Lab 6: Peripheral Pin Select

ECE 3720

Preview

Peripheral Pin Select will be used to enable a second external interrupt. The two outputs of a rotary encoder will trigger the interrupts, causing a count variable, displayed on LEDs, to increase or decrease depending on the direction of rotation.

Peripheral Pin Select (PPS)

- Peripheral Pin Select allows the inputs and outputs of certain peripherals (such as timers or external interrupts) to be mapped to various external pins.
- Recall that in Lab 5 we only used the one interrupt already mapped to a pin.
- In this lab, you will use PPS to access an additional external interrupt.
- See section 11.3 (pg. 145) of the PIC32 datasheet for details on PPS.

TABLE 1-1: PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number(1)				Pin Type	Buffer Type	Description
	28-pin QFN	28-pin SSOP/ SPDIP/ SOIC	36-pin VTLA	44-pin QFN/ TQFP/ VTLA			
INT0	13	16	17	43	I	ST	External Interrupt 0
INT1	PPS	PPS	PPS	PPS	I	ST	External Interrupt 1
INT2	PPS	PPS	PPS	PPS	I	ST	External Interrupt 2
INT3	PPS	PPS	PPS	PPS	I	ST	External Interrupt 3
INT4	PPS	PPS	PPS	PPS	I	ST	External Interrupt 4

PIC32 datasheet, pg. 21

The Peripheral Pin Select (PPS) configuration provides an alternative to these choices by enabling peripheral set selection and their placement on a wide range of I/O pins. By increasing the pinout options available on a particular device, users can better tailor the device to their entire application, rather than trimming the application to fit the device.

PIC32 datasheet, pg. 145

Using PPS

- This process is explained in sections 11.3.4 and 11.3.5 of the datasheet (pgs. 145-148).
- Refer to tables 11-1 and 11-2. To map a peripheral's input/output to a pin, write the corresponding value from column 4 to the control register in column 3.

TABLE 11-1: INPUT PIN SELECTION

Peripheral Pin	*[pin name]*R SFR	*[pin name]*R bits	*[pin name]*R Value to RPn Pin Selection
INT4	INT4R	INT4R<3:0>	0000 = RPA0 0001 = RPB3 0010 = RPB4 0011 = RPB15 0100 = RPB7 0101 = RPC7(2) 0110 = RPC0(1) 0111 = RPC5(2) 1000 = Reserved . . . 1111 = Reserved
T2CK	T2CKR	T2CKR<3:0>	
IC4	IC4R	IC4R<3:0>	
$\overline{SS1}$	SS1R	SS1R<3:0>	
REFCLKI	REFCLKIR	REFCLKIR<3:0>	

PIC32 datasheet, pg. 146

These are just the first sections of each table. Look in the datasheet for the full tables.

TABLE 11-2: OUTPUT PIN SELECTION

RPn Port Pin	RPnR SFR	RPnR bits	RPnR Value to Peripheral Selection
RPA0	RPA0R	RPA0R<3:0>	0000 = No Connect 0001 = U1TX 0010 = U2RTS 0011 = SS1 0100 = Reserved 0101 = OC1 0110 = Reserved 0111 = C2OUT 1000 = Reserved : : 1111 = Reserved
RPB3	RPB3R	RPB3R<3:0>	
RPB4	RPB4R	RPB4R<3:0>	
RPB15	RPB15R	RPB15R<3:0>	
RPB7	RPB7R	RPB7R<3:0>	
RPC7	RPC7R	RPC7R<3:0>	
RPC0	RPC0R	RPC0R<3:0>	
RPC5	RPC5R	RPC5R<3:0>	

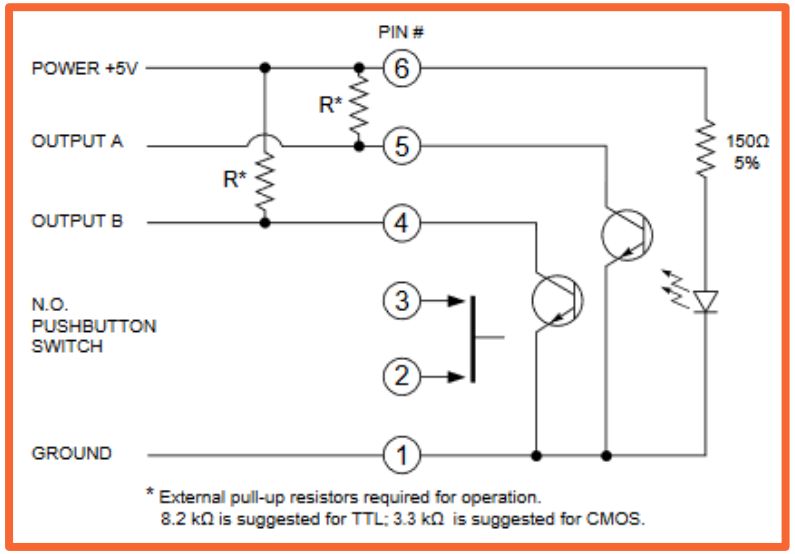
PIC32 datasheet, pg. 148

Example usage

```
T2CKR = 0b0110; // map Timer 2 ext input to C0
RPB3R = 0b0001; // map UART 1 transmit output to B3
```

Grayhill 61C Optical Encoder

- Datasheet available here
- Used to track rotational position
- Pay close attention to the diagram and table on page 2 of the datasheet.
  - The pull-up resistors are external to the encoder and must be supplied by the user.
  - Notice how the two outputs represent position. One of them changes with each tick of the dial.

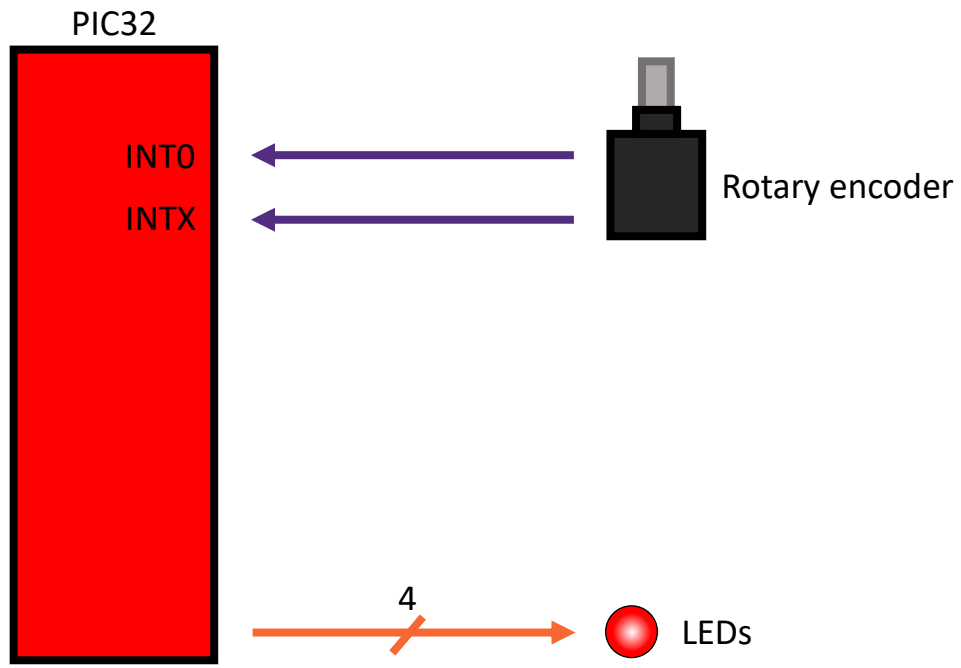


61C datasheet, pg. 2

Lab Goals

- Like Labs 1 and 5, a variable will count between 0 and 15 and be displayed on 4 LEDs.
  - The count should loop from 15 back to 0 and from 0 up to 15.
  - Unlike Labs 1 and 5, the count will not change automatically.
- Use PPS to enable a second interrupt.
  - Remember to pay attention to 5V tolerance
- Connect each output of the encoder to one interrupt.
- Increment the count on CW rotations and decrement on CCW rotations*.
  - Only modify the count in the ISRs
  - This means the count variable must be global
  - Do not save the previous state of the encoder

Simple Diagram



*Make sure to read the next slide for details on how to determine the direction of rotation.

Determining Direction of Rotation

- Pay close attention to the table from the 61C datasheet to determine whether to increment or decrement count.
- Each encoder output will have its own interrupt and ISR, which will be triggered any time the output changes.
  - Remember to clear the flag at the end of the ISR.
- Since the rotations cause both rising and falling edges, the interrupts' polarities will have to change accordingly.
  - e.g., if an interrupt was just triggered by a rising edge, it must next watch for a falling edge.
  - You will need to modify the interrupt's polarity from inside the ISR.
- For example, suppose the interrupt for output A is triggered:
  - You can read A and B to get the current position.
  - You know A just changed, so you can determine the previous position.
  - Then you have enough information to modify count and update the polarity.

Clockwise Rotation		
Position	Output A	Output B
1		
2	●	
3	●	●
4		●

● Indicates logic high; blank indicates logic low. Code repeats every 4 positions.

61C datasheet, pg. 2

This is an example of gray code, since successive values always differ by only one bit.