

Sistema de Monitoreo Remoto para Ecosistemas Acuáticos

Tracker LoRa-APRS - Grupo 3

Marlon Daniel Ramírez Cruz
Heiner Esteban Rivera Víquez
Gabriel Ignacio Miranda Lopez

Taller Integrador - II Semestre 2025

21 de noviembre de 2025

Contenido

- 1 Introducción y Objetivos
- 2 Diseño del Sistema
- 3 Componentes y Hardware
- 4 Implementación de Software
- 5 Presupuesto
- 6 Resultados y limitaciones
- 7 Conclusiones y recomendaciones

- Los ecosistemas acuáticos requieren monitoreo constante de parámetros críticos para mantener condiciones óptimas
- Variaciones bruscas en temperatura, turbidez o nivel de agua pueden resultar fatales para los organismos acuáticos
- Necesidad de supervisión remota continua sin intervención humana directa
- Aplicación en peceras domésticas, acuarios, estanques de cultivo

Objetivos

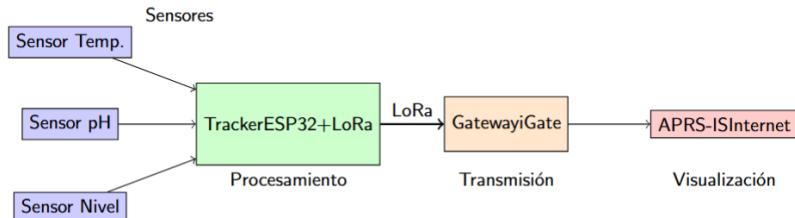
Objetivo Principal

Diseñar e implementar un sistema de monitoreo remoto para peceras domésticas que integre sensores de temperatura, turbidez y nivel de agua, con comunicación de datos mediante protocolo LoRa-APRS.

Objetivos Específicos

- 1 Integrar sensores de temperatura, turbidez y nivel de agua a un microcontrolador ESP32 con módulo LoRa
- 2 Desarrollar firmware para envío eficiente de telemetría via LoRa-APRS
- 3 Validar el funcionamiento en condiciones reales evaluando precisión, cobertura y confiabilidad

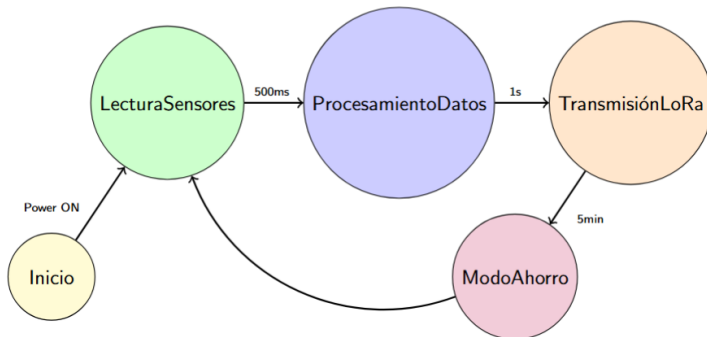
Arquitectura del Sistema



Flujo operacional:

- 1 Inicialización de sensores y módulo LoRa
- 2 Medición de temperatura, turbidez y nivel de agua
- 3 Conversión y concatenación de datos en string
- 4 Empaquetado en formato APRS como comentario
- 5 Transmisión LoRa hacia gateway a 433.775 MHz

Máquina de Estados del Firmware



Estados principales:

- **INICIALIZACIÓN:** Hardware, LoRa y sensores
- **LECTURA:** Adquisición temperatura, turbidez, nivel
- **PROCESAMIENTO:** Validación y formateo APRS
- **TRANSMISIÓN:** Envío paquetes LoRa
- **SLEEP:** Bajo consumo 5 minutos entre ciclos

Características

- MCU: ESP32 dual-core 240MHz
- LoRa: SX1276/SX1278
- Potencia: hasta +20dBm
- Alcance: 2-15km (rural)
- GPS: U-Blox Neo-6M
- Consumo: 240mA activo, 0.01mA sleep

Ventajas

- LoRa integrado nativo
- GPS para geolocalización
- Múltiples interfaces I/O
- Bajo consumo energético

Sensores Seleccionados

DS18B20 - Temperatura

- Rango: -55°C a $+125^{\circ}\text{C}$, Precisión: $\pm 0.5^{\circ}\text{C}$
- Protocolo: 1-Wire digital, resistente al agua
- Tiempo conversión: 750ms, Resolución: 12-bit programable

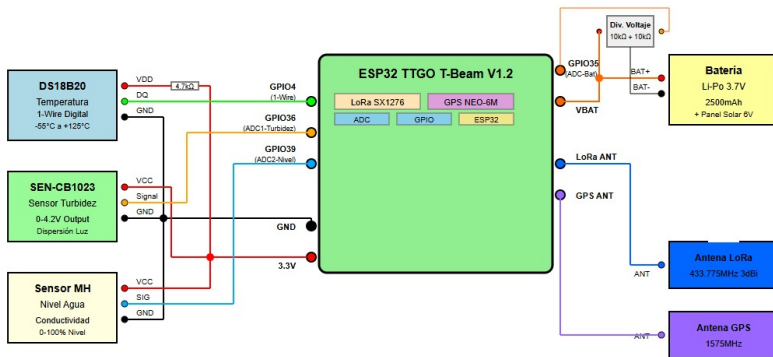
SEN-CB1023 - Turbidez

- Principio: Dispersión de luz por partículas suspendidas
- Salida: Analógica 0-4.2V DC, Alimentación: 5V
- Aplicación: Medición directa calidad de agua

MH Sensor - Nivel

- Principio: Conductividad eléctrica del agua
- Salida: Analógica proporcional al nivel
- Detección: Continua según inmersión

Implementación en Hardware



Flujo del Firmware

INICIO:

```
Configurar pines de sensores (GPIO4, GPIO36, GPIO39)
Inicializar bus OneWire para DS18B20
Configurar entradas analógicas ADC
Inicializar módulo LoRa (433.775 MHz)
Configurar parámetros APRS (callsign, path, etc.)
```

LOOP_PRINCIPAL:

```
// Activar sensores para medición
activarAlimentacionSensores()

// Adquisición de datos
tempC = leerTemperaturaDS18B20()
nivel = leerNivelAguaAnalogico() // 0-520 → %
turbValue = leerTurbidezAnalogico() // voltaje

// Desactivar sensores (ahorro energía)
desactivarAlimentacionSensores()
```

Formateo y Envío

```
// Formar string de telemetría
datos = "T:" + tempC + "C N:" + nivel + " Tu:" + turbValue

// Construcción paquete APRS
paqueteAPRS = crearPaqueteAPRS(
    callsign: "TI2XXX-7",
    destino: "APLRT1",
    ruta: "WIDE1-1",
    posicion: coordenadasGPS,
    comentario: datos
)

// Transmisión LoRa
enviarPorLoRa(paqueteAPRS, 433.775MHz)

// Entrar en modo de ahorro energético
esp_deep_sleep(300000ms) // 5 minutos

FIN_LOOP
```

Estructura base

- Código control de Tracker de Ricardo Guzman (richonguzman).
- Lectura de sensores agregado al paquete APRS.
- Comentario "Ingeniería Electronica-ITCR"
- Módulos DeepSleep y SmartBeacon.

Módulos eliminados

- Bluetooth, Ble, Joystick, Keyboard, Touch, Web, Wifi, Winlink y Wx.

Módulos adaptados

- Station, Utils, Configuration y LoRa APRS Tracker.

SmartBeaconing Estático

En el caso del SmartBeaconing, se implementó una versión estática ya que el dispositivo no requiere cálculos de velocidad o rumbo como en implementaciones tradicionales con GPS.

Características

- Transmisión periódica cada 5 minutos (configurable)
- Transmisión inmediata ante condiciones críticas:
 - Temperatura $> 30^{\circ}\text{C}$ o $< 18^{\circ}\text{C}$
 - Nivel de agua $< 30\%$
 - Cambios bruscos: $\Delta T > 5^{\circ}\text{C}$ o $\Delta \text{Nivel} > 20\%$

Ciclo Operativo

- 1 Despertar por temporizador RTC
- 2 Activar sensores (GPIO25, GPIO26)
- 3 Adquisición de datos (≈ 5 segundos)
- 4 Verificación SmartBeaconing
- 5 Transmisión LoRa (≈ 2 segundos)
- 6 Apagar sensores
- 7 DeepSleep por 5 minutos

Mejora Energética

- Sin DeepSleep: 240 mA \rightarrow 10.4 horas de autonomía
- Con DeepSleep: 0.47 mA \rightarrow 221 días (7.4 meses)
- **Mejora: 99.8 % en eficiencia energética**

Diagrama de Flujo

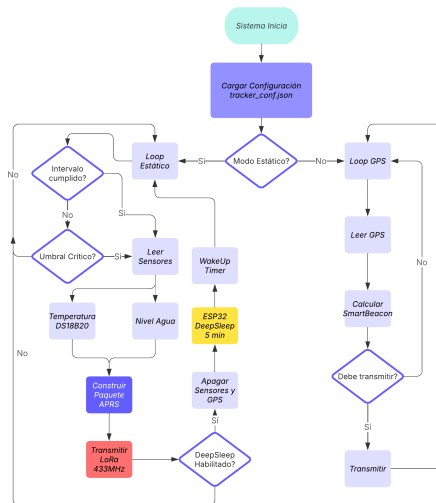


Figura: Diagrama de flujo de datos

Presupuesto del Proyecto

Componente	Cantidad	Costo (\$)
Hardware		
ESP32 T-Beam V1.2	1	30.77
DS18B20 (Temperatura)	1	8.50
Sensor Nivel MH	1	3.60
Batería Li-Po 2500mAh	1	12.00
Conectores y cables	-	8.00
PCB prototipo	1	5.00
Carcasa IP65	1	12.00
Subtotal Hardware		\$79.87
Ingeniería (25 horas × \$60/hora)		
Investigación y diseño	10 h	600.00
Implementación y pruebas	15 h	900.00
Subtotal Ingeniería		\$1500.00
COSTO TOTAL		\$1579.87

Resultados

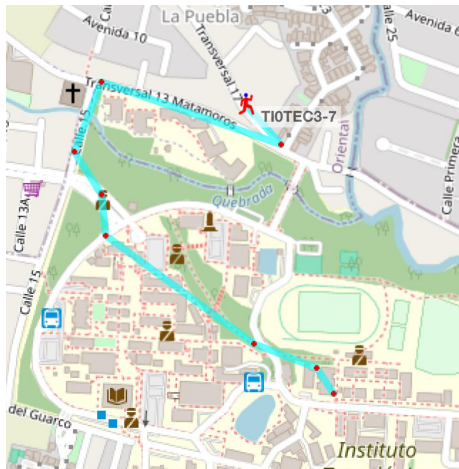


Figura: Registro del Tracker en la zona del campus Cartago, ITCR.

Resultados

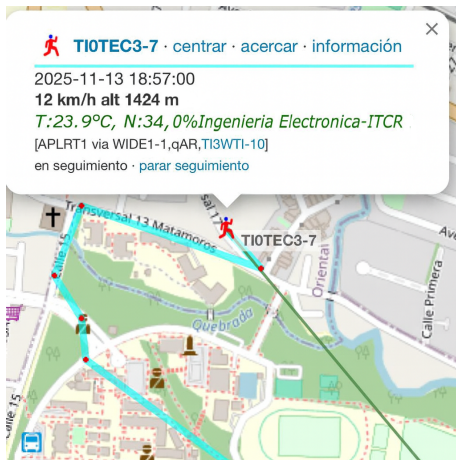


Figura: Información registrada por el dispositivo Tracker.

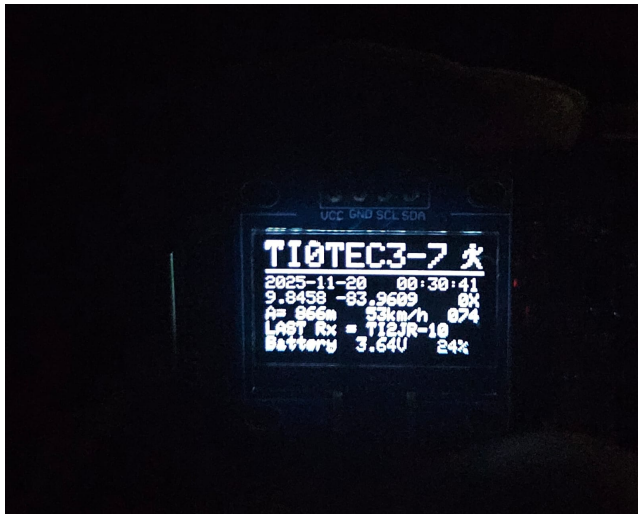


Figura: Información registrada por el dispositivo Tracker.



Figura: Implementación del prototipo.

Limitaciones

- Sensor medidor de PH: escasez en la disponibilidad de sensores medidores de PH.
- Sensor de turbidez: Alternativa al sensor de PH, no se encontró disponibilidad del mismo.
- Para una aplicación real se deben utilizar tanto sensores como una caja hermética con protección mínima IP67
- Se encontró una limitación en el uso en interiores del dispositivo para poder conectar con algún igate

Conclusiones

- Se logró implementar un prototipo de un sistema de monitoreo remoto para ecosistemas acuáticos con comunicación de datos mediante el protocolo LoRa-APRS
- Se lograron integrar dos de los tres sensores propuestos de manera funcional enviando datos al ESP32
- Se desarrolló un firmware correcto para el envío de telemetría
- Se validó el funcionamiento del sistema mediante el tracking web y la medición de los sensores implementados

¡Gracias por su atención!

Sistema de Monitoreo Remoto para Ecosistemas Acuáticos
Tracker LoRa-APRS - Grupo 3

¿Preguntas?