

BSPE 00217-364-7

遠隔 水質모니터링 시스템 開發研究

A Study on the Development of the Remote
Water Quality Monitoring System

1991. 4.

韓國海洋研究所

提 出 文

韓國海洋研究所長 貴下

本 研究書を “遠隔 水質모니터링 시스템 開發 研究”에 관한 研究 報告書로 提出
합니다.

1991年 4月 日

韓國海洋研究所

研究 責任 者 : 全 鎬 景 (韓國海洋研究所 海洋機器室)
技術諮問委員 : 李 種 和 (順天鄉大學校 生物學科 教授)
研 究 員 : 朴 建 泰 (韓國海洋研究所 海洋機器室)
咸 錫 賢 (韓國海洋研究所 海洋機器室)
朴 東 元 (韓國海洋研究所 海洋機器室)
黃 槿 春 (韓國海洋研究所 海洋機器室)
研 究 助 員 : 李 忠 雨 (韓國海洋研究所 海洋機器室)
韓 東 熙 (韓國海洋研究所 海洋機器室)
金 美 垆 (韓國海洋研究所 海洋機器室)

요약문

I. 제 목

원격 자동 수질모니터링 시스템 개발연구

II. 연구개발의 목적 및 중요성

- 원격 자동 수질 검출장치 및 자료의 중앙집중 시스템 개발
- 양식장, 증식장 및 어장의 지속적인 수질 관리의 과학화로 생산성 향상 도
모
- 강, 하천, 바다 등의 환경보전 경보 시스템으로 발전

III. 연구개발의 내용 및 범위

1. 수질 검출

- 기존 센서를 이용한 고정 또는 이동식 검출기 제작
수온, 용존산소량, pH, 탁도, 염분도

2. 자료전송 및 중앙집중

- 전기 통신 기술을 이용한 자료 전송
- 수신부와 PC를 이용한 자료의 연속 기록, 분석
- 각 현장 또는 지역의 관측결과 및 검출 자료의 중앙 집중 체제 구축

IV. 연구개발의 활용에 대한 건의

1. 환경보전, 오염예방 관측망으로 활용가능
2. 무선통신과 컴퓨터를 이용한 자료의 원거리 전송방식 기술 응용

SUMMARY

I. Title

A study on the development of the remote water quality monitoring system.

II. Objective and significance

- Development of the remote water quality monitoring and data telemetering system.
- Increase the productivity of cultivating industry through the continuous management of water quality.
- The protection of environment of river and ocean.
- Development of one of warning system to the protection of environment of river and ocean.

III. Contents

1. To measure water quality

- A production of the equipment of a water quality monitor which is a fixed or portable type. (The sensors are temperature, dissolved oxygen, conductivity, turbidity and pH etc.)

2. Data telemetering

- Development of the data telemetering technique by the radio communication

- Establishment a system of the central data control station from the each local stations

IV. Applications

1. Could be used a equipment of the protection of environment and water pollution.
2. Practical use, the technicque of data telemetering method, the combined of radio communication and computer

目 次

要約文

영문요약문	5
List of figures	9
List of tables	13
List of photos	15
Abstract	17
제 1 장 序 論	19
제 2 장 材 料 및 方 法	20
제 1 절 材 料	20
1. System의 構成 및 特性	20
2. 센서 회로의 構成	22
제 2 절 方 法	34
1. Data Acquisition Board	34
2. 送信部	48
3. 受信部	58
4. Data Acquisition Program	67
제 3 장 結 論 및 討 議	109
1. 結 論 및 資 料 分 析	109

參考文獻	113
Appendix I(Photography)	115
Appendix II(現場 實測 資料)	119

그림 목 차

그림 1-1	Data 전송 시스템 구성	22
그림 1-2	온도와 Thermistor 저항값 및 A/D 변화 값과의 관계	24
그림 1-3	온도의 측정 오차	26
그림 1-4	전기전도도 센서의 구조	27
그림 1-5	전기전도도와 A/D 변환값 및 오차관계	28
그림 1-6	포화 용존산소량의 온도의존성 및 염소 이온량에 따른 변화	31
그림 1-7	온도 센서와 전체 회로	32
그림 1-8	전기전도도 회로	33
그림 2-1	ADC 1210의 시간 계통도	37
그림 2-2	A/D 변환기의 입출력 특성	39
그림 2-3	직렬 전송자료의 파형	40
그림 2-4	수질 모니터링 시스템의 계통도	41
그림 2-5	클럭회로	42
그림 2-6	고정자료 출력회로(A)	43
그림 2-7	고정자료 출력회로(B)	44
그림 2-8	8 Channel 출력회로	45
그림 2-9	카운타 회로	46
그림 2-10	병렬-직렬 Converter	47
그림 2-11	FM파 발생 계통도	51
그림 2-12	송신 국부 발진회로	52
그림 2-13	벡터 합성 계통도	53

그림 2-14 송신 회로	55
그림 2-15 수신 국부 발진회로	56
그림 2-16 변별기 회로	60
그림 2-17 변별기 및 가청회로	64
그림 2-18 수신기 회로	65
그림 2-19 파형 보정회로 및 RS-232C 입력 파형 모양	66

List of figures

Fig. 1-1	Block diagram of the data transmitting system	22
Fig. 1-2	Characteristic curve for the temperature sensor	24
Fig. 1-3	Tolerance values as a function of temperature for the temperature sensor	26
Fig. 1-4	Four-electrode conductivity sensor	27
Fig. 1-5	Characteristic curve and tolerance values for the conductivity sensor	28
Fig. 1-6	D.O. calibration curve for the temperature & Cl	31
Fig. 1-7	Temperature sensor circuit	32
Fig. 1-8	Conductivity sensor circuit	33
Fig. 2-1	Timing diagram for ADC 1210 A/D converter	37
Fig. 2-2	A/D transfer characteristic curve	39
Fig. 2-3	Serial data waveform	40
Fig. 2-4	Block diagram for the water quality monitoring system	41
Fig. 2-5	Clock circuit	42
Fig. 2-6	Fixed data circuit (A)	43
Fig. 2-7	Fixed data circuit (B)	44
Fig. 2-8	8 channel A/D converter	45
Fig. 2-9	Counter circuit	46
Fig. 2-10	Parallel to serial converter	47
Fig. 2-11	Block diagram of the frequency modulation	51
Fig. 2-12	Local oscillator of the summing of vector	52

Fig. 2-13	Block diagram of the summing of vector	53
Fig. 2-14	Transmitter circuit	55
Fig. 2-15	Local oscillator of the receiver	56
Fig. 2-16	A circuits of the discriminator	60
Fig. 2-17	Discriminator and audio circuit	64
Fig. 2-18	Receiver circuit	65
Fig. 2-19	Compensation circuit for the waveforms	66

표 목 차

표 1-1	출력 신호 형식	21
표 1-2	온도와 Thermistor 저항 및 A/D 변화 값과의 관계	24
표 1-3	온도의 실측치와 교정치와의 오차	25
표 1-4	전기전도도와 A/D 변환 값과의 관계	28
표 1-5	포화 용존 산소량의 온도의존성 및 염분 이온량에 따른 변화	30
표 2-1	A/D 변환기의 입출력 특성 및 오차	38

List of tables

Table 1-1	Formating of output signal	21
Table 1-2	The relationship of temperature/resistance/AD converter	24
Table 1-3	Calibration data for the temperature sensor	25
Table 1-4	Calibration data for the conductivity sensor	28
Table 1-5	D.O. calibration values for various temperature & Cl	30
Table 2-1	A/D transfer characteristic and tolerance values	38

사 진 목 차

사진 1-1	센서의 내부 모양	115
사진 1-2	센서의 외부 모양	115
사진 1-3	A/D converter의 모양	116
사진 1-4	송신기 및 센서	116
사진 1-5	수신기	117
사진 1-6	수신기 및 PC 자료처리	117
사진 1-7	송신 안테나(146 MHz대)	118
사진 1-8	수신 안테나(146 MHz대)	118

List of photos

Photo 1-1	Inside of sensors	115
Photo 1-2	Outside of sensors	115
Photo 1-3	A/D converter	116
Photo 1-4	Transmitter and sensor	116
Photo 1-5	Receiver	117
Photo 1-6	Receiver and data terminal(PC)	117
Photo 1-7	Transmitter antenna	118
Photo 1-8	Receiver antenna	118

Abstract

This paper describes a simple system used to measure water quality and to monitor the open fishfarm in a river or ocean.

It consists of a small sensor package and water-tight case containing a radio transmitter and power source. This system took advantage of the fact that the temperature, conductivity, D.O., pH and turbidity sensors were attached to the sensor module.

Consequently, these sensors received the signals which are directly related to water quality and characteristics. The data acquisition board has 8 single-ended analog input and 2 digital outputs. One is CMOS voltage level and the other is EIA voltage level for telecommunication with a computer. These two outputs are used to retrieve data from the data acquisition board and to send the information to a computer. The communication with the data acquisition board is through radio telecommunications. The receive station also consists of a radio receiver, a frequency discriminator which has a $\pm 10[V]$ pulse code output, and suitable recording system for the PC.

These device are tested experimental prototypes, so it may be expected that subsequent improvements will be made, especially to facilitate the handling of the device in the field.

In this study, the settings of the transmitter power, power source and the frequency band used were 0.8[W], +12[V] DC and 146.2 [MHz] respectably.

Ideally the design of these equipments which are used in the field, should be extremely small for easier handling in the installing space.

In the experiment described above, the data were recorded on a printer, as well as, digitized and recorded on the computer.

제 1 장 서 론

이 보고서는 養殖場 또는 增殖場이나 培養場등에서 科學的으로 수질을 관리하여 줌으로서 體系的이고 持續的인 生産性 向上과 더 나아가서는 강이나 河川등의 환경보전 및 오염예방을 目的으로 始作하였다. 通信技術과 컴퓨터 기술이 結合됨에 따라 데이터 通信시스템은 여러지역에서 發生된 必要狀況을 電氣 通信 技術을 利用하여 傳送하고 컴퓨터로 處理하는 것이다.

본 研究는 이러한 시스템을 利用한 現場에서 獲得한 資料를 실시간으로 必要한 사람이 利用할 수 있도록 設計되었다. 水溫, 鹽分, 溶存산소량, pH 및 탁도의 센서 Module과 送信機 및 受信機, 資料를 分析할 수 있는 PC(Personal Computer)로 構成되어 있다. 즉 이러한 센서들은 水質 또는 물의 特性을 繼續的으로 感知한다. 8채널로 構成된 아날로그 回路의 人力된 資料들은 2개의 디지털 出力으로 되어 컴퓨터의 資料로서 利用되는 것이다.

實驗에 의하면 送信 출력은 +12[V] DC에서 0.8[W]이며 146.200[MHz] 대이다. 수신 설비 역시 受信機와 電原으로 構成되었으며 受信 出力단에서 RS-232C의 適合한 크기의 枱형이 檢出된다.

可能한한 現場에서 좁은 空間에서 取扱이 容易하도록 모든 機器를 最少化하였다. 센서로 부터 檢出된 資料는 送信機, 受信機를 통하여 실시간으로 컴퓨터에서 記錄 分析하여 處理된다.

제 2 장 材料 및 方法

제 1 절 材料

1. System의 構成 및 特性

計測制御 시스템의 基本 構造는 計測制御 대상으로 부터 얼마간의 물리량이 發生하고 그것이 센서의 感應으로 인한 入力 信號로 된다. 이 물리량은 溫度, 壓力, 流速, 유향, 電氣傳導度, DO, 탁도 등 다양한 形態의 要素라고 할 수 있다.

이러한 각종 센서를 그 出力의 信號 形式에 따라 分類하면 Table 1-1과 같이 나누어 볼 수 있다. 그리고 그 出力 信號의 形式은 다시 비전기량 형식과 전기량 형식으로 分類된다. 비전기량이라 함은 出力 信號가 어떠한 Image 또는 색깔 등의 形態로 나타남을 말하는 것으로 금번 研究 課程에서는 제외시켰다. 즉 電氣量 形式의 것은 그 結果를 利用한 物理的 變化를 쉽게 끄집어 내어 利用할 수 있다. 다시말하면 Analogue 形式, Digital 形式 등은 電氣量 形式으로 分類하며 資料의 傳送, 記錄의 보통 方式인 것이다.

一般的으로 信號波를 原型 그대로 通信路로 通過시키려면 여러가지 어려운 점들이 따르게 된다. 따라서 원래의 信號를 주어진 通信路로 보내기에 알맞게 變形하는 것을 아날로그 通信에서는 變調, 디지털 通信에서는 等號化라고 한다. 따라서 受信쪽에서는 다시 信號를 復舊시킬 必要가 있으며 이것을 아날로그 通信에서는 복조, 디지털 通信에서는 복호화라 한다.

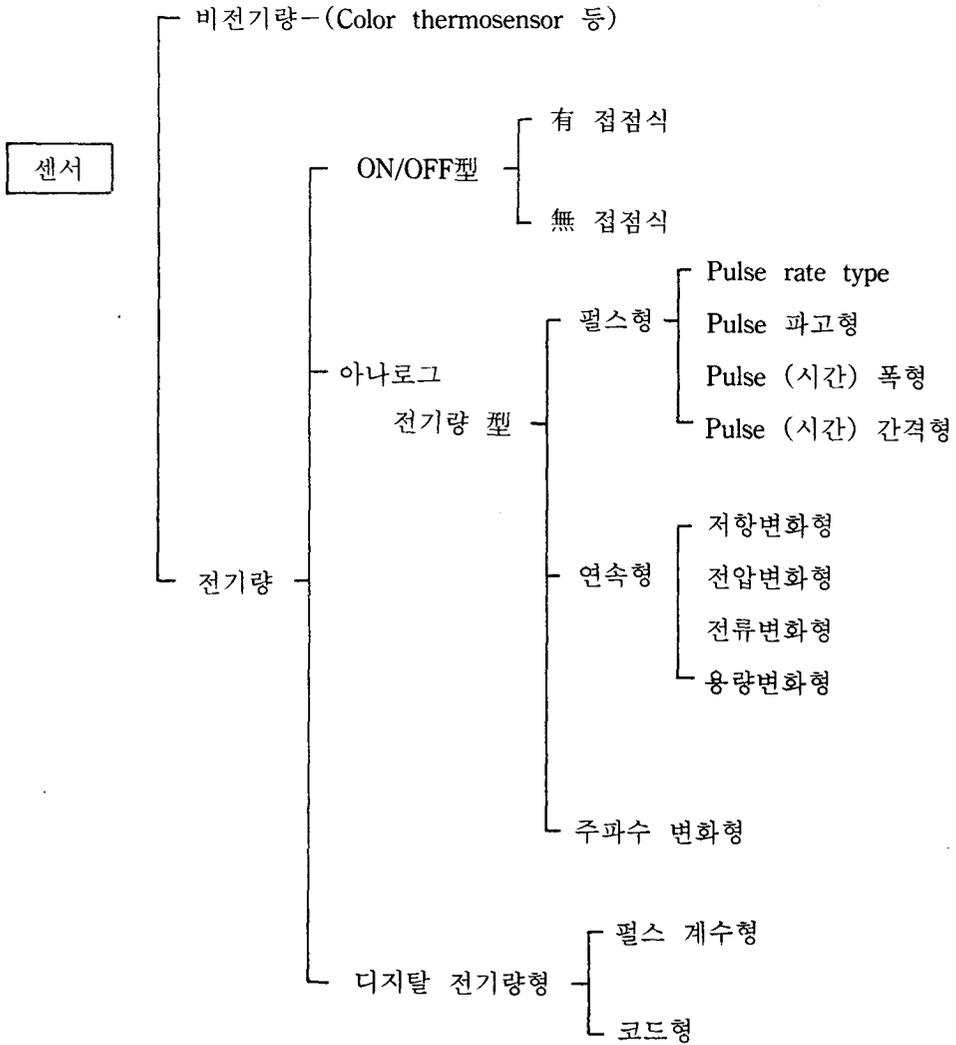


Table 1-1 Formating of output signal

과거에는 Self recording 식으로 記錄후에 資料를 회수시키는 方法이 주 였으나 지금은 실시간(Real time)의 資料 取得 形態로 바뀌어 가고 있다. 즉 센서의 出力 信號가 人力 Interface에 接續되어 그 情報가 즉시 사용자의 컴퓨터로 人力되도록 되어 있어 매우 便利하게 되어 있다. 그러나 이러한 간단한 理論은 그 內容이

여러 段階를 거쳐야 하는 복잡함과 복합적인 技術이 숨어있는 것이다.

Fig. 1-1에서와 같이 본 裝備의 構成은 Sensor 부분, Data logger 부분 및 送信部, 受信部를 거쳐 RS-232C를 통하여 PC에 連結되며 受信된 資料는 컴퓨터가 이해할 수 있는 情報 形式은 Digital 수치 정보이어야 한다. 때문에 Sensor의 出力을 Digital화 시킬 必要가 있는 것이다. 이 機能을 인터페이스 處理라고 한다.

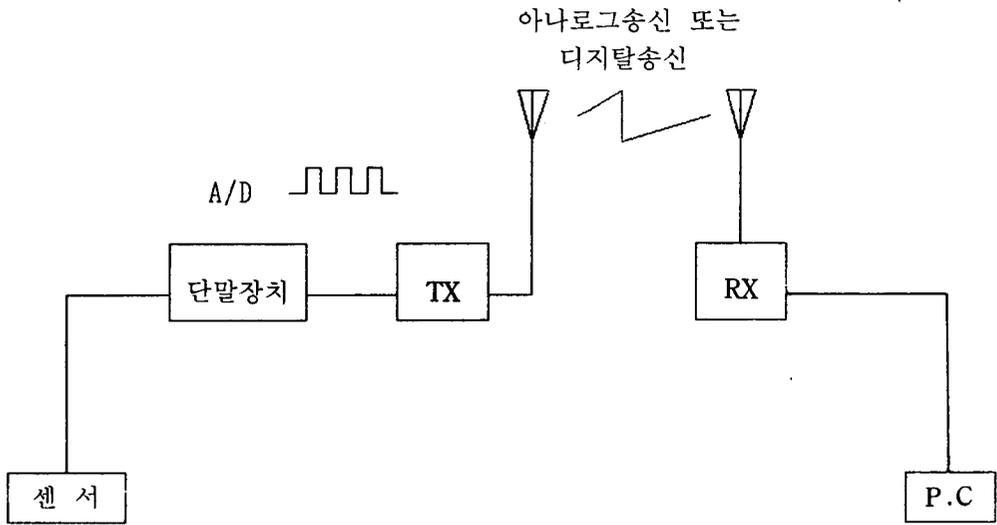


Fig. 1-1 Block diagram of the data transmitting system

2. 센서회로의 구성

水質모니터링 시스템에 사용된 센서는 “日本 東亞電波工業株式會社”에서 제작된 “WQC-2A 水質 Checker”의 센서와 회로를 一部 變更하여 使用하였다.

“WQC-2A 水質 Checker”에서는 센서 選擇스위치로 測定 센서를 選擇하고, 그 出力을 Full scale 100[μ A] 電流計로 測定하였다. 그러나 본 研究에서는 이 測定値를 0[V]~5[V]의 人力 範圍를 가지는 A/D변환기에 使用하기 위하여 增幅회로의 利得을 調整하였다.

1) 溫度

溫度센서는 Thermistor를 사용하였으며 그 특성을 Table 1-2와 Fig. 1-2에 나타내었다. 測定方法是 Thermistor와 直列로 固定抵抗을 연결하고 양단에 3[V]의 정전압을 가하여 溫度에 따른 Thermistor의 전압 변화를 측정하여 온도를 측정하였다. 測定回路는 Fig. 1-7과 같다. 回路에서 U1은 差動增幅器이며, U2는 非反轉增幅器이다. 差動增幅器는 反轉人力端子에 Thermistor 出力 電壓을, 非反轉人力端子에 “0”점 調整 電壓을 加하므로 온도가 0°C일때 出力 電壓을 0[V]로 하고 온도가 上昇함에 따라 出力 電壓이 上昇하도록 하기 위하여 사용되었으며, 非反轉增幅器는 差動增幅器의 出力을 A/D 변환기의 人力 範圍로 증폭하기 위하여 사용되었다.

測定回路의 入·出力電壓의 關係式을 求해보면

$$V = - \left(1 + \frac{R13}{R14} \right) \cdot \frac{R8}{R6} \cdot (V_{th} - V_{0p}) \text{ 이다.}$$

여기서 V_{th} 는 Thermistor 전압이고

V_{0p} 는 “0”점 조정 전압이다.

入·出力 關係式에서 R6, R8, R13, R14는 增幅器의 電壓利得을 결정하는 고정 저항이며, V_{0p} 는 “0”점 조정후에는 고정된 값이므로 增幅器의 出力電壓은 Thermistor의 出力電壓에 따라 변함을 알 수 있다.

온도와 A/D 변환값과의 關係를 Table 1-2, Fig. 1-2에 나타내었다.

Table 1-2. The relationship of temperature/resistance/AD converter

Constant Temperature Bath [°C]	Guildline 9540 PRT Thermometer [°C] (Y)	Thermistor [KΩ]	A/D 변환값 (N)
50	49.50	11.61	3696
45	44.52	12.72	3329
40	39.58	13.88	2950
35	34.57	15.72	2576
30	29.59	17.50	2195
25	24.60	19.42	1813
20	19.60	21.57	1440
15	14.68	24.05	1069
10	9.68	26.82	710
5	4.71	30.15	348
0	-0.40	33.90	26

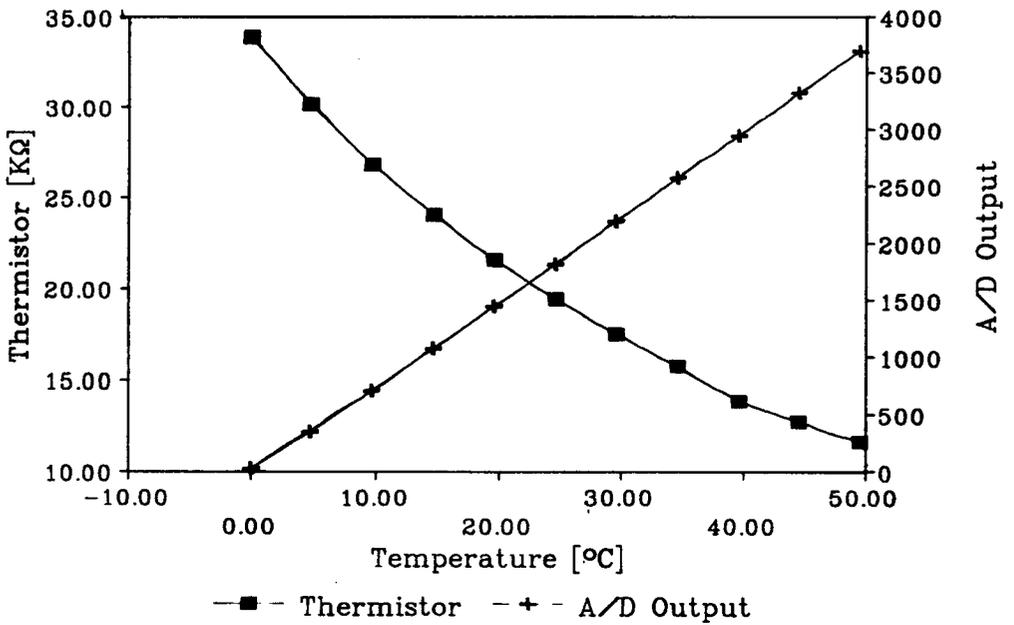


Fig. 1-2. Characteristic curve for the temperature sensor

Table 1-2를 이용하여 온도와 A/D 변환값과의 관계식을 최소자승법을 이용하여 1차식 $Y=A \cdot N+B$ 로 나타내면 상수 A와 B는 각각

$$A=1.340 \times 10^{-2}$$

$B=8.347 \times 10^{-2}$ 이다. 이때의 오차 범위를 Table 1-3, Fig. 1-3에 나타내었다.

Table 1-3. Calibration data for the temperature sensor

실측치 [°C]	A/D 변환값	교정치 [°C]	오차(교정치-실측치) [°C]
-0.04	26	0.43	0.47
4.71	348	4.75	0.04
9.68	710	9.60	-0.08
14.68	1069	14.41	-0.27
19.60	1440	19.38	-0.22
24.60	1813	24.38	-0.22
29.59	2195	29.50	-0.09
34.57	2576	34.60	0.03
39.58	2950	39.62	0.04
44.52	3329	44.70	0.18
49.50	3696	49.62	0.12

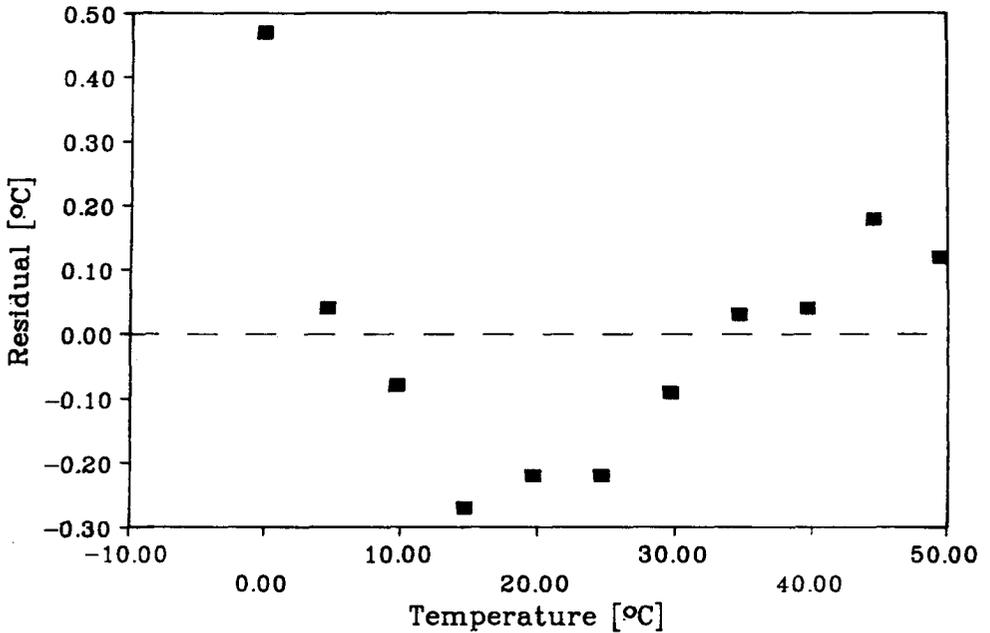


Fig. 1-3. Tolerance values as a function of temperature for the temperature sensor

2) 電氣 傳導度

電氣 傳導度는 電流와 電壓의 比로 표현되므로, 溶液중에서의 電氣 傳導度는 2개 또는 4개의 電極을 이용하여 測定할 수 있으며 보통 4개의 電極이 이용된다. 電極 금속으로는 백금이나 은이 箔板, 망 또는 바늘 형태의 모양으로 보통 사용된다. 그러나 溶液중에서 전위를 측정하는 경우, 電極에서의 이온 운동이 전기 화학적 반응 때문에 생겨난 分極 電壓 때문에 방해를 받게되어 측정 오차의 원인이 된다.

본 연구에서는 Fig. 1-4와 같이 4개의 電極으로 구성되어 있는 電氣 傳導度 센서를 사용하였다.

측정 방법은 전류전극에 전류를 공급하면 海水의 導電率에 따라 전압전극 양단의 전압이 변한다. 이 때 전압전극 양단의 전압을 일정하게 유지하기 위하여 필요한 전류량의 변화를 측정하므로 電氣 傳導度를 측정하였다.

측정회로는 Fig. 1-8이며 전원 발생부와 전류 측정회로로 나누어져 있다. 전원 발생부는 演算 增幅器(U1)를 사용한 Wien Bridge 발진회로를 사용하여 약 200[Hz]의 교류 전원을 발생하여 전극 양단에 가하였으며, 전류 측정은 전류-전압 변환회로(U6)를 사용하여 전류를 전압으로 변환한후 반파정류회로(U5)를 사용하여 整流하여 直流로 변환한후 측정하였다. 반파정류회로는 演算 增幅器를 사용하였으며 계환 소자에 Thermistor를 사용하여 온도 보상을 하였다.

電氣 傳導度와 A/D 변환값과의 관계식을 최소자승법을 이용하여 4차식 $Y=A + B \cdot N + C \cdot N^2 + D \cdot N^3 + E \cdot N^4$ 으로 나타내면 상수

$$A = 3.855 \times 10^{-1}$$

$$B = 2.624 \times 10^{-3}$$

$$C = 1.078 \times 10^{-5}$$

$$D = -4.708 \times 10^{-9}$$

$E = 7.075 \times 10^{-13}$ 이며 그 결과를 Table 1-4와 Fig. 1-5에 나타내었다.

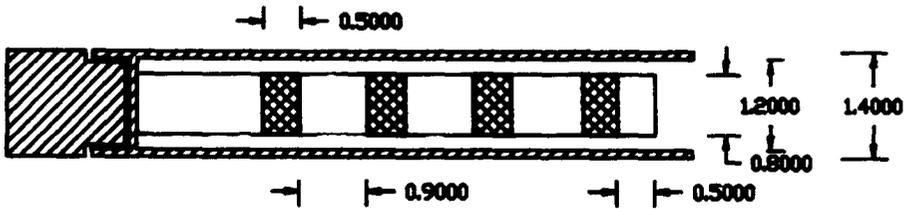


Fig. 1-4. Four-electrode conductivity sensor

Table 1-4. Calibration data for the conductivity sensor

전기전도도 실측치 [mmho] (Y)	A/D 변환값 (N)	교정치 [mmho]	오차(교정치-실측치) [mmho]
44.805	3441	44.474	-0.331
40.561	3293	41.045	0.484
29.323	2547	29.012	-0.311
20.653	1867	20.832	0.179
15.976	1483	16.061	0.085
10.754	1065	10.635	-0.119
0.471	33	0.484	0.013

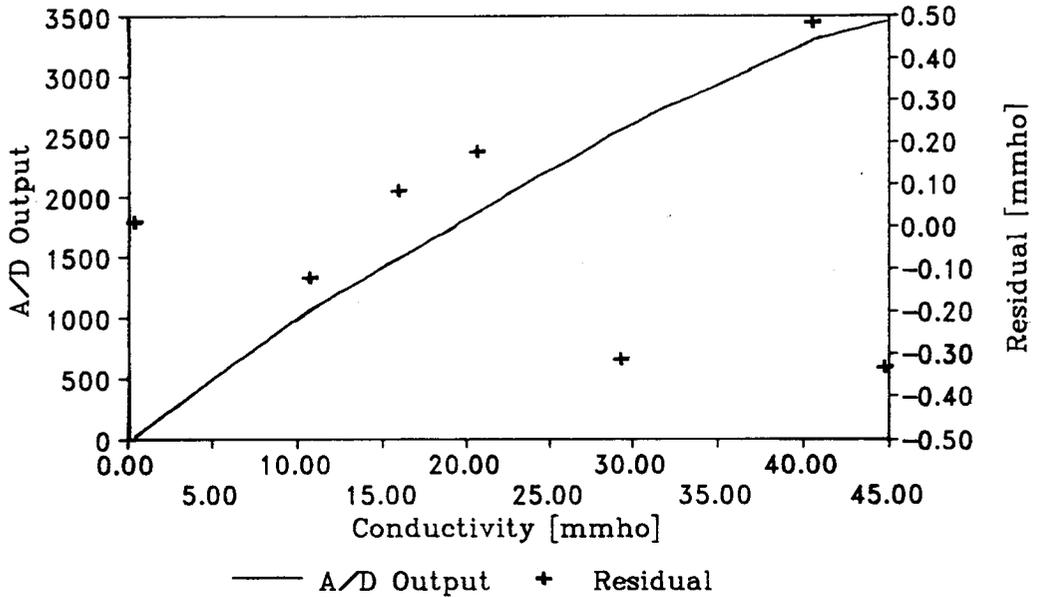


Fig. 1-5. Characteristic curve and tolerance values for the conductivity sensor

3) 溶存 酸素量

溶存 酸素 센서는 Galvanic 電池式 酸素 센서로 作用極으로 銀電極이 對極으로 鉛電極이 쓰였다. 電極에는 酸素가 透過할 수 있는 얇은 Membrane으로 싸여 있고 그 전극과 막 사이에는 강 알칼리성 용액인 수산화칼륨 수용액을 사용하였다. Galvanic 電池式 酸素 센서는 외부에서 전위를 줄 필요가 없고, 센서 자체의 전지 작용에 의한 전위를 이용하여 용존 산소량을 측정하므로 편리한 점도 있으나, 내부액이 측정중에 유출되면 측정 결과에 악영향을 미칠 수 있으므로 내부액의 밀봉을 엄밀하게 하여야 하며 對極으로 鉛電極이 사용된 경우에는 납이 산화되어 소모되므로 센서의 수명을 결정하는 주요한 요소가 된다.

測定時에는 센서 부근의 濃度 勾配를 없애고, Membrane으로 擴散 과정을 균등한 속도로 유지하기 위하여 測定液을 均一하게 攪拌할 필요가 있다. 이는 측정용액중의 용존 산소가 플라스틱막을 擴散하여 전극 표면에서 전기 화학적으로 환원되므로 센서 부근의 산소 분압이 감소하여 용액 내부에 농도 구배가 생기기 때문이다.

飽和 溶存 酸素量은 溫度에 현저한 영향을 받으며, 용액의 농도에 따라서 용존 산소량이 변화한다. 水中에서의 飽和 溶存 酸素量의 溫度 依存性과, 鹽素 이온량 [ppm]에 따른 酸素量의 변화를 Table 1-5와 Fig. 1-6에 나타내었다.

Table 1-5. D.O. calibration values for various temperature & Cl

온 도 [°C]	鹽素이온 100ppm 마다 감소 하는 용존 산소량 [ppm]	水中의 鹽素이온량[ppm]			
		0	10000	20000	30000
0	0.0153	14.15	12.62	11.09	9.56
1	0.0148	13.77	12.29	10.81	9.33
2	0.0144	13.40	11.96	10.52	9.08
3	0.0140	13.04	11.64	10.24	8.84
4	0.0135	12.70	11.35	10.00	8.65
5	0.0131	12.37	11.06	9.75	8.44
6	0.0128	12.06	10.78	9.50	8.22
7	0.0124	11.75	10.51	9.27	8.03
8	0.0120	11.47	10.27	9.07	7.87
9	0.0117	11.19	10.02	8.85	7.68
10	0.0113	10.92	9.79	8.66	7.53
11	0.0110	10.67	9.57	8.47	7.37
12	0.0107	10.43	9.36	8.29	7.22
13	0.0104	10.20	9.16	8.12	7.08
14	0.0101	9.97	8.96	7.95	6.94
15	0.0099	9.76	8.77	7.78	6.79
16	0.0096	9.56	8.60	7.64	6.68
17	0.0094	9.37	8.43	7.49	6.55
18	0.0091	9.18	8.27	7.36	6.45
19	0.0089	9.01	8.12	7.23	6.34
20	0.0087	8.84	7.97	7.10	6.23
21	0.0086	8.68	7.82	6.96	6.10
22	0.0084	8.53	7.69	6.85	6.01
23	0.0082	8.39	7.57	6.75	5.93
24	0.0081	8.25	7.44	6.63	5.82
25	0.0079	8.11	7.32	6.53	5.74
26	0.0078	7.99	7.21	6.43	5.65
27	0.0077	7.87	7.10	6.33	5.56
28	0.0076	7.75	6.99	6.23	5.47
29	0.0076	7.64	6.88	6.12	5.36
30	0.0075	7.53	6.78	6.03	5.28
31	0.0075	7.43	6.68	5.93	5.18
32	0.0074	7.32	6.58	5.84	5.10
33	0.0074	7.23	6.49	5.75	5.01
34	0.0074	7.13	6.39	5.65	4.91
35	0.0074	7.04	6.30	5.56	4.82

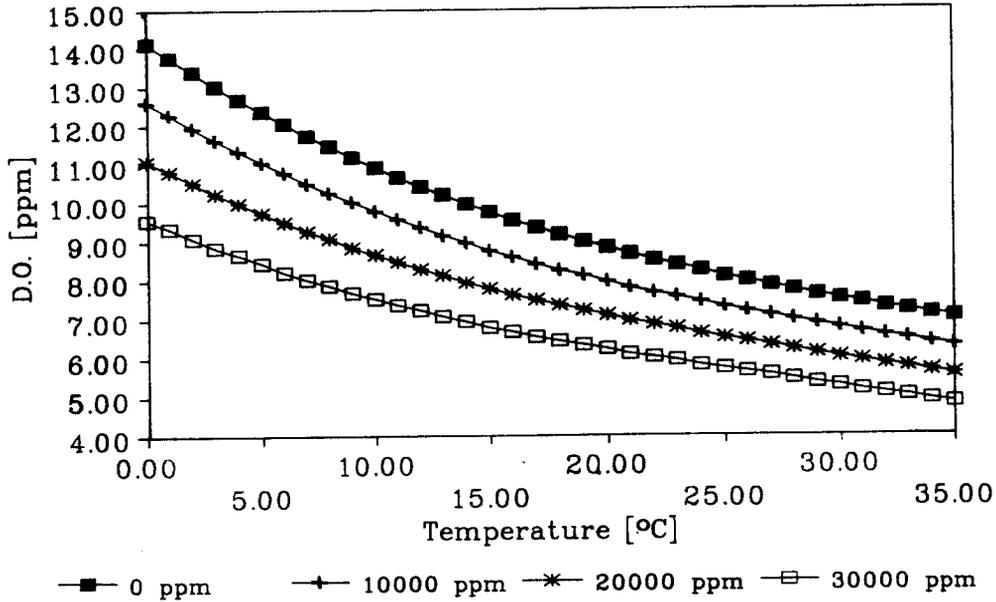
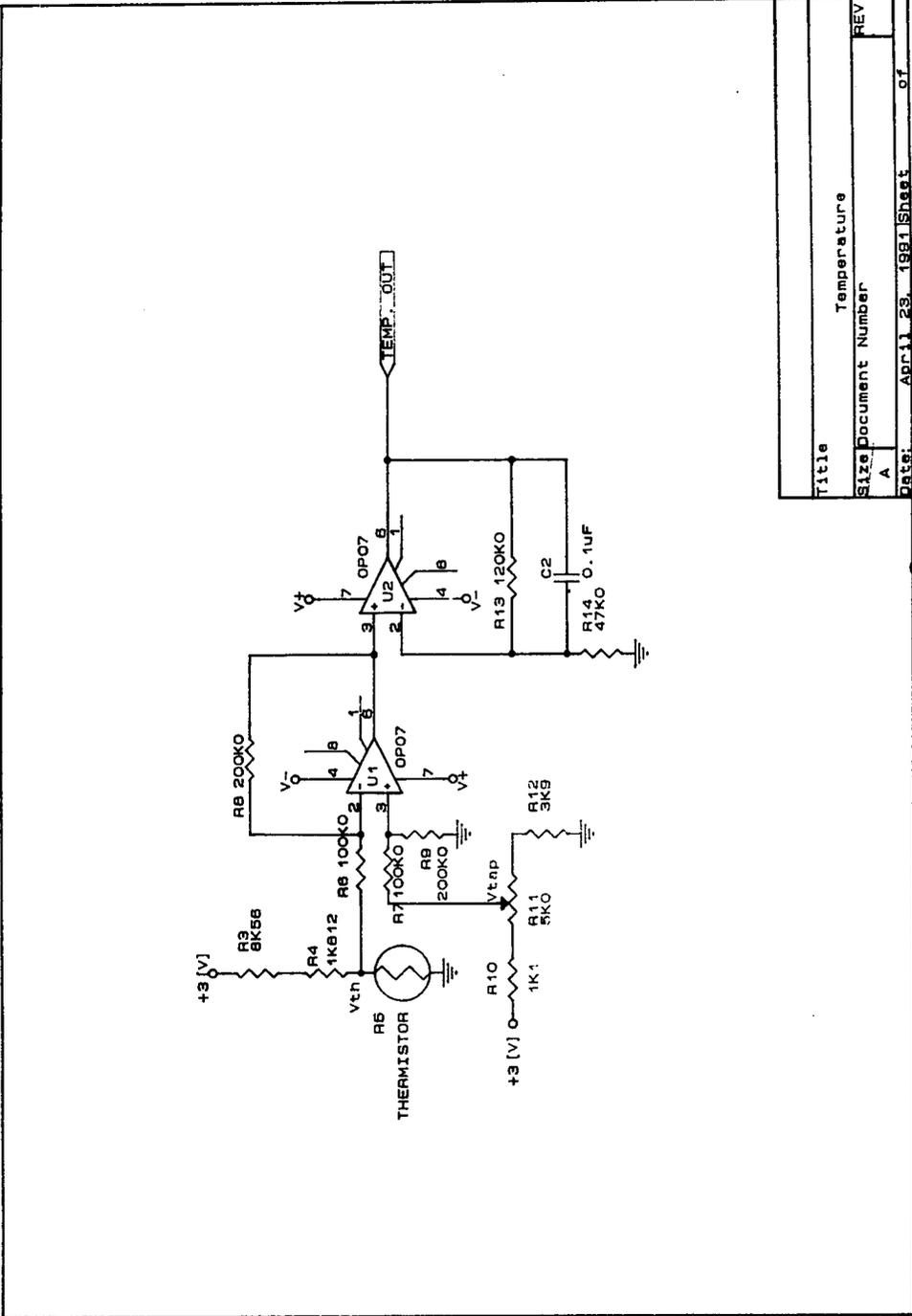


Fig. 1-6. D.O. calibration curve for the temperature & Cl

4) 濁度

濁度 센서로는 光 센서를 利用하였다. 光源으로는 램프를 受光素子로는 光起電力 센서인 포토다이오드를 사용하였다. 포토다이오드는 센서양단을 開放하였을때 開放 電壓은 다이오드의 順方向 전압으로 나타나고, 短絡하였을때는 逆方向 단락 전류로 나타난다. 順방향 電壓은 光量에 대하여 對數적으로 변화하지만, 逆방향 단락 전류는 직선성이 우수하므로 光량을 측정하기 위하여 逆방향 전류를 사용하였다. 逆방향 電流 測定은 演算 增幅器를 사용하여, 전류-전압 변환을 행하면 演算增幅器의 帰환 저항(R_i)과 포토다이오드의 逆방향 短絡電流(I_{sh})의 곱으로 출력 전압- $(I_{sh} \times R_i)$ 을 구하므로써 光量에 比例하는 電壓을 얻을 수 있다.



Title	Temperature
Size Document Number	REV
A	
Date:	April 23, 1991 Sheet
	of

Fig. 1 --7 Temperature sensor circuit

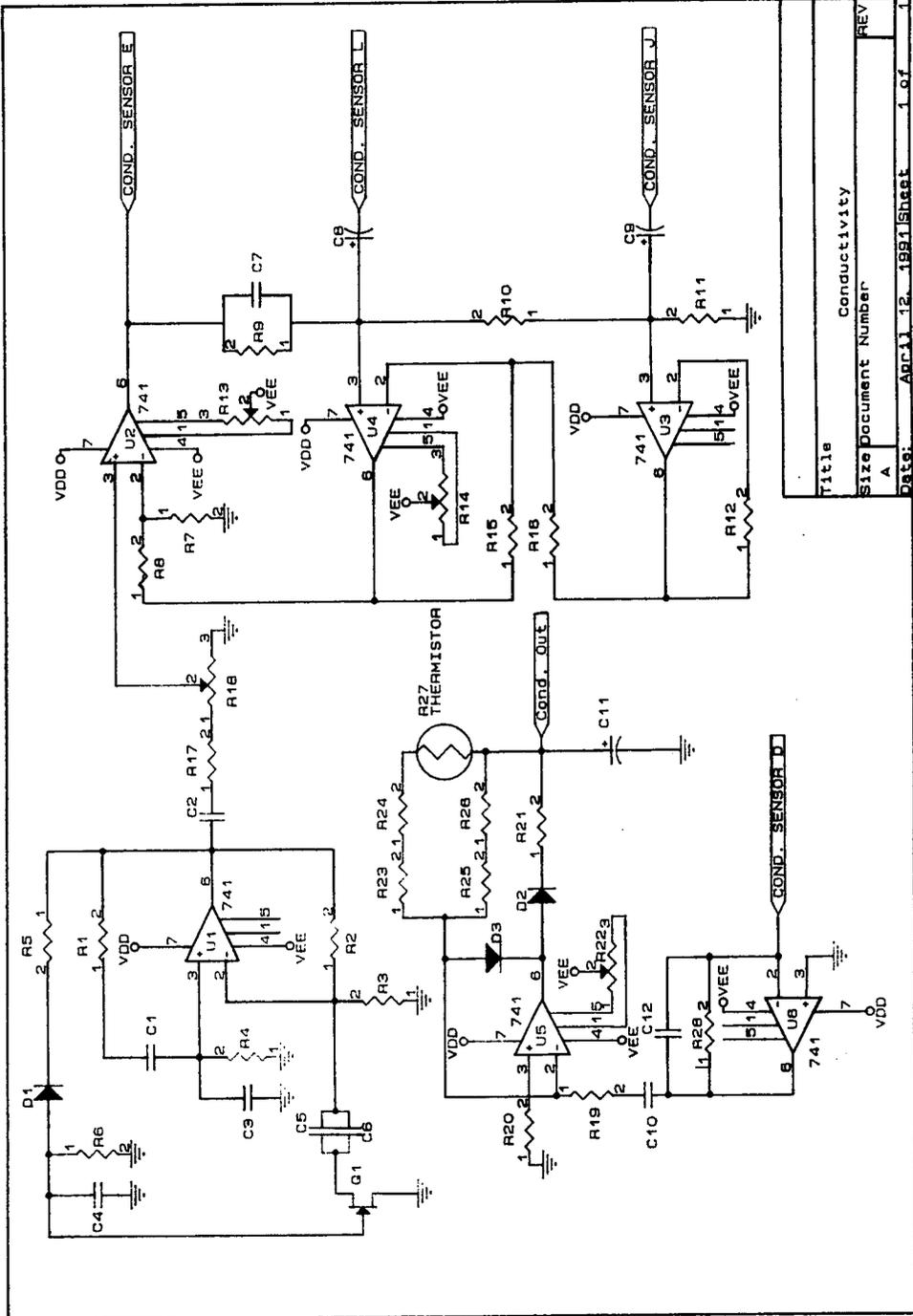


Fig. 1-8 Conductivity sensor circuit

제 2 절 方 法

1. Data Acquisition Board

Data acquisition board는 Clock회로, Fixed data회로, 12 Bits A/D converter회로, Counter회로 및 Parallel to Serial converter회로로 구성 되어 있으며, 그 구성도를 Fig. 2-4에 나타내었다.

출력 자료는 다음과 같은 형태를 가지고 있다.

Fixed Data 1 (Serial Number)

Fixed Data 2 (0000 16진수)

Fixed Data 3 (FFFF 16진수)

Fixed Data 4 (AAAA 16진수)

4 bits Channel Number (0000 2진수)+12 bits A/D Data

4 bits Channel Number (0001 2진수)+12 bits A/D Data

4 bits Channel Number (0010 2진수)+12 bits A/D Data

4 bits Channel Number (0011 2진수)+12 bits A/D Data

4 bits Channel Number (0100 2진수)+12 bits A/D Data

⋮

4 bits Channel Number (0000 2진수)+12 bits A/D Data

4 bits Channel Number (0001 2진수)+12 bits A/D Data

4 bits Channel Number (0010 2진수)+12 bits A/D Data

4 bits Channel Number (0011 2진수)+12 bits A/D Data

4 bits Channel Number (0100 2진수)+12 bits A/D Data

1) Clock 회로

Clock 회로는 A/D converter와 Parallel to Serial converter의 Clock 신호와 관측 주기를 발생하는 회로이다.

그 구성은 Fig. 2-5와 같이 384.0[KHz] Ceramic resonator와 Inverter IC 4069(U1A)를 사용한 발진부와 Counter IC 4017(U4), 4040(U2)를 사용한 분주회로로 구성되어 있다. 발진주파수는 384.0[KHz]이며 이를 분주하여 2400[Hz] 출력을 Parallel to Serial converter의 Clock 신호로, 600[Hz] 출력을 A/D Converter의 Clock 입력으로 사용하였다.

관측 주기를 설정하기 위하여 4040(U2)의 Q12 출력(9.375[Hz])을 4020(U3)의 Clock 신호로 사용하여 각 출력을 Dip switch에 연결하여 선택 가능하도록 하였다. 관측 주기는 1.7초, 3.4초, 6.8초, 13.7초, 27.4초, 54.8초, 1.8분, 3.7분, 7.3분, 14.6분, 29.2분 중에서 선택하여 사용할 수 있다.

2) Fixed Data 회로

Fixed data 회로는 觀測 기기의 固有番號와 試驗 信號를 만드는 회로로 그 회로도들 Fig. 2-6과 Fig. 2-7에 나타내었다. 觀測 기기의 固有番號는 Dip switch(SW 3, SW4)를 사용하여 16비트 수로 任意로 선택 가능하다. 試驗信號는 16비트 Digital 신호로 각 비트를 試驗하기 위한 것으로 試驗신호 0000(16진수)는 3 state Hex Inverter 4502(U6, U7, U8)를 Inhibit상태로 만들어 출력을 모두 Low상태로 하여 사용하였으며, FFFF(16진수)와 AAAA(16진수)는 D Flip-flop 4013(U9)의 출력을 이용하여 만들었다.

Fixed data는 觀測時 固有番號, 0000, FFFF, AAAA순으로 A/D변환 자료가 전송되기 前 약 106.7[msec]동안 전송된다.

3) A/D Converter 회로

A/D Converter 회로는 ADC1210(U14) 12-bit A/D Converter, 4051(U15) 8-Channel analog multiplex, 4510(U17) BCD Counter, 4089(U19) Binary rate multiplier 등으로 구성되어 있으며 그 회로도를 Fig. 2-8에 나타내었다.

ADC1210은 12-bit CMOS 축차비교형 A/D Converter이다. 내부 구성은 비교 전압을 만드는 D/A Converter, 입력전압과 A/D 출력을 비교하는 Comparator, 비교 결과를 기억해두는 축차비교 레지스트(SAR) 및 제어부등으로 구성 되어 있다. 동작원리는 변환 개시 입력(START input : pin 13)이 High 상태에서 Low상태로 된후, 다음 Clock(pin 24)이 Low상태에서 High상태로 변할때 축차비교 레지스터를 Reset시키고, Q11(MSB)을 Low상태로, Q10에서 Q0까지를 High 상태로 만든다. 레지스터의 값은 변환 개시 입력이 High상태가 될 때까지 이 상태를 유지하다가, 변환 개시 입력(pin 13)이 High 상태로 되고, Clock 입력이 Low 상태에서 High상태로 변환될때 변환이 시작되어 비교출력단자(Comparator Output : pin 23)가 High이면 Q11은 High 상태로 변하고, Low이면 Q11도 그대로 Low상태를 유지한다. 그리고 동시에 다음 Bit인 Q10을 Low상태로 만들고 나머지 Q9에서 Q0까지는 그대로 High 상태를 유지시킨다. 이러한 동작을 Q0(LSB)가 결정될때 까지 반복한다. Q0의 상태가 결정되면 변환종료출력(CONVERSION COMPLETE : pin 14)이 High상태에서 Low상태로 변하면서 변환이 종료되었음을 알린다. 이때 출력은 Complemented-binary로 0.5[V] 이하일때 Logic "1"상태, +5[V]일때 Logic "0"상태를 나타내며, 다음 변환개시입력이 들어올 때까지는 이 상태를 그대로 유지하므로 이때에 Data判讀이 가능하며, Timing diagram을 Fig. 2-1에 나타내었다.

入力은 基準電壓을 +5[V]를 사용하여 0[V]~+5[V] Unipolar, Single-ended로 하고 4051 Multiplexer를 사용하여 8 Channel까지 확장하였으며 Dip switch(SW5)를 사용하여 Channel 수를 변경할 수 있도록 하였다. Channel 선택은 4510 BCD Coun-

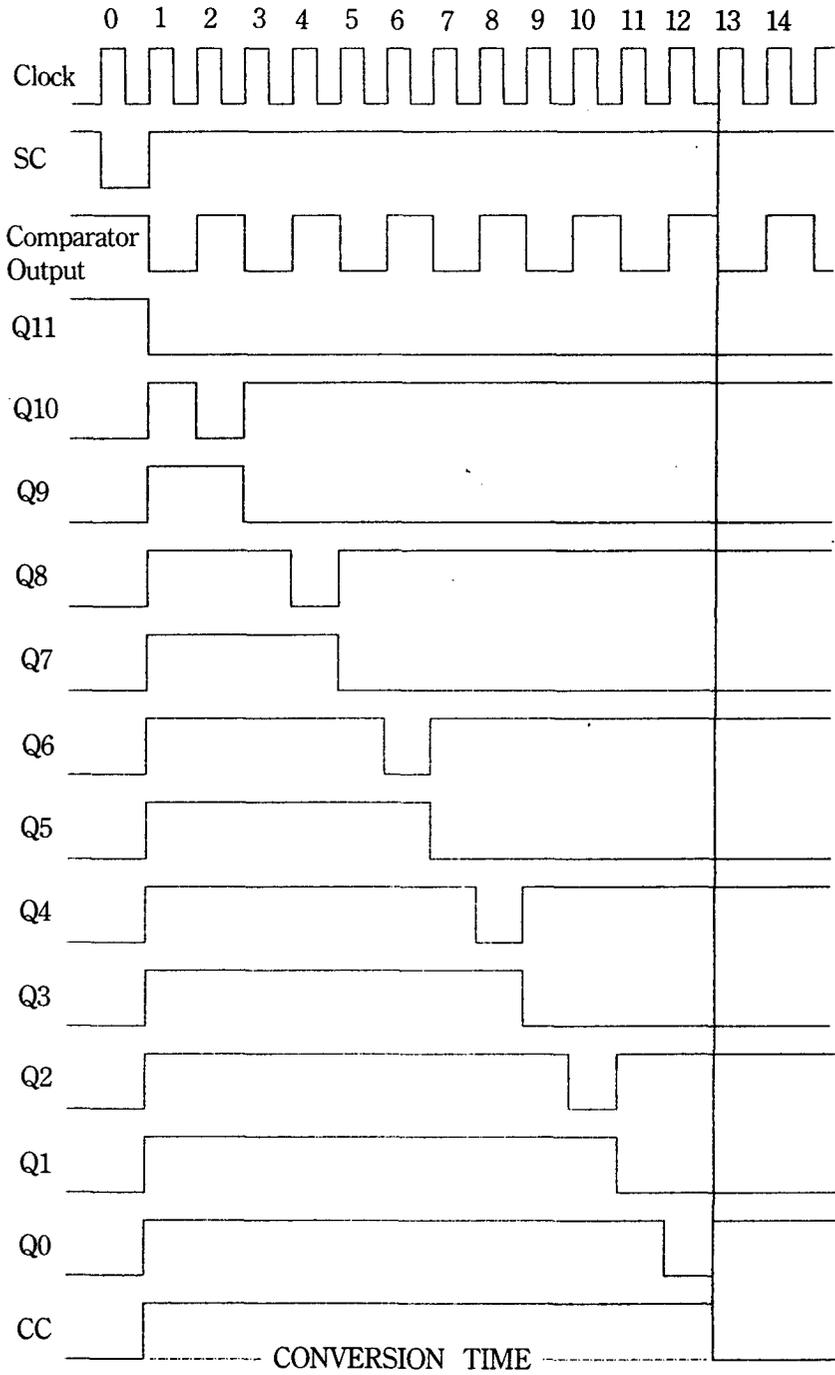


Fig. 2-1. Timing diagram for ADC 1210 A/D converter

ter를 사용하여 A/D Converter의 변환종료출력(pin 14)을 計數하여, 그 출력을 Multiplexer의 Channel Control 입력 신호와 Channel 번호로 사용하였다.

入力電壓대 出力값을 Table 2-1, Fig. 2-2에 나타내어 입출력 특성을 나타내었으며 출력값을 전압으로 환산하여 그 오차를 함께 나타내었다.

Table 2-1. A/D Transfer characteristic and tolerance values

입력전압 [V]	A/D 출력값	A/D출력을 전압으로 환산 [V]	오 차 입력전압 - 환산값 [V]
0.0010	1	0.012	-0.002
0.6250	513	0.6262	-0.0012
1.2500	1025	1.2512	-0.0012
1.8750	1535	1.8738	0.0012
2.5000	2046	2.4976	0.0024
3.1250	2558	3.1226	0.0024
3.7500	3069	3.7463	0.0037
4.3740	3582	4.3726	0.0014
4.9990	4094	4.9976	0.0014

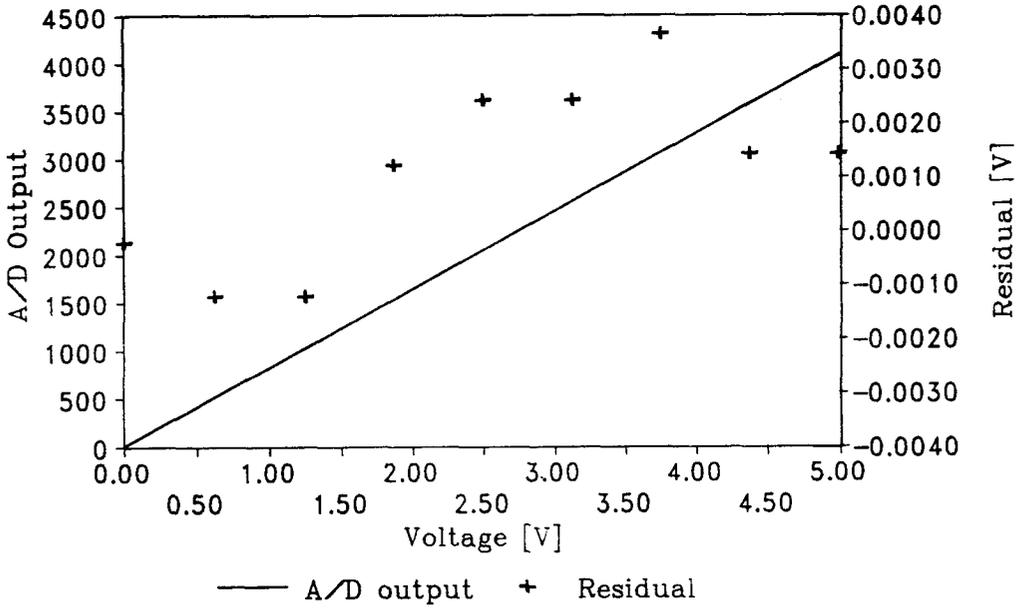


Fig. 2-2. A/D Transfer characteristic curve

4) Counter 회로

Counter 회로는 관측 자료의 수를 계수하여 관측시간을 결정하기 위한 회로이다. 회로 구성은 Fig. 2-9와 같이 4522 BCD Counter(U23, U24, U25, U26) 4개를 직렬로 사용하여 4-Stage programmable down counter 회로를 구성하여 0에서 9999개 까지 관측자료의 수를 선택할 수 있도록 하였다. 동작 원리는 관측주기에 따라서 관측 開始 신호가 들어오면, 이를 이용하여 4522를 Reset하고 BCD Switch(SW6, SW7, SW8, SW9)의 출력을 입력단에 Preset하여 초기치를 설정한후 Clock 수를 計數하여 출력이 "0"이 되면 관측이 종료된다. 여기서 SW9가 최상위 숫자이고 SW6가 최하위 숫자이다.

관측자료의 수는 관측주기를 넘지 않도록 선택하여야 하며 관측 시간은 다음식에 의하여 계산된다.

$$\text{관측시간} = 106.7[\text{msec}] + (133.3[\text{msec}] \times \text{관측자료의 수}).$$

5) Parallel to Serial Converter

Parallel-to-Serial Converter 회로는 16-비트의 병렬 Data를 順次的으로 1비트씩 출력하기 위한 회로로 그 회로를 Fig. 2-10에 나타내었다. 또한 이 출력을 Personal computer의 직렬 포트로 전송하기 위하여 출력 레벨을 CMOS 레벨에서 EIA 레벨로 변환하고, 자료의 출력 형태도 8-비트씩 나누어 8 Data bits, No parity bit, 1 Stop bit로 변형하여 전송하도록 하였으며, 전송 속도는 2400[bps]로 하였다. 출력 자료 형태를 Fig. 2-3에 나타내었으며, “Mark”는 Signal ground에 대해서 $-3[V_{dc}]$ 에서 $-15[V_{dc}]$ 사이의 전압 레벨이고, Binary state “1”로 표시되며, “Space”는 $+3[V_{dc}]$ 에서 $+15[V_{dc}]$ 사이의 전압이고, Binary state “0”로 표시된다.

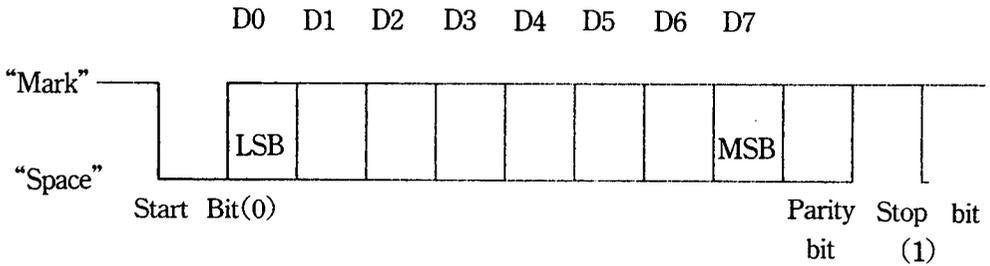
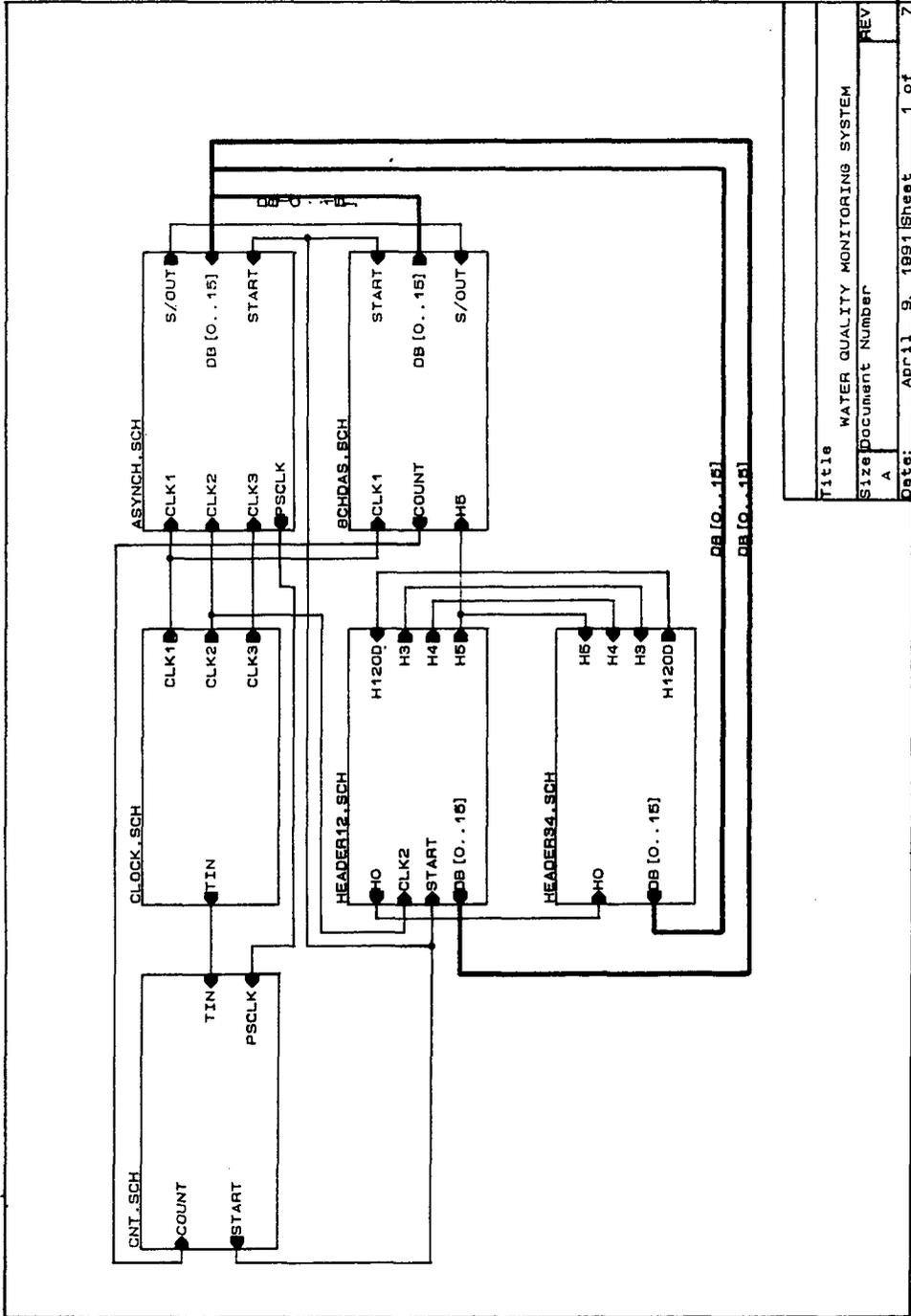
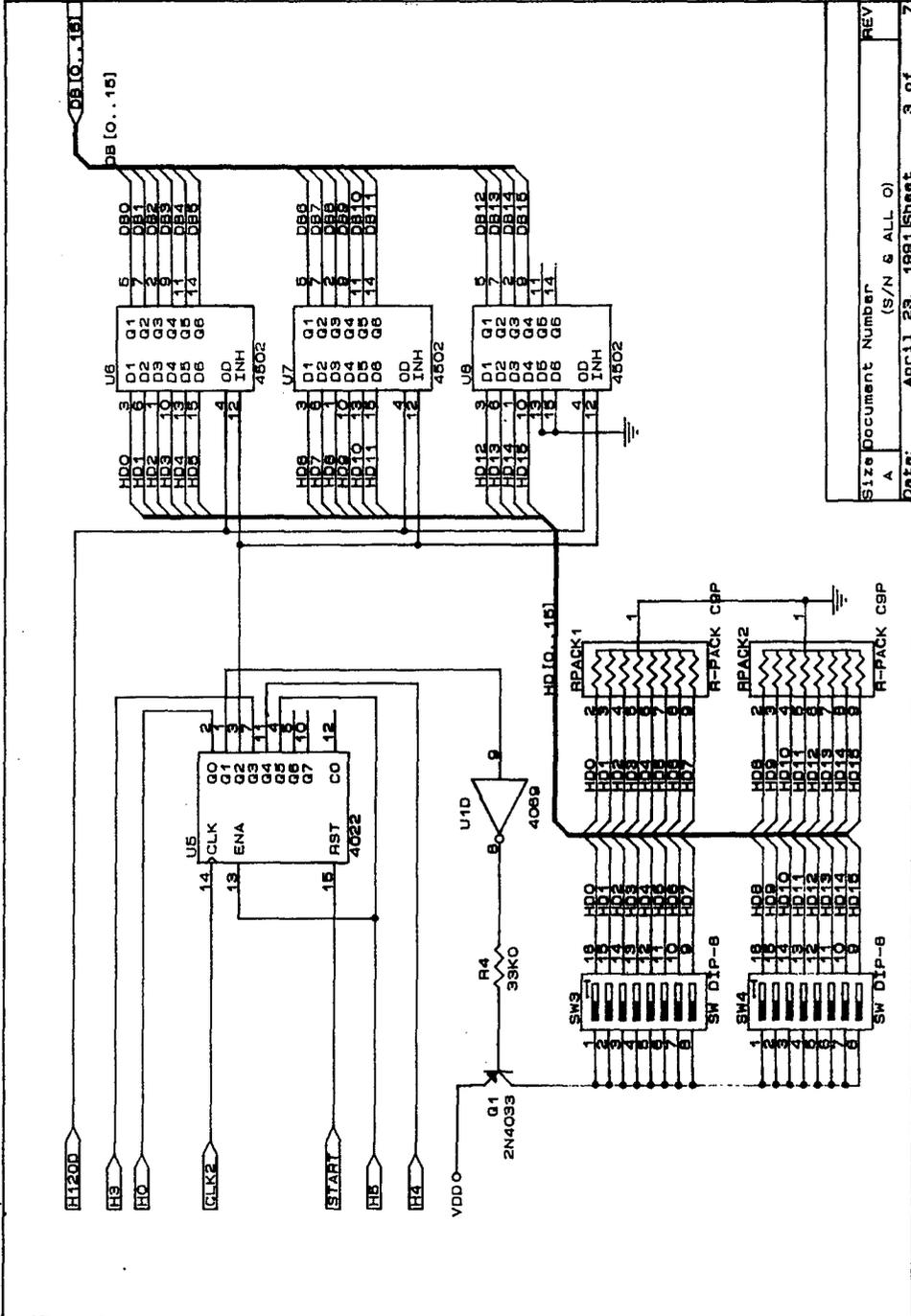


Fig. 2-3. Serial data waveform



Title	WATER QUALITY MONITORING SYSTEM
Size	Document Number
REV	A
Date:	April 9, 1991 Sheet 1 of 7

Fig. 2-4. Block diagram for the water quality monitoring system



Size Document Number	REV
A	
Date: April 23, 1981	Sheet 3 of 7

Fig. 2-6. Fixed data circuit (A)

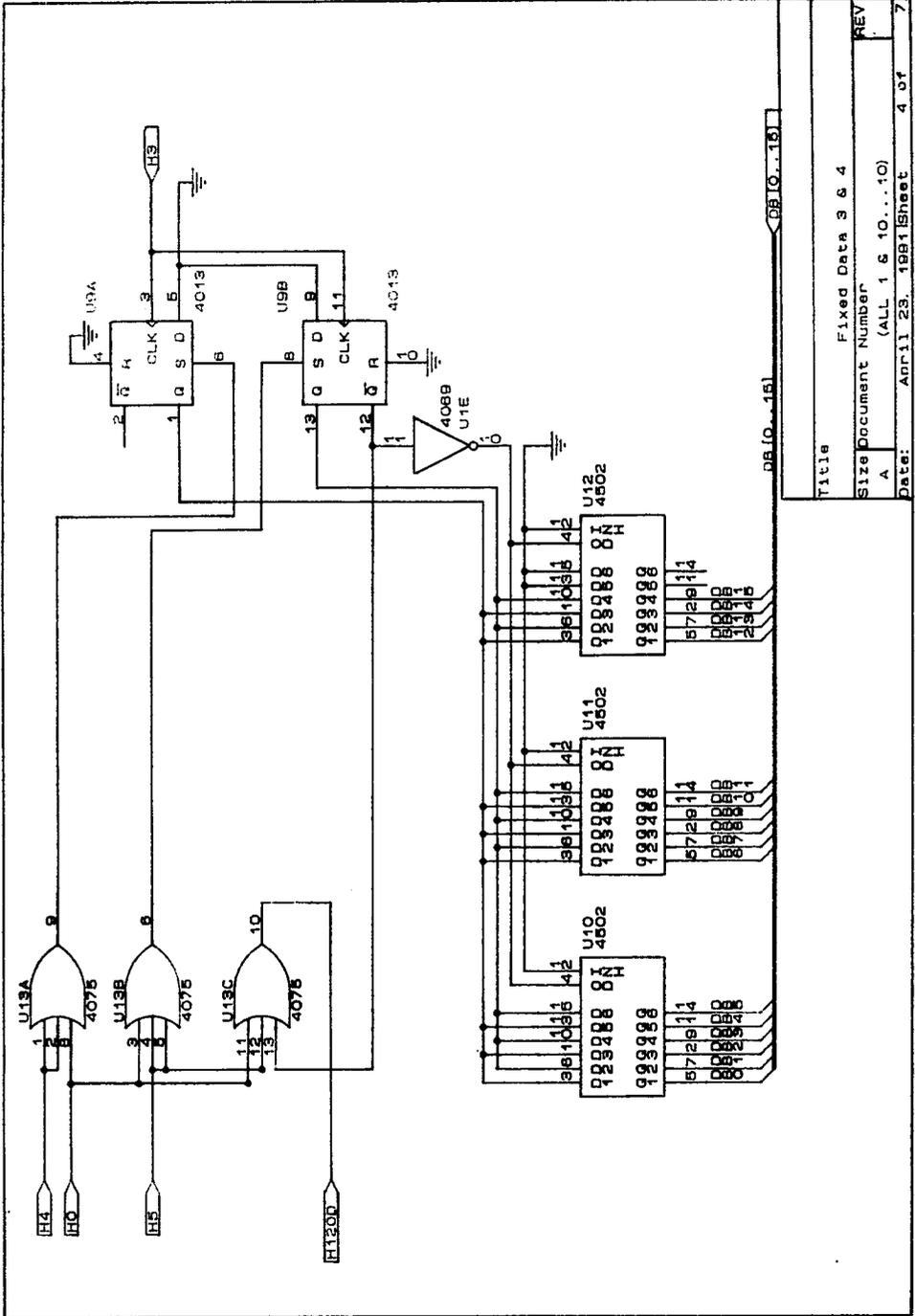
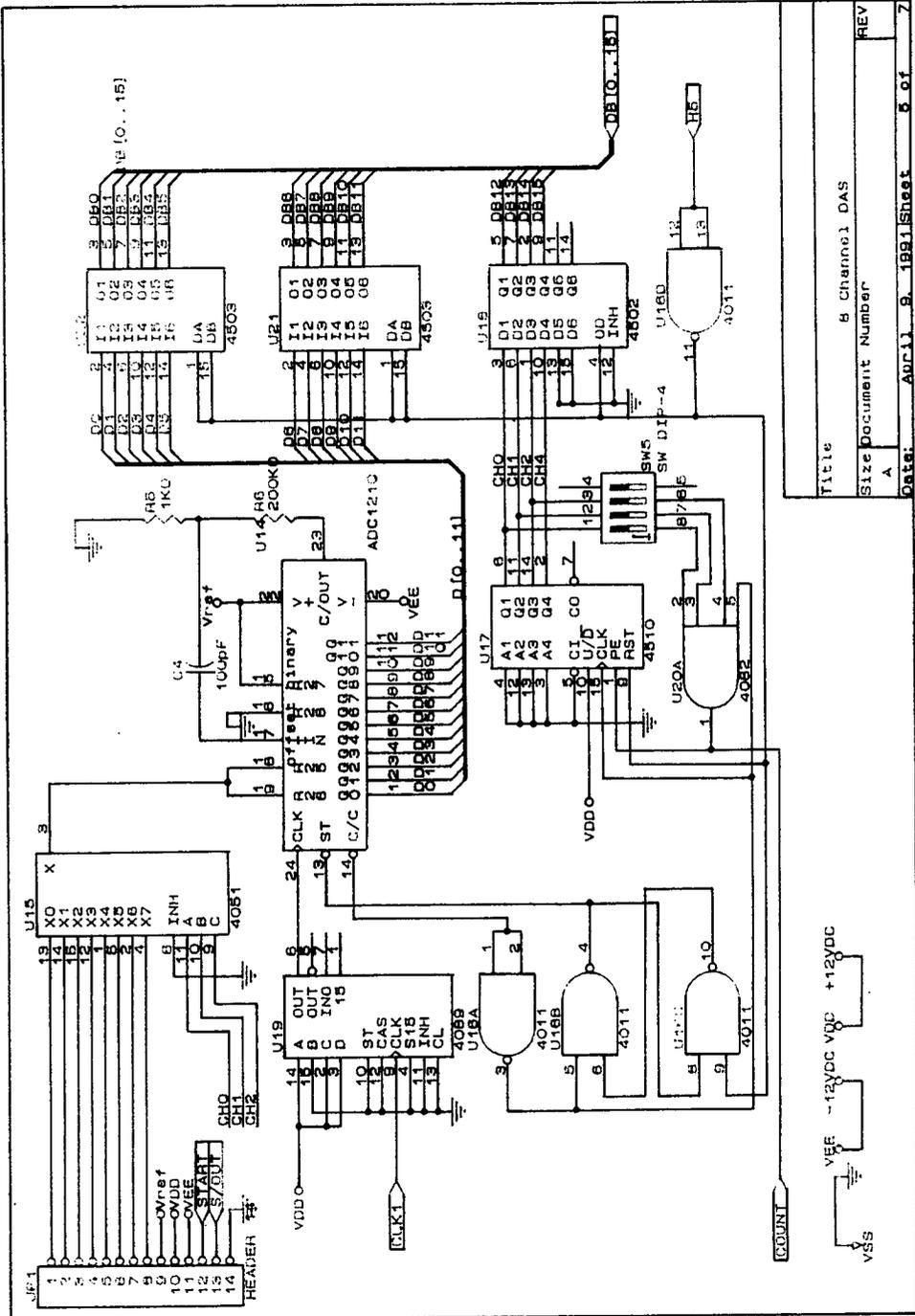


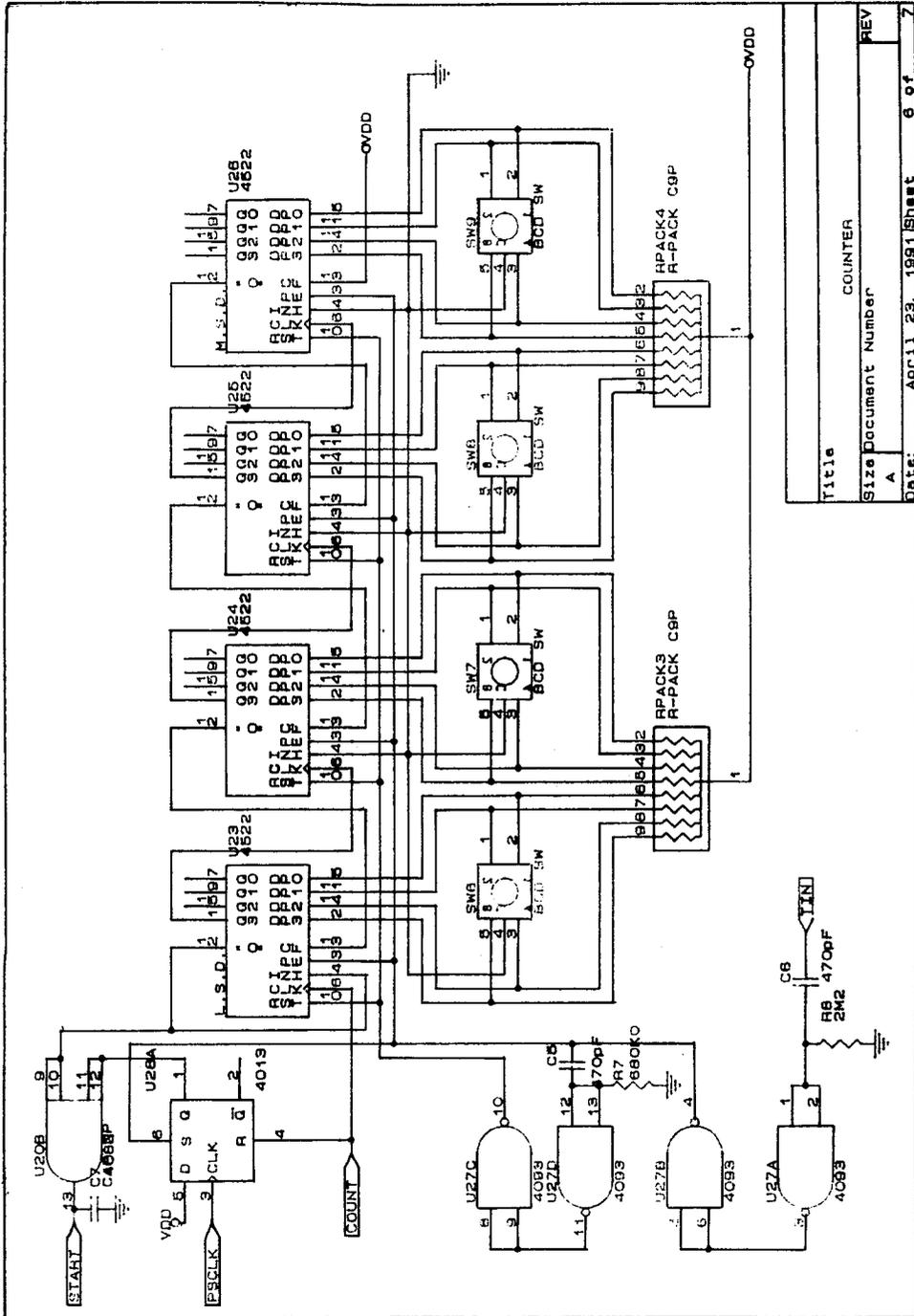
Fig. 2-7. Fixed data circuit (B)

Title		Fixed Data 3 & 4
Size Document Number		A
Date:		April 29, 1981 Sheet 4 of 7



Title	8 Channel DAS
Size	A
Document Number	APC-11
Sheet	5 of 7
REV	

Fig. 2-8. 8 channel A/D converter



Title	COUNTER
Size Document Number	A
Date:	April 23, 1991 Sheet 6 of 7

Fig. 2-9. Counter circuit

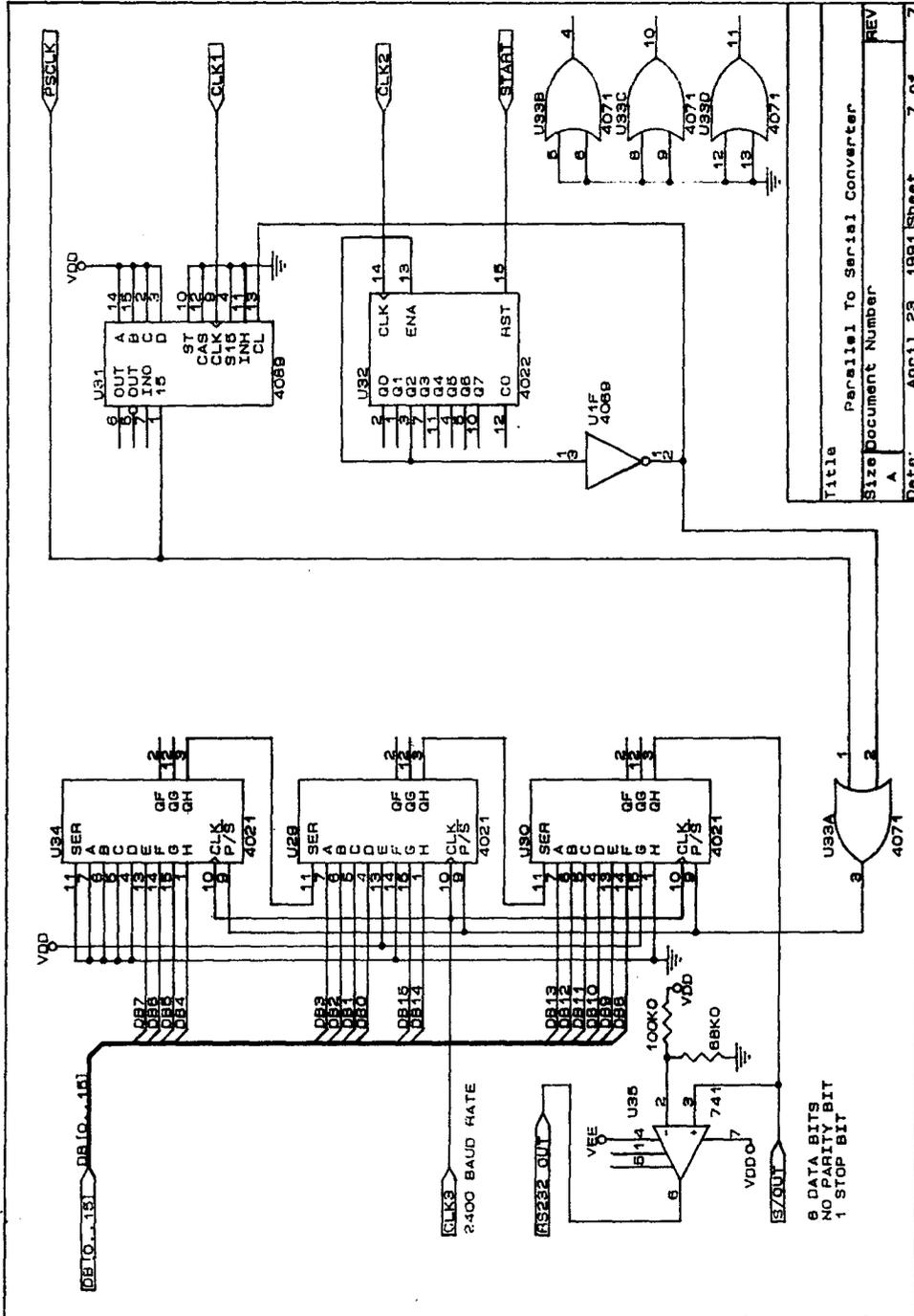


Fig. 2 - 10. Parallel to serial converter

2. 送信部

디지털 통신 시스템에서 필수적 처리 과정인 Formatting은 전송하려는 데이터를 디지털 처리에 적합하도록 변환하는 과정으로서 데이터를 디지털 심볼로 변환하는 과정으로 정의되어지고 원천 코딩은 포메팅에 데이터 압축 과정이 첨가된 신호 처리를 의미한다.

원 신호가 디지털 형태로 된 데이터 통신에서와 달리 디지털 통신은 디지털 기술을 이용하여 아날로그 정보를 전송한다. 아날로그 신호에서 디지털 신호로 변환된 정보를 다시 아날로그 신호의 형태로 변조되어 전송한다.

受信段에서 아날로그 신호는 디지털 신호로서 PC로 자료처리가 될 수 있도록 신호처리 과정을 만들어주었다(Fig. 1-1 참조).

이미 제2장 재료 및 방법 중 System의 구성 및 특성에서 언급하였듯이 센서에서 검출된 아날로그 신호를 디지털 신호로 바꾸어 146.2[MHz] 대역 FM방식으로 디지털 신호를 변조하여 송신하였다.

수신측에서 보면 受信部 出力에서 거의 디지털 형태의 신호가 검파되어 나타나며 이를 다시 과형 보정 회로를 거쳐 완전한 데이터의 형태로 만들어 주었다.

이 과정을 다시 원리적으로 분석하여 본다면,

AM 전송은 보내고자 하는 신호 $f(t)$ 에 의해서 반송파의 진폭을 변화시켜서 얻은 것이었다. 또한 정현파는 진폭, 주파수, 위상을 변화시켜 보낼 수가 있다.

따라서 정현파의 진폭을 일정하게 유지하고 정현파의 주파수를 메시지 신호 $f(t)$ 에 의해서 변하게 하면 주파수 변조(Frequency Modulation ; FM)파가 된다.

즉, 반송파의 각을 변화시켜서 정보를 전송할 수 있으며 보내고자 하는 신호에 따라서 어떠한 형태로 반송파의 각을 변화시키는 변조 방식을 각 변조(Angle modulation)라 하며 일반적으로 각변조에는 위상변조(Phase modulation : PM)와 주파수 변조(Frequency modulation : FM)가 있다.

반송파의 瞬時 周波數(fLo)가 變調 信號(fMo)에 正比例하면 이것은 周波數 變調가 된다. 따라서 周波數 變調는 搬送波의 경우에 瞬時 周波數 ω_i 는

$$\omega_i = \omega_c + K_f \cdot f(t) \dots\dots\dots \textcircled{1} \text{로 표시되고}$$

$$\theta(t) = \int \omega_i dt$$

$$= \omega_c t + K_f \int f(t) dt + \theta_0 \dots\dots\dots \textcircled{2} \text{로 표시된다.}$$

따라서 信號 $A \cos[\omega_c t + \theta_0 + K_f \int f(t) dt]$ 는 周波數 變調된 搬送波를 나타내게 된다.

여기서 ω_c 는 반송주파수이고 K_f 는 상수이며 $f(t)$ 는 메세지 신호이다.

즉 FM에서는 角이 變調 信號를 積分한 값에 正比例한다. 그러므로 먼저 變調信號를 積分한 다음에 이것으로서 搬送波를 位相變調하면 周波數 變調를 얻을 수 있다.

FM 波의 一般式은

$$\Psi_{FM}(t) = A e^{j(\omega_c t + K_f g(t))} \text{이며 } g(t) = \int f(t) dt \text{이다.}$$

瞬時 周波數 ω_i 는

$$\omega_i = \frac{d\theta}{dt} = \omega_c + K_f \frac{dg}{dt} = \omega_c + K_f \cdot f(t) \text{로서}$$

이미 $\textcircled{1}$ 식에서 언급하였다.

여기서 瞬時 周波數는 變調 信號에 比例함을 알 수 있다.

즉 이 식에서 $K_f(t)$ 는 搬送波의 周波數가 중심 周波數 ω_c 로 부터의 편이를 나타내고 있다.

그러므로 상수 K_f 는 搬送波 周波數의 편이량을 定해준다. K_f 의 값이 작으면 周波數 편이는 작아지며 따라서 FM 信號의 Spectrum은 좁은 대역 폭을 갖게 된다. 반대로 K_f 가 큰 값을 가지게 되면 이에따라 周波數 대역폭이 넓어진다.

— FM波의 Spectrum比較

FM의 Spectrum과 AM의 Spectrum을 比較하여 보면 두 變調波 사이에 同一한 點과 次異點이 있음을 알 수 있다.

두 變調波는 다같이 搬送波 成分과 ω_c 를 中心으로 한 側帶波成分이 있는 것은 동일하다.

그러나 FM에서는 측대파는 搬送波에 대하여 90° 位相이 다르게 나타나고 있으나 AM波에서는 측대파 成分의 位相이 搬送波와 同相이다.

즉 AM 信號波는 : $\Psi_{AM}(t) = A \cos \omega_c t + f(t) \cos \omega_c t$ 이고

협대역에서의 FM 및 PM波는 :

$$\begin{aligned} \Psi_{FM}(t) &= A \cos \omega_c t - A K_f g(t) \sin \omega_c t \\ \Psi_{PM}(t) &= A \cos \omega_c t - A K_p f(t) \sin \omega_c t \end{aligned} \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} \text{--- ③}$$

단, K_f 및 K_p 는 상수이다.

이것은 평형 변조기를 사용하여 협대역 FM 및 PM 信號를 發生시킬 수 있는 方法을 암시하고 있으며 Fig. 2-11은 FM波의 發生 블록다이어그램을 나타낸 것이다.

— FM 信號의 發生 및 變調

FM 信號의 發生에는 메세지 信號로 直接 搬送波의 周波數를 變化시키는 직접 周波數 變調 方式과 메세지 信號를 먼저 적분한 다음 이 적분한 信號를 搬送波로 位相 變調하는 간접 周波數 變調 方式이다.

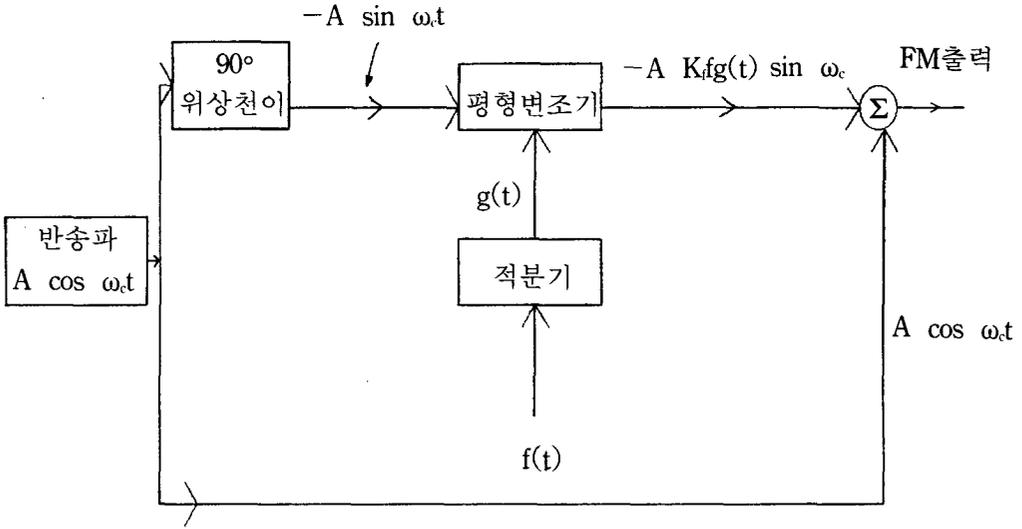


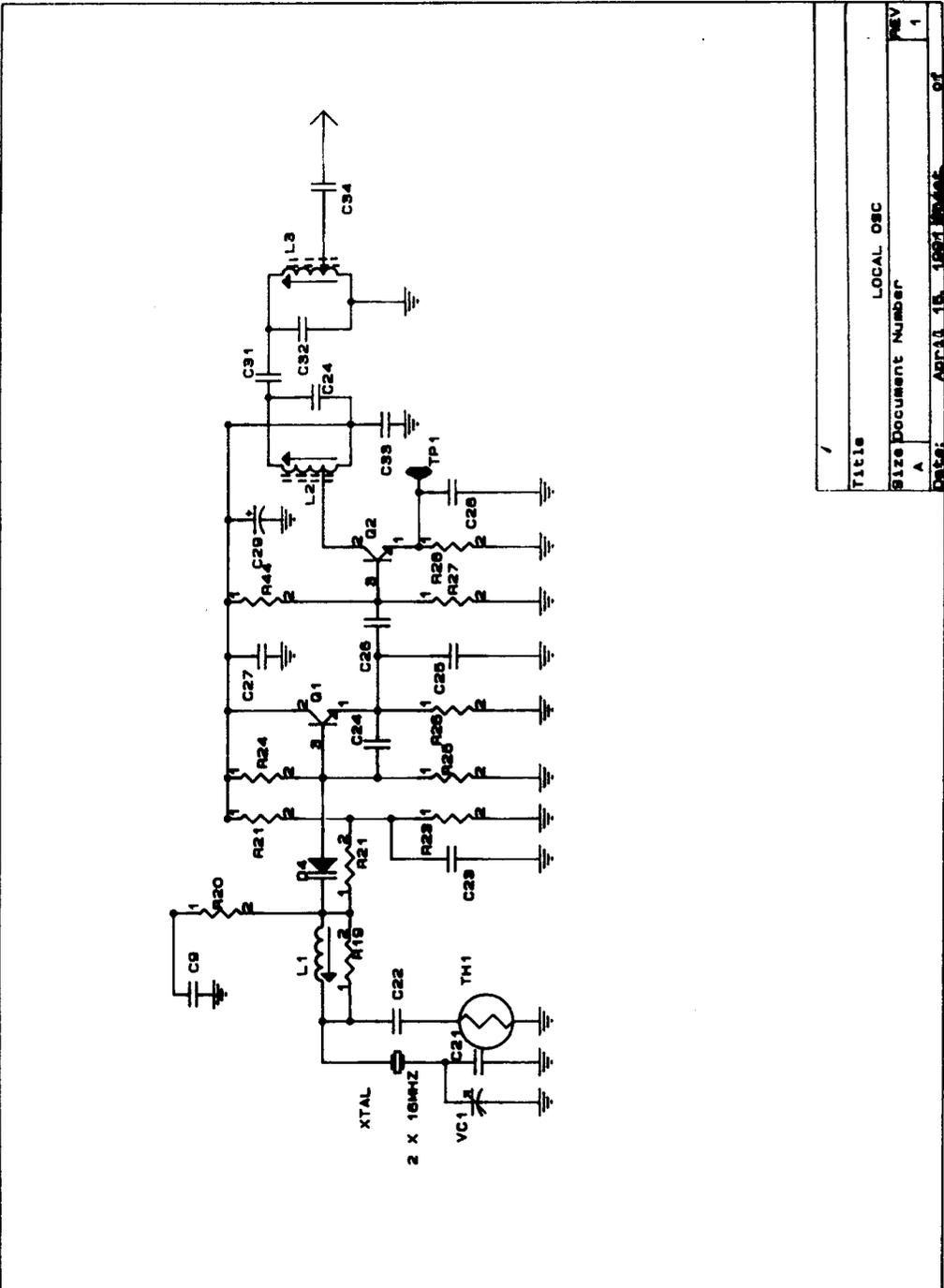
Fig. 2-11. Block diagram of the frequency modulation

Fig. 2-12 송신 국부발진 회로에서와 같이 본 研究에서 채택한 방식은 一般的으로 搬送波를 位相 變調하기가 용이한 間接 FM 變調 方式을 취하였으며 搬送波를 수정 발진기로 부터 얻을 수 있으므로 搬送波의 周波數 變動이 극히 적다. 따라서 이러한 FM變調 方式은 搬送波의 周波數가 표동하기 쉬운 VHF, UHF 대에서 주로 사용된다.

Fig. 2-13에서와 같이 搬送波 發振器의 出力을 둘로 나누어서 한쪽을 平衡 變調器에 넣으면 振幅이 변하는 Vector $AB' \sim AC'$ 를 얻게 된다.

이와같이 하여 얻은 Vector와 90° 位相이 다른 搬送波 發振器 출력 OA과 合成시켜 FM波를 얻게 된다. 이때 변조도 θ 가 커지면 오차가 커져서 일그러짐이 增加한다 (1987, 이충웅).

따라서 변조도를 크게하여 必要한 광대역 FM信號를 벌기 위해서는 Fig. 2-2와 같이 周波數 채배기를 使用하여 변조도를 크게하는 것이다.



Title	LOCAL OSC
Size Document Number	REV
A	1
Date:	April 15, 1957

Fig. 2-12. Local oscillator of the transmitter

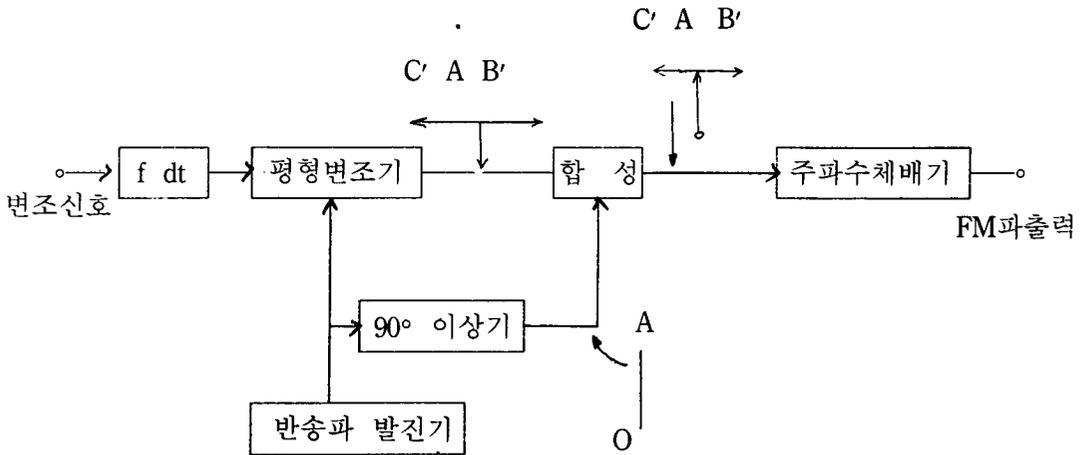


Fig. 2-13. Block diagram of the summing of vector

— 周波數 채배기

周波數 채배기는 入力 信號와 周波數를 遞倍하는 비선형 素子를 使用한 회로이다. 예를들면 간단한 자승 특성을 가진 素子는 周波數를 2배로 채배한다. 자승 素子는 入力 $e_i(t)$ 와 出力 $e_o(t)$ 와의 關係는

$$e_o(t) = K[e_i(t)]^2 \text{ 이며}$$

$$e_i(t) = \Psi_{FM}(t) = \cos [\omega_c t + K_f \int f(t) dt] \text{에서}$$

$$e_o(t) = 1/2[1 + \cos(2 \omega_c t + 2K_f \int f(t) dt)] \text{ 이다.}$$

즉, 入力 信號의 搬送波의 周波數와 周波數 편이가 모두 2배 되었음을 알 수 있다.

또한 周波數 채배는 이와같이 入力와 出力이 자승 특성을 가진 것이 아니더라도 가능하다.

즉, 高周波 增幅器를 B급 또는 C급으로 動作시키면 高周波 增幅器의 出力波에는 고주파 成分이 많이 들어 있게 되므로 고주파 증폭기의 出力 回路에 必要한 배수의 高周波 成分에 同調시킨 공진 回路를 사용하여 入力 信號의 周波數를 채배할 수 있다.

Fig. 2-12에서와 같이 送信器의 局部發振은 수동振動子を 利用한 方式으로 채배를 하지않고 必要한 周波數를 一時에 만들어 낼수가 없기때문에 基本 周波數를 必要 搬送주파수가 되도록 채배 시켜야 한다.

VC1, C21, C22 및 L1에 의하여 Q1에서 發振되는 基本주파수 16.2444[MHz]를 Q1 Emitter에서 測定할 수 있다.

이 信號는 Q2 Collector에서 기본 周波數의 3배수인 기수발진에 공진되도록 L2와 L3의 回路를 構成하였으며 C34를 거친 信號는 Q3에서 다시한번 3채배를 하여 결국 기본 周波數에 대하여 9채배가 된 周波數가 發振되어 Q4에 認可되는 것이다. Fig. 2-14는 송신기의 전체 회로도이다.

Fig. 2-15는 수신기의 국부 발진 회로로서 送信器 국부 발진 方式과 동일하다. 즉, 16.2444[MHz]의 수정 진동자를 콜피츠 발진 원리를 이용하여 Q1, L1, VC1, C1으로 발진을 시킨 후 L2, C3, C4에서 3채배시켜 Q2 Base에 入力된다.

이 信號는 L3와 L4에서 다시 각각 3채배식 모두 9채배 기수 발진이 되어 Mixer 단인 Q3 Base에 입력된다.

Q1에서 발진된 Osc세력은 매우 안정하여야 하며 供給 전원 8[V]~15[V]까지 그 세력이 양호하다.

— 變調器

FM 變調器는 다음과 같은 方式이 있다.

- Diode Reactance 變調器
- 飽和 Reactance 變調器
- Reactance FET 變調器

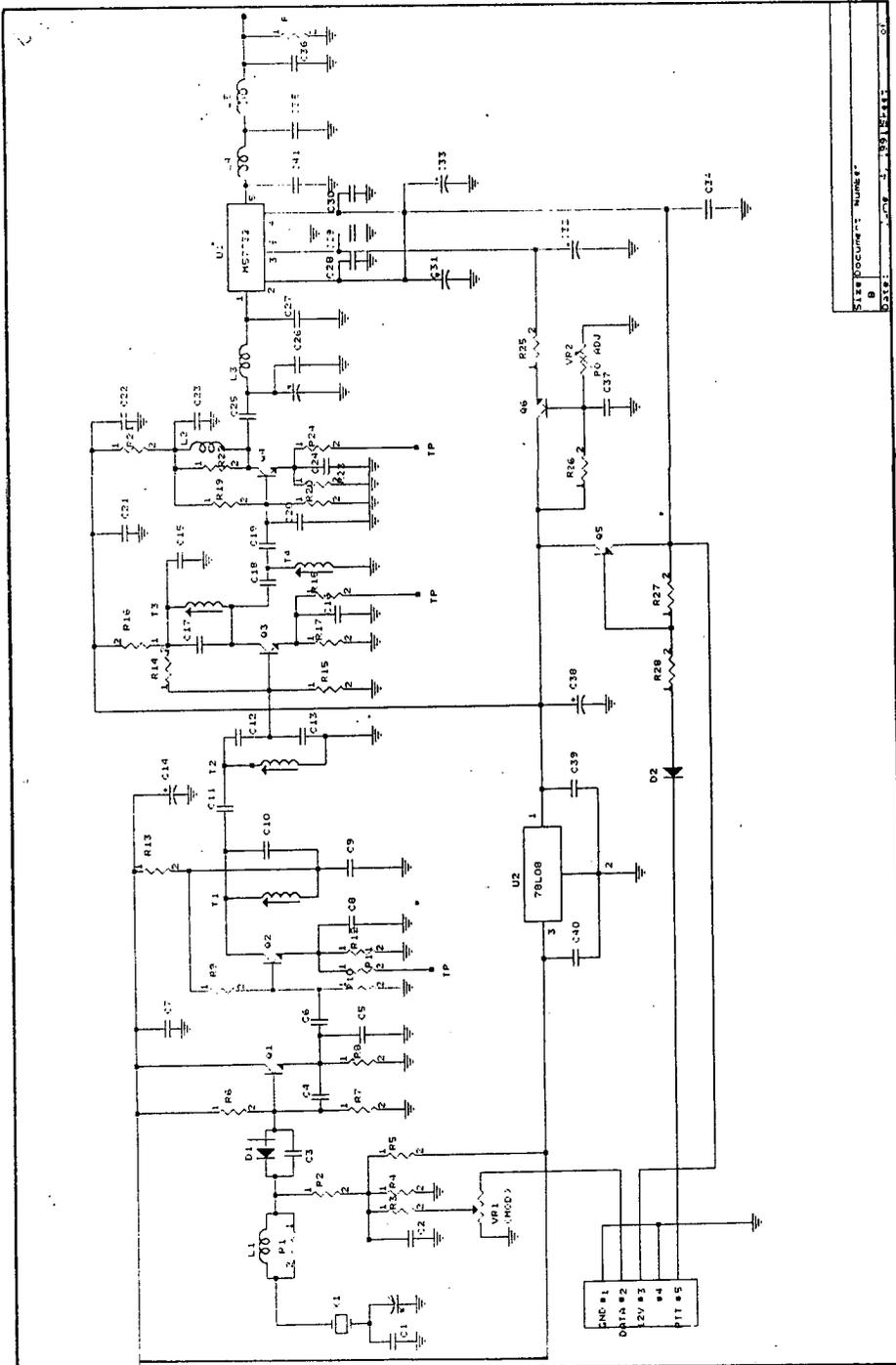
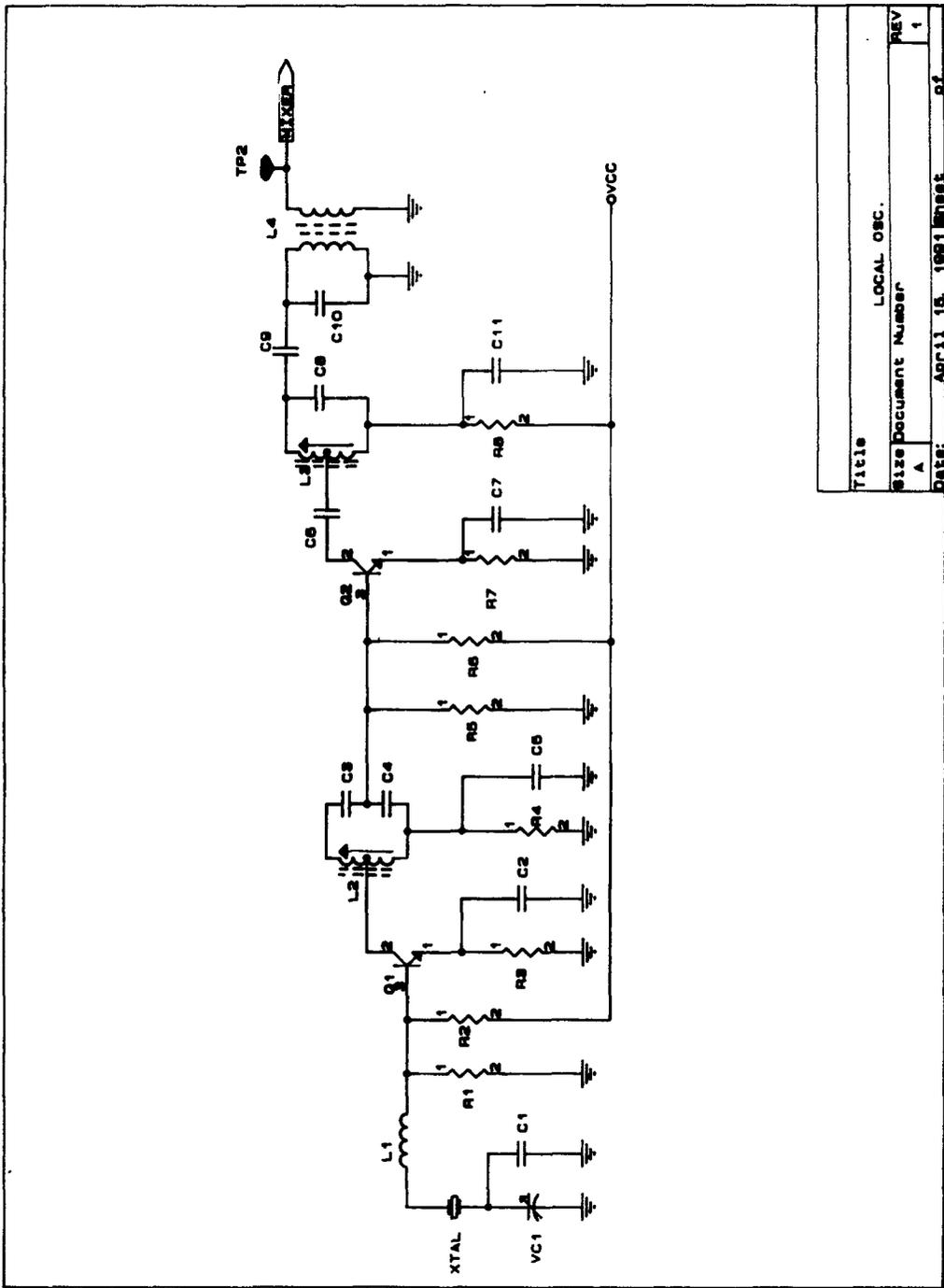


Fig. 2-14. Transmitter circuit



TITLE LOCAL OSC.		
Size Document Number		
A		REV 1
DATE: APR 15 1961 Sheet 02		

Fig. 2-15. Local oscillator of the receiver

◦ Diode Reactance Modulator ;

Varactor Diode를 사용하여 FM變調를 할 수 있다. Diode에 역 方向의 電壓을 걸면 Diode가 접합용량을 가지며 이 접합 용량은 인가 電壓 變動에 따라 변한다.

Fig. 2-12 送信 국부 발진 회로에서와 같이 Q1 Base에 연결된 D4는 R2를 통하여 일정한 電壓이 걸리게 된다. 그러나 메세지 信號가 인가되는 순간 信號에 의한 電壓 變動이 생겨 접합 용량성 다이오드(VVC Diode) D4에 전압 변화를 주게된다. 따라서 용량이 변하게 되어 發振 周波數 變化를 준다. 즉 FM 變調가 되는 것이다.

◦ 飽和 Reactance Modulator ;

포화리액터 變調器(Saturable reactor modulator)에서는 變調 信號 $f(t)$ 에 따라 변하는 Inductor가 사용된다.

이와같은 Reactor로 사용되는 웨라이트 core의 투자율은 외부에서 印加한 자계에 따라 변한다.

코어에는 동조 회로용 Coil과 변조 信號 전류를 흘려 동조시킬 수 있는 두가지 Coil이 감겨 있으며 이 변조 方式은 주파수 체배없이 바로 ± 100 [KHz]의 주파수 편이를 얻을 수 있다.

◦ 리액턴스 FET 변조기

變調 信號 $f(t)$ 에 比例하여 임피던스가 유도성 또는 용량성으로 變化하는 可變 리액턴스 變調 방식이다.

一般的으로 리액턴스 FET 變調 方式에서 搬送波 周波數는 높지 않고 周波數편이(변조지수)도 작다.

따라서 搬送波의 周波數는 周波數 체배 및 周波數 편이를 하여 要求되는 周波數로 높이고 周波數 편이는 周波數 체배로 必要한 만큼 크게한다.

3. 受信部

- 복조(Demodulation)

變調된 信號로 부터 원래의 信號를 찾아내는 것이 복조기(Demodulator)의 역할이다.

復調 課程은 變調(Modulation)의 相對的인 概念으로 搬送波 周波數를 中心으로 이동시켰던 Spectrum을 기저 대역으로 다시 천이시키는 것으로서 出力을 얻을 때 대역통과 필터를 使用함으로써 원래의 信號를 검파할 수 있다.

受信部에서 信號를 복조하는 方式으로는 아나로그 信號의 복조 課程인 두가지 方式 技術 즉, 동기 검파와 포락선 검파 方式이 利用되고 있다(우광방, 황금찬, 1991). 동기검파는 수신부에서 변조된 신호의 반송파 주파수와 같은 정현파를 곱한후 저역 通過 필터에 통과시켜 원래의 信號를 검파하는 方法이고 포락선 검파는 간단한 高周波 成分을 除去하는 방식이다.

- FM 信號의 복조 및 음성 出力

FM 信號의 복조는 먼저 FM 信號를 時間에 관하여 微分한 다음 포락선 검파를 하면 된다는 것을 이미 送信部에서 수식으로 설명하였다.

즉 FM 信號는 변조 信號 f(t)를 적분하여 만들었으며 FM 信號를 복조하려면 FM信號를 미분하여야 한다.

즉 식 ③을 미분하면

$$\begin{aligned} \frac{d\psi_{FM}}{dt} &= -A[\omega_c + K_f f(t)] \sin [\omega_c t + K_f \int f(t) dt] \\ &= -A\omega_c \left(1 + \frac{K_f}{\omega_c} f(t)\right) \sin [\omega_c t + K_f \int f(t) dt] \dots\dots\dots ④ \end{aligned}$$

가 된다. 이 식은 AM 信號波와 비슷한 모양을 하고 있으며 따라서 f(t)를 빼내려면 포락선 검파가 必要함을 알 수 있다.

즉, FM 檢波器의 同調 回路의 共振 周波數 f 는 FM 信號의 搬送 周波數 f 보다 높거나 낮게 정하면 同調 回路에서 共振되는 波형이 단봉형으로 同調 回路 兩端의 電壓은 入力 周波數에 大略 比例하므로 同調 回路 양단의 전압은 AM파로 바뀐다.

이때 FM파에서 AM파로 變化된 AM파의 搬送 周波數 變化는 ④식와 같다.

이때 FM파 때와 同一한 점에 留意하여야 한다.

이와 같이 AM파로 바꾼다음 Diode검파 回路로 포락선을 뽑아내면 $f(t)$ 를 얻게 된다. 이러한 方式을 동조형 FM변별기(Resonance FM discriminator)라고하며 Fig. 2-16, 2-17에서와 같이 近來에 많이 普及되고 있는 TV의 音聲 검파 回路 등에 利用되는 동조 回路는 한개만 있으면 되는 Quadrature FM discriminator를 이용하였다.

이 Quadrature FM 변별기는 앞에서 說明한 바와 같이 입력 信號를 먼저 微分하여 AM파로 바꾼 다음에 AM 검파를 하지않고 다른 方式으로 한 것으로 FM信號를 먼저 PWM(Pulse width modulation)으로 바꾼 다음에 저역 필터로 通過시켜서 Message signal을 뽑아 내는 것이다.

즉 Fig. 2-17에서와 같이 F1 Filter를 거친 중간 주파수는 Q6과 Q7에 의하여 다시한번 다듬어져서 C30을 통하여 MC3316 pin 16으로 入力된다.

이 信號는 pin 1, 2에 인가된 2nd osc인 10.245[MHz]의 발진 세력과 혼합하여 pin 3번을 통하여 CF 101 Filter를 거쳐 Limiter 回路로 認可되며 R126, L113 등에 의해 共振되어 복조기를 거쳐 pin 9번인 音聲 出力으로 나타난다.

여기서 波형을 分析하여 보면 送信 Carrier에 Digital 形態의 信號가 실리어서 轉送되므로 受信된 信號는 다시 Digital pulse의 모양으로 나타난다.

약 0.5[V_{p-p}]의 이 信號는 C45를 통하여 AF Gain과 최종 Audio 出力 증폭단인 IC LM 368을 거쳐 C54에서 나타난다(Fig. 2-18은 수신기 전체 회로도 이다).

C54(Audio 최종단)에 나타난 波형은 Fig. 3-1에서와 같이 Pulse波형이 出力될



MC3361

Advance Information

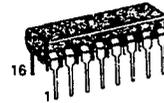
LOW POWER NARROW BAND FM IF

... includes Oscillator, Mixer, Limiting Amplifier, Quadrature Discriminator, Active Filter, Squelch, Scan Control, and Mute Switch. The MC3361 is designed for use in FM dual conversion communications equipment.

- Operates From 1.8 V to 7.0 V
- Low Drain Current 4.0 mA Typ @ VCC = 4.0 Vdc
- Excellent Sensitivity: Input Limiting Voltage — -3.0 dB = 2.0 μ V Typ
- Low Number of External Parts Required

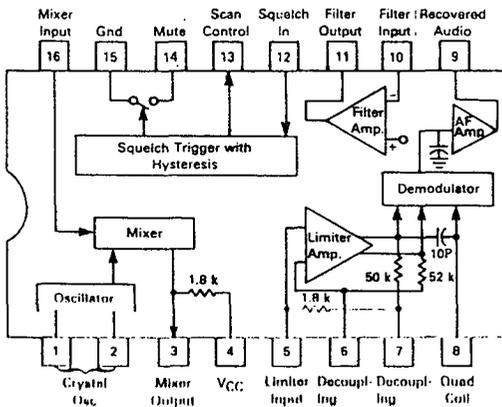
LOW POWER FM IF

SILICON MONOLITHIC INTEGRATED CIRCUIT

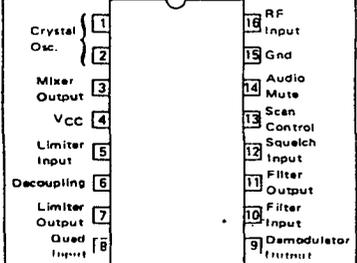


**P SUFFIX
PLASTIC PACKAGE
CASE 648-05**

FIGURE 1 — FUNCTIONAL BLOCK DIAGRAM



PIN CONNECTIONS



*All technical data is subject to change without notice.

Fig. 2-16. A circuit of the discriminator

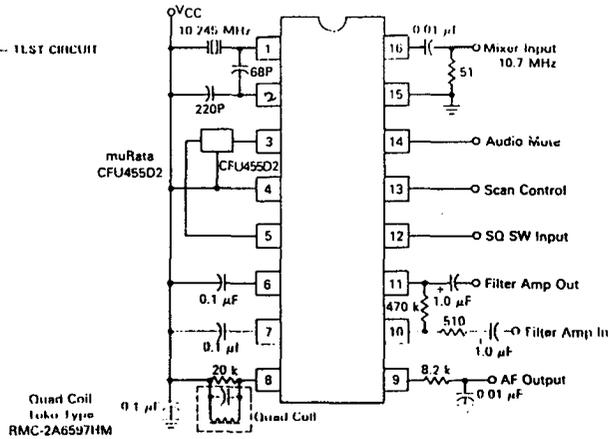
MAXIMUM RATINGS ($T_A = 25^\circ\text{C}$, unless otherwise noted)

Rating	Pin	Symbol	Value	Unit
Power Supply Voltage	4	$V_{CC(max)}$	8.0	Vdc
Operating Supply Voltage Range	4	V_{CC}	1.8 to 7.0	Vdc
Detector Input Voltage	8	—	1.0	Vp-p
Input Voltage ($V_{CC} > 4.0$ Volts)	16	V_{16}	1.0	VRMS
Mute Function	14	V_{14}	-0.5 to 5.0	Vpk
Junction Temperature	—	T_J	150	$^\circ\text{C}$
Operating Ambient Temperature Range	—	T_A	-30 to +70	$^\circ\text{C}$
Storage Temperature Range	—	T_{stg}	-65 to +150	$^\circ\text{C}$

ELECTRICAL CHARACTERISTICS ($V_{CC} = 4.0$ Vdc, $f_o = 10.7$ MHz, $\Delta f = \pm 3.0$ kHz, $f_{mod} = 1.0$ kHz, $T_A = 25^\circ\text{C}$ unless otherwise noted.)

Characteristic	Pin	Min	Typ	Max	Unit
Drain Current Squelch Off Squelch On	4	— —	4.0 6.0	— —	mA
Input Limiting Voltage (-3.0 dB Limiting)	16	—	2.0	—	μV
Detector Output Voltage	9	—	2.0	—	Vdc
Detector Output Impedance	—	—	400	—	Ω
Recovered Audio Output Voltage ($V_{in} = 10$ mV)	9	100	150	—	mVrms
Filter Gain (10 kHz) ($V_{in} = 5.0$ mV)	—	40	48	—	dB
Filter Output Voltage	11	—	1.5	—	Vdc
Trigger Hysteresis	—	—	50	—	mV
Mute Function Low	14	—	10	—	Ω
Mute Function High	14	—	10	—	M Ω
Scan Function Low (Mute Off) ($V_{12} = 2.0$ Vdc)	13	—	—	0.5	Vdc
Scan Function High (Mute On) ($V_{12} = \text{Gnd}$)	13	3.0	—	—	Vdc
Mixer Conversion Gain	3	—	24	—	dB
Mixer Input Resistance	16	—	3.3	—	k Ω
Mixer Input Capacitance	16	—	2.2	—	pF

FIGURE 2 -- 1LS1 CIRCUIT



MOTOROLA LINEAR/INTERFACE DEVICES

Communication Circuits

RF Communications

Narrowband Dual Conversion Receivers	4-48
AM Receiver, Medium-Short Wave	4-48
Wideband Data Receivers	4-48
Narrowband IFs	4-48
Transmitters	4-48
Balanced Modulator/Demodulator	4-49

Telecommunications

PBX Architecture (Analog Transmission)	4-50
Voice/Data Communication (Digital Transmission)	4-52
Electronic Telephone	4-54
Tone Ringers	4-55
Speech Networks	4-56
Speakerphone	4-57
Telephone Accessory	4-58
Modem/Filter	4-59
CVSD Modulator/Demodulator	4-60
Phase-Locked Loop Components	4-61
Package Overview	4-63

RF Communications

Narrowband Dual Conversion Receivers — FM/FSK — VHF

Type	V _{CC}	I _{CC}	Sensitivity	RF Input (Max)	IF1 (Max)	IF2 (limiter in)	Mute	RSSI	Max Data Rate	Notes	Package	Case Suffix
MC3362	2-7 V	3 mA	<1 μ V	180 MHz	10.7 MHz	455 kHz	—	✓	1.2 kb	Includes buffered VCO output	24 Pin DIP, SOIC	P/724 DW/751E
MC3363	2-7 V	4 mA	<1 μ V	180 MHz	10.7 MHz	455 kHz	✓	✓	1.2 kb	Includes RF amp, mute	28 Pin SOIC	DW/751F

AM Receiver Medium/Short Wave

MC13041	6.5-16.5 V	25 mA	6 μ V	10 MHz	455 kHz	—	—	✓	—	Includes scan stop	20 Pin DIP	P/738 DW/751D
---------	------------	-------	-----------	--------	---------	---	---	---	---	--------------------	------------	---------------

Wideband Data (FM/FSK) Receiver — VHF

Type	V _{CC}	I _{CC}	Sensitivity	IF1 (Max)	IF2 (limiter in)	Mute	RSSI	Max Data Rate	Notes	Package	Case Suffix
MC3356	3-9 V	25 mA	30 μ V	200 MHz	10.7 MHz	✓	✓	500 kb	Includes front end mixer/L.O.	20 Pin DIP/PLCC	P/738 FN/775

Narrowband IF's — Wideband (FM/FSK) IF

MC3357	4-8 V	5 mA	5 μ V	45 MHz	455 kHz	✓	—	—		16 Pin DIP/SOIC	P/648 D/751B
MC3359	4-9 V	7 mA	2 μ V	45 MHz	455 kHz	✓	—	—		18 Pin DIP/SOIC	P/707 DW/751C
MC3361	2-8 V	6 mA	2 μ V	60 MHz	455 kHz	✓	—	—		16 Pin DIP/SOIC	P/648 D/751B
MC3367	1-5 V	1 mA	<1 μ V	75 MHz	455 kHz	✓	—	1.2 kb	1 Coll Operation	28 Pin SOIC	DW/751F
MC3371	2-8 V	6 mA	2 μ V	60 MHz	455 kHz	✓	✓	—	(3088 Intro)	16 Pin DIP/SOIC	P/648 D/751B
MC13055	3-12 V	25 mA	20 μ V	—	40 MHz	✓	✓	2 Mb	Wideband Data IF	16 Pin DIP/SOIC	P/648 D/751B

Transmitters — FM/FSK

Type	V _{CC}	I _{CC}	P _{out}	Max RF Freq. Out	Battery Check	Tone OSC	Max Mod. Freq.	Notes	Package	Case Suffix
MC2831A	3-8 Vdc	5 mA	-30 dBm	50 MHz	✓	✓	5.0 kHz (xtal ctrl)	Includes low battery checker, tone osc.	16 Pin DIP/SOIC	P/648 D/751B
MC2833	3-8 Vdc	3 mA	30 dBm to 110 dBm	150 MHz	—	—	5.0 kHz (xtal ctrl)	Includes two frequency multiplier/amplifier transistors	16 Pin DIP/SOIC	P/648 D/751B

Balanced Modulator/Demodulator

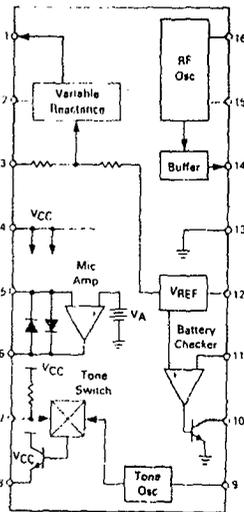
Type	V _{CC}	I _{CC}	Function	Package	Case Suffix
MC1596	5-30 V	10 mA	Carrier Balance >50 dB General purpose balanced modulator/ demodulator for AM, SSB, FM Detection	10 Pin Metal	G/603
MC1496	5-30 V	10 mA		14 Pin Ceramic DIL, DIP, SOIC	L/632 P/646 D/751A

Low Power FM Transmitter System

MC2831A — T_A = -30° to +75°C,
Case 648, 751B

- Complete VHF FM Transmitter/Exciter
- Mike Preamp with Limiting
- Tone Generator for CTSS or AFSK
- Crystal or L-C VCO Operation
- Buffer/Multiplier Output Stage
- Low Voltage (internal reference) Warning Circuit
- Easily Partitioned for Semicustom Applications

MC2831A



Low Voltage FM Narrowband Receiver

MC3367 — T_A = 0°C to +70°C, Case 751F

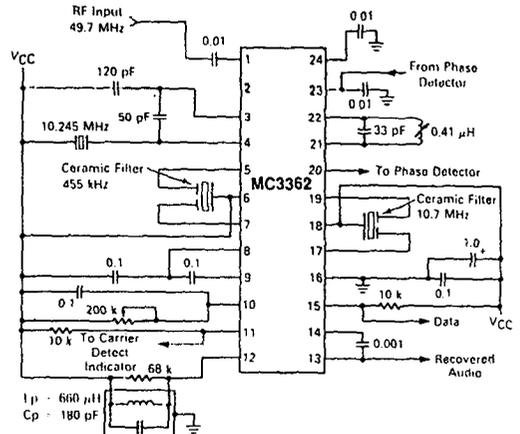
- Single Cell Operation to 0.9 V_{CC}
- Single Conversion Operation to 75 MHz
- Current Drain of 1 mA
- Split I.F. Amplifier for Single or Dual Filters
- Analog and Data Outputs
- Sensitivity of 0.7 μV Typ for 20 dB Quieting
- Low Battery Voltage Indicator

MOSAIC™ 1.5

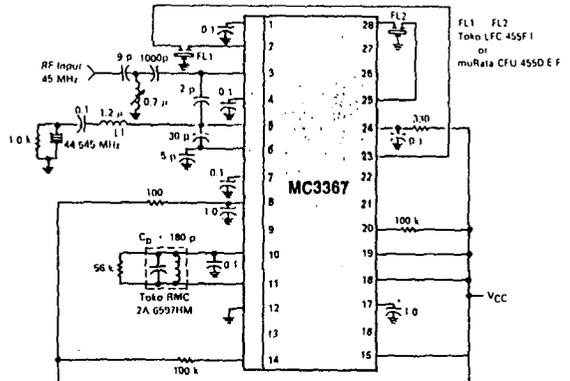
VHF Narrowband Dual-Conversion Receivers

MC3362/MC3363 — T_A = -40°C to +85°C,
Case 724, 751A

- Operation to 180 MHz
- 2-8 V dc Supply
- >1 μV for 20 dB Quieting Sensitivity
- Analog and Data Modulation Recovery
- >60 dB Dynamic Range RSSI
- Crystal or VCO First L.O. Operation
- On-Chip RF Amp/MC3363



(All capacitors in pF unless otherwise stated. Inductors in ohms, Inductors in Henries.)



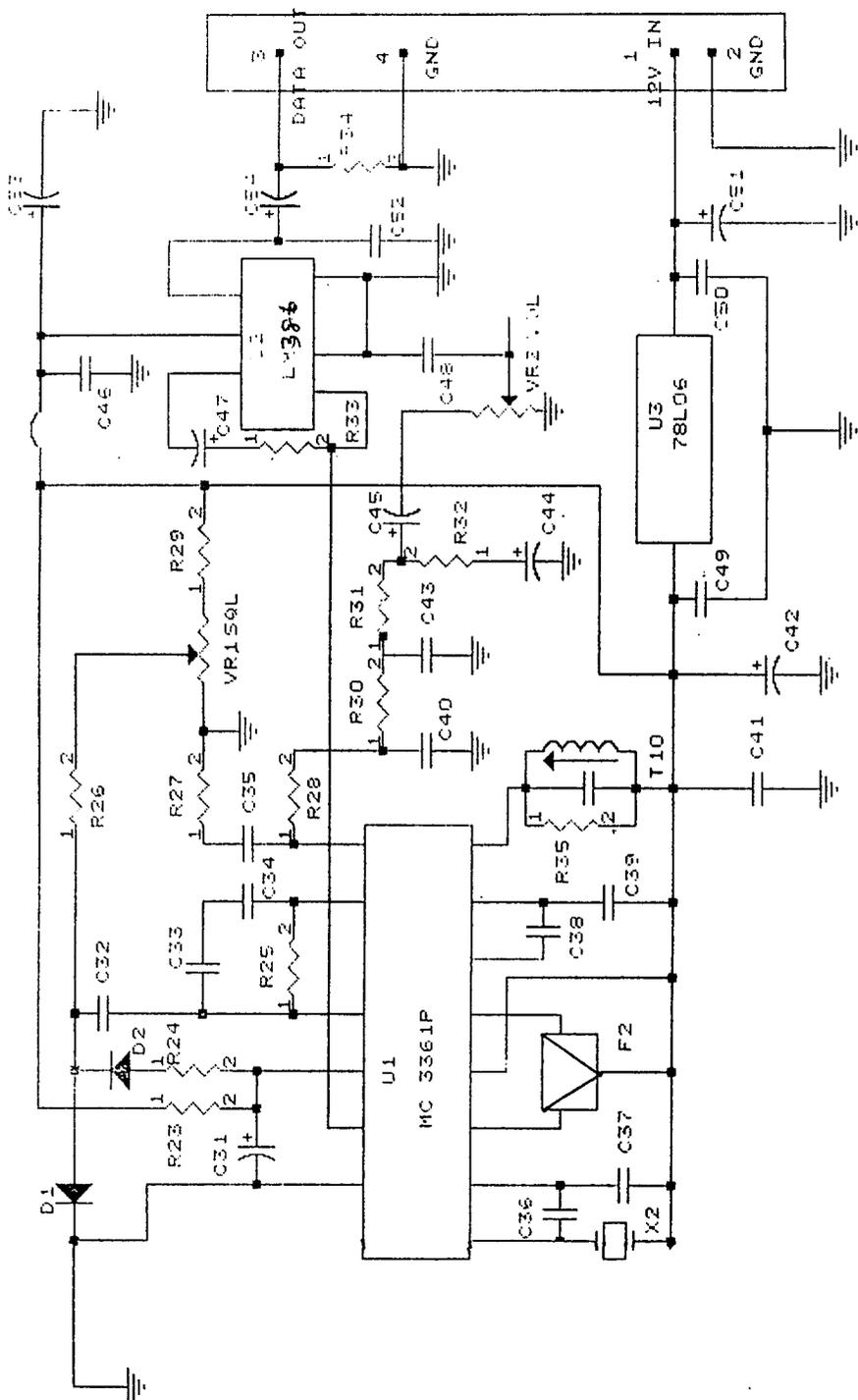
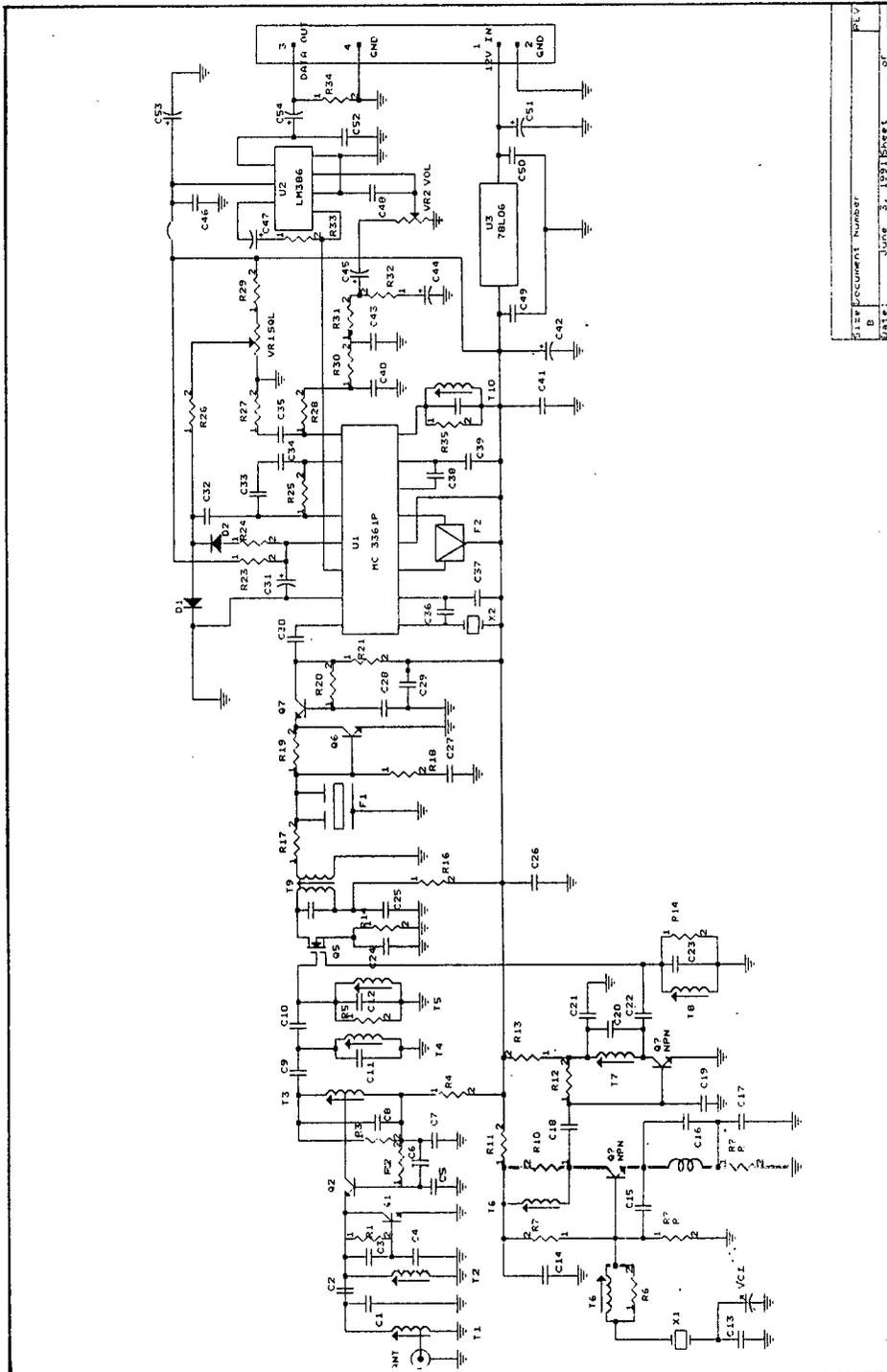


Fig. 2-17. Discriminator and audio circuit



REV: _____
 SHEET DOCUMENT NUMBER: _____
 B _____
 DATE: June 3, 1991 Sheet 37

Fig. 2-18. Receiver circuit

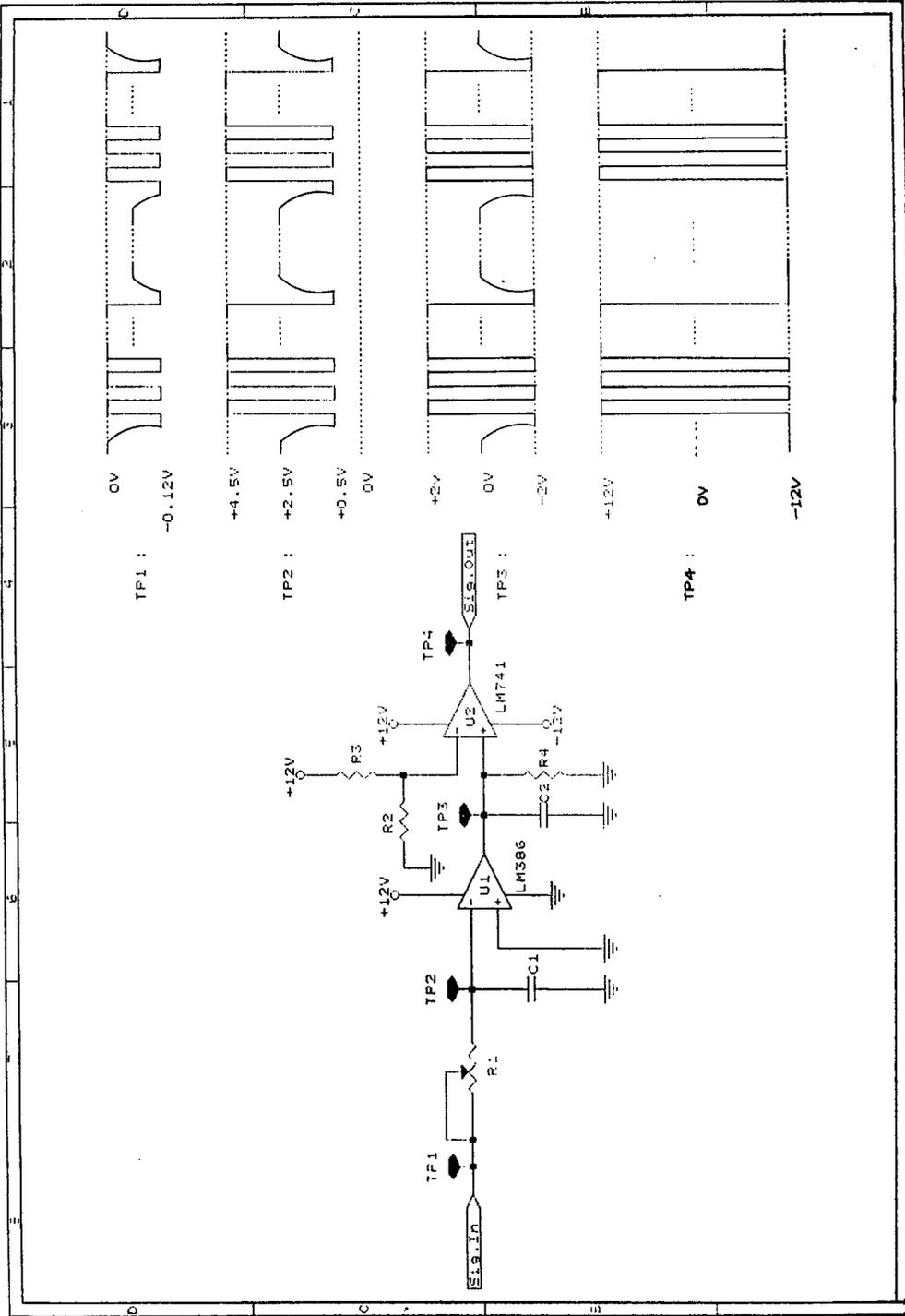


Fig. 2-19 Compensation circuit for the waveforms

때 마다 기준 Level이 Shift되어 Data로서 불안한 상태로 되기 때문에 최종 信號를 안정화시켜 줄 필요가 있다.

Fig. 2-19는 최종 信號 安定化를 시키기 위한 것으로 IC741을 썼으며 pin 6 번에서의 出力은 $\pm 12[V]$ 정도이며 TP4에서와 같이 정확한 Pulse가 되어 PC로 입력된다.

4. Data Acquisition Program

Data acquisition program인 WQMDAP.PAS는 Turbo Pascal(Borland Version 5.5)을 사용하여 작성하였다.

프로그램은

- 1) Data acquisition board에서 출력된 자료를 읽어들이는 프로그램
- 2) 入力된 資料를 處理하여 화면에 나타내는 프로그램
- 3) Disk나 Printer에 자료를 기록하는 프로그램
- 4) 프로그램을 종료 시키거나, 화면의 내용을 프린터로 옮기거나, 앞 페이지 자료를 볼 수 있는 Key 명령어 프로그램등으로 구성되어 있다.

Data acquisition board의 출력을 Personal computer의 COM2 : Port(RS-232C Asynchronous serial port)로 읽어 들었다. COM2 : Port를 制御하기 위하여 TURBO ASYNCH Version 1.0을 Unit 파일로 변경하여 사용하였다. COM2 : Port를 Data acquisition board의 출력과 같은 형태인 2400 [bps], 8 Data bits, No parity bit, 1 Stop bit로 設定하였으며, 한 개의 資料가 16비트로 구성되어 있으므로 8비트씩 나누어 2바이트 文字列 형태로 자료를 읽었다.

資料處理는 文字列 형태의 入力資料를 整數形으로 變換하여 사용하였다. 優先 16비트(DB15, DB14, ..., DB1, DB0)의 資料를 채널 番號(상위 4비트 : DB15, DB14, DB13, DB12)와 A/D 變換값(DB11, DB10, ..., DB1, DB0)으로 分離하여, 채널

別로 資料를 정리하여 “채널 0을 온도”, “채널 1을 電氣 傳導度”, “채널 2를 溶存酸素量”, “채널 3을 수소 이온 농도”, “채널 4를 濁度”로 나타내었다. 이를 시간과 함께 Disk에 기록하고, 平均, 最少, 最大值를 구하여 그 값을 화면에 나타내었고, 平均值를 그래프로 나타내었다.

Keyboard 명령어로는 프로그램 遂行중 입력 Key에 해당하는 명령을 수행하기 위한 것으로 다음과 같이 3종류의 명령을 사용하였다.

<PageUp> : 前 화면의 그래픽 내용을 현재 화면에 겹쳐서 나타내는 기능.

<P> : 화면의 내용을 프린트로 옮기는 기능.

<ESC> : 사용중인 화일과 COM2 : Port를 Close하고 화면을 Text 모드로 전환한 후 프로그램을 종료한다.

프로그램 동작은

- 1) Personal Computer의 현재 날짜와 시간을 확인하고 틀린 경우에는 수정, 입력하고
- 2) “Turbo Asynch” Unit를 사용하기 위하여 RAM상주 프로그램인 “LOCM.EXE”를 실행 시킨다.
- 3) Data acquisition program인 “WQMDAP.EXE”를 실행시킨다.

Program WQMDAP ;

```
{ ----- }
{
{ Program name : WQMDAP }
{
{ Language : TURBO Pascal Version 5.5 }
{ ----- }
```

uses

Dos, Crt, Printer, Asynch, Graph, EPGGraph ;

const

ADRange = 4096 ; {12-bit A/D Converter 0~4095}

type

str80 = string[80] ;

str4 = string[4] ;

str2 = string[2] ;

DataFmt = record{Disk Data Format}

Hour : word ;

Min : word ;

Sec : word ;

HSec : word ;

Temp : word ; {온도}

Cond : word ; {염분도}

DsOx : word ; {용존사소량}

PeHa : word ; {pH}

Turb : word ; {탁도}

end ;

var

SN : word ;

OutFileName : str80 ;

OutFile : file of DataFmt ;

NoOfch, NoOfData : integer ;

InKey : char ;

Xmax, Ymax : integer ;
BackColor : interger ;
MaxColors : word ;
GrImage : array[1..5] of pointer ; {Marker 기록}
DataPoint, DataPoint1 : pointer ;
MinTemp, MaxTemp, MinCond, MaxCond, MinDsOx,
MaxDsOx, MinPeHa, MaxPeHa, MinTurb, MaxTurb : word ;
SumTemp, AvgTemp, SumCond, AvgCond, SumDsOx,
AvgDsOx, SumPeHa, AvgPeHa, SumTurb, AvgTurb : real ;
AvgTempStr, MinTempStr, MaxTempStr,
AvgCondStr, MinCondStr, MaxCondStr,
AvgDsOxStr, MinDsOxStr, MaxDsOxStr,
AvgPeHaStr, MinPeHaStr, MaxPeHaStr,
AvgTurbStr, MinTurbStr, MaxTurbStr,
TempStr, CondStr, DsOxstr, PeHaStr, TurbStr : string ;
TempData, CondDate, DoData, PehaData, TurbData : integer ;
StrTime : string ;
TextHeight1, TextHeight2, TextWidth1, TextWidth2 : integer ;
TitleWnhdX1, TitleWndY1, TitleWndX2, TitleWndY2,
PlotWndX1, PlotWndY1, PlotWndX2, PlotWnhdY2,
PltWndX1, PltWndY1, PltWndX2, PltWndY2,
TempWndX1, TempWndY1, TempWndX2, TempWndY2,
CondWndX1, CondWndY1, CondWndX2, CondWndY2,
DoWndX1, DoWndY1, DoWndX2, DoWndY2,
pHWndX1, pHWndY1, pHWndX2, pHWndY2,

TurbWndX1, TurbWndY1, TurbWndX2, TurbWndY2,	
XAxisX1, XAxisX2, XAxisY1, XAxisY2,	
YAxisX1, YAxisX2, YAxisY1, YAxisY2	: integer ;
X1Pt, Y1Pt, X2Pt, Y2Pt	: integer ;
YPoint, PTxtYPt	: integer ;
MarkInt, MarkPt	: integer ;
XWdth, YHght, XZero	: integer ;
XScale	: real ;
MRange	: integer ;
size1	: integer ;
UpPage	: boolean ;
status, PorStatus, QueueSize	: integer ;
AsynchQueue	: array[1..1024] of char ;
ChNo	: byte ;
RawData	: DataFmt ;
SerNo	: word ;
StrSerNo	: string ;
ComWord	: Word ;
InComStr	: str2 ;
DtBut	: array[1..3000] of DataFmt ;

```

procedure InitVar ;
{변수의 초기치 설정}
begin
    InKey          := '' ;
    NoOfData       := 1 ;
    UpPage         := false ;
    YPoint         := 0 ; MarkPt := 10 ;
    MaxTemp        := 0 ; MaxCond := 0 ; MaxDsOx := 0 ; MaxPeHa := 0 ;
    MaxTurb        := 0 ;
    SumTemp        := 0 ; SumCond := 0 ; SumDsOx := 0 ; SumPeHa := 0 ;
    SumTurb        := 0 ;
    TempData       := 0 ; CondData := 0 ; DoData := 0 ; PehaData := 0 ;
    TurbData       := 0 ;
    MinTemp        := 4095 ; MinCond := 4095 ; MinDsOx := 4095 ; MinPeHa := 4095 ;
    MinTurb        := 4095 ;
end ; {InitVar}

function Int2Hex(i : word) : str4 ;
{수치를 문자열로 변환}
var
    HexStr : string[8] ;
    b      : array[1..2] of byte absolute i ;
    bt     : byte ;

```

```

function Translate(b : byte) : char ;
begin
    if b < 10 then
        Translate := Chr(b+48)
    else
        Translate := Chr(b+55) ;
end ; { Translate}

begin
    HexStr := '' ;
    HexStr := HexStr + Translate(b[2] shr 4) ;
    HexStr := HexStr + Translate(b[2] and 15) ;
    HexStr := HexStr + Translate(b[1] and 4) ;
    HexStr := HexStr + Translate(b[1] and 15) ;
    Int2Hex := HexStr ;
end ; { Int2Hex}

procedure ClsComPort ; forward ;
procedure ComPortError (AsynchQueue, PortStatus : integer ; proc : str80) ;
{Com : Port Error Message}
var
    StatusWord           : integer ;
    LineStatus, ModemStatus : integer ;
begin
    Writeln ;
    Writeln('TURBO ASYNC' 'status' 'returned in', proc, '=', AsynchQueue) ;

```

```

case AsyncQueue of
  OK : Writeln('Everything is okay. Why are you calling me?');
  Inv_Port : Writeln('Invalid port number');
  Port_Not_Open : Writeln('The port is not currently open');
  Inv_Parameter : Writeln('Invalid parameter. Check port address and interrupt
line');
  No_Port_Found : Writeln('No serial port found at specified address');
  Out_Q_Full : Writeln('The output queue is full');
  Port_Open_Already : Writeln('The port is already open');
  In_Q_Empty : Writeln('The input queue was empty');
  Internal_Error : Writeln('Internal TURBO ASYNCH level zero error');
end;

writeln('TURBO ASYNC port status word', Int2Hex(PortStatus));
statusWord := statal(com2, LineStatus, ModemStatus);
writeln('BIOS interrupt hex 14 line/modem status word=', Int2Hex(Status-
Word));

ClsComPort;

halt(1);

end; {ComPortError}

Procedure ClsComPort;
{Close Com2 : Port}
begin
  status := closeal (com2);

```

```

if not(status in [OK, Port__Not__Open])then
  ComPortError(status, 0, 'Close-Port');
end : {ClsComPort}

procedure SetUpComPort ;
procedure OpenComPort ;
{Open Com2 : Port}
const
  InSize  = 1000 ;
  OutSize = 24 ;
  irq3    = 3 ;
  Portaddr= $ 02F8 ;
begin
  status := openal(com2, InSize, OutSize, irq3, Portaddr, addr(AsynchQueue)) ;
  if status=Port__Open__Already then
    begin
      ClsComPort ;
      status=openal(com2, InSize, OutSize, irq3, Portaddr, addr(AsynchQueue)) ;
      if status=Port__Open__Already then
        begin
          ClsComPort ;
          status := openal (com2, InSize, OutSize, irq3, Portaddr, addr(AsynchQueue)) ;
        end
      else

```

```
    if not(status=OK) then ComPortError(status, 0, 'OpenComPort');  
end ; {OpenComPort}
```

```
Procedure SetComPrmt ;
```

```
{COM2 : Port Parameter 설정 }
```

```
{2400 BPS, 8 data bits, No Parity, 1 Stop Bit }
```

```
const
```

```
    SetBaudRate      =1 ;
```

```
    SetParityBite    =2 ;
```

```
    SetDataBits      =3 ;
```

```
    SetStpBits       =4 ;
```

```
    Baud2400         =5 ;
```

```
    NoParity         =0 ;
```

```
    EghtDtBits       =3 ;
```

```
    OneStopBits      =0 ;
```

```
begin
```

```
    status := setopal (com2, SetBaudRate, Baud2400) ;
```

```
    if not(status=OK)then
```

```
        ComPortError(status, 0, 'SetComPrmt(1)') ;
```

```
    status := setopal (com2, SetParityBits, NoParity) ;
```

```
    if not(status=OK) then
```

```
        ComPortError(status, 0, 'SetComPrmt(2)') ;
```

```
    status := setopal (com2, SetBaudRate, Baud2400) ;
```

```
    if not(status=OK) then
```

```

    ComPortError(status, 0, 'SetComPrmt(3)');
    status := setopal (com2, SetStpBits, OneStopBits);
    if not(status=OK)then
        ComPortError(status, 0, 'SetComPrmt(4)');
end : {SetComPrmt}

```

```

begin
    PorStatus := 0;
    OpenComPort;
    SetComPrmt;
end : {SetUpComPort}

```

```

Procedure DspInitMsg;
{Serial Number, 샘플 주기당 Data 갯수 입력}

```

```

begin
    write('Enter the serial number:');
    readln(SN);
    write('Enter the number of data:');
    readln(NoOfCh);
end ; {DspInitMsg}

```

```

Procedure OpenOutFile;

```

```

begin
    write('Enter output File name:');
    readln(OutFileName);

```

```

assign(OutFile, OutFileName) ;
{ $I- } reset(OutFile) ; { $I+ }
if IOResult=0 then
begin
  repeat
    writeln(OutFileName, 'already exists!');
    writeln('Press "R" to replace this file or');
    write('    "S" to skip this file(R/S):');
    InKey := readkey ;
    InKey := upcase(InKey) ;
  until(InKey in ['R', 'S']);
  if(InKey='S')then
  begin
    close(OutFile) ;
    ClrScr ;
    OpenOutFile ;
  end ;
end ;
rewrite(OutFile) ;
writeln ;
write('Strike any key to start ---');
InKey := readkey ;
end ;

```

Procedure FlshInQueue ;

begin

status := iflshal(com2) ;

if not(status=OK) then

ComPortError(status, 0, 'FlshInQueue') ;

end ; {FlshInQueue}

Procedure WaitForMsg(len : integer) ;

begin

repeat

status := qsizeal(com2, QueueSize, PortStatus) ;

if not(status=OK)then

ComPortError(status, PortStatus, 'WaitForMsg') ;

if KeyPressed then

begin

InKey := ReadKey ;

if InKey= #112 then PrintGraph(6, Portrait) ;

if InKey= #0 then

begin

Inkey := ReadKey ;

if InKey= #73 then PutImage(PltWndX1, PltWndY1, DataPoint1, XOrPut) ;

end ;

end ;

until(QueueSize>=len) or (InKey= #27) ;

```

if InKey= #27 then
begin
    CloseGraph ;
    RestoreCrtMode ;
    ClsComPort ;
    close(OutFile) ;
    end ;
end ; {WaitForMsg}

procedure MinMax(var dataset    : word ;
                 var minval     : word ;
                 var maxval     : word ;
                 var sum        : real ;
                 var avg        : real) ;

var
    i :          integer ;
    min, max :   word ;

begin
    min := minval ;
    max := maxval ;
    sum := sum + dataset ;
    avg := sum/NoOfData ;

```

```

if dataset < min then minval := dataset ;
if dataset > max then maxval := dataset ;
end ; {MinMax}

```

```

Procedure RdStrComPrt ;

```

```

{COM2 : Port로부터 문자열 형태로 Data를 읽어 드림.}

```

```

{문자열의 크기 : 2-Byte}

```

```

const

```

```

    UpBound=2 ;

```

```

var

```

```

    NoRead : integer ;

```

```

begin

```

```

    status := rdstal(com2, UpBound, addr(InComstr[1]), NoRead, QueueSize, PortSta-
tus) ;

```

```

    if not(status=OK) then

```

```

        ComPortError(status, 0, 'RdStrComPrt') ;

```

```

        InComStr[0] := chr(NoRead) ;

```

```

end ; {RdStrComPet}

```

```

procedure ComStr2Word ;

```

```

{Com2 : port에서 입력된 문자열 형태의 Data를 정수로 변환}

```

```

{16-bit중 상위 4-bit : A/D 변환기이 채널 번호}

```

```

{나머지 12-bit      : A/D 변환 자료}

```

```

var

```

```

    msbl      : byte ;

```

```

    msb, lsb : word ;

```

begin

FlshInQueue ;

WaitForMsg(2) ;

RdStrComPrt ;

msb1 := ord(InComStr[1]) ;

ChNo := msb1 shr 4 ; {채널번호 계산}

msb := msb1 shl 8 ;

lsb := ord(InComStr[2]) ;

ComWord := msb + lsb ; {16-bit 정수로 계산}

end ; {ComStr2Word}

procedure ValToStr ; forward ;

Procedure ArngComWord ;

{채널번호 별로 분류하여 최대, 최소, 평균치 계산}

{채널0 : 온도, 채널1 : 전도도, 채널2 : 용존산소량}

{채널3 : pH, 채널4 : 탁도}

var

msb1 : byte ;

msb, lsb : word ;

index : integer ;

begin{ArngComWord}

ComWord := ComWord and \$0FFF ; {16-bit중에서 12-bit A/D Data만 선택}

with RawData do

```

begin
case ChNo of
  0 : begin
    GetTime(Hour, Min, Sec, HSec) ;
    Tem := ComWord ;
    MinMax(Temp, MinTemp, MaxTemp, SumTemp, AvgTemp) ;
    end ;
  1 : begin
    Cond := ComWord ;
    MinMax(Cond, MinCond, MaxCond, SumCond, AvgCond) ;
    end ;
  2 : begin
    DsOx := ComWord ;
    MinMax(DsOx, MinDsOx, MaxDsOx, SumDsOx, AvgDsOx) ;
    end ;
  3 : begin
    PeHa := ComWord ;
    MinMax(PeHa, MinPeHa, MaxPeHa, SumPeHa, AvgPeHa) ;
    end ;
  4 : begin
    Turb := ComWord ;
    MinMax(Turb, MinTurb, MaxTurb, SumTurb, AvgTurb) ;
    {Disk에 정수형태로 시간과 A/D 변환 자료 기록}
    DtBuf[NoOfData] := RawData ;
    inc(NoOfData) ;

```

```

if NoOfData=NoOfCh+1 then
begin
  for index :=1 to NoOfData-1 do
    write(OutFile, DtBuf[index]);
  colse(OutFile);
  reset(OutFile);
  seek(OutFile, Filesize(OutFile));
  {숫자 값을 문자열로 변환}
  ValToStr;

  { 최대, 최소, 합의 초기치 선정}
  MinTemp :=4095; MinCond :=4095; MinDsOx :=4095;
  MinPeHa :=4095; MinTurb :=4095;
  MaxTemp :=0; MaxCond :=0; MaxDsOx :=0;
  MaxPeHa :=0; MaxTurb :=0;
  SumTemp :=0; SumCond :=0; SumDsOx :=0;
  SumPeHa :=0; SumTurb :=0;

  end;
end;
end;
end;
end; {ArngComWord}

```

```

procedure HeadDtChk ;
{Header Data Check : Header Data 1 : serial number}
{
                Header Data 2 : 65535      }
{
                Header Data 3 :    0      }
{
                Header Data 4 : 21845      }
var
    i : integer ;
begin
    NoOfData := 1 ;
    SerNo := ComWord ;
    str(SerNo : 5, StrSerNo) ;
    for i := 1 to 3 do
    begin
        ComStr2Word ;
        case i of
            1 : begin
                if ComWord <> 65535 then
                begin
                    sound(2000) ;
                    delay(100) ;
                    nosound ;
                end ;
            end ;
        end ;
    end ;
end ;

```

```

2 : begin
    if ComWord <> 0 then
        begin
            sound(2000) ;
            delay(100) ;
            nosound ;
        end ;
    end ;

3 : begin
    if Comword <> 21845 then
        begin
            sound(2000) ;
            delay(100) ;
            nosound ;
        end ;
    end ;
end ;

end ; {HeadDtChk}

procedure TestGraphError(Grapherr ; integer) ;
begin
    if GraphErr <> grOk then begin
        Writeln('Graphics error : ', GrapherrorMsg(GraphErr)) ;
        repeat until keypressed ;
    end ;
end ;

```

```

    InKey := readkey ;
    Halt(1) ;
end ;
end ; {TestGraphError}

```

```

procedure InitGrph ;
var
    GraphDriver, GraphMode ; integer ;
begin
    GraphDriver := Detect ;
    InitGraph(GraphDriver, GraphMode, '' ) ;
    TestGraphError(GraphResult) ;
    MaxColors := GetmaxColor ;
    Xmax := GetMaxX ;
    Ymax := GetMaxY ;
end ; {InitGrph}

```

Procedure MarkSymbls ;

```

{Graphic Symbol을 그려서 GrImage[I] pointer 변수에 기록}
{Symbol 1 : "+", GrImage[1], 온도 }
{Symbol 2 : "□", GrImage[2], 전도도 }
{Symbol 3 : "◆", GrImage[3], 용존산소량 }
{Symbol 4 : "X", GrImage[4], pH }
{Symbol 5 : "▲", GrImage[5], 탁도 }

```

```

const
    figure1 : array[1..8] of integer=(3, 0, 6, 6, 0, 6, 3, 0) ;
var
    i : integer ;
begin
    SetFillStyle(EmptyFill, 0) ;
    if MaxColors > 4 then Setcolor(10) ;
    Line(3, 0, 3, 6) ;
    Line(0, 3, 6, 3) ;
    GetMem(GrImage[1], ImageSize(0, 0, 6, 6)) ;
    GetImage(0, 0, 6, 6, GrImage[1]) ;
    PutImage(0, 0, GrImage[1], XOrPut) ;

    if MaxColors > 4 then SetColor(11) ;
    Rectangle(0, 0, 6, 6) ;
    GetMem(GrImage[2], ImageSize(0, 0, 6, 6)) ;
    GetImage(0, 0, 6, 6, GrImage[2]) ;
    PutImage(0, 0, GrImage[2], XOrPut) ;

    if MaxColors > 4 then SetColor(12) ;
    Circle(3, 3, 3) ;
    SetFillStyle(SolidFill, GetColor) ;
    FloodFill(3, 3, GetColor) ;
    GetMem(GrImage[3], ImageSize(0, 0, 6, 6)) ;
    GetImage(0, 0, 6, 6, GrImage[3]) ;
    PutImage(0, 0, GrImage[3], XOrPut) ;

```

```

if MaxColors > 4 then SetColor(13) ;
Line(0, 0, 6, 6) ;
Line(6, 0, 0, 6) ;
GetMem(GRImage[4], ImageSize(0, 0, 6, 6)) ;
GetImage(0, 0, 6, 6, GRImage[4]) ;
PutImage(0, 0, GRImage[4], XOrPut) ;

if MaxColors > then SetColor(14) ;
FillPoly(4, figure1) ;
GetMem(GRImage[5], ImageSize(0, 0, 6, 6)) ;
GetImage(0, 0, 6, 6, GRImage[5]) ;
PutImage(0, 0, GRImage[4], XOrPut) ;
end ; {MarkSymbols}

procedure FullPort ;
begin
  SetViewPort(0, 0, Xmax, Ymax, ClipOn) ;
end ; {FullPort}

procedure Frame ;
{화면을 4부분으로 나누어 Title Window, Command Window}
{Data Plotting Window, Data Display Window로 지정      }
var
  CmdWndX1, CmdWndY1, CmdWndX2, CmdWndY2 : integer ;

```

DtWndX1, DtWndY1, DtWndX2, DtWndY2 : integer ;

MsgStPt : integer ;

TickX1, TickInc, i, WndInc, WndDiv : integer ;

procedure TitleWnd ;

{Title Window : 제목과 우측에 마지막 관측 시간 나타냄}

begin

TitleWndX1 := 1 ;

TitleWndY1 := 1 ;

TitleWndX2 := TextWidth1 * 69 ;

TitleWndY2 := TextHeight2 + 5 ;

SetBkColor(Blue) ;

SetColor(MaxColors) ;

SetLineStyle(SolidLn, 0, NormWidth) ;

Rectangle(0, 0, Xmax, TitleWndY2) ;

SetViewPort(TitleWndX1, TitleWndY1, TitleWndX2, TitleWndY2-1, ClipOn) ;

SetColor(white) ;

SetTextJustify(LeftText, TopText) ;

SeTextWtyle(DefaultFont, HorizDir, 1) ;

OutTextXy(5, TextHeight1, 'KORDI') ;

SetColor(yellow) ;

SetTextStyle(TriplexFont, HorizDir, 2) ;

OutTextXY(Textwidth2*5, TitleWndY1-2, 'WATER QUALITY MONITORING SY-
STEM') ;

SetTextStyle(DefaultFont, HorizDir, 1) ;

```

SetColor(white) ;
FullPort ;
end ; {TitleWnd}

procedure CmdWnd ;
{Command Window : <Esc> --- 프로그램을 끝낼때 }
{
    <P> ----- 화면의 내용을 EPSON Printer로 Screen Dump }
{
    <PageUp> - 前화면을 볼때 }
begin
    CmdWndX1 := 0 ;
    CmdWndY1 := Ymax - TextHeight1 - 5 ;
    CmdWndX2 := Xmax ;
    CmdWndY2 := Ymax ;
    Rectangle(CmdWndX1, CmdWndY1, CmdWndX2, CmdWndY2) ;
    SetViewPort(CmdWndX1+1, CmdWndY1+1, CmdWndX2-1, CmdWndY2-1, ClipOn) ;
    OutTextXY(1, 1, '<Esc>-quit <P>-PrintScreen <PgUp>-UpPage') ;
    FullPort ;
end ; {CmdWnd}

procedure PlotWnd ;
Procedure DrawAxis ;
procedure XAxis ;
var

```

```

    i : integer ;
procedure DrawTick(Para, Tick1, Tick2, Tick3, Tick4, Tick5 : string) ;
var
    i : integer ;
    TickLabelY : integer ;
begin
    TickX1 := XAxisX1 ;
    TickInc := (XAxisX2 - XAxisX1) div 4 ;
    TickLabelY := XAxisY1 - 5 - TextHeight1 ;
    for i := 0 to 4 do
        begin
            bar(TickX1, XAxisY1 - 5, TickX1, XAxisY1) ; {Tick}
            case i of
                0 : OutTextXY(TickX1, TickLabelY, Tick1) ;
                1 : OutTextXY(TickX1, TickWidth2, TickLabelY, Tick2) ;
                2 : OutTextXY(TickX1, TickWidth2, TickLabelY, Tick3) ;
                3 : OutTextXY(TickX1, TickWidth2, TickLabelY, Tick4) ;
                4 : OutTextXY(TickX1, TickWidth2, TickLabelY, Tick5) ;
            end ;
            TickX1 := TickX1 + TickInc ;
        end ;
        OutTextXY(5, TickLabelY, Para) ;
    end ; {DrawTick}

```

```

begin{XAxis}
  XAxisX1 := TextWidth2 shl 2 + TextWidth1 ;
  Inc(XAxisX1 PlotWndX1) ;
  XAxisY1 := TextHeight2 ;
  Inc(XAxisY1, PlotWndY1) ;
  XAxisX2 := PlotWndX2 - TextWidth2 ;
  XAxisY2 := XAxisY1 ;
  for i := 0 to 4 do
  begin
    bar(XAxisX1, XAxisY1, XAxisX2, XAxisY2) ;
    case i of
      0 : begin
          DrawTick('TEMP : ', '0.00', '12.5', '25.0', '37.5', '50.0') ;
          PutImage(55, XAxisY1 - TextHeight1 - 3, GrImage[1], XOrPut) ;
          Inc(XAxisY1, TextHeight2) ;
          XAxisY2 := XAxisY1 ;
        end ;
      1 : begin
          DrawTick('COND : ', '0.00', '12.5', '25.0', '37.5', '50.0') ;
          PutImage(55, XAxisY1 - TextHeight1 - 3, GrImage[2], XOrPut) ;
          Inc(XAxisY1, TextHeight2) ;
          XAxisY2 := XAxisY1 ;
        end ;
    end ;
  end ;
end ;

```

```

2 : begin
    DrawTick('DsOx : ', '0.00', '25.0', '50.0', '75.0', '100');
    PutImage(55, XAxisY1-TextHeight1-3, GrImage[3], XOrPut);
    Inc(XAxisY1, TextHeight2);
    XAxisY2 := XAxisY1;
end ;

3 : begin
    DrawTick('PeHa : ', '2.00', '4.50', '7.00', '9.50', '12.0');
    PutImage(55, XAxisY1-TextHeight1-3, GrImage[4], XOrPut);
    Inc(XAxisY1, TextHeight2);
    XAxisY2 := XAxisY1;
end ;

4 : begin
    DrawTick('Turb : ', '0.00', '25.0', '50.0', '75.0', '100');
    PutImage(55, XAxisY1-TextHeight1-3, GrImage[5], XOrPut);
end ;

end ;

end ; {XAxis}

procedure YAxis ;
begin
    YAxisX1 := TextWidth2 sh1 2+TextWidth1 ;
    Inc(YAxisX1, PlotWndX1) ;
    YAxisY1 := TextHeight2 ;

```

```

Inc(YAxisY1, PlotWndY1) ;
YAxisX2 := YAxisX1 ;
YAxisY2 := PlotWndY2 - TextHeight2 ;
bar(YAxisX1, YAxisY1, YAxisX2, YAxisY2) ;
end ; {YAxis}

```

```

begin{DrawAxis}
  XAxis ;
  YAxis ;
  PltWndX1 := XAxisX1 + 1 ;
  PltWndY1 := XAxisY1 + 1 ;
  PltWndX2 := XAxisX2 ;
  PltWndY2 := XAxisY2 ;
end ; {DrawAxis}

```

```

begin{PlotWnd}
  PlotWndX1 := 0 ;
  PlotWndY1 := TitleWndY2 + 2 ;
  PlotWndX2 := Xmax - (Xmax div 3) ;
  PlotWndY2 := CmdWndY1 - 2 ;
  Rectangle(PlotWndX1, PlotWndY1, PlotWndX2, PlotWndY2) ;
  DrawAxis ;
  FullPort ;
end ; {PlotWnd}

```

```

procedure DtWnd ;
var
    i : integer ;
begin
    DtWndX1 := PlotWndX2 + 2 ;
    DtWndY1 := PlotWndY1 ;
    DtWndX2 := Xmax ;
    DtWndY2 := PlotWndY2 ;
    Rectangle(DtWndX1, DtWndY1, DtWndX2, DtWndY2) ;
    SetViewPort(DtWndX1+1, DtWndY1+1, DtWndX2-1, DtWndY2-1, ClipOn) ;
    SetColor(yellow) ;
    OutTextXY(TextWidth2+2, 5, 'Average. Minimum. Maximum') ;
    bar(1, TextHeight2, DtWndX2, TextHeight2+1) ;
    bar(TextWidth2, 1, TextWidth2+1, CmdWndY1-DtWndY1-TextHeight1) ;

    FullPort ;
    SetViewPort(DtWndX1, DtWndY1+TextHeight2+2, DtWndX2, DtWndY2, ClipOn) ;
    SetTextStyle(DefaultFont, VertDir, 1) ;
    SetTextJustify(RightText, CenterText) ;
    WndInc := (DtWndY2-DtWndY1-TextHeight2-2) Div 5 ;
    WndDiv := 0 ;
    for i := 0 to 4 do
        begin
            Inc(WndDiv, WndInc) ;

```

PTxtYPt := WndDiv - WndInc Div 2 ;

case i of

0 : begin

OutTextXY(TextHeight2, PTxtYpt, 'Temp') ;

TempWndX1 := DtWndX1 + TextWidth2 + 3 ;

TempWndY1 := DtWndY1 + TextHeight 2 + 3 ;

TempWndX2 := DtWndX2 - 2 ;

TempWndY2 := TempWndY1 + WndDiv - 2 ;

end ;

1 : begin

OutTextXY(TextHeight2, PTxtYpt, 'Cond') ;

CondWndX1 := TempWndX1 ;

CondWndY1 := TempWndY2 + 2 ;

CondWndX2 := TempWndX2 ;

CondWndY2 := TempWndY1 + WndDiv - 2 ;

end ;

2 : begin

OutTextXY(TextHeight2, PTxtYpt, 'DisOx') ;

DoWndX1 := TempWndX1 ;

DoWndY1 := CondWndY2 + 2 ;

DoWndX2 := TempWndX2 ;

DoWndY2 := TempWndY1 + WndDiv - 2 ;

end ;

```

3 : begin
    OutTextXY(TextHeight2, PtxtYpt, 'PeHa');
    pHWndX1 := TempWndX1;
    pHWndY1 := DoWndY2 + 2;
    pHWndX2 := TempWndX2;
    pHWndY2 := TempWndY1 + WndDiv - 2;
end;

4 : begin
    OutTextXY(TextHeight2, PtxtYpt, 'Turb');
    TurbWndX1 := TempWndX1;
    TurbWndY1 := DoWndY2 + 2;
    TurbWndX2 := TempWndX2;
    TurbWndY2 := TempWndY1 + WndDiv - 2;
end;

end;

bar(1, WndDiv, Xmax - DtWndX1, WndDiv);

end;

FullPort;

end; {DtWnd}

begin{Frame}
    TextHeight1 := TextHeight('M');
    TextHeight1 := TextHeight1 shl 1;
    TextWidth1 := TextWidth('M');

```

```
TextWidth2 :=TextWidth1 shl 1 ;
FullPort ;
TitleWnd ;
CmdWnd ;
PlotWnd ;
DtWnd ;
SetColor(white) ;
SetTextStyle(DefaultFont, HorizDir, 1) ;
    SetTextJustify(LeftText, TopText) ;
end ; {Frame}
```

```
procedure GetYData ;
begin
    Y1Pt : Ypoint ;
    Inc(Ypoint) ;
    Y2pt : =Ypoint ;
end ; {GetYdata}
```

```
procedure PltData(Par : byte) ;
begin
    SetViewPort(PltWndX1, PltWndY1, PltWndX2, PltWndY2, ClipOn) ;
    MoveTo(X1Pt, Y1Pt) ;
    LineTo(X2Pt, Y2Pt) ;
    if Y2Pt=MarkPt then
```

```

begin
  case Par of
    1 : PutImage(X2Pt-3, Y2Pt-3, GrImage[1], XOrPut) ;
    2 : PutImage(X2Pt-3, Y2Pt-3, GrImage[2], XOrPut) ;
    3 : PutImage(X2Pt-3, Y2Pt-3, GrImage[3], XOrPut) ;
    4 : PutImage(X2Pt-3, Y2Pt-3, GrImage[4], XOrPut) ;
    5 : PutImage(X2Pt-3, Y2Pt-3, GrImage[5], XOrPut) ;
  end ;
  SetViewPort(PlotWndX1+1, PltWndY1+MarkPt,
              XaxisX1-1, PltWndY1+MarkPt+TextHeight1+3, ClipOn) ;
  ClearViewPort ;
  OutTextXY(1, 1, StrTime) ;
end ;
SetColor(White) ;
FullPort ;
end ; {PltData}

procedure TextOutGpMode ;
var
  ErrCode : integer ;
  size    : integer ;
procedure Temp(TempStr : string) ;
begin
  SetViewPort(TempWndX1, TempWndY1, TempWndX2, TempWndY2, ClipOn) ;
  GetMem(DataPoint, size) ;

```

```

GetImage(0, 10, 190, 82, DataPoint^);
ClearViewPort ;
PutImage(0, 0, DataPoint^, XOrPut) ;
OutTextXY(0, 72, TempStr) ;
FreeMem(DataPoint, size) ;
X1Pt :=TempData ;
TempData :=trunc(AvgTemp*XScale) ;
X2Pt :=TempData ;
PltData(1) ;
FullPort ;
end ; {Temp}

```

```

procedure Cond(CondStr : string) ;
begin
setViewPort(CondWndX1, CondWndY1, CondWndX2, CondWndY2, ClipOn) ;
GetMen(DataPoint, size) ;
GetImage(0, 10, 190, 82, DataPoint^);
ClearViewPort ;
PutImage(0, 0, DataPoint^, XOrPut) ;
OutTextXY(0, 72, CondStr) ;
FreeMem(DataPoint, size) ;
X1Pt :=CondData ;
CondData :=trunc(AvgCond * XScale) ;
X2Pt :=CondData ;

```

```
PltData(2) ;  
FullPort ;  
end ; {Cond}
```

```
procedure DisOx(DsOxStr : string) ;  
begin  
  SetViewPort(DoWndX1, DoWndY1, DoWndX2, ClipOn) ;  
  GetMem(DataPoint, size) ;  
  GetImage(0, 10, 190, 82, DataPoint) ;  
  ClearViewPort ;  
  PutImage(0, 0, DataPoint, XOrPut) ;  
  OutTextXY(0, 72, DsOxStr) ;  
  FreeMem(DataPoint, size) ;  
  X1Pt := DoData ;  
  DoData := trunc(AvgDsOx * XScale) ;  
  X2Pt := DoData ;  
  PltData(3) ;  
  FullPort ;  
end ; {DisOx}
```

```
procedure PeHa(PeHaStr : string) ;  
begin  
  SetViewPort(pHWndX1, pHWndY1, pHWndX2, pHWndY2, ClipOn) ;  
  GetMem(DataPoint, size) ;
```

```

GetImage(0, 10, 190, 82, DataPoint^);
ClearViewPort ;
PutImage(0, 0, DataPoint^, XOrPut) ;
OutTextXY(0, 72, PeHaStr) ;
FreeMem(DataPoint, size) ;
X1Pt := PeHaData ;
PeHaData := trunc(AvgPeHa * XScale) ;
X2Pt := PeHaData ;
PltData(4) ;
FullPort ;
end ; {PeHa}

```

```

procedure Turb(TurbStr : string) ;
begin
  SetViewPort(TurbWndX1, TurbWndY1, TurbWndX2, TurbWndY2, ClipOn) ;
  GetMem(DataPoint, size) ;
  GetImage(0, 10, 190, 82, DataPoint^);
  ClearViewPort ;
  PurImage(0, 0, DataPoint^, XOrPut) ;
  OutTextXY(0, 72, TurbStr) ;
  FreeMem(DataPoint, size) ;
  X1Pt := TurbData ;
  TurbData := trunc(AvgTurb * XScale) ;
  X2Pt := TurbData ;

```

```
PltData(5) ;  
FullPort ;  
end ; {Turb}
```

```
begin{TextOutGpMode}
```

```
size :=ImageSize(0, 10, 190, 82) ;
```

```
Temp(TempStr) ;
```

```
Cond(CondStr) ;
```

```
DisOx(DsOxStr) ;
```

```
PeHa(PeHaStr) ;
```

```
Turb(TurbStr) ;
```

```
if Y2Pt = MarkPt then
```

```
    MarkPt :=MarkPt + MarkInt ;
```

```
if MarkPt >=YHght then
```

```
    MarkPt :=10 ;
```

```
if Y2Pt >=YHght then
```

```
begin
```

```
    Ypoint :=0 ;
```

```
    SetViewPort(PltWndX1, PltWndY1, PltWndX2, PltWndY2, ClipOn) ;
```

```
    size1 :=ImageSize(0, 0, XWdth, YHght) ;
```

```
    if UpPage = true then FreeMem(DataPoint1, size1) ;
```

```
    GetMem(DataPoint1, size1) ;
```

```
    GetImage(0, 0, XWdth, YHght, DataPoint1^) ;
```

```
    ClearViewPort ;
```

```

    UpPage := true ;
    FullPort ;
end ;
end ; {TextOutGpMode}

```

```

procedure DspTime ;
function LeadingZero(w : word) : String ;
var
    s : string ;
begin
    Str(w : 0, S) ;
    if Length(s) = 1 then
        s := '0' + s ;
    LeadingZero := s ;
end ; {LeadingZero}

```

```

begin{DspTime}
    with RawData do
        StrTime := LeadingZero(Hour) + ':' + LeadingZero(Min)
                + ':' + LeadingZero(Sec) ;
        SetViewPort(TitleWndX2, TitleWndY1, Xmax-1, TitleWndY2-1, ClipOn) ;
        ClearViewPort ;
        OutTextXY(5, TextHeight1, StrTime) ;
        FullPort ;
    end ; {DspTime}

```

```

procedure ValToStr ;
begin
    str(AvgTemp : 7 : 1, AvgTempStr) ; str(MinTemp : 7, MinTempStr) ;
    str(MaxTemp : 7, MaxTempStr) ;
    str(AvgCond : 7 : 1, AvgCondStr) ; Str(MinCond : 7, MinCondStr) ;
    · str(MaxCond : 7, MaxCondStr) ;
    str(AvgDsOx : 7 : 1, AvgDsOxStr) ; str(MinDsOx : 7, MinDsOxStr) ;
    str(MaxDsOx : 7, MaxDsOxStr) ;
    str(AvgPeHa : 7 : 1, AvgPeHaStr) ; str(MinPeHa : 7, MinPeHaStr) ;
    str(MaxPeHa : 7, MaxPeHaStr) ;
    str(AvgTurb : 7 : 1, AvgTurbStr) ; str(minTurb : 7, MinTurbStr) ;
    str(MaxTurb : 7, MaxTurbStr) ;
    GetYData ;
    TempStr := concat(AvgTempStr, MinTempStr, MaxTempStr) ;
    CondStr := concat(AvgCondStr, MinCondStr, MaxCondStr) ;
    DsOxStr := concat(AvgDsOxStr, MinDsOxStr, MaxDsOxStr) ;
    PeHaStr := concat(AvgPeHaStr, MinPeHaStr, MaxPeHaStr) ;
    TurbStr := concat(AvgTurbStr, MinTurbStr, MaxTurbStr) ;
    TextOutGpmode ;
    DspTime ;
end ; {ValToStr}

```

```

procedure RdDtPlt ;
begin

```

```

repeat
  ComStr2Word ;
  if ComWord=SN then
    HeadDtChk ;
  ArngComWord ;
  If KeyPressed then
  begin
    InKey :=ReadKey ;
    if InKey= #112 then PrintGraph(6, Portrait) ;
    if InKey= #0 then
      begin
        InKey :=ReadKey ;
        if InKey= #73 then PutImage(PltWndX1, PltWndY1, DataPoint1^, XOrPut) ;
      end ;
    end ;
  until InKey = #27 ;
  CloseGraph ;
  RestoreCrtMode ;
end ; {RdDtPlt}

```

```

procedure InitGrphMode ;
begin
  InitGrph ;
  MarkSymbols ;

```

```

ClearViewPort ;
Frame ;
XWdth  :=PltWndX2-PltWndX1 ;
XScale :=XWdth/ADRange ;
YHght  :=PltWndY2-PltWndY1 ;
MarkInt :=YHght div 10 ;
end ; {InitGrphmode}

begin{Main}
    InitVar ; {변수 초기치 설정}
    SetUpComPort ; {RS232C Com2 : Port 설정}
    DspInitMsg ; {Keyboard 입력 자료}
    OpenOutFile ; {Disk file}
    ClrScr ;
    InitGrphMode ; {Graph 모드 설정}
    RdDtPlt ; {Com2 : 에서 자료 읽기 및 처리}
    ClsComPort ; {Close Com : port}
    close(OutFile) ; {Close disk file}
end. {Main}

```

제 3 장 結論 및 討議

1. 結論 및 資料 分析

본 연구를 진행함에 가장 고충적인 問題는 센서의 應用이 간단하게 되는 것이 아니라는 문제이다.

즉, 기존의 센서를 이용하여 하나의 완전한 시스템을 構築하고자 하나 각 센서의 特性이 올바르게 規格화 되었거나 상세한 資料가 부족하여 研究에 必要한 시간이 豫想보다 오래 걸리게 되었다.

센서로부터 檢出된 資料는 A/D Converter에서 2400[bps]의 速度로 약 5 [Vp-p]의 크기로 만들었다.

A/D Converter의 速度는 360, 1200, 2400, 3600, 9800[bps]등 선택적으로 만들어 줄 수 있으나 送信器의 變調 形式 및 수신에서의 복조 形式에 적합한 速度로 決定하여야 한다.

Appendix II에서와 같이 실제 送受信機를 통하여 PC에 受信한 資料를 보면 매 2초에 한번으로 5 channel의 情報를 記錄할 수 있다.

현장의 송신부와 수신 단말기까지의 距離는 送信器의 出力에 관계가 있으며 본 研究에서는 0.8[W]의 送信 出力으로 반경 약 5[Km] 내에서는 무난하게 資料를 傳送할 수 있다.

사진 1-7, 1-8에서 보는 바와 같이 送信 및 受信 안테나는 周波數 146.0[MHz]대에서 부터 146.8[MHz]까지 運用이 가능하도록 研究室에서 여러가지 實驗을 거쳐 제작하였다. 사진은 센서부분, A/D Converter, 완성된 송신기와 센서 수신기 및 수신 資料처리를 위한 PC와 그 주변 장치이다.

送信기의 出力은 공급 電力에 關係가 있기 때문에 AC 전원의 供給이 不可能한 장소에서는 Battery의 容量과 使用시간 등을 考慮하여 出力에 適合한 량을 결정한다.

受信機의 感도는 약 $0.6[\mu V]$ 로 매우 양호한 편이며 지향성이 예민한 形態의 受信用 안테나를 設計하여 사용하였다.

資料 傳送의 시간 간격은 사용자가 必要한 만큼 그 間隔을 調節할 수가 있으며 역시 送信器의 전원과도 고려할 필요가 있다.

무엇보다 중요한 사항은 현장에서 센서를 한곳에 장시간 使用할 경우 센서 표면에 이물질이 생기게 되어 測定 資料에 오차를 가져올 것이며 이로 인한 영향을 최소화한 方法과 特殊한 센서의 開發이 있어야만 하는 또다른 研究 課題가 있다.

다음은 送信機와 受信機의 製作 課程과 實驗 結果이다.

－ 送信機

- 주파수 : 146.2000[MHz]
- 송신 출력 : 0.8[W] (최대)
- 출력 임피던스 : 50[ohm]
- 주파수 안정도 : $\pm 0.001\%$ ($-10^{\circ}C$ 에서 $+60^{\circ}C$)
- 변조 방식 : $16F3 \pm 5$ [KHz] (100% 변조시)
- 발진 방식 : 수정발진 9체배
- 기생발진 및 역압도 : 반송 주파수에서 50[dB]이하
- 가청 출력 왜곡 : ± 3 [dB] (2100[Hz]에서 2700 [Hz])
- 전원 : DC 12[V] 마이너스 접지

－ 受信機

- 주파수 : 146.2000[MHz]
- 수신시스템 : Double 슈퍼 헤테로다인방식

제1국부발전 : 반송 주파수-10.7[MHz]
중간 주파수 : 1st 10.7[MHz], 2nd 455[KHz]
주파수 안정도 : $\pm 0.001\%$ (-10°C 에서 $+60^{\circ}\text{C}$)
수신 감도 : $0.6[\mu\text{V}]$ (20 [dB] 억압)
가청 출력 : 1[W] (최대) 4 ohm Load
전원 : DC 12[V] 마이너스 접지

참 고 문 헌(Reference)

- Kris Jamsa & Steven Mameroff 1988 Turbo Pascal Programmer's Library Borland—Osborne/MoGRAW—Hill
- Ben Ezzell 1990 Graphics Programming in Turbo Pascal 5.5 An Object—Oriented Approach Addison—Wesley Publishing Company, Inc.
- Stephen K. O'Brien 1989 Turbo Pascal 5.5 : the complete reference Borland—Osborne/McGRAW—Hill
- Fred Halsall 1989 Data Communications, Computer Networks AND OSI Second Edition Addison—Wesley publishing Company, Inc.
- 이욱양 편 1990 C 프로그래머를 위한 시리얼 커뮤니케이션 가남사
- 편집부 역편 1988 센서의 원리와 사용법(I) [광, 전자기, 음, 길이] 도서출판 세화
- 金榮海 譯 1986 센서인터페이싱 No.1 溫度·濕度센서 活用篇
- 동역 메카트로닉스 연구소 편 1990 디지털/아나로그 전자제어 회로집 가남사
- 동역 메카트로닉스 연구소 편 1990 계측제어 시스템 영진출판사
水質 チエツカ WQC—2A 取扱説明書 東亞
電波工業株式會社

1982 Linear Databook II National Semiconductor
Corporation

1980 COS/MOS Integrated Circuits RCA solid
state

이충용

1987 통신방식의 기초, 집문당

강창언

1989 디지털 통신, 청문각

손현, 김난숙

1990 테이타통신, 대림 출판사

禹廣芳, 黃金燦 共譯

1991 信號와 시스템 淸文閣

APPENDIX I (Photography)

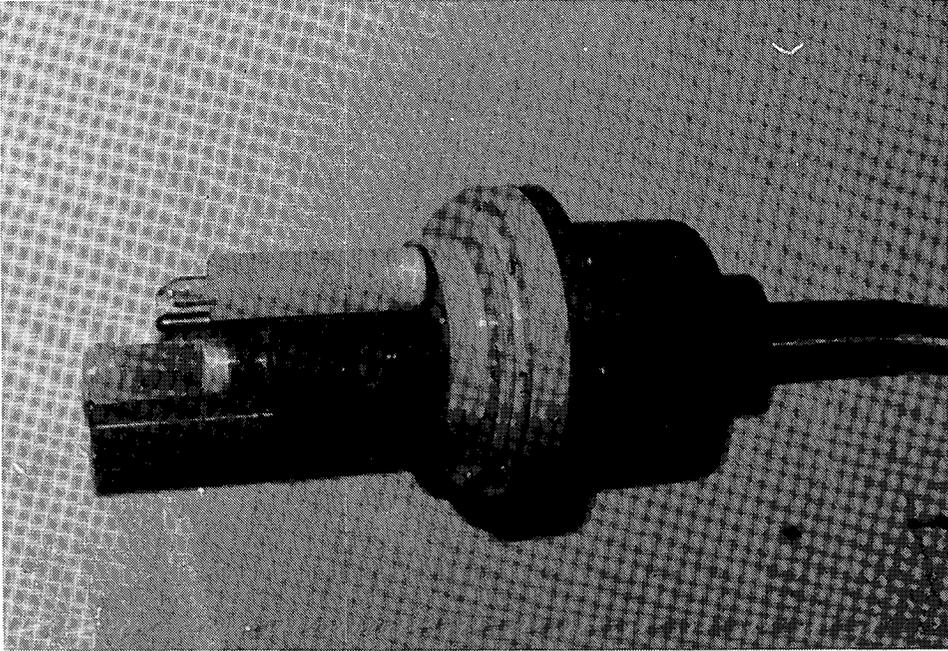


Photo 1-1 Inside of sensors

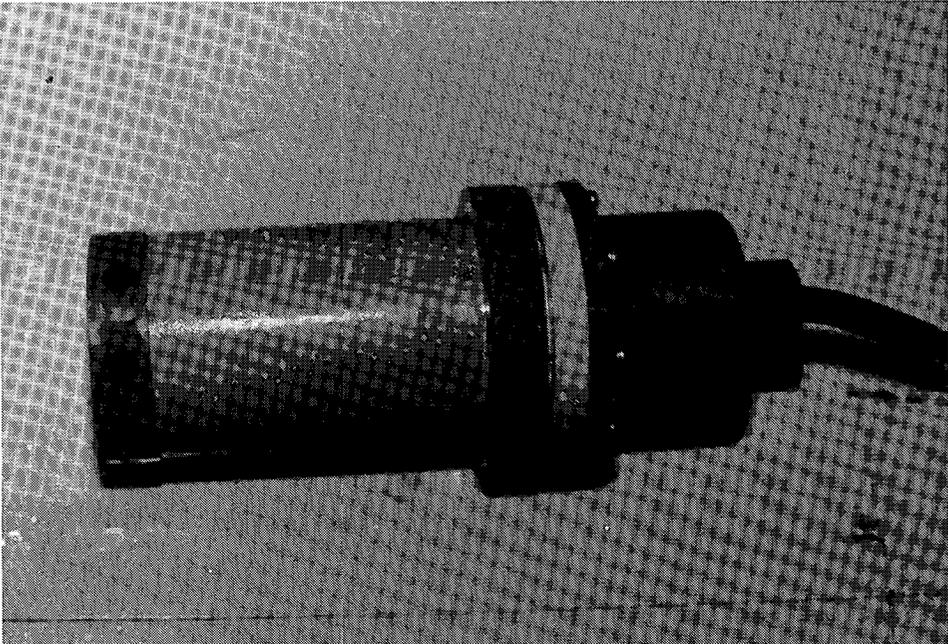


Photo 1-2 Outside of sensors

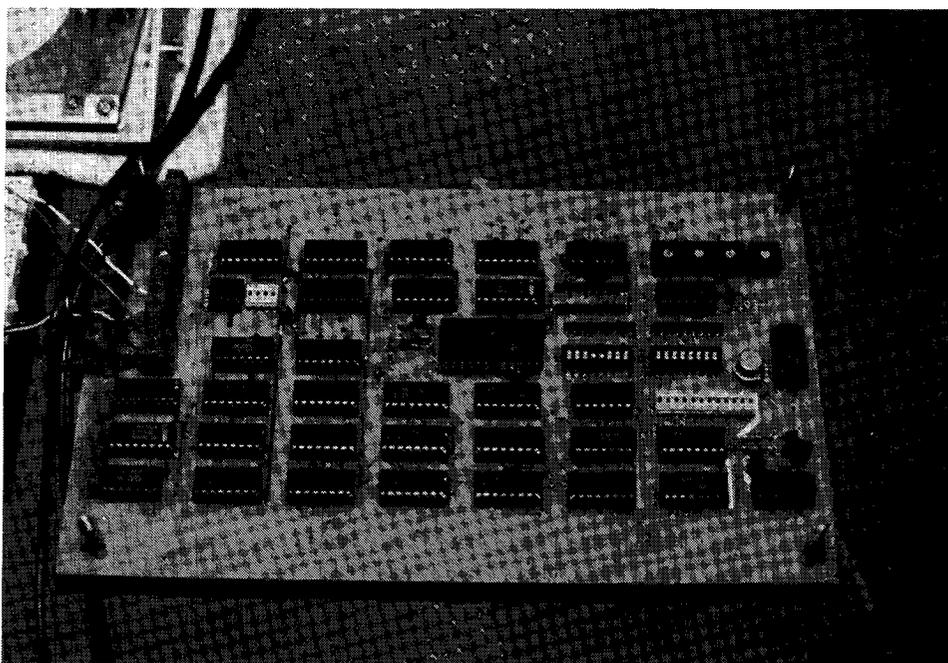


Photo 1-3 A/D Converter

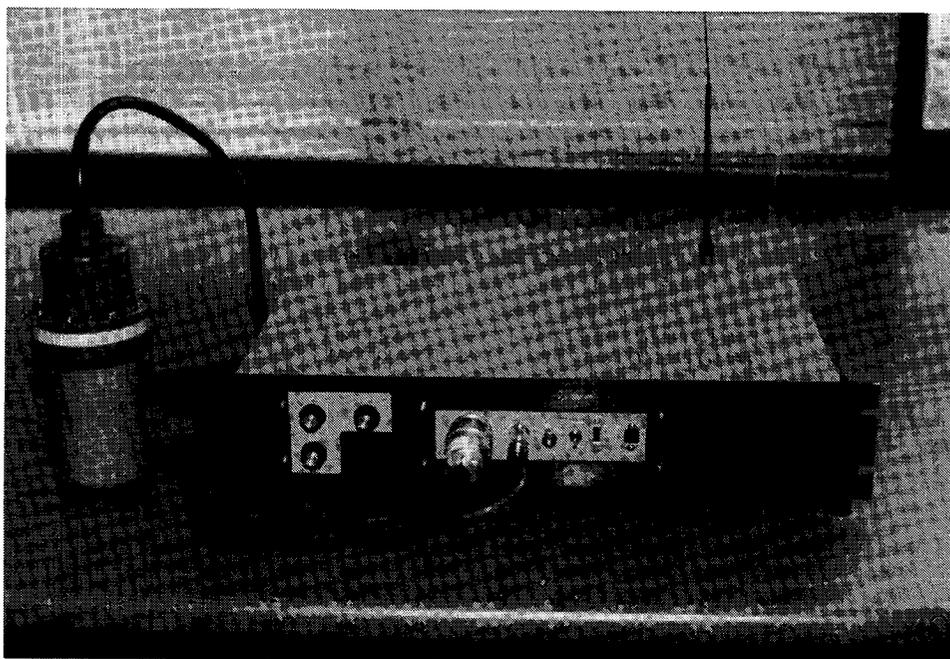


Photo 1-4 Transmitter

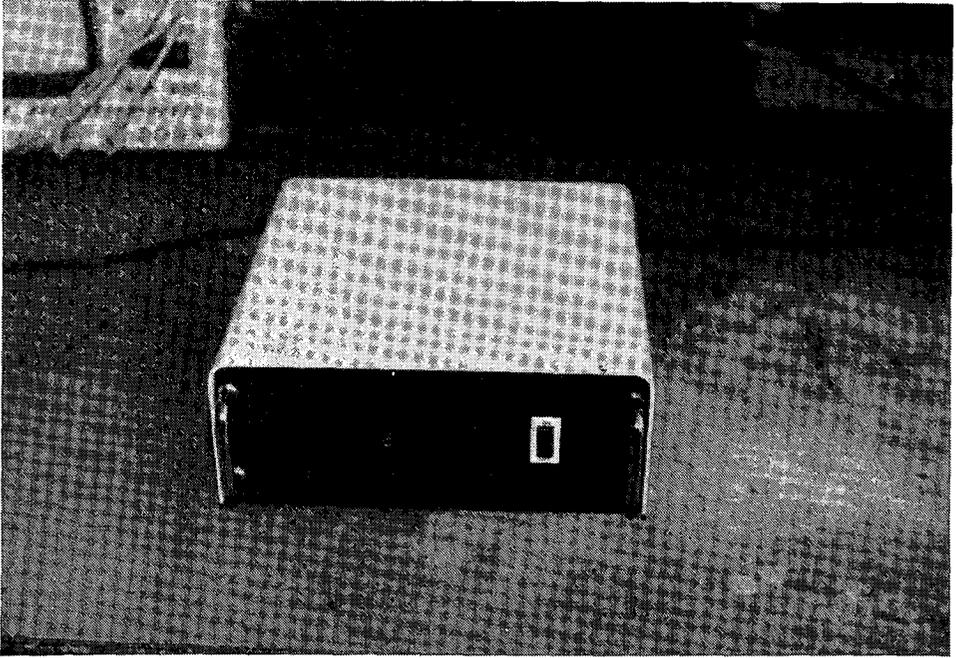


Photo 1-5 Receiver

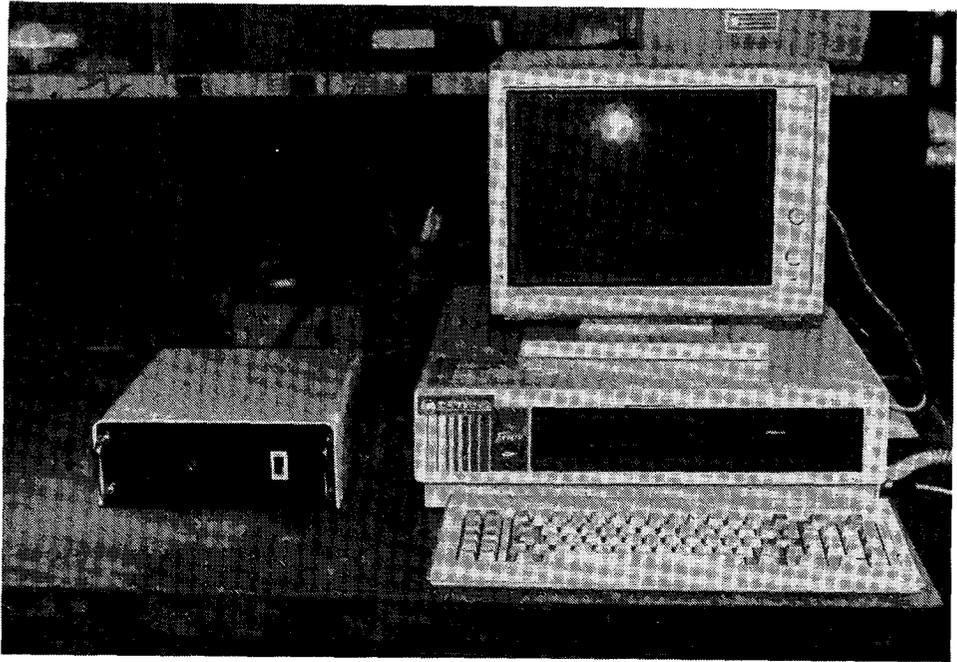


Photo 1-6 Receiver and data terminal(PC)



Photo 1-7 Transmitter antenna

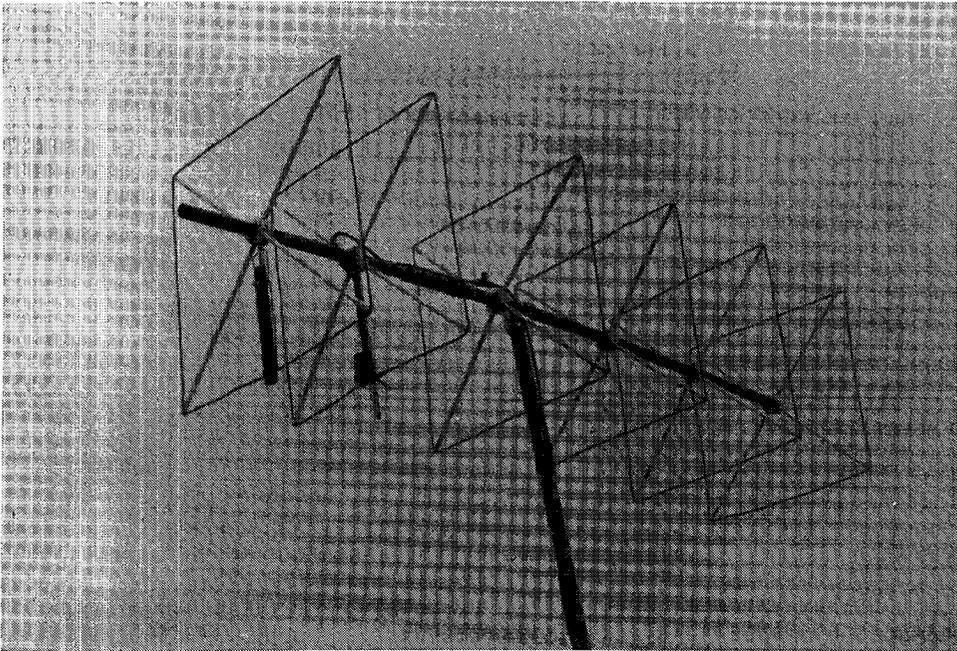


Photo 1-8 Receiver antenna

APPENDIX II

(현장실측자료)

1991. 4.

Time	Ch. 0	Ch. 1	Ch. 2	Ch. 3	Ch. 4
9 : 45 : 26	2045	2047	2045	2047	2047
9 : 45 : 27	2045	2045	2045	2045	2047
9 : 45 : 29	2047	2047	2045	2047	2047
9 : 45 : 31	2045	2045	2047	2045	2045
9 : 45 : 32	2047	2049	2047	2047	2049
9 : 45 : 34	2049	2049	2049	2049	2049
9 : 45 : 36	2049	2049	2049	2049	2049
9 : 45 : 37	2049	2049	2049	2049	2049
9 : 45 : 39	2049	2049	2049	2049	2049
9 : 45 : 41	2049	2049	2049	2049	2049
9 : 45 : 43	2049	2049	2049	2049	2049
9 : 45 : 44	2049	2049	2049	2049	2049
9 : 45 : 46	2049	2049	2049	2049	2049
9 : 45 : 48	2487	2487	2487	2487	2487
9 : 45 : 49	2935	2935	2937	2937	2939
9 : 45 : 51	3125	3125	3125	3125	3125
9 : 45 : 53	3105	3103	3099	3103	3099
9 : 45 : 55	3085	3087	3087	3087	3085
9 : 45 : 56	3077	3075	3077	3075	3077
9 : 45 : 58	3073	3073	3075	3075	3073
9 : 46 : 0	3073	3071	3071	3073	3071
9 : 46 : 1	3073	3073	3071	3073	3073
9 : 46 : 3	3069	3069	3069	3069	3069
9 : 46 : 5	3069	3069	3069	3067	3067
9 : 46 : 6	3069	3067	3069	3067	3069
9 : 46 : 8	3069	3067	3067	3067	3069
9 : 46 : 10	3069	3069	3069	3067	3069

Time	Ch. 0	Ch. 1	Ch. 2	Ch. 3	Ch. 4
9 : 46 : 12	3069	3069	3069	3069	3069
9 : 46 : 13	3067	3067	3067	3069	3067
9 : 46 : 15	3069	3069	3069	3069	3067
9 : 46 : 17	3071	3069	3067	3067	3071
9 : 46 : 18	3069	3069	3067	3069	3069
9 : 46 : 20	3069	3071	3067	3069	3067
9 : 46 : 22	3069	3069	3069	3069	3071
9 : 46 : 24	3261	3263	3263	3261	3263
9 : 46 : 25	3517	3517	3517	3517	3517
9 : 46 : 27	3771	3771	3771	3771	3771
9 : 46 : 29	3897	3897	3897	3899	3899
9 : 46 : 30	4087	4087	4087	4085	4087
9 : 46 : 32	4095	4095	4095	4093	4093
9 : 46 : 34	4093	4093	4093	4093	4093
9 : 46 : 36	4093	4093	4093	4093	4093
9 : 46 : 37	4093	4093	4093	4093	4093
9 : 46 : 39	4093	4093	4093	4093	4091
9 : 46 : 41	4093	4091	4093	4091	4091
9 : 46 : 42	4093	4093	4091	4093	4093
9 : 46 : 44	4093	4091	4091	4091	4093
9 : 46 : 46	4091	4091	4093	4091	4093
9 : 46 : 48	4093	4093	4091	4093	4091
9 : 46 : 49	4093	4091	4093	4093	4093
9 : 46 : 51	4093	4093	4091	4093	4091
9 : 46 : 53	4093	4093	4093	4091	4093
9 : 46 : 54	4091	4093	4093	4093	4093
9 : 46 : 56	4093	4093	4093	4093	4091
9 : 46 : 58	4093	4093	4093	4093	4093
9 : 46 : 59	4093	4093	4093	4093	4093
9 : 47 : 1	4093	4093	4093	4093	4093
9 : 47 : 3	4093	4093	4093	4091	4093

Time	Ch. 0	Ch. 1	Ch. 2	Ch. 3	Ch. 4
9 : 47 : 5	4093	4093	4093	4093	4093
9 : 47 : 6	4091	4091	4093	4091	4091
9 : 47 : 8	4093	4093	4093	4093	4093
9 : 47 : 10	4091	4093	4091	4093	4091
9 : 47 : 11	4093	4093	4093	4091	4093
9 : 47 : 13	4093	4093	4093	4093	4093
9 : 47 : 15	4093	4093	4093	4093	4093
9 : 47 : 17	4093	4093	4093	4093	4093
9 : 47 : 18	4093	4093	4093	4093	4093
9 : 47 : 20	4091	4091	4093	4093	4091
9 : 47 : 22	4091	4093	4091	4093	4091
9 : 47 : 23	4091	4091	4087	4091	4091
9 : 47 : 25	4091	4091	4091	4091	4093
9 : 47 : 27	4091	4091	4093	4091	4091
9 : 47 : 28	4091	4091	4089	4091	4091
9 : 47 : 30	4091	4091	4093	4091	4091
9 : 47 : 32	4091	4091	4091	4093	4091
9 : 47 : 34	4091	4093	4091	4091	4091
9 : 47 : 35	4091	4091	4091	4093	4091
9 : 47 : 37	4091	4091	4091	4091	4093
9 : 47 : 39	4089	4093	4091	4091	4091
9 : 47 : 40	4091	4091	4091	4091	4093
9 : 47 : 42	4091	4091	4091	4091	4091
9 : 47 : 44	4091	4091	4091	4091	4091
9 : 47 : 46	4091	4091	4091	4093	4091
9 : 47 : 47	4093	4091	4093	4091	4091
9 : 47 : 49	4093	4091	4091	4093	4093
9 : 47 : 51	4093	4091	4093	4093	4091
9 : 47 : 52	4091	4093	4091	4091	4091
9 : 47 : 54	4093	4093	4093	4093	4093
9 : 47 : 56	4091	4093	4093	4091	4093

Time	Ch. 0	Ch. 1	Ch. 2	Ch. 3	Ch. 4
9 : 47 : 58	4093	4093	4093	4093	4093
9 : 47 : 59	4093	4093	4091	4093	4093
9 : 48 : 1	4093	4093	4093	4093	4093
9 : 48 : 3	4093	4093	4093	4093	4093
9 : 48 : 4	4093	4093	4093	4093	4093
9 : 48 : 6	4093	4093	4093	4093	4093
9 : 48 : 8	4093	4091	4091	4093	4093
9 : 48 : 10	4093	4093	4093	4093	4093
9 : 48 : 11	4093	4093	4093	4093	4091
9 : 48 : 13	4091	4091	4093	4093	4093
9 : 48 : 15	4091	4093	4093	4093	4093
9 : 48 : 16	4093	4091	4093	4091	4093
9 : 48 : 18	4091	4091	4093	4093	4091
9 : 48 : 20	4093	4093	4093	4093	4093
9 : 48 : 21	4093	4091	4091	4093	4093
9 : 48 : 23	4091	4093	4093	4093	4093
9 : 48 : 25	4091	4093	4093	4093	4093
9 : 48 : 27	4091	4091	4093	4093	4093
9 : 48 : 28	4093	4093	4093	4093	4091
9 : 48 : 30	4093	4091	4091	4093	4093
9 : 48 : 32	4093	4091	4093	4091	4091
9 : 49 : 49	4091	4091	4091	4091	4089
9 : 49 : 50	4091	4091	4091	4093	4089
9 : 49 : 52	4091	4091	4091	4091	4091
9 : 49 : 54	4091	4091	4091	4091	4091
9 : 49 : 55	4091	4091	4091	4091	4091
9 : 49 : 57	4091	4091	4093	4091	4091
9 : 49 : 59	4091	4091	4091	4091	4089
9 : 50 : 1	4091	4091	4093	4091	4091
9 : 50 : 2	4091	4091	4091	4091	4089
9 : 50 : 4	4089	4091	4091	4089	4089

Time	Ch. 0	Ch. 1	Ch. 2	Ch. 3	Ch. 4
9:50:6	4091	4091	4089	4089	4089
9:50:7	4091	4091	4089	4091	4091
9:50:9	4089	4089	4091	4089	4089
9:50:11	4091	4091	4089	4089	4091
9:50:12	4091	4091	4091	4091	4091
9:50:14	4089	4091	4091	4091	4091
9:50:16	4089	4091	4091	4091	4091
9:50:18	4091	4089	4091	4091	4091
9:50:19	4089	4091	4091	4089	4091
9:50:21	4091	4091	4091	4091	4089
9:50:23	4091	4089	4091	4091	4089
9:50:24	4091	4089	4091	4091	4089
9:50:26	4089	4091	4089	4089	4089
9:50:28	4091	4091	4089	4091	4091
9:50:30	4091	4089	4089	4091	4091
9:50:31	4089	4091	4089	4091	4091
9:50:33	4091	4091	4089	4091	4091
9:50:35	4089	4091	4091	4091	4091
9:50:36	4091	4089	4089	4091	4091
9:50:38	4089	4091	4089	4089	4089
9:50:40	4091	4089	4089	4091	4089
9:50:41	4091	4091	4089	4089	4089
9:50:43	4089	4089	4091	4091	4091
9:50:45	4091	4091	4091	4089	4091
9:50:47	4091	4089	4091	4091	4089
9:50:48	4089	4091	4091	4089	4091
9:50:50	4091	4089	4089	4091	4091
9:50:52	4091	4089	4089	4089	4091
9:50:53	4089	4091	4091	4091	4089
9:50:55	4089	4091	4091	4091	4089
9:50:57	4091	4089	4091	4091	4091

Time	Ch. 0	Ch. 1	Ch. 2	Ch. 3	Ch. 4
9 : 50 : 59	4091	4089	4091	4091	4091
9 : 51 : 0	4089	4089	4091	4089	4091
9 : 51 : 2	4091	4091	4089	4091	4091
9 : 51 : 4	4091	4089	4091	4091	4091
9 : 51 : 5	4091	4091	4089	4091	4091
9 : 51 : 7	4091	4091	4091	4089	4091
9 : 51 : 9	4089	4091	4091	4089	4091
9 : 51 : 11	4089	4091	4091	4091	4091
9 : 51 : 12	4091	4089	4091	4091	4091
9 : 51 : 14	4089	4089	4089	4091	4091
9 : 51 : 16	4091	4091	4089	4089	4091
9 : 51 : 17	4091	4089	4091	4089	4091
9 : 51 : 19	4089	4089	4091	4089	4091
9 : 51 : 21	4091	4091	4091	4091	4091
9 : 51 : 23	4091	4091	4089	4089	4091
9 : 51 : 24	4091	4091	4089	4089	4091
9 : 51 : 26	4091	4089	4089	4089	4091
9 : 51 : 28	4091	4089	4091	4091	4091