17.2.2016.

Microprocessor systems (IR4MS) project

Miloš Stojanović

Table of contents

Problem definition:

Making a device for measuring and displaying the temperature. The temperature is measured with the thermal sensor, shown on the display and the display mode is controlled by the joystick. If the temperature exceeds the defined boundaries the appropriate LED will start blinking.

Hardware description and connection scheme:

Clicker 2 board, joystick, thermal sensor, display and a 10kΩ potentiometer (or a resistor assembly).

***Note:** the thermal sensor, display and potentiometer need a 5V source. The joystick occupies one on BUS 1 and the only remaining source is on BUS 2 which is shared by the remaining components.

• **Display (WH1602B-TMI-ET) and 10kΩ potentiometer:**

The display and potentiometer are connected to the right side of the board (the one with the USB port) except for one GND pin which is on the left side. The background contrast should be tuned with the potentiometer.

Table of display connections:

Pin num.	Name on the display	Place on Clicker 2
1	Vss	GND (lower right)
2	Vdd	5V (mirko BUS 2)
3	VO	to potentiometer
4	RS	PD10
5	RW	GND (upper right)
6	E	PD11
11	DB4	PC7
12	DB5	PB7
13	DB6	PD13
14	DB7	PD14

* pins 7, 8, 9, 10, 15 and 16 aren't connected because the display uses 4-bit mode and the backlight isn't being used

The potentiometer is connected to GND (upper left), 5V (mikro BUS 2) and the VO pin on the display.

• **Thermal sensor (DS1820):**

The thermal sensor is connected on the left side of the board (the side with the on/off switch) except for the Vdd which is on the BUS 2 5V pin. DQ is connected to the PC13 pin and GND is in the lower right corner.

* There's no need to connect a resistor to the DQ pin because a software enabled resistor from the Clicker 2 board is being used.

• **Joystick click™ (AS5013):**

Connected to mikro BUS 1.

Solution description:

The launched starts with initialisation. This usually consists of enabling the clock for the required peripheries, pin definitions and interrupt definitions. The wait instruction uses the system clock which is initialised to 128 MHz. The reason for that is that the wait instruction requires a resolution of 1 μs, primarily because of the thermal sensor.

The **2 buttons** on the board itself are also initialised. They can be used for testing (button 1 has the same functionality as the button on the joystick and button 2 is the same as sliding the joystick to the right). After the initialisation, communication with the components is established.

Initially a hardware reset signal is sent to the **joystick**, then communication is established through the I²C protocol. Availability is checked and an additional bit is written to register 1 which signals that the interrupt regime wont be used. Instead the CPU will read and process the values of joystick registers every 100 ms, which is enough for normal use. After every reading, a function is called that's fits the direction of the joystick. That function is user implemented and is used for interactions with the menu.

The **display** uses it's own protocol which can be found in the data-sheet. It's started in the 4-bit mode. Different printing functions, as well as functions for defining unique characters can be found in the my_display.c file.

The **thermal sensor** uses the One Wire protocol. Before reading the temperature a command demanding a conversion of the temperature into sensible register values must be sent. Since it takes some time for the conversion to finish the unnecessary waiting is avoided by first sending the conversion command and after that, every second the values are read and a new conversion request is sent. With every temperature read a check is preformed to see if the set boundaries have been breached and if so LED blinking is activated.

At the end the **menu** is initialised and run. Its role is to visually depict and control the movement through the menu windows and print the cursor. The menu windows and selectable elements that can be found in them behave like objects, they need to be initialised and hierarchically assigned. Since the Clicker 2 doesn't have a heap on its own they need to be allocated on the stack. The menu window can be declared as a radio window and static text or a function for writing user controlled text can be assigned to it. When entering a menu window or chosing a radio option the return value will be the uniques number of that windown and it can be used for ineracting with the menu.

After all the initialisations are done the CPU enters a loop of sleeping and awaiting of interrupts than can be processed.

User manual:

Its best to orient the board horizontally so that the display is nearer and the joystick further form the user. In this way the joysticks directional commands are aligned with the orientation. Moving the joystick left and right goes through the the menu window items in the order they were added. Moving the joystick up and down goes to the item that is in the next closest row to the current cursor position. All the movement commands cyclically return the cursor after it reaches the end. Accessing a specific item is done by pressing the joystick button.

The initial text should appear on the display after starting the device. It contains the item “Temperature” which leads to the window for displaying, formatting and adjusting the alarm of the temperature. The item “Other” leaves room for further expansions and so it can't be accessed. Every menu window except the initial contains a symbol ’↲’ in the lower right corner. Its function is to return to the previous window in the hierarchy. In the window “Show current” the temperature is shown in the selected format and it’s periodically refreshed which is indicated with the moving squares. In the options window things like the number of decimal places, the temperature scale and the maximum and minimum temperature that will trigger the alarm can be adjusted. The first two settings functions like normal radio menus while in the alarm setting, one can choose the increment (in ºC) with which the maximal or minimal temperature will be increased or decreased. If at any time the temperature goes beyond the set boundaries, LED1 or LED2 will start blinking depending if it was the upper or lower bound, respectively.

Pictures: