



## **Mechatronic module for monitoring the hydraulic parameters of industrial equipment**

Marian BLEJAN<sup>1)</sup>, Petrin DRUMEA<sup>1)</sup>, Ioana ILIE<sup>1)</sup>, Dumitru VLAD<sup>2)</sup>  
<sup>1)</sup> INOE 2000 - IHP, Bucharest, Romania, <sup>2)</sup> INCDMF, Bucharest, Romania,

**Abstract:** The large scale use of hydropneumatic equipments into fabrication process, combined with the production pursuit informational systems impose complex measurements of parameters must be transmitted to the informational systems. The integration of some mechatronic ensembles into the hydropneumatic equipments, ensembles which allow the measurement and the network transmission of the measurement results through the informational systems, makes an easy way for the production centers to informational improve their fabrication process whit all the benefits implied. The global evolution of embedded systems, combined whit new tips of sensors and measuring methods of hydro pneumatic devices specific physic parameters, allows the tackle of those measuring mechatronic ensembles in partnership whit the European research centers that have the same goals in those fields.

### **1. Introduction**

The electronic system for monitoring hydraulic oil allows data acquisition from the temperature, flow and pressure transducers; the data from the transducers are scaled and transmitted within the serial communication line. The electronic module calculates the hydraulic power as a product between pressure and flow and transfers it through the serial communication line. The electronic module is equipped with a console which allows visualization and modification of the functional parameters and the visualization of the measured parameters. The module is also equipped with two general purpose analog in entrances, which may be configured by the user. The feeding voltage of the electronic module is between the range 9..36V rectified and filtered DC, which covers the range of available voltages in the usual automation equipment.

### **2. GENERAL MODULE DESCRIPTION**

The electronic module is build based on a general purpose PIC16F876A microcontroller, made by Microchip. Few of the main characteristics of this type of microcontroller are:

- RISC architecture with 35 instructions with one word length (14 bits);
- the instructions are executed within a machine cycle except the jump instructions which need two machine cycle as execution time.
- the maximum operation speed is 200 ns per instruction cycle;
- the program memory is a flash memory with the capacity of 8096 instructions;
- the data memory (RAM) has 386 bytes;
- the parameters memory (EEPROM data memory) consists in 256 bytes;
- pin out compatible with all 28 pins microcontrollers made by Microchip;

The PIC16F876A microcontroller has a large variety of integrated peripherals such as:

- a 8 bit timer and two 16 bit timers;

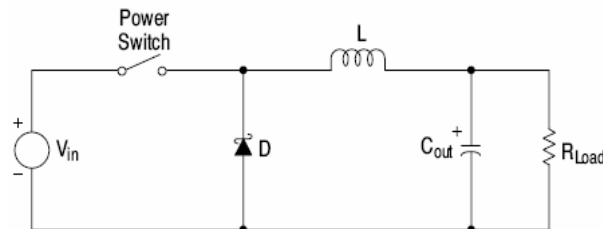
- two Capture/Compare/PWM modules with the speed of 12,5 ns in Capture mode, 200 ns in Compare mode and the maximum PWM resolution of 10 bits;
- synchronous serial communication port with SPI support (Master mode) and I2C;
- USART communication port with address detection;
- brown-out detection circuit for Brown-out Reset (BOR);
- 5 channels 10 bits analog to digital converter;
- two analog comparators;
- programmable voltage reference.

### 3. FUNCTIONAL DESCRIPTION

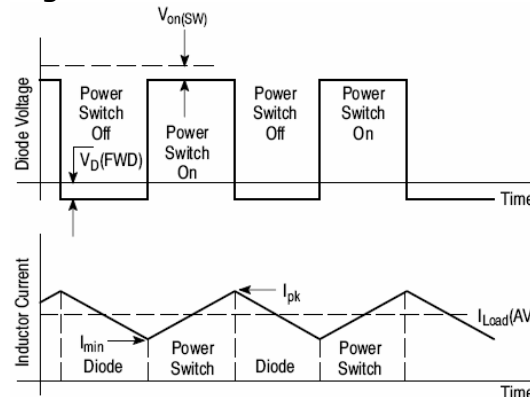
The electronic module is made from the stabilized power supply, analog input circuits, analog output circuits, the 7 segments 3 digits led display, 4 state leds, 4 buttons keyboard (UP, DOWN, CONFIRM, CANCEL), serial communication and in-circuit programming (ICSP).

#### 3.1. The regulated power supply

The regulated power supply receives on the input terminal the module's supply voltage, rippled DC voltage within the range of 9...36 V, providing on the output terminal a regulated voltage of 5 V with current limitation at 0,5 A. The power supply is based on the LM2574-05 integrated circuit (LM2574-05 is a "step-down"/buck converter commutation regulator type).



**Fig. 1.** "Buck Converter" main schematic



**Fig. 2.** The "buck converter" theoretical wave forms



To understand the “buck converter” based commutation voltage regulator operation, figures 1 and 2 are shown. Figure 1 shows the main schematic of the convertor from which, its structure can be observed: a power switch, a Schottky diode, the energy storage coil and the filter capacitor. Figure 2 allows the understanding of the “buck” converter operation; therefore, in the beginning the taken energy is stored from the power supply  $V_{IN}$  into the coil L by closing the *Power Switch*. Next, the switch opens, the diode D closes, and the energy stored into the coil L during the previous state is transferred through the *Load R*. Theoretically, the ratio between the output and the input voltage for the “buck” converter equals the duty cycle of the power switch’s command signal.

The switching power supply solution has been chosen because it offers advantages regarding high efficiency, wide field of output voltage, small printed boards, reliability and others.

### **3.2. The analog circuits**

The analog input circuits allow the connection of four transducers with 4..20 mA output current, powered through 2 or 3 wires. Each input has a three terminal connector: supply +, supply - and the input current signal which enters the electronic module. A current/voltage converter is placed onto the input signal, that changes the 4..20mA current range into the 1..5 V voltage range. This voltage is applied by one analog input of the analog to digital converter with successive approximations integrated into the microcontroller. The input voltage signal is available into the microcontroller as a 10 bits integer numeric value. The analog to digital converter is configured to function with a 5V reference voltage; so, the range  $0..(5-5/1024=4,995)$  V will be converted into numeric values within 0..1023 range. The current signal supplied by the transducer will be converted into numeric values corresponding to 1..5 V range respectively  $1024/5 = 205 \dots 1023$ ; the obtained resolution on transducer’s working range is  $1023-205 = 818$  divisions. To improve the resolution of the signal read from the transducer is used the “circular buffer” mediation technique which contains the last 20 pieces of the signal; in this mode, a resolution of 10.000 divisions onto transducer’s measurement range of analog inputs is obtained.

The analog output circuits allow users to interface the electronic module with various modules which accept at the input voltage or current signals. Depending on the application, the user can configure the two analog outputs for numeric values calculated for the transducers connected at the analog inputs or for indirect measured values. The electronic module’s analog outputs use the two Capture/Compare/PWM modules integrated into the microcontroller; these modules are programmed in the PWM mode allowing a width modulated signal with 0 V and 5 V levels to be obtained, the duty cycle’s resolution of 10 bits (1024 values) and a frequency of 19.53 kHz for the output signal. The logic signal, width modulated, present at the output of the PWM module is applied to a first order low – pass filter endowed with an output buffer, which is built with operational amplifier in repeater connection. The output of the low-pass filter is the analog output signal of the electronic module. In other words, the PWM signal, with constant period and duty cycle proportional with the contained information, is converted into analog signal using the first order low-pass filter.

Every analog output is endowed with a two terminal connector at which ground and user desired range analog output signal are connected.



### **3.2. The user interfaces**

*The numeric display* allows a numeric value in range of -99.999 with decimal point in every position to be displayed. Using the numeric display, the user can observe the values calculated by the electronic module and its configuration parameters values. The numeric display consists of three seven segments led digits with decimal point. To reduce the number of microcontroller's pins used for driving the display (numeric display), together with the state indicator leds, are multiplexed at digit level with a  $\frac{1}{4}$  multiplexing factor. So, to drive three digits and one digit made from the four state leds 12 microcontroller pins = eight pins for segments + four pins for digits are needed. There is to mention that the pins for digits command are used also to activate the read of the four keys of the keyboard; at a certain point a single digit is active and a single key is active to be read. Because the provided/absorbed current by a microcontroller pin is limited at 25mA, a driver was needed, consist a transistor in "common emitter" connection, for digits drive.

*The state indicator leds* monitor the serial communication state the operation within normal parameters of electronic module. For monitoring the MODBUS communication over serial line, MODBUS standard ([www.modbus.org](http://www.modbus.org)) imposes three leds to be present: Send Receive and Error. The four state indicator leds, from which three are used to monitor the communication's state and one to monitor the electronic module's state, are common cathode connected, being considered the fourth display digit. They are multiplexed the three numeric digits (see previous paragraph).

*The keyboard*, is contained by the module, made by four push-buttons, allows user to navigate through the visualization and configuration menus of the electronic module and numeric values editing. The meaning of the buttons are: UP (next menu, value increment), DOWN (previous menu, value decrement), OK (submenu activation, circular increment of edited position, action confirmation) and CANCEL (submenu deactivation, cancel editing mode, cancel action). To read the keyboard, a single microcontroller pin is used, the buttons being successively activated according to the digit display multiplexing.

### **3.3. The communication interface**

*The serial communication* allows, using some corresponding level adaptors, the communication with the electronic module using serial communication standards RS232, RS485 and RS422. The communication connector endows the internal 5V DC regulated power supply, used to power the level adaptor needed for the communication standard, RX and TX TTL level signals, the transmission line (module output) and the receiving line (module input); for the multipoint communication lines an adaptor's working state between receiving and transmission regime selection signal is needed. The working regime selection signal is a TTL level signal called DIR (direction).

The electronic module is endowed with a header type connector, used for programming the microcontroller, ICSP header (*In Circuit Serial Programming*).

The microcontroller operates with a clock generated by a 20 MHz quartz oscillator which corresponds to a execution period of 200 ns for one instruction.

## **4. PROGRAM DESCRIPTION**

The control program, written in the flash memory of the microcontroller, has been developed using the programming language ANSI C. The program consists in a initializing



sequence of the integrated peripherals in the microcontroller, the main loop which consists in a state machine implementing the functionality of the electronic module and the interrupt treating routine, which realize: analog acquisition, the real time clock, the display multiplexing, the keyboard reading and handling of the emission/reception buffers for serial communication.

The PIC16F876A microcontroller, which is contained in the electronic module, manage the interrupts generated by various peripherals when certain events occur, by saving into the hardware stack of program counter and deviating the instruction sequence execution at a fixed program memory location: 0x004 address, where the programmer inserts the interrupt handling code and as last instruction, *reti* (the return from interrupt instruction); the *reti* instruction continues the program execution from its interrupt point by recalling from stack the program counter. There is to mention that at the beginning of interrupt handling routine, the possibility of another interrupt occurring the interrupt routine execution, is deactivated; the possible interrupt will be memorized and handled later. A typical interrupt routine contains the main program context saving, the generated interrupt source identification, the determination of the event that generated the interrupt, handle of the respective interrupt, recalling the context of the existing situation at interrupt moment and return from interrupt into normal flux of execution.

The implemented interrupt routine handles the interrupts generated by the following peripheral equipments integrated into the microcontroller:

- the 16 bits timer TMR1, is used for 1 ms clock generation. In the initializing sequence, *TMR1* is configured to increment at a 200 ns delay (instruction clock) and the interrupt generation when an overflow event occurs (the *TMR1* content passes from the 0xFFFF value to the 0x0000 value); in the interrupt routine the content of the timer is set to *t1ms* value which allows the overflow event to occur with 1 ms delay  $(0xFFFF - t1ms) \times 200 \text{ ns} = 1 \text{ ms}$ ;
- the analog to digital converter is configured to generate a interrupt when the conversion's result is available; then, after taking over the conversion's result, the next readable analog input is chosen.
- *The USART module* is configured to generate interrupts when a character has been transmitted / received; the transmission of a character activates the pass to the next character to be transmitted from the transmission buffer, when the receiving of a character means the receiving buffer actualization with the received character. The ending of the transmission is signaled to the interrupt routine by the presence of a 0 ASCII code in the transmission buffer; the transmission ending is signaled to the main program by setting the TXIE bit (transmission interrupt enable) to 0 value. When receiving, the appearance of the 0xa or 0xd presumes, according to the MODBUS ASCII specifications, the end of the receiving; the interrupt routine detects the receiving of these characters and signals the event to the main program by inserting in the receiving buffer a 0 ASCII character. There is to mention that there are two pointers, one at the transmission buffer and one at the receiving buffer, that marks the current character, received or to be transmitted.

*The 1 ms interrupt handling routine* has the next duties:

- The update of the real time clocks used for processes in time synchronization.
- The display multiplexing by establishing the digit and the segments which will be activated (powered);

- Keyboard update;
- The analog to digital conversion start.
- The 1 ms interrupt initialization by loading into TMR1 the *t1ms* value and resetting the *T1IF* interrupt flag.

The analog to digital converter interrupt handling routine implements the next duties:

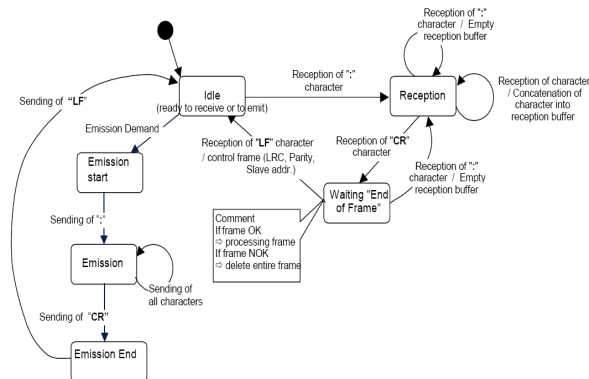
- Fetching the conversion result for the current analog input.
- The current analog input analog signal numeric filtering.
- Establishing the analog input that will be handled at the next interrupt as current analog input.
- The analog to digital converter interrupt initialization by resetting the *ADIF* interrupt flag.

The main execution loop contains a state machine which implements the electronic module functionality. The state machine is implemented using the specific notations UML (Unified Modeling Language). This way are treated the parameterization and monitoring menus of the electronic module, the signal numerical processing for obtaining the monitored values, and is implemented the MODBUS ASCII protocol.

The MODBUS function implemented with this module are:

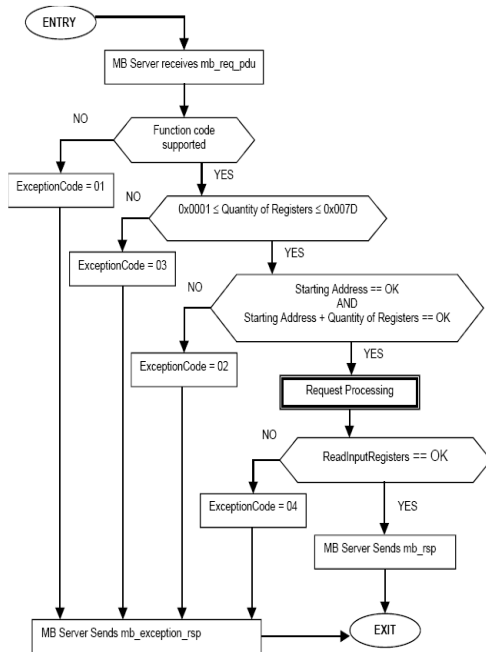
- 04 function (0x4), Read Input Registers
- 16 function (0x10), Write Multiple Registers
- 23 function (0x17), Read/Write Multiple Registers

The state machine that implements the MODBUS ASCII protocol is depicted in figure 3.

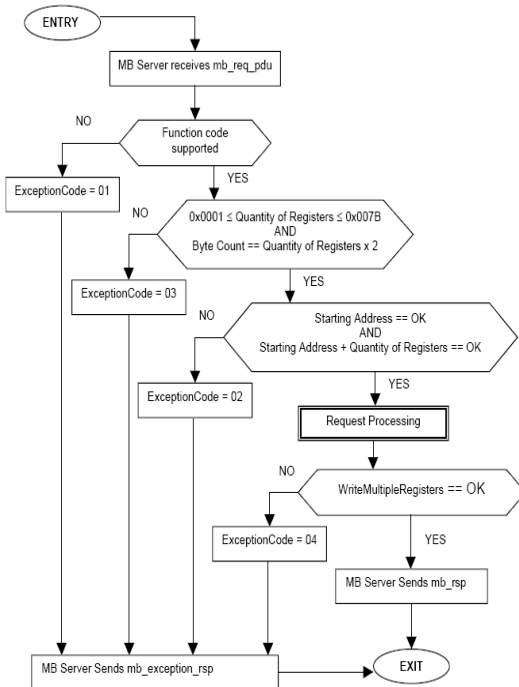


**Fig.3.** State diagram for the MODBUS ASCII protocol

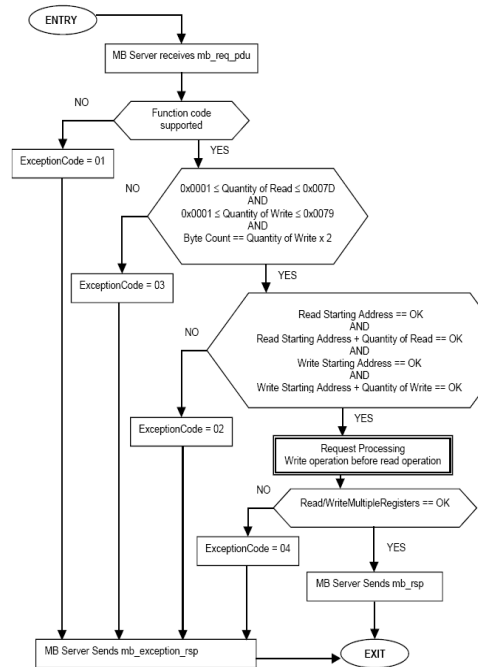
The state diagrams for the implemented MODBUS functions are depicted in figures 4, 5 and 6.



**Fig.4.** The state diagram for 04 function, Read Input Registers



**Fig.5.** The state diagram for 16 function, Write Multiple Registers



**Fig.6.** The state diagram for 23 function, Read/Write Multiple Registers

## 5. RESEARCH RESULTS

The presented electronic module is built on the mechatronic module which has been developed in INOE 2000 – IHP as an experimental module.



**Fig.7.** Mechatronic module for monitoring hydraulic parameters

## REFERENCES

- [1] <http://www.omg.org/technology/documents/formal/uml.htm>
- [2] <http://www.modbus.org>
- [3] <http://www.microchip.com>