



UNIVERSIDAD DE LA REPÚBLICA  
FACULTAD DE INGENIERÍA



Proyecto de fin de carrera

# RED DE INSTRUMENTACIÓN INALÁMBRICA

Octubre 2009

Integrantes:

- Alfonso González
- Natacha Leone
- Mauricio Murdoch

Tutores:

- Pablo Mazzara
- Julián Oreggioni

# Índice

<b>CAPÍTULO 1</b>	<b>8</b>
<b>INTRODUCCIÓN</b>	<b>8</b>
1.1. Motivación	8
1.2. Vínculo con el cliente	8
1.3. Descripción del proyecto	8
1.4. Objetivos generales	10
1.5. Requerimientos	11
1.6. Estructura del documento	12
<b>CAPÍTULO 2</b>	<b>13</b>
<b>PLATAFORMA HARDWARE Y SOFTWARE</b>	<b>13</b>
2.1. Elección de la plataforma hardware	13
2.1.1 Microcontrolador	14
2.1.2 Radio	15
2.1.3 Memoria externa	16
2.1.4 Interfaces	17
2.1.5 Alimentación	17
2.1.6 Elección final	17
2.1.7 Mote Iris	18
2.2. Elección de la plataforma software	19
2.3. Programación para la comunicación	20
2.3.1 Protocolos	20
2.3.2 Parámetros	21
<b>CAPÍTULO 3</b>	<b>23</b>
<b>SENSORES</b>	<b>23</b>
3.1. Sensor de temperatura	23
3.1.1 Elección del integrado	23
3.1.2 DS620	24
3.1.3 Circuito de medición de temperatura	27
3.1.4 Programación del Mote para integrar el sensor de temperatura	27
3.2. Sensor de pH	30
3.2.1 Definición de pH	30
3.2.2 Electrodo de vidrio	30
3.2.3 Electrodo de pH de estado sólido (ISFET)	33
3.2.4 Elección del electrodo	34
3.2.5 Circuito de medición de pH	35
3.2.6 Programación del Mote para integrar sensor de pH	46

<b>CAPÍTULO 4</b>	<b>49</b>
<b>INTERFAZ CON EL USUARIO</b>	<b>49</b>
4.1. Elección de la plataforma	49
<b>4.2. Programación</b>	<b>49</b>
4.2.1 Formato de los mensajes	50
4.2.2 Enviar mensajes a los nodos	51
4.2.3 Recibir mensajes de los nodos	51
4.2.4 Calibrar el sensor de pH de un nodo determinado	53
4.3. Compatibilidad con el hardware del cliente	57
<b>CAPÍTULO 5</b>	<b>58</b>
<b>IMPLEMENTACIÓN</b>	<b>58</b>
<b>5.1. Realización de los sensores</b>	<b>58</b>
5.1.1 Temperatura	58
5.1.2 pH	59
5.1.3 Consideraciones para los diseños de los impresos	59
<b>5.2. Montaje mecánico</b>	<b>60</b>
5.2.1 Vaina de temperatura	61
<b>CAPÍTULO 6</b>	<b>63</b>
<b>VALIDACIÓN</b>	<b>63</b>
<b>6.1. Consumo</b>	<b>63</b>
6.1.1 Consumo teórico	63
6.1.2 Consumo experimental	64
<b>6.2. Parámetro del sensor de pH</b>	<b>69</b>
6.2.1 Tiempo de encendido a pH constante	69
6.2.2 Tiempo de respuesta al escalón	70
<b>6.3. Incertidumbre en la medida de pH</b>	<b>72</b>
6.3.1 Incertidumbre tipo A	72
6.3.2 Incertidumbre tipo B	72
<b>CAPÍTULO 7</b>	<b>75</b>
<b>CONCLUSIONES</b>	<b>75</b>
7.1. Introducción	75
7.2. Objetivos Cumplidos	75
7.3. Resultados Conseguídos	76
7.4. Principales aportes	77

<b>7.5. Principales problemas encontrados</b>	<b>77</b>
<b>7.6. Puntos a mejorar</b>	<b>78</b>
<b>Anexo I</b>	<b>80</b>
<b>Electrodo ISFET</b>	<b>80</b>
<b>Anexo II</b>	<b>84</b>
<b>Protocolo de comunicación I2C</b>	<b>84</b>
<b>Anexo III</b>	<b>86</b>
<b>Funcionamiento llave digital</b>	<b>86</b>
<b>Anexo IV</b>	<b>87</b>
<b>Funcionamiento filtro pasabajos</b>	<b>87</b>
<b>Anexo V</b>	<b>89</b>
<b>Interfaz Mote - sensores</b>	<b>89</b>
<b>Anexo VI</b>	<b>93</b>
<b>Manual de usuario</b>	<b>93</b>
<b>Anexo VII</b>	<b>94</b>
<b>Manual de instalación</b>	<b>94</b>
<b>Anexo VIII</b>	<b>96</b>
<b>Costos</b>	<b>96</b>
<b>REFERENCIAS</b>	<b>97</b>

# Índice de Figuras

Figura 1.1: Esquema de alto nivel de la aplicación desarrollada	10
Figura 2.1: Mote Iris	18
Figura 2.2: Esquema interno del Mote	18
Figura 2.3: Arquitectura de red tipo árbol	21
Figura 3.1: Integrado DS620 de Maxim	24
Figura 3.2: Esquemático del PCB del sensor de temperatura	27
Figura 3.3: Diagrama de flujo de la función de encendido del sensor de temperatura	28
Figura 3.4: Diagrama de flujo de la función de lectura de dato de temperatura	29
Figura 3.5: Esquema del electrodo de combinación	31
Figura 3.6: Pendiente del electrodo ideal a 25°C	32
Figura 3.7: Modelo eléctrico de un electrodo	33
Figura 3.8: Electrodo Mettler Toledo InLab 413	35
Figura 3.9: Esquema del amplificador operacional INA333	37
Figura 3.10: Modelo para estimar el modo común	38
Figura 3.11: Circuito diseñado para la medición de pH	39
Figura 3.12: Esquema interno de un potenciómetro digital de 127 pasos	41
Figura 3.13: Integrado MCP4652 de Microchip	41
Figura 3.14: Esquema interno del integrado de voltaje de referencia MAX6018	43
Figura 3.15: Integrado TS3A4742 de Texas Instruments	45
Figura 3.16: Esquema del circuito del sensor de pH	46
Figura 3.17: Diagrama de flujo de la función de lectura del sensor de pH	47
Figura 3.18: Diagrama de flujo de la función de calibración del sensor de pH	48
Figura 4.1: Display principal de la aplicación	50
Figura 4.2: Diagrama de la función de llegada de mensaje	53
Figura 4.3: Ventana emergente para la calibración del sensor de pH	54
Figura 4.4: Display inicial de calibración de la aplicación	54
Figura 4.5: Diagrama de la función de calibración	55
Figura 4.6: Display de calibración cuando se está tomando el pH de la solución buffer 7	56
Figura 4.7: Display final de la calibración	57
Figura 5.1: Foto del PCB de temperatura realizado	59
Figura 5.2: Foto del PCB de pH realizado	59
Figura 5.3: Soporte para el Mote sensor	60
Figura 5.4: Caja estanca y vaina del sensor de temperatura	61
Figura 5.5: Diseño de la vaina del sensor de temperatura	62
Figura 6.1: Cambio de estado (On-Off) en impreso de pH	64
Figura 6.2: Consumo del impreso de temperatura	65
Figura 6.3: Consumo del impreso de temperatura (ampliación de la Figura 6.2)	66
Figura 6.4: Consumo de la radio con un ciclo de trabajo del 50%	67
Figura 6.5: Consumo de la radio con un ciclo de trabajo del 5%	68
Figura 6.6: Salida analógica, $t_{on}=12s$	69
Figura 6.7: Salida analógica, $t_{on}=45s$	70
Figura 6.8: Respuesta al escalón creciente	71
Figura 6.9: Respuesta al escalón decreciente	71
Figura I.1: Transistor MOSFET (a). Electrodo ISFET (b)	80
Figura I.2: Modelo del electrodo ISFET	80
Figura I.3: Esquema de medida a corriente constante	82
Figura I.4: Esquema para la medición del voltaje de pH	83
Figura II.1: Diagrama de conexionado de una comunicación I2C	84
Figura II.2: Secuencias de inicio y parada del protocolo I2C	85
Figura II.3: Secuencia de bits al comienzo de la comunicación	85
Figura III.1: Esquema de la llave TS3A4742	86
Figura III.2: Resistencia del interruptor	86
Figura IV.1: Filtro pasabajos de primer orden	87
Figura IV.2: Transferencia de un filtro pasabajos de primer orden	87
Figura V.1: Foto del conector de expansión	89

Figura V.2: Conector macho (izquierda) y conector hembra (derecha)	90
Figura V.3: Esquemático del conector	90
Figura V.4: Molex macho (izquierda) y Molex hembra (derecha)	91
Figura V.5: Base de programación Mib510	92
Figura VII.1: Ventana de administrador de dispositivos de WinXP	94

# Índice de Tablas

Tabla 1.1: Resumen de requerimientos generales del sistema	11
Tabla 2.1: Componentes de plataformas hardware comparadas	13
Tabla 2.2: Componentes de plataformas hardware comparadas (continuación)	13
Tabla 2.3: Comparación de los microcontroladores a partir de sus hojas de datos	14
Tabla 2.4: Comparación de los microcontroladores a partir de sus hojas de datos	15
Tabla 2.5: Características de la plataforma Iris	19
Tabla 3.1: Requerimientos principales del sensor de temperatura	24
Tabla 3.2: Características principales del integrado DS620	25
Tabla 3.3: Tabla de mapeo de memoria del integrado DS620	25
Tabla 3.4: Registro de configuración	26
Tabla 3.5: Descripción de los bits del registro de configuración del integrado DS620	26
Tabla 3.6 Características del amplificador de instrumentación INA333	36
Tabla 3.7: Características del interruptor TS3A4742	45
Tabla 6.1: Consumo de los componentes en impresos	63
Tabla 6.2: Incertidumbres y puntos de trabajo	73
Tabla 7.1: Requerimientos vs. Resultados	76
Tabla V.1: Pines de interés del conector de expansión	91
Tabla VIII.1: Costos del proyecto	96

# Capítulo 1

## Introducción

### 1.1. Motivación

La idea del proyecto surge a partir de una charla con el Ing. Conrado Rossi donde nos comento el trabajo que el grupo de microelectrónica estaba haciendo en el área de transmisión de datos en bajo consumo. En particular utilizando los equipos denominados Motes.

En la misma charla surgió la interrogante de probar estos equipos en alguna aplicación en la cual la facultad no halla podido hacerlo antes. Debido al ámbito laboral de alguno de los integrantes del equipo surgió la idea de probar el sistema en un ambiente industrial. Fue así que se comenzó a investigar que posibles aplicaciones podía haber y quien estaría interesado.

### 1.2. Vínculo con el cliente

Después de analizar distintas posibilidades se llego a la conclusión de una de las aplicaciones que mas necesitaría de un sistema de estas características seria el tratamiento del cuero dentro de una curtiembre.

Se decidió plantearle esa misma idea a la curtiembre Zenda ya que por ser una planta de gran tamaño podrían, además de interesarles, tener el capital necesario para financiar el proyecto.

Luego de comentarles la idea general por teléfono se agendó una reunión en la planta. Desde un principio se mostraron interesados en el proyecto y luego de varias reuniones decidieron aceptar que se realizara el prototipo allí y a financiarlo.

### 1.3. Descripción del proyecto

Para la producción del cuero es fundamental poder controlar las variables que afectan la calidad del mismo.

Una vez que los cueros son trasladados a la curtiembre, son almacenados en el saladero hasta que llega el momento de procesarlos. El procesamiento se realiza



en varias etapas.

La primera es la etapa de ribera, en la cual el cuero es preparado para ser curtido, en ella es limpiado y acondicionado asegurándole un correcto grado de humedad. Luego se ejecuta la etapa de piquelado, en el que se realiza la preparación química de la piel para el proceso de curtido, principalmente mediante la utilización de ácido fórmico y sulfúrico. A continuación se lleva a cabo la etapa de curtido, que es el proceso por el cual se estabiliza el colágeno de la piel mediante agentes curtientes minerales, principalmente sales de cromo, o vegetales, transformándola en cuero. Luego viene la etapa de postcurtición, con procesos mecánicos donde se pretende dar un espesor específico y homogéneo al cuero y procesos húmedos tendientes a modificar el colágeno para adecuarlas a artículos determinados. Por último se realiza la etapa de secado y terminación en donde los cueros son desaguados y retenidos para eliminar el exceso de humedad y además son estirados y preparados para luego secarlos<sup>1</sup>

El objetivo del proyecto es controlar las variables en el proceso de pelambre que está incluido dentro de la etapa de ribera. En este proceso se coloca las pieles dentro de un fulón y se disuelve el pelo utilizando cal y sulfuro de sodio. Un fulón es un recinto de madera de gran tamaño como se observa en la figura 2.1. A lo largo del proceso el fulón esta en constante rotación para lograr que todo el cuero dentro del mismo pueda ser tratado.

Las variables de mayor relevancia para este proceso son la temperatura y el pH. El proceso dura alrededor de 24 horas y durante este tiempo el fulón se encuentra permanentemente girando.

Actualmente para sensar la temperatura y el pH se detiene el proceso, se abre una puerta lateral y se toma una muestra de la sustancia del interior. Este método tiene varios inconvenientes. Por un lado no pueden tomarse una gran cantidad de medidas ya que se requiere parar el proceso cada vez que se va a realizar una medición. Como cualquier en cualquier proceso manual pueden existir errores humanos como anotar incorrectamente la medida u olvidar hacerla, además de requerir tener un operador permanentemente para realizarlas. Por otro lado otra gran desventaja es que al abrir el fulón el operario queda expuesto a los vapores del interior que pueden ser nocivos para la salud.

Otro método que se emplea actualmente es la utilización de dataloggers que recaban la información durante el proceso y luego de que termina el mismo se puede levantar la información con un PC y analizarla. Este proceso evita los problemas del anterior, ya que ni es manual ni se detiene el proceso. Como contrapartida deja de ser en tiempo real, no se puede controlar el proceso que está sucediendo sólo se puede analizar el proceso que ya ha concluido. Además el cliente comenta que a veces se pierden los dataloggers cuando sacan el cuero del fulón generando una pérdida económica y de información.

Para el cliente es muy deseable automatizar el proceso en tiempo real y esto puede realizarse de distintas formas.

Una forma de hacerlo es utilizando contactos deslizantes<sup>1</sup> tanto para alimentar a los sensores dentro del fulón como para enviar al exterior las medidas

---

<sup>1</sup> También llamados escobillas de carbón.

tomadas por los mismos. Este método requiere de un alto mantenimiento debido a que los carbones de las escobillas se gastan frecuentemente<sup>2</sup>.

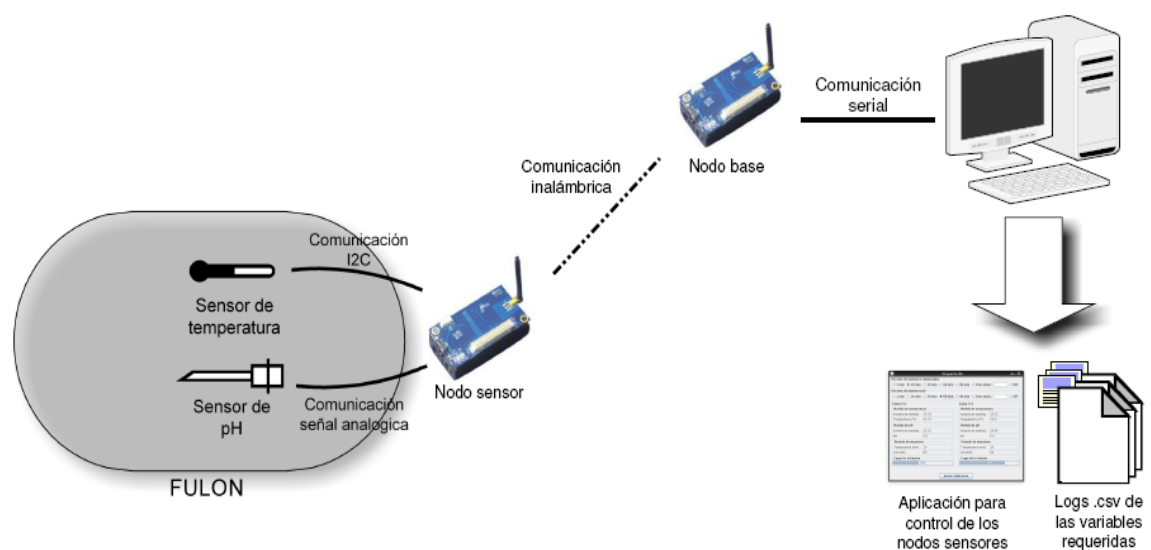
Otra forma de lograr automatizar el sensado es mediante sensores inalámbricos con fuente de alimentación propia. Este tipo de sensado permite obtener medidas tan frecuentemente como se quiera en tiempo real sin necesidad de interrumpir el proceso. Además se eliminará la posibilidad de errores humanos al tomar o registrar la medida y no habrá riesgos para la salud del operario. El sistema a montar en los fulones deberá de ser confiable, robusto y de muy bajo consumo para requerir el menor mantenimiento posible.

Este proyecto intentará implementar un sistema de este tipo. Para ello se utilizará un sistema de transmisión inalámbrica, que constará de un nodo base conectado a un PC y nodos montados en cada recinto donde se quiere realizar las mediciones y que tienen asociados sensores para obtener las mismas.

## 1.4. Objetivos generales

El objetivo del proyecto es realizar medidas dentro del fulón, a través de un dispositivo de transmisión inalámbrica de bajo consumo que permita una alimentación a pilas. Para esto se utilizarán unos nodos inalámbricos llamados Motes por su ligereza y tamaño reducido<sup>2</sup>. Los Motes son dispositivos electrónicos capaces de recibir información a través de sus interfaces, procesarla y transmitirla inalámbricamente hacia otro Mote o hacia una PC. Un Mote consiste en un PCB que tiene un microprocesador que implementa todo el control del Mote, una radio para comunicarse y un dispositivo para almacenar información (memoria FLASH).

En la Figura 1.1 se observa un esquema de alto nivel de la aplicación desarrollada.



**Figura 1.1: Esquema de alto nivel de la aplicación desarrollada**

<sup>2</sup> Mote significa pequeña partícula en inglés.

Se realizará e instalará un prototipo dentro de un fulón donde se sensarán las variables temperatura y pH, para luego enviar los datos en forma inalámbrica a un nodo base conectado a un PC. También se realizará una aplicación capaz de recibir, procesar y mostrar los datos recabados por el Mote remoto.

## 1.5. Requerimientos

El cliente impone ciertos requerimientos a partir del proceso que se quiere controlar. El sensor de temperatura debe poder trabajar en un rango desde 15°C hasta 45°C con una precisión de 0,5°C. El sensor de pH debe poder trabajar entre valores de 3 y 7 pH con una precisión de 0,1 pH.

Para que la solución propuesta sea práctica la autonomía de cada nodo sensor deberá ser de al menos dos meses utilizando pilas alcalinas y con periodos de muestreo no menores a una hora.

Además debe ser desmontable para poder realizar la calibración del pHímetro en el laboratorio y para poder cambiarle las pilas.

---

### Resumen de requerimientos del sistema

---

#### Requerimientos generales

Costo	El costo total del desarrollo deber ser inferior a U\$S 1500.
Interfaz con el usuario	Sencilla y clara. Debe mostrar los últimos datos recabados por los nodos sensores.
Registro de datos	Guardar en un archivo csv todas las medidas que le llega al nodo base.
Escalabilidad	Lograr a partir de un prototipo, aumentar el número de sensores en la red fácilmente

#### Requerimientos del nodo base

Comunicación con la PC	Debe poder funcionar con Windows XP.
Tiempo de vida	Máximo posible, al menos un año.

#### Requerimientos del nodo sensor

Variable temperatura	Rango: entre 15 y 45 °C, precisión: $\pm 0,5$ °C.
Variable pH	Rango: 3 y 7 pH, precisión: $\pm 0,1$ pH.
Autonomía	2 meses con 2 pilas AA alcalinas y periodos de muestreo de al menos una hora.
Portabilidad	Debe poderse desmontar del fulón.
Tiempo de vida	Máxima posible.
Periodo de muestreo	Debe poderse seleccionar, el período mínimo debe ser de un minuto.
Calibración de pH	Debe ser lo más sencilla posible del lado del usuario.

---

Tabla 1.1: Resumen de requerimientos generales del sistema

## 1.6. Estructura del documento

La presente documentación está formada por 7 capítulos y 8 anexos.

En el segundo capítulo se presentan las plataformas hardware y software consideradas para este proyecto y cual de ellas finalmente se eligió y porqué. En este capítulo se explica también las bases para la realización de la programación de la plataforma.

En el capítulo tres se exponen los sensores de temperatura y de pH. Se explican las distintas tecnologías que se estudiaron, cuales fueron las elegidas y los motivos por los cuales se eligió. También se explica como se realizó el diseño de la adaptación de los sensores con las plataformas elegidas en el capítulo 2.

El capítulo cuatro se refiere a la interfaz con el usuario, es decir a la programación realizada para crear la aplicación con la que el usuario interactúa.

En el quinto capítulo se detalla como fueron realizados los distintos dispositivos requeridos en el proyecto. También se expone como se realizó el montaje de los componentes en la planta del cliente.

El capítulo seis muestra los ensayos realizados con el fin de validar el sistema creado. También se muestran las pruebas realizadas en planta con el mismo fin.

El último capítulo, el siete, se refiere a las conclusiones que se obtuvieron. Estas se refieren a cuales objetivos se alcanzaron, que resultados se obtuvieron, cuales fueron los principales aportes y los principales inconvenientes del proyecto. También se mencionan cuales son los puntos a mejorar por posibles futuros proyectos.

# Capítulo 2

## Plataforma hardware y software

### 2.1. Elección de la plataforma hardware

El hardware de un Mote consta de cinco grandes partes:

- Radio
- Microcontrolador
- Memoria externa
- Interfaces
- Alimentación

En la Tabla 2.1 y en la Tabla 2.2 se muestran distintas plataformas encontradas y sus respectivas características.

Componentes de plataformas hardware comparadas			
Nombre	Microcontrolador	Radio	Interfaces
BTnode	Atmel ATmega 128L	Chipcon CC1000	UART, SPI, I2C, GPIO, ADC 12bit
Iris	Atmel ATmega 1281	Atmel AT86RF230	UART, SPI, I2C, GPIO, ADC 10bit
MicaZ	Atmel ATmega 128L	Chipcon CC2420	UART, SPI, I2C, GPIO, ADC 10bit
TelosB	Texas Instruments MSP430	Chipcon CC2420	UART, SPI, I2C, GPIO, ADC 12bit
TinyNode	Texas Instruments MSP430	Xemics XE 1205	UART, SPI, GPIO, ADC 12bit

Tabla 2.1: Componentes de plataformas hardware comparadas

Componentes de plataformas hardware comparadas (continuación)			
Nombre	Memoria de programación	Memoria externa	Plataforma software soportada
BTnode	64k RAM	128k Flash	BTnut y TinyOS
Iris	8k RAM	128k Flash	Moteworks y TinyOS
MicaZ	4k RAM	128k Flash	Moteworks y TinyOS
TelosB	10k RAM	512k Flash	Contiki, TinyOS, SOS y Mantis
TinyNode	8k RAM	512k Flash	TinyOS

Tabla 2.2: Componentes de plataformas hardware comparadas (continuación)

En todas las plataformas encontradas la alimentación de los Motes se tiene a través de dos pilas AA.

Cuando se tuvo que elegir la plataforma hardware se enfrentó el problema de que cada proveedor entregaba una hoja de datos con distinta información. En algunos casos la información brindada no daba a conocer bajo que condiciones se realizaron los experimentos. Fue por esto que se decidió comparar los componentes de cada plataforma ya que sobre estos si se contaba con la información suficiente.

### 2.1.1 Microcontrolador

El microprocesador es el componente que interpreta los datos que le llega por sus interfaces y los procesa para su transmisión a otro nodo o a un PC.

En la Tabla 2.3 se muestran las características más importantes de los microcontroladores de las plataformas comparadas.

<b>Comparación de los microcontroladores a partir de sus hojas de datos</b>			
	Atmel ATmega 128L	Atmel ATmega 1281	Texas Instruments MSP430
Voltaje de alimentación	2.7...5.5V	1.8...5.5V	1.8...3.6V
Consumo en modo activo	3.6 mA @ 4MHz, 3.0V	0.5 mA @ 1MHz, 1.8V	0.2 mA @ 1MHz, 2.2V
Consumo en modo idle	1.0 mA @ 1MHz, 3.0V	---	0.7 $\mu$ A @ 1.8V
Consumo en modo apagado	0.5 $\mu$ A @ 3.0V	0.1 $\mu$ A @ 1.8V	0.1 $\mu$ A @ 2.2V

**Tabla 2.3: Comparación de los microcontroladores a partir de sus hojas de datos**

El microprocesador Atmega 128L tiene un requerimiento de voltaje mucho más elevado que el resto de los micros, funciona con al menos 2.7V. El sistema va a ser diseñado para trabajar entre 1.8V y 3.6V por lo que un microcontrolador que funcione hasta 2.7V estaría bajando la autonomía mismo.

En modo activo se observa una diferencia grande entre los consumos del Atmega 128L y los 2 restantes. Esto también incide en la autonomía del sistema. Cabe aclarar que nuestro sistema esta diseñado para minimizar los tiempo en que el microcontrolador esta en modo activo. Una de las claves de los Motes son los diferentes modos de operación de los componentes del mismo (micro y radio). Y la eficiencia en su uso dependerá también de la aplicación corriendo en el microcontrolador por lo que una comparación en profundidad de los consumos involucra temas que van más allá del alcance de este proyecto.

Sin embargo la comparación de consumos en modos activos y “idle” nos da una aproximación a la realidad. Dentro del capítulo de validación se verificara si los consumos teóricos son correctos.

Teniendo en cuenta lo anteriormente mencionado podemos descartar las plataformas MicaZ y BTnode ya que tienen el microcontrolador Atmega 128L que

tiene consumos más grandes.

Otro punto importante en la elección del microprocesador son los espacios de memoria que ofrece. Ya sea memoria no volátil, para cargar la aplicación a correr, como la memoria RAM para usar en tiempo de ejecución. En estos casos los 3 microprocesadores analizados cumplen con los requisitos mínimos.

## 2.1.2 Radio

La radio permite enviar y recibir datos para comunicarse con otros dispositivos dentro de su rango de transmisión.

Todos los Motes utilizan la banda ISM (llamada así por las siglas en inglés de Industrial, Scientific and Medical). Esta banda es reservada internacionalmente, no necesitándose licencia para utilizarla pero teniendo que respetar ciertos niveles de potencia transmitida.

Las radios están formadas básicamente por un transceiver que es un dispositivo capaz de transmitir y de recibir señales electromagnéticas. Existen cuatro estados de operación, estos son emisión, recepción, dormido y apagado. En estos sistemas la radio se encuentra la mayor parte del tiempo apagada, despertándose de a intervalos para escuchar el canal a la espera de paquetes o para transmitirlos.

En la Tabla 2.4 se muestran las características más importantes de las radios de las plataformas comparadas. Nos centraremos en la comparación de las radios Xemics XE y Atmel AT86RF230 ya que estas corresponden a las plataformas que cuentan a priori con las mejores características de sus microprocesadores.

Comparación de los radios a partir de sus hojas de datos				
	Chipcon CC1000	Atmel AT86RF230	Chipcon CC2420	Xemics XE 1205
Voltaje de alimentación	2.1...3.6V	1.8...3.6V	2.1...3.6V	2.4...3.6V
Consumo en modo dormido	0.2 $\mu$ A	0.1 $\mu$ A	426 $\mu$ A	0.2 $\mu$ A
Consumo al recibir	9.6 mA	16 mA	18.8 mA	14 mA
Consumo al transmitir	25.4 mA @ 5dBm	17 mA @ 3dBm	17.4 mA @ 0dBm	33 mA @ 5dBm
Throughput	1.2-500Kbps	250Kbps	250Kbps	---
Frecuencias	315/433/915 MHz	2.4 GHz	2.4 GHz	---

**Tabla 2.4: Comparación de los microcontroladores a partir de sus hojas de datos**

En principio cabe acotar que la comparación con la radio C1000 no es posible ya que esta es obsoleta frente a las restantes, esta radio se basa en bytes. Pero aún así es considerada ya que sigue siendo usada por algunas plataformas hardware (en particular TinyNode).

Se observa que las frecuencias en que trabajan los radios (menos para el radio CC1000) son en el orden de los GHz. Esto es fundamental ya que el prototipo se instalará dentro de un ambiente industrial y se necesita transmitir en un ancho de banda donde las interferencias electromagnéticas no den problemas.

Una característica importante de todos los componentes del sistema es el rango de alimentación. En el caso de los radios la única que cumpliría este requerimiento sería la AT86RF230. Aunque algunas otras tienen rangos similares, para terminar optando por una de estas otras se tendría que haber una diferencia importante en el consumo y así tener una autonomía mayor.

Para hacer un análisis del consumo de la radio se necesitan analizar varios aspectos, algunos de estos se encuentran en la Tabla 2.4. El consumo en la transmisión va a depender de la potencia a la cual la radio transmite y el tiempo que necesita para transmitir 1 paquete. Para esto último no solo necesitamos la velocidad de transmisión sino que se tendría que analizar los paquetes a enviar, en particular un análisis de la capa MAC en la aplicación cargada en el Mote. También para el caso de la recepción se tendría que tomar en cuenta varios factores, no solo el consumo de la radio al recibir. En este aspecto no se notan grandes diferencias entre los radios en el consumo.

Como en el caso del microcontrolador este análisis exhaustivo no está comprendido en el alcance del proyecto. Pero para poder hacer una elección se compararon algunos datos de los radios para poder tener una aproximación.

Tomando en consideración lo anteriormente mencionado se observa que la AT86RF230 tiene el rango de alimentación más bajo y consumos ligeramente menores.

### **2.1.3 Memoria externa**

Los Motes tienen una memoria integrada en el chip del microcontrolador y una memoria flash.

La memoria flash es usada debido a su bajo costo y a su gran capacidad de almacenamiento. La memoria flash es una forma desarrollada de la memoria EEPROM que permite que múltiples posiciones de memoria sean escritas o borradas en una misma operación de programación. La memoria flash es de bajo consumo y de carácter no volátil.

En la Tabla 2.2 se encuentran los tamaños de memoria tanto de programación y datos, como los de memoria flash para las distintas plataformas hardware.

Para esta aplicación los tamaños de memoria de todas las plataformas resultan suficientes. Esto se debe a que las rutinas de configuración de los Motes son relativamente pequeñas, no se guardarán medidas adquiridas en cada nodo y se guardarán pocos parámetros de configuración de los sensores en la memoria flash.



### 2.1.4 Interfaces

Todos los Motes tienen las interfaces UART, SPI, I2C y entradas y salidas de propósito general. Para este proyecto se utilizará algunas de las dos interfaces de comunicación serial, SPI o I2C para la comunicación con el sensor de temperatura. También se utilizarán algunas salidas y entradas digitales y al menos una entrada analógica.

Los Motes se diferencian en la cantidad de bits que tienen sus conversores analógicos digitales, son o de 10 bits o de 12 bits. Para la resolución que se utilizará ya es suficiente un convertor de 10 bits.

Con respecto a las interfaces cualquiera de los Motes comparados servirían para esta aplicación.

### 2.1.5 Alimentación

En todas las plataformas encontradas la alimentación de los Motes se tiene a través de dos pilas AA. Entonces, el voltaje nominal de la alimentación será de 3V, pudiendo inicialmente ser mayor a este valor, e irá decayendo con el tiempo. Del análisis de la hoja de datos de una pila AA se estima que el sistema podría permanecer operativo hasta  $1.8V^3$ . Es por esto que es tan importante elegir un Mote que pueda trabajar con el menor requerimiento de tensión posible.

En el caso del Mote que se usara como base (nodo central de comunicaciones) y que se conectara a la interfaz con el usuario (en el PC) es importante que pueda tener alimentación externa.

En este aspecto TelosB tiene una importante ventaja ya que tiene el puerto USB integrado al Mote. Esto permite usar la misma plataforma tanto para los nodos sensores como para el nodo base.

La línea Mica (Iris y MicaZ) de Crossbow tiene interfaces para conectar a los Motes con un PC, llamadas placas de programación. Esta interfaz se puede usar tanto para programar como para comunicar entre el Mote y el PC. La conexión entre el Mote y la placa de programación se realiza a través del puerto de 51 pines. La conexión entre la placa y PC puede ser tanto por USB, serial (RS232) o ethernet, dependiendo del modelo de placa elegido.

### 2.1.6 Elección final

De las comparaciones de los componentes de cada Mote concluimos que las plataformas que mejor se adecuan a nuestros requerimientos son TelosB e Iris, ambas de la marca Crossbow. Para elegir entre estas dos plataformas se realizó una comparación cualitativa de las mismas.

TelosB es una plataforma que ya tiene varios años con lo que existe una gran cantidad de información sobre la misma. Además, es la plataforma que está utilizando el Grupo de Microelectrónica del Instituto de Ingeniería Eléctrica, por lo que se puede tener un mayor soporte por parte de los docentes.

TelosB es un desarrollo de UC Berkeley comprado por Crossbow que no brinda soporte sobre la plataforma. En cambio, Iris es uno de los desarrollos más nuevos de Crossbow para el que se brinda soporte.

Finalmente se decidió por el modelo Iris ya que no había sido usado en la facultad anteriormente, y por tanto puede brindar nuevos conocimientos.

### 2.1.7 Mote Iris

El modelo Iris es la última plataforma diseñada por Crossbow. En la Figura 2.1 se muestra una foto del mismo.



Figura 2.1: Mote Iris

En la Figura 2.2 se observa un esquema interno del Mote. En este se ven los principales componentes del Mote, como son la radio, la memoria no volátil, los leds y el conector de expansión de 51 pines. El conector de expansión está íntegramente conectado al microprocesador del Mote, salvo la alimentación que está conectada directamente a la pila. Cabe destacar que no todas las señales del micro están accesibles a través del conector.

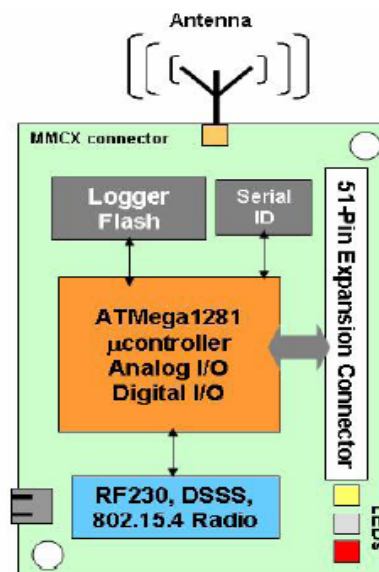


Figura 2.2: Esquema interno del Mote

En la Tabla 2.5 se muestran las principales características de la plataforma seleccionada. Estos datos fueron obtenidos a partir de la hoja de datos obtenida en la página oficial de Crossbow.

<b>Características de la plataforma Iris</b>	
<b>Performance del procesador</b>	
Memoria flash programable	128 kB
RAM	8 kB
EEPROM de configuración	4 kB
Comunicación serial	UART
Convertor digital analógico	10 bit ADC
Otras interfaces	E/S digital, I2C, SPI
Consumo en modo activo	8 mA
Consumo en modo “sleep”	8 $\mu$ A
<b>Transceiver de RF</b>	
Banda de frecuencia	2405 a 2480 MHz
Velocidad de transmisión	250 kbps
Potencia de RF	3 dBm (típica)
Sensibilidad al recibir	101 dBm (típica)
Rango de cobertura en exterior	> 300 m
Rango de cobertura en interior	> 50 m
Consumo al transmitir	10 mA @ -17 dBm 17 mA @ 3 dBm
Consumo al recibir	16 mA
<b>Electromecánico</b>	
Pilas	2 pilas AA
Tamaño	58 x 32 x 7 (sin el porta pila)
Peso	18 g (sin pilas)
Conector de expansión	51-pin

**Tabla 2.5: Características de la plataforma Iris**

## 2.2. Elección de la plataforma software

Dado que elegimos la plataforma hardware Iris tenemos dos plataformas software posibles: TinyOS 2.0 y MoteWorks.

TinyOS es un sistema operativo de código abierto específico para redes de bajo consumo. Proporciona módulos y funciones que permiten armar las aplicaciones de los nodos. Estas son escritas en nesC que es un lenguaje de programación basado en C diseñado para optimizar la programación de TinyOS.<sup>4</sup>

Moteworks 2.0 es una aplicación propietaria de Crossbow basada en TinyOS 1.1. Provee un ambiente de desarrollo completo para realizar las aplicaciones. Para hacer las aplicaciones se utilizan las librerías de TinyOS así como también librerías propias de Moteworks.

Se decidió utilizar TinyOS por los motivos que se detallan a continuación:

- MoteWorks tiene una política de derechos de autor que TinyOS no la tiene. Si bien dice que puede ser utilizado en proyectos con fines académicos dice que no puede ser utilizado libremente para productos comerciales.
- TinyOS es la plataforma utilizada por el Grupo de Microelectrónica del Instituto de Ingeniería Eléctrica lo que permite un mayor asesoramiento por parte de ellos.
- TinyOS permite un manejo más flexible del Mote ya que es de más bajo nivel que Moteworks. Por lo que TinyOS permite centrarse más en el manejo de cada elemento como puede ser la radio o el micro.

## 2.3. Programación para la comunicación

Las medidas y los parámetros de configuración necesitan ser enviados entre los nodos sensores y el nodo base. Para esto se programó una aplicación que utiliza protocolos de comunicación ya implementados en TinyOS.

Para comunicar los datos recabados por los nodos sensores se implementa una arquitectura de red de tipo árbol, donde en las puntas del mismo se encuentran los nodos sensores y en la base el nodo base. De esta forma los paquetes son encaminados al nodo base a través del árbol. Para esta comunicación se utiliza el protocolo de TinyOS Collection.

También se necesita configurar parámetros en los nodos sensores. En este caso los valores van a ser iguales en todos los nodos sensores pero pueden cambiar dependiendo de las necesidades del operador. Para esto se utiliza un protocolo de TinyOS llamado Dissemination.

### 2.3.1 Protocolos

#### Collection

Collection es un protocolo de comunicación que arma una estructura de árbol con todos los nodos de la red. El protocolo recolecta la información de los nodos sensores hacia el nodo base.

El protocolo es de tipo *multi-hop, best-efford*. Que sea *best-efford* quiere decir que no asegura la llegada de paquetes a la base. Esto es uno de los puntos claves en la validación ya que la aplicación necesita de una baja tasa de pérdida de paquetes.

De igual manera el protocolo implementa la retransmisión de paquetes hacia otros nodos, al ser *multi-hop*, lo cual mejora sensiblemente la tasa de pérdida de paquetes. Cada Mote examina la calidad del enlace con todos los nodos que tiene

en su alcance y rutea paquetes (de él mismo o de otros) hacia ese Mote. Esta tabla de calidades de enlaces es modificada dinámicamente de forma que la red se adapta a cambios.

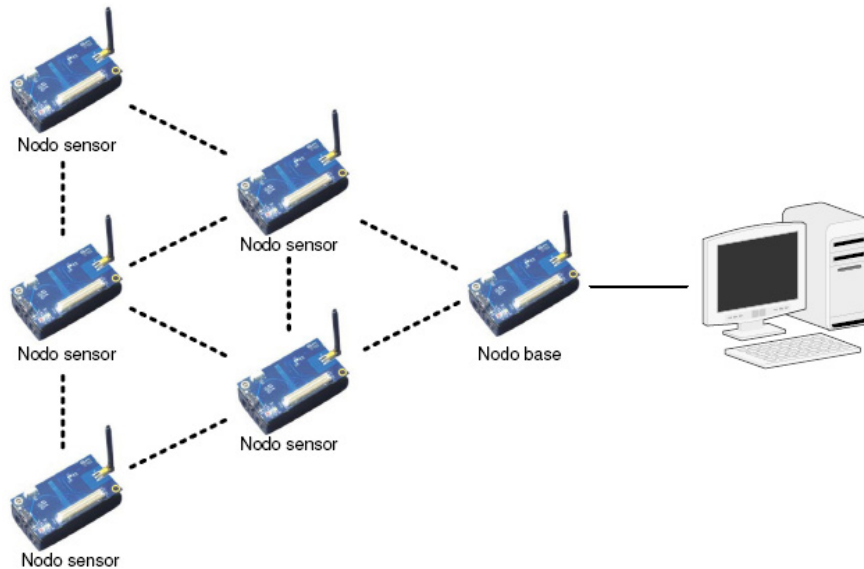


Figura 2.3: Arquitectura de red tipo árbol

## Dissemination

Este protocolo sincroniza valores para todos los nodos de la red. No transmite paquetes aislados con datos sino que se asegura que cierto valor será el mismo en todos los nodos de la red. No existen diferencias entre los nodos en la red (como si existen en el Collection) sino que cualquiera puede cambiar los datos sincronizados y el cambio se verá reflejado en todos los demás nodos. A pesar de esto, en este proyecto no se lo utilizará así, sino que el nodo base será el único que cambiara los valores para sincronizar en la red.

Esto hace que sea adecuado para sincronizar parámetros de configuración de los nodos sensores. Aunque los cambios no se verán reflejados inmediatamente ya que el protocolo asegura que la red va a tender al sincronismo pero no asegura cuanto tiempo le llevará.

### 2.3.2 Parámetros

Dentro de la aplicación existen algunos parámetros para configurar la comunicación que hacen cambiar su performance.

La radio es el componente hardware de mayor consumo y por tanto su uso eficiente va a tener fuerte incidencia en la autonomía de los nodos. Dentro de las interfaces usadas para la comunicación se encuentra *LowPowerListening (LPL)*. Esta provee las funciones necesarias para manejar los periodos en que la radio se encuentra prendida, permitiendo que la misma este más tiempo en modo *dormida* y así lograr bajar sustancialmente el consumo promedio de la radio. Esto nos trae en contraposición que es más difícil que lleguen mensajes a los nodos sensor, aunque

sin cambiar la tasa de pérdida de paquetes enviados del nodo sensor al nodo base. Lo anterior se debe a que si el microprocesador necesita enviar algo la radio se prende transmitirlo.

La pérdida de paquetes enviados por el nodo base al nodo sensor es grande, debido a que para lograr comunicar el mensaje, éste tiene que ser enviado cuando la radio de receptor este encendida y dicha radio sólo esta encendida un pequeño porcentaje del tiempo. Aún así se logra enviar datos a los nodos sensores ya que el protocolo que se utiliza (*Dissemination*) en lugar de enviar un sólo paquete lo que hace es mantener sincronizada la red enviando paquetes constantemente.

En la sección 6.1.2 se analizaran los ensayos de consumo y las pruebas para determinar el valor del parámetro.

# Capítulo 3

## Sensores

### 3.1. Sensor de temperatura

#### 3.1.1 Elección del integrado

El sensor de temperatura deberá cumplir con requerimientos impuestos por el cliente y por el Mote. El cliente requiere poder medir temperatura con una precisión de  $\pm 0.5$  °C en un rango de temperatura de 15 a 45 °C. Por otro lado, es deseable que el sensor de temperatura pueda trabajar mientras lo hace el Mote, por lo que deberá de funcionar en al menos un rango de 1.8 a 3 V. Además puesto que toda la instrumentación deberá ser de bajo consumo, el integrado de temperatura también deberá serlo. Las características principales que debe cumplir el integrado de temperatura son:

- *Rango de alimentación:* entre 1.8 y 3.0 V para poder trabajar mientras esté funcionando el mote
- *Rango de temperatura a medir:* entre 15 y 45 °C, este es el rango de temperatura en el que se encontrarán los recintos a medir.
- *Precisión:* el cliente pide que se tenga una precisión en la medida de temperatura de no menos de 0.5°C.
- *Interfaz Serial:* puede ser SPI o I2C. Que se tenga una comunicación serial tiene dos principales ventajas: las señales digitales son más inmunes al ruido y es más sencilla su integración al sistema.
- *Bajo consumo*

Se buscó un integrado para sensar la temperatura con comunicación serial debido a que con este tipo de tecnología se contaban con ciertas características que se requería para el sistema:

- *Facilidad a la hora de integrarlo al sistema:* Para comunicarse con el dispositivo se dispone de una interfaz serial. Esto facilita la integración por varios motivos. En primera instancia elimina todo tipo de circuito para adaptar la señal analógica (para leer el dato de temperatura). Además debido a que la comunicación serial es direccional es posible la configuración dinámica según los requerimientos del sistema.

- *Eliminación del ruido:* la medida de temperatura será comunicada en forma digital por serial lo que la hace más inmune a ruido electromagnético.
- *Menor consumo:* Los dispositivos permiten modos de trabajo donde no se encuentran prendidos todo el tiempo sino que la mayor parte del tiempo pasan “dormidos”.

A partir de los requisitos anteriores se preseleccionaron integrados que cumplieran con ellos para luego realizar una comparación más profunda de los mismos. En la Tabla 3.1 se muestran las características más relevantes de los integrados considerados.

Principales características de los integrados de temperatura comparados			
	DS620	DS7505	LM95071
Rango de entrada	1.7 – 3.5 V	1.7 – 3.7 V	2.4 – 5.5 V
Precisión en el rango de operación <sup>3</sup>	± 0.5 °C	± 0.5 °C	± 1.0 °C
Interfaz de comunicación	I2C	I2C	SPI
Consumo en modo activo	800 µA	750 µA	520 µA
Consumo en modo apagado	2 µA	2 µA	28 µA
Tiempo de conversión de temperatura	25 ms @ 0.5°C de res.	25 ms @ 0.5°C de res.	130 ms @ 1.0°C de res.

Tabla 3.1: Requerimientos principales del sensor de temperatura

De la Tabla 3.1 se ve que el LM90571 tiene una precisión menor que la de los demás integrados, por otra parte no llega a funcionar con 1.8V de alimentación como es requerido. Los otros dos integrados, DS620 y DS7505, son muy similares por lo que servirían cualquiera de ellos.

Se importaron tanto el DS7505 como el DS620 y eligió utilizar el modelo DS620 porque fue el que llegó primero al momento de la importación.

### 3.1.2 DS620

Un esquema del integrado seleccionado se puede ver en la Figura 3.1 y sus características se muestran en la Tabla 3.2.

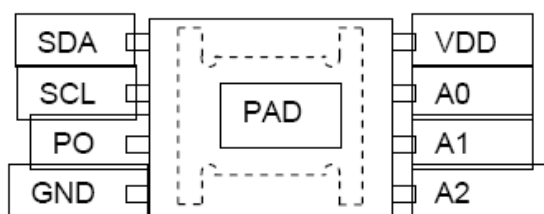


Figura 3.1: Integrado DS620 de Maxim<sup>4</sup>

<sup>3</sup> Rango de operación de 15 a 45 °C

<sup>4</sup> Imagen extraída de la hoja de datos del dispositivo, <http://datasheets.maxim-ic.com/en/ds/DS620.pdf>



---

### Características principales del integrado DS620

---

Rango de voltaje de entrada	1,7 ... 3,5 V
Rango de operación	-55 ... 125 °C
Precisión	± 0.5 °C @ 0...70 °C
Resolución	10, 11, 12, 13 bits (0,5; 0,25; 0,125 y 0,0625 °C)
Velocidades de conversión	25, 50, 100 y 200 ms respectivamente
Interfaz	Serial I2C <sup>5</sup>
Consumo modo idle	2 uA
Consumo en conversión	800 uA

---

**Tabla 3.2: Características principales del integrado DS620**

También cuenta con otras características como lo es el termostato, una salida digital cuyo rango es configurable. Pero estas no son utilizadas para el sensor de temperatura de este proyecto.

El integrado tiene 3 patas para fijar su ID. Con este cada dispositivo es diferenciado en el bus I2C. La dirección es una composición de 1 0 0 1 A2 A1 A0.

Para el funcionamiento el DS620 cuenta con 2 modos de operación, “1-shot mode” y conversiones continuas. En el modo “1-shot” la conversión solo se inicia con comando externo (por I2C) y luego de esta el dispositivo pasa de nuevo a modo idle. Además permite seleccionar como arranca el dispositivo, convirtiendo o esperando.

Todos estos parámetros y los presentados en la son modificables en “runtime”. Para esto se envían escrituras (I2C) al dispositivo a direcciones de memoria específicas, como se muestra en la Tabla 3.3: Tabla de mapeo de memoria del integrado DS620.

---

**Tabla de mapeo de memoria del integrado DS620**

---

A0	TH MSB
A1	TH LSB
A2	TL MSB
A3	TL LSB
A4	Definida por usuario
A5	Definida por usuario
A6	Definida por usuario
A7	Definida por usuario
A8	No definida
A9	No definida
AA	Temperatura MSB
AB	Temperatura LSB
AC	Configuración MSB
AD	Configuración LSB

---

**Tabla 3.3: Tabla de mapeo de memoria del integrado DS620**

Como se observa en la Tabla 3.3 para configurar el dispositivo se necesitan escribir los registros AC y AD (hex). En la Tabla 3.3 se muestra como utilizar el

---

<sup>5</sup> Por más información sobre el protocolo de comunicaciones I2C ver anexo II

registro de configuración y en la Tabla 3.5 se explica a que corresponde cada bit.

<b>MS Byte</b>	<i>Bit 15</i>	<i>Bit 14</i>	<i>Bit 13</i>	<i>Bit 12</i>	<i>Bit 11</i>	<i>Bit 10</i>	<i>Bit 9</i>	<i>Bit 8</i>
	DONE	NVB	THF	TLF	R1	R0	AUTOC	1SHOT
<b>LS Byte</b>	<i>Bit 7</i>	<i>Bit 6</i>	<i>Bit 5</i>	<i>Bit 4</i>	<i>Bit 3</i>	<i>Bit 2</i>	<i>Bit 1</i>	<i>Bit 0</i>
	PO2	PO1	A2	A1	A0	M	M	M

**Tabla 3.4: Registro de configuración**

<b>Tabla de mapeo de memoria del integrado DS620</b>		
Nombre del bit	Tipo de acceso	Descripción
DONE	solo lectura	1: Conversión en progreso 0: conversión terminada
NVB	solo lectura	1: Escritura en EEPROM en progreso 0: EEPROM no ocupada
THF, TLF	lectura/escritura	Registros para manejar termostatos
R1, R0	lectura/escritura	Fija resolución: 00 – 10 bits 01 – 11 bits 10 – 12 bits 11 – 13 bits
AUTOC	lectura/escritura	1: Inicio (del apagado) realizando una conversión 0: Inicia sin conversión
1SHOT	lectura/escritura	Configuración de conversión 1: convierte una vez por cada comando 0: convierte continuamente
PO2, PO1	lectura/escritura	Registros para configuración de pines PO1 y PO2
A2, A1, A0	solo lectura	Muestra estado de los pines A2, A1 y A0
M	lectura/escritura	Bits de uso general

**Tabla 3.5: Descripción de los bits del registro de configuración del integrado DS620**

Además se tiene una lista de comandos necesarios para interactuar con el dispositivo:

- *Comenzar a convertir [51h]:* Inicia una conversión o si esta en modo continuo pero sin arrancar a convertir entonces inicia las conversiones.
- *Parar de convertir [22h]:* Interrumpe las conversiones si esta en modo continuo.
- *Copiar desde EEPROM [B8h]:* Copia registros desde la EEPROM
- *Copiar a EEPROM [48h]:* Copia registros a EEPROM

### 3.1.3 Circuito de medición de temperatura

A partir de las especificaciones de la hoja de datos y las notas de aplicación se realizó el diseño del PCB de temperatura. En la Figura 3.2 se observa un esquemático del circuito diseñado.

En el esquemático de la Figura 3.2 se aprecia el integrado de temperatura DS620 con resistencias de pull-up en sus patas de comunicación I2C y un condensador entre fuente y tierra.

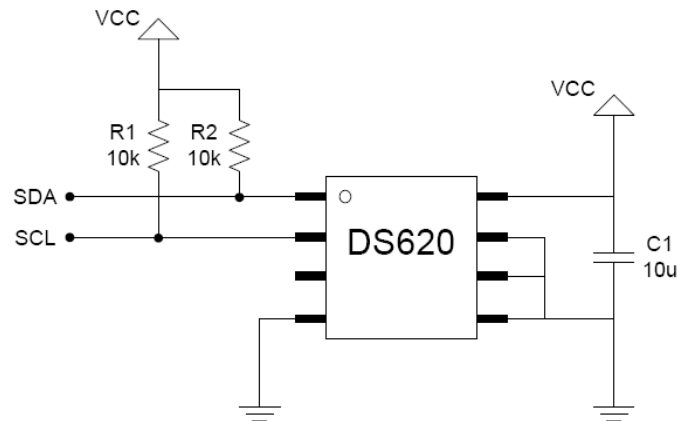


Figura 3.2: Esquemático del PCB del sensor de temperatura

### 3.1.4 Programación del Mote para integrar el sensor de temperatura

Para obtener la medida del sensor de temperatura se utilizó uno de los componentes implementados por TinyOS (*Atm128I2CMasterC*). Este ofrece 2 interfaces: *Resource* y *I2CPacket*. Este componente nos ofrece las funciones necesarias para la comunicación I2C entre el micro y el dispositivo (DS620).

Además se programo un componente específico del dispositivo para acceder desde la aplicación principal. Este cuenta con 2 interfaces: *SplitControl* (para manejar el encendido y apagado) y *Read* (que devuelve la medida). Este componente cuenta con la información necesaria para configurarlo y tiene implementadas las rutinas necesarias para su uso (encendido y lectura).

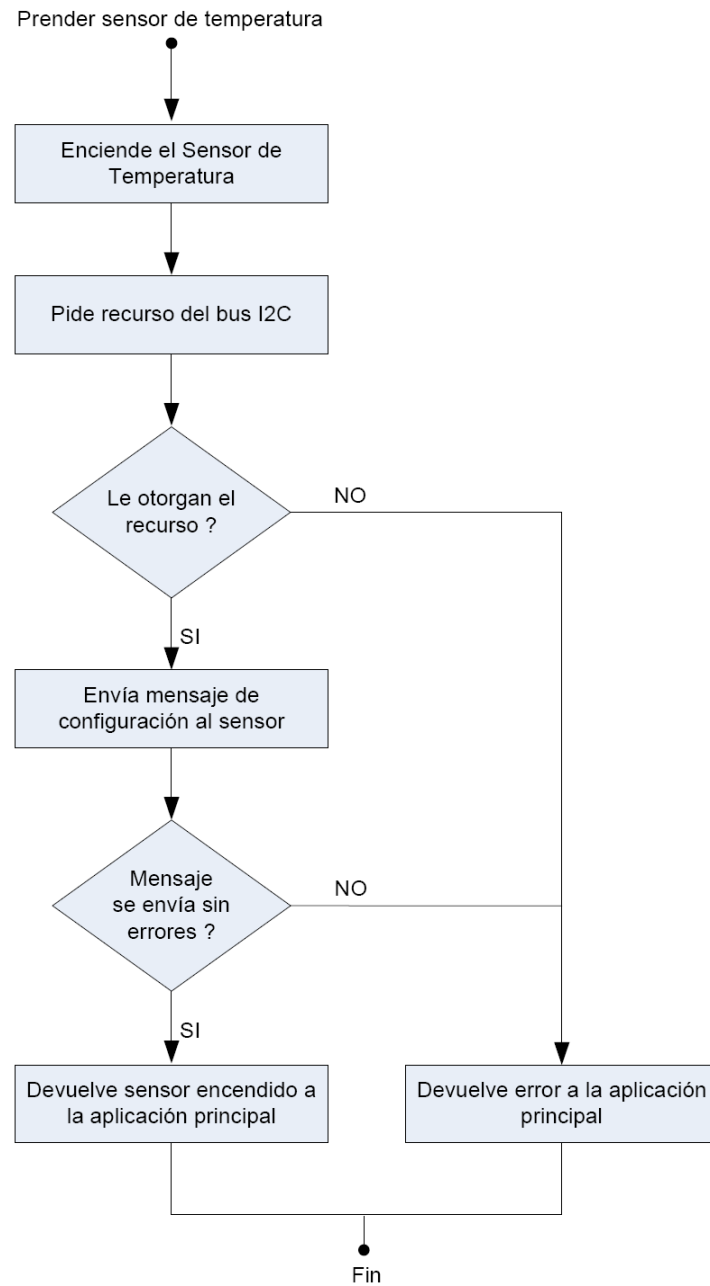
La interfaz *Resource* maneja el pedido del recurso, debido a que el bus I2C puede ser compartido por varios componentes corriendo en la misma aplicación. Por lo que antes de utilizar cualquiera de las funciones de la interfaz *I2CPacket* primero se debe pedir y esperar confirmación de que el recurso fue cedido.

#### Encendido:

En el inicio de la aplicación principal existe una llamada a la función de encendido del componente del sensor de temperatura. Esta rutina consiste el configurar el dispositivo. Para esto se le escribe en una dirección particular de memoria, pero esta última es volátil por lo que se necesita volver a realizar el procedimiento con cada encendido del nodo sensor.

Los parámetros de configuración son los siguientes:

- Resolución de temperatura: 0,25 °C
- Modo: “one shot”.



**Figura 3.3: Diagrama de flujo de la función de encendido del sensor de temperatura**

El resto de los parámetros de configuración con que cuenta el dispositivo no son de importancia porque son para funcionalidades que no se usan.

## Lectura

En la Figura 3.4 se muestra un diagrama de flujo de la función de lectura del sensor de temperatura-

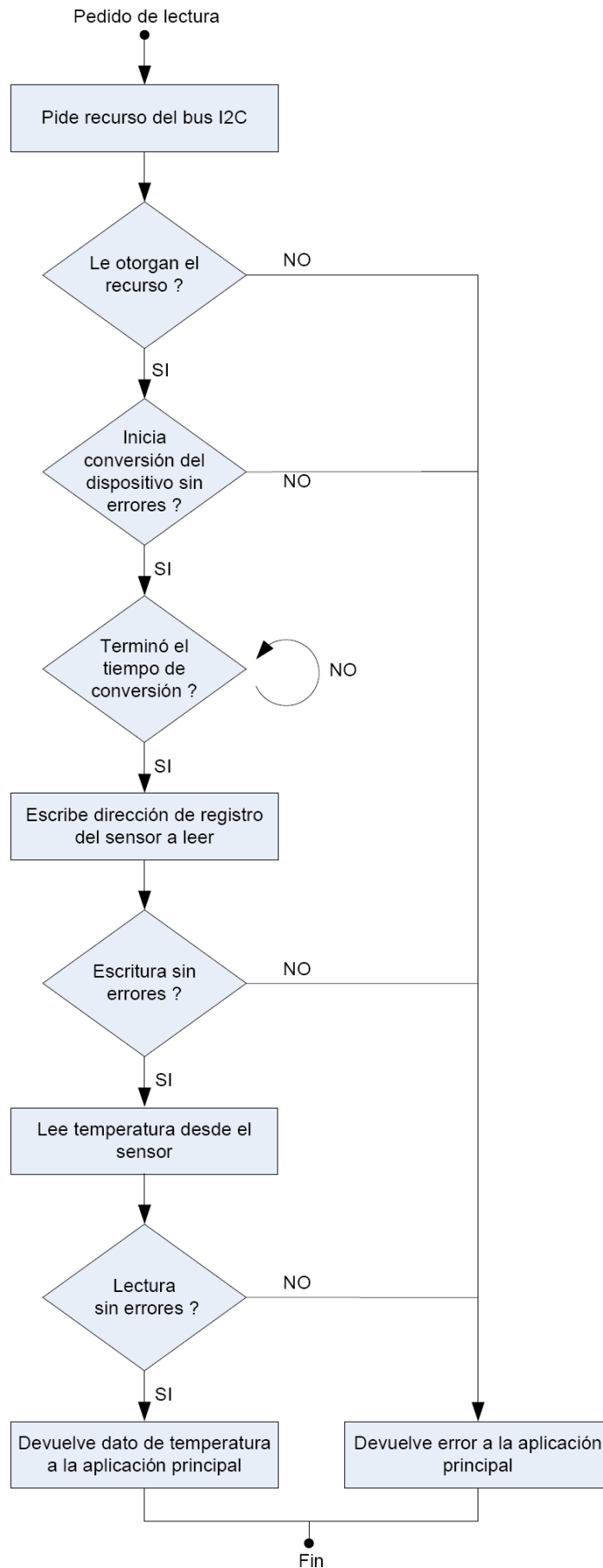


Figura 3.4: Diagrama de flujo de la función de lectura de dato de temperatura

La lectura de temperatura consta de 3 pasos, cada uno de ellos un mensaje I2C. Primero se envía una escritura al dispositivo con el comando de empezar la conversión. Debido al modo en que se configuró el dispositivo se necesita este paso antes de la lectura tener una medida actual de la temperatura. Luego se envía otra petición de lectura pero esta vez con la dirección de memoria de la cual se quiere leer. Esto le dice al dispositivo que en el siguiente pedido de lectura esta dirección es de la cual se requiere el dato. Por último se envía una petición de lectura y se espera el dato.

Si cualquiera de estas peticiones da error entonces el componente devuelve a la aplicación principal un mensaje de error.

Por encima de ese componente se programó un componente específico para el sensor con los parámetros de configuración en el encendido y direcciones de memoria usados de forma que solo se tiene una interfaz Read y otra SplitControl para manejar el sensor desde la aplicación. La primera devuelve la medida de temperatura mientras que la segunda inicia el componente y envía mensaje de configuración del DS620.

## 3.2. Sensor de pH

### 3.2.1 Definición de pH <sup>5 6</sup>

El pH se define como el logaritmo negativo de la actividad de los iones de hidrógeno en una sustancia:

$$pH = -\log_{10}(a_{H^+}) \quad (3.1)$$

Si la solución a medir es una solución básica o ácida no demasiado diluida (por encima de 0.001M) la actividad de los iones de hidrógeno se puede aproximar por la concentración de los mismos. En ese caso se tiene que:

$$pH = -\log_{10}[H]^+ \quad (3.2)$$

El valor de pH de una sustancia está relacionado con las concentraciones de iones de hidrógeno  $[H^+]$  y con las de los iones de hidroxilos  $[OH^-]$ . Si hay mayor concentración de iones de hidrógeno que de hidroxilos la sustancia es ácida, tiene iones de hidrógeno libres y su pH es menor que 7. Si se encuentran en equilibrio la sustancia es neutra y el pH es igual a 7. Por último si hay una mayor concentración de iones de hidroxilos que de hidrógeno la sustancia es básica, tiene iones de hidroxilos libres y su pH se encuentra entre 7 y 14.

### 3.2.2 Electrodo de vidrio

Para la medición de pH, el método convencional en el laboratorio y en la industria es a través del electrodo de vidrio, éste utiliza el mismo principio de

funcionamiento que una celda galvánica<sup>7</sup>.

Constructivamente el sensor consta de dos electrodos, uno de medida, que provee un potencial proporcional al logaritmo de la actividad ( $[H]^+$ ) del ion de hidrógeno en la muestra y otro de referencia, que provee un potencial constante y estable, independiente de la actividad de la muestra. Estos electrodos pueden estar en cápsulas independientes o en una misma cápsula, en este último caso se lo denomina electrodo de combinación (es el utilizado en este proyecto). En la Figura 3.5 se muestra un esquema del electrodo de combinación.

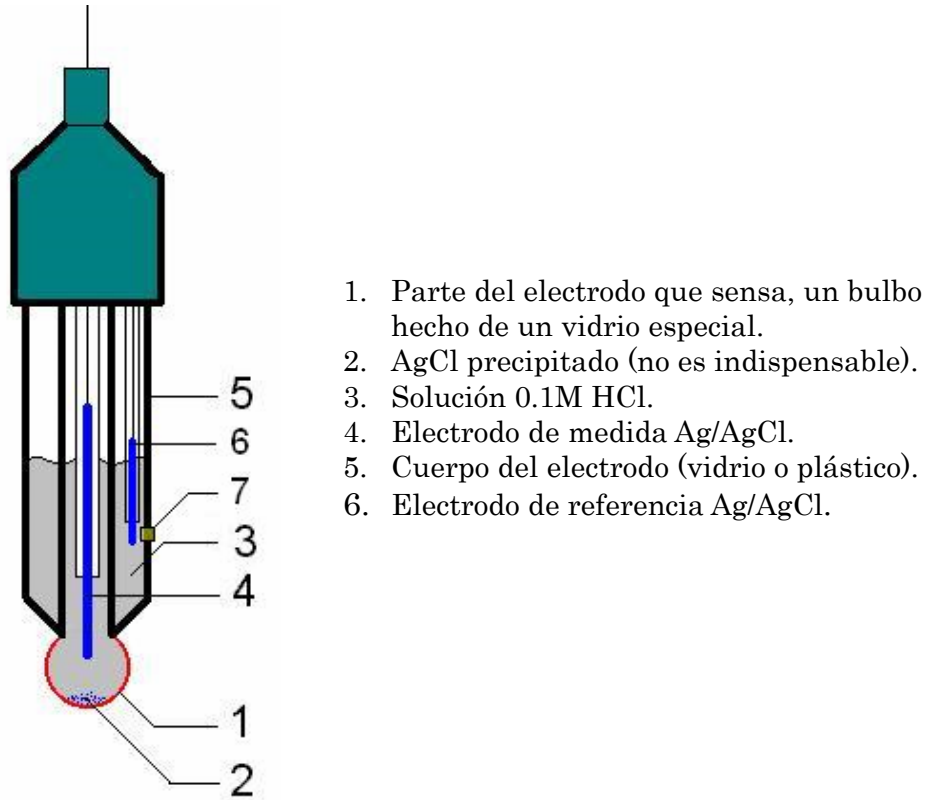


Figura 3.5: Esquema del electrodo de combinación<sup>8</sup>

## Relación Voltaje - pH

El vínculo entre el potencial que se desarrolla en bornes del electrodo y el pH de la solución está determinado por la ecuación de Nernst<sup>9</sup>:

$$E = E_0 + \frac{\ln(10) \cdot R \cdot T}{n \cdot F} \cdot \log(a_h) \quad (3.3)$$

E: potencial eléctrico del electrodo de medida

E<sub>0</sub>: potencial del electrodo de referencia, se define para obtener potencial 0 a pH=7.

R: Constante de los gases,  $R = 8.31439 \text{ J/mol.K}$

T: Temperatura en K

n: Valencia iónica (valencia del hidrógeno  $n=1$ )

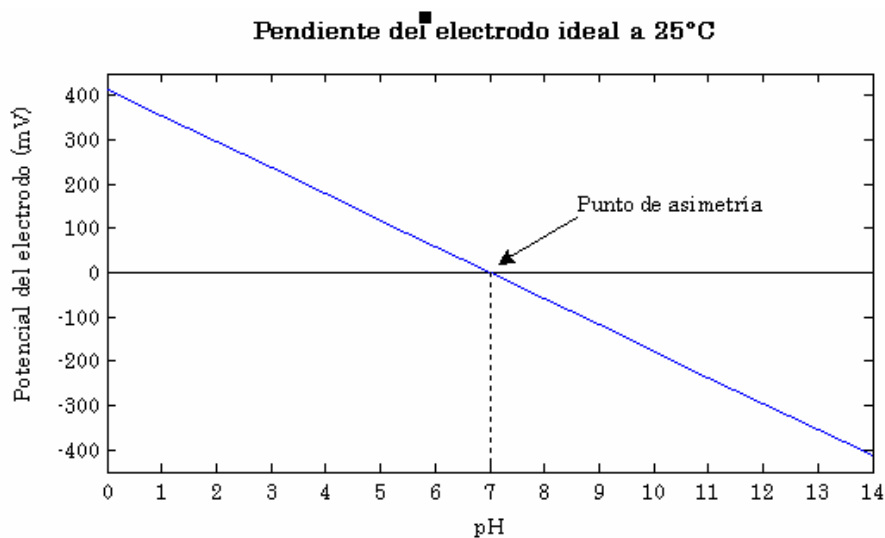
F: Constante de Faraday,  $F = 96493.1 \text{ C/mol}$

Para un electrodo ideal, la diferencia de potencial entre los electrodos de medida y referencia es de 0 mV para pH = 7, con lo que reescribiendo la ecuación de Nernst se obtiene<sup>10</sup>:

$$\Delta E = -0.1984 \cdot T \cdot (pH - 7) \quad (3.4)$$

$$\Delta E = -59.16 \cdot (pH - 7) @ 25^\circ C \quad (3.5)$$

En la práctica, la respuesta ideal de Nernst no corresponde con la real. Si el electrodo es nuevo y bajo ciertas consideraciones de limpieza y uso la respuesta se aproxima bastante a la ideal. Es por ese motivo que es necesaria la calibración periódica del mismo. Durante la calibración el pH-metro determina la pendiente actual y el punto de asimetría, con lo que obtiene una caracterización de la curva que se ve en la Figura 3.6.



**Figura 3.6: Pendiente del electrodo ideal a 25°C**

En la curva real se observa una desviación de algunos mV a pH=7, esta desviación es una medida de performance del electrodo de referencia. Si la desviación es mayor a  $\pm 30$  mV el electrodo de vidrio empieza a comportarse de forma inestable<sup>11</sup>, lo que puede generar errores en las medidas. Para que el usuario del sistema pueda usar la condición anterior como una medida de desempeño del electrodo, se recomienda agregar una lectura de voltaje entre bornes del electrodo, con el fin de que este sepa al insertar el electrodo en pH 7 si es necesaria una limpieza o no.

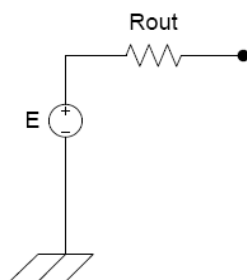
Otra característica que se presenta con el uso del electrodo, es la caída en la pendiente con respecto a la teórica de  $-59.16$  mV/pH, para un correcto funcionamiento, puede variar entre  $-55$  y  $-60$  mV/pH que se corresponde con un 95 y 102% de la teórica a 25°C.

## Modelo eléctrico

El modelo eléctrico de un sensor de pH basado en electrodo de vidrio consiste en fuente de voltaje continuo (proporcional al pH de la sustancia en la cual está inmerso) y una resistencia del orden de los cientos de M $\Omega$  (Figura 3.7).



El potencial ( $E$ ) lo modela la ecuación de Nernst, la resistencia ( $R_{out}$ ) modela muchas resistencias<sup>12</sup>, pero la preponderante está en el bulbo donde está la transferencia de iones de hidrógeno (resistencia de 1 en la Figura 3.5).



**Figura 3.7: Modelo eléctrico de un electrodo**

Una característica de la resistencia  $R_{out}$  es que aumenta al disminuir la temperatura, esto tiene un efecto en el tiempo de respuesta del electrodo. Sin embargo para las temperaturas de trabajo en este proyecto esto no es apreciable. Otras propiedades de la membrana también afectan la resistencia, como el espesor, el material utilizado, la forma, etc.

Los fabricantes informan un tiempo de respuesta de 20s ante cambios abruptos de 7 a 4 en el pH de la solución medida<sup>13</sup>.

## Calibración

Una solución buffer es una solución de pH conocido, comercialmente se venden con los valores 4, 7 y 10 de pH e incertidumbre del orden  $\pm 0.02$  pH. Dependiendo de la cantidad de soluciones buffer que se tengan a disposición, se tienen dos métodos para la calibración del sensor.

El primer método se llama “calibración de un punto”, utiliza una solución buffer de pH=7, con la cual se determina la deriva del electrodo respecto a los 0 mV. Luego, se calculan los demás valores de pH utilizando la pendiente teórica, que depende de ecuación de Nernst y de la temperatura (ecuación 3.4).

El segundo método se llama “calibración de dos puntos” y será el utilizado en este proyecto. Para realizarlo se necesitan dos soluciones buffers, una de ellos debe ser pH=7 y la otra pH=4 o pH=10, según las sustancias a medir. Como en este proyecto se miden soluciones en el rango de 3 a 7 pH se calibrará con soluciones buffers de 4 y 7. Este método permite, a partir de los dos puntos medidos (en soluciones buffer 4 y 7) hallar la pendiente y el punto de asimetría de la curva de pH.

### 3.2.3 Electrodo de pH de estado sólido (ISFET)

El electrodo ISFET<sup>6</sup> se utiliza para medir la concentración de iones en una solución, particularmente se puede utilizar para medir la concentración de iones de

---

<sup>6</sup> ISFET denominado por sus siglas en inglés Ion Sensitive Field Effect Transistor

hidrógeno, y por tanto medir el pH de la solución. Cuando la concentración de iones cambia en la solución, la corriente a través del transistor cambia de forma acorde. Este electrodo es muy similar en construcción y funcionamiento a los transistores MOSFET. (Por más información sobre los electrodos ISFETs ver Anexo I)

### 3.2.4 Elección del electrodo

#### Elección de la tecnología

Para la aplicación es deseable que el electrodo de pH cumpla con las siguientes características:

- *Bajo consumo:* si bien se preñerá el integrado de sensado de pH solo cuando sea necesario es deseable que el electrodo consuma lo menos posible cuando esté energizado. El consumo de un electrodo depende principalmente de la tecnología del mismo.
- *Medida rápida y confiable:* el bajo consumo depende tanto de la corriente como del tiempo que se tiene energizado al integrado: cuanto más rápido se tome la medida, mejor. Asimismo, la medida debe ser confiable.
- *Soportar las sustancias que se encuentran dentro del fulón:* las sustancias dentro del fulón son altamente corrosivas y el electrodo estará en contacto con las mismas durante todo el proceso.
- *Adaptable al Mote:* deberá tener una salida que pueda ser inyectada en una de las entradas analógicas que tiene el Mote.
- *Bajo costo de mantenimiento:* deberá de tener un costo de mantenimiento lo suficientemente bajo para que sea redituable utilizar el mismo.

Las opciones que se consideraron fueron dos: electrodos de vidrio y electrodos ISFET.

Las principales ventajas de los electrodos ISFET son su rapidez al tomar medidas y su bajo costo de mantenimiento debido a que no necesitan calibración y limpieza como sí lo requieren los electrodos de vidrio.

Por otro lado, los electrodos ISFETs no ofrecen la misma estabilidad y precisión que los electrodos de vidrio. Debe evitarse su uso en sustancias abrasivas y evitando ciertas sustancias químicas. También deben evitarse descargas eléctricas o la estática, ya que el electrodo se puede dañar<sup>14</sup>. También tiene un consumo mayor al electrodo de vidrio debido a su principio de funcionamiento, ya que requiere que se le suministre corriente para poder funcionar a diferencia del electrodo de vidrio. Aunque, como contrapartida, le toma menos tiempo tener una medida con una precisión dada.

Otra desventaja del electrodo ISFET es su costo, que es considerablemente superior al de los electrodos de vidrio. A pesar de que los electrodos ISFETs fueron

desarrollados en la década de los 70s, su aplicación comercial recién se está desarrollando en los últimos años. Es por esto que se encuentran mucho menos desarrollados que los electrodos de vidrio, los cuales son considerados un estándar en el ámbito industrial

Finalmente, se decidió utilizar un electrodo de vidrio debido a su menor costo y a su amplio desarrollo en el ámbito industrial

### Elección del modelo del electrodo

Para la elección del electrodo, se consultó al cliente que electrodos utilizaban ellos realizar sus mediciones, suponiendo que los mismos serían los más adecuados para las sustancias a relevar. El cliente informó que principalmente utilizan dos marcas, Oakton y Mettler-Toledo y dentro de estas utilizan electrodos de laboratorio (Ag/AgCl) para realizar sus medidas manuales. Debido a que los mismos son de laboratorio no sería posible montar los mismos en los fulones principalmente por dos razones. Primero, porque son electrodos direccionales que no permiten medir en cualquier orientación (solo vertical) y por otro lado los electrodos de laboratorio no presentan rosca como para fijarlos en los fulones.

Para ahorrar tiempo se decidió utilizar un electrodo prestado por el cliente para luego adquirir uno industrial de características similares que si pueda ser montado al fulón. El modelo de electrodo suministrado por el cliente fue el InLab 413 de la marca Mettler Toledo. En la Figura 3.8 se muestra una foto del electrodo de laboratorio que se utilizó.



Figura 3.8: Electrodo Mettler Toledo InLab 413

Cabe destacar que por un tema de tiempos no fue posible montar el sensor de pH en el fulón y por ende no se terminó utilizando el electrodo industrial. De haberse podido montar, se podría haber adquirido el modelo de electrodo industrial PHE-5316-10 de la marca Omega que tiene características muy similares al de laboratorio utilizado.

### 3.2.5 Circuito de medición de pH

El rango dinámico de las soluciones a medir se encuentra entre 0 y 300mV, que corresponden a un pH entre 7 y 2 (de la ecuación 3.5 se deduce que hay aproximadamente 60 mV / pH a 25°C), debemos amplificar la señal eléctrica para aumentar la resolución y disminuir el error en la medida.

En esta sección se explicará la elección de los componentes para el circuito de medición de pH

## Amplificador

Para la etapa amplificadora se optó por un amplificador de instrumentación (AI) que debía cumplir los requerimientos detallados a continuación:

- *Bajo consumo:* el bajo consumo es indispensable al trabajar con pilas, puesto que la autonomía del sensor depende de ello.
- *Alimentación:* como el Mote funciona en el rango 1,8...3,6V, es necesario que todos los circuitos integrados que se utilicen trabajen en ese rango. La alimentación deberá ser asimétrica porque la entrada ADC del Mote no trabaja con valores de tensión negativos.
- *Muy alta impedancia de entrada:* debido a la alta impedancia de salida del electrodo de medida de pH, del orden de los cientos M $\Omega$ , es necesario que la impedancia de entrada del amplificador de instrumentación sea mucho mayor a ese valor.
- *Ganancia:* para obtener la mejor resolución posible, con el menor error en la medida, la ganancia óptima es de 5 V/V, esto se debe a que el rango dinámico a la entrada es de 300mV y a la salida es de 1800mV.

Una vez establecidos los requerimientos para el AI, se buscó entre los fabricantes de integrados para ver cuál era el que mejor se adaptaba, optándose por el INA333 de Texas Instruments (TI), a pesar que habían otras opciones, el que tenía menor tensión de alimentación era este.

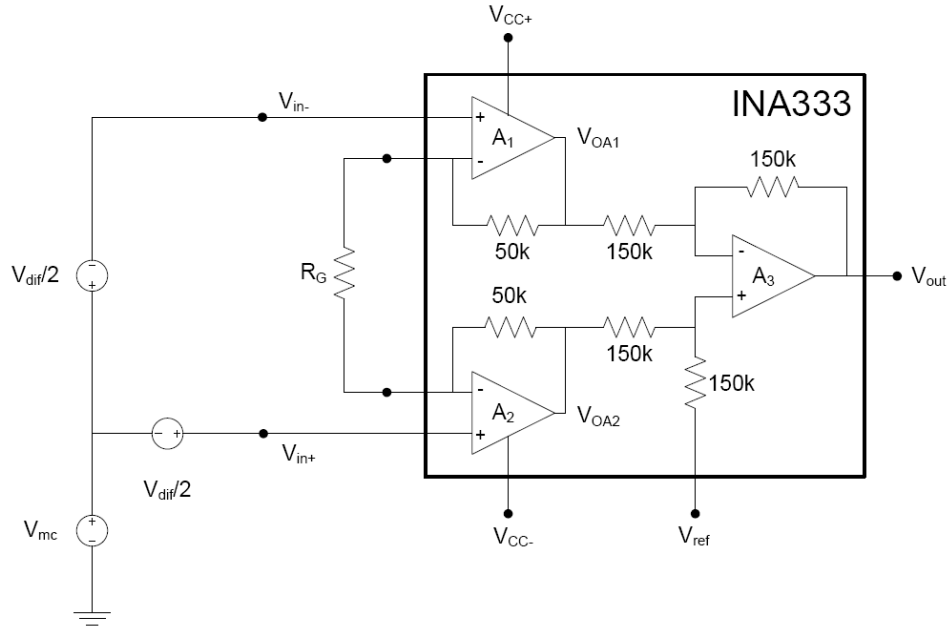
<b>Características del amplificador de instrumentación INA333</b>	
Rango de alimentación	1.8 ... 5.5 V
Impedancia de entrada	100 G $\Omega$
Ganancia	1 ... 1000 V/V
Encapsulado	MSOP-8
Consumo	50 $\mu$ A

**Tabla 3.6 Características del amplificador de instrumentación INA333**

También se podría haber implementado el amplificador por medio de Amplificadores Operacionales (AO) individuales, logrando reducir un poco el consumo y aumentar la impedancia de entrada, pero con la complejidad adicional de tener que balancear las resistencias para lograr el máximo rechazo al modo común.

## Consideraciones para la integración del amplificador

En la Figura 3.9 podemos ver cómo está realizado el AI y cuáles son las condiciones necesarias para que funcione correctamente, esto implica respetar los rangos de entrada en modo común para no saturar ninguno de los AO internos del mismo.



**Figura 3.9: Esquema del amplificador operacional INA333**

A continuación se muestran la tensión de puntos claves del circuito

$$V_{in+} = V_{mc} + \frac{V_{dif}}{2} \quad \text{y} \quad V_{in-} = V_{mc} - \frac{V_{dif}}{2} \quad (3.6)$$

$$V_{OA1} = V_{mc} - \frac{V_{dif}}{2} \cdot \left(1 + \frac{100k}{R_G}\right) \quad \text{y} \quad V_{OA2} = V_{mc} + \frac{V_{dif}}{2} \cdot \left(1 + \frac{100k}{R_G}\right) \quad (3.7)$$

$$V_{out} = V_{ref} + G \cdot V_{dif} = V_{ref} + (V_{in+} - V_{in-}) \cdot \left(1 + \frac{100k}{R_G}\right) \quad (3.8)$$

Para no saturar los amplificadores A1 y A2 es necesario:

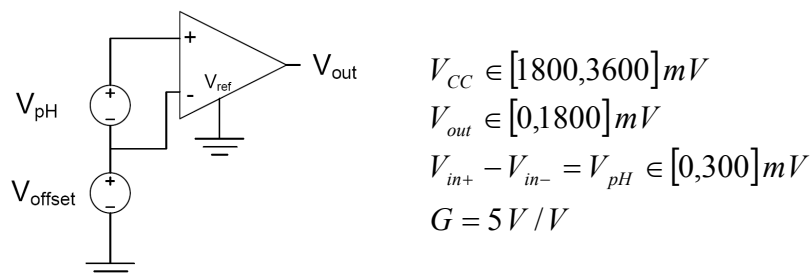
$$0 < V_{OA1} < V_{cc} \quad \text{y} \quad 0 < V_{OA2} < V_{cc} \quad (3.9)$$

$$0 < V_{cm} - G \cdot \frac{V_{dif}}{2} < V_{cc} \quad \text{y} \quad 0 < V_{cm} + G \cdot \frac{V_{dif}}{2} < V_{cc} \quad (3.10)$$

Sustituyendo y dejando las condiciones restrictivas:

$$(I) \quad 0 < V_{cm} - \frac{V_{out}}{2} \quad \text{y} \quad (II) \quad V_{cm} + \frac{V_{out}}{2} < V_{cc_{min}} \quad \text{considerando } V_{ref} = 0 \quad (3.11)$$

Entonces, para estimar el modo común necesario para no saturar, podemos aproximar:



**Figura 3.10: Modelo para estimar el modo común**

Del esquema de la Figura 3.10 se deduce:

$$V_{mc} = V_{offset} + \frac{V_{pH}}{2} \quad (3.12)$$

Con las ecuaciones anteriores es posible determinar el Voffset para que el amplificador funcione correctamente. El rango de la pila lo determina el rango de funcionamiento del Mote y está indicado por Vcc. Como el AI está alimentado por Vcc que varía con la carga de la pila, se decidió amplificar un poco por debajo del Vcc mínimo (1800 mV). Para lograr la ganancia de 5 en el AI, se colocó un potenciómetro de 100 kΩ (formado por R<sub>A</sub> + R<sub>p</sub> + R<sub>B</sub> en la Figura 3.11, siendo R<sub>p</sub> la resistencia variable), para poder poner 25 kΩ en R<sub>g</sub>.

Cálculo del Voffset mínimo, sustituyendo 3.20 en 3.19.I:

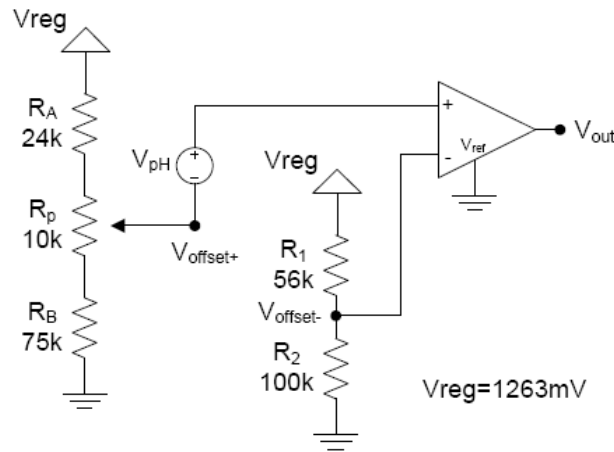
$$\left. \begin{array}{l} V_{pH} = 300mV \\ V_{out} = 1800mV \end{array} \right\} \Rightarrow V_{offset} > 750mV \quad (3.13)$$

Cálculo del Voffset máximo sustituyendo 3.20 en 3.19.II:

$$\left. \begin{array}{l} V_{pH} = 300mV \\ V_{out} = 1800mV \\ V_{cc} = 2000mV \end{array} \right\} \Rightarrow V_{offset} < 950mV \quad (3.14)$$

A partir del rango de variación del offset, se obtuvieron los valores de los componentes del circuito (Figura 3.11)

Como se puede apreciar en la Figura 3.11, se colocaron referencias de distinto valores en la pata inversora y no inversora, siempre respetando la restricción en el rango de entrada en modo común.



**Figura 3.11: Circuito diseñado para la medición de pH**

*Divisor resistivo para V<sub>offset-</sub>:*

El nivel de tensión en  $V_{in}^-$  ( $V_{offset}^-$ ), se impone a través del divisor resistivo y de la referencia de voltaje (impuesta por el integrado MAX6018 de tensión 1263mV). Este nivel se calcula para que el modo común quede en el rango de trabajo (en particular para un  $V_{offset}^- = 800mV$ ). Debido a que la resistencia vista hacia el amplificador es del orden de los  $G\Omega$ , este no toma corriente, entonces con los valores de la Figura 3.11, se obtiene:

$$V_{offset-} = 810 mV \quad \text{y} \quad \text{Consumo} = 8 \mu A$$

*Divisor resistivo para V<sub>offset+</sub>:*

El nivel de tensión en  $V_{offset+}$  tiene dos propósitos, el primero es elevar el modo común del amplificador hasta un nivel adecuado y el segundo es compensar la deriva temporal del electrodo de medida de pH. Es por el segundo propósito que se utiliza un potenciómetro digital (MCP4652 de Microchip), de modo de modificar el offset a medida que sea necesario.

Como el circuito no puede manejar niveles negativos de tensiones y como  $V_{pH} \leq 0$  para  $pH \geq 7$ , hay que garantizar que  $V_{in}^+ \geq V_{in}^-$ . Es por este motivo que se impone un  $V_{offset+} \geq 850 mV$ , para asegurar no inyectarle nunca una entrada diferencial negativa al amplificador.

Resolviendo el divisor resistivo se tiene:

$$V_{offset+} \in [869,985] mV \quad \text{y} \quad \text{Consumo} = 11 \mu A$$

Donde los límites de  $V_{offset+}$  se dan en los extremos del potenciómetro digital.

Como se puede ver de los valores anteriores, hay una leve diferencia entre los potenciales de “offsets”. Esto tiene dos explicaciones, una es que el amplificador de instrumentación no presenta un comportamiento perfectamente lineal para valores de salida cercanos a 0V y la otra es que ese desbalance permite darle un

poco más de vida útil al electrodo a medida que la deriva del mismo aumenta.

Otra propiedad del amplificador que también influyó a la hora de colocar “offsets” en los pines de entrada es que, por requerimiento del mismo, es necesario que ambas patas de entrada del amplificador tengan una tensión de al menos 100 mV.

#### *Verificación de las restricciones impuestas:*

A partir de las tensiones de offset se puede calcular las tensiones de entrada del amplificador, obteniéndose:

$$V_{in-} = 810 \text{ mV} \quad \text{y} \quad V_{in+} \in [870, 1170] \text{ mV}$$

$$V_{mc} \in [840, 990] \text{ mV}$$

Por lo que considerando que la ganancia del amplificador es 5V/V, se deduce que:

$$V_{out} \in [300, 1800] \text{ mV}$$

A continuación se verifican las inecuaciones de 3.19:

$$\begin{aligned} 0 < 840 \text{ mV} - 150 \text{ mV} &= 690 \text{ mV} \\ &\text{y} \\ 990 \text{ mV} + 900 \text{ mV} &= 1890 \text{ mV} < 2000 \text{ mV} \end{aligned} \quad (3.15)$$

#### *Estimación del uso máximo del electrodo:*

Como tenemos 175mV máximos de diferencia entre los offsets de las patas positiva y negativa, se puede estimar el uso máximo que se le puede dar al electrodo de medición de pH, esto es cuando la deriva del mismo alcanza los 175mV anteriores. Suponiendo una deriva de 3mV/semana<sup>15</sup>, se concluye que la vida útil del electrodo sería de 58 semanas.

Por otro lado la mayoría de los fabricantes recomiendan cambiar el electrodo si la deriva de este es mayor o igual a 30 mV, con lo la vida útil del este estará determinada por el propio electrodo y no por la circuitería diseñada.

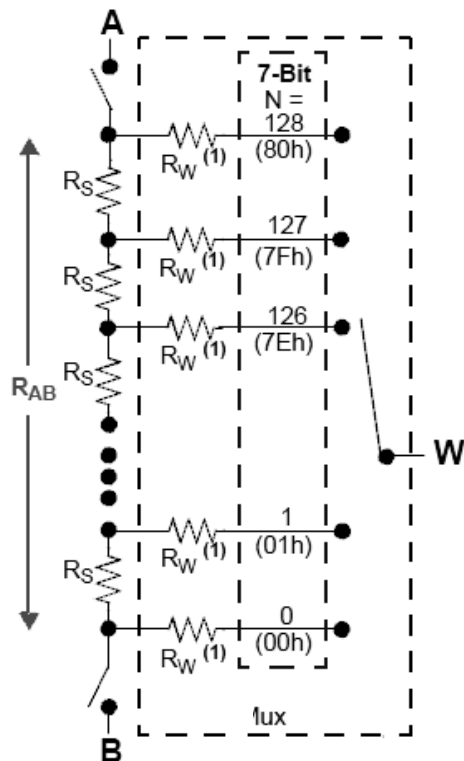
## **Potenciómetro digital**

Como se explicó anteriormente, los electrodos de vidrio para la medida de pH presentan una deriva en función del tiempo de uso. Esta deriva se traduce en una disminución del potencial del electrodo a un pH=7, lo que genera un potencial negativo para ese valor de pH. La entrada analógica del Mote no permite medir valores negativos, porque la alimentación es asimétrica. Para compensar esta caída de tensión, se hace necesario tener una manera automática de aumentar la referencia del electrodo de pH.

Para resolver este problema se decidió utilizar un potenciómetro digital.



Este consiste en un “array” de resistencias, cuyo conexionado se puede configurar digitalmente, como se puede ver en la Figura 3.12. Dependiendo de donde se cierra el circuito (en que punto del array se coloca la llave de W) es el valor de resistencia que se obtiene.



$R_{AB}$  es la resistencia total del potenciómetro (para el modelo elegido  $R_{AB} = 10k$ ).

W es la pata del medio por donde se obtiene las resistencias variables.

$$R_S = \frac{R_{AB}}{127}$$

$$R_{WB} = \frac{R_{AB}N}{128} + R_W \text{ siendo } N \text{ un entero entre } 0 \text{ y } 128$$

$R_W$  es la resistencia del “Wiper” y depende de muchos factores incluyendo el nivel de alimentación, la tensión en los terminales y la temperatura

Figura 3.12: Esquema interno de un potenciómetro digital de 127 pasos<sup>16</sup>

Los requerimientos que debe cumplir el potenciómetro digital para poder adaptarse al circuito son los siguientes:

- *Bajo consumo*
- *Alimentación* en el rango de 1.8 a 3.0 V
- *$R_{AB}$  suficientemente grande* como para tener poca corriente
- *Configurable a través del protocolo I2C*

Luego de determinados los requerimientos, se seleccionó el integrado que mejor cumple con ellos: MCP4652 de la firma Microchip que se muestra en la Figura 3.13. De la cantidad de variantes del mismo se seleccionó un potenciómetro de  $R_{AB} = 10k\Omega$ .

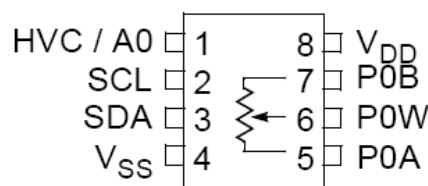


Figura 3.13: Integrado MCP4652 de Microchip

El consumo varía según haya comunicación serial o no:

Con comunicación:  $I=650\mu A$  (cuando se está configurando).

Sin comunicación:  $I=5\mu A$ .

Como la comunicación serial se da solamente en la calibración y durante unos instantes, se observa que la mayor parte del tiempo el consumo es despreciable.

Puesto que el bus de comunicación I2C ya estaba implementado para la medida de temperatura, se resolvió utilizar el mismo bus para comunicarse con el potenciómetro digital.

El modelo seleccionado solo permitía colocar hasta dos potenciómetros a manejar por el Bus I2C. La dirección I2C con el pin I2C bajo es: 0101110 binario, el ultimo cero corresponde a A0.

Como se mencionó anteriormente, para obtener un modo común que permita el correcto funcionamiento del amplificador de instrumentación era necesario colocar resistencias adicionales al potenciómetro digital, porque con solo  $10\text{ k}\Omega$  no se obtenía el offset necesario. Colocando las resistencias  $R_A$  y  $R_B$  (Figura 3.11) se movió el rango en el cual se puede variar el  $V_{\text{offset+}}$ .

$$V_{\text{offset+}} \in [869,985]mV \quad \text{y} \quad I = 11.4\mu A$$

Del consumo se observa que la resistencia de  $10\text{ k}\Omega$  es una buena elección.

## Voltaje de referencia

La tensión de alimentación del circuito disminuye con el tiempo debido a que está alimentado por pilas. Tanto el amplificador como el electrodo necesitan un potencial de offset constante (que no dependa de la caída de voltaje de la pila por el uso de la misma). Por esto se hizo necesario obtener un potencial de referencia constante, se buscaron varios métodos para realizarlo.

El primer método investigado fue en base a integrados comerciales. Los requerimientos para estos integrados fueron:

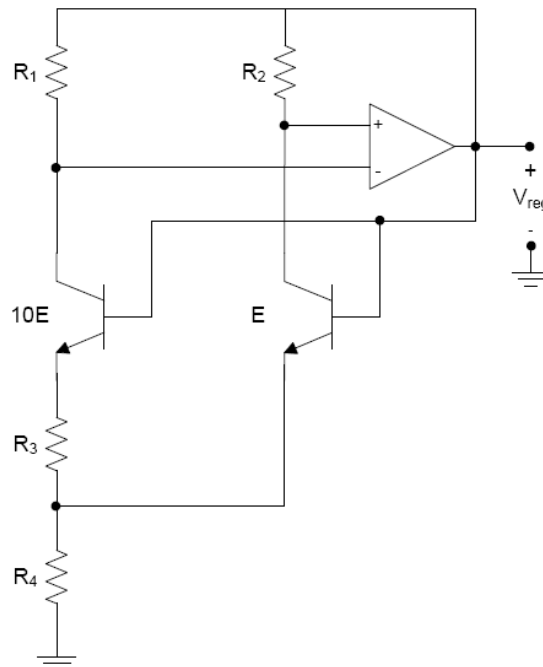
- Configurables a través del puerto I2C
- Rango de alimentación entre 1.8 y 3.0 V
- Bajo consumo
- Salida en el rango de los offsets requeridos por el amplificador.

Si se hubiera encontrado un integrado con estas características se hubiera podido prescindir del potenciómetro digital y el divisor resistivo. Pero ninguno de los integrados encontrados las cumplían.

Otro método considerado para obtener una tensión de alimentación estable

fue utilizar un integrado que otorgara un voltaje de referencia lo suficientemente exacto, para luego con un divisor resistivo lograr los offsets necesarios. Este fue el método utilizado ya que se pudieron encontrar integrados adaptables al circuito.

Para este diseño se utilizó el integrado MAX6018 de la firma Maxim. En la Figura 3.14 se muestra el esquema interno del mismo.



**Figura 3.14: Esquema interno del integrado de voltaje de referencia MAX6018**

El integrado de voltaje de referencia a partir del potencial que le entrega la batería otorga un voltaje estable de  $1.263V \pm 160\mu V$ , con un consumo cercano a los  $5\mu A$ , logrando una corriente de salida de hasta  $1mA$ , muy superior a la requerida para el diseño. El encapsulado de este integrado es SOT23.

Las condiciones que impone el integrado, a través de la hoja de datos, es de poner a la entrada y a la salida del mismo, capacitores de  $0.1\mu F$ .

Otra motivo para colocar el integrado de referencia de voltaje, se debe a que el Mote no cuenta con una forma de colocar un potencial de referencia para el fondo de escala fijo en la entrada analógica del Mote. Esto provoca que las medidas varíen a medida que lo hace la pila, porque los pasos de cuantización van disminuyendo con esta. Es por esto que se hace imprescindible medir con el Mote un potencial constante y conocido.

Como el Mote tiene un conversor A/D de 10 bits para medir señales analógicas, es posible describir su funcionamiento a partir de la siguiente ecuación:

$$V_{in} = \frac{N_{in} \cdot V_{FS}}{N_{FS}} \quad (3.16)$$

$V_{in}$ : Señal analógica a medir.

$N_{in}$ : Valor cuantificado de  $V_{in}$ .

$V_{FS}$ : Valor del potencial de referencia del conversor (FS: fondo de escala).

$N_{FS}$ : Valor cuantificado del fondo de escala,  $N_{FS}=2^{10}=1024$ .

Aplicando lo anterior, para  $V_{in} = V_{ref} = 1.263V$  y un fondo de escala basado en la pila, se obtiene la siguiente relación:

$$V_{ref} = \frac{N_{ref} \cdot V_{bat}}{N_{bat}} \quad (3.17)$$

Este valor de  $N_{ref}$  se obtiene midiendo el canal 1 de la entrada analógica del Mote.

Suponiendo ahora un valor de entrada genérico  $V$ , para un valor de cuantificación  $N$  (medido en el canal 0 del Mote), se llega a:

$$V = \frac{N \cdot V_{bat}}{N_{bat}} \quad (3.18)$$

Combinando las ecuaciones de ambos canales se logra independizarse del voltaje de la pila, como se muestra a continuación:

$$V = \frac{N \cdot V_{ref}}{N_{ref}} \quad (3.19)$$

Hay una segunda opción para determinar el voltaje de la pila y por ende lograr independizarse de la caída del fondo de escala, esto es a través del voltaje de band gap ( $V_{bg}$ ) del microprocesador. Este, al igual que el voltaje de referencia del circuito de medición de pH, entrega un potencial constante. El único problema que presenta es que el fabricante garantiza que  $1.0V \leq V_{bg} \leq 1.2V$ , con  $V_{bg} = 1.1V$  típico. Por este motivo este método no resulta conveniente para tener una referencia para la entrada analógica, ya que la incertidumbre de la referencia resulta demasiado grande. En cambio este voltaje que brinda el microprocesador si es utilizado para determinar el voltaje de la batería que se muestra en pantalla en la aplicación de usuario, (como se verá en la sección 4.2). Esto es así porque el usuario no necesita tener gran precisión para saber cuando se gastó la batería y necesita cambiarla.

## Interruptor digital

Por la necesidad de consumir la menor corriente de las pilas posible y como la aplicación no requiere una medida continua de pH, se optó por implementar un mecanismo de apagado del circuito de medida de pH.

Para apagar el sensor la mejor opción que se encontró fue a través de una llave digital con transistores MOS, que corte la alimentación de todo el circuito de medida. Al igual que con los demás dispositivos, se buscó un integrado comercial que cumpla con los requerimientos de consumo y de alimentación.

La gran mayoría de los fabricantes cuentan con interruptores digitales de

prestaciones muy parecidas. Se eligió el TS3A4742 de Texas Instruments. El integrado cuenta con dos interruptores analógicos independientes normalmente abiertos (NA) de los cuales solamente se utiliza uno.

Un esquema de la llave digital se muestra en la Figura 3.15 y en la Tabla 3.7 se muestran las principales características de la misma. (Para más información ver Anexo III)

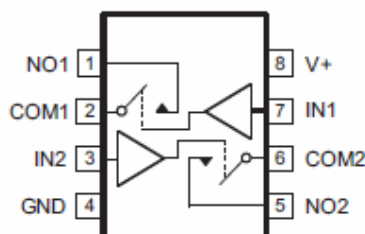


Figura 3.15: Integrado TS3A4742 de Texas Instruments<sup>17</sup>

Características del interruptor TS3A4742	
Rango de alimentación	1.8 ... 3.6 V
Consumo	0.5 $\mu$ A
R <sub>ON</sub> máxima	1.5 $\Omega$
I <sub>out</sub> máxima	100 mA
On time	14ns
Off time	9 ns
Encapsulado	MSOP-8

Tabla 3.7: Características del interruptor TS3A4742

### Filtro pasabajos de salida

Como la señal de pH es una señal continua, de baja amplitud, se hace necesario filtrar todo lo que sea señal en frecuencia, principalmente el ruido de la línea de 220V a 50 Hz. Debido a que el conversor A/D del Mote es de 10 bits, se obtiene un paso aproximado de 3.0mV cuando las pilas son nuevas. Por lo que puede tenerse errores de cuantización de por lo menos 1.5mV. Esto es despreciable frente al ruido del orden de los 100mV antes de pasar por el filtro, observado en el osciloscopio.

Para el diseño del filtro se eligió un filtro pasabajos RC de primer orden. Se diseñó el filtro con  $R = 10 \text{ k}\Omega$  y  $C = 4 \text{ }\mu\text{F}$ , con lo que el polo está en  $f = 4.0 \text{ Hz}$  y el tiempo de subida de 0 a 90% es menor a un segundo. (Por más información ver Anexo IV)

### Diseño final del circuito para medir pH

Una vez estudiados y seleccionados todos los integrados que componen el circuito de adaptación de la señal de pH se obtiene el esquemático que se presenta en la Figura 3.16.

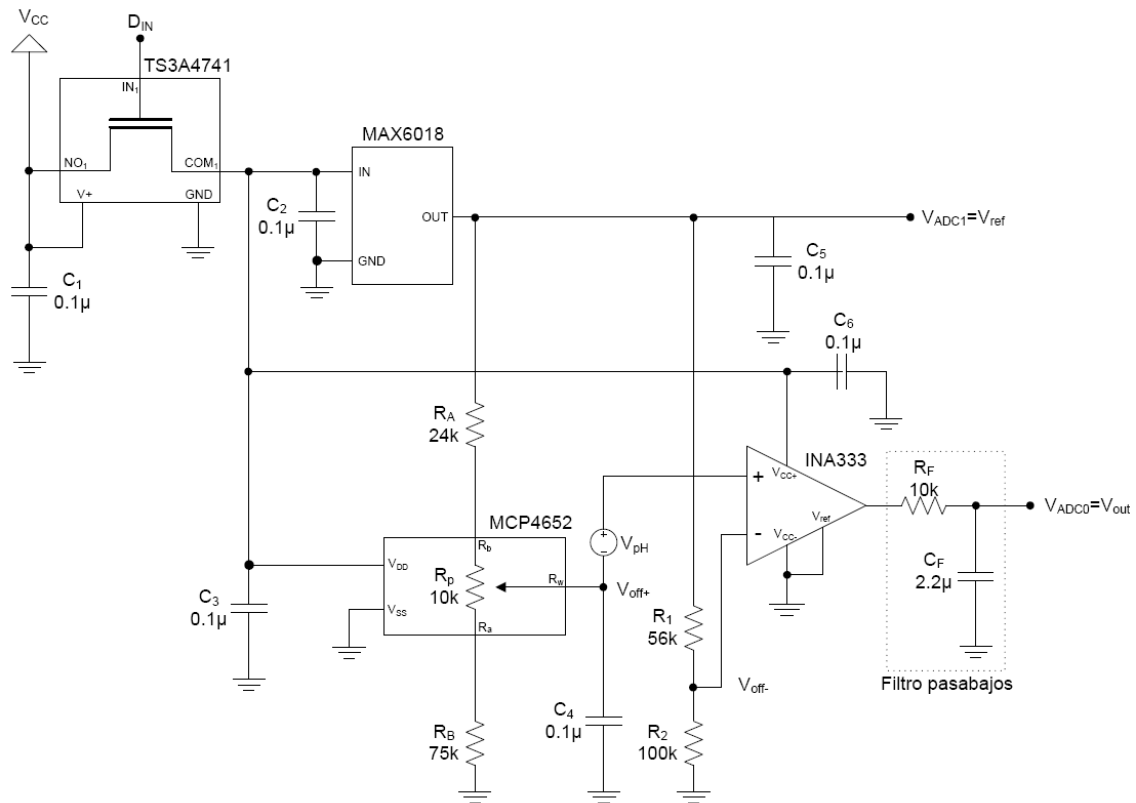


Figura 3.16: Esquema del circuito del sensor de pH

### 3.2.6 Programación del Mote para integrar sensor de pH

Se implementa un componente *PhimetroC* que provee a la aplicación de 3 interfaces: *Read*, *StdControl* y *SplitControl*. A través de éstas, la aplicación principal puede manejar las funcionalidades del sensor de pH.

- *StdControl*: Implementa las funciones para el encendido y el apagado del componente y todos los subcomponentes.
- *SplitControl*: Implementa las funciones para el ajuste del potenciómetro digital durante la calibración.
- *Read*: Implementa la lectura de la medida de pH, devuelve el voltaje en la entrada del conversor A/D en mV.

#### Encendido

En el encendido el microprocesador lee de la memoria flash la posición de potenciómetro digital. De esta forma no se pierde la información guardada en el último proceso de calibración.

#### Lectura

En la Figura 3.17 se muestra un diagrama de bloques de la función de

## lectura del sensor de pH

El tiempo de asentamiento de la lectura de pH es el tiempo que le lleva al circuito sensor entregar una medida estable cuando es energizado. Este parámetro debe utilizarse en la programación de la lectura de pH y se determina experimentalmente en la sección 6.2.1.

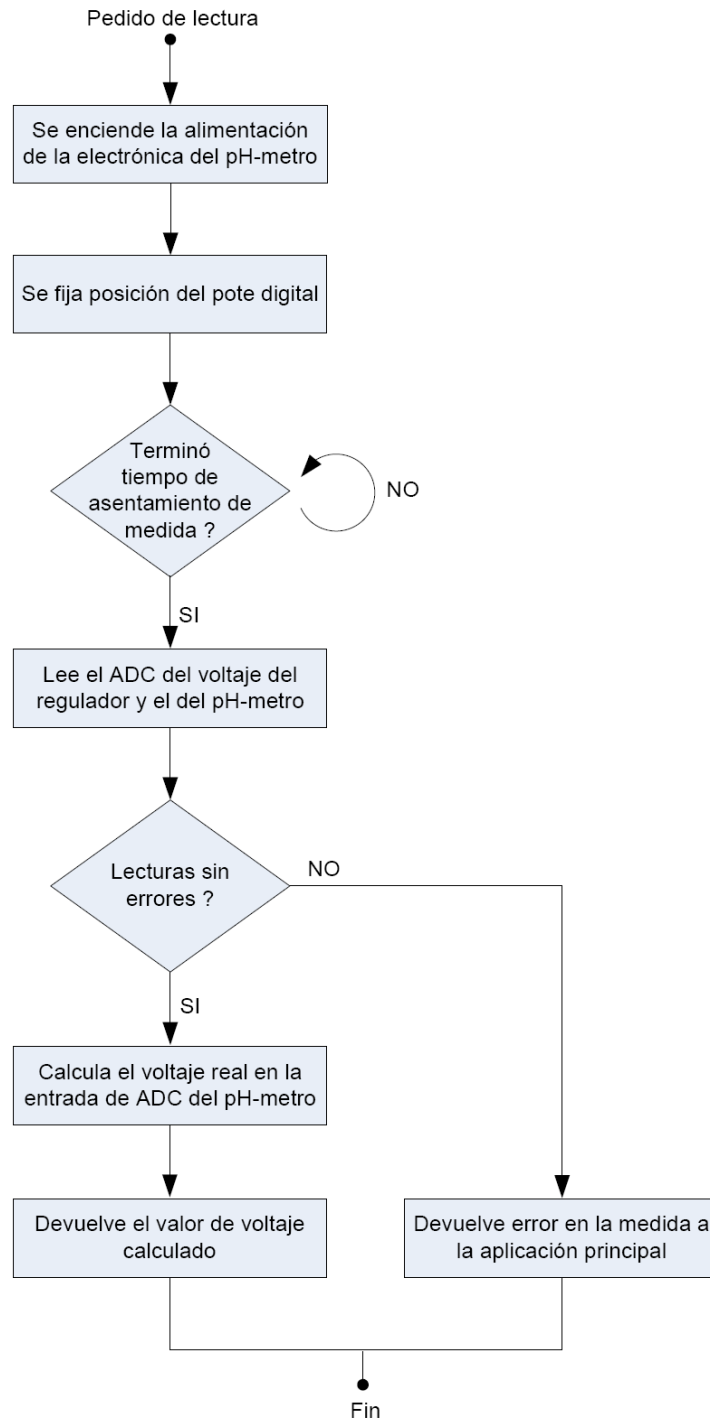
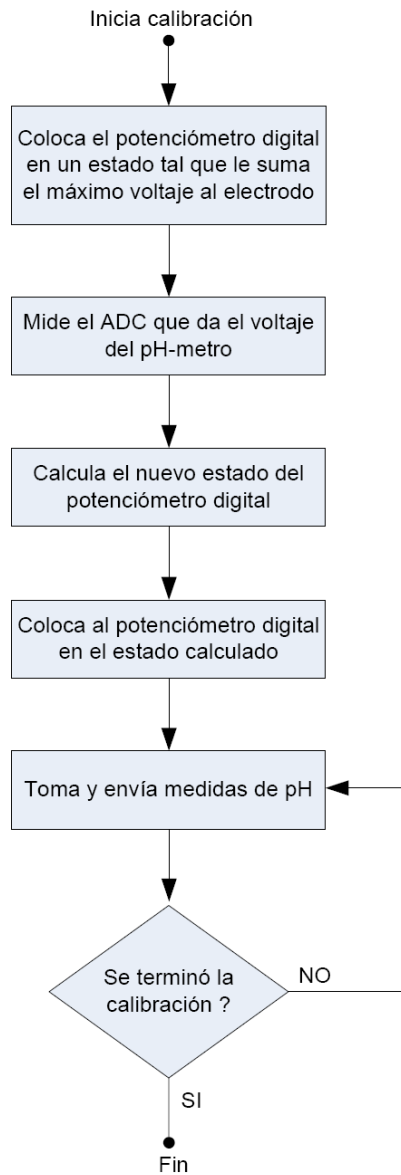


Figura 3.17: Diagrama de flujo de la función de lectura del sensor de pH

## Calibración



**Figura 3.18: Diagrama de flujo de la función de calibración del sensor de pH**

En la calibración del sensor de pH el nodo sensor fija la posición del potenciómetro digital.

El proceso de ajuste del potenciómetro se debe de realizar cuando el electrodo esta en solución buffer 7, este es el primer punto para la calibración. Realiza una medida de voltaje del pHímetro con el potenciómetro digital en su posición máxima, es decir con el mayor offset que se le puede agregar. Luego en base a esa medición se sabe cuanto offset, se debe agregar a la medida en el electrodo.



# Capítulo 4

## Interfaz con el usuario

### 4.1. Elección de la plataforma

La interfaz con el usuario fue desarrollada en Java. Esto se debe en primera instancia a que existe una biblioteca de TinyOS en Java que permite un manejo sencillo de los mensajes que llegan por el puerto USB desde el Mote base y los que se envían al mismo.

Por otro lado Java es una plataforma muy interesante para este desarrollo ya que la aplicación puede ser ejecutada desde distintos sistemas operativos sin tener que realizar ningún tipo de cambio en el código de la misma.

### 4.2. Programación

Al ejecutar la aplicación, en Windows haciendo doble clic en el archivo “*AplicacionRIL.bat*”, se abre una ventana como la que se muestra en la Figura 4.1. En esta ventana se muestra el último estado de las variables que le llega al Mote base desde los Motes sensores. Es también a través de ella que se puede seleccionar la frecuencia con la que se quieren obtener mediciones o ingresar al panel de calibración de pH.

El display se divide en tres grandes secciones. En la parte superior se puede seleccionar el periodo de muestreo de temperatura y de pH de los nodos ( todos tendrán el mismo periodo de muestreo para cada variable). En la parte central se encuentran los displays de cada uno de los fulones donde se encuentra un nodo sensor. En la Figura 4.1 se muestra el display en el caso de que se tengan dos nodos sensores. Si hubiera un nodo más el display sería idéntico al de la figura pero con un display de nodo sensor más correspondiente al tercer nodo. En estos se muestran los últimos valores de las variables recabadas por cada uno de los sensores y la hora en que se obtuvieron. Además se muestran los periodos de muestreo de temperatura y de pH enviados por los nodos sensores, que si no existen inconvenientes en la comunicación deberán coincidir con el periodo seleccionado por el usuario. Por último en la sección inferior se tiene un botón para iniciar la calibración de pH de un nodo sensor.

Además de mostrarse en pantalla los valores actuales de las variables temperatura y pH, se guardan los mismos en un archivo .csv por cada variable y por cada nodo, también se guardan por mes para limitar en cierta forma los tamaños de los archivos.

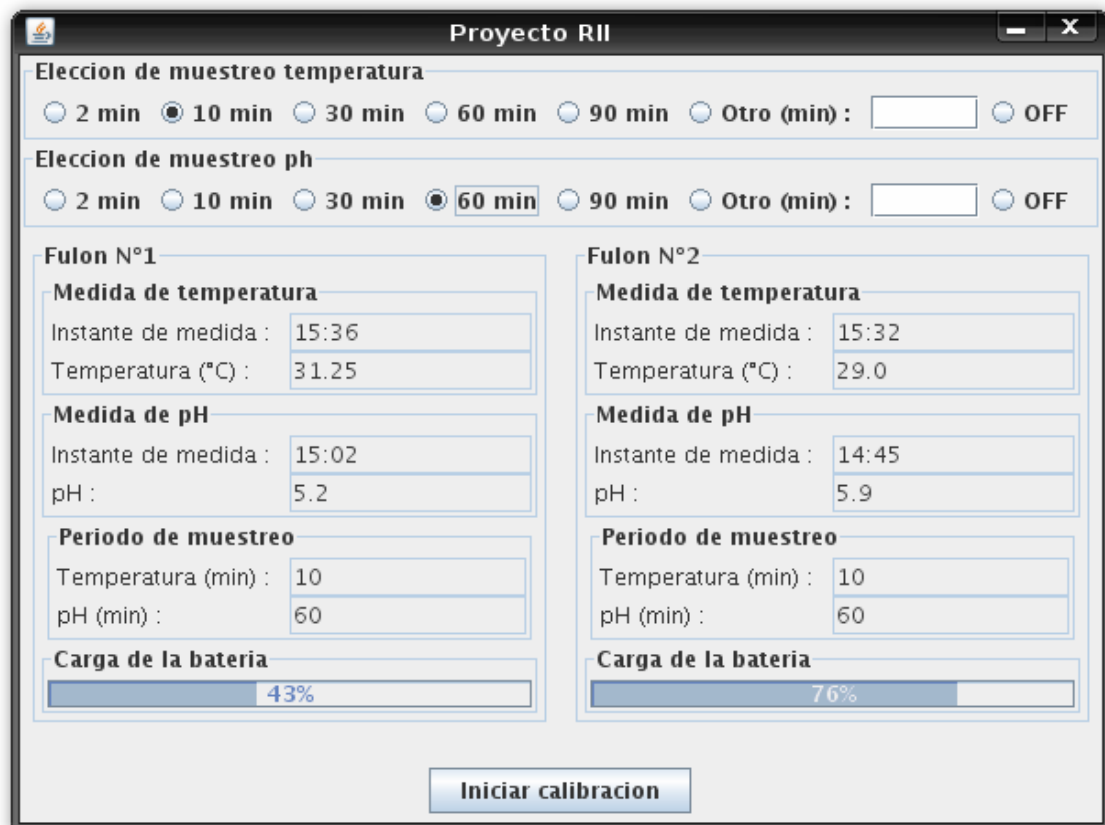


Figura 4.1: Display principal de la aplicación

La aplicación se divide en tres grandes partes

- *Enviar mensajes a los nodos sensores:* permite al usuario cambiar los periodos de muestreo de los sensores. Además es utilizado por la sección de calibración para avisarle a la red que se está calibrando un nodo determinado.
- *Procesar mensajes recibidos de los nodos sensores:* procesa los mensajes que le llegan al PC desde el nodo base a través del puerto serial, mensajes que a su vez le llegan al nodo base por aire desde los nodos sensores.
- *Realizar la calibración de pH:* permite al usuario llevar a cabo la calibración de un nodo específico que el usuario deberá seleccionar.

#### 4.2.1 Formato de los mensajes

Dentro de la librería que posee TinyOS en Java se encuentra una clase llamada "MoteIF. Esta clase provee una interfaz para recibir mensajes desde y enviarlos a los motes a través del puerto serial (o de cualquier otra forma de comunicación que se tenga).

La estructura de los mensajes que utilizan los motes se especifica en un archivo de extensión .h. A partir de este archivo es posible, al momento de compilar la aplicación en TinyOS, crear clases automáticas para cada tipo de mensajes que

se tenga. Estas clases proveen funciones que permiten la lectura o la modificación de cada uno de los campos del mensaje. Cuando se quiere leer un campo de un mensaje que llega se utiliza la función *get\_NombreDelCampo* y cuando se quiere escribir en uno se utiliza *set\_NombreDelCampo*.

## 4.2.2 Enviar mensajes a los nodos

Los mensajes enviados por la base hacia los nodos contienen los siguientes campos:

- *Periodo\_temp*: es un entero ( $P_T$ ) que se utiliza para seleccionar el periodo de muestreo de la temperatura. El periodo se obtiene de multiplicar  $P_T$  por un contador que tienen los nódos sensores. El tamaño de este contador es lo que fija el periodo más pequeño que se puede utilizar. En esta aplicación el contador dispara cada dos minutos.
- *Periodo\_pH*: es análogo al de temperatura.
- *N\_mote*: este campo se utiliza para identificar al nódodo que se está calibrando. Si el valor de este campo es 0 quiere decir que no se está calibrando ningún nódodo. Al comenzar a calibrar en el valor de este campo se coloca en número de identificador (ID) del nódodo a calibrar. Al terminar la calibración este campo vuelve a ser 0. Mirando este valor cada nódodo sensor se dará cuenta si lo están calibrando o no.

La función de enviar mensajes se encuentra en un loop permanente. Cuando el usuario elige un nuevo periodo de muestreo, ya sea de temperatura o de pH, se envía un mensaje de configuración con todos los parámetros, hayan cambiado ó no.

Además, cuando se inicia la calibración el usuario debe indicar cuál es el nódodo a calibrar, como se explicará más adelante. También en este caso se envía un mensaje con todos los parámetros.

## 4.2.3 Recibir mensajes de los nodos

Los mensajes que recibe la base provenientes de los nódodos sensores tienen los siguientes campos:

- *ID*: es el número de identificación del mote sensor que generó el mensaje. Esta es la forma que se tiene para saber a quién corresponde el mensaje que llega.
- *Temp*: en este campo se encuentra la medida de temperatura enviada por el nódodo sensor en hexadecimal. Es el mismo valor que el integrado de temperatura le envía, por medio del bus I2C, al nódodo sensor.
- *pH*: es el voltaje que se utiliza, junto con la temperatura, para calcular el pH de la sustancia medida por el nódodo sensor. El nódodo

sensor obtiene este valor luego de aplicar la ecuación 3.19 con las lecturas de los dos conversores A/D del mote.

- *Periodo\_temp*: este campo se refiere a lo mismo que el campo de igual nombre que se encuentra en los mensajes enviados desde la base. Debido a que el protocolo de transmisión es de “best-effort” se necesita de alguna forma saber si a los nodos sensores le llegó el cambio realizado por el nodo base. Es por esto que los nodos sensores envían los períodos que tienen hacia el nodo base.
- *Periodo\_pH*: es análogo al de temperatura.
- *Voltaje\_bat*: es un valor obtenido por el mote que representa el voltaje de las pilas. Para obtener el voltaje real a partir de este campo hay que aplicar la ecuación 4.1.

$$V_{bat}^{real} = \frac{1100 \cdot 1024}{Voltaje\_bat} \quad (4.1)$$

Al recibir un mensaje por el puerto serial, la aplicación procesa el mismo como se muestra en el diagrama de la Figura 4.2.

Cuando le llega un mensaje primero se ve si corresponde a un nodo que se esté calibrando y en caso de que sea así se ejecuta una rutina de la función de calibración. Si esto no es así se procesa el mismo desplegando en el display correspondiente los valores de las variables del mensaje. Además se guardan las medidas tomadas en archivos .csv (valores separados por coma). Por cada nodo y por cada variable sensada hay un archivo. Para que no se genere un archivo extremadamente largo, se decidió crear un archivo distinto por cada mes del año.

Como se ve en el diagrama de la Figura 4.2 cuando llega un mensaje válido con muestras tomadas por el nodo sensor existen dos posibilidades. La primera es que solo llegue la medida de temperatura y en ese caso se procesa la misma sin tomar en cuenta el pH. En el caso del pH la situación cambia, ya que para calcular el mismo es necesario conocer la temperatura a la que se encuentra la sustancia. Por lo que por cada medida que se quiera obtener de pH se obtiene también una de temperatura. Esta medida se decidió desplegarla en pantalla y loguearla porque no tiene sentido no utilizar un valor válido de sensado si ya se obtuvo, independientemente de que se haya solicitado o no.

Se definen ciertas palabras para que el Mote sensor le envíe al nodo base para identificar situaciones especiales. Estos códigos son solamente para que la aplicación de Java pueda identificar las distintas situaciones y tomar las acciones correspondientes según la situación. Para el usuario que utiliza la aplicación esto es invisible. Un primer código, “*SinMed*”, se define para que el nodo base coloque esta palabra en el campo de temperatura o pH en un mensaje si no ha sensado las mismas. Esto se debe a que por el protocolo de transmisión, en cada mensaje enviado por un nodo sensor se encuentran todas las variables, se haya obtenido una medida de la misma o no. Se define otro código, “*ErrCom*”, para indicar que hubo un error en la comunicación I2C entre el Mote sensor y el sensor de temperatura.

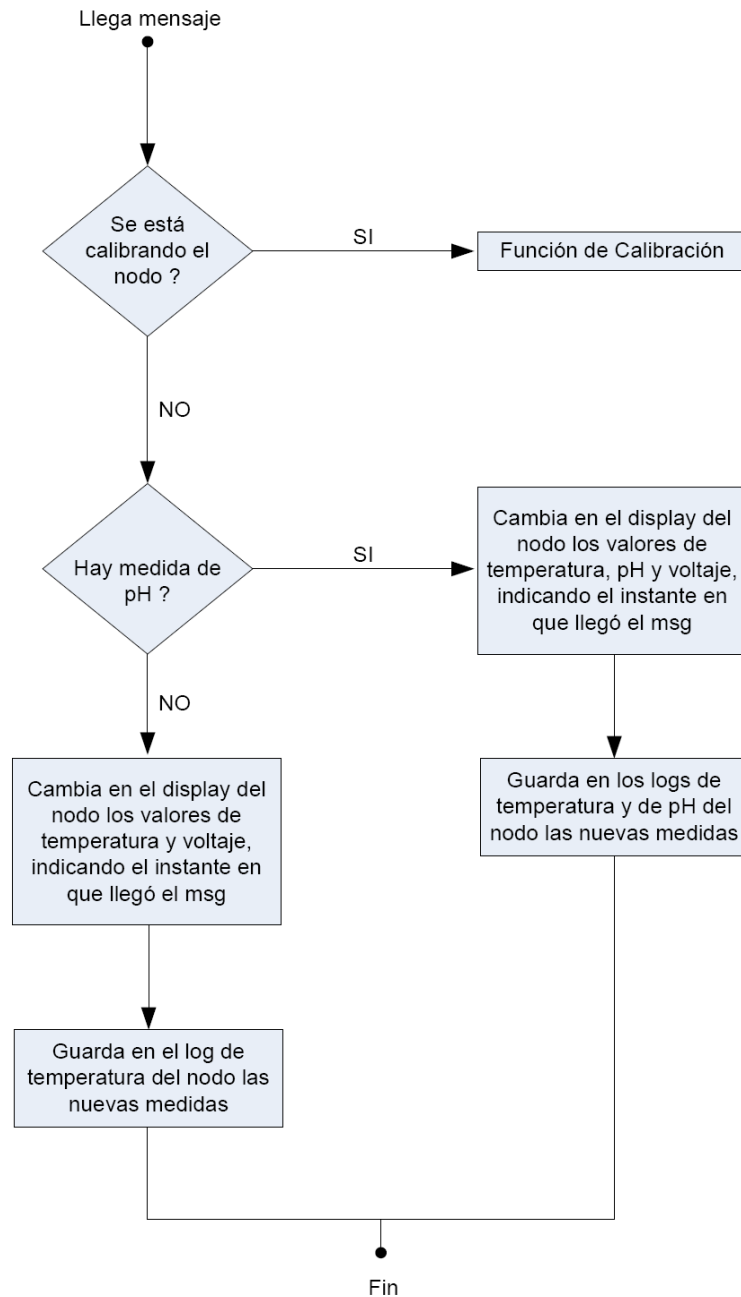


Figura 4.2: Diagrama de la función de llegada de mensaje

#### 4.2.4 Calibrar el sensor de pH de un nodo determinado

Para la calibración se necesita que el usuario coloque el electrodo en soluciones buffers. Es por esto que se deberá desmontar el electrodo de pH del fulón y por tanto también el Mote sensor.

Al apretar el botón “Iniciar calibración” del panel principal, se le pregunta al usuario mediante una ventana emergente cuál es el nodo que quiere calibrar, como se ve en la Figura 4.3.

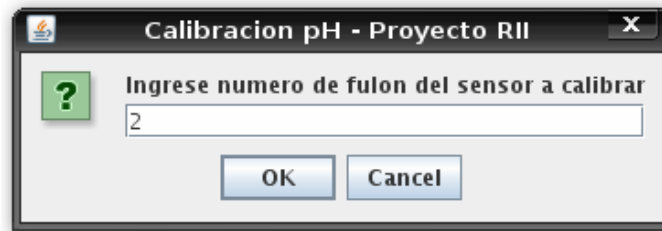


Figura 4.3: Ventana emergente para la calibración del sensor de pH

Luego que éste elige el nodo a calibrar se abre otro panel que lo guiará en el proceso de calibración del sensor de pH (Figura 4.4). A partir de que el usuario selecciona el Mote a calibrar ya no se loguearán ni mostrarán en pantalla las medidas que lleguen del mismo hasta que se salga de la calibración.

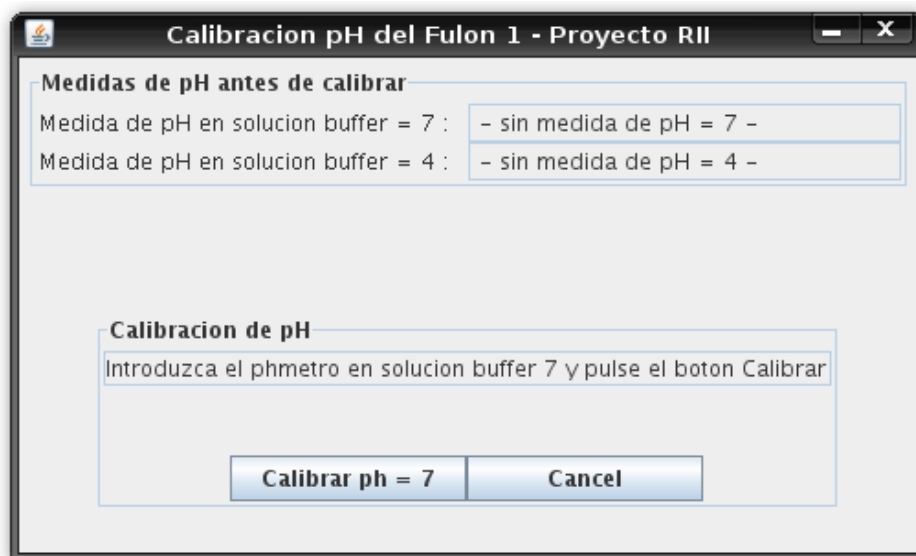
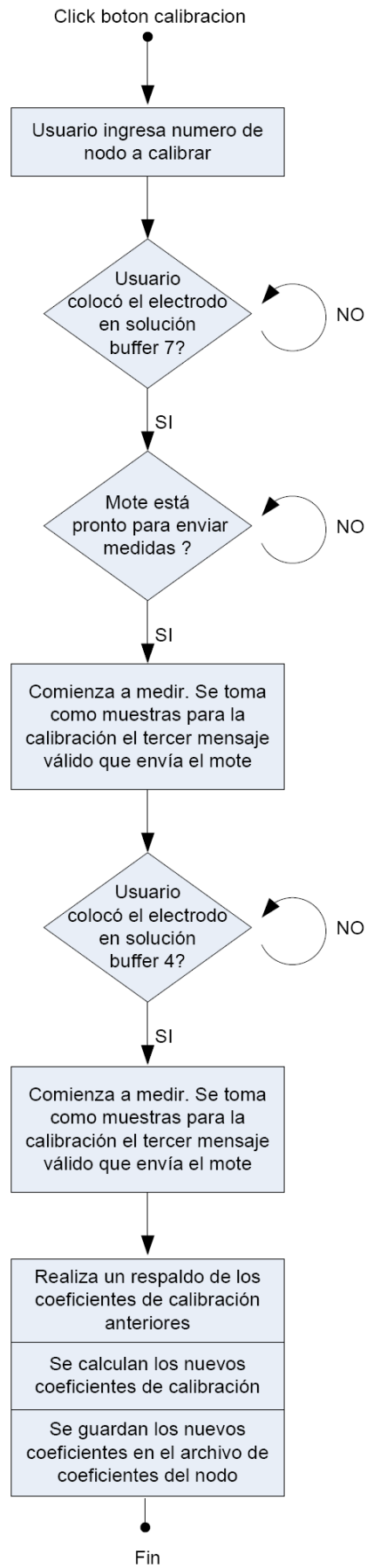


Figura 4.4: Display inicial de calibración de la aplicación

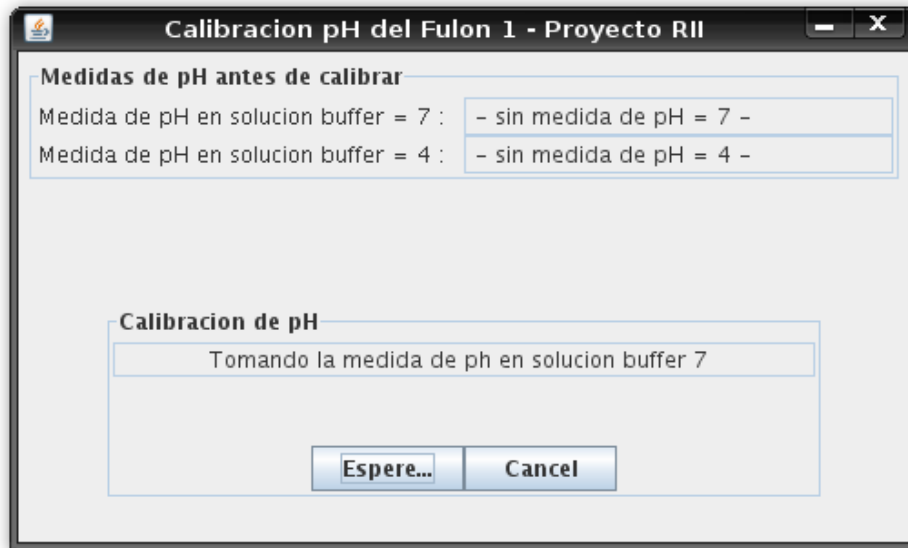
La calibración del sensor se realiza como lo muestra el diagrama de la Figura 4.5.

Para que el Mote sensor inicie la calibración, la base envía en los mensajes a los nodos sensores un campo con el número de Mote que se está calibrando, como se vio en la sección 4.2.2. Cuando el nodo sensor escuche el mensaje y ve que su número de Mote aparece en dicho campo se da cuenta que lo están calibrando, entonces cambia el período de muestreo de datos a 2 minutos y comienza a realizar las tareas necesarias para la calibración, como se explicó en la sección 3.2.6. Luego que el nodo sensor termina, avisa a la aplicación del nodo base que está pronto para enviar medidas para la calibración ingresando el código “EmpCal” en el campo de medida de pH.

Luego de que el usuario ingresa el nodo a calibrar, se le pide que coloque el electrodo en solución buffer 7. El orden con que se utilizan las sustancias buffers es importante. Primero debe tomarse la medida de la solución buffer 7 y luego en 4, ya que la solución 7 tiene una menor concentración de iones de hidrógeno y por tanto no contaminará a la otra. Cuando el usuario coloca el electrodo en la solución buffer 7 avisa que está pronto apretando el botón “Calibrar pH = 7”.



**Figura 4.5: Diagrama de la función de calibración**



**Figura 4.6: Display de calibración cuando se está tomando el pH de la solución buffer 7**

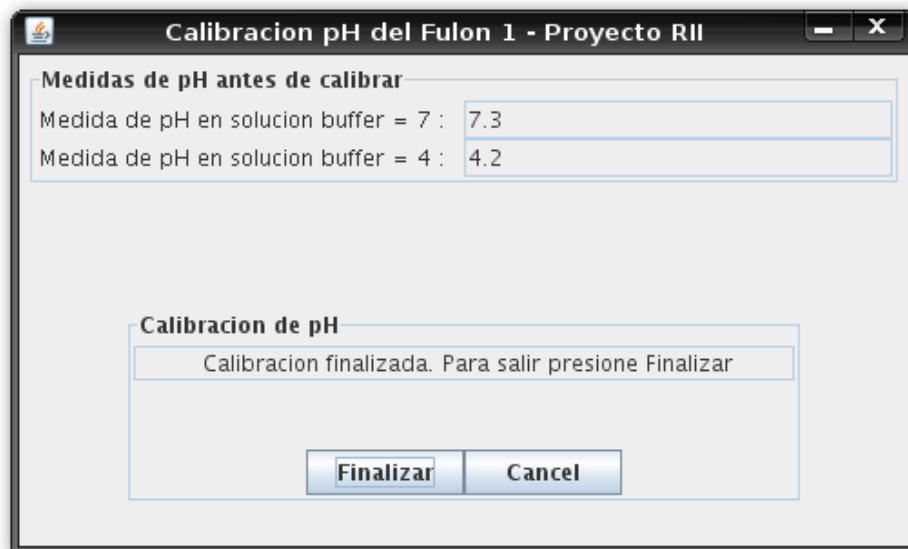
Una vez cumplidas las dos condiciones anteriores: el nodo sensor avisó que está pronto para enviar medidas de calibración y el usuario avisó que el electrodo ya se encuentra en solución buffer 7, la función calibración comienza a contar los mensajes que le llega al nodo base provenientes del Mote que se está calibrando. El mensaje que se usará para hallar los parámetros de calibración será el tercer mensaje válido, es decir el tercer mensaje que le llegue por el puerto serial al PC con valores de temperatura y pH (para obtener los parámetros de calibración se necesita tanto el pH como la temperatura, como fue explicado en la sección 3.2.5). Se espera al tercer mensaje para forzar al sensor de pH a por lo menos estar tres periodos de muestreo de calibración (6 minutos) inmerso en la solución antes de tomar la medida que se utilizará en la calibración. En la Figura 4.6 se muestra el display que ve el usuario mientras se está tomando la medida de pH de la muestra.

Luego de tomar las medidas para la solución buffer 7, se le pide al usuario que coloque el electrodo en la solución buffer 4 e ingrese cuando esté pronto. A partir de que el usuario aprieta el botón “Calibrar pH = 4” la función de calibración comienza a contar los mensajes procesando los mismos de igual forma que lo hizo con la solución buffer 7. Por su parte, el display sigue manteniendo la misma estructura.

Luego de cada uno de los puntos de pH relevados, se muestra en el display de calibración el valor de pH que se obtiene utilizando los parámetros de calibración anterior a la que se está desarrollando. Esto sirve para que el usuario pueda evaluar el desgaste del electrodo y corregir la frecuencia de calibración del mismo.

Al finalizar la calibración el display de calibración queda como el de la Figura 4.7.





**Figura 4.7: Display final de la calibración**

Cuando se obtienen medidas válidas para los dos puntos de calibración se hallan los parámetros  $\alpha$  y  $\beta$  de calibración de acuerdo a lo visto en la sección 3.2.5. Al hacer esto, primero se guarda en un archivo de backup los valores de la calibración anterior y luego se guardan los nuevos en el archivo de los parámetros de calibración del nodo correspondiente. Estos son archivos de texto plano, en donde la primera línea corresponde al coeficiente  $\alpha$  y la segunda al  $\beta$ .

### 4.3. Compatibilidad con el hardware del cliente

Si bien Java permite correr el código compilado en un sistema operativo diferente al cual fue compilado, no significa que no sea necesario realizar algunos pasos previos de configuración de la PC en la cual se va a ejecutar la aplicación. Incluso esto también debe hacerse si se utiliza el mismo sistema operativo con que se compiló la aplicación.

Dado que se trabajó en una distribución de Linux (específicamente Ubuntu) y la curtiembre dispone en planta de una PC con Windows XP, se debió buscar el software necesario para correr la aplicación en Windows XP.

Luego de obtener las librerías de Java y los drivers necesarios se realizó un CD con el software requerido y un manual de instalación para que la misma pueda ser llevada a cabo por parte del cliente (por más información ver Anexo VII).

# Capítulo 5

## Implementación

En esta parte se mostrará como quedaron terminados los distintos PCB de medida, que condiciones puso el fabricante para realizarlos y como se resolvió el montaje mecánico del mismo para adaptarlo al fulón, logrando su funcionalidad in situ.

### 5.1. Realización de los sensores

Todos los PCBs fueron realizados por la firma PCBCOM. Se le entregó el diseño de las placas realizado en el programa de diseño Eagle de Cadsoft. También se le suministró todos los componentes necesarios para realizarlos.

Los componentes de las placas dibujadas en el Eagle se obtuvieron a partir de encapsulados estándares de la biblioteca del programa. Las resistencias usadas son fueron de montaje superficial con encapsulado 1206 (son las que se consiguen en plaza) y los capacitores también son de montaje superficial pero encapsulado 0805 (se consiguen en plaza, son un poco más pequeñas que las 1206).

De los componentes de las placas, los integrados, el conector de 51 pines y el conector BNC fueron importados de Estados Unidos, algunos a través de muestras gratis que dan los fabricantes para pruebas (caso Texas y Maxim), y otros a través de compras por Internet (en la página web de Digikey). Todos fueron solicitados por Web y enviados a la empresa MiamiBox en Miami que se encargó de ingresarlos al país. Los demás componentes como las resistencias, capacitores y conectores molex, se compraron en comercios de plaza.

#### 5.1.1 Temperatura

El sensor de temperatura se realizó tan pequeño como fue posible. Una de las principales razones para hacerlo así es que este no se encontrará en contacto directo con la sustancia a medir, sino que estará dentro de una vaina metálica empotrada al fulón. Por lo tanto es necesario disminuir al máximo posible el volumen de aire entre la sustancia a medir y el sensor y con esto obtener una transferencia de calor tan buena como sea posible.

El diseño construido en el Eagle se basó en el esquema de la Figura 3.2 y el sensor construido se muestra en la Figura 5.1

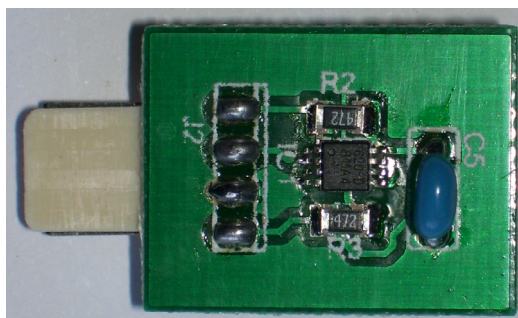


Figura 5.1: Foto del PCB de temperatura realizado

### 5.1.2 pH

A partir del esquema de la Figura 3.16, en el Eagle, se realizó el diseño para construir el impreso. El PCB construido se muestra en la Figura 5.2.

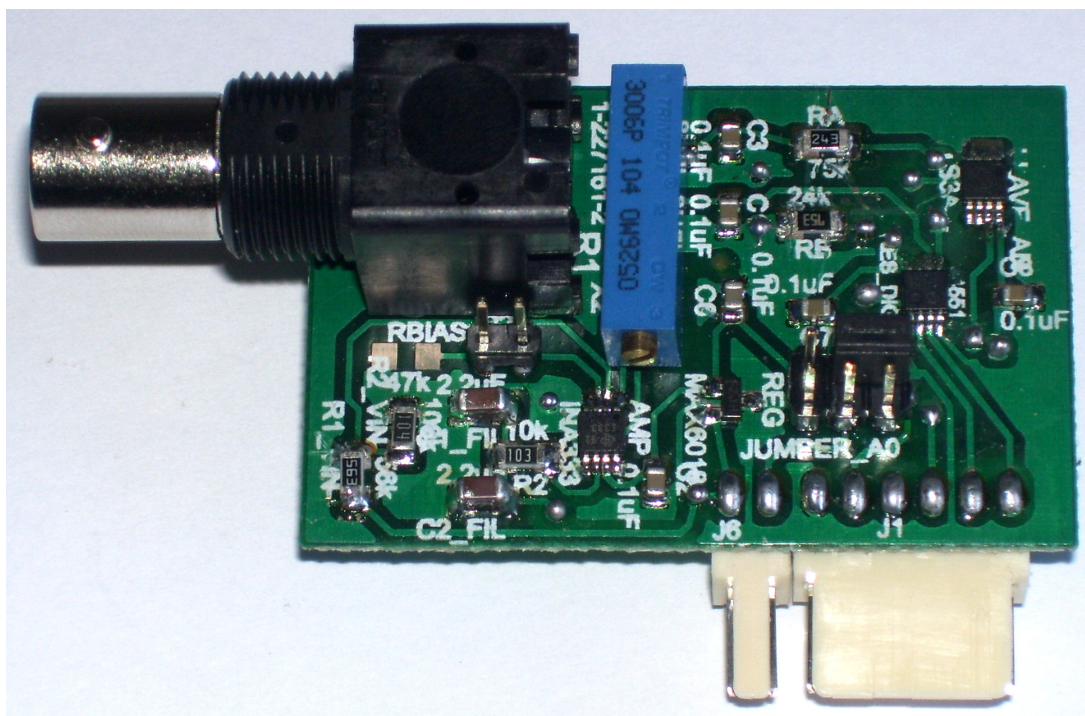


Figura 5.2: Foto del PCB de pH realizado

Como se puede observar en la Figura 5.2 este diseño es mucho más complejo que el diseño de temperatura. Fue necesaria la utilización de dos capas del impreso, debido a la cantidad de pistas trazadas.

### 5.1.3 Consideraciones para los diseños de los impresos

A la hora de diseñar el circuito en Eagle el fabricante del PCB impuso algunas restricciones, que se detallan a continuación:

- Las curvas de las pistas no deben ser en ángulo recto. Esta condición

la impone el perclorato de hierro que disuelve al cobre entre pistas y si estos son rectos el perclorato no trabaja óptimamente

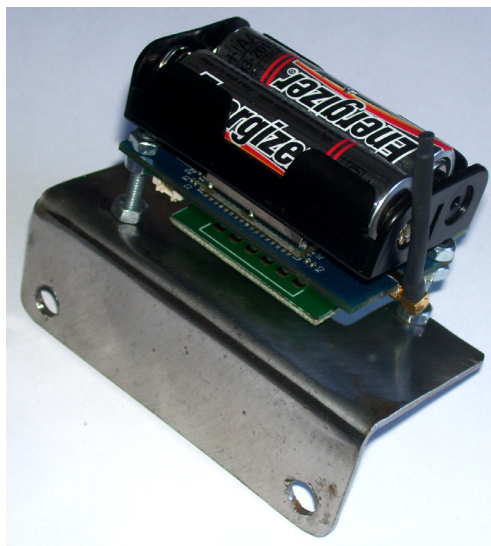
- Realizar la menor cantidad posible de bias (circuitos que unen las pistas entre la capa superior y la inferior). Esto se debe a que el fabricante tiene que hacerlas a mano, artesanalmente, estas conexiones.
- Respetar ciertos anchos mínimos de pistas y separación al Ratsnest (el Ratsnest es cobre que se deja en los lugares entre pistas y que se conecta al potencial de referencia).

Fue necesario realizar varios diseños antes de que el fabricante estuviera en condiciones de realizar el circuito impreso.

## 5.2. Montaje mecánico

Para la prueba en la planta del sensor de temperatura fueron necesarias varias consideraciones. Estas vienen principalmente del hecho de que es necesario montar el Mote al fulón, para poder colocar la vaina del sensor en contacto térmico con la solución dentro del fulón.

Para realizar el montaje fueron necesarios dos dispositivos, el primero, una caja estanca, de IP 66, apta para soportar los vapores corrosivos que salen de los fulones cuando descargan los cueros. A la caja hubo que hacerle un soporte por dentro para que el Mote no quedara suelto dentro de la misma, soporte mostrado en la Figura 5.3. El segundo, la vaina de metal que estará en contacto con la solución.



**Figura 5.3: Soporte para el Mote sensor**

En la Figura 5.4 se puede apreciar la caja estanca utilizada y como de ella sale la vaina en donde se aloja el sensor de temperatura, la mitad de la vaina queda en el interior del fulón. En el lateral de la caja, están los tornillos que fijan el

soporte del Mote, este soporte se puede ver en la Figura 5.4. La caja se fija con tornillos autoroscantes a la madera del fulón.

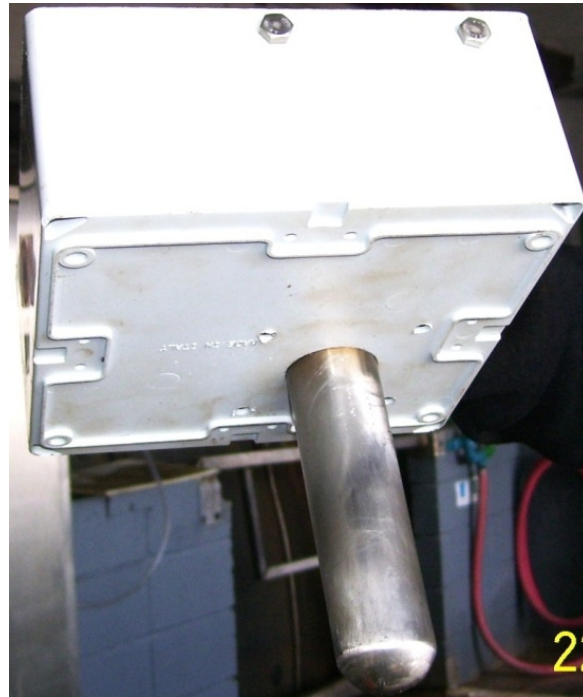


Figura 5.4: Caja estanca y vaina del sensor de temperatura

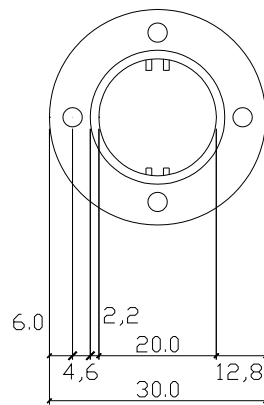
### 5.2.1 Vaina de temperatura

La vaina de temperatura se diseñó en base a dos factores. Uno de ellos es que debe ser lo suficientemente robusta como para soportar las sustancias que se encuentran dentro del fulón y también soportar los golpes del cuero al girar el mismo. Para la elección del material a utilizar se consultó al cliente, quien recomendó el acero inoxidable 316 porque soporta las sustancias utilizadas. El otro es que debe ser lo más pequeña posible para tener una buena transferencia de calor entre el integrado que sensa la temperatura y la sustancia que a la que se quiere medirle la misma. Obviamente la vaina debe tener el tamaño suficiente como para alojar al sensor de temperatura construido.

La Figura 5.5 muestra el diseño de la vaina de temperatura. Este diseño se realizó con la ayuda del cliente quien especificó ciertos puntos del diseño como el grosor de la vaina y el material a utilizar. Luego de tener un diseño final se entregó el mismo al cliente quien se encargó de su realización.

Otro detalle importante, es que se realizo un tapón de espuma plast, de diámetro igual al de la vaina, para aislar térmicamente lo más posible el integrado de la temperatura exterior.

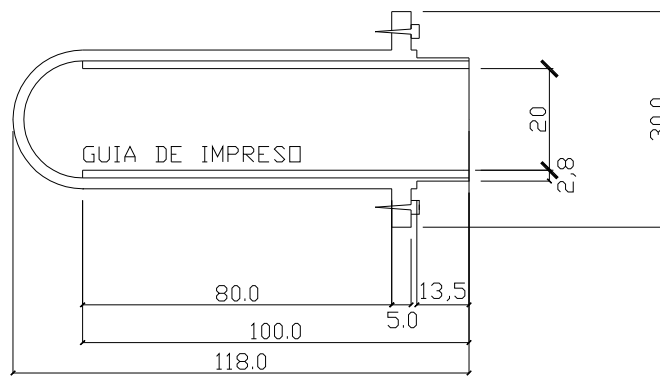
Corte BB trasero (Con rosca)



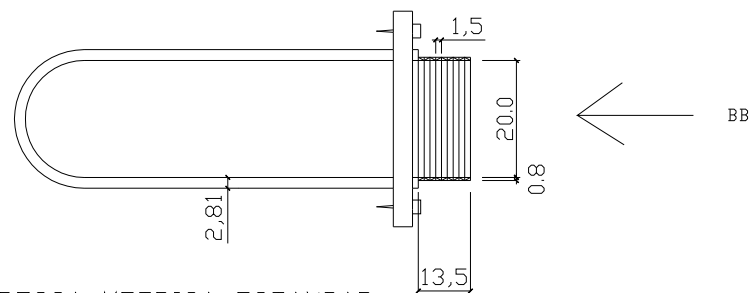
NOTA: UNIDADES MILIMETROS

↖: TORNILLO PARA FULÓN

Corte transversal (interior)



Vista Transversal (Exterior)



ROSCA METRICA ESTANDAR  
M32 x 1,5

**Figura 5.5: Diseño de la vaina del sensor de temperatura**

# Capítulo 6

## Validación

### 6.1. Consumo

Para estimar el consumo del sensor se utilizaron básicamente dos metodologías distintas, una teórica y otra empírica. La teórica se utilizó para determinar el consumo de los impresos de medida de temperatura y pH, con lo cual se relevó el consumo (máximo) de los distintos integrados que los componen, además de las corrientes por caminos a tierra en los circuitos. El empírico se utilizó para determinar el consumo relacionado con el microprocesador y radio del Mote, variando los parámetros de encendido de la radio, para ver cómo influyen estos en el mismo.

#### 6.1.1 Consumo teórico

Consumo de los componentes en impresos		
Integrados	Componente	Corriente ( $\mu\text{A}$ )
pH	Amplificador de Instrumentación	75
	Interruptor Analógico	0.75
	Referencia de Voltaje	5
	Potenciómetro digital	5
	Caminos a Tierra :Voffset <sup>-</sup>	8.1
	Caminos a Tierra : Voffset <sup>+</sup>	11.6
Temperatura	DS620 – Inactivo	2
	DS620 – Activo	800
	Caminos a tierra: I2C Pulls Up	600
<b>Total</b>	Activo	1505
	Inactivo	2.75

**Tabla 6.1: Consumo de los componentes en impresos**

Como se puede apreciar en la Tabla 6.1, la diferencia en el consumo entre ambos estados es abismal, además, observando los consumos en modo activo, se observa que es mucho mayor el del impreso de temperatura que el de pH. Cabe señalar, que estando en modo activo la mayor parte del tiempo, el consumo está dado por el impreso de medida de pH, debido a que se requieren solo 50ms para obtener una medida de temperatura, contra 40s para obtener una medida estable de pH. Si estuviera los circuitos de medida constantemente en modo activo, con una pila de 3000mAh, se lograría una vida media por los sensores de 2000h, (el consumo más importante se da en el integrado DS620). Pero como solo se toman dos medidas por hora, ponderando a un tiempo de 100ms por hora (2 medidas), se

logra aumentar exponencialmente la autonomía, haciendo despreciable el consumo del mismo respecto a por ejemplo la radio del Mote.

### 6.1.2 Consumo experimental

Todos los ensayos realizados, tanto para observar el consumo del sistema o verificar el correcto funcionamiento del mismo, se hicieron colocando una resistencia (de  $2\Omega$  al 1%) en serie con las pilas. Se midió con el osciloscopio la diferencia de potencial entre los bornes la misma.

#### pH

En la Figura 6.1, se observa el ensayo realizado para determinar la diferencia de consumos entre el impreso de pH encendido y apagado, verificando el cálculo teórico. Para realizar este ensayo, no se tomaron medidas de temperatura, así se tiene certeza de que la medida depende solo de este impreso.

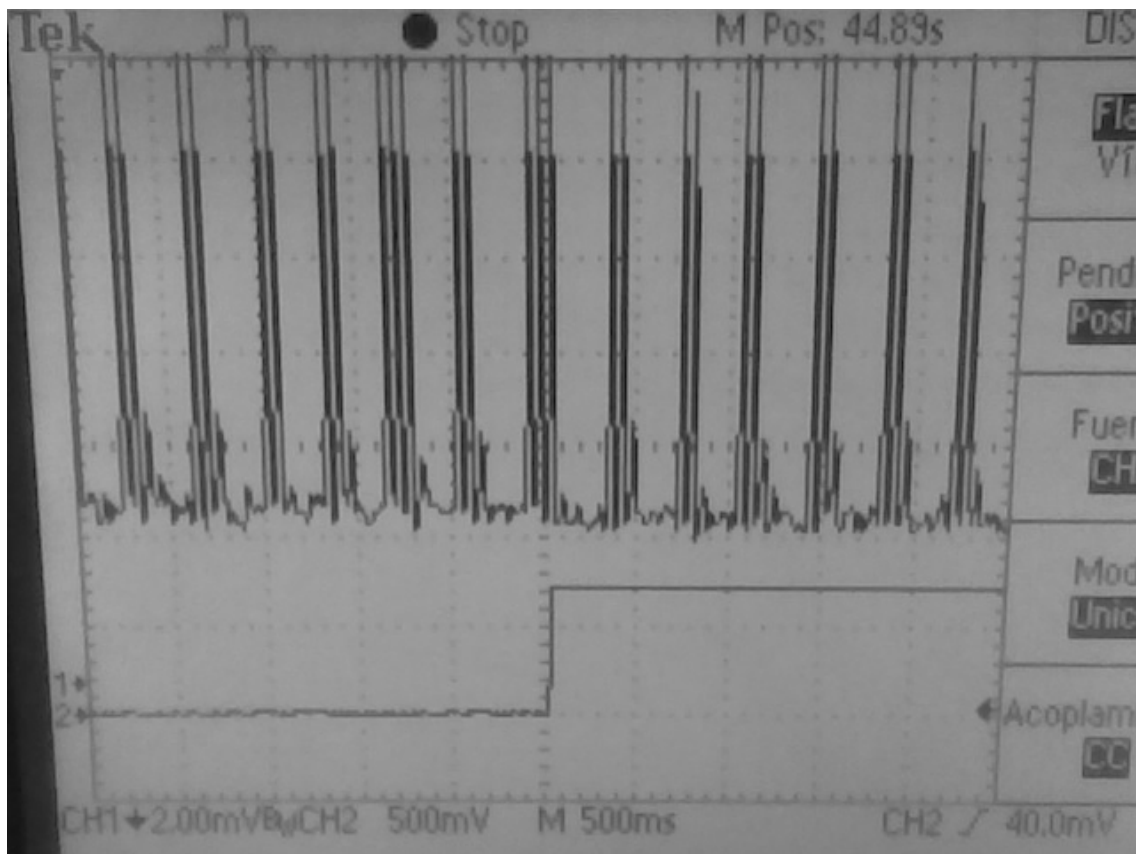


Figura 6.1: Cambio de estado (On-Off) en impreso de pH

En la misma se ve el consumo en la parte superior de la imagen y el cambio de estado en la llave del impreso en la inferior. En la primera mitad de la imagen (llave en 0 lógico) la alimentación del impreso estaba activada. En la segunda mitad estaba el circuito desenergizado, con lo cual en esta transición se observa un muy leve descenso en el consumo (casi imperceptible). Esto valida la suposición de que pesa muy poco (encendido constantemente o no) en el consumo total.

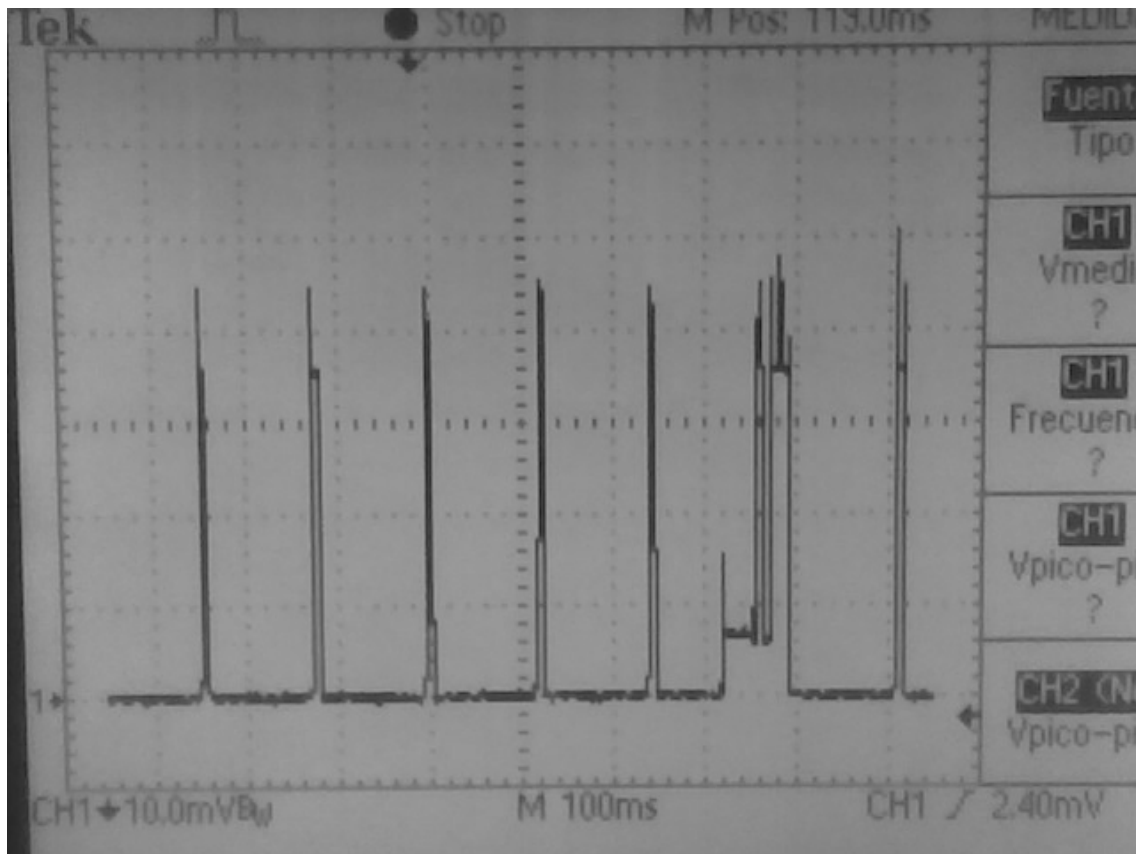


Otra particularidad que se aprecia en la figura, es el nivel de continua que está unos 6mV por encima de la referencia. Esto no se debe a exceso de consumo en el circuito (ya que es del orden de microamperios), sino que se explica por el encendido de los leds, que al momento de la prueba se utilizaban para depurar el programa.

En la imagen también se observan picos periódicos en el consumo, esto es por el encendido de la radio y se explicara más adelante.

## Temperatura

En la Figura 6.2 se aprecia el ensayo realizado para observar el consumo del integrado de temperatura. Para esto se realizó un programa en el cual, se tomaba una medida de temperatura cada segundo, lo que permitía sincronizarse con la medida, barriendo el osciloscopio a 100ms/div (tiene 10 divisiones).



**Figura 6.2: Consumo del integrado de temperatura**

En la Figura 6.2, a pesar de no verse muy claramente, puede observarse en la imagen los picos de la radio esperando para recibir, luego, en el sexto pico, se ve cómo se activa el integrado de temperatura, un instante después se activa la radio esperando para recibir y luego de eso se activa nuevamente la radio para transmitir el valor de temperatura al Mote base.

Una Ampliación de ese intervalo se ve en la Figura 6.3, este ensayo también se realizó con los leds encendidos, por lo tanto, hay que considerar los 3mA (6mV) de consumo de los mismos. Observando mejor el instante de medida de

temperatura, se ve, en un primer momento la suma de los leds con el consumo del DS620 (30ms), luego el DS620 pasa a estado inactivo (6ms siguientes), apreciándose sólo el consumo de los leds. Esa leve caída de potencial, se debe al integrado de temperatura, situado en el entorno de 1mV, lo que se traduce en unos 500uA de corriente, que se acercan a lo establecido por la hoja de datos del integrado. Se nota además que el tiempo de conversión es menor a 50ms (30ms), tiempo máximo posible de conversión para una resolución de 0.25°C.

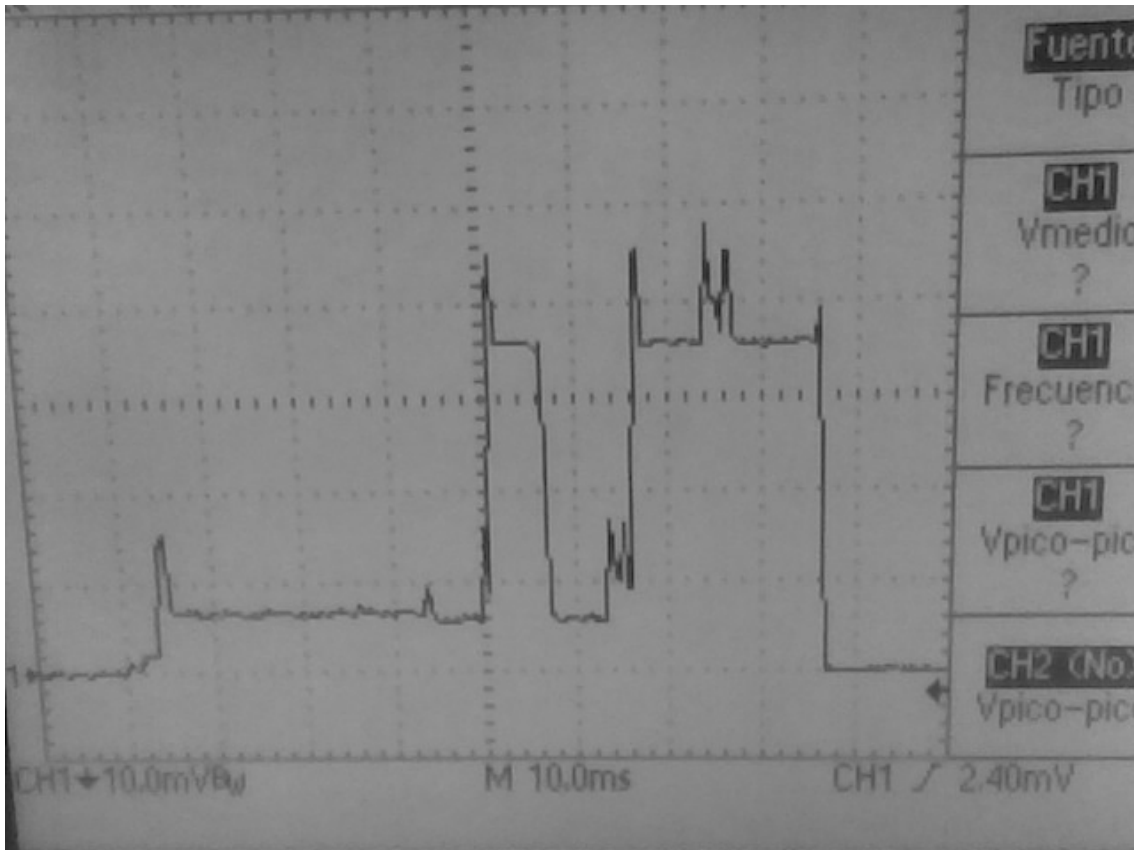


Figura 6.3: Consumo del impreso de temperatura (ampliación de la Figura 6.2)

## Mote

Como lo que más pesa en el consumo del sistema de sensado es la radio del Mote, se enfocó esta parte en tratar de observar a fondo como afecta esta a largo plazo en la autonomía y el desempeño.

La radio del Mote en estado activo consume al escuchar 16mA, un nivel imposible de mantener a base de pilas. Si se quisiera estar constantemente escuchando la autonomía del Mote pasaría a ser menor que 6.5 días, un valor completamente inaceptable. Por esto se decidió utilizar un protocolo orientado al bajo consumo que regule el encendido de la radio al escuchar y enviar paquetes, este protocolo se conoce como “Low Power Listening” (LPL, Escucha de bajo consumo). Este desarrollo regula cada cuanto se enciende la radio para escuchar y cuanto tiempo queda encendida.

El protocolo LPL permite variar el ciclo de trabajo de la radio. Basándose en ese parámetro se analizó el consumo y el desempeño de la misma. Para medir el desempeño, cada vez que se cambiaba el ciclo de trabajo, se medía cuanto

demoraba en responder el Mote sensor a cambios en el Mote base. Para esto se le enviaba un mensaje a través de la base al sensor y una vez que el sensor escuchaba el mensaje, este respondía instantáneamente al mismo. Evidentemente a menor tiempo de ciclo, aumentaba el tiempo de respuesta disminuyendo el desempeño del sistema. Este desempeño resulta importante sobre todo a la hora de la calibración del sensor de pH.

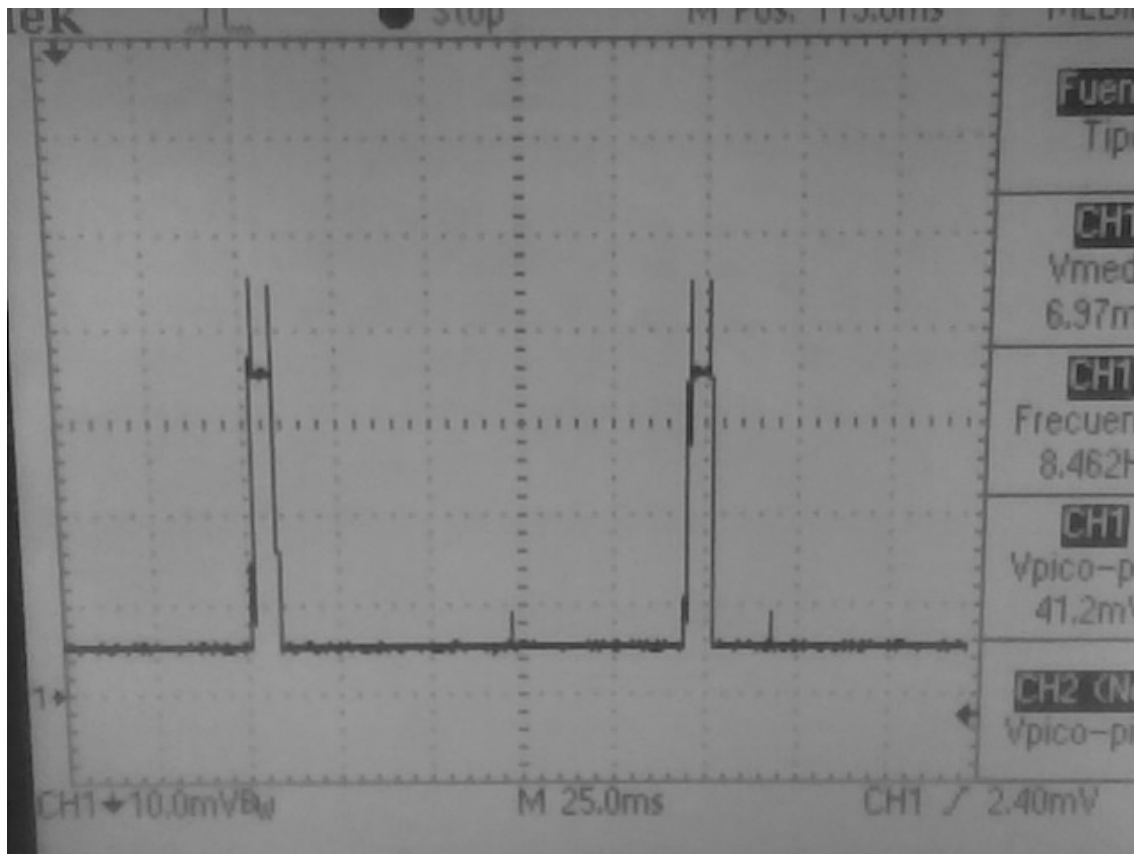
Inicialmente se tomo un ciclo de trabajo de 50% o sea la mitad del tiempo del ciclo definido en el LPL, que está escuchando, como se puede ver en la Figura 6.4.



**Figura 6.4: Consumo de la radio con un ciclo de trabajo del 50%**

Este ciclo de trabajo generaba una disminución del desempeño de la radio imperceptible respecto a un ciclo del 100%. El gran defecto es que, a pesar de disminuir el consumo a la mitad respecto a la radio encendida permanentemente, se seguía teniendo un consumo demasiado elevado.

Se decidió aumentar la apuesta y bajar el ciclo de trabajo al 5%, como se observa en la Figura 6.5 y nuevamente se procedió a probar el desempeño del sistema. Para esto se envió el mensaje al Mote, pero no se logró respuesta del Mote sensor en un tiempo razonable. Esto incrementaría considerablemente el tiempo necesario para hacer una calibración, con los inconvenientes que esto acarrea para el usuario.



**Figura 6.5: Consumo de la radio con un ciclo de trabajo del 5%**

Entonces, para buscar una solución aceptable para el consumo y el desempeño, se decidió utilizar un ciclo de trabajo tal que permita utilizar el Mote 60 días (2 meses) sin necesidad de cambio de pilas. Para esto fue necesario un ciclo de trabajo de 9%. El tiempo de respuesta obtenido para ese ciclo fue de algunos segundos, que se considero aceptable para esta aplicación. Por tanto el consumo promedio del Mote sensor es de 1.53mA.

Podrían haberse implementado otras soluciones para bajar el ciclo de trabajo a niveles mucho menores dado que el problema radica en el poco tiempo en que la radio está encendida. Una manera hubiese sido agregando un pulsador al Mote sensor con lo que sería posible colocarlo en un estado de calibración, que aumente este ciclo de trabajo, para disminuir el tiempo de calibración y luego de finalizada la misma volver a disminuirlo para trabajar en bajo consumo.

Dado que para el envío de datos desde el Mote base al Mote sensor se utiliza el protocolo "collection tree protocol", este no garantiza que los paquetes lleguen a destino si el ciclo de trabajo es muy bajo.

## 6.2. Parámetro del sensor de pH

### 6.2.1 Tiempo de encendido a pH constante

Una vez observados los consumos de los distintos componentes que forman el sensor, se estudió el tiempo mínimo necesario de encendido del circuito de medida de pH. Con lo cual progresivamente se iba aumentando el tiempo de encendido del circuito, al mismo tiempo se observaba la medida analógica del circuito, para ver como variaba esa salida y poder medir el tiempo que demora el circuito en obtener una salida constante, puesto que todas las medidas se realizaban a pH constante y luego de un tiempo prudente de sumergido en la solución el electrodo.

En la Figura 6.6 se observa el comportamiento de la salida, desde el encendido hasta el apagado del circuito. Este tiempo empezó siendo de 12s. Claramente la salida todavía no se había estabilizado al momento del apagado del circuito, generando una lectura incorrecta en el Mote.

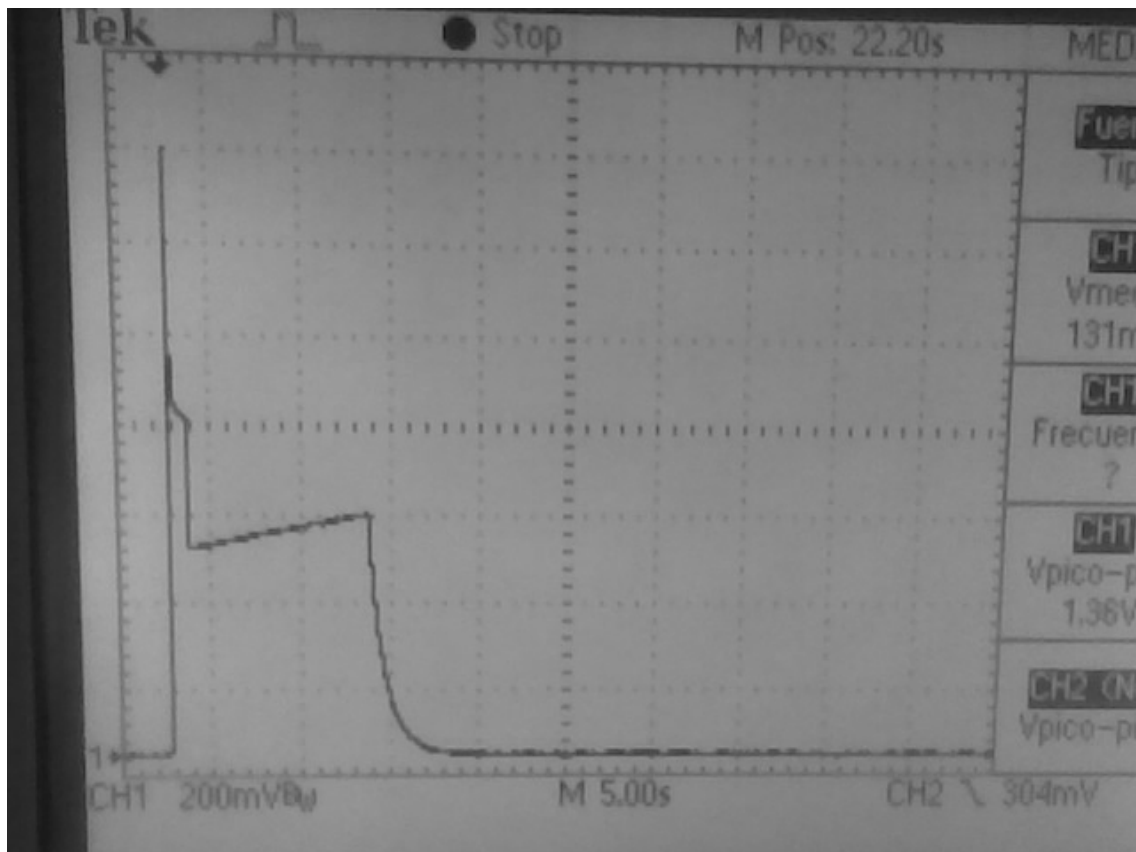


Figura 6.6: Salida analógica,  $t_{on}=12s$

Después de varias pruebas para obtener el tiempo óptimo, se decidió por colocar el mayor tiempo en que fuera posible observarlo en el osciloscopio, este tiempo máximo de visión es de 50s, por ende se utilizó un periodo de 45s, como se observa en la Figura 6.7, en la misma se ve como en los últimos 6 segundos la medida está estable en su valor final, otorgándonos una medida confiable.

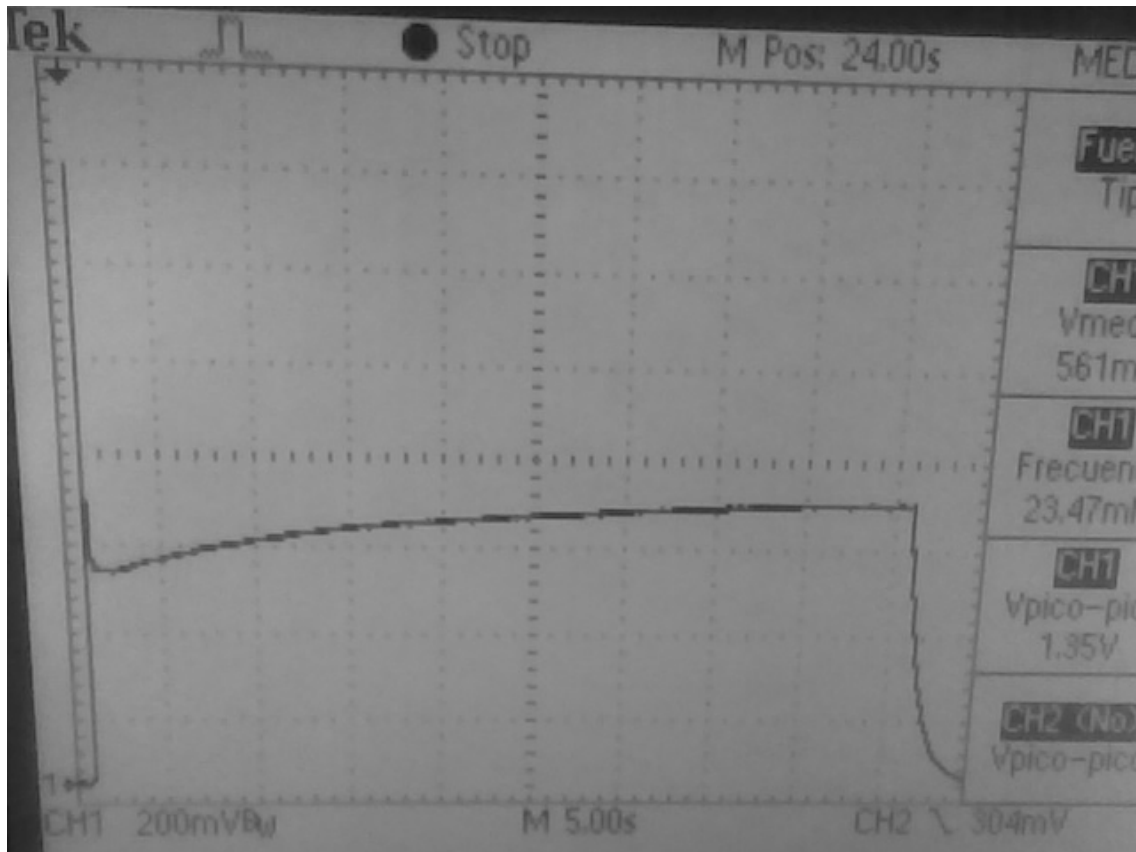


Figura 6.7: Salida analógica,  $t_{on}=45s$

### 6.2.2 Tiempo de respuesta al escalón

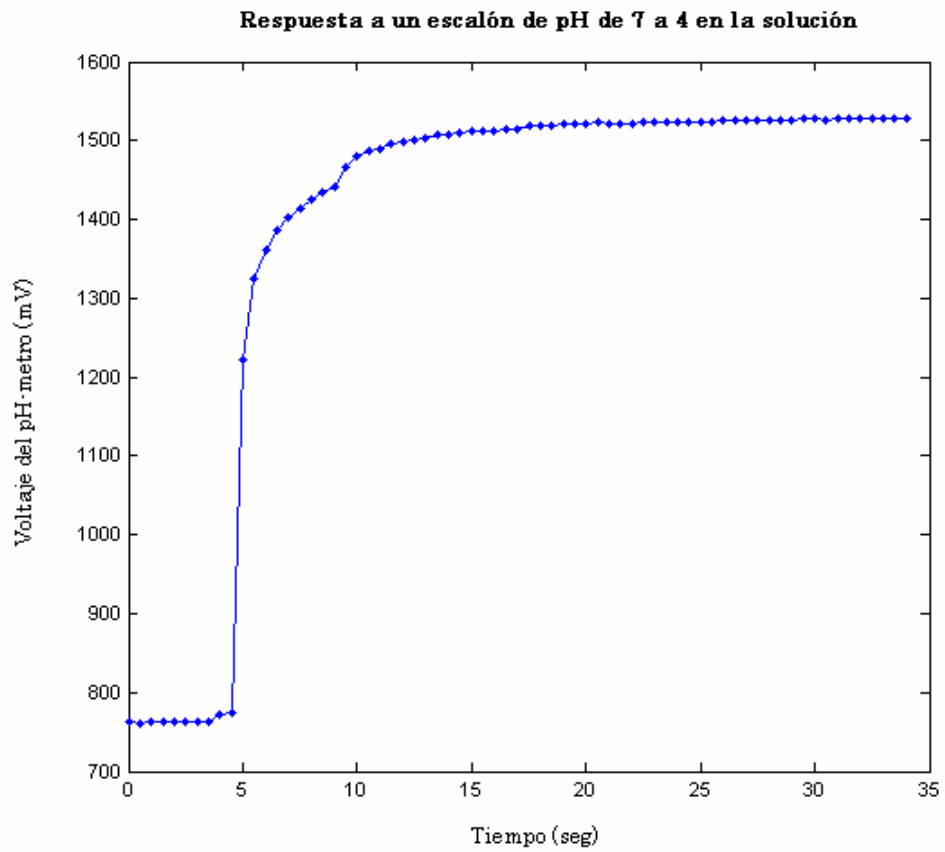
Otro de los tiempos importantes para el sensado de pH es la respuesta a un escalón. Para estos experimentos se colocó el nodo sensor a tomar medidas de pH en intervalos muy cortos (0,5 segundos) y sin apagar el sensor, de forma que se podían observar cambios rápidos en el valor de pH sensado.

En primera instancia se colocó el sensor de pH en solución buffer 7. Luego de dejar estabilizar la medida se cambió a solución buffer 4.

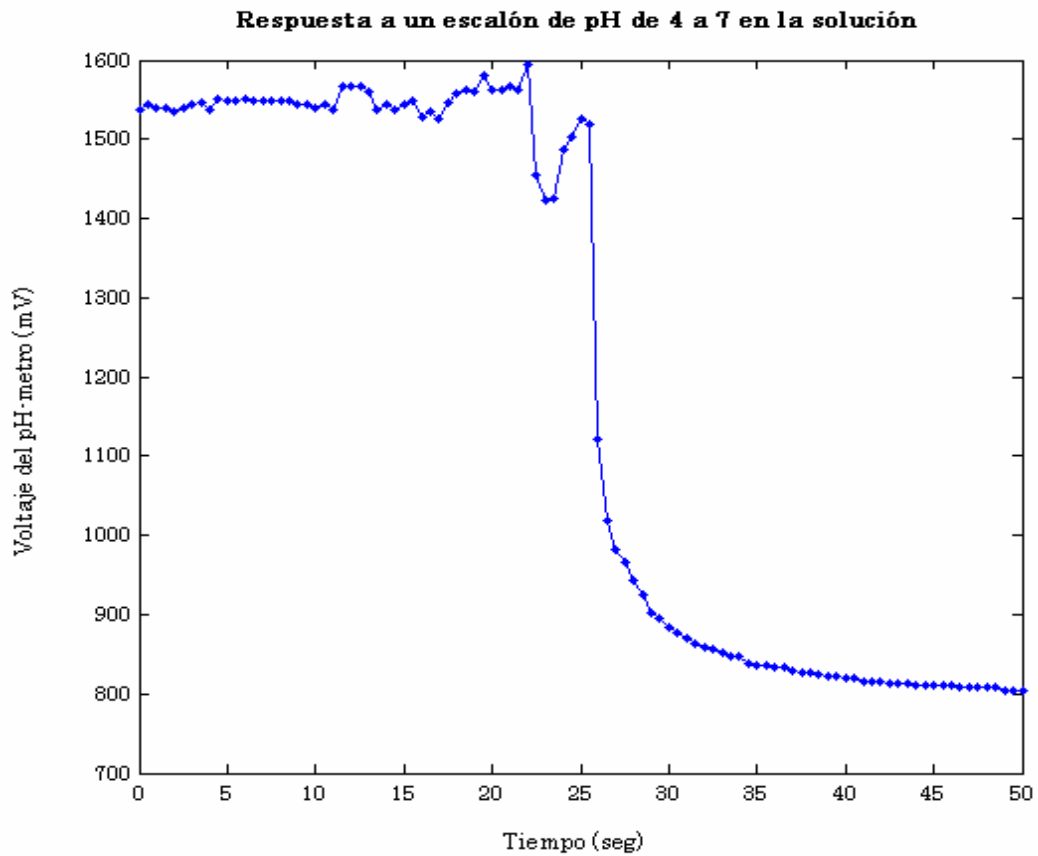
En la Figura 6.8 se ve la entrada del convertor analógico digital del Mote cuando se pasa el electrodo de la solución buffer 7 a la 4. Se observa que a la entrada del convertor se tiene un escalón de 800mV y que la medida llega a su valor final luego de aproximadamente 25 segundos.

El caso inverso, es decir cuando se pasa de medir una solución buffer 4 a una 7, se muestra en la Figura 6.9. En este caso se notan ciertas medidas erróneas al principio. Esto se debe a que en este cambio el electrodo necesita ser limpiado y por es por esto que no puede cambiarse de solución enseguida.

En la Figura 6.9 también se observa una caída de 800mV. El tiempo en el cual la medida llega a su valor final es de alrededor 35 segundos. Esto incluye el tiempo de limpieza del electrodo que fue de aproximadamente 8 segundos.



**Figura 6.8: Respuesta al escalón creciente**



**Figura 6.9: Respuesta al escalón decreciente**

De esta forma se tiene una estimación de la velocidad de respuesta a un cambio máximo en el pH de la solución, que tiene el sistema. De lo anterior se ve que se deduce que se puede llegar a un valor de velocidad: 8,33 seg/pH.

## 6.3. Incertidumbre en la medida de pH

### 6.3.1 Incertidumbre tipo A

La incertidumbre tipo A se calcula mediante la desviación estándar de la media de los resultados que se obtuvieron al medir repetidas veces el mismo mesurando.

Para el cálculo de las incertidumbres tipo A, tanto para pH = 4 como para pH = 7, se tomaron n = 32 muestras del pH medido para cada una de las soluciones buffers.

La desviación estándar de los n resultados se calcula a partir:

$$\sigma_{pH} = \sqrt{\frac{1}{(n-1)} \sum_{i=1}^n (pH_i - \overline{pH})^2} \quad \text{donde} \quad \overline{pH} = \frac{1}{n} \sum_{i=1}^n pH_i \quad (6.1)$$

Luego, la desviación estándar de la media de los n resultados es:

$$\sigma_{\overline{pH}} = \frac{\sigma_{pH}}{\sqrt{n}} \quad (6.2)$$

A partir de lo anterior se calcula la incertidumbre tipo A de los valores relevados:

$$\begin{aligned} \partial pH_A &= 0.004 & \text{para } pH = 4 \\ \partial pH_A &= 0.002 & \text{para } pH = 7 \end{aligned}$$

### 6.3.2 Incertidumbre tipo B

De la ecuación de Nernst (vista en la sección 3.2.2) se obtiene una relación entre el voltaje medido por el electrodo y el pH de la solución. Dado a que todos los términos de la ecuación de Nernst son constantes, a excepción de la temperatura, la relación se puede escribir como:

$$pH = \frac{\alpha}{T} \cdot V_{out} + \frac{\beta}{T} \quad (6.3)$$

La incertidumbre asociada se expresa como:

$$\partial pH = \frac{V_{out}}{T} \cdot \partial \alpha + \frac{1}{T} \cdot \partial \beta + \frac{\alpha}{T} \cdot \partial V_{out} + \frac{(\alpha \cdot V_{out} + \beta)}{T^2} \cdot \partial T \quad (6.4)$$



$\alpha$  y  $\beta$  son constantes que se obtienen por medio de la calibración. A partir de dos soluciones con pH conocidos (solución buffer), midiendo el voltaje del electrodo en cada caso y la temperatura a la que se encuentra cada una de las soluciones, se obtienen los valores de  $\alpha$  y  $\beta$  con sus incertidumbres asociados. A continuación se muestran las ecuaciones de  $\alpha$  y  $\beta$  y de sus respectivas incertidumbres. En este caso se utilizarán soluciones buffers 7 y 4.

$$\alpha = \frac{pH_7 \cdot T_7 - pH_4 \cdot T_4}{V_{out7} - V_{out4}} \quad y \quad \beta = \frac{-pH_7 \cdot T_7 \cdot V_{out4} + pH_4 \cdot T_4 \cdot V_{out7}}{V_{out7} - V_{out4}} \quad (6.5)$$

$$\partial\alpha = \left| \frac{pH_7 + pH_4}{V_{out7} - V_{out4}} \right| \cdot \partial T + \left| \frac{T_7 + T_4}{V_{out7} - V_{out4}} \right| \cdot \partial pH_{buffer} + 2 \cdot \left| \frac{pH_7 \cdot T_7 + pH_4 \cdot T_4}{(V_{out7} - V_{out4})^2} \right| \cdot \partial V_{out} \quad (6.6)$$

$$\begin{aligned} \partial\beta = & \left| \frac{pH_7 \cdot V_{out4} + pH_4 \cdot V_{out7}}{V_{out7} - V_{out4}} \right| \cdot \partial T + \left| \frac{T_7 \cdot V_{out4} + T_4 \cdot V_{out7}}{V_{out7} - V_{out4}} \right| \cdot \partial pH_{buffer} \\ & + \left| \frac{(pH_7 \cdot T_7 - pH_4 \cdot T_4)(V_{out7} + V_{out4})}{(V_{out7} - V_{out4})^2} \right| \cdot \partial V_{out} \end{aligned} \quad (6.7)$$

En la Tabla 6.2 se muestra el valor de las incertidumbres de los parámetros de medición y de las incertidumbres relevadas.

<b>Incertidumbres y puntos de trabajo</b>	
$V_{out7} = 300 \text{ mV}$	$\partial V_{out} = 3600/1024 \text{ mV}$
$V_{out4} = 1800 \text{ mV}$	
$T_7 = 25 \text{ }^\circ\text{C}$	$\partial T = 0.25 \text{ }^\circ\text{C}$
$T_4 = 25 \text{ }^\circ\text{C}$	
$pH_7 = 7 \text{ pH}$	$\partial pH_{buffer} = 0.01 \text{ pH}$
$pH_4 = 4 \text{ pH}$	
$\alpha = -0.5963$	$\partial\alpha = 0.0086$
$\beta = 2265.9$	$\partial\beta = 9.4090$

**Tabla 6.2: Incertidumbres y puntos de trabajo**

A partir de los cálculos anteriores se puede obtener la incertidumbre tipo B tanto para pH 7 como para pH4.

$$\begin{aligned} \partial pH_B &= 0.088 \text{ pH} & \text{para } pH &= 4 \\ \partial pH_B &= 0.050 \text{ pH} & \text{para } pH &= 7 \end{aligned}$$

### Incertidumbre combinada

A partir de las incertidumbres tipo A y B se obtiene la incertidumbre combinada como:

$$\partial pH = \sqrt{\partial pH_A^2 + \partial pH_B^2} \quad (6.8)$$

Con esto se tienen las incertidumbres para pH = 4 y para pH = 7:

$$\begin{array}{ll} \hat{\partial}pH = 0.088 \text{ pH} & \text{para pH} = 4 \\ \hat{\partial}pH = 0.050 \text{ pH} & \text{para pH} = 7 \end{array}$$

Utilizando un factor de cobertura  $K = 2$  para obtener las incertidumbres con un nivel de confianza del 95%, se obtiene:

$$\begin{array}{ll} \hat{\partial}pH = 0.177 \text{ pH} & \text{para pH} = 4 \\ \hat{\partial}pH = 0.100 \text{ pH} & \text{para pH} = 7 \end{array}$$

Teniendo en cuenta el peor caso posible, cuando el pH es más bajo, se llega a la conclusión que la incertidumbre asociada a la medida de pH del dispositivo construido es:

$$\hat{\partial}pH = 0.2 \text{ pH}$$

# Capítulo 7

## Conclusiones

### 7.1. Introducción

En este capítulo se comentaran los objetivos cumplidos, los resultados conseguidos, los principales problemas encontrados que pudieron ser resueltos y algunas ideas de posibles mejoras para futuro.

### 7.2. Objetivos Cumplidos

El principal y primer objetivo era lograr obtener un prototipo, que fuese capaz de medir temperatura y pH dentro de un ambiente industrial.

En el laboratorio se probó y utilizó toda la funcionalidad del sistema. Se lograron ajustar los parámetros del mismo y se obtuvieron buenos resultados en el sistema de transmisión. Esto permitió pasar a la siguiente etapa de pruebas en planta.

En la planta industrial, se probó durante una semana el sistema de transmisión inalámbrica junto con la medida de temperatura, cumpliéndose los requerimientos impuestos por el cliente, tanto en la tasa de paquetes perdidos, como en la precisión de la medida de temperatura. Para el sensor de pH no se llevo a probar el sistema en el fulón, debido a complicaciones en el montaje y falta de tiempo (además de que no se contaba con un electrodo industrial para poder montarlo). Lo que si se realizó en la planta fue una calibración del sistema de medida de pH, como un acercamiento al sistema de medición completo.

Luego de estos ensayos se recibieron las opiniones del inversor, el cual estuvo conforme con el desempeño del sistema y expresando su deseo de extender las pruebas para agregar mas nodos sensores al sistema.

El costo del prototipo realizado se aproxima a los 1500U\$S en materiales, mas unos 5000U\$S de horas hombre (a lo largo del año). Esto contrasta con el costo de seguir agregando nodos, que ronda los 400U\$S por nodo (dependiendo un poco del electrodo utilizado para el pH-metro). Además la escalabilidad del mismo, permite que el costo de horas hombre para el aumento de nodos sea mucho menor, porque el desarrollo para un nodo sensor es extensible a los demás.

## 7.3. Resultados Conseguídos

Para los resultados conseguidos, se tomo en cuenta, tanto los requerimientos que nos impuso el inversor, como los propios para hacer el proyecto viable, en la tabla X se observan los mismo

<b>Requerimientos vs. Resultados</b>	
<b>Sistema de transmisión confiable</b>	
Tasa de pérdidas de paquetes menores al 10%	FALTAN PRUEBAS
<b>Operación del sistema</b>	
Modificación de parámetros en línea	
Visualización	Interfaz con el usuario
Registro de medidas	
<b>Medidas</b>	
Entorno de pH: $3 \leq \text{pH} \leq 7$	$2.5 \leq \text{pH} \leq 8$
Incertidumbre de pH: $\text{pH} \leq 0.1$	0.2 pH
Entorno de temperatura: $15 \leq \text{pH} \leq 45 \text{ }^\circ\text{C}$	$0 \leq \text{pH} \leq 70 \text{ }^\circ\text{C}$
Incertidumbre de temperatura: $\leq 0.5 \text{ }^\circ\text{C}$	$\leq 0.5 \text{ }^\circ\text{C}$
<b>Autonomía</b>	
Autonomía del nodo sensor: 2 meses	Estimación mayor a 2 meses
<b>Pruebas en planta</b>	
Temperatura	3 días ininterrumpidos
pH	FALTA CALIBRAR
<b>Escalabilidad</b>	
Fácil aumento en la cantidad de nodos	Se cumplió
<b>Costo</b>	
Costo del prototipo $\leq 1500 \text{ U}\$\$$	Inferior a $\text{U}\$\$1000$

**Tabla 7.1: Requerimientos vs. Resultados**

Se logró un sistema para medir temperatura y pH inalámbrico. Esto permitió la instalación en una aplicación que requería de las medidas sin tener que cablearse los sensores.

Las medidas se muestran en un PC que también guarda los datos para su posterior procesamiento. Además posibilita modificar parámetros del sistema (tiempos de muestreo de pH, temperatura y calibración de pH). Las medidas de temperatura y pH pueden ser tomadas en intervalos definidos por el usuario y pueden ser cambiados según los requerimientos del momento.

El nodo sensor comunica las medidas inalámbricamente. La cantidad máxima de nodos sensores esta limitado en 64, trabajando todos con el mismo programa.

Las incertidumbres en las medidas están en los rangos requeridos para la aplicación.

## 7.4. Principales aportes

### Para el cliente (inversor)

Se logro obtener medidas de temperatura y pH inalámbricamente. Esto permite tener información en tiempo real de las variables importantes del proceso.

Además, este sistema permite cambiar el procedimiento con que relevaban las medidas que consistía en detener el proceso y tomar una muestra.

Automatiza el proceso del registro de datos para su posterior análisis.

El sistema en su conjunto ayuda a automatizar una parte del proceso, mejorándolo y eliminando así posibles errores de operación.

### Para la Universidad:

La integración de las medidas de pH y temperatura a los Motes. Logrando su utilización en una mayor variedad de aplicaciones.

Prueba del sistema en un ambiente industrial. Análisis de los datos en una aplicación que no se había probado antes.

Prueba de una nueva plataforma hardware. Del modelo de Mote (IRIS) usado en este proyecto no había experiencia en facultad.

## 7.5. Principales problemas encontrados

El primer problema encontrado fue que se estimaron mal los tiempos para la importación de los componentes. Hubieron muchos tiempos muertos debido a que las importaciones llevaron considerablemente más tiempo de lo estimado, se llegó al punto de tener que esperar por importaciones para poder seguir avanzando. Este inconveniente se solucionó parcialmente reordenando las tareas, ya que se trató de seguir con otras partes del proyecto que no necesitaban esos materiales. Este problema fue mayor en la importación de los Motes (al inicio del proyecto) ya que se tuvo el agregado de tener que efectuar las compras a través del inversor. Para otros materiales, más económicos, las compras se efectuaron directamente.

TinyOS es un sistema operativo en constante desarrollo. Además se estaba usando una plataforma hardware nueva (desarrollo de Crossbow del año 2007). Esto derivó en algunos problemas en la programación. A pesar de que no se encontraron errores en el código de los módulos y funciones dados por TinyOS si existían muchísimos errores u omisiones en la documentación de estos.

En el diseño de la electrónica para el pH-metro existieron varios problemas por lo que hubo que rehacer el PCB. El primer PCB que se mando construir no cumplía con los requerimientos de modo común en la entrada del amplificador de instrumentación. Para el diseño final esto fue solucionado. Otro de los problemas con la electrónica del pH-metro fue la inmunidad al ruido electromagnético. Para esto se colocaron capacitores de desacople y posteriormente un filtro a la salida del circuito. A pesar de no eliminar completamente el ruido se logró bajar lo suficiente para que este fuese del mismo orden que el error de resolución de la entrada del ADC.

## 7.6. Puntos a mejorar

Se estimó un tiempo de vida de más de 2 meses. Como fue presentado en el capítulo de validación, el componente que más afectaba al consumo era la radio. El apagado de la misma durante más tiempo disminuía el consumo pero aumentaba el tiempo que demoraban ciertos datos en llegar al nodo sensor, como lo era los tiempos de muestreo y la orden de calibración.

De no necesitar que estos valores sean transmitidos inalámbricamente (por ejemplo con dip switches en la electrónica) entonces la radio podría estar mucho más tiempo apagada.

La electrónica del pH-metro sigue siendo sensible a interferencias. Durante la etapa de pruebas se comprobó que estas eran inducidas en el circuito a través del cable del electrodo de pH. Acortando el cable se puede disminuir la incidencia del ruido en el pH-metro. Cabe aclarar que esto no se realizó debido a que el electrodo utilizado fue prestado por el cliente.

El pH-metro implementa una compensación de la caída en la deriva del electrodo, de forma de usar el máximo rango posible a la salida del mismo. Además existe una pequeña variación en la pendiente de la curva voltaje (pH) lo hace que al cabo de un tiempo prolongado que el rango del pH-metro sea menor. De querer resoluciones mayores y tiempos de vida aún más grandes se necesitaría implementar una compensación de la caída de esta pendiente. En un circuito similar el de esta aplicación, esto se solucionaría colocando un potenciómetro digital en la resistencia que ajusta la ganancia del operacional de instrumentación.

# ANEXOS

# Anexo I

## Electrodo ISFET<sup>18 19</sup>

### I.1. Generalidades

Los electrodos ISFET miden, por medio de transistores, la perturbación que sufre el campo eléctrico como consecuencia de la presencia de determinados iones en la sustancia a medir. Esta tecnología es similar a la del transistor MOSFET (del inglés Metal Oxide Semiconductor Field Effect Transistor). Esto se muestra esquemáticamente en la Figura I.1.

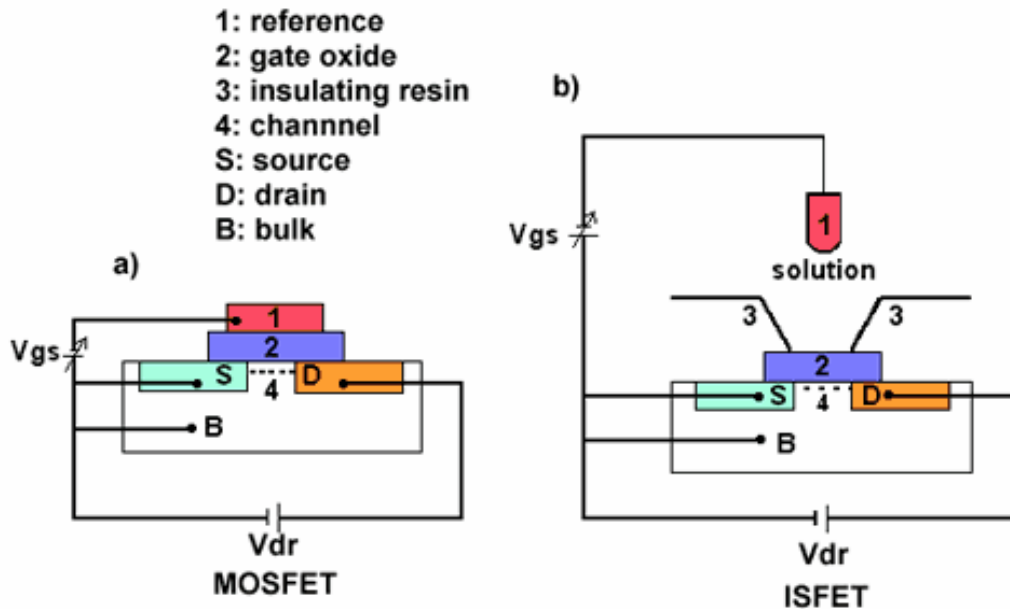


Figura I.1: Transistor MOSFET (a). Electrodo ISFET (b)<sup>20</sup>

En el electrodo ISFET se ha sustituido el metal del gate por la solución a analizar. Además el óxido de la compuerta ya no es de  $\text{SiO}_2$  sino que es un óxido de  $\text{Si}_3\text{Na}_4$ , porque tiene mejor sensibilidad al pH y peor sensibilidad a iones de  $\text{Na}^+$  y  $\text{K}^+$ . La sensibilidad de estos dispositivos es del entorno de 55 mV/pH.

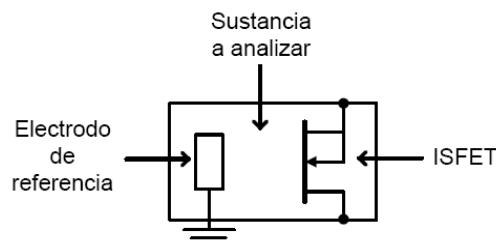


Figura I.2: Modelo del electrodo ISFET



## I.2. Ecuación característica

Del transistor MOSFET se obtiene la ecuación de la corriente de canal en modo triodo:

$$I_d = \frac{\mu \cdot C_{ox} \cdot W}{L} \cdot \left( (v_{gs} - V_t) \cdot v_{ds} - \frac{v_{ds}^2}{2} \right) \quad \text{si } v_{ds} < v_{gs} - V_t \quad (7.1)$$

$\mu$ ,  $C_{ox}$ ,  $W$  y  $L$ : Constantes características del transistor.

$v_{gs}$ : Diferencia de potencial entre gate y source.

$v_{ds}$ : Diferencia de potencial entre drain y source.

$V_t$ : Voltaje umbral

El voltaje umbral  $V_t$  es el valor de  $V_{gs}$  en el cual un número suficiente de electrones móviles se acumula en la región del canal para formar un canal conductor. Este parámetro es sumamente importante porque es el que tiene la sensibilidad en función del pH para un electrodo del tipo ISFET.

De la teoría de centros activos se observa que el voltaje umbral puede escribirse como:

$$V_t = E_{ref} + \psi_0 - \chi^{sol} + \Phi_{Si} + \frac{Q_{ox} + Q_{SS}}{C_{ox}} + \frac{Q_B}{C_{ox}} - 2\psi_B \quad (7.2)$$

Los términos anteriores, de la ecuación del voltaje de umbral son, a excepción de  $\psi_0$ , todos constantes dependientes de la solución en que se está inmerso y de la composición del ISFET. La relación de  $\psi_0$  con el pH se encuentra en la siguiente ecuación:

$$\psi_0 = 2.303 \cdot \frac{k \cdot T}{q} \cdot \frac{\beta}{\beta + 1} \cdot (pH_{pzc} - pH) \quad (7.3)$$

$pH_{pzc}$ : constante en la que el pH en el punto en el que la carga neta debido a la carga neta de los centros activos en la superficie es nula

$\beta$ : constante que depende de la capacidad de reacción de los centros activos del óxido.

$q$ : carga del electrón.

$k$ : constante de Boltzman

$T$ : temperatura en Kelvin.

Con las ecuaciones anteriores, calibrando el sistema con muestras buffer, podemos empíricamente determinar el valor de las constantes, a priori desconocidas, para lograr tener un dispositivo de medida basado en ISFET.

### I.3. Circuito de medida para determinar el pH

El método que se utiliza es llamado método de medidas a corriente constante. Este consiste en realizar medidas del potencial de puerta  $V_{gs}$ , manteniendo constante la corriente  $I_d$  y el potencial  $V_{ds}$ .

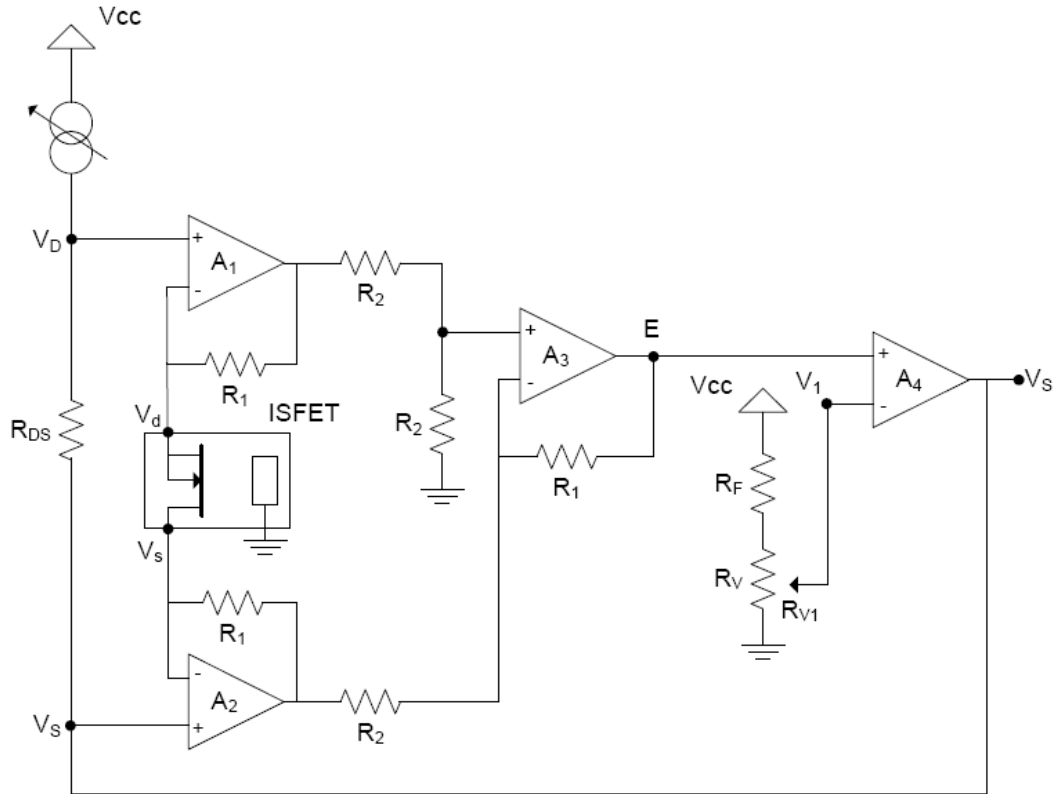


Figura I.3: Esquema de medida a corriente constante

La curva de sensibilidad es:

$$S = \frac{\partial V_{gs}}{\partial pH} = \frac{\partial V_t}{\partial pH} = \frac{\partial \psi_0}{\partial pH} \quad (7.4)$$

De la ecuación anterior se aprecia que la sensibilidad viene dada directamente por una tensión.

## I.4. Aplicación Práctica

Con el diseño de la Figura I.4 es posible determinar el voltaje de pH.

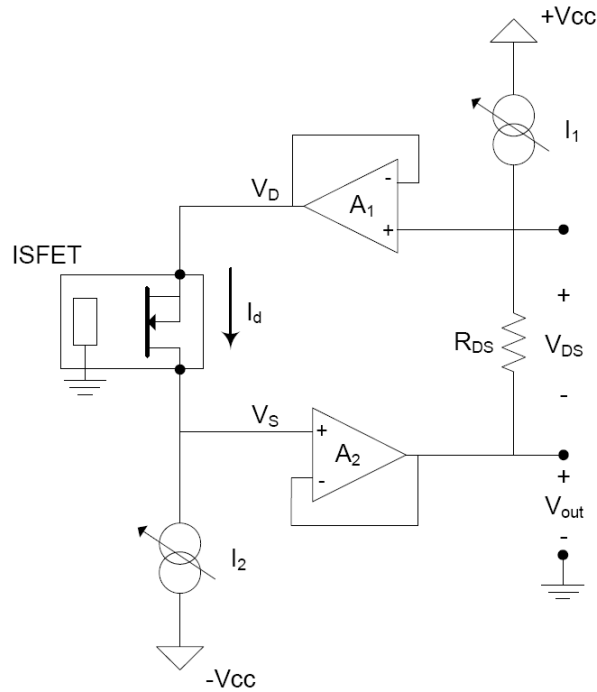


Figura I.4: Esquema para la medición del voltaje de pH

Donde los valores de  $I_2 = 100\mu\text{A}$ ,  $V_{DS} = 500\text{mV}$  y  $R_{DS} = 5\text{k}\Omega$ .

$$I_d = \beta \cdot \left( (v_{gs} - V_t) \cdot v_{ds} - \frac{v_{ds}^2}{2} \right) \quad (7.5)$$

$$v_{gs} = -v_s = -V_{out} \quad (7.6)$$

$$V_{out} = -\frac{V_t}{\beta} - \frac{1}{\beta} \cdot \left( \frac{I_d}{v_{ds}} + \frac{v_{ds}}{2} \right) \quad (7.7)$$

$$V_{out} = -\frac{V_t}{\beta} - \frac{1}{\beta} \cdot \left( \frac{I_2}{I_1 R_{DS}} + \frac{I_1 R_{DS}}{2} \right) \quad (7.8)$$

De la ecuación anterior se puede decir que la tensión de salida es directamente proporcional a la tensión de umbral y por ende al pH, cuando los ISFET trabajan en zona lineal, puesto que los demás parámetros son constantes.

## Anexo II

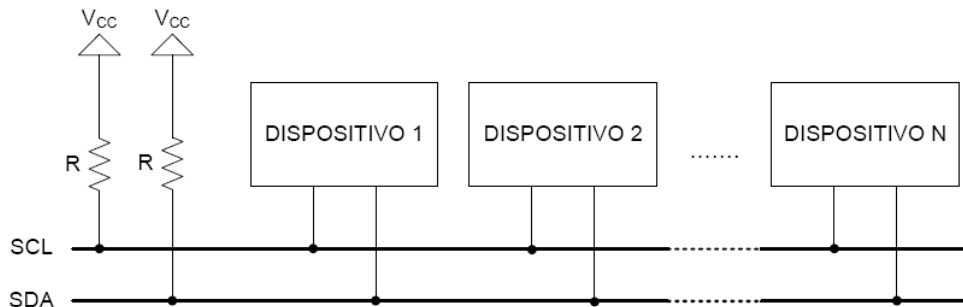
# Protocolo de comunicación I2C<sup>21</sup>

El bus I2C es un protocolo maestro-esclavo diseñado por Philips para facilitar la comunicación entre microcontroladores y sus periféricos. En el bus I2C todos los dispositivos comparten el canal de comunicaciones.

El bus I2C utiliza dos líneas para una comunicación establecer bidireccional half-duplex y una tercera línea que utiliza como referencia. La comunicación de datos del bus I2C es en serial y sincrónica. Una de las señales del bus marca el tiempo que sincronizan al sistema (SCL) y la otra se utiliza para intercambiar datos (SDA).

### II.1. Conexión para la comunicación I2C

En la Figura II.1 se muestra un esquema del bus I2C.

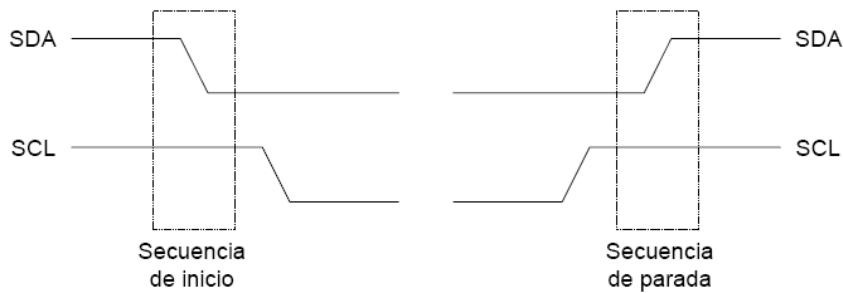


**Figura II.1: Diagrama de conexión de una comunicación I2C**

Las líneas SDA y SCL son del tipo “open-drain” por lo que se deben colocar resistencias de “pull-up”. Esto permite conectar al bus en paralelo múltiples dispositivos. La cantidad máxima de dispositivos que se puede poner en el bus es de 128 debido a la cantidad de bits utilizados para la dirección del dispositivo (7 bits). También existe un protocolo I2C con direccionamiento de 10 bits que es inter operable con el anterior, pero que no se encuentra actualmente ampliamente utilizado.

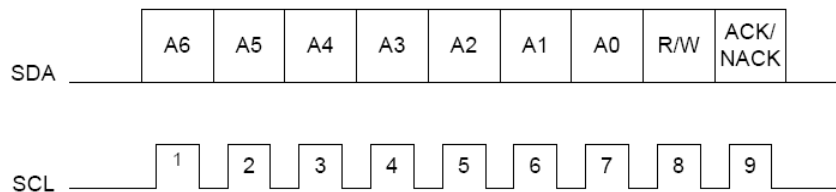
### II.2. Funcionamiento del protocolo I2C

El bus se encuentra libre cuando ambas señales están en nivel lógico alto. Solo los dispositivos maestros están habilitados para iniciar la comunicación. Con el bus en estado libre cualquier dispositivo maestro puede ocuparlo, estableciendo la condición de inicio (start) como se muestra en la Figura II.2.



**Figura II.2: Secuencias de inicio y parada del protocolo I2C**

El primer byte que se transmite luego de la condición de inicio contiene siete bits que componen la dirección del dispositivo que se desea seleccionar, y un octavo bit que corresponde a la operación que se quiere realizar con él, lectura o escritura (R/W). Si el dispositivo que tiene esa dirección está presente en el bus contesta con un pulso bajo de un tiempo de bit como reconocimiento (ACK), ubicado inmediatamente luego del octavo bit que ha enviado el dispositivo maestro y se establece la comunicación. Si por algún motivo no se pudo procesar la transmisión, ya sea porque no se encuentra el receptor, o que por algún motivo no pueda procesar la transferencia, se envía un pulso alto de un tiempo de bit como no-reconocimiento (NACK). Esta secuencia se puede apreciar en la Figura II.3.



**Figura II.3: Secuencia de bits al comienzo de la comunicación**

Si el bit de lectura/escritura fue puesto en el nivel lógico bajo (escritura), el dispositivo maestro envía datos al dispositivo esclavo, siempre que continúe recibiendo señales de reconocimiento. La comunicación concluye estableciendo la secuencia de parada que se muestra en la Figura II.2.

Cuando el bit de lectura/escritura se coloca en el nivel lógico alto (lectura), el dispositivo maestro genera pulsos de reloj para que el esclavo pueda enviar los datos. Luego de cada byte recibido el dispositivo maestro genera un pulso de reconocimiento. Cuando ya no necesita recibir nada más del dispositivo esclavo, el maestro envía un bit NACK inmediatamente antes de la generar la condición de parada.

## Anexo III

# Funcionamiento llave digital

En la Figura III.1 se muestra un esquema de una compuerta de transmisión CMOS como la que se utiliza para apagar el circuito de medición de pH.

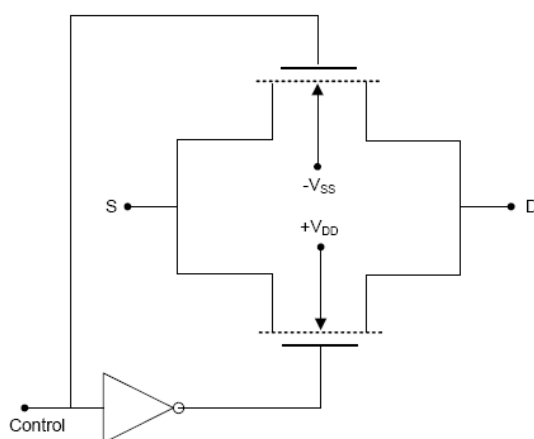


Figura III.1: Esquema de la llave TS3A4742

Como podemos ver en la Figura III.1 la llave consiste en colocar en paralelo dos transistores MOSFET, uno tipo p y otro tipo n. El inversor es utilizado para tener una sola señal de control para encender y apagar los transistores. Cuando el  $V_{Control}$  se encuentra en nivel bajo tanto el transistor NMOS como el CMOS se apagan. Cuando  $V_{Control}$  pasa a nivel alto ambos transistores se prenden.

El interruptor también podría implementarse con un solo transistor, aunque la mejora con dos, se encuentra en que la resistencia source-drain es mucho menor para estos, como se ve en la Figura III.2 (aunque en cualquier de los dos casos es muy pequeña).

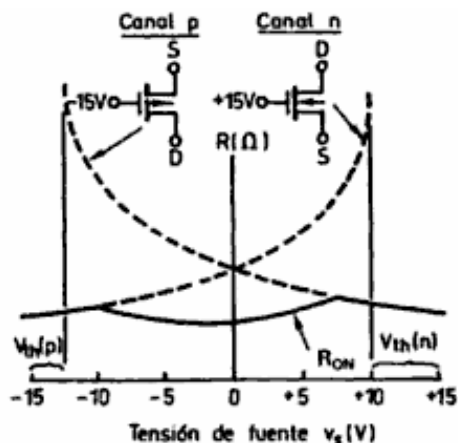


Figura III.2: Resistencia del interruptor

## Anexo IV

# Funcionamiento filtro pasabajos

En la Figura IV.1 se observa un filtro RC pasabajos de primer orden. La función de transferencia de dicho filtro es:

$$G(s) = \frac{V_0(s)}{V_1(s)} = \frac{1}{1 + sRC} \quad (7.9)$$

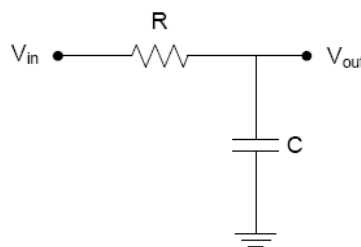


Figura IV.1: Filtro pasabajos de primer orden

Para voltaje continuo ( $s=0$ ) la ganancia es unitaria. La transferencia del filtro se muestra en la Figura IV.2.

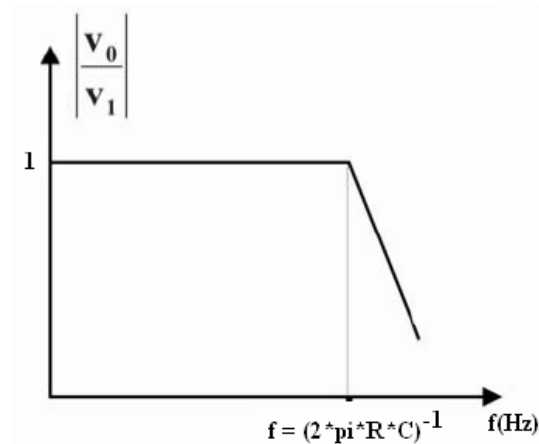


Figura IV.2: Transferencia de un filtro pasabajos de primer orden

El filtro pasabajos tiene un polo en  $f = (2 \cdot \pi \cdot R \cdot C)^{-1}$ . Para tener un buen rechazo al ruido de 50Hz, lo conveniente es tener el polo una década por debajo, a 5Hz o menos. Por otro lado, el fabricante del microprocesador del Mote, recomienda una impedancia de entrada de 10k $\Omega$  para un óptimo uso del convertor A/D. Finalmente, debe tenerse en cuenta que el rango de capacitores cerámicos que se encuentran en plaza es de hasta 2.2  $\mu\text{F}$ , pero además al momento de la compra lo que se consiguió fue capacitores de 1  $\mu\text{F}$ . Para solucionar este inconveniente se colocaron 4 capacitores en paralelo de 1  $\mu\text{F}$  lo que equivale a colocar uno de 4  $\mu\text{F}$ .

Utilizando  $R = 10 \text{ k}\Omega$  y  $C = 4 \text{ }\mu\text{F}$  se obtiene el polo en  $f = 4.0 \text{ Hz}$ .

Para ver el tiempo que demora en responder el filtro se calcula el tiempo de subida al 90% de la carga del capacitor, obteniéndose un tiempo menor a  $0.1\text{s}$  lo que resulta aceptable para esta aplicación.

$$\left( e^{\frac{-T}{RC}} - 1 \right) = 0.9 \quad \Rightarrow \quad T_{subida} = 2.3RC < 0.1s \quad (7.10)$$



# Anexo V

## Interfaz Mote - sensores

El nodo sensor en su conjunto consta básicamente de tres grandes bloques independientes (salvo en la alimentación que es común): el Mote, el circuito de medición de temperatura y el circuito de medición de pH.

Debido a que es necesaria la interacción de los circuitos con el Mote, se hace indispensable el conexionado de los mismos. Las señales a intercambiar son todas con el microprocesador del Mote (a excepción de la alimentación eléctrica que se cablea directamente de las pilas a través del conector de 51 pines). Para el integrado de temperatura se utiliza el puerto de comunicación I2C y para obtener la señal de pH y se utilizan dos entradas analógicas, una salida digital y el puerto I2C.

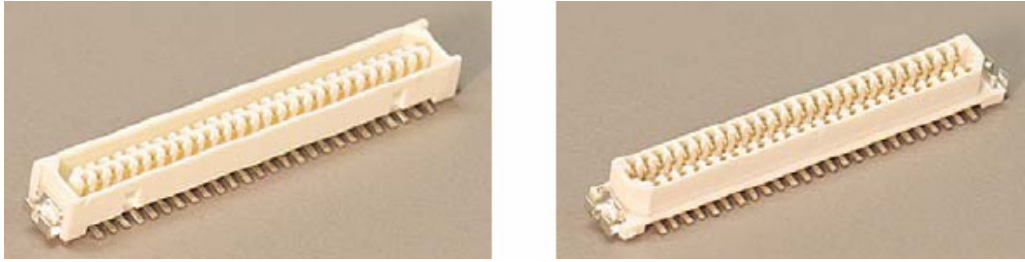
### V.1. Conector de expansión de 51 pines

En la Figura V.1 se aprecia el conector de 51 pines, este se encuentra en la parte inferior del Mote. Como ya se vio en la Figura 2.2 de la sección 2.1.7 el conector está íntegramente cableado al microprocesador del Mote, menos la alimentación.



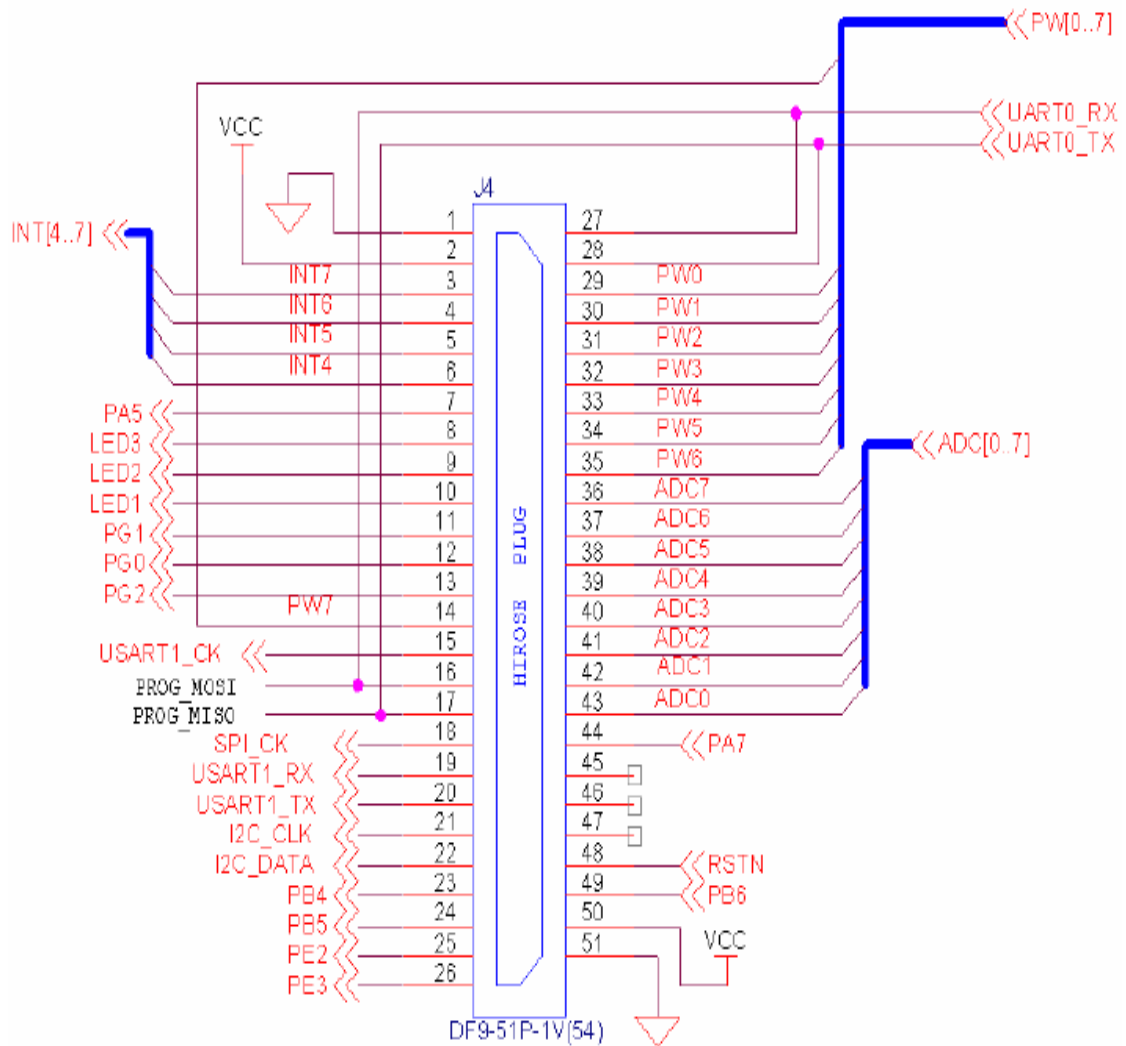
**Figura V.1: Foto del conector de expansión**

Los conectores de expansión macho y hembra se observan en la Figura V.2. El macho es el utilizado por el Mote (Hirose DF-51P-1V(54)), por lo que fue necesario conseguir el conector hembra (Hirose DF9-51S-1V(54)), para realizar un circuito impreso con las señales de interés para la aplicación.



**Figura V.2: Conector macho (izquierda) y conector hembra (derecha)**

En el esquemático de la Figura V.3 se detalla el conexionado del conector de 51 pines con el Mote.



**Figura V.3: Esquemático del conector**

Los pines del conector utilizados son los que se muestran en la Tabla V.1.

Pines de interés del conector de expansión		
Pin	Nombre	Señal
1	GND	Tierra, referencia de las pilas
2	Vcc	Potencial de las pilas
21	I2C_CLK	Reloj para comunicación serial I2C
22	I2C_DATA	Línea de datos para comunicación I2C
29	PW0 <sup>7</sup>	GPIO <sup>8</sup> salida que activa el circuito de medición de pH
30	PW1	GPIO, reservada
42	ADC1	Entrada analógica para medir el voltaje de referencia
43	ADC0	Entrada analógica para medir la tensión de sensor de pH

Tabla V.1: Pines de interés del conector de expansión

## V.2. Conector Molex

Para derivar las señales hacia los circuitos de medida, se buscaron en plaza conectores que fueran fáciles de soldar a los impresos y de fácil armado de los cables. Con lo cual la mejor opción encontrada, fue la de conectores Molex de 2.54 mm que se muestran en la Figura V.4.



Figura V.4: Molex macho (izquierda) y Molex hembra (derecha)

Otra opción para los conectores podría haber sido conectores USB, pero como solo pueden tener 4 señales por conector se eligió por los anteriores.

Para los cables se utilizaron cables multifilar de 0.25 mm<sup>2</sup> de diámetro, porque estos hacen mejor contacto entre los conectores.

## V.3. Comunicación entre el nodo base y PC

Otro punto importante para el proyecto es la base de programación que hace de interfaz entre el Mote y la computadora. Esta base tiene principalmente dos funciones: programar los Motes y hacer de interfaz entre el nodo base y el PC. Esta comunicación serial es necesaria para poder transmitir los paquetes de datos, desde el Mote a la PC y viceversa que serán utilizados por la aplicación de usuario.

<sup>7</sup> El pin PW0 corresponde al puerto digital C0 del micro del Mote

<sup>8</sup> GPIO se refiere a Señales de entrada / salida de propósito general, llamado así por las siglas en inglés (General Purpose Input Output)

El modelo de la base de programación es el Mib510 y en la Figura V.5 se muestra una foto de la misma. Este dispositivo consiste en un conversor USB serial, en el cual el Mote se conecta a través de su conector de 51 pines, en la superficie de la misma. Con esto se logra además de comunicar datos, alimentar el Mote de la fuente del PC. Es por esto que no es necesario tener un programa de bajo consumo en el Mote base, pudiendo de esa manera tener siempre prendida la radio del mismo, a la escucha de paquetes.



**Figura V.5: Base de programación Mib510**

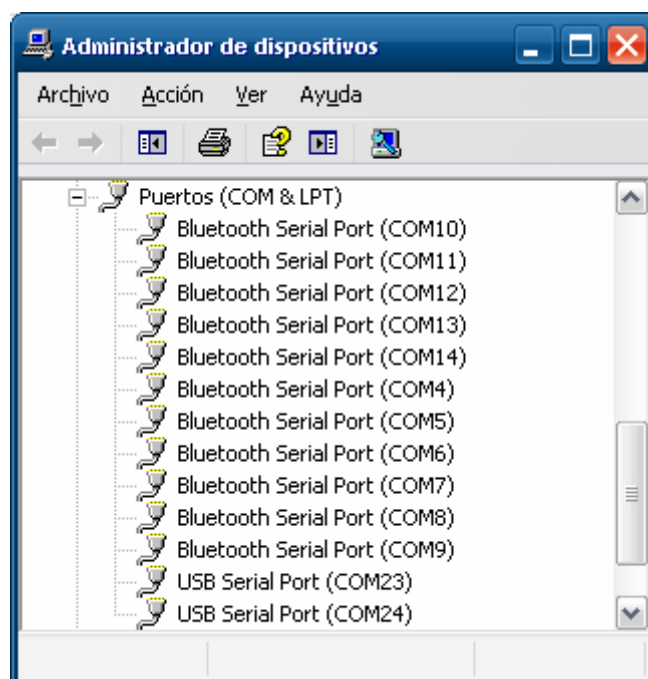
Anexo VI

Manual de usuario

## Anexo VII

# Manual de instalación

1. Instalar Java Runtime Edition: *jre-6u13-windows-i586-p-s*
2. Copiar los archivos de la carpeta win32 que se encuentra en el CD de instalación como se detalla a continuación.
  - De win32\bin en Archivos de programa\Java\jre6\bin
  - De win32\lib\ext en Archivos de programa\Java\jre6\lib\ext
  - De win32\lib\javax.comm.properties en Archivos de programa\Java\jre6\lib
3. Conectar el mote al puerto USB, esperar que se instale el controlador. Si no lo encuentra seleccionar para que busque el controlador en la carpeta USB\_Drivers que se encuentra en el disco de instalación,
4. Ver puerto USB a utilizar en System → Hardware → Administrador de dispositivos.



**Figura VII.1: Ventana de administrador de dispositivos de WinXP**

En este caso el puerto utilizado es el COM24.

Editar con un editor de texto el archivo AplicacionRII.bat para que utilice el

puerto encontrado en el punto 4.

Por ejemplo en este caso sería:

```
java -jar CommMotePC.jar -comm serial@COM24:57600
```

5. Colocar en un mismo directorio los archivos AplicacionRII.bat y CommMotePC.jar. Por ultimo correr el archivo AplicacionRII.bat

# Anexo VIII

## Costos

En la se detallan los costos del proyecto.

<b>Costos del proyecto</b>			
Descripción	Cantidad de unidades	Precio unitario (U\$S)	Costo (U\$S)
<b>Componentes pasivos</b>			
Resistencias (varias)	1	22,00	22,00
Capacitores (varias)	1	19,84	19,84
Presets multivuelta	2	1,00	2,00
<b>Conectores</b>			
Jumpers	1	1,82	1,82
Molex (2,4 y 6 pines sumados)	2	1,81	3,62
BNC	2	9,65	19,30
51 pines	5	3,16	15,80
<b>Integrados</b>			
Sensor de temperatura	3	2,28	6,84
Tensión de referencia	3	1,00	3,00
Llave digital	3	0,81	2,43
Potenciómetro digital	3	1,05	3,15
Amplificador de instrumentación	3	4,05	12,15
<b>PCBs</b>			
Placa temperatura	2	47,73	95,45
Placa conector 51 pines	2	43,18	86,36
Placa pH	2	51,14	102,27
<b>Plataforma hardware</b>			
Mote	3	115,00	345,00
Gateway	2	95,00	190,00
<b>Otros</b>			
Soluciones buffers	2	4,36	8,73
Agua destilada	1	0,64	0,64
<b>Gastos de envios</b>			
MiamiBox	5	5,50	27,50
MiamiBox	2	4,50	9,00
<b>TOTAL</b>			<b>976,91</b>

Tabla VIII.1: Costos del proyecto

Cabe aclarar que se desconocen los costos de importación de los Motes y Gateways debido a que fueron importados por el cliente.



# Referencias

- 
- <sup>1</sup> [http://www.biologia.edu.ar/tesis/forcillo/proceso\\_de\\_curtido.htm](http://www.biologia.edu.ar/tesis/forcillo/proceso_de_curtido.htm)
  - <sup>2</sup> “Manual del montador electricista”. T. Croft, C. C. Carr y J. H. Watt
  - <sup>3</sup> <http://data.energizer.com/PDFs/E91.pdf>
  - <sup>4</sup> <http://es.wikipedia.org/wiki/TinyOS>
  - <sup>5</sup> <http://www.omega.com/techref/ph.html>
  - <sup>6</sup> <http://www.ph-meter.info>
  - <sup>7</sup> [http://en.wikipedia.org/wiki/Glass\\_electrode](http://en.wikipedia.org/wiki/Glass_electrode)
  - <sup>8</sup> Imagen extraída de [http://en.wikipedia.org/wiki/Glass\\_electrode](http://en.wikipedia.org/wiki/Glass_electrode)
  - <sup>9</sup> <http://www.omega.com/techref/ph.html>
  - <sup>10</sup> [www.modelingandcontrol.com/repository/ISA55thIISPaperIIS09-P054.pdf](http://www.modelingandcontrol.com/repository/ISA55thIISPaperIIS09-P054.pdf)
  - <sup>11</sup> [http://www.mbhes.com/practical\\_ph\\_theory.htm](http://www.mbhes.com/practical_ph_theory.htm)
  - <sup>12</sup> [www.modelingandcontrol.com/repository/ISA55thIISPaperIIS09-P054.pdf](http://www.modelingandcontrol.com/repository/ISA55thIISPaperIIS09-P054.pdf)
  - <sup>13</sup> [http://us.mt.com/us/en/home/products/Laboratory\\_Analytics\\_Browse/Electrodes\\_family\\_page/ph\\_electrodes\\_family\\_page/Peek\\_electrodes\\_family\\_page/InLab4I3\\_SG\\_Premium\\_3-in-1\\_pH\\_Electrode\\_1.html](http://us.mt.com/us/en/home/products/Laboratory_Analytics_Browse/Electrodes_family_page/ph_electrodes_family_page/Peek_electrodes_family_page/InLab4I3_SG_Premium_3-in-1_pH_Electrode_1.html)
  - <sup>14</sup> <http://www.phmeters.com/faqs.html>
  - <sup>15</sup> <http://www.modelingandcontrol.com/repository/ISA55thIISPaperIIS09-P054.pdf>
  - <sup>16</sup> <http://ww1.microchip.com/downloads/en/DeviceDoc/DS-22096a.pdf>
  - <sup>17</sup> <http://focus.ti.com/lit/ds/symlink/ts3a4742.pdf>
  - <sup>18</sup> [http://www.tesisenxarxa.net/TESIS\\_UAB/AVAILABLE/TDX-1024105-224207//nz1de1.pdf](http://www.tesisenxarxa.net/TESIS_UAB/AVAILABLE/TDX-1024105-224207//nz1de1.pdf)
  - <sup>19</sup> Sacado de “Naturación urbana: cubiertas ecológicas y mejora medioambiental” Autor: J. Briz Editorial: Mundi-Prensa Libros, 2004
  - <sup>20</sup> Figura extraída de <http://csrg.ch.pw.edu.pl/tutorials/isfet/>
  - <sup>21</sup> “UM10204 I2C-bus specification and user manual” Rev.03