

EPIC Rotary Module

Rotary Module

The EPIC rotary module will support 20 rotaries and 4 optical joysticks. The rotaries can be 2 phase optical, 2 phase mechanical, or Clockwise/Counter-clockwise(CW/CCW) pulse (such as ALPS SRBM series).

The rotaries can be assigned CW/CCW (default) or 2 phase by setting the configuration bit associated with the rotary. The rotary module number is 7. Rotary 0 is controlled by bit 0 of row 0, rotary 1 by bit 1 of row 0, rotary 8 bit 0 of row 1, etc.

Example:

```
#define OUTPUT 2           ;EPIC type assignment for output modules.
Definemodule(7,OUTPUT,0,3) ;only assign first 3 rows as outputs to allow SETPOINT
    ;command to work.
:INIT{
    Senddata(7,2,6)       ;will assign rotaries 17 and 18 as 2 phase
                          ;row 2 = rotaries 16-19 (bits 4-7 not used)
}
```

Each rotary has 2 bits associated with it. When assigned a CW/CCW one bit is CW and the other is CCW. When assigned as 2 phase one bit is direction and the other is count. The count toggles with each detent.

Typical CW/CCW (Alps) code:

```
:inc_A{keyhit(equal)}
:dec_A{keyhit(kbminus)}

definebutton(700,on,dec_A)   ;CCW
definebutton(700,off,dec_A)  ;CCW
definebutton(701,on,inc_A)   ;CW
definebutton(701,off,inc_A)  ;CW
```

Typical 2 phase code:

```
:inc_A{keyhit(equal)}
:dec_A{keyhit(kbminus)}

:rotary1{ifactive(701) jump inc_A else jump dec_A} ;701 is direction bit.

definebutton(700,on,rotary1)
definebutton(700,off,rotary1)
```

To assign the button numbers in BUTTON.CFG:

1. create a SCANALL.EPL file that will scan all buttons on default modules.

```
#define FASTSCAN 0
#define SLOWSCAN 1
definemodule(0,SLOWSCAN,0,7)
definemodule(1,SLOWSCAN,0,16)
definemodule(2,SLOWSCAN,0,16)
definemodule(7,FASTSCAN,0,5) ;rotarys 0-19 (module 7, rows 0-4)
:init{setpostkey(2)} ;this is needed to put something in the
```

;command area for checksum

;calculations

2. compile:

EPL SCANALL

3. LOADEPIC SCANALL

4. TEST128

5. select module 7

6. click on Edit

7. click on button to assign. Enter the desired button number. If already assigned you will get an error message stating the current assignment. Entering "-1" will unassign a button.

8. Exit and BUTTON.CFG file will be written.

******* Optical Joysticks *******

The RJ45 ports marked OPTO 0/1 and OPTO 2/3 are 16 bit 2 phase optical rotary ports.

OPTO 0 is on Module 7, row 8,9 with row 8 as the low byte.

OPTO 1 is on Module 7, rows 10,11

OPTO 2 is on Module 7, rows 12,13

OPTO 3 is on Module 7, rows 14,15

OPTOs 0/1 can be preset by writing to the rows with a SENDDATA(mod,row,data) command or SendRawData16(mod,row,data). The Senddata command is 8 bit.

DONOT specify these rows(8-15) in a DEFINEMODULE command. Any rows that are assigned as OUTPUT with a DEFINEMODULE command will be refreshed every 1 second and will keep overwriting the rows.

These rows should also not be defined as input except for test purposes to see them in TEST128.

To include the opto rows add:

DEFINEMODULE(7,FASTSCAN,8,8)

to the SCANALL.EPL

EPL SCANALL

LOADEPIC SCANALL

TEST128

select module 7

To use the optical joysticks:

******* Optical Joystick *******

; Tempory method for extracting optical joysticks

;

; Optical Joysticks are on module 7

;Each joystick uses 2 rows

;Opto0 rows 8,9

;Opto1 rows 10,11

;Opto2 rows 12,13

;Opto3 rows 14,15

;

;Controls should be moved to minimum position and CALIBRATE16 button pressed. This will reset the Opto registers to 0

;DONOT define rows above 3 on rotary module as OUTPUT

#define RESET16 7,2,0b0001000 ;clears optos to 0

#define FASTSCAN 0

#define SLOWSCAN 1

#define OUTPUT 2

definemodule(0,1,0,7)

```

definemodule(7,FASTSCAN,0,5) ;Rotary rows
definemodule(7,OUTPUT,0,3) ;define control bit rows

definedisplay(0,2,24,5,9999,0,TRUE,0b00011111) ;define a display to show value
word(opto0)
word(opto1)
Byte(control0)
Byte(control1)

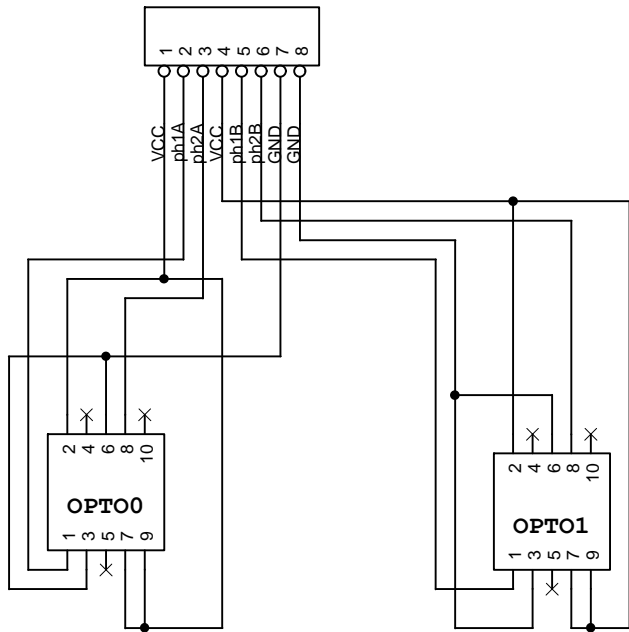
:init{
    wrepic(0x425C,0x78) ;define source module, row of 16bit analogs. Opto0
    wrepic(0x425D,0x7A) ;define source module, row of 16bit analogs. Opto1
; wrepic(0x425E,0x7C) ;define source module, row of 16bit analogs. Opto2
; wrepic(0x425F,0x7E) ;define source module, row of 16bit analogs. Opto3
    setmap(0,-1) ;set analog0 (ailerons) to use internal variable
    setmap(1,-1) ;set analog1(elevator) to use internal variable
    jump(loop)
}
:loop{
    pushc(0)
    exec(92) ;value of 16 bit analog to stack
    popv16(opto0) ;stack to optical0 value
    setdisplay(0,opto0) ;show optical value on display
    ; may want to shift value to the right to scale if range is greater than 255
    pushv16(opto0)
    ; pushc(1) ;amount to shift right. 1= divide by 2, 2 = divide by 4
    ; exec(89)
    ;
    popv8(control0) ;lower 8 bits of 16 to control0
    setpot(0,control0) ;set aileron value

    pushc(1)
    exec(92) ;value of 16 bit analog to stack
    ; opto1 already on the stack, no need to remove unless displaying
    ; may want to shift value to the right to scale if range is greater than 255
    ; pushc(1) ;amount to shift right. 1= divide by 2, 2 = divide by 4
    ; exec(89)
    ;
    popv8(control1) ;lower 8 bits of 16 to control0
    setpot(1,control1) ;set aileron value
    delay(2) ;40 milliseconds
    jump(loop)
}
:Calibrate16{setpoint(RESET16)}

definebutton(15,on,calibrate16)

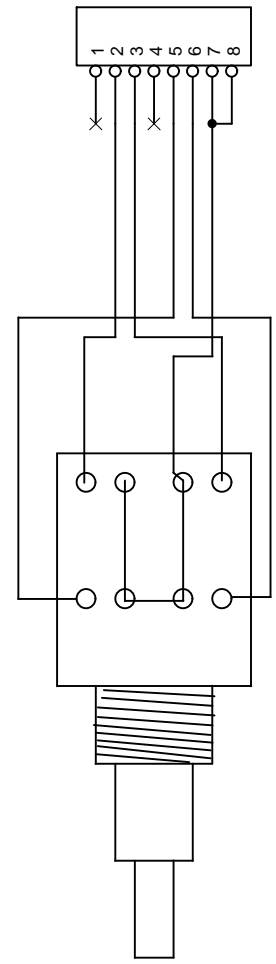
```

Top view clip down
RJ45



Hewlett Packard HEDS-7163
Servo Systems ADC-240
800-922-1103
www.servosystems.com

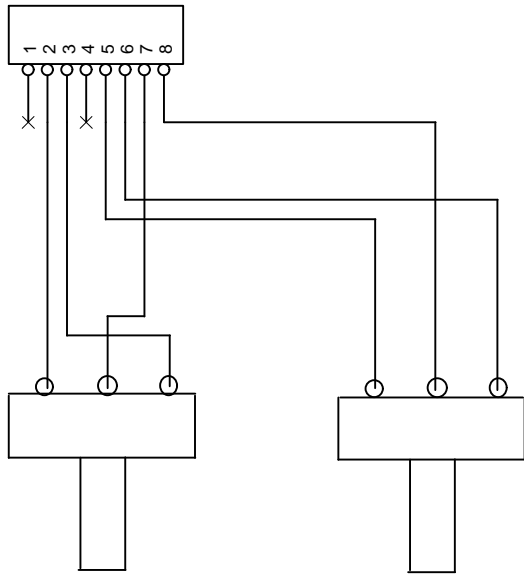
2 phase Optical Encoder wiring



Alps Dual Concentric
<http://www2.alps.co.jp/switchca/rotary/e-srbm.htm>

Title		
Rotary wiring		
Size A	Document Number	Rev
Date:	Tuesday, April 06, 1999	Sheet 1 of 1

2 phase mechanical



- DIGI-KEY 3315C-1-006
- DIGI-KEY 3315Y-1-006
- DIGI-KEY ECW1JB24BC0024

Title		
MECHANICAL ROTARIES		
Size	Document Number	Rev
A	<Doc>	
Date:	Tuesday, April 06, 1999	Sheet 1 of 1

