CPE403 – Advanced Embedded Systems

# Design Assignment 4

Name: Elmer Mejia

Email: mejiae4@unlv.nevada.edu

Github Repository link (root): [assignments](https://github.com/cpemejia/Launchpad_assignments.git)

Youtube Playlist link (root): [Tiva\_C](https://www.youtube.com/playlist?list=PL6PbL2NpuYCIJs2P3WZJMcGl-yEjS-LJJ)

**Follow the submission guideline to be awarded points for this Assignment.**

1. Block diagram and/or Schematics showing the components, pins used, and interface.

A picture containing text, electronics

Description automatically generated



1. Code for Tasks. for each task submit the modified or included code (from the base code) with highlights and justifications of the modifications. Also include the comments. If no base code is provided, submit the base code for the first task only. Use separate page for each task.

sunllghtService.c

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Filename: sunlightService.c

\*

\* Description: This file contains the implementation of the service.

\*

\* Copyright (c) 2015-2019, Texas Instruments Incorporated

\* All rights reserved.

\*

\* Redistribution and use in source and binary forms, with or without

\* modification, are permitted provided that the following conditions

\* are met:

\*

\* \* Redistributions of source code must retain the above copyright

\* notice, this list of conditions and the following disclaimer.

\*

\* \* Redistributions in binary form must reproduce the above copyright

\* notice, this list of conditions and the following disclaimer in the

\* documentation and/or other materials provided with the distribution.

\*

\* \* Neither the name of Texas Instruments Incorporated nor the names of

\* its contributors may be used to endorse or promote products derived

\* from this software without specific prior written permission.

\*

\* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"

\* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,

\* THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

\* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR

\* CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,

\* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,

\* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS;

\* OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,

\* WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR

\* OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,

\* EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* INCLUDES

\*/

**#include** <string.h>

**#include** <icall.h>

/\* This Header file contains all BLE API and icall structure definition \*/

**#include** "icall\_ble\_api.h"

**#include** "sunlightService.h"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* MACROS

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* CONSTANTS

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* TYPEDEFS

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* GLOBAL VARIABLES

\*/

// sunlightService Service UUID

CONST uint8\_t sunlightServiceUUID[ATT\_BT\_UUID\_SIZE] =

{

LO\_UINT16(SUNLIGHTSERVICE\_SERV\_UUID), HI\_UINT16(SUNLIGHTSERVICE\_SERV\_UUID)

};

// sunlightValue UUID

CONST uint8\_t sunlightService\_SunlightValueUUID[ATT\_UUID\_SIZE] =

{

TI\_BASE\_UUID\_128(SUNLIGHTSERVICE\_SUNLIGHTVALUE\_UUID)

};

// updatePeriod UUID

CONST uint8\_t sunlightService\_UpdatePeriodUUID[ATT\_UUID\_SIZE] =

{

TI\_BASE\_UUID\_128(SUNLIGHTSERVICE\_UPDATEPERIOD\_UUID)

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* LOCAL VARIABLES

\*/

**static** sunlightServiceCBs\_t \*pAppCBs = NULL;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Profile Attributes - variables

\*/

// Service declaration

**static** CONST gattAttrType\_t sunlightServiceDecl = { ATT\_BT\_UUID\_SIZE, sunlightServiceUUID };

// Characteristic "SunlightValue" Properties (for declaration)

**static** uint8\_t sunlightService\_SunlightValueProps = GATT\_PROP\_NOTIFY;

// Characteristic "SunlightValue" Value variable

**static** uint8\_t sunlightService\_SunlightValueVal[SUNLIGHTSERVICE\_SUNLIGHTVALUE\_LEN] = {0};

// Characteristic "SunlightValue" CCCD

**static** gattCharCfg\_t \*sunlightService\_SunlightValueConfig;

// Characteristic "UpdatePeriod" Properties (for declaration)

**static** uint8\_t sunlightService\_UpdatePeriodProps = GATT\_PROP\_READ | GATT\_PROP\_WRITE;

// Characteristic "UpdatePeriod" Value variable

**static** uint8\_t sunlightService\_UpdatePeriodVal[SUNLIGHTSERVICE\_UPDATEPERIOD\_LEN] = {0};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Profile Attributes - Table

\*/

**static** gattAttribute\_t sunlightServiceAttrTbl[] =

{

// sunlightService Service Declaration

{

{ ATT\_BT\_UUID\_SIZE, primaryServiceUUID },

GATT\_PERMIT\_READ,

0,

(uint8\_t \*)&sunlightServiceDecl

},

// SunlightValue Characteristic Declaration

{

{ ATT\_BT\_UUID\_SIZE, characterUUID },

GATT\_PERMIT\_READ,

0,

&sunlightService\_SunlightValueProps

},

// SunlightValue Characteristic Value

{

{ ATT\_UUID\_SIZE, sunlightService\_SunlightValueUUID },

GATT\_PERMIT\_READ,

0,

sunlightService\_SunlightValueVal

},

// SunlightValue CCCD

{

{ ATT\_BT\_UUID\_SIZE, clientCharCfgUUID },

GATT\_PERMIT\_READ | GATT\_PERMIT\_WRITE,

0,

(uint8 \*)&sunlightService\_SunlightValueConfig

},

// UpdatePeriod Characteristic Declaration

{

{ ATT\_BT\_UUID\_SIZE, characterUUID },

GATT\_PERMIT\_READ,

0,

&sunlightService\_UpdatePeriodProps

},

// UpdatePeriod Characteristic Value

{

{ ATT\_UUID\_SIZE, sunlightService\_UpdatePeriodUUID },

GATT\_PERMIT\_READ | GATT\_PERMIT\_WRITE,

0,

sunlightService\_UpdatePeriodVal

},

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* LOCAL FUNCTIONS

\*/

**static** bStatus\_t **sunlightService\_ReadAttrCB**( uint16\_t connHandle, gattAttribute\_t \*pAttr,

uint8\_t \*pValue, uint16\_t \*pLen, uint16\_t offset,

uint16\_t maxLen, uint8\_t method );

**static** bStatus\_t **sunlightService\_WriteAttrCB**( uint16\_t connHandle, gattAttribute\_t \*pAttr,

uint8\_t \*pValue, uint16\_t len, uint16\_t offset,

uint8\_t method );

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* PROFILE CALLBACKS

\*/

// Simple Profile Service Callbacks

CONST gattServiceCBs\_t sunlightServiceCBs =

{

sunlightService\_ReadAttrCB, // Read callback function pointer

sunlightService\_WriteAttrCB, // Write callback function pointer

NULL // Authorization callback function pointer

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* PUBLIC FUNCTIONS

\*/

/\*

\* SunlightService\_AddService- Initializes the SunlightService service by registering

\* GATT attributes with the GATT server.

\*

\*/

**extern** bStatus\_t **SunlightService\_AddService**( uint8\_t rspTaskId )

{

uint8\_t status;

// Allocate Client Characteristic Configuration table

sunlightService\_SunlightValueConfig = (gattCharCfg\_t \*)ICall\_malloc( **sizeof**(gattCharCfg\_t) \* linkDBNumConns );

**if** ( sunlightService\_SunlightValueConfig == NULL )

{

**return** ( bleMemAllocError );

}

// Initialize Client Characteristic Configuration attributes

**GATTServApp\_InitCharCfg**( LINKDB\_CONNHANDLE\_INVALID, sunlightService\_SunlightValueConfig );

// Register GATT attribute list and CBs with GATT Server App

status = GATTServApp\_RegisterService( sunlightServiceAttrTbl,

GATT\_NUM\_ATTRS( sunlightServiceAttrTbl ),

GATT\_MAX\_ENCRYPT\_KEY\_SIZE,

&sunlightServiceCBs );

**return** ( status );

}

/\*

\* SunlightService\_RegisterAppCBs - Registers the application callback function.

\* Only call this function once.

\*

\* appCallbacks - pointer to application callbacks.

\*/

bStatus\_t **SunlightService\_RegisterAppCBs**( sunlightServiceCBs\_t \*appCallbacks )

{

**if** ( appCallbacks )

{

pAppCBs = appCallbacks;

**return** ( SUCCESS );

}

**else**

{

**return** ( bleAlreadyInRequestedMode );

}

}

/\*

\* SunlightService\_SetParameter - Set a SunlightService parameter.

\*

\* param - Profile parameter ID

\* len - length of data to right

\* value - pointer to data to write. This is dependent on

\* the parameter ID and WILL be cast to the appropriate

\* data type (example: data type of uint16 will be cast to

\* uint16 pointer).

\*/

bStatus\_t **SunlightService\_SetParameter**( uint8\_t param, uint16\_t len, **void** \*value )

{

bStatus\_t ret = SUCCESS;

**switch** ( param )

{

**case** SUNLIGHTSERVICE\_SUNLIGHTVALUE\_ID:

**if** ( len == SUNLIGHTSERVICE\_SUNLIGHTVALUE\_LEN )

{

memcpy(sunlightService\_SunlightValueVal, value, len);

// Try to send notification.

**GATTServApp\_ProcessCharCfg**( sunlightService\_SunlightValueConfig, (uint8\_t \*)&sunlightService\_SunlightValueVal, FALSE,

sunlightServiceAttrTbl, GATT\_NUM\_ATTRS( sunlightServiceAttrTbl ),

INVALID\_TASK\_ID, sunlightService\_ReadAttrCB);

}

**else**

{

ret = bleInvalidRange;

}

**break**;

**case** SUNLIGHTSERVICE\_UPDATEPERIOD\_ID:

**if** ( len == SUNLIGHTSERVICE\_UPDATEPERIOD\_LEN )

{

memcpy(sunlightService\_UpdatePeriodVal, value, len);

}

**else**

{

ret = bleInvalidRange;

}

**break**;

**default**:

ret = INVALIDPARAMETER;

**break**;

}

**return** ret;

}

/\*

\* SunlightService\_GetParameter - Get a SunlightService parameter.

\*

\* param - Profile parameter ID

\* value - pointer to data to write. This is dependent on

\* the parameter ID and WILL be cast to the appropriate

\* data type (example: data type of uint16 will be cast to

\* uint16 pointer).

\*/

bStatus\_t **SunlightService\_GetParameter**( uint8\_t param, uint16\_t \*len, **void** \*value )

{

bStatus\_t ret = SUCCESS;

**switch** ( param )

{

**case** SUNLIGHTSERVICE\_UPDATEPERIOD\_ID:

memcpy(value, sunlightService\_UpdatePeriodVal, SUNLIGHTSERVICE\_UPDATEPERIOD\_LEN);

**break**;

**default**:

ret = INVALIDPARAMETER;

**break**;

}

**return** ret;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* @fn sunlightService\_ReadAttrCB

\*

\* @brief Read an attribute.

\*

\* @param connHandle - connection message was received on

\* @param pAttr - pointer to attribute

\* @param pValue - pointer to data to be read

\* @param pLen - length of data to be read

\* @param offset - offset of the first octet to be read

\* @param maxLen - maximum length of data to be read

\* @param method - type of read message

\*

\* @return SUCCESS, blePending or Failure

\*/

**static** bStatus\_t **sunlightService\_ReadAttrCB**( uint16\_t connHandle, gattAttribute\_t \*pAttr,

uint8\_t \*pValue, uint16\_t \*pLen, uint16\_t offset,

uint16\_t maxLen, uint8\_t method )

{

bStatus\_t status = SUCCESS;

// See if request is regarding the SunlightValue Characteristic Value

**if** ( ! **memcmp**(pAttr->type.uuid, sunlightService\_SunlightValueUUID, pAttr->type.len) )

{

**if** ( offset > SUNLIGHTSERVICE\_SUNLIGHTVALUE\_LEN ) // Prevent malicious ATT ReadBlob offsets.

{

status = ATT\_ERR\_INVALID\_OFFSET;

}

**else**

{

\*pLen = MIN(maxLen, SUNLIGHTSERVICE\_SUNLIGHTVALUE\_LEN - offset); // Transmit as much as possible

memcpy(pValue, pAttr->pValue + offset, \*pLen);

}

}

// See if request is regarding the UpdatePeriod Characteristic Value

**else** **if** ( ! **memcmp**(pAttr->type.uuid, sunlightService\_UpdatePeriodUUID, pAttr->type.len) )

{

**if** ( offset > SUNLIGHTSERVICE\_UPDATEPERIOD\_LEN ) // Prevent malicious ATT ReadBlob offsets.

{

status = ATT\_ERR\_INVALID\_OFFSET;

}

**else**

{

\*pLen = MIN(maxLen, SUNLIGHTSERVICE\_UPDATEPERIOD\_LEN - offset); // Transmit as much as possible

memcpy(pValue, pAttr->pValue + offset, \*pLen);

}

}

**else**

{

// If we get here, that means you've forgotten to add an if clause for a

// characteristic value attribute in the attribute table that has READ permissions.

\*pLen = 0;

status = ATT\_ERR\_ATTR\_NOT\_FOUND;

}

**return** status;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* @fn sunlightService\_WriteAttrCB

\*

\* @brief Validate attribute data prior to a write operation

\*

\* @param connHandle - connection message was received on

\* @param pAttr - pointer to attribute

\* @param pValue - pointer to data to be written

\* @param len - length of data

\* @param offset - offset of the first octet to be written

\* @param method - type of write message

\*

\* @return SUCCESS, blePending or Failure

\*/

**static** bStatus\_t **sunlightService\_WriteAttrCB**( uint16\_t connHandle, gattAttribute\_t \*pAttr,

uint8\_t \*pValue, uint16\_t len, uint16\_t offset,

uint8\_t method )

{

bStatus\_t status = SUCCESS;

uint8\_t paramID = 0xFF;

// See if request is regarding a Client Characteristic Configuration

**if** ( ! **memcmp**(pAttr->type.uuid, clientCharCfgUUID, pAttr->type.len) )

{

// Allow only notifications.

status = **GATTServApp\_ProcessCCCWriteReq**( connHandle, pAttr, pValue, len,

offset, GATT\_CLIENT\_CFG\_NOTIFY);

}

// See if request is regarding the UpdatePeriod Characteristic Value

**else** **if** ( ! **memcmp**(pAttr->type.uuid, sunlightService\_UpdatePeriodUUID, pAttr->type.len) )

{

**if** ( offset + len > SUNLIGHTSERVICE\_UPDATEPERIOD\_LEN )

{

status = ATT\_ERR\_INVALID\_OFFSET;

}

**else**

{

// Copy pValue into the variable we point to from the attribute table.

memcpy(pAttr->pValue + offset, pValue, len);

// Only notify application if entire expected value is written

**if** ( offset + len == SUNLIGHTSERVICE\_UPDATEPERIOD\_LEN)

paramID = SUNLIGHTSERVICE\_UPDATEPERIOD\_ID;

}

}

**else**

{

// If we get here, that means you've forgotten to add an if clause for a

// characteristic value attribute in the attribute table that has WRITE permissions.

status = ATT\_ERR\_ATTR\_NOT\_FOUND;

}

// Let the application know something changed (if it did) by using the

// callback it registered earlier (if it did).

**if** (paramID != 0xFF)

**if** ( pAppCBs && pAppCBs->pfnChangeCb )

pAppCBs->pfnChangeCb(connHandle, paramID, len, pValue); // Call app function from stack task context.

**return** status;

}

Sunlight service.h

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Filename: sunlightService.h

\*

\* Description: This file contains the sunlightService service definitions and

\* prototypes.

\*

\* Copyright (c) 2015-2019, Texas Instruments Incorporated

\* All rights reserved.

\*

\* Redistribution and use in source and binary forms, with or without

\* modification, are permitted provided that the following conditions

\* are met:

\*

\* \* Redistributions of source code must retain the above copyright

\* notice, this list of conditions and the following disclaimer.

\*

\* \* Redistributions in binary form must reproduce the above copyright

\* notice, this list of conditions and the following disclaimer in the

\* documentation and/or other materials provided with the distribution.

\*

\* \* Neither the name of Texas Instruments Incorporated nor the names of

\* its contributors may be used to endorse or promote products derived

\* from this software without specific prior written permission.

\*

\* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"

\* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,

\* THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

\* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR

\* CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,

\* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,

\* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS;

\* OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,

\* WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR

\* OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,

\* EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**#ifndef** \_SUNLIGHTSERVICE\_H\_

**#define** \_SUNLIGHTSERVICE\_H\_

**#ifdef** \_\_cplusplus

**extern** "C"

{

**#endif**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* INCLUDES

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* CONSTANTS

\*/

**#ifdef** DeviceFamily\_CC26X0R2

**#define** LINKDB\_CONNHANDLE\_INVALID CONNHANDLE\_INVALID

**#endif** //DeviceFamily\_CC26X0R2

// Service UUID

**#define** SUNLIGHTSERVICE\_SERV\_UUID 0xBA55

// Characteristic defines

**#define** SUNLIGHTSERVICE\_SUNLIGHTVALUE\_ID 0

**#define** SUNLIGHTSERVICE\_SUNLIGHTVALUE\_UUID 0x2BAD

**#define** SUNLIGHTSERVICE\_SUNLIGHTVALUE\_LEN 4

// Characteristic defines

**#define** SUNLIGHTSERVICE\_UPDATEPERIOD\_ID 1

**#define** SUNLIGHTSERVICE\_UPDATEPERIOD\_UUID 0x2BAE

**#define** SUNLIGHTSERVICE\_UPDATEPERIOD\_LEN 2

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* TYPEDEFS

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* MACROS

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Profile Callbacks

\*/

// Callback when a characteristic value has changed

**typedef** **void** (\*sunlightServiceChange\_t)(uint16\_t connHandle, uint8\_t paramID, uint16\_t len, uint8\_t \*pValue);

**typedef** **struct**

{

sunlightServiceChange\_t pfnChangeCb; // Called when characteristic value changes

sunlightServiceChange\_t pfnCfgChangeCb;

} sunlightServiceCBs\_t;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* API FUNCTIONS

\*/

/\*

\* SunlightService\_AddService- Initializes the SunlightService service by registering

\* GATT attributes with the GATT server.

\*

\*/

**extern** bStatus\_t **SunlightService\_AddService**( uint8\_t rspTaskId);

/\*

\* SunlightService\_RegisterAppCBs - Registers the application callback function.

\* Only call this function once.

\*

\* appCallbacks - pointer to application callbacks.

\*/

**extern** bStatus\_t **SunlightService\_RegisterAppCBs**( sunlightServiceCBs\_t \*appCallbacks );

/\*

\* SunlightService\_SetParameter - Set a SunlightService parameter.

\*

\* param - Profile parameter ID

\* len - length of data to right

\* value - pointer to data to write. This is dependent on

\* the parameter ID and WILL be cast to the appropriate

\* data type (example: data type of uint16 will be cast to

\* uint16 pointer).

\*/

**extern** bStatus\_t **SunlightService\_SetParameter**(uint8\_t param, uint16\_t len, **void** \*value);

/\*

\* SunlightService\_GetParameter - Get a SunlightService parameter.

\*

\* param - Profile parameter ID

\* value - pointer to data to write. This is dependent on

\* the parameter ID and WILL be cast to the appropriate

\* data type (example: data type of uint16 will be cast to

\* uint16 pointer).

\*/

**extern** bStatus\_t **SunlightService\_GetParameter**(uint8\_t param, uint16\_t \*len, **void** \*value);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**#ifdef** \_\_cplusplus

}

**#endif**

**#endif** /\* \_SUNLIGHTSERVICE\_H\_ \*/

Screenshots of the IDE, physical setup, debugging process - Provide screenshot of successful compilation, screenshots of registers, variables, graphs, etc.

Graphical user interface, text, application

Description automatically generated

1. Declaration

I understand the Student Academic Misconduct Policy - http://studentconduct.unlv.edu/misconduct/policy.html

“This assignment submission is my own, original work”.

-Elmer Mejia