GROUP ENGINEERING SERVICES : EXETER

SPA2 - Software Design Description

John Wilkes

2 June 2008

ISSUE : 5

PD-5208-9011

This document describes the embedded software that runs on the DSP(s) inside the SPA2 family of products.

INTRODUCTION

The software described in this document provides the functionality required by the SPA2 family of products. The family includes SPA2, SPA2-2, SPAlite, Aries SPA, and the Gyro PCI card. The code is required to take either velocity or torque demands from a higher level controller, via a real time serial interface, and generate the correct demand signals to the power stages of the amplifier. Servo set-up is done entirely through the serial interface by a suitable host (either a PC or the higher level controller). The code runs on a TMS320F2812 DSP supplied by Texas Instruments (most products contain multiple instances of the processor). This is a 32 bit fixed point processor that runs at 150 MIPS. The source code is written mostly in C and is split into a number of source files as shown in Figure 1. Complete list of files used is given in Appendix D.

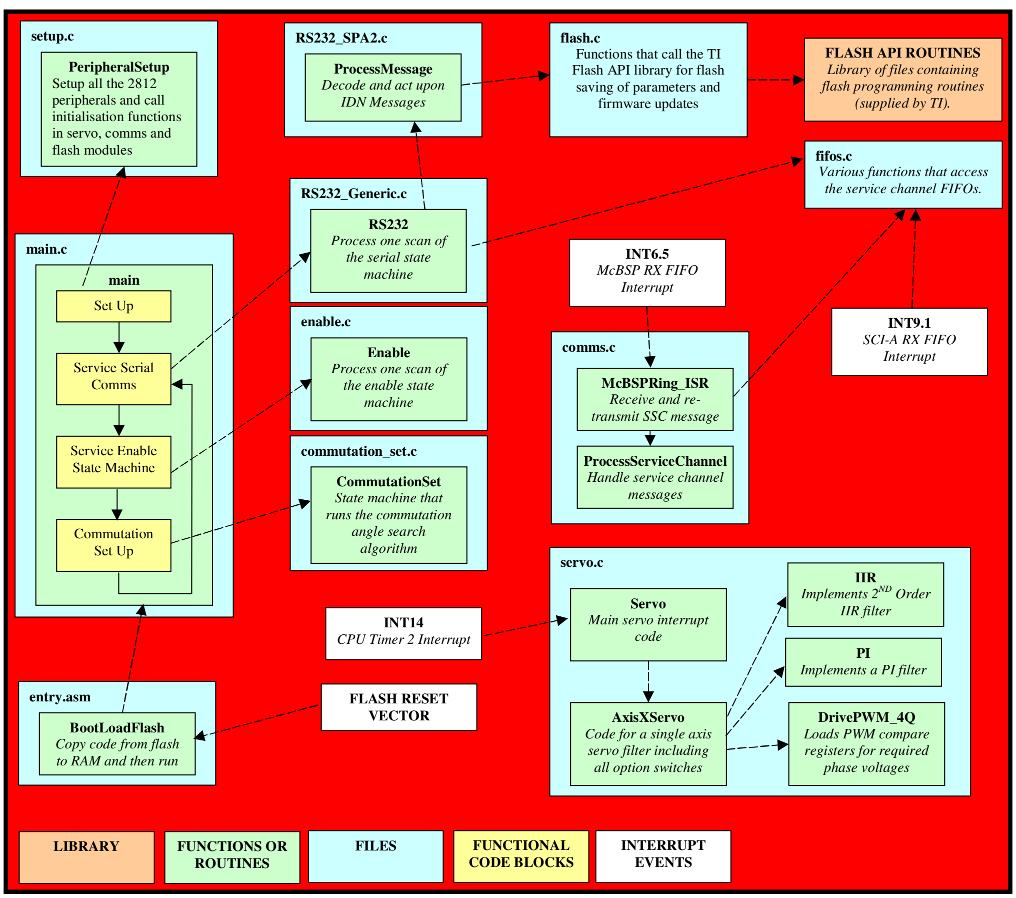

Figure 1 – Overall Code Structure

Although the code that runs is identical regardless of the host hardware (i.e. which variant of the SPA2 product family), the behaviour will vary slightly to take into account slight differences between the products. Each DSP is responsible for up to three axes of motor control and the nomenclature used in each hardware variant to refer to the DSPs and the axes under control is slightly different and is shown in Table 1.

Variant	No Of DSPs	First DSP Name	Second DSP Name
SPA2	2	DSP 1 (Axes 1 & 2)	DSP 2 (Axis 3 & 4)
SPA2-2	2	DSP XY (Axes 1 & 2)	DSP ZW (Axes 3 & 4)
SPAlite	2	DSP XY (Axes 1 & 2)	DSP Z (Axes 4)
3 Axis Aries SPA	1	DSP PQR (Axes 1, 2 & 3)	-
5 Axis Aries SPA	2	DSP PQR (Axes 1, 2 & 3)	DSP ST (Axes 4 & 5)
Gyro PCI	1	DSP (Axes D and E)	-

Table 1 – Nomenclature Used To Refer To DSPs

For the remainder of this document, all variants of the hardware are referred to as SPA2. Unless indicated otherwise, the descriptions apply to all products of the SPA2 family.

The main part of this document is divided into four sections: Section one describes the start up and set up procedure that initialises the code and the on chip DSP peripherals. Section two describes the serial communications to the SPA2 which include synchronous servo demands and asynchronous set up commands. Section three defines the enable state machine and the various amplifier faults that can occur. Section four describes the servo code. There are also a number of appendices that contain more detailed information of various aspects of the design: Appendix A has flowcharts that describe how some of the code modules work. Appendix B contains a memory map that defines how the DSP memory space is used. Appendix C is a comprehensive list of the messages that can be sent across the asynchronous communication interface. Appendix D contains a list of the source files used. Finally, Appendix E briefly describes some supporting code that runs on a separate module that allows the SPA2 communicate with a standard PC.

This document describes the operation of the SPA2 when loaded with firmware version 05.00.00.

References

[1] SPA2/SPAlite – Hardware Design Description. John Wilkes and Lyn Yarwood. Issue 10. 26 July 2005. SPA2 HDD 10.doc. (For SPA2)
[2] SPA2/SPAlite – Hardware Design Description. John Wilkes and Lyn Yarwood. Issue 11. 12 June 2006. SPA2 HDD 11.doc. (For SPA2-2)
[3] Aries SPA – Hardware Design Description. Jon Fuge. Issue 01. 7 June 2007. PD-5504-9011. PD-5504-9011-01 Aries HDD.doc
[4] TMS320F2810, TMS320F2812 Digital Signal Processors. Data Manual. SPRS174O. April 2001 – revised July 2007. Texas Instruments.
[5] TMS320F28x DSP System Control and Interrupts Reference Guide. Texas Instruments. SPRU078E. April 2002 – Revised March 2008.
[6] TMS320F28x DSP External Interface Reference Guide. Texas Instruments. SPRU067C. May 2002 Revised November 2004.
[7] TMS320F28x DSP Event Manager (EV) Reference Guide. Texas Instruments. SPRU065E. November 2004 – Revised June 2007.
[8] TMS320F28x DSP Multichannel Buffered Serial Port (McBSP) Reference Guide. Texas Instruments. SPRU061C. May 2003 – Revised November 2007.
[9] TMS320F28xx, 28xxx DSP Serial Peripheral Interface (SPI) Reference Guide. Texas Instruments. SPRU059D. June 2002 – Revised November 2006.
[10] TMS320F28x Analog-to-Digital Converter (ADC) Reference Guide. Texas Instruments. SPRU060D. June 2002 – Revised July 2005.
[11] TMS320F28xx, 28xxx DSP Serial Communication Interface (SCI) Reference Guide. Texas Instruments. SPRU051B. November 2004.
[12] TMS320F28x DSP Boot ROM Reference Guide. Texas Instruments. SPRU095C. May 2003 – Revised December 2006.

[13] Serial link for real-time communication between controls and drives – Part 1: General Requirements. Electrical equipment of industrial machines. TC 44/WG 5N06 Revision 8. June 1994.
[14] TMS320F2810, TMS320F2811 and TMS320F2812 Flash API. Texas Instruments. SPRC125 Version 2.1. 5 August 2005.
[15] Determination of Commutation Angle. John Wilkes. Issue 1. 10 June 2005. Commutation Set.doc.
[16] Determination and correction of quadrature fringe measurement errors in interferometers. Peter L. M. Heydemann, Applied Optics, 20, No 19, pp. 3382-3384, 1981.

SECTION 1 – START AND SET UP CODE

The SPA2 code can be compiled to run from flash (normal – release) or from RAM (during development when connected to an emulator – debug). In fact, the majority of the code always runs in RAM but when running in so called flash mode, the code is stored in flash and the start up routine ('entry.asm' – also in flash) copies most of the code to RAM where it will run faster. To start this way, the DSP must be configured for flash boot using the boot mode jumpers – see reference [12]. When running in RAM mode, the code must be loaded directly into RAM and forced to start from the 'main' label using an emulator. The main difference between the two modes is the memory map that is used, see Appendix B (page 47).

The entry point to the code, then, is the 'main' function that is contained in the main.c file. This function acts as supervisory control for the SPA2. The code will call the set up routine and then go into an infinite loop which will service all asynchronous processes one by one. There are three asynchronous processes. The first handles host communications via the service channel embedded in the SSC (Synchronous Serial Communications) while the second handles the enable state of the SPA2 and processes faults and limits. The third asynchronous process implements a method for determining the commutation angle of a brush-less or linear motor. The comment header in main.c contains the version history of the code.

The set up routine 'PeripheralSetup' in setup.c is run once when a processor reset occurs. It is called from the 'main' function. The routine sets up all the on-chip 2812 peripherals that are used by the SPA2 and initialises the servo, communications and flash writing code. Figure 2 shows how the 2812 peripherals are used by the SPA2 and more detail is given in the sections below.

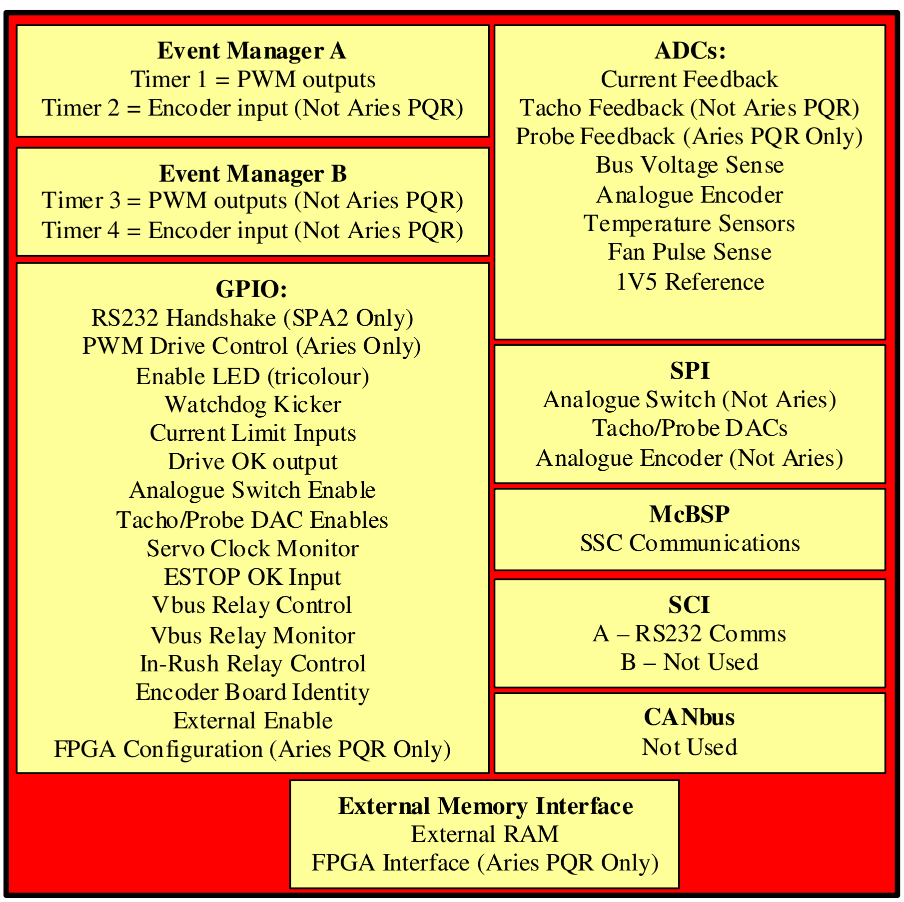

Figure 2 – Use of 2812 Built in Peripherals

General Processor Setup

See references [5] and [6].

- Enable internal watch dog.
- Set PLL multiplier to ×5 (SYSCLKIN = 150MHz).
- Set High Speed and Low Speed clock multipliers to one (both clocks = 150MHz).
- Enable clock modules (EVA, EVB, SPI, McBSP, ADC and SCIA)
- Enable flash pipeline mode and set flash wait states to 5 (paged and random)
- Configure FPGA (Aries PQR Only)
- Set up interrupt vector table and enable interrupts. Following interrupts are used:
  - INT 1.8 – Watchdog (only used during Flash API call to prevent a reset)
  - INT 6.5 – McBSP RX (SSC Comms Interrupt)
  - INT 9.1 – SCI A, RX (RS232 Comms Interrupt)
  - INT 14 – CPU Timer 2 (Servo Interrupt)

GPIO Set up

There are 56 i/o pins available on the 2812, but these are multiplexed with the standard peripheral i/o lines and so not all are available for general purpose IO depending on which peripherals are used, see reference [5] for more details about the GPIO. References [1], [2] & [3] have details of the pins used.

Note that the PWM lines are encoded to specify hardware variant and version, these lines are read prior to the being reconfigured as PWM outputs.

EMIF Set up

See reference [6] for details of the processors External Memory Interface. Zone 6 is used for external RAM and Zone 0 is used to interface to the Aries FPGA. The EMIF is setup up as follows:

For all variants except Aries PQR:

- XTIMCLK bit = 0 (XTIMCLK = SYSCLKOUT = 150MHz)
- CLKMODE bit = 0 (XCLKOUT = XTIMCLK = 150MHz)
- CLKOFF bit = 1 (don't put clock to external pin)
- Zone 6 wait states: Read = 1 lead, 2 active, 1 trail; Write = 1 lead, zero write and zero trail

For Aries PQR

- XTIMCLK bit = 1 (XTIMCLK = SYSCLKOUT / 2 = 75MHz)
- CLKMODE bit = 1 (XCLKOUT = XTIMCLK / 2 = 37.5MHz)
- CLKOFF bit = 0 (put clock to external pin to clock FPGA)
- Zone 6 wait states: Read = 1 lead, 1 active, 0 trail; Write = 1 lead, zero write and zero trail
- Zone 0 wait states: Read = 1 lead, 1 active, 0 trail; Write = 1 lead, zero write and zero trail

ADC Set Up

See reference [10] for details of the DSP's on board ADCs. The ADC inputs are connected differently depending upon the hardware variant, see References [1], [2] & [3] for details. There are fifteen inputs to convert (sixteen for Aries PQR) and they are implemented using a cascaded sequence in the order as shown in Table 2. The sequence is run in start/stop mode and is started in software by the servo interrupt routine. The sequence does not generate any interrupts.

Some inputs are not used in all of the hardware variants (e.g. Phase B Current Feedback in SPAlite or Aries ST). These are connected to the 1.5V reference in hardware and are set zero in the software.

	All Variants Except Aries PQR		Aries PQR	
Conversion	Channel	What	Channel	What
1	A7	1.5V reference	A7	1.5V reference
2	A0	Current feedback (axis 1, phase A)	A0	Current feedback (axis 1)
3	A1	Current feedback (axis 1, phase B)	A5	RLS tacho cosine (axis 1)
4	A2	Tacho feedback (axis 1)	A2	Scanning probe (X Axis)
5	A4	Encoder Sine (axis 1)	A4	RLS tacho sine (axis 1)
6	A5	Encoder Cosine (axis 1)	B0	Current feedback (axis 2)
7	B0	Current feedback (axis 2, phase A)	B5	RLS tacho cosine (axis 2)
8	B1	Current feedback (axis 2, phase B)	B2	Scanning probe (Y Axis)
9	B2	Tacho feedback (axis 2)	B4	RLS tacho sine (axis 2)
10	B4	Encoder sine (axis 2)	A1	Current feedback (axis 3)
11	B5	Encoder cosine (axis 2)	B3	RLS tacho cosine (axis 3)
12	A3	Bus voltage sense (axis 1)	B1	Scanning probe (Z Axis)
13	B3	Bus voltage sense (axis 2)	A6	RLS tacho sine (axis 3)
14	B6	Temperature	A3	Bus voltage sense
15	B7	Fan pulse sense / Heater temperature	B6	Temperature
16	-	-	B7	Fan pulse sense

Table 2 – ADC Convert Sequence

Timer (Event Manager) Set Up

See reference [7] for details about the DSP's event managers.

Timer 1/3 –PWM Outputs

For non-Aries variants, full control of the H-bridge FETs is provided using the compare units associated with Timers 1 and 3 (for Axes 1 and 2 respectively). For Aries, the bridge control is performed in hardware with a Siliconix device and the DSP only has to generate a single PWM signal per axis together with enable and brake signals that are implemented with GPIO. The compare units associated with Timer 1 are sufficient to supply PWM to all three axes and Timer 3 is not used. Set up of the timers is, however, common to all variants (with the exception of dead band control) and is specified below:

- No interrupts generated
- 28x Enhancements switched off
- Timer runs free of EMU halts
- Compare output disabled (use PWM compare outputs)
- Timer does not start ADC
- Use internal clock (High Speed clock with x1 pre-scaler)
- Continuous count up mode
- Timer period set from flash (default 97 kHz).
- Dead band timers enabled (Disabled for Aries)
- Dead band pre-scaler set to ÷16
- Dead band timer period set from flash (default 5, = 533ns)
- PWM compare outputs enabled
- Action Control Register set depending upon enable state (see below).
- Action Control Registers reloaded immediately
- Space vector mode disabled
- Compare Registers reloaded on zero

Action Control Register (ACTRx) Set Up – Non Aries

The action control registers define the state and sense of the compare unit outputs that produce the PWM signals. For non Aries variants these registers are used to control the enable/braking state of the drive stages. For this case, Table 3 shows how each of the 12 PWM lines are used.

The value of the action control register will depend on which mode the axis servo is in, as defined in Table 4.

	Axis 1			Axis 2		
Phase	A	B	C	A	B	C
High	PWM1	PWM3	PWM5	PWM7	PWM9	PWM11
Low	PWM2	PWM4	PWM6	PWM8	PWM10	PWM12

Table 3 – PWM Line Usage (Non Aries)

		Brushless 3 Phase Motor		Brushed DC Motor	
FET	DISABLED	ENABLED	BRAKING	ENABLED	BRAKING
A High	Forced High	Active High	Forced High	Active High	Forced High
A Low	Forced High	Active Low	Active Low	Active Low	Active Low
B High	Forced High	Active High	Forced High	Forced High	Forced High
B Low	Forced High	Active Low	Active Low	Forced High	Forced High
C High	Forced High	Active High	Forced High	Active High	Forced High
C Low	Forced High	Active Low	Active Low	Active Low	Active Low
ACTRx	0xFFFF	0x999	0xBBB	0x9F9	0xB9B

Table 4 – Action Control Register Values (Non Aries)

In the case of Aries, the enable/braking state is controlled using GPIO lines and the action control register is permanently set so the active outputs (PWM1, PWM 3 and PWM5) are Active Low (0xDDD).

Timer 2/4 – Axis 1/2 Encoder Inputs

With the exception of Aries PQR, these timers are used to interface to digital quadrature encoders with the following set up:

- No interrupts generated
- 28x Enhancements switched off
- Timer does not start ADC
- Enable the timer
- Clock pre-scale × 1 (ignored anyway)
- Compare output disabled
- Use QEP as clock source
- Run free of EMU Halts
- Directional up/down count mode
- Use timer 2 enable bit and period register
- Counter wraps at full range (0xFFFF)

Aries PQR interfaces to digital quadrature encoders via an on board FPGA, see page 9 for more details.

SCI Set Up

See reference [11] for details about the DSP’s SCI ports. SCI port A is used for RS232 serial communications to a host. SCI port B is not used. The set up of port A is as follows:

- Enable port
- Enable RX and TX FIFOs
- No FIFO Transfer Delay
- Disable TX Interrupt
- Enable RX Interrupt on one FIFO level
- Set IDLE Line (not multiprocessor)
- Disabled Sleep mode
- Disable loop back mode
- Disable Auto Baud Alignment

- Run free of EMU halts
- Word length 8 bits
- Two stop bits *
- Even Parity *
- 115200 baud *

* These values are set from flash at boot up, the default values are given here.

SPI Set Up

See reference [9] for details about the DSP's SPI port. The SPA2 uses the SPI port to communicate with the analogue switches and the tacho/probe DACs in the analogue input circuitry. It is also used to interface to the ADCs of a version 2 analogue encoder board (limited availability option for SPA2) and to configure the on board FPGA for Aries PQR V1.

The analogue switches control analogue gains in the tacho and current feedback. They are set once at start up (as part of the servo initialisation code) and whenever the settings are changed. The chip enable to the switches is controlled with a GPIO pin (D6). At all other times, if a V2 analogue encoder board is NOT installed, the SPI is configured to communicate with the tacho DACs which are part of the high resolution tacho input tracker. New values are written to the DACs every servo cycle (unless a write to the analogue switches is in progress). Communications to the DACs are enabled using GPIO pins B14 and B15 for axes 1 and 2 respectively.

If a V2 analogue encoder board IS installed then, when an analogue switch change is not in progress, the SPI port is set up to read the ADCs on the encoder board. At the start of the servo interrupt, the ADC data is read out, stored and then the SPI port is reconfigured for Tacho DAC use. It remains so for the rest of the servo interrupt then is configured back to ADCs at the end, ready for next time. Communication to the ADCs is enabled using GPIO pin F3.

For Aries PQR V1, the FPGA configuration file is stored in flash and sent to the FPGA during startup via the SPI port. After this, the port is used as normal for the analogue probe DACs (Aries does not have analogue switches or external ADCs).

Table 5 show the configuration of the SPI port for its various uses.

	Analogue Switches	Tacho DACs	Analogue Enc. ADCs	FPGA Conf.
Clock polarity	0 (rising edge)	0 (rising edge)	1 (falling edge)	1 (falling edge)
Clock phase	1 (with delay)	0 (no delay)	1 (with delay)	0 (no delay)
Loop back mode	Disabled	Disabled	Disabled	Disabled
Run Free of EMU Halts	Yes	Yes	Yes	Yes
Data length	8/16 bits*	8 bits	16 bits	16 bits
Interrupts	Disabled	Disabled	Disabled	Disabled
Master/Slave	Master	Master	Master	Master
Transmission Enabled	Yes	Yes	Yes	Yes
Baud rate	2 Mb/s	21.4Mb/s	21.4Mb/s	21.4Mb/s
Transmit FIFO Enabled	No	Yes	Yes	Yes
Receive FIFO enabled	No	No	Yes	No
FIFO TransferDelay	N/A	None	None	None
* Note that prototype boards and SPAlite use 16 bit data length. SPA2 uses 8 bit data length				

Table 5 – SPI Set Up

McBSP Set Up

See reference [8] for details about the DSP's McBSP (Multi-channel Buffered Serial Port) which is used to implement real time synchronous serial communications (SSC) between the host and the multiple DSPs of the SPA2. The SSC has two modes, normal and extended. In normal mode, ten 16-bit words plus a checksum word are transmitted in each message. In extended mode, it is twenty 16-bit words plus a checksum. Note that the McBSP is set up to handle 32-bit words to gain maximum benefit from the FIFOs. The port is normally configured as a slave but can be reconfigured as a master. Described below is the default set up of the McBSP for a slave in normal mode:

- Receive and Transmit FIFOs enabled
- TX Interrupts disabled
- Enable RX Interrupt on 6 FIFO levels
- No FIFO transfer delay
- Right Justify Data
- Run free on EMU halts
- Disable loop back mode
- Disable clock stop mode
- Disable DX enabler
- Disable A-bis mode
- Each frame is a single phase of six words of 32 bits, not companded
- One data bit delay
- Repeated frame syncs ignored
- Sample rate generator clock is free running and derived from input clock
- Sample rate generator frame sync is derived from DXR to XSR transfer
- Transmit clock and frame sync from sample generator
- Receive clock and frame sync from input
- All clock and frame sync polarities 0

When extended mode is detected, the following settings take effect:

- RX Interrupt on 11 FIFO levels
- Frame size of eleven words of 32 bits

When the SPA2 is configured as a master, the following settings take effect (note that when configured as a master, the SPA2 always uses extended mode):

- RX Interrupt on 11 FIFO levels
- Frame size of eleven words of 32 bits
- Sample rate generator clock is derived internally
- Clock rate set to 37.5 Mbps
- RX interrupts disabled

CPU Timers

See reference [5] for details about the DSP's 32 bit CPU timers. CPU Timer 2 is used to generate the servo interrupt. By default the period of the timer is set to 15000 with a ÷1 pre-scale, giving a 100µs servo period. This values can be user configured allowing a custom servo period to be used. Timers 0 and 1 are not used.

Servo Code Set Up

The set up routine makes a call to the 'ServoInit' routine in the 'servo.c' file. This routine checks to see if parameter values have previously been saved to flash memory. If so, they are used to initialise the relevant parameters, see description of P-0-208 (page 76) for a list of these.

In Addition a number of global variables and pointers, some dependant upon hardware type, are initialised 'ServoInit' function.

Aries FPGA

The Aries PQR variant interfaces to an FPGA, via the EMIF, to provide three axes of quadrature encoder input. Each encoder counter can be latched by its own reference mark input and all three can be latched by a TTP (Touch Trigger Probe) input. The registers that control this function are shown in Table 6 and described in the subsequent sections.

For version 1 of the Aries PCB, the FPGA configuration file is stored in DSP flash and sent to the FPGA via SPI when the DSP boots. Subsequent versions of the Aries PCB contain a much bigger FPGA to provide, amongst other things, a PCI express interface to the host computer. Consequently, the configuration file is too big for DSP flash so a separate configuration device is used.

Address	Register
0x2000	Reserved
0x2001	General Purpose Control Register
0x2002	General Purpose Status Register
0x2010	Axis P Count Register
0x2011	Axis P Capture Register
0x2012	Axis P Control Register
0x2013	Axis P Status Register
0x2020	Axis Q Count Register
0x2021	Axis Q Capture Register
0x2022	Axis Q Control Register
0x2023	Axis Q Status Register
0x2020	Axis R Count Register
0x2021	Axis R Capture Register
0x2022	Axis R Control Register
0x2023	Axis R Status Register

Table 6 – Aries FPGA Memory Map

General Purpose Control Register (Address = 0x2001)

This read and write register contains a global reset for the encoder interface and some set up bits for the TTP input. The layout of this register is shown in Table 7

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">Not
Used</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">Not
Used</td><td>TTP Fall
Edge Ena.</td><td>TTP Rise
Edge Ena.</td><td>Clear
Errors</td><td>Reset</td></tr>
</table>

Table 7 – General Purpose Control Register Bit Definitions

Bit 0 Reset
0 – Hold encoder interface circuitry in reset
1 – Release encoder interface circuitry from reset

Bit 1 Clear Errors
0 – Normal operation
1 – Clear all overflow and QEP fault errors

Bit 2 TTP Rising Edge Enable
0 – TTP input rising edge will not latch counters
1 – TTP input rising edge will latch counters

Bit 3 TTP Falling Edge Enable
0 – TTP input falling edge will not latch counters
1 – TTP input falling edge will latch counters

Bit 4-15 Not Used

General Purpose Status Register (Address = 0x2002)

This read only register reports the instantaneous state of the TTP input. The layout of this register is shown in Table 8.

15	14	13	12	11	10	9	8
Not Used							
7	6	5	4	3	2	1	0
Not Used							TTP Input

Table 8 – General Purpose Status Register Bit Definitions

Bit 0 Instantaneous TTP Input
0 – TTP input state is currently 0
1 – TTP input state is currently 1

Bit 1-15 Not Used

Axis n Count Register (Address = 0x20n0)

This read only register contains the current count value of the quadrature input.

Axis n Capture Register (Address = 0x20n1)

This read only register contains the count value of the quadrature input at the last enabled latch event – either this axes reference mark input or the TTP input.

Axis n Control Register (Address = 0x20n2)

This read and write register contains set up bits for this axis of encoder input. The layout of this register is shown in Table 9.

15	14	13	12	11	10	9	8
Not Used		Enable TTP	Enable Ref Marks	Not Used			
7	6	5	4	3	2	1	0
Not Used				Ref Fall Edge Ena.	Ref Rise Edge Ena.	Not Used	

Table 9 – Axis Control Register Bit Definitions

Bit 0-1 Not Used

Bit 2 Reference Mark Rising Edge Enable
0 – Reference mark input rising edge will not latch counters
1 – Reference mark input rising edge will latch counters

Bit 3 Reference Mark Falling Edge Enable
0 – Reference mark input falling edge will not latch counters
1 – Reference input falling edge will latch counters

Bit 4-11 Not Used

Bit 12 Reference Mark Capture Enable
0 – Reference mark input cannot latch counters
1 – Reference mark input can latch counters

Bit 13 TTP Capture Enable
0 – TTP input cannot latch counters
1 – TTP input can latch counters

Bit 14-15 Not Used

Axis n Status Register (Address = 0x20n3)

This read only register contains status bits for this axis of encoder input. The layout of this register is shown in Table 10.

15	14	13	12	11	10	9	8
Not Used		Capture Overflow	Capture Flag	Quadrature Error	Ref Mark Input	Not Used	
7	6	5	4	3	2	1	0
Not Used							

Table 10 – Axis Status Register Bit Definitions

Bit 0-9 Not Used

Bit 10 Instantaneous Reference Mark Input
0 – Reference mark input state is currently 0
1 – Reference mark input state is currently 1

Bit 11 Quadrature Error
0 – No quadrature errors detected
1 – Quadrature errors detected

Bit 12 Capture Flag
0 – A latch event (either reference mark or TTP) has occurred
1 – A latch event (either reference mark or TTP) has not occurred

Bit 13 Capture Overflow
0 – No more than one latch events have occurred without capture register being read
1 – More than one latch events have occurred without capture register being read

Bit 14-15 Not Used

SECTION 2 – SERIAL COMMUNICATIONS

The SPA2 uses the McBSP port of the 2812 DSP to communicate with a bus master (probably a UCC2 CMM controller) and other SPA2 DSPs using a scheme known as Synchronous Serial Communication (SSC). This is configured as a ring around which messages are passed from the bus master to each slave in turn and then back to the master again. The messages can contain a checksum word which can be used to ensure data integrity around the ring.

Two kinds of data transfer have to be considered: Synchronous and Asynchronous data. Synchronous data includes that which is required in order to close the servo loops, i.e. demands and feedback. These data are transmitted at a regular rate. Asynchronous data is everything else and includes things like controller gains, mode settings and even firmware updates. The SSC is a synchronous protocol with an asynchronous protocol (the service channel) embedded within it. This is very loosely based upon the SERCOS standard (see reference [13] and Appendix C). This section describes the details of this scheme.

In the descriptions below:

- a word is considered to be 16 bits long.
- a slave is a single SPA2 DSP (i.e. two slaves make up a four axis SPA2).

Data Requirement

Each axis is assigned 32 bits of input and 32 bits of output data. To make the best use of this, what this data represents can be arbitrarily set.

In addition, a 8 bit control or status byte and a 8 bit service channel byte are assigned to each pair of axes (since normally each DSP will look after 2 axes – page 19 contains details of extensions for the Aries PQR variant, which has to look after three axes). Thus, the packet of data (known as a Two Axis Packet, or TAP) that is sent in each transmission will be five words as shown in Table 11.

Word	Purpose	
1	Axis 1 Demand/Feedback High Word	
2	Axis 1 Demand/Feedback Low Word	
3	Axis 2 Demand/Feedback High Word	
4	Axis 2 Demand/Feedback Low Word	
5	Control / Status Byte	Service Channel Byte

Table 11 – A Two Axis Packet (TAP)

In SPA2 the data received for the first Demand depends upon the servo loop mode as shown in Table 12. The data received for the second demand is specified using P-X-242 (page 92). The user can choose what is contained in the feedback data that is returned to the master by setting the ‘SSC Feedback Mode’ see P-X-238 (page 91).

MODE	Byte 1	Byte 2	Byte 3	Byte 4
Uncommutated Current Loop	Phase 1 Current Demand		Set with P-X-242	
Commutated Current Loop	Main Current Demand		Set with P-X-242	
Velocity Loop	Velocity Demand		Set with P-X-242	

Table 12 – SSC Synchronous Demands Used For SPA2

Message Transmission

The SSC has two modes of operation: normal and extended. In normal mode, each message is two TAPs long and is used to communicate to one or two slaves. Extended mode uses messages that are four TAPs long and can be used to communicate to up to four slaves. Changing between modes is dynamic and is controlled by the master via a bit in the control word. The slaves will receive two TAPs and check the mode bit in one of the control words. If this bit is set, the slave assumes extended mode and receives a further two TAPs. Otherwise, the slave assumes normal mode and receives no more data.

The slave will always take the last TAP received (either the second or the fourth) as its demand and will transmit its feedback TAP first followed by the other TAP or TAPs it received. The slave must also set the extended mode bit of its status byte to match the bit in the received control byte.

Figure 3 shows how this works for a normal mode transmission consisting of one master and two slaves. A message containing two TAPs is transmitted between each node. The master transmits two TAPs: the demand for slave 2 followed by the demand for slave 1. Each slave receives two TAPs, takes the second as its demand and then transmits two TAPs: its feedback followed by the first received TAP. Since there are two slaves in the ring then the two TAPs finally received by the master will be the feedback from slave 2 followed by the feedback from slave 1.

Figure 4 shows an extended mode transmission with a master and four slaves.

Note that if there are fewer slaves than TAPs in the message (i.e. only one slave in normal mode or three or less slaves in extended mode – as indicated in red in Figure 3 and Figure 4), the master will see the demands to the missing slaves echoed in the response it receives and the feedback from the present slaves will be located in the wrong TAPs (as shown in the figures below). The master could use this behaviour to determine how many slaves are present in the ring.

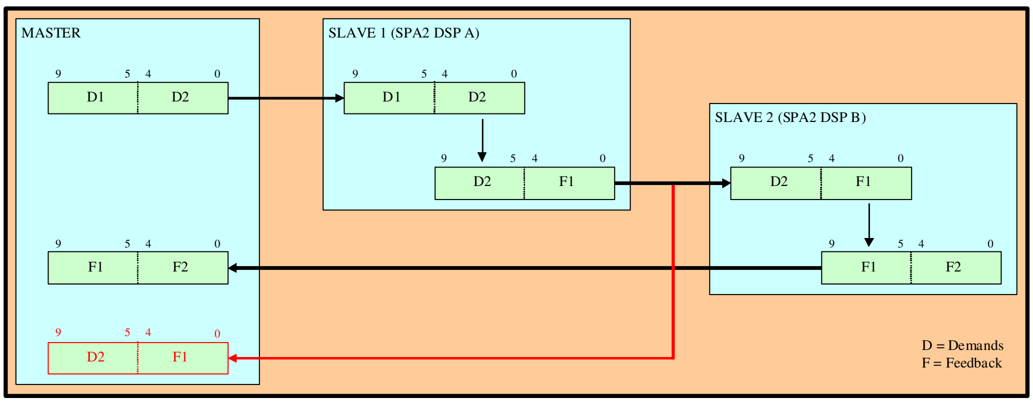

Figure 3 – Normal Mode Data Transmission Around The SSC Ring

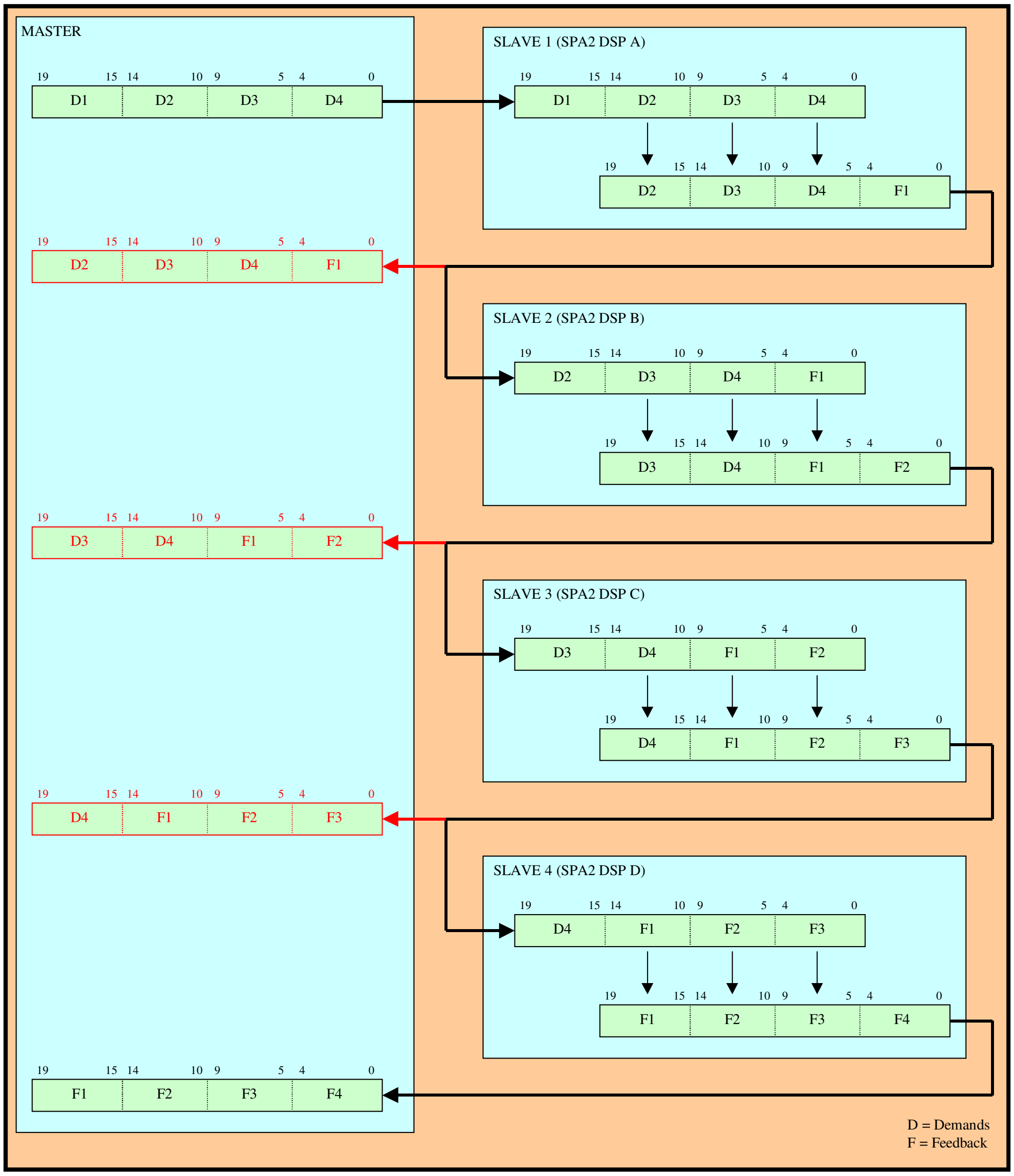

Figure 4 – Extended Mode Data Transmission Around The SSC Ring

Control Byte

The control byte is sent by the master as the first byte of the fifth word of each TAP. Its format is as follows:

7	6	5	4	3	2	1	0
Enable	Not Used				Extended Mode Bit	SC Do Not Send	SC Sync Bit

Table 13 – SSC Control Byte

Bit 7 – Enable
When set this will instruct the slave to enable itself. This acts as a third enable signal along side the software enable input (P-0-206) and the hardware enable input (GPIO B13). All enable inputs must be active (and no faults be present) in order to enable the amplifier. See Section 3 (starting page 21) for a description of the enable state machine.

Bits 3-6 – Not Used

Bit 2 – Extended Mode Bit
This bit, when set, indicates that the message being transmitted is in extended mode. That is, the message is four TAPs long.

Bit 1 – Service Channel Do Not Send Bit
This bit, when set, prohibits the slave from sending data via the service channel.

Bit 0 – Service Channel Sync Bit
This bit will be toggled by the master every time a new Service Channel byte is valid.

Status Byte

A status byte is sent by each slave as the first byte of the fifth word of its TAP. Its format is as follows:

7	6	5	4	3	2	1	0
Enable	Amp OK	ESTOP OK	Latch Event	Not Used	Extended Mode bit	SC Do Not Send	SC Sync Bit

Table 14 – SSC Status Byte

Bit 7 – Amplifier Enabled
When set this will indicate that the amplifier is enabled and ready to accept servo demands. This bit can be used in conjunction with bit 6 to determine the enable state, see below.

Bit 6 – Amplifier OK
When set, this indicates that there are no faults present that will prevent the amplifier from enabling. This bit can be used in conjunction with bit 7 to determine the enable state, see Table 15. The enable state is either RED, indicating a fault is present, AMBER, indicating no faults but amplifier not enabled and GREEN, indicating amplifier is enabled. See Section 3 (starting page 21) for a description of the enable state machine. Note that braking states are not represented in the status byte. The 'BRAKE TO RED' state will be reported as 'RED' and the 'BRAKE TO AMBER' state will be reported as 'AMBER'.

Bit 6	Bit7	Enable State
0	0	RED
1	0	AMBER
1	1	GREEN

Table 15 – Determining Enable State From Status Byte

Bit 5 – ESTOP OK
When set, this bit indicates that the ESTOP chain is not broken. This bit reflects the state of the ESTOP OK input (on GPIO D0)

Bit 4 – Latch Event
This bit indicates that either a reference mark has been hit or a touch trigger probe event has occurred. This bit is reflected in P-0-690 (page 126).

Bit 3 – Not Used

Bit 2 – Extended Mode Bit
This bit, when set, indicates that the message being transmitted is in extended mode. That is, the message is four TAPs long.

Bit 1 – Service Channel Do Not Send Bit
This bit, when set, prohibits the master from sending data via the service channel.

Bit 0 – Service Channel Sync Bit
This bit will be toggled by the SPA2 every time a new Service Channel byte is valid.

Service Channel Byte

The service channel is used to send asynchronous data across the synchronous serial link. It is through this that amplifier configuration data (gains, limits etc) and firmware updates can be sent. The protocol works in the following way:

Sending :
To send a byte across the service channel, the byte should be loaded into the Service Channel Byte (second byte of the fifth word of a TAP) and the Service Channel Sync Bit (bit 0 of the control or status word) should be toggled.

Receiving :
The receiver should monitor the Service Channel Sync Bit of the incoming transmissions and when a change of state is detected, the Service Channel Byte should be copied into a receive buffer.

Note that as well as the service channel, asynchronous communications can also run via the SCI port using RS232.

Checksum Protection

The SSC can include a checksum to ensure the integrity of data transmitted. If enabled, the checksum is appended to the end of the entire message (so the message will now be eleven words in normal mode and 21 words in extended mode). If a slave detects a bad check sum, it will ignore and retransmit the message (including the bad checksum) as its response. This ensures that all subsequent slaves and the master will also detect a bad checksum. If the master detects a bad checksum, it ignores the message and if it is in the middle of a service channel transmission, it will re-send the last byte in the next message and set the 'SC Do Not Send' bit to all slaves. This ensures that if a slave had already transmitted a byte, and the checksum failed occurred in the ring after (which means the master never received the byte), the slave will transmit again.

The checksum functionality can be switched on and off in SPA2 using IDN P-0-228 (see page 88). Changing the value of this IDN will not immediately change the state of the checksum. The new value must be saved to flash (using P-0-208, page 76) and the change will take effect on the next reset.

Asynchronous Communications Protocol

The format of the messages that are transmitted across the service channel (or RS232 via the SCI port) is known as the Asynchronous Communications Protocol (ACP). This is a custom protocol, based around a master (host) - slave (DSP) system with binary data. The message IDs are based upon the SERCOS standard (see Appendix C, page 49). The communication cycle starts with the master sending a 'Message' which consists of three bytes. The first two bytes make up the IDN and the third byte is know as the Header which contains information about data length and whether the command is a read or write. A write command will be followed by a number of data bytes and then an acknowledge byte (AAh). A Read command will just be followed by the acknowledge byte. The slave will interpret the command and respond accordingly. In the case of a read, the slave will send the required number of bytes back to the master followed by an acknowledge byte (A0h + E, where E is the error code as described in Table 16). For a write, just the acknowledge byte is sent.

There are three data lengths supported: Fixed length two byte data, Fixed length four byte data and variable length two byte data. For variable length data, the first two bytes of the data section of the message (sent either by master or slave) contain the number of two byte words to follow. There are, then, six types of message that can be sent by the master, Figure 5 shows each of these and the resulting response from the slave.

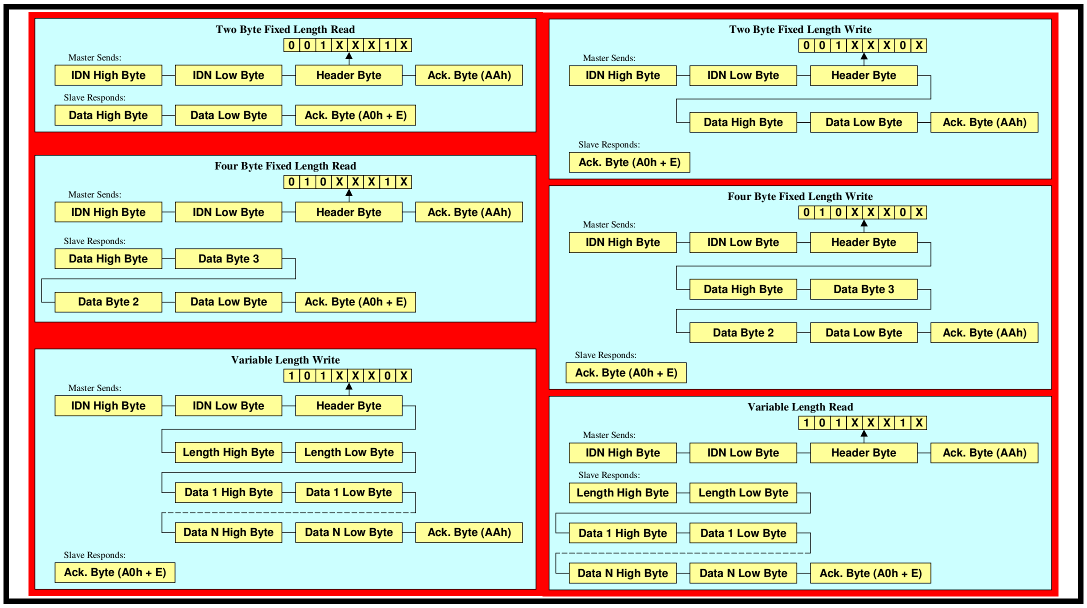

Figure 5 – Serial Communication Messages

The Error code is contained in the low four bits of the slave's acknowledge byte, the meaning of the bits is shown in Table 16. The high four bits of the acknowledge byte is always Ah (1010b). The master acknowledge byte is always AAh.

Bit 3	Bit 2	Bit 1	Bit 0
Unsupported IDN The supplied IDN is not supported	**No Ack Byte** No Acknowledge byte was received from the master.	**Invalid Length Code** Length code contained within header is invalid	**Invalid Data Length** Data length supplied in header does not correspond to the expected length of data associated with the given IDN.

Table 16 – Slave Acknowledge Byte Error Codes

The structure of the header byte is shown in Table 17. Note that only fixed two byte, fixed four byte and variable two byte data lengths are supported. These are highlighted in yellow in Table 18.

7	6	5	4	3	2	1	0
Data Length (see Table 18)			Not Used			R/W-	Not Used

Table 17 – Header Byte

Bit 7	Bit 6	Bit 5	Data Length
0	0	0	Reserved
0	0	1	2 Bytes
0	1	0	4 Bytes
0	1	1	Reserved
1	0	0	Variable 1 Byte
1	0	1	Variable 2 Byte
1	1	0	Variable 4 Byte
1	1	1	Reserved

Table 18 – Data Length Codes (bits 5-7 of header byte)

The code that implements the serial interface is written as a collection of C functions contained within the 'RS232_Generic.c' and 'RS232_SPA2.c' files. The former handles the actual protocol and takes the form of a state machine so that it can run along side other asynchronous processes on the DSP. At every state transition, control is given back to the main function (in ‘main.c’) for other processes to execute. Figure 6 shows the state transition diagram of the state machine that runs on the DSP.

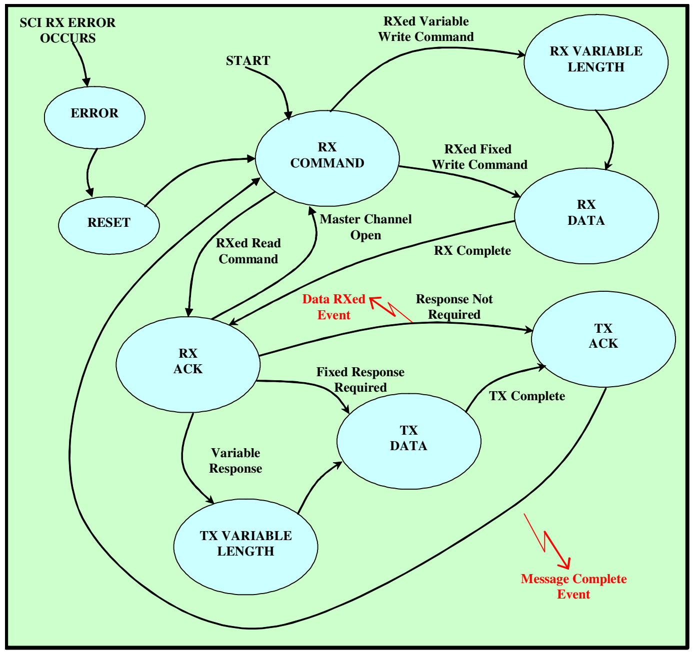

Figure 6 – Serial Comunications State Machine

Appendix A (page 43) contains flowcharts on how the individual states are implemented. The code in ‘RS232_Generic.c’ is a generic module that is used to implement the ACP in several other projects and therefore should not be modified for specific use by SPA2. This module can be configured to use the SCI port (for RS232) or the SSC service channel for transmission and it can also be made to re-transmit certain messages received via RS232 across the SSC service channel. This latter configuration allows the module to act as a bridge between the two formats – see Appendix E (page 135) for more details of an application that does this.

The ‘RS232_SPA2.c’ file contains code that is specific to the SPA2 project and it is this module that is responsible for decoding the IDNs that SPA2 uses. The IDNs as listed functionally and numerically in Appendix C (starting page 49).

By default, the ACP is set to use the service channel. However, if data is received by the SCI port, the ACP will automatically switch to use RS232. If, subsequently, data is received by the service channel, the ACP will automatically switch back to use this.

While the ACP is configured to use RS232, the SPA2 can be changed into an SSC bus master (using IDN P-0-350, see page 101) and while in this mode, IDNs from parameter sets 2 and greater are transmitted across the SSC service channel. In this way, upto four DSPs can be configured via a single RS232 link. To enable this, the SSC ring must be complete, i.e. the SSC out of the last slave DSP must be connected to the SSC in of the DSP acting as a master.

Aries PQR Extension

The Aries PQR variant has a slightly different behaviour within an SSC ring because it has three axes to look after and therefore requires more data than is available in a single TAP. The Aries PQR node, although a single DSP, appears as two nodes in the ring. Figure 7 shows this for a three axis Aries configuration which has a single DSP but consumes both TAPs of a normal mode transmission. Similarly, a five axis Aries system appears as three nodes (despite only physically consisting of two DSPs) and so requires extended mode transmission, see Figure 8.

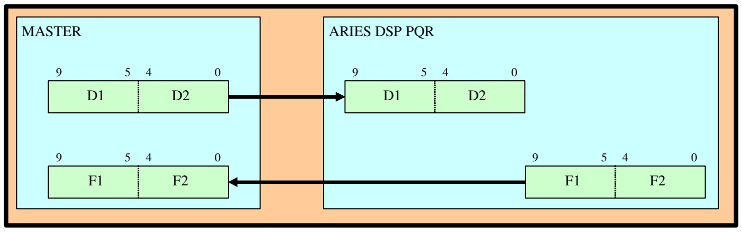

Figure 7 – Three Axis Aries SSC Ring

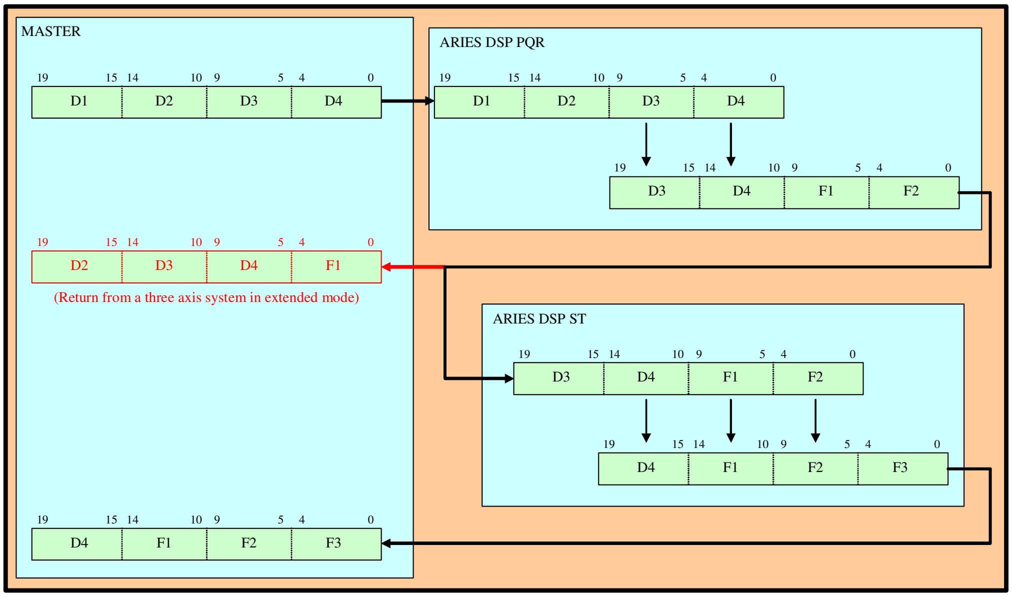

Figure 8 – Five Axis Aries SSC Ring

The control byte received by the Aries PQR node will be the logical OR of the control bytes of the two received TAPs and the same status byte is sent in both transmitted TAPs. Service channel messages from both received TAPs will be processed and the parameter set of IDNs received in the second TAP will be adjusted accordingly. For example, if a message with parameter set 0 is received by the second TAP, it is interpreted as parameter set 2. Responses are sent back via the service channel of the TAP in which the message was received. Messages should NOT be sent on both service channels simultaneously.

SECTION 3 – ENABLE STATE, LIMITS AND FAULTS

The enable state of the SPA2 is determined by a state machine that runs asynchronously to the servo clock. This state machine is scanned once during the main while loop that also scans the serial communications state machine. Figure 9 shows the I/O logic of the state machine, it takes inputs from enable signals and the fault condition of the amplifier and it has two outputs. The first output is ‘DriveOK’, which is an external output of the SPA2. This indicates whether faults are present: high means no faults (states AMBER or GREEN) and low means faults (RED). The second output is ‘Drive Enabled’ which is used to enable the PWM driver circuits (in GREEN only). Figure 10 shows the transition diagram of the state machine.

Note that individual axes can be set to remain disabled (see P-X-200, page 70) so it is possible to have a GREEN state but with one or all of the axes disabled.

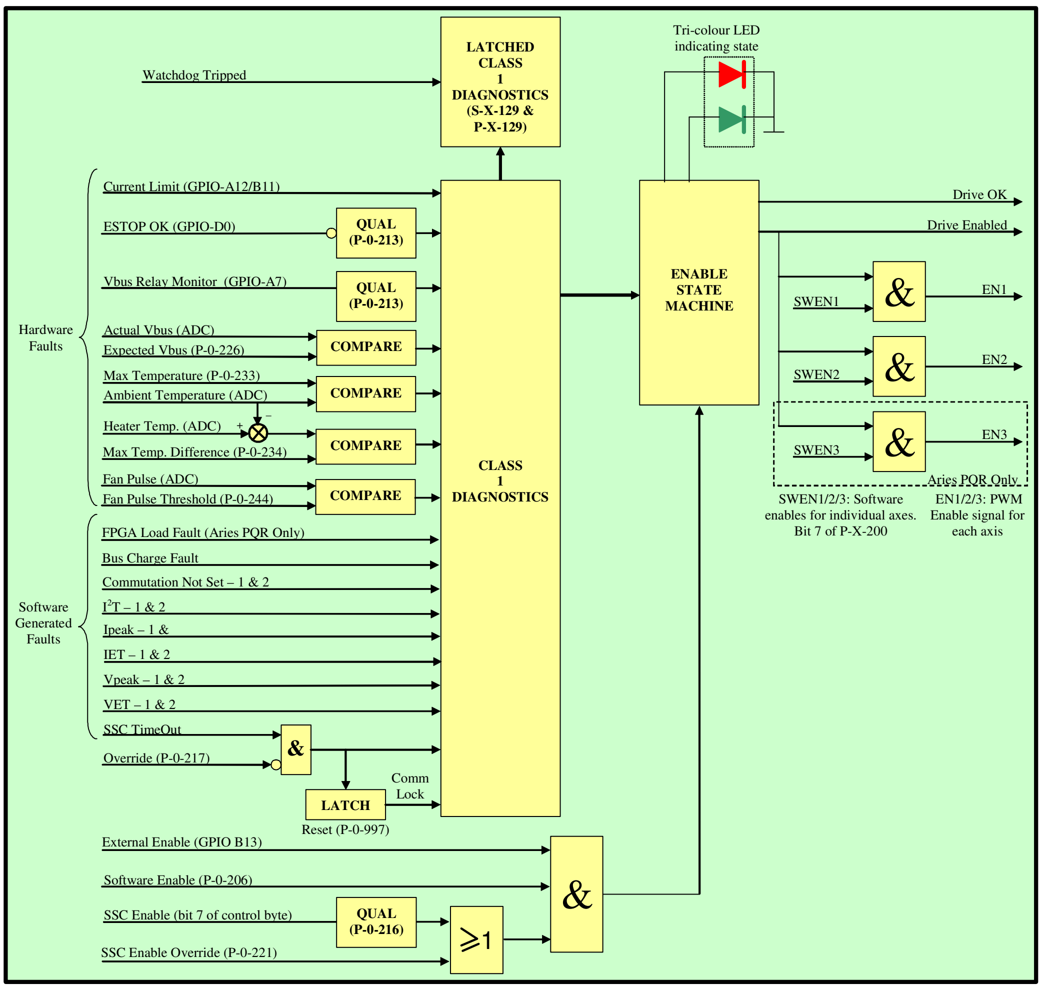

Figure 9 – Enable Logic

Output Table :

STATE	DRIVE OK	DRIVE ENABLED	LED
Green	TRUE	TRUE	Green
Amber	TRUE	FALSE	Amber
Brake To Amber	TRUE	FALSE	Green Amber Flash
Red	FALSE	FALSE	Red
Brake To Red	FALSE	FALSE	Green Red Flash

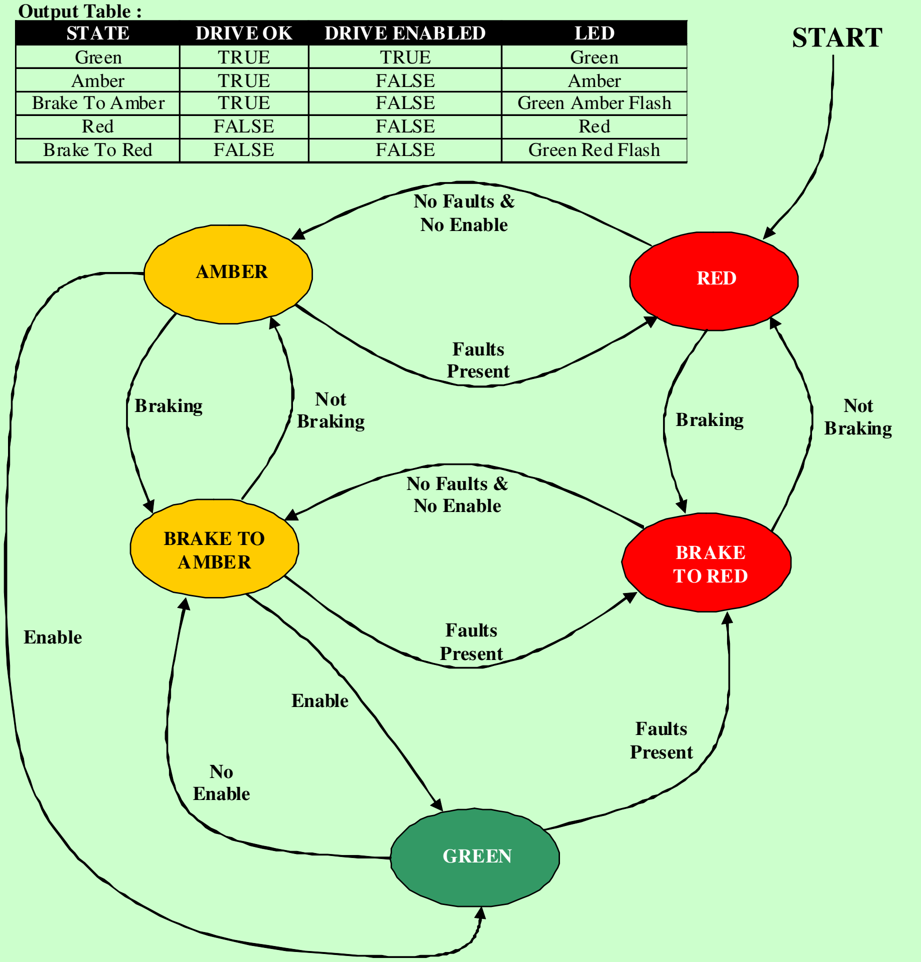

Figure 10 – Enable State Machine

The enable input to the state machine is actually the logical AND of a software enable (represented by P-0-206), a hardware enable (GPIO B13) and the SSC enable (bit 7 of the incoming control byte). The SSC enable is qualified in that it must change state for a minimum number of SSC messages before the change takes effect within the amplifier. The number of messages that the signal is qualified for can be set via P-0-216 (see page 81, default is 10).

During either of the braking states, the duty cycle of the bottom FETs in the drive bridge is controlled to attempt to maintain braking current (as set by P-X-235, page 90) through the load. This in effect acts as a load to the regenerating motor causing it to stop quickly. Once motion has ceased, all the bottom FETs will be switched on, presenting a short circuit load across the motor. There are two different modes (set via P-0-223, page 86) which control the time spent in the braking states: 'Timed' mode and 'Host Controlled' mode. In 'Timed' mode (the default), the braking modes will be active for a specified time (set via P-0-222, page 86, default is 2 seconds). It is up to the user to set this time so that all axes are bought to rest from maximum speed in the set time. In 'Host Controlled' mode, braking is determined by the host using P-0-224 (see page 86).

There are a number of faults that can cause the SPA2 to go into the RED state. When these faults are no longer present the amplifier can be re-enabled. The operation of the state machine is such that if a fault occurs while the axis is enabled, the state will stay red (even if the fault is removed) until at least one of the enable signals is set to disable (at which point, state changes to AMBER if the fault is gone). The register that reads these faults is latched every scan of the enable state machine and these latched values, along with the watchdog tripped bit, can be read as S-X-129 (page 59) and P-X-129 (page 70). The faults in this latched register will NOT clear when a fault is removed, the register can only be reset by the user writing 0 to S-X-129 and P-X-129. The amplifier can be re-enabled if no faults are present regardless of the contents of S-X-129 and P-X-129.

The sections below give more details about each of the faults and the watchdogs. For more details refer to the relevant IDNs in Appendix C.

ESTOP Tripped

This fault indicates that the ESTOP chain has been broken. The SPA2 software can only report this fault, it has no control over the ESTOP condition. This fault is indicated by bit 6 of S-X-129. The 'ESTOP_OK' input (GPIO-D0) that reports the status of the ESTOP chain is qualified in that it must change state for a minimum number of servo cycles before the change takes effect within the amplifier. This number can be set via P-0-213 (see page 80, default is 40 i.e. 4ms). The qualified signal is also reported back to the host in the SSC status word.

SSC Time Out

This fault occurs if a user settable time period (P-0-215, page 81) elapses without a SSC message being received. This fault can be disabled with P-0-217 (see page 82). This fault is indicated by bit 7 of S-X-129. Note that if SSC checksum is enabled (with P-0-228, page 88) a message with a bad checksum does not count as a message received for the purposes of this fault.

Note that if a SSC Time Out fault exists, the SSC enable signal is considered to be true. This prevents the enable state from changing immediately to GREEN upon re-establishment of the link. It also, in conjunction with the SSC Time Out Override flag (P-0-217, see page 82), allows the amplifier to be used without an SSC link.

SSC Comm Lock

This is a latched version of the SSC Time Out fault which is indicated by bit 14 of S-X-129. This bit is set when the SSC Time Out fault is set but can only be reset by the host writing the number 0x5AFE to IDN P-0-997 (see page 132).

PWM Over Current Limit

This fault is over current as detected by the PWM driver circuits and will occur if the axis current instantaneously exceeds a maximum and/or there is a current imbalance in the motor phases. The actual value of the limit is dependant upon the hardware type as shown in Table 19. This fault is indicated by bit 0 of S-X-129.

Hardware Variant	Max Current	Max Imbalance
SPA2	12A	±1A
SPA2-2	n/a	±1A
SPAlite	6.33A	n/a
Aries	20A	n/a
Gyro	n/a	±300mA

Table 19 – PWM Over Current Values

Vbus Fault

The PWM bus voltage on each axis is monitored via an ADC channel. A Vbus fault occurs if the measured value of this voltage differs (i.e. greater or less) from the expected value (entered in P-0-226, page 87) by more than the tolerance value (entered in P-0-248, page 95). The default tolerance is set to 4V for Aries and 10V for other variants. This fault is indicated by bit 4 of S-X-129.

Vbus Relay Open

Power to the PWM bus comes via a relay that is controlled by the external SPA2 watchdog, an external input and the SPA2 itself. The status of the relay is monitored (GPIO-A7) and this fault results if the relay is sensed open. This relay monitor input, like the 'ESTOP_OK' input, is qualified for the number of servo cycles specified in P-0-213. This fault is indicated by bit 8 of S-X-129.

The SPA2 can control the state of this relay via GPIO-D5. The relay is opened if an ESTOP Tripped or Comm Lock fault exists or if P-0-995 = 0 and is closed in the absence of these conditions.

Bus Charge Fault

This fault occurs when the bus voltage does not rise to the required level in the required time after the Vbus relay closes. It is generated by the state machine that controls the In-rush relay (see description starting on page 26). This fault is indicated by bit 13 of S-X-129.

Peak Current Limit

This is a software generated fault and occurs if the axis current ever reaches 'Full Scale Deflection' (FSD) of the ADC input. This should never happen since the current demand is limited to I_{PEAK} by the servo. I_{PEAK} is 50% of I_{FSD} and I_{FSD} is set using the analogue switches (P-X-207, page 74). This limit is disabled when the axis is configured in open loop or current loop bypass mode. This fault is indicated by bit 1 of S-X-129.

I^2t Limit

This is a software generated fault that protects against motor heating. The fault occurs if too much motor current flows for too long. Continuous current, or anything less, will be allowed to flow for unlimited time without fault. Peak current (I_{PEAK}) will be allowed to flow for the specified time. Anything between continuous and peak current will be allowed to flow for a time governed by the I^2t law. The limit is controlled by two user parameters: S-X-111 (see page 59) which is the allowable motor continuous current expressed as a proportion of I_{FSD} and P-X-204 (see page 73) is the maximum time that peak current (i.e. 50% of I_{FSD}) can safely be applied to the motor, expressed in servo cycles. The default values are S-X-111 = 8192 and P-X-204 = 19999, which will allow a quarter full scale current (i.e. half peak current) continuously and peak current for 2 seconds. This limit is disabled when the axis configured in open loop mode or if S-X-111 is set to higher than 16384 (I_{PEAK}). This fault is indicated by bit 2 of S-X-129.

Current Error Time Limit

This is a software generated fault that protects against excessive servo error in the current loop. This works in a similar way to I^2t except that the lower threshold is fixed at 5% of I_{FSD} and the law governing a trip is linear, not square. This fault will occur if there is a break in the loop after the DSP (e.g. no motor power or motor not connected) or if the current servo is badly tuned. This limit is governed by a single parameter: P-X-205 (see page 73), which is the length of time the current error is allowed to remain at I_{PEAK} (50% of I_{FSD}) measured in servo cycles. The default is 20000 which will allow $I_{ERR} = I_{PEAK}$ for 2 seconds. This limit is disabled when the axis configured in open loop mode or if P-X-205 is set to zero. This fault is indicated by bit 3 of S-X-129.

Peak Velocity Limit

This is a software generated fault that protects against excessive motor speed. This fault works in a similar way to the peak current limit, except that the V_{FSD} condition must persist for a minimum time period (set in P-X-220, page 85) for the fault to occur. This is to make allowance for noise on the tacho input signal and small amounts of overshoot in the velocity servo. Another difference is that whereas the current demand is limited to 50% of I_{FSD}, the velocity demand is limited to 95% of V_{FSD}. This limit is only enabled when in velocity loop mode. This fault is indicated by bit 5 of S-X-129.

Velocity Error Time Limit

This is a software generated fault that checks to see whether there is excessive velocity error for too long. This works along the same lines as the Current Error Time limit but uses 10% as its safe threshold instead of 5% and P-X-236 (page 90) to set the time limit. This fault is indicated by bit 11 of S-X-129.

The purpose of this fault is to detect if a velocity sensor (tacho or encoder) has become disconnected. Note, however, that while this offers some protection, it will not operate if there is zero velocity demand even though the motor may well be turning uncontrollably because of noise on the unconnected velocity input. This limitation can be removed if the host controller is sending a velocity feedback signal via an SSC synchronous input. In this case, the Velocity Error Time fault can be set to use the error between this velocity signal and the on board tacho/encoder signal (as opposed to the velocity servo error) as an input. This is done using bit 15 of the servo options word (P-X-200, page 70).

Over Temperature Fault

The temperature of the air entering the SPA2 enclosure is measured with a sensor that is mounted close to the air intake fan. This representation of ambient temperature is fed to the DSP via an ADC input and can be read directly with P-0-231 (page 89). If the value, at any time, exceeds the maximum allowable temperature (set in P-0-233, page 89) then the Over Temperature Fault will occur. This fault is indicated by bit 9 of S-X-129. The max temperature parameter (P-0-233) should be set on an amplifier by amplifier basis depending upon the expected ambient temperature of the application.

Fan Fault

This fault, indicated by bit 10 of S-X-129, suggests that there is a problem with the fan in the amplifier box. How this is detected is dependant upon hardware type:

SPA2

There is a second temperature sensor mounted close to the air intake fan in SPA2. This one is artificially heated by a power resistor mounted close by. The temperature reported by this sensor can be read with P-0-232 (page 89). If the fan is working, the difference between the two sensors is smaller than if the fan is not working or if the air intake is blocked. If the temperature difference is greater than the value set in P-0-234 (page 90) the Fan Fault will occur.

Note that the temperature difference will depend upon anything that affects air flow within the SPA2 enclosure, such as which PSU is fitted or which fan is used. Therefore the trip value (P-0-234) should be set on an amplifier by amplifier basis according to the box configuration.

SPA2-2 and Aries

The pulse tacho on the fan is filtered and applied to an ADC input on the DSP. Assuming that the fan is in motion, the output of the filter should be close to the midrange of the ADC (Vr). If the fan stops the filter output will settle at one extreme (0V or full range of ADC). The fan fault will exist if the value read by the ADC is differs from Vr by more than the specified threshold (set in P-0-244, page 93).

SPAlite and Gyro

This fault is disabled for SPAlite and Gyro which do not have a second, heated temperature sensor or the ability to a read fan pulse tacho.

Commutation Fault

In the case of encoder only commutation (P-X-680 = 0), the purpose of this fault is to prevent a brushless or linear motor from being enabled in a commutated mode before the correct commutation angle has been set. After a valid pole search has been performed on an axis, the commutation set flag (P-X-654) is set. The flag helps determine whether the fault is present, which it will be if all of the following are true:

1. Brushless motor mode is selected
2. Loop mode is commutated current or velocity
3. Commutation Set flag (P-X-654) is NOT set

When one of the Hall code commutated modes (P-X-680 = 1 or 2), the fault will be present if an illegal Hall code is read. An illegal code is either 0 or 7 (all on or all off). Like before, the fault is disabled if not in a commutated loop mode or if 'Brushed Motor' is selected.

This fault is indicated by bit 12 of S-X-129.

Watchdog Fault

The SPA2 is protected by two independent watchdogs, both are set during the scan of the enable state machine and reset during the servo interrupt. Thus both parts of code have to be functioning correctly and running frequently enough for the watchdogs not to trip.

The first watchdog is internal to the DSP chip and if not kicked for more than 4.4ms a chip reset will be generated and the watchdog tripped bit will be set. This bit is mirrored in S-X-129 bit 15. This bit will also be set after a successful firmware download has been performed using P-0-400. An internal watchdog trip does not open the Vbus relay because the reset line is not held low enough for the relay to react (it is low for approx. 17µs).

The second watchdog is external to the DSP and will trip if not kicked for more than 112ms. This causes the DSP to be reset and opens the Vbus relay. Note, however, that the watchdog output is only in reset for a short time (typically 25ms) and so the Vbus relay will immediately close again if all of the other controlling signals are in a safe state. Note also that the external watchdog will not set the watchdog tripped bit. Table 20 summarises the properties of the two watchdogs.

Watchdog	Time Out	Tripped Bit	Vbus Relay
Internal	4.4ms	Set	No Effect
External	112ms	Not Affected	Opens Momentarily

Table 20 – Watchdog Details

The tripping of either watchdog will not necessarily cause a RED state if no other faults are present and the saved state of software enable is false (in which case the reset state will be AMBER; reset state can never be GREEN).

FPGA Load Fault (Aries PQR Only)

This fault is activated during start up of the Aries PQR V1 variant if the FPGA program is not detected in flash (i.e. memory location 0x3DC000 is not zero) or if the FPGA configuration process does not complete properly. Once set it can never be cleared. For variants other than Aries PQR V1, this fault will never be set. This fault is indicated by bit 0 of P-X-129.

In-Rush Relay Control

The power stage of the amplifier contains a large amount of capacitance which results in a high charging current when the bus voltage is applied. To protect the power supply and the relay contacts that switch the bus voltage, a resistor is placed in series with the main power input to limit the charging current. So that this resistor does not affect normal operation of the amplifier, it is shorted out by a relay once the bus voltage is fully established. From hardware version 5 of SPA2 (and from V2 of SPAlite) this ‘in-rush’ relay is under the control of the DSP firmware via one of the GPIO lines (B7). Upon start up, this line is configured as an input and is read. A zero indicates that the line is connected to the relay control circuit and so it is re-configured as an output and the code that controls the relay is enabled. A one indicates that the line is not connected to anything, therefore the relay control code is disabled. This allows the same code to run on both DSPs even though only one DSP will actually control the relay (the one that is wired to the control circuit). It also means that the firmware will also work on earlier hardware versions of the amplifier where the relay is not under control of the DSP.

Control of the relay is based on the principle that the bus charge profile is governed by a first order RC law. A property of this is that it takes nine times as long for the bus to charge to 100% (within 0.2%) as it does to 50%, regardless of the actual value of the bus voltage. The charge time is monitored, starting from when the Vbus relay closes, and if it has not reached 50% of expected value (P-0-226) in a set time (T_{FAULT}) then a ‘Bus Charge Fault’ occurs. This could indicate a short circuit in the power stage (though it could also indicate that P-0-226 has been incorrectly set) and thus prolonged application of voltage to the inrush resistor could cause it to overheat and cause damage to the board and even become a fire risk. Therefore, the fault condition will remain for a predetermined time (T_{SAFE}) before the process is allowed to start again, allowing the resistor to cool down.

If 50% of bus voltage is attained within T_{FAULT}, the actual time taken is noted (T_{ON}) and a further wait of eight times this is made, i.e. overall time from start of charge is nine times T_{ON}. After this wait, the bus is assumed to be fully charged and the in-rush relay is closed, shorting out the charging resistor.

The control of the in-rush relay is achieved using a state machine as shown in Figure 11. Current values for T_{FAULT} and T_{SAFE} are 0.8 and 21 seconds respectively.

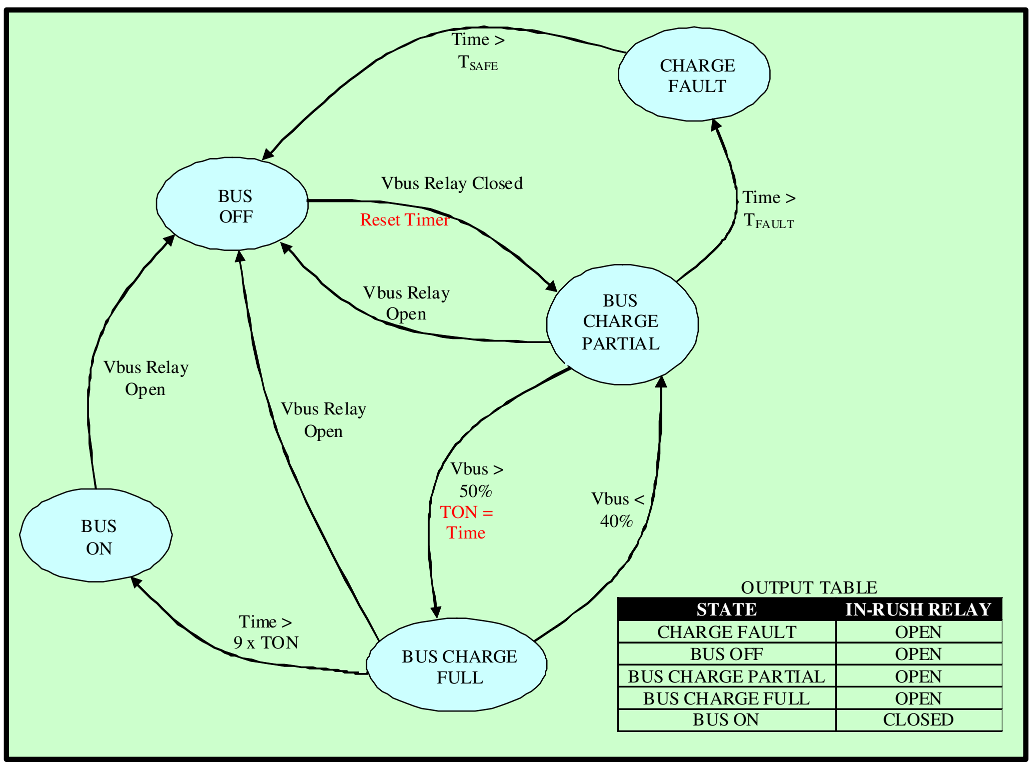

Figure 11 – In-Rush Relay Control State Machine

SECTION 4 – SERVO LOOP CODE

This code runs when the CPU Timer 2 interrupt occurs. The servo interrupt is set to occur every 100µs by default but can be changed using P-0-247 (see page 94). The code is contained within the file 'servo.c'. The following tasks are performed each servo cycle :

- Start ADC Convert.
- Latch encoder values.
- Start analogue encoder board ADC convert (if fitted)
- The servo clock monitor output (GPIO E0) is set.
- Re-enable interrupts (allow comms ISR to interrupt servo)
- Test input generator output is updated.
- For each axis in turn (3 axes for Aries PQR, 2 axes for all other variants):
  - Wait for ADC Conversion of axis channels.
  - Call the Single Axis Servo Filter code (see below).
- Handle TTP (Aries PQR only)
- If data gathering is enabled, data is transferred to the gather buffer.
- Calibrate ADCs if Calibration mode is active.
- Servo counter is incremented.
- If in SSC Master mode, send SSC transmission.
- Reset ADC sequencer.
- Reset SPI port to analogue encoder board ADCs (if fitted).
- Kicks watchdogs (internal and external).
- Servo clock monitor output (IOPE4) is reset.
- Store time in servo routine.

Some of these tasks are described in greater detail in the sections below.

Test Generator

The test generator function supplies the servo inputs with a test stimulus. This can either be a square wave that is generated inside the DSP or a user defined waveform that can be downloaded. P-0-100 controls the type of test signal and determines which axes and phases it is applied to.

In-Built Square Wave Generator

If bit 7 of P-0-100 is 0, the test generator output is derived from and internal square wave generator. It is controlled by three IDNs: P-0-101 sets the period, P-0-102 sets the high amplitude and P-0-103 sets the low amplitude.

Downloadable Wave

If bit 7 of P-0-100 is 1, the test generator output is derived from a buffer of data, up to 16384 samples long, that is set via P-0-104. The sample period of this data is set in P-0-105 and P-0-106 sets the waveform length. The current sample in the buffer is given in P-0-110 and this can be written to change the current phase of the output. Under normal operation (bits 14 and 15 of P-0-100 set to '0') the generator output continuously repeats, i.e. the buffer pointer wraps around when it gets to the end. Setting bit 15 of P-0-100 will freeze the buffer pointer, and hence the generator output, at the current position. Setting bit 14 of P-0-100 will cause the pointer to freeze once it reaches the end of the buffer giving 'single shot' operation – the output is re-triggered by manually setting the pointer back to 0 and setting bit 15 of P-0-100 to '0'.

When using the downloadable wave, the test generator has two outputs, the second is phase delayed with respect to the first by the number of samples indicated in P-0-109. This second output is applied to all phase B current inputs (when selected). For the In-Built Square Wave Generator, the phased output is the same as the main output.

For more details, see the IDN descriptions for the test generator (Appendix C, starting page 66).

Single Axis Servo Filter

The servo filter for each axis can run in a number of different modes and loop configurations, and these are set using the ‘Options’ word for that axis (P-X-200, page 70). See Figure 13 for a block diagram representation of the different modes and configurations. The SPA2 servo can operate in the following modes:

Brushed / Brushless Motor Mode

The SPA2 can be configured to drive either a three phase brushless motor or a single phase DC brushed motor. This mode is set using bit 13 of P-X-200. In the brushless case, the motor can be driven with or without commutation – if without, commutation must be done by the higher level controller and individual phase currents sent to the SPA2 via the SSC link. Setting brushless mode on an Aries or SPAlite variant is meaningless since the hardware will not support brushless motors.

Current Loop Bypassed Mode

If this mode is set (valid in any loop configuration except Open Loop), then the current servos are bypassed and the current demand is passed directly to the PWM duty cycle. This means the motors are being driven in voltage rather than current mode. This mode is set using bit 8 of P-X-200.

In addition to these modes, the servo loop can have one of four possible servo loop configurations as described below.

Open Loop

In open loop mode, no servos are enabled and PWM duty cycles (equivalent to phase voltages) can be specified for all three phases of each axis (using P-X-61/2/3). This mode is useful for test purposes only.

Non-Commutated Current Loop

This mode is only available in brushless motor mode. Three current servos are implemented for each axis, one for each phase. The demands for phases A and B are the sum of three sources: SSC demand (A and B), Digital demand (via P-X-21 and P-X-22) and the test generator. The demand for phase C is derived from the inverse of the sum of the combined demands of phases A and B. The combined demands of phases A and B are limited to 50% of full scale (i.e. 16384) and phase B is further limited to ensure that the demand to phase C does not overflow.

Each of the three current loops consist of a PI filter with exactly the same coefficients. In addition, there are independently settable 2nd order IIR filters (see below) on phases A and B current feedback.

Trying to set this loop configuration in brushed motor mode will result in Commutated Current Loop being selected.

Commutated Current Loop

In brushed motor mode, the current demand is passed to a single current loop which drives a single motor load between phases A and C of the bridge.

In brushless motor mode commutation is applied to the incoming demand in conjunction with position feedback to provide two phase demands to the phase current loops (the third derived from the other two as described above). The current demand is the sum of three sources: SSC demand (A), Digital demand (via S-X-81) and the test generator. The two phase currents (*Ia* and *Ib*) are given by:

$$Ia = I\sin(\theta)\ \ \&\ \ Ib = I\sin(\theta + 120)$$

Where θ is given by:

$$\theta = \frac{PositionCounts}{CountsPerPolePair} \times 360$$

And *CountsPerPolePair* is given by:

$$CountsPerPolePair = \frac{CountsPerRev\,(\text{P–X}-651)}{NoOfPolePairs\,(\text{P}-\text{X}-652)}$$

Position information for commutation can be derived from the on board encoder inputs (analogue or digital) or SSC Input 'B' from the host, selectable with bit 3 of P-X-200. Since these are incremental or relative devices the initial value has to be set after the DSP has been reset and prior to any motion being possible. This can be set externally by poking the correct number into P-X-650 or there is a commutation angle search algorithm built into the SPA2 firmware. See page 39 for more details on this.

Commutation position can also be derived from digital Hall sensors. The sensors should be placed at thirds around the electrical commutation cycle so that six transitions can be detected per cycle, all 60º apart. Since these signals are absolute in nature, there is no need to find the commutation angle at start up, however, the resolution achieved is low (60º) which leads to greater torque ripple from the motor.

The signals from the sensors are received via the digital encoder channel on the appropriate axis (reference mark input used for the third sensor) and therefore it is not possible to use the hall sensors and a digital encoder at the same time. There are two modes that take advantage of the Hall feedback and these are selected using P-X-680 (see page 124). The two modes are:

Hall mode (P-X-680 = 1)

This mode simply converts the hall code into an angle (six angles per electrical cycle) and then uses this in the above calculation of commutation. This is akin to six step trapezoidal commutation. The sequence of codes as the motor traverses an electrical cycle is 1, 3, 2, 6, 4, 5 corresponding to 0, 60, 120, 180, 240 and 300 degrees. The Hall code itself is returned in P-X-682 (see page 125) and the Hall angle is returned in P-X-683 (see page 125). The commutation angle is the Hall angle plus an offset plus 30 degrees. The offset is set in P-X-681 (see page 125) and is defined as the desired commutation angle at the first Hall transition (i.e. the transition from code 5 to 1). The extra 30 degrees is added to make the commutation point in the middle of the Hall transitions. The direction in which the Hall codes count can be changed with P-X-684 (see page 126) which allows the hall code and motor polarities to be matched.

Encoder Reset From Hall mode (P-X-680 = 2)

On start up, this mode works in the same way as Hall mode. When the first Hall transition from code 5 to code 1 occurs, commutation switches to using the selected incremental encoder which is reset to the Hall offset value (P-X-681). On all subsequent detections of the first Hall transition (i.e. 5 to 1 or 1 to 5) the commutation is reloaded with the offset.

This mode allows full sinusoid commutation from an encoder without the need of an initial angle search. However, since the digital encoder inputs are required to interface to the Hall sensors, this mode will only work in conjunction with an analogue encoder (via the analogue encoder board). Extra hardware and a code change would be required to enable the (very useful) combination of Hall sensors and a digital encoder.

Velocity Loop

The velocity demand is the sum of three sources: SSC demand A, Digital demand (via S-X-37 see page 54) and the test generator. The output of the velocity servo is passed on to the commutation calculation (in brushless motor mode) or the current servo (in brushed motor mode).

Velocity Feedback

Feedback for the velocity loop can come from either an incremental encoder or an analogue tacho (or the sum of the two if bit 6 of P-X-200 is set) as described in the sections below. In addition, the second SSC demand can also be added to the velocity feedback, in the case that this signal represents position (e.g. for commutation), it can be differentiated prior to use as velocity feedback.

Analogue Tacho Feedback (P-X-200 bit 14 set)
The analogue tacho input utilises a tracker function that uses an external 16-bit DAC to extend the resolution of the ADC inputs on the DSP. The job of the tracker is to estimate the analogue voltage on the tacho input and output this estimate to the DAC. In hardware, the actual input and the estimate are subtracted from one another and the error is multiplied by a gain and then applied to an ADC input. This error signal is used to revise the value applied to the DAC next servo cycle. The tacho feedback is given by the addition of the DAC output and the ADC input. The resolution of this signal is given by the resolution of the ADC input (12 bits or 4096 levels in this case) multiplied by the hardware gain used (10). Thus the resolution is 40960 levels for the full bipolar range.

Incremental Encoder Feedback (P-X-200 bit 14 reset)
Encoder feedback can be from an analogue (sin/cos) or digital (A quad B) input. It can also be received from the host controller via the second SSC Demand. The actual source is set using bits 2 & 3 of P-X-200 (see page 71), whichever is selected is also used for commutation.

In the case of the analogue encoder, the position value is interpolated by performing an arctangent function on the received sine and cosine signals. There is no independent record kept of number of number of cycles, so the input signals must be slow enough to be tracked by the servo code. This means that the inputs must not change by more than half a cycle (i.e. 180º) in one servo cycle (default = 100µs) meaning that the maximum input frequency is 5kHz (less if the servo cycle time is increased). The result of the interpolation is stored as encoder counts (P-X-600 see page 107) scaled as 65536 counts per cycle.

There are a couple of techniques that can be applied to the analogue encoder signals to improve the integrity of the feedback.

Heydemann compensation is a technique used in laser interferometry to correct a Lissajous before interpolation (see reference [16]). The technique corrects for offset, gain and phase error in the sine and cosine signals and works by fitting an ellipse to the raw signals and calculating a set of parameters that can be applied in real time to transform that ellipse into a circle. There are four parameters in the set that control offset (p and q), gain error (G) and phase error (a). They are applied to the raw input signals as follows:

$$S_C = S_R - p$$

$$C_C = \frac{(S_C \times \sin a) + G(C_R - q)}{\cos a}$$

Where S_R & C_R are the raw sine and cosine input signals and S_C & C_C are compensated the signals. To reduce calculations within the DSP, five parameters are actually supplied with a specified as sin(a) and 1/cos(a). The parameters are sent using P-X-670 through P-X-674 (see page 122).

In addition to, or instead of, Heydemann compensation, a look up table can be supplied to correct a Lissajous that is not elliptical. The table is 16 elements long and acts on the angle after interpolation. Each input data point is interpolated between the relevant two map points. There are actually 17 elements stored, the last being the same as the first to ease calculation. The result is a correction that is subtracted from the input angle to give a corrected angle. The table is sent to the amplifier using P-X-675 (see page 124) and it is activated by setting bit 5 of P-X-200 (see page 71).

Extended Velocity Servo

The velocity servo is a PID filter with the following additions:

Acceleration Feed-forward
The velocity demand is differentiated, filtered and multiplied by an acceleration feed-forward gain (P-X-612 see page 110). The result is added to the output of the velocity servo filter.

Note that if the SSC inputs are enabled (bit 12 of P-X-200 see page 71) then the calculation to derive acceleration is only performed when a new valid demand arrives. Therefore, the filter coefficients and (P-X-504 see page 105) and acceleration derivation gain (P-X-618 see page 112) will be based upon the incoming sample frequency, not the internal sample frequency of the velocity loop.

The acceleration demand that is derived here is also used by the Advanced Friction Offset (see below).

Simple Friction Offset
An offset (P-X-613 see page 111) is applied to the velocity servo output with the same sign as the velocity demand. This is intended to compensate for the effect of static friction which can limit the performance of a servo upon reversal.

Advanced Friction Offset
The simple friction offset can suffer from an oscillation problem caused by the fact that the servo input supplied by UCC via SSC is not a pure velocity demand, but rather the output of a position servo. This means that, when holding position, the demand is generally very small in magnitude but tends to change sign a lot. This causes the simple friction offset to put rather large step changes into the current demand.

The Advance Friction Offset (AFO) is designed to get around this problem by only applying the step of torque demand when the machine is actually changing direction. Further, the offset will decay away to nothing after the initial step is applied at a rate that can be easily tracked by the velocity servo.

The AFO algorithm achieves its goal by applying the offset (amount in P-X-616 see page 112) to the output of the velocity servo when a change in sign of velocity is detected if, and only if, the derived demand acceleration is greater than a certain threshold (P-X-614 see page 111). The offset is applied in the same sense as the acceleration demand and then decays to nothing with a first order roll off, time constant given by P-X-615 (see page 111). Figure 12 shows the desired operation of this algorithm, notice the lack of AFO output at the beginning and end of each move.

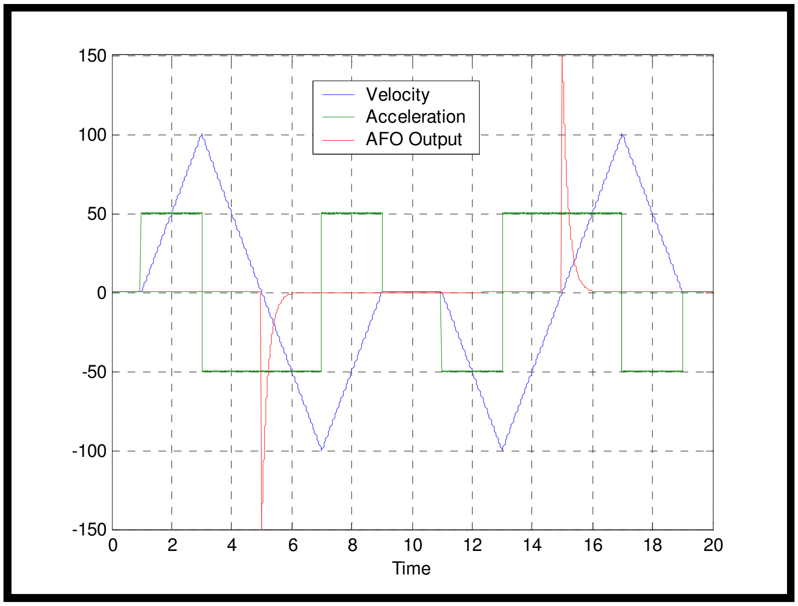

Figure 12 – Plot Showing Desired Operation Of AFO

Coupled Mass Mode Rejection

This is designed to modify motor motion to compensate for dynamics due to a non stiffly connected load which can lead to stability problems in some machines. The SSC Velocity feedback value (P-X-630 see page 114) is used as an indication of what is happening to the load, usually from a linear encoder mounted on the axis of the machine. This is scaled (P-X-611 see page 110) and filtered (P-X-500 see page 103) and the result (P-X-631 see page 115) has the on board velocity feedback (i.e. from tacho or encoder usually mounted on the motor) subtracted from it. This difference (V_{DIFF}, P-X-632 see page 115) has a gain (K_{CM}, P-X-633 see page 115) applied and is added into the velocity servo output. Note that V_{DIFF} can also be used by the Velocity Error Time fault (see page 25).

An alternative way of applying this compensation is to add the scaled and filtered velocity from SSC to the suitably scaled on board velocity. The result makes up the actual velocity feedback. This method makes setting the velocity scaling difficult since two feedback gains have to be co-ordinated to set the overall scaling factor of the velocity loop. This mode is selected using bit 15 of P-X-200 (see page 71).

SSC Current Offset

The second SSC demand can be used to feed an offset into the input of the current loop, thus allowing the implementation, in conjunction with the host, of current or torque feed forward.

Torque Ripple Compensation

This feature allows a (motor) position and direction dependant offset to the applied to the current loop input. This allows correction of torque ripple in a motor due to things like cogging.

The compensation is specified as two look up tables (one for each direction) that can be calculated by correlating current demand with motor position from a suitable data log. The tables are variable in length (specified with P-0-644, page 117, default is 129) and the points are linearly interpolated. To ease calculation the first and last entry in the table correspond to the same point. The tables are stored in and directly used from flash memory (separately from the other flash saved parameters) and can be loaded using P-X-642 and P-X-643 (see page 116). Which of the two tables is used depends upon the sign of velocity feedback (S-X-40, page 55) with a small amount of hysteresis (50). The positive map is used until the velocity feedback falls below –50, and then the negative map is used until the velocity feedback is greater than +50.

Although primarily intended for use with Aries, the Torque Ripple Compensation feature is available for general purpose use. It is, however, restricted to working from analogue encoder position feedback only, and then only if input is one cycle per rev of the motor. Under these conditions, the interpolated result is absolute within one rev of the motor. Any other scenario (i.e. more than one cycle per rev or a digital encoder) means that the position used as input to the look up table must be referenced some how (e.g. using encoder reference mark). This restriction suits the Aries application – extra work would be required to make this feature truly generic.

IIR filters
There are eleven 2nd Order IIR filters used in the servo loops:

1. Main velocity feedback
2. SSC velocity feedback
3. Forward path following PID (2 of these)
4. Acceleration demand
5. Derivative term in PID
6. Current loop feedback (Phase A and B)
7. Analogue Encoder Inputs (Sine and Cosine)
8. Analogue Tacho Input

The coefficients of each filter can be independently specified so each filter can take any configuration (e.g. low pass, high pass, band pass, notch etc.).

Sample Rate Decimation
The sample rate of the velocity servo can be set to be slower than the rest of the servo code (set by an integer divisor in P-X-604), and in this case the velocity feedback can be set to be the averaged value of intermediate samples.

Figure 13 shows a block diagram of the servo configuration. Figure 26 in Appendix A shows a flowchart of how the actual code is implemented.

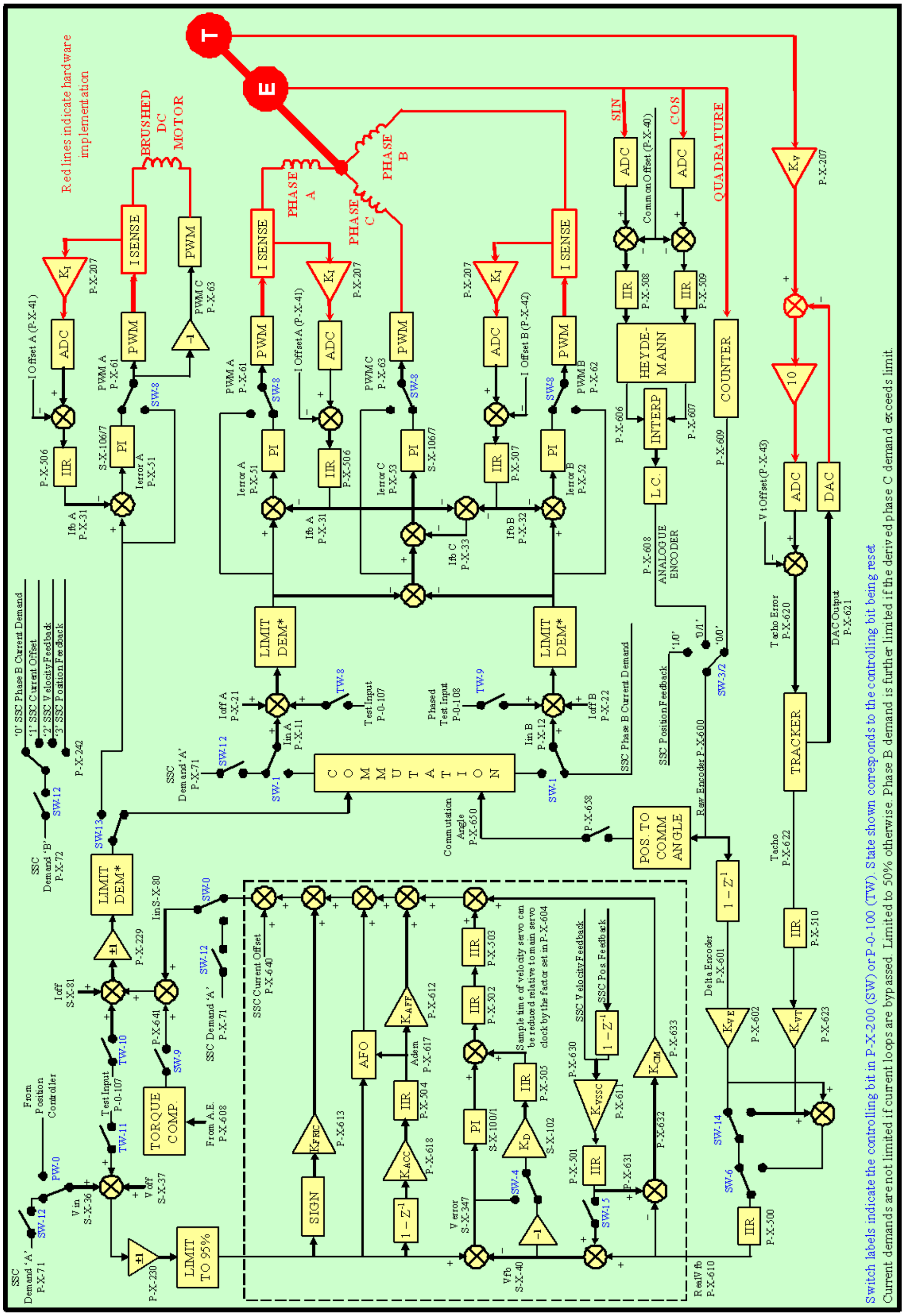

Figure 13 – Servo Filter For A Single Axis

PI Filter

The PI filter is a generic function that is used in both the velocity and current servos. It is implemented as shown in Figure 14.

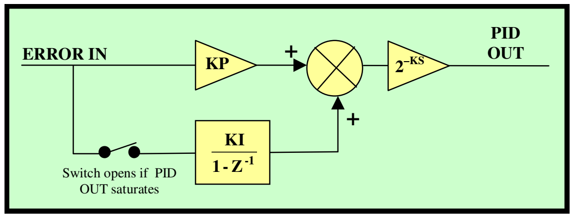

Figure 14 – Block Diagram Of PI Filter

The transfer function of the filter is :

$$U(Z) = \frac{\left[\frac{KI \times Ts}{1 - Z^{-1}} + KP\right]E(Z)}{2^{KS}}$$

This is equivalent to the following difference equation which is implemented by the function 'PID' in the 'servo.c' file. Note that the integral gain entered via the relevant IDN actually represents $KI \times Ts$.

$$U_k = U_{k-1} + \frac{(KI \times Ts + KP)E_k - KP.E_{k-1}}{2^{KS}}$$

Position Loop

The SPA2 can be used to control position. The setting of this is mutually exclusive to the other loop modes in that the position loop can be activated with velocity loop or with current loop, the output of the position servo is passed on as the demand to whatever main loop is selected. Position loop is activated by setting bit 0 of the position loop options word (P-X-700, page 128).

Position feedback can be derived from the same three encoder sources that are available to the velocity encoder signal, i.e. Analogue encoder, Digital encoder or SSC Position feedback. The selection is made using bits 1 and 2 of P-X-700. Feedback is extended to 32 bits and returned via S-X-51 (see page 55). The Demand (also 32 bits) is the sum of S-X-48 (see page 55) and the output of the test generator (the latter is extended to 32 bits by a multiplier in P-X-111, see page 70). Note that there is no SSC 'A' position demand in the current implementation, i.e. the position demand cannot be received via the SSC synchronous demands.

Though the demand and feedback are 32 bits, the error is limited to 16 bits (returned by P-X-701, page 129) and applied to a PI filter (Ks, Kp and Ki = S-X-77, S-X-104 and S-X-105 respectively) of the same type used in the velocity and current loops.

Trajectory Calculator

On its own, the position loop is a bit brutal since it pays no regard to velocity or acceleration limits and simply attempts to reduce any position error by issuing whatever velocity demand is required (which will generally be maximum velocity). Therefore, to make the position loop useful, the position demand can be derived from a trajectory calculator which limits any step change in position to a maximum velocity and acceleration. The generator also supplies a true velocity demand which can be fed forward, via the velocity feed-forward gain (P-X-705, page 130), to the output of the position loop PI filter.

The trajectory calculator is enabled by bit 3 of P-X-700 (see page 128). During a move (i.e. when the input target position is not equal to the current output position), the distance it would take to stop at maximum deceleration from the current velocity is continually calculated. If this distance is equal or less to the distance remaining in the move (i.e. difference between the current position and the target position) then constant deceleration is applied to the output. Otherwise, either constant acceleration (if the current velocity is less than max velocity) or constant velocity (if it isn't) is applied.

Max velocity and acceleration are entered using P-X-703 and P-X-704 respectively (see page 130). Figure 15 shows a block diagram of the position controller.

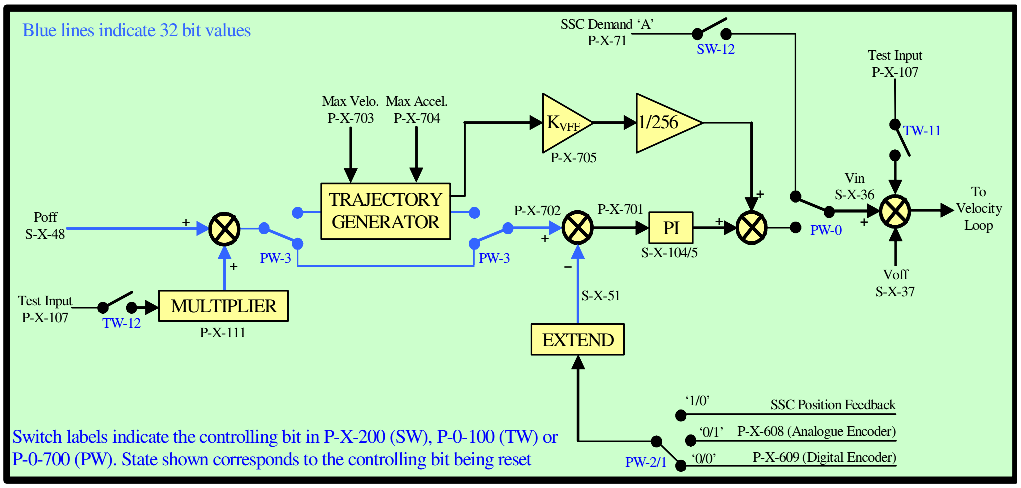

Figure 15 – Block Diagram Of The Position Loop

Encoder Reference Marks and Aries TTP

Each digital scale input has a reference mark input associated with it. These can be set to latch the counter value of the associated quadrature inputs. In addition, the Aries PQR hardware interfaces to a Touch Trigger Probe (TTP) input which can be used to latch the count values of all three axes (i.e. P, Q and R). Note that the reference mark inputs are not implemented in SPAlite hardware.

The inputs (reference marks and TTP) can be individually configured to latch on either or both edges (i.e. rising or falling) or can be disabled using the appropriate 'Edge Control IDN' (P-X-691 for reference marks and P-0-692 for TTP, see page 126). Each input has also has a 'Status IDN' (P-X-693 for reference marks and P-0-694 for TTP, see page 127) which an be read to determine whether a latch has occurred.

Note that there is no distinction made in the hardware between a latch caused by a reference mark input or a latch due to the TTP input other than the fact that a TTP latches all three encoder counters. Therefore, a latch flag for the TTP is set if the all of the axis reference mark latched flags are set and so a TTP hit is indistinguishable from simultaneous hits on all three reference marks. Although the latter event is unlikely, it is recommended that only TTP or Reference marks are enabled at any one time.

A latch on either a reference mark or TTP causes the latch event flag to be set and the current value of the appropriate encoder counter(s) to be copied to a latched register. The flag is accessible via a bit in the SSC status byte (bit 12) or by P-0-690 (see page 126). The flag is read only, it can only be reset by resetting the status flag or flags of whatever caused the latch event. The latched count registers can be read using P-X-695 (see page 128), all three registers are updated for a TTP event.

Here is an example of how the interface could be used:

1. Configure the amplifier to respond to either TTP or reference mark and set the appropriate edge(s) with P-0-692 or P-X-691.
2. Clear all latched and overflow registers by writing zero to TTP status register (P-0-694)
3. Start moving.
4. Monitor the Latch Event bit (bit 12) in the SSC feedback (from all TAPs) or P-0-690.

5. As soon as a latch event occurs stop moving and interrogate P-X-693 and P-0-694 to see what caused the latch event. If TTP latched bit is set (bit 1 of P-0-694) it suggests that a TTP event caused the latch event, but it could also have been caused by simultaneous hits on all reference marks. Although this would be unlikely, it is recommended that only TTP or Reference marks are enabled at any one time.
6. If any of the overflow flags are set then declare an error.
7. Read the latched position for the appropriate axis from P-X-695, for a TTP event read latched position for all three axes.
8. Clear the latched bit and the Latch event flag by writing zero to the TTP status register (P-0-694) or individual reference mark status registers (P-X-693).

Data Gathering

A data gathering facility is built into the servo code. This allows a buffer of data (up to 49137 samples) to be taken and this can be shared between up to 15 sources. The rate at which the specified data is logged can be set from one to 65535 servo cycles (corresponds to 10 kHz to 0.15 Hz) and the gather can be triggered from one of the sources.

Use

To set up a gather, first send a list of memory locations to log to P-0-303. This list can be up to 15 locations long. Lists shorter than 15 locations should be terminated with a zero. Any valid 2812 memory location can be logged. The memory location for the most useful parameters can be found using P-0-308. Secondly, the gather log period must be written to P-0-301 and the nominal buffer length to P-0-307. For an un-triggered single shot log, this is all that needs to be done. The log is started by setting bit 0 of trigger word, P-0-302. When the log is complete, this bit will be reset by the SPA2.

To do a triggered log, the trigger threshold (P-0-304) and polarity (bit 4 of P-0-302) must also be set along with the desired trigger delay (P-0-306). The log is started by setting bits 0,1 and 3 of P-0-302. After the trigger has occurred and the data logged, bits 0-3 of P-0-302 will be reset by the SPA2. Note that the trigger source is always the first parameter given in the P-0-303 list.

When the log is complete, the gathered data can be found in P-0-300. For triggered logs, the data starts at the sample of the buffer indicated by P-0-305 and wraps around the end of the buffer. In order to decode this, the actual length of the buffer is required which will depend upon the nominal buffer length (given by P-0-307) and the number of parameters being logged.

See the relevant IDN descriptions for more details (Appendix C, starting page 95).

Operation

The triggering function can be thought of as a state machine with four states: DISABLED, PREARMED, ARMED and TRIGGERED. The machine will be in the DISABLED state until forced into the PREARMED state by the user setting bit 3 of P-0-302. At the same time the user must also enable gathering in wrap-around mode by also setting bits 0 and 1 and clear the Buffer Filled flag by resetting bit 5 (hence 0xB or 0x1B must be written to P-0-302). The state will remain PREARMED until the buffer has wrapped around at least once (indicated by the buffer filled flag) and the trigger source (always channel 1) has fallen below the trigger threshold (P-0-304), at which point the state changes to ARMED. The state remains ARMED until the source rises above the threshold at which point the state becomes TRIGGERED and the delay time (P-0-306) is loaded into a delay counter. This counter decrements every sample instant and the state remains TRIGGERED until the counter becomes zero. At this, point logging is stopped and the state changes back to DISABLED (lower four bits of P-0-302 are reset). Note that if bit 4 of P-0-302 was set then the source value is inverted before it is compared with the threshold. Flowcharts describing this operation are shown in Figure 27 and Figure 28 in Appendix A.

ADC Calibrate Function

The 2812's ADC suffers from offset and gain errors. These can be corrected in the SPA2 code by applying a fixed offset to the ADC results as they are read. (e.g. current feedback offsets are defined in P-X-41 and P-X-42 for phases A and B respectively, see page 64). While these offsets can be set manually, the code contains a calibration function that will automatically calculate the correct values. This function is controlled using P-0-214 (see page 80): setting this parameter to 1 will start the calibration function and when complete, the SPA2 sets the parameter back to 0. When set, the servo loop code calculates the mean value of all of the relevant ADC inputs over 16384 samples (thus the process takes 1.6384 seconds at the default servo rate). The results are the offsets to be applied and are stored in the relevant parameters. It is up to the user to ensure that all the ADC inputs are zero during calibration (e.g. amplifier disabled and no motor movement).

The inputs that read bus voltage and temperature cannot be calibrated since is not possible to ensure that the analogue inputs are zeroed during the process. These inputs are corrected using the mean value of the offsets found for other channels. The assumption being that at least some of the error will be common to all ADC channels. Whilst not perfect, this gives better results than no calibration at all. See the description of P-0-214 (page 80) for more details.

Later hardware versions (SPA2-2 and Aries) utilise a spare input to attempt an automatic ADC offset correction. A 1.5V reference voltage is feed into this spare input which ideally should convert to 0. The actual converted value is treated as a common offset and is subtracted from the conversion of all other ADC channels. This should reduce the overall amount of offset but does not replace the calibrate function as described above since it only corrects for common ADC offset and not differential offsets between channels. The process should be transparent to the user but it can be disabled by setting P-0-245 (see page 93) to 0. The value from the 1.5V ADC channel can be read using P-0-44 (see page 65).

Commutation Angle Set Up

As previously described, in brushless motor mode the initial value of commutation angle (CA) needs to be set prior to motion being possible. Reference [13] describes a number of methods for determining the initial value of commutation angle and a series of test conducted to assess suitability for inclusion in SPA2 firmware. The best overall method was found to be the Successive Approximation Method (SAM) the main algorithm of which is described below:

1. Disconnect the link between the encoder and the commutation calculator and set CA to 180°.
2. Enable servos.
3. A ramp (final value = P-0-653 and ramp time = 2 seconds) is applied to main current demand.
4. As soon as the axis has moved by more than a set threshold (0.27°) the axis should be disabled and the CA re-estimated.
5. A re-estimation consists of an offset being added or subtracted (depending upon direction of movement) to the current value of CA. The magnitude of this offset is halved every time so that estimated angle eventually converges with the actual angle. After re-estimation, return to step 2.
6. If the ramp completes with no movement then the axis is disabled and the algorithm terminates with the current angle estimate as the result. The link between the encoder and the commutation calculator is restored. If the very first ramp (with angle estimate = 180°) completes with no movement then the estimate is set to 90° and the algorithm re-started from step 2 – this is to overcome the '180° Problem' as described in reference [13].
7. The algorithm also terminates if the magnitude of the correction offset falls below 1°. This happens after 8 cycles.
8. A watch is keep on axis movement throughout the process. If the algorithm is still detecting movement after 8 cycles AND an overall movement of 5° or greater has occurred, the algorithm is

restarted from step 1. This is to prevent movement due to outside influences (e.g. cogging torque or forces due to cables) causing the algorithm to jump the wrong way and fail.

Before the main algorithm starts, the set up words of all axes in the amp are stored. The active axis is set in commutated current loop mode and all other axes are disabled.

After the main algorithm is complete, the axis is put into non-commutated current loop mode and phase currents are set in order to lock the axis to the angle found. A real commutation angle (+90°) is then set. All set up words are then restored.

If a fault state is detected at any time during the algorithm, the process is immediately terminated.

The algorithm is implemented as a state machine in the SPA2 firmware which is called once during the main while loop. Figure 16 shows the state transition diagram and a short description of each state is given below

The state machine will sit permanently in the DISABLED state until the process is started by externally forcing the state variable to GENRATE RAMP. This is done by writing 1 to P-0-662. When the process is complete P-0-662 will return to zero, signifying the DISABLED state. Prior to starting the algorithm in this way, the axis for which commutation is to be set should be written to P-0-663.

This algorithm will be adequate for the vast majority of cases. If greater accuracy is required the Commutation Angle Servo can be enabled at the end of a search. See IDN P-0-655 for details.

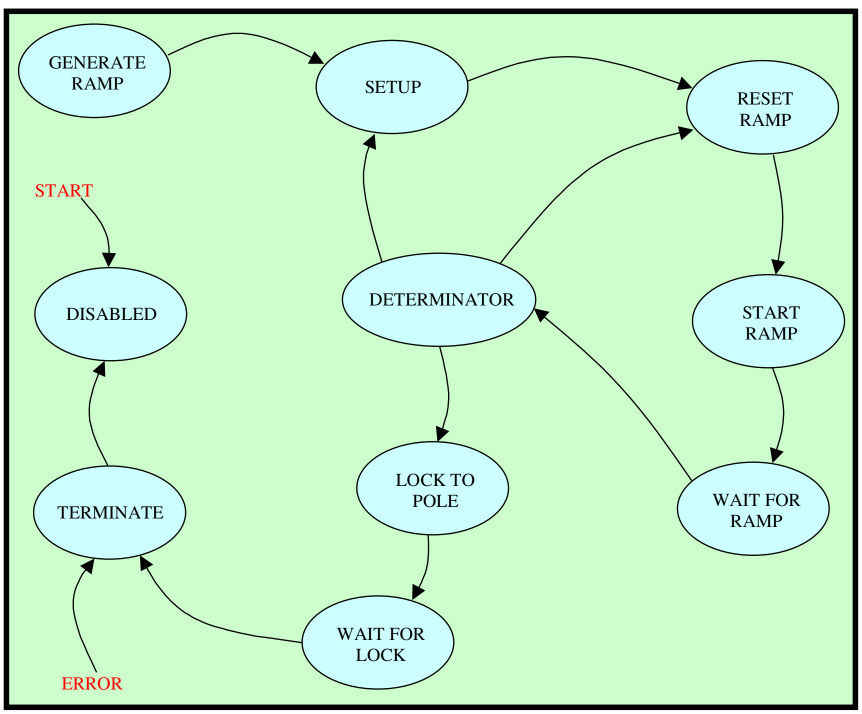

Figure 16 – State Machine For SAM Implementation

DISABLED

This state does nothing

GENERATE RAMP

In this state:

- The signal generator's Input Data Buffer (IDB) is loaded with a ramp that is 2 seconds long and has a max value as set in P-X-653.
- The IDB is set in single shot mode.
- The set up words of all axes and the software enable state of the DSP are stored and the amplifier is put into the required configuration.
- A number of parameters that control the algorithm are calculated
- The next state is always SETUP

SETUP

In this state:

- Store start encoder value (so that overall movement can be calculated at the end)
- Set initial CA estimate to 180° and correction offset to 90°
- Set the Commutation Disconnect flag
- The next state is always RESET RAMP

RESET RAMP

In this state:

- Reset the signal generator
- Update real CA with estimate
- Enable drive
- The next state is always START RAMP

START RAMP

Remain in this state until no change is detected in the encoder input and then:

- Enable code in main servo loop that detects motion in the axis (the movement sensor)
- Trigger the signal generator (starts the ramp)
- Go to the next state: WAIT FOR RAMP

WAIT FOR RAMP

Remain in this state until the drive disables (caused by movement sensor being triggered) or the ramp completes. In either case we go to the next state which is DETERMINATOR. If the ramp is completed we do the following first:

- Reset commutation disconnect flag
- Disable Drive
- Disable movement sensor

DETERMINATOR

It is this state that implements the decision making process of the algorithm. We remain in this state until no change is detected in the encoder input and then proceed according to the following:

IF Ramp Completed AND Estimated CA is 180°:

> This means that we have not moved anywhere! Try again with initial estimate of 90° in case we are experiencing the '180° problem':
>
> - Set CA Estimate to 90°
> - Set the Commutation Disconnect flag
> - Next state is RESET RAMP

IF Ramp Completed AND Estimated CA is not 180°:

> This means main algorithm is complete, next state is LOCK TO POLE

IF Ramp Not Completed AND Correction Offset greater than 1°:

> - Re-estimate CA
> - Half correction offset
> - Next state is RESET RAMP

IF Ramp Not Competed AND Correction Offset less than 1°:

- Calculate overall movement during algorithm
- If overall movement less than 5° then finish, next state is LOCK TO POLE
- If overall movement greater than 5° then algorithm probably compromised, start all over again, next state is SETUP

LOCK TO POLE

In this state:

- Reset commutation disconnect flag
- Change axis to Non-commutated current loop mode
- Set phase current demands according to CA
- Enable Drive
- Next state is always WAIT FOR LOCK

WAIT FOR LOCK

Remain in this state until no change is detected in the encoder input and then:

- Set CA to value previously found plus 90°
- Disable drive
- Remove phase current demands
- Go to next state: TERMINATE

TERMINATE

In this state:

- Reset the commutation disconnect flag
- Ensure movement sensor is disabled
- Disconnect the signal generator
- Restore all axis settings and amplifier software enable setting
- The next state is always DISABLED

APPENDIX A – FLOWCHARTS AND STATE DIAGRAMS

Serial Communications

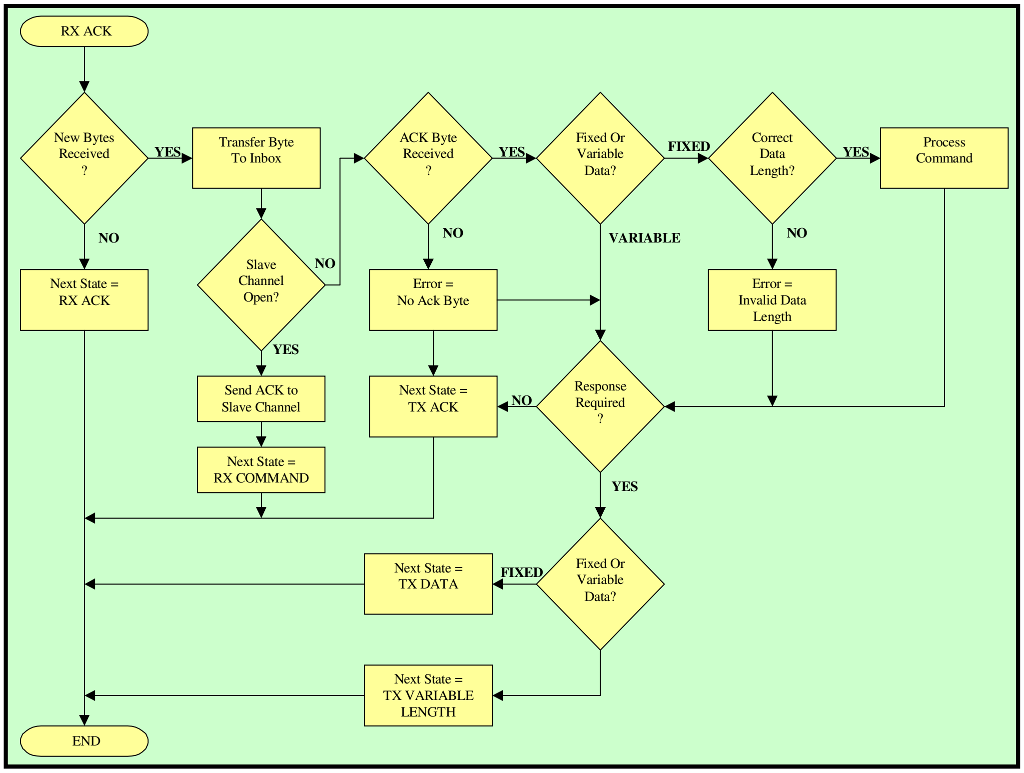

Figure 17 – Flowchart for RX ACK

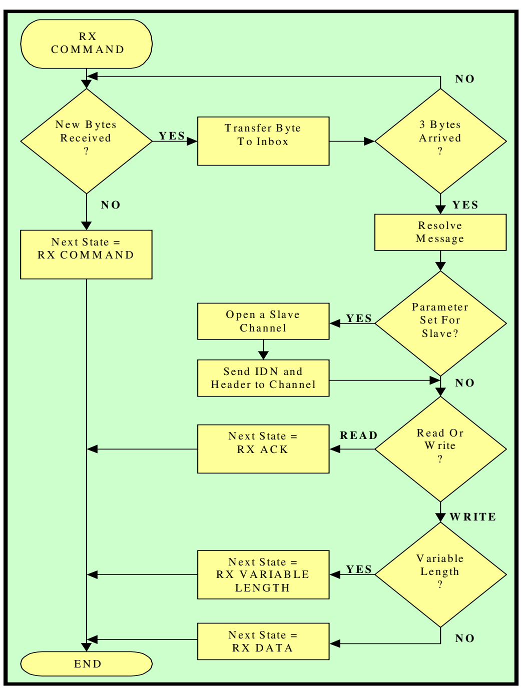

Figure 18 – Flowchart for RX COMMAND

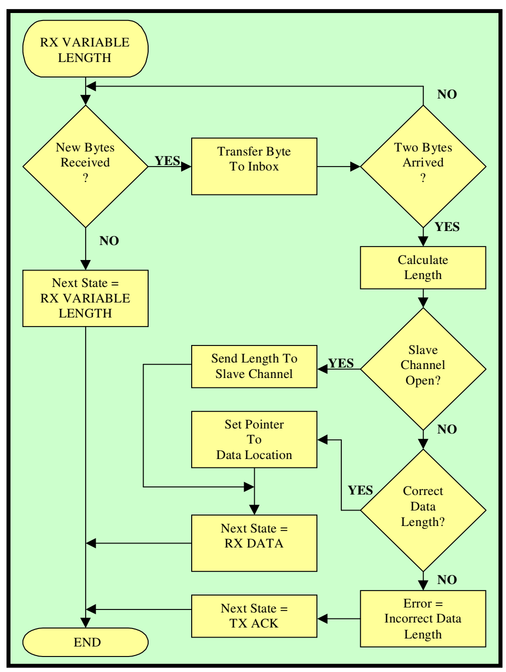

Figure 19 – Flowchart for RX VARIABLE LENGTH

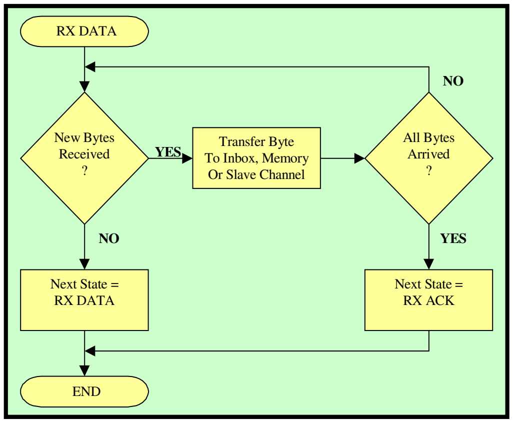

Figure 20 – Flowchart for RX DATA

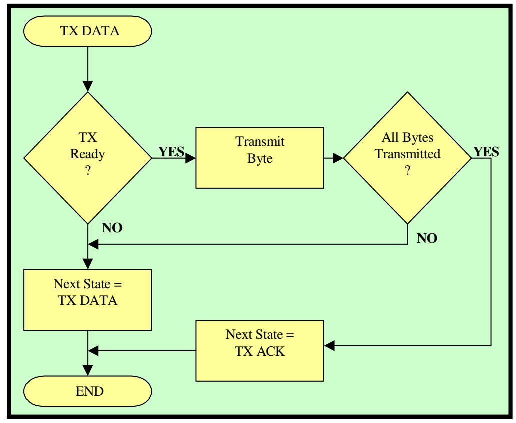

Figure 21 – Flowchart for TX DATA

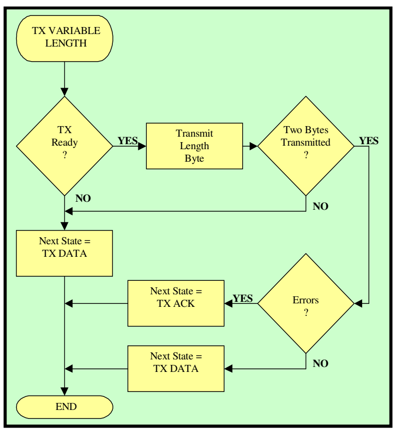

Figure 22 – Flowchart for TX VARIABLE LENGTH

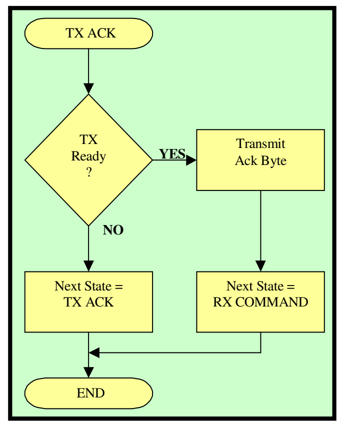

Figure 23 – Flowchart for TX ACK

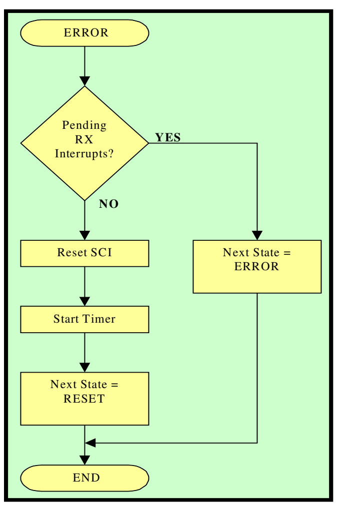

Figure 24 – Flowchart for ERROR

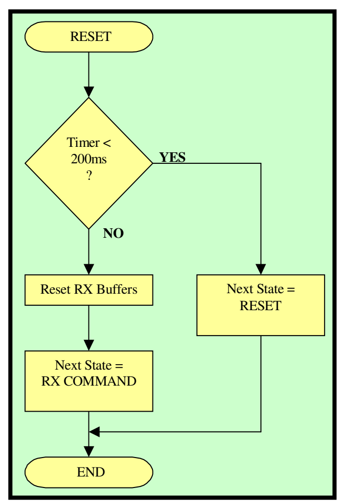

Figure 25 – Flowchart for RESET

Servo Code

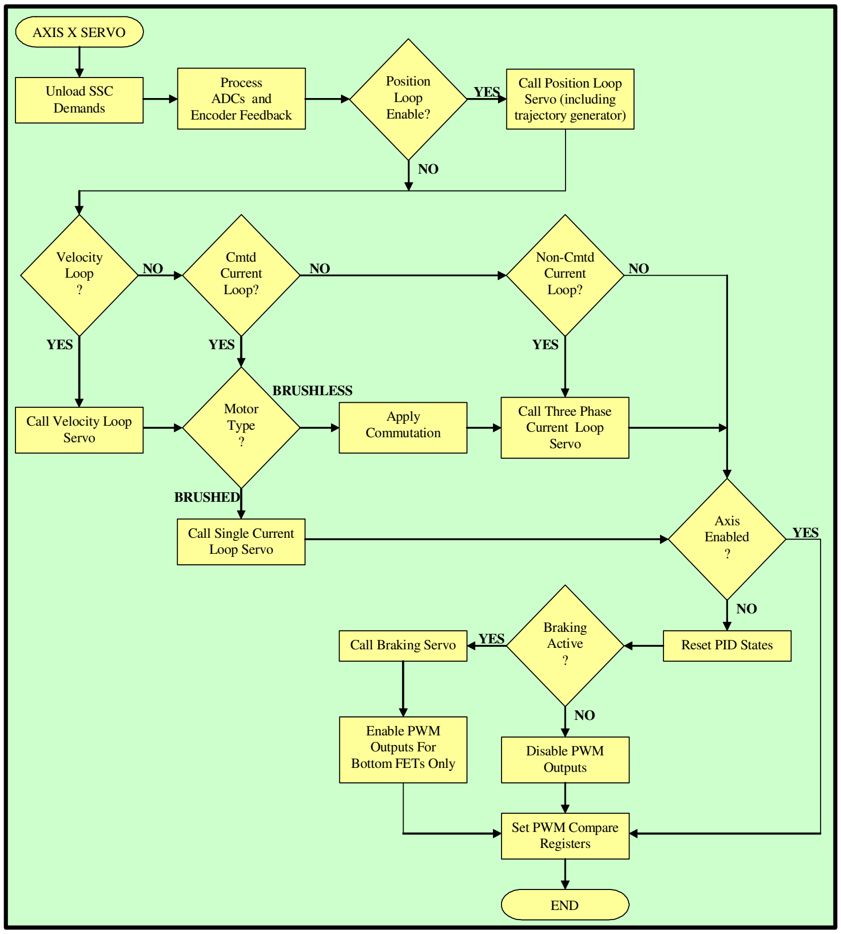

Figure 26 – Flowchart For A Single Axis Servo

Data Logging

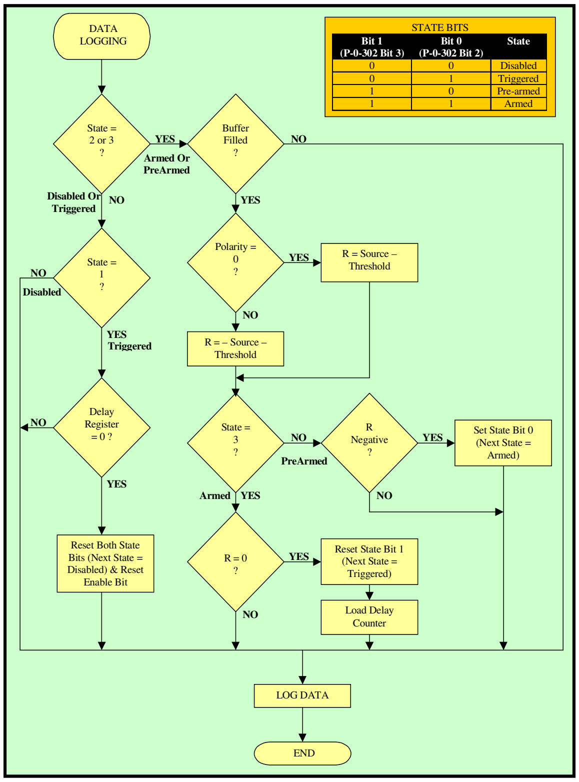

Figure 27 – Triggering State Machine

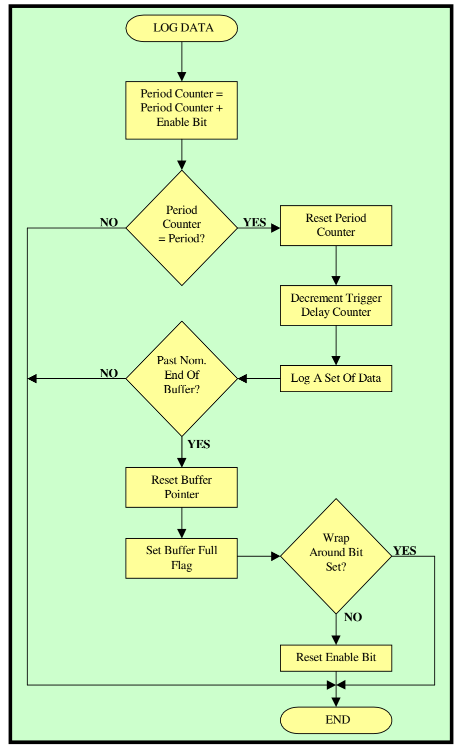

Figure 28 – Log Data

APPENDIX B – DSP MEMORY MAP

The 2812 DSP contains 18K words of on chip SARAM. This is divided into a block of 8K (H0), two blocks of 4K (L0 and L1) and two blocks of 1K (M0 and M1). In addition there is 128K words of flash (six blocks of 16K and four blocks of 8K) and provision for up to 1056K of external memory.

For SPA2, there are two modes that the code can run in:

> RAM mode: during development the code can be loaded directly into RAM using an emulator connected to the JTAG port of the DSP. This mode uses the memory map defined in the 'SPA2_Ram.cmd' linker file and the code must be compiled with the global symbol 'RUN_FROM_RAM=1'.
>
> Flash mode: In normal standalone operation, the code is stored in flash and copied to RAM on DSP boot. This mode uses the memory map defined in the 'SPA2_Flash.cmd'linker file and the code must be compiled with the global symbol 'RUN_FROM_RAM=0'.

There are 64K words of external RAM implemented for SPA2. This is used for data gathering and the input waveform buffer and is located in zone 6 of the DSP's EMIF (see reference [6] for details of the EMIF and page 5 for how it is set up). RAM L0 is used for data storage and the program code is run from H0 and L1. Since more space is required for code than is for data, L0 and L1 have been redefined for the purposes of SPA2 as being 2K and 6K words respectively. Flash is used to store the program code and most of this is copied to H0 and L1 upon start up by the code in 'entry.asm'. Some code is stored AND run from flash to save space in RAM. The access time for flash is slower than RAM (by a factor of five), so only functions that are infrequently accessed are run in this way. When running in RAM mode, this flash code is run in external memory which means that the whole of external memory is not available for data. Therefore the following restrictions apply when running in RAM mode:

- Maximum number of samples for data logging is reduced to 32753
- A memory test (see P-0-998) does not cover the whole of the external memory and the error code returned will have bit 15 set.

Certain amplifier parameters are also stored in flash and used for initialisation by the set up code. Flash is also used to store a look up table for the arctan function, the torque compensation look up tables and the Aries FPGA program. Interface to the Aries FPGA is via zone 0 of the DSP's EMIF.

Table 21 shows a summary of the memory sections used and how they are allocated when running in the two different modes and Table 22 summarises the usage of the Flash. Memory usage defined here is for SPA2 firmware V5.0.0.

Section Name	Purpose	Size (words)	RAM Mode	Flash Mode
.text	Main Program Code	7942	H0	FLASH C → H0
Text2	Main Program Code	3212	L1	FLASH C → L1
TextFlash	Main Program Code	7996	XINTF Zone 6	FLASH D
FbootLoad	Entry Code	43	-	FLASH A
.switch	C Switch Tables	228	L1	FLASH C → L1
.stack	The C Stack	1024	M1	M1
.cinit	C initialisation	417	XINTF Zone 6	FLASH D
.econst	C Constants	730	XINTF Zone 6	FLASH D
FlashAPIeconst	.econst From Flash API	50	-	FLASH C → H0
.ebss	Data	1567	L0	L0
.esysmem	The C Heap	100	L0	L0
AtanTable	Arctan Look Up Table	8192	FLASH E	FLASH E
TorqueTable	Torque Compensation	774 (default)	FLASH J	FLASH J
SavedParamMemory	Saved Parameters	428	FLASH B	FLASH B
DataLogBuff	Logging and Test I/P	65536*	XINTF Zone 6	XINTF Zone 6
FPGARegsFile	Aries FPGA I/F	64	XINTF Zone 0	XINTF Zone 0
-	Aries FPGA Program	14780	FLASH G & H	FLASH G & H

Table 21 – Memory Map Summary

* 49152 for RAM Mode

Sector	Size	Address	SPA2 Use
J	8kw	0x3D8000	Torque Compensation Tables
I	8kw	0x3DA000	Not Used
H	16kw	0x3DC000	Aries FPGA Program
G	16kw	0x3E0000	Aries FPGA Program
F	16kw	0x3E4000	Not Used
E	16kw	0x3E8000	Arctan Look Up Tables
D	16kw	0x3EC000	Run From Flash Program
C	16kw	0x3F0000	Run From Ram Program
B	8kw	0x3F4000	Saved Parameters
A	8kw	0x3F6000	Boot load Boot to Flash Entry Point Security Password

Table 22 – SPA2 Flash Usage

APPENDIX C – SUPPORTED SERCOS PARAMETERS

SERCOS is a standard for real time serial communications for motion controllers. It is a master/slave synchronous protocol based on the sending and receiving of telegrams at regular clock intervals. The content of these messages is organised into a system of parameters which are all given identity numbers (or IDNs). The serial communication protocol used in the SPA2, as described in the main part of this document (starting on page 13), differs significantly from the SERCOS standard in many ways. However, the same system of IDNs is used to organise the data that can be read from or written to the amplifier.

IDNs are 16 bit numbers which are divided into two ranges, S and P, with 32768 parameters in each range. Each range is further divided into 8 parameter sets of 4096 parameters. S parameters are predefined by the SERCOS standard and include generic motion control parameters like controller gains and limits. P parameters are undefined in the standard and so are free to be assigned by the application. In SPA2, S parameters are used for those values that have appropriate IDNs defined and P parameters have been created for those that do not. Each axis on SPA2 corresponds to a different parameter set in the IDNs, axes 1 and 2 correspond to parameter sets 0 and 1 respectively. Parameters that apply to the controller as a whole rather than to a specific axis (e.g. controller firmware version) are defined in parameter set zero. Table 23 shows how the 16 bit IDN is formed from its component parts.

15	14	13	12	11	10	9	8
Range P=1 or S=0	Parameter Set 0 – 7			High Four Bits Of Parameter Number			
7	6	5	4	3	2	1	0
Low Eight Bits Of Parameter Number							

Table 23 – Format Of SERCOS IDN

Throughout this document (and other SERCOS literature) the notation, *R-P-XXXX*, is used to refer to an IDN. *R* is the range (either S or P), *P* is the parameter set or axis (0–7) and *XXXX* is the parameter number (in the range 0–4095). Sometimes the parameter set number is replaced by an X, this indicates that the IDN is defined for all axes and the X can be replaced accordingly. Here are a couple of examples:

- S-0-30 means parameter 30, for parameter set 0 (axis 1), in the S range. The actual IDN is 30.
- P-X-200 means parameter 200, of all parameter sets, in the P range. Actual IDNs are: 32968 and 37064 for axes 1 and 2 respectively.

Table 24 shows a summary of the supported SERCOS parameters. The rest of this appendix describes each of the supported IDNs in detail.

IDN	R/W	SIZE	DESCRIPTION	VAR NAME
MISCELLANEOUS				
S-0-30	R	V2‡	Firmware Version	Version
P-0-0	R	V2	Amplifier Type (Simple)	AmpType
P-0-1	R	V2	Amplifer Type (Specific)	AmpType
P-0-219	R	2	Modified Hardware Version	BodgedHardwareVersion
P-0-246	R	2	Unmodified Hardware Version	HardwareVersion
P-0-212	R	2	Time Spent in Servo Routine	ServoTime
P-0-996	R & W	2	Sensitive Param. Protection Lock	SPPKey
P-0-40	R & W	2	Common ADC Offset	CommonADCOffset
P-0-2	R	2	Amplifier Minimum Bus Voltage	-
P-0-3	R	V2	Amplifer Current Ranges	-
P-0-44	R	2	ADC 1.5V Reference Input	ADCReference
P-0-245	R & W	2	ADC Auto Correct Flag	AutoCorrectADC
TEST GENERATOR				
P-0-100	R & W	2	Test Input Set Up Word	TestSetup
P-0-101	R & W	2	Test Square Wave Period	TestPeriod
P-0-102	R & W	2	Test Square Wave High Amplitude	TestHigh
P-0-103	R & W	2	Test Square Wave Low Amplitude	TestLow
P-0-104	R & W	V2	Test Input Waveform	InBuff
P-0-105	R & W	2	Input Waveform Sample Time	InPeriod
P-0-106	R & W	2	Test Input Waveform Length	WrapLen
P-0-109	R & W	2	Test Phase Offset	PhaseOffset
P-0-110	R & W	2	Input Buffer Pointer	ptrInData – InBuff
P-0-107	R	2	The Test Input	Tin
P-0-108	R	2	Phased Test Input	TinPhase
P-0-111	R & W	2	Position Input Multiplier	TestPosMultiplier
SERVO SETUP				
P-X-200	R & W	2	Servo Option Switches	SA[X].Options
P-0-247	R & W	2	Servo Interrupt	ServoPeriod
POSITION LOOP				
P-X-700	R & W	2	Position Loop Options Word	SA[X].PosOptions
S-X-77	R & W	2	Position Loop Scaling Factor	SA[X].Ppi.Ks
S-X-104	R & W	2	Position Loop Proportional Gain	SA[X].Ppi.Kp
S-X-105	R & W	2	Position Loop Integral Gain	SA[X].Ppi.Ki
S-X-48	R & W	4	Position Offset	SA[X].Pos
P-X-702	R	4	Position Combined Demand	SA[X].Pcd
S-X-51	R & W	4	Position Feedback	SA[X].Position
P-X-701	R	2	Position Loop Error	SA[X].Perr
P-X-703	R & W	2	Trajectory Calculator Max Velocity	SA[X].TrajCalc.MaxVelo
P-X-704	R & W	2	Trajectory Calculator Max Accel.	SA[X].TrajCalc.MaxAccel
P-X-705	R & W	2	Velocity Feed-Forward Gain	SA[X].Kvff
VELOCITY LOOP				
S-X-45	R & W	2	Velocity Loop Scaling Factor	SA[X].Vpid.Ks
S-X-100	R & W	2	Velocity Loop Proportional Gain	SA[X].Vpid.Kp
S-X-101	R & W	2	Velocity Loop Integral Gain	SA[X].Vpid.Ki
S-X-102	R & W	2	Velocity Loop Derivative Gain	SA[X].Vpid.Kd
P-X-505	R & W	V2	Velocity Derivative Term Filter	SA[X].Vdfilter
P-X-618	R & W	2	Acceleration Derivation Gain	SA[X].Kacc
P-X-504	R & W	V2	Acceleration Derivation Filter	SA[X].Afilter
P-X-612	R & W	2	Acceleration Feed-Forward Gain	SA[X].Kaff
P-X-613	R & W	2	Friction Offset	SA[X].Kfric
P-X-614	R & W	2	AFO Acceleration Threshold	SA[X].Kat_AFO
P-X-615	R & W	2	AFO Decay Time Constant	SA[X].Ktc_AFO
P-X-616	R & W	2	AFO Offset	SA[X].Kos_AFO

IDN	R/W	SIZE	DESCRIPTION	VAR NAME
P-X-604	R & W	2	Velocity Loop Decimation Factor	SA[X].Vdec
P-X-633	R & W	2	Coupled Mass Gain	SA[X].Kcm
P-X-500	R & W	V2	Velocity Feedback IIR Filter	SA[X].VfilterReal
P-X-501	R & W	V2	SSC Velocity Feedback IIR Filter	SA[X].VfilterSSC
P-X-502	R & W	V2	Velocity Forward Path IIR Filter 1	SA[X].VfilterForward[1]
P-X-503	R & W	V2	Velocity Forward Path IIR Filter 2	SA[X].VfilterForward[2]
S-X-36	R	2†	Velocity Demand	SA[X].Vdm
P-X-617	R	2	Acceleration Demand	SA[X].Adm
S-X-37	R & W	2†	Additive Velocity Demand	SA[X].Vos
P-X-230	R & W	2	Velocity Demand Polarity	SA[X].Vpol
S-X-40	R	2†	Velocity Feedback	SA[X].Vfb
P-X-610	R	2	Real Velocity Feedback	SA[X].RealVfb
P-X-630	R	2	SSC Velocity Feedback	SA[X].RawSSCVfb
P-X-631	R	2	Filtered SSC Velocity Feedback	SA[X].SSCVfb
P-X-611	R & W	2	SSC Velocity Scale Factor	SA[X].Kvssc
P-X-632	R	2	Velocity Difference (Real & SSC)	SA[X].Vdiff
S-X-347	R	2†	Velocity Error	SA[X].Verr
TACHO FEEDBACK				
P-X-43	R & W	2	Tacho ADC Offset	ADCOffset[X*3 + 2]
P-X-620	R	2	Tacho Tracker Error	SA[X].TachoError
P-X-621	R	2	Tacho Tracker DAC output	SA[X].DACOutput
P-X-622	R	2	Tacho Tracker Output	SA[X].Tacho
P-X-623	R & W	2	Tacho Velocity Trim Scale Factor	SA[X].Kvt
P-X-510	R & W	V2	Tacho Filter	SA[X].TachoFilter
ENCODER FEEDBACK				
P-X-606	R	2	Analogue Encoder Sine Channel	SA[X].AESin
P-X-607	R	2	Analogue Encoder Cosine Channel	SA[X].AECos
P-X-508	R & W	V2	Analogue Encoder Sine Filter	SA[X].AEfilter[0]
P-X-509	R & W	V2	Analogue Encoder Cosine Filter	SA[X].AEfilter[1]
P-X-608	R	2	Analogue Encoder Counts	SA[X].AEEncoder
P-X-609	R	2	Digital Encoder Raw Counts	DigEncoder
P-X-600	R	2	Raw Encoder Counts	SA[X].RawEncoder
P-X-601	R	2	Differential Of Encoder Counts	SA[X].DeltaEncoder
P-X-602	R & W	2	Encoder Velocity Scale Factor	SA[X].Kve
P-X-670	R & W	2	Heydemann Compensation – p	SA[X].Heydemann.p
P-X-671	R & W	2	Heydemann Compensation – q	SA[X].Heydemann.q
P-X-672	R & W	2	Heydemann Compensation – G	SA[X].Heydemann.G
P-X-673	R & W	2	Heydemann Compensation – sin(a)	SA[X].Heydemann.SinA
P-X-674	R & W	2	Heydemann Comp. – 1/cos(a)	SA[X].Heydemann.OneOverCosA
P-X-675	R & W	V2	Lissajous Correction LUT	LissajousMap[X]
P-X-605	R	2	Encoder Reference Mark	SA[X].EncoderRefMark
P-0-690	R	2	Latch Event Flag	LatchEventFlag
P-X-691	R & W	2	Reference Mark Edge Control	SA[X].RefMarkEdgeControl
P-0-692	R & W	2	TTP Edge Control	TTPEdgeControl
P-X-693	R & W	2	Reference Mark Status	SA[X].RefMarkStatus
P-0-694	R & W	2	TTP Status	TTPStatus
P-X-695	R & W	2	Latched Event Value	SA[X].LatchedEncoder
COMMUTATION				
P-X-680	R & W	2	Commutation Mode	SA[X].CommutationMode
P-X-650	R & W	2	Commutation Angle	SA[X].ScaledCommAngle
P-X-651	R & W	4	Encoder Counts Per Revolution	SA[X].CountsPerRev
P-X-652	R & W	2	Number Of Pole Pairs Per Rev.	SA[X].NoOfPoles
P-X-681	R & W	2	Hall Offset	SA[X].ScaledHallOffset
P-X-682	R	2	Hall Code	SA[X].HallCode

IDN	R/W	SIZE	DESCRIPTION	VAR NAME
P-X-683	R	2	Hall Angle	SA[X].HallAngle
P-X-684	R & W	2	Hall Direction	SA[X].HallDirection
P-X-654	R & W	2	Commutation Set Flag	CmtnSet[X]
P-X-653	R & W	2	Commutation Setup Offset	CmtnSetupOffset[X]
P-0-662	R & W	2	SAM State Variable	CommutationSetState
P-0-663	R & W	2	SAM Active Axis	ActiveAxis
P-X-658	R & W	2	Commutation Disconnect Flag	SA[X].CommutationDisconnect
P-X-659	R & W	2	Movement Sensor Threshold	SA[X].SAM_MoveThreshold
P-X-660	R & W	2	Movement Sensor Moved Amount	SA[X].SAM_Movement
P-X-655	R & W	2	CAS Flag	SA[X].CAS_Flag
P-X-656	R & W	2	CAS I Gain	SA[X].CAS_Igain
P-X-661	R & W	2	CAS P Gain	SA[X].CAS_Pgain
P-X-657	R & W	2	CAS Output	SA[X].CAS_IntOutput
CURRENT LOOP				
S-X-93	R & W	2	Current Loop Scaling Factor	SA[X].Ipid[*].Ks
S-X-106	R & W	2	Current Loop Proportion Gain	SA[X].Ipid[*].Kp
S-X-107	R & W	2	Current Loop Integral Gain	SA[X].Ipid[*].Ki
P-X-506	R & W	V2	Phase A Current Feedback Filter	SA[X].Ifilter[0]
P-X-507	R & W	V2	Phase B Current Feedback Filter	SA[X].Ifilter[1]
S-X-80	R	2	Main Current Demand	SA[X].Tdm
S-X-81	R & W	2	Main Additive Current Demand	SA[X].Tos
P-X-229	R & W	2	Main Current Demand Polarity	SA[X].Tpol
P-X-640	R	2	SSC Current Offset	SA[X].Tos_ssc
P-X-641	R	2	Torque Compensation Offset	SA[X].Tcorr
P-X-642	R & W	V2	Torque Compensation Map	TorqueCorrectionLUT
P-X-643	R & W	2	Flash Torque Compensation Map	TCLUTFlashWritingError
P-0-644	R & W	2	Size of Torque Compensation Map	TCMapN
P-X-1Y	R	2	Phase A & B I Demand	SA[X].Idm[Y-1]
P-X-2Y	R & W	2	Phase A & B Additive I Demand	SA[X].Ios[Y-1]
P-X-3Y	R	2	Ph. A, B & C Compensated I F/B	SA[X].Ifb[Y-1]
P-X-30	R	2	Square Of Total Axis Current	SA[X].Isquared
P-X-4Y	R & W	2	Phase A & B Current ADC Offset	ADCOffset[X*3 + Y-1]
P-X-5Y	R	2	Phase A, B & C Current Error	SA[X].Ierr[Y-1]
ACTIVE BRAKING				
P-0-223	R & W	2	Braking Mode (Timed or Host)	TimedBrakingMode
P-0-222	R & W	2	Time Spent In Braking State	BrakeDuration
P-0-224	R & W	2	Host Controlled Brake Flag	Brake
P-X-235	R & W	2	Braking Current	SA[X].BakingCurrent
PWM				
P-X-6Y	R & W	2	Phase A, B & C PWM Output	SA[X].PWMOut[Y-1]
P-X-225	R & W	2	Axis Maximum Duty Cycle	SA[X].VbusScale
P-0-203	R & W	2	PWM Period	EvaRegs.T1PR
P-0-218	R & W	2	PWM Dead Band Time	DBT
LIMITS AND FAULTS				
S-X-129	R & W	2	Manufacturer Class 1 Diag.(Low)	C1DiagnosticsLatched[X]
P-X-129	R & W	2	Manufacturer Class 1 Diag.(High)	C1DiagnosticsLatched[X]
S-X-111	R & W	2†	Motor Continuous Current	SA[X].I2Tthreshold
P-X-204	R & W	2	Peak Current Max Time	SA[X].I2Tlimit
P-X-205	R & W	2	Current Error Time Limit	SA[X].IETLimit
P-X-220	R & W	2	Velocity Limit Time	SA[X].VeloLimitTime
P-X-236	R & W	2	Velocity Error Time Limit	SA[X].VETLimit
P-0-226	R & W	2	Expected Bus Voltage Input	VbusInput
P-X-227	R	2	Scaled Bus Voltage Input	ScaledVbus[X]
P-0-248	R & W	2	Vbus Fault Tolerance	VbusFaultTolerance

IDN	R/W	SIZE	DESCRIPTION	VAR NAME
P-0-995	R & W	2	Vbus Relay Control	BusEngage
P-X-207	R & W	2	Analogue Gain Settings	SA[X].AGS
P-0-211	R	2	Enable State	EnableState
P-0-206	R & W	2	Software Enable	SWEnable
P-0-213	R & W	2	Qualification Period for E-Stop OK	EstopOKQualPeriod
P-0-249	R & W	2	Qual. Period for PWM Fault Input	PWMFaultQualPeriod
P-0-231	R	2	Ambient Temperature	AmbientTemperature
P-0-232	R	2	Heater Temperature	HeaterTemperature
P-0-233	R & W	2	Max Temperature	MaxTemperature
P-0-234	R & W	2	Max Temperature Difference	MaxTempDifference
P-0-243	R	2	Pulse Input From Fan	FanPulse
P-0-244	R & W	2	Threshold Of Fan Pulse Fault	FanPulseThreshold
SPECIAL INSTRUCTIONS				
P-0-214	R & W	2	Calibration Mode Flag	Calibrate
P-0-208	R	2	Save Parameters to Flash	SaveStatus
P-0-400	W	V2	Reprogram Flash	-
P-0-401	R	2	Reprogram Flash Error Code	FlashReprogramError
P-0-402	W	V2	FPGA Program Download	-
P-0-403	R & W	2	Flash FPGA Program	FlashReprogramError
P-0-998	R	2	External Memory Test	MemTestStatus
P-0-999	W	2	Trip The Watchdogs	-
DATA LOGGING				
P-0-300	R & W	V2	Data Gather Buffer	OutBuff
P-0-301	R & W	2	Data Gather Period	GatherPd
P-0-302	R & W	2	Data Gather Trigger Flags	GatherTrigger
P-0-303	R & W	V2	Data Gather Parameter List	GatherLocs
P-0-304	R & W	2	Data Gather Trigger Threshold	GatherThres
P-0-305	R & W	2	Data Gather Buffer Start Position	(GatherPtr – OutBuff)
P-0-306	R & W	2	Data Gather Trigger Delay	GatherDelay
P-0-307	R & W	2	Gather Nominal Buffer Length	GatherLen
P-0-308	R & W	2	Memory Location Look Up Table	LoggingIDN
COMMUNICATIONS				
P-0-209	R & W	2	Serial Comms Baud Rate Register	RS232Baud
P-0-210	R & W	2	Serial Comms Control Register	RS232Settings
P-0-216	R & W	2	Qual. Period for SSC Enable	EnableQualPeriod
P-0-215	R & W	2	SSC Time Out	CommTimeOut
P-0-217	R & W	2	SSC Time Out Override	CommTimeOutOverride
P-0-997	R & W	2	Comm Lock Release Key	CommLockKey
P-0-221	R & W	2	SSC Enable Override	CommEnableOverride
P-0-228	R & W	2	SSC Checksum Enable	CheckSumEnabledElect
P-0-241	R & W	2	Checksum Failure Counter	CSFailCount
P-X-238	R & W	2	SSC Feedback Mode	SA[X].SSC_FB_Mode
P-X-71	R	2	SSC Demand ‘A’	SSCDemands[X][0]
P-X-72	R	2	SSC Demand ‘B’	SSCDemands[X][1]
P-X-242	R & W	2	SSC Demand 'B' Selector	SA[X].SSCDemandBSelector
P-X-239	R & W	2	SSC F/B, User Defined Address 1	SA[X].SSC_FB_Address[0]
P-X-240	R & W	2	SSC F/B, User Defined Address 2	SA[X].SSC_FB_Address[1]
P-0-350	R & W	2	SSC Master / Slave Switch	CommNode
P-0-351	W	2	Send Byte To Service Channel 1	-
P-0-352	W	2	Send Byte To Service Channel 2	-
P-0-353	W	2	Send Byte To Service Channel 3	-

Table 24 – Supported Sercos Parameters

Notes for Table 24:
SA refers to the ServoArguments structure

* indicates that setting applies to all phases
Y can be replaced with 1, 2 or 3 for phases A, B and C respectively
† SERCOS specifies these as 4 byte parameters.
‡ SERCOS specifies this as a variable one byte parameter

S-0-30 Firmware Version

Main Properties

Read Only
Two Byte Variable Length
Parameter set 0 only

Description

Returns an ASCII string (one character per word) containing firmware version.

The version number is made up of three numbers separated by periods. The third number increments every time a significant change is made to the code. The second number increments (and the third resets to zero) when the firmware has been verified by passing functional test on all hardware variants. The first number increments (and the other two reset to zero) when the code has been passed for use in production (verified by installation on a set of internal CMMs). Thus only code with a version number of the form X.0.0 should be shipped outside the company and code with a non-zero third number should be considered un-tested and treated with caution.

S-X-36 Velocity Demand

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter returns the velocity demand received via the SSC link (in velocity loop mode only). See Figure 13 for details of how the servo is structured.

S-X-37 Additive Velocity Demand

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0
Logable
Flash Saved

Description

This parameter defines the offset applied to the velocity loop. See Figure 13 for details of how the servo is structured.

S-X-40 Velocity Feedback Value

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter returns the feedback value used by the velocity servo. The value is only updated when in velocity loop mode. See Figure 13 for details of how the servo is structured.

S-X-45 Velocity Loop Scaling Factor

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 10
Flash Saved

Description

The KS shift factor for the velocity loop PID filter. See page 35 for more details.

S-X-48 Additive Position Demand

Main Properties

Read and Write
Four Byte Fixed Length
Parameter set indicates axis.
Default = 0
Flash Saved

Description

This parameter defines the offset applied to the position loop. See Figure 15 for details of how the servo is structured.

S-X-51 Position Feedback

Main Properties

Read Only
Four Byte Fixed Length
Parameter set indicates axis.
Logable (lower 16 bits are logged)

Description

This parameter returns the feedback value used by the position servo. See Figure 15 for details of how the servo is structured.

S-X-77 Position Loop Scaling Factor

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 10
Flash Saved

Description

The KS shift factor for the position loop PI filter. See page 35 for more details.

S-X-80 Main Current Demand

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

The returned value of S-X-80 will depend upon the servo loop mode. When in velocity loop mode, S-X-80 will return the output of the velocity servo filter. When in current loop mode, S-X-80 will return the current demand received via the SSC link. See Figure 13 for details of how the servo is structured.

The scaling of this parameter is proportional to the setting of I_{FSD} (see IDN P-X-207, page 74), where $\pm 32767 = \pm I_{FSD}$.

S-X-81 Additive Current Demand

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0
Logable
Flash Saved

Description

This parameter defines the current offset applied to the current loop. When in velocity loop mode, the value of S-X-81 is added to the output of the velocity servo filter. When in current loop mode, the value of S-X-80 is added to the current demand received via the SSC link. See Figure 13 for details of how the servo is structured. Scaling is the same as S-X-80.

S-X-93 Current Loop Scaling Factor

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 8
Flash Saved

Description

The KS shift factor for the current loop PI filters (three filters all use the same gains). See page 35 for more details.

S-X-100 Velocity Loop Proportional Gain

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 14000
Flash Saved

Description

The KP gain for the velocity loop PID filter. See page 35 for more details.

S-X-101 Velocity Loop Integral Gain

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 300
Flash Saved

Description

The KI gain for the velocity loop PID filter. See page 35 for more details.

S-X-102 Velocity Loop Derivative Gain

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0
Flash Saved

Description

The KD gain for the velocity loop PID filter. See Figure 13 for more details.

S-X-104 Position Loop Proportional Gain

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 10
Flash Saved

Description

The KP gain for the position loop PI filter. See page 35 for more details.

S-X-105 Position Loop Integral Gain

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0
Flash Saved

Description

The KI gain for the position loop PI filter. See page 35 for more details.

S-X-106 Current Loop Proportion Gain

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 180
Flash Saved

Description

The KP gain for the current loop PI filters (three filters all use the same gains). See page 35 for more details.

S-X-107 Current Loop Integral Gain

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 80
Flash Saved

Description

The KI gain for the current loop PI filters (three filters all use the same gains). See page 35 for more details.

S-X-111 Motor Continuous Current

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 8192 (¼ of I_{FSD})
Flash Saved

Description

This parameter sets the threshold above which the I^2t limit starts integrating. Any current below this amount will be allowed to flow in the load continuously. It is scaled relative to I_{FSD} (i.e. $\pm 32767 = \pm I_{FSD}$). Since the current demand is limited to 50% of I_{FSD} (I_{PEAK}), setting S-X-111 to a value greater than 50% (i.e. greater than 16384) will effectively disable the I^2t limit.

Note that changing this parameter will change the value of the Peak Current Max Time parameter (P-X-204, see page 73). P-X-204 must be reset if S-X-111 is changed.

S-X-129 Class 1 Diagnostics (Low Word)

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter contains indications of fatal faults and limits, i.e. those that cause a RED enable state. In SERCOS speak this is known as "Manufacturer Class 1 Diagnostics". Internally, the diagnostics word is actually a 32 bit long word, This parameter contains the lower 16 bits and P-X-129 contains the higher 16 bits. Table 25 shows the definition of all of the diagnostic bits (i.e. both P-X-129 and S-X-129). Note that the word is latched and so will still indicate a fault even if the cause has been removed. P-X-129 and S-X-129 can be reset by writing a 0 to them. See page 19 for more details on the different limits and faults.

P-X-129							
15	14	13	12	11	10	9	8
Not Used							
7	6	5	4	3	2	1	0
Not Used							FPGA Load Fault
S-X-129							
15	14	13	12	11	10	9	8
Watchdog Tripped	Comm Lock	Bus Charge Fault	Cmtn Not Set	Velocity Error Time	Fan Fault	Over Temp Fault	Vbus Relay Open
7	6	5	4	3	2	1	0
SSC Time Out	ESTOP Tripped	Over Velocity	Vbus Fault	Current Error Time	I^2t Limit	Peak Current	PWM Over Current

Table 25 – P-X-129 and S-X-129, Manufacturer Specific Class 1 Diagnostics

The definition of the individual bits is given below:

S-X-129, Bit 0 PWM Over Current
0 – External over current fault has not occurred since S-X-129 was last reset.
1 – External over current fault has occurred since S-X-129 was last reset.

S-X-129, Bit 1 Peak Current
0 – Internal over current fault has not occurred since S-X-129 was last reset.
1 – Internal over current fault has occurred since S-X-129 was last reset.

S-X-129, Bit 2 I^2t Limit
0 – I^2t Limit has not occurred since S-X-129 was last reset.
1 – I^2t Limit has occurred since S-X-129 was last reset.

S-X-129, Bit 3 Current Error Time Fault
0 – IET Limit has not occurred since S-X-129 was last reset.
1 – IET Limit has occurred since S-X-129 was last reset.

S-X-129, Bit 4 Vbus Fault
0 – Vbus has been within tolerance of expected value since S-X-129 was last reset.
1 – Vbus has been outside tolerance of expected value since S-X-129 was last reset.

S-X-129, Bit 5 Over Velocity Fault
0 – Over Velocity fault has not occurred since S-X-129 was last reset.
1 – Over Velocity fault has occurred since S-X-129 was last reset.

S-X-129, Bit 6 ESTOP Tripped
0 – ESTOP has not tripped since S-X-129 was last reset.
1 – ESTOP has tripped since S-X-129 was last reset.

S-X-129, Bit 7 SSC Time Out
0 – A SSC Time Out fault has not tripped since S-X-129 was last reset.
1 – A SSC Time Out fault has tripped since S-X-129 was last reset.

S-X-129, Bit 8 Vbus Relay Open
0 – The Vbus Relay has been closed since S-X-129 was last reset.
1 – The Vbus Relay has opened since S-X-129 was last reset.

S-X-129, Bit 9 Over Temperature Fault
0 – The ambient temperature has remained below Max Temp since S-X-129 was last reset.
1 – The ambient temperature has risen above Max Temp since S-X-129 was last reset.

S-X-129, Bit 10 Fan Fault
0 – Heater and ambient temp difference has remained with limit since S-X-129 was last reset.
1 – Heater and ambient temp difference has exceeded limit since S-X-129 was last reset.

S-X-129, Bit 11 Velocity Error Time Fault
0 – VET Limit has not occurred since S-X-129 was last reset.
1 – VET Limit has occurred since S-X-129 was last reset.

S-X-129, Bit 12 Commutation Not Set Fault
0 – The commutation set flag has remained reset since S-X-129 was last reset.
1 – The commutation set flag has been set since S-X-129 was last reset.

S-X-129, Bit 13 Bus Charge Fault
0 – Bus charge fault has not occurred since S-X-129 was last reset.
1 – Bus charge fault has occurred since S-X-129 was last reset.

S-X-129, Bit 14 Comm Lock Fault
0 – An SSC Time Out fault has not occurred since the Comm Lock key was last sent.
1 – An SSC Time Out fault has occurred since the Comm Lock key was last sent.

S-X-129, Bit 15 Watchdog Tripped
0 – Internal Watchdog has not tripped since S-X-129 was last reset.
1 – Internal Watchdog has tripped since S-X-129 was last reset.

P-X-129, Bit 0 FPGA Load Fault (Aries PQR only)
0 – No problem occurred during the configuration of the FPGA
1 – A problem occurred during the configuration of the FPGA, cannot be cleared.

P-X-129, Bits 1-15 Not Used

S-X-347 Velocity Error

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter will return the value of the velocity loop servo error, i.e. the difference between the velocity demand and feedback.

P-0-0 Amplifier Type (Simple)

Main Properties

Read Only
Two Byte Variable Length
Parameter set 0 only

Description

Returns an ASCII string (one character per word) containing a simple description of the amplifier type. This returns “SPA2” for all hardware variants except SPAlite for which

“SPAlite” is returned. This is included for backwards compatibility with previous firmware versions, P-0-1 gives more information and should be used in preference to this parameter.

P-0-1 Amplifier Type (Specific)

Main Properties

Read Only
Two Byte Variable Length
Parameter set 0 only

Description

Returns an ASCII string (one character per word) containing a description of the amplifier type. One of the following strings will be returned: “SPA2”, “SPAlite”, “SPA2-2”, “GYRO”, “ARIES-PQR”, “ARIES-ST” or “UNKNOWN”.

P-0-2 Amplifier Minimum Bus Voltage

Main Properties

Read Only
Two Byte Variable Length
Parameter set 0 only

Description

This parameter will return one of two values depending upon amplifier type. The value returned indicates the minimum bus voltage that that should be used with that hardware variant. The scaling is the same as used for P-0-266 and P-X-227 (i.e. 1V=491, see page 87). For SPAlite, 11784 is returned corresponding to 24V. For all other variants 5892 is returned corresponding to 12V.

Note that this parameter is supplied for information only, the firmware does not impose a minimum limit on the expected bus voltage (as set with P-0-226).

P-0-3 Amplifier Current Ranges

Main Properties

Read Only
Two Byte Variable Length
Parameter set 0 only

Description

This parameter will return a string of comma separated values that represent the I_{PEAK} values for the eight current ranges set with the analogue gain settings (see P-X-207 on page 74). There are three possible returns:

For SPA2 and SPA2-2 the returned string is “9.50,8.26,7.01,5.77,4.61,3.38,2.13,0.89”.

For SPAlite the returned string is “4.75,4.13,3.51,2.89,2.31,1.69,1.07,0.45”.

All other hardware variants do not have programmable analogue gains so the return from this parameter is “?.??,?.??,?.??,?.??,?.??,?.??,?.??,?.??”

P-X-1n Phase Current Demand (n = 1 & 2)

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

P-X-11 and P-X-12 will return the current demands to phases A and B respectively. The source of these demands depends upon the servo loop mode and configuration. These parameters are only updated in brushless motor mode. In non-commutated current loop the demands are received by the SSC link. In commutated current loop or velocity loop the demands are generated by commutation in the SPA2. See Figure 13 for details of how the servo is structured.

The scaling of these parameters is proportional to the setting of I_{FSD} (see IDN P-X-207, page 74), where ±32767 = ±I_{FSD}.

P-X-2n Phase Additive Current Demand (n = 1 & 2)

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0
Logable
Flash Saved

Description

P-X-21 and P-X-22 define the offset applied to the current inputs of phase A and B respectively. See Figure 13 for details of how the servo is structured. Scaling is the same as P-X-11.

P-X-30 Square Of Axis Total Current

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter returns the square of the total current in the motor. In brushed motor mode, this is simply the square of the current feedback. In brushless motor mode, it is given thus:

$$\frac{2 \times \left(Ia^2 + Ib^2 + Ic^2\right)}{3}$$

This value gives an indication of motor power and is used internally in the implementation of current limits.

P-X-3n Phase Current Feedback (n = 1, 2 & 3)

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

P-X-31, P-X-32 and P-X-33 return the currents for phases A, B and C respectively. The currents for phases A and B are derived from the on board current transducers and corrected for offset inside the DSP (by the addition of the values specified in P-X-41 and P-X-42).

Phase C current is derived from the other two phases ($Ic = -Ia - Ib$).

See Figure 13 for details of how the servo is structured.

P-0-40 Common ADC Offset

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 0
Flash Saved

Description

This parameter defines the offset that is applied to the ADC inputs for temperature and bus voltage sensing. This value can be manually entered or calculated using the calibrate function (see page 39 and IDN P-0-214 on page 80).

P-X-4n Phase Current ADC Offset (n = 1 & 2)

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0
Logable
Flash Saved

Description

P-X-41 and P-X-42 define the offset applied to the current feedback values of phase A and phase B respectively. This compensates the current feedback for ADC offset and gain errors. These values can be manually entered or calculated using the calibrate function (see page 39 and IDN P-0-214 on page 80). Scaling is the same as P-X-11.

P-X-43 Tacho Correction Offset

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0
Logable
Flash Saved

Description

This parameter defines the offset applied to the analogue tacho raw ADC reading. This compensates for gain and offset errors in the ADC and the tacho tracking DAC. This value can be manually entered or calculated using the calibrate function (see page 39 and IDN P-0-214 on page 80).

P-0-44 ADC 1.5V Reference Input

Main Properties

Read Only
Two Byte Fixed Length
Parameter set 0 only
Logable

Description

This parameter returns the value read on the ADC input that is connected to the 1.5V reference. It is this value that is used to automatically correct the offset on the other ADC channels. See page 39 for more details.

P-X-5n Phase Current Error (n = 1, 2 & 3)

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

P-X-51, P-X-52 and P-X-53 will return the current loop servo error from phases A, B and C respectively. The servo error is the difference between the current demand and feedback.

P-X-6n Phase PWM Output (n = 1, 2 & 3)

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0 (in open loop mode)
Logable

Description

P-X-61, P-X-62 and P-X-63 reflect the demand to the PWM stages for phases A, B and C respectively. The scaling is proportional to the duty cycle of the output (±32767 = ±Maximum Duty Cycle) and is therefore proportional to volts applied to the phase. The Maximum Duty Cycle can be set via P-X-225 (see page 87).

Writing to this parameter will only be effective when in open loop mode and when the axis is enabled. When disabled, the parameter is forced to zero and when in other loop modes the parameter is set by the active servos.

P-X-71 SSC Demand 'A'

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter returns the first SSC synchronous demand word. What this represents is dependant upon the loop mode of the axis as shown in Table 12 on page 13.

P-X-72 SSC Demand 'B'

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter returns the second SSC synchronous demand word. What this represents is set using P-X-242 (see page 92).

P-0-100 Test Input Set Up Word

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 0 (All inputs disabled)

Description

This parameter defines the type of waveform applied and to which axes and phases it will be applied. Table 26 shows the definition of P-0-100.

15	14	13	12	11	10	9	8
Freeze	Single Shot	Not Used	Position Enable	Velocity Enable	Current Enable	Phase B Enable	Phase A Enable
7	6	5	4	3	2	1	0
Waveform Mode	Not Used				Axis 3 Enable	Axis 2 Enable	Axis 1 Enable

Table 26 – Test Generator Set Up Word (TestSetup)

The definition of the individual bits is given below:

Bit 0 Axis 1 Enable
0 – Test input not applied to axis 1
1 – Test input applied to axis 1

Bit 1 Axis 2 Enable
0 – Test input not applied to axis 2
1 – Test input applied to axis 2

Bit 2 Axis 3 Enable (meaningful for Aries PQR only)
0 – Test input not applied to axis 3
1 – Test input applied to axis 3

Bits 3-6 Not Used

Bit 7 Waveform Mode
0 – Internally generated square wave
1 – Downloaded waveform

Bit 8 Phase A Enable
0 – Test input not applied to phase A current demand
1 – Test input applied to phase A current demand

Bit 9 Phase B Enable
0 – Test input not applied to phase B current demand
1 – Test input applied to phase B current demand

Bit 10 Main Current Enable
0 – Test input not applied to main current demand
1 – Test input applied to main current demand

Bit 11 Velocity Enable
0 – Test input not applied to velocity demand
1 – Test input applied to velocity demand

Bit 12 Postion Enable
0 – Test input not applied to position demand
1 – Test input applied to position demand

Bit 13 Not Used

Bit 14 Single Shot
0 – The buffer pointer wraps around to the beginning of the buffer when it reaches the end
1 – The buffer pointer stops (freeze bit set) when it reaches the end

Bit 15 Freeze
0 – The buffer pointer increments according to the sample time set in P-0-105
1 – The buffer pointer does not update, generator output is frozen on current sample

P-0-101 Test Square Wave Period

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 10000 (one second for default P-0-247)

Description

This parameter defines the time period (in servo cycles) of one half of the test input square wave.

P-0-102 Test Square Wave High Amplitude

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 5000

Description

This parameter defines the high level of the test input square wave. The low level is set with P-0-103.

P-0-103 Test Square Wave Low Amplitude

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = –5000

Description

This parameter defines the low level of the test input square wave. The high level is set with P-0-102.

P-0-104 Test Input Waveform

Main Properties

Read and Write
Two Byte Variable Length
Parameter set 0 only

Description

This parameter defines the test waveform that can be applied to the servo inputs. The buffer can contain up to 16384 samples.

P-0-105 Test Input Waveform Sample Time

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 1

Description

This parameter defines the sample period (in servo cycles) of the input waveform. That is, the number of servo cycles that each element of the input waveform buffer (P-0-104) corresponds to.

P-0-106 Test Input Waveform Length

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 1000

Description

This parameter should be set to the number of samples contained within the waveform downloaded to P-0-104. This defines the point at which the input buffer will wrap around to its start.

P-0-107 Test Input

Main Properties

Read Only
Two Byte Fixed Length
Parameter set 0 only
Logable

Description

This parameter returns the current instantaneous value of the test input.

P-0-108 Phased Test Input

Main Properties

Read Only
Two Byte Fixed Length
Parameter set 0 only
Logable

Description

This parameter returns the current instantaneous value of the ‘phased’ test input. When the test generator is in ‘Downloaded Waveform’ mode, the phased test input is applied to Phase B current demands (when selected) rather than the main test input (P-0-107). The Phased test input is phase delayed relative to the main test input by the number of samples indicated by P-0-109. When in ‘Internally generated square wave’ mode, the phased output is the same as the main output (i.e. P-0-107 = P-0-108).

P-0-109 Test Phase Offset

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only

Description

This parameter defines the delay (in test generator samples) between the main and phased test inputs.

P-0-110 Input Buffer Pointer

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only

Description

This parameter returns the current value of the buffer pointer. Writing a new value will instantaneously change the value of the test generator outputs.

P-0-111 Position Input Multiplier

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 1

Description

The output of the test generator is a 16 bit word. The position loop demand is a 32 bit long word. If the test generator output is connected to the position loop demand (bit 12 of P-0-100 set), then this parameter defines a multiplier that is applied to the 16 bit output before being applied to the 32 bit demand.

P-X-129 Class 1 Diagnostics (High Word)

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter contains indications of fatal faults and limits, i.e. those that cause a RED enable state. In SERCOS speak this is known as “Manufacturer Class 1 Diagnostics”. Internally, the diagnostics word is actually a 32 bit long word, This parameter contains the higher 16 bits and S-X-129 contains the lower 16 bits. See description of S-X-129 (page 59) for more details.

P-X-200 Servo Option Switches

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
SPA2 Default = 0x1001
SPAlite and Aries Default = 0x3003
Flash Saved

Description

The options word contains the various switches that control the modes of the servo filter (see Figure 13 and description starting on page 29). Table 27 shows the structure of the options word.

15	14	13	12	11	10	9	8
Use SSC Velo. F/B	Velo. F/B Type	Motor Type	SSC Dem. Enable	VET Source	Average Velocity	Torque Comp.	Curr. Loop Bypass
7	6	5	4	3	2	1	0
Axis S/W Enable	Encoder + Tacho	Lissajous Correction	Velocity D Source	Velocity/Commutation Encoder Source		Loop mode	

Table 27 – Options Word

The definition of the individual bits is given below:

Bits 0-1 Loop Mode:

Bit 1	Bit 0	Mode
0	0	Open Loop
0	1	Non-Commutated Current Loop
1	1	Commutated Current Loop
1	0	Velocity Loop

Table 28 – Loop Mode Bits

Bits 2-3 Velocity and Commutation Encoder Source:

Bit 3	Bit 2	Source
0	0	Digital Encoder Input
0	1	Analogue Encoder Input
1	1	SSC Synchronous Demand ‘B’
1	0	No Source Selected

Table 29 – Velocity Encoder Source Bits

Bit 4 Velocity PID Derivative Source
0 – Derived from feedback
1 – Derived form error

Bit 5 Enable Lissajous Correction
0 – Analogue encoder Lissajous is not corrected
1 – Analogue encoder Lissajous is corrected using downloaded map

Bit 6 Sum Velocity Encoder and Analogue Tacho
0 – Velocity feedback is derived from Encoder OR Tacho (depending on bit 14)
1 – Velocity feedback is derived from the sum of Encoder and Analogue Tacho

Bit 7 Axis Software Enable
0 – Axis is disabled
1 – Axis is enabled

Bit 8 Current Loops Bypass
0 – Current loops active
1 – Current loops bypassed (current demand becomes voltage/PWM demand)

Bit 9 Enable Torque Compensation
0 – Torque compensation is disabled
1 – Position based demand is applied to current input using downloaded map

Bit 10 Average Velocity
0 – No velocity averaging occurs
1 – With velocity decimation greater than one, intervening feedback samples are averaged.

Bit 11 VET Source
0 – Velocity servo error is used as the basis of the VET limit
1 – Difference between on-board and SSC velocity is used as the basis of the VET limit

Bit 12 SSC Demand Enable
0 – Demands that arrive via the SSC are ignored
1 – Demands that arrive via the SSC are used

Bit 13 Motor Type
0 – Brushless motor mode (three phase)
1 – Brushed motor mode (single phase)

Bit 14 Velocity Feedback Type
0 – Encoder used for velocity feedback
1 – Analogue tacho used for velocity feedback

Bit 15 Use SSC Velocity
0 – Velocity feedback depends upon on-board sensors (i.e. encoder and/or tacho) only
1 – Velocity feedback is sum of on-board sensors and SSC Velocity signal

P-0-203 PWM Period

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 1545 (for 150 MHz processor, gives 97 kHz PWM frequency)
Flash Saved

Description

This parameter sets the PWM frequency. It directly maps to the Period register of the timers that control the PWM outputs (T1PR and T3PR – see reference [7]) . The value is given by:

$$P\text{-}0\text{-}203 = \frac{f_{CLK}}{f_{PWM}} - 1$$

Where f_{CLK} is the processor clock frequency (150 MHz) and f_{PWM} is the desired PWM frequency. To determine PWM frequency from P-0-203 use:

$$f_{PWM} = \frac{f_{CLK}}{(P\text{–}0\text{–}203) + 1}$$

P-X-204 Peak Current Max Time

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 19999 (2 seconds for default P-0-247)
Flash Saved

Description

This parameter sets the maximum time (in servo cycles) that peak current (i.e. 50% of I_{FSD}) can safely be applied to the motor. After this time, the I^2t fault will be activated.

Note that changing the Motor Continuous Current parameter (S-X-111, see page 59) will change this parameter and so P-X-204 must be reset if S-X-111 is changed.

P-X-205 Current Error Time Limit

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 20000 (2 seconds for default P-0-247)
Flash Saved

Description

This parameter defines the length of time (in servo cycles) that current error is allowed to remain at I_{PEAK} (50% of I_{FSD}) before a Current Error Time fault occurs. Setting to zero will disable the limit.

P-0-206 Software Enable

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 1 (enabled)
Flash Saved

Description

This parameter sets the software enable state of the controller. Note that the enable input to the controller is actually the logical AND of the hardware enable input, the SSC enable, and P-0-206. A value of ‘1’ means enabled and ‘0’ means disabled.

P-X-207 Analogue Gain Settings

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
SPA2 Default = 0xA5 (I_{FSD} = 6.75A; I_{PEAK} = 3.38A)
SPAlite Default = 0xA5 (I_{FSD} = 3.38A; I_{PEAK} = 1.69A)
Flash Saved

Description

In order to make best use of the limited resolution ADCs, the analogue pre-conditioning circuitry includes some programmable gain elements. These are programmed using P-X-207. Table 30 shows the structure of the Analogue Gain Settings word. Table 31 and Table 32 define the current switch settings for SPA2 and SPAlite respectively. Table 33 shows the analogue tacho switch settings. Setting bit 12 will reduce the tacho gain by a factor of ten.

Note that Table 31 also shows the settings for hardware version 3 and below, which had different current scaling. This will affect current loop PID, I^2t and IET settings and so the configuration must be changed when transitioning from a V3 to a V4 main board (or vice versa).

For SPAlite, there are no phase B current switches (indeed there is no phase B) therefore bits 5-7 of the settings word are not defined and writing to them has no effect. For compatibility purposes, the same set of codes can be used as in SPA2 to obtain the same gains. Note, however, that all the current ranges are halved in SPAlite relative to V4 SPA2. Tacho settings in SPAlite remain identical to SPA2.

All other hardware variants do not have programmable analogue gains so this parameter does nothing. For Aries the current scaling is fixed such that I_{FSD} = 19A and I_{PEAK} = 9.5A (i.e. the same as the highest SPA2 range). For Gyro, I_{FSD} = 1.92A and I_{PEAK} = 0.96A.

15	14	13	12	11	10	9	8
Not Used			Tacho ÷ 10	Analogue Tacho Switch Settings			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Phase B Current Switch Settings			Not Used		Phase A Current Switch Settings		

Table 30 – Analogue Gain Settings Word

P-X-207			Hardware V3 & below			Hardware V4 & above			P-X-207
Bits 2 & 7	Bits 1 & 6	Bits 0 & 5	G	I_{FSD} (A)	I_{PEAK} (80%)	G	I_{FSD} (A)	I_{PEAK} (50%)	(Low Byte)
0	0	0	0.33	11.40	9.12	0.20	19.00	9.50	0x0
0	0	1	0.38	9.91	7.93	0.23	16.52	8.26	0x21
0	1	0	0.45	8.41	6.73	0.27	14.02	7.01	0x42
0	1	1	0.54	6.93	5.54	0.32	11.54	5.77	0x63
1	0	0	0.68	5.54	4.43	0.41	9.23	4.61	0x84
1	0	1	0.93	4.05	3.24	0.56	6.75	3.38	0xA5
1	1	0	1.47	2.55	2.04	0.88	4.25	2.13	0xC6
1	1	1	3.53	1.06	0.85	2.12	1.77	0.89	0xE7

Table 31 – AGS Current Settings (SPA2)

P-X-207			SPAlite			P-X-207
Bit 2	Bit 1	Bit 0	G	I_{FSD} (A)	I_{PEAK} (50%)	(Low Byte)
0	0	0	0.20	9.50	4.75	0x0
0	0	1	0.23	8.26	4.13	0x21
0	1	0	0.27	7.01	3.51	0x42
0	1	1	0.32	5.77	2.89	0x63
1	0	0	0.41	4.61	2.31	0x84
1	0	1	0.56	3.38	1.69	0xA5
1	1	0	0.88	2.13	1.07	0xC6
1	1	1	2.12	0.89	0.45	0xE7

Table 32 – AGS Current Settings (SPAlite)

P-X-207				
Bit 11	Bit 10	Bit 9	Bit 8	Gain
0	0	0	0	1.00
0	0	0	1	1.60
0	0	1	0	2.26
0	0	1	1	2.87
0	1	0	0	3.46
0	1	0	1	4.06
0	1	1	0	4.72
0	1	1	1	5.32
1	0	0	0	5.89
1	0	0	1	6.49
1	0	1	0	7.15
1	0	1	1	7.76
1	1	0	0	8.35
1	1	0	1	8.95
1	1	1	0	9.61
1	1	1	1	10.21

Table 33 – AGS Analogue Tacho Settings

P-0-208 Save Parameters to Flash

Main Properties

Read Only
Two Byte Fixed Length
Parameter set 0 only

Description

Reading this parameter will cause a flash save to occur. This effectively changes the reset defaults of some important IDNs. The parameter will return an error code generated by the flash save task. Table 34 describes the error codes that could result. Note for the algorithm errors (erase, program or verify), the error code (see reference [14]) is returned with 0x100, 0x200 or 0x300 added to indicate which algorithm cause the error.

Error Code	Meaning
0	Flash save was successful
1	Controller is enabled (flash save cannot proceed when controller is enabled)
0x1XX	Error in flash erase algorithm
0x2XX	Error in flash program algorithm
0x3XX	Error in flash verify algorithm

Table 34 – Flash Saving Error Codes

Table 35 defines the parameters that are stored during a flash save.

Parameter Type	IDNs
Current Loop PI Gains	S-X-93, S-X-106, S-X-107
Velocity Loop PID Gains	S-X-45, S-X-100, S-X-101, S-X-102
Position Loop PI gains	S-X-77, S-X-104, S-X-105
Velocity Feed-forward Gain	P-X-705
Acceleration Derivation Gain	P-X-618
Accel. Feed-forward Gain	P-X-612
Friction Offset	P-X-613
AFO Parameters	P-X-614, P-X-615, P-X-616
Coupled Mass Gain	P-X-633
IIR Filter Coefficients	P-X-500, P-X-501, P-X-502, P-X-503, P-X-504, P-X-505, P-X-506, P-X-507, P-X-508, P-X-509, P-X-510
Velocity Decimation Factor	P-X-604
Velocity Scale Factors	P-X-602, P-X-611, P-X-623
Encoder Counts Per Rev	P-X-651
No Of Poles Per Rev	P-X-652
Commutation Mode	P-X-680
Hall Offset and Direction	P-X-681, P-X-684
Analogue Gain Settings	P-X-207
Servo Period	P-0-247
Option Words	P-X-200
Position Loop Options	P-X-700
Trajectory Control Settings	P-X-703, P-X-704
Digital Inputs	P-X-21, P-X-22, S-X-81, S-X-37
Demand Polarities	P-X-229, P-X-230
ADC Offsets	P-0-40, P-X-41, P-X-42, P-X-43
ADC Auto Correct Enable	P-0-245
Current Limits	S-X-111, P-X-204, P-X-205
Velocity Limit Times	P-X-220, P-X-236
Software Enable	P-0-206
PWM Period	P-0-203
PWM Deadband Time	P-0-218
Max PWM Duty Cycle	P-X-225
Expected Vbus Input	P-0-226
Vbus Fault Tolerance	P-0-248
Qualification Period For Estop	P-0-213
Qual. Period For SSC Enable	P-0-216
SSC Demand ‘B’ Selector	P-X-242
SSC Time Out	P-0-215
SSC Time Out Override	P-0-217
SSC Checksum Enable	P-0-228
RS232 Baud and Settings	P-0-209, P-0-210
Braking Controls	P-0-222, P-0-223, P-0-224, P-X-235
Temperature Settings	P-0-233, P-0-234
Fan Pulse Threshold	P-0-244
SSC Feedback	P-X-238, P-X-239, P-X-240
Commutation Setup Offset	P-X-653
CAS Gains	P-X-656, P-X-661
Heydemann Comp. Parameters	P-X-670, P-X-671, P-X-672, P-X-673, P-X-674
Lissajous Correction Map	P-X-675
Torque Comp. Map Size	P-0-644
Reference Mark/TTP Control	P-X-691, P-0-692

Table 35 – Parameters That Are Saved To Flash

Given that this instruction has to erase an entire section of flash, it takes a relatively long time to complete (up to 7 seconds on some DSPs). The ACP host must take this in to account when issuing this IDN as there will be a delay before the response occurs.

P-0-209 Serial Comms Baud Rate Register

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 162 (115200 bps)
Flash Saved

Description

This parameter sets the baud rate at which the SPA2 will communicate via the RS232 port. It directly maps to the SCI Baud Rate Registers (SCIHBAUD and SCILBAUD – see reference [11]). The value is given by:

$$P\text{ - }0\text{ - }209 = \frac{f_{CLK}}{BAUD \times 8} - 1$$

Where f_{CLK} is the processor clock frequency (150 MHz) and $BAUD$ is the baud rate in bits per second. Table 36 gives P-0-209 values for popular baud rates. Note that changing this value will not immediately change the baud rate. The new value must be saved to flash (using P-0-208) and the change will take effect on the next reset.

Baud	P-0-209
9600	1952
19200	976
38400	487
57600	325
115200	162

Table 36 – Popular Baud Settings

P-0-210 Serial Comms Control Register

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 0xE7 (No Handshaking; 2 Stop Bits; Even Parity; 8 Data Bits)
Flash Saved

Description

This parameter controls the properties of the RS232 port, Table 37 shows the structure of this.

15	14	13	12	11	10	9	8
Not Used							RTS/CTS Handshake
7	6	5	4	3	2	1	0
No. Of Stop Bits	Parity		0	0	Number of data bits (111)		

Table 37 – Serial Communications Control Word

The definition of the individual bits is given below:

Bits 0-2 Number of data bits
Must be set to 111 (8 data bits) for serial protocol to work

Bits 3-4 Reserved
Must be set to 00

Bits 5-6 Parity

Bit 6	Bit 5	Parity
X	0	None
0	1	Odd
1	1	Even

Table 38 – Parity Control Bits

Bit 7 Number of stop Bits
0 – One stop bit
1 – Two stop bits

Bit 8 RTS/CTS Handshaking Enable (SPA2 Only)
0 – Hardware handshaking disabled
1 – Hardware handshaking enabled

Bits 9-15 Not Used

Note that changing this value will not immediately change the port settings. The new value must be saved to flash (using P-0-208) and the change will take effect on the next reset. Only the SPA2 variant is capable of implementing RTS/CTS handshaking, the relevant hardware does not exist in other variants.

P-0-211 Enable State

Main Properties

Read Only
Two Byte Fixed Length
Parameter set 0 only
Logable

Description

This parameter returns the current enable state of SPA2 Table 39 shows how each of the five enable states is represented by the value of P-0-211.

Enable State	P-0-211
RED	0
AMBER	1
GREEN	2
BRAKE TO RED	3
BRAKE TO AMBER	4

Table 39 – P-0-211 Definition

P-0-212 Time Spent in Servo Routine

Main Properties

Read Only
Two Byte Fixed Length
Parameter set 0 only
Logable (see note)

Description

This parameter will return the time, t_S (in processor clock ticks = 1/150MHz), spent in the last servo interrupt service routine. This gives an indication of how loaded the DSP is.

Note that when this parameter is logged using the data gathering function, the actual value stored is $T_S - t_S$ where T_S is the servo period (also in processor clock ticks – P-0-247 × 150).

P-0-213 Qualification Period For EStop OK

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 40 (4ms for default P-0-247)
Flash Saved

Description

This parameter sets the time, in servo cycles, that the ESTOP OK signal must remain changed before that change takes effect within the SPA2. This prevents short glitches on the signal erroneously causing amplifier faults. The same qualification period will also be used for the Vbus relay monitor signal.

P-0-214 Calibration Mode Flag

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 0

Description

This parameter controls the ADC calibration function. Setting P-0-214 to 1 will start the calibration routine. The SPA2 will reset the parameter back to 0 when the calibration is complete. The process takes about 1.6 seconds.

Calibration consists of taking the mean value of all the current and tacho ADC inputs over 16384 samples. These values are then used to correct subsequent ADC readings. For current feedback, the ADC corrections calculated by the calibration routine can be found in P-X-41 and P-X-42. For tacho feedback, the ADC corrections can be found in P-X-43.

Of course, it is important that the ADC inputs are zero at the time of calibration. It is up to the user to ensure that this is the case (e.g. PWM bridge must be disabled and machine not moving).

In addition, a common ADC offset is calculated which is the mean of the offsets found from the current input channels. This offset is applied to the ADC channels that cannot be calibrated in the normal way because the inputs cannot be forced to zero (i.e. bus voltage and temperatures). This offset is stored in P-0-40.

Each hardware variant has a different number of current sensors and the missing ones are forced to zero in the code. To avoid a bias the calculation of Common ADC offset is slightly different for each variant as shown in Table 40.

Amp Type	Common ADC Offset Is The Mean Of:
SPA2 (both DSPs)	All current inputs (4 channels per DSP)
SPA2-2 & Gyro	As SPA2 (4 channels per DSP)
SPAlite (XY DSP)	Phase A current inputs only (2 channels)
SPAlite (Z DSP)	Phase A current input of first axis (Z) only (1 channel)
Aries (PQR DSP)	Phase A current inputs only (3 channels)
Aries (ST DSP)	Phase A current inputs only (2 channels)

Table 40 – Calculation of Common ADC Offset Depending Upon Amplifier Type

P-0-215 SSC Time Out

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 100 (10ms for default P-0-247)
Flash Saved

Description

This parameter sets the maximum time, in servo cycles, that can elapse without receiving a SSC message. If this time is exceeded, the SSC Time Out fault is activated.

P-0-216 Qualification Period For SSC Enable

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 10 (1ms for default P-0-247)
Flash Saved

Description

This parameter sets the time, in servo cycles, that the SSC Enable signal (bit 7 of the SSC control byte) must remain changed before that change takes effect within the SPA2. This prevents short glitches on SSC causing erroneous amplifier operation.

P-0-217 SSC Time Out Override

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 0 (Fault Not Overriden)
Flash Saved

Description

Setting this parameter to 1 will disable the SSC Time Out fault. This allows the amplifier to run without the SSC connected.

Note that if an SSC Time Out fault exists, the SSC enable signal is forced true. Conversely, if there is no such fault the SSC enable signal is used as normal, regardless of the state of the SSC Time Out Override flag (P-0-217). Table 41 shows how the SSC Enable Override flag (P-0-221), the SSC Time Out Override flag (P-0-217) and the SSC Time Out fault effect the SSC enable signal.

Enable Override (P-0-221)	SSC Time Out	Time Out Override (P-0-217)	SSC Enable Input
0	No Fault	Don’t Care	As Received On SSC
0	Fault	1	As SSC when fault occurred (true at start up)
0	Fault	0	Forced True
1	Don’t Care	Don’t Care	Forced True

Table 41 – Effect Of Override Flags and SSC Time Out Fault On SSC Enable Input

P-0-218 PWM Dead Band Time

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 5 (533ns)
Flash Saved

Description

This parameter defines the time between the bottom FETs of the PWM bridge switching off and the top FETs switching on (and vice versa). This ensures that at no point both FETs are on together and so protects against the shoot through condition. The value is given by:

$$T_{DEAD} = P\text{-}0\text{-}218 \times 106.67ns$$

P-0-218 directly maps to the DBT bits of DBTCONA and DBTCONB event manager registers (see Reference [7]).

This parameter is protected from accidental change by the Sensitive Parameter Protection Lock (P-0-996, page 131).

This parameter is only meaningful for hardware variants that directly control the individual bridge FETs, i.e. SPA2, SPA2-2 and SPAlite. For Aries and Gyro, P-0-218 has no effect.

P-0-219 Hardware Version

Main Properties

Read Only
Two Byte Fixed Length
Parameter set 0 only

Description

This word contains information about the hardware variant and version that the firmware is running on. The information is mostly encoded on the PWM output lines which are read during boot and then reconfigured as outputs for normal use. In addition, for SPA2 and SPA2-2, GPIO bits A6, B6 & D1 are used to identify the type of SPA2 encoder board fitted (if any).

The key to decoding this parameter is the bottom five bits which reveal the basic type of the hardware (Table 45). The structure of the remaining bits depends upon hardware type as defined in Table 42, Table 43 and Table 44 below.

15	14	13	12	11	10	9	8
Not Used	Encoder Board Type Code			Fourth Axis Expansion Board Version Code			Main Brd Ver. Code
7	6	5	4	3	2	1	0
Main Board Version Code		DSP ID	Type Code / DSP Board Version (SPA2-2 = 00111) (SPA2 = DSP Board Version)				

Table 42 – Hardware Version Structure For SPA2 and SPA2-2

15	14	13	12	11	10	9	8
Not Used				Hardware Version (High 4 bits)			
7	6	5	4	3	2	1	0
Hardware Version (Low 2 bits)		DSP ID	Type Code (SPAlite = 00110) (Aries = 01001)				

Table 43 – Hardware Version Structure For SPAlite and Aries

15	14	13	12	11	10	9	8
Not Used				Hardware Version (High 4 bits)			
7	6	5	4	3	2	1	0
Hardware Version (Low 3 bits)			Type Code Gyro = 01000				

Table 44 – Hardware Version Structure For Gyro

The definition of the individual bits is given below:

Bits 0-4 (all variants) – Hardware Type Code / SPA2 DSP Board Version

Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Type or DSP Board Revision
0	0	0	0	0	SPA2 Prototype Mk1
0	0	0	0	1	SPA2 Prototype Mk2
0	0	0	1	0	SPA2 DSP Board Mk1
0	0	0	1	1	SPA2 DSP Board Mk2
0	0	1	0	0	SPA2 DSP Board Mk3
0	0	1	0	1	SPA2 DSP Board Mk4 & 5
0	0	1	1	0	SPAlite
0	0	1	1	1	SPA2-2
0	1	0	0	0	Gyro
0	1	0	0	1	Aries

Table 45 – Type Codes / SPA2 DSP Versions

Bit 5 (except Gyro) – DSP ID

Bit 5	SPA2	SPA2-2	SPAlite	Aries
0	DSP 1	DSP XY	DSP XY	DSP PQR
1	DSP 2	DSP ZW	DSP Z	DSP ST

Table 46 – DSP ID Bit

Bits 6-8 (SPA2) – SPA2 Main Board Version Code

Bit 8	Bit 7	Bit 6	Main Board Revision
0	0	0	Prototype Board
0	0	1	Main Board Mk1
0	1	0	Main Board Mk2
0	1	1	Main Board Mk3
1	0	0	Main Board Mk4
1	0	1	Main Board Mk5

Table 47 – Valid Main Board Version Codes

Bits 6-8 (SPA2-2) – Board Version (see also P-0-246, page 93)

Bit 8	Bit 7	Bit 6	SPA2-2 Board Revision
1	0	0	V1
1	0	1	V2 & V3
1	1	0	V4
1	1	1	V5

Table 48 – SPA2-2 Board Version

Bits 9-11 (SPA2 and SPA2-2) – Fourth Axis Expansion Board Version Code

Only applies to second DSP (i.e. if Bit 5 is set). Note that there was never a Mk 1 expansion board.

Bit 11	Bit 10	Bit 9	Expansion Board Revision
0	0	0	Board Not Fitted
1	0	0	Expansion Board Mk 2
0	1	0	Expansion Board Mk 3
1	1	0	Expansion Board Mk 4
0	0	1	Expansion Board Mk 5

Table 49 – Fourth Axis Expansion Board Version Codes

Bits 6-11 (SPAlite) – Hardware Version Code

Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	SPAlite Revision
0	0	0	0	0	0	SPAlite Version 1
0	0	0	0	0	1	SPAlite Version 2
0	0	0	0	1	0	SPAlite Version 3

Table 50 – SPAlite Board Version

Bits 6-11 (Aries) – Hardware Version Code

Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Aries Revision
0	0	0	0	0	0	Aries Version 1
0	0	0	0	0	1	Aries Version 2

Table 51 – Aries Board Version

Bits 12-14 (SPA2 and SPA2-2) – Encoder Board Type Code

Bit 14	Bit 13	Bit 12	Encoder Board Revision
0	0	0	Encoder Board Not Fitted
0	0	1	Digital Encoder Board Mk1, 2 & 4
0	1	0	Analogue Encoder Board V1
0	1	1	Analogue Encoder board V2

Table 52 – Encoder Board Type Codes

P-X-220 Velocity Limit Time

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 1000 (100 ms for default P-0-247)
Flash Saved

Description

The time, in servo cycles, that the velocity feedback signal must remain at full scale before a Over Velocity fault occurs. This makes allowance for noise on the tacho input signal and small amounts of overshoot in the velocity servo.

P-0-221 SSC Enable Override

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 0 (Enable Input Not Overridden)

Description

Setting this parameter to 1 will cause the SPA2 enable state machine to ignore the enable signal arriving via the SSC link (bit 7 of the control byte). In this case the enable state of the amplifier is determined by the software enable signal (P-0-206, page 74) and the hardware enable input. The remainder of the SSC link functions as normal.

Note that if an SSC Time Out fault exists, the SSC enable signal is also overridden. See P-0-217 (page 82) for more details.

P-0-222 Braking Time

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 1250 (2 seconds for default P-0-247)
Flash Saved

Description

This parameter determines the time spent in active braking mode after a disable or fault when in ‘Timed Braking Mode’. When in ‘Host Controlled Braking Mode’ this parameter has no effect. The units of this parameter are such that 1 = 16 servo cycles.

P-0-223 Braking Mode

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 1 (Timed Braking Mode)
Flash Saved

Description

This parameter determines the active braking mode. A value of 0 indicates ‘Host Controlled Mode’, in which the braking mode can be set by the host using the Brake Flag (P-0-224). A value of 1 indicates ‘Timed Mode’, in which braking mode is active for a set time (determined by P-0-222).

P-0-224 Brake Flag

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 0 (Braking Not Active)
Flash Saved

Description

This parameter determines whether braking is active during a disabled state when in ‘Host Controlled Braking Mode’. When in ‘Timed Braking Mode’ this parameter has no effect.

P-X-225 Axis Maximum PWM Duty Cycle

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 65535 (100%)
Flash Saved

Description

This parameter sets the maximum PWM duty cycle that can be applied to the axis. This sets a voltage limit on the motor output, which can be calculated thus:

$$P\text{–}X\text{–}225 = \frac{Vlimit}{Vbus} \times 65535$$

Where *Vlimit* is the max voltage to be applied to the motor and *Vbus* is the input bus voltage (set in P-0-226).

P-0-226 Expected Bus Voltage Input

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
SPAlite Default = 23568 (48V)
Aries Default = 11784 (24V)
Other variants Default = 29460 (60V)
Flash Saved

Description

This parameter sets the nominal bus voltage level. If the actual bus voltage differs from this value by more than a tolerance value (set in P-0-248, page 95) then a Vbus fault will occur. The units of this parameter are such that 491 = 1V.

P-X-227 Scaled Bus Voltage Input

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter returns the bus voltage as read by the axis ADC input. If this differs from the expected value (entered in P-0-226) by more than a tolerance value (set by P-0-248, page 95) then a Vbus fault will occur. The units of this parameter are such that 491 = 1V.

P-0-228 SSC Checksum Enable

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 1 (Checksum Enabled)
Flash Saved

Description

This parameter determines whether the SSC communications are protected by a checksum. Note that changing this value will not immediately change the state of the checksum. The new value must be saved to flash (using P-0-208, page 76) and the change will take effect on the next reset.

P-X-229 Main Current Demand Polarity

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 1 (Positive)
Flash Saved

Description

This parameter applies a ±1 gain after the main current demands (I_{in}, I_{off} and test input) have been combined. The magnitude of this parameter is not important, only the sign is used. Figure 13 shows the position of this gain in the servo structure.

P-X-230 Velocity Demand Polarity

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 1 (Positive)
Flash Saved

Description

This parameter applies a ±1 gain after the velocity demands (V_{in}, V_{off} and test input) have been combined. The magnitude of this parameter is not important, only the sign is used. Figure 13 shows the position of this gain in the servo structure.

P-0-231 Ambient Temperature

Main Properties

Read Only
Two Byte Fixed Length
Parameter set 0 only
Logable

Description

This parameter returns the temperature as read by the onboard temperature sensor. It is scaled such that 699 = 1°C.

P-0-232 Heater Temperature

Main Properties

Read Only
Two Byte Fixed Length
Parameter set 0 only
Logable

Description

SPA2 variant only: This parameter returns the temperature as read by the heated onboard temperature sensor that is mounted close to the air intake fan in the enclosure. This sensor is artificially heated by a power resistor mounted nearby. The value is scaled such that 699 = 1°C.

This parameter will return zero for all variants other than SPA2.

P-0-233 Max Temperature

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 59415 (85°C)
Flash Saved

Description

This parameter defines the maximum ambient temperature that the amplifier should experience. If the temperature exceeds this value an Over Temperature fault will occur. The value is scaled such that 699 = 1°C.

P-0-234 Max Temperature Difference

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 34950 (50°C)
Flash Saved

Description

SPA2 variant only: This parameter defines the maximum temperature difference that can exist between the ambient and heated temperature sensors. If this is exceeded a Fan fault will occur. The value is scaled such that 699 = 1°C. Note that the default value of 50°C effectively disables the fan fault.

This parameter has no effect for all variants other than SPA2.

P-X-235 Braking Current

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 16384 (I_{PEAK})
Flash Saved

Description

This parameter sets the amount of current that the amplifier tries to maintain during active braking. Note that it can only do this while there is sufficient energy in the regenerating motor. After the motor has slowed sufficiently to no longer provide the specified amount of current, the bottom FETs will turn fully on – effectively shorting the motor terminals.

P-X-236 Velocity Error Time Limit

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 1000 (100ms for default P-0-247)
Flash Saved

Description

This parameter defines the length of time (in servo cycles) that the velocity error (or velocity difference if this mode is selected) is allowed to remain at V_{PEAK} (95% of full scale velocity) before a Velocity Error Time fault occurs. Setting to zero will disable the fault.

P-X-238 SSC Feedback Mode

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0 (Current Mode)
Flash Saved

Description

This parameter defines what is contained in the output data that is returned to the master in the SSC link. There are three possible settings as described in Table 53. In user defined mode (P-X-238 = 2), the returned data is defined by using P-X-239 and P-X-240 as pointers to 2812 memory locations. P-0-308 can be used to find the memory locations associated with a variety of IDNs.

MODE	P-X-238	Byte 1	Byte 2	Byte 3	Byte 4
Current	0	Phase 1 Current (P-X-31)		Phase 2 Current (P-X-32)	
Velocity	1	Velocity Feedback (S-X-40)		Current Demand (S-X-80)	
User Defined	2	Contents of Address in P-X-239		Contents of Address in P-X-240	

Table 53 – Synchronous Feedback Used For SPA2

P-X-239 SSC Feedback, User Defined Address 1

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0 (Not Defined)
Flash Saved

Description

When the SSC Feedback mode is set to user defined (P-X-238 = 2) this parameter specifies the 2812 memory address of the data to be returned as the first word of the SSC transmission back to the master for the specified axis. This parameter has no effect for all other SSC feedback modes. P-0-308 (see page 99) can be used to find the memory locations associated with a variety of IDNs.

P-X-240 SSC Feedback, User Defined Address 2

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0 (Not Defined)
Flash Saved

Description

When the SSC Feedback mode is set to user defined (P-X-238 = 2) this parameter specifies the 2812 memory address of the data to be returned as the second word of the SSC transmission back to the master for the specified axis. This parameter has no effect for all other SSC feedback modes. P-0-308 (see page 99) can be used to find the memory locations associated with a variety of IDNs.

P-0-241 Checksum Failure Counter

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Logable

Description

This parameter records the number of SSC checksum fails. Every time such a fail occurs the value of this parameter is incremented by one. The value is zero at start up and can be reset by writing zero to it.

P-X-242 SSC Demand ‘B’ Selector

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0 (Phase B Current demand)
Flash Saved

Description

This parameter defines the purpose of the second synchronous demand that is received by the SSC link (SSC demand 'B'). This can be one of four values:

P-X-242	Use of SSC Demand ‘B’
0	Phase B Current demand
1	SSC Current Offset
2	SSC Velocity Feedback
3	SSC Position feedback

Figure 13 shows where these signals fit into the overall servo scheme. The firmware does not protect against meaningless configurations, e.g. setting P-X-242 to 0 in any other mode than non-commutated current loop is pointless since the Phase B current demand is only used in this mode.

P-0-243 Pulse Input From Fan

Main Properties

Read Only
Two Byte Fixed Length
Parameter set 0 only
Logable

Description

For SPA2-2 and Aries, this parameter returns the value read by the ADC channel connected to the fan pulse tacho. See page 25 for more details. Other variants do not have a fan tacho input and so this parameter returns zero.

P-0-244 Fan Pulse Fault Threshold

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 16384
Flash Saved

Description

For SPA2-2 and Aries, this parameter specifies the maximum deviation from the mid point that is allowed by the fan pulse input (as returned by P-0-243) before a fan fault occurs. Setting to 32767 effectively disables the fan fault. Other variants do not have a fan tacho input and so this parameter does nothing.

P-0-245 ADC Auto Correct Flag

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
SPA2-2 and Aries Default = 1 (Enabled)
Other Variants Default = 0 (Disabled)
Flash Saved

Description

This parameter can be used to disable the ADC auto correct feature in SPA2-2 and Aries. See page 39 for more details. This parameter must not be set to one (enabled) for variants that do not support the auto correct feature (i.e. any except SPA2-2 and Aries), as it will force gross offsets on the ADC inputs.

P-0-246 Unmodified Hardware Version

Main Properties

Read Only
Two Byte Fixed Length
Parameter set 0 only

Description

For all variants other than SPA2-2 this parameter returns exactly the same as P-0-219 (see page 83). For SPA2-2 the main board version code is modified for P-0-219 in that 0x4 is added to the raw code read from the PWM lines. This was done to ensure backwards compatibility with previous UCC software (i.e. Renicis) and effectively reduces the number of available codes from eight to four. P-0-246 will return the version code without this modification, i.e. the main board version code is returned as encoded on the PWM lines.

The bit descriptions for the returned word are the same as P-0-219 with the exception that the following table replaces Table 48.

Bits 6-8 (SPA2-2) – Board Version

Bit 8	Bit 7	Bit 6	SPA2-2 Board Revision
0	0	0	V1
0	0	1	V2 & V3
0	1	0	V4
0	1	1	V5

Table 54 – Unmodified SPA2-2 Board Version

P-0-247 Servo Interrupt

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 100 (100µs)
Flash Saved

Description

This parameter sets the servo update period (in microseconds). Changing this directly changes the period of the timer that generates the servo interrupt.

There are many functions that base their timing on the servo interrupt and so will be affected by changing this parameter. Any IDN that is specified as being scaled in terms of servo cycles will be affected, this includes the following parameters:

- Servo Integral Gains (S-X-101, S-X-105, S-X-107)
- Velocity Loop Differential Gain (S-X-102)
- Test Square Wave Period (P-0-101)
- Input Waveform Sample Time (P-0-105)
- Peak Current Max Time (I^2T Time) (P-X-204)
- Current Error Time Limit (IET Time) (P-X-205)
- Qualification Period For EStop OK (P-0-213)
- SSC Time Out (P-0-215)
- Qualification Period For SSC Enable (P-0-216)
- Velocity Limit Time (P-X-220)
- Brake Duration (P-0-222)
- Velocity Error Time (P-X-236)
- Qualification Period For PWM Fault Inputs (P-0-249)
- Data Gather Period (P-X-301)
- All IIR Filters (P-X-5XX)
- Encoder Velocity Scale Factor (P-X-602)
- Velocity loop Decimation Factor (P-X-604)
- SSC Velocity Scale Factor, if using SSC position feedback (P-X-611)
- AFO Decay Time Constant (P-X-615)
- CAS I Gain (P-X-656)
- Trajectory Calculator Max Velocity and Acceleration (P-X-703 & P-X-704)
- Velocity Feed-Forward Gain (P-X-705)

Note that Vbus Qualification time is hard coded as five servo cycles regardless of the servo period set in P-0-247.

It is the responsibility of the host to ensure that these dependant IDNs are modified to suit any new value for P-0-247.

P-0-248 Vbus Fault Tolerance

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Aries Default = 492 (4V)
Other Variants Default = 1100 (9V)
Flash Saved

Description

This parameter sets the maximum voltage deviation (from the expected bus voltage, as set by P-0-226, see page 87) at which a Vbus fault will occur. The scaling is four times less than the scaling of the P-0-226, i.e. 1V = 123. Using default values as an example, an Aries amplifier will register a Vbus Fault if the bus voltage drops below 20V or rises above 28V. For SPA2 and SPA2-2 the thresholds would be 51V and 69V.

P-0-249 Qualification Period For PWM Fault Inputs

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 10 (1ms for default P-0-247)
Flash Saved

Description

This parameter sets the time, in servo cycles, that the PWM over current input signals must remain changed before that change takes effect within the SPA2. This prevents short glitches due to noise on the signal erroneously causing amplifier faults. The same qualification period is used for all axes.

P-0-300 Data Gather Buffer

Main Properties

Read and Write
Two Byte Variable Length
Parameter set 0 only

Description

This parameter will return the contents of the data gather buffer. This is used to upload the data to the host once the gather is complete. The data returned is in an interlaced format as shown in Table 55, which shows the returned buffer from a log of M samples of N memory locations.

Word No	Data
1	Location 1, Sample 1
2	Location 2, Sample 1
:	:
N-1	Location N-1, Sample 1
N	Location N, Sample 1
N+1	Location 1, Sample 2
N+2	Location 2, Sample 2
:	:
2N-1	Location N-1, Sample 2
2N	Location N, Sample 2
2N+1	Location 1, Sample 3
2N+2	Location 2, Sample 3
:	:
(S x N) + L	Location L, Sample (S +1)
:	:
(M-1) x N – 1	Location N-1, Sample M
(M-1) x N	Location N, Sample M

Table 55 – Format Of Data Returned From P-0-300

Note that Table 55 is correct for data acquired in un-triggered mode. Triggered data is in the same format but will not start at the first sample of the buffer, but on the sample given by P-0-305 (page 98).

P-0-301 Data Gather Period

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 1

Description

This parameter determines the data logging sample period in servo cycles.

P-0-302 Data Gather Trigger Flags

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 0

Description

This parameter is used control the data gathering function. Table 56 shows how this word is configured. See page 38 for more details on data gathering functionality.

15	14	13	12	11	10	9	8
Not Used							
7	6	5	4	3	2	1	0
Not Used		Buffer Filled Flag	Trigger Polarity	Trigger State		Wrap Around	Gather Enable

Table 56 – Data Gathering Set Up Word

The definition of the individual bits is given below:

Bit 0 – Gather Enable
0 – Gathering is disabled
1 – Gathering is enabled

Bit 1 – Wrap around mode
0 – Gather disables when buffer is full (resets bit 0)
1 – Gather wraps around when buffer is full

Bits 2 & 3 – Trigger State

Bit 3	Bit 2	State
0	0	Disabled, no trigger checking
0	1	Triggered, gathering will continue until delay is done
1	0	Pre-armed, waiting for data to fall below trigger threshold and the buffer to wrap around at least once
1	1	Armed, waiting for data to go above trigger threshold

Table 57 – Trigger State Bits

Bit 4 – Polarity Bit
0 – Trigger occurs when source is going positive
1 – Trigger occurs when source is going negative
Note: Setting this bit also inverts the trigger threshold given in P-0-304.

Bit 5 – Buffer Filled Flag
0 – Buffer has not been filled since the flag was last reset
1 – Buffer has been filled since the flag was last reset

Bits 6-15 Not Used

P-0-303 Data Gather Parameter List

Main Properties

Read and Write
Two Byte Variable Length
Parameter set 0 only
Default = all zeros

Description

This parameter defines the memory locations that should be logged during data gathering. This is specified as a list of up to 15 valid memory locations. The list must be terminated with a zero if less than 15 locations are specified. P-0-308 (see page 99) can be used to find memory locations from IDN values.

P-0-304 Data Gather Trigger Threshold

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 0

Description

This parameter defines the level for the triggering function. For a positive trigger polarity (bit 4 of P-0-302 = 0) a trigger will occur when the source (channel 1) is increasing in value and equals the value of P-0-304. For a negative trigger polarity (bit 4 of P-0-302 = 1) a trigger will occur when the source is decreasing in value and equals minus the value of P-0-304.

P-0-305 Data Gather Triggered Buffer Start Position

Main Properties

Read Only
Two Byte Fixed Length
Parameter set 0 only

Description

When using a trigger, the resulting logged data will not appear in the gather buffer (P-0-300) starting at the first sample. It will start at the sample number given by this parameter and wrap around the buffer to finish at the sample number given by one less than this parameter. Bear in mind when decoding the buffer that the actual buffer length, and therefore the point at which it wraps around, will depend upon the nominal buffer length and the number of channels being logged (see P-0-307 for more details).

P-0-306 Data Gather Trigger Delay

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 501 (centre of buffer)

Description

This parameter specifies the position of the trigger, in samples, relative to the end of the buffer, plus one. E.g. P-0-306 = 1 will put the trigger point at the end of the buffer. By default, the trigger point is set to the middle of the buffer.

P-0-307 Data Gather Nominal Buffer Length

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 1000

Description

This parameter defines the data gather buffer length that is used in the DSP. Note that the actual buffer returned by P-0-300 will be 15 words longer than the nominal length to allow overflow in order to accommodate a complete set of data points for every logged parameter. The actual length of the buffer (i.e. the number of words containing valid data) is given by:

$$B + (N - (B \bmod N)) \bmod N$$

Where:
N = Number of parameters being logged
B = Nominal length of buffer
and the operator 'mod' means the 'the remainder upon dividing by'

This information is important to enable triggered data to be rescheduled after it is retrieved since it is at this point that the buffer will wraparound. The Maximum value for P-0-307 is 49137 (Note this drops to 32753 when running in debug mode using an emulator – see page 47).

P-0-308 Memory Location Look Up Table

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only

Description

This parameter acts as a look up table that converts IDNs to 2812 memory locations (low 64K). This is used in conjunction with the Data Gather Parameter List (P-0-303) to allow the user to specify data log locations as IDNs rather than actual memory locations.

Reading P-0-308 will return the memory location associated with the IDN number that was last written to it. So, to use this function, the required IDN should be written to P-0-308 and the associate memory location can then be read back.

Not all parameters are 'logable' as this would be pointless, meaningless and/or impossible in some cases. If an parameter is logable, P-0-308 will return a valid memory address, otherwise 0x8000 is returned. Logable parameters are signified by the word 'logable' in the 'Main Properties' section of the descriptions in this appendix. In addition, the list of 'logable' parameters is shown in Table 58.

IDN	Parameter
P-X-702	Pos Loop Combined Demand (lower 16 bits)
P-X-701	Position Loop Error
S-X-51	Position Feedback (lower 16 bits)
S-X-36	Velocity SSC Demand
S-X-37	Velocity Digital Demand
P-X-617	Acceleration Demand
S-X-40	Velocity Feedback
P-X-610	Real Velocity Feedback
P-X-630, P-X-631	SSC Velocity Feedback
P-X-632	Velocity Difference
S-X-347	Velocity Error
S-X-80	Main Current Demand
S-X-81	Main Current Digital Demand
P-X-640	SSC Current Offset
P-X-641	Torque Correction Offset (from LUT)
P-X-30	Square of Axis Current
P-X-71, P-X-72	SSC Demands
P-X-11, P-X-12	Phase SSC Current Demands
P-X-21, P-X-22	Phase Digital Current Demands
P-X-31, P-X-32, P-X-33	Phase Current Feedback
P-X-51, P-X-52, P-X-53	Phase Current Errors
P-X-61, P-X-62, P-X-63	Phase PWM Outputs
P-X-41, P-X-42, P-X-43	ADC Offsets
P-0-44	ADC 1.5V Reference Input
P-X-606, P-X-607, P-X-608	Analogue Encoder Inputs
P-X-609	Digital Encoder Counts
P-X-600	Raw Encoder Counts
P-X-601	Delta Encoder Counts
P-X-605	Encoder Reference Mark
P-X-650	Commutation Angle
P-0-662	SAM State Variable
P-X-682 & P-X-683	Hall Code and Angle
P-X-620	Tacho Tracker Error
P-X-621	Tacho Tracker DAC Output
P-X-622	Tacho Tracker Output
P-0-107, P-0-108	Test Inputs
P-0-690	Latch Event Flag
P-X-693, P-0-694	Reference Mark and TPP Status
P-X-695	Latched Encoder Value
P-0-211	Enable State
S-X-129	Diagnostics Word
P-X-227	Vbus Input
P-0-231, P-0-232	Temperature Inputs
P-0-243	Fan Pulse Input
P-0-212	Servo Time (inverse)
P-0-241	SSC Checksum Failure Counter

Table 58 – Logable Parameters

P-0-350 SSC Mode

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 0 (Slave)

Description

Setting this parameter to '1' will cause the DSP to act as a SSC bus master. In this configuration, up to three further DSPs can be connected and the parameters of all DSPs can be accessed via a single RS232 link. When acting as an SSC master, the DSP sends zeros on all synchronous demands and sets the amplifier enable bits in all control bytes.

This parameter can only be changed via RS232 and not via the SSC service channel since if an SSC connection exists, the DSP must always be a slave.

The SSC loop must be made in order for the DSP to communicate to the other nodes, i.e. the SSC out of the last node must be connected to the SSC in of this DSP. The state is not flash saved: the DSP is always configured as a slave after a reboot.

P-0-35n Send Byte On Service Channel (n = 1, 2 & 3)

Main Properties

Write Only
Two Byte Fixed Length
Parameter set 0 only

Description

Parameters P-0-351, P-0-352 and P-0-353 are used in SSC master node to send single bytes across the service channel to slaves 1, 2 and 3 respectively.

P-0-400 Reprogram Flash

Main Properties

Write Only
Two Byte Variable Length
Parameter set 0 only

Description

This parameter enables program updates to be downloaded using the ACP interface rather than via the boot loader or the JTAG port. This allows an update to occur without having to open the box to fit jumpers and without having to cycle the power to the SPA2.

To use this feature, the program machine code should be written to P-0-400 in the following format: The first word is ignored, and the rest of the data is arranged in blocks. Each block starts with a three word header which defines the length of the block, the address it should be loaded into and the memory page it should be loaded into (0 for program memory and 1 for data memory). The rest of the block is the data to be loaded. Table 59 shows an example of this format. Note that SPA2 assumes that all data is destined for program memory, so all memory page words should be zero.

Word Number	Contents
1	Ignored
2	Block 1 Length (L1)
3	Block 1 Address
4	Block 1 Memory page
5	Block 1 Data Word 1
6	Block 1 Data Word 2
:	:
5 + L1 – 1	Block 1 Data Word L1
5 + L1	Block 2 Length (L2)
5 + L1 + 1	Block 2 Address
5 + L1 + 2	Block 2 Memory page
5 + L1 + 3	Block 2 Data Word 1
5 + L1 + 4	Block 2 Data Word 2
:	:
5 + L1 + 3 + L2 – 1	Block 2 Data Word L2
5 + L1 + 3 + L2	Block 3 Length (L2)
5 + L1 + 3 + L2 + 1	Block 3 Address
5 + L1 + 3 + L2 + 2	Block 3 Memory page
5 + L1 + 3 + L2 + 3	Block 3 Data Word 1
5 + L1 + 3 + L2 + 4	Block 3 Data Word 2
:	:

Table 59 – Reprogram Data Format

When the IDN is received, data gathering is disabled and the incoming data is loaded into external data memory starting at location 0x100000 (XINTF Zone 6). The routines of the Flash programming API are then called to perform the reprogram of flash. Once, reprogramming is successfully complete, a reset is generated by allowing the internal watchdog to trip. It is for this reason that the watchdog tripped bit will be set by this instruction. If an error occurs during the flash reprogram, a reset will not occur and an error code will be saved in P-0-401. In this event, it is possible that that the Flash memory could have been corrupted and therefore it is important to ensure that a successful re-program is performed before the SPA2 is powered down. Failure to do this could render the automatic boot of the SPA2 useless and it will need re-programming via some other method.

During reprogramming, the SPA2 will not respond to serial communications.

P-0-401 Reprogram Flash Error Code

Main Properties

Read Only
Two Byte Fixed Length
Parameter set 0 only

Description

In the event of an error occurring during flash reprogram, the SPA2 will not be reset and this parameter will contain the relevant error code.

An error code of 1 indicates that there was a check sum error with the program download and the flash re-program was not attempted. Since the flash has not been changed, it is safe to reboot the SPA2 and it will reset to the same code version that was previously running.

Any other non-zero value returned form P-0-401 indicates a flash algorithm error (see reference [14]). In this case, the state of the flash is undetermined and therefore rebooting could result in undefined operation – the reprogram function must be successfully re-run before the DSP is reset.

P-0-402 Download FPGA Program

Main Properties

Write Only
Two Byte Variable Length
Parameter set 0 only

Description

This parameter will receive the configuration file for the Aries FPGA and store it in external memory ready for it to be written to flash with P-0-403. Data gathering is automatically stopped before the data is received.

The data sent must contain a short header consisting of three words: the first word is always zero then the second and third specify the number of data words that follow (least significant word first). The actual configuration data then follows as an array of words that will be sequentially shifted out to the FPGA (most significant bit first) during start up.

P-0-403 Flash FPGA Program

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only

Description

Writing to this parameter will cause the Aries FPGA configuration data (previously stored in external memory with P-0-402) to be written to flash starting at sector G.

Reading this parameter will return the error code generated the last time that a Flash FPGA Program was performed, i.e. the last time P-0-403 was written to. The meaning of the error code is the same as that read by P-0-208 (see page 76 and Table 34).

P-X-500 Real Velocity Feedback Filter Coefficients

Main Properties

Read and Write
Two Byte Variable Length
Parameter set indicates axis.
Default = 16384, 0, 0, 0, 0 (Filter Disabled)
Flash Saved

Description

This parameter defines the coefficients of the IIR filter that is applied to the 'Real' velocity feedback. That is the velocity feedback as derived from the on board hardware inputs (either digital encoder or analogue tacho). Figure 13 shows the position of this filter in the velocity servo.

This is a variable length IDN that contains the five coefficients of the filter in the order: b_0, b_1, b_2, a_1, a_2. The transfer function of the filter is:

$$\frac{Y}{U}(Z) = \frac{b_0 + b_1 Z^{-1} + b_2 Z^{-2}}{1 + a_1 Z^{-1} + a_2 Z^{-2}}$$

Where Y is the filter output and U is the filter input. The coefficients are signed 16 bit Q14 numbers. This means that a value 1.0 corresponds to an integer value of 16384 and the range of each coefficient is almost ±2.

P-X-501 SSC Velocity Feedback Filter Coefficients

Main Properties

Read and Write
Two Byte Variable Length
Parameter set indicates axis.
Default = 16384, 0, 0, 0, 0 (Filter Disabled)
Flash Saved

Description

This parameter defines the coefficients of the IIR filter that is applied to the SSC velocity feedback. That is the velocity feedback received in the SSC synchronous data. Figure 13 shows the position of this filter in the velocity servo.

The format and scaling of this parameter is the same as P-X-500.

P-X-502 Velocity Forward Path Filter 1 Coefficients

Main Properties

Read and Write
Two Byte Variable Length
Parameter set indicates axis.
Default = 16384, 0, 0, 0, 0 (Filter Disabled)
Flash Saved

Description

This parameter defines the coefficients of the first IIR filter that is applied to the output of the velocity PID filter. Figure 13 shows the position of this filter in the velocity servo.

The format and scaling of this parameter is the same as P-X-500.

P-X-503 Velocity Forward Path Filter 2 Coefficients

Main Properties

Read and Write
Two Byte Variable Length
Parameter set indicates axis.
Default = 16384, 0, 0, 0, 0 (Filter Disabled)
Flash Saved

Description

This parameter defines the coefficients of the second IIR filter that is applied to the output of the velocity PID filter. Figure 13 shows the position of this filter in the velocity servo.

The format and scaling of this parameter is the same as P-X-500.

P-X-504 Acceleration Derivation Filter Coefficients

Main Properties

Read and Write
Two Byte Variable Length
Parameter set indicates axis.
Default = 16384, 0, 0, 0, 0 (Filter Disabled)
Flash Saved

Description

This parameter defines the coefficients of the IIR filter that is applied to the derivation of acceleration demand from the velocity demand. Figure 13 shows the position of this filter in the velocity servo.

The format and scaling of this parameter is the same as P-X-500.

P-X-505 Velocity Derivative Term Filter Coefficients

Main Properties

Read and Write
Two Byte Variable Length
Parameter set indicates axis.
Default = 16384, 0, 0, 0, 0 (Filter Disabled)
Flash Saved

Description

This parameter defines the coefficients of the IIR filter that is applied to the derivative term of the velocity servo PID filter. Figure 13 shows the position of this filter in the velocity servo.

The format and scaling of this parameter is the same as P-X-500.

P-X-506 Phase A Current Feedback Filter Coefficients

Main Properties

Read and Write
Two Byte Variable Length
Parameter set indicates axis.
Default = 16384, 0, 0, 0, 0 (Filter Disabled)
Flash Saved

Description

This parameter defines the coefficients of the IIR filter that is applied to the Phase A current feedback value. Figure 13 shows the position of this filter in the velocity servo.

The format and scaling of this parameter is the same as P-X-500.

P-X-507 Phase B Current Feedback Filter Coefficients

Main Properties

Read and Write
Two Byte Variable Length
Parameter set indicates axis.
Default = 16384, 0, 0, 0, 0 (Filter Disabled)
Flash Saved

Description

This parameter defines the coefficients of the IIR filter that is applied to the Phase B current feedback value. Figure 13 shows the position of this filter in the velocity servo.

The format and scaling of this parameter is the same as P-X-500.

P-X-508 Analogue Encoder Sine Filter Coefficients

Main Properties

Read and Write
Two Byte Variable Length
Parameter set indicates axis.
Default = 16384, 0, 0, 0, 0 (Filter Disabled)
Flash Saved

Description

This parameter defines the coefficients of the IIR filter that is applied to the sine channel of the analogue encoder input. Figure 13 shows the position of this filter in the velocity servo.

The format and scaling of this parameter is the same as P-X-500.

P-X-509 Analogue Encoder Cosine Filter Coefficients

Main Properties

Read and Write
Two Byte Variable Length
Parameter set indicates axis.
Default = 16384, 0, 0, 0, 0 (Filter Disabled)
Flash Saved

Description

This parameter defines the coefficients of the IIR filter that is applied to the cosine channel of the analogue encoder input. Figure 13 shows the position of this filter in the velocity servo.

The format and scaling of this parameter is the same as P-X-500.

P-X-510 Analogue Tacho Filter

Main Properties

Read and Write
Two Byte Variable Length
Parameter set indicates axis.
Default = 16384, 0, 0, 0, 0 (Filter Disabled)
Flash Saved

Description

This parameter defines the coefficients of the IIR filter that is applied to the analogue tacho input. Figure 13 shows the position of this filter in the velocity servo.

The format and scaling of this parameter is the same as P-X-500.

P-X-600 Raw Encoder Counts

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter returns the current value of the incremental encoder input (either digital or analogue).

P-X-601 Change In Encoder Counts

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter returns the change in the incremental encoder input during the last servo cycle. This is used as the basis of a velocity measurement.

P-X-602 Encoder Velocity Scale Factor

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 360
Flash Saved

Description

This parameter is a gain that is applied to the encoder difference signal (P-X-601) to give a velocity signal with correct scaling. Figure 13 shows the position of this gain in the axis servo.

If the resolution of the encoder and the maximum speed of the axis are known then P-X-602 can be determined by calculation:

$$P - X - 602 = \frac{32767}{N \times \tau_s \times V_{\max}}$$

Where: N is number of encoder counts per unit distance
τ_s is the servo sampling time in seconds (P-0-247)
V_{max} is the max speed in unit distance per second

For example if a rotary system had 101248 counts per rev and a max speed of 9 rev/sec, then P-X-602 should be 360. If a linear system had 1000 counts per mm and a max speed of 500 mm/s then P-X-602 should be 655.

Note that V_{max} in the above equation corresponds to a full scale velocity demand (32767). Remember that velocity demand is limited to 95% of full scale (31129) so that a peak velocity limit can be implemented. This means that the maximum speed that can be demanded is 95% of V_{max}.

P-X-604 Velocity Loop Decimation Factor

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 1
Flash Saved

Description

This parameter determines the sample period of the velocity servo. It is expressed in servo cycles, thus a value of one means that the velocity servo is updated every servo cycle (as set by P-0-247), a value of two means that the velocity servo is update every other servo cycle etc.

When set to a value greater than one, the velocity feedback signal can be derived from the average of intervening samples by setting bit 10 of P-X-200 (see page 70).

P-X-605 Encoder Reference Mark

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter returns the current state of the digital encoder reference mark.

P-X-606 Analogue Encoder Sine Channel

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter will return the current value of the analogue encoder sine input. The value returned is after filtering (P-X-508, see page 106) and Heydemann compensation (see page 31) has been applied. Figure 13 shows the position of this signal in the overall servo structure.

P-X-607 Analogue Encoder Cosine Channel

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter will return the current value of the analogue encoder cosine input. The value returned is after filtering (P-X-509, see page 106) and Heydemann compensation (see page 31) have been applied. Figure 13 shows the position of this signal in the overall servo structure.

P-X-608 Analogue Encoder Interpolated Counts

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter will return the analogue encoder interpolated angle. The value is returned after the application of the Lissajous map correction (see page 32) and is scaled such that 65536 = 360º. Figure 13 shows the position of this signal in the overall servo structure.

P-X-609 Digital Encoder Raw Counts

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter will return the value of the digital encoder counter. Each edge of the quadrature signal is counted, so the scaling of this parameter is four times the pulse resolution of the encoder. Figure 13 shows the position of this signal in the overall servo structure.

P-X-610 Real Velocity Feedback

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter returns the current velocity feedback value as derived from the on board hardware inputs (i.e. either incremental encoder or analogue tacho). This parameter is always updated regardless of the loop mode.

P-X-611 SSC Velocity Scale Factor

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0
Flash Saved

Description

This parameter is a gain that is applied to the velocity feedback signal that is received via SSC synchronous data. Figure 13 shows the position of this gain in the axis servo.

P-X-612 Acceleration Feed-Forward Gain

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0
Flash Saved

Description

This parameter sets the acceleration feed-forward gain of the velocity servo. Figure 13 shows the position of this gain in the axis servo.

P-X-613 Friction Offset

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0
Flash Saved

Description

This parameter defines a constant that is added to the output of the velocity servo filter (i.e. the main current demand) in the same sign as the combined velocity demand. This is useful for overcoming drives with a lot of viscous friction. Figure 13 shows the position of this offset in the axis servo. Since this offset is applied to the current loop input, scaling is the same as the current loop, i.e. 32768 = I_{FSD} as defined with P-0-207 (see page 74).

P-X-614 AFO Acceleration Threshold

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 1000
Flash Saved

Description

This parameter sets the minimum acceleration value for which the AFO algorithm will allow the offset to be applied. See page 32 for a description of the AFO algorithm.

P-X-615 AFO Decay Time Constant

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 1985 (0.1985 seconds for default P-0-247)
Flash Saved

Description

This parameter sets the rate at which the AFO offset decays after it has been applied. This is specified as a first order time constant in servo cycles, i.e. after the time period specified by this parameter, the offset will have fallen to 36.8% of its initial value. See page 32 for a description of the AFO algorithm.

P-X-616 AFO Offset

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0
Flash Saved

Description

This parameter sets the magnitude of the offset that is applied by the AFO algorithm – see page 32 for a description of this algorithm. Since this offset is applied to the current loop input, scaling is the same as the current loop, i.e. 32768 = I_{FSD} as defined with P-0-207 (see page 74).

P-X-617 Demand Acceleration

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter returns the acceleration demand that is derived from the main velocity demand using differentiation and a filter. If the SSC demands are enabled (bit 12 of P-X-200 set) the acceleration calculation is only performed when a new SSC demand arrives. This is to prevent a noisy result when the incoming demands arrive more slowly than the servo rate of 100µs. When the SSC demands are disabled, the calculation is performed every servo cycle. Figure 13 shows the position of this signal in the axis servo.

P-X-618 Acceleration Derivation Gain

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 10000 (for a sample period of 100µs)
Flash Saved

Description

This parameter sets the scaling gain for the differentiation that is performed on the velocity demand to obtain the acceleration demand. So that both demands conform to the same scaling, this parameter should be set to the inverse of the sampling period used in the differentiation. The default of 10000 is set for the standard SPA2 servo period of 100µs. However, if the SSC demands are enabled and are arriving more slowly than 100µs, P-X-618 should be changed accordingly. For example, if SSC messages were arriving every 500µs (standard for UCC2), P-X-618 should be set to 2000. Figure 13 shows the position of this gain in the axis servo.

P-X-620 Tacho Tracker Error

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter returns the error signal of the tacho tracker. That is, the signal received by the ADC input that acts as the input to the software part of the tracker.

P-X-621 Tacho Tracker DAC Output

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter returns the tacho trackers estimate of the incoming tacho signal. That is the signal that is applied to the DAC. Figure 13 shows the position of this signal in the axis servo.

P-X-622 Hi Res Tacho Reading

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter returns the output of the tacho tracker. That is the combination of the tracker estimate and the error. As long as the error is not saturated, this signal will be a high resolution representation of the analogue voltage applied to the tacho input. Figure 13 shows the position of this signal in the axis servo.

P-X-623 Tacho Scale Factor

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 16384 (1.0)
Flash Saved

Description

This parameter is a gain that is applied to the tacho tracker output (P-X-622) to give a velocity signal with correct scaling. Figure 13 shows the position of this gain in the axis servo.

Note that this parameter is a signed 16 bit Q14 number. This means that a value 1.0 corresponds to an integer value of 16384 and the range of the gain is almost ±2. This gain is intended for fine adjustments of tacho scaling only, coarse adjustments should be made using the AGS settings (P-X-207, page 74). To avoid saturation of the analogue inputs, P-X-623 should always be greater than one.

If the sensitivity of the tacho and the maximum speed of the axis is known then the tacho scaling factor (K_{VT}), and hence P-X-623, can be determined by calculation:

$$K_{VT} = \frac{10}{K_T \times V_{\max}}$$

Where: K_T is the sensitivity of the tacho in volts per unit distance per second
V_{max} is the max speed in unit distance per second

To find a value for P-X-623 from K_{VT}, first use Table 33 (page 75) to determine K_V, which is the AGS tacho gain closest, but less than, K_{VT}. The require value is then given by:

$$P{-}X{-}623 = \frac{K_{VT}}{K_V}$$

For example, for a rotary system that has a tacho sensitivity of 14V per 1000 RPM and a max speed of 2000RPM, K_{VT} is 0.357, K_V is 0.346 (3.46 ÷10 attenuation) and P-X-623 is 1.0322 or 16912. In a linear system that has a tacho sensitivity of 3mV per mm/s and a max speed of 500mm/s, K_{VT} is 6.67, K_V is 6.49 and P-X-623 is 1.0272 or 16830.

Note that V_{max} in the above equation corresponds to a full scale velocity demand (32767). Remember that velocity demand is limited to 95% of full scale (31129) so that a peak velocity limit can be implemented. This means that the maximum speed that can be demanded is 95% of V_{max}.

P-X-630 SSC Velocity Feedback

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter returns the value of the raw SSC velocity feedback signal. This signal is received as the second word in the SSC message from the master (SSC Demand 'B') when the selector is set to 'SSC Velocity Feedback' (P-X-242 = 2). Alternatively, if the selector is set to 'SSC Position Feedback' (P-X-242 = 3), the signal is the derivative of 'SSC Demand B'. The signal is zero for any other value of P-X-242. Figure 13 shows the position of this signal in the axis servo.

P-X-631 Filtered SSC Velocity Feedback

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter returns the value of the SSC velocity feedback signal after it has been filtered (by P-X-500) and had a gain applied (P-X-611). Figure 13 shows the position of this signal in the axis servo.

P-X-632 Velocity Difference Between Real & SSC

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter returns the difference between the chosen on board velocity input (encoder or tacho) and the SSC velocity feedback signal. This difference is used by the Coupled Mass gain (P-X-633) and can be used as in input to the Velocity Error Time fault. Figure 13 shows the position of this signal in the axis servo.

P-X-633 Coupled Mass Gain

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0
Flash Saved

Description

This parameter sets the gain that is applied the velocity difference and the result is applied to the velocity servo output. This helps to reject oscillations due to non-stiffly coupled masses, see page 33 for more details. Figure 13 shows the position of this gain in the axis servo.

P-X-640 SSC Current Offset

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter returns the SSC current offset value which is derived from the second SSC demand if P-X-242 is set to 1 (see page 92). Figure 13 shows the position of this signal in the overall servo structure.

P-X-641 Torque Compensation Offset

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter returns the current torque offset value as derived from the torque compensation map. See page 33 for more details about torque compensation.

P-X-642 Torque Compensation Map

Main Properties

Read and Write
Two Byte Variable Length
Parameter set indicates axis.
Default = All –1

Description

When written to, this parameter will receive the torque compensation look up tables for the specified axis and store it in external memory ready for it to be written to flash with P-X-643. Data gathering is automatically stopped before the data is received. The data should be arranged as follows: The positive going map is stored first and should be $2^N + 1$ words long (N is given by P-0-644) – i.e. the motor rotation is divided into 2^N points plus an extra point which is a repeat of the first point. The negative going map is stored directly after and is the same length.

Reading this parameter will return the look up tables from each axis in the same format as above.

P-X-643 Flash Torque Compensation Map

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.

Description

Writing to this parameter will cause the torque compensation look up tables for the specified axis (previously stored in external memory with P-X-642) to be written to flash memory.

Reading this parameter will return the error code generated the last time that a Flash Torque Compensation Map was performed, i.e. the last time any P-X-643 was written to. The meaning of the error code is the same as that read by P-0-208 (see page 76 and Table 34)

P-0-644 Size Of Torque Compensation Map

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 7 (map is 129 entries long)

Description

This parameter sets or returns the size of the torque compensation maps. The size of each map is actually $2^N + 1$ (N is the value of this parameter) as described under P-X-642, page 116. The download size is actually twice this number of words because there are two maps – one for each direction.

P-X-650 Commutation Angle

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter returns the current commutation angle as determined from the digital encoder input. The units are 65536 = 360 electrical degrees. The parameter can be written to reset the commutation angle. Figure 13 shows the position of this signal in the axis servo.

P-X-651 Encoder Counts Per Revolution

Main Properties

Read and Write
Four Byte Fixed Length
Parameter set indicates axis.
Default = 101248
Flash Saved

Description

This parameter sets the number of encoder counts per revolution of the motor. This is required, along with the number of motor poles (P-X-652), to calculate commutation angle. In actuality, only one piece of information is required, that is number of counts per pole. In reality, this is not necessarily an integer and rounding could cause large errors in commutation angle to occur after many revolutions. Being able to specify counts per rev and poles per rev separately (both are always integers) eliminates any rounding errors.

Note that for a linear motor the concept of ‘revolution’ is meaningless and so the number of counts per pole could be entered for P-X-651 and P-X-652 set to one. However, the rounding problem described above still applies and by setting P-X-652 to something other than one and P-X-651 accordingly, any rational number for counts per pole can be set.

P-X-652 Number Of Pole Pairs

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 3
Flash Saved

Description

This parameter sets the number pole pairs (or commutation cycles) per revolution of the motor. This is required, along with the number of encoder counts per rev (P-X-651), to calculate commutation angle. See P-X-651 for more details about the purpose of these two parameters.

Note in the case of a linear motor, this parameter can be thought of as: ‘the number of pole pairs (or commutation cycles) covered by the encoder counts specified in P-X-651’.

P-X-653 Commutation Set Up Offset

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 8192
Flash Saved

Description

This parameter is used as the final value of the ramp in the commutation pole search algorithm. See page 39 for more details.

P-X-654 Commutation Set Flag

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0

Description

This parameter is a flag that indicates that a valid pole search algorithm has been completed. When set to zero, with the axis in a commutated mode, the Commutation Not Set fault will be active.

This flag is automatically set during the commutation pole search algorithm and can be set by the host if the commutation is set externally. In addition, the host could reset the flag if anything occurs that could compromise the validity of commutation (e.g. an encoder fault). This parameter is always reset upon power up and is not flash saveable.

P-X-655 CAS Flag

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0

Description

Setting this parameter to '1' enables the Commutation Angle Servo (CAS) which can be used to improve the accuracy of a commutation pole search. The CAS replaces the normal link between encoder position and commutation calculator as shown in Figure 29.

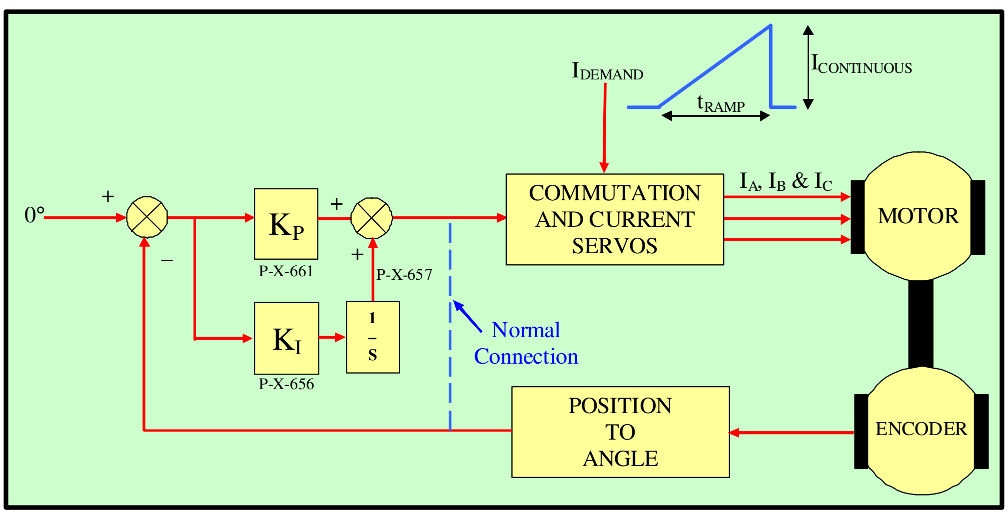

Figure 29 – Commutation Angle Servo

Once the servo has been enabled, a ramp current demand should be applied to the main current input. At the end of the ramp, the output of the servo should be the actual commutation angle.

Although this method of setting commutation is very accurate, it suffers form a couple of drawbacks:

- The P and I gains have to be separately tuned for each axis
- The servo goes unstable if the initial guess of commutation angle is near 180° away from the actual angle. The CAS can only be used once the angle is roughly known (i.e. after using the SAM method).

Because of these drawbacks, the CAS method will be rarely used, but since the implementation is simple, it is left in the firmware in case a high accuracy pole search is required.

P-X-656 CAS Integral Gain

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 10
Flash Saved

Description

This parameter sets the integral gain (K_I) of the Commutation Angle Servo as described in P-X-655.

P-X-657 CAS Integral Output

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 32768 (180°)

Description

This parameter returns the current stored value of the integrator in the Commutation Angle Servo as described in P-X-655. This should be poked with a suitable starting value before the servo is activated.

P-X-658 Commutation Disconnect Flag

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0

Description

Setting this parameter to '1' will remove the connection between the encoder input and the commutation calculator (the blue dashed line shown in Figure 29). It is used by the pole search algorithm.

P-X-659 Movement Sensor Threshold

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0

Description

A non-zero value in this parameter enables the movement sensor as used by the pole search algorithm. When the axis detects movement of the encoder by more counts than that specified in P-X-659, the axis and the movement sensor itself is disabled. The actual amount of movement detected is returned in P-X-660.

P-X-660 Movement Sensor Moved Amount

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0

Description

This parameter returns the amount of movement, in encoder counts, detected by the movement sensor used by the pole search algorithm.

P-X-661 CAS Proportional Gain

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0
Flash Saved

Description

This parameter sets the proportional gain (K_P) of the Commutation Angle Servo as described in P-X-655.

P-0-662 SAM State Variable

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 0 (DISABLED)
Logable

Description

This parameter returns the state variable of the state machine that controls the pole search algorithm. See page 39 for more details. The algorithm is started by writing a ‘1’ to P-X-662, i.e. forcing the state machine into the GENERATE RAMP state. Table 60 shows the mapping between the values of P-X-662 and state names.

P-X-662	STATE
0	DISABLED
1	GENERATE RAMP
2	SETUP
3	RESET RAMP
4	START RAMP
5	WAIT FOR RAMP
6	DETERMINATOR
7	LOCK TO POLE
8	WAIT FOR LOCK
9	TERMINATE

Table 60 – SAM State Machine States

P-0-663 SAM Active Axis

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 0

Description

This parameter is used to set which axis is used in the pole search algorithm. A value of ‘0’ indicates axis 1 and ‘1’ indicates axis 2, etc. This parameter should be set prior to the SAM state machine being started.

P-X-670 Heydemann Compensation – p

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0
Flash Saved

Description

This parameter is the channel 1 offset parameter (p) as used by the Heydemann compensation technique for the analogue encoder input. See description starting on page 31 for more details.

P-X-671 Heydemann Compensation – q

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0
Flash Saved

Description

This parameter is the channel 2 offset parameter (q) as used by the Heydemann compensation technique for the analogue encoder input. See description starting on page 31 for more details.

P-X-672 Heydemann Compensation – G

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 16384 (1.0)
Flash Saved

Description

This parameter is the gain parameter (G) as used by the Heydemann compensation technique for the analogue encoder input. See description starting on page 31 for more details. Parameter is in Q14 fixed point scaling (i.e. 16384 represents 1.0).

P-X-673 Heydemann Compensation – sin(a)

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0
Flash Saved

Description

This parameter is the sine of the quadrature error (a) as used by the Heydemann compensation technique for the analogue encoder input. See description starting on page 31 for more details. Parameter is in Q14 fixed point scaling (i.e. 16384 represents 1.0).

P-X-674 Heydemann Compensation – 1/cos(a)

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 16384 (1.0)
Flash Saved

Description

This parameter is the reciprocal of the cosine of the quadrature error (a) as used by the Heydemann compensation technique for the analogue encoder input. See description starting on page 31 for more details. Parameter is in Q14 fixed point scaling (i.e. 16384 represents 1.0).

P-X-675 Lissajous Correction LUT

Main Properties

Read and Write
Two Byte Variable Length
Parameter set indicates axis.
Default: All Zeros
Flash Saved

Description

This parameter allows the upload and download of the Lissajous compensation map. This can be used as well or instead of Heydemann compensation to correct analogue encoder signals. See description starting on page 32 for details. The map is 17 words long – the Lissajous is divided into 16 sections and the last word is a repeat of the first.

P-X-680 Commutation Mode

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0 (Encoder Mode)
Flash Saved

Description

This parameter defines the type of commutation used for a brushless motor. It has three possible values as shown in below:

Mode	P-X-680
Encoder	0
Hall	1
Encoder Reset From Hall	2

Table 61 – Commutation Modes

See page 29 for a description of commutation.

P-X-681 Hall Offset

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0 (Encoder Mode)
Flash Saved

Description

This parameter is the angular offset of the Hall sensors within the commutation cycle of a brushless motor. It is effectively the desired commutation angle at the first Hall transition (i.e. the transition from code 5 to 1). See description on page 30 for more details.

P-X-682 Hall Code

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter returns the current state of the Hall sensors in a brushless motor. The hall sensors should be set up to follow a grey code sequence. All zeros and all ones are not allowed so a six code sequence will occur:

I/P 1	I/P2	I/P 3	P-X-682	Angle
0	0	1	1	0°
0	1	1	3	60°
0	1	0	2	120°
1	1	0	6	180°
1	0	0	4	240°
1	0	1	5	360°

Table 62 – Hall Code Sequence (Positive Rotation)

If the motor direction reverses, the above sequence is inverted. See description on page 30 for more details.

P-X-683 Hall Angle

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter returns electrical motor angle as determined from the Hall sensors in a brushless motor. The parameter is scaled such that 65536 = 360°. Table 62 shows the mapping between Hall code and angle, see description on page 30 for more details.

P-X-684 Hall Direction

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 1
Flash Saved

Description

This parameter sets the direction in which the Hall codes count relative to motor rotation. This must coincide with the phasing of the motor in order for Hall based commutation to work. See description on page 30 for more details.

P-0-690 Latch Event Flag

Main Properties

Read Only
Two Byte Fixed Length
Parameter set 0 only
Logable

Description

This parameter returns the current value of the ‘Latch Event Flag’ which is a latch on any reference mark input or the Aries TTP input. The flag is also reported via bit 12 in the SSC status byte. The flag is read only, it can only be reset by resetting the status flag or flags of whatever is causing the latch event. See page 37 for more details concerning the reference mark and TTP inputs.

P-X-691 Reference Mark Edge Control

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0
Flash Saved

Description

Bits 0 and 1 of this parameter define the edges on which the encoder input reference mark will be active: bit 0 enables the capture on the rising edge of the associated reference mark and bit 1 enables for the falling edge. Setting both bits causes capture to occur on both edges. Setting neither, effectively disables reference mark capture. All other bits are not used.

See page 37 for more details concerning the reference mark and TTP inputs.

P-0-692 TTP Edge Control

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 0
Flash Saved

Description

Bits 0 and 1 of this parameter define the edges on which the Touch Trigger Probe input on an Aries SPA will be active: bit 0 enables the capture on the rising edge of the associated reference mark and bit 1 enables for the falling edge. Setting both bits causes capture to occur on both edges. Setting neither, effectively disables reference mark capture.

See page 37 for more details concerning the reference mark and TTP inputs.

P-X-693 Reference Mark Status

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter contains status information concerning the encoder reference mark input for this axis. There are three active bits: bit 0 is the instantaneous value of the reference mark input, bit 1 is a latched version of the reference mark input and bit 2 is an overflow flag. The latched bit will be set when a valid edge occurs and it must be reset by the host. The overflow bit will be set if a valid edge occurs when the latched bit is already set. The overflow flag can also be cleared by the host and bits are reset by writing zeros to them.

See page 37 for more details concerning the reference mark and TTP inputs.

P-0-694 TTP Status

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Logable

Description

This parameter contains status information concerning the Touch Trigger Probe input on an Aries SPA. There are three active bits: bit 0 is the instantaneous value of the TTP input, bit 1 is the logical AND of the latched reference mark inputs of all axes (i.e. bit 1 of all P-X-693) and bit 2 is the logical AND of the reference mark overflow flag of all axes (i.e. bit 2 of all P-X-693).

If this parameter is written to, it is copied to the reference mark status register (P-X-693) of all the axes. Hence all latched and overflow flags can be reset with a single write.

See page 37 for more details concerning the reference mark and TTP inputs.

P-X-695 Latched Event Value

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter returns the encoder value as latched by the latest latch event – either the encoder reference mark for this axis or a TTP input (Aries SPA only). For reference mark inputs, only the latched value of the relevant axis is updated. For the TTP input, all axis latched positions are updated.

See page 37 for more details concerning the reference mark and TTP inputs.

P-X-700 Position Loop Options Word

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0x6
Flash Saved

Description

This parameter contains the various switches that control the modes of the position loop. Table 63 shows the structure of the options word. See page 36 and Figure 15 for more details concerning the position loop.

15	14	13	12	11	10	9	8
Not Used							
7	6	5	4	3	2	1	0
Not Used				Traj. Calc Enable	Position Encoder Source		Enable

Table 63 – Position Loop Options Word

The definition of the individual bits is given below

Bit 0 Position Loop Enable:
0 – Position Loop Enabled
1 – Position Loop Disabled

Bits 1-2 Position Feedback Encoder Source:

Bit 2	Bit 1	Source
0	0	Digital Encoder Input
0	1	Analogue Encoder Input
1	1	SSC Synchronous Demand 'B'
1	0	No Source Selected

Table 64 – Position Encoder Source Bits

Bit 3 Trajectory Calculator Enable
0 – Trajectory Calculator Disabled
1 – Trajectory Calculator Enabled

Bits 4-15 Not Used

P-X-701 Position Loop Error

Main Properties

Read Only
Two Byte Fixed Length
Parameter set indicates axis.
Logable

Description

This parameter will return the value of the position loop servo error, i.e. the difference between the combined position demand and feedback. See page 36 and Figure 15 for more details concerning the position loop.

P-X-702 Position Combined Demand

Main Properties

Read Only
Four Byte Fixed Length
Parameter set indicates axis.
Logable (lower 16 bits)

Description

This parameter returns the combined demand of the position loop, that is the sum of the position offset (S-X-48, page 55) and the extended test generator output. If the trajectory generator is enabled, the demand is processed by this before being applied to the position loop servo. See page 36 and Figure 15 for more details.

When this parameter is logged with the data gatherer, only the lower 16 bits are stored.

P-X-703 Trajectory Calculator Max Velocity

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 50
Flash Saved

Description

This parameter sets the velocity limit to be used by the trajectory calculator when computing position demands. The velocity is scaled as encoder counts per servo cycle. See page 36 for a description of the trajectory calculator.

P-X-704 Trajectory Calculator Max Acceleration

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 1638
Flash Saved

Description

This parameter sets the acceleration and deceleration limit to be used by the trajectory calculator when computing position demands. The acceleration is scaled as encoder counts per servo cycle squared divided by 65536. See page 36 for a description of the trajectory calculator.

P-X-705 Velocity Feed-Forward Gain

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set indicates axis.
Default = 0
Flash Saved

Description

This parameter sets the velocity feed-forward gain which is applied to the velocity calculated in the trajectory calculator and the result added to the position loop PI filter output. The gain should be set to equate the velocity scaling of the trajectory calculator (encoder counts per servo cycle) with that of the velocity loop input, i.e:

$$K_{VFF} = \frac{32768 \times E_R}{V_{FSD} \times T_S} \times 256$$

Where E_R is the position encoder resolution in distance units and V_{FSD} is the full scale of the velocity loop in distance units per second. T_S is the servo sample time, in seconds, as set in

P-0-247 (see page 94), default is 100µs. The 256 multiply is there because the gain is scaled as a 16 bit Q8 number, i.e. its range is almost ±128.

See page 36 and Figure 15 for more details.

P-0-995 Control Vbus Relay

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only
Default = 1 (Relay Closed)

Description

The Vbus relay can remove the bus voltage from the drive stage of the amplifier. The following conditions must be met in order to close this relay:

- External watchdog must not be tripped (direct hardware control)
- External 'Engage' input must be grounded (direct hardware control)
- EStop relay must be closed (controlled via firmware)
- SSC Comm Lock fault must not be active (controlled via firmware)
- P-X-995 must be '1' (controlled via firmware)

From this it can be seen that setting P-X-995 to '0' will remove bus power whereby setting it to '1' does not necessarily mean that power is restored. The status of the relay can be determined by bit 8 of S-X-129 (Vbus Relay Open fault).

P-0-996 Sensitive Parameter Protection Lock

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only.
Default = 0

Description

The sensitive parameter protection feature is designed to protect sensitive parameters from being accidentally changed. In order to modify such parameters, the 'Key' (0x5AFE) has to be written to the 'Lock' (P-0-996) in the ACP message immediately before the one that changes the sensitive parameter. If it is not, or if another message occurs between the write to the lock and the write to the parameter, no change is made.

Reading this parameter will always return zero except for directly after the key has been written in which case the return will be 0x5AFE (the key!).

Currently, the only parameter protected in this way is P-0-218 (PWM dead time).

P-0-997 SSC Lock Release

Main Properties

Read and Write
Two Byte Fixed Length
Parameter set 0 only

Description

The SSC lock fault is a latched version of the SSC time out fault. The time out fault will disappear as soon as SSC link is restored. The lock fault, however, will remain until reset by writing the number 0x5AFE to this parameter.

Reading this parameter will return one if the lock fault exists and zero if it doesn't.

P-0-998 External Memory Test

Main Properties

Read Only
Two Byte Fixed Length
Parameter set 0 only

Description

Reading this parameter will perform a check of the external memory. Two types of test are conducted: Address Integrity Tests and Data Integrity Tests. For data integrity, every external memory location is written to with the same number and then read back. The test fails if an incorrect read is made. This test is repeated for a series of numbers: 0, 0xFFFF, 0x5555 and 0xAAAA. For address integrity, every memory location is written with a different number and then read back. Each location must read back its correct number in order for the test to pass. Data gathering and the test input generator are both switched off prior to the test.

The parameter will return an error code, each bit of which corresponds to a different test as shown in Table 65. A set bit indicates an error so a zero return means no errors.

Bit	Meaning
0	Data Integrity 0xFFFF fail
1	Data Integrity zero fail
2	Data Integrity 0x5555 fail
3	Data Integrity 0xAAAA fail
4-8	Not Used
9	Address Integrity failed
10-14	Not Used
15	Full memory test not conducted*

Table 65 – External Memory Test Error Code

* When running in development mode with an emulator directly connected to the JTAG port of the DSP (RAM mode), some of the external memory (higher 16K) contains executable code which would be corrupted by a memory test as described above. Therefore, when the code is compiled to run in RAM mode, the memory test only checks the first 48K of external memory. This is signified in the returned error code by bit 15 being set. For more details see the memory map description on page 47.

P-0-999 Trip The Watchdogs

Main Properties

Write Only
Two Byte Fixed Length
Parameter set 0 only

Description

This parameter can be written to force one of the watchdogs to trip. Write '1' to trip the internal watch dog and '2' to trip the external watch dog. Obviously, the SPA2 will be reset by these actions.

APPENDIX D – LIST OF SOURCE FILES

File Name	Description
comms.c	Handles McBSP comms from master and other DSPs
commutation_set.c	Handles the commutation pole search algorithm
enable.c	Handles enables, limits and faults
entry.asm	Flash boot loader
fifos.c	Handles service channel FIFOs
flash.c	Provides flash saving functionality
main.c	C program entry & supervisory program
program_fpga.c	Code to configure the Aries FPGA
RS232_SPA2.c	Serial ACP code specific to SPA2
servo.c	ISR that runs the servo filters
setup.c	Code that initialises everything
useful28.asm	Useful general purpose routines

Table 66 – Main Source Files

File Name	Description
ariesFPGA.h	Register bit definitions for the Aries FPGA
atan_table.c	Defines the look up table for atan
atan_table.h	Header file for atan_table.c
c28xx.h	Defines all the 28x register structures
comms.h	Header file for comms.c
commutation_set.h	Header file for commutation_set.c
enable.h	Header file for enable.c
fifos.h	Header file for fifos.c
flash.h	Header file for flash.c
Flash281x_API_Config.h	Header file for Flash API
Flash281x_API_Library.h	Header file for Flash API
Flash2812_API_V210.lib	Flash API Library
RS232.h	Header file for serial ACP code
RS232_Generic.c	Generic serial ACP code
interrupts.c	Default set up of the PIE
rts2800_ml.lib	C Libraries
servo.h	Header file for servo.c
setup.h	Header flle for setup.c
SPA2_Ram.cmd	Link file for RAM loading (defines memory map)
SPA2_Flash.cmd	Link file for FLASH loading (defines memory map)
SPA2.pjt	The project file
useful28.h	Header file for useful28.c
vardef.c	Defines the 28x memory mapped registers

Table 67 – Other Files Used By The Project

APPENDIX E – SSC BUS MASTER USING A 2812 DSK

For development and in the absence of a host controller (e.g. UCC2) it is useful to be able to communicate to the SPA2 directly from a RS232 COM port on a standard PC. This can be achieved using a 2812 eZdsp board (available from Texas Instruments) and a custom built interface PCB containing a RS232 driver chip and LVDS transceiver chips for SSC. Alternatively, a modified SPA2 DSP board (hardware versions 2 and 3 only) or an SPA2 prototype board can be used for a master unit. This appendix briefly describes the code that runs on this module which acts a SSC bus master for up to four SPA2 DSPs. This description is valid for version 0.0.15 of the SPA2 master code.

The master can run in one of two modes: Normal (P-4-1504 = 0) and Extended (P-4-1504 = 1). Normal mode has up to two slave nodes and each message is two TAPs or 11 words long. Extended mode has up to four slave nodes and each message is four TAPs or 21 words long. Note that a checksum is always sent and checked for.

Service Channel Handling

The master code uses the same ACP state machine as used in the main SPA2 code (RS232_Generic.c). This is set up so that IDNs from parameter sets 0 and 1 are sent across the SSC to the first SPA2, IDNs from subsequent parameter sets are renumbered to parameters sets 0 and 1 and sent across the SSC as defined in Table 68.

Parameter Set	Renumbered As	Sent To
0	0	First DSP
1	1	First DSP
2	0	Second DSP
3	1	Second DSP
4	0	Third DSP
5	1	Third DSP
6	0	Fourth DSP
7	1	Fourth DSP

Table 68 – How Master Handles ACP Messages

IDNs from any parameter set with block numbers between 1000 and 1999 are decoded on the master itself. Responses from all DSPs are sent back across RS232 to the PC. Figure 30 shows this diagrammatically. A VB program running on the PC (e.g. MDM Setup) can be used to talk to the SPA2s via the master module

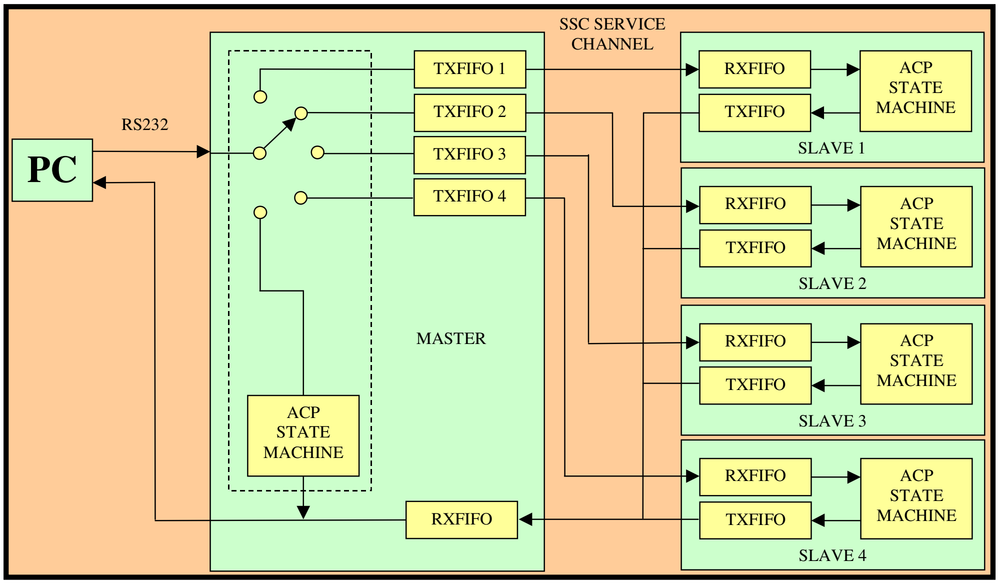

Figure 30 – ACP State Machine Running In Master Mode

Synchronous Data

The master will interface with the synchronous data fields and the control and status bytes of the SSC. These can be specified and reported back using master IDNs. There is also a signal generator built in to the master that can send two sinusoids, out of phase by 120°, to each SPA2 axis. This signal generator function is controlled by further master IDNs.

The enable bit of the control byte is set according to the logical AND of an external input (GPIO B13) and IDN P-4-12X0.

The master module sends an SSC message at a rate set by P-4-1507 (default 100µs). An LED is flashed at 5Hz if a corresponding response is received. If a response is not received, the LED is switched off. This provides an indication of the integrity of the SSC ring.

IDN Descriptions

Table 69 shows the IDNs that are used on the master module. Each IDN is then described. Note that all master IDNs occupy parameter set 4, this is for historical reasons.

IDN	R/W	SIZE	DESCRIPTION	VAR NAME
SYNCHRONOUS DATA				
P-4-1000	R & W	2	TX Amp 1, Axis 1, Data Word 1	TXMessage[5/15*]
P-4-1001	R & W	2	TX Amp 1, Axis 1, Data Word 2	TXMessage[6/16*]
P-4-1002	R & W	2	TX Amp 1, Axis 2, Data Word 1	TXMessage[7/17*]
P-4-1003	R & W	2	TX Amp 1, Axis 2, Data Word 2	TXMessage[8/18*]
P-4-1010	R & W	2	TX Amp 2, Axis 1, Data Word 1	TXMessage[0/10*]
P-4-1011	R & W	2	TX Amp 2, Axis 1, Data Word 2	TXMessage[1/11*]
P-4-1012	R & W	2	TX Amp 2, Axis 2, Data Word 1	TXMessage[2/12*]
P-4-1013	R & W	2	TX Amp 2, Axis 2, Data Word 2	TXMessage[3/13*]
P-4-1020	R & W	2	TX Amp 3, Axis 1, Data Word 1	TXMessage[5]
P-4-1021	R & W	2	TX Amp 3, Axis 1, Data Word 2	TXMessage[6]
P-4-1022	R & W	2	TX Amp 3, Axis 2, Data Word 1	TXMessage[7]
P-4-1023	R & W	2	TX Amp 3, Axis 2, Data Word 2	TXMessage[8]
P-4-1030	R & W	2	TX Amp 4, Axis 1, Data Word 1	TXMessage[0]
P-4-1031	R & W	2	TX Amp 4, Axis 1, Data Word 2	TXMessage[1]
P-4-1032	R & W	2	TX Amp 4, Axis 2, Data Word 1	TXMessage[2]
P-4-1033	R & W	2	TX Amp 4, Axis 2, Data Word 2	TXMessage[3]
P-4-1100	R	2	RX Amp 1, Axis 1, Data Word 1	RXMessage[5/15*]
P-4-1101	R	2	RX Amp 1, Axis 1, Data Word 2	RXMessage[6/16*]
P-4-1102	R	2	RX Amp 1, Axis 2, Data Word 1	RXMessage[7/17*]
P-4-1103	R	2	RX Amp 1, Axis 2, Data Word 2	RXMessage[8/18*]
P-4-1110	R	2	RX Amp 2, Axis 1, Data Word 1	RXMessage[0/10*]
P-4-1111	R	2	RX Amp 2, Axis 1, Data Word 2	RXMessage[1/11*]
P-4-1112	R	2	RX Amp 2, Axis 2, Data Word 1	RXMessage[2/12*]
P-4-1113	R	2	RX Amp 2, Axis 2, Data Word 2	RXMessage[3/13*]
P-4-1120	R	2	RX Amp 3, Axis 1, Data Word 1	RXMessage[5]
P-4-1121	R	2	RX Amp 3, Axis 1, Data Word 2	RXMessage[6]
P-4-1122	R	2	RX Amp 3, Axis 2, Data Word 1	RXMessage[7]
P-4-1123	R	2	RX Amp 3, Axis 2, Data Word 2	RXMessage[8]
P-4-1130	R	2	RX Amp 4, Axis 1, Data Word 1	RXMessage[0]
P-4-1131	R	2	RX Amp 4, Axis 1, Data Word 2	RXMessage[1]
P-4-1132	R	2	RX Amp 4, Axis 2, Data Word 1	RXMessage[2]
P-4-1133	R	2	RX Amp 4, Axis 2, Data Word 2	RXMessage[3]
CONTROL / STATUS				
P-4-1200	R&W	2	Amp 1, Enable Control	TXMessage[9/19*], Bit 15
P-4-1201	R	2	Amp 1, Enabled Status	RXMessage[9/19*], Bit 15
P-4-1202	R	2	Amp 1, Amp OK Status	RXMessage[9/19*], Bit 14
P-4-1203	R	2	Amp 1, ESTOP OK Status	RXMessage[9/19*], Bit 13
P-4-1204	R	2	Amp 1, Latch Event Flag	RXMessage[9/19*], Bit 12
P-4-1209	R	2	Amp 1, Status Word	RXMessage[9/19*]
P-4-1410	R & W	2	Amp 1, Latched Status Bits	LatchedAmpStatus[0]
P-4-1210	R&W	2	Amp 2, Enable Control	TXMessage[4/14*], Bit 15
P-4-1211	R	2	Amp 2, Enabled Status	RXMessage[4/14*], Bit 15
P-4-1212	R	2	Amp 2, Amp OK Status	RXMessage[4/14*], Bit 14
P-4-1213	R	2	Amp 2, ESTOP OK Status	RXMessage[4/14*], Bit 13
P-4-1214	R	2	Amp 2, Latch Event Flag	RXMessage[4/14*], Bit 12
P-4-1219	R	2	Amp 2, Status Word	RXMessage[4/14*]
P-4-1411	R & W	2	Amp 2, Latched Status Bits	LatchedAmpStatus[1]
P-4-1220	R&W	2	Amp 3, Enable Control	TXMessage[9], Bit 15
P-4-1221	R	2	Amp 3, Enabled Status	RXMessage[9], Bit 15
P-4-1222	R	2	Amp 3, Amp OK Status	RXMessage[9], Bit 14
P-4-1223	R	2	Amp 3, ESTOP OK Status	RXMessage[9], Bit 13
P-4-1224	R	2	Amp 3, Latch Event Flag	RXMessage[9], Bit 12
P-4-1229	R	2	Amp 3, Status Word	RXMessage[9]
P-4-1412	R & W	2	Amp 3, Latched Status Bits	LatchedAmpStatus[2]
P-4-1230	R&W	2	Amp 4, Enable Control	TXMessage[4], Bit 15
P-4-1231	R	2	Amp 4, Enabled Status	RXMessage[4], Bit 15
P-4-1232	R	2	Amp 4, Amp OK Status	RXMessage[4], Bit 14
P-4-1233	R	2	Amp 4, ESTOP OK Status	RXMessage[4], Bit 13
P-4-1234	R	2	Amp 4, Latch Event Flag	RXMessage[4], Bit 12

IDN	R/W	SIZE	DESCRIPTION	VAR NAME
P-4-1239	R	2	Amp 4, Status Word	RXMessage[4]
P-4-1413	R & W	2	Amp 4, Latched Status Bits	LatchedAmpStatus[3]
		SIGNAL GENERATOR		
P-4-1300	R & W	2	Signal Generator Enable	SigGenEnabled
P-4-1301	R & W	2	Signal Generator Active Axes	SigGenAxes
P-4-1302	R & W	2	Signal Generation Amplitude	SigGenAmplitude
P-4-1303	R & W	2	Signal Generation Update Period	UpdatePeriod
		TEST		
P-4-1401	W	2	Send single byte to Amp 1 S.C.	-
P-4-1402	W	2	Send single byte to Amp 2 S.C.	-
P-4-1403	W	2	Send single byte to Amp 3 S.C.	-
P-4-1404	W	2	Send single byte to Amp 4 S.C.	-
		ENCODER INPUT		
P-4-1600	R & W	2	Encoder 1 Destination Word	Encoder[0].Destination
P-4-1601	R	2	Encoder 1 Raw Counts	Encoder[0].Raw
P-4-1602	R	2	Encoder 1 Delta Counts	Encoder[0].Delta
P-4-1650	R & W	2	Encoder 2 Destination Word	Encoder[1].Destination
P-4-1651	R	2	Encoder 2 Raw Counts	Encoder[1].Raw
P-4-1652	R	2	Encoder 2 Delta Counts	Encoder[1].Delta
		MISCELLANEOUS		
P-4-1505	R	V2	Firmware Version	Version
P-4-1506	R	V2	Amplifier Type	AmpType
P-4-1501	R & W	2	SSC Clock Speed	McbspRegs.SRGR1.bit.CLKGDV
P-4-1502	R & W	2	SSC Data Delay	McbspRegs.RCR2.bit.RDATDLY
P-4-1503	R	2	External Enable Input	GPIO B13
P-4-1504	R & W	2	Extended Mode Select	-
P-4-1507	R & W	4	SSC Message Period	CpuTimer1Regs.PRD
P-4-1508	R & W	2	Checksum Fail Counter	CSFailCount
P-4-1509	R & W	2	Overflow Failure Counter	OverflowCount

Table 69 – Summary Of IDNs Used By Master

* Array index depends on mode: normal/extended

P-4-10XY SSC TX Synchronous Data

Main Properties

Read and Write
Two Byte Fixed Length
X = 0, 1, 2, 3 for slaves 1, 2, 3, 4 respectively
Y = 0-3 for words 1-4 of the TAP
Default = 0

Description

This set of parameters will define the data that is sent in the SSC synchronous data fields (first four words of each TAP).

P-4-11XY SSC RX Synchronous Data

Main Properties

Read Only
Two Byte Fixed Length
X = 0, 1, 2, 3 for slaves 1, 2, 3, 4 respectively
Y = 0–3 for words 1–4 of the TAP

Description

This set of parameters returns the data received via the SSC synchronous data fields (first four words of each TAP).

P-4-12X0 SSC Amplifier Enable Bit

Main Properties

Read and Write
Two Byte Fixed Length
X = 0, 1, 2, 3 for slaves 1, 2, 3, 4 respectively
Default = 1 (Enable)

Description

This parameter helps to determine the state of the amplifier enable bit (bit 7) in the control word. The enable bit will be the logical AND of this parameter and the external enable input (GPIO B13, reported by P-4-1503, page 142).

P-4-12X1 SSC Amplifier Enabled Bit

Main Properties

Read Only
Two Byte Fixed Length
X = 0, 1, 2, 3 for slaves 1, 2, 3, 4 respectively

Description

This parameter returns the state of the amplifier enabled bit (bit 7) in the status word.

P-4-12X2 SSC Amplifier OK Bit

Main Properties

Read Only
Two Byte Fixed Length
X = 0, 1, 2, 3 for slaves 1, 2, 3, 4 respectively

Description

This parameter returns the state of the amplifier OK bit (bit 6) in the status word.

P-4-12X3 SSC ESTOP OK Bit

Main Properties

Read Only
Two Byte Fixed Length
X = 0, 1, 2, 3 for slaves 1, 2, 3, 4 respectively

Description

This parameter returns the state of the ESTOP OK bit (bit 5) in the status word.

P-4-12X4 Latch Event Flag

Main Properties

Read Only
Two Byte Fixed Length
X = 0, 1, 2, 3 for slaves 1, 2, 3, 4 respectively

Description

This parameter returns the state of the Latch Event bit (bit 4) in the status word.

P-4-12X9 Status Word

Main Properties

Read Only
Two Byte Fixed Length
X = 0, 1, 2, 3 for slaves 1, 2, 3, 4 respectively

Description

This parameter returns the whole status word (including the service channel byte) for the specified node.

P-4-1300 Signal Generator Enable

Main Properties

Read and Write
Two Byte Fixed Length
Default = 0 (Disabled)

Description

This parameter defines whether the signal generator functionality of the master module is enabled (1 for enabled, 0 for disabled). When enabled, the master will send two sinusoids, out of phase by 120°, across the synchronous data fields of the SSC to the axes specified by P-4-1301.

P-4-1301 Signal Generator Active Axes

Main Properties

Read and Write
Two Byte Fixed Length
Default = 0x1 (Amp 1, Axis 1 only)

Description

This parameter determines which axes will receive the sinusoidal output of the signal generator function. Table 70 shows the structure of this parameter.

15	14	13	12	11	10	9	8
Not Used							
7	6	5	4	3	2	1	0
Amp 4 Axis 2	Amp 4 Axis 1	Amp 3 Axis 2	Amp 3 Axis 1	Amp 2 Axis 2	Amp 2 Axis 1	Amp 1 Axis 2	Amp 1 Axis 1

Table 70 – Signal Generator Active Axes Bit Specification

P-4-1302 Signal Generator Amplitude

Main Properties

Read and Write
Two Byte Fixed Length
Default = 4000

Description

This parameter defines the amplitude of the sinusoids produced by the signal generator function.

P-4-1303 Signal Generator Update Period

Main Properties

Read and Write
Two Byte Fixed Length
Default = 5 (2 kHz)

Description

This parameter defines the sample period of the sinusoids produced by the signal generator function. The period is expressed in numbers of SSC Messages, i.e. multiples of 100µs. Note that there are 512 samples in each sine wave so the period of the sinusoids produced is given by:

$$f_{SIN} = \frac{1}{P\text{–}4\text{–}1303 \times 100\mu \times 512}$$

Given this, the default sinusoidal frequency is 3.9 Hz.

P-4-140X Slave X Service Channel Test Byte

Main Properties

Write Only
Two Byte Fixed Length
X = 1, 2, 3, 4 for slaves 1, 2, 3, 4 respectively

Description

A byte written to these parameters will be sent immediately across the service channel to the relevant slave. This is useful when initialising and synchronising the ACP state machines on the master and slaves.

P-4-141X Slave X Latched Amp Status Bits

Main Properties

Read and Write
Two Byte Fixed Length
X = 0, 1, 2, 3 for slaves 1, 2, 3, 4 respectively

Description

The top three bits of each nodes status byte (Amp Enabled, Amp OK and ESTOP OK) are inverted (so Amp Disabled, Amp Fault and ESTOP Fault) and latched. The latched value is returned by this parameter. Latched status can be cleared by writing a zero.

P-4-1501 SSC Clock Speed

Main Properties

Read and Write
Two Byte Fixed Length
Default = 3 (37.5 Mbps)

Description

This parameter determines the clock speed of the SSC link. It is given by:

$$Mbps = \frac{150}{P\text{-}4\text{-}1501 + 1}$$

P-4-1501 directly maps to the CLKGDV bits of the SRGR1 McBSP register. See reference [8] for more details.

P-4-1502 SSC Data Delay

Main Properties

Read and Write
Two Byte Fixed Length
Default = 1

Description

This parameter determines the number of bits delay between the SSC frame sync signal and the data. The default of one means that the first bit of data occurs in the clock cycle following the frame sync. P-4-1502 directly maps to the RDATDLY and XDATDLY bits of RCR2 and XCR2 McBSP registers. See reference [8] for more details.

P-4-1503 External Enable Input

Main Properties

Read Only
Two Byte Fixed Length

Description

This parameter returns the value of the external enable input as read on GPIO input B13. This value is logically ANDed with the individual amplifier enables (P-4-12X0) to set the SSC amplifier enable bit (bit 7) in the transmitted control bytes.

P-4-1504 SSC Loop Mode

Main Properties

Read and Write
Two Byte Fixed Length
Default = 0 (Normal mode)

Description

This parameter defines the mode that the SSC loop runs in. Zero indicates 'Normal' mode, i.e. up to two slave nodes. One indicates 'Extended' mode i.e. up to four slave nodes.

P-4-1505 Firmware Version

Main Properties

Read Only
Two Byte Variable Length

Description

Returns an ASCII string (one character per word) containing firmware version of the master code.

P-4-1506 Amplifier Type

Main Properties

Read Only
Two Byte Variable Length

Description

Returns an ASCII string (one character per word) containing the amplifier type. This always returns the string "SPA2-MASTER".

P-4-1507 SSC Message Period

Main Properties

Read & Write
Four Byte Fixed Length
Default = 15000 (100µs)

Description

This parameter determines the rate at which the master sends messages across the SSC link. The period, in microseconds, is given by:

$$T_{SSC} = \frac{P\text{–}4\text{–}1507}{150}$$

P-4-1507 directly maps to the CPU Timer 1 Period register. See reference [5] for more details.

P-4-1508 Checksum Failure Counter

Main Properties

Read & Write
Two Byte Fixed Length

Description

This parameter returns the number of checksum failures that have occurred since the counter was last reset. It can be reset by writing zero.

P-4-1509 Overflow Failure Counter

Main Properties

Read & Write
Two Byte Fixed Length

Description

If the response from the slaves takes longer than the message period (set by P-4-1507), the Overflow Failure Counter is incremented. This parameter returns the current value of the counter and it can be reset by writing zero.

P-4-1600 Encoder 1 Destination Word

Main Properties

Read & Write
Two Byte Fixed Length
Default = 0

Description

This parameter defines whether or not the result of the DSP's first encoder input (QEP1) is passed on to one of the slaves as an SSC synchronous demand. Either position or derived velocity can be sent to the second data word of any of the eight slave axes. Table 71 shows the structure of this parameter.

15	14	13	12	11	10	9	8
Position / Velocity	Enable	Not Used					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Not Used					Destination Axis		

Table 71 – Encoder Destination Word

The definition of individual bits is given below:

Bit 0-2 Destination Axis
Specifies which axis (0-7) the encoder signal is sent to. Always sent as the second data word.

Bit 3-13 Not Used

Bit 14 Enable Bit
0 – Encoder signal is not sent
1 – Encoder signal is sent across SSC Link

Bit 15 Position/Velocity Select
0 – Difference in encoder signal since last message is sent (velocity)
1 – Raw encoder data is sent (position)

P-4-1601 Encoder 1 Raw Counts

Main Properties

Read Only
Two Byte Fixed Length

Description

This parameter returns the current output of the DSP's first encoder input (QEP1)

P-4-1602 Encoder 1 Delta Counts

Main Properties

Read Only
Two Byte Fixed Length

Description

This parameter returns the change in the DSP's first encoder input (QEP1) over the last SSC message period. This gives an indication of velocity.

P-4-1650 Encoder 2 Destination Word

Main Properties

Read & Write
Two Byte Fixed Length
Default = 0

Description

This parameter defines whether or not the result of the DSP's second encoder input (QEP2) is passed on to one of the slaves as an SSC synchronous demand. The format of this parameter is the same a P-4-1600.

P-4-1651 Encoder 2 Raw Counts

Main Properties

Read Only
Two Byte Fixed Length

Description

This parameter returns the current output of the DSP’s second encoder input (QEP2)

P-4-1652 Encoder 2 Delta Counts

Main Properties

Read Only
Two Byte Fixed Length

Description

This parameter returns the change in the DSP’s second encoder input (QEP2) over the last SSC message period. This gives an indication of velocity.