APPLICATION OF THE PHOTODIODE IN DESIGN AND IMPLEMENTATION OF A 2-D POSITION DETECTOR

BY

JAE H. CHA

PROJECT SUBMITTED TO THE FACULTY OF THE

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

ELECTRICAL ENGINEERING

APPROVED:

DR. F. RICCI, CHAIRMAN

DR. D. SCHAEFER

DR. J. KNIGHT

MAY 1994

Blacksburg, Virginia

Table of Contents

1. Foreword

Silicon photodiodes are optoelectronic transducers which convert a light signal into an electrical signal. They have been used in many applications such as position and angle measurement, remote optical alignment, and guidance systems. The purpose of this project is to design and implement a low cost position sensing system which uses a silicon photodiode to detect two dimensional (2-D) x, y coordinates of a small light source. As a result, a working prototype is built and also tested; it is able to produce promising results although simple in construction.

2. Summary

The goal of this project is to design and build a 2-D position sensing system. The design is composed of three major tasks: (1) design of the optoelectronic circuit, (2) design of the data acquisition and control circuit and (3) design of the system software. For the major sensor, the SC-10D photodiode manufactured by the United Detector Technology Inc. (UDT Sensors) is used, and for the system controller, the 8051 microcontroller manufactured by Intel is used.

During the operation, the current output from the SC-10D is converted into a voltage signal by operational amplifiers, and then converted into an 8-bit digital signal by the ADC0809 analog-to-digital (A/D) converter. The 8-bit data is then transmitted through the RS-232C channel to a host computer where numerical processing of the data is performed and the position values are displayed on the CRT screen. The system can detect the 2-D position of a small light source at 2 to 3 m. Methods of improving the resolution and the operating distance are also explored. The functional block diagram is shown in Figure 2.

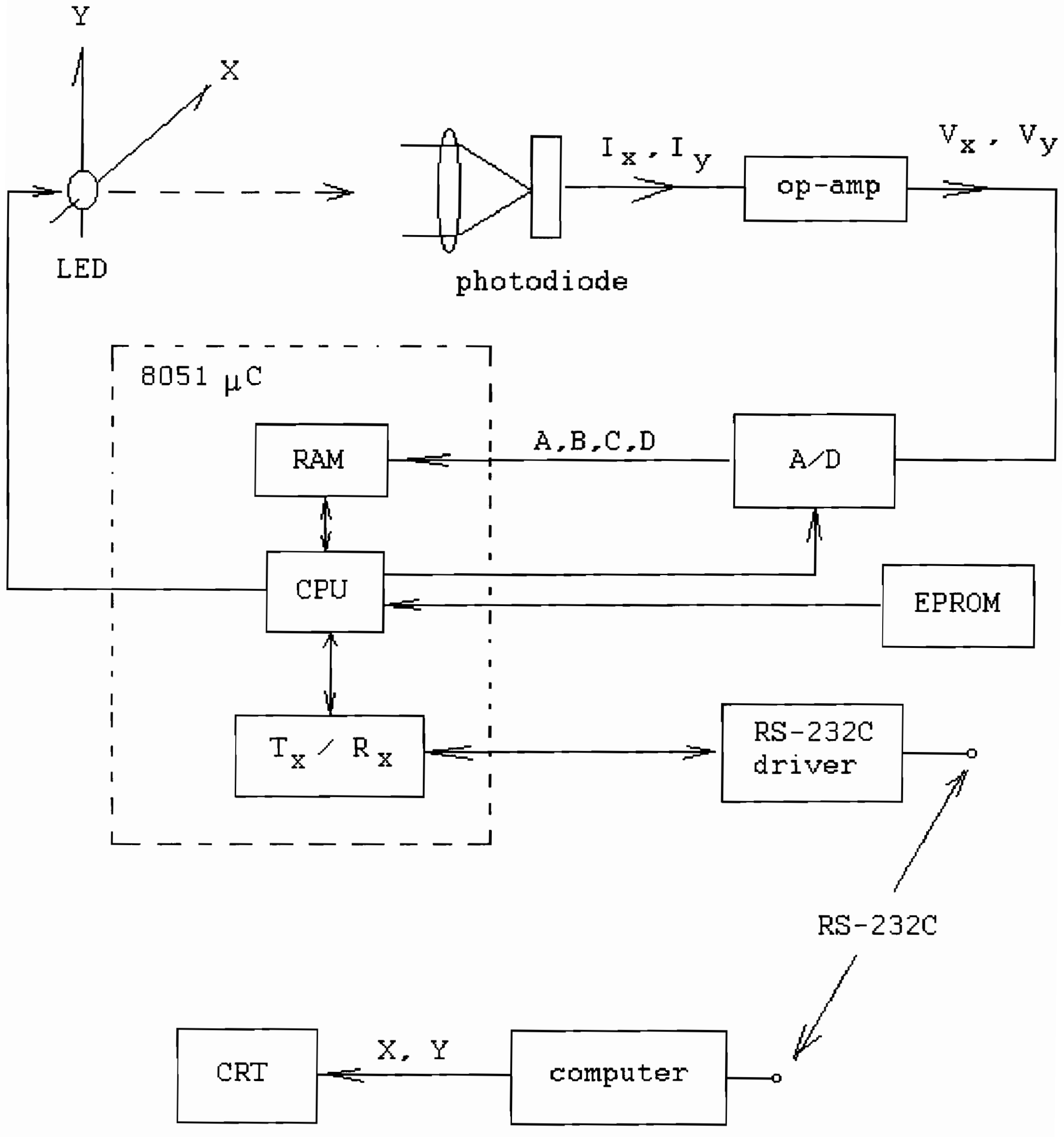

Figure 2. The Functional Block Diagram

3. Introduction

Silicon photodiodes have been around for more than two decades and have gotten much popularity due to their long term stability, output linearity and wide light level detection capability[1]. It is known that by using a photodiode, very accurate position measurement can be achieved down to the order of 0.1 mils[2]. Nowadays, many position and angle measurement systems are commercially available and typically cost several thousands of dollars. As mentioned earlier, the purpose of this project is to design and implement a low cost 2-D position sensing system by using a silicon photodiode. Its level of sophistication can never be compared to those expensive commercial systems'. It is, however, possible to get reasonably good performance results with minor improvements in the optic part of the system. During the design and development, the following objectives are considered:

(1) simpler design
(2) digital processing
(3) flexibility of data handling
(4) interface capability with computers
(5) long range detection
(6) inexpensive system

4. Optoelectronic Design

4.1 Photodiode

The silicon photodiode selected for the project is the SC-10D position sensing photodiode manufactured by the United Detector Technology Inc. It has one anode and four cathodes connected to four sides of the active photo-sensitive area of 0.4 in x 0.4 in (Illustration 4.1 a and b) For convenience, symbols are arbitrarily assigned to the cathodes to distinguish them throughout the project, namely, A, B, C, and D (Illustration 4.1 c).

The manufacturer's specification is shown in Appendix D[3].

4.2 Light Source

The light source used is the LN261CAL(UR) high brightness GaAlAs light emitting diode (LED) manufactured by Panasonic. This LED generates red light at 665 nm with sufficient intensity for the project. The manufacturer's specification is shown in Appendix E[4].

4.3 Lens and Filter

The lens used is a student-grade double convex lens. It is 27 mm in diameter with the focal length of 46 mm. The optical filter used has the bandpass range from 600 nm to 700 nm with 85 % average transmission efficiency.

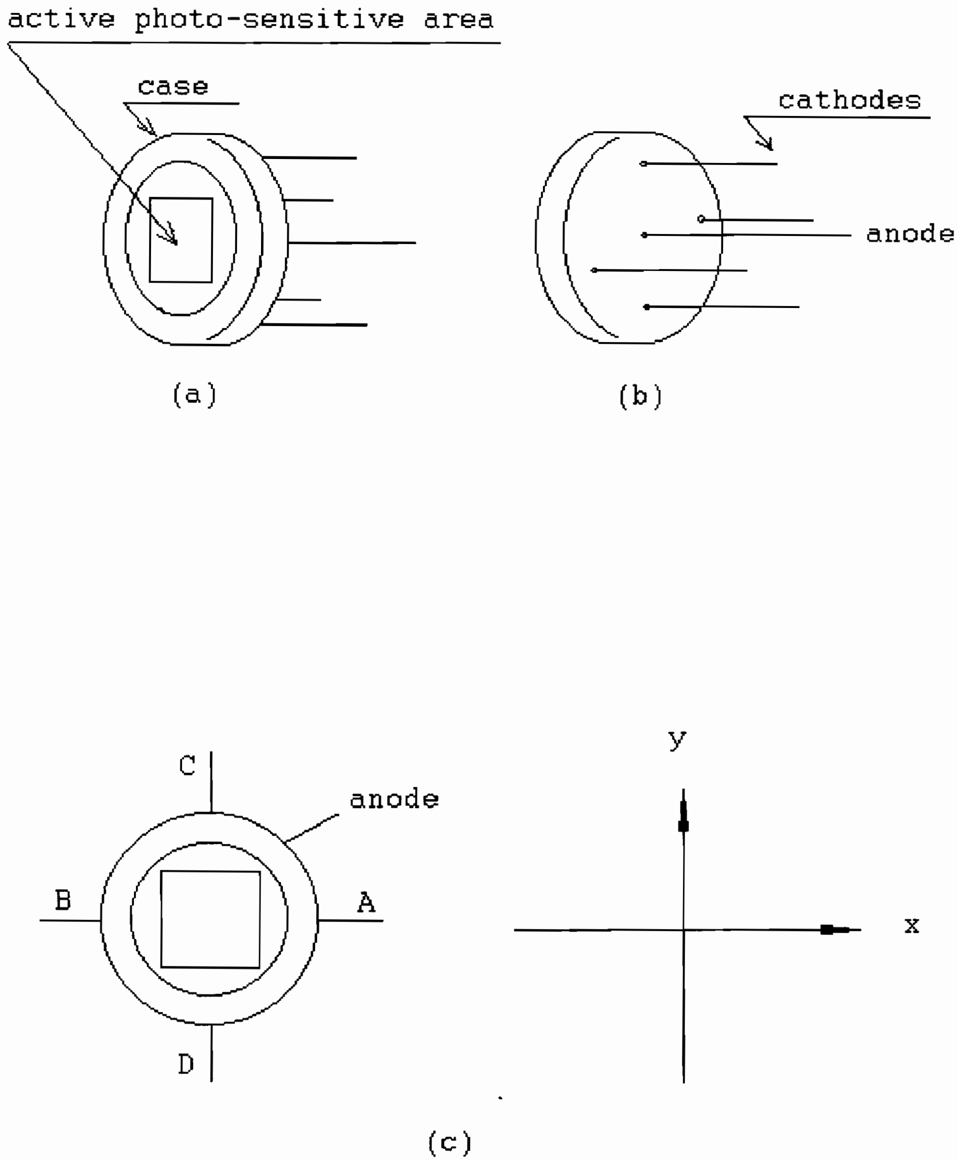

Illustration 4.1 SC-10D (a) front (b) back

(c) symbol assignment to the cathodes and x, y coordinates

Assembling the lens, the filter, and the photodiode required some mechanical skills. Perhaps this part of the project needs more attention in the future improvement (See Illustration 4.3). Analytical design is discussed in Appendix F.

4.4 Support Electronics

The purpose of the surrounding electronics circuit is basically to provide voltage outputs as a function of the x, y coordinates of a light source. The voltage outputs are then digitized and processed by the host computer.

When the SC-10D is biased, a negative voltage is applied to the anode to put it in the photoconductive mode[8]. For this project the negative 5 V bias is enough to get a good response. It was observed that connecting the anode to ground is acceptable, which would put the photodiode in the photovoltaic mode. The response is a little slow but it does not cause much change in the overall performance of the system.

The current output from the SC-10D is converted into a voltage output by the LM324N quad operational amplifier (op-amp) configured as a transconductance amplifier shown in Figure 4.4.1. In principle, the output voltage is given by

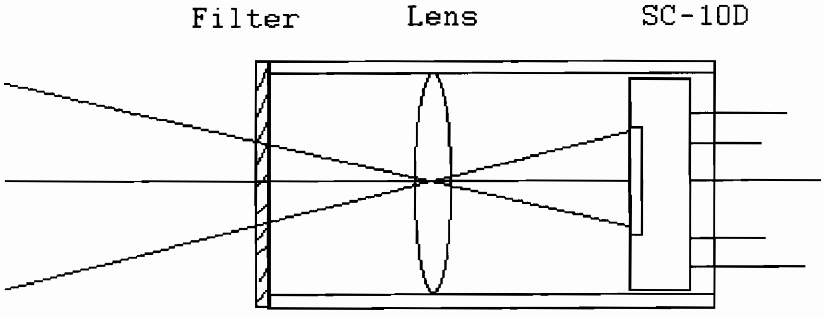

Illustration 4.3 Lens and Photodiode Assembly

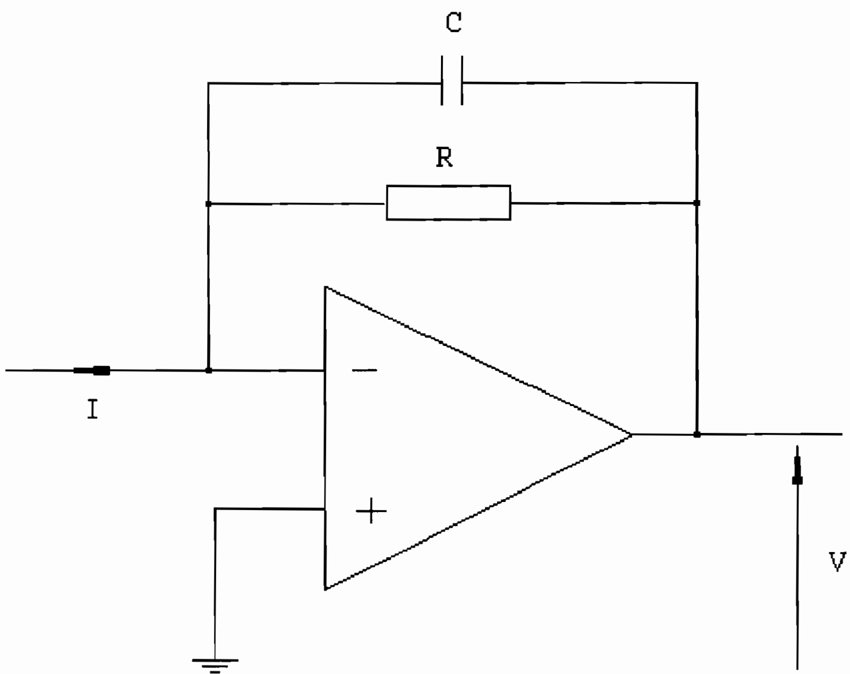

Figure 4.4.1 Transconductance amplifier, R = 1MΩ, C = 80pF

$$V = R \times I$$

assuming an ideal op-amp. While choosing R value, it was observed that the exact value of R can be determined empirically to achieve the accurate balance between the cathodes.

It is important to deliver stable inputs to the A/D converter. Otherwise, digitizing will result in erroneous values. More work needs to be done to isolate the noise sources and to prevent the inputs from being corrupted by the spurious noises. The capacitor C (See Figure 4.4.1) helped filter out some high frequency noises.

In Figure 4.4.2 is shown the wiring of the photodiode and the op-amps. The voltage outputs, A', B', C', and D' are fed to the data acquisition circuit.

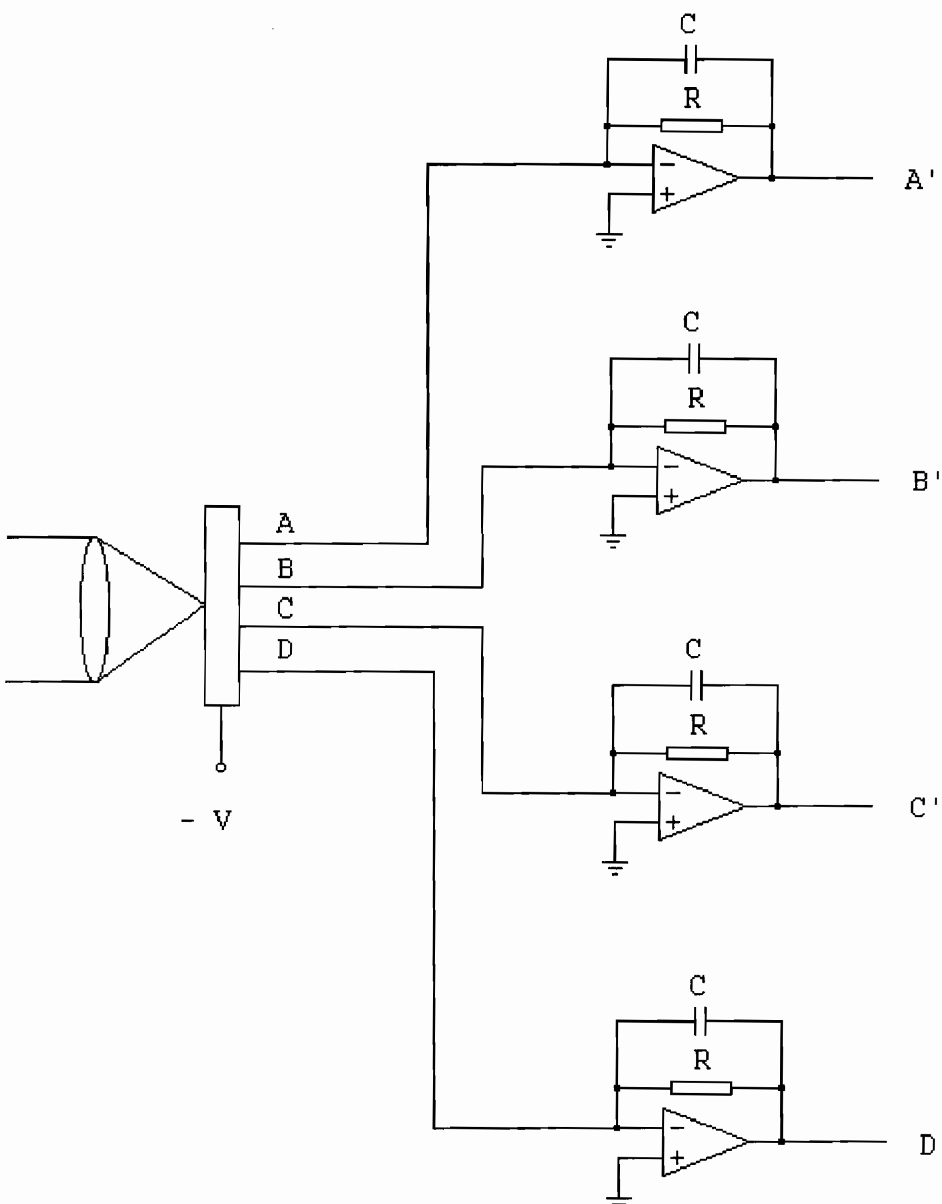

Figure 4.4.2 Photodiode and Op-Amp Circuit

5. Data Acquisition and Control Circuit Design

The purpose of this portion of the project is to design a support digital hardware for Section 4.

5.1 Data Acquisition

The data acquisition is handled by the ADC0809 A/D converter. Since almost all components are on the chip, the circuit design could be greatly simplified. There are 8 multiplexed input channels to which the four outputs from the LM324N are connected; only channel 0, channel 1, channel 2, and channel 3 are used. Access to these channels is based on the memory mapped I/O scheme; the channels have corresponding memory address starting at 800h.

An attempt to write to a channel triggers the start of A/D conversion of that channel. When the conversion is over, an interrupt signal is sent to the microcontroller (See Section 5.2) and the digitized data is stored in the internal buffer of the ADC0809. Reading the data in is the last step of the data acquisition. The address assignment to the channels is shown in Table 5.1.

The ADC0809 interfacing is shown in Figure 5.1. The address lines, A0, A1, A2, and A11 determine the channel addresses.

Table 5.1 I/O Memory Mapping

address	corresponding ADC0809 channel
800h	channel 0
801h	channel 1
802h	channel 2
803h	channel 3

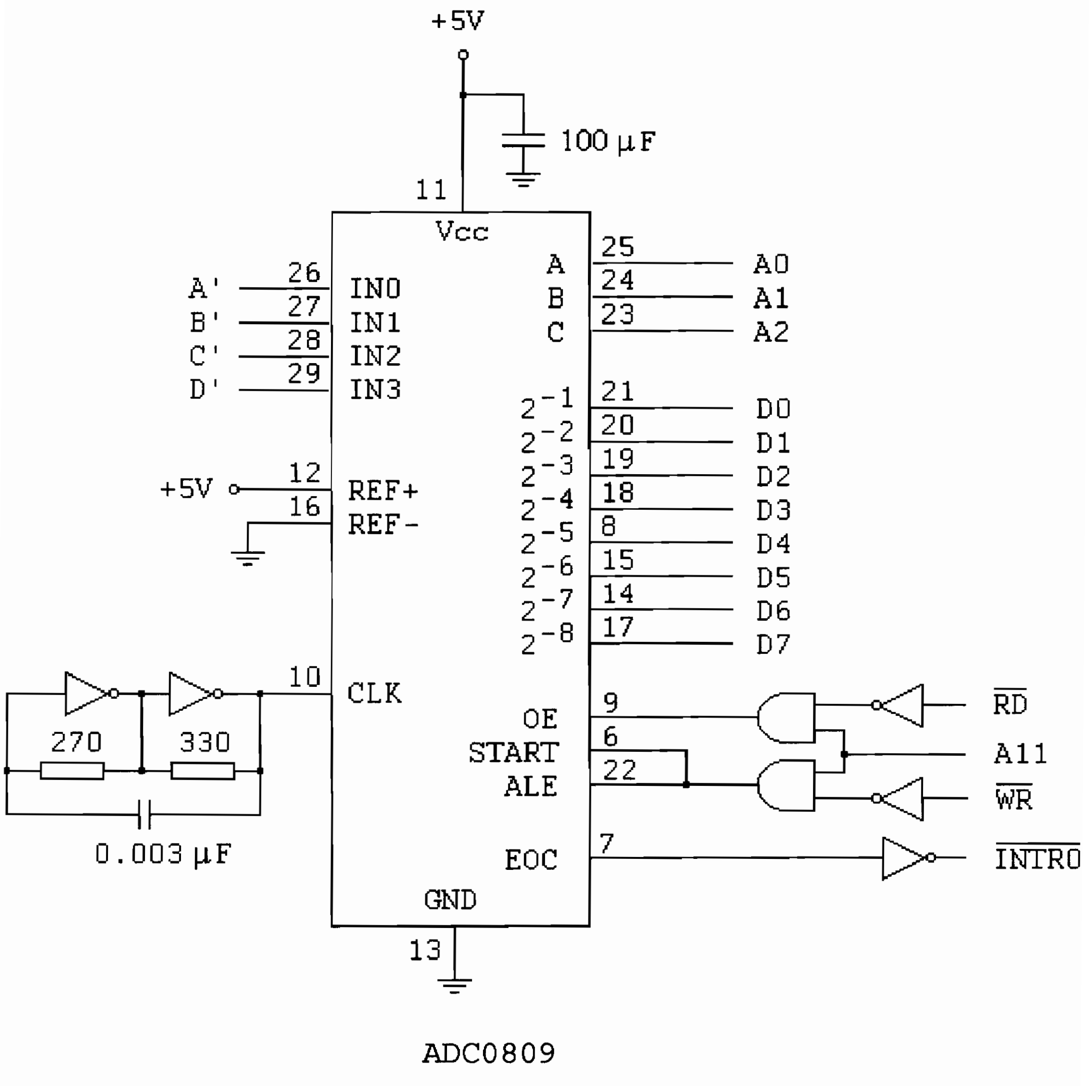

Figure 5.1 ADC0809 Interfacing

5.2 System Control

Several tasks are identified as necessary for the proper operation of the system. They are as follows.

(1) turn on/off the LED

(2) initiate the A/D conversion

(3) store the A/D converted data in memory

(4) transmit the A/D converted data to the host computer through the RS-232C channel

The nature of the above tasks calls for the use of a microcontroller which is a low cost, programmable one-chip computer. Among many brands available on the market, the 8051 microcontroller by Intel is selected for its popularity and functional capability. The 8051 has 4 Kbytes of factory-masked ROM, 128 bytes of RAM, 32 I/O lines, two 16 bit counter/timers, a duplex serial port and up to five sources of interrupts[7]. The factory-masked on-chip ROM is useless and can be bypassed by connecting the pin #31 of the 8051 to ground. The system program is stored in the external EPROM. (See Figure 5.2.1)

The amount of the internal RAM is plenty enough for this project. This eliminated the need for the external RAM.

The control signals, $\overline{RD}$, $\overline{WR}$, $\overline{INTR0}$ are connected to the

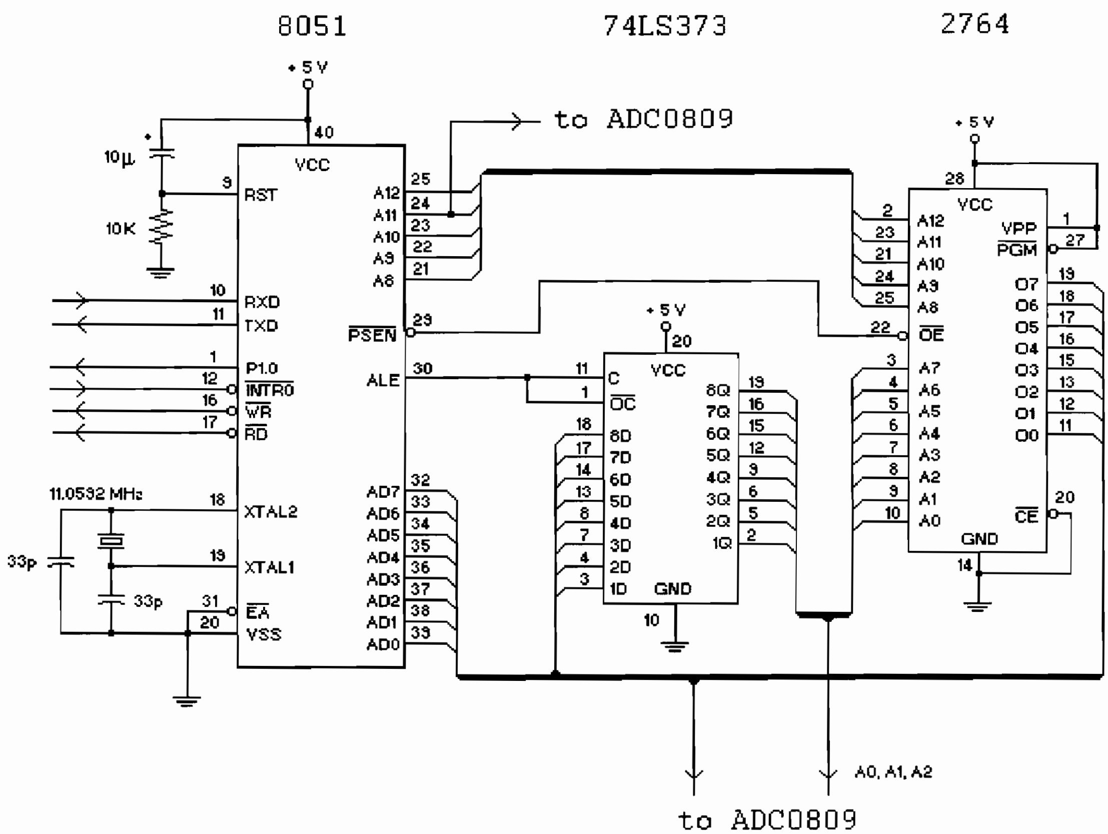

Figure 5.2.1 the 8051 with 8 Kbyte EPROM

ADC0809. Another control signal, P1.0, is connected to the LED driver circuit to turn the LED on/off. (Figure 5.2.2)

5.3 Serial Data Communication

The serial communication with the host computer is based on RS-232C. In RS-232C standard, one is designated as a Data Communications Equipment (DCE) and the other as a Data Terminal Equipment (DTE). In this project, the host computer is assumed to be a DTE; when the DB-25 socket is used, the pin #2 is used for transmission and the pin #3 for reception. The opposite is true for a DCE; the pin #2 for reception and the pin #3 for transmission. The pin #7 is the signal ground. The 8051 as a DCE is programmed to send 8-bit data at 9600 baud rate with one start bit, one stop bit and no parity bit.

The circuit diagram for the serial communication is shown in Figure 5.3. Both transmitter and receiver are packaged in one IC, MAX232N, which requires a sigle +5V supply.

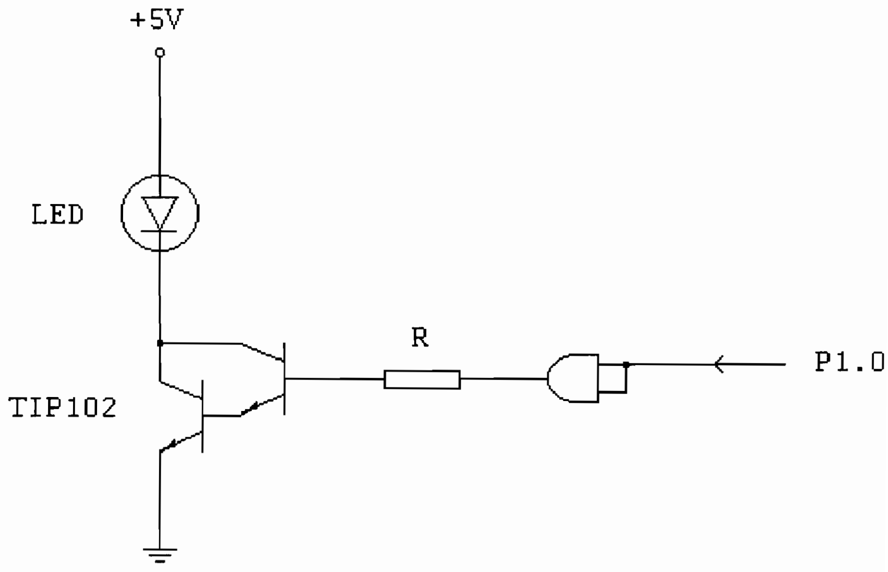

Figure 5.2.2 LED Driver Circuit

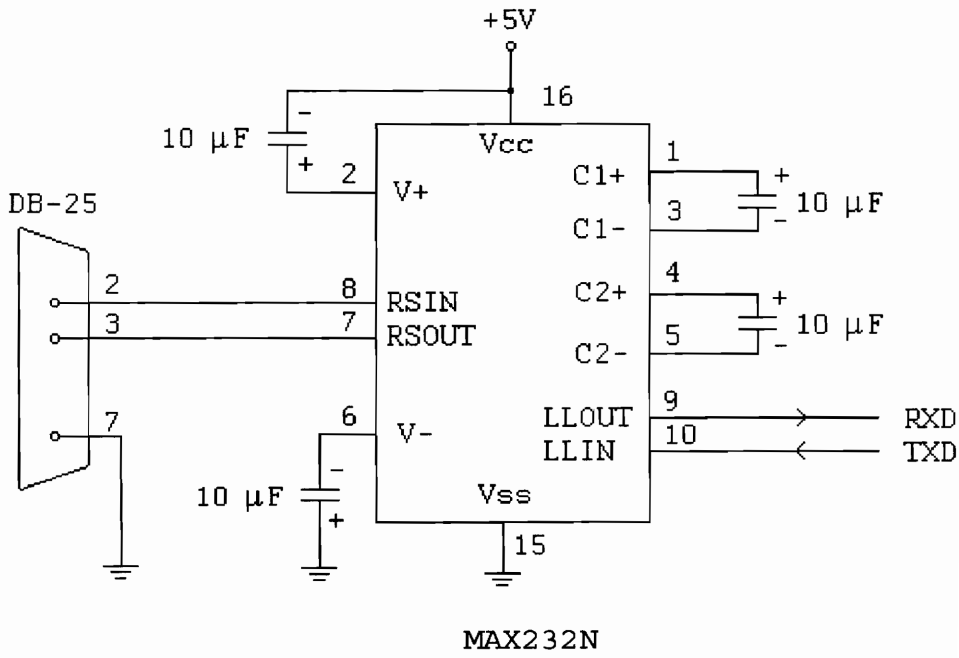

Figure 5.3 Serial Communication Circuit Diagram

6. System Software Design

6.1 The 8051 Programming

The functional requirements are as specified in Section 5.2. At the start of the process, the LED is turned on and then A/D conversion begins after a certain delay. (See Figure 6.1.1) This delay is necessary to compensate the response time of the photodiode and the op-amps. About 0.25 msec delay allows the voltages to reach their steady state.

The voltage outputs, A', B', C', and D' (See figure 4.4.2) are then successively digitized and stored in the internal RAM of the 8051.

The LED is turned off after all channels are digitized since the LED is no longer needed and the serial transmission takes place for the rest of the processing. This allows the LED to deliver the high power output without exceeding the maximum power rating and also to keep the average illumination below the safety level.

The serial transmission begins with sending a one-byte header followed by four bytes of digitized data. The header is actually a command received from the host at the start of the process. The purpose of sending back the command is to check the system's operation. The host can check if the

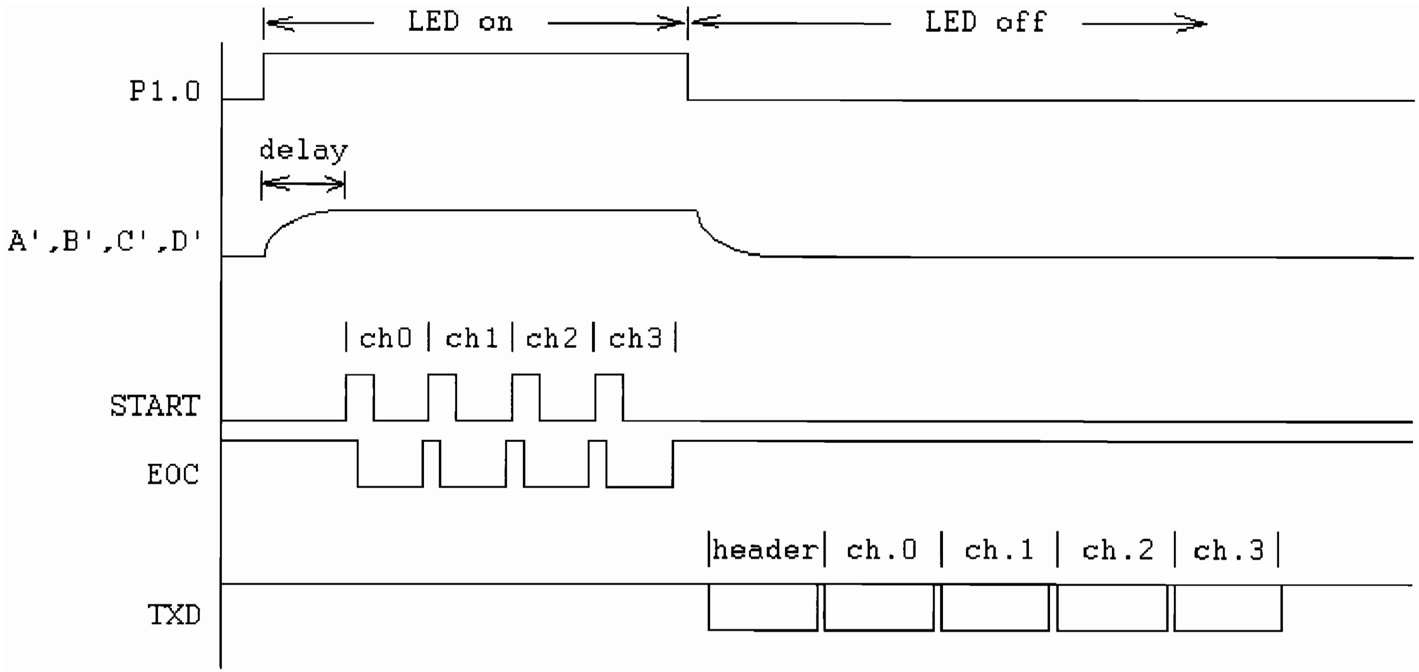

Figure 6.1.1 Timing Diagram

header sent by the 8051 is equal to the command transmitted. If they are equal, it means that at least the communication channel is operational. This may be omitted in the future to speed up the data acquisition.

After the transmission is completed, the system enters the standby mode and waits for a command from the host. When the command is received, the LED is turned on again and the whole process is repeated. Figure 6.1.2 illustrates the sequence of events. The documented and assembled source code is listed in Appendix A.

6.2 Host Programming

The host computer used is an IBM compatible PC with the Intel 80386DX processor running MS-DOS 5.0 at 33 MHz. The functional requirements for the host program are as follows.

(1) get the digitized data from the 8051

(2) perform numerical processing on the data to get the x, y position values

The process begins with sending a command to the 8051 as mentioned in the previous section. The content of the command is the amount of delay needed for the delay loop in Figure 6.1.2; thus, the delay is software controllable.

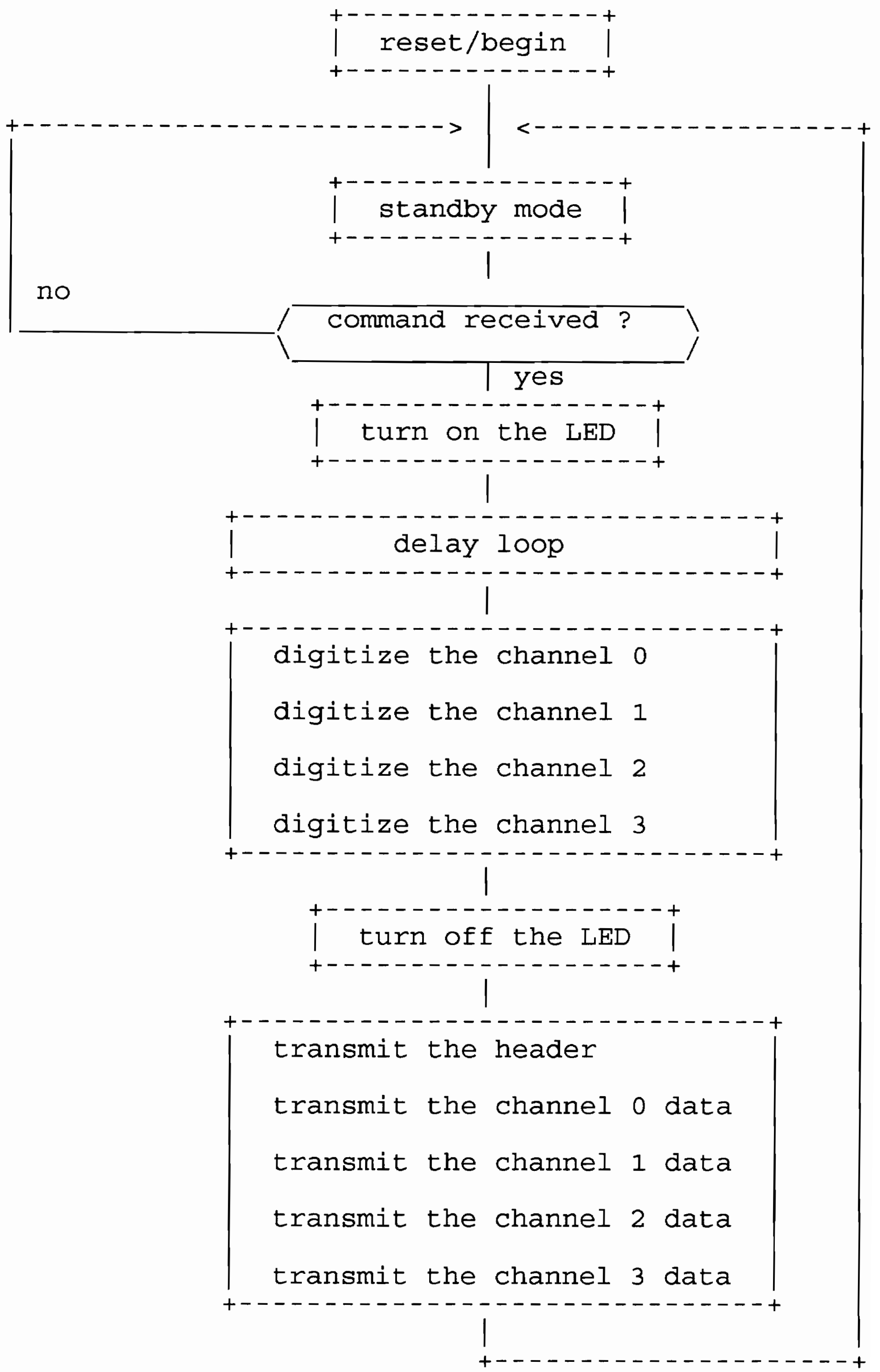

Figure 6.1.2 Flowchart of the 8051 program

From the data received from the 8051, the x, y coordinates can be obtained as follows[5].

$$x = \frac{A' - B'}{A' + B'}$$

$$y = \frac{C' - D'}{C' + D'}$$

x and y range from -1 to +1.

Some additional processing can be done at the host such as averaging to improve the stability of x and y. It was observed that the variation in the raw data is too big to ignore probably due to noises. This and other problems are discussed in Section 7.

The flowchart for the host program is illustrated in Figure 6.2. The documented source code written in C is listed in Appendix B.

```
                +--------------+
                |    begin     |
                +--------------+
                       |
                       |  <-----------------------+
                       |                          |
      +-----------------------------------+       |
      |  send a command to the 8051       |       |
      +-----------------------------------+       |
                       |                          |
      +-----------------------------------+       |
      |  get the data from the 8051       |       |
      +-----------------------------------+       |
                       |                          |
      +-----------------------------------+       |
      |  perform numerical processing     |       |
      +-----------------------------------+       |
                       |                          |
      +-----------------------------------+       |
      |  display/update x, y coordinates  |       |
      +-----------------------------------+       |
                       |                          |
                       +--------------------------+
```

Figure 6.2 Flowchart of the host program

7. Performance Evaluation

7.1 Problems

During the performance testing of the system, a few problems were identified as follows.

(1) accurate calibration of the optic assembly

(2) tuning of the circuit parameters

(3) noises

(4) non-linearity of outputs

Most of the problems are related to the optoelectronics portion of the system. The lens and the photodiode should be placed accurately to reduce the error in x, y reading. The prototype, however, does not fully satisfy this precision requirement.

The transconductance amplifier gain should be set so that the op-amp is not saturated due to strong illumination at short range. One alternative solution is to reduce the brightness of the LED by changing the base resistor in Figure 5.2.2. During the calibration, the LED is placed in the center field of view and the resistor is adjusted to bring the op-amp voltages to 2 V.

Also, the gain should be carefully tuned to maintain the balance between the outputs of the op-amp since the unbalanced outputs may cause incorrect estimation of the coordinates.

The reference voltage of the ADC0809 should be adjusted to cover the input voltage range. This adjustment, the op-amp gain change, and the LED brightness control are necessary elements of the system calibration.

The most serious problem is the noise in the input to the ADC0809 that impedes the accurate digitizing. A few shunt capacitors helped filter out high frequency noises.

In addition, the ADC0809 has inherent conversion error of +/- one least significant bit[6]. The SC-10D also exhibits inherent non-linearity of output[3]. All of these problems affect the resolution and accuracy of the position measurement.

7.2 Results

Approximately 3 mm position change of the LED can be detected over 0.5 m x 0.5 m area at 2 m from the detector (See Illustration 7.2). Very limited x, y reading is possible at up to 3 m distance. Additional numerical

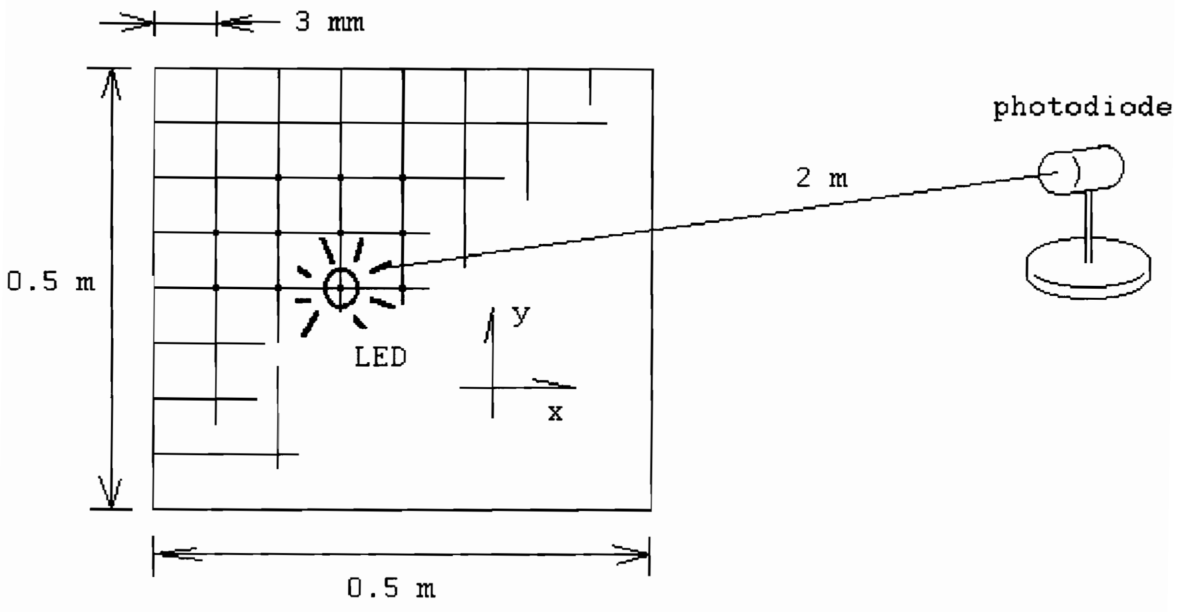

Illustration 7.2 Operating Range and Distance

processing of the raw data improves the stability of x, y reading. The frequency of data acquisition is about 25 Hz to 40 Hz.

8. Conclusion

A 2-D position sensing system is designed and built. The system can measure the position change of the LED at 2 m with 3 mm resolution. The system has the following characteristics:

(1) interface capability with the computer through the RS-232C channel

(2) software control of data acquisition and delay loop

(3) digital processing of photodiode sensor data

(4) use of a synchronous light source

(5) low cost system (See Appendix C)

9. Recommendation and Follow-On

Use of improved optoelectronic circuit is recommended for better accuracy and stability. As a matter of fact, the accuracy of the lens and the filter does not seem to be critical in position measurement because the location of the centroid of a light spot on the photodiode active area gives the hint of x, y coordinates. However, good quality optic system can produce a sharp light spot. Thus, the improvement of resolution would be achieved.

Use of high power infrared LED at 900 - 950 nm is also recommended since the sensitivity of silicon photodiodes is at the maximum in that spectral range.

A follow-on project would put more emphasis on the linear optoelectronics circuit design and also the digital hardware programming to maximize the use of the system resources.

References

[1] P. H. Wendland, "Silicon Photodiodes Come Into Their Own", Optical Spectra, October, 1973.

[2] B. O. Kelly and R. I. Nemhauser, "Techniques for Using the Position Sensitivity of Silicon Photodetectors to Provide Remote Machine Control", United Detector Technology Inc., Santa Monica, California, 1973.

[3] Optoelectronic Components Catalog, UDT Sensors Inc., Hawthorne, California, 1994.

[4] Catalog No. 942, Digi-Key Co., Thief River Falls, Minnesota, 1994.

[5] I. Edwards, "Using Photodetectors for Position Sensing", Sensors, December, 1988.

[6] L. Wakeman, "Using the ADC0808/ADC0809 8-bit µP Compatible A/D Converters with 8-Channel Analog Multiplexer", Application Note 247, Linear Applications Handbook, National Semiconductor Co., Santa Clara, California, 1991.

[7] J. W. Stewart, "The 8051 Microcontroller", Regents/Prentice Hall, Englewood Cliffs, New Jersey, 1993, p. 20.

[8] J. Wilson and J. F. B. Hawkes, "Optoelectronics", Second Edition, Prentice Hall International (UK) Ltd., Cambridge, 1989, p. 280-292.

[9] W. Light, "Non-contact Optical Position Sensing Using Silicon Photodetectors", Application Note #11, United Detector Technology, Hawthorne, California, 1982.

Bibliography

K. J. Ayala, The 8051 Microcontroller: architecture, programming, and applications, West Publishing Company, St. Paul, Minnesota, 1991.

Operational Amplifiers Databook, National Semiconductor Co., Santa Clara, California, 1993.

Embedded Applications, Intel Co., Mt. Prospect, Illinois, 1994.

Embedded Microcontrollers and Processors, Vol. 2, Intel Co., Mt. Prospect, Illinois, 1993.

J. Millman, Microelectronics: digital and analog circuits and systems, McGraw-Hill, Inc., New York, 1979.

A. L. Wyatt, Using Assembly Language, Que Co., Carmel, Indiana, 1987.

W. Feibel, QuickC: the complete reference, Osborne McGraw-Hill, Berkeley, California, 1990.

The TTL Data Book, Vol. 2, Texas Instruments, Dallas, Texas, 1985.

J. Uffenbeck, Microcomputers and Microprocessors: the 8080, 8085, and Z-80 programming, interfacing, and troubleshooting, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1985.

J. Smith, Modern Communication Circuits, McGraw-Hill, Inc., New York, 1986.

Appendix A. The 8051 assembly program

```
0000                 ;****************************************************
0000                 ;    name         : i8051.asm
0000                 ;    date         : April, 1994
0000                 ;    programmer   : Jae H. Cha
0000                 ;    purpose      : the 8051 program for EE 5904 project
0000                 ;****************************************************
0000
0000                 ;---------------------------------------------
0000                 ;    Initialize the interrupt vector table.
0000                 ;---------------------------------------------
0000                         .org    0h        ; power up / reset
0000 020100                  ljmp    main      ; jump to the main program
0003
0003                         .org    03h       ; external interrupt 0
0003 020300                  ljmp    int0sv
0006
000B                         .org    0bh       ; timer 0 interrupt
000B 00                      nop
000C 32                      reti
000D
0013                         .org    013h      ; external interrupt 1
0013 00                      nop
0014 32                      reti
0015
001B                         .org    01bh      ; timer 1 interrupt
001B 00                      nop
001C 32                      reti
001D
0023                         .org    023h      ; serial port interrupt
0023 020200                  ljmp    serisv
0026
0026                 ;---------------------------------------------------
0026                 ;    Initialize the control registers.
0026                 ;---------------------------------------------------
0100                         .org    0100h      ; main program
0100 75A800          main:   mov ie,#0h         ; disable all interrupts
0103 758140                  mov sp,#040h       ; move the stack pointer to 40h
0106
0106 758700                  mov pcon,#0h       ; set smod = 0
0109 758DFD                  mov th1,#0fdh      ; baud rate = 9600, f = 11.0592 MHz
010C 758BFD                  mov tl1,#0fdh
010F 758920                  mov tmod,#020h     ; initialize timer 1 in mode 2
0112 758841                  mov tcon,#041h     ; initialize timer control &
0115                                            ;      intr0 level trigger
0115 759870                  mov scon,#070h     ; set serial mode 1,
0118                                            ;  (10 bits per frame = 8 bit data
0118                                            ;      + 1 start bit + 1 stop bit)
0118 75B800                  mov ip,#0h         ; no priority interrupts
011B 75A891                  mov ie,#091h       ; enable interrupts
011E
011E                 ;-----------------------------------------------------------
011E                 ;    Start the process with the LED light source turned off.
011E                 ;-----------------------------------------------------------
011E C290                    clr p1.0    ; turn off the LED
0120
0120 D202            begin:  setb 02h    ; standby mode
0122 2002FD          rx0:    jb 02h,rx0  ; wait for a command from the host
0125
0125 D290                    setb p1.0   ; turn on the LED
0127
0127 E599                    mov a,sbuf  ; receive the command and
0129 F534                    mov 34h,a   ; store the command in the internal RAM
012B
```

```
012B             ;-----------------------
012B             ;  delay loop
012B             ;-----------------------
012B 14          wait:   dec a
012C 70FD                jnz wait
012E
012E             ;-----------------------------------------------
012E             ;   Begin A/D conversion.
012E             ;-----------------------------------------------
012E D201                setb 01h          ; set the A/D flag
0130 900800              mov dptr,#800h    ; channel 0
0133 F0                  movx @dptr,a      ; start digitizing the channel 0
0134 2001FD      loop1:  jb 01h,loop1      ; wait till done
0137 E0                  movx a,@dptr      ; get the digitized A'
0138 F530                mov 30h,a         ; store A' in the internal RAM
013A
013A D201                setb 01h
013C 900801              mov dptr,#801h    ; channel 1
013F F0                  movx @dptr,a      ; start digitizing the channel 1
0140 2001FD      loop2:  jb 01h,loop2      ; wait till done
0143 E0                  movx a,@dptr      ; get the digitized B'
0144 F531                mov 31h,a         ; store B' in the internal RAM
0146
0146 D201                setb 01h
0148 900802              mov dptr,#802h    ; channel 2
014B F0                  movx @dptr,a      ; start digitizing the channel 2
014C 2001FD      loop3:  jb 01h,loop3      ; wait till done
014F E0                  movx a,@dptr      ; get the digitized C'
0150 F532                mov 32h,a         ; store C' in the internal RAM
0152
0152 D201                setb 01h
0154 900803              mov dptr,#803h    ; channel 3
0157 F0                  movx @dptr,a      ; start digitizing the channel 3
0158 2001FD      loop4:  jb 01h,loop4      ; wait till done
015B
015B C290                clr p1.0          ; turn off the LED
015D
015D E0                  movx a,@dptr      ; get the digitized D'
015E F533                mov 33h,a         ; store D' in the internal RAM
0160
0160             ;-------------------------------------
0160             ;   A/D conversion done.
0160             ;   Begin transmission.
0160             ;-------------------------------------
0160 D200                setb 00h          ; set the user transmission flag
0162 E534                mov a,34h         ; retrieve the command
0164 F599                mov sbuf,a        ; send the command back
0166 2000FD      tx0:    jb 00h,tx0        ; wait till done
0169
0169 D200                setb 00h
016B E530                mov a,30h         ; transmit A'
016D F599                mov sbuf,a        ;
016F 2000FD      tx1:    jb 00h,tx1        ; wait until done
0172
0172 D200                setb 00h
0174 E531                mov a,31h         ; transmit B'
0176 F599                mov sbuf,a        ;
0178 2000FD      tx2:    jb 00h,tx2        ; wait until done
017B
017B D200                setb 00h
017D E532                mov a,32h         ; transmit C'
017F F599                mov sbuf,a        ;
0181 2000FD      tx3:    jb 00h,tx3        ; wait until done
0184
0184 D200                setb 00h
0186 E533                mov a,33h         ; transmit D'
0188 F599                mov sbuf,a        ;
018A 2000FD      tx4:    jb 00h,tx4        ; wait until done
018D
```

```
0000133 018D          ;-------------------------
0000134 018D          ;   transmission done.
0000135 018D          ;   Repeat the process.
0000136 018D          ;-------------------------
0000137 018D 020120          ljmp begin
0000138 0190
0000139 0190          ;-----------------------------------------------------
0000140 0190          ;   interrupt service routine for serial communication
0000141 0190          ;-----------------------------------------------------
0000142 0200                 .org    0200h
0000143 0200 309905   serisv: jnb ti,nxt    ; check the transmission flag
0000144 0203 C299             clr ti        ; if set, clear the flag
0000145 0205 C200             clr 00h       ; clear the user flag for transmission
0000146 0207 32               reti
0000147 0208 C298     nxt:    clr ri        ; if set, clear the flag
0000148 020A C202             clr 02h       ; clear the standby flag
0000149 020C 32               reti
0000150 020D
0000151 020D          ;----------------------------------------------
0000152 020D          ;   interrupt service routine for Interrupt 0
0000153 020D          ;-----------------------------------------------
0000154 0300                 .org    0300h
0000155 0300 C201     int0sv: clr 01h       ; clear the A/D flag
0000156 0302 32               reti
0000157 0303
0000158 0303                 .end
```

AC	=00D6	IP.1	=00B9	PT1	=00BB	TX2	=0178
ACC	=00E0	IP.2	=00BA	PT2	=00BD	TX3	=0181
ACC.0	=00E0	IP.3	=00BB	PX0	=00B8	TX4	=018A
ACC.1	=00E1	IP.4	=00BC	PX1	=00BA	TXD	=00B1
ACC.2	=00E2	IP.5	=00BD	RB8	=009A	WAIT	=012B
ACC.3	=00E3	IP.6	=00BE	RCAP2H	=00CB		
ACC.4	=00E4	IP.7	=00BF	RCAP2L	=00CA		
ACC.5	=00E5	IT0	=0088	RCLK	=00CD		
ACC.6	=00E6	IT1	=008A	REN	=009C		
ACC.7	=00E7	LOOP1	=0134	RI	=0098		
B	=00F0	LOOP2	=0140	RS0	=00D3		
B.0	=00F0	LOOP3	=014C	RS1	=00D4		
B.1	=00F1	LOOP4	=0158	RX0	=0122		
B.2	=00F2	MAIN	=0100	RXD	=00B0		
B.3	=00F3	NXT	=0208	SBUF	=0099		
B.4	=00F4	OV	=00D2	SCON	=0098		
B.5	=00F5	P	=00D0	SCON.0	=0098		
B.6	=00F6	P0	=0080	SCON.1	=0099		
B.7	=00F7	P0.0	=0080	SCON.2	=009A		
BEGIN	=0120	P0.1	=0081	SCON.3	=009B		
CCAPM0	=00DA	P0.2	=0082	SCON.4	=009C		
CCAPM1	=00DB	P0.3	=0083	SCON.5	=009D		
CCAPM2	=00DC	P0.4	=0084	SCON.6	=009E		
CCAPM3	=00DD	P0.5	=0085	SCON.7	=009F		
CCAPM4	=00DE	P0.6	=0086	SERISV	=0200		
CCON	=00D8	P0.7	=0087	SM0	=009F		
CCON.0	=00D8	P1	=0090	SM1	=009E		
CCON.1	=00D9	P1.0	=0090	SM2	=009D		
CCON.2	=00DA	P1.1	=0091	SP	=0081		
CCON.3	=00DB	P1.2	=0092	T2CON	=00C8		
CCON.4	=00DC	P1.3	=0093	T2CON.0	=00C8		
CCON.5	=00DD	P1.4	=0094	T2CON.1	=00C9		
CCON.6	=00DE	P1.5	=0095	T2CON.2	=00CA		
CCON.7	=00DF	P1.6	=0096	T2CON.3	=00CB		
CMOD	=00D9	P1.7	=0097	T2CON.4	=00CC		
CPRL2	=00C8	P2	=00A0	T2CON.5	=00CD		
CT2	=00C9	P2.0	=00A0	T2CON.6	=00CE		
CY	=00D7	P2.1	=00A1	T2CON.7	=00CF		
DPH	=0083	P2.2	=00A2	TB8	=009B		
DPL	=0082	P2.3	=00A3	TCLK	=00CC		
EA	=00AF	P2.4	=00A4	TCON	=0088		
ES	=00AC	P2.5	=00A5	TCON.0	=0088		
ET0	=00A9	P2.6	=00A6	TCON.1	=0089		

ET1	=00AB	P2.7	=00A7	TCON.2	=008A
ET2	=00AD	P3	=00B0	TCON.3	=008B
EX0	=00A8	P3.0	=00B0	TCON.4	=008C
EX1	=00AA	P3.1	=00B1	TCON.5	=008D
EXEN2	=00CB	P3.2	=00B2	TCON.6	=008E
EXF2	=00CE	P3.3	=00B3	TCON.7	=008F
F0	=00D5	P3.4	=00B4	TF0	=008D
IE	=00A8	P3.5	=00B5	TF1	=008F
IE.0	=00A8	P3.6	=00B6	TF2	=00CF
IE.1	=00A9	P3.7	=00B7	TH0	=008C
IE.2	=00AA	PCON	=0087	TH1	=008D
IE.3	=00AB	PS	=00BC	TH2	=00CD
IE.4	=00AC	PSW	=00D0	TI	=0099
IE.5	=00AD	PSW.0	=00D0	TL0	=008A
IE.7	=00AF	PSW.1	=00D1	TL1	=008B
IE0	=0089	PSW.2	=00D2	TL2	=00CC
IE1	=008B	PSW.3	=00D3	TMOD	=0089
INT0	=00B2	PSW.4	=00D4	TR0	=008C
INT0SV	=0300	PSW.5	=00D5	TR1	=008E
INT1	=00B3	PSW.6	=00D6	TR2	=00CA
IP	=00B8	PSW.7	=00D7	TX0	=0166
IP.0	=00B8	PT0	=00B9	TX1	=016F

Appendix B. The Host Program

```
/*----------------------------------------------------
name        : host.c
date        : April, 1994
programmer  : Jae H. Cha
purpose     : main program for EE5904 project
compiler    : QuickC 2.5
----------------------------------------------------*/

#define REP 1           /* number of replication */
#define DELTA 1         /* tolerance             */

#include <dos.h>
#include <process.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

main()
{
        union REGS inreg, outreg;
        int i,j, intnum;
        float xcor, ycor, xsum, ysum;
        float oldsample[5], newsample[5];

        /*-------------------------------------------------
                Initialize the serial communication port
        -------------------------------------------------*/
        inreg.h.al = 0xe3; /* 9600 bps, no parity, 1 stop bit, 8 bit data      */
        inreg.x.dx = 0;    /* set com1:                                        */
        inreg.h.ah = 0x0;  /* service 0 : to initialize the port              */
        intnum = 0x14;     /* BIOS interrupt service number for communications */
                           /* services                                         */
        int86(intnum,&inreg,&outreg);

        for (i = 0; i < 5; i++) oldsample[i] = 0;
  while (1)
        {
         for (i = 0; i < 5; i++)
                { newsample[i] = 0; }

        /*--------------------------------------------------------------
                get the data from the 8051
                if necessary, take multiple samples and average them
        --------------------------------------------------------------*/
         for (j = 0; j < REP; j++)
                {
                 inreg.h.al = 0x48;      /* transmit a command              */
                 inreg.x.dx = 0;         /* set COM1:                       */
                 inreg.h.ah = 1;         /* service 1 : transmit a character */
                 int86(0x14,&inreg,&outreg);

                 inreg.x.dx = 0;          /* set COM1:                       */
                 inreg.h.ah = 0x2;        /* service 2 : receive a character */
                 for (i = 0; i < 5; i++)  /* receive five bytes              */
                        {
                                int86(0x14,&inreg,&outreg);
                                newsample[i] += (float)outreg.h.al;
                        }
                }

                /* take an average if multiple samples had been taken */
                for (i = 0; i < 5; i++)
                        { newsample[i] = newsample[i] / REP; }
```

```
/* discard the new sample if it is within the tolerance range */
for (i = 1; i < 5; i++)
  {
   oldsample[i] = (abs((int)(oldsample[i] - newsample[i])) > DELTA) \
                       ? newsample[i] : oldsample[i];
  }

/*--------------------------------------
       estimate x, y coordinates
--------------------------------------*/
xsum = oldsample[1] + oldsample[2];
ysum = oldsample[3] + oldsample[4];
if (xsum != 0.0)
                      xcor = (oldsample[1] - oldsample[2]) / xsum;
               else
                      xcor = 0.0;
if (ysum != 0.0)
                      ycor = (oldsample[3] - oldsample[4]) / ysum;
               else
                      ycor = 0.0;

/*----------------------------------------
       display/update x, y reading
----------------------------------------*/
system("cls");
fprintf(stdout,"\n\n\n\n\n(x,y) = (%2d,%2d)\n\n",
               (int)(xcor*10),(int)(ycor*10));
fprintf(stdout,"---------- Raw Data -----------------------\n");
fprintf(stdout,"delay= %3.0f\n\n",oldsample[0]);
fprintf(stdout,"A= %5.0f, B= %5.0f, C= %5.0f, D= %5.0f\n\n",
               oldsample[1],oldsample[2],oldsample[3],oldsample[4]);
fprintf(stdout,"x= %5.2f, y= %5.2f\n",xcor, ycor);
   }
}
```

Appendix C. Parts List and Cost Summary

item	description	unit price
SC-10D	position sensing photodiode	94.50
lens	27/46 double convex lens	17.00
filter	600-700 nm bandpass filter	15.75
LM324N	low power quad op-amp	.26
TIP102	darlington power transistor	.39
LN261CAL(UR)	665 nm high brightness LED	.93
8051	8 bit microcontroller	2.49
74S373	octal D type latch	.33
27C64	8192 x 8 EPROM	3.49
ADC0809	8 bit A/D converter	3.49
MAX232N	dual RS232 Rx/Tx	3.99
74AHCT04	hex inverter	.19
74AHCT08	quad 2 input AND gate	.19
crystal	11.0592 MHz cylinder-type xtal.	1.35
misc.	resistor, capacitor, sockets, board, hardware, wire, etc.	35.65
total		180.00

Appendix D. The SC-10D Photodiode Electrical Specifications[3]

model # -- SC-10D
manufacturer ----------- the United Detector Technology Inc.
overall active surface: area ---------------------- 103 mm^2
dimensions in inches ---- 0.4 x 0.4
typical responsivity @ 632.8nm -------------------- 0.35 A/W
typical uniformity of response with 1 mm spot ------ +/- 2 %
typical position linearity: central 25 % --------------- 5 %
central 75 % -------------- 20 %
typical dark current @ 10 V ----------------------- 0.15 μA
typical source resistance -------------------------- 5 MΩ
minimum breakdown voltage --------------------------- 25 V
capacitance @ 0 V --------------------------------- 1200 pF
@ 10 V ---------------------------------- 260 pF
@ 25 V ---------------------------------- 166 pF
typical rise time (10% - 90%) w/R_L = 50Ω, 10 V -- 2000 nsec
NEP @ 10 V ---------------------------------- 10^{-11} $W/Hz^{1/2}$
positioning resistance ---------------------------- 4 KΩ
temperature range: operating -------------------- -25/+80 °C
storage -------------------- -55/+100 °C

Appendix E. The LN261CAL(UR) Light Emitting Diode Electrical Specifications[4]

part # ------------------------ LN261CAL(UR)
manufacturer ------------------ Panasonic

electro-optical characteristics (T_a = 25 °C)

- I_o @ 5 mA ------- 470 mcd
- P_D --------------- 70 mW
- V_R ---------------- 3 V
- I_F --------------- 30 mA
- I_{oMAX} ---------- 2000 mcd
- λ_p -------------- 665 nm

Appendix F. Analytical design of Sec. 4

F.1 Optics

Let the focal length be f and the detector size d. Then the field of view (FOV) is

$$FOV = 2\ \tan^{-1}(d/2f) \qquad \text{(Fig. F.1)}$$

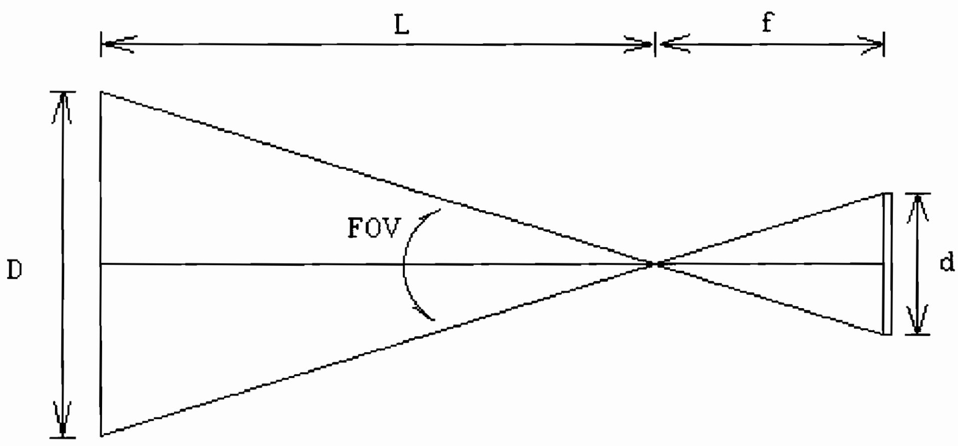

Fig. F.1 Field of View

According to the manufacturer's specification sheet for the SC-10D, the central 75 % of the detector can be used allowing up to 20 % error in linearity. For L = 2 m, f = 46 mm, d = 10 mm, the operating range D is

$$D = L \times 0.75\ d\ /\ f$$

$$= 2\ m \times 0.75 \times 10\ mm\ /\ 46\ mm = 326\ mm = 13\ in$$

$$FOV = 2\ \tan^{-1}(0.75\ d\ /\ 2\ f)$$

$$= 2\ \tan^{-1}(0.75 \times 10\ /\ 2 \times 46) = 9\ degrees$$

F.2 Signal power requirement

The output current from the lateral effect photodiode is

$I_A = I_\lambda\ (1 - k/d)$ [9] (Fig. F.2.1)

and $I_B = I_\lambda\ (k/d)$,

where I_λ is the photoinduced current given by

$I_\lambda = R_\lambda \times P_i$ [3] (R_λ = responsivity of the photodiode, P_i = incident signal power)

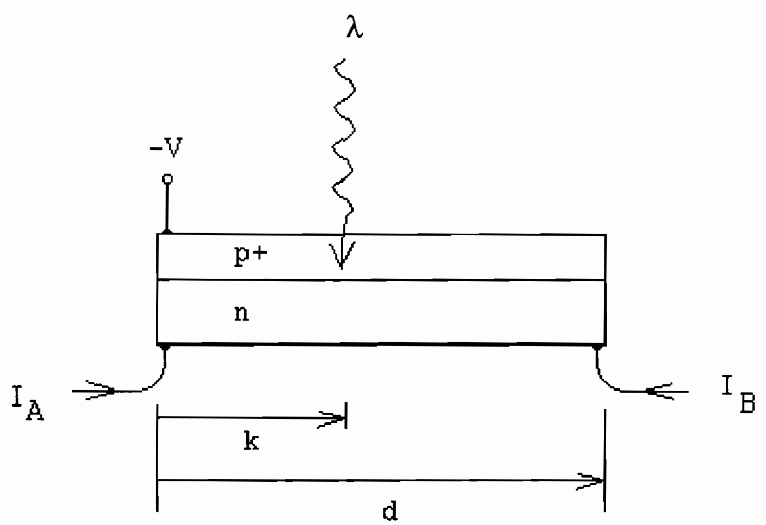

Fig. F.2.1

When the light spot falls on the center of the photodiode,

$I_A = I_B = I_\lambda / 2$

Assuming R_λ = 0.4 A/W at 660 nm, R_f = 1 MΩ and $V_A = V_B$ = 2V (See Fig. F.2.2), the incident signal power P_i is

$P_i = 2V_A / R_f R_\lambda$

$= 2 \times 2V / 1\ M\Omega \times 0.4\ A/W = 10\ \mu W$

Thus, the light source should be able to deliver at least 10 μW incident signal power to the photodiode at the operating distance L.

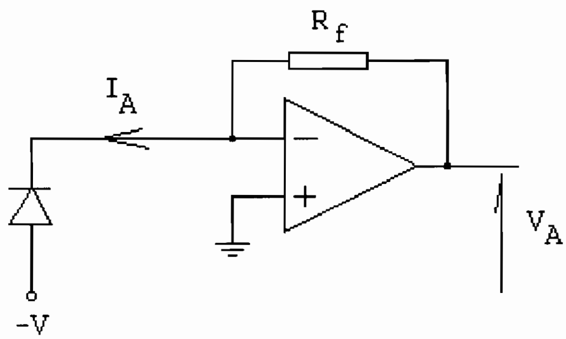

Fig. F.2.2

F.3 Noise analysis

Detailed noise analysis is beyond the scope of this project and subject to the follow-on project.