Development board for the microcontroller lab

Fernando Pardo, Jose A. Boluda, Esther de Ves

Departamento de Informática - Universidad de Valencia
Avda. Vicente Andrés Estellés s/n, 46950, Burjassot, Valencia, SPAIN
Fernando.Pardo@uv.es, Jose.A.Boluda@uv.es, Esther.DeVes@uv.es

Abstract

The laboratory dedicated to microcontrollers is present at some University studies, especially at electrical engineering, electronics engineering and computer engineering. We present in this article the development of a board based on PIC microcontrollers, which features have been specially suited for laboratory experiments and education, in fact we call our lab board “EduPic” (Educational PIC).

1. Introduction

For several years we have taught microcontrollers using standard development boards based on the 8051 microcontroller family. The lab sessions were fine but we found several improvements that could be achieved: our set of peripherals was not very high since most of the board was dedicated to the microcontroller itself; program download was performed on-line but it was written to a RAM externally to the microcontroller, thus small or stand alone systems were not possible to be implemented or tested; we were also constrained to the existing “Operating System” to run the program. Other problems had to do with the features of the microcontroller employed: usually these boards come with a specific microcontroller with some fixed features that cannot be changed; even when there is the possibility of changing the microcontroller, it happens sometimes that the new one does not match the present peripherals of the board.

For these reasons, we considered changing to a new strategy for our lab sessions dedicated to microcontroller teaching. First we find that Microsystem’s family of microcontrollers (PIC) offered not one but many different microcontrollers suitable for almost any application, with most things in common making them interchangeable and easily upgradeable [1]. This is especially important since we wanted a platform to develop a wide broad of experiments impossible to cover with a single microcontroller. For example, we use the board mainly for the microcontroller lab, but it is also employed for the computer peripheral course where the USB port is programmed as a slave device; in each case we use the PIC that best suits the task to be performed (USB teaching or plain microcontroller teaching).

Our microcontroller lab is specially addressed to teach the development of stand-alone and embedded systems, which are the best places where a microcontroller fits. This is the reason why we introduced many peripherals in a single board; in one hand it is more comfortable than having extra boards around, and in the other hand, the program tested in the board will have almost the same environment as a real development

Figure 1 shows a picture of the board. The different peripherals and board elements have been separately placed for clarity and pedagogic reasons. Complete information about this board, software, examples and manuals can be found at [2].

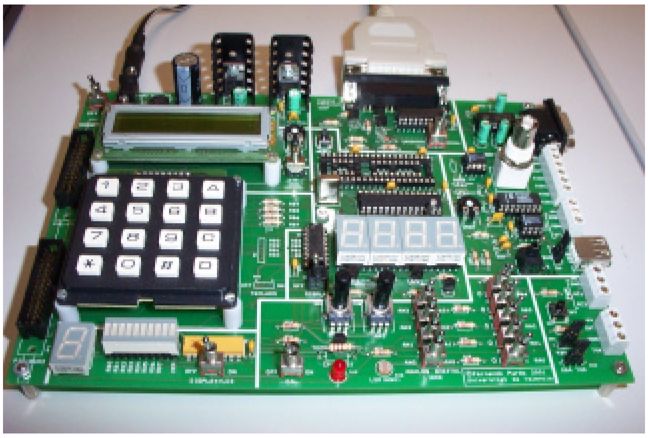

Fig. 1. *Picture of the EduPIC board*

Main characteristics and features of the board follow:

- It is prepared to support PIC families 12, 16 and 18 (DIP packages of 8, 18, 28 and 40 pins)
- On board programming of PIC families 16 and 18 supporting ICSP (In Circuit Serial Programming) which are most of them. Programming is performed through the parallel port with software that supports W95, W98, NT, W2000, WMe, XP and Linux.
- One programming switch is provided to not interfere with peripherals during programming.
- Alphanumeric standard LCD with 2x16 characters.
- Matrix keyboard with 4x4 keys.
- Four integrated analogue sources:
  - 2 lineal trimmers (0-5 V)
  - 1 temperature sensor.
  - 1 light sensor.
- One resistor is provided to heat the temperature sensor.
- LED bar with a line of 8 leds.
- One single 7-segment display.
- Set of four 7-segment display connected to a single BCD to 7-segment driver.
- 5 digital inputs selectable (0/1) by switch.

- Analogue or digital common inputs selectable thanks to three state switches (Analogue/Free/Digital). These switches allow disconnection of individual digital or analogue inputs.
- I2C bus peripherals:
- PCF8591: Single chip A/D and D/A converter, one analogue output and five analogue inputs.
- PCF8582: E2PROM memory with 256 bytes.
- Serial RS232 connection, driver and interface.
- USB interface (only for PICs supporting USB).
- Press Button to generate external interrupt (INT).
- Individual enable switches for each peripheral on the board.
- Individual connectors for I2C lines, analogue inputs and outputs of the I2C bus, PWM (RC1 and RC2), external interrupt and external timer0 clock (T0K).
- Two extra IDC 26 pin connectors provide access to any signal of the on board PIC. First connector provides all signals of PICs up to 28 pins, while second connector adds the extra lines of larger PICs.
- Power is provided by an external unfiltered source. Any power source (AC or DC) providing more than 15 V (limited to 30 V) is ok. It is possible to use two 9V batteries.

2. On board peripherals and devices

The most important components of the EduPIC board are discussed in this section.

Programming unit

One of the most interesting features of this board is the possibility of on board programming of the PIC microcontroller. This feature is especially important for lab sessions where one single device must be programmed several times.

Figure 2 shows the circuit of the programming unit of the board. PIC programming lines (RB7 and RB6) are connected to the parallel connector through a simple buffer.

It is necessary to provide a disconnection switch to separate these lines from other peripherals; if other devices are loading these programming lines, the PIC cannot be programmed.

In this figure, the reset circuitry is also shown. It is a simple reset without power-up circuit, since the PIC already provides the power-up reset feature.

Analogue and digital sources

There are 5 digital and 4 analogue on board sources. A general purpose board must provide a mechanism to completely disconnect these sources to the PIC inputs, otherwise these sources would interfere with other external sources that we would like to include.

These digital and analogue sources has been connected to the less significant bits of the PIC A port, since the PIC analogue inputs are always located at this port.

There are 4 analogue sources: two of them are trimmers to provide any voltage between 0 and 5 volts. Another analogue source comes from a temperature sensor (LM35) which provides a linear relation between temperature and voltage at a rate of 10 mV/C starting at 0 C.

The last analogue source comes from a resistor sensible to light intensity. The circuit is as simple as two resistors in series (one is the LDR); the voltage measured in the voltage between the two resistors that are connected to 0 and 5 volts.

The digital sources have been implemented using simple switches to put high or low.

Three position switches have been employed to select analogue source, digital source or complete disconnection.

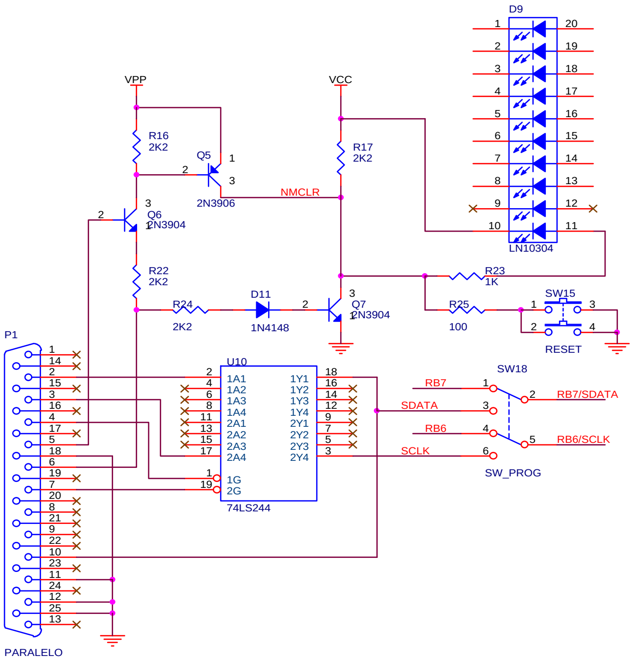

Fig. 2. Circuit of the on board programming unit

8-led bar and 7-segment display

The basic digital output is to put one single led for each bit of a port. In this case we have chosen a led-bar to simplify connection and reduce space.

The second basic digital output could be a 7-segment display connected to the 8 bits of an output port.

We connected the B port to the 7-segment display and the led-bar. Figure 3 shows the circuitry of this connection.

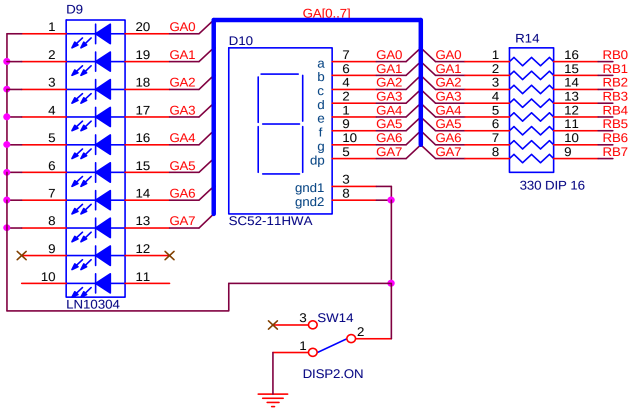

Fig. 3. Circuit of the led-bar and 7-segment display

Both devices (display and leds) share the same resistor array, but this is not very good since current through the diodes of each device can be very different, in fact, it is possible than one of the devices take most of the current. The solution is to choose well balanced diodes for each device, or to drive them separately. The led-bar takes most of the current and this is the reason why the 7-segment display is less visible. This can be solved employing a more brilliant 7-segment display.

Multi-digit display using 7-segment displays

In order to increase the complexity of output devices we considered the inclusion of a 4 digit numeric displays created from 7-segment displays. There are several possibilities for such a display, but we opted for a refreshing display since it allows us to introduce concepts as background tasks during normal program execution.

Figure 4 shows the circuitry for this 4 digit display. The four less significant bits of B port feeds the BCD to 7-segment decoder/driver (74LS48). The 7-segment output is common to all displays, thus they all should display the same digit. Each display is enabled, using a simple transistor, by one of the four most significant bits of the B port.

Displaying a complete number is as simple as to put the digit in the lower part of the B port at the same time that one of the displays is enabled while the others are disabled; repeating this procedure for each digit in fast alternation it is possible to see a complete 4 digit number.

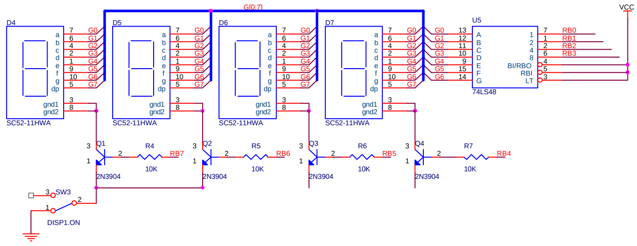

Fig. 4. Circuit of the 4 digit display

Matrix keyboard

It is important to include a keyboard in a general purpose experimental board. We opted for a matrix keyboard because it provides 2^n keys for n lines and it is very simple to use.

The 4x4 keyboard (16 keys) is connected to the B port. The four significant bits are connected to the matrix rows while the most significant bits are connected to the columns.

There is no need for keyboard pull-up resistors since the PIC already provides internal pull-ups for the B port. Should the keyboard be connected to other port, it would be necessary to include pull-up resistors in all lines of the keyboard.

LCD display

The last step toward the display complexity is the LCD. There are more complex display devices, but there are few that are so powerful, easy to use and standard.

We have connected a simple LCD display to the B port for the data bus and the three less significant bits of the A port for the LCD control lines.

Figure 5 shows the connection of the LCD to the PIC at the board.

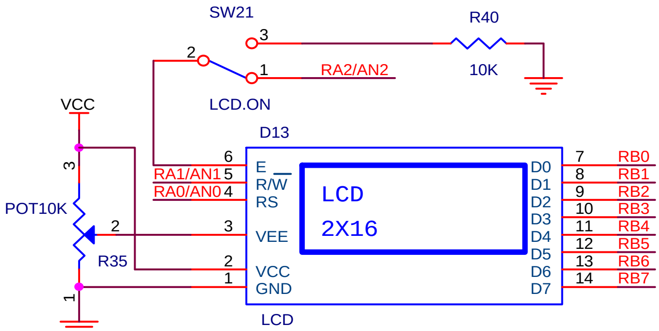

Fig. 5. LCD circuitry

RS232 Interface

It is necessary to provide a general purpose experimental board with some mechanism of communication and data transmission.

We decided to use the serial RS232 transmission for its simplicity and availability; almost any personal computer has at least one RS232 port.

Most modern PIC microcontrollers provide the UART for the serial communication, but they do not provide the signal voltage range required for the interface. It is then necessary to include a RS232 driver to accommodate the signal voltage as seen on Figure 6.

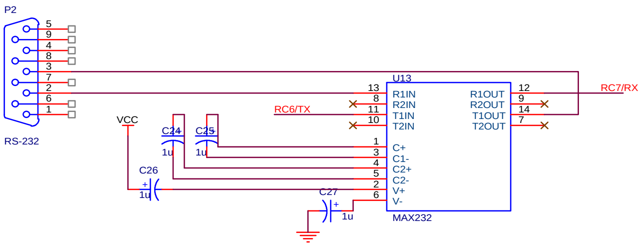

Fig. 6. RS232 interface

I2C Bus

The communication protocol I2C allows the information transmission among several devices from just two wires. The simplicity and easy to use are some of the reasons for its inclusion in an embedded system.

Most current microcontrollers (including PICs) have two pins for the I2C communication. There also some interesting simple peripherals which include the I2C bus as the only digital interface, simplifying inter-chip connections.

Two of these peripherals have been included in the I2C bus provided by most of PICs. One of these peripherals is an A/D and D/A converter. The inclusion of this peripheral complements the analogue features of the PIC that does not include any digital to analogue converter.

The other I2C peripheral is a simple serial EEPROM that expands the storage capabilities of the PIC, specially thinking that some of the PICs do not include any internal EEPROM for permanent storage.

USB interface

There are some PIC microcontrollers (still very few) that include a complete USB interface on chip. Not only provides the logical interface, but also the physical, making it not necessary to provide external complex driver for the

USB signals; only few resistors and capacitors are necessary for the USB interface.

For a general purpose experimental board it is necessary to include some configuration jumpers, otherwise, the USB function would interfere in other peripherals. For example, one of the USB signals is shared with one of the I2C signals, thus it is not possible to have an I2C bus working together with the USB interface. A couple of jumpers decide between USB or I2C function.

In the other hand, there are different configurations for the USB interface since it can act as a master or slave.

An USB type A connector is provided on the board for connecting the board to any USB master.

Other interfaces

All interfaces commented here have extra connectors to enhance function, enlarge connectivity and expand the board.

3. Software

There is no firmware in the board; this was one of the requirements for the lab board: it should be open to any program and configuration, not having any firmware that could interfere with user programs.

Nevertheless, there should be a mechanism to download our program into the PIC and then run the program. This is provided by the already mentioned ICSP provided by most PICs.

It is then necessary to have a communication program that would transmit the program to the PIC. This program has been developed and its function is to read the program in a standard format (HEX the Intel) and then transmit it to the PIC through the personal computer parallel port.

Program transmission is the main function, but it is not the only one. The software also permits reading the PIC, programming the PIC internal EEPROM, configuration word and identifier. It is also able to unprotect Flash PICs that are read and write protected.

All these features are combined with a friendly user interface based on forms and windows.

4. Laboratory sessions

The laboratory work is divided into 5 sessions with different goals. The main objective is to cover most of the microcontroller and peripheral features and their combination to solve real life problems. In the last session the student has the opportunity to show all acquired skills developing an embedded systems of its invention.

Session 1: Assembler programming

This first session is dedicated to the PIC assembler programming. They will use C in all other sessions, but it is important that they know what it is behind the C, especially since the cross compiler just makes a translation from C to ASM and then machine code.

There are also some PIC specific tricks that are interesting to know even if C is always employed. These features are the memory bank organization, the access to tabular data, etc.

This first lesson also shows the use of basic debugging tools as simulators, etc.

Session 2: Interrupts, timing and multi-digit display

The language C is introduced in this second session. Also the concept and use of interrupts and timers is explained. The four digit display is also introduced in this session.

For this session the students must implement a simple count-down timer with two controls for count, stop, reset and initialize.

They must use the timer (timer1) to generate interrupts to decrement the count, stop it, etc. At the same time, the main program is refreshing the display showing the four digit number.

Session 3: LCD and keyboard

In this session the students learn how to manage a general matrix keyboard and how to manage a standard LCD display.

They have to implement a temperature converter embedded application. Messages and results are shown in the LCD and data is entered through the keyboard.

Session 4: Analogue inputs and RS232 communication

Student will learn how to read the PIC analogue inputs and also the bi-directional data transmission.

They have to develop an application remotely controlled though the serial port. From a HyperTerminal on a personal computer the user can send basic commands to select the analogue input and receive the data to the terminal.

Session 5: Free project

The student must present a project of its invention. It must run on the board and must make use of as much as peripherals and features as possible.

5. Acknowledgements

We would like to thank the Computer Science Department at the University of Valencia for its funding of the EduPIC board development.

References

[1] Microchip Inc, “Many manuals and application notes”, *http://www.microchip.com*

[2] Fernando Pardo, “EduPic, Tarjeta de desarrollo para sistemas basados en PIC: Manual del usuario”, *http://tapec.uv.es/edupic*