



Ingeniare. Revista Chilena de Ingeniería

ISSN: 0718-3291

facing@uta.cl

Universidad de Tarapacá

Chile

Pérez Lobato, Eduardo; Muñoz Tapia, Marcelo; Ramírez Fernández, Jorge
A demodulator of PWM signals generated for a digital accelerometer is developed using a
microcontroller
Ingeniare. Revista Chilena de Ingeniería, vol. 14, núm. 2, mayo-agosto, 2006, pp. 119-123
Universidad de Tarapacá
Arica, Chile

Available in: <http://www.redalyc.org/articulo.oa?id=77214205>

- How to cite
- Complete issue
- More information about this article
- Journal's homepage in redalyc.org

redalyc.org

Scientific Information System
Network of Scientific Journals from Latin America, the Caribbean, Spain and Portugal
Non-profit academic project, developed under the open access initiative

A DEMODULATOR OF PWM SIGNALS GENERATED FOR A DIGITAL ACCELEROMETER IS DEVELOPED USING A MICROCONTROLLER

UN DEMODULADOR DE SEÑALES PWM GENERADAS POR UN ACELERÓMETRO DIGITAL ES DESARROLLADO USANDO UN MICROCONTROLADOR

Eduardo Pérez Lobato¹ Marcelo Muñoz Tapia² Jorge Ramírez Fernández³

Recibido 14 de marzo de 2005, aceptado 2 de enero de 2006

Received: March 14, 2005 Accepted: January 2, 2006

RESUMEN

Esta publicación presenta el uso de un microcontrolador para demodular dos señales PWM que están siendo generadas por un acelerómetro digital, obtener sus anchos y enviarlas en forma serial al puerto paralelo de un computador de propósitos generales.

Palabras clave: Acelerómetro, microcontrolador, control por computador, PWM.

ABSTRACT

This paper presents the use of a microcontroller to demodulate two Pulse Width Modulated (PWM) signals which are being generated by a digital accelerometer, to obtain their pulse widths and transmit them serially to a parallel port of a general purpose computer.

Keywords: Accelerometer, microcontroller, computer-controlled, PWM.

INTRODUCTION

Nowadays, there is a new generation of electronic accelerometers which are more precise, cheaper, smaller and with a wider range than their mechanical counterparts. The scope of its use is very wide, which usually involves force related applications, such as mechanical stress, rotary torque, seismic events, dynamic weight and so on. Thus, the introduction of these tiny digital accelerometers opens the chances of dynamically monitoring acceleration in a real time fashion. In addition, if these devices are integrated with general purpose computers, it is feasible to go beyond just displaying them on a screen, calculating their related energy and forces, etc., saving them up onto disk for further analysis and distributing them through the Internet using the Transmission Control Protocol/Internet Protocol (TCP/IP). These accelerometers, as integrated circuits, generate two PWM squared-waveforms as shown in Figure 1, and the value of the accelerations A_X and A_Y can be calculated through the equations shown in Figure

2 and Figure 3 respectively. To demodulate PWM signals, Karthaus and Fischer [1] used the envelope detection technique and Pascual [2] developed a low pass filter. This proposal deals with developing a PWM demodulator using a microcontroller as a single hardware piece.

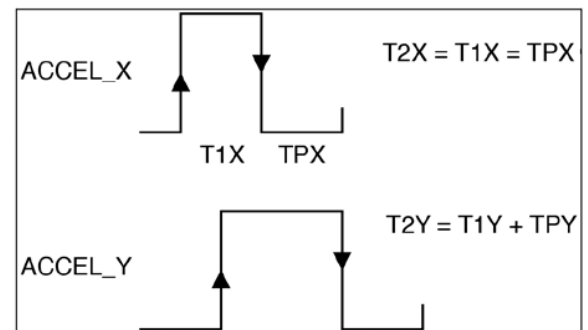


Figure 1. X-Y acceleration signals generated by the accelerometer.

¹ Departamento de Ingeniería Industrial, Facultad de Ingeniería Universidad de Antofagasta, Avenida Angamos 601, Antofagasta, Chile, perezlobato@uantof.cl

² Centro de Computación, Universidad de Antofagasta, Avenida Angamos 601, Antofagasta, Chile, mmunoz@uantof.cl

³ Departamento de Geomensura, Universidad de Antofagasta, Avenida Angamos 601, Antofagasta, Chile, jramirez@uantof.cl

$$AX = \left(\frac{T1X}{T2X} - 0.5 \right) * \frac{1}{8} \quad (1)$$

Equation that uses the T1X and T2X measured pulse widths to calculate the X-acceleration value

$$AY = \left(\frac{T1Y}{T2Y} - 0.5 \right) * \frac{1}{8} \quad (2)$$

Equation that uses the T1Y and T2Y measured pulse widths to calculate the Y-acceleration value

PWM DEMODULATOR REQUIREMENTS

To demodulate a PWM signal it is necessary to know either the width of the high state pulse and the width of the cycle. The present work presents the next strategy

1. Detect when a signal starts or finishes, either if it is in the high or low state. To accomplish this, detection either of the falling or raising edges is needed.
2. Counting slices of time whenever a falling or a raising edge has been detected.

To implement the above strategy, a microcontroller will be used, mostly because it has the following advantages

- Flexible detection either of the falling or raising edges through embedded programming.
- Easiness to count slices of time through the internal programmable timer.
- Fast development of an interface between the computer parallel port and a microcontroller, because electrical compatibility.
- Easy serial communication implementation, which includes protocol developing, handshaking control and data transmission through embedded programming.

PWM DEMODULATOR HARDWARE SETUP

Accelerometer. An ADXL-202 accelerometer was used as an acceleration sensor. It generates two PWM squared-waveforms signals, 90° shifted, 1mV/g resolution, ± 2 g scope and, with a 5 KHz central operation frequency

each, TTL electrically leveled, one designated as the ACCEL_X and the other as the ACCEL_Y.

Microcontroller. The general purpose PIC-16F84 was chosen because its technical features suits with the requested capabilities and its widespread use makes it a simpler task to integrate it with further projects.

Figure 2 depicts the block diagram between the computer parallel port, the microcontroller and the accelerometer, giving details only of the inter-connection lines used and their related pin outs.

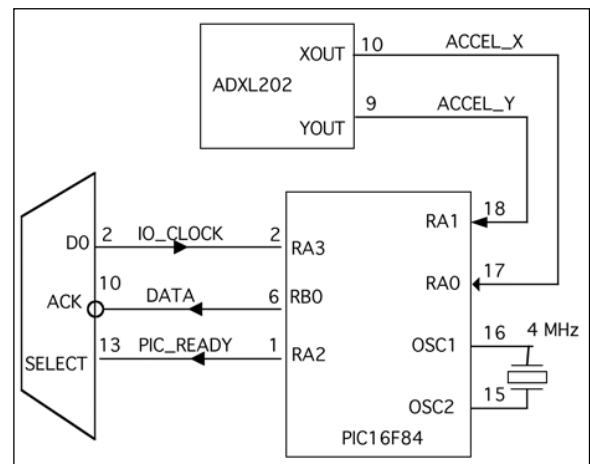


Figure 2. The general layout shows up the interconnected signal lines among the computer, the microcontroller and the accelerometer.

SEQUENCE TO DEMODULATE TWO PWM SIGNALS

To measure the width of the low and the high state of the ACCEL_X and ACCEL_Y signals, a time slice counting is carried out, using the internal timer TMR0, which was programmed to count slices of one microsecond, because both ACCEL_X and ACCEL_Y signals have a width time of 200 micro seconds each, so the high and low time are 100 microseconds width. Thus, since TMR0 is an 8-bit register, the maximum count is 255, or, 255 microseconds as maximum width time to record. The sequence to accomplish it is described below and the block diagram to demodulate the PWM signals is shown in Figure 3.

1. To measure the high state of the X-acceleration signal, the software waits until a rising edge of the ACCEL_X signal is detected. Once detected, TMR0 is activated to count until a falling edge of the ACCEL_X signal is detected and then TMR0 is stopped and its content is stored into the T1X register.

2. To measure the low state of the X-acceleration signal, the software waits until a falling edge of the ACCEL_X signal is detected. Once detected, TMR0 is activated to count until a rising edge of ACCEL_X is detected and then TMR0 is stopped and its content is stored into the TPX register.
3. To calculate the full width of the waveform, the content of the registers T1X and TPX are added and the result is stored in T2X register.
4. The above proceeding is repeated in order to have the pulse widths of the ACCEL_Y signal stored in the T1Y, TPY and T2Y registers.
5. The content of T1X is moved to TTY and then the SEND_TTY subroutine is called.
6. The content of T2X is moved to TTY and then the SEND_TTY subroutine is called.
7. The content of T1Y is moved to TTY and then the SEND_TTY subroutine is called.
8. The content of T2Y is moved to TTY and then the SEND_TTY subroutine is called.

SEQUENCE TO TRANSMIT 32 BITS TO THE COMPUTER PARALLEL PORT

The SEND_TTY subroutine transmits the 8-bit TTY register to the parallel port of the computer, in a bit to bit fashion, so the next actions are performed and repeated eight times. This subroutine is described below and the block diagram is shown in Figure 4.

- ❑ The PIC_READY line is raised to let computer know that the microcontroller is ready to send 32-bits.
- ❑ TTY is left-rotated, putting the most significant bit into the Carry bit of the microcontroller STATUS register.
- ❑ The Carry bit is outputted to the DATA line.
- ❑ A falling edge detection of the IO_CLOCK line is waited.

The PIC_READY line is turned to low let computer know that the microcontroller has already sent the eight-bits.

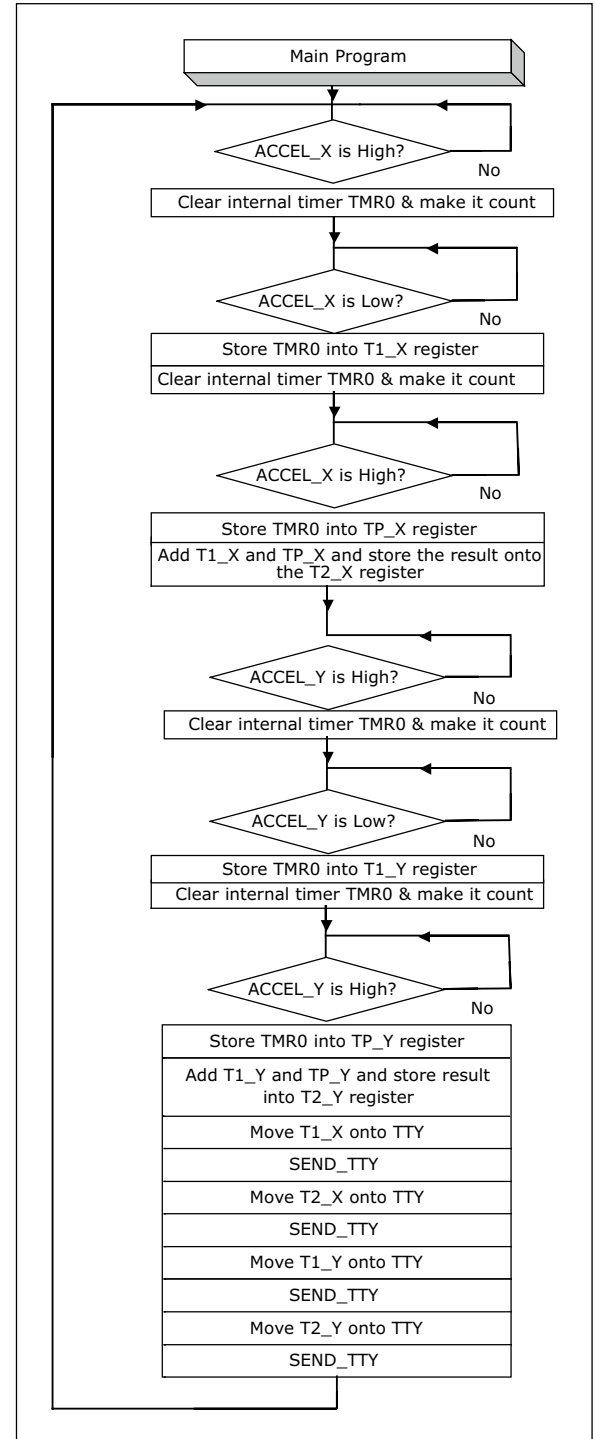


Figure 3. The algorithm shows up the full steps sequence of the PWM demodulator in order to obtain the AX and AY values and send them to the parallel port of the computer.

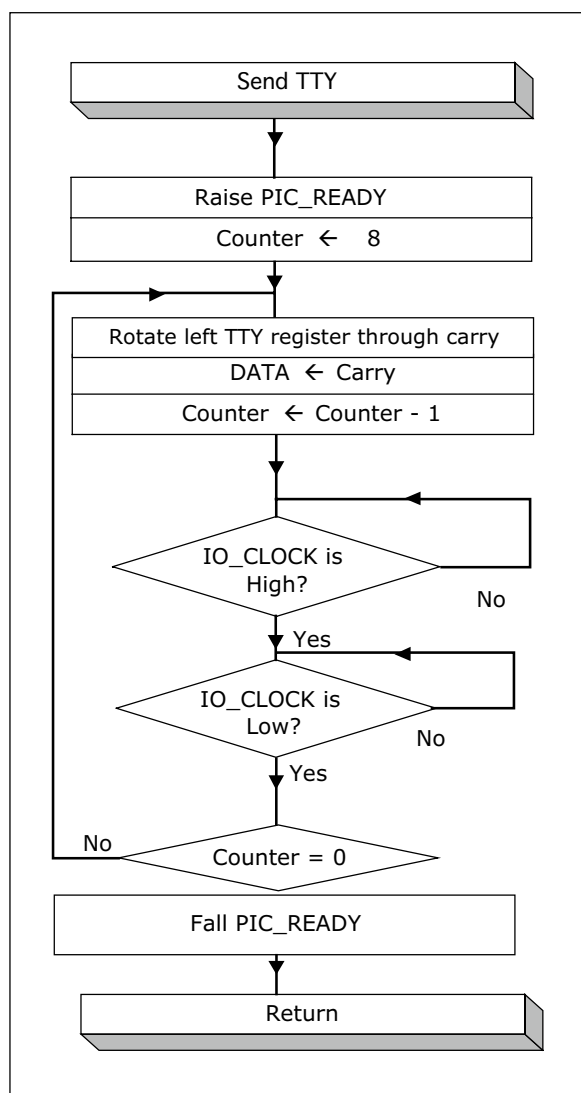


Figure 4. Communication routine between the microcontroller and the computer, along with the related handshaking.

CONCLUSIONS

Once built, this device was debugged and tested. There are some outcomes which are interesting to comment on.

- Developing, error correcting and modifying source code is by far a real easy task, taking into account it is the embedded hardware that is under a correction process. Analogous process with either active or passive components is clearly a far difficult task to make.

- One chip demodulator means tiny space and light weight, which makes it extremely portable, either to locate it wherever in the lab or install it in remote sites, such as in seismic applications.
- Old mechanical accelerometers, pendulum style, along with their heavy carcasses and gears, are several times more expensive than the digital ones.
- Use of the parallel port instead of the serial port, because a simpler protocol could be devised.

ACKNOWLEDGMENTS

The authors wish to thank to the Universidad de Antofagasta, which funded the Project PEI-1334-01, named “Construcción de un acelerógrafo prototipo inteligente para multipropósitos, sismo resistentes...”. Besides we want to thank to the European Southern Observatory organization and Compañía Minera Zaldivar for help funding this work.

REFERENCES

- [1] U. Karthaus, M. Fischer. “Fully Integrated Passive UHF RFID Transponder IC with 16.7 mW minimum RF input power”. IEEE Journal Of Solid-State Circuits. Vol. 38. N° 10. October 2003.
- [2] C. Pascual, Z. Song, P.T. Krein, D.V. Sarwate. “High-Fidelity PWM Inverter for Audio Amplification Based On Real-Time DSP”. IEEE Transactions on Power Electronics. Vol. 18, pp. 473 485. January 2003.
- [3] J.A. de Lima, S.F. Silva, A.S. Cordeiro. M. Verleysen. “A CMOS/SOI Single-input PWM Discriminator for Low-voltage Body-implanted Applications”. VLSI Design. Volume 15. N° 1, pp. 469-476(8). January 2002.
<http://www.dice.ucl.ac.be/~verleyse/papers/vlsidesign02jdl.pdf>
- [4] C. Bolton. “PLD code creates PWM generators”. EDN, pp 108.110. August 8, 2002.
<http://www.edn.com/contents/images/80802di.pdf>
- [5] J. Mahoney. “Circuit converts pulse width to voltage”. EDN, pp.92-94. October 25, 2001.
<http://www.edn.com/contents/images/102501di.pdf>

APPENDIX A. SOURCE CODE EMBEDDED INTO THE MICROCONTROLLER

The next assembler list shows the assembler source code either of the PWM demodulator and the transmitter, which was written onto the PIC16F84 with a EEPROM Writer.

```

ASM
T1X    equ 0x10
T2X    equ 0x11
TPX    equ 0x12
      T1Y    equ 0x13
T2Y    equ 0x14
TPY    equ 0x15
cnt    equ 0x16
TTY    equ 0x17
TOCS   equ 0x01
bcf    OPTION_REG, PSA
bcf    OPTION_REG, TOCS
bcf    OPTION_REG, PS2
bcf    OPTION_REG, PS1
bsf    OPTION_REG, PS0
bcf    STATUS, RP0
clr    PORTA
clrf   PORTB
movlw  0x0B
movwf  TRISA
movlw  0x00
movwf  TRISB
Init_X    nop
          bcf  PORTA,2
          btfss PORTA,0
          GoTo Init_X
          clrf  TMR0
Rise_edge_X  nop
          btfsc PORTA,0
          GoTo Rise_edge_X
          movf  TMR0,0
          movwf T1X
          clrf  TMR0
Fall_edge_X  nop
          btfss PORTA,0
          GoTo Fall_edge_X
          movf  TMR0,0
          movwf TPX
          movf  TPX,0
          addwf T1X,0
          movwf T2X
Init_Y    nop
          btfss PORTA,1
          GoTo Init_Y
          clrf  TMR0

```

```

Rise_edge_Y  nop
              btfsc PORTA,1
              GoTo Rise_edge_Y
              movf  TMR0,0
              movwf T1Y
              clrf  TMR0
Fall_edge_Y  nop
              btfss PORTA,1
              GoTo Fall_edge_Y
              bsf  PORTA,2
              movf  TMR0,0
              movwf TPY
              movf  TPY,0
addwf T1Y,0
              movwf T2Y
              movf  T1X,0
              movwf TTY
              Call  PIC_to_PC
              movf  T2X,0
              movwf TTY
              Call  PIC_to_PC
              movf  T1Y,0
              movwf TTY
              Call  PIC_to_PC
              movf  T2Y,0
              movwf TTY
              Call  PIC_to_PC
              bcf  PORTA,2
              GoTo Init_X
PIC_to_PC
              nop
LowSt        nop
              btfss PORTA,3
              GoTo LowSt
HiSt        nop
              btfsc PORTA,3
              GoTo HiSt
              movlw 0x08
              movwf cnt
Loop        rlf  TTY,0
              btfss STATUS,C
              bcf  PORTB,0
              btfsc STATUS,C
              bsf  PORTB,0
              decfsz cnt,1
              GoTo Loop
              nop
Return      ENDASM

```