

The **GS1***package

GS1 Code Handler and Barcode Generator[†]

Markus Kohm[‡]

Released 2012/08/01

Abstract

There are several barcode packages out in the world, but they either need PStricks, or are restricted to EAN-13 barcodes. And most of all, they are all $\text{\LaTeX} 2_{\epsilon}$. I've decided to write a package, that supports several GS1 codes, and at almost the same time, I've decided to give L3 a chance. So I've started an experimental GS1 package using `expl3`. Using `expl3` was the main reason writing this package.

Contents

1	L3 Functions and Variables for GS1 Codes	2
2	$\text{\LaTeX} 2_{\epsilon}$ User Interface for GS1 Codes	3
3	Internal Functions and Variables	4
4	GS1 implementation	5
4.1	Implementation of Functions and Variables	5
4.1.1	Constants	5
4.1.2	Settings and Variables	7
4.1.3	Messages	8
4.1.4	Functions	8
4.2	Implementation of the User Interface	12

Index	13
--------------	-----------

*GS1 is a registered trademark of GS1 AISBL. Nevertheless, GS1 AISBL neither has any rights on this package, nor is responsible for it in any kind. The package's name should indicate only, that the package implements some aspects of GS1 codes.

[†]This file describes v9, last revised 2012/08/01.

[‡]E-mail: komascript@gmx.info

Preface

Design and implementation of this package based on

GS1: “Allgemeine GS1 Spezifikation”, Version 12.0, Januar 2012, Ausgabe 1.

This is the official GS1 specification for Germany, Austria and Switzerland.

Currently only EAN-8 and EAN-13 codes and bar codes without extension have been implemented. Others may follow in future.

1 L3 Functions and Variables for GS1 Codes

First of all: Please note, that the concept of private functions and variables is not well defined in \TeX . Several variables, that I'd have made private in C++, haven't been declared to be private in this implementation. Maybe I should change this.

You should also know, that several test files may be created from the package source, and each of those may be used as an example for using the code. Nevertheless, \LaTeX 2_{ϵ} users will not need the following functions and should continue reading with [section 2](#).

$\backslash\text{GS_set_code_digit_seq:Nn}$	$\backslash\text{GS_set_code_digit_seq:Nn}$ $\langle\text{GS1 sequence variable}\rangle$ $\{\langle\text{token list}\rangle\}$
---	--

Makes a GS1 sequence, that consist in digits only, from a $\langle\text{token list}\rangle$. To do so, only the tokens from 0 up to 9 of the $\langle\text{token list}\rangle$ are set to the $\langle\text{GS1 sequence variable}\rangle$. All other tokens are ignored. So may, e.g., convert the string “ISBN 978-3-86541-459-5” into a GS1 sequence with the digits “9783865414595”, where each digit is one item of the sequence.

$\backslash\text{GS_cut_EAN_control_digit:N}$	$\backslash\text{GS_cut_EAN_control_digit:N}$ $\langle\text{GS1 sequence variable}\rangle$
---	--

The $\langle\text{GS1 sequence variable}\rangle$ should store either a EAN-8 or EAN-13 code with or without control digit. If the code has seven or twelve digits, nothing happens. If the code has eight or 13 digits, the last one will be removed. All other cases result in an error message.

$\backslash\text{int_set_to_EAN_control_digit:NN}$	$\backslash\text{int_set_to_EAN_control_digit:NN}$ $\langle\text{integer variable}\rangle$ $\langle\text{GS1 sequence variable}\rangle$
---	--

Calculates the control digit of the $\langle\text{GS1 sequence variable}\rangle$ using the EAN control digit algorithm and stores it into the $\langle\text{integer variable}\rangle$. Note, that the $\langle\text{GS1 sequence variable}\rangle$ may be a sequence of digits of any length not only seven digits for EAN-8 or twelve digits for EAN-13.

$\backslash\text{GS_use_as_EAN_barcode:N}$	$\backslash\text{GS_use_as_EAN_barcode:N}$ $\langle\text{GS1 sequence variable}\rangle$
--	---

Prints an EAN-8 or EAN-13 bar code depending on $\backslash\text{l_GS_code_size_int}$. Note, that the $\langle\text{GS1 sequence variable}\rangle$ may have more than 8 resp. 13 items but not less! Use $\backslash\text{EANBarcode}$ if you need a more save function.

2 $\text{\LaTeX 2}_{\epsilon}$ User Interface for GS1 Codes

This section describes the $\text{\LaTeX 2}_{\epsilon}$ -compatible user interface. Note, that the test files `EANControlDigit.tex`, `EANBarcode.tex`, `GSSetup.tex`, and the resulting PDF files may be used as examples of the following commands.

`\EANControlDigit`

`\EANControlDigit{string}`

Only the digits of the *string* will be used. All other tokens will be ignored. If the *string* has 7 or 8 digits, the control digit of an EAN-8 code will be calculated and output. If the *string* has 12 or 13 digits, the control digit of an EAN-13 code will be calculated and output. If the *string* has 8 or 13 digits the last digit will be ignored. Any other number of digits will result in an error message.

`\EANBarcode`

`\EANBarcode[options]{string}`

Creates the EAN bar code corresponding with *string*. The optional argument *options* may be used to use different settings from the defaults set by `\GSSetup`.

Each digit of a EAN bar code is represented by seven modules. Each module is either black or white. A black module is a black, vertical line. A white module is just a gap. The seven modules start either with a black sequence of up to four modules, followed by a white sequence of up to four modules, followed by a black sequence of up to four modules, finished by a white sequence of up to four modules, or they start with a white sequence of up to four modules, followed by a black sequence of up to four modules, followed by a white sequence of up to four modules, finished by a black sequence of up to four modules.

<hr/> <hr/>	\GSSetup $\{ \langle options \rangle \}$
	$\langle options \rangle$ is a list of $\langle key \rangle = \langle value \rangle$ pairs. They are used to setup the default of several settings:
	ocrb = $\langle boolean \rangle$ If $\langle boolean \rangle$ is true the digits at the bottom of the bar code will be printed using OCR-b font ocrb/T1/m/n in 9pt. Predefined default is ocrb=true .
	module_width = $\langle dimension\ expression \rangle$ This is the width of one module. GS1 specifies a minimum module width of 0.264 mm and a normal width of 0.33 mm. You should not set a width below the minimum!
	module_height = $\langle dimension\ expression \rangle$ This is the height of a black module. GS1 specifies a normal bar code height of 21.31 mm for EAN-8 and 25.01 mm for EAN-13. Both values are inclusive the digits at the bottom of the bar code. Some marker modules are higher than the digit modules.
	code = $\langle string \rangle$ The $\langle string \rangle$ should either be EAN-8 or EAN-13 . The predefined default is EAN-13 . More types will be supported in future.
	scale = $\langle floating\ point \rangle$ This is the scale factor for the bar code. GS1 specifies scale classes from 0.8 up to 2.0 with steps of 0.05. Factors less than 0.8 shouldn't be used. <i>Currently scale won't be used!</i>
	scale_to_font = $\langle boolean \rangle$ Ignore scale and calculate your own scale by the width of digit 0 of the current font. <i>See note to scale above!</i>
	add_control = $\langle boolean \rangle$ Add the control digit to the GS1 code. If there's already a control digit, replace it by the calculated one. The predefined default is add_control=false .

3 Internal Functions and Variables

You should not use or manipulate these! So, maybe it's better to stop reading now.

<hr/> <hr/>	_GS_set_key_code:nn $\{ \langle token\ list \rangle \} \{ \langle integer\ expression \rangle \}$
	Sets \l_GS_code_type_tl to $\langle token\ list \rangle$ and \l_GS_code_size_int to value of $\{ \langle integer\ expression \rangle \}$.

<code>_GS_new_seq_c:cn</code>	<code>_GS_new_seq_c:cn {<sequence name>} {<token list>}</code>
--------------------------------	---

Creates a sequence constant `_GS_<sequence name>_seq`. The value of the constant will be build by the tokens of the *<token list>*. These tokens should be either characters “A” or “B” for selection constants or digits 1–4 for module constants.

<code>_GS_modules:Nn</code>	<code>_GS_modules:Nn <sequence variable> {<dimension expression>}</code>
<code>_GS_modules:(cn NnN)</code>	<code>_GS_modules:cn {<sequence variable name>} {<dimension expression>}</code>
	<code>_GS_modules:NnN <sequence variable> {<dimension expression>} <boolean variable></code>

Draws the modules given by the *<sequence variable>* with height *{<dimension expression>}*. The arguments are:

- #1 : *<sequence variable>* or *<sequence variable name>*, each item of the sequence stays for a number of modules with the same color. `\l_GS_black_bool` signals, whether the (first) modules are black or white and will be reversed after every item. Each module has the width `\l_GS_module_wd_dim`.
- #2 : *<dimension expression>*, the height of the black modules. The modules will be raised by `\l_GS_module_ht_dim`.
- #3 : *<boolean variable>*, `true` indicates, that the first module should be black. With `false`, the first module will be white.

<code>_GS_modules_start_black:Nn</code>	<code>_GS_modules_start_black:Nn <sequence variable> {<dimension expression>}</code>
--	---

Same like `_GS_modules:NnN <sequence variable> {<dimension expression>} \c_true_bool`.

<code>_GS_modules_start_white:Nn</code>	<code>_GS_modules_start_white:Nn <sequence variable> {<dimension expression>}</code>
--	---

Same like `_GS_modules:NnN <sequence variable> {<dimension expression>} \c_false_bool`.

4 GS1 implementation

The implementation has been done in two parts. The first part is the L3 code with all the functions and variables. The second part is the L^AT_EX 2_ε lookalike user interface.

But before this, we just declare, what this is:

```

1 \ProvidesExplPackage
2   {\ExplFileName}{\ExplFileDate}{\ExplFileVersion}{\ExplFileDescription}

```

and what it requires:

```

3 \RequirePackage{expl3}[2012/07/16]
4 \RequirePackage{rule-D}

```

4.1 Implementation of Functions and Variables

4.1.1 Constants

`_GS_new_seq_c:cn` While this is an internal function, that should allow only some tokens at the arguments, it is declared `nopar`.

```

5 \cs_new_nopar:Npn \@@_new_seq_c:cn #1#2
6 {
7   \seq_new:c {c@@_ #1 _seq}
8   \seq_set_split:Nnn \l_tmpa_seq {} {#2}
9   \seq_gset_eq:cN {c@@_ #1 _seq} \l_tmpa_seq
10 }

```

(End definition for _GS_new_seq_c:cn This function is documented on page 5.)

\c__GS_AB0_seq These constants represent the generation rules of the left side of an EAN-13 barcode. See figure 5.2.1.3.1-1 of the GS1 specification.

```

\c__GS_AB1_seq 11 \@@_new_seq_c:cn {AB0} {AAAAAA}
\c__GS_AB2_seq 12 \@@_new_seq_c:cn {AB1} {AABABB}
\c__GS_AB3_seq 13 \@@_new_seq_c:cn {AB2} {AABBAB}
\c__GS_AB4_seq 14 \@@_new_seq_c:cn {AB3} {AABBBAA}
\c__GS_AB5_seq 15 \@@_new_seq_c:cn {AB4} {ABAABB}
\c__GS_AB6_seq 16 \@@_new_seq_c:cn {AB5} {ABBAAB}
\c__GS_AB7_seq 17 \@@_new_seq_c:cn {AB6} {ABBBAA}
\c__GS_AB8_seq 18 \@@_new_seq_c:cn {AB7} {ABABAB}
\c__GS_AB9_seq 19 \@@_new_seq_c:cn {AB8} {ABABBA}
20 \@@_new_seq_c:cn {AB9} {ABBABA}

```

(End definition for \c__GS_AB0_seq and others. These variables are documented on page ??.)

\c__GS_A0_seq These constants represent the module sequences of digits and markers. See figure 5.2.1.2.1-1 and 5.2.1.2.2-1 of the GS1 specification. Note, that the module sequences of type C are same like type A but start with a black module instead of a white one.

```

\c__GS_A1_seq 21 \@@_new_seq_c:cn {A0} {3211} % start white (C0 same but start with black)
\c__GS_A2_seq 22 \@@_new_seq_c:cn {A1} {2221}
\c__GS_A3_seq 23 \@@_new_seq_c:cn {A2} {2122}
\c__GS_A4_seq 24 \@@_new_seq_c:cn {A3} {1411}
\c__GS_A5_seq 25 \@@_new_seq_c:cn {A4} {1132}
\c__GS_A6_seq 26 \@@_new_seq_c:cn {A5} {1231}
\c__GS_A7_seq 27 \@@_new_seq_c:cn {A6} {1114}
\c__GS_A8_seq 28 \@@_new_seq_c:cn {A7} {1312}
\c__GS_A9_seq 29 \@@_new_seq_c:cn {A8} {1213}
\c__GS_B0_seq 30 \@@_new_seq_c:cn {A9} {3112}
\c__GS_B1_seq 31 \@@_new_seq_c:cn {B0} {1123} % start white
\c__GS_B2_seq 32 \@@_new_seq_c:cn {B1} {1222}
\c__GS_B3_seq 33 \@@_new_seq_c:cn {B2} {2212}
\c__GS_B4_seq 34 \@@_new_seq_c:cn {B3} {1141}
\c__GS_B5_seq 35 \@@_new_seq_c:cn {B4} {2311}
\c__GS_B6_seq 36 \@@_new_seq_c:cn {B5} {1321}
\c__GS_B7_seq 37 \@@_new_seq_c:cn {B6} {4111}
\c__GS_B8_seq 38 \@@_new_seq_c:cn {B7} {2131}
\c__GS_B9_seq 39 \@@_new_seq_c:cn {B8} {3121}
\c__GS_margin_seq 40 \@@_new_seq_c:cn {B9} {2113}
\c__GS_separator_seq 41 \@@_new_seq_c:cn {margin} {111} % start black
\c__GS_special_seq 42 \@@_new_seq_c:cn {separator} {11111} % start white
\c__GS_extra_margin_seq 43 \@@_new_seq_c:cn {special} {111111} % start white
\c__GS_extra_separator_seq 44 \@@_new_seq_c:cn {extra_margin} {112} % start black

```

45 \@@_new_seq_c:cn {extra_separator} {11} % start white
 (End definition for \c__GS_A0_seq and others. These variables are documented on page ??.)

There are some basic dimensions for the modules at the specification:

\c__GS_module_min_width_dim \c@@_module_min_width_dim Currently not used.
 \c__GS_module_norm_width_dim 46 \dim_const:Nn \c__GS_module_min_width_dim {0.264mm}
 47 \dim_const:Nn \c__GS_module_norm_width_dim {0.33mm}
 (End definition for \c__GS_module_min_width_dim and \c__GS_module_norm_width_dim These variables are documented on page ??.)

4.1.2 Settings and Variables

These settings will influence the work of several of the user functions. They are defined as keys of family GS1.

\l_GS_use_ocrb_bool Needed to set both \l_GS_code_type_tl and \l_GS_code_size_int with one key. Together they are the type of code, to be handled.
 \l_GS_module_wd_dim
 \l_GS_module_ht_dim
 \l_GS_scale_fp
 \l_GS_scale_to_font_bool
 \l_GS_add_control_bool
 \l_GS_code_type_tl
 \l_GS_code_size_int
 _GS_set_key_code:nn

```

48 \cs_new_nopar:Npn \@@_set_key_code:nn #1#2
49 {
50   \tl_if_exist:NF \l_GS_code_type_tl { \tl_new:N \l_GS_code_type_tl }
51   \tl_set:Nn \l_GS_code_type_tl { #1 }
52   \int_if_exist:NF \l_GS_code_size_int { \int_new:N \l_GS_code_size_int }
53   \int_set:Nn \l_GS_code_size_int { #2 }
54 }
55 \keys_define:nn { GS1 }
56 {
57   ocrb .bool_set:N = \l_GS_use_ocrb_bool,
58   ocrb .initial:n = true,
59   module_width .dim_set:N = \l_GS_module_wd_dim,
60   module_width .initial:V = \c__GS_module_norm_width_dim,
61   module_height .dim_set:N = \l_GS_module_ht_dim,
62   module_height .initial:V = \c_zero_dim,
63   code .choice:,
64   code / EAN-8 .code:n = { \@@_set_key_code:nn { EAN } { 8 } },
65   code / EAN8 .code:n = { \@@_set_key_code:nn { EAN } { 8 } },
66   code / EAN-13 .code:n = { \@@_set_key_code:nn { EAN } { 13 } },
67   code / EAN13 .code:n = { \@@_set_key_code:nn { EAN } { 13 } },
68   code .initial:n = EAN-13,
69   scale .fp_set:N = \l_GS_scale_fp,
70   scale .initial:n = 1.0,
71   scale_to_font .bool_set:N = \l_GS_scale_to_font_bool,
72   scale_to_font .initial:n = false,
73   add_control .bool_set:N = \l_GS_add_control_bool,
74   add_control .initial:n = false,
75 }

```

(End definition for \l_GS_use_ocrb_bool and others. These functions are documented on page 4.)

Note: Later I'll define a $\text{\LaTeX} 2_{\epsilon}$ command to change the defaults of those keys. Additionally local changes of those keys may be done using the optional argument of the $\text{\LaTeX} 2_{\epsilon}$ user commands. See [subsection 4.2](#) for more information.

`\l__GS_code_seq` This will be used later for several local GS1 sequences. It is private and also shouldn't be used in global context.

```
76 \seq_new:N \l__GS_code_seq
```

(End definition for `\l__GS_code_seq` This variable is documented on page ??.)

4.1.3 Messages

```
77 \msg_new:nnnn { GS1 } { EAN-code-size }
78 {
79   #1-isn't~a~valid-EAN~code~\msg_line_context:.
80 }
81 {
82   The~given~code~is~neither~a~EAN-8~with~or~without~control~digit,\\
83   nor~a~EAN-13~with~or~without~control~digit.\\
84   The~GS1~module~currently~only~supports~EAN-8~and~EAN-13.
85 }
```

4.1.4 Functions

`\GS_set_code_digit_seq:Nn` Convert a string into a code sequence ignoring all but digits.

```
86 \cs_new_nopar:Npn \GS_set_code_digit_seq:Nn #1#2
87 {
88   \seq_set_eq:NN #1 \c_empty_seq
89   \tl_map_inline:nn
90     { #2 }
91     {
92       \tl_if_in:nnT
93         {0123456789}
94         {##1}
95         {
96           \seq_put_right:Nn #1 { ##1 }
97         }
98     }
99 }
```

The following test files are used for this code: `GS_set_code_digit.tex`.

(End definition for `\GS_set_code_digit_seq:Nn` This function is documented on page 2.)

`\GS_cut_EAN_control_digit:N` EAN code sequences with control digit are either 8 or 13 digits. To remove the control digit we just have to remove the right most digit from a 8 or 13 digits sequence. 7 or 12 digit sequences are already without control digit. All other sequences are not supported.

```
100 \cs_new_nopar:Npn \GS_cut_EAN_control_digit:N #1
101 {
102   \int_case:nnn
103     { \seq_count:N #1 }
104     {
105       { 7 } { }
106       { 8 } { \seq_pop_right:NN #1 \l_tempa_tl }
107       { 12 } { }
108       { 13 } { \seq_pop_right:NN #1 \l_tempa_tl }

```



```

109     }
110     {
111         \msg_error:nnn { GS1 } { EAN-code-size } { #1 }
112     }
113 }

```

*The following test files are used for this code: `GS_cut_EAN_control_digit.tex`.
(End definition for `\GS_cut_EAN_control_digit:N` This function is documented on page 2.)*

`\int_set_to_EAN_control_digit:NN` Sets an integer to the control digit calculated with the EAN control digit algorithm for a given code sequence. Note, that the complete code sequence will be used to calculate the control digit. So, if you have a EAN-8 or EAN-13 code sequence, you should cut of the control digit first.

```

114 \cs_new_nopar:Npn \int_set_to_EAN_control_digit:NN #1#2
115 {
116     \int_zero:N #1
117     \seq_set_eq:NN \l_tmpa_seq #2
118     \bool_until_do:nn
119     { \seq_if_empty_p:N \l_tmpa_seq }
120     {
121         \seq_pop_left:NN \l_tmpa_seq \l_tmpb_tl
122         \int_if_even:nTF
123         { \seq_count:N \l_tmpa_seq }
124         {
125             \int_add:Nn #1 { 3 * \l_tmpb_tl }
126         }
127         {
128             \int_add:Nn #1 { \l_tmpb_tl }
129         }
130     }
131     \int_set:Nn #1 { \int_mod:nn { 10 - \int_mod:nn { #1 } { 10 } } { 10 } }
132 }

```

*The following test files are used for this code: `int_set_to_EAN_control_digit.tex`.
(End definition for `\int_set_to_EAN_control_digit:NN` This function is documented on page 2.)*

`\GS_set_EAN_control_digit:N` Add a new control digit to a EAN sequence

```

133 \cs_new_nopar:Npn \GS_set_EAN_control_digit:N #1
134 {
135     \GS_cut_EAN_control_digit:N #1
136     \int_set_to_EAN_control_digit:NN \l_tmpa_int #1
137     \seq_put_right:NV #1 \l_tmpa_int
138 }

```

*The following test files are used for this code: `GS_set_EAN_control_digit.tex`.
(End definition for `\GS_set_EAN_control_digit:N` This function is documented on page ??.)*

`__GS_modules:Nn`

`__GS_modules:NnN`

`__GS_modules_start_black:Nn`

`__GS_modules_start_white:Nn`

```

139 \cs_new_nopar:Npn \__GS_modules:Nn #1#2
140 {
141     \seq_map_inline:Nn #1

```

```

142     {
143         \bool_if:NTF \l_@@_black_bool
144         {
145             \bool_set_false:N \l_@@_black_bool
146             \hbox_set:Nn \l_tmpa_box { 0 }
147             \rule:nnn
148             {
149                 \dim_eval:n
150                 {
151                     \box_ht:N \l_tmpa_box + \l_GS_module_wd_dim / 2
152                     - \dim_eval:n { #2 } + \l_GS_module_ht_dim
153                 }
154             }
155             { \dim_eval:n { \l_GS_module_wd_dim * ##1 } }
156             { \dim_eval:n { #2 } }
157         }
158         {
159             \bool_set_true:N \l_@@_black_bool
160             \hbox_to_wd:nn { \l_GS_module_wd_dim * ##1 } { }
161         }
162     }
163 }
164
165 \cs_new_nopar:Npn \@@_modules:NnN #1#2#3
166 {
167     \bool_if_exist:NF \l_@@_black_bool { \bool_new:N \l_@@_black_bool }
168     \bool_set_eq:NN \l_@@_black_bool #3
169     \@@_modules:Nn #1 { #2 }
170 }
171
172 \cs_new_nopar:Npn \@@_modules_start_black:Nn #1#2
173 {
174     \@@_modules:NnN #1 { #2 } \c_true_bool
175 }
176
177 \cs_new_nopar:Npn \@@_modules_start_white:Nn #1#2
178 {
179     \@@_modules:NnN #1 { #2 } \c_false_bool
180 }
181
182 \cs_generate_variant:Nn \@@_modules:Nn { c }

```

The following test files are used for this code: EANBarcode.tex.
(End definition for _GS_modules:Nn and others. These functions are documented on page 5.)

\GS_use_as_EAN_barcode:N Puts the digits, rules, and gaps for an EAN barcode into the input stream.

```

183 \cs_new_nopar:Npn \GS_use_as_EAN_barcode:N #1
184 {
185     \seq_set_eq:NN \l_@@_code_seq #1
186 }

```

```

187 \leavevmode
188
189 \int_compare:nNnTF { \l_GS_code_size_int } { = } { 8 }
190 {
191   \hbox_to_wd:nn { \l_GS_module_wd_dim * 7 } { }
192   \seq_set_eq:Nc \l_GS_system_seq { c_@@_AB0_seq }
193 }
194 {
195   \hbox_to_wd:nn { \l_GS_module_wd_dim * 11 } { }
196   \seq_pop_left:NN \l_@@_code_seq \l_tmpa_tl
197   \seq_set_eq:Nc \l_GS_system_seq { c_@@_AB \l_tmpa_tl _seq }
198   \hbox_overlap_left:n { \l_tmpa_tl }
199 }
200
201 \@@_modules_start_black:Nn \c_@@_margin_seq
202 { \l_GS_module_ht_dim + \l_GS_module_wd_dim * 5 }
203
204 \int_step_inline:nnnn { 1 } { 1 }
205 { \int_div_truncate:nn { \l_GS_code_size_int } { 2 } }
206 {
207   \seq_pop_left:NN \l_@@_code_seq \l_tmpa_tl
208   \hbox_overlap_right:n { \l_tmpa_tl }
209   \seq_pop_left:NN \l_GS_system_seq \l_tmpb_tl
210   \@@_modules:cn { c_@@_ \l_tmpb_tl \l_tmpa_tl _seq }
211   { \l_GS_module_ht_dim }
212 }
213
214 \@@_modules_start_white:Nn \c_@@_separator_seq
215 { \l_GS_module_ht_dim + \l_GS_module_wd_dim * 5 }
216
217 \int_step_inline:nnnn { 1 } { 1 }
218 { \int_div_truncate:nn { \l_GS_code_size_int } { 2 } }
219 {
220   \seq_pop_left:NN \l_@@_code_seq \l_tmpa_tl
221   \hbox_overlap_right:n { \l_tmpa_tl }
222   \@@_modules:cn { c_@@_A \l_tmpa_tl _seq }
223   { \l_GS_module_ht_dim }
224 }
225
226 \@@_modules_start_black:Nn \c_@@_margin_seq
227 { \l_GS_module_ht_dim + \l_GS_module_wd_dim * 5 }
228
229 \hbox_to_wd:nn { \l_GS_module_wd_dim * 7 } { }
230 }

```

The following test files are used for this code: EANBarcode.tex.
(End definition for \GS_use_as_EAN_barcode:N This function is documented on page 2.)

4.2 Implementation of the User Interface

For this, additional packages are needed:

```
231 \RequirePackage{xparse}
```

\EANControlDigit

```
232 \NewDocumentCommand \EANControlDigit
233 { m }
234 {
235   \group_begin:
236     \GS_set_code_digit_seq:Nn \l_@@_code_seq { #1 }
237     \GS_cut_EAN_control_digit:N \l_@@_code_seq
238     \int_set_to_EAN_control_digit:NN \l_tmpa_int \l_@@_code_seq
239     \int_to_arabic:n { \l_tmpa_int }
240   \group_end:
241 }
```

*The following test files are used for this code: EANControlDigit.tex.
(End definition for \EANControlDigit This function is documented on page 3.)*

\EANBarcode

```
242 \NewDocumentCommand \EANBarcode
243 { o m }
244 {
245   \group_begin:
246     \IfNoValueF{#1}{ \keys_set:nn { GS1 } { #1 } }
247     \dim_compare:nNnT { \l_GS_module_ht_dim } { = } { \c_zero_dim }
248     {
249       \int_compare:nNnTF { \l_GS_code_size_int } { = } { 8 }
250       { \dim_set:Nn \l_GS_module_ht_dim { 21.31 mm } }
251       { \dim_set:Nn \l_GS_module_ht_dim { 25.01 mm } }
252     }
253     \bool_if:nT \l_GS_use_ocrb_bool
254     {
```

ToDo: Use fontspec if available.

```
255       \usefont{OT1}{ocrb}{m}{n}\fontsize{9}{9}\selectfont
256     }
257
258     \GS_set_code_digit_seq:Nn \l_@@_code_seq { #2 }
259
260     \bool_if:NT \l_GS_add_control_bool
261     {
262       \GS_set_EAN_control_digit:N \l_@@_code_seq
263     }
264
265     \int_compare:nNnT
266     { \seq_count:N \l__GS_code_seq }
267     { > }
268     { \l_GS_code_size_int }
269     {
```

```

270         \msg_error:nnn { GS1 } { EAN-code-size } { #2 }
271     }
272     \int_while_do:nNnn
273     { \seq_count:N \l_@@_code_seq }
274     { < }
275     { \l_GS_code_size_int }
276     {
277         \seq_put_left:Nn \l_@@_code_seq { 0 }
278     }
279

```

ToDo: Use options `scale` and `scale_to_font`.

```

280         \GS_use_as_EAN_barcode:N \l_@@_code_seq
281     \group_end:
282 }

```

*The following test files are used for this code: `EANBarcode.tex`.
(End definition for `\EANBarcode` This function is documented on page 3.)*

\GSSetup

```

283 \NewDocumentCommand \GSSetup
284 { m }
285 { \keys_set:nn { GS1 } { #1 } }

```

*The following test files are used for this code: `GSSetup.tex`.
(End definition for `\GSSetup` This function is documented on page 4.)*

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

Symbols	
<code>\@@_modules:Nn</code>	139, 169, 182
<code>\@@_modules:NnN</code>	165, 174, 179
<code>\@@_modules:cn</code>	210, 222
<code>\@@_modules_start_black:Nn</code>	172, 201, 226
<code>\@@_modules_start_white:Nn</code>	177, 214
<code>\@@_new_seq_c:cn</code>	5, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45
<code>\@@_set_key_code:nn</code>	48, 64, 65, 66, 67
<code>\@</code>	82, 83
<code>_GS_modules:Nn</code>	4, 139
<code>_GS_modules:NnN</code>	139
<code>_GS_modules_start_black:Nn</code>	4, 139
<code>_GS_modules_start_white:Nn</code>	4, 139
<code>_GS_new_seq_c:cn</code>	4, 5
<code>_GS_set_key_code:nn</code>	4, 48
B	
<code>\bool_if:NT</code>	260
<code>\bool_if:nT</code>	253
<code>\bool_if:NTF</code>	143
<code>\bool_if_exist:NF</code>	167
<code>\bool_new:N</code>	167
<code>\bool_set_eq:NN</code>	168
<code>\bool_set_false:N</code>	145
<code>\bool_set_true:N</code>	159
<code>\bool_until_do:nn</code>	118
<code>\box_ht:N</code>	151
C	
<code>\c_@@_margin_seq</code>	201
<code>\c_@@_module_min_width_dim</code>	46

\c_@@_module_norm_width_dim	47, 60	\dim_eval:n	149, 152, 155, 156
\c_@@_separator_seq	214	\dim_set:Nn	250, 251
\c__GS_A0_seq	21	E	
\c__GS_A1_seq	21	\EANBarcode	3, 242, 242
\c__GS_A2_seq	21	\EANControlDigit	2, 232, 232
\c__GS_A3_seq	21	\ExplFileDate	2
\c__GS_A4_seq	21	\ExplFileDescription	2
\c__GS_A5_seq	21	\ExplFileName	2
\c__GS_A6_seq	21	\ExplFileVersion	2
\c__GS_A7_seq	21	F	
\c__GS_A8_seq	21	\fontsize	255
\c__GS_A9_seq	21	G	
\c__GS_ABO_seq	11	\group_begin:	235, 245
\c__GS_AB1_seq	11	\group_end:	240, 281
\c__GS_AB2_seq	11	\GS_cut_EAN_control_digit:N	2, 100, 100, 135, 237
\c__GS_AB3_seq	11	\GS_set_code_digit_seq:Nn	2, 86, 86, 236, 258
\c__GS_AB4_seq	11	\GS_set_EAN_control_digit:N	133, 133, 262
\c__GS_AB5_seq	11	\GS_use_as_EAN_barcode:N	2, 183, 183, 280
\c__GS_AB6_seq	11	\GSSetup	3, 283, 283
\c__GS_AB7_seq	11	H	
\c__GS_AB8_seq	11	\hbox_overlap_left:n	198
\c__GS_AB9_seq	11	\hbox_overlap_right:n	208, 221
\c__GS_B0_seq	21	\hbox_set:Nn	146
\c__GS_B1_seq	21	\hbox_to_wd:nn	160, 191, 195, 229
\c__GS_B2_seq	21	I	
\c__GS_B3_seq	21	\IfNoValueF	246
\c__GS_B4_seq	21	\int_add:Nn	125, 128
\c__GS_B5_seq	21	\int_case:nnn	102
\c__GS_B6_seq	21	\int_compare:nNnT	265
\c__GS_B7_seq	21	\int_compare:nNnTF	189, 249
\c__GS_B8_seq	21	\int_div_truncate:nn	205, 218
\c__GS_B9_seq	21	\int_if_even:nTF	122
\c__GS_extra_margin_seq	21	\int_if_exist:Nf	52
\c__GS_extra_separator_seq	21	\int_mod:nn	131
\c__GS_margin_seq	21, 226	\int_new:N	52
\c__GS_module_min_width_dim	46	\int_set:Nn	53, 131
\c__GS_module_norm_width_dim	46	\int_set_to_EAN_control_digit:NN	2, 114, 114, 136, 238
\c__GS_separator_seq	21	\int_step_inline:nnnn	204, 217
\c__GS_special_seq	21	\int_to_arabic:n	239
\c_empty_seq	88	\int_while_do:nNnn	272
\c_false_bool	179	\int_zero:N	116
\c_true_bool	174	D	
\c_zero_dim	62, 247	\dim_compare:nNnT	247
\cs_generate_variant:Nn	182	\dim_const:Nn	46, 47
\cs_new_nopar:Npn	5, 48, 86, 100, 114, 133, 139, 165, 172, 177, 183		

K		N	
\keys_define:nn	55	\NewDocumentCommand	232, 242, 283
\keys_set:nn	246, 285	P	
L		\ProvidesExplPackage	1
\l_@@_black_bool	143, 145, 159, 167, 168	R	
\l_@@_code_seq	76, 185, 196, 207, 220, 236, 237, 238, 258, 262, 273, 277, 280	\RequirePackage	3, 4, 231
\l_GS_code_seq	76, 266	\rule:nnn	147
\l_GS_add_control_bool	48, 73, 260	S	
\l_GS_code_size_int	48, 52, 53, 189, 205, 218, 249, 268, 275	\selectfont	255
\l_GS_code_type_tl	48, 50, 51	\seq_count:N	103, 123, 266, 273
\l_GS_module_ht_dim	48, 61, 152, 202, 211, 215, 223, 227, 247, 250, 251	\seq_gset_eq:cN	9
\l_GS_module_wd_dim	48, 59, 151, 155, 160, 191, 195, 202, 215, 227, 229	\seq_if_empty_p:N	119
\l_GS_scale_fp	48, 69	\seq_map_inline:Nn	141
\l_GS_scale_to_font_bool	48, 71	\seq_new:c	7
\l_GS_system_seq	192, 197, 209	\seq_new:N	76
\l_GS_use_ocrb_bool	48, 57, 253	\seq_pop_left:NN	121, 196, 207, 209, 220
\l_tempa_tl	106, 108	\seq_pop_right:NN	106, 108
\l_tmpa_box	146, 151	\seq_put_left:Nn	277
\l_tmpa_int	136, 137, 238, 239	\seq_put_right:Nn	96
\l_tmpa_seq	8, 9, 117, 119, 121, 123	\seq_put_right:NV	137
\l_tmpa_tl	196, 197, 198, 207, 208, 210, 220, 221, 222	\seq_set_eq:Nc	192, 197
\l_tmpb_tl	121, 125, 128, 209, 210	\seq_set_eq:NN	88, 117, 185
\leavevmode	187	\seq_set_split:Nnn	8
M		T	
\msg_error:nnn	111, 270	\tl_if_exist:NF	50
\msg_line_context:	79	\tl_if_in:nnT	92
\msg_new:nnnn	77	\tl_map_inline:nn	89
		\tl_new:N	50
		\tl_set:Nn	51
		U	
		\usefont	255