Integrated Monitoring and Control System

iMAC & iMAC2 Controller Serial Communications Manual

Version: 3 - October 2020

Designed and Manufactured in Australia by Ampcontrol Pty Ltd

WARNING!

The **warning** symbol highlights a potential risk of **injury or death**.

Please share these warnings with other operators.

CAUTION!

The **caution** symbol highlights a potential risk of **damage to equipment**.

Please share these cautions with other operators.

NOTE

The **note** symbol highlights **key information**.

Please share these notes with other operators.

ENVIRO

The **enviro** (environmental) symbol highlights areas which may have an impact on the surrounding **fauna and/or flora**.

Copyright Notice

Disclaimer

While every effort has been made to assure the accuracy and clarity of this document, AMPCONTROL PTY LTD assumes no liability resulting from any omissions in this document, or from misuse of the information obtained herein. The information in this document has been carefully checked and is believed to be entirely reliable with all of the necessary information included. AMPCONTROL PTY LTD reserves the right to make changes to any products described herein to improve reliability, function, or design, and reserves the right to revise this document and make changes from time to time in content hereof with no obligation to notify any persons of revisions or changes. AMPCONTROL PTY LTD does not assume any liability arising out of the application or any use of any product or circuit described herein; neither does it convey license under its patent rights or the rights of others.

Before You Begin

Thank you for purchasing the Ampcontrol iMAC System.

WARNING!

In the interests of **safety and correct equipment operation**, please take the time to read and understand the content in this manual.

This manual (version 3) pertains to iMAC/iMAC2 Controller Operating System Firmware V4.2. For superseded Operating System Firmware V4.0 and V4.1, please refer to V2 of this manual. For superseded Operating System Firmware V3.5 or less, please refer to V1 of this manual.

If your iMAC controller is not running the latest Operating System Firmware V4.2, it is critically recommended to update as soon as practicable to ensure the safety and reliability of your system.

Ampcontrol Contact Details

7 Billbrooke Close, Cameron Park, NSW, 2285

P +61 1300 267 373 | F +61 2 4903 4888

EMAIL: customerservice@ampcontrolgroup.com

WEB: ampcontrolgroup.com

TABLE OF CONTENTS

TABLE OF FIGURES

1 DOCUMENT SCOPE

1.1 Document Scope

This document is intended to provide a detailed explanation of the communications protocols supported by the iMAC Controller's serial port.

This document is not intended to provide information on the operation of the overall iMAC System, individual modules or instruction on programming the iMAC Controller or modules. Please refer the relevant supplementary documents for this information.

1.2 Supplementary Documents

The iMAC Serial Communications Manual is intended to be read in conjunction with the following documents:

1.2.1 System Documentation

- IMACB068 iMAC System User Manual
- IMACB094 iMAC System Installation Requirements
- IMACB182 iMAC SIL Emergency Stop Qualification
- IMACB005 iMAC Module Programming Manual

1.2.2 Module Documentation

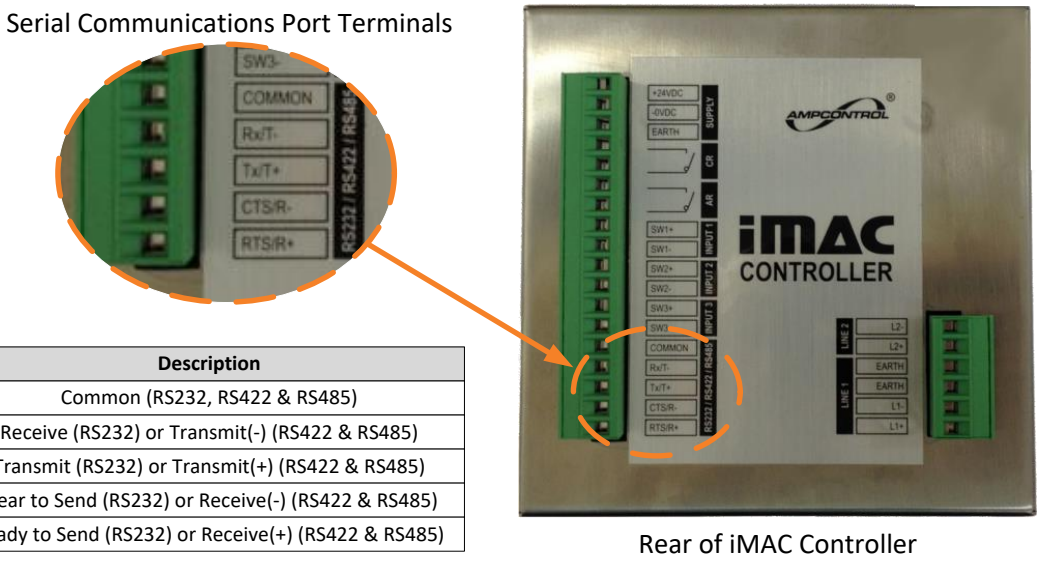
- IMACB003 iMAC RO4 Module Technical Datasheet
- IMACB018 iMAC LPU Module Technical Datasheet
- IMACB020 iMAC DI8 Module Technical Datasheet
- IMACB045 iMAC IIM Module Technical Datasheet
- IMACB046 iMAC DI4 Module Technical Datasheet
- IMACB047 iMAC EOL/MEOL Module Technical Datasheet
- IMACB060 iMAC LED4 Module Technical Datasheet
- IMACB061 iMAC SSW Module Technical Datasheet
- IMACB062 iMAC SQM Module Technical Datasheet
- IMACB066 iMAC AIM Module Technical Datasheet
- IMACB067 iMAC RTD1 Module Technical Datasheet
- IMACB141 iMAC ARM Module Technical Datasheet
- IMACB142 iMAC CRM Module Technical Datasheet
- IMACB143 iMAC EMM Module Technical Datasheet
- IMACB144 iMAC GRM Module Technical Datasheet
- IMACB146 iMAC IRK Keypad Technical Datasheet
- IMACB147 iMAC PIM Module Technical Datasheet
- IMACB148 iMAC SIM-G Module Technical Datasheet
- IMACB149 iMAC SIM-G2 Module Technical Datasheet
- IMACB150 iMAC SIM-T Module Technical Datasheet
- IMACB151 iMAC MLB Barrier Technical Datasheet
- IMACB152 iMAC SLB Barrier Technical Datasheet
- IMACB154 iMAC SIM-P Module Technical Datasheet
- IMACB172 iMAC LIM Module Technical Datasheet

2 iMAC CONTROLLER COMMUNICATIONS OVERVIEW

The iMAC Controller is equipped with a serial communication port. This port is electrically isolated and supports RS232, RS422 or RS485 modes with a selectable baud rate of 2400, 4800 or 9600.

> **NOTE**
>
> The **communication port has shared use**: it can be used as the Controller **programming port** (for loading of application (SLP) software) or as a **serial communications port** (for exchanging data with compatible serial devices).

2.1 Specifications



Terminal	Description
COMMON	Common (RS232, RS422 & RS485)
Rx/T-	Receive (RS232) or Transmit(-) (RS422 & RS485)
Tx/T+	Transmit (RS232) or Transmit(+) (RS422 & RS485)
CTS/R-	Clear to Send (RS232) or Receive(-) (RS422 & RS485)
RTS/R+	Ready to Send (RS232) or Receive(+) (RS422 & RS485)

Figure 2.1: Location of the Serial Communications Port Terminals

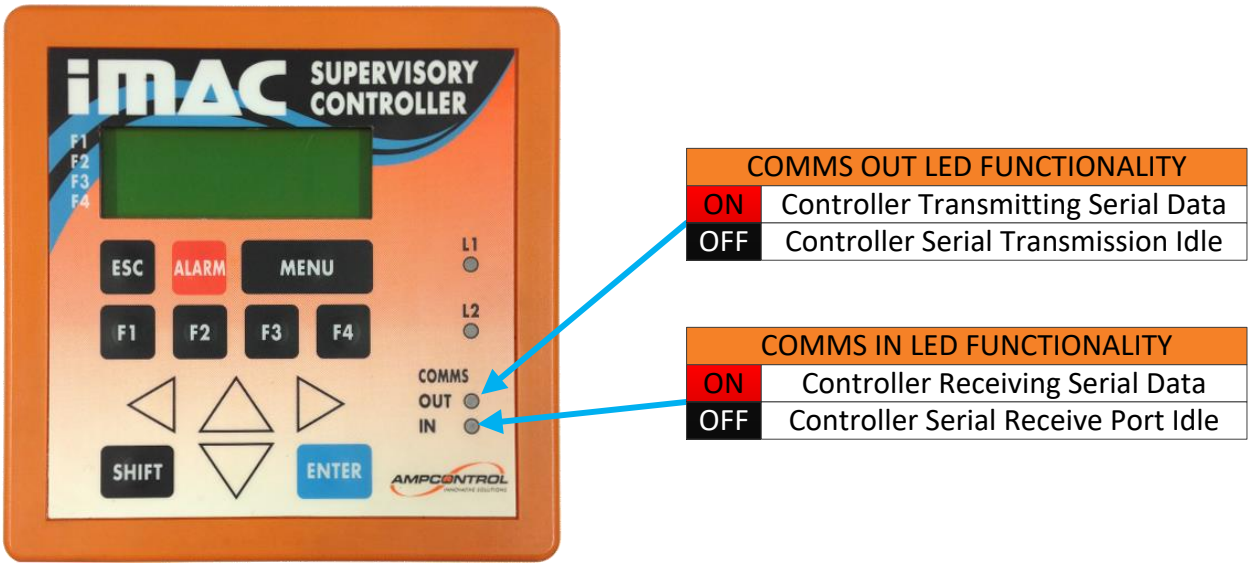


Figure 2.2: COMMS LED Functionality

Table 1: Serial Communications Port Specifications

Serial Communications Port Specifications	
Use	Serial Communication / Programming Port
Physical	5x Individual Screw Clamp Terminals
Available Interfaces	RS232/RS422-RS485
Supported Protocols	Modbus Slave RTU Protocol Modbus Master Protocol Ampcontrol IP2 Protocol L1 Maintenance Protocol L2 Maintenance Protocol
Baud Rate	2400, 4800, 9600
Optically Isolated	2.5kV

2.2 Serial Port Wiring

2.2.1 RS232 Wiring

NOTE

While the iMAC Controller supports hardware flow control via CTS/RTS signals for compatibility with RS232 devices, the iMAC Controller itself doesn't require these connections to function correctly.

Most RS232 devices will only require 0V, Rx and Tx connections.

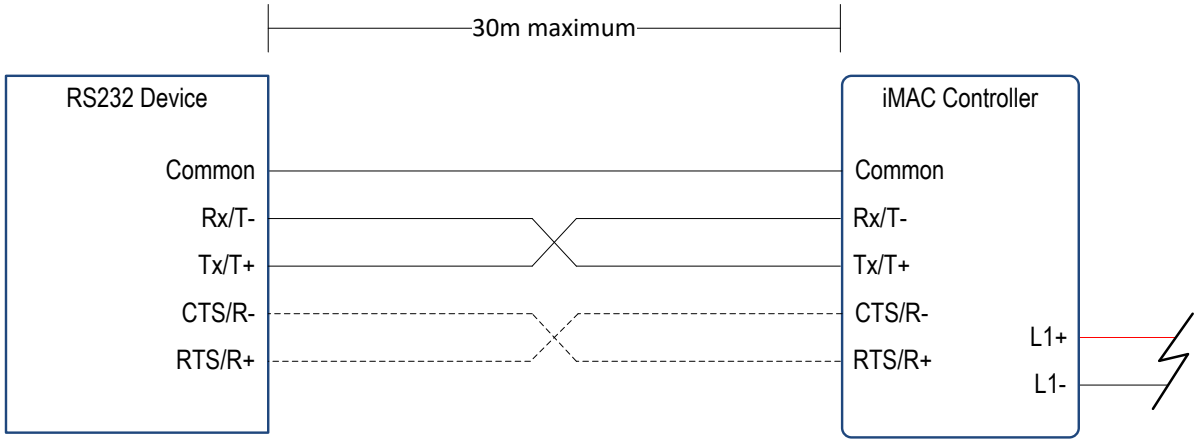


Figure 2.3: Serial Port Wiring Diagram - RS232

2.2.2 RS422 Wiring

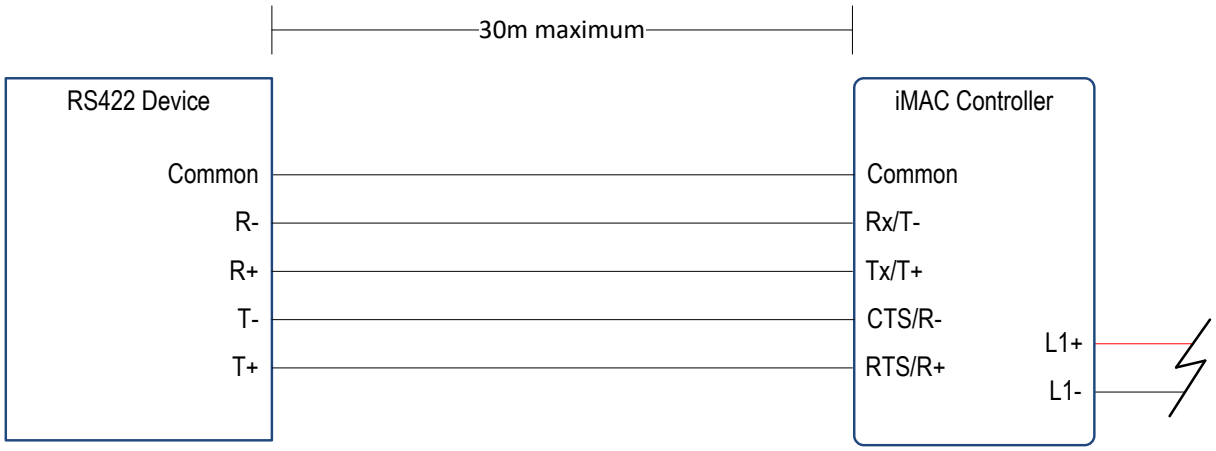


Figure 2.4: Serial Port Wiring Diagram – RS422

2.2.3 RS485 Wiring Diagram

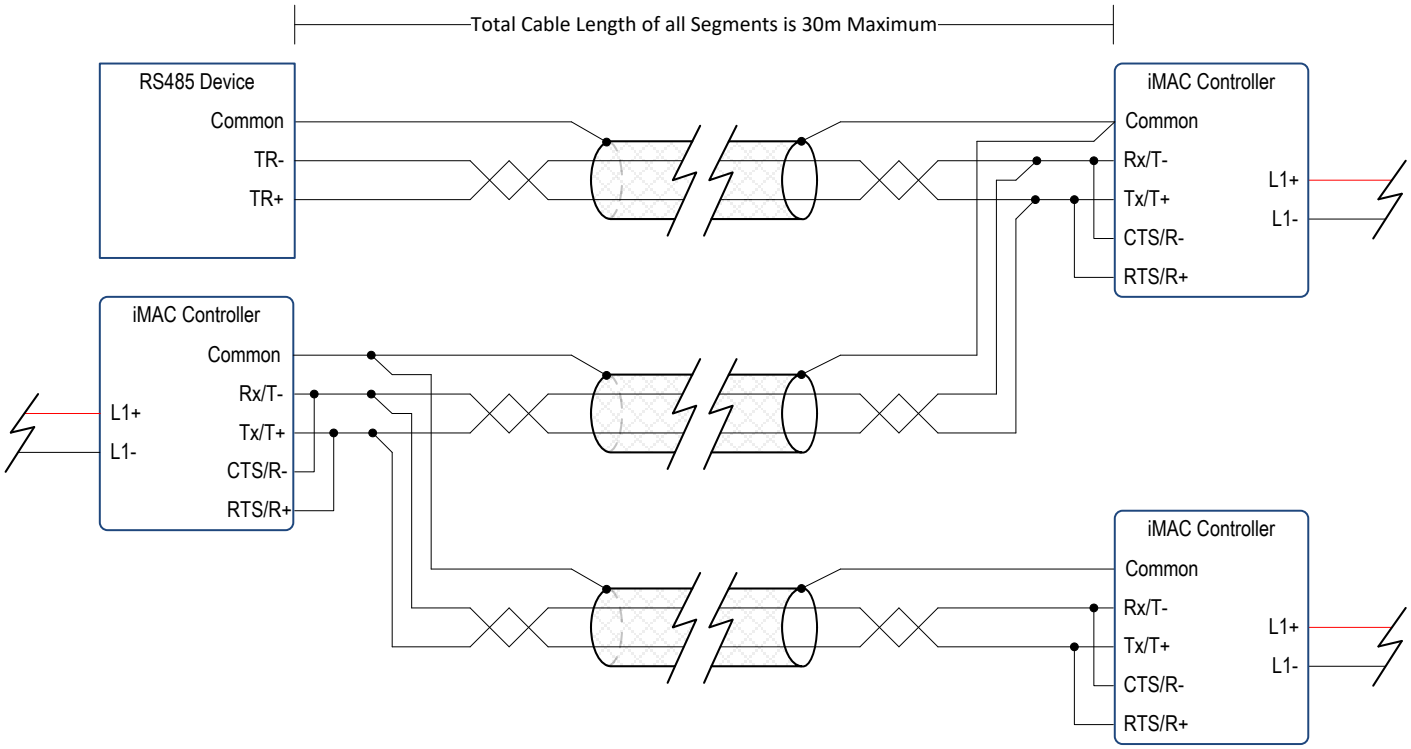


Figure 2.5: Serial Port Wiring Diagram – RS485

2.3 Serial Port Mode

2.3.1 Programming Mode

Programming mode supports loading of application (SLP) software. Programming mode is entered by pressing and holding down the iMAC Controller keypad [SHIFT+ENTER] keys during power-up. This mode allows the controller to connect to Ampcontrol's 'itools' PC software which can be used to download compiled application (SLP) software HEX files. Unless programming mode is entered during power up, the controller's serial port defaults to communications mode.

2.3.2 Communications Mode

Serial communication mode supports the following communication protocols:

- Slave Modbus (7 commands supported)
- Master Modbus (3 commands supported, with up to 8 x 32 word transfer blocks)
- Ampcontrol IP/IP2 protocols (compatible with legacy Ampcontrol IP, IP2 or SLIP systems)
- Ampcontrol Field Service protocols (L1 Maintenance and L2 Maintenance)

The communications port is set up via the iMAC Controller System Menu.

The user can usually access the System Menu by pressing [SHIFT+MENU]. In some cases the application (SLP) software may have disabled this function, in these cases the System Menu will be assessable according to an application software defined procedure, refer to the specific application software documentation for instructions on accessing the System Menu in these cases.

The System Menu allows access to the following pages:

- Maintenance Page
- Serial Communications Page
- Version Page

To access the Serial Communications Page from the System Menu, press the [F3] function key.

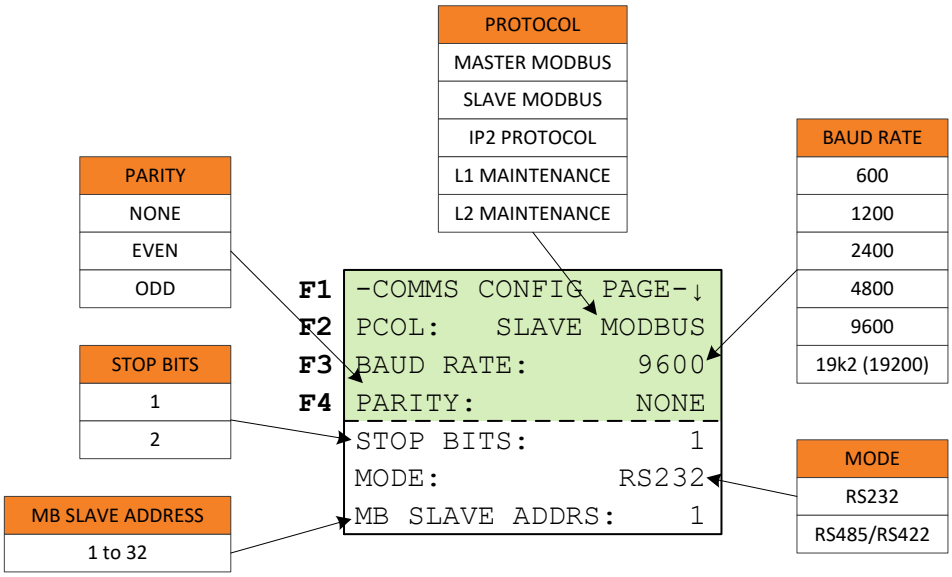


Figure 2.6: iMAC System Menu - Serial Communications Page

2.4 Modbus Protocol

The iMAC Controller uses industry standard Modbus protocol for exchanging data with compatible serial devices.

To configure the iMAC Controller serial communications protocol, select the required Modbus protocol using the **PCOL** parameter in the Serial Communications section of the System Menu. From this menu it is also possible to set the **BAUD RATE**, **PARITY**, **STOP BITS**, **MODE** & the **MB SLAVE ADDRESS** as required.

2.4.1 Master Modbus Protocol (for legacy system support only)

If Master Modbus protocol is selected, the iMAC Controller will act as the master device, interrogating connected slave devices as specified by the iMAC application (SLP) software. There can only be one Master device in a network, in this case the master is the iMAC Controller. The Controller controls information exchange according to its application (SLP) software instructions. Master Modbus Mode supports 3 Modbus commands:

- Read Holding Registers (03)
- Read Input Registers (04)
- Write Multiple Registers (16)

A summary of each supported command is as follows:

Read Holding Registers (03)

This reads the holding registers in the slave device with slave address ModbusSlvAddr (1..32)
The registers are read from the selected slave, starting at register address RegAddressSRC ($0000..$FFFF).
These will be copied into the iMAC Modbus register address starting at RegAddressDST ($0000..$05FF).
A consecutive block of registers (words) will be copied NumRegisters.

Read Input Registers (04)

This reads the input registers in the slave device with slave address ModbusSlvAddr (1..32)
The registers are read from the selected slave, starting at register address RegAddressSRC ($0000..$FFFF).
These will be copied into the iMAC Modbus register address starting at RegAddressDST ($0000..$05FF).
A consecutive block of registers (words) will be copied NumRegisters.

Preset Multiple Registers (16)

This writes registers into the slave device with slave address ModbusSlvAddr (1..32)
The registers are read, starting from the iMAC Modbus register address RegAddressSRC ($0000..$05FF).
These will be copied into the selected slave device, starting at address RegAddressDST ($0000..$FFFF).
A consecutive block of registers (words) will be copied NumRegisters.

The iMAC Controller Application (SLP) Software Syntax for implementing these commands is as follows:

CommsPortMode { MasterModbus, ModbusCMD, ModbusSlvAddr, RegAddressSRC/DST, RegAddressSRC/DST, NumRegisters }

Where:

ModbusCMD = RHR (Read Holding Register) or RIR (Read Input Register) or PMR (Preset Multiple Register)
RegAddressSRC/DST and NumRegisters are values as per the command definitions above.

Up to four Master Modbus commands can be comer separated into a single CommsPortMode{ } function, for example:

CommsPortMode{MasterModbus, RHR, 1, $480+1, DataTable1, 16, PMR, 2, DataTable2, $480+1, 24, RHR, 1, $498+1, DataTable3, 8, PMR, 2, DataTable4, $490+1, 16}

This function results in four Modbus transactions as follows:

Read 16 Holding Registers from iMAC Controller with slave address 1, starting at DPT $480* and copy into Controller memory array DataTable1.

Write 24 registers from Controller memory array DataTable2 to iMAC Controller with slave address 2, starting at DPT $480*

Read 8 Holding Registers from iMAC Controller with slave address 1, starting at DPT $498* and copy into Controller memory array DataTable3.

Write 16 registers from Controller memory array DataTable4 to iMAC Controller with slave address 2, starting at DPT $490*

*Note: +1 is required in the function call for slave device register addresses to convert from iMAC Base 0 addresses to Modbus Protocol Base 1 addresses.

Only one instance of the CommsPortMode{ } function is permitted per application (SLP) software.

2.4.2 Slave Modbus Protocol (recommended communications protocol)

If Slave Modbus (RTU) protocol is selected, the iMAC Controller will act as a slave device, with the slave address able to be set in the Serial Communications section of the System Menu. Seven commands are supported in Slave Mode:

- Read Holding Registers (03)
- Read Input Registers (04)
- Force Single Coil (05)
- Write Single Register (06)
- Read Exception Status (07)
- Write Multiple Registers (16)
- Report Slave ID (17)

A summary of each supported command is as follows:

Read Holding Registers (03)

This command reads registers within the iMAC register address space ($0000..$05FF) into the master device.

Read Input Registers (04)

This command reads registers within the iMAC register address space ($0000..$05FF) into the master device.

Force Single Coil (05)

This forces a single bit in the iMAC address space ($0000..$05FF:BITx).
To address the single bit, the Modbus command must set the bit address as:
Bit Addr = iMAC Register x 16 + BITx, where iMAC Register = $0000..$05FF and BITx = 0..15.

Write Single Registers (06)

This command writes a single register into the iMAC register address space ($0000..$05FF) from the master device.

Read Exception Status (07)

The iMAC response to this command is to send back the least significant byte of address $0000. This provides a fast read of a general purpose byte, the contents of which can be setup by the application (SLP) software. This byte can therefore be used for status & data change flags. These flags can then trigger the reading of large tables by the Modbus master device.

Write Multiple Registers (16)

This command writes multiple registers into the iMAC register address space ($0000..$05FF) from the master device.

Report Slave ID (17)

The iMAC response to this command is:

<ID><CRSTATE><HARDWARE><FIRMWARE><SOFTWARE><USERWARE>

The first byte is slave ID type, which is $55 for the iMAC controller.
The second byte, CRSTATE is $00 if the control relay is open, $FF if the control relay is closed.
The HARDWARE, FIRMWARE, SOFTWARE, USERWARE parameters are each 20 byte, left justified non-terminated ASCII strings.
The HARDWARE string is the iMAC controller hardware version.
The FIRMWARE string is the iMAC controller firmware version.
The SOFTWARE string is the iMAC controller main operating software version.
The USERWARE string is the iMAC controller application (SLP) software version.

Illegal Commands (01, 02, 08 to 15, 18 to 127)

The iMAC response to any of these commands is a Modbus exception.
To all other commands (0,128...255) the iMAC controller will not respond.

2.5 Ampcontrol IP2 Protocol (for legacy system support only)

The Ampcontrol IP protocols maintain compatibility with legacy Ampcontrol IP, IP2 and SLIP systems. For new installations it is highly recommended to utilise Modbus Slave protocol.

The IP protocol allows data exchange between iMAC Controller and other Ampcontrol products using a simple ASCII serial protocol. The IP protocol allows direct point to point communications between iMAC Controller and another Ampcontrol Master IP device. The IP2 protocol extends the IP protocol allowing multi-drop communications between a single Ampcontrol IP2 master device and multiple slave iMAC Controllers/IP2 devices. The IP2 protocol allows up to up to 32 slave devices.

Set the **PCOL** parameter to "IP2 Protocol" in the Serial Communications section of the System Menu. From this menu it is also possible to set the **BAUD RATE**, **PARITY**, **STOP BITS**, **MODE** & the **MB SLAVE ADDRESS** as required.

The hardware communications *mode* is either RS232 or RS485, however RS485 mode is required for multi-drop IP2 protocol if two or more slave devices are to be connected.

When Ampcontrol IP2 Protocol is used, the PLC is the master device and the iMAC Controller is the slave device. The *slave* parameter is the iMAC Controller's "Slave Address" which can be set between 1 and 32.

The IP2 Protocol reads data from the iMAC General Buffer register address space ($0480 to $04FF and $0580 to $05FF). The General Buffer register address space is mapped to mimic eight (8) IPSI (Integrated Protection Serial Interface) data blocks. Requests for IPSI 1 to 8 will read the General Buffer. Requests for IPSI 9 to 15 will be reported as "OFFLINE".

The application (SLP) software is responsible for transferring data into the General Buffer for access by the IP2 Protocol. The data should be copied to the address locations for retrieval. Each "IPSI data block" is constructed as a group of 32 registers, of which only the first 30 registers can be used for exporting data.

IPSI Block Number	Data Point Table Address Range	
	Decimal	Hexadecimal
1	1152 to 1182	480h to 49Eh
2	1184 to 1214	4A0h to 4BEh
3	1216 to 1246	4C0h to 4DEh
4	1248 to 1278	4E0h to 4FEh
5	1408 to 1438	580h to 59Eh
6	1440 to 1470	5A0h to 5BEh
7	1472 to 1502	5C0h to 5DEh
8	1504 to 1534	5E0h to 5FEh

Note: the 31st & 32nd registers in each IPSI block are reserved and should not be used. These registers are reserved for the IP2 Protocol checksum and should not be written to by the iMAC Controller application (SLP) software.

3 THE iMAC CONTROLLER'S DATA POINT TABLE

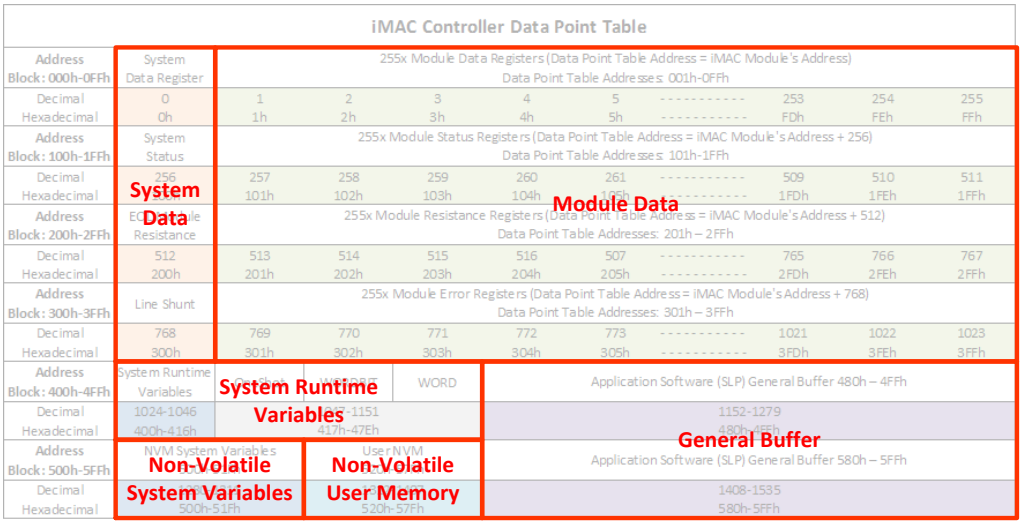
The iMAC Controller's Data Point Table is a list of 1536 16-bit registers. Each register has its own address in the range 0000h to 05FFh.

The iMAC Controller's Data Point Table is used for the exchange of all serial communication data, with the exception of IP2 protocols which can only access designated blocks of data in the two General Buffer segments.

3.1 Data Point Table

iMAC Controller Data Point Table										
Address Block: 000h-0FFh	System Data Register	255x Module Data Registers (Data Point Table Address = iMAC Module's Address) Data Point Table Addresses: 001h-0FFh								
Decimal	0	1	2	3	4	5	----------	253	254	255
Hexadecimal	0h	1h	2h	3h	4h	5h	----------	FDh	FEh	FFh
Address Block: 100h-1FFh	System Status	255x Module Status Registers (Data Point Table Address = iMAC Module's Address + 256) Data Point Table Addresses: 101h-1FFh								
Decimal	256	257	258	259	260	261	----------	509	510	511
Hexadecimal	100h	101h	102h	103h	104h	105h	----------	1FDh	1FEh	1FFh
Address Block: 200h-2FFh	EOL Module Resistance	255x Module Resistance Registers (Data Point Table Address = iMAC Module's Address + 512) Data Point Table Addresses: 201h – 2FFh								
Decimal	512	513	514	515	516	507	----------	765	766	767
Hexadecimal	200h	201h	202h	203h	204h	205h	----------	2FDh	2FEh	2FFh
Address Block: 300h-3FFh	Line Shunt	255x Module Error Registers (Data Point Table Address = iMAC Module's Address + 768) Data Point Table Addresses: 301h – 3FFh								
Decimal	768	769	770	771	772	773	----------	1021	1022	1023
Hexadecimal	300h	301h	302h	303h	304h	305h	----------	3FDh	3FEh	3FFh
Address Block: 400h-4FFh	System Runtime Variables	OneShot	WORDBIT	WORD	Application Software (SLP) General Buffer 480h – 4FFh					
Decimal	1024-1046	1047-1151			1152-1279					
Hexadecimal	400h-416h	417h-47Eh			480h-4FFh					
Address Block: 500h-5FFh	NVM System Variables 500h-51Fh		User NVM 520h-57Fh		Application Software (SLP) General Buffer 580h – 5FFh					
Decimal	1280-1311		1312-1407		1408-1535					
Hexadecimal	500h-51Fh		520h-57Fh		580h-5FFh					

The Data Point Table is broken up into a number of segments: System Data, Module Data, System Runtime Variables, Non-Volatile System Variables, Non-Volatile User Memory & two General Buffer segments. These are explained in detail in the following sections.



3.2 System Data Segment

The System Data is a group of 4 registers that are addressed as follows:

Address		Register Content
Decimal	Hexadecimal	
0	000h	System Data (see bit definitions)
256	100h	System Status (see bit definitions)
512	200h	EOL Module Resistance
768	300h	L1 Fieldbus Shunt Resistance

3.2.1 System Data Register

The bits within the System Data register are defined by the Controllers application (SLP) software. Typical bit definitions are as below. Please refer to your specific application (SLP) software for actual definitions.

WARNING!

CRM and ARM modules require specific application (SLP) software to operate, please ensure that your application (SLP) software is configured correctly if you require CRM and/or ARM functions

Address 0 (000h) System Data Register						
Address		Bit	Name	Description	r/w	SLP Tag
Dec.	Hex.					
0	000h	0	CRM Output/ GRM0 Output	Set by specific application (SLP) software when the CR is closed. CRM or GRM modules can output this information. LED4 or RO4 modules can output this information if set to Address 0	w	
		1	ARM Output/ GRM1 Output	Set by specific application (SLP) software when the AR is closed. ARM or GRM modules can output this information. LED4 or RO4 modules can output this information if set to Address 0	w	
		2	GRM2 Output	Set as required in application (SLP) software. GRM modules can output this information. LED4 or RO4 modules can output this information if set to Address 0	w	
		3	GRM3 Output	Set as required in application (SLP) software. GRM modules can output this information. LED4 or RO4 modules can output this information if set to Address 0	w	
		4	GRM4 Output	Set as required in application (SLP) software. GRM modules can output this information	w	
		5	GRM5 Output	Set as required in application (SLP) software. GRM modules can output this information	w	
		6	GRM6 Output	Set as required in application (SLP) software. GRM modules can output this information	w	
		7	GRM7 Output	Set as required in application (SLP) software. GRM modules can output this information	w	

Address 0 (000h) System Data Register						
Address		Bit	Name	Description	r/w	SLP Tag
Dec.	Hex.					
		8 to 15	*Application (SLP) Software Dependent*			

3.2.2 System Status Register

The bits within the System Status register have the following definitions:

Address 256 (100h) System Status Register						
Address		Bit	Name	Description	r/w	SLP Tag
Dec.	Hex.					
256	100h	0	Refresh Cycle Complete	Set by the controller at the end of every refresh cycle (refresh cycle = all 255 module addresses have been scanned). May be cleared by writing a 0 into this bit using application (SLP) software.	r/w	RefreshCycle
		1	L1 Checksum Error Latch	Set by the controller when a checksum error is detected on L1 fieldbus. May be cleared by writing a 0 into this bit using application (SLP) software.	r/w	L1ChksErr
		2	L1 OK Latch	Set by the controller when a L1 fieldbus scan completes with no errors detected. May be cleared by writing a 0 into this bit using application (SLP) software.	r/w	L1Ok
		3	L2 Checksum Error Latch	Set by the controller when a checksum error is detected on L2 fieldbus. May be cleared by writing a 0 into this bit using application (SLP) software.	r/w	L2ChksErr
		4	L2 EdgeDetect	Set by controller when it detects a single (one shot) valid iMAC Fieldbus command on its L2 port following a controller power up. Once a valid command has been detected, this bit is maintained provided L2 Fieldbus rising and falling signal edges continue to be detected with a 39ms window. Set if edges are detected within this window, cleared if edges are not detected within this window.	r/w	L2EdgeDetect
		5	*Not Used / System Use Only / Do not use*			
		6	*Not Used / System Use Only / Do not use*			

Address 256 (100h) System Status Register						
Address		Bit	Name	Description	r/w	SLP Tag
Dec.	Hex.					
		7	L2 Active	Set by controller when it detects four valid iMAC Fieldbus commands on its L2 port with no L2 EOL_Clash conditions. Cleared if: valid iMAC Fieldbus commands are not received within 2 seconds, or EOL_Clash conditions persist for longer than 2 seconds, or rising and falling L2 Fieldbus signal transitions are not detected within a 39ms timeframe.	r/w	L2Active
		8	Control Relay (CR) state	Status of the Control Relay. 1 = Energised. 0 = De-energised.	r	CtrlRelayState
		9	Auxiliary Relay (AR) state	Status of the Auxiliary Relay. 1 = Energised. 0 = De-energised.	r	AuxRelayState
		10	EOL OK	Set by controller when EOL module is being detected on the L1 fieldbus with no faults, clear otherwise.	r	EOL_OK
		11	EOL Open Circuit	Set by controller when EOL module is not detected on the L1 fieldbus and the fieldbus is not short circuited, clear otherwise.	r	EOL_OC
		12	EOL Short Circuit	Set by controller when the L1 fieldbus is short circuited (R<1000Ω), clear otherwise	r	EOL_SC
		13	EOL Clashed	Set by controller when two EOL modules are detected on the L1 fieldbus, clear otherwise.	r	EOL_CLASH
		14	Rx Active	Set by controller when its serial port is receiving data	r	UART_Rx
		15	Tx Active	Set by controller when its serial port is transmitting data	r	UART_Tx

3.2.3 EOL Series Resistance Register

The EOL Module Resistance register can be accessed at Address 512 (200h). This register contains the iMAC fieldbus loop resistance measured between iMAC controller and connected EOL module.

Address 512 (200h) EOL Series Resistance Register						
Address		Bit	Name	Description	r/w	SLP Tag
Dec.	Hex.					
512	200h	0 to 15	EOL Resistance	Measured series resistance of the L1 fieldbus loop from controller to EOL Module. Values are one to one, e.g. a reading of 00FFh (255) corresponds to a loop resistance of 255Ω Range: 0..1000	r	EolSeriesRes

3.2.4 Line Shunt Resistance Register

The Line Shunt Resistance register can be accessed at Address 768 (300h). This register contains the measured shunt resistance of the iMAC L1 fieldbus.

Address 768 (300h) Line Shunt Resistance Register						
Address		Bit	Name	Description	r/w	SLP Tag
Dec.	Hex.					
768	300h	0 to 15	Shunt Resistance	Measured leakage resistance between controller L1 fieldbus terminals. Values are one to one, e.g. a reading of 0FFFh (4095) corresponds to a shunt resistance of 4095Ω Range: 0..32768	r	EolShuntRes

3.3 Module Data Segment

The Module Data is a group of 1020 registers. Every iMAC module address (1..255) has four associated registers: DATA, STATUS, RESISTANCE and ERRORS. The relationship between module address and its four corresponding registers is defined as:

Module Register Addresses	Example: Register Addresses for a Module at Address 1		Register Content
	Decimal	Hexadecimal	
Module Address (1..255)	1	001h	Module Data (module dependant)
Module Address + 256	257	101h	Module Status (see bit definitions)
Module Address + 512	513	201h	Module Resistance
Module Address + 768	769	301h	Module Error Counters

3.3.1 Module Data Register

The function of the bits within the Module Data register is dependent upon the type of module that is addressed to that location. For information on bit functionality for each module, refer to the individual iMAC module datasheet.

Address 001 - 255 (001h - 0FFh) Module Data Register						
Address		Bit	Name	Description	r/w	SLP Tag
Dec.	Hex.					
001 to 255	001h to 0FFh	0	D0 / SYS0	Module Dependent (Refer to the individual module datasheets for bit definitions). Note: depending on corresponding Module Status Address ownership flags, this register will be input (default), output (system owned), or low-byte input/high-byte output (byte owned).	Module Dependent	
		1	D1 / SYS1			
		2	D2 / SYS2			
		3	D3 / SYS3			
		4	D4 / SYS4			
		5	D5 / SYS5			
		6	D6 / SYS6			
		7	D7 / SYS7			
		8	D8 / SYS8			
		9	D9 / SYS9			
		10	D10 / SYS10			
		11	D11 / SYS11			
		12	D12 / SYS12			
		13	D13 / SYS13			
		14	D14 / SYS14			
		15	D15 / SYS15			

3.3.2 Module Status Register

The Module Status register can be accessed at the address in the Data Point Table that corresponds to the Module Address + 256. The bits within the Module Status register have the following definitions, regardless of the module type that is configured at that particular address.

Address 257 - 511 (101h - 1FFh) Module Status Register						
Address		**Bit**	**Name**	**Description**	**r/w**	**SLP Tag**
Dec.	**Hex.**					
257 to 511	101h to 1FFh	0	On Scan Bit	Set by the controller when it first detects an input module address online on its L1 fieldbus. May be cleared by application (SLP) software. (POR=0)	r/w	OnScanBit
		1	L1 Clash Bit	Set by the controller when two input modules transmit different data on the same address, clear otherwise.	r	L1ClashBit
		2	Global Select	Must be intentionally set by application (SLP) software if the corresponding data address needs to be published up the L2 fieldbus, thus making it a Global Address.	r/w	Global
		3	L1 Owned Bit	Set by the controller when it detects an input module address on its L1 fieldbus, cleared if module not detected.	r	L1OwnBit
		4	L2 Owned Bit	Set by the controller when it detects the corresponding address on its L2 fieldbus. The Global Select bit for this address must be set to allow the controller to “see” up its L2 fieldbus. This should be done during the STARTUP segment of the application (SLP) software.	r	L2OwnBit
		5	System Owned Bit	Must be intentionally set application (SLP) software when the corresponding data register is to be owned by the system. Setting this bit forces the corresponding data register to be an output. All addresses are assumed to be an input unless this bit is set. If an input module is connected to a system owned address on L1, a L1 clash error will occur.	w	SysOwnBit
		6	L2 Clash Bit	Set by the controller when it detects the address is Online on both its L1 and L2 ports if the address Global Select bit is set.	r	L2ClashBit
		7	Byte Owned	Must be intentionally set by application (SLP) software when the corresponding data register is to be partially owned by the system. Low byte of data register remains as input, but SLP can now write into high byte for output functions.	w	HighByteBit
		8 to 15	-	General purpose use in application (SLP) software.	w	

3.3.3 Module Series Resistance Register

The Module Series Resistance register can be accessed at the address in the Data Point Table that corresponds to the Module Address + 512. This register contains the L1 fieldbus loop resistance to the connected module at the specified address.

Address 513 - 767 (201h - 2FFh) Module Series Resistance Register						
Address		**Bit**	**Name**	**Description**	**r/w**	**SLP Tag**
Dec.	**Hex.**					
513 to 767	201h to 2FFh	0 to 15	Module Series Resistance	Measured series resistance of the fieldbus loop between the iMAC Controller and the closest Module on the fieldbus with the corresponding address. Values are one to one, e.g. a reading of 00FFh (255) corresponds to a loop resistance of 255Ω Range: 0..1000Ω	r	User Defined

3.3.4 Module Error Register

The Module Error register can be accessed at the address in the Data Point Table that corresponds to the Module Address + 768. This register contains two counters that monitor the number of data scans that occur with an 'Offline Error' or 'Clash Error' present at this address. The corresponding counter increments each time there is either an Offline or Clash error detected on the L1 Fieldbus. Once the counter reaches 255 it will auto-roll over to 0 on the next count.

Address 769 - 1023 (301h - 3FFh) Module Error Register						
Address		**Bit**	**Name**	**Description**	**r/w**	**SLP Tag**
Dec.	**Hex.**					
769 to 1023	301h to 3FFh	0 to 7	Offline Count	This count is incremented every data scan that there is no owner of the module address for which there was once an owner.	r	User Defined
		8 to 15	Clash Count	This count is incremented every time that a data scan occurs when L1 Clash = 1 (ie increments each time an address is scanned for which there are two modules with the same address that are transmitting different input data).	r	User Defined

3.4 System Runtime Segment

The System Runtime Registers are located in address range 1024 - 1151 (400h – 47Eh). The first 20 runtime registers are defined below, the remaining registers are reserved for system use and should not be accessed by the application (SLP) software.

Address 1024 - 1151 (400h – 47Eh) System Runtime Registers				
Address		Register Content	Description	SLP Tag
Dec.	Hex.			
1024	400h	System Control	Relay Control, Digital Input Status, Sequence Control (see individual bit definitions)	SystemControl
1025	401h	System ID and LED control	Rotary Switch and DIP Switch status, L1/L2 LED control (see individual bit definitions)	SysIdLeds
1026	402h	Reserved – Do not use	-	
1027	403h	Reserved – Do not use	-	
1028	404h	EOL Serial Number	EOL Module Serial Number	EolSerNum
1029	405h	Priority Scan Counter	Exception Scan Counter and Rollcall Scan Counter	L1PriorityScans
1030	406h	L1 Data Block Just Complete	Contains block number of last successfully scanned block number.	L1BlockJustDone
1031	407h	Reserved – Do not use	-	
1032	408h	Application (SLP) Software Loop Timer Counter	SLP loop time in milliseconds	LoopTimeSLP
1033	409h	Module Rollcall Control	Used for roll calling and programming iMAC modules (see individual register definitions)	RollcallControl
1034	40Ah	Rollcall Address		RollcallAddress
1035	40Bh	Rollcall Serial Number		RollcallSerNum
1036	40Ch	Rollcall Type Register		RollcallModType
1037	40Dh	Rollcall Block Number		RollcallBlckNum
1038	40Eh	Rollcall Parameter 1		RollcallParam1
1039	40Fh	Rollcall Parameter 2		RollcallParam2
1040	410h	Rollcall Parameter 3		RollcallParam3
1041	411h	Rollcall Parameter 4		RollcallParam4
1042	412h	Remote Key Press	Valid Key Press codes will mimic the front panel keystrokes.	KeyBoardOut
1043	413h	Message System Page Control	Contains page number of currently shown LCD display.	PageTransfer
1044	414h	Reserved – Do not use	-	
1045	415h	EOL Offline Count	Incremental counter for the number of scans the EOL module has been detected as offline.	EOLS_OC
1046	416h	EOL Clash Count	Incremental counter for the number of scans the EOL module has been detected as clashed.	EOLS_Clash

3.4.1 System Control Register

The bits within the System Control register, Address 1024 (400h), have the following definitions:

Address 1024 (400h) System Control Register						
Address		**Bit**	**Name**	**Description**	**r/w**	**SLP Tag**
Dec.	**Hex.**					
1024	400h	0	Start-up bit	Set by controller on first SLP loop following power up	r	
		1	Always 1	This bit is always set to 1	r	
		2	Down bit from L2	Sequence Down Bit from Upstream iMAC Controller	r	SeqDownFromL2
		3	Up bit from L1	Sequence Up Bit from Downstream iMAC Controller/SQM module.	r	SeqUpFromL1
		4	Input 1	iMAC Controller SW1 status: 1 = closed, 0 = open.	r	SwInput1
		5	Input 2	iMAC Controller SW2 status: 1 = closed, 0 = open.	r	SwInput2
		6	Input 3	iMAC Controller SW3 status: 1 = closed, 0 = open.	r	SwInput3
		7	Reserved	System Use Only	r	
		8	Assert Control Relay	Set by application (SLP) software to attempt to energise the CR relay. CR will energise if and only if EOL Module comms is also healthy. Clear to de-energise CR relay.	w	AsrtCtrlRelay
		9	Assert Auxiliary Relay	Set by application (SLP) software to energise AR relay. Clear to de-energise AR relay.	w	AsrtAuxRelay
		10	Assert Up bit on L2	Upstream sequence control bit	r/w	SeqUpOnL2
		11	Assert Down bit on L1	Downstream sequence control bit	r/w	SeqDownOnL1
		12	Enable Modbus Flash Write	Enables data to be written to the iMAC Controller's flash memory using Modbus write commands	r/w	EnFlashWrite
		13	Enable Full System Menu	The following menu functions are disabled when FullSysMenu = 0 & enabled when FullSysMenu = 1 • System Menu 'CLR DPT' item • Maintenance page F1 - Clear individual OnScan Bit	r/w	FullSysMenu
		14	Enable Min System Menu	The following menu functions are disabled when MinSysMenu = 0 & enabled when MinSysMenu = 1 • System Menu (SHIFT+MENU) • Debug Menu • Maintenance page F4 – Clear individual OFFLINE/CLASH COUNTERS	r/w	MinSysMenu

		15	Disable Modbus Master Write Commands to iMAC Slave	Prevents Modbus Master devices writing to iMAC Controller Modbus registers	r/w	

3.4.2 System ID and LED Control Register

The bits within the System ID and LED Control register, Address 1025 (401h), have the following definitions:

Address 1025 (401h) System ID and LED Control Register						
Address		**Bit**	**Name**	**Description**	**r/w**	**SLP Tag**
Dec.	**Hex.**					
1025	401h	0	Rotary Switch Bit0	Binary Coded Decimal (BCD) value of the 16 Position Rotatory Switch on the side of the iMAC Controller. Rotary Switch ID is the index variable in many SLP commands	r	IDSW
		1	Rotary Switch Bit1		r	
		2	Rotary Switch Bit2		r	
		3	Rotary Switch Bit3		r	
		4	DIP Switch 1	State of DIP Switch 1: 1 = ON, 0 = OFF	r	DipSw0
		5	DIP Switch 2	State of DIP Switch 2: 1 = ON, 0 = OFF	r	DipSw1
		6	DIP Switch 3	State of DIP Switch 3: 1 = ON, 0 = OFF	r	DipSw2
		7	DIP Switch 4	State of DIP Switch 4: 1 = ON, 0 = OFF	r	DipSw3
		8	LED L1 ON	Sets the L1 LED to ON	w	Led1On
		9	LED L1 FAST Flash	Sets the L1 LED to a fast flash	w	Led1Fast
		10	LED L1 SLOW Flash	Sets the L1 LED to a slow flash	w	Led1Slow
		11	LED L1 TOGGLE Now	Toggles the state of the L1 LED	w	Led1Togg
		12	LED L2 ON	Sets the L2 LED to ON	w	Led2On
		13	LED L2 FAST Flash	Sets the L2 LED to a fast flash	w	Led2Fast
		14	LED L2 SLOW Flash	Sets the L2 LED to a slow flash	w	Led2Slow
		15	LED L2 TOGGLE Now	Toggles the state of the L2 LED	w	Led2Togg

3.4.3 EOL Serial Number Register

The EOL Serial Number register can be accessed at Address 1028 (404h). This register contains the serial number of the EOL Module connected to the fieldbus.

Address 1028 (404h) EOL Serial Number Register						
Address		**Bit**	**Name**	**Description**	**r/w**	**SLP Tag**
Dec.	**Hex.**					
1028	404h	0 to 15	EOL Serial Number	Contains the serial number of the EOL Module that is connected to the L1 fieldbus.	r	EolSerNum

Note: if EOL clash errors are occurring (e.g. two EOL modules have been inadvertently connected to the L! fieldbus at the same time) the EOL serial number register value will be invalid.

3.4.4 Priority Scan Counter Register

The Priority Scan Counter can be accessed at Address 1029 (405h). This register contains the Exception Scan Counter in the Low Byte and the Rollcall Scan counter in the High Byte.

Address 1029 (405h) Priority Scan Counter Register						
Address		**Bit**	**Name**	**Description**	**r/w**	**SLP Tag**
Dec.	**Hex.**					
1029	405h	0 to 7	Exception Scan Counter	This register stores the counter for the exception scans that were performed on the previous refresh cycle. A refresh cycle in this context refers to the completion of scanning all addresses within the iMAC Data Scans addressable range. This feature is only available in iMAC Controllers with O.S. Firmware V4.2 and greater. Prior to this, the register should always read zero.	r	L1PriorityScans
		8 to 15	Rollcall Scan Counter	The counter described by this register refers to the total number of Rollcall operations that were performed on the bus in the previous refresh cycle. Referring to operations such as Roll Calling, Roll Call Resetting, and Serial Number Reading and Writing. This feature is only available in iMAC Controllers with O.S. Firmware V4.2 and greater. Prior to this, the register should always read zero.	r	

3.4.5 L1 Data Block Just Complete Register

The L1 Data Block Just Complete register can be accessed at Address 1030 (406h). This register contains the block number of the Data Scan that has just been completed on the L1 fieldbus.

Address 1030 (406h) L1 Data Block Just Complete Register						
Address		**Bit**	**Name**	**Description**	**r/w**	**SLP Tag**
Dec.	**Hex.**					
1030	406h	0 to 15	L1 Data Block Just Complete	Contains the block number of the last completed DataScan	r	L1BlockJustDone

3.4.6 Application (SLP) Software Loop Timer Counter Register

The Application (SLP) Software Loop Timer Counter register can be accessed at Address 1032 (408h). This register contains the number of milliseconds that elapses during each loop of the SLP code.

Address 1032 (408h) Application (SLP) Software Loop Timer Counter Register						
Address		**Bit**	**Name**	**Description**	**r/w**	**SLP Tag**
Dec.	**Hex.**					
1032	408h	0 to 15	Application (SLP) Software Loop Timer Counter	Contains the number of milliseconds that the iMAC Controller took to complete the last main loop of the application (SLP) software.	r	LoopTimeSLP

3.4.7 Module Rollcall Registers

Registers in address range 1033 to 1041 (409h to 411h) control the iMAC Controller's Rollcall functions:

Address 1033 to 1041 (409h to 411h) Module Rollcall Registers						
Address		Bit	Name	Description	r/w	SLP Tag
Dec.	Hex.					
1033	409h	0	Reset Rollcall	Before a Rollcall procedure can commence, this bit must be set. The Controller will clear this bit when the Reset Rollcall is complete.	r/w	ResetRollcall
		1	Next Rollcall	Set this bit to initiate a Rollcall for the next serial number. The Controller will clear this bit when the Next Rollcall is complete.	r/w	NextRollcall
		2	Next Rollcall, Force Address	Set this bit to initiate a Forced Next Rollcall for the module with address specified in the Rollcall Address register (40Ah). The Controller will clear this bit when complete.	r/w	NextRollForce
		3	Read Serial Number	Set this bit to initiate a read from the module with serial number, generation number and block number specified in the Rollcall Serial Number register (40Bh), Generation number (409h bits 8 & 9) and the Rollcall Block Number register (40Dh)	r/w	RollcallSerNum
		4	Write Serial Number	Set this bit to initiate a write to the module with serial number, generation number and block number specified in the Rollcall Serial Number register (40Bh) , Generation number (409h bits 8 & 9) and the Rollcall Block Number register (40Dh)	r/w	RollcallWrite
		5	Rollcall Fail	Set by controller following either a Next Rollcall or Next Rollcall Force Address function if it fails. Cleared otherwise.	r/w	RollcallFail
		6	Read SN Fail	Set by controller following a Read Serial Number function if it fails. Cleared otherwise.	r/w	RollReadFail
		7	Write SN Fail	Set by controller following a Write Serial Number function if it fails. Cleared otherwise.	r/w	RollWriteFail
		8,9	Generation ID [9,8]	[11] - Generation 4 [10] - Generation 3 [01] - Generation 2 [00] - Generation 1 These bits are set by controller following a Reset Rollcall or Next Rollcall function. The value of these bits is also used during Read Serial Number and Write Serial Number functions to target the required 'generation number' of the module.	r/w	-
		10 to 15	Rollcall Generation Display Flags	RollcallGen1Flg RollcallGen2Flg RollcallGen3Flg RollcallGen4Flg These bits are set by controller following a Reset Rollcall or Next Rollcall function.	r	RollcallGen1Flg RollcallGen2Flg RollcallGen3Flg RollcallGen4Flg

Address 1033 to 1041 (409h to 411h) Module Rollcall Registers						
Address		Bit	Name	Description	r/w	SLP Tag
Dec.	Hex.					
1034	40Ah	0 to 15	Rollcall Address	This register contains the address of the module that has been rollcalled. Alternatively, this register is used to specify the address of the module for a forced rollcall function.	r/w	RollcallAddress
1035	40Bh	0 to 15	Rollcall Serial Number	This register contains the serial number for the module that has been rollcalled. Alternatively, this register is used to specify the serial number for read/write serial number functions.	r/w	RollcallSerNum
1036	40Ch	0 to 15	Rollcall Type	This register contains the module type number (0..63) of the module that has been rollcalled.	r	RollcallModType
1037	40Dh	0 to 15	Rollcall Block Number	This register contains the block number of the module that has been rollcalled. Alternatively, this register is used to specify the block number for read/write serial number functions.	r/w	RollcallBlckNum
1038	40Eh	0 to 15	Rollcall Parameter 1	This register contains Parameter 1 for the module that has been rollcalled. Alternatively, this register is used to specify Parameter 1 for read/write serial number functions.	r/w	RollcallParam1
1039	40Fh	0 to 15	Rollcall Parameter 2	This register contains Parameter 2 for the module that has been rollcalled. Alternatively, this register is used to specify Parameter 2 for read/write serial number functions.	r/w	RollcallParam2
1040	410h	0 to 15	Rollcall Parameter 3	This register contains Parameter 3 for the module that has been rollcalled. Alternatively, this register is used to specify Parameter 3 for read/write serial number functions.	r/w	RollcallParam3
1041	411h	0 to 15	Rollcall Parameter 4	This register contains Parameter 4 for the module that has been rollcalled. Alternatively, this register is used to specify Parameter 4 for read/write serial number functions.	r/w	RollcallParam4

Rollcall Procedure:

1. Assert the “Reset Rollcall” bit (409h: Bit 0) and wait for the bit to be cleared by the system.
2. Read the Rollcall Control Word.
3. Assert the “Next Rollcall” bit (409h: Bit 1), preserving bits 8 to 15 you read in the previous step, and wait for the bit to be cleared by the system.
4. Confirm that the “Rollcall Fail” (409h: Bit 5) bit is not set.
5. Module data will now be available in registers 1033 to 1041 (409h to 411h).
6. The next module can be rollcalled by repeating steps 2 to 5. This process can be repeated until the Serial Number register (40Bh) reads as 0, indicating that all modules have been rollcalled. Modules will rollcall in order of address followed by serial number, from highest to lowest.

Rollcall, Force Address Procedure:

1. Ensure that the Rollcall Address register (40Ah) is set up as per the address that is to be rollcalled.
2. Assert the “Reset Rollcall” bit (409h: Bit 0) and wait for the bit to be cleared by the system.
3. Read the Rollcall Control Word.

4. Assert the “Next Rollcall, Force Address” bit (409h: Bit 2), preserving bits 8 to 15 you read in the previous step, and wait for the bit to be cleared by the system.
5. Confirm that the “Rollcall Fail” (409h: Bit 5) bit is not set.
6. Confirm that the Rollcall Serial Number register (40Bh) contains a serial number. If this register reads as 0, then there is no module with address specified by the Rollcall Address register (40Ah).
7. If a module exists with the address specified by the Rollcall Address register, this module’s data will be available in registers 1034 to 1041 (40Ah to 411h).
8. Repeat Steps 3 to 5 until the Rollcall Serial Number register (40Bh) reads as 0. This will ensure that all modules programmed to this address are read. Modules with the same address will be read in order of serial number, from highest to lowest.

Read Serial Number Procedure:

1. Ensure that the Rollcall Serial Number register (40Bh), the Generation ID (409h bits 8 & 9) and the Rollcall Block Number register (40Dh) are set up as per the module that is to be read.
2. Assert the “Read Serial Number” bit (409h: Bit 3), preserving Generation ID bits 8 & 9 setup in step 1, and wait for the bit to be cleared by the system
3. Confirm that the “Read SN Fail” (409h: Bit 6) bit is not set.
4. Module Parameters 1, 2, 3 and 4 will now be available in registers 1038 to 1041 (40Eh to 411h).

Write Serial Number Procedure:

1. Ensure that the Rollcall Serial Number register (40Bh), the Generation ID (409h bits 8 & 9) and the Rollcall Block Number register (40Dh) are set up as per the module that is to be written.
2. Ensure the parameters that are to be written to the module are set up in registers 1038 to 1041 (40Eh to 410h).
3. Assert the “Write Serial Number” bit (409h: Bit 4), preserving Generation ID bits 8 & 9 setup in step 1, and wait for the bit to be cleared by the system.
4. Confirm that the “Write SN Fail” (409h: Bit 7) bit is not set.
5. Module Parameters 1,2,3, and 4 will now have been successfully written to the module from registers 1038 to 1041 (40Eh to 411h).

The below table provides the modules that are associated with the possible values of the “Rollcall Type” register located at register address 1036 (40Ch).

Module Type Table Corresponding Values for Rollcall Type Register (Address 1036, 40Ch)					
Value		Module Type	Value		Module Type
Decimal	Hexadecimal		Decimal	Hexadecimal	
0	0000h	Reserved	32	0020h	LED4 Module
1	0001h	Controller	33	0021h	EMM Module
2	0002h	TCD2 DIPSwitch	34	0022h	Undefined #34
3	0003h	EOL Module	35	0023h	SIM-P Module
4	0004h	SQM Module	36	0024h	SIM-T Module
5	0005h	DI2/4 Module	37	0025h	SIM-G Module
6	0006h	IIM-OLC Module	38	0026h	DI5 Module
7	0007h	LIM Module	39	0027h	RO4 Module
8	0008h	TCD4 Long	40	0028h	TO4 Module
9	0009h	TCD4 Module	41	0029h	GCA Flags
10	000Ah	RTD3 Flags	42	002Ah	GCA 15Min Tally
11	000Bh	RTD3 Temp 1	43	002Bh	GCA 8Hr Tally
12	000Ch	RTD3 Temp 2	44	002Ch	GCA 24Hr Tally
13	000Dh	RTD3 Temp 3	45	002Dh	GCA Raw Count
14	000Eh	DI4L Module	46	002Eh	DI8 Module
15	000Fh	DI4 Module	47	002Fh	RIS Module
16	0010h	IIM Module	48	0030h	AIM Flags
17	0011h	PGM-A Programr	49	0031h	AIM Analog
18	0012h	MEOL Module	50	0032h	AIM PwrSupply
19	0013h	Undefined #19	51	0033h	CRM Module
20	0014h	SSW Flags	52	0034h	ARM Module
21	0015h	SSW Control	53	0035h	GRM Module
22	0016h	SSW % Slip	54	0036h	RTD1 Flags
23	0017h	SSW % Speed	55	0037h	RTD1 Temp
24	0018h	SSW Linr Speed	56	0038h	SIM-G2 Module
25	0019h	Undefined #25	57	0039h	FCP DigInputs
26	001Ah	Undefined #26	58	003Ah	FCP DigOutputs
27	001Bh	GAI3 Flags	59	003Bh	FCP AnaInputs
28	001Ch	GAI3 Analogue #1	60	003Ch	FCP AnaOutputs
29	001Dh	GAI3 Analogue #2	61	003Dh	GG2 Flags
30	001Eh	GAI3 Analogue #3	62	003Eh	GG2 Analog
31	001Fh	RKM Keypad	63	003Fh	GG2 PwrSupply

3.4.8 Remote Key Press Register

The Remote Key Press register can be accessed at Address1042 (412h). Valid key press codes, written to this register will mimic an actual iMAC Controller keypad press.

Address 1042 (412h) Remote Key Press Register			
Value		Mimicked Key	SLP Tag
Decimal	Hexadecimal		
0	0000h	No Key	
1	0001h	ESC Key	ESC_KEY
2	0002h	Alarm Key	ALARM_KEY
3	0003h	Menu Key	MENU_KEY
5	0005h	F1 Key	F1_KEY
6	0006h	F2 Key	F2_KEY
7	0007h	F3 Key	F3_KEY
8	0008h	F4 Key	F4_KEY
9	0009h	Left Arrow Key	LEFT_KEY
10	000Ah	Up Arrow Key	UP_KEY
11	000Bh	Right Arrow Key	RIGHT_KEY
12	000Ch	Down Arrow Key	DOWN_KEY
13	000Dh	Enter Key	ENTER_KEY
16	0010h	Shift Mode Key	SHIFT_MOD

To affect a remote keypress, write the value of the desired key press into this register. To affect a shift + key press, add the value of the keypress modifier to the desire key value eg to affect a [SHIFT]+[MENU] keypress write the value 13h (10h+03h) into this register.

3.4.9 Message System Page Control Register

The Message System Page Control register can be accessed at Address 1043 (413h). This register contains the page number of the iMAC Controller's LCD Display.

Address 1043 (413h) Message System Page Control Register						
Address		Bit	Name	Description	r/w	SLP Tag
Dec.	Hex.					
1043	413h	0 to 15	Message System Page Control	Contains the page number of the current page that is being shown on the display of the iMAC Controller	r/w	PageTransfer

3.4.10 Block Request Register

The Block Request register SHOULD NOT be accessed directly by the user. It can be accessed indirectly using the application (SLP) software function ScanAddress{ }. This function can be used to prioritise address scanning and should be used only by Ampcontrol Applications engineers. Incorrect use of this function can disrupt normal operation.

Address 1044 (414h) Block Request Register						
Address		Bit	Name	Description	r/w	SLP Tag
Dec.	Hex.					
1044	414h	0 to 15	Block Request	Contains the block number for a requested data scan.	r/w	-

3.4.11 EOL Offline Count Register

The EOL Offline Count register can be accessed at Address 1045 (415h). This register contains a counter that monitors the number of data scans that occur with an 'EOL Offline Error'. When this counter reaches its maximum value of FFFFh, it will wrap back around to zero on the next count.

Address 1045 (415h) EOL Offline Count Register						
Address		Bit	Name	Description	r/w	SLP Tag
Dec.	Hex.					
1045	415h	0 to 15	EOL Offline Count	This count is incremented every time that a data scan occurs when an EOL module is not detected on the L1 fieldbus.	r	EOLS_OC

3.4.12 EOL Clash Count Register

The EOL Clash Count register can be accessed at Address 1046 (416h). This register contains a counter that monitors the number of data scans that occur with an 'EOL Clash Error'. When this counter reaches its maximum value of FFFFh, it will wrap back around to zero.

Address 1046 (416h) EOL Clash Count Register						
Address		Bit	Name	Description	r/w	SLP Tag
Dec.	Hex.					
1046	416h	0 to 15	EOL Clash Count	This count is incremented every time that a data scan occurs when multiple EOL modules are detected on the L1 fieldbus.	r	EOLS_Clash

3.5 General Buffer Segments

The Data Point Table has two separate blocks of register addresses that are purposed as “General Buffers”. These two blocks have register Address ranges from 1152 to 1279 (480h to 4FFh) and from 1408 to 1535 (580h to 5FFh). These registers can be used as general purpose memory for the application (SLP) software and/or for serial communication data exchange.

If the iMAC Controller’s serial communication protocol is set to Ampcontrol IP2, the General Buffer registers are the only addresses that are able to be accessed using this protocol. Application (SLP) software must move all data that is required to be exchanged using IP or IP2 protocols into these registers to enable access.

3.6 Non-Volatile Memory (NVM) System Variables Segment

The Data Point Table has a block of addresses that are set aside for the iMAC Controller's non-volatile system variables. This block is located at Address 1280 to 1311 (500h to 51Fh). This information is saved in the iMAC Controller's non-volatile flash memory, meaning data is stored and retained during a power outage.

The first 20 registers of the NVM data block have special functions. These registers are located at address range 1280 to 1299 (500h to 513h).

Address 1280 to 1299 (500h to 513h) NVM System Variables					
Address		Name	Description	r/w	SLP Tag
Dec.	Hex.				
1280:LSB	500h:LSB	Protocol	Serial Communications Port Settings. Set using the iMAC Controller Serial Communications Page.	w	
1280:MSB	500h:MSB	BAUD Rate		w	
1281:LSB	501h:LSB	Parity		w	
1281:MSB	501h:MSB	Stop Bits		w	
1282:LSB	502h:LSB	Mode		w	
1282:MSB	502h:MSB	Modbus Slave Address		w	
1283	503h	Signal ZERO Period	L1 Fieldbus communication speed settings. The application (SLP) software function: LineSpeed{ } sets these registers.	r	
1284	504h	Signal EXTEND Period		r	
1285	505h	Signal SYNC Period		r	
1286	506h	Power ZERO Period		r	
1287	507h	Power ONE Period		r	
1288	508h	Power SYNC Period		r	
1289	509h	Not Used / System Use Only			
1290	50Ah	SLP Timer Time Base.	Sets application (SLP) software timer tick rate. The application (SLP) software function: TimerSetup{ } sets this register: 1=10ms, 10=100ms, 50=500ms 100=1second etc.	w	
1291	50Bh	SLP Max Number of Active Timers.	Sets maximum number of timers that the application (SLP) software can utilise. The application (SLP) software function: TimerSetup{ } sets this register: Min = 4, Max = 16.	w	
1292	50Ch	SLP Fault Register.	Application (SLP) Software Error register (see bit definitions)	w	
1293	50Dh	SLP Fault Address.	Memory address where application software fault occurred.	w	
1294	50Eh	Not Used / System Use Only			
1295	50Fh	Not Used / System Use Only			
1296	510h	Series VI Offset	Factory Use Only	-	

Address 1280 to 1299 (500h to 513h) NVM System Variables					
Address		Name	Description	r/w	SLP Tag
Dec.	Hex.				
1297	511h	Series OHMS Offset	This value is subtracted from the calculated Series Resistance. Factory Use Only	w	
1298	512h	Shunt VI Offset	Factory Use Only	-	
1299	513h	Shunt OHMS Offset	This value is subtracted from the calculated Shunt Resistance Factory Use Only	w	

3.6.1 Serial Communication Setup Registers

The Serial Communication setup registers are used to configure the RS232/422/485 port settings. These registers are broken up into two bytes: most significant byte and least significant byte (MSB, LSB); with each byte being associated with a different setting as follows:

Address 1280:LSB (500h:LSB) Protocol Setting		
Byte Value		Corresponding Setting
Dec.	Hex.	
0	0h	Not Configured
1	1h	Modbus Master
2	2h	Modbus Slave
3	3h	IP2 Protocol
4	4h	L1 Maintenance
5	5h	L2 Maintenance

Address 1280:MSB (500h:MSB) BAUD Rate Setting		
Byte Value		Corresponding Setting
Dec.	Hex.	
0	0h	(9600)
1	1h	Not Configured
2	2h	600
3	3h	1200
4	4h	2400
5	5h	4800
6	6h	9600
7	7h	19200*

Address 1281:LSB (501h:LSB) Parity Setting		
Byte Value		Corresponding Setting
Dec.	Hex.	
0	0h	(Even)
1	1h	None
2	2h	Even
3	3h	Odd

Address 1281:MSB (501h:MSB) Stop Bits Setting		
Byte Value		Corresponding Setting
Dec.	Hex.	
0	0h	(One)
1	1h	One
2	2h	Two

Address 1282:LSB (502h:LSB) Mode Setting		
Byte Value		Corresponding Setting
Dec.	Hex.	
0	0h	RS232
1	1h	RS485/RS422

Address 1282:MSB (502h:MSB) Modbus Slave Setting		
Byte Value		Corresponding Setting
Dec.	Hex.	
1..32	1h..20h	Slave Address

*Note: May not be compatible with all third party devices, if communication errors occur using this baud rate, revert to 9600 baud

3.6.2 SLP Fault Register

The SLP Fault register can be accessed at Address 1092 (50Ch). This register contains a number of bits that indicate the status of application (SLP) software logic engine.

Address 1092 (50Ch) System Error Register						
Address		**Bit**	**Name**	**Description**	**r/w**	**SLP Tag**
Dec.	**Hex.**					
1092	50Ch	0	Fatal SLP Fault		w	FatalSLPFault
		1	Stack Overflow		w	StackOverFlow
		2	Stack Underflow		w	StackUnderFlow
		3	Address out of Bounds		w	AddressBounds
		4	Illegal Command		w	IllegCommand
		5	Timer Parameter Fault Base/Numb		w	TimeBaseFault
		6	Not Used / System Use Only			
		7	Not Used / System Use Only			
		8	TON No Timer Slots Delay		r	TonDelayWarn
		9	TOFF No Timer Slots Delay		r	ToffDelayWarn
		10	L1 Power Period Overflow		r	L1PwrPerOver
		11	L2 Power Period Overflow		r	L2PwrPerOver
		12	L2 Loss of SYNC		r	L2SyncLoss
		13	Down Active		r	
		14	Down Finished		r	
		15	Not Used / System Use Only			

Note: Bits 0 to 7 represent critical SLP faults. These are not expected during normal operation and should be reported to an Ampcontrol applications engineer. Bits 8 to 15 are warnings and can be expected to occur from time to time.

3.7 User Non-Volatile Memory (NVM) Segment

The Data Point Table has a block of addresses that are set aside for non-volatile user memory. This block is located at Address 1312 to 1407 (520h to 57Fh). This allows the user to save information that will be retained in the iMAC Controllers Flash memory at power down.