**File Name**: gui\_pedometer\_objects.c

**File Description:** Pedometer Object File to define the default parameters for the Hexiwear Screen.

Default parameters like Font size, font color, left or right orientation, for the objects are defined in this file and further on used in the driver file.

**Source Code:**

*/\*\**

*\* \file gui\_pedometer\_objects.c*

*\* \version 1.00*

*\* \brief this file contains pedometer app related GUI objects*

*\**

*\* 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 NXP, 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 HOLDER 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.*

*\**

*\* visit: http://www.mikroe.com and http://www.nxp.com*

*\**

*\* get support at: http://www.mikroe.com/forum and https://community.nxp.com*

*\**

*\* Project HEXIWEAR, 2015*

*\*/*

*#include "screens\_common.h"*

*#include "gui\_pedometer\_private.h"*

*#include "gui\_pedometer.h"*

*guiScreen\_t*

*pedometerScreen =*

*{*

*.navigation =*

*{*

*.up = &heartRateScreen,*

*.down = &heartRateScreen,*

*.left = &fitnessScreen,*

*.right = NULL*

*},*

*.initFunction = gui\_pedometer\_Init,*

*.createTaskFunction = gui\_pedometer\_CreateTasks,*

*.destroyTaskFunction = gui\_pedometer\_DestroyTasks*

*};*

*guiLabel\_t*

*gui\_pedometer\_stepCounter\_label =*

*{*

*.dynamicArea =*

*{*

*.xCrd = 55,*

*.yCrd = 31,*

*.width = 35,*

*.height = 25*

*},*

*.textProperties =*

*{*

*.font = guiFont\_Tahoma\_8\_Regular,*

*.fontColor = GUI\_COLOR\_GREEN,* // This will render the Green color to the text appearing in the screen

*.alignParam = OLED\_TEXT\_ALIGN\_LEFT,*

*.background = NULL*

*},*

*.caption = NULL,*

*.captionLength = 20*

*};*

*guiLabel\_t*

*gui\_pedometer\_stepText\_label =*

*{*

*.dynamicArea =*

*{*

*.xCrd = 58,*

*.yCrd = 55,*

*.width = 28,*

*.height = 14*

*},*

*.textProperties =*

*{*

*.font = guiFont\_Tahoma\_8\_Regular,*

*.fontColor = GUI\_COLOR\_GREEN,* // This will render the Green color to the text appearing in the screen

*.alignParam = OLED\_TEXT\_ALIGN\_LEFT,*

*.background = NULL*

*},*

*.caption = NULL,*

*.captionLength = 20*

*};*

*guiLabel\_t*

*gui\_pedometer\_calCounter\_label =*

*{*

*.dynamicArea =*

*{*

*.xCrd = 9,*

*.yCrd = 31,*

*.width = 35,*

*.height = 25*

*},*

*.textProperties =*

*{*

*.font = guiFont\_Tahoma\_14\_Regular,*

*.fontColor = GUI\_COLOR\_WHITE,*

*.alignParam = OLED\_TEXT\_ALIGN\_LEFT,*

*.background = NULL*

*},*

*.caption = NULL,*

*.captionLength = 6*

*};*

*guiLabel\_t*

*gui\_pedometer\_calText\_label =*

*{*

*.dynamicArea =*

*{*

*.xCrd = 18,*

*.yCrd = 55,*

*.width = 15,*

*.height = 14*

*},*

*.textProperties =*

*{*

*.font = guiFont\_Tahoma\_8\_Regular,*

*.fontColor = GUI\_COLOR\_WHITE,*

*.alignParam = OLED\_TEXT\_ALIGN\_LEFT,*

*.background = NULL*

*},*

*.caption = NULL,*

*.captionLength = 4*

*};*

*guiImage\_t*

*gui\_pedometer\_icon =*

*{*

*.dynamicArea =*

*{*

*.xCrd = 24,*

*.yCrd = 23*

*},*

*.img = pedometer\_icon\_bmp*

*};*

**File Name**: gui\_pedometer\_driver.c

**File Description:** Pedometer Driver File to render the objects in the Hexiwear Screen.

Guardian Angel Logo and Flashlight will appear on the Hexiwear screen and color will change based upon the set values in this File.

**Source Code:**

*/\*\**

*\* \file gui\_pedometer\_driver.c*

*\* \version 1.00*

*\* \brief this file contains pedometer app GUI main task and functionality*

*\**

*\* 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 NXP, 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 HOLDER 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.*

*\**

*\* visit: http://www.mikroe.com and http://www.nxp.com*

*\**

*\* get support at: http://www.mikroe.com/forum and https://community.nxp.com*

*\**

*\* Project HEXIWEAR, 2015*

*\*/*

*#include "gui\_pedometer.h"*

*#include "gui\_pedometer\_private.h"*

*#include "screens\_common.h"*

*#include "fsl\_os\_abstraction.h"*

*#include "error.h"*

*#include "pedometer\_driver.h"*

*#include "flashlight.h"*

*#include "flashlight\_private.h"*

*#define SetFlashOFF1() isFlashlightOn = false; flashlight\_icon.img = flashlight\_off\_bmp; GREEN\_LED\_OFF()*

*#define SetFlashON1() isFlashlightOn = true; flashlight\_icon.img = flashlight\_on\_bmp; GREEN\_LED\_ON()*

*#define SetFlashOFF() isFlashlightOn = false; flashlight\_icon.img = flashlight\_off\_bmp; RED\_LED\_OFF()*

*#define SetFlashON() isFlashlightOn = true; flashlight\_icon.img = flashlight\_on\_bmp; RED\_LED\_ON()*

*static bool*

*isFlashlightOn;*

*static hostInterface\_packet\_t*

*flashlight\_dataPacket;*

*/\*\* private variables \*/*

*MSG\_QUEUE\_DECLARE( gui\_pedometer\_queue, 1, sizeof(uint32\_t) );*

*static void gui\_pedometer\_Task( task\_param\_t param );*

*static void gui\_pedometer\_UpdateStepsTask( void\* param );*

*static void gui\_pedometer\_Update();*

*static gui\_pedometer\_status\_t gui\_pedometer\_QueueMsgGet( uint32\_t timeout );*

*static msg\_queue\_handler\_t*

*gui\_pedometer\_queueHnd;*

*/\*\* new value given by the pedometer task \*/*

*static uint32\_t*

*newPedVal = 0;*

*/\*\* steps \*/*

*static uint16\_t*

*oldStepVal = 0,*

*newStepVal = 0;*

*/\*\* calories \*/*

*static uint16\_t*

*oldCalVal = 0,*

*newCalVal = 0;*

*static task\_handler\_t*

*gui\_pedometer\_step\_counter\_taskHandler,*

*gui\_pedometer\_taskHandler;*

*static hostInterface\_packet\_t*

*gui\_pedometer\_packet;*

*static bool*

*isPedometerActive = false;*

*/\*\**

*\* initialize pedometer app GUI*

*\* @param param optional parameter*

*\*/*

*void gui\_pedometer\_Init( void\* param )*

*{*

*// create pedometer packet queue*

*gui\_pedometer\_queueHnd = OSA\_MsgQCreate (*

*gui\_pedometer\_queue,*

*1,*

*sizeof(uint32\_t) / sizeof(uint32\_t)*

*);*

*#if defined( HEXIWEAR\_DEBUG )*

*vQueueAddToRegistry( gui\_pedometer\_queueHnd, (char\*)"GUI Pedometer Queue" );*

*#endif*

*GuiDriver\_ImageAddToScr( &screen\_buttonStart );*

*GuiDriver\_LabelCreate( &gui\_pedometer\_stepCounter\_label );*

*//Guardian Angel Logo will be rendered with the help of the below 5 lines along with the Green color*

*GuiDriver\_LabelSetCaption( &gui\_pedometer\_stepCounter\_label, (uint8\_t\*)"Angel" );*

*gui\_pedometer\_stepCounter\_label.textProperties.fontColor=GUI\_COLOR\_GREEN;*

*GuiDriver\_LabelCreate( &gui\_pedometer\_stepText\_label );*

*GuiDriver\_LabelSetCaption( &gui\_pedometer\_stepText\_label, (uint8\_t\*)"Guardian" );*

*gui\_pedometer\_stepText\_label.textProperties.fontColor=GUI\_COLOR\_GREEN;*

*GuiDriver\_RegisterForNavigation( GUI\_NAVIGATION\_RIGHT );*

*// initiate the pedometer process structures*

*oldStepVal = newStepVal = 0;*

*pedometer\_Init();*

*GuiDriver\_NotifyKW40( GUI\_CURRENT\_APP\_PEDOMETER );*

*}*

*/\*\**

*\* create the pedometer GUI app tasks*

*\* @param param optional parameter*

*\*/*

*void gui\_pedometer\_CreateTasks( void\* param )*

*{*

*osa\_status\_t*

*taskStatus;*

*pedometer\_CreateTask();*

*taskStatus = OSA\_TaskCreate (*

*gui\_pedometer\_Task,*

*(uint8\_t\*) "pedometer window",*

*GUI\_PEDOMETER\_STACK\_SIZE,*

*NULL,*

*GUI\_PEDOMETER\_PRIO,*

*(task\_param\_t)0,*

*false,*

*&gui\_pedometer\_taskHandler*

*);*

*taskStatus |= OSA\_TaskCreate(*

*gui\_pedometer\_UpdateStepsTask,*

*(uint8\_t\*) "pedometer step counter",*

*GUI\_PEDOMETER\_STEP\_COUNTER\_STACK\_SIZE,*

*NULL,*

*GUI\_PEDOMETER\_STEP\_COUNTER\_PRIO,*

*(task\_param\_t)0,*

*false,*

*&gui\_pedometer\_step\_counter\_taskHandler*

*);*

*if ( kStatus\_OSA\_Success != taskStatus )*

*{*

*catch( CATCH\_INIT );*

*}*

*vTaskSuspend( gui\_pedometer\_step\_counter\_taskHandler );*

*}*

*/\*\**

*\* destroy the pedometer GUI app tasks*

*\* @param param optional parameter*

*\*/*

*void gui\_pedometer\_DestroyTasks( void\* param )*

*{*

*OSA\_TaskDestroy( gui\_pedometer\_taskHandler );*

*OSA\_TaskDestroy( gui\_pedometer\_step\_counter\_taskHandler );*

*OSA\_MsgQDestroy( gui\_pedometer\_queueHnd );*

*pedometer\_Deinit();*

*OLED\_DestroyDynamicArea();*

*GuiDriver\_LabelDestroy( &gui\_pedometer\_stepCounter\_label );*

*GuiDriver\_LabelDestroy( &gui\_pedometer\_stepText\_label );*

*GuiDriver\_LabelDestroy( &gui\_pedometer\_calCounter\_label );*

*GuiDriver\_LabelDestroy( &gui\_pedometer\_calText\_label );*

*GuiDriver\_UnregisterFromNavigation( GUI\_NAVIGATION\_RIGHT );*

*isPedometerActive = false;*

*power\_EnablePowerSave();*

*if ( false == gui\_sensorTag\_IsActive() )*

*{*

*GuiDriver\_NotifyKW40( GUI\_CURRENT\_APP\_IDLE );*

*}*

*}*

*/\*\**

*\* put the packet into pedometer app GUI queue*

*\* @param packet data to be put into the queue*

*\* @return status flag*

*\*/*

*gui\_pedometer\_status\_t gui\_pedometer\_QueueMsgPut( uint32\_t\* packet )*

*{*

*BaseType\_t*

*status = xQueueSendToBack ( gui\_pedometer\_queueHnd, (uint32\_t\*)packet, OSA\_WAIT\_FOREVER );*

*if ( pdPASS == status )*

*{*

*return GUI\_PEDOMETER\_STATUS\_SUCCESS;*

*}*

*else*

*{*

*catch( CATCH\_QUEUE );*

*return GUI\_PEDOMETER\_STATUS\_ERROR;*

*}*

*}*

*/\*\* private API \*/*

*/\*\**

*\* pedometer app GUI main task,*

*\* get data from pedometer app and display it*

*\* @param task\_param\_t optional parameter*

*\*/*

*static void gui\_pedometer\_Task( task\_param\_t param )*

*{*

*while (1)*

*{*

*gui\_status\_t*

*rightClickStatus = GuiDriver\_QueueMsgGet( &gui\_pedometer\_packet , OSA\_WAIT\_FOREVER );*

*if( GUI\_STATUS\_SUCCESS == rightClickStatus )*

*{*

*if ( packetType\_pressRight == gui\_pedometer\_packet.type )*

*{*

*if ( false == isPedometerActive )*

*{*

*isPedometerActive = true;*

*power\_DisablePowerSave();*

*GuiDriver\_CleanMainArea();*

*GuiDriver\_LabelDraw( &gui\_pedometer\_stepCounter\_label );*

*GuiDriver\_LabelDraw( &gui\_pedometer\_stepText\_label );*

*GuiDriver\_ImageDraw(&screen\_buttonStop);*

*vTaskResume( gui\_pedometer\_step\_counter\_taskHandler );*

*pedometer\_Resume();*

*}*

*else*

*{*

*pedometer\_Pause();*

*vTaskSuspend( gui\_pedometer\_step\_counter\_taskHandler );*

*GuiDriver\_CleanMainArea();*

*GuiDriver\_ImageDraw(&screen\_buttonStart);*

*power\_EnablePowerSave();*

*isPedometerActive = false;*

*}*

*haptic\_Vibrate();*

*}*

*}*

*}*

*}*

*/\*\**

*\* monitor and update GUI when step counter changes*

*\* @param task\_param\_t optional parameter*

*\*/*

*static void gui\_pedometer\_UpdateStepsTask( task\_param\_t param )*

*{*

*while (1)*

*{*

*// save the old values*

*oldStepVal = newStepVal;*

*oldCalVal = newCalVal;*

*// wait for the new one*

*gui\_pedometer\_status\_t*

*guiStatus = gui\_pedometer\_QueueMsgGet( OSA\_WAIT\_FOREVER );*

*if ( GUI\_PEDOMETER\_STATUS\_SUCCESS != guiStatus )*

*{*

*catch ( CATCH\_HEALTH );*

*}*

*else*

*{*

*gui\_pedometer\_Update();*

*}*

*}*

*}*

*/\*\**

*\* update pedometer GUI*

*\*/*

*static void gui\_pedometer\_Update()*

*{*

*newStepVal = (uint16\_t)( newPedVal );*

*newCalVal = (uint16\_t)( newPedVal >> 16 );*

*if ( ( oldStepVal != newStepVal ) && ( 0 != newStepVal ) )*

*{*

*if ((newStepVal)<10)*

*{*

*//Guardian Angel Logo will be Green in color if the steps walked with the Hexiwear Device are below 10*

*gui\_pedometer\_stepText\_label.textProperties.fontColor=GUI\_COLOR\_GREEN;*

*gui\_pedometer\_stepCounter\_label.textProperties.fontColor=GUI\_COLOR\_GREEN;*

*GuiDriver\_LabelDraw( &gui\_pedometer\_stepCounter\_label );*

*GuiDriver\_LabelDraw( &gui\_pedometer\_stepText\_label );*

*{*

*//Flashlight will be green in color if the steps walked with the Hexiwear Device are below 10*

*SetFlashON1();*

*SetFlashOFF();*

*}*

*else*

*{*

*//Guardian Angel Logo will be red in color if the steps walked with the Hexiwear Device are above or equal to 10*

*gui\_pedometer\_stepText\_label.textProperties.fontColor=GUI\_COLOR\_RED;*

*gui\_pedometer\_stepCounter\_label.textProperties.fontColor=GUI\_COLOR\_RED;*

*GuiDriver\_LabelDraw( &gui\_pedometer\_stepCounter\_label );*

*GuiDriver\_LabelDraw( &gui\_pedometer\_stepText\_label );*

*//Haptic Feedback will be initiated if the steps are above or equal to 10*

*haptic\_Run();*

*//Flashlight will be red in color if the steps walked with the Hexiwear Device are above or equal to 10*

*SetFlashON();*

*SetFlashOFF1();*

*}*

*}*

*}*

*/\*\**

*\* get data from the GUI queue*

*\* @param timeout time value to wait for data in [ms]*

*\* @return status flag*

*\*/*

*static gui\_pedometer\_status\_t gui\_pedometer\_QueueMsgGet( uint32\_t timeout )*

*{*

*osa\_status\_t*

*status = OSA\_MsgQGet( gui\_pedometer\_queueHnd, &newPedVal, timeout );*

*if ( kStatus\_OSA\_Error == status )*

*{*

*catch( CATCH\_QUEUE );*

*return GUI\_PEDOMETER\_STATUS\_ERROR;*

*}*

*else*

*{*

*return GUI\_PEDOMETER\_STATUS\_SUCCESS;*

*}*

*}*