ARM XMC - experiment No. 5

• E5.1 Name:

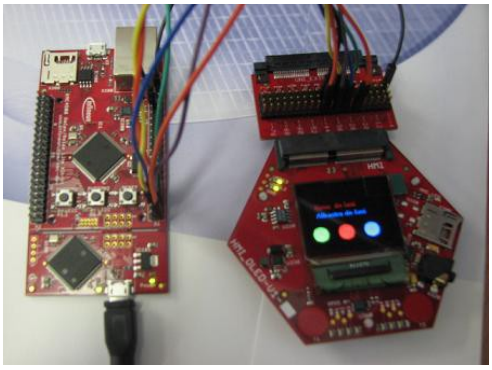
Infineon Relax kit – OLED for instrumentation

• E5.2 Overview and purpose

This experiment aims to explore Human Machine Interaction type resources using Infineon development boards. We will introduce the OLED displays with SEP525 controller and the emWIN5 Segger library for embedded applications.

At the end of this experiment, you will have detailed information about how to realize a graphical interface for instrumentation for Infineon Relax Kit platform.

• E5.3 Resources



Hardware	XMC4500 Relax kit HMI board Connexion wires
Software	Infineon DAVE 3.1; Segger emWin5 graphical library

- **E5.4 Software example:**

Figure 1 shows the functional scheme of the prototype program.

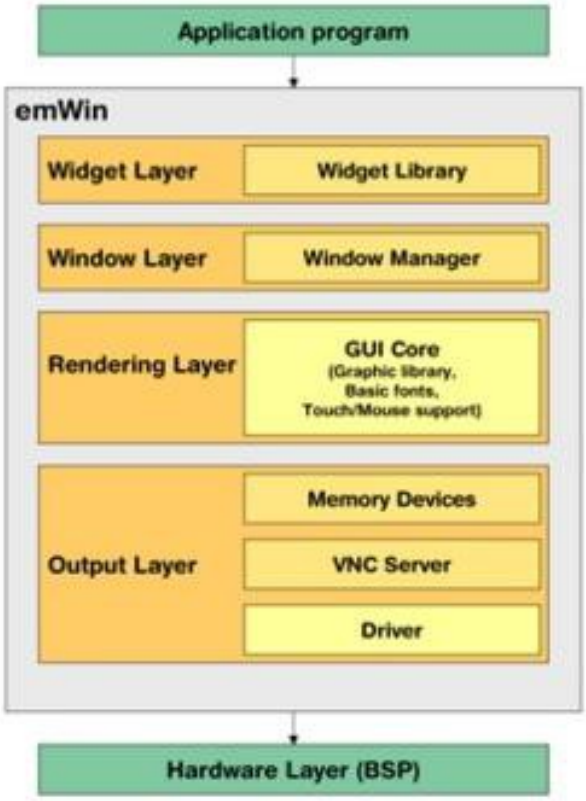


Figure 5.1 The functional scheme of the prototype program

```
/*
 * Hello_color_Iasi - prototype program
 */

#include <DAVE3.h>
#include "GUI.h"

int i;
int main(void)
{
  /* Dave Apps Initialization */
  DAVE_Init();
```

```
  /* Initialize the GUI */
  GUI_Init();

  while(1)
  {
      GUI_Clear();

      GUI_SetFont(&GUI_Font16B_1); // set the font

      GUI_SetColor(GUI_GREEN);  // set the color
      GUI_DispStringAt("Verde de Iasi", 18, 18); // show message

      GUI_SetColor(GUI_RED);
      GUI_DispStringAt("Rosu  de Iasi", 18, 36);

      GUI_SetColor(GUI_BLUE);
      GUI_DispStringAt("Albastru de Iasi", 18, 54);

      for(i=1;i<15;i++)
            GUI_DrawCircle(120, 100, i); //blue button

      GUI_SetColor(GUI_RED);
      for(i=1;i<15;i++)
            GUI_DrawCircle(70, 100, i);  //red button

      GUI_SetColor(GUI_GREEN);
      for(i=1;i<15;i++)
            GUI_DrawCircle(20, 100, i);   // green button
  }
}
```

- **E5.5 Method of running experiment:**
  - Analyze the interconnection scheme from Relax Kit to HMI (Figure 2);

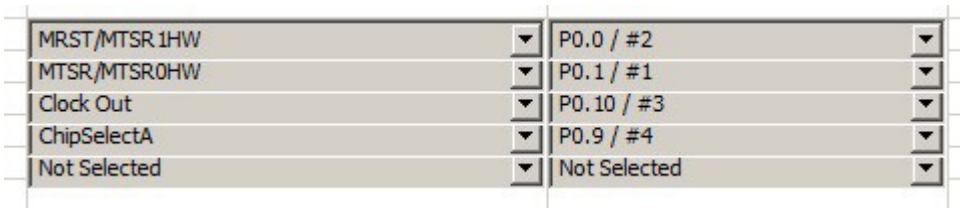


Figure 5.2 The interconnection scheme

  - Analyze the interconnection code from SEP256F module to ARM Cortez M4;

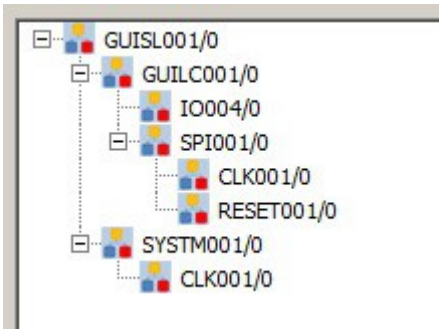


Figure 5.2 The GUI and System Timer modules

- Explore the functions of the emWin2 Segger library;
- Create a DAVE project with the source code from the prototype program and follow the execution on Relax Kit platform;
- Resolve the proposed problems.

• E5.6 Problems proposed:

1. Create a program that allows to show the floating-point numbers (Figure 4);

Main.c GUI.h

```
*       Drawing of binary, decimal and hexadecimal values
*/
void GUI_DispBin  (U32  v, U8 Len);
void GUI_DispBinAt(U32  v, I16P x, I16P y, U8 Len);
void GUI_DispDec  (I32 v, U8 Len);
void GUI_DispDecAt (I32 v, I16P x, I16P y, U8 Len);
void GUI_DispDecMin(I32 v);
void GUI_DispDecShift(I32 v, U8 Len, U8 Shift);
void GUI_DispDecSpace(I32 v, U8 MaxDigits);
void GUI_DispHex  (U32 v, U8 Len);
void GUI_DispHexAt(U32 v, I16P x, I16P y, U8 Len);
void GUI_DispSDec(I32 v, U8 Len);
void GUI_DispSDecShift(I32 v, U8 Len, U8 Shift);

/*********************************************************************
*
*       Drawing of floating point values
*/
void GUI_DispFloat    (float v, char Len);
void GUI_DispFloatFix (float v, char Len, char Fract);
void GUI_DispFloatMin (float v, char Fract);
void GUI_DispSFloatFix(float v, char Len, char Fract);
void GUI_DispSFloatMin(float v, char Fract);

/*********************************************************************
```

Figure 5.3 The GUI header

2. Using the right methods (Figure 5), create a program that will display a classing board indicator.

Main.c GUI.h

```
*       2d - GL
*/
void GUI_Clear              (void);
void GUI_ClearRect          (int x0, int y0, int x1, int y1);
void GUI_ClearRectEx        (const GUI_RECT * pRect);
void GUI_CopyRect           (int x0, int y0, int x1, int y1, int
void GUI_DrawArc            (int x0, int y0, int rx, int ry, int
void GUI_DrawBitmap         (const GUI_BITMAP GUI_UNI_PTR * pBM,
void GUI_DrawBitmapMag      (const GUI_BITMAP GUI_UNI_PTR * pBM,
void GUI_DrawBitmapEx       (const GUI_BITMAP GUI_UNI_PTR * pBM,
void GUI_DrawBitmapExp      (int x0, int y0, int XSize, int YSize
void GUI_DrawBitmapHWAlpha(const GUI_BITMAP GUI_UNI_PTR * pBM,
void GUI_DrawCircle         (int x0, int y0, int r);
void GUI_DrawEllipse        (int x0, int y0, int rx, int ry);
void GUI_DrawGradientH      (int x0, int y0, int x1, int y1, GUI_
void GUI_DrawGradientV      (int x0, int y0, int x1, int y1, GUI_
void GUI_DrawGradientRoundedH(int x0, int y0, int x1, int y1, i
void GUI_DrawGradientRoundedV(int x0, int y0, int x1, int y1, i
void GUI_DrawGraph          (I16 * pay, int NumPoints, int x0, ir
void GUI_DrawGraphEx        (I16 * pay, int NumPoints, int x0, ir
void GUI_DrawHLine          (int y0, int x0, int x1);
void GUI_DrawLine           (int x0, int y0, int x1, int y1);
```

Figure 5.4 The methods from the emWIN5 library

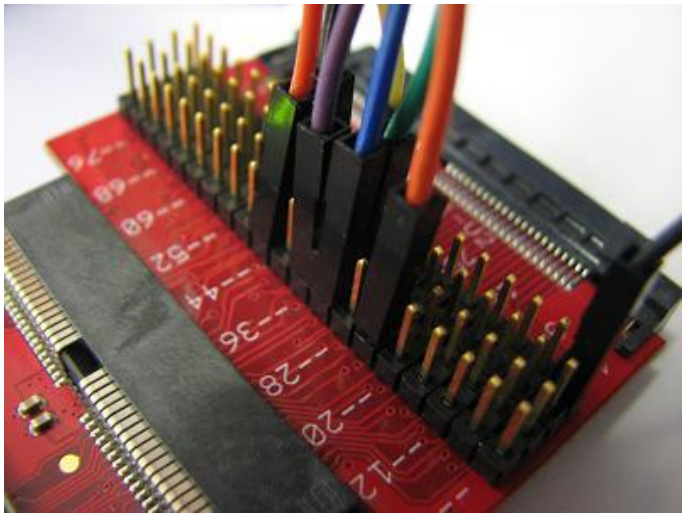


Figure 5.5 The OLED pins connections

3. Create a program that will draw colored squares and circles;
4. Create a program that will draw a moving ball inside a square;

Main.c GUI.h

```
void GUI_DrawLine          (int x0, int y0, int x1, int y1);
void GUI_DrawLineRel       (int dx, int dy);
void GUI_DrawLineTo        (int x, int y);
void GUI_DrawPie           (int x0, int y0, int r, int a0, int a
void GUI_DrawPixel         (int x, int y);
void GUI_DrawPoint         (int x, int y);
void GUI_DrawPolygon       (const GUI_POINT * pPoints, int NumPo
void GUI_DrawPolyLine      (const GUI_POINT * pPoints, int NumPo
void GUI_DrawFocusRect     (const GUI_RECT  * pRect, int Dist);
void GUI_DrawRect          (int x0, int y0, int x1, int y1);
void GUI_DrawRectEx        (const GUI_RECT * pRect);
void GUI_DrawRoundedFrame  (int x0, int y0, int x1, int y1, int
void GUI_DrawRoundedRect   (int x0, int y0, int x1, int y1, int
void GUI_DrawVLine         (int x0, int y0, int y1);
void GUI_FillCircle        (int x0, int y0, int r);
void GUI_FillEllipse       (int x0, int y0, int rx, int ry);
void GUI_FillPolygon       (const GUI_POINT * pPoints, int NumPo
void GUI_FillRect          (int x0, int y0, int x1, int y1);
void GUI_FillRectEx        (const GUI_RECT * pRect);
void GUI_FillRoundedFrame  (int x0, int y0, int x1, int y1, int
void GUI_FillRoundedRect   (int x0, int y0, int x1, int y1, int
void GUI_FillRoundedRectB  (int x0, int y0, int x1, int y1, int
void GUI_FillRoundedRectT  (int x0, int y0, int x1, int y1, int
void GUI_GetClientRect     (GUI_RECT * pRect);
void GUI_InvertRect        (int x0, int y0, int x1, int y1);
void GUI_MoveRel           (int dx, int dy);
void GUI_MoveTo            (int x, int y);
```

Figure 5.6 the method to be used in program 4

5. Create a program that will display binary data;

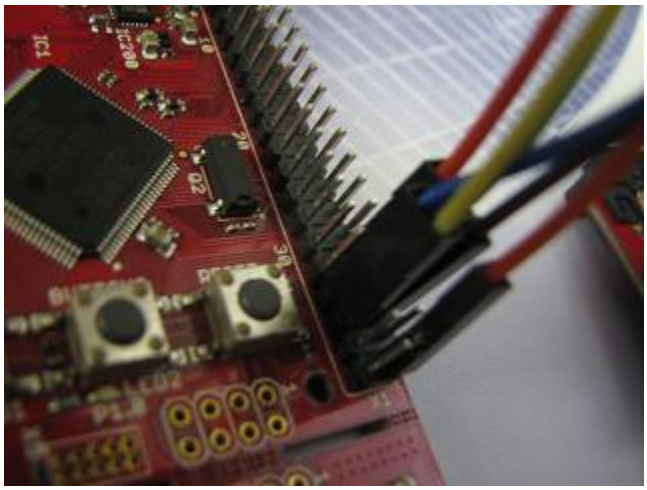


Figure 5.7 The connection pins on Relax Kit

- **E5.7 The experiment can be extended to be used for:**
  - Making active nodes with color graphic interface;
  - Making smart instrumentation with graphical display;
  - Making board systems for automobiles;
  - Making mobile systems for processing images / sounds;

Figure 5.8 The prototype program result

- **E5.8 More helpful information:**

1. **Segger -** (https://www.segger.com/emwin.html)
2. **Display Types** - (https://www.segger.com/emwin-display-drivers.html)
3. **Virtual instruments** - (https://www.segger.com/emwin-virtual-screen-support.html)
4. **Infineon HMI extension documentation** - (http://www.infineon.com/dgdl/Board_Users_Manual_Standard_HMI_Card_R1.0.pdf?fileId=db3a304335f1f4b60135f24d0fe50038)
5. **Segger emWin5 graphical library documentation** - (https://www.google.ro/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwiMq_GI_bzJAhXCtBoKHUMmDE4QFggbMAA&url=https%3A%2F%2Fwww.segger.com%2Fdownloads%2Femwin%2FUM03001_emWin5.pdf&usg=AFQjCNGriPxyOkHZwRVRQwVzMFyRQM5z8w&cad=rja)
6. **Operating principles** - (http://www.oled-info.com/oled-technology)

7. **Segger - embedded software solution Internet** - (https://www.segger.com/emwin.html)
8. **Segger RTOS** - (https://www.segger.com/admin/uploads/productDocs/UM01001_embOS_Generic.pdf)
9. **Future Display** - (http://spectrum.ieee.org/computing/hardware/the-electronic-display-of-the-future/0)