

Escuela de Educación Secundaria Técnica N°1
de Vicente López "Eduardo Ader"

Tecnología Electrónica

6to Año

Trabajo Práctico Trabajo Práctico
Multiplexado de displays

Alumno: Agustin Menechino

Profesor: Martin Lopez

Curso: 6° 1°



Multiplexado de displays

Informe de punto 4

- Empiezo agregando las librerías y luego empiezo a definir los puertos DDRB y PORTD con PRENDER_DISPLAY y NUM para hacerme el trabajo mas sencillo y para que el código quede de una forma mas prolija.

```
lexado de displays Punto4 Agustin Menechino > src > main.cpp > ...  
//Agustin Menechino 6°1 PUNTO 4 Multiplexado de displays  
  
//Agrego las librerías que voy a necesitar  
#include <Arduino.h>  
#include <avr/io.h>  
#include <avr/interrupt.h>  
  
#define PRENDER_DISPLAY_1 DDRB = 0b0000001;  
#define PRENDER_DISPLAY_2 DDRB = 0b0000010;  
#define PRENDER_DISPLAY_3 DDRB = 0b0000100;  
#define PRENDER_DISPLAY_4 DDRB = 0b0001000;  
  
#define NUMERO_0 PORTD = 0b0111111;  
#define NUMERO_1 PORTD = 0b0000110;  
#define NUMERO_2 PORTD = 0b1011011;  
#define NUMERO_3 PORTD = 0b1001111;  
#define NUMERO_4 PORTD = 0b1100110;  
#define NUMERO_5 PORTD = 0b1101101;  
#define NUMERO_6 PORTD = 0b1111101;  
#define NUMERO_7 PORTD = 0b0000111;  
#define NUMERO_8 PORTD = 0b1111111;  
#define NUMERO_9 PORTD = 0b1100111;
```

```
//Creo una función para luego definir que display voy a utilizar y que numero voy a mostrar en el display  
void FUNCION_DISPLAY (char NUM_DISPLAY , char NUM)  
{  
    if (NUM_DISPLAY == 1) //Utilizo un if para preguntar si NUM_DISPLAY vale 1  
    {  
        PRENDER_DISPLAY_1; //si es verdadero prende el display 1  
    }  
}
```

- Creo la función llamada “FUNCION_DISPLAY”, su función es ingresas el numero del display (en NUM_DYSPLAY) que deseas encender y el numero que quieres que aparezca (NUM).
- El if pregunta si NUM_DISPLAY es igual a 1, y si es verdadero prende el display 1, lo mismo para el resto de displays.

Escuela de Educación Secundaria Técnica N° 1
De Vicente López

Alumno: Agustin Menechino
Docente: Martin Lopez

Observaciones:

Curso: 6°1° Grupo: B
Multiplexado de displays

Calificación:

Firma del docente:



Multiplexado de displays

```

8
9  if (NUM == 0)                                //Pregunto si num es igual a 0
10 {
11     NUMERO_0;                                //Si es verdadero muestra el numero 0
12 }
13
14  if (NUM == 1)
15  {
16     NUMERO_1;
17  }
18
19  if (NUM == 2)

```

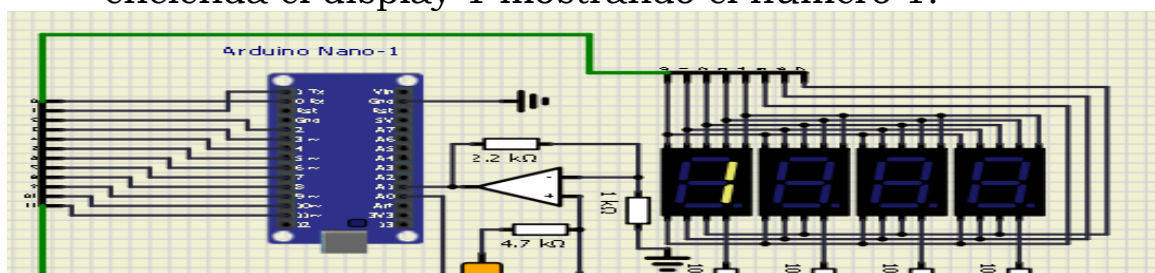
- En la parte de los números es lo mismo, el if pregunta si NUM es igual a 0, y si es verdadero muestra el numero 0, lo mismo con el resto de números(1,2,3,4, etc..).

```

20
21 int main()
22 {
23     DDRD = 0xFF; //Pone todo el puerto "D" en 1
24
25     while (1)
26     {
27         FUNCION_DISPLAY (1 , 1); //Procedo a elegir que display y numero quiero mostrar
28     }
29 }

```

- Y en el final del código agrego un **int main ()**, adentro de el pongo que todo el puerto "D" este en 1 y después de eso agrego un loop infinito (**while(1)**), adentro de el llamo a la función FUNCION_DISPLAY y le digo que encienda el display 1 mostrando el numero 1.



Escuela de Educación Secundaria Técnica N° 1
De Vicente López

Alumno: Agustin Menechino
Docente: Martin Lopez

Curso: 6°1° Grupo:B
Multiplexado de displays

Observaciones:

Calificación:

Firma del docente:



Multiplexado de displays

Informe de punto 5

```

multiplexado de displays Punto 5 > src > main.cpp > @ NUMERO_MCDU[char, char, char, char]
19 #define NUMERO_5 PORTD = 0b1101101;
20 #define NUMERO_6 PORTD = 0b1111101;
21 #define NUMERO_7 PORTD = 0b0000111;
22 #define NUMERO_8 PORTD = 0b1111111;
23 #define NUMERO_9 PORTD = 0b1100111;
24
25 char sum = 0; //Inicializo sum con el valor 0
26
27 void config_TIMER0(void)
28 {
29     TCCR0A = (1 << WGM01); //Activa el bit CTC (clear timer on compare match)
30     //del TCCR0A (timer counter/control register)
31     OCR0A = 155; //valor de comparación de int cada 10ms
32     TCCR0B = (1 << CS00) | (1 << CS01);
33     TIMSK0 = (1 << OCIE0A); //Habilita las interrupciones en timer compare
34 }
35
36
37 void FUNCION_DISPLAY (char NUM_DISPLAY , char NUM)
38 {
39
40     if (NUM_DISPLAY == 1) //Utilizo un if para preguntar si NUM_DISPLAY vale 1
41     {
42         PRENDER_DISPLAY_1; //si es verdadero prende el display 1
43     }
44
45     if (NUM_DISPLAY == 2)
46     {

```

- Empiezo con los define que son los mismo que en el punto 4, solo que acá agrego un char llamado “sum”, este me va a servir para el timer0. Lo defino como cero porque la función de mi ISR va sumar para que cumpla las condiciones.
- Agrego la función del timer0 con prescaler a 1024 a 10 ms.
- Agrego la función “FUNCION_DISPLAY” .

Escuela de Educación Secundaria Técnica N° 1
De Vicente López

Alumno: Agustin Menechino
Docente: Martin Lopez

Curso: 6°1° Grupo:B
Multiplexado de displays

Observaciones:

Calificación:

Firma del docente:



Multiplexado de displays

```

modo de displays Punto 5 > src > main.cpp > NUMERO_MCDU(char, char, char, char)
//Creo la función de NUMERO_MILES_CENTENA_DECENA_UNIDAD
void NUMERO_MCDU(char Mil, char Cent, char Dec, char Unid)
{
    if(sum == 20){          //Hasta que el char sum no sea 20 siempre va a pasar por ISR
        FUNCION_DISPLAY(1, Mil); //y el ISR lo va a sumar hasta que logre el numero pedido
    }

    if(sum == 40){
        FUNCION_DISPLAY(2, Cent);
    }

    if(sum == 60){
        FUNCION_DISPLAY(3, Dec);
    }

    if(sum == 80){
        sum = 0;
        FUNCION_DISPLAY(4, Unid);
    }
}

```

- Creo la función de unidad, decena, centena y unidad de mil.
- Lo que hace es pregunta si sum llego a 20 y si es verdad llama la “FUNCION_DISPLAY” y enciende el display 1, según el numero que pongamos en la centena de mil va a mostrarlo en el display 1 y así lo mismo con el resto, lo único que cambia es la condición para entrar a la función va pidiendo mas para que se logre el barrido y no sea bloqueante.
- Al final de la función sum vuelve a valer 0 para que se repita la función.

Escuela de Educación Secundaria Técnica N° 1
De Vicente López

Alumno: Agustin Menechino
Docente: Martin Lopez

Observaciones:

Curso: 6°1° Grupo:B
Multiplexado de displays

Calificación:

Firma del docente:



Multiplexado de displays

```
int main()
{
    config_TIMER0(); //Habilito timer
    sei(); //interrupciones globales

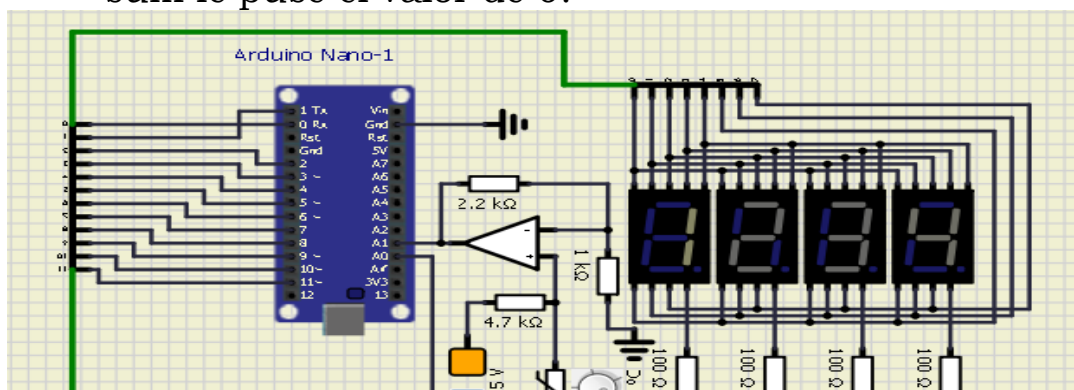
    DDRD = 0xFF; //Pone todo el puerto "D" en 1

    while (1)
    {
        NUMERO_MCDU(1,2,3,4);
    }
}

ISR(TIMER0_COMPA_vect)
{
    sum++;
}
```

Creo un int main() adentro llamo la función de timer0, habilito las interrupciones globales (sei()), pongo en 1 todos los puertos "D", creo un loop infinito que adentro esta la función para mostrar en el display los 4 dígitos que quiero.

- El ISR es lo que va hacer que sum valla incrementando para poder cumplir las condiciones de los if, por eso sum le puse el valor de 0.



Escuela de Educación Secundaria Técnica N° 1
De Vicente López

Alumno: Agustin Menechino
Docente: Martin Lopez

Observaciones:

Curso: 6°1° Grupo: B
Multiplexado de displays

Calificación:

Firma del docente:



Multiplexado de displays

Informe de punto 6

```
//Aca creo la funcion para descomponer un numero de 4 digitos
void NUM_ENTERO(int num)
{
    int Mil, Cent, Dec, Unid;

    Mil = num / 1000;
    Cent = (num - (Mil * 1000)) / 100;
    Dec = (num - (Mil * 1000 + Cent * 100)) / 10;
    Unid = (num - (Mil * 1000 + Cent * 100 + Dec * 10));

    if (sum == 20) //Hasta que el char sum no sea 20 siempre va a pasar por ISR
    {
        //y el ISR lo va a sumar hasta que logre el numero pedido
        FUNCION_DISPLAY(1, Mil);
    }

    if (sum == 40)
    {
        FUNCION_DISPLAY(2, Cent);
    }

    if (sum == 60)
    {
        FUNCION_DISPLAY(3, Dec);
    }

    if (sum == 80)
    {
        sum = 0; //Cuando llegue hasta aca se reinicia el contador llevandolo a 0, empezando el ciclo otra vez
        FUNCION_DISPLAY(4, Unid);
    }
}
```

- Básicamente agregé una función que me descompone los números de 4 dígitos en unidad, decena, centena y unidad de mil. Creo un if preguntando si sum llego a 20 y si es verdad llama la función “FUNCION_DISPLAY” y enciende el display 1 con el numero en la unidad de mil que pongamos, lo mismo en el resto de condiciones para el resto de displays.

Escuela de Educación Secundaria Técnica N° 1 De Vicente López	
Alumno: Agustin Menechino Docente: Martin Lopez	Curso: 6°1° Grupo:B Multiplexado de displays
Observaciones:	Calificación: Firma del docente:



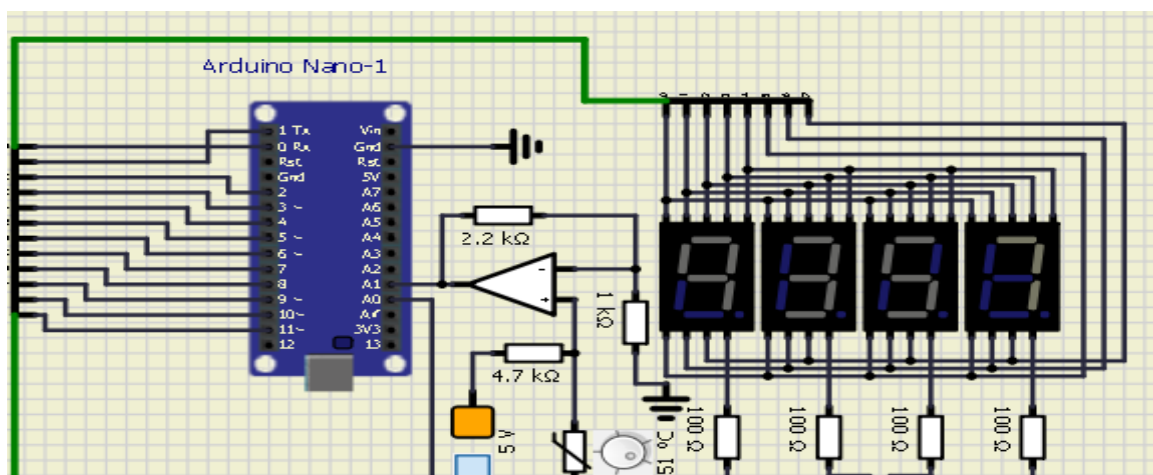
Multiplexado de displays

```

166
167   int main()
168   {
169
170       config_TIMER0(); //Habilito timer
171       sei(); //interrupciones globales
172
173       DDRD = 0xFF; //Pone todo el puerto "D" en 1
174
175       while (1)
176       {
177           NUM_ENTERO(9357);
178       }
179   }
180
181   ISR(TIMER0_COMPA_vect)
182   {
183       sum++;
184   }

```

Bueno adentro del loop infinito llamo la función NUM_ENTERO pidiendo que muestre el numero 9357.



<p>Escuela de Educación Secundaria Técnica N° 1 De Vicente López</p>	
<p>Alumno: Agustin Menechino Docente: Martin Lopez</p>	<p>Curso: 6°1° Grupo: B Multiplexado de displays</p>
<p>Observaciones:</p>	<p>Calificación: Firma del docente:</p>