LCD\_THREAD\_ENTRY

(**PRIMERA REVISION 180719**)

(**SEGUNDA REVISION 190719**)

(**TERCERA REVISION**)

**#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 \* p\_window\_root;

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

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

GX\_VALUE ReceiveBuffer360;

GX\_VALUE ReceiveBufferRpm;

/\* 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)

{

}

}

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

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

LANGUAGE\_ENGLISH,

DISPLAY\_1\_THEME\_1,

&p\_window\_root);

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

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

{

**while** (1)

{

}

}

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

GX\_WIDGET \* p\_first\_screen = NULL;

**while** (GX\_NULL != \*pp\_studio\_widget)

{

// 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** \*) (\*pp\_studio\_widget)->widget\_name))

{

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

}

**else**

{

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

}

// Move to next top-level widget

pp\_studio\_widget++;

}

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

gx\_widget\_attach(p\_window\_root, p\_first\_screen);

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

{

**while** (1)

{

}

}

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

status = gx\_widget\_show(p\_window\_root);

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

{

**while** (1)

{

}

}

/\* 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, ReceiveBuffer, TX\_WAIT\_FOREVER);

//Assign data to send to the widgets

**char** text[8];

**char** text2[8];

ReceiveBuffer360 = (GX\_VALUE)((ReceiveBuffer[0]\*-360/100));//SIGNED SHORT [−32,767, +32,767]

ReceiveBufferRpm = (GX\_VALUE)((ReceiveBuffer[1]\*-360/800));

gx\_utility\_ltoa((LONG) ReceiveBuffer[0], text, 8);

gx\_utility\_ltoa((LONG) ReceiveBuffer[1], text2, 8);

// Change the prompt text

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

status = gx\_prompt\_text\_set(&window1.window1\_prompt\_1, text2);

//Change the radial value

gx\_radial\_progress\_bar\_value\_set(&window1.window1\_radial\_progress\_bar, ReceiveBuffer360);

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

// Redraw the prompt

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

gx\_system\_canvas\_refresh();

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);

}

}

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

**#include** "lcd\_thread.h"

**#include** "gx\_api.h"

GX\_WINDOW\_ROOT \* p\_window\_root;

/\* LCD Thread entry function \*/

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

{

/\* Initializes GUIX. \*/

gx\_system\_initialize();

/\* Initializes GUIX drivers. \*/

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

g\_sf\_el\_gx.p\_api->canvasInit(g\_sf\_el\_gx.p\_ctrl, p\_window\_root);

/\* Lets GUIX run. \*/

gx\_system\_start();

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

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

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

ILI9341V\_Init();

**while** (1)

{

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

tx\_thread\_sleep (1);

}

}

//Comunicacion de pantalla

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

{

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

{

}

}