

Estructura básica de un programa (compilador MikroC PRO)

//declaración de variables globales

//declaración (y definición) de funciones

//declaración y definición de las rutinas de atención de las interrupciones [void interrupt()]

void main()

{

//declaración de variables

ADCON1 = 0x07; //configuración de los canales *analógicos* (AN) como *digitales* (PIC18F452)

//configuración de puertos

//configuración e inicialización de los módulos del PIC que se utilicen (si se utilizan)

//configuración de interrupciones (si se utilizan)

//instrucciones

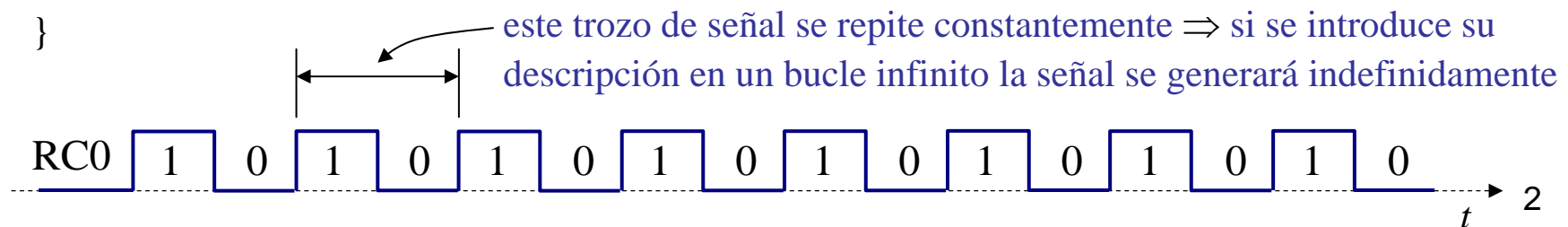
while(1) //bucle infinito

asm nop; //en este bucle se pueden poner instrucciones

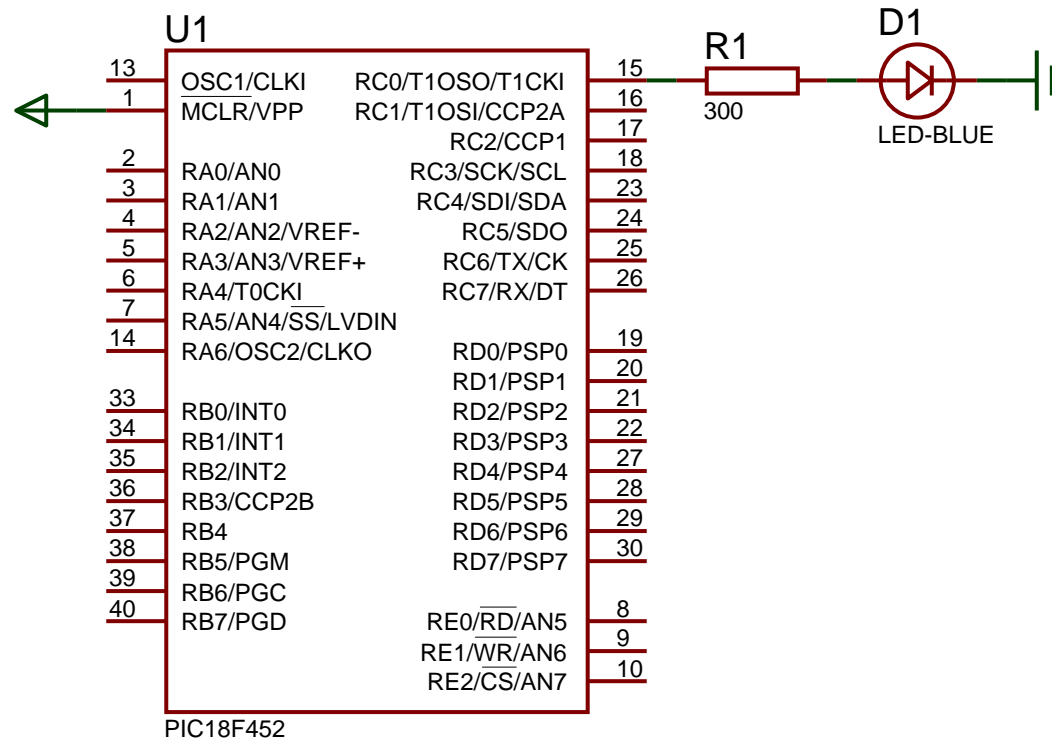
}

Ejemplo: este programa enciende y apaga un led conectado al terminal RC0 con una frecuencia de 1 segundo.

```
void main()
{
    ADCON1 = 0x07; //configuración de los canales analógicos (AN) como digitales (PIC18F452)
    TRISC.B0 = 0; //se configura el terminal RC0 como una salida digital
    PORTC.B0 = 0; //se pone la salida RC0 a 0
    while(1) //bucle infinito (se describe 1 periodo de la señal a generar)
    {
        PORTC.B0 = 1; //se pone la salida RC0 a 1
        delay_ms(600); //se introduce un retardo de 600ms en la ejecución del código
        PORTC.B0 = 0; //se pone la salida RC0 a 0
        delay_ms(300); //se introduce un retardo de 300ms en la ejecución del código
    }
}
```



Componentes ISIS: PIC18F452, RES, LED-BLUE



Nota: el led está encendido siempre que $RC0 = 1$ y está apagado siempre que $RC0 = 0$.

Tipos de datos admitidos por el compilador MikroC PRO de Mikroelektronika

Tipo	Tamaño	Rango de valores
(unsigned) <i>char</i>	8 bits	0 . . . 255
signed <i>char</i>	8 bits	-128 . . . +127
unsigned short (<i>int</i>)	16 bits	0 . . . 65535
(signed) short (<i>int</i>)	16 bits	-32768 . . . +32767
(signed) <i>int</i>	32 bits	-2147483648 . . . +2147483647
unsigned (<i>int</i>)	32 bits	0 . . . 4294967295
(signed) long (<i>int</i>)	64 bits	-9223372036854775808 . . . +9223372036854775807
unsigned long (<i>int</i>)	64 bits	0 . . . 18446744073709551615
<i>float</i>	32 bits	$\pm 1.17549435082 \cdot 10^{-38}$ to $\pm 6.80564774407 \cdot 10^{38}$
<i>double</i>	64 bits	$\pm 1.17549435082 \cdot 10^{-38}$ to $\pm 6.80564774407 \cdot 10^{38}$
long double	128 bits	$\pm 1.17549435082 \cdot 10^{-38}$ to $\pm 6.80564774407 \cdot 10^{38}$