

# seqcnt software ver.A01 設計書

A 版

2025-05-22

<b>1 Introduction</b>	<b>4</b>
1.1 Purpose	4
1.2 Reference	4
<b>2 General Description</b>	<b>5</b>
2.1 Glossary and Abbreviations	5
2.1.1 Glossary (用語集)	5
2.1.2 Abbreviations (略語集)	5
2.2 Architecture	5
2.2.1 Structure (機能構成)	5
2.2.2 Block Diagram (ブロック図)	6
2.2.3 State Transition Diagram (状態遷移図)	6
2.2.4 Sequence Data Structure (シーケンスデータ構造)	6
2.3 Function	6
<b>3 Data Structure Index</b>	<b>7</b>
3.1 Data Structures	7
<b>4 File Index</b>	<b>8</b>
4.1 File List	8
<b>5 Data Structure Documentation</b>	<b>9</b>
5.1 event_data_t Struct Reference	9
5.1.1 Detailed Description	9
5.2 exec_data_t Struct Reference	9
5.2.1 Detailed Description	10
5.3 judge_data_t Struct Reference	10
5.3.1 Detailed Description	10
5.4 seq_context_t Struct Reference	10
5.4.1 Detailed Description	11
<b>6 File Documentation</b>	<b>12</b>
6.1 src/seqcnt.c File Reference	12
6.1.1 Detailed Description	13

6.1.2 Function Documentation . . . . .	13
6.2 src/seqcnt.h File Reference . . . . .	15
6.2.1 Detailed Description . . . . .	17
6.2.2 Enumeration Type Documentation . . . . .	17
6.2.3 Function Documentation . . . . .	18

# 1 Introduction

## 1.1 Purpose

本書は seqcnt software の設計書である。

seqcnt (シーケンス制御機能) はシーケンスステップの遷移条件の判定機能、シーケンスステップ遷移時の処置実行機能を提供する。

ヘッダファイルに予め登録されたシーケンスデータベースに従いシーケンス制御を行う。

## 1.2 Reference

関連文書を以下に示す。

N/A

## 2 General Description

### 2.1 Glossary and Abbreviations

#### 2.1.1 Glossary（用語集）

#	Name	Description
1	TBD	あああ
2	TBD	あああ
3	TBD	あああ

#### 2.1.2 Abbreviations（略語集）

#	Name	Description
1	TBD	あああ
2	TBD	あああ
3	TBD	あああ

### 2.2 Architecture

#### 2.2.1 Structure（機能構成）

機能構成の概要を以下に示す。

L/N	Name	Description
1	seq_ctrl シーケンス制御メイン処理	シーケンス制御機能の状態管理、シーケンスステップの管理、イベント判定処理呼び出し、イベント実行処理呼び出しを行う。
2	event_judge イベント判定処理	Database に登録された参照変数が条件を満たしているか判定する。
3	count_check 判定回数チェック処理	イベント判定が真と判定された回数をカウントし、閾値を上回ったらイベント成立を返す。
4	event_exec イベント実行処理	Database に登録された変数に値を代入する。

2.2.2 Block Diagram (ブロック図)

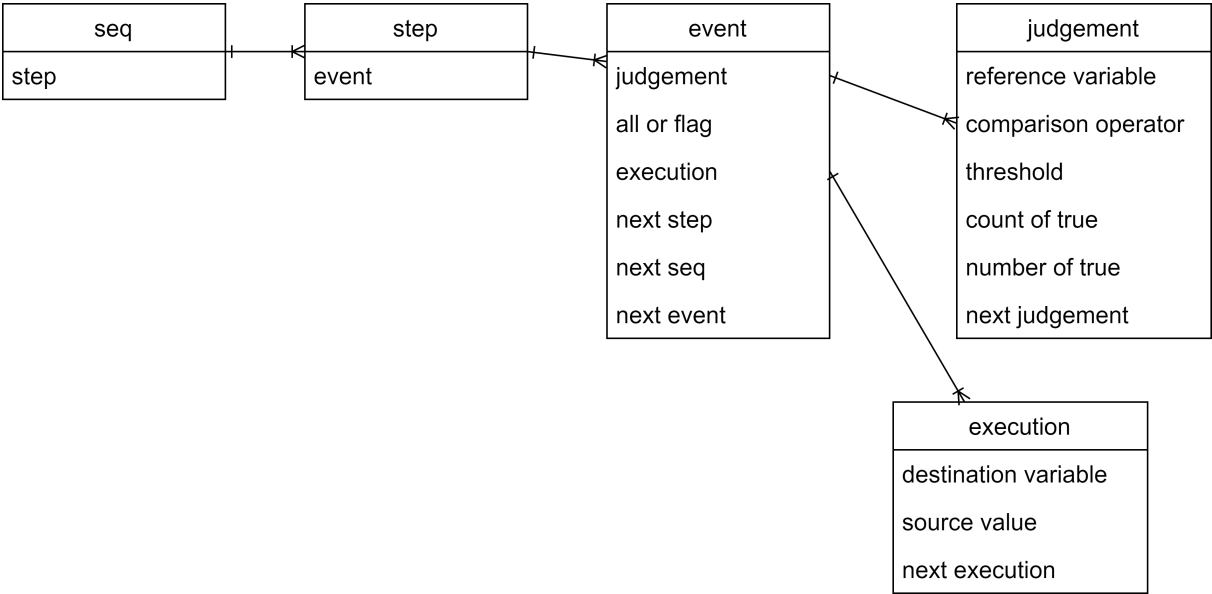
N/A

2.2.3 State Transition Diagram (状態遷移図)

N/A

2.2.4 Sequence Data Structure (シーケンスデータ構造)

シーケンスは下図のデータ構造で配列及び構造体に登録されている。



2.3 Function

機能詳細を以下に示す。

L/N	Name	Description
1	Under Construction 作成中	作成中
2	Under Construction 作成中	作成中

## 3 Data Structure Index

### 3.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">event_data_t</a>	
イベント情報	9
<a href="#">exec_data_t</a>	
イベント実行処理情報	9
<a href="#">judge_data_t</a>	
イベント判定処理情報	10
<a href="#">seq_context_t</a>	
シーケンス制御機能内部状態	10

## 4 File Index

### 4.1 File List

Here is a list of all documented files with brief descriptions:

src/ <a href="#">seqcnt.c</a>	
Sequence control	12
src/ <a href="#">seqcnt.h</a>	
Functions and structures of sequence control	15



## 5 Data Structure Documentation

### 5.1 event\_data\_t Struct Reference

イベント情報

```
#include <seqcnt.h>
```

#### Data Fields

- bool **judge\_all\_or**
- ADDRESS\_T **judge\_data\_address**
- ADDRESS\_T **exec\_data\_address**
- int **next\_step**
- int **next\_seq**
- ADDRESS\_T **next\_event**

#### 5.1.1 Detailed Description

イベント情報

The documentation for this struct was generated from the following file:

- [src/seqcnt.h](#)

### 5.2 exec\_data\_t Struct Reference

イベント実行処理情報

```
#include <seqcnt.h>
```

#### Data Fields

- [var\\_type\\_t](#) **type**
- ADDRESS\_T **dest**
- uint64\_t **src**
- ADDRESS\_T **next**

### 5.2.1 Detailed Description

#### イベント実行処理情報

The documentation for this struct was generated from the following file:

- [src/seqcnt.h](#)

## 5.3 judge\_data\_t Struct Reference

#### イベント判定処理情報

```
#include <seqcnt.h>
```

#### Data Fields

- [var\\_type\\_t](#) **type**
- [judge\\_ope\\_t](#) **ope**
- ADDRESS\_T **ref**
- uint64\_t **thrs**
- int **count**
- int **num**
- ADDRESS\_T **next**

### 5.3.1 Detailed Description

#### イベント判定処理情報

The documentation for this struct was generated from the following file:

- [src/seqcnt.h](#)

## 5.4 seq\_context\_t Struct Reference

#### シーケンス制御機能内部状態

```
#include <seqcnt.h>
```

## Data Fields

- [seq\\_state\\_t](#) **state**
- int **id**
- int **step**
- int **error**

### 5.4.1 Detailed Description

#### シーケンス制御機能内部状態

The documentation for this struct was generated from the following file:

- [src/seqcnt.h](#)

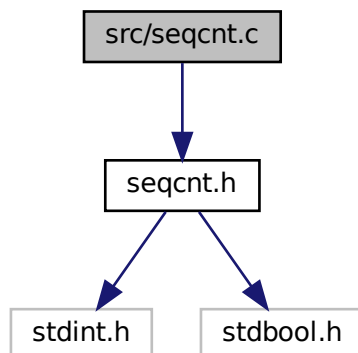
## 6 File Documentation

### 6.1 src/seqcnt.c File Reference

sequence control

```
#include "seqcnt.h"
```

Include dependency graph for seqcnt.c:



#### Functions

- void `seqcnt` (`seq_context_t` \*seq\_context, `seq_context_t` \*seq\_context\_next, const int seq\_id\_data\_base[SEQ\_ID\_MAX], `event_data_t` \*seq\_step\_data\_base[SEQ\_STEP\_MAX])  
シーケンス制御機能メイン処理
- void `test_func` (int input, int \*output)  
テスト処理
- int `event_judge` (ADDRESS\_T judge\_data\_address, bool judge\_all\_or, bool reset\_count, bool \*judge\_result)  
イベント判定処理
- int `check_count` (int num, int \*ctr)  
判定回数チェック処理
- int `event_exec` (ADDRESS\_T exec\_data\_address)  
イベント実行処理

## Variables

- int **seq\_step** = 0
- int **seq\_timer** = 0
- int **step\_timer** = 0
- int **free\_timer** = 0

### 6.1.1 Detailed Description

sequence control

### 6.1.2 Function Documentation

**6.1.2.1 check\_count()** int check\_count (  
    int num,  
    int \* ctr )

判定回数チェック処理

真と判定された回数をカウントし、イベント成立に必要な回数以上であれば、成立 (1) を返す。

**6.1.2.2 event\_exec()** int event\_exec (  
    ADDRESS\_T exec\_data\_address )

イベント実行処理

Database に登録されたイベント実行処理を行う。

**6.1.2.3 event\_judge()** `int event_judge (`  
    `ADDRESS_T judge_data_address,`  
    `bool judge_all_or,`  
    `bool reset_count,`  
    `bool * judge_result )`

#### イベント判定処理

Database に登録された条件に従いイベント判定処理を行う。また、判定回数チェック処理を呼び出す。

**6.1.2.4 seqcnt()** `void seqcnt (`  
    `seq_context_t * seq_context,`  
    `seq_context_t * seq_context_next,`  
    `const int seq_id_data_base[SEQ_ID_MAX],`  
    `event_data_t * seq_step_data_base[SEQ_STEP_MAX] )`

#### シーケンス制御機能メイン処理

以下の処理を実行する。

- ・シーケンス制御機能の状態管理
- ・シーケンスステップの管理
- ・イベントデータの呼び出し
- ・イベント判定処理の呼び出し
- ・イベント実行処理の呼び出し

**6.1.2.5 test\_func()** `void test_func (`  
    `int input,`  
    `int * output )`

#### テスト処理

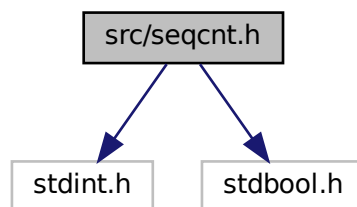
いずれ消す。

## 6.2 src/seqcnt.h File Reference

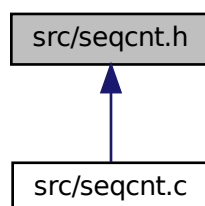
functions and structures of sequence control

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

Include dependency graph for seqcnt.h:



This graph shows which files directly or indirectly include this file:



### Data Structures

- struct `seq_context_t`  
シーケンス制御機能内部状態
- struct `judge_data_t`

- イベント判定処理情報
- struct [exec\\_data\\_t](#)  
イベント実行処理情報
- struct [event\\_data\\_t](#)  
イベント情報

## Macros

- #define **ADDRESS\_T** uint64\_t
- #define **EVENT\_MAX** (10)
- #define **JUDGE\_MAX** (10)
- #define **EXEC\_MAX** (10)
- #define **SEQ\_ID\_MAX** (10)
- #define **SEQ\_STEP\_MAX** (100)
- #define **EVENT\_MAX** (10)
- #define **TIMER\_MAX** (INT32\_MAX)

## Enumerations

- enum [seq\\_state\\_t](#) { **STOP** = 0 , **EXEC** }  
シーケンス制御機能状態定義
- enum [var\\_type\\_t](#) { **INT** = 0 , **DBL** }  
イベント判定/実行処理変数型指定 *ENUM*
- enum [judge\\_op\\_e\\_t](#) {  
**EQ** = 0 , **NEQ** , **LT** , **LTE** ,  
**GT** , **GTE** }  
イベント判定処理比較演算子指定 *ENUM*

## Functions

- void [seqcnt](#) ([seq\\_context\\_t](#) \*seq\_context, [seq\\_context\\_t](#) \*seq\_context\_next, const int seq\_id\_data\_base[SEQ\_ID\_MAX], [event\\_data\\_t](#) \*seq\_step\_data\_base[SEQ\_STEP\_MAX])  
シーケンス制御機能メイン処理
- void [test\\_func](#) (int input, int \*output)  
テスト処理
- int [event\\_judge](#) (ADDRESS\_T judge\_data\_address, bool judge\_all\_or, bool reset\_count, bool \*judge\_result)  
イベント判定処理
- int [check\\_count](#) (int num, int \*ctr)  
判定回数チェック処理
- int [event\\_exec](#) (ADDRESS\_T exec\_data\_address)  
イベント実行処理



## Variables

- int **seq\_step**
- int **seq\_timer**
- int **step\_timer**
- int **free\_timer**

### 6.2.1 Detailed Description

functions and structures of sequence control

### 6.2.2 Enumeration Type Documentation

#### 6.2.2.1 **judge\_ope\_t** enum [judge\\_ope\\_t](#)

イベント判定処理比較演算子指定ENUM

- EQ : =
- NEQ : !=
- LT : <
- LTE : <=
- GT : >
- GTE : >=

#### 6.2.2.2 **seq\_state\_t** enum [seq\\_state\\_t](#)

シーケンス制御機能状態定義

- STOP : シーケンス停止
- EXEC : シーケンス実行中

### 6.2.2.3 `var_type_t` enum `var_type_t`

イベント判定/実行処理変数型指定ENUM

- INT : int
- DBL : double

## 6.2.3 Function Documentation

**6.2.3.1 `check_count()`** `int check_count (`  
    `int num,`  
    `int * ctr )`

判定回数チェック処理

真と判定された回数をカウントし、イベント成立に必要な回数以上であれば、成立 (1) を返す。

**6.2.3.2 `event_exec()`** `int event_exec (`  
    `ADDRESS_T exec_data_address )`

イベント実行処理

Database に登録されたイベント実行処理を行う。

**6.2.3.3 `event_judge()`** `int event_judge (`  
    `ADDRESS_T judge_data_address,`  
    `bool judge_all_or,`  
    `bool reset_count,`  
    `bool * judge_result )`

イベント判定処理

Database に登録された条件に従いイベント判定処理を行う。また、判定回数チェック処理を呼び出す。

**6.2.3.4 seqcnt()** void seqcnt (

```
    seq_context_t * seq_context,  
    seq_context_t * seq_context_next,  
    const int seq_id_data_base[SEQ_ID_MAX],  
    event_data_t * seq_step_data_base[SEQ_STEP_MAX] )
```

#### シーケンス制御機能メイン処理

以下の処理を実行する。

- ・シーケンス制御機能の状態管理
- ・シーケンスステップの管理
- ・イベントデータの呼び出し
- ・イベント判定処理の呼び出し
- ・イベント実行処理の呼び出し

**6.2.3.5 test\_func()** void test\_func (

```
    int input,  
    int * output )
```

#### テスト処理

いずれ消す。