1. Project title: Human machine interface

Student: Balaita Madalina – Elena
Mail address: madalinae.balaita@gmail.com
Student: Dumea Liviu
Mail address: liviu.dumea.ro@gmail.com
Student:Negrut Alina
Mail address: alinane92@gmail.com

2. Abstract

People will always be interested in gadgets and different ways to manipulate objects more easily.Exploring the XMC4500 Relax Kit and the HMI OLED V1 was an interesting experience for us to discover more and more things about the embedded world.

3. Introduction, project aims and objectives

This hardware components are used often in developing almost for any kind of applications.Using the HMI OLED V1 we can create mini projects like using leds, buttons, touch buttons, micro sd card , passive matrix OLED dislplay or stereo audio application or create a more complicated application where we can put them to interact with each other .
Our system is based on readings from this two sensors and put informations on display.We used touch buttons from the HMI OLED and buttons from XMC4500 to change the way informations are displayed.Also our intentions was to interact with the 8 leds from HMI OLED.

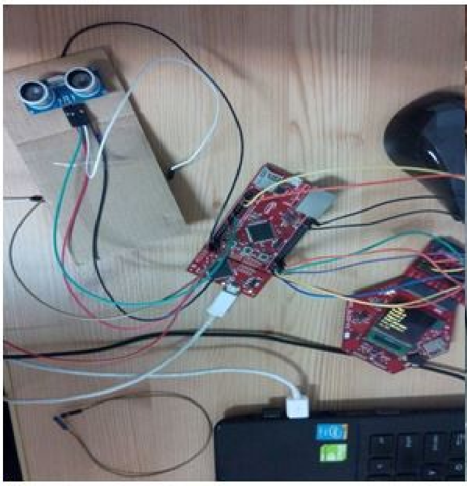
4. System overview

The system is composed from the following hardware components:

- XMC4500 relax kit evaluation board
- HMI OLED V1
- LM35 temperature sensor
- HC SR 04 distance sensor
- Wires

The system is made from the XMC4500 Relax Kit microcontroller (ARM® Cortex™-M4F based) connected with the HMI OLED V1 and two sensors.
For writing code we used DAVE 3 tool.

5. Schematics and components



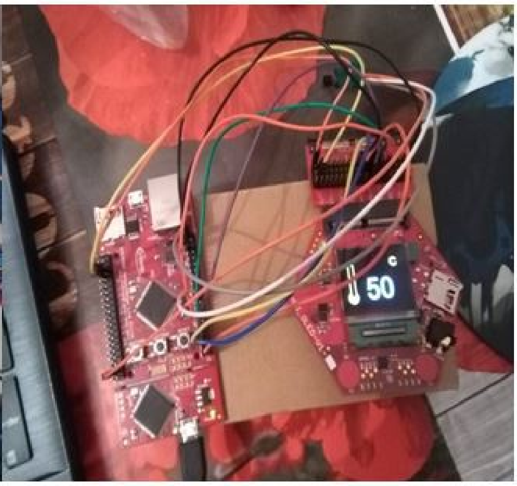


Fig1. Distance sensor

Fig2. Display format on HMI OLED V1

For analogical readings we used the ADC from the XMC4500 Relax Kit board. This development board is produced by Infineon and is based on ARM Cortex M4F CPU running at 120MHz, 1MB Flash and 160KB RAM.It also has an Ethernet port and allows equipping it with a microSD card slot.

LM35 is a temperature sensor is an integrated-circuit temperature device that can output a voltage linearly-proportional with the temperature. The analog output can be read with an ADC and transformed in Celsius grades by dividing the voltage given in mV by 10.

The HC-SR04 ultrasonic sensor uses sonar to determine distance to an object like bats or dolphins do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package. From 2cm to 400 cm or 1” to 13 feet. We had to provide the TRIGGER signal, for at least 10µS High Level (5V) pulse. The module will automatically transmit eight 40KHz ultrasonic burst. If there is an obstacle in-front of the module, it will reflect the ultrasonic burst. If the signal is back, ECHO output of the sensor will be in HIGH state (5V) for a duration of time taken for sending and receiving ultrasonic burst. Pulse width ranges from about 150µS to 25mS and if no obstacle is detected, the echo pulse width will be about 38ms.

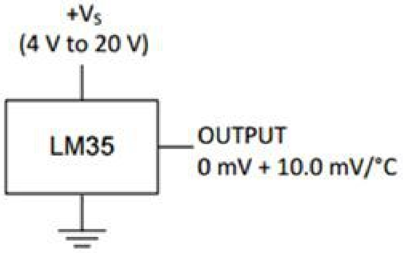


Fig. 3 LM35 connection

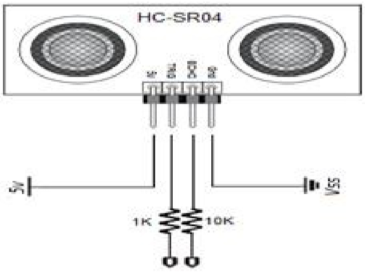


Fig. 4 Distance sensor conection

6. Software

For this project we used Dave 3 tool. This is a free eclipse-based IDE that can be use to develop applications oriented or component based programs for XMC development boards.

From software point of view the following DAVE applications are used:

- Timer/PWM (Capture, Compare)
- ADC
- DAC
- GPIO

Some code example :

```
void draw_thermometer(int x, int y, int v) {
        int circleR = 13;
        int barW = 14;
        int barH = 75;
        int posx = x + circleR/2;
        int posy = y + barH;
        int barx = x + (circleR-barW)/2;
        int bary = y;
        int barH2 = barH * (100 - v) / 100;
        GUI_SetColor(GUI_YELLOW);
        GUI_FillCircle(posx, posy, circleR);
        GUI_FillRect(barx, bary, barx + barW, bary + barH);
        GUI_ClearRect(barx+2, bary+2, barx + barW - 2, bary + barH2 - 4);
}
void draw_UM(int x, int y) {
        GUI_SetFont(&GUI_Font32B_1); // stabilire bfont
        GUI_SetColor(GUI_YELLOW); // stabilire culoare
        if(modAfis == 0) {
                GUI_DispStringAt("C", x, y); // afisare mesaj
        } else {
                GUI_DispStringAt("F", x, y); // afisare mesaj
        }
}
void draw_val(int x, int y, int v) {
        GUI_SetFont(&GUI_FontD48); // stabilire bfont
        GUI_SetColor(GUI_WHITE); // stabilire culoare
        GUI_DispDecAt(v, x, y, 2);
}
void draw_time(int x, int y) {
        int time;
        time = baseTime + timer2_ticks;
        int s = time % 60;
        int m = time /60 % 12;
        GUI_SetFont(GUI_FONT_24B_1);
        GUI_SetColor(GUI_WHITE); // stabilire culoare
        GUI_DispDecAt(m, x, y, 2);
        if (time %2 == 0) {
                GUI_DispString(":");
        } else {
                GUI_DispString(" ");
        }
        GUI_DispDec(s, 2);
}
void GlobalResultEvent(void) {
        ADC001_GetResult(&ADC001_Handle0, &Result);
}
int main(void) {
        ..............................................
        /* Dave Apps Initialization */
        DAVE_Init();
        /* Initialize the GUI */
        GUI_Init();
        ADC001_GenerateLoadEvent(&ADC001_Handle0);
        timer1_active = 0;
        timer2 = SYSTM001_CreateTimer(1000,
                                                                        SYSTM001_PERIODIC, timer2_callback, NULL);
        if (timer2 != 0) {
                //Timer is created successfully
                Status = SYSTM001_StartTimer(timer2);
                if (Status == DAVEApp_SUCCESS) {
                        timer2_ticks = 0;
                }
        }
        while (1) {
                ADC001_GetResult(&ADC001_Handle0, &Result);
                val1 = IO004_ReadPin(IO004_Handle1);
                val2 = IO004_ReadPin(IO004_Handle2);
                button1 = IO004_ReadPin(IO004_Handle5);
                gradeK = ((Result.Result/4095.0)*3300)/10 ;
                gradeC = gradeK - 273;
                gradeF = ((gradeC*9)/5+32);
                if(redraw) {
                        draw();
                        redraw = 0;
                        ADC001_GenerateLoadEvent(&ADC001_Handle0);
                }
                ...............................
}
```

7. Project results & applications

We experienced difficulties reading the two sensors and handling the 8 LEDs connected to i2c in DAVE 3 tool and also we had limitations on using RTOS applications because display uses all the neccesary resources.

8. Reference

1. www.embedac.ro,

2. www.infineon.com,

3. www.segger.com

4. http://www.micropik.com/PDF/HCSR04.pdf

5. http://www.ti.com/lit/ds/symlink/lm35.pdf