LCD Instructions

- lcd_init()   Must be called before any other function
- lcd_putc(c) Will display c on the next position of the LCD.
- The following have special meaning:
- \f Clear display
- \n Go to start of second line
- \b Move back one position
- lcd_gotoxy(x,y) Set write position on LCD (upper left is 1,1)
- Printf(lcd_putc,”%s”,x);

  %s ➔ means the type of x variable is string.(ffloat....etc)

  X is the variable you want to print on LCD.

Example Print Text on LCD

```
#include <16f877a.h>
#fuses xt,nowdt
#use delay(clock=4000000)
#include <lcd.c>
void main(){
char text[]="BME366";
lcd_init();
printf(lcd_putc,"%s",text);}
```

A/D

- setup_adc_ports (***value***)

- setup_adc_ports(ALL_ANALOG); // All pins analog.

- setup_adc_ports(RA0_RA1_RA3_ANALOG);

// Pins A0, A1 and A3 are analog and all others

// are digital. The +5v is used as a reference.

- □ setup_adc (***mode***);
- □ ***mode***– Analog to digital mode. The valid options vary depending on the device. See the devices .h file for all options. Some typical options include:
  - · ADC_OFF
  - · ADC_CLOCK_INTERNAL
  - · ADC_CLOCK_DIV_32
- □ setup_adc(ADC_CLOCK_INTERNAL);

- set_adc_channel (***chan***)
- ***chan*** is the channel number to select. Channel numbers start at 0 and are labeled in the data sheet AN0, AN1
- Examples:

```
set_adc_channel(2);
value = read_adc();
```

- value = read_adc ([***mode***])
- ***mode*** is an optional parameter. If used the values may be:

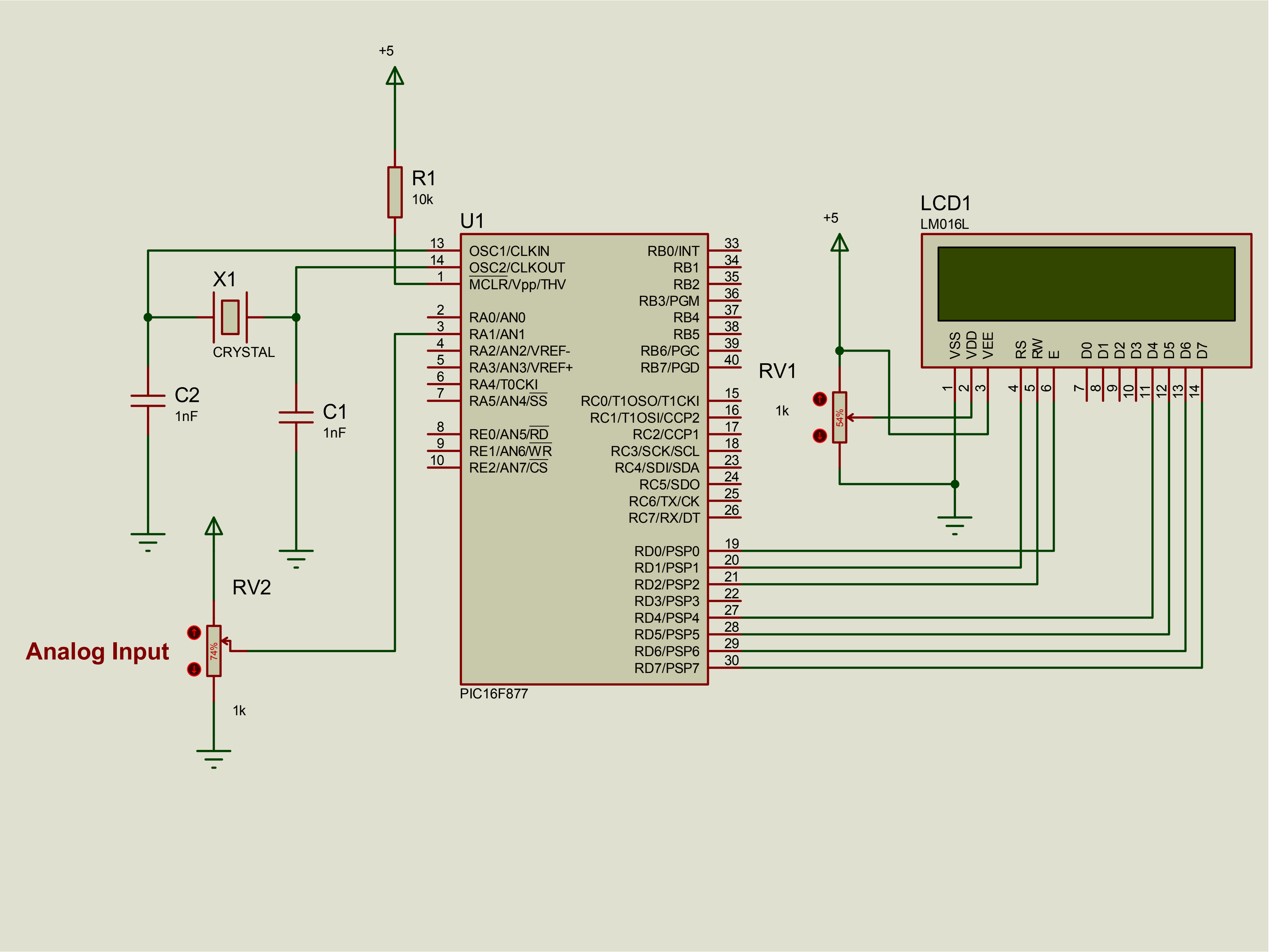
  ADC_START_AND_READ (this is the default)

  ADC_START_ONLY (starts the conversion and returns)

  ADC_READ_ONLY (reads last conversion result)

Example : ADC,Display on LCD

```
#include <16f877.h>
#fuses xt,nowdt
#use delay(clock=4000000)
#include <lcd.c>
int value;
void main(){
lcd_init();
setup_adc( ADC_CLOCK_INTERNAL );
setup_adc_ports( ALL_ANALOG );
set_adc_channel(1);
while ( 1)  {
value = read_adc();
lcd_gotoxy(1,1);
printf(lcd_putc,"A/D value = %u  ", value);}}
```

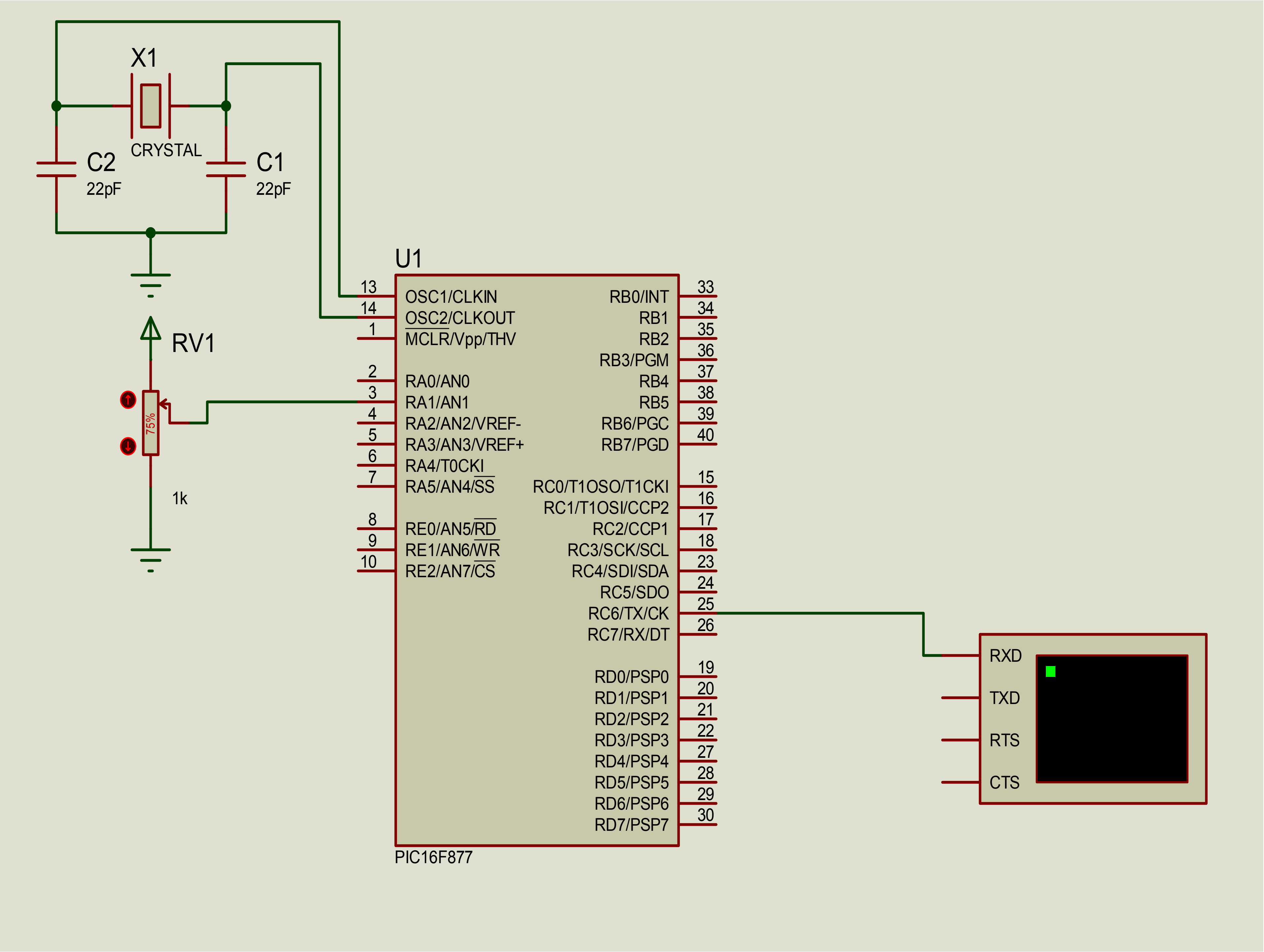

+5
R1
10k
U1
X1
CRYSTAL
C2
1nF
C1
1nF
13 OSC1/CLKIN
14 OSC2/CLKOUT
1 MCLR/Vpp/THV
2 RA0/AN0
3 RA1/AN1
4 RA2/AN2/VREF-
5 RA3/AN3/VREF+
6 RA4/T0CKI
7 RA5/AN4/SS
8 RE0/AN5/RD
9 RE1/AN6/WR
10 RE2/AN7/CS
RB0/INT 33
RB1 34
RB2 35
RB3/PGM 36
RB4 37
RB5 38
RB6/PGC 39
RB7/PGD 40
RC0/T1OSO/T1CKI 15
RC1/T1OSI/CCP2 16
RC2/CCP1 17
RC3/SCK/SCL 18
RC4/SDI/SDA 23
RC5/SDO 24
RC6/TX/CK 25
RC7/RX/DT 26
RD0/PSP0 19
RD1/PSP1 20
RD2/PSP2 21
RD3/PSP3 22
RD4/PSP4 27
RD5/PSP5 28
RD6/PSP6 29
RD7/PSP7 30
PIC16F877
+5
RV1
1k
54%
LCD1
LM016L
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
1 2 3 4 5 6 7 8 9 10 11 12 13 14
RV2
74%
1k
Analog Input

RS232

- #users232(baud=9600,xmit=PIN_C6,rcv=PIN_C7
- BAUD=x; Set baud rate to x units bits/sec
- XMIT=pin ; Set transmit pin
- RCV=pin; Set receive pin
- printf("%u",value); the printf command is used to write data to serial port.

Example:RS232

```
#include <16f877.h>
#fuses hs,nowdt
#use delay (clock=20000000)
#use rs232 (baud=9600,xmit=pin_c6,rcv=pin_c7)
void main()
{int value;
setup_adc(  ADC_CLOCK_INTERNAL  );
setup_adc_ports( ALL_ANALOG );
while (1)  {
set_adc_channel(1);
value= read_adc();
printf("A/D value = %u\n\r",value);}}
```


X1
CRYSTAL
C2
22pF
C1
22pF
RV1
75%
1k
U1
13
14
1
2
3
4
5
6
7
8
9
10
OSC1/CLKIN
OSC2/CLKOUT
MCLR/Vpp/THV
RA0/AN0
RA1/AN1
RA2/AN2/VREF-
RA3/AN3/VREF+
RA4/T0CKI
RA5/AN4/SS
RE0/AN5/RD
RE1/AN6/WR
RE2/AN7/CS
RB0/INT
RB1
RB2
RB3/PGM
RB4
RB5
RB6/PGC
RB7/PGD
RC0/T1OSO/T1CKI
RC1/T1OSI/CCP2
RC2/CCP1
RC3/SCK/SCL
RC4/SDI/SDA
RC5/SDO
RC6/TX/CK
RC7/RX/DT
RD0/PSP0
RD1/PSP1
RD2/PSP2
RD3/PSP3
RD4/PSP4
RD5/PSP5
RD6/PSP6
RD7/PSP7
33
34
35
36
37
38
39
40
15
16
17
18
23
24
25
26
19
20
21
22
27
28
29
30
PIC16F877
RXD
TXD
RTS
CTS