dsPIC/PIC 'Command Module' – Adapter Board
EDP-CM-PIC-PIM User Manual

Version 1.03

Contents

1. Introduction

The RS-EDP platform is a system, which has been designed to utilise many different manufacturers' microprocessors. To support Microchip's family of devices, the RS-EDP platform uses an adapter board to connect between the RS-EDP baseboard and the Microchip PIM modules. This is referred to as the EDP-CM-PIM module.

Microchip have most of their MCU's available, pre-mounted on 80 pin and 100 pin PIM modules. These are square in shape and use pin headers to connect down to a daughter board, which in the Microchip system is usually some form of evaluation board.

The RS-EDP therefore uses these modules with an adapter board to gain access to the Microchip portfolio of devices. One adapter board can be used with pretty much all of the Microchip PIM modules, from the 8 bit PIC16Cxx family, through to the 16bit dsPIC family and the new 32bit PIC32 family.

The PIM modules are available to order directly from the RS website, and are separately listed in the catalogue from the RS-EDP platform.

The adapter board is configured as a 'Command Module'. The 'Command Module' in a system dictates whether the whole system is a 3.3V one or a 5.0V one. The module has a link option which decides which of the two voltages is used within the system. The Vcc_CM line is set to this level by the link option on this adapter module. This Vcc_CM is used as a reference by the other modules, such as the analogue module for example.

The daughter board remaps the I/O of the PIM module on to the backplane of the RS-EDP system. As the RS-EDP system has a similar concept to the PIM module system from Microchip, you will find most of the PIM modules will correctly map out to the RS-EDP backplane.

During the development of this board, several devices were chosen for trial fits to the RS-EDP system. These included the dsPIC33FJ256MC710, the dsPIC33FJ256GP710 and the PIC32Mx4xxFxxxL devices.

2. MCU Mapping

2.1 MCU Pin Allocation

The MCU pins have been allocated to the backplane as follows

dsPIC33FJ256MC710			**RS-EDP-BASE BOARD**
Pin	**Name**	**Comment**	Name
1	RG15	2 link options	GPIO5_I2S_TX_WS
		2 link options	GPIO9_I2S_RX_WS
2	Vdd		3.3V
3	PWM3H/RE5		MOTORP2H
4	PWM4L/RE6		AN8
5	PWM4H/RE7		AN9
6	AN16/T2CK/T7CK/RC1		GPIO2_MCI_DAT0
7	AN17/T3CK/T6CK/RC2		GPIO4_MCI_DAT1
8	AN18/T4CK/T9CK/RC3		GPIO6_MCI_DAT2
9	AN19/T5CK/T8CK/RC4		GPIO8_MCI_DAT3
10	SCK2/CN8/RG6		SPI_SSC_CLK
11	SDI2/CN9/RG7		SPI_SSC_MTSR_MOSI
12	SDO2/CN10/RG8		SPI_SSC_MRST_MISO
13	#MCLR		#RESIN

14	#SS2/CN11/RG9		SPI_SSC_#CS_NSS
15	Vss		SGND
16	Vdd		3.3V
17	TMS/RA0		GPIO10_MCI_CLK
18	AN20/#FLTA/INT1/RE8		AN2
19	AN21/#FLTB/INT2/RE9		AN3
20	AN5/QEB/CN7/RB5		MOTORH0_ENC0
21	AN4/QEA/CN6/RB4		MOTORH1_ENC1
22	AN3/INDX/CN5/RB3		MOTORH2_ENC2
23	AN2/#SS1/CN4/RB2		CNTRL_SPI_#CS_NSS
24	PGC3/EMUC3/AN1/CN3/RB1	2 link options	AN4
		2 link options	Local EMUC
25	PGD3/EMUD3/AN0/CN2/RB0	2 link options	AN5
		2 link options	Local EMUD
26	PGC1/EMUC1/AN6/OCFA/RB6	2 link options	GPIO33_AD11
		2 link options	Local EMUC
27	PGD1/EMUD1/AN7/RB7	2 link options	GPIO34_AD2
		2 link options	Local EMUD
28	Vref-/RA9		GPIO14_MCI_PWR
29	Vref+/RA10		GPIO12_MCI_CMD
30	Avdd		AN_REF
31	Avss		VAGND
32	AN8/RB8		AN6
33	AN9/RB9		AN7
34	AN10/RB10		AN0
35	AN11/RB11		AN1
36	Vss		SGND
37	Vdd		3.3V
38	TCK/RA1		GPIO24_AD7
39	#U2RTS/RF13		GPIO25_AD15
40	#U2CTS/RF12		GPIO26_AD6
41	AN12/RB12		GPIO27_AD14
42	AN13/RB13		GPIO28_AD5
43	AN14/RB14		GPIO29_AD13
44	AN15/OCFB/CN12/RB15		GPIO30_AD4
45	Vss		SGND
46	Vdd		3.3V
47	IC7/#U1CTS/CN20/RD14		EVM10_GPIO68_ASCO_CTS
48	IC8/#U1RTS/CN21/RD15		EVG20_GPIO69_ASCO_RTS
49	U2RX/CN17/RF4		ASC1_RX_TTL
50	U2TX/CN18/RF5		ASC1_TX_TTL
51	U1TX/RF3		ASC0_TX_TTL
52	U1RX/RF2		ASC0_RX_TTL
53	SDO1/RF8		CNTRL_SPI_MRST
54	SDI1/RF7		CNTRL_SPI_MTSR
55	SCK1/INT0/RF6		CNTRL_SPI_CLK
56	SDA/RG3		CNTRL_I2C_SDA
57	SCL1/RG2		CNTRL_I2C_SCL
58	SCL2/RA2		I2CGEN1_SCL
59	SDA2/RA3		I2CGEN1_SDA
60	TDI/RA4		EVG7_GPIO54
61	TDO/RA5		EVG8_GPIO56
62	Vdd		3.3V
63	OSC1/CLKIN/RC12		Not applicable
64	OSC2/CLKO/RC15		Not applicable
65	Vss		SGND
66	INT3/RA14		EMG_TRAP
67	INT4/RA15		EVM6_GPIO49
68	IC1/RD8		EVM7_GPIO51
69	IC2/RD9		MOTOR_TCO_FB

70	IC3/RD10		EVM8_GPIO53
71	IC4/RD11		EVM9_GPIO55
72	OC1/RD0		MOTORPWM
73	PGD2/EMUD2/SOSCI/CN/RC13	2 link options	GPIO0
		2 link options	local 32kHz sub clock
74	PGC2/EMUC2/SOSCO/T1CK/CN0/RC14	2 link options	GPIO1
		2 link options	local 32kHz sub clock
75	Vss		SGND
76	OC2/RD1		EVG0_GPIO40
77	OC3/RD2		EVG1_GPIO42
78	OC4/RD3		EVG2_GPIO44
79	IC5/RD12		EVM2_GPIO41_CAPADC
80	IC6/CN19/RD13		EVM3_GPIO43
81	OC5/CN13/RD4		EVG3_GPIO46
82	OC6/CN14/RD5		EVG4_GPIO48
83	OC7/CN15/RD6		EVG5_GPIO50
84	OC8/UPDN/CN16/RD7		EVG6_GPIO52
85	Vddcore		Not applicable
86	Vdd		3.3V
87	C1RX/RF0	2 link options	CAN0_RX
		2 link options	GPIO35_AD10
88	C1TX/RF1	2 link options	CAN0_TX
		2 link options	GPIO36_AD1
89	C2TX/RG1	3 link options	GPIO31_ADI2
		3 link options	USB_HOST_D-
		3 link options	USB_DEV_D-
90	C2RX/RG0	3 link options	GPIO32_AD3
		3 link options	USB_HOST_D+
		3 link options	USB_DEV_D+
91	AN22/CN22/RA6		EVM4_GPIO45
92	AN23/CN23/RA7		EVM5_GPIO47
93	PWM1L/RE0		MOTORP0L
94	PWM1H/RE1		MOTORP0H
95	RG14	2 link options	GPIO13_I2S_TX_CLK
		2 link options	GPIO7_I2S_RX_CLK
96	RG12		GPIO11_I2S_RX_SDA
97	RG13		GPIO15_I2S_TX_SDA
98	PWM2L/RE2		MOTORP1L
99	PWM2H/RE3		MOTORP1H
100	PWM3L/RE4		MOTORP2L

2.2 Backplane Resources Used by the MCU

Resources used/available
3.3V
Vcc_CM
AN_REF
#RESIN
SGND
VAGND
AN0
AN1
AN2
AN3
AN4
AN5
AN6
AN7
AN8

AN9
ASC0_RX_TTL
ASC0_TX_TTL
ASC1_RX_TTL
ASC1_TX_TTL
CAN0_RX
CAN0_TX
CNTRL_I2C_SCL
CNTRL_I2C_SDA
I2CGEN1_SCL
I2CGEN1_SDA
CNTRL_SPI_#CS_NSS
CNTRL_SPI_CLK
CNTRL_SPI_MTSR
CNTRL_SPI_MRST
SPI_SSC_#CS_NSS
SPI_SSC_CLK
SPI_SSC_MTSR_MOSI
SPI_SSC_MRST_MISO
EVG0_GPIO40
EVG1_GPIO42
EVG2_GPIO44
EVG3_GPIO46
EVG4_GPIO48
EVG5_GPIO50
EVG6_GPIO52
EVG7_GPIO54
EVG8_GPIO56
EVG20_GPIO69_ASCO_RTS
EVM2_GPIO41_CAPADC
EVM3_GPIO43
EVM4_GPIO45
EVM5_GPIO47
EVM6_GPIO49
EVM7_GPIO51
EVM8_GPIO53
EVM9_GPIO55
EVM10_GPIO68_ASCO_CTS
GPIO0
GPIO1
GPIO5_I2S_TX_WS
GPIO7_I2S_RX_CLK
GPIO9_I2S_RX_WS
GPIO11_I2S_RX_SDA
GPIO13_I2S_TX_CLK
GPIO15_I2S_TX_SDA
GPIO24_AD7
GPIO25_AD15
GPIO26_AD6
GPIO27_AD14
GPIO28_AD5
GPIO29_AD13
GPIO30_AD4
GPIO31_ADI2
GPIO32_AD3
GPIO33_AD11
GPIO34_AD2
GPIO35_AD10
GPIO36_AD1
GPIO10_MCI_CLK
GPIO12_MCI_CMD

GPIO14_MCI_PWR
GPIO2_MCI_DAT0
GPIO4_MCI_DAT1
GPIO6_MCI_DAT2
GPIO8_MCI_DAT3
MOTORP0H
MOTORP0L
MOTORP1H
MOTORP1L
MOTORP2H
MOTORP2L
MOTORH0_ENC0
MOTORH1_ENC1
MOTORH2_ENC2
MOTORPWM
EMG_TRAP
MOTOR_TCO_FB
USB_HOST_D+
USB_HOST_D-
USB_DEV_D+
USB_DEV_D-

2.3 Alphabetical Listing of MCU Pins

Pin	Alphabetical Pin Function
13	#MCLR
14	#SS2/CN11/RG9
40	#U2CTS/RF12
39	#U2RTS/RF13
23	AN2/#SS1/CN4/RB2
22	AN3/INDX/CN5/RB3
21	AN4/QEA/CN6/RB4
20	AN5/QEB/CN7/RB5
32	AN8/RB8
33	AN9/RB9
34	AN10/RB10
35	AN11/RB11
41	AN12/RB12
42	AN13/RB13
43	AN14/RB14
44	AN15/OCFB/CN12/RB15
6	AN16/T2CK/T7CK/RC1
7	AN17/T3CK/T6CK/RC2
8	AN18/T4CK/T9CK/RC3
9	AN19/T5CK/T8CK/RC4
18	AN20/#FLTA/INT1/RE8
19	AN21/#FLTB/INT2/RE9
91	AN22/CN22/RA6
92	AN23/CN23/RA7
30	Avdd
31	Avss
87	C1RX/RF0
88	C1TX/RF1
90	C2RX/RG0
89	C2TX/RG1
68	IC1/RD8
69	IC2/RD9
70	IC3/RD10
71	IC4/RD11

79	IC5/RD12
80	IC6/CN19/RD13
47	IC7/#U1CTS/CN20/RD14
48	IC8/#U1RTS/CN21/RD15
66	INT3/RA14
67	INT4/RA15
72	OC1/RD0
76	OC2/RD1
77	OC3/RD2
78	OC4/RD3
81	OC5/CN13/RD4
82	OC6/CN14/RD5
83	OC7/CN15/RD6
84	OC8/UPDN/CN16/RD7
63	OSC1/CLKIN/RC12
64	OSC2/CLKO/RC15
26	PGC1/EMUC1/AN6/OCFA/RB6
74	PGC2/EMUC2/SOSCO/T1CK/CN0/RC14
24	PGC3/EMUC3/AN1/CN3/RB1
27	PGD1/EMUD1/AN7/RB7
73	PGD2/EMUD2/SOSCI/CN/RC13
25	PGD3/EMUD3/AN0/CN2/RB0
94	PWM1H/RE1
93	PWM1L/RE0
99	PWM2H/RE3
98	PWM2L/RE2
3	PWM3H/RE5
100	PWM3L/RE4
5	PWM4H/RE7
4	PWM4L/RE6
96	RG12
97	RG13
95	RG14
1	RG15
55	SCK1/INT0/RF6
10	SCK2/CN8/RG6
57	SCL1/RG2
58	SCL2/RA2
56	SDA/RG3
59	SDA2/RA3
54	SDI1/RF7
11	SDI2/CN9/RG7
53	SDO1/RF8
12	SDO2/CN10/RG8
38	TCK/RA1
60	TDI/RA4
61	TDO/RA5
17	TMS/RA0
52	U1RX/RF2
51	U1TX/RF3
49	U2RX/CN17/RF4
50	U2TX/CN18/RF5
2	Vdd
16	Vdd
37	Vdd
46	Vdd
62	Vdd
86	Vdd
85	Vddcore
28	Vref-/RA9
29	Vref+/RA10

15	Vss
36	Vss
45	Vss
65	Vss
75	Vss

2.4 Backplane Signal Names and Connections

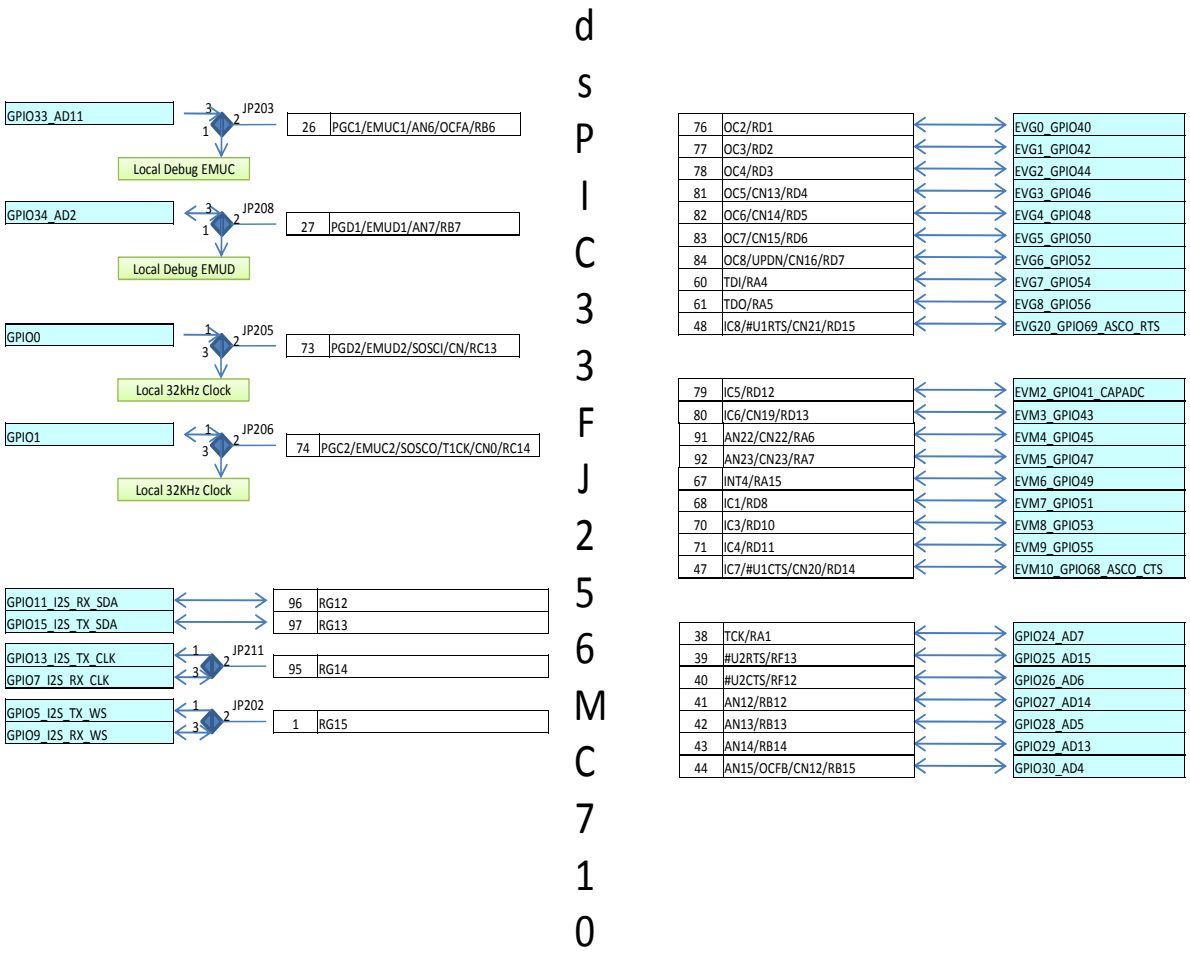
Base Board Signal Name	EDPCON1	EDPCON2	Break Out Connector	
#CS0		53 & 54		
#CS1		55 & 56		
#CS2		57 & 58		
#CS3		59 & 60		
#PSEN		51 & 52		
#RD		45 & 46		
#RESIN		1 & 2	P603	26
#RESOUT		3 & 4	P603	27
#WR		47 & 48		
#WRH		49 & 50		
12V	133		P603	47
12V	134		P603	47
12V	135		P603	47
12V	136		P603	47
12V GND	137		P603	48
12V GND	138		P603	48
12V GND	139		P603	48
12V GND	140		P603	48
3.3V	127		P603	44
3.3V	128		P603	44
3.3V		95 & 96	P603	44
3V BAT	124		P603	42
5.0V	129		P603	45
5.0V	130		P603	45
5.0V		97 & 98	P603	45
A0_AD0		41 & 42		
A1_AD1		39 & 40		
A2_AD2		37 & 38		
A3_AD3		35 & 36		
A4_AD4		33 & 34		
A5_AD5		31 & 32		
A6_AD6		29 & 30		
A7_AD7		27 & 28		
A8_AD8		25 & 26		
A9_AD9		23 & 24		
A10_AD10		21 & 22		
A11_AD11		19 & 20		
A12_AD12		17 & 18		
A13_AD13		15 & 16		
A14_AD14		13 & 14		
A15_AD15		11 & 12		
ALE		43 & 44		
AN_REF	1		P601	6
AN0	3		P603	2
AN1	4		P603	6
AN2	5		P603	1
AN3	6		P603	5
AN4	7		P602	2
AN5	8		P602	4
AN6	9		P602	1

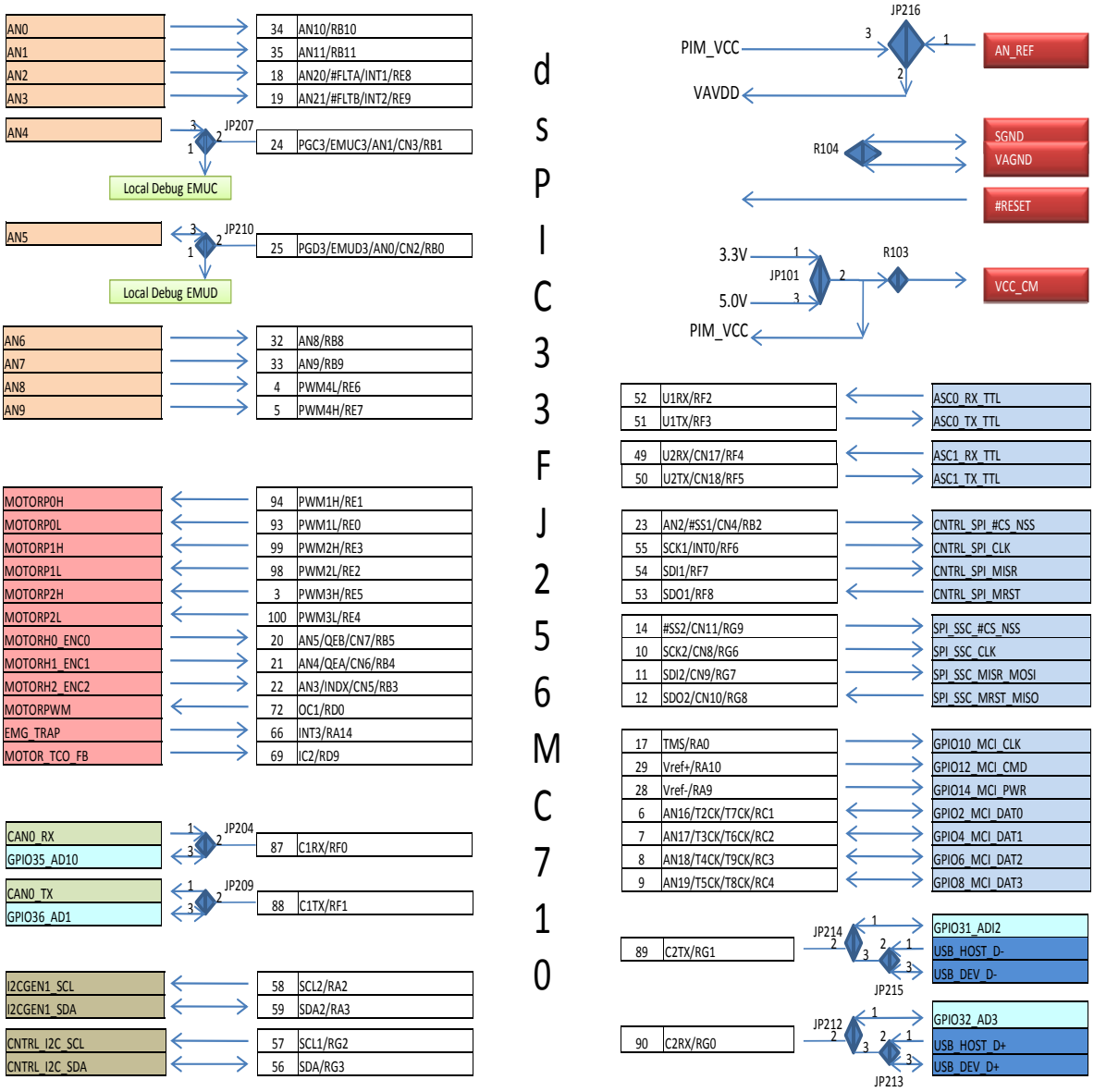
AN7	10		P602	3
AN8	11		P601	2
AN9	12		P601	4
AN10	13		P601	1
AN11	14		P601	3
AN12	15		P603	4
AN13	16		P602	6
AN14	17		P603	3
AN15	18		P602	5
ASC0_RX_TTL	89		P602	30
ASC0_TX_TTL	91		P602	31
ASC1_RX_TTL	93		P602	32
ASC1_RX_TTL_ASC0_DSR	99		P602	35
ASC1_TX_TTL	95		P602	33
ASC1_TX_TTL_ASC0_DTR	97		P602	34
CAN0_RX		61 & 62		
CAN0_TX		63 &64		
CAN1_RX	121		P602	46
CAN1_TX	123		P602	47
CANH0		89 & 90	P603	40
CANL0		91 & 92	P603	41
CNTRL_I2C_SCL		79 & 80	P603	35
CNTRL_I2C_SDA		77 & 78	P603	34
CNTRL_SPI_#CS_NSS		75 & 76	P603	33
CNTRL_SPI_CLK		69 & 70	P603	30
CNTRL_SPI_MRST		71 & 72	P603	31
CNTRL_SPI_MTSR		73 & 74	P603	32
CPU_DACO0_GPIO17	38		P603	7
CPU_DACO1_GPIO19	40		P601	7
EMG_TRAP	114		P601	44
ETH_LNK_LED	111		P602	41
ETH_RX-	109		P602	40
ETH_RX_LED	113		P602	42
ETH_RX+	107		P602	39
ETH_SPD_LED	115		P602	43
ETH_TX-	105		P602	38
ETH_TX+	103		P602	37
EVG0_GPIO40	61		P602	16
EVG1_GPIO42	63		P602	17
EVG2_GPIO44	65		P602	18
EVG3_GPIO46	67		P602	19
EVG4_GPIO48	69		P602	20
EVG5_GPIO50	71		P602	21
EVG6_GPIO52	73		P602	22
EVG7_GPIO54	75		P602	23
EVG8_GPIO56	77		P602	24
EVG9_GPIO57	78		P601	26
EVG10_GPIO58	79		P602	25
EVG11_GPIO59	80		P601	27
EVG12_GPIO60	81		P602	26
EVG13_GPIO61	82		P601	28
EVG14_GPIO62	83		P602	27
EVG15_GPIO63	84		P601	29
EVG16_GPIO64	85		P602	28
EVG17_GPIO65	86		P601	30
EVG18_GPIO66	87		P602	29
EVG19_GPIO67	88		P601	31
EVG20_GPIO69_ASCO_RTS	92		P601	33
EVM0_GPIO21	42		P601	8
EVM1_GPIO23	44		P601	9
EVM2_GPIO41_CAPADC	62		P601	18

EVM3_GPIO43	64		P601	19
EVM4_GPIO45	66		P601	20
EVM5_GPIO47	68		P601	21
EVM6_GPIO49	70		P601	22
EVM7_GPIO51	72		P601	23
EVM8_GPIO53	74		P601	24
EVM9_GPIO55	76		P601	25
EVM10_GPIO68_ASCO_CTS	90		P601	32
GPIO0	21		P603	13
GPIO1	22		P603	15
GPIO2_MCI_DAT0	23		P603	14
GPIO3	24		P603	16
GPIO4_MCI_DAT1	25		P603	17
GPIO5_I2S_TX_WS	26		P603	19
GPIO6_MCI_DAT2	27		P603	18
GPIO7_I2S_RX_CLK	28		P603	20
GPIO8_MCI_DAT3	29		P603	22
GPIO9_I2S_RX_WS	30		P603	21
GPIO10_MCI_CLK	31		P603	23
GPIO11_I2S_RX_SDA	32		P603	24
GPIO12_MCI_CMD	33			
GPIO13_I2S_TX_CLK	34		P603	25
GPIO14_MCI_PWR	35		P603	12
GPIO15_I2S_TX_SDA	36		P603	8
GPIO24_AD7	45		P602	8
GPIO25_AD15	46		P601	10
GPIO26_AD6	47		P602	9
GPIO27_AD14	48		P601	11
GPIO28_AD5	49		P602	10
GPIO29_AD13	50		P601	12
GPIO30_AD4	51		P602	11
GPIO31_ADI2	52		P601	13
GPIO32_AD3	53		P602	12
GPIO33_AD11	54		P601	14
GPIO34_AD2	55		P602	13
GPIO35_AD10	56		P601	15
GPIO36_AD1	57		P602	14
GPIO37_AD9	58		P601	16
GPIO38_AD0	59		P602	15
GPIO39_AD8	60		P601	17
I2C_GEN0_SCL		7 & 8	P603	29
I2C_GEN0_SDA		5 & 6	P603	28
I2C_GEN1_SCL	119		P602	45
I2C_GEN1_SDA	117		P602	44
IRQ_GPIO16_CNTRL_I2C_INT	37		P603	11
IRQ_GPIO18_I2C_GEN0_INT	39		P603	10
IRQ_GPIO20_I2C_GEN1_INT	41		P603	9
IRQ_GPIO22_I2C_INT	43		P602	7
MOTOR_TCO_FB	122		P601	48
MOTORH0_ENC0	116		P601	45
MOTORH1_ENC1	118		P601	46
MOTORH2_ENC2	120		P601	47
MOTORP0H	102		P601	38
MOTORP0L	100		P601	37
MOTORP1H	106		P601	40
MOTORP1L	104		P601	39
MOTORP2H	110		P601	42
MOTORP2L	108		P601	41
MOTORPWM	112		P601	43
SGND	131		P603	46
SGND	132		P603	46

SGND		9 & 10	P603	46
SGND		99 & 100	P603	46
SPI_SSC_#CS_NSS	101		P602	36
SPI_SSC_CLK	98		P601	36
SPI_SSC_MRST_MISO	94		P601	34
SPI_SSC_MTSR_MOSI	96		P601	35
USB_DEBUG_D-		67 & 68		
USB_DEBUG_D+		65 & 66		
USB_DEV_D-		87 & 88	P603	39
USB_DEV_D+		85 & 86	P603	38
USB_HOST_D-		83 & 84	P603	37
USB_HOST_D+		81 & 82	P603	36
VAGND	19		P601	5
VAGND	20		P601	5
Vcc_CM	125		P603	43
Vcc_CM	126		P603	43
Vcc_CM		93 & 94	P603	43

2.5 Mapping Aids





3. Jumper Options

Vcc_CM Command module Voltage Selection Jumper – JP101

The EDP-CM-PIM with a PIM module fitted is designed to be a 'Command Module' in the system. When the module is used as a Command Module, the operating voltage of the complete system needs to be decided via the link option JP101. This provides the back plane with the necessary voltage (Vcc_CM) to instruct all the other modules that the system is either a 3.3V or 5.0V system. i.e. The Analogue Module for example will provide signals up to 3.3V/5.0V accordingly.

This Vcc_CM is also used by the RESET circuitry on the base board. The RESET button will not work for example if this link is not made. There are two possible positions for this jumper, position 1-2 for 3.3V and position 2-3 for 5.0V.

Normally the operating voltage of the PIM module PIC device will determine the operating voltage of the system. Hence the selection jumper is altered accordingly, depending on whether you use a 5.0V or 3.3V PIC device.

The Vcc-CM line and operating voltage for the adapter board and the PIM module are usually the same. They are connected via a zero ohm resistors, R103 & R201. It would be possible to de-solder these resistors and to operate the PIM and adapter board at a different voltage to the Vcc_CM line. Some consideration would have to be given to the different voltages then present in the system and possible bleed paths for current when I/O are possibly at different voltages. The users should check the circuit to ensure safe operation is guaranteed during this case. No further design guidance is offered at this point and this dual voltage feature has not been tested.

Leaving this jumper open would suggest that another module is going to decide what the Vcc_CM voltage would be.

JP101 – option 1-2	Vcc_CM is set to 3.3V
JP101 – option 2-3	Vcc_CM is set to 5.0V
open	Vcc_CM is decided by another module (DEFAULT)

Table: 01 Vcc_CM 'Command Module' Voltage Selection Jumper - JP101 options

Voltage Reference Selector – JP216

The PIM modules have a separate analogue voltage reference supply pin called AVdd. This pin can be connected to either the power supply voltage of the module, Vcc_PIM, or to an external reference provided through the back plane. If the analogue module is fitted, a voltage reference is provided via the backplane called AN_Ref. This selector option will allow the user to connect either of the two voltages to the PIM module reference voltage pin. This jumper should also be used in conjunction with the analogue ground zero ohm link. See section on zero ohm links below.

JP216 – option 1-2	AN Ref is selected as the AVdd signal
JP216 – option 2-3	PIM_Vcc is selected as the AVdd signal (DEFAULT)

Table:02 Voltage Reference Selector Jumper – JP216 options

Emulation & Programmer Jumpers – JP203, JP208 & JP207, JP210

There are two possible channels to which the emulation and programming system can be connected. These are connected via the daughter board to the programming pins on the PIC or dsPIC MCU. These are usually referred to as EMUCx and EMUDx where x is either 1, 2 or 3. The actual designation and pins used, will depend on the PIC/dsPIC fitted to the module.

Most of the development for the PIM module was based on the highly successful dsPIC33FJ256MC710 module, which effectively maps both EMUC1/D1 and EMUC3/D3 to the emulator connections.

Have a look at your PIM module and identify the pins that are responsible for the programming and flashing of the device. These should be connected to pins 24, 25, 26 and 27 of the PIM module. These pins are then selectable via the link options. The ones not used for programming can be used for other functions within the system. Only two pins are used at any one time for debugging and flashing. Select the correct link options accordingly.

For the emulation/programming system to be able to communicate with the PIC, the fuse options for the microcontroller also need to be correctly set.
This is usually done with a few lines of source code.

For the dsPIC33FJ256MC710 device the appropriate fuse option is...

```
/* Fuse In Circuit Debug - 0xf8000e */

_FICD(ICS_PGD1 & JTAGEN_OFF)

/* bit15-bit8 - always set to 1's */
/* bit7 - BKBUG  */
   /* bit6 - COE  */
   /* bit5 - JTAGEN  */
   /* bit4,3,2 - Reserved - write 1's */
   /* bit1,0 - ICS<1:0> - 01 - Communicate on PGC3/EMUC3 & PGD3/EMUD3 */

   /* Possible options are... _FICD( OPT1_ON & OPT2_OFF & OPT3_PLL )

   **   Background Debug Enable Bit:
   **     BKBUG_OFF            Device will Reset in Debug mode
   **     BKBUG_ON             Device will Reset in user mode
   **
   **   Debugger/Emulator Enable Bit:
   **     COE_OFF              Reset in clip-on operational mode
   **     COE_ON               Reset in operational mode
   **
   **   JTAG Enable Bit:
   **     JTAGEN_OFF           JTAG is disabled
   **     JTAGEN_ON            JTAG is enabled
   **
   **   ICD communication channel select bits:
   **     ICS_NONE             Reserved
   **     ICS_PGD3              communicate on PGC3/EMUC3 and PGD3/EMUD3
   **     ICS_PGD2             communicate on PGC2/EMUC2 and PGD2/EMUD2
   **     ICS_PGD1             communicate on PGC1/EMUC1 and PGD1/EMUD1
   **
   */
```

As you can see from here the relevant option is the ICS_PGD1. For this PIM module we could use channel 3 rather than channel 1, in which case the ICS_PGD1 options gets replaced with the ICS_PGD3 option.

JP207 – option 1-2	PIM pin 24 is connected to emulator/programmer
JP210 – option 1-2	PIM pin 25 is connected to emulator/programmer
JP207 – option 2-3	PIM pin 24 is connected to analogue channel AN4 on backplane.
JP210 – option 2-3	PIM pin 25 is connected to analogue channel AN5 on backplane.
JP207 – option – open	PIM pin 24 is not connected (DEFAULT)
JP210 – option – open	PIM pin 25 is not connected (DEFAULT)

Table: 03 Emulation Selection Jumpers - JP207 & JP210 options

JP203 – option 1-2	PIM pin 26 is connected to emulator/programmer (DEFAULT)
JP208 – option 1-2	PIM pin 27 is connected to emulator/programmer (DEFAULT)
JP203 – option 2-3	PIM pin 26 is connected to GPIO33_AD11 on the backplane
JP208 – option 2-3	PIM pin 27 is connected to GPIO34_AD11 on the backplane
JP203 – option – open	PIM pin 26 is not connected
JP208 – option - open	PIM pin 27 is not connected

Table: 04 Emulation Selection Jumpers - JP203 & JP208 options

I2S Jumper Options – JP202, JP211

On the backplane of the RS-EDP module are some signals dedicated to the serial I2S communication interface (Inter Integrated-circuit Sound).
I2S is a high speed serial standard used primarily for digital audio. This digital audio used a clock and a data signal.

The Wikipedia definition of I2S is detailed below.

I²S consists, as stated above, of a bit clock, a word select and the data line. The bit clock pulses once for each discrete bit of data on the data lines. The bit clock will operate at a frequency which is a multiple of the sample rate. The bit clock frequency multiplier depends on number of bits per channel, times the number of channels. So, for example, CD Audio with a sample frequency of 44.1kHz, with 32 bits of precision per (2) stereo channels will have a bit clock frequency of 2.8224MHz. The word select clock lets the device know whether channel 1 or channel 2 is currently being sent, since I²S allows two channels to be sent on the same data line. Transitions on the word select clock also serve as a start-of-word indicator. The Word clock line pulses once per Sample, so while the Bit clock runs at some multiple of the sample frequency, the word clock will always match the sample frequency. For a 2 channel (stereo) system, the word clock will be a square wave, with an equal number of Bit clock pulses clocking the data to each channel. In a Mono system, the word clock will pulse one bit clock length to signal the start of the next word, but will no longer be square, rather all Bit clocking transitions will occur with the word clock either high or low.

Standard I²S data is sent from MSB to LSB, starting at the left edge of the word select clock, with one bit clock delay. This allows both the Transmitting and Receiving devices to not care what the audio precision of the remote device is. If the Transmitter is sending 32 bits per channel to a device with only 24 bits of internal precision, the Receiver may simply ignore the extra bits of precision by not storing the bits past the 24th bit. Likewise, if the Transmitter is sending 16 bits per channel to a Receiving device with 24 bits of precision, the receiver will simply Zero-fill the missing bits. This feature makes it possible to mix and match components of varying precision without reconfiguration.

There are left justified I²S streams, where there is no bit clock delay and the data starts right on the edge of the word select clock, and there are also right justified I²S streams, where the data lines up with the right edge of the word select clock. These configurations however are not considered standard I²S.

The PIM module has support for I2S and data can be transmitted as a master from the PIC/dsPIC or received into the module as a slave from a master somewhere else in the system.

The backplane has two signals for I2S_WS, one for I2STX_WS and one for I2SRX_WS. As the PIC can be either master or slave the user can select the appropriate settings for this.

The backplane also has two signals for the I2S_CLK as well, one for I2SRX_CLK and one for I2STX_CLK. The user can therefore select which one he wants depending on whether the PIC is transmitting as a master or receiving as a slave.

JP202 – option 1-2	I2STX_CLK – PIC is master I2S device
JP211 – option 1-2	I2STX_WS - PIC is master I2S device
JP202 – option 2-3	I2SRX_CLK - PIC is slave I2S device

JP211 – option 2-3	I2SRX_WS - PIC is slave I2S device
JP202 – option – open	I2STX_CLK is not connected (DEFAULT)
JP211 – option – open	I2STX_WS is not connected (DEFAULT)

Table 05: I2S selection jumpers – JP202 & JP211 options

It is worth checking that if other modules are using I2S, which lines are being used.

CAN Bus – Jumper Option – JP204 & JP209

The jumper options can used to route the CAN Tx and CAN Rx signals from the PIC/dsPIC on to the backplane. These signals can then be converted to physical layer signals via a communications module. If the CAN bus is not being used or the PIC does not support CAN bus then these signals can be used for other purposes.

JP204 – options 1-2	PIM pin 87 is routed to CAN TX on backplane
JP209 – options 1-2	PIM pin 88 is routed to CAN RX on backplane
JP204 – options 2-3	PIM pin 87 is routed to GPIO35_AD10
JP209 – options 2-3	PIM pin 88 is routed to GPIO36_AD1
JP204 – option – open	PIM pin 87 is not connected (DEFAULT)
JP209 – option – open	PIM pin 88 is not connected (DEFAULT)

Table:06 CAN Bus selection – JP204 & JP209 options

USB Link Options – JP212, JP213, JP214, JP215

The RS-EDP is equipped with several lines for USB communication including a separate USB host and a separate general USB. The PIM module can have access to both of these signals via connections present on the adapter board.

JP212 – option 1-2	PIM pin 90 is routed through to D+
JP214 – option 1-2	PIM pin 89 is routed through to D-
JP212 – option 2-3	PIM pin90 is routed through to GPIO32_AD3
JP214 – option 2-3	PIM pin89 is routed through to GPIO31_AD12
JP212 - option – open	PIM pin 90 is not connected (DEFAULT)
JP214 – option – open	PIM pin 89 is not connected (DEFAULT)

Table: 07 USB selection jumpers – JP212 & JP214 options

The USB link options can further be routed to either the USB host channel or the standard USB connections on the back plane.

JP213 – option 1-2	D+ is routed to USB HOST D+
JP215 – option 1-2	D- is routed to USB HOST D-
JP213 – option 1-2	D+ is routed to USB DEV D+
JP215 – option 1-2	D- is routed to USB DEV D-
JP213 – option – open	D+ is not connected (DEFAULT)
JP215 – option – open	D- is not connected (DEFAULT)

Table: 08 USB Connections - JP213 & JP215 options

Note: USB On The Go uses additional signals as well, notably USB ID, and VBUSON. These can be routed on to the backplane. Check the PIM module configuration to see the pins actually used for these.

32KHz Sub Clock Option – JP205 & JP206

The adapter module has been designed to support a 32KHz watch crystal sub clock. To enable this feature for the PIM modules that can support it, JP205 & JP206 should be inserted as detailed in the table below. If the 32KHz sub clock is not required, it can be bypassed and the circuitry re-routed to the backplane GPIO signals.

JP205 – option 2-3	PIM pin 73 is routed to 32KHz circuitry
JP206 – option 2-3	PIM pin 74 is routed to 32KHz circuitry
JP205 – option 1-2	PIM pin 73 is routed to GPIO0 on the backplane
JP206 – option 1-2	PIM pin 74 is routed to GPIO1 on the backplane
JP205 – option – open	PIM pin 73 is not connected (DEFAULT)
JP206 – option – open	PIM pin 74 is not connected (DEFAULT)

Table: 09 Sub clock jumpers – JP205 & JP206 options

Vdd_Core Jumper – JP201

The VDDCORE voltage is usually managed on the PIM module itself with various jumpers and capacitors to ensure this voltage is correct. The PIM adapter module provides some additional flexibility with this function. For normal operation it should be left disconnected.

JP201 – option open	The VDDCORE voltage will be managed by the PIM module (DEFAULT)
JP201 – option 1-2	The VDDCORE voltage is Vcc_CM
JP201 – option 2-3	The VDDCORE voltage is SGND

Table: 10 The VDDCORE Jumper – JP201 options

4. Zero Ohm Links

Analogue Ground Reference – R104

The analogue ground from the back plane can be made to be the same as the signal ground if required. AD signals can be passed from the analogue module to the command module. These analogue signals have their own separate ground return. The signal ground and the analogue ground can be connected together on the PIM module adapter board via a zero ohm link resistor R104.

R104 – option populated	SGND and VAGND are shorted together on the PIM adapter (DEFAULT)
R104 – option removed	SGND and VAGND are not connected on the PIM adapter

Table: 11 Analogue Ground Reference link R104

PIM Voltage Vcc_PIM & Vcc_CM Connection - R103

This connects the Vcc_CM voltage to the operating voltage of the PIM adapter module. The default setting for this is to populate this position with a zero ohm link. This is discussed more in the section on jumper settings, for the Vcc_CM.

With this link in place the voltage used on the PIM adapter module will be made the same as for the Vcc_CM voltage which is selectable via link option JP101.

R103 – option – populated	Vcc_PIM adapter board & Vcc_CM are connected together.(DEFAULT)
R103 – option - removed	Vcc_PIM adapter board is not connected to the Vcc_CM line

Table: 11 Adapter board voltage Vcc_PIM & Vcc_CM Conection – R103

PIM Module Voltage & PIM Adapter Module Voltage – R201

The PIM module operating voltage can be made the same as the operating voltage for the PIM adapter board by inserting the zero ohm link, R201 on the board. This is the normal setting for operation and should not be removed without good reason.

R201 – option – populated	The Vcc_PIM adapter voltage is connected to the Vcc_PIM on the module.(DEFAULT)
R202 – option - removed	The supply voltage for the adapter board is disconnected from the PIM module.

Table : 12 Adapter board voltage to PIM voltage connection – R201

Main Oscillator Selections – R202, R203, R204

The main oscillator on the PIC/dsPIC devices can be selected as either, an internal RC oscillator, an external Xtal resonator type or an external clock module. The adapter board has been designed to accommodate all of these options.

To use the internal RC oscillator the zero ohm links, R203 and R204 need to be removed.

To use an external crystal or ceramic oscillator then the zero ohm links R202, R203 need to be populated.

To use an external clock oscillator module the module needs to be soldered on to the board. R203 and R204 should be removed.

Clock Type	R202	R203	R204
External Xtal (DEFAULT)	Removed	Populated	Populated
External Clock module	Don’t care	Removed	Removed
Internal RC	Don’t care	Removed	Removed

Table:13 Main Oscillation Selection Links

Hardware Setup

Select the appropriate jumper and link options for your design and then insert your PIM module into the adapter board. Once inserted, plug the adapter board into the RS-EDP base board along with the Communication Module. The communications board is fitted with a nine way D connector, which is mapped to one of the serial channels on the PIM module. Connect this to the host PC.
Power up the RS-EDP board via the power adapter provided.

Check the Vcc_CM voltage on the break out connector of the base board, to see if it is what you have selected via the Vcc_CM jumper. If all is ok, the +5V, the 3.3V and the Vcc_CM voltages should be present on the break out connector.

The green power on LED lamp should illuminate when power is applied. Press the reset button and the red LED should illuminate. This should go off when the button is released.

5. Software Support

5.1 dsPIC33FJ256MC710

The Microchip adapter board is designed to be used with Microchip PIM modules and the user is expected to have some familiarity with the series of MCU's he is looking to use.

As the adapter board can be used with many platforms including PIC16Cxx, dsPIC30Fxxxx, dsPIC33FJxxxx and PIC32 devices it is not possible to provide a whole series of drivers for all of these devices.

The basic development work done was using a dsPIC33FJ256MC710 device a 16 bit device from the latest generation of dsPIC devices from Microchip. Consequently driver support is included for this part to communicate with the analogue module, the digital module, the communications module, and the MC2 motor drive module. It may be worth purchasing one of these PIM modules so you have a working reference by which you can check the base board and the modules.

The software was written and developed using MPLAB Version 8.14 and the C30 compiler from Microchip. These are available to download free of charge from Microchips web site, although some registration is required to obtain the compiler.

The software drivers allow the user to read signals from most of the popular peripheral boards and also to control the motors for the MC2 module.

These modules can be tested via a test menu, which will require the use of the 'Communications Module' and a terminal emulator for the host PC.

The configuration of the terminal emulator is as follows...

The serial configuration at the time of writing this support documents is as follows.
Check the C source code to see if this has changed since this document was written.

Baud rate: 115,200 baud
Data bits: 8
Stop bits: 1
Parity: None
Flow control: None

Start MPLAB and open up the MPLAB project, 'PIM_dsPIC33FJ256MC710_General'.

Plug in the emulator/programmer into the PIM adapter board and attempt to connect to it.

Try starting with the Real-ICE connected as a programmer and selected for 'Release' rather than 'Debug'. Recompile the code and ensure all of your paths are correct for the project and then attempt to flash the board.

The fuse options you have selected may be important at this stage and make sure you have the correct oscillator selections and the correct debug/programmer setting for the fuse options.
For the dsPIC mentioned and the provided software the fuse options can be viewed in the header file called 'fuse_options.h'

The emulator selection is important and this is detailed in the sections relating to jumper options earlier, and must be read before proceeding.

If you are able to flash the PIM module with this code then you should see some serial output on the terminal emulator when the code starts to run. Problems with flashing code would almost certainly be due to the fuse options not being correct or the link and jumper options not being correctly set.

Work through the menu options provided with the test suite. The default oscillator configuration provided is for the external crystal populated on the adapter module.

If you are having problems with the oscillator, then change the fuse setting to run with on board high speed RC oscillator. The code to do this is included and commented out in the 'fuse_header.h' There is also a #define in the 'defines.h' header file which needs to be changed also.