Un cruce con tiempos de paso visibles

Un programa de fases coordina dos semáforos e informa del tiempo restante de paso.

RESPONSABLE

Daniel Serrano Carrasco

ÁREA / CURSO

DPIOT / 2026-2027

EL RECORRIDO DE LA DECISIÓN

01
ESP32

→

02
2 semáforos LED

PREGUNTA DE DISEÑO

¿Qué estados deben excluirse entre sí?

CRITERIO DE DECISIÓN

La coherencia del cruce exige que las dos direcciones no obtengan verde simultáneamente.

ENSAYO PROPUESTO

Recorrer un ciclo completo y revisar cuenta atrás y dos destellos amarillos.

Memoria de planteamiento. Las conexiones y los importes son propuestas de trabajo; la construcción y sus resultados requieren comprobación en el aula.

Mapa del documento

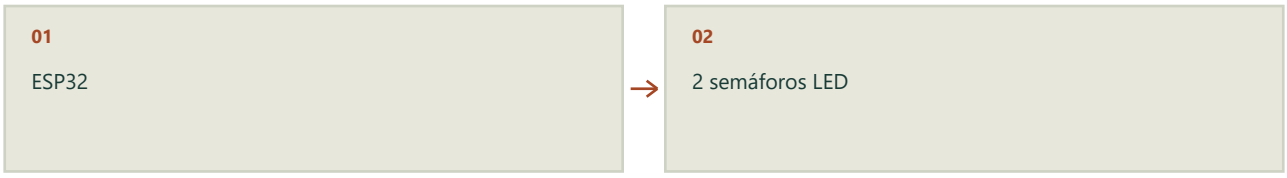
El texto sigue la decisión del sistema antes de detallar su construcción. Las pruebas incluidas son criterios de trabajo: no se presentan como ensayos ya superados.

01 / Finalidad

Representar con ESP32 una regulación de cruce que combine rojo, verde y amarillo con un contador para las fases temporizadas.

02 / Cómo se conectan las funciones

El ESP32 establece la secuencia luminosa y envía al visualizador el tiempo que falta para el cambio.



Salida complementaria: Visualización Pantalla Muestra la cuenta atrás

03 / Recorrido de una decisión

1. Rojo + cuenta atrás
2. Verde + cuenta atrás
3. Amarillo parpadea 2 veces
4. Volver a rojo
5. Repetir secuencia

04 / Recursos y estimación de coste

Elementos previstos

- 1 ESP32.
- 6 LED: 2 rojos, 2 amarillos y 2 verdes.
- Resistencias para los LED.
- 1 pantalla para la cuenta atrás.
- Protoboard y cables.
- Material para construir la maqueta y la carretera.

Estimación inicial de recursos

Cantidad	Recurso previsto	Estimación inicial
1	ESP32	8 €
6	LED + resistencias	4 €
1	Pantalla	6 €
1	Protoboard y cableado	7 €
1	Material para maqueta	10 €

Total estimado: 35 €

05 / Propuesta y límites del montaje

Los dos grupos se coordinan para que no coincidan sus verdes. Cada recorrido pasa por rojo, verde, amarillo y rojo; el rojo y el verde disponen de cuenta atrás, y el cierre de verde da lugar a dos destellos amarillos.

Incluido en esta versión

- Control automático de dos semáforos.
- Temporización de las luces roja y verde.
- Parpadeo de la luz amarilla dos veces.
- Visualización de la cuenta atrás en una pantalla.

Aspectos que no cubre esta versión

- Detección de vehículos.
- Pulsador para peatones.
- Control remoto mediante móvil o Internet.

06 / Qué permite aprender

La aplicación permite razonar sobre estados, duración de cada fase y exclusión de pasos incompatibles. Los LED y la pantalla hacen observable la secuencia.

07 / Condiciones de funcionamiento

Comportamientos esperados

1. Encender la luz roja durante el tiempo programado.
2. Pasar de rojo a verde y mostrar su cuenta atrás.
3. Cerrar la fase verde con exactamente dos destellos del amarillo.
4. Volver al rojo y repetir automáticamente la secuencia.

5. Excluir el estado en el que las dos direcciones obtienen verde simultáneamente.

Condiciones de construcción

- Programación mediante Arduino IDE para ESP32.
- Alimentación de baja tensión.
- Montaje inicial en protoboard.

08 / Conexiones propuestas: pendientes de verificar

Elemento	Interfaz	Conexión propuesta	Criterio de montaje
Semáforo 1 - rojo	Digital	GPIO 18	LED rojo
Semáforo 1 - amarillo	Digital	GPIO 19	LED amarillo
Semáforo 1 - verde	Digital	GPIO 21	LED verde
Semáforo 2	Digital	GPIO 25, 26, 27	Rojo, amarillo y verde
Pantalla I2C	SDA / SCL	GPIO 32 / 33	Cuenta atrás

09 / Secuencia de desarrollo

Etapa	Trabajo previsto	Dedicación orientativa	Situación prevista
1. Anteproyecto	Documento y presupuesto	1 sesión	Planificado: sin validación física
2. Montaje	Montar semáforos y pantalla	2 sesiones	Planificado: sin validación física
3. Firmware	Programar secuencia y tiempos	2 sesiones	Planificado: sin validación física
4. Pruebas	Comprobar luces y temporizadores	1 sesión	Planificado: sin validación física
5. Memoria	Fotos, código y conclusiones	1 sesión	Planificado: sin validación física

10 / Qué tendría que demostrar el prototipo

- Los tiempos mostrados corresponden a las fases roja y verde en curso.
- El cambio de rojo a verde se realiza automáticamente.
- La transición posterior al verde contiene los dos destellos previstos.

- Después del amarillo, el sistema vuelve al rojo.
- Los dos semáforos permanecen correctamente coordinados.

11 / Puntos que deben revisarse antes de alimentar

Punto a revisar	Qué puede ocurrir	Comprobación preventiva
Secuencia incorrecta	Conflicto entre semáforos	Comprobar los estados antes de las pruebas.
Temporización incorrecta	Funcionamiento irregular	Ajustar los tiempos en el programa.
Conexión incorrecta de LED	LED dañado o sin luz	Usar resistencias y revisar polaridad.

Validación y procedencia

Esta hoja separa lo que el diseño propone de lo que deberá demostrarse en la maqueta. Las casillas de situación no acreditan pruebas realizadas.

Comprobación	Evidencia que debe incorporarse	Situación
Conexiones y alimentación	Esquema contrastado con los componentes utilizados	Pendiente de registrar
Programa del demostrador	Versión compilada y cargada en la placa	Pendiente de registrar
Comportamiento previsto	Tabla de entradas y respuestas observadas	Pendiente de registrar
Incidencias	Descripción del fallo y de la corrección aplicada	Pendiente de registrar
Cierre de la entrega	Fotografías y conclusiones de las pruebas	Pendiente de registrar

Referencias y alcance de esta memoria

Consigna de DPIOT: presentar diez anteproyectos básicos con título, objetivo, material y funcionamiento. Esta memoria desarrolla una de esas ideas y acompaña a su página web.

Ideas de partida: resumen facilitado del vídeo Top 10 Arduino Projects 2026, del canal Simple Circuits, y colección documental recuperada durante la preparación del sitio. La redacción de esta edición se corresponde con el desarrollo revisado de ORIGINAL_DANIEL.

Los importes, tiempos y conexiones son estimaciones o asignaciones de proyecto, no presupuestos comerciales actualizados ni un cableado validado. El programa y el montaje deben contrastarse con la placa y las instrucciones del aula.

La entrega final debe incorporar el programa realmente utilizado, las evidencias del montaje y los resultados observados. Los modelos de navegador explican las respuestas previstas y no sustituyen esas pruebas.