CAN 2.0B Communication Protocol

Version 01 Pre-Release
Published April 18, 2012

1.0 Overview

The CAN2.0B beta systems comprise individual connectors for a CAN2.0B link for 1Mbps serial communication, and an RS-232 serial link for programming.

Upon power up, all Sunstream servos will transmit two sequential 8 bit words, #1Bh & #57h, on the RS-232 link – this series is transmitted twice. If a handheld teaching pendant is connected to the RS-232 port, or if a PC is connected to the RS-232 port, with the Sunstream programming software running, the software will enter the programming mode (see Beta OI Terminal Manual). If no connection is present, the software will proceed with initialization.

This manual describes how a Host can communicate with the servo controller, via CAN2.0B link, to:

- Request information from the Sunstream controller.
- Enable and disable operation of the actuator.
- Change parameter settings, such as maximum acceleration and velocity.
- Command the servo actuator to perform a specific task, such as Move to a Position.

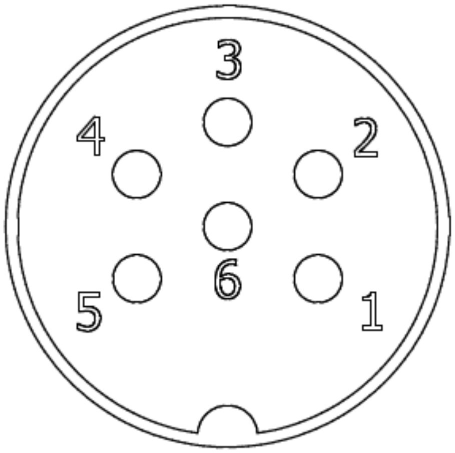
2.0 Hardware

- 6 pin Binder Series 680 connector: CANH, CANL, N/C, N/C. This GND pin is an isolated ground reference shared both RS-232 and CAN2.0B circuits. www.binder-usa.com
- The CAN2.0B transceiver is a Texas Instruments SN65HVD251, and it meets the requirements of ISO 11896.
- All communication pins have 4kV peak galvanic isolation provided by Texas Instruments series ISO7220 digital isolators. These digital isolators operate off a +5V supply (referenced to the isolated GND) and have TTL input thresholds. Outputs are 4mA CMOS.
- Pull pin 3 to GND to disconnect the 120Ω termination resistor. Leave open, or pull to +5V to connect the termination resistor.

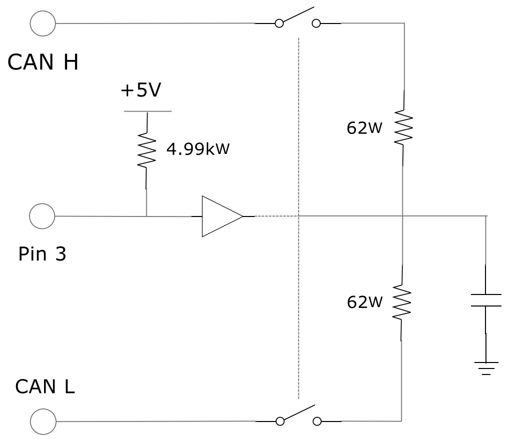
2.1 Connector Pinout

View shown looking into male connector on servo.

Pin 1:	CAN L
Pin 2:	NC
Pin 3:	Termination Disable
Pin 4:	NC
Pin 5:	CAN H
Pin 6:	GND



2.2 Termination Schematic



Pull **Pin 3** to +5V, or leave open, to connect RC filter network between CAN H and CAN L. Pull Pin 3 to GND to open switches and disconnect RC filter network.

3.0 Communication Settings

The Sunstream servo controller is not programmable through the 12 bit I/O parallel communication link. To learn more about programming the Sunstream servo controller, refer to the GUI Manual.

4.0 Data Formatting

4.1 Data Frame description transmitted by Host (received by Sunstream controller)

MBOXnA

- The 16bit word in MBOXnA is divided into four 4bit segments allocated to requesting specific information from the Sunstream Servo. Each information request has a 4bit code – and these codes are specified in Sect. 1.5. If an improper request is made for a specific mailbox, that mailbox will be loaded with AAAAh.
- Bits 15-12: A request code placed in this section corresponds to MBOXnA of the message response from the Sunstream controller. For example (refer to Sect. 1.5) if bits 15-12 are set to 03h, the Sunstream controller will decode these bits as a request for the cylinder position, and store the 16bit cylinder position value to MBOXnA.
- Bits 11-8: A request code placed in this section corresponds to MBOXnB of the message response from the Sunstream controller.
- Bits 7-4: A request code placed in this section corresponds to MBOXnC of the message response from the Sunstream controller.
- Bits 3-0: A request code placed in this section corresponds to MBOXnD of the message response from the Sunstream controller.

MBOXnB

- This 16bit word is reserved for an Instruction Code. See Sect. 1.6 for further details.

MBOXnC

- This word is reserved for a quantitative value associated with an Instruction Code. See Sect. 1.6 for further details.

MBOXnD

- This word may be used for a value associated with an Instruction Code.
- Some Instruction Codes are associated with a 32bit quantitative value. In this circumstance, the upper 16bits need to be stored to MBOXnC, and the lower 16bits stored to MBOXnD.

4.2 Data Frame description received by Host (transmitted by Sunstream controller)

MBOXnA:

- Contains data requested by bits 15-12 of transmitted MBOXnA.

MBOXnB

- Contains data requested by bits 11-8 of transmitted MBOXnA.

MBOXnC

- Contains data requested by bits 7-4 of transmitted MBOXnA.

MBOXnD

- Contains data requested by bits 3-0 of transmitted MBOXnA.

5.0 Protocol

5.1 Information Requests

Status Request

- **ReqCode:** 01h
- **Reply Length:** 16 bits
- **Reply:**
  - If the servo is busy executing an instruction, it will transmit the word 8xxxh.
  - If the servo is ready to accept a new instruction, it will transmit the word 9xxxh.
  - The final four bits of status is dedicated to the servo mode: xxxFh indicates that the position and force servo loops are disengaged, and spool position is directly controlled; xxx0h indicates that the position servo loop is engaged; xxx1h indicates that the force servo loop is engaged. See Sect. 1.7

Acknowledge Request

- **ReqCode:** 02h
- **Reply Length:** 16 bits
- **Reply:**
  - Sunstream controller reflects the previous Instruction code stored into MBOXnB.

Cylinder Position Request

- **ReqCode:** 03h
- **Reply Length:** 16 bits
- **Reply:**
  - Cylinder position[in.] = (response)*stroke[in.] / 32768
  - Stroke[in.] is the maximum stroke of the actuator.

Cylinder Command Request

- **ReqCode:** 04h
- **Reply Length:** 16 bits
- **Reply:**
  - Cylinder command[in.] = (response)*stroke[in.] / 32768
  - Stroke[in.] is the maximum stroke of the actuator.

Head Pressure Request

- **ReqCode:** 05h
- **Reply Length:** 16 bits
- **Reply:**
  - Head Chamber Pressure[psi] = (response)*/ 327.68

Rod Pressure Request

- **ReqCode:** 06h
- **Reply Length:** 16 bits
- **Reply:**
  - Rod Chamber Pressure[psi] = (response)*/ 327.68

Spool Position Request

- **ReqCode:** 07h
- **Reply Length:** 16 bits
- **Reply:**
  - Spool position[in.] = (response)*(sgn) / 327680
  - Leading bit is sign bit

Force Request

- **ReqCode:** 08h
- **Reply Length:** 16 bits
- **Reply:**
  - Force [lbf.] = 8*(response)*(sgn)/327.68
  - Leading bit is sign bit

5.2 Instruction Set

An Instruction commands the servo to carry out a specific physical task, such as moving to a specific point; an informational task, such as modifying the maximum permissible acceleration; or a logical task, such as initiating operation after a DSP reset. Each Instruction carries a unique code, named the Instruction Code.

5.2.1 Logic Instructions

Halt

- **Instruction Code:** 0001h
- **Value Length:** N/A
- **Polarity:** N/A
- **Description:**
    - When the Sunstream actuator receives a Halt instruction, it will immediately cancel execution of any operation currently in execution, disable the servo position loop, and set the spool position to null.

Operate

- **Instruction Code:** 0002h
- **Value Length:** N/A
- **Polarity:** N/A
- **Description:**
    - Enables the servo position loop after initial boot up and maintains current position. Also, unlocks the Sunstream actuator from Halt mode and re-engages the servo position loop. This instruction is also required to re-engage the servo loop if the servo faults out from a loss of input pressure condition.

Reset

- **Instruction Code:** 0003h
- **Value Length:** N/A
- **Polarity:** N/A
- **Description:**
    - Valve drivers are shut down and the processor is reset through software.

Override

- **Instruction Code:** 0004h
- **Value Length:** N/A
- **Polarity:** N/A
- **Description:**
    - Allows an external override of the program sequence. If an instruction is presently in execution, such as a move to a new Position Point, the servo controller will ignore any new commands until the present command has completed. By transmitting an Override signal to the servo controller, this feature is temporarily disabled.

Initiate

- **Instruction Code:** 000Fh
- **Value Length:** N/A
- **Polarity:** N/A
- **Description:**
    - Begins to sequence to initialize all registers and sensors. When the servo controller receives this instruction, it will reply by transmitting a 1000h in the first mailbox. After initialization is complete, the servo controller will transmit a 2000h.

5.2.2 Dynamic Instructions

Move to a Point

- **Instruction Code:** 0010h
- **Value Length (C):** 7 bits
- **Polarity (C):** None
- **Scaling (C):** N/A
- **Value Length (D):** 16 bits
- **Polarity (D):** None. Set leading bit to 0.
- **Scaling (D):** Value = 512*velocity[in/s] / (max. stroke[in.])
- **Description:**
  - This commands the servo to move to one of 128 points in data memory. Each point can be programmed to a position with 0.001" resolution (see OI Terminal Manual). After receiving this instruction, the servo will move from its present location to the position represented by the data point with acceleration, deceleration, and velocity limits imposed through programmed values. The velocity limit can be modified by storing any non-zero velocity value to MBOXnD.
- **Notes:**
  - Store Instruction Code to MBOXnB, right hand justified
  - Store Position value to MBOXnC, right hand justified
    - Value 0000h represents point 000
    - Value 007Fh represents point 127
  - Store velocity value to MBOXnD, right hand justified
    - A value of 0000h does not modify velocity setting

Move to a Position

- **Instruction Code:** 0020h
- **Value Length (C):** 16 bits
- **Polarity (C):** None. Set leading bit to 0.
- **Scaling (C):** Value = 32767*position[in.] / (max. stroke + 0.25 [in.])
- **Value Length (D):** 16 bits
- **Polarity (D):** None. Set leading bit to 0.
- **Scaling (D):** Value = 512*velocity[in/s] / (max. stroke[in.])
- **Description:**
  - This instruction commands the servo to move to a specific 15bit position. After receiving this instruction, the servo will move from its present location to the position instructed with acceleration, deceleration, and velocity limits imposed through programmed values. The velocity limit can be modified by storing any non-zero velocity value to MBOXnD.
- **Notes:**
  - Store Instruction Code to MBOXnB, right hand justified
  - Store Position value to MBOXnC, right hand justified
    - Value 0000h represents maximum retraction
    - Value 7FFFh represents maximum extension
  - Store velocity value to MBOXnD, right hand justified
    - A value of 0000h does not modify velocity setting

Ramp to a Force

- **Instruction Code:** 0030h
- **Value Length:** 32 bits
- **Polarity:** Leading bit (Bit 15) of MBOXnC is sign bit.
- **Scaling:** Value = (sgn)*(327.67)*force[lbf]
- **Description**

- This instruction commands the servo to ramp to a specific force, regardless of position. The maximum force of cylinder is assumed to be (100psi)*(Bore Area) lbf – where a positive force extends the cylinder, and a negative force retracts the cylinder. The servo will ramp from present force to commanded force at a rate defined by the instruction "Set Force Rate" (see Sect. 1.6.3).

- **Notes:**
  - FFFC0000h represents Force of -800 lbf.
  - 00040000h represents Force of +800 lbf.
  - Store Instruction Code to MBOXnB, right hand justified
  - Store upper 16 bits to MBOXnC, right hand justified
  - Store lower 16 bits to MBOXnD

Soft Touch

- **Instruction Code:** 0040h
- **Value Length (C):** 16 bits
- **Polarity (C):** None. Set leading bit to 0
- **Scaling (C):** Value = 32767*position[in.] / (max. stroke + 0.25 [in.])
- **Value Length (D):** 16 bits
- **Polarity (D):** Leading bit (Bit 15) of MBOXnD is sign bit.
- **Scaling (D):** Value = (sgn)*(1/8)*(327.67)*force[lbf]
- **Description:**
  - This instruction commands the servo to move to a 15bit position as specified in the upper byte of MBOXnC, until a force limit is exceeded, as specified by MBOXnD. As with the "Move to Position" instruction, the servo will move from its present location to the instructed position with acceleration, deceleration, and velocity limits imposed through programmed values. If the force limit is exceeded before the assigned position is achieved, the servo will exit the position loop and transition into a force loop. Force limits can be defined with positive (extend) or negative (retract) values, and have a 12bit resolution. ***Errata: A "Move to Point", "Move to Position", or "Step to Position" instruction must precede a "Soft Touch" instruction. Software has not yet been finalized to allow the servo to autonomously transition from force to position mode with this instruction.***
- **Notes:**
  - Store Instruction Code for position to MBOXnB, right hand justified.
  - Store Position value to MBOXnC, right hand justified
    - Value 0000h represents maximum retraction
    - Value 7FFFh represents maximum extension
  - Store Force Offset value to MBOXnD, right hand justified
    - Value 7FFFh represents +800lbf
    - Value 8000h represents -800lbf

Step to a Position and Set Force Offset

- **Instruction Code:** 0050h
- **Value Length (C):** 16 bits
- **Polarity (C):** None
- **Scaling (C):** Value = 32767*position[in.] / (max. stroke + 0.25 [in.])
- **Value Length (D):** 16 bits
- **Polarity (D):** Leading bit (Bit 15) of MBOXnD is sign bit
- **Scaling (D):** Value = (sgn)*(1/8)*(327.67)*force[lbf]
- **Description:**
  - This instruction simultaneously updates the cylinder position command and the force offset. It can be useful in systems where the motion of the actuator is subject to a known

disturbance force that varies with time – such as compression of a spring. With this command, the Force Offset is limited to +/-800 lbs, and has 12bit resolution.

- **Notes:**
    - Refer to Sect. 1.6.3 “Set Force Offset”
    - Store Instruction Code to MBOXnB, right hand justified
    - Store Position value to MBOXnC, right hand justified
        - Value 0000h represents maximum retraction
        - Value 7FFFh represents maximum extension
    - Store Force Offset value to MBOXnD, right hand justified
        - Value 7FFFh represents +800lbf
        - Value 8000h represents -800lbf

Position Spool

- **Instruction Code:** 0060h
- **Value Length:** 16 bits
- **Polarity:** Leading bit (Bit 15) of MBOXnC is a sign bit.
- **Scaling:** Value = (sgn)*327670*position[in.]
- **Description:**
    - This instruction immediately updates the position command of the valve spool. By employing this instruction, all other servo loops are disengaged. A positive command to the spool will act to extend the actuator. A negative command to the spool will act to retract the actuator.
- **Notes:**
    - Value 8000h represents maximum retraction
    - Value 7FFFh represents maximum extension
    - Store Instruction Code to MBOXnB, right hand justified
    - Store value to MBOXnC, right hand justified

Step to a Position

- **Instruction Code:** 0070h
- **Value Length:** 16 bits
- **Polarity:** None. Set leading bit to 0.
- **Scaling:** Value = 32767*position[in.] / (max. stroke + 0.25 [in.])
- **Description:**
    - This instruction immediately updates the cylinder position command internal to the Sunstream processor. No acceleration, deceleration, and velocity limits are imposed. This instruction is chosen when the operation requires the actuator follow a custom position trajectory, or step to a position. CAUTION: Large step displacements with this command may cause oscillatory behavior when the system inertia is high.
- **Notes:**
    - Value 0000h represents maximum retraction
    - Value 7FFFh represents maximum extension
    - Store Instruction Code to MBOXnB, right hand justified
    - Store value to MBOXnC, right hand justified

Step to a Force

- **Instruction Code:** 0080h
- **Value Length:** 32 bits
- **Polarity:** Leading bit (Bit 15) of MBOXnC is sign bit.
- **Scaling:** Value = (sgn)*(327.67)*force[lbf]
- **Description**

  - This instruction commands the servo to output a specific force, regardless of position. The maximum force of cylinder is assumed to be (100psi)*(Bore Area) lbf – where a positive force extends the cylinder, and a negative force retracts the cylinder. No rate limit is applied (unlike as in Ramp to Force) when this instruction is employed.
- **Notes:**
  - FFFC0000h represents Force of -800 lbf.
  - 00040000h represents Force of +800 lbf.
  - Store Instruction Code to MBOXnB, right hand justified
  - Store upper 16 bits to MBOXnC, right hand justified
  - Store lower 16 bits to MBOXnD

Step to a Load

- **Instruction Code:**	0090h
- **Value Length (C):**	16 bits
- **Polarity (C):**	Leading bit (Bit 15) of MBOXnC is sign bit.
- **Scaling (C):**	Value = (sgn)*(327.67)*force[lbf] / 8
- **Value Length (D):**	16 bits
- **Polarity (D):**	Leading bit (Bit 15) of MBOXnD is sign bit.
- **Scaling (D):**	Value = (sgn)*(327.67)*force[lbf] / 8
-	

- **Description**
  - This instruction closes a force loop about a load cell input. The desired force is loaded in MBOXnC, and the Load Cell Measurement (from an external load cell) is loaded in MBOXnD. The Sunstream servo controller will close the loop about the desired force and measured force, using the error (Force Input – Load Cell Measurement) to drive the spool. The maximum force of cylinder is assumed to be (100psi)*(Bore Area) lbf – where a positive force extends the cylinder, and a negative force retracts the cylinder. No rate limit is applied.
- **Notes:**
  - F000h represents Force of -100 lbf.
  - 1000h represents Force of +100 lbf.
  - Store Instruction Code to MBOXnB, right hand justified
  - Store 16 bit Force Input (desired force) to MBOXnC, right hand justified
  - Store 16 bit Load Cell Feedback to MBOXnD, right hand justified

5.2.3 State Programming

Set Maximum Acceleration

- **Instruction Code:**	0100h
- **Value Length:**	16 bits
- **Polarity:**	None. Set leading bit to 0.
- **Scaling:**	Value = 32*acceleration[in./s^2] / (max. stroke[in.])

- **Description:**
  - Limits the rate the servo accelerates from rest to terminal velocity. Setting applies to both extension and retraction. Acceleration setting may be changed at the completion of any servo command.
- **Notes:**
  - 1.0g = 386 in/s^2
  - Store Instruction Code to MBOXnB, right hand justified
  - Store Acceleration value to MBOXnC, right hand justified

Set Maximum Deceleration

- **Instruction Code:** 0200h
- **Value Length:** 16 bits
- **Polarity:** None. Set leading bit to 0.
- **Scaling:** Value = 32*deceleration[in./s^2] / (max. stroke[in.])
- **Description:**
  - Limits the rate the servo decelerates from velocity to rest. Setting applies to both extension and retraction. Deceleration setting may be changed at the completion of any servo command. ***Errata: This instruction not yet available.***
- **Notes:**
  - 1.0g = 386 in/s^2
  - Store Instruction Code to MBOXnB, right hand justified
  - Store Deceleration value to MBOXnC, right hand justified

Set Maximum Velocity

- **Instruction Code:** 0300h
- **Value Length:** 16 bits
- **Polarity:** None. Set leading bit to 0.
- **Scaling:** Value = 512*velocity[in/s] / (max. stroke[in.])
- **Description:**
  - Limits the maximum velocity of the servo. Setting applies to both extension and retraction. Velocity setting may be changed at the completion of any servo command. Velocity may also be modified in conjunction with a Move to Point or Move to Position command.
- **Notes:**
  - Store Instruction Code to MBOXnB, right hand justified
  - Store Velocity value to MBOXnC, right hand justified

Set Force Offset

- **Instruction Code:** 0400h
- **Value Length:** 32 bits
- **Polarity:** Leading bit (Bit 15) of MBOXnC is sign bit.
- **Scaling:** Value = (sgn)*(327.67)*force[lbf]
- **Description:**
  - This input informs the servo to expect a disturbance force which either acts to extend or retract the cylinder. This information increases the positional accuracy of the servo actuator. A force which will cause extension of the cylinder is a positive disturbance force, and a force which will cause retraction of the cylinder is a negative disturbance force. For example, an actuator disposed in a vertical orientation which extends against gravity will be considered to have a negative force disturbance equal to the weight of the load. Force Offset can be updated at the completion of any instruction, or can be combined with a move to position instruction (see Sect. 1.6.2).
- **Notes:**
  - FFFC0000h represents Force Offset of -800 lbf.
  - 00040000h represents Force Offset of +800 lbf.
  - Store Instruction Code to MBOXnB, right hand justified
  - Store upper 16 bits of Value to MBOXnC, right hand justified
  - Store lower 16 bits of Value to MBOXnD

Define Positional Resolution

- **Instruction Code:** 0500h
- **Value Length:** 16 bits

- **Polarity:** None. Set leading bit to 0.
- **Scaling:** Value = 32767*Resolution[in.] / (max. stroke[in.])
- **Description:**
  - After the servo has been sent an instruction to move to a point, it must reach an error window within the Positional Resolution before the servo is ready to accept a subsequent instruction.
- **Notes:**
  - 0010h represents a Positional Resolution of 0.0078 in. for a 16 inch stroke.
  - Store Instruction Code to MBOXnB, right hand justified
  - Store Resolution value to MBOXnC, right hand justified

Set Force Resolution

- **Instruction Code:** 0600h
- **Value Length:** 16 bits
- **Polarity:** None. Set leading bit to 0.
- **Scaling:** Value = (327.67)*force[lbf]
- **Description:**
  - After the servo has been sent an instruction to Ramp to a Force, it must reach an error window within the Force Resolution before the servo is ready to accept a subsequent instruction.
- **Notes:**
  - Store Instruction Code to MBOXnB, right hand justified

Set Force Rate

- **Instruction Code:** 0700h
- **Value Length:** 16 bits
- **Polarity:** None. Set leading bit to 0.
- **Scaling:** Value = (1/32)*force[lbf]/s.
- **Description:**
  - This value defines the rate by which the force increases or decreases after a Ramp to Force instruction is received.
- **Notes:**
  - Store Instruction Code to MBOXnB, right hand justified
  - 0001h represents force rate of 32 lbf. / sec.

Set Force Limit

- **Instruction Code:** 0800h
- **Value Length:** 32 bits
- **Polarity:** Leading bit (Bit 15) of MBOXnC is sign bit.
- **Scaling:** Value = (sgn)*(327.67)*force[lbf]
- **Description:**
  - ***Errata: This instruction will be employed with a future variation of the Soft Touch instruction which incorporates Velocity as a variable rather than Force Limit.***
- **Notes:**
  - FFFC0000h represents Force Limit of -800 lbf.
  - 00040000h represents Force Limit of +800 lbf.
  - Store Instruction Code to MBOXnB, right hand justified
  - Store upper 16 bits of Value to MBOXnC, right hand justified
  - Store lower 16 bits of Value to MBOXnD

5.3 Status & Diagnostic Feedback

- **(80xxh) Busy**
  - Servo is busy with a present command.
- **(90xxh) Ready**
  - Servo is ready to accept a new input from PLC
- **(1000h) Initializing**
  - Servo controller has received the Initialize instruction from the host (000Fh) and is in the process of initializing all sensors and registers. This value will be transmitted once in MBXnA after the servo has received the Initialize instruction.
- **(2000h) Initialization Completed**
  - Initialization is complete, and the servo position loop is ready to be enabled. This value will be transmitted once in MBXnA after the servo has received the Initialize instruction. At this point, the servo has been enabled in Flow Control Mode, and is ready to operate.
- **(F1xxh) Pressure fault**
  - The pressure in each cylinder chamber has fallen below 1 bar, indicating that there has been a loss of input pressure. The servo position loop is disabled, and the spool position is set to null. Once there is a loss of input pressure, the user must transmit an Operate command (0002h) to the servo to re-engage the servo position loop.
- **(F2xxh) Invalid Argument**
  - An invalid instruction has been received by the Sunstream Controller.
- **(xxxFh) Flow mode**
  - The position and force loops are disengaged, and the position of the spool is directly controlled.
- **(xxx0h) Position mode**
  - The servo position loop is engaged. All inputs directly control the cylinder position.
- **(xxx1h) Force mode**
  - The servo force loop is engaged. All inputs directly control the cylinder force.

6.0 Operational Example

The following is a step-by-step guide to initializing, enabling, and operating the Sunstream servo through the CAN2.0B link.

- Transmit to the Sunstream controller the word “**000Fh**” (Initiate) in MBOXnB.

- When the Sunstream servo receives the Initiate instruction, it will reply by transmitting “1000h” MBOXnA. The controller will then proceed with initializing all registers and synchronizing the servo sensors. During this period, the host will not be able to communicate with the Sunstream servo controller over the CAN link. Servo status is set to 1000h.

- After completion of the initialization process, the Sunstream servo will transmit “2000h” in MBOXnA. Servo status is set to 2000h. The host will once again be able to communicate with the Sunstream controller, and the servo is set in Flow Control Mode.

- To obtain initial state information, the host can transmit a request for information – such as “**3567h**” in MBOXnA to request the cylinder position, head pressure, rod pressure, and spool position.

- Poll the Sunstream controller to verify that the servo is in ready mode by requesting the servo status in any segment of MBOXnA. If the servo is in ready mode, the status will be set to “9xxxh”, and the servo will be ready to accept a new command. The servo can be placed in either Flow Control Mode or Position Control Mode by transmitting to the servo any Force or Position instruction.

- To move the servo to a mid-stroke position at 10in/s (for a 10” stroke), and check the present cylinder position and cylinder position command, transmit the following:

MBOXnA:	3400h
MBOXnB:	0020h
MBOXnC:	4000h
MBOXnD:	0200h