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 LaTeX 2ε. 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

[*] 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 2ε 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 ⟨*token list*⟩. To do so, only the tokens from 0 up to 9 of the ⟨*token list*⟩ are set to the ⟨*GS1 sequence variable*⟩. 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 ⟨*GS1 sequence variable*⟩ 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 ⟨*GS1 sequence variable*⟩ 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 ⟨*GS1 sequence variable*⟩ using the EAN control digit algorithm and stores it into the ⟨*integer variable*⟩. Note, that the ⟨*GS1 sequence variable*⟩ 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`

`\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 ⟨*GS1 sequence variable*⟩ may have more than 8 resp. 13 items but not less! Use `\EANBarcode` if you need a more save function.

2 LaTeX 2ε User Interface for GS1 Codes

This section describes the LaTeX 2ε-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.

\GSSetup

`\GSSetup {⟨options⟩}`

⟨*options*⟩ is a list of ⟨*key*⟩=⟨*value*⟩ pairs. They are used to setup the default of several settings:

`ocrb=`⟨***boolean***⟩
: If ⟨*boolean*⟩ 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=`⟨***dimension expression***⟩
: 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=`⟨***dimension expression***⟩
: 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=`⟨***string***⟩
: The ⟨*string*⟩ should either be `EAN-8` or `EAN-13`. The predefined default is `EAN-13`. More types will be supported in future.

`scale=`⟨***floating point***⟩
: 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. *Currently* `scale` *won't be used!*

`scale_to_font=`⟨***boolean***⟩
: Ignore `module_width` 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 `scale`.

`add_control=`⟨***boolean***⟩
: 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.

__GS_set_key_code:nn

`\__GS_set_key_code:nn {⟨token list⟩} {⟨integer expression⟩}`

Sets `\l_GS_code_type_tl` to ⟨*token list*⟩ and `\l_GS_code_size_int` to value of `{`⟨*integer expression*⟩`}`.

__GS_new_seq_c:cn

`\__GS_new_seq_c:cn {⟨sequence name⟩} {⟨token list⟩}`

Creates a sequence constant `\c__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.

__GS_modules:Nn
__GS_modules:cn
__GS_modules:NnN

```
\__GS_modules:Nn ⟨sequence variable⟩ {⟨dimension expression⟩}
\__GS_modules:cn {⟨sequence variable name⟩} {⟨dimension expression⟩}
\__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. `\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.

__GS_modules_start_black:Nn

```
\__GS_modules_start_black:Nn ⟨sequence variable⟩ {⟨dimension expression⟩}
```

Same like `\__GS_modules:NnN` ⟨*sequence variable*⟩ {⟨*dimension expression*⟩} `\c_true_bool`.

__GS_modules_start_white:Nn

```
\__GS_modules_start_white:Nn ⟨sequence variable⟩ {⟨dimension expression⟩}
```

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 LaTeX 2ε lookalike user interface.

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

```
\ProvidesExplPackage
  {\ExplFileName}{\ExplFileDate}{\ExplFileVersion}{\ExplFileDescription}
```

and what it requires:

```
\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`.

```
\cs_new_nopar:Npn \@@_new_seq_c:cn #1#2
  {
    \seq_new:c {c@@_ #1 _seq}
    \seq_set_split:Nnn \l_tmpa_seq {} {#2}
    \seq_gset_eq:cN {c@@_ #1 _seq} \l_tmpa_seq
  }
```

(*End definition for* `\__GS_new_seq_c:cn`.)

\c__GS_AB0_seq
\c__GS_AB1_seq
\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

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.

```
\@@_new_seq_c:cn {AB0} {AAAAAA}
\@@_new_seq_c:cn {AB1} {AABABB}
\@@_new_seq_c:cn {AB2} {AABBAB}
\@@_new_seq_c:cn {AB3} {AABBBA}
\@@_new_seq_c:cn {AB4} {ABAABB}
\@@_new_seq_c:cn {AB5} {ABBAAB}
\@@_new_seq_c:cn {AB6} {ABBBAA}
\@@_new_seq_c:cn {AB7} {ABABAB}
\@@_new_seq_c:cn {AB8} {ABABBA}
\@@_new_seq_c:cn {AB9} {ABBABA}
```

(*End definition for* `\c__GS_AB0_seq` *and others.*)

\c__GS_A0_seq
\c__GS_A1_seq
\c__GS_A2_seq
\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_B0_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

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.

```
\@@_new_seq_c:cn {A0} {3211} % start white (C0 same but start with black)
\@@_new_seq_c:cn {A1} {2221}
\@@_new_seq_c:cn {A2} {2122}
\@@_new_seq_c:cn {A3} {1411}
\@@_new_seq_c:cn {A4} {1132}
\@@_new_seq_c:cn {A5} {1231}
\@@_new_seq_c:cn {A6} {1114}
\@@_new_seq_c:cn {A7} {1312}
\@@_new_seq_c:cn {A8} {1213}
\@@_new_seq_c:cn {A9} {3112}
\@@_new_seq_c:cn {B0} {1123} % start white
\@@_new_seq_c:cn {B1} {1222}
\@@_new_seq_c:cn {B2} {2212}
\@@_new_seq_c:cn {B3} {1141}
\@@_new_seq_c:cn {B4} {2311}
\@@_new_seq_c:cn {B5} {1321}
\@@_new_seq_c:cn {B6} {4111}
\@@_new_seq_c:cn {B7} {2131}
\@@_new_seq_c:cn {B8} {3121}
\@@_new_seq_c:cn {B9} {2113}
\@@_new_seq_c:cn {margin}          {111} % start black
\@@_new_seq_c:cn {separator}    {11111} % start white
\@@_new_seq_c:cn {special}     {111111} % start white
\@@_new_seq_c:cn {extra_margin}   {112} % start black
\@@_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

```
\dim_const:Nn \c_@@_module_min_width_dim {0.264mm}
\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`
`\__GS_set_key_code:nn`

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.

```
\cs_new_nopar:Npn \@@_set_key_code:nn #1#2
  {
    \tl_if_exist:NF \l_GS_code_type_tl { \tl_new:N \l_GS_code_type_tl }
    \tl_set:Nn \l_GS_code_type_tl { #1 }
    \int_if_exist:NF \l_GS_code_size_int { \int_new:N \l_GS_code_size_int }
    \int_set:Nn \l_GS_code_size_int { #2 }
  }
```

```
\keys_define:nn { GS1 }
  {
    ocrb          .bool_set:N = \l_GS_use_ocrb_bool,
    ocrb          .initial:n  = true,
    module_width  .dim_set:N  = \l_GS_module_wd_dim,
    module_width  .initial:V  = \c_@@_module_norm_width_dim,
    module_height .dim_set:N  = \l_GS_module_ht_dim,
    module_height .initial:V  = \c_zero_dim,
    code          .choice:,
    code / EAN-8  .code:n = { \@@_set_key_code:nn { EAN } {  8 } },
    code / EAN8   .code:n = { \@@_set_key_code:nn { EAN } {  8 } },
    code / EAN-13 .code:n = { \@@_set_key_code:nn { EAN } { 13 } },
    code / EAN13  .code:n = { \@@_set_key_code:nn { EAN } { 13 } },
    code          .initial:n = EAN-13,
    scale         .fp_set:N   = \l_GS_scale_fp,
    scale         .initial:n  = 1.0,
    scale_to_font .bool_set:N = \l_GS_scale_to_font_bool,
    scale_to_font .initial:n  = false,
    add_control   .bool_set:N = \l_GS_add_control_bool,
    add_control   .initial:n  = false,
  }
```

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

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

```
\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.

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

```
  {
    #1~isn't~a~valid~EAN~code~\msg_line_context:.
  }
  {
    The~given~code~is~neither~a~EAN-8~with~or~without~control~digit,\\
    nor~a~EAN-13~with~or~without~control~digit.\\\\
    The~GS1~module~currently~only~supports~EAN-8~and~EAN~13.
  }
```

Another message is only a warning message. It will be used whenever the used module width would be less than the minimum module width given by the GS1 specification.

```
\msg_new:nnn { GS1 } { module/minwidth }
  {
    Resulting~module~width~is~less~than~allowed~minimum~\msg_line_context:.\\\\
    GS1~specification~declares~a~minimum~module~width~of~#2.\\
    Currently~the~module~with~would~be~#1.\\
    To~avoid~problems,~I'll~increase~module~width~to~#2.
  }
```

4.1.4 Functions

`\GS_set_code_digit_seq:Nn`

Convert a string into a code sequence ignoring all but digits.

```
\cs_new_nopar:Npn \GS_set_code_digit_seq:Nn #1#2
  {
    \seq_set_eq:NN #1 \c_empty_seq
    \tl_map_inline:nn
      { #2 }
      {
        \tl_if_in:nnT
          {0123456789}
          {##1}
          {
            \seq_put_right:Nn #1 { ##1 }
          }
      }
  }
```

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.

```
\cs_new_nopar:Npn \GS_cut_EAN_control_digit:N #1
  {
    \int_case:nnF
      { \seq_count:N #1 }
      {
        {  7 } {  }
        {  8 } { \seq_pop_right:NN #1 \l_tempa_tl }
        { 12 } {  }
        { 13 } { \seq_pop_right:NN #1 \l_tempa_tl }
      }
      {
        \msg_error:nnn { GS1 } { EAN-code-size } { #1 }
      }
  }
```

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.

```
\cs_new_nopar:Npn \int_set_to_EAN_control_digit:NN #1#2
  {
    \int_zero:N #1
    \seq_set_eq:NN \l_tmpa_seq #2
    \bool_until_do:nn
      { \seq_if_empty_p:N \l_tmpa_seq }
      {
        \seq_pop_left:NN \l_tmpa_seq \l_tmpb_tl
        \int_if_even:nTF
          { \seq_count:N \l_tmpa_seq }
          {
            \int_add:Nn #1 { 3 * \l_tmpb_tl }
          }
          {
            \int_add:Nn #1 { \l_tmpb_tl }
          }
      }
    \int_set:Nn #1 { \int_mod:nn { 10 - \int_mod:nn { #1 } { 10 } } { 10 } }
  }
```

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

```
\cs_new_nopar:Npn \GS_set_EAN_control_digit:N #1
  {
    \GS_cut_EAN_control_digit:N #1
    \int_set_to_EAN_control_digit:NN \l_tmpa_int #1
    \seq_put_right:NV #1 \l_tmpa_int
  }
```

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`

```
\cs_new_nopar:Npn \@@_modules:Nn #1#2
  {
    \seq_map_inline:Nn #1
      {
        \bool_if:NTF \l_@@_black_bool
          {
            \bool_set_false:N \l_@@_black_bool
            \hbox_set:Nn \l_tmpa_box { 0 }
            \rule:nnn
              {
                \dim_eval:n
                  {
                    \box_ht:N \l_tmpa_box + \l_GS_module_wd_dim / 2
                    - \dim_eval:n { #2 } + \l_GS_module_ht_dim
                  }
              }
              { \dim_eval:n { \l_GS_module_wd_dim * ##1 } }
              { \dim_eval:n { #2 } }
          }
          {
            \bool_set_true:N \l_@@_black_bool
            \hbox_to_wd:nn { \l_GS_module_wd_dim * ##1 } { }
          }
      }
  }

\cs_new_nopar:Npn \@@_modules:NnN #1#2#3
  {
    \bool_if_exist:NF \l_@@_black_bool { \bool_new:N \l_@@_black_bool }
    \bool_set_eq:NN \l_@@_black_bool #3
    \@@_modules:Nn #1 { #2 }
  }

\cs_new_nopar:Npn \@@_modules_start_black:Nn #1#2
  {
    \@@_modules:NnN #1 { #2 } \c_true_bool
  }

\cs_new_nopar:Npn \@@_modules_start_white:Nn #1#2
  {
    \@@_modules:NnN #1 { #2 } \c_false_bool
  }

\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.

```
\cs_new_nopar:Npn \GS_use_as_EAN_barcode:N #1
  {
    \seq_set_eq:NN \l_@@_code_seq #1

    \int_compare:nNnTF { \l_GS_code_size_int } { = } { 8 }
      {
        \hbox_to_wd:nn { \l_GS_module_wd_dim * 7 } { }
        \seq_set_eq:Nc \l_GS_system_seq { c_@@_AB0_seq }
      }
      {
        \hbox_to_wd:nn { \l_GS_module_wd_dim * 11 } { }
        \seq_pop_left:NN \l_@@_code_seq \l_tmpa_tl
        \seq_set_eq:Nc \l_GS_system_seq { c_@@_AB \l_tmpa_tl _seq }
        \hbox_overlap_left:n { \l_tmpa_tl }
      }

    \@@_modules_start_black:Nn \c_@@_margin_seq
      { \l_GS_module_ht_dim + \l_GS_module_wd_dim * 5 }

    \int_step_inline:nnnn { 1 } { 1 }
      { \int_div_truncate:nn { \l_GS_code_size_int } { 2 } }
      {
        \seq_pop_left:NN \l_@@_code_seq \l_tmpa_tl
        \hbox_overlap_right:n { \l_tmpa_tl }
        \seq_pop_left:NN \l_GS_system_seq \l_tmpb_tl
        \@@_modules:cn { c_@@_ \l_tmpb_tl \l_tmpa_tl _seq }
          { \l_GS_module_ht_dim }
      }

    \@@_modules_start_white:Nn \c_@@_separator_seq
      { \l_GS_module_ht_dim + \l_GS_module_wd_dim * 5 }

    \int_step_inline:nnnn { 1 } { 1 }
      { \int_div_truncate:nn { \l_GS_code_size_int } { 2 } }
      {
        \seq_pop_left:NN \l_@@_code_seq \l_tmpa_tl
        \hbox_overlap_right:n { \l_tmpa_tl }
        \@@_modules:cn { c_@@_A \l_tmpa_tl _seq }
          { \l_GS_module_ht_dim }
      }

    \@@_modules_start_black:Nn \c__GS_margin_seq
      { \l_GS_module_ht_dim + \l_GS_module_wd_dim * 5 }

    \hbox_to_wd:nn { \l_GS_module_wd_dim * 7 } { }
  }
```

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:

```
\RequirePackage{xparse}
```

`\EANControlDigit`

```
\NewDocumentCommand \EANControlDigit
  { m }
  {
    \group_begin:
      \GS_set_code_digit_seq:Nn \l_@@_code_seq { #1 }
      \GS_cut_EAN_control_digit:N \l_@@_code_seq
      \int_set_to_EAN_control_digit:NN \l_tmpa_int \l_@@_code_seq
      \int_to_arabic:n { \l_tmpa_int }
    \group_end:
  }
```

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

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

`\EANBarcode`

```
\NewDocumentCommand \EANBarcode
  { o m }
  {
    \group_begin:
      \IfNoValueF{#1}{ \keys_set:nn { GS1 } { #1 } }
      \dim_compare:nNnT { \l_GS_module_ht_dim } { = } { \c_zero_dim }
        {
          \int_compare:nNnTF { \l_GS_code_size_int } { = } { 8 }
            { \dim_set:Nn \l_GS_module_ht_dim { 21.31 mm } }
            { \dim_set:Nn \l_GS_module_ht_dim { 25.01 mm } }
        }
      \bool_if:nT \l_GS_use_ocrb_bool
        {
```

ToDo: Use fontspec if available.

```
          \usefont{OT1}{ocrb}{m}{n}\fontsize{9}{9}\selectfont
        }

      \GS_set_code_digit_seq:Nn \l_@@_code_seq { #2 }

      \bool_if:NT \l_GS_add_control_bool
        {
          \GS_set_EAN_control_digit:N \l_@@_code_seq
        }

      \int_compare:nNnT
        { \seq_count:N \l__GS_code_seq }
        { > }
        { \l_GS_code_size_int }
        {
          \msg_error:nnn { GS1 } { EAN-code-size } { #2 }
        }
      \int_while_do:nNnn
        { \seq_count:N \l_@@_code_seq }
        { < }
        { \l_GS_code_size_int }
        {
          \seq_put_left:Nn \l_@@_code_seq { 0 }
        }

      \bool_if:NT \l_GS_scale_to_font_bool {
        \hbox_set:Nn \l_tmpa_box { 0 }
        \dim_set:Nn \l_GS_module_wd_dim { \box_wd:N \l_tmpa_box / 7 }
      }

     \dim_set:Nn \l_tmpa_dim
       { \fp_to_decimal:N \l_GS_scale_fp \l_GS_module_wd_dim }

     \dim_compare:nNnT
        { \l_tmpa_dim }
```

```
    { < }
    { \c_@@_module_min_width_dim }
    {
     \msg_warning:nnxx { GS1 } { module/minwidth }
      { \dim_use:N \l_GS_module_wd_dim }
      { \dim_use:N \c_@@_module_min_width_dim }
     \dim_set:Nn \l_GS_module_wd_dim
     {
      \c_@@_module_min_width_dim *
      100 / \fp_to_int:n { 100 * \l_GS_scale_fp }
     }
    }

   \hbox_set:Nn \l_tmpa_box { \GS_use_as_EAN_barcode:N \l_@@_code_seq }
   \box_scale:Nnn \l_tmpa_box
    { \fp_to_int:n { 100 * \l_GS_scale_fp } / 100 }
    { \fp_to_int:n { 100 * \l_GS_scale_fp } / 100 }
   \box_use:N \l_tmpa_box
  \group_end:
 }
```

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

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

\GSSetup

```
\NewDocumentCommand \GSSetup
 { m }
 { \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.