

Software-Datentypen

Im Zeitalter der Normung, gibt es im Bereich Programmierung, leider immer noch keine Norm, wie welche Datentypen auszusehen haben. Da macht die Programmiersprache „C“ keine Ausnahme. Ein Integer kann z.B. 16, 32 oder auch 64 Bit breit sein. Die Interpretation hängt von dem verwendeten Prozessor und der Entwicklungsumgebung ab. Durch die sogenannte Integer Promotion kann der erzeugte Programmcode auf unterschiedlichen Systemen zu unterschiedlichem Verhalten führen. Um Missverständnisse zu vermeiden, werden für den gesamten Software Entwicklungsprozess **eindeutige Datentypen** verwendet. Diese befinden sich in der Datei: **Types.H** Die beliebten universellen (jedoch nicht festgelegten) Datentypen „**int**“ werden generell vermieden. Die eigens für dieses und spätere Projekte erstellte Datei Types.H legt alle Datentypen wie folgt fest. Der erste Bezeichnerbuchstabe ist „**S**“ für signed bzw. „**U**“ für unsigned. Dann folgt die Bitbreite des Datentyps. Also ein „**S16**“ ist ein 16 Bit Wert mit Vorzeichen.

Bezeichner	Bitbreite	Vorzeichen	min. Wert	max. Wert
U8	8	nein	0	255
S8	8	ja	-128	+127
U16	16	nein	0	65535
S16	16	ja	-32768	+32767
U32	32	nein	0	4294967296
S32	32	ja	-2147483648	+2147483647
BOOL	8	nein	0 entspricht FALSE 1 entspricht TRUE	<> 0 entspricht TRUE != 0 entspricht TRUE
char	8	nein	0	255

Byte

Wenn in der Software bzw. Dokumentation der Ausdruck „**Byte**“ verwendet wird, ist generell ein **8 Bit** Wert gemeint. Als Datentyp innerhalb des Programmcodes sollte es jedoch niemals auftauchen. Wir verwenden dafür den Typ **U8**.

Word

Bei den Beschreibungen der ARM Prozessoren wird oft vom „WORD“ gesprochen. Nicht nur in Pascal war und ist ein Word 16 Bit breit. ! **Bei den ARM Prozessoren beschreibt es jedoch einen 32 Bit Wert.** Andere Bezeichnungen wären auch Double Word, DWORD oder long longint usw.. Diese leider nicht festgelegte Typenvielfalt führt oft zur Falschinterpretation von Dokumenten. Nach unserer „eindeutigeren“ Typdefinition ist bei unserem Arm Prozessor LPC1768 also ein **U32** gemeint, wenn man das Vorzeichen unberücksichtigt lässt. Der Datentyp Word taucht in unserem Programmcode also niemals auf.

BOOL

Einen BOOL kennt „C“ eigentlich gar nicht. Hierfür verwenden wir ein Byte (**8-Bit**) Wert, wobei der Wert **0** (Null) **FALSE** und ein Wert ungleich **0** (!Null) **TRUE** entspricht. Boolean Werte werden oft in der Software verwendet und sollten eigentlich nur die Werte TRUE oder FALSE annehmen. TRUE ist aber nicht immer mit dem Wert 1 (eins) gleichzusetzen.

Merke : TRUE ist alles was nicht 0 ist.

Die genannten Datentypen werden für die beiden verwendeten Sprachen "C" und Pascal wie folgt festgelegt:

Pascal Deklarationen:

```
Type U8   = Byte;      { unsigned  8 Bit }
Type S8   = ShortInt;  { signed    8 Bit }
Type U16  = Word;      { unsigned 16 Bit }
Type S16  = SmallInt;  { signed   16 Bit }
Type U32  = LongWord;  { unsigned 32 Bit }
Type S32  = LongInt;   { signed   32 Bit }
Type U64  = QWord;     { unsigned 64 Bit }
Type S64  = Int64;     { signed   64 Bit }
Type F32  = Single;    { Floatingpoint 32 Bit }
Type F64  = Double;    { Floatingpoint 64 Bit }
Type BOOL = Boolean;
```

"C" Deklarationen:

```
#define FALSE (0)          /* only the 0 is FALSE */
#define TRUE  (!FALSE)     /* all others are TRUE */

// Deklaration ohne der Notwendigkeit von <stdint.h>
typedef unsigned char  BOOL; /* 0=FALSE all others are TRUE */
typedef unsigned char  U8;   /* 8 Bit without sign */
typedef signed char    S8;   /* 8 Bit with sign */
typedef unsigned short U16;  /* 16 Bit without sign */
typedef signed short   S16;  /* 16 Bit with sign */
typedef unsigned int   U32;  /* 32 Bit without sign */
typedef signed int     S32;  /* 32 Bit with sign */
typedef float          F32;  /* 32 Bit floating point */
                          /* ==>Delphi/Lazarus Type Single */

// Diese Deklaration funktioniert auch, da aber stdint.h benötigt wird, wäre es SOUP
// Es wurde jedoch getestet und funktionierte auch
// #include <stdint.h> // ! Dieser File inkludiert aber noch diverse andere Files...
// typedef uint8_t  BOOL; /* 0=FALSE all others are TRUE */
// typedef uint8_t  U8;   /* 8 Bit without sign */
// typedef int8_t   S8;   /* 8 Bit with sign */
// typedef uint16_t U16;  /* 16 Bit without sign */
// typedef int16_t  S16;  /* 16 Bit with sign */
// typedef uint32_t U32;  /* 32 Bit without sign */
// typedef int32_t  S32;  /* 32 Bit with sign */
// typedef float    F32;  /* 32 Bit floating point ==> Delphi Typ Single */
```