

The **GS1**^{*}package

GS1 Code Handler and Barcode Generator[†]

Markus Kohm[‡]

Released 2017/07/15

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 $\text{\LaTeX}\ 2\varepsilon$. 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	$\text{\LaTeX}\ 2\varepsilon$ User Interface for GS1 Codes	2
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	7
4.1.4	Functions	8
4.2	Implementation of the User Interface	11

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 v22, last revised 2017/07/15.

[‡]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](#).

`\GS_set_code_digit_seq:Nn \GS_set_code_digit_seq:Nn <GS1 sequence variable> {<token list>}`

Makes a GS1 sequence, that consist in digits only, from a $\langle token list \rangle$. To do so, only the tokens from 0 up to 9 of the $\langle token list \rangle$ are set to the $\langle GS1 sequence variable \rangle$. 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.

`\GS_cut_EAN_control_digit:N \GS_cut_EAN_control_digit:N <GS1 sequence variable>`

The $\langle GS1 sequence variable \rangle$ 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.

`\GS_set_EAN_control_digit:N \GS_set_EAN_control_digit:N <GS1 sequence variable>`

The $\langle GS1 sequence variable \rangle$ should store either a EAN-8 or EAN-13 code with or without control digit. A new control digit will be calculated. If the code has seven or twelve digits the new control digit will be added. If the code has eight or 13 digits, the old control digit will be replaced by the new one. All other cases result in an error message.

`\int_set_to_EAN_control_digit:NN \int_set_to_EAN_control_digit:NN <integer variable> <GS1 sequence variable>`

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

`\GS_use_as_EAN_barcode:N <GS1 sequence variable>`

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

2 L^AT_EX 2 _{ϵ} User Interface for GS1 Codes

This section describes the L^AT_EX 2 _{ϵ} -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.

<code>\GSSetup</code>	<code>\GSSetup {<options>}</code>
	<p><code><options></code> is a list of <code><key>=<value></code> pairs. They are used to setup the default of several settings:</p>
	<code>ocrb=<boolean></code>
	<p>If <code><boolean></code> is <code>true</code> 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 <code>ocrb=true</code>.</p>
	<code>module_width=<dimension expression></code>
	<p>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!</p>
	<code>module_height=<dimension expression></code>
	<p>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.</p>
	<code>code=<string></code>
	<p>The <code><string></code> should either be EAN-8 or EAN-13. The predefined default is EAN-13. More types will be supported in future.</p>
	<code>scale=<floating point></code>
	<p>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></p>
	<code>scale_to_font=<boolean></code>
	<p>Ignore <code>module_width</code> and instead set the module width depending on the width of digit 0 of the current font. Note, that this will not scale the whole bar code but only the module width. To scale the whole bar code, you should use <code>scale</code>.</p>
	<code>add_control=<boolean></code>
	<p>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 <code>add_control=false</code>.</p>

3 Internal Functions and Variables

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

<code>__GS_set_key_code:nn</code>	<code>__GS_set_key_code:nn {<token list>} {<integer expression>}</code>
	<p>Sets <code>\l_GS_code_type_t1</code> to <code><token list></code> and <code>\l_GS_code_size_int</code> to value of <code>{<integer expression>}</code>.</p>
<code>__GS_new_seq_c:cn</code>	<code>__GS_new_seq_c:cn {<sequence name>} {<token list>}</code>
	<p>Creates a sequence constant <code>\c__GS_<sequence name>_seq</code>. The value of the constant will be build by the tokens of the <code><token list></code>. These tokens should be either characters "A" or "B" for selection constants or digits 1-4 for module constants.</p>

```
\_\_GS\_modules:Nn          \_\_GS\_modules:Nn <sequence variable> {\<dimension expression>}
\_\_GS\_modules:cn          \_\_GS\_modules:cn {\<sequence variable name>} {\<dimension expression>}
\_\_GS\_modules:NnN         \_\_GS\_modules:NnN <sequence variable> {\<dimension expression>} <boolean variable>

```

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. $\backslash 1__GS_black_bool$ signals, whether the (first) modules are black or white and will be reversed after every item. Each module has the width $\backslash 1__GS_module_wd_dim$.
- #2 : *<dimension expression>*, the height of the black modules. The modules will be raised by $\backslash 1__GS_module_ht_dim$.
- #3 : *<boolean variable>*, true indicates, that the first module should be black. With false, the first module will be white.

```
\_\_GS\_modules_start_black:Nn \_\_GS\_modules_start_black:Nn <sequence variable> {\<dimension expression>}

```

Same like $\backslash 1__GS_modules:NnN$ *<sequence variable>* *{<dimension expression>}* $\backslash c_true_bool$.

```
\_\_GS\_modules_start_white:Nn \_\_GS\_modules_start_white:Nn <sequence variable> {\<dimension expression>}

```

Same like $\backslash 1__GS_modules:NnN$ *<sequence variable>* *{<dimension expression>}* $\backslash c_false_bool$.

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 $\text{\LaTeX}\ 2\varepsilon$ lookalike user interface.

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

```
1 \ProvidesExplPackage
2   {\ExplFileName}{\ExplFileVersion}{\ExplFileDescription}
and what it requires:
3 \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*.

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

(End definition for $\backslash 1__GS_new_seq_c:cn$.)

```

\c_GS_AB0_seq These constants represent the generation rules of the left side of an EAN-13 barcode. See
\c_GS_AB1_seq figure 5.2.1.3.1-1 of the GS1 specification.
\c_GS_AB2_seq
\c_GS_AB3_seq
\c_GS_AB4_seq
\c_GS_AB5_seq
\c_GS_AB6_seq
\c_GS_AB7_seq
\c_GS_AB8_seq
\c_GS_AB9_seq
10 \@@_new_seq_c:cn {AB0} {AAAAAA}
11 \@@_new_seq_c:cn {AB1} {AABABB}
12 \@@_new_seq_c:cn {AB2} {AABBAB}
13 \@@_new_seq_c:cn {AB3} {AABBBA}
14 \@@_new_seq_c:cn {AB4} {ABAABB}
15 \@@_new_seq_c:cn {AB5} {ABBAAB}
16 \@@_new_seq_c:cn {AB6} {ABBBAA}
17 \@@_new_seq_c:cn {AB7} {ABABAB}
18 \@@_new_seq_c:cn {AB8} {ABABBA}
19 \@@_new_seq_c:cn {AB9} {ABBABA}

```

(End definition for `\c_GS_AB0_seq` and others.)

```

\c_GS_A0_seq These constants represent the module sequences of digits and markers. See figure
\c_GS_A1_seq 5.2.1.2.1-1 and 5.2.1.2.2-1 of the GS1 specification. Note, that the module sequences
\c_GS_A2_seq of type C are same like type A but start with a black module instead of a white one.
\c_GS_A3_seq
\c_GS_A4_seq
\c_GS_A5_seq
\c_GS_A6_seq
\c_GS_A7_seq
\c_GS_A8_seq
\c_GS_A9_seq
\c_GS_BO_seq
\c_GS_B1_seq
\c_GS_B2_seq
\c_GS_B3_seq
\c_GS_B4_seq
\c_GS_B5_seq
\c_GS_B6_seq
\c_GS_B7_seq
\c_GS_B8_seq
\c_GS_B9_seq
\c_GS_margin_seq
\c_GS_separator_seq
\c_GS_special_seq
\c_GS_extra_margin_seq
\c_GS_extra_separator_seq
20 \@@_new_seq_c:cn {AO} {32111} % start white (C0 same but start with black)
21 \@@_new_seq_c:cn {A1} {22211}
22 \@@_new_seq_c:cn {A2} {21222}
23 \@@_new_seq_c:cn {A3} {14111}
24 \@@_new_seq_c:cn {A4} {11322}
25 \@@_new_seq_c:cn {A5} {12311}
26 \@@_new_seq_c:cn {A6} {11144}
27 \@@_new_seq_c:cn {A7} {13122}
28 \@@_new_seq_c:cn {A8} {12133}
29 \@@_new_seq_c:cn {A9} {31122}
30 \@@_new_seq_c:cn {BO} {11233} % start white
31 \@@_new_seq_c:cn {B1} {12222}
32 \@@_new_seq_c:cn {B2} {22122}
33 \@@_new_seq_c:cn {B3} {11411}
34 \@@_new_seq_c:cn {B4} {23111}
35 \@@_new_seq_c:cn {B5} {13211}
36 \@@_new_seq_c:cn {B6} {41111}
37 \@@_new_seq_c:cn {B7} {21311}
38 \@@_new_seq_c:cn {B8} {31211}
39 \@@_new_seq_c:cn {B9} {21133}
40 \@@_new_seq_c:cn {margin} {1111} % start black
41 \@@_new_seq_c:cn {separator} {111111} % start white
42 \@@_new_seq_c:cn {special} {1111111} % start white
43 \@@_new_seq_c:cn {extra_margin} {112} % start black
44 \@@_new_seq_c:cn {extra_separator} {11} % start white

```

(End definition for `\c_GS_A0_seq` and others.)

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

```

\c_GS_module_min_width_dim
\c_GS_module_norm_width_dim
45 \dim_const:Nn \c_@_module_min_width_dim {0.264mm}
46 \dim_const:Nn \c_@_module_norm_width_dim {0.33mm}

(End definition for \c_GS_module_min_width_dim and \c_GS_module_norm_width_dim.)

```

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
\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
\l_GS_set_key_code:nn

47 \cs_new_nopar:Npn \@@_set_key_code:nn #1#2
48 {
49     \tl_if_exist:NF \l_GS_code_type_tl { \tl_new:N \l_GS_code_type_tl }
50     \tl_set:Nn \l_GS_code_type_tl { #1 }
51     \int_if_exist:NF \l_GS_code_size_int { \int_new:N \l_GS_code_size_int }
52     \int_set:Nn \l_GS_code_size_int { #2 }
53 }

54 \keys_define:nn { GS1 }
55 {
56     ocrb          .bool_set:N = \l_GS_use_ocrb_bool,
57     ocrb          .initial:n = true,
58     module_width .dim_set:N = \l_GS_module_wd_dim,
59     module_width .initial:V = \c_@@_module_norm_width_dim,
60     module_height.dim_set:N = \l_GS_module_ht_dim,
61     module_height.initial:V = \c_zero_dim,
62     code          .choice:,
63     code / EAN-8 .code:n = { \@@_set_key_code:nn { EAN } { 8 } },
64     code / EAN8   .code:n = { \@@_set_key_code:nn { EAN } { 8 } },
65     code / EAN-13 .code:n = { \@@_set_key_code:nn { EAN } { 13 } },
66     code / EAN13  .code:n = { \@@_set_key_code:nn { EAN } { 13 } },
67     code          .initial:n = EAN-13,
68     scale          .fp_set:N = \l_GS_scale_fp,
69     scale          .initial:n = 1.0,
70     scale_to_font .bool_set:N = \l_GS_scale_to_font_bool,
71     scale_to_font .initial:n = false,
72     add_control   .bool_set:N = \l_GS_add_control_bool,
73     add_control   .initial:n = false,
74 }
```

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

Note: Later I'll define a L^AT_EX 2 _{ε} command to change the defaults of those keys. Additionally local changes of those keys may be done using the optional argument of the L^AT_EX 2 _{ε} 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.

```
75 \seq_new:N \l_@@_code_seq
```

(End definition for `\l_GS_code_seq`.)

4.1.3 Messages

We need a message for not supported lengths of EAN codes, because currently only EAN-8 and EAN-13 are supported, both with or without control digit. This message will be used as an error message.

```
76 \msg_new:nnnn { GS1 } { EAN-code-size }
```

```

77  {
78      #1~isn't~a~valid~EAN~code~\msg_line_context:..
79  }
80  {
81      The~given~code~is~neither~a~EAN-8~with~or~without~control~digit,\\
82      nor~a~EAN-13~with~or~without~control~digit.\\\\\\
83      The~GS1~module~currently~only~supports~EAN-8~and~EAN-13.
84  }
85 \msg_new:nnn { GS1 } { module/minwidth }
86  {
87      Resulting~module~width~is~less~than~allowed~minimum~\msg_line_context:..\\\\\\
88      GS1~specification~declares~a~minimum~module~width~of~#2.\\\
89      Currently~the~module~with~would~be~#1.\\\
90      To~avoid~problems,~I'll~increase~module~width~to~#2.
91  }

```

4.1.4 Functions

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

```

92 \cs_new_nopar:Npn \GS_set_code_digit_seq:Nn #1#2
93  {
94      \seq_set_eq:NN #1 \c_empty_seq
95      \tl_map_inline:nn
96          { #2 }
97          {
98              \tl_if_in:nnT
99                  {0123456789}
100                 {##1}
101                 {
102                     \seq_put_right:Nn #1 { ##1 }
103                 }
104             }
105         }

```

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.

```

106 \cs_new_nopar:Npn \GS_cut_EAN_control_digit:N #1
107  {
108      \int_case:nnF
109          { \seq_count:N #1 }
110          {
111              { 7 } {  }
112              { 8 } { \seq_pop_right:NN #1 \l_tempa_tl }
113              { 12 } {  }
114              { 13 } { \seq_pop_right:NN #1 \l_tempa_tl }
115          }
116          {

```

```

117         \msg_error:n { GS1 } { EAN-code-size } { #1 }
118     }
119 }
```

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 off the control digit first.

```

120 \cs_new_nopar:Npn \int_set_to_EAN_control_digit:NN #1#2
121 {
122     \int_zero:N #1
123     \seq_set_eq:NN \l_tmpa_seq #2
124     \bool_until_do:nn
125     { \seq_if_empty_p:N \l_tmpa_seq }
126     {
127         \seq_pop_left:NN \l_tmpa_seq \l_tmpb_tl
128         \int_if_even:nTF
129         { \seq_count:N \l_tmpa_seq }
130         {
131             \int_add:Nn #1 { 3 * \l_tmpb_tl }
132         }
133         {
134             \int_add:Nn #1 { \l_tmpb_tl }
135         }
136     }
137     \int_set:Nn #1 { \int_mod:nn { 10 - \int_mod:nn { #1 } { 10 } } { 10 } }
138 }
```

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

```

139 \cs_new_nopar:Npn \GS_set_EAN_control_digit:N #1
140 {
141     \GS_cut_EAN_control_digit:N #1
142     \int_set_to_EAN_control_digit:NN \l_tmpa_int #1
143     \seq_put_right:NV #1 \l_tmpa_int
144 }
```

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 2.)

```

\__GS_modules:Nn
\__GS_modules:cn
\__GS_modules:NnN
\__GS_modules_start_black:Nn
\__GS_modules_start_white:Nn
145 \cs_new_nopar:Npn \@@_modules:Nn #1#2
146 {
147     \seq_map_inline:Nn #1
148     {
149         \bool_if:NTF \l_@@_black_bool
150         {
151             \bool_set_false:N \l_@@_black_bool
```

```

152          \hbox_set:Nn \l_tmpa_box { 0 }
153          \rule:nnn
154          {
155              \dim_eval:n
156              {
157                  \box_ht:N \l_tmpa_box + \l_GS_module_wd_dim / 2
158                  - \dim_eval:n { #2 } + \l_GS_module_ht_dim
159              }
160          }
161          { \dim_eval:n { \l_GS_module_wd_dim * ##1 } }
162          { \dim_eval:n { #2 } }
163      }
164      {
165          \bool_set_true:N \l_@@_black_bool
166          \hbox_to_wd:nn { \l_GS_module_wd_dim * ##1 } { }
167      }
168  }
169 }
170
171 \cs_new_nopar:Npn \@@_modules:NnN #1#2#3
172 {
173     \bool_if_exist:NF \l_@@_black_bool { \bool_new:N \l_@@_black_bool }
174     \bool_set_eq:NN \l_@@_black_bool #3
175     \@@_modules:Nn #1 { #2 }
176 }
177
178 \cs_new_nopar:Npn \@@_modules_start_black:Nn #1#2
179 {
180     \@@_modules:NnN #1 { #2 } \c_true_bool
181 }
182
183 \cs_new_nopar:Npn \@@_modules_start_white:Nn #1#2
184 {
185     \@@_modules:NnN #1 { #2 } \c_false_bool
186 }
187
188 \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.)

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

```

189 \cs_new_nopar:Npn \GS_use_as_EAN_barcode:N #1
190 {
191     \seq_set_eq:NN \l_@@_code_seq #1
192
193     \int_compare:nNnTF { \l_GS_code_size_int } { = } { 8 }
194     {
195         \hbox_to_wd:nn { \l_GS_module_wd_dim * 7 } { }
196         \seq_set_eq:Nc \l_GS_system_seq { c_@@_ABO_seq }
197     }
198     {
199         \hbox_to_wd:nn { \l_GS_module_wd_dim * 11 } { }
200         \seq_pop_left:NN \l_@@_code_seq \l_tmpa_tl

```

```

201     \seq_set_eq:Nc \l_GS_system_seq { c_@@_AB \l_tmpa_tl _seq }
202     \hbox_overlap_left:n { \l_tmpa_tl }
203 }
204
205 \@@_modules_start_black:Nn \c_@@_margin_seq
206   { \l_GS_module_ht_dim + \l_GS_module_wd_dim * 5 }
207
208 \int_step_inline:nnnn { 1 } { 1 }
209   { \int_div_truncate:nn { \l_GS_code_size_int } { 2 } }
210 {
211   \seq_pop_left:NN \l_@@_code_seq \l_tmpa_tl
212   \hbox_overlap_right:n { \l_tmpa_tl }
213   \seq_pop_left:NN \l_GS_system_seq \l_tmpb_tl
214   \@@_modules:cn { c_@@_ \l_tmpb_tl \l_tmpa_tl _seq }
215   { \l_GS_module_ht_dim }
216 }
217
218 \@@_modules_start_white:Nn \c_@@_separator_seq
219   { \l_GS_module_ht_dim + \l_GS_module_wd_dim * 5 }
220
221 \int_step_inline:nnnn { 1 } { 1 }
222   { \int_div_truncate:nn { \l_GS_code_size_int } { 2 } }
223 {
224   \seq_pop_left:NN \l_@@_code_seq \l_tmpa_tl
225   \hbox_overlap_right:n { \l_tmpa_tl }
226   \@@_modules:cn { c_@@_A \l_tmpa_tl _seq }
227   { \l_GS_module_ht_dim }
228 }
229
230 \@@_modules_start_black:Nn \c__GS_margin_seq
231   { \l_GS_module_ht_dim + \l_GS_module_wd_dim * 5 }
232
233 \hbox_to_wd:nn { \l_GS_module_wd_dim * 7 } { }
234 }

```

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:

```
235 \RequirePackage{xparse}
```

\EANControlDigit

```

236 \NewDocumentCommand \EANControlDigit
237   { m }
238 {
239   \group_begin:
240     \GS_set_code_digit_seq:Nn \l_@@_code_seq { #1 }
241     \GS_cut_EAN_control_digit:N \l_@@_code_seq
242     \int_set_to_EAN_control_digit:NN \l_tmpa_int \l_@@_code_seq
243     \int_to_arabic:n { \l_tmpa_int }
244   \group_end:
245 }
```

The following test files are used for this code: *EANControlDigit.tex*.

(End definition for `\EANControlDigit`. This function is documented on page 3.)

\EANBarcode

```
246 \NewDocumentCommand \EANBarcode
247   { o m }
248   {
249     \group_begin:
250     \IfNoValueF{#1}{\keys_set:nn { GS1 } { #1 } }
251     \dim_compare:nNnT { \l_gs_module_ht_dim } { = } { \c_zero_dim }
252     {
253       \int_compare:nNnTF { \l_gs_code_size_int } { = } { 8 }
254       { \dim_set:Nn \l_gs_module_ht_dim { 21.31 mm } }
255       { \dim_set:Nn \l_gs_module_ht_dim { 25.01 mm } }
256     }
257     \bool_if:nT \l_gs_use_ocrb_bool
258     {
ToDo: Use fontspec if available.
259       \usefont{OT1}{ocrb}{m}{n}\fontsize{9}{9}\selectfont
260     }
261
262     \GS_set_code_digit_seq:Nn \l_@@_code_seq { #2 }
263
264     \bool_if:NT \l_gs_add_control_bool
265     {
266       \GS_set_EAN_control_digit:N \l_@@_code_seq
267     }
268
269     \int_compare:nNnT
270       { \seq_count:N \l__gs_code_seq }
271       { > }
272       { \l_gs_code_size_int }
273       {
274         \msg_error:nnn { GS1 } { EAN-code-size } { #2 }
275       }
276     \int_while_do:nNnn
277       { \seq_count:N \l_@@_code_seq }
278       { < }
279       { \l_gs_code_size_int }
280       {
281         \seq_put_left:Nn \l_@@_code_seq { 0 }
282       }
283
284     \bool_if:NT \l_gs_scale_to_font_bool {
285       \hbox_set:Nn \l_tmpa_box { 0 }
286       \dim_set:Nn \l_gs_module_wd_dim { \box_wd:N \l_tmpa_box / 7 }
287     }
288
289     \dim_set:Nn \l_tmpa_dim
290       { \fp_to_decimal:N \l_gs_scale_fp \l_gs_module_wd_dim }
291
292     \dim_compare:nNnT
293       { \l_tmpa_dim }
```

```

294 { < }
295 { \c_@@_module_min_width_dim }
296 {
297     \msg_warning:n { GS1 } { module/minwidth }
298     { \dim_use:N \l_GS_module_wd_dim }
299     { \dim_use:N \c_@@_module_min_width_dim }
300     \dim_set:Nn \l_GS_module_wd_dim
301     {
302         \c_@@_module_min_width_dim *
303         100 / \fp_to_int:n { 100 * \l_GS_scale_fp }
304     }
305 }
306
307 \hbox_set:Nn \l_tmpa_box { \GS_use_as_EAN_barcode:N \l_@@_code_seq }
308 \box_scale:Nnn \l_tmpa_box
309     { \fp_to_int:n { 100 * \l_GS_scale_fp } / 100 }
310     { \fp_to_int:n { 100 * \l_GS_scale_fp } / 100 }
311 \box_use:N \l_tmpa_box
312 \group_end:
313 }

```

The following test files are used for this code: *EANBarcode.tex*.

(End definition for `\EANBarcode`. This function is documented on page 3.)

\GSSetup

```

314 \NewDocumentCommand \GSSetup
315   { m }
316   { \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	
@@ commands:	
\l_@@_black_bool	149, 151, 165, 173, 174
\l_@@_code_seq	75, 191, 200, 211, 224, 240, 241, 242, 262, 266, 277, 281, 307
\c_@@_margin_seq 205
\c_@@_module_min_width_dim 45, 295, 299, 302
\c_@@_module_norm_width_dim 46, 59
\@@_modules:Nn	145, 175, 188, 214, 226
\@@_modules:NnN 171, 180, 185
\@@_modules_start_black:Nn 178, 205, 230
\@@_modules_start_white:Nn 183, 218
\c_@@_separator_seq 218
\c_@@_set_key_code:nn	47, 63, 64, 65, 66
\\ 81, 82, 87, 88, 89
B	
bool commands:	
\bool_if:NTF 149, 264, 284
\bool_if:nTF 257
\bool_if_exist:NTF 173
\bool_new:N 173
\bool_set_eq:NN 174

\bool_set_false:N	151
\bool_set_true:N	165
\bool_until_do:nn	124
\c_false_bool	5, 185
\c_true_bool	5, 180
box commands:	
\box_ht:N	157
\box_scale:Nnn	308
\box_use:N	311
\box_wd:N	286
\l_tmpa_box	
. 152, 157, 285, 286, 307, 308, 311	
C	
cs commands:	
\cs_generate_variant:Nn	188
\cs_new_nopar:Npn	4, 47, 92, 106, 120, 139, 145, 171, 178, 183, 189
D	
dim commands:	
\dim_compare:nNnTF	251, 292
\dim_const:Nn	45, 46
\dim_eval:n	155, 158, 161, 162
\dim_set:Nn	254, 255, 286, 289, 300
\dim_use:N	298, 299
\l_tmpa_dim	289, 293
\c_zero_dim	61, 251
E	
\EANBarcode	2, 3, 246
\EANControlDigit	3, 236
\ExplFileVersion	2
\ExplFileDescription	2
\ExplFileName	2
F	
\fontsize	259
fp commands:	
\fp_to_decimal:N	290
\fp_to_int:n	303, 309, 310
G	
group commands:	
\group_begin:	239, 249
\group_end:	244, 312
GS commands:	
\l_GS_add_control_bool	47, 264
\l_GS_code_size_int	2, 4, 7, 47, 193, 209, 222, 253, 272, 279
\l_GS_code_type_tl	4, 7, 47
\GS_cut_EAN_control_digit:N	2, 106, 141, 241
\l_GS_module_ht_dim	5, 47, 158, 206, 215, 219, 227, 231, 251, 254, 255
\l_GS_module_wd_dim	5, 47, 157, 161, 166, 195, 199, 206, 219, 231, 233, 286, 290, 298, 300
\l_GS_scale_fp	47, 290, 303, 309, 310
\l_GS_scale_to_font_bool	47, 284
\GS_set_code_digit_seq:Nn	2, 92, 240, 262
\GS_set_EAN_control_digit:N	2, 139, 266
\l_GS_system_seq	196, 201, 213
\GS_use_as_EAN_barcode:N	2, 189, 307
\l_GS_use_ocrb_bool	47, 257
GS internal commands:	
\c__GS_{sequence name}_seq	4
\c__GS_A0_seq	20
\c__GS_A1_seq	20
\c__GS_A2_seq	20
\c__GS_A3_seq	20
\c__GS_A4_seq	20
\c__GS_A5_seq	20
\c__GS_A6_seq	20
\c__GS_A7_seq	20
\c__GS_A8_seq	20
\c__GS_A9_seq	20
\c__GS_AB0_seq	10
\c__GS_AB1_seq	10
\c__GS_AB2_seq	10
\c__GS_AB3_seq	10
\c__GS_AB4_seq	10
\c__GS_AB5_seq	10
\c__GS_AB6_seq	10
\c__GS_AB7_seq	10
\c__GS_AB8_seq	10
\c__GS_AB9_seq	10
\c__GS_BO_seq	20
\c__GS_B1_seq	20
\c__GS_B2_seq	20
\c__GS_B3_seq	20
\c__GS_B4_seq	20
\c__GS_B5_seq	20
\c__GS_B6_seq	20
\c__GS_B7_seq	20
\c__GS_B8_seq	20
\c__GS_B9_seq	20
\l__GS_black_bool	5
\l__GS_code_seq	75, 270
\c__GS_extra_margin_seq	20
\c__GS_extra_separator_seq	20
\c__GS_margin_seq	20, 230
\c__GS_module_min_width_dim	45
\c__GS_module_norm_width_dim	45
__GS_modules:Nn	5, 145

__GS_modules:NnN	5, 5, 5, 145	\msg_new:nnnn	76
__GS_modules_start_black:Nn	5, 145	\msg_warning:nnnn	297
__GS_modules_start_white:Nn	5, 145		
__GS_new_seq_c:Nn	4, 4	N	
\c__GS_separator_seq	20	\NewDocumentCommand	236, 246, 314
__GS_set_key_code:nn	4, 47		
\c__GS_special_seq	20	P	
\GSSetup	3, 4, 314	\ProvidesExplPackage	1
H			
hbox commands:			
\hbox_overlap_left:n	202		
\hbox_overlap_right:n	212, 225		
\hbox_set:Nn	152, 285, 307		
\hbox_to_wd:nn	166, 195, 199, 233		
I			
\IfNoValueTF	250		
int commands:			
\int_add:Nn	131, 134		
\int_case:nnTF	108		
\int_compare:nNnTF	193, 253, 269		
\int_div_truncate:nn	209, 222		
\int_if_even:nTF	128		
\int_if_exist:NTF	51		
\int_mod:nn	137		
\int_new:N	51		
\int_set:Nn	52, 137		
\int_set_to_EAN_control_digit:NN	2, 120, 142, 242		
\int_step_inline:nnnn	208, 221		
\int_to_arabic:n	243		
\int_while_do:nNnn	276		
\int_zero:N	122		
\l_tmpa_int	142, 143, 242, 243		
K			
keys commands:			
\keys_define:nn	54		
\keys_set:nn	250, 316		
M			
msg commands:			
\msg_error:nnn	117, 274		
\msg_line_context:	78, 87		
\msg_new:nnn	85		
R			
\RequirePackage	3, 235		
rule commands:			
\rule:nnn	153		
S			
\selectfont	259		
seq commands:			
\c_empty_seq	94		
\seq_count:N	109, 129, 270, 277		
\seq_gset_eq:NN	8		
\seq_if_empty_p:N	125		
\seq_map_inline:Nn	147		
\seq_new:N	6, 75		
\seq_pop_left:NN	127, 200, 211, 213, 224		
\seq_pop_right:NN	112, 114		
\seq_put_left:Nn	281		
\seq_put_right:Nn	102, 143		
\seq_set_eq:NN	94, 123, 191, 196, 201		
\seq_set_split:Nnn	7		
\l_tmpa_seq	7, 8, 123, 125, 127, 129		
T			
tl commands:			
\l_tempa_tl	112, 114		
\tl_if_exist:NTF	49		
\tl_if_in:nnTF	98		
\tl_map_inline:nn	95		
\tl_new:N	49		
\tl_set:Nn	50		
\l_tmpa_tl	200, 201, 202, 211, 212, 214, 224, 225, 226		
\l_tmpb_tl	127, 131, 134, 213, 214		
U			
\usefont	259		