

Reporte de la Actividad 20. Auto, 0-F, 0-9 c/DEC, 0-9999 Display en CCS

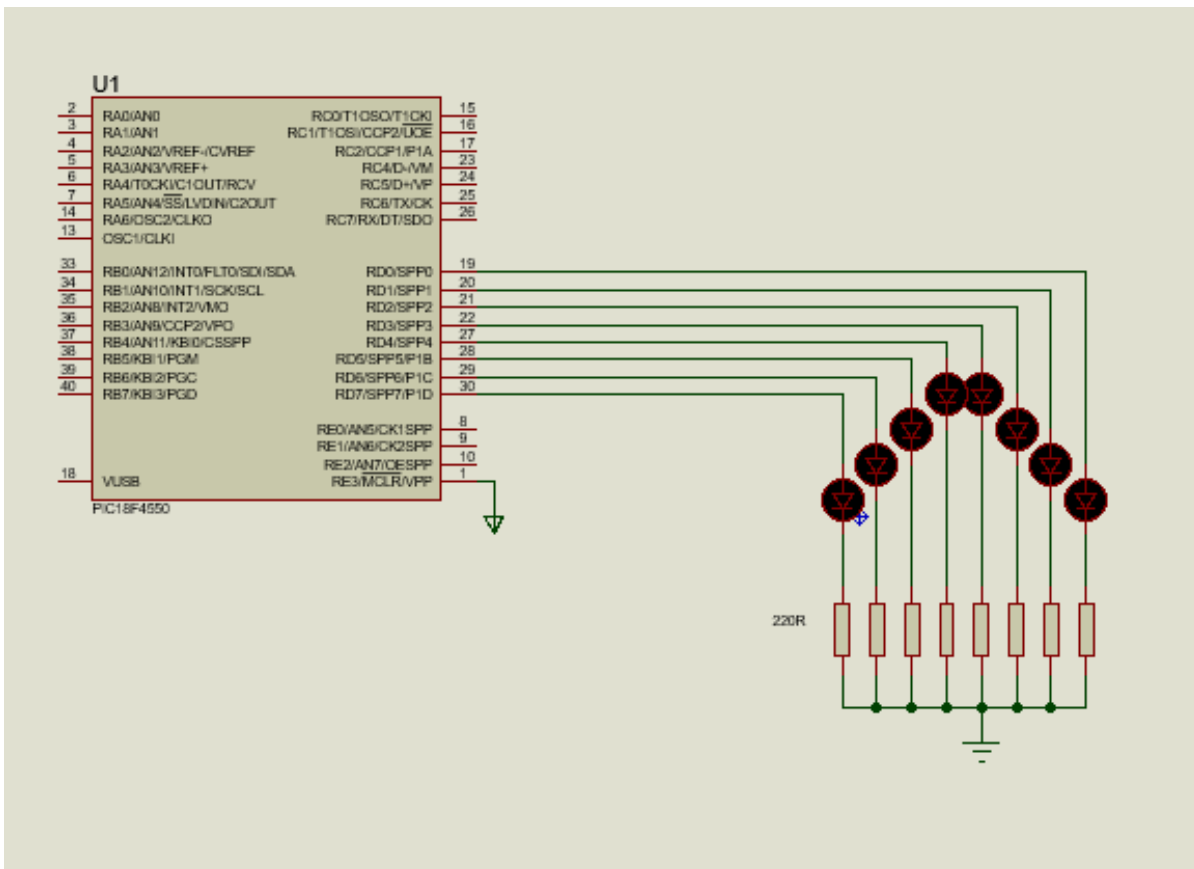
NOMBRE: Orlando Contreras Reyes	NL: 6
--	--------------

Título: Auto, 0-F, 0-9 c/DEC, 0-9999 Display en CCS

- El enunciado del problema. Escríbelo de forma digital
Realiza los programas Auto, 0-F, 0-9 c/DEC, 0-9999 Display en PIC C COMPILER
- El croquis (si es que se usó). Dibújalo de forma digital
- La tabla de verdad (si es que se usó). Realízala de forma digital.

PORTC								Valor Hex
.7	.6	.5	.4	.3	.2	.1	.0	
1	0	0	0	0	0	0	1	81H
0	0	0	1	1	0	0	0	18H
0	1	0	0	0	0	1	0	42H
0	0	1	0	0	1	0	0	24H
0	0	0	1	1	0	0	0	18H
0	0	1	0	0	1	0	0	24H
0	1	0	0	0	0	1	0	42H
1	0	0	0	0	0	0	1	81H

- El diagrama electrónico. Realízalo a mano.



e) Código en CCS

```
//----- MAIN LIBRARY -----
#include <18f4550.h>
//----- FUSES CONFIGURATION -----
#fuses NOWDT,HS,PUT,NOPROTECT,NOBROWNOUT,NOMCLR,NOLVP,NOCPPD
#use delay(clock=4000000)

//----- EXT LIBRARIES -----

//----- SET OUTPUTS -----
//--Ports-
#byte Port_D = 0X0F83 // Port_D es equivalente a la dirección de RAM 0F83
#byte Tris_D = 0x0F95 // Tris_D es equivalente a la dirección de RAM 0F95
//--Var--

//--Inicio--
void main(){
    Tris_D = 0x00; // Set TRISD as OUTPUT
    Port_D = 0X00; // Leds OFF

    while(true){
        Port_D = 0x81; // 81H -> 1000 0001
        delay_ms(500);

        Port_D = 0x18; // 18H -> 0001 1000
        delay_ms(200);

        Port_D = 0x42; // 42H -> 0100 0010
        delay_ms(500);

        Port_D = 0x24; // 24H -> 0010 0100
        delay_ms(500);

        Port_D = 0x18; // 18H -> 0001 1000
        delay_ms(500);
        Port_D = 0x24; // 24H -> 0010 0100
        delay_ms(500);
        Port_D = 0x42; // 42H -> 0100 0010
    }
}
```

f) Descripción del código en CCS

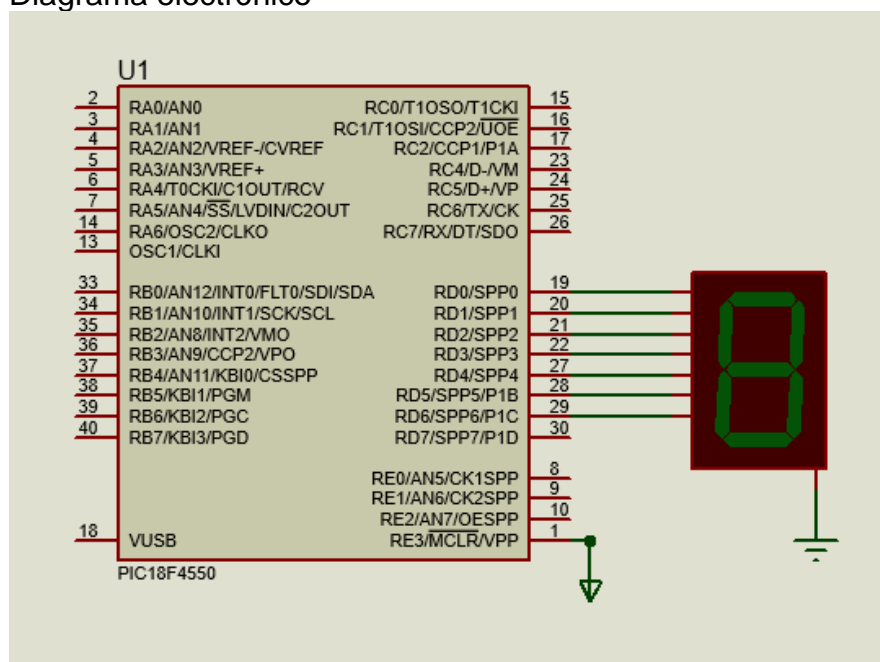
- 1) Se incluye la librería del PIC en el que trabajaremos en este caso es del 18f4550
- 2) Se configuran los fusibles a usar y se establece la velocidad del cristal que será de 4MHz

- 3) Se declaran variables en la RAM con la directriz #byte nombrevariable = lugarderam (Se declara el puerto D y el Trisd)
- 4) En el void main se configura el puerto d como salida mediante la instrucción Tris_D= 0x00 y el valor inicial de puerto d es 0x00
- 5) Se mueven los valores que se tienen en la tabla de verdad y se mueven al puerto D además de ponerle un delay de 500 microsegundos.(0x81,0x18,0x42,0x24,0x18,0x24,0x42)

g) Tabla de verdad 0-F.

				PORTC								Valor Hex	Display
				.7	.6	.5	.4	.3	.2	.1	.0		
D	C	B	A	g	f	e	d	c	b	a			
0	0	0	0	0	0	1	1	1	1	1	1	3F	0
0	0	0	1	0	0	0	0	0	1	1	0	06	1
0	0	1	0	0	1	0	1	1	0	1	1	5B	2
0	0	1	1	0	1	0	0	1	1	1	1	4F	3
0	1	0	0	0	1	1	0	0	1	1	0	66	4
0	1	0	1	0	1	1	0	1	1	0	1	6D	5
0	1	1	0	0	1	1	1	1	1	0	1	7D	6
0	1	1	1	0	0	0	0	0	1	1	1	07	7
1	0	0	0	0	1	1	1	1	1	1	1	7F	8
1	0	0	1	0	1	1	0	0	1	1	1	67	9
1	0	1	0	0	1	1	1	0	1	1	1	77	A
1	0	1	1	0	1	1	1	1	1	0	0	7C	B
1	1	0	0	0	0	1	1	1	0	0	1	39	C
1	1	0	1	0	1	0	1	1	1	1	0	5E	D
1	1	1	0	0	1	1	1	1	0	0	1	79	E
1	1	1	1	0	1	1	1	0	0	0	1	71	F

h) Diagrama electrónico



i) Código en CCS

```
//----- MAIN LIBRARY -----
#include <18f4550.h>
```

```

//----- FUSES CONFIGURATION -----
#fuses NOWDT,HS,PUT,NOPROTECT,NOBROWNOUT,NOMCLR,NOLVP,NOCPPD
#use delay(clock=4000000)

//----- EXT LIBRARIES -----

//----- SET OUTPUTS -----
//--Ports-
#byte Port_D = 0X0F83 // Port_D es equivalente a la dirección de RAM 0F83
#byte Tris_D = 0x0F95 // Tris_D es equivalente a la dirección de RAM 0F95
//--Var--

//--Inicio--
void main(){
    Tris_D = 0x00; // Set TRISD as OUTPUT
    Port_D = 0X00; // Leds OFF

    while(true){
        Port_D = 0x3F; //3FH-> 0
        delay_ms(500);
        Port_D = 0x06; //06H-> 1
        delay_ms(500);
        Port_D = 0x5B; //5BH-> 2
        delay_ms(500);
        Port_D = 0x4F; //4FH-> 3
        delay_ms(500);

        Port_D = 0x66; //66H-> 4
        delay_ms(500);
        Port_D = 0x6D; //6DH-> 5
        delay_ms(500);
        Port_D = 0x7D; //07H-> 6
        delay_ms(500);
        Port_D = 0x07; //07H-> 7
        delay_ms(500);

        Port_D = 0x7F; //7FH-> 8
        delay_ms(500);
        Port_D = 0x67; //67H-> 9
        delay_ms(500);
        Port_D = 0x77; //77H-> A
        delay_ms(500);
        Port_D = 0x7C; //7CH-> B
        delay_ms(500);

        Port_D = 0x39; //39H-> C
        delay_ms(500);
        Port_D = 0x5E; //5EH-> D
        delay_ms(500);
        Port_D = 0x79; //79H-> E
        delay_ms(500);
    }
}

```

```

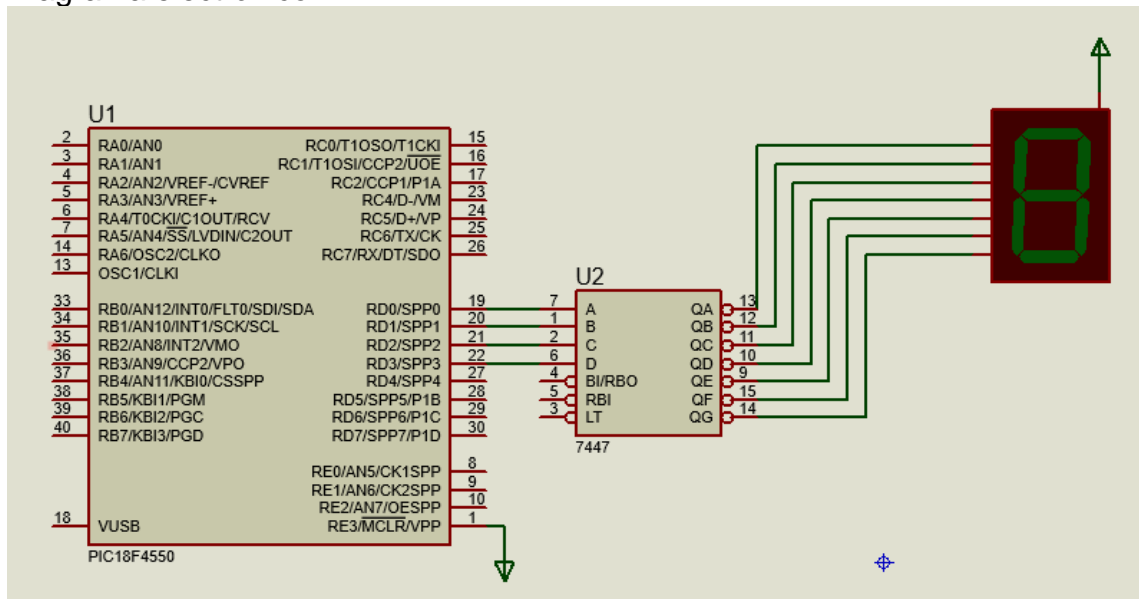
Port_D = 0x71;//71H-> F
    delay_ms(500);
}
}

```

j) Explicación del código en CCS

- 1) Se incluye la librería del PIC en el que trabajaremos en este caso es del 18f4550
- 2) Se configuran los fusibles a usar y se establece la velocidad del cristal que será de 4MHz
- 3) Se declaran variables en la RAM con la directriz #byte nombrevariable = lugarderam (Se declara el puerto D y el Trisd)
- 4) En el void main se configura el puerto d como salida mediante la instrucción Tris_D= 0x00 y el valor inicial de puerto d es 0x00
- 5) Se mueven los valores que se tienen en la tabla de verdad y se mueven al puerto D además de ponerle un delay de 500 microsegundos.
(0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x67,0x77,0x7C,0x39,0x5E,0x79,0x71)

k) Diagrama electrónico



l) Codigo en CCS

```

//----- MAIN LIBRARY -----
#include <18f4550.h>
//----- FUSES CONFIGURATION -----
#fuses NOWDT,HS,PUT,NOPROTECT,NOBROWNOUT,NOMCLR,NOLVP,NOCPPD
#use delay(clock=4000000)

//----- EXT LIBRARIES -----
#include <lcd.c>

//----- SET OUTPUTS -----
//--Ports-
#byte Port_D = 0x0F83 // Port_D es equivalente a la dirección de RAM 0F83
#byte Tris_D = 0x0F95 // Tris_D es equivalente a la dirección de RAM 0F95
//--Var--

```

```
int8 cont;
```

```
//--Inicio--
```

```
void main(){
```

```
    Tris_D = 0x00; // Set TRISD as OUTPUT
```

```
    Port_D = 0x00; // Leds OFF
```

```
    cont = 0x00;
```

```
    while(true){
```

```
        for (cont = 0; cont<=9; cont++){ //If we put <= the variable can take the value
```

```
            Port_D=cont;
```

```
            delay_ms(200);
```

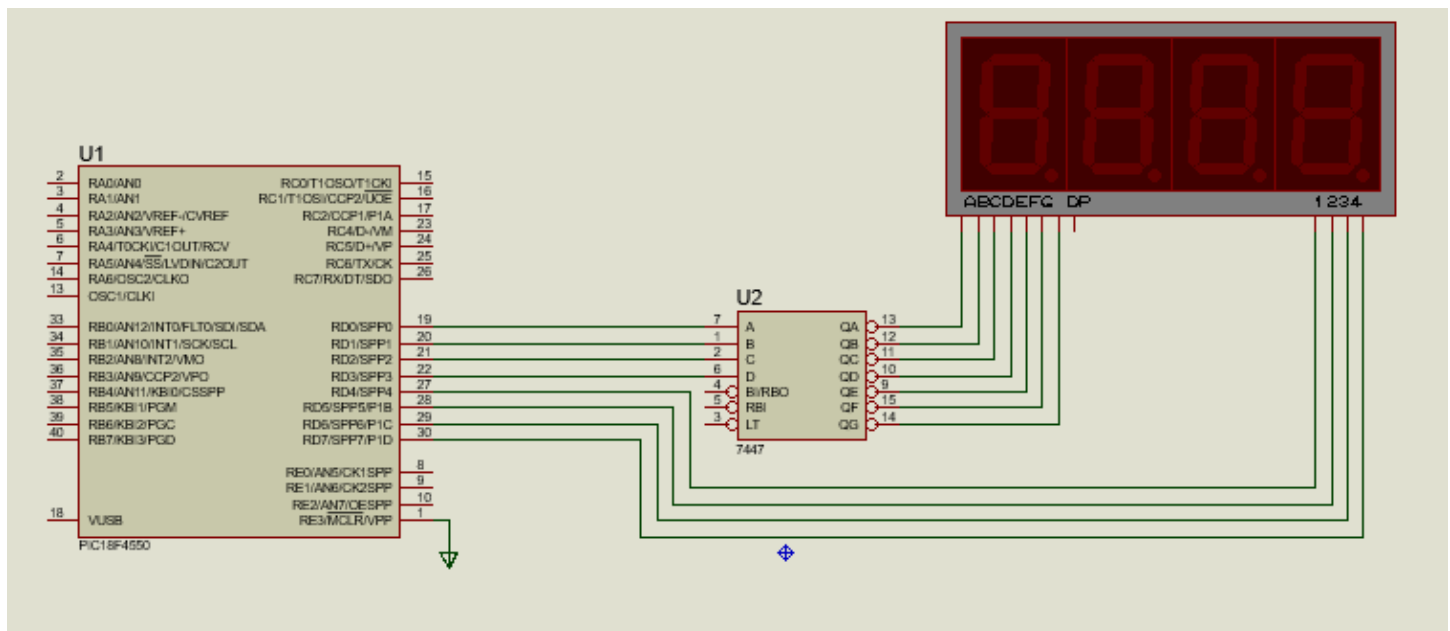
```
        }
```

```
    }
```

m) Explicación del código en CCS

- 1) Se incluye la librería del PIC en el que trabajaremos en este caso es del 18f4550
- 2) Se configuran los fusibles a usar y se establece la velocidad del cristal que será de 4MHz
- 3) Se declaran variables en la RAM con la directriz #byte nombredevariable = lugarde ram (Se declara el puerto D y el Trisd)
- 4) En el void main se configura el puerto d como salida mediante la instrucción Tris_D= 0x00 y el valor inicial de puerto d es 0x00
- 5) Se utiliza un ciclo for para incrementar los números del 0 al 9 que pasarán por un decodificador para que se puedan verse en el display

n) Diagrama electrónico 0-9999



o) Código en CCS

```
//----- MAIN LIBRARY -----
```

```
#INCLUDE <18f4550.h>
```

```
//----- FUSES CONFIGURATION -----
```

```
#fuses NOWDT,HS,PUT,NOPROTECT,NOBROWNOUT,NOMCLR,NOLVP,NOCPSD
```

```
#use delay(clock=4000000)
```

```
//----- EXT LIBRARIE -----
```

```
#include <lcd.c>
```

```

//----- SET OUTPUTS -----
//--Ports-
#byte Port_D = 0X0F83 // Port_D es equivalente a la dirección de RAM 0F83
#byte Tris_D = 0x0F95 // Tris_D es equivalente a la dirección de RAM 0F95
//--Var--
int8 uni,dec,cen,mil;

//--Inicio--
void main(){
    Tris_D = 0x00;// Set TRISD as OUTPUT
    Port_D = 0X00; // Leds OFF

    while(true){
        for(mil=0;mil<=9;mil++)
        {
            for(cen=0;cen<=9;dec++)
            {
                for(dec=0;dec<=9;dec++)
                {
                    for (uni = 0;uni<=9;uni++)
                    {
                        Port_D=mil | 0x10;
                        delay_ms(200);
                        Port_D=cen | 0x20;
                        delay_ms(200);
                        Port_D=dec | 0x40;
                        delay_ms(200);
                        Port_D=uni | 0x80;
                        delay_ms(200);
                    }//End Uni for
                }//End Dec For
            }//End Cen for
        }//End Mil For
    }//End While

} //End main

```

p) Explicación del código en CCS

- 1) Se incluye la librería del PIC en el que trabajaremos en este caso es del 18f4550
- 2) Se configuran los fusibles a usar y se establece la velocidad del cristal que será de 4MHz
- 3) Se declaran variables en la RAM con la directriz #byte nombredevariable = lugarderam (Se declara el puerto D y el Trisd)
- 4) En el void main se configura el puerto d como salida mediante la instrucción Tris_D= 0x00 y el valor inicial de puerto d es 0x00
- 5) Se utilizan varios ciclos for de forma anidada para ir evaluando si ya llego a su limite y si no ha llegado incrementar para que al final los valores se les aplique una Or con el valor 0x10,0x20,0x40,0x80 según sea su posición pues estos valores servirán para indicar a los 4 displays en que posición se mostrará el numero.