
Panasonic (Minas Series) AC Servo Drive

Application Note

This document describes how to configure a Paradigm operator interface terminal to allow communications with a Panasonic Minas Series AC Servo Drive. It describes the parameters that can be accessed and the controls actions that can be issued to the Drive. Please read this document carefully before attempting to configure communications with these devices.

Introduction

The EDICT-97 configuration software has been designed to allow the user to enter a Parameter mnemonic and Number if applicable in a manner that should be familiar to a user of a Panasonic Minas Series AC Servo Drive. The driver allows the exchange of data with the Drive, and control actions to be issued to the Drive.

Accessing Data

The Minas Series communications protocol allows access to a number of Parameters over a serial communications link. The driver described here supports a subset of these parameters and these are given in the table below.

Parameter	Range	Description
User	00..3E	Servo Parameter
System	00..08	Servo Parameter
Step	01..28	NC Parameters – Step Data
Velocity	00..15	NC Parameters – Velocity Data
Data	00..12	NC Parameters – NC Data
Offset	n/a	NC Parameters –Offset Data
Input	n/a	NC Parameters – Input Port
Output	n/a	NC Parameters – Output Port
Position	n/a	Current Position
Speed	n/a	Current Speed
Torque	n/a	Current Torque
Error	n/a	Current Position Error
Alarm	0..6	Alarm Contents
Hist	1..12	Alarm History
CmdError	n/a	Last <i>DeviceCommand</i> Error

In addition, the Step Data is composed of three sub elements, Position Data, Speed Select No, and Positioning Mode Select. These are accessible individually.

In addition, the Offset Data is composed of four sub elements, Origin Offset, Positive Software Limit, Negative Software Limit, and Auxiliary Information 1 and 2. These are accessible individually.

In addition CmdError gives the most recent error reported by the Minas Series Drive in response to a *DeviceCommand* request. The value gives the Error in the low byte and the Command in the high byte, of the low word.

Control Actions

Often control actions are required to be executed in a particular order in the Drive. EDICT 97 provides the function *DeviceCommand()* which when called will guarantee the command is issued to the Drive in the order that they occur in the terminal. It has the following syntax :

$$DeviceCommand(device, "command", parameter)$$

where :

device is the device number from the Device Table
command is the is a string interpreted in the following way
parameter is the value associated with the Command.

There are three distinct command string formats, described below. In all cases the C is the command, and M is the Mode, in Hexadecimal format.

$$DeviceCommand(D, "CMn", P)$$

When the third character is lower-case *n*, the parameter number *P* is included in the command issued to the Drive. This is most applicable for a Step Command where the parameter is the Step Number.

For example, the following command *DeviceCommand(1, "60n", 3)*, would result in the Stepping command with step No 3 being sent to the Minas Drive on device 1.

$$DeviceCommand(D, "CM123", P)$$

When the command and more characters are followed by any number of numerical characters, these will appear in the command exactly as entered. This is most applicable for the Jog command.

For example, the following command *DeviceCommand(1, "5001", 0)*, would result in the Jog command with Jogging Low Speed in the (+) direction being sent to the Minas Drive on device 2. The parameter number *P* is ignored.

DeviceCommand(D, "CM", P)

When only the Command and Mode characters appear in the string, the simplest command is sent to the drive. This is most applicable for commands such as the Stop or Org command.

For example, the following command *DeviceCommand(5, "30", 0)*, would result in the Stop command being sent to the Minas Drive on device 2. The parameter number *P* is ignored.

Knowledge of Unit Operation Is Assumed

In all cases, the simple principle of 'pass-through' is maintained: there is no attempt to validate a value in terms of the end use of the unit: both familiarity with the control functions and knowledge of system operation are assumed.

Communications

Communications with the Minas Series AC Drive is via an RS-232, point to point link, with default serial communications format of baud rate of 9600, 8 data bits, No parity, and 1 stop bit.

The connections details are described in the table below.

Paradigm unit (RS 232 port)	Minas Series Drive (MNI DIN8)
Pin 1 (Tx)	Pin 5 (RXD)
Pin 2 (Rx)	Pin 3 (TXD)
Pin 3 (RTS)	
Pin 4 (CTS)	
Pin 5 (0v)	Pin 4 (0v)

In addition a link must be fitted between Pin 3 (RTS) and Pin 4 (CTS) on the Paradigm unit.