Guía de Trabajo de Laboratorio

Tema: Enclavamiento y Desenclavamiento con PPDuino Nano

# 🎯 Objetivos

• Comprender el concepto de enclavamiento electrónico mediante el uso de un pulsador y un LED.

• Aplicar un algoritmo de control en el módulo educativo PPDuino Nano.

• Reconocer la importancia del anti-rebote en sistemas digitales.

• Relacionar el uso del pulsador externo como dispositivo de entrada con el control de salidas.

# 📦 Materiales requeridos

• 1 módulo educativo PPDuino Nano

• 1 pulsador externo

• 1 resistencia de 10 kΩ (solo si no se usa la resistencia interna de pull-up)

• Cables de conexión (jumpers)

• Computadora con Arduino IDE instalado

• Cable USB para conexión al PPDuino Nano

# 📖 Conceptos previos

• Enclavamiento: un sistema que mantiene su estado (encendido/apagado) hasta que reciba una nueva orden.

• Toggle: cada pulsación del pulsador alterna el estado de la salida.

• Anti-rebote: evita lecturas falsas cuando se presiona un botón mecánico.

• PPDuino Nano: módulo educativo con componentes integrados (LEDs, resistencias, salidas), donde el estudiante se enfoca en el algoritmo.

# ⚙️ Circuito a implementar

- LED integrado en el pin 9 del PPDuino Nano (no requiere montaje externo).  
- Pulsador externo conectado entre el pin 17 y GND.  
- El programa usará la resistencia de pull-up interna del microcontrolador.

# 💻 Programa en Arduino IDE

const int pinPulsador = 17; // Pulsador externo  
const int pinSalida = 9; // LED integrado en el PPDuino Nano  
  
unsigned long lastDebounceTime = 0;  
const unsigned long debounceDelay = 50; // Tiempo de anti-rebote (ms)  
  
int lastButtonState = HIGH;   
int buttonState = HIGH;  
bool salidaLatched = false; // Estado actual del LED  
  
void setup() {  
 pinMode(pinPulsador, INPUT\_PULLUP);  
 pinMode(pinSalida, OUTPUT);  
 digitalWrite(pinSalida, LOW);  
}  
  
void loop() {  
 int reading = digitalRead(pinPulsador);  
  
 if (reading != lastButtonState) {  
 lastDebounceTime = millis();  
 }  
  
 if ((millis() - lastDebounceTime) > debounceDelay) {  
 if (reading != buttonState) {  
 buttonState = reading;  
 if (buttonState == LOW) {  
 salidaLatched = !salidaLatched;  
 digitalWrite(pinSalida, salidaLatched ? HIGH : LOW);  
 }  
 }  
 }  
  
 lastButtonState = reading;  
}

# 🧪 Procedimiento

• Conectar el pulsador externo entre el pin 17 del PPDuino Nano y GND.

• Conectar la placa al computador mediante el cable USB.

• Abrir Arduino IDE y copiar el código del programa.

• Compilar y cargar el programa en el PPDuino Nano.

• Presionar el pulsador y observar el comportamiento del LED en el pin 9: Una pulsación lo enciende, la siguiente lo apaga, y así sucesivamente.

# 📚 Actividades de análisis

• Dibujar el diagrama lógico del sistema (entrada: pulsador, salida: LED).

• Identificar en el código la línea que genera el enclavamiento (salidaLatched = !salidaLatched;).

• Explicar por qué se usa la instrucción INPUT\_PULLUP en el pulsador.

• Analizar qué pasaría si no se aplicara anti-rebote.

• Investigar cómo se implementa el enclavamiento en circuitos eléctricos industriales (ej. contactores con botones de marcha y paro).

• Proponer una modificación al código para incluir un segundo pulsador de RESET que apague el LED.

# 📝 Conclusiones

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_