**LCD\_THREAD\_ENTRY**

(**PRIMERA REVISION 180719**)

(**SEGUNDA REVISION 190719**)

(**TERCERA REVISION 200719**)

(**CUARTA REVISION 220719**)

(**QUINTA REVISION 240719**)

(**SEXTA REVISION 250719**)

(**SEPTIMA REVISION 260719 VARS**)

(**OCTAVA REVISION 260719 OPTIMIZACION**)

#include <lcd\_thread.h>

#include <system\_thread.h>

#include "gx\_api.h"

#include "gui/gui\_adc\_specifications.h"

#include "gui/gui\_adc\_resources.h"

#include "lcd\_setup/lcd.h"

GX\_WINDOW\_ROOT \* psWindowRoot;

**extern** GX\_CONST GX\_STUDIO\_WIDGET \* gui\_adc\_widget\_table[];

uint16\_t au16ReceiveBuffer [2] = {0};

GX\_VALUE i16ReceiveBuffer360;

GX\_VALUE i16ReceiveBufferRpm;

/\* New Thread entry function \*/

**void** **lcd\_thread\_entry** (**void**)

{

ssp\_err\_t err;

UINT status = TX\_SUCCESS;

/\* Initializes GUIX. \*/

status = gx\_system\_initialize();

**if** (TX\_SUCCESS != status)

{

**while** (1)

{

}

}

/\* Initializes GUIX drivers. \*/

err = g\_sf\_el\_gx.p\_api->open(g\_sf\_el\_gx.p\_ctrl, g\_sf\_el\_gx.p\_cfg);

**if** (*SSP\_SUCCESS* != err)

{

**while** (1)

{

}

}

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*FUNCION CREATE WIDGETS-START\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

gx\_studio\_display\_configure(DISPLAY\_1,

g\_sf\_el\_gx.p\_api->setup,

LANGUAGE\_ENGLISH,

DISPLAY\_1\_THEME\_1,

&psWindowRoot);

err = g\_sf\_el\_gx.p\_api->canvasInit(g\_sf\_el\_gx.p\_ctrl, psWindowRoot);

**if** (*SSP\_SUCCESS* != err)

{

**while** (1)

{

}

}

GX\_CONST GX\_STUDIO\_WIDGET \*\* ppsStudioWidget = &gui\_adc\_widget\_table[0];

GX\_WIDGET \* psFirstScreen = NULL;

**while** (GX\_NULL != \* ppsStudioWidget)

{

// We must first create the widgets according the data generated in GUIX Studio.

// Once we are working on the widget we want to see first, save the pointer for later.

**if** (0 == **strcmp**("window1", (**char** \*) (\*ppsStudioWidget)->widget\_name))

{

gx\_studio\_named\_widget\_create((\*ppsStudioWidget)->widget\_name, (GX\_WIDGET \*) p\_window\_root, GX\_NULL);

}

**else**

{

gx\_studio\_named\_widget\_create((\*ppsStudioWidget)->widget\_name, GX\_NULL, GX\_NULL);

}

// Move to next top-level widget

ppsStudioWidget++;

}

// Attach the first screen to the root so we can see it when the root is shown

gx\_widget\_attach(psWindowRoot, psFirstScreen);

**if** (TX\_SUCCESS != status)

{

**while** (1)

{

}

}

/\* Shows the root window to make it and patients screen visible. \*/

status = gx\_widget\_show(psWindowRoot);

**if** (TX\_SUCCESS != status)

{

**while** (1)

{

}

}

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*FUNCION CREATE WIDGETS-END\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

/\* Lets GUIX run. \*/

status = gx\_system\_start();

**if** (TX\_SUCCESS != status)

{

**while** (1)

{

}

}

/\*\* Open the SPI driver to initialize the LCD (SK-S7G2) \*\*/

err = g\_spi\_lcdc.p\_api->open(g\_spi\_lcdc.p\_ctrl, g\_spi\_lcdc.p\_cfg);

**if** (err)

{

**while** (1)

{

}

}

/\*\* Setup the ILI9341V (SK-S7G2) \*\*/

ILI9341V\_Init();

/\* **TODO**: add your own code here \*/

**while** (1)

{

tx\_queue\_receive(&Message\_Queue, au16ReceiveBuffer, TX\_WAIT\_FOREVER);

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*FUNCION UPDATE LCD-START\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

//Assign data to send to the widgets

**char** lu8Text [8];

gx\_utility\_ltoa((LONG) au16ReceiveBuffer [1], lu8Text, 8);

i16ReceiveBuffer360= (GX\_VALUE)(( au16ReceiveBuffer [0]\*-360/100));//SIGNED SHORT [−32,767, +32,767]

i16ReceiveBufferRpm = (GX\_VALUE)(( au16ReceiveBuffer [1]\*-360/800));

// Change the prompt text

status = gx\_prompt\_text\_set(&window1.window1\_prompt\_1, lu8Text);

//Change the radial value

gx\_radial\_progress\_bar\_value\_set(&window1.window1\_radial\_progress\_bar, i16ReceiveBuffer360);

gx\_radial\_progress\_bar\_value\_set(&window1.window1\_radial\_progress\_bar\_1, i16ReceiveBufferRpm);

// Redraw the prompt

gx\_system\_dirty\_mark((GX\_WIDGET \*) &window1.window1\_prompt);

gx\_system\_canvas\_refresh();

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*FUNCION UPDATE LCD-END\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

tx\_thread\_sleep(10);//100

}

}

**void** **g\_lcd\_spi\_callback** (spi\_callback\_args\_t \* p\_args)

{

**if** (p\_args->event == *SPI\_EVENT\_TRANSFER\_COMPLETE*)

{

tx\_semaphore\_ceiling\_put(&g\_main\_semaphore\_lcdc, 1);

}

}