Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

New Jersey's Science & Technology University

The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.

ABSTRACT

An Embedded Controller-Based Fiber Optic Sensor For Use in Fresh Concrete

by
Zhaowu Zhang

The principle of refraction and reflection of light in optical fibers is applied to detection of air bubbles in a previous thesis. In the earlier work, a microcomputer was employed in acquisition and processing of air bubble data. In the present thesis, an embedded controller system is interfaced and programmed in order to replace the microcomputer. In this way, the system can be more easily integrated into field construction operations. Reflected light intensity signals are detected and analyzed by a microcontroller. The entrained percentage of air in concrete is displayed immediately. The signal can be transferred through a UART (Universal Asynchronous Receiver-Transmitter) to a master IBM compatible personal computer. Analysis of reflected light intensity produces a graphic measurement of the air content of fresh concrete.

Accuracy of the system is evaluated through extensive laboratory and field trials.

AN EMBEDDED CONTROLLER-BASED FIBER OPTIC SENSOR FOR USE IN FRESH CONCRETE

by

Zhaowu Zhang

A Thesis
Submitted to the Faculty of New Jersey
Institute of Technology in
Partial Fulfillment of the Requirements for the Degree
of Master of Science
Department of Electrical and Computer Engineering
May 1992

APPROVAL PAGE
An Embedded
Controller-Based Fiber
Optic Sensor for Use in Fresh Concrete

by
Zhaowu Zhang

May 6, 1992

Dr. Farhad Ansari, Thesis Adviser
Professor of Department of Civil Engineering, NJIT

May 6 /92

Dr. Sol Rosenstark, Committee Member
Professor of Department of Electrical and Computer Engineering, NJIT

May 6, 1992

Dr. Edip Niver, Committee Member
Associate Professor of Department of Electrical and Computer Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Zhaowu Zhang

Degree: Master of Science in Electrical and Computer Engineering

Date: May, 1992

Date of Birth:

Place of Birth:

Undergraduate and Graduate Education:

- Master of Science in Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, 1992
- Bachelor of Science in Electrical and Computer Engineering, Shanghai Jiao Tong University, Shanghai, P. R. of China, 1982

Major: Electrical and Computer Engineering

This thesis is dedicated to
my mother and to my wife

ACKNOWLEDGEMENTS

I would like to express my gratitude and thanks to my thesis advisor, Professor Farhad Ansari, for his selection of the thesis topic, guidance and encouragement at all stages of the work.

Many thanks are due to the distinguished members of my committee: Dr. Sol Rosenstark, Dr Edip Niver. Their suggestions and stimulating discussions have considerably improved the quality of the thesis.

Last but not the least, I would like to thank my family. Many thanks also go to my mother and to my wife, for their deep love and encouragement.

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Air Content in Concrete

Quality control and condition analysis through non-destructive testing in concrete is one of the important problem in real life application. In the placement of cast-in-place concrete, fresh concrete must be tested to ensure that it meets contractual specifications. Among these specifications is the entrained air percentage. The percentage of entrained air in concrete must be well controlled. Since the freezing and thawing durability would be impaired if the concrete containes an insufficient amount of air, and the strength would be unnecessarily reduced if the percentage of air becomes excessive.

There are many factors influencing the amount of entrained air [1] :

- The higher the slump of the concrete, the higher the air content, until high slumps are reached, when the air content drops off some what.
- The greater the proportion of sand in the total aggregate, the greater the air content of the concrete.
- The presence of finely divided materials causes a reduction in the air content of concrete.
- A higher temperature of the concrete results in a lower air content, and vice versa.
- Cements with a high alkali content air more easily than do low alkali cements.

- A higher mixing speed gives a higher air content than does a lower one.
- Vibration reduces the air content of concrete.
- The more air entraining agent that is need in the mixture, the higher will be the air content.

Conventional methods for measuring air content in freshly mixed concrete are :

- the pressure method
- the volumetric method
- the gravimetric method

Some of the traditional devices used to determine content are : Precision air meter (pressure method) and Roll-A-Meter (volumetric method). These devices are cumbersome to use properly , time consuming and of questionable accuracy.

1.2 Measuring Methodology

As explained in an earlier thesis [6], when a ray of light impinges on the interface separating two different media, a portion of the incident flux density will be diverted back in the form of reflected light, while the remainder will be transmitted across the boundary as refracted light. (Fig. 1.1.)

The refracted light also lies in the plane of incidence and on the opposite side of the normal. This relationship, experimentally established by Snell, is known as Snell's law. [2]

$$\frac{\sin(\theta_1)}{\sin(\theta_2)} = \frac{n_2}{n_1} \tag{1.1}$$

where θ_1 is the angle of incidence, θ_2 is the angle of refraction. n_1, n_2 are the refractive indices of the two different media.

Furthermore, according to Fresnel's laws [3] of reflection, the ratio of the amplitudes of the reflected and incident waves can be approximately expressed as

$$\frac{R}{E} = \frac{n_2 - n_1}{n_2 + n_1} \tag{1.2}$$

where the symbols E and R are amplitudes of the electric vectors in the incident and reflected light, respectively.

The intensity varies as the square of the amplitude, hence the reflection at normal incidence is

$$\frac{R^2}{E^2} = \left(\frac{n_2 - n_1}{n_2 + n_1} \right)^2 \tag{1.3}$$

The simple concept of light intensity loss as a result of refractive index difference between two media when coupled with fiber optics will produce in a practical measurement instrument.

As shown in Fig. 1.2 , if the light exiting an optical fiber (N = 1.44) meets an air bubble (N =1.0003), most of the light will reflect back due to the large contract between the indexes of these two mediums.

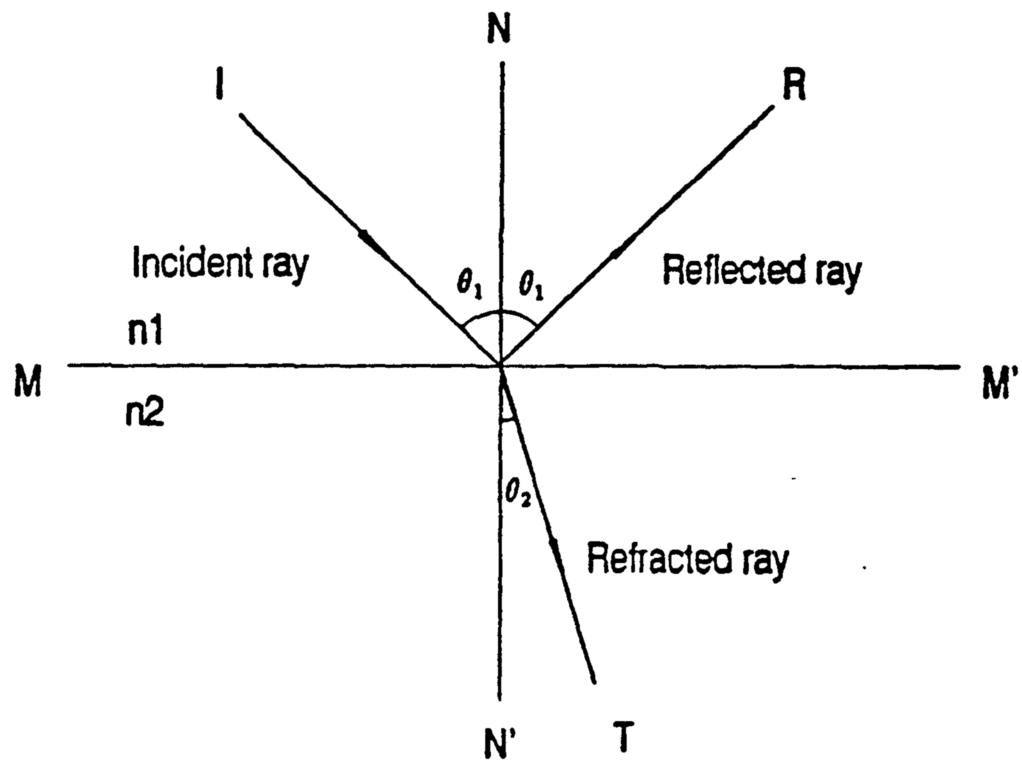


Figure 1.1 Reflection and refraction at the boundary separating two media with refractive n1 and n2, respectively.

On the other hand, the same fiber tip, when immersed in fresh concrete, exhibits a large decrease in the intensity of reflected light. Because of the matched index between glass and concrete, the light is no longer totally reflected and most of the optical power is transmitted from the fiber terminal into the concrete.

1.3 System Structure and Basic Components

The block diagram in Fig. 1.3 describes the components of the fiber optic sensor system developed earlier [6].

A diode-laser emitting visible light at 670 nm wavelength is employed for generating an optical signal. A silica glass multimode optical fiber delivers the light to a coupler; the coupler directs the light to the exit terminal of the fiber optical sensor. Depending on the medium (air bubble or concrete), the more air content is in the concrete, the more light reflectes. The coupler separates the reflected and transmitted signals and transfers the reflected signal to a photodetector. This signal, which is called the air bubble signal, is converted to an electrical current by the photodetector and amplified by an integrated circuit (IC) amplifier. The air bubble signal is transferred to a microcontroller through A/D converter. Once the real time data is processed, the controller displays the air percent on liquid crystal display (LCD). At the completion of test,

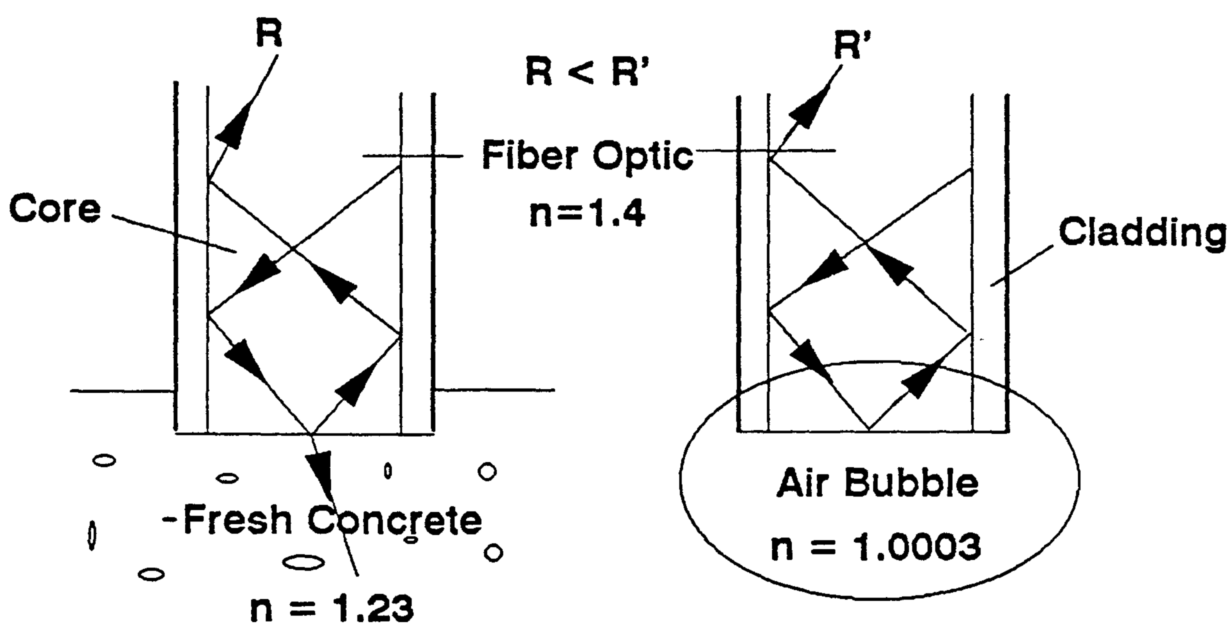


Figure 1.2 Reflected light intensity at the tip of a multimode fiber optic sensor in fresh concrete and air bubble.

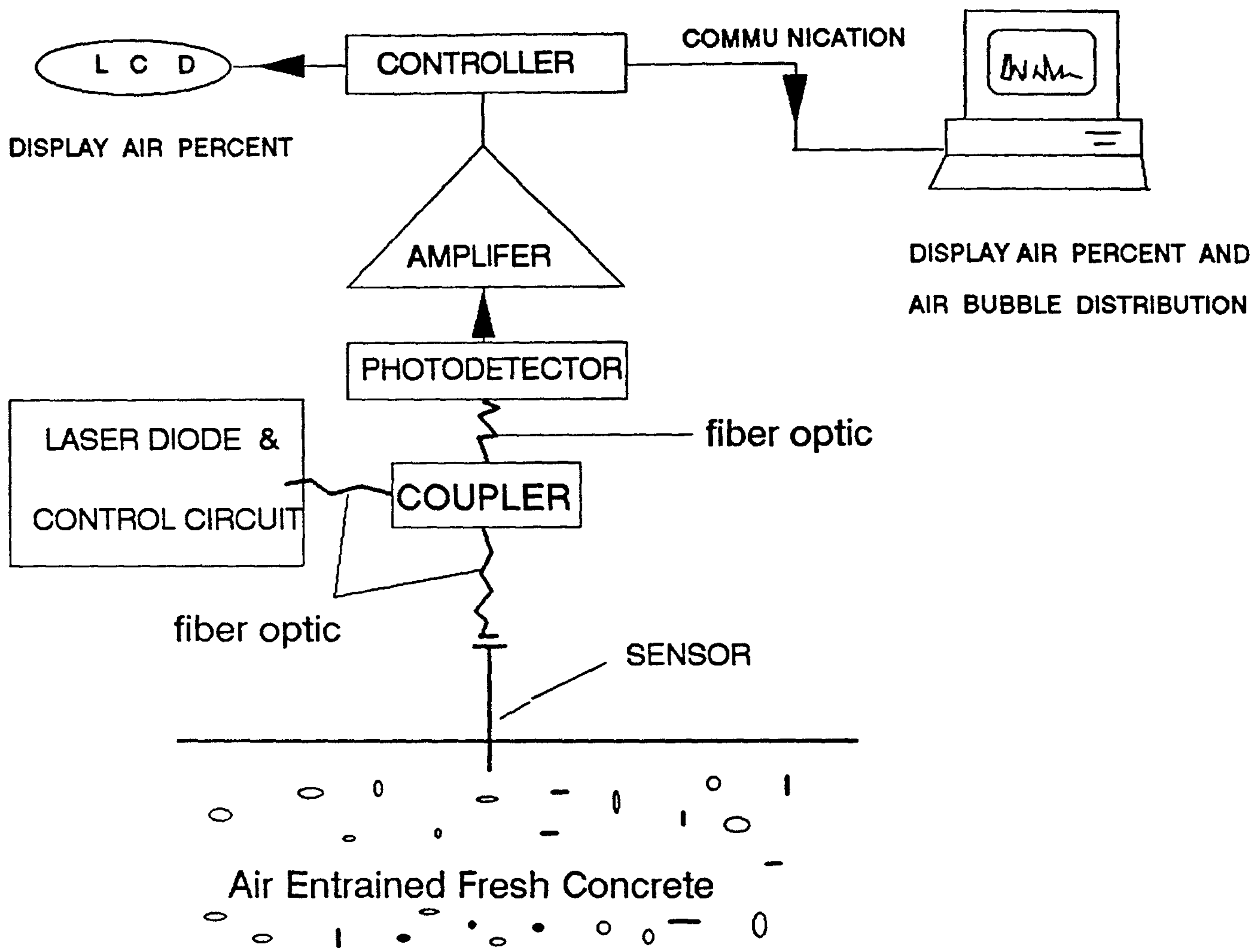


Figure 1.3 System Block Diagram.

the controller can transfer the data to an IBM PC. Computer processes the data , displays the graphic distribution of air, prints the result and stores the data to a floppy disk.

1.4 Fiber Optic Sensor

Fiber optic sensor development has matured to the point where the impact of this new technology is now evident. There are many advantages in using fiber optic sensors [4].

- good electrical isolation
- immunity to electromagnetic interference
- safety in explosive environments
- low signal attenuation
- compactness and lightness
- flexibility

Furthermore, fiber optic sensors exhibit high sensitivity and can be used in contact measurements.

The fiber optic sensor used in this system belongs to the fiber optic intensity sensor category. A multimode fiber optic cable (core/cladding, 50/125 um) is used to prepare the sensor. Extreme care should be involved in the preparation of the sensor [6]. The structure of the sensor is shown in Fig. 1.4 . The main purpose for using a needle shaped sensor head was to enhance response speed (reduce welting effect), and to protect the sensor tip.

This design uses only one fiber optic for bidirectional light transmission [7]. A guided light, emitted by a laser diode , is allowed to reflect from the fiber optic sensor tip (interface) that is modulated by the varying index of air, concrete and water. A light intensity portion is reflected back, depending on the medium (air, concrete or water) present at the fiber tip and is detected by a photodetector after separation from emitted light in the coupler as shown in Fig. 1.5 . Fig. 1.6 shows different analog output voltage values that represent reflected light intensity in air, concrete and water. It shows that this sensor can detect the index difference between different mediums effectively. In general, two parameters define sensor performance : (1) Relative response level in air and fresh concrete. (2) Response time. Acceptable relative response level in air and fresh concrete should be larger than 0.7 volts. Fig. 1.7 shows the relative response level in air and concrete measured by two different sensors; the large and small relative response levels are compared. In this system, the fiber optic sensor response time is defined as the slope of the reflected intensity response curve from water to air (Fig. 1.8). It has been found that a good fiber optic sensor should have a response time of less than 0.2 second. Fig. 1.8 compares the response time between a fast and a slow sensor. The latter is not so sensitive and therefore it will not be able to detect all the phenomena effectively. According

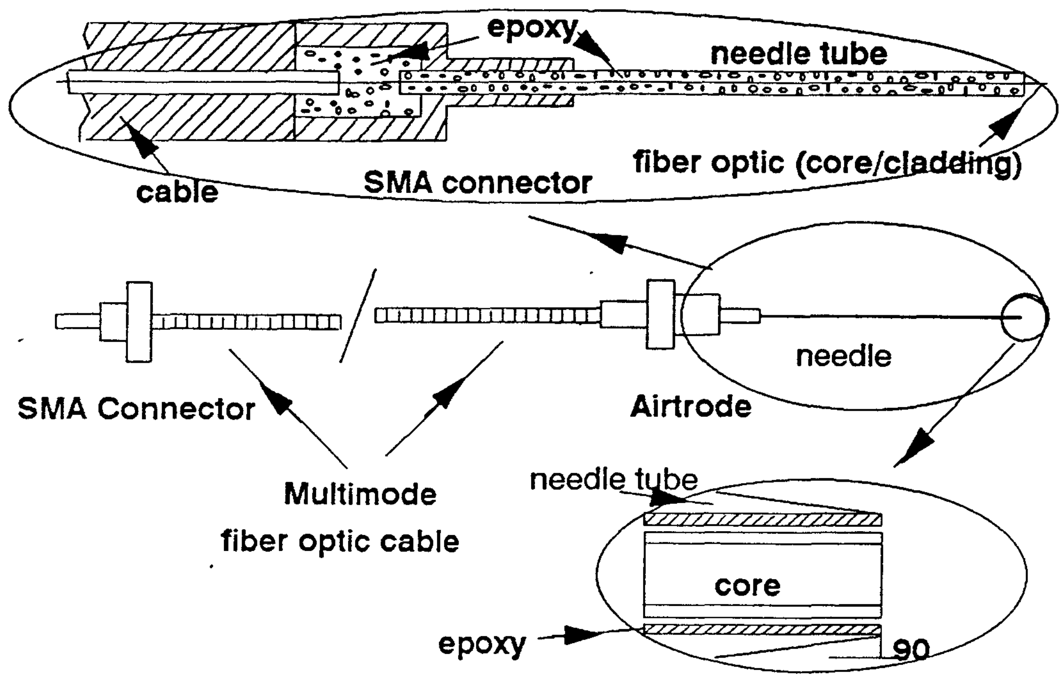


Figure 1.4 Fiber Optic Sensor Structure.

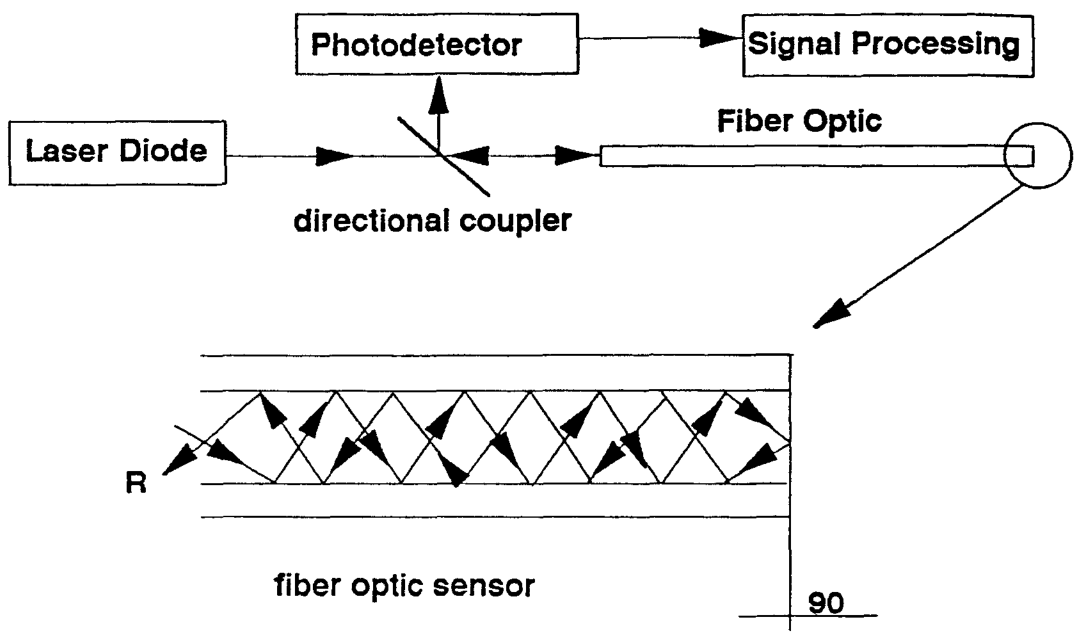


Figure 1.5 Principle of System Fiber Optic Sensor.

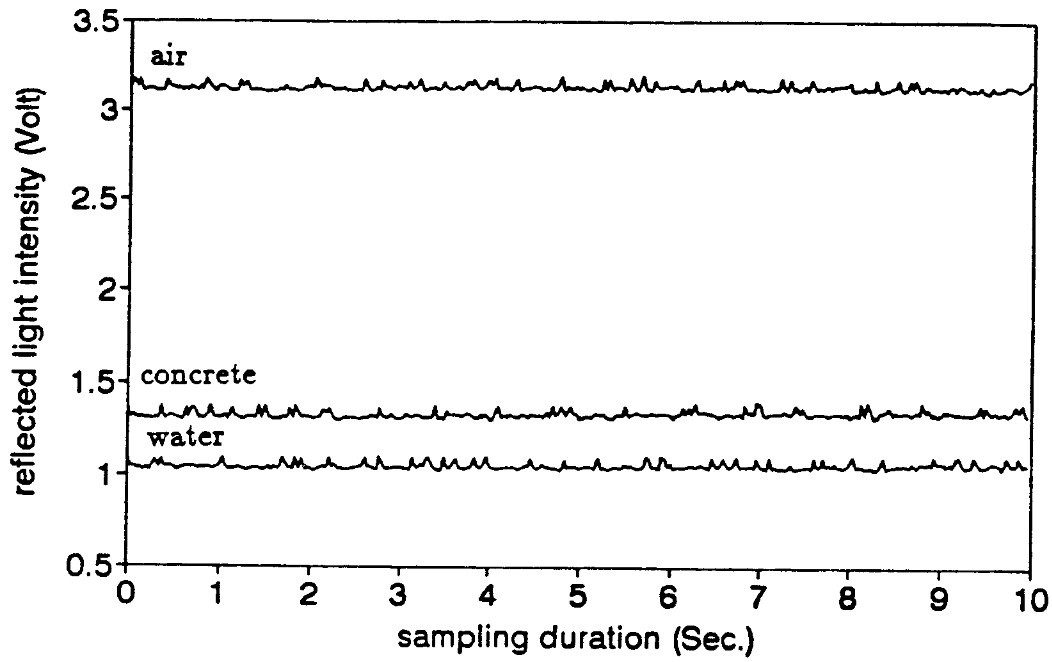


Figure 1.6 Reflected light intensity in air, concrete and water.

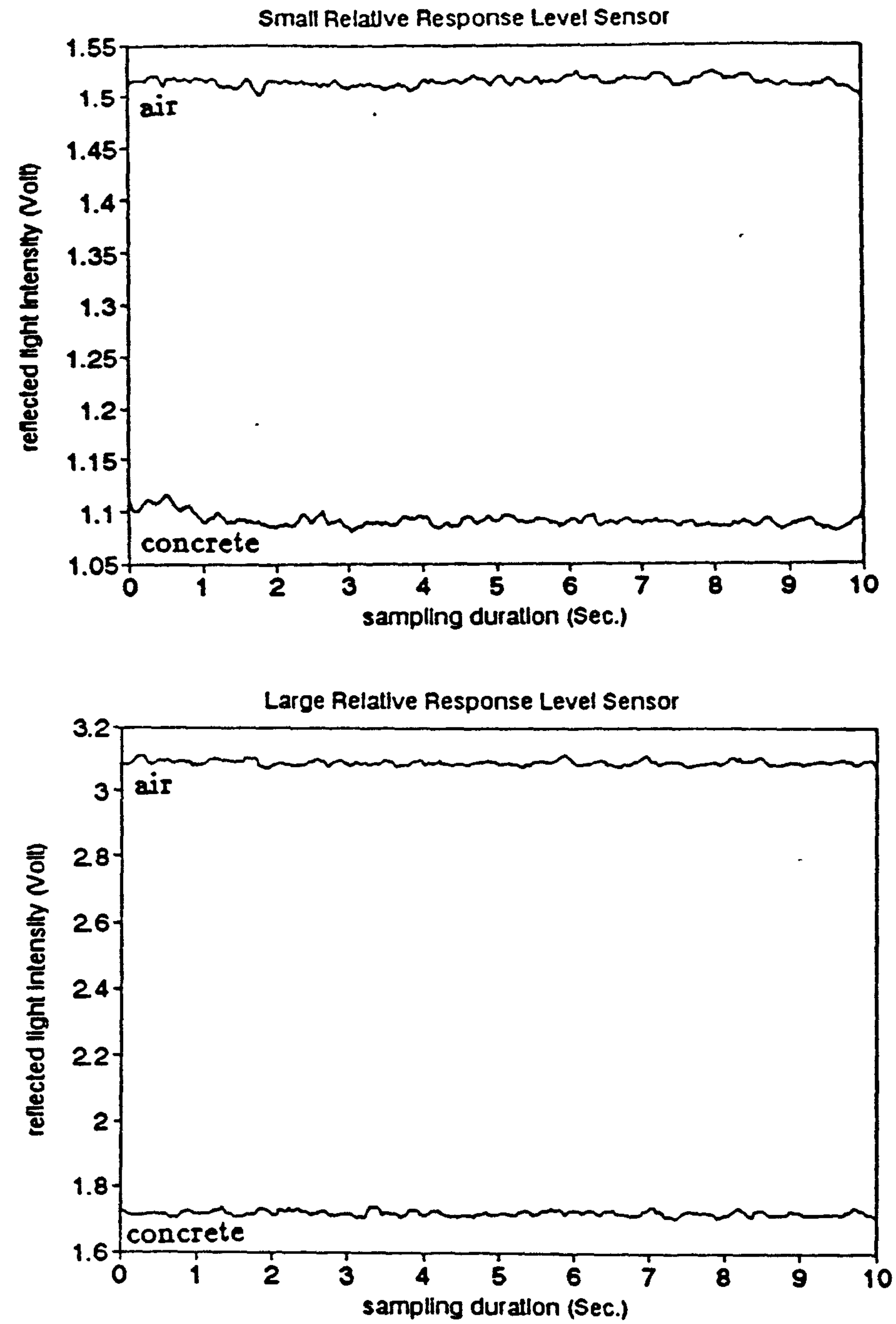


Figure 1.7 Fiber optic sensor relative response level in air and fresh concrete.

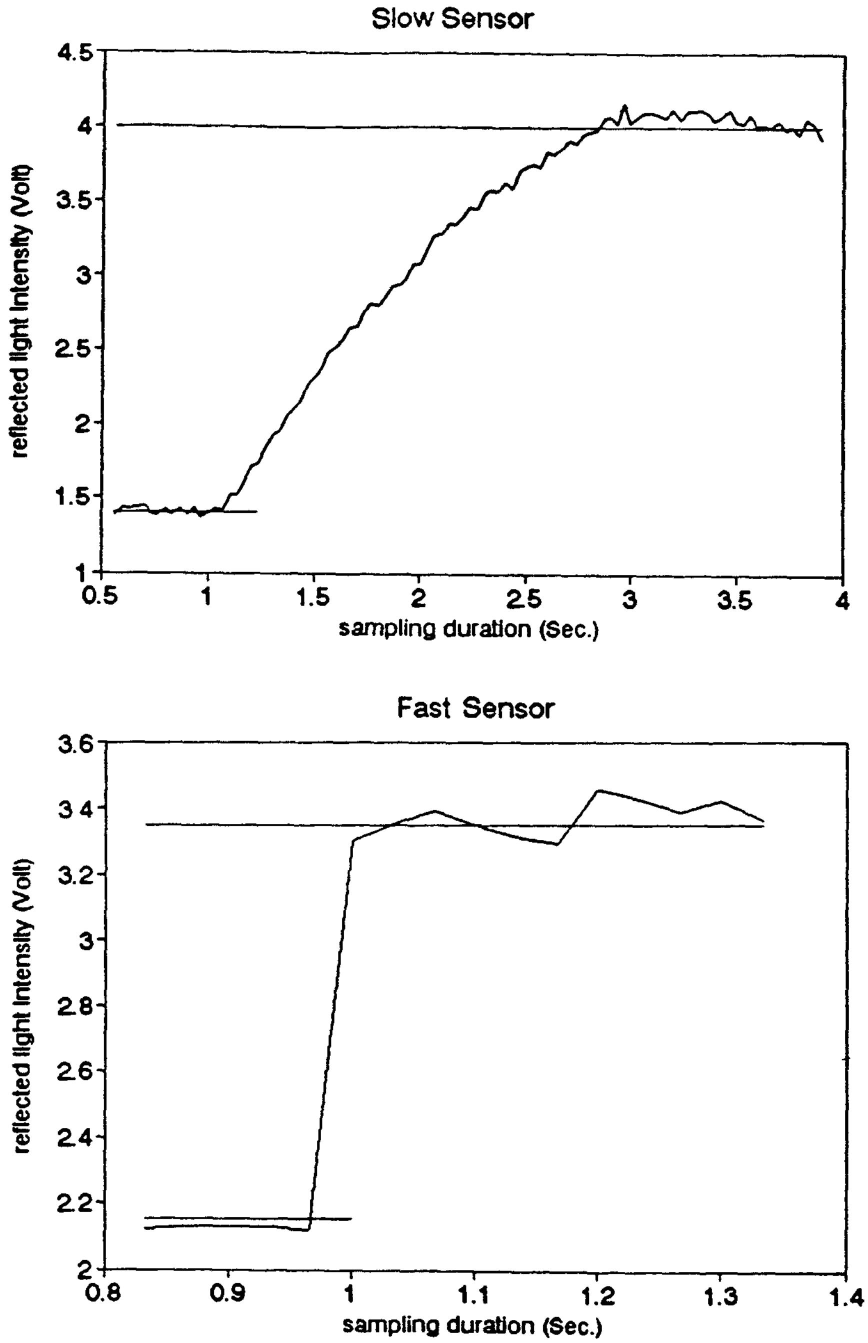


Figure 1.8 Fiber optic sensor response speed.

to the present studies, it was found that the sensor speed depends on the probe head polishing procedures and the welting effect. In Fig. 1.8 slope of the fast sensor is 13.4 volts/sec.

1.5 Optical Source —— Laser Diode

Fiber optic systems can use every conceivable light source from light emitting diode to semiconductor laser diode or high - power laser. In this system, a visible laser diode emitting coherent light at a wavelength of 670 nm [8] is employed. It has advantages of small size, ability to operate on DC power and low input power.

1.6 Multimode Fiber Directional Coupler

The directional coupler used in this system is the multimode three - port PC3 - C - 50 coupler (CANSTAR). It provides a high directivity (-25 DB), low -loss beam- splitting for a fiber system operating at 400 to 1600 nano - meters and a convenient beam splitting with negligible temperature and polarization dependence. The beam splitting ratio is 50 percent . It serves as an alignment-free beam splitter, power combiner and energy tap.

1.7 Laser to Fiber Source Coupler

The laser source coupler is used to overcome the problem of precision alignment while maintaining low losses in the intensity output. In this system, a multimode laser to fiber source coupler (OZ Optics Ltd) is used. This source coupler utilizes the properties of lenses in combination with a novel tilt method to achieve submicron resolution. This method is based on precision control of the angle between the laser beam and receiver. The operating principle is shown in Appendix C.

CHAPTER 2

OPTICAL DATA SIGNAL PROCESSING

Determination of percent air content in fresh concrete requires calibration and signal processing of an optical data. This chapter presents a brief discussion of digital signal processing techniques for processing the acquired optical data.

2.1 Signal Analysis

2.1.1 Signal Analysis

The signal flow chart of the system is shown in Fig. 2.1. Reflected light is transformed to an electrical signal by a photodetector. Following the amplification, analog signal A(t) is sent to A/D converter channel 0. Synchronously, the same data is filtered, it's through a low-pass filter and sent to channel number one of the A/D converter. In this way, channels 0 and 1 contain the test and filtered data respectively. Discrete time signals D0(n) and D1(n) are obtained from sampling the analog signals A(t) and A1(t); then

$$D0(n) = A(nT) \; ; \; D1(n) = A1(nT)$$

where T is the sampling period and n is the sampling point number. The test signal D0(n) is displayed in Fig. 2.2 . It contains small and large peaks corresponding to small and large air bubbles in fresh concrete. The filtered data D1(n) is shown in Fig. 2.3 . It represents the baseline for fresh

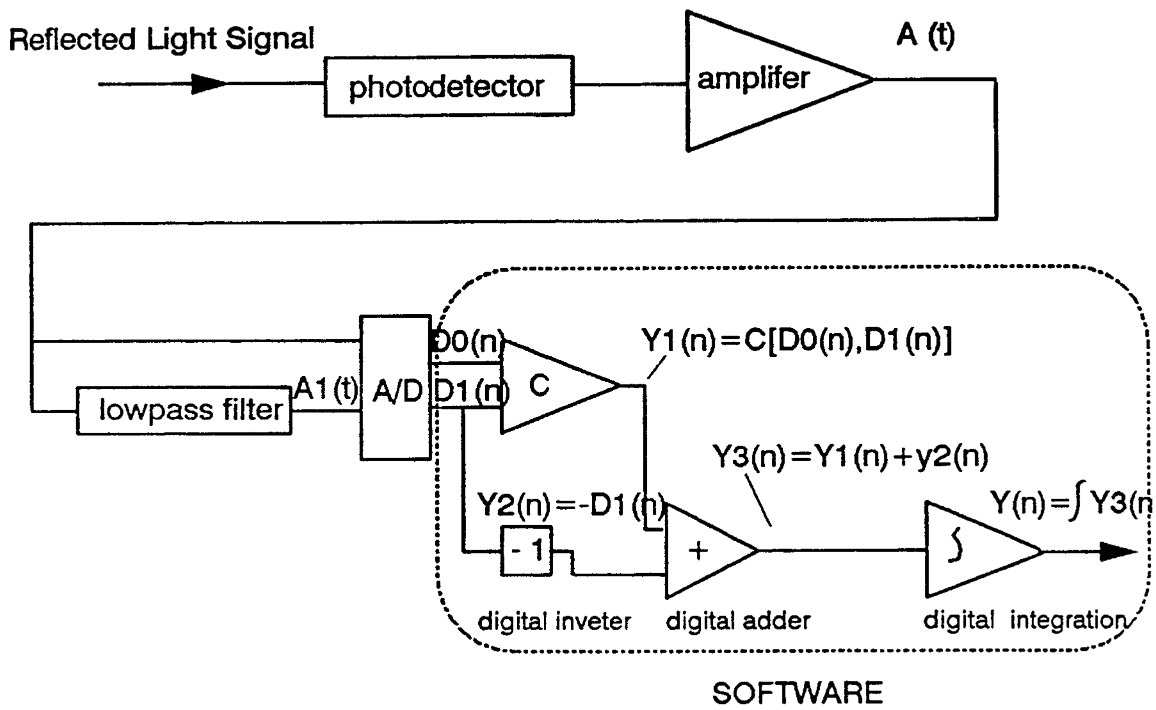


Figure 2.1 Signal processing flow chart.

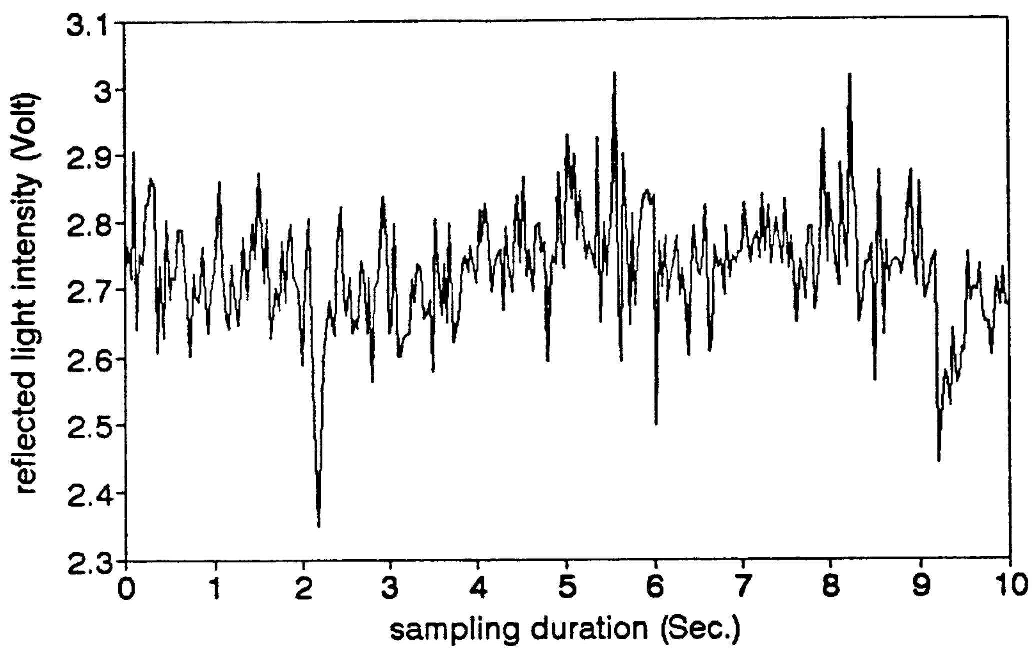


Figure 2.2 Test signal D0(n).

concrete. Baseline describes minute variations of refract index within concrete along the sensor's travelling path.

Four digital operational steps which are executed by software are involved in the flowchart :

(1) Digital comparator operation

$$Y1(n)=C[D0(n),D1(n)]$$

where C means digital comparator operation. Digital comparator algorithm is defined as :

$$Y1(n) = \begin{cases} D0(n) & \text{if } D0(n) > D1(n) \\ D1(n) & \text{if } D0(n) \leq D1(n) \end{cases} \quad (2.1)$$

At this stage, test signal in Fig. 2.2 is compared with its baseline counterpart in Fig. 2.3. Amplitudes of the individual signal above the baseline represent relative measure of air bubble size distribution. Values below the baseline represent water which is not thoroughly mixed in concrete. The large refractive index in water brings about more refraction, and therefore lower reflected intensity. In a well mixed concrete, the number of data points below the baseline is minimal. For example, data represented in Fig. 2.2 contain 24 points below the baseline, 5 points on the baseline and 271 points above the baseline. After digital comparator operation, all the values below the baseline are given the values of the baseline at that point, and the remaining values are kept the same for later signal analysis (Fig. 2.4). In this way, the contribution from improperly mixed water is eliminated.

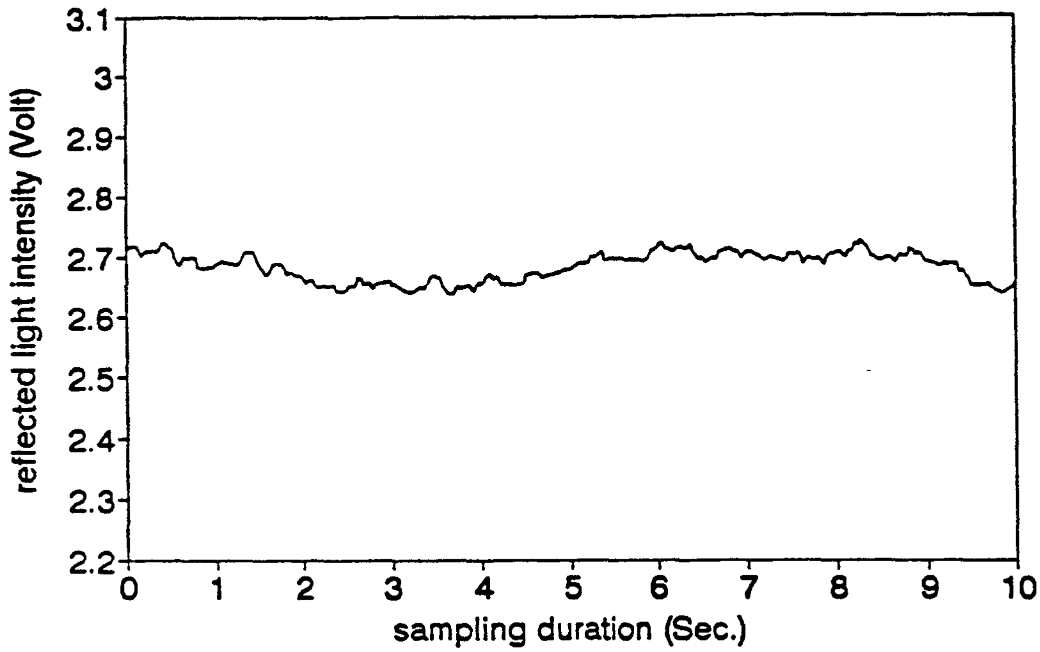


Figure 2.3 Filtered data D1(n) recorded by channel 1 representing the baseline along the sensor path.

(2) The digital inverter operation is performed as

$$Y2(n) = - D1(n)$$

where the minus sign means digital inverter operation.

The digital inverter algorithm is defined as :

$$Y2(n) = - D1(n) \qquad (2.2)$$

(3) Digital adder operation is performed as:

$$Y3(n) = Y1(n) + Y2(n)$$

where the plus sign means digital adder operation.

The digital adder algorithm is defined as :

$$Y3(n) = Y1(n) + Y2(n) \qquad (2.3)$$

Fig. 2.5 represents a test signal after a digital adder operation. The digital adder sums the outputs of the digital comparator and the outputs from an inverter. The output of the inverter is the negative value of the baseline. Therefore, in this way, the signal from the comparator is subtracted from the baseline. In other words, Fig. 2.5 depicts the net response with respect to the baseline.

(4) Digital integration operation $Y(n) = \int Y3(n)$

where $\int$ means digital integration operation.

Digital integration algorithm (Tapezoid Rule) is defined as :

$$Y(n) = \sum_{i=1}^{n} \frac{(Y3_i + Y3_{i+1})(T_{i+1} - T_i)}{2} \qquad (2.4)$$

The area below the net response of the test signal in Fig. 2.5 is calculated through digital integration operation. The Trapezoid Integration Rule is used to perform

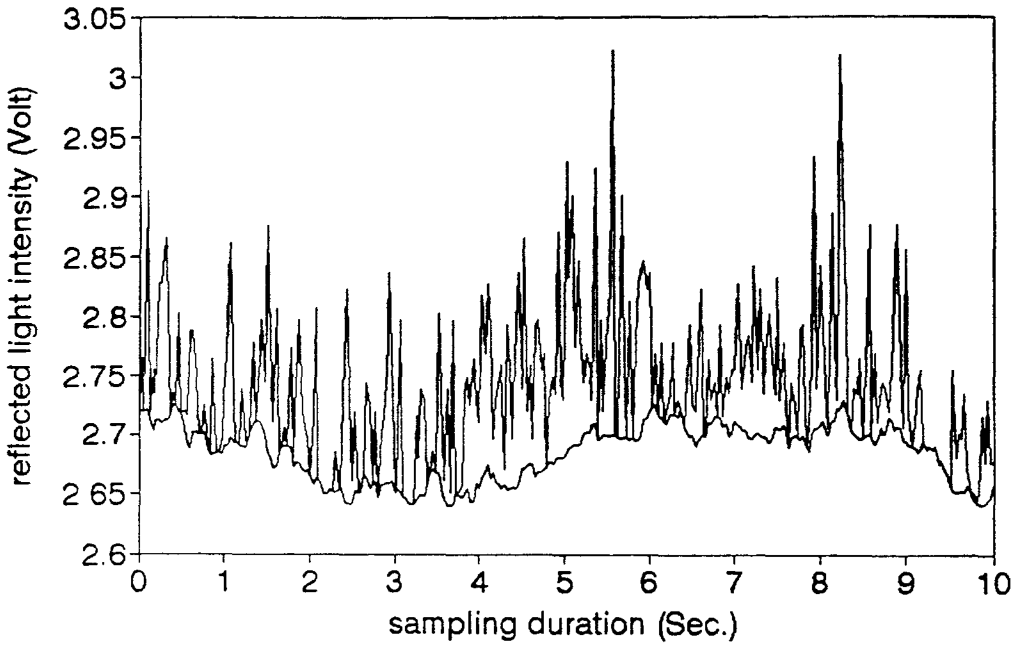


Figure 2.4 Signal after a digital comparator operation.

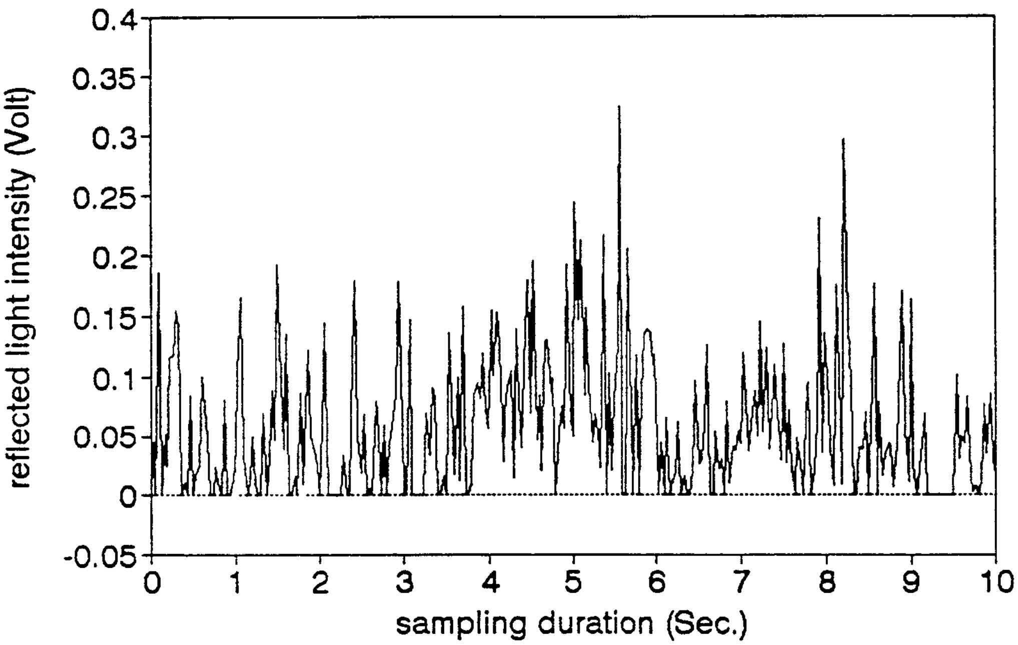


Figure 2.5 Signal after a digital adder operation.

the same. The area below the net response will be used in the determination of air content.

2.1.2 Signal Frequency Analysis

Fast Fourier Transform (FFT) plays an important role in applied engineering practices. The central concept of Fourier analysis is that virtually any waveform can be expressed as a discrete or continuous sum of sine and cosine functions, or equivalently, as a sum of complex (i.e. real and imaginary) exponential functions. This simple idea proves extremely useful because we often gain more information by viewing a signal as a sum of functions rather than as a signal phenomenon. In this thesis, FFT is only used as a method to obtain a frequency spectra for every step of the signal operation and to improve these operation after analyzing frequency spectra. The signal analysis software DADiSP is used for FFT analysis of test data.

A Fast Fourier Transform applied to data shown in Figs. 2.2 through 2.5, results in the frequency spectra in Figs. 2.6 through 2.9. As can be observed from the frequency spectra shown in Fig. 2.6, the components of low frequency signal (less than 0.2H) have higher spectral amplitude. This high amplitude low frequency component is made up of two parts : (1) component of the signal baseline, (2) component of the low frequency test signal. As it is shown in Fig. 2.7, the spectral amplitude of the low frequency signal is also large in the baseline, primarily because it is a low-

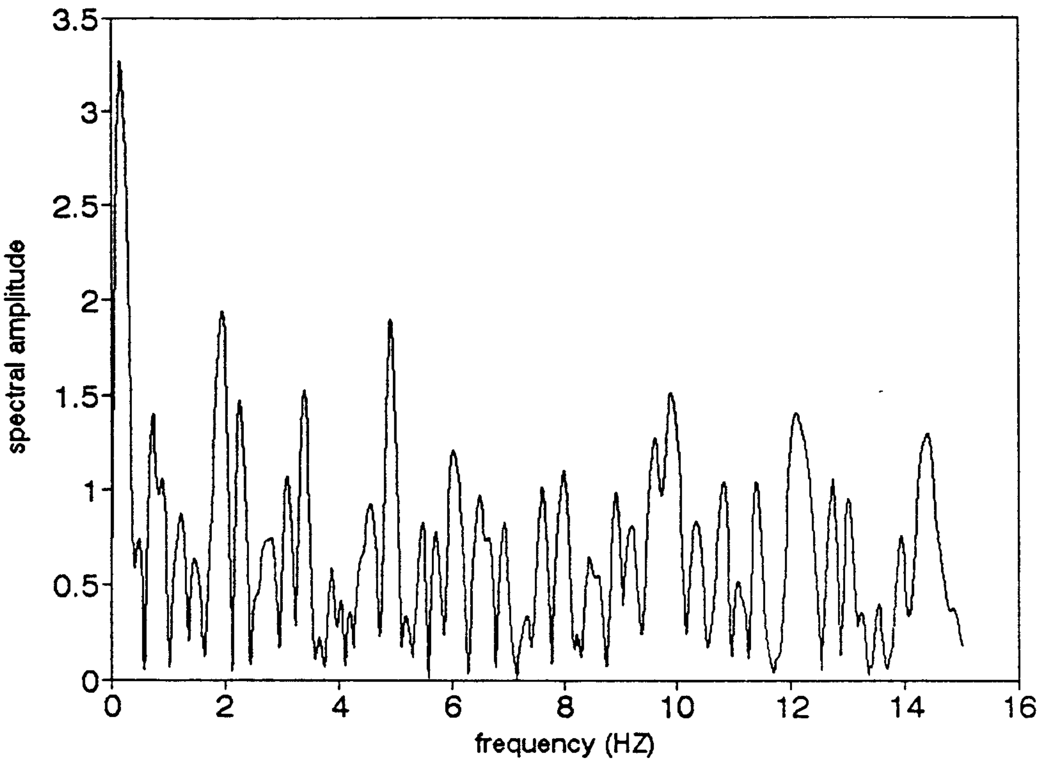


Figure 2.6 FFT of test data signal.

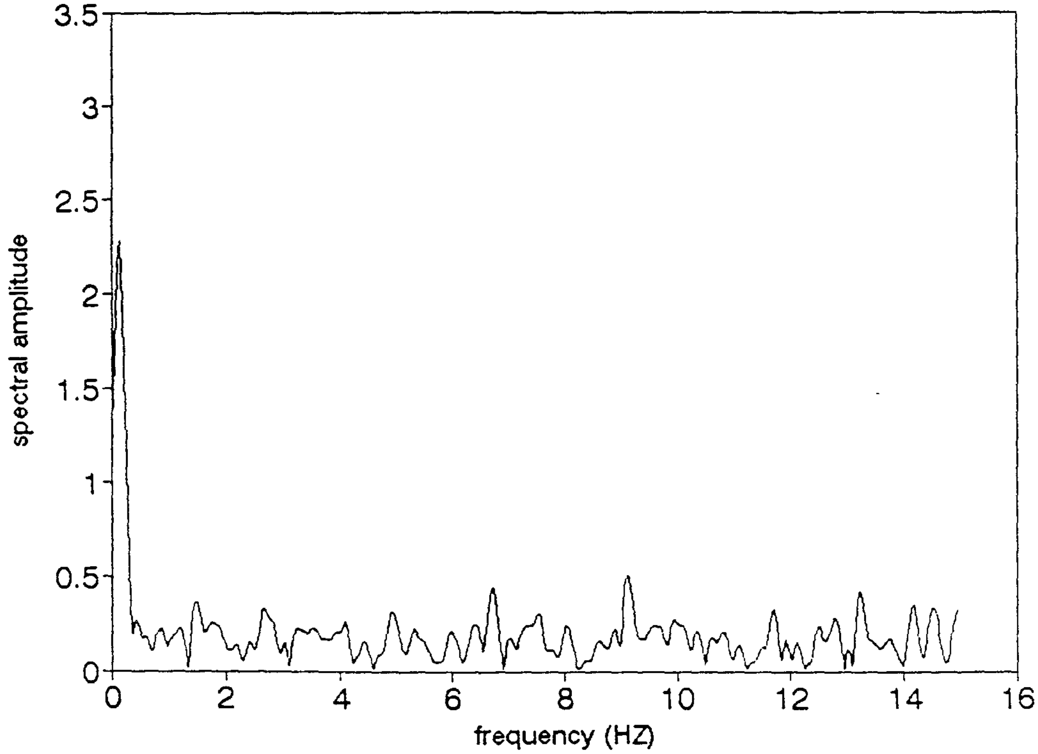


Figure 2.7 FFT of baseline signal.

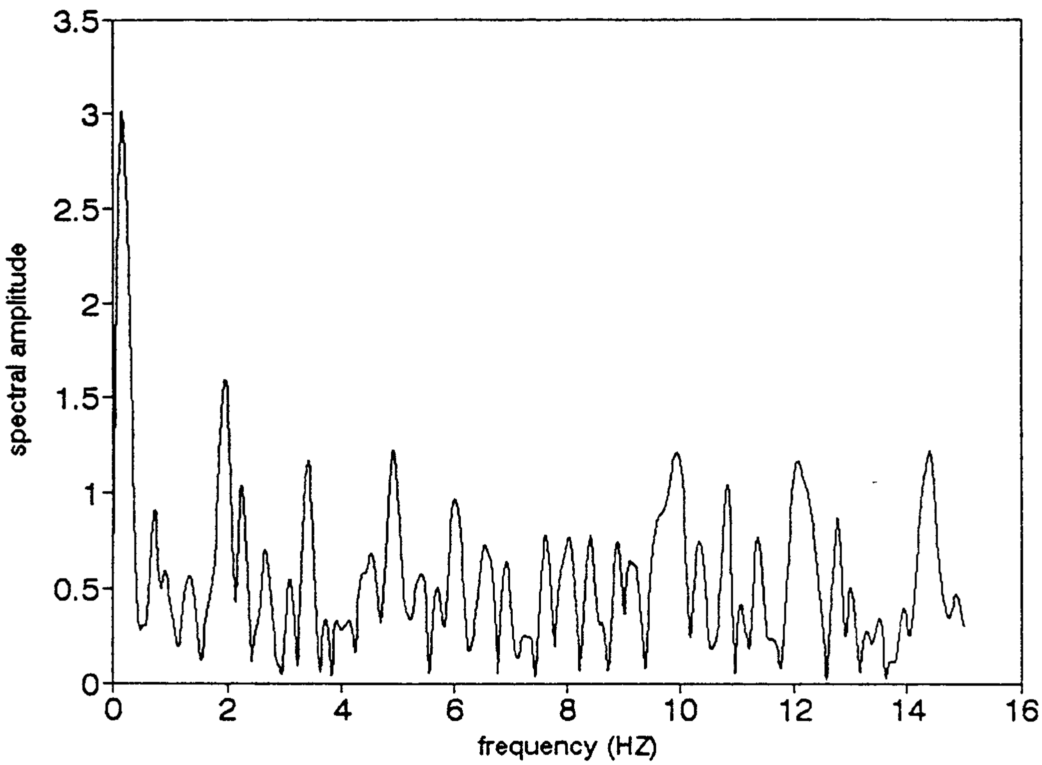


Figure 2.8 FFT of signal Y1(n).

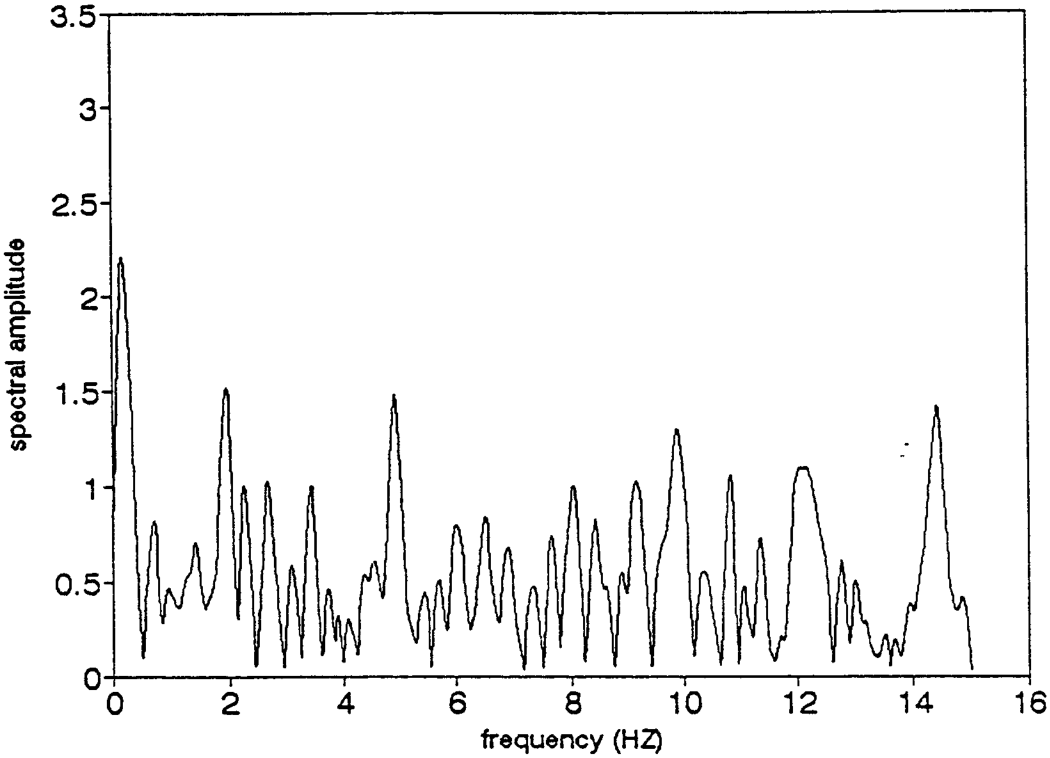


Figure 2.9 FFT of signal Y3(n).

passed filter data. Therefore, the baseline high value baseline high value of spectral amplitude in the lower frequencies has a dominant effect on the spectral response of the test signal (Fig. 2.6). Following the digital adder operation (Fig. 2.9), the baseline influence is significantly reduced as the low frequency spectral amplitudes are lowered.

The bubble frequency is defined as $F_n = N / L$, where N is total number of bubbles and L is the total length of traverse in concrete. F_n is the number of bubbles intercepted per unit length of total traverse. It is usually calculated by the microscopic examination of hardened concrete by the linear traverse or the point count methods as described in ASTM [17]. The larger the value of F_n, the better the protection against freezing, at least up to a point, because smaller bubbles contribute maximum to the air content. The PCA recommends that the bubble frequency per inch to be one and a half to two times the percentage air content. So properly air entrained concrete (4% to 8%) will have a bubble frequency of about 8 to 16 per inch (300 - 600 per meter). The sensor moving speed is about 2 inch per second. Therefore, the bubble signal frequency is about 12 - 30 Hz. This agrees well with the signal analysis through FFT. FFT results indicate the spectrum range to be within 12-30 Hz as well. Therefore, the sampling frequency chosen according to the sampling procedure technique [16] is chosen as:

$$f_s \geq 2*f_{max} = 2*30 = 60Hz.$$

In the present system a sampling frequency 100 Hz is employed.

2.1.3 Calibration Method

Conversion of a test signal to a single value represents percentage of air requires for a calibration procedure at the beginning of measurements. The maximum intensity readings corresponding to air are then acquired first. The calibration reading in regard to the minimum value associated with concrete is the curve corresponding to the baseline.

The area between the base line and the horizontal lines, represents the response in concrete and air respectively (Fig. 2.13) corresponds to 100 percent air content, A_w. In the example shown in Fig. 2.10, this value is equal to 8.4 ($A_w = (3.51 - 2.67) * 10 = 8.4$). The area under the test signal data A_p is achieved by integration; in the example shown in Fig. 2.5, $A_p = 0.3385$. Percent air in concrete is then calculated as $(A_p / A_w) * 100$. In this case, percent air in concrete is equal to 4.03.

2.2 System Operation

Two modes have been developed in this system, one is Lab mode, the other is on-site mode. In Appendix H, operational steps leading to the evaluation of percent air in concrete is outlined separately.

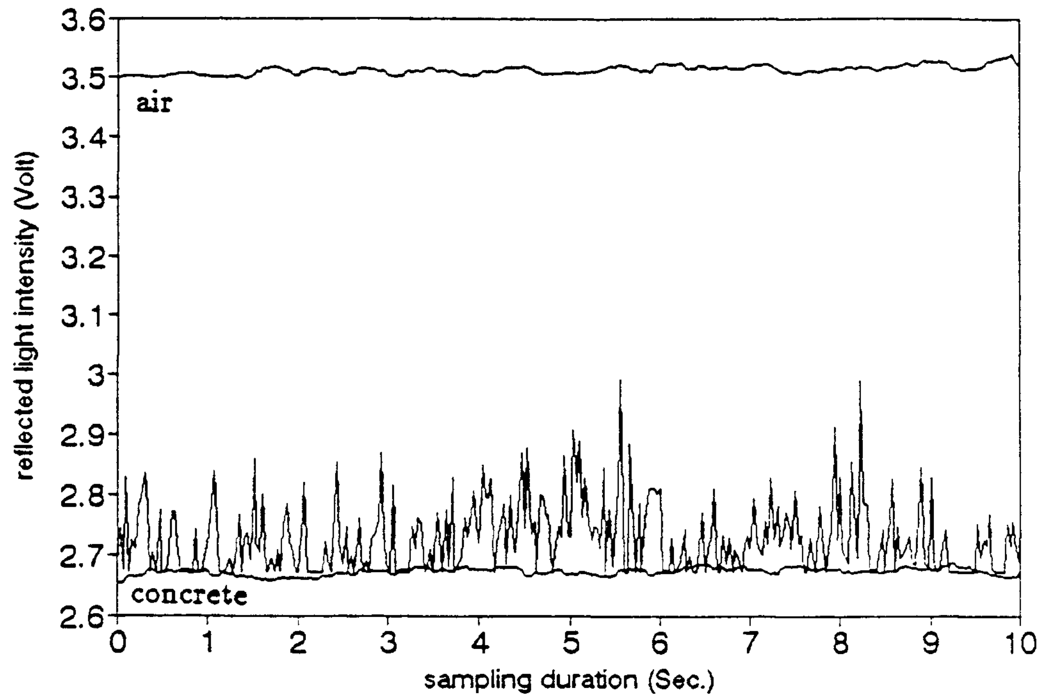


Figure 2.10 Example of one experiment test data calibration.

CHAPTER 3

HARDWARE DESCRIPTIONS

3.1 Overview

Fig.3.1 shows the on-site mode system hardware block diagram. The system consists of three parts: The development tool, the master IBM PC, and the microcontroller system. The microcontroller development tool is an In Circuit Emulator (ICE). The controller system consists of a mini controller board including with the 8031 controller, 8K RAM, 8k EPROM, A/D converter 0809 and extending I/O chip 8155, I/O interface circuit (digital and analog) and an LCD display.

An IBM compatible PC is employed for further data processing, displaying more information about air bubbles in concrete and storing test data in hard disk or floppy disk. An RS 232 (UART) is used for communication between master PC and the controller system.

3.2 In Circuit Emulator (ICE)

During the development of a microcontroller system, we needed some way to debug our application software. We must be able to view its execution and see its response to external events. In support of this, a cross simulator was used, which is a program runs on another machine (e. g. , the IBM PC) and simulates the operation of the microcontroller. An emulator, on the other hand, permited us to connect our development environment to the target

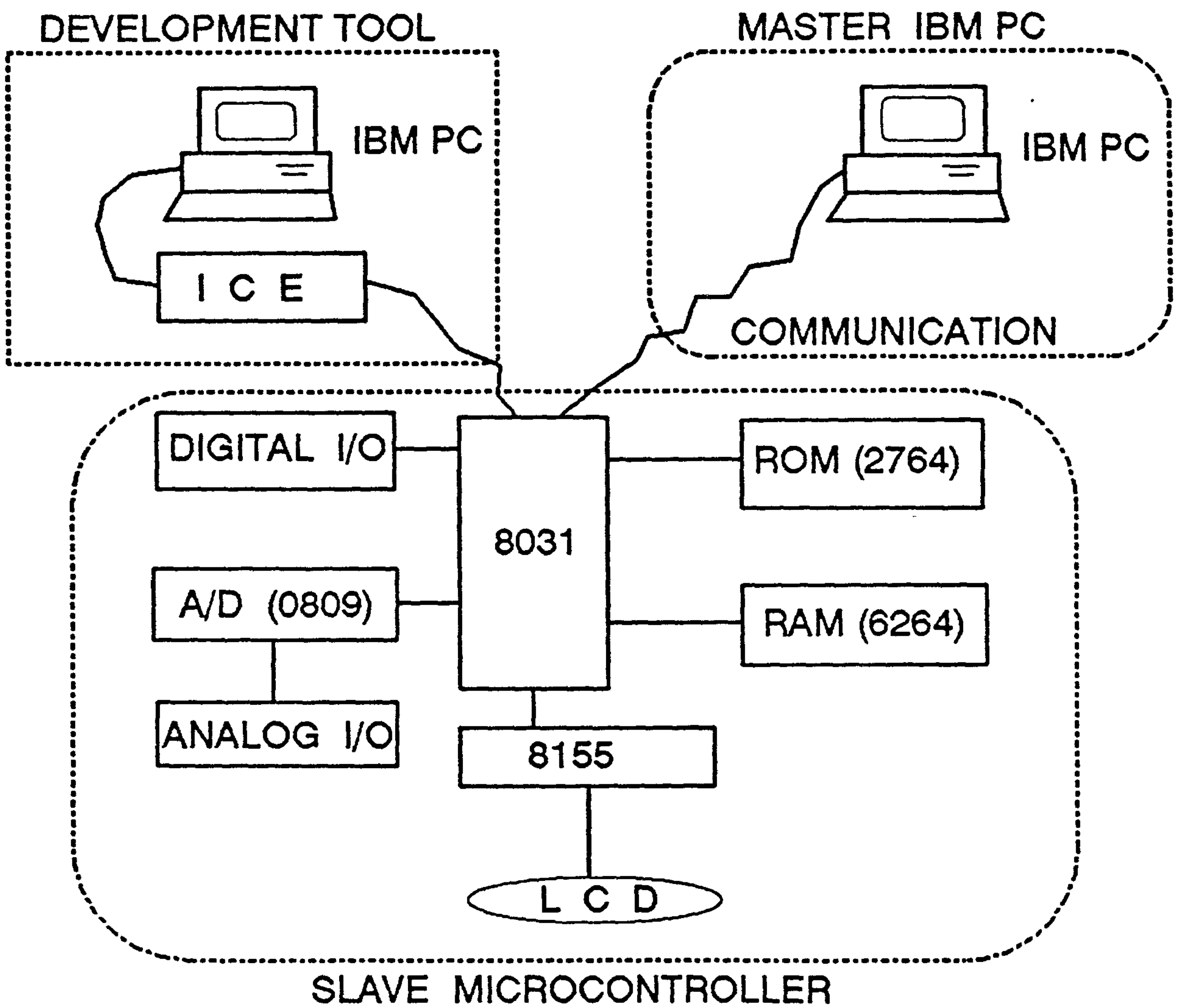


Figure 3.1 System hardware block diagram.

system hardware and to run it. It emulated the behavier of the microcontroller in the target system whilepermitting us to load programs and to monitor and control the innards of the (emulated) microcontroller chip. An emulator permits us to carry out debugging in the actual target-system environment whereas a simulator must deal with a reconstructed version of reality.

The ICE used in our system is SICE - III [10]. It is a complete hardware/software debug environment for developing embedded control applications based on the Intel 8051 family of peripherals. With high- performance 12 MHz emulation, symbolic debugging, and flexible memory mapping, the SICE - III emulator expedites all stages of development : hardware development, software development, system integration, and system test.

3.3 Laser Diode Control and Driver Circuit

The power output of a laser diode is easily changed by the fluctuation in the ambient temperature. Therefore, it is important to keep the power output constant in this system. A drive circuit with an automatic power control (APC) function, acting as a feedback loop from the output of the monitor photodiode to the input of the laser diode, is normally used to maintain constant power output in an environment where temperatures may vary. The circuit and device list are shown in Appendix B.

3.4 I/O Interface and LCD

The I/O interface circuit is shown in Appendix E Fig. E.1. In this device, four buttons are used for operation. The "Reset" button is for resetting the microcontroller and making the program run from the beginning. The "Air Value" button is used for measuring the constant value of the reflected signal in the open air . The "Test" button is used for starting a test in the concrete. The "Comu" button is used for communication with master IBM PC. At the press of each one of the above-mentioned buttons, a high level ("1") signal is given to the controller. The RC filter and the Hex Schmitt Trigger IC 74LS14 are used for resisting and suppressing noise.

An OPTREX's Dot Matrix Liquid Crystal Display (LCD) [11] is used for displaying the air value and air content in concrete as a percentage. The LCD is connected to 8031 via an 8155 extended I/O port chip . The 8 bit data pins are connected to 8155 port A. The control signal pins, namely the register select (RS), read/write (R/S), and enable (E) are connected to 8155 port B (p0, p1, p2) respectively. The timing requirements of the LCD are a little slow for a full speed 8031. The critical timing parameters are the enable pulse width of 450 ns, and the data delay time during read cycles (t_{DDR}) of 320 ns. Signal delay is achieved through software manipulations.

3.5 Analog Interface

In the analog interface circuit, First, a photodiode AX65-R2F is used to transfer the reflected light signal to electrical current. After that a zero bias photoamperic circuit is designed. Because of the fixed load resistance, the output voltage is linearly dependent on the generated photo current. The circuit is a part of the amplifier circuit shown in Appendix E, Fig. E.2. It has the advantages of linear response and low noise due to the almost complete elimination of leakage current.

Second, the operational amplifier is provided to shift the range so that expected reflected light signal will produce a voltage in a range between zero and five volts to make use of the full range of the analog to digital convertor. Two stages of amplification have been employed. After zero bias photoamperic operation, the reflected light intensity output ranges between 0 - 7 mv. The gain of first stage amplifier is set between 50 - 70. Adjustments to the laser-to-fiber coupler bring about reflected light intensity changes. The first stage gain provides flexibility in light intensity adjustments by being able to vary within the above mentioned range. The second stage gain is fixed (gain = 10). Therefore, the gain in the amplifier varies from 500-700. In the second step amplifier, one offset is added for changing the maximum signal output range within the accepted maximum input range of the data acquisition interface board system sensor is changed, the reflected light intensity output would change depending on the sensor tip characteristics and the sensor to directional coupler connection. The offset adjustment can make the reflected light intensity output of the analog amplifier fall within the linear range.

A low-pass filter is employed in obtaining a baseline for the test signal. The operational amplifier together with resistance capacitance networks are used extensively in the implementation of various types of filter network. The cutoff frequency of the filter is given by :

$$f_c = \frac{1}{2\pi(R_{11}R_{12}C_1C_2)^{\frac{1}{2}}} \qquad (3.1)$$

where f_c is the frequency in Hz, R_{11}, R_{12} are resistance in ohms and C_1, C_2 are capacitances in farads [12]. For example, if R_{11} = 100k, R_{12} = 100k, R_{13} = 200k, C_1 = 20uf, and C_2 = 4.7uf. Then F_c = 0.16 Hz. A cutoff frequency of 0.16 Hz was chosen for the present system by trials. In on-site mode, the reference voltage of A/D is +5V, we put a inverter after the low-pass filter, since the output of low-pass filter is negative.

3.6 Microcontroller System

The distinguishing characteristic of a microcontroller is the inclusion, on one chip, of all the resources which permit it to serve as the controller in a device or instrument.

In this system, an Intel 8031, one member of the MCS-51 family, is used as controller. The device includes the following features :

- 8-bit CPU optimized for control applications
- Extensive Boolean processing (signal-bit logic) capabilities
- 64K Program Memory address space
- 64K Data Memory address space
- Four register banks
- 128 bytes of on-chip Data RAM
- 16 bidirectional and individually addressable I/O lines
- Two multiple mode, 16-bit timer/counter
- Full duplex UART
- 6-source/5-vector interrupt structure with two priority levels
- On-chip clock oscillator

Fig. F.1 shows a typical 8031 based design mini system (FD-31-S6) which includes a port expander (8155), an 8-bit A/D converter (0809), an EPROM containing stable code and 8K RAM. They are memory mapped into the external data RAM address area. The addresses are as follows:

EPROM	2764	0000H - 1FFFH
RAM	6264	C000H - DFFFH
8155		
RAM		7E00H-7EFFH
I/O	COMMAND/STATUS	7FF8H

I/O	COMMAND/STATUS	7FF8H
	PART A	7FF9H
	PART B	7FFAH
	PART C	7FFBH
	TIMER LOW 8-BIT	7FFCH
	TIMER HIGH 6-BIT	7FFDH
A/D	0809	BFFFH

The 8031 controls the address latch, start and output enable of the A/D converter by address line P2.6, write and read control line (Fig. F1). The clock of A/D is supplied by ALE (Address Latch Enable) which continues to be activated twice per machine cycle (1 MHz). The EOC (End Of Conversion) pin of 0809 is connected to p1.2 pin of 8031. When conversion is finished EOC assumes the high value. According to the timing of 0809, after starting the converter, the EOC becomes unstable for about 10 μs. This instability should be considered and corrected in the software.

3.7 UART Interface

In this system, the full duplex UART serial port is used for communication between the microcontroller system and the master IBM PC. The hardware block diagram is shown in Fig. 3.2.

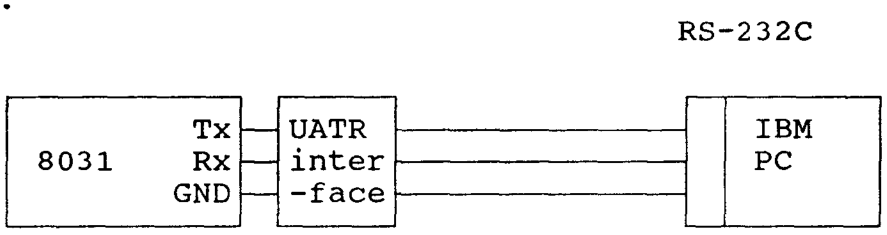


Figure 3.2 Hardware block diagram of communication.

The interface circuit is supplied for signal level conversion. The detail circuit is shown in Appendix E Fig. E.3.

CHAPTER 4

SOFTWARE DESIGN

4.1 Overview

The microcontroller programs were written in MBASIC which is a language combining BASIC and ASM51 assembly language. The master programs were for the most part written in BASIC to ease the programming effort. A number of subroutines were written in assembly language which is supplied by the DASh-8 software package. The programs are all done in a top-down fashion.

4.2 Microcontroller program

To make things easy, the controller program is broken into a main program and several subroutines as follows:

- data processing subroutines
- carry subroutines
- display subroutine
- time delay subroutines
- a division subroutine

4.2.1 Main Program

The main program is documented in Appendix I.1. The flow chart of main program is shown in Fig. G.1. Some peculiarities are described as follows:

1. Initialization

The first part of main program is initialization. It includes defining variables, setting stack and data base pointer. The MCS-51 family has only one 16-bit data pointer for addressing extending memory. We should always store the pointer on the stack. When signal data is stored in the data base, the pointer is popped from the stack. After storage, the pointer is incremented and pushed back onto the stack.

2. Button Debouncing

The two-position switch or button is a common device, alone, in groups and in keyboards. Fig. 4.1 below shows how a simple switch can be connected to the computer.

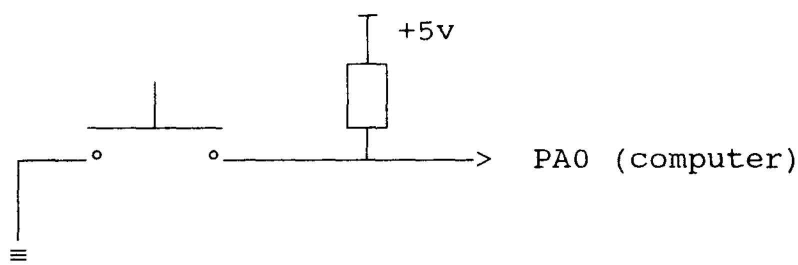


Figure 4.1 Switch connected to computer.

When the button is up, line PA0 is pulled high and a read operation by the computer will find a binary 1. When the button is down, the line is pulled low, and the computer will find a binary 0. But switches generally have bounce. Just after the contact position is changed, there is a brief period during which the position of the switch is indefinite. If we should sense the switch during this transition, unpredictable results may lead to inaccurate results. One way of avoiding this phenomenon is to write into the program a delay, testing the switch position both before and after the delay. If they disagree, the switch is moving. If they agree, the switch is unstable, otherwise the switch is in a stable position. Depending on the switch structure this delay is generally from 1 to about 20 ms; a 20 ms delay period was selected for the present system.

3. Data acquisition loop

A loop has been designed for data acquisition in the main program. The total execution time of this loop is about 604 μs (including two A/D converting time). Therefore, the maximum sampling frequency which we can select is about 1.6 KHZ. This is sufficient for our system. The sampling frequency selected in this system is 100 Hz, the duration is 5 seconds. The total signal data is comprised of 500 hundred pointers. Time 0 of 8031 is set to get the sampling frequency.

The timer produces and accumulates errors due to the delay between sampling intervals. When the set time of timer 0 is ended, it means the sampling time is available . If the timer for the next sampling does not immdeiately atart, and other statements are executed first, they will produce sampling frequency errors. These errors will accumulate to a large value because of the large sampling times. To remedy the program, the timer (t0) for the next sampling interval is started immediately when sampling time becomes available. Moreover, we use "ORL TL0, #77H" statement instead of "MOV TL0,#77H" to reduce accumulating error. The former needs less execution time than that of the latter.

4. Integration algorithm

The integration algorithm is simplified in this program. It makes the program execute more effectively and faster. As described in chapter 2 ,four digital operation algorithms are defined as (2.1), (2.2), (2.3), and (2.4). To reduce duplications, the integration algorithm is modified as follows: First, the Y3(i) should be added two times in equation 2.4 ; second, when D1(i) $\geq$ D0(i), Y3(i) = 0, adding zero is unnecessary in equation 2.4.

The integrating algorithm in this program is defined as follows :

$$Y(n) = \tfrac{1}{2}Y3(1) + \tfrac{1}{2}Y3(n) + \sum_{i=2}^{n-1} Y3(i) \qquad (4.1)$$

The program flowchart used for this algorithm is shown in Fig. 4.2.

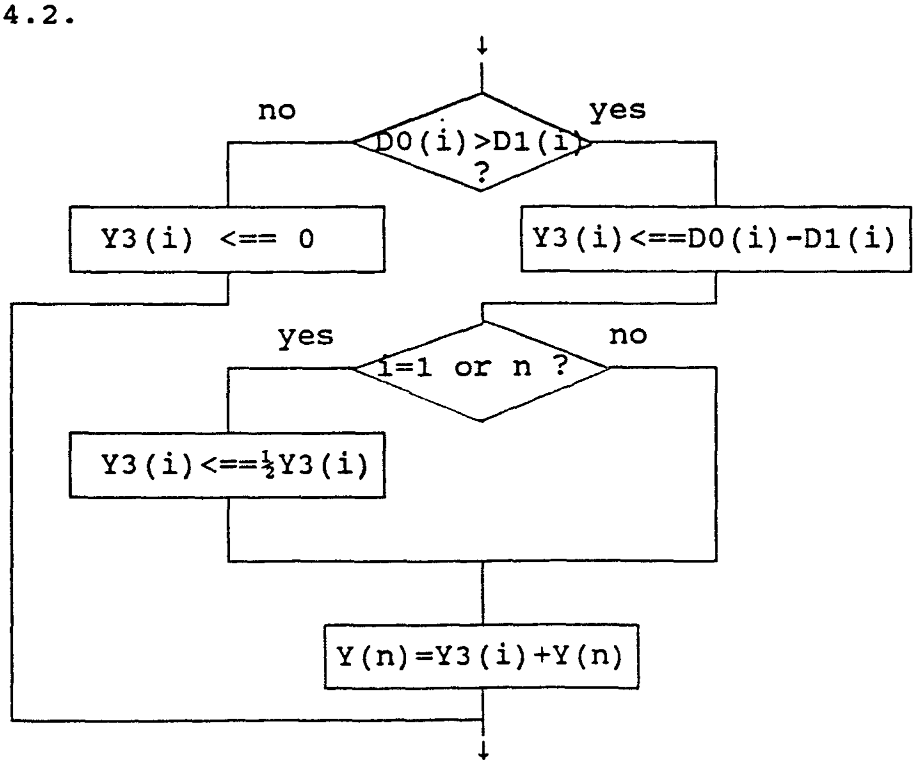


Figure 4.2 Flow chart of integrating algorithm.

4.2.2 Data Processing

A major part of the design of a program is the design of data structures. In other words, how we organize the data is a very important part of the design effort. Once we know that organization, we can proceed with the programming. Until we know it, we can not. In the present system, the data structure will form the foundation for all of the information that our program works with.

In this system, the A/D converter is 8-bit, and the LCD is worked with ASCII code. A particular data structure has been chosen that fits this hardware design. An unsigned binary data structure is used for performing A/D conversions and all the required computations. The code and flowchart of unsigned 2-byte division using compared method are shown in Appendixes I.1 and Fig. G.3 respectively. It's simple and needs less execution time. After computations, the unsigned binary number is changed to binary floating-point number. The formula of binary floating-point number is shown as follows:

address	7	6	5 0
(Ri)	sign of number	sign of exponent	exponent
	7 0		
(Ri)+1	high byte of mantissa		
	7 0		
(Ri)+2	low byte of mantissa		

In this way, the floating-point number is processed according to the actual voltage value. The binary floating-point number is converted to a decimal floating-point number for the LCD display. The decimal floating-point number consists of 6 bytes. The first byte is the exponent (twos-complement), whereas the remains bytes (2 through 6) are the mantissa. LCD will display calculated values after the completion of data processing. The programming code and flow chart for the data processing subroutine are shown in Appendix I.1 and Fig. G.2 respectively.

Resident data memory is organized as 128 by 8-bits wide in the 8031. The lowest 32 bytes are grouped into 4 banks of 8 registers. Program instructions call out these registers as R0 through R7. Two bits (RS1, RS0) in the Program Status Word (PSW) select which register bank is in use. This allows more efficient use of code space, since register instructions are shorter than instructions that use direct addressing. All of the bytes in the data memory can be accessed by either direct or indirect addressing. The data memory map designed for this system is shown in Fig. 4.3. Because of the difference in defining the formula for floating-point numbers between standard definition and MBASIC, sometimes conversion is necessary.

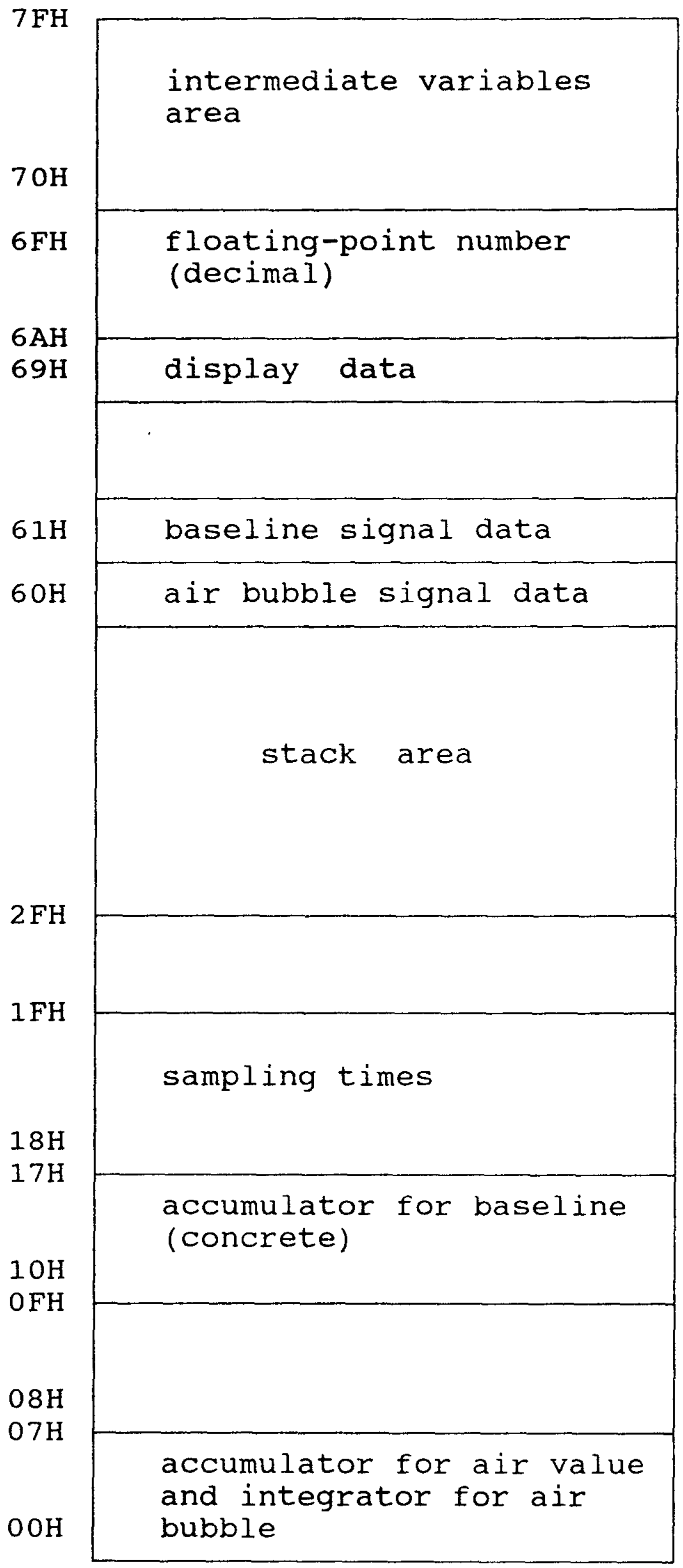


Figure 4.3 Data Memory Map.

4.2.3 Communication with the master IBM PC

Configuring the 8031's serial port for a given data rate and protocol requires three short sections of software. On power-up or hardware reset, the serial port and timer control words must be initialized to the appropriated values. Additional software is also needed in transmit routine to load the serial port data register and in the receive routine to unload the data as it arrives.

In this system, the 8031 communicates with an IBM PC at 1200 baud (bits per second). Each character is transmitted as 8 data bits, one start bit, and one stop bit. This results in a character rate of 1200-10=120 characters per second.

For the sake of clarity, the transmit and receive subroutines are driven by simple-minded software status polling code rather than interrupts. The serial port must be initialized to 8-bit UART mode (M0,M1=0,1) enabled to receive all messages(M2=0, REN=1). The flag indicating that the transmit register is free for more data will be artificially set in order to let the output software know the output register is available.

Timer 1 of 8031 is used in auto-reload mode as a baud rates generator. The reloaded value is calculated in the following formula:

$$\text{Baud rate} = (f_{osc}/12)/(256\text{-TH1})/(32/2^{SMOD}) \qquad (4.2)$$

where f_{osc} is Oscillator Frequency.

SMOD is 7th bit of special register PCON (it can be set by program).

TH1 is time setting value of timer

In this system, fosc=11.0592 MHZ, Baud rate=1200, and SMOD=0. Substituting these values to Eq. 4.2, results in TH1=232. The timer 1 must reload the value 0E8H (Hex).

4.3 Master Computer Program

The BASIC code for master IBM PC is listed in Appendix I.2. The master program consists of the LAB measuring mode and the master mode. Before every section, a set of declarations is used for explaining the main function of this section.

In the body of the program, a sign is set to distinguish several different operations. When sign=1, the program runs for viewing the test results. Sign=2 means the program is running for performing a test. When sign=3, the program runs for communicating with microcontroller.

In the graphics section of main program, the program first scans the data files and sets the Max/Min value of X, Y axis, then locates the axis x, y and plot them out with the small tick marks identifying the scale. So, all the data files are displayed on the same scale.

In LAB measuring mode, an eight channel data acquisition board DAS-8 should be installed on IBM PC, and a "DAS8.BIN" file should be loaded at the beginning of the main program. DAS8.BIN supplied with the DAS-8 board is a machine language I/O driver for control of A/D, timer, and digital I/O functions via a BASIC CALL. DAS8.BIN can select multiplexer channels, set channel scan limits, perform triggered A/D conversions, control interrupt driven conversions and scans, set/read the timer counter and measure frequency and period [9].

In master mode, the IBM PC communicates with a slave controller by means of an RS 232 series port. When the program opens a communication file, the parameter should be set the same as that of the slave controller. After receiving data, the program should process it to get the actual value.

The main program allows you to perform many functions from a single menu-driven package. Data acquisition and process control, data storage and screen display of experimental data, data manipulation and curve fitting, summary of the results of the experiment and communication with slave controller are all placed at the user's fingertips. All commands are menu driven and there is no need to remember command names or sequences. Data files generated by the main program are formatted for direct link to LOTUS 1-2-3 providing the full analytical power and graphing functions of LOUTS for data reduction.

CHAPTER 5

EXPERIMENTAL RESULTS AND DISCUSSION

5.1 Experimental Results

Air percent in fresh concrete is controlled by using an air entraining admixture (MB-VR) manufactured by Master Builders. A range of mixes of concrete with varying percentages of air content are obtained.

Three devices were used in experiments to determine the entrained air percentage:

- Precision Air Entrainment Meter (Model CT 126-A, made by Soiltest, pressure meter)
- Roll-a-meter (Volumetric meter, made by Soiltest)
- Fiber Optic System

A concrete mix proportion of 1 : 1.5 : 1.5 : 0.55 (cement : sand : aggregate : water) by weight was employed, and the amount of air entraining admixture was varied for each batch. For comparison, the pressure meter, the roll-a-meter and fiber optic sensor were employed for measurement of the air content on the same batch of concrete. For the fiber optic method, three measurements were taken in each test. The average of these three tests were used for comparison with the other two methods. Experimental results are listed in Table 5.1. The average of fiber optic results compared with the other two methods is shown in Fig. 5.1. Some laboratory test results are shown in Table 5.2 through 5.6.

Table 5.1 Test Results.

Volume of MB-VR Add (ml MB-VR /100 kg cement)	Percentage of Air Entrainment			
	Pressure Meter	Volumetric Meter	Data File Name	Fiber Optic Reading
5.4	2.1	2.6	N16D.DAT N16E.DAT N16F.DAT	2.47 2.34 1.94 2.25 (Avg)
10.8	3.3	3.6	N28G.DAT N28H.DAT N28L.DAT	3.41 3.02 3.22 3.19 (Avg)
16.3	3.6	3.6	N28O.DAT N28P.DAT N28Q.DAT	3.28 2.97 3.52 3.25 (Avg)
21.7	3.8	3.4	N27O.DAT N27P.DAT N27Q.DAT	3.55 3.49 3.55 3.53 (Avg)
32.5	4.7	4.5	N27U.DAT N27K.DAT N27L.DAT	3.80 4.03 4.03 3.95 (Avg)
43.3	4.9	5.0	D8G.DAT D8H.DAT D8I.DAT	5.31 4.99 4.71 5.00 (Avg)
48.8	6.6	6.8	D4J.DAT D4Y.DAT D4L.DAT	6.40 6.71 6.17 6.42 (Avg)

5.2 Air Bubble numbers and Size Analysis

Determination of air bubble size distribution is even more important than measurement of total air content. In the present system, the higher the amplitude of the reflected signal, the larger the air bubble size. The effect of bubble size and spacing on the amplitude of reflected light signal is shown in Fig. 5.2. In another parallel study [16], an automatic image analysis system is employed for the study of air bubble size distribution. in hardened concrete. A series of experiments as explained in the following were performed to check the possibility of determining the above-mentioned parameters.

Four different batches with varying amounts of MB-VR and a mixed proportion of 1 : 2.1 : 3.7 : 0.5 by weight of cement : sand : aggregate : water were prepared. A prism of 4"*4"*12" was cast with each batch for later microscopic analysis. Air content in each batch was measured by the pressure and fiber optic methods. Ten fiber optic tests were taken for each batch. Hardened concrete prisms were cut into a thin slice (3/8 inch thick) for microscopic analysis. Results are shown in Table 5.3 through 5.6 respectively. They indicate existence of a correlation between the air bubble size distribution obtained by the fiber optic and the

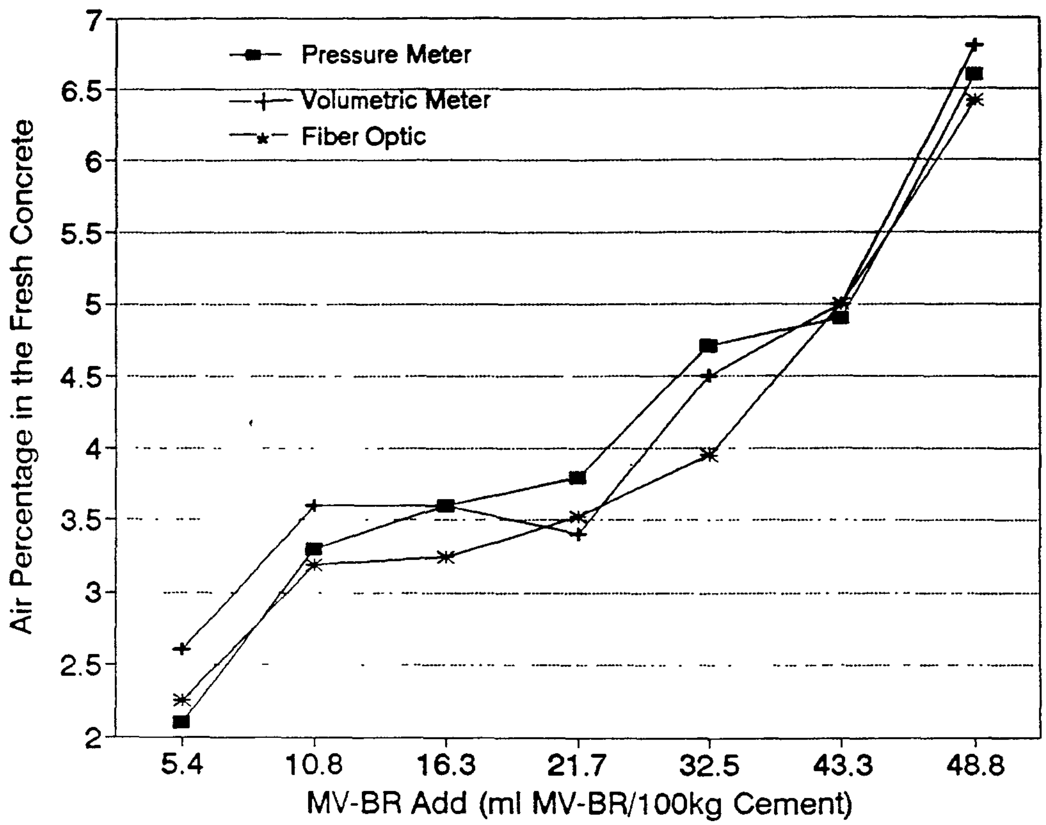


Figure 5.1 Three test methods comparison

Table 5.2 Typical experimental result

The Results of Experiment

09-16-1991

TEST NUMBER	AIR PERCENT
1	5.79 %
2	5.62 %
3	5.74 %
4	5.95 %
5	5.78 %
6	5.22 %
7	5.24 %
8	5.40 %
9	5.75 %
10	5.70 %
THE AVERAGE	5.62 %

Table 5.3 Experiment data for air bubble size analysis(1)

Date File Name	Fiber Optic Result	Pressure Meter Result
2F1.WK1	2.22	
2F2.WK1	2.49	
2F3.WK1	2.42	
2F4.WK1	2.45	
2F5.WK1	2.43	
2F6.WK1	2.32	
2F7.WK1	2.32	
2F8.WK1	2.27	
2F9.WK1	2.23	
2F10.WK1	2.04	
Average	2.32	2.2

Table 5.4 Experiment data for air bubble size analysis(2)

Date File Name	Fiber Optic Result	Pressure Meter Result
3F1.WK1	3.87	
3F2.WK1	4.37	
3F3.WK1	5.80	
3F4.WK1	4.52	
3F5.WK1	4.87	
3F6.WK1	3.77	
3F7.WK1	4.40	
3F8.WK1	3.88	
3F9.WK1	5.00	
3F10.WK1	3.87	
Average	4.43	4.0

Table 5.5 Experiment data for air bubble size analysis(3)

Date File Name	Fiber Optic Result	Roll-A-Meter Result
5F1.WK1	4.25	
5F2.WK1	3.49	
5F3.WK1	3.49	
5F4.WK1	4.52	
5F5.WK1	3.06	
5F6.WK1	4.06	
5F7.WK1	3.61	
5F8.WK1	4.31	
5F9.WK1	3.47	
5F10.WK1	3.90	
Average	3.82	3.90

Table 5.6 Experiment data for air bubble size analysis(4)

Date File Name	Fiber Optic Result	Roll-A-Meter Result
6F1.WK1	5.33	
6F2.WK1	5.90	
6F3.WK1	5.96	
6F4.WK1	6.13	
6F5.WK1	6.33	
6F6.WK1	5.88	
6F7.WK1	5.94	
6F8.WK1	5.58	
6F9.WK1	6.14	
6F10.WK1	5.89	
Average	5.91	5.75

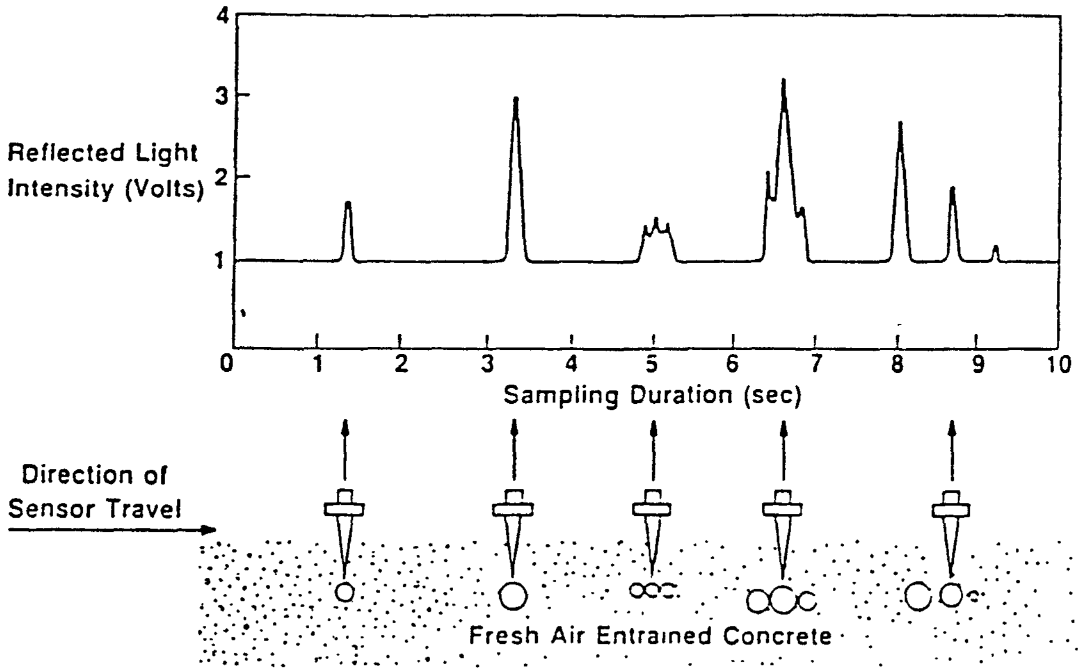


Figure 5.2 Effect of bubble size and spacing on the amplitude of reflected light signal.

image analysis methods. Relative air bubble size values are determined as the difference in between the amplitudes of the test signal and the baseline. These relative size value are given in voltage units, and can be divided into a few groups representing range of sizes as shown in Table 5.7.

Fig. 5.3 and 5.5 show the results of the fiber optic analysis method from Table 5.8 and 5.9. Fig. 5.4 and 5.6 show the results from the microscopic analysis of hardened concrete from the same samples. Two methods get the same conclusions. It can be seen that the total number of bubbles are much larger for specimens which are air entrained. Also the percentage of small bubbles is greater in the case of specimens with MBVR (AEA). This is because the air in specimens without an AEA is largely entrapped air which consists of large irregular air voids and are formed due to insufficient compaction. Air in specimens with AEA consists of small and spherical, regularly spaced voids which are stabilized by the AEA.

Table 5.7 Air bubble size groups.

Group	Amplitude Range (relative)
very very small air bubble	0 - 0.05 V
very small air bubble	0.05 - 0.075 V
small air bubble	0.075 - 0.10 V
medium air bubble	0.10 - 0.15 V
large air bubble	0.15 - 0.20 V
very large air bubble	> 0.20 V

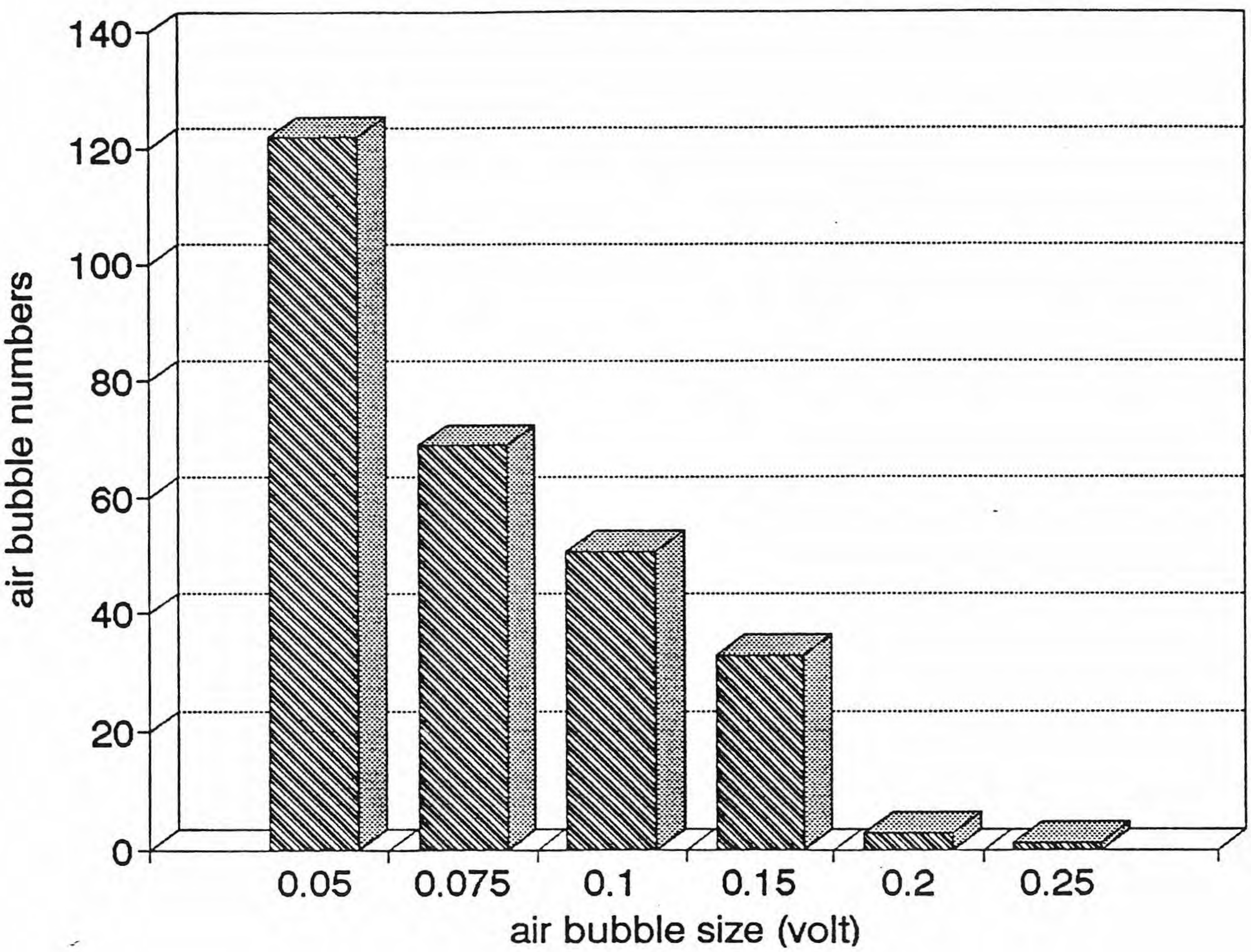
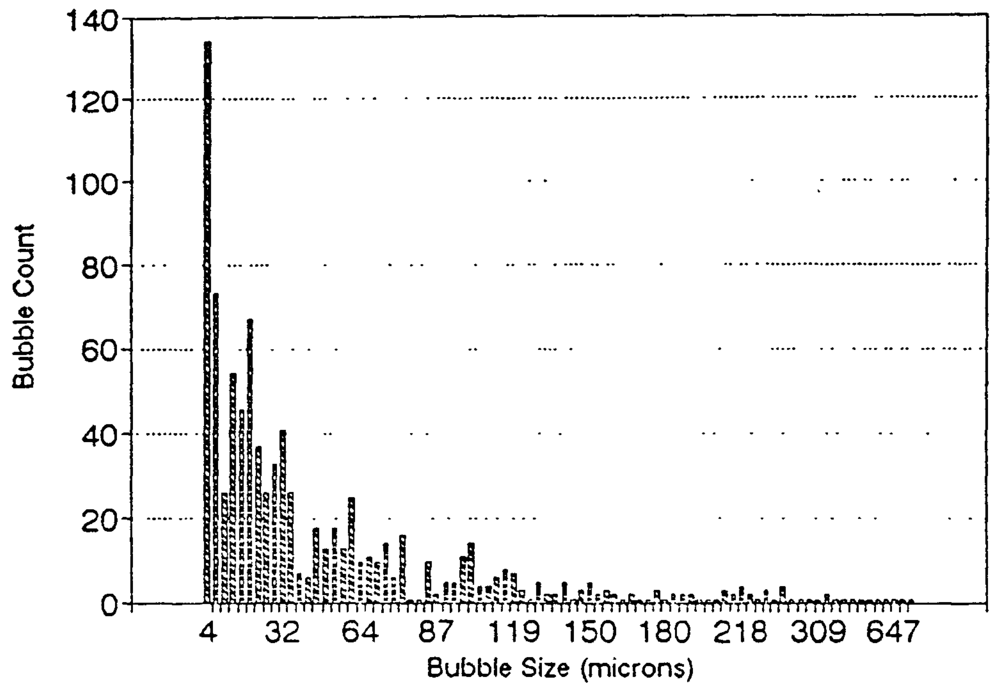


Figure 5.3 Air bubble size and numbers in concrete analysis using fiber optic method for test sample one.



Total Length of Traverse: 2553.6 mm (100.5 inches)
Number of Air Bubblee Intercepted: 975
Sum of Bubble Length: 52.571 mm

Air Void Content: 2.06%

Figure 5.4 Air bubble size and numbers in concrete analysis using microscopic method for test sample one.

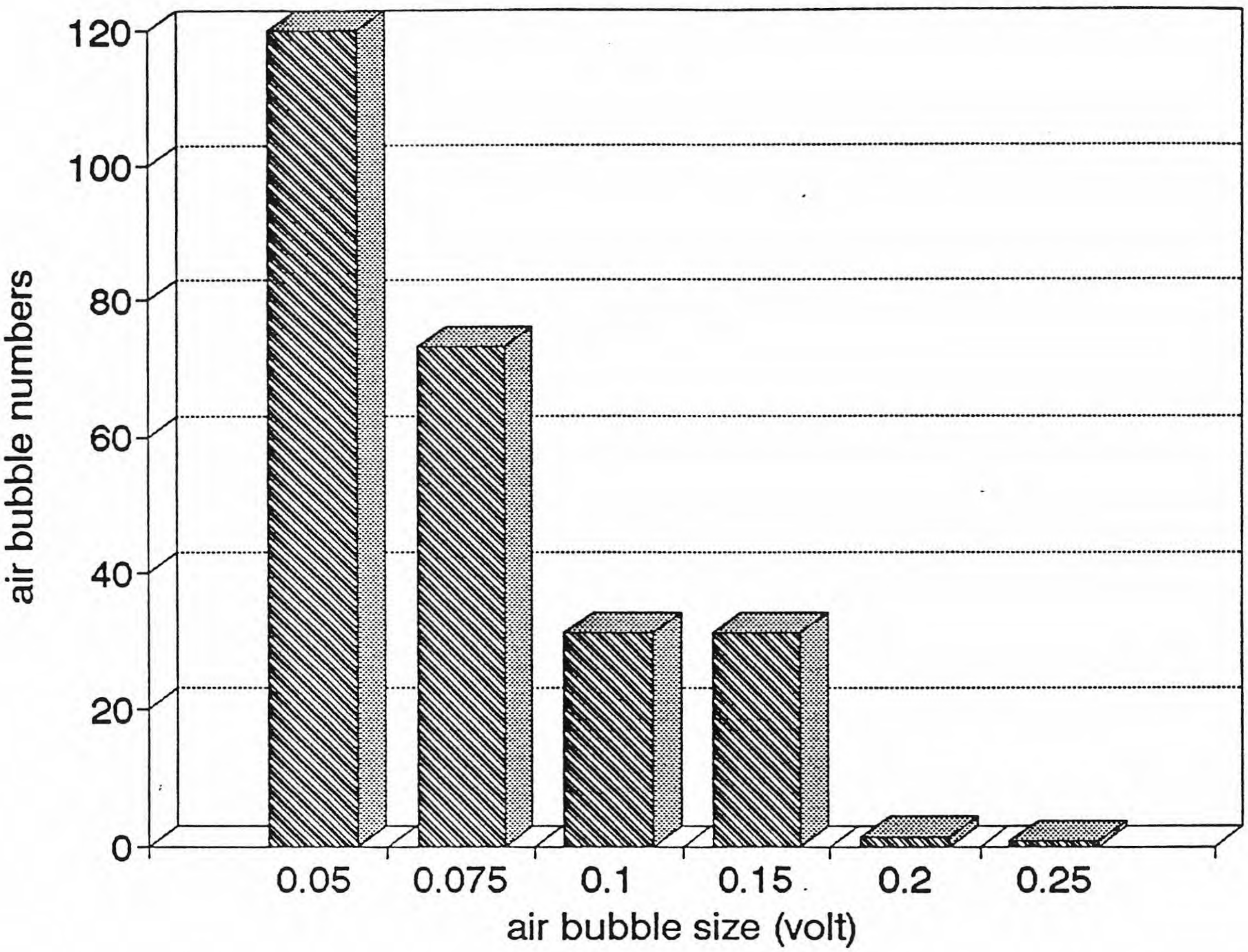
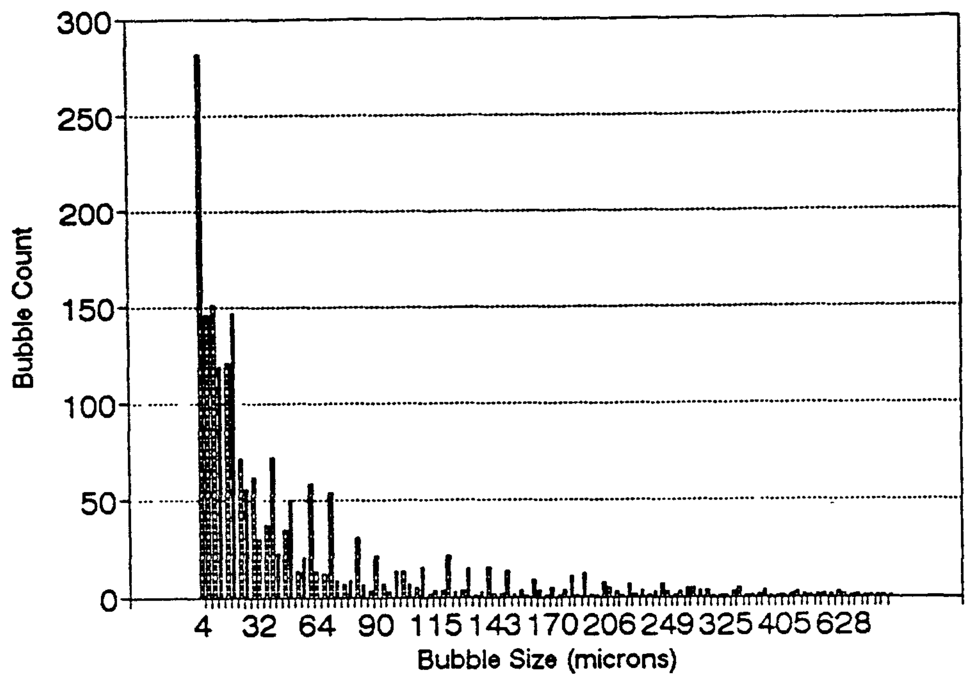


Figure 5.5 Air bubble size and numbers in concrete analysis using fiber optic method for test sample two.



Total Length of Traverse: 2553.6 mm (100.5 inches)
Number of Air Bubblee Intercepted: 975
Sum of Bubble Length: 52.571 mm

Air Void Content: 2.06%

Figure 5.6 Air bubble size and numbers in concrete analysis using microscopic method for test sample two.

Table 5.7 Air bubble number and size for test sample one

Data File Name	Air Bubble Number and Size					
	Very Very Small Air Bubble No.	Very Small Air Bubble No.	Small Air Bubble No.	Medium Air Bubble No.	Large Air Bubble No.	Very Large Air Bubble No.
2F1.WK1	117	70	45	37	2	0
2F2.WK1	126	62	65	29	1	2
2F3.WK1	105	81	56	36	2	0
2F4.WK1	111	66	67	35	0	0
2F5.WK1	109	79	58	37	1	0
2F6.WK1	115	72	44	36	4	0
2F7.WK1	130	66	60	29	1	0
2F8.WK1	106	89	40	28	1	3
2F9.WK1	141	74	47	22	1	2
2F10.WK1	136	69	46	25	0	0
Average	120	73	31	31	1.3	0.7

* Air percent: Fiber Optic: 2.32%, Pressure Meter: 2.2%

Table 5.8 Air bubble number and size for test sample two

Data File Name	Air Bubble Number and Size					
	Very Very Small Air Bubble No.	Very Small Air Bubble No.	Small Air Bubble No.	Medium Air Bubble No.	Large Air Bubble No.	Very Large Air Bubble No.
3F1.WK1	110	68	48	32	3	1
3F2.WK1	122	71	53	38	4	2
3F3.WK1	133	72	49	36	3	1
3F4.WK1	126	63	64	29	3	1
3F5.WK1	123	61	58	34	2	2
3F6.WK1	108	73	41	39	2	0
3F7.WK1	128	63	48	28	2	0
3F8.WK1	124	68	52	31	3	2
3F9.WK1	132	81	41	29	4	1
3F10.WK1	115	72	56	33	3	1
Average	122	69	51	33	3	1.1

* Air percent: Fiber Optic: 4.43%, Pressure Meter: 4.0%

5.3 Conclusions

Development of a hierarchical microcomputer based fiber optic air percent detection system for use in fresh concrete was presented.

1. A embedded controller mini system was developed in order to replace the microcomputer in the microcomputer based fiber optic sensor for use in fresh concrete. The new system can be more easily integrated in the field construction operations.

2. Communication is set up between microcontroller and IBM PC.

3. A menu driven and easily operation software package is developed in IBM PC for further analyzing air content signal and saving the test data.

The microcontroller system used in this system is an 8031. A more effective and powerful microcontroller of the 8096 family can be used. The MSC-96 family has an on chip A/D converter and also EPROM. we can make this system very small portable size suitable for on-site use. The mini system of 8096 is shown in Appendix F, Fig. F.2..

APPENDICES

Appendix A: Laser Diode Technical Data

A.1 TOLD-9200 Features

- Laser Wavelength : ÁP = 670 nm (typ.)
- Optical Output Power : P0 = 3 mW (CW)
- Provided with a PIN-PD for Monitoring

A.2 PIN Connection

Fig. A.1 shows the PIN connection of TOLD-9200 (Unit in mm).

A.3 Maximum Ratings

See Table A.1.

A.4 Optical-Electrical Characteristics

See Table A.2.

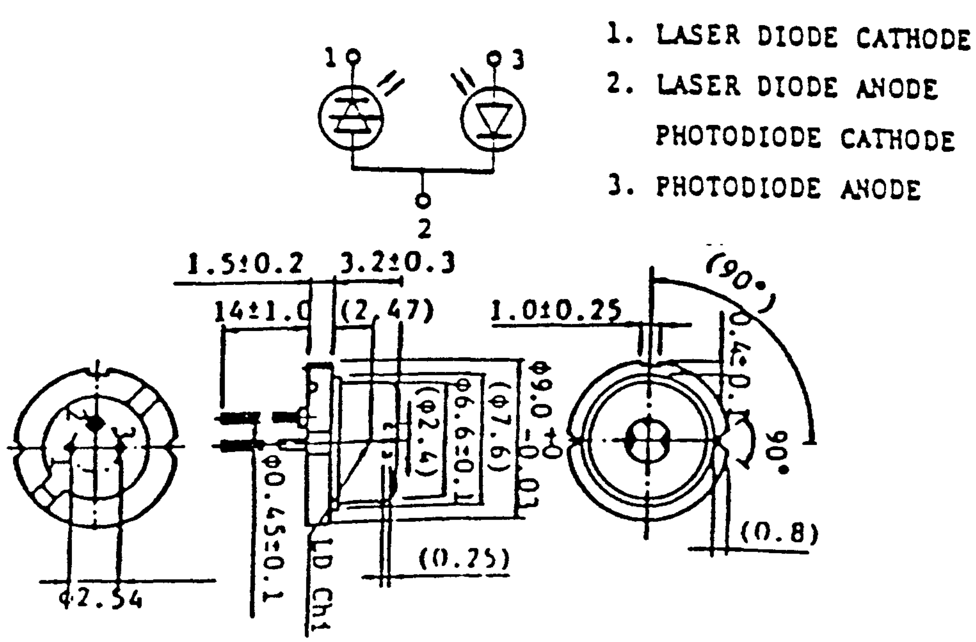


Figure A.1: PIN Connection of TOLD-9200

Table A.1: Maximum Ratings of TOLD-9200 (T_C=25°C).

CHARACTERISTIC	SYMBOL	RATING	UNIT
Optical Output Power	P_O	3	mW
LD Reverse Voltage	V_R(LD)	2	V
PD Reverse Voltage	V_R(PD)	30	V
Operation Case Temp.	T_C	-10-50	°C
Storage Temperature	T_{stg}	-40-85	°C

Table A.2: Optical-Electrical Characteristics of TOLD-9200 (T_C = 25 ° C).

CHARACTERISTIC	SYMBOL	TEST CONDITION	MIN.	TYP.	MAX	UNIT
Threshold Current	I_{th}	CW operating	-	76	90	mA
Operation Current	I_{op}	P_O = 2 mW	-	85	100	mA
Operation Voltage	V_{op}	P_O = 2 mW	-	2.3	3.0	V
Lasing Wavelength	α_p	P_O = 2 mW	-	670	680	nm
Beam Divergence	$\theta_{//}$	P_O = 2 mW	-	7	-	deg.
	$\theta_{\hat{U}}$	P_O = 2 mW	-	34	-	
Monitor Current	I_m	P_O = 2 mW	0.15	0.45	0.7	mA
PD Dark Current	I_D(PD)	V_R = 10V	-	-	100	nA
PD Total Capacitance	C_T(PD)	V_R = 10V f = 1MHz	-	-	20	pF

Appendix B

Laser Diode Control and Drive Circuit

B.1 Laser Diode Control and Drive Circuit

See Fig. B.1

B.2 Circuit Devices List

See Table B.1

Table B.1 : Circuit Device List.

Device Name	Symbol	Unit	Values	Other
Battery	BT	V	5.0	
Capacitance	C1	μ	20	
Zener Diode	ZD	V	2.4	NTE 5061A
Resistance	R1	Ω	470	
Transistor	Q1		2SA1015	NTE 290A
Transistor	Q2		2SC1959	NTE 85
Capacitance	C2	μ	10	
Resistance	R2	Ω	10	
Adjustable Resistance	RV	KΩ	3	
Capacitance	C3	μ	1	
Laser Diode	PD,LD			TOSHIBA TOLD 9200

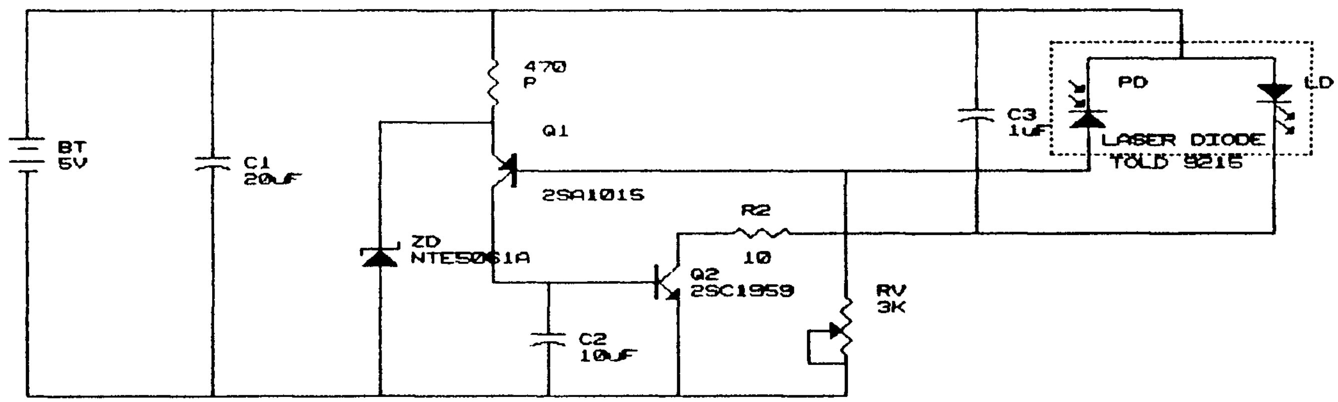


Figure B.1 Laser Diode Control and Driver Circuit.

Title LASER DIODE AND CONTROL CIRCUIT		
Size A	Document Number	REV
Date: March 12, 1992	Sheet	of

Appendix C

Laser to Fiber Source Coupler

Operation Principle

The source coupler utilizers the properties of lenses in combination with a novel tilt method to achieve submicron resolution. The method is based on precision control of the angle between the laser beam and receiver lens. The source coupler is comprised of two baseplates each having axial bore. One of the baseplates is adapted to receive a lens holder carries a lens and a fiber. The other baseplate is attached onto the laser. A resilient member such as a rubber O-ring is sandwiched between the baseplates. Threaded screws interconnect the two baseplates. The screws can then be adjusted by a screwdriver to alter the angular orientation of one baseplate relative to the other. The fiber output is monitored as the screws are adjusted until the detected output is optimized. The effects of angular misalignment are shown in Fig. B.1. If the lens is tilted by an angle Õ relative to the laser beam, the focused laser spot on the focal place of the lens will be displaced relative to the receiver lens axis by an amount given by the equation Z=ftanÕ, where f is the focal length of the lens. For 80 TPI screws, a lens with 1 mm focal length, and 20 mm lever arm = 0.1 micron.

The tilt angle is changed until the focused laser beam spot overlaps with the receiver fiber core. The coupler could also be used as variable attenuator by merely offsetting the focused beam with respect to the fiber core. In addition, this method also reduces the light coupled back to the laser cavity. This is done by placing the fiber off-center with respect to lens axis which will result in non-zero tilt angle between the laser beam and the lens axes.

Appendix D

Photodetector AF65-R2F Technical Data

D.1 Feature

The Ax65-R2F is a high quality low cost silicon photodiode specially designed for fiber optic applications operating in the 850 nm range. Frequency response of 100 MHz can be attained when used with maximum reverse bias voltage.

The photodiode is mounted in an hermetically sealed three pin TO-18 equivalent header with a low profile can. The cap contains a specially designed aperture facilitating easy and accurate fiber attachment and positioning. The photodiode active area of 1 mm^2 is larger than the aperture so that a diverging light beam emanating from the fiber is intercepted to a maximum degree. The chip is precisely centered and is positioned very close to the window. The photodiode is also electrically isolated from the case.

D.2 Absolute Maximum Ratings

* Storage Temperature : -55° to +125° C
* Operating Temperature : -55° to +120° C
* Active Element Dimensions : 1 mm * 1 mm
* Window Diameter : 0.8 mm
* Chip to Window Spacing : 0.2 to 0.7 mm
* Recommended Wavelength Range : 400 nm to 1000 nm
* High Frequency Response : up to 100 MHZ

D.3 PIN Connection

See Fig. D.1.

D.4 Electro-optical Characteristics

See Table D.1.

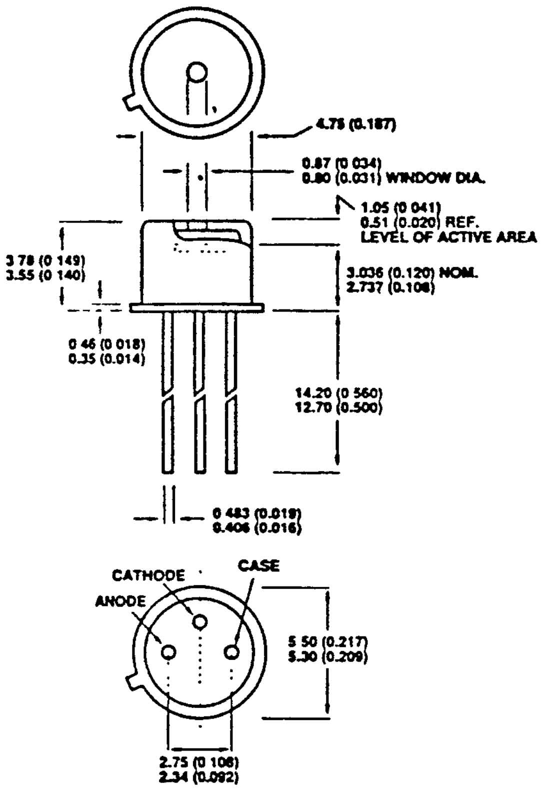


Figure D.1 Photodetector PIN Connection.

Table D.1 : Electro-Optical Characteristics

ELECTRO-OPTICAL CHARACTERISTICS*	MIN.	TYPICAL	MAX.	UNIT
Peak Sensitivity			850	nm
Operating Voltage			30	V
Peak DC Current			10	mA
Peak Pulse Current (1μs 1% duty cycle)			200	mA
Output Current Temperature Coefficient		0.15		%/°C
Responsivity at 900 nm	0.52	0.55		A/W
Risetime at 900 nm(10% to 70%) (V_r = 20 V)		2.5		ns
Capacitance (V_r = 0V)		15		pF
Capacitance (V_r = 20V)		3.5		pF
Dark Current (V_r = 20V)			5	nA
Noise Equivalent Power at 900 nm (V_r = 20v)		$3.3*10^{14}$		$WHz^{\frac{1}{2}}$

* All the parameters are characteristic of a photodiode operating at 23 c. and connected to a load resistance of 50 ohms (where appropriate).

Appendix E

Interface Circuit

E.1 I/O Interface Circuit

See Fig. E.1. Circuit devices list is shown in Table E.1.

E.2 Analog Interface Circuit

See Fig. E.2. Circuit devices list is shown in Table E.2.

E.3 UATR Interface Circuit

See Fig. E.3. Circuit devices list is shown in Table E.3.

Table E.1 I/O interface circuit devices list

Devices Name	Symbol	Unit	Values	Other
Button	SW1-SW4			
Digital IC	U1		74LS14	
Resistance	R1,R2,R4,R5,R7,R8 R10,R11	õ	100	
Resistance	R3,R6,R9,R12	kõ	4.7	
Capacitance	C1,C2,C4,C5,C7,C8 C10,C11	pF	1000	
Capacitance	C3,C6,C9,C12	μF	0.01	

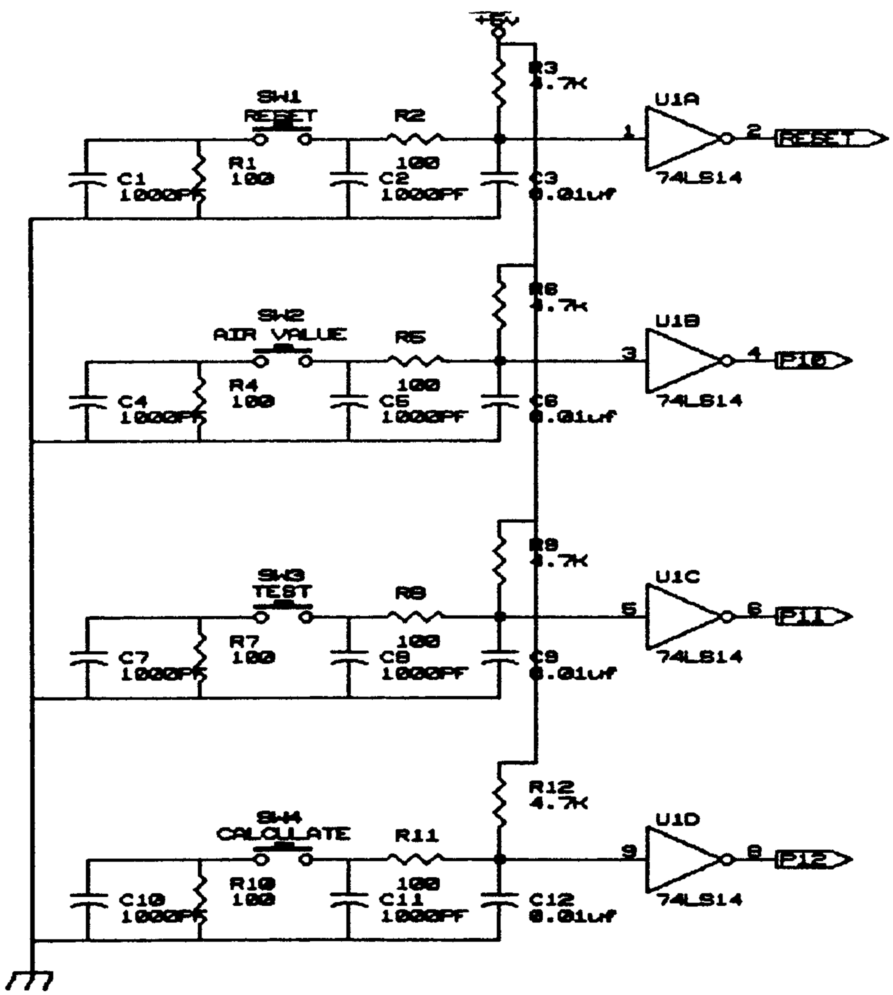


Figure E.1 I/O Interface Circuit.

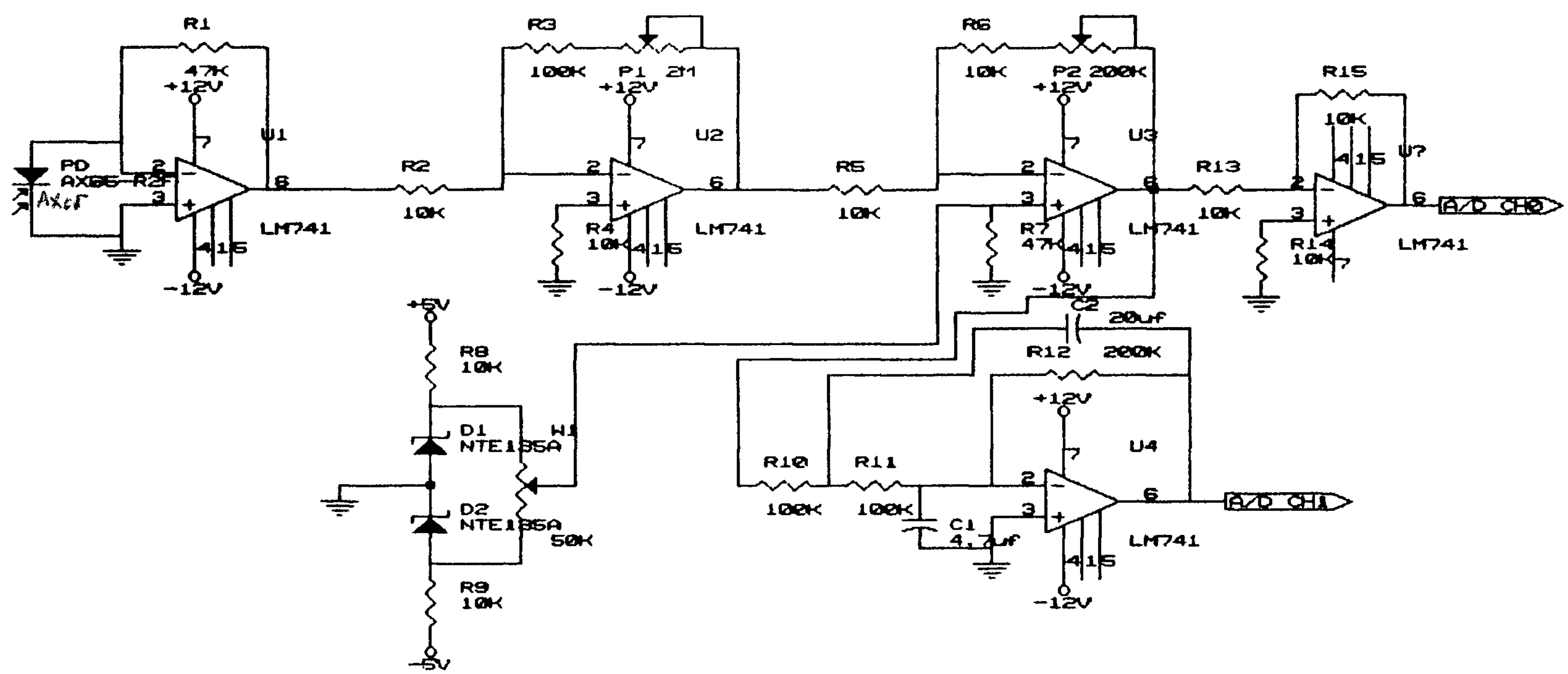


Figure E.2 Analog Interface Circuit.

ANALOG SIGNAL IN

Title

LASER DETECTOR

Size	Document Number	REV
A		

Date: March 13, 1992 Sheet 2 of 3

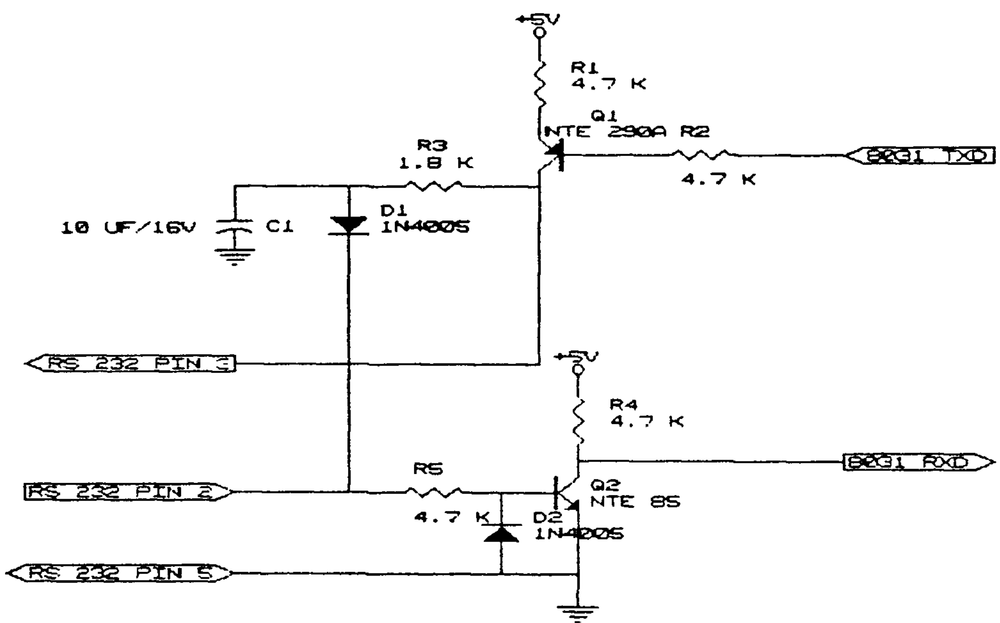


Figure E.3 UATR Interface Circuit.

Table E.2 Analog interface circuit devices list

Device Name	Symbol	Unit	Value	Other
Photodiode	PD		AX65-R2F	Centronic Inc.
Operational Amp.	U1-U5		UA741CP	
Zener Diode	ZD1,ZD2	V	5.1	NET135A
Resistance	R1,R7	Kõ	47	
Resistance	R2,R4,R5,R6,R8 R13,R14,R15	Kõ	10	
Resistance	R3,R10,R11	Kõ	100	
Resistance	R12	Kõ	200	
Capacitance	C1	μF	4.7	
Capacitance	C2	μF	20	
Adjustable R	P1	Mõ	0-2M	
Adjustable R	P2	Kõ	0-200	
Potentiometers	W1	Kõ	0-50	73JA-JB

Table E.3 UATR interface circuit devices list

Device Name	Symbol	Unit	Values	Other
Transistor	Q1		PNP	NTE290A
Transistor	Q2		NPN	NTE85
Diode	D1,D2			1N4005
Resistance	R1,R2,R4,R5	Kõ	4.7	
Resistance	R3	Kõ	1.8	
Capacitance	C1	μF	10	

Appendix F

Controller Circuit

F.1 Mini Controller Circuit

See Fig. F.1.

F.2 Circuit Devices List

See Table F.1

F.3 8096 Mini System Circuit

See Fig. F.2.

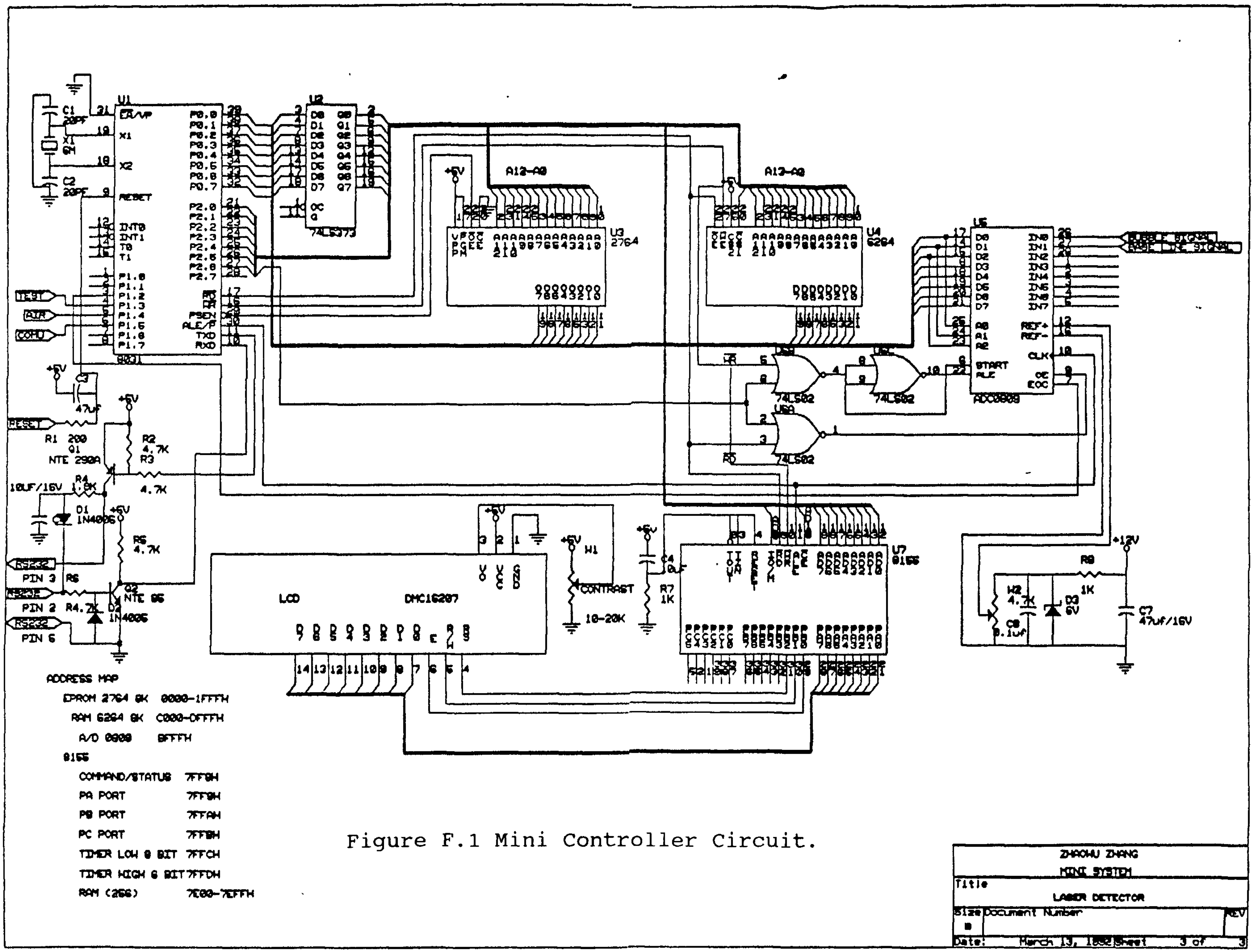


Figure F.1 Mini Controller Circuit.

Table F.1 Controller Circuit Devices list.

Device Name	Symbol	Unit	Value	Other
CPU	U1		8031	
Octal D-Type Latch	U2		74LS373	
RAM	U3	K	8	2764
EPROM	U4	K	8	6264
A/D convertor	U5		0809	
Quad 2-Input Pos NOR Gate	U6		74LS02	
Peripheral Interface	U7		8155	
Liquid Crystal Display	LCD		DMC16207	
Crystal	X1	MHZ	6	
Capacitance	C1,C2	PF	20	
Capacitance	C3,C6	μF	47	
Capacitance	C4	μF	10	
Capacitance	C5	μF	0.01	
Zener Diode	D1	V	5	
Resistance	R1	õ	200	
Resistance	R2,R3	Kõ	1	
Adjustable R	W1	Kõ	10	
Adjustable R	W2	Kõ	4.7	

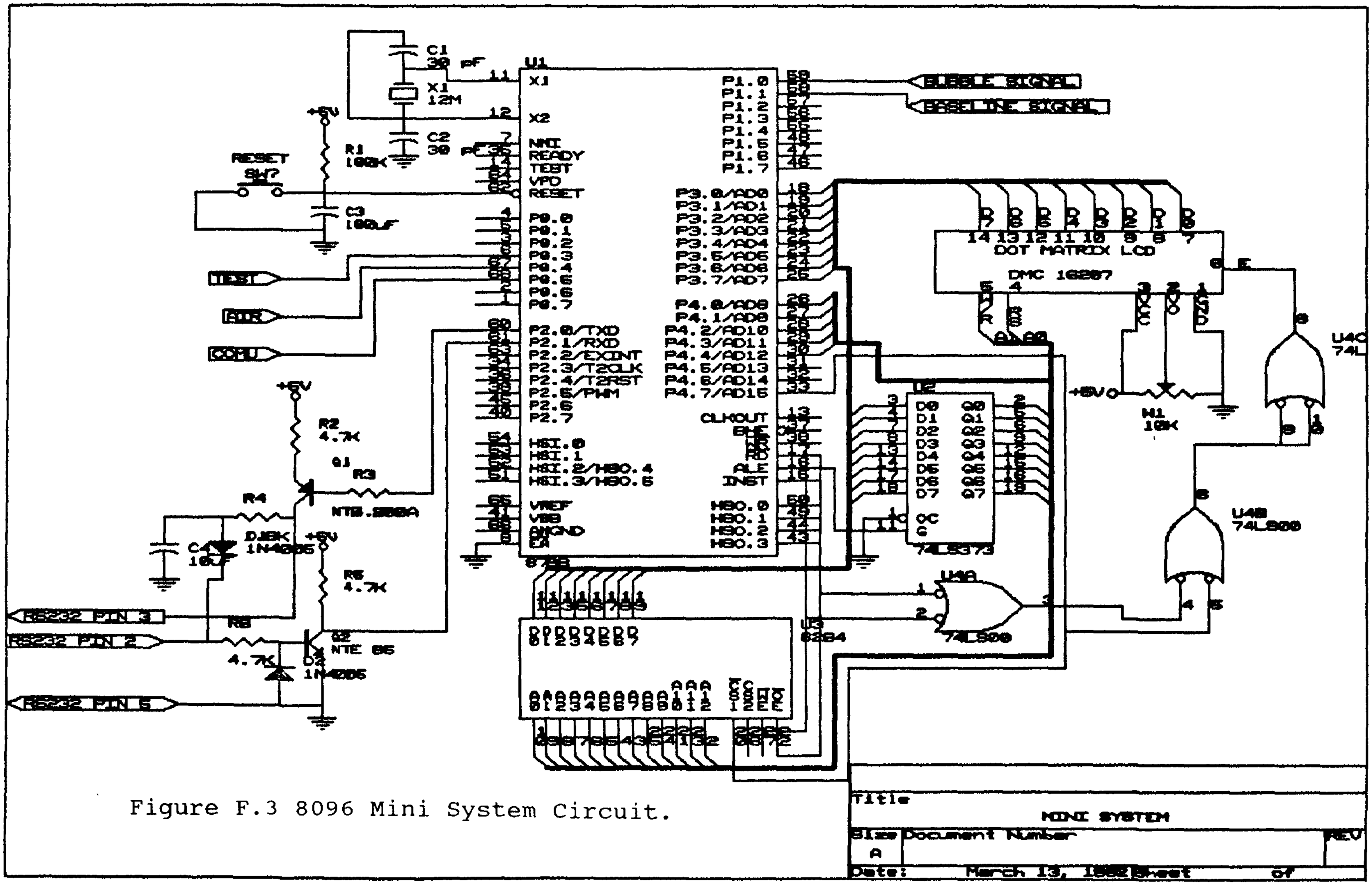


Figure F.3 8096 Mini System Circuit.

Appendix G

Program Flow Chart

G.1 Controller Main Program Flow Chart

See Fig. G.1.

G.2 Data Processing Subroutine Flow Chart

See Fig. G.2.

G.3 Division Subroutine Flow Chart

See Fig. G.3

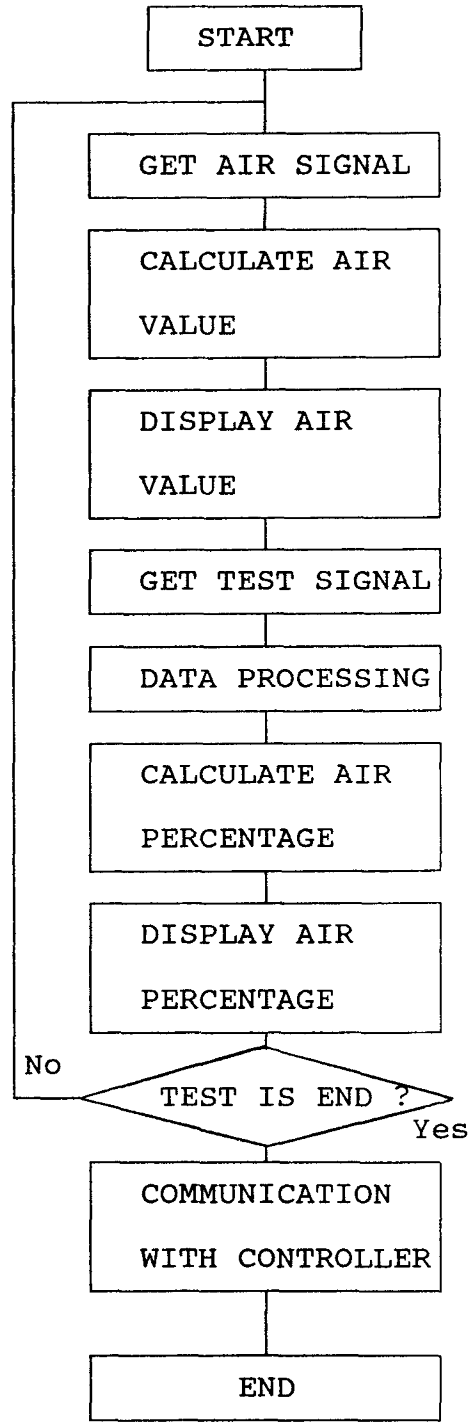


Figure G.1 Controller Main Program Flow Chart.

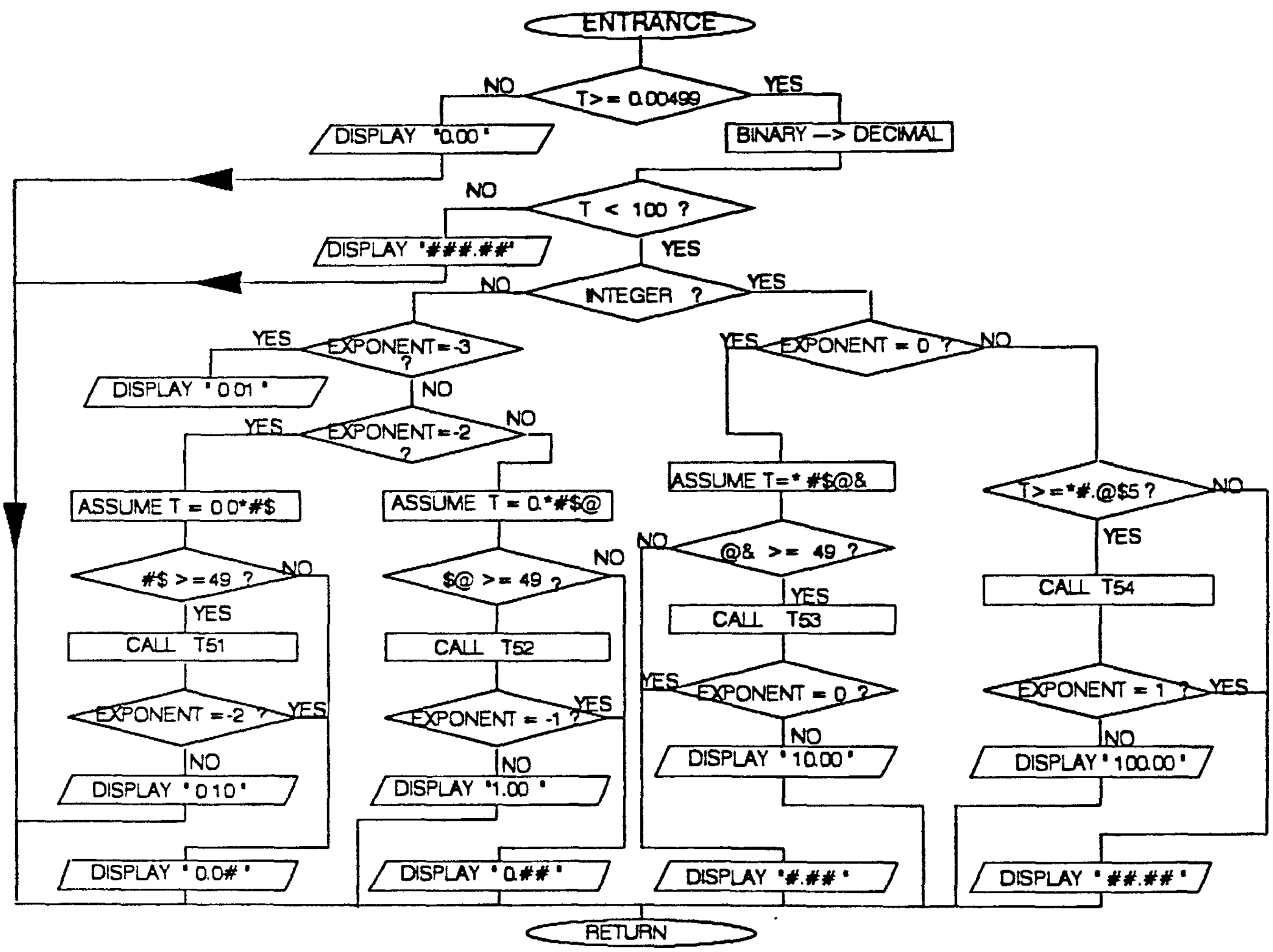


Figure G.2 Data Processing Subroutine Flow Chart.

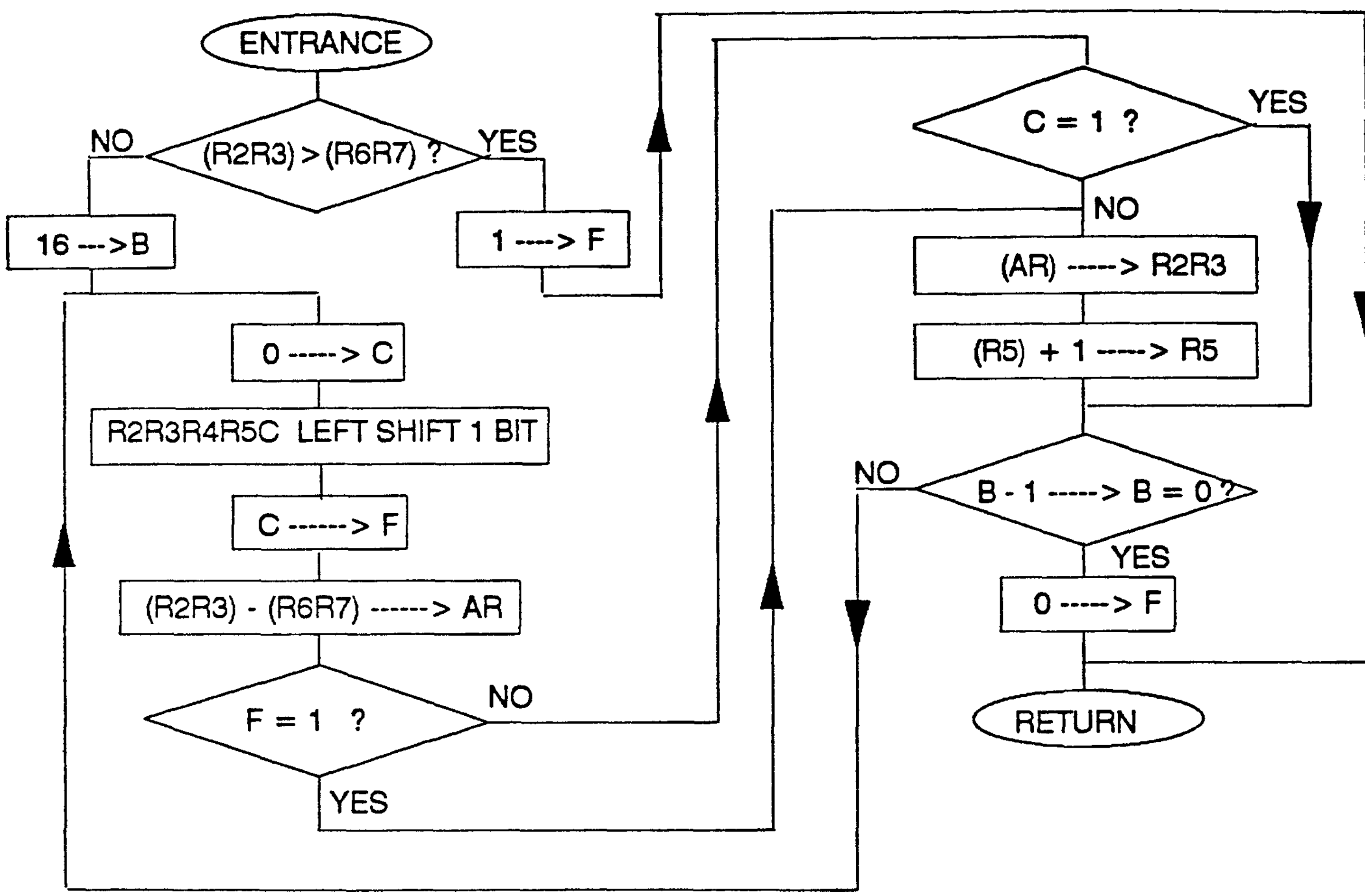


Figure G.3 Division Subroutine Flow Chart.

Appendix H

System Operation

H.1 Lab mode

In lab mode, user can select sampling frequency and running duration and see the sampling processing on screen.

All the experimental data are recorded using an eight channels data acquisition board DAS-8 which is IBM PC bus compatible [9] and the process is controlled by the master program. Steps involved in operation of the sensor and the system are outlined as follows:

Step 1 : Connect the sensor measuring device to the computer by a supplied cable. Turn on both power. Hold on sensor in the open air and adjust offset knob to relative value (3 - 5 V, read from voltage meter). Until it's stable, record the air value, which is the reflected light intensity amplitude in volts due to air. On computer, run master program. Screen will display as follows:

LASER DETECTOR

PROF. F. ANSARI

CIVIL DEPT. NJIT

Press any key to get main menu as shown in Fig. H.1 . Select <1> . According to the program requirement, enter sampling frequency, and running duration .

Step 2 : Enter real time data file name. After enter data file name, plunge the sensor into fresh concrete and then press enter key. At the same time, move the sensor onward in order to sense the presence of air bubbles at

```
CHOOSE FROM THE FOLLOWING :
  <1> - DO EXPERIMENT
  <2> - PRINT AN EXPERIMENT RESULTS
  <3> - VIEW ONE TEST RESULT
  <4> - CALCULATE A TEST RESULT AGAIN
  <5> - COMMUNICATION WITH SLAVE CONTROLLER
  <6> - EXIT TO BASIC
ENTER SELECTION NUMBER (1 - 6):
```

Figure H.1 Main menu

different location in concrete. Real time Plot of reflected light intensity versus time is displayed and the intensity values are saved in a file for further processing. When the data acquisition is finished, computer will inform user by a beep , display " data acquisition is OK " and require user to enter air value which was obtained earlier. As soon as the air value is entered the computer will process the data. Following the data processing stage, the screen will display the following:

Step 3 : If you want see the data , you can select <1>. Select <2>, the user will be prompted to enter the processed data file name. After few seconds, screen will display

The percentage of air entrainment is ##.## %

```
CHOOSE FROM THE FOLLOWING :

<1> - DISPLAY DATA ON SCREEN AND RETURN TO MENU

<2> - GENERATE A DATA FILE & PLOT USING THE DAS8 GRAPHICS
      PACKAGE

<3> - REPEAT THE TEST

<4> - CALCULATE ONCE AGAIN

<5> - CONTINUE DOING EXPERIMENT

<6> - FINISH EXPERIMENT

<7> - EXIT TO MAIN MENU

ENTER SELECTION NUMBER (1 - 7) :
```

Figure H.2 Lab mode submenu

the air bubble distribution and air percentage value. User will have the choice to save the graph or chain to other graph or return to the submenu.

If the test was not performed properly, the user can select <3> to repeat the test. All the data recorded by last test will be replaced by the new test.

If the user wants to calculate the results once again, option number <4> is selected. The new air value is entered and calculations are repeated. To continue performing more tests, select <5>, and repeat steps 2 and 3.

One example result is shown in Table 5.2 . The result will be saved in a result data file.

H.2 On - site Mode

A portable measuring device has been developed for on- site operation. In this mode, the sample frequency is fixed at 100 Hz, the running duration is 5 second and it only displays air value and air percent on LCD. Data can be transferred to an IBM PC for display and further analysis through an RS232 interface. The operation steps are outlined as follows:

Step 1 : Turn on the measuring device.

Step 2 : Hold the sensor in the open air. Press air value button. After 5 seconds LCD will display " air value is : *##.##* v ". If the value is too small (< 3V) or too large (>= 5v), please adjust the offset knob and repeat the above mentioned procedure once again.

Step 3 : Plunge the sensor into fresh concrete, press the test button. At the same time, move the sensor onward in order to sense the presence of air bubbles at different location in concrete. LCD will display the air content in five seconds.

If you want to do more test, repeat step 2, 3 . The device will save update 16 times (8k RAM).

For further analysis, upon completion of tests, the system can be interfaced with a IBM PC. Master program is then run. The main menu will be displayed (Fig.H.1) . Select

<5> - communication with slave controller. Screen will display " COMMUNICATION IS GOING ON " . Press "comu" button on the device. After few seconds Screen will display "air percentage is ##.## % " and submanu, as shown in Fig. H.3 .

Step 5 : To see the test data, select <1>. Data Pertaining to the air bubble signal will be displayed on screen. Then program will return to submenu.

CHOOSE FROM THE FOLLOWING :

<1> - DISPLAY DATA ON SCREEN AND RETURN TO MENU

<2> - GENERATE A DATA FILE & PLOT

<3> - CALCULATE ONCE AGAIN

<4> - CONTINUE COMMUNICATION

<5> - TERMINATE COMMUNICATION

<6> - EXIT TO MAIN MENU

ENTER SELECTION NUMBER (1 - 6) :

Figure H.3 Communication Menu

Select <2>. The user will be prompted to enter file name. The plot of air bubble distribution in concrete and the air percentage will be displayed on screen. The information will be saved in a file . The user will be able to save the graph or chain to other graph or return to communication menu.

If the user selects <3> - repeat calculation, air value will be required again for calculation of the air percent.

Step 4 : Select <4> - continue communication . Before this step, the user should save data by selecting <2>, otherwise the data will be lost. Press "comu" button on the device, and the screen will display Fig. H.3 .

Repeat step 4 until the screen displays " COMMUNICATION IS FINISHED, PLEASE SELECT <5>"

Step 5 : Select <5> - terminate communication. screen will display the result. One example result is shown in Table 5.2 . Results will be saved in a data file.

Appendix I

Program List

I.1 Microcontroller Program

I.2 Master IBM PC Program

I.1 Microcontroller Program

Main Program

```
      M0 EQU 6AH
      M1 EQU 6BH
      M2 EQU 6CH
      M3 EQU 6DH
      M4 EQU 6EH
      M5 EQU 6FH
      X  EQU 69H
ST1:  ORG 0
      LJMP STAR
      ORG 10
STAR:MOV SP,#2FH              ;Set stack pointer.
      MOV DPTR,#0F000H        ;Set data base pointer.
      PUSH DPL                ;Save the pointer.
      PUSH DPH
AIRV:MOV A,P1                 ;Air volue signal in.
      JB ACC.4,AIRV           ;No,wait.
      LCALL D20M              ;Yes,delay 20ms.
      MOV A,P1                ;Signal input once more.
      JB ACC.4,AIRV           ;No,go.
      DREL AT 7FH D           ;Define variables.
      DREL AT 79H B
      DREL AT 76H P
      DREL AT 73H T
      DBYTE AT 69H M
AIR1:MOV TMOD,#01H            ;Set sampling frequency.
      MOV TH0,#0ECH           ;f=100Hz.
      MOV TL0,#77H
      MOV TCON,#10H
TM1:  JNB TF0,TM1             ;Sampling time is available.
      MOV R2,#00H             ;Reset the accumulator.
      MOV R3,#00H
      MOV R4,#00H
      MOV R5,#00H
      ORL PSW,#18H
      MOV R7,#02H             ;Set sampling times.
TT1:  MOV R6,#0FAH            ;Number=500.
TT2:  ORL TL0,#77H            ;Start timer.
      MOV TH0,#0ECH
      MOV TCON,#10H
      ANL PSW,#0E7H
      MOV DPTR,#0BFFFH        ;Start A/D converter.
      MOV A,#00H              ;Channel 0
      MOVX @DPTR,A
      MOV R7,#0AH             ;Delay some time.
WE1:  DJNZ R7,WE1             ;
      MOV A,P1                ;Read A/D status.
WEND:JB ACC.2,WEND            ;Not finished,wait.
      MOVX A,@DPTR            ;Read A/D data.
      ADD A,R5                ;Add to accumulator.
```

```
      MOV R5,A
      JNC TM2
      MOV A,#01H
      ADD A,R4
      MOV R4,A
      JNC TM2
      MOV A,#01H
      ADD A,R3
      MOV R3,A
      JNC TM2
      INC R2
      LJMP TM2
TT3:  LJMP TT2                ;
TT4:  LJMP TT1
TM2:  JNB TF0,TM2             ;Sampling time is available?
      ORL PSW,#18H            ;Sampling times - 1
      DJNZ R6,TT3
      DJNZ R7,TT4
      ANL PSW,#0E7H           ;Calculating the air average
      MOV R6,#01H             ;value.
      MOV R7,#0F4H
      LCALL NDIV              ;Call division subroutine
      POP DPH                 ;Save air value.
      POP DPL
      MOV A,R5
      MOVX @DPTR,A
      INC DPTR
      PUSH DPL
      PUSH DPH
      MOV R2,#00H
      MOV R3,05H
      MOV R1,#7AH             ;
      CALL INTF               ;Call subroutine for converting
      MOV 7DH,7AH             ;integer to floating point number
      MOV 7EH,7CH
      MOV 7FH,7BH
      D=D*0.0196078           ;Data processing
      XR(7FF8H)=03H           ;LED initialization
      XR(7FFAH)=00H           ;
      MOV PSW,#18H
      MOV R6,#03H
TT6:  MOV PSW,#00H
      XR(7FF9H)=3CH
      XR(7FFAH)=01H
      LCALL D5M
      XR(7FFAH)=00H
      MOV PSW,#18H
      DJNZ R6,TT6
      MOV PSW,#00H
      XR(7FF9H)=0FH
      XR(7FFAH)=01H
      LCALL D5M
      XR(7FFAH)=00H
      XR(7FF9H)=01H
```

```
     XR(7FFAH)=01H
     LCALL D5M
     XR(7FFAH)=00H
     XR(7FF9H)=06H
     XR(7FFAH)=01H
     LCALL D5M
     XR(7FFAH)=04H
     XR(7FF9H)=82H
     XR(7FFAH)=01H
     LCALL D5M
     XR(7FFAH)=04H
     M=41H                ;Display "AIR VALUE IS :"
     LCALL ST2
     M=49H
     LCALL ST2
     M=52H
     LCALL ST2
     M=0A0H
     LCALL ST2
     M=56H
     LCALL ST2
     M=41H
     LCALL ST2
     M=4CH
     LCALL ST2
     M=55H
     LCALL ST2
     M=45H
     LCALL ST2
     M=0A0H
     LCALL ST2
     M=49H
     LCALL ST2
     M=53H
     LCALL ST2
     M=3AH
     LCALL ST2
     XR(7FF9H)=0C5H
     XR(7FFAH)=01H
     LCALL D5M
     XR(7FFAH)=04H
     T=D
     LCALL DPST           ;Call data processing subroutine.
     M=20H                ;Display "**.** V"
     LCALL ST2
     M=56H
     LCALL ST2
     XR(7FFAH)=00H
     XR(7FF9H)=0CH
     XR(7FFAH)=01H
     LCALL D5M
     XR(7FFAH)=00H
TES0:MOV A,P1             ;Input button signal.
     JB ACC.4,TES1        ;Get air value once more?
```

```
        LCALL D20M              ;No,go tes1
        MOV A,P1                ;Yes,go air1
        JB ACC.4,TES1
        LJMP AIR1
TES1:MOV A,P1                   ;Test signal in.
        JB ACC.3,TES0           ;Want to test ?
        LCALL D20M              ;No,wait.
        MOV A,P1
        JB ACC.3,TES0
        CLR PSW.3
        CLR PSW.4
        MOV R2,#00H             ;Reset integrator
        MOV R3,#00H
        MOV R4,#00H
        MOV R5,#00H
        CLR PSW.3
        SETB PSW.4
        MOV R2,#00H             ;Reset accumulator.
        MOV R3,#00H
        MOV R4,#00H
        MOV R5,#00H
        ANL PSW,#0E7H           ;Set sampling frequency.
        MOV TMOD,#01H           ;f=100Hz.
        MOV TH0,#0ECH
        MOV TL0,#77H
        MOV TCON,#10H
TES2:JNB TF0,TES2               ;Sampling time is available.
        ORL PSW,#18H
        MOV R7,#02H             ;Set sampling times.
TT7:    MOV R6,#0FAH            ;number=500.
TT8:    ANL PSW,#0E7H           ;Start sampling timer.
        ORL TL0,#77H
        MOV TH0,#0ECH
        MOV TCON,#10H
        MOV DPTR,#0BFFFH        ;Start A/D convertor
        MOV A,#00H              ;channel 0.
        MOVX @DPTR,A
        MOV R7,#0FH             ;Delay some time.
ADW:    DJNZ R7,ADW
        MOV A,P1                ;Read A/D status.
ADW1:JB ACC.2,ADW1              ;not finished,wait.
        MOVX A,@DPTR            ;Read A/D data.(bubble signal)
        MOV 60H,A
        MOV A,#01H              ;Start A/D channel 1.
        MOVX @DPTR,A
        MOV R7,#0FH             ;Delay some time.
ADW3:DJNZ R7,ADW3
        MOV A,P1                ;Read A/D status,
ADW2:JB ACC.2,ADW2              ;not finished, wait.
        MOVX A,@DPTR            ;Read A/D data. (base line)
        MOV 61H,A
        CLR PSW.3
        SETB PSW.4
        ADD A,R5                ;Accumulating base line data.
```

```
      MOV R5,A
      JNC BS1
      MOV A,#01H
      ADD A,R4
      MOV R4,A
      JNC BS1
      MOV A,#01H
      ADD A,R3
      MOV R3,A
      JNC BS1
      INC R2
BS1:  POP DPH                   ;Restore data base pointer.
      POP DPL
      MOV A,61H                 ;Comparing two signal.
      CLR C
      SUBB A,60H
      JC TES4
      MOV A,#00H                ;Save the air bubble signal
      MOVX @DPTR,A              ;to data base.
      INC DPTR
      PUSH DPL
      PUSH DPH
      LJMP TES9
TES4:CLR C
      MOV A,60H
      SUBB A,61H
      MOVX @DPTR,A
      INC DPTR                  ;Increasing the data base pointer
      PUSH DPL                  ;Saving the pointer.
      PUSH DPH
      ORL PSW,#18H              ;Integrating the air bubble
      CJNE R7,#02H,TES5         ;signal.
      CJNE R6,#0FAH,TES5
      LJMP TES6
TES5:CJNE R7,#01H,TES7
      CJNE R6,#00H,TES7
TES6:MOV A,60H
      CLR C
      SUBB A,61H
      CLR C
      RRC A
      LJMP TES8
TES7:MOV A,60H
      CLR C
      SUBB A,61H
TES8:CLR PSW.3
      CLR PSW.4
      ADD A,R5
      MOV R5,A
      JNC BB1
      MOV A,#01H
      ADD A,R4
      MOV R4,A
      JNC BB1
```

```
        MOV A,#01H
        ADD A,R3
        JNC BB1
        INC R2
BB1:    LJMP TES9
TT9:    LJMP TT7
TTA:    LJMP TT8
TES9:JNB TF0,TES9           ;Is the sampling time available?
        ORL PSW,#18         ;Yes,smapling next times.
        DJNZ R6,TTA
        DJNZ R7,TT9
        ANL PSW,#0E7H
        MOV 1DH,R5
        MOV 1CH,R4
        MOV 1BH,R3
        MOV 1AH,R2
        CLR PSW.3
        SETB PSW.4
        MOV R6,#01H         ;Calculating the average value
        MOV R7,#0F4H        ;of concrete.
        LCALL NDIV
        MOV A,R5
        POP DPH             ;Save the average value of
        POP DPL             ;concrete.
        MOVX @DPTR,A
        INC DPTR
        PUSH DPL
        PUSH DPH
        CLR PSW.3
        CLR PSW.4
        MOV R3,A
        MOV R2,#00H
        MOV R1,#7AH
        CALL INTF
        MOV 77H,7AH
        MOV 78H,7CH
        MOV 79H,7BH
        MOV R2,1AH
        MOV R3,1BH
        MOV R4,1CH
        MOV R5,1DH
        CLR C               ;
        MOV A,R2
        RRC A
        MOV R2,A
        MOV A,R3
        RRC A
        MOV R3,A
        MOV A,R4
        RRC A
        MOV R4,A
        MOV A,R5
        RRC A
        MOV R5,A
```

```
MOV 02H,04H
MOV 03H,05H
MOV R1,#7AH
CALL INTF
MOV 74H,7AH
MOV 75H,7CH
MOV 76H,7BH
B=B*0.0196078           ;Calculating air percentage
P=P*0.0392              ;P=P*0.0196078*2
D=D-B
P=P/D
P=P*0.2
XR(7FFAH)=04H           ;Display "AIR PERCENTAGE IS :".
XR(7FF9H)=80H
XR(7FFAH)=01H
LCALL D5M
XR(7FFAH)=04H
M=41H
LCALL ST2
M=49H
LCALL ST2
M=52H
LCALL ST2
M=20H
LCALL ST2
M=50H
LCALL ST2
M=45H
LCALL ST2
M=52H
LCALL ST2
M=43H
LCALL ST2
M=45H
LCALL ST2
M=4EH
LCALL ST2
M=54H
LCALL ST2
M=41H
LCALL ST2
M=47H
LCALL ST2
M=45H
LCALL ST2
M=3AH
LCALL ST2
XR(7FF9H)=0C5H
XR(7FFAH)=01H
LCALL D5M
XR(7FFAH)=04H
T=P
LCALL DPST              ;Call data processing subroutine.
M=20H
```

```
      LCALL ST2
      M=25H
      LCALL ST2
      XR(7FFAH)=00H
      XR(7FF9H)=0CH
      XR(7FFAH)=01H
      LCALL D5M
      XR(7FFAH)=00H
      MOV R2,#04H           ;Set counter.
COM0:MOV A,P1               ;Button signal in.
      JB ACC.4,COM1         ;Test ?
      LCALL D20M            ;No,go com1
      MOV A,P1              ;Yes,go air1.
      JB ACC.4,COM1
      LJMP CM1
CMM:  LJMP AIR1
CM1:  POP DPH               ;Is RAM full?
      MOV A,DPH
      PUSH DPH
      CJNE A,#0F7H,CMM      ;No,go air1.
      JC CMM
      POP DPH
      POP DPL
      MOV DPTR,#0F000H      ;Yes,restore pointer.
      PUSH DPL
      PUSH DPH
      LJMP AIR1             ;Go to air1
COM1:MOV A,P1               ;Button signal in.
      JB ACC.5,COM0         ;Communication to IBM PC?
      LCALL D20M            ;No,go com0.
      MOV A,P1
      JB ACC.5,COM0
      MOV TCON,#70H
      MOV TMOD,#21H
      MOV SCON,#040H        ;Yes,set communication mode 2.
      MOV TH1,#0E8H         ;Baud rate 1200 bit/s.
      MOV TL1,#0E8H
      MOV TMOD,#20H
      MOV PCON,#00H
      CJNE R2,#04H,COM3     ;If it's the first time,go com2
      MOVE DPTR,#0F000H
COM3:MOV R0,#02H            ;Set data numbers(502)
COM4:MOV R1,#0FBH
LOOP:MOVX A,@DPTR           ;Send data.
      MOV SBUF,A
COM5:JBC TI,CONT            ;Master computer don't pick up
      SJMP COM5             ;data,wait.
CONT:INC DPTR               ;Increase the pointer.
      DJNZ R1,LOOP          ;Communication is not finished,
      DJNZ R0,COM4          ;continue.
      DJNZ R2,COM0          ;If it does not reach 4 times,go
ABC:  LJMP ABC
```

Data Processing Subroutine

* This subroutine converts the floating point number to ASCII code and display the number on the LED.

Entrance: T (floating point number)

```
DPST:IF T>=0.00499 THEN OKI;
     M=30H                   ;If t<0.00499 ,display 0.00.
     LCALL ST2
     M=2EH
     LCALL ST2
     M=30H
     LCALL ST2
     M=30H
     LCALL ST2
     LJMP RE0                ;Return
OKI: XD(6AH)=BTD1(T)         ;Converting Binary floating point
     IF T<100 THEN OK2       ;number to decimal floating point
     MOV A,M1                ;number.
     ADD A,#30H              ;If t>=100 display "***.**"
     MOV X,A
     LCALL ST2
     MOV A,M2
     ADD A,#30H
     MOV X,A
     LCALL ST2
     MOV A,M3
     ADD A,#30H
     MOV X,A
     LCALL ST2
     M=2EH
     LCALL ST2
     MOV A,M4
     ADD A,#30H
     MOV X,A
     LCALL ST2
     MOV A,M5
     ADD A,#30H
     MOV X,A
     LCALL ST2
     LJMP RE0                ;Return.
OK2: MOV A,M0                ;If t is a integer then go
     JNB ACC.7,DSS
     CJNE A,#0FDH,DD2        ;If the exponent of t is not
     M=30H                    equit to -3 , go.
     LCALL ST2               ;If the exponent of t is equit
     M=2EH                   ;-3,its means 0.0049<t<0.0099
     LCALL ST2               ;display "0.01".
     M=30H
     LCALL ST2
     M=31H
     LCALL ST2
     LJMP RE0                ;Return.
```

```
DSS: LJMP D54
DD2: DJNE A,#0FEH,D53       ;If the exponent of t is not
     MOV A,M2               ;equit to -2,go.
     CJNE A,#05H,DB1        ;Assume t=0.00*#,if *<>5,go.
     LJMP DB2               ;If *=5,go DB2
DB1: JNC DB2                ;If *>5,go DB2
     CJNE A,#04H,DB3        ;If *<>4,go DB3
     MOV A,M3               ;If #<>9,go DB3
     CJNE A,#09H,DB3        ;If t>=0.0049 call carry
DB2: LCALL T51              ;subroutine(T51)
DB3: M=30H                  ;Display "0.
     LCALL ST2
     M=2EH
     LCALL ST2
     MOV A,M0               ;
     CJNE A,#0FEH,DB4       ;If exponent of t is not equit
     M=30H                  ;to -2,go DB4.
     LCALL ST2              ;Display "0.0*"
     MOV A,M1
     ADD A,#30H
     MOV X,A
     LCALL ST2
     LJMP RE0               ;Return.
DB4: M=31H                  ;Display "0.10"
     LCALL ST2
     M=30H
     LCALL ST2
     LJMP RE0               ;Return.
D53: MOV A,M3               ;Assume t=0.*#$@
     CJNE A,#05H,DC1        ;If $<>5,go DC1;
     LJMP DC2               ;If $=5 ,go DC2.
DC1: JNC DC2                ;If $>5 ,go DC2.
     CJNE A,#04H,DC3        ;If $<>4,go DC3;
     MOV A,M4               ;If $=4,check @=9 ?
     CJNE A,#09H,DC3        ;@<>9,go DC3.
DC2: LCALL T52              ;Call carry subroutine.
DC3: MOV A,M0               ;If exponent is not equit
     CJNE A,#0FFH,DC4       ;to -1 ,go DC4.
     M=30H                  ;Display "0.**".
     LCALL ST2
     M=2EH
     LCALL ST2
     MOV A,M1
     ADD A,#30H
     MOV X,A
     LCALL ST2
     MOV A,M2
     ADD A,#30H
     MOV X,A
     LCALL ST2
     LJMP RE0               ;Return.
DC4: MOV A,M0               ;Display "1.00"
     M=31H
     LCALL ST2
```

```
      M=2EH
      LCALL ST2
      M=30H
      LCALL ST2
      M=30H
      LCALL ST2
      LJMP RE0                  ;Return.
D54:  MOV A,M0                  ;If the exponent of t is not
      CJNE A,#00H,DDX           ;equit 0,go DDX.
      MOV A,M4                  ;Assume t=*.#@$&
      CJNE A,#05H,DU1           ;If $<>5,go DU1;
      LJMP DU2                  ;If $=5 ,go DU2.
DU1:  JNC DU2                   ;If $>5 ,go DU2.
      CJNE A,#04H,DU3           ;If $<>4,go DU3.
      MOV A,M5                  ;If $=4,check &.
      CJNE A,#09H,DU3           ;If &<>9,go DU3.
DU2:  LCALL T53                 ;If t>=*.#@49,call carry
DU3:  MOV A,M0                  ;subroutine.
      CJNE A,#00H,DU4           ;If the exponent of t is not
      MOV A,M1                  ;equit 0,go DU4.
      ADD A,#30H                ;Display "*.**"
      MOV X,A
      LCALL ST2
      M=2EH
      LCALL ST2
      MOV A,M2
      ADD A,#30H
      MOV X,A
      LCALL ST2
      MOV A,M3
      ADD A,#30H
      MOV X,A
      LCALL ST2
      LJMP RE0                  ;Return.
DDX:  LJMP D55                  ;
DU4:  M=31H                     ;Display "1.00"
      LCALL ST2
      M=30H
      LCALL ST2
      M=2EH
      LCALL ST2
      M=30H
      LCALL ST2
      M=30H
      LCALL ST2
      LJMP RE0                  ;Return.
D55:  MOV A,M5                  ;Assume t=*#.@$&
      CJNE A,#05H,DE1           ;If &<>5, go DE1.
      LJMP DE2                  ;If &=5, go DE2.
DE1:  JC DE3                    ;If &<5, go DE3
DE2:  LCALL T54                 ;If t>=*#.@$5, call carry
      MOV A,M0                  ;subroutine.
      CJNE A,#01H,DE4           ;If the exponent of t<>1, go DE4.
DE3:  MOV A,M1                  ;Display "**.**"
```

```
        ADD A,#30H
        MOV X,A
        LCALL ST2
        MOV A,M2
        ADD A,#30H
        MOV X,A
        LCALL ST2
        M=2EH
        LCALL ST2
        MOV A,M3
        ADD A,#30H
        MOV X,A
        LCALL ST2
        MOV A,M4
        ADD A,#30H
        MOV X,A
        LCALL ST2
        LJMP RE0                ;Return.
DE4:    M=31H                   ;Display "100.00"
        LCALL ST2
        M=30H
        LCALL ST2
        M=30H
        LCALL ST2
        M=2EH
        LCALL ST2
        M=30H
        LCALL ST2
        M=30H
        LCALL ST2
RE0:    RET                     ;Return.
```

Carry Subroutine

```
T54:    INC M4
        MOV A,M4
        CJNE A,#0AH,RE4
        MOV M4,#00H
        LCALL T53
RE4:    RET
T53:    INC M3
        MOV A,M3
        CJNE A,#0AH,RE3
        MOV M3,#00H
        LCALL T52
RE3:    RET
T52:    INC M2
        MOV A,M2
        CJNE A,#0AH,RE2
        MOV M2,#00H
        LCALL T51
RE2:    RET
T51:    INC M1
```

```
     MOV A,M1
     CJNE A,#0AH,RE1
     MOV M1,#00H
     INC M0
RE1: RET
```

Display Subroutine

```
ST2: XR(7FF9H)=M
     XR(7FFAH)=05H
     LCALL D5M
     XR(7FFAH)=04H
     RET
```

Delay 5 ms Subroutine

```
D5M: MOV R7,#0AH
DMS1:MOV R6,#0FFH
DMS2:DJNZ R6,DMS2
     DJNZ R7,DMS1
     RET
```

Delay 20 ms Subroutine

```
D20M:MOV R7,#28H
DM1: MOV R6,#0FFH
DM2: DJNZ R6,DM2
     DJNZ R7,DM1
     RET
```

Division Subroutine

This subroutine is for fixed binary number division.

```
Entrance: (R2R3R4R5)=divided number.
          (R6R7)=dividing number.
Return:   (R4R5)=quotient.
          (R2R3)=remainder.
```

```
NDIV:MOV A,R3               ;Is it overflow?
     CLR C
     SUBB A,R7
     MOV A,R2
     SUBB A,R6
     JNC NDVE
     MOV B,#10H             ;No, executing division.
NDVL:CLR C                  ;Shift left for divided number.
     MOV A,R5
     RLC A
     MOV R5,A
     MOV A,R4
     RLC A
     MOV R4,A
     MOV A,R3
```

```
     RLC A
     MOV R3,A
     XCH A,R2
     RLC A
     XCH A,R2
     MOV F0,C                ;save the MSB
     CLR C
     SUBB A,R7               ;Comparing partial remainder with
     MOV R1,A                ;dividing.
     MOV A,R2
     SUBB A,R6
     JB F0,NDVM
     JC NDVD
NDVM:MOV R2,A                ;Executing the subtraction.
     MOV A,R1
     MOV R3,A
     INC R5                  ;Add 1 to quotient.
NDVD:DJNZ B,NDVL             ;Executing 16 times.
     CLR F0                  ;Normal return.
     RET
NDVE:SETB F0                 ;Overflow return.
     RET
```

Two fixing-point integer transfer to floating-point number subroutine (INTF):

Function : (R2R3) is transferred to 3 bytes floating-point number pointed by (R1).
R2R3 is a two-byte integer (original code), bit 3CH is its sign.
Running time : Depend on number, about several ten to several hundred μs.
Stack number : 4 levels.
Length : 18 bytes.

```
INFT:MOV   R6,#10H
     SETB  C
     CLR   D5H           ;Clear F0 flag.
     LCALL FSDT          ;Call Regularize subroutine.
     MOV   A,R6          ;
     MOV   C,3CH
     MOV   E7H,C
     MOV   R4,A
     LCALL FSTR          ;Call result return subroutine.
     RET
```

Three-byte floating-point number regularizing subroutine (FSDT) :

Entrance : CY=0, right regularizing 1 bit, entering value is stored at bit 39H.
F0=0, right regularizing to R6(exponent)R2R3.
F0=1, right regularizing to R7(exponent)R4R5.
CY=1, left regularizing, until

R6(exponent)R2R3 is became regularized number or zero. F0 is the first entering value (later entering is zero). A Register is used.

Exit : C is output value of right regularized. The sign of R6, R7(exponent)(MSB) are unvalued.

Running time: depend on the number and operation, about from several ten to several hundred μs.

Stack number: 2 levels.

Length : 46 bytes.

```
FSDT:JC     LRG             ;CY=1, GO LRG.
     MOV    C, 39H          ;
     JB     D5H, RG7        ;f0=1, GO RG7.
     MOV    A, R2
     RRC    A
     MOV    R2, A
     MOV    A, R3
     RRC    A
     MOV    R3, A
     INC    R6
     RET
RG7 :MOV    A, R4
     RRC    A
     MOV    R4, A
     MOV    A, R5
     RRC    A
     MOV    R5, A
     INC    R7
RETN:RET
LRG :MOV    A, R2
     JNZ    LRG1
     CJNE   R3, #00H, LRG2
     MOV    R6, #42H
     RET
     JB     E7H, RETN
     MOV    C, D5H
     MOV    A,R3
     RLC    A
     MOV    R3, A
     MOV    A, R2
     RLC    A
     MOV    R2, A
     CLR    D5H
     DEC    R6
     SJMP   LRG
```

Three-byte floating-point number result returning subroutine (FSTR) :

Function : Transfer R4(exponent)R2R3 to memory which is pointed by (R1).
A register is used, and don't change recent working register area.

Running time : 12 μs.

Stack number : 2 levels.
Length : 11 bytes.

```
FSTR:MOV    A, R4
     MOV    @R1, A
     INC    R1
     MOV    A, R2
     MOV    @R1, A
     INC    R1
     MOV    A, R3
     MOV    @R1, A
     DEC    R1
     DEC    R1
     RET
```

I.2 Master IBM PC Program

```
10   '*****************************************************
20   '*                                                   *
30   '*Program for LASER DETECTOR using in plastic concrete*
40   '*                                                   *
50   '*                                  ZHAOWU ZHANG     *
60   '*****************************************************
70   'The bubble signal input to DAS-8 channel #0 and the
75   'baseline signal input to channel #1.
80   'Pin 6 (Counter 2 out) should be connected to the
      digital input #1 (pin 25)
90   'on the rear connector.
100  '
110  '---STEP 1: Contract BASIC workspace, load DAS8.BIN and
      initialize ---
120  '
130  'See LOADCALL>BAS for a fuller explanation of step 1
140  'and an alternative way of loading outside workspace.
150  '
160  CLEAR,49152!
170  DEF SEG = 0
180  SG = 256 * PEEK(&H511) + PEEK(&H510)
190  SG = SG + 49152!/16
200  DEF SEG = SG
210  BLOAD "DAS8.BIN", 0
220  OPEN "DAS8.ADR" FOR INPUT AS #1
230  INPUT #1,BASADR% 'Initialize & Declare CALL parameters.
240  CLOSE #1
250  DAS8 = 0
260  FLAG% = 0
270  MD% = 0          'mODE 0 = INITIALIZATION.
280  CALL DAS8 (MD%,BASADR%,FLAG%)
290  IF FLAG% <> 0 THEN PRINT "Installation error"
300  CLS:KEY OFF
310  SCREEN 1,0,0
320  LOCATE 11,15:PRINT "LASER DETECTOR"
330  LOCATE 18,22:PRINT "PROF. F. ANSARI"
340  LOCATE 22,22:PRINT "CIVIL DEPT. NJIT"
350  A$ = INKEY$:IF A$ = "" GOTO 350
360  CLS: SCREEN 2
370  DIM X(100),Y(100),F$(15)
380  CLS: LOCATE 4,1:PRINT "CHOOSE FROM THE FOLLOWING:"
390  PRINT :PRINT "  <1> - DO EXPERIMENT"
400  PRINT :PRINT "  <2> - PRINT AN EXPERIMENT RESULTS"
410  PRINT :PRINT "  <3> - VIEW ONE TEST RESULT"
420  PRINT :PRINT "  <4> - CALCULATE A TEST RESULT AGAIN"
430  PRINT :PRINT "  <5> - COMMUNICATION WITH SLAVE
                          MICROCONTROLLER"
440  PRINT :PRINT "  <6> - EXIT FROM MAIN MENU"
450  PRINT : PRINT "ENTER SELECTION NUMBER (1-6):";
460  A$ = INKEY$ : IF A$ = "" GOTO 460
470  PRINT A$
```

```
480  IF VAL(A$) = 1 THEN SIGN = 2 : GOTO 540
490  IF VAL(A$) = 2 GOTO 6150
500  IF VAL(A$) = 3 THEN SIGN = 1 : GOTO 3660
510  IF VAL(A$) = 4 GOTO 6450
520  IF VAL(A$) = 5 THEN SIGN = 3 : GOTO 6760
530  IF VAL(A$) = 6 THEN CLS:END
540  Q% = 0
550  '--- STEP 2: SET SAMPLE RATE AND CHECK SUFFICIENT
          MEMORY -------------------------------------
560  CLS
570  INPUT "Enter desired sample rate (samples/sec) :",F
580  PRINT
590  ' OUTPUT FREQUENCY = 2386.4/N KHz
600  N = 2385.4 * 1000 / 1000 :N = INT(N + 0.5)
610  IF N<2 OR N>65535! THEN PRINT "Warning! A sample rate
     of";F;"samples/sec is outside the range of counter 2"
     :goto 570
620  INPUT "DURATION OF SCAN (IN SECOND) :",DS
630  PRINT
640  'Translate duration in a number of conversions for mode
      5
650  'Number of conversions = duration * sample rate
660  NC = DS * F
670  '--- CHECK THERE IS ENOUGH MEMORY TO HOLD THIS ARRAY
680  IF (FRE(0) - 2000 - NC * 5) < 0 THEN PRINT "WARNING
     THERE IS INADEQUATE MEMORY WITHIN BASIC TO HOLD THIS
     DATA":PRINT "RE-RUN PROGRAM:END
690  DIM A(NC,2),B(NC),U(NC),P(NC)
700  INPUT "LOTUS FILE NAME :(AUTOMATIC .W EXT.):";D$
710  D$ = D$ + ".W"
720  Q% = Q% + 1
730  '
740  '--- STEP 3 SET MODE 5 GOING AND ACQUIRE DATA ---------
750  'PRINT TIME$
760  '--- SET UP COUNTER 2 IN PULSE ON TERMINAL COUNT ------
770  MD% = 10  'Mode 10 for setting counter configuration
780  D%(0) = 2 'Operate on counter #2
790  D%(1) = 3 'Configuration #2 = Square wave generator
800  CALL DAS8 (MD%,D%(0),FLAG%)
810  IF FLAG% <>  0 THEN PRINT "Error in setting counter 2
     configuration":stop
820  MD% = 11  'Mode 11 to load counter #2
830  D%(0) = 2 'Operate on counter #2
840  IF N < 32767 THEN N1 = N ELSE N1 = N - 65536! 'Correct
                                              for integer
850  D%(1) = N1
860  CALL DAS8 (MD%, D%(0), FLAG%)
870  IF FLAG% <> 0 THEN PRINT "Error in loading counter # 2"
     STOP
880  MD% = 10    'Mode 10 for setting counter configuration
890  D%(0) = 1   'Operate on counter #1
900  D%(1) = 0   'Configuration #1 = pulse on terminal count
910  CALL DAS8 (MD%, D%(0), FLAF%)
920  IF FLAG% <> 0 THEN PRINT "Error in setting counter 1
```

```
     configuration":stop
930  MD% = 11     'Mode 11 to load counter #1
940  D%(0) = 1    'Operate on counter #1
950  F1 = 2000 / F :F1 = INT(F1 + 0.5)
960  D%(1) = F1
970  CALL DAS8 (MD%, D%(0), FLAG%)
980  IF FLAG% <> 0 THEN PRINT "Error in loading counter 1"
     :STOP
990  KEY OFF : SCREEN 0,0,0
1000 OX = 70 : OY = 0 : QX = 1. 8: QY = 0.9
1010 CLS:COLOR 7,0,0 : SCREEN 2
1020 Y1 = 0 : Y2 = 4!
1030 X1 = 0 : X2 = DS
1040 DX = ABS(CINT((X2 - X1) / 25) + 0.5))
1050 DY = ABS(CINT(((Y2 - Y1) / 25) + 0.5))
1060 SX = 260 / (X2 - X1)
1070 SY = 140 / (Y2 - Y1)
1080 IF Y2 <= 0 THEN YA = 10 : GOTO 1110
1090 IF Y1 => 0 THEN YA = 150 : GOTO 1110
1100 YA = 10 + SY * Y2
1110 '
1120 'Set the range limites
1130 '
1140 IF X2 <= 0 THEN XA = 270 : GOTO 1170
1150 IF X1 => 0 THEN XA = 10 : GOTO 1170
1160 XA = 10 - SX * X1
1170 LOCATE 1,1 : PRINT USING "##.##(V)"; Y2 ;
1180 LOCATE 19,1 : PRINT USING "##.##(V)"; Y1 ;
1190 LOCATE 20,10 : PRINT USING "##.##(S)"; X1 ;
1200 LOCATE 20,68 : PRINT USING "##.##(SO"; X2 ;
1210 PSET (QX * 0 + OX,QY * 0 + OY)
1220 LINE - (QX * 279 + OX,QY * 0 + OY)
1230 LINE - (QX * 279 + OX,QY * 159 + OY)
1240 LINE - (QX * 0 + OX,QY * 159 + OY)
1250 LINE - (QX * 0 + OX,QY * 0 + OY)
1260 LINE (QX*XA+OX,QY*10+OY)-(QX*XA+OX,QY*150+OY)
1270 LINE (QX*10+OX,QY*YA+OY)-(QX*270+OX,QY*YA+OY)
1280 '
1290 'Mark the x axis ticks
1300 '
1310 K = 0
1320 B = YA - 2
1330 C = YA + 2
1340 K = K + 1
1350 A = DX * K
1360 AA = XA + SX * A
1370 IF AA > 271 GOTO 1400
1380 LINE (QA*AA+OX,QY*B+OT)-(QX*AA+OX,QY*C+OY)
1390 GOTO 1340
1400 K = 0
1410 K = K + 1
1420 A = DX * K
1430 AA = XA -SX * A
1440 IF AA < 9 GOTO 1500
```

```
1450 LINE (QX*AA+OX,QY*B+OY)-(QX*AA+OX,QY*C+OY)
1460 GOTO 1410
1470 '
1480 'Mark the y axis ticks
1490 '
1500 K = 0
1510 A = XA - 2
1520 C = XA + 2
1530 K = K + 1
1540 B = DY * K
1550 BB = YA - SY * B
1560 IF BB < 9 GOTO 1590
1570 LINE (QX*A+OX,QY*BB+OY)-(QX*C+OX,QY*BB+OY)
1580 GOTO 1530
1590 K = 0
1600 K = K + 1
1610 B = DY * K
1620 BB = YA + SY * B
1630 IF BB > 151 GOTO 1660
1640 LINE (QX*A+OX,QY*BB+OY)-(QX*C+OX,QY*BB+OY)
1650 GOTO 1600
1660 X = 0
1670 XTMP = 0
1680 YTMP = 0
1690 FOR J = 0 TO NC - 1
1700 MD% = 13
1710 CALL DAS8 (MD%, IP%, FLAG%)
1720 IF FLAG% <> 0 THEN PRINT "Error in read digital input
     #1 ": STOP
1730 IF IP% <> 1 GOTO 1700
1740 MD% = 11     'Mode 11 to load counter #2
1750 D%(0) = 2    'Operate on counter #2
1760 D%(1) = N1
1770 CALL DAS8 (MD%, D%(0), FLAG%)
1780 IF FLAG% <> 0 THEN PRINT "Error in loading counter #2 "
     : STOP
1790 D%(0) = 1    'Operate on counter #1
1800 D%(1) = F1
1810 CALL DAS8 (MD%, D%(0), FLAG%)
1820 IF FLAG% <> 0 THEN PRINT "Error in loading counter #1"
     : STOP
1830 'Lock DAS8 to channel #0 (Experiment input signal)
      using mode 1.
1840 MD% = 1 : LT%(0) = 0 : LT%(1) = 0
1850 CALL DAS8 (MD%, CJ%, FLAG%)
1860 IF FLAG% <> 0 THEN PRINT "Error in setting #0 channel"
     : STOP
1870 MD% = 4 : CJ% = 0
1880 CALL DAS8 (MD%, CJ%, FLAG%)
1890 A(J,0) = CJ% * 5 / 2048
1900 Y = ABS(A(J,0)
1910 X = X + 1 / F
1920 PS = 0
1930 AA = XA + SX * X
```

```
1940 AT = XA + SX * XTMP
1950 BB = YA - SY * Y
1960 BT = YA - SY * YTMP
1970 IF AA < 9 OR BB < 9 THEN PS = 1
1980 IF AA > 271 OR BB > 151 THEN PS = 1
1990 IF PS GOTO 2010
2000 LINE (QX*AT+OX,QY*BT+OY)-(QX*AA+OX,QY*BB+OY) 'LINE PLOT
2010 XTMP = X : YTMP = Y
2020 'Lock DAS8 to channel #1 (Baseline input signal) using
      mode 1
2030 MD% = 1 : LT%(0) = 1 : LT%(1) = 1
2040 CALL DAS8 (MD%, LT%(0),FLAG%)
2050 IF FLAG% <> 0 THEN PRINT "Error in setting #1 channel"
     : STOP
2060 MD% = 4 : D% = 0
2070 CALL DAS8 (MD%,D%,FLAG%)
2080 A(J,1) = D5 * 5 / 2048
2090 NEXT J
2100 LOCATE 10,30 : PRINT "DATA ACQUISITION IS OK":BEEP
2110 LOCATE 24,1 : INPUT "Please enter the air value ";AI
2120 OPEN D$ FOR OUTPUT AS #1
2130 S1 = 0
2140 FOR J = 0 TO NC - 1
2150 IF ABS(A(J,0))>A(J,1) THEN B(J)=ABS(A(J,0)) ELSE B(J)=
     A(J,1)
2160 U(J) = B(J) - A(J,1)
2170 PRINT #1, STR$(A(J,0)),STR$(A(J,1)),STR$(U(J))
2180 S1 = S1 + A(J,1)
2190 NEXT J
2200 CLOSE #1
2210 SUM = 0
2220 FOR J = 0 TO NC - 2
2230 SUM = SUM + 0.5 * (U(J) + U(J+1))
2240 NEXT J
2250 A2(Q%)=(SUM*(1/F)*100)/((AI-S1/NC)*DS)
2260 CLS: LOCATE 2,10 : PRINT "The percentage of air
     entrainment is ";A2(Q%);"%"
2270 '
2280 '-----STEP 4: CHOOSE SUBMUNE ------------------------
2290 LOCATE 4,1 : PRINT "Choose from the following:"
2300 PRINT : PRINT "   <1> - DISPLAY DATA ON SCREEN AND
     RETURN TO MENU "
2310 PRINT : PRINT "   <2> - GENERATE A DATA FILE & PLOT
2320 PRINT : PRINT "   <3> - DO TEST ONCE AGAIN "
2330 PRINT : PRINT "   <4> - CALCULATE ONCE AGAIN "
2340 PRINT : PRINT "   <5> - CONTINUE DOING EXPERIMENT "
2350 PRINT : PRINT "   <6> - FINISH EXPERIMENT "
2360 PRINT : PRINT "   <7> - EXIT TO MAIN MENU "
2370 PRINT : PRINT "ENTER SELECTION NUMBER (1-7):";
2380 A$=INKEY$:IF A$= "" GOTO 2380
2390 PRINT A$
2400 IF VAL(A$) = 1 GOTO 3140
2410 IF VAL(A$) = 2 GOTO 3420
2420 IF VAL(A$) = 3 THEN CLS : Q% = Q% - 1 : GOTO 700
```

```
2430 IF VAL(A$) = 4 GOTO 2110
2440 IF VAL(A$) = 5 THEN CLS : GOTO 700
2450 IF VAL(A$) = 6 THEN GOTO 2480
2460 IF VAL(A$) = 7 THEN ERASE A,B,U,P : GOTO 380
2470 PRINT "[";A4;"] IS NOT A VALID ENTRY.PLEASE RE-ENTER"
     : GOTO 2370
2480 PRINT : PRINT
2490 INPUT "RESULTS FILE NAME : (AUTOMATIC .R EXT)"; R$
2500 R$ = R$ + ".R"
2510 OPEN R$ AS #2 LEN = 25
2520 FIELD #2,25 AS FI$
2530 T$ = "THE RESULTS OF EXPERIMENT"
2540 LSET FI$ = T$
2550 PUT #2,1
2560 D$ = DATE$
2570 LSET FI$ = D$
2580 PUT #2,2
2590 T1$ = "TEST NUMBER ": T2$ = "AIR PERCENT"
2600 LSET FI$ = T1$
2610 PUT #2,3
2620 LSET FI$ = T2$
2630 PUT #2,4
2640 CLS:LOCATE 1,25 : PRINT T$
2650 LOCATE 3,55 : PRINT DATE$
2660 LOCATE 5,20 : PRINT T1$,T2$
2670 K = 0
2680 LSET FI$ = MKI$(Q%)
2690 PUT #2,5
2700 SUM1 = 0
2710 FOR I = 1 TO Q%
2720 SUM1 = SUM1 + A2(I)
2730 LOCATE 7+I,25 : PRINT I,,"  ";USING "###.##";A2(I);
     : PRINT " %"
2740 LSET FI$ = MKS$(A2(I))
2750 PUT #2,5+I
2760 K = K + 1
2770 NEXT I
2780 E$ = "-----------------------------------"
2790 PRINT SPC(18); E$
2800 S1 = SUM1 / Q%
2810 V$ = "THE AVERAGE"
2820 LOCATE 11+K,20 : PRINT V$,"  ";USING "###.##";S1;
     : PRINT " %"
2830 Q = Q%
2840 LSET FI$ = V$
2850 PUT #2, 6 + K
2860 LSET FI$ = MKS$(S1)
2870 PUT #2, 7 + K
2880 CLOSE #2
2890 GOSUB 2910
2900 CLS: GOTO 380
2910 LOCATE 25,1 : PRINT "DO YOU WANT TO PRINT ? PRESS Y OR
     y(ES) OR PRESS N OR n(0)"
2920 LOCATE 23,63 : INPUT "",A$
```

```
2930 IF A$ = "N" OR A$ = "n" GOTO 3130
2940 IF A$ = "Y" OR A$ = "y" GOTO 2960
2950 LOCATE 25,1 : PRINT "RE-ENTER" : GOTO 2910
2960 LOCATE 24,1 : PRINT "PRINTER IS PUT ON ? IF YES,PRESS P
     OR p KEY TO PRINT"
2970 LOCATE 23,63 : INPUT "",A$
2980 IF A$ = "P" OR A$ = "p" GOTO 3000
2990 LOCATE 25,1 : PRINT "RE-ENTER" : GOTO 2960
3000 LPRINT SPC(20);T$
3010 LPRINT : LPRINT
3020 LPRINT SPC(50);D$
3030 LPRINT : LPRINT : LPRINT
3040 LPRINT SPC(10);T1$,,T2$
3050 LPRINT : LPRINT
3060 FOR I = 1 TO Q
3070 LPRINT SPC(13);I,,USING "##.##";A2(I);:LPRINT " %"
3080 LPRINT : LPRINT
3090 NEXT I
3100 LPRINT E$
3110 LPRINT : LPRINT
3120 LPRINT SPC(10);V$,,USING"##.##";S1;: LPRINT " %"
3130 RETURN
3140 '-----DISPLAY DATA ON SCREEN AND RETURN TO MENU -------
3150 CLS
3160 LOCATE 25,1 : PRINT"PRESS ANY KEY TO STOP/START DISPLAY
     ,<ESC> KEY TO RETURN TO DATA STORAGE MENU ";:LOCATE 1,1
3170 IF SIGN = 3 GOTO 3250
3180 FOR I = 1 TO NC - 1
3190 PRINT A(I,0),A(I,1),U(I)
3200 A$ = INKEY$
3210 IF A$ = CHR$(27) THEN I = NC + 3
3220 IF A$ <> "" GOTO 3370
3230 NEXT I
3240 GOTO 3320
3250 PRINT "AIR VALUE = ";AI,"CONCRETE VALUE = ";S1/NC
3260 FOR I = 0 TO NC/4-1
3270 PRINT"U(";I*4;")=";U(I*4);"U(";I*4+1;")=";U(I*4+1);
     "U(";I*4+2;")=";U(I*4+2);"U(";I*4+3;")=";U(I*4+3)
3280 A$ = INKEY$
3290 IF A$ = CHR$(27) THEN I = NC + 3
3300 IF A$ <> "" GOTO 3370
3310 NEXT I
3320 IF I = NC + 3 THEN CLS : GOTO 3350
3330 LOCATE 25,1 : PRINT " PRESS ANY KEY TO RETURN TO DATA
     STORAGE MENU";
3340 IF INKEY$ = "" GOTO 3340
3350 IF SIGN = 3 THEN CLS : GOTO 7100
3360 CLS : GOTO 2280
3370 FOR K = 1 TO 50 : NEXT K   'DELAY
3380 IF INKEY$ = "" GOTO 3380
3390 IF SIGN = 3 GOTO 3310 ELSE GOTO 3230
3400 IF SIGN = 3 THEN CLS : GOTO 7100
3410 CLS : GOTO 2280
3420 '----- Generate DAS8 graphics package data file & plot
```

```
     -----
3430 CLS
3440 LOCATE 1,1 : INPUT "DATA FILE NAME [DRIVER]:NAME.EXT";
     FILX$
3450 OPEN FILX$ AS #1 LEN = 30
3460 FIELD #1, 15 AS X$, 15 AS Y$
3470 'Enter number of test and air value
3480 LSET X$ = MKI$(Q%) : LSET Y$ = MKS$(AI)
3490 PUT #1,1
3500 'Enter air persent and integral sum
3510 LSET X$ = MKS$(A2(Q%)) : LSET Y$ = MKS$(SUM)
3520 PUT #1,2
3530 'Enter sampling freq. and duration
3540 LSET X$ = MKS$(F) : LSET Y$ = MKS$(DS)
3550 PUT #1,3
3560 'Enter number of data points and base line sum
3570 LSET X$ = MKS$(NC) : LSET Y$ = MKS$(S1)
3580 PUT #1,4
3590 'Enter data in remaining records
3600 FOR I = 5 TO NC + 4
3610 LSET X$ = MKS$((I-5) / F) : LSET Y$ = MKS$(U(I-5))
3620 PUT #1,I
3630 NEXT I
3640 CLOSE #1
3650 GOTO 3670
3660 LOCATE 23,1 : INPUT "PLEASE ENTER THE DATA FILE NAME :"
     ;FILX$
3670 'Generate RLINPLT.LNK plotting file for LINPLT.BAS to
      use
3680 OPEN "RLINPLT.LNK" AS #1 LEN = 30
3690 FIELD #1,30 AS RLNK$
3700 LSET RLNK$ = MKI$(1)    'One data file
3710 PUT #1,1
3720 CLS
3730 Y$ = "REFLECTED LIGHT INTENSITY(VOLT)"
3740 IF LEN(Y$) > 30 THEN Y$ = LEFT$(Y$,30)
3750 LSET RLNK$ = Y$         'Y axis label
3760 X$ = "SAMPLING DURATION (SEC)"
3770 IF LEN(X$) > 30 THEN X$ = LEFT$(X$,30)
3780 PUT #1,2
3790 LSET RLNK$ = X$         'X axis label
3800 PUT #1,3
3810 LSET RLNK$ = FILX$      'Data file name
3820 PUT #1,4
3830 CLOSE #1
3840 '
3850 '   This ection is where the starting graph coordinates
     and screen scale
3860 'ratio is set.OX,OY = starting point of graph.QX,QY =
      screen scale x:y.
3870 '
3880 KEY OFF : SCREEN 0,0,0
3890 OX = 70 : OY = 0 : QX = 1.8 : QY = 0.9000001
3900 CLS:LOCATE 12,30 : COLOR 31,0,0 : PRINT "COMPUTING
```

```
     GRAPHICS FILES";
3910 '
3920 '
3930 '  This is the main link to the plotting program.The
3940 'file RLINPLT.LNK has the file name and disk of the
      file(s) to be plotted.
3950 '
3960 '
3970 OPEN "RLINPLT.LNK" AS #1 LEN = 30
3980 FIELD #1,30 AS RLNK$
3990 GET #1,1
4000 NF = CVI(RLNK$)
4010 GET #1,2
4020 YLB$ = RLNK$
4030 GET #1,3
4040 XLB$ = RLNK$
4050 IP = 1
4060 FOR I = 1 TO NF
4070 GET #1,I+3
4080 F$(IP) = RLNK$
4090 IP = IP + 1
4100 NEXT I
4110 CLOSE
4120 '
4130 '
4140 '  This loop scans the data files and sets the X,Y
      Max/Min value so
4150 'all the data files are on the same scale.
4160 '
4170 '
4180 X1 = 0 : XTMP = 0
4190 X2 = 0 : YTMP = 0
4200 Y1 = 0
4210 Y2 = 0
4220 FOR I = 1 TO NF
4230 IC = 0
4240 OPEN F$(I) AS #1 LEN = 30
4250 FIELD #1, 15 AS XVAL$, 15 AS YVAL$
4260 GET #1,4
4270 NI = CVS(XVAL$)
4280 FOR J = 1 TO NI
4290 IC = IC + 1
4300 GET #1, IC + 4
4310 A = CVS(XVAL$)
4320 B = CVS(YVAL$)
4330 IF A < X1 THEN X1 = A
4340 IF A > X2 THEN X2 = A
4350 IF B < Y1 THEN Y1 = B
4360 IF B > Y2 THEN Y2 = B
4370 NEXT J
4380 CLOSE #1
4390 NEXT I
4400 '
4410 '    This section establishes the actual scale factors
```

```
4420 'used to plot the data as the files are read. These
4430 'values are also the same values saved on the disk when
      a save command is executed.
4440 '
4450 CLS : COLOR 7,0,0 : SCREEN 2
4460 DX = ABS(CINT((X2 - X1) / 25 + 0.5))
4470 DY = ABS(CINT(((Y2 - Y1) / 25) + 0.5))
4480 SX = 260 / (X2 - X1)
4490 SY = 140 / (Y2 - Y1)
4500 '
4510 '   This section locates the Axis x,y and plots them
4520 'out with the small tick marks identifing the scale.
      The axis labels are also printed.
4530 '
4540 '
4550 IF Y2 <= 0 THEN YA = 10 : GOTO 4590
4560 IF Y1 => 0 THEN YA = 150 : GOTO 4590
4570 YA = 10 + SY * Y2
4580 '
4590 'Set the range limits
4600 '
4610 IF X2 <= 0 THEN XA = 270 : GOTO 4640
4620 IF X1 => 0 THEN XA = 10 : GOTO 4640
4630 XA = 10 - SX * X1
4640 FOR YLBL = 1 TO 16
4650 LOCATE 2 + YLBL,6 : PRINT MID$(YLB$,YLBL,1);
4660 NEXT YLBL
4670 LOCATE 1,1 : PRINT USING "#.##(V)";Y2;
4680 LOCATE 19,1 : PRINT USING "#.##(V)";Y1;
4690 LOCATE 20,10 : PRINT USING "#.##(S)";X1;
4700 LOCATE 20,68 : PRINT USING "#.##(S)";X2;
4710 PSET (QX*0+OX,QY*0+OY)
4720 LINE - (QX*279+OX,QY*0+OY)
4730 LINE - (QX*279+OX,QY*159+OY)
4740 LINE - (QX*0+OX,QY*159+OY)
4750 LINE - (QX*0+OX,QY*0+OY)
4760 LINE (QX*XA+OX,QY*10+OY)-(QX*XA+OX,QY*150+OY)
4770 LINE (QX*10+OX,QY*YA+OY)-(QX*270+OX,QY*YA+OY)
4780 '
4790 ' Mark the x axis ticks
4800 '
4810 K = 0
4820 B = YA - 2
4830 C = YA + 2
4840 K = K + 1
4850 A = DX * K
4860 AA = XA + SX * A
4870 IF AA > 271 GOTO 4900
4880 LINE (QX*AA+OX,QY*B+OY)-(QX*AA+OX,QY*C+OY)
4890 GOTO 4840
4900 K = 0
4910 K = K + 1
4920 A = DX * K
4930 AA = XA - SX * A
```

```
4940 IF AA < 9 GOTO 5000
4950 LINE (QX*AA+OX,QY*B+OY)-(QX*AA+OX,QY*C+OY)
4960 GOTO 4910
4970 '
4980 'Mark the y axis ticks
4990 '
5000 K = 0
5010 A = XA - 2
5020 C = XA + 2
5030 K = K + 1
5040 B = DY * K
5050 BB = YA - SY * B
5060 IF BB < 9 GOTO 5090
5070 LINE (QX*A+OX,QY*BB+OY)-(QX*C+OX,QY*BB+OY)
5080 GOTO 5030
5090 K = 0
5000 K = K + 1
5110 B = DY * K
5120 BB = YA + SY * B
5130 IF BB > 151 GOTO 5160
5140 LINE (QX*A+OX,QY*BB+OY)-(QX*C+OX,QY*BB+OY)
5150 GOTO 5100
5160 LOCATE 22,1
5170 PRINT "       X SPACING= ";INT(DX);"      Y SPACING=";
     INT(DY)
5180 LOCATE 20,22 : PRINT XLB$; : LOCATE 22,1
5190 '
5200 '  This section recalls all the files and plots them
      one by one.
5210 '
5220 FOR I = 1 TO NF
5230 OPEN F$(I) AS #1 LEN = 30
5240 FIELD #1,15 AS XVAL$, 15 AS YVAL$
5250 GET #1,4
5260 NI = CVS(XVAL$) : DOT = 1
5270 FOR J = 5 TO NI + 4
5280 GET #1,J
5290 X = CVS(XVAL$)
5300 Y = CVS(YVAL$)
5310 IF J = 4 THEN XTMP = X : YTMP = Y
5320 PS = 0
5330 AA = XA + SX * X
5340 AT = XA + SX * XTMP
5350 BB = YA -SY * Y
5360 BT = YA - SY * YTMP
5370 IF AA < 9 OR BB < 9 THEN PS = 1
5380 IF AA . 271 OR BB > 151 THEN PS = 1
5390 IF PS GOTO 5450
5400 IF INT(DOT + 0.5) = 2 GOTO 5470  'No plot
5410 IF INT(DOT + 0.5) = 0 GOTO 5460  ,Dot plot
5420  LINE (QX*AT+OX,QY*BT+OY) - (QX*AA+OX,QY*BB+OY) 'Line
      plot.
5430 GOTO 5450
5440 PSET (QX*AA+OX,QY*BB+OY)
```

```
5450 XTMP = X : YTMP = Y
5460 NEXT J
5470 GET #1,2
5480 Y1 = CVS(XVAL$)
5490 LOCATE 3,10 : PRINT "AIR PERCENT =";Y1;" %"
5500 CLOSE #1
5510 NEXT I
5520 '
5530 '----- PROGRAM PLOT COMPLETED -----------------------
5540 '
5550 '  This is the conversational part of the program where
      the user may save the plot/graph or add more data or
      print plot/graph.
5570 '
5580 '
5590 LOCATE 24,1 : INPUT;"ENTER OPTION {(S)AVE, (C)HAIR,
     (E)XIT} ";OPTX$
5600 IF OPTX$ = "S" OR OPTX$ = "s" THEN 5660
5610 IF OPTX$ = "C" OR OPTX$ = "c" THEN 5800
5620 IF OPTX$ = "E" OR OPTX$ = "e" THEN CLS:ON SIGN GOTO
     380,2280,7100
5630 LOCATE 24,1 : PRINT SPC(79) : LOCATE 24,1 : GOTO 5590
5640 '
5650 '
5660 'SAVE SCREEN ROUTINE
5670 '
5680 LOCATE 25,1 : PRINT SPACE$(78);:LOCATE 25,1
5690 INPUT ;"ENTER [DSK:] FILENAME(NO EXTENSION)";FIL$
5700 OPEN FIL$ + ".PAR" AS #1 LEN = 30
5710 GOSUB 5860
5720 LOCATE 24,1 : PRINT SPACE$(78);:LOCATE 25,1 : PRINT
     SPACE$(78);:LOCATE 23,1
5730 DEF SEG
5760 LOCATE 25,1 : PRINT "FILE ";CHR$(340;FIL$ ;CHR$(34);
     " SAVED ";
5770 LOCATE 23,1 : CLS : GOTO 2280
5780 LOCATE 24,1 : PRINT SPACE$(78);:LOCATE 23,1
5790 CLS:ON SIGN GOTO 2280,380
5800 LOCATE 24,1 : PRINT SPACE$(78);:LOCATE 24,1
5810 INPUT ;"Enter DSK:filename.ext -- ";FLX$
5820 LOCATE 24,1 : PRINT SPACE$(78);:LOCATE 24,1
5830 GOSUB 5850
5840 GOTO 6130
5850 OPEN "CHAINDAT.PAR" AS #1 LEN = 30
5860 FIELD #1, 15 AS XPAR$, 15 AS YPAR$
5870 LSET XPAR$ = MKS$(SX)
5880 LSET YPAR$ = MKS$(SY)
5890 PUT #1,1
5900 LSET XPAR$ = MKS$(DX)
5910 LSET YPAR$ = MKS$(DY)
5920 PUT #1,2
5930 LSET XPAR$ = MKS$(X2)
5940 LSET YPAR$ = MKS$(Y2)
5950 PUT #1,3
```

```
5960 LSET XPAR$ = MKS$(X1)
5970 LSET YPAR$ = MKS$(Y1)
5980 PUT #1,4
5990 LSET XPAR$ = MKS$(QX)
6000 LSET YPAR$ = MKS$(QY)
6010 PUT #1,5
6020 LSET XPAR$ = MKS$(OX)
6030 LSET YPAR$ = MKS$(OY)
6040 PUT #1,6
6050 LSET XPAR$ = MKS$(XA)
6060 LSET YPAR$ = MKS$(YA)
6070 PUT #1,7
6080 XE = 1 / ABS(X2 - X1) : LSET XPAR$ = MKS$(XE)
6090 YE = 1 / ABS(Y2 - Y1) : LSET YPAR$ = MKS$(YE)
6100 PUT #1,8
6110 CLOSE #1
6120 RETURN
6130 CHAIN FLX$
6140 CLS : ON SIGN GOTO 2280,380
6150 LOCATE 23,1 : INPUT "PRINT FILE NAME (*.R):";R1$
6160 OPEN R1$ AS #2 LEN = 25
6170 FIELD #2,25 AS FI$
6180 GET #2,1
6190 T$ = FI$
6200 CLS : LOCATE 1,25 : PRINT T$
6210 GET #2,2
6220 D$ = FI$
6230 LOCATE 3,55 : PRINT D$
6240 GET #2,3
6250 T1$ = LEFT$(FI$,11)
6260 GET #2,4
6270 T2$ = FI$
6280 LOCATE 5,20 : PRINT T1$,T2$
6290 GET #2,5
6300 Q = CVI(FI$)
6310 K = 0
6320 FOR I = 1 TO Q
6330 GET #2,I + 5
6340 A2(I) = CVS(FI$)
6350 LOCATE 7 + I,25 : PRINT I,,"  ";USING "###.##";A2(I)
     ; : PRINT " %"
6360 K = K + 1
6370 NEXT I
6380 E$ = "-------------------------------------"
6390 PRINT SPC(15);E$
6400 GET #2,K+6:V$ = LEFT$(FI$,11):GET #2,K+7:S1=CVS(FI$)
6410 LOCATE 11+K,20:PRINT V$,"  ";USING "###.##";S1;:PRINT
     " %"
6420 CLOSE #2
6430 GOSUB 2910
6440 CLS : GOTO 380
6450 CLS;INPUT"PLEASE ENTER RESULTS FILE NAME:(*.R)";R1$
6460 PRINT : PRINT
6470 INPUT "PLEASE ENTER THE DATA FILE NAME:";FILX$
```

```
6480 OPEN FILX$ AS #1 LEN = 30
6490 FIELD #1,15 AS X$,15 AS Y$
6500 GET #1,1
6510 Q% = CVI(X$) : AI = CVS(Y$)
6520 LOCATE 7,1 : PRINT "NOW THE AIR VALUE IS :";AI
6530 GET #1,2
6540 A2(Q%) = CVS(X$) : SUM = CVS(Y$)
6550 LOCATE 9,1 : PRINT "NOW THE PERCENTAGE OF AIR
     ENTRAINMENT IS:";A2(Q%);" %"
6560 GET #1,3
6570 F = CVS(X$) : DS = CVS(Y$)
6580 GET #1,4
6590 NC = CVS(X$) : S1 = CVS(Y$)
6600 LOCATE 11,1 : INPUT "PLEASE ENTER THE AIR VALUE:";AI
6610 A2(Q%) = (SUM*(1/F)*100)/((AI-S1/NC)*DS)
6620 CLS:LOCATE 10,10:PRINT "THE PERCENTAGE OF AIR
     ENTRAINMENT IS:";A2(Q%);" %"
6630 LSET X$ = MKI$(Q%) : LSET Y$ =MKS$(AI)
6640 PUT #1,1
6650 LSET X4 = MKS$(A2(Q%)) : LSET Y$ = MKS$(SUM)
6660 PUT #1,2
6670 CLOSE #1
6680 OPEN R1$ AS #2 LEN = 25
6690 FIELD #2,25 AS FI$
6700 LSET FI$ = MKS$(A2(Q%))
6710 PUT #2,5 + Q%
6720 CLOSE #2
6730 LOCATE 24,1 : PRINT "PLEASE PRESS ANY KEY TO RETURN
     TO MENU:"
6740 A$ = INKEY$ : IF A$ = "" GOTO 6740
6750 GOTO 380
6760 '---- COMMUNICATION WITH SLAVE MICROCONTROLLER ----
6770 KEY OFF : SCREEN 0,0,0
6780 CLS:LOCATE 12,30:COLOR 31,0,0:PRINT "COMMUNICATION IS
     GOING ";
6790 F = 100
6800 DS = 5
6810 NC = F * NC
6820 DIM U(NC)
6830 Q% = 0
6840 OPEN "COM1:1200,N,8,1,CS,DS,CD" AS #1
6850 A$ = INPUT$(1,#1)
6860 B$ = "00"
6870 C$ = A$ + B$
6880 AI = (CVI(C$)-12288)/51
6890 FOR I = 0 TO NC - 1
6900 A$ = INPUT$(1,#1)
6910 B$ = "00"
6920 C$ = A$ + B$
6930 U(I) = CVI(C$)
6940 U(I) = (U(I) - 12288) / 51
6950 NEXT I
6960 A$ = INPUT$(1,#1)
6970 B$ = "00"
```

```
6980 C$ = A$ + B$
6990 S1 = ((CVI(C$) - 12288) / 51) * NC
7000 CLOSE #1
7010 Q% = Q% + 1
7020 SUM = 0
7030 FOR J = 0 TO NC - 2
7040 SUM = SUM + 0.5 * (U(J) + U(J+1))
7050 NEXT J
7060 CLS:COLOR 7,0,0 : SCREEN 2
7070 A2(Q%) = SUM * (1 / F) * 100 / ((AI - S1 / NC) * DS)
7080 CLS:LOCATE 2,10:PRINT :"The percentage of air
     entrainment is";A2(Q%);" %"
7090 IF Q% > 3 THEN LOCATE 4,1:PRINT "COMMUNICATION IS
     FINISHED,PLEASE PRESS 5 "
7100 '
7110 '---- DISPLAY COMMUNICATION MENU ---------------------
7120 '
7130 LOCATE 6,1:PRINT "CHOOSE FROM THE FOLLOWING:"
7140 PRINT:PRINT"   <1> - DISPLAY DATA ON SCREEN AND RETURN
                         TO MENU "
7150 PRINT:PRINT"   <2> - GENERATE A DATA FILE & PLOT "
7160 PRINT:PRINT"   <3> - CALCULATE ONCE AGAIN "
7170 PRINT:PRINT"   <4> - CONTINUE COMMUNICATION "
7180 PRINT:PRINT"   <5> - FINISH COMMUNICATION "
7190 PRINT:PRINT"   <6> - EXIT TO MAIN MENU "
7200 PRINT:PRINT"ENTER SELECTION NUMBERS (1-6) :";
7210 A$ = INKEY$ : IF A$ = "" GOTO 7210
7220 PRINT A$
7230 IF VAL(A$) = 1 GOTO 3140
7240 IF VAL(A$) = 2 GOTO 3420
7250 IF VAL(A$) = 3 GOTO 7290
7260 IF VAL(A$) = 4 GOTO 7320
7270 IF VAL(A$) = 5 GOTO 2480
7280 IF VAL(A$) = 6 THEN ERASE U : GOTO 380
7290 '---- Calculate once again --------------------------
7300 CLS : LOCATE 4,1 : INPUT "Please enter the air value:"
     ;AI
7310 GOTO 7070
7320 KEY OFF : SCREEN 0,0,0
7330 CLS:LOCATE 12,30:COLOR 31,0,0:PRINT "COMMUNICATION IS
     GOING ";
7340 GOTO 6840
7350 CLS:COLOR 7,0,0 : SCREEN 2 : GOTO 7100
7360 END
```

BIBLIOGRAPHY

1. Ramachandran, V. S. " Concrete Admixtures Handbook - Properties, Science, and Technology." (1987): 269-283.

2. Born, Max. " Principles of Optics." (Fourth Edition), Pergamon Press, (1970): 38.

3. Jenkins, Francis A. " Fundamental of Optics." (Fourth Edition), McGRAW Hill (1976): 523-526.

4. Nakayama, Takashi. " Optical Sensing Technologies by Multimode Fibers ." Proceedings of 1984 SPIE The International Society for Optical Engineering,(1984): 19-26.

5. Giallorenzi, Etal T. G. " Optical Fiber Sensor Technology."IEEE Journal on Quantum Electronics, QE18, (1982): 626.

6. Chen, Qiyu. "A Multimode Index Detecting Sensor for Use in Plastic Concrete." Thesis. E.E. Dept. NJIT, Directed by Professor Farhad Ansari.(1991): 15-16.

7. Newport Corporation. " Projects in Fiber Optics Applications Handbook." (1988).

8. Junya, Motodate. "Short Wavelength Visible Laser Diode TOLD-9200 Developped by Toshiba has lasing Wavelength of 670 nm." JEE, August (1988): 34.

9. OMEGA ENGINEERING, INC."DAS-8PGA A/D INTERFACE BOARD." Operator's Manual.

10. China Shanghai FuDan Univ. " SICE-III Single Chip Microcomputer In circuit Emulater." Operator's Manual. Juny, (1989).

11. OPTREX Co. "Dot Matrix LCD Module." Manual.

12. Hughes, Fredrick W. "OP-AMP Handbook."(Second Edition), Prentice Hall, Englewood Cliffs, NJ (1986): 71.

13. Intel "8-Bit Embeded Controllers." (1991).

14. Intel "Embedded Applications." (1991).

15. Intel "16-Bit Embedded Controllers." (1991).

16. Sarup, Anita. " Image Analysis of Air Void Characteristics in Hardened Concrete." Thesis, Civil and Environment Engineering, NJIT, Directed by Professor Faehad Ansari.

17. Recommended Practice for the Microscopical Determination of Air Void Content and Parameters of the Air Void System in Hardened Concrete, ASTM C57-82a, Annual Book of ASTM standard, Vol 4.02, ASTM, Philadelphia, (1989).

18. Roden, Martin S. " Analog and Digital Communication Systems." (1979): 43-44.