



University of San Carlos | Department of
COMPUTER ENGINEERING

CpE 3201
Embedded Systems

PIC16F877A I/O Port Configuration

General Purpose I/O

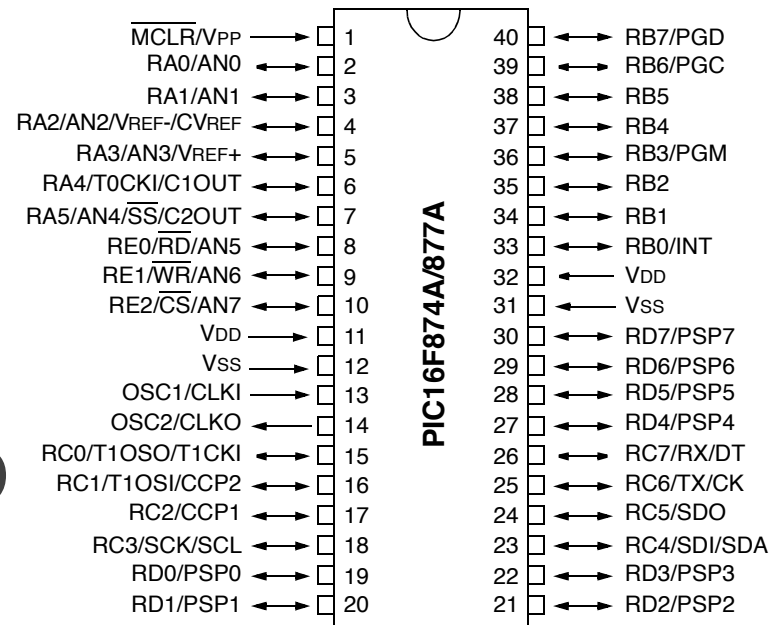
- The PIC16F877A has 33 GPIO ports divided into 5 port groups:
 - PORTA (6)
 - PORTB (8)
 - PORTC (8)
 - PORTD (8)
 - PORTE (3)

General Purpose I/O

- Some I/O are also multiplexed to other functions such as:
 - analog input
 - external clock source
 - external interrupt source
 - serial communication lines (USART, I2C and SPI)

IO Port Convention

- The individual I/O ports in a port group is also denoted by the name **Rxn** where x is the port group letter and n is the bit position.
- For example, RA0 is an I/O port which belongs in PORTA and it represent the LSB of the port group.



Port Direction Control

- Each I/O port can be programmed as either input or output when used as a general purpose I/O.
- The port direction is controlled by the following registers: TRISA, TRISB, TRISC, TRISD and TRISE.
- Each bit in these registers corresponds to the individual I/O ports in the group.

Port Direction Control

- Each I/O port can be programmed as either input or output when used as a general purpose I/O. To program the individual I/O ports in the group, set the bits of the TRISx register to the following:
 - TRISx bit \leq '0' - function as an output pin
 - '1' - function as an input pin

Port Direction Control

- For example, if all ports of PORTB will be input, then the value of the latter will be 00000000_2 or $0x00$. If RB0 is set to input and the rest are output then the value of TRISB will be 00000001_2 or $0x01$.

	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
TRISB	0	0	0	0	0	0	0	1
PORTB (direction)	o	o	o	o	o	o	o	i

Port Programming in C

The following is an example code to program the I/O port using C and writing and reading to/from an I/O port.

```
void main()
{
    unsigned char data;
    OPTION_REG = 0xC0; // configuring the OPTION register
    TRISA = 0x0F;      // set RA0-RA3 as input, the rest are output
    TRISB = 0x00;      // set all of PORTB as output
    for(;;)
    {
        PORTB = 0x00; // set all ports in PORTB is '0'
        data = PORTA & 0x0F; // read data in PORTA and mask
        ...
        ...
        ...
    }
}
```


Port Programming in C

- You can also access an individual port and perform write/read operation without having to configure the entire port register.
- For example, to set only RB0 to '1', you can write `RB0 = 1` in your code. On the other hand, you can read an individual I/O port for example; `if (!RA0)`, in this case it is evaluating the value of RA0.

PORTA

- PORTA is a 6-bit wide, bidirectional port. The corresponding data direction register is TRISA.
- Pin RA4 is multiplexed with the Timer0 module clock input to become the RA4/T0CKI pin. The RA4/T0CKI pin is a Schmitt Trigger input and an open-drain output. All other PORTA pins have TTL input levels and full CMOS output drivers.

PORTB

- PORTB is an 8-bit wide, bidirectional port. The corresponding data direction register is TRISB.
- Three pins of PORTB are multiplexed with the In-Circuit Debugger and Low-Voltage Programming function: RB3/PGM, RB6/PGC and RB7/PGD.
- Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is performed by clearing bit **RBPU** (OPTION_REG<7>).

PORTC

- PORTC is an 8-bit wide, bidirectional port. The corresponding data direction register is TRISC.
- PORTC is multiplexed with several peripheral functions:

TABLE 4-5: PORTC FUNCTIONS

Name	Bit#	Buffer Type	Function
RC0/T1OSO/T1CKI	bit 0	ST	Input/output port pin or Timer1 oscillator output/Timer1 clock input.
RC1/T1OSI/CCP2	bit 1	ST	Input/output port pin or Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output.
RC2/CCP1	bit 2	ST	Input/output port pin or Capture1 input/Compare1 output/PWM1 output.
RC3/SCK/SCL	bit 3	ST	RC3 can also be the synchronous serial clock for both SPI and I ² C modes.
RC4/SDI/SDA	bit 4	ST	RC4 can also be the SPI data in (SPI mode) or data I/O (I ² C mode).
RC5/SDO	bit 5	ST	Input/output port pin or Synchronous Serial Port data output.
RC6/TX/CK	bit 6	ST	Input/output port pin or USART asynchronous transmit or synchronous clock.
RC7/RX/DT	bit 7	ST	Input/output port pin or USART asynchronous receive or synchronous data.

Legend: ST = Schmitt Trigger input

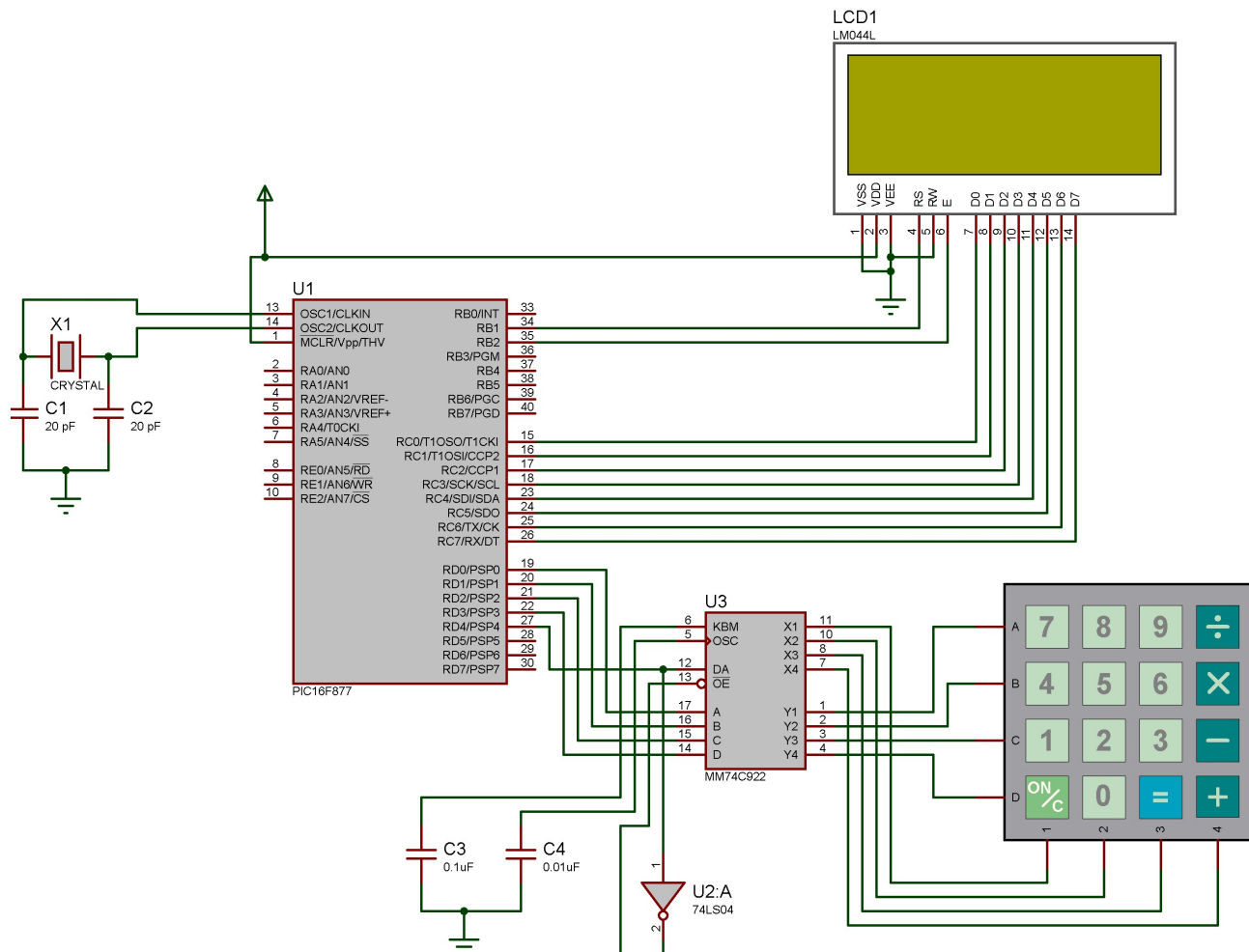
PORTD

- PORTD is an 8-bit port with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.
- PORTD can be configured as an 8-bit wide microprocessor port (Parallel Slave Port) by setting control bit, **PSPMODE** (TRISE<4>). In this mode, the input buffers are TTL.

PORTE

- PORTE has three pins (RE0/ $\overline{\text{RD}}$ /AN5, RE1/ $\overline{\text{WR}}$ /AN6 and RE2/ $\overline{\text{CS}}$ /AN7) which are individually configurable as inputs or outputs. These pins have Schmitt Trigger input buffers.
- The PORTE pins become the I/O control inputs for the microprocessor port when bit **PSPMODE** (TRISE<4>) is set.
- Both PORTD (data) and PORTE (control) are used in parallel-slave configuration of the microcontroller.

I/O Interfacing



I/O Interfacing

- For more information about interfacing LCD and numeric keypad, you can refer to the following Canvas pages:
 - [Interfacing Displaytech 204A LCD](#)
 - [Interfacing Numeric Keypad](#)
 - [Interfacing Stepper Motor](#)



University of San Carlos | Department of
COMPUTER ENGINEERING

CpE 3201
Embedded Systems

End of Lecture

References:

- PIC16F87X Data Sheet, Microchip Technology Inc. 2003.