **LGTIN EPC** (a Uniform Resource Name (URN) for GTIN plus batch/lot number)
    - **SGTIN EPC** (a Uniform Resource Name (URN) for GTIN plus serial number)

  NOTE: *as per ISO 29010, SGTIN EPC URI is the only format that will be described further in this document. Tire applications use SGTIN EPC and not LGTIN so LGTIN will not be described further in this document.*
- **Databases & IT systems:** The GTIN provides a single product identifier that can be used in all systems (e.g., purchasing, inventory management, logistics, analytics, reporting, etc.). This maintains the connection between systems so they can be used collectively to enhance the quality and amount of data available to support operational processes as well as business intelligence and analytics.

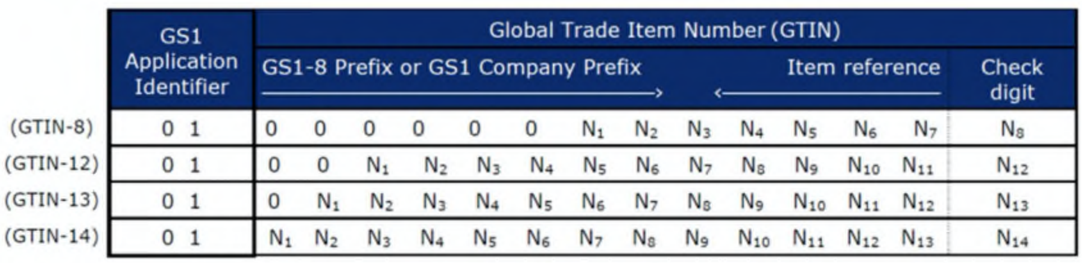
A.4. The different types of GTINs

There are four types of GTIN: GTIN-8, GTIN-12, GTIN-13, and GTIN-14. There are summarised in the Figure below.



GTIN-8 are mainly used for very small products for which the space is restricted. GTIN-14 are used when indicator digit is required for packaging level identification. Therefore, both GTIN-8 and GTIN-14 shall not be used for unique tire identification.

> ***NOTE:*** *Whatever the GTIN, GS1 recommends that GTINs are stored in a 14-digit reference field in a database, to accommodate all GTIN data structures. This can be done by adding leading zero(s) in front of the GTIN to left fill the 14-digit field. This is shown in the Figure below.*



	GS1 Application Identifier	Global Trade Item Number (GTIN)													
		GS1-8 Prefix or GS1 Company Prefix →								←	Item reference				Check digit
(GTIN-8)	0 1	0	0	0	0	0	0	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8
(GTIN-12)	0 1	0	0	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}	N_{11}	N_{12}
(GTIN-13)	0 1	0	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}	N_{11}	N_{12}	N_{13}
(GTIN-14)	0 1	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}	N_{11}	N_{12}	N_{13}	N_{14}

NOTE: *For tire applications and reference to ISO 20910, only GTIN-12 and GTIN-13 shall be considered.*

A.5. GTIN-12

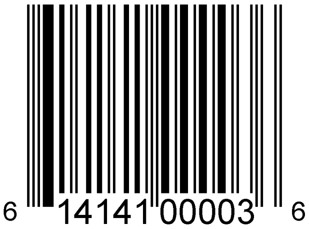
GTIN-12s are primarily used in UPC-A, UPC-E, GS1-128, GS1 DataBar, ITF-14, GS1 DataMatrix and GS1 QR code barcodes and in EPC/RFID. It requires the use of a U.P.C. Company Prefix, a specific subset of a GS1 Company Prefix.

A GTIN-12 is 12 digits in length and this, together with the U.P.C. Company Prefix length, will give you the total number of different GTINs you will be able to create.

Simply put the construction method is as follows:

U.P.C. Company Prefix + Item Reference Number + Check Digit = 12 digits

The figure below shows an example of a UPC-A barcode



A.6. GTIN-13

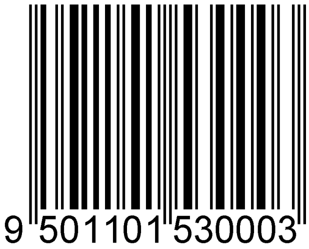
GTIN-13s are primarily used in EAN-13, GS1-128, GS1 DataBar, GS1 DataMatrix, ITF-14 barcodes, GS1 QR codes and in EPC/RFID.

A GTIN-13 is 13 digits in length, the length of the GCP will determine the number of GTIN-13s that can be created.

The GTIN-13 structure is as follows:

GS1 Company Prefix + Item Reference Number + Check Digit = 13 digits

The figure below shows an example of an EAN-13 barcode



Annex B. Advanced features of RAIN RFID tags

The Gen2 Air interface protocol is based on 3 “basic” set of commands:

- Select (allow to filter out unwanted tags)
- Inventory (allow to count tags that are in front of the reader)
- Access (allow to manage tags one by one)

The *Select* command allows the reader to filter the tags of interest from a mixed population. For example, you can decide to read only the GS1 encoded tags. To do that, you can issue a *Select* command on the value of the Toggle bit of the EPC memory. You can also decide to only inventory tags that are encoded with an SGTIN-96. In that case, you can issue a *Select* command on bits 20h to 28h that match the EPC Header value of 30h for SGTIN-96.

As more and more tags are used for different purposes and applications, the *Select* command becomes imperative because it allows you to keep the number of inventoried tags as low as possible.

The Inventory commands are made for counting the number of selected tags. During the inventory process, tag backscatters StoredPC word and EPC. Other information is accessed using *Read* command.

When you want to write data in tags or read data other than the one backscattered during the inventory process, you use the *Access* commands *Write* or *Read*. The *Lock* (reversible or permanent) command allows you to protect the content of the different memory banks. The *Kill* command permanently deactivates the targeted tag. This requires the use of the kill password that must be properly set to avoid any misuse of *Kill* command (see section 8.2.2).

Since the first release of version 2 of the Gen2 Air interface in 2015, additional optional commands have been standardised. The most important new commands deal with privacy, data security and authentication.

Untraceable

Untraceable allows an Interrogator with an asserted Untraceable privilege to instruct a Tag to:

- alter the L and U bits in EPC memory,
- hide memory from Interrogators with a deasserted Untraceable privilege, and/or
- reduce its operating range for all Interrogators.

The memory that a Tag may hide includes words of EPC memory, the Tag serialisation in TID memory, all of TID memory, and/or User memory. Untraceable and traceable Tags behave identically from a state-machine and command-response perspective; the difference between them is (a) the memory the Tag exposes to an Interrogator with a deasserted Untraceable privilege and/or (b) the Tag’s operating range. A Tag only executes an *Untraceable* in the **secured** state. Untraceable has the following fields:

- U specifies a value for the U bit in XPC_W1. Upon receiving an *Untraceable* command, a Tag that supports the U bit shall overwrite bit 21Ch of XPC_W1 with the provided U value regardless of the lock or permalock status of EPC memory. If the Tag does not support the U bit, then the Tag shall ignore the provided U value but continue to process the remainder of the *Untraceable*.
- EPC includes a show/hide bit and 5 length bits. These fields operate independently.
  - Show/hide specifies whether a Tag untraceably hides part of EPC memory. If show/hide=0_2 then a Tag exposes EPC memory. If show/hide=1_2 then a Tag untraceably hides EPC memory above that set by its EPC length field (i.e., StoredPC bits 10h – 14h) to bit 20Fh (inclusive).
  - Length specifies a new EPC length field (L bits). Upon receiving an *Untraceable* command, a Tag shall overwrite its EPC length field (StoredPC bits 10h – 14h) with the provided length bits regardless of the lock or permalock status of EPC memory. In response to subsequent ACKs the Tag backscatters an EPC whose length is set by the new length bits.

- TID specifies the TID memory that a Tag untraceably hides. If TID=00_2 then a Tag exposes TID memory. If TID=01_2 and a Tag's allocation class identifier is E0h then the Tag untraceably hides TID memory above 10h, inclusive; if the Tag's allocation class identifier is E2h then the Tag untraceably hides TID memory above 20h, inclusive. If TID=10_2 then the Tag untraceably hides all TID memory. TID=11_2 is RFU.
- User specifies whether a Tag untraceably hides User memory. If User=0_2 then the Tag exposes User memory. If User=1_2 then the Tag untraceably hides User memory.
- Range specifies a Tag's operating range. If range=00_2 then the Tag persistently enables normal operating range. If range=10_2 then the Tag persistently enables reduced operating range. If range=01_2 then the Tag temporarily toggles its operating range (if normal, then to reduced; if reduced then to normal) but reverts to its prior persistent operating range when the Tag loses power. Temporary toggling allows an Interrogator to confirm that a Tag is still readable before committing range-reduced untraceability to the Tag's non-volatile memory (by sending a subsequent Untraceable with range=10_2). Range=11_2 is RFU. A Tag shall execute a range change prior to replying to the Untraceable. The range-reduction details, including its magnitude and the commands to which it applies, are manufacturer-defined. If a Tag does not support range reduction, then it shall ignore range but continue to process the remainder of the Untraceable.

NOTE: *Because Untraceable is executed from the* ***secured*** *state, it must be handled carefully. The Access password will play an important role. Since ISO 20910 defines the readability of the Access password as a way to know if the tire has been retreaded or not, the tire manufacturers will probably not lock it. Other stakeholders will therefore have the opportunity to encode a non-zero value and permalock it. This stakeholder will be able to make use of the Untraceable command and will be able to either hide the entire EPC or reduce the read range. We therefore strongly recommend making use of RFID chips that DO NOT support Untraceable.*

Authenticate

The *Authenticate* command was introduced in Gen2V2 to allow an Interrogator to perform a cryptographic authentication with a Tag. *Authenticate* allows an Interrogator to perform Tag, Interrogator, or mutual authentication. The generic nature of the *Authenticate* command allows it to support a variety of cryptographic suites. At the date of writing this document, the two cryptographic suites available on the market are: AES-128 (ISO/IEC 29167-10) and Grain-128A (ISO/IEC 29167-13).

For Tag authentication, the Interrogator will issue an *Authenticate* command which specifies several parameters such as:

- the cryptographic suite to be used (that obviously needs to be supported by the Tag)
- the cryptographic key that is to be used (this key is a secret one that needs to be encoded in the Tag and shall never be revealed)
- the challenge (the message that the tag will encrypt)

Based, on the Tag response, the Interrogator will be able to verify that the Tag carries the right key and is therefore genuine.

Other cryptographic related commands like *AuthComm* or *SecureComm* allows either authenticated or encrypted communications from Interrogator to Tag by encapsulating another command in the *AuthComm*/*SecureComm*'s message field.

ABOUT RAIN RFID ALLIANCE

The RAIN RFID Alliance is an organization supporting the universal adoption of RAIN UHF RFID technology. A wireless technology that connects billions of everyday items to the internet, enabling businesses and consumers to identify, locate, authenticate, and engage each item. The technology is based on the EPC Gen2 UHF RFID specification, incorporated into the ISO/IEC 18000-63 standard.

Join the RAIN RFID Alliance to enable connectivity for your business and consumers: identify, locate, authenticate, and engage items in our everyday world. For more information, visit *www.RAINRFID.org*.

ABOUT GS1

GS1 is a neutral, not-for-profit organisation that develops and maintains the most widely used global standards for efficient business communication. We are best known for the barcode, named by the BBC as one of "the 50 things that made the world economy". GS1 standards improve the efficiency, safety and visibility of supply chains across physical and digital channels in 25 sectors. Our scale and reach – local Member Organisations in 115 countries, 2 million user companies and 6 billion transactions every day – help ensure that GS1 standards create a common language that supports systems and processes across the globe. Find out more at www.gs1.org Follow us on Twitter: @GS1 and LinkedIn

RAIN RFID Alliance

401 Edgewater Place
Suite 600
Wakefield, MA 01880

Visit the RAIN RFID website – RAINRFID.org. If you are interested in learning more about the RAIN RFID Alliance, contact us at info@rainrfid.org.