FUZZY LOGIC IN EMBEDDED MICROCOMPUTERS AND CONTROL SYSTEMS. HARDWARE CONSIDERATIONS

ROTAR DAN

University of Bacau, Energetic Mecatronic and Computers Science Department

Abstract: The computing or controlling elements of embedded control applications are embedded inside the application. The embedded control systems with PIC microcontroller are discussed. The PIC microcontroller family is presented and the fuzzy logic main resources are analyzed. The fuzzy logic technique implies specific hardware configurations for the input variables and the main author solutions are presented.

Keywords: microcontroller, interface, fuzzy logic, linguistic variables, peripheral, memory, crisp numbers

1. INTRODUCTION

Unlike "processor" applications such as personal computers and workstations, the computing or controlling elements of embedded control applications are embedded inside the application. The consumer is only concerned with the very top-level user interface such as keypads, displays and high-level commands. Very rarely does an end-user know (or care to know) the embedded controller inside (unlike the conscientious PC users, who are intimately familiar not only with the processor type, but also its clock speed, DMA capabilities and so on).

A microcontroller (or MCU) is a computer-on-a-chip used to control electronic devices. It is a type of microprocessor emphasizing self-sufficiency and cost-effectiveness, in contrast to a general-purpose microprocessor (the kind used in a PC). A typical microcontroller contains all the memory and interfaces needed for a simple application, whereas a general-purpose microprocessor requires additional chips to provide these functions.

It is, however, most vital for designers of embedded control products to select the most suitable controller and companion devices. Embedded control products are found in all market segments: consumer, commercial, PC peripherals, telecommunications, automotive and industrial. Most embedded control products must meet special requirements: cost effectiveness, low-power, small-footprint and a high level of system integration. Typically, most embedded control systems are designed around an MCU, which integrates on-chip program memory, data memory (RAM) and various peripheral functions, such as timers and serial communication. In addition, these systems usually require complementary Serial EEPROM, analog/interface devices, display drivers, keypads or small displays.

Microchip provides useful and innovative solutions for embedded system designs. The PIC microcontrollers offers a wide range of solutions for embedded systems. The combination of high-performance PIC12CXXX, PIC16C5X, PIC16CXXX, PIC17CXXX and PIC18CXXX MCU families with Migratable Memory™ technology, along with non-volatile memory products, provide the basis for control systems. These ones can be used with fuzzy logic technique for many applications.

The fuzzy techniques avoid the complex calculus and activities and allow the control of the complex processes with a simple microcontroller. This means that, unlike a conventional control system that easily implements a single well behaved control of a system, the fuzzy logic design can have many solutions (or rules) which apply to a single problem and the combined solutions can be appropriately weighted to a controlling action.

This paper presents the particularities of the fuzzy logic implementation to the embedded systems with Microchip PIC microcontrollers.

2. THE CURRENT PIC MICROCONTROLLERS FAMILIES

PICmicro devices are grouped by the size of their Instruction Word. The three current PICmicro families are:

- PIC12CXXX 8-pin 12-bit/14-bit program word
- PIC16C5X 12-bit program word
- PIC16CXXX 14-bit program word
- PIC17CXXX 16-bit program word
- PIC18CXXX enhanced 16-bit program word

All families offer OTP, low-voltage and low-power options, with a variety of package options. Selected members are available in ROM, EEPROM or reprogrammable FLASH versions.

Each part of a device can be placed into one of three groups: core, peripherals, and special features.

The core pertains to the basic features that are required to make the device operate. These include: device Oscillator, reset logic, CPU (Central Processing Unit), ALU (Arithmetic Logical Unit), device memory map organization, interrupt operation, instruction set.

Peripherals are the features that add a differentiation from a microprocessor. These ease in interfacing to the external world (such as general purpose I/O, LCD drivers, A/D inputs, and PWM outputs), and internal tasks such as keeping different time bases (such as timers).

The special features are: device Configuration bits, on-chip power-on Reset (POR), brown-out Reset (BOR) logic, Watchdog Timer, low power mode (Sleep), internal RC device oscillator, in-Circuit Serial Programming™ (ICSP™).

PIC12CXXX: 8-Pin, Family
The PIC12CXXX family packs Microchip's powerful RISC-based PICmicro architecture into 8-pin DIP and SOIC packages. These PIC12CXXX products are available with either a 12-bit or 14-bit wide instruction set, a low operating voltage of 2.5V, small package footprints, interrupt handling, a deeper hardware stack, multiple channels and EEPROM data memory. All of these features provide an intelligence level not previously available in applications because of cost or size considerations.

PIC16C5X: 12-Bit Architecture Family
The PIC16C5X is the well-established base-line family that offers the most cost-effective solution. These PIC16C5X products have a 12-bit wide instruction set and are currently offered in 14-, 18-, 20- and 28-pin packages. In the SOIC and SSOP packaging options, these devices are among the smallest footprint MCU's in the industry. Low-voltage operation, down to 2.0V for OTP MCU's, makes this family ideal for battery-operated applications. Additionally, the PIC16HV5XX can operate up to 15 volts for use directly with a battery.

PIC16CXXX: 14-Bit Architecture Family
With the introduction of new PIC16CXXX family members, Microchip now provides the industry's highest performance Analog-to-Digital Converter capability at 12-bits for an MCU. The PIC16CXXX family offers a wide-range of options, from 18- to 68-pin packages as well as low to high levels of peripheral integration. This family has a 14-bit wide instruction set, interrupt handling capability and a deep, 8-level hardware stack. The

PIC16CXXX family provides the performance and versatility to meet the more demanding requirements of today's cost-sensitive marketplace for mid-range applications.

PIC17CXXX: 16-Bit Architecture Family
The PIC17CXXX family offers the world's fastest execution performance of any MCU family in the industry. The PIC17CXXX family extends the PICmicro MCU's high-performance RISC architecture with a 16-bit instruction word; enhanced instruction set and powerful vectored interrupt-handling capabilities. A powerful array of precise on-chip peripheral features provides the performance for the most demanding applications.

PIC18CXXX: 16-Bit Enhanced Architecture Family
The PIC18CXXX is a family of high performance, CMOS, fully static, 16-bit MCU's with integrated analog-to-digital (A/D) converter. All PIC18CXXX MCU's incorporate an advanced RISC architecture. The PIC18CXXX has enhanced core features, 32 level-deep stacks, and multiple internal and external interrupts sources. The separate instruction and data busses of the Harvard architecture allow a 16-bit wide instruction word with the separate 8-bit wide data. The two-stage instruction pipeline allows all instructions to execute in a single cycle, except for program branches, which require two cycles. A total of 77 instructions (reduced instruction set) are available. Additionally, a large register set gives some of the architectural innovations used to achieve a very high performance of 10 MIPS for an MCU. The PIC18CXXX family has special features to reduce external components, thus reducing cost, enhancing system reliability and reducing power consumption. These include programmable Low Voltage Detect (LVD) and programmable Brown-Out Detect (BOD).

3. FLEXIBLE PROGRAMMING OPTIONS

For fuzzy logic systems is very important the memory programming mechanism with the object of the system fitness to the condition changes. For the PIC microcontroller the following innovative programming options are offered.

FLASH (electrically reprogrammable)
PICmicro FLASH MCU's allow erase and reprogramming of the MCU program memory. Reprogrammability offers a highly flexible solution to today's ever-changing market demands – and can substantially reduce time to market. Users can program their systems very late in the manufacturing process or update systems in the field. This allows easy code revisions, system parameterization or customer-specific options with no scrap page. Reprogrammability also reduces the design verification cycle.
One-Time Programmable (OTP) PICmicro OTP MCU's are manufactured in high volumes without customer specific software and can be shipped immediately for custom programming. This is useful for customers who need rapid time to market and flexibility for frequent software updates.

In-Circuit Serial Programming™ (ICSP™)
Microchip's PICmicro FLASH and OTP MCU's feature ICSP capability. ICSP allows the MCU to be programmed after being placed in a circuit board, offering tremendous flexibility, reduced development time, increased manufacturing efficiency and improved time to market. This popular technology also enables reduced cost of field upgrades, system calibration during manufacturing, the addition of unique identification codes to the system and system calibration. Requiring only two I/O pins for most devices, Microchip offers the most non-intrusive programming methodology in the industry.

Self-Programming
Microchip's PIC16F87X family features self-programming capability. Self-programming enables remote upgrades to the FLASH program memory and the end equipment through a variety of medium ranging from Internet and Modem to RF and Infrared. To setup for self-programming, the designer programs a simple boot loader algorithm in a code protected area of the FLASH program memory. Through the selected medium, a secure command allows entry into the PIC16F87X MCU through the USART, I2C or SPI serial communication ports. The boot loader is then enabled to reprogram the PIC16F87X FLASH program memory with data received over the desired medium. And, of course, self-programming is accomplished without the need for external components and without limitations on the PIC16F87X's operating speed or voltage.

Quick-Turn Programming (QTP)
Microchip offers a QTP programming service for factory production orders. This service is ideal for customers who choose not to program a medium to high unit volume in their own factories, and whose production code patterns have stabilized.

Serialized Quick-Turn Programming (SQTPSM)
SQTP is a unique, flexible programming option that allows Microchip to program serialized, random or pseudo-random numbers into each device. Serial programming allows each device to have a unique number, which can serve as an entry-code, password or ID number.

Masked ROM
Microchip offers Masked ROM versions of many of its most popular PICmicro MCU's, giving customers the lowest cost option for high volume products with stable firmware.

4. FUZZY LOGIC IN EMBEDDED MICROCOMPUTERS

Fuzzy logic operators provide a formal method of manipulating linguistic variables. It is a reasonable comment to describe fuzzy logic as just another programming paradigm. Fuzzy logic critics are correct in stating that they can do with conventional code everything that fuzzy logic can do. For that matter, so can machine code, but I am not going to argue the point.

Central to fuzzy logic manipulations are linguistic variables. Linguistic variables are non-precise variables that often convey a surprising amount of information. Linguistic variables represent crisp information in a form and precision appropriate for the problem.

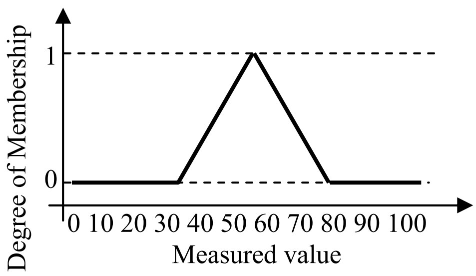

Figure 1. Graphic representation for a linguistic variable.

Linguistic variables in a computer require a formal way of describing a linguistic variable in crisp terms the computer can deal with.

Often is used a graph (Figure 1.) which represent the relationship between measured value and the linguistic term. The horizontal axis in the graph shows the measured or crisp value of temperature. The vertical axis describes the degree to which a linguistic variable fits with the crisp measured data.

The PIC microcontrollers offers a lot of possibilities for introducing the crisp values of the linguistic variables. First, the simplest microcontrollers have a number of digital inputs that can be used in many manners.

The input codes result of an external analog to digital converter is applied to the digital inputs. The microcontroller generates the control signal for conversions of many input sources (Figure 2).

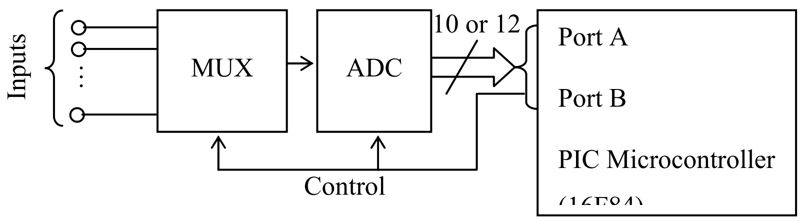

Figure 2. External ADC

Instead of analog to digital converter we can use a tension to frequency converter for analog inputs. This solution is less expensive and suitable for simplest applications. In this method the analog input signal is converted in a frequency modulated signal applied directly to the numerical inputs of the microcontroller. Figure 3 shows this solution.

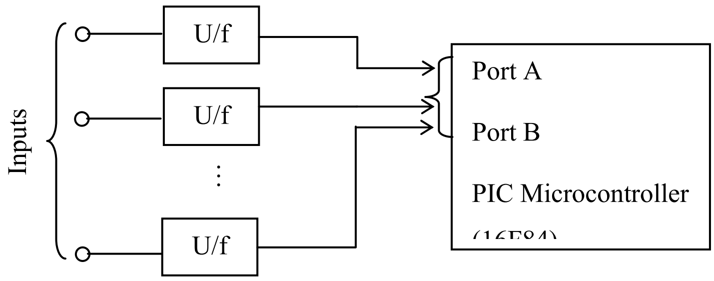

Figure 3. Frequency modulated inputs.

The port pin is sampled in an instruction cycle (four clock input period). At a reasonable routine for the data manipulation the microcontroller can execute one read of the port pin approximately in 15-20 instruction cycles. This action establishes the maximum rate for the external analog to digital converter.

In the second situation the internal timer is used for the frequency measurement. It is necessary a measurement for the each input channel. As a rule the inputs frequency is measured by numbers of the input transitions in a certain time interval. The experimental determination show a limit of the maximum input signal frequency at microcontroller clock input / 5000.

For the enhanced architecture family the internal analog to digital converter is used (Figure 4). The conversion of an analog input signal results in a corresponding 10-bit digital number. The analog to digital module has high and low voltage reference input, which is software selectable.

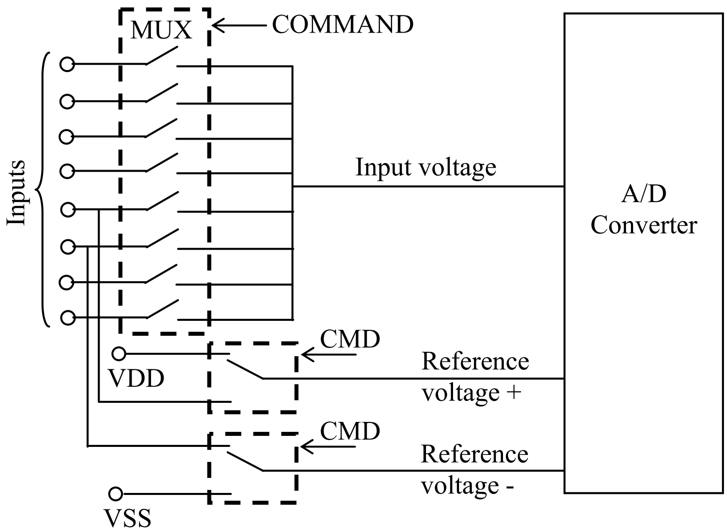

Figure 4. Internal analog to digital converter

The source impedance and the internal sampling switch impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (RSS) impedance varies over the device voltage (VDD). To calculate the minimum acquisition time, Equation 1 may be used.

$$t_{acq} = t_{amp} + t_c + t_{coff} \quad (1)$$

where: t_{amp} is the amplifier settling time, t_c is the hold capacitor charging time and t_{coff} is the temperature coefficient.

This equation assumes that 1/2 LSb error is used (1024 steps for the A/D). The 1/2 LSb error is the maximum error allowed for the A/D to meet its specified resolution. The maximum recommended impedance for analog sources is 10 kΩ. As the impedance is decreased, the acquisition time may be decreased.

The analog to digital conversion time per bit is defined as T_{AD}. The analog to digital conversion requires a minimum $12T_{AD}$ per 10-bit conversion. The source of the analog to digital conversion clock is software selected. For correct analog to digital conversions, the conversion clock (T_{AD}) must be selected to ensure a minimum T_{AD} time of 1.6 μs.

5. CONCLUSIONS

The input technique of the variables for the fuzzy controller is not very different from the classical embedded systems.

The hardware characteristics of the PIC microcontrollers allow the achievement in good conditions of the embedded fuzzy controllers.

The experiments of the author demonstrate the possibilities of system simplification for processes control using the fuzzy technique. Preparations of the inputs variables are made in the hardware structure or in software structure of the fuzzy controller.

REFERENCES

[1] Gupta, Madan M. and Sanchez, Elie, ed., *Fuzzy information and decision processes*, Amsterdam: North-Holland, ISBN 0-44-486491-1. (Companion vol.: Approximate reasoning in decision analysis), 1982.
[2] Yager, Ronald R. and Zadeh, Lotfi A., ed., *An Introduction to Fuzzy Logic Applications in Intelligent Systems*, Kluwer Academic Publishers, ISBN 0-7923-9191-8, 1992.
[3] Sibigtroth, James M., *Creating Fuzzy Micros*, December, Emedded Systems Programming, pp. 20-31, 1991.
[4] C. Von Altrock, B. Krause, and H.J. Zimmermann, *Advanced Fuzzy Logic Control Technologies in Automotive Applications*, Proc. IEEE Int'l Conf. Fuzzy Systems, IEEE, Piscataway, N.J., pp. 835-842, 1992.
[5] M. Sasaki, F. Ueno, and T. Inoue, *A 7.5 MFLIPS Fuzzy Microprocessor Using SIMD and Logic-in-Memory Structure*, Proc. 2nd IEEE Int'l Conf. Fuzzy Logic, IEEE, pp. 527-534, 1993.
[6] E. Avogadro et al., *An Optimized RISC Instruction Set for Fuzzy Applications*, Proc. Third IEEE Int’l Conf. Fuzzy Systems, IEEE Service Center, Piscataway, N.J., pp. 133-137, 1994.
[7]*** Microchip, PICmicro™ Mid-Range Reference Manual (DS33023).