xassert Module

IN THIS DOCUMENT

- Assertions
- Unreachable
- Fail
- Controlling assertions

This module provides a lightweight and flexible replacement for the standard C header `assert.h`.

The assertions in this module can be enabled/disabled and configured as to how much debug information they show. This configuration can be per "debug unit" (i.e. for sets of files).

To use the module you need to use `module_xassert` in your application and include the `xassert.h` header.

1 Assertions

An assertion can be inserted into code with the `assert` macro e.g.:

```
assert(i < n);
```

Optionally a debug message can be added with the `msg` macro:

```
assert(i < n && msg("i must be less than the array bound"));
```

If assertions are enabled and the expression in the assertion is false than a trap will occur.

2 Unreachable

If the logic of a program dictates that certain code cannot be reached, the `unreachable` macro can be used e.g.:

Publication Date: 2014/3/25 Document Number: XM-UNKNOWNA

```
switch (message) {
case 0:
   ...
case 1:
   ...
default:
  unreachable("message must be 0 or 1");
  break;
}
```

If assertions are enabled then this macro will cause a trap if executed.

3 Fail

A failure can be indicated with the `fail` macro e.g.:

```
if (reg_value != 0xA5)
  fail("device not connected properly")
```

A fail will always cause a trap if executed. A failure differs from unreachable in that an unreachable macro should never execute in a correct program whereas a fail could happen in catastrophic circumstances even if the program is correct.

4 Controlling assertions

Assertions can be enabled/disabled in `debug_conf.h` in your application with the following defines.

XASSERT_ENABLE_ASSERTIONS
: This define can be used to turn assertions on or off (defaults to 1).

XASSERT_ENABLE_DEBUG
: This define will cause assertions to print out the failing expression before trapping (defaults to 0). Note that this option could significantly increase the code size of your application.

XASSERT_ENABLE_LINE_NUMBERS
: This define will cause assertions to print the file and line number of the assertion before trapping. Note that this option could significantly increase the code size of your application.

If `DEBUG_UNIT` is defined when `xassert.h` is included then all the assertions in that file belong to that unit. Assertions can then be controlled per debug unit. The mechanism is similar to that used in `module_logging`.

XASSERT_ENABLE_ASSERTIONS_[debug unit]
: Enable asserts for a particular debug unit. If set to 1, this overrides the default set by `XASSERT_ENABLE_ASSERTIONS` for that debug unit.

XASSERT_ENABLE_DEBUG_[debug unit]
Enable debug messages for a particular debug unit. If set to 1, this overrides the default set by `XASSERT_ENABLE_DEBUG` for that debug unit .

XASSERT_DISABLE_ASSERTIONS_[debug unit]
Disable asserts for a particular debug unit. If set to 1, this overrides the default set by `XASSERT_ENABLE_ASSERTIONS` for that debug unit.

XASSERT_DISABLE_DEBUG_[debug unit]
Disable debug messages for a particular debug unit. If set to 1, this overrides the default set by `XASSERT_ENABLE_DEBUG` for that debug unit .