
tinymt-cpp

Release 0.0.1

Takahiro Ueda

Aug 15, 2020

CONTENTS

1	Changelog	5
1.1	[Unreleased]	5
2	tinymt-cpp	7
2.1	Example	7
2.2	Development	7
2.3	License	8
2.4	References	8
	Index	9


```
using tinytmt::tinytmt32 = tinytmt_engine<uint_fast32_t, 32, detail::tinytmt32_default_param_mat1, detail::tinytmt32_default_param_mat2, detail::tinytmt32_default_param_tmat, bool DoPeriodCertification>;
```

TinyMT32 generator engine with the parameter set specified in RFC 8682.

```
template<class UIntType, std::size_t WordSize, UIntType Mat1, UIntType Mat2, UIntType TMat, bool DoPeriodCertification>
class tinytmt::tinytmt_engine
    Pseudo-random number generator engine based on the TinyMT algorithms.
```

Note Currently WordSize must be 32.

Note When Mat, Mat2 and TMat are all zero, the generator parameter set must be provided in initialization (“Dynamic Creation” (DC) mode).

Template Parameters

- UIntType: unsigned integral type generated by the engine
- WordSize: word size for the generated numbers
- Mat1: parameter used in the linear state transition function
- Mat2: parameter used in the linear state transition function
- TMat: parameter used in the non-linear output function
- DoPeriodCertification: whether or not period certification is performed in initialization

Public Types

```
using result_type = UIntType
    Integral type generated by the engine.

using param_type = typename impl::param_type
    Type of the generator parameter set.
```

Public Functions

```
template<TINYMT_CPP_ENABLE_WHEN(!status_type::is_dynamic::value)> tinytmt_engine (result_type seed)
    Constructs the engine (non-DC mode).
```

Parameters

- seed: random seed

```
template<TINYMT_CPP_ENABLE_WHEN(status_type::is_dynamic::value)>
tinytmt_engine (const param_type &param, result_type seed = default_seed)
    Constructs the engine (DC mode).
```

Parameters

- param: parameter set
- seed: random seed

```
void seed (result_type value = default_seed)
    Reinitializes the engine.
```

Parameters

- value: random seed

void **discard** (unsigned long long z)
 Advances the state of the engine by the given amount.

Parameters

- z: number of advances

result_type **operator ()** ()
 Returns the next pseudo-random number.

Return generated value

Public Static Functions

constexpr *result_type* **min** ()
 Returns the smallest possible value in the output range.

Return smallest value

constexpr *result_type* **max** ()
 Returns the largest possible value in the output range.

Return largest value

Public Static Attributes

constexpr std::size_t **word_size** = *WordSize*
 Word size that determines the range of numbers generated by the engine.

constexpr std::size_t **state_size** = impl::state_size
 Size of the state of the generator in words.

constexpr *result_type* **default_seed** = 1
 Default seed.

Friends

friend bool **operator==** (const *tinymt_engine* &a, const *tinymt_engine* &b)
 Compares two engines.

Return true if the engines are equivalent including their internal states, false otherwise

Parameters

- a: first engine
- b: second engine

friend bool **operator!=** (const *tinymt_engine* &a, const *tinymt_engine* &b)
 Compares two engines.

Return true if the engines are not equivalent including their internal states, false otherwise

Parameters

- a: first engine

- `b`: second engine

```
template<class CharT, class Traits>  
friend std::basic_ostream<CharT, Traits> &operator<< (std::basic_ostream<CharT, Traits> &os,  
                                                    const tinymt_engine &e)
```

Serializes the state of the given engine into a stream.

Return `os`

Parameters

- `os`: output stream
- `e`: engine to be serialized

```
template<class CharT, class Traits>  
friend std::basic_istream<CharT, Traits> &operator>> (std::basic_istream<CharT, Traits> &is,  
                                                    tinymt_engine &e)
```

Deserializes the state of the given engine from a stream.

Return `is`

Parameters

- `is`: input stream
- `e`: engine to be deserialized

CHANGELOG

1.1 [Unreleased]

TINYMT-CPP

A C++11 header-only implementation of the [TinyMT](#) pseudo-random number generator.

tinymt-cpp is a one-header-file library written in C++11, which implements a pseudo-random number generator engine based on the TinyMT algorithms. It can be used in the same way as random number generators in the standard library and combined with random number distributions.

2.1 Example

```
#include <iostream>
#include <tinymt/tinymt.h>

int main() {
    tinymt::tinymt32 rng;

    std::cout << rng() << std::endl;
    std::cout << rng() << std::endl;

    return 0;
}
```

2.2 Development

```
# Prerequisites (including optional ones)
brew install cmake doxygen gcc lcov llvm pre-commit

# Pre-commit hooks.
pre-commit install
pre-commit install --hook-type commit-msg

# Linters.
pre-commit run --all-files
```

(continues on next page)

(continued from previous page)

```
# Tests.
cmake -S . -B build/debug -DCMAKE_BUILD_TYPE=Debug
cmake --build build/debug --target check

# Documents.
cmake -S . -B build/docs -DBUILD_TESTING=OFF
cmake --build build/docs --target doc

# Code coverage.
cmake -S . -B build/coverage -DCMAKE_BUILD_TYPE=Debug -DUSE_CODE_COVERAGE=ON
lcov -z -d build/coverage
cmake --build build/coverage --target check
lcov -c -d build/coverage -o build/coverage/coverage.info
genhtml -o build/coverage/html build/coverage/coverage.info

# Compiler sanitizers.
cmake -S . -B build/sanitizer -DCMAKE_BUILD_TYPE=Debug -DUSE_SANITIZER=ON
cmake --build build/sanitizer --target check

# Clang-Tidy.
cmake -S . -B build/clang-tidy -DCMAKE_BUILD_TYPE=Debug -DUSE_CLANG_TIDY=ON
cmake --build build/clang-tidy --target check

# Benchmarking.
cmake -S . -B build/release -DCMAKE_BUILD_TYPE=Release -DBUILD_BENCHMARKING=ON
cmake --build build/release --target bench
```

2.3 License

BSD-3-Clause

2.4 References

- Mutsuo Saito, Makoto Matsumoto, Vincent Roca and Emmanuel Baccelli, TinyMT32 Pseudorandom Number Generator (PRNG), RFC 8682 (2020) 1-12.
- , , , HPC 2011-HPC-131(3) (2011) 1-6.
- Tiny Mersenne Twister (TinyMT): A small-sized variant of Mersenne Twister.
- MersenneTwister-Lab/TinyMT.

T

tinymt::tinyMt32 (C++ *type*), 1
 tinymt::tinyMt_engine (C++ *class*), 1
 tinymt::tinyMt_engine::default_seed
 (C++ *member*), 2
 tinymt::tinyMt_engine::discard (C++ *function*), 1
 tinymt::tinyMt_engine::max (C++ *function*), 2
 tinymt::tinyMt_engine::min (C++ *function*), 2
 tinymt::tinyMt_engine::operator!= (C++
 function), 2
 tinymt::tinyMt_engine::operator() (C++
 function), 2
 tinymt::tinyMt_engine::operator== (C++
 function), 2
 tinymt::tinyMt_engine::operator>> (C++
 function), 3
 tinymt::tinyMt_engine::operator<< (C++
 function), 3
 tinymt::tinyMt_engine::param_type (C++
 type), 1
 tinymt::tinyMt_engine::result_type (C++
 type), 1
 tinymt::tinyMt_engine::seed (C++ *function*),
 1
 tinymt::tinyMt_engine::state_size (C++
 member), 2
 tinymt::tinyMt_engine::tinyMt_engine
 (C++ *function*), 1
 tinymt::tinyMt_engine::word_size (C++
 member), 2