MICROCHIP

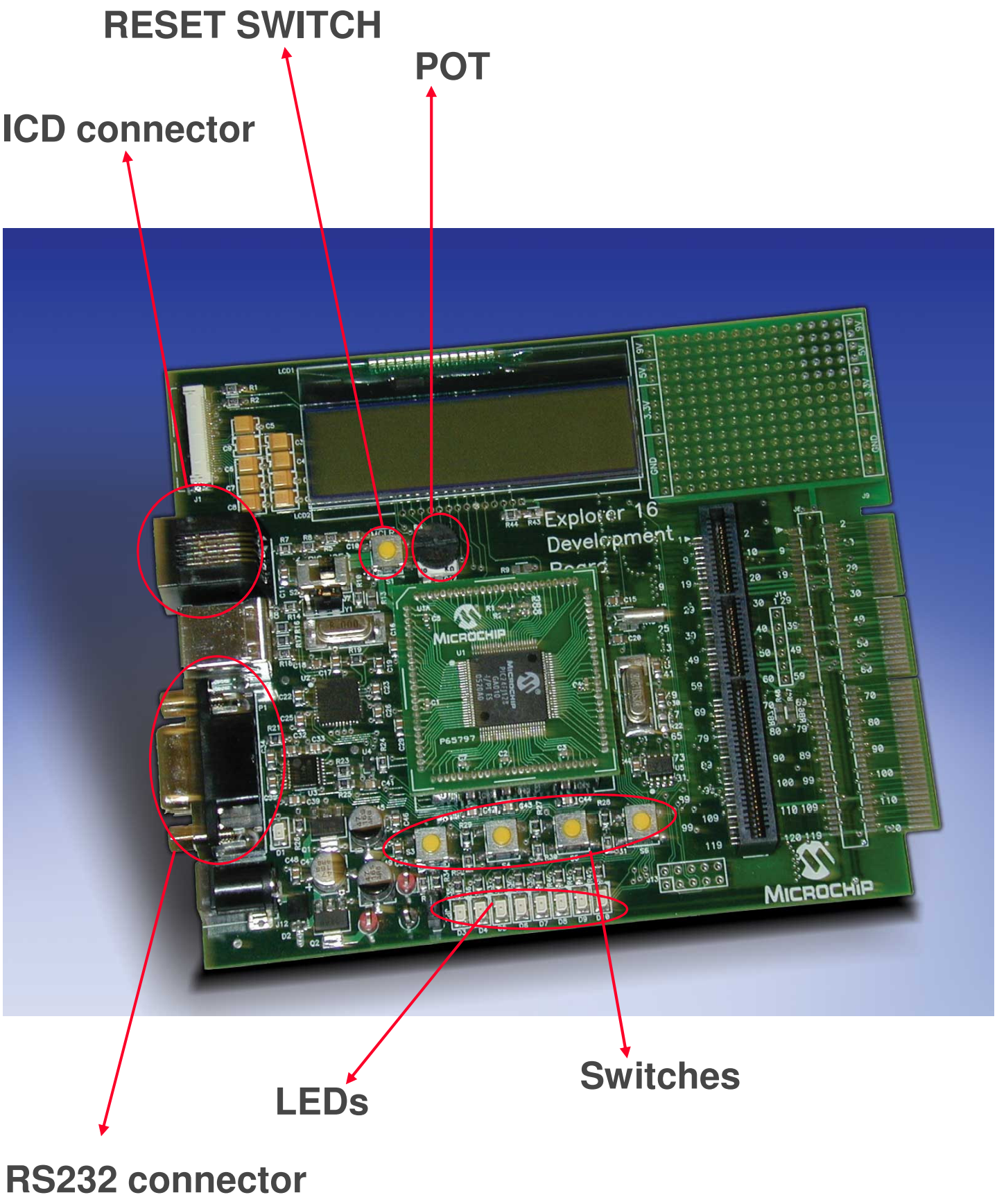
*Regional Training Centers*

# MCU 3122

## 16-bits Advanced Peripherals

MICROCHIP
*Regional Training Centers*

# EXPLORER 16



MICROCHIP
Regional Training Centers

# MPLAB Navigation

- **Quick ways to find functions or variables in MPLAB**
  - Source Locator
    - **To Enable**
      - Right-click on editor and go to “Properties…”
      - Check “Enable Source Locator”
      - On the Project window, click on the “Symbols” tab. Right click and check “Enable Tag Locators”
    - **Use this feature to quickly navigate through large applications**
      - Right-click on a function or variable in code and select “Goto Locator” to jump its definition
      - In the project window under the symbols tab, you can browse through and double click items to jump there in code
  - Edit->Find in Files (ctrl+shift+F)
    - **Use this to search all files in the project for a variable, function name, or anything else**

MICROCHIP

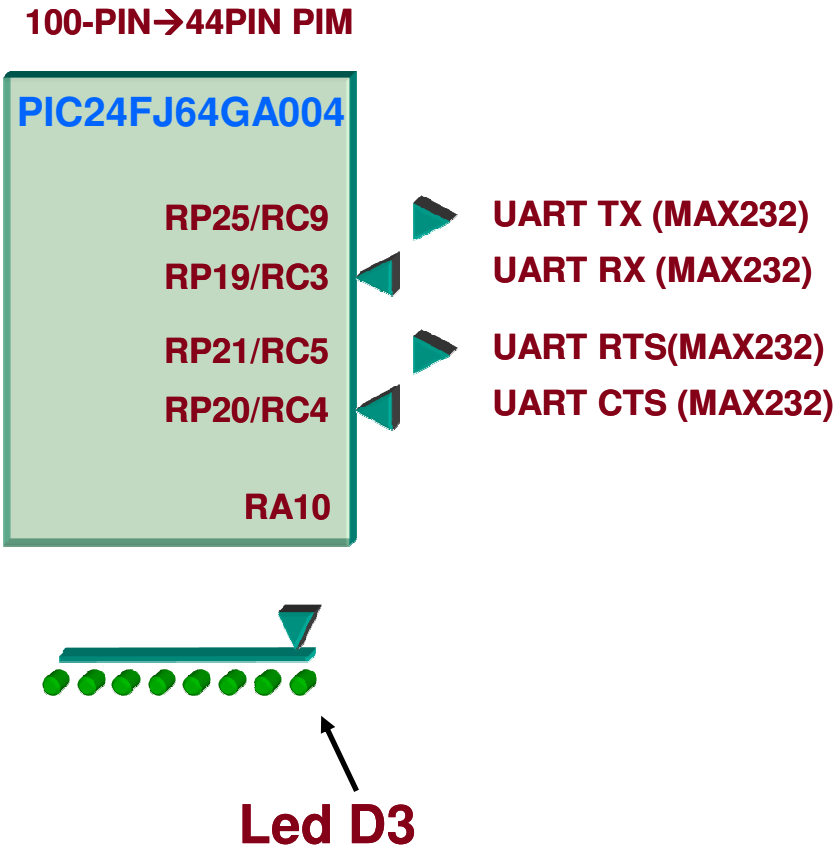
*Regional Training Centers*

# Lab 1
# Peripheral Pin Select (PPS)

MICROCHIP
Regional Training Centers

# Lab 1 Goals

- To configure the PPS module with UART2 using the PIC24FJ64GA004 PIM (processor module) mounted on the Explorer16 board
- To echo typed characters to HyperTerminal



MICROCHIP
Regional Training Centers

# Lab 1 To Do

- **Make sure to install/exchange (if not yet done) the PIC24FJ64GA004 PIM (processor module) on the EXPLORER 16 board**
- **Open the workspace**

  **C:\RTC\MCU3122\Lab1\Lab1-PPS.mcw**
- **Open the file "pps.c" and modify the following :**
  - STEP 1
    - **Set up the Configuration Bits using _CONFIG1 and _CONFIG2 directives :**
      - JTAG = Disabled
      - OSC = Primary – HS
      - WDT = Disabled
      - IOLOCK = 1 Way
      - EMUC/EMUD share PGC1/PGD1
  - STEP 2
    - **In function "`ioMap()`", issue the unlock sequence for the Reprogrammable Pin Mechanism (see function `lockIO()`)**
  - STEP 3-6
    - **In function "`ioMap()`":**
      - Map RP19 pin to input function U2RX
      - Map RP20 pin to input function U2CTS
      - Map Output Function U2TX to RP25
      - Map Output Function U2RTS to RP21
  - STEP 7
    - **In function "`ioMap()`", issue the lock sequence for the Reprogrammable Pin Mechanism (see function `lockIO()`)**
- **Open HyperTerminal file "PPS.ht" (change COM port # if necessary according to your PC.**
  **Setup : 9600 / 8 / N / 1 / Hardware Flow Control**

MICROCHIP
Regional Training Centers

# Lab 1
# RPINRx

TABLE 9-1: SELECTABLE INPUT SOURCES (MAPS INPUT TO FUNCTION)(1)

| Input Name | Function Name | Register | Configuration Bits |
|---|---|---|---|
| External Interrupt 1 | INT1 | RPINR0 | INT1R[4:0] |
| External Interrupt 2 | INT2 | RPINR1 | INT2R[4:0] |
| Timer 2 External Clock | T2CK | RPINR3 | T2CKR[4:0] |
| Timer 3 External Clock | T3CK | RPINR3 | T3CKR[4:0] |
| Timer 4 External Clock | T4CK | RPINR4 | T4CKR[4:0] |
| Timer 5 External Clock | T5CK | RPINR4 | T5CKR[4:0] |
| Input Capture 1 | IC1 | RPINR7 | IC1R[4:0] |
| Input Capture 2 | IC2 | RPINR7 | IC2R[4:0] |
| Input Capture 3 | IC3 | RPINR8 | IC3R[4:0] |
| Input Capture 4 | IC4 | RPINR8 | IC4R[4:0] |
| Input Capture 5 | IC5 | RPINR9 | IC5R[4:0] |
| Output Compare Fault A | OCFA | RPINR11 | OCFAR[4:0] |
| Output Compare Fault B | OCFB | RPINR11 | OCFBR[4:0] |
| UART 1 Receive | U1RX | RPINR18 | U1RXR[4:0] |
| UART 1 Clear To Send | U1CTS | RPINR18 | U1CTSR[4:0] |
| UART 2 Receive | U2RX | RPINR19 | U2RXR[4:0] |
| UART 2 Clear To Send | U2CTS | RPINR19 | U2CTSR[4:0] |
| SPI 1 Data Input | SDI1 | RPINR20 | SDI1R[4:0] |
| SPI 1 Clock Input | SCK1IN | RPINR20 | SCK1R[4:0] |
| SPI 1 Slave Select Input | SS1IN | RPINR21 | SS1R[4:0] |
| SPI 2 Data Input | SDI2 | RPINR22 | SDI2R[4:0] |
| SPI 2 Clock Input | SCK2IN | RPINR22 | SCK2R[4:0] |
| SPI 2 Slave Select Input | SS2IN | RPINR23 | SS2R[4:0] |

**Note 1:** Unless otherwise noted, all inputs use the Schmitt input buffers

**Refer to PIC24FJ64GA004 Datasheet Section 9.4, page 100**

MICROCHIP
Regional Training Centers

# Lab 1
# Peripheral Output Function Numbers

TABLE 9-2: SELECTABLE OUTPUT SOURCES (MAPS FUNCTION TO OUTPUT)

| Function | Output Function Number(1) | Output Name |
|---|---|---|
| NULL(2) | 0 | NULL |
| C1OUT | 1 | Comparator 1 Output |
| C2OUT | 2 | Comparator 2 Output |
| U1TX | 3 | UART 1 Transmit |
| U1RTS | 4 | UART 1 Ready To Send |
| U2TX | 5 | UART 2 Transmit |
| U2RTS | 6 | UART 2 Ready To Send |
| SDO1 | 7 | SPI 1 Data Output |
| SCK1OUT | 8 | SPI 1 Clock Output |
| SS1OUT | 9 | SPI 1 Slave Select Output |
| SDO2 | 10 | SPI 2 Data Output |
| SCK2OUT | 11 | SPI 2 Clock Output |
| SS2OUT | 12 | SPI 2 Slave Select Output |
| OC1 | 18 | Output Compare 1 |
| OC2 | 19 | Output Compare 2 |
| OC3 | 20 | Output Compare 3 |
| OC4 | 21 | Output Compare 4 |
| OC5 | 22 | Output Compare 5 |

**Note 1:** Value assigned to the RPn[4:0] registers corresponds to the peripheral output function number.

**2:** The NULL function is assigned to all RPn outputs at device Reset and disables the RPn output function.

**Refer to PIC24FJ64GA004 Datasheet Section 9.4, page 101**

MICROCHIP
Regional Training Centers

# Lab 1 Expected Result

- Typed keys on the PC keyboard are echoed back to the terminal software (Hyperterminal or other)
- LED D3 (most right LED) on EXPLORER16 alternately switched ON and OFF for every key pushed

MICROCHIP

*Regional Training Centers*

# Lab 2
# Parallel Master Port (PMP)

MICROCHIP
Regional Training Centers

# Lab 2 Goals

- To configure the internal PMP module using the PIC24FJ128GA010 on the Explorer16 board
- To understand the signals required to interface Microchip 16 bits devices with parallel bus driven LCD modules
- To display a string on the LCD display

MICROCHIP
Regional Training Centers

# Lab 2 To Do

- **Open the workspace**
  **C:\RTC\MCU3122\Lab2\Lab2-PMP.mcw**

- **In lcd.c**
  - STEP 1
    - Configure PMCON (8 bits data) : PMP enabled, address/data not multiplexed, PMBE disabled, PMENB enabled, RD/~WR enabled, no CS, PMENB active high, PMRD/~PMWR active high
  - STEP 2
    - Configure PMMODE: No interrupt (polling Busy flag), inc/dec off, 8 bit mode, combined read/write with byte enable signals, assume maximum number of wait states for beginning, middle and final phase of PMENB strobe.
    - N.B. : in actual systems, the number of wait states need to be optimised depending upon the LCD electrical specification and the MCU operating frequency.

# Lab 2 To Do

- STEP 3
  - Configure PMAEN: Enable only A0 to control RS and disable all other PMP address pins and CS.
- STEP 4
  - Configure PMADDR: A0 selects type of instruction, either command or data.
  - This is a command so A0 should be low.

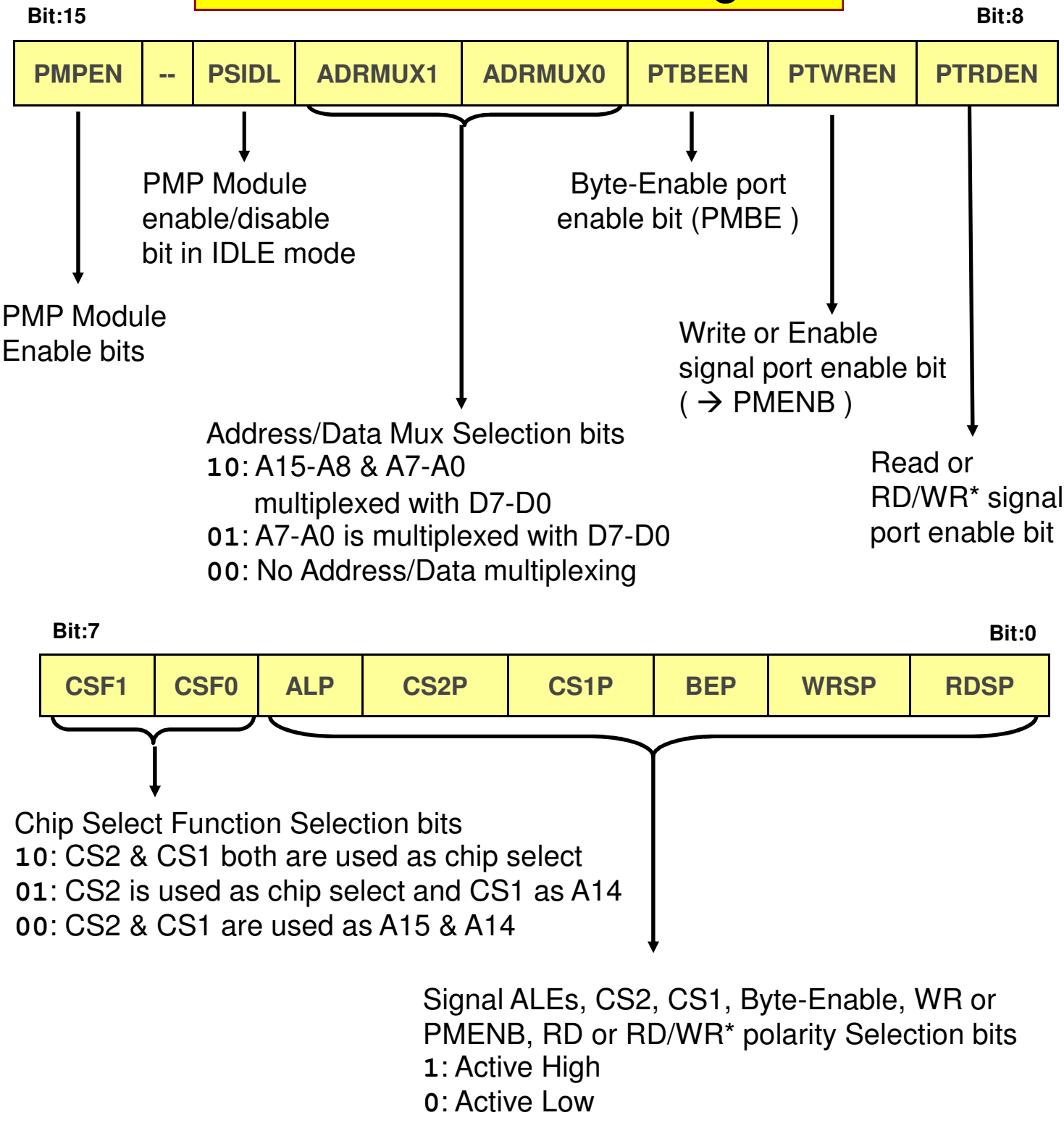
- **In main.c**
  - STEP 5:
    - Change text to enter your name

- **Extra Credit for advanced users**
  - Modify code provided to display the rotating banners in my_banner
  - Useful variables and functions: pban, num_banners, wait_time, mLCDPutChar(char), and mLCDClear()
  - Refer to comments in code for explanation of functions

MICROCHIP
Regional Training Centers

# Lab 2
# PMP Registers

## PMCON: PMP Control Register




**Refer to PIC24FJ128GA010 Data Sheet Section 17, page 162**

MICROCHIP
Regional Training Centers

# Lab 2
# PMP Registers

## PMMODE: PMP Mode Selection Register

Bit:15 Bit:8

| BUSY | IRQM1 | IRQM0 | INCM1 | INCM0 | MODE16 | MODE1 | MODE0 |
|---|---|---|---|---|---|---|---|

PMP Module Status bits

Interrupt Mode Selection bits
11: Interrupt is generated when buffer is full
01: Interrupt is generated at the end of Read/Write cycle
00: No Interrupt is generated

Address inc/dec Selec. bits
11: Buffer mode
10: Auto address decrement
01: Auto address increment
00: No Address inc/dec

Data Bus Width Selection bit
1: 16-bit
0: 8-bit

Mode Selection bits
11: Master Mode with controls, CSs,RD,WR,ENB,BE,ALEs
10: Master Mode with controls, CSs, RD,WR,BE,ALEs
01: Enhanced Slave Mode
00: Legacy Slave Mode

Bit:7 Bit:0

| WAITB1 | WAITB0 | WAITM3 | WAITM2 | WAITM1 | WAITM0 | WAITE1 | WAITE0 |
|---|---|---|---|---|---|---|---|

Begining phase Wait Selection bits
11: 4 Tcycles
10: 3 Tcycles
01: 2 Tcycles
00: 1 Tcycle

Middle phase Wait Selection bits
1111: 15 Tcycles
1110: 14 Tcycles
..
0001: 1 Tcycle
0000: No Wait cycles

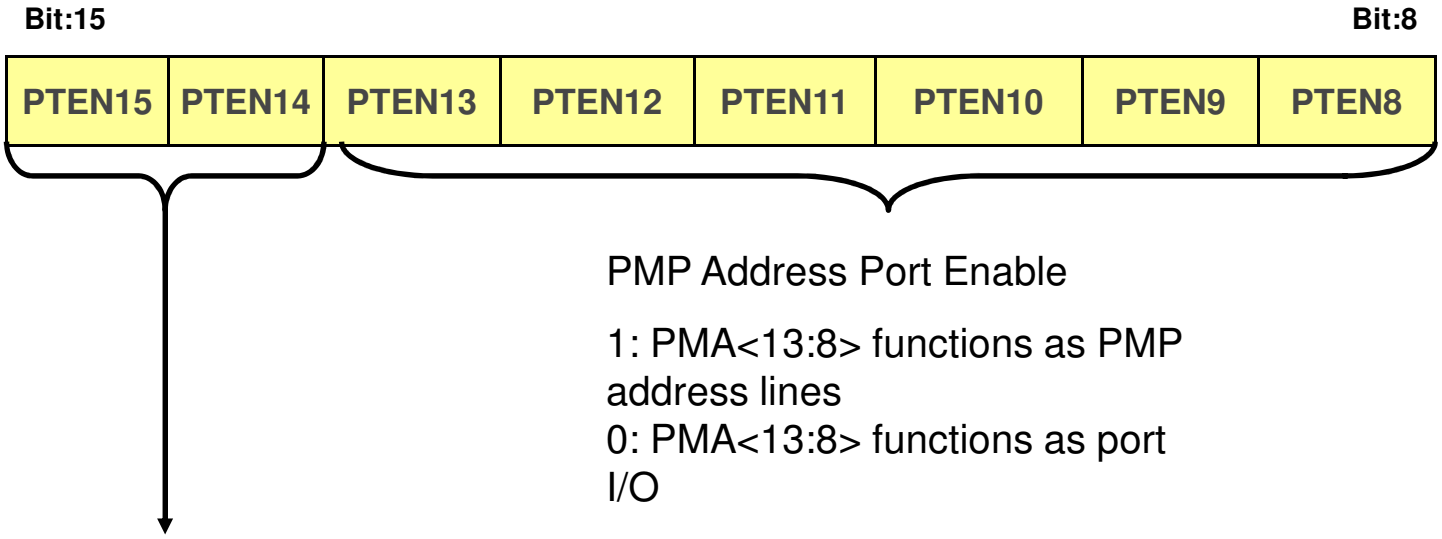
Final phase Wait Selection bits
11: 4 Tcycles
10: 3 Tcycles
01: 2 Tcycles
00: 1 Tcycle

**Refer to PIC24FJ128GA010 Data Sheet Section 17, page 164**

MICROCHIP
Regional Training Centers

# Lab 2
# PMP Registers

## PMAEN: ADDRESS ENABLE REGISTER



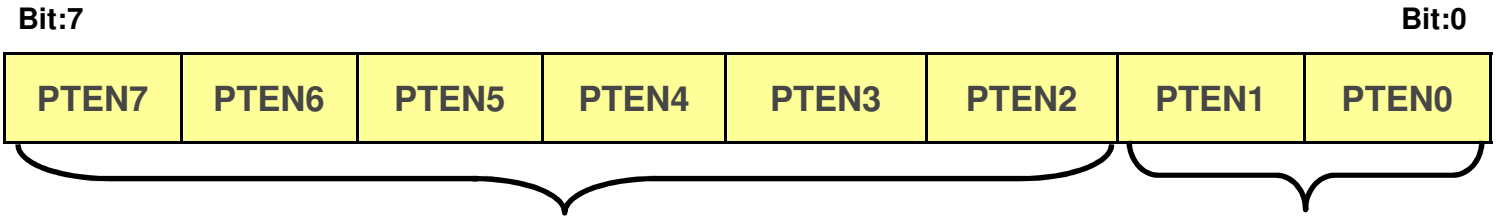

PMP Address Port Enable

1: PMA<13:8> functions as PMP address lines
0: PMA<13:8> functions as port I/O

PMP Chip Select Strobe Enable

1: PMA15 and PMA14 function as either PMA<15:14> or as PMCS1/2

0: PMA<15:14> functions as port I/O




PMP Address Port Enable

1: PMA<7:2> functions as PMP address lines
0: PMA<7:2> functions as port I/O

PMALH/PMALL Strobe Enable

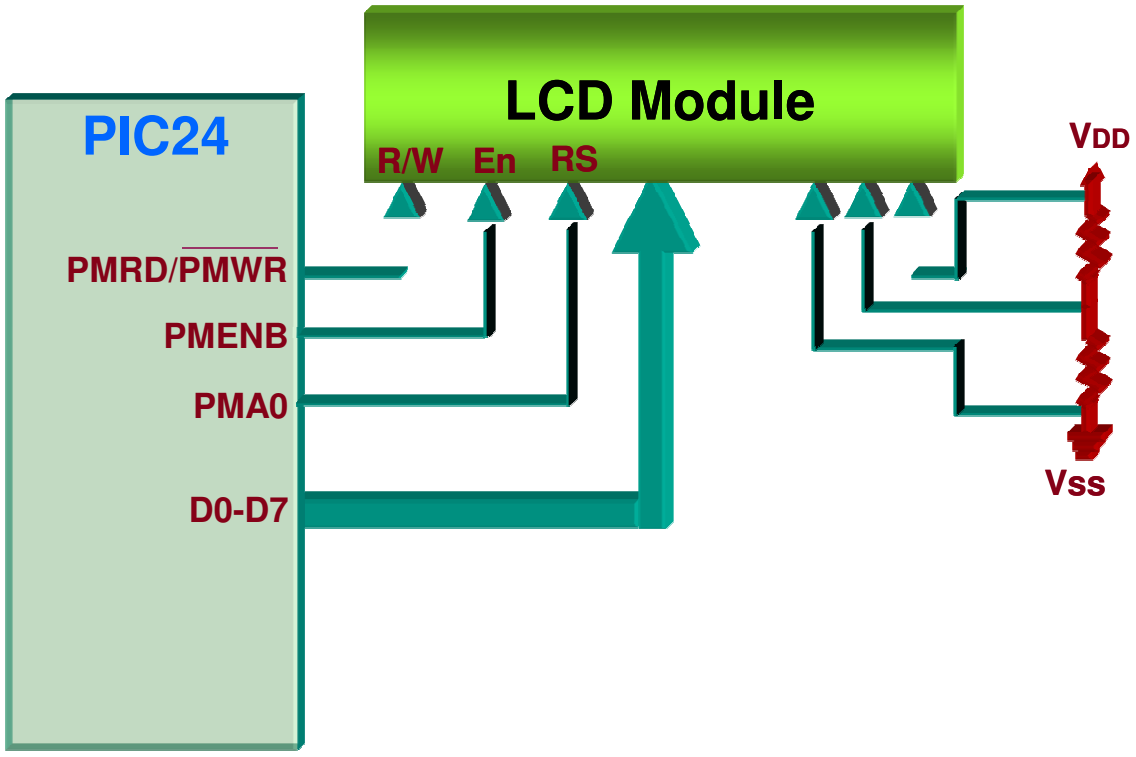
1: PMA1 and PMA0 pads function as either PMA<1:0> or PMALH/L
0: PMA1 and PMA0 pads function as port I/O

**Refer to PIC24FJ128GA010 Data Sheet Section 17, page 165**

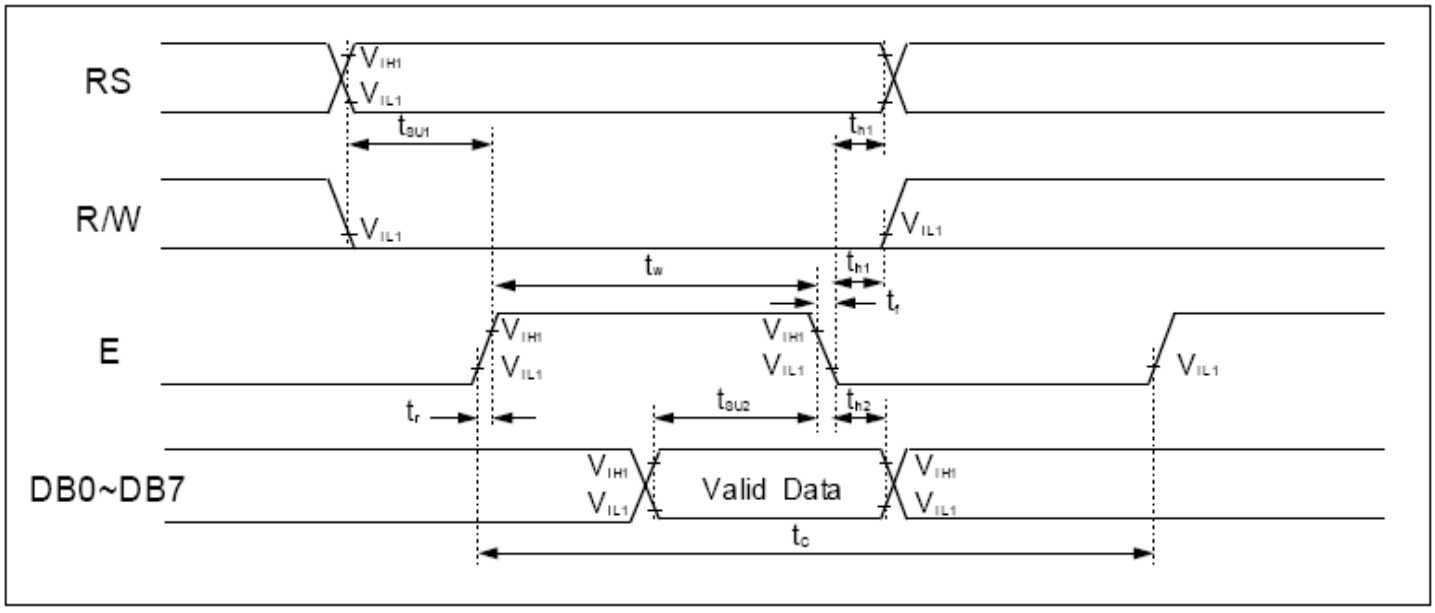
MICROCHIP
Regional Training Centers

# Lab 2 Operation

- PMP to LCD Connections




- LCD write timing



MICROCHIP
Regional Training Centers

# Lab 2 Expected Result

- Name is displayed on the LCD

- For extra credit: Rotate banner displayed on LCD once approximately every 2 seconds.

MICROCHIP

*Regional Training Centers*

# LAB 3
# Real Time Clock and Calendar (RTCC)

MICROCHIP
Regional Training Centers

# Lab 3 Goals

- Configure the internal RTCC module using the PIC24FJ128GA010 on the Explorer16 board

- Set the RTCC Time and Alarm

MICROCHIP
Regional Training Centers

# Lab 3 To Do

- **Open the workspace** C:\RTC\MCU3122\Lab3\RTCC.mcw

- **In rtcc.c**
  - STEP 1:
    - **Unlock RTCC registers**
      - Hint: look for mRTCCUnlock macro
  - STEP 2:
    - **Configure RCFGCAL, RTCPTR Auto-decrementing pointer**
  - STEP 3:
    - **Write Year To RTCVAL**
    - **Write Month & Day To RTCVAL**
    - **Write Weekday & Hour To RTCVAL**
    - **Write Minutes & Seconds To RTCVAL**

MICROCHIP
Regional Training Centers

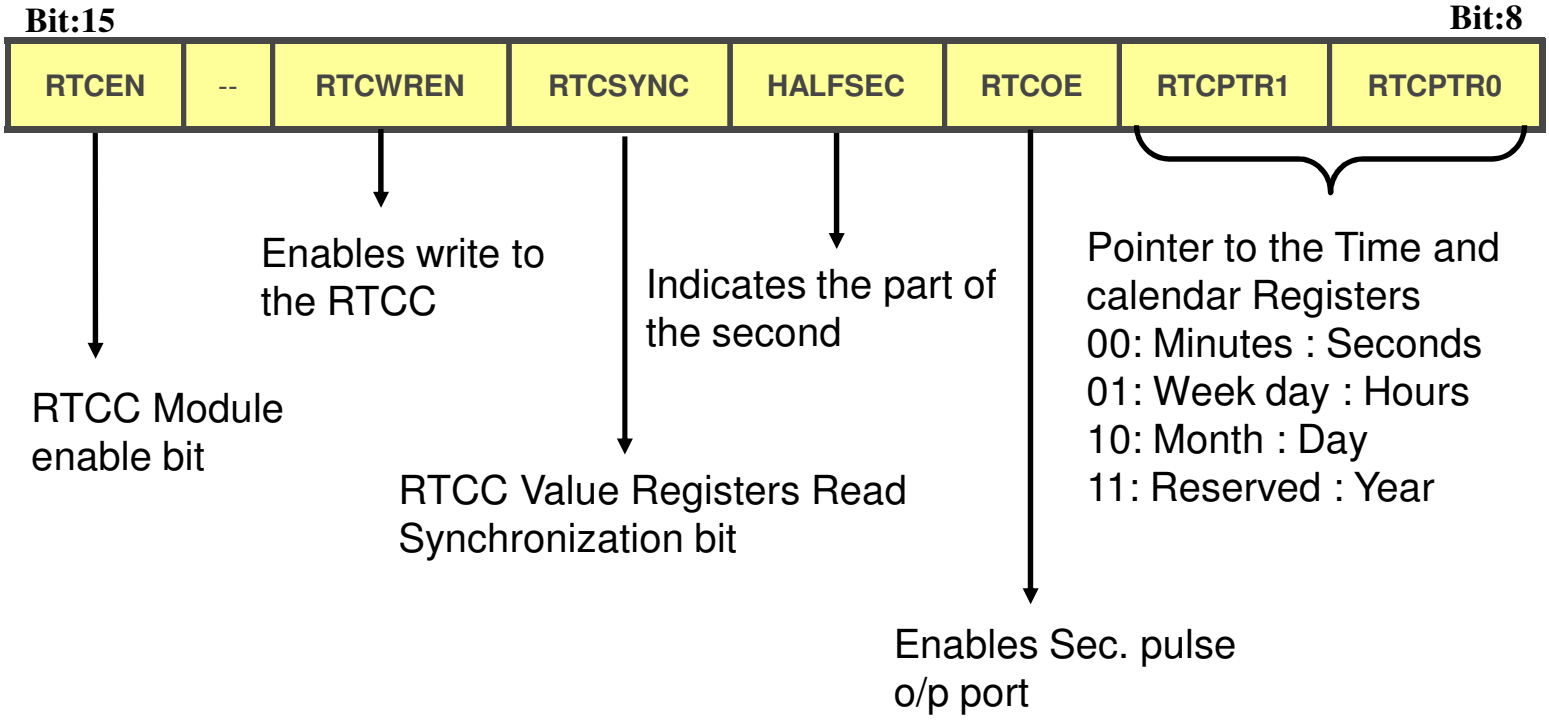
# Lab 3 To Do

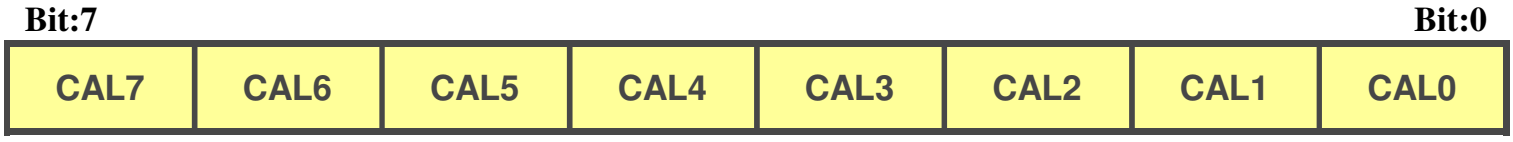
- STEP 4:
  - **Enable RTCC**
- STEP 5:
  - **Lock RTCC registers**
    - Hint: look for mRTCCLock macro
- STEP 6:
  - **In ALCFGRPT, configure alarm frequency @ 0.1Hz (every 10s)**
  - **In ALCFGRPT, configure alarm to repeat 3 times**
- STEP 7:
  - **Enable alarm**

MICROCHIP
Regional Training Centers

# Lab 3
# RTCC Registers

## RCFGCAL: RTCC calibration and configuration register



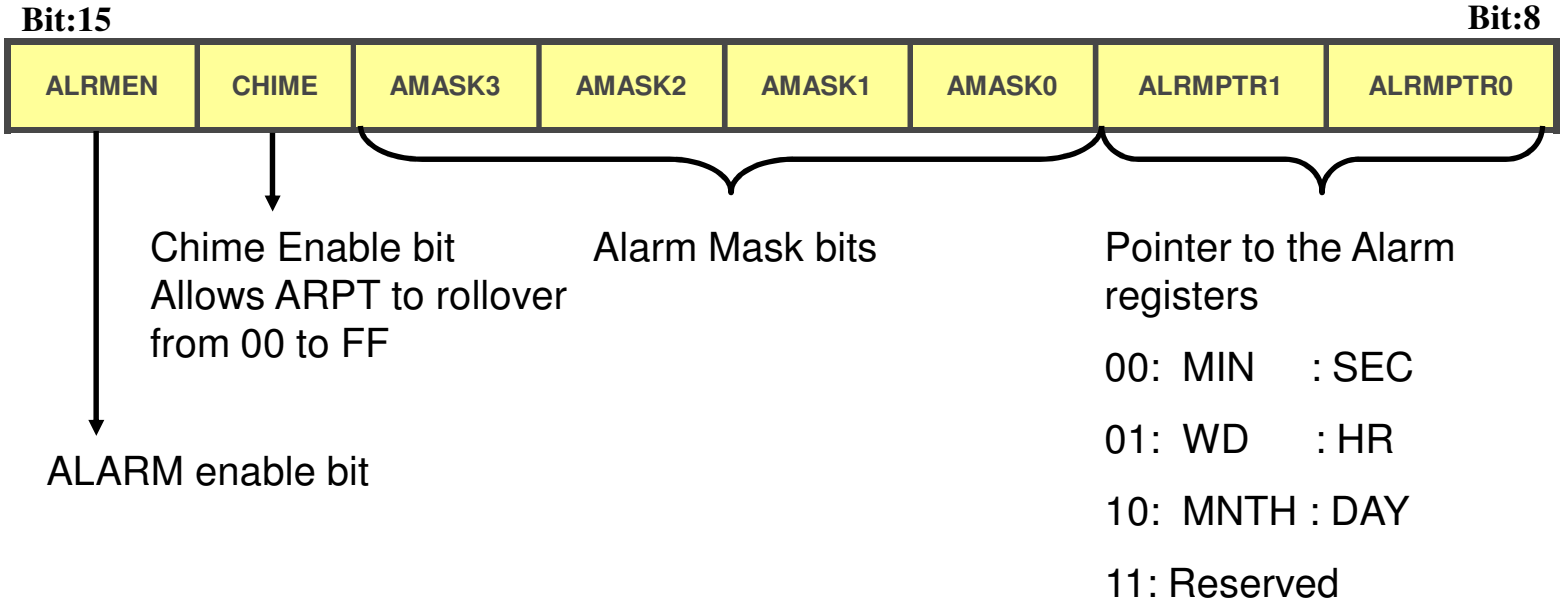

Crystal offset calibration bits (RTCC Drift calibration bits)

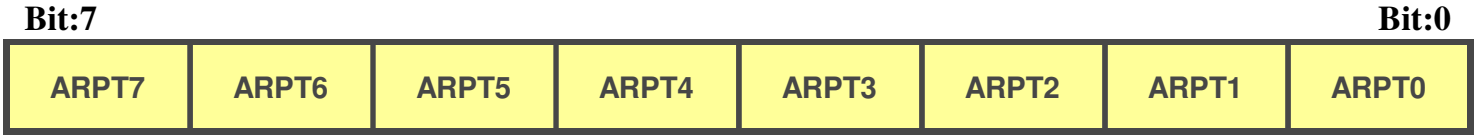
**Refer to PIC24FJ128GA010 Data Sheet Section 18, page 173**

MICROCHIP
Regional Training Centers

# Lab 3
# RTCC Registers

## ALCFGRPT: RTCC Alarm configuration register




Alarm Repeat Counter Value bits (Repeat count = $2^n$)

**Refer to PIC24FJ128GA010 Data Sheet Section 18, page 175**

MICROCHIP
Regional Training Centers

# Lab 3
# RTCC Registers

## RTCVAL: RTCC value register

- RTCPTR<1:0> auto decrements when RTCVAL<15:8> is read or written until it reaches '00'

| RTCPTR<1:0> | RTCVAL<15:8> | RTCVAL<7:0> |
|---|---|---|
| 11 | --- | YEAR |
| 10 | MONTH | DAY |
| 01 | WEEKDAY | HOURS |
| 00 | MINUTES | SECONDS |

## ALRMVAL: RTCC Alarm value register

- ALRMPTR<1:0> auto decrements when ALRMVAL<15:8> is read or written until it reaches '00'

| ALRMPTR<1:0> | ALRMVAL<15:8> | ALRMVAL<7:0> |
|---|---|---|
| 11 | --- | --- |
| 10 | ALRMMNTH | ALRMDAY |
| 01 | ALRMWD | ALRMHR |
| 00 | ALRMMIN | ALRMSEC |

Refer to PIC24FJ128GA010 Data Sheet Section 18, page 176-179

MICROCHIP
Regional Training Centers

# Lab 3 Expected Results

- The time and date will be displayed on the LCD.
- An LED should blink once every 10 seconds for 3 blinks when the RTCC seconds value equals Alarm seconds value (5).

MICROCHIP

*Regional Training Centers*

# LAB 4
# Cyclic Redundancy Check Generator (CRC)

MICROCHIP
Regional Training Centers

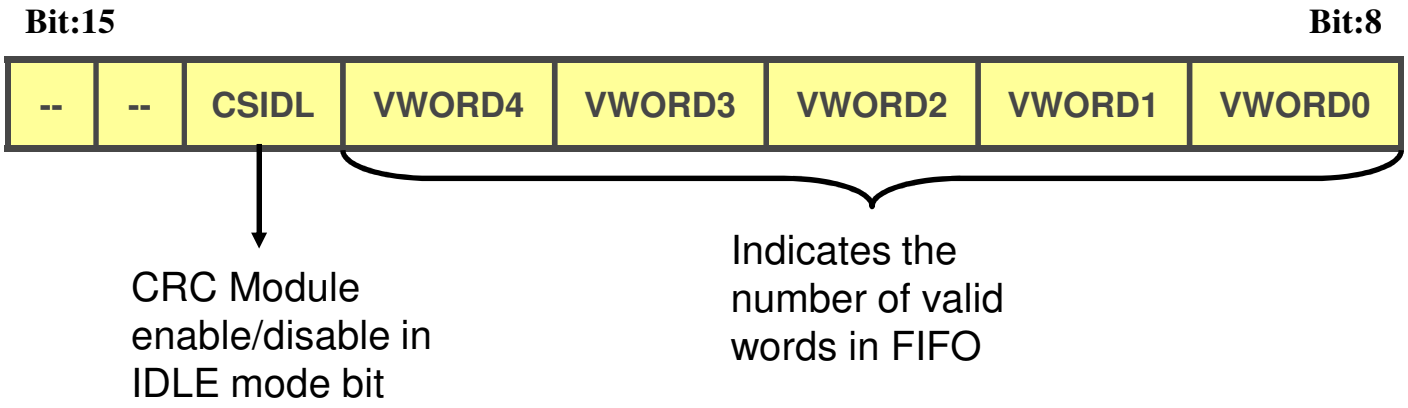
# Lab 4 Goals

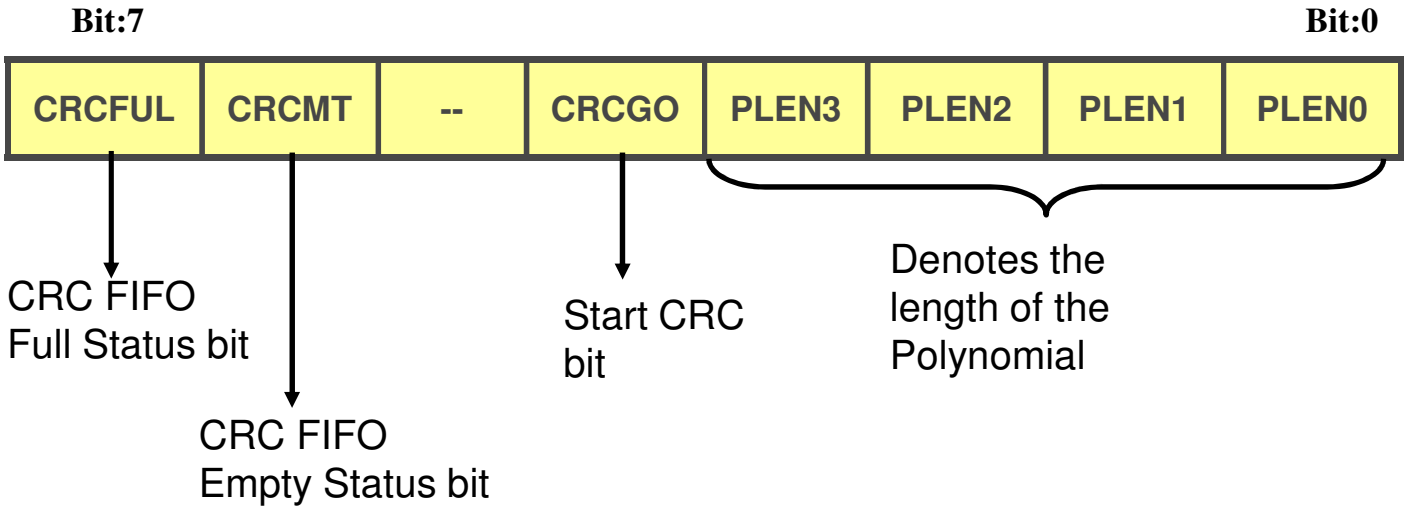
- Understand the configuration of CRC module using the PIC24FJ128GA010 on the Explorer16 board
- Understand CRC operation
- Find the CRC Result of a data transmission

MICROCHIP
Regional Training Centers

# Lab 4 CRC Registers

## CRCCON: CRC Control register




Refer to PIC24FJ128GA010 Data Sheet Section 19, page 181

MICROCHIP
Regional Training Centers

# Lab 4 To Do

- **Open the workspace**
  C:\RTC\MCU3122\Lab4\CRC.mcw

- **In main.c:**
  - STEP 1:
    - **In CRCCON, Configure The Polynomial Length (PLEN) for the Polynomial:**
      - x^16 + x^15 + x^2 + 1
  - STEP 2:
    - **In CRCXOR, Configure for the Polynomial x^16 + x^15 + x^2 + 1**
  - STEP 3:
    - **Clear CRCWDAT**
  - STEP 4:
    - **In CRCCON, enable the CRC Generator**

MICROCHIP
Regional Training Centers

# Lab 4 To Do

- Step 5 – Open Needed Files & Programs:
  - **Open HyperTerminal with CRC.ht profile (9600,N-8-1, no flow control), in the Lab4 directory**
  - **Open CRC spreadsheet, CRCCalc.xls, in the Lab4 directory**
    - If there are errors, go to Tools->Add-Ins and check “Analysis Toolpack” and “Analysis Toolpack – VBA”
  - **Open CRC.txt in the Lab4 directory**

- Step 6 – Calculate a known good CRC value:
  - **Enter 10 words of data in the CRC spreadsheet in blue cells A4 to A13**
  - **Copy the green cell C13 into The CRC.txt file, This is your data message and CRC checksum**

MICROCHIP
Regional Training Centers

# Lab 4 To Do

- Step 7 – Transmit Data message + CRC value
  - **Compile and run the code**
  - **Send the data with HyperTerminal using copy then right click -> “Paste to host” or Transfer -> “Send text file…”**
  - **Check the LCD display and verify that “CRC verified OK” is displayed**

- Step 8 – Corrupt the data message
  - **Change any value in the text file to corrupt the message**
  - **Send the data with HyperTerminal using copy then right click -> “Paste to host” or Transfer -> “Send text file…”**
  - **Check the LCD display and verify that “CRC verified BAD” is displayed. This indicates that the CRC verification failed.**

MICROCHIP
Regional Training Centers

# Lab 4 Expected Results

- With a correct data transmission the LCD displays :

  “CRC Verified OK”

- With a corrupted data transmission the LCD displays

  “CRC Verified BAD”

- Try both

MICROCHIP

*Regional Training Centers*