Random number generation module

IN THIS DOCUMENT

This module provides a random number generator that can be seeded from the hardware. The module can be used by including `module_random` in the `USED_MODULES` variable in an application Makefile and then including the `random.h` header.

The random number generator uses a CRC for generation which is fast in terms of performance and code size and produces reasonably good random numbers. The generation algorithm is equivalent to a Linear Feedback Shift Registor and has a cycle of 2^{32}.

1 API

Configuration defines can be set by creating a file called `random_conf.h` in the application that uses the module.

The module works by creating generators of type `random_generator_t`. These generators can be used to create random numbers (which updates the state of the generator.

`random_generator_t`

Type representing a random number generator.

`random_generator_t random_create_generator_from_seed(unsigned seed)`

Function that creates a random number generator from a seed.

This function has the following parameters:

`seed`	seed for the generator.

This function returns:

a random number generator.

`random_generator_t random_create_generator_from_hw_seed(void)`

Function that attempts to create a random number generator from a true random value into the seed, using an asynchronous timer.

To use this function you must enable the `RANDOM_ENABLE_HW_SEED` define in your application's `random_conf.h`.

This function returns:

a random number generator.

`unsigned random_get_random_number(random_generator_t &g)`

Function that produces a random number.

The number has a cycle of 2^32 and is produced using a LFSR.

This function has the following parameters:

`g`	the used generator to produce the seed.

This function returns:

a random 32 bit number.

For hardware generated seed the following define should be used:

`RANDOM_ENABLE_HW_SEED`

This define controls whether hardware seeded random numbers can be used.

By setting this define, one of the devices ring oscillators will be set running at startup and then can be used later on to seed a random number generator.

Xmos Ltd. is the owner or licensee of this design, code, or Information (collectively, the "Information") and is providing it to you "AS IS" with no warranty of any kind, express or implied and shall have no liability in relation to its use. Xmos Ltd. makes no representation that the Information, or any particular implementation thereof, is or will be free from any claims of infringement and again, shall have no liability in relation to any such claims.