STRICTLY CONFIDENTIAL

PROJECT

Gas Lift

TITLE

MODBUS Protocol

ISSUE STATUS

Issue	Author	Date	Release status	Change request	AMENDMENTS
A	P.Hockley	18/05/10	Superseded	68	Reissued on new number
B	DM Reaves	19/7/2010	In-Review	N/A	Implement comments from 18Jun10 & 07Jul10 meetings

TABLE OF CONTENTS

1. PURPOSE

The purpose of this document is to define the MODBUS protocol and command structure for the Gaslift System.

2. OVERVIEW

This section describes the basic Hardware and software models. It does not concern itself with the location of the various system elements.

2.1 Software Overview

The system implements the following specifications, subject to any exclusions or deviations specified herein:

a) MODBUS-IDA Protocol specification V1.1b

b) MODBUS over serial line specification and implementation guide V1.02.

The system consists of controlling software (located on a PC/Laptop/Customer system) interfacing to the SCS, which in turn controls the DHCS. The MODBUS protocol is defined for the PC to SCS link only.

2.2 Hardware Overview

The Controlling PC hardware is customer dependant. This will however interface via MODBUS to the SCS, which in turn connects to the DHCS. The SCS when sending and receiving data from the DHCS cannot be interrupted by the Controlling software. To this end the SCS will require two processors, a single board computer (SBC) to provide the MODBUS interface to the controlling Software, and a PIC to interface to the DHCS. The PIC will use CTS to indicate to the SBC when the SCS/DHCS is busy and cannot be interrupted. The PC only needs to be aware that there will be times when a MODBUS command will not be actioned due to SCS/DHCS operation.

3. MODBUS INTERFACE DESCRIPTION

This section describes the active MODBUS commands that the SCS will respond to as slave. All other commands will receive the appropriate MODBUS exception code.

Data is transmitted in RTU format. An RS485 MODBUS interface is required initially with an option to extend to a LAN as a future enhancement. Both are configurable via the RS232 SBC configuration interface. Each byte is sent as 1 start bit, 8 data bits (least significant bit sent first), 1 parity bit, 1 stop bit. The default speed is 9,600 bps and default parity is even. Options allow other standard speeds and the use of odd or no parity. The no parity option requires 2 stop bits. Bytes are transmitted least significant bit first and multi-byte words most significant byte first except for the CRC word which is transmitted least significant byte first.

The fields in this section are preceded by a one-byte Device Address (configurable via the RS232 SBC configuration interface) and are followed by the standard MODBUS CRC field. Lower protocol levels are as defined by MODBUS for RS485. Fields are transmitted in the order listed.

The MODBUS interface structure will be split into commands and responses. The main point to note is that commands involving the DHCS take around 2 seconds to complete. This means that a MODBUS command to close an actuator will consist of the original command (and MODBUS response), followed two seconds later by a register read (registers are updated by the SCS once the action has been completed). Any command involving the DHCS that is sent before they have been updated will be ignored. This structure is shown in Diagram 1.

Each mandrel is allocated a unique MODBUS slave address in the range 1 to 247. Hence an SCS with multiple mandrels will be allocated multiple slave addresses.

Diagram 1: Command/Response Structure

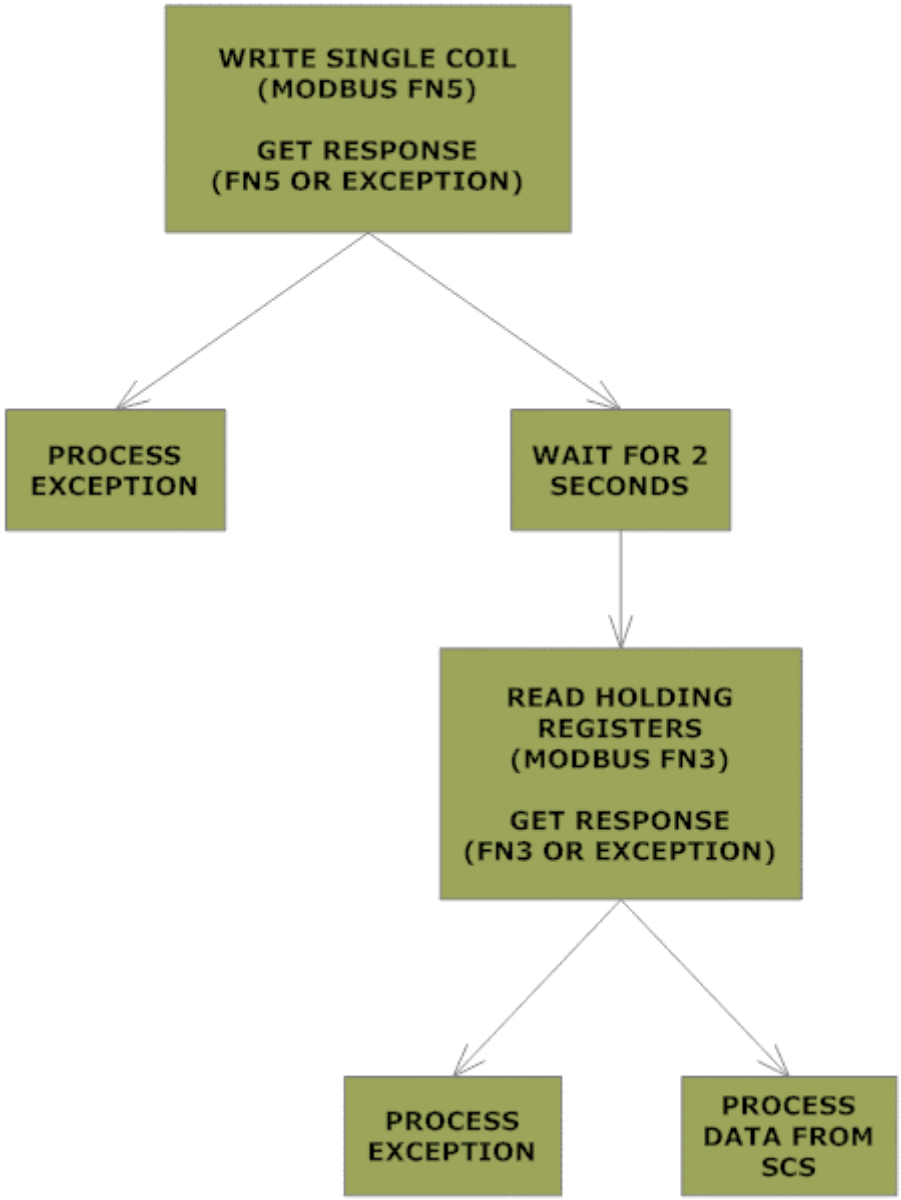

Diagram to be revised to omit “exceptions” if the response to a bad or corrupted command is to ignore it rather than respond to it.

Exception “Slave Device Busy” was recommended for deletion by Wingpath.

3.1 Write Single Coil (Fn5)

Important: The reason write coil is used to access the actuators (and not just putting them in a holding register) is to force the user to change one coil at a time (as this is how the DHCS works). The use of a register may encourage the user to believe that actuators can be changed as a group, and this should be avoided.

The command is also used to cause the SCS to request a status update from the DHCS without changing any actuator state. Status is updated whether an actuator state is changed or not, but due to the DHCS delay, is not included in the response to this command. Instead, all versions of this command are normally followed by a Read Holding Registers (Fn3) command.

The form for command and response is given below.

Command:

Function Code	1 Byte	0x05	
Output Address	2 Bytes	0x0000 to 0x0006	0, DHCS Status only, 1-6, actuator open/close
Output Value	2 Bytes	0x0000 or 0xFF00	0x0000 Close Actuator 0xFF00 Open Actuator

Normal Response:

Function Code	1 Byte	0x05	
Output Address	2 Bytes	0x0000 to 0x0006	0, DHCS Status only, 1-6, actuator open/close
Output Value	2 Bytes	0x0000 or 0xFF00	0x0000 Close Actuator 0xFF00 Open Actuator

Note that the response is identical to the command; this informs the Master that the command was accepted and is being processed.

3.2 Read Holding Registers (Fn3)

The form for command and response is given below.

This command is used to read the status of the DHCS after changing an actuator state or just to read the most recent status of the DHCS held by the SCS. It should be preceded by an Fn5 command (Open/Close Actuator or DHCS Status Only) if current data is required from the DHCS (eg temperature) because data is only updated following an Fn5 command.

Command:

Function Code	1 Byte	0x03	
Starting Address	2 Bytes	0x0000	Read All registers, starting at address 0
Quantity of Registers	2 Bytes	0x0008	Read all 8 Registers

Note: This is the only form the command can take, it must always request all 8 registers, starting at address 0. Any other combination will result in an exception error.

Response:

Function Code	1 Byte	0x03	
Byte Count	1 Byte	0x0010	Returns all 8 registers, 16 bytes
Registers Value	16 Bytes		Always return 16 bytes

The 9 registers (2 bytes per register) contain the following information.

Register	Data	Format / Meaning
0	SCS Manual Mode Status	0x0001 if the SCS box is in Manual Mode, otherwise 0x0000
1	Communication Status	0x000N. Communication status of DHCS, includes error codes. See Section 3.2.1.
2	Actuator Status (1)	0xNNNN. Actuator position and Stuck Faults. See Section 3.2.2
3	Actuator Status (2)	0xNNNN. Actuator short & open circuit Faults. See Section 3.2.3
4	Cap Bank Health	0x00NN. Capacitor Bank Health 0-100%
5	Annulus Pressure	0xNNNN. Pressure in the Annulus (0-10,000 Psi)
6	Production Pipe Pressure	0xNNNN. Pressure in the Production Pipe (0-10,000 Psi)
7	Production Pipe Temperature	0x00NN. Temperature in the production pipe (0-125°C).

Register	Data	Format / Meaning
8	DHCS PCB Temperature	0x00NN. Temperature on the DHCS PCB (-40 to 150°C), transmitted with an offset of 40 as 0 to 190)

3.2.1 Communication Status

The least significant byte of this register contains the communication status of the last command sent to the DHCS by the SCS. Valid responses are given in the table below

Response	Meaning
0x0000	Communications OK.
0x0081	No response received from DHCS to command
0x0082	Checksum error in DHCS response

3.2.2 Actuator Status (1)

The actuator Status (1) word is split into two bytes, described below (MSB first).

Actuator Position Byte:

Bit	Name	Meaning
7	None	Always 0
6	None	Always 0
5	A6	Actuator 6 Position: 0 = closed, 1 = open
4	A5	Actuator 5 Position: 0 = closed, 1 = open
3	A4	Actuator 4 Position: 0 = closed, 1 = open
2	A3	Actuator 3 Position: 0 = closed, 1 = open
1	A2	Actuator 2 Position: 0 = closed, 1 = open
0	A1	Actuator 1 Position: 0 = closed, 1 = open

Actuator Stuck Byte:

Bit	Name	Meaning
7	None	Always 0
6	None	Always 0
5	A6	Actuator 6 Stuck: 0 = OK, 1 = Stuck in current Position
4	A5	Actuator 5 Stuck: 0 = OK, 1 = Stuck in current Position
3	A4	Actuator 4 Stuck: 0 = OK, 1 = Stuck in current Position
2	A3	Actuator 3 Stuck: 0 = OK, 1 = Stuck in current Position
1	A2	Actuator 2 Stuck: 0 = OK, 1 = Stuck in current Position
0	A1	Actuator 1 Stuck: 0 = OK, 1 = Stuck in current Position

3.2.3 Actuator Status (2)

The actuator Status (2) word is split into two bytes, described below (MSB first).

Open Circuit Coil Byte:

Bit	Name	Meaning
7	None	Always 0
6	None	Always 0
5	A6	Actuator 6: 0 = OK, 1 = Open Circuit
4	A5	Actuator 5: 0 = OK, 1 = Open Circuit
3	A4	Actuator 4: 0 = OK, 1 = Open Circuit
2	A3	Actuator 3: 0 = OK, 1 = Open Circuit
1	A2	Actuator 2: 0 = OK, 1 = Open Circuit
0	A1	Actuator 1: 0 = OK, 1 = Open Circuit

Short Circuit Coil Byte:

Bit	Name	Meaning
7	None	Always 0
6	None	Always 0
5	A6	Actuator 6: 0 = OK, 1 = Short Circuit
4	A5	Actuator 5: 0 = OK, 1 = Short Circuit
3	A4	Actuator 4: 0 = OK, 1 = Short Circuit
2	A3	Actuator 3: 0 = OK, 1 = Short Circuit
1	A2	Actuator 2: 0 = OK, 1 = Short Circuit
0	A1	Actuator 1: 0 = OK, 1 = Short Circuit

4. READ DEVICE IDENTIFICATION (Fn43)

This command will return Vendor Name, Product Code, and system revision numbers that have been entered into an SCS. Where an SCS responds to multiple slave addresses, the data returned will be identical for each address for that SCS.

Command:

Function Code	1 Byte	0x2B	
MEI Type	1 Byte	0x0E	
Read Dev Id code	1 Byte	0x01	Request to get basic device identification
Object Id	1 Byte	0x00	First (and only) object to obtain

Note: This is the only form the command can take. Any other combination will be ignored. [Note for Wingpath: should we use an exception response here or anywhere? See note against diagram 1]

Response:

Function Code	1 Byte	0x2B	
MEI Type	1 Byte	0x0E	
Read Dev Id code	1 Byte	0x01	Request to get basic device identification
Conformity level	1 Byte	0x01	basic identification
More Follows	1 Byte	0x00	
NextObjectId	1 Byte	0x00	
Number Of Objects	1 Byte	0x03	
Object Id	1 Byte	0x00	
Object Length	1 Byte	0xnn	nn = length of vendor name field
Object Value	n Bytes		Vendor Name (free text)
Object Id	1 Byte	0x01	
Object Length	1 Byte	0xnn	nn = length of product code field
Object Value	n Bytes		Product code (free text)
Object Id	1 Byte	0x02	
Object Length	1 Byte	0xnn	nn = length of Major Minor Revision field
Object Value	n Bytes		Major Minor Revision (free text)