

Programa en MikroC que lee y transmite asíncronamente bytes en serie para un PIC16F887 con módulo USART (Universal synchronous/asynchronous receiver/transmitter)

```

unsigned short dato; // Se declara "dato" para almacenar un byte u octeto

void main() {

    ANSEL=ANSELH=0x00; // Deshabilita entradas analógicas, pone todo en digital

    PORTB=0x00; // Pone a 0 todos los bits del registro RB

    TRISB=0x00; // Habilita todos los bits de RB como salidas

    // Inicializa módulo USART (8 bit, 9600 bauds, sin bit de paridad)
    Usart_Init(9600);

    do { // Ciclo infinito para checar continuamente la transmisión

        if (Usart_Data_Ready()) { // Si un byte es recibido

            dato = Usart_Read(); // Lee el byte recibido del pin Rx del PIC en "dato"

            switch (dato){ // Verifica el valor del byte recibido por el PIC

                case 'A': PORTB.F0=1; break; // Si el dato es "A" pone 1 en el bit 0 de RB
                case 'R': PORTB.F1=1; break; // Si el dato es "R" pone 1 en el bit 1 de RB
                case 'V': PORTB.F2=1; break; // Si el dato es "V" pone 1 en el bit 2 de RB
                case 'a': PORTB.F0=0; break; // Si el dato es "a" pone 0 en el bit 0 de RB
                case 'r': PORTB.F1=0; break; // Si el dato es "r" pone 0 en el bit 1 de RB
                case 'v': PORTB.F2=0; break; // Si el dato es "v" pone 0 en el bit 2 de RB

            }

            Usart_Write(dato); // Envía el dato leído vía USART por el pin Tx

        }

    } while (1);

}

```

