IoT Embedded Systems

Alexandre Silva v1.0

Generated by Doxygen 1.8.16

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Chapter 2

File Index

2.1 File List

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

Chapter 3

Module Documentation

3.1 Main application functions.

This package provides the functions to be used in the main application.

Macros

- #define DEFAULT_WIFI_CONNECT false
- #define DEFAULT_SSID ""
- #define DEFAULT_PASS ""
- #define MQTT_PUBLISH_INTERVAL_S 30
- #define DEFAULT_CONNECTION_KEEPALIVE_S 60UL
- #define DEFAULT_MQTT_ADDRESS "iot-ps.ddns.net"
- #define DEFAULT_MQTT_PORT 1883
- #define DEFAULT_MQTT_DEVICE_TOKEN "G2-SEIoT"

Functions

- void RTOS_Init ()

  Asserts correct initialization of every FreeRTOS-Drivers API.
- void NETWORK_Init ()

  Asserts correct initialization of every FreeRTOS-Drivers API.
- void APPLICATION_Init ()

  Asserts correct initialization of every Application layer task.
- void appMENU_NormalAffixHandler (void ∗a)

  Handler for the display affixing option in normal mode.
- void appMENU_NormalManageLightISR (TimerHandle_t xTimer)

  Light managing ISR.
- void appMENU_NormalPublishISR (TimerHandle_t xTimer)

  MQTT data publishing ISR.
- void appMENU_NormalMode (void ∗arg)

  Normal mode task.
- void appMENU_MaintenanceMode (void ∗arg)

  Maintenance mode task.

3.1.1 Detailed Description

This package provides the functions to be used in the main application.

3.1.2 Macro Definition Documentation

3.1.2.1 DEFAULT_CONNECTION_KEEPALIVE_S

```
#define DEFAULT_CONNECTION_KEEPALIVE_S 60UL
```

Default keepalive time for the MQTT connection, in seconds.

3.1.2.2 DEFAULT_MQTT_ADDRESS

```
#define DEFAULT_MQTT_ADDRESS "iot-ps.ddns.net"
```

Default MQTT server address to connect to.

3.1.2.3 DEFAULT_MQTT_DEVICE_TOKEN

```
#define DEFAULT_MQTT_DEVICE_TOKEN "G2-SEIoT"
```

Default access token for the MQTT server to connect to.

3.1.2.4 DEFAULT_MQTT_PORT

```
#define DEFAULT_MQTT_PORT 1883
```

Default MQTT server port to connect to.

3.1.2.5 DEFAULT_PASS

```
#define DEFAULT_PASS ""
```

Default password of the AP for the system to connect to upon boot, if the respective option was enabled

3.1.2.6 DEFAULT_SSID

```
#define DEFAULT_SSID ""
```

Default SSID of the AP for the system to connect to upon boot, if the respective option was enabled

3.1.2.7 DEFAULT_WIFI_CONNECT

```
#define DEFAULT_WIFI_CONNECT false
```

Allows the system to connect to a predefined network before booting into normal mode.

3.1.2.8 MQTT_PUBLISH_INTERVAL_S

```
#define MQTT_PUBLISH_INTERVAL_S 30
```

Default period for the publication of data through MQTT, in seconds.

3.1.3 Function Documentation

3.1.3.1 APPLICATION_Init()

```
void APPLICATION_Init ( )
```

Asserts correct initialization of every Application layer task.

Returns

None.

3.1.3.2 appMENU_MaintenanceMode()

```
void appMENU_MaintenanceMode (
             void * arg )
```

Maintenance mode task.

Parameters

arg	Arguments.

Returns

None.

3.1.3.3 appMENU_NormalAffixHandler()

```
void appMENU_NormalAffixHandler (
            void * a )
```

Handler for the display affixing option in normal mode.

Parameters

a	Arguments.

Returns

None.

3.1.3.4 appMENU_NormalManageLightISR()

```
void appMENU_NormalManageLightISR (
            TimerHandle_t xTimer )
```

Light managing ISR.

Parameters

xTimer	Handle of the timer being used.

Returns

None.

3.1.3.5 appMENU_NormalMode()

```
void appMENU_NormalMode (
            void * arg )
```

Normal mode task.

Parameters

arg	Arguments.

Returns

None.

3.1.3.6 appMENU_NormalPublishISR()

```
void appMENU_NormalPublishISR (
            TimerHandle_t xTimer )
```

MQTT data publishing ISR.

Parameters

xTimer	Handle of the timer being used.

Returns

None.

3.1.3.7 NETWORK_Init()

```
void NETWORK_Init ( )
```

Asserts correct initialization of every FreeRTOS-Drivers API.

Returns

None.

3.1.3.8 RTOS_Init()

```
void RTOS_Init ( )
```

Asserts correct initialization of every FreeRTOS-Drivers API.

Returns

None.

3.2 Menus to be interfaced in the application.

This package provides the menus to be interfaced in the main application.

Macros

- #define MAX_AP_SCAN 5
- #define DEFAULT_UTC_ADJUST 1

Functions

- char ∗ appMENU_GetWeekDay (int index)

  Returns the string correspondent to the given week day.
- void appMAINTENANCE_SetTimeHandler (void ∗arg)

  Handles the selection of the SET TIME option in the maintenance mode.
- void appMAINTENANCE_SetDateHandler (void ∗arg)

  Handles the selection of the SET DATE option in the maintenance mode.
- void appMAINTENANCE_SyncTimeHandler (void ∗arg)

  Handles the selection of the SYNC TIME option in the maintenance mode.
- void appMAINTENANCE_SetLightHandler (void ∗arg)

  Handles the selection of the SET LIGHT option in the maintenance mode.
- void appMAINTENANCE_SetWifiHandler (void ∗arg)

  Handles the selection of the SET WIFI option in the maintenance mode.
- void appMAINTENANCE_SetMQTTHandler (void ∗arg)

  Handles the selection of the SET MQTT option in the maintenance mode.

3.2.1 Detailed Description

This package provides the menus to be interfaced in the main application.

3.2.2 Macro Definition Documentation

3.2.2.1 DEFAULT_UTC_ADJUST

```
#define DEFAULT_UTC_ADJUST 1
```

Default number of hours to be added to the retrieved synced time.

3.2.2.2 MAX_AP_SCAN

```
#define MAX_AP_SCAN 5
```

Maximum number of AP's to be scanned.

3.2.3 Function Documentation

3.2.3.1 appMAINTENANCE_SetDateHandler()

```
void appMAINTENANCE_SetDateHandler (
            void * arg )
```

Handles the selection of the SET DATE option in the maintenance mode.

Parameters

arg	Arguments passed to the handler.

Returns

None.

3.2.3.2 appMAINTENANCE_SetLightHandler()

```
void appMAINTENANCE_SetLightHandler (
            void * arg )
```

Handles the selection of the SET LIGHT option in the maintenance mode.

Parameters

arg	Arguments passed to the handler.

Returns

None.

3.2.3.3 appMAINTENANCE_SetMQTTHandler()

```
void appMAINTENANCE_SetMQTTHandler (
            void * arg )
```

Handles the selection of the SET MQTT option in the maintenance mode.

Parameters

arg	Arguments passed to the handler.

Returns

None.

3.2.3.4 appMAINTENANCE_SetTimeHandler()

```
void appMAINTENANCE_SetTimeHandler (
            void * arg )
```

Handles the selection of the SET TIME option in the maintenance mode.

Parameters

arg	Arguments passed to the handler.

Returns

None.

3.2.3.5 appMAINTENANCE_SetWifiHandler()

```
void appMAINTENANCE_SetWifiHandler (
            void * arg )
```

Handles the selection of the SET WIFI option in the maintenance mode.

Parameters

arg	Arguments passed to the handler.

Returns

None.

3.2.3.6 appMAINTENANCE_SyncTimeHandler()

```
void appMAINTENANCE_SyncTimeHandler (
            void * arg )
```

Handles the selection of the SYNC TIME option in the maintenance mode.

Parameters

arg	Arguments passed to the handler.

Returns

None.

3.2.3.7 appMENU_GetWeekDay()

```
char* appMENU_GetWeekDay (
             int index )
```

Returns the string correspondent to the given week day.

Parameters

index	Number of the weekday, starting from 0.

Returns

Week day string.

3.3 FreeRTOS-Application

Modules

- Main application functions.

  This package provides the functions to be used in the main application.
- Menus to be interfaced in the application.

  This package provides the menus to be interfaced in the main application.

3.3.1 Detailed Description

Chapter 4

File Documentation

4.1 C:/Users/alext/Desktop/SE/Workspace/FreeRTOS-↩ Template/inc/main.h File Reference

Contains the functions to be used in the main application.

```
#include <timers.h>
```

Macros

- #define DEFAULT_WIFI_CONNECT false
- #define DEFAULT_SSID ""
- #define DEFAULT_PASS ""
- #define MQTT_PUBLISH_INTERVAL_S 30
- #define DEFAULT_CONNECTION_KEEPALIVE_S 60UL
- #define DEFAULT_MQTT_ADDRESS "iot-ps.ddns.net"
- #define DEFAULT_MQTT_PORT 1883
- #define DEFAULT_MQTT_DEVICE_TOKEN "G2-SEIoT"

Functions

- void RTOS_Init ()

  Asserts correct initialization of every FreeRTOS-Drivers API.
- void NETWORK_Init ()

  Asserts correct initialization of every FreeRTOS-Drivers API.
- void APPLICATION_Init ()

  Asserts correct initialization of every Application layer task.
- void appMENU_NormalAffixHandler (void ∗a)

  Handler for the display affixing option in normal mode.
- void appMENU_NormalManageLightISR (TimerHandle_t xTimer)

  Light managing ISR.
- void appMENU_NormalPublishISR (TimerHandle_t xTimer)

  MQTT data publishing ISR.
- void appMENU_NormalMode (void ∗arg)

  Normal mode task.
- void appMENU_MaintenanceMode (void ∗arg)

  Maintenance mode task.

4.1.1 Detailed Description

Contains the functions to be used in the main application.

Version

1.0

Date

12 jul 2023

Author

Alexandre Silva

Software that is described herein is for illustrative purposes only which provides customers with programming information regarding the products. This software is supplied "AS IS" without any warranties.

4.2 C:/Users/alext/Desktop/SE/Workspace/FreeRTOS-↩ Template/inc/menus.h File Reference

Contains the menus to be used in the application.

```
#include <time.h>
```

Macros

- #define MAX_AP_SCAN 5
- #define DEFAULT_UTC_ADJUST 1

Functions

- char ∗ appMENU_GetWeekDay (int index)

  Returns the string correspondent to the given week day.
- void appMAINTENANCE_SetTimeHandler (void ∗arg)

  Handles the selection of the SET TIME option in the maintenance mode.
- void appMAINTENANCE_SetDateHandler (void ∗arg)

  Handles the selection of the SET DATE option in the maintenance mode.
- void appMAINTENANCE_SyncTimeHandler (void ∗arg)

  Handles the selection of the SYNC TIME option in the maintenance mode.
- void appMAINTENANCE_SetLightHandler (void ∗arg)

  Handles the selection of the SET LIGHT option in the maintenance mode.
- void appMAINTENANCE_SetWifiHandler (void ∗arg)

  Handles the selection of the SET WIFI option in the maintenance mode.
- void appMAINTENANCE_SetMQTTHandler (void ∗arg)

  Handles the selection of the SET MQTT option in the maintenance mode.

4.2.1 Detailed Description

Contains the menus to be used in the application.

Version

1.0

Date

12 jul 2023

Author

Alexandre Silva

Software that is described herein is for illustrative purposes only which provides customers with programming information regarding the products. This software is supplied "AS IS" without any warranties.