José Roberto Rodriguez Reyes 21060
Cindy Michelle Gualim Pérez 21226

PROYECTO No. 3

Descripción de la resolución del temario asignado:

Para la resolución del temario asignado lo primero que se hizo fue inicializar los pines de lectura y escritura, posteriormente para cada led se crearon los pines de salida y lectura para poder cambiar el estado de las leds entre encendido y apagado, después se creó una función llamada, “esperandoBoton”, en la cual utilizando los pines de lectura se detecta si el botón ha sido presionado y si lo ha sido se inicia el programa también, se creó la clase “esperandoTecla”, en el cual de manera similar, se ve si ha sido presionada o no la tecla “y” y al igual que con el botón, si ha sido presionado se inicia el programa. Posteriormente se realizó la función de “identificarEstado”, en el que se compara el pin del jugador, ya sea 1 o 2, utilizando el comando cmp, con los valores 1, 2, 3, 4, 5, 6, 7, 8, posteriormente se implementó la función “print1”, en la cual utilizando los pines de salida creados previamente y la función de identificarEstado, se encendió el led que le toca respectivamente al jugador 1, y se implementó la función “print2”, que funciona de exactamente la misma manera, pero para el jugador 2. Finalmente se implemento una espera de 1.5 segundos para cada led y haciendo uso de las funciones creadas previamente, se llaman en cada led avanzando el “juego”, hasta que un jugador se encuentre en el último estado y el juego llegue a su final.

Especificación del uso de los registros y puertos GPIO utilizados:

```
215 myPinmodes:
216     // Inicializar los pines de salida
217     mov r0, #0 // salidas
218     mov r1, #1
219     bl pinMode
220
221     mov r0, #1
222     mov r1, #1
223     bl pinMode
224
225     mov r0, #2
226     mov r1, #1
227     bl pinMode
228
229     mov r0, #3
230     mov r1, #1
231     bl pinMode
232
233     mov r0, #4
234     mov r1, #1
235     bl pinMode
236
237     mov r0, #5
238     mov r1, #1
239     bl pinMode
240
241     mov r0, #6
242     mov r1, #1
243     bl pinMode
244
245     mov r0, #7
246     mov r1, #1
247     bl pinMode
248
249     mov r0, #21 // entradas
250     mov r1, #0
251     bl pinMode
```

```
mov r0, #21 // entradas
mov r1, #0
bl pinMode

mov r0, #25
mov r1, #0
bl pinMode

mov r0, #22 // salida jugador 1
mov r1, #1
bl pinMode

mov r0, #23 // salida jugador 2
mov r1, #1
bl pinMode

b esperandoTecla
```

Los GPIOS utilizados son del 0 al 7, 21,25,22 y 23 tal y como se muestra en las imágenes anteriores , con el manejo de comandos que se muestra en la imagen 1.2

Imagen 1.2:

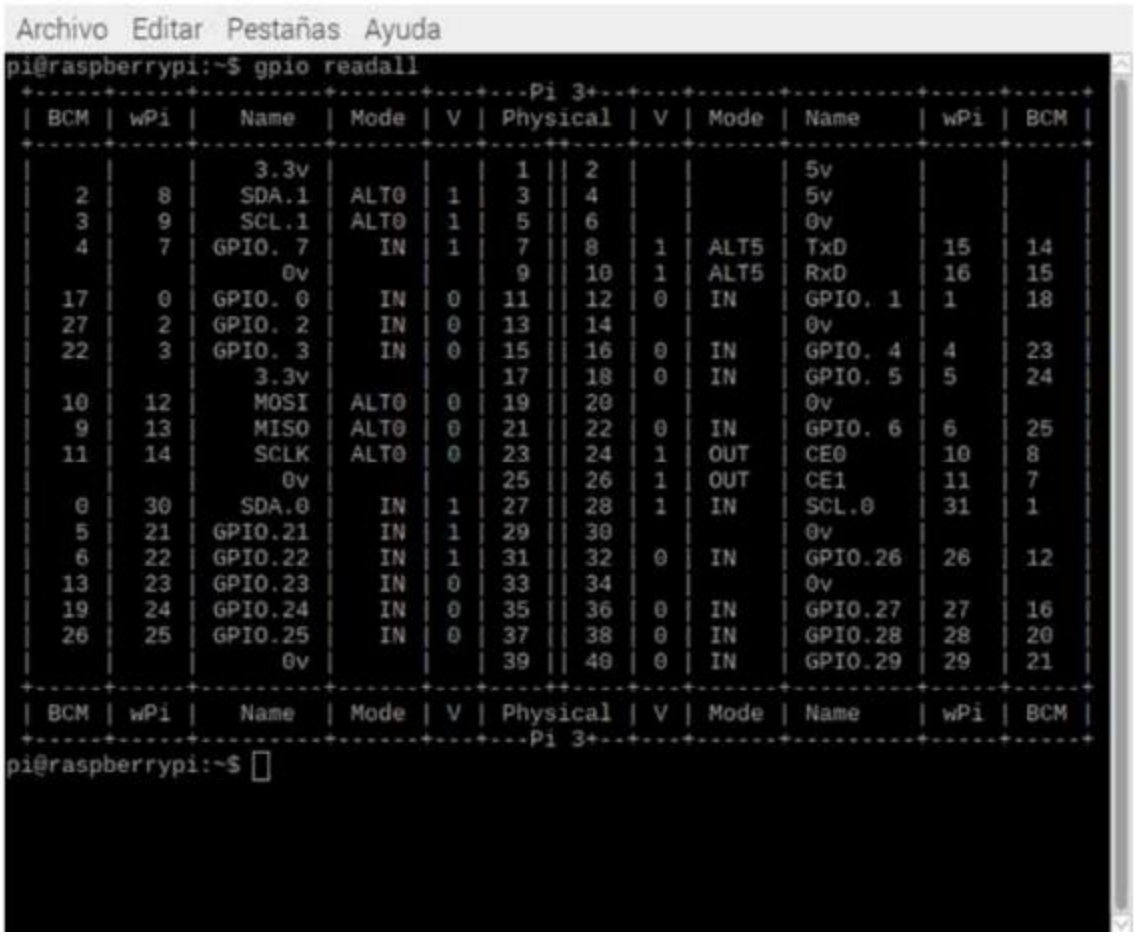


Diagrama de flujo del programa hecho en una herramienta de software o algoritmo narrativo:

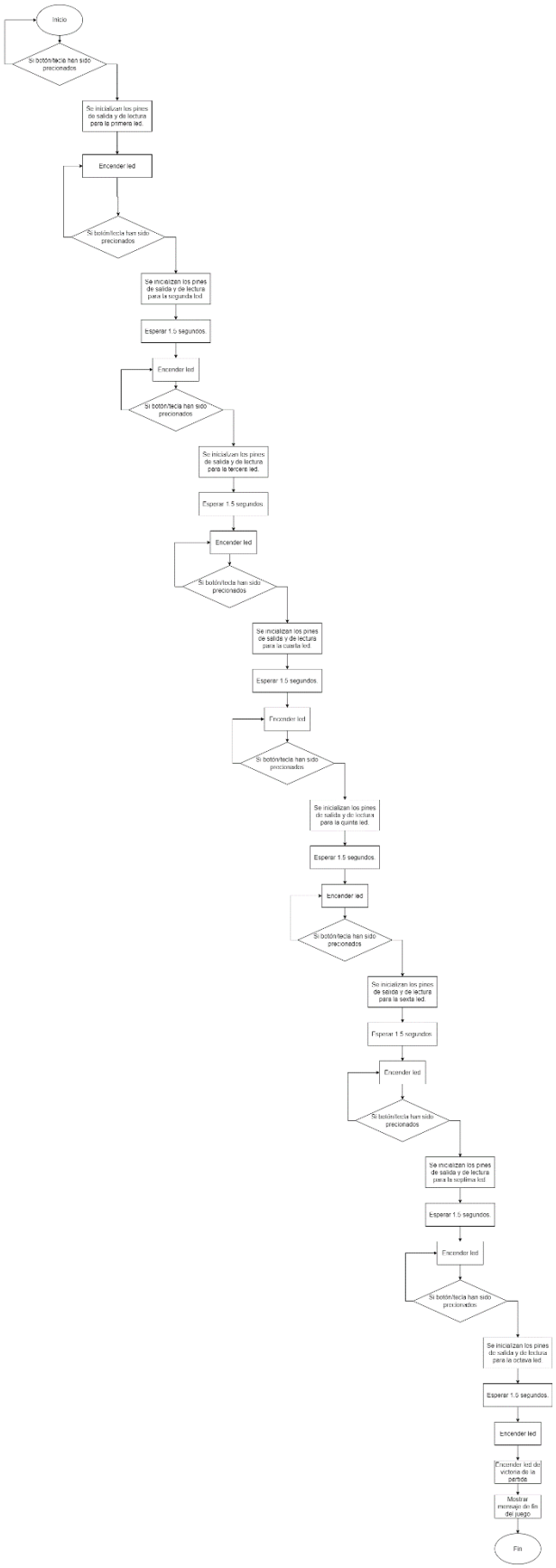
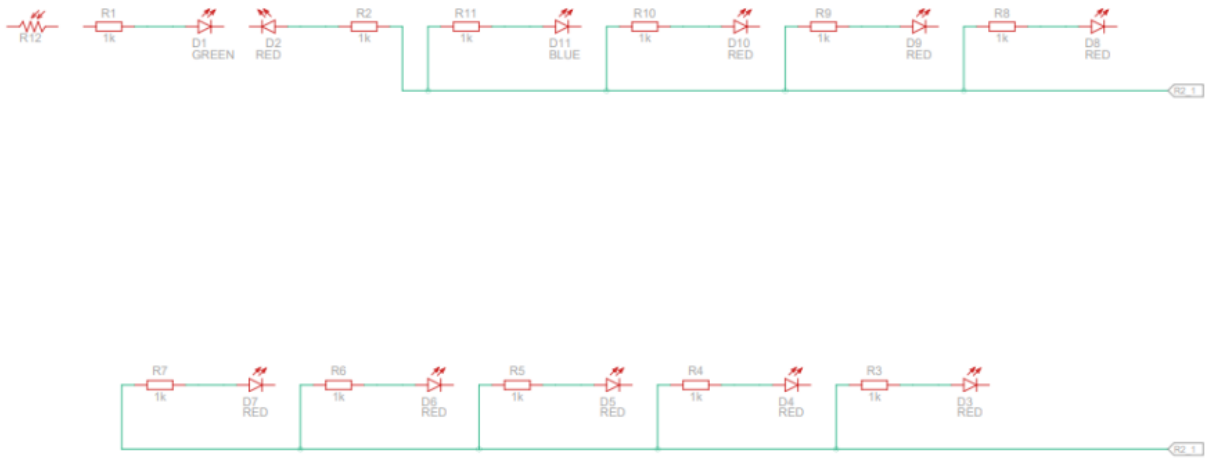

Inicio
Si botón/tecla han sido precionados
Se inicializan los pines de salida y de lectura para la primera led.
Encender led
Si botón/tecla han sido precionados
Se inicializan los pines de salida y de lectura para la segunda led
Esperar 1.5 segundos.
Encender led
Si botón/tecla han sido precionados
Se inicializan los pines de salida y de lectura para la tercera led.
Esperar 1.5 segundos.
Encender led
Si botón/tecla han sido precionados
Se inicializan los pines de salida y de lectura para la cuarta led.
Esperar 1.5 segundos.
Encender led
Si botón/tecla han sido precionados
Se inicializan los pines de salida y de lectura para la quinta led.
Esperar 1.5 segundos.
Encender led
Si botón/tecla han sido precionados
Se inicializan los pines de salida y de lectura para la sexta led.
Esperar 1.5 segundos.
Encender led
Si botón/tecla han sido precionados
Se inicializan los pines de salida y de lectura para la septima led
Esperar 1.5 segundos.
Encender led
Si botón/tecla han sido precionados
Se inicializan los pines de salida y de lectura para la octava led.
Esperar 1.5 segundos.
Encender led
Encender led del victoria de la partida
Mostrar mensaje de fin del juego
Fin

Diagrama del circuito:



Conclusiones:

- El uso adecuado de los puertos GPIO es importante para el buen funcionamiento del programa.
- La planeación lógica para la implementación del programa es importante para el orden y buen funcionamiento de este.
- El armado adecuado del circuito es tan importante como el funcionamiento del programa en software.

Bibliografía:

Paul A. Carter. (2007). Lenguaje Ensamblador para PC. 2022, de - Sitio web: http://pacman128.github.io/static/pcasm-book-spanish.pdf

Simon Humphreys. (2022). Assembly Language on the Pi: “Learning how to walk again”. 2022, de HelloWorld Sitio web: https://helloworld.raspberrypi.org/articles/hw16-assembly-language-on-the-pi