SPI Protocol Document

Specification and protocol information for SPI communication with the Multicount unit.

Revision 5.0

Date September 2004

Trumeter Co Ltd
Web: www.trumeter.com
General: sales@trumeter.com
Support: support@trumeter.com

Contents

CONTENTS **2**

1 INTRODUCTION **3**

2 OPERATIONAL SPECIFICATION **5**

2.1.1 Software Command Format *5*
2.1.2 Commands *7*
2.1.2.1 READ FLAGS - CMD 00 7
2.1.2.2 CLEAR ERRORS - CMD 01 8
2.1.2.3 ADD BYTE TO COUNTER - CMD 02 8
2.1.2.4 ADD WORD TO COUNTER - CMD 03 8
2.1.2.5 READ COUNTER VALUE - CMD 04 9
2.1.2.6 DISPLAY COUNTER VALUE - CMD 05 9
2.1.2.7 DISPLAY COUNTER TEXT - CMD 06 9
2.1.2.8 DISPLAY COUNTER VALUE & TEXT - CMD 07 9
2.1.2.9 SET COUNTER TEXT - CMD 08 10
2.1.2.10 SET SERIAL NUMBER - CMD 09 10
2.1.2.11 READ SERIAL NUMBER - CMD 0A 10
2.1.2.12 LOCK SERIAL NUMBER - CMD 0B 11
2.1.2.13 READ SOFTWARE VERSION - CMD 0C 11
2.1.2.14 DISPLAY SOFTWARE VERSION - CMD 0D 11
2.1.2.15 RESET COUNTER - CMD 0E 11
2.1.2.16 RESET ALL COUNTERS - CMD 0F 12
2.1.2.17 LOCK COUNTER RESET - CMD 10 12
2.1.2.18 STORE USER DATA - CMD 11 12
2.1.2.19 READ USER DATA - CMD 12 12
2.1.2.20 DEFINE USER CHARACTERS - CMD 13 13
2.1.2.21 READ USER CHARACTER - CMD 14 13
2.1.2.22 START CYCLING - CMD 15 13
2.1.2.23 STOP CYCLING - CMD 16 13
2.1.2.24 SHOW MESSAGE - CMD 17 14
2.1.2.25 SHOW LARGE MESSAGE - CMD 18 14
2.1.2.26 GET LAST VALID TX COUNT - CMD 19 14
2.1.2.27 ASSIGN LARGE MESSAGE TEXT - 1A 15
2.1.2.28 DEFINE PUSH BUTTON COUNT - CMD 1B 15
2.1.3 Examples of Communications *16*
2.1.4 Further Information *17*

3 PERFORMANCE SPECIFICATION **19**

4 CONNECTION DETAILS **20**

5 ENVIRONMENTAL SPECIFICATION **20**

1 Introduction

Designed for use in the Gaming & Amusement industry, and general industry, the Multi-count electronic counter is a new concept in counting. The main features of this new counter are:-

1. Two operating modes (Serial and/or Parallel input).
2. 32 independent counters (Serial mode), 6 independent counters (Parallel mode).
3. Can directly replace up to six electromechanical counters in existing installations.
4. E^2PROM memory, giving memory retention without power for more than 100 years with multiple checksums and encoding for added security.
5. Full 8 digit counters with leading zero blanking and annunciation.
6. Alphanumeric display (with optional backlight) makes messaging a real possibility.
7. STN (super twist) LCD display makes the display easy to read from almost any angle.
8. Panel or base mounting industry standard size housing.
9. Sealed to provide security and impossible to mechanically damage without this being obvious even to the untrained eye.
10. Each unit is electronically serial numbered.
11. Operates from 5 to 24V DC.
12. –10°C to 60°C in operation.
13. -20°C to 70°C in storage.
14. CE and UL and cUL approval pending.

The Multi-count has 32 counters stored in a non-volatile memory (E^2PROM), each counter counts from 0 to 99,999,999, and are displayed on the 1 line by 8 character start-burst STN (super twist) alphanumeric LCD display. Each counter can also have up to 8 text characters associated with it, alternately displaying the text and the count value, enabling individual counter readings to be easily found and displayed by the user. Additionally, a separate 64 character scrolling text message can be displayed.

Connecting the counter to either a host machine or to a PC (using the software available from separately) the text associated with each counter can be changed. This enables the gaming or amusement machine owner to change the real name of the counters to be specific to the machine the counter is fitted to. In the case of an update from the host machine (Serial configured counters) then the host machine can automatically download the text.

The counters are resetable, but can be programmed to be non-resetable (this is permanent and cannot be undone).

If a count value over 99,999,999 should occur, then the display would show OVERFLOW, the maximum value that can be recorded is 4,278,190,080.

Serial Input Mode

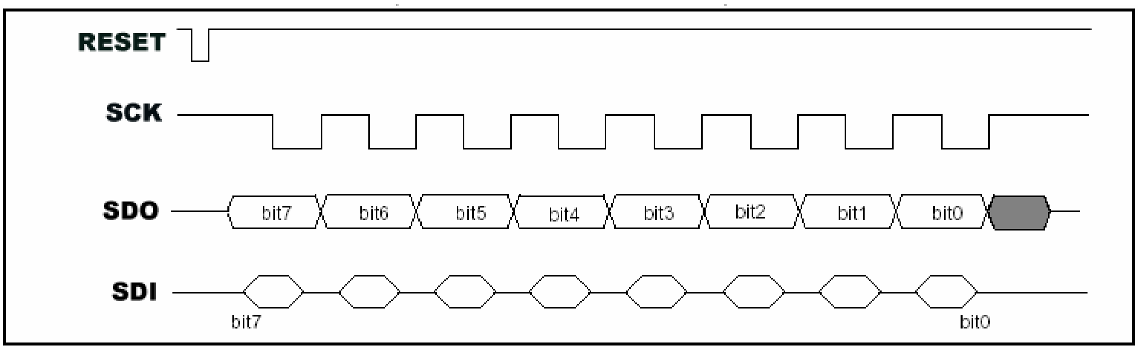
In serial input mode, the host machine can download text to be displayed or download text associated with any counter, and is also used to update any counter. This is done using the Motorola SPI (Serial interface standard) protocol, which is detailed in the instruction manual.

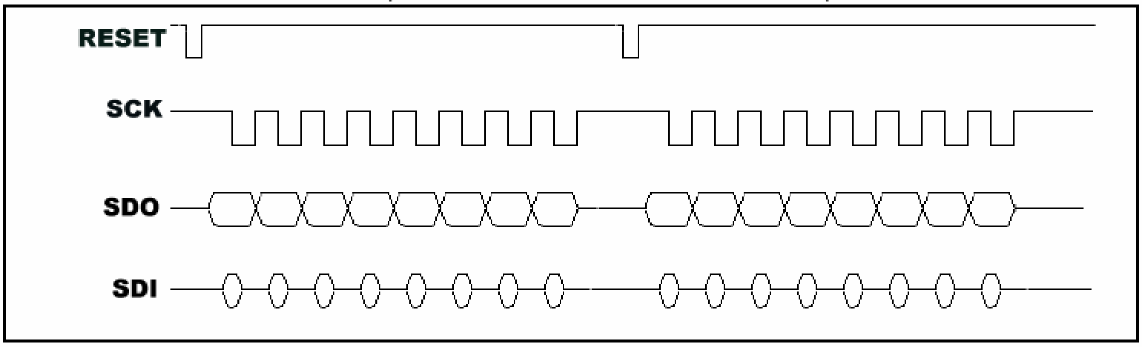
Parallel Input Mode

In parallel input mode, the six counters are incremented using the six independent count inputs, which are on the same connector as the serial input. Connecting the counter to a PC and using the software available separately enables the text associated with the six counters to be changed. In this mode it is also possible to program each counter with a monetary value and status of whether the counter is associated with an input device (for example a coin validater) or an output device (for example a token dispenser). It is also possible to have a totaliser, which shows the total for a selection of counters. The six counters can be viewed in sequence by pressing the front panel button. This button can be programmed to limit viewing to the first N counters where N is 0 to 31. ie, if N=2, only counters 0, 1, and 2 will be displayed. To Increment a counter, connect the respective input to the 0 volts line using either a switch, relay, open collector (sink) output or similar.

Communications Interface

The clocked serial interface conforms to the Motorola SPI (Serial Peripheral Interface) standard. The SPI standard details 4 modes of operation, dependant upon the clock polarity and phase. The Multi-count uses SPI mode 2. In this mode, the serial clock line (SCLOCK) idle state is high (logic 1). To transmit, data is set up on the SIN line, the clock is taken low and then taken high again. The next data bit is set up on the SIN line, the clock taken low and then high and so the transmission continues. (Data is clocked in on the falling edge, MSB is transmitted first). Once the transmission is complete and the command has been executed, the command can be clocked out. First the clock is taken low and then taken high again. The received data is sampled as the clock edge falls. This is repeated for all the data bits. Max clock speed at 50% duty ratio is 5kHz. Slowest clock speed is 100Hz.





Display

The LCD Display is an alphanumeric display using a star-burst segment layout. The display is super twisted nematic in yellow/green mode.

2 Operational Specification

2.1.1 Software Command Format

Commands sent to the counter consist of a variable length data packet, comprising of a header, command, data, and checksum: -

[*] [MC] [CMD] [DATA] [CS]

Response from counter consists of a similar format: -

[&] [MC] [CMD] [DATA] [CS]   or   [NAK]

	Transmit to unit	Receive from unit
[*]	Start of message character	
[&]		Start of message character
[MC]	Message count number, host increments this count for each new message.	Message count number, same as transmitted to counter.
[CMD]	Command for counter specifying required action	Commanded action performed
[DATA]	Optional, any data required for command.	Optional, any data required with response.
[CS]	checksum of all characters in the message	checksum of all characters in the message
[NAK]		Fault

* and & are sent as ASCII, numbers are sent as 2 digit ASCII Hex values
e.g.

Transmit   *000CFD
- *   transmit char
- 00   message counter
- 0C   command (read software version)
- FD   checksum (correct for this example)

Receive   &000C30313030AA
- &   receive char
- 00   transmitted message counter
- 0C   command (read software version)
- 30,31,30,30 received data, in this case ASCII = 0100
- AA   checksum (not correct for this example)

The checksum is the 8 bit sum of each ASCII char sent, not including the checksum itself.
Eg.
Message to transmit = *000C
`‘*’ + ‘0’ + ‘0’ + ‘0’ + ‘C’`
`42  + 48  + 48  + 48  + 67 = 253 (Decimal) = FD (Hex)`
Full Message to transmit with checksum= *000CFD
If the sum results in a value larger the 255 decimal (FF Hex) then only the right two digits of the Hex value are used eg 2F4C becomes 4C.
If the sum is less than 10 Hex then a leading zero is appended eg 9 becomes 09

As the SPI interface is a synchronous full duplex serial interface, for every character sent to the unit, one is received simultaneously. Therefore while the unit is waiting for the start of message character (“*”) and having received it, the unit will transmit Idle (“I”), whilst receiving a message, it will transmit Receiving (“R”), and when the message has been fully received and verified it will transmit Busy (“Z”) until it’s response is ready, at which time it will transmit its data.
So for the above example, below will be the final data (where “x” is ignored)
Transmit `xxxx*000CFDxxxxxxxxxxxxxxxxxxxxxx`
Receive `IIIIIRRRRRZZZZ&000C30313030AAIII`
For every character sent to the unit while it is waiting for a start of message character (“*”), the unit resets the SPI port. Therefore to guarantee a valid start of message keep sending “*” until a “R” is received.

2.1.2 <u>Commands</u>

CMD (Hex)	Name	Description
00	Flags	Returns errors and general status
01	Clear Flags	Clear any re-settable flags
02	Add byte to Counter	Add value between 0 and 255 onto specified counter
03	Add word to Counter	Add value between 0 and 65535 onto counter
04	Read Counter Value	Returns specified counter value
05	Display Counter Value	Show specified counter value on display
06	Display Counter Text	Show specified counter text on display
07	Display Value & Text	Alternate between value and text for specified counter
08	Set Counter Text	Assigns text to be displayed with each counter
09	Set Serial Number	Sends the serial number to the counter
0A	Read Serial Number	Read the serial number of the counter
0B	Lock Serial Number	Makes the serial number read only permanently
0C	Read Software Version	Software version
0D	Display Software Version	Software version
0E	Reset Counter	Clears the contents of the counter
0F	Reset All Counters	Clears the contents of all the counters
10	Lock Counter Reset	Disables counter reset
11	Store User Data	User definable 4 bytes.
12	Read User Data	User definable 4 bytes.
13	Define User Char	Define up to 70 special characters
14	Read User Char	Allows read back of special chars
15	Start Cycling	Show counter text and value for all counters
16	Stop Cycling	Stop cycling and display counter 0 value
17	Show Message	Show message on display
18	Show Large Message	Show a large message (up to 64 characters)
19	Get Last Valid TX count	Returns the transmission counter for the last valid transmission
1A	Assign Large message text	Defines the text to show for the large message
1B	Define Push Button Count	Define how many counters can be accessed by the front panel push button

Please note, in all cases, the MC, CS, D1, D2, etc are sent as two ASCII bytes.
Eg. If the MC byte value is 75, this is 4B in hex, the ASCII for ‘4’ is hex 34, and the ASCII for ‘B’ is hex 42. It is these two bytes (34H and 42H) which are sent. The examples shown below show what would be displayed on a terminal (hyperterminal on a PC)

2.1.2.1 READ FLAGS - CMD 00

Transmit	*	MC	00	CS		
Response	&	MC	00	D1	D2	CS

Where D1 and D2 are the error bytes. Once set this field is never reset until cleared.

Bit D1	Description	Clearable
0	Receiver buffer overflow	Y
1	Receiver checksum corrupt	Y
2	Received command unknown	Y
3	Received data invalid	Y
4	One of counter values repaired	Y
5	Bit set if any counters exceed 99,999,999.	N

6	Set to indicate serial number has been set.	Set No. to 0
7	Busy	N

Bit D2	Description	Clearable
0	EPROM read/write fault	Y
1	EPROM bus failure	Y
2	Single EPROM checksum failure	Y
3	Double EPROM checksum failure	Y

Example
Read status of the flags - shows a Receiver buffer overflow has occurred
Message number is 21
Transmitted *2100ED
Received &21000100AA

2.1.2.2 CLEAR ERRORS - CMD 01

Transmit	*	MC	01	D1	D2	CS
Response	&	MC	01	D1	D2	CS

Where D1 and D2 are the error bytes. Set the required bit in order to clear the corresponding bit in the error bytes.

Example
Clear all flags possible
Message number is 22
Transmitted *2201FFFF07
Received &22010000AB

2.1.2.3 ADD BYTE TO COUNTER - CMD 02

Transmit	*	MC	02	D1	D2	CS
Response	&	MC	02	D1	D2	CS

Where D1 is a value between 0 and 1F (31), specifying the counter to add to.
Where D2 is a value between 0 and FF (255), specifying the value to add to the counter.

Example
Adds the hex value 21 to counter 02
Message number is 23
Transmitted *23020215B9
Received &23020215B5

2.1.2.4 ADD WORD TO COUNTER - CMD 03

Transmit	*	MC	03	D1	D2	D3	CS
Response	&	MC	03	D1	D2	D3	CS

Where D1 is a value between 0 and 1F (31), specifying the counter to add to.
Where D2/D3 is a value between 0 and FFFF (65535), specifying the value to add to the counter.

Example
Adds the hex value 1234 to counter 05
Message number is 24
Transmitted *240305123422
Received &24030512341E

2.1.2.5 READ COUNTER VALUE - CMD 04

Transmit	*	MC	04	D1	CS				
Response	&	MC	04	D1	D2	D3	D4	D5	CS

Where D1 is a value between 0 and 1F (31), specifying the counter to read.
D2, D3, D4, and D5 is the counter value in hex. D2 is the MSB and D5 lower nibble is the LSB.
Max value is FFFFFFFF (4,294,967,295).

Example
Reads the count value of counter 05, returns the hex value 00001234
Transmitted *2504055A
Received &25040500001234E0

2.1.2.6 DISPLAY COUNTER VALUE - CMD 05

Transmit	*	MC	05	D1	CS
Response	&	MC	05	D1	CS

Where D1 is a value between 0 and 1F (31), specifying the counter to display.

Example
Display the contents of counter 05 on the LCD (shows 4660 = 1234 hex)
Transmitted *2605055C
Received &26050558

2.1.2.7 DISPLAY COUNTER TEXT - CMD 06

Transmit	*	MC	06	D1	CS
Response	&	MC	06	D1	CS

Where D1 is a value between 0 and 1F (31), specifying the counter text to display.

Example
Display the text associated with counter 05 on the LCD, (shows " CNTR 05")
Transmitted *2706055E
Received &2706055A

2.1.2.8 DISPLAY COUNTER VALUE & TEXT - CMD 07

Transmit	*	MC	07	D1	CS
Response	&	MC	07	D1	CS

Where D1 is a value between 00 and 1F (31). Specifying the counter value and text to display.

Example
Display the text and count value for counter 05 (shows " CNTR 05" and "   4660" alternately)
Transmitted *28070560
Received &2807055C

2.1.2.9 SET COUNTER TEXT - CMD 08

Transmit	*	MC	08	D1	D2	D3	D4	D5	D6
	D7	D8	D9	CS					
Response	&	MC	08	D1	CS				

Where D1 is a value between 0 and 1F (31), specifying the counter to set text for.
D2 to D9 are ASCII characters, D2 is the left hand char, D9 is the right hand char.

<u>Example</u>
This example assigns the text "TRUMETER" to counter 05, and then sends the command 06 - display counter text - to display the text associated with counter 5 on the LCD (shows "TRUMETER")

Transmitted	*2908055452554D45544552B6
Received	&2908055E
Transmitted	*2A060568
Received	&2A060564

2.1.2.10 SET SERIAL NUMBER - CMD 09

Transmit	*	MC	09	D1	D2	D3	D4	CS
Response	&	MC	09	D1	D2	D3	D4	CS

Where D1, D2, D3, and D4 are 32bits of binary serial number, valid values are 00 to FF (255)

<u>Example</u>
This sets the stored serial number to 12345678 hex, data return shows stored serial number.

Transmitted	*2B0912345678AB
Received	&2B0912345678A7

2.1.2.11 READ SERIAL NUMBER - CMD 0A

Transmit	*	MC	0A	CS				
Response	&	MC	0A	D1	D2	D3	D4	CS

Where D1, D2, D3, and D4 are 32bits of binary serial number, valid values are 00 to FF (255)

<u>Example</u>
This reads back the stored serial number, returns 12345678 hex

Transmitted	*2C0A10
Received	&2C0A12345678B0

2.1.2.12 LOCK SERIAL NUMBER - CMD 0B

Transmit	*	MC	0B	CS
Response	&	MC	0B	CS

Makes the serial number read only, cannot be undone. Set serial number command will respond with the old serial number.

Example

This example locks the stored serial number (12345678), and then tries to set the serial number to 00000000, but as can be seen in the reply, the stored number is still 12345678

Transmitted	*2D0B12
Received	&2D0B0E
Transmitted	*2E09000000008A
Received	&2E0912345678AA

2.1.2.13 READ SOFTWARE VERSION - CMD 0C

Transmit	*	MC	0C	CS				
Response	&	MC	0C	D1	D2	D3	D4	CS

Where D1, D2, D3, and D4 are 4 ASCII chars defining the software version/revision

Example

The software version in this case is 30323031 which is Ver 02 Rev 01

Transmitted	*2F0C15
Received	&2F0C30323031A0

2.1.2.14 DISPLAY SOFTWARE VERSION - CMD 0D

Transmit	*	MC	0D	CS
Response	&	MC	0D	CS

Example

The LCD now shows "`VER 0201`"

Transmitted	*300D01
Received	&300DFD

2.1.2.15 RESET COUNTER - CMD 0E

Transmit	*	MC	0E	D1	CS	
Response	&	MC	0E	D1	D2	CS

Where D1 is a value between 0 and 1F (31), specifying the counter to reset.

Where D2

= 0 if counter has been reset,

= FF if counter cannot be reset.

Example

Counter 05 is reset, as can be confirmed by the response 00, and also by sending the command 04 - read counter value, returns 00000000

Transmitted	*310E0568
Received	&310E0500C4
Transmitted	*32040558
Received	&32040500000000D4

2.1.2.16 RESET ALL COUNTERS - CMD 0F

Transmit	*	MC	0F	CS	
Response	&	MC	0F	D1	CS

Where D1
= 0 if counters have been reset,
= FF if counters cannot be reset.

Example
All counters are reset, as can be confirmed by the response 00
Transmitted *330F06
Received &330F0062

2.1.2.17 LOCK COUNTER RESET - CMD 10

Transmit	*	MC	10	CS
Response	&	MC	10	CS

Makes the counters un-re-settable, cannot be undone.

Example

Transmitted *3410F2
Received &3410EE

2.1.2.18 STORE USER DATA - CMD 11

Transmit	*	MC	11	D1	D2	D3	D4	CS
Response	&	MC	11	D1	D2	D3	D4	CS

Where D1 to D4 can be used to store user data. Not used internally.

Example
Stores the value 18273645, and returns the stored data as verification
Transmitted *35111827364598
Received &35111827364594

2.1.2.19 READ USER DATA - CMD 12

Transmit	*	MC	12	CS				
Response	&	MC	12	D1	D2	D3	D4	CS

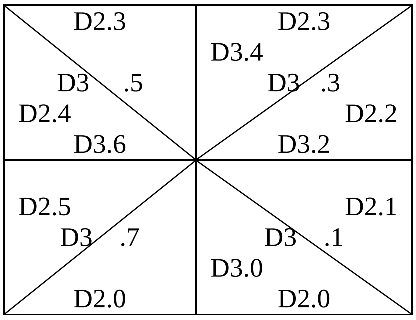
Where D1 to D4 is user data. Not used internally

Example
Reads the stored data, returns the value 18273645
Transmitted *3612F6
Received &36121827364596

2.1.2.20 DEFINE USER CHARACTERS - CMD 13

Transmit	*	MC	13	D1	D2	D3	CS
Response	&	MC	13	D1	D2	D3	CS

D1 is the char value to assign. 71 chars can be defined, 0 to 70 (dec). When using a user character add 128 (80Hex) to the number. Ie. User char number 5 can be displayed by writing ASCII 133 (85Hex)
D2 & D3 are the bit image of the character.



2.1.2.21 READ USER CHARACTER - CMD 14

Transmit	*	MC	14	D1	CS		
Response	&	MC	14	D1	D2	D3	CS

D1 is the char value to assign. 71 chars can be defined, 0 to 70 (dec). When using a user character add 128 (80Hex) to the number. Ie. User char number 5 can be displayed by writing ASCII 133 (85Hex)
D2 & D3 are the bit image of the character.

2.1.2.22 START CYCLING - CMD 15

Transmit	*	MC	15	D1	CS
Response	&	MC	15	D1	CS

D1 is the maximum counter to display. Max 31

<u>Example</u>
Start cycling counters 00 to 02, Display shows repeatedly :- `“ CNTR 00”, “        0”, “ CNTR 01”, “        0”, “ CNTR 02”, “        0”`

Transmitted *4D15026A
Received &4D150266

2.1.2.23 STOP CYCLING - CMD 16

Transmit	*	MC	16	CS
Response	&	MC	16	CS

<u>Example</u>
Display shows `“        0”` which is the count value for counter 00
Transmitted *4E160A
Received &4E1606

2.1.2.24 SHOW MESSAGE - CMD 17

Transmit	*	MC	17	D1	D2	D3	D4	D5	D6
	D7	D8	CS						
Response	&	MC	17	CS					

D1 to D8 are ASCII characters to immediately display,
D1 is the left hand char, D8 is the right hand char.

<u>Example</u>
This displays the word "MULTICNT" on the LCD, it is not stored anywhere.

Transmitted	*4F174D554C5449434E5484
Received	&4F1708

2.1.2.25 SHOW LARGE MESSAGE - CMD 18

Transmit	*	MC	18	D1	CS
Response	&	MC	18	D1	CS

Enable the large message to be displayed. D1 is the number of char to display, min 8, max 64
<u>Example</u>
See command 1A for the example

2.1.2.26 GET LAST VALID TX COUNT - CMD 19

Transmit	*	MC	19	CS	
Response	&	MC	19	D1	CS

Where D1 is the MC number from the last valid message

<u>Example</u>
The last valid command message number sent was 4F

Transmitted	*5019F9
Received	&50194F6F

2.1.2.27 ASSIGN LARGE MESSAGE TEXT - 1A

Transmit	*	MC	1A	D1	D2	D3	D4	D5	D6
	D7	D8	D9	CS					
Response	&	MC	1A	D1	CS				

Where D1 is the block number, and D2 to D9 are the ASCII chars to store in the block. The large message consists of 8 blocks of 8 chars (64 chars max) all displayed sequentially. A message length of 69 characters would be unacceptable which is why it has been broken down into 8 blocks.

<u>Example</u>
The message to display is:-
"INTRODUCING THE TRUMETER MULTICOUNT - WITH 32 COUNTERS - ".

	ASSIGN THE TEXT MESSAGE	
Transmitted	*001A00494E54524F445543C5	"INTRODUC"
Received	&001AF8	
Transmitted	*011A01494E472054484520B1	"ING THE "
Received	&011AF9	
Transmitted	*021A025452554D45544552B4	"TRUMETER"
Received	&021AFA	
Transmitted	*031A03204D554C5449434FD4	" MULTICO"
Received	&031AFB	
Transmitted	*041A04554E54202D205749C3	"UNT - WI"
Received	&041AFC	
Transmitted	*051A055448203332204 34FAB	"TH 32 CO"
Received	&051AFD	
Transmitted	*061A06554E54455253202DC4	"UNTERS -"
Received	&061AFE	
Transmitted	*071A0720202020202020207A	" "
Received	&071AFF	
	DISPLAY THE TEXT MESSAGE	
Transmitted	*08183967	Display 39 hex (57) character
Received	&0818F7	message

2.1.2.28 DEFINE PUSH BUTTON COUNT - CMD 1B

Transmit	*	MC	1B	D1	CS
Response	&	MC	1B	D1	CS

Where D1 is the maximum counter number displayable. If D1 = 2 then the push button would access counters 0,1, and 2. Default is 5 (for 6 electrical count inputs)

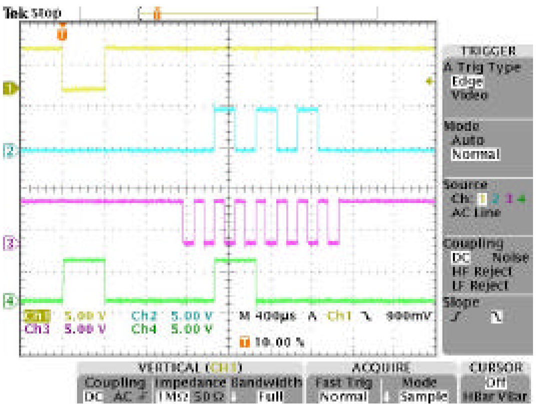
<u>Example</u>
This example limits the counters that can be access by the front button to counters 0, 1, and 2. Counters 3 to 31 are now hidden. These counters can still be displayed but only by using software communications messages.

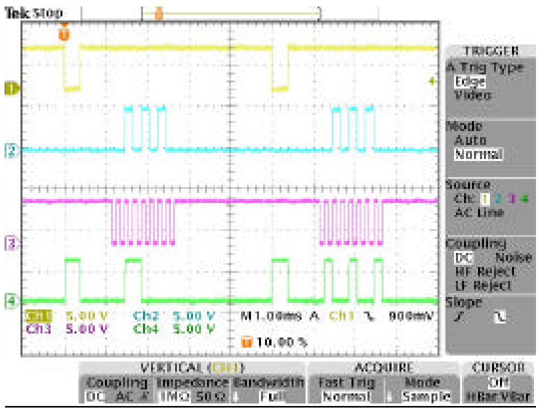
Transmitted	*001B025F
Received	&001B025B

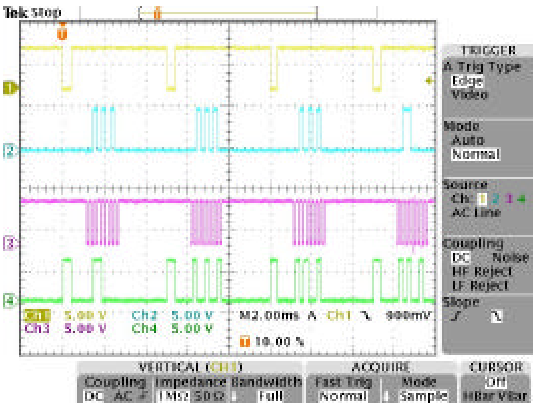
2.1.3 Examples of Communications

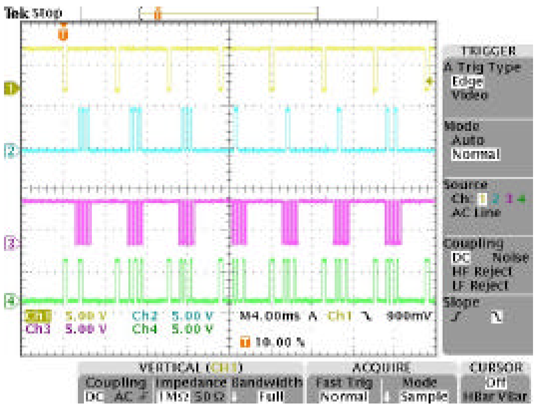
Traces:-

Yellow	Reset Input
Blue	Data Input
Purple	Clock Input
Green	Data Output









2.1.4 Further Information

RESET LINE.

The Reset line is used ONLY for the SPI interface. It has no effect on any counters etc. Its function is to reset the SPI hardware interface prior to a character being received. Theoretically, it should only be needed at the beginning of a message, but it has been found it to be more reliable if it is sent prior to each and every character. The reset input is a active low hardware reset input and needs to only be several uS in width, however there is a transistor buffer input which has a considerable slugging effect on the signal, so a recommended low time of 100uS followed by a high of 100uS minimum is required.

The transition on the Data output line is only due to the SPI output becoming hi impedance when the reset is applied. It is only an effect of the buffer circuitry and has no importance.

MESSAGE COUNTER

The count value that is sent in a transmission to the unit is returned in its reply. The idea for this is that the count value is incremented (by the host machine) prior to each transmission. The host machine can then determine by the returned count value, that the reply was from the last transmission, and not the one before it. The unit doe’s no checking on the value or even use the value, it just returns it in the reply message

SPI CLOCK INPUT

The term "synchronous" refers to the requirement that the data, whether transmitted or received, changes on a clock edge (rising edge in this case) and when received the data is sampled on the opposite clock edge (falling edge in this case). The clock must not be constant (unless data is being transmitted), as each clock pulse will latch in more data into the unit, loss of communications will almost certainly then occur. To transmit a byte of data to the unit, 8 clock pulses are sent, each data bit is sent in turn, being latched in by the respective clock pulse.

SPI PULLUP RESISTORS

All SPI inputs and outputs have, fitted as standard, pull-up resistors to 5V. The value of these resistors is 100K. No external resistors should be required.

If external resistors are to be fitted, the minimum value for the data output should be chosen so that no more than 20mA of current flows.

For the counter digital inputs, this internal value might be too high for a sensor to work properly (eg a standard optical sensor), and so a external pull-up resistor can be fitted between the input and the supply voltage, the value of resistor being chosen to provide enough loading for the sensor to work correctly. If a simple switch is to be used, an external resistor should not be required, unless excessive interference has an effect on the counter.

MOTOROLA SPI MODES

Standard definition:-

SPI MODE	CPOL	CPHA
0	0	0
1	0	1
2	1	0
3	1	1

If the phase of the clock is zero, i.e. CPHA = 0, data is latched on the rising edge of the clock with CPOL=0, and the falling edge of the clock with CPOL=1. If CPHA=1 then the polarities are reversed.

Both the competition and the Multicount use the SPI Mode 2 Standard Protocol.

This defines that the phase of the clock is 0, and the polarity is 1, which results in the data being setup by the data source on the rising edge, and sampled by the receiver on the falling edge of the clock.

NAK

A "NAK" is a possible (fault) response, value is 0x15 (21).
It is sent when the received command is properly formatted, correct checksum etc, but it is trying to access an invalid counter number, or a value is out of range.
There is no need to do anything about it apart from access the right counter!.

FIXED LENGTH MESSAGES

The commands sent to and received from the Multicount do not include a length field however all the messages are of fixed length (specific to each command). If it is required to have a dumb interface microprocessor (eg a RS232 to SPI converter) a solution would be to transmit every character received from the RS232 (in this case) to the SPI port, and then retransmit the byte received from the SPI port back out on the RS232 port.
An alternative is to use a semi dumb microprocessor, and include in the command sent to it an additional extra byte specifying the length of the message to receive. This method then reduces the burden placed on the master controller. The microprocessor can keep sending a start character ('*') until the Multicount replies with a 'R' character (message receive mode). It then sends the rest of the message, the Multicount will respond with a 'R' character for each character sent to it, unless an error occurs and it sends a NAK character. When the last character is sent, the Multicount will reply with a 'Z' character (busy mode). The microprocessor should then keep sending the character '0' to the Multicount until it responds with a '&' (reply start character) and then further '0' till the reply message is complete.
A further method (as used in our SPIIF) is for the microprocessor to decode the command. It will then know the length of every message.

INTER-CHARACTER DELAY

Due to the scanning of the digital inputs, a inter character delay of 1mS (abs min 750uS) is needed, otherwise a overflow situation occurs and the SPI hardware performs a self reset.

INVALID CHARACTERS

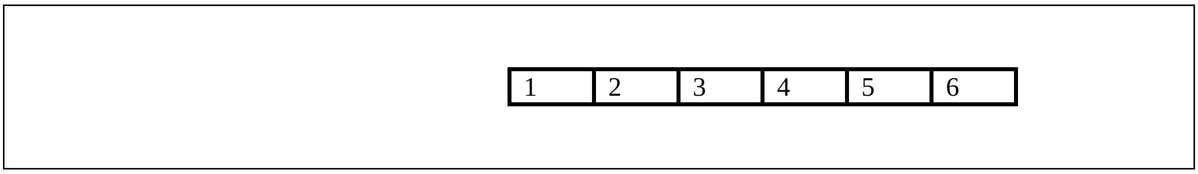
If a message has not been started, the only valid character the Multicount will receive is the start character ('*'). If any other characters are received, the SPI hardware automatically gets reset. This could result in a random character being transmitted from the Multicount when the next character is transmitted to it. Once a message has started the only valid characters are ASCII hexadecimal characters (0 to 9 & A to F). If a '*' is received mid message, the characters received to this point will be ignored and the message reception restarted (even when the Multicount is returning its reply).

3 Performance Specification

Supply Voltage	5v (min) to 26v (max)
Operating Current	3mA
Display	8 Digit 9mm character height STN LCD Black characters
Count Inputs	
Max count rate	200 Hz
Count on	Negative edge.
Max contact bounce	3mS
Default state	Resistive pull-up to 5V
Low level input (Max)	0.8v
High level input (Min)	2v
High level input (Max)	26v
Input source current	50uA
Reset	Re-settable, Can be programmed permanently to be Non-Re-settable.
Serial Communications	
Clock Speed (Max)	5kHz
Clock Speed (Min)	100Hz
Low level input (Max)	0.8v
High level input (Min)	2v
High level input (Max)	26v
Input source current	50uA
Output high voltage	Switch to GND with weak pull-up to 5v, max voltage 18V
Output low voltage	0v(min) – 0.6v(max)
Output current Sink/Source	20mA
Memory	
Type	EEPROM
Write cycles	>1,000,000 Guaranteed
Data retention	>100 years

4 Connection Details

REAR VIEW



Pin	Input/Output	Signal Name	Description
1	Output	SOUT	SPI data output line
2	Input	SCK	SPI clock
3	Input	SIN	SPI data input line
4	Input	/RST	SPI Reset / character synchronisation
5	Input	+12V	DC power supply voltage
6	Input	0V	DC 0V

5 Environmental Specification

Environmental	IP54
Operating temperature range	-10°C to 60°C
Storage temperature range	-20°C to 70°C