

Arduino y Microcontroladores en Supervivencia: Automatización de Sistemas

Autor: EA4IPV

Fecha: 23/03/2026

Categoría: Trampas Defensivas

Etiquetas: Sin etiquetas

Arduino y Microcontroladores en Supervivencia: Automatización de Sistemas

Un microcontrolador es un ordenador completo en un solo chip: procesador, memoria RAM, memoria flash para el programa y periféricos de entrada/salida. El Arduino Uno, basado en el ATmega328P de Atmel (ahora Microchip), opera a 16 MHz con 32 KB de flash, 2 KB de SRAM y 1 KB de EEPROM, consumiendo apenas 45 mA a 5 V (225 mW). Esta combinación de bajo consumo, robustez y facilidad de programación lo convierte en el cerebro ideal para automatizar sistemas de supervivencia: reguladores de carga solar, controladores de riego, estaciones meteorológicas, alarmas perimetrales y temporizadores de cualquier tipo. Alternativas como el ESP32 (240 MHz, WiFi y Bluetooth integrados, 520 KB SRAM) amplían las posibilidades a comunicación inalámbrica entre puestos. Este artículo cubre la selección del microcontrolador adecuado, la alimentación desde baterías de 12 V, la conexión de sensores y actuadores reales, y ejemplos concretos de automatización en un refugio.

Selección del microcontrolador y alimentación desde 12 V

La elección del microcontrolador depende de tres factores: número de entradas/salidas necesarias, consumo energético y necesidad de comunicación inalámbrica. Para la mayoría de automatizaciones en un refugio, el Arduino Nano (mismo chip ATmega328P, formato compacto de 18×45 mm) o el ESP32 DevKit son las mejores opciones.

Parámetro

Arduino Nano (ATmega328P)

ESP32 DevKit

ATtiny85 (mínimo)

Frecuencia de reloj

16 MHz

240 MHz (dual core)

8 MHz (interno)

Flash / SRAM

32 KB / 2 KB

4 MB / 520 KB

8 KB / 512 bytes

Pines digitales I/O

14 (6 PWM)

34 (16 PWM)

6 (3 PWM)

Entradas analógicas (ADC)

8 canales, 10 bits

18 canales, 12 bits

4 canales, 10 bits

Consumo en operación

19-45 mA a 5 V

80-240 mA a 3,3 V

1-9 mA a 5 V

Consumo en deep sleep

~5 mA (con LED)

10 μ A

0,1 μ A

WiFi / Bluetooth

No

WiFi 802.11 b/g/n + BLE 4.2

No

Precio aproximado

3-5 €

5-8 €

1-2 €

Voltaje de operación

5 V (regulador a bordo acepta 7-12 V)

3,3 V (regulador acepta 5 V por USB)

2,7-5,5 V

Para alimentar un Arduino Nano desde una batería de 12 V se usa el pin VIN, que alimenta el regulador lineal AMS1117-5.0 integrado. Este regulador disipa la diferencia de voltaje como calor: $P = (12 - 5) \times 0,045 = 0,315 \text{ W}$, aceptable para el encapsulado SOT-223. Para el ESP32, que opera a 3,3 V, es mejor usar un convertidor buck DC-DC (como el MP1584 o LM2596) que baje de 12 V a 5 V con eficiencia del 90-96 %, y luego alimentar por el pin USB o el regulador de 3,3 V de la placa.

Regulador buck MP1584EN: Módulo miniatura (22×17 mm) que acepta 4,5-28 V de entrada y entrega 0,8-20 V ajustables a hasta 3 A. Eficiencia del 96 % a 2 A. Ideal para alimentar cualquier microcontrolador desde baterías de 12 V o paneles solares. Ajustar la salida a 5,0 V con el potenciómetro antes de conectar el Arduino.

Protección contra inversión de polaridad: Colocar un diodo Schottky SS54 (5 A, 40 V, caída de 0,5 V) en

serie con la alimentación positiva, o un MOSFET de canal P (IRF9540N) para protección sin caída de voltaje. Una inversión de polaridad accidental destruye instantáneamente el regulador y el microcontrolador.

Consumo ultra-bajo con deep sleep: El ESP32 en modo deep sleep consume solo 10 μ A. Se programa para despertar cada 5 minutos (o por interrupción de un sensor), leer datos, actuar si es necesario y volver a dormir. Con una batería de 12 V / 7 Ah, un ESP32 en este modo puede funcionar más de un año sin recarga.

Niveles lógicos 3,3 V vs 5 V: El ESP32 opera a 3,3 V y sus pines NO toleran 5 V. Conectar directamente un sensor de 5 V destruirá el pin. Usar un divisor resistivo (10 k Ω + 20 k Ω) o un level shifter bidireccional basado en BSS138 (4 canales por menos de 1 €) para comunicar dispositivos de 5 V con el ESP32.

Sensores prácticos para automatización de refugio

Los sensores son los ojos y oídos del microcontrolador. En un contexto de supervivencia, los más útiles miden temperatura, humedad, nivel de agua, luz ambiental, presencia humana y voltaje de baterías. Todos se conectan a los pines del Arduino o ESP32 con cables de 2-3 hilos.

Sensor
Magnitud
Rango
Precisión
Interfaz
Consumo

DS18B20
Temperatura
-55 a +125 °C
 $\pm 0,5$ °C
1-Wire (1 pin digital)
1 mA

DHT22 (AM2302)
Temperatura + humedad
-40 a +80 °C / 0-100 % HR
 $\pm 0,5$ °C / ± 2 % HR
Digital propietario (1 pin)
1,5 mA

BMP280
Presión + temperatura
300-1.100 hPa / -40 a +85 °C
 ± 1 hPa / ± 1 °C
I2C o SPI

2,7 μ A (modo forzado)

HC-SR04

Distancia (nivel de agua)

2-400 cm

± 3 mm

2 pines digitales (Trig/Echo)

15 mA

HC-SR501 (PIR)

Movimiento (infrarrojo)

3-7 m, ángulo 120°

Presencia/ausencia

1 pin digital (alto/bajo)

65 μ A en reposo

LDR (GL5528)

Luz ambiental

10 Ω (luz) - 1 M Ω (oscuridad)

Relativa

Analógica (divisor resistivo)

ACS712-20A

Corriente DC/AC

± 20 A

$\pm 1,5$ %

Analógica (66 mV/A)

10 mA

Divisor resistivo 10:1

Voltaje de batería

0-50 V

± 1 % (depende de resistencias)

Analógica

Para medir el voltaje de una batería de 12 V con un Arduino (ADC de 10 bits, referencia de 5 V), se usa un divisor resistivo con $R1 = 47 \text{ k}\Omega$ y $R2 = 10 \text{ k}\Omega$. El voltaje en el pin analógico será $V_{\text{pin}} = V_{\text{bat}} \times R2 / (R1 + R2) = V_{\text{bat}} \times 10 / 57 = 2,1 \text{ V}$ para 12 V. La lectura del ADC será $2,1 / 5 \times 1023 = 430$. Para obtener el voltaje real: $V_{\text{bat}} = \text{lectura_ADC} \times 5 / 1023 \times (R1 + R2) / R2$. Usar resistencias de precisión del 1 % para minimizar el error.

Múltiples DS18B20 en un solo pin: El protocolo 1-Wire permite conectar hasta 100 sensores DS18B20 en paralelo en un único pin digital con una resistencia pull-up de 4,7 k Ω a 5 V. Cada sensor tiene una dirección ROM única de 64 bits grabada de fábrica. Ideal para monitorizar temperatura en múltiples puntos del refugio con un solo cable de tres hilos (VCC, datos, GND).

Sensor de nivel por ultrasonidos: El HC-SR04 mide distancia enviando un pulso ultrasónico de 40 kHz y midiendo el tiempo de eco. Montado boca abajo sobre un depósito de agua, calcula el nivel como: $\text{nivel_cm} = \text{profundidad_total} - (\text{tiempo_us} / 58)$. Precisión de ± 3 mm hasta 4 metros. No sumergir: montarlo en la tapa del depósito apuntando hacia abajo.

Actuadores y control de cargas

Los actuadores son los músculos del sistema: relés, MOSFETs, servomotores y bombas que el microcontrolador controla para actuar sobre el mundo físico. La regla fundamental es que el pin del microcontrolador NUNCA alimenta directamente una carga de potencia: un pin de Arduino entrega máximo 40 mA a 5 V (200 mW), totalmente insuficiente para un relé, motor o bomba.

Actuador

Voltaje/Corriente

Control desde Arduino

Aplicación en refugio

Módulo relé 5 V (SRD-05VDC)

250 VAC / 10 A o 30 VDC / 10 A

Pin digital → optoacoplador integrado

Encender/apagar bomba de agua, calefactor, iluminación

MOSFET IRLZ44N (logic level)

55 V / 47 A ($R_{ds_on} = 22 \text{ m}\Omega$)

Pin digital directo al gate (umbral 1-2 V)

PWM de iluminación LED, ventiladores, calefactores DC

Servomotor SG90

5 V / 200 mA

Pin PWM (pulso 1-2 ms cada 20 ms)

Válvulas de agua, orientación de panel solar

Bomba peristáltica 12 V

12 V / 300 mA

MOSFET + diodo flyback 1N4007

Dosificación de nutrientes, transferencia de líquidos

Electroválvula solenoide 12 V

12 V / 500 mA

MOSFET + diodo flyback 1N4007

Apertura/cierre de paso de agua

El MOSFET IRLZ44N es el componente clave para controlar cargas DC desde un microcontrolador. Es un MOSFET «logic level», lo que significa que su gate se activa completamente con los 5 V de un pin de

Arduino (a diferencia de un IRF540N estándar que necesita 10 V en el gate). Conectar: gate al pin digital (con resistencia pull-down de 10 k Ω a GND para evitar activación espuria), source a GND y drain a la carga. La carga se conecta entre +12 V y drain.

Diodo flyback obligatorio: Toda carga inductiva (relé, motor, solenoide, bomba) DEBE tener un diodo flyback 1N4007 conectado en antiparalelo (cátodo a +V, ánodo a drain del MOSFET). Al cortar la corriente de una bobina, esta genera un pico de voltaje inverso de cientos de voltios (Lenz) que destruye el MOSFET instantáneamente. El diodo absorbe este pico. Sin diodo, el MOSFET se destruirá en la primera o segunda conmutación.

Proyectos concretos de automatización

Con los componentes descritos se pueden construir sistemas completos de automatización para un refugio. Los tres proyectos más útiles son: regulador de carga solar con monitorización, sistema de riego automático y estación meteorológica con registro de datos.

Regulador de carga solar básico: Arduino Nano + MOSFET IRLZ44N + divisor resistivo para medir voltaje de batería + sensor ACS712 para medir corriente. El Arduino mide el voltaje de la batería cada 10 segundos. Si supera 14,4 V (batería de plomo-ácido llena), corta la carga solar desactivando el MOSFET. Si baja de 13,8 V, reconecta. Si baja de 11,8 V (batería al 20 %), desconecta las cargas de consumo con un segundo MOSFET para proteger la batería. Coste total: menos de 10 €.

Riego automático por humedad del suelo: Sensor capacitivo de humedad del suelo (no resistivo, que se corroe) conectado a entrada analógica del Arduino. Umbral calibrado: si la lectura analógica supera 600 (suelo seco), activar la bomba peristáltica 12 V durante 30 segundos a través de un MOSFET. Esperar 5 minutos para que el agua percole y volver a medir. Incluir un sensor de nivel en el depósito de agua (HC-SR04) para no activar la bomba si el depósito está vacío.

Estación meteorológica con registro en SD: ESP32 + BMP280 (presión y temperatura) + DHT22 (humedad) + anemómetro de cazoletas (interrupción digital, cada pulso = una revolución). Los datos se registran cada 10 minutos en una tarjeta microSD vía módulo SPI. El ESP32 permite consulta WiFi desde cualquier dispositivo móvil a través de un servidor web embebido en el puerto 80. La tendencia barométrica (subida o bajada de presión en las últimas 3 horas) permite predecir cambios meteorológicos a 12-24 horas vista.

Monitor de baterías con alarma: ATtiny85 programado con Arduino IDE (requiere programador USBasp o Arduino como ISP). Mide el voltaje de un banco de baterías mediante divisor resistivo cada 30 segundos. Si el voltaje baja del umbral crítico (11,5 V para 12 V, 23 V para 24 V), activa un buzzer piezoeléctrico y un LED rojo. Consumo total del sistema: 2 mA, lo que permite funcionar con una pila de 9 V durante semanas.

⚠ Advertencia: Esta información es orientativa y educativa. En situaciones de emergencia real, consulte a profesionales cualificados siempre que sea posible. No ponga en riesgo su vida ni la de otros sin la formación adecuada.