

# **TRABAJO ESPECIAL DE GRADO**

## **DESARROLLO DE UN EQUIPO DE CARACTERÍSTICAS REDUCIDAS PARA LA RECOLECCIÓN Y ALMACENAMIENTO DE DATOS METEOROLÓGICOS**

Presentado ante la Ilustre  
Universidad Central de Venezuela  
por el Br. Malavé V, Diego M.  
para optar al título de  
Ingeniero Electricista

Caracas, 2013

## **TRABAJO ESPECIAL DE GRADO**

### **DESARROLLO DE UN EQUIPO DE CARACTERÍSTICAS REDUCIDAS PARA LA RECOLECCIÓN Y ALMACENAMIENTO DE DATOS METEOROLÓGICOS**

Prof. Guía: Ing. Pedro Pinto  
Tutor Industrial: Ing. Diógenes Santander

Presentado ante la Ilustre  
Universidad Central de Venezuela  
por el Br. Malavé V, Diego M.  
para optar al título de  
Ingeniero Electricista

Caracas, 2013

## CONSTANCIA DE APROBACIÓN

Caracas, 21 de noviembre de 2013

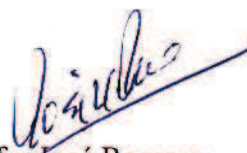
Los abajo firmantes, miembros del Jurado designado por el Consejo de Escuela de Ingeniería Eléctrica, para evaluar el Trabajo Especial de Grado presentado por el Bachiller Diego Malave:

***“DESARROLLO DE UN EQUIPO CON CARACTERÍSTICAS REDUCIDAS  
PARA LA RECOLECCIÓN Y ALMACENAMIENTO DE DATOS  
METEOROLÓGICOS”***

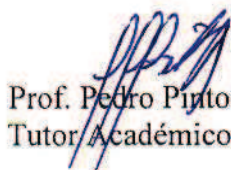
Consideran que el mismo cumple con los requisitos exigidos por el plan de estudios conducente al Título de Ingeniero Electricista en la mención de Electrónica y Control, y sin que ello signifique que se hacen solidarios con las ideas expuestas por el autor, lo declaran APROBADO.



Prof. Simón Morales  
Jurado



Prof. José Romero  
Jurado



Prof. Pedro Pinto  
Tutor Académico

## **DEDICATORIA**

A mis padres y mi familia.

## **RECONOCIMIENTOS Y AGRADECIMIENTOS**

A mi madre por todo su apoyo a lo largo de toda mi vida.

A los profesores Pedro Pinto, Rafael Rivero, Servando Álvarez, así como a todos los profesores de la Facultad y la Escuela de Ingeniería Eléctrica.

A Diógenes Santander, Roger Roa, Luis Santana, Winder Pérez, y todo el personal de TECNUM Electrónica por toda la ayuda que han facilitado.

**Malavé V., Diego M.**

**DESARROLLO DE UN EQUIPO DE CARACTERÍSTICAS REDUCIDAS PARA LA RECOLECCIÓN Y ALMACENAMIENTO DE DATOS METEOROLÓGICOS**

**Prof. Guía: Ing. Pedro Pinto. Tutor Industrial: Ing. Diógenes Santander. Tesis. Caracas. U.C.V. Facultad de Ingeniería. Escuela de Ingeniería Eléctrica. Ingeniero Electricista. Opción: Electrónica, computación y control. Empresa: TECNUM Electrónica C.A. 2013. 98 h.**

**Palabras claves:** Adquisición de Datos. Medición Meteorológica. Datalogger. Módulo Bluetooth. Memoria Flash Serial. SPI. USART. RS232.

**Resumen:** se plantea el diseño e implementación de un equipo de adquisición de datos para estaciones hidrometeorológicas de superficie, de características reducidas. El núcleo del dispositivo lo conforma un microcontrolador del fabricante Microchip, modelo PIC18F46J50, que trae incorporado un reloj de tiempo real programable, el cual es usado para realizar adquisiciones de forma periódica. Así mismo, el equipo cuenta con un convertidor Analógico/Digital de 10 bits interno del microcontrolador, para 4 canales de adquisición, y 1 canal digital que permite detectar pulsos emitidos por un sensor de precipitación tipo balancín. También cuenta con 2 puertos USART, de los cuales uno está destinado a la comunicación RS-232, y el otro permite comunicación vía BlueTooth entre el equipo de adquisición de datos y otro equipo que cuente con la misma tecnología inalámbrica. Los datos son almacenados en una memoria que está integrada en el dispositivo y es de tipo flash serial cuya capacidad es de 8 MB, aunque puede ser reemplazada por otras de 4 MB, 2 MB y 1 MB. La herramienta PCLogger.NET, desarrollada previamente, permite la configuración, la supervisión de la adquisición de los distintos canales, la descarga de los datos y su conversión a un formato de texto que puede ser procesado desde la computadora. El equipo de adquisición que ha sido desarrollado llevará el nombre comercial de NanoLogger y está basado en su antecesor, el  $\mu$ Logger 3.2, que ya ha sido probado en campo y ha demostrado ser una solución práctica para el registro y almacenamiento de la información hidrometeorológica.

# ÍNDICE GENERAL

	Pág.
<b>CONSTANCIA DE APROBACION</b> .....	iii
<b>DEDICATORIA</b> .....	iv
<b>RECONOCIMIENTOS Y AGRADECIMIENTOS</b> .....	v
<b>RESUMEN</b> .....	v
<b>ÍNDICE GENERAL</b> .....	vi
<b>ÍNDICE DE TABLAS</b> .....	x
<b>ÍNDICE DE FIGURAS</b> .....	xi
<b>ACRÓNIMOS</b> .....	xiii
<b>INTRODUCCIÓN</b> .....	15
<b>CAPÍTULO 1. DESCRIPCIÓN DEL PROYECTO</b> .....	16
1.1. Planteamiento del Problema.....	16
1.2. Justificación.....	17
1.3. Objetivos .....	18
1.3.1. Objetivo General.....	18
1.3.2. Objetivos Específicos .....	18
<b>CAPÍTULO 2. MARCO TEÓRICO</b> .....	19
2.1. Estaciones Meteorológicas.....	19
2.2. Variables Hidrológicas y Meteorológicas.....	20
2.3. Registradores de Datos .....	20
2.4. Circuitos Integrados de Almacenamiento (Memorias).....	21
2.5. Comunicación serial.....	23
2.5.1. Serial Peripheral Interface (SPI) .....	23
2.5.2. Inter-Integrated Circuit (Inter-Circuitos Integrados) o I <sup>2</sup> C .....	25
2.5.3. Estándar RS-232.....	27
2.6. El Estándar IEEE 802.15.1 (Bluetooth) .....	28
2.6.1. Funcionalidad .....	29
2.7. El $\mu$ Logger 3.x de TECNUM Electrónica .....	29
<b>CAPÍTULO 3. DESCRIPCIÓN DEL EQUIPO</b> .....	32
3.1. Características del Equipo (Nanologger v1.0x) .....	32
3.1.1. El Microcontrolador .....	33

3.1.2. Reloj en Tiempo Real con Calendario (RTCC) .....	36
3.1.3. Conversión Analógica Digital .....	36
3.1.5. Módulo de Memoria.....	37
3.1.6. Módulo Bluetooth .....	38
3.1.7. Fuente de Tensión .....	38
3.1.8. Transistores FET .....	39
3.2. Definición del Software .....	39
3.2.1. El Firmware.....	40
3.2.2. Software de Configuración.....	41
<b>CAPÍTULO 4. CARACTERÍSTICAS PARTICULARES (PROCESO DE DESARROLLO)</b> .....	<b>42</b>
4.1. Estudio y Selección de Componentes. Criterios de Selección.....	42
4.1.1. Criterios generales de selección .....	43
4.1.1.1. Rango de Tensiones de Alimentación, corrientes de operación y estado de bajo consumo .....	43
4.1.1.2. Rangos de Temperatura .....	43
4.1.1.3. Capacidad Integrada de Comunicación Serial.....	43
4.1.1.4. El Costo del Circuito Integrado y la Disponibilidad .....	44
4.1.2. El microcontrolador.....	44
4.1.2.1. Máxima frecuencia del Oscilador.....	44
4.1.2.2. Cantidad de Entradas / Salidas .....	44
4.1.2.3. Tamaño de la Memoria RAM y de Programa. ....	45
4.1.2.4. Módulo de Conversión Analógica/Digital.....	45
4.1.2.5. Reloj en Tiempo Real (RTC).....	45
4.1.2.6. El PIC16LF1847 de Microchip .....	46
4.1.2.7. El PIC18LF14K50 de Microchip.....	46
4.1.2.8. La familia PIC18F46J50 de Microchip .....	47
4.1.2.9. El P89LPC915 de NXP/Phillips .....	47
4.1.2.10. El Controlador elegido.....	47
4.1.3. Módulos de Memorias.....	48
4.1.3.1. Tipo de Memoria .....	48



4.1.3.2. Velocidad máxima de escritura .....	48
4.1.3.3. Ciclo de escritura .....	49
4.1.3.4. Cantidad de memorias que se pueden conectar en cascada.....	49
4.1.3.5. Durabilidad (ciclos de escritura-borrado) .....	49
4.1.3.6. Retención de datos .....	49
4.1.3.7. Capacidad (Densidad / Organización) .....	49
4.1.4. Módulo Bluetooth .....	51
4.1.4.1. Versión de Bluetooth .....	51
4.1.4.2. Máxima Tasa de bits .....	51
4.1.4.3. Rango de cobertura (Clase) .....	51
4.1.5. Driver RS232.....	53
4.2. Características específicas del Software .....	54
4.2.1. Estructura del Firmware .....	55
4.2.1.1. Núcleo del Firmware: la rutina Atender Tareas .....	57
4.2.1.1.1. Banderas de Tareas y Máscaras .....	58
4.2.1.2. Máquinas de estados del Nanologger .....	58
4.2.1.3. La máquina de Adquisición Temporizada (o por intervalos) ....	59
4.2.1.4. Máquina de comunicación (USART) .....	67
4.2.1.5. Máquina: Switch Bluetooth .....	73
4.2.1.6. Máquina: Iniciar Adquisición .....	74
4.2.1.7. Máquina: Parar Adquisición .....	75
4.2.1.8. Máquina de Configuración del Bluetooth .....	76
4.2.1.9. Máquina de Lectura de Caracteres Bluetooth.....	80
4.2.2. Otros aspectos .....	81
4.2.2.1. La secuencia de Adquisición del Logger.....	81
4.2.2.2. El Pluviómetro tipo Balancín ó Tipping Bucket .....	83
4.2.2.3. El Archivo de Datos.....	84
4.2.2.4. Las Adquisiciones Forzadas .....	85
4.2.2.5. El software de dispositivos externos.....	86
4.2.2.6. Manejo del Consumo.....	87

4.2.2.7. Módulo Bluetooth incorporado a la tarjeta.....	89
4.3. Pruebas Realizadas y Resultados .....	90
4.3.1. Pruebas de comunicación .....	90
4.3.2. Pruebas de Adquisición de Datos .....	91
4.3.3. Pruebas de Registro .....	91
4.3.4. Pruebas de Consumo .....	92
4.3.5. Pruebas de Compatibilidad.....	93
4.3.6. Pruebas de Comunicación Inalámbrica .....	94
<b>CONCLUSIONES</b> .....	95
<b>RECOMENDACIONES</b> .....	96
<b>BIBLIOGRAFÍA</b> .....	98

## ÍNDICE DE TABLAS

Tabla 1. Estudio de las opciones para la elección del microcontrolador. ....	46
Tabla 2. Estudio de las opciones para la elección de la memoria. ....	50
Tabla 3. Estudio de las opciones para la elección del módulo Bluetooth. ....	52
Tabla 4. Estudio de las opciones para la selección del driver RS232. ....	53
Tabla 5. Consumo del NanoLogger. ....	93

## ÍNDICE DE FIGURAS

Figura 1. Registrador de Datos genérico, con mínima funcionalidad. ....	21
Figura 2. Diagrama de conexiones del protocolo SPI. SCLK: reloj, MOSI: Master out slave in, MISO: Master in slave out, SS: Slave select. ....	23
Figura 3. Diagrama en tiempo de la señal de datos, el reloj y el selector de esclavo. ....	24
Figura 4. Modo A - Espera en nivel alto, lectura en flancos de subida. ....	24
Figura 5. Modo B – Espera en nivel bajo, lectura con flancos de subida. ....	25
Figura 6. Modo C – Espera en nivel alto, operación en flancos de bajada. ....	25
Figura 7. Modo D – Espera en nivel bajo, operación en flancos de bajada. ....	25
Figura 8. Conexión del bus I2C. ....	26
Figura 9. Paquete de datos del protocolo I2C. ....	26
Figura 10. Diagrama de la transferencia del paquete. ....	27
Figura 11. Diagrama de tiempo de la señal del reloj y la señal de datos del bus I2C. ....	27
Figura 12. Forma de la trama RS-232. ....	28
Figura 13. El $\mu$ Logger 3.x de TECNUM Electrónica. ....	31
Figura 14. Diagrama de Bloques funcionales del NanoLogger. ....	33
Figura 15. Diagrama esquemático del convertidor DC-DC Step-Up MCP1624 alimentando a un Microcontrolador. ....	39
Figura 16. Programa Principal. ....	55
Figura 17. Izquierda: Inicialización de Periféricos. Derecha: Inicialización de variables del Logger. ....	56
Figura 18. Rutina: Atender Tareas. ....	57
Figura 19. Máquina de Adquisición Temporizada. ....	59
Figura 20. Medición de Batería y Verificación de Inicio de Adquisición. ....	62
Figura 21. Fijación de tiempo de establecimiento. ....	63
Figura 22. Subrutina: adquirir canales analógicos. ....	64
Figura 23. Subrutina: construir bloques de datos. ....	65
Figura 24. Salvar datos en memoria. ....	66

Figura 25. Subrutinas: Creación de Archivo, y Almacenamiento de bloques de datos en memoria flash externa. ....	67
Figura 26. Máquina de comunicación USART.....	68
Figura 27. Leer Carácter. ....	70
Figura 28. Subrutina: convertir carácter ASCII a Nibbles Hex. ....	71
Figura 29. Ejecutar Comando. ....	72
Figura 30. Transmitir Comando.....	73
Figura 31. Máquina Switch Bluetooth. ....	74
Figura 32. Máquina: Iniciar Adquisición.....	75
Figura 33. Máquina: Parar Adquisición.....	76
Figura 34. Máquina: Configurar Bluetooth (parte 1).....	78
Figura 35. Máquina: Configurar Bluetooth (parte 2).....	79
Figura 36. Máquina: Leer Caracteres Bluetooth (parte 1). ....	80
Figura 37. Máquina: Leer Caracteres Bluetooth (parte 2). ....	81

## ACRÓNIMOS

ADC: Analog to digital converter, Convertidor analógico-digital.

ASCII: American Standard Code for Information Interchange.

BCD: Binary-coded Decimal, Decimal codificado en binario.

CRC: Cyclic Redundancy Check, comprobación de redundancia cíclica.

E/S: Entrada/Salida.

ECCP: Enhanced Capture/Compare/PWM – Comparador/Capturador/PWM Mejorado.

EDR: Enhanced Data Rate (Tasa de datos mejorada).

EPROM: Erasable Programmable Read-Only Memory.

EEPROM: Electrically-Erasable Programmable Read-Only Memory.

EMA: Estación Meteorológica Automatizada.

EUSART: Enhanced Universal Synchronous-Asynchronous Receiver-Transmitter.

FAT: File Allocation Table.

GPRS: General packet radio service.

HCI: Host/Controller Interface (Interfáz controlador/huesped).

I/O: Input/output.

I<sup>2</sup>C: Inter-Integrated Circuit (Inter-Circuitos Integrados).

INAMEH: Instituto Nacional de Meteorología e Hidrología.

ISM: Industrial, Scientific and Medical radio band.

MARNR: Ministerio del Ambiente y de los Recursos Naturales Renovables.

MMC: Multi-Media Card.

MSSP: Master Synchronous Serial Port – Puerto Maestro Síncrono Serial.

OMM: Organización Meteorológica Mundial.

OSI: Open System Interconnection.

PC: Computador Personal.

PCB: Printed Circuit Board (Circuito Impreso).

PNUD: Programa de las Naciones Unidas para el Desarrollo.

PROM: Programmable Read-Only Memory.

PWM: Pulse Width Modulation – Modulación por Ancho de Pulsos.

RAM: Random Access Memory, Memoria de Acceso Aleatorio.

ROM: Read-only Memory, Memoria de solo lectura.

RTC: Real Time Clock, Reloj de Tiempo Real.

RTCC: Real Time Clock-Calendar, Reloj de Tiempo Real con Calendario.

SCP: Sistema Central de Procesamiento.

SD: Secure Digital.

SDCA: Secure Digital Card Association.

SIG: Special Interest Group (Bluetooth).

SIMCARSP: Sistema Integral de Monitoreo de la Cuenca Alta del Río San Pedro.

SPI: Serial Peripheral Interface.

SPP: Serial Port Profile.

SRAM: Static Random Access Memory.

UAD: Unidad de Adquisición de Datos.

UART: Universal Asynchronous Receiver-Transmitter, Transmisor-Receptor  
Asíncrono Universal.

USART: Universal Synchronous-Asynchronous Receiver-Transmitter.

UNDP: United Nations Development Program (ver PNUD).

USB: Universal Serial Bus.

VENEHMET: Proyecto Venezolano de Hidrología y Meteorología.

WMO: World Meteorological Organization (ver OMM).

μC: Microcontrolador.

μP: Microprocesador.

## INTRODUCCIÓN

El presente proyecto tuvo como finalidad el desarrollo de un equipo de características reducidas para la recolección y el almacenamiento de datos provenientes de una estación meteorológica de superficie, propuesto por la empresa TECNUM Electrónica, encargada del desarrollo, y distribución de equipos que faciliten la elaboración de estudios climáticos, así como de la instalación de estaciones meteorológicas, entre otras actividades.

Dicho equipo fue desarrollado primordialmente pensando en la compatibilidad con la tecnología propia de TECNUM, para formar parte del conjunto de elementos que conforman las estaciones meteorológicas con las que dicha empresa trabaja, tales como los sensores, los equipos de configuración y los sistemas de telemetría.

El primer capítulo de este documento realiza una descripción general del proyecto, señalando los objetivos generales y específicos.

El segundo capítulo recopila y sintetiza información que ha servido de base teórica para el desarrollo del equipo electrónico, tal como el concepto de registrador de datos, la estación meteorológica, los protocolos de comunicación serial, las características del equipo previamente desarrollado por TECNUM.

El tercer capítulo contiene la descripción de las características del equipo tanto a nivel de hardware como a nivel de software.

El cuarto capítulo expone información respecto al proceso de diseño, así como la composición del firmware del equipo a nivel de diagramas de flujo, entre otros aspectos de su funcionalidad.



# **CAPÍTULO 1. DESCRIPCIÓN DEL PROYECTO**

## **1.1.Planteamiento del Problema**

La información que suministran los equipos que conforman una estación meteorológica es de suma importancia para varios aspectos vitales para la vida del ser humano.

Entre estos aspectos se puede mencionar la generación de pronósticos que ayuden a la optimización de la eficiencia en el aprovechamiento de los recursos hídricos, la disminución de accidentes viales, aéreos y marítimos, la maximización de la producción agrícola mediante el apropiado uso de fertilizantes, pesticida y riegos, etc. También es importante para reducir el impacto de los desastres naturales en la actividad humana, tanto comercial como cotidiana.

Por muchos años, los equipos de adquisición de datos meteorológicos usados en el país eran de tecnología importada. Sin embargo, estos contaban con poco soporte técnico en el país. Por esta razón en el año 1993 el MARNR propuso a la empresa TECNUM Electrónica C.A., el desarrollo de tecnología propia para el campo de la meteorología, adquiriendo experiencia y ofreciendo soluciones a las necesidades actuales. La incesante necesidad de adaptar los instrumentos de medición y electrónicos a las tecnologías actuales, así como de reducir costos y ofrecer soluciones más versátiles crea el interés en hacerle seguimiento al desarrollo de los equipos de adquisición que forman parte de la base de las estaciones meteorológicas.

En este trabajo se describe el desarrollo de un equipo de adquisición y recolección de datos con características reducidas, adaptado a las necesidades actuales. Este equipo cumple con características de tamaño y bajo consumo, plataforma de comunicación actualizada, y prestaciones económicas que satisfacen las tareas requeridas para una estación meteorológica, tales como capacidad de almacenamiento, poder de procesamiento, canales de entrada, durabilidad, resistencia

a condiciones ambientales, etc. La unidad de adquisición de datos desarrollada se ha basado en versiones previas desarrolladas por TECNUM.

## **1.2. Justificación**

TECNUM Electrónica C.A. es una empresa venezolana fundada en 1989, dedicada al desarrollo e integración de equipos y sistemas de medición, en la cual fue desarrollado el presente trabajo.

La empresa TECNUM ha desarrollado productos para diferentes aplicaciones en las áreas de instrumentación y control. Entre los productos desarrollados se encuentran estaciones de medición ambiental, registradores, sensores, software, adquisición de datos, instrumentos de laboratorio) y educación (mecatrónica, y control de procesos). Estudiantes de la UCV han desarrollado para la misma, equipos de recolección y almacenamiento de datos, cuyo modelo más reciente es el conocido comercialmente como  $\mu$ Logger 3.2, unidad de adquisición de datos diseñada para la tarea de atender de forma compacta y versátil las necesidades de automatización de estación de recolección de datos meteorológicos.

Debido a la necesidad de realizar dichas tareas con mayor portabilidad e independencia energética, se requiere del desarrollo de una versión más simplificada, de bajo costo, bajo consumo, dimensiones reducidas, además de plataforma de comunicaciones adaptada a la tecnología actual para ponerla a disposición del usuario.

El dispositivo debe ser capaz de realizar la recolección de datos provenientes de un sistema de medición de variables meteorológicas, así como también el almacenamiento de dichos datos en una memoria, y la lectura de éstos mediante conexión entre el dispositivo y una PC, o por medio de comunicación vía Bluetooth con un dispositivo móvil.

La empresa TECNUM tiene contemplado darle el nombre comercial de “NanoLogger” al equipo de almacenamiento de datos a ser desarrollado.

### **1.3.Objetivos**

#### **1.3.1. Objetivo General**

Diseñar, y construir el prototipo de un equipo de recolección y almacenamiento de datos de características reducidas, de fácil utilización y que ofrezca soluciones a las necesidades de medición meteorológicas que sean compactas.

#### **1.3.2. Objetivos Específicos**

- Estudiar las tecnologías disponibles en el mercado de equipos de recolección y almacenamiento de información, así como los componentes posibles que los conforman.
- Discutir y realizar el diseño de la estructura física del dispositivo de recolección y almacenamiento de datos que realice todas las tareas requeridas.
- Realizar la codificación en el lenguaje de programación disponible de todas las rutinas necesarias para garantizar el funcionamiento adecuado del equipo electrónico, las cuales van desde la recopilación y almacenamiento de datos hasta la comunicación con dispositivos externos.
- Realizar las tareas pertinentes de simulación, depuración, montaje físico y pruebas con el fin de garantizar el cumplimiento de las condiciones fijadas para el equipo de recolección y almacenamiento de datos.
- Garantizar que el equipo cumpla las características de dimensiones reducidas, bajo consumo respecto a sus antecesores, e integrable a las tecnologías móviles de actualidad.
- Redactar toda la documentación que debe acompañar a la presentación del equipo, tal es el caso del tomo de tesis, la descripción técnica, y los diagramas descriptivos del equipo.

## CAPÍTULO 2. MARCO TEÓRICO

### 2.1. Estaciones Meteorológicas

Una estación meteorológica es un conjunto de instrumentos instalados en un ambiente, con el propósito de realizar mediciones de las condiciones meteorológicas, ambientales y climatológicas de una zona en particular para su posterior análisis y estudio.

Las estaciones meteorológicas que son capaces de realizar mediciones y transmisión de datos de forma automática se denominan estaciones meteorológicas automatizadas (E.M.A.), y actualmente abarcan la mayor parte de las instalaciones para dichos fines, requiriendo poco mantenimiento. Éstas son capaces de realizar mediciones periódicas durante largos períodos de tiempo, generando bancos de datos que permiten determinar variaciones, extremos y tendencias del estado del tiempo.

Las estaciones meteorológicas tienen aplicación en el campo de la actividad humana que se ve más afectada por los efectos de las variaciones del tiempo, tales como:

- Planificación Urbana
- Generación de alertas a las comunidades
- Estudios climáticos
- Agricultura y Ganadería.
- Transporte.
- Generación de energía.
- Turismo.

## **2.2. Variables Hidrológicas y Meteorológicas**

En el campo de la meteorología existen diversas variables de sumo interés para su observación y estudio, de entre los cuales, los más comunes son:

- Precipitación.
- Temperatura del aire y del suelo.
- Humedad relativa del aire.
- Presión.
- Radiación Solar.
- Velocidad y dirección del Viento.
- Caudal.
- Nivel de un cuerpo de agua.

## **2.3. Registradores de Datos**

En el campo de la meteorología, se ha automatizado el proceso del registro de las mediciones, a través de las unidades de Adquisición de Datos, a las cuales también se les conoce como Data Loggers (o registradores de datos). Los DataLoggers son equipos electrónicos destinados a la recolección de datos, obtenidos mediante sensores e instrumentos de medición, internos o externos, para su recopilación y almacenamiento en una memoria interna. El diseño de los registradores está enfocado hacia la durabilidad y la autonomía para poder ser instalados en ambientes de condiciones hostiles.

Estos sistemas electrónicos están compuestos por un procesador central, un módulo de conversión analógica-digital en conjunto con una electrónica de retención de datos y unos canales de entrada analógica, un reloj en tiempo real que permita al sistema registrar el tiempo para el cual se realizan las mediciones, y unos módulos de comunicación remota para eventualmente poder transferir los datos a un terminal.

Pueden o no disponer de otros periféricos de entrada/salida como displays y teclado para la configuración y monitoreo en el sitio.

Las unidades de adquisición se integran a las estaciones meteorológicas, conectándose con los sensores que toman las mediciones y las convierten en señales eléctricas analógicas, que son convertidas a formato digital para ser almacenado en módulos de memoria. También pueden ser transmitidos por distintos medios, ya sean cableados o inalámbricos.

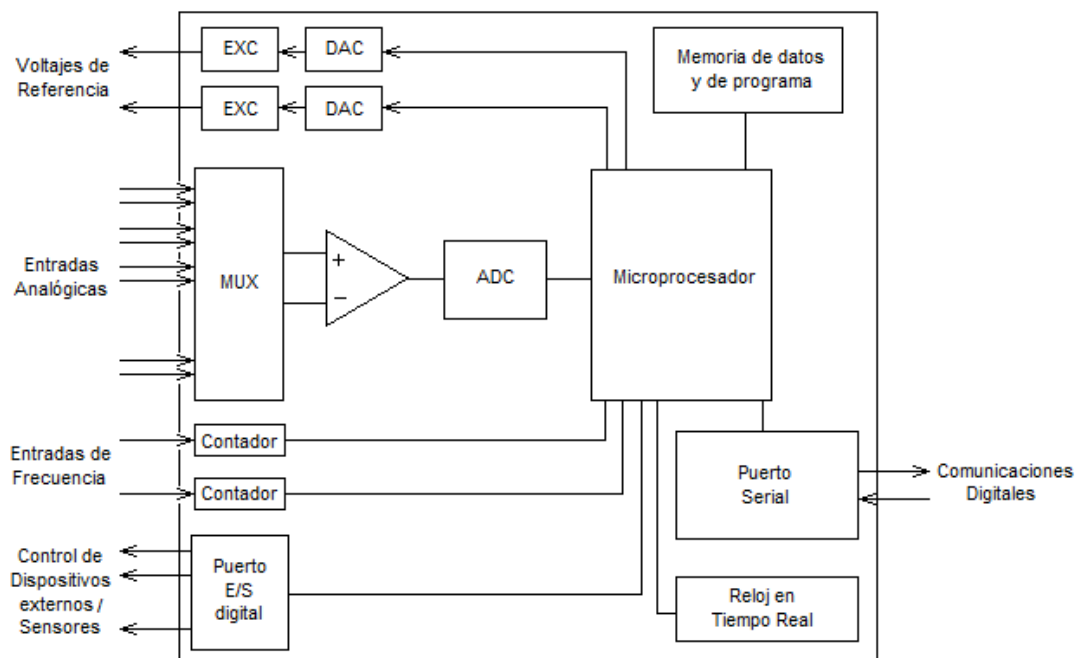


Figura 1. Registrador de Datos genérico, con mínima funcionalidad.

## 2.4. Circuitos Integrados de Almacenamiento (Memorias)

Los módulos de almacenamiento de datos o memorias que existen en el mercado son muy diversos, tanto por el método que utilizan para almacenar o borrar, como por la conexión física o la topología. En esta última clasificación se encuentran las memorias de conexión paralela o serial. Las memorias paralelas cuentan con la ventaja de que se consiguen por capacidades del orden de varios GigaBytes de

tamaño de almacenamiento, pero su topología puede aumentar la complejidad del circuito final.

Según el método que utilizan las memorias para la escritura y borrado de los datos, existen 2 tipos fundamentales, los cuales son las memorias RAM y las ROM. Las primeras son las memorias de acceso aleatorio y, aunque el acceso a los datos ocurre con gran fluidez debido a su velocidad de escritura o borrado, su integridad es volátil, es decir, los datos solo permanecerán en la memoria mientras ésta esté encendida, si se le apaga, los mismos se pierden, porque requiere de alimentación para su retención.

Las memorias tipo ROM son memorias de sólo lectura, y la integridad de los datos es fija al sólo permitir la escritura una vez. Estas memorias han ido evolucionando a lo largo del tiempo, ramificándose a varios subtipos, de entre los cuales se puede encontrar a las siguientes:

Memorias PROM: *Programmable Read-Only Memory*, de sus siglas en inglés, cuentan con un arreglo de fusibles que son quemados como método de programación de la información. Luego de eso, ya no permiten escritura de datos.

Memorias EPROM: *Erasable Programmable Read-Only Memory*, o memorias de solo lectura programables y borrables. El método de borrado de la información sólo es posible a través de un método especial mediante luz ultravioleta.

Memorias EEPROM ó E<sup>2</sup>PROM: *Electrically-Erasable Programmable Read-Only Memory*, o memorias de solo lectura programables y borrables electrónicamente. El proceso de borrado y escritura es más versátil por lo que su uso se extiende al de almacenamiento de datos dinámicos.

Memorias Flash: permiten la escritura y borrado de sus datos de forma electrónica, y presentan características de mayor velocidad puesto que los datos son escritos en bloques (o por páginas), a diferencia de las EEPROM, además, se les

encuentra en mayor capacidad y a dimensiones y costos reducidos. Pero la desventaja es que presentan una durabilidad limitada (pocos ciclos de escritura/borrado).

## 2.5. Comunicación serial

La comunicación serial que se da entre dos componentes electrónicos necesita de, al menos, una línea de comunicación por la que fluye la información en forma de bits (niveles altos o niveles bajos de tensión) desde el transmisor hacia el receptor.

Existen dos tipos de comunicación serial: la síncrona y la asíncrona. La primera se caracteriza por disponer de una tercera línea conocida como reloj, y ésta es la que hace uso de pulsos para dictar la ocurrencia del reconocimiento de los datos. El segundo tipo de comunicación serial no requiere de dicha línea y los datos son enviados con una tasa fija de transferencia para que el receptor pueda hacer el reconocimiento adecuadamente. Tanto el transmisor como el receptor necesitan estar configurados con dicha tasa de transferencia.

Mientras que los protocolos SPI e I<sup>2</sup>C son síncronos, el estándar RS232 es asíncrono.

### 2.5.1. Serial Peripheral Interface (SPI)

La comunicación SPI requiere de 4 hilos, de los cuales uno va del maestro al esclavo, el otro del esclavo al maestro, un tercer hilo para la señal del reloj, y el 4to hilo para la selección del esclavo. Se pueden conectar múltiples esclavos a un mismo maestro, requiriendo un hilo de selección para cada esclavo, pudiendo éstos compartir los 3 hilos nombrados anteriormente.

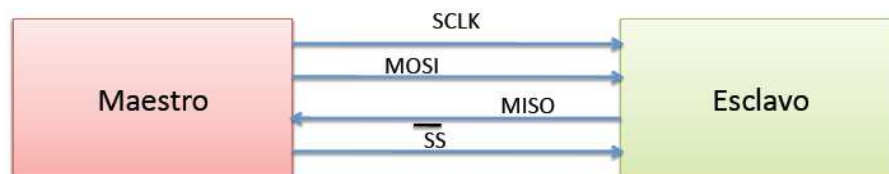


Figura 2. Diagrama de conexiones del protocolo SPI. SCLK: reloj, MOSI: Master out slave in, MISO: Master in slave out, SS: Slave select.



La comunicación SPI es Full Dúplex, es decir, puede haber un envío de datos al mismo tiempo que ocurre una recepción. La señal del reloj marca la pauta para la lectura de los datos desde el instante en que es seleccionado el esclavo (mediante lógica inversa, “0” lógico). La figura 3 ilustra el proceso:

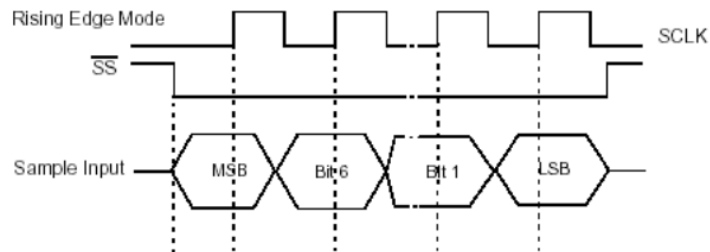


Figura 3. Diagrama en tiempo de la señal de datos, el reloj y el selector de esclavo.

Hay dos parámetros que influyen en el modo como se leen los datos durante la comunicación: La polaridad del reloj (CPOL) y la fase del reloj (CPHA). CPOL indica si el estado de espera del reloj es en “0” ó “1” lógico, y CPHA indica si es un flanco de subida o un flanco de bajada de la señal del reloj el que induce la lectura de los datos.

La Figura 4 ilustra la comunicación SPI cuando el estado de espera del reloj se mantiene en 1, y las lecturas ocurren con un flanco de subida. En la Figura 5 el estado de espera del reloj permanece en 0, y las lecturas ocurren con flancos de bajada. Las Figuras 6 y 7 ilustran los 2 casos restantes de la polaridad y la fase del reloj.

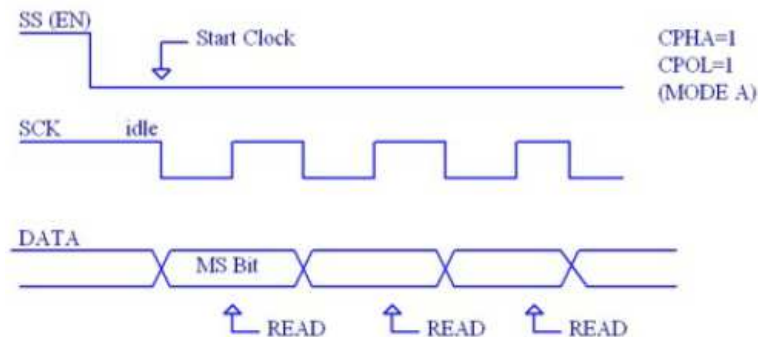


Figura 4. Modo A - Espera en nivel alto, lectura en flancos de subida.

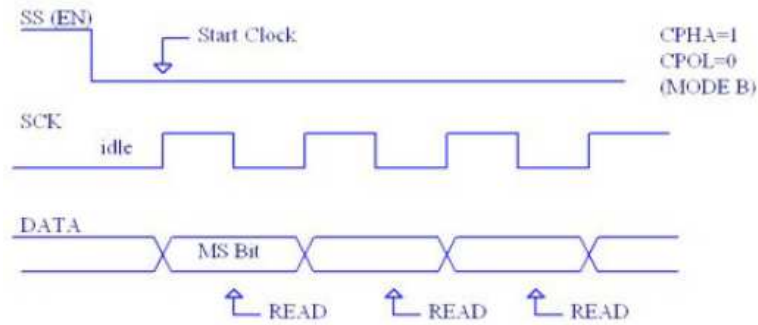


Figura 5. Modo B – Espera en nivel bajo, lectura con flancos de bajada.

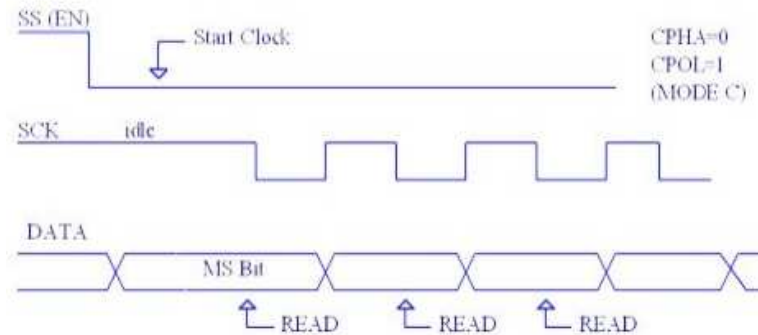


Figura 6. Modo C – Espera en nivel alto, operación en flancos de bajada.

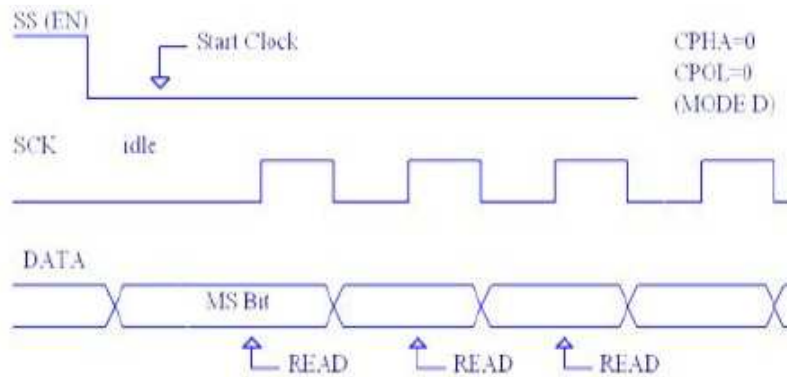


Figura 7. Modo D – Espera en nivel bajo, operación en flancos de bajada.

### 2.5.2. Inter-Integrated Circuit (Inter-Circuitos Integrados) o I<sup>2</sup>C

El protocolo I<sup>2</sup>C, desarrollado y patentado por Phillips, permite las comunicaciones entre dos o más módulos mediante únicamente 2 hilos conectados a unas resistencias de pull-up. Uno de estos hilos es el SDA, por donde viajan los datos,

y el segundo hilo es SCK, que suministra la señal del reloj. También requiere de una tercera línea para ser conectada a tierra (o masa).

La comunicación es Half Duplex, lo que significa que es bidireccional, pero la transferencia sólo ocurre en un sentido a la vez. Su velocidad estándar es de 100kbits/s pero también puede llegar a 3,4 Mbit/s. Más de un esclavo puede estar conectado en paralelo con el maestro, por lo que éste último inicia la comunicación enviando la dirección del esclavo antes de realizar la operación de lectura/escritura. Así mismo, no necesariamente un módulo maestro ejerce siempre la misma función, por lo que un bus I<sup>2</sup>C puede ser multimaestro.

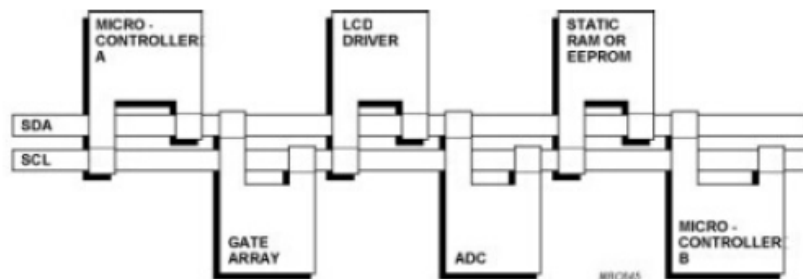


Figura 8. Conexión del bus I2C.

La comunicación I2C posee una trama cuya estructura se ilustra en la Figura 9, mientras que la figura 10 ilustra el proceso de transferencia y fin de la comunicación:

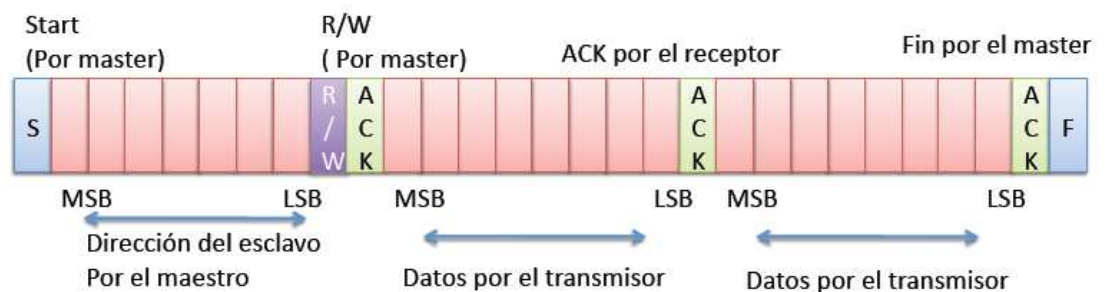


Figura 9. Paquete de datos del protocolo I2C.

El maestro inicia la comunicación enviando 7 bits que corresponden a la dirección del esclavo y un 8vo bit que indica si la operación va a ser lectura o escritura. El esclavo responde con un bit de reconocimiento (*Acknowledge*) en bajo, y luego ocurre la transferencia de los siguientes 8 bits.

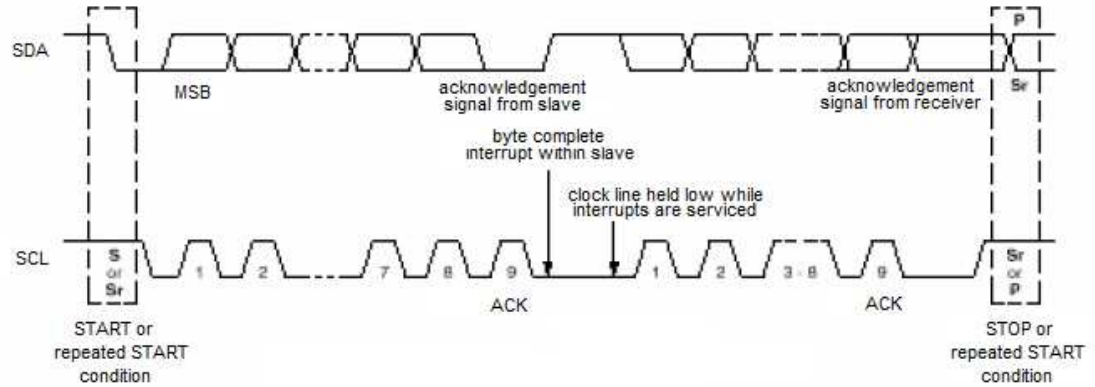


Figura 10. Diagrama de la transferencia del paquete.

Durante la comunicación I<sup>2</sup>C, la línea de datos debe mantenerse estable mientras la señal del reloj esté en alto, porque es el momento del reconocimiento del dato. Cuando la señal del reloj está en bajo, es el momento en el que el dato conmuta.

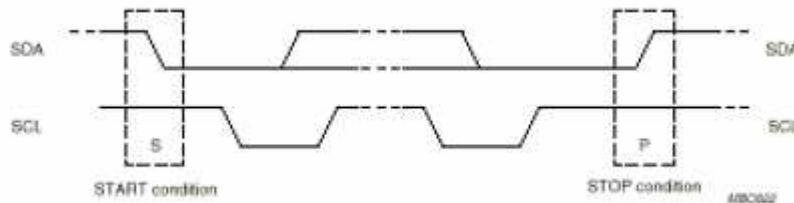


Figura 11. Diagrama de tiempo de la señal del reloj y la señal de datos del bus I2C.

### 2.5.3. Estándar RS-232

El transmisor permanece en un estado de espera hasta que la transferencia es iniciada con el bit de arranque, luego es transmitido el byte desde su bit menos significativo (LSB) hasta su bit más significativo (MSB), y culmina con el bit de parada, para retornar al estado de espera.

La figura 12 ilustra el proceso de comunicación RS-232.

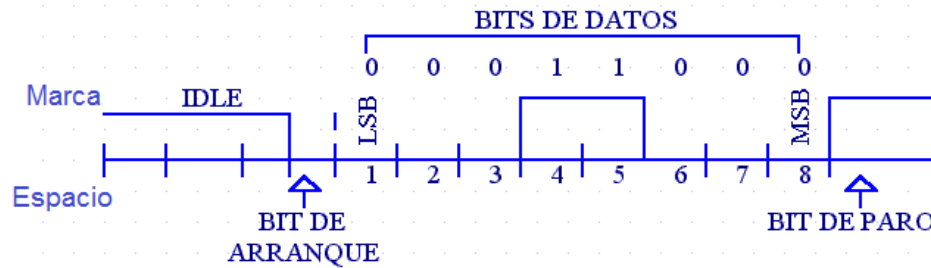


Figura 12. Forma de la trama RS-232.

Para que el reconocimiento de los datos pueda realizarse de forma efectiva, tanto el transmisor como el receptor deben tener la misma configuración de velocidad, paridad, número de bits y bit de parada.

Para asegurar la comunicación y disminuir los efectos de distorsión y ruido a través de grandes distancias, los rangos de tensión con los que trabaja el estándar RS-232, desarrollado por la EIA (*Electronics Industry Association*) van desde: -5V a -15V en el transmisor, y -3V a -25V en el receptor para el “1” lógico, y +5V a +15V en el transmisor, y +3V a +25V en el receptor para el “0” lógico.

## 2.6. El Estándar IEEE 802.15.1 (Bluetooth)

Bluetooth es una tecnología orientada a la comunicación y transferencia de datos de forma inalámbrica entre una diversa variedad de dispositivos, tales como PCs, PDAs, Teléfonos móviles, Electrodomésticos, etc. Fue desarrollada inicialmente por *Ericsson Mobile Communications*, y luego continuado por el grupo conocido como SIG (*Special Interest Group*), que ha incorporado la participación de más de 14.000 empresas a lo largo del mundo.

El Bluetooth se ha desarrollado como una especificación abierta para poder asegurar la conectividad de una amplia gama de dispositivos sin importar sus fabricantes o sus tecnologías, permitiendo el reemplazo de cables, y con esto, una reducción en costo, consumo de potencia y complejidad. El estándar que engloba al Bluetooth es el IEEE 802.15.1.

### **2.6.1. Funcionalidad**

El Bluetooth opera en la banda de frecuencias que se encuentra entre 2,4 y 2,5 MHz aproximadamente, la cual es conocida como la banda ISM (Médico-Científica Industrial), aunque existen restricciones en ciertos países, como Francia, Japón y España. La banda ISM es abierta en todo el mundo, y por lo tanto no requiere licencia.

Dos dispositivos se pueden sincronizar cuando se encuentran dentro del radio apropiado, en cuyo caso forman una picored. Una picored puede estar compuesta de hasta 8 dispositivos enlazados. Cuando una picored se conecta con otra picored se forma una red dispersa o Scatternet. En toda comunicación siempre debe haber al menos un maestro, que es quien inicia la conexión, y al menos un esclavo, aunque es posible que un maestro tenga a 2 o más esclavos. Un dispositivo puede actuar de maestro en una picored, y al mismo tiempo estar participando en otra picored como esclavo o como maestro, más no es posible que haya más de un maestro en una picored.

Cuando están enlazados, la comunicación la realizan haciendo saltos aleatorios en frecuencia, lo cual ocurre a 1600 saltos/seg. Cada unidad dentro de la picored posee una identidad de maestro y un reloj nativo, los esclavos le agregan un ajuste a su reloj para sincronizar con el maestro. La frecuencia del reloj nativo siempre es constante, pero los ajustes que se le añaden solo duran mientras exista la conexión.

### **2.7. El $\mu$ Logger 3.x de TECNUM Electrónica**

La unidad de adquisición de datos que ha sido tomada como referencia para el desarrollo del actual proyecto ha sido el  $\mu$ Logger en su versión 3, desarrollado por TECNUM Electrónica, del cual, la revisión más actual es la 3.2. Esta unidad es una versión completa que cuenta con características de portabilidad y bajo consumo, y está basada en un microcontrolador de 8 bits.

Entre las características más resaltantes del equipo se encuentran:

- Consumo de 65  $\mu$ A en promedio en modo stand-by y 29,6 mA promedio en modo activo.
- Temporización mediante RTC (externo al microcontrolador), respaldado por batería de litio (independiente de la alimentación principal).
- 8 canales analógicos de 12 bits (externos al microcontrolador).
- 4 canales analógicos de 10 bits (internos del microcontrolador).
- 8 canales digitales de entrada (2 de ellos con funcionalidad para pluviómetros tipo balancín).
- 2 canales digitales de salida.
- Adaptable a múltiples sensores de diversas marcas.
- Almacenamiento en tarjetas flash extraíbles tipo Secure Digital (hasta 1 GB), formato FAT16.
- 1 puerto UART (interno del  $\mu$ C) y capacidad de extensión a otros 2 puertos UART (externos al  $\mu$ C).
- 1 puerto RS-485.
- 1 puerto SDI-12, para sensores inteligentes.
- Capacidad para manejar equipos de transmisión satelital tipo GOES u ORBCOMM.



*Figura 13. El  $\mu$ Logger 3.x de TECNUM Electrónica.*

Debido a todas sus características, el equipo es el principal producto con el que cuenta la compañía TECNUM para la instalación de estaciones meteorológicas. Sin embargo, existen aplicaciones sencillas en las que este equipo resulta poco conveniente debido a las siguientes razones:

- El microcontrolador que es el corazón del equipo tiene un encapsulado de dimensiones grandes, y debe tener que incorporar en el circuito un expansor de puertos.
- El consumo es relativamente elevado para poder alimentar al equipo con simples baterías AA o AAA.
- No en todas las aplicaciones son necesarias las características que tiene el equipo, lo que lo convertiría en dichos casos en una solución sobredimensionada.
- La velocidad de descarga de los datos está limitada, debido a la baja velocidad de comunicación de los UART.



## CAPÍTULO 3. DESCRIPCIÓN DEL EQUIPO

### 3.1. Características del Equipo (Nanologger v1.0x)

A lo largo del proyecto, durante la etapa de selección de componentes, discusión de alternativas, y desarrollo del firmware, se ha estado definiendo y corrigiendo la estructura física del equipo de adquisición de datos. Durante esos procesos, han sido necesarias ciertas modificaciones para la optimización de los recursos con los que se cuenta.

Al equipo desarrollado se le ha dado el nombre comercial de Nanologger, y éste se encuentra en su versión 1.0, el cual se le considera un prototipo inicial que será puesto a prueba.

El Nanologger puede manejar diversos tipos de sensores comunes en el área de meteorología. Cuenta con un microcontrolador PIC que incorpora varios periféricos internos y controla el encendido y apagado de otros periféricos controlados externamente, lo que le permite el ahorro de energía. Entre sus características se encuentra:

- a) Bajo consumo ( $\sim 6 \mu\text{A}$  en estado de sueño), lo que le permite al Nanologger ser alimentado por baterías AA.
- b) 4 canales analógicos, de los cuales, 2 están destinados a la medición de batería y temperatura del equipo, y las 2 restantes permiten al usuario conectar 2 sensores de múltiples variedades y marcas. En algunos casos puede requerir acondicionadores de señal externos.
- c) 1 canal digital que permite conectar al Nanologger con un pluviómetro tipo balancín o Tipping Bucket.
- d) Reloj en Tiempo Real con Calendario, interno del microcontrolador, que permite al equipo conocer la fecha y la hora en cualquier momento para poderlas almacenar junto al registro de los datos.

- e) Almacenamiento de los datos realizado sobre un módulo de memoria flash serial integrado al equipo.
- f) Un módulo Bluetooth que permite la comunicación entre el Nanologger y cualquier dispositivo Bluetooth que maneje el perfil de puerto serial (SPP). El tipo de comunicación Bluetooth puede ser expandido y desarrollado.
- g) Un puerto de comunicación RS232 que le permite al Nanologger ser configurado desde la PC.

El diagrama de bloques funcionales del Nanologger se muestra en la Figura 14:

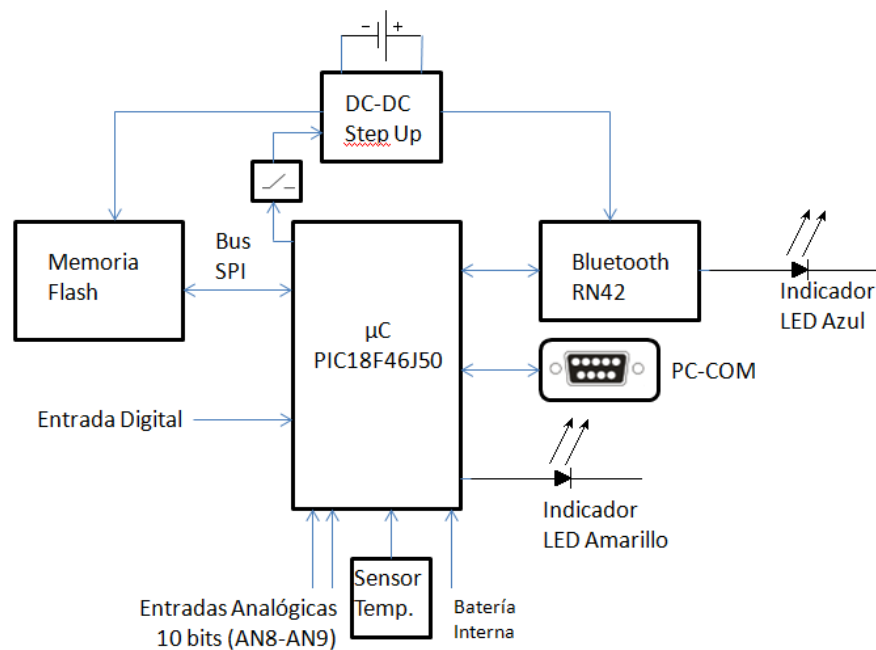


Figura 14. Diagrama de Bloques funcionales del NanoLogger.

Los criterios de selección de componentes del equipo pueden observarse en la sección 4.1.

### 3.1.1. El Microcontrolador

**Los Microcontroladores PIC18F26J50, el PIC18F46J50 y PIC18F47J53 de Microchip**

Los modelos PIC18F26J50 y PIC18F46J50 son 2 controladores pertenecientes a una misma familia, y el PIC18F47J53 presenta las mismas características más algunas prestaciones adicionales, así como compatibilidad de terminales con el 46J50, siendo estos últimos circuitos integrados de 44 pines.

Los microcontroladores antes nombrados poseen varias prestaciones, entre las cuales, se puede nombrar la característica de bajo consumo, esto es, el microcontrolador cuenta con tecnología NanoWatt XLP™, lo que le permitiría reducir su corriente al orden de los nano-Ampere bajo condiciones especiales de mínima actividad. Estos presentan varios niveles de consumo, los cuales son los modos de actividad (Run), espera (Idle), sueño, y sueño profundo.

En el estado de sueño profundo el microcontrolador apaga todos los periféricos salvo el Reloj de Tiempo Real con Calendario (desde ahora, RTCC), un Watchdog especial denominado *DeepSleep Watchdog*, y un detector especial de Brown-out (*DeepSleep Brown-out Reset*). Los únicos servicios que pueden sacar al microcontrolador del estado de sueño profundo son: un reset desde el terminal Master Clear, un evento de Brown-out, una interrupción de la alarma del RTCC, un evento desde el *DS Watchdog* y la interrupción externa 0, y cuando el controlador sale de dicho estado, éste se reinicia, perdiéndose la información que había en la memoria RAM, por lo que se consideró que no era conveniente hacer uso del estado de sueño profundo para el proyecto.

Los controladores elegidos para el proyecto también disponen de varias opciones de oscilador: éste puede ser tanto externo (mediante un cristal) o interno (configurado mediante software y acelerado mediante PLLs internos del microcontrolador). Además de ello se puede colocar al terminal del Temporizador un cristal externo adicional de 32 KHz para suministrar la frecuencia del RTCC, así como también se le podría programar dicha frecuencia a través de un circuito RC interno del microcontrolador, con menor precisión.

De los microcontroladores antes mencionados, el PIC18F26J50 y el PIC18F46J50 cuentan con un convertidor analógico digital de 10 bits de resolución, contando con 10 y 13 canales analógicos respectivamente. El PIC18F47J53 cuenta con 13 canales analógicos y su convertidor puede ser configurado para 10 y 12 bits de resolución.

Ninguno de los microcontroladores seleccionados para el proyecto cuenta con memoria EEPROM, sin embargo la memoria de programa es del tipo Flash, a la cual se le puede reservar regiones de memoria para ser utilizadas para el mismo fin de una memoria no volátil de datos. Las memorias flash de estos microcontroladores presentan la característica de durabilidad de 100.000 ciclos de borrado/escritura, lo que los hace incapaces de soportar funciones de actualización de datos con alta frecuencia.

Los PIC18F26J50, PIC18F46J50 y PIC18F47J53 cuentan con una funcionalidad que les permite la dinámica reasignación de funciones para los terminales de E/S (Re-mapeo de terminales a través del PPS – *Peripheral Pin Select*), lo que permite mayor flexibilidad en el diseño del hardware.

Entre los módulos incorporados, se tiene el de USB, al que no se le ha desarrollado funcionalidad en este proyecto, pero que pudiese desarrollarse en futuras mejoras. El procesador también posee el módulo de ECCP, el de Capturador/Comparador/PWM (sin uso para el proyecto), el módulo MSSP (el cual tiene 2 puertos SPI y 2 puertos I<sup>2</sup>C), con los que se puede cubrir las necesidades de la comunicación serial entre componentes del Hardware, y se tiene al módulo EUSART, el cual trae 2 puertos USARTs, los encargados de la comunicación serial/RS232.

Los microcontroladores PIC18F46J50 y PIC18F47J53 poseen adicionalmente un puerto paralelo maestro de 8 bits, y puerto esclavo (PMP/PSP), actualmente no usado para el proyecto.

La unidad de adquisición de datos se ha desarrollado con firmware compatible con los anteriores microcontroladores descritos, aunque se tiene previsto hacer uso principalmente del PIC18F46J50.

### **3.1.2. Reloj en Tiempo Real con Calendario (RTCC)**

Cualquiera de los PIC18 elegidos para el proyecto trae incorporado un RTCC que le suministra en todo momento la fecha y la hora, codificada en BCD, con una precisión ajustable por diversos medios. Éste suministra los datos de segundos, minutos, horas, día de la semana, día, mes, y año. El formato de la hora siempre se encuentra en 24 horas. El RTCC dispone de una alarma activable y configurable para distintos intervalos de tiempo que es capaz de interrumpir al microcontrolador.

El RTCC presenta un consumo bajo, de aproximadamente 800 nA, por lo que se le puede dejar permanentemente encendido, y al ser interno no puede ser energizado con una batería exclusiva, a diferencia de los RTC convencionales, sin embargo, con la batería principal se le estima un tiempo de vida de varios años para el equipo completo (Nanologger).

### **3.1.3. Conversión Analógica Digital**

La tensión de referencia para el convertidor es fijado en 2,5 V mediante el uso de un regulador zener modelo LT1009. Debido a que la resolución del convertidor A/D es de 10 bits, la conversión se hace de 0 a 1023. Sin embargo, el proceso de lectura sobre el canal analógico se realiza tomando 64 muestras de éste, las cuales se suman, y se dividen entre 16, de ese modo, la nueva lectura poseerá una escala de  $1024 \cdot 64 / 16 = 4096$ , es decir, de 0 a 4096.

Para poder lograr un consumo óptimo de energía, ciertos dispositivos ( Zener, e instrumentos de mediciones) permanecerán apagados normalmente y serán alimentados solo cuando sea necesario su uso. Para ello se utilizarán transistores tipo FETs, controlados por el microcontrolador, que energizarán a estos elementos sólo al momento de hacer la medición y durante el intervalo de tiempo que sea necesario

para asegurar una medición precisa. Dicho tiempo puede ser configurado por el usuario, ya que puede depender de los instrumentos de medición.

El Nanologger contará con 4 canales analógicos, de los cuales 2 son de uso general, además de uno para medir la batería de alimentación del equipo y uno para medir la temperatura interna.

### **3.1.5. Módulo de Memoria**

La memoria Flash serial SST25VF064C de Microchip que cumplirá el propósito de medio de almacenamiento de datos de la unidad de adquisición desarrollada para el presente proyecto cuenta con las siguientes características:

- Rangos de tensión de 2,7V a 3,6V.
- Compatible con SPI, modo 0 y modo 3.
- Durabilidad: 100.000 ciclos de escritura/borrado. Más de 100 años de retención de datos.
- Consumo de corriente en modo stand-by: 5  $\mu$ A.
- Capacidad de borrado: en sectores de 4KB, 32KB y 64KB.
- Rapidez de borrado: borrado total de memoria – 35 ms, borrado de sector – 18 ms.
- Programación de páginas (escritura de datos): 256 bytes por cada página.

Entre todas las características de las que dispone este circuito integrado, hay una que resulta desventajosa para los propósitos del proyecto, y es la durabilidad, pues se considera que 100.000 ciclos de escritura/borrado podrían resultar insuficientes para garantizar un tiempo de vida en algunos casos extremos en los que se desee configurar a la unidad para realizar muchas adquisiciones al día. Esta desventaja puede ser resuelta mediante software y dicha solución será descrita a lo largo del capítulo 4.

### **3.1.6. Módulo Bluetooth**

El módulo Bluetooth RN-42 de Roving Networks, el cual se desea incorporar a la unidad de adquisición de datos para la comunicación inalámbrica con tecnologías móviles, cuenta con las siguientes características:

- Bluetooth versión 2.1 + EDR (Tasa de datos mejorada). Retrocompatibilidad con versiones 2.0, 1.2 y 1.1.
- Consumo de: 26  $\mu$ A en modo stand-by, 3 mA conectado, y 30 mA transmitiendo.
- Interfaces de conexión por UART (SPP y HCI) y USB (HCI).
- Certificación Bluetooth SIG.
- Velocidades de baudios: desde 1.200 bps hasta 921 Kbps Pueden ser programadas tasas de baudios no estándar.
- Clase 2, distancia de 20 metros.
- Frecuencias 2,402 – 2,480 MHz
- Modulación FHSS/GFSK, 79 canales a 1 MHz de intervalo.
- Comunicaciones seguras, encriptación de 128 bits.

### **3.1.7. Fuente de Tensión**

Originalmente no se había planteado el uso de una fuente de tensión para el proyecto, debido al consumo de energía adicional que éste supondría, sin embargo, debido al uso de componentes con distintos rangos de tensión, donde unos limitarían la funcionalidad completa de la unidad de adquisición, se hizo necesaria la inclusión de una fuente elevadora de tensión (step-up) para poder obtener el máximo de provecho a la batería.

El circuito integrado escogido para ejercer tal función ha sido el MCP1623. En la Figura 15 se puede observar el diagrama esquemático de una aplicación típica del MCP1623 en una fuente Step-up, extraída a partir de la hoja del fabricante:

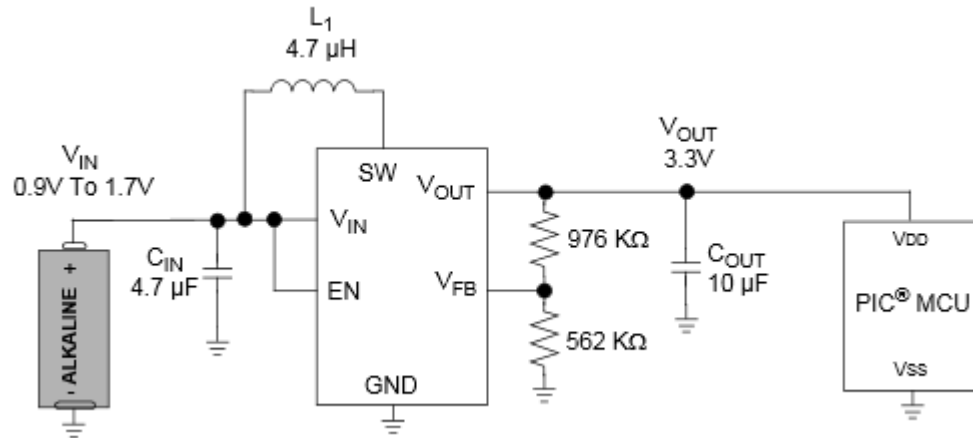


Figura 15. Diagrama esquemático del convertidor DC-DC Step-Up MCP1623/24 alimentando a un Microcontrolador.

### 3.1.8. Transistores FET

Para que el microcontrolador pueda ejercer el encendido y apagado de ciertos dispositivos, ha sido necesaria la inclusión de transistores de conmutación con el propósito de la reducción del consumo de corriente durante los estados de estabilidad o de no-conmutación. Se ha escogido el uso de dispositivos FET (o *transistores de efecto de campo*) debido a que poseen una baja resistencia en estado de saturación, lo que garantiza una baja caída de tensión debido a la incorporación de este componente.

Los FET que se han utilizado para la conformación de los conmutadores del sistema han sido los BSH202, transistores de canal p, cuya resistencia drain-source durante el estado de saturación es de  $0.9 \Omega$ .

## 3.2. Definición del Software

El equipo electrónico que será utilizado y comercializado por TECNUM bajo el nombre de Nanologger involucra 3 componentes de software, los cuales son el firmware para el equipo, el software para el ordenador, y el software para la unidad móvil. El firmware para el equipo ha sido desarrollado durante el presente proyecto,



mientras que ya se contaba con un software previo para la comunicación con la computadora, y se hicieron distintas pruebas para la comunicación con unidades móviles (como teléfonos inteligentes) que disponían de software de emulación de hyperterminal.

Actualmente existe un software que permite la comunicación entre un computador y un equipo registrador de datos, el cual es el PCLogger de TECNUM Electrónica; si el desarrollo del Nanologger se orienta en la dirección de la comunicación aceptada por el mismo protocolo del PCLogger, no sería necesario en lo absoluto el desarrollo de un software de computadora para éste.

### **3.2.1. El Firmware**

El desarrollo del firmware constituye a nivel de software el objetivo principal del proyecto. Se desea que el firmware tenga la capacidad del manejo de todas las tareas esenciales de un equipo registrador de datos, lo cual significa que debe poder realizar tareas de lecturas de canales analógicos por intervalos de tiempo configurables por el usuario, así como también la extracción de los datos adquiridos mediante alguna línea de comunicación.

Así mismo, se desea asegurar la compatibilidad de la unidad de adquisición de datos con el software de configuración previamente desarrollado por TECNUM para sus equipos registradores. La unidad dispondrá de un conjunto de variables que serán configuradas y/o monitoreadas por dicho software desde la computadora.

El formato bajo el cual se almacenan los datos en la memoria flash serial no se considera relevante, pues esta memoria no podrá ser leída directamente por ningún otro dispositivo salvo la unidad de adquisición, sin embargo, para poder lograr compatibilidad con la actual versión del software de comunicación, es necesario cuidar del formato bajo el cual son transferidos dichos datos y los requerimientos del envío de la información.

### **3.2.2. Software de Configuración**

El software disponible, desarrollado por TECNUM, es el PCLogger, el cual se encuentra en la versión V1.11.1124.1, y permite la configuración del antecesor de la unidad de adquisición planteada para el presente proyecto, el  $\mu$ Logger v3.x. Este software fue desarrollado en base al .NET framework, utilizando el lenguaje C#.

El PCLogger ofrece una interfaz gráfica sencilla, que permite la comunicación entre la computadora y la unidad de adquisición, pudiendo realizar cualquiera de las siguientes operaciones: lectura y modificación de fecha y hora, configuración y supervisión de adquisición, identificación, y lectura de archivos de datos. El software fue desarrollado con compatibilidad para distintas versiones de Windows, así como para otros Sistemas Operativos como Mac OS y Linux.

Los cambios que se necesitarían hacer en el software de configuración para lograr la óptima funcionalidad con la nueva unidad de adquisición son muy pocos, y no se encuentran en los alcances del actual proyecto.

## **CAPÍTULO 4. CARACTERÍSTICAS PARTICULARES**

### **(PROCESO DE DESARROLLO)**

Durante la fase previa al diseño y desarrollo del equipo de adquisición de datos se ha hecho un estudio de las distintas gamas de componentes electrónicos para poder realizar la selección de los que presenten las mejores características para el logro de los objetivos planteados.

#### **4.1. Estudio y Selección de Componentes. Criterios de Selección**

Los objetivos del proyecto plantean el desarrollo de un equipo con las siguientes características:

- a) Bajo consumo.
- b) Capacidad de conversión Analógica-Digital. Al menos 2 canales analógicos.
- c) Al menos 1 canal digital.
- d) Capacidad de reconocimiento de fecha y hora actual en todo momento en el que se requieran dichos datos.
- e) Capacidad de comunicación serial, vía RS-232 (para la conexión con una computadora).
- f) Compatibilidad con tecnologías modernas de comunicaciones (inalámbricas, móviles).
- g) Capacidad de almacenamiento de al menos 1 MB.

Por lo tanto, para poder asegurar los objetivos, la selección de los componentes se ha encaminado en la dirección las anteriores características. A continuación se expone, uno a uno, el estudio y selección de cada uno de los componentes.

#### **4.1.1. Criterios generales de selección**

En el estudio de cada uno de los componentes se ha tomado en consideración una serie de parámetros importantes de análisis para la elaboración de tablas comparativas. Algunos de éstos, los cuales son comunes para todos los componentes, son:

##### **4.1.1.1. Rango de Tensiones de Alimentación, corrientes de operación y estado de bajo consumo**

Tanto el rango de tensiones de alimentación como las corrientes de consumo serán importantes a la hora de determinar la duración de la batería de la Unidad de Adquisición de Datos. Sin embargo, para los integrados que actúan como esclavos del microcontrolador, las corrientes de consumo no son parámetros tan prioritarios puesto que se diseñará una estructura electrónica tal que se mantengan sin alimentación durante el período de inactividad, en cuyo caso, la corriente sería cero indistintamente de la magnitud de la del estado de bajo consumo especificada por sus fabricantes.

En cuanto a las tensiones, se puede recurrir a fuentes elevadoras de auxiliares de energía (independientes de la alimentación del microcontrolador) para mantener operativos los módulos esclavos, y de esa manera, no requerir de rangos de tensión tan amplios para éstos.

##### **4.1.1.2. Rangos de Temperatura**

Es un parámetro con prioridad baja, siempre y cuando todos los componentes que conforman la circuitería del equipo se encuentren dentro de un mismo rango de temperatura.

##### **4.1.1.3. Capacidad Integrada de Comunicación Serial**

Resulta fundamental para el microcontrolador disponer de un manejador físico de protocolo ya que en caso de no tenerlo, habría que incorporar uno externo, lo que aumentaría complejidad electrónica, o realizar un programa que lo haga de forma

manual, lo que aumentaría de forma innecesaria la complejidad del software. Por otro lado, la disponibilidad del protocolo presente en las opciones de memorias y módulos Bluetooth condicionará a la escogencia del microcontrolador y viceversa.

#### **4.1.1.4. El Costo del Circuito Integrado y la Disponibilidad**

Parámetro esencial que influirá enormemente en la elección del componente, puesto que es importante no solo un dispositivo de bajo costo sino también disponible en el mercado.

#### **4.1.2. El microcontrolador.**

El microcontrolador es la unidad central de todo equipo moderno capaz de ejecutar un proceso inteligente. Diversos fabricantes han sido tomados en consideración a la hora del análisis de toda la gama de microcontroladores posibles para el diseño del equipo de adquisición.

Los siguientes criterios han sido tomados en consideración para la construcción de las opciones a considerar:

##### **4.1.2.1. Máxima frecuencia del Oscilador.**

Aunque no es la característica más prioritaria, se le consideró un parámetro a tomar en consideración a la hora del estudio de los  $\mu\text{C}$ . Aunque las velocidades altas de oscilación implican mayor consumo, permitirán al microcontrolador realizar más rápido las tareas, y de esa manera, reducir el tiempo de operación activa.

##### **4.1.2.2. Cantidad de Entradas / Salidas**

El  $\mu\text{C}$  que formará parte del equipo debe disponer de la cantidad de puertos o terminales E/S que más se aproxime (por encima) a las necesitadas. Si se llegase a elegir un  $\mu\text{C}$  con menos cantidad de las que se necesita, se necesitaría añadir una expansión de puertos, lo que le añadiría innecesaria complejidad al circuito electrónico.

#### **4.1.2.3. Tamaño de la Memoria RAM y de Programa.**

Elegir un  $\mu\text{C}$  con memorias de datos y de programa muy bajas limitará al firmware del equipo, así como a la cantidad de rutinas que se pueden incorporar a la memoria ó la cantidad de datos que se pueden procesar y almacenar.

#### **4.1.2.4. Módulo de Conversión Analógica/Digital**

Si el  $\mu\text{C}$  dispone de su propio módulo ADC, el diseño del circuito electrónico se simplifica al no requerir la incorporación de componentes adicionales, y por ende, se reducen los costos de producción. Así mismo, la cantidad de canales analógicos y la resolución del ADC son parámetros a analizar en conjunto con esta característica, aunque en el caso particular del actual proyecto, la meta de cubrir al menos 2 canales analógicos no significa mucha dificultad.

#### **4.1.2.5. Reloj en Tiempo Real (RTC)**

Los relojes en tiempo real externos presentan características de consumo elevado, y se encontró que existen microcontroladores con sus RTC integrados, por lo que ésta característica resultó vital para la selección del circuito integrado.

A partir de estos criterios, se ha podido elaborar la Tabla 1, que muestra las opciones que recauda información de diversos microcontroladores, gamas y fabricantes.

Tabla 1. Estudio de las opciones para la elección del microcontrolador.

	PIC16LF1847	PIC18LF14K50	PIC18F46J50	P89LPC915
<i>Fabricante</i>	Microchip	Microchip	Microchip	Phillips/NXP
<i>Rangos de Tensiones de Alimentación</i>	1.8V-3.6V	1.8V-3.6V	2.0 – 3.6 V	2.4 – 3.6 V
<i>Corriente en Activo</i>	34 $\mu$ A/MHz	170 $\mu$ A	2.6 mA	16 mA
<i>Corriente en estado de bajo consumo</i>	30 nA	24 nA	13 nA	1 $\mu$ A
<i>Frecuencia Máxima del Oscilador</i>	32 MHz	48 MHz	48 MHz	18 MHz
<i>Terminales de Entrada/Salida</i>	15	14	20/33	33
<i>Memoria RAM</i>	1024 bytes	768 bytes	3776 bytes	256 bytes
<i>Memoria de Programa</i>	8K Words	16K Bytes	64K Bytes	2K Bytes
<i>Entradas Analógicas</i>	12	11	10	4
<i>Resolución del ADC</i>	10bit	10bit	10bit	8bit
<i>Puerto Serial</i>	EUSART	UART	EUSART	UART
<i>Otros módulos de Comunicación Serial</i>	SPI / I <sup>2</sup> C	SPI / I <sup>2</sup> C / USB	SPI / I <sup>2</sup> C / USB	I <sup>2</sup> C
<i>RTC</i>	-	Si	RTCC	Si
<i>Costo</i>	1.92 USD	2.66 USD	4.00 USD	2.11 USD
<i>Disponibilidad</i>	Si	Si	Si	Si
<i>Sist. De desarrollo adquirido de antemano</i>	Si	Si	Si	Si

De todos los microcontroladores observados en la tabla, han sido resaltadas las opciones más completas, entre las cuales tenemos:

#### 4.1.2.6. El PIC16LF1847 de Microchip

Posee un amplio rango de tensiones de alimentación, características de bajo consumo en modo activo y en modo de sueño, numerosos canales de conversión analógica-digital a 10 bits de resolución, capacidad de comunicación EUSART, SPI e I<sup>2</sup>C y muy bajo costo. No tiene RTC, lo cual lo deja por debajo de otras opciones disponibles.

#### 4.1.2.7. El PIC18LF14K50 de Microchip

Amplio rango de tensiones de alimentación, bajo consumo, mayor frecuencia máxima del oscilador que el anterior, inferior capacidad de RAM pero superior capacidad de memoria de programa, numerosos canales de conversión analógica-

digital a 10 bits de resolución, comunicación EUSART, SPI, I<sup>2</sup>C y USB. No se considera una prioridad el USB, pero representaría posibilidades de mejora en términos de comunicación a futuro. Tiene un RTC incorporado, y su costo, aunque más elevado que el anterior, sigue siendo bajo.

#### **4.1.2.8. La familia PIC18F46J50 de Microchip**

Si bien su rango de tensiones de alimentación es inferior al de los anteriores controladores analizados, la diferencia no resulta muy grande. Las capacidades de la memoria RAM y de la memoria de programa son muy superiores, y además de todas las características similares, éstos traen incorporado un RTC con calendario, lo que los deja como la primera opción.

#### **4.1.2.9. El P89LPC915 de NXP/Phillips**

Este controlador particular fue destacado por la razón de que para el actual proyecto ya se cuenta con un módulo de desarrollo de NXP/Phillips, por lo que si se llegara a elegir a un controlador de la misma casa, estos costos ya estarían cubiertos, y el P89LPC915, de entre todos los controladores encontrados del fabricante, es el único que cuenta con un RTC propio, sin embargo, existen muchas limitaciones entre sus otras características, entre las cuales destaca la baja capacidad de memoria de programa/RAM.

#### **4.1.2.10. El Controlador elegido**

El microcontrolador elegido ha sido el PIC18F46J50, de la casa Microchip. Las razones principales han sido las siguientes: Posee su propio dispositivo RTC, su consumo es muy bajo, posee suficiente memoria para asegurar el desarrollo sin mayores limitaciones, cuenta con todos los dispositivos de comunicación serial necesarios, se cuentan con herramientas para su desarrollo, pertenece a la misma familia de microcontroladores (PIC18) que el usado en el equipo antecesor  $\mu$ Logger 3.2. A pesar que el costo de la unidad es el más alto, esto se compensa por el hecho



de ahorro en dispositivos externos, facilidad de desarrollo, y uso de rutinas previamente desarrolladas.

### **4.1.3. Módulos de Memorias**

Para incorporar a funcionalidad de almacenamiento de datos de (al menos) 1 MB, se ha considerado para el proyecto el estudio de las memorias EEPROM, debido a su simpleza de escritura, topología reducida y bajo costo, así como también a las memorias Flash, debido a las gamas de capacidades que ofrecen.

Los criterios de estudio que se han utilizado para la comparación de las diversas opciones son los siguientes:

#### **4.1.3.1. Tipo de Memoria**

Para la selección del componente de almacenamiento, se tomaron en cuenta los siguientes tipos: las memorias EEPROM, la cuales destacan por la sencillez en la escritura/lectura/borrado y por su durabilidad (ciclos de escritura-borrado), las memorias Flash, cuya principal ventaja es su bajo costo y las capacidades de las que disponen.

También se tomaron en consideración de estudio las memorias tipo serial F-RAM, aunque en dicho estudio se determinó que no tenía ninguna ventaja que las convirtiera en una opción importante.

#### **4.1.3.2. Velocidad máxima de escritura**

Este parámetro pudiera verse limitado por la velocidad de escritura que con que se programe al microcontrolador, por lo que no se considera una prioridad en el criterio para la elección del módulo de memoria.

#### **4.1.3.3. Ciclo de escritura**

Parámetro que sólo existe para las memorias EEPROM, por lo que la prioridad sobre el mismo se considera muy baja.

#### **4.1.3.4. Cantidad de memorias que se pueden conectar en cascada**

Parámetro poco importante, pues la solución de conectar varias memorias en cascada se consideró muy poco elegante para el presente proyecto.

#### **4.1.3.5. Durabilidad (ciclos de escritura-borrado)**

La durabilidad o la capacidad de borrar para luego reescribir si se convierte en un factor importante, pues una memoria con durabilidad baja condicionará enormemente el diseño del firmware de la unidad de adquisición.

#### **4.1.3.6. Retención de datos**

También es un parámetro importante, aunque como se podrá observa en la tabla comparativa, las memorias en su mayoría presentan tiempos apropiados de retención de los datos.

#### **4.1.3.7. Capacidad (Densidad / Organización)**

Uno de los parámetros más importantes de todos, pues el proyecto requiere de una memoria lo más grande posible para garantizar, junto al objetivo del bajo consumo, un tiempo de autonomía considerablemente prolongado para la unidad de adquisición de datos.

Tabla 2. Estudio de las opciones para la elección de la memoria.

	<b>M95M02-DR</b>	<b>M24M02-DR</b>	<b>FM25V20</b>	<b>SST25VF064C</b>
<i>Tipo de Memoria</i>	EEPROM	EEPROM	Serial V-RAM	Serial Flash
<i>Fabricante</i>	ST Microelectronics	ST Microelectronics	RAMTRON	SST Microchip
<i>Rango de Vcc</i>	1.8 - 5.5 V	1.8 - 5.5 V	2.0 - 3.6V	2.7 - 3.6V
<i>Icc de Operación</i>	3.0 mA	2.5 mA	0.30 mA	14 mA
<i>Icc de bajo consumo</i>	5 $\mu$ A	5 $\mu$ A	100 $\mu$ A/3 $\mu$ A	5 $\mu$ A
<i>Tamaño de Página</i>	256 bytes	256 bytes	-	256 bytes
<i>Frecuencia Máxima</i>	5 MHz	1 MHz	40 MHz	80 MHz
<i>Ciclo de Escritura</i>	10 ms	10 ms	-	-
<i>Cascada</i>	2	-	-	-
<i>Durabilidad Ciclos E/L</i>	1.000.000	1.000.000	100 Trillion	100.000
<i>Retención de datos</i>	>40 Years	>40 Years	10 Years	>100 Years
<i>Rango de Temperatura</i>	-40°C to +85°C	-40°C to +85°C	-40°C to +85°C	-40°C to +85°C
<i>Capacidad</i>	256 KB	256 KB	256 KB	8MB
<i>Comunicación Serial</i>	SPI	I <sup>2</sup> C	SPI	SPI
<i>Costo</i>	2.52 USD	2.52 USD	9.37 USD	4.76 USD
<i>Disponibilidad</i>	Si	Si	Si	Si

De las características de las EEPROM se puede observar que, todas las capacidades disponibles de memoria son insuficientes, pues ninguna alcanza a cubrir el mínimo de 1 MB requerido para la unidad de adquisición. Una posibilidad sería poder conectarlas en cascada, para lo que 4 memorias de 256 Kbytes alcanzarían para cubrir los requerimientos de almacenamiento, sin embargo esa solución se considera muy poco atractiva debido a la complejidad que añadiría a la circuitería electrónica. Las memorias F-RAM, son memorias no volátiles, pero también tienen capacidad muy baja, lo que enrumbaría la decisión en el área de las memorias Flash.

Entre estas memorias, hay muchas más opciones disponibles con capacidades de almacenamiento apropiadas para el proyecto. Todas cuentan con el protocolo de comunicación serial SPI, y tienen el mismo encapsulado, así como también disponen de compatibilidad entre terminales, por lo que se pueden reemplazar unas por otras.

La memoria seleccionada para el proyecto ha sido la SST25VF064C de Microchip por ser la de mayor capacidad en formato serial encontrada en el mercado a la vez de adaptarse adecuadamente a los requerimientos del proyecto.

#### **4.1.4. Módulo Bluetooth**

La unidad de adquisición de datos planteada para el presente proyecto necesita de adaptabilidad en la comunicación con tecnologías de actualidad, y se consideró al Bluetooth como la opción más económica y portable para lograr dicho fin, además de ser una opción ampliamente utilizada por un gran número de equipos móviles.

Se han investigado una gama de módulos Bluetooth, tomando en cuenta los siguientes aspectos:

##### **4.1.4.1. Versión de Bluetooth**

Se consideró como un parámetro importante el saber con cuál versión el módulo está capacitado para la comunicación, aún si, las distintas versiones o especificaciones cumplen con la característica de retro-compatibilidad, es decir, las versiones más recientes poseen la capacidad de establecer comunicación con las más antiguas, bajo las limitaciones de éstas últimas.

##### **4.1.4.2. Máxima Tasa de bits**

No se consideró un parámetro fundamental de análisis, pues la comunicación Bluetooth con el dispositivo externo seguramente va a verse limitada por la velocidad de comunicación serial entre el microcontrolador y el módulo Bluetooth; sin embargo se tomó en cuenta al momento de la elaboración de la tabla de opciones.

##### **4.1.4.3. Rango de cobertura (Clase)**

Si se consideró un parámetro prioritario, pues la clase determina tanto la distancia máxima de emparejamiento entre el maestro y la Unidad de Adquisición, como el consumo de energía del módulo, que no es otra cosa sino una consecuencia

de la distancia máxima. Para el proyecto se consideró que no es necesario permitir una distancia de emparejamiento tan grande, en cambio, si es importante mantener un consumo lo más reducido posible, por lo que la clase 2 (intermedia) se estableció como la deseada para el módulo Bluetooth.

*Tabla 3. Estudio de las opciones para la elección del módulo Bluetooth.*

	<b>CSR BC417 (EGBT-045MS)</b>	<b>RN-42/RN-42-N</b>
<i>Fabricante</i>	?	Roving Networks
<i>Vcc operación</i>	3.1 - 4.2 V	3.0 - 3.6 V
<i>Icc operación</i>	40 mA	30 mA
<i>Sleep/Stndby Icc</i>	-	26 $\mu$ A
<i>Bluetooth Versión</i>	V2.0	V2.1/2.0/1.2/1.1
<i>Max tasa de bits</i>	3 Mbps	921 Kbps
<i>Clase</i>	2	2
<i>Comunicación Serial</i>	UART	UART/USB/SPI
<i>Temperatura</i>	-20°C to +55°C	-40°C to +85°C
<i>Costo</i>	6.60 USD	17.65 USD
<i>Disponibilidad.</i>	Muy Baja	Si

De la tabla se observa que, económicamente, el módulo CSR BC417 es quien tiene mejores características, sin embargo, el estudio de disponibilidad lo deja muy por debajo del que le sigue de cerca en la lista, el módulo RN-42, un módulo Bluetooth ampliamente utilizado y encontrado en el mercado. Éste módulo es clase 2, lo cual significa que puede sincronizar con dispositivos Bluetooth a un alcance máximo de 10 metros y debido a ello, el consumo de energía en modo activo es medio. También cuenta con comunicación SPI, USB y UART, las cuales otorgan flexibilidad al diseño del prototipo electrónico de la unidad de adquisición de datos.

#### 4.1.5. Driver RS232

La Unidad de Adquisición de Datos necesitará de una interfaz de comunicación serial con la Computadora a través del protocolo RS232, para poder ser configurada con el software desarrollado para este fin.

Como la tensión de alimentación de la unidad será de 3.3 V, se requiere de un convertidor a RS232 que opere para dicha tensión. Aquello deja fuera de consideración a muchos circuitos integrados convencionales que solo funcionan para 5V (TTL), y entre las opciones que se encontraron se construyó la siguiente tabla:

Tabla 4. Estudio de las opciones para la selección del driver RS232.

	<b>Opto acoplador MCT6</b>	<b>MAX3233</b>	<b>MAX3227</b>	<b>2N3904 y 74HC00</b>
<i>Rango de Tensión de Alim.</i>	-	3.0V - 3.6V	3.0V - 3.6V	2.0V - 6.0V
<i>Consumo en estado de inactividad</i>	-	1 $\mu$ A	1 $\mu$ A	-
<i>Máxima tasa de transferencia</i>	< 19,2 Kbps	250 Kbps	250 Kbps	~3 Mbps
<i>Disponibilidad</i>	Si	Si	Si	Si
<i>Costo</i>	1,05 USD	10,26 USD	2,87 USD	0,6 USD

Los dos componentes que se consideraron para suplir la función de driver o transductor de niveles de tensión fueron el MCT6, un opto acoplador, que cuenta con la ventaja de que no consume corriente durante la actividad, y acepta cualquier rango de tensión, y el circuito integrado MAX3233, que opera para 3.3 V, presenta bajo consumo y permite operar con la tasa de baudios máxima de 115200.

El MAX3233 presentó la desventaja del consumo adicional de corriente de aproximadamente 2,5 mA, tras 30 segundos aproximados luego de finalizada la transferencia de datos. Pero el opto acoplador, módulo utilizado como driver RS232 del  $\mu$ Logger 3.2, unidad de adquisición tomada como referencia para el desarrollo del actual proyecto, presentó la desventaja crucial de que la máxima tasa de transferencia de datos se veía muy limitada, no permitía velocidades altas, como de 115.200

baudios. También se puede ver que el costo del MAX3233 es muy elevado respecto a las otras opciones.

Por otra parte se tiene al MAX3227, que cuenta con todas las características del anterior, a un costo inferior. También se analizó la posibilidad de construir el circuito conversor a RS232 mediante una compuerta NAND basada en CMOS y un transistor 2N3906, logrando para esta opción unas características que satisfacen todas las necesidades de velocidad de transferencia de datos a un costo por debajo del resto de las opciones, por lo que se ha optado por la elección de estos componentes para el conversor (driver) RS232.

## **4.2. Características específicas del Software**

El firmware para el microcontrolador fue creado bajo la plataforma de desarrollo MPLAB IDE® de la casa Microchip, usando el compilador MPLAB C18 (ahora llamado “*MPLAB C Compiler for PIC18*”) del mismo fabricante, que es un compilador de lenguaje C compatible con el estándar ANSI C89.

El firmware está compuesto de un conjunto de códigos divididos en librerías, cada una para manejar una función en particular, todas llamadas desde un archivo principal que sólo se encarga de manejar rutinas de alto nivel. Algunas de las librerías utilizadas pertenecen a Microchip y son concernientes a los periféricos del microcontrolador.

Como ya se mencionó anteriormente, se ha desarrollado a nivel de firmware compatibilidad con los siguientes microcontroladores: PIC18F26J50, PIC18F46J50 y PIC18F47J53, de los cuales, los 2 últimos, de encapsulados de 44 patillas (terminales), disponen de compatibilidad entre terminales.

La estructura del firmware de la unidad de adquisición de datos está basada ampliamente en la de su antecesor, la unidad  $\mu$ Logger 3.x de TECNUM.

### 4.2.1. Estructura del Firmware

Lo primero que hace el programa del equipo de adquisición de datos diseñado al comenzar su operación es Inicializar todos sus periféricos, y luego inicializa sus variables. Una vez terminado el proceso de inicialización, se ejecuta el núcleo del programa, el cual se compone de un lazo infinito, que consiste en la verificación de tareas pendientes por realizar, y, de haber sido todas realizadas, el equipo entra en un estado de bajo consumo hasta que una interrupción lo saque de dicho estado para entrar en una nueva verificación de tareas pendientes.

El diagrama de la figura 16 ilustra el proceso antes descrito:

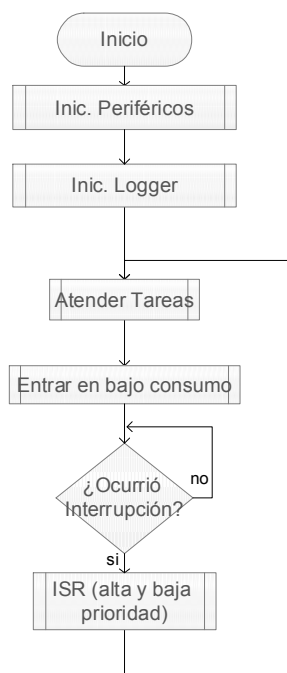


Figura 16. Programa Principal.

El bloque de inicialización de periféricos enciende y configura todos los módulos y periféricos que posee el microcontrolador, de tal forma que éstos estén listos para ser utilizados durante el resto del programa. Entre los periféricos inicializados se encuentran los módulos USART (uno de ellos manejará al Bluetooth, y el otro se comunicará vía RS232 hacia una computadora), la configuración de las



E/S, así como aquellas E/S que necesitan ser activadas y re mapeadas hacia los terminales correspondientes, y el módulo del Reloj en Tiempo Real (RTCC).

Algunos periféricos no son inicializados durante esta etapa, sino sólo al momento en el que se les utiliza para poder ser apagadas inmediatamente, y de esa forma ahorrar consumo de energía. Dichos periféricos son el módulo SPI, la memoria Flash serial, los temporizadores y el ADC.

El bloque de inicialización de variables prepara todas las estructuras que le darán forma al flujo del programa. Los dos diagramas de la Figura 17 ilustran el contenido de dichos bloques:

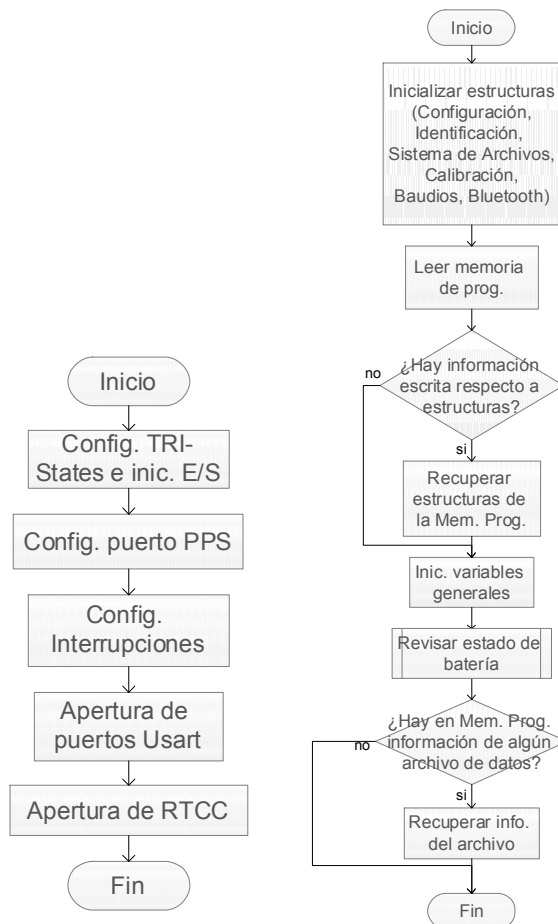


Figura 17. Izquierda: Inicialización de Periféricos. Derecha: Inicialización de variables del Logger.

### 4.2.1.1. Núcleo del Firmware: la rutina Atender Tareas

La rutina *Atender Tareas* es la encargada de administrar la ejecución de las tareas que estén pendientes por realizar, así como de liberar el proceso de atención cuando ya todas las tareas hayan sido realizadas.

La activación de las tareas se lleva a cabo generalmente en rutinas de atención de eventos especiales como el RTC, interrupción por eventos externos (Tipping Bucket), temporizadores, comunicación serial, etc. También pudieran activarse durante la ejecución de otras tareas.

En la Figura 18 se ilustra el proceso que se lleva a cabo mediante la rutina antes descrita:

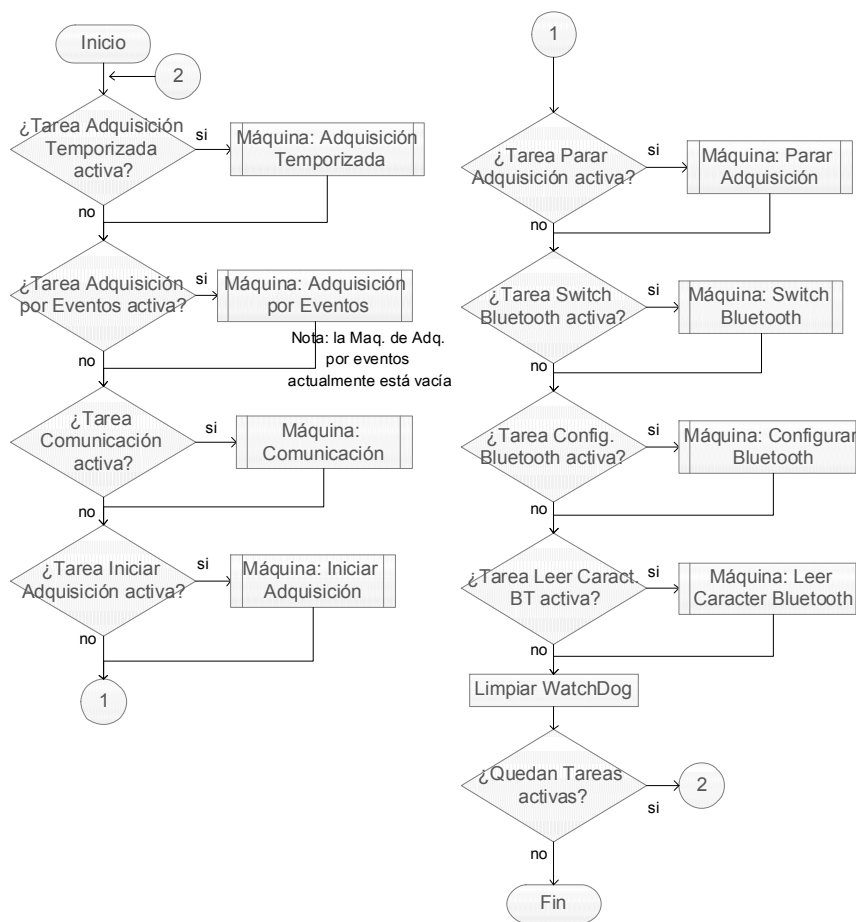


Figura 18. Rutina: Atender Tareas.

En el caso de que la bandera de una tarea esté activa, la rutina administrativa *Atender Tareas* llama a una rutina que funciona como máquina de estados concerniente a dicha tarea. Esta rutina tendrá la potestad de limpiar la bandera de la tarea una vez finalizada.

Mientras aún queden banderas de tareas sin limpiar, la rutina de atención de tareas se mantiene en un bucle llamando a la rutina correspondiente a las tareas por realizar. Las banderas de las tareas pueden ser activadas durante una interrupción, o desde la ejecución de otra tarea.

#### **4.2.1.1.1. Banderas de Tareas y Máscaras**

La rutina de *Atender Tareas* trabaja con banderas especiales que sirven como condición para ejecutar cada una de las subrutinas (máquinas de estado) asociadas a una tarea. Éstas son las banderas de Tareas y Máscaras.

Cada subrutina tiene asociada una bandera de tarea y una bandera de máscara. La rutina *Atender Tareas* ejecutará cada subrutina si la operación AND entre la bandera de tarea y la bandera de máscara (asociadas a la subrutina) da como resultado verdadero (“1” lógico), o lo que es lo mismo, ambas están en “1”. Esto se hace para que sea posible “enmascarar” tareas (hacer su bandera de máscara cero), o lo que es lo mismo, detener su ejecución sin necesidad de levantar su bandera de tarea. De ésta forma, es posible “desenmascarar” una tarea desde cualquier parte del código sin que ello afecte su estado de activa o inactiva.

#### **4.2.1.1.2. Máquinas de estados del Nanologger**

Todos los procesos que es capaz de ejecutar la unidad de adquisición de datos se encuentran distribuidos a través de las máquinas de estado, que son gestionadas por el administrador de tareas, que se encargan desde la adquisición hasta la comunicación serial con un equipo externo mediante el estándar RS232. Las máquinas de estado solo ejecutan un estado y luego retornan al administrador de tareas, quien los volverá a llamar en caso de que la tarea no haya finalizado. De esta

manera se logra que el CPU distribuya su capacidad de procesamiento entre múltiples tareas que requieren atención en forma simultánea.

#### 4.2.1.3. La máquina de Adquisición Temporizada (o por intervalos)

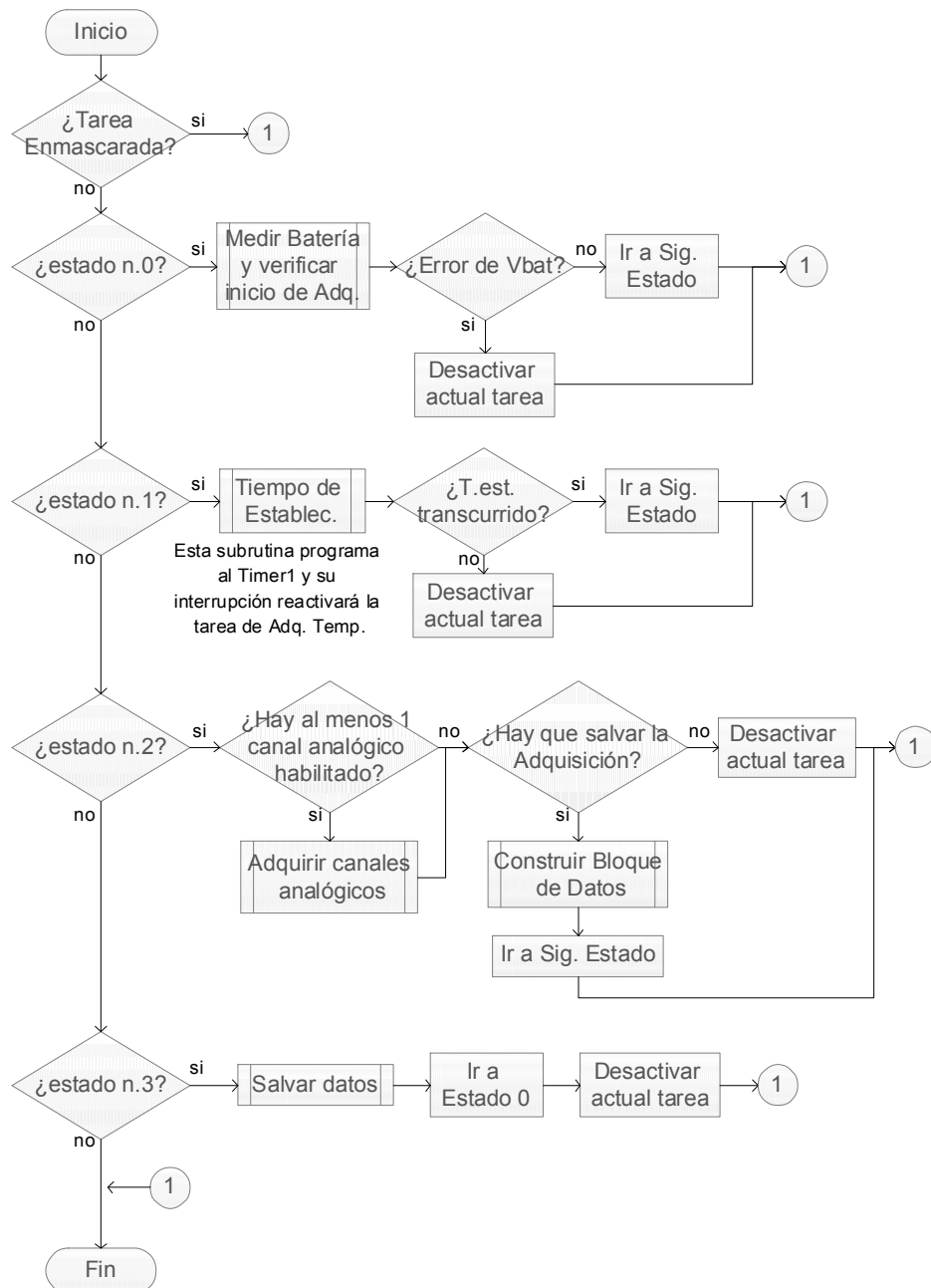


Figura 19. Máquina de Adquisición Temporizada.

Esta máquina contiene el proceso más importante del logger, pues es la encargada de realizar las lecturas de los canales y registrar los datos en la memoria. Esta tarea es activada por la rutina de atención a la interrupción del RTCC, la cual a su vez es iniciada una vez que el usuario envía por puerto serial a la unidad de adquisición el comando “Iniciar Adquisición”. Esta tarea también puede ser activada en forma “forzada” para adquirir datos en forma asincrónica, por ejemplo, durante la supervisión de variables.

La máquina de Adquisición temporizada se divide en varios subprocesos, los cuales son: medición de batería, fijar tiempo de establecimiento, adquisición y registro de datos. El primer subproceso, medición de batería, arrojará de salida una respuesta booleana que indicará si hay que seguir con la adquisición, o si por el contrario, lo mejor es parar debido a la deficiencia en la batería.

En el subproceso de medición de batería que se detalla más adelante se observa que la condición para que se cumpla el caso de la batería deficiente es que se detecte 5 veces seguidas que el valor de la tensión de batería es inferior a un valor mínimo establecido.

Una vez que la batería ha sido medida, el siguiente estado correspondiente de la máquina es el del tiempo de establecimiento, y lo que hace es configurar un temporizador para esperar durante un tiempo pre configurado. El propósito de fijar esta espera es darle tiempo a los posibles medidores conectados al logger a que se estabilicen, tras haberse encendido durante el proceso de medición de batería.

El tercer subproceso de la máquina es el de la adquisición, que verifica cuales canales analógicos y digitales están habilitados para poder realizar las lecturas correspondientes. También se verifica la bandera de salvar adquisición, la cual en caso de estar activa, se habilita el siguiente estado, que corresponde a salvar en memoria las lecturas realizadas. Dicha bandera es habilitada si la tarea de adquisición fue producida por RTCC, y si no lo está, es porque la adquisición está forzada, en

cuyo caso las lecturas solo se desean para ser monitoreadas desde un dispositivo externo al logger (computador o dispositivo móvil).

Si el caso correspondiente es salvar los datos, lo que hace el tercer subproceso es llamar a una rutina que construye en RAM el bloque de datos concerniente a las lecturas realizadas y devuelve el tamaño de dicho bloque, que será necesario más tarde, a la hora de deducir el tamaño del archivo, así como el espacio disponible en memoria.

El último subproceso transfiere el bloque de datos a la memoria flash serial externa haciendo uso de la interfaz SPI. Si es la primera adquisición, entonces el subproceso inicia el archivo de datos antes de transferir las lecturas.

A continuación se muestran los diagramas de flujo de cada uno de los subprocesos de la máquina de adquisición:

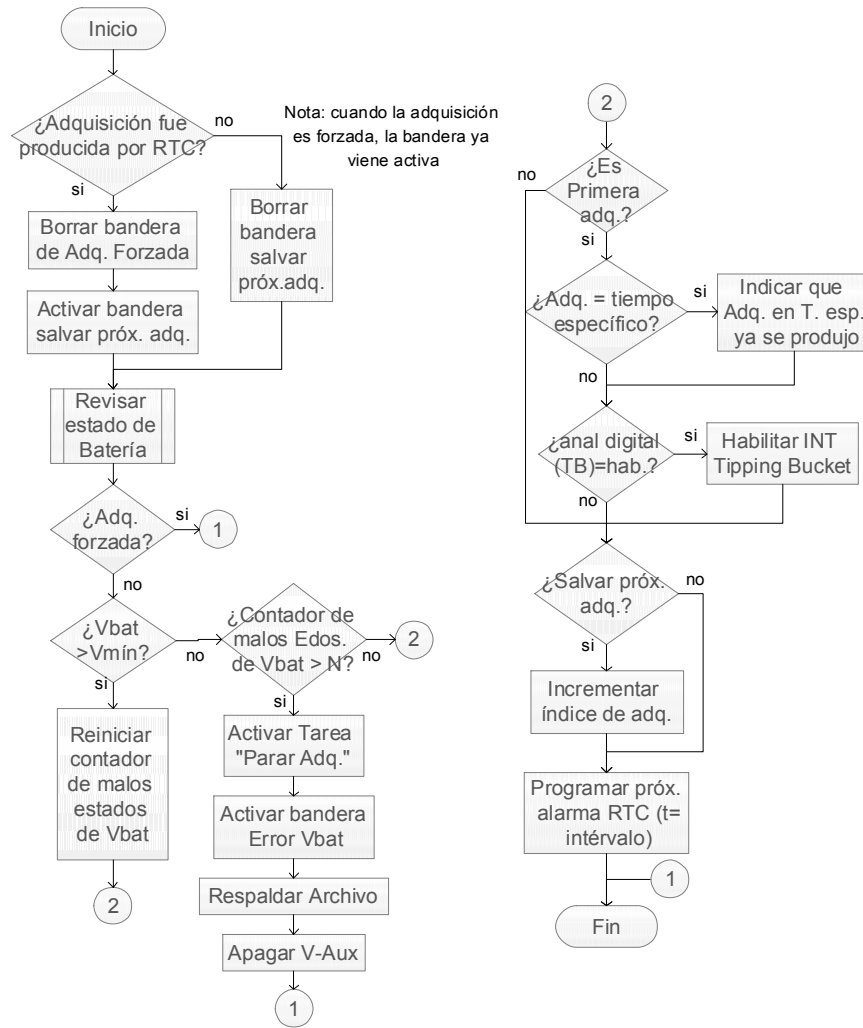


Figura 20. Medición de Batería y Verificación de Inicio de Adquisición.

Esta subrutina comprueba el estado de la batería, tomando las decisiones según el nivel en el que ésta se encuentre. También se encarga de inicializar ciertas variables y periféricos necesarios para la adquisición, y de programar la próxima alarma del RTCC.

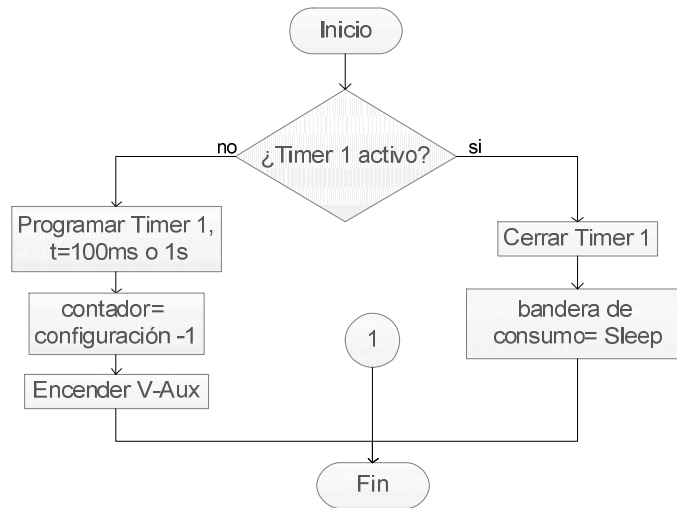


Figura 21. Fijación de tiempo de establecimiento.

La subrutina Tiempo de Establecimiento configura al Timer 1 para contar durante un intervalo que depende del byte de configuración que indique el usuario, y termina la tarea de adquisición para retornar al logger en modo de bajo consumo, y cuando éste es interrumpido por el final del conteo del Timer, entonces en la rutina de atención de la interrupción la tarea es reactivada, y el programa continúa en el estado que se había quedado.



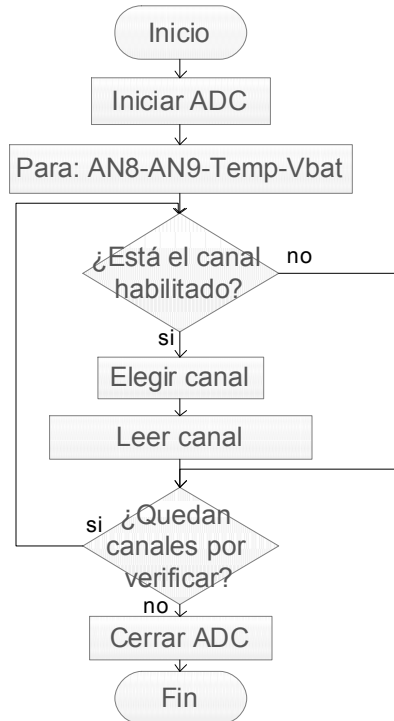


Figura 22. Subrutina: adquirir canales analógicos.

Para cada uno de los canales analógicos, la rutina verifica si el canal ha sido habilitado previamente por la configuración, y de ser así, se selecciona el canal y se realiza una lectura del mismo. El módulo ADC es encendido al inicio de la rutina y una vez que han sido verificados todos los canales, se apaga.

La rutina de construir bloque de datos selecciona la información que va a agrupar dentro de memoria RAM para más adelante transferirla a la memoria flash en bloque. El tamaño de este bloque puede depender de ciertos parámetros de configuración como cuántos canales están habilitados, o si está activa o no la bandera de indexación de adquisiciones, la cual hace que se añada el índice consecutivo de cada adquisición a cada bloque de datos.

El bloque de tamaño máximo reúne los siguientes datos: índice de adquisición (2 byte), lecturas de 4 canales analógicos (2 bytes cada uno, 8 en total), y 2 bytes del canal digital, dando en total 12 bytes.

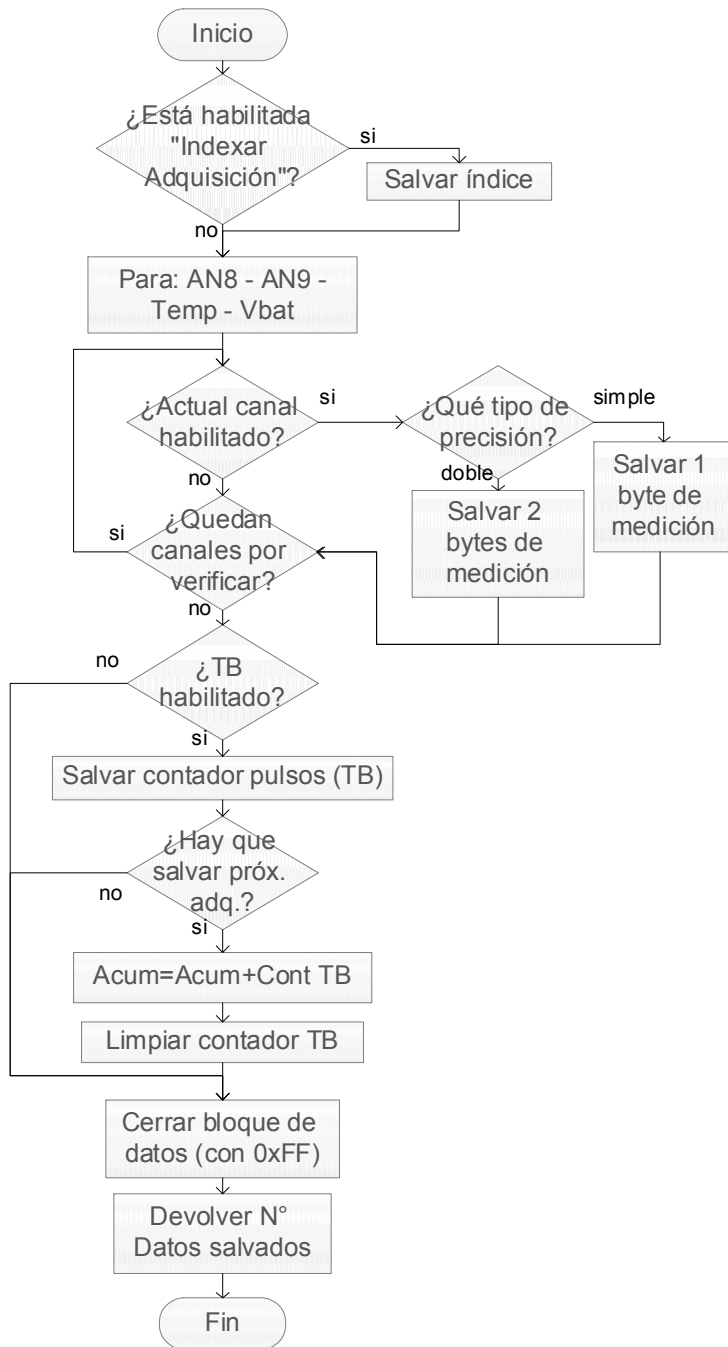


Figura 23. Subrutina: construir bloques de datos.

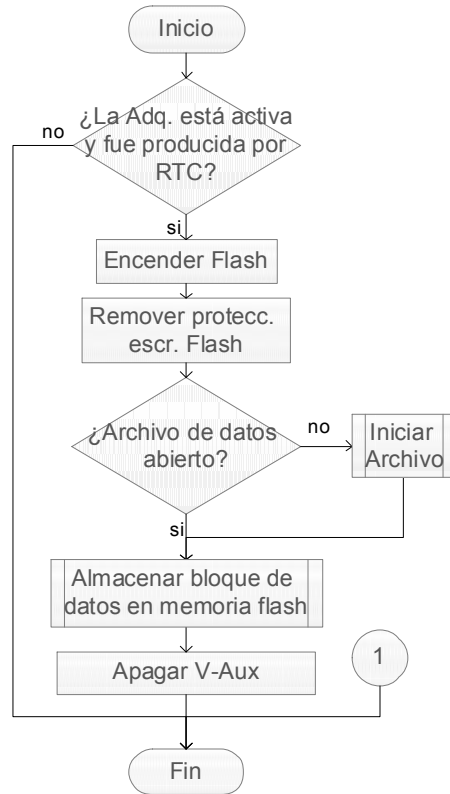


Figura 24. Salvar datos en memoria.

La rutina *Salvar Datos en Memoria* verifica si la adquisición fue producida por RTCC, en cuyo caso enciende la memoria Flash (previamente energizada mediante activación de Vaux) y la inicializa, habilitando la escritura por software, para luego llamar a la subrutina de transferir los datos. Si no ha sido abierto ningún archivo, entonces la rutina llama primero a la subrutina correspondiente de creación de archivo.

El archivo contiene la información básica de un fichero interpretable por una computadora, como el nombre corto y nombre largo, fecha de creación, fecha de modificación, tamaño, atributos, etc. Al apagar Vaux se apaga simultáneamente la memoria Flash y se activa automáticamente el modo de protección de escritura.

A continuación se muestran los diagramas de flujo de las subrutinas llamadas desde la rutina *Salvar Datos*:

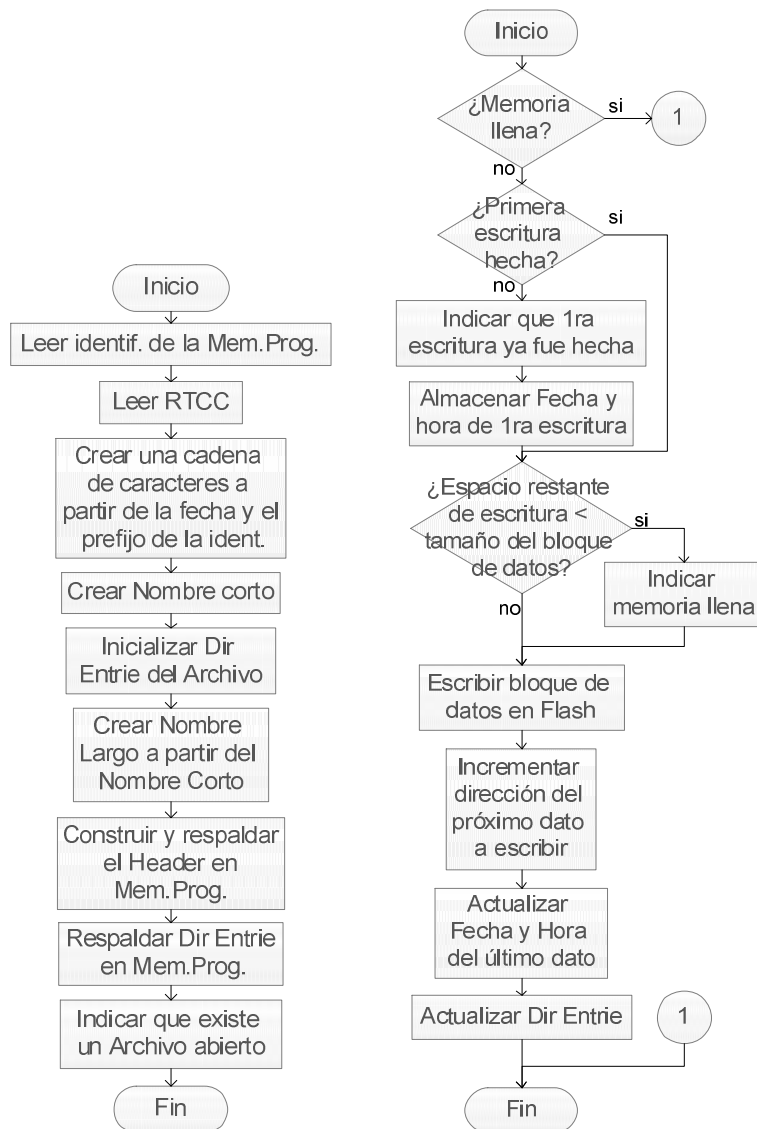


Figura 25. Subrutinas: Creación de Archivo, y Almacenamiento de bloques de datos en memoria flash externa.

#### 4.2.1.4. Máquina de comunicación (USART)

Todos los procesos que son llevados a cabo cuando la unidad de adquisición recibe una serie de caracteres desde cualquiera de sus 2 puertos USART están contenidos en la máquina de comunicación. Los subprocesos de ésta máquina son: lectura, ejecución y transmisión de comandos. Los comandos que puede interpretar el logger obedecen a un protocolo desarrollado por TECNUM.

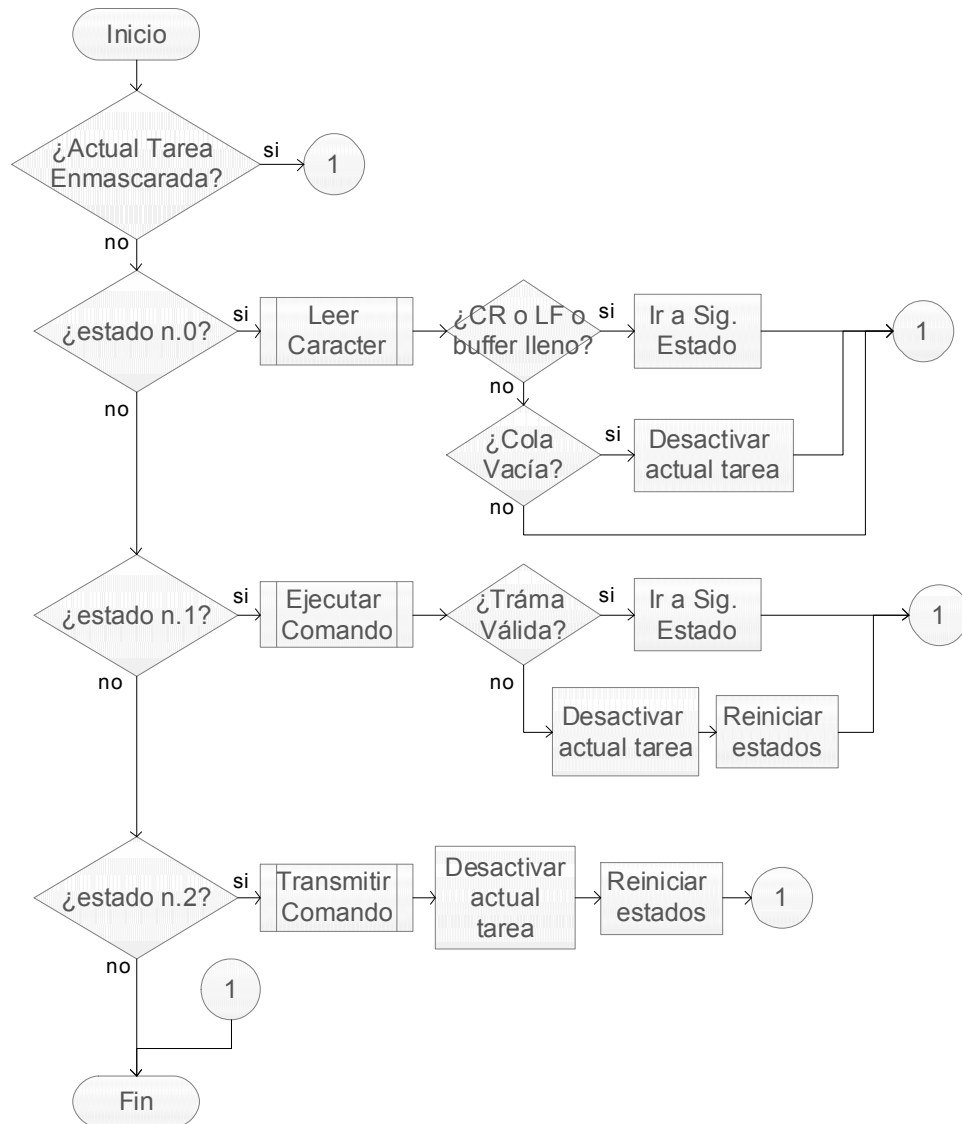


Figura 26. Máquina de comunicación USART.

La tarea de comunicación es activada tras la interrupción de recepción de un carácter por cualquiera de los dos puertos USART. Una vez que el carácter es recibido, éste es ubicado en una cola desde la rutina de atención a la interrupción para ser atendido luego por la máquina.

Una vez llamada la máquina, se pasa a cada uno de los tres estados, de los cuales, el primero llama a la subrutina *Leer Carácter*, que analiza la cola carácter por carácter hasta que ésta es vaciada. Los caracteres allí contenidos son interpretados en

formato ASCII, debido al protocolo de comunicación del logger, y cada 2 caracteres ASCII son convertidos a un byte hexadecimal del protocolo (correspondientes al nibble alto y nibble bajo).

Cada vez que ocurre la recepción de un carácter, y por consiguiente, la subrutina *Leer Comando* es llamada, es configurado un temporizador para contar el tiempo de inactividad de los puertos USART. Si es rebosado el tiempo sin que ocurra actividad en los puertos seriales (ningún carácter es recibido), entonces el microcontrolador es configurado para irse al estado Sleep la próxima vez que pase por la rutina de “entrar en bajo consumo”, ya que se considera que hubo una condición de TimeOut por desconexión en las comunicaciones. De lo contrario, al pasar por la rutina de “entrar en bajo consumo”, el microcontrolador se quedará en modo de espera (Idle), lo que le permite seguir atendiendo entradas de caracteres en el puerto serial. Cuando finaliza el tiempo de inactividad también se apaga el módulo Bluetooth, en caso de que éste se encuentre encendido.

El proceso de interpretación de caracteres es reiniciado si el carácter recibido corresponde con “dos puntos” (“:”), y finaliza cuando es recibido el CR (*Carry Return*) (0x0D), o cuando se rebasa la cantidad máxima de caracteres interpretados. En el caso del CR, es habilitado el segundo estado de la máquina, que llama a la subrutina *Ejecutar Comando*, la cual tiene el propósito de validar el comando recibido, así como de tomar todas las acciones correspondientes al mismo.

Si el comando es válido, el tercer y último estado es habilitado, y la subrutina *Transmitir Comando* es llamada, la cual tiene el propósito de enviar por el puerto serial (en el que fue recibido) el comando la respuesta que se ajusta al protocolo de comunicación.

A continuación se muestran los diagramas de flujo correspondientes a las subrutinas *Leer Carácter*, *Ejecutar* y *Transmitir Comando*:

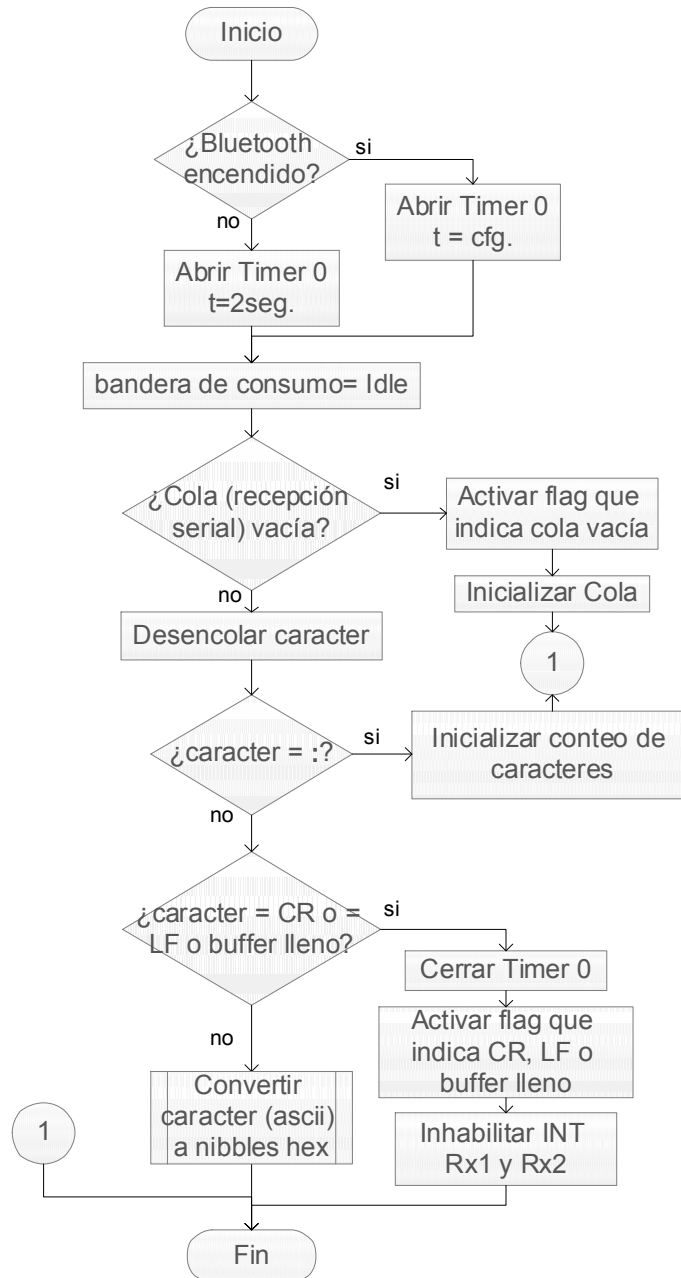


Figura 27. Leer Carácter.

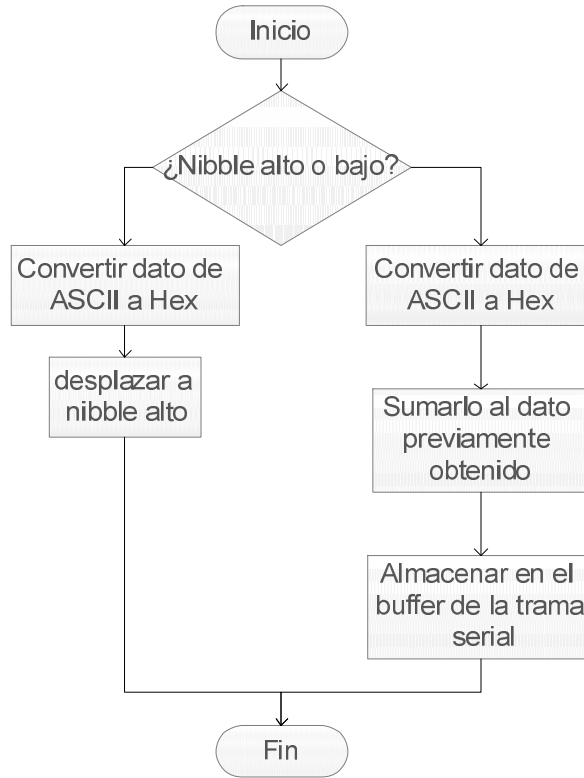


Figura 28. Subrutina: convertir carácter ASCII a Nibles Hex.



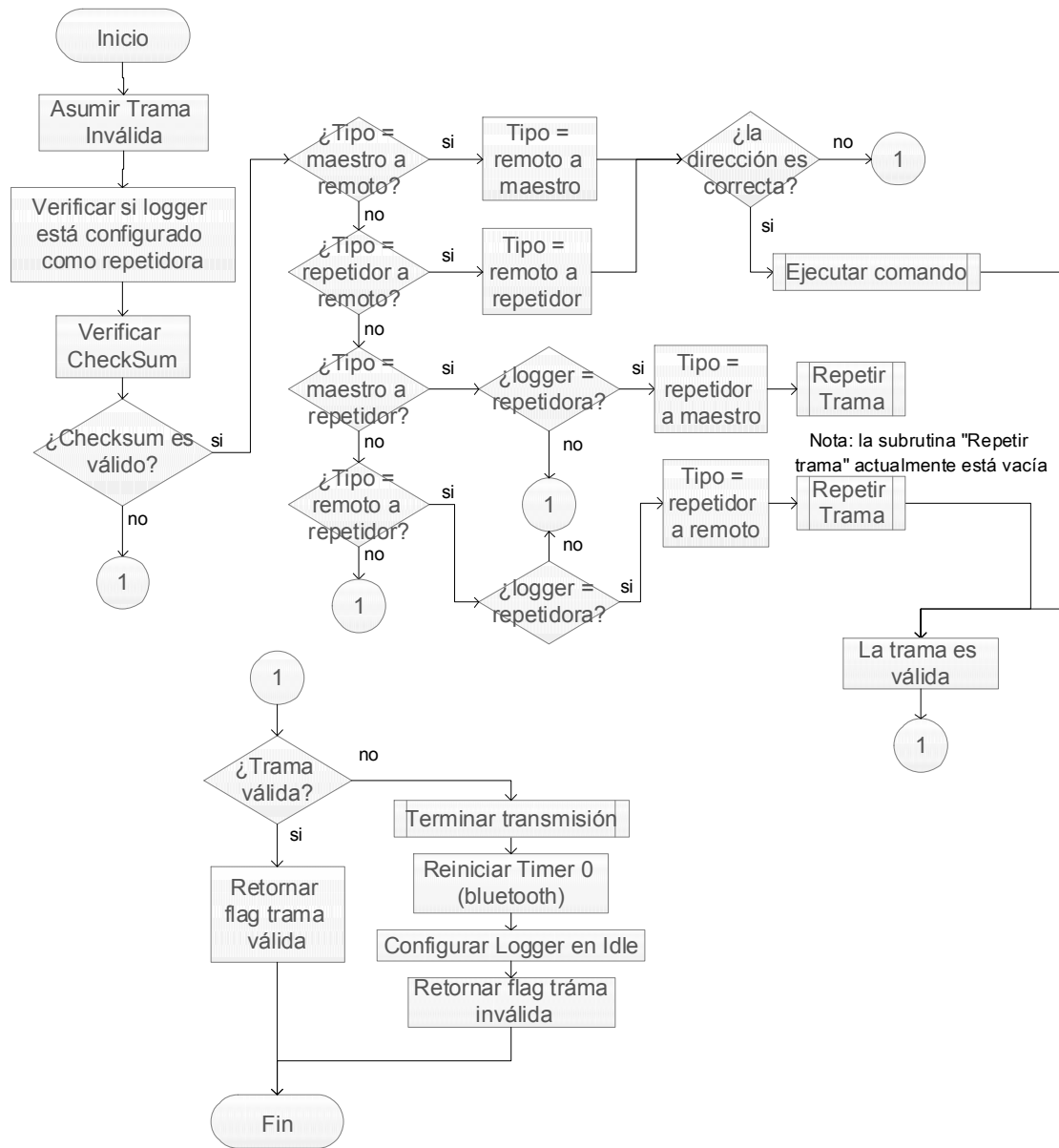


Figura 29. Ejecutar Comando.

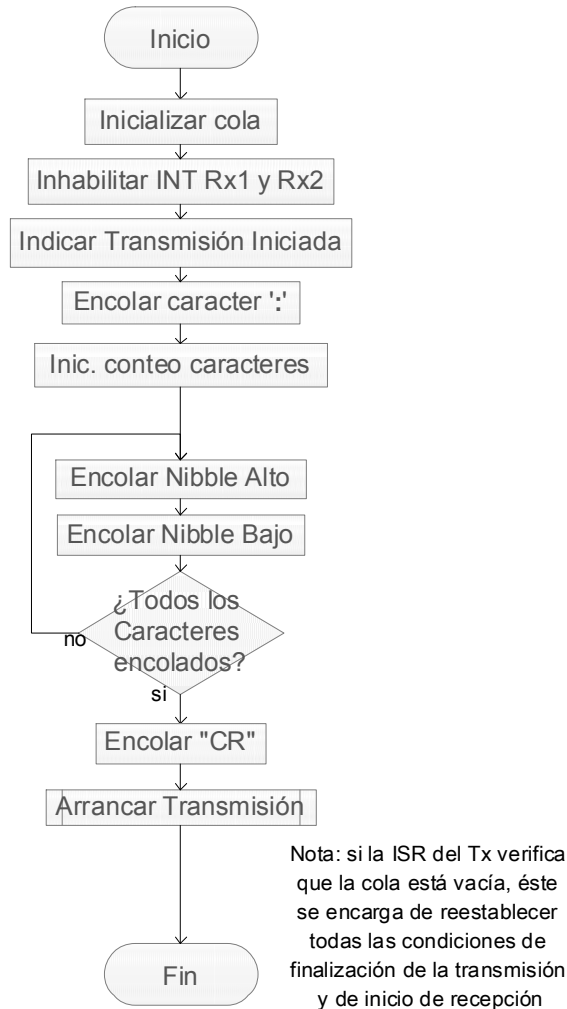


Figura 30. Transmitir Comando.

#### 4.2.1.5. Máquina: Switch Bluetooth

La máquina del Switch Bluetooth es la encargada de encender/apagar el módulo Bluetooth tras la activación del switch externamente. La tarea es activada tras la interrupción externa (INT1) que activa el switch.

Tras el encendido del módulo Bluetooth, también es configurado un temporizador que se encarga de fijar un tiempo, tras el cual se apagará el mismo de forma automática para evitar un posible desgaste innecesario de energía en caso que

se deje encendido inadvertidamente luego de terminar la sesión de comunicación. Si el temporizador ya estaba contando previamente, solo será reiniciado en este punto.

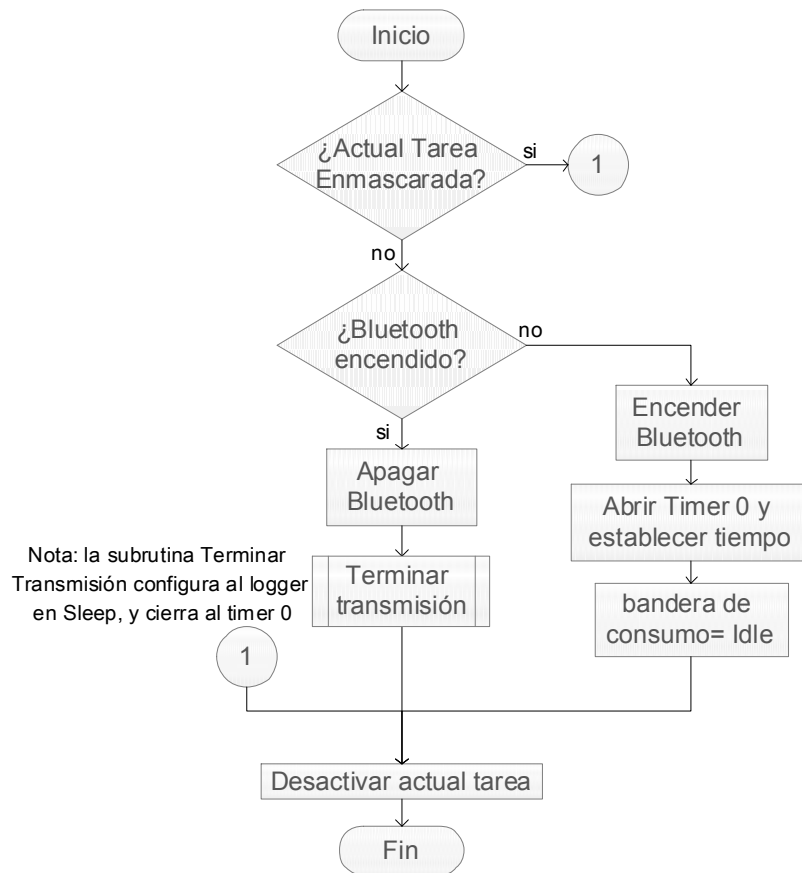


Figura 31. Máquina Switch Bluetooth.

#### 4.2.1.6. Máquina: Iniciar Adquisición

La máquina de iniciar adquisición es la encargada de inicializar todas las variables necesarias para arrancar el proceso que se ejecuta durante la máquina de adquisición temporizada, También configura la primera alarma del RTC, que es la que dará marcha al proceso. La memoria flash es completamente borrada por esta máquina.

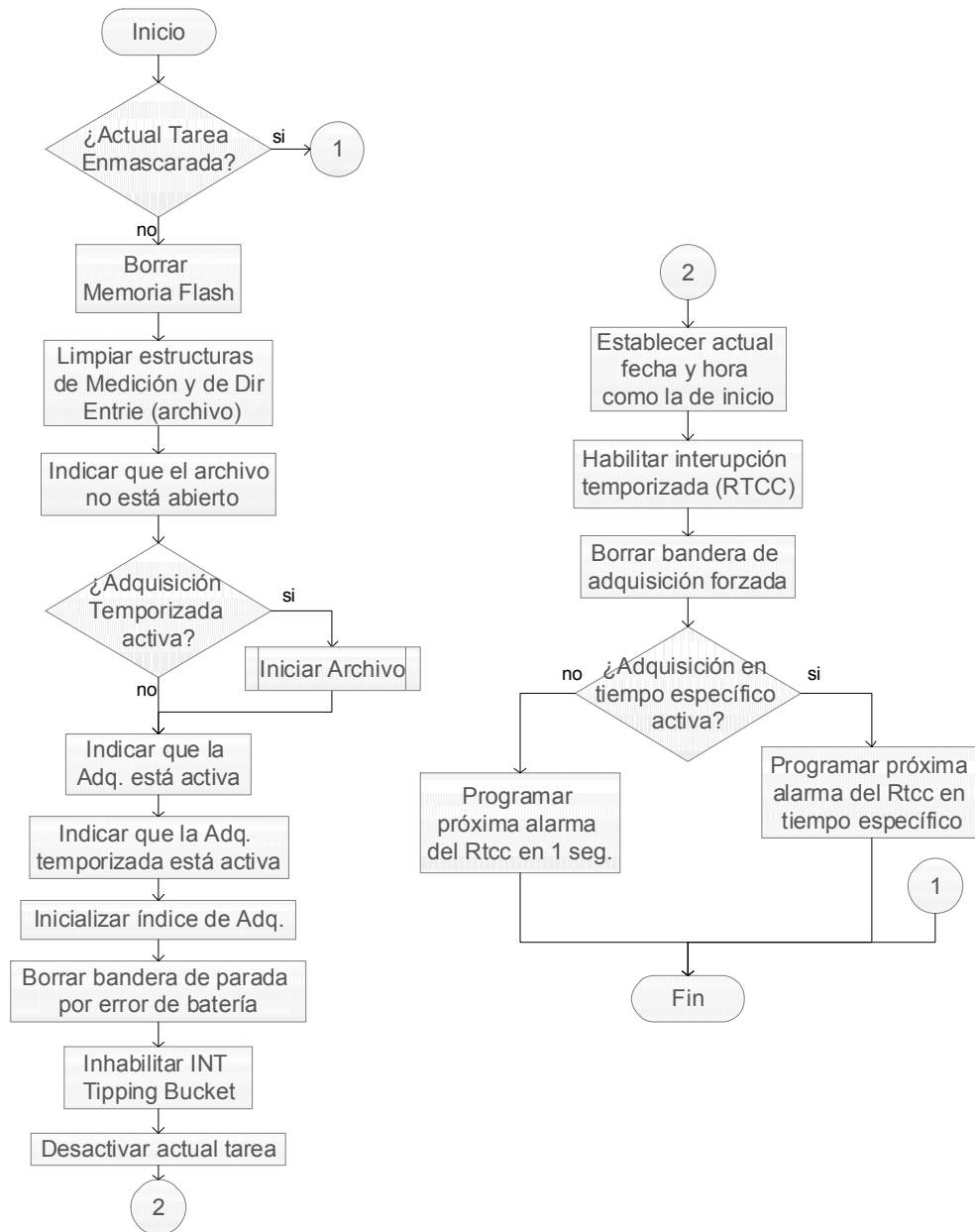


Figura 32. Máquina: Iniciar Adquisición.

#### 4.2.1.7. Máquina: Parar Adquisición

Esta máquina se encarga de inhabilitar la alarma del RTCC, y todas las interrupciones asociadas al proceso de adquisición. La cabecera del archivo de datos contenida en memoria RAM es respaldada en memoria no volátil y el archivo es cerrado.

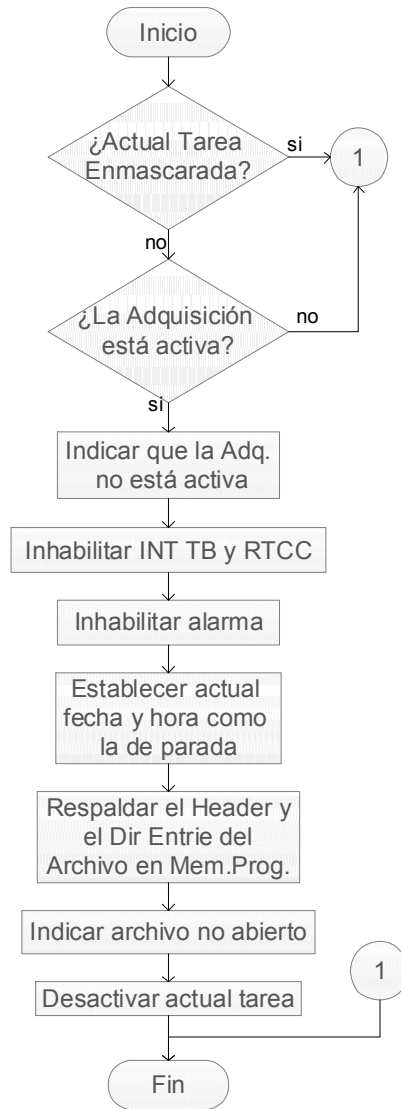


Figura 33. Máquina: Parar Adquisición.

#### 4.2.1.8. Máquina de Configuración del Bluetooth

La máquina de Configuración del Bluetooth es, junto a la Máquina de Lectura de Caracteres Bluetooth (detallada más adelante), la encargada de la inicialización y la configuración de los parámetros más importantes del módulo Bluetooth, esenciales para la correcta comunicación del equipo con un dispositivo externo que cuente con la tecnología inalámbrica. Los parámetros a ser configurados son el nombre, la contraseña, la tasa de baudios, y el tiempo que demora el módulo Bluetooth en

apagarse (de forma automática) desde que es encendido mediante la activación del switch. Los tres primeros parámetros son propios del módulo Bluetooth, mientras que el último (el tiempo) depende única y exclusivamente del equipo de adquisición de datos.

Cada estado de la máquina de configuración del Bluetooth se encarga del envío de los caracteres correspondientes al módulo, tanto para entrar en el modo de comandos (en la sección 4.2.2.7 se puede profundizar un poco más respecto al modo de comandos del Bluetooth), como para la configuración de cada parámetro que se requiera específicamente.

La variable que controla los estados en esta máquina es, a diferencia del resto de las máquinas de estado, de carácter global, y por lo tanto, puede cambiarse desde afuera de dicha rutina. La máquina de configuración del Bluetooth contiene cada uno de los pasos de la configuración, pero el orden en que éstos se ejecutan es, en gran parte, controlado desde la otra máquina: Lectura de Caracteres Bluetooth.

La máquina de configuración del Bluetooth envía caracteres al módulo, y luego es enmascarada, para de esta forma, permitir al microcontrolador entrar en un estado de bajo consumo, mientras espera a que el módulo Bluetooth le responda, proceso en el cual la máquina de lectura de caracteres bluetooth se encargará de interpretar adecuadamente dicha respuesta, y desenmascarar nuevamente la tarea, así como decidir a qué nuevo estado de la máquina de configuración entrar.

Existe una bandera especial que ha sido creada para indicarle al microcontrolador cómo debe interpretar los caracteres que reciba, por lo cual existen 2 escenarios: cuando el logger se comunica con un dispositivo externo mediante el protocolo TECNUM, y cuando el logger se encuentra configurando al módulo Bluetooth. Esta bandera es activada al inicio de la máquina de configuración y desactivada bajo cualquier condición que exige que dicha tarea de configuración sea cerrada (ya sea porque finalizó el proceso de configuración, o porque ocurrió un error debido a que la respuesta no fue recibida correctamente, o jamás ocurrió).

Dependiendo de cómo se encuentre esta bandera, la interrupción serial (recepción de un carácter) activará la tarea de lectura del carácter bluetooth, o la tarea de comunicación USART (encargada de interpretar y ejecutar los comandos bajo el protocolo TECNUM).

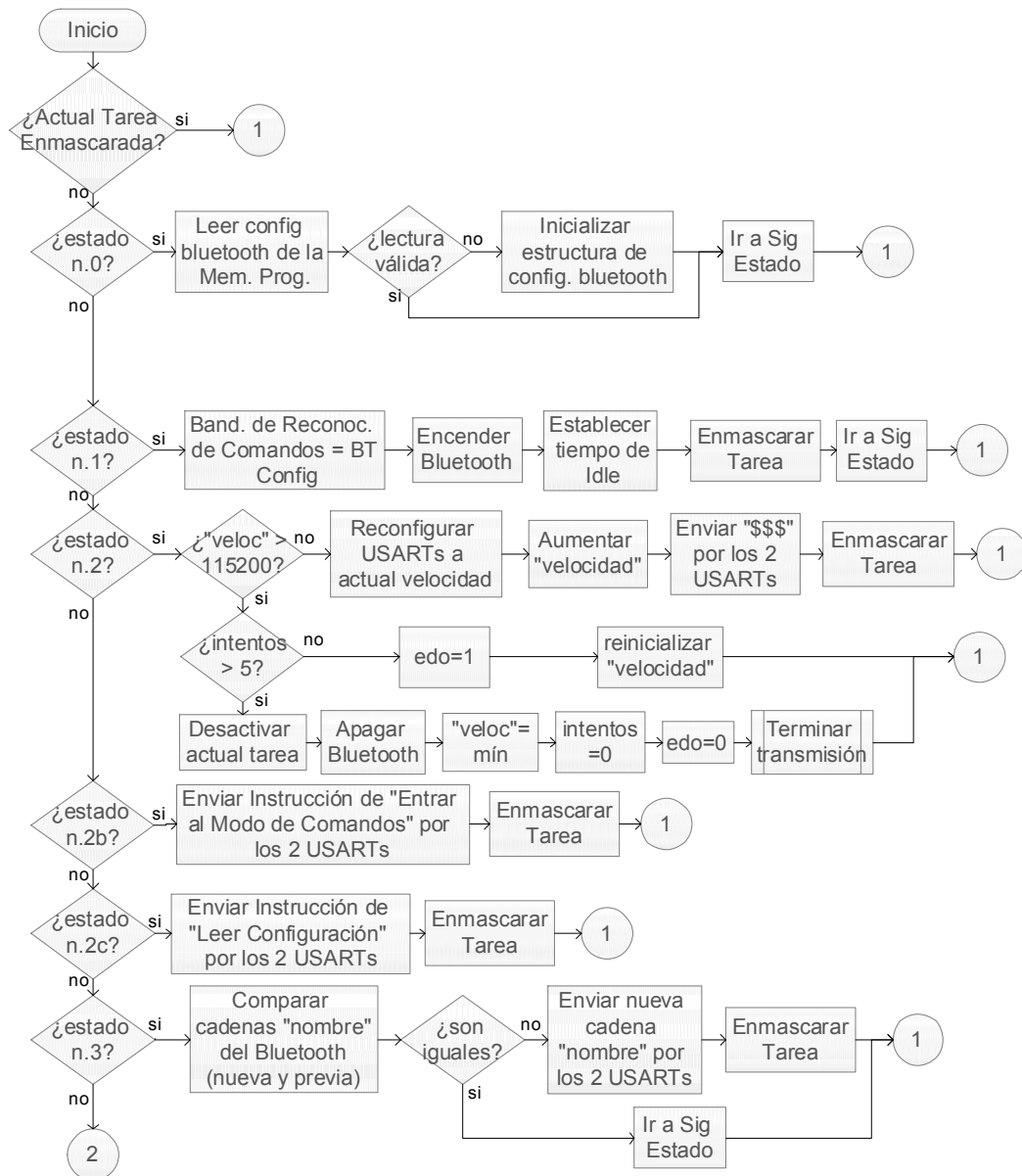


Figura 34. Máquina: Configurar Bluetooth (parte 1).

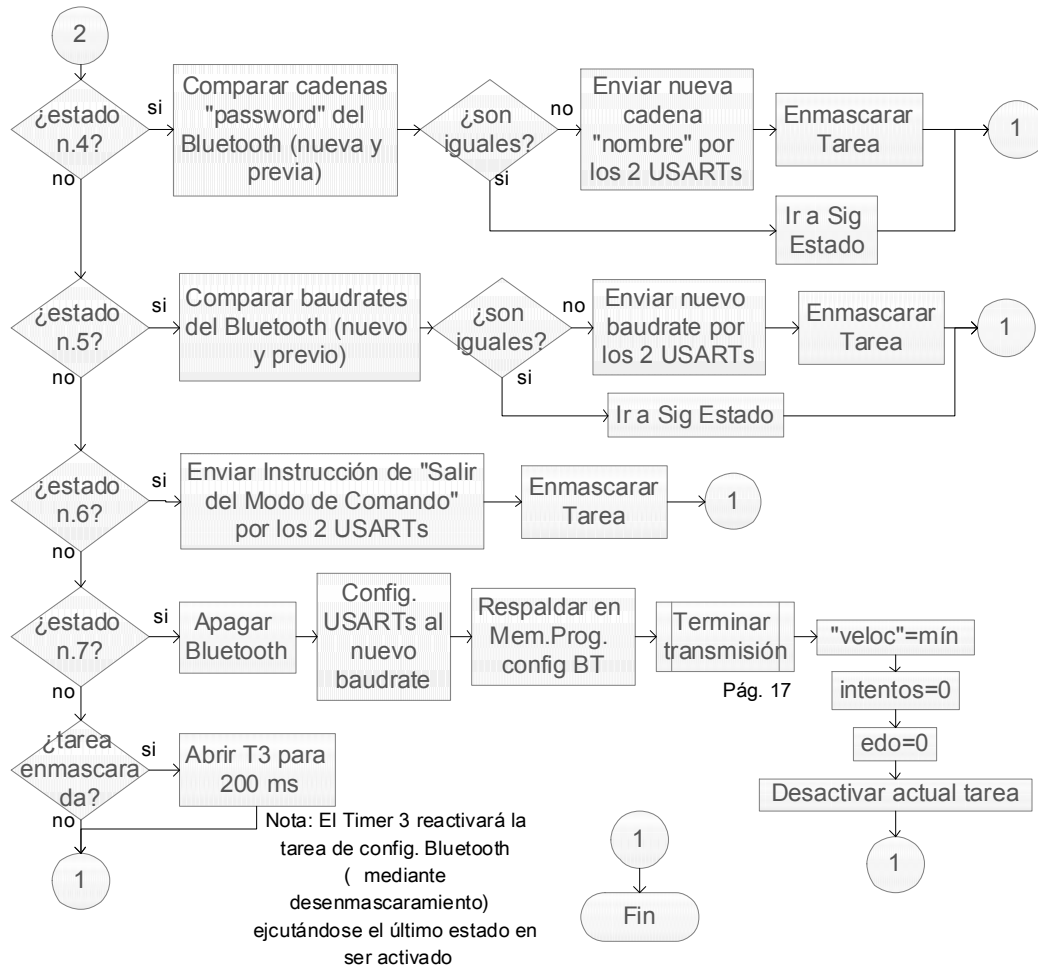


Figura 35. Máquina: Configurar Bluetooth (parte 2).

La tarea de configuración de Bluetooth siempre es activada desde el momento en que es iniciado un logger. La rutina de configuración leerá los parámetros previos del dispositivo Bluetooth, y sólo cambiará la configuración de los que sean necesarios. Si el valor que se le desea asignar a un parámetro particular es idéntico al valor que poseía previamente, entonces la máquina se saltará el paso de la configuración de éste parámetro.

La tarea de configuración del Bluetooth también puede ser activada mediante la recepción del comando de fijar nueva configuración Bluetooth (protocolo TECNUM).



### 4.2.1.9. Máquina de Lectura de Caracteres Bluetooth

Como se mencionó previamente, la máquina de lectura Bluetooth se encarga, junto a la máquina de configuración, del proceso de reconfigurar al módulo Bluetooth. La máquina de lectura de caracteres es activada durante cada recepción de caracteres por el puerto USART 1 (que es el que está físicamente conectado con el módulo Bluetooth).

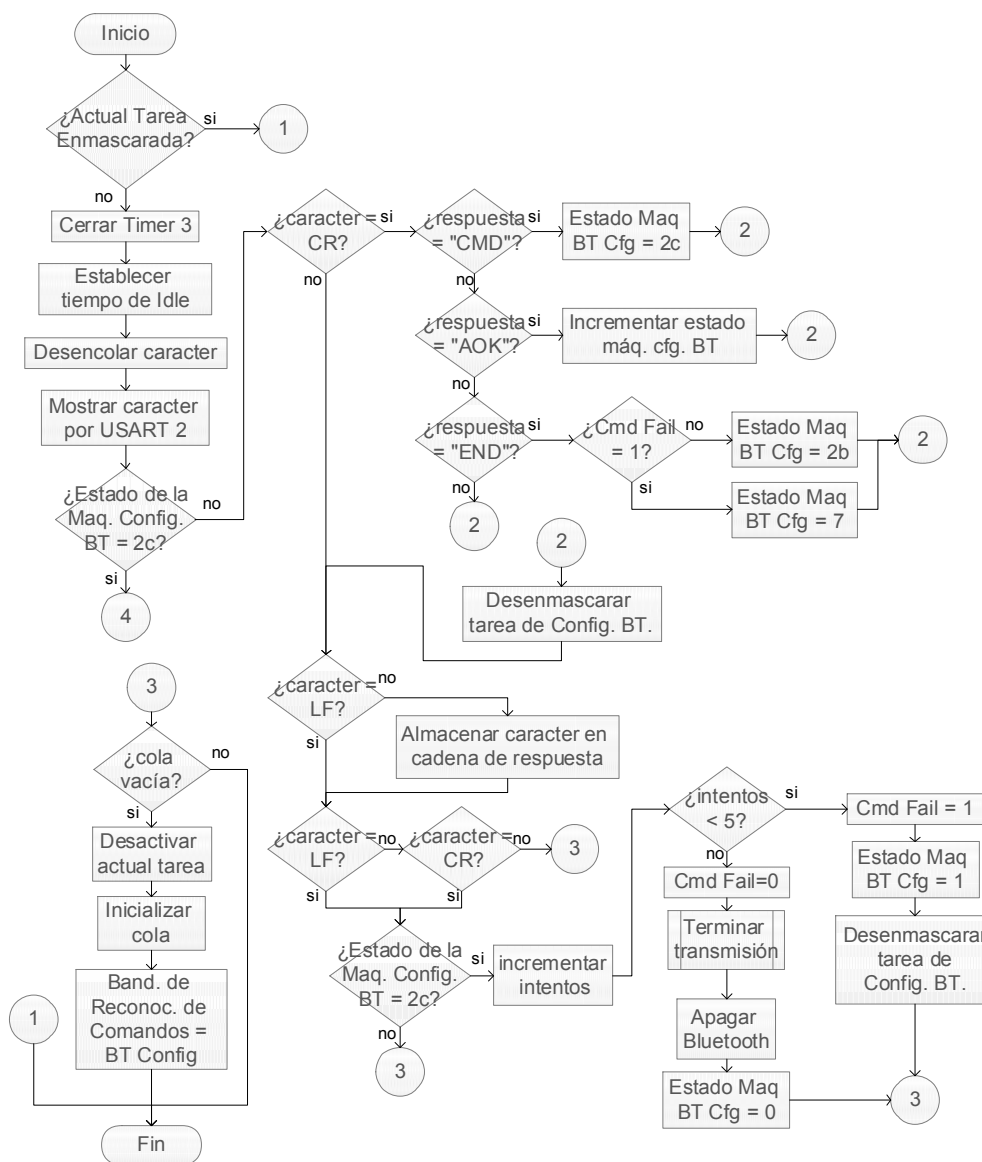


Figura 36. Máquina: Leer Caracteres Bluetooth (parte 1).

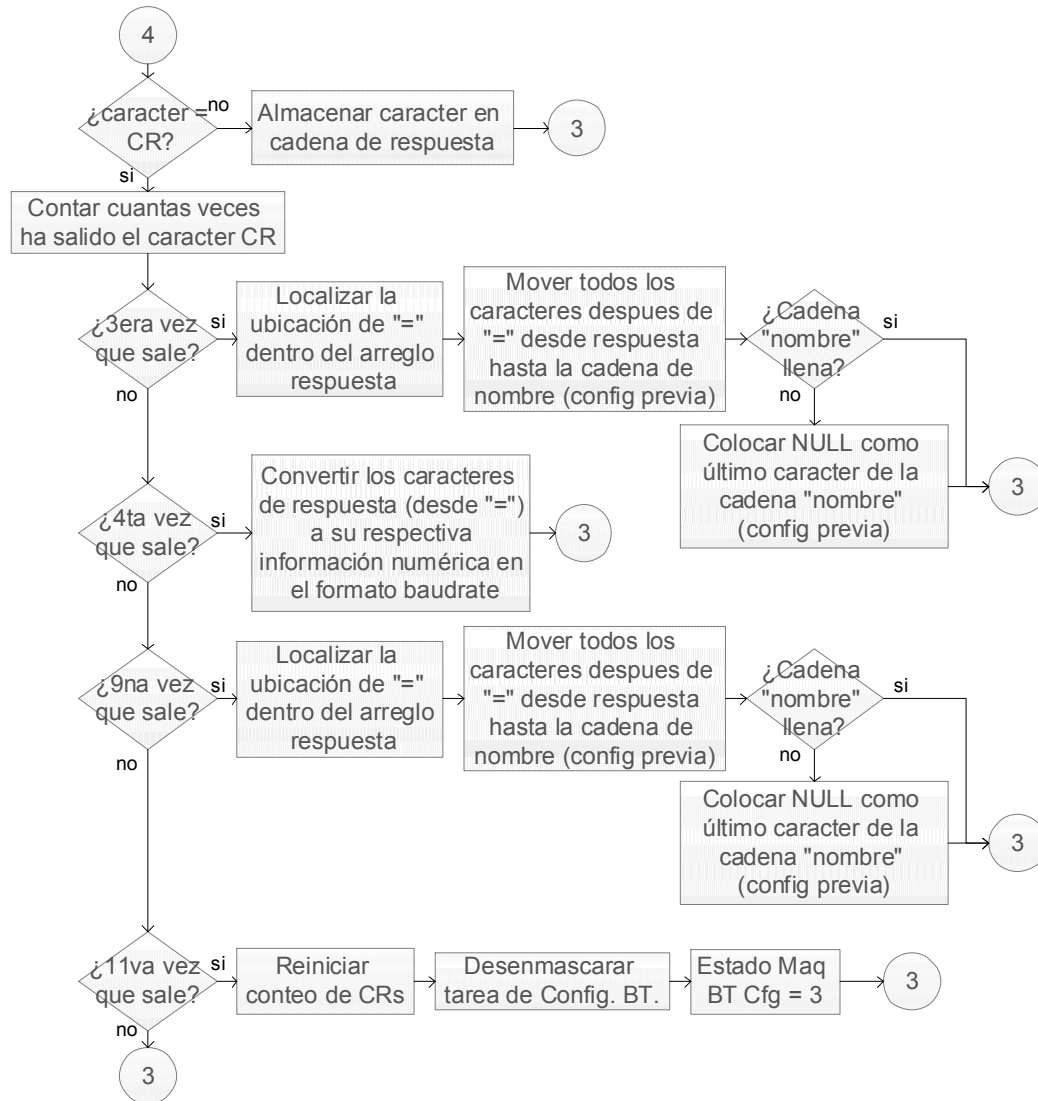


Figura 37. Máquina: Leer Caracteres Bluetooth (parte 2).

## 4.2.2. Otros aspectos

### 4.2.2.1. La secuencia de Adquisición del Logger

Para poder activar la secuencia de adquisición, es necesario que el usuario, con la ayuda del software de configuración desde un dispositivo externo en comunicación con el logger, fije la configuración deseada. Entre las opciones de configuración se encuentran la cantidad de canales a habilitar, el tiempo de intervalo de adquisición,

tiempo de establecimiento y algunas banderas de configuración. Una vez fijada la configuración deseada, y sincronizado el reloj, se envía el comando de *Iniciar Adquisición* para que el logger lleve a cabo dicha tarea.

La tarea *Iniciar Adquisición* no es iniciada hasta que finaliza el proceso de transmisión de la trama de respuesta (máquina de comunicación serial), y en ésta el logger inicializa banderas y programa la alarma del RTCC para que se ejecute en el menor tiempo posible después del momento presente (el cual es 1 segundo). En ese momento finaliza la tarea de iniciar adquisición.

La interrupción producida por la alarma del RTCC activa la tarea de adquisición, la cual llamará a la máquina respectiva. Esta máquina habilitará en una primera ocasión la interrupción de eventos externos (conectado con el pluviómetro tipo balancín o Tipping Bucket), y programa la alarma del RTCC para que se active nuevamente en un tiempo correspondiente al intervalo configurado, contado a partir del instante actual. Previamente el programa debe haber realizado el proceso de medir la tensión de la batería y comprobar que no ha resultado baja por 5 veces seguidas para parar el proceso. Luego ejecuta el resto del procedimiento propio de la adquisición, el cual consiste en los siguientes pasos:

- Encender la tensión que alimenta a los sensores y a la circuitería auxiliar.
- Esperar por el tiempo de establecimiento previamente configurado.
- Comprobar los canales y realizar la lectura en aquellos que se encuentren activos.
- Construir el bloque de datos a ser almacenados.
- Iniciar archivo de datos (solo en una primera ocasión).
- Transferir bloque de datos al archivo (ubicado en memoria flash externa).

Una vez realizado este proceso, la tarea de *Adquisición Temporizada* es desactivada y el logger regresa a su estado de bajo consumo hasta el momento en que ocurre la próxima interrupción del RTC. El proceso de adquisición finaliza cuando ocurre uno de los siguientes escenarios:

- a) El usuario envió el comando *Parar Adquisición* desde un dispositivo externo.
- b) Durante la comprobación de batería que ocurre al inicio del proceso, se detectó nivel deficiente por quinta vez consecutiva.
- c) Durante el proceso de transferencia del bloque de datos a la memoria, sólo había espacio para un bloque más (la memoria flash externa se llenó).

Durante cualquiera de estos escenarios, es activada la tarea de parar adquisición, la cual cierra el archivo de datos, e inhabilita la alarma del RTCC para evitar que siga ocurriendo el proceso de adquisición, así como es inhabilitada la interrupción externa relacionada con el Tipping Bucket.

#### **4.2.2.2. El Pluviómetro tipo Balancín ó Tipping Bucket**

El pluviómetro tipo balancín va conectado a la entrada del logger cuya interrupción se considera la de más alta prioridad, debido a la naturaleza de los eventos de pulsación que ocurren durante un breve instante y el microcontrolador no debe estar ocupado por otra tarea de igual o mayor prioridad, o dicho evento pudiera no ser atendido y se perdería, cuestión inaceptable para una estación meteorológica.

Las pulsaciones del Tipping Bucket sólo son atendidas cuando el logger se encuentra en modo de adquisición, y cuando ocurren, éste incrementa un contador de pulsos (puesto en cero al Iniciar Adquisición). El logger cuenta la cantidad de pulsos que ocurren entre una alarma del RTCC y la siguiente, y durante el proceso de construcción del bloque de datos en memoria RAM, el valor del contador de pulsos es almacenado en dicho bloque y reinicializado a cero. Además de ello, el logger cuenta con otra variable donde almacena el acumulado de todos estos conteos de pulsaciones para poder mostrarlo durante la supervisión, la cual es una modalidad del software de configuración en computadoras y dispositivos externos.

### 4.2.2.3. El Archivo de Datos

La información que forma parte del archivo de datos que recoge las mediciones realizadas está compuesta de un cierto número de elementos de tal forma que a partir de éstos se puede construir un archivo completo y legible por una computadora. Estos elementos pueden ser divididos en 3 categorías:

- a) Entrada de directorio: necesaria para construir un archivo con cualquiera de los formatos que manejan diversos sistemas operativos (FAT, NTFS, exFAT, etc.)
- b) Cabecera: contiene información necesaria para la correcta interpretación de los datos de las mediciones, entre otra información relevante al archivo (identificación, etc.)
- c) Los datos de las mediciones en sí.

Durante la etapa de la creación del archivo, el programa inicializa una entrada de directorio y una cabecera, las que respalda en memoria no volátil. Luego, cada vez que se escriben datos concernientes a las mediciones, son actualizadas las secciones de la entrada de directorio y de la cabecera que corresponden.

El antecesor del Nanologger, el  $\mu$ Logger 3.x, disponía de una memoria extraíble del tipo *Secure Digital* para el almacenamiento de los datos, lo cual justificaba que los datos fueran escritos en un sistema de archivos compatible con el sistema FAT16 y así lograr su lectura en equipos de computación dotados con esta interfaz. Sin embargo, en el Nanologger la memoria es fija, por lo que se ha planificado un orden de almacenamiento distinto y más sencillo, pero compatible en varios aspectos con la estructura del  $\mu$ Logger 3.x.

Debido a las limitaciones que presenta la memoria serial en cuanto a capacidad y en cuanto al tamaño mínimo del sector borrable (4 KB), se ha decidido almacenar la cabecera y la entrada de directorio fuera de ésta, dentro de la memoria flash interna del microcontrolador (memoria de programa), ya que estas dos estructuras, que requieren ser actualizadas durante cada adquisición, necesitan ser almacenadas en

sectores independientes de los datos. Si no fuese así, habría que borrar parte de los datos al tener que borrar dichas estructuras para poderlas actualizar, o reservar un sector de 4 KB únicamente para estas estructuras que ocupan menos de la mitad de ese espacio.

Debido a las limitaciones que presentan las memorias no volátiles con las que cuenta el Nanologger, por ejemplo su limitada durabilidad (sólo 100.000 ciclos de borrado/escritura, tanto para la flash interna como para la externa), el firmware se ha diseñado para que la cabecera y la entrada de directorio no sean respaldadas cada vez que necesiten ser actualizadas, sino cada N cantidad de veces, donde la N es un número que depende del intervalo de adquisición. Intervalos muy frecuentes harían que N tenga que ser muy grande, e intervalos de baja frecuencia podrían hacer que N sea un número bajo ó cero.

Aún no se ha integrado ningún método que le permita al equipo comprobar la validez o integridad de sus datos almacenados, pero se ha pensado en ello como una posibilidad para futuras versiones.

#### **4.2.2.4. Las Adquisiciones Forzadas**

Paralelamente al proceso de adquisición, existe otro proceso que también puede ocupar la máquina de adquisición, y es el proceso de las adquisiciones forzadas. Ocurren cuando el logger recibe desde los puertos de comunicación serial (USART) el comando *Actualizar Mediciones*, que activa la tarea de adquisición, y adicionalmente, una bandera de adquisición forzada.

La máquina de adquisición, durante su fase de comprobación de batería, verifica si la adquisición fue producida por interrupción RTCC y también si fue forzada (producida por el comando), en cuyo caso indica mediante otra bandera que no hay que salvar la adquisición, lo que hará que se salte las etapas de construir bloque de datos y almacenar en memoria.

El comando *Actualizar Mediciones* únicamente tiene el propósito de mandar al logger a realizar una adquisición de todos los canales que sean indicados, y actualizar en RAM dicho contenido, para luego, con el comando complementario *Leer Mediciones* solicitar dichos datos por puerto serial. Todo esto se hace mediante la opción de Supervisión desde el software de dispositivos externos (PC o móviles).

#### **4.2.2.5. El software de dispositivos externos**

El software existente para configurar al Nanologger, así como a otros equipos desarrollados por TECNUM es el PCLogger.NET v1.11.1124.1. Dicho software fue desarrollado exclusivamente para el  $\mu$ Logger 3.x, pero el Nanologger ha sido desarrollado con compatibilidad para este software. Los futuros software de configuración se desean con las mismas funcionalidades que tiene el PCLogger.NET, las cuales se listan a continuación:

1. Sincronización y lectura de fecha y hora: el software permite pasar los valores actuales de fecha y hora que tiene la computadora por puerto serial a la unidad esclava, así solicitarla.
2. Escritura y lectura de la Identificación: permite la lectura y modificación de los datos distintivos de la estación.
3. Configuración: el software permite ver y modificar todos los datos que condicionan el proceso de adquisición, así como habilitar e inhabilitar canales, configurar los polinomios de los mismos, entre otros datos y banderas.
4. Supervisión: permite monitorear los resultados de adquisiciones al logger sin necesidad de mandarlo a adquirir, o también permite monitorearlo en estado de adquisición, pudiendo graficar las lecturas realizadas de los canales que estén habilitados.
5. Descargar Datos: permite leer los archivos de datos que se encuentren disponibles en la unidad esclava. El archivo una vez leído es construido en formato binario.
6. Convertir los Datos: permite la conversión del archivo del formato binario a uno que pueda ser visualizado por el usuario desde la computadora.

7. Otras funciones: permite activar o desactivar las salidas digitales y/o la salida de alimentación de sensores, reiniciar la unidad, reinicializar los contadores, leer sectores físicos de la memoria, enviar y recibir comandos SDI-12 a sensores conectados a ese bus y configurar una unidad a partir de la información almacenada en un archivo binario de adquisición. Muchos de estos comandos son incompatibles con el Nanologger.

#### **4.2.2.6. Manejo del Consumo**

El microcontrolador PIC18F46J50, que es la base del equipo Nanologger, tiene 4 estados diferentes de consumo: el estado activo, el de mayor consumo; el estado de espera, con consumo moderado y que depende de qué periféricos estén activos y cuáles no; el estado de sueño, de bajo consumo; y el de sueño profundo, con el mínimo consumo y el menor uso de periféricos.

El estado de sueño profundo jamás es utilizado por el logger, debido a que son pocos los periféricos que son capaces de sacarlo de este estado, y cuando retorna, lo hace reiniciándose y reiniciando su memoria RAM. Por esta razón el estado de sueño profundo no es utilizado en esta aplicación. Sin embargo se aprovecharon los 3 estados restantes.

El estado de bajo consumo más utilizado por el logger es el de sueño, el cual apaga el oscilador principal y todos sus periféricos salvo el RTCC y el Watchdog. El logger entra en modo sueño cada vez que termina de realizar todas las tareas pendientes y el administrador de tareas libera al CPU mediante la activación del modo sueño. Una vez en estado de sueño, los periféricos que son capaces de sacarlo de este estado son:

1. Cualquiera de las interrupciones externas,
2. La interrupción del RTCC,
3. La interrupción de la recepción del USART configurado en modo *auto wake-up*.



Los temporizadores se encuentran entre los periféricos que no pueden sacar al controlador de su estado de sueño, dado que en este modo el oscilador se encuentra apagado.

Cuando la bandera de configuración del USART *Auto Wake-Up* es activada, éste es automáticamente apagado, por lo que resulta necesario activarla sólo antes del momento de entrar en sueño. Cuando el Microcontrolador recibe un carácter, el USART necesita de un tiempo para encenderse y entrar en completa funcionalidad, en cuyo tiempo, ese carácter será perdido irremediablemente dado que el oscilador toma un tiempo en reactivarse luego de la interrupción. Por esta razón resulta inviable que el logger entre en estado sueño entre los caracteres sucesivos. Debido a esta limitación de este modo, con cada recepción de caracteres el microcontrolador es configurado para que la próxima vez que llegue a la rutina de entrar en estado de bajo consumo, no vaya al modo sueño, sino al modo de energía más alto que le sigue: el de espera ó idle, el cual mantiene encendido el oscilador.

Cuando el logger recibe un carácter es configurado para el modo Idle, y además de eso, se programa un temporizador que contará el tiempo de inactividad de los puertos USART. Si un nuevo carácter es recibido, entonces el conteo del temporizador es reiniciado. Si el temporizador finaliza el conteo que se le configuró, quiere decir que el plazo de espera de caracteres fue rebosado y éste debe regresar al modo de sueño para ahorrar energía.

Otra condición para llevar al logger al modo de espera (Idle) es cuando el usuario enciende el módulo Bluetooth a través de un switch especialmente incorporado para este propósito. Cuando esto sucede, se desea que el Bluetooth se apague de forma automática al cabo de cierto tiempo de inactividad, por lo que se programa un temporizador para que cumpla con dicha función. Como el módulo Bluetooth también hace uso de un puerto USART para comunicarse con el Microcontrolador, entonces el temporizador utilizado es el mismo que cuenta la

inactividad de los USART, y el Bluetooth se apaga si ningún carácter es recibido durante el tiempo programado. Este tiempo depende de la configuración del Bluetooth, y es elegible por el usuario, pudiendo variar entre 26 y 281 segundos ( $26+255=281$ , donde 255 es el tamaño del byte). El tiempo mínimo fue estimado como aquel en cuyo tiempo se puede establecer un emparejamiento exitoso con el módulo.

El tiempo que se le programa al temporizador tras la recepción de un carácter (indistintamente del puerto USART por donde lo reciba) dependerá de si el módulo Bluetooth está encendido o no. En caso de estarlo, el tiempo será el configurado según se explicó anteriormente (es el tiempo largo) y si el módulo Bluetooth está apagado entonces no hay necesidad de un tiempo de espera largo ya que se asume que la comunicación se está llevando a cabo directamente por el puerto RS232 y se le programa a 2 segundos para el apagado.

#### **4.2.2.7. Módulo Bluetooth incorporado a la tarjeta**

El módulo Bluetooth que ha sido seleccionado e incorporado al NanoLogger es el RN-42 de la casa Roving Networks, como se vio en el Capítulo 4, sección 4.1.4. Este módulo posee la versión de Bluetooth 2.1, y pertenece a la clase 2, lo que significa que tiene un rango de alcance máximo de 10m.

Tal y como lo establece el fabricante a través del *Command Set* del RN-42, una vez energizado, el módulo Bluetooth dispone de un lapso de tiempo en el que es capaz de entrar al modo de configuración, mediante el envío al mismo de la secuencia de caracteres \$\$\$\$. En el modo de comando se puede reconfigurar el nombre, password, parámetros propios de la comunicación UART (velocidad de transferencia de datos, paridad, número de bits), entre muchos otros parámetros de configuración. Una vez transcurrido el lapso de tiempo desde que el Bluetooth es energizado, si el módulo no ha recibido el comando de entrada al modo de configuración, éste ignorará dicho comando (“\$\$\$”, en formato ASCII).

La máquina de configuración del Bluetooth (parte del firmware del NanoLogger), es la encargada de manejar la parte de la reconfiguración del módulo a través del envío de los comandos correspondientes para el establecimiento de sus parámetros, tales como la velocidad, el nombre y el password. El módulo es inicializado sólo una vez durante toda la vida del equipo, y cada vez que el usuario elige cambiar la configuración, dicha máquina es activada.

### **4.3. Pruebas Realizadas y Resultados**

Durante una fase inicial del desarrollo, se experimentó con el PIC18F46J50 para lograr la funcionalidad de cada una de sus características más relevantes por separado. Entre estas características se cuentan el proceso de conversión analógica-digital, el reloj de tiempo real, la comunicación serial, las interrupciones, el proceso para entrar en modo de bajo consumo, los puertos de entrada-salida, etc.

Luego se incorporaron todas sus características en un programa inicial de prueba que realizaba las tareas más elementales de un Registrador de Datos, incluida la comunicación serial con la computadora. El correcto funcionamiento del prototipo inicial fue comprobado mediante software de simulación de circuitos electrónicos.

Más adelante, en la necesidad del desarrollo de compatibilidad de comunicación entre el equipo real y las tecnologías existentes, se inició una fase de estudio del firmware del equipo previo de TECNUM, el  $\mu$ Logger 3.2, para incorporar las rutinas necesarias para lograr la compatibilidad.

#### **4.3.1. Pruebas de comunicación**

La comunicación se logró positivamente a 9600 baudios durante la etapa inicial del trabajo, y así se mantuvo durante la mayor parte del desarrollo. Luego se realizaron pruebas a mayores velocidades. La utilización del *Opto acoplador* MCT6 como conversor a RS-232 originó problemas en las comunicaciones a velocidades mayores de 19.200 baudios, pero su reemplazo por el circuito integrado MAX3233

corrigió e hizo posible la comunicación a velocidades grandes, incluyendo 115.200 baudios.

#### **4.3.2. Pruebas de Adquisición de Datos**

Las rutinas desarrolladas para el prototipo inicial del DataLogger resultaron positivas, demostrando que eran capaces de convertir la tensión leída por el canal analógico a su valor equivalente en hexadecimal, donde el valor de referencia de 3.3V era traducido a 3Fh, número correspondiente al máximo en 10 bits (11 1111 1111b), o en decimal 1023.

Al incorporarse las rutinas de conversión al firmware definitivo, se obtuvieron las mismas tendencias en los resultados de los canales adquiridos, y gracias a estructuras especiales definidas en dicho firmware, las conversiones son traducidas, no desde el firmware, sino desde software externo, a valores interpretables por el ser humano, tales como expresiones en voltaje, temperatura, entre otros.

#### **4.3.3. Pruebas de Registro**

Como ya se ha visto anteriormente, el registro es impulsado gracias a las interrupciones que ocurren de forma temporizada, mediante el RTCC. Se ha observado que a lo largo de las pruebas realizadas al logger, el proceso periódico de registro, así como el posterior almacenamiento de los datos en la memoria Flash Serial, se han dado sin problemas.

Eventualmente se han presentado errores de funcionamiento en torno a aspectos que si bien se encuentran relacionados con el registro, no han sido debido al registro en sí, tales como errores durante el proceso de escritura de datos en la memoria, o inadecuado manejo de consumo cuando sucede una operación en medio del registro. Cada error que ha sido detectado se ha estudiado a fondo y corregido.

#### 4.3.4. Pruebas de Consumo

En un principio, se deseaba aprovechar la característica del PIC18F46J50 de consumo al mínimo, mediante su estado de “Sueño Profundo” (Deep Sleep), por lo que se realizaron pruebas en las que se deseaba saber cuáles eran las condiciones para entrar y salir de dicho estado. Se comprobó que es complicado, puesto que son muy limitados los periféricos que son capaces de sacar al microcontrolador de dicho estado, y además, toda la información contenida en la memoria RAM se pierde puesto que el microcontrolador sufre de un reset al salir del Sueño Profundo, y que hay que usar las banderas de un registro especial para comprobar cual periférico lo reactivó.

El fabricante especifica que la corriente de consumo del microcontrolador es cerca de 13 nA en estado de Sueño Profundo, pero aunque se encuentre en dicho estado, el tener activo el RTCC agrega unos 850 nA al consumo, y en estado de Sueño, el cual se comprobó tiene más versatilidad, apenas consume 1µA, por lo que, el ahorro no se considera tan grande a cambio de tener un equipo que cambia de estados con menos pérdida de datos.

Ciertos cambios y correcciones tuvieron que hacerse al firmware y al hardware para poder lograr la máxima reducción de consumo, entre los cuales se pueden mencionar:

- Borrado de la bandera de tensión de banda de referencia (*1.2V Band Gap Reference Enable Bit*) dentro de un registro de control del convertidor analógico digital, con el fin de que éste no mantenga dicha tensión encendida siempre, sino solo al momento de la adquisición. Esta bandera se encarga de activar un circuito que crea una tensión de ~1.2V, sobre el cual es posible realizar lecturas y conversiones con el módulo convertidor A/D. También puede ser activado por el módulo comparador, el regulador de tensión del circuito integrado (*On-Chip Voltage Regulator*) y el módulo HLVD (*High-Low Voltage Detect*).

- Establecimiento de la bandera del regulador de voltaje del circuito integrado (*On-Chip Voltage Regulator*), para que éste entre en bajo consumo al momento en que el microcontrolador entra en modo Sueño. El regulador de voltaje permite al circuito integrado del microcontrolador operar internamente bajo una tensión regulada de 2,5V para reducir consumo, y utiliza el voltaje suministrado por el pin de Vdd (+3,3V) para el nivel de las entradas/salidas.
- La Memoria Flash Serial permanece sin alimentación de forma regular, sólo se le enciende al momento de adquirir y salvar los datos.
- Un condensador, con las especificaciones que suministra el fabricante, conectado en el terminal de *Vddcore*, permite la reducción del consumo del equipo.

La siguiente tabla muestra un aproximado de las corrientes de consumo del NanoLogger en cada uno de los distintos estados de operación:

*Tabla 5. Consumo del NanoLogger.*

<b>Estado</b>	<b>Corriente de Consumo</b>
Sueño	6.5 $\mu$ A
Sueño con el driver RS232 en Espera	2.58 mA
Estado de Adquisición/Comunicación	8 mA
Activo/Descargando Datos	10 mA
Bluetooth encendido/no conectado	12 mA
Bluetooth encendido/conectado	40 mA

#### **4.3.5. Pruebas de Compatibilidad**

A pesar de que la comunicación fue una de las primeras tareas en lograrse, no fue así respecto a la compatibilidad. En un principio se estudió cómo funcionaba el protocolo TECNUM, para escribir desde cero el código para que el equipo lo manejara y fuese capaz de elaborar comandos a transmitir y reconocer los recibidos,

pero la documentación era muy limitada y había muchos detalles respecto al protocolo que no eran cubiertos.

La plena compatibilidad se logró exportando todas las rutinas necesarias al firmware que provenían del anterior logger, en donde se sabe con exactitud qué hace cada comando, qué condiciones de entrada requiere y qué envía a la salida, así como las tareas pertinentes a ejecutar en cada caso.

#### **4.3.6. Pruebas de Comunicación Inalámbrica**

Las pruebas de comunicación inalámbricas se realizaron mediante un teléfono inteligente marca LG, modelo Optimus One P500, bajo el sistema operativo Android 2.2, y mediante las aplicaciones *BlueTerm*, y *S2 Terminal for Bluetooth*. La comunicación fue lograda en un principio a 9600 baudios, y luego se le añadió funcionalidad al logger para poder cambiar la configuración del módulo bluetooth, de tal modo de poder enviar y transmitir datos para distintas velocidades a la anterior, logrando comunicación efectiva incluso a 115.200 baudios.

## CONCLUSIONES

Tras todo el trabajo del desarrollo, tanto a nivel de hardware como firmware, del presente proyecto, los resultados que se han obtenido se listan a continuación:

- Se realizó el estudio de las tecnologías disponibles en el mercado, con el fin de orientar correctamente la planificación del equipo. Se encontraron componentes adecuados para el desarrollo.
- Se discutieron todas las alternativas para el diseño de la estructura física del dispositivo.
- Se realizó la codificación de las rutinas necesarias para darle funcionalidad y operatividad al equipo. Así mismo, se realizaron todas las pruebas necesarias durante el proceso, tales como: simulación, depuración, y montaje en protoboard.
- Se logró un producto que cumple con la funcionalidad que caracteriza a un equipo de medición y registros meteorológicos, y se lograron los objetivos adicionalmente planteados de dimensiones reducidas, bajo consumo y con periféricos adaptables a tecnologías de actualidad.
- Se realizaron pruebas de funcionalidad, adquisición de datos, registro temporizado, comunicación serial con una computadora, comunicación inalámbrica con dispositivos móviles, obteniendo resultados satisfactorios.
- La fase de construcción del prototipo físico (PCB), usable en campo, no estaba concluida para el momento de la elaboración del documento, por lo que los resultados mostrados corresponden al montaje en protoboard.



## RECOMENDACIONES

De entre las posibles mejoras que se le pudiesen realizar al equipo, se ha estimado de gran importancia el uso del módulo HLVD (*High-Low Voltage Detect*) dentro del microcontrolador, el cual se prevé es capaz de generar una interrupción tras la detección de un valor muy bajo de tensión en la alimentación (configurable desde el firmware), de modo tal que en dicha interrupción se ejecute una rutina que, en caso de encontrarse el equipo adquiriendo datos, se realice un respaldo del archivo, adicionalmente a todos los métodos actuales de respaldo por cada número de adquisiciones, o durante consultas al equipo, para el refuerzo de la integridad de la medición.

También se recomienda implementar alternativos formatos de la adquisición, por ejemplo, aquellos donde se registra durante la ocurrencia de un evento asincrónico, como lo es la pulsación del Pluviómetro tipo balancín. . De esa forma se ahorraría espacio en memoria, ya que no se estaría realizando escrituras en los casos en los que existen grandes períodos de sequía. Actualmente el método implementado de registro es en forma periódica.

Es importante realizar una versión ampliada del algoritmo de creación y cierre de los archivos de datos, puesto que, a diferencia del registrador que le antecede al Nanologger, éste sólo puede tener un archivo de datos a la vez, lo que significa que en el caso de ocurrir un evento de error de batería, la adquisición es detenida automáticamente, y el equipo no hará nada hasta que se visitado y reconfigurado por un operador. Para la versión actual de éste equipo, el tener un archivo a la vez no significa mayor problema de funcionalidad, puesto que se ha diseñado para que la energía suministrada por las baterías le brinden un tiempo de vida alto, y además, la rutina que verifica la ocurrencia del error de batería está reforzada, y requiere de al menos 5 lecturas consecutivas de batería baja para poder reconocer el error.

La actual versión del equipo ha sido diseñada para que haga la menor cantidad de borrados y escritura en las memorias no volátiles disponibles (memoria Flash interna del microcontrolador y memoria Flash externa, conectada en forma serial con el controlador), con el fin de garantizar su durabilidad y correcto funcionamiento durante toda su vida útil. Sin embargo, futuras versiones necesitarían integrar adicionalmente rutinas que comprueben, en cada lectura y extracción del archivo de datos, la validez de los mismos mediante técnicas apropiadas como el Checksum o CRC.

Por último, se recomienda el desarrollo de software para unidades móviles (teléfonos inteligentes) que permita la comunicación con la unidad de adquisición de datos (logger), integrando entre sus funcionalidades la fijación y lectura de la fecha y la hora, la configuración de la adquisición, la identificación del dispositivo, así como también la lectura de los datos, la supervisión y la opción de iniciar/parar adquisición.

## BIBLIOGRAFÍA

Brock, F., Richardson, S. Meteorological Measurement Systems. New York: Oxford University Press, 2001.

Gálvez N., Jaime. Control de un módulo Bluetooth mediante microcontrolador/ Jaime Gálvez Navarro (Tesis).-- Universitat Politècnica de Catalunya, 2005.

Manual de Referencia: *MPLAB® C18 C COMPILER LIBRARIES (DS51297D)*, Microchip Technology Inc., 2004.

Manual de Referencia: *PIC18F46J50 Data Sheet (DS39931D)*, Microchip technology Inc., 2004.

Manual de Referencia: (Versión 2.31r) *RN-42/RN-42-N Class 2 Bluetooth Module Data Sheet*, Roving Networks, 2012.

Manual de Referencia: (version 4.77) *Roving Networks Bluetooth™ Product User Manual*, Roving Networks, 2009.

Manual de Referencia: *SST25VF064C Datasheet (DS25036A)*, Silicon Storage Technology, Inc., 2011.

Márquez Z., Iván C. Diseño de una Unidad de Adquisición de Datos para estaciones hidrometeorológicas de superficie, adaptada a las necesidades del país/ Iván Márquez (Tesis).-- Caracas: Universidad Central de Venezuela, 2009.