Curso Completo: Sistemas IoT Industriales con Arduino y Raspberry Pi (Avanzado)

Video 01: Modulo 01: De Sistemas Embebidos a IoT Arquitectura y Niveles Logicos 5V vs 3.3V (Modulo 01 Arquitectura

1. SINTESIS TECNICA Y APUNTES CLAVE DE INGENIERIA

* Por que nunca conectar un pin de 5 V de Arduino directo al GPIO de 3.3 V de la Raspberry P
  i.
* Calculo de divisor resistivo (1.8 kOhm / 3.3 kOhm) y desplazador de nivel con MOSFET BSS
  138 para I2C.
* Etapa de potencia inductiva con MOSFET IRLZ44N y diodo de rueda libre (Flyback) contra p
  icos dI/dt.
* Analisis de consumo energetico y autonomia en baterias con perfiles de modo activo y sueno
  profundo.

2. ARQUITECTURA DE HARDWARE Y PROTOCOLO DE LABORATORIO

Paso 1: Verificar alimentacion electrica, tierra comun (GND) y niveles logicos antes de energizar.

Paso 2: Montar el esquema segun los planos de alta resolucion incluidos en el paquete .ZIP.

Paso 3: Compilar y cargar el firmware determinista (.ino) y ejecutar el script de supervision (.py).

Paso 4: Registrar las mediciones experimentales y validar el funcionamiento sin bloqueos.

3. CODIGO FUENTE DE REFERENCIA (C++ / PYTHON)

```
// ================================================================================
// CURSO: Curso Completo: Sistemas IoT Industriales con Arduino y Raspberry Pi (Avanzado)
// VIDEO 01: Modulo 01: De Sistemas Embebidos a IoT   Arquitectura y Niveles Logicos 5V vs 3.3V
// ================================================================================
#include <Arduino.h>

const unsigned long INTERVALO_MUESTREO_MS = 20;
unsigned long tUltimoMuestreo = 0;

void setup() {
  Serial.begin(115200);
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  unsigned long ahora = millis();
  if (ahora - tUltimoMuestreo >= INTERVALO_MUESTREO_MS) {
    tUltimoMuestreo = ahora;
    int lecturaAdc = analogRead(A0);
    float voltios = (lecturaAdc * 5.0f) / 1024.0f;
    Serial.print("{\"video\":\"01\",\"v_in\":");
    Serial.print(voltios, 3);
    Serial.println("}");
  }
}
```