

Anteproyecto

Sistema de Monitoreo Electrónico

para Vehículos

Agostini, Emiliano(autor)
Departamento de Ingeniería Electrónica,
Facultad Regional San Francisco, Córdoba, Argentina.
emilianoagostini14@gmail.com

Cortesse, Agustín(autor)
Departamento de Ingeniería Electrónica,
Facultad Regional San Francisco, Córdoba, Argentina.
agucortesse19@gmail.com



Índice General

1. Objetivo General	1
2. Introducción	1
3. Objetivos Específicos	1
4. Alternativas	2
4.1. Microcontrolador	2
4.2. Medición de Temperatura	4
4.3. Medición de velocidad	5
4.4. Métodos de medición del voltaje de la batería del automóvil	6
4.5. Medición de presión de aceite	6
4.6. Display LCD	7
5. REFERENCIAS	8



1. Objetivo General

Desarrollar un sistema electrónico capaz de medir y visualizar en tiempo real la temperatura del motor, presión del aceite, el nivel de carga de la batería y la velocidad del automóvil, mediante el uso de sensores y un display digital integrado, brindando una solución moderna para vehículos con tableros analógicos o defectuosos.

2. Introducción

En este anteproyecto se plantea el diseño e implementación de un sistema electrónico orientado a adquirir y mostrar en tiempo real parámetros clave del funcionamiento de un vehículo: la temperatura del motor, la presión del aceite, el nivel de carga de la batería y la velocidad. El sistema busca brindar una solución práctica a vehículos que no cuentan con un tablero funcional o que aún utilizan sistemas analógicos y mecánicos obsoletos.

La falta de información precisa y confiable en vehículos más antiguos puede comprometer tanto el mantenimiento como la seguridad del conductor. Para resolver esta problemática, se propone el uso de sensores digitales, un microcontrolador y un módulo de visualización, que en conjunto permitan una lectura continua, precisa y clara de los valores medidos.

Desde el punto de vista técnico, el sistema sustituye mecanismos tradicionales por una alternativa basada en tecnologías electrónicas modernas. Se integrarán sensores, un microcontrolador y una interfaz visual (display), lo que no solo mejora la experiencia del usuario, sino que también ofrece una aplicación real de conceptos fundamentales en la ingeniería electrónica, tales como la adquisición de datos, el procesamiento de señales y el desarrollo de sistemas embebidos.

Este proyecto tiene aplicaciones prácticas en la modernización de vehículos con equipamiento obsoleto, así como valor académico, ya que permite profundizar en instrumentación electrónica, programación de microcontroladores y diseño de interfaces hombre-máquina.

3. Objetivos Específicos

- Seleccionar e integrar sensores digitales adecuados para la medición de temperatura del motor, presión del aceite, voltaje de batería y velocidad del vehículo.
- Diseñar y construir el circuito electrónico necesario para acondicionar, procesar y digitalizar las señales provenientes de los sensores.



- Implementar un sistema de visualización digital que muestre en tiempo real los valores medidos mediante un display adecuado.
- Desarrollar el software embebido para adquisición de datos, procesamiento de señales y actualización del display.
- Realizar pruebas funcionales y validaciones en laboratorio o entorno controlado, verificando precisión, estabilidad y confiabilidad del sistema.

4. Alternativas

A lo largo del diseño se evaluaron diversas opciones para la implementación del sistema. A continuación, se presentan algunas alternativas consideradas para los diferentes componentes del circuito.

4.1. Microcontrolador

Característica	Memoria, potencia y núcleos		
	Arduino Nano	ESP32-WROOM-32	Raspberry Pi Pico
Arquitectura	AVR 8 bits	Xtensa LX6 32 bits	ARM Cortex-M0+ 32 bits
Núcleos	1	2	2
Frecuencia	16 MHz	hasta 240 MHz	133 MHz
Memoria Flash	32 KB	4 MB SPI	2 MB QSPI externa
SRAM	2 KB	520 KB	264 KB
EEPROM	1 KB	—	—
Voltaje de operación	5 V	3.0–3.6 V	1.8–5.5 V (I/O 3.3 V)
Consumo típico	19 mA	80 mA	100 mA

Cuadro 1: Comparación de memoria, núcleos y consumo



Entradas y Salidas			
Tipo de Pin	Arduino Nano	ESP32-WROOM-32	Raspberry Pi Pico
GPIO digitales	22 (5 V)	34 (3.3 V)	26 (3.3 V)
Entradas analógicas	8 (10 bits)	15 (12 bits)	3 (12 bits)
PWM	6	16 canales configurables	16 canales (8×2)
DAC	No	2	No

Cuadro 2: Comparación de pines de E/S

Conectividad			
Interfaz	Arduino Nano	ESP32-WROOM-32	Raspberry Pi Pico
USB	Micro-USB (Serial TTL)	Micro-USB / USB-C	USB-C (Host/Device)
Wi-Fi	No	Sí (802.11 b/g/n)	No
Bluetooth	No	Sí (v4.2 + BLE)	No
UART	1	3	2
SPI	1	2	2
I2C	1	2	2
I2S	No	2 canales	No
CAN	No	2 (TWAI)	No

Cuadro 3: Comparación de conectividad

Todas las placas pueden comunicarse mediante UART, SPI o I2C, aunque el nivel lógico debe ser adaptado: Arduino Nano trabaja a 5 V, mientras que ESP32 y Pico lo hacen a 3.3 V. No existe compatibilidad binaria entre códigos, aunque muchas librerías del entorno Arduino son adaptables entre plataformas.



4.2. Medición de Temperatura

Característica	LM35	DS18B20	NTC (Bulbo VW)
Tipo de sensor	Analógico lineal	Digital (1-Wire)	Termistor NTC (pasivo)
Rango de temperatura	-55 °C a +150 °C	-55 °C a +125 °C	-40 °C a +125 °C (aprox.)
Precisión típica	±0.5 °C (a 25 °C)	±0.5 °C (-10 a +85 °C)	±2 a 5 °C (depende del modelo)
Resolución	10 mV/°C	9–12 bits configurable	Depende del ADC y divisor resistivo
Señal de salida	Voltaje analógico	Digital (protocolo 1-Wire)	Resistencia variable
Alimentación	4 – 30 V	3.0 – 5.5 V	No requiere (pasivo)
Linealidad	Alta	Alta	Baja (curva logarítmica)
Tiempo de respuesta	1 s	1 s	Medio-lento
Facilidad de uso	Alta (requiere ADC)	Alta (requiere librería)	Media-baja (requiere calibración)
Robustez automotriz	Media	Media (si es encapsulado)	Alta (diseñado para motores)
Requiere calibración	No	No	Sí (curva R vs T)
Protección física	Bajo (plástico)	Medio (estancos disponibles)	Alto (rosca metálica, uso rudo)
Nivel lógico requerido	ADC (0–5 V)	Digital (3.3/5 V)	ADC con divisor resistivo

Cuadro 4: Comparación de instalación y robustez



4.3. Medición de velocidad

Módulo	Interfaz	Voltaje de operación	Frecuencia de actualización	Precisión típica (CEP)	Antena incluida
u-blox NEO-6M	UART / TTL	3.3–5 V	hasta 5 Hz	2.5 m	Sí (patch cerámico)
u-blox NEO-7M	UART / TTL	5 V	hasta 10 Hz	2.5 m	Sí (patch cerámico)
u-blox NEO-M8N	UART / TTL 5 V	3.3–5 V	hasta 10 Hz	2.5 m	Sí (patch cerámico)
u-blox ZED-F9P	UART / I ² C / SPI	3.3–5 V	hasta 10 Hz	2.5 m (centimétrica con RTK)	No (requiere antena activa externa)

Cuadro 5: Comparativa técnica de módulos GPS compatibles con Arduino Nano, ESP32 y Raspberry Pi Pico.

La elección de un módulo GPS como sensor de velocidad en vehículos antiguos se justifica por las siguientes razones principales:

- **Instalación no invasiva:** No requiere modificar el sistema mecánico ni eléctrico del vehículo, evitando interferencias con componentes originales.
- **Medición precisa y directa:** Proporciona la velocidad real respecto al suelo, sin depender del tamaño de ruedas ni sistemas de transmisión.
- **Compatibilidad:** Utiliza interfaz UART TTL, fácilmente integrable con microcontroladores como Arduino Nano, ESP32 y Raspberry Pi Pico.
- **Funcionalidad adicional:** Permite obtener también posición geográfica, hora precisa y trazado de rutas si se desea ampliar el sistema.
- **Bajo mantenimiento y costo razonable:** Es compacto, confiable y no requiere ajustes frecuentes ni piezas móviles.



4.4. Métodos de medición del voltaje de la batería del automóvil

Para conocer el estado de carga de la batería de un vehículo, es necesario medir su voltaje de forma precisa y segura. Dado que la mayoría de los microcontroladores trabajan con tensiones de referencia de 3.3 V o 5 V, es indispensable adaptar la señal proveniente de la batería antes de ingresarla a un pin analógico.

- **Divisor resistivo:** Es el método más simple y económico. Se utilizan dos resistencias conectadas en serie para reducir proporcionalmente el voltaje de entrada. El punto medio se conecta al pin analógico del microcontrolador. Es fundamental seleccionar valores adecuados para que el voltaje resultante nunca supere el máximo permitido por el ADC.
- **Aislación mediante opto acoplador o amplificador operacional:** En aplicaciones más sensibles o cuando se desea proteger el sistema contra picos de tensión, se puede aislar la medición mediante opto acopladores o usar amplificadores operacionales configurados como seguidores de tensión o comparadores.
- **Módulos externos ADC de alta precisión:** En sistemas que requieran mayor resolución o medición diferencial (por ejemplo, monitoreo de baterías en tiempo real con precisión de milivoltios), se pueden emplear conversores analógico-digitales externos como el ADS1115, que se comunican con el microcontrolador vía I²C.
- **Módulos dedicados de monitoreo de baterías:** Existen módulos como el INA219 o INA226 que permiten medir voltaje, corriente y potencia con comunicación digital, ofreciendo un diagnóstico más completo del estado de la batería.

4.5. Medición de presión de aceite

- **Uso del sensor original:** El vehículo cuenta de fábrica con un bulbo de presión que actúa como interruptor, activando una luz de advertencia cuando la presión desciende por debajo de cierto umbral (típicamente 0.3–0.5 bar). Este tipo de sensor no permite conocer la presión de forma continua, solo detectar situaciones críticas.
- **Sensor electrónico:** Para una lectura en tiempo real, se puede sustituir el bulbo por un sensor de presión electrónico con salida analógica proporcional (por ejemplo, 0–5 V para 0–10 bar). Este sensor se conecta a la entrada analógica del microcontrolador de forma directa.



4.6. Display LCD

Pantalla	Tecnología	Interfaz	Resolución / Tamaño
LCD 16×2 I2C	LCD HD44780	I2C	16×2 caracteres
LCD 20×4 I2C	LCD HD44780	I2C	20×4 caracteres
OLED 0.96" I2C	OLED SSD1306	I2C	128×64 px
OLED 1.3" I2C	OLED SH1106	I2C	128×64 px
TFT 1.8" SPI	TFT ST7735	SPI	128×160 px
TFT 2.4" SPI	TFT ILI9341	SPI	240×320 px
TFT 2.8" táctil	TFT ST7735 + Touch	SPI	240×320 px

Cuadro 6: Comparativa técnica de pantallas para microcontroladores

Pantalla	Voltaje	Visibilidad	Compatibilidad
LCD 16×2 I2C	5V	Media, pobre al sol	Arduino - ESP32/Pico (adaptador)
LCD 20×4 I2C	5V	Media, pobre al sol	Arduino - ESP32/Pico (adaptador)
OLED 0.96" I2C	3.3V	Excelente (alto contraste)	Compatible con todos
OLED 1.3" I2C	3.3V	Muy buena incluso al sol	Compatible con todos
TFT 1.8" SPI	3.3V	Moderada	Arduino (adaptador) - ESP32/Pico
TFT 2.4" SPI	3.3V	Buena	Arduino (adaptador) - ESP32/Pico
TFT 2.8" táctil	3.3V	Buena - interacción táctil	Arduino(adaptador) - ESP32/Pico

Cuadro 7: Condiciones eléctricas y compatibilidad de pantallas



5. REFERENCIAS

- [1] *Arduino Nano - Datasheet*. Arduino Documentation. Disponible en: <https://docs.arduino.cc/resources/datasheets/A000005-datasheet.pdf>
- [2] *ESP32 Series Datasheet*. Espressif Systems. Disponible en: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
- [3] *RP2040 Datasheet*. Raspberry Pi Ltd. Disponible en: <https://datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf>
- [4] *Datasheet LM35*. Texas Instruments. Disponible en: <https://www.ti.com/lit/ds/symlink/lm35.pdf>
- [5] *Datasheet DS18B20*. Analog Devices. Disponible en: <https://www.analog.com/media/en/technical-documentation/data-sheets/ds18b20.pdf>
- [6] *Beginner's guidance: GPS Module NEO 6M*. Arduino Forum. Disponible en: <https://forum.arduino.cc/t/beginners-guidance-gps-module-neo-6m/889018>
- [7] *I2C Liquid Crystal Displays*. Arduino Project Hub. Disponible en: https://projecthub.arduino.cc/arduino_uno_guy/i2c-liquid-crystal-displays-5eb615
- [8] *Guide for I2C OLED Display with Arduino*. Random Nerd Tutorials. Disponible en: <https://randomnerdtutorials.com/guide-for-oled-display-with-arduino/>
- [9] *Using the ST7735 1.8 Color TFT Display with Arduino*. Electronics-Lab. Disponible en: <https://www.electronics-lab.com/project/using-st7735-1-8-color-tft-display-arduino/>