

Máquina Simon

Angelo Maximilian Tulbure y
Cayetana Lloris Rodríguez

05/06/2024

Objetivo



Control de LEDs:

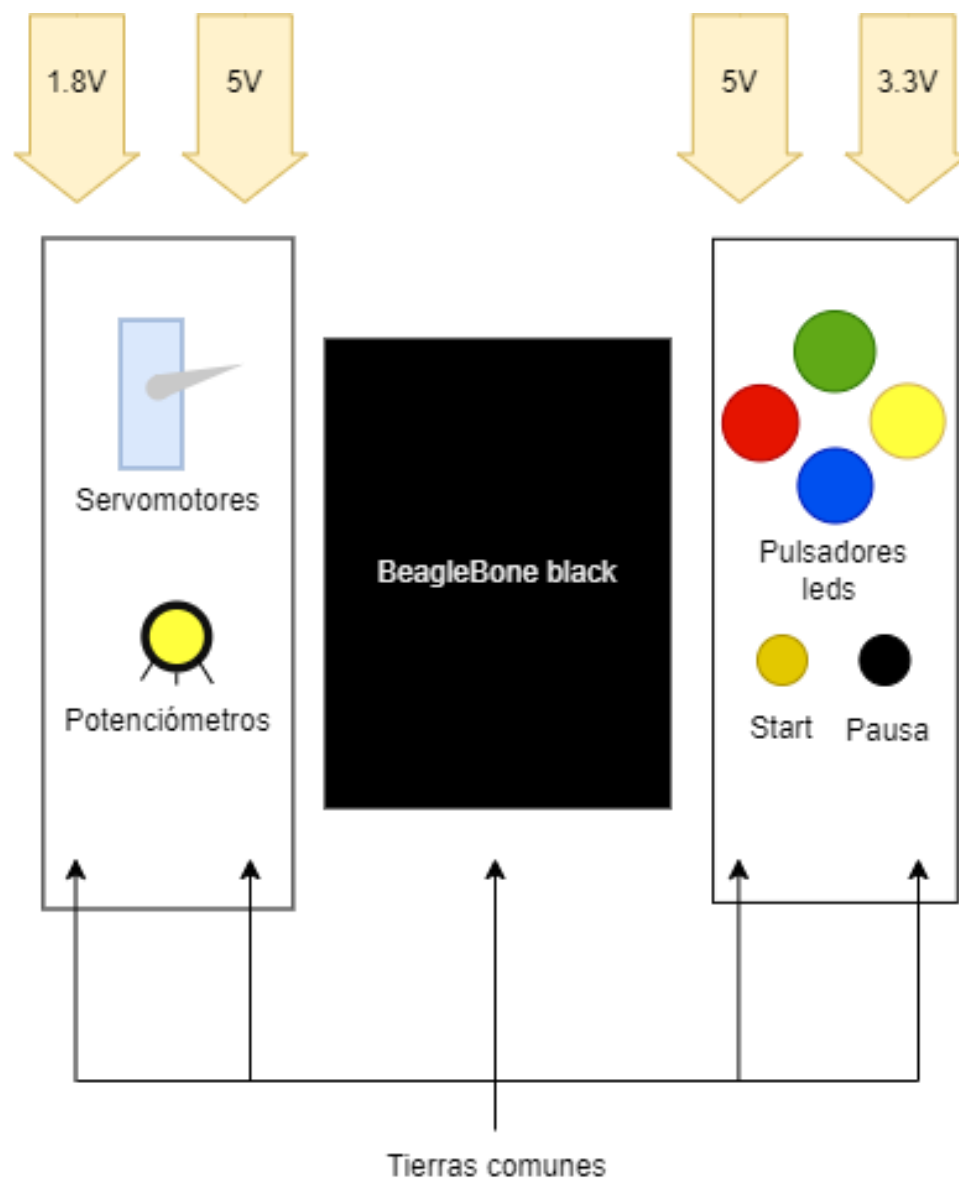
Mostrar secuencias con diferentes niveles de dificultad.



Monitoreo de Estados:

Estados - INICIO, MOSTRAR, INSERTAR, PAUSA. Transiciones de estado basadas en pulsaciones de botones.

Descripción del Sistema



Conexiones LEDs y Botones

- `GPIOuniv outGPIO(GPIOuniv::P9_12); //GPIO60 1º led verde`
- `GPIOuniv outGPIO1(GPIOuniv::P9_23); //GPIO49 2º led amarillo`
- `GPIOuniv outGPIO2(GPIOuniv::P9_30); //GPIO112 3º led rojo`
- `GPIOuniv outGPIO3(GPIOuniv::P9_27); //GPIO115 4º led azul`

- `GPIOuniv inGPIO(GPIOuniv::P9_15); //GPIO48 1º botón verde`
- `GPIOuniv inGPIO1(GPIOuniv::P9_41); //GPIO20 2º botón amarillo`
- `GPIOuniv inGPIO2(GPIOuniv::P8_16); //GPIO46 3º botón rojo`
- `GPIOuniv inGPIO3(GPIOuniv::P8_18); //GPIO65 4º botón azul`

- `GPIOuniv startGPIO(GPIOuniv::P8_12); //GPIO44 Start button`
- `GPIOuniv pauseGPIO(GPIOuniv::P8_14); //GPIO26 PAUSA button`

Conexiones Potenciómetros y Servos:

- PWM `pwm1(PWM::P9_14);` // Servo 1: P9_14 - DIFICULTAD
- PWM `pwm2(PWM::P9_22);` // Servo 2: P9_22 - TIMEOUT
- ADC `adcCH0(0);` // Potentiometer 1: P9_39 - DIFICULTAD
- ADC `adcCH1(1);` // Potentiometer 2: P9_40 - TIMEOUT

Estructura del Código

Archivos:

- main.cpp
- ThreadConf.cpp
- ThreadConf.h
- PotentiometerServo.cpp
- PotentiometerServo.h
- StateMonitor.cpp
- StateMonitor.h

Configuración de Hilos (ThreadConf.cpp/h)

Descripción: Gestiona el uso de hilos y tiempos para mostrar secuencias y obtener la entrada del jugador.

Funciones Clave:

- `void thread_start();`
- `void thread_pause();`
- `void thread_principal();`
- `void thread_player();`
- `void thread_sound();`
- `void show_sequence();`
- `int calculate_timeout(int pot_value, int num_intervals);`
- `int calculate_speed_delay(int pot_value, int num_intervals);`

Control de Potenciómetros y Servos (PotentiometerServo.cpp/h)

Descripción: Maneja la lectura de valores de potenciómetros y ajusta el comportamiento.

- PWM `pwm1(PWM::P9_14); // Servo 1: P9_14 - DIFICULTAD`
- PWM `pwm2(PWM::P9_22); // Servo 2: P9_22 - TIMEOUT`
- ADC `adcCH0(0); // Potentiometer 1: P9_39 - DIFICULTAD`
- ADC `adcCH1(1); // Potentiometer 2: P9_40 - TIMEOUT`

Funcion Clave:

- `void control_servo()`

Monitoreo de Estados (StateMonitor.cpp/h)

Descripción: Monitorea y maneja los diferentes estados del sistema.

- `GPIOuniv outGPIO(GPIOuniv::P9_12); // GPIO60 1º led verde`
- `GPIOuniv outGPIO1(GPIOuniv::P9_23); // GPIO49 2º led amarillo`
- `GPIOuniv outGPIO2(GPIOuniv::P9_30); // GPIO112 3º led rojo`
- `GPIOuniv outGPIO3(GPIOuniv::P9_27); // GPIO115 4º led azul`
- `GPIOuniv inGPIO(GPIOuniv::P9_15); // GPIO48 1º botón verde`
- `GPIOuniv inGPIO1(GPIOuniv::P9_41); // GPIO20 2º botón amarillo`
- `GPIOuniv inGPIO2(GPIOuniv::P8_16); // GPIO46 3º botón rojo`
- `GPIOuniv inGPIO3(GPIOuniv::P8_18); // GPIO65 4º botón azul`
- `GPIOuniv startGPIO(GPIOuniv::P8_12); // GPIO44 Start button`
- `GPIOuniv pauseGPIO(GPIOuniv::P8_14); // GPIO26 PAUSA button`

Funciones Clave:

- `void setupGPIO();`
- `void check_buttons();`
- `int calculate_pot_level(int pot_value, int num_intervals);`

Funcionalidad Principal (main.cpp)

Descripción: Punto de entrada principal de la aplicación.

Funciones Clave:

- Inicialización de componentes
- Bucle principal que maneja las transiciones de los estados START, SHOW, INSERT, PAUSE.

Desafíos y Soluciones

- **Selección de componentes para la parte del hardware.**
- **Riesgo de sobrecarga en la placa. La solución es medir el voltaje frecuentemente y tener tierras comunes**
- **Implementación de la máquina de estados. Uso de mecanismos de demora y hilos adecuados, Implementación de funciones robustas de verificación de estados.**



Gracias por
vuestra atención

¿Alguna pregunta?