

I have STM32F412G-Discovery board. When the icons appear for the start of the corresponding module. All modules work. You can add and remove them. I decided to make a module using SWIPELIST widget. It works and scrolls. But when it comes to event press exit button, the program goes to HardFault. The remaining modules are normally closed.

```
#include "dialog.h"
#include "k_module.h"
#include "information_res.c"
#include "SWIPELIST.h"
```

```
static void Startup(WM_HWIN hWin, uint16_t xpos, uint16_t ypos);
```

```
K_ModuleItem_Typedef INFORMATION_board = { 4, (uint8_t*)"", &bmImageMenuConfig_96x96,
Startup, };
```

```
#define ID_FRAMEWIN_INFO      (GUI_ID_USER + 0x01)
#define ID_SWIPELIST_0        (GUI_ID_USER + 0x02)
```

```
#define ID_RADIO_0            (GUI_ID_USER + 0x03)
#define ID_CHECK_0            (GUI_ID_USER + 0x04)
```

```
#define NUM_SEPARATORS 3      // Number of available separators
```

```
#define NUM_ITEMS 5          // Number of available items
```

```
static const char * paItemText[3][5] = {
{
"Wifi",
"Mobile Data",
"Bluetooth",
"Hot Spot",
"Mobil",
},
{
"Brightness",
"Background",
"Idle",
"Night Mode",
"Font Size",
},
{
"Account",
"Sound",
"Memory",
"Alert",
"Clock",
},
};
```

```
static const char * paSeperatorText[] = {
"Network",
"Display",
"Settings"
};
```

```
static int _SwipeOwnerDraw(const WIDGET_ITEM_DRAW_INFO * pDrawItemInfo)
{
//
// Distinguish between different commands
```

```

//
switch (pDrawItemInfo->Cmd) {
case WIDGET_ITEM_DRAW_SEP:
    GUI_DrawGradientH(pDrawItemInfo->x0, pDrawItemInfo->y0, pDrawItemInfo->x1 - (pDrawItemInfo->x1 - pDrawItemInfo->x0) / 2, pDrawItemInfo->y1, GUI_DARKGRAY, GUI_BLACK);
    GUI_DrawGradientH(pDrawItemInfo->x1 - (pDrawItemInfo->x1 - pDrawItemInfo->x0) / 2, pDrawItemInfo->y0, pDrawItemInfo->x1, pDrawItemInfo->y1, GUI_BLACK, GUI_DARKGRAY);
    break;
case WIDGET_ITEM_DRAW_TEXT:
    //
    // Just set a text mode but let the default owner draw routine handle the rest
    //
    GUI_SetTextMode(GUI_TM_TRANS);
    SWIPELIST_OwnerDraw(pDrawItemInfo);
    break;
case WIDGET_ITEM_DRAW_BITMAP:
    break;
case WIDGET_ITEM_DRAW_BACKGROUND:
    //
    // Handle drawing of the background of the items
    //
    switch (pDrawItemInfo->ItemIndex) {
    case 0: // 1st separator
    case 5: // 2nd separator
    case 10: // 3rd separator
    case 15: // 4th separator
        GUI_SetColor(GUI_BLACK);
        GUI_FillRect(pDrawItemInfo->x0, pDrawItemInfo->y0, pDrawItemInfo->x1, pDrawItemInfo->y1);
        GUI_DrawGradientV(pDrawItemInfo->x0, pDrawItemInfo->y0, pDrawItemInfo->x1, pDrawItemInfo->y1 - (pDrawItemInfo->y1 - pDrawItemInfo->y0) / 2, GUI_BLACK, GUI_GRAY);
        GUI_SetColor(GUI_LIGHTGRAY);
        GUI_DrawRoundedRect(pDrawItemInfo->x0, pDrawItemInfo->y0, pDrawItemInfo->x1, pDrawItemInfo->y1 - 1, 5);
        return 0;
    default: // Any other item
        //
        // Determin if the item to drawn is the currently selected
        //
        if (SWIPELIST_GetSelItem(pDrawItemInfo->hWin) == pDrawItemInfo->ItemIndex) {
            //
            // Draw the selected one different
            //
            GUI_SetColor(GUI_DARKRED);
            GUI_FillRect(pDrawItemInfo->x0, pDrawItemInfo->y0, pDrawItemInfo->x1, pDrawItemInfo->y1);
            GUI_DrawGradientV(pDrawItemInfo->x0, pDrawItemInfo->y0, pDrawItemInfo->x1, pDrawItemInfo->y1 - (pDrawItemInfo->y1 - pDrawItemInfo->y0) * 2 / 3, GUI_BLACK, GUI_DARKRED);
            GUI_DrawGradientV(pDrawItemInfo->x0, pDrawItemInfo->y1 - (pDrawItemInfo->y1 - pDrawItemInfo->y0) / 3, pDrawItemInfo->x1, pDrawItemInfo->y1, GUI_DARKRED, GUI_BLACK);
        } else {
            //
            // Draw any other items this way
            //
            GUI_SetColor(GUI_DARKGRAY);
            GUI_FillRect(pDrawItemInfo->x0, pDrawItemInfo->y0, pDrawItemInfo->x1, pDrawItemInfo->y1);
        }
    }
    break;
default:

```

```

//
// Anything we do not catch in this routine gets handled by the default owner draw
//
return SWIPELIST_OwnerDraw(pDrawItemInfo);
}
return 0;
}

```

```

static void _cbSwipe(WM_MESSAGE *pMsg)
{
int NCode;
int Id;
WM_HWIN hItem;
int Value;

switch (pMsg->MsgId) {
case WM_NOTIFY_PARENT:
//
// Get Id of the child got something
//
Id = WM_GetId(pMsg->hWinSrc);
//
// Get an idea of what it got
//
NCode = pMsg->Data.v;
switch(Id) {
case ID_RADIO_0: // Notifications sent by the RADIO widget
switch(NCode) {
case WM_NOTIFICATION_VALUE_CHANGED:
hItem = WM_GetDialogItem(pMsg->hWin, ID_RADIO_0);
Value = RADIO_GetValue(hItem);
break;
}
break;
case ID_CHECK_0: // Notifications sent by CHECKBOX widget
switch(NCode) {
case WM_NOTIFICATION_VALUE_CHANGED:
hItem = WM_GetDialogItem(pMsg->hWin, ID_CHECK_0);
Value = CHECKBOX_GetState(hItem);
break;
}
break;
}
break;
default:
SWIPELIST_Callback(pMsg);
break;
}
}
}

```

```

static const GUI_WIDGET_CREATE_INFO _aDialogInfo[] =
{
{ WINDOW_CreateIndirect, "Information", 0, 0, 0, 240, 240, FRAMEWIN_CF_MOVEABLE },
// { WINDOW_CreateIndirect, "Information", ID_FRAMEWIN_INFO, 0, 0, 240, 240, 0, 0, 0 },
{ SWIPELIST_CreateIndirect, "SwipeList", ID_SWIPELIST_0, 5, 55, 230, 180, 0, 0x0, 0 },
{ BUTTON_CreateIndirect, "Button", ID_INFO_EXIT, 192, 0, 48, 48, 0, 0x0, 0 }
};

```

```

static void _OnPaint_exit(BUTTON_Handle hObj)
{
    GUI_DrawBitmap(&bmImageExit_48x48, 0, 0);
}

static void _cbButton_exit(WM_MESSAGE * pMsg) {
    switch (pMsg->MsgId) {
    case WM_PAINT:
        _OnPaint_exit(pMsg->hWin);
        break;
    default:
        /* The original callback */
        BUTTON_Callback(pMsg);
        break;
    }
}

static void _cbDialogInfo(WM_MESSAGE * pMsg) {
    WM_HWIN hItem;
    WM_HWIN hItem1;
    int Id, NCode;
    int i;
    int j;

    switch (pMsg->MsgId) {
    case WM_INIT_DIALOG:

        //hItem1 = WM_GetDialogItem(pMsg->hWin, ID_FRAMEWIN_INFO);
        /*hItem1 = WM_CreateWindowAsChild(0, 0, LCD_GetXSize(), LCD_GetYSize(), pMsg->hWin,
WM_CF_SHOW, NULL, 0); */

        //hItem = SWIPELIST_CreateEx(5, 55, 230, 180, hItem1, WM_CF_SHOW, 0, ID_SWIPELIST_0);
        //hItem = SWIPELIST_CreateEx(5, 55, 230, 180, pMsg->hWin, WM_CF_SHOW, 0,
ID_SWIPELIST_0);
        hItem = WM_GetDialogItem(pMsg->hWin, ID_SWIPELIST_0);
        WM_SetCallback(hItem, _cbSwipe);
        SWIPELIST_SetOwnerDraw(hItem, _SwipeOwnerDraw);

        WM_MOTION_SetMoveable(hItem, WM_CF_MOTION_Y, 1);
        WM_MOTION_SetMoveable(hItem, WM_CF_MOTION_X, 0);
        WM_MOTION_SetSpeed(hItem, GUI_COORD_Y, 20);

        for (j = 0; j < NUM_SEPARATORS; j++) {
            SWIPELIST_AddSepItem(hItem, paSeperatorText[j], 48);
            for (i = 0; i < NUM_ITEMS; i++) {
                SWIPELIST_AddItem(hItem, paltemText[j][i], 48);
            }
        }
        for (i = 0; i < 5; i++) {
            hItem1 = CHECKBOX_Create(0, 0, 32, 32, WM_HBKWIN, 0, WM_CF_SHOW);
            SWIPELIST_ItemAttachWindow(hItem, 1 + i, hItem1, 190, 8);
        }

        //hItem = BUTTON_CreateEx(192, 0, 48, 48, hItem1, WM_CF_SHOW, 0, ID_INFO_EXIT);
        //hItem = BUTTON_CreateEx(192, 0, 48, 48, pMsg->hWin, WM_CF_SHOW, 0, ID_INFO_EXIT);
        hItem = WM_GetDialogItem(pMsg->hWin, ID_INFO_EXIT);
    }
}

```

```

    WM_SetCallback(hItem, _cbButton_exit);

    break;

case WM_PAINT:
    /* Draw information background */
    GUI_SetBkColor(0xFFFFFFFF);
    GUI_Clear();
    GUI_SetColor(0xCCC5B6);
    GUI_FillRect(0, 0, 240, 48);
    break;

case WM_NOTIFY_PARENT:
    Id = WM_GetId(pMsg->hWinSrc); /* Id of widget */
    NCode = pMsg->Data.v; /* Notification code */

    switch (Id) {

    case ID_INFO_EXIT:
        if(NCode == WM_NOTIFICATION_RELEASED)
        {
            /*hItem1 = WM_GetDialogItem(pMsg->hWin, ID_FRAMEWIN_INFO);
            WM_DeleteWindow(hItem1);*/
            GUI_EndDialog(pMsg->hWin, 0);
        }
        break;

    }
    break;

default:
    WM_DefaultProc(pMsg);
    break;
}
}

static void Startup(WM_HWIN hWin, uint16_t xpos, uint16_t ypos)
{
    GUI_CreateDialogBox(_aDialogInfo, GUI_COUNTOF(_aDialogInfo), _cbDialogInfo, hWin, xpos, ypos);
}

```

The most interesting is that when I comment the code the program works as it should, only SWIPELIST widget remains blank

```

/*for (j = 0; j < NUM_SEPARATORS; j++) {
    SWIPELIST_AddSepItem(hItem, paSeperatorText[j], 48);
    for (i = 0; i < NUM_ITEMS; i++) {
        SWIPELIST_AddItem(hItem, paltemText[j][i], 48);
    }
}*/
/*for (i = 0; i < 5; i++) {
    hItem1 = CHECKBOX_Create(0, 0, 32, 32, WM_HBKWIN, 0, WM_CF_SHOW);
    SWIPELIST_ItemAttachWindow(hItem, 1 + i, hItem1, 190, 8);
}*/

```

I tried this code on STM32F207VGT6\_MIKROMEDIYA. It all works. The windows are created equally. But STM32F412G-Discovery hangs when you press the EXIT. The code window Call STM32F207VGT6\_MIKROMEDIYA

```
void StartDefaultTask(void const * argument)
{
    GUI_Init();

    while(1)
    {
        switch(appState){
        case MAIN:
            hWin = CreateWindowMAIN();
            appState = GUI_ExecCreatedDialog(hWin);
            break;
        case PLAYER:
            hWin = CreateWindowPLAYER();
            appState = GUI_ExecCreatedDialog(hWin);
            break;
        case RECORD:
            hWin = CreateWindowRECORD();
            appState = GUI_ExecCreatedDialog(hWin);
            break;
        case FOLDER:
            hWin = CreateWindowFOLDER();
            appState = GUI_ExecCreatedDialog(hWin);
            break;
        case SETTING:
            hWin = CreateWindowSETTING();
            appState = GUI_ExecCreatedDialog(hWin);
            break;
        default:
            break;
        }
    }
}
```

I tried in different ways. In some cases, the window appears on the scrolling ICONVIEW (ie, the window moves and gets out of the side ICONVIEW). In some cases, after clicking on the exit window remains, and after him there ICONVIEW (and continues to work correctly, you can still run 20 times this module and freezes .. memory ends). The feeling that is inherited is not correct or parent loses. With other widgets such is not present (at least with simple widgets). How do SWIPELIST to work properly?

