SmartMotor™ Interfacing

SmartMotor™ Interfacing

The following pages cover various schematics to help interface electrically to SmartMotors™

SmartMotor™ Connections:

Note: All Connections on the 7W2 combo connectors are available on the DB-15 connector as well. They are a direct internal connection on all standard D-Sub connector motors.

If 1 or less I/O Points and RS-232 Port Connection is all that is needed, then the 7W2 connector is the only connection needed to operate the motor.

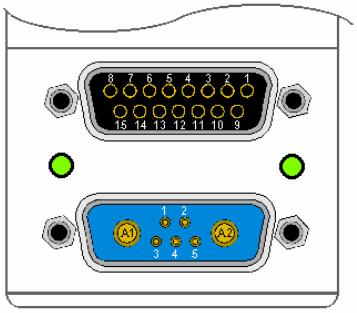
Any additional I/O features are found on the DB-15 connector.

Please refer to other documentation on programmable operation of I/O.

This is a Standard D-sub connector SmartMotor™

15 Pin D-Sub I/O:

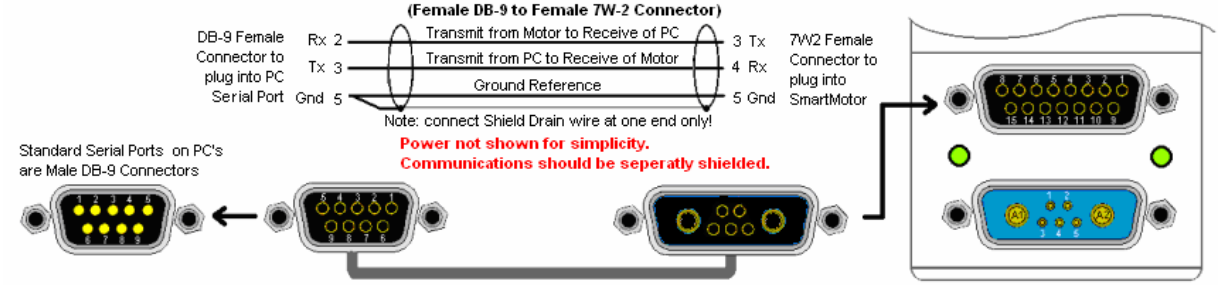
1 I/O A
2 I/O B
3 I/O C
4 I/O D
5 I/O E
6 I/O F
7 I/O G
8 Encoder A Out
9 Encoder B Out
10 RS-232 Transmit
11 RS-232 Receive
12 +5V Out
13 Ground
14 Pwr GND
15 Power



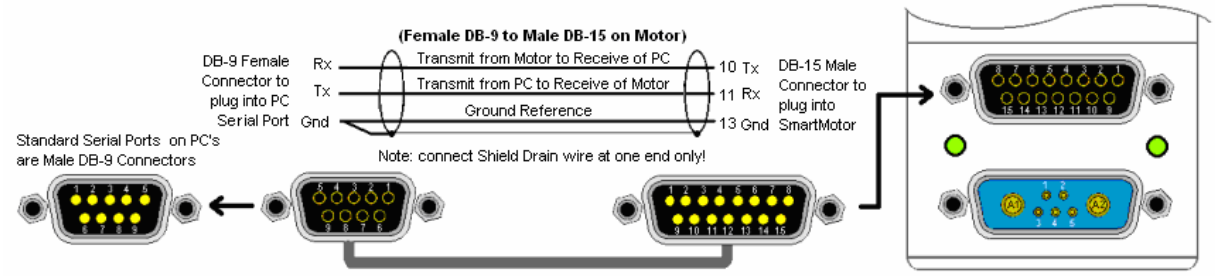
7 Pin Combo D-Sub Power and I/O:

1 Sync or I/O G
2 +5V Out
3 RS-232 Transmit
4 RS-232 Receive
5 RS-232 Ground
A1 Power
A2 PWR GND

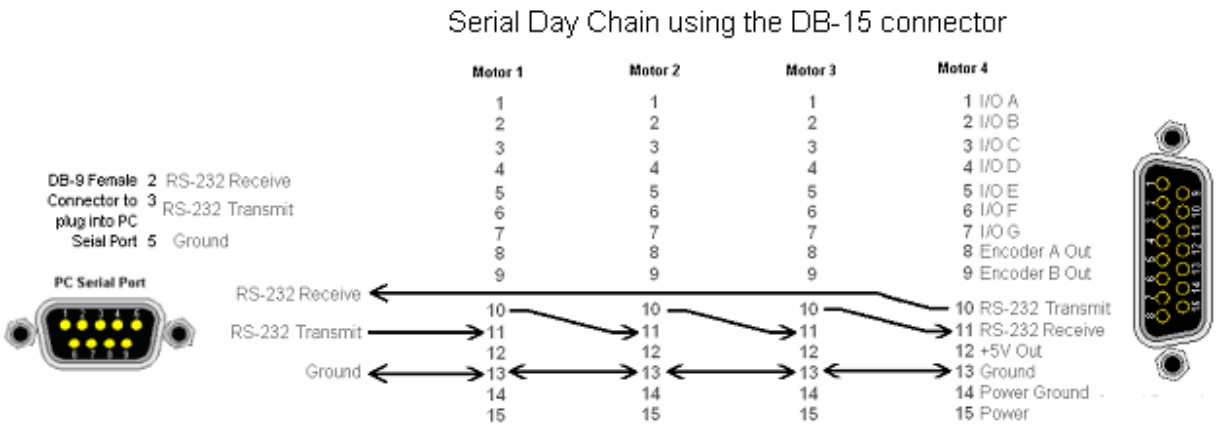
RS-232 Programming cable schematic to communicate with one motor via the main 7W2 Connector:



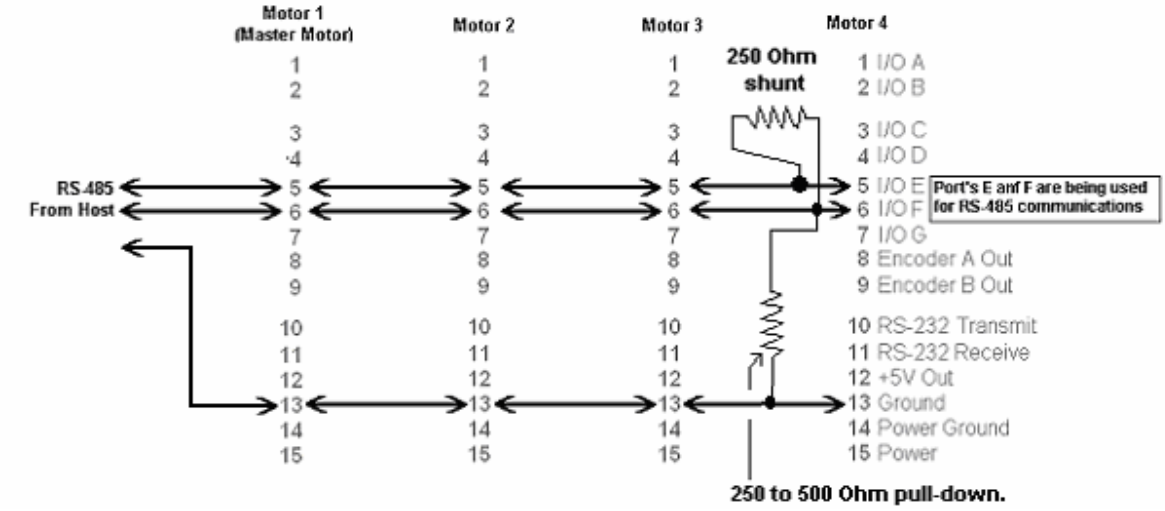
RS-232 Programming cable schematic to communicate with one motor via the DB-15 Connector:



RS-232 Serial Daisy-Chain cable to communicate to multiple motors via the DB-15 Connector:



RS-485 Parallel Daisy-Chain to communicate to multiple motors via the DB-15 Connector:



Note: All RS-485 networks require >= 400mVolt differential bias to work properly. The SmartMotor employs 5Kohm pull-ups on all I/O pins, as a result, the pull down resistor shown is needed for proper operation. The shunt resistor may be required of the distance to the last motor is significantly long.

Port E is referred to as the “A” or “Posative” side of the RS-485 bus while Port F is referred to as the “B” or “Negative” side of the bus.

Ideal cable would be dual twisted pair with shield. The shield should be tied to ground at one point only. The ideal point would be the host, or if no host, the first motor. The shield should not be used as a ground reference or be tied to any more than one ground point. This would cause noise to be induced into the bus.

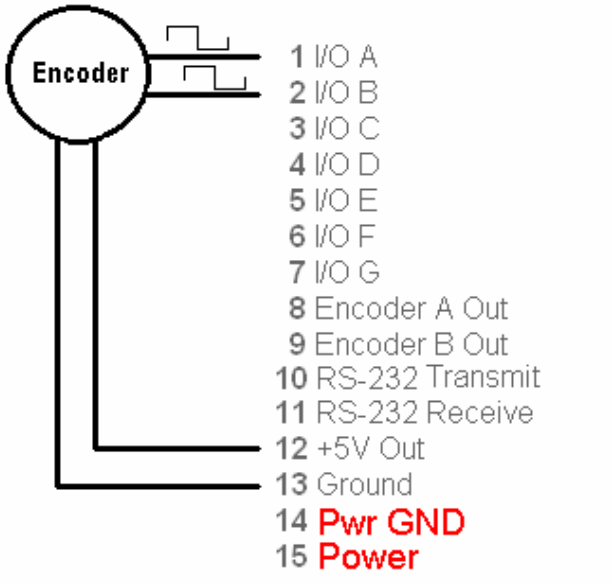
If RS485ISO adapters are used, they make use of the Main RS-232 port. As a result, the bus needs to be powered from a separate 5VDC source. If a motor is used as that source, opto-isolation would be defeated.
The best means to power it would be from the host.

ANIMATICS®

Connecting an external encoder for External closed-loop operation or for electronic gearing:

Note: The schematic shown is for an Encoder that can be powered from the internal 5VDC supply of the motor. The motor can only supply a maximum of 150mAmps. Ensure the chosen encoder does not draw too much current.

If the external encoder has differential outputs, such as A(+), A(-) and B(+), B(-), then just wire the plus connections to Ports A and B inputs respectively. Maximum input frequency is 2MHerts.



Example Code to initiate Encoder Following:

```
MF4          ' Interpolate incoming pulses in full quadrature
MFMUL=4      ' Multiply incoming counts by 4
MFDIV=7      ' Divide incoming counts by 7
MFR          'Calculate Mode-Follow-Ratio
G            ' Begin following at that ratio
```

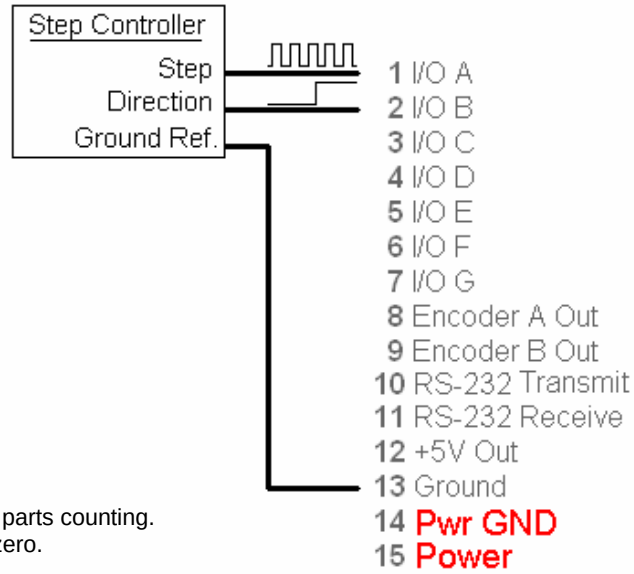
Side Note:
Ports A and B can also be used as a high speed input counter. Issue the command "MF0", and the counter will be set to zero. The command "RCTR" will report counter value. The value will be total full quadrature counts received since MF0 was issued. This method can be used to trigger events in one motor based off of positions from another motor.

Example:

```
MF0                          ' Set counter to zero
WHILE CTR<20000 LOOP         ' Loop until count exceeds 20000
V=100000
A=100
MV
G                            ' Start moving in velocity mode.
```

Connection to a PLC or stepper card output for running in Step Mode:

Note: Schematic shown is for sinking-output stepper controllers. Each I/O Port has a 5Kohm Pull-up resistor that the step controller would need to pull down.
Maximum step input frequency is 2MHerts.



Example Code to initiate Step/Direction Following:

```
MS           ' Set motor to Mode-Step
MFMUL=4      ' Multiply incoming counts by 4
MFDIV=7      ' Divide incoming counts by 7
MSR          ' Calculate Mode-Step-Ratio
G            ' Begin following at that ratio
```

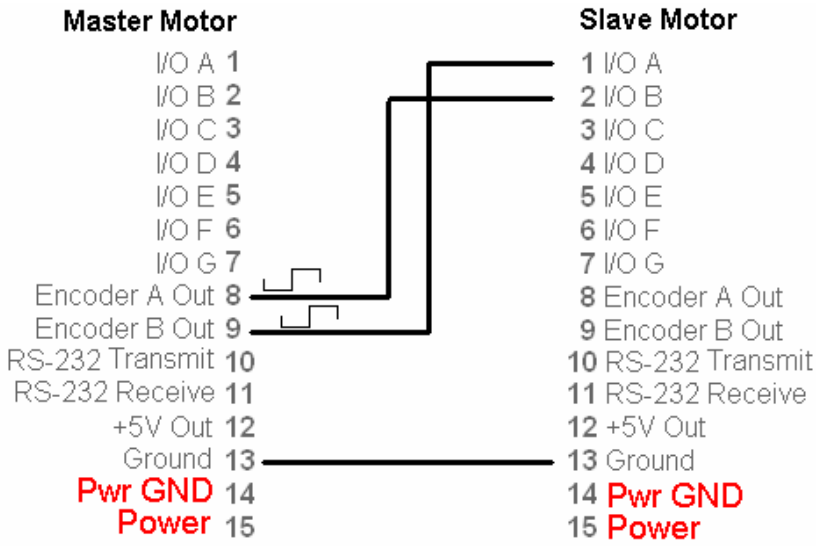
Side Note:
Port A can also be used as a high speed input counter for parts counting.
Issue the command "MS0", and the counter will be set to zero.
The command "RCTR" will report counter value.
The value will be total step pulses received since MS0 was issued.
Port B will control whether it counts up or down from zero.

SMART motor

Connecting 2 motors for Electronic Gearing:

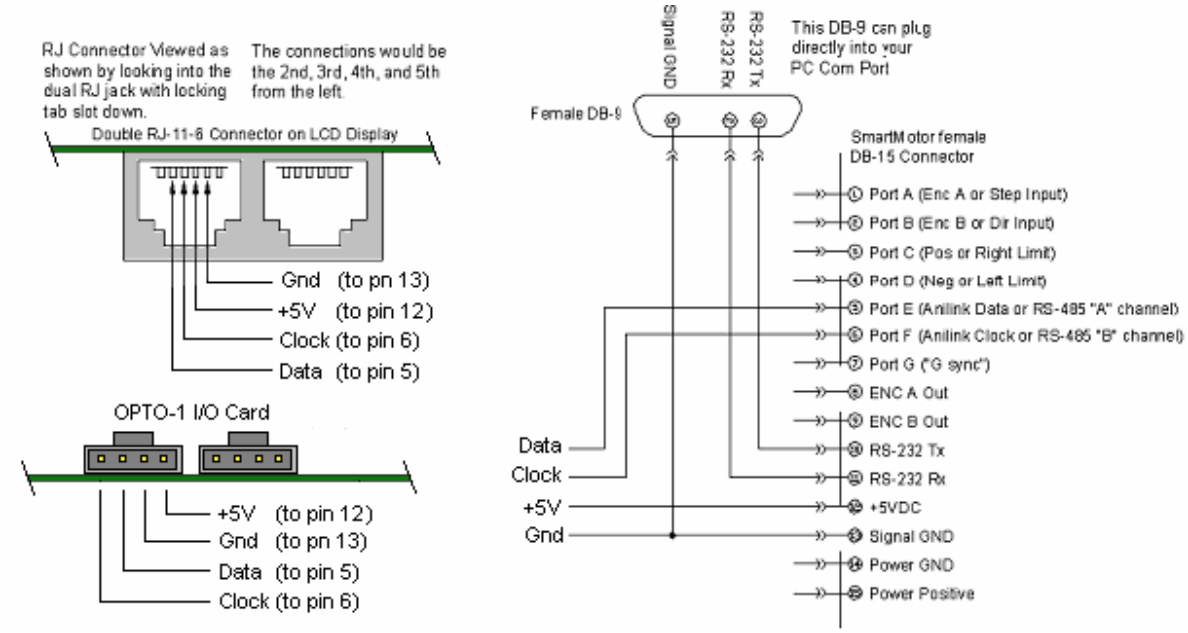
Note:
Pin's 8 and 9 are the inverted outputs of the motor's internal encoder. This is why A-out is connected to B in and vice versa.
Otherwise the slave motor would spin the opposite direction.
Software code still allows for reversal if hardware change is not desired.

Ground reference is needed for proper operation.



See previous page for example program code.

Connection to Anilink Devices (Both LCD RJ Connection and OPTO-1 Molex connection shown)



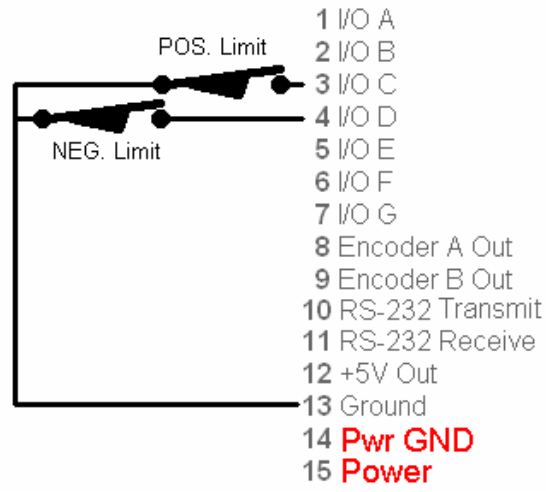
Note: Maximum distance for Anilink devices is 4 feet.
RS-232 communications is shown for clarity.
RS-485 communications is not available when Anilink Devices are used.

Typical Limit Switch Inputs:

As Shown, they are Active-High Asserted. This means when the limit switches open or the connection breaks, the motor will stop.
This is because each input on the motor has a 5Kohm pull-up to 5VDC.

In Versions of firmware **prior to** 4.76, this requires the LIMH command to make them active-high. Version 4.76 and later default to active high.

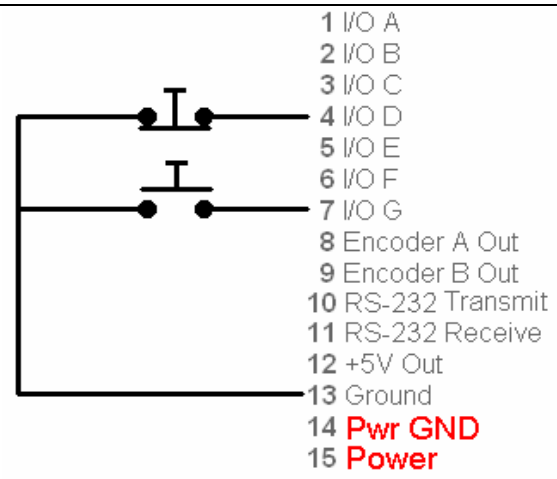
Only 5VDC sensors or dry-contact switches can be used. If solid state sensors are used, they should be NPN or "Sinking" type outputs to pull down the 5Kohm pull-ups that are in the motor.



Simple Start/Stop set-up using the "G-sync" function of Port G for start/go and Limit switch input to Stop.

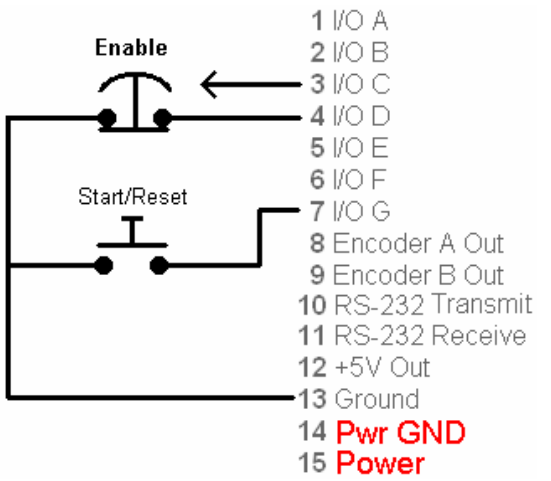
Note: By default, When Port G is grounded, the processor interprets it as a "G" command being issued.

Port D limit input was used for this example.
Either Port C or port D could have been used to stop motion as long as the respective limit input is enabled and active-high.



Similar to above, with Limit used as an E-Stop Enable.

Note: Port C pointing out. In Versions 4.15b or later firmware the command BRKC can be used to control an external brake.
As a result, it can also be used as a fault-output to another device.

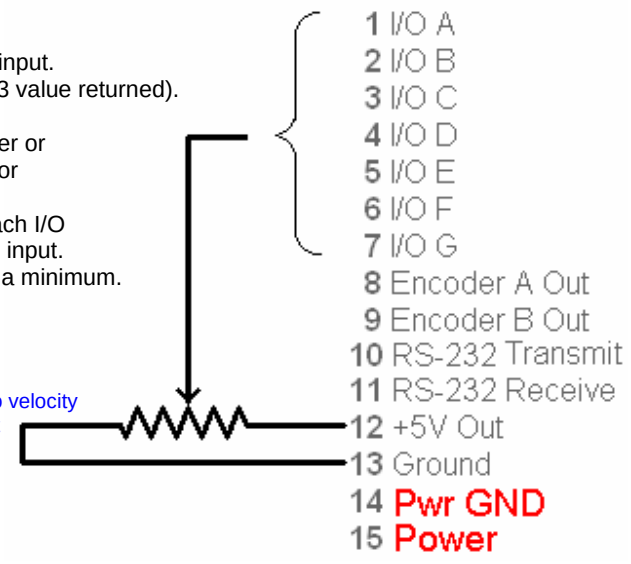


Analog Input to a SmartMotor:

Any of the 7 I/O Ports can be read as a 10 Bit analog input.
The voltage range must be from zero to 5VDC (0-1023 value returned).
Example: a=UEA would assign the analog value of Port E to the variable "a". If a standard Potentiometer or linear adjustable resistor is used, it should be 1Kohm or less in value to give the best response.
This is because the motor's internal 5K Pull-ups on each I/O port pin must be "pulled down" via the external analog input.
It is best to use shielded cable to keep noise levels to a minimum.

Example code:

```
MV                          'Set to Velocity Mode
A=100                       'Set acceleration
WHILE 1             'While forever
        V=UAA*1000          'Assign Port-A analog value to velocity
        G                   'Make new velocity take effect
LOOP
```



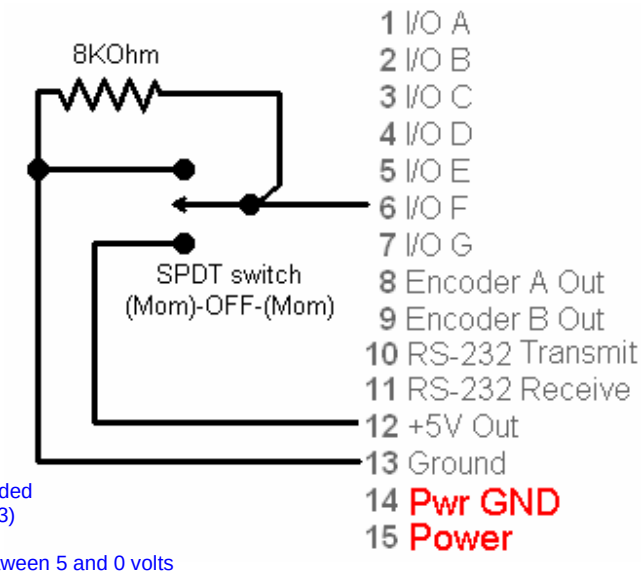
Note:

I/O Ports can be read as analog inputs even while being used as Digital Inputs or Outputs as seen in this next example:

Obtaining 2 functions out of one Input:

In this example, a spring-to-center toggle switch and an 8KOhm pull-down resistor is connected to Port F. Combined with the 5K-internal Pull-up, the port will normally read a logic high when read as a digital input. When read as an analog input, it will read about 600 (0-1023 for 0-5VDC).
If the switch is swung to ground, it will read a digital zero. When swung to 5VDC, it will read ~1023 on the analog scale. This means a single Input pin could be used as a Jog Up/Down switch.

An added benefit is that if the connector comes off the motor, you will know it because the input will always read high and it's analog value will be ~1023.



Example Code:

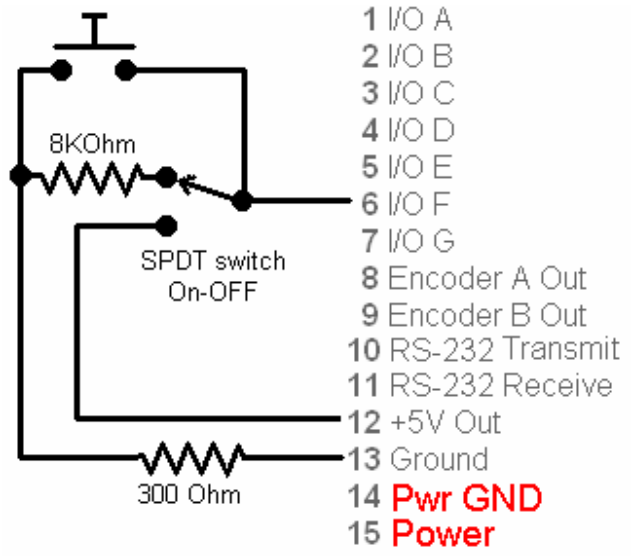
```
        WHILE 1
                IF UFI==0               'if Port F is hard grounded
                        PRINT("Pushbutton pressed",#13)
                        WHILE UFI==0 LOOP
                ELSEIF UFA<600    ' If Port F is biased between 5 and 0 volts
                        PRINT("Switch in upper position",#13)
                        WHILE UFA<600 LOOP
                ELSE                    ' If Port F is hard pulled to 5VDC
                        PRINT("Switch in lower position",#13)
                        WHILE UFA>700 LOOP
                ENDIF
        LOOP
```

Note: The above toggle switch could have been 2 separate momentary pushbuttons as well., But if someone were to press both at once, the 5VDC supply would be shorted out. To avoid this, an extra resistor could be employed on the ground line. (See next Example)

Push-Button and Toggle Switch into single input:

This is a twist to the last example. This time a simple on-of switch is being used with the same 8K Pull-up. Normally the Port will read about 600 as an Analog value. If the Momentary pushbutton is pressed, it will read zero and logic level zero. If the switch is opened, it will read 1023 on the Analog scale.
If the Toggle switch is spring return to the shown position, then it is possible to detect if the connection came loose.

The 300 Ohm resistor is to prevent a 5VDC supply short in case the pushbutton and toggle were pressed at the same time. (See Note on previous example)

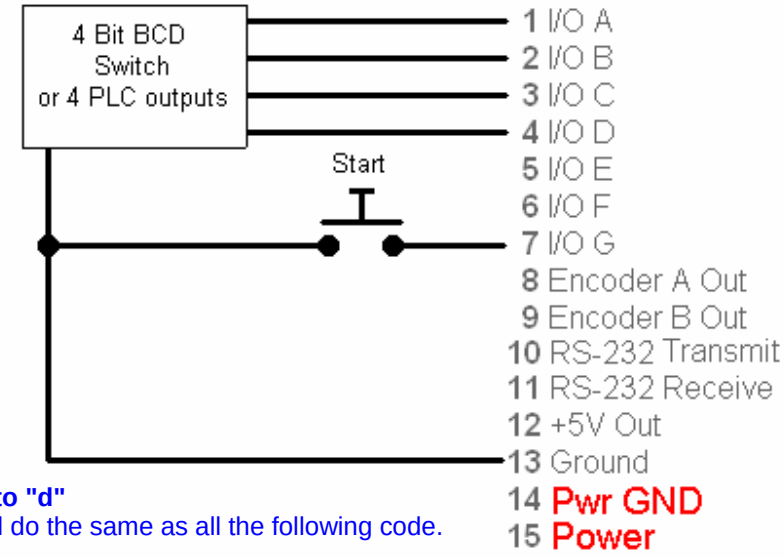


Example Code:

```
WHILE 1
IF UFI==0              'if Port F is hard grounded
        PRINT("Pushbutton pressed",#13)
        WHILE UFI==0 LOOP
ELSEIF UFA<600   ' If Port F is biased between 5 and 0 volts
        PRINT("Switch in upper position",#13)
        WHILE UFA<600 LOOP
ELSE                   ' If Port F is hard pulled to 5VDC
        PRINT("Switch in lower position",#13)
        WHILE UFA>700 LOOP
ENDIF
LOOP
```

Binary (4 Bit BCD) input control:

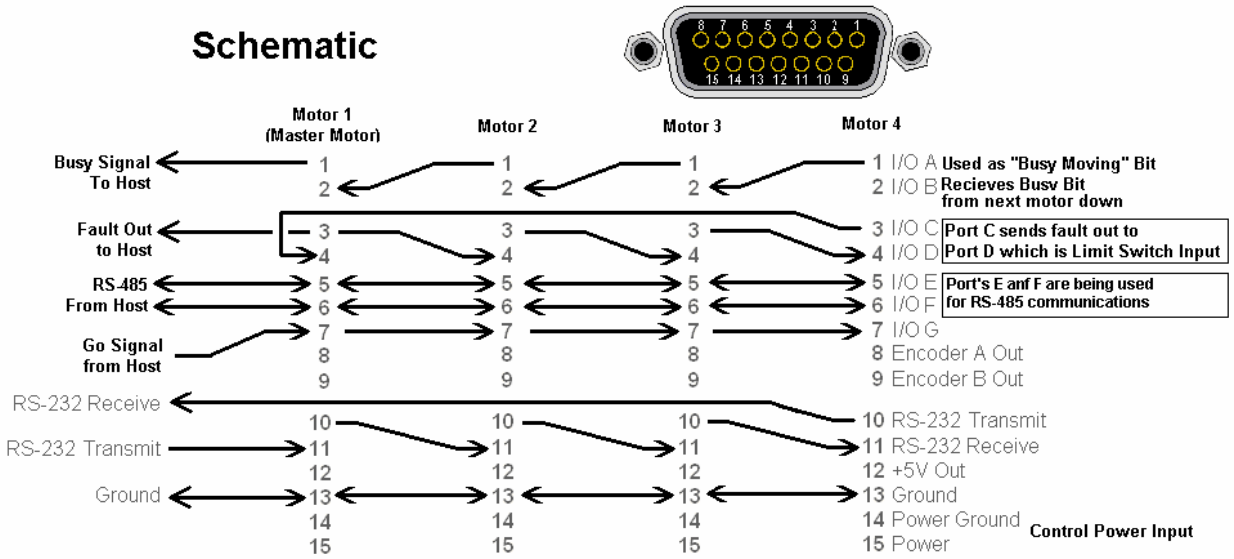
By using 4 inputs and a binary switch or 4 PLC outputs, up to 16 functions can be achieved.
This could be 16 subroutine calls, 16 pre-set speeds or positions or any of the above combinations. Via programming capabilities, It could be a sequencing operation even controlled from another SmartMotor.
A common use for the other 2 I/O pins is to use them as Busy and Fault outputs back to a PLC.



Example Code:

```
C4      'check binary switch, assign it to "d"
' Note, For Versions >=4.76, d=U&511 will do the same as all the following code.
b=UBI*2
a=UAI+b
c=UCI*4
d=UDI*8
d=c+d
d=d+a                  'd now contains the 4 bit value of the inputs
RETURN
```

The following is a more complex arrangement of I/O handling that includes hard wired Fault protection, RS-232 and RS-485 communications.



Notice how I/O is cascaded through from one motor to the next.
By using Port G's function as a G-sync line, we can trigger moves on all motors at one time.

If a PLC or Host controller does the triggering, it will need conformation back that the motors are busy. Port A and B are being used for this. Each motor sends a busy signal out of port A and into Port B of the next motor up the chain. When all motors are completed, the first motor will signal the PLC.

Example:

```
'Slave motor code for sending Busy signal up the chain:
WHILE 1             ' standing by to detect a move
        IF Bt                   ' If port G was grounded, the Bt bit will go from 0 to 1 indicating Busy Trajectory
                UAI=1           ' Set Port A to 1
                TWAIT           ' Wait for move to complete
                WHILE UBI LOOP  ' Wait for Port B to go to 0, as fed from next motor down.  (last motor should omit this code)
                UAI=0           ' Set Port A to zero to trigger next motor up.
        ENDIF
LOOP
```

By utilizing the BRKC function of V4.15 and later firmware, Port C becomes a fault output. By feeding this into Port D with Port D being defaulted as a limit switch, if any one motor faults, it will trigger a chain reaction and stop all motors immediately.

The RS-232 chain can be used for PLC, HMI, or PC connection while the RS-485 communications bus can be used between each motor to allow isolated control from the host.

DE (Drive Enable) Option

All SM23XX and SM34XX motors come with a "-DE" option.
This option separates Pin 15 of the DB-15 connector from Pin A1 of the 7W2 connector allowing separate power supplies to run the controller and Drive amplifier sections of the motor.

The connection between A2 (Power Ground) and pin 14 are maintained though. This means that if separate power supplies are used, they cannot have their grounds tied together outside of the motor. To do so would cause a serious ground loop with drive currents being placed on the controller ground.

The reason for the –DE option is 3-fold.:

1. It allows the controller to b "kept alive" under an E-Stop condition so re-homing is not necessary.
2. The controller is protected from current surges caused by the drive amplifier (or other motors on the same supply)
3. Better protection against Back-EMF voltage spikes. (they will not reach the controller).
4. The drive Amplifier can take much higher spikes than the controller.