Development of Data Acquisition Module Based on PIC18F4550 Microcontroller

Marko Anđelić[1], Viša Tasić[2], Marijana Pavlov-Kagadejev[3], Darko Brodić[4], Ivana Stojković[5]

Abstract – **This paper describes the development of low-cost data acquisition module which should be implemented in the existing process control systems. The main goal of such implementation is to improve the A/D conversion resolution of the realized and applied control systems. The new data acquisition module is based on PIC18F4550 microcontroller. Some specific hardware and software details of the realized module are specially emphasised.**

Keywords – **data acquisition module, A/D conversion, microcontroller**

I. Introduction

Department of Industrial Informatics, at the Mining and Metallurgy Institute Bor, has been designing real-time systems for monitoring and control of industrial processes since 1990. Three generations of Microprocessor Measuring Station (MMS) have been developed. MMS was used as a programmable logic controller (PLC) in the control systems mainly applied in the metallurgy. It is based on Motorola MC68HC11 microcontroller with integrated 8-bit A/D converter, 64 analog inputs, 64+64 digital state signals (input +output), RS232 communication port, 48 kB RAM and16 kB EPROM [1].

In order to improve the characteristics of A/D conversion, a new data acquisition module is developed. It is based on PIC18F4550 microcontroller with 10 bit A/D converter (13 analog input channels), integrated USB module, 32 kB of flash (program) memory, 2 kB RAM and 256 bytes of EEPROM memory [2].

Block diagram of PIC18xx microcontroller family architecture is shown in Fig. 1. Layout of the realized data acquisition module is shown in Fig. 2.

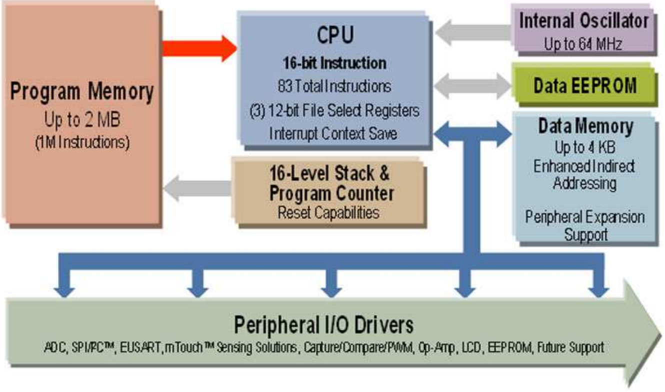


Fig. 1. PIC18xx Architecture Block Diagram [3]

During the realization of this project, some firmware and hardware tools for the development of PIC18F4550 microcontroller based systems were used:

- PICkit 2 In-Circuit Debugger/Programmer [4]
- PICBASIC PRO™ Compiler 3.0 [5]
- MPASM Microchip assembler [6]

For testing purposes, a special application for the PC workstation was developed by using Borland Delphi and mcHID.dll library [7].

In the paper, a brief overview of PIC18F4550 microcontroller A/D conversion process is presented. After that, an example of A/D conversion programming is explained in details.

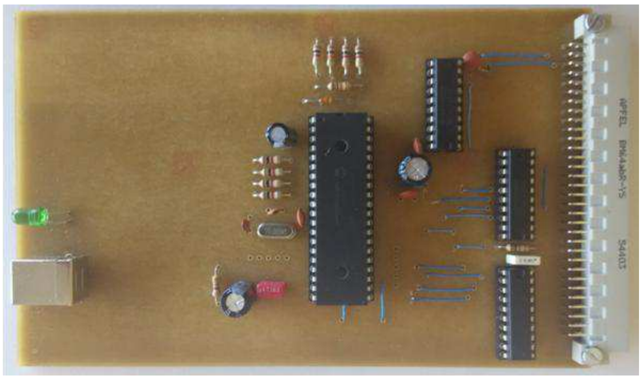


Fig. 2. Layout of the realized data acquisition module based on PIC18F4550 microcontroller

[1] Marko Anđelić is with the Mining and Metallurgy Institute Bor, Department of Science, Zeleni bulevar 35, 19210 Bor, Serbia, e-mail: marko.andjelic@hotmail.com,

[2] Viša Tasić is with the Mining and Metallurgy Institute Bor, Department of Science, Zeleni bulevar 35, 19210 Bor, Serbia, e-mail: visa.tasic@irmbor.co.rs,

[3] Marijana Pavlov-Kagadejev is with the Mining and Metallurgy Institute Bor, Department of Industrial Informatics, Zeleni bulevar 35, 19210 Bor, Serbia, e-mail: marijana.pavlov@irmbor.co.rs,

[4] Darko Brodić is with the University of Belgrade, Technical Faculty in Bor, Vojske Jugoslavije 12, 19210 Bor, Serbia, e-mail: dbrodic@tf.bor.ac.rs,

[5] Ivana Stojković is with the University of Niš, Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Niš, Serbia, e-mail: ivana.stojkovic@elfak.ni.ac.rs

II. A/D CONVERSION PROCESS IN REALIZED MODULE

The A/D converter module integrated in PIC18F4550 has 13 inputs and allows conversion of an analog input signal to corresponding 10-bit digital number as shown in Fig. 3.

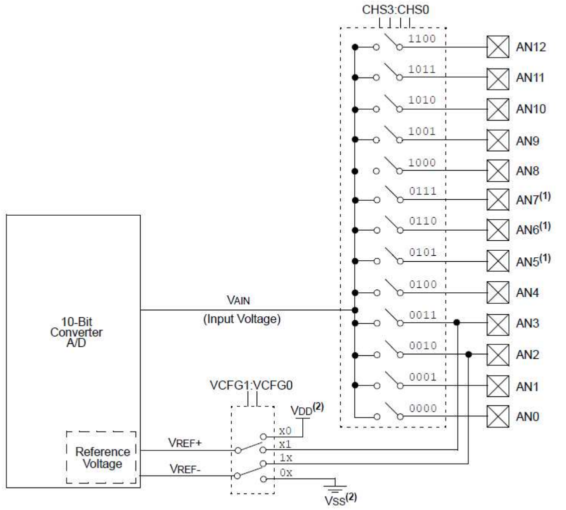


Fig. 3. A/D Block diagram [2]

The A/D conversion (ADC) module has five registers (2 data registers and 3 control registers) [2]:

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register 0 (ADCON0)
- A/D Control Register 1 (ADCON1)
- A/D Control Register 2 (ADCON2)

ADRESH and ADRESL registers serve as data registers for storing the A/D conversion results. ADCON0 control register configures ADC operations (start of A/D conversion and input channel selecting). ADCON1 register configures voltage reference and whether the input pins are analog or digital. ADCON2 register configures A/D acquisition time, ADC clock and A/D result format (justification).

In the realized data acquisition module, reference voltage for A/D conversion is set to device's positive and negative supply voltage (V_{DD} = 5 V and V_{SS} = 0 V). The output of the sample and hold is the input into the converter, which generates the result via successive approximation. When the A/D conversion is complete, the result is loaded into the ADRESH:ADRESL register pair, the GO/DONE bit (ADCON0 register) is cleared and A/D Interrupt Flag bit, ADIF, is set.

After the configuration of A/D module, and elapsing of the acquisition time, the A/D conversion of the selected channel can be started. The next steps should be followed to perform an A/D conversion:

- Configure the A/D module,
- Configure A/D interrupt,
- Wait the required acquisition time,
- Start conversion (ADCON0 register),
- Wait for A/D conversion to complete, by waiting for the A/D interrupt,
- Read A/D result registers (ADRESH:ADRESL),

When the analog input channel is selected, it must be sampled for at least the minimum acquisition time before starting a conversion. Calculation of the minimum required acquisition time T_{ACQ} is done with following equation:

$$T_{ACQ} = T_{AMP} + T_C + T_{COFF} \qquad (1)$$

Where T_{AMP} is amplifier settling time, T_C is holding capacitor charging time and T_{COFF} is temperature coefficient. Refering to datasheet specification of the microcontroller, calculation (1) is based on folowing system preferences: C_{HOLD} = 25 pF, Rs = 2.5 kΩ, Conversion Error ≤ 1/2 LSb, V_{DD} = 5V → R_{SS} = 2 kΩ, Temperature = 85˚C (system max.), and is given by:

$$T_{ACQ} = 0.2\ \mu s + 1.05\ \mu s + 1.2\ \mu s = 2.45\ \mu s \qquad (2)$$

The A/D conversion time per bit is defined as T_{AD} and per 10-bit conversion 11 T_{AD} is required. The acquisition time parameter is the amount of T_{AD} cycles to delay before the actual A/D conversion starts. The source of the A/D conversion clock is software selectable by bits ADCS2:ADCS0 in the ADCON2, in terms of the microcontroller clock (T_{OSC}). T_{AD} have to be selected as short as possible, but greater than T_{ADmin} (0.8 µs for PIC18F4550 [2]).

Since developed module has main crystal of 20 MHz, T_{OSC} is 0.05 µs. This means that lowest selectable value of T_{AD} is 16 T_{OSC} = 0.8µs, and the ADCS2:ADCS0 bits should be set to 101 B (binary). It is possible to either keep track of T_{ACQ} delay in the program code or to simply set the A/D acquisition time select bits to an even multiple of T_{AD}.

Based on the previous calculations, the minimum A/D processing time per channel in our case is given by:

$$T_{ACQ} + 11\ T_{AD} = 11.25\ \mu s \qquad (3)$$

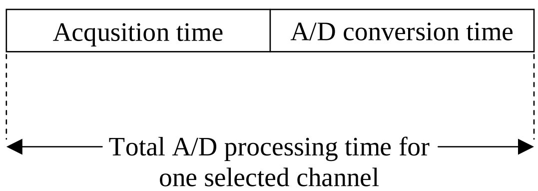


Fig. 4. A/D conversion processing time

III. The A/D Conversion Module Programming Example

PIC assembler program is written using the PICBASIC PRO™ Compiler 3.0 and MPASM universal assembler.

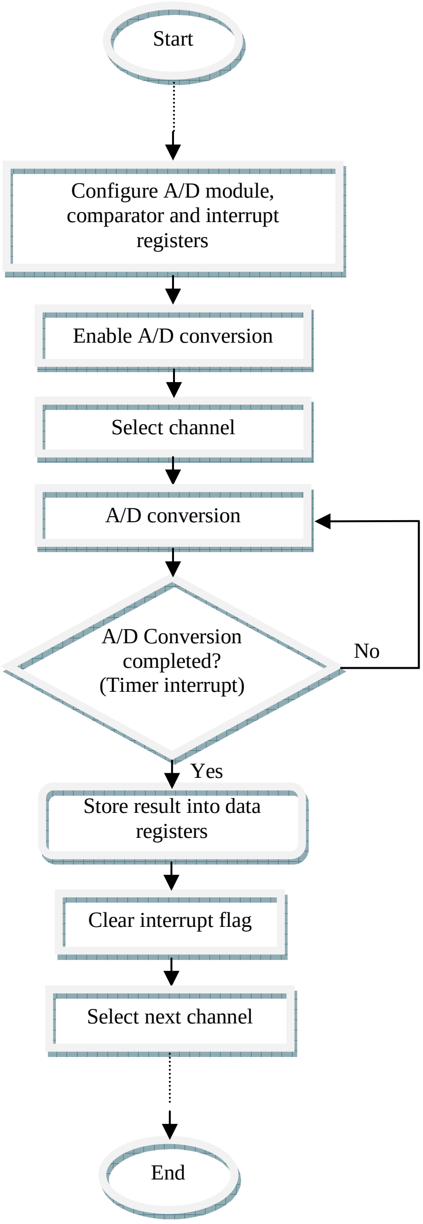
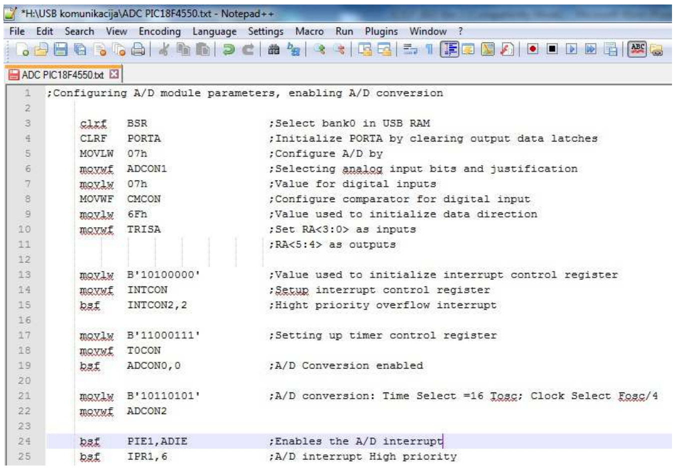


Fig. 5. Data acquisition flowchart (one channel)

Generally speaking two basic principles are used: polling and interrupt mechanism. Polling mechanism is used for sampling, and the handling of different tasks execution is achieved using an interrupt mechanism. Data acquisition flowchart is shown in Fig. 5. Assembler program for ADC includes all steps as described in the previous section that is required for the ADC operation. It was downloaded into the microcontroller using the PICkit2 In-Circuit Debugger/Programmer.

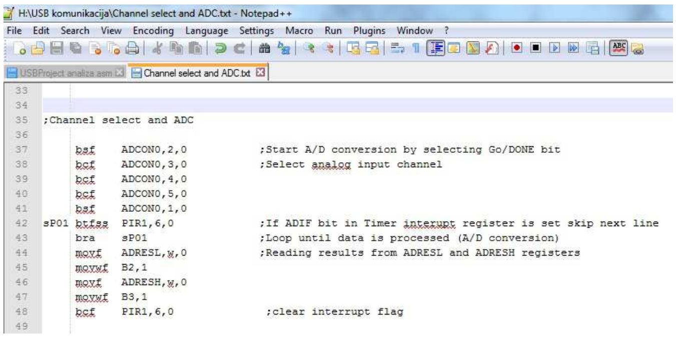
Assembler code sequence, which is used to configure and enable A/D module, comparator, data direction and interrupt registers is shown in Fig. 6. Also, code sequence for the selection of input channel, A/D conversion loop and storing the ADC results is shown in Fig. 7.

Timer predefined for ADC has to be reset to automatically repeat the A/D acquisition period with the minimal software overhead. The ADC module has the ability to generate an interrupt upon completion of an A/D conversion.




```
;Configuring A/D module parameters, enabling A/D conversion

      clrf    BSR                 ;Select bank0 in USB RAM
      CLRF    PORTA               ;Initialize PORTA by clearing output data latches
      MOVLW   07h                 ;Configure A/D by
      movwf   ADCON1              ;Selecting analog input bits and justification
      movlw   07h                 ;Value for digital inputs
      MOVWF   CMCON               ;Configure comparator for digital input
      movlw   6Fh                 ;Value used to initialize data direction
      movwf   TRISA               ;Set RA<3:0> as inputs
                                  ;RA<5:4> as outputs

      movlw   B'10100000'         ;Value used to initialize interrupt control register
      movwf   INTCON              ;Setup interrupt control register
      bsf     INTCON2,2           ;Hight priority overflow interrupt

      movlw   B'11000111'         ;Setting up timer control register
      movwf   T0CON
      bsf     ADCON0,0            ;A/D Conversion enabled

      movlw   B'10110101'         ;A/D conversion: Time Select =16 Tosc; Clock Select Fosc/4
      movwf   ADCON2

      bsf     PIE1,ADIE           ;Enables the A/D interrupt
      bsf     IPR1,6              ;A/D interrupt High priority
```

Fig. 6. ADC module configuration sequence




```


;Channel select and ADC

      bsf     ADCON0,2,0          ;Start A/D conversion by selecting Go/DONE bit
      bcf     ADCON0,3,0          ;Select analog input channel
      bcf     ADCON0,4,0
      bcf     ADCON0,5,0
      bsf     ADCON0,1,0
sP01  btfss   PIR1,6,0            ;If ADIF bit in Timer interupt register is set skip next line
      bra     sP01                ;Loop until data is processed (A/D conversion)
      movf    ADRESL,w,0          ;Reading results from ADRESL and ADRESH registers
      movwf   B2,1
      movf    ADRESH,w,0
      movwf   B3,1
      bcf     PIR1,6,0             ;clear interrupt flag

```

Fig. 7. A/D input channel selection and conversion

IV. Testing the Functionality of the Data Acquisition Module

Once the program code has been built and checked from the syntax point of view, it needs to be tested. In order to test the code, appropriate software for supporting the data acquisition module operation from the PC workstation is developed. It is developed using Borland Delphi programming environment (application window is shown in Fig. 8.). API (Application Program Interface) calls are transferred to USB port and to connected module with built-in system mcHID.dll library that is dynamically linked with the application. Since the used A/D converter has 10 bit resolution each analog input can be represented as a number ranged from 0 to 1023.

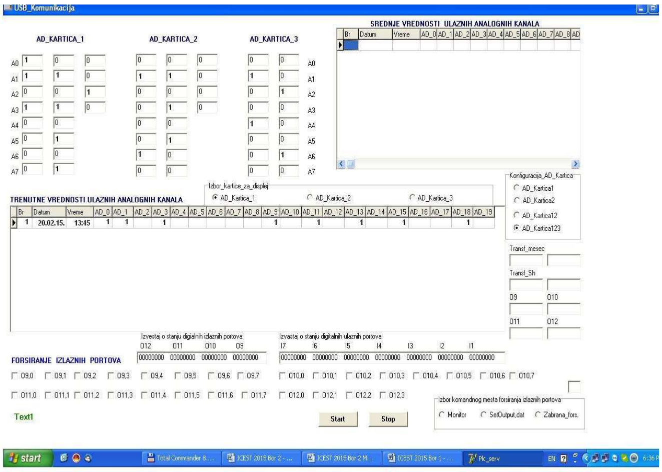


Fig. 8. Test application screenshot

Testing has revealed that for the input signal in the range of 0-3.75 V A/D conversion error of the new acquisition module, measured under laboratory conditions, is not greater than 1 LSB (0.0036 V). A/D conversion error of the old ADC module [8], for the same input range, is also 1 LSB (0.0146 V). The old ADC module has the 8-bit A/D converter so the accuracy of the new ADC module is actually four times higher.

V. Conclusion

From the need for a more precise A/D conversion, a new data acquisition module was realized. The appropriate programs for testing the operation of data acquisition module were downloaded into the microcontroller and installed on a personal computer. By testing the module in laboratory it was obtained that the A/D conversion accuracy is improved (four times better accuracy in comparation with the old ADC module).

Also, the data acquisition module works reliable, has USB interface, it is low-cost and enables further development. In the coming period it is necessary to develop adequate software on the PC workstation which will suit industrial requirements, or simply allow the monitoring of specific industrial processes. Minor corrections in the PIC program are also planned to allow some additional functionalities.

Acknowledgement

This work was supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia, project no. TR33037: “Development and Application of the Distributed System for Monitoring and Control of Electrical Energy Consumption for Large Consumers."

References

[1] D.R.Milivojevic, V.Tasic, “MMS in Real Industrial Network,” Information Technology and control, vol. 36, no. 3, pp. 318-322, 2007.

[2] Microchip Technology Inc., “PIC18F2455/2550/4455/4550 Data Sheet”, 2009.

[3] http://www.microchip.com/pagehandler/en-us/family/8bit/architecture/pic18.html (accessed 10 February 2015)

[4] Microchip Technology Inc., PICkit™ 2 Programmer/Debugger User’s Guide, 2008.

[5] MicroEngineering Labs, Inc., Pic Basic Pro™ Compiler Reference Manual, 2011

[6] http://ww1.microchip.com/downloads/en/DeviceDoc/33014L.pdf (accessed 10 February 2015)

[7] http://delphiandpic.jimdo.com/delphi-usb-y-herramientas/usb-hid/la-mchid-dll/ (accessed 10 February 2015)

[8] V.Tasić, D.Milivojević, M.Pavlov, V.Despotović, D. Brodić, “Microcontroller Based Systems for Peak Load Reduction,” Proceedings of 35th International Convention, MIPRO 2012, pp. 919-923, Opatija, Croatia, 21.5.-25.5.2012.