

Teuthida Technologies Home

All about embedded systems engineering.

It takes all types... [Updated]

Posted on [August 20, 2012](#)

All programming languages have data types. This article focuses on integer data types in statically typed environments; like C, C++ and Java. When examining integers, two characteristics are important:

1. The signed-ness of the value; Signed vs Unsigned. Signed integers may represent negative values through the [twos-complement](#) representation. [Note that other signed integer representation systems exist, they just are no longer used]
2. The number of bits in the value. Most typically 8, 16, 32, or 64 bits. [Note that many odd lengths like 12, 18, and 36 bits were once common; the machines that support them are pretty much extinct.]

In classical (pre-C99) C, the signage of integers was specified where the size was implementation dependent. I suppose the idea was that if the programmer specified that a variable was an “int” the compiler would select the most appropriate size for the system in question. It is true that certain minimums were required. An “int” was required to be at least 16 bits in size, but beyond that you never could be sure. The only hard rule could be summarized as:

```
sizeof(short) ≤ sizeof(int) ≤ sizeof(long)
```

and that's it. All three could be 64 bits and it would still be legal under the ANSI rules. In embedded systems, this sort of ambiguity is not acceptable primarily because hardware registers need to be mapped precisely to a data structure laid out exactly the same, to the bit. This can be difficult when the specification says so little about what “int” means.

Other languages have taken different approaches. In Java, four integer data types are defined:

Name	Width	Value Range
Byte	8	-128 ... 127

Short	16	−32,768 to 32,767
Int	32	−2,147,483,648 to 2,147,483,647
Long	64	−9,223,372,036,854,775,808 to 9,223,372,036,854,775,807

This is better, as far as it goes, but there are three serious flaws. First Java does not have any support for unsigned data; a serious omission and second, Java is generally unavailable for embedded systems. These make it quite difficult indeed to use Java. Finally the Java spec still has too much wiggle room. The sizes specified are only **minimums**, nothing prevents an implementer from using 32 bits for Bytes, Shorts and Ints. This is rare, but makes portability problematic again.

So what's to be done to solve this dilemma? The traditional approach has been for projects to have a common include file named something like "GenericTypes.h" that contains a series of typedefs that specify the integer types supported by that compiler. The problem is that there is absolutely no standardization of the names used by these user defined integer types. An 8 bit unsigned integer could be UINT8, UINT_8, uint8, U8 or any one of a myriad possible names. There has to be a better way, and with the ANSI C99 standard there is. C99 introduced the stdint.h header file which defines standard sized integers data types. Our unsigned 8 bit example is now the standard type "uint8_t"! This is great!

There's only one problem. To be useful, code needs to

```
#include <stdint.h>
```

and then exclusively use the new types, or types derived from them. I am as guilty as any. My library code contains a emTypes.h file the defines my own peculiar names for integer data types. The time has come to abandon the proprietary scheme and fully adopt the standard conventions. Over time I plan to shift my code over to the C99 stdint types wherever possible and to build upon stdint types when needed.

[UPDATE] My proposal is this: In cases where a general purpose integer is required, without concern for space, alignment, in a structure, or other special concerns, use plain old "int", as this is supposed to be the "native" integer data type. In all other cases, use the types of <stdint.h>. I call on library writers to adopt the same approach. It will promote interoperability and portability of code and promote standardization of data types.

What are your thoughts? Good idea? As always, your views, comments, suggestions and ideas are most welcomed.

Peter Camilleri (aka Squidly Jones)

SHARE THIS:



LIKE THIS:

Like

Be the first to like this.

This entry was posted in [Embedded Development](#) by [Peter Camilleri](#). Bookmark the [permalink](#) [\[http://teuthida-technologies.com/?p=1159\]](http://teuthida-technologies.com/?p=1159) .

