ЗАСОБИ ВИМІРЮВАНЬ ЕЛЕКТРИЧНИХ І МАГНІТНИХ ВЕЛИЧИН

DATA ACQUISITION MADE WITH MICROCONTROLLER ON THE 1-WIRE LINE

© *Borowik Barbara, Borowik Bohdan E., 2008*

Akademia Techniczno-Humanistyczna W Bielsku-Białej, ul Willowa 2
Wydział Budowy Maszyn I Informatyki, Katedra Elektrotechniki I Automatyki
borowik@tatrynet.pl

Our research aimed at exploring the possibility of the communication between the microcontroller and the 1-Wire device. After receiving communication data transfer, we were to acquire data from numerous 1-Wire devices, such as identification devices, temperature, humidity or other sensors, EEPROM Memory devices etc. Transferring data (and power) through the single line would be much more convenient, providing, that the transfer will be reliable and accurate.

In our study we used original testing board with the Microchip PIC 16F628 microcontroller. We also made use of the HP 16 500 A logic analyzer for controlling the waveforms on the 1-Wire bus. The logic analyzer appeared invaluable in testing time waves.

1-Wire devices, made by Maxim/Dallas, have the rich technical documentation. On the base of their Application Note 2420 we prepared an assembler program for the reading the 8-bytes long identification number from the iButton DS1990R.

However, obtaining the positive result in performing the communication on the 1-Wire bus appeared to be very hard. In our research we tried to employ as well the intermediate device, namely DS2480B Line Driver. Its role was to relieve the host microcontroller of generating the communication waveforms. It simply took over the role of the host.

Also we used the UART channel of the PIC microcontroller. All our efforts, however, remained without any positive result. Then we decided to come back and to connect the iButton to one of free lines on the I/O port of the PIC 16F628 microcontroller. Our choice was line 1 of PORT A, while most of the PORT B lines were dedicated to communicate with the LCD 2x16 display.

Positive communication results we achieved, when we adjusted manually the communication parameters, while simultaneously testing on the logic analyzer the resulting waveforms.

1. One-Wire network overview. 1-Wire system has a single bus master and one or more slaves. The role of bus master plays typically a microcontroller or PC. For small configurations the 1-Wire communication signals can be generated under software control using a single port pin. Alternatively can be used some devices of Maxim/Dallas: the DS2480B 1-Wire line driver chip or serial port adapters based on this chip (DS9097U series). This frees the microprocessor from responding in real time.

The 1-Wire protocol defines bus transactions in terms of the bus state during specific time slots that are initiated on the falling edge of sync pulses from the bus master. The 1-Wire bus has only a single line. It is important, that each device on the bus be able to drive it at the appropriate time. Therefore each device attached to the 1-Wire bus must have open drain or tri-state outputs.

A multidrop bus has multiple slaves attached. At standard speed the 1-Wire bus has a maximum data rate of 16.3 kbps. The value of the pull-up resistor primarily depends on the network size and load conditions. For most applications the optimal value of the pull-up resistor is approximately 4.7 kΩ. The idle state for the 1-Wire bus is high. If, for any reason, a transaction has to be suspended, the bus must be left in the High state, if the transaction is to resume.

All transactions on the 1-Wire bus begin with an initialization sequence. This sequence consists of a reset pulse transmitted by the bus master, followed by presence pulse issued by the slave. The presence pulse lets the bus master know, that the slave device is on the bus and is ready to operate.

Once the bus master detects a presence pulse, it can issue one of ROM function commands, that the slave device supports. All ROM function commands are 8 bits long. One of the function command is Read ROM (33h), that allows the bus master to read from the slave device its unique 64 bits long serial number.

One-Wire devices require strict protocol, to ensure data integrity. The protocol consists of four types of signaling on the 1-Wire bus: Reset/Presence Pulse, Write 0, Write 1 and Read Data. All those signals initiates the bus master.

To get from idle to active, the voltage on the 1-Wire line needs to fall. To get from active to idle, the voltage needs to rise. The time it takes for the voltage to make its rise depends on the value of the pullup resistor and capacitance of the 1-Wire network attached.

A Reset Pulse followed by a Present Pulse indicates that the slave device is ready to receive A ROM function command.

Data communication with the slave takes place in time slots. Each time slot carry a single bit. Write time slots transport data from bus master to slave. Read time slots transfer data from slave to master.

All communication begins with the master pulling the 1-Wire line low. As the voltage on the data line falls down, the slave device starts its internal timing generator, that determines when the data line will be sampled during a write time slot and how long data will be valid during a read time slot.

For a write 0 time slot the voltage on the line must stay low until the write 0 time is expired.

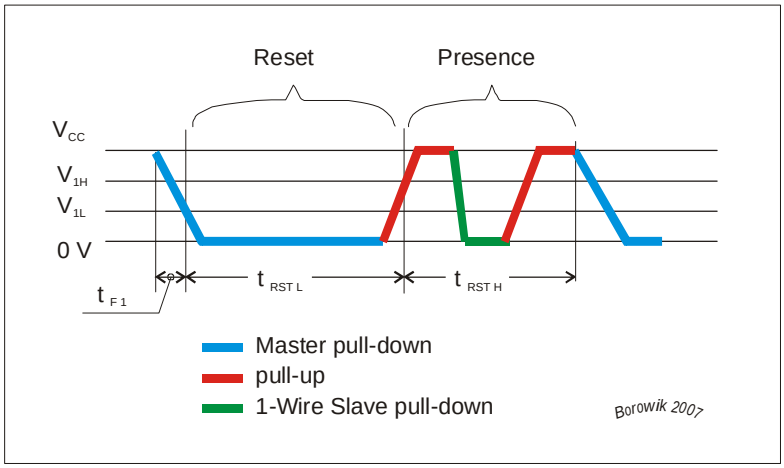

Fig 1. Reset/Presence pulse

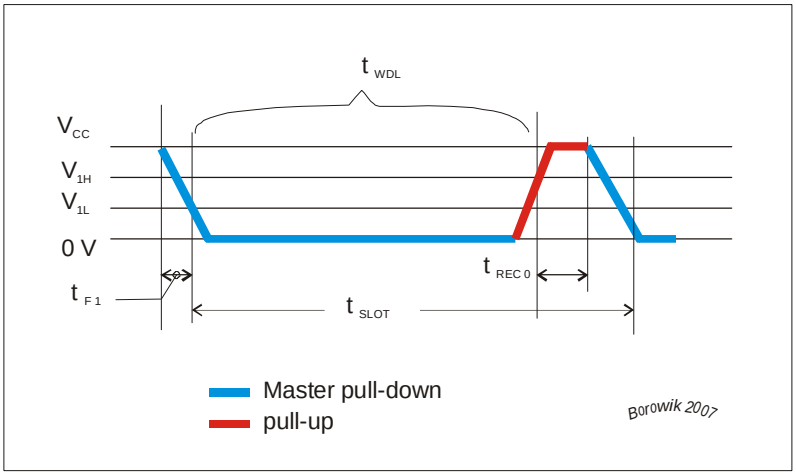

Figure 2. Write 0 Time Slot

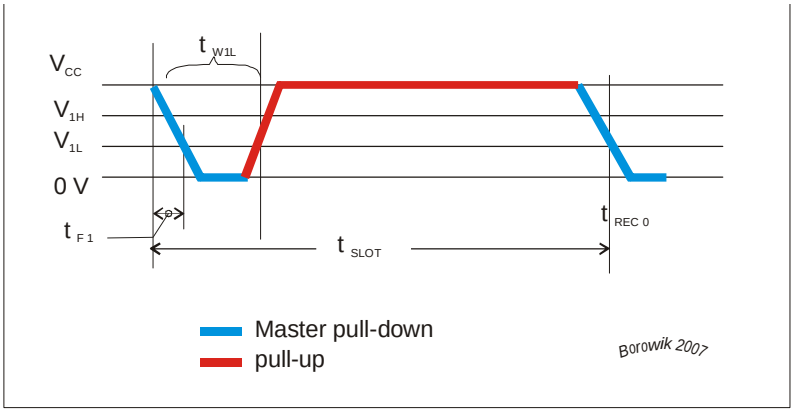

Figure 3. Write 1 Time Slot

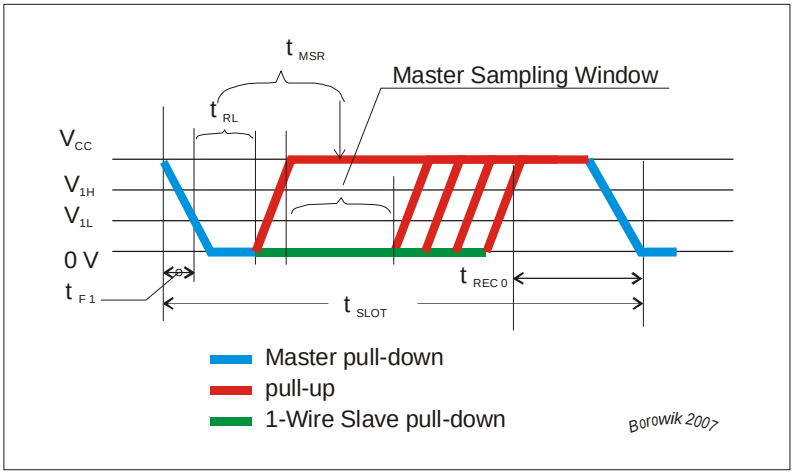

Figure 4. Read Data Time Slot

For a write 1 time slot the voltage on the line must have risen after the write –one low time is expired. After the voltage has risen the slave device needs the recovery time before it is ready for the next time slot.

A read-data time slot begins like a write–one time slot. The voltage on the line must be low over the time t_{RL} . During this time the slave will start pooling the data line low, when responding with 0 or wil not hold the line low at all when responding with 1.

2. Hardware description. The slave device takes the energy it needs to operate from the I/O line as indicated by the Parasite Power Circuit. Diagram on the figure 1 shows the ROM Function Unit, 1-Wire Interface and the logic implement the ROM function command, which accesses 64 bits of lasered ROM .

This system provides power and data delivery along with data bit level communication. Single contact 1-Wire interface provide an unmatched solutions to provide key functions to systems where interconnect must be minimized. Such system performs half duplex bi-directional communications between a host / master controller and slaves sharing a common data lines. Both power and data communications for slave devices are transmitted over this singe 1-Wire line.

For power delivery, slaves capture charge on an internal capacitor when the line is in high state and then use this charge for device operation, when line is low during data transmission. A typical 1-Wire master consist of an open drain I/O port with pull-up resistor connected to a 3-5 V supply. This type of communications also allows efficiently adding memory, authentication and mixed-signal functions. An important feature in 1-Wire system is unique, unalterable 64-bit factory lasered serial number ID in ROM that will never be repeated in another device. This 64-bit ID allows the master device to select a slave device among the many that can be connected to the same bus wire.

In the above schematic we have used DS2480B Line Driver, Multichannel RS232 Drivers/Receiver MAX232. Also we used the UART module of the PIC16F628. After the communications failure between DS2480B and UART

we made troubleshooting in the circuit. It was difficult to adjust the signaling protocol on the bus between Multichannel RS232 Drivers/Receiver MAX232 and DS2480B. We found out that it is better to use firmware controlled 1-Wire protocol and to remove all intermittent devices. This way the microcontroller directly controls 1-Wire Line and imposes appropriate data frame, time slots and recovery time. Through the assembler program stored in the microcontroller we can fully control the communications with iButton. During developing stage the program was tested with the Logic analyzer. Below is shown the schematic diagram of workable version of the 1-Wire communication system for data acquisition.

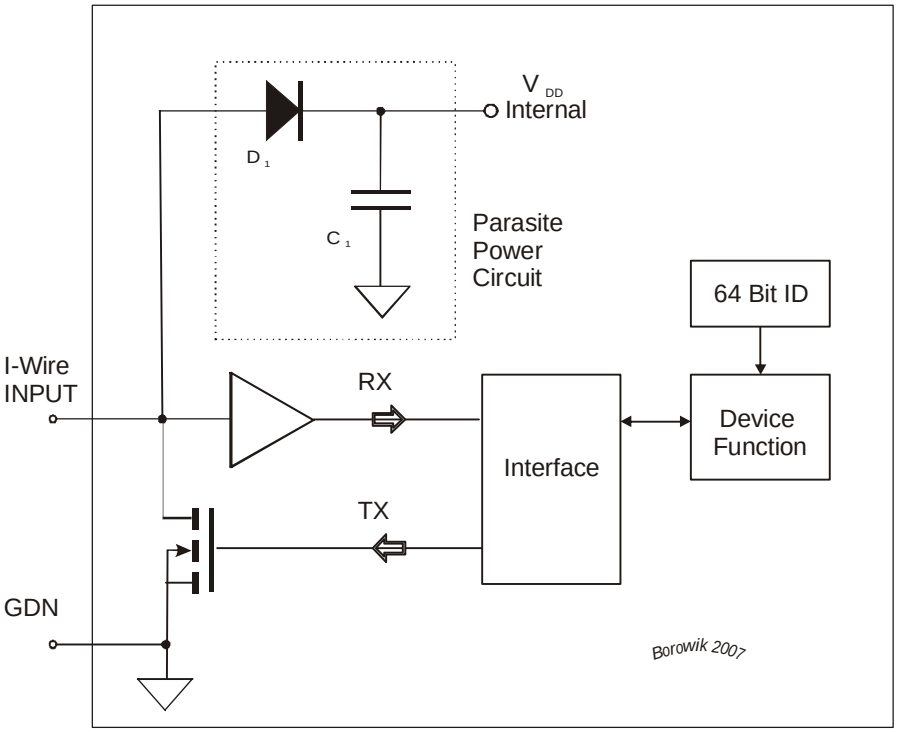

Figure 5. Block diagram of the 1-Wire Power Parasite Circuit with the ROM Function Unit.

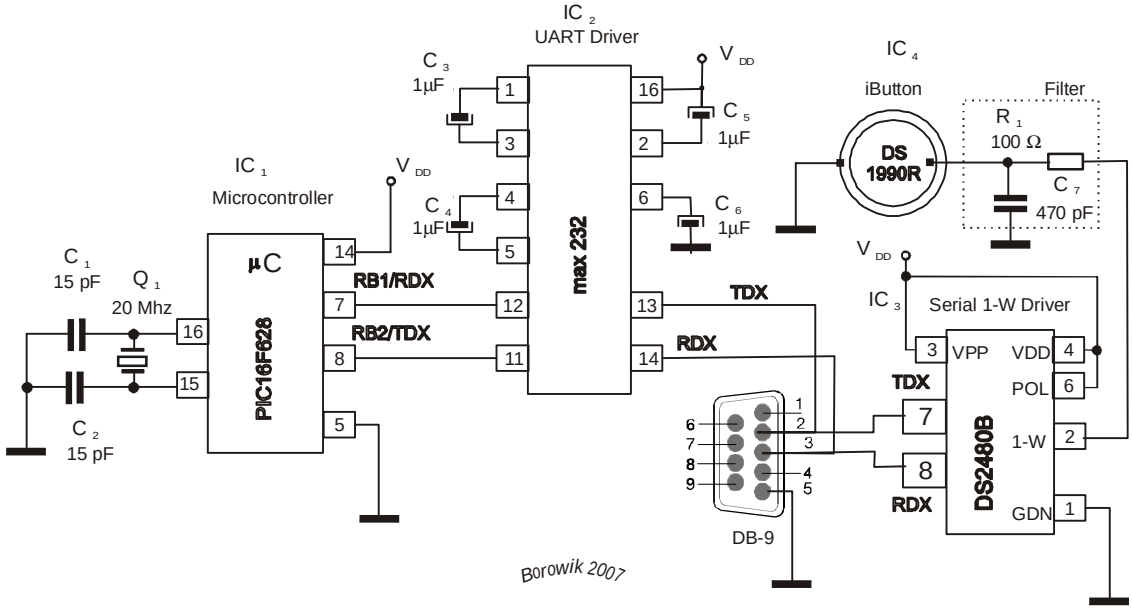

Figure 6. Schematic for the circuit of the 1-Wire Data Acquisition with DS2480 UART/1-W driver.

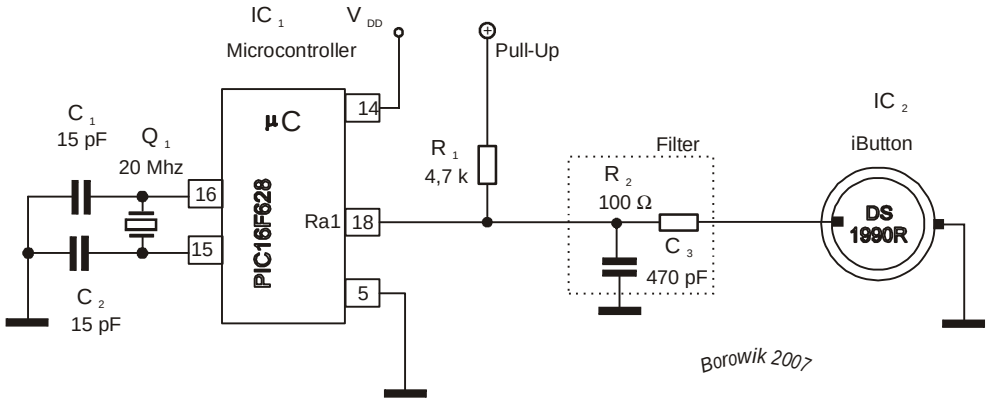

Figure 7. Schematic of workable version of 1-Wire communications system for Data Acquisition.

3. Selected parts of the software listing

```
;ow5.asm     communicating with the iButton DS1990R
. . . . . .
 #define DQ  porta,1      ; connected to 1-Wire
 #define DS  TRISA,1
. . . . . .


WAIT:MACRO   TIME
                    ;Delay for TIME µs.
                    ;Variable time must be in multiples of 5µs.
       MOVLW  (TIME/5) - 1 ;1µs to process
       MOVWF  TMP          ;1µs to process
       CALL   WAIT5U       ;2µs to process
       ENDM

OW_HIZ:      MACRO  ; 3 cykle m.
             ;Force the DQ line into a high impedance state.
       BSF    STATUS,RP0   ;Select Bank 1 of data memory
       BSF    TRISA, 1     ;Make DQ pin High Z
       BCF    STATUS,RP0   ;Select Bank 0 of data memory
       ENDM

OW_LO: MACRO  ; 5 cykli m.
                           ;Force the DQ line to a logic low.
       BCF    STATUS,RP0   ;Select Bank 0 of data memory
       clrf   PORTA  ;Clear the DQ bit
       BSF    STATUS,RP0   ;Select Bank 1 of data memory
       BCF    TRISA, 1     ;Make DQ pin an output
       BCF    STATUS,RP0   ;Select Bank 0 of data memory
       ENDM

       org 0
. . . . . .
; RESET I PRESENCE PULSE
       OW_HIZ
       nop
       OW_LO
```

```
       wait .480
       OW_HIZ
       wait .480
       ; sending the Read ROM command 33h
       wait .480    ; the recovery time after the reset/present pulse
       movlw 33h
       movwf  iobyte
       nop
       nop
       nop
       nop
       nop
       nop
       call wr_ds
       wait .480           ; the recovery time after sending the Read command
       ; reading the 8-bytes ROM ID
       movlw  .8
       movwf  licz2
odb3   call rd_ds
       movf   iobyte, w
       movwf  indf
       incf   fsr, f
       ;call crcw   ; for the sake of simplicity we didn't determined the CRC
       wait .480           ; the recovery time after receiving  all the ROM ID
       decfsz licz2, f
       goto   odb3
       bcf    OK_CRC, 0
;      movf   crc1, w             ; CRC wasn't determined
;      btfsc  status, Z
;      bsf    OK_CRC, 0
       goto $
wr_ds.       ; sending the byte to the slave device
       movlw  .8
       movwf  licznik
wr1
       OW_LO  ; 5 us
       nop
       nop
       nop
       rrf    iobyte, f
       bsf    status, rp0
       btfsc  status, C
       bsf    TRISA, 1
       bcf    status, rp0
       wait   .70
       OW_HIZ
       nop
       nop
;      nop
;      nop
       decfsz licznik, f
       goto   wr1
       return
. . . . . .
```

```
WAIT5U:
                          ;This takes 5µs to complete
       NOP                ;1µs to process
       NOP                ;1µs to process
       DECFSZ       TMP,F ;1µs if not zero or 2µs if zero
       GOTO   WAIT5U             ;2µs to process
       RETLW  0
. . . . . . . . . . .
rd_ds.       ; receiving the byte from the slave device
       movlw  .8
       movwf  licznik
rd2
       OW_LO
       ;nop
       OW_HIZ
       ;wait .5
       bcf    status, c
       btfsc  porta,1
       bsf    status, c
       rrf    iobyte, f
       wait   .70
       decfsz licznik, f
       goto   rd2
       movf   iobyte, w
       call piszh    ; for displaying on the LCD display the received
                     ; byte as two hexadecimal digits
       return
. . . . . . . . . . . . . .
;-------------------------------------------------------------
  end
```

4. Selected snapshots from the logic analyzer and their description. We attached three channels to the 1-Wire bus for the sake of clarity and for better triggering.

The first picture shows the reset and present pulse. The pictures don't display the slopes on the rising and falling edges, because of low sampling frequency of the logic analyzer.

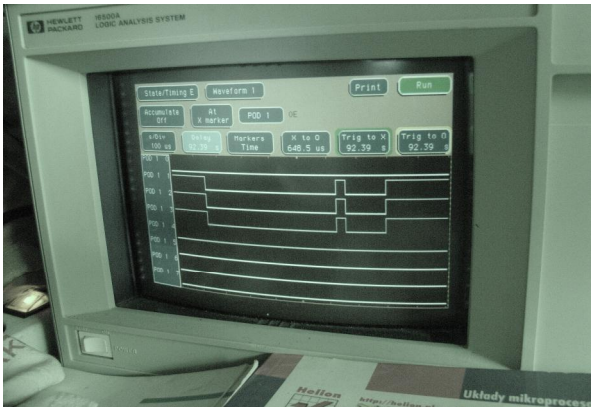

Figure 8. The snapshot of the reset and present pulse

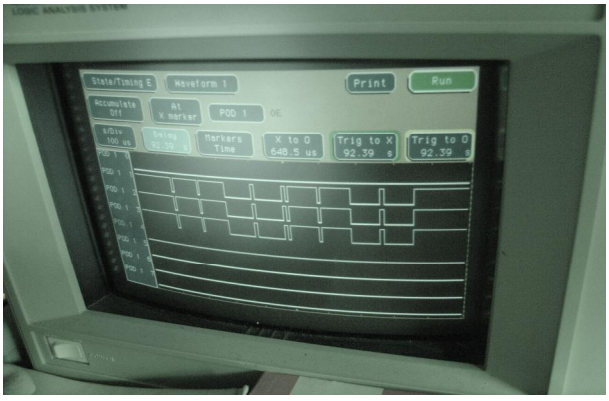

Figure 9. The snapshot of the sending the command READ ROM, 33h

The waveforms above show sending the command READ ROM, 33h, which is binary: 0011 0011. The LSB (Least Significant Bit) is being sent first, therefore the waveforms shows signaling: High High Low Low, High High Low Low. The byte is written by the host to the slave.

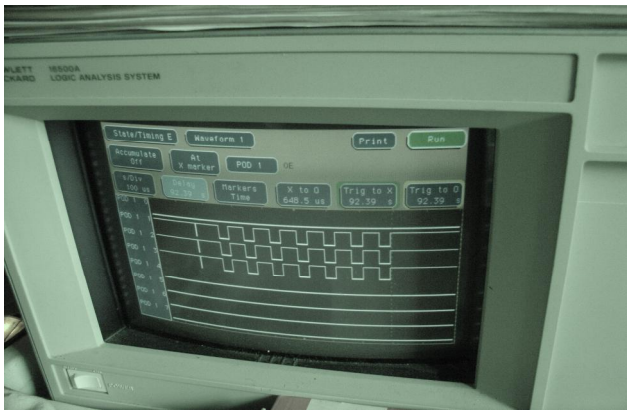

Figure 10. The snapshot of the first byte received from the slave

The third picture: The response from the slave. The first byte received is the family code, which for DS1990R iButton is 01h. It is binary: 0000 0001. In the reverse order (LSB goes first) it is being displayed as:

High Low Low Low Low Low Low Low.

After receiving each of 8 bytes we forced the recovery time of the length of 480 μs.

5. Conclusions

1. The software, written in the assembly language for the PIC 16F628 microcontroller was executed in the much longer time, than one could expect, counting the number of instruction and calculating due time through multiplying it by the time required for every instruction. The above is especially true, when the software consists of multiple subroutines, if it is long and nested.

2. When the 1-Wire network consists of one single slave device, only assuring the long enough recovery time allows to settle the communication signaling on the 1-Wire bus.

3. The recovery time, when the line is in the High state (see the duty cycle in the PWM) assures the appropriate amount of the parasite power for the slave device.

УДК 621.317.73

ПРОБЛЕМНО-ОРІЄНТОВАНИЙ АНАЛІЗАТОР ІМПЕДАНСУ НА ОСНОВІ СУЧАСНОЇ КОМПЛЕКТУЮЧОЇ БАЗИ

© Походило Євген[1], Бондирєв Віктор[2], 2008

Національний університет "Львівська політехніка", вул. С. Бандери, 12, Львів, Україна
[1]pokhodylo@polynet.lviv.ua, [2]chip_victor@rambler.ru

Описано практичну реалізацію вимірювальної системи з використанням мікросхеми інтегрального перетворювача імпедансу AD5933. Наведено експериментальні результати вимірювань. Проаналізовано перспективи застосування цього перетворювача імпедансу у портативних вимірювальних приладах.

Представлена практическая реализация измерительной системы с использованием интегрального преобразователя импеданса AD5933. Приведены экспериментальные результаты измерений. Проанализирована перспектива использования этого преобразователя импеданса в портативных измерительных приборах.

In the paper practical implementation of measuring system based on integral impedance converter AD5933is described. Experimental measurements results are presented. Prospects of presented impedance converter usage in portable measuring devices are analyzed.

Вступ. Імпедансні методи знаходять застосування в електрохімічному аналізі речовин, біологічних дослідженнях (імпедансна мікробіологія), імпедансній спектроскопії, для контролю якості елементів живлення та акумуляторних батарей, цементів, деревини, продуктів харчування тощо. Для реалізації та ширшого впровадження цих методів у практичну діяльність необхідною умовою є створення відповідної апаратної бази, яка би задовольняла такі критерії: точність і швидкість вимірювання, виконання вимірювань у широкому частотному діапазоні, можливість пересилання інформації у комп'ютер для подальшого оброблення і зберігання, мала споживана потужність, невисока собівартість.