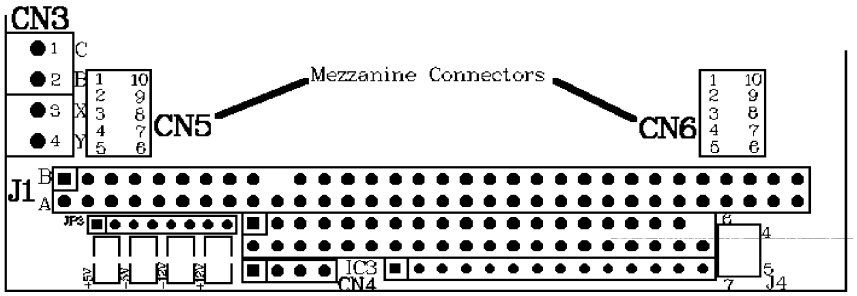
BC104 Battery Charger and PM104 Power Management Units

1. Description
When the BC104 and PM104 units are both installed on either the V104 or HE104 (hereafter referred to as PSU), an universal battery charger can be setup, and the PSU unit made into an UPS (uninterruptable power supply).

The BC104 is a constant current "buck" switching regulator, with an adjustable "float" voltage. The float voltage is adjusted via a potentiometer. The PM104 is programmed by the user using a "control basic" called PBASIC. A sample program is supplied to show a typical NiCd charging control. Before using the BC104 and PM104 the battery charging program must be set up for the intended battery pack. The sample program has separate settings for normal charge current, and trickle charge current. In addition, the charge termination methods should be set, including maximum charge time, negative delta V. The user must set these for the type and size of battery to be charged. Typically charge current will be 1/3 to 1/6 of battery capacity, and trickle charge current 1/20 to 1/30 of battery capacity. Addition of a battery temperature sensor will allow charge termination when elevated battery temperatures (which indicates battery is fully charged).

The PM104 can be programmed for many additional features not included in the sample program. Features such as setting a PC/104 bus interrupt when main power fails, stopping the PSU after running on battery backup power for a set time, tracking power consumption so that backup battery charging can be terminated when same amount restored to the battery. These features are left to the OEM integrate into their design.

2. Connections:
The BC104 is mounted on two connectors under the PSU heatsink, CN5 and CN6. The PM104 is factory installed directly in front of the PC/104 bus, location IC3. Connector CN4 is for connection to a PC parallel port for programming.
Batteries are connected to the screw terminal block, CN3. The PSU accepts DC battery voltages



in the range 6.5V to 20VDC through the Battery Power Connector CN3. Two external signals can be connected to the battery terminal block for use by add-on modules plugged into the mezzanine header connectors. Connect to the HE104 Battery Terminal Block as follows:

- Terminal 1: Common of battery
- Terminal 2: Positive Battery Terminal
- Terminal 3: External signal 1, normally connected to terminal 2
- Terminal 4: External signal 2, 0 to 40V input

The two external signals are feed into the 12bit analog to digital convertor, and will accept voltages up to 40V. The sample program requires the External signal 1(Terminal 3) be connected to Positive Battery voltage (Terminal 2) for battery voltage sensing.

A variety of temperature sensors can be connected including thermistors, and conditioned sensors such as LM35s. The LM35 series is particularly nice because their output voltage is directly proportional to temperature (ie 10mV/C or 10mV/F). In any case, the OEM can "experiment" to determine what works best in their application.

3. Programming Cable:
The programming cable is plugged into the connector CN4 on the PSU and the other end into the 25pin DB parallel port connector on a PC. The programming cable has the following connections:

CN4-1	No connection
CN4-2	Connect to pin 25 on parallel port
CN4-3	Connect to pin 11 parallel port
CN4-4	Connect to pin 2 parallel port

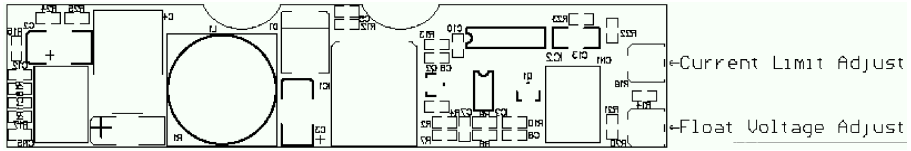
4. Download and Edit Software:
All programs for the PM104 are written in a "Control Basic" program language, and are saved into an ASCII file with a "BAS" extension. Any text editor can be used to create, edit and save these programs. The Download program called "Stamp.exe" also has simple editing capabilities.

After the program cable is connected between the PSU and the parallel port, the PSU unit can be turned on, thus providing power to the PM104 unit. The Download program Stamp.exe is started by typing from DOS, STAMP.EXE. The program to be downloaded is opened by pressing the keys "ALT" and "L" simultaneously. Using the arrow keys select the desired file, and press ENTER key. To download the program press the keys "ALT" and "R" simultaneously. If the cable is properly connected, and power applied the screen will show a horizontal bar graph indicating the percent of program downloaded. The red area of the bar graph is the portion used, and the remainder is program space available.

5. Program Command and Syntax:
Please refer to the Adobe files BSBOOK1.PDF and BSBOOK2.PDF.

6. Adjusting the BC104 Float Limits:
When a PC/104 power supply is equipped with a BC104 and a PM104 the BC104 has a Float Voltage Adjust potentiometer. However, the Current Limit Adjust potentiometer is not installed and is controlled via the PM104.

Using a small screwdriver (flexible nylon works best), turn the potentiometer until the desired float voltage is obtained. No load should be present when adjusting.



7. Sample Battery Charging Program Listing
The following program listing is intended for use as a guide to customizing the BC104, and PM104 operation. Additional functions and features can be added, including temperature monitoring are left up to the OEM to implement.

```
' ==========================================================
' BC104, Battery Charger Sample program code
' ==========================================================

SYMBOL Pwr_Status = 0                ' Status of input power
SYMBOL Pwrp_Status = pin0            ' Pin number of status of input power
SYMBOL PSU_OnOff = 1                 ' OnOff control of power supply
SYMBOL PSUp_OnOff = pin1
SYMBOL CS1 = 2                        ' Chip select  A/D on Battery Charger; 0 = active
SYMBOL CS1p = pin2
SYMBOL Int2 = 3                       ' PC/104 bus interrupt
SYMBOL Int2p = pin3
SYMBOL DIO = 4                        ' Pin _number_ of data input/output.
SYMBOL DIOp = pin4            ' Variable_name_ of data input/output.
SYMBOL CLK = 5                ' Clock to ADC; out on rising, in on falling edge.
SYMBOL CLKp = pin5
SYMBOL Int1 = 6                      ' PC/104 bus interrupt
SYMBOL Int1p = pin6
SYMBOL Chrg_Limit = 7                ' PWM output for current limit
SYMBOL Chrgp_Limit = pin7


SYMBOL Bat_Sel = bit1
SYMBOL ADbits = b1                   ' Counter variable for serial bit reception.
SYMBOL Bat1_Chrg = bit1
SYMBOL D0 = bit2                     ' LSB of ADC channel selection
SYMBOL D1 = bit3                     ' second bit of ADC channel selection
      'D1 = 0, D0 = 0 channel 0 input, pin3 of connector CN3
      'D1 = 0, D0 = 1 channel 1 input, pin4 of connector CN3
```

```
        'D1 = 1, D0 = 0 channel 2 input, monitors input voltage of battery regulator
        'D1 = 1, D0 = 1 channel 3 input, monitors battery charging current
'Note: channel 0 is usually jumpered to CN3 term2 for monitoring battery voltage
'Note: channel 2 tracks  main power input when greater than battery voltage
'Note: channel 2 approx. 0.6V less than battery voltage when main input less than battery voltage

SYMBOL AD = w1                          ' 12-bit ADC conversion result.
SYMBOL Chrg_Time = w2                   ' 16 bit timer
SYMBOL Batt_Peak = w3                   ' Peak voltage detected
SYMBOL TCnt = b8
SYMBOL Batt_V = w5
SYMBOL sglDif = 1                        ' Single-ended, two-channel mode.
SYMBOL msbf = 1                 ' Output 0s after data transfer is complete.
SYMBOL AO1_LVL = 5                      ' Maximum current level (50 = 1A 75 = 1.5A)
SYMBOL BattV_Max = 1100                 ' Maximum battery pack charge voltage (14.25V)
SYMBOL Chrg_Time_Max = 10800            ' Maximum battery charging time (10,800 = 3hr.)
SYMBOL Neg_DeltaV = 8                   ' AD convertor points for -deltaV (74pt./V IE 0.2V = 18pts.)
SYMBOL Trickle_LVL = 0                  ' Trickle Charge Level (12 =.25A) See below:
                                        '     Trickle charge on 50% duty cycle
SYMBOL BattV_Min = 740                  ' Minimum battery voltage (10V)

' ========================================================
' Main Loop
' ========================================================
Init:
 Low PSU_OnOff                          'Turn PSU on
 let Bat1_chrg = 0
Main_Batt1:
 Low PSU_OnOff                          'Just making sure PSU stays on!
 if Bat1_Chrg = 1 then Batt_Trickle     'is battery  already charged?
goto Batt_Chrg

Batt_Trickle:
' debug "trickle"
 gosub Chk_Pwr
 let D1 = 0
 let D0 = 0
 gosub Convert                          'Get battery charging voltage
 let Batt_V = AD
 PWM  Chrg_Limit,Trickle_LVL,1000       'turn on Trickle current
 low Chrg_Limit                         'trickle to minimum current
goto Main_Batt1

Chk_Pwr:
 let D1 = 1
 let D0 = 0
 gosub Convert                          'Get battery charging voltage
' debug AD, Batt_V
 if AD < Batt_V then No_Power
return

No_Power:
'debug "no_pwr"
 pause 50
 let Bat1_Chrg = 0                      'Indicate battery has been discharged
 PWM  Chrg_Limit,0,50           'turn off charge current
goto Main_Batt1

' ========================================================
' Battery charger program
' ========================================================

Batt_Chrg:
 Let Chrg_Time = 0                      'Initialize charge timer (counts in sec.)
Chrg_Lp:
 gosub Chk_Pwr
 for TCnt = 0 to 1
  PWM  Chrg_Limit,AO1_LVL,1000                  'first apply charge current then
                                                'check for charge termination
                                                '1. on max cell voltage
                                                '2. on time
                                                '3. -ve delta V
                                                '4. cell temperature
 next TCnt
 let D1 = 0
 let D0 = 0
 gosub Convert                          'Get battery charging voltage
 let Batt_V = AD
' debug "charge"
 if AD > BattV_Max then Batt_Chrg_Term
 Chrg_Time = Chrg_Time + 1
 if Chrg_Time > Chrg_Time_Max then Batt_Chrg_Term 'Used maximum charge time
 if AD < Batt_Peak then Batt_DeltaV
  Let Batt_Peak = AD                    ' Save peak value
Batt_DeltaV:
 Let AD = AD + Neg_DeltaV
 if AD < Batt_Peak then Batt_Chrg_Term  ' Detected negative deltaV in battery pack

'**** Insert battery pak termperature code here****
 goto Chrg_Lp                           ' Continue until charging terminated

Batt_Chrg_Term:
 PWM  Chrg_Limit,0,50                   'turn off charge current
 let Bat1_Chrg = 1                      'Indicate battery has been charged
goto Main_Batt1

Convert:

' ========================================================
' ADC Interface Pins
' ========================================================
' The LTC1594 uses a four-pin interface, consisting of chip-select, clock,
' data input, and data output. In this application, we tie the data lines
' together and connect to the PM104 pin designated DIO.
' Here's where the conversion occurs. The PM104 first sends the setup
' bits to the LTC1594, then clocks in two bits followed by (sample time),
' one null bit (a dummy bit that always reads 0, followed by the conversion data.

 high CS1                       ' Deactivate the ADC to begin.
 high CLK                       ' Clock data on rising edge, so start with CLK high
 high DIO
 pulsout CLK,2
 low DIO
 pulsout CLK,2
 let DIOp = D1                  ' next bit of command
 pulsout CLK,2
 let DIOp = D0                          ' next bit of command
 pulsout CLK,2                          ' command requesting A/D channel completed
 low CS1                                ' Activate the LTC1594
 input DIO                              ' Get ready for input from LTC1594

 pulsout CLK,2
 input DIO                              ' Dummy statement for delay
 pulsout CLK,2                          ' Sampling requires two clocks

 let AD = 0                             ' Clear old ADC result.
 for ADbits = 1 to 13           ' Get null bit + 12 data bits.
  pulsout CLK,2                         ' Clock next data bit in.
  let AD = AD * 2 + DIOp                ' Shift AD left, add new data bit.
 next Adbits                            ' Get next data bit.
 high CS1                               ' Turn off the ADC
return                                  ' Return to program.
```