BACnetDemo

CS Lab GmbH | Römerstraße 15 | 47809 Krefeld | Germany | © 2016

Generated by Doxygen 1.8.11

Contents

Chapter 1

BACnet API demo application

This demo application creates a BACnet device with following objects: device object binary input objects (2 switches) binary output objects (2 LED's) analog input objects (2 potentiometers and 2 temperature sensors) analog output object (1 object used by the scheduler object) scheduler object (with entries for every full hour from Monday to Sunday) trend log object (connected to potentiometer 1) calendar object (for the scheduler) notification class object (for alarming)

Only the UDP/IP data link is used by this demo application. You can change to MSTP with enabling the macro 'BACNET_DATA_LINK_MSTP'.

The foreign device service is disabled by default for this demo application. You can enable the service with enabling the macro 'BACNET_DEMO_WITH_FOREIGN_DEVICE'.

The entry point for demo application is the function 'BacnetDemoMain'.

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Data Structure Documentation

4.1 user_analog_input_t Struct Reference

Data Fields

- uint16_t adc_sum_count
- uint32_t adc_sum
- uint16_t adc_value
- float bacnet_value
- float resolution

4.1.1 Detailed Description

Definition at line 83 of file user.c.

4.1.2 Field Documentation

4.1.2.1 uint32_t adc_sum

Definition at line 86 of file user.c.

4.1.2.2 uint16_t adc_sum_count

Definition at line 85 of file user.c.

4.1.2.3 uint16_t adc_value

Definition at line 87 of file user.c.

4.1.2.4 float bacnet_value

Definition at line 88 of file user.c.

4.1.2.5 float resolution

Definition at line 89 of file user.c.

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

- src/demo/user.c

4.2 user_binary_input_t Struct Reference

Data Fields

- bool button
- bool state

4.2.1 Detailed Description

Definition at line 77 of file user.c.

4.2.2 Field Documentation

4.2.2.1 bool button

Definition at line 79 of file user.c.

4.2.2.2 bool state

Definition at line 80 of file user.c.

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

- src/demo/user.c

4.3 user_button_t Struct Reference

Data Fields

- ioport_port_pin_t pin
- ioport_level_t level

4.3.1 Detailed Description

Definition at line 71 of file user.c.

4.3.2 Field Documentation

4.3.2.1 ioport_level_t level

Definition at line 74 of file user.c.

4.3.2.2 ioport_port_pin_t pin

Definition at line 73 of file user.c.

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

- src/demo/user.c

4.4 user_record_t Struct Reference

```
#include <user.h>
```

Data Fields

- BACNET_LOG_RECORD data

4.4.1 Detailed Description

Definition at line 28 of file user.h.

4.4.2 Field Documentation

4.4.2.1 BACNET_LOG_RECORD data

Definition at line 30 of file user.h.

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

- src/demo/user.h

4.5 user_trendlog_t Struct Reference

Data Fields

- uint8_t pos
- uint32_t count
- user_record_t records [USER_MAX_RECORDS]

4.5.1 Detailed Description

Definition at line 92 of file user.c.

4.5.2 Field Documentation

4.5.2.1 uint32_t count

Definition at line 95 of file user.c.

4.5.2.2 uint8_t pos

Definition at line 94 of file user.c.

4.5.2.3 user_record_t records[USER_MAX_RECORDS]

Definition at line 96 of file user.c.

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

- src/demo/user.c

Chapter 5

File Documentation

5.1 src/demo/adc.c File Reference

```
#include "hal_data.h"
#include "adc.h"
#include "sf_cslab_bacnet_common_api.h"
```

Macros

- #define ADC_REGISTER_COUNT (4)

Functions

- bool AdcInit (void)

  Initialize ADC Unit.
- void AdcCompleteCallback (adc_callback_args_t ∗p_args)

  This Callback is used to store the current values read from used registers.
- bool AdcGetValue (uint8_t index, uint16_t ∗p_value)

  Returns the current value of a register.

Variables

- const adc_instance_t g_adc
- const adc_register_t g_adc_register [ADC_REGISTER_COUNT]
- static uint16_t g_adc_data [ADC_REGISTER_COUNT]
- static bool g_adc_started = false

5.1.1 Macro Definition Documentation

5.1.1.1 #define ADC_REGISTER_COUNT (4)

Definition at line 22 of file adc.c.

5.1.2 Function Documentation

5.1.2.1 void AdcCompleteCallback (adc_callback_args_t ∗ *p_args*)

This Callback is used to store the current values read from used registers.

Definition at line 106 of file adc.c.

5.1.2.2 bool AdcGetValue (uint8_t *index,* uint16_t ∗ *p_value*)

Returns the current value of a register.

Parameters

`in`	*index*	Index for the register value array.
`out`	*p_value*	Pointer for returning the register value.

Return values

true	Returned register value is valid.
false	Demo application is not completely initialized or wrong index for the array.

Definition at line 140 of file adc.c.

5.1.2.3 bool AdcInit (void)

Initialize ADC Unit.

The Open function applies power to the A/D peripheral, sets the operational mode, trigger sources, interrupt priority, and configurations for the peripheral as a whole. Configure the ADC scan parameters and starts a software scan or enables the hardware trigger for a scan depending on how the triggers were configured in the Open() call.

The global register values were initialized with 0.

Return values

true	Initialization was successful and ADC has started.
false	Invalid p_ctrl or p_cfg pointer.

Definition at line 69 of file adc.c.

5.1.3 Variable Documentation

5.1.3.1 const adc_instance_t g_adc

5.1.3.2 uint16_t g_adc_data[ADC_REGISTER_COUNT] `[static]`

Definition at line 48 of file adc.c.

5.1.3.3 const adc_register_t g_adc_register[ADC_REGISTER_COUNT]

Initial value:

```
=
{
    ADC_SAMPLE_STATE_CHANNEL_0,
    ADC_SAMPLE_STATE_CHANNEL_1,
    ADC_SAMPLE_STATE_CHANNEL_3,
    ADC_SAMPLE_STATE_CHANNEL_4
}
```

Definition at line 40 of file adc.c.

5.1.3.4 bool g_adc_started = false [static]

Definition at line 50 of file adc.c.

5.2 src/demo/adc.h File Reference

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

Functions

- bool AdcInit (void)

  Initialize ADC Unit.
- void AdcCompleteCallback (adc_callback_args_t ∗p_args)

  This Callback is used to store the current values read from used registers.
- bool AdcGetValue (uint8_t index, uint16_t ∗p_value)

  Returns the current value of a register.

5.2.1 Function Documentation

5.2.1.1 void AdcCompleteCallback (adc_callback_args_t ∗ *p_args*)

This Callback is used to store the current values read from used registers.

Definition at line 106 of file adc.c.

5.2.1.2 bool AdcGetValue (uint8_t *index*, uint16_t ∗ *p_value*)

Returns the current value of a register.

Parameters

in	*index*	Index for the register value array.
out	*p_value*	Pointer for returning the register value.

Return values

true	Returned register value is valid.
false	Demo application is not completely initialized or wrong index for the array.

Definition at line 140 of file adc.c.

5.2.1.3 bool AdcInit (void)

Initialize ADC Unit.

The Open function applies power to the A/D peripheral, sets the operational mode, trigger sources, interrupt priority, and configurations for the peripheral as a whole. Configure the ADC scan parameters and starts a software scan or enables the hardware trigger for a scan depending on how the triggers were configured in the Open() call.

The global register values were initialized with 0.

Return values

true	Initialization was successful and ADC has started.
false	Invalid p_ctrl or p_cfg pointer.

Definition at line 69 of file adc.c.

5.3 src/demo/bacnetdemo.c File Reference

```
#include <demo/bacnetdemo.h>
#include "nx_api.h"
#include "hal_data.h"
#include "bacnet.h"
#include "definitions.h"
#include "adc.h"
#include "user.h"
#include "sf_cslab_bacnet_common.h"
#include "sf_cslab_bacnet_ip.h"
```

Macros

- #define BACNET_DEMO_WITH_FOREIGN_DEVICE

Functions

- static bool bacnet_init (BACNET_TIME_STAMP ∗p_start_time)

    Initialize the BACnet server.
- static void bacnet_reset (void)

    Called in case of a reset request.
- static BACNET_CALLBACK_STATUS bacnet_write (BACNET_INST_NUMBER device_id, BACNET_OBJ←ECT_TYPE obj_type, BACNET_INST_NUMBER inst_number, BACNET_PROPERTY_ID property_id, BA←CNET_ARRAY_INDEX index, const BACNET_PROPERTY_CONTENTS ∗p_property_contents, BACNET←_BOOLEAN ∗p_property_persistent)

    This function is called by the BACnet API for every changed property of an object of a device.
- static BACNET_LOG_RECORD ∗ bacnet_trend (BACNET_READ_RANGE_INFO ∗p_service_info, BAC_←UINT ∗p_entries, BAC_BOOLEAN ∗p_first, BAC_BOOLEAN ∗p_last, BAC_BOOLEAN ∗p_more)

    Request for trend log data.
- static BAC_BOOLEAN bacnet_storage_read (void)

    Notification that the user should read persistent stored data.
- static BAC_BOOLEAN bacnet_storage_write (void)

    Notification that the user should write persistent stored data.
- static BAC_BOOLEAN bacnet_storage_delete (void)

    Notification that the user should delete persistent stored data.
- void BacnetDemoMain (void)

    This is the main function of the demo application and is called by the thread-x framework.

Variables

- static sf_cslab_bacnet_ctrl_t g_bacnet_data_link_ctrl
- static const sf_cslab_bacnet_on_sf_cslab_bacnet_ip_cfg g_bacnet_data_link_ip_ext_cfg
- static const sf_cslab_bacnet_cfg_t g_bacnet_data_link_cfg
- const sf_cslab_bacnet_instance_t g_bacnet_instance
- static sf_cslab_bacnet_common_ctrl_t g_bacnet_common_ctrl
- static sf_cslab_bacnet_common_cfg_t g_bacnet_common_cfg
- const sf_cslab_bacnet_common_instance_t g_bacnet_common_instance
- static BACNET_LOG_RECORD g_log_record [OBJ_TR1_LOG_BUFFER_SIZE]

5.3.1 Macro Definition Documentation

5.3.1.1 #define BACNET_DEMO_WITH_FOREIGN_DEVICE

Enable this macro if the data link is MSTP instead of IP Enable this macro if the demo application use a foreign device

Definition at line 42 of file bacnetdemo.c.

5.3.2 Function Documentation

5.3.2.1 static bool bacnet_init (BACNET_TIME_STAMP ∗ *p_start_time*) [static]

Initialize the BACnet server.

Returns

Initialization was successful or not.

Definition at line 169 of file bacnetdemo.c.

5.3.2.2 static void bacnet_reset (void) [static]

Called in case of a reset request.

Definition at line 246 of file bacnetdemo.c.

5.3.2.3 static BAC_BOOLEAN bacnet_storage_delete (void) [static]

Notification that the user should delete persistent stored data.

Returns

Deleting was successful or not.

Definition at line 386 of file bacnetdemo.c.

5.3.2.4 static BAC_BOOLEAN bacnet_storage_read (void) [static]

Notification that the user should read persistent stored data.

Returns

Reading was successful or not.

Definition at line 366 of file bacnetdemo.c.

5.3.2.5 static BAC_BOOLEAN bacnet_storage_write (void) [static]

Notification that the user should write persistent stored data.

Returns

Writing was successful or not.

Definition at line 376 of file bacnetdemo.c.

5.3.2.6 static BACNET_LOG_RECORD ∗ bacnet_trend (BACNET_READ_RANGE_INFO ∗ *p_service_info,* BAC_UINT ∗ *p_entries,* BAC_BOOLEAN ∗ *p_first,* BAC_BOOLEAN ∗ *p_last,* BAC_BOOLEAN ∗ *p_more*) [static]

Request for trend log data.

Parameters

in	*p_service_info*	A pointer to a data structure which contains all service parameters this service provides and the application needs to process.
out	*p_entries*	Number of records in returned trend log array.
out	*p_first*	First record is in returned trend log array.
out	*p_last*	Last record is in returned trend log array.
out	*p_more*	More records available but not in returned trend log array.

Returns

On success a pointer to an array with trend log records will be returned otherwise a NULL pointer.

Definition at line 263 of file bacnetdemo.c.

5.3.2.7 static BACNET_CALLBACK_STATUS bacnet_write (BACNET_INST_NUMBER *device_id*, BACNET_OBJECT_TYPE *obj_type*, BACNET_INST_NUMBER *inst_number*, BACNET_PROPERTY_ID *property_id*, BACNET_ARRAY_INDEX *index*, const BACNET_PROPERTY_CONTENTS ∗ *p_property_contents*, BACNET_BOOLEAN ∗ *p_property_persistent*) `[static]`

This function is called by the BACnet API for every changed property of an object of a device.

For this demo application only binary output, analog output and trend log objects were processed.

Parameters

`in`	*device_id*	The device of the changed property.
`in`	*obj_type*	The object type of the changed property.
`in`	*inst_number*	The instance number of the changed property.
`in`	*property_id*	This property that has been changed.
`in`	*index*	Index for array properties.
`in`	*p_property_contents*	The new content of the property (property type depending).
`out`	*p_property_persistent*	The property should be stored by the user or not.

Return values

status	Informs the BACnet API about success or errors while processing. Is always CALLBACK_STATUS_DEFAULT for this demo application.

Definition at line 407 of file bacnetdemo.c.

5.3.2.8 void BacnetDemoMain (void)

This is the main function of the demo application and is called by the thread-x framework.

This function open the defined common bacnet instance and the connected data link. On success the common bacnet instance will be initialized and started. These two calls to the open() below will be called by the ISDE when there are Synergy Configurator XMLs available.

Open BACnet common code.

Open up data link layer. This could be IP or MS/TP depending on the pointers in g_bacnet_data_link_instance.

At this point this is basically the user's application. This is code they will be responsible for writing. All of the BACnet objects are taken care of in a different file.

Definition at line 454 of file bacnetdemo.c.

5.3.3 Variable Documentation

5.3.3.1 sf_cslab_bacnet_common_cfg_t g_bacnet_common_cfg [static]

Initial value:

```
=
{
    .period_ticks = 1
}
```

Definition at line 140 of file bacnetdemo.c.

5.3.3.2 sf_cslab_bacnet_common_ctrl_t g_bacnet_common_ctrl [static]

Initial value:

```
=
{
    .p_instance = &g_bacnet_instance
}
```

Definition at line 135 of file bacnetdemo.c.

5.3.3.3 const sf_cslab_bacnet_common_instance_t g_bacnet_common_instance

Initial value:

```
=
{
    .p_ctrl   = &g_bacnet_common_ctrl,
    .p_cfg    = &g_bacnet_common_cfg,
    .p_api    = &g_sf_cslab_bacnet_common_on_sf_cslab_bacnet_common,
}
```

Definition at line 145 of file bacnetdemo.c.

5.3.3.4 const sf_cslab_bacnet_cfg_t g_bacnet_data_link_cfg [static]

Initial value:

```
=
{
    .p_server    = &g_server,
    .priority    = 2,


    .p_extend    = &g_bacnet_data_link_ip_ext_cfg

}
```

Definition at line 113 of file bacnetdemo.c.

5.3.3.5 sf_cslab_bacnet_ctrl_t g_bacnet_data_link_ctrl [static]

Initial value:

```
=
{
    .p_instance_ctrl    = NULL,
    .init               = bacnet_init,
    .reset              = bacnet_reset,
    .write              = bacnet_write,
    .trend              = bacnet_trend,
    .storage_read       = bacnet_storage_read,
    .storage_write      = bacnet_storage_write,
    .storage_delete     = bacnet_storage_delete
}
```

Definition at line 88 of file bacnetdemo.c.

5.3.3.6 const sf_cslab_bacnet_on_sf_cslab_bacnet_ip_cfg g_bacnet_data_link_ip_ext_cfg [static]

Initial value:

```
=
{
    .ip     = {DEV_DATALINK_ETH_IP1, DEV_DATALINK_ETH_IP2,
      DEV_DATALINK_ETH_IP3, DEV_DATALINK_ETH_IP4},
    .port   = DEV_DATALINK_ETH_PORT
}
```

Definition at line 106 of file bacnetdemo.c.

5.3.3.7 const sf_cslab_bacnet_instance_t g_bacnet_instance

Initial value:

```
=
{
    .p_ctrl = &g_bacnet_data_link_ctrl,
    .p_cfg  = &g_bacnet_data_link_cfg,


    .p_api  = &g_sf_cslab_bacnet_on_sf_cslab_bacnet_ip

}
```

Definition at line 124 of file bacnetdemo.c.

5.3.3.8 BACNET_LOG_RECORD g_log_record[OBJ_TR1_LOG_BUFFER_SIZE] [static]

Definition at line 158 of file bacnetdemo.c.

5.4 src/demo/bacnetdemo.h File Reference

```
#include "bacnet.h"
```

Enumerations

- enum demo_obj_ids_t {
  DEMO_OBJ_ID_DEV = 0, DEMO_OBJ_ID_BI1, DEMO_OBJ_ID_BI2, DEMO_OBJ_ID_BO1,
  DEMO_OBJ_ID_BO2, DEMO_OBJ_ID_AI1, DEMO_OBJ_ID_AI2, DEMO_OBJ_ID_AI3,
  DEMO_OBJ_ID_AI4, DEMO_OBJ_ID_AO1, DEMO_OBJ_ID_SC1, DEMO_OBJ_ID_CA1,
  DEMO_OBJ_ID_TR1, DEMO_OBJ_ID_NC1, DEMO_OBJ_ID_MAX }

Functions

- void BacnetDemoMain (void)

  This is the main function of the demo application and is called by the thread-x framework.

5.4.1 Enumeration Type Documentation

5.4.1.1 enum demo_obj_ids_t

Enumerator

DEMO_OBJ_ID_DEV

DEMO_OBJ_ID_BI1

DEMO_OBJ_ID_BI2

DEMO_OBJ_ID_BO1

DEMO_OBJ_ID_BO2

DEMO_OBJ_ID_AI1

DEMO_OBJ_ID_AI2

DEMO_OBJ_ID_AI3

DEMO_OBJ_ID_AI4

DEMO_OBJ_ID_AO1

DEMO_OBJ_ID_SC1

DEMO_OBJ_ID_CA1

DEMO_OBJ_ID_TR1

DEMO_OBJ_ID_NC1

DEMO_OBJ_ID_MAX

Definition at line 25 of file bacnetdemo.h.

5.4.2 Function Documentation

5.4.2.1 void BacnetDemoMain (void)

This is the main function of the demo application and is called by the thread-x framework.

This function open the defined common bacnet instance and the connected data link. On success the common bacnet instance will be initialized and started. These two calls to the open() below will be called by the ISDE when there are Synergy Configurator XMLs available.

Open BACnet common code.

Open up data link layer. This could be IP or MS/TP depending on the pointers in g_bacnet_data_link_instance.

At this point this is basically the user's application. This is code they will be responsible for writing. All of the BACnet objects are taken care of in a different file.

Definition at line 454 of file bacnetdemo.c.

5.5 src/demo/definitions.h File Reference

```
#include "obj_dev.h"
#include "obj_bi.h"
#include "obj_bo.h"
#include "obj_ai.h"
#include "obj_ao.h"
#include "obj_sc.h"
#include "obj_ca.h"
#include "obj_tr.h"
#include "obj_nc.h"
```

Macros

- #define DEV_DATALINK_COUNT (1)
- #define DEV_DATALINK_ETH_IP1 (192)
- #define DEV_DATALINK_ETH_IP2 (168)
- #define DEV_DATALINK_ETH_IP3 (0)
- #define DEV_DATALINK_ETH_IP4 (111)
- #define DEV_DATALINK_ETH_PORT (47808)
- #define DEV_DATALINK_ETH_PORT_ID (1)
- #define DEV_DATALINK_ETH_NET_ID (1)
- #define DEV_DATALINK_FOREIGN_IP1 (192)
- #define DEV_DATALINK_FOREIGN_IP2 (168)
- #define DEV_DATALINK_FOREIGN_IP3 (0)
- #define DEV_DATALINK_FOREIGN_IP4 (100)
- #define DEV_DATALINK_FOREIGN_PORT (47808)
- #define DEV_DATALINK_FOREIGN_INTERVAL (150)
- #define DEV_DATALINK_MSTP_PORT (3)
- #define DEV_DATALINK_MSTP_NET (33)
- #define DEV_DATALINK_MSTP_ADDRESS (12)
- #define DEVICE_DDC_PASSWORD_INIT_STRING ("s7g2-DCC")
- #define DEVICE_REINIT_PASSWORD_INIT_STRING ("s7g2-Reinit")

Functions

- CFG_WRITABLE_ADDRESS (g_server_device_address)
- CFG_WRITABLE_APDU_PROPERTIES (g_server_device_apdu_properties, OBJ_DEV_MAX_APDU_SI↩ ZE, TARGET_SUPPORTED_SEGMENTATION_TYPE, TARGET_MAX_SUPPORTED_NPDU_SEGMEN↩ TS, 1, 2000, 3000, 5, 60, 5)
- CFG_READONLY_STRING (g_server_device_dcc_password, DEVICE_DDC_PASSWORD_INIT_STRING)
- CFG_READONLY_STRING (g_server_device_reinit_password, DEVICE_REINIT_PASSWORD_INIT_ST↩ RING)
- CFG_SERVER_INIT (g_server, g_server_device, DEV_DATALINK_COUNT, g_device_data_links[0], 1)

Variables

- const BACNET_SERVER_TEMPLATE_DEVICE g_server_device
- const BACNET_SERVER_TEMPLATE_OBJECT g_obj_array [DEMO_OBJ_ID_MAX]
- BACNET_DCC_VALUE g_server_device_dcc_value = DCC_ENABLE
- BACNET_SERVER_DATALINK g_device_data_links [DEV_DATALINK_COUNT]

5.5.1 Macro Definition Documentation

5.5.1.1 #define DEV_DATALINK_COUNT (1)

Definition at line 30 of file definitions.h.

5.5.1.2 #define DEV_DATALINK_ETH_IP1 (192)

Definition at line 32 of file definitions.h.

5.5.1.3 #define DEV_DATALINK_ETH_IP2 (168)

Definition at line 33 of file definitions.h.

5.5.1.4 #define DEV_DATALINK_ETH_IP3 (0)

Definition at line 34 of file definitions.h.

5.5.1.5 #define DEV_DATALINK_ETH_IP4 (111)

Definition at line 35 of file definitions.h.

5.5.1.6 #define DEV_DATALINK_ETH_NET_ID (1)

Definition at line 38 of file definitions.h.

5.5.1.7 #define DEV_DATALINK_ETH_PORT (47808)

Definition at line 36 of file definitions.h.

5.5.1.8 #define DEV_DATALINK_ETH_PORT_ID (1)

Definition at line 37 of file definitions.h.

5.5.1.9 #define DEV_DATALINK_FOREIGN_INTERVAL (150)

Definition at line 45 of file definitions.h.

5.5.1.10 #define DEV_DATALINK_FOREIGN_IP1 (192)

Definition at line 40 of file definitions.h.

5.5.1.11 #define DEV_DATALINK_FOREIGN_IP2 (168)

Definition at line 41 of file definitions.h.

5.5.1.12 #define DEV_DATALINK_FOREIGN_IP3 (0)

Definition at line 42 of file definitions.h.

5.5.1.13 #define DEV_DATALINK_FOREIGN_IP4 (100)

Definition at line 43 of file definitions.h.

5.5.1.14 #define DEV_DATALINK_FOREIGN_PORT (47808)

Definition at line 44 of file definitions.h.

5.5.1.15 #define DEV_DATALINK_MSTP_ADDRESS (12)

Definition at line 49 of file definitions.h.

5.5.1.16 #define DEV_DATALINK_MSTP_NET (33)

Definition at line 48 of file definitions.h.

5.5.1.17 #define DEV_DATALINK_MSTP_PORT (3)

Definition at line 47 of file definitions.h.

5.5.1.18 #define DEVICE_DDC_PASSWORD_INIT_STRING ("s7g2-DCC")

Definition at line 51 of file definitions.h.

5.5.1.19 #define DEVICE_REINIT_PASSWORD_INIT_STRING ("s7g2-Reinit")

Definition at line 52 of file definitions.h.

5.5.2 Function Documentation

5.5.2.1 CFG_READONLY_STRING (g_server_device_dcc_password , DEVICE_DDC_PASSWORD_INIT_STRING)

5.5.2.2 CFG_READONLY_STRING (g_server_device_reinit_password , DEVICE_REINIT_PASSWORD_INIT_STRING)

5.5.2.3 CFG_SERVER_INIT (g_server , g_server_device , DEV_DATALINK_COUNT , g_device_data_links *[0],* 1)

5.5.2.4 CFG_WRITABLE_ADDRESS (g_server_device_address)

5.5.2.5 CFG_WRITABLE_APDU_PROPERTIES (g_server_device_apdu_properties , OBJ_DEV_MAX_APDU_SIZE , TARGET_SUPPORTED_SEGMENTATION_TYPE , TARGET_MAX_SUPPORTED_NPDU_SEGMENTS , 1 , 2000 , 3000 , 5 , 60 , 5)

5.5.3 Variable Documentation

5.5.3.1 BACNET_SERVER_DATALINK g_device_data_links[DEV_DATALINK_COUNT]

Definition at line 108 of file definitions.h.

5.5.3.2 const BACNET_SERVER_TEMPLATE_OBJECT g_obj_array[DEMO_OBJ_ID_MAX]

Initial value:

```
=
{
    CFG_OBJECT(g_obj_dev, g_server_device, g_obj_dev_cov_buffer, NULL,
      g_obj_dev_function_buffer),
    CFG_OBJECT(g_obj_bi1, g_server_device, g_obj_bi1_cov_buffer, g_obj_bi1_int_buffer,
      g_obj_bi1_function_buffer),
    CFG_OBJECT(g_obj_bi2, g_server_device, g_obj_bi2_cov_buffer, g_obj_bi2_int_buffer,
      g_obj_bi2_function_buffer),
    CFG_OBJECT(g_obj_bo1, g_server_device, NULL,                 NULL,                 NULL)
      ,
    CFG_OBJECT(g_obj_bo2, g_server_device, NULL,                 NULL,                 NULL)
      ,
    CFG_OBJECT(g_obj_ai1, g_server_device, g_obj_ai1_cov_buffer, g_obj_ai1_int_buffer, NULL)
      ,
    CFG_OBJECT(g_obj_ai2, g_server_device, g_obj_ai2_cov_buffer, g_obj_ai2_int_buffer, NULL)
      ,
    CFG_OBJECT(g_obj_ai3, g_server_device, g_obj_ai3_cov_buffer, g_obj_ai3_int_buffer, NULL)
      ,
    CFG_OBJECT(g_obj_ai4, g_server_device, g_obj_ai4_cov_buffer, g_obj_ai4_int_buffer, NULL)
      ,
    CFG_OBJECT(g_obj_ao1, g_server_device, g_obj_ao1_cov_buffer, g_obj_ao1_int_buffer, NULL)
      ,
    CFG_OBJECT(g_obj_sc1, g_server_device, NULL,                 g_obj_sc1_int_buffer,
      g_obj_sc1_function_buffer),
    CFG_OBJECT(g_obj_ca1, g_server_device, NULL,                 NULL,                 NULL)
      ,
    CFG_OBJECT(g_obj_tr1, g_server_device, NULL,                 g_obj_tr1_int_buffer,
      g_obj_tr1_function_buffer),
    CFG_OBJECT(g_obj_nc1, g_server_device, NULL,                 NULL,                 NULL)
      ,
}
```

Definition at line 59 of file definitions.h.

5.5.3.3 const BACNET_SERVER_TEMPLATE_DEVICE g_server_device

Initial value:

```
=
{
    &g_server_device_address,
    &g_server_device_dcc_value,
    &g_server_device_dcc_password,
    &g_server_device_reinit_password,
    &g_server_device_apdu_properties,
    &g_obj_array[0],

}
```

Definition at line 94 of file definitions.h.

5.5.3.4 BACNET_DCC_VALUE g_server_device_dcc_value = DCC_ENABLE

Definition at line 79 of file definitions.h.

5.6 src/demo/obj_ai.c File Reference

```
#include <string.h>
#include "obj_ai.h"
#include "obj_dev.h"
```

Functions

- void ObjAIInit (void)

  Initialize the analog input objects for this demo application.
- void ObjAIReset (void)

  Reset the analog input object for this demo application.
- void ObjAIPresentValueUpdate (BACNET_INST_NUMBER inst_number, BACNET_REAL value)

  Notifies the BACnet API that the property Present_Value has changed.

Variables

- BACNET_OBJECT_ID g_obj_dev_id
- BAC_UINT g_obj_ai1_description [ ]
- BAC_UINT g_obj_ai2_description [ ]
- BAC_UINT g_obj_ai3_description [ ]
- BAC_UINT g_obj_ai4_description [ ]
- BAC_UINT g_obj_ai1_evt_msg_txt [ ]
- BAC_UINT g_obj_ai2_evt_msg_txt [ ]
- BAC_UINT g_obj_ai3_evt_msg_txt [ ]
- BAC_UINT g_obj_ai4_evt_msg_txt [ ]
- BACNET_REAL g_obj_ai1_present_value
- BACNET_REAL g_obj_ai2_present_value
- BACNET_REAL g_obj_ai3_present_value

- BACNET_REAL g_obj_ai4_present_value
- BACNET_BOOLEAN g_obj_ai1_out_of_service
- BACNET_BOOLEAN g_obj_ai2_out_of_service
- BACNET_BOOLEAN g_obj_ai3_out_of_service
- BACNET_BOOLEAN g_obj_ai4_out_of_service
- BACNET_SHORT_BIT_STRING g_obj_ai1_state_flags
- BACNET_SHORT_BIT_STRING g_obj_ai2_state_flags
- BACNET_SHORT_BIT_STRING g_obj_ai3_state_flags
- BACNET_SHORT_BIT_STRING g_obj_ai4_state_flags

5.6.1 Function Documentation

5.6.1.1 void ObjAIInit (void)

Initialize the analog input objects for this demo application.

Used here for setting some writable texts.

Definition at line 71 of file obj_ai.c.

5.6.1.2 void ObjAIPresentValueUpdate (BACNET_INST_NUMBER *inst_number*, BACNET_REAL *value*)

Notifies the BACnet API that the property Present_Value has changed.

This demo application provide 4 analog input objects. The values for these objects where updated by 2 potentiometers and 2 temperature sensors.

Parameters

in	*inst_number*	The instance number for the analog input object.
in	*value*	The new value for the analog input object.

Definition at line 109 of file obj_ai.c.

5.6.1.3 void ObjAIReset (void)

Reset the analog input object for this demo application.

Definition at line 87 of file obj_ai.c.

5.6.2 Variable Documentation

5.6.2.1 BAC_UINT g_obj_ai1_description[ ]

5.6.2.2 BAC_UINT g_obj_ai1_evt_msg_txt[ ]

5.6.2.3 BACNET_BOOLEAN g_obj_ai1_out_of_service

5.6.2.4 BACNET_REAL g_obj_ai1_present_value

5.6.2.5 BACNET_SHORT_BIT_STRING g_obj_ai1_state_flags

5.6.2.6 BAC_UINT g_obj_ai2_description[ ]

5.6.2.7 BAC_UINT g_obj_ai2_evt_msg_txt[ ]

5.6.2.8 BACNET_BOOLEAN g_obj_ai2_out_of_service

5.6.2.9 BACNET_REAL g_obj_ai2_present_value

5.6.2.10 BACNET_SHORT_BIT_STRING g_obj_ai2_state_flags

5.6.2.11 BAC_UINT g_obj_ai3_description[ ]

5.6.2.12 BAC_UINT g_obj_ai3_evt_msg_txt[ ]

5.6.2.13 BACNET_BOOLEAN g_obj_ai3_out_of_service

5.6.2.14 BACNET_REAL g_obj_ai3_present_value

5.6.2.15 BACNET_SHORT_BIT_STRING g_obj_ai3_state_flags

5.6.2.16 BAC_UINT g_obj_ai4_description[ ]

5.6.2.17 BAC_UINT g_obj_ai4_evt_msg_txt[ ]

5.6.2.18 BACNET_BOOLEAN g_obj_ai4_out_of_service

5.6.2.19 BACNET_REAL g_obj_ai4_present_value

5.6.2.20 BACNET_SHORT_BIT_STRING g_obj_ai4_state_flags

5.6.2.21 BACNET_OBJECT_ID g_obj_dev_id

5.7 src/demo/obj_ai.h File Reference

```
#include "bacnetdemo.h"
```

Macros

- #define OBJ_AI1_NAME_INIT_STRING ("Demo - AI.1")
- #define OBJ_AI1_DESCRIPTION_INIT_STRING ("Potentiometer 1")
- #define OBJ_AI1_PRESENT_VALUE_INITIAL (50)
- #define OBJ_AI1_COV_INCREMENT_INITIAL (1)
- #define OBJ_AI1_UNIT_INITIAL (UNIT_PERCENT)
- #define OBJ_AI1_LIMIT_HIGH_INITIAL (90)
- #define OBJ_AI1_LIMIT_LOW_INITIAL (10)
- #define OBJ_AI1_LIMIT_DEADBAND_INITIAL (5)
- #define OBJ_AI2_NAME_INIT_STRING ("Demo - AI.2")
- #define OBJ_AI2_DESCRIPTION_INIT_STRING ("Potentiometer 2")
- #define OBJ_AI2_PRESENT_VALUE_INITIAL (50)
- #define OBJ_AI2_COV_INCREMENT_INITIAL (1)
- #define OBJ_AI2_UNIT_INITIAL (UNIT_PERCENT)
- #define OBJ_AI2_LIMIT_HIGH_INITIAL (90)
- #define OBJ_AI2_LIMIT_LOW_INITIAL (10)
- #define OBJ_AI2_LIMIT_DEADBAND_INITIAL (5)
- #define OBJ_AI3_NAME_INIT_STRING ("Demo - AI.3")
- #define OBJ_AI3_DESCRIPTION_INIT_STRING ("Temperature 1")
- #define OBJ_AI3_PRESENT_VALUE_INITIAL (20)
- #define OBJ_AI3_COV_INCREMENT_INITIAL (0.5)
- #define OBJ_AI3_UNIT_INITIAL (UNIT_DEGREES_C)
- #define OBJ_AI3_LIMIT_HIGH_INITIAL (50)
- #define OBJ_AI3_LIMIT_LOW_INITIAL (0)
- #define OBJ_AI3_LIMIT_DEADBAND_INITIAL (2)
- #define OBJ_AI4_NAME_INIT_STRING ("Demo - AI.4")
- #define OBJ_AI4_DESCRIPTION_INIT_STRING ("Temperature 2")
- #define OBJ_AI4_PRESENT_VALUE_INITIAL (20)
- #define OBJ_AI4_COV_INCREMENT_INITIAL (0.5)
- #define OBJ_AI4_UNIT_INITIAL (UNIT_DEGREES_C)
- #define OBJ_AI4_LIMIT_HIGH_INITIAL (50)
- #define OBJ_AI4_LIMIT_LOW_INITIAL (0)
- #define OBJ_AI4_LIMIT_DEADBAND_INITIAL (2)

Functions

- void ObjAIInit (void)

  Initialize the analog input objects for this demo application.
- void ObjAIReset (void)

  Reset the analog input object for this demo application.
- void ObjAIPresentValueUpdate (BACNET_INST_NUMBER inst_number, BACNET_REAL value)

  Notifies the BACnet API that the property Present_Value has changed.

5.7.1 Macro Definition Documentation

5.7.1.1 #define OBJ_AI1_COV_INCREMENT_INITIAL (1)

Definition at line 42 of file obj_ai.h.

5.7.1.2 #define OBJ_AI1_DESCRIPTION_INIT_STRING ("Potentiometer 1")

Definition at line 40 of file obj_ai.h.

5.7.1.3 #define OBJ_AI1_LIMIT_DEADBAND_INITIAL (5)

Definition at line 46 of file obj_ai.h.

5.7.1.4 #define OBJ_AI1_LIMIT_HIGH_INITIAL (90)

Definition at line 44 of file obj_ai.h.

5.7.1.5 #define OBJ_AI1_LIMIT_LOW_INITIAL (10)

Definition at line 45 of file obj_ai.h.

5.7.1.6 #define OBJ_AI1_NAME_INIT_STRING ("Demo - AI.1")

Definition at line 39 of file obj_ai.h.

5.7.1.7 #define OBJ_AI1_PRESENT_VALUE_INITIAL (50)

Definition at line 41 of file obj_ai.h.

5.7.1.8 #define OBJ_AI1_UNIT_INITIAL (UNIT_PERCENT)

Definition at line 43 of file obj_ai.h.

5.7.1.9 #define OBJ_AI2_COV_INCREMENT_INITIAL (1)

Definition at line 51 of file obj_ai.h.

5.7.1.10 #define OBJ_AI2_DESCRIPTION_INIT_STRING ("Potentiometer 2")

Definition at line 49 of file obj_ai.h.

5.7.1.11 #define OBJ_AI2_LIMIT_DEADBAND_INITIAL (5)

Definition at line 55 of file obj_ai.h.

5.7.1.12 #define OBJ_AI2_LIMIT_HIGH_INITIAL (90)

Definition at line 53 of file obj_ai.h.

5.7.1.13 #define OBJ_AI2_LIMIT_LOW_INITIAL (10)

Definition at line 54 of file obj_ai.h.

5.7.1.14 #define OBJ_AI2_NAME_INIT_STRING ("Demo - AI.2")

Definition at line 48 of file obj_ai.h.

5.7.1.15 #define OBJ_AI2_PRESENT_VALUE_INITIAL (50)

Definition at line 50 of file obj_ai.h.

5.7.1.16 #define OBJ_AI2_UNIT_INITIAL (UNIT_PERCENT)

Definition at line 52 of file obj_ai.h.

5.7.1.17 #define OBJ_AI3_COV_INCREMENT_INITIAL (0.5)

Definition at line 60 of file obj_ai.h.

5.7.1.18 #define OBJ_AI3_DESCRIPTION_INIT_STRING ("Temperature 1")

Definition at line 58 of file obj_ai.h.

5.7.1.19 #define OBJ_AI3_LIMIT_DEADBAND_INITIAL (2)

Definition at line 64 of file obj_ai.h.

5.7.1.20 #define OBJ_AI3_LIMIT_HIGH_INITIAL (50)

Definition at line 62 of file obj_ai.h.

5.7.1.21 #define OBJ_AI3_LIMIT_LOW_INITIAL (0)

Definition at line 63 of file obj_ai.h.

5.7.1.22 #define OBJ_AI3_NAME_INIT_STRING ("Demo - AI.3")

Definition at line 57 of file obj_ai.h.

5.7.1.23 #define OBJ_AI3_PRESENT_VALUE_INITIAL (20)

Definition at line 59 of file obj_ai.h.

5.7.1.24 #define OBJ_AI3_UNIT_INITIAL (UNIT_DEGREES_C)

Definition at line 61 of file obj_ai.h.

5.7.1.25 #define OBJ_AI4_COV_INCREMENT_INITIAL (0.5)

Definition at line 69 of file obj_ai.h.

5.7.1.26 #define OBJ_AI4_DESCRIPTION_INIT_STRING ("Temperature 2")

Definition at line 67 of file obj_ai.h.

5.7.1.27 #define OBJ_AI4_LIMIT_DEADBAND_INITIAL (2)

Definition at line 73 of file obj_ai.h.

5.7.1.28 #define OBJ_AI4_LIMIT_HIGH_INITIAL (50)

Definition at line 71 of file obj_ai.h.

5.7.1.29 #define OBJ_AI4_LIMIT_LOW_INITIAL (0)

Definition at line 72 of file obj_ai.h.

5.7.1.30 #define OBJ_AI4_NAME_INIT_STRING ("Demo - AI.4")

Definition at line 66 of file obj_ai.h.

5.7.1.31 #define OBJ_AI4_PRESENT_VALUE_INITIAL (20)

Definition at line 68 of file obj_ai.h.

5.7.1.32 #define OBJ_AI4_UNIT_INITIAL (UNIT_DEGREES_C)

Definition at line 70 of file obj_ai.h.

5.7.2 Function Documentation

5.7.2.1 void ObjAIInit (void)

Initialize the analog input objects for this demo application.

Used here for setting some writable texts.

Definition at line 71 of file obj_ai.c.

5.7.2.2 void ObjAIPresentValueUpdate (BACNET_INST_NUMBER *inst_number,* BACNET_REAL *value*)

Notifies the BACnet API that the property Present_Value has changed.

This demo application provide 4 analog input objects. The values for these objects where updated by 2 potentiometers and 2 temperature sensors.

Parameters

in	*inst_number*	The instance number for the analog input object.
in	*value*	The new value for the analog input object.

Definition at line 109 of file obj_ai.c.

5.7.2.3 void ObjAIReset (void)

Reset the analog input object for this demo application.

Definition at line 87 of file obj_ai.c.

5.8 src/demo/obj_ao.c File Reference

```
#include <string.h>
#include "obj_ao.h"
#include "user.h"
```

Functions

- void ObjAOInit (void)

  Initialize the analog output object for this demo application.
- void ObjAOReset (void)

  Reset the analog output object for this demo application.
- BACNET_CALLBACK_STATUS ObjAOWriteHandling (BACNET_INST_NUMBER inst_number, BACNET←_PROPERTY_ID property_id, const BACNET_PROPERTY_CONTENTS ∗p_property_contents)

  Notification that a property has changed for an analog output object.

Variables

- BAC_UINT g_obj_ao1_description [ ]
- BAC_UINT g_obj_ao1_evt_msg_txt [ ]
- BACNET_REAL g_obj_ao1_present_value
- BACNET_BOOLEAN g_obj_ao1_out_of_service
- BACNET_SHORT_BIT_STRING g_obj_ao1_state_flags

5.8.1 Function Documentation

5.8.1.1 void ObjAOInit (void)

Initialize the analog output object for this demo application.

Used here for setting some writable texts.

Definition at line 55 of file obj_ao.c.

5.8.1.2 void ObjAOReset (void)

Reset the analog output object for this demo application.

Definition at line 65 of file obj_ao.c.

5.8.1.3 BACNET_CALLBACK_STATUS ObjAOWriteHandling (BACNET_INST_NUMBER *inst_number*, BACNET_PROPERTY_ID *property_id*, const BACNET_PROPERTY_CONTENTS ∗ *p_property_contents*)

Notification that a property has changed for an analog output object.

For this demo application the analog output object is only created to be used by the scheduler. There is no hardware connected which is controlled by that analog output object.

Parameters

in	*inst_number*	The instance number of the analog output object.
in	*property_id*	Id of the changed property.
in	*p_property_contents*	Buffer with content of the property.

Return values

CALLBACK_STATUS_DEFAULT	Always CALLBACK_STATUS_DEFAULT for this demo application.

Definition at line 84 of file obj_ao.c.

5.8.2 Variable Documentation

5.8.2.1 BAC_UINT g_obj_ao1_description[ ]

5.8.2.2 BAC_UINT g_obj_ao1_evt_msg_txt[ ]

5.8.2.3 BACNET_BOOLEAN g_obj_ao1_out_of_service

5.8.2.4 BACNET_REAL g_obj_ao1_present_value

5.8.2.5 BACNET_SHORT_BIT_STRING g_obj_ao1_state_flags

5.9 src/demo/obj_ao.h File Reference

```
#include "bacnetdemo.h"
```

Macros

- #define OBJ_AO1_NAME_INIT_STRING ("Demo - AO.1")
- #define OBJ_AO1_DESCRIPTION_INIT_STRING ("Scheduler modified")

Functions

- void ObjAOInit (void)

  Initialize the analog output object for this demo application.
- void ObjAOReset (void)

  Reset the analog output object for this demo application.
- BACNET_CALLBACK_STATUS ObjAOWriteHandling (BACNET_INST_NUMBER inst_number, BACNET←_PROPERTY_ID property_id, const BACNET_PROPERTY_CONTENTS ∗p_property_contents)

  Notification that a property has changed for an analog output object.

5.9.1 Macro Definition Documentation

5.9.1.1 #define OBJ_AO1_DESCRIPTION_INIT_STRING ("Scheduler modified")

Definition at line 36 of file obj_ao.h.

5.9.1.2 #define OBJ_AO1_NAME_INIT_STRING ("Demo - AO.1")

Definition at line 35 of file obj_ao.h.

5.9.2 Function Documentation

5.9.2.1 void ObjAOInit (void)

Initialize the analog output object for this demo application.

Used here for setting some writable texts.

Definition at line 55 of file obj_ao.c.

5.9.2.2 void ObjAOReset (void)

Reset the analog output object for this demo application.

Definition at line 65 of file obj_ao.c.

5.9.2.3 BACNET_CALLBACK_STATUS ObjAOWriteHandling (BACNET_INST_NUMBER *inst_number,* BACNET_PROPERTY_ID *property_id,* const BACNET_PROPERTY_CONTENTS ∗ *p_property_contents*)

Notification that a property has changed for an analog output object.

For this demo application the analog output object is only created to be used by the scheduler. There is no hardware connected which is controlled by that analog output object.

Parameters

`in`	*inst_number*	The instance number of the analog output object.
`in`	*property_id*	Id of the changed property.
`in`	*p_property_contents*	Buffer with content of the property.

Return values

CALLBACK_STATUS_DEFAULT	Always CALLBACK_STATUS_DEFAULT for this demo application.

Definition at line 84 of file obj_ao.c.

5.10 src/demo/obj_bi.c File Reference

```
#include <string.h>
#include "obj_bi.h"
#include "obj_dev.h"
```

Functions

- void ObjBIInit (void)

Initialize the binary input object for this demo application. Used here for setting some writable texts.

- void ObjBIReset (void)

  Reset the binary input object for this demo application.
- void ObjBIPresentValueUpdate (BACNET_INST_NUMBER inst_number, BAC_BOOLEAN value)

  Notifies the BACnet API that the property Present_Value has changed.

Variables

- BACNET_OBJECT_ID g_obj_dev_id
- BAC_UINT g_obj_bi1_description [ ]
- BAC_UINT g_obj_bi2_description [ ]
- BAC_UINT g_obj_bi1_evt_msg_txt [ ]
- BAC_UINT g_obj_bi2_evt_msg_txt [ ]
- BACNET_ENUM g_obj_bi1_present_value
- BACNET_ENUM g_obj_bi2_present_value
- BACNET_BOOLEAN g_obj_bi1_out_of_service
- BACNET_BOOLEAN g_obj_bi2_out_of_service
- BACNET_SHORT_BIT_STRING g_obj_bi1_state_flags
- BACNET_SHORT_BIT_STRING g_obj_bi2_state_flags

5.10.1 Function Documentation

5.10.1.1 void ObjBIInit (void)

Initialize the binary input object for this demo application. Used here for setting some writable texts.

Definition at line 59 of file obj_bi.c.

5.10.1.2 void ObjBIPresentValueUpdate (BACNET_INST_NUMBER *inst_number,* BAC_BOOLEAN *value*)

Notifies the BACnet API that the property Present_Value has changed.

This demo application provide 2 binary input objects. The values for these objects where updated by 2 switches.

Parameters

`in`	*inst_number*	The instance number for the binary input object.
`in`	*value*	The new value for the binary input object.

Definition at line 89 of file obj_bi.c.

5.10.1.3 void ObjBIReset (void)

Reset the binary input object for this demo application.

Definition at line 71 of file obj_bi.c.

5.10.2 Variable Documentation

5.10.2.1 BAC_UINT g_obj_bi1_description[ ]

5.10.2.2 BAC_UINT g_obj_bi1_evt_msg_txt[ ]

5.10.2.3 BACNET_BOOLEAN g_obj_bi1_out_of_service

5.10.2.4 BACNET_ENUM g_obj_bi1_present_value

5.10.2.5 BACNET_SHORT_BIT_STRING g_obj_bi1_state_flags

5.10.2.6 BAC_UINT g_obj_bi2_description[ ]

5.10.2.7 BAC_UINT g_obj_bi2_evt_msg_txt[ ]

5.10.2.8 BACNET_BOOLEAN g_obj_bi2_out_of_service

5.10.2.9 BACNET_ENUM g_obj_bi2_present_value

5.10.2.10 BACNET_SHORT_BIT_STRING g_obj_bi2_state_flags

5.10.2.11 BACNET_OBJECT_ID g_obj_dev_id

5.11 src/demo/obj_bi.h File Reference

```
#include "bacnetdemo.h"
```

Macros

- #define OBJ_BI1_NAME_INIT_STRING ("Demo - BI.1")
- #define OBJ_BI1_DESCRIPTION_INIT_STRING ("Switch 1")
- #define OBJ_BI1_INACTIVE_INIT_STRING ("off")
- #define OBJ_BI1_ACTIVE_INIT_STRING ("on")
- #define OBJ_BI2_NAME_INIT_STRING ("Demo - BI.2")
- #define OBJ_BI2_DESCRIPTION_INIT_STRING ("Switch 2")
- #define OBJ_BI2_INACTIVE_INIT_STRING ("off")
- #define OBJ_BI2_ACTIVE_INIT_STRING ("on")

Functions

- void ObjBIInit (void)

  Initialize the binary input object for this demo application. Used here for setting some writable texts.
- void ObjBIReset (void)

  Reset the binary input object for this demo application.
- void ObjBIPresentValueUpdate (BACNET_INST_NUMBER inst_number, BAC_BOOLEAN value)

  Notifies the BACnet API that the property Present_Value has changed.

5.11.1 Macro Definition Documentation

5.11.1.1 #define OBJ_BI1_ACTIVE_INIT_STRING ("on")

Definition at line 36 of file obj_bi.h.

5.11.1.2 #define OBJ_BI1_DESCRIPTION_INIT_STRING ("Switch 1")

Definition at line 34 of file obj_bi.h.

5.11.1.3 #define OBJ_BI1_INACTIVE_INIT_STRING ("off")

Definition at line 35 of file obj_bi.h.

5.11.1.4 #define OBJ_BI1_NAME_INIT_STRING ("Demo - BI.1")

Definition at line 33 of file obj_bi.h.

5.11.1.5 #define OBJ_BI2_ACTIVE_INIT_STRING ("on")

Definition at line 41 of file obj_bi.h.

5.11.1.6 #define OBJ_BI2_DESCRIPTION_INIT_STRING ("Switch 2")

Definition at line 39 of file obj_bi.h.

5.11.1.7 #define OBJ_BI2_INACTIVE_INIT_STRING ("off")

Definition at line 40 of file obj_bi.h.

5.11.1.8 #define OBJ_BI2_NAME_INIT_STRING ("Demo - BI.2")

Definition at line 38 of file obj_bi.h.

5.11.2 Function Documentation

5.11.2.1 void ObjBIInit (void)

Initialize the binary input object for this demo application. Used here for setting some writable texts.

Definition at line 59 of file obj_bi.c.

5.11.2.2 void ObjBIPresentValueUpdate (BACNET_INST_NUMBER *inst_number,* BAC_BOOLEAN *value*)

Notifies the BACnet API that the property Present_Value has changed.

This demo application provide 2 binary input objects. The values for these objects where updated by 2 switches.

Parameters

in	*inst_number*	The instance number for the binary input object.
in	*value*	The new value for the binary input object.

Definition at line 89 of file obj_bi.c.

5.11.2.3 void ObjBIReset (void)

Reset the binary input object for this demo application.

Definition at line 71 of file obj_bi.c.

5.12 src/demo/obj_bo.c File Reference

```
#include <string.h>
#include "obj_bo.h"
#include "obj_dev.h"
#include "user.h"
```

Functions

- void ObjBOInit (void)

  Initialize the binary output object for this demo application. Used here for setting some writable texts.
- void ObjBOReset (void)

  Reset the binary output object for this demo application.
- BACNET_CALLBACK_STATUS ObjBOWriteHandling (BACNET_INST_NUMBER inst_number, BACNET↩_PROPERTY_ID property_id, const BACNET_PROPERTY_CONTENTS ∗p_property_contents)

  Notification that a property has changed for a binary output object.

Variables

- BAC_UINT g_obj_bo1_description [ ]
- BAC_UINT g_obj_bo2_description [ ]
- BACNET_ENUM g_obj_bo1_present_value
- BACNET_ENUM g_obj_bo2_present_value
- BACNET_BOOLEAN g_obj_bo1_out_of_service
- BACNET_BOOLEAN g_obj_bo2_out_of_service
- BACNET_SHORT_BIT_STRING g_obj_bo1_state_flags
- BACNET_SHORT_BIT_STRING g_obj_bo2_state_flags

5.12.1 Function Documentation

5.12.1.1 void ObjBOInit (void)

Initialize the binary output object for this demo application. Used here for setting some writable texts.

Definition at line 57 of file obj_bo.c.

5.12.1.2 void ObjBOReset (void)

Reset the binary output object for this demo application.

Definition at line 66 of file obj_bo.c.

5.12.1.3 BACNET_CALLBACK_STATUS ObjBOWriteHandling (BACNET_INST_NUMBER *inst_number*, BACNET_PROPERTY_ID *property_id*, const BACNET_PROPERTY_CONTENTS ∗ *p_property_contents*)

Notification that a property has changed for a binary output object.

For this demo application only a changed Present_Value for the binary output objects will be processed. This property controls the state of a LED. The LED will glow if the Present_Value of the corresponding binary object is on.

Parameters

`in`	*inst_number*	The instance number of the binary output object.
`in`	*property_id*	Id of the changed property.
`in`	*p_property_contents*	Buffer with content of the property.

Return values

CALLBACK_STATUS_DEFAULT	Always CALLBACK_STATUS_DEFAULT for this demo application.

Definition at line 91 of file obj_bo.c.

5.12.2 Variable Documentation

5.12.2.1 BAC_UINT g_obj_bo1_description[ ]

5.12.2.2 BACNET_BOOLEAN g_obj_bo1_out_of_service

5.12.2.3 BACNET_ENUM g_obj_bo1_present_value

5.12.2.4 BACNET_SHORT_BIT_STRING g_obj_bo1_state_flags

5.12.2.5 BAC_UINT g_obj_bo2_description[ ]

5.12.2.6 BACNET_BOOLEAN g_obj_bo2_out_of_service

5.12.2.7 BACNET_ENUM g_obj_bo2_present_value

5.12.2.8 BACNET_SHORT_BIT_STRING g_obj_bo2_state_flags

5.13 src/demo/obj_bo.h File Reference

```
#include "bacnetdemo.h"
```

Macros

- #define OBJ_BO1_NAME_INIT_STRING ("Demo - BO.1")
- #define OBJ_BO1_DESCRIPTION_INIT_STRING ("LED 1")
- #define OBJ_BO1_INACTIVE_INIT_STRING ("off")
- #define OBJ_BO1_ACTIVE_INIT_STRING ("on")
- #define OBJ_BO2_NAME_INIT_STRING ("Demo - BO.2")
- #define OBJ_BO2_DESCRIPTION_INIT_STRING ("LED 2")
- #define OBJ_BO2_INACTIVE_INIT_STRING ("off")
- #define OBJ_BO2_ACTIVE_INIT_STRING ("on")

Functions

- void ObjBOInit (void)

  Initialize the binary output object for this demo application. Used here for setting some writable texts.
- void ObjBOReset (void)

  Reset the binary output object for this demo application.
- BACNET_CALLBACK_STATUS ObjBOWriteHandling (BACNET_INST_NUMBER inst_number, BACNET↩ _PROPERTY_ID property_id, const BACNET_PROPERTY_CONTENTS ∗p_property_contents)

  Notification that a property has changed for a binary output object.

5.13.1 Macro Definition Documentation

5.13.1.1 #define OBJ_BO1_ACTIVE_INIT_STRING ("on")

Definition at line 38 of file obj_bo.h.

5.13.1.2 #define OBJ_BO1_DESCRIPTION_INIT_STRING ("LED 1")

Definition at line 36 of file obj_bo.h.

5.13.1.3 #define OBJ_BO1_INACTIVE_INIT_STRING ("off")

Definition at line 37 of file obj_bo.h.

5.13.1.4 #define OBJ_BO1_NAME_INIT_STRING ("Demo - BO.1")

Definition at line 35 of file obj_bo.h.

5.13.1.5 #define OBJ_BO2_ACTIVE_INIT_STRING ("on")

Definition at line 43 of file obj_bo.h.

5.13.1.6 #define OBJ_BO2_DESCRIPTION_INIT_STRING ("LED 2")

Definition at line 41 of file obj_bo.h.

5.13.1.7 #define OBJ_BO2_INACTIVE_INIT_STRING ("off")

Definition at line 42 of file obj_bo.h.

5.13.1.8 #define OBJ_BO2_NAME_INIT_STRING ("Demo - BO.2")

Definition at line 40 of file obj_bo.h.

5.13.2 Function Documentation

5.13.2.1 void ObjBOInit (void)

Initialize the binary output object for this demo application. Used here for setting some writable texts.

Definition at line 57 of file obj_bo.c.

5.13.2.2 void ObjBOReset (void)

Reset the binary output object for this demo application.

Definition at line 66 of file obj_bo.c.

5.13.2.3 BACNET_CALLBACK_STATUS ObjBOWriteHandling (BACNET_INST_NUMBER *inst_number,* BACNET_PROPERTY_ID *property_id,* const BACNET_PROPERTY_CONTENTS ∗ *p_property_contents*)

Notification that a property has changed for a binary output object.

For this demo application only a changed Present_Value for the binary output objects will be processed. This property controls the state of a LED. The LED will glow if the Present_Value of the corresponding binary object is on.

Parameters

`in`	*inst_number*	The instance number of the binary output object.
`in`	*property_id*	Id of the changed property.
`in`	*p_property_contents*	Buffer with content of the property.

Return values

CALLBACK_STATUS_DEFAULT	Always CALLBACK_STATUS_DEFAULT for this demo application.

Definition at line 91 of file obj_bo.c.

5.14 src/demo/obj_ca.c File Reference

```
#include <stdbool.h>
#include <stdint.h>
#include <string.h>
#include "bacnet.h"
#include "obj_ca.h"
#include "obj_dev.h"
```

Functions

- void ObjCAInit (void)

    Initialize the schedule object for this demo application. Used here for setting some writable texts.

Variables

- BAC_UINT g_obj_ca1_description [ ]

5.14.1 Function Documentation

5.14.1.1 void ObjCAInit (void)

Initialize the schedule object for this demo application. Used here for setting some writable texts.

Definition at line 56 of file obj_ca.c.

5.14.2 Variable Documentation

5.14.2.1 BAC_UINT g_obj_ca1_description[ ]

5.15 src/demo/obj_ca.h File Reference

```
#include "bacnetdemo.h"
```

Macros

- #define OBJ_CA1_NAME_INIT_STRING ("Demo - CA.1")
- #define OBJ_CA1_DESCRIPTION_INIT_STRING ("Calendar for the scheduler")

Functions

- void ObjCAInit (void)

    Initialize the schedule object for this demo application. Used here for setting some writable texts.

5.15.1 Macro Definition Documentation

5.15.1.1 #define OBJ_CA1_DESCRIPTION_INIT_STRING ("Calendar for the scheduler")

Definition at line 33 of file obj_ca.h.

5.15.1.2 #define OBJ_CA1_NAME_INIT_STRING ("Demo - CA.1")

Definition at line 32 of file obj_ca.h.

5.15.2 Function Documentation

5.15.2.1 void ObjCAInit (void)

Initialize the schedule object for this demo application. Used here for setting some writable texts.

Definition at line 56 of file obj_ca.c.

5.16 src/demo/obj_dev.c File Reference

```
#include <string.h>
#include <stdio.h>
#include "obj_dev.h"
```

Functions

- void ObjDevInit (BACNET_TIME_STAMP ∗p_start_time)

    Initialize the device object for this demo application. Used here for setting the instance number of the device and some writable texts.

Variables

- BACNET_OBJECT_ID g_obj_dev_id
- BAC_UINT g_obj_dev_name [ ]
- BAC_UINT g_obj_dev_description [ ]
- BAC_UINT g_obj_dev_location [ ]
- BACNET_TIME_STAMP g_obj_dev_time_of_restart [ ]

5.16.1 Function Documentation

5.16.1.1 void ObjDevInit (BACNET_TIME_STAMP ∗ *p_start_time*)

Initialize the device object for this demo application. Used here for setting the instance number of the device and some writable texts.

Definition at line 52 of file obj_dev.c.

5.16.2 Variable Documentation

5.16.2.1 BAC_UINT g_obj_dev_description[ ]

5.16.2.2 BACNET_OBJECT_ID g_obj_dev_id

5.16.2.3 BAC_UINT g_obj_dev_location[ ]

5.16.2.4 BAC_UINT g_obj_dev_name[ ]

5.16.2.5 BACNET_TIME_STAMP g_obj_dev_time_of_restart[ ]

5.17 src/demo/obj_dev.h File Reference

```
#include "bacnetdemo.h"
```

Macros

- #define OBJ_DEV_INST_NUMBER (2000)
- #define OBJ_DEV_NAME_INIT_STRING ("Demo - Device")
- #define OBJ_DEV_DESCRIPTION_INIT_STRING ("BACnet API for Renesas Synergy™")
- #define OBJ_DEV_LOCATION_INIT_STRING ("Demo - Board")
- #define OBJ_DEV_VENDOR_NAME_STRING ("CS Lab GmbH")
- #define OBJ_DEV_VENDOR_ID_NUMBER (794)
- #define OBJ_DEV_MODEL_NAME_STRING ("Demo Renesas Synergy™")
- #define OBJ_DEV_FIRMWARE_REV_STRING ("FW 1.0.0.0")
- #define OBJ_DEV_SOFTWARE_VER_STRING ("SW 1.0.0.0")
- #define OBJ_DEV_MAX_APDU_SIZE (1024)

Functions

- void ObjDevInit (BACNET_TIME_STAMP ∗p_start_time)

  Initialize the device object for this demo application. Used here for setting the instance number of the device and some writable texts.

5.17.1 Macro Definition Documentation

5.17.1.1 #define OBJ_DEV_DESCRIPTION_INIT_STRING ("BACnet API for Renesas Synergy™")

Definition at line 25 of file obj_dev.h.

5.17.1.2 #define OBJ_DEV_FIRMWARE_REV_STRING ("FW 1.0.0.0")

Definition at line 30 of file obj_dev.h.

5.17.1.3 #define OBJ_DEV_INST_NUMBER (2000)

Definition at line 22 of file obj_dev.h.

5.17.1.4 #define OBJ_DEV_LOCATION_INIT_STRING ("Demo - Board")

Definition at line 26 of file obj_dev.h.

5.17.1.5 #define OBJ_DEV_MAX_APDU_SIZE (1024)

Definition at line 33 of file obj_dev.h.

5.17.1.6 #define OBJ_DEV_MODEL_NAME_STRING ("Demo Renesas Synergy™")

Definition at line 29 of file obj_dev.h.

5.17.1.7 #define OBJ_DEV_NAME_INIT_STRING ("Demo - Device")

Definition at line 24 of file obj_dev.h.

5.17.1.8 #define OBJ_DEV_SOFTWARE_VER_STRING ("SW 1.0.0.0")

Definition at line 31 of file obj_dev.h.

5.17.1.9 #define OBJ_DEV_VENDOR_ID_NUMBER (794)

Definition at line 28 of file obj_dev.h.

5.17.1.10 #define OBJ_DEV_VENDOR_NAME_STRING ("CS Lab GmbH")

Definition at line 27 of file obj_dev.h.

5.17.2 Function Documentation

5.17.2.1 void ObjDevInit (BACNET_TIME_STAMP ∗ *p_start_time*)

Initialize the device object for this demo application. Used here for setting the instance number of the device and some writable texts.

Definition at line 52 of file obj_dev.c.

5.18 src/demo/obj_nc.c File Reference

```
#include <stdbool.h>
#include <stdint.h>
#include <string.h>
#include "bacnet.h"
#include "obj_nc.h"
#include "obj_dev.h"
```

Functions

- void ObjNCInit (void)

    Initialize the schedule object for this demo application. Used here for setting some writable texts.

Variables

- BAC_UINT g_obj_nc1_description [ ]

5.18.1 Function Documentation

5.18.1.1 void ObjNCInit (void)

Initialize the schedule object for this demo application. Used here for setting some writable texts.

Definition at line 56 of file obj_nc.c.

5.18.2 Variable Documentation

5.18.2.1 BAC_UINT g_obj_nc1_description[ ]

5.19 src/demo/obj_nc.h File Reference

```
#include "bacnetdemo.h"
```

Macros

- #define OBJ_NC1_NAME_INIT_STRING ("Demo - NC.1")
- #define OBJ_NC1_DESCRIPTION_INIT_STRING ("Notification class")

Functions

- void ObjNCInit (void)

  Initialize the schedule object for this demo application. Used here for setting some writable texts.

5.19.1 Macro Definition Documentation

5.19.1.1 #define OBJ_NC1_DESCRIPTION_INIT_STRING ("Notification class")

Definition at line 35 of file obj_nc.h.

5.19.1.2 #define OBJ_NC1_NAME_INIT_STRING ("Demo - NC.1")

Definition at line 34 of file obj_nc.h.

5.19.2 Function Documentation

5.19.2.1 void ObjNCInit (void)

Initialize the schedule object for this demo application. Used here for setting some writable texts.

Definition at line 56 of file obj_nc.c.

5.20 src/demo/obj_sc.c File Reference

```
#include <stdbool.h>
#include <stdint.h>
#include <string.h>
#include "bacnet.h"
#include "obj_sc.h"
#include "obj_dev.h"
```

Functions

- void ObjSCInit (void)

  Initialize the schedule object for this demo application. Used here for setting some writable texts.
- void ObjSCReset (void)

  Reset the schedule object for this demo application.

Variables

- BAC_UINT g_obj_sc1_description [ ]
- BACNET_BOOLEAN g_obj_sc1_out_of_service
- BACNET_SHORT_BIT_STRING g_obj_sc1_state_flags

5.20.1 Function Documentation

5.20.1.1 void ObjSCInit (void)

Initialize the schedule object for this demo application. Used here for setting some writable texts.

Definition at line 58 of file obj_sc.c.

5.20.1.2 void ObjSCReset (void)

Reset the schedule object for this demo application.

Definition at line 66 of file obj_sc.c.

5.20.2 Variable Documentation

5.20.2.1 BAC_UINT g_obj_sc1_description[ ]

5.20.2.2 BACNET_BOOLEAN g_obj_sc1_out_of_service

5.20.2.3 BACNET_SHORT_BIT_STRING g_obj_sc1_state_flags

5.21 src/demo/obj_sc.h File Reference

```
#include "bacnetdemo.h"
```

Macros

- #define OBJ_SC1_DAILY_SCHEDULE_SIZE (24)
- #define OBJ_SC1_NAME_INIT_STRING ("Demo - SC.1")
- #define OBJ_SC1_DESCRIPTION_INIT_STRING ("Every full hour")

Functions

- void ObjSCInit (void)

  Initialize the schedule object for this demo application. Used here for setting some writable texts.
- void ObjSCReset (void)

  Reset the schedule object for this demo application.

5.21.1 Macro Definition Documentation

5.21.1.1 #define OBJ_SC1_DAILY_SCHEDULE_SIZE (24)

Definition at line 38 of file obj_sc.h.

5.21.1.2 #define OBJ_SC1_DESCRIPTION_INIT_STRING ("Every full hour")

Definition at line 41 of file obj_sc.h.

5.21.1.3 #define OBJ_SC1_NAME_INIT_STRING ("Demo - SC.1")

Definition at line 40 of file obj_sc.h.

5.21.2 Function Documentation

5.21.2.1 void ObjSCInit (void)

Initialize the schedule object for this demo application. Used here for setting some writable texts.

Definition at line 58 of file obj_sc.c.

5.21.2.2 void ObjSCReset (void)

Reset the schedule object for this demo application.

Definition at line 66 of file obj_sc.c.

5.22 src/demo/obj_tr.c File Reference

```
#include <stdbool.h>
#include <stdint.h>
#include <string.h>
#include "bacnet.h"
#include "obj_tr.h"
#include "obj_dev.h"
#include "user.h"
```

Functions

- void ObjTRInit (void)

    Initialize the trend log object for this demo application. Used here for setting some writable texts.
- BACNET_CALLBACK_STATUS ObjTRWriteHandling (BACNET_INST_NUMBER inst_number, BACNET←_PROPERTY_ID property_id, const BACNET_PROPERTY_CONTENTS ∗p_property_contents)

    Notification that a property has changed for a trend log object.

Variables

- BAC_UINT g_obj_tr1_description [ ]
- BAC_UINT g_obj_tr1_evt_msg_txt [ ]

5.22.1 Function Documentation

5.22.1.1 void ObjTRInit (void)

Initialize the trend log object for this demo application. Used here for setting some writable texts.

Definition at line 58 of file obj_tr.c.

5.22.1.2 BACNET_CALLBACK_STATUS ObjTRWriteHandling (BACNET_INST_NUMBER *inst_number,* BACNET_PROPERTY_ID *property_id,* const BACNET_PROPERTY_CONTENTS ∗ *p_property_contents*)

Notification that a property has changed for a trend log object.

The BACnet API is only an interface, the data handling must be done on user side.

Parameters

in	*inst_number*	The instance number of the trend log object.
in	*property_id*	Id of the changed property.
in	*p_property_contents*	Buffer with content of the property.

Return values

CALLBACK_STATUS_DEFAULT	Always CALLBACK_STATUS_DEFAULT for this demo application.

Definition at line 76 of file obj_tr.c.

5.22.2 Variable Documentation

5.22.2.1 BAC_UINT g_obj_tr1_description[ ]

5.22.2.2 BAC_UINT g_obj_tr1_evt_msg_txt[ ]

5.23 src/demo/obj_tr.h File Reference

```
#include "bacnetdemo.h"
```

Macros

- #define OBJ_TR1_NAME_INIT_STRING ("Demo - TR.1")
- #define OBJ_TR1_DESCRIPTION_INIT_STRING ("Trend for AI.1")
- #define OBJ_TR1_LOG_BUFFER_SIZE (20)

Functions

- void ObjTRInit (void)

  Initialize the trend log object for this demo application. Used here for setting some writable texts.
- BACNET_CALLBACK_STATUS ObjTRWriteHandling (BACNET_INST_NUMBER inst_number, BACNET↩ _PROPERTY_ID property_id, const BACNET_PROPERTY_CONTENTS ∗p_property_contents)

  Notification that a property has changed for a trend log object.

5.23.1 Macro Definition Documentation

5.23.1.1 #define OBJ_TR1_DESCRIPTION_INIT_STRING ("Trend for AI.1")

Definition at line 40 of file obj_tr.h.

5.23.1.2 #define OBJ_TR1_LOG_BUFFER_SIZE (20)

Definition at line 41 of file obj_tr.h.

5.23.1.3 #define OBJ_TR1_NAME_INIT_STRING ("Demo - TR.1")

Definition at line 39 of file obj_tr.h.

5.23.2 Function Documentation

5.23.2.1 void ObjTRInit (void)

Initialize the trend log object for this demo application. Used here for setting some writable texts.

Definition at line 58 of file obj_tr.c.

5.23.2.2 BACNET_CALLBACK_STATUS ObjTRWriteHandling (BACNET_INST_NUMBER *inst_number,* BACNET_PROPERTY_ID *property_id,* const BACNET_PROPERTY_CONTENTS ∗ *p_property_contents*)

Notification that a property has changed for a trend log object.

The BACnet API is only an interface, the data handling must be done on user side.

Parameters

in	*inst_number*	The instance number of the trend log object.
in	*property_id*	Id of the changed property.
in	*p_property_contents*	Buffer with content of the property.

Return values

CALLBACK_STATUS_DEFAULT	Always CALLBACK_STATUS_DEFAULT for this demo application.

Definition at line 76 of file obj_tr.c.

5.24 src/demo/user.c File Reference

```
#include <math.h>
#include "tx_api.h"
#include "hal_data.h"
#include "adc.h"
#include "user.h"
#include "obj_bi.h"
#include "obj_bo.h"
#include "obj_ai.h"
#include "obj_ao.h"
#include "obj_sc.h"
#include "obj_dev.h"
```

Data Structures

- struct user_button_t
- struct user_binary_input_t
- struct user_analog_input_t
- struct user_trendlog_t

Macros

- #define USER_MAX_BINARY (2)
- #define USER_MAX_ANALOG (4)
- #define USER_MAX_RECORDS (20)
- #define USER_MIN_VALUE_DIFF (0.01)
- #define USER_BINARY_VALUE_IS_BUTTON (false)
- #define USER_ANALOG_AVERAGE_COUNT (50)
- #define FLASH_DEVICE_PAGE_SIZE (64)
- #define FLASH_DEVICE_FLASH_SIZE (FLASH_DEVICE_PAGE_SIZE ∗ 1024)
- #define FLASH_DEVICE_START_ADDRESS (0x40100000)
- #define FLASH_MEM_MAGIC_SIZE (4)

Functions

- void user_loop (ULONG entry_input)

  Checking the state of the input devices (Switches, potentiometers and temperature sensors).
- static bool show_led (uint32_t index, bool on_off)

  Setting the state of a LED.
- static bool button_pressed (uint8_t index)

  Checking the state of a button.
- static bool get_switch_state (uint8_t index, bool ∗p_switch_state)

  Checking the state of a switch.
- static bool analog_value_changed (uint8_t index, uint16_t adc_value)

  Calculate the average value for an analog input and compare it with the current stored value.
- static float calc_analog_value (uint8_t index, uint16_t adc_value, float resolution)

  Calculate the float value for an ADC value.
- static void init_scheduler (void)

  Initialize the scheduler with user defined settings.
- static void init_trendlog (void)

  Initialize the structure for the trend log data.
- static bool is_trendlog_index_valid (uint32_t index)

  Check if the entry with the given index is valid or not. True for all entries up to the array limit execpt the array is still not completed.
- static uint32_t get_trendlog_index_next (uint32_t index)

  Getting the next index in the trend log array. Looping between 0 and array limit.
- static uint32_t get_trendlog_index_prev (uint32_t index)

  Getting the previous index in the trend log array. Looping between 0 and array limit.
- static uint32_t get_trendlog_index_start (void)

  Getting the start index for the trend log array. Is 0 for a not completed array. After completion the start index is running between 0 and array limit.
- static bool is_trendlog_time_smaller (BACNET_DATE_TIME record_time, BACNET_DATE_TIME end_time, bool or_equal)

  Compare 2 BACNET_DATE_TIME's.
- static bool copy_trendlog_data (uint32_t index, uint32_t count, BACNET_LOG_RECORD p_records[ ], uint32_t ∗p_count)

  Copy data from internal memory to the given array p_records.
- static void get_trendlog_first_last (BACNET_LOG_RECORD p_records[ ], uint32_t count, bool ∗p_first, bool ∗p_last)

  Check if the trend log array contains the first and last record.
- static void user_reset_values (VOID)

  Set the user variables to initial state.
- static bool is_property_persistent (const BACNET_SERVER_PROPERTY_INSTANCE ∗p_prop)

  User decision it the given property should be stored or not.
- static BAC_BYTE get_crc_checksum (BAC_BYTE check, BAC_BYTE memory)

  Calculate the crc checksum for the stored data.
- static void flash_wait (void)

  Wait until flash is ready for next operation.
- static bool flash_write (uint32_t flash_address, uint32_t source_address)

  Write data to the flash memory.
- static bool is_flash_magic_activated (void)

  Check if the flash magic token is set or not.
- static void set_flash_magic (void)

  Set the flash magic token.
- static bool flash_write_page (BAC_BYTE ∗p_buffer, BAC_WORD buffer_size)

Function which writes a page of changed persistent property data.

- bool UserInit (void)

  Initialize the user interface.

- void UserReset (void)

  Reset the user interface.

- bool UserSetLed (uint32_t number, bool on_off)

  Setting the state of the LED for the binary output objects present value.

- bool UserTrendlogClearBuffer (uint32_t number)

  Clears the internal memory to store the trend log records.

- bool UserTrendlogPutRecord (uint32_t number, BACNET_LOG_RECORD ∗p_record)

  Stores a record in the internal memory for a trend log object.

- bool UserTrendlogGetRecords (uint32_t number, BACNET_LOG_RECORD p_records[ ], uint32_t ∗p_count, bool ∗p_first, bool ∗p_last, bool ∗p_more)

  Getting all stored records for a trend log object.

- bool UserTrendlogGetRecordsByPosition (uint32_t number, uint32_t position, int32_t count, BACNET_LO↩ G_RECORD p_records[ ], uint32_t ∗p_count, bool ∗p_first, bool ∗p_last, bool ∗p_more)

  Getting stored records for a trend log object. Beginning with the given record position and limited by the given count.

- bool UserUserTrendlogGetRecordsBySequence (uint32_t number, uint32_t sequence, int32_t count, BAC↩ NET_LOG_RECORD p_records[ ], uint32_t ∗p_count, bool ∗p_first, bool ∗p_last, bool ∗p_more)

  Getting stored records for a trend log object. Beginning with the record with the given sequence number and limited by the given count.

- bool UserTrendlogGetRecordsByTime (uint32_t number, BACNET_DATE_TIME start_time, int32_t count, BACNET_LOG_RECORD p_records[ ], uint32_t ∗p_count, bool ∗p_first, bool ∗p_last, bool ∗p_more)

  Getting stored records for a trend log object. Beginning with the record with a time stamp greater or equal to the given start_time and limited by the given count.

- bool UserTrendlogGetRecordsByTimeRange (uint32_t number, BACNET_DATE_TIME start_time, BACN↩ ET_DATE_TIME end_time, BACNET_LOG_RECORD p_records[ ], uint32_t ∗p_count, bool ∗p_first, bool ∗p_last, bool ∗p_more)

  Getting stored records for a trend log object. Beginning with the record with a time stamp greater or equal to the given start_time and limited by the given end_time.

- bool UserStorageIsPropertyPersistant (BACNET_INST_NUMBER device_id, BACNET_OBJECT_TYPE obj_type, BACNET_INST_NUMBER inst_number, BACNET_PROPERTY_ID property_id)

  Request if the given property should the stored persistent.

- bool UserStorageRead (const BACNET_SRVR_INIT ∗p_server)

  Function which reads out the data stored in the flash pages and copies them into the object property structures.

- bool UserStorageWrite (const BACNET_SRVR_INIT ∗p_server)

  Function which writes changed persistent property data.

- bool UserStorageDelete (const BACNET_SRVR_INIT ∗p_server)

  Request for deleting the persistent stored data.

Variables

- BAC_UINT g_obj_sc1_daily_schedule [CALC_WRITABLE_DAILY_SCHEDULE_ARRAY_SIZE(OBJ_SC1↩ _DAILY_SCHEDULE_SIZE)]
- BACNET_DEV_OBJ_PROP_REFERENCE g_obj_sc1_obj_prop_ref_list [ ]
- BAC_UINT g_obj_sc1_obj_prop_ref_listCount
- static user_button_t g_user_button [2]
- static const ioport_port_pin_t g_user_led_pins [ ]
- static const bsp_leds_t g_user_leds
- static user_binary_input_t g_user_binary_values [USER_MAX_BINARY]
- static user_analog_input_t g_user_analog_values [USER_MAX_ANALOG]
- static user_trendlog_t g_user_trendlog

- static uint32_t g_sequence
- static TX_THREAD g_user_thread
- static uint8_t g_user_thread_stack [0x1800]
- static TX_MUTEX g_user_mutex
- static BAC_BYTE g_flash_mem_page [FLASH_DEVICE_PAGE_SIZE]
- static BAC_BYTE ∗ gp_flash_mem = (BAC_BYTE ∗)FLASH_DEVICE_START_ADDRESS
- static BAC_BYTE g_flash_magic [FLASH_MEM_MAGIC_SIZE] = {0xEF, 0xBE, 0xC0, 0xBA}
- static BAC_WORD g_flash_offset
- static BAC_BYTE g_flash_checksum
- static BAC_WORD g_page_offset
- static BAC_BOOLEAN g_page_is_new
- static BAC_BOOLEAN g_page_data_changed
- static BAC_BOOLEAN g_page_any_page_changed

5.24.1 Macro Definition Documentation

5.24.1.1 #define FLASH_DEVICE_FLASH_SIZE (FLASH_DEVICE_PAGE_SIZE ∗ 1024)

Definition at line 62 of file user.c.

5.24.1.2 #define FLASH_DEVICE_PAGE_SIZE (64)

Definition at line 60 of file user.c.

5.24.1.3 #define FLASH_DEVICE_START_ADDRESS (0x40100000)

Definition at line 64 of file user.c.

5.24.1.4 #define FLASH_MEM_MAGIC_SIZE (4)

Definition at line 66 of file user.c.

5.24.1.5 #define USER_ANALOG_AVERAGE_COUNT (50)

Definition at line 57 of file user.c.

5.24.1.6 #define USER_BINARY_VALUE_IS_BUTTON (false)

Definition at line 52 of file user.c.

5.24.1.7 #define USER_MAX_ANALOG (4)

Definition at line 39 of file user.c.

5.24.1.8 #define USER_MAX_BINARY (2)

Definition at line 38 of file user.c.

5.24.1.9 #define USER_MAX_RECORDS (20)

Definition at line 40 of file user.c.

5.24.1.10 #define USER_MIN_VALUE_DIFF (0.01)

Definition at line 41 of file user.c.

5.24.2 Function Documentation

5.24.2.1 static bool analog_value_changed (uint8_t *index*, uint16_t *adc_value*) `[static]`

Calculate the average value for an analog input and compare it with the current stored value.

Parameters

`in`	*index*	Index for the ADC value.
`in`	*adc_value*	New measured value for the analog input.

Returns

True, if there is a new calculated value for the analog input.

Definition at line 259 of file user.c.

5.24.2.2 static bool button_pressed (uint8_t *index*) `[static]`

Checking the state of a button.

Parameters

`in`	*index*	Index of the button.

Return values

true	Button state changed from unpressed to pressed pressed.
false	Button was released or is still pressed.

Definition at line 189 of file user.c.

5.24.2.3 static float calc_analog_value (uint8_t *index*, uint16_t *adc_value*, float *resolution*) [static]

Calculate the float value for an ADC value.

A ADC value is a 12 bit value between 0 and 4095.

Parameters

in	*index*	Index for the ADC value.
in	*adc_value*	ADC value from ADC Interface.
in	*resolution*	Resolution of the ADC value.

Return values

true	Button state changed from released to pressed.
false	Button was released or is still pressed.

Definition at line 301 of file user.c.

5.24.2.4 static bool copy_trendlog_data (uint32_t *index*, uint32_t *count*, BACNET_LOG_RECORD *p_records[ ]*, uint32_t ∗ *p_count*) [static]

Copy data from internal memory to the given array p_records.

Parameters

in	*index*	Start index in the trend log array.
in	*count*	Maximum count of entries to copy.
out	*p_records*	Array which hold the returned records.
out	*p_count*	Number of records really copied.

Returns

Returns true if the records could be copied from internal memory to the array p_records.

Definition at line 575 of file user.c.

5.24.2.5 static void flash_wait (void) [static]

Wait until flash is ready for next operation.

Definition at line 772 of file user.c.

5.24.2.6 static bool flash_write (uint32_t *flash_address*, uint32_t *source_address*) [static]

Write data to the flash memory.

Parameters

in	*flash_address*	Address of the flash memory.
in	*source_address*	Address of the source.

Returns

Returns true if the data could be written to the flash memory.

Definition at line 795 of file user.c.

5.24.2.7 static bool flash_write_page (BAC_BYTE ∗ *p_buffer,* BAC_WORD *buffer_size*) `[static]`

Function which writes a page of changed persistent property data.

Parameters

in	*p_buffer*	Pointer to the data which should be stored.
in	*buffer_size*	Size of the data.

Returns

Returns true if the data could be written to the flash memory.

Definition at line 874 of file user.c.

5.24.2.8 static BAC_BYTE get_crc_checksum (BAC_BYTE *check,* BAC_BYTE *memory*) `[static]`

Calculate the crc checksum for the stored data.

Parameters

in	*check*	Current checksum.
in	*memory*	New byte in memory.

Returns

Returns the checksum for the input parameter.

Definition at line 756 of file user.c.

5.24.2.9 static bool get_switch_state (uint8_t *index,* bool ∗ *p_switch_state*) `[static]`

Checking the state of a switch.

Parameters

in	*index*	Index of the switch.
out	*p_switch_state*	Switch setting.

Returns

Switch setting has changed or not.

Definition at line 230 of file user.c.

5.24.2.10 static void get_trendlog_first_last (BACNET_LOG_RECORD *p_records[ ]*, uint32_t *count*, bool ∗ *p_first*, bool ∗ *p_last*) [static]

Check if the trend log array contains the first and last record.

Parameters

in	*p_records*	Array with trend log records.
in	*count*	Maximum count of entries in the trend log array.
out	*p_first*	Flag if the first record is in the trend log array.
out	*p_last*	Flag if the last record is in the trend log array.

Definition at line 616 of file user.c.

5.24.2.11 static uint32_t get_trendlog_index_next (uint32_t *index*) [static]

Getting the next index in the trend log array. Looping between 0 and array limit.

Parameters

in	*index*	Index in the trend log array.

Returns

Index for the next entry.

Definition at line 438 of file user.c.

5.24.2.12 static uint32_t get_trendlog_index_prev (uint32_t *index*) [static]

Getting the previous index in the trend log array. Looping between 0 and array limit.

Parameters

in	*index*	Index in the trend log array.

Returns

Index for the previous entry.

Definition at line 453 of file user.c.

5.24.2.13 static uint32_t get_trendlog_index_start (void) [static]

Getting the start index for the trend log array. Is 0 for a not completed array. After completion the start index is running between 0 and array limit.

Returns

Index for the first entry.

Definition at line 467 of file user.c.

5.24.2.14 static void init_scheduler (void) [static]

Initialize the scheduler with user defined settings.

Definition at line 355 of file user.c.

5.24.2.15 static void init_trendlog (void) [static]

Initialize the structure for the trend log data.

Definition at line 400 of file user.c.

5.24.2.16 static bool is_flash_magic_activated (void) [static]

Check if the flash magic token is set or not.

Returns

Returns true if the flash magic token is set.

Definition at line 836 of file user.c.

5.24.2.17 static bool is_property_persistent (const BACNET_SERVER_PROPERTY_INSTANCE ∗ *p_prop*) [static]

User decision it the given property should be stored or not.

Parameters

in	*p_prop*	Pointer to property.

Returns

Returns true if the property should be stored.

Definition at line 714 of file user.c.

5.24.2.18 static bool is_trendlog_index_valid (uint32_t *index*) [static]

Check if the entry with the given index is valid or not. True for all entries up to the array limit execpt the array is still not completed.

Parameters

in	*index*	Index in the trend log array.

Returns

The trend log array item can be used or not.

Definition at line 415 of file user.c.

5.24.2.19 static bool is_trendlog_time_smaller (BACNET_DATE_TIME *record_time,* BACNET_DATE_TIME *end_time,* bool *or_equal*) [static]

Compare 2 BACNET_DATE_TIME's.

Parameters

in	*record_time*	Time of a record.
in	*end_time*	Time to compare with.
in	*or_equal*	Equality is allowed or not.

Returns

Returns true for record times which are before the end_time and if the equal flag is set true will be returned if the times are equal.

Definition at line 482 of file user.c.

5.24.2.20 static void set_flash_magic (void) [static]

Set the flash magic token.

Definition at line 855 of file user.c.

5.24.2.21 static bool show_led (uint32_t *index,* bool *on_off*) [static]

Setting the state of a LED.

Parameters

`in`	*index*	Index of the LED.
`in`	*on_off*	State of the LED (on or off).

Returns

Successfully changed state of the LED.

Definition at line 164 of file user.c.

5.24.2.22 void user_loop (ULONG *entry_input*)

Checking the state of the input devices (Switches, potentiometers and temperature sensors).

Note

This function must be called by the main loop.

Sleep for 1/100 second

Definition at line 980 of file user.c.

5.24.2.23 static void user_reset_values (VOID) `[static]`

Set the user variables to initial state.

Definition at line 665 of file user.c.

5.24.2.24 bool UserInit (void)

Initialize the user interface.

Set the initial values for: binary input objects (Switches) analog input objects (Potentiometers and temperature sensors) schedule object (Every full hour from Monday to Sunday) trend log object (Internal array to store the data given and requested by the BACnet API)

Return values

true	Always true for this demo application.

Definition at line 1064 of file user.c.

5.24.2.25 void UserReset (void)

Reset the user interface.

Set the user variables and device state to initial state

Definition at line 1097 of file user.c.

5.24.2.26 bool UserSetLed (uint32_t *number,* bool *on_off*)

Setting the state of the LED for the binary output objects present value.

Parameters

in	*number*	Number of the LED (ID of binary output object).
in	*on_off*	New State of the LED (Present value of binary output object).

Returns

Successfully changed state of the LED.

Definition at line 1112 of file user.c.

5.24.2.27 bool UserStorageDelete (const BACNET_SRVR_INIT ∗ *p_server*)

Request for deleting the persistent stored data.

Parameters

in	*p_server*	Pointer to the server structure of the demo application.

Returns

Successful deleting of the stored data or not.

Definition at line 1789 of file user.c.

5.24.2.28 bool UserStorageIsPropertyPersistant (BACNET_INST_NUMBER *device_id,* BACNET_OBJECT_TYPE *obj_type,* BACNET_INST_NUMBER *inst_number,* BACNET_PROPERTY_ID *property_id*)

Request if the given property should the stored persistent.

Parameters

in	*device_id*	Device number of the property.
in	*obj_type*	Object type of the property.
in	*inst_number*	Object instance of the property.
in	*property_id*	Property id of the property.

Returns

User decision if the requested should be stored or not.

Definition at line 1543 of file user.c.

5.24.2.29 bool UserStorageRead (const BACNET_SRVR_INIT ∗ *p_server*)

Function which reads out the data stored in the flash pages and copies them into the object property structures.

Parameters

in	*p_server*	Pointer to the server structure of the demo application.

Returns

Successful reading of the stored data or not.

Definition at line 1572 of file user.c.

5.24.2.30 bool UserStorageWrite (const BACNET_SRVR_INIT ∗ *p_server*)

Function which writes changed persistent property data.

Parameters

in	*p_server*	Pointer to the server structure of the demo application.

Returns

Successful writing of the stored data or not.

Definition at line 1702 of file user.c.

5.24.2.31 bool UserTrendlogClearBuffer (uint32_t *number*)

Clears the internal memory to store the trend log records.

Parameters

in	*number*	Number of the trend log object.

Returns

For this demo application only true for a trend log object with number 1.

Definition at line 1127 of file user.c.

5.24.2.32 bool UserTrendlogGetRecords (uint32_t *number,* BACNET_LOG_RECORD *p_records[ ],* uint32_t ∗ *p_count,* bool ∗ *p_first,* bool ∗ *p_last,* bool ∗ *p_more*)

Getting all stored records for a trend log object.

Parameters

in	*number*	Number of the trend log object.
out	*p_records*	Pointer to an array to store the record data.
out	*p_count*	Number of returned records.

Returns

For this demo application only true for a trend log object with number 1 and a valid index.

Definition at line 1180 of file user.c.

5.24.2.33 bool UserTrendlogGetRecordsByPosition (uint32_t *number,* uint32_t *position,* int32_t *count,* BACNET_LOG_RECORD *p_records[ ],* uint32_t ∗ *p_count,* bool ∗ *p_first,* bool ∗ *p_last,* bool ∗ *p_more*)

Getting stored records for a trend log object. Beginning with the given record position and limited by the given count.

Parameters

in	*number*	Number of the trend log object.
in	*position*	Start position in the trend log array.
in	*count*	Maximum count of records to be returned.
out	*p_records*	Pointer to an array to store the record data.
out	*p_count*	Number of returned records.

Returns

For this demo application only true for a trend log object with number 1 and a valid index.

Definition at line 1220 of file user.c.

5.24.2.34 bool UserTrendlogGetRecordsByTime (uint32_t *number,* BACNET_DATE_TIME *start_time,* int32_t *count,* BACNET_LOG_RECORD *p_records[ ],* uint32_t ∗ *p_count,* bool ∗ *p_first,* bool ∗ *p_last,* bool ∗ *p_more*)

Getting stored records for a trend log object. Beginning with the record with a time stamp greater or equal to the given start_time and limited by the given count.

Parameters

in	*number*	Number of the trend log object.
in	*start_time*	Minimum time stamp for a record time stamp.
in	*count*	Maximum count of records to be returned.
out	*p_records*	Pointer to an array to store the record data.
out	*p_count*	Number of returned records.

Returns

For this demo application only true for a trend log object with number 1 and a valid index.

Definition at line 1390 of file user.c.

5.24.2.35 bool UserTrendlogGetRecordsByTimeRange (uint32_t *number,* BACNET_DATE_TIME *start_time,* BACNET_DATE_TIME *end_time,* BACNET_LOG_RECORD *p_records[ ],* uint32_t ∗ *p_count,* bool ∗ *p_first,* bool ∗ *p_last,* bool ∗ *p_more*)

Getting stored records for a trend log object. Beginning with the record with a time stamp greater or equal to the given start_time and limited by the given end_time.

Parameters

in	*number*	Number of the trend log object.
in	*start_time*	Minimum time stamp for a record time stamp.
in	*end_time*	Maximum time stamp for a record time stamp.
out	*p_records*	Pointer to an array to store the record data.
out	*p_count*	Number of returned records.

Returns

For this demo application only true for a trend log object with number 1 and a valid index.

Definition at line 1477 of file user.c.

5.24.2.36 bool UserTrendlogPutRecord (uint32_t *number,* BACNET_LOG_RECORD ∗ *p_record*)

Stores a record in the internal memory for a trend log object.

Parameters

in	*number*	Number of the trend log object.
in	*p_record*	Pointer to the record data.

Returns

For this demo application only true for a trend log object with number 1.

Definition at line 1148 of file user.c.

5.24.2.37 bool UserUserTrendlogGetRecordsBySequence (uint32_t *number,* uint32_t *sequence,* int32_t *count,* BACNET_LOG_RECORD *p_records[ ],* uint32_t ∗ *p_count,* bool ∗ *p_first,* bool ∗ *p_last,* bool ∗ *p_more*)

Getting stored records for a trend log object. Beginning with the record with the given sequence number and limited by the given count.

Parameters

in	*number*	Number of the trend log object.
in	*sequence*	Sequence number to start with.
in	*count*	Maximum count of records to be returned.
out	*p_records*	Pointer to an array to store the record data.
out	*p_count*	Number of returned records.

Returns

For this demo application only true for a trend log object with number 1 and a valid index.

Definition at line 1306 of file user.c.

5.24.3 Variable Documentation

5.24.3.1 BAC_BYTE g_flash_checksum `[static]`

Definition at line 147 of file user.c.

5.24.3.2 BAC_BYTE g_flash_magic[FLASH_MEM_MAGIC_SIZE] = {0xEF, 0xBE, 0xC0, 0xBA} `[static]`

Definition at line 145 of file user.c.

5.24.3.3 BAC_BYTE g_flash_mem_page[FLASH_DEVICE_PAGE_SIZE] `[static]`

Definition at line 143 of file user.c.

5.24.3.4 BAC_WORD g_flash_offset `[static]`

Definition at line 146 of file user.c.

5.24.3.5 BAC_UINT g_obj_sc1_daily_schedule[CALC_WRITABLE_DAILY_SCHEDULE_ARRAY_SIZE(OBJ_SC1_DAILY_S↩ CHEDULE_SIZE)]

5.24.3.6 BACNET_DEV_OBJ_PROP_REFERENCE g_obj_sc1_obj_prop_ref_list[ ]

5.24.3.7 BAC_UINT g_obj_sc1_obj_prop_ref_listCount

5.24.3.8 BAC_BOOLEAN g_page_any_page_changed `[static]`

Definition at line 151 of file user.c.

5.24.3.9 BAC_BOOLEAN g_page_data_changed `[static]`

Definition at line 150 of file user.c.

5.24.3.10 BAC_BOOLEAN g_page_is_new `[static]`

Definition at line 149 of file user.c.

5.24.3.11 BAC_WORD g_page_offset `[static]`

Definition at line 148 of file user.c.

5.24.3.12 uint32_t g_sequence `[static]`

Definition at line 137 of file user.c.

5.24.3.13 user_analog_input_t g_user_analog_values[USER_MAX_ANALOG] `[static]`

Definition at line 134 of file user.c.

5.24.3.14 user_binary_input_t g_user_binary_values[USER_MAX_BINARY] `[static]`

Definition at line 133 of file user.c.

5.24.3.15 user_button_t g_user_button[2] `[static]`

Initial value:

```
=
{
    {IOPORT_PORT_00_PIN_05, IOPORT_LEVEL_HIGH},
    {IOPORT_PORT_00_PIN_06, IOPORT_LEVEL_HIGH}
}
```

Definition at line 115 of file user.c.

5.24.3.16 const ioport_port_pin_t g_user_led_pins[ ] `[static]`

Initial value:

```
=
{
    IOPORT_PORT_06_PIN_00,
    IOPORT_PORT_06_PIN_01,
}
```

Definition at line 121 of file user.c.

5.24.3.17 const bsp_leds_t g_user_leds [static]

Initial value:

```
=
{
    .led_count = 2,
    .p_leds     = g_user_led_pins
}
```

Definition at line 127 of file user.c.

5.24.3.18 TX_MUTEX g_user_mutex [static]

Definition at line 141 of file user.c.

5.24.3.19 TX_THREAD g_user_thread [static]

Definition at line 139 of file user.c.

5.24.3.20 uint8_t g_user_thread_stack[0x1800] [static]

Definition at line 140 of file user.c.

5.24.3.21 user_trendlog_t g_user_trendlog [static]

Definition at line 136 of file user.c.

5.24.3.22 BAC_BYTE* gp_flash_mem = (BAC_BYTE *)FLASH_DEVICE_START_ADDRESS [static]

Definition at line 144 of file user.c.

5.25 src/demo/user.h File Reference

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

Data Structures

- struct user_record_t

Functions

- bool UserInit (void)

  Initialize the user interface.
- void UserReset (void)

  Reset the user interface.
- bool UserSetLed (uint32_t number, bool on_off)

  Setting the state of the LED for the binary output objects present value.
- bool UserTrendlogClearBuffer (uint32_t number)

  Clears the internal memory to store the trend log records.
- bool UserTrendlogPutRecord (uint32_t number, BACNET_LOG_RECORD ∗p_record)

  Stores a record in the internal memory for a trend log object.
- bool UserTrendlogGetRecords (uint32_t number, BACNET_LOG_RECORD p_records[ ], uint32_t ∗p_count, bool ∗p_first, bool ∗p_last, bool ∗p_more)

  Getting all stored records for a trend log object.
- bool UserTrendlogGetRecordsByPosition (uint32_t number, uint32_t position, int32_t count, BACNET_LO↩ G_RECORD p_records[ ], uint32_t ∗p_count, bool ∗p_first, bool ∗p_last, bool ∗p_more)

  Getting stored records for a trend log object. Beginning with the given record position and limited by the given count.
- bool UserUserTrendlogGetRecordsBySequence (uint32_t number, uint32_t sequence, int32_t count, BAC↩ NET_LOG_RECORD p_records[ ], uint32_t ∗p_count, bool ∗p_first, bool ∗p_last, bool ∗p_more)

  Getting stored records for a trend log object. Beginning with the record with the given sequence number and limited by the given count.
- bool UserTrendlogGetRecordsByTime (uint32_t number, BACNET_DATE_TIME start_time, int32_t count, BACNET_LOG_RECORD p_records[ ], uint32_t ∗p_count, bool ∗p_first, bool ∗p_last, bool ∗p_more)

  Getting stored records for a trend log object. Beginning with the record with a time stamp greater or equal to the given start_time and limited by the given count.
- bool UserTrendlogGetRecordsByTimeRange (uint32_t number, BACNET_DATE_TIME start_time, BACN↩ ET_DATE_TIME end_time, BACNET_LOG_RECORD p_records[ ], uint32_t ∗p_count, bool ∗p_first, bool ∗p_last, bool ∗p_more)

  Getting stored records for a trend log object. Beginning with the record with a time stamp greater or equal to the given start_time and limited by the given end_time.
- bool UserStorageIsPropertyPersistant (BACNET_INST_NUMBER device_id, BACNET_OBJECT_TYPE obj_type, BACNET_INST_NUMBER inst_number, BACNET_PROPERTY_ID property_id)

  Request if the given property should the stored persistent.
- bool UserStorageRead (const BACNET_SRVR_INIT ∗p_server)

  Function which reads out the data stored in the flash pages and copies them into the object property structures.
- bool UserStorageWrite (const BACNET_SRVR_INIT ∗p_server)

  Function which writes changed persistent property data.
- bool UserStorageDelete (const BACNET_SRVR_INIT ∗p_server)

  Request for deleting the persistent stored data.

5.25.1 Function Documentation

5.25.1.1 bool UserInit (void)

Initialize the user interface.

Set the initial values for: binary input objects (Switches) analog input objects (Potentiometers and temperature sensors) schedule object (Every full hour from Monday to Sunday) trend log object (Internal array to store the data given and requested by the BACnet API)

Return values

true	Always true for this demo application.

Definition at line 1064 of file user.c.

5.25.1.2 void UserReset (void)

Reset the user interface.

Set the user variables and device state to initial state

Definition at line 1097 of file user.c.

5.25.1.3 bool UserSetLed (uint32_t *number,* bool *on_off*)

Setting the state of the LED for the binary output objects present value.

Parameters

in	*number*	Number of the LED (ID of binary output object).
in	*on_off*	New State of the LED (Present value of binary output object).

Returns

Successfully changed state of the LED.

Definition at line 1112 of file user.c.

5.25.1.4 bool UserStorageDelete (const BACNET_SRVR_INIT ∗ *p_server*)

Request for deleting the persistent stored data.

Parameters

in	*p_server*	Pointer to the server structure of the demo application.

Returns

Successful deleting of the stored data or not.

Definition at line 1789 of file user.c.

5.25.1.5 bool UserStorageIsPropertyPersistant (BACNET_INST_NUMBER *device_id,* BACNET_OBJECT_TYPE *obj_type,* BACNET_INST_NUMBER *inst_number,* BACNET_PROPERTY_ID *property_id*)

Request if the given property should the stored persistent.

Parameters

in	*device_id*	Device number of the property.
in	*obj_type*	Object type of the property.
in	*inst_number*	Object instance of the property.
in	*property_id*	Property id of the property.

Returns

User decision if the requested should be stored or not.

Definition at line 1543 of file user.c.

5.25.1.6 bool UserStorageRead (const BACNET_SRVR_INIT ∗ *p_server*)

Function which reads out the data stored in the flash pages and copies them into the object property structures.

Parameters

in	*p_server*	Pointer to the server structure of the demo application.

Returns

Successful reading of the stored data or not.

Definition at line 1572 of file user.c.

5.25.1.7 bool UserStorageWrite (const BACNET_SRVR_INIT ∗ *p_server*)

Function which writes changed persistent property data.

Parameters

in	*p_server*	Pointer to the server structure of the demo application.

Returns

Successful writing of the stored data or not.

Definition at line 1702 of file user.c.

5.25.1.8 bool UserTrendlogClearBuffer (uint32_t *number*)

Clears the internal memory to store the trend log records.

Parameters

`in`	*number*	Number of the trend log object.

Returns

For this demo application only true for a trend log object with number 1.

Definition at line 1127 of file user.c.

5.25.1.9 bool UserTrendlogGetRecords (uint32_t *number,* BACNET_LOG_RECORD *p_records[ ],* uint32_t ∗ *p_count,* bool ∗ *p_first,* bool ∗ *p_last,* bool ∗ *p_more*)

Getting all stored records for a trend log object.

Parameters

`in`	*number*	Number of the trend log object.
`out`	*p_records*	Pointer to an array to store the record data.
`out`	*p_count*	Number of returned records.

Returns

For this demo application only true for a trend log object with number 1 and a valid index.

Definition at line 1180 of file user.c.

5.25.1.10 bool UserTrendlogGetRecordsByPosition (uint32_t *number,* uint32_t *position,* int32_t *count,* BACNET_LOG_RECORD *p_records[ ],* uint32_t ∗ *p_count,* bool ∗ *p_first,* bool ∗ *p_last,* bool ∗ *p_more*)

Getting stored records for a trend log object. Beginning with the given record position and limited by the given count.

Parameters

`in`	*number*	Number of the trend log object.
`in`	*position*	Start position in the trend log array.
`in`	*count*	Maximum count of records to be returned.
`out`	*p_records*	Pointer to an array to store the record data.
`out`	*p_count*	Number of returned records.

Returns

For this demo application only true for a trend log object with number 1 and a valid index.

Definition at line 1220 of file user.c.

5.25.1.11 bool UserTrendlogGetRecordsByTime (uint32_t *number*, BACNET_DATE_TIME *start_time*, int32_t *count*, BACNET_LOG_RECORD *p_records[ ]*, uint32_t ∗ *p_count*, bool ∗ *p_first*, bool ∗ *p_last*, bool ∗ *p_more*)

Getting stored records for a trend log object. Beginning with the record with a time stamp greater or equal to the given start_time and limited by the given count.

Parameters

in	*number*	Number of the trend log object.
in	*start_time*	Minimum time stamp for a record time stamp.
in	*count*	Maximum count of records to be returned.
out	*p_records*	Pointer to an array to store the record data.
out	*p_count*	Number of returned records.

Returns

For this demo application only true for a trend log object with number 1 and a valid index.

Definition at line 1390 of file user.c.

5.25.1.12 bool UserTrendlogGetRecordsByTimeRange (uint32_t *number*, BACNET_DATE_TIME *start_time*, BACNET_DATE_TIME *end_time*, BACNET_LOG_RECORD *p_records[ ]*, uint32_t ∗ *p_count*, bool ∗ *p_first*, bool ∗ *p_last*, bool ∗ *p_more*)

Getting stored records for a trend log object. Beginning with the record with a time stamp greater or equal to the given start_time and limited by the given end_time.

Parameters

in	*number*	Number of the trend log object.
in	*start_time*	Minimum time stamp for a record time stamp.
in	*end_time*	Maximum time stamp for a record time stamp.
out	*p_records*	Pointer to an array to store the record data.
out	*p_count*	Number of returned records.

Returns

For this demo application only true for a trend log object with number 1 and a valid index.

Definition at line 1477 of file user.c.

5.25.1.13 bool UserTrendlogPutRecord (uint32_t *number*, BACNET_LOG_RECORD ∗ *p_record*)

Stores a record in the internal memory for a trend log object.

Parameters

in	*number*	Number of the trend log object.
in	*p_record*	Pointer to the record data.

Returns

For this demo application only true for a trend log object with number 1.

Definition at line 1148 of file user.c.

5.25.1.14 bool UserUserTrendlogGetRecordsBySequence (uint32_t *number,* uint32_t *sequence,* int32_t *count,* BACNET_LOG_RECORD *p_records[ ],* uint32_t ∗ *p_count,* bool ∗ *p_first,* bool ∗ *p_last,* bool ∗ *p_more*)

Getting stored records for a trend log object. Beginning with the record with the given sequence number and limited by the given count.

Parameters

`in`	*number*	Number of the trend log object.
`in`	*sequence*	Sequence number to start with.
`in`	*count*	Maximum count of records to be returned.
`out`	*p_records*	Pointer to an array to store the record data.
`out`	*p_count*	Number of returned records.

Returns

For this demo application only true for a trend log object with number 1 and a valid index.

Definition at line 1306 of file user.c.

Index