ASPII / MDII SERIES

DART CONTROLS

Instruction Manual

PROGRAMMABLE DIGITAL "CLOSED LOOP" DC MOTOR SPEED CONTROL with P-I-D algorithm and serial communications

P.O. Box 10
5000 W. 106th Street
Zionsville, Indiana 46077

Phone (317) 733-2133
Fax (317) 873-1105
www.dartcontrols.com

TABLE OF CONTENTS

WARRANTY ---------- 2
OVERVIEW OF THE MDII / ASPII ---------- 3
FEATURES ---------- 3-4
SPECIFICATIONS ---------- 5
DIMENSION CHART ---------- 5
JUMPERS, SERIAL CONNECTOR & TERMINAL STRIP ACCESS ---------- 6
MOUNTING GASKET INSTALLATION ---------- 6
INSTALLATION ---------- 7-8
PU-E SERIES INSTALLATION ---------- 8
WIRING DIAGRAMS ---------- 9-10
HOOK-UP DIAGRAMS ---------- 11
INITIAL CHECK-OUT OF THE ASPII / MDII ---------- 12
OPERATING THE ASPII / MDII ---------- 13
CUSTOMIZING OPERATING PARAMETERS - OVERVIEW ---------- 14
FRONT PANEL CUSTOMIZING OF OPERATING PARAMETERS ---------- 14-15
RESETTING FACTORY DEFAULTS ---------- 16
CUSTOMIZING P-I-D ---------- 16
ASPII/MDII MENU TREE FLOWCHART ---------- 17
DISPLAY INTERPRETATIONS ---------- 18
USING THE ANALOG INPUT ON THE MDII ---------- 18-22
- OVERVIEW ---------- 19
- TYPES OF SIGNAL SOURCES ---------- 19-20
- ROUTING OF ANALOG SIGNALS ---------- 20
- FIELD CUSTOMIZING ---------- 21-22
- HOOKING IT UP ---------- 22

TROUBLESHOOTING GUIDE ---------- 23
USING THE FREQUENCY GENERATOR OUTPUT ---------- 24
USER-ASSIGNABLE OUTPUT ---------- 25
USER-ASSIGNABLE OUTPUT LOGIC DIAGRAMS ---------- 26
SETTING SOFTSWITCHES ---------- 27
COMMUNICATIONS & NETWORKING ---------- 28-34
- COMMONLY USED SOFTWARE ---------- 28
- HARDWARE - JUMPER SELECTION ---------- 28-29
- RS232 & RS422/RS485 CONNECTIONS TO THE RJ45 MODULAR JACK ---------- 29-30
- SOFTWARE - DATA FORMAT ---------- 31
- FORMAT FOR MESSAGES ---------- 32
- DIRECT COMMANDS FROM A COMPUTER / PLC ---------- 33-34

DOWNLOADING DATA BETWEEN TWO UNITS ---------- 34
APPENDIX ---------- 35-39
- DESCRIPTION OF ITEM FUNCTIONS ---------- 35-37
- INDEX TO THE ASPII / MDII MEMORY ---------- 38-39

GLOSSARY ---------- 40-42
REPAIR PROCEDURE & PRODUCT LINE ---------- BACK COVER

WARNING

IMPROPER INSTALLATION OR OPERATION OF THIS CONTROL MAY CAUSE INJURY TO PERSONNEL OR CONTROL FAILURE. THE CONTROL MUST BE INSTALLED IN ACCORDANCE WITH LOCAL, STATE, AND NATIONAL SAFETY CODES.

WARRANTY

Dart Controls, Inc. (DCI) warrants its products to be free from defects in material and workmanship. The exclusive remedy for this warranty is DCI factory replacement of any part or parts of such product which shall within 12 months after delivery to the purchaser be returned to DCI factory with all transportation charges prepaid and which DCI determines to its satisfaction to be defective. This warranty shall not extend to defects in assembly by other than DCI or to any article which has been repaired or altered by other than DCI or to any article which DCI determines has been subjected to improper use. DCI assumes no responsibility for the design characteristics of any unit or its operation in any circuit or assembly. This warranty is in lieu of all other warranties, express or implied; all other liabilities or obligations on the part of DCI, including consequential damages, are hereby expressly excluded.

CAREFULLY CHECK THE CONTROL FOR SHIPPING DAMAGE. REPORT ANY DAMAGE TO THE CARRIER IMMEDIATELY. **DO NOT ATTEMPT TO OPERATE THE DRIVE IF VISIBLE DAMAGE IS EVIDENT TO EITHER THE CIRCUIT OR TO THE ELECTRONIC COMPONENTS.**

All information contained in this manual is intended to be correct, however information and data in this manual are subject to change without notice. DCI makes no warranty of any kind with regard to this information or data. Further, DCI is not responsible for any omissions or errors or consequential damage caused by the user of the product. DCI reserves the right to make manufacturing changes which may not be included in this manual.

WARNING

MAKE CERTAIN THAT THE POWER SUPPLY IS DISCONNECTED BEFORE ATTEMPTING TO SERVICE OR REMOVE ANY COMPONENTS!!! IF THE POWER DISCONNECT POINT IS OUT OF SIGHT, LOCK IT IN DISCONNECTED POSITION AND TAG TO PREVENT UNEXPECTED APPLICATION OF POWER.

ONLY A QUALIFIED ELECTRICIAN OR SERVICE PERSONNEL SHOULD PERFORM ANY ELECTRICAL TROUBLESHOOTING OR MAINTENANCE.

AT NO TIME SHOULD CIRCUIT CONTINUITY BE CHECKED BY SHORTING TERMINALS WITH A SCREWDRIVER OR OTHER METAL DEVICE.

Overview of the MDII / ASPII

OVERVIEW:

MDII: The Micro Drive II series digital motor speed controls employ an advanced 16-bit microprocessor. The MD20P is designed for digital closed-loop operation of up to 1/2 horsepower DC permanent magnet motors, while the MD30P and MD30E range is from 1/4 through 2 horsepower. This control features a *true* P-I-D algorithm, with user-programmable gains for extremely responsive and precise control over a wide variety of desired speeds and applications. Set or Actual speed is displayed directly in RPM, Process Time, or other engineering units. Field customizing permits setting of specific operating parameters. The MD20P and MD30P offer improved performance over the MD10P and MD3P respectively and come in the same 1/8 DIN and 1/4 DIN sizes. Both sizes are rated to NEMA 4X when properly installed in a panel. The MD30E offers the same electrical features as the MD30P in a wall mount NEMA 4/12 enclosure.

ASPII: The Accu-Set II is a device which allows conventional AC and DC motor drives to be operated in a digital closed loop configuration. By simply replacing the conventional 3-wire speed pot of the drive, closed loop operation is achieved. This configuration provides for extremely tight control of motor speed. The ASP20 control features a true P-I-D algorithm with user definable parameters for extremely responsive and precise control over a wide variety of desired speeds and applications. Set or Actual speed may be displayed directly in RPM, Process Time, or other engineering units. Field customizing permits setting of specific operating parameters. The ASP20 is a companion to, or replacement for, the ASP10 and offers significantly improved performance over the ASP10. The ASP20 is compatible with most drives having a supply voltage of 3 - 24VDC and an input impedance of 1,000 to 10,000,000 ohms. Output inversion allows the ASP20 to operate with drives that use a negative voltage supply. The ASP20 comes in a NEMA 4X 1/8 DIN panel mount package.

ASPII / MDII: The integrated RS485/422/232 serial interface port is ideal for monitoring/control using almost any computer or process controller. Units can even be attached in a Local Area Network, and can then be controlled and even customized either individually or all at once. "Multi" mode allows the user to choose between a "menu" of up to six programmed configurations, if desired.

FEATURES:

- Compact NEMA 4X 1/8 or 1/4DIN sturdy housing for panel mounting; or NEMA 4/12 wall mount enclosure.
- Microprocessor based; utilizes powerful 16-bit Motorola MC68HC11
- Field Programmable operating parameters
- Displays actual or desired speed directly in RPM, FPM, process time, or other engineering units
- P-I-D digital closed loop control; gains setable for optimum system performance; Fast settling time
- Accuracy ± 1 RPM of set speed
- Master/Follower operation
- Accepts a variety of pick-up inputs; hall-effect, photoelectric, TTL, or any open collector NPN capable of sinking 2.5mA; control accepts 1 to 1.2 million pulses per minute
- Internal A/D interface permits using potentiometer, 4 to 20mA or 0 to +5 VDC signal in lieu of digital pick-up signal or to control target speed, current program or frequency generator output
- Non-volatile memory retains speed setting and all field programmable parameters
- Inhibit circuit permits start and stop without breaking AC lines; pre-selecting speed, or simultaneous start-up of multiple control units
- Up/down pushbuttons for set points - slow-fast sweep
- Self-contained power supply for transducer (+5V, 25mA)
- Exclusive user assignable output - to drive relays, alarms, etc. Can be activated by any combination of conditions; upper speed limit exceeded, etc.
- Independent frequency generator allows units to produce their own leader frequency

(continued on following page)

- European-style terminal block standard (Pluggable version available)
- G.E. Lexan™ membrane seals faceplate from environment
- Multi-mode of operation allows multiple constants, settings, and upper/lower limits. Up to six different configurations can be selected from the front panel via the up/down pushbutton switches
- Programmable stall detector sets minimum acceptable speed for annunciation or shutdown
- Selectable front panel "Lockout" or hardware lockout can be used to prevent accidental or unauthorized changes

Programming features:

All customizing can be done from the front panel. Simple "menu-driven" operation for selecting/viewing/editing of all field parameters

User-selectable "programming protect" to prevent unwanted access to field customizing functions.

A wide assortment of operating parameters or "items" can be customized, including:

- Display Scalars to allow control/display in user "Engineering Units". Up to 6 different Scalars can be programmed in "Multi" mode
- Desired speed setting. Up to 6 different desired settings can be programmed in "Multi" mode
- Lower/Upper limits for speed setting. Up to 6 different limit pairs can be programmed in "Multi" mode
- Individual P-I-D gain settings
- Separate Acceleration and Deceleration times
- Stall Detect Timeout
- Displayed decimal point, or colon displayed automatically in process time mode. Up to 6 different decimal point locations can be programmed in "Multi" mode
- Selectable display blanking point. Up to 6 different blanking positions can be programmed in "Multi" mode
- Selectable display modes. Can be set to display the following:
  - Desired "target" speed (or time)
  - Actual speed (or time) "tach"
  - Program *name* (each name can have up to 4 characters)
- Operation Mode (Master Rate, Master Time, Standard Follower {rate})
- Frequency Generator rate
- Front Panel Lockout for Speed setting and/or "program" selection
- In "Multi" mode, you can select whether the speed setting is "attached" to each program, or whether the speed remains the same when selecting a new program

- Four LED function indicators make operating and customizing easy

Communications features:

- RS485; RS422; RS232 serial interface port for remote monitoring/control/programming allows the following:
  - Continuous output of actual shaft speed
  - Remote speed setting
  - Programming or listing of all field programmable parameters
  - Dartnet network allows multiple controls to be attached via one cable. Controls can be individually programmed or integrated
  - Programmable communication baud rate for 300 to 9600 buad

Specifications

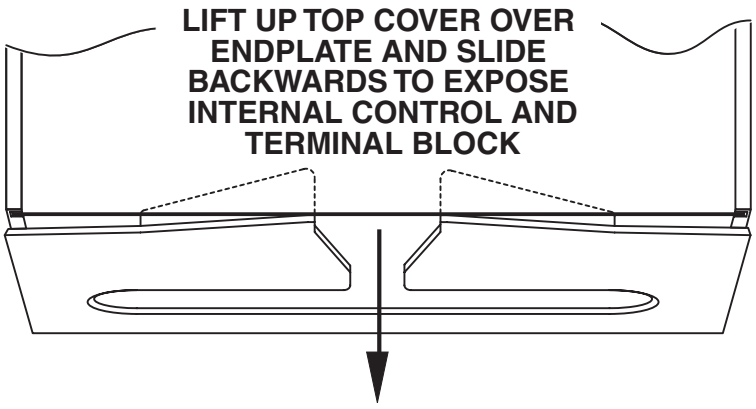
AC Input Voltage ---------- 85 - 265VAC
Input Frequency ---------- 50/60 Hertz
Operating Ambient Temperature Range ---------- -10° to 45°C (14° to 113°F)
Overload Capacity ---------- 200% for 1 Minute*
Service Factor ---------- 1.0
Efficiency ---------- 85% Typical*
Output Voltage ---------- 0-90VDC@120VAC IN / 0-180VDC @240VAC IN**
Maximum Output Current ---------- 5ADC (MD20P); 10ADC (MD30P, MD30E)*
Minimum Output Current ---------- 150mA DC*
Maximum Horsepower Rating ---------- 1/2 h.p. (MD20P); 1 h.p. (MD30P, MD30E)*
Accel/Decel Time ---------- 0 to 30 seconds for a 1000RPM Change
Serial Interface ---------- RS232/422/485
Baud Rate ---------- Selectable 300, 1200, 2400, 4800, 9600
Analog Input Voltage Range ---------- 0 to 5VDC
Motor Pick-up Input Rate ---------- 10 to 1.2M Pulses Per Minute (.167 to 20KHz.)
Leader Signal Input Rate ---------- 10 to 1.2M Pulses Per Minute (.167 to 20KHz.)
Stall Detector ---------- Banner Message and Output Shutdown
Frequency Generator Output Rate ---------- 0-9999 Pulses per Minute
Frequency Generator Accuracy ---------- ± .7% of Full Scale
Update Time ---------- 32 mS
Connection Type ---------- European-Style Terminal Block (Pluggable Version Available)
Faceplate Material ---------- Polycarbonate with GE Lexan™ Overlay
Power Supply for Transducer ---------- +5 VDC, 25mA
Supply Voltage applied across output (ASP20 only) ---------- 3 to 24VDC
Input Impedance of Driven unit (ASP20 only) ---------- 1K to 10M Ohms
Weight ---------- MD20P - 14.72 oz. (417.3g); MD30P - 24.64 oz. (698.52g);
---------- MD30E - 32.2 oz. (910g); ASP20 - 15.04 oz. (426.37g).

* Does not apply to ASPII Series
** Maximum Motor Output Voltage will be peak AC Input Voltage.

Dimension Chart

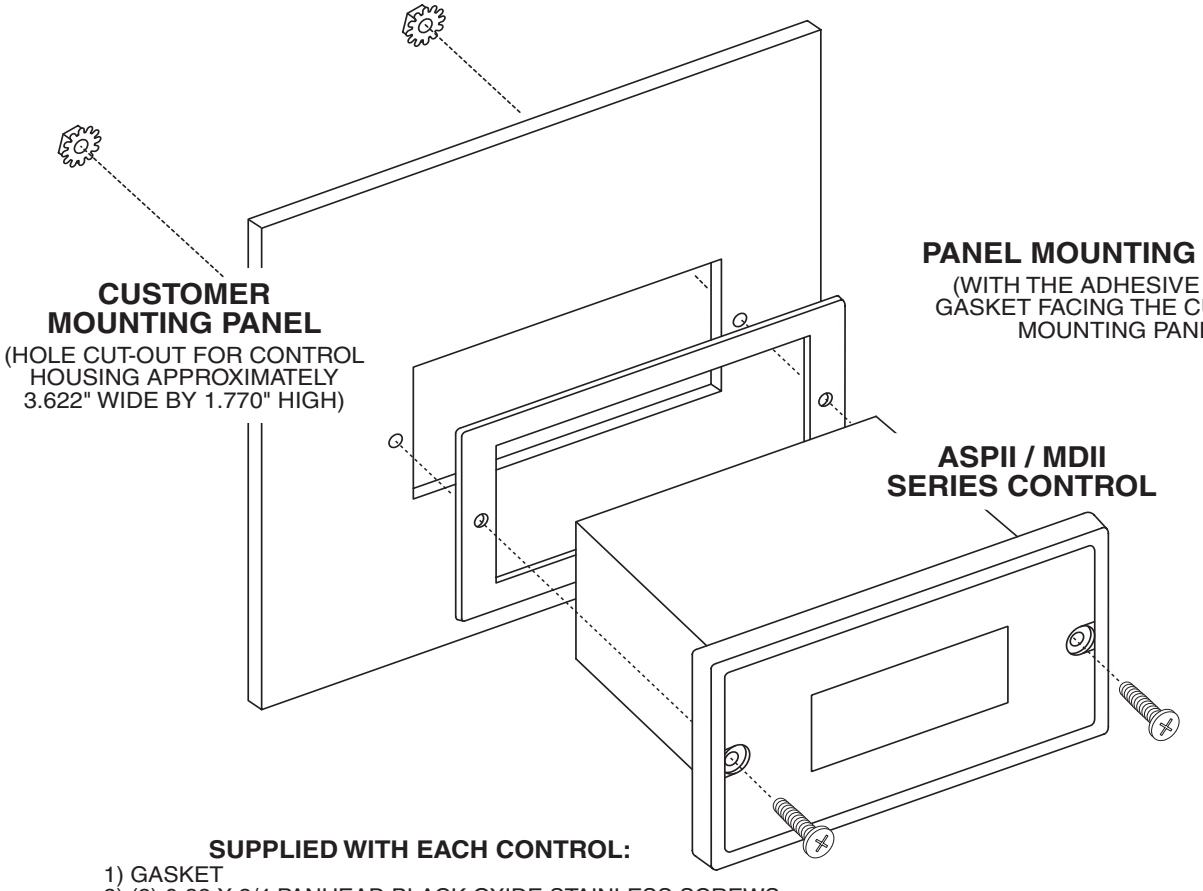
Model	*Width*	*Height*	*Depth*
MD20P/ASP20 English (inches)			
Housing	3.62	1.66	4.625
Lens	4.539	2.289	0.375
MD20P/ASP20 Metric (millimeters)			
Housing	91.94	42.16	117.27
Lens	115.28	58.14	9.53
MD30P English (inches)			
Housing	3.60	3.497	4.625
Lens	4.539	4.179	0.375
MD30P Metric (millimeters)			
Housing	91.44	88.82	117.27
Lens	115.28	106.15	9.53
MD30E English (inches)			
Assembly	5.53	7.40	3.90
MD30E Metric (millimeters)			
Assembly	140.46	187.96	99.06

Jumpers, Serial Connector and Terminal Strip Access



See “MD30E INSTALLATION” section for internal access of MD30E.
See “COMMUNICATION AND NETWORKING” section for
location of Jumpers, Serial Connector and Terminal Strip Access.

Mounting Gasket Installation



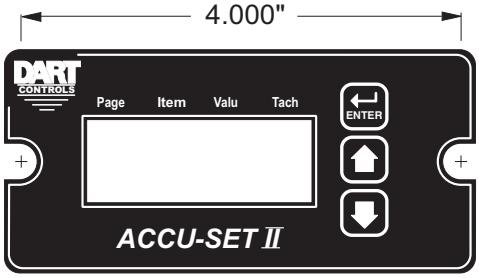
SUPPLIED WITH EACH CONTROL:

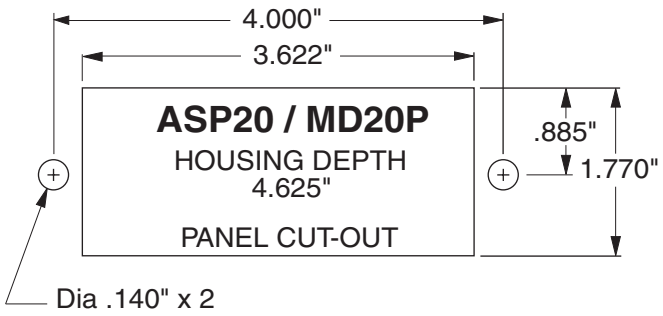
1) GASKET
2) (2) 6-32 X 3/4 PANHEAD BLACK OXIDE STAINLESS SCREWS
3) (2) #6 NUT WITH LOCKWASHER

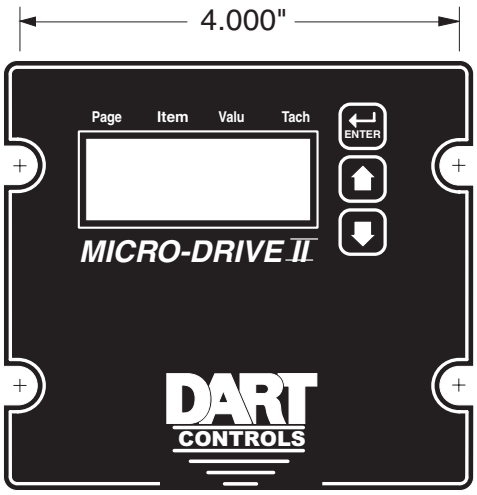
Installation

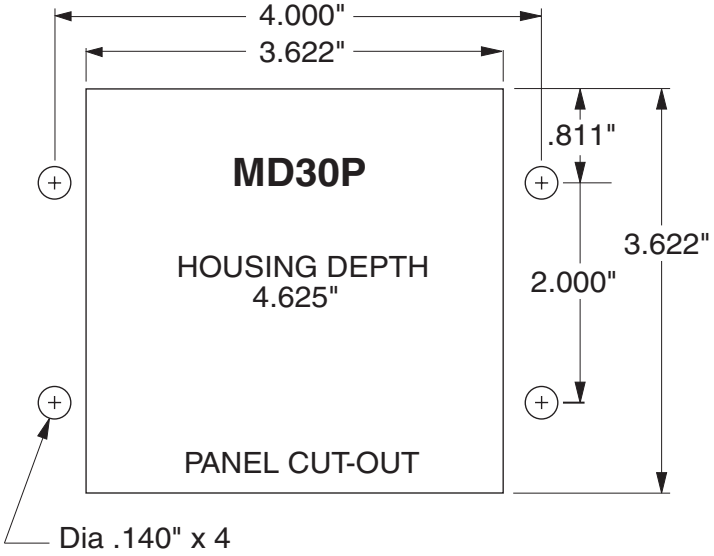
ASP20 / MD20P/30P INSTALLATION

STEP 1. Cut appropriate hole in panel using the following diagrams.
STEP 2. Mount the control into panel cut-out.
STEP 3. Secure control to panel. The control itself may be used as a template for hole location. The two (ASP20/MD20) or four (MD30) mounting holes have a diameter of .160 in. Use the supplied #6 hardware to fasten to the panel. Be careful not to over tighten mounting bolts.









MD30E INSTALLATION

STEP 1.

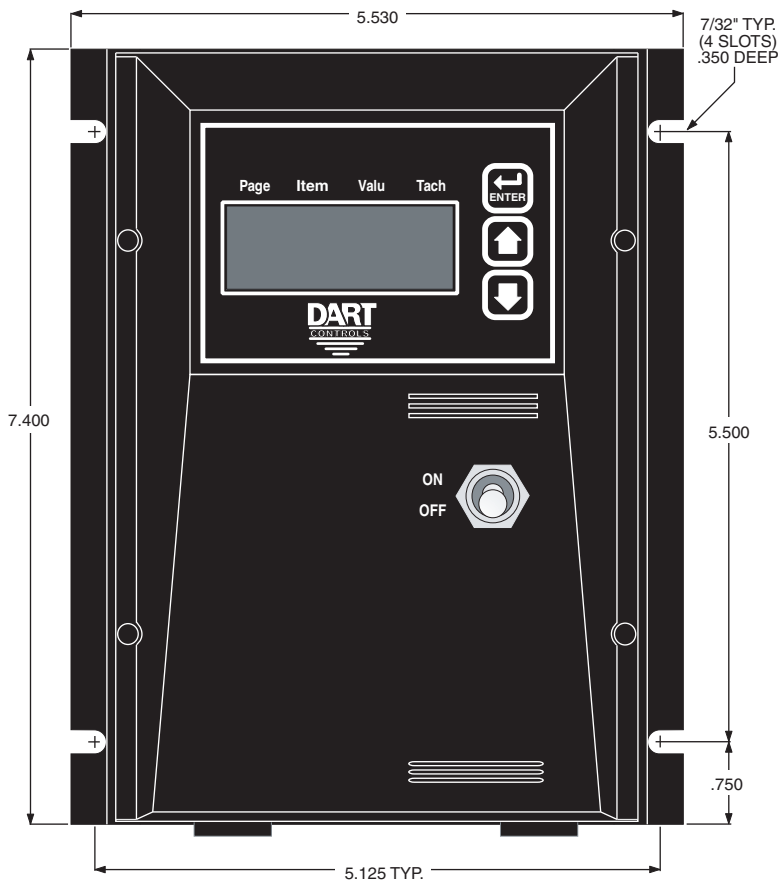
Mount the MD30E to the panel using the appropriate hardware in the four 7/32" mounting slots (the control may be used as a mounting template for hole placement).

STEP 2.

Remove the four (4) #6 screws that attach the cover to the aluminum extrusion.

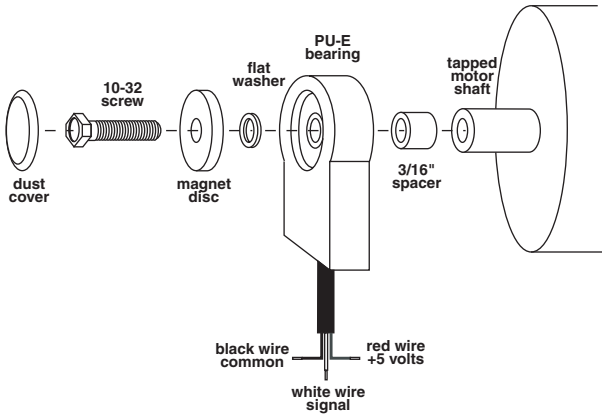
STEP 3.

Wire the control to the motor, pick-up, AC line and any communications/analog inputs that are to be used in your application. Run your wiring through the 1/2" NPT endplate conduit holes, located at the terminal strip end of the control housing.



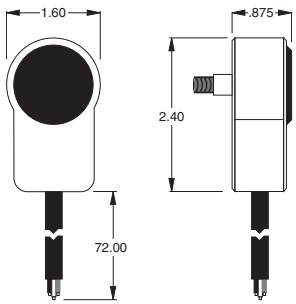
PU-E Series Installation

The Dart PU-E series pick-up is an economical way to monitor motor speed. Its patented design provides for ease of installation in otherwise difficult to reach areas. The PU-E operates from the +5V power supply in the MDII/ASPII, producing a 5 volt square wave whose frequency is proportional to speed. This signal is fed into the control as a speed reference for the microprocessor.



MODE	PULSES PER REVOLUTION
PU-2E	1
PU-4E	2
PU-20E	10
PU-40E	20

Dimensions



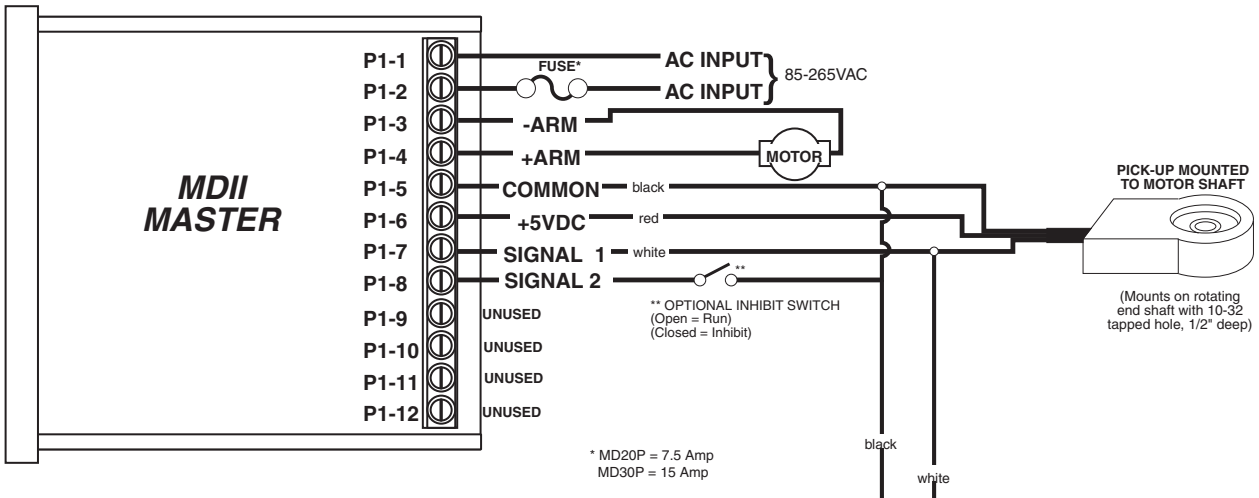
CAUTION: DO NOT OVER TIGHTEN MOUNTING SCREW !!

No other mounting screws are necessary, as the cord will keep the unit from rotating. The PU-E gives a high signal when the north pole of the magnetic disc crosses the Hall-Effect transistor. The signal is switched low when the south pole crosses this same transistor.

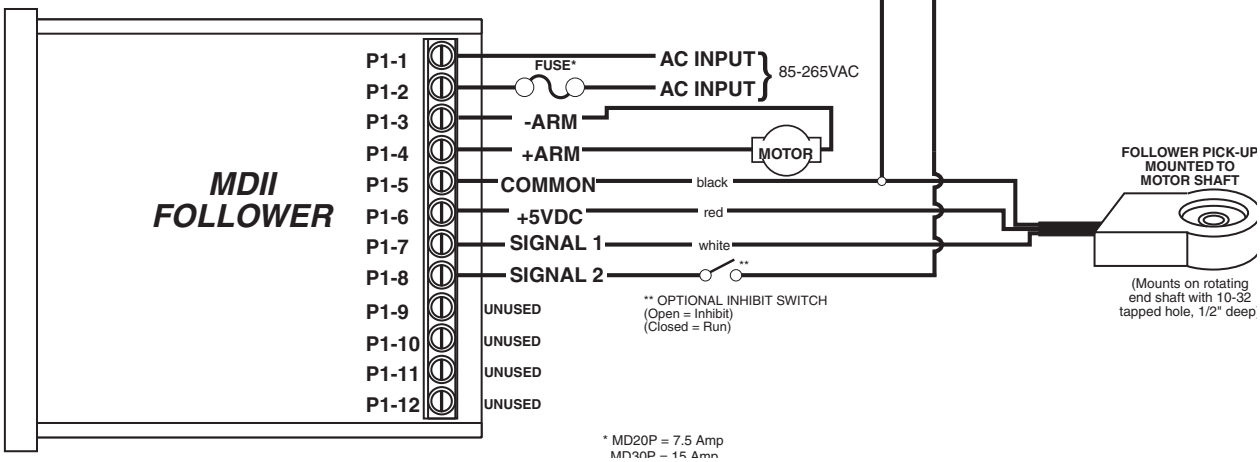
CAUTION: THE PU-E CORD SHOULD NOT BE GROUPED WITH ANY OTHER WIRES OR CORDS. FOR APPLICATIONS WITH PU-E WIRE OVER 6 FEET LONG, OR NOISY ENVIRONMENTS, A SHIELDED CABLE IS RECOMMENDED. CONNECT THE SHIELD TO THE COMMON TERMINAL ON THE MDII/ASPII, LEAVING THE SHIELD ON THE PU-E END FLOATING.

Wiring Diagrams

MDII SERIES WIRING DIAGRAM - STANDARD LEADER ONLY



MDII SERIES WIRING DIAGRAM - STANDARD FOLLOWER



STEP 1: Connect the proper input voltage to P1-1 and P1-2. NOTE: On MD20P and MD30P, fusing should be added in AC line to protect the control.
STEP 2: With AC power off, connect the PU-E as shown in hook-up diagram.
STEP 3: Connect motor by attaching -Arm to P1-3 and +Arm to P1-4.
STEP 4: Connect pick-up as shown.
STEP 5: You are now ready to apply power to your system. If the motor is rotating in the wrong direction, turn power off and reverse the armature leads.

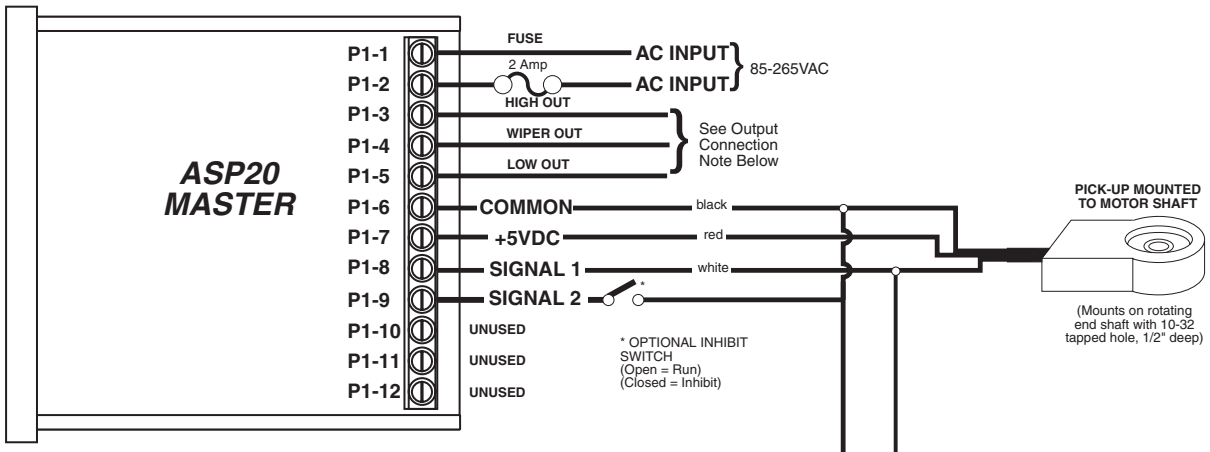
NOTE: SHIELDED CABLE IS RECOMMENDED FOR APPLICATIONS WHERE PICK-UP CORD LENGTH IS IN EXCESS OF 6 FEET. Connect the shield to the common terminal of the MDII, leaving the shield at the pick-up end "floating".

CAUTION: When master pick-up signal “A” is lost, the master MDII will run at full speed, while the follower MDII will go to zero speed. If the follower pick-up signal “B” is lost, the follower MDII will run at full speed and the master will be unaffected.

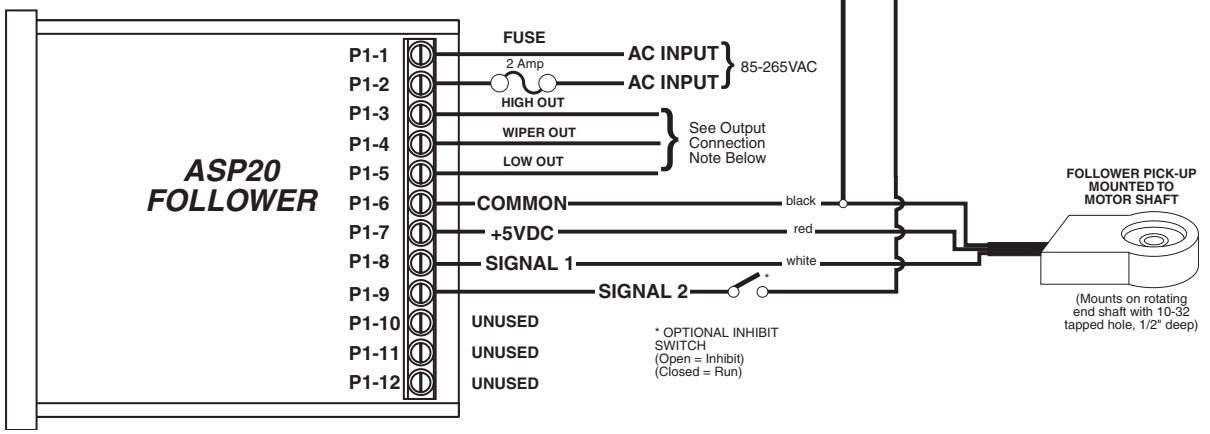
WARNING !
DO NOT ATTEMPT TO PERFORM A HI-POT TEST ACROSS AC LINES WITH THE CONTROL IN CIRCUIT. THIS WILL RESULT IN IMMEDIATE OR LONG TERM DAMAGE TO THE CONTROL.

Wiring Diagrams

ASPII SERIES WIRING DIAGRAM - STANDARD LEADER ONLY



ASPII SERIES WIRING DIAGRAM - STANDARD FOLLOWER



OUTPUT CONNECTION NOTE:

Connect to Speedpot input of the device being driven (P1-3 must be positive in respect to P1-5). If driven device has a positive supply, connect P1-3 to the Positive supply terminal (Pot High) and set Page 7 Item 11 to a value of zero. If driven device has a negative power supply, connect P1-3 to the Common terminal (Pot Low) and set Page 7 Item 11 to a value of 1.

HOOK-UP PROCEDURE:

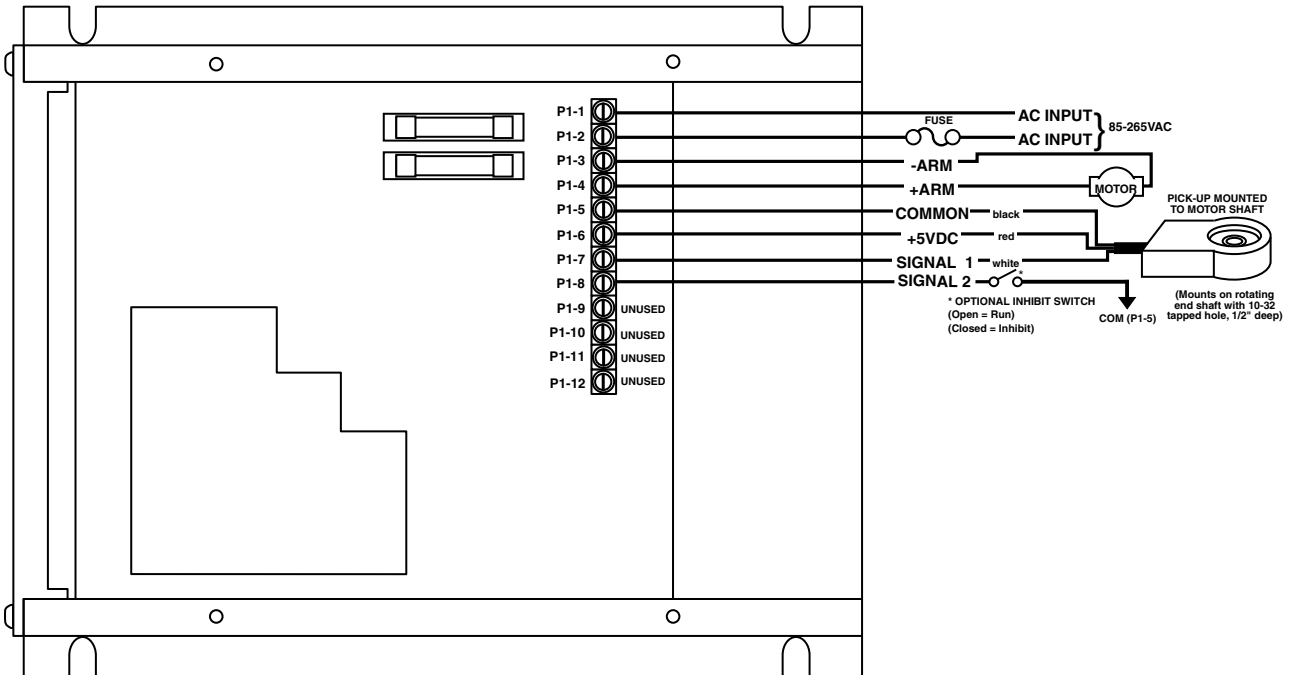
STEP 1: Connect the proper input voltage to P1-1 and P1-2. NOTE: Fusing should be added in the AC line to protect the control. A 2 amp fuse is recommended.
STEP 2: Connect the PU-E as shown in hook-up diagram above.
STEP 3: Wire the pot output of the ASP20 to the control being driven.
STEP 4: You are now ready to apply power to your system.
STEP 5: Turn IR Comp, Accel and Decel to full off position on driven control unit.

NOTE: SHIELDED CABLE IS RECOMMENDED FOR APPLICATIONS WHERE PICK-UP CORD LENGTH IS IN EXCESS OF 6 FEET. Connect the shield to the common terminal of the ASP20, leaving the shield at the pick-up end floating.

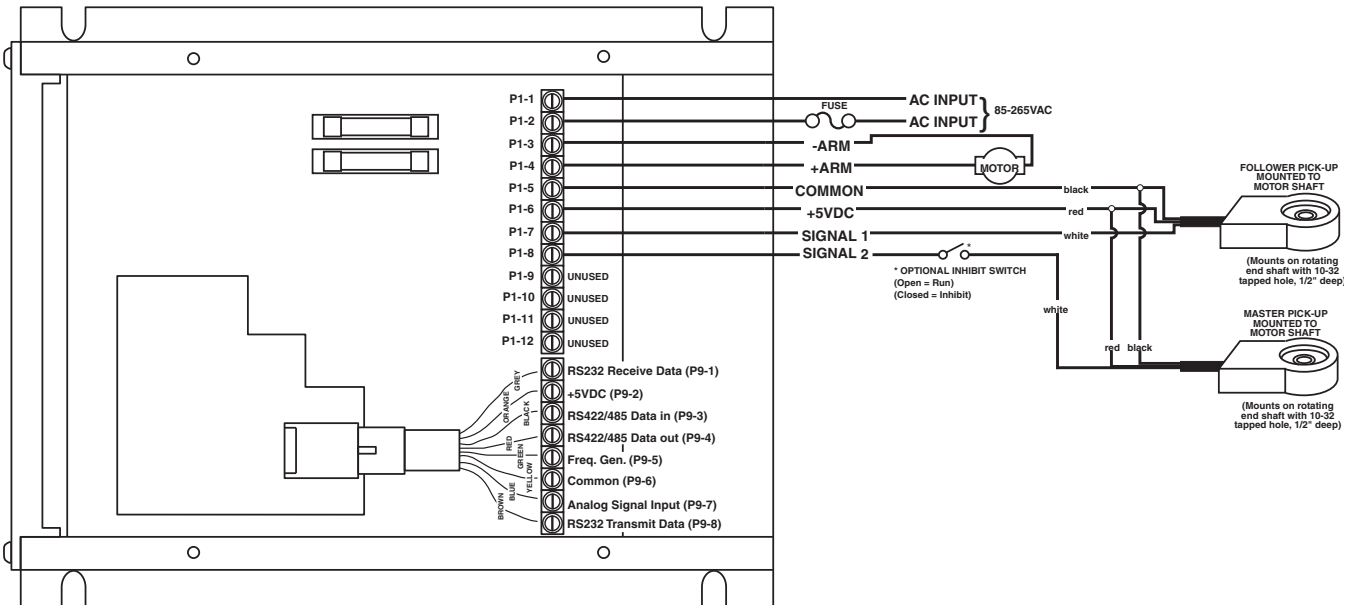
CAUTION: When master pick-up signal “A” is lost, the master ASP20 will run at full speed, while the follower ASP20 will go to zero speed. If the follower pick-up signal “B” is lost, the follower ASP20 will run at full speed and the master will be unaffected.

MD30E Hook-Up Diagrams

MD30E HOOK-UP



MD30E-7 HOOK-UP



Initial Check-out of the ASPII /MDII

The ASPII / MDII is shipped with a minimum standard program. This program not only covers the needs for a standard control configuration, but also allows the user to easily check out the basic operation of the entire system of ASPII / MDII, motor and pickup, before embarking on any custom setup changes. **The unit, as shipped, is configured as follows:**

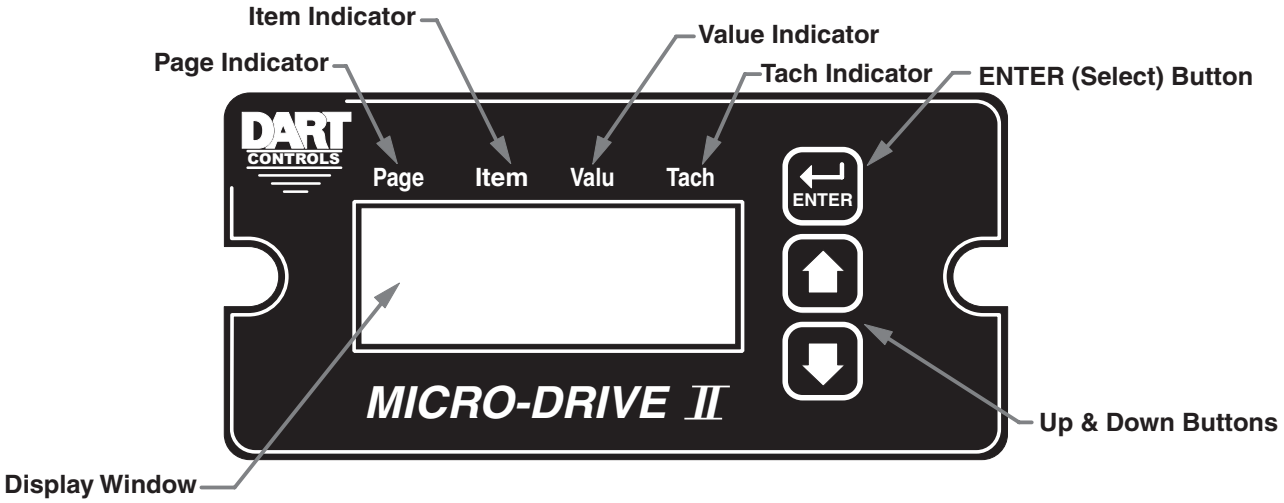
Operation Mode:	Master Rate Control
Display Mode:	Display Target (set) speed
Display Reference Value:	1
Display Reference RPM	1
Minimum Allowed Setting:	0 (Stopped)
Maximum Allowed Setting:	3600 RPM
Acceleration/Deceleration:	0 (fast as possible)
Stall Timeouts:	0 (disabled)
Pickup:	1 Pulse/Revolution (for PU-2E)
Programs:	1 (Program #1)
Front-Panel Lockout:	Target (setting) Allowed; Program Selection *Locked*
Security Code:	0 (none)
Displayed Decimal Point:	None
Communications Rate:	300 baud
Network Address:	1
"P" Adjustment:	.400
"I" Adjustment:	.031
"D" Adjustment:	.002

NOTE: For a complete explanation of these and many other parameters, please refer to the section "Customizing the ASPII / MDII" in this manual.

After control has been wired, apply AC power to the unit. You will have a display of **[0]**, and the motor should
be stopped. Next, run the motor; To do this, first press the⇑button until a reasonable (for your system) *RPM* target speed is displayed. The new speed will take effect when either the ENTER button is pressed, or after you "let up" on the ⇑(or ⇓) button for about 1.5 seconds. If the motor starts running at a constant speed, everything is o.k. If there are problems at this point, please refer to the "Troubleshooting" section *before* proceeding!
Also, note the direction of the motor's rotation. If it is backwards from desired, disconnect input power and reverse the armature wires going to the motor. You may now customize the behavior (or *configure*) the ASPII / MDII for your particular application. Please refer to the Section "Customizing the ASPII / MDII" for instructions.

Operating the ASPII / MDII

One of the most powerful features of the ASPII/MDII is its ability to be customized in many different ways. The computer in the ASPII/MDII is powerful enough to allow the person responsible for this customizing, who we call the *editor*, to customize the way the ASPII/MDII operates to meet the needs of many applications. Unfortunately, it would take a gigantic manual to cover every combination possible for the operation of the ASPII/MDII. Instead, this section will outline the front panel controls, displays, and menus, and what their possible functions are. The various "modes of operation" are explained in more detail in the section "Customizing the ASPII/MDII". Below is a drawing of the front panel of the ASPII/MDII with explanations of its controls and displays.



Controls

⇑and⇓buttons: These are used for changing things. During normal operation (*run mode*), they are used to change the desired speed setting, process time, or percentage of leader speed. They can also be used to choose from up to 6 different preset configurations, or "programs", if so desired. In *edit mode*, they are used to choose what you want to edit, as well as the actual editing itself.

ENTER button: This pushbutton has many functions. In both *run* and *edit*ing modes, the primary function of this control is to "accept" values or *menu* choices after they have been changed with the ⇑ and ⇓ buttons. In *run mode*, holding this button for about 2 seconds will cause the ASPII/MDII to enter the *Function Menu* (see "Menus", below). This button is also used during *run mode* to temporarily change display modes between displaying actual (*tach*) speed or process time, and desired (*target*) speed setting or process time.

Displays

Page Indicator: Used only in *edit mode*. Lights up when you are choosing which *page* to edit.

Item Indicator: Used only in *edit mode*. Lights up when you are choosing which *item* to edit.

Valu Indicator: Used for editing purposes both in *edit* and *run mode*s. Lights up when you are actually displaying/editing a value. Also used in *run mode* to show when you are changing the *target* value (setting).

Tach Indicator: Used only in *run mode*. Lights up when the ASPII/MDII is displaying the *actual speed* or process time of the motor pickup.

***Display Window*:** Primary information display. Used to display values, menus, error condition alarms, etc. The display is four character, seven segment, LED type, with programmable decimal points after each character, as well as a programmable colon between the second and

Customizing Operating Parameters - Overview

To facilitate the ease of understanding the concepts involved in customizing the ASPII/MDII operating parameters, we will use the following analogy: The permanent, or *non-volatile* storage area in the ASPII/MDII can be thought of as an "operation manual" or "book". The computer in the ASPII/MDII can likewise be thought of as the operator of a machine (the MicroDrive itself). The operator of this machine can look up settings on the pages of that book (the storage area) that he needs to operate his machine correctly in various situations. For example, when the operator (the computer) wants to know how quickly to accelerate his machine from one speed to another, he can look on the *page* in the book where that number is written, and set his machine (the ASPII/MDII) accordingly.

Just like a well-written operation manual, the storage area, or "memory" of the ASPII/MDII has certain *items* of information grouped together in a logical order. Settings and functions that pertain to one another appear on the same *page*, so they may be easily looked-up or changed, if necessary. Taking this idea one step further, the manufacturers of the machine (the ASPII/MDII) publish general guidelines in the operation manual (the storage area). These guidelines, or *defaults* are useful as a starting point for the operation of the ASPII/MDII. You will modify or *edit*, these guidelines as required to help the operator of the ASPII/MDII (the computer) run it as well as can be done.

Let's use acceleration time again as an example. When an ASPII/MDII leaves the factory, it has been set up to use a minimum acceleration time. This *value* is known as the *default* acceleration time. Now, let's say that this particular ASPII/MDII is to be used to control the speed of a conveyor. Unfortunately, on this conveyor are travelling crystal wine glasses! Now what do you think will happen when the ASPII/MDII is turned on?... The operator of the machine (the computer) is very strict at following the setups that his supervisor (you) have given him in his operation manual. If the book says "minimum acceleration time", that's what he'll use! So, it's up to you to change the appropriate *item* on the correct *page*, so that when he "looks it up", Mr. Computer will use a *value* for acceleration that will not spill crystal all over the floor when he turns on the machine! This process of "fine-tuning" is better known as *field customizing*. We will cover the complete process from *your* point of view in the following section.

Front-Panel Customizing of Operating Parameters

Step-by-Step Instructions

Changing, or *editing* the *values* of the *items* of information contained on the *pages* in the memory of the ASPII/MDII is quite straightforward and simple. You will need three things to do the job:

1) **A list, or *index* of the ASPII/MDII memory, so that you will know what *item* on which *page* to look at, or change. Such an index can be found in the Appendix : Index to the ASPII/MDII Memory.**

2) An ASPII/MDII, with power-applied.

3) Your finger (the button-pushing one!)

For example, if you wanted the ASPII/MDII to constantly display the actual speed or process time of the motor being controlled, you would want to change a certain *item's value* from the factory *default* of showing "target" speed. To do this, simply look up the *page* and *item* numbers of the appropriate information (display mode). You would then open the "operation manual" (memory) of the ASPII/MDII to the appropriate place, and change the number, or *value* accordingly. Let's go through this process step-by-step.

(continued on following page)

(continued)

1. First, place the ASPII/MDII in "Edit Mode". To do this:
   a) Make sure the ASPII/MDII is powered-on.
   b) Press and hold the **ENTER** button.
   c) After about 2 seconds, the word "run" will appear in the display.
   d) Push the ↓button *twice*. The word "Edit" will appear in the display.
   e) Push and release the **ENTER** button. The word "Code" will appear in the display. If it does *not*, skip to step "g".
   f) Use the ↑ and ↓ buttons until the security code number is reached, then press and release **ENTER**.
   g) The "Page" light will turn on above the display. You are now ready to select a page number...
2. Decide what you need to change, (in this example, "Display Mode").
3. In the *Appendix of this manual*, look-up the *page* and *item* number for "Display Mode" (Page 7, Item 1).
4. With the "Page" light on, use the ↑and ↓buttons until "7" (for *Page* 7) appears. Press and release the **ENTER** button. The "Item" light will turn-on.
5. Use the ↑and ↓buttons until "1" (for *item* 1) appears. Press and release the **ENTER** button. The "Valu" light will turn-on.
6. The current *value* for "Display Mode" will appear. Use the ↑and ↓buttons to change to the correct value **[0002]**. Press and release the **ENTER** button.
7. The display will "invite" you to change another item on this page. (The "Item" light will come back on, and the display will show "1", the currently selected item).
8. Since we don't need to change anything else on this page, use the ↑and ↓buttons to select item "0" (exit this page). Press and release the **ENTER** button. The "Page" light will come on, inviting you to select another page to edit. The current page (7) will appear.
9. Since we don't need to edit any other pages, use the ↑and ↓buttons to select page "0" (exit editing mode). Press and release the **ENTER** button. The "Tach" light should come on.
10. The ASPII/MDII is now back in "run" mode, with the display now indicating the "Actual" (*tach*) rather than "Set" (*target*) speed of the motor.

Congratulations!!! You have just successfully field-customized the ASPII/MDII.

NOTE: The ASPII/MDII is specifically designed to facilitate easy customizing of more than one item on more than one page. To program multiple items on the same page, repeat steps 5-7, substituting the appropriate item numbers. To select a different page to program, select item "0", then select the desired page.

Remember: Selecting *item* "0" will allow you to select another *page*. Selecting *page* "0" will complete editing, and return the ASPII/MDII to "Run" mode.

To better understand how to customize all the operating parameters and how to navigate through them when using the front panel, please refer to the sections "ASPII/MDII Menu Tree Flowchart" and "Index to the ASPII/MDII Memory".

NOTE: You can also customize the ASPII/MDII through the RS232/422/485 serial port. In fact, we recommend doing this if you are changing more than just a few items.

Resetting Factory Defaults

In the event that the editor gets hopelessly lost while customizing the ASPII/MDII control and wishes to return to the factory defaults (settings), proceed with the following:

1. Make sure the Program Enable Jumper "JP1", located on the upper board, is in the ON position.
2. Turn off power to the ASPII/MDII.
3. Depress the front panel “down arrow” and “**ENTER**” buttons simultaneously.
4. While holding both buttons in, apply power to the ASPII/MDII control. The letters “INIT” should appear in the display window.
5. Release both buttons simultaneously. The display will flash and reset to “0”.

Factory defaults (settings) have now been reset.

Customizing P-I-D

A true P-I-D speed control algorithm is employed in the ASPII/MDII which allows precise and quick response to set speed or load changes. The three items (Proportioner, Integral, Derivative) are adjustable as shown on pages 1-6. P-I-D can be set in each program to get precise speed response and regulation.

When adjusting P-I-D, begin by using the factory defaults the control is preset to: P (Pages 1-6, Item 3) to 0.400, I (Pages 1-6, Item 4) to 0.031, D (Pages 1-6, Item 5) to 0.002. If further adjustment of P-I-D is needed, follow the steps below.

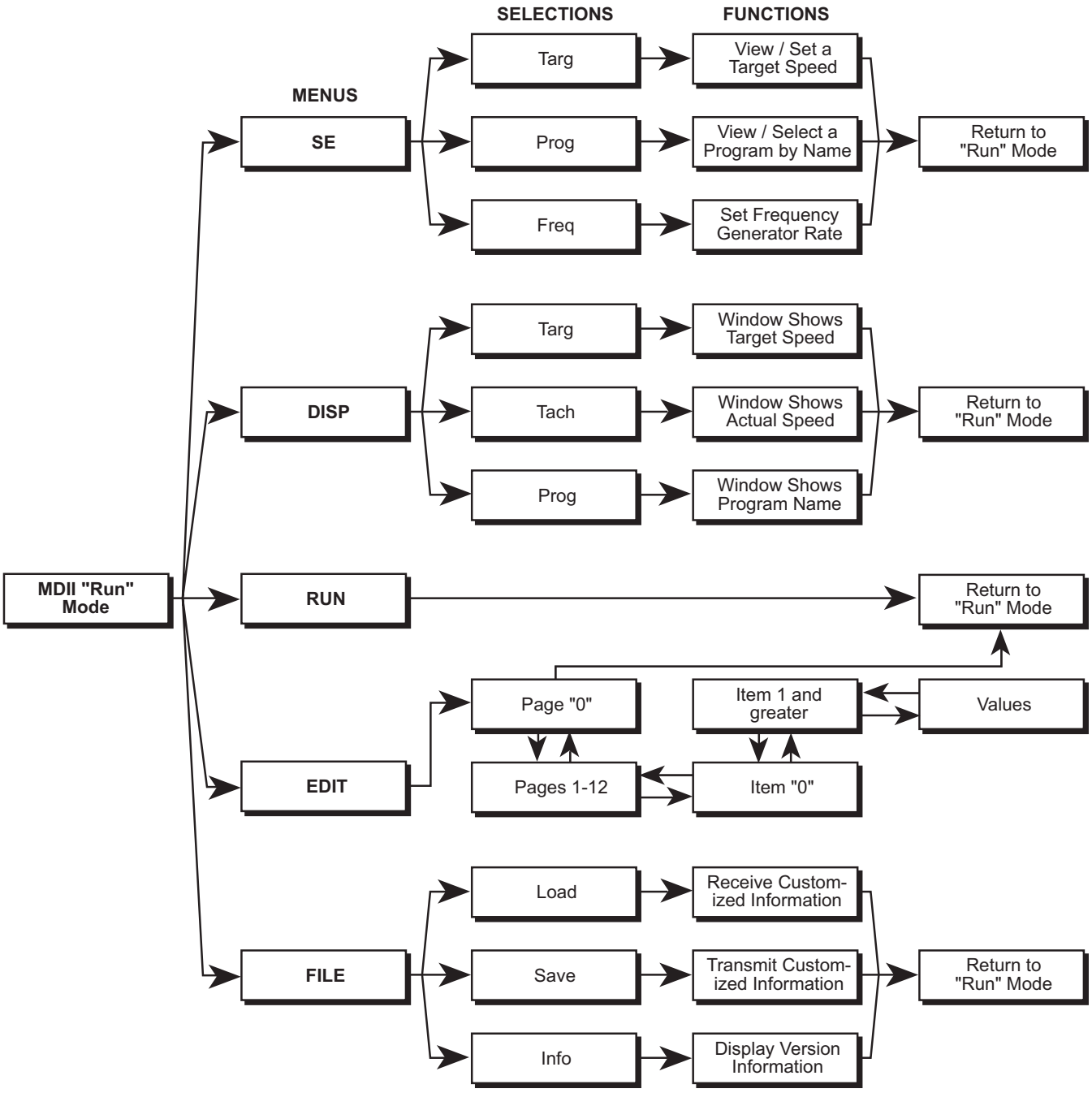
To adjust P: (Item 3)
Run the motor from zero speed to your set speed. If the start up response of your motor is too slow, increase “P” in increments of .020 until the desired start up response time is obtained. If the start up response time is too fast, decrease “P” in increments of .010 until the desired response is reached. “P” is used to adjust the start up response time only. The start up response time is approximately 0 to 60% of the set speed. “I” can be used if adjustment of the upper response time (60 to 100% of the set speed) is needed.

To adjust I: (Item 4)
Run the motor from zero speed to your set speed. If the upper response time (60 to 100% of the set speed) has any hesitation or has too slow of a response, then increase “I” in increments of .005 until the hesitation is eliminated and/or the desired upper response time is obtained. If the upper response time is too fast or has too much overshoot, decrease “I” in increments of .003 until the overshoot is eliminated and/or the desired upper response time is reached.

To adjust D: (Item 5)
“D” can be used to dampen the effect of “P”. By making “D” too large, the response time of the control can be reduced, so keep “D” as small as possible on non-regenerative controls.

Note: The proportion of P-I-D seems to be more critical than the individual values i.e.. values of 0.050-0.050-0.050 will achieve virtually the same results as 0.099-0.099-0.099.

ASPII/MDII MENU TREE FLOWCHART



Using the Menu Tree Flowchart:

The menu tree flowchart is simply a block diagram that shows how to navigate through and customize operating parameters via the front panel pushbuttons. To *enter* the "menu tree", press and hold the **ENTER** button for approximately two seconds until the "Run" menu shows on the display. To *navigate* through the "menu tree" (except "Edit" menu), simply use the "up/down arrows" to scroll up or down through the associated column and the **ENTER** button to advance to the next column. The "Edit" menu works much the same way except when **ENTER** is depressed while in "Page 0", "Item 0" or any value, you will move back to the previous column.

To better understand how to customize all the operating parameters and how to navigate through them when using the front panel, please refer to the sections "Front Panel Customizing of Operating Parameters" and "Index to the ASPII/MDII Memory".

This information should give you a good start in understanding the basic operating "feel" of the ASPII / MDII. As was said before, it is impractical to go down every possible operational "path", but this should give you some guidelines of what to expect.

Display Interpretations

Custom program names may be used if the ASPII/MDII is customized via an external computer. Since the ASPII/MDII has a 4-digit 7-segment display, some letters are displayed in a way that makes them hard to read. To assist in display interpretation the following reference is provided. (Note: To display a ° use the @ key).

0	0	A	A	P	P
1	1	B	b	Q	q
2	2	C	C	R	r
3	3	D	d	S	5
4	4	E	E	T	t
5	5	F	F	U	u
6	6	G	9	V	U
7	7	H	h	W	u
8	8	I	ı	X	ϰ
9	9	J	J	Y	y
-	-	K	ϰ	Z	2
!	!	L	L		
?	?	M	n		
Ú	o	N	n		
=	=	O	o		

Using the Analog Input on the MDII

The ASPII/MDII has a built-in analog to digital converter. This input may be used in lieu of a digital pick-up signal or to control Target speed, current program, or frequency generator output frequency. To use this input with a potentiometer source, connect a pot wiper to Pin 6 of the serial interface port, high to the +5V terminal, low to the Common terminal. The pot should have a resistance between 100 to 5000 ohms. Page 11 Items set destination source type and range for the analog input. Setting items 3 and 4 allows the pot source range to be scaled between the min and max set points. A 0 to +5V signal may be used instead of a potentiometer, connect signal to pin 7 of RJ45 and common to pin 6 of the RJ45 (P5-6). Item 2 on page 11 should be set for source type. **FOR MORE information on using the analog input, see the following 4 pages (formerly "DAN-1a" Dart Application Note).**

OVERVIEW

NOTE: This Applications Note Covers *all* controls in the MDII series with software revision numbers of 2004 and above. To determine if this applies to your control, hold the ↑ button and apply power to the unit. The MDII will cycle through a series of 3 numbers, then "--". The *second* of these numbers is the software revision (it will be 2xxx). The control will continue to cycle through the series of numbers until you release the ↑ button, then it will enter normal operation.

The MDII series of controls features an analog input. This input, which uses 2 pins on the RJ45 "modular" connector, can be used with a variety of signals. The signal from the analog input can be routed to one of several points, to provide control or "feedback" information to the MDII. This applications note will cover four major topics:

- What types of signal sources can be used
- How those signals can be used by the MDII
- Using the Field Customizing for this application
- How to hook up the analog input

TYPES of SIGNAL SOURCES

The analog input of the MDII has been designed to use basically three types of analog signal sources:

- Potentiometer (1KΩ to 10KΩ only)
- 0 to +5VDC Voltage Source (source impedance < 10KΩ)
- 4 to 20ma. Current Source (input impedance = 250Ω .1%)

NOTE: Signals should be postive-going only, and should be within the range of zero to +5 volts DC.

The selection of signal source type tells the MDII how to best use the analog information. How this selection is done is covered under the topic "Field Customizing" later in this document. Be sure to select the appropriate type of signal source, or the results will probably be less than optimal. Let's explore why this is true, by looking at the way the MDII treats the various signal source types.

Potentiometer Input:

Because noise is often a problem with potentiometer signals, and to keep the MDII from unnecessarily updating information (such as "target speed") just because the potentiometer jiggled a little bit, selection of the "Potentiometer" type of input forces the MDII to behave as follows:
First, the analog input must change its value by a small amount before the MDII will "pay attention" to it. Once this has happened, the MDII will "track" the analog input (but will *not* change anything) until the analog input (potentiometer) *stops changing* for about 1 second. When this occurs, the MDII will update itself (including non-volatile storage, if needed) according to the new value of the analog signal. This behavior is quite pleasant for use with a potentiometer used to control, say, *Target Speed*, but is totally inappropriate for use as a feedback signal, due to the "stickiness" of the current value.
NOTE: ONLY this source type will work with the Frequency Generator.

Voltage Source Input:

Selecting this source type will allow rapidly changing analog voltage signals to be "tracked" by the MDII. It is primarily intended for use as a "feedback" signal, *replacing* the conventional "pickup" signal *or* "leader" signal (for master/follower), although it is not restricted to this use. Values obtained in this way are *not* stored in "non-volatile" memory, thus they will "go away" when power is shut off to the MDII. This should not be a problem for most applications, since the value of the analog input is re-acquired when power is restored. By the way, although this input is actually designed for zero to +5VDC signals, other signal ranges (positive voltage only) such as zero to +10VDC can be accommodated by the use of an external resistive "divider".

4 to 20ma. Current Source Input:
The behavior of this type of analog source is identical to the "Voltage Source" input, with the exception that the value for 4ma. is subtracted mathematically from the signal before it is used by the MDII. The value for 4ma. is *fixed*. Currents of less than 4ma. will be treated as the minimum value, and currents greater than 20ma. will be treated as the maximum value. Use of this type of signal *requires* an external resistor (typically 250Ω). This resistor should be of fairly low tolerance (.1% or better). Using a standard 5% or even 1% resistor can result in significant accuracy and drift problems. The wiring of this resistor, and the proper value for it is covered later in this document.

ROUTING of ANALOG SIGNALS

Analog signals acquired by the MDII can be "routed" to *one* of several destinations. Except for the Frequency Generator, there is no restriction on what type of source can be routed to what destination, but a particular type of source will generally work best with the destination(s) intended to use that type. For example, it is probably not very desirable to route a rapidly changing voltage source signal to the "Program # " destination, although it *is* possible. Currently, the possible destinations for analog signals are:

- NONE (Analog Input *Ignored*)
- Target "Speed" Setting
- *Percent of Target S*etting
- Program # (1 to 6)
- Frequency Generator Rate
- Main "Tach" signal (*replaces* regular pickup)
- Leader "Tach" signal (*replaces* regular signal)

The *possible* types of analog input sources for each destination is shown below:

Destination	Pot	Voltage	Current
Target Speed	**Y**	**Y**	**Y**
% of Target	**Y**	**Y**	**Y**
Freq. Gen.	**Y**	**N**	**N**
Program #	**Y**	**Y**	**Y**
Main "Tach"	**Y**	**Y**	**Y**
Leader "Tach"	**Y**	**Y**	**Y**

NOTE: If the analog input is routed to a function that can also be controlled by the front panel buttons or serial communications messages, *any* of those sources can potentially change the value of that function, depending on who changes it most recently. This will *not* harm the MDII in any way, but should be considered when setting up the system in which the MDII is to be used. See the MDII Manual section "Index to the MDII Memory" for details on how to lock-out functions so that they cannot be changed from the front panel.

WARNING:
When the analog input is routed to either the Main "Tach" signal or the Leader "Tach" signal, the regular "barrier strip" input for that signal should *not* be connected. Doing so will give unpredictable results, but will *not* damage the MDII.

FIELD CUSTOMIZING

Field Customizing the MDII to use the analog input requires the person doing the customizing, or the "editor", to provide the MDII with a few pieces of information. These are as follows:

- The analog signal "destination" routing
- The analog signal "source type"
- The minimum value desired
- The maximum value desiredFor our purposes, the terms "Minimum Value" and "Maximum Value" are described below:

Minimum Value:
"Minimum Value" refers to the number that will be obtained when the analog source is at its *lowest* point. For potentiometer sources, that means when the pot "wiper", (the part that moves with the pot shaft) is closest to the pot terminal that is connected to signal common (zero volts on the wiper). For voltage sources, that means zero volts measured between the analog input and signal common. For current sources, the minimum value is obtained with a current of 4ma. or less.

Maximum Value:
"Maximum Value" refers to the number that will be obtained when the analog source is at its *highest*, or most-positive point. For potentiometer sources, that means when the pot "wiper" (the part that moves with the pot shaft) is closest to the pot terminal that is connected to the +5VDC barrier strip terminal on the MDII (+5VDC on the wiper). For voltage sources, that means +5VDC measured between the analog input and signal common. For current sources, the maximum value is obtained with a current of 20ma. or greater.

Setting Minimum and Maximum Values
It may help to think of the analog input on the MDII as a "black box". This box will send out an integer (whole) number somewhere between the "Minimum Value" and the "Maximum Value", depending on the signal present at the analog input when it is "sampled". What this number represents is determined by the *destination* routing you have selected; however, there is a catch: The analog input "black box" on the MDII *may* not be able to send out *every* possible integer between the Minimum and Maximum values that you set. This is due to a phenomenon known as "quantizing". This fancy word simply means that *any* time you measure something, the measurement can *only* be as accurate as the smallest "markings on your ruler". For example, if you had a ruler that was marked only in inches, you could only *accurately* measure something to the nearest *whole* inch. Anything else would involve guesswork (which the MDII is rather ill-equipped to handle!) So, in this case, any measurements made with that ruler would be *quantized* to whole inches. Now, taking this example one step further, imagine that this is a magic ruler; that is, the ruler can be stretched or squeezed to fit any distance. Only there is a problem: The magic ruler can change its length, but when you do this, instead of more "inch" markings appearing (or disappearing), the ones that are already on the ruler get *closer or farther apart*!!
So, if your ruler is 10 inches ("real inches") long to start with, it will have ten "inch marks" on it, numbered from 1 to 10. Now let's say you stretch your ruler to measure something that is 20 inches long. For-tunately, although we cannot "grow" more markings on our magic ruler, the ten markings that *are* there change their numbers so that they always show the "real" inches at that point. This means that when your ruler has been stretched from its original length of 10 inches to 20 inches, the markings will read "2, 4, 6,...20". Remember, you will still only have ten markings on your ruler. *Now* how accurately can you measure something for sure? That's right, we have changed the "quantizing" of our ruler from 1 inch to 2 inches per marking.
The analog input "black box" also has a limited number of "markings" on its "ruler". This number is determined by two things: 1) The electronics of the analog input; and 2) The analog "source type" selected. For "potentiometer" and "voltage sources", the number of "markings" or "*steps*" that the analog input has available is 256. For "current source (4 to 20ma.)" use, the number of *steps* is reduced some-what, to 192. That means if we are using a potentiometer to control the "target speed" of the MDII, and you have set the

"Minimum Value" to be 100 (engineering units), and the "Maximum Value" to be 1000 (engineering units), the closest you will be able to set the target speed to a desired value is equal to 1000 - 100 ÷ 256 = 3.51 (or about 4) of whatever your engineering units is. Values between these numbers simply "don't exist", as far as settings for the "target speed" is concerned.

Units for Destination Routes

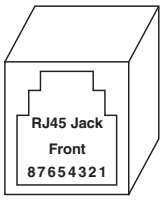
As stated above, the analog input destination routing determines what "units" the analog signal rep-resents. See the table "Page 11: Analog Input Items" for the relationship between Destination Routing and the type of units that each one represents.

Page and Items Used by the Analog Input

NOTE: For a complete discussion on how to program the MDII series of controls, please refer to the MDII Operating Manual section "Field Customizing the MDII".
On the next page is a table showing the various items involved with the analog input on the MDII. Note that all of the items reside on Page 11. The text of this applications note should help in determining the proper value to set each item to for a particular application.

HOOKING IT UP

Below is a drawing of the female RJ45 "modular " connector in the MDII, showing the pins that are used for analog input. For the sake of clarity, the other pins on this connector are not described here. See the MDII manual section "Communications and Networking" for descriptions of those pins.



RJ12 Pin	RJ45 Pin	Signal Description
5	6	Signal Common (Ground)
6	7	Analog Input (0 to +5VDC)

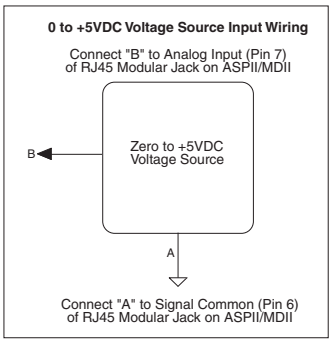
You will need an *eight* conductor RJ45 male as well as *eight* conductor modular cable to use the analog input. A six position RJ12 connector may also be used as an alternative. Your local computer store or Radio Shack should have all of the supplies you need to fabricate this cable, or should be able to fabricate one for you. Depending on the type of signal you will be using, you may also have to use an external resistor, or an external potentiometer. Wiring for the various signal input configurations is shown below.

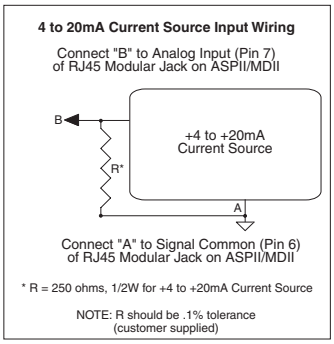
Wiring Diagrams for Each Source Type

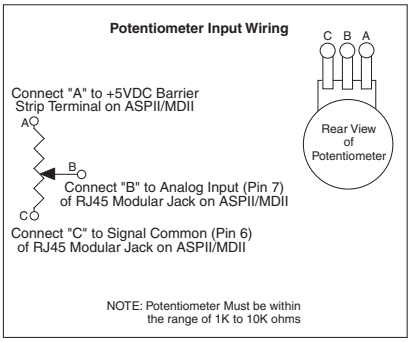
Below are three diagrams showing the basic input wiring necessary for each type of analog input source. You should find these helpful in determining the proper wiring for your application.

WARNING:

The *Absolute Maximum* signal range that can be safely applied to the analog input is from –0.5VDC to +20VDC. Signals outside this range can result in permanent damage to the MDII. This is *not* the maximum useful signal range, but rather the maximum overload capacity of the analog input. For proper operation, the *useful* signal range is from zero to +5VDC. Voltages that are outside of this range will be treated as the minimum and maximum values, respectively.







Troubleshooting Guide

If a newly installed control will not operate properly, it is possible that a terminal or connection is loose. Check to make sure that all wiring connections are secure and correct. If control still malfunctions see chart below.

Problem	Possible Causes	Corrective Action
Motor runs only at full speed	No signal from pick-up	Check pick-up wiring for correct wiring or loose connections. Check pick-up for proper operation
Motor doesn't operate	Blown fuse No AC power applied Target Speed set to 0 Worn Motor Brushes	Replace fuse Apply AC power Increase Target Speed Replace brushes
AC fuse blows upon power-up	Defective motor Accel Time too fast Motor overloaded	Replace motor Customize to a longer time Reduce load
Armature output voltage cannot be adjusted, output is a constant DC level	No motor or load connected Loss of pick-up signal	Check that motor or load is connected to armature terminals Verify incoming pick-up signal
Motor runs too slow	Worn motor brushes Incorrect customizing Incorrect pick-up signals due to electrical noise	Replace brushes Re-customize Route pick-up wires separate from AC and armature wires, use shielded wire on pick-up
Motor continually hunts	P-I-D set improperly	Adjust P-I-D
During customizing, the editor gets hopelessly lost and wishes to return to the factory presets		Load factory “defaults”. With AC power off, hold both the ENTER and ↓ buttons down and turn on AC power

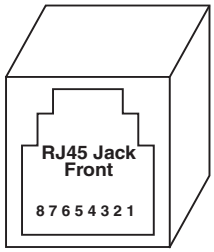
If problems persist consult your Dart Controls, Inc. Distributor or Representative

Using the Frequency Generator Output*

The ASPII/MDII has a frequency generator output that can be used as the leader signal to drive a network of followers. The ASPII/MDII that is generating this signal can even follow its own generator's output. This allows a ratio system to operate without a dedicated leader motor and pick-up. To follow this signal, connect pin 5 of the RJ45 connector to the spare input (P1-8) of the follower ASPII/MDII controls. This signal may also drive the master input of other Dart controls such as the ASP10, and standard MDP series.

The frequency generator output is available on pin 5 of the RJ45 connector. The common for this signal is available either from pin 6 of the RJ45 connector or from the main terminal block P1-5. Use only one of these common terminals to make electrical connections between devices. The output is an Open-Collector NPN transistor which is capable of sinking a maximum of 500mA up to 100 volts DC. The frequency of this output is set using Item 8 on page 7. The frequency is set in pulses-per-minute from 0 to 9999. To change this frequency using the front panel pushbuttons, Item 4 on Page 9 must be set to a value of 2 (for "code restricted" access) or 3, (for unrestricted access).
The RJ45 connector pin out is as follows:

RJ12 Pin	RJ45 Pin	Description for Frequency Generator Output
-	1	No Connect
1	2	No Connect
2	3	No Connect
3	4	No Connect
4	5	Frequency Generator Out (Do not use for RS232 Communications)
5	6	Circuit Common
6	7	No Connect
-	8	No Connect



NOTE: An RJ12 connector may be use in place of RJ45 connector. The 4 or 6 conductor cables will give you the signals shown for pins 3 through 6 or 2 through 7 above. This is sufficient when using the frequency generator output. Another important point is that most commercially available modular cables invert the wires going between the connectors. For example, pin 3's wire at one end connects to pin 6 on the other end, pin 4's wire connects to pin 5 at the other end, and so forth. Since most RJ's males connectors are transparent, you can easily look at the colors on the wires to see if this is the case. Improperly wiring the RJ12 or RJ45 connector will not damage the control, but improper operation will result.

* NOTE: The Frequency Generator Output on the ASPII/MDII can also be used as a "User Assignable Output" *instead of* a Frequency Generator Output. See the section "User Assignable Outputs" for details.

WARNING: DO NOT connect the ASPII/MDII directly to the phone system! Permanent damage to the control WILL result.

User-Assignable Output

The ASPII/MDII incorporates an advanced “User-Assignable Output”. This feature allows an editor to customize the *physical* output circuit normally used as a Frequency Generator Output, for use as a general-purpose output capable of driving LEDs, small relays, solid-state alarm modules, and the like. This output is an Open-Collector NPN transistor which is capable of sinking a maximum of 500mA up to 100 volts DC. See the section “Using the Frequency Generator Output” for more information about proper wiring of the User-Assignable Output. The output can be “connected” to *any combination* of up to eight different “conditions”, such as “Pickup Stalled”, “Upper Speed Limit Exceeded”, etc. by using its "Assignment Matrix" switches, "Programming" Page 12, Item 3. These conditions are the same ones found in the “Drive Status” Monitor Item, "Programming" Page 0, Item 4. Through the use of the "Inverter Matrix" switches, "Programming" Page 12, Item 4, the *opposite* of any condition(s) can be used instead ("Pickup *not* Stalled", "Upper Speed Limit *not* Exceeded", etc.) for even more flexibility. Additionally, the output can be set to either "Normally Open" (N.O.) or "Normally Closed" (N.C.) operation with the "Output Selection" Switch, "Programming" Page 12, Item 2. See the Appendix "Description of Item Functions" for details. This output arrangement provides the user with literally *hundreds* of possible combinations, while keeping the actual electronic complexity (and cost) of the control to a minimum.

There are two different ways that the "circuitry" of the User-Assignable Output can be expressed. For those familiar with digital logic symbols, please refer to the top figure on the next page for a "circuit" description. If you are more familiar with "Ladder Logic" symbology, please refer to the bottom figure instead for a "schematic" representation. However, *regardless* of which method you are more comfortable with, it is important to remember a few things:

- This "circuitry" is actually implemented in <u>software</u>, *not* <u>hardware,</u> and although it actually makes little difference to the final output produced, that fact should be kept in mind.
- Often there is more than one way to "connect-up" the signals and inverters to achieve the desired result; but NOTE: Signal "polarity" is *always* a tricky problem in "logic circuits". Sometimes it takes a bit of careful planning to arrange "inverted" and "non-inverted" signals properly.

Examples:

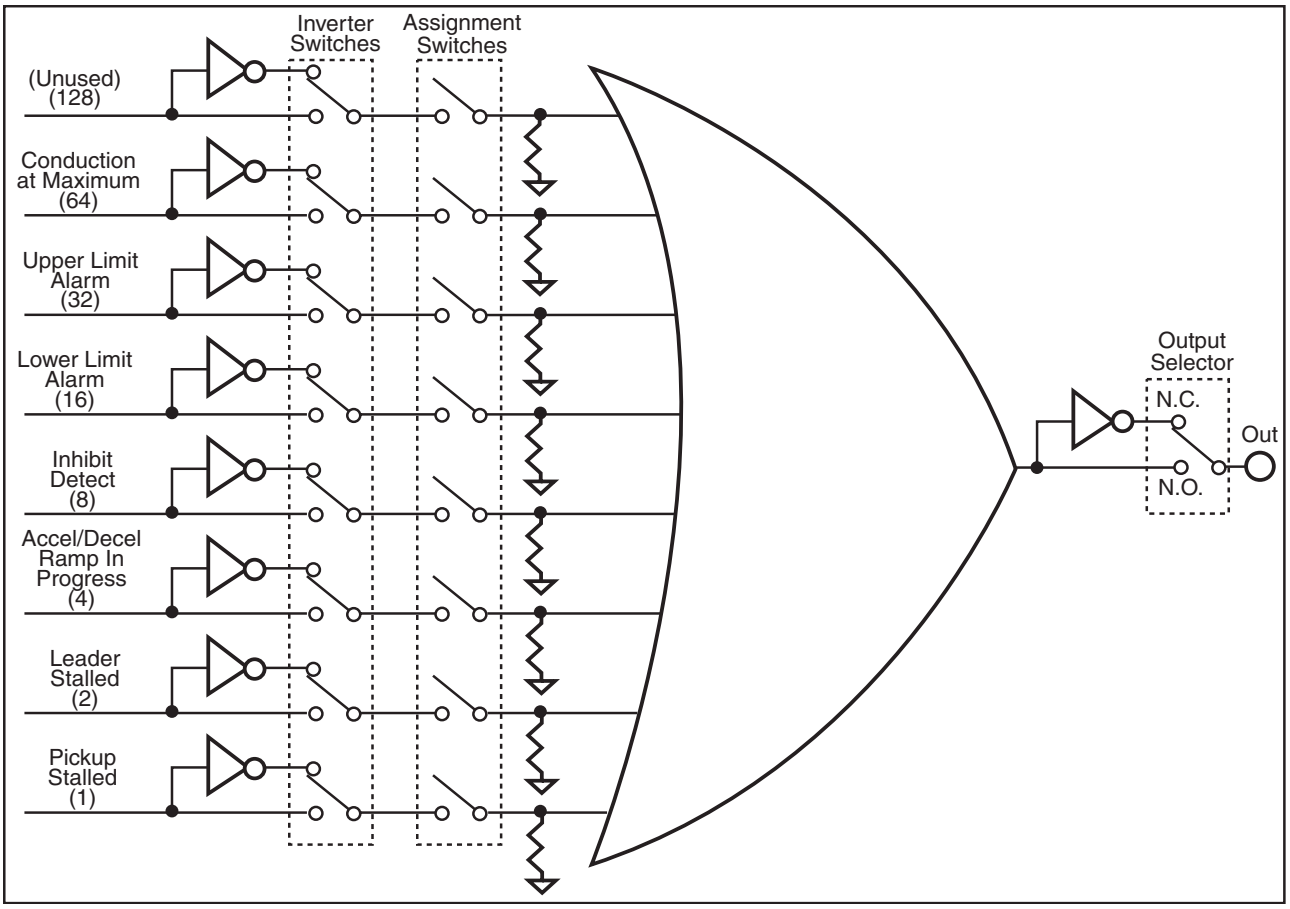
Problem: Setup Output to activate a Pickup Stall Detect Alarm.

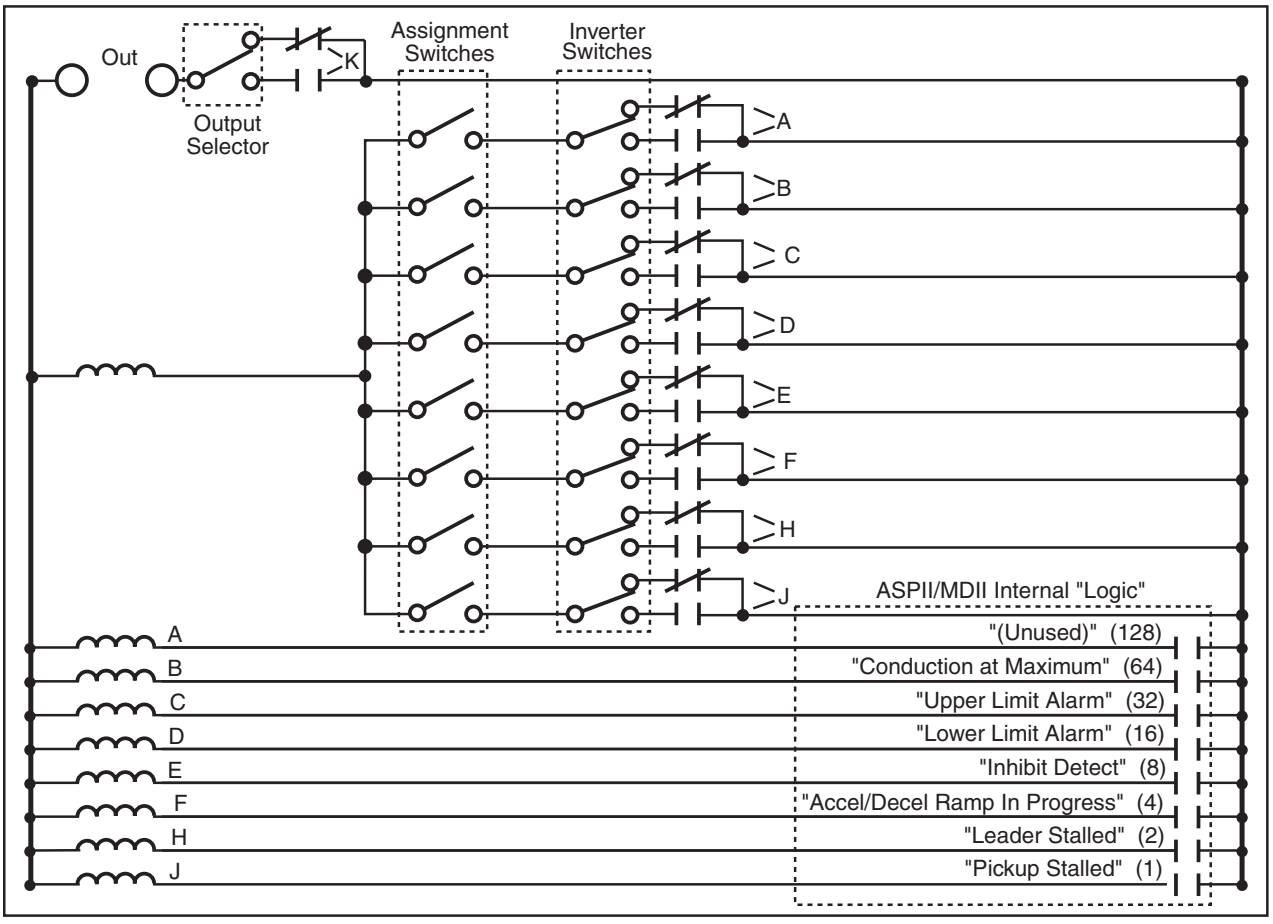
Solution: Set "Programming" Page 12, Item 3 (Output "Assignment Switches") to a value of 1 (selects the "Pickup Stalled" condition); set "Programming" Page 12, Item 4 (Output "Inverter Switches") to a value of 0 ("none inverted"); set "Programming" Page 12, Item 2 ("Output Selector") to a value of 0 (Output N.O.).

Problem: Setup Output to activate an Alarm when either the Lower Speed Limit *or* Upper Speed Limit has been exceeded.

Solution: Set "Programming" Page 12, Item 3 (Output "Assignment Switches") to a value of 48 (32+16, or "Upper Limit Alarm"+ "Lower Limit Alarm"); set "Programming" Page 12, Item 4 (Output "Inverter Switches") to a value of 0 ("none inverted"); set "Programming" Page 12, Item 2 ("Output Selector") to a value of 0 (Output N.O.).

User-Assignable Output Logic Diagrams





Setting SoftSwitches

Like many other devices, the ASPII/MDII has the ability to select between a number of "yes/no" or "on/off" options, depending upon the application. Traditionally, this sort of option-selecting is done with some sort of physical switch or switches (such as a DIPswitch assembly), or other means, such as the "jumper blocks" used to select RS485/RS232 data communication on the ASPII/MDII. There are two problems with this approach to option selection: 1) Both DIPswitch assemblies and "jumper blocks" are somewhat bulky, and most require that the device be at least partially disassembled to gain access to them; 2) On a device with more than just a very few options, the number and combinations of switches quickly becomes overwhelming.

Because of these drawbacks, the ASPII/MDII takes a different approach (where appropriate): SoftSwitches.

It is easiest to think of an Item containing SoftSwitches as a DIPswitch assembly containing from one to eight switches. But intstead of actually "flipping" a switch to the appropriate position, the editor (or user) can set and read these switches as a Binary-Coded-Decimal, or BCD number. Now before you say "Binary numbers! Those are for computers!", let's look at this another way: Each switch, from switch #1 through switch #8, has been assigned a decimal number that represents its position in the make-believe DIPswitch assembly. When that number is used, it means that switch is "on". For example, the decimal number that represents switch #4 is 8, the number that represents switch # 6 is 32, and so on. See the table below for a full explanation of these values.

Switch#	BCD Value
1	**1**
2	**2**
3	**4**
4	**8**
5	**16**
6	**32**
7	**64**
8	**128**

So, the Binary-Coded-Decimal number containted in a SoftSwitch Item is nothing more than the sum of the numbers representing the "on" switches. For example, if you wanted to set switches #1, #4, and #7 to the "on" position, you would place the number 73 (1 + 8 + 64) into the Item containing those SoftSwitches; if you wanted to set switches #5 and #6 "on", you would place the number 48 (16 + 32) into the Item, and so forth. Simply "add-up" the BCD values of the switches you wish to turn "on", and place the total, or "sum" into the Item containing the SoftSwitches.

You can also "read" the settings of SoftSwitches in the same way: If an Item containing SoftSwitches has the number 11 in it, you can tell that switches #1, #2, and #4 (or 1 + 2 + 8) are "on". You can tell this by *subtracting* the BCD values, from highest to lowest, *starting at the highest value that is less than or equal to the "total"*. Keep subtracting, but if you get a negative number as a result, then don't subtract that BCD value (add it back in before proceeding). Work your way "downward" toward switch #1, but when your total reaches *zero*, you are finished.

Try a few examples of your own, and you will soon get the hang of setting and reading SoftSwitches.

Communications and Networking

The ASPII/MDII can be connected through its built-in RS232/422/485 serial port to a terminal, computer, process controller, other controls, or various other devices to greatly expand its ability to monitor, control and report in many ways. The best thing about the ASPII/MDII (and the DartNet network) is that it only requires an ordinary serial port, rather than special network hardware or "transporter" cards, to establish communications. A convenient "jumper block" in the ASPII/MDII is used to select either RS232 or RS422/485 communications. This section discusses the hardware and software issues involved in communicating with the ASPII/MDII.

Commonly Used Software

When using a computer to configure the ASPII / MDII series controls, a few things are needed. First, a computer with either an RS-232 or RS-485 communications port must be located. On IBM-compatible computers, this is typically an RS-232 port and is located on the back of the machine. It will be either a DB-9 or DB-25 male connector. Second, a software package that allows the computer to easily communicate with selected serial port must be obtained. Fortunately, there are a number of available packages that are capable of filling this need. Windows 3.1 is shipped with a free terminal program named Terminal. Windows 95, 98, and 2000 are shipped with a package titled HyperTerminal. Although these included packages will work, Procomm, a commercially-available package, does a far superior job and offers more features to the user.

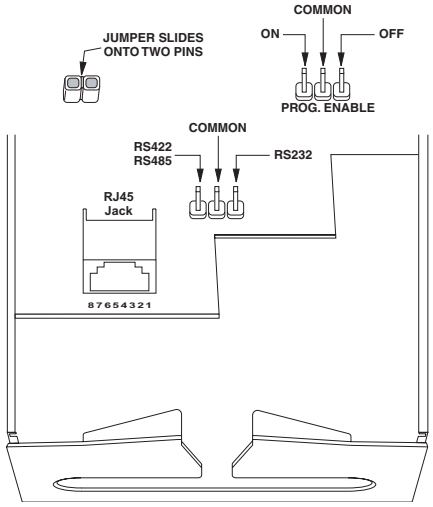
Regardless of the chosen software package, it will be necessary to configure the software appropriately to allow it to communicate with the ASPII / MDII series unit. These products ship with their serial communications port set at 300 baud, 8 data bits, and 1 stop bit. **Setting the terminal program to half duplex——sometimes called local echo——will allow the user to see the text as they type. It is also very important that the flow-control setting be set to OFF.** Read the next several pages for more information on serial communications.

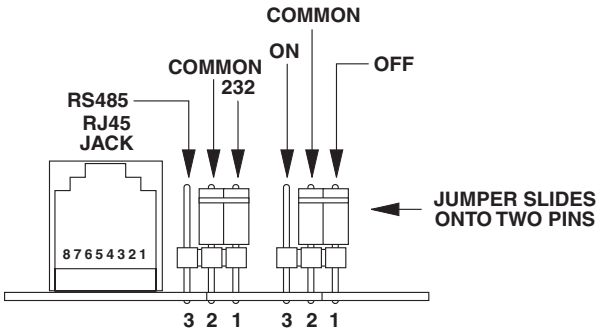
Hardware - Jumper Selection

RS232 - RS422/485 Jumper Selection

To choose between RS232 or RS422/485 communications, you must make sure the 232/422/485 "Jumper" is in the correct position. The drawing showing the location of this jumper appears below.

REAR VIEW OF MD20P, ASP20, AND MD30P





Once you have found the jumper, move it like this:

RS232: Place the "jumper block" on pins 1 & 2 (farthest from the RJ45 Jack)
RS422/485: Place the "jumper block" on pins 2 & 3 (closest to the RJ45 Jack)

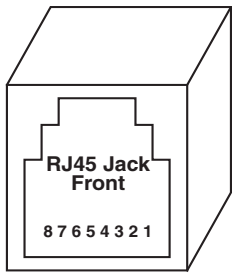
Connect the cable to the RJ45 "Modular" Jack. Now that you have selected which communications standard you wish to use, see following pages to help you wire the Modular Cable properly for RS232 or RS422/485 operation.

RS232 and RS422/485 Connections to the RJ45 Modular Jack

The ASPII/MDII RS232 and RS422/RS485 port uses a standard 8 pin RJ45 modular connector. A standard 4 or 6 pin RJ11 or RJ12 modular connector may also be used as an alternative when making RS422/485 connections. When using a RJ11 or RJ12 cable, that has 4 or 6 conductors, with the RJ45 Modular Jack on the ASPII/MDII, you are only connecting to the middle 4 or 6 pins of the RJ45 Modular Jack. A 4 pin modular connector uses the middle 4 pins, numbers 3 through 6 and the 6 pin modular connector uses the middle 6 pins, numbers 2 through 7. The use of an 8 pin RJ45 modular connector will be required when making connections for RS232.

There is currently NO standard wiring format for a RJ45 Modular connector when used in RS232 or RS422/485 applications. We have chosen to allow the use of standard 4, 6, or 8 pin modular connector in our wiring scheme. The RJ45 Modular Jacks pinout for the ASPII/MDII is as follows:

RJ11 Pin	RJ12 Pin	RJ45 Pin	RS232 & RS422/485 Signal Description
-	-	1	RS232 Receive Data (Data to Control)
-	1	2	No Connect
2	2	3	TxD+, RxD+ (+ transmit/receive data line)
3	3	4	TxD-, RxD- (- transmit/receive data line)
4	4	5	No Connect
5	5	6	Circuit Common, Signal Ground
6	6	7	No Connect
-	-	8	RS232 Transmit Data (Data from Control)



It is important to understand the minor differences between RS-232 and RS-422/485. Simply put, RS-232 is a point-to-point interface standard which is intended to allow two——and only two——devices to be attached to one another; whereas, RS-422 and RS-485 are multi-point interface standards supporting as many as 32 devices on the same lines or bus. Another key difference is that RS-422 and RS-485 are considerably more tolerant to noisy environments. With this in mind, it is typically not recommended that RS-232 be run more than 50 feet. The vast majority of computers in the world have only an RS-232 port. Specialty converters can be purchased, however, for under a hundred dollars to allow an RS-232 device to interface with one or more units on a shared multi-point bus. Once the desired interface standard has been selected, jumper JP2 should be positioned accordingly. Failing to do so will not damage the drive, but will result in a lack of communication.

When connecting a device with an RS-422 or RS-485 port to an ASPII / MDII series unit, the multi-point bus should be connected in half-duplex mode. In this mode, the four bus wires are connected in two pairs to the unit. Specifically, the Transmit+ and Receive+ are connected to the unit's RJ45 pin 3. The Transmit- and Receive- are connected to the unit's RJ45 pin 4. Under some circumstances, it may also be necessary to connect the unit's Signal Ground, RJ45 pin 6, to the remote device's Signal Ground. In this configuration, the RJ45 pins 1 and 8 should remain disconnected.

When connecting a computer with an RS-232 port to an ASPII / MDII series unit, not all of the available wires have to be used. If a port with a DB-9 (9 pins) connector is used, then the following connections should be made: DB9 pin 2 to RJ45 pin 8, DB9 pin 3 to RJ45 pin 1, and DB9 pin 5 to RJ45 pin 6. If a port with a DB-25 (25 pins) connector is used, then the following connections should be made: DB25 pin 2 to RJ45 pin 1, DB25 pin 3 to RJ45 pin 8, and DB25 pin 7 to RJ45 pin 6. In this configuration, the RJ45 pins 3 and 4 should remain disconnected.

When communicating with a computer, there are a number of pitfalls to overcome. Initially, it is important to make sure the terminal program is communicating with the correct port (ie: COM1, COM2, etc.). This can be difficult because the port numbers are not labeled on the back of most machines. Use the following outline when troubleshooting new connections:

1) Set the terminal program to 300 baud, 8 data bits, no parity, 1 stop bit, and half duplex.
2) Set the terminal program to the selected serial port: COM1, COM2, etc. (guess if unknown)
3) If necessary, unplug the cable from the selected serial port on the back of the computer.
4) Regardless of DB-9 or DB-25, carefully short pins 2 and 3 together on the computer.
5) Enter the following string of characters in the terminal window: 123123
6) 112233112233 should be displayed in the terminal window. If 123123 was displayed instead, it is likely that the wrong serial port has been chosen. Go back to step 2.
7) Now that the correct port has been located on the computer, remove the short from pins 2 and 3.
8) Connect the cable to the back of the computer, but do not connect the other end to the ASPII / MDII series control. Instead, temporarily short pins 1 and 8 of the RJ-45 plug on the cable.
9) Enter the following string of characters in the terminal window: 123123
10) 112233112233 should be displayed in the terminal window. If 123123 was displayed instead, it is likely that either the cable is bad or pins 1 and 8 are not shorted correctly. Go back to step 8 and test again.
11) At this point, the computer is configured and cable is probably wired correctly. Remove the temporary short and plug the cable into the drive's modular jack.
12) Type the following command into the terminal window (using all capital letters): SP1,0 (followed by pressing the computer's Enter key)
13) If the unit responds with Y or N, then the connection is complete and functional; otherwise, it will be necessary to double-check the wiring and repeat some of the above tests.

NOTE: Many of the commercially available modular connectors cables you buy off the shelf will invert the wires going between the two connectors on the ends of the cable. Improperly wiring the serial ports will not damage the control or the communications peripheral but the communications will not work. In case of difficulty, Dart's Technical Support Department will be happy to help you, or ask your computer dealer for assistants.

WARNING: DO NOT connect the ASPII/MDII directly to the phone system! Permanent damage to the control WILL result.

Software - Data Format

The ASPII/MDII uses a standard serial interface protocol found on most, if not all, computers, process controllers, terminals, etc. The format for data is as follows:

Start Bits:	1
Data Bits:	8
Stop Bits:	1
Parity:	NONE
Duplex:	Half (No Echo)
Baud Rate:	Selectable (see below)

There are also three *items* on page 10 that are involved with communications. To avoid confusion, we suggest that you do not program these items through the serial port (although that is possible). The *items* are as follows:

Item 1, Page 10: **Network Address**. This can be a value from 0 to 99 . Setting this item to "0" will cause the ASPII/MDII to ignore commands sent to any address. If you have more than one ASPII/MDII on your network, you probably want to set each unit to a unique address. By the way, commands sent to address "0" will be acted upon by every ASPII/MDII on your network. This can be quite helpful for example, to do an "All Stop" command.

Item 2, Page 10: **Baud Rate**. This can be a value from 1 to 5. A table showing the baud rates for the possible values of this item is shown below:

Value	Baud Rate
1	300
2	1200
3	2400
4	4800
5	9600

Obviously, all of the members of a network should have the same baud rate.
NOTE: Baud Rate changes only take effect upon power-up.

Item 3, Page 10: **Transmit Turnaround Delay**. Sets delay time before ASPII/MDII responds back to external device. Ranges from 1 to 50 in 10ms increments.

Format for Messages

NOTE: First a word about the acceptable (to the ASPII/MDII) letters and numbers. The ASPII/MDII has been designed to reject all but the following:

- The Numbers from 0 thru 9
- UPPERCASE ONLY letters from A thru Z
- The "comma" (,) character (used to separate different parts of a message)
- The "carriage return" character (used to end each message)
- For "Name Program" only, the symbols - ! ? ½ = (space)

ALL OTHER CHARACTERS WILL BE IGNORED! In the case of numbers with a decimal point (or colon) in them, do not send the decimal point (or colon). The ASPII/MDII "knows" the proper format for each number, and will internally "insert" the decimal place (or colon) correctly. For example, if the number you wish to send is in the format "nn.nn" (or "nn:nn"), simply send "nnnn". Also, it is not necessary to send any "leading zeros" for a number. "0001", "001", "01", and "1" are all the same to the ASPII/MDII.

All messages sent to the ASPII/MDII follow a common structure, or "format". The command message format is shown below:

<command><address>,<"command stuff"><CR>

<command> A two-letter UPPERCASE command. There is NO comma between the command and the <address>

<address> A number (0 thru 99) representing the device that this message is intended for. Using an address of "0" will cause all devices to react to the message. This number is followed by a comma.

<"command stuff"> The portion of the command message that is unique to each command. See the details of the commands for this information. There is NO comma after the end of this portion of the message, although there may be commas within "command stuff".

<CR> A "Carriage Return" character (ASCII 13 decimal {0D hexadecimal}). This means the end of this message. This is the character you get when you press "Return" or "Enter" on your computer keyboard. Some computers also send a "Linefeed" character (ASCII 10 decimal { 0A hex}). The ASPII/MDII will ignore linefeeds sent to it, and it will not send linefeeds after <CR> on its outgoing messages.

example: "Set the speed of the ASPII/MDII addressed as "1" to 1000"
format: **SP1,1000**

Remember, there is "carriage return" character at the end of the message.

Direct Commands from a Computer/PLC

There are five commands implemented on the ASPII/MDII; however, these five commands will allow you to do quite a number of things (commands must be in upper case). These commands are discussed below and on the following page:

SP Set Target Speed (or Time in seconds)

Description: Allows the setting of a desired speed (or Process Time)

Format: **SP<address>,<speed><CR>**
<speed> is usually scaled in Engineering Units; however, in Follower modes, the "speed" setting is always expressed as a "percent of leader". Additionally, **<speed>** is expressed as seconds when in Time mode. The current program's "Setting Limits" determine the lowest and highest values you can use for **<speed>**.

Example: Set the speed of the ASPII/MDII addressed as "3" to 500 ("whatevers")
Format: **SP3, 500<CR>**

Response if successful: **Y<CR>**
Response if unsuccessful: **N<CR>**

NOTE: The Set Target Speed commands are not stored to non-volatile memory, consequently upon power-up they will revert to the previous number stored in NOV-RAM. If you need these commands to be stored you must use the Set Variable command to set the desired target speed or program. Also the Read Variable command will show *Page 7, Item 4* as the number stored in NOV-RAM, which may not be the currently set value.

SV Set Variable

Description: Allows the setting of any *item* on any *page* to any *value* within its range.

Format: **SV<address>,<page>,<item>,<value><CR>**
<page> and **<item>** and **<value>** are the same as if you were editing from the ASPII/MDII's front panel. (See "Appendix A: Index to the ASPII/MDII Memory" for details).

Example: Set the ASPII/MDII addressed as "6" to display *"actual" speed (tach)*
Format: **SV6, 7, 1, 2<CR>**

Response if successful: **Y<CR>**
Response if unsuccessful: **N<CR>**

NOTE: This command can actually be used to great advantage. For example, sending the message **SV0, 8, 1, 3** to a network of controls with multiple *programs*, would effectively switch all controls on the network to that *program* at one time. The most obvious example of such a usage might be in a multiple-auger materials blender.

NP Name Program

Description: Applies custom name to a program

.

Format: **NP<address>,<program#>,<4 character name><CR>**
The program name must be 4 characters long, no more or less. A space may be used for a blank position.

Example: Name program 3 of, the ASPII/MDII addressed as "2","FAST"
Format: **NP2, 3, FAST<CR>**

Response if successful: **Y<CR>**
Response if unsuccessful: **N<CR>**

(continued on following page)

CP Change Program

Description: Allows change of current program

Format: **CP<address>,<program#><CR>**
If programs have been named, you must still use the program # for this command.

Example: To change the unit addressed as 2 to program # 3, use CP2,3<CR>. The program you are seeing must be within the range set on *Page 8 Items 2 and 3*.
Format: **CP2, 3<CR>**

Response if successful: **Y<CR>**
Response if unsuccessful: **N<CR>**

NOTE: The Change Program command is not stored to non-volatile memory, consequently upon power-up it will revert to the previous number stored in NOV-RAM. If you need these commands to be stored you must use the Set Variable command to set the desired target speed or program. Also the Read Variable command will show *Page 7, Item 4* as the number stored in NOV-RAM, which may not be the currently set value.

RV Read Variable

Description: Outputs the *value* of any *item* on any *page* plus its decimal point location.

Format: **RV<address>,<page>,<item><CR> Itemq**
<page> and **<item>** are the same as if you were editing from the ASPII/MDII's front panel. The exception to this is Page Zero "Monitor Items". These are ONLY available through the Serial Port (See "Appendix A: Index to the ASPII/MDII Memory" for details).

Example: Read the ASPII/MDII addressed as "2" for the actual speed of the pickup, in RPM
Format: **RV2, 0, 2<CR>**

Response if successful: **<value>,<dec. pt.><CR>**
Response if unsuccessful: **N<CR>**

NOTE: In the response from the ASPII/MDII, <value> and <dec. pt.> will ALWAYS be transmitted as 4 digits, with leading zeros present. <dec. pt.> is the location of the displayed decimal point for that variable. Its values are: xxxx. = 0, xxx.x = 1, xx.xx = 2, xxxx = 4.

Downloading Data Between Two Units

Downloading Custom Values Between Two ASPII/MDII's or Between an ASPII/MDII and a Computer - If you have several controls to be customized, a quicker way of doing it is to use a customized control as a "master" and load its data into the other controls. This is done by setting the controls up on a Local Area Network via the serial interface port using an RS422/RS485 configuration with all ports set at RS485. Apply AC power to all controls. There is no need to connect a motor or pick-up for this procedure. On all units to be customized, select Load under the File menu, then depress and release the Select **ENTER** button one time. Next on the "Master" control select "Save", under the file menu, then depress the select **ENTER** button. After three seconds all units will begin to count from 0 to 511. When they reach 511, if all the information has been transfered correctly, the controls will reset to the new configuration. They may then be disconnected from the network and installed in your application. If a unit displays **(err)**, this indicates an error occurred during this procedure. On this unit only, turn off the AC power. Depress and hold both the **ENTER** and ↓ and apply AC power, then repeat the above procedures.

Appendix
Description of Item Functions

Pages 1-6

Items 1 and 2:

When operated as a Master Rate or Master Time control, the ASPII/MDII can be customized to display/control using units other than the standard "RPM" unit[1]. Additionally, these units can be programmed differently for each of the 6 *programs* in the ASPII/MDII. The "*Display Reference Value* " and *"Display Reference RPM"* can be found on *pages* 1 thru 6 (for *programs* 1 thru 6), as *items* 1 and 2 (see Index to the ASPII/MDII Memory) for details.

- ***Item* 1 *Display Reference Value*** is the number to be displayed in the window at a given motor RPM.
- ***Item* 2 *Display reference RPM*** is the speed (in RPM) that the pick-up is turning at the *Display Reference Value*

Example:

A desired setting of 14.00 feet/min. (1400) would give us 1275 *RPM* at the pickup

***Display Reference Value* = 1400 (Ignore Decimal Points, i.e "14.00" would be considered "1400")**

Display Reference RPM* = 1275 *RPM* (not PPM) of *Pickup* at the *Display Reference Value.

NOTE: The *Display Reference Value* and *Display Reference RPM* range is from 1 to 9999

Items 3, 4, and 5:

See "Customizing P-I-D" on page 15

Item 6:

Acceleration Time is set via this item value. A value of 0 defeats Accel. Decreasing the value increases the Accel time (1 = maximum accel; 9999 = minimum accel).
Note: In "follower" mode the accel/decel is automatically defeated.

Item 7 and 8:

The upper and lower limits of the target speed (time) are entered using these item values.

Item 9:

The displayed Decimal point is set by the value of item 9. A value of 4 = [0000], 3 = [0.000], 2 = [00.00], 1 = [000.0], 0 = [0000].

Item 10:

Display Blanking Place is determined by entering this item's value. The value is equal to the number of leading zeros ie. a value of 3 would display "1" as [0001].

Item 11:

Programmed Speed Setting For This Program. Sets a predetermined target speed in "Multi-mode", (may be over-ridden by value of Item 3, Page7.

Item 12:

Deceleration Time is set via this item value. A value of 0 defeats Decel. Decreasing the value increases the Decel time (1 = maximum decel; 9999 = minimum decel).
Note: In "follower" mode the accel/decel is automatically defeated.

Page 7

Item 1 Display Mode:

Selects the information shown on the display when in run mode. 1= Target (or Set) Speed, 2= Actual Speed, 3= Program Name.

Item 2 Operation Mode:

Sets the Mode of Operation. 1= Master Rate, 2= Master Time, 3= Standard Follower.

Item 3 Target Setting Source:

Instructs the ASPII/MDII where to look for Target Speed information upon power-up or program change. 1= use "current" setting (return to setting used previous to power-down), 2= Use "program setting" (determined by Item 11, Pages 1-6).

---

[1] In Follower Modes, the CONTROL unit is ALWAYS "Percent of Leader", but the *Display Reference Value* and *Display Reference RPM* DOES affect the Tach display. This feature allows you to monitor the follower as RPM, percent of master, or whatever engineering units are desired.

(continued on following page)

(page 7 continued)

Item 4 Current Target Setting:
This Item is where the "current" speed setting is placed.

Item 5 Initial Stall Timeout:
When starting the ASPII/MDII a delay time is preset here to prevent stall indicators. Value may be set in seconds from 2.0 to 25.0

Item 6 Running Stall Timeout:
Determines the time (in seconds) between a loss of pick-up pulses due to stalled motor and display of stall detect "second motor shutdown". May be set from 0.0 (defeat) to 20.0 seconds.

Item 7 Pick-up Pulses Per Revolution:
Enter the number of pulses per revolution here. Value may be from 1 to 2500; maximum input rate is 1,200,000 pulses per minute.

Item 8 Frequency Generator Output Rate:
Determines output frequency (in pules per minute) of the ASPII/MDII frequency output. This output is found on the modular jack pin 4. May be set from 0 (defeat) to 9999.

Item 9 Leader Running stall Timeout:
Determines the time (in seconds) between a loss of leader pulses and display of leader stopped message. May be set from 0.0 (defeat) to 20.0 seconds. Note: The leader initial stall detect is automatically defeated.

Item 10 Leader Pick-up Pulses Per Revolution
Enter the number of pulses per revolution of leader pick-up here, value may be from 1 to 2500 (maximum input rate is 1,200,000 pulses per minute).

Item 11 Output Operation (Not used in MDII Series)
Corresponds to the power supply of the contol being driven. For a positive supply use a value of 0, for a negative supply (inverted output) use value 1.

Item 12 Actual Speed (Tach) Display Update Rate
Allows the user to "trade off" display stability for faster display update rate (in seconds) as desired for a particular application. May be set from 0.2 to 8.0 seconds.

Item 13 Follower Startup Lag Compensation for *Slow-Starting* Leaders
Can be used to help overcome follower startup delay when used with Leaders with extremely slow startup rates. May be set from 0 (no compensation) to 9999 (ridiculous amount of compensation).

Page 8

Item 1 Current Program number:
Picks the Program currently being used. May be set from 1 to 6 but should be within limits of Items 2&3 below.

Item 2 Minimum Program #:
Sets the minimum program # which may be selected in run mode. May be set 1-6, must be - Item 3 value.

Item 3 Maximum Program #:
Sets the maximum program # which may be selected in run mode. May be set 1-6, must be • Item 2 value.

Page 9

Item 1 *Front Panel* Program Change access:
Selects the access to user program changes. 0=No Access (locked), 1=Full Access (no restrictions), 2="Code" required

Item 2 *Front Panel* Target Setting Change access:
Sets the user access to Target Speed changes. See Item 1, above, for an access level explanation.

Item 3 *Front Panel* Frequency Generator rate change access:
Determines the access privileges to the frequency generator rate setting. See Item 1, above, for an access level explanation.

Item 4 Pushbutton Default Destination:
This Item selects how a ¬ or pushbutton depression is acted upon. A value of 1= change Target Speed, 2= Change between programs, 3=change frequency generator rate.

Item 5 Security Code for restricted *Front Panel* Access:
Enter desired security code here. A value of 0 bypasses code. A value of 1-9999 is the security code that is entered when requested.

Item 6 *Front Panel* Function Lockout Switches:
This Item value selects which menu functions are accessible from the front panel. Each function has a BCD value.The sum of the BCD values, of the items to be accessible, is entered for item 6. BCD values are: Set=1, Run=2, Disp=4, File=8, Edit=16. A value of 31 sets all functions active, a value of 16 would make only the edit function available, 12 would give access to display and file functions.

Page 10

Item 1 Network Address:
Sets the address for network identification. A setting of 0 *ignores* all incoming messages.

Item 2 Baud Rate:
Sets the Baud Rate used. A value of 1= 300, 2= 1200, 3= 2400, 4= 4800, 5=9600.

(continued on following page)

(page 10 continued)

Item 3 Transmit Turnaround Delay

Sets the delay before the ASPII/MDII transmits a response to any incoming message addressed to it. Values are in 10 millisecond increments, from 1 to 50 (10 to 500ms). This allows an external device enough time to switch from transmit to receive mode, and communication wires to "quiet down"

Page 11

Item 1 Analog Input Destination Routing:

This Item selects how the analog input is acted upon. A value of 0= None, 1= change Target Speed, 2= change % of Target Speed, 3= change Program #, 4= change frequency generator rate, 5=main tach signal, 6=leader tach signal.

Item 2 Analog Input Source Type:

1= Potentiometer, 2= 0 to +5 volt signal, 3= 4 to 20mA current.

Item 3 Lower Limit:

Sets the numeric value which will be created when the the analog input is at its *lowest* voltage (or *lowest* current for 4-20ma input source type).

Item 4 Upper Limit:

Sets the numeric value which will be created when the the analog input is at its *highest* voltage (or *highest* current for 4-20ma input source type).

Page 12

Item 1 (For Future Expansion):

This item has been set aside for future expansion. For now, it *must* be set to a value of 1.

Item 2 User Output N.O./N.C. Selection:

Determines whether User Assignable Output is Normally Open (0) or Normally Closed (1).

Item 3 User Output Assignment Switches:

Selects the Status Condition(s) that will be used as "inputs" for the User Assignable Output. A value of 0 means *no* status conditions selected. NOTE: See Page 0, Item 4 for an explanation of the values to use.

Item 4 User Output *Input* Inverter Switches

When used *along with* Page 12, Item 3, this item allows the user to *invert* one or more Status conditions that have been assigned as inputs for the User Assignable Output. A value of 0 means *no* condtions are inverted. NOTE: See Page 0, Item 4 for an explanation of the valudes to use.

Page 0

Item 1 Device Type:

Displays device for which the software is programmed.

Item 2 Software Revision Level:

Shows revision level of currently installed software.

Item 3 DartNet Revision Level:

Indicates revision level of communications software.

Item 4 Drive Status:

This item can be used to check a number of conditions in the drive. Each condition has a BCD value. The *sum* of the Status BCD values is available for monitoring at this item. The values for each condition are: 1=Main pick-up stalled, 2=Leader input stalled, 4=Accel/decel ramp in progress, 8=Drive inhibited, 16=Lower limit exceeded, 32=Upper limit exceeded, 64=Conduction angle at maximum

Item 5 Pick-up Speed:

Used to monitor speed in RPM (*not* averaged over Display Update Rate).

Item 6 Pick-up Speed:

Used to monitor speed in Engineering units.

Item 7 Leader Speed:

Used to monitor Leader speed in RPM.

Index to the ASPII/MDII Memory

Page 0: Monitor Items (READ ONLY, NOT available from Front Panel of ASPII/MDII)

Item	Description	Values or Limits	Units
1	**Device Type**	1021 = MDII 1041 = ASPII	NONE
2	**Software Revision Level**	2001 - 2999	NONE
3	**DartNet Revision Level**	3001 - 3999	NONE
4	**Drive Status** (0 = Everything's O.K.)	0 - 255	Flags
5	**Pickup Speed** (NOT averaged over 1 second)		RPM
6	**Pickup Speed**		Engr. Units
7	**Leader Speed**		RPM

Page 1 to 6: Motor Control Items for Programs 1 to 6

Item	Description	Values or Limits	Units	Defaults
1	Display **Reference Value**	1 - 9999	NONE	1
2	Display **Reference RPM**	1 - 9999	RPM	1
3	**Proportioner** Stability Gain	0.001 - 9.999	NONE	0.400
4	**Integrator** Stability Gain	0.001 - 9.999	NONE	0.031
5	**Differentiator** Stability Gain	0.001 - 9.999	NONE	0.002
* 6	**Acceleration** Time (see table below)	0, 1 - 9999		0
7	**Lower "Setting" Limit**	0 - 9999	Engr. Units	0
8	**Upper "Setting" Limit**	0 - 9999	Engr. Units	3600
9	**Displayed Decimal Place** (Setting "Time" modes will force a Colon to be displayed instead)	0 = xxxx. 1 = xxx.x 2 = xx.xx 3 = x.xxx 4 = NONE	NONE	4
10	Display **Blanking Defeat Place**	0 = [""""0] 1 = [""00] 2 = ["000] 3 = [0000]	NONE	0
11	**Programmed Speed Setting** for *this* Program (See Page 7, Item 3)	0 - 9999	Engr. Units	0
* 12	**Deceleration** Time (see table below)	0, 1 - 9999	NONE	0

* **Accel/Decel Value Table (per 1000 RPM change)**

display value	1	3	7	20	40
accel/decel	30 seconds	15 seconds	7 seconds	3 seconds	1 second

Page 7: Other Control Items (Used by *ALL* Programs)

Item	Description	Values or Limits	Units	Defaults
1	**Display Mode**	1 = Target Speed 2 = Actual Speed 3 = Program Name	Engr. Units Engr. Units NONE	1
2	**Operation** (Must restart ASPII/MDII after changes are made from Master to Follower (or vice versa). To restart, shut off then re-apply AC power to unit).	1 = Master Rate 2 = Master Time 3 = Standard Follower	NONE	1
3	**Target "Setting" Source** upon power-up or Program Change	1 = Use "current" setting 2 = Use "program" setting	NONE	1
4	***"Current"* Target Setting** May be overridden by Item 3, above	Lower & Upper Limits	Engr. Units	0
5	***Initial* Stall Timeout**	2.0 - 25.0	Seconds	025.0
6	***Running* Stall Timeout** 0.0 = Defeat	0.0 - 20.0	Seconds	000.0
7	**Pickup** Pulses per Revolution	1 - 2500	Pulses/Rev.	1
8	**Frequency Generator** Output Rate 0 = Disable	0 - 9999	Pulses/Min	0
9	**Leader Running Stall Timeout** (initial timeout =) 0 = Disable	0 - 20.0	Seconds	006.5
10	**Leader Pulses per Revolution**	1 - 2500	Pulses/Rev.	1
11	**Output Operation** (Not used in MDII Series)	0 = Normal Output 1 = Inverted Output	NONE	0
12	**Actual Speed (Tach) Display Update Rate**	0.2 - 8.0	Seconds	001.0
13	**Follower Startup Lag Compensation** (For Slow-Starting Leaders) (0= No Compensation)	0 - 9999	NONE	0

Page 8: Program Control Items

Item	Description	Values or Limits	Units	Defaults
1	**Current Program #** (Should be within the limits of items 2 & 3)	1 - 6	NONE	1
2	**Minimum Program #**	1 - 6	NONE	1
3	**Maximum Program #**	1 - 6	NONE	1

Page 9: Front Panel Access ("Lockout") Control Items

Item	Description	Values or Limits	Units	Defaults
1	**Program Change Access** (Must restart control after changes are made)	0 = Locked 1 = Full Access 2 = Sec. Code	NONE	1
2	**Target "Setting" Change Access** (Must restart control after changes are made)	0 = Locked 1 = Full Access 2 = Sec. Code	NONE	1
3	**Frequency Generator Rate Change Access** (Must restart control after changes are made)	0 = Locked 1 = Full Access 2 = Sec. Code	NONE	1
4	**Pushbutton *Default* Destination** (Can be Overridden by using the "Set" Menu)	1 = ↑ and ↓ buttons change *target* speed 2 = ↑and ↓ buttons choose a Program 3 = ↑ and ↓ buttons change *Freq. Gen.* rate	NONE	1
5	**Security Code** 0 = Security Code bypassed completely	0 - 9999	NONE	0
6	"Function Menu" **Function Lockout Switches** (See "Setting Softswitches" for details)	0 - 31	SoftSwitches	31

Page 10: Communication Control Items

Item	Description	Values or Limits	Units	Defaults
1	**Network Address** 0 = Ignore ALL Incoming Messages	0 - 99	NONE	1
2	**Baud** (Must restart ASPII/MDII after changes are made from Master to Follower (or vice versa). To restart, shut off then re-apply AC power to unit).	1 = 300 2 = 1200 3 = 2400 4 = 4800 5 = 9600	NONE	1
3	**Transmit Turnaround Delay**	1 - 50	10ms	1

Page 11: Analog Input Items

Item	Description	Values or Limits	Units	Defaults
1	**Analog Input Destination Routing**	0 = NONE (Analog OFF) 1 = Target Speed 2 = Percent of Target 3 = Program # (1 - 6) 4 = Freq. Gen. Rate 5 = Main Tach Signal 6 = Leader Tach Signal	 Engr. Units Engr. Units NONE Pulses/Min. RPM RPM	0
2	**Analog Input Source Type**	1 - Potentiometer 2 - 0 to +5 VDC Voltage 3 - 4 to 20 mA Current	NONE	1
3	**Minimum NUMERIC Value Sent Out by Analog Input**	0 - 9999	(See Item 1)	0
4	**Maximum NUMERIC Value Sent Out by Analog Input**	0 - 9999	(See Item 1)	0

Page 12: User-Assignable-Output Items

Item	Description	Values or Limits	Units	Defaults
1	**(Unused — Future Expansion)**	Must Be Set to 1	NONE	1
2	**Output Selector (N.O./N.C.)**	0 = Normally Open (N.O.) 1 = Normally Closed (N.C.)	NONE	0
3	**Input Selector Switches** (See "Setting SoftSwitches for details)	0 - 255 (See Page 0, Item 4 for switch "layout")	SoftSwitches	0
4	**Input Signal Inverter Switches** (See "Setting SoftSwitches for details)	0 - 255 (See Page 0, Item 4 for switch "layout")	SoftSwitches	0

Glossary

Actual Speed — The current rate of rotation of the pickup. This rate is expressed in engineering units, which are not necessarily Revolutions Per Minute. See also “Target Speed”; “Tach”

Algorithm — A way of doing something, like a process, or method

Blanking Point — For the display window of the ASPII/MDII, the digit position at which leading zero suppression stops. For example, with various “blanking points” chosen, the number “1” could be displayed [01], [001], [0001], or just [1]

Closed-Loop — 1) In general, any control system that employs negative feedback to adjust its operation. 2) In the ASPII/MDII, it is the “algorithm” by which the “actual speed” of the motor’s pickup is used in determining any adjustments to its speed necessary to maintain the “target speed” at any given time. See also “algorithm”; “actual speed”; “target speed”

Dart Control Language — Message types supported by Dart Controls, Inc. networkable products. These messages are used primarily for process control/monitoring. Not all products support all types of messages

DartNet — A communications standard under which Dart Controls, Inc. networkable products communi cate with one another, as well as process contollers and other computer equipment. This standard defines the allowable baud rates, basic message format, but not the message types themselves. See also “Dart Control Language”

Default — The “normal” setting of something. For example, the “default” pushbutton destination is the thing that changes, or the “destination of the change” when you press the arrow buttons. You could “override” the default in this case by using the “Set” menu

Derivative — The “D” portion of the P-I-D method of “closed-loop” control. It is used to develop a signal that is derived from the rate-of-change of the “actual speed” of the motor at any given time. The primary job of the “derivative” is to limit the “proportioner”’s tendency to overcompensate for a large instantaneous difference between “target” and “actual” speeds. See also “P-I-D”; ‘Closed-Loop”; “Integrator”; “Proportioner”; “Target”; “Actual Speed”

Display Window — The main display of the ASPII/MDII. The display window is a four-character, seven segment LED type, with right-hand decimal points, as well as a “colon” between the second and third characters.

Edit Mode — The operation mode of the ASPII/MDII available from the front-panel, that is responsible for looking at and/or changing the various “items” involved in “field-customizing” of the control. See also “Run Mode”; “Edit”; “Editor”

Edit, Editing — Looking at, and/or changing various “items” involved in determining the operation of ASPII/MDII for a particular job, or application. See also “Edit Mode”; “Editor”

Editor — The person responsible for field-customizing the ASPII/MDII for a specific job. This person will sometimes, although not often, also be the “operator” of the control. See also “Edit Mode”; “Ed iting”

Expansion Bus — A way for a device to increase its number of available inputs and outputs. This is not to be confused with a network, which is a more “formal” form of communication

Field Customizing — See “Editing”

Follower — In “Standard Follower” mode of operation, this is the device (or devices) that will “slave-to” or “follow” at some desired percentage, a rate signal that is supplied by the “leader” or “master” device. See also “Leader”; “Standard Follower Control”

Function Menu — The “top-most” menu on the menu “tree”. It is obtainable from the front-panel by holding the **"ENTER"** button for about two seconds. Unless it has been customized, the choices on the Function Menu are as follows: DISPlay, RUN, SET, FILE, EDIT. See also “Menu”

Integrator — The "I" portion of the P-I-D method of "closed-loop" control. It is the only part of the "P-I-D" loop that acutally maintains the desired, or "target", speed which neither the "proportioner" nor "derivative" can do alone. See "P-I-D"; 'Closed-Loop"; "Derivative"; "Proportioner"; "Target"; "Actual Speed"

Item — 1) Any particular setting which controls or customizes the behavior of the ASPII/MDII. You can think of "items" as make-believe knobs and switches. The "editor" adjusts these during the field customizing process to make the ASPII/MDII work in the desired way for the job. Similar or related items are grouped on the same "page". 2) The indicator that lights up on the ASPII/MDII in "edit mode" when it is prompting the "editor" to choose which "item" to "edit". See also "Page"; "Value"; "Edit Mode"; "Editing"; "Editor"

Leader — In "Standard Follower" mode of operation, this is the device which supplies the rate that the followers will "follow", at whatever percentage of that rate is desired. See also "Follower"; "Standard Follower Control"

Master Rate Control — The mode of operation during "run mode" in which the ASPII/MDII is set up to control and/or monitor the rate at which something is occuring, in whatever engineering units are desired. In this mode, the ASPII/MDII uses two numbers, the "target" speed setting, and the "actual" speed of the pickup for its basic information. See also "Master Time Control"; "Standard Follower Control"

Master Time Control — The mode of operation during "run mode" in which the ASPII/MDII is set up to control and/or monitor the time that a particular process is requiring in minutes and seconds, such as the time necessary to pass through a conveyor oven. In this mode, the ASPII/MDII uses two numbers, the "target" speed setting, and the "actual" speed of the pickup for its basic information. See also "Master Rate Control"; "Standard Follower Control"

Menu — A group of choices displayed as simple words, are selectable with the arrow pushbuttons. Menus are used on ASPII/MDII to allow the "editor" and "operator" easy access to various functions on the control

Non-volatile — That which does not go away when the power is shut off. The ASPII/MDII contains "non-volatile" memory to store its customized items, as well as such things as the "target" setting

Operator — The person that will actually be using the ASPII/MDII in a particular job. See also "Editor"

P-I-D — Stands for "Proportioner-Integrator-Derivative". What this mouthful of syllables describes is the method by which the ASPII/MDII attempts to control the speed of the motor to which it is attached. The P-I-D method, or "algorithm" of control was first designed to control furnaces in central heating systems, although it appears, in general, to be a good method for "closed-loop" control of almost anything. See Also "Algorithm"; "Proportioner"; "Integrator"; "Derivative"; "Closed-Loop"

Page — 1) A logical group of related "items", used in field-customizing the ASPII/MDII by the "editor". 2) The indicator that lights up on the ASPII/MDII in "edit mode" when it is prompting the "editor" to choose which "page" to "edit". See also "Item"; "Value"; "Edit Mode"; "Editing"; "Editor"

Parity — A serial communications term describing a single error-detection bit transmitted along with each character of information. The ASPII/MDII has a permanent setting of NO parity

Program — In the ASPII/MDII, a program describes one of six possible field custom setups, or configurations that can be used to select multiple preset speeds, drive-train ratios, and so forth. The up and down pushbuttons can even be used to choose from a "menu" by name which program is in use, if desired. See also "Default"; "Menu"

Proportioner — The "P" portion of the P-I-D method of closed-loop control. It is used to develop a signal that is proportional, or varies with, the difference between the "target" and "actual" speeds. It is primarily used to provide quick response to changes in the workload on the motor. See also "P-I-D"; "Closed-Loop"; "Integrator"; "Derivative"; "Target"; "Actual Speed"

RS232/422/485 — Standards published by the Electronic Industries Association which describe to some extent the physical, or wires and signal levels, specifications for three different serial communications methods: RS232C, RS422, and RS485. RS232 is the oldest of the three standards. It consists of non-balanced signal lines, rather than twisted pair cable, for its signals. RS232 was primarily intended to attach one device with one other device, in an exclusive arrangement. RS232 was not intended for networking, although with the proper hardware and software, it is possible to support a small (less than 10) member network over a short (less than 100 feet total) amount of cable. Both noise immunity and transmission ("baud") rate become a problem with larger networks. RS422 and RS485 are both standards that have a multi-member network in mind. Both use a balanced-line transmission system, intended primarily for use with twisted-pair cabling. This kind of system offers excellent noise immunity. These standards also allow for faster transmission rates, although with the short length of the average DartNet "message", this is not usually an issue. The primary difference betweeen RS422 and RS485 is the addition of an enable line on the transmitters for RS485. This enable line allows the RS485-equipped device to avoid loading the network signals down when not actually transmitting a message. This, in turn, makes it possible to have larger and longer length networks. It is always possible to put an RS485 device on a network with RS422 devices, but the reverse is not always possible

Run Mode — The normal mode of operation of the ASPII/MDII, as opposed to "Edit" mode. See Also "Edit"

Serial Port — A connector on a device provided to supply RS232/422/485 serial communications. Most computer equipment has an RS232 port, although some have RS422. See also "RS232/422/485"

Serial — A form of the physical portion of data transfer in which the information for each character is transmitted and received one bit at a time over a single conductor in a cable, as opposed to parallel, where each bit of the character is presented at the same time over many individual conductors. RS232, RS422, and RS485 are all serial transfer protocols. See also "RS232/422/485"; "Serial Port"

Standard Follower Control — The mode of operation during "run mode" in which the ASPII/MDII is set up to control a motor at a rate that is some percentage of a "leader" rate. In this mode, the ASPII/MDII uses three numbers, the "target" percentage setting, the "leader" signal's rate, and the "actual" speed of the pickup, for its basic information. See also "Master Rate Control"; "Master Time Control"; "Leader"; "Follower"; "Actual Speed"

Tach — 1) A device for measuring and displaying "Actual Speed". 2) The indicator that lights up on the ASPII/MDII when it is displaying "Actual Speed". 3) The "menu" choice available on the DISP "menu" for placing the ASPII/MDII in a display mode to show "Actual Speed". See also "Actual Speed"; "Menu"; "Function Menu"

Target, Target Speed — The desired speed (or process time) of the motor; what the ASPII/MDII is attempting to maintain. See also "Actual Speed"; "Tach"

Value — 1) The number that an "item" is set to. 2) The indicator that lights up on the ASPII/MDII under two conditions: a) In "edit mode", when ASPII/MDII is displaying the current setting of an "item", prompting the "editor" to either accept the displayed value, or "edit" it with the up and down pushbuttons; b) In "run mode", when the "operator" is changing the "target speed". See also "Page"; "Item"; "Edit Mode"; "Run Mode"; "Edit"; "Editor"; "Target Speed"

REPAIR PROCEDURE

In the event that a Product manufactured by Dart Controls Incorporated (DCI) is in need of repair service, it should be shipped, freight paid, to: Dart Controls, Inc., 5000 W. 106th Street, Zionsville, IN. 46077, ATTN: Repair Department. Please include Name, Shipping Address (no P.O. Box), Phone Number and if possible, e-mail address.

Those orders received from anyone without an existing account with DCI must specify if they will be paying COD or Credit Card (Master Card or Visa). This information is required before work will begin. If you have an account with Dart your order will be processed according to the terms listed on your account. Products with Serial Number date codes over 5 years old will automatically be deemed Beyond Economical Repair (BER). A new, equivalent device will be offered at a substantial discount.

Completed repairs are returned with a Repair Report that states the problem with the control and the possible cause. Repair orders are returned via UPS Ground unless other arrangements are made. If you have further questions regarding repair procedures, contact Dart Controls, Inc. at 317-733-2133.

YOUR MOTOR SPEED CONTROL SOLUTIONS PROVIDER

125D SERIES
AC INPUT - VARIABLE DC OUTPUT
1/50 HP through 1.0 HP

250G SERIES
AC INPUT - VARIABLE DC OUTPUT
1/50 HP through 2.0 HP

65 SERIES
DC INPUT - VARIABLE DC OUTPUT
CURRENT RATINGS OF 20, 40, AND 60 AMPS

700/COMMUTROL SERIES
DC BRUSHLESS
5 & 20 Amp for
12,24,& 36VDC Inputs

MDP SERIES
PROGRAMMABLE
CLOSED LOOP DC
SPEED CONTROL

DM SERIES
FIELD PROGRAMMABLE
DIGITAL TACHOMETER

Dart Controls, Inc. is a designer, manufacturer, and marketer of analog and digital electronic variable speed drives, controls, and accessories for AC, DC, and DC brushless motor applications.

Shown above is just a sampling of the expanded line of Dart controls that feature the latest in electronic technology and engineering. Products are manufactured in the U.S.A. at our Zionsville (Indianapolis, Indiana) production and headquarters facility - with over 2,000,000 variable speed units in the field.

In addition to the standard off-the-shelf products, you can select from a wide variety of options to customize controls for your specific application. For further information and application assistance, contact your local Dart sales representative, stocking distributor, or Dart Controls, Inc.

Dart Controls, Inc.

Manufacturer of high quality DC and AC motor speed controls and accessories since 1963.

P.O. Box 10
5000 W. 106th Street
Zionsville, Indiana 46077
Phone: (317) 733-2133
Fax: (317) 873-1105

www.dartcontrols.com

ISO9001:2000 REGISTERED