PPS usage – PIC24F

PPS Stands for Peripheral Pin Select. PPS is an advanced feature that allows to map a function on a pin. Only digital functions are remappable, I2C excluded. Functions are divided in Input Functions and Output Functions.

This document covers only pin/functions available on PIC24FJ suitable for ORbit16™ board: PIC24FJ32GB002, PIC24FJ64GB002.

On PIC24FJ remappable functions can be assigned to any remappable pin.

PIC24FJ calls remappable pins as RPy, where y is the number of remappable pin.

Not every pin on PIC24FJ is remappable, please consider the silkscreen on expansion headers of ORbit16™ : remappable pins have a “P” letter in the name: A0/P5 is RA0/RP5 pin, BP13 is RB13/RP13 pin and so on.

LOCK/UNLOCK PPS

First than remap a pin you must unlock the PPS registers with builtin macro:

```
__builtin_write_OSCCONL(OSCCON & ~(1<<6));
```

After you’ve remapped a pin, you must re-lock PPS registers with builtin macro:

```
__builtin_write_OSCCONL(OSCCON | (1<<6));
```

INPUT FUNCTIONS

	Function	Bits
INT1	External Interrupt 1	RPINR0bits.INT1R
INT2	External Interrupt 2	RPINR1bits.INT2R
T2CK	Timer 2 clock source	RPINR3bits.T2CKR
T3CK	Timer 3 clock source	RPINR3bits.T3CKR
T4CK	Timer 4 clock source	RPINR4bits.T4CKR
T5CK	Timer 5 clock source	RPINR4bits.T5CKR
IC1	Input capture 1	RPINR7bits.IC1R
IC2	Input capture 2	RPINR7bits.IC2R
IC3	Input capture 3	RPINR8bits.IC3R
IC4	Input capture 4	RPINR8bits.IC4R
IC5	Input capture 5	RPINR9bits.IC5R
OCFA	Output compare Fault A	RPINR11bits.OCFAR
OCFB	Output compare Fault B	RPINR11bits.OCAFBR
U1RX	UART1 Receive	RPINR18bits.U1RXR
U1CTS	UART1 Clear to send	RPINR18bits.U1CTSR
U2RX	UART2 Receive	RPINR19bits.U2RXR
U2CTS	UART2 Clear to send	RPINR19bits.U2CTSR
SDI1	SPI 1 Data Input	RPINR20bits.SDI1R
SCK1IN	SPI 1 Clock Input	RPINR20bits.SCK1R
SS1IN	SPI 1 Slave Select Input	RPINR21bits.SS1R
SDI1	SPI 2 Data Input	RPINR22bits.SDI2R
SCK2IN	SPI 2 Clock Input	RPINR22bits.SCK2R
SS2IN	SPI 2 Slave Select Input	RPINR23bits.SS2R

Example:

You want UART1RX on RP15 pin: `RPINR18bits.U1RXR=15;` *(15 is the remappable pin number: RP15)*

OUTPUT FUNCTIONS

	Function	Value
C1OUT	Comparator 1 Output	1
C2OUT	Comparator 2 Output	2
U1TX	UART 1 Transmit	3
U1RTS	UART 1 Request to send	4
U2TX	UART 2 Transmit	5
U2RTS	UART 2 Request to send	6
SDO1	SPI 1 Data Output	7
SCK1OUT	SPI 1 Clock Output	8
SS1OUT	SPI 1 Slave Select Output	9
SDO2	SPI 2 Data Output	10
SCK2OUT	SPI 2 Clock Output	11
SS2OUT	SPI2 Slave Select Output	12
OC1	Output Compare 1	18
OC2	Output Compare 2	19
OC3	Output Compare 3	20
OC4	Output Compare 4	21
OC5	Output Compare 5	22

Example:

For Pin RP(n) [where (n) is an even number] code is: RPOR(n/2)bits.RP(n)R = value
For Pin RP(n+1) [where (n+1) is an odd number] code is: RPOR(n/2)bits.RP(n+1)R = value

You want UART1TX on RP14 (n=14) : `RPOR7bits.RP14R=3;` *(3 is the value associated with UART1TX function)*

You want UART1TX on RP15 (n+1=15; n=14) : `RPOR7bits.RP15R=3;`

Complete example

```
__builtin_write_OSCCONL(OSCCON & ~(1<<6)); // unlock PPS registers
RPINR18bits.U1RXR=15; // set UART1 Receive on RP15 pin
RPOR7bits.RP14R=3; // set UART 1 Transmit on RP14 pin
__builtin_write_OSCCONL(OSCCON | (1<<6)); // re-lock PPS registers
```