

UNIVERSIDAD SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
LABORATORIO ARQUITECTURA DE
COMPUTADORAS Y ENSAMBLADORES 1
AUXILIAR: RONALD RODRIGO MARÍN SALAS



MANUAL TÉCNICO

Nombre

Oscar Fernando López Pirir

Brayan Hamllelo Estevem Prado Marroquín

Kelly Mischel Herrera Espino

Alberto Gabriel Reyes Ning

Eduardo René Agustin Mendoza

Carné:

201908376

201801369

201900716

201612174

201801627

INTRODUCCIÓN

En el siguiente manual técnico se detalla tanto la parte que se realizó en proteus como lo que se hizo en arduino. Para que el proyecto pudiera cumplir con los aspectos pedidos por el cliente. En la parte de proteus se utilizó un arduino mega, dos matrices con leds verdes, un driver y —. En el arduino se utilizaron varios métodos, también se hizo uso de la librería LedControl para poder encender la leds de una de las matrices.

Para el cambio de se realizó lo siguiente:

En el caso de tener una pulsación se cambia el estado de la variable dirección. En caso de ser 1 el valor se cambia la dirección. Y se le agrega de valor -1.

```
int reading = digitalRead(13);
t1 = millis();
if ((t1 - t0) >= 150) {
    t0 = millis();
    aumentar = aumentar + direccion;
}

if (reading != lastButtonState) {
    lastDebounceTime = millis();
}
if ((millis() - lastDebounceTime) > debounceDelay) {
    if (reading != buttonState) {
        buttonState = reading;
        if (buttonState == HIGH) {
            if (direccion == 1) {
                direccion = -1;
            } else {
                direccion = 1;
            }
        }
    }
}
lastButtonState = reading;
```

Menú principal

Para el menú se hizo una matriz la cual se llenó con ceros y unos, los unos se fueron colocando en donde se validaría que la led encendiera en esa posición.

```
void mostrarMenu() {
    //Mostrar menu con driver
    for (int i = 0; i < 8; i++) {
        for (int j = 0; j < 8; j++) {
            matriz.setLed(0, i, j, menu[i][j] == 1 ? true : false);
        }
    }

    //Mostrar menu sin driver
    for (int i = 0; i < 8; i++) {
        for (int j = 8; j < 16; j++) {
            int residuo = 80;

            if (menu[i][j] == 1) {
                pintarLED(j-8, i);
            }
        }
    }
}
```

```

void loop() {
  //Mostrar Menu
  if (state == 1) {
    mostrarMenu();
    int reading = digitalRead(12);
    if (reading != lastButtonState12) {
      lastDebounceTime12 = millis();
    }

    if ((millis() - lastDebounceTime12) > debounceDelay) {
      if (reading != buttonState12) {
        buttonState12 = reading;
        if (buttonState12 == HIGH) {
          state = 2;
          vida = vidaInicial;
        }
      }
    }
    lastButtonState12 = reading;
  } else if (state == 2) {
    iniciarJuego();
    int reading = digitalRead(10);
    if (reading != lastButtonState10) {
      lastDebounceTime10 = millis();
    }

    if ((millis() - lastDebounceTime10) > debounceDelay) {
      if (reading != buttonState10) {
        buttonState10 = reading;
        if (buttonState10 == HIGH) {
          state = 3;
        }
      }
    }
  }
}

```

Niveles

Tanto para los niveles y las vidas se hicieron matrices del 0 al 10.

```

int uno[8][16] = {
  { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
  { 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0 },
  { 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0 },
  { 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0 },
  { 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0 },
  { 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0 },
  { 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0 },
  { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }
};

```

```

//tres
> int tres[8][16] = { ...
};
//cuatro
> int cuatro[8][16] = { ...
};
//Cinco
> int cinco[8][16] = { ...
};
//seis
> int seis[8][16] = { ...
};
//siete
> int siete[8][16] = { ...
};
//ocho
> int ocho[8][16] = { ...
};
//nuevo
> int nueve[8][16] = { ...
};
//diez
> int diez[8][16] = { ...
};

```

Después de realizar las matrices se hicieron varios métodos para poder recorrer las matrices y así poder visualizarlas en proteus en una matriz de leds.

Ejemplo del método:

```
9 void mostrarUno() {
10     //Mostrar 1 con driver
11     for (int i = 0; i < 8; i++) {
12         for (int j = 0; j < 8; j++) {
13             matriz.setLed(0, i, j, uno[i][j] == 1 ? true : false);
14         }
15     }
16     //Mostrar 1 sin driver
17     for (int i = 0; i < 8; i++) {
18         for (int j = 8; j < 16; j++) {
19             int residuo = 80;
20
21             if (uno[i][j] == 1) {
22                 pintarLED(j-8, i);
23             }
24         }
25     }
26 }
27 //Metodo para mostra el numero dos
```

```
9 > void mostrarUno() { ...
6 }
7 //Metodo para mostra el numero dos
8 > void mostrarDos() { ...
5 }
6 //Metodo para motrar el tres
7 > void mostrarTres() { ...
4 }
5 //Metodo para motrar el cuatro
6 > void mostrarCuatro() { ...
3 }
4 //Metodo para motrar el cinco
5 > void mostrarCinco() { ...
2 }
3 //Metodo para motrar el seis
4 > void mostrarSeis() { ...
1 }
2 //Metodo para motrar el siete
3 > void mostrarSiete() { ...
0 }
1 //Metodo para motrar el ocho
2 > void mostrarOcho() { ...
9 }
0 //Metodo para motrar el nueve
1 > void mostrarNueve() { ...
8 }
9 //Metodo para motrar el diez
0 > void mostrarDiez() { ...
7 } // _____Mostrar Menu_____
```

Con una variable global con nombre nivel , dependiendo del valor que esta tuviera sería se llamaría al método que correspondía a dicho número. En la siguiente imagen se puede visualizar el método:

```
int nivel = 0;
void mostrarNiveles() {
    if (nivel == 0) {
        mostrarCero();
    } else if (nivel == 1) {
        mostrarUno();
    } else if (nivel == 2) {
        mostrarDos();
    } else if (nivel == 3) {
        mostrarTres();
    } else if (nivel == 4) {
        mostrarCuatro();
    } else if (nivel == 5) {
        mostrarCinco();
    } else if (nivel == 6) {
        mostrarSeis();
    } else if (nivel == 7) {
        mostrarSiete();
    } else if (nivel == 8) {
        mostrarOcho();
    } else if (nivel == 9) {
        mostrarNueve();
    } else if (nivel == 10) {
        mostrarDiez();
    }
}
```

Vidas

En el caso de las vidas se usaron los métodos de los números mencionados anteriormente. Se creó un método para mostrar las vidas, verificando el valor que tomaba la variable global vida y así poder mostrar el numero dependiendo del valor.

```
void mostrarVidas() {
    if (vida == 0) {
        if (t >= 0 && t < 20) {
            mostrarCero();
            t++;
        } else if (t >= 20 && t < 30) {
            mostrarMatriz();
            t++;
        } else {
            mostrarMenu();
            matriz.clearDisplay(0);
        }
    } else if (vida == 1) {
        mostrarUno();
    } else if (vida == 2) {
        mostrarDos();
    } else if (vida == 3) {
        mostrarTres();
    } else if (vida == 4) {
        mostrarCuatro();
    } else if (vida == 5) {
        mostrarCinco();
    } else if (vida == 6) {
        mostrarSeis();
    } else if (vida == 7) {
        mostrarSiete();
    }
}
```

Edificios

Se creó la siguiente matriz para poder saber el tamaño que tendrá el edificio y en qué posición se encontrará.

```
//{ x, y }  
int edificios[16][2] = { { 0, 0 },  
                          { 1, 3 },  
                          { 2, 3 },  
                          { 3, 0 },  
                          { 4, 0 },  
                          { 5, 0 },  
                          { 6, 0 },  
                          { 7, 0 },  
                          { 8, 4 },  
                          { 9, 0 },  
                          { 10, 0 },  
                          { 11, 0 },  
                          { 12, 0 },  
                          { 13, 0 },  
                          { 14, 0 },  
                          { 15, 0 },  
                        };
```

```
void borrarEdificio(int x) {  
    int y = edificios[x][1];  
    for(int i = 0; i < y; i++){  
        buffer[7-i][x] = 0;  
    }  
    edificios[x][1] = 0;  
}  
  
void crearedificios() {  
    //CON DRIVER  
    for(int k = 0; k < 16; k++){  
        if (edificios[k][1] != 0){  
            for(int i = 0; i < edificios[k][1]; i++){  
                buffer[7 - i][edificios[k][0]] = 1;  
            }  
        }  
    }  
}
```

Avión

Para el avión se crearon los siguientes métodos:

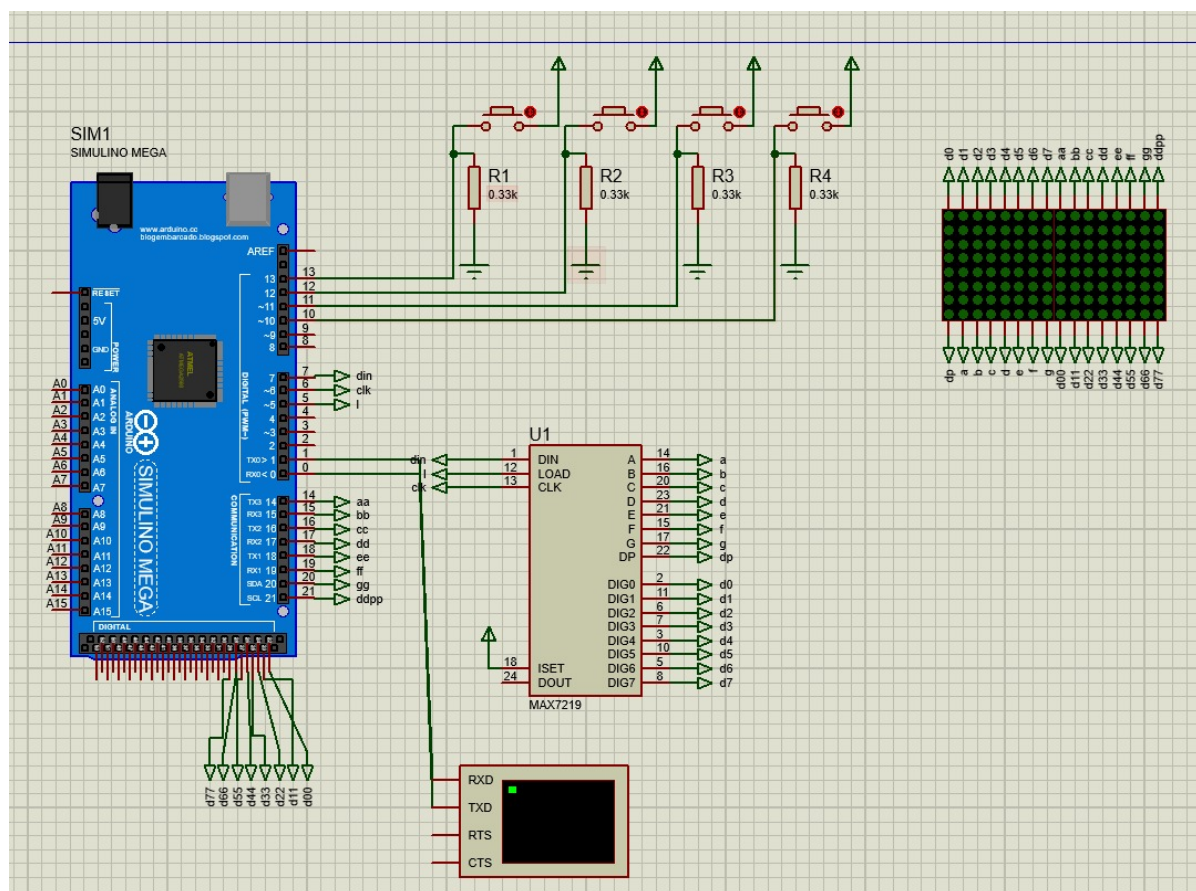

```

6
7 > void pintarAvion() { ...
23 }
24
25 > void borrarAvion() { ...
30 }
31
32 > void mina() { ...
38 }
39
40 > void moverMina() { ...
58 }
59
60 > void mostrarMatriz() { ...
73 }

```

Configuración

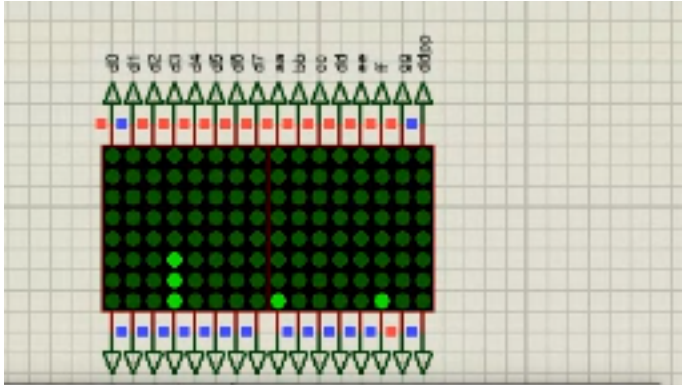
Parte en proteus



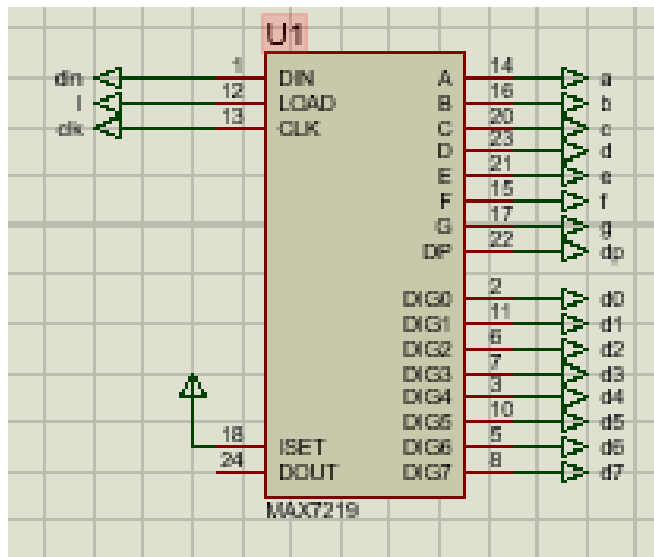
Componentes Utilizados:

- Dos matrices de leds verdes
- Driver
- Arduino mega
- 4 resistencias
- 4 botones

Matrices



Driver



Arduino Mega

