

Programming: The Anilink Port

Programming: The Anilink Port

The AniLink port (using I2C protocol)

AOUTB,c	Send variable "c" out to Analog I/O board addressed as "B"
DOUTB0,c	Send variable "c" out to Digital I/O board addressed as "B0"
c=AINB2	Set variable "c" to input 2 from Analog I/O board addressed as "B"
c=DINB0	Set variable "c" to input from Digital I/O board addressed as "B0"
PRINTB("Temp:",c,#32)	'Print to LCD on network - text, data and ASCII code

The AniLink port (using RS-485 protocol)

OCHN(RS4,1,N,38400,1,8,D)	'Open - No parity, 38.4k bps, 1 stop, 8 data, as data
IF LEN1>0	Check to see if any data is in the 16 byte input buffer
c=GETCHR1	Get byte from buffer into variable "c"
PRINT1("Char Rcd:",c,#13)	'Print text, data and ASCII code
ECHO1	Echo back all received characters
SILENT1	Suppress print messages
SLEEP1	Ignore all commands except "WAKE"
WAKE1	Consider all following commands

The AniLink port as general I/O:

UEI	'Set port E to input (UFI for port F)
UEO	'Set port E to output (UFO for port F)
UE=0	'Set port E Low (UF=0 for port F, or UF=c to set to variable "c")
UE=1	'Set port E High (UF=1 for port F)
c=UEI	'Set variable "c" to digital input (UFI for port F)
c=UEA	'Set "c" to analog input, 0 to 1024 = 0 to 5V (UFA for port F)

Ports "E" and "F" have internal 5k Ohm resistive pull-ups.

The SmartMotor is designed to be much more than a motion controller. With its vast programmability and expansive I/O it is equipped with the capability of running an entire machine. One of our challenges has been to connect all of the I/O within the small spaces available on the side or back of a motor. This led to the decision to make each I/O point multi-function, so if the integrator needed seven outputs, seven digital inputs or seven analog inputs, the requirement would be met, even with only seven points of I/O in total. In order to get these multiple functions out of each I/O point, however, it was necessary to make the I/O TTL, rather than 24V which is more typically used in machine control.

What we did to keep the flexibility and still satisfy the requirements of industrial machinery was to create a series of very small I/O adapters.

These adapters solder to the end of a 24V sensor, relay or valve cable and provide a connector to plug into the SmartMotor. They are equipped with the micro-circuitry to optically isolate as well as provide the switching power for common 24 volt devices.

To maximize the success of your SmartMotor installation you should know, not just about how the I/O works, but about the three areas

where the SmartMotor can additionally reduce costs within your system.

The first additional layer of savings can be realized if the programmability of the SmartMotor can free the integrator from having to purchase and install a PLC (Programmable Logic Controller).

The second additional layer of savings results from freeing the application from the need of a large NEMA cabinet that would otherwise house the servo drive components, PLC hardware and switching relays.

The third rests on the fact that an application developed with the SmartMotor automation system is burdened with a minimum of cabling and is entirely plugable. Not only is the system more reliable, but were there to be a failure with a customer in the field, finding the root cause of the failure, or simply upgrading a customer's hardware can quickly be accomplished by component swapping by non-technical personnel. This can save the cost and down time of sending technicians to the field, or having them on hand 24 hours a day.

Every port on the SmartMotor brings out 5 volt power and ground, as well as one or more I/O points. There are seven I/O points and each one has multiple functions. They are UA, UB, UC, UD, UE, UF and UG and have the following functions:

UA

- Digital Input, TTL, 0 to 5 volts
- Digital Output, TTL, 0 to 5 volts
- Analog Input, 10 bit, 0 to 1024
- External Encoder "A" Input
- Step & Direction, Step Input

UB

- Digital Input, TTL, 0 to 5 volts
- Digital Output, TTL, 0 to 5 volts
- Analog Input, 10 bit, 0 to 1024
- External Encoder "B" Input
- Step & Direction, Direction Input

UC

- Digital Input, TTL, 0 to 5 volts
- Digital Output, TTL, 0 to 5 volts
- Analog Input, 10 bit, 0 to 1024
- Positive Limit Input

UD

- Digital Input, TTL, 0 to 5 volts
- Digital Output, TTL, 0 to 5 volts
- Analog Input, 10 bit, 0 to 1024
- Negative Limit Input

UE

- Digital Input, TTL, 0 to 5 volts
- Digital Output, TTL, 0 to 5 volts
- Analog Input, 10 bit, 0 to 1024
- AniLink I2C Data I/O
- AniLink RS-485 Signal "A"*

UF

- Digital Input, TTL, 0 to 5 volts
- Digital Output, TTL, 0 to 5 volts

Analog Input, 10 bit, 0 to 1024
AniLink I2C Clock Output
AniLink RS-485 Signal "B"*

UG

Digital Input, TTL, 0 to 5 volts
Digital Output, TTL, 0 to 5 volts
Analog Input, 10 bit, 0 to 1024
Start Motion (or "GO") Input

*Not available on SMXXX5 SmartMotors

When used as general I/O, the following commands will select the port function.

UAI	Assign pin A to input state
UAO	Assign pin A to output state
UA=0	Set UA to zero volts
UA=1	Set UA to 5 volts
a=UAI	Read UA digital value, 0 or 1
a=UAA	Read UA analog voltage
UBI	Assign pin B to input state
UBO	Assign pin B to output state
UB=0	Set UB to zero volts
UB=1	Set UB to 5 volts
a=UBI	Read UB digital value, 0 or 1
a=UBA	Read UB analog voltage
UCI	Assign pin C to input state
UCO	Assign pin C to output state
UCP	Re-assign pin C to +limit action
UC=0	Set UC to zero volts
UC=1	Set UC to 5 volts
a=UCI	Read UC digital value, 0 or 1
a=UCA	Read UC analog voltage
UDI	Assign pin D to input state
UDO	Assign pin D to output state
UDM	Re-assign pin D to -limit action
UD=0	Set UD to zero volts
UD=1	Set UD to 5 volts
a=UDI	Read UD digital value, 0 or 1
a=UDA	Read UD analog voltage
UEI	Assign pin E to input state
UEO	Assign pin E to output state
UE=0	Set UE to zero volts
UE=1	Set UE to 5 volts
a=UEI	Read UE digital value, 0 or 1
a=UEA	Read UE analog voltage
UFI	Assign pin F to input state
UFO	Assign pin F to output state
UF=0	Set UF to zero volts
UF=1	Set UF to 5 volts
a=UFI	Read UF digital value, 0 or 1
a=UFA	Read UF analog voltage

UGI	Assign pin G to input state
UGO	Assign pin G to output state
UG	Re-assign pin G to "GO"
UG=0	Set UB to zero volts
UG=1	Set UB to 5 volts
a=UGI	Read UB digital value, 0 or 1
a=UGA	Read UB analog voltage

The UAA, UBA, UCA, UDA, UEA, UFA and UGA variables reflect the analog voltages at the port pins regardless of how the pins are configured. Any time the variable is used it is loaded with the current value. You can read a pin without effecting it this way. That can be helpful in debugging. For example you could be using a pin as an output and then read the analog input value to see if it were shorted.

The encoder and step counting capabilities of ports A and B are described in the section on [External Encoder Modes](#). The serial data capabilities of ports E and F are described in the section on communications.