California AHMCT Program
University of California at Davis
California Department of Transportation

DESCRIPTION AND OPERATIONAL INSTRUCTIONS OF THE DEMO TMRR*

Daehie Hong
Keith J. Mueller
James M. Stiles
Steven A. Velinsky

AHMCT Research Report
UCD-ARR-97-01-31-01

Final Report of Contract
IA 65X875 T.O. 95-10

January 31, 1997

* This work was supported by California Department of Transportation (Caltrans) Contract Number IA 65X875 T.O. 95-10 through the Advanced Highway Maintenance and Construction Technology Research Center at the University of California at Davis.

ABSTRACT

The Tethered Mobile Routing Robot (TMRR) has been developed as a general crack sealing machine that will sense, rout, and seal both longitudinal and transverse cracks in a given workspace located behind a support vehicle. This robot has been developed by the Advanced Highway Maintenance & Construction Technology Research Center at the University of California at Davis. The full sized TMRR weighs approximately 4400 N [1000 lb.] and requires power from a large support vehicle. Due to its size, it was deemed infeasible to transport the large TMRR to trade shows and conferences. A demo TMRR was constructed for the express purpose to demonstrate capabilities and special design features of the larger version. This report describes its mechanical and control structure, and also includes operational instructions.

EXECUTIVE SUMMARY

The purpose of the demo TMRR is to represent mechanical and control features of the full-sized TMRR. It is fabricated in the way that its appearance is as close to the full-sized TMRR as possible but in down-sized version. It is also designed to be easily transported, assembled, and disassembled. The demo TMRR unit consists of three major components; the robot itself, tether (cable), and interface board. The tether is the only necessary connection when it is assembled (or disassembled). The interface board can be modified depending on the display purposes. The toggle switches and LED's are provided on the side in order to shift the control modes and to display the robot's status. Also, the system is designed to make it possible to monitor the status of the host computer from a remote computer by connecting through a serial cable to the interface board.

This document includes the descriptions of the systems, operational instructions, and necessary software and manufacturers' specifications.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

DISCLAIMER/DISCLOSURE

"The research reported herein was performed as part of the Advanced Highway Maintenance and Construction Technology Program (AHMCT), within the Department of Mechanical and Aeronautical Engineering at the University of California, Davis and the Division of New Technology and Research of the California Department of Transportation. It is evolutionary and voluntary. It is a cooperative venture of local, state and federal governments and universities."

"The contents of this report reflect the views of the author(s) who is (are) responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the STATE OF CALIFORNIA or the FEDERAL HIGHWAY ADMINISTRATION and the UNIVERSITY OF CALIFORNIA. This report does not constitute a standard, specifications or regulation."

CHAPTER 1

INTRODUCTION

The Tethered Mobile Routing Robot (TMRR) is a prototype highway maintenance vehicle developed by the Advanced Highway Maintenance and Construction Technology Research Center at the University of California at Davis. It is designed to function as a general crack sealing machine that will sense, rout, and seal both longitudinal and transverse cracks in a given workspace located behind a support vehicle. However, the TMRR has some unique characteristics; it is very mobile and has accurate real-time positioning in a rugged environment. These particular features allow the TMRR to be used as a multipurpose robotic platform and not limited its use to just crack routing and sealing.

The full sized TMRR weighs approximately 4400 N [1000 lb.] and requires power from a large support vehicle. Due to its size, it was deemed infeasible to transport the large TMRR to trade shows and conferences. A demo TMRR was constructed for the purpose of demonstrating capabilities and special design features of the larger version. It also could be used for experiments to verify a new idea or control algorithm.

CHAPTER 2

DESCRIPTION

The Demo TMRR, shown in Figure 1, weighs approximately 194 N [44 lb.] and is 50.8cm by 50.8cm [20"x20"]. The c-channel outer frame and black "router box" were assembled much in the same fashion as that of the larger TMRR. The motors and drive wheels are centrally located with respect to the "router box" and similarly, the router cutting wheel. The reason for this is to illustrate the TMRR's ability to turn with a very small radius of curvature while following cracks. The mushroom switch is installed on the top plate of the robot to cut off power in the event of an emergency. Also, the switches are provided to manage the control modes and LEDs to indicate the robot's status.

In Figure 2, the Demo TMRR is connected via a tether (#4 in Figure 1) which supplies power and signals. The tether has enough slack to permit 360° rotation of not only the cart, but also the entire conduit arm. Power for the system is located beneath the base platform. The computer on top of the platform is used to initialize the TMR's own computer and for path planning. Communication, in this case, is also routed through the tether.

The platform has an interface circuit on the bottom side of its top plate as shown in Figure 3. The interface board provides access to the external world, that is, communication, power, and hand-held control device (joystick). The platform and the interface board may be modified for display purposes.

Figures 4 and 5 show the inside of the robot. The DC motors, their drivers, and a DC to DC converter are installed on the bottom of the robot. They are separated from the computer to avoid motor switching noise. Since the computer, DC to DC converter, and motor drivers generate heat, the bottom of the robot should be open to circulate air.

Figure 6 shows an overall picture of the signal and power wiring. Also, Figure 7 illustrates detailed wiring of the interface board. The numbers in this figure are consistent with the numbers in Figure 3.

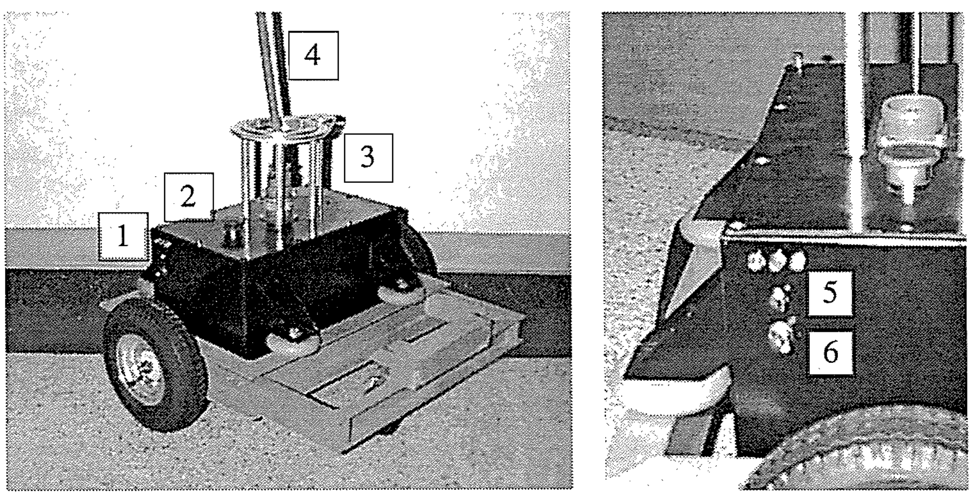


Figure 1 *The Demo TMRR - 1) Indicator lights - yellow, red, green from the front. 2) Emergency mushroom switch.. 3) Rotational sensor. 4) Tether. 5) Motor enable/disable switch. 6) Control mode switch*

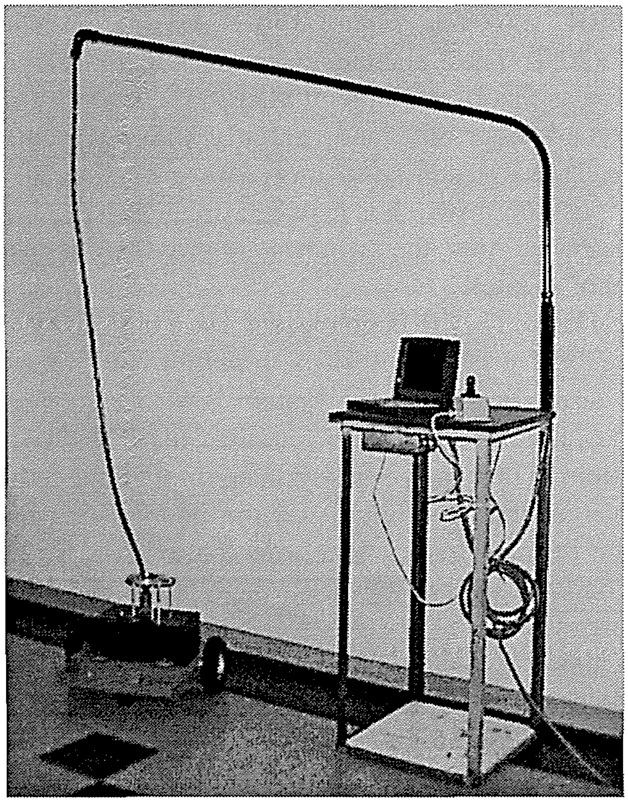


Figure 2 *The tether and base platform connected to the Demo TMRR*

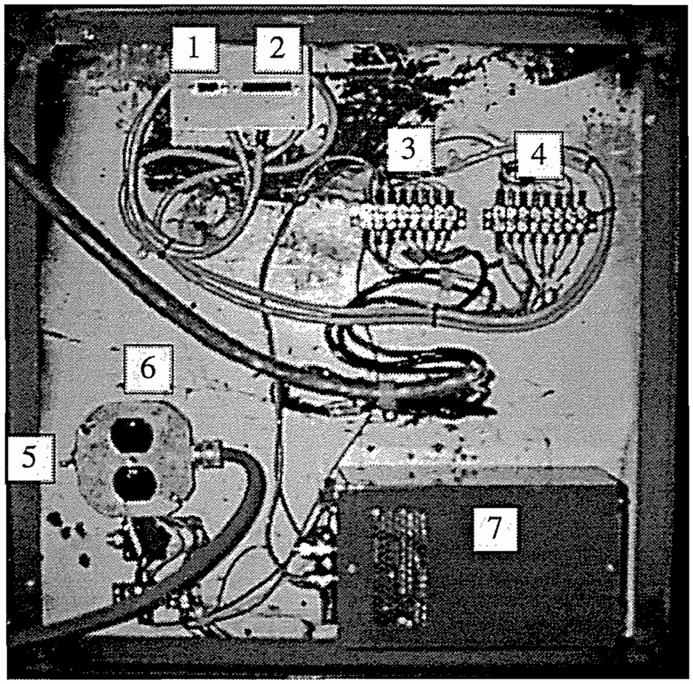


Figure 3 *Wiring layout on interface board. 1) DB9 connector for joystick. 2) DB25 connector for RS-232C serial communication. 3) Terminal block for joystick connection. 4) Terminal block for RS-232C connection. 5) Main Power switch. 6) Power outlet for external use. 7) DC power supplier.*

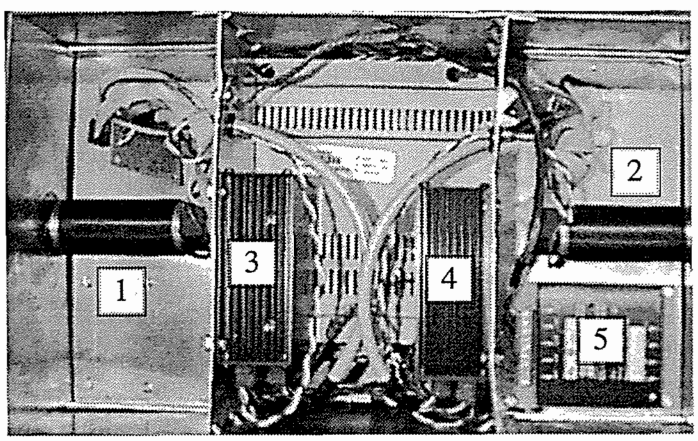


Figure 4 *Bottom view of demo TMRR. 1,2) DC motors. 3,4) Motor drives. 5) DC-DC converter.*

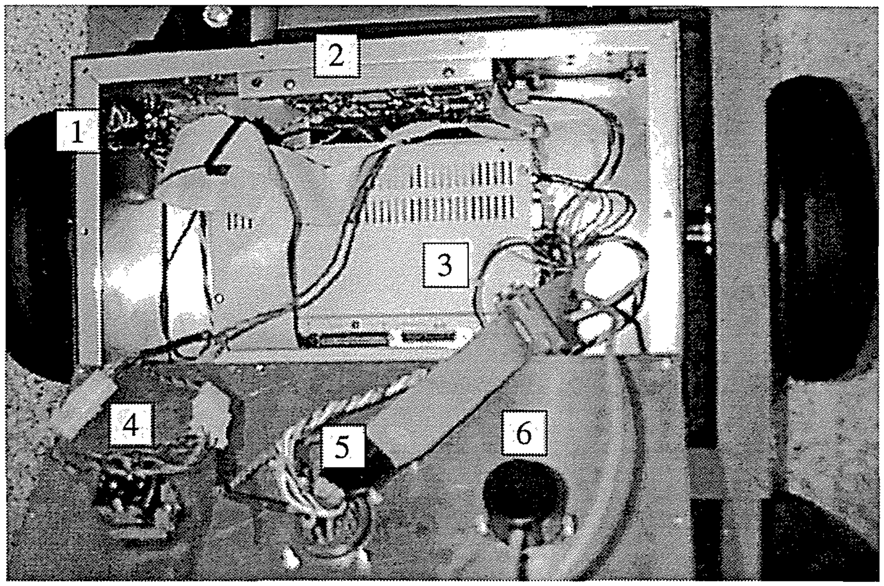


Figure 5 *Top inside view of demo TMRR. 1) Wiring for LED's, motor enable/disable switch, and control mode switch. 2) Hard drive. 3) Computer. 4) Emergency mushroom switch. 5) Amphenol connector. 6) Encoder for rotational sensor.*

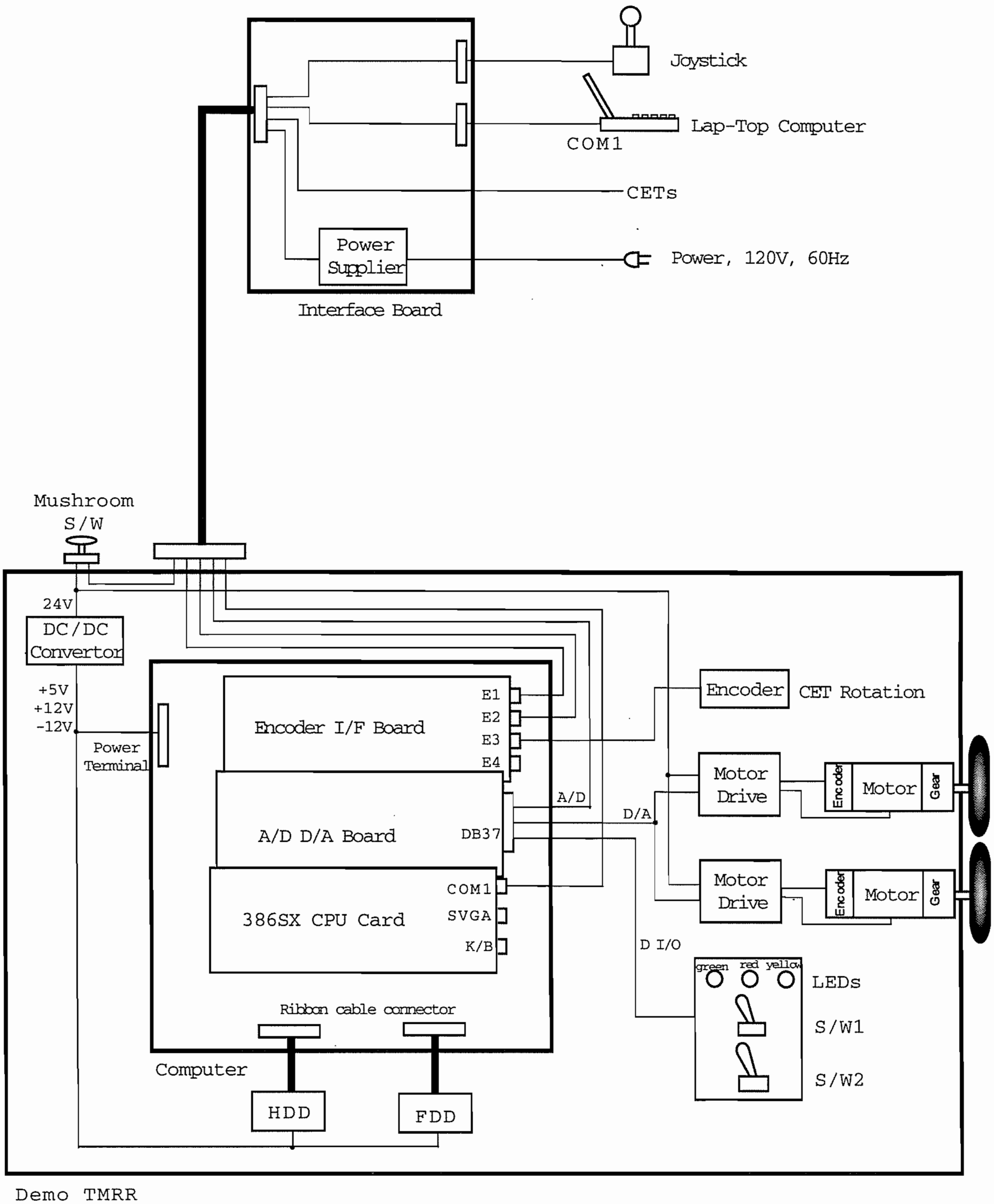


Figure 6 *Overall wiring diagram.*

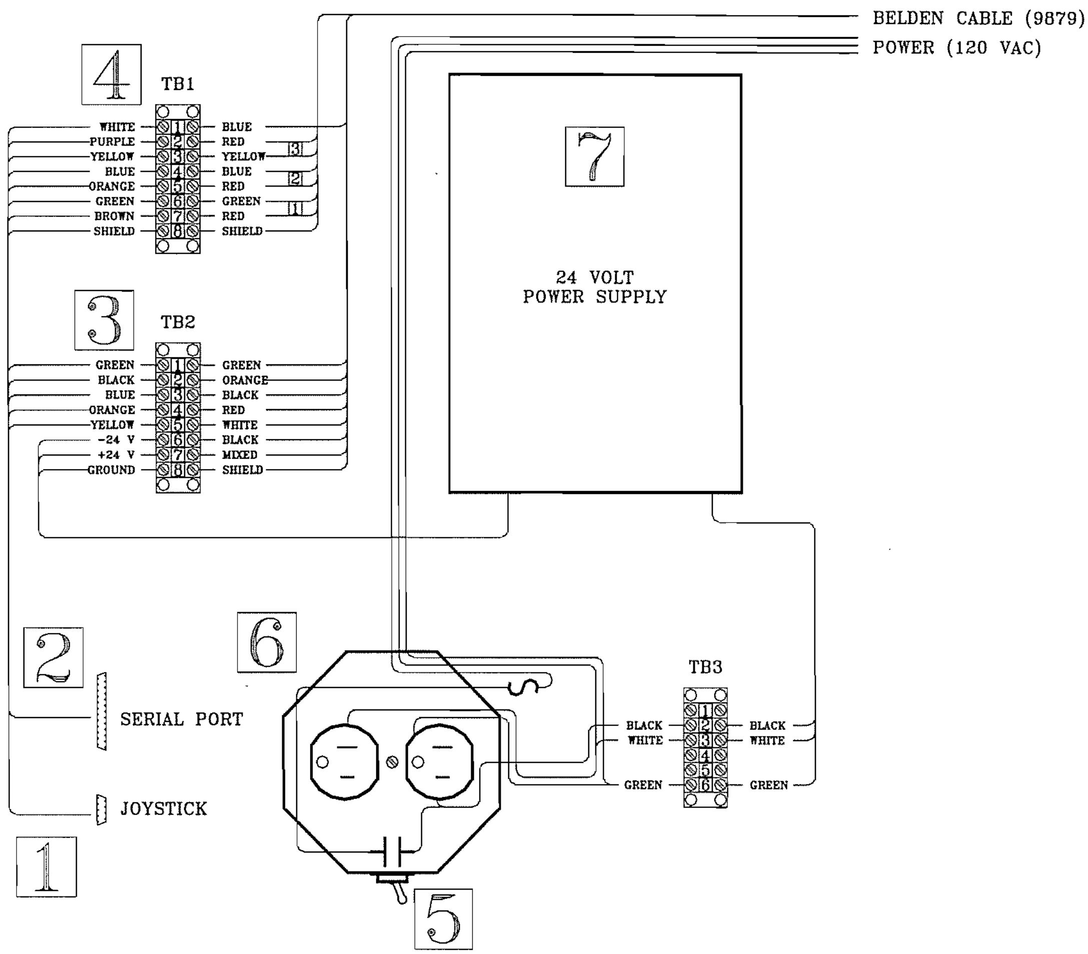


Figure 7 Detail wiring diagram of interface board

CHAPTER 3

OPERATING INSTRUCTIONS

When it has been determined where the Demo TMRR system is to be set up, the robot itself must be attached to its tether as seen in Figure 2. In order for the robot to work, the mushroom switch must be in its "pulled" position. The motor enable switch (#5, Figure 1) needs to be in the left position and the control mode switch (#6, Figure 1) in its neutral position.

The interface board must be plugged in and the power turned on in order for the Demo TMRR system to work. The computer must be plugged into the interface board at two places: at #6 in Figure 3 for power and at #2 in Figure 3 for serial communications. The joystick needs to be plugged in at #1 in Figure 3 as well. Once all of these connections have been made, the power may then be turned on for the interface board. The power switch for the system is located at #5 in Figure 3.

After turning on the computer, run the program "cremote" by double clicking on it twice. This executable file may be found in the folder similarly called cremote. The cremote software establishes two-way communication between the two computers. It is important to realize at this point that the computer on the robot (#3, Figure 5) is going to be the *host* computer and the computer hooked to the interface board is the *remote* computer.

After communication has been established a number of command prompts i.e. C:\DEMOTMR> will appear on the remote computer. At this point, in order to run the program, type "demotmr." Initially, the red green and yellow LEDs were lit, but after typing demotmr they should start to flash. After flashing 5 times, only the middle red LED should be lit. This is the DemoTMR's *neutral mode*.

In order to enable the *joystick mode,* first flip the motor enable switch to the right and then switch the mode switch to its right position. The yellow LED will be illuminated and the robot can then be controlled with the joystick. To disable the *joystick mode*, flip the mode switch to its middle position.

In order to enable the *path tracking mode*, switch the mode switch to the left. The green LED will be lit at this time. To disable, again flip the mode switch to the middle or neutral position. However, in order for the robot to move independently from the joystick, a path must first be prepared.

The program for learning a path is called "teach". It is a different executable program from the "demotmr" program. Before running "demotmr", type "teach" at the C:\DEMOTMR> prompt. You will then be asked to type an input file name, something along the lines of "path.dat" would suffice. Before pressing the return key after this filename, be sure that the robot is in its *neutral mode.* As soon as the return key is pressed, switch the motors on and enable the *joystick mode.* From this point on, every movement of the robot that is performed with the joystick will be recorded in a buffer file. In order to stop this recording, set the robot back into the *neutral mode* and exit the program by pressing the spacebar or any other key.

Looking back at the *path tracking mode* procedure above, once the mode switch is selected, the robot will follow the prescribed path that was programmed with the "teach" program. The robot will follow the path indefinitely with a number 1 appearing on the screen with every iteration. Care must be taken since certain paths may make the robot go beyond the boundary of its workspace as defined by the length of its tether. The path.dat file could be renamed to keep track of a number of different paths that the robot could follow, i.e. path.001, path.002, etc. However, "demotmr" will ONLY run the path that is currently named path.dat.

At any point during the execution of the "demotmr" program, the program itself can be stopped by pressing any key on the computer keyboard. Turning off the motor enable switch is another way of disabling the robot if it is traveling beyond a predefined workspace. Pressing the mushroom switch on the top of the robot freezes the entire system. Communication with the computer will first have to be re-established before running the "demotmr" program once again.

CHAPTER 4

COMPONENT LIST

Table 1. Component list

Name	Model	Manufacturer	E.t.c.
MicroBox IPC Chasis	MPBC-640	Advantech Co., Ltd.	3 slots, half size
Joystick	Induction Type Joystick Control	Maurey	Allied Electronics, Co.
Super VGA Piggyback Module	PCA-6442	Advantech Co., Ltd.	
Half-size All-in-one 386SX CPU Card	PCA-6134	Advantech Co., Ltd.	With Flash/ROM Disk
PC to Incremental Encoder Interface Card	PC7166	U.S. Digital Co.	4 ch.'s encoder i/f card
DC Motor	RE035-071-34EAD200A	Maxon Motor	
Gear Head	2932.702-0035.0-000	Maxon Motor	35:1 ratio
Encoder	3407-500	Maxon Motor	HEDS-5310-329F
Motor Drive	mmc-QR030024-02 LE00A	Maxon Motor	
A/D D/A board	CYDAS 8JR-AO	CyberResearch, Inc.	

APPENDIX A

DEMO TMRR CONTROL PROGRAM

```
/*********************************************************** */
/* DEMO TMR CONTROL PROGRAM                                  */
/*                                                           */
/* FILE NAME: DEMOTMR.C                                      */
/*                                                           */
/* REVISION: revised the followings for ITS show at Orlando 1996, */
/*      CET readings are disabled                            */
/*      Boundary check is disabled                           */
/*                                                           */
/* AUTHOR: Daehie Hong                                       */
/*********************************************************** */
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <dos.h>
#include <time.h>
#include <malloc.h>
#include <stdlib.h>
#include "linear.h"

// #define TEACH

#define BASE            0x300
#define ADC_LSB         BASE
#define ADC_MSB         BASE + 1
#define ADC_START       BASE + 1
#define ADC_CONTROL     BASE + 2
#define ADC_STATUS      BASE + 2
#define ADC_CH_0        0
#define ADC_CH_1        1
#define DAC_START       BASE + 3
#define DAC_LSB         BASE + 4
#define DAC_MSB         BASE + 5
#define DIO_REG         BASE + 3

/* addresses */
#define DEF_BASE        0X310  /* default base address of interface board */

#define DATA1           0       /* data register of LS7166 */
#define CONTROL1        1       /* control register of LS7166 */
#define DATA2           2       /* data register of LS7166 */
#define CONTROL2        3       /* control register of LS7166 */
#define DATA3           4       /* data register of LS7166 */
#define CONTROL3        5       /* control register of LS7166 */
#define DATA4           6       /* data register of LS7166 */
```

```
#define CONTROL4        7            /* control register of LS7166 */

#define LATCH           8            /* a read causes all LS7166s to latch counter */

/* LS7166 commands */
#define MASTER_RESET    0X20         /* master reset command */
#define INPUT_SETUP     0X68         /* command to setup counter input */
#define QUAD_X1         0XC1         /* command to setup quadrature multiplier to 1 */
#define QUAD_X2         0XC2         /* command to setup quadrature multiplier to 2 */
#define QUAD_X4         0XC3         /* command to setup quadrature multiplier to 4 */
#define ADDR_RESET      0X01         /* command to reset address pointer */
#define LATCH_CNTR      0X02         /* command to latch counter */
#define CNTR_RESET      0X04         /* command to reset counter */
#define PRESET_CTR      0X08         /* transfer preset to counter */

/*  Timer commands from cet.c */

#define TIMER_FREQ      1193180L
#define TIMER_COUNT     0x42
#define TIMER_MODE      0x43
#define TIMER_OSC       0xb6
#define OUT_8255        0x61
#define SPKRON          3

/* Workspace boundary */
#define x0              0
#define y0              6.74
#define x1              43.0
#define y1              6.7
#define x2              43.
#define y2              73.0
#define x3              -13.
#define y3              73.0
#define x4              -13.0
#define y4              38.0
#define x5              -43.0
#define y5              6.74


#define PI              3.141592654

/* Global variables */

int  base = DEF_BASE;          /* base address of interface board */
float t1[N_grid][N_grid];   /* theta1 representing the preset grid */
```

```
float N1d, N2d, N3d, N4d, N5d, N6d, N7d, N8d, N9d; /* coef.'s for Lagrange */
float Volt0_i,Volt1_i;

FILE *outfile;

void DA_convert(float, int);
float AD_convert(int);
void wait(unsigned int TimeDelay);
void joystick(int flag);
void init_joystick(void);
void init(void);
void hard_latch(void);
long read_position1(void);
long read_position2(void);
long read_position3(void);
void Robot_Posture(float *x, float *y, float *th);
void sound_on(unsigned freq);
void sound_off(void);
int get_xyt(float *L1, float *L2, float *x, float *y, float *th1);
void boundary_check(float xcheck, float ycheck, float thcheck, int *flag);
void LED_on(int id);
void LED_on_all(void);
void LED_on_13(void);
void LED_off(void);
void LED_flashing(int in);
int read_sw(void);

main()
{
   float      Volt0, Volt1;
   char       ch;
   clock_t    ticksnow;
   float      time1, time2;
   double     tused, dt1;
   float      x,y,th,th00;
   char       cc1;
   int        i,j,k,sw,count,flag, pflag=0;
   FILE       *fp_t1;             /* File where precalculated theta 1's exist */
   FILE       *infile;            /* File where preplanned path is */

#ifdef TEACH
   outfile = fopen("path.out","w");
#endif

/* reset D to A converter  */
```

```
   DA_convert(0,0);
   DA_convert(0,1);

/* flashing LED's to indicate program start */
   LED_flashing(5);

/* initialize CET                         */
//   LED_on_13();
//   while( read_sw() != 1 );       // sw:  0-neutral, 1-forward, 2-backward
//   while( read_sw() != 0 );       // LED: 2-neutral, 1-forward, 3-backward
   init();          /* Initialization routine */
   init_joystick();   /* get initial position of joystick  */
   LED_on(1);

//   if ((fp_t1 = fopen("t","r")) == NULL)
//   {
//     LED_on_all();
//     exit(0);
//   }

//   for (i=0 ; i < 151 ; i++)   /* Scanning in i-j array for theta 1 */
//     {
//       for (j=0 ; j < 151 ; j++ )
//        {
//          fscanf(fp_t1," %f",&t1[i][j]);
//        }
//     }

  /* Calculation of coefficients for Lagrange polynomial */

/*   N1d = (ksi1-ksi2)*(ksi1-ksi3)*(yeta1-yeta8)*(yeta1-yeta7);
  N2d = (ksi2-ksi1)*(ksi2-ksi3)*(yeta2-yeta9)*(yeta1-yeta6);
  N3d = (ksi3-ksi1)*(ksi3-ksi2)*(yeta3-yeta4)*(yeta3-yeta5);
  N4d = (ksi4-ksi8)*(ksi4-ksi9)*(yeta4-yeta3)*(yeta4-yeta5);
  N5d = (ksi5-ksi7)*(ksi5-ksi6)*(yeta5-yeta3)*(yeta5-yeta4);
  N6d = (ksi6-ksi7)*(ksi6-ksi5)*(yeta6-yeta2)*(yeta6-yeta9);
  N7d = (ksi7-ksi6)*(ksi7-ksi5)*(yeta7-yeta1)*(yeta7-yeta8);
  N8d = (ksi8-ksi9)*(ksi8-ksi4)*(yeta8-yeta7)*(yeta8-yeta1);
  N9d = (ksi9-ksi8)*(ksi9-ksi4)*(yeta9-yeta2)*(yeta9-yeta6);
*/

 flag = 1;


  time1 = (float) clock();
```

```
  while (((sw = read_sw()) < 3 ) && !kbhit()) {
          pflag = 0;
          if (sw == 0)     /* Center switch setting - neutral */
           {
            pflag = 1;
            LED_on(2);
            DA_convert(0,0);
            DA_convert(0,1);
//         if (kbhit()) exit(0);
           }

          if ( sw == 1 )        /* Joystick operation               */
           {
            LED_on(1);
            joystick(0);
//         hard_latch();
//         Robot_Posture(&x,&y,&th);
//         printf("\n%5.2f  %5.2f  ",x,y);
//         boundary_check(x,y,th,&flag);
            while( (time2 = (float) clock()) == time1);
            time1 = time2;
           }

          if ( sw == 2 )        /* Path following operation */
           {
            LED_on(3);
            if(pflag < 5) {
/* open path file and scan in data */
            infile = fopen("path.dat","r");
            while((fscanf(infile,"%f %f\n",&Volt0, &Volt1) != EOF)) {
                  DA_convert(Volt0,0);
                  DA_convert(Volt1,1);
                  if ( (read_sw()) != 2) break;
                  while( (time2 = (float) clock()) == time1);
                  time1 = time2;
                  if(kbhit()) {
                        DA_convert(0,0);
                        DA_convert(0,1);
                        fclose(infile);
//                exit(0);
                  }

             }
             fclose(infile);
             pflag++;
```

```
                printf("%d\n",pflag);
                }
              }
        }
#ifdef TEACH
        fclose(outfile);
#endif
}


/*****************************************************/
void DA_convert(float Volt, int ch_no)
/*****************************************************/
{
   unsigned    DAvalue, MSB, LSB, dummy;

   DAvalue = (unsigned) ((Volt + 5) * 409.5);
   MSB = (DAvalue / 256) & 0xf;
   LSB = DAvalue & 0xff;
   outp(DAC_LSB + 2 * ch_no, LSB);
   outp(DAC_MSB + 2 * ch_no, MSB);
   dummy = inp(DAC_START);
}

/*****************************************************/
float AD_convert(int ch_no)
/*****************************************************/
{
   unsigned ADC_value, MSB, LSB;
   int i;
   float Volt;

   outp(ADC_CONTROL, ch_no);
   outp(ADC_START, 0);
   for (i=0; i<10; i++);       // wait for a while to aviod over-running the ADC
   while( inp(ADC_STATUS) & 0x80 );
   MSB = inp(ADC_MSB) & 0xff;
   LSB = inp(ADC_LSB) & 0xf0;
   ADC_value = MSB * 16 + LSB / 16;
   Volt = ((float) ADC_value) / 409.5 - 5;
   return(Volt);
}

/*****************************************************/
void wait(unsigned int TimeDelay)
```

```
/*******************************************************/
{
  union REGS xr;
  unsigned int TimeDelayHigh, TimeDelayLow;

  TimeDelayHigh=(unsigned)(TimeDelay*1000./65536);
  TimeDelayLow=(unsigned)(TimeDelay*1000.-TimeDelayHigh*65536.);

  xr.h.ah=0x86;
  xr.x.cx=TimeDelayHigh;
  xr.x.dx=TimeDelayLow;
  int86(0x15,&xr,&xr);

}

/*******************************************************/
void init_joystick(void)
/*******************************************************/
{
  int i;

  Volt0_i = 0;
  Volt1_i = 0;

  for (i=0; i<30; i++) {
        Volt0_i = AD_convert(0) + Volt0_i;
        Volt1_i = AD_convert(1) + Volt1_i;
  }
  Volt0_i = Volt0_i / 30;
  Volt1_i = Volt1_i / 30;
}

/*******************************************************/
void joystick(int flag)
/*******************************************************/
{
  float Volt0, Volt1, Left, Right;

        Volt0 = AD_convert(0) - Volt0_i;
        Volt1 = AD_convert(1) - Volt1_i;
        if (fabs(Volt0) < 0.05) Volt0 = 0;
        if (fabs(Volt1) < 0.05) Volt1 = 0;

        Left = Volt1 * 2.0 + 1.8 * Volt0;
        Right = - Volt1 * 2.0 + 1.8 * Volt0;
```

```
        if (flag) {
           Left = Left * 0.2;
           Right = Right * 0.2;
        }

        if (Left > 5) Left = 5;
        if (Left < -5) Left = -5;
        if (Right > 5) Right = 5;
        if (Right < -5) Right = -5;

        DA_convert(Left, 0);
        DA_convert(Right, 1);
#ifdef TEACH
        fprintf(outfile,"%4.2f %4.2f\n",Left,Right);
#endif
}

void init(void) {                          /* initialize the LS7166s */
        outp(base + CONTROL1, MASTER_RESET);
        outp(base + CONTROL1, INPUT_SETUP);
        outp(base + CONTROL1, QUAD_X4);
        outp(base + CONTROL1, CNTR_RESET);

        outp(base + CONTROL2, MASTER_RESET);
        outp(base + CONTROL2, INPUT_SETUP);
        outp(base + CONTROL2, QUAD_X4);
        outp(base + CONTROL2, CNTR_RESET);

        outp(base + CONTROL3, MASTER_RESET);
        outp(base + CONTROL3, INPUT_SETUP);
        outp(base + CONTROL3, QUAD_X4);
        outp(base + CONTROL3, CNTR_RESET);

        outp(base + CONTROL4, MASTER_RESET);
        outp(base + CONTROL4, INPUT_SETUP);
        outp(base + CONTROL4, QUAD_X4);
        outp(base + CONTROL4, CNTR_RESET);
}

void hard_latch(void)    {     /* latch position with low pulse on pin 3 */
        outp(base + LATCH, 0);              /* toggle pin 3 of LS 7166 */
}

long read_position1(void) {                /* read position of encoder */
```

```
        long position;
        outp(base + CONTROL1, ADDR_RESET);      /* reset address pointer */
        position  = (long)inp(base + DATA1);    /* least significant byte */
        position += (long)inp(base + DATA1) << 8;
        position += (long)inp(base + DATA1) <<16; /* most significant byte */
        return position;
}

long read_position2(void) {               /* read position of encoder */
        long position;
        outp(base + CONTROL1, ADDR_RESET);      /* reset address pointer */
        position  = (long)inp(base + DATA2);    /* least significant byte */
        position += (long)inp(base + DATA2) << 8;
        position += (long)inp(base + DATA2) <<16; /* most significant byte */
        return position;
}

long read_position3(void) {               /* read position of encoder */
        long position;
        outp(base + CONTROL1, ADDR_RESET);      /* reset address pointer */
        position  = (long)inp(base + DATA3);    /* least significant byte */
        position += (long)inp(base + DATA3) << 8;
        position += (long)inp(base + DATA3) <<16; /* most significant byte */
        return position;
}

/***************************************************************************/
void    Robot_Posture(float *x, float *y, float *th)
/***************************************************************************/
{
   float   Sensitivity=400.24;            /* 100.06 pulses per inch */
   long    MaxEncoder=16777216; /* Maximum number of the 24 bit counter */
   long    a_cnt, b_cnt, c_cnt;                /* Counter values */
   float  IL1,IL2,xdist,ydist;
   float  L1, L2, thx, xt, yt, th0; /* Values to be calculated and outputted */

        IL1=16.262; IL2=16.142; /*1-3*/
        a_cnt = read_position1();              /* Reading counters */
        b_cnt = read_position2();
        c_cnt = read_position3();

        if ( a_cnt > MaxEncoder/2 ) L1 = - MaxEncoder + a_cnt;
        else L1 = a_cnt;
        L1 = L1/Sensitivity+IL1;
        if ( b_cnt > MaxEncoder/2 ) L2 = - MaxEncoder + b_cnt;
```

```
        else L2 = b_cnt;
        L2 = L2/Sensitivity+IL2;
        if ( c_cnt > MaxEncoder/2 ) thx = - MaxEncoder + c_cnt;
        else thx = c_cnt;
        get_xyt(&L1,&L2,&xt,&yt,&th0);
        *x = xt;
        *y = yt;
        *th = -(thx*360/10000)/3.416 - (th0*57.3 - 26.6);
//      printf("%ld %ld",a_cnt,b_cnt);

}

void sound_on(unsigned freq)
{
        unsigned status, ratio, part_ratio;

        status = inp(OUT_8255);
        outp(TIMER_MODE,TIMER_OSC);
        ratio = (unsigned)(TIMER_FREQ/freq);
        part_ratio = ratio & 0xff;
        outp(TIMER_COUNT,part_ratio);
        part_ratio = (ratio >> 8) & 0xff;
        outp(TIMER_COUNT,part_ratio);
        outp(OUT_8255,(status | SPKRON));
}

void sound_off(void)
{
        unsigned status;
        status = inp(OUT_8255);
        outp(OUT_8255,(status & ~SPKRON));
}

/*****************************************************************
*  A subroutine for quadratic lagrange interpolation
*  Arguments passed: L1, L2
*  Arguments returned: x, y, th1
*  Boolean values: 0 if impossible combination of L1, L2
*              1 if proper combination of L1, L2
*****************************************************************/

int get_xyt(float *L1, float *L2, float *x, float *y, float *th1)

{
  int m,n,b;
```

```
float ksi, yeta, Nx[9], Px[9], sinTx, cosTx;
extern float t1[N_grid][N_grid];

 /* L1 and L2 must first be checked - 2 sides of a triangle > third */

if ( *L1 + *L2 <= D )
  return(0);
else if ( *L1 + D <= *L2 )
  return(0);
else if ( *L2 + D <= *L1 )
  return(0);
else        /* If there is a proper case, interpolate theta1, x & y */
{
  m = (int) *L1;
  n = (int) *L2;
  ksi = *L1 - m;
  yeta = *L2 - n;

                /* Lagrange polynomial used for quadratic interpolation */

  Nx[0] = (ksi-ksi2)*(ksi-ksi3)*(yeta-yeta8)*(yeta-yeta7)/N1d;
  Nx[1] = (ksi-ksi1)*(ksi-ksi3)*(yeta-yeta9)*(yeta-yeta6)/N2d;
  Nx[2] = (ksi-ksi1)*(ksi-ksi2)*(yeta-yeta4)*(yeta-yeta5)/N3d;
  Nx[3] = (ksi-ksi8)*(ksi-ksi9)*(yeta-yeta3)*(yeta-yeta5)/N4d;
  Nx[4] = (ksi-ksi7)*(ksi-ksi6)*(yeta-yeta3)*(yeta-yeta4)/N5d;
  Nx[5] = (ksi-ksi7)*(ksi-ksi5)*(yeta-yeta2)*(yeta-yeta9)/N6d;
  Nx[6] = (ksi-ksi6)*(ksi-ksi5)*(yeta-yeta1)*(yeta-yeta8)/N7d;
  Nx[7] = (ksi-ksi9)*(ksi-ksi4)*(yeta-yeta7)*(yeta-yeta1)/N8d;
  Nx[8] = (ksi-ksi8)*(ksi-ksi4)*(yeta-yeta2)*(yeta-yeta6)/N9d;

                 /* theta1 values at each nodes of quadratic element */

  Px[0] = t1[n][m];
  Px[1] = t1[n][m+1];
  Px[2] = t1[n][m+2];
  Px[3] = t1[n+1][m+2];
  Px[4] = t1[n+2][m+2];
  Px[5] = t1[n+2][m+1];
  Px[6] = t1[n+2][m];
  Px[7] = t1[n+1][m];
  Px[8] = t1[n+1][m+1];

  *th1 = 0;
  for (b=0; b<9; b++)            /* Interpolation of theta1 */
  {
```

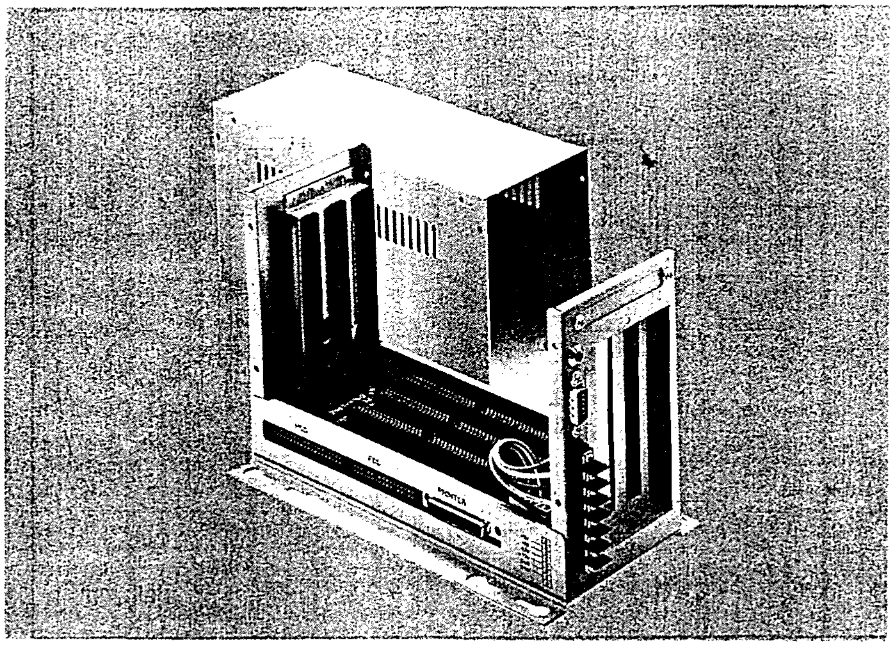
```
        *th1 = *th1 + Nx[b] * Px[b];
    }
                                        /* Calculation of x and y */
    sinTx = (float) sin(*th1) + 0.000000001;
            cosTx = (float) cos(*th1);
    *y = ( *L1 - r * ( pi - *th1 ) ) * sinTx + r * cosTx;
    *x = -H - r / sinTx + *y * cosTx / sinTx;
    return(1);
  }
}

void boundary_check(float xcheck, float ycheck, float thcheck, int *flag)
{
  float   x,y;

  *flag = 0;
  if (y0 > ycheck) *flag = 1;
  if (x1 < xcheck) *flag = 2;
  if (y3 < ycheck) *flag = 3;
  if ( (x4 > xcheck) && (ycheck > y4) ) *flag = 4;
  y = ((y5-y4)/(x5-x4))*(xcheck-x4) + y4;
  if ( (y < ycheck) && (ycheck <= y4) ) *flag = 5;

  if (fabs(thcheck) >= 360) *flag = 6;

  if (*flag == 0) sound_off();
  else sound_on(700);
}

void LED_on(int id)
{
  if (id == 1) outp(DIO_REG, 6);
  if (id == 2) outp(DIO_REG, 5);
  if (id == 3) outp(DIO_REG, 3);
}

void LED_on_all(void)
{
  outp(DIO_REG,0);
}

int read_sw(void)
{
  return( inp(DIO_REG) & 3 );
}
```

```
void LED_on_13(void)
{
   outp(DIO_REG,2);
}

void LED_off(void)
{
   outp(DIO_REG,7);
}

void LED_flashing(int in)
{
   int i;

   for (i=0; i<in; i++) {
         LED_on_all();
         wait(500);
         LED_off();
         wait(500);
   }
}
```

APPENDIX B

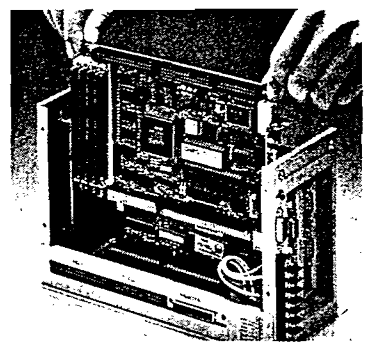
MANUFACTURERS' SPECIFICATIONS

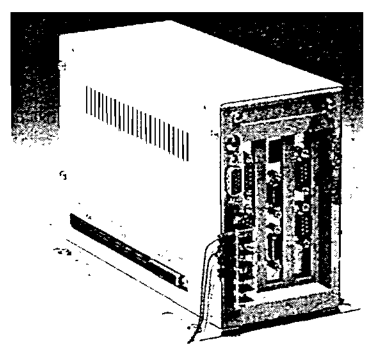
MBPC-640 MicroBox IPC Chassis

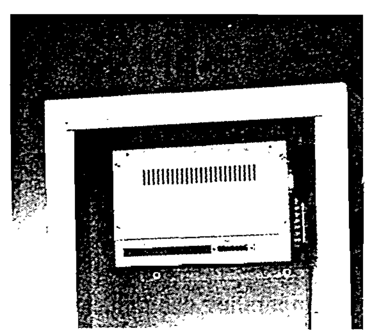


Features

- 3-slot ISA-bus (PC/AT-compatible) passive backplane
- Supports most half-size CPU cards
- Bidirectional mounting bracket
- Provides connectors for one HDD, one FDD and one printer or parallel device
- Mounting panel for special connectors or extension cables
- Compact and rugged design







Introduction

The MBPC-640 MicroBox IPC Chassis is a compact computer chassis ideal for embedded control systems or other limited space applications. Its small 8.6" (L) x 3.5" (W) x 5.9" (H) (218 mm x 88 mm x 150 mm) dimensions are well suited for vending machines, mobile computer systems and unattended controllers. This rugged chassis will withstand temperatures from 32 to 140°F (0 to 60°C).

The MBPC-640's 3-slot ISA-bus (PC/AT compatible) passive backplane supports half-size CPU cards as well as video adapter, I/O and network interface cards. The MBPC-640 provides connectors for hard and floppy disk drives and a parallel/ printer port. Power ON/OFF LED status indicators and a special mounting panel for extension cables or special connectors are also included.

CPU Cards

The MBPC-640's passive backplane works perfectly with our half-size all-in-one CPU cards, such as the PCA-6134 and PCA-6143. The PCA-6134 comes equipped with an 80386SX-33 MHz CPU, a math coprocessor socket, sockets for up to 16 MB DRAM, two serial ports (one RS-232, one RS-485), a built-in 1.44 MB Flash/ROM Disk, one parallel port, hard and floppy disk drive controllers, a watchdog timer and a piggyback module connector. The PCA-6134's piggyback module connector is a 64-pin, 8-bit PC-bus link to modules such as the PCD-8931 Flash/RAM/ROM Disk Piggyback Module and the PCA-6443 Flat-panel/CRT VGA Piggyback Module.

Installation

Simply plug the PCA-6134 or any other half-size CPU card into the MBPC-640's passive backplane.

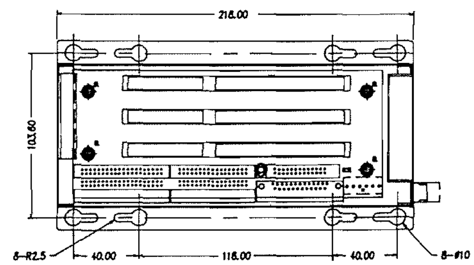
Once you have configured the MicroBox, you can attach it to a wall, panel or just about any other surface with its two mounting brackets.

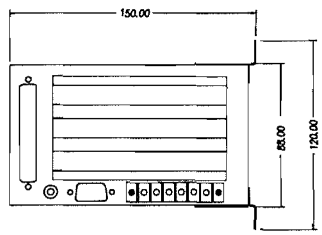
Together with the PCA-6134 the MBPC-640 provides a flexible, easy-to-maintain system suitable for almost any industrial environment.

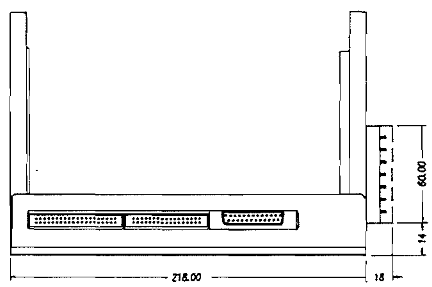
Applications

- Wall or panel mounted IPCs
- Industrial PC embedded controllers
- Vending machines
- Mobile computer systems
- Unattended (run-only) controllers

Dimensions







Unit: mm

General

- **Cooling**: Provided by vents along chassis surface
- **Construction**: Heavy duty steel
- Convenient connectors for HDD, FDD and parallel/printer device
- LED indicator for +5 V power ON/OFF
- Terminals for +5 V, -5 V, +12 V and -12 V_{DC} power source input
- **Dimensions**: 8.6" (L) x 3.5" (W) x 5.9" (H)
  (218 mm x 88 mm x 150 mm)
- **Net weight**: 3.3 lbs (1.5 Kg)
- **Shipping weight**: 4.4 lbs (2.0 Kg)
- **Shipping dimensions**: 12.5" (L) x 3.5" (W) x 5.9" (H)
  (317 mm x 135 mm x 193 mm)

PS-150: 150-watt Switching Power Supply

- **Input voltage**: 90 to 132 V_{AC} or 180 to 264 V_{AC}, automatic range switching
- **Input frequency**: 47 to 63 Hz
- **Output voltage**: +5 V @ 15 A, -5 V @ 1 A
  +12 V @ 5 A, -12 V @ 1 A
- **Operating temperature**: 32 to 122°F (0 to 50°C)
- **Relative humidity**: 10% to 95%
- **Safety**: UL, CSA and VDE approved

Passive Backplane

- **Slots**: Three, for half-size plug-in cards
- **PC board**: 4-layer PCB with ground and power planes for reduced noise and lower power supply impedance
- **Slot connectors**: 30-micron gold-plating on all contact pins
- **Termination**:
  Plug-in termination resistors for high-speed signals

PCA-6134
All-In-One 386SX CPU Card with Flash/ROM Disk

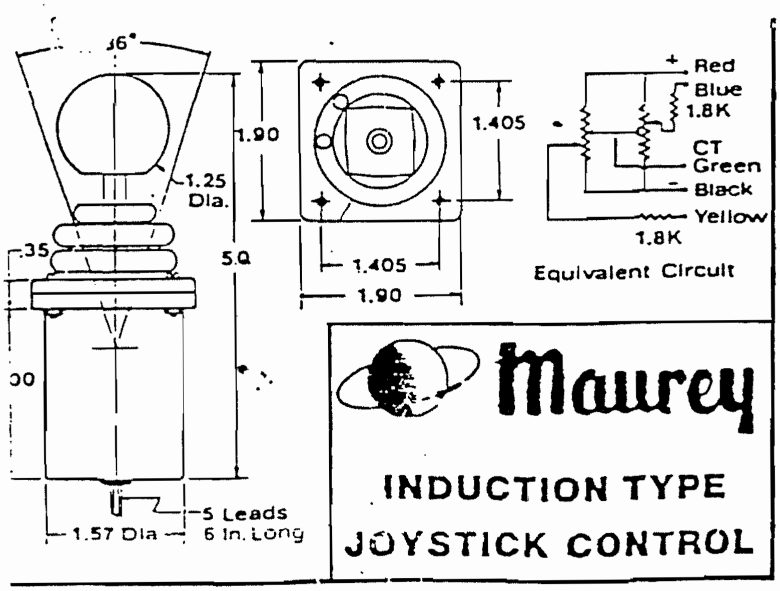
- Fully AT-386SX compatible 33-MHz 80386SX half-size card
- On-board 1.44 MB Flash/ROM Disk
- Up to 16 MB of on-board DRAM
- Watchdog timer
- Socket for 80387SX math coprocessor

Ordering Information

❑ **MBPC-640**:
MicroBox IPC Chassis

❑ **PCA-6134**:
Half-size All-in-One 386SX CPU Card with Flash/ROM Disk

❑ **PS-150**: 150-watt Switching Power Supply

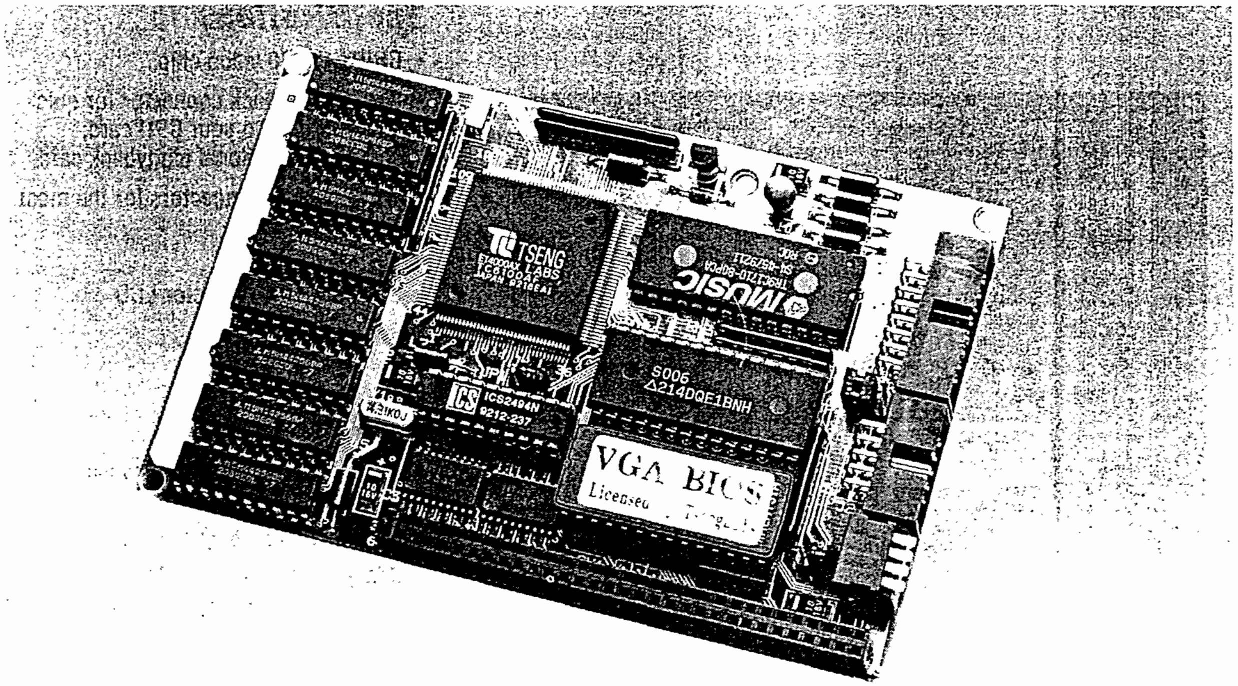


Operating Voltage:	4.5 to 15.0 Volts, D.C.
Current:	15 ma at 10 Volts, D.C.
Output Impedance:	1800 Ω (Signals)
Center Tap Impedance:	340 Ω
Equivalent Noise Resistance:	NONE
Temperature Range:	-30° to 105° C
Center Tap Voltage:	50% of Input Voltage
[illegible]solution:	Infinite
[illegible]	20.000.000 cycles
[illegible]ads:	6 inch standard length color coded

Knob has spring return to center.

Voltage Swing: **+/— 10% of Operating Voltage**

PCA-6442 Super VGA Module for CPU Cards



Introduction

The PCA-6442 plugs directly into the piggyback module connector on your CPU card to perform all the functions of a standard Super VGA plug-in card. It works with CPU cards such as the PCA-6123, PCA-6133, PCA-6134, PCA-6143, PCA-6136, PCA-6137, PCA-6146 and PCA-6147. The PCA-6442 significantly reduces MTTR (Mean Time To Repair) and frees up an expansion slot for a much needed DA&C card.

The PCA-6442 includes an auto-sensing feature for quick installation. If it detects that its DB-15 connector is attached to an analog monitor, it will automatically set itself to VGA mode regardless of its DIP switch settings.

Programs on the included utility diskettes allow you to switch between different graphics adapters and select interlaced and non-interlaced modes. We also include drivers for popular software packages such as Windows, AutoCAD, WordPerfect, etc.

Features

- Connects quickly to your CPU card through the card's 64-pin piggyback connector. Accepts a second piggyback module
- Supports MGA, Hercules, CGA, EGA, VGA and Super VGA standards
- DB-15/DB-9 adapter bracket for analog and TTL monitors
- Supports 640 x 480, 800 x 600 and 1024 x 768 resolutions, 256 colors out of 256 K colors, interlaced and non-interlaced modes
- Includes drivers for Windows, AutoCAD, Lotus 1-2-3, WordPerfect, etc.
- Supports 132 column text display
- Based on the Tseng Labs ET4000 VGA chipset
- Single power supply: +5 V

Specifications

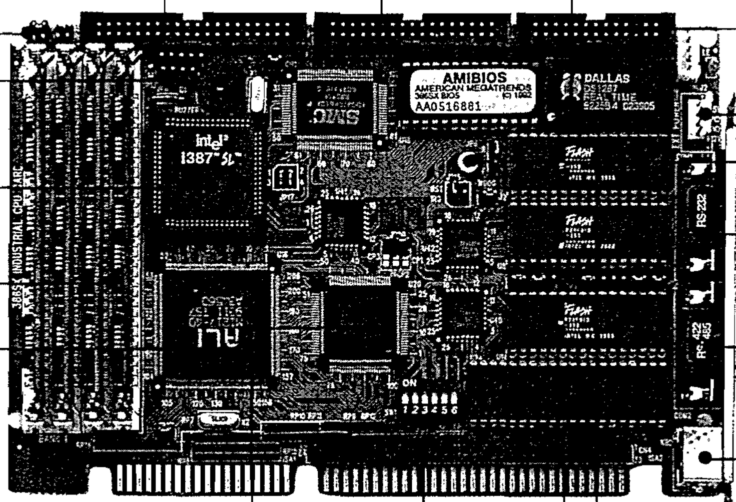
- 64-pin piggyback connector for quick installation onto your CPU card; socket for additional piggyback card
- Resolution:
  800 x 600 with 256 colors
  1024 x 768 with 256 colors
- 1 MB DRAM for high-speed memory access
- 8-bit data bus
- Dimensions:
  5.0" x 3.3" (128 x 84 mm)

Ordering Information

❑ **PCA-6442:**
Super VGA Module for CPU Cards with 1 MB DRAM

ISBC Series

PCA-6134 *Half-size All-in-One 386SX CPU Card with Flash/ROM Disk*



Introduction

The half-size PCA-6134 comes equipped with an 80386SX-33 CPU and a socket for an 80387SX math coprocessor. Also included on-board are one serial RS-232 port, one serial RS-422/RS-485 port, a bi-directional parallel port, an IDE hard disk drive interface (which controls up to two hard disk drives), a floppy disk controller (which supports two floppy disk drives) and a watchdog timer.

The watchdog timer ensures that the CPU will be reset if it stops due to a program or EMI problem, allowing the PCA-6134 to be used in stand-alone systems or unattended environments. The PCA-6134's industrial-grade construction ensures reliable operation in harsh industrial environments at temperatures up to 140°F (60°C).

We designed the PCA-6134 with SBC (Single Board Computer) applications in mind. It features a single-voltage power supply (+5 V), built-in Flash/ROM disk (which emulates a 1.44 MB disk drive, A or B) and a connector for piggyback accessory modules (such as a Flat-panel/CRT VGA controller or Flash/RAM/ROM disk).

The PCA-6134 is built using CMOS technology so it consumes very little power. Its four SIMM (Single In-line Memory Module) DRAM sockets accept 256 KB, 1 MB or 4 MB SIMM modules for total memory of 1 to 16 MB.

You can also use this 6-layer CPU card to transform any system into a 32-bit 386 compatible computer, its all-in-one configuration freeing up valuable expansion slots. The PCA-6134's highly compact form and numerous features make it an ideal cost/performance solution for high-end commercial and industrial applications where high CPU speed and low mean-time-to-repair are critical.

Features

- Fully PC/AT compatible 33 MHz 80386SX CPU
- Half-size all-in-one CPU card
- Operating temperature: 32 to 140°F (0 to 60°C)
- 3 sockets for 1.44 MB Flash/ROM disk
- Optional Flash/RAM/ROM Disk Piggyback Module (PCD-8931) and/or Flat-panel/CRT VGA Piggyback Module (PCA-6443) install on piggyback connector
- Socket for 80387SX math coprocessor
- Two serial ports: one RS-232, one RS-422/RS-485
- One bi-directional parallel port (PS/2 compatible)
- Uses CMOS devices for low power consumption
- Up to 16 MB of on-board DRAM
- Built-in IDE (AT bus) hard disk drive interface
- Built-in floppy disk drive controller
- Watchdog timer
- On-board keyboard connector
- Lithium battery backup for real-time clock/calendar
- Single power supply (+5 V)
- External power connector
- AMI BIOS
- Four holes for SBC installation

Specifications

- **CPU**: 33 MHz 80386SX
- **Bus interface**: ISA (PC/AT) bus
- **Data bus:** 16 bit
- **Processing ability**: 32 bit
- **Chipset**: ALI M1217
- **RAM memory**: 512 KB to 16 MB. Using 256Kx9 (SIMM-256-8), 1Mx9 (SIMM-1000-8) or 4Mx9 (SIMM-4000-8) SIMMs with access time of 80 ns or less

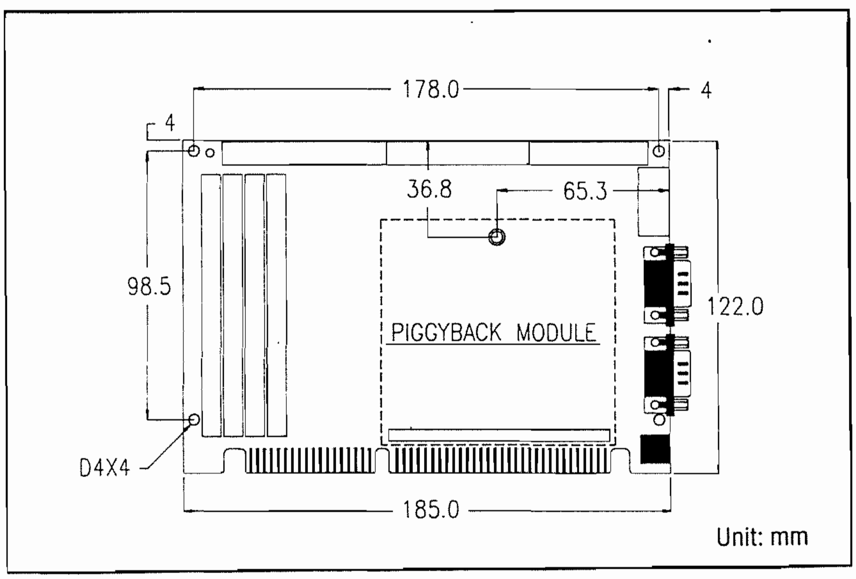
Total Memory	SIMM-256-8	SIMM-1000-8	SIMM-4000-8
512 KB	2 pcs	-	-
1 MB	4 pcs	-	-
2 MB	-	2 pcs	-
4 MB	-	4 pcs	-
8 MB	-	-	2 pcs
16 MB	-	-	4 pcs

- **Shadow RAM memory**: Support for system and video BIOS of up to 256 KB in 32 KB blocks
- **IDE hard disk drive interface**: Supports up to two IDE (AT bus) hard disk drives. BIOS enabled/disabled
- **Floppy disk drive interface**: Supports up to two floppy disk drives, 5¼" (360 KB and 1.2 MB) and/or 3½" (720 KB and 1.44 MB). Can be enabled/disabled
- **Bi-directional parallel port**: Configurable to LPT1, LPT2, LPT3 or disabled. Standard female DB-25 connector provided
- **Serial ports**: one RS-232 port, one RS-422/RS-485 port. Ports can be individually configured to COM1, COM2 or disabled
- **Real-time clock/calendar**: Uses DS-1287 RTC chip and quartz oscillator, powered by a lithium battery for 10 years of data retention
- **Watchdog timer**: Jumper-configurable to always ON, always OFF or programmable ON/OFF. The time-out interval is 1.6 seconds. Your program uses hex 043 and 443 to control the watchdog and generate a system reset
- **Piggyback connector**: 64-pin, 8-bit bus connector for option modules, such as a Flash/RAM/ROM disk module and/or Flat-panel/CRT VGA module
- **DMA channels**: 7
- **Interrupt levels**: 15
- **Keyboard connectors**: A 6-pin mini-DIN keyboard connector is located on the mounting bracket for easy access. An external keyboard adapter is also included. An on-board keyboard pin header connector is also available
- **Bus speed**: 8 MHz
- **System performance**: 40 MHz (Landmark speed V1.14)
- **Max power requirements**: + 5 V @ 1.5 A
- **Power supply voltage**: +5 V (4.75 V to 5.25 V)
- **Operating temperature**: 32 to 140°F (0 to 60°C)
- **Board size**: 7.3" (L) x 4.8" (W) (185 mm x 122 mm)
- **Board weight**: 1.1 lbs (0.5 Kg)
- **EMI**: Pending
- **MTBF**: Pending

Ordering Information

- ❑ **PCA-6134-33/0K**: All-in-One 80386SX-33 CPU Card with user's manual, IDE hard disk drive cable, floppy disk drive cable, parallel port adapter, keyboard adapter and utility diskette
- ❑ **M-29C010x3**: Package of three 128 KB Flash ROM memories for a 360 KB Flash ROM disk
- ❑ **M-27C040x3**: Package of three 512 KB EPROM memories for a 1.44 MB ROM disk
- ❑ **SIMM-256-8x4**: Package of four 4 MB, 80 ns SIMM type DRAM modules for 1 MB of memory
- ❑ **SIMM-1000-8x4**: Package of four 4 MB, 80 ns SIMM type DRAM modules for 4 MB of memory
- ❑ **SIMM-4000-8x4**: Package of four 4 MB, 80 ns SIMM type DRAM modules for 16 MB of memory

Dimensions for SBC Installation



PC7166 PC to Incremental Encoder Interface Card

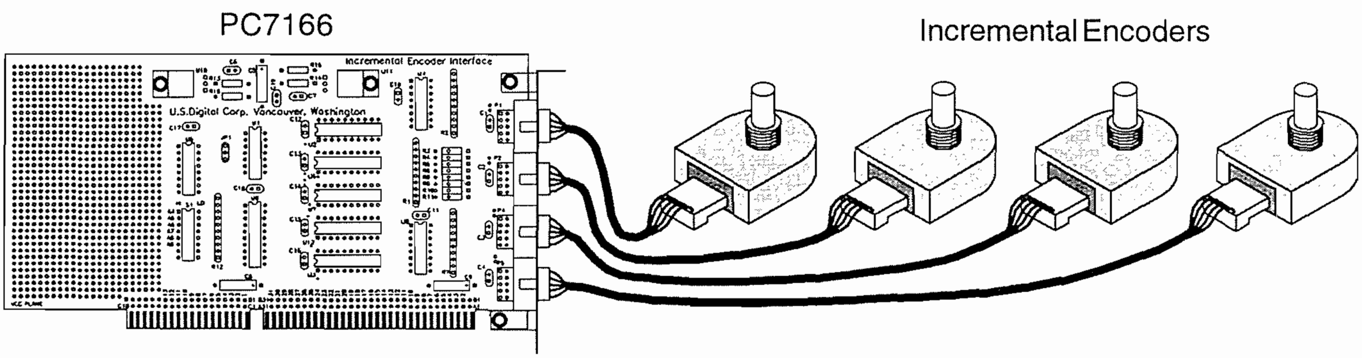
Technical Data, Rev. 05.09.96, May 1996

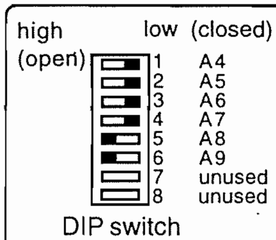
The **PC7166** daughter board plugs into a standard 8 or 16 bit ISA slot of IBM PC compatible computers. It includes four LS7166 24-bit quadrature counters and accepts single ended TTL signals or RS422 differential quadrature inputs. This card also provides 5V power to the encoders.

Software with source code in C for DOS, Windows 3.1, and Windows 95/NT is included. The software displays the position of each encoder and allows the user to change the parameters of the PC7166. This is a good starting point for the development of your custom software.

Features

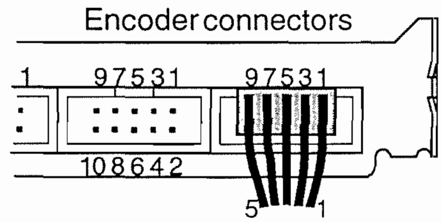
- 4 channels
- Preloadable up/down 24 bit counters
- Latched counter outputs
- X1, X2, X4 resolution multiplier
- TTL and RS422 differential interface
- Interfaces to S1, S2, E2 encoders and T2 inclinometers
- Demo software
- Prototyping area on board





The PC7166 card uses 16 consecutive IO addresses. The location is determined by the setting of addresses A4 to A9 on the DIP switch. This address block must not be used by other devices on the bus. Probable choices are: 220, 240, 250, 260, 300, 310, 330, 340, 350, 360 hex (factory default is 300).

When using a 5 pin connector in place of a 10-pin connector, use the upper row (pin 1 to pin 1).



The connector should be centered and well seated.

Functional pin description

Pin	Name	Description
1	GND	Ground, common for power and data.
2	GND	Ground, common for power and data.
3, 4		No connection
5	A-	Quadrature input, differential or TTL, see note 1.
6	A+	Quadrature input, differential only, see note 1.
7	PWR	Power supply output to encoder (190 mA max per encoder).
8	PWR	Power supply output to encoder (190 mA max per encoder).
9	B-	Quadrature input, differential or TTL, see note 1.
10	B+	Quadrature input, differential only, see note 1.

Part Number:

PC7166

PC7166 Price:

$155 / 100
$205 / 10
$225 / 2
$250 / 1

Absolute maximum ratings

Parameter	Min.	Max.	Units
Storage temperature	-40	100	°C
Operating temperature	0	70	°C
Humidity (non-condensing)	0	95	%

Phone (360) 696-2468 • Sales (800) 736-0194 • Fax-Back (360) 696-3836 • Fax (360) 696-2469
Internet E-Mail: sales@usdigital.com • World Wide Web: http://www.usdigital.com
U.S. Digital Corporation • 3800 N.E. 68th Street, Suite A3 • Vancouver, WA 98661-1353

DC Electrical Characteristics

Over operating temperature range. Typical values are specified at 25°C

Parameter	Min	Typ.	Max.	Units	Notes
+5V input current (from ISA bus)			310	mA	independent of load
+12V input current (from ISA bus)			80	mA	no load. See note 1
Output voltage PWR (to encoders)	5.00	5.25	5.50	Volts	See note 1
Output current (per encoder)			190	mA	See note 1
Differential input voltage |A+ - A-|, |B+ - B-|	0.2		14	Volts	See note 2
Common mode input voltage (A-+A+)/2, (B-+B+)/2	-7		12	Volts	See note 2
Input current (Vin = 0 to 5V) A-, B- A+ , B+	-0.4 -2.7		2.7 2.7	mA mA	no termination resistors installed
Single ended input voltage low A- , B-			1.8	Volts	See note 2
Single ended input voltage high A- , B-	3.2			Volts	See note 2
Count frequency	0		10	MHz	

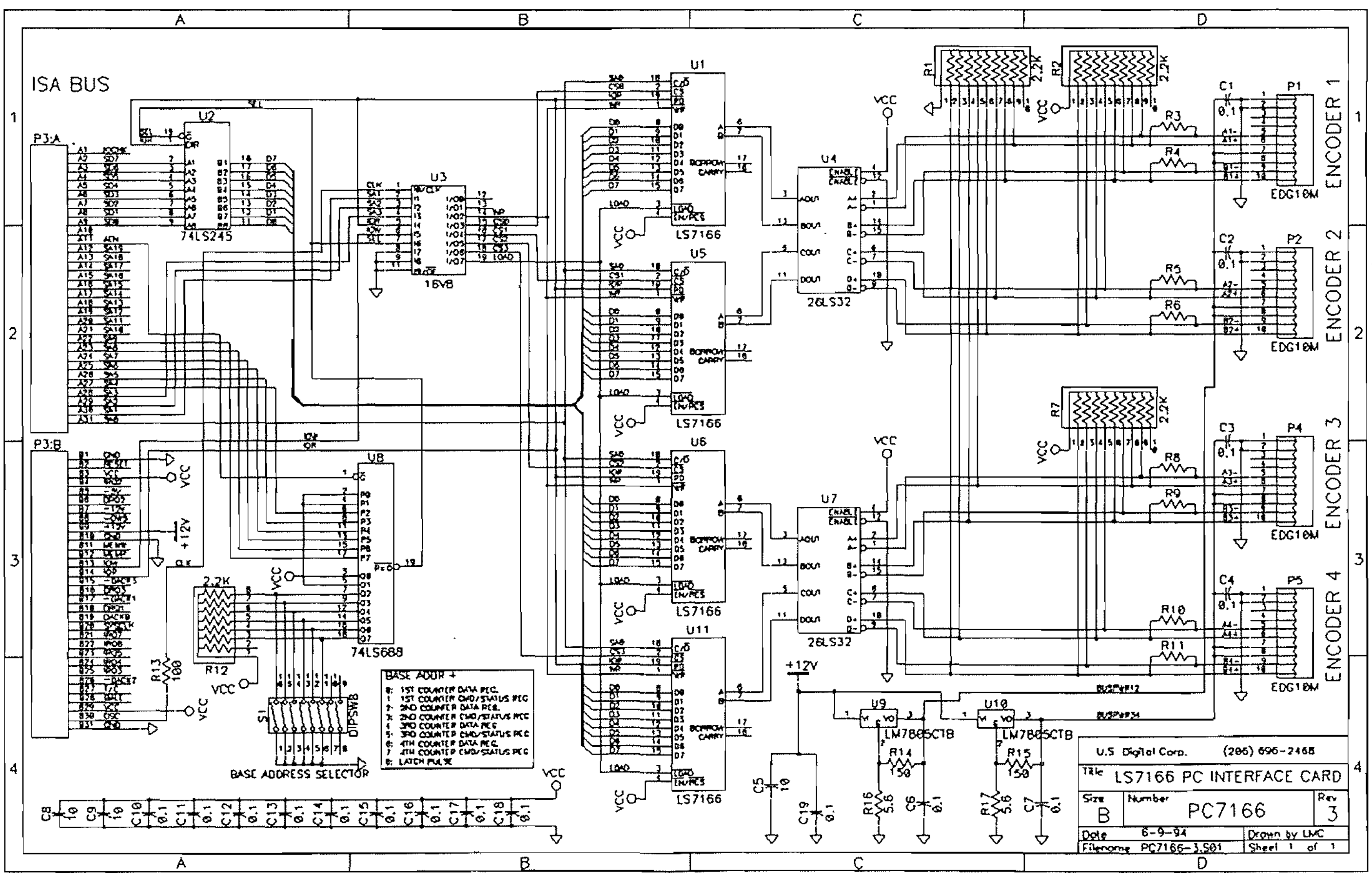
Notes:

1) The power output (PWR) to the encoders is +5.25V regulated from the +12V supply of the ISA bus. Therefore the current drawn from the +12V supply is ≤ 80mA plus current drawn by the encoders (and termination resistors, if installed). The power output to the encoders is protected against shorts, but a peak current of about 2.5A can occur for each encoder pair during a short. When using long cables, consider the voltage drop due to the current consumption of the encoder.

2) The quadrature inputs are setup to receive differential signals (RS422) or single ended TTL signals. When using the single ended interface, use A- (pin 5) and B- (pin 9). Those pins have a 2.2 kΩ pullup to +5V. The A+ and B+pins have 2.2 kΩ resistors to +5V and topins have 2.2 kΩ resistors to +5V and to ground, effectively 1.1 kΩ to 2.5 V, to keep them at that level when they are not used.

3) When using the differential interface, termination resistors can be optionally installed in the socket provided (R3 & R4 for encoder 1, R5 & R6 for encoder 2, R8 & R9 for encoder 3, R10 & R11 for encoder 4). Those termination resistors must be removed if a single ended, TTL encoder is to be used. For twisted pair cables, the typical termination resistor value is 100 Ω.

4) Differential interface is recommended for noisy environments, cables longer than 6 feet, and high speed applications. The PC4 option can be added to our S1, S2, E2, E3, H1, H3 and T2 encoders to convert them to differential outputs.

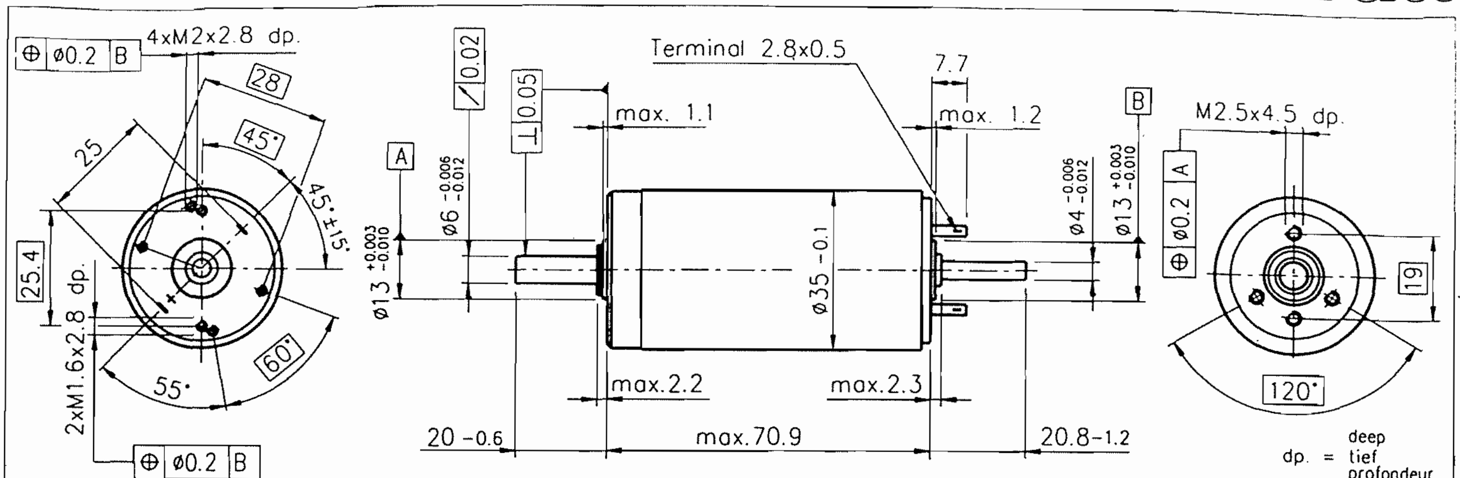


All information subject to change without notice.

Phone (360) 696-2468 • Sales (800) 736-0194 • Fax-Back (360) 696-3836 • Fax (360) 696-2469
Internet E-Mail: sales@usdigital.com • World Wide Web: http://www.usdigital.com
U.S. Digital Corporation • 3800 N.E. 68th Street, Suite A3 • Vancouver, WA 98661-1353

maxon DC motor RE 035 - 071 - . .
Graphite Brushes

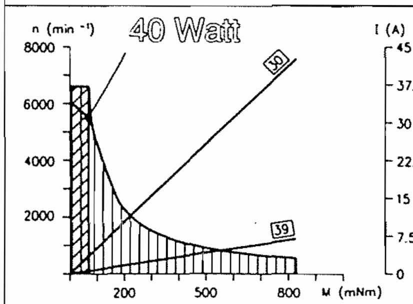
40 Watt



RE 035 - 071 - _ _ E A B 2 00 A

Motor Data	Winding Number (Order Number)	[illegible]	[illegible]	[illegible]	35	36	37	38	[illegible]	40	41	42	43	44	45	46
1 Assigned power rating	W	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
2 Nominal voltage	Volt	9.00	18.00	24.00	24.00	30.00	36.00	36.00	48.00	48.00	48.00	48.00	48.00	48.00	48.00	48.00
3 No load speed	rpm	4160	4300	4280	3610	4140	4470	3550	3820	3140	2580	2110	1630	1300	1070	865
4 Stall torque	mNm	498	571	615	487	558	588	471	508	406	330	261	200	160	129	103
5 Speed/torque gradient	rpm/mNm	8.88	7.74	7.10	7.56	7.55	7.71	7.67	7.61	7.85	7.95	8.22	8.30	8.29	8.54	8.68
6 No load current	mA	174	88.0	65.1	52.1	49.7	45.7	33.9	27.8	21.7	17.0	13.3	9.85	7.60	6.15	4.88
7 Starting current	mA	25600	14700	11700	7840	8200	7760	4940	4290	2820	1880	1220	725	462	308	199
8 Terminal resistance	Ohm	0.351	1.22	2.05	3.06	3.66	4.64	7.29	11.2	17.0	25.5	39.3	66.2	104	156	241
9 Max. permissible speed	rpm	6600	6600	6600	6600	6600	6600	6600	6600	6600	6600	6600	6600	6600	6600	6600
10 Max. continuous current	mA	3970	2300	1810	1500	1380	1230	992	804	656	538	434	336	268	219	176
11 Max. continuous torque	mNm	77.02	89.47	95.03	93.30	93.84	93.23	94.44	95.68	94 46	94.15	92.88	92.74	92.73	91.54	90 64
12 Max. power output at nominal voltage	mW	50100	61700	66900	44800	59100	67500	43000	50100	33000	22000	14200	8400	5330	3520	2260
13 Max. efficiency	%	76.7	81 9	83 4	82.3	83.4	84.0	83.0	83.8	82 6	81.3	79.7	77.6	75 6	73.4	70.9
14 Torque constant	mNm/A	19.4	38.9	52.5	62.2	68.0	75.8	95.2	119	144	175	214	276	346	418	515
15 Speed constant	rpm/V	491	246	182	154	140	126	100	80.6	66.4	54.6	44.7	34.6	27.6	22.9	18.5
16 Mechanical time constant	ms	5.92	5.16	5.03	5.00	4.96	4.91	4.90	4.86	4.85	4.84	4.84	4.83	4.82	4.82	4.81
17 Rotor inertia	gcm^2	63.6	63.6	67.6	63.1	62.7	60.9	61.0	60.9	59.0	58.1	56.2	55.5	55.5	53.9	53.0
18 Terminal inductance	mH	0.09	0.34	0.62	0.87	1.04	1.29	2.04	3.16	4.66	6.89	10.3	17.1	26.9	39.3	59.7
19 Thermal resistance housing-ambient	K/W	7.00	7.00	7.00	7.00	7.00	7.00	7.00	7.00	7.00	7.00	7.00	7.00	7.00	7.00	7.00
20 Thermal resistance rotor-housing	K/W	2.60	2.60	2.60	2.60	2.60	2.60	2.60	2.60	2.60	2.60	2.60	2.60	2.60	2.60	2.60

Operating Range



Comments

(details on page 26)

Recommended operating range

Continuous operation
In observation of above listed thermal resistances (lines 19 and 20) the maximum permissible rotor temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.

Short term operation
The motor may be briefly overloaded (recurring)

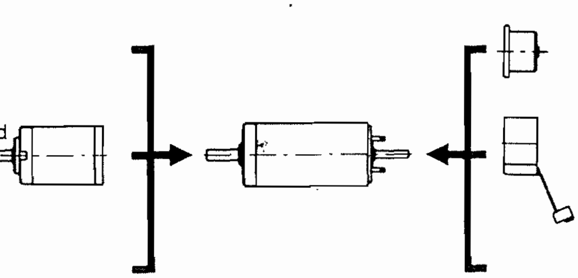
Stock program

Standard program

Special program

- Axial play 0.05 - 0.15 mm
- Preloaded ball bearings (increased no load current!) for motor-combination with encoder
  min. preloading 2.4 - 3.3 N
- Max. ball bearing loads
  axial (dynamic)
  not preloaded 5.6 N
  preloaded 2.4 N
  radial (5 mm from flange) 28 N
  press-fit force (static) 110 N
  same as above, shaft supported 1200 N
- Radial play/ball bearings 0.025 mm
- Ambient temperature range -20/+100°C
- Max. rotor temperature +125°C
- Number of commutator segments 13
- Weight of motor 370 g
- Values listed in the table are nominal
  For applicable tolerances (see page 25) and additional details please request our computer printout.

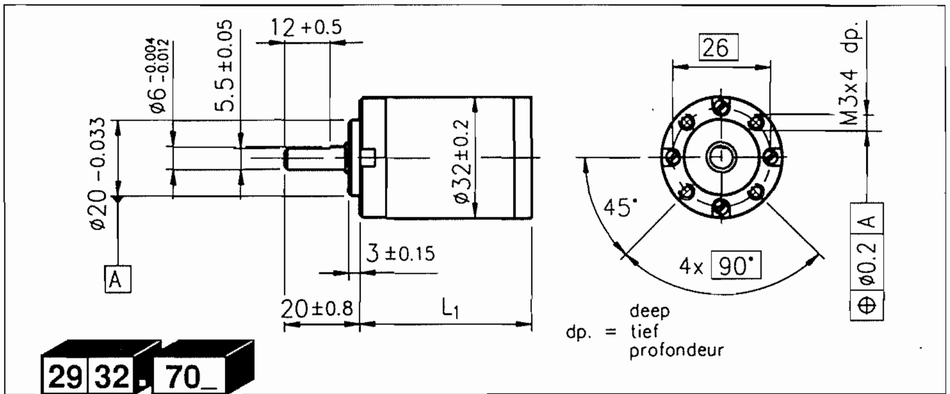
maxon Modular System



Planetary Gearhead
29 32 . 70.
Details page 77

DC-Tacho
28 22 . 800
Details page 85

Digital Encoder
HP HEDS 5010
34 07 . . . .
Details page 90



Technical Data

Planetary gearhead	straight teeth
Bearing at output	ball bearing
Radial play, 5 mm from flange	max. 0.08 mm
Axial play	max. 0.7 mm
Max. permissible axial load	120 N
Max. permissible force for press fits	120 N
Average backlash no load per stage	<0.9°
Recommended input speed	<4000 rpm
Recommended temperature range	-20/+80°C

	1 stage	2 stages	3 stages
Max. radial load, 12 mm from flange	70 N	140 N	210 N
Efficiency	80%	75%	70%

continued from page 76

Gearhead Order Number	Reduction	No. of stages	max. Torque contin-uous Nm	max. Torque inter-mittent Nm	Sense of direction	Weight g	L1* max. mm	L2 max. mm	L3 max. mm	L4 max. mm	L5 max. mm	L6 max. mm	L7 max. mm	L8 max. mm	L9 max. mm	L10 max. mm
2932.701-0005.0-000	5.2 : 1	1	0.75	1.12	=	130	26.6	81.1	103.5	93.6	101.85	97.5	115.6	118.45	93.0	111.3
2932.702-0019.0-000	19.2 : 1	2	2.25	3.37	=	170	36.1	90.6	113.0	103.1	111.35	107.0	125.1	127.95	102.5	129.8
2932.702-0027.0-000	27 : 1	2	2.25	3.37	=	170	36.1	90.6	113.0	103.1	111.35	107.0	125.1	127.95	102.5	129.8
2932.702-0035.0-000	35 : 1	2	2.25	3.37	=	170	36.1	90.6	113.0	103.1	111.35	107.0	125.1	127.95	102.5	129.8
2932.703-0100.0-000	100 : 1	3	4.50	6.75	=	215	45.6	100.1	122.5	112.6	120.85	116.5	134.6	137.45	112.0	139.3
2932.703-0181.0-000	181 : 1	3	4.50	6.75	=	215	45.6	100.1	122.5	112.6	120.85	116.5	134.6	137.45	112.0	139.3
2932.703-0236.0-000	236 : 1	3	4.50	6.75	=	215	45.6	100.1	122.5	112.6	120.85	116.5	134.6	137.45	112.0	139.3

* Motor EC032...., L1 is + 6.4 mm.

+ Motor Order Number

RE 025 - 055 - _ _ E B A 2 01 A A
Basic motor **RE 025-055-.. EBA201A**
with graphite brushes see page 50

RE 025 - 055 - _ _ E B A 2 01 A A
Basic motor **RE 025-055-.. EBA201A**
with graphite brushes see page 50

RE 025 - 055 - _ _ E B B 2 01 A A
Basic motor **RE 025-055-.. EBA201A**
with graphite brushes see page 50

RE 025 - 055 - _ _ E B A 2 01 A A
Basic motor **RE 025-055-.. EBA201A**
with graphite brushes see page 50

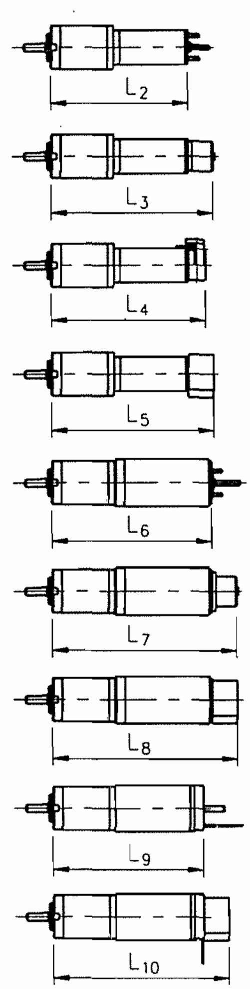
RE 035 - 071 - _ _ E A C 2 00 A A
Basic motor **RE 035-071-.. EAB200A**
see page 51

RE 035 - 071 - _ _ E A E 2 00 A A
Basic motor **RE 035-071-.. EAB200A**
see page 51

RE 035 - 071 - _ _ E A D 2 00 A A
Basic motor **RE 035-071-.. EAB200A**
see page 51

EC 032 - 060 - _ _ E _ B 2 00 B A
Basic motor **EC 032-060-.. E.B200B**
see page 55

EC 032 - 060 - _ _ E _ B 2 00 B A
Basic motor **EC 032-060-.. E.B200B**
see page 55



+ Tacho/Encoder

DC-Tacho 28 22 . 800
Details see page 85

Digital Encoder 34 16 . ...
Details see page 87

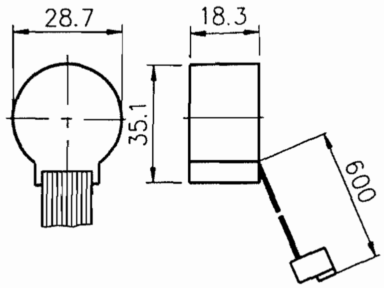
Digital Encoder HP HEDS 5010 34 07 . ...
Details see page 90

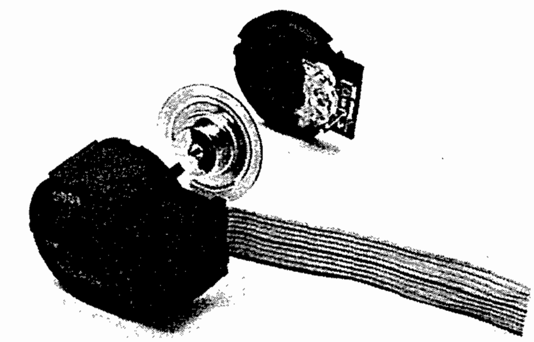
DC-Tacho 28 22 . 800
Details see page 85

Digital Encoder HP HEDS 5010 34 07 . ...
Details see page 90

Digital Encoder HP HEDS 5500 34 09 . ...
Details see page 91

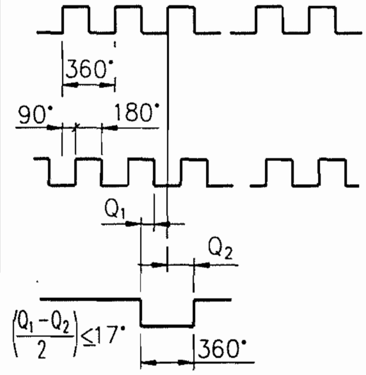
Digital Encoder HP HEDS 5010 34 07 . . . .





Encoder Order Number

34 07 . _ _ _



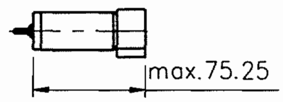
continued from page 89

+ Motor Order Number

+ Gearhead

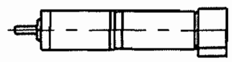
RE 025 - 055 - _ _ E A A 2 00 A

Basic motor **RE 025-055-..EAA200A**
with precious metal brushes see page 49



RE 025 - 055 - _ _ E A A 2 00 A B

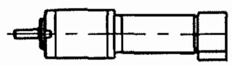
Basic motor **RE 025-055-..EAA200A**
with precious metal brushes see page 49



Planetary Gearhead 29 26 . 71.
Details see page 73

RE 025 - 055 - _ _ E A A 2 00 A A

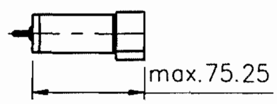
Basic motor **RE 025-055-..EAA200A**
with precious metal brushes see page 49



Planetary Gearhead 29 32 . 70.
Details see page 76

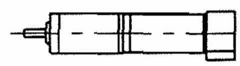
RE 025 - 055 - _ _ E B A 2 01 A

Basic motor **RE 025-055-..EBA201A**
with graphite brushes see page 50



RE 025 - 055 - _ _ E B A 2 01 A B

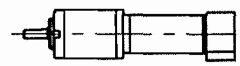
Basic motor **RE 025-055-..EBA201A**
with graphite brushes see page 50



Planetary Gearhead 29 26 . 71.
Details see page 73

RE 025 - 055 - _ _ E B A 2 01 A A

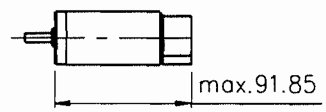
Basic motor **RE 025-055-..EBA201A**
with graphite brushes see page 50



Planetary Gearhead 29 32 . 70.
Details see page 77

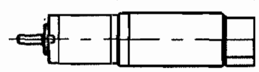
RE 035 - 071 - _ _ E A F 2 00 A

Basic motor **RE 035-071-..EAB200A**
see page 51



RE 035 - 071 - _ _ E A D 2 00 A A

Basic motor **RE 035-071-..EAB200A**
see page 51

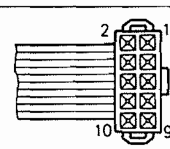


Planetary Gearhead 29 32 . 70.
Details see page 77

Technical Data

Supply voltage	5 V ± 10%	Moment of inertia of code wheel	0.4 gcm^2
Output signal	TTL compatible	Max. acceleration	250 000 rad s^{-2}
Number of channels	2 + 1 Index channel	Max. output current per channel	5 mA
Counts per turn	500	Phase shift (typical)	90° ± 25°
Operating temperature range	[illegible]	Max. encoder frequency	≥130 kHz

Pin Allocation



Type Berg 65-692-001

1 Channel A	6 Ground
2 Vcc	7 Vcc
3 Ground	8 Channel B
4 N.C. or Gnd.	9 Vcc
5 N.C. or Gnd	10 Channel I

April 1991 edition / subject to change

maxon motor

maxon motor control LSC

Operating Instructions — Edition January 1991

The linear servo controller (LSC) is a servo amplifier for the speed controlled operation of permanent magnet DC motors of up to 50 Watt output power. It works on the linear (proportional) principle.

Technical Data:

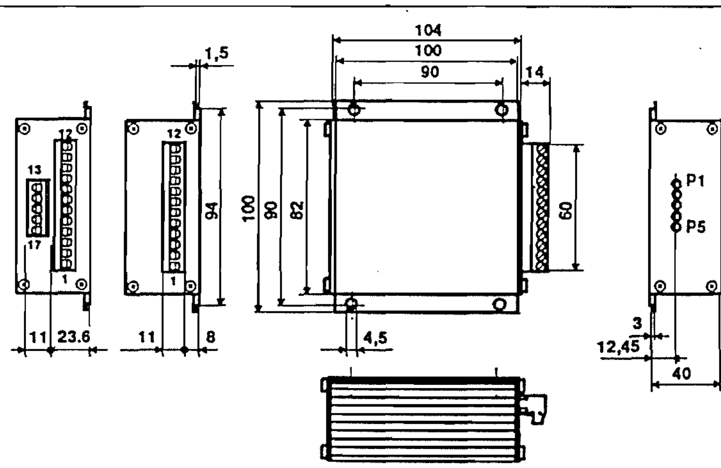
- Max. continuous output power: 50 Watt
- Operating voltage range from +12 to +30V
- Motor operating voltage from ± 6 to ± 24V
- Motor current limit adjustable from 0 to 2A
- Set Value input -10... +10V (Diff. voltage)
- Max. DC tacho input voltage from ± 2V to ± 50V
- Max. encoder frequency 100 kHz
- Supply voltage for encoder: 5V, 60 mA max.
- Reference voltage : ± 3.9V, max. load 2 mA, for external potentiometer (preferably 47 KOhm)
- Disable feature with switch having floating terminals, active low. (Switch closed = output stage disabled)
- Overtemperature protection: Shut down when housing reaches approx. 80°C.
- Ambient operating temp. -10 . . . +45°C.
- Ambient storage temp. -40 . . . +85°C.
- Weight 330 g

Advantages	Features
Different modes of operation	DC tacho speed control, encoder speed control, IxR compensation or use as voltage regulator.
4-Quadrant operation	Speed controlled operation driving and braking in both directions.
Small dimensions	100 x 104 x 40 mm, matching the Eurocard format, in protective housing.
Input setting	By means of an external potentiometer (preferably 47 KOhm) or with external control voltage level.
Measurement of actual speed	With speed proportional voltage of a DC tacho or the TTL-compatible pulses off a digital encoder.
Excellent price/ performance ratio	The output power at 24V max. and a continuous current of 2A is rated at approx. 50W.
Only one supply voltage required	Supply voltages for LSC and the encoder are generated internally.
Current limit	Adjustable using potentiometer P4.
Overtemperature protection	Protection circuit shuts off the final stage of the LSC when housing temperature reaches approx. 80°C.
Disable feature	The motor can be separated from the final stage with an external switch.
Ballast circuit = «braking help»	The LSC absorbs up to 5W (continuously) during braking (Quadrant II and IV).
Simple to connect	Connector strips with screw-type terminals.
Simple adjustment	Mode of operation selection by replugging up to three plug-in jumpers. Potentiometers are located in front.

Pin assignment:

1	Motor +
2	Motor -
3	Operating voltage V_{cc} +
4	Operating voltage V_{cc} - Gnd
5	Reference voltage V_{ref} +
6	Reference voltage V_{ref} -
7	Disable input
8	Disable Gnd
9	Set Value U_{Set} Gnd
10	Set Value U_{Set}
11	Tacho -
12	Tacho +

13	Encoder channel B
14	Encoder channel A
15	Encoder Index channel
16	Gnd
17	V_{cc}



Interelectric AG, CH-6072 Sachseln Tel.: 041/60 15 00 Tlx.: 866 414 Fax: 041/60 16 50

Subsidiaries:

FRG
maxon motor gmbh
Wardeinstrasse 3
D-8000 München 82
Tel.: 089 - 42 10 85
Tlx.: 523 726
Fax: 089 - 42 10 89

FRG (PLZ 1-49)
maxon motor gmbh
Belthenstrasse 16
D-3579 Schreckbach
Tel. 06698 - 16 81
Fax: 06698 - 17 16

USA (West)
maxon precision motors, inc.
838 Mitten Road
Burlingame CA 94010 USA
Tel: 415 697 9614
Fax: 415 697 2887

USA (East)
maxon precision motors, inc.
916 County Street
Somerset, MA 02726 USA
Tel: 001 508 677 0520
Fax: 001 508 677 0530

Japan
maxon japan corporation
5-1-15 Shinjuku
Shinjuku-Ku
Tokyo 160 Japan
Tel: 03-350-4261
Fax: 03-350-4230

Representatives:

Australia/New Zealand
M. Rutty & Co. Pty. Ltd.
Unit 1/38 Leighton Place
Hornsby NSW 2077 Australia
Tel.: 00612 476 40 66
Fax.: 00612 476 22 54

BENELUX
Sanders-Bimle B.V.
Het Lentfert 70
NL-7500 AS Enschede
Tel: 053 - 32 05 66
Fax: 053 - 30 36 65

Denmark
Compower El. Components A/S
Mileparken 22
DK-2740 Skovlunde
Tel: 44 - 53 66 22
Fax: 44 - 53 66 03

Finland
Oy Stork AB
Laurinniityntie 6
SF-00440 Helsinki
Tel: 90 - 562 11 31
Fax: 90 - 562 13 70

France (North)
mdp
7, rue Ernest Chausson
F-78114 Magny les Hameaux
Tel: (1) 30 43 56 27
Fax: (1) 30 57 07 21

France (South)
mdp
161 bis Boulevard Pinel
F-69500 Bron
Tel: 78 00 06 52
Fax: 78 00 80 19

Great Britain
Trident Engineering Ltd.
Berks, RG11 5AS, England
Tel. 0734 - 78 64 44
Fax: 0734 - 77 63 45

Korea
Schurkor Corporation
Hajin Electronics Complex
15-2, 2-Ka Hankang-Ro,
Yongsan-Ku, Seoul, Korea
Tel. 02 - 701 77 94
Fax: 02- 701 77 95

Israel
Swiss Electric Rep. Ltd.
IL-61044 Tel-Aviv
Tel: 03 - 29 29 83
Fax: 03 - 28 17 88

Italy
Abaco S.r.L.
Viale Resegone 16/a
I-20020 Arese MI
Tel: 02 - 938 28 88
Fax: 02 - 938 33 85

Austria
Kwapil & Co. GmbH
Althanstrasse 14
Postfach 64
A-1090 Wien
Tel: 0222 - 34 25 97
Fax: 0222 - 31 12 03

Sweden
AB D.J. Stork
Vretenvägen 4, Box 1037
S-172 21 Sundbyberg 1
Tel: 08 - 28 92 15,
Fax: 08 - 733 58 77

Spain
Kelvin S.A.
Miguel Yuste 39
E-28037 Madrid
Tel: 1 - 754 03 25
Fax: 1 - 204 96 39

PL/CS/R/AL
Uzimex AG
CH-8834 Schindellegi
Tel.: 01 - 784 57 60
Fax: 01 - 784 82 65

Taiwan
USE Electronics Co., Ltd
3 FL, No. 54, Sec. 1
Chung Hsiao E. Road
Taipei, Taiwan, R.O.C
Tel.: 02 - 393 48 25
Fax: 02 - 341 49 00

Yugoslavia (North)
Uzimex AG Ljubljana-Crnuce
Brnciceva 7 (Crnuce)
Yu-61231 Ljubljana-Crnuce
Tel. 061 - 372 968
Fax: 061 - 371 319

Yugoslavia (South)
Uzimex AG Novi Sad
Bulevar 23 Oktobro 46
YU-21000 Novi Sad
Tel: 021 - 339 377 / 338 008
Fax: 021 - 337 311

Hungary
Uzimex AG Budapest
Pethenyi köz 10.III. epület II/333
H-1122 Budapest
Tel: 1 - 155 82 11 / 103 / 448
Tlx: 22 42 98
Fax: 1 - 156 06 82

Bulgaria
Uzimex AG Sofia
c/o Interpred Büro «Rila»
16 Bulgaro-Savetska
Droujba Blvd.
BG-1057 Sofia
Tel: 70 80 01
Tlx: 2 32 84
Fax: 70 00 06 / 70 85 87

Adjustment tasks
Different adjustments are required depending on the mode of operation. All tasks are to be done in the sequence shown in the table.

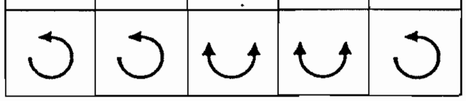
Tools required are:
One small screwdriver for the adjustments of the potentiometers.
One small Phillips type screwdriver for the LSC front plate.
One Multimeter for current and voltage measurements .

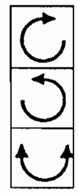
Item Task \ Mode of operation	DC	I x R	DE	U_{adj}
	DC tacho speed control	IxR Compensation	Digital encoder speed control	Voltage regulator operation
1. Pre-adjust potentiometers	x	x	x	x
2. Connect Set value circuit	x	x	x	x
3. Install plug-in jumpers	x	x	x	x
4. Arrange power source	x	x	x	x
5. Make remaining connections	x	x	x	x
6. Adjust max. desired speed	x	x	x	x
7. Adjust current limit	x	x	x	x
8. Adjust Offset	x	x	x	x
9. Adjust amplification (XP)	x	x	x	
10. Options	if required			

Item	Task	DC	I x R	DE	U_{adj}
1.	**Pre-adjust potentiometers**	X	X	X	X

By pre-adjusting the potentiometers you put yourself in a preferred starting position. Damage to the motor due to high currents is avoided. LSC units in original packing are already pre-adjusted. In all other cases or if in doubt proceed as shown.

P1	P2	P3	P4	P5
n_{max}	I x R	offset	I_{max}	XP





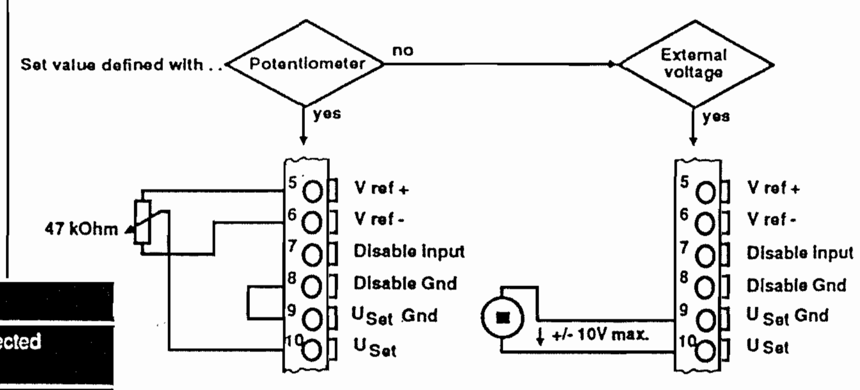
Important: The total adjustment range of the potentiometers is 20 turns (7200°). This is guarantee for precise adjustment. A built-in friction clutch prevents any damage to the potentiometers in case of over-adjustment.

Item	Task	DC	I x R	DE	U_{adj}
2.	Connect Set value circuit	X	X	X	X

The Set value must be defined by either an external potentiometer or an external voltage source.

a) Potentiometer
- Connect potentiometer (recommended value 47 kOhm).
- Install a wire jumper between terminal 8 and 9.

b) External voltage source
- Connect a voltage source of +/- 10V max. betw een terminals 9 and 10.



Warning:

Disable Gnd Set Value Gnd	8 9	These may be interconnected
Motor - V_{cc} Gnd Ref. Voltage -	2 4 6	None of these may be interconnected with any of the others, including Disable Gnd (Pin 8) and Set Value Gnd (Pin 9). Potentials are not identical i.e. at different levels.

Item	Task	DC	I x R	DE	U_{adj}
3.	Install plug-in jumpers	X	X	X	X

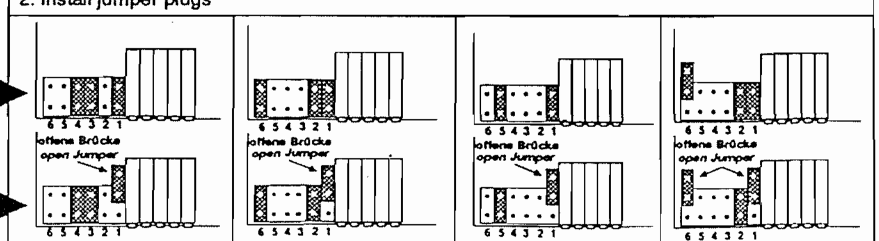
The jumpers No. 1 - 6 are located directly behind the LSC front cover. Installing a plug on top of a pair of pins interconnects these two pins. A maximum of three jumper plugs is required. Jumpers in locations 2 - 6 define the operating mode and location 1 defines the chosen Set value.

With Set value potentiometer

With external Set value voltage

1. Remove front cover

2. Install jumper plugs



3. Re-install front cover

Item	Task	DC	I x R	DE	U_{adj}
4.	Arrange power source	X	X	X	X

You may use any power supply of your own choice as long as it meets the requirements shown on the right.

Power supply requirements:	
Output voltage	According to the motor's minimum requirements, 30V DC max, 12V DC min.
Output current	2 A max.
Ripple	<5%
Potential	Galvanic separation from the line is recommended.

Please note:

- Local Safety regulations.
- While installing and adjusting we recommend: Machanically separate the motor from the driven device to preclude damage due to unexpected motion.

Important:

- The maximum voltage must be in proper relationship with the desired operating point of the motor.
- The necessary voltage can be determined as follows:

Known data:

Operating torque M_B [mNm]
Operating speed n_B [rpm]
Nominal motor voltage U_N [Volt]
(According to catalog, motor data, line 2)
Motor no load speed n_0 [rpm] at U_N
(According to catalog, motor data, line 3)
Motor speed / Torque gradient $\Delta n/\Delta M$ [rpm/mNm] (According to catalog, motor data, line 5)

Data sought:

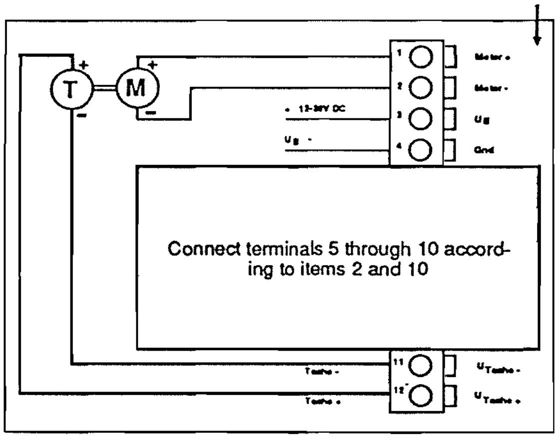
Operating Voltage U_B [Volt]

Solution:

$$U_B = \frac{U_N}{n_0}(n_B + \Delta n/\Delta M \cdot M_B) + 6$$

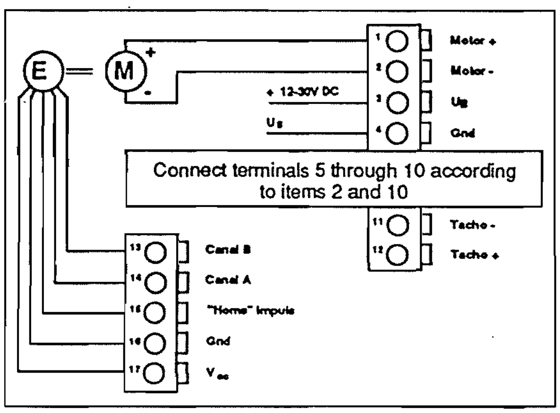
- Select a power supply capable of providing this voltage under load.
- The formula takes a 6 Volt voltage drop in the LSC into account.

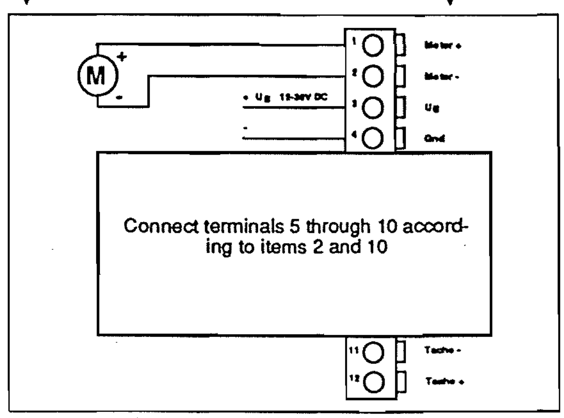
Item	Task	DC	I x R	DE	U_{adj}
5.	Make remaining connections	X	X	X	X



Warning:

Disable Gnd Set Value Gnd	8 9	These may be interconnected
Motor - Vcc Gnd Ref. Voltage -	2 4 6	None of these may be interconnected with any of the others, including Disable Gnd (Pin 8) and Set Value Gnd (Pin 9). Potentials are not identical i.e. at different levels.





Item	Task	DC	I x R	DE	U_{adj}
6.	Adjust max. desired speed	X	X	X	X

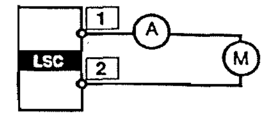
When either the potentiometer is in one of its extreme positions or the Set value voltage at its maximum level, then the motor is to achieve its maximum speed. Different Set values result in respectively changed motor shaft speeds. Change in polarity causes reversal in shaft rotation.

1.	Operating voltage pins 3/4	Power ON
2.	Set value input pin 10	Adjust to max.
3.	Potentiometer P_1	Turn P_1 CW until the desired shaft speed (Voltage on motor terminals resp. in case of U_{adj}) is reached.

Item	Task	DC	I x R	DE	U_{adj}
7.	Adjust current limit	X	X	X	X

Adjust the max. permissible motor current with potentiometer P_4.
As a result, the motor can only be loaded down to the defined max. current.

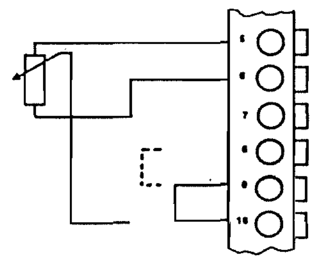
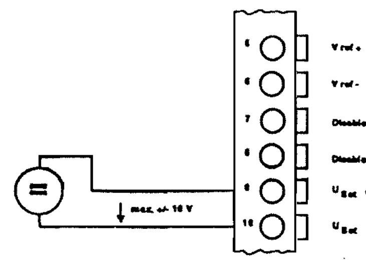
Important: A current range from 0 to 2 A applies for both directions of rotation.

1a.	Connect DC Amp-meter in series with motor, lock motor shaft.	
2.	Adjust set value	Turn P_4 CW or CCW until the desired max. motor current has been reached.

Item	Task	Potentiometer	External voltage
8.	Adjust Offset	X	X

The Offset adjustment (=Zero Adjust) causes the speed control circuit to be in position «Null» when Set value is «Null». Adjustment is done with potentiometer P_3.

As a result, the motor shaft stands still when the Set value is 0 Volt.

	Potentiometer		External voltage
1.	Remove Potentiometer connection from Pin 10.	1.	Adjust set value to 0 Volt.
			
2.	Switch wire jumper from 8-9 to 9-10	2.	Turn P_3 CW or CCW until motor shaft stands still or the voltage at the motor terminals is 0.
3.	Turn P_3 CW or CCW until motor shaft stands still or the voltage at the motor terminals is 0.		
4.	Move wire jumper from position 9/10 back to 8/9. Reconnect potentiometer.		

Item	Task	DC	DE	I x R
9.	Adjust amplification (XP)	X	X	X

Potentiometers P_5, P_2 respectively, are used to adjust the control accuracy of the system.

Caution: Poor adjustment (excessive amplification) causes the LSC to oscillate. Motor as well as LSC heat up.

Current amplification is ineffective in case of voltage regulator operation. P_5 only influences the output voltage.

1.	Max. Set value	External potentiometer, external voltage respectively in MAX position.
2.	Turn P_5 CW until the circuit begins to oscillate, the motor hums.	Turn P_2 CW until the circuit begins to oscillate, the motor hums.
3.	Turn P_5 CCW until oscillations cease.	Turn P_2 CCW until oscillations cease, over the entire motor speed range.

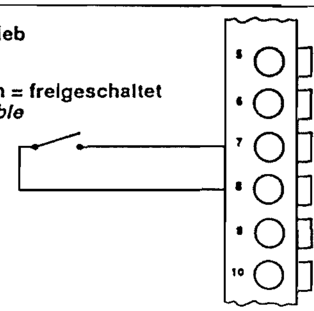
Item	Task	DC	I x R	DE	U_{adj}
9.	Options	X	X	X	X

Disable:

- The LSC can be disabled/enabled with a simple SPST switch.
- Closing the contact causes the motor to come to an un-aided stop.
- An offset current of typically 10 mA measured in the motor circuit is normal. This means that the unloaded motor could keep rotating slowly.

Schalter offen = Betrieb
open switch = run

Schalter geschlossen = freigeschaltet
closed switch = disable

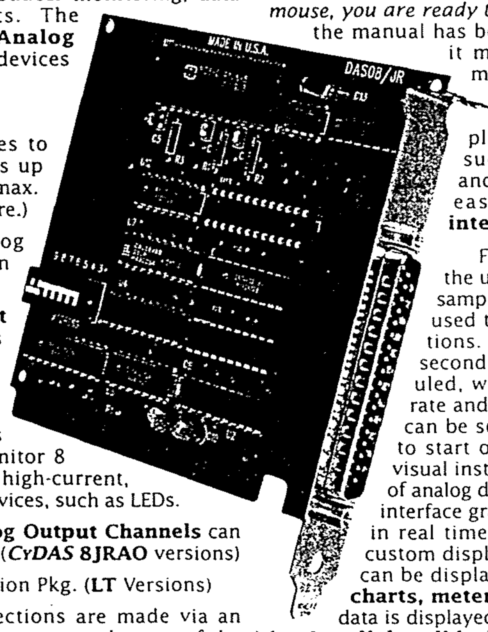


CyDAS™ 8JR Multifunction A/D Boards from $99!

The Lowest-Cost Solution for Data Acquisition

The *CyDAS* **8JR Series** of multifunction analog and digital I/O boards plug directly into a PC expansion slot. They're designed specifically for educational & high-volume OEM applications where cost is the primary consideration. With Labtech NOTEBOOK software, you get **a complete data acquisition solution for only $199!**

The *CyDAS* 8JR is ideal for low-speed (up to 20kHz) data acquisition applications with signals in the ±5V range, such as: test and measurement, process control, transducer monitoring, data collection, and laboratory experiments. The ***CyDAS* 8JRAO, with 2 channels of Analog Output (D/A),** can be used to control devices such as proportional valves.



***CyDAS* 8JR Series** boards feature:

- **12-bit A/D** converter which resolves to 2.4mV steps. Sustained Sample Rates up to **20,000/sec.** (1,000 samples/sec. max. using bundled ver. of NOTEBOOK software.)
- **8 Channels of Single Ended** Analog Input (A/D) with overvoltage protection to ±30V max., continuous.
- Analog Inputs have a **fixed ±5V input range**. A 2μsec sample & hold captures the signal for the A/D converter. Acquisition/transfer cycles can only be triggered via a software command.
- **16 bits of Digital I/O,** 8 out/8 inputs can control 8 discrete devices, & monitor 8 contact closures. Digital outputs are high-current, able to sink 24mA to drive electronic devices, such as LEDs.
- **Option: 2 Independent 12-bit Analog Output Channels** can output voltage at ±5V, in 2.4mV steps. (***CyDAS* 8JRAO** versions)
- **Option: LABTECH NOTEBOOK** Solution Pkg. (**LT** Versions)
- All the analog and digital I/O connections are made via an industry-standard 37-pin "D-type" connector at the rear of the PC. A selection of matching, economically-priced terminal boards and interface cables is available starting on page 124.

CYDAS 8JR + LABTECH NOTEBOOK™ = Solution Package

CyberResearch has developed a new Solution Package which includes the **CyDAS 8JR** with a Special Version of the powerful **LABTECH NOTEBOOK** software package for DOS or Windows, complete with all necessary drivers for the CyDAS 8JR. **This is the full $495 version of LABTECH NOTEBOOK,** with limited speed capability, and drivers for use with the CyDAS 8JR series only.

The ***CyDAS* 8JRLT solution package converts your PC into a powerful data acquisition system.** *With just 2 clicks of the mouse, you are ready to collect data.* The product is so simple that the manual has been replaced by extensive on-line help. Yet it maintains the powerful features that have made NOTEBOOK such a popular program for data acquisition and control. You can collect, analyze, display, & store data, plus monitor and control physical variables such as force, pressure, temperature, flow, and transducer outputs. Setups are fast and easy with NOTEBOOK's **iconic graphical interface** and **Windows' On-Line Help**.

Flexible process monitoring capabilities allow the user to configure applications with a variety of sampling rates and sensor types. Simple menus are used to initiate data logging, and real-time calculations. Sampling rates from a thousand points per second (max) to a few points per day may be scheduled, with each I/O point having its own sampling rate and triggering conditions. **Software triggering** can be set on an analog, digital, or calculated value, to start or stop monitoring of an I/O point, or offer visual instructions to the operator. For accurate control of analog devices, **PID control is included**. The operator interface graphically displays data and controls processes in real time. Users can quickly design and implement custom displays using the **Windows Icon tool-bar**. Data can be displayed in a number of formats, including **strip charts, meters, and bar charts**. You can change the way data is displayed while it is being acquired. **Users can create knobs, dials, slide bars, and buttons**; and use the powerful drawing and animation tools to customize displays which can be **animated** to best demonstrate what is happening **in real time.**