Laboratory Project
Embedded Computational Systems
IST - 2020/2021

Monitoring and Alarm System (Part 1 – PIC)

1 Introduction

Many embedded systems are developed using relatively simple platforms, with low resources, where the utilization of operating systems to support the application may not be viable. In these type of systems there is essentially the need to do some elementary processing and access several input / output devices (e.g. access sensors to collect information). However, in several other cases, the existence of support at operating system level (even if with reduced functionality) may significantly facilitate the development of applications.

The laboratory project to be implemented (composed of 2 parts) has as main goal the familiarization of students with the development of embedded systems, both with medium / low complexity, using microcontrollers and without the support of an operating system, and also using multitasking kernels for the development of concurrent applications.

In particular, they should acquire some expertise in the utilization of communication and synchronization mechanisms between tasks, in the context of concurrent applications, and should familiarize with other embedded systems characteristics such as: access devices using simple digital input/output lines or network/buses such as I2C (Inter Integrated Circuit) or SPI (Serial Peripheral Interface); save information in non-volatile memory devices (EEPROM); utilization of analog-digital conversion; programming timers; use interrupts; and use of serial communication RS232.

They should also be aware of aspects related to energy consumption, resorting to operation modes that will allow as much as possible to increase system autonomy.

2 Problem description

The project is decomposed into two distinct parts, that can be interconnected later. This first part corresponds to a monitoring and alarm system that has a rudimentary user interface (switches, LEDs, LCD) and no operating system support. The other part, with a more elaborated user interface and processing capability, has a multitasking kernel to support a concurrent application and will allow remote access (using a RS232 serial line) to the system developed in the first part.

2.1 Monitoring and alarm system

The first part of the project is implemented using the development board **"Curiosity High Pin Count Development Board"** [1], which includes a microcontroller **PIC16F18875** [2]. The application is programmed using the **C programming language** ("MPLAB XC8 C Compiler" [3, 4, 5]) and the development environment "MPLABX Integrated Development Environment (IDE)" [6, 7] from Microchip. Reading the related documentation is fundamental for the correct project implementation.

The main functions to provide are essentially:

- Clock (to keep current time and to allow alarm management and timestamping of relevant events).
- Periodic sensor reading (with storing of data and the possibility of handling alarm situations).
- Basic user interface.
- Incorporation of power-saving mechanisms.
- Remote access support (using RS232 communication) to allow reconfiguration operations and data transfer (**to be implemented in conjunction with the second part of the project**).

The clock is built based on one of the timers available in the PIC, and must provide the current time in the form of hours, minutes and seconds.

The system will monitor in a periodic fashion (period `PMON`) the values of temperature (sensor TC74 [9] – to be connected to the board) and "luminosity" (emulated using the potentiometer available in the developing board and converting the obtained value into 8 different levels {0,...,7}). Depending on the values collected, they may be saved in non-volatile memory (PIC's internal EEPROM), and associated with alarm situations as well. If `PMON` is zero there will be no periodic information collected.

Saving the collected values in non-volatile memory is only performed if at least one of them (temperature or luminosity) is different from the one previous saved. They are saved as a register that, in addition to the values obtained from the sensors, also contains a timestamp with the current time. It is possible to save up to `NREG` registers in a "ring-buffer" (when the buffer is full, new registers replace older registers). The size of a register is 5 bytes (h,m,s,T,L – corresponding to the timestamp (hours, minutes, seconds) and the values of temperature and luminosity).

When enabled, alarms are generated when the specified alarm time is reached or when the new values of temperature or luminosity are respectively **above** or **below** pre-defined thresholds. The notification of an alarm situation is done through signalization on the LCD (letters CTL, respectively) **and** generation of a PWM signal that is applied to one LED changing its brightness. The letter will stay in the LCD until user interaction (switch S1). The PWM signal will have a duration of `TALA`.

Relevant configuration parameters for the correct system operation should be preserved in non-volatile memory (PIC's internal EEPROM) so as to try to use the latest defined values after a reset operation, if possible (validation using a "magic word" and a checksum). They should be updated in the EEPROM when they are modified. The clock value (only hours and minutes) should also be saved periodically (every minute). (Some reconfiguration commands will be available only in the second part.)

Initial values for those parameters are:

NREG	25	number of data registers
PMON	3 sec	monitoring period
TALA	5 sec	duration of alarm signal (PWM)
ALAH	12	hours of alarm clock
ALAM	0	minutes of alarm clock
ALAS	0	seconds of alarm clock
ALAT	28 °C	threshold for temperature alarm
ALAL	4	threshold for luminosity level alarm
ALAF	0	alarm flag – initially disabled
CLKH	0	initial value for clock hours
CLKM	0	initial value for clock minutes

2.2 Remote access

To be implemented in Part-2 of Laboratory Project.

3 User interface

3.1 Monitoring and alarm system

The user interface, in this part of the project, is performed through the use of 2 switches (S1, S2 – connected to RB4 and RC5, respectively), 4 LEDs (D2-D5 – connected to RA4-RA7), and an LCD (2 lines of 16 characters – to be connected to the board) [8]. The switches are used for setting the correct time of the clock, to define alarms, and to select operation mode. The LEDs and LCD are used to show the state of the system, and may have different meanings in normal mode and in modification mode.

The information presented on the LEDs, in normal mode, will have the following meaning:

C	A	T	L
X	X	X	X
D5	D4	D3	D2
RA7	.	.	RA4

`C` (LED D5)	clock activity (blink – toggling every 1 second)
`A` (LED D4)	alarm (when alarm occurs, change in brightness – PWM)
`T` (LED D3)	temperature is above threshold
`L` (LED D2)	luminosity is below threshold

The information presented on the LCD will have the following aspect (some fields may have different meanings in normal mode and modification mode):

hh:*mm*:*ss*	`CTL A`
tt `C`	`L` *l*

In normal mode, current values of clock (“`hh:mm:ss`”), temperature (“`tt`”) and luminosity level (“`l`”) are presented. Letter “`A`” (or “`a`”) will indicate if alarms are enabled (or disabled). Letters

"`CTL`" will appear only in the case where the respective alarm (clock, temperature or luminosity) has occurred.

Switch S1 (RB4) is used to select the desired operation, and the switch S2 (RC5) is used to change the selected fields, or enable the desired operation.

More precisely, starting from normal operation mode, pressing S1 will make the system enter into modification mode allowing the modification of the first modifiable field (Clock – hours). The cursor will blink over that field, and it will be possible to increment the value of the field by pressing S2. Whenever S1 is pressed the cursor will move to the next field, until reaching normal operation mode again. When S2 is pressed, the value of the field that is being modified is incremented module the range of possible values for that field, or the given function is enabled.

Range values:

hh - hours [00 .. 23]

mm - minutes [00 .. 59]

ss - seconds [00 .. 59]

tt - temperature [00 .. 50]

l - luminosity level [0 .. 7]

In modification mode, "`tt`" and "`l`" will present the current threshold values for temperature and luminosity alarms. The letters "`CTL`" allow the selection for modification of the alarm thresholds: clock, temperature and luminosity, respectively. When the cursor is on top of a given letter, the current specified alarm threshold value should be presented in the respective field (clock, temperature or luminosity). If that function is selected (user presses S2), the cursor is placed on top of that current value, allowing its modification.

The letter "`a`" (or "`A`" – it works in "toggle" mode) shows if the alarms are disabled or enabled, and make it possible to enable/disable the alarms (all of them).

The characteristics of the IO devices that are used should be consulted on the manual of the board [1], and/or on the "Data Sheets" of the devices.

4 Project development

In the development of the project, the utilization of a modular structure and a phased testing is advised.

In the target board there will be no operating system to support the execution of the application. It is students responsibility to structure the program in a modular fashion according to the several tasks that must be performed. The execution support to those tasks should be organized in the form of a "cyclic executive", resorting to the use of interrupts in the situations where that is justifiable. If needed, a state machine approach should also be used.

Underlying all aspects of project implementation, there should be the concern of, without jeopardizing the desired functionality, trying to optimize energy consumption, resorting to the instruction "`SLEEP()`" (power saving mode) whenever possible.

5 Project delivery

This part of the project should be delivered by $\boxed{08/11/2020}$.

The delivery consists of a digital copy (ZIP file) of all developed programs. All these elements should be correctly identified (group number and students). Project demonstrations (to be done later) will be based on the delivered material.

References

[1] Microchip Technology Inc. *Curiosity High Pin Count Development Board User's Guide*. 2016-2018.

[2] Microchip Technology Inc. *PIC16(L)F18855/75 Data Sheet*. 2015-2018.

[3] Microchip Technology Inc. *MPLAB XC8 C Compiler User's Guide for PIC.* 2012-2018.

[4] Microchip Technology Inc. *MPLAB XC8 Getting Started Guide.* 2013.

[5] Microchip Technology Inc. *MPLAB XC8 C Compiler User's Guide*. 2011-2016.

[6] Microchip Technology Inc. *MPLABX IDE User's Guide*. 2011-2015.

[7] Microchip Technology Inc. *MPLAB Code Configurator v3.xx User's Guide*. 2018.

[8] Hitachi. *HD44780U (LCD-II) (Dot Matrix Liquid Crystal Display Controller/Driver).*

[9] Microchip Technology Inc. *TC74 - Tiny Serial Digital Thermal Sensor*. 2002.