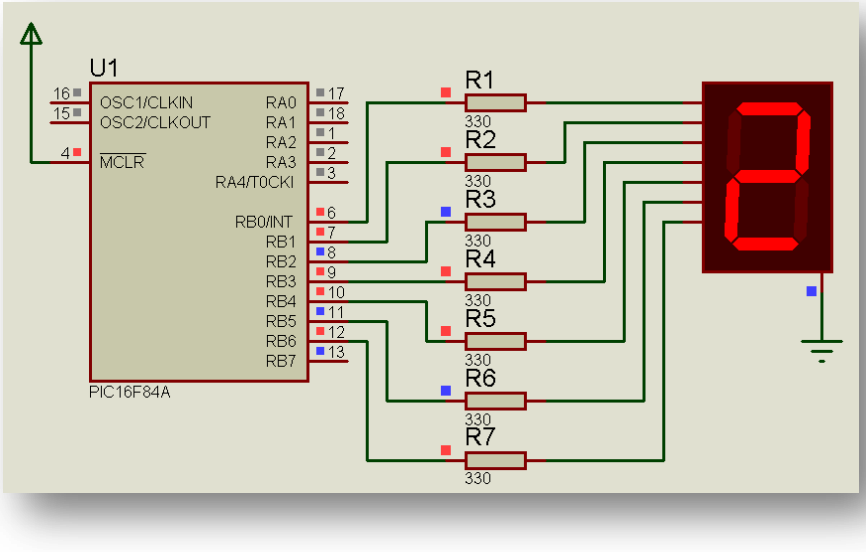
```
            delay_ms(1000); //Se hace un retardo de 1 segundo.
        }
}
```

El paso siguiente es realizar la simulación en ISIS, para este fin se buscan los siguientes dispositivos: 16F84A, RES, y 7SEG-COM-CATHODE, para la conexión entre el PIC, y el display se deben usar resistencias de 330Ω. Posteriormente se arma el siguiente circuito:



Circuito 5-1

Al correr la simulación se debe ver un conteo en el display del 0 al 9, con una cadencia de un segundo entre dígito y dígito.

5.1.2 Control de displays 7 segmentos dinámicos

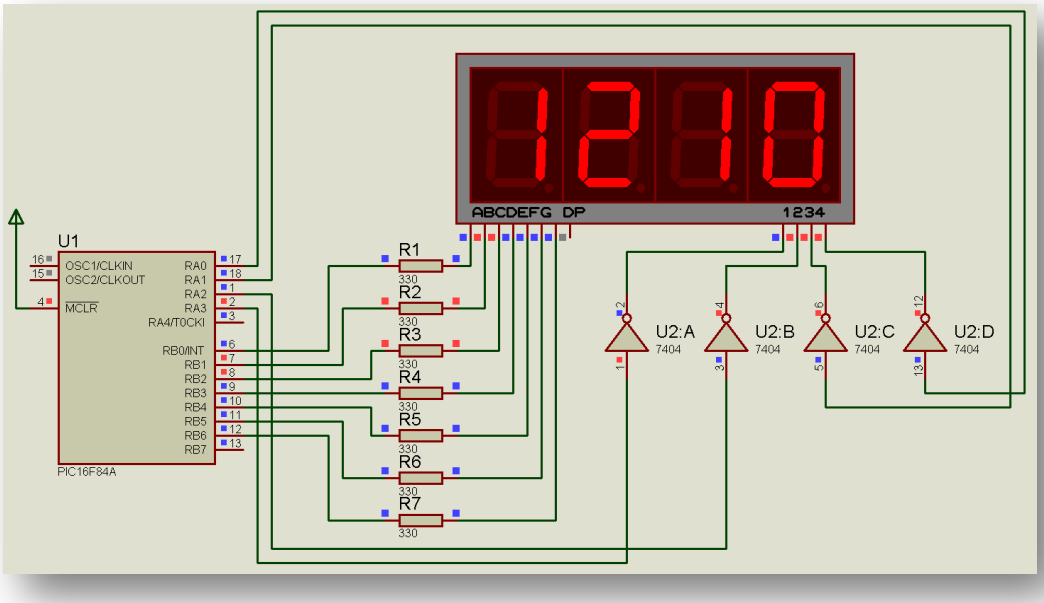
La implementación de displays dinámicos usa la misma teoría de un solo display, la visualización dinámica consiste en mostrar un solo dígito al mismo tiempo. Por ejemplo si se muestran cuatro dígitos se activa el display de las unidades, después se apagan y se activa el dígito de las decenas, posterior mente se apaga y se activa el dígito de las centenas, y por último se hace lo mismo con las unidades de mil. Este proceso se debe hacer con una velocidad de tal manera que engañe al ojo humano y así se verá como si todos los dígitos estuvieran activos. Este arreglo minimiza las conexiones eléctricas y el consumo de energía, dado que en realidad solo un dígito está activo para todo tiempo. A la vista del ojo humano los cambios deben ser de 25Hz o más, por lo tanto todos los dígitos deben verse durante un periodo igual al inverso de 25Hz, en este caso es 40m segundos. Para este ejemplo se usarán 4 displays por lo tanto el tiempo visible de cada dígito debe ser la cuarta parte del periodo es decir 10m segundos. La forma más eficiente de mostrar los números en el display es por medio de una función. Con 4 dígitos es posible visualizar un número de 0 a 9999, por esto el número puede ser consignado en una variable de tipo entera. Para ver cada dígito por separado se hace necesario calcular cada uno de los dígitos, por ejemplo para deducir las unidades de mil vasta con dividir el número en mil, y luego restár las unidades de mil del número completo, este proceso se repite hasta llegar a las unidades. La activación de los displays se debe hacer por medio de otro puerto para este ejemplo se hará por el puerto A.
Para comprender este proceso observe y analice la siguiente función:

```
// Declaración de las constantes para el display.
const unsigned short DIGITOS[] =
{
        0x3F, //Código del dígito 0
        0x06, //Código del dígito 1
        0x5B, //Código del dígito 2
        0x4F, //Código del dígito 3
        0x66, //Código del dígito 4
        0x6D, //Código del dígito 5
        0x7D, //Código del dígito 6
        0x07, //Código del dígito 7
        0x7F, //Código del dígito 8
        0x6F, //Código del dígito 9
};

//Función para visualizar el display dinámico.
void VerDisplay( int Numero )
{
        unsigned short U; //Variable para guardar las unidades.
        unsigned short D; //Variable para guardar las decenas.
        unsigned short C; //Variable para guardar las centenas.
        unsigned short UM; //Variable para guardar las unidades de mil.
        UM = Numero/1000; //Cálculo de las unidades de mil.
        C = (Numero-UM*1000)/100; //Cálculo de las centenas.
        D = (Numero-UM*1000-C*100)/10; //Cálculo de las decenas.
        U = (Numero-UM*1000-C*100-D*10); //Cálculo de las unidades.

        PORTB = DIGITOS[U];  //Visualiza las unidades.
        PORTA.F0=1; //Activa en alto el primer display
        delay_ms(10); //Retado de 10m segundos
        PORTA=0; //Desactiva todos los displays.

        PORTB = DIGITOS[D]; //Visualiza las decenas.
        PORTA.F1=1; //Activa en alto el segundo display
        delay_ms(10); //Retado de 10m segundos
        PORTA=0; //Desactiva todos los displays.

        PORTB = DIGITOS[C]; //Visualiza las centenas.
        PORTA.F2=1; //Activa en alto el tercer display
        delay_ms(10); //Retado de 10m segundos
        PORTA=0; //Desactiva todos los displays.

        PORTB = DIGITOS[UM]; //Visualiza las unidades de mil.
        PORTA.F3=1; //Activa en alto el cuarto display
        delay_ms(10); //Retado de 10m segundos
        PORTA=0; //Desactiva todos los displays.
}
```

Por último se completa el código fuente con la función *main* de la siguiente forma:

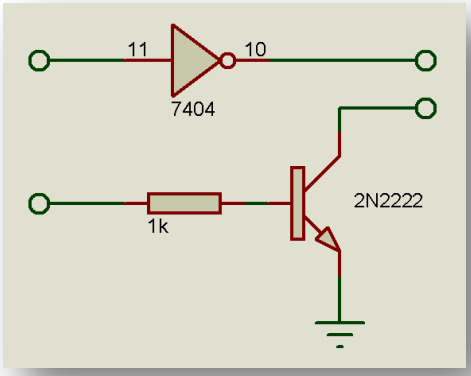
```
void main ( void )
{
        unsigned short N=0; //Variable de conteo.
        int Numero=0;
        TRISB = 0; //Configura el puerto B como salida
        TRISA = 0; //Configura el puerto A como salida
        PORTA = 0; //Se desactiva todos los displays
        while( 1 ) //Bucle infinito
        {
          //Se visualiza el valor de Número.
          VerDisplay( Numero ); //Está función dura aproximadamente 40m segundos.
          //Se cuentan 12 incrementos en N para hacer un incremento
          //en Número aproximadamente cada 500m segundos.
          N++;
          if( N==12 )
          {
             N=0; //Se reinicia el conteo de N.
             Numero++; //Se incrementa el valor de Número.
             if( Numero==10000 ) //Se evalúa si Número vale 10000
              Numero=0;          //y se reinicia en 0 si es 10000.
          }
        }
}
```

Al terminar y compilar el programa se realiza la simulación en ISIS, con los siguiente dispositivos: 16F84A, RES, 7SEG-MPX4-CC, 7404. Finalmente se construye el siguiente circuito:



Circuito 5-2

Para fines prácticos los inversores 7404 se pueden remplazar por un arreglo de transistores como se ven en la siguiente imagen:



Circuito 5-3

Cuando la simulación este en marcha el circuito debe mostrar un conteo de 0 a 9999 con una cadencia de 500m segundos entre cada incremento.

5.2 Display LCD de caracteres

Los display de caracteres LCD, son módulos prefabricados que contienen controladores incluidos. Estos displays cuentan con un bus de datos y un bus de control, para el manejo de estos dispositivos el compilador MikroC PRO, tiene una librería predefinida para el control de estos LCD. La apariencia física y la vista en ISIS de estos LCD es la que se puede apreciar en las siguientes gráficas:

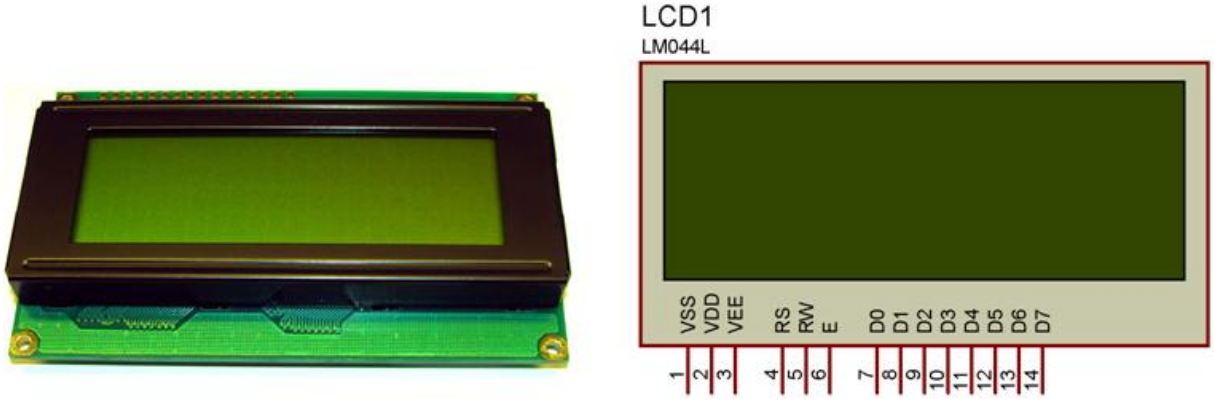


Figura 5-4

Los displays LCD, permiten graficar los caracteres contemplados en el código ASCII. Además del código ASCII, los displays LCD admiten graficar hasta 8 caracteres diseñados por el desarrollador, otra característica fundamental de los LCD, es la conexión del bus de datos, físicamente tienen 8 bits, pero es posible configurar las conexiones con solo 4 bits. La conexión de 8 bits implica una mayor cantidad de cables para su uso, pero la velocidad de trabajo es mayor, por consiguiente la conexión de 4 bits minimiza las conexiones pero disminuye la velocidad de trabajo. La librería predefinida en MikroC PRO, funciona con la configuración de 4 bits.