

M68HCS12 Parallel I/O (Excerpt from a completely new chapter on HCS12 parallel I/O).

The complete chapter details are at

http://www.coe.montana.edu/ee/courses/ee/ee371/pdffiles_parallel_io.pdf

Port P

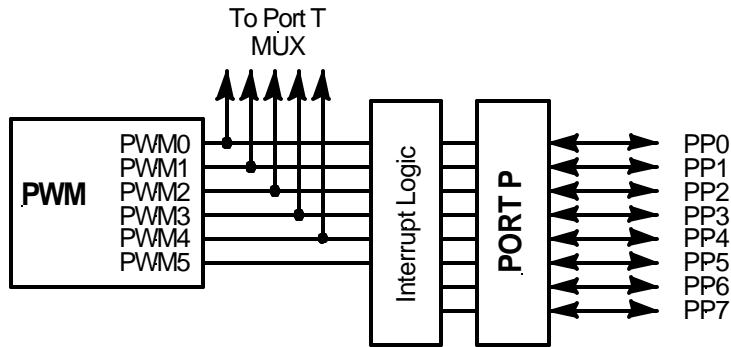


Figure 1 Port P (Only PP5 available in the Lab)

Port P is an 8-bit, general-purpose I/O port. Alternatively, up to six pulse-width modulated waves can be output. In some versions of the microcontroller, such as the 9S12C32 where the Port P pins are not available at the chip, five of the six PWM outputs can be routed to Port T. The *Module Routing Register – MODRR*, as discussed below is used to do this.

PTP – Base + \$0258 – Port P I/O Register

	Bit 7	6	5	4	3	2	1	0
Read:	PTP7	PTP6	PTP5	PTP4	PTP3	PTP2	PTP1	PTP0
Write:								
Reset:	0	0	0	0	0	0	0	0
PWM	-	-	PWM5	PWM4	PWM3	PWM2	PWM1	PWM0

See Also:

Topic	Register	Chapter
Port P Input Register	PTIP (Base + \$0259)	
Data Direction Register	DDRP (Base + \$025A)	
Reduced Drive Control	RDRP (Base + \$025B)	
Pull-Up or Pull-Down Enable	PERP (Base + \$025C)	
Polarity Select Register	PPSP (Base + \$025D)	
Port P Interrupt Enable	PIEP (Base + \$025E)	
Port P Interrupt Flag	PIFP (Base + \$025F)	
Pulse-Width Modulation		

Data Direction Registers

Each bit in the bidirectional data registers may be programmed to be either input or output. When the CPU is reset, all registers are placed in the input mode, and you must set bits in a *Data Direction Register – DDR* to change input bits to be outputs. Each DDR has the format shown

below where x is A, B, AD, E, J, M, P, S or T; the addresses for each are given in shows how to initialize the most significant nibble in Port P for output. If a port has a mixture of input and output bits, writing to the port affects only those bits that are outputs. Reading the port returns the values on the input bits as well as the last values output to the output bits.

DDRx – Base + \$(Table 1) – Port x Data Direction Register

	Bit 7	6	5	4	3	2	1	0
Read:	DDRx7	DDRx6	DDRx5	DDRx4	DDRx3	DDRx2	DDRx1	DDRx0
Write:								
Reset:	0	0	0	0	0	0	0	0
Default:	Input	Input	Input	Input	Input	Input	Input	Input
	= reserved, unimplemented or cannot be written to.							
	Read: Anytime. Write: Anytime							
DDRx7:DDRx0								
Data Direction Control Bits.								
0 = Associated pin is high-impedance input (default).								
1 = Associated pin is an output.								
Data direction register bits determine the direction of the corresponding data register. The direction can be set individually for each bit in the port.								

Table 1 Data Direction Register Addresses

Data Direction Register	Address
DDRA	\$0002
DDRB	\$0003
DDRAD	\$0272
DDRE ¹	\$0009
DDRJ	\$026A
DDRM	\$0252
DDRP	\$025A
DDRS	\$024A
DDRT	\$0242

Example 1 Initializing the Data Direction Register

Metrowerks HC12-Assembler
 (c) COPYRIGHT METROWERKS 1987-2003

```

Rel.Loc   Obj. code Source line
-----
1           ; Set the register BASE address
2      0000 0000 BASE:   EQU   $0000
3      0000 0258 PTP:    EQU   BASE+$0258; Port P I/O
4      0000 025A DDRP:   EQU   BASE+$025A; DDR Port P
5           ; Define bits to be output and input
6           ; 1 = output, 0 = input
7      0000 00F0 OBITS: EQU   %11110000
  
```

¹ You cannot configure bits-0 and -1 in Port E to be output because they are associated with interrupt inputs.

```

8          ;          . . .
9          ; Set direction register for Port P
10000000 1C02 5AF0          bset  DDRP,OBITS
11         ;          . . .
12         ; Output data to bits 7 - 4
13000004 86B0          ldaa  #%10110000
14000006 7A02 58          staa  PTP
15         ; Read data on bits 3 - 0
16000009 B602 58          ldaa  PTP

```

Example 2 Initializing the Data Direction Register with C

```

/*****
 * Define bits to be output and input
 *   1 = output, 0 = input
 *****/
#define OBITS 0xf0 /* 240 */
void main(void) {

    /* Set direction register for Port P */
    DDRP = OBITS;
    /* . . . */
    /* Output data to bits 7 - 4 */
    PTP = 0xB0;
    /* . . . */
    /* Or here is another way */
    /* These compile to bit set and bit clr instructions */
    /* Set direction register for Port P */
    DDRP_DDRP7 = 1;
    DDRP_DDRP6 = 1;
    DDRP_DDRP5 = 1;
    DDRP_DDRP4 = 0;
    /* . . . */
    /* Output data to bits 7 - 4 */
    PTP_PTP7 = 1;
    PTP_PTP6 = 0;
    PTP_PTP5 = 1;
    PTP_PTP4 = 1;
}

```

I/O Port Bit Electronics

Reduced Drive Control

Drive refers to the capability of the output circuitry to source current to whatever is connected to the pin. High drive current is an advantage when the output must drive a capacitive load. High drive current results in higher speed switching between logic levels. Unfortunately, high drive current means higher power consumption and the increased likelihood of radio frequency interference (RFI). The *RDRx*, *Port x Reduced Drive* register allows you to reduce the drive level

to reduce power consumption and RFI emissions for the registers in the Port Integration Module as shown in Table 2. If any of the port bits are used for an input, the reduced drive control bit is ignored.

RDRx – Base + \$(Table 2) – Port x Reduced Drive Register

	Bit 7	6	5	4	3	2	1	0
Read:	RDRx7	RDRx6	RDRx5	RDRx4	RDRx3	RDRx2	RDRx1	RDRx0
Write:								
Reset:	0	0	0	0	0	0	0	0
RDRAD:	Full Drive	Full Drive	Full Drive	Full Drive	Full Drive	Full Drive	Full Drive	Full Drive
RDRJ:	Full Drive	Full Drive						
RDRM:			Full Drive	Full Drive	Full Drive	Full Drive	Full Drive	Full Drive
RDRP:	Full Drive	Full Drive	Full Drive	Full Drive	Full Drive	Full Drive	Full Drive	Full Drive
RDRS:					Full Drive	Full Drive	Full Drive	Full Drive
RDRT:	Full Drive	Full Drive	Full Drive	Full Drive	Full Drive	Full Drive	Full Drive	Full Drive
	= reserved, unimplemented or cannot be written to.							
	Read: Anytime. Write: Anytime							
RDRx7:RDRx0								
Reduced Drive for Ports								
0 = Full drive strength at output.								
1 = Associated pin drives at about 1/3 the full drive output.								

Table 2 Reduced Drive Enable Registers

Pull Device Enable Register	Address Base +
RDRAD	\$0273
RDRJ	\$026B
RDRM	\$0253
RDRP	\$025B
RDRS	\$024B
RDRT	\$0243

Pull-up or Pull-down Control

It is a good electronic design practice to tie unused input pins to either a high or low logic level. In CMOS devices this reduces the chance for a potentially destructive condition called *latch-up* to occur. The M68HCS12 provides a variety of registers to enable pull-up or pull-down resistors on ports that are configured as inputs. The choice of implementing a pull-up or pull-down is controlled by the polarity selection registers, and the pull-ups and pull-downs are enabled by the PERx register for Ports AD, J, M, P, S and T. See Table 3. It is not possible to enable a pull-up or pull-down resistor if the port bit is used as an output.

PERx – Base + \$(Table 3) – Port x Pull Device Enable Register

	Bit 7	6	5	4	3	2	1	0
Read:	PERx7	PERx6	PERx5	PERx4	PERx3	PERx2	PERx1	PERx0
Write:								
Reset:								
PERAD:	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
PERJ:	Disabled	Disabled						
PERM:			Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
PERP:	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
PERS:					Enabled	Enabled	Enabled	Enabled
PERT:	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
	= reserved, unimplemented or cannot be written to.							
	Read: Anytime. Write: Anytime							
PERx7:PERx0								
Pull Device Enable Bits.								
0 = Pull-up or pull-down device is disabled.								
1 = Either a pull-up or pull-down device is enabled.								
<u>Port Specific Notes:</u>								
PERAD (ATD): It is not possible to enable pull devices when an associated A/D channel is enabled simultaneously.								
PERJ (Port J): Configures pull-up or pull-down for bits 6 and 7 used as input or wired-or output. It has no effect if the port is used as a normal, active pull-up (push-pull) output.								
PERM (Port M): Pull-up devices enabled at reset.								
PERP (Port P): Pull devices disabled at reset.								
PERS (Port S): Configures pull device for input or wired-or output. Pull-up devices enabled at reset.								
PERT (Port T): Pull devices disabled at reset.								

Table 3 Pull Device Enable Register Addresses

Pull Device Enable Register	Address Base +
PERAD	\$0274
PERJ	\$026C
PERM	\$0254
PERP	\$025C
PERS	\$024C
PERT	\$0244

Polarity Selection

The polarity (pull-up or pull-down) of the pull devices enabled by the PERx registers is controlled by the *Polarity Select Registers*.

PPSx – Base + \$(Table 4) – Port Polarity Select Register

	Bit 7	6	5	4	3	2	1	0
Read:	PPSx7	PPSx6	PPSx5	PPSx4	PPSx3	PPSx2	PPSx1	PPSx0
Write:								
Reset:	0	0	0	0	0	0	0	0
PPSAD:	Pull-up	Pull-up	Pull-up	Pull-up	Pull-up	Pull-up	Pull-up	Pull-up
PPSJ:	Pull-up	Pull-up						
PPSM:			Pull-up	Pull-up	Pull-up	Pull-up	Pull-up	Pull-up
PPSP:	Pull-up	Pull-up	Pull-up	Pull-up	Pull-up	Pull-up	Pull-up	Pull-up
PPSS:					Pull-up	Pull-up	Pull-up	Pull-up
PPST:	Pull-up	Pull-up	Pull-up	Pull-up	Pull-up	Pull-up	Pull-up	Pull-up
	= reserved, unimplemented or cannot be written to.							
	Read: Anytime. Write: Anytime							
PPSx7:PPSx0								
Pull Device Polarity Select.								
0 = Pull-up device is connected to the associated port pin, if enabled by the associated bit in the PERx register and the port pin is used as an input (default).								
1 = Pull-down device is connected to the associate port pin.								
<u>Port Specific Notes:</u>								
PPSP (Port P): The register selects both the polarity of the pull device and the active edge of associated interrupt request.								
0 = Pull-up selected and falling edge on the associated Port P pin sets the associate flag bit in the PIFP register.								
1 = Pull-down selected and rising edge on the associated Port P pin sets the associate flag bit in the PIFP register.								
PPSJ (Port J): The register selects the polarity of both the pull device and the active edge of associated interrupt request.								
0 = Pull-up selected and falling edge on the associated Port J pin sets the associate flag bit in the PIFP register.								
1 = Pull-down selected and rising edge on the associated Port J pin sets the associate flag bit in the PIFP register.								

Table 4 Pull Device Polarity Select Register Addresses

Pull Device Enable Register	Address Base +
PPSAD	\$0275
PPSJ	\$026D
PPSM	\$0255
PPSP	\$025D
PPSS	\$024D
PPST	\$0245

Port P Interrupts

The Port P interrupt vector is at \$FF8E:FF8F. Include the following line in your .prm file:

```
VECTOR ADDRESS 0xFF8E name_of_your_isr
```