$5

PC-60
millivolt signal conditioning card

&

PC-61
milliamp signal conditioning card

User manual

PC-60 & PC-61 MILLIVOLT & MILLIAMP SIGNAL CONDITIONING CARDS FOR IBM-PC/XT & COMPATIBLES

Contents:

Introduction

The PC-60 (millivolt) and PC-61 (milliamp) signal conditioning cards are computer control cards suitable for IBM-PC/XT and compatible machines. Both cards support a 12-bit A/D converter with +/–1 bit linearity. The PC-60 has 16 differential input lines, while the PC-61 has 8. The CPU can use any I/O channel for communication via the 8255 PPI, as selected by software. These cards are eminently suitable for industrial control systems.

Specifications

Built-in 12-bit A/D converter with +/–1 bit linearity
A/D transfer speed: 30Hz
Dual slope integrated mode A/D transfer
User-selectable port address
Multiple cards can be used in parallel
Interference rejection: 40dBm
Auto zero adjustment
Automatic signal polarity selection
Polling mode signal request
256 possible addresses (0-3FCH)
Each card requires four addresses
All address, data & control signals are TTL compatible
Power requirements: +/–5V DC at 200mA, and +/–12V DC at 200mA
Operating temperature: 0 deg. C to + 55 deg. C
Relative humidity: 0% to 90%

PC-60 specifications

16 differential input lines prevent interference
Input signal accuracy: 1mV
Input voltage overrange: –6.2V <Vin<+6.2V
Differential voltage between any two channels: < = 6.2V DC
Full scale input voltage: 0 – 4.096 V
PCB size: 165.2 x 99.3 x 13.4 mm
Input connector: 37-pin D-type

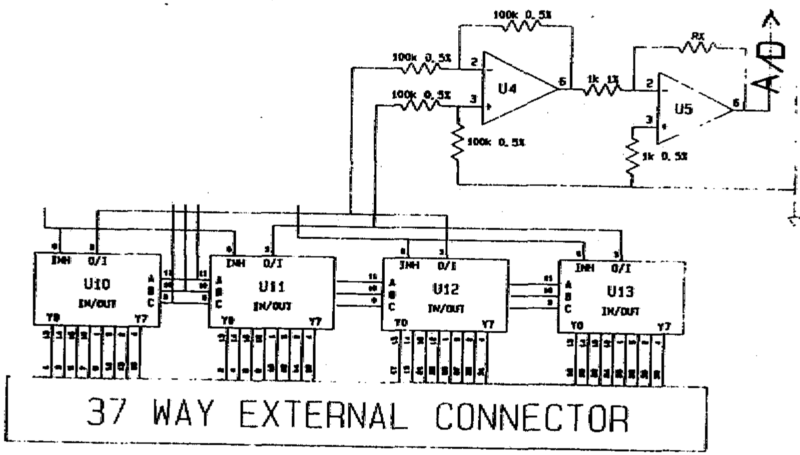
PC-61 specifications

Input signal range: 0-20mA
Input voltage overrange: +/–5V
8 differential input lines
PCB size: 161.3 x 99.3 x 13.4 mm
Input connector: 25-pin D-type

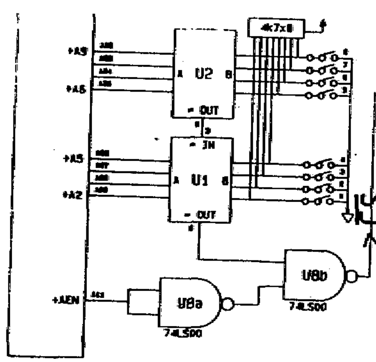
CIRCUIT PRINCIPLES

PC-60 mV card:

The 16-channel analogue switch circuit is composed of four CD4051 I.C.s, and can detect AC signals of 0 – 5V. Vcc is +6.2V, and Vee is –6.2V. Input resistors have a very low leakage value when the channels are off. A signal source can be connected directly to the circuit, but maximum voltage must not exceed 6.2V, or the analogue multiplexer CD4051 will break down.

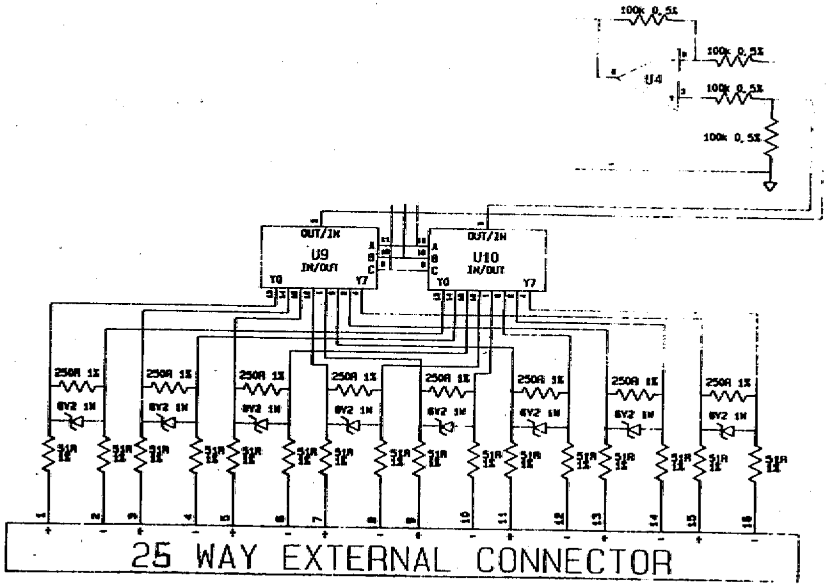


Address decoding is performed by two 74LS85 digital comparators. The addresses are set by DIP switch. When A2 – A9 match, the chip select signals enable the 8255 PPI. The address decoding circuit is shown below.

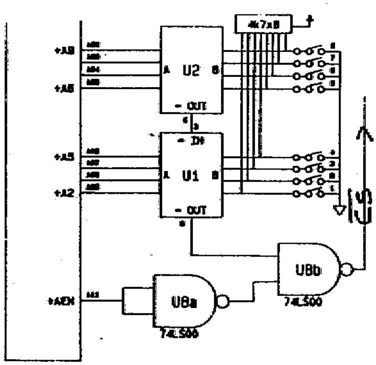


PC-61 mA Card:

The analogue selection switch for the PC-61 is composed of two CD4051 I.C.s. The circuit converts 0-20mA signals to 1 – 5V DC. Channel selection is software-controlled by the CPU via 3 bits of the 8255 PPI port A.



Address decoding is done by two 74LS85 digital comparators, and the circuit is shown below. When the address selection jumpers have been set, there will be a positive output from the second comparator (A=B), and the chip select signal will be enabled.



The 8255 PPI

The A/D converter communicates with the CPU via the 8255 PPI, which has three ports (A, B and C).

Port A is the output port: its address is the card address + 0.

b7	b6	b5	b4	b3	b2	b1	b0
				Analogue channel select			

Port B (address: card address + 1) is the input port.

b7	b6	b5	b4	b3	b2	b1	b0

When port C b4 equals 0, then port B b0 – b7 is the low order byte of converted data, and when port C b5 equals 0, then port B b0 – b4 is the high order byte of converted data. Port B b6 is the overflow status bit, and b7 is the polarity status bit. Port C b0 is the A/D status bit.

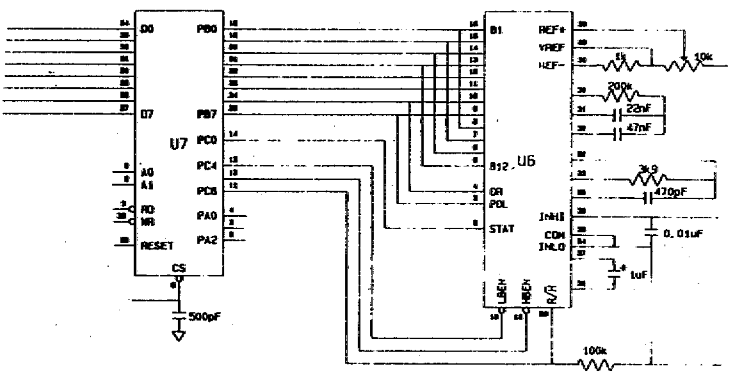
Port C (card address + 2) is the output port for control of the A/D converter.

The command port is the 8255 PPI control port (card address + 3).

A/D converter (ICL7109) control mode

1] Let R/$\overline{H}$ be logic 1.
2] Check status bit from 0 to 1.
3] If status bit moves from Low to High, then set R/H Low.
4] Let $\overline{LBEN}$ be Low, and read low byte data (DB0 – DB7).
5] Let $\overline{HBEN}$ be Low and read high byte data (DB8 – DB11, X,X,OF,POL)[1].
6] End.

[1] X = don't care
OF = overflow
POL = polarity



Address selection

The 8-way DIP switch corresponds to the address lines as follow:

ADDR LINE	A9	A8	A7	A6	A5	A4	A3	A2
DIP SWITCH	8	7	6	5	4	3	2	1

If a switch is ON, the line will be a logic 0. When a switch is OFF, its corresponding line is at logic 1. For example, if it is desired to set the port address at 0380H-0383H, then switches 1,2,3,4 and 5 should be on, but switches 6, 7 and 8 should be off.

Note: The IBM-PC/XT specification requires that the address of the I/O slot is not between 0000H and 01FAH inclusive, and A9 must always be at logic 1 (off).

SWITCH ADDRESSES ON PC-60/61

Switch OFF = logic 1

SW1 adds 4 bytes to address
SW2 adds 8 bytes to address
SW3 adds 16 bytes to address (10H)
SW4 adds 32 bytes to address (20H)
SW5 adds 64 bytes to address (40H)
SW6 adds 128 bytes to address (80H)
SW7 adds 256 bytes to address
SW8 must be off

Addresses must also be altered in software

PC-60 millivolt card pin assignment table

Pin no.	Signal flow	Description
1	in	Channel 0 negative
2	in	Channel 0 positive
3	in	Channel 1 negative
4	in	Channel 1 positive
5	in	Channel 2 negative
6	in	Channel 2 positive
7	in	Channel 3 negative
8	in	Channel 3 positive
9	in	Channel 4 negative
10	in	Channel 4 positive
11	in	Channel 5 negative
12	in	Channel 5 positive
13	in	Channel 6 negative
14	in	Channel 6 positive
15	in	Channel 7 negative
16	in	Channel 7 positive
17	in	Channel 8 negative
18	in	Channel 8 positive
19	in	Channel 9 negative
20	in	Channel 9 positive
21	in	Channel 10 negative
22	in	Channel 10 positive
23	in	Channel 11 negative
24	in	Channel 11 positive
25	in	Channel 12 negative
26	in	Channel 12 positive
27	in	Channel 13 negative
28	in	Channel 13 positive
29	in	Channel 14 negative
30	in	Channel 14 positive
31	in	Channel 15 negative
32	in	Channel 15 positive

PC-61 (mA) card 25-pin D-connector pin assignment table:-

PIN NO.	SIGNAL FLOW	DESCRIPTION
1	IN	CH0 ANALOG I/P POSITIVE END
3	IN	CH1 ANALOG I/P POSITIVE END
5	IN	CH2 ANALOG I/P POSITIVE END
7	IN	CH3 ANALOG I/P POSITIVE END
9	IN	CH4 ANALOG I/P POSITIVE END
11	IN	CH5 ANALOG I/P POSITIVE END
13	IN	CH6 ANALOG I/P POSITIVE END
15	IN	CH7 ANALOG I/P POSITIVE END
2	IN	CH0 ANALOG I/P NEGATIVE END
4	IN	CH1 ANALOG I/P NEGATIVE END
6	IN	CH2 ANALOG I/P NEGATIVE END
8	IN	CH3 ANALOG I/P NEGATIVE END
10	IN	CH4 ANALOG I/P NEGATIVE END
12	IN	CH5 ANALOG I/P NEGATIVE END
14	IN	CH6 ANALOG I/P NEGATIVE END
16	IN	CH7 ANALOG I/P NEGATIVE END

Unused pins are grounded.

A/D full scale voltage adjustment

The transfer characteristic of the PC-60 and PC-61 is 1mV/bit at the factory. VR1 and VR2 can be adjusted to change full scale voltage on both cards as required.

Examples

PC-60mV card:
Program name: MVTEST.COM
Source program: MVTEST.PAS

1. Set the PC-60's address to the range 330H-333H (switches 1,2, 5 and 6 on, other switches off).
2. With the PC power off, plug the PC-60 into an expansion slot, and switch the PC power on.
3. Apply an input of 0-5V to channel 0 (pins 1 and 2 of the 37-pin connector).
4. Run MVTEST.COM. The CRT will display the input voltage, which can be checked with a DVM. Compare the input voltage to the displayed value, and adjust VR1 and VR2 to equalize input and display.
5. Pressing any key will exit the program, and return to system.

PC-61mA card:
Program name: MATEST.COM
Source program: MATEST.PAS

1. Set the PC-61's address to the range 330H-333H (switches 1,2, 5 and 6 on, other switches off).
2. Ensuring that the PC power is off, plug the PC-61 into an expansion slot, and turn on the PC power supply.
3. Input a 4–20mA signal to channel 1 (pins 1 and 2 of the 25-pin connector).
4. Run MATEST.COM. The CRT will display the input voltage value.
5. Pressing any key will exit the program, and return to system.

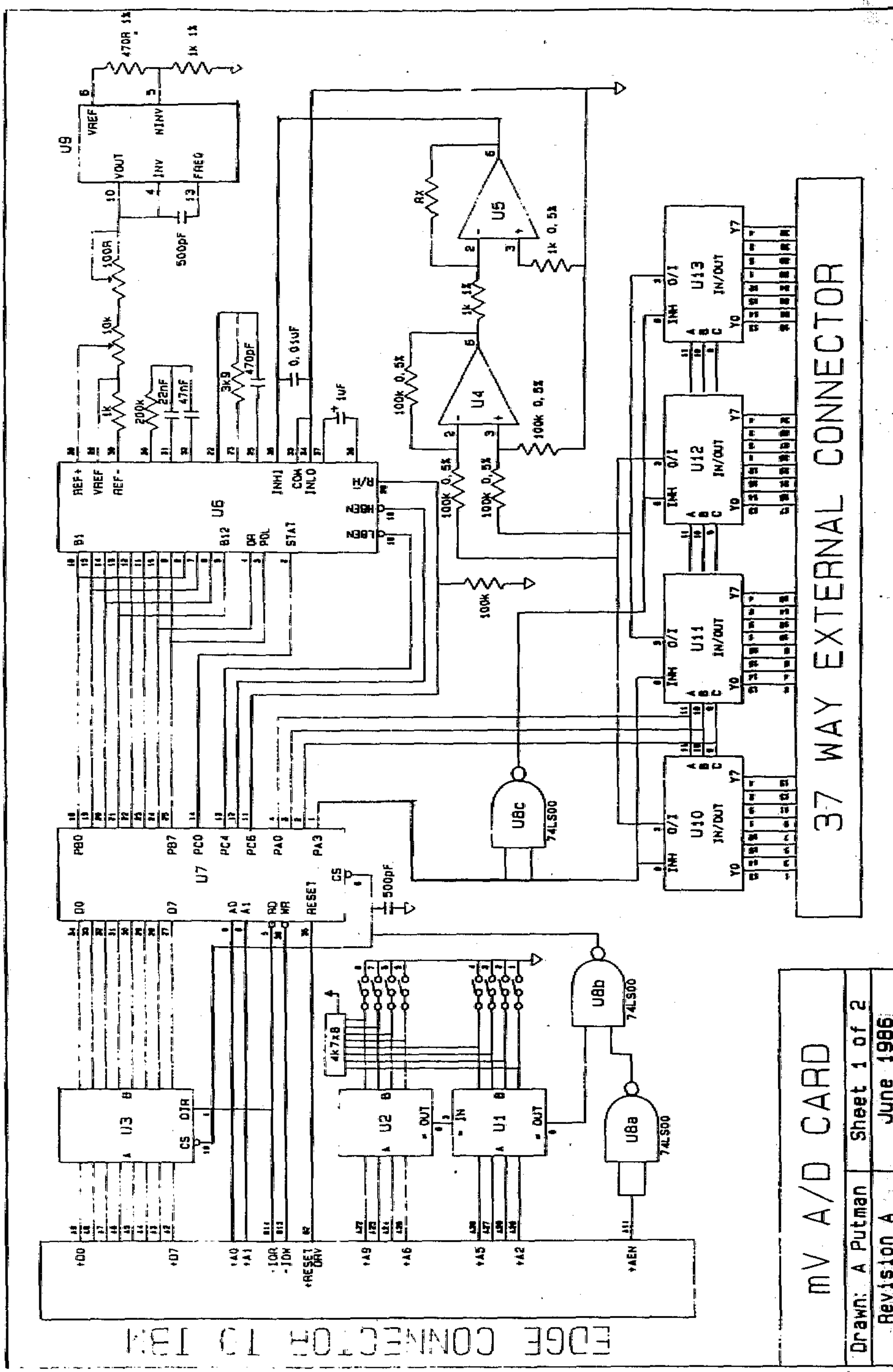
MVTEST.PAS PC-60

```
{$C-}
program mV_Card_Test;

const
  mV_Card = $330;
  Gain = 1;
  number_of_channels = 16;

var i, j, k : integer;
    line, column : integer;
    volts    : real;
    ch       : char;

procedure initialise;
  begin
    clrscr;
    gotoxy(20, 1); write('mV Card (PC-60) test program (V1.2)');
    port[mV_Card + 3] := $83;
    port[mV_Card + 2] := $FF;
    port[mV_Card + 0] := 0
  end {initialise};

procedure cursor_off;
  begin
    port[$3B4] := 10;
    port[$3D4] := 10;
    port[$3B5] := 13;
    port[$3D5] := 13;
  end {cursor_off};

procedure cursor_on;
  begin
    port[$3B4] := 10;
    port[$3D4] := 10;
    port[$3B5] := 11;
    port[$3D5] := 6;
  end {cursor_on};

procedure write_headings;
  begin
    gotoxy(1, 5); write('Setup Procedure: ');
    write('Switches 1, 2, 5 and 6 to be on, the others off.'); gotoxy(18, 6);
    write('Adjust the trimpots until the voltage between pins 39 and 36');
    gotoxy(18, 7); write('of U6 (ICL 7109) is 2.048 volts.');
    gotoxy(10, 10); write('Channel     Voltage(V)');
    gotoxy(45, 10); write('Channel     Voltage(V)');
    gotoxy(1, 22); write('Press any Key to exit');
  end {write_headings};
```

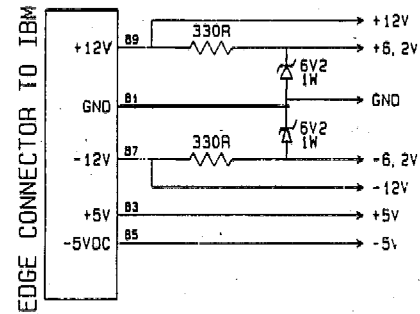
```
function Read_Voltage : integer;
  var temp : integer;
      neg  : boolean;
  begin
    port[mV_Card + 2] := $FF;
    repeat until (port[mV_Card + 2] and 1) = 1;
    port[mV_Card + 2] := $3F;
    repeat until (port[mV_Card + 2] and 1) = 0;
    port[mV_Card + 2] := $1F;
    temp := port[mV_Card + 1] and $4F;
    neg := (port[mV_Card + 1] and $80) = $80;
    port[mV_Card + 2] := $2F;
    temp := 256 * temp + port[mV_Card + 1];
    port[mV_Card + 2] := $3F;
    if neg then temp := -temp;
    Read_Voltage := temp
  end { Read_Voltage };

begin
  initialise;
  cursor_off;
  write_headings;
  repeat
    for j := 1 to number_of_channels do begin
      port[mV_Card + 0] := j-1;
      line := 11 + j; column := 13;
      if j > number_of_channels div 2 then begin
        line := line - (number_of_channels div 2);
        column := 48;
      end {if};
      gotoxy(column, line); write(j:2, '        ');
      volts := Read_Voltage/Gain/1000;
      if abs(volts) > 4096/Gain/1000 then
        write('Overflow')
      else begin
        write(volts:7:3, ' ')
      end { if };
    end { for };
  until Keypressed;
  read(KBD, ch);
  gotoxy(1, 22); write('Test terminated normally');
  cursor_on
end.
```

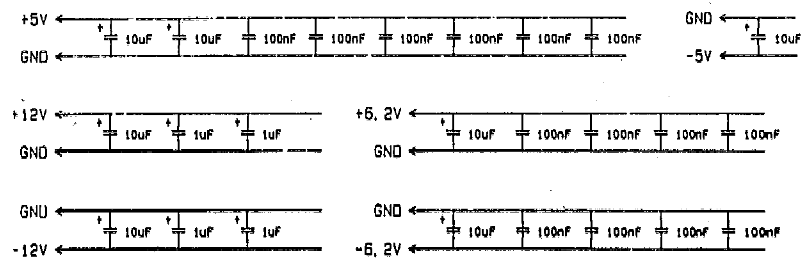
```
{$C-}
program mA_Card_Test;

const
  mA_Card = $330;
  number_of_channels = 8;
  Gain = 0.68;
  Resistance = 270 {ohms};

var i, j, k     : integer;
    milliamps : real;
    ch        : char;

procedure initialise;
  begin
    clrscr;
    gotoxy(20, 1); write('mA Card (PC-61) test program (V1.3)');
    port[mA_Card + 3] := $83;
    port[mA_Card + 2] := $FF;
    port[mA_Card + 0] := 0
  end {initialise};

procedure cursor_off;
  begin
    port[$3B4] := 10;
    port[$3D4] := 10;
    port[$3B5] := 13;
    port[$3D5] := 13
  end {cursor_off};

procedure cursor_on;
  begin
    port[$3B4] := 10;
    port[$3D4] := 10;
    port[$3B5] := 11;
    port[$3D5] := 6
  end {cursor_on};

procedure write_headings;
  begin
    gotoxy(1, 5);  write('Setup procedure: ');
    write('Switches 1, 2, 5 and 6 must be on, the others off.'); gotoxy(18, 6);
    write('Adjust the trimpots until the voltage between pins 39 and 36');
    gotoxy(18, 7); write('of U6 (ICL7109) is 2.048 volts.');
    gotoxy(20, 10); writeln('Channel          Current(mA)');
    gotoxy(1, 22); write('Press any key to exit');
  end {write_headings};
```

```
function Read_Voltage : integer;
  var temp : integer;
      neg  : boolean;
  begin
    port[mA_Card + 2] := $FF;
    repeat until (port[mA_Card + 2] and 1) = 1;
    port[mA_Card + 2] := $3F;
    repeat until (port[mA_Card + 2] and 1) = 0;
    port[mA_Card + 2] := $1F;
    temp := port[mA_Card + 1] and $4F;
    neg := (port[mA_Card + 1] and $80) = $80;
    port[mA_Card + 2] := $2F;
    temp := 256 * temp + port[mA_Card + 1];
    port[mA_Card + 2] := $3F;
    if neg then temp := -temp;
    Read_Voltage := temp
  end { Read_Voltage };

begin
  initialise;
  cursor_off;
  write_headings;
  repeat
    for j := 1 to number_of_channels do begin
      port[mA_Card + 0] := j - 1;
      gotoxy(22, 11+j); write(j:2);
      milliamps := (Read_Voltage/Gain)/Resistance;
      gotoxy(37, 11+j);
      if abs(milliamps) > (4096/Gain/Resistance) then
        write('Overflow')
      else begin
        write(milliamps:7:3, ' ')
      end { if };
    end { for }
  until Keypressed;
  cursor_on;
  gotoxy(1, 22); write('Test terminated normally');
  read(KBD, ch)
end.
```


mV A/D CARD
Drawn: A Putman
Sheet 1 of 2
June 1986
Revision A
37 WAY EXTERNAL CONNECTOR
EDGE CONNECTOR TO IBM
U1
U2
U3
U4
U5
U6
U7
U8a
U8b
U8c
74LS00
U9
U10
U11
U12
U13
IN/OUT
4k7x8
500pF
100k 0,5%
1k 1%
470R 1%
100R
10k
200k
22nF
47nF
3k9
470pF
0,01uF
1uF
Rx
1k 0,5%
+D0
+D7
+A0
+A1
-IOR
-IOW
+RESET DRV
+A9
+A6
+A5
+A2
+AEN

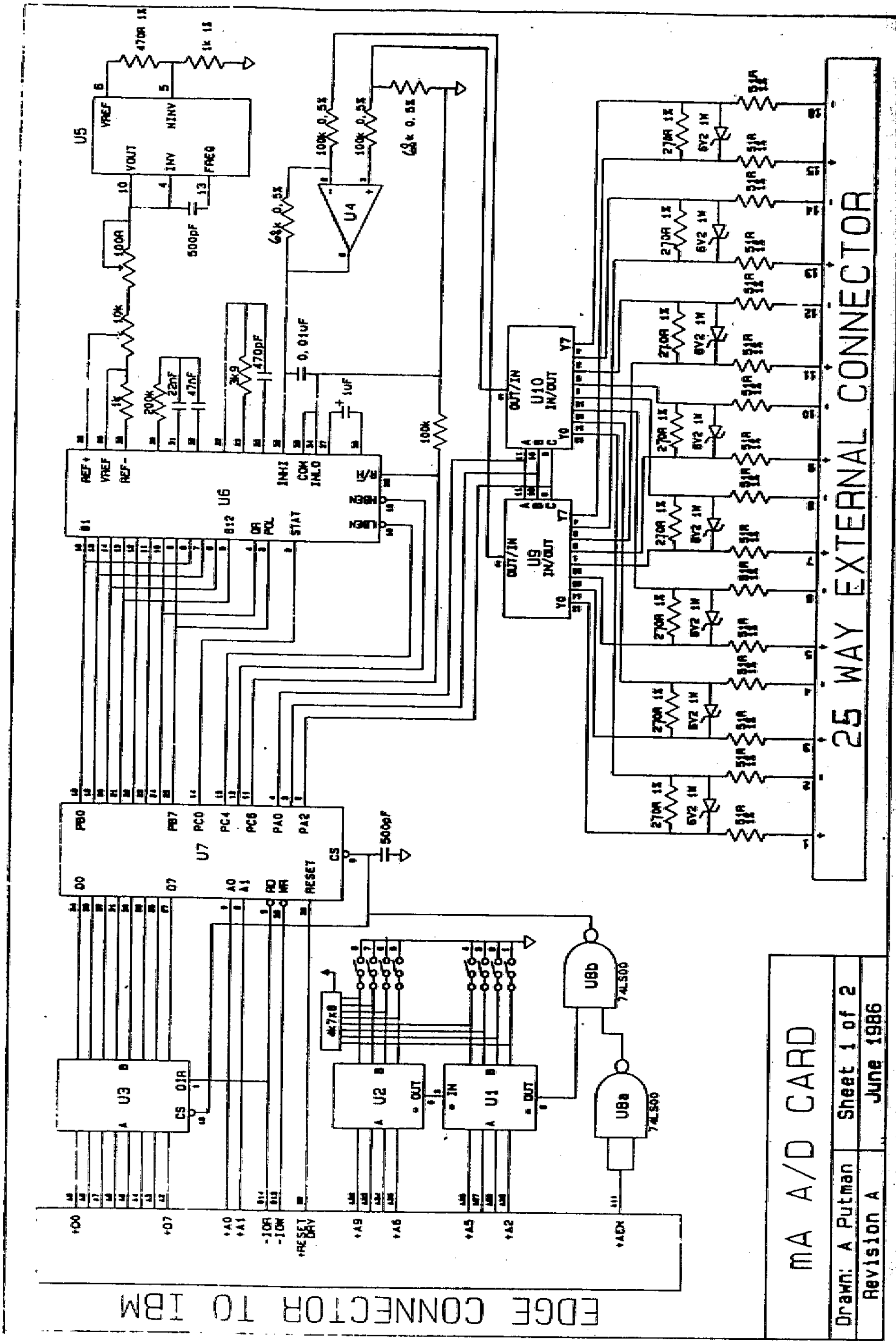
IC#	IC TYPE	-12V	-6, 2V	-5V	GND	+5V	+6, 2V	+12V
1	74LS85	-	-	-	2, 4, 8	16	-	-
2	74LS85	-	-	-	2, 4, 8	3, 16	-	-
3	74LS04	-	-	-	10	20	-	-
4	OP-07	4	-	-	-	-	-	7
5	OP-07	4	-	-	-	-	-	7
6	ICL7109	-	-	28	1, 20, 21	17, 40	-	-
7	8255A-5	-	-	-	7	26	-	-
8	74LS00	-	-	-	7	14	-	-
9	UA723	-	-	-	7	-	-	11, 12
10	CD4051	-	7	-	8	-	16	-
11	CD4051	-	7	-	8	-	16	-
12	CD4051	-	7	-	8	-	16	-
13	CD4051	-	7	-	8	-	16	-





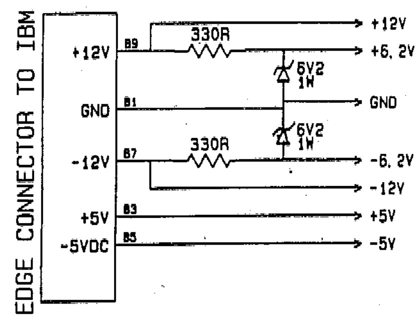
mV A/D CARD

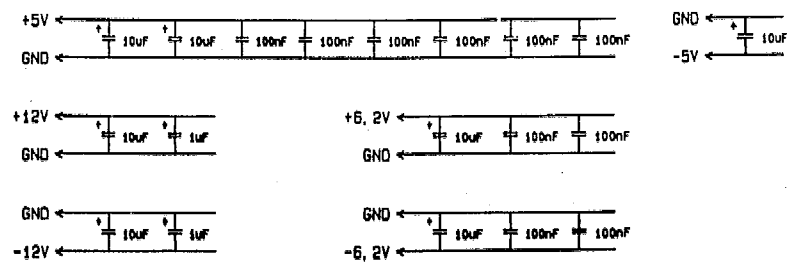
Drawn: A Putman	Sheet 2 of 2
Revision A	June 1986



25 WAY EXTERNAL CONNECTOR
EDGE CONNECTOR TO IBM
U1
U2
U3
U4
U5
U6
U7
U8a
U8b
U9
U10
74LS00
4k7x8
MA A/D CARD
Sheet 1 of 2
June 1986
Drawn: A Putman
Revision A

IC#	IC TYPE	-12V	-6, 2V	-5V	GND	+5V	+6, 2V	+12V
1	74LS85	-	-	-	2, 4, 8	16	-	-
2	74LS85	-	-	-	2, 4, 8	3, 16	-	-
3	74LS04	-	-	-	10	20	-	-
4	OP-07	4	-	-	-	-	-	7
5	UA723	-	-	-	7	-	-	11, 12
6	ICL7109	-	-	28	1, 20, 21	17, 40	-	-
7	8255A-5	-	-	-	7	26	-	-
8	74LS00	-	-	-	7	14	-	-
9	CD4051	-	7	-	8	-	16	-
10	CD4051	-	7	-	8	-	16	-





mA A/D CARD

Drawn: A Putman	Sheet 2 of 2
Revision A	June 1986