Designer Systems .co.uk
Bespoke Electronic Design & Manufacturing in the UK

Fast RS232 serial & I²C Wireless Communications Module

Technical Data

DS-FWCM
Firmware version 3

Features

- **8 times (x8) faster than the standard DS-WCM.**
- **2 times (x2) the distance of the standard DS-WCM**
- **Wirelessly network 255 OOPic (II) controllers / PC's with 100% data accuracy, just think 255 individually controlled robots !!**
- **Not just TX & RX modules, but a complete addressable network.**
- **Form factor identical to OOPic embedded control module (51mm x 89mm high quality PCB)**
- **RS232 or I²C communication interface for simple connection to PC, OOPic, BASIC Stamp etc.**
- **Full bi-directional communication with packet checking, error correction and anti-collision.**
- **Built in 8 line IO port and 4 input 8bit ADC port with remote analysis.**
- **Direct Servo Connections.**
- **Autonomous mode for remote control of eight servos without additional OOPic (II), BASIC Stamp or microcontroller.**
- **433MHz MPT1340 licence exempt communication.**

Description

The Designer Systems DS-FWCM is a highly integrated fast wireless communications module which allows the formation of a bi-directional wireless network capable of transferring data over a distance of up to 300* metres. Specifically targeted at the OOPic (II) & OOPic-R embedded control module and the Personal Computer user the FWCM features I²C and RS232 communication.

Communication between FWCMs' is through self assigned node addresses with a maximum network consisting of 255 FWCM nodes. Data is transferred by simply writing a value (Nine (9) values in expanded data mode) and a destination node to the local FWCM and waiting for confirmation that the data has been sent successfully. The FWCM handles all the error correction, anti-collision and data confirmation protocol necessary for error free transfer.

In addition the DS-FWCM features an on-board fully configurable eight line IO port and a four line analogue input port with automatic measurement. These ports are fully controllable over the wireless network without the intervention of the connected OOPic (II) etc.

For robotic and control applications the remote DS-FWCM can also be configured for autonomous mode. This mode allows eight servos to be controlled remotely without the need for an additional OOPic (II) etc.
Four sensors can also be connected to the ADC inputs to allow the monitoring of distance, light, temperature etc.

Applications

The DS-FWCM has applications in robotics, remote control, data transfer etc. Application notes for the OOPic (II) & OOPic-R controller are provided.

COMMS MODULES

Selection Guide

Description	Part Number
Fast Wireless Communication Module (Bi-directional)	DS-FWCM
OOPic I²C / power 'Y' cable 150mm	DS-WI2CCAB
PC (DB9) to WCM RS232 cable 1.5m	DS-W232CAB

* Distance based on open-air communication and without external interference.

Power requirements

The DS-FWCM may take the power necessary for operation from the OOPic +5V supply or from an external battery or power adaptor supplying between 6 and 16V DC.
The two pin DC power input connector marked '6 – 16V DC' is marked '+' & '-' which should be connected to positive and negative battery/power supply terminals respectively. *Warning: Mis-connection of this connection may damage the DS-FWCM.*

Analogue input port

The DS-FWCM features a four input 8bit Analogue to Digital Converter port 'AN0' to 'AN3' (see Fig. 4.0). Each input is automatically updated every 100mS from an external input voltage of 0 - 5V and the result stored in an internal register which can be read by the connected RS232 or I^{2}C device or over the remote link (see register details further on in this datasheet). *Warning: These inputs are not over-voltage protected and should not be subjected to voltages over 5V.*

Input/Output port

The DS-FWCM also features an eight line logic level IO port 'I/O0' to 'I/O7' (see Fig. 4.0). Each line can be individually configured as input or output with outputs capable of driving a maximum load of 20mA and inputs capable of reading the state of switches, sensors and other input devices. Full control of IO direction, input and output can be accessed by the connected RS232 or I^{2}C device and input / output over the remote link (see register details further on in this datasheet). This port is also used to drive up to eight (8) servos when the DS-FWCM is configured for 'Autonomous Mode' (see below). *Warning: These inputs are not over-voltage protected and should not be subjected to voltages over 5V*

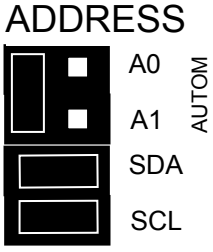
Normal Mode

The DS-FWCM when configured for normal mode allows two or more (255 max.) FWCMs' to form a wireless network capable of transmitting and receiving bytes of data to/from any other FWCM on the network. Each FWCM can allocate its own local node address, which allows peer-to-peer or server based transfer of data as required by the wireless application. Data is transferred a byte at a time (Nine using expanded data mode) with each transmission being checked and confirmed by the remote FWCM node before another can be sent. This protocol ensures that the data sent to the remote FWCM is always 100% correct, thus preventing errors that could cause robot failure or destruction.
During byte transmission or reception the 'STATUS' led is flashed for each packet of data.

Broadcast mode…
If a write to remote node address 254 is undertaken the sending and receiving FWCM(s) automatically enter broadcast mode. This mode allows the sending node, i.e. PC, to broadcast data to multiple remote FWCM(s) simultaneously to allow, for example, the control of multiple rovers. In this mode the remote FWCM(s) will receive the broadcast data even if their local node address is not set to 254 and make it available to the connected device i.e. OOPic etc.
Broadcast mode differs from a standard mode transfer by the fact that the sending FWCM node does not wait for a confirmation from the remote FWCM(s) and the remote FWCM(s) do not send one. If data confirmation is required a timed write from each of the remote FWCM(s) can be undertaken.

Autonomous Mode

A remote DS-FWCM can be configured to act as a complete robotic control solution in conjunction with another local FWCM connected to a PC serial port, OOPic controller, BASIC Stamp etc.
Servo positions are transmitted to the remote FWCM by writing internal registers on the local FWCM via the RS232 or I^{2}C interface. These position values (1-255$_{decimal}$ 0- OFF) are output as servo pulses of 980uS to 2000uS (4uS resolution) on the remote FWCMs' I/O0 to I/O7 lines.
Autonomous mode is configured by connecting the I^{2}C address selection links 'A0' & 'A1' together and setting the SDA/SCL pull-ups on the remote FWCM. This mode is confirmed by the 'STATUS' led continually flashing.



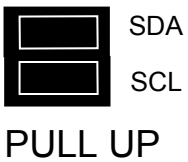
When configured for this mode the node address is automatically set to 255 but may be changed to 244, 245 or 246 by placing additional jumpers on the unused 'I^{2}C Input' and 'I^{2}C Output' connectors (see figure 5.0) as follows:

Address	SDA-GND	SCL-GND
255 (default)	OFF	OFF
246	ON	OFF
245	OFF	ON
244	ON	ON

Each servo output consists of three (3) pins GND, V+ & SVn with most servos complying with the standard colour code of BLACK, RED & YELLOW respectively. Servo power is applied to the connector marked 'SV PWR' (6pin header) see figure 2.0 for connection diagram. Additionally the analogue inputs of the remote autonomous FWCM are passed back to the local FWCM to allow sensors to be monitored.

I^{2}C connector(s)

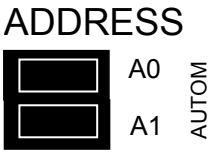
The I^{2}C connector marked 'I^{2}C Input' (4pin header) allows connection to the OOPic controller – using a DS-WI2CCAB cable – or another I^{2}C Master device. There is also an additional connector marked 'I^{2}C Output' (4pin header) that can be used to connect to additional DS-FWCMs' or I^{2}C devices.
The DS-FWCM is fitted with pull-up jumpers that can be configured to provide the source current necessary for I^{2}C communication. The following jumpers should ONLY be set if the I^{2}C bus does not have existing pull-up's:



I^{2}C communication

Up to four FWCM modules may be connected to the same I^{2}C bus and

accessed individually using their own individual address.
The address is configured with the following jumpers:



The following table shows how the jumpers are placed for the different binary addresses:

Address xx	A0	A1
00 (default)	ON	ON
01	OFF	ON
10	ON	OFF
11	OFF	OFF

The binary address (xx) above is used in conjunction with the device ID 11000xxD to form the complete device address i.e. if both jumpers are left connected (default) then the device address would be $1100000D_{binary}$.
The 'D' bit determines if a read or a write to the WCM is to be performed. If the 'D' bit is set '1' then a register read is performed or if clear '0' a register write.

To access individual registers a device write must be undertaken by the OOPic / I^2C Master which consists of a Start condition, device ID ('D' bit cleared), register to start write, one or more bytes of data to be written and a stop condition (see Figure 1.0 for I^2C write protocol).
There are 15 individual registers that can be written within the FWCM that control functions such as node address setup, data transmission control, local & remote IO port setup and control and remote servo positions as follows:

	N_7	N_6	N_5	N_4	N_3	N_2	N_1	N_0
FWCM I2C address								
1.	1	1	0	0	0	X	X	0

XX = Address select pins A1 & A0

Register address

2.	U	U	U	B	B	B	B	B

B..B = 0 to 14 (0 to 26 for read operation)
U..U = unused on this implementation

Local node address

R0	A	A	A	A	A	A	A	A

A..A = 0 to 255 (Address to which this local node is to be allocated)

Remote node address to write too

R1	A	A	A	A	A	A	A	A

A..A = 0 to 255 (Remote node address to which data is to be written)

Remote node timeout value and local / remote reset

R2	A	B	U	U	T	T	T	T

T..T = 1 to 15 (Time in seconds to wait for remote FWCM to respond to write)
A = Local FWCM reset request bit (1 = Reset) *
B = Remote autonomous FWCM reset request bit (1 = Reset) *
U..U = unused on this implementation

Data to write to remote node

R3	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data to be written to remote node)

Local I/O port direction register

R4	X	X	X	X	X	X	X	X

X = 1 or 0 (1 = I/O is input, 0 = I/O is output)

Local I/O port output data register

R5	X	X	X	X	X	X	X	X

X = 1 or 0 (1 = output pin is high, 0= output pin is low)

Remote I/O port output data register

R6	X	X	X	X	X	X	X	X

X = 1 or 0 (1 = output pin is high, 0= output pin is low)

Remote data / servo 1 position

R7	P	P	P	P	P	P	P	P

P..P = 0 to 255 (Extended data value or servo position if remote node is in autonomous mode)

Remote data / servo 2 position

R8	P	P	P	P	P	P	P	P

P..P = 0 to 255 (Extended data value or servo position if remote node is in autonomous mode)

Remote data / servo 3 position

R9	P	P	P	P	P	P	P	P

P..P = 0 to 255 (Extended data value or servo position if remote node is in autonomous mode)

Remote data / servo 4 position

R10	P	P	P	P	P	P	P	P

P..P = 0 to 255 (Extended data value or servo position if remote node is in autonomous mode)

Remote data / servo 5 position

R11	P	P	P	P	P	P	P	P

P..P = 0 to 255 (Extended data value or servo position if remote node is in autonomous mode)

Remote data / servo 6 position

R12	P	P	P	P	P	P	P	P

P..P = 0 to 255 (Extended data value or servo position if remote node is in autonomous mode)

Remote data / servo 7 position

R13	P	P	P	P	P	P	P	P

P..P = 0 to 255 (Extended data value or servo position if remote node is in autonomous mode)

Remote data / servo 8 position

R14	P	P	P	P	P	P	P	P

P..P = 0 to 255 (Extended data value or servo position if remote node is in autonomous mode)

** Warning: A node reset will immediately terminate all communication on the node being reset, use carefully.*

The transfer of data (servo positions or IO) to a remote FWCM node is only activated when a write to Register 3 (Data to write to remote node) is processed. This allows other registers such as the remote node address, servo positions etc. to be stored before the data is transferred.

To read individual status registers a device write then read must be undertaken by the OOPic / I^2C Master. The write consists of a Start condition, device ID ('D' bit cleared), register to start read and a Stop condition.
This is followed by a read, which consists of a Start condition, device ID ('D' bit set), followed by data from the register specified and terminated with a Stop condition. The FWCM also auto increments the register specified for every additional read requested by the Master I^2C device, which allows more than one register to be read in one transaction. This allows for example Register 1 'Write completion flag' and Register 2 'Write time-out error flag' to be read together (see Figure 1.1 for I^2C read protocol).
There are 27 individual registers that can be read within the FWCM as follows:

	N_7	N_6	N_5	N_4	N_3	N_2	N_1	N_0
FWCM Address								
1.	1	1	0	0	0	X	X	1

XX = Address select pins

Remote write completion flag

R0	F	F	F	F	F	F	F	F

F..F = 0 (Write has not completed yet)
F..F = 255 (Write has completed successfully)

Remote write error flag register

R1	F	F	F	F	F	F	F	F

F..F = 0 (Write did not produce an error)
F..F = 255 (Write timeout error, remote WCM did not respond within time specified)

Remote read status register

R2	F	F	F	F	F	F	F	F

F..F = 0 (No data has been received from remote node)
F..F = 255 (Data has been received and is waiting)

Remote read error flag register

R3	F	F	F	F	F	F	F	F

F..F = 0 (No error has occurred)
F..F = 255 (Data overflow condition has occurred) set if data register R5 has not been read and another byte of data has been received from remote WCM.

Remote read node address

R4	A	A	A	A	A	A	A	A

A..A = 0 to 255 (Address from which data has been received)

Remote read data register

R5	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data received from remote node)

Local analogue input AN0 value

R6	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Analogue input value for AN0 input)

Local analogue input AN1 value

R7	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Analogue input value for AN1 input)

Local analogue input AN2 value

R8	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Analogue input value for AN2 input)

Local analogue input AN3 value

R9	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Analogue input value for AN3 input)

Local I/O port input value

R10	X	X	X	X	X	X	X	X

X = 1 or 0 (1 = input pin is high, 0= input pin is low)

Remote status validity flag register

R11	F	F	F	F	F	F	F	F

F..F = 0 (Remote values are invalid)
F..F = 127 (Remote values are valid and remote node is configured for autonomous mode)
F..F = 255 (Remote values are valid and remote node is configured for normal mode)

Remote analogue input AN0 value

R12	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Analogue input value for AN0 input on remote node)

Remote analogue input AN1 value

R13	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Analogue input value for AN1 input on remote node)

Remote analogue input AN2 value

R14	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Analogue input value for AN2 input on remote node)

Remote analogue input AN3 value

R15	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Analogue input value for AN3 input on remote node)

Remote I/O port direction register

R16	X	X	X	X	X	X	X	X

X = 1 or 0 (1 = I/O is input, 0 = I/O is output)

Remote I/O port input value

R17	X	X	X	X	X	X	X	X

X = 1 or 0 (1 = input pin is high, 0= input pin is low)

DS-FWCM Status

R18	U	U	U	U	V	V	V	V

V..V = Firmware version number 1-15

Extended data read validity register

R19	X	X	X	X	X	X	X	X

X..X = 0 to 255 (Value indicates extended data register containing data e.g. = 5 then extended registers 1 and 3 contain data from remote node)

Extended data register 1

R20	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data received from remote node)

Extended data register 2

R21	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data received from remote node)

Extended data register 3

R22	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data received from remote node)

Extended data register 4

R23	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data received from remote node)

Extended data register 5

R24	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data received from remote node)

Extended data register 6

R25	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data received from remote node)

Extended data register 7

R26	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data received from remote node)

Extended data register 8

R27	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data received from remote node)

Registers R12 to R17 contain invalid data until either a data write has been requested and confirmed to/from the remote node or a data read has been received from the remote node. Flags register R11 confirms validity. Registers R20 to R27 contain invalid data until a data read has been received from the remote node. Validity register R19 confirms data that has changed from last reception.

Example.

To set the local FWCM node address to 1 and write a value of 45 to remote FWCM node 9 (Timeout value = 3 seconds, device address = default) write:

Byte 1 (WCM Adr) 11000000_{binary}
Byte 2 (Set register) $0_{decimal}$, 00_{hex}
Byte 3 (Register 0) $1_{decimal}$, 01_{hex}
Byte 4 (Register 1) $9_{decimal}$, 09_{hex}
Byte 5 (Register 2) $3_{decimal}$, 03_{hex}
Byte 6 (Register 3) $45_{decimal}$, $2D_{hex}$

Once this command has been issued the local FWCM can be polled by the OOPic / I^2C Master to determine when the data transfer to the remote node has been successfully completed or a time-out error has occurred. This is accomplished by interrogating Register 1 (Write confirmation flag) followed by a Register 2 (Write time-out flag) as follows:

'Point to register 0
Byte 1 - 11000000_{binary}
Byte 2 – $0_{decimal}$, 00_{hex}

'Read register 0 & 1
Byte 1 - 11000001_{binary}
Byte 2 – $255_{decimal}$ if confirmed, $0_{decimal}$ if not
Byte 3 – $255_{decimal}$ if error, $0_{decimal}$ if not

Example.

To determine if a byte of data has been received from a remote FWCM:

'Point to register 2
Byte 1 - 11000000_{binary}
Byte 2 – $2_{decimal}$, 02_{hex}

'Read register 1 & 2
Byte 1 - 11000001_{binary}
Byte 2 – $255_{decimal}$ if received, $0_{decimal}$ if not
Byte 3 – $255_{decimal}$ if error, $0_{decimal}$ if not

Example.

To set the local FWCM node address to 1 and write nine (9) data values (10,20,30,40,50,60, 70,80,90) to remote FWCM node 9 (Timeout value = 3 seconds, device address = default) write:

Byte 1 (WCM Adr) 11000000_{binary}
Byte 2 (Set register) $7_{decimal}$, 07_{hex}
Byte 3 (Register 7) $20_{decimal}$
Byte 4 (Register 8) $30_{decimal}$
Byte 5 (Register 9) $40_{decimal}$
Byte 6 (Register 10) $50_{decimal}$
Byte 7 (Register 11) $60_{decimal}$
Byte 8 (Register 12) $70_{decimal}$
Byte 9 (Register 13) $80_{decimal}$
Byte 10 (Register 14) $90_{decimal}$

Then write:

Byte 1 (WCM Adr) 11000000_{binary}
Byte 2 (Set register) $0_{decimal}$, 00_{hex}
Byte 3 (Register 0) $1_{decimal}$, 01_{hex}
Byte 4 (Register 1) $9_{decimal}$, 09_{hex}
Byte 5 (Register 2) $3_{decimal}$, 03_{hex}
Byte 6 (Register 3) $10_{decimal}$

The first write sequence loads the eight values into the extended registers the second sets the local / remote node address, time-out and activates transmission to the remote node.

Example.

To write eight (8) servo positions to a remote FWCM configured for autonomous mode (node 255) set the local FWCM node address to 1 and activate transmission (Timeout value = 3 seconds, device address = default) write:

Byte 1 (WCM Adr) 11000000_{binary}
Byte 2 (Set register) $7_{decimal}$, 07_{hex}
Byte 3 (Register 7) $32_{decimal}$, 20_{hex}
Byte 4 (Register 8) $64_{decimal}$, 40_{hex}
Byte 5 (Register 9) $96_{decimal}$, 60_{hex}
Byte 6 (Register 10) $128_{decimal}$, 80_{hex}
Byte 7 (Register 11) $160_{decimal}$, $A0_{hex}$
Byte 8 (Register 12) $192_{decimal}$, $C0_{hex}$
Byte 9 (Register 13) $224_{decimal}$, $E0_{hex}$
Byte 10 (Register 14) $255_{decimal}$, FF_{hex}

Then write:

Byte 1 (WCM Adr) 11000000_{binary}
Byte 2 (Set register) $0_{decimal}$, 00_{hex}
Byte 3 (Register 0) $1_{decimal}$, 01_{hex}
Byte 4 (Register 1) $255_{decimal}$, FF_{hex}
Byte 5 (Register 2) $3_{decimal}$, 03_{hex}
Byte 6 (Register 3) $0_{decimal}$, 0_{hex}

The first write sequence sets up the servo positions the second sets the remote node address (255 for autonomous FWCM), time-out and activates transmission by writing to the data register.

RS232 connection & setup

The RS232 connector marked 'RS232 COMMS' (6pin header) allows connection to a Personal Computer – using a DS-W232CAB cable – or serial RS232 device. Connection is via. a six (6) pin vertical header, pinned as follows:

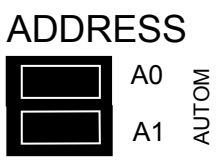
Header connection	Description
1	Serial output
2	Serial input
3	Ground (Servo power -)
4	Serial output
5	Serial input
6	Servo power +

The connection is formatted to allow the maximum of four (4) DS-FWCM modules to be connected in a 'daisy chain' configuration similar to an I^2C bus.
This is accomplished by connecting all 'Serial inputs' together, all 'Serial outputs' together using diode steering (see Figure 3.0) and changing the FWCM 'ADDRESS' links on each module.
The RS232 connection will support any modern RS232D/E compliant device and must be configured for the following protocol:

19200 baud (bps)
8 Data bits
1 Stop Bit
No Parity
No handshaking (if configurable)

RS232 communication

Up to four FWCM modules may be connected to the same RS232 device and accessed individually using their own individual address. The address is configured with the following jumpers:



The following table shows how the jumpers are placed for the different binary addresses:

Address xx	A0	A1
00 (default)	ON	ON
01	OFF	ON
10	ON	OFF
11	OFF	OFF

The binary address (xx) above is used in conjunction with the device ID 11000xxD to form the complete device address i.e. if both jumpers are left connected (default) then the device address would be $1100000D_{binary}$.
The 'D' bit determines if a read or a write to the FWCM is to be performed. If the 'D' bit is set '1' then a register read is performed or if clear '0' a register write.

To access individual registers a device write must be undertaken by the RS232 device which consists of a Prefix character, device ID ('D' bit cleared), register to start write, one or more bytes of data to be written and two terminator characters.

There are 15 individual registers that can be written within the FWCM that control functions such as node address setup, data transmission control, local & remote IO port setup and control and remote servo positions/data values as follows:

	N_7	N_6	N_5	N_4	N_3	N_2	N_1	N_0

RS232 command prefix

1.	0	1	0	1	1	0	1	1

ASCII character ' ['

FWCM RS232 address

2.	1	1	0	0	0	X	X	0

XX = Address select pins A1 & A0

Register address

3.	U	U	U	U	B	B	B	B

B..B = 0 to 15
U..U = unused on this implementation

Local node address

R0	A	A	A	A	A	A	A	A

A..A = 0 to 255 (Address to which this local node is to be allocated)

Remote node address to write too

R1	A	A	A	A	A	A	A	A

A..A = 0 to 255 (Remote node address to which data is to be written)

Remote node timeout value and local / remote reset

R2	A	B	U	U	T	T	T	T

T..T = 1 to 15 (Time in seconds to wait for remote FWCM to respond to write)
A = Local FWCM reset request bit (1 = Reset) *
B = Remote autonomous FWCM reset request bit (1 = Reset) *
U..U = unused on this implementation

Data to write to remote node

R3	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data to be written to remote node)

Local I/O port direction register

R4	X	X	X	X	X	X	X	X

X = 1 or 0 (1 = I/O is input, 0 = I/O is output)

Local I/O port output data register

R5	X	X	X	X	X	X	X	X

X = 1 or 0 (1 = output pin is high, 0= output pin is low)

Remote I/O port output data register

R6	X	X	X	X	X	X	X	X

X = 1 or 0 (1 = output pin is high, 0= output pin is low)

Remote data / servo 1 position

R7	P	P	P	P	P	P	P	P

P..P = 0 to 255 (Extended data value or servo position if remote node is in autonomous mode)

Remote data / servo 2 position

R8	P	P	P	P	P	P	P	P

P..P = 0 to 255 (Extended data value or servo position if remote node is in autonomous mode)

Remote data / servo 3 position

R9	P	P	P	P	P	P	P	P

P..P = 0 to 255 (Extended data value or servo position if remote node is in autonomous mode)

Remote data / servo 4 position

R10	P	P	P	P	P	P	P	P

P..P = 0 to 255 (Extended data value or servo position if remote node is in autonomous mode)

Remote data / servo 5 position

R11	P	P	P	P	P	P	P	P

P..P = 0 to 255 (Extended data value or servo position if remote node is in autonomous mode)

Remote data / servo 6 position

R12	P	P	P	P	P	P	P	P

P..P = 0 to 255 (Extended data value or servo position if remote node is in autonomous mode)

Remote data / servo 7 position

R13	P	P	P	P	P	P	P	P

P..P = 0 to 255 (Extended data value or servo position if remote node is in autonomous mode)

Remote data / servo 8 position

R14	P	P	P	P	P	P	P	P

P..P = 0 to 255 (Extended data value or servo position if remote node is in autonomous mode)

RS232 command terminator

n.	0	1	0	1	1	1	0	1

ASCII character '] '

RS232 command terminator

n.	0	1	0	1	1	1	0	1

ASCII character '] '

The RS232 protocol differs from the I^2C in that it is able to respond asynchronously (when ever it likes) to certain conditions such as confirming when data has been transferred or when data has been received. This is accomplished by the FWCM transmitting response packets to the RS232 device in the following formats:

Write confirmation…
(sent upon write confirmation or time-out):

	N_7	N_6	N_5	N_4	N_3	N_2	N_1	N_0

RS232 command prefix

1.	0	1	0	1	1	0	1	1

ASCII character ' ['

FWCM RS232 address

2.	1	1	0	0	0	X	X	0

XX = Address select pins A1 & A0

FWCM Status flag

3.	F	F	F	F	F	F	F	F

F..F = 0 (Write confirmation / time-out identifier)

Remote write error flag register

4.	F	F	F	F	F	F	F	F

F..F = 0 (Write did not produce an error)
F..F = 255 (Write timeout error, remote WCM did not respond within time specified)

RS232 command terminator

5.	0	1	0	1	1	1	0	1

ASCII character '] '

Read confirmation…
(Sent upon receipt of data from remote FWCM node):

	N_7	N_6	N_5	N_4	N_3	N_2	N_1	N_0

RS232 command prefix

1.	0	1	0	1	1	0	1	1

ASCII character ' ['

FWCM RS232 address

2.	1	1	0	0	0	X	X	0

XX = Address select pins A1 & A0

FWCM Status flag

3.	F	F	F	F	F	F	F	F

F..F = 255 (Data received from remote node identifier)

Remote read node address

4.	A	A	A	A	A	A	A	A

A..A = 0 to 255 (Address from which data has been received)

Remote read data register

5.	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data received from remote node)

Extended data read validity register

6.	X	X	X	X	X	X	X	X

X..X = 0 to 255 (Value indicates extended data register containing data e.g. = 5 then extended registers 1 and 3 contain data from remote node)

Extended data register 1

7.	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data received from remote node)

Extended data register 2

8.	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data received from remote node)

Extended data register 3

9.	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data received from remote node)

Extended data register 4

10.	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data received from remote node)

Extended data register 5

11.	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data received from remote node)

Extended data register 6

12.	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data received from remote node)

Extended data register 7

13.	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data received from remote node)

Extended data register 8

14.	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data received from remote node)

RS232 command terminator

15.	0	1	0	1	1	1	0	1

ASCII character '] '

Values 7 to 14 contain data with validity confirmed by value 6.

Example:
To set the local FWCM node address to 1 and write a value of 45 to remote WCM node 9 (Timeout value = 3 seconds, device address = default) write:

[#192 #0 #1 #9 #3 #45]]

Where #nn is a value

On confirmation that the value has been sent successfully or that a time-out error has occurred the FWCM will reply with:

Sent OK [#192 #0 #0]

Timeout error [#192 #0 #255]

Example:
To set the local FWCM IO port lines 0 – 3 to inputs, 4 – 7 to outputs and set lines 4 & 6 high, 5 & 7 low:

[#192 #4 #15 #80]]

Where #nn is a value

To read the internal status registers a device read command must be received from the RS232 device i.e. PC etc.

The read request consists of a Prefix character, device ID (‘D’ bit set) and two terminator characters as follows:

	N_7	N_6	N_5	N_4	N_3	N_2	N_1	N_0
RS232 command prefix								
1.	0	1	0	1	1	0	1	1
ASCII character ‘ [’								
FWCM RS232 address								
2.	1	1	0	0	0	X	X	1
Read request. XX = Address select pins A1 & A0								
RS232 command terminator								
3.	0	1	0	1	1	1	0	1
ASCII character ‘] ’								
RS232 command terminator								
4.	0	1	0	1	1	1	0	1
ASCII character ‘] ’								

Once this command has been received, the relevant FWCM responds by transmitting the contents of all the 28 registers in the following format:

	N_7	N_6	N_5	N_4	N_3	N_2	N_1	N_0
RS232 command prefix								
1.	0	1	0	1	1	0	1	1
ASCII character ‘ [’								
FWCM RS232 address								
2.	1	1	0	0	0	X	X	0
XX = Address select pins A1 & A0								

Remote write completion flag

R0	F	F	F	F	F	F	F	F

F..F = 0 (Write has not completed yet)
F..F = 255 (Write has completed successfully)

Remote write error flag register

R1	F	F	F	F	F	F	F	F

F..F = 0 (Write did not produce an error)
F..F = 255 (Write timeout error, remote WCM did not respond within time specified)

Remote read status register

R2	F	F	F	F	F	F	F	F

F..F = 0 (No data has been received from remote node)
F..F = 255 (Data has been received and is waiting)

Remote read error flag register

R3	F	F	F	F	F	F	F	F

F..F = 0 (No error has occurred)
F..F = 255 (Data overflow condition has occurred) set if data register R5 has not been read and another byte of data has been received from remote WCM.

Remote read node address

R4	A	A	A	A	A	A	A	A

A..A = 0 to 255 (Address from which data has been received)

Remote read data register

R5	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data received from remote node)

Local analogue input AN0 value

R6	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Analogue input value for AN0 input)

Local analogue input AN1 value

R7	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Analogue input value for AN1 input)

Local analogue input AN2 value

R8	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Analogue input value for AN2 input)

Local analogue input AN3 value

R9	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Analogue input value for AN3 input)

Local I/O port input value

R10	X	X	X	X	X	X	X	X

X = 1 or 0 (1 = input pin is high, 0= input pin is low)

Remote status validity flag register

R11	F	F	F	F	F	F	F	F

F..F = 0 (Remote values are invalid)
F..F = 127 (Remote values are valid and remote node is configured for autonomous mode)
F..F = 255 (Remote values are valid and remote node is configured for normal mode)

Remote analogue input AN0 value

R12	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Analogue input value for AN0 input on remote node)

Remote analogue input AN1 value

R13	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Analogue input value for AN1 input on remote node)

Remote analogue input AN2 value

R14	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Analogue input value for AN2 input on remote node)

Remote analogue input AN3 value

R15	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Analogue input value for AN3 input on remote node)

Remote I/O port direction register

R16	X	X	X	X	X	X	X	X

X = 1 or 0 (1 = I/O is input, 0 = I/O is output)

Remote I/O port input value

R17	X	X	X	X	X	X	X	X

X = 1 or 0 (1 = input pin is high, 0= input pin is low)

DS-FWCM Status

R18	U	U	U	U	V	V	V	V

V..V = Firmware version number 1-15

Extended data read validity register

R19	X	X	X	X	X	X	X	X

X..X = 0 to 255 (Value indicates extended data register containing data e.g. = 5 then extended registers 1 and 3 contain data from remote node)

Extended data register 1

R20	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data received from remote node)

Extended data register 2

R21	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data received from remote node)

Extended data register 3

R22	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data received from remote node)

Extended data register 4

R23	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data received from remote node)

Extended data register 5

R24	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data received from remote node)

Extended data register 6

R25	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data received from remote node)

Extended data register 7

R26	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data received from remote node)

Extended data register 8

R27	D	D	D	D	D	D	D	D

D..D = 0 to 255 (Data received from remote node)

RS232 command terminator

n.	0	1	0	1	1	1	0	1

ASCII character ‘] ’

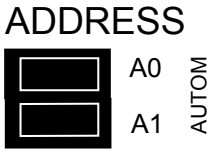
Registers R12 to R17 contain invalid data until either a data write has been requested and confirmed to/from the remote node or a data read has been received from the remote node. Flags register R11 confirms validity. Registers R20 to R27 contain invalid data until a data read has been received from the remote node. Validity register R19 confirms data that has changed from last reception.

SCP communication

The Savage Innovations SCP allows a remote PC, Pocket PC, Palm Pilot, or any other device with a serial port to control and read the FWCM. The serial protocol is as follows:

19200 baud (bps), 8 Data bits
1 Stop Bit, No Parity
No handshaking (if configurable)

Up to four DS-FWCM units may be connected to the same RS232 port and accessed individually with their own node address. The node address is configured with the following jumpers:



The following table shows how the jumpers are placed for the different node addresses:

Node Address	A0	A1
‘0’ (default)	ON	ON
‘1’	OFF	ON
‘2’	ON	OFF
‘3’	OFF	OFF

The entire character set used by SCP is composed of human readable characters so that a serial terminal program can be used to manually control FWCM data transfer. The following commands are only briefly described as the full SCP is not within the scope of this data sheet, a full explanation being available from Savage Innovations.

To enable SCP:

Send “**0**V**”*	replace 0 with node address 0 - 3
Receive “v”	indicates FWCM is functional.

To set Memory type:

Send “128H”	128 + number of registers to access 0 = 1, 1=2
Receive “h”	confirms set.

To set register location to start access:

Send “15J”	15 is register number 0 to 31.
Receive “j”	confirms set.

To write to previously selected register:

Send “80N”	80 is sample hexadecimal value to write to register (must be in two character notation)
Receive “n”	confirms write.

To read previously selected register:

Send “M”	Request register read.
Receive “80m”	80 is sample hexadecimal value.

To read register location:

Send “I”	Request register location.
Receive “16i”	16 is current register location.

To read memory type:
Send "G" Request memory type.

Receive "128g" 128 is current memory type.

To reset FWCM:
Send "W" FWCM is reset.

No response generated.

To query SCP buffer:
Send "Q" Request SCP buffer contents.
Receive "124q" Characters in buffer are returned e.g. 124 followed by "q". This command does not effect the buffers contents.

To disable SCP:
Send "X" Request exit from SCP.

Receive "x" SCP has exited.

If command format or value is not correct then a "!" response will be received and the command will not be executed.

Use the I^2C register set for SCP.

Example:
To enable SCP, setup memory type register, location register, set the local FWCM node address to 1 and write a value of 45 to remote FWCM node 9 (Timeout value = 3 seconds, device address = default) write:

Send "\0V"	Enable SCP.
Receive "v"	SCP enabled.
Send "131H"	Set memory type to 128 + 3=131 (4 registers to be accessed).
Receive "h"	Command confirmed.
Send "0J"	Set register location to 0 to allow access to R0 to R3.
Receive "j"	Command confirmed.
Send "0109032dN"	Write data to required registers.
Receive "n"	Write confirmed.

Licensing conformance

UK approval to MPT1340:

This module carries an inspection mark located on the PCB:

MPT1340
Licence exempt

Should this module be used for UK commercial application the Radio communications Agency (RA) must be notified on form RA 286 (Cat 4) obtainable from RA's library service, Tel: 0171 211 0502/ 0505.

European approval ETS 300-220-3:

This module is also compliant with European standard ETS 300-200-3 but does not bear an inspection mark. Should full compliancy be required then the following mark should be applied:

ETS 300-220-3
Licence exempt

USA FCC approval:

This module is not FCC approved. It is designed to comply with the FCC Part 15 rules & regulations for a non-licensed low power device. This product when used in the United States of America is strictly intended for experimental use and should you wish to use this module within an actual product (i.e. non-experimental use) then this module must firstly be designed into the product, then the whole product must be approved by the FCC. For more information please go to the FCC web site at www.fcc.gov.

Antenna considerations

The flexible whip antenna on the DS-FWCM is designed to provide the best possible signal under normal conditions. The antenna will be significantly 'de-tuned' if placed close to a large metal object or a device that generates large amounts of interference such as a PC. To ensure the best possible communication the FWCM module should be sited away from metal objects and interference sources. The OOPic controller does not effect the operation of the FWCM and can be sited as close as required.

Firmware revision history:

V1 Release version of the firmware.
V2 Corrected bug in I2C register access.
V3 Corrected bug in servo drive routine, added multiple autonomous node address setup, added local and remote (autonomous only) reset to power-on state bits to time-out register.

Electrical Characteristics (T_A = 25°C Typical)

Parameter	Minimum	Maximum	Units	Notes
Supply Voltage (+5V) (Vcc)	4.5	5.5	V	1
Supply Voltage (6-16V)	6	16	V	1
Supply Current	40	50	mA	2
RS232 TX data output level	0.3	Vcc-0.8V	V	
RS232 RX data input level	-15	+15	V	
RS232 speed	-	19200	bps	
I^2C speed	-	400	kHz	
I^2C pull-up resistance	-	4700	Ω	3
Wireless frequency	-	433.9	MHz	
Distance	-	300	Metres	4
Retry speed	-	100	mS	
ADC input voltage	0	Vcc	V	
ADC measurement cycle	-	100	mS	
IO line output voltage	0.3	Vcc-0.8V	V	
IO line output current	-	20	mA	
IO line input voltage	0	Vcc+0.3V	V	
Servo output pulse duration	980	2000	uS	5
Servo output pulse repetition	-	20	mS	5
Servo output pulse resolution	-	4	uS	5

Absolute Maximum Ratings

Parameter	Minimum	Maximum	Units
Supply Voltage (+5V)	-0.5	+6	V
Supply Voltage (6-16V)	-0.5	+20	V

Environmental

Parameter	Minimum	Maximum	Units
Operating Temperature	0	70	°C
Storage Temperature	-10	80	°C
Humidity	0	80	%
Dimensions	Length 89mm, Width 51mm, Height 145mm		
Weight	20g		
Immunity & emissions	EMC compliance to 89/336/EEC		

Notes:
1. Vcc is supply rail from OOPic or any other +5V supply.
2. Maximum value given is during byte transmission.
3. Value given is to Vcc when activated with appropriate jumpers.
4. Based on open-air communication and no interference.
5. Autonomous mode only.

WEEE Consumer Notice

This product is subject to Directive 2002/96/EC of the European Parliament and the Council of the European Union on Waste of Electrical and Electronic Equipment (WEEE) and, in jurisdictions adopting that Directive, is marked as being put on the market after August 13, 2005, and should not be disposed of as unsorted municipal/public waste. Please utilise your local WEEE collection facilities in the disposition and otherwise observe all applicable requirements. For further information on the requirements regarding the disposition of this product in other languages please visit www.designersystems.co.uk

RoHS Compliance

This product complies with Directive 2002/95/EC of the European Parliament and the Council of the European Union on the Restriction of Hazardous Substances (RoHS) which prohibits the use of various heavy metals (lead, mercury, cadmium, and hexavalent chromium), polybrominated biphenyls (PBB) and polybrominated diphenyl ethers (PBDE).

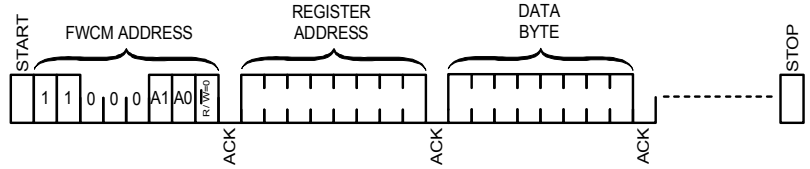
Example files for OOPic communication:

The following OOPic BASIC sub-routines can be used to communicate between OOPic controllers fitted with DS-FWCM modules. Please see the DS-FWCM demonstration diskette for code samples that include OOPic controller loop testing and servo positioning when configured for Autonomous mode.

```
' DS-FWCM application sub-routines
'
' Subroutines:
'
' SetUpFWCM - Sets up I2C communication to FWCM and configures local node address.
' WriteFWCM - Takes the value in 'WriteVal' and sends to the remote FWCM node 'RemNode' waiting 'Tout' seconds for transfer to
' complete. Sets flag 'Error' if data was unable to be sent and FWCM had tried for 'Tout' seconds.
' ReadFWCM - Checks for data from remote FWCM node, if received sets flag 'ReadOK', stores remote node FWCM node address in 'ReadAdr'
' and data in 'ReadVal'. Flag 'Oerror' is set if overflow error occurred
'
' Created : 09/07/02                    Revision : 1.00              Written by : Designer Systems
'
Dim FWCM As New oi2c
Dim WriteVal As New oByte                          'Value to write to remote node
Dim LocNode As New oByte                           'Local node address
Dim RemNode As New oByte                           'Remote node address to write to
Dim ReadVal As New oByte                           'Value read from remote node
Dim ReadAdr As New oByte                           'Node address of remote node value came from
Dim RemADC0 As New oByte                           'Remote node ADC value for channel 0
Dim RemADC1 As New oByte                           'Remote node ADC value for channel 1
Dim RemADC2 As New oByte                           'Remote node ADC value for channel 2
Dim RemADC3 As New oByte                           'Remote node ADC value for channel 3
Dim RemSrv1 As New oByte                           'Remote servo 1 position storage register
Dim RemSrv2 As New oByte                           'Remote servo 2 position storage register
Dim RemSrv3 As New oByte                           'Remote servo 3 position storage register
Dim RemSrv4 As New oByte                           'Remote servo 4 position storage register
Dim RemSrv5 As New oByte                           'Remote servo 5 position storage register
Dim RemSrv6 As New oByte                           'Remote servo 6 position storage register
Dim RemSrv7 As New oByte                           'Remote servo 7 position storage register
Dim RemSrv8 As New oByte                           'Remote servo 8 position storage register
Dim LocADC0 As New oByte                           'Local ADC value for channel 0
Dim LocADC1 As New oByte                           'Local ADC value for channel 1
Dim LocADC2 As New oByte                           'Local ADC value for channel 2
Dim LocADC3 As New oByte                           'Local ADC value for channel 3
Dim LocIO As New oByte                             'Local IO port input value
Dim Error As New oBit
Dim OError As New oBit
Dim ReadOK As New oBit
Dim Done As New oBit
Dim TOut As New oByte


Sub Main()

          LocNode = 1                              'Define local node address (0-255)
          RemNode = 2                              'Define remote node address (0-255)
          Const WriteTOut = 6                      'Define write timeout in seconds (0-15)
          Const FWCMAdr = &h60                     'FWCM A0 & A1 jumpers ON (Range &h60-&h63)
          Call SetUpFWCM                           'Setup FWCM ready for communication

          .
          . Main code goes in here
          .

End Sub
'
' Subroutine to setup I2C communication to DS-FWCM and
' set Local node address
'
Sub SetUpFWCM()

          'Set the DS-FWCM I2C address shifted right by 1 bit

          FWCM.Node = FWCMAdr                      'Setup I2C address for FWCM

          'Setup I2C addressing to FWCM

          FWCM.Width = cv8bit                      'Control Info is 1-byte
          FWCM.Mode = cv10bit                      'I2C mode is 10-Bit Addressing
          FWCM.NoInc = cvFalse                     'Increment on every read/write
          FWCM.Location = 0                        'Specify local node address location
          FWCM.Value = LocNode                     'and write local node address

End Sub
'
' Subroutine to write data to remote FWCM (RemNode) address and upon
' confirmation that data has been written store remote node
' ADC values in RemADC0 - RemADC3
' On exit 'Error' = False if write completed OK
'
Sub WriteFWCM()

          FWCM.Location = 1                        'Start at remote node address
          FWCM.Value = RemNode                     'Set remote node address to write to
          FWCM.Value = WriteTOut                   'Set write timeout value 1-15 seconds
          FWCM.Value = WriteVal                    'Send data to remote node

          call      WaitFWCM                       'Wait for confirmation or error

End Sub
```

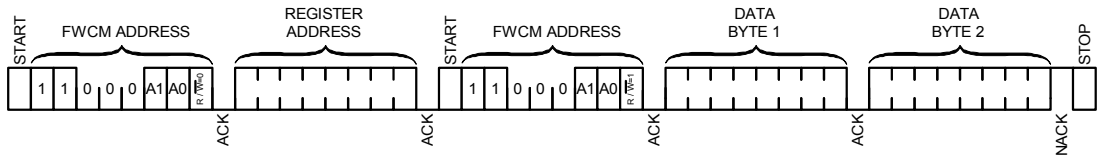
```
'
' Subroutine to check for data from remote FWCM(s) and upon
' confirmation that data has arrived store data in 'ReadVal' and
' remote FWCM address in 'ReadAdr'.
' On exit 'ReadOK' = True if data was read and 'OError' = True
' if internal data register has overflowed i.e. one or more data
' bytes may have been lost
'
Sub ReadFWCM()

          FWCM.Location = 2                     'Start at read status flag (R2)

                    If FWCM.Value = 255 then    'Has data been received (R2) ?

                    If FWCM.Value = 0 then      'If so has overflow error occurred (R3) ?
                              ReadAdr = FWCM.Value'If not then store FWCM address data came from
                              ReadVal = FWCM.Value'and store data
                              OError = cvFalse  'Indicate no overflow error
                              ReadOK = cvTrue   'Indicate data has been read and stored
                    else
                              OError = cvTrue   'Indicate overflow error has occurred
                              ReadOK = cvTrue   'Indicate data has been read
                              ReadAdr = FWCM.Value'Store FWCM address data came from
                              ReadVal = FWCM.Value'and store data
                    end if

                    else

                    OError = cvFalse            'If no data read then no error
                    ReadOK = cvFalse            'and no data

                    end if

          FWCM.Location = 6                     'Start at local analogue inputs (R6)

          LocADC0 = FWCM.Value                  'Store ADC 0 value
          LocADC1 = FWCM.Value                  'Store ADC 1 value
          LocADC2 = FWCM.Value                  'Store ADC 2 value
          LocADC3 = FWCM.Value                  'Store ADC 3 value
          LocIO = FWCM.Value                    'Store IO port input value

End Sub
'
' Subroutine to wait for confirmation that data has been successfully sent to
' the remote node or that a timeout error has occurred
'
Sub WaitFWCM()

          Done = cvFalse
do
          FWCM.Location = 0                     'Start at write completion status flag (R0)

                    If FWCM.Value = 255 then    'Has write been completed (R0) ?

                    If FWCM.Value = 0 then      'Has write resulted in error (R1) ?

                    FWCM.Location = 6           'If not then setup remote node AD value register
                    RemADC0 = FWCM.Value        'Store remote ADC value 0
                    RemADC1 = FWCM.Value        'Store remote ADC value 1
                    RemADC2 = FWCM.Value        'Store remote ADC value 2
                    RemADC3 = FWCM.Value        'Store remote ADC value 3
                    Error = cvFalse             'Indicate no error has resulted
                    Done = cvTrue               'and request exit from sub-routine

                    else

                    Error = cvTrue              'If so then indicate error
                    Done = cvTrue               'and request exit from sub-routine

                    end if

                    else

                    If FWCM.Value = 255 then    'Has write resulted in error (R1) ?
                    Error = cvTrue              'If so then indicate error
                    Done = cvTrue               'and request exit from sub-routine
                    end if

                    end if

loop until Done = cvTrue

End Sub
```

Figure 1.0 (I^2C write protocol)



Multiple bytes may be written before the 'STOP' condition. Data is written into registers starting at 'REGISTER ADDRESS', then 'REGISTER ADDRESS' +1, then 'REGISTER ADDRESS' +2 etc.
Each byte transfer is acknowledged 'ACK' by the FWCM until the 'STOP' condition.

Figure 1.1 (I²C read protocol)



'DATA BYTE 1 & 2' are register values returned from the FWCM. Each byte written is acknowledged 'ACK' by the FWCM , every byte read is acknowledged 'ACK' by the I²C Master. A Not-acknowledge 'NACK' condition is generated by the I²C Master when it has finished reading.

Figure 1.2 (Connection Schematic for OOPic (II) I²C communication)

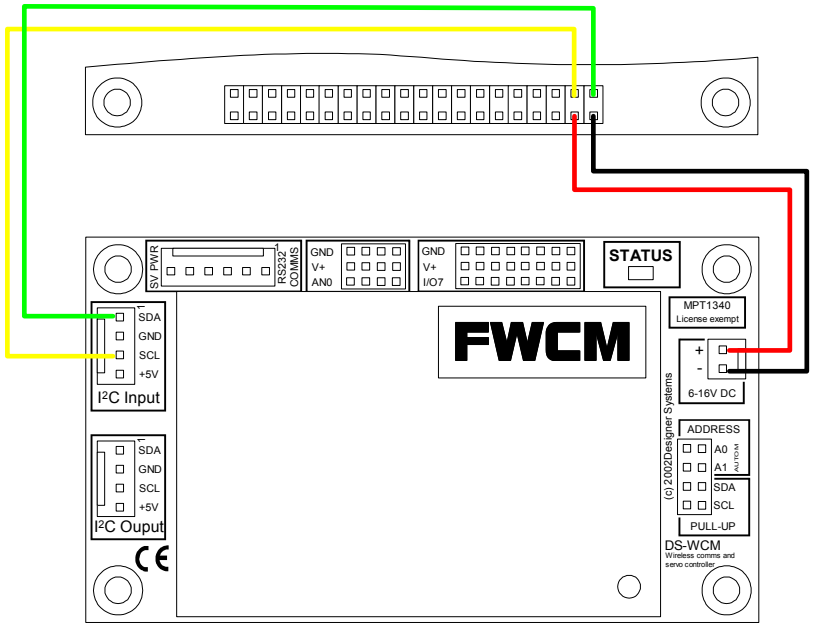


Figure 2.0 (Servo/analogue connection in autonomous mode)

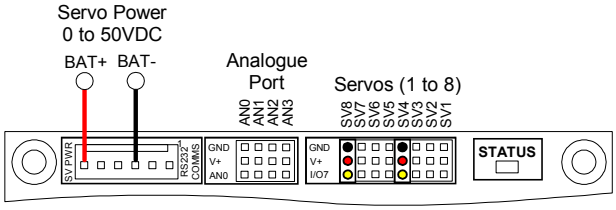


Figure 3.0 (Multi-drop RS232 connection)

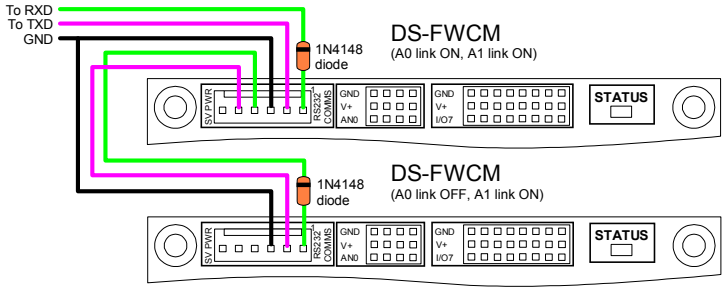


Figure 4.0 (I/O connection)

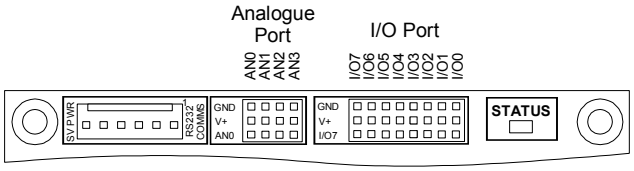
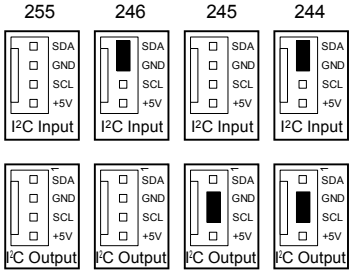
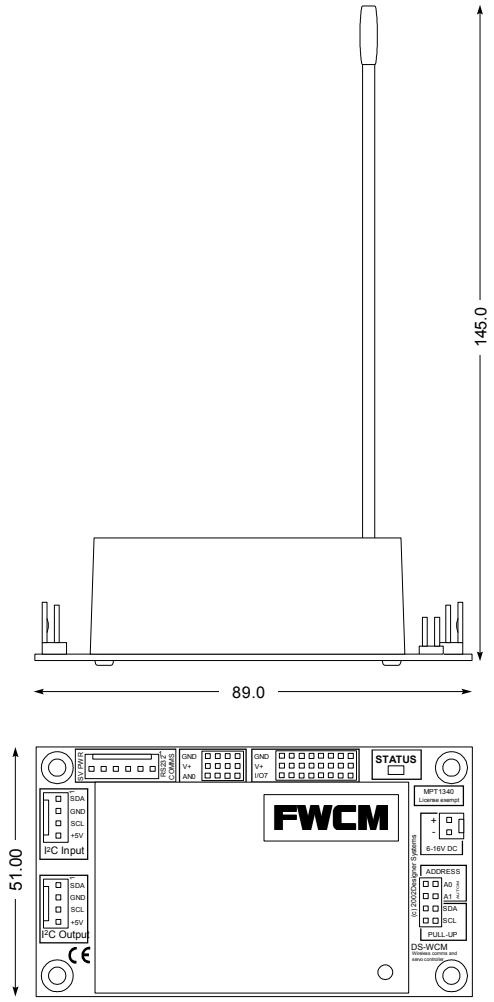


Figure 5.0 (Autonomous address setup)



Mechanical Specifications – Units millimetres



Declaration of Conformity

Apparatus name / model number DS-FWCM

Conformity via Generic Standard EN50081-1
Generic Standard EN50082-1

Conformity criteria For use only within commercial, residential and light industrial applications

We certify that the apparatus identified above conforms to the requirements of Council Directive 89/336/EEC & 73/23/EEC

Signed. **Date** 1/6/02

Having made this declaration the CE mark is affixed to this product, its packaging, manual or warranty.

Manufacturer Designer Systems, 15 Andrew Place, Truro, Cornwall TR1 3HZ, United Kingdom

Description of apparatus Robotic interface peripheral