

---

# Animatics SmartMotor

## Application Note

---

This document describes some special requirements when programming the operator interface for communications with an Animatics SmartMotor. The communications protocol supports access to virtually all the commands recognized by the SmartMotor at the time the protocol was written. 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 select commands, user variables, and other values in the SmartMotor.

## Accessing Data

If one uses the Communications Blocks of Edict, there are no special considerations required. Just select the item in the Block Address List under Address. If one is using the Direct PLC Reference method of selecting items, there are some special considerations.

The Direct PLC Reference format is [ Device Number, Command ], for example [ 3, AMPS ] for the User Maximum Current on the SmartMotor which is assigned to line 3 in Communications Devices. For systemic reasons, the commands programmed outside of the Communications Blocks will make all the command letters uppercase. Therefore, one cannot read/write certain single letter lower case variables using this method, since the system will make them uppercase, and therefore a command. A particular example is the s variable. It will be made into S, which is STOP. While using the Communications Blocks is the preferred method, the simplest solution is to use ss, or sss, e.g. [7,ss].

## GOSUB Commands

There are additional selections available, involving the GOSUB command. By writing a number to the reference to the command GOSUB, one can cause that numbered subroutine in the SmartMotor to execute. For example, if in Comm Block Z, selected with Access = Write, GOSUB is assigned to Z[0], then an action Z[0] := 7 would send the command GOSUB7. The subroutine should not respond to this command.

There are also ten GOSUB commands, assigned from 900 to 909, that will, when called, expect a string of ASCII characters, terminated by a Carriage Return ( 13 Decimal, 0xD Hex ) in response. A user would assign a string variable, e.g. s901Response := GOSUB901. Subroutine 901 might then return something like "Motor is Hot" followed by the Carriage Return. A ? will be returned if an error code is returned by the SmartMotor.

## 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 parameters and knowledge of system operation are assumed.

## RS-232 Connection

Paradigm	SmartMotor
Tx - ( 1 )	Receive Data ( 4 )
Rx - ( 2 )	Transmit Data ( 3 )
0V - ( 5 )	RS-232 Ground ( 5 )
RTS ( 3 ) jumpered to CTS ( 4 )	