

“MANEJO DE PERIFÉRICOS DE E/S (Teclado Matricial y barrido de displays)”

Informe N°6

Laboratorio de Sistemas Embebidos

Melanny Dávila

Ingeniería en Telecomunicaciones
Facultad de Eléctrica y Electrónica
Quito, Ecuador
melanny.davila@epn.edu.ec

2nd Jonathan Álvarez

Ingeniería en Telecomunicaciones
Facultad de Eléctrica y Electrónica
Quito, Ecuador
jonathan.alvarez@epn.edu.ec

Abstract—El manejo de periféricos de entrada y salida es importante en sistemas embebidos para brindar de versatilidad a los proyectos que estén controlando. En el presente informe se presentará un ejemplo con un teclado y un sistema de barrido de displays simulando una calculadora. Posteriormente se hablará de las ventajas y desventajas de dichos barridos y finalmente se describirán aplicaciones del mismo.

Index Terms—Arduino, barrido, teclado, display.

I. INTRODUCCIÓN

Un periférico de entrada/salida o E/S es aquel tipo de dispositivo periférico de un dispositivo capaz de interactuar con los elementos externos a ese sistema de forma bidireccional, es decir, que permite tanto que sea ingresada información desde un sistema externo, como emitir información a partir de ese sistema [1].

El teclado matricial es un dispositivo bastante común en proyectos con sistemas microcontrolados como el Arduino y está conformado básicamente por un arreglo de pulsadores distribuidos en forma de matriz, que puede ser leído utilizando pocos pines del Arduino y utilizando el concepto de la multiplexación. En el mercado podemos encontrar Teclados matriciales de diferentes tamaños, como el teclado matricial 4×4, 3×3, 4×3, entre otros. Estos teclados pueden tener una estructura de membrana o una estructura rígida [1].

II. OBJETIVOS

- Relacionar al estudiante con el uso y manejo de periféricos de entrada y salida en Arduino
- Conocer acerca de las diferentes librerías para el control de periféricos de entrada o salida en Arduino [2].

III. CUESTIONARIO

A. Implementar un circuito que simule la operación de una calculadora básica (suma, resta, multiplicación y división) utilizando dos números que serán ingresados por teclado (A y B). El funcionamiento de la calculadora se describe a continuación: Se utilizará los dos displays de la derecha para mostrar el número A y los dos de la izquierda para mostrar el número B, los números deberán ser mostrados en los displays conforme se vayan ingresando por teclado. Solo se pueden ingresar hasta dos dígitos, es decir, el número máximo que se puede ingresar será de 99.

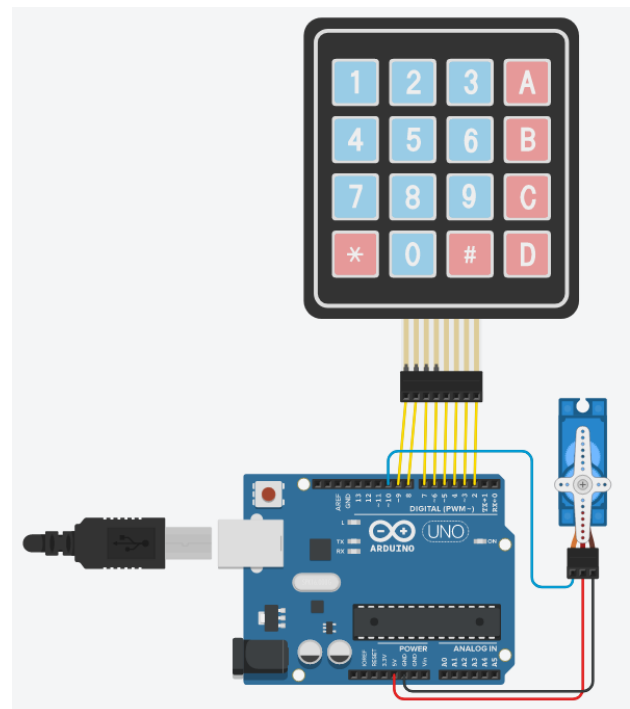


Fig. 1. Esquemático de la simulación de una calculadora básica.

```
• #include <Keypad.h>
```

```

2 #include <Servo.h>
3 // Informe 6
4 //En el presente ejercicio se realiza las
  operaciones de solicitudes
5 //haciendo uso de un teclado y displays
6 //Si se presiona cualquier tecla de operacion se
  cambia al segundo
7 //numero, se puede aplastar la tecla AC par
  avolver al inicio
8 //y una vez obtenido el resultado si se presiona
  cualquier tecla
9 //se borra el resultado
10 // Declaracion de la parte de control del
    teclado
11 const byte ROWS = 4; //4 filas
12 const byte COLS = 4; //4 columnas
13 // Valores que tendran las teclas
14 char keys[ROWS][COLS] = {
15   {'1', '2', '3', 'A'},
16   {'4', '5', '6', '+'},
17   {'7', '8', '9', '-'},
18   {'*', '0', '.', '/'};
19 byte rowPins[ROWS] = {A0, A1, A2, A3};
20 byte colPins[COLS] = {A4, A5, 1, 0};
21 // Creacion del objeto teclado
22 Keypad keypad = Keypad( makeKeymap(keys),
    rowPins, colPins, ROWS, COLS );
23
24
25 // Parte del barrido de displays de 7 segmentos
26 int display7c[10]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,
    0x7d,0x07,0x7f,0x67}; //Se inicializa el
    display
27 byte a=2;
28 byte b=3;
29 byte c=4;
30 byte d=5;
31 byte e=6;
32 byte f=7;
33 byte g=8;
34 //Declaracion pines de Tbj
35 byte t1=9; //miles
36 byte t2=10; //centenas
37 byte t3=11; //decenas
38 byte t4=12; //unidades
39 //Contadores para hacer el temporizador
40 long tempor=0000, contret=0;
41
42 //Funcion que coloca en el puerto de salida los
    bits comenzando
43 //desde el pin de ini hasta el pin fin
44
45 void puerto(int bits, int ini, int fin){
46   for(int i=ini; i<=fin; i++){
47     {
48       digitalWrite(i, bitRead(bits, i-ini));
49     }
50   }
51 //Funcion encargada de la multiplexacion de los
    displays
52 void mostrar()
53 {
54   int dig[4];
55   //digito millar
56   dig[0]=tempor/1000;
57   //digito centena
58   dig[1]=(tempor-dig[0]*1000)/100;
59   //digito Decena
60   dig[2]=(tempor-dig[0]*1000-dig[1]*100)/10;
61   //digito Unidad
62   dig[3]=(tempor-dig[0]*1000-dig[1]*100-dig
    [2]*10);
63
64   //Rutina de multiplexacion

```

```

65   for(int i=t1; i<=t4; i++){
66     puerto(display7c[dig[i-t1]], a, g);
67     digitalWrite(i, HIGH); //Enciende el display
    de unidades
68     delay(1);
69     digitalWrite(i, LOW); //Apaga el display de
    unidades
70   }
71 }
72
73 void temporizacion() //lazo de temporizacion
    de retardo
74 {
75   contret=50; //Variable de control de
    velocidad
76   while (contret>0)
77   {
78     mostrar();
79     contret--;
80   }
81 }
82 int a0 = 0, a1 = 0, b0 = 0, b1 = 0; //Variables
    auxiliares
83 char num = 'a'; //Control de estado de operacion
84 int num = 0, numb = 0, oper=-1; //Numero A, B y
    operacion
85 void setup(){
86   //config 8 pines digitales como salida
87   //Serial.begin(9600);
88   for(int i=a; i<=t4; i++){
89     pinMode(i, OUTPUT);
90   }
91 }
92 //Funcion para realizar las operaciones
93 void resultado(int op){
94   int res = 0;
95   //Seleccionamos la operacion de lo que se
    escogio en el teclado
96   switch(op){
97     case 1:
98       tempor = num + numb;
99       break;
100    case 2:
101      res = num - numb;
102      if(res<0){ //Si el resultado es negativo se
        muestra como positivo
103        tempor = -1*res;
104      }else{tempor = res;}
105      break;
106    case 3:
107      tempor = num * numb;
108      break;
109    case 4:
110      if(numb != 0){ //Se impide division por cero
        tempor = num / numb;
111      }else{tempor = 0;}
112      break;
113   }
114 }
115 }
116 void loop() {
117   mostrar();
118   //Se espera una tecla
119   char key = keypad.getKey();
120   if (key != NO_KEY){ //Si se presiona una tecla
    ingresa
121     // Serial.println(key);
122     if(num == 'c'){key = 'A';} //Si existe
    resultado reinicia
123     switch (key){
124       case 'A': //Caso AC reinicia todos los
        valores y operaciones
125         tempor = 0;
126         num = 0;
127         numb = 0;

```

```

128     num = 'a';
129     a0 = 0;
130     a1 = a0;
131     b0 = a1;
132     b1 = b0;
133     oper = 0;
134     break;
135     // Casos que eligen la operacion y cambio
136     al numero B
137     case '+':
138     num = 'b';
139     oper = 1;
140     break;
141     case '-':
142     num = 'b';
143     oper = 2;
144     break;
145     case '/':
146     num = 'b';
147     oper = 4;
148     break;
149     case '*':
150     num = 'b';
151     oper = 3;
152     break;
153     case '='://Llamo a la funcion para obtener
154     resultado
155     num = 'c';
156     resultado(oper);
157     break;
158     default:
159     if(num == 'a'){
160     //Desplazo los numeros hacia la
161     izquierda
162     a1 = a0;
163     a0 = (int)key-48;//Transformo a entero
164     correspondiente
165     numa = 10*a1+a0;
166     tempor = numa;
167     }
168     if(num == 'b'){
169     //Manejo de numero B
170     b1 = b0;
171     b0 = (int)key-48;//Transformo a entero
172     correspondiente
173     numb = 10*b1+b0;
174     tempor = 100*numb+numa;
175     }
176     }
177 }

```

Script 1. Código de la simulacion de una calculadora basica

B. Escribir ventajas y desventajas del uso de barrido de displays vs el uso de displays individuales.

• Ventajas:

- Para el barrido de displays se necesita únicamente una función que controle la multiplexación, y con esta es posible mostrar diferentes números de manera sencilla.
- Mediante el uso de barrido de displays se puede mostrar más números que sólo hasta 15 hexadecimal como sucede cuando se usa un solo display.
- Se puede realizar la creación de palabras completas de forma simultánea, lo que no sucede con un solo display.
- Menor uso de pines de la placa Arduino UNO.

- Mayor libertad y capacidad para la construcción de programas complejos.

• Desventajas:

- Mayor complejidad al momento de realizar la codificación del programa o función que controlará a los displays.
- El uso de memoria al momento de usar barrido de displays es mayor que cuando se usa displays individuales.
- La configuración de una frecuencia a la cual el cambio entre displays no se distinga es de vital importancia para una correcta implementación.
- El usuario debe poseer una alta capacidad para comprender multiplexación.

C. Escribir ventajas y desventajas del uso de barrido de teclado vs el uso de pulsadores individuales.

• Ventajas:

- Menor cantidad de pines utilizados, debido al arreglo interno de los pulsadores; el teclado matricial únicamente se tiene que ocupar 8 pines de la placa Arduino y gracias al barrido por filas y columnas se pueden tener 16 opciones al momento de presionar una tecla.
- Frecuencia de rebote menor.
- No se requiere hacer uso de un protoboard o algún dispositivo para conectar los pulsadores porque ya vienen las conexiones realizadas en un solo encapsulado.
- Se puede ingresar diferentes tipos de caracteres o números con la pulsación de una tecla de teclado matricial, por otra parte, al usar pulsadores únicamente se leería un HIGH o LOW, y se deberá hacer una lógica para asignar un valor dependiendo el pin.

• Desventajas:

- Al momento de realizar conexiones equivocadas, las pulsaciones serán detectadas de forma equivocada.
- Se deben exponer librerías adicionales para el uso de barrido de teclado.
- Cantidad de código más extensa que la usada para detectar a un pulsador.

D. Describir detalladamente 3 aplicaciones del uso de Barrido de teclado y de display.

- Máquina dispensadora de golosinas: Se requiere de un teclado y de al menos 4 displays, donde se colocaría el número de la celda que contiene a la golosina a través del teclado, la cual sería una combinación de números y letras, posteriormente de ingresar el número se mostrará el precio a pagar por ella y luego de que se haya ejecutado el pago se activará un servomotor que empujará la golosina de su posición hacia un lugar donde el consumidor tiene que tomarla.
- Reloj digital: Haciendo uso de 4 displays es posible mostrar todas las horas y minutos; podría realizarse la

construcción de un reloj aún más preciso mediante el uso de 6 displays, de esta manera se mostrarán los segundos, minutos y horas. En el caso de que el usuario desee, se podría emplear una la emisión de una alarma mediante el uso de una bocina o piezo el cual se activará cuando se emita una señal que indique se que ha llegado a un valor en específico.

- Ascensor: Esto lo podrá implementar a través de un teclado y dos displays en el caso en que el edificio tenga más de 10 pisos. Podría usarse dos LEDs para indicar si el ascensor esta subiendo o bajando o a su vez un display de 16 segmentos. Además, si se desea que la placa se haga cargo del motor que sube y baja al ascensor se requería aislar la placa del dispositivo que controla directamente el motor por medio de un acoplamiento óptico, y claramente se requeriría de mayor potencia para poder utilizar todas las funcionalidades requeridas.

E. Conclusiones:

Jonathan Álvarez

- El barrido de displays aprovecha la frecuencia con la que puede percibir el ojo humano para mostrar simultáneamente varios números prendiendo y apagando los displays.
- Se permite el ahorro en la circuiteria de control para los displays y la complejidad del circuito a implementar en caso de que se conectara por separado los displays
- Podemos ligar el teclado a interrupciones para tener respuestas rápidas en nuestro circuito.

Melanny Dávila

- El uso de barrido de displays permite reducir el número de componentes para desplegar información que junto con el uso y a su vez el tiempo de utilización de diferentes dispositivos periféricos permiten la creación de proyectos mucho más realistas y complejos.
- Se puede realizar el barrido de displays de una manera adecuada, debido a que los LEDs de los displays se encienden y apagan a velocidades que el ojo humano no puede detectar, por lo que aparentemente el display se encuentra siempre encendido.
- El uso de interrupciones es de gran utilidad debido que permiten simular aplicaciones en tiempo real, debido a que este comando se ejecuta cuando un parámetro de los dispositivos usados dentro del circuito cambie. Lo que hará que el código sea más apto para la recepción y muestra de datos.

F. Recomendaciones:

Jonathan Álvarez

- Revisar que los displays se encuentren conectados correctamente para la implementación ya sea en cátodo común o en ánodo común
- Usar una frecuencia optima para que los cambios en el barrido de displays no sean notables

Melanny Dávila

- Se recomienda dividir el código en funciones, de esta manera será más fácil modificarlo o encontrar un error si fuera el caso.
- Verificar la inclusión de las librerías necesarias para el uso correcto de los dispositivos implementados en el desarrollo de los circuitos.
- Se recomienda revisar la polarización en la que se encuentran los displays, ya que en base a esto se realizara la conexión y codificación de los mismos.

REFERENCES

- [1] "Teclado Matricial con ARDUINO- [4x4, 4x3 y Cualquier TECLADO]". <https://controlautomaticoeducacion.com/arduino/teclado-matricial-keypad/> (accedido ene. 19, 2021).
- [2] E. Espinosa, E. Tatayo, "MANEJO DE PERIFÉRICOS DE E/S (Teclado Matricial y barrido de displays)". C.P. SISTEMAS EMBEBIDOS, Accedido: ene. 19, 2020. [En línea].
- [3] "Display multiple de 7 segmentos - Display multiple de 7 segmentos," Diarioelectronicohoy.com, 2020. [En línea]. Available: <https://www.diarioelectronicohoy.com/blog/display-multiple-de-7-segmentos..> [Último acceso: 19 Enero 2021].
- [4] "Teclados matriciales — Tienda y Tutoriales Arduino," Prometec.net, 2020. [En línea]. Available: <https://www.prometec.net/teclados-matriciales/>. [Último acceso: 18 Enero 2021].
- [5] "teclado matricial – Aprendiendo Arduino," Aprendiendo Arduino, 2020. [En línea]. Available: <https://aprendiendoarduino.wordpress.com/tag/teclado-matricial/>. [Último acceso: 20 Enero 2021].