

Curso de Arduino: Nivel III - Robótica Móvil y Control Autónomo

Nivel 3: Precisión y Autonomía

Domina el tiempo y el movimiento

La verdadera magia ocurre cuando un robot puede hacer varias cosas a la vez con precisión milimétrica. En este nivel, nos adentramos en el corazón del control robótico: el tiempo real y el movimiento coordinado. Aprenderás el secreto profesional de `millis()` para eliminar las pausas torpes, y usarás interrupciones para crear respuestas instantáneas. Programarás motores que giran grados exactos, construirás un seguidor de líneas que navega por sí mismo y diseñarás sistemas que "recuerdan" cuándo activarse. Este es el salto de hacer proyectos a crear sistemas autónomos.

A manera de Introducción:

Este Nivel III convierte a los estudiantes en diseñadores de sistemas robóticos completos, enfocándose en el problema clásico del seguidor de líneas, pero con profundidad técnica suficiente para participar en competencias de robótica educativa. El uso exclusivo de Arduino Nano garantiza que las soluciones sean transferibles a proyectos personales posteriores.

Duración: 20 horas (10 sesiones de 2 horas, continuación del Nivel II)

Enfoque: Robótica móvil con Arduino Nano - Del seguidor de líneas básico al control PID
Fundamento Pedagógico

- **Prerrequisito:** Nivel II completo o conocimientos equivalentes
- **Plataforma:** Arduino Nano (compatible con UNO pero compacto)
- **Material principal por equipo:**
 - 2x Motorreductores TT con ruedas
 - 2x Driver L298N (o módulo dual)
 - Chasis robótico (kit básico o fabricado)
 - 3-5x Sensores infrarrojos TCRT5000/QTR-8
 - 2x Motores paso a paso 28BYJ-48 + drivers ULN2003
 - Batería 9V o power bank
 - Caster o rueda loca
- **Metodología:** Diseño iterativo, prueba-error controlado, ingeniería de sistemas

Planificación Semanal Nivel III

Semana 1: Fundamentos de Robótica Móvil

Tiempo: 2 horas

Objetivo: Construir plataforma robótica básica y control manual

Contenido:

- Anatomía de un robot móvil (15 min)
- Selección de chasis y distribución de masa (20 min)
- Especificaciones motorreductores TT (15 min)

Práctica 24: Plataforma robótica básica

- Ensamblaje mecánico del chasis (40 min)
- Conexión motorreductores a L298N (25 min)
- Configuración pines Arduino Nano (15 min)
- Reto: Verificar giro correcto de ambos motores

Práctica 25: Control manual por Bluetooth/Serial

- Programación de movimientos básicos (25 min)
 - “avanzar()”, “retroceder()”, “girarDerecha()”, “girarIzquierda()”
- Control por comandos de teclado (20 min)
- Reto: Control proporcional con teclas WASD

Semana 2: Motores Paso a Paso y Precisión

Tiempo: 2 horas

Objetivo: Dominar motores 28BYJ-48 para aplicaciones de posicionamiento

Contenido:

- Teoría de motores paso a paso unipolares (25 min)
- Secuencias de excitación (wave, full-step, half-step) (20 min)
- Driver ULN2003 y consumo energético (15 min)

Práctica 26: Brazo o mecanismo de precisión

- Montaje de 2 motores paso a paso (30 min)
- Librería <Stepper.h> y personalización (25 min)
- Control de ángulos exactos (0-360°) (20 min)
- Reto: Dibujar formas geométricas con puntero

Práctica 27: Robot omnidireccional (concepto)

- Configurar 3 motores paso a paso a 120° (25 min)
- Simulación matemática de movimientos (20 min)
- Reto: Mover robot en cualquier dirección sin rotar

Semana 3: Sensado de Línea Avanzado

Tiempo: 2 horas

Objetivo: Implementar arrays de sensores y calibración automática

Contenido:

- Tipos de sensores IR (reflectivos vs. emisor-receptor) (20 min)
- Teoría de muestreo y promediado (15 min)
- Calibración automática de sensores (25 min)

Práctica 28: Array de 3 sensores TCRT5000

- Distribución óptima en chasis (30 min)
- Circuito con resistencias de pull-up (20 min)
- Lectura digital y umbral adaptable (25 min)
- Reto: Detectar cruces en intersecciones

Práctica 29: Seguidor de línea básico (2 sensores)

- Lógica de seguimiento (P: proporcional simple) (35 min)
 - “if(sensorIzq) girarIzquierda()”
 - “if(sensorDer) girarDerecha()”
- Ajuste fino de velocidad base (15 min)
- Reto: Recorrer circuito simple sin perder línea

Semana 4: Algoritmos de Búsqueda y Decisión

Tiempo: 2 horas

Objetivo: Implementar lógica para laberintos e intersecciones

Contenido:

- Estados finitos para robótica (20 min)

- Algoritmo de la mano derecha/izquierda (25 min)
- Memoria de decisiones (15 min)

Práctica 30: Robot laberinto básico

- Detección de intersecciones en T y + (40 min)
- Implementación de giros memorizados (30 min)
- Salida de laberinto simple (20 min)
- Reto: Mostrar camino en monitor serial

Práctica 31: Seguidor con 5 sensores

- Montaje array compacto (25 min)
- Mapeo de lectura: "[0,0,1,0,0]" = centro (30 min)
- Mejora de precisión en curvas (15 min)
- Reto: Distinguir entre línea negra y borde de mesa

Semana 5: Introducción al Control PID

Tiempo: 2 horas

Objetivo: Comprender y aplicar control proporcional-integral-derivativo básico

Contenido:

- Analogía del PID (conducción de auto) (25 min)
- Término P (proporcional): error actual (20 min)
- Términos I y D (conceptos básicos) (15 min)

Práctica 32: Seguidor P (Proporcional)

- Cálculo de error con 5 sensores (30 min)
 - "error = posición_sensor - posición_deseada"
- Velocidad proporcional al error (25 min)
 - "velMotorIzq = velocidadBase + Kp * error"
- Ajuste de constante Kp (20 min)
- Reto: Encontrar Kp óptimo para circuito

Práctica 33: Visualización de error

- Envío de datos a Processing/Plotter serial (25 min)
- Gráfica en tiempo real de error y corrección (20 min)
- Reto: Guardar datos para análisis posterior

Semana 6: Control PID Completo

Tiempo: 2 horas

Objetivo: Implementar PID completo y sintonización

Contenido:

- Término I (integral): corrección de error acumulado (25 min)
- Término D (derivativo): predicción de error futuro (20 min)
- Método de sintonización Ziegler-Nichols (simplificado) (15 min)

Práctica 34: PID completo para seguimiento

- Implementación de las 3 componentes (40 min)

error = calcularError();

P = Kp * error;

I = Ki * (errorAcumulado);

D = Kd * (error - errorAnterior);

correccion = P + I + D;

- Ajuste manual de constantes (35 min)
- Reto: Competencia de tiempo en circuito

Práctica 35: Anti-windup básico

- Límite al término integral (25 min)
- Reset de integral en situaciones especiales (20 min)
- Reto: Mejorar estabilidad en rectas largas

Semana 7: Sistemas Híbridos Motorreductor + Paso a Paso

Tiempo: 2 horas

Objetivo: Integrar diferentes actuadores en un mismo robot

Contenido:

- Selección de actuador por tarea (20 min)
- Sincronización de múltiples controladores (25 min)
- Gestión de potencia y recursos (15 min)

Práctica 36: Robot con brazo manipulador

- Motorreductores para movimiento base (30 min)
- Motor paso a paso para brazo elevador (25 min)
- Coordinación para recoger objetos (30 min)
- Reto: Apilar 2 objetos en zona específica

Práctica 37: Seguidor de línea con cabezal orientable

- Cabezal con sensor IR en motor paso a paso (35 min)
- Búsqueda activa de línea perdida (25 min)
- Reto: Recuperarse después de perder línea 180°

Semana 8: Proyecto Intermedio - Robot Competitivo

Tiempo: 2 horas

Objetivo: Diseñar robot optimizado para circuito estándar

Contenido:

- Análisis de circuitos típicos (25 min)
- Estrategias de carrera (20 min)
- Optimización de constantes PID (15 min)

Práctica 38: Tuning por circuito

- Prueba en 3 tipos de curvas (40 min)
 - Curva suave (radio grande)
 - Curva cerrada (90°)
 - Esquina aguda
- Ajuste dinámico de parámetros (35 min)
- Reto: Menor tiempo sin salidas de pista

Práctica 39: Sistema de inicio/parada

- Detección de línea de inicio/meta (25 min)
- Temporizador preciso con `millis()` (20 min)
- Reto: Diferentes modos (prueba, competencia)

Semana 9: Proyecto Final - Diseño y Prototipado

Tiempo: 2 horas

Objetivo: Planificar robot seguidor de líneas avanzado

Contenido:

- Definición de especificaciones técnicas (30 min)
- Diseño modular del sistema (40 min)
- Plan de pruebas y validación (30 min)

Opciones de Proyecto Final:

1. Robot velocista: Optimizado para velocidad en circuitos conocidos
2. Robot explorador: Mapea circuito y elige ruta óptima
3. Robot de rescate: Sigue línea y detecta "víctimas" (LEDs)
4. Robot sumo seguidor: Sigue línea pero empuja oponentes fuera
5. Robot artista: Dibuja patrones mientras sigue líneas

Fases de Implementación:

1. Módulo sensado: 5-8 sensores con calibración automática
2. Módulo control: PID con constantes adaptativas
3. Módulo actuación: Control de velocidad diferencial preciso
4. Módulo decisión: Lógica para intersecciones complejas
5. Módulo interfaz: LED indicadores, serial debugging

Semana 10: Proyecto Final - Competencia y Evaluación

Tiempo: 2 horas

Objetivo: Prueba final en circuito estándar y presentación

Contenido:

- Preparación final y ajustes (30 min)
- Competencia en 3 circuitos de dificultad creciente (50 min)
- Presentación técnica del diseño (30 min)
- Retroalimentación y premiación (30 min)

Circuitos de Competencia:

1. Circuito básico: Óvalo simple con 4 curvas
2. Circuito intermedio: Incorpora intersecciones en T
3. Circuito avanzado: Incluye cruces en + y curvas S

Criterios de Evaluación:

- Tiempo de completado
- Número de salidas de pista
- Consistencia en múltiples vueltas
- Innovación en diseño/solución

Estructura Pedagógica Nivel III

Metodologías Específicas:

1. Aprendizaje basado en retos: Cada sesión tiene meta concreta
2. Ingeniería inversa: Analizar robots existentes antes de diseñar
3. Prototipado rápido: Construir-testear-iterar en ciclos cortos
4. Documentación técnica: Diagramas de conexión, flujogramas, tablas de constantes
5. Trabajo en equipo especializado: Roles definidos (hardware, software, testing)

Material Didáctico Adicional:

- Plantillas de chasis para impresión/corte (opcional)
- Circuitos estándar impresos en papel A1

- Kit de calibración (cinta negra, fondos blancos/grises)
- Instrumentos de medición (cronómetro, flexómetro)
- Estaciones de carga para baterías

Manejo de Dificultades Comunes:

1. Problemas de calibración: Procedimiento paso a paso estandarizado
2. Variabilidad en componentes: Margen de tolerancia documentado
3. Interferencia entre sensores: Distanciamiento y shielding básico
4. Ruido en lecturas: Filtro software de promedio móvil
5. Consistencia en superficies: Uso de materiales estandarizados

Evaluación Nivel III

Distribución de Puntaje:

- 50%: Proyecto final (funcionalidad, velocidad, robustez)
- 20%: Portafolio de prácticas (evolución del robot)
- 15%: Participación en competencias semanales
- 15%: Documentación técnica completa

Rúbrica de Proyecto Final:

1. Diseño mecánico (20pts): Estabilidad, distribución de peso, accesibilidad
2. Sistema sensorial (25pts): Precisión, calibración, inmunidad a ruido
3. Algoritmo de control (30pts): Eficacia del PID, manejo de casos límite
4. Rendimiento (15pts): Velocidad, consistencia, recuperación de errores
5. Innovación (10pts): Soluciones creativas a problemas identificados

Competencias Evaluadas:

- Pensamiento algorítmico
- Resolución de problemas técnicos
- Trabajo colaborativo
- Adaptabilidad a nuevas situaciones
- Capacidad de análisis y optimización

Recursos y Herramientas Especializadas

Software Complementario:

1. PID Tuner: Interfaz gráfica para ajuste de constantes
2. Robot Simulator: Simulación básica antes de implementación física
3. Data Logger: Captura y análisis de datos de sensores
4. Circuit Designer: Diseño de pistas personalizadas

Kit de Desarrollo Avanzado (opcional):

- Encoders rotacionales para odometría
- Sensor IMU (acelerómetro/giroscopio)
- Cámara simple para visión básica
- Módulo radiofrecuencia para control simultáneo múltiple
- Display OLED para telemetría en tiempo real

Organización del Taller:

1. Zona de construcción: Mesas amplias con herramientas
2. Zona de pruebas: Circuitos permanentes montados

3. Zona de diagnóstico: Equipos de medición y computadoras
4. Zona de competición: Circuito oficial para pruebas finales