DEVELOPMENT OF AN INTELLIGENT NICKEL-CADMIUM BATTERY CHARGER

By Dirk Francois Raath

Submitted in part fulfillment of the requirements laid down

for the

Master's Diploma

in the School of Electrical Engineering

at the Cape Technikon

November 1991

Research And Development Centre

Department of Posts and Telecommunications

Declaration

I declare that the contents of this thesis represents my own work and the opinions contained here are my own. It has not been submitted before for any examination at this or any other institute.

D.F.Raath

(Signature)

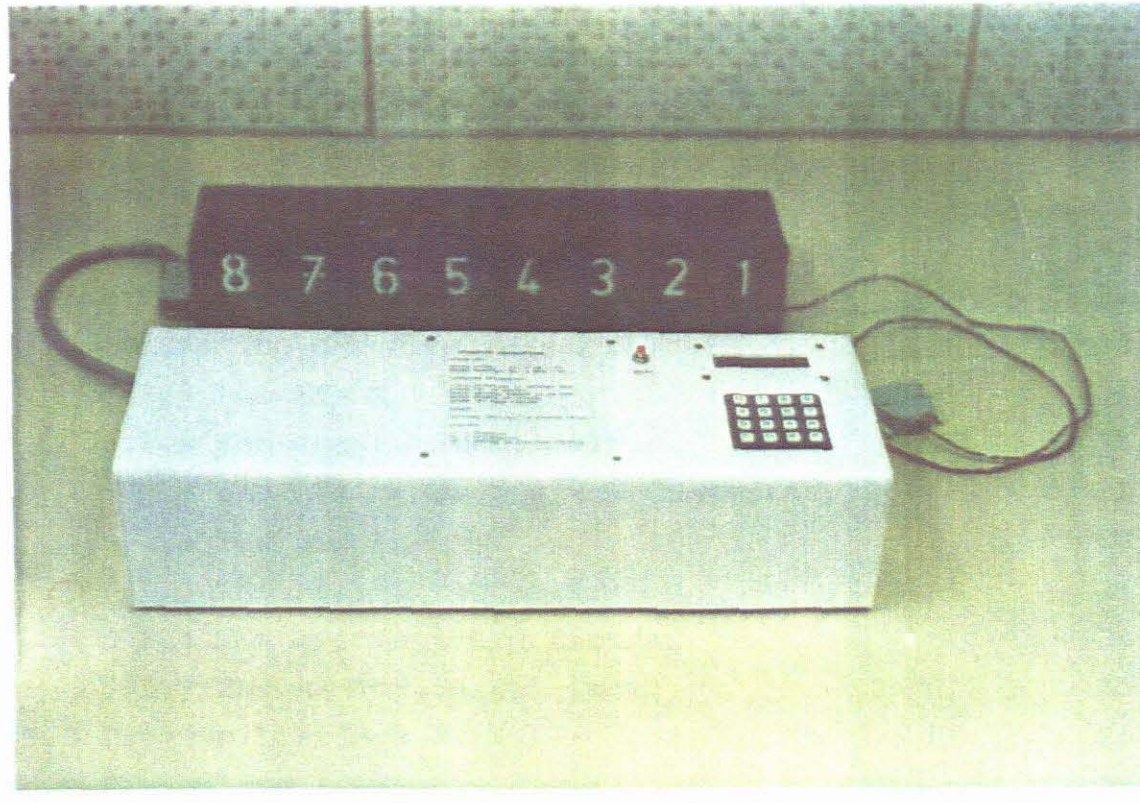
Abstract

This thesis describes the control system as well as the charge and discharge circuitry developed to efficiently charge nickel-cadmium batteries. Due to incorrect usage and charging of these batteries their life-spans and output capacities are drastically decreased. Efficiency is improved by cycling the batteries which extends their life-span and performance. "Cycling" refers to a discharge and charge process.

Opsomming

Hierdie verhandeling beskryf die beheereenheid sowel as die laai en ontlaai kringe wat ontwerp is om nikkel-cadmium batterye effektief te laai. As gevolg van die verkeerde gebruik en laaiproses van die batterye word hul lewensduurte en leweringsvermoë drasties beperk. Doeltreffendheid word verhoog deur die batterye te "sirkuleer" wat hul lewensduurte en lewering verbeter. "Sirkuleer" verwys na 'n ontlaai en laai proses.

INTELLIGENT NI-CAD CHARGER COUPLED TO PC FOR DATA LOGGING



INTELLIGENT NI-CAD CHARGER AS IT WOULD BE USED IN PRACTICE

CONTENTS

1. Objective.

The objective of this project is to improve the efficiency of nickel-cadmium (Ni-Cad) batteries which are used by the Security Section of the Department of Posts and Telecommunications and others. These 12 volt batteries are used to power two-way radio transceivers.

Limited usage of these radio systems causes only a slight discharge of the batteries. The batteries are not allowed to discharge fully under normal conditions as they are too frequently charged. If a battery is repeatedly cycled shallowly it appears to lose its latent capacity. For example, if a battery is repeatedly cycled to a 20 percent depth of discharge, the additional 80 percent of its capacity will become unavailable to the user. This result is called the "memory effect". A consequence of the memory effect is that the batteries no longer function properly when used extensively before being charged again. It is evident that the life-span of these batteries is shortened due to this improper usage.

A further improvement, which is achieved as a result of the first, is to increase the life-span of the batteries as they are costly to replace.

2. Introduction.

The Research And Development Centre undertakes development of electronic circuitry (analog and digital) by modifying existing equipment and building new systems. Specific projects like this one are usually requested by various sections of the Department of Posts and Telecommunications.

2.1 The Batteries and Specifications.

A battery consists of ten 1.25 volt sintered plate nickel-cadmium cells connected in series. A cell is an alkaline battery. Its electrolyte is a neutral base which does not react chemically with active materials. The electrolyte acts only as a transfer medium for electrons. The cell is constructed of nickel support plates which are impregnated with the active materials in a powdered form. These plates are then formed and assembled into a cell. The case is made of nickel plated steel. The use of powdered active materials results in large surface areas which lowers the internal resistance of the cell. This makes high rates of discharge for short periods of time possible. The cells contain a one-shot safety vent in case the battery is massively overcharged and produces gases. If the safety vent is used, the cell's electrolyte quickly dries out and the cell is effectively ruined.

The cell has a lifetime of up to 500 cycles. Since most batteries are improperly charged, it is common to see them fail after less than 100 cycles.

The battery specifications are:

Type: Sealed Sintered Plate Nickel-Cadmium
Voltage: 12 V Nominal
Capacity: 600mA/Hr

2.2 Existing Charging System.

The batteries are charged regularly by inserting them into a self contained charger unit. It automatically starts to charge a battery when it is inserted and continues to do so until the battery is removed. The charging time is not controlled and the batteries may be over- or undercharged when removed from the unit.

Referring to the circuit diagram shown in Fig. 1. it can be seen that there is no provision for current regulation. The charge current is therefore not constant and varies depending on the battery's condition. The maximum specified charge current (See Section 4.2.1) is easily exceeded and can cause damage to the cells. The charging current also drops over a period as the battery voltage increases.

$$I\ (\text{charge}) = \frac{\text{Charging Voltage - Battery Voltage}}{\text{resistance}}$$

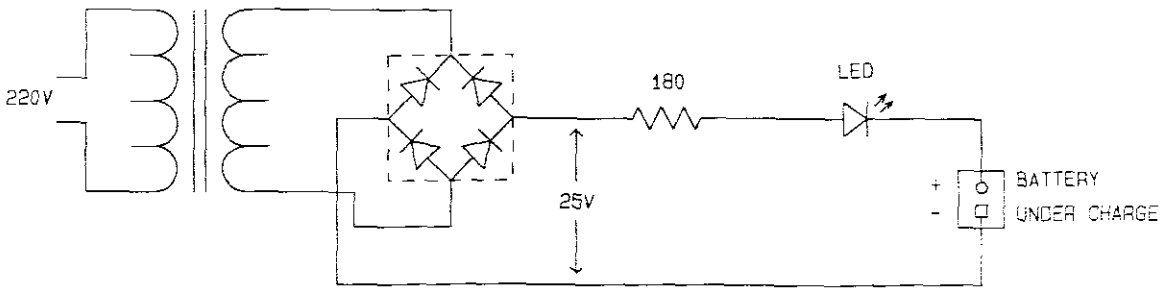


Fig. 1.

3. Use Of The Intelligent Battery Charger.

The operating instructions of the unit are described in Section 7.

3.1 Operation Of The Charger.

Refer to Fig. 3. Once the batteries have been inserted and an 'F' entered via the keypad the process will start. A status word is generated to indicate at which positions batteries were inserted. This status word is rotated through the carry flag which will be set every time a battery is detected. Since the condition of a battery is not known when it is inserted in the charger, the battery voltage is scanned and compared with the discharge cutoff voltage. When under load the output voltage at which a cell is considered to be empty is called the "discharge cutoff voltage" and between 10 and 20 percent of the capacity remains at this point. Figure 2 gives the cutoff voltages for the various discharge rates. Using the C/3 rate the battery voltage is therefore 10 Volts for 10 cells (C = Capacity of battery in AHrs; 3 = Charge time in Hrs). Discharging beyond this point, ie. C/0 to C/2, lowers the battery's efficiency and shortens its life.

If the battery voltage exceeds this cutoff voltage, the battery will be discharged until the battery voltage has dropped to the cutoff value. The discharge process will then be stopped automatically.

If the battery voltage is smaller or equal to the cutoff voltage the battery will be charged for 14 hours. The charging process is started by ensuring that the discharge circuit is switched off and by taking the time from the real-time clock and adding 14 hours to it. This new time is then stored as an alarm time and the charging circuit is activated. While this charging process is in progress, the alarm time is regularly compared with the

real-time clock. When they are found to be equal, 14 hours have passed and the charging process will be stopped. Although the battery voltage rises above the cutoff value during the charging process, the discharge circuit is disabled for that specific battery.

After completion of the charge process, the cycle counter is decremented. If it is unequal to zero then the battery will be discharged again until the cutoff value is reached and then charged again for 14 hours. This cycling will continue until all the cycles have been completed. A flag-bit in the status word will then be reset to prevent the battery from being cycled further. The process will always stop at the end of a charge cycle. While the process is in progress more batteries may be entered. When another battery is entered the status word is changed (a flag-bit will be set) to indicate its presence.

This cycling process will be performed simultaneously and independently for all batteries entered. The status of all batteries entered will be indicated on the display.

3.2 Improved Efficiency.

The first and main objective is to improve the efficiency of the batteries. This is achieved by cycling the batteries which ensures that the batteries do not build up a memory effect or to erase the effect in batteries where it exists already.

The discharge cycle ensures that the batteries are artificially loaded to simulate a normal discharge process. This prevents the formation of a memory. The whole process is controlled and the batteries are not over- or under-charged or over-discharged. The resultant absence of the memory effect ensures that the battery will deliver the required power over the full period.

3.3 Increased Life-Span.

As a result of the improved efficiency the batteries are able to deliver the required power over the expected period. Previously a battery would fail as soon as it was used outside its memory capacity. This means that a battery can now be used over a longer period before it needs to be charged again. The overall life-span is also increased as the batteries will last longer.

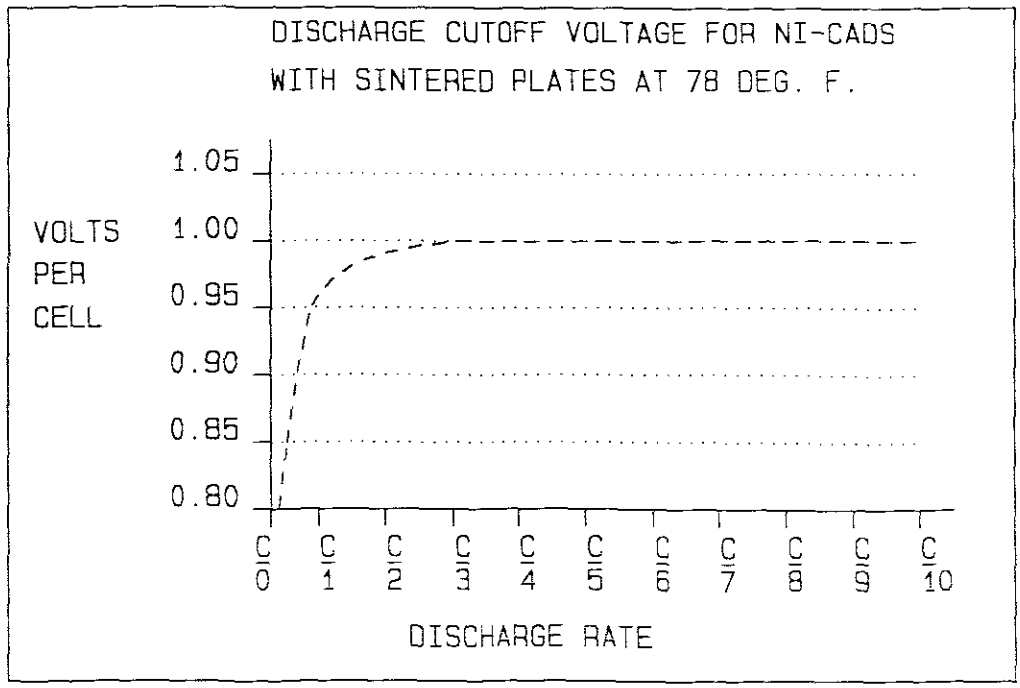


Fig. 2.

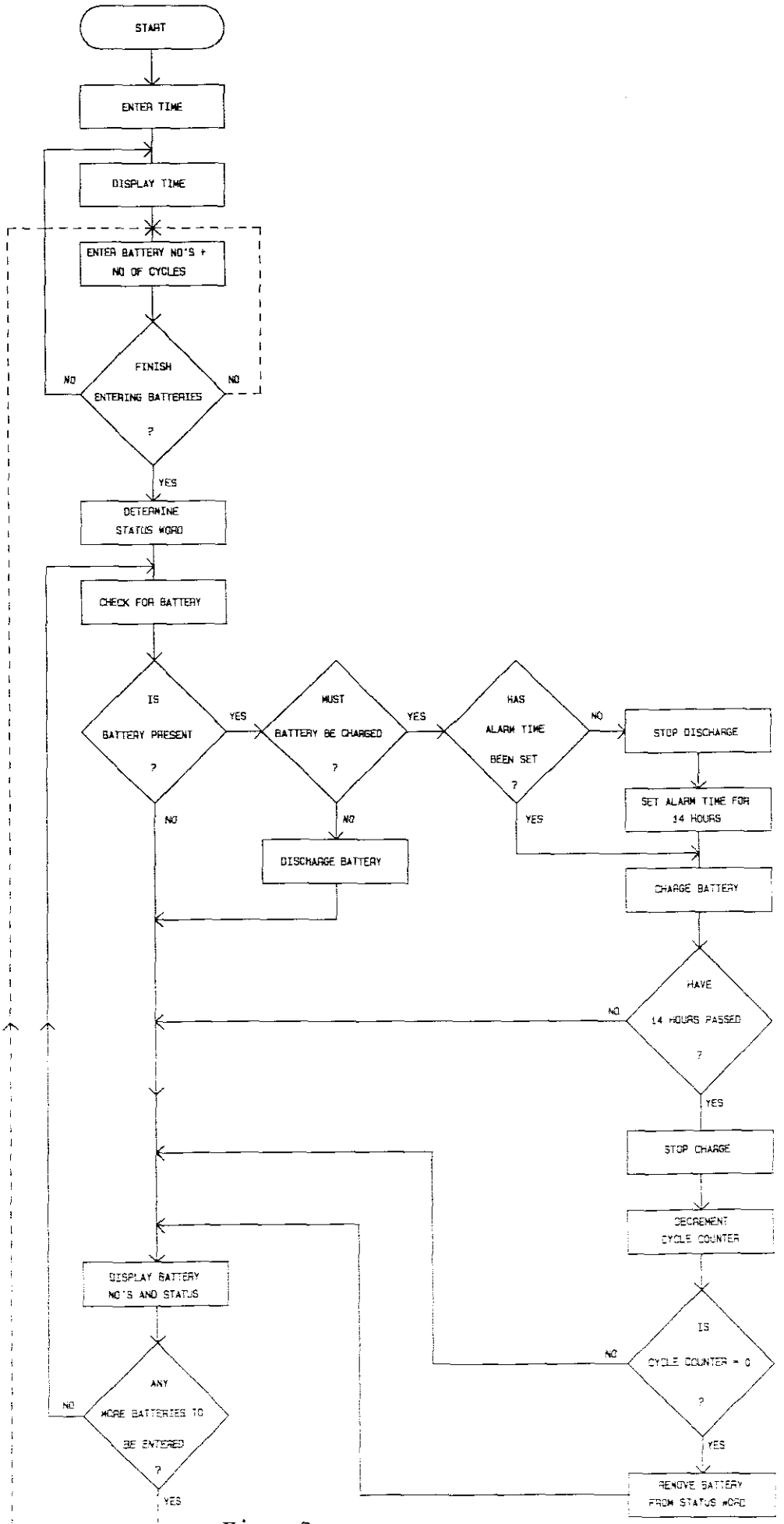

START
ENTER TIME
DISPLAY TIME
ENTER BATTERY NO'S & NO OF CYCLES
FINISH ENTERING BATTERIES ?
NO
NO
YES
DETERMINE STATUS WORD
CHECK FOR BATTERY
IS BATTERY PRESENT ?
YES
NO
MUST BATTERY BE CHARGED ?
YES
NO
DISCHARGE BATTERY
HAS ALARM TIME BEEN SET ?
NO
YES
STOP DISCHARGE
SET ALARM TIME FOR 14 HOURS
CHARGE BATTERY
HAVE 14 HOURS PASSED ?
NO
YES
STOP CHARGE
DECREMENT CYCLE COUNTER
IS CYCLE COUNTER = 0 ?
NO
YES
REMOVE BATTERY FROM STATUS WORD
DISPLAY BATTERY NO'S AND STATUS
ANY MORE BATTERIES TO BE ENTERED ?
NO
YES

Fig. 3.

4. Hardware Design.

4.1 The Main Power Supply.

The processor unit requires a 5 volt supply. Testing the unit under full load has shown that a current of about 150 mA is required. The 5 volt supply is also used to switch the charge and discharge circuits on and off.

The processor, input/output port and EPROM are sensitive to supply voltage variations. A steady supply must therefore be provided. Refer to Fig. 4. A rectifier bridge is used for rectification. Capacitor C2 provides the first stage of smoothing. The voltage at this point is about 7.5 volts. By varying the variable resistor R2 the output voltage of the regulator REG2 can be adjusted to ensure a voltage of 5V at the processor (Refer to section 6.1). The output capacitor C3 acts as a further smoothing circuit to filter any remaining ripple voltage.

The charge/discharge unit requires a supply voltage of about 24 volt which is only used in the charging circuit. Rectification is again done by a rectifier bridge. The output thereof is smoothed by C1 giving an input voltage of about 29.6 volt to the regulator. This 24 volt regulator requires a minimum input voltage of 27.1 volt. The output voltage was found to be a constant 23.2 volt. Further smoothing is not required as no sensitive devices are present in the charge circuit.

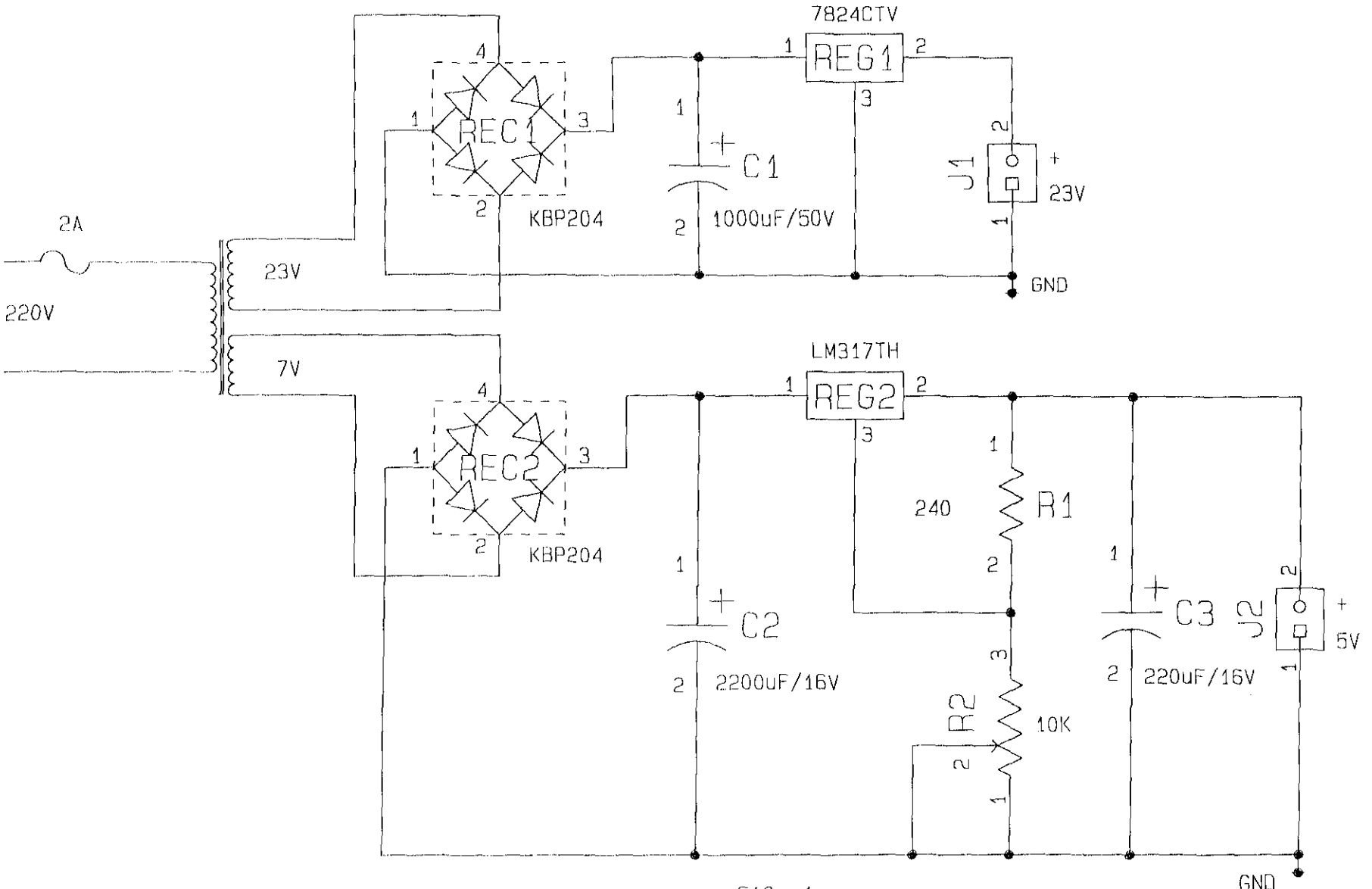

7824CTV
REG1
REC1
KBP204
C1
1000uF/50V
J1
23V
2A
220V
23V
7V
GND
LM317TH
REG2
REC2
KBP204
240
R1
C2
2200uF/16V
R2
10K
C3
220uF/16V
J2
5V
GND

FIG. 4.

4.2 The Charge/Discharge Unit.

4.2.1 The Charge Circuit.

Ni-cads left in a discharged state for extended periods will become inefficient and short lived. It is important to charge a battery at a constant current or else the charging period is influenced. A good general rapid charge rate for the sealed sintered plate battery is the C/10 rate for 12 to 15 hours (C = Capacity of battery in AHrs; 10 = Charge time in Hrs). The use of a simple resistive network will cause the current to vary depending on the condition of the batteries as is the case with the existing charger.

Refer to Fig. 5 (Page 14). A 5 volt regulator X1 is used and connected as indicated. The current is determined by the value of R6. The output voltage of the regulator across R6 will always be kept constant at 5 volts, regardless of the battery's condition, which ensures that the charging current stays constant. The required charge current is determined as follows:

Charge Current = Battery Capacity/10 (C/10)
= 600mAhrs/10Hrs
= 60mA

R6 = Regulator Output Voltage/Charge current
= 5V/60mA
= 83 ohm
≈ 82 ohm

The switching circuit consists of transistors T1 and T2. The values of resistors are determined as follows:

T1 Ic = 60mA (charge current)
Ib = Ic/hfe
= 60mA/40
= 1.5mA (to switch T1 on fully)

R49 = (Vcc-Vbe)/Ib
= (23-0.7)V/1.5mA
= 14.867 K ohm
~ 15 K ohm

T2 Ic = Ib (T1)
Ib = Ic/hfe
= 1.5mA/110
= 13.6 μA

R1 = (switching voltage-Vbe)/Ib
= (5-0.7)V/13.6μA
= 316 K ohm

The maximum base current allowed for T1 is 2mA. A 10 K ohm resistor is used to switch it on fully. The base current is 0.43mA and well within the limit.

4.2.2 The Discharge Circuit.

Due to the low internal resistance of the battery it is possible to discharge it very rapidly. Discharge rates of between C/3 and C/20 are normally used. (C/3 is used in this application).

This circuit is virtually the same as that of the charge circuit. Refer to Fig. 5. Only the differences will be discussed.

The battery is discharged through T3, the 5 volt regulator and R5. The current is determined as follows:

Discharge Current = Battery Capacity/3 (C/3)
= 600mAHrs/3Hrs
= 200mA

R5 = Regulator Output Voltage/Discharge current

= 5V/200mA
= 25 ohm (22 ohm used)

The switching circuit consists of T3 and T4. The value of resistor R50 is determined as follows:

T3 Ic = 200mA (discharge current)
Ib = Ic/hfe
= 200mA/40
= 5mA (to switch T3 on fully)

R50 = (Battery Voltage-Vbe)/Ib
= (13-0.7)V/5mA
= 2.46 K ohm
~ 2.7 K ohm

The maximum base current for T3 is 5mA (hfe = 40). A maximum battery voltage of 13 volt is assumed. A value of 2.7 K ohm is selected for R50 to prevent the base current from exceeding the maximum allowed value.

R65 is required to ensure that the discharge circuit is switched off during a power failure to the unit. It pulls the base of T4 to ground and keeps it switched off. During the discharge process a potential of 5 volt is applied to the base of T4 via R2. The actual base voltage is:

base voltage = switching voltage x R65/(R2 + R65)
= 5V x 4.7K/(10K + 4.7K)
= 1.6 V

This voltage is adequate to switch T4 on fully.

4.2.3 The Voltage Converter.

Refer to Figure 5. Before a battery is cycled its condition is checked by taking a voltage reading via the A_D line. A simple voltage divider is used to scale the battery

voltage down to about 5 volt. In the idle condition (before starting or after completing the cycling process) a minimum current must be drawn from the battery. A current of 1mA and a maximum battery voltage of 14 volts is assumed. R3 and R4 are determined as follows:

$$I = \text{Battery Voltage}/(R3 + R4)$$
$$1mA = 14V/(R3 + R4)$$
$$R3 + R4 = 14V/1mA$$
$$= 14 \text{ K ohm}$$

$$5V = R4/(R3 + R4) \times \text{Battery Voltage}$$
$$= R4/14K \times 14V$$
$$R4 = 5V/14V \times 14K$$
$$= 5.0 \text{ K ohm}$$
$$\tilde{} \ 5.1 \text{ K ohm}$$

$$R3 = 14K - R4$$
$$= 14K - 5.1K$$
$$= 8.9 \text{ K ohm}$$
$$\tilde{} \ 9.1 \text{ K ohm}$$

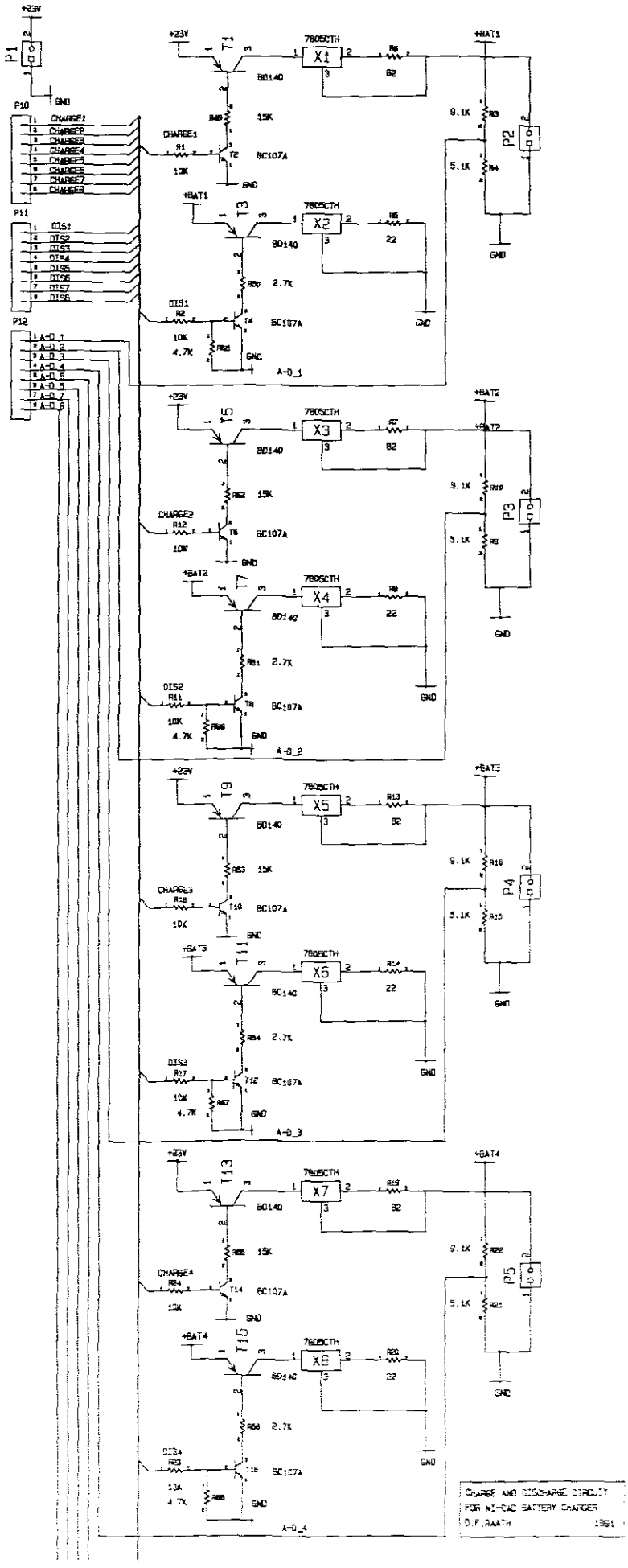
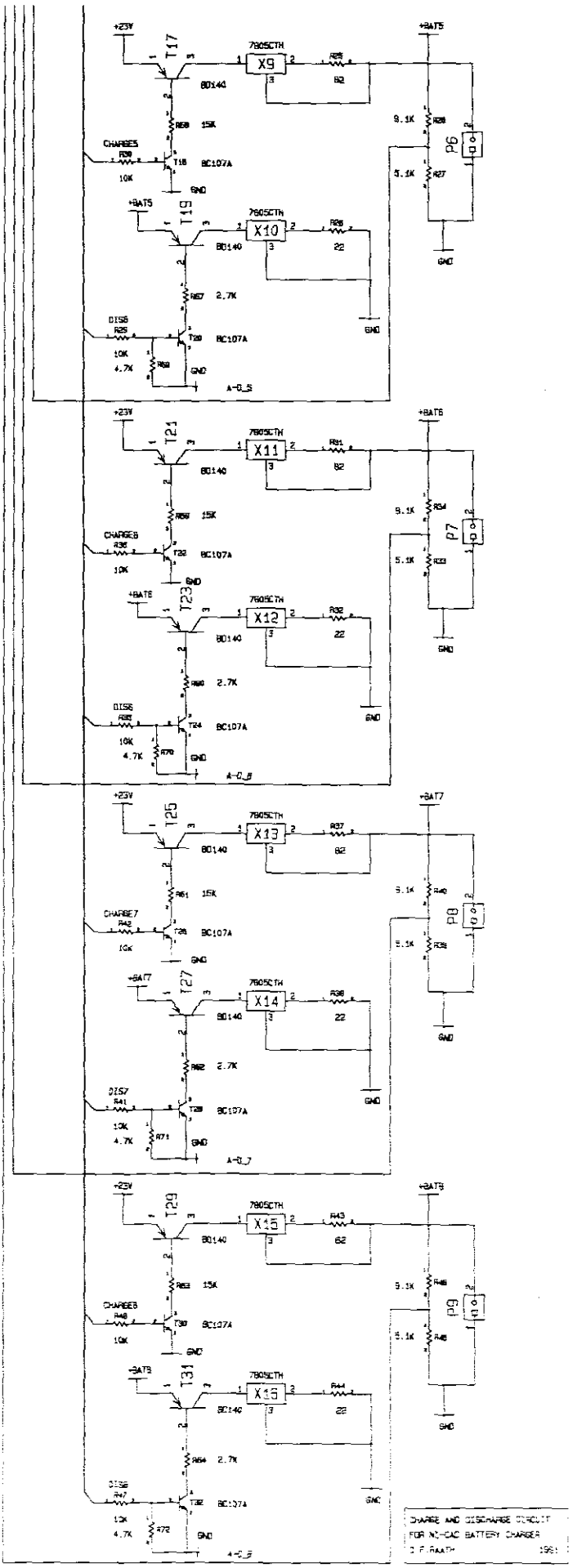

CHARGE AND DISCHARGE CIRCUIT
FOR NI-CAD BATTERY CHARGER
D.P.RAATH 1991

Fig. 5.


+23V
T17
7805CTH
X9
R85
82
BD140
R58 15K
CHARGE5
R59
T18 BC107A
10K
GND
+BAT5
T19
7805CTH
X10
R86
22
BD140
R57 2.7K
DIS5
R25
T20 BC107A
10K
4.7K
R60
GND
A-D_5
9.1K
R28
P6
5.1K
R27
GND
+23V
T21
7805CTH
X11
R91
82
BD140
R59 15K
CHARGE6
R56
T22 BC107A
10K
GND
+BAT6
T23
7805CTH
X12
R92
22
BD140
R60 2.7K
DIS6
R26
T24 BC107A
10K
4.7K
R70
GND
A-D_6
9.1K
R34
P7
5.1K
R33
GND
+23V
T25
7805CTH
X13
R37
82
BD140
R61 15K
CHARGE7
R42
T26 BC107A
10K
GND
+BAT7
T27
7805CTH
X14
R38
22
BD140
R62 2.7K
DIS7
R41
T28 BC107A
10K
4.7K
R71
GND
A-D_7
9.1K
R40
P8
5.1K
R39
GND
+23V
T29
7805CTH
X15
R43
82
BD140
R63 15K
CHARGE8
R48
T30 BC107A
10K
GND
+BAT8
T31
7805CTH
X16
R44
22
BD140
R64 2.7K
DIS8
R47
T32 BC107A
10K
4.7K
R72
GND
A-D_8
9.1K
R46
P9
5.1K
R45
GND
CHARGE AND DISCHARGE CIRCUIT
FOR NI-CAC BATTERY CHARGER

4.3 The Processor Unit.

Refer to Fig. 6 for the controller card schematic diagram.

4.3.1 The Processor And Latch.

An 8031AH 8-Bit Control processor is used which provides extensive on-chip support for one-bit variables as a separate data type, allowing direct bit manipulation and testing in control and logic systems that require Boolean processing. The Address/Data is shared on Port 0 and the lower address-byte is latched by the 74LS573. The processor specifications are:

128 Bytes Internal Data Memory
32 I/O Lines (Four 8-Bit Ports) Providing 5 Interrupts
2 x 16-Bit Timers/Event Counters
64K Program Memory (Only 8K is used in this application)
5V Operating Voltage
3.5 to 12.0 MHz Oscillator Freq. (external clock source used)

4.3.2 The Input/Output Port.

The 8255A has 24 Programmable I/O pins (three 8-bit ports) and is used to interface the charge/discharge unit with the processor system. Port A is used to switch the charge circuits and port B to switch the discharge circuits. The eight data pins (D0-D7) receive data from the processor and send it to the appropriate output port. Port control is done via the two address pins A0 and A1. Operating voltage is 5V. The output port-voltage switches between 0V and 5V.

4.3.3 The Analog-To-Digital Converter.

The ADC0804 is an 8-bit successive approximation A/D

converter. The TRI-STATE output latches drive the data bus directly and no interfacing logic is needed. It converts the battery voltage to a digital value varying between 0 and 255 eg. 0V = 0 (digital) and 5V = 255 (digital). The reference voltage, which is obtained by using a 2.5V zener diode (LM336-2.5), is doubled internally by the converter. The battery voltage is compared with this reference voltage and converted. The specifications are:

4.5 to 6.3V supply voltage (5V used)
0 to 5V analog input voltage
100uS Conversion time

4.3.4 The Hex Keyboard Encoder.

The 74C922 16-Key encoder has an internal debounce circuit that needs only a single external capacitor. The Data Available output is used as an interrupt to indicate when a key is pressed. Key-values are encoded into digital values from 0 to 15. The supply voltage may vary between 3 and 15V.

4.3.5 The Multiplexer.

The CD4051B is a single 8-Channel analog multiplexer/ demultiplexer which is used as a multiplexer in this application. It has three binary control inputs A, B and C which gives an 8-channel selection. The output, pin 3, is taken to the input of the analog-to-digital converter. The specifications are:

5V to 15V Supply voltage (5V used)
-0.5V to 15.5V Input voltage (analog) on the 8 channels (0V to 5V used)
-0.5V to 15.5V Output voltage to the A/D converter. (0V to 5V used)

4.3.6 The Real-Time Clock.

The MC146818A Real-Time Clock provides the oscillator frequency for the processor via the CKOUT pin, which eliminates the need for a crystal at the processor. It also keeps the time. With the CKFS-pin tied to +5V, the CKOUT frequency is the same as that of the crystal. The clocks' main function however is to time the charging period of 14 hours. The specifications are:

Complete Time-of-the-day 12- to 24-Hour clock.
Alarm and one hundred year calendar.
Bus Compatible Interrupt.
Clock Output to be used as Microprocessor Clock Input.
50 Bytes of lowpower static RAM.
3 to 6V Operation.

Only the Time-of-the-day, Interrupt and Clock Output are used.

4.3.7 The Display.

The LTN211 is a 5x7 dot, 16-character, 2-line dot matrix LCD module, with driver and controller mounted on a single printed circuit board. Contrast is adjusted (R9) by varying the voltage between 0 and 5V. It has a built-in 160 character generator. The supply voltage must be between 4.75 and 5.25V.

4.3.8 The EPROM.

The Intel 2764 is a 5V only, 65,536-bit ultraviolet erasable and electrically programmable read-only memory (EPROM). It has a standby mode which reduces power consumption without increasing access time. It is selected by applying a TTL-high signal to the Chip-Enable input (pin 20). The supply voltage may vary by approximately 5%.

4.3.9 The RS-232 Port Facility.

The levels for the serial port are provided by an analog circuit using two transistors, one to transmit and one to receive. The positive RS-232 transmit level (OT) is provided by using C2 as a voltage doubling capacitor. The negative receive level (IR) is provided by the existing serial port from the personal computer. The serial port connections are:

IR to Transmit Pin (PC-side)
OT to Receive Pin (PC-side)

Refer to PC manual for RS-232 connections.

4.3.10 The 3-To-8 Line Decoder.

The processor board is memory-mapped and selection of the various devices eg. RT-Clock, display etc. is done by the 74LS138 under processor control. Three inputs (A, B and C) provides a selection of eight outputs. Only six outputs are used in this application. The specifications are:

2 to 6V supply voltage
0 to 6V input/output voltage

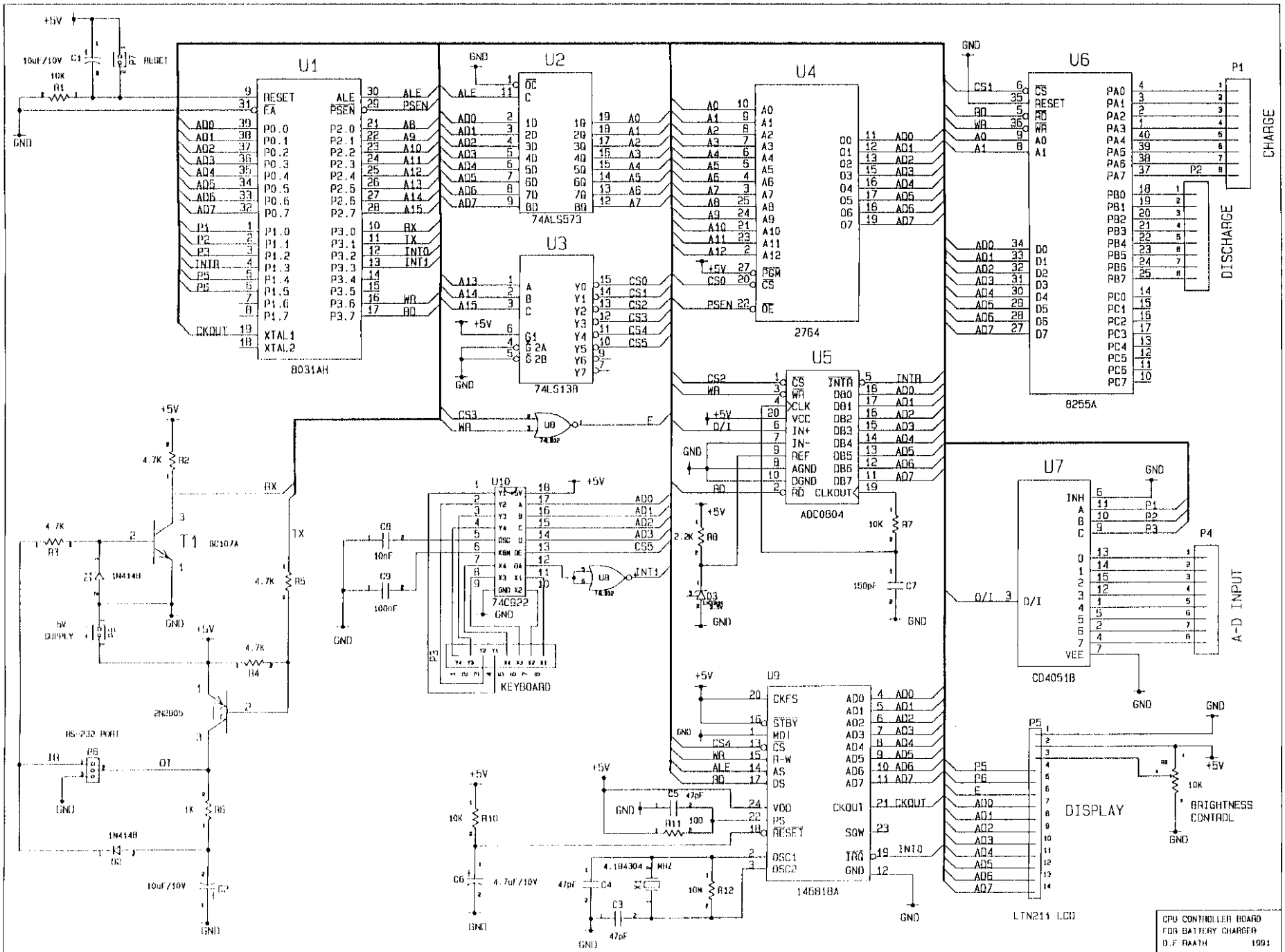

U1
8031AH
U2
74ALS573
U3
74ALS138
U4
2764
U5
ADC0804
U6
8255A
U7
CD4051B
U9
14681BA
U10
74C922
T1
BC107A
2N2905
8051 PORT
KEYBOARD
DISPLAY
LTN211 LCD
A-D INPUT
CHARGE
DISCHARGE
BRIGHTNESS CONTROL
CPU CONTROLLER BOARD
FOR BATTERY CHARGER
D.F RAATH 1991

Fig. 6.

5. Software Development.

The main program was written in assembler using the ASM51 compiler. A modular programming approach was used in order to simplify the writing thereof, debugging and to ease understanding of the programming sequence. Writing in assembler also reduces execution time to a minimum. This is important to ensure that after 14 hours of charging time the charge circuit will be switched off.(Refer to Section 6.2).

A program was also written in GWBASIC to capture data via the serial port of a PC and to convert it to battery voltages. See Section 5.2 and 5.3 for program listings and flowcharts.

5.1. Addressing And Control Words.

The processor board is memory-mapped and the 8K x 8-bit EPROM allows for eight 8K bit-blocks to be addressed.

5.1.1 The EPROM.

CS-0 selects the EPROM. No control words are necessary.

Start Address: 0000-Hex
End Address : 1FFF-Hex

5.1.2 The Input/Output Port.

CS-1 selects the 8255-I/O Port. The Port has 24 I/O pins and is programmed in three groups of 8 pins, port A, B and C. Port C is not used. Ports A and B are programmed to be output ports only. The Control Word is written to the Control Word Address to set the port up for the appropriate mode of operation.

Control Word: 81-Hex
Control Word Address: 2003-Hex

Port A Address: 2000-Hex
Port B Address: 2001-Hex
Port C Address: 2002-Hex (Not used)

Port A and port B are used to switch the charge and discharge circuits on/off respectively.

5.1.3 The Analog-To-Digital Converter.

CS-2 selects the ADC0804 Converter. No Control Words are required to initialise the Converter. Once the pointer (DPTR) is directed to the Converter, any code may be sent to start the conversion process eg.:

```
MOV     DPTR,#4000H  (Converter Address = 4000-Hex)
MOVX    @DPTR,A (Contents of Accumulator sent to Converter)
```

5.1.4 The Display.

CS-3 selects the LTN211 Display. The LTN211 incorporates an extensive instruction set: display clear, cursor home, display ON/OFF, character blink, cursor shift and display shift. The instructions are sent to Control Word Address 6000-Hex. The RS-line must be low when Control Words are sent to the display. It must be set when data is written to or read from the display. The R/W-line must be low when data is written to the display and set when data is read from the display. The instruction set used is:

	CONTROL WORDS									
INSTRUCTION	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
Display Clear	0	0	0	0	0	0	0	0	0	1
Cursor Home	0	0	0	0	0	0	0	0	1	*
Entry mode set	0	0	0	0	0	0	0	1	I/D	S
Disp on/off Control	0	0	0	0	0	0	1	D	C	B
Cursor Disp shift	0	0	0	0	0	1	S/C	R/L	*	*
Function set	0	0	0	0	1	DL	1	0	*	*

* Undefined

I/D =1: Increment	I/D =0: Decrement
S =1: Display shift	S =0: Display freeze
D =1: Display on	D =0: Display off
C =1: Cursor on	C =0: Cursor off
B =1: Char. at cursor position blinks	B =0: Char. at cursor position does not blink
S/C =1: Display shift	S/C =0: Cursor move
R/L =1: Right shift	R/L =0: Left shift
DL =1: 8-Bits	DL =0: 4-Bits

5.1.5 The Real-Time Clock.

CS-4 selects the real-time clock. The clock has a memory consisting of 50 general purpose RAM bytes, 10 RAM bytes which normally contain the time, calendar, and alarm data, and four control and status bytes. All 64 bytes can be read from or written to by the processor program except for the following: a) Registers C and D are read only, b) bit 7 of Register A is read only, and c) the high-order bit of the seconds byte is read only.

ADDRESS MAP

	CONTENTS	Address (Hex)
0	SECONDS	8000
1	SECONDS ALARM	8001
2	MINUTES	8002
3	MINUTES ALARM	8003
4	HOURS	8004
5	HOURS ALARM	8005
6	DAY OF WEEK	8006
7	DAY OF MONTH	8007
8	MONTH	8008
9	YEAR	8009
10	REGISTER A	800A

```
11|  REGISTER B    | 800B
12|  REGISTER C    | 800C
13|  REGISTER D    | 800D
-----------------------------
14
 |  50 Bytes User RAM
63
    -----------------------------
```

The User RAM and Registers A and D are not used. The Control Words for register B are:

9A-Hex	Code to stop updating of RT-Clock
1A-Hex	Code to start updating of RT-Clock

The register bits are set as follows:

LSB 0: Set to 0- Disable daylight saving.
    1: Set to 1- 24-Hour mode.
    2: Set to 0- Specifies binary-coded-decimal.
    3: Set to 1- Not used.
    4: Set to 1- Enables the update-end flag bit in Reg.C.
    5: Set to 0- Not used.
    6: Set to 0- Disable the periodic interrupt.
MSB 7: Set to 1- Stop updating.
          0- Start updating.

Register C: This register is read only to determine whether the update cycle is completed or not. This is indicated by bit 7 and 4 being set. All other bits are unused.

5.1.6 The Hex Keyboard Encoder.

CS-5 selects the Encoder. The Encoder Address is 800A-Hex. No Control Words are required to initialise the Encoder. When a key is pressed, an interrupt is sent to the processor to initiate the reading of the keypad.

5.1.7 The RS-232 Serial Port.

No addressing is required. The serial port is set up as follows:

```
MOV TH1,#0F7H This sets the timer up for a baud rate of 1200
MOV TMOD,#20H Control word to specify 8-bit auto-reload
              timer
MOV SCON,#52H Control word to set up the UART register
SETB TR1      To start the timer
```

Information is sent to the serial port as follows:

```
MOV SBUF,A  Move accumulator contents to serial port
JNB TI,$    Wait until byte has been sent
```

5.2 Program Listings.

The main program's listing is shown from page 26 through 44.

The GWBASIC listing (CAPTURE) on page 45 shows how the captured data from the serial port is converted to battery voltages. Refer to section 7.4 for use of this program.

5.3 Flowcharts.

The main program's operation is illustrated by the detailed flowcharts from page 46 through 51.

The flowchart on page 52 shows how the battery voltage is relayed to a PC in character format, via the serial port at five minute intervals. Operation of the serial port is described in section 7.4.

```
LOC  OBJ            LINE     SOURCE

0025                   1     KEYSTORE1  DATA   25H
0026                   2     KEYSTORE2  DATA   26H
0027                   3     KEYSTORE3  DATA   27H
0028                   4     KEYSTORE4  DATA   28H
0029                   5     HRS        DATA   29H
002A                   6     MIN        DATA   2AH
002C                   7     CYCLE1     DATA   2CH
002D                   8     CYCLE2     DATA   2DH
002E                   9     CYCLE3     DATA   2EH
002F                  10     CYCLE4     DATA   2FH
0030                  11     CYCLE5     DATA   30H
0031                  12     CYCLE6     DATA   31H
0032                  13     CYCLE7     DATA   32H
0033                  14     CYCLE8     DATA   33H
0034                  15     ALM1SEC    DATA   34H
0035                  16     ALM1MIN    DATA   35H
0036                  17     ALM1HR     DATA   36H
0037                  18     ALM2SEC    DATA   37H
0038                  19     ALM2MIN    DATA   38H
0039                  20     ALM2HR     DATA   39H
003A                  21     ALM3SEC    DATA   3AH
003B                  22     ALM3MIN    DATA   3BH
003C                  23     ALM3HR     DATA   3CH
003D                  24     ALM4SEC    DATA   3DH
003E                  25     ALM4MIN    DATA   3EH
003F                  26     ALM4HR     DATA   3FH
0040                  27     ALM5SEC    DATA   40H
0041                  28     ALM5MIN    DATA   41H
0042                  29     ALM5HR     DATA   42H
0043                  30     ALM6SEC    DATA   43H
0044                  31     ALM6MIN    DATA   44H
0045                  32     ALM6HR     DATA   45H
0046                  33     ALM7SEC    DATA   46H
0047                  34     ALM7MIN    DATA   47H
0048                  35     ALM7HR     DATA   48H
0049                  36     ALM8SEC    DATA   49H
004A                  37     ALM8MIN    DATA   4AH
004B                  38     ALM8HR     DATA   4BH
004C                  39     BATTNO     DATA   4CH
004D                  40     BSTATUSO   DATA   4DH
004E                  41     BSTATUSN   DATA   4EH
004F                  42     CHARSTATUS DATA   4FH
0050                  43     DISCSTATUS DATA   50H
0051                  44     COUNT      DATA   51H
0052                  45     BFLAG1     DATA   52H
0053                  46     BFLAG2     DATA   53H
0054                  47     BFLAG3     DATA   54H
0055                  48     BFLAG4     DATA   55H
0056                  49     BFLAG5     DATA   56H
0057                  50     BFLAG6     DATA   57H
0058                  51     BFLAG7     DATA   58H
0059                  52     BFLAG8     DATA   59H
005A                  53     TEMP       DATA   5AH
```

```
LOC  OBJ            LINE     SOURCE

  005B                54     TEMPBSO    DATA    5BH
  005C                55     TEMPCHS    DATA    5CH
  005D                56     FLAG       DATA    5DH
  005E                57     RETDEL     DATA    5EH
  005F                58     SCANC      DATA    5FH
  0093                59     FINISH     BIT     P1.3
  0094                60     RS         BIT     P1.4
  0095                61     WRITE      BIT     P1.5
0000                  62     ORG        0000H
0000 020016           63     LJMP       BEGIN          ;GOTO BEGIN WITH POWER ON
0013                  64     ORG        0013H          ;INT1 VECTOR ADDRESS
0013 02055D           65     LJMP       INT_SCAN       ;WITH INTERRUPT: SCAN KEYBOARD + STORE CHAR
                      66
                      67     BEGIN:
                      68
0016 C2AF             69     CLR    EA                 ;DISABLE ALL INTERRUPTS
0018 C295             70     CLR    WRITE              ;SET DISP WR TO LOW
001A D2B3             71     SETB   P3.3               ;SET P3.3 HIGH TO PREPARE AS INTERRUPT
001C D2AA             72     SETB   EX1                ;ENABLE EXTERNAL INTERRUPT 1
001E 755D00           73     MOV    FLAG,#00H          ;CLEAR FLAG
0021 754D00           74     MOV    BSTATUSO,#00H      ;CLEAR BSTATUSO
0024 754E00           75     MOV    BSTATUSN,#00H      ;CLEAR BSTATUSN
0027 754F00           76     MOV    CHARSTATUS,#00H    ;CLEAR CHARSTATUS
002A 755000           77     MOV    DISCSTATUS,#00H    ;CLEAR DISCSTATUS
002D 120401           78     LCALL  SETUART            ;SETUP FOR SERIAL PORT
0030 12040D           79     LCALL  CWDISP             ;SEND CONTR WORDS TO DISPLAY
0033 120433           80     LCALL  CWRTC              ;SEND CONTR WORDS TO RT-CLOCK
0036 12043F           81     LCALL  CW8255             ;SEND CONTR WORDS TO 8255
0039 90058C           82     MOV    DPTR,#HOURS        ;SET POINT TO HOURS MESSAGE
003C 12047F           83     LCALL  MESSAGE            ;DISP HOURS MESSAGE
003F 20B3FD           84     JB     P3.3,$             ;JMP UNTILL KEY IS PRESSED
0042 120575           85     LCALL  SCAN               ;ENTER 1ST DIGIT
0045 F525             86     MOV    KEYSTORE1,A        ;STORE 1ST DIGIT
0047 1204D5           87     LCALL  CURSH
004A 1204E3           88     LCALL  CLEARDISP
004D 120500           89     LCALL  LONGDELAY
0050 E525             90     MOV    A,KEYSTORE1
0052 1204C8           91     LCALL  OPDISP             ;DISP 1ST DIGIT
0055 120500           92     LCALL  LONGDELAY
0058 20B3FD           93     JB     P3.3,$             ;JMP UNTILL KEY IS PRESSED
005B 120575           94     LCALL  SCAN               ;ENTER 2ND DIGIT
005E F526             95     MOV    KEYSTORE2,A        ;STORE 2ND DIGIT
0060 1204C8           96     LCALL  OPDISP             ;DISP 2ND DIGIT
0063 120500           97     LCALL  LONGDELAY
0066 1204D5           98     LCALL  CURSH
0069 1204E3           99     LCALL  CLEARDISP
006C 120500          100     LCALL  LONGDELAY
006F 90059D          101     MOV    DPTR,#MINUTES      ;SET POINT TO MINUTES MESSAGE
0072 12047F          102     LCALL  MESSAGE            ;DISP MINUTES MESSAGE
0075 20B3FD          103     JB     P3.3,$             ;JMP UNTILL KEY IS PRESSED
0078 120575          104     LCALL  SCAN               ;ENTER 1ST DIGIT
007B F527            105     MOV    KEYSTORE3,A        ;STORE 1ST DIGIT
007D 1204D5          106     LCALL  CURSH
```

```
LOC  OBJ            LINE     SOURCE

0080 1204E3           107     LCALL   CLEARDISP
0083 120500           108     LCALL   LONGDELAY
0086 E527             109     MOV     A,KEYSTORE3
0088 1204C8           110     LCALL   OPDISP          ;DISP 1ST DIGIT
008B 120500           111     LCALL   LONGDELAY
008E 20B3FD           112     JB      P3.3,$          ;JMP UNTILL KEY IS PRESSED
0091 120575           113     LCALL   SCAN            ;ENTER 2ND DIGIT
0094 F528             114     MOV     KEYSTORE4,A     ;STORE 2ND DIGIT
0096 1204C8           115     LCALL   OPDISP          ;DISP  2ND DIGIT
0099 120500           116     LCALL   LONGDELAY
009C 120509           117     LCALL   COMBHRS
009F 120519           118     LCALL   COMBMIN
00A2 120529           119     LCALL   SETHRS          ;SET HRS OF RT-CLOCK
00A5 120533           120     LCALL   SETMIN          ;SET MIN OF RT-CLOCK
00A8 120541           121     LCALL   SETSEC          ;SET SEC OF RT-CLOCK
00AB 12044F           122     LCALL   CWRTSU
00AE 7B18             123     MOV     R3,#18H         ;R3 = COUNT = 24
00B0 7834             124     MOV     R0,#ALM1SEC     ;SET R0 TO START OF ALARM ADDRESSES
00B2 74AA             125     MOV     A,#0AAH         ;LOAD AAH INTO ACC
                      126     ALM_AA:
00B4 F6               127     MOV     @R0,A           ;SET CONTENTS OF ALARM ADDRESS TO AAH
00B5 08               128     INC     R0              ;INC R0 TO NEXT ALARM ADDRESS
00B6 DBFC             129     DJNZ    R3,ALM_AA       ;REPEAT UNTIL R3 = 0
                      130
00B8 7B08             131     MOV     R3,#08H         ;R3 = COUNT = 8
00BA 782C             132     MOV     R0,#CYCLE1      ;SET R0 TO START OF CYCLE ADDRESSES
00BC 7400             133     MOV     A,#00H          ;LOAD 00H INTO ACC
                      134     CLR_CYCLE:
00BE F6               135     MOV     @R0,A           ;SET CONTENTS OF CYCLE ADDRESS TO 00H
00BF 08               136     INC     R0              ;INC R0 TO NEXT CYCLE ADDRESS
00C0 DBFC             137     DJNZ    R3,CLR_CYCLE    ;REPEAT UNTIL R3 = 0
                      138
00C2 7B08             139     MOV     R3,#08H         ;R3 = COUNT = 8
00C4 7852             140     MOV     R0,#BFLAG1      ;SET R0 TO START OF BFLAG ADDRESSES
00C6 7400             141     MOV     A,#00H          ;LOAD 00H INTO ACC
                      142     CLR_FLAG:
00C8 F6               143     MOV     @R0,A           ;SET CONTENTS OF BFLAG ADDRESS TO 00H
00C9 08               144     INC     R0              ;INC R0 TO NEXT BFLAG ADDRESS
00CA DBFC             145     DJNZ    R3,CLR_FLAG     ;REPEAT UNTIL R3 = 0
                      146
                      147     UPDATE1:
                      148
00CC 752800           149     MOV     KEYSTORE4,#00H  ;CLEAR KEYSTORE4
00CF 755E1F           150     MOV     RETDEL,#1FH     ;SET COUNT TO 1FH
                      151
                      152     UPDATE:
                      153
00D2 C3               154     CLR     C               ;CLEAR CARRY FLAG
00D3 D2AF             155     SETB    EA              ;ENABLE ALL INTERRUPTS
00D5 E55D             156     MOV     A,FLAG          ;MOVE FLAG INTO ACC
00D7 94BB             157     SUBB    A,#0BBH         ;FLAG - BBH
00D9 6003             158     JZ      CONT            ;IF ACC = 0 GOTO CONT
00DB 120456           159     LCALL   CWRTFU          ;UPDATE TIME
```

```
LOC  OBJ            LINE     SOURCE

                     160     CONT:
00DE E528            161     MOV    A,KEYSTORE4       ;IF INTERRUPT: KEYSTORE4 = FFH
00E0 B4FF1C          162     CJNE   A,#0FFH,SKIP5     ;IF NO KEY PRESSED: GOTO SKIP5
00E3 E527            163     MOV    A,KEYSTORE3       ;MOVE CHAR INTO ACC
00E5 B40B03          164     CJNE   A,#0BH,NEXTKEY    ;IF CHAR <> 'B' GOTO NEXTKEY
00E8 020112          165     LJMP   B_PROCESS         ;ELSE GOTO B_PROCESS
                     166
                     167     NEXTKEY:
                     168
00EB B40FDE          169     CJNE   A,#0FH,UPDATE1    ;IF CHAR <> 'F' GOTO UPDATE1
00EE 1204E3          170     LCALL  CLEARDISP         ;CLEAR THE DISPLAY
00F1 120500          171     LCALL  LONGDELAY
00F4 E54D            172     MOV    A,BSTATUS0        ;MOVE BSTATUS0 INTO ACC
00F6 B40003          173     CJNE   A,#00H,SKIP1      ;CHECK IF ANY BATTERIES WERE ENTERED
00F9 0200CC          174     LJMP   UPDATE1           ;IF NOT - GOTO UPDATE1
                     175     SKIP1:
00FC 0201E2          176     LJMP   F_PROCESS1        ;ELSE GOTO F_PROCESS1
                     177     SKIP5:
00FF E55D            178     MOV    A,FLAG
0101 B4BBCE          179     CJNE   A,#0BBH,UPDATE    ;IF FLAG <> BB: GOTO UPDATE
0104 120500          180     LCALL  LONGDELAY
0107 C3              181     CLR    C
0108 E55E            182     MOV    A,RETDEL
010A 14              183     DEC    A
010B F55E            184     MOV    RETDEL,A
010D 70C3            185     JNZ    UPDATE            ;IF RETDEL <> 0: GOTO UPDATE
010F 0201E2          186     LJMP   F_PROCESS1        ;ELSE CONTINUE WITH PROCESS
                     187
                     188     B_PROCESS:
                     189
0112 1204D5          190     LCALL  CURSH
0115 1204E3          191     LCALL  CLEARDISP
0118 120500          192     LCALL  LONGDELAY
011B 9005AE          193     MOV    DPTR,#BATMES      ;SET POINT TO BATTERY MESSAGE
011E 12047F          194     LCALL  MESSAGE           ;DISPLAY BATTERY MESSAGE
0121 20B3FD          195     JB     P3.3,$            ;JMP UNTILL KEY IS PRESSED
0124 120575          196     LCALL  SCAN              ;ENTER BATTERY NUMBER
0127 F525            197     MOV    KEYSTORE1,A       ;STORE BATTERY NUMBER
0129 B40103          198     CJNE   A,#01H,ONE        ;CHECK FOR VALID NUMBER
012C 020156          199     LJMP   B_CONT
                     200     ONE:
012F B40203          201     CJNE   A,#02H,TWO        ;CHECK FOR VALID NUMBER
0132 020156          202     LJMP   B_CONT
                     203     TWO:
0135 B40303          204     CJNE   A,#03H,THREE      ;CHECK FOR VALID NUMBER
0138 020156          205     LJMP   B_CONT
                     206     THREE:
013B B40403          207     CJNE   A,#04H,FOUR       ;CHECK FOR VALID NUMBER
013E 020156          208     LJMP   B_CONT
                     209     FOUR:
0141 B40503          210     CJNE   A,#05H,FIVE       ;CHECK FOR VALID NUMBER
0144 020156          211     LJMP   B_CONT
                     212     FIVE:
```

```
LOC  OBJ            LINE     SOURCE

0147 B40603           213     CJNE   A,#06H,SIX        ;CHECK FOR VALID NUMBER
014A 020156           214     LJMP   B_CONT
                      215     SIX:
014D B40703           216     CJNE   A,#07H,SEVEN      ;CHECK FOR VALID NUMBER
0150 020156           217     LJMP   B_CONT
                      218     SEVEN:
0153 B408BC           219     CJNE   A,#08H,B_PROCESS  ;CHECK FOR VALID NUMBER
                      220
                      221     B_CONT:
                      222
0156 9005EF           223     MOV    DPTR,#DETBATT     ;SET POINT TO START OF DETBATT'S TABLE
0159 93               224     MOVC   A,@A+DPTR         ;DETERMINE THE BATTERY CODE
015A F54C             225     MOV    BATTNO,A          ;STORE BATTERY CODE
015C E54D             226     MOV    A,BSTATUSO        ;MOVE EXISTING BATTERY CODE INTO ACC
015E 454C             227     ORL    A,BATTNO          ;LOGICAL ORR BATTNO WITH BSTATUSO
0160 F54D             228     MOV    BSTATUSO,A        ;SET BIT FOR THAT BATTERY
0162 1204D5           229     LCALL  CURSH
0165 1204E3           230     LCALL  CLEARDISP
0168 120500           231     LCALL  LONGDELAY
016B E525             232     MOV    A,KEYSTORE1       ;MOVE BATTERY NUMBER INTO ACC
016D 1204C8           233     LCALL  OPDISP            ;DISP BATTERY NUMBER
0170 120500           234     LCALL  LONGDELAY
                      235
                      236     C_PROCESS:
                      237
0173 1204D5           238     LCALL  CURSH
0176 1204E3           239     LCALL  CLEARDISP
0179 120500           240     LCALL  LONGDELAY
017C 9005BF           241     MOV    DPTR,#CYCLEMES    ;SET POINT TO CYCLE MESSAGE
017F 12047F           242     LCALL  MESSAGE           ;DISPLAY CYCLE MESSAGE
0182 20B3FD           243     JB     P3.3,$            ;JMP UNTILL KEY IS PRESSED
0185 120575           244     LCALL  SCAN              ;ENTER NO OF CYCLES
0188 F526             245     MOV    KEYSTORE2,A       ;STORE NO OF CYCLES
018A B40103           246     CJNE   A,#01H,ONE1       ;CHECK FOR VALID NUMBER
018D 0201B7           247     LJMP   C_CONT
                      248     ONE1:
0190 B40203           249     CJNE   A,#02H,TWO2       ;CHECK FOR VALID NUMBER
0193 0201B7           250     LJMP   C_CONT
                      251     TWO2:
0196 B40303           252     CJNE   A,#03H,THREE3     ;CHECK FOR VALID NUMBER
0199 0201B7           253     LJMP   C_CONT
                      254     THREE3:
019C B40403           255     CJNE   A,#04H,FOUR4      ;CHECK FOR VALID NUMBER
019F 0201B7           256     LJMP   C_CONT
                      257     FOUR4:
01A2 B40503           258     CJNE   A,#05H,FIVE5      ;CHECK FOR VALID NUMBER
01A5 0201B7           259     LJMP   C_CONT
                      260     FIVE5:
01A8 B40603           261     CJNE   A,#06H,SIX6       ;CHECK FOR VALID NUMBER
01AB 0201B7           262     LJMP   C_CONT
                      263     SIX6:
01AE B40703           264     CJNE   A,#07H,SEVEN7     ;CHECK FOR VALID NUMBER
01B1 0201B7           265     LJMP   C_CONT
```

```
LOC  OBJ            LINE     SOURCE

                     266     SEVEN7:
01B4 B4088C          267     CJNE    A,#08H,C_PROCESS  ;CHECK FOR VALID NUMBER
                     268
                     269     C_CONT:
                     270
01B7 1204D5          271     LCALL   CURSH
01BA 1204E3          272     LCALL   CLEARDISP
01BD 120500          273     LCALL   LONGDELAY
01C0 E526            274     MOV     A,KEYSTORE2       ;MOVE NUMBER OF CYCLES INTO ACC
01C2 1204C8          275     LCALL   QPDISP            ;DISPLAY NO OF CYCLES
01C5 120500          276     LCALL   LONGDELAY
01C8 782C            277     MOV     R0,#CYCLE1        ;MOVE ADDRESS OF CYCLE1 INTO R0
01CA E525            278     MOV     A,KEYSTORE1       ;MOVE BATTERY NUMBER INTO ACC
01CC 14              279     DEC     A                 ;DEC ACC
01CD 28              280     ADD     A,R0              ;DETERMINE CYCLE ADDRESS FOR THAT BATTERY
01CE F8              281     MOV     R0,A              ;LOAD CYCLE ADDRESS INTO R0
01CF E526            282     MOV     A,KEYSTORE2       ;MOVE NUMBER OF CYCLES INTO ACC
01D1 F6              283     MOV     @R0,A             ;STORE NO OF CYCLES IN CYCLE ADDRESS
01D2 C3              284     CLR     C                 ;CLEAR CARRY FLAG
01D3 E55D            285     MOV     A,FLAG            ;MOVE FLAG INTO ACC
01D5 94BB            286     SUBB    A,#0BBH           ;FLAG - BBH
01D7 7006            287     JNZ     SKIP4             ;IF FLAG <> BBH GOTO SKIP4
01D9 120495          288     LCALL   DISP_MES          ;ELSE DISP ADD BATTERY/FINISH MESSAGE
01DC 0200CC          289     LJMP    UPDATE1
                     290
                     291     SKIP4:
01DF 0200CC          292     LJMP    UPDATE1
                     293
                     294     F_PROCESS1:
                     295
01E2 90800B          296     MOV     DPTR,#800BH       ;POINT TO REG B OF RT-CLOCK
01E5 741A            297     MOV     A,#1AH            ;MOVE START CODE OF RT-CLOCK INTO ACC
01E7 F0              298     MOVX    @DPTR,A           ;START RT-CLOCK
01E8 752800          299     MOV     KEYSTORE4,#00H    ;CLEAR KEYSTORE4
01EB 755D00          300     MOV     FLAG,#00H         ;CLEAR FLAG
                     301
                     302     F_PROCESS:
                     303
01EE D2AF            304     SETB    EA                ;ENABLE ALL INTERRUPTS
01F0 7551F8          305     MOV     COUNT,#0F8H       ;SET ADDRESS COUNT TO 1111 1000
01F3 AC40            306     MOV     R4,BSTATUS0       ;LOAD BATT CODE INTO R4
01F5 E528            307     MOV     A,KEYSTORE4       ;MOVE FLAG INTO ACC
01F7 B4FF14          308     CJNE    A,#0FFH,TOP1      ;CHECK IF KEY WAS PRESSED
01FA E527            309     MOV     A,KEYSTORE3       ;MOVE CHAR INTO ACC
01FC B4080F          310     CJNE    A,#08H,TOP1       ;IF CHAR <> 'B' GOTO TOP1
01FF 90800B          311     MOV     DPTR,#800BH       ;SET POINT TO REG B OF RT-CLOCK
0202 749A            312     MOV     A,#9AH            ;MOVE STOP CODE OF RT-CLOCK INTO ACC
0204 F0              313     MOVX    @DPTR,A           ;STOP RT-CLOCK
0205 755DBB          314     MOV     FLAG,#0BBH        ;SET FLAG TO BBH
0208 120495          315     LCALL   DISP_MES
020B 0200CC          316     LJMP    UPDATE1
                     317
                     318     TOP1:
```

```
LOC  OBJ            LINE     SOURCE

                     319
020E C3              320      CLR     C                ;CLEAR CARRY FLAG
020F EC              321      MOV     A,R4             ;RESTORE BATT CODE TO ACC
0210 13              322      RRC     A                ;ROTATE THROUGH CARRY TO CHECH FOR BATTERIES
0211 FC              323      MOV     R4,A             ;STORE BATT CODE IN R4
0212 4072            324      JC      ADC              ;IF BATT EXIST - GOTO ADC
                     325
                     326      TOP2:
                     327
0214 C295            328      CLR     WRITE
0216 1204D5          329      LCALL   CURSH
0219 12027B          330      LCALL   LF
021C 90057C          331      MOV     DPTR,#MENU       ;SET POINT TO MENU MESSAGE
021F 12047F          332      LCALL   MESSAGE          ;DISPLAY MENU MESSAGE
0222 0551            333      INC     COUNT            ;INC ADDRESS COUNT
0224 7400            334      MOV     A,#00H           ;CLEAR ACC
0226 B551E5          335      CJNE    A,COUNT,TOP1     ;IF COUNT <> 0 GOTO TOP1
0229 1204D5          336      LCALL   CURSH
022C 854F5C          337      MOV     TEMPCHS,CHARSTATUS
022F 854D5B          338      MOV     TEMPBSO,BSTATUSO
0232 755108          339      MOV     COUNT,#08H       ;SET COUNT = 8
0235 906000          340      MOV     DPTR,#6000H      ;SET POINT TO DISPLAY
                     341
                     342      STATUSDISP:
                     343
0238 C3              344      CLR     C                ;CLEAR CARRY FLAG
0239 E55B            345      MOV     A,TEMPBSO        ;MOV TEMPBSO INTO ACC
023B 13              346      RRC     A                ;ROTATE TEMPBSO THROUGH CARRY
023C F55B            347      MOV     TEMPBSO,A        ;STORE RESULT BACK IN TEMPBSO
023E 5016            348      JNC     DISP_F           ;IF CARRY = 0 GOTO DISP_F
0240 C3              349      CLR     C                ;CLEAR CARRY FLAG
0241 E55C            350      MOV     A,TEMPCHS        ;MOVE TEMPCHS INTO ACC
0243 13              351      RRC     A                ;ROTATE TEMPCHS THROUGH CARRY
0244 F55C            352      MOV     TEMPCHS,A        ;STORE RESULT BACK IN TEMPCHS
0246 5021            353      JNC     DISP_D           ;IF CARRY = 0 GOTO DISP_D
0248 7443            354      MOV     A,#43H           ;MOVE CHAR 'C' INTO ACC
024A F0              355      MOVX    @DPTR,A          ;SEND 'C' TO DISPLAY
024B 1204FB          356      LCALL   DELAY
024E 7420            357      MOV     A,#20H           ;MOVE 'SPACE' CHAR INTO ACC
0250 F0              358      MOVX    @DPTR,A          ;SEND 'SPACE' CHAR TO DISPLAY
0251 1204FB          359      LCALL   DELAY
0254 801F            360      SJMP    DECR             ;JUMP TO DECR
                     361
                     362      DISP_F:
                     363
0256 E55C            364      MOV     A,TEMPCHS        ;MOVE TEMPCHS INTO ACC
0258 03              365      RR      A                ;ROTATE TEMPCHS RIGHT
0259 F55C            366      MOV     TEMPCHS,A        ;STORE RESULT BACK IN TEMPCHS
025B 7446            367      MOV     A,#46H           ;MOVE CHAR 'F' INTO ACC
025D F0              368      MOVX    @DPTR,A          ;SEND 'F' TO DISPLAY
025E 1204FB          369      LCALL   DELAY
0261 7420            370      MOV     A,#20H           ;MOVE 'SPACE' CHAR INTO ACC
0263 F0              371      MOVX    @DPTR,A          ;SEND 'SPACE' CHAR TO DISPLAY
```

```
LOC  OBJ            LINE     SOURCE

0264 1204FB          372      LCALL  DELAY
0267 800C            373      SJMP   DECR              ;JUMP TO DECR
                     374
                     375      DISP_D:
                     376
0269 7444            377      MOV    A,#44H            ;MOVE 'D' INTO ACC
026B F0              378      MOVX   @DPTR,A           ;SEND 'D' TO DISPLAY
026C 1204FB          379      LCALL  DELAY
026F 7420            380      MOV    A,#20H            ;MOVE 'SPACE' CHAR INTO ACC
0271 F0              381      MOVX   @DPTR,A           ;SEND 'SPACE, CHAR TO DISPLAY
0272 1204FB          382      LCALL  DELAY
                     383
                     384      DECR:
                     385
0275 0551C0          386      DJNZ   COUNT,STATUSDISP  ;IF COUNT <> 0 GOTO STATUSDISP
0278 0201EE          387      LJMP   F_PROCESS         ;ELSE GOTO F_PROCESS
                     388
                     389      LF:
                     390
027B C294            391      CLR    RS                ;TO SEND CONTR WORDS TO DISP
027D 906000          392      MOV    DPTR,#6000H       ;SET POINT TO DISPLAY
0280 74C0            393      MOV    A,#0C0H           ;'LINE FEED' TO ACC
0282 F0              394      MOVX   @DPTR,A           ;'LINE FEED' TO DISP (CONTR WORD)
0283 D294            395      SETB   RS                ;TO WR/RD DATA TO/FROM DISP
0285 22              396      RET
                     397
                     398      ADC:
                     399
0286 D293            400      SETB   FINISH            ;PREPARE P1.3 FOR INPUT
0288 9005F8          401      MOV    DPTR,#C_POINT     ;SET POINT TO BEGINNING OF ALARM ADDRESSES
028B E551            402      MOV    A,COUNT           ;LOAD ADDRESS COUNT INTO ACC
028D 5407            403      ANL    A,#07H            ;MASK HIGHER-5 BITS TO GET BATT-NO
028F 04              404      INC    A                 ;ADDRESS 000 = BATT-NO 1
0290 F54C            405      MOV    BATTNO,A          ;STORE BATT-NO IN BATTNO
0292 93              406      MOVC   A,@A+DPTR         ;GET CORRECT ALARM ADDRESS FOR THAT BATTERY
0293 F9              407      MOV    R1,A              ;STORE ALARM ADDRESS IN R1
0294 9005EF          408      MOV    DPTR,#DETBATT     ;SET POINT TO START OF DETBATT
0297 E54C            409      MOV    A,BATTNO          ;MOVE BATTNO INTO ACC
0299 93              410      MOVC   A,@A+DPTR         ;GET CORRECT BATTERY CODE
029A F54E            411      MOV    BSTATUSN,A        ;STORE CODE IN BSTATUSN
029C 855190          412      MOV    P1,COUNT          ;SEND ADDRESS TO CD4051B (MULTIPLEXER)
029F 1204FB          413      LCALL  DELAY             ;DELAY TO STABALISE A-D VOLTAGE
02A2 904000          414      MOV    DPTR,#4000H       ;POINT TO ADC
02A5 F0              415      MOVX   @DPTR,A           ;ACTIVATE WR LINE TO START CONVERSION OF ADC
02A6 2093FD          416      JB     FINISH,$          ;FINISH = 1 DURING CONVERSION
02A9 C3              417      CLR    C                 ;CLEAR CARRY FLAG
02AA E0              418      MOVX   A,@DPTR           ;ADC VALUE INTO ACC
02AB C082            419      PUSH   DPL
02AD C083            420      PUSH   DPH
02AF C0E0            421      PUSH   ACC
02B1 908002          422      MOV    DPTR,#8002H       ;SET POINTER TO RTC-MINUTES ADDRESS
02B4 E0              423      MOVX   A,@DPTR           ;READ MINUTES INTO ACC
02B5 B55F0F          424      CJNE   A,SCANC,RETURN    ;IF 4 MIN HAVE PASSED CONTINUE ELSE JUMP
```

```
LOC  OBJ            LINE     SOURCE

02B8 E55F            425      MOV    A,SCANC          ;ACC = SCANC
02BA 3405            426      ADDC   A,#05            ;SCANC = SCANC + 5
02BC D4              427      DA     A                ;CONVERT SCANC TO A BCD VALUE
02BD B46002          428      CJNE   A,#60H,RSCAN     ;IF SCANC = 60 CONT ELSE JUMP
02C0 7400            429      MOV    A,#00            ;SET ACC = 00
                     430
                     431      RSCAN:
02C2 F55F            432      MOV    SCANC,A
02C4 1203ED          433      LCALL  TX               ;SEND ADC-VALUE TO SERIAL PORT
                     434
                     435      RETURN:
02C7 D0E0            436      POP    ACC
02C9 D083            437      POP    DPH
02CB D082            438      POP    DPL
02CD 94B7            439      SUBB   A,#0B7H          ;ADC VALUE - B7H (3.61 VOLT)
02CF 4003            440      JC     RTC1_ALM         ;IF ADC < 3.61 VOLT GOTO RTC1_ALM
02D1 02030B          441      LJMP   RTC2_ALM         ;IF ADC > 3.61 VOLT GOTO RTC2_ALM
                     442
                     443      RTC1_ALM:
                     444
02D4 E9              445      MOV    A,R1             ;LOAD ALM-SEC ADDRESS INTO ACC
02D5 F8              446      MOV    R0,A             ;STORE ADDRESS IN R0
02D6 908000          447      MOV    DPTR,#8000H      ;POINT TO SEC ADDRESS OF RTC
02D9 E0              448      MOVX   A,@DPTR          ;LOAD RTC-SEC INTO ACC
02DA 12054B          449      LCALL  BCD_HEX          ;TO CONVERT FROM BCD TO HEX
02DD F55A            450      MOV    TEMP,A           ;MOVE RTC-SEC INTO TEMP
02DF E6              451      MOV    A,@R0            ;LOAD ALM-SEC INTO ACC
02E0 B55A1F          452      CJNE   A,TEMP,SEC1_AA   ;IF RTC-SEC <> ALM-SEC GOTO SEC1_AA
02E3 08              453      INC    R0               ;INC R0 TO ALM-MIN ADDRESS
02E4 908002          454      MOV    DPTR,#8002H      ;POINT TO MIN ADDRESS OF RTC
02E7 E0              455      MOVX   A,@DPTR          ;LOAD MIN INTO ACC
02E8 12054B          456      LCALL  BCD_HEX          ;TO CONVERT FROM BCD TO HEX
02EB F55A            457      MOV    TEMP,A           ;MOVE MIN INTO TEMP
02ED E6              458      MOV    A,@R0            ;LOAD ALM-MIN INTO ACC
02EE B55A11          459      CJNE   A,TEMP,SEC1_AA   ;IF RTC-MIN <> ALM-MIN GOTO SEC1_AA
02F1 08              460      INC    R0               ;INC R0 TO ALM-HR ADDRESS
02F2 908004          461      MOV    DPTR,#8004H      ;POINT TO HR ADDRESS OF RTC
02F5 E0              462      MOVX   A,@DPTR          ;LOAD HR INTO ACC
02F6 12054B          463      LCALL  BCD_HEX          ;TO CONVERT FROM BCD TO HEX
02F9 F55A            464      MOV    TEMP,A           ;MOVE HR INTO TEMP
02FB E6              465      MOV    A,@R0            ;LOAD ALM-HR INTO ACC
02FC B55A03          466      CJNE   A,TEMP,SEC1_AA   ;IF RTC-HR <> ALM-HR GOTO SEC1_AA
02FF 0203B4          467      LJMP   STP_CHRG         ;STOP CHARGING OF BATTERY
                     468
                     469      SEC1_AA:
0302 B7AA03          470      CJNE   @R1,#0AAH,SKIP3  ;IF SEC <> AAH GOTO SKIP3
0305 020346          471      LJMP   SETALM           ;SET THE ALARM
                     472
                     473      SKIP3:
0308 020214          474      LJMP   TOP2
                     475
                     476      RTC2_ALM:
                     477
```

```
LOC  OBJ            LINE     SOURCE

030B E9              478      MOV     A,R1              ;LOAD ALM-SEC ADDRESS INTO ACC
030C F8              479      MOV     R0,A              ;STORE ADDRESS IN R0
030D 908000          480      MOV     DPTR,#8000H       ;SET POINT TO SEC ADDRESS OF RTC
0310 E0              481      MOVX    A,@DPTR           ;LOAD RTC-SEC INTO ACC
0311 12054B          482      LCALL   BCD_HEX           ;TO CONVERT FROM BCD TO HEX
0314 F55A            483      MOV     TEMP,A            ;MOVE RTC-SEC INTO TEMP
0316 E6              484      MOV     A,@R0             ;LOAD ALM-SEC INTO ACC
0317 B55A1F          485      CJNE    A,TEMP,FLAG_FF    ;IF RTC-SEC <> ALM-SEC GOTO FLAG_FF
031A 08              486      INC     R0                ;INC R0 TO ALM-MIN ADDRESS
031B 908002          487      MOV     DPTR,#8002H       ;SET POINT TO MIN ADDRESS OF RTC
031E E0              488      MOVX    A,@DPTR           ;LOAD MIN INTO ACC
031F 12054B          489      LCALL   BCD_HEX           ;TO CONVERT FROM BCD TO HEX
0322 F55A            490      MOV     TEMP,A            ;MOVE MIN INTO TEMP
0324 E6              491      MOV     A,@R0             ;LOAD ALM-MIN INTO ACC
0325 B55A11          492      CJNE    A,TEMP,FLAG_FF    ;IF RTC-MIN <> ALM-MIN GOTO FLAG_FF
0328 08              493      INC     R0                ;INC R0 TO ALM-HR ADDRESS
0329 908004          494      MOV     DPTR,#8004H       ;SET POINT TO HR ADDRESS OF RTC
032C E0              495      MOVX    A,@DPTR           ;LOAD HR INTO ACC
032D 12054B          496      LCALL   BCD_HEX           ;TO CONVERT FROM BCD TO HEX
0330 F55A            497      MOV     TEMP,A            ;MOVE HR INTO TEMP
0332 E6              498      MOV     A,@R0             ;LOAD ALM-HR INTO ACC
0333 B55A03          499      CJNE    A,TEMP,FLAG_FF    ;IF RTC-HR <> ALM-HR GOTO FLAG_FF
0336 0203B4          500      LJMP    STP_CHRG          ;STOP CHARGING OF BATTERY
                     501
                     502      FLAG_FF:
                     503
0339 7852            504      MOV     R0,#BFLAG1        ;R0 = POINT TO BFLAG
033B 18              505      DEC     R0                ;DET CORRECT POSITION FOR POINTER
033C E8              506      MOV     A,R0              ;LOAD START POSITION INTO ACC
033D 254C            507      ADD     A,BATTNO          ;DET CORRECT BFLAG ADDRESS
033F F8              508      MOV     R0,A              ;STORE ADDRESS IN R0
0340 B6FF64          509      CJNE    @R0,#0FFH,DISCHARGE ;IF SEC <> FFH GOTO TOP2
0343 020214          510      LJMP    TOP2              ;SET THE ALARM
                     511
                     512      SETALM:
                     513
                     514      SETS:
0346 90800C          515      MOV     DPTR,#800CH       ;SET POINT TO REG C OF RT-CLOCK
0349 E0              516      MOVX    A,@DPTR           ;READ REG C FOR END OF UPDATE
034A 30E7F9          517      JNB     ACC.7,SETS        ;CONTINUE IF BIT 7 IS SET
034D 30E4F6          518      JNB     ACC.4,SETS        ;CONTINUE IF BIT 4 IS SET
0350 908000          519      MOV     DPTR,#8000H       ;SET POINT TO SEC ADDRESS OF RTC
0353 E0              520      MOVX    A,@DPTR           ;MOVE SEC TO ACC
0354 12054B          521      LCALL   BCD_HEX           ;TO CONVERT FROM BCD TO HEX
0357 F7              522      MOV     @R1,A             ;STORE SEC IN ALM*SEC
                     523      SETM:
0358 90800C          524      MOV     DPTR,#800CH       ;SET POINT TO REG C OF RT-CLOCK
035B E0              525      MOVX    A,@DPTR           ;READ REG C FOR END OF UPDATE
035C 30E7F9          526      JNB     ACC.7,SETM        ;CONTINUE IF BIT 7 IS SET
035F 30E4F6          527      JNB     ACC.4,SETM        ;CONTINUE IF BIT 4 IS SET
0362 908002          528      MOV     DPTR,#8002H       ;SET POINT TO MIN ADDRESS OF RTC
0365 E0              529      MOVX    A,@DPTR           ;MOVE MIN TO ACC
0366 12054B          530      LCALL   BCD_HEX           ;TO CONVERT FROM BCD TO HEX
```

```
LOC  OBJ            LINE     SOURCE

0369 09              531      INC     R1              ;CHANGE ADDRESS TO THAT FOR MINUTES
036A F7              532      MOV     @R1,A           ;STORE MIN IN ALM*MIN
                     533      SETH:
036B 90800C          534      MOV     DPTR,#800CH     ;SET POINT TO REG C OF RT-CLOCK
036E E0              535      MOVX    A,@DPTR         ;READ REG C FOR END OF UPDATE
036F 30E7F9          536      JNB     ACC.7,SETH      ;CONTINUE IF BIT 7 IS SET
0372 30E4F6          537      JNB     ACC.4,SETH      ;CONTINUE IF BIT 4 IS SET
0375 908004          538      MOV     DPTR,#8004H     ;SET POINT TO HRS ADDRESS OF RTC
0378 E0              539      MOVX    A,@DPTR         ;MOVE HRS TO ACC
0379 12054B          540      LCALL   BCD_HEX         ;TO DO BCD TO HEX CONVERSION
037C 240E            541      ADD     A,#0EH          ;ADD 14 HRS CHARGING TIME TO ALARM TIME
037E 09              542      INC     R1              ;CHANGE ADDRESS TO THAT FOR HOURS
037F F7              543      MOV     @R1,A           ;STORE HRS IN ALM*HR
0380 C3              544      CLR     C               ;CLEAR CARRY FLAG
0381 9418            545      SUBB    A,#18H          ;IF ALARM HRS < 24 SET CARRY FLAG
0383 4001            546      JC      SET_FLAG        ;IF CARRY GOTO SET_FLAG
0385 F7              547      MOV     @R1,A           ;LOAD CORRECT HOURS INTO ALM*HR
                     548
                     549      SET_FLAG:
                     550
0386 7852            551      MOV     R0,#BFLAG1      ;R0 = POINT TO BFLAG
0388 18              552      DEC     R0              ;DET CORRECT POSITION FOR POINTER
0389 E8              553      MOV     A,R0            ;LOAD START POSITION INTO ACC
038A 254C            554      ADD     A,BATTNO        ;DET CORRECT BFLAG ADDRESS
038C F8              555      MOV     R0,A            ;STORE ADDRESS IN R0
038D 76FF            556      MOV     @R0,#0FFH       ;SET BFLAG ADDRESS CONTENTS TO FFH
                     557
                     558      STP_DISCH:
                     559
038F 902001          560      MOV     DPTR,#2001H     ;SET POINT TO PORT B OF 8255
0392 E54E            561      MOV     A,BSTATUSN      ;LOAD BATTERY CODE INTO ACC
0394 F4              562      CPL     A               ;COMPLIMENT ACC- RESET BIT TO STOP DISCHARGE
0395 5550            563      ANL     A,DISCSTATUS    ;DET CODE TO STOP DISCH FOR SPECIFIC BATTERY
0397 F550            564      MOV     DISCSTATUS,A    ;STORE NEW DISCSTATUS CODE
0399 F0              565      MOVX    @DPTR,A         ;SEND CODE TO PORT B OF 8255 TO STOP DISC OF BATT
                     566
                     567      CHARGE:
                     568
039A 902000          569      MOV     DPTR,#2000H     ;SET POINT TO PORT A OF 8255
039D E54E            570      MOV     A,BSTATUSN      ;LOAD CHARGE CODE FOR BATT INTO ACC
039F 454F            571      ORL     A,CHARSTATUS    ;DET NEW CHARGE CODE: DO NOT AFFECT EXIST BATT
03A1 F54F            572      MOV     CHARSTATUS,A    ;STORE NEW CHARGE CODE
03A3 F0              573      MOVX    @DPTR,A         ;SEND CHARGE CODE TO PORT A OF 8255
03A4 020214          574      LJMP    TOP2
                     575
                     576      DISCHARGE:
                     577
03A7 902001          578      MOV     DPTR,#2001H     ;SET POINT TO PORT B OF 8255
03AA E54E            579      MOV     A,BSTATUSN      ;LOAD DISC CODE FOR BATT INTO ACC
03AC 4550            580      ORL     A,DISCSTATUS    ;DET NEW DISC CODE: DO NOT AFFECT EXIST BATT
03AE F550            581      MOV     DISCSTATUS,A    ;STORE NEW DISCHARGE CODE
03B0 F0              582      MOVX    @DPTR,A         ;SEND DISC CODE TO PORT B OF 8255
03B1 020214          583      LJMP    TOP2
```

```
LOC  OBJ            LINE     SOURCE

                     584
                     585     STP_CHRG:
                     586
03B4 902000          587     MOV     DPTR,#2000H       ;SET POINT TO PORT A OF 8255
03B7 E54E            588     MOV     A,BSTATUSN        ;LOAD CHARGE CODE FOR BATT INTO ACC
03B9 F4              589     CPL     A                 ;COMPLIMENT ACC - RESET BIT TO STOP CHARGE
03BA 554F            590     ANL     A,CHARSTATUS      ;DET CODE TO STOP CHARGE FOR SPECIFIC BATT
03BC F54F            591     MOV     CHARSTATUS,A      ;STORE NEW CHARGE CODE
03BE F0              592     MOVX    @DPTR,A           ;SEND CODE TO PORT A OF 8255 TO STOP CHARGE
                     593
                     594     CLR_ALT_FLG:
                     595
03BF 77AA            596     MOV     @R1,#0AAH         ;SET SEC ALARM ADDRESS CONTENTS TO AAH
03C1 7852            597     MOV     R0,#BFLAG1        ;R0 = POINT TO BFLAG1 ADDRESS
03C3 18              598     DEC     R0                ;DET CORRECT POSITION FOR POINTER
03C4 E8              599     MOV     A,R0              ;LOAD START POSITION INTO ACC
03C5 254C            600     ADD     A,BATTNO          ;DET CORRECT BFLAG ADDRESS
03C7 F8              601     MOV     R0,A              ;STORE ADDRESS IN R0
03C8 7600            602     MOV     @R0,#00H          ;CLEAR CONTENTS OF BFLAG ADDRESS
                     603
                     604     DEC_CYCOUNT:
                     605
03CA 742C            606     MOV     A,#CYCLE1         ;LOAD THE ADDRESS OF CYCLE1 INTO ACC
03CC 14              607     DEC     A                 ;DET CORRECT START POSITION
03CD 254C            608     ADD     A,BATTNO          ;DET CORRECT ADDRESS FOR THAT BATTERY
03CF F8              609     MOV     R0,A              ;MOVE ADDRESS TO R0
03D0 E6              610     MOV     A,@R0             ;MOVE NO OF CYCLES INTO ACC
03D1 14              611     DEC     A                 ;DECREMENT CYCLES
03D2 F6              612     MOV     @R0,A             ;STORE NO OF CYCLES IN CYCLE ADDRESS
03D3 6003            613     JZ      CH_STWORD         ;IF ZERO GOTO CH_STWORD
03D5 020214          614     LJMP    TOP2              ;ELSE GOTO TOP2
                     615
                     616     CH_STWORD:
                     617
03D8 E54E            618     MOV     A,BSTATUSN        ;MOVE BSTATUS TO ACC
03DA F4              619     CPL     A                 ;COMPLIMENT ACC- RESET BIT TO STOP CYCLING
03DB 554D            620     ANL     A,BSTATUSO        ;DET NEW BSTATUSO CODE
03DD F54D            621     MOV     BSTATUSO,A        ;STORE NEW CODE IN BSTATUSO
03DF B40008          622     CJNE    A,#00H,SKIP2      ;IF BSTATUSO <> 0 GOTO SKIP
03E2 C295            623     CLR     WRITE             ;SET DISP WRITE LOW
03E4 755DBB          624     MOV     FLAG,#0BBH        ;SET FLAG = BBH
03E7 0200CC          625     LJMP    UPDATE1
                     626
                     627     SKIP2:
03EA 020214          628     LJMP    TOP2
                     629
                     630     TX:
03ED 5390F8          631     ANL     P1,#0F8H
03F0 904000          632     MOV     DPTR,#4000H       ;SET POINTER TO ADC
03F3 F0              633     MOVX    @DPTR,A           ;ACTICATE ADC TO START CONVERSION
03F4 2093FD          634     JB      FINISH,$          ;WAIT UNTIL CONVERSION IS FINISH
03F7 C3              635     CLR     C
03F8 E0              636     MOVX    A,@DPTR           ;READ ADC VALUE INTO ACC
```

```
LOC  OBJ            LINE     SOURCE

03F9 C299             637     CLR     TI              ;CLEAR TRANSMIT INTERRUPT FLAG
03FB F599             638     MOV     SBUF,A          ;MOVE ADC TO SERIAL PORT
03FD 3099FD           639     JNB     TI,$            ;WAIT UNTIL FINISH
0400 22               640     RET
                      641
                      642     SETUART:
                      643
0401 758DF7           644     MOV     TH1,#0F7H       ;1200 BAUD
0404 758920           645     MOV     TMOD,#20H
0407 759852           646     MOV     SCON,#52H       ;SETUP
040A D28E             647     SETB    TR1             ;SERIAL PORT
040C 22               648     RET
                      649
                      650     CWDISP:
                      651
040D 906000           652     MOV     DPTR,#6000H     ;SET POINT TO DISP
0410 C294             653     CLR     RS              ;SET RS LOW TO SEND CONTR- WORDS TO DISP
0412 7401             654     MOV     A,#01H          ;"DISP CLEAR" TO ACC
0414 F0               655     MOVX    @DPTR,A         ;"DISP CLEAR" TO DISP (CONTR WORD)
0415 1204FB           656     LCALL   DELAY
0418 7402             657     MOV     A,#02H          ;"CURSER HOME" TO ACC
041A F0               658     MOVX    @DPTR,A         ;"CURSER HOME" TO DISP (CONTR WORD)
041B 1204FB           659     LCALL   DELAY
041E 7438             660     MOV     A,#38H          ;"8-BITS FUNCTION" TO ACC
0420 F0               661     MOVX    @DPTR,A         ;"8-BITS FUNCTION" TO DISP
0421 1204FB           662     LCALL   DELAY
0424 7406             663     MOV     A,#06H          ;"REG INC + DISP FREEZE" TO ACC
0426 F0               664     MOVX    @DPTR,A         ;"REG INC + DISP FREEZE" TO DISP (CONTR WORD)
0427 1204FB           665     LCALL   DELAY
042A 740C             666     MOV     A,#0CH          ;"DISP-ON CURSER-OFF" TO ACC
042C F0               667     MOVX    @DPTR,A         ;"DISP-ON CURSER-OFF" TO DISP (CONTR WORD)
042D 1204FB           668     LCALL   DELAY
0430 D294             669     SETB    RS              ;TO WR/RD DATA TO/FROM DISP
0432 22               670     RET
                      671
                      672     CWRTC:
                      673
0433 90800B           674     MOV     DPTR,#800BH     ;SET POINTER TO REG B OF RT-CLOCK
0436 749A             675     MOV     A,#9AH          ;10011010 - CONTR WORD
0438 F0               676     MOVX    @DPTR,A         ;MOVE CONTROL WORD TO REG B
0439 1582             677     DEC     DPL             ;SET POINT TO REG A OF RT-CLOCK
043B 740F             678     MOV     A,#0FH          ;LOAD CONTROL WORD INTO ACC 500mS PULSE TO SQW
043D F0               679     MOVX    @DPTR,A         ;MOVE CONTROL WORD TO REG A
043E 22               680     RET
                      681
                      682     CW8255:
                      683
043F 902003           684     MOV     DPTR,#2003H     ;SET POINT TO CONTR- WORD ADDRESS OF 8255
0442 7481             685     MOV     A,#81H          ;CONTR- WORD TO ACC
0444 F0               686     MOVX    @DPTR,A         ;CONTR- WORD TO 8255
0445 902000           687     MOV     DPTR,#2000H     ;SET POINT TO CHARGE PORT OF 8255
0448 E54F             688     MOV     A,CHARSTATUS    ;CLEAR ACC (CHARSTATUS = 0)
044A F0               689     MOVX    @DPTR,A         ;DISABLE ALL CHARGE PORTS
```

```
LOC  OBJ            LINE     SOURCE

044B 0582            690      INC     DPL              ;SET POINT TO DISCHARGE PORT OF 8255
044D F0              691      MOVX    @DPTR,A          ;DISABLE ALL DISCHARGE PORTS
044E 22              692      RET
                     693
                     694      CWRTSU:
                     695
044F 90800B          696      MOV     DPTR,#800BH      ;SET POINT TO REG B OF RT- CLOCK
0452 741A            697      MOV     A,#1AH           ;LOAD CONTROL WORD INTO ACC START UPDATING
0454 F0              698      MOVX    @DPTR,A          ;MOVE CONTROL WORD TO REG B OF RT- CLOCK
0455 22              699      RET
                     700
                     701      CWRTFU:
                     702
0456 90800C          703      MOV     DPTR,#800CH      ;SET POINT TO REG C OF RT- CLOCK
0459 E0              704      MOVX    A,@DPTR          ;READ REG C FOR END OF TIME UPDATE
045A 30E7F9          705      JNB     ACC.7,CWRTFU     ;IF BIT 7 IS SET CONTINUE
045D 30E4F6          706      JNB     ACC.4,CWRTFU     ;IF BIT 4 IS SET CONTINUE
0460 1204D5          707      LCALL   CURSH
0463 908004          708      MOV     DPTR,#8004H      ;SET POINT TO HRS- ADDRESS
0466 E0              709      MOVX    A,@DPTR          ;READ HRS TO ACC
0467 1204B1          710      LCALL   OPTIME           ;SEND HRS TO DISP
046A 1204F1          711      LCALL   COLON
046D 908002          712      MOV     DPTR,#8002H      ;SET POINT TO MIN- ADDRESS
0470 E0              713      MOVX    A,@DPTR          ;READ MIN TO ACC
0471 1204B1          714      LCALL   OPTIME           ;SEND MIN TO DISP
0474 1204F1          715      LCALL   COLON
0477 908000          716      MOV     DPTR,#8000H      ;SET POINT TO SEC- ADDRESS
047A E0              717      MOVX    A,@DPTR          ;READ SEC TO ACC
047B 1204B1          718      LCALL   OPTIME           ;SEND SEC TO DISP
047E 22              719      RET
                     720
                     721      MESSAGE:
                     722
047F E4              723      CLR     A                ;CLEAR ACC
0480 93              724      MOVC    A,@A+DPTR        ;MOVE MESSAGE CHARACTER TO ACC
0481 C082            725      PUSH    DPL              ;STORE POINTER-L POSITION ON STACK
0483 C083            726      PUSH    DPH              ;STORE POINTER-H POSITION ON STACK
0485 906000          727      MOV     DPTR,#6000H      ;SET POINTER TO DISP
0488 F0              728      MOVX    @DPTR,A          ;SEND CHARACTER TO DISPLAY
0489 1204FB          729      LCALL   DELAY
048C D083            730      POP     DPH              ;RESTORE POINTER-H POSITION
048E D082            731      POP     DPL              ;RESTORE POINTER-L POSITION
0490 A3              732      INC     DPTR             ;INCREMENT TO NEXT CHARACTER
0491 B4A0EB          733      CJNE    A,#0A0H,MESSAGE  ;REPEAT UNTIL ALL CHAR'S ARE DONE
0494 22              734      RET
                     735
                     736      DISP_MES:
                     737
0495 1204D5          738      LCALL   CURSH
0498 1204E3          739      LCALL   CLEARDISP
049B 1204FB          740      LCALL   DELAY
049E 9005CD          741      MOV     DPTR,#ADD_BATT   ;SET POINT TO ADD_BATT MESSAGE
04A1 12047F          742      LCALL   MESSAGE          ;DISP ADD_BATT MESSAGE
```

```
LOC  OBJ            LINE     SOURCE

04A4 1204D5          743     LCALL  CURSH
04A7 12027B          744     LCALL  LF
04AA 9005DE          745     MOV    DPTR,#FIN        ;SET POINT TO FIN MESSAGE
04AD 12047F          746     LCALL  MESSAGE          ;DISP FIN MESSAGE
04B0 22              747     RET
                     748
                     749     OPTIME:
                     750
04B1 906000          751     MOV    DPTR,#6000H      ;SET POINT TO DISP
04B4 F8              752     MOV    R0,A             ;STORE ACC IN R0
04B5 54F0            753     ANL    A,#0F0H          ;MASK LOWER ORDER BITS
04B7 C4              754     SWAP   A                ;EXCH LOWER- AND HIGHER-FOUR BITS
04B8 2430            755     ADD    A,#30H           ;CONVERT TO BCD VALUE
04BA F0              756     MOVX   @DPTR,A          ;SEND CHARACTER TO DISP
04BB 1204FB          757     LCALL  DELAY
04BE E8              758     MOV    A,R0             ;MOVE CONTENTS OF R0 TO ACC
04BF 540F            759     ANL    A,#0FH           ;MASK HIGHER ORDER BITS
04C1 2430            760     ADD    A,#30H           ;CONVERT TO BCD VALUE
04C3 F0              761     MOVX   @DPTR,A          ;SEND CHARACTER TO DISP
04C4 1204FB          762     LCALL  DELAY
04C7 22              763     RET
                     764
                     765     OPDISP:
                     766
04C8 906000          767     MOV    DPTR,#6000H      ;SET POINT TO DISP
04CB F8              768     MOV    R0,A             ;STORE ACC CONTENTS IN R0
04CC 540F            769     ANL    A,#0FH           ;MASK HIGHER ORDER BITS
04CE 2430            770     ADD    A,#30H           ;CONVERT TO BCD VALUE
04D0 F0              771     MOVX   @DPTR,A          ;SEND CHARACTER TO DISP
04D1 120500          772     LCALL  LONGDELAY
04D4 22              773     RET
                     774
                     775     CURSH:
                     776
04D5 C294            777     CLR    RS               ;SET RS LOW TO SEND CONTR- WORD TO DISP
04D7 906000          778     MOV    DPTR,#6000H      ;SET POINT TO DISP
04DA 7402            779     MOV    A,#02H           ;"CURSER HOME" TO ACC
04DC F0              780     MOVX   @DPTR,A          ;"CURSER HOME" TO DISP
04DD 1204FB          781     LCALL  DELAY
04E0 D294            782     SETB   RS               ;TO WR/RD DATA TO/FROM DISP
04E2 22              783     RET
                     784
                     785     CLEARDISP:
                     786
04E3 C294            787     CLR    RS               ;RS MUST BE LOW WHEN WRITING CONTROL WORDS TO DISP
04E5 906000          788     MOV    DPTR,#6000H      ;SET POINT TO DISP
04E8 7401            789     MOV    A,#01H           ;"DISP CLEAR" TO ACC
04EA F0              790     MOVX   @DPTR,A          ;"DISP CLEAR" TO DISP (CONTR WORD)
04EB 1204FB          791     LCALL  DELAY
04EE D294            792     SETB   RS               ;TO WR/RD DATA TO/FROM DISP
04F0 22              793     RET
                     794
                     795     COLON:
```

```
LOC  OBJ            LINE     SOURCE

                     796
04F1 906000          797     MOV     DPTR,#6000H       ;SET POINT TO DISP
04F4 743A            798     MOV     A,#3AH            ;"COLON CODE" TO ACC
04F6 F0              799     MOVX    @DPTR,A           ;"COLON CODE" TO DISP
04F7 1204FB          800     LCALL   DELAY
04FA 22              801     RET
                     802
                     803     DELAY:
                     804
04FB 7E9F            805     MOV     R6,#9FH           ;START VALUE OF DELAY LOOP
04FD DEFE            806     DJNZ    R6,$              ;DEC COUNTER UNTIL = 0
04FF 22              807     RET
                     808
                     809     LONGDELAY:
                     810
0500 7F6F            811     MOV     R7,#6FH           ;SET OUTER TIME LOOP
                     812     INNER:
0502 7EFF            813     MOV     R6,#0FFH          ;SET INNER TIME LOOP
0504 DEFE            814     DJNZ    R6,$              ;DEC INNER LOOP UNTIL ZERO
0506 DFFA            815     DJNZ    R7,INNER          ;DEC OUTER LOOP UNTIL ZERO
0508 22              816     RET
                     817
                     818     COMBHRS:
                     819
0509 E525            820     MOV     A,KEYSTORE1       ;MOVE 1 ST HR-DIGIT INTO ACC
050B 540F            821     ANL     A,#0FH            ;MASK HIGHER FOUR BITS
050D C4              822     SWAP    A                 ;EXCH LOWER- AND HIGHER-FOUR BITS
050E F525            823     MOV     KEYSTORE1,A       ;STORE IN KEYSTORE1
0510 E526            824     MOV     A,KEYSTORE2       ;MOVE 2 ND HR-DIGIT INTO ACC
0512 540F            825     ANL     A,#0FH            ;MASK HIGHER FOUR BITS
0514 4525            826     ORL     A,KEYSTORE1       ;LOGICAL ORR WITH KEYSTORE1
0516 F529            827     MOV     HRS,A             ;STORE RESULT IN HRS
0518 22              828     RET
                     829
                     830     COMBMIN:
                     831
0519 E527            832     MOV     A,KEYSTORE3       ;1ST DIGIT OF MIN TO ACC
051B 540F            833     ANL     A,#0FH            ;MASK HIGHER BITS
051D C4              834     SWAP    A                 ;EXCH LOWER- AND HIGHER-FOUR BITS
051E F527            835     MOV     KEYSTORE3,A       ;SAVE ACC CONTENTS
0520 E528            836     MOV     A,KEYSTORE4       ;2ND DIGIT OF MIN TO ACC
0522 540F            837     ANL     A,#0FH            ;MASK HIGHER BITS
0524 4527            838     ORL     A,KEYSTORE3       ;COMBINE 1ST + 2ND DIGITS OF MIN
0526 F52A            839     MOV     MIN,A             ;STORE ACC IN MIN
0528 22              840     RET
                     841
                     842     SETHRS:
                     843
0529 908004          844     MOV     DPTR,#8004H       ;SET POINT TO HOURS- ADDRESS OF RTC
052C E529            845     MOV     A,HRS             ;HOURS TO ACC
052E F0              846     MOVX    @DPTR,A           ;HOURS TO RTC
052F 1204FB          847     LCALL   DELAY
0532 22              848     RET
```

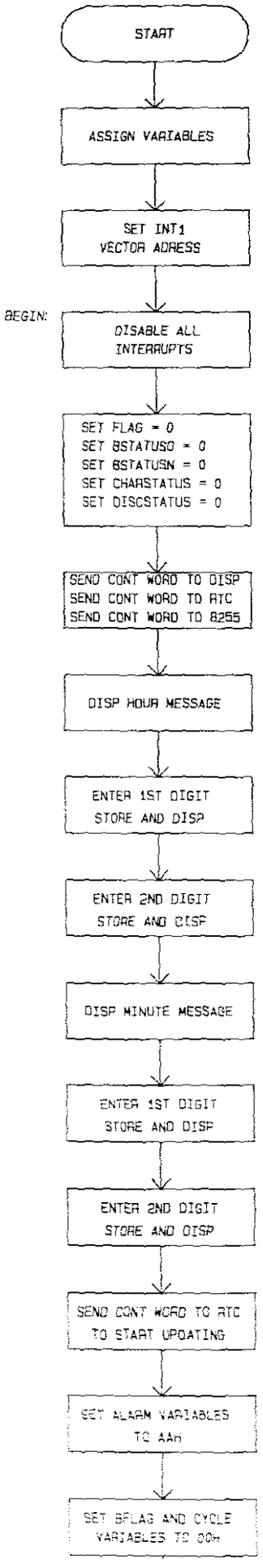
```
LOC  OBJ            LINE     SOURCE

                     849
                     850     SETMIN:
                     851
0533 908002          852     MOV     DPTR,#8002H       ;SET POINT TO MIN- ADDRESS OF RTC
0536 E52A            853     MOV     A,MIN             ;MIN TO ACC
0538 F0              854     MOVX    @DPTR,A           ;MIN TO RTC
0539 2405            855     ADD     A,#05             ;ACC = RTC MIN + 5
053B F55F            856     MOV     SCANC,A           ;SCANC = RTC MIN + 5
053D 1204FB          857     LCALL   DELAY
0540 22              858     RET
                     859
                     860     SETSEC:
                     861
0541 908000          862     MOV     DPTR,#8000H       ;SET POINT TO SEC- ADDRESS OF RTC
0544 7400            863     MOV     A,#00H            ;00 SEC TO ACC
0546 F0              864     MOVX    @DPTR,A           ;00 SEC TO RTC
0547 1204FB          865     LCALL   DELAY
054A 22              866     RET
                     867
                     868     BCD_HEX:
                     869
054B F525            870     MOV     KEYSTORE1,A       ;STORE HOURS TEMP
054D 54F0            871     ANL     A,#0F0H           ;MASK LOWER-4 BITS
054F C4              872     SWAP    A                 ;SWAP HIGHER- AND LOWER-4 BITS
0550 75F00A          873     MOV     B,#0AH            ;MOVE 0AH INTO REG B
0553 A4              874     MUL     AB                ;ACC * REG B
0554 F5F0            875     MOV     B,A               ;STORE ANSWER IN REG B
0556 E525            876     MOV     A,KEYSTORE1       ;LOAD HOURS INTO ACC
0558 540F            877     ANL     A,#0FH            ;MASK HIGHER-4 BITS
055A 25F0            878     ADD     A,B               ;ACC + REG B = BCD VALUE
055C 22              879     RET
                     880
                     881     INT_SCAN:
                     882
055D C083            883     PUSH    DPH
055F C082            884     PUSH    DPL
0561 C0E0            885     PUSH    ACC
0563 7528FF          886     MOV     KEYSTORE4,#0FFH   ;SET KEYSTORE4 = FFH DUE TO INTERRUPT
0566 90A000          887     MOV     DPTR,#0A000H      ;SET POINT TO 74C922
0569 E0              888     MOVX    A,@DPTR           ;READ KEY FROM KEYBOARD INTO ACC
056A 540F            889     ANL     A,#0FH            ;MASK OFF HIGHER-FOUR BITS
056C F527            890     MOV     KEYSTORE3,A       ;STORE CHAR IN KEYSTORE3
056E D0E0            891     POP     ACC
0570 D082            892     POP     DPL
0572 D083            893     POP     DPH
0574 32              894     RETI                      ;RETURN AFTER INTERRUPT
                     895
                     896     SCAN:
                     897
0575 90A000          898     MOV     DPTR,#0A000H      ;SET POINT TO 74C922
0578 E0              899     MOVX    A,@DPTR           ;READ KEY FROM KEYBOARD INTO ACC
0579 540F            900     ANL     A,#0FH            ;MASK OFF HIGHER-FOUR BITS
057B 22              901     RET
```

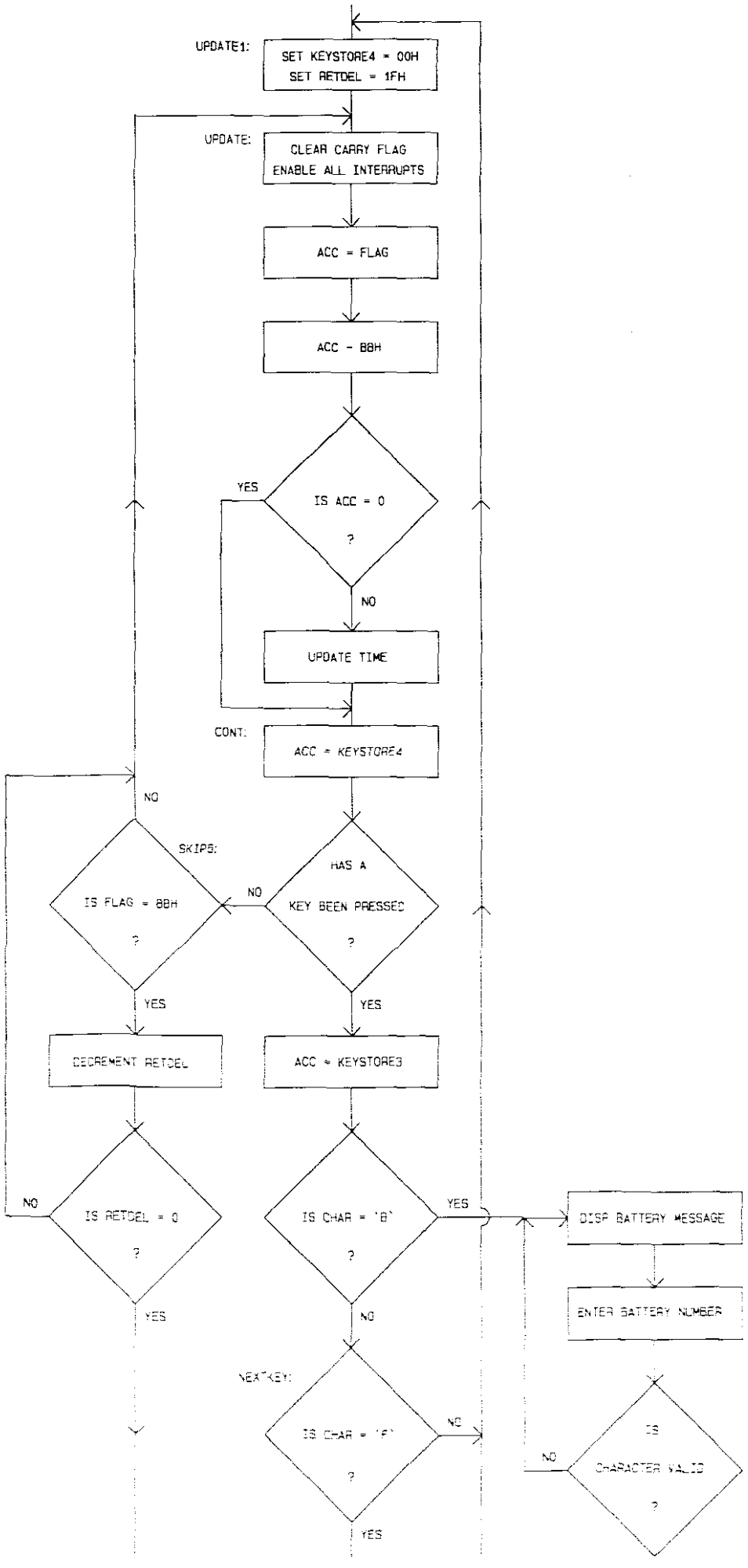
```
LOC  OBJ           LINE     SOURCE

                    902
057C 31203220       903     MENU:     DB '1 2 3 4 5 6 7 8',0A0H   ;BATTERY NO'S
0580 33203420
0584 35203620
0588 372038
058B A0
                    904
058C 454E5452       905     HOURS:    DB 'ENTR HRS 00-23 ?',0A0H  ;MESSAGE FOR HOURS
0590 20485253
0594 2030302D
0598 3233203F
059C A0
059D 454E5452       906     MINUTES:  DB 'ENTR MIN 00-59 ?',0A0H  ;MESSAGE FOR MINUTES
05A1 204D494E
05A5 2030302D
05A9 3539203F
05AD A0
                    907
05AE 454E5452       908     BATMES:   DB 'ENTR BATT: 1-8 ?',0A0H  ;MESSAGE FOR BATT-NO.
05B2 20424154
05B6 543A2031
05BA 2D38203F
05BE A0
05BF 454E5452       909     CYCLEMES: DB 'ENTR CYCLES ?',0A0H     ;MESSAGE FOR NO OF CYCLES
05C3 20435943
05C7 4C455320
05CB 3F
05CC A0
                    910
05CD 41444420       911     ADD_BATT: DB 'ADD MORE BATT OR',0A0H  ;MESSAGE FOR ADDING BATTERIES
05D1 4D4F5245
05D5 20424154
05D9 54204F52
05DD A0
05DE 46494E49       912     FIN:      DB 'FINISH? ENTR B/F',0A0H  ;MESSAGE FOR FINISH
05E2 53483F20
05E6 454E5452
05EA 20422F46
05EE A0
                    913
05EF 00             914     DETBATT:  DB  00H,01H,02H,04H,08H,10H,20H,40H,80H
05F0 01
05F1 02
05F2 04
05F3 08
05F4 10
05F5 20
05F6 40
05F7 80
                    915
05F8 00             916     C_POINT:  DB  00H,34H,37H,3AH,3DH,40H,43H,46H,49H
05F9 34
05FA 37
```

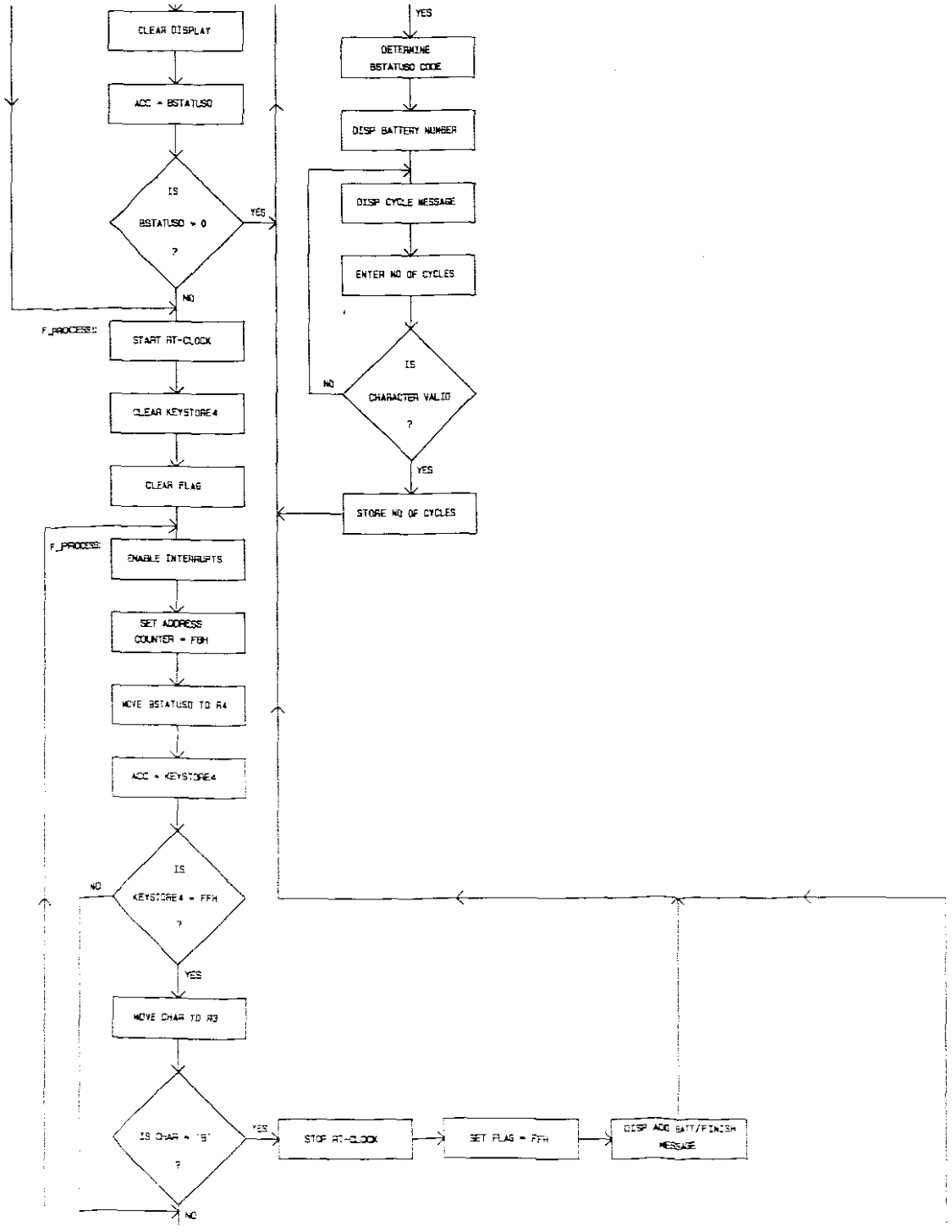
```
LOC  OBJ            LINE     SOURCE

05FB 3A
05FC 3D
05FD 40
05FE 43
05FF 46
0600 49
                    917
                    918      END
```

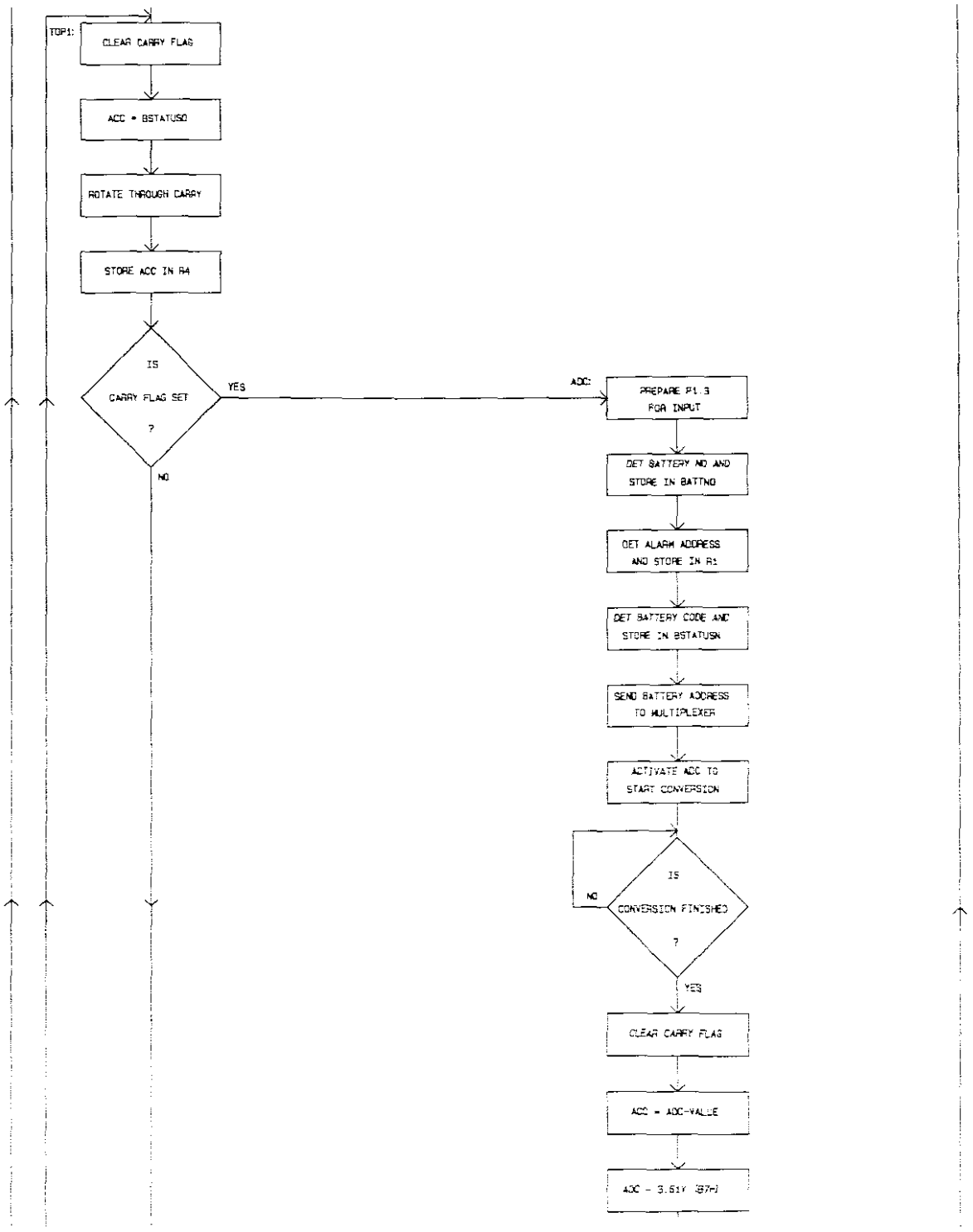
Program Capture (GWBASIC- Program)

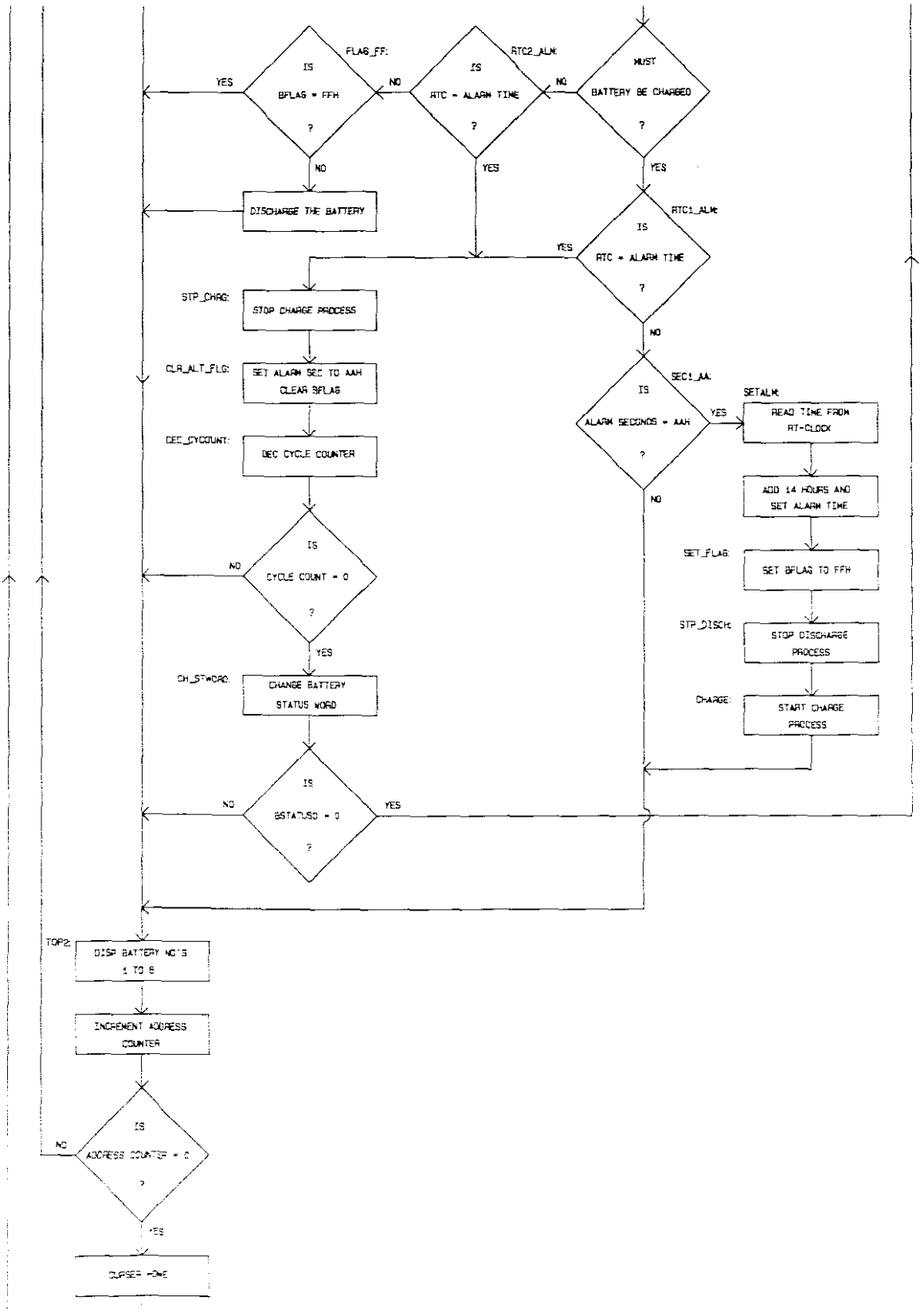
```
10 OUT 1019,128
20 OUT 1016,96
30 OUT 1017,0
40 OUT 1019,3
50 D=0                                  'Set time counter = 0
60 LINE INPUT "FILE?";DSKFIL$           'Enter the file name
70 OPEN DSKFIL$ FOR OUTPUT AS #1        'Open file
802 Z=INP(1016)                         'Clear the port
90 FOR I=1 TO 40                        'Set counter
100 A=INP(1021)                         'Read port register
110 A=A AND 1                           'Mask data
120 IF A=1 THEN GOTO 150                'Data received?
130 C$=INKEY$:IF C$= "" THEN 100        'Any key pressed?
140 GOTO 200                            'If keypressed goto 200
150 B=INP(1016)                         'Read data from port
160 D=D+5                               'Time counter + 5 minutes
170 PRINT D"Minutes Passed",B/18.53"Volt" 'Disp time + data
180 PRINT #1,B/18.53                    'Write data to file
190 NEXT I                              'Repeat until I = 0
200 CLOSE #1                            'Close file
210 END
```

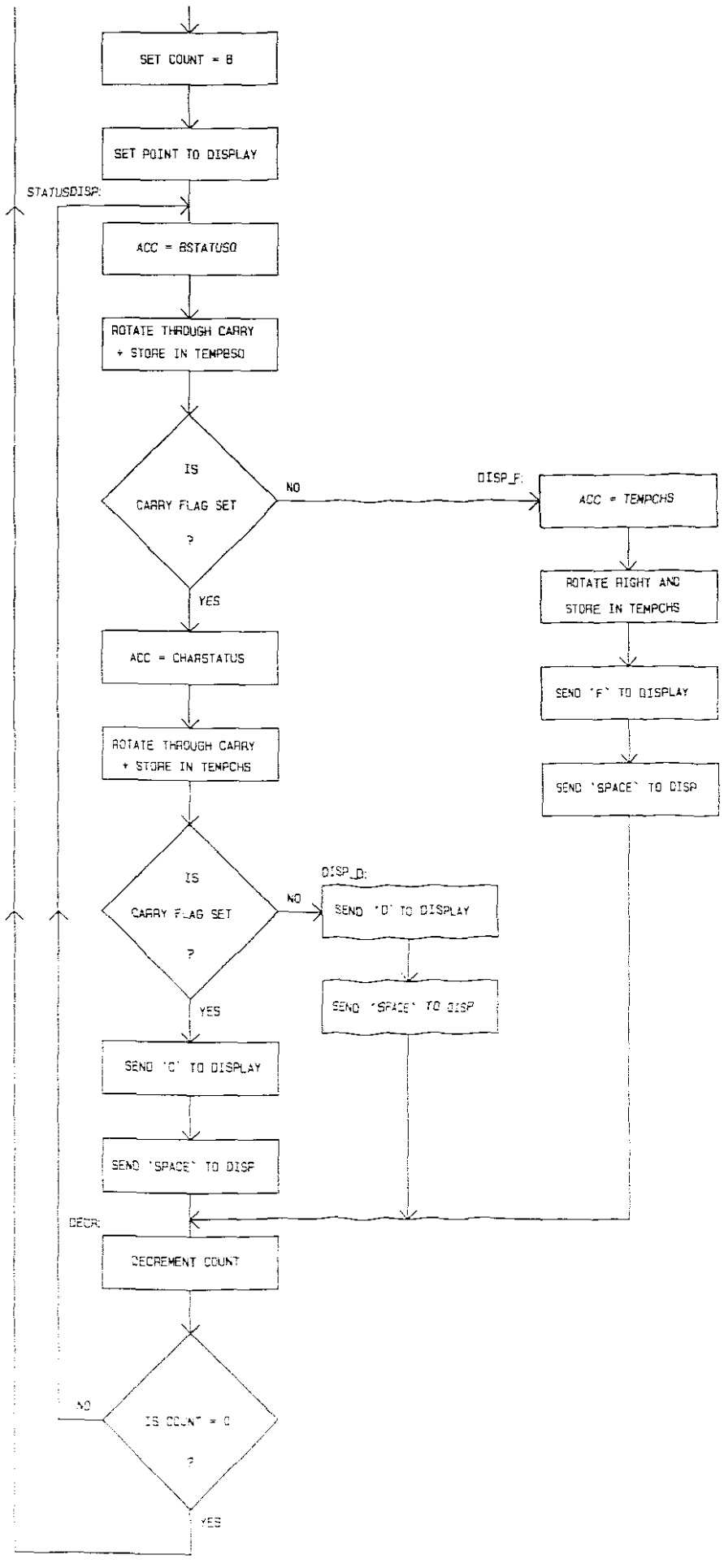

START
ASSIGN VARIABLES
SET INT1
VECTOR ADRESS
BEGIN:
DISABLE ALL
INTERRUPTS
SET FLAG = 0
SET BSTATUSO = 0
SET BSTATUSN = 0
SET CHARSTATUS = 0
SET DISCSTATUS = 0
SEND CONT WORD TO DISP
SEND CONT WORD TO RTC
SEND CONT WORD TO 8255
DISP HOUR MESSAGE
ENTER 1ST DIGIT
STORE AND DISP
ENTER 2ND DIGIT
STORE AND DISP
DISP MINUTE MESSAGE
ENTER 1ST DIGIT
STORE AND DISP
ENTER 2ND DIGIT
STORE AND DISP
SEND CONT WORD TO RTC
TO START UPDATING
SET ALARM VARIABLES
TO AAh
SET BFLAG AND CYCLE
VARIABLES TO 00h

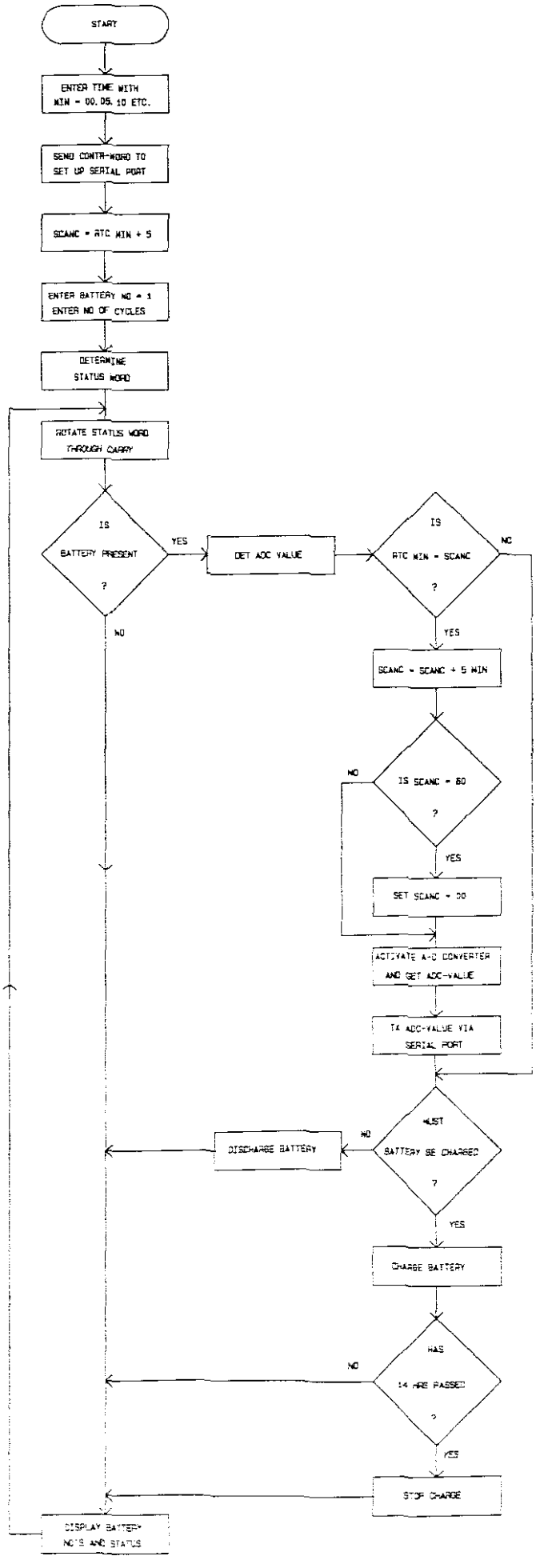

UPDATE1:
SET KEYSTORE4 = 00H
SET RETDEL = 1FH
UPDATE:
CLEAR CARRY FLAG
ENABLE ALL INTERRUPTS
ACC = FLAG
ACC = 88H
IS ACC = 0 ?
YES
NO
UPDATE TIME
CONT:
ACC = KEYSTORE4
NO
SKIPS:
IS FLAG = 88H ?
NO
HAS A KEY BEEN PRESSED ?
YES
YES
DECREMENT RETDEL
ACC = KEYSTORE3
NO
IS RETDEL = 0 ?
YES
IS CHAR = 'B' ?
YES
DISP BATTERY MESSAGE
ENTER BATTERY NUMBER
NO
NEXTKEY:
IS CHAR = 'F' ?
NO
YES
NO
IS CHARACTER VALID ?


CLEAR DISPLAY
ACC = BSTATUS0
IS BSTATUS0 = 0 ?
YES
NO
F_PROCESS:
START RT-CLOCK
CLEAR KEYSTORE4
CLEAR FLAG
F_PROCESS:
ENABLE INTERRUPTS
SET ADDRESS COUNTER = FBH
MOVE BSTATUS0 TO R4
ACC = KEYSTORE4
IS KEYSTORE4 = FFH ?
NO
YES
MOVE CHAR TO R3
IS CHAR = '0' ?
YES
NO
STOP RT-CLOCK
SET FLAG = FFH
DISP ADD BATT/FINISH MESSAGE
YES
DETERMINE BSTATUS0 CODE
DISP BATTERY NUMBER
DISP CYCLE MESSAGE
ENTER NO OF CYCLES
IS CHARACTER VALID ?
NO
YES
STORE NO OF CYCLES


TOP1:
CLEAR CARRY FLAG
ACC = BSTATUS0
ROTATE THROUGH CARRY
STORE ACC IN R4
IS
CARRY FLAG SET
?
YES
NO
ADC:
PREPARE P1.3
FOR INPUT
GET BATTERY NO AND
STORE IN BATTNO
GET ALARM ADDRESS
AND STORE IN R1
GET BATTERY CODE AND
STORE IN BSTATUSN
SEND BATTERY ADDRESS
TO MULTIPLEXER
ACTIVATE ADC TO
START CONVERSION
IS
CONVERSION FINISHED
?
NO
YES
CLEAR CARRY FLAG
ACC = ADC-VALUE
ADC = 3.61V (87H)


FLAG_FF:
IS BFLAG = FFH ?
YES
NO
RTC2_ALM:
IS RTC = ALARM TIME ?
NO
MUST BATTERY BE CHARGED ?
YES
NO
DISCHARGE THE BATTERY
RTC1_ALM:
IS RTC = ALARM TIME ?
YES
NO
STP_CHRG:
STOP CHARGE PROCESS
CLR_ALT_FLG:
SET ALARM SEC TO AAH
CLEAR BFLAG
DEC_CYCOUNT:
DEC CYCLE COUNTER
IS CYCLE COUNT = 0 ?
NO
YES
CH_STWORD:
CHANGE BATTERY STATUS WORD
IS BSTATUS0 = 0 ?
NO
YES
SEC1_AA:
IS ALARM SECONDS = AAH ?
YES
NO
SETALM:
READ TIME FROM RT-CLOCK
ADD 14 HOURS AND SET ALARM TIME
SET_FLAG:
SET BFLAG TO FFH
STP_DISCH:
STOP DISCHARGE PROCESS
CHARGE:
START CHARGE PROCESS
TOP2:
DISP BATTERY NO'S 1 TO 6
INCREMENT ADDRESS COUNTER
IS ADDRESS COUNTER = 0 ?
NO
YES
CURSER HOME


SET COUNT = 8
SET POINT TO DISPLAY
STATUSDISP:
ACC = BSTATUSO
ROTATE THROUGH CARRY + STORE IN TEMPBSO
IS CARRY FLAG SET ?
NO
DISP_F:
ACC = TEMPCHS
ROTATE RIGHT AND STORE IN TEMPCHS
SEND 'F' TO DISPLAY
SEND 'SPACE' TO DISP
YES
ACC = CHARSTATUS
ROTATE THROUGH CARRY + STORE IN TEMPCHS
IS CARRY FLAG SET ?
NO
DISP_D:
SEND 'D' TO DISPLAY
SEND 'SPACE' TO DISP
YES
SEND 'C' TO DISPLAY
SEND 'SPACE' TO DISP
DECR:
DECREMENT COUNT
IS COUNT = 0 ?
NO
YES


START
ENTER TIME WITH MIN = 00, 05, 10 ETC.
SEND CONTR-WORD TO SET UP SERIAL PORT
SCANC = RTC MIN + 5
ENTER BATTERY NO = 1 ENTER NO OF CYCLES
DETERMINE STATUS WORD
ROTATE STATUS WORD THROUGH CARRY
IS BATTERY PRESENT ?
YES
NO
GET ADC VALUE
IS RTC MIN = SCANC ?
NO
YES
SCANC = SCANC + 5 MIN
IS SCANC = 60 ?
NO
YES
SET SCANC = 00
ACTIVATE A-D CONVERTER AND GET ADC-VALUE
TX ADC-VALUE VIA SERIAL PORT
MUST BATTERY BE CHARGED ?
NO
YES
DISCHARGE BATTERY
CHARGE BATTERY
HAS 14 HRS PASSED ?
NO
YES
STOP CHARGE
DISPLAY BATTERY NO'S AND STATUS

6. Problems Encountered.

6.1 Hardware.

Any batteries connected to the charger, whether they are being cycled or not, will start to discharge in the case of a power failure. To prevent this from happening, pull-down resistors were put on the base of the first switching transistor to keep it switched off until the discharge circuit is switched on by a +5V signal from the control unit.

To select the display a NOR-gate is required to combine the CS-3 and WR signals in order to overcome timing problems between the CPU and the display.

Initially a 5V regulator was used to supply a constant voltage to the processor unit. As a result of the voltage drop across the tracks the processor malfunctioned intermittently. To overcome this problem an adjustable power supply is used in order to supply the required voltage at the input terminals of the processor.

6.2 Software.

It is important that the charging process must be stopped after 14 hours. The comparison of alarm-times with RT-Clock time is aborted when more batteries are being entered. While batteries are being entered, any alarm-time might equal the RT-Clock time. This will not be noted when returned to the cycling process and the battery will be charged indefinitely. To prevent this from happening, the RT-Clock is stopped when batteries are added, and started again when returned to the cycling process.

It is obvious that the RT-Clocks' time will not be correct after more batteries have been entered. This is of no concern as the time is not displayed in the idle condition.

7. Operating Instructions.

7.1 Initial Power-On.

The user will be prompted to enter the hours by the following displayed message:

ENTR HRS 00-23 ?

The 24-hour mode is used and the time must be entered accordingly eg: 9 am must be entered as 09 and 9 pm as 21. After the second digit has been entered the following message will be displayed:

ENTR MIN 00-59 ?

Two digits must be used to enter the minutes eg: 4 minutes must be entered as 04. After the last digit has been entered the time will be displayed eg:

09:04:00

7.2 Charging After Initial Power-On.

The batteries that need to be cycled may be placed in any spare positions in the unit. Enter a 'B' and the following message will be displayed:

ENTR BATT: 1-8 ?

Enter the battery number corresponding to the number on the unit and the following message will be displayed:

ENTR CYCLES ?

After entering the number of cycles for that specific battery the time will be displayed again.

More batteries may be entered by entering a 'B' again. When all batteries have been entered an 'F' needs to be entered to start the cycling process. After a short delay the following, by way of example, will be displayed:

```
C F D C F F F F
1 2 3 4 5 6 7 8
```

1-> 8 indicates the battery numbers on the unit.
'C' means that the battery is being charged.
'D' means that the battery is being discharged.
'F' means that the battery has finished cycling or that the position is spare (unused).

7.3 Charging Subsequent Batteries.

To add batteries while others are being cycled or if all positions are spare (unused). Enter a 'B' and the following message will be displayed:

```
ADD MORE BATT OR
FINISH? ENTR B/F
```

To add a battery enter 'B' within 5 seconds and the following message will be displayed:

```
ENTR BATT: 1-8 ?
```

Enter the battery number and the following message will be displayed:

```
ENTR CYCLES ?
```

Enter the number of cycles and the following message will be displayed:

```
ADD MORE BATT OR
FINISH? ENTR B/F
```

Enter a 'B' within 5 seconds to enter more batteries or an 'F' when finished. If nothing is entered within 5 seconds the unit will automatically return to the process and the following will be displayed, for example:

C D D C C F F F
1 2 3 4 5 6 7 8

When all the cycles have been completed or if all positions are spare (unused) the following will be displayed:

F F F F F F F F
1 2 3 4 5 6 7 8

NB: ALWAYS ENSURE THAT THE UNIT RETURNS TO THE PROCESSING MODE OTHERWISE THE BATTERIES MAY BE OVER-CHARGED OR OVER-DISCHARGED, RESULTING IN POSSIBLE DAMAGE TO THE BATTERIES.

7.4 Using The RS-232 Serial Port.

The Serial Port may be used to plot the charge and discharge curve of a particular battery. A serial cable, Personal Computer (PC) and software packages that can capture the data from the serial port and plot a graph are required. The BASIC program, Capture (Refer to page 45), is used to capture the data from the processor unit at a baud rate of 1200. This program automatically sets up the mode for the serial port (Com1) ie: 1200,n,8,2,p. Connect the charger unit to the serial port of the PC and follow the procedure below:

Run the Capture program. When prompted, enter a file name to which the data must be captured, eg. Test1.txt. The charger unit must not be in use. Reset the unit with the reset button. Place the battery to be tested in position number 1. Enter the time. The minutes must be entered as 00 or a multiple of 5 eg. 05,10,15 etc. Enter the battery number (1) and number of cycles as described in section 7.2.

Start the process by entering an 'F'. Incoming data is captured and converted to the appropriate battery voltages which are then displayed on the PC monitor at five minute intervals. This will be done irrespective of whether the battery is being charged or discharged. This information is automatically stored in the text file.

A package, like Quattro Pro, may now used to import the captured file. Enter the x-axis values as five minute intervals and plot the graph.

8. Test Results.

It was the objective to improve the efficiency of the Ni-Cad batteries through a cycling process to erase any memory effect and/or to prevent the formation thereof.

It must be stressed again that the existing charger is a charge only system with no current or time control. The new charger has current and time control and depending on the battery's condition, also discharges it to the voltage cutoff point.

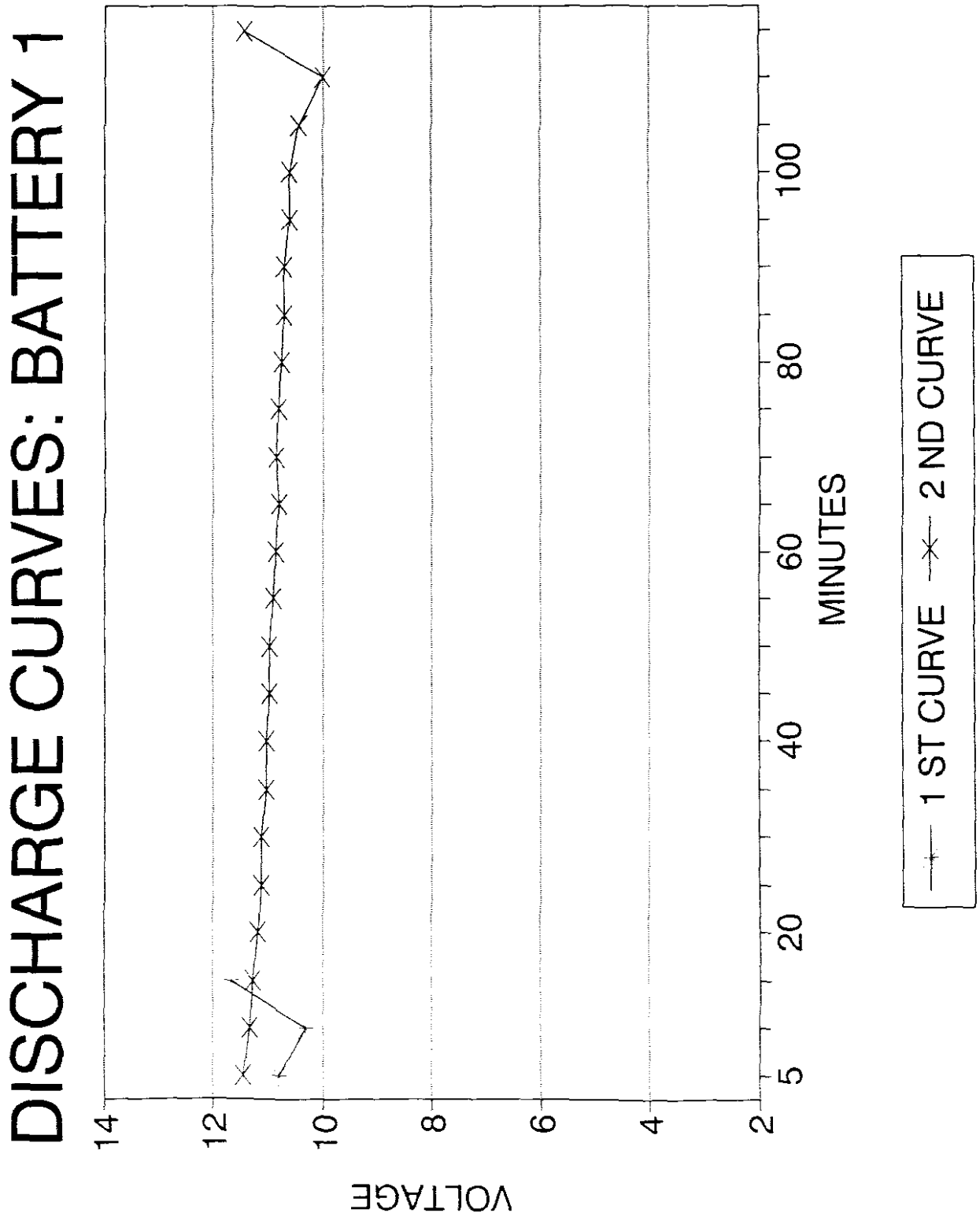
The age of the batteries, and whether they had the memory effect or not, was not known when they were received from the Security Section. Firstly all batteries were fully charged with the existing charger. They were then cycled by this new charger and their discharge curves were plotted. The times taken to discharge to the cutoff voltage point is of importance as it indicates the efficiency of the batteries.

The following results were obtained:

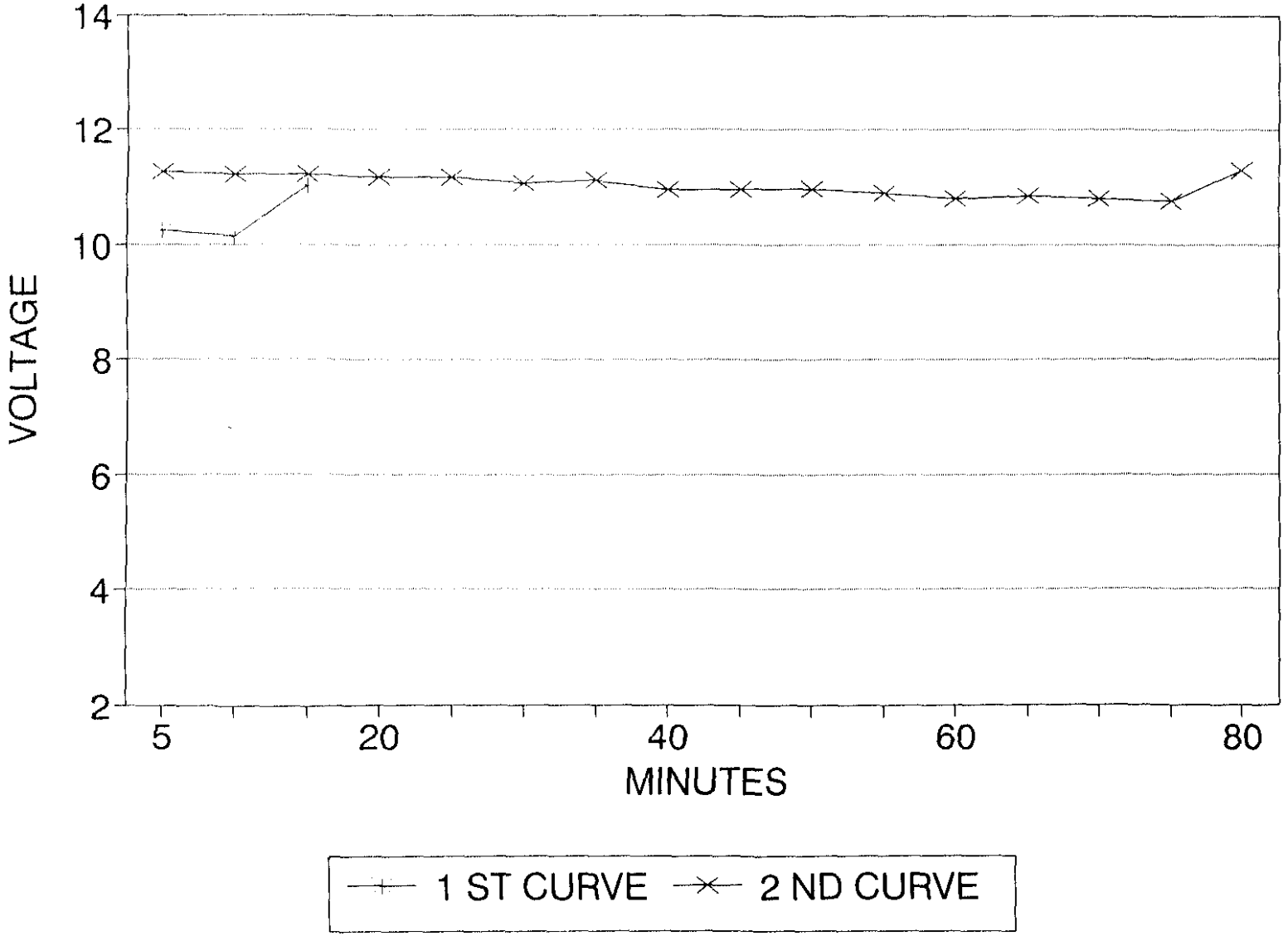
Battery Number	1st Disch. Curve	2nd Disch. Curve	% Efficiency Improvement
1	+10 Min	+110 Min	1000 %
2	+10 Min	+75 Min	650 %
3	+25 Min	+140 Min	460 %

The above results and the curves on pages 59 to 61 show that the batteries suffered from the memory effect. Clearly there was a vast improvement after only one cycle with the new charger. The curve for battery 4 on page 62 is that of a normally used battery without the memory effect.

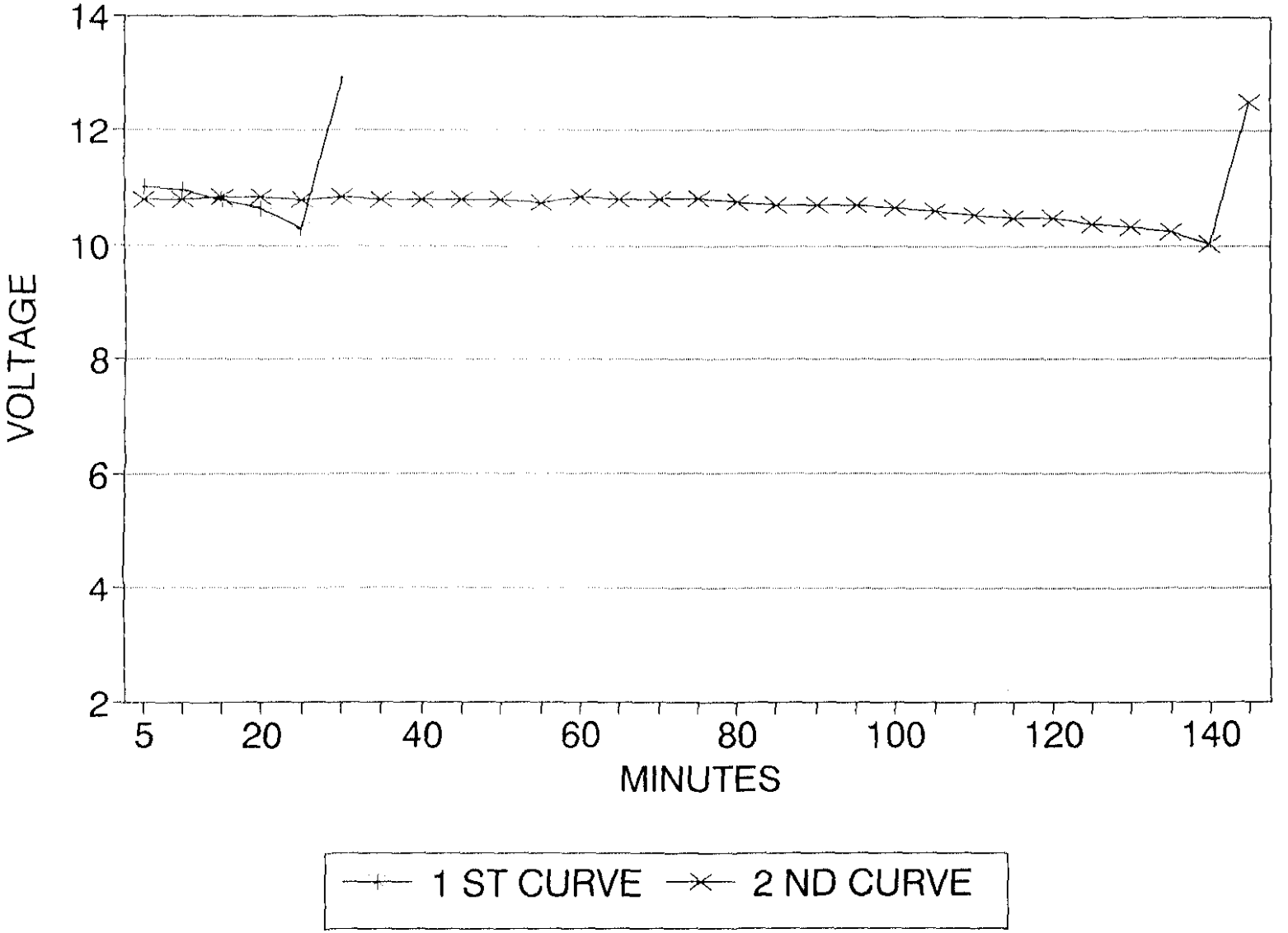
As a result of this proper charging technique the life-span of these batteries will also be increased.

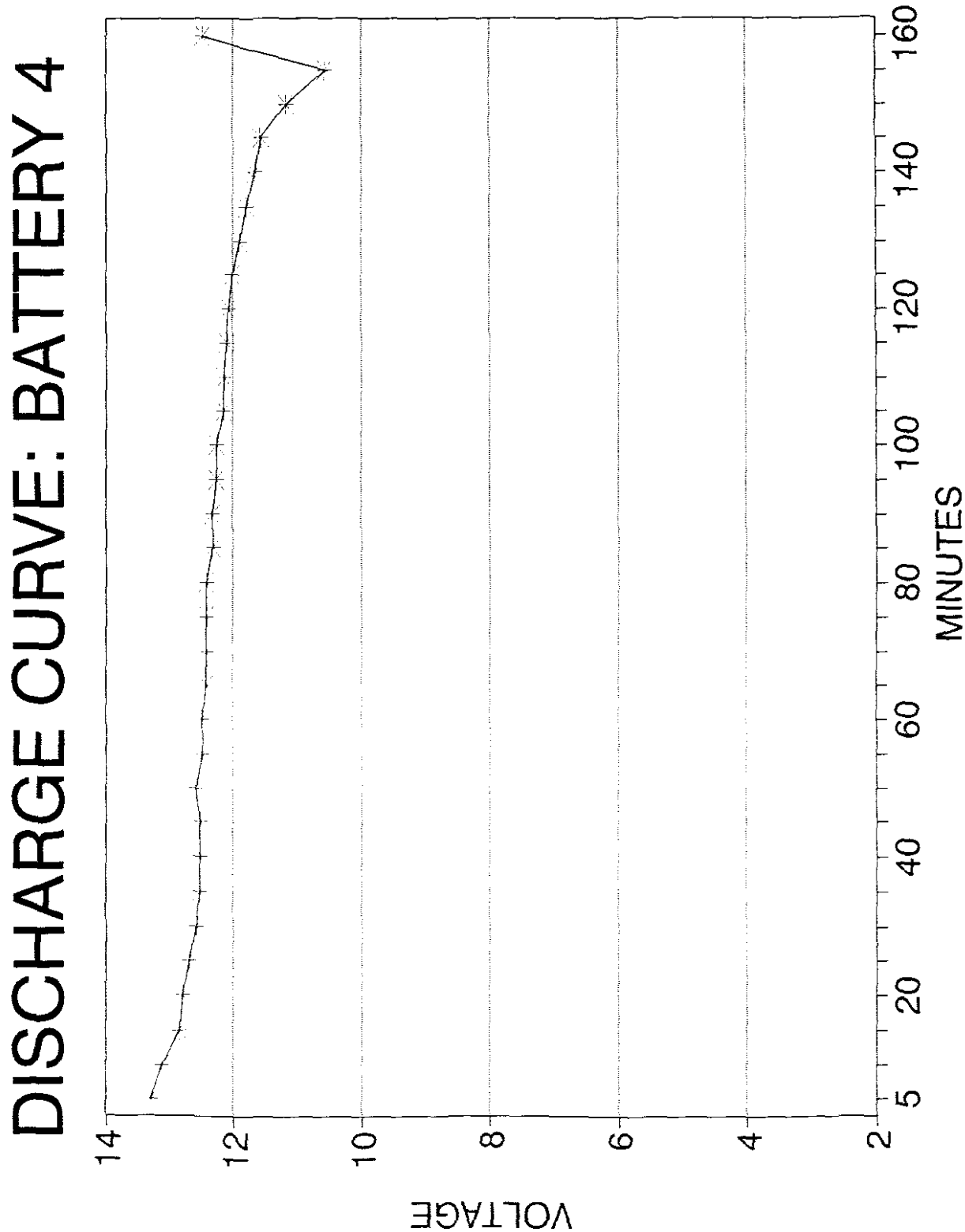

DISCHARGE CURVES: BATTERY 1
VOLTAGE
14
12
10
8
6
4
2
5
20
40
60
80
100
MINUTES
1 ST CURVE
2 ND CURVE

DISCHARGE CURVES: BATTERY 2



DISCHARGE CURVES: BATTERY 3




DISCHARGE CURVE: BATTERY 4
VOLTAGE
14
12
10
8
6
4
2
5
20
40
60
80
100
120
140
160
MINUTES

9. Bibliography.

Intel Corporation. 1990. **8-Bit Embedded Controllers.** Mt. Prospect, Intel Literature Sales. 6-1 to 6-73.

Intel Corporation. 1988. **Memory Components Handbook.** Santa Clara, Intel Literature Sales. 4-10 to 4-20.

Intel Corporation. 1988. **Microprocessor and Peripheral Handbook.** Santa Clara, Intel Literature Sales. 2-63 to 2-86.

Motorola Semiconductors. 1979. **The European CMOS Selection.** Switzerland. 9-141 to 9-146.

Motorola Semiconductors. 1988. **Microprocessor Microcontroller And Peripheral Data.** Switzerland.

National Semiconductor Corporation. 1988. **Linear Databook 1.** Santa Clara. 1-163 to 1-173, 1-254 to 1-258.

National Semiconductor Corporation. 1984. **CMOS Databook.** Santa Clara. 1-308 to 1-310.

National Semiconductor Corporation. 1988. **CMOS Logic Databook** Santa Clara. 6-151 to 6-156.

Perez, R.A. 1985. **The Complete Battery Book.** Blue Ridge Summit, Tab Books Inc. 41 to 62.

Philips. 1991. **Data Handbook Liquid Crystal Displays LCD01.**

Texas Instruments. 1989. **The TTL Databook Volume 1.** Germany. 3-411 to 3-412.

Towers, T.D. 1980. **Towers' International Transistor Selector Third Edition.** Blue Ridge Summit, Tab Books Inc. 50,149,171.

10. Acknowledgments.

I wish to thank my colleagues in the Research And Development Centre of the Department of Posts and Telecommunications, especially Mr. G.V. Williams, with whom many helpful discussions were held.