ANEXOS

ANEXO A. CALCULOS DE LA GANANCIA Y EL OFFSET

El objetivo de este anexo es detallar los cálculos de la ganancia y el offset del bloque de amplificación para cada uno de los sensores.

A la salida de la plataforma tenemos una señal de voltaje cuyos valores varían en función del sensor colocado y el valor de la magnitud física. Para maximizar la precisión del interfaz debemos hacer que el rango de voltaje a la salida del interfaz sea entre V_{DD} y GND. Esto se consigue restando a la señal de salida de la plataforma un nivel de continua(offset) y posteriormente aplicando una determinada ganancia. Para ello se dispone en el interfaz de dos amplificadores de instrumentación, el primero resta el offset y el segundo amplifica con la ganancia adecuada. Los valores del offset y de la ganancia son diferentes en función del tipo de sensor y dependen fundamentalmente de la relación entrada-salida y los rangos de medida de los sensores.

Debido a que los amplificadores de instrumentación utilizados tienen limitaciones, es necesario dejar márgenes de seguridad a la hora de calcular el offset y la ganancia para evitar saturar el sistema e introducir errores importantes. Así trabajaremos con dos tipos de márgenes:

- Los amplificadores de instrumentación no son *rail-to-rail* en la entrada. Como se explica en el apartado 2.1.3 de la memoria, para sensores no diferenciales es necesario tener en el segundo amplificador una pequeña tensión en su entrada V_{IN-}. Esto implica que en el segundo amplificador de instrumentación también se produzca una pequeña resta de voltaje, que debe ser tenida en cuenta para el cálculo del offset. Por ello a la hora de calcular el offset se deja un margen de 0.1 voltios.
- Idealmente la salida de los amplificadores de instrumentación puede tener cualquier valor entre V_{DD} y GND. En la realidad estos amplificadores se saturan

de forma que no consiguen salidas que lleguen hasta V_{DD}. Para evitar que se saturen se deja un margen de 0.1 voltios, de forma la ganancia se calcula para que el valor máximo sea V_{DD} menos el margen, en nuestro caso 3.2 voltios.

A continuación se expone el cálculo del offset y de la ganancia. El primer paso es calcular los valores de tensión máxima(V_{max}) y mínimo(V_{min}) a la salida de la plataforma. Estos se calculan gracias a las curvas de caracterización de los sensores y sus rangos de variación. El valor del offset(V_{offset}) se calcula como V_{min} menos el margen de 0.1 voltios anteriormente comentado. Como los potenciómetros digitales tienen un número finito de pasos, o valores de resistencia, elegimos el valor factible más cercano a V_{offset}. Este valor se denomina V_{offset}' y es el valor real del offset que se resta en el primer amplificador de instrumentación del interfaz. Una vez conocido V_{offset}', el valor de la ganancia(G), se calcula con la siguiente ecuación:

$$G = \frac{V_{DD} - 0.1}{V_{max} - V_{offset}'}$$

La ganancia del segundo amplificador viene determinada por el valor resistivo de un potenciómetro programable, al igual que con el offset elegimos el valor inferior más cercano a G como ganancia del sistema. Esta ganancia se denota como G'.

Sensor	Rango de medida	V_{min} (v)	V_{max} (v)	V_{offset}	V_{offset}'	G	G'
Hall	1.78-1.9 cm	0,3106	0,6597	0,2106	0,188	6,996	6,8114
Termopar	10-300 ºC	2,35E-4	0,00934	0	0	342	291,503
LDR	10-1000 lux	0,4074	3	0,3074	0,287	1,1795	1,1723
NTC	-20-80 ºC	0,1278	2,7876	0,0278	0,0268	1,1591	1,1584
Fotodiodo	100-2000 lux	0,0629	0,65795	0	0	4,48636	4,7708
Humedad	30-85 %	0,0647	2,8536	0	0	1,12139	1,1036

Tabla 1: Cálculos del offset y de la ganancia.

En la tabla 1 se muestran el valor de todas las variables intermedias, anteriormente explicadas, y de la ganancia(G') y del offset(V_{offset}') que finalmente se aplican para cada sensor. En el caso del termopar, fotodiodo y sensor de humedad, la señal de salida de la plataforma no pasa por la etapa de resta del offset, sino que van directamente a la etapa de ganancia. Por ello los valores de V_{offset}' en la tabla son 0.

ANEXO B. DISTRIBUCIÓN DE LAS SEÑALES DE CONTROL DEL INTERFAZ

El objetivo de este anexo es concretar como el microcontrolador, ayudado por los tres registros de desplazamiento, controlan toda la electrónica del interfaz. Para ello es necesario determinar todas las señales de control necesarias en los amplificadores de instrumentación, multiplexores, potenciómetros digitales y VFC. A continuación se muestran los esquemáticos de cada uno de los componentes utilizados en el diseño del interfaz, en los que se identifica los pines de control.

- Amplificadores de instrumentación. Los dos amplificadores de instrumentación elegidos en el diseño son INA327 de Texas Instruments. La figura 1 muestra su distribución de pines.

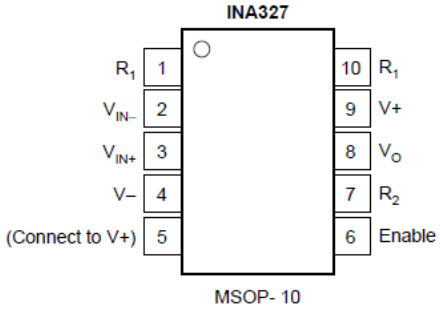

Figura1: Distribución de pines del INA327

El único pin de control de ambos amplificadores es el pin de habilitación(*enable)* que permite activar o desactivar el dispositivo. La señal de control es la misma para los dos amplificadores de instrumentación ya que tienen que estar activos durante el mismo periodo de tiempo, su notación es ENABLEAI.

- Potenciómetros digitales. En el diseño del interfaz se utilizan seis potenciómetros digitales. Cuatro de ellos son MAX5400(50 kΩ) y dos MAX5401(100 kΩ). La distribución de pines, el modo de funcionamiento e

incluso el *datasheet* es común para los dos, ya que sólo se diferencian en la resistencia nominal.

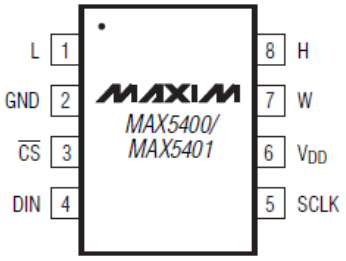

Figura 2: Distribución de pines del MAX5400/MAX5401

Estos potenciómetros tienen tres pines de control que sirven para programar digitalmente el valor de resistencia del potenciómetro. El pin CS sirve para habilitar la programación, DIN son los datos de entrada para configurar la palabra digital y SCLK es el reloj de programación. En la figura 3 se observa el cronograma de programación de los potenciómetros digitales.

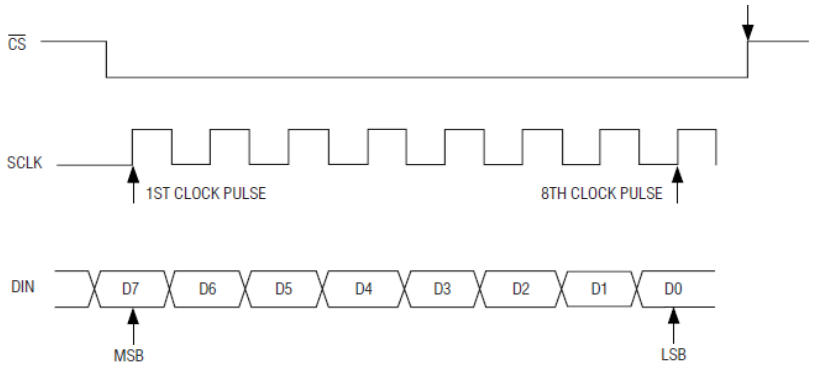

Figura 3: Cronograma de programación MAX5400/5401

Como se ha explicado en el apartado 2.2 de la memoria, el micro programa los potenciómetros en serie. Esto permite tener una línea común de datos para todos los potenciómetros que denota como DIN. El reloj utilizado es el proveniente del protocolo I2C, línea SCL. La entrada CS debe ser distinta para cada uno de los potenciómetros, se denotará CS1 en el caso del potenciómetro 1 y se numera sucesivamente los otros seis potenciómetros.

- VFC. En el interfaz se utiliza un conversor tensión frecuencia AD7740 de Analog Devices. En la figura 4 se observa su distribución de pines.

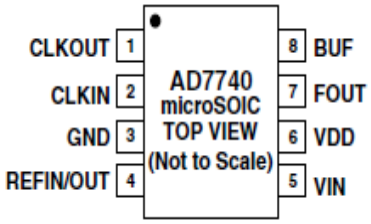

Figura 4: Distribución de pines AD7740

Este integrado no posee pin de habilitación, es decir no permite activar o desactivar externamente el dispositivo, una vez alimentado está activo. Por ello se alimenta con la segunda línea de alimentación(V_{DD2}) como se ha explicado en el apartado 2.1.6 de la memoria. El pin CLKIN es el de entrada de la señal de referencia del VFC, esta señal es generada por el microprocesador y se denota como CLKVFC.

- Multiplexores 2:1. En el interfaz se utilizan 5 multiplexores ADG719 de Analog Devices. En la figura 5 se muestra la distribución de pines del integrado escogido.

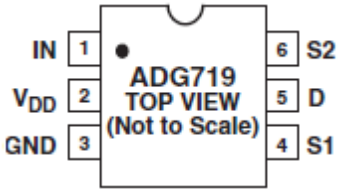

Figura 5: Distribución de pines del ADG719.

Como en el caso del VFC, los multiplexores ADG719 no poseen un pin de habilitación, por lo tanto para que no se encuentren todo el tiempo activos son alimentados por la segunda línea de alimentación(V_{DD2}). El pin IN es el que se encarga de seleccionar como salida(D), una de las entradas(S1,S2). Esta señal es controlada por los registros de desplazamiento. Su notación es MUX&IN donde & simboliza el número de multiplexor.

- Multiplexores 4:1. En el diseño del interfaz se han utilizado 4 multiplexores 4:1 ADG704 de Analog Devices. En la figura 6 se muestra la distribución de pines del integrado.

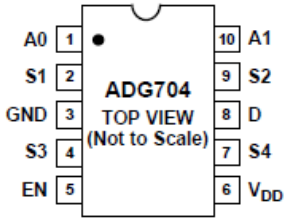

Figura 6: Distribución de pines del ADG704

Este integrado posee un pin de habilitación(EN) para activar/desactivar el dispositivo, esta señal de control se denota MUX&EN, donde & simboliza el número de multiplexor. También posee otros dos pines de control(A0,A1), con los que se selección la salida(D) entre las 4 entradas posibles(S1,S2,S3,S4).La notación para estas dos señales de control es MUX&A0, MUX&A1, donde & simboliza el número de multiplexor.

A continuación paso a describir el microcontrolador y los registros de desplazamiento que generan las señales de control de todos los integrados expuestos anteriormente.

- Registros de desplazamiento. En el diseño del interfaz se han utilizado tres registros de desplazamiento de 8 bits SN74HC594 de Texas Instruments para ayudar al micro a gestionar toda la electrónica. En la figura 7 podemos observar la distribución de pines de dichos registros.

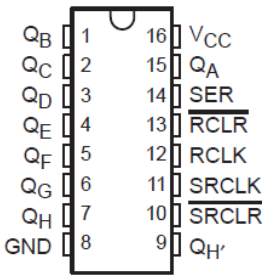

Figura 7: Distribución de pines del SN74HC594

Los pines SER, RCLR, RCLK, SRCLK, SRCLR son entradas del integrado que se utilizan para establecer el valor de cada una de las salidas del registro

de desplazamiento Q_A-Q_H. Las señales RCLR Y SRCLR sirven para limpiar el contenido del registro, como en nuestra aplicación no nos interesa limpiar el registro se conectan ambas a $V_{DD,}$ forzando un 1 lógico. Las señales SER RCLK y SRCLK son controladas por el microcontrolador, siguiendo el cronograma de la figura 8 para escribir adecuadamente en el registro. El pin $Q_{H'}$ permite encadenar los tres registros formando un registro de desplazamiento de 24 bits. Así sólo el primer registro utiliza la señal de datos del microcontrolador SER y las señales RCLK y SRCLK son comunes para los tres registros.

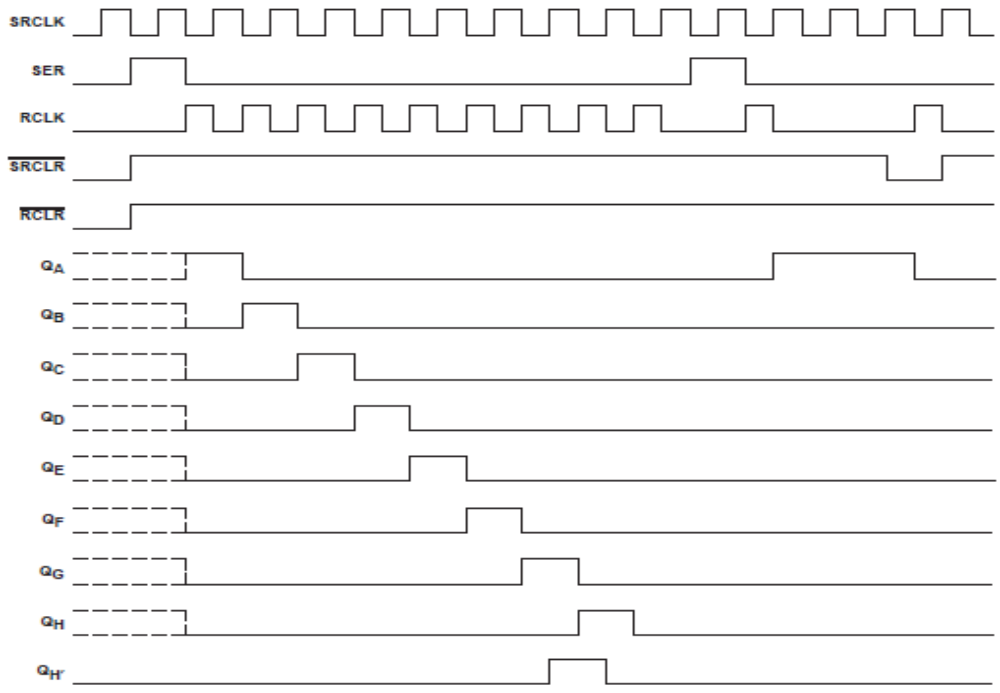

Figura 8: Cronograma de escritura los SN74HC594

El micro escribe en el registro para realizar una conversión serie/paralelo, de forma que las salidas de los registros Q_A-Q_H se conviertan en señales de control de multiplexores y amplificadores de instrumentación. En la tabla siguiente se observa la relación entre las salidas de los registros de desplazamiento y las señales de control asociadas.

Número de registro	Pin registro	Señal de control
Registro 3	QH	MUX1IN
	QG	MUX2A0
	QF	MUX2A1
	QE	MUX2EN
	QD	MUX3A0
	QC	MUX3A1
	QB	MUX3EN
	QA	MUX4IN
Registro 2	QH	MUX5A0
	QG	MUX5A1
	QF	MUX5EN
	QE	MUX6A0
	QD	MUX6A1
	QC	MUX6EN
	QB	MUX7IN
	QA	MUX8IN
Registro 1	QH	P5
	QG	AIEN
	QF	MUX9IN
	QE	Sin uso
	QD	Sin uso
	QC	Sin uso
	QB	Sin uso
	QA	Sin uso

Tabla 1: Relación entre salidas de los registros y señales de control

- Microcontrolador. El encargado de gestionar las comunicaciones con el exterior programar la electrónica digital del interfaz es un microcontrolador MC9RS08KA8 de Freescale. En la figura 8 se observa la distribución de pines de dicho microcontrolador.

Figura 9: Distribución de pines del MC9RS08KA8

El micro se encarga de programar los potenciómetros digitales, configurar correctamente los registros, proporcionar la señal de referencia del VFC y de las comunicaciones con el exterior. La tabla 2 muestra la relación entre los pines del micro y las señales de control asociadas.

Pin del micro	Señal de control
PTA0	SRCLK
PTA1	RCLK
PTA2	SDA
PTA3	SCL
PTA4	RESET
PTA5	BKGD
PTB0	CS1
PTB1	CS2
PTB2	CS3
PTB3	CS4
PTB4	CLKVFC
PTB5	CS6
PTB6	DIN
PTB7	SER
PTC0	SDAI
PTC1	CS5
PTC2	IN7
PTC3	Sin uso

Tabla 2: Relación entre pines del micro y señales de control.

ANEXO C. PROTOCOLO I2C

Este anexo explica detalladamente las características del bus I2C que ha sido elegido como protocolo para comunicar el interfaz con el exterior. Está basado en el artículo desarrollado en la siguiente dirección de internet: http://robots-argentina.com.ar/Comunicacion_busI2C.htm , último acceso 21/08/2010.

INTRODUCCIÓN

El bus I2C, un estándar que facilita la comunicación entre microcontroladores, memorias y otros dispositivos con cierto nivel de "inteligencia", sólo requiere de dos líneas de señal y un común o masa. Fue diseñado a este efecto por Philips y permite el intercambio de información entre muchos dispositivos a una velocidad aceptable, de unos 100 Kbits por segundo, aunque hay casos especiales en los que el reloj llega hasta los 3,4 MHz.

La metodología de comunicación de datos del bus I2C es en serie y sincrónica. Una de las señales del bus marca el tiempo (pulsos de reloj) y la otra se utiliza para intercambiar datos.

DESCRIPCIÓN DE LAS SEÑALES

- SCL (*System Clock*) es la línea de los pulsos de reloj que sincronizan el sistema.
- SDA (*System Data*) es la línea por la que se mueven los datos entre los dispositivos.
- GND (Masa) común de la interconexión entre todos los dispositivos "enganchados" al bus.

Las líneas SDA y SCL son del tipo drenaje abierto, es decir, un estado similar al de colector abierto, pero asociadas a un transistor de efecto de campo (o FET). Se deben polarizar en estado alto (conectando a la alimentación por medio de resistores "*pull-up*")

lo que define una estructura de bus que permite conectar en paralelo múltiples entradas y salidas.

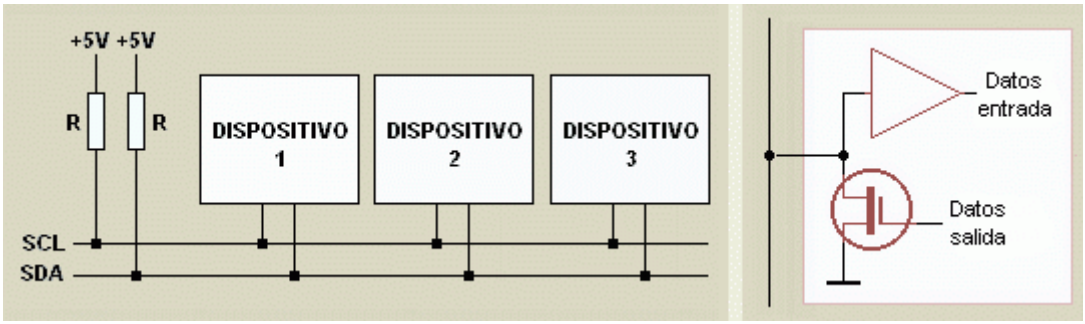

Figura1: Diagrama de interconexión del bus I2C.

El diagrama es suficientemente autoexplicativo. Las dos líneas del bus están en un nivel lógico alto cuando están inactivas. En principio, el número de dispositivos que se puede conectar al bus no tiene límites, aunque hay que observar que la capacidad máxima sumada de todos los dispositivos no supere los 400 pF. El valor de los resistores de polarización no es muy crítico, y puede ir desde 1K8 (1.800 ohms) a 47K (47.000 ohms). Un valor menor de resistencia incrementa el consumo de los integrados pero disminuye la sensibilidad al ruido y mejora el tiempo de los flancos de subida y bajada de las señales. Los valores más comunes en uso son entre 1K8 y 10K.

PROTOCOLO DE COMUNICACIÓN DEL BUS I2C

Habiendo varios dispositivos conectados sobre el bus, es lógico que para establecer una comunicación a través de él se deba respetar un protocolo. Digamos, en primer lugar, lo más importante: existen dispositivos maestros y dispositivos esclavos. Sólo los dispositivos maestros pueden iniciar una comunicación.

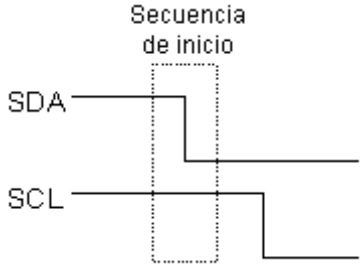

Figura 2: Secuencia de inicio del bus I2C.

La condición inicial, de bus libre, es cuando ambas señales están en estado lógico alto. En este estado cualquier dispositivo maestro puede ocuparlo, estableciendo la condición de inicio (*start*). Esta condición se presenta cuando un dispositivo maestro pone en estado bajo la línea de datos (SDA), pero dejando en alto la línea de reloj (SCL).

El primer byte que se transmite después 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).

Si el dispositivo cuya dirección corresponde a la que se indica en los siete bits (A0-A6) está presente en el bus, éste contesta con un bit en bajo, ubicado inmediatamente después del octavo bit que ha enviado el dispositivo maestro. Este bit de reconocimiento (ACK) en bajo le indica al dispositivo maestro que el esclavo reconoce la solicitud y está en condiciones de comunicarse. Aquí la comunicación se establece en firme y comienza el intercambio de información entre los dispositivos.

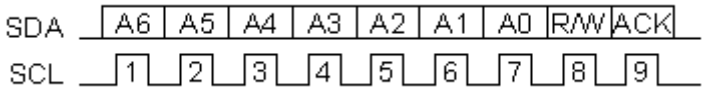

Figura 3: Secuencia de envío del primer byte.

Si el bit de lectura/escritura (R/W) fue puesto en esta comunicación a nivel lógico bajo (escritura), el dispositivo maestro envía datos al dispositivo esclavo. Esto se mantiene mientras continúe recibiendo señales de reconocimiento, y el contacto concluye cuando se hayan transmitido todos los datos.

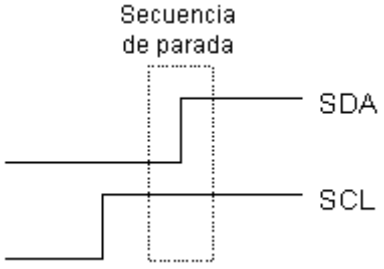

Figura 4: Secuencia de parada del bus I2C.

En el caso contrario, cuando el bit de lectura/escritura estaba a nivel lógico alto (lectura), el dispositivo maestro genera pulsos de reloj para que el dispositivo esclavo pueda enviar los datos. Luego de cada byte recibido el dispositivo maestro (quien está recibiendo los datos) genera un pulso de reconocimiento.

El dispositivo maestro puede dejar libre el bus generando una condición de parada (o detención; stop en inglés).

Si se desea seguir transmitiendo, el dispositivo maestro puede generar otra condición de inicio en lugar de una condición de parada. Esta nueva condición de inicio se denomina "inicio reiterado" y se puede emplear para direccionar un dispositivo esclavo diferente o para alterar el estado del bit de lectura/escritura.

DEFINICIÓN DE TÉRMINOS

- Maestro (*Master*): Dispositivo que determina los tiempos y la dirección del tráfico en el bus. Es el único que aplica los pulsos de reloj en la línea SCL. Cuando se conectan varios dispositivos maestros a un mismo bus la configuración obtenida se denomina "multi-maestro".
- Esclavo (*Slave*): Todo dispositivo conectado al bus que no tiene la capacidad de generar pulsos de reloj. Los dispositivos esclavos reciben señales de comando y de reloj generados desde el maestro.
- Bus libre (*Bus Free*): Estado en el que ambas líneas (SDA y SCL) están inactivas, presentando un estado lógico alto. Es el único momento en que un dispositivo maestro puede comenzar a hacer uso del bus.
- Comienzo (*Start*): Se produce cuando un dispositivo maestro ocupa el bus, generando la condición. La línea de datos (SDA) toma un estado bajo mientras que la línea de reloj (SCL) permanece alta.
- Parada (*Stop*): Un dispositivo maestro puede generar esta condición, dejando libre el bus. La línea de datos y la de reloj toman un estado lógico alto.
- Dato válido (*Valid Data*): Situación presente cuando un dato presente en la línea SDA es estable al tiempo que la línea SCL está a nivel lógico alto.

- Formato de Datos (*Data Format*): La transmisión de un dato a través de este bus consiste de 8 bits de dato (1 byte). A cada byte transmitido al bus le sigue un noveno pulso de reloj durante el cual el dispositivo receptor del byte debe generar un pulso de reconocimiento.
- Reconocimiento (*Acknowledge*): El pulso de reconocimiento, conocido como ACK (del inglés *Acknowledge*), se logra colocando la línea de datos a un nivel lógico bajo durante el transcurso del noveno pulso de reloj.
- Dirección (*Address*): Todo dispositivo diseñado para funcionar en este bus posee su propia y única dirección de acceso, preestablecida por el fabricante. Hay dispositivos que permiten definir externamente parte de la dirección de acceso, lo que habilita que se pueda conectar en un mismo bus un conjunto de dispositivos del mismo tipo, sin problemas de identificación. La dirección 00 es la denominada "de acceso general"; a ésta responden todos los dispositivos conectados al bus.
- Lectura/Escritura (Bit R/W): Cada dispositivo tiene una dirección de 7 bits. El octavo bit (el menos significativo) que se envía durante la operación de direccionamiento, completando el byte, indica el tipo de operación a realizar. Si este bit es alto el dispositivo maestro lee información proveniente de un dispositivo esclavo. Si este bit es bajo, el dispositivo maestro escribe información en un dispositivo esclavo.

LA COMUNICACIÓN EN MÁS DETALLE

Cuando el dispositivo maestro quiere comunicarse con un esclavo, produce una secuencia de inicio en el bus. La secuencia de inicio es una de las dos secuencias especiales que se han definido en el bus I2C; la otra es la secuencia de parada. Las secuencias de inicio y la de parada son especiales porque son los dos únicos casos en que se permite que la línea de datos (SDA) cambie cuando la línea de reloj (SCL) está alta. Cuando se están transmitiendo datos, la línea SDA debe permanecer estable, y jamás cambiar, mientras la línea SCL está alta. Las secuencias de inicio y de parada señalan el comienzo y el final de una transacción con los dispositivos esclavos.

Los datos se transfieren en secuencias de 8 bits. Estos bits se colocan en la línea SDA comenzando por el bit de más peso (o más significativo). Una vez puesto un bit en SDA, se lleva la línea SCL a alto. Debemos recordar que el chip no puede llevar la línea a un estado alto, en realidad, lo que hace es "soltarla", y el que la pone en nivel lógico alto es el resistor de polarización. Por cada 8 bits que se transfieren, el dispositivo que recibe el dato envía de regreso un bit de reconocimiento, de modo que en realidad por cada byte de dato se producen 9 pulsos sobre la línea SCL (es decir, 9 pulsos de reloj por cada 8 bits de dato). Si el dispositivo que recibe envía un bit de reconocimiento bajo, indica que ha recibido el dato y que está listo para aceptar otro byte. Si retorna un alto, lo que indica es que no puede recibir más datos y el dispositivo maestro debería terminar la transferencia enviando una secuencia de parada.

DIRECCIONAMIENTO DE DISPOSITIVOS EN EL BUS I2C

Lo más común en los dispositivos para el bus I2C es que utilicen direcciones de 7 bits, aunque existen dispositivos de 10 bits. Este último caso es raro.
Una dirección de 7 bits implica que se pueden poner hasta 128 dispositivos sobre un bus I2C, ya que un número de 7 bits puede ir desde 0 a 127. Cuando se envían las direcciones de 7 bit, de cualquier modo la transmisión es de 8 bits. El bit extra se utiliza para informarle al dispositivo esclavo si el dispositivo maestro va a escribir o va a leer datos desde él. Si el bit de lectura/escritura (R/W) es cero, el dispositivo maestro está escribiendo en el esclavo. Si el bit es 1 el maestro está leyendo desde el esclavo. La dirección de 7 bit se coloca en los 7 bits más significativos del byte y el bit de lectura/escritura es el bit menos significativo.

El hecho de colocar la dirección de 7 bits en los 7 bits más significativos del byte produce confusiones entre quienes comienzan a trabajar con este bus. Si, por ejemplo, se desea escribir en la dirección 21 (hexadecimal), en realidad se debe enviar un 42, que es un 21 desplazado un bit hacia arriba. También se pueden tomar las direcciones del bus I2C como direcciones de 8 bit, en las que las pares son de sólo escritura y las impares son de sólo lectura. Para dar un ejemplo, el integrado de brújula magnética CMPS03 es fijado en fábrica en la dirección 0xC0 ($C0). La dirección 0xC0 se utiliza para escribir en él y la dirección 0xC1 es para leer de él.

PROTOCOLO DE PROGRAMACIÓN PARA EL BUS I2C

Lo primero que ocurre en un bus I2C es que el dispositivo maestro envía una secuencia de inicio. Esto alerta a los dispositivos esclavos, poniéndolos a la espera de una transacción. Éstos quedan atentos para ver si se trata de una solicitud para ellos. A continuación el dispositivo maestro envía la dirección de dispositivo. El dispositivo esclavo que posee esa dirección continuará con la transacción, y los otros ignorarán el resto de los intercambios, esperando la próxima secuencia de inicio.

Habiendo direccionado ya el dispositivo esclavo, lo que debe hacer ahora el maestro es enviar la ubicación interna o número de registro desde el que desea leer o al que va a escribir. La cantidad depende, obviamente, de qué dispositivo es y de cuántos registros internos posee. Algunos dispositivos muy simples no tienen ninguno, pero la mayoría sí los poseen.

Siguiendo con el ejemplo del CMPS03, éste posee 16 ubicaciones internas, numeradas desde el 0 al 15. Otro dispositivo, el medidor ultrasónico de distancia SRF08, tiene 36 registros.

Una vez que el maestro ha enviado la dirección del dispositivo en el bus I2C y la dirección del registro interno del dispositivo, puede enviar ahora el byte o bytes de datos. El dispositivo maestro puede seguir enviando bytes al esclavo, que normalmente serán puestos en registros con direcciones sucesivas, ya que el esclavo incrementa automáticamente la dirección del registro interno después de recibir cada byte. Cuando el maestro ha terminado de escribir datos en el esclavo, envía una secuencia de parada que concluye la transacción.

ESCRITURA EN UN DISPOSITIVO ESCLAVO

- 1. Enviar una secuencia de inicio.
- 2. Enviar la dirección de dispositivo con el bit de lectura/escritura en bajo.
- 3. Enviar el número de registro interno en el que se desea escribir.

- 4. Enviar el byte de dato.
- 5. [Opcionalmente, enviar más bytes de dato].
- 6. Enviar la secuencia de parada.

Como ejemplo, veamos un SRF08, que tiene una dirección de bus fijada en fábrica de 0xE0. Para comenzar una medición de distancia con el SRF08 se debe escribir 0x51 en el registro de comando, ubicado en la dirección interna 0x00. La secuencia es la que sigue:

- Enviar una secuencia de inicio.
- 2. Enviar 0xE0 (La dirección de dispositivo del SRF08 con el bit de lectura/escritura en bajo).
- 3. Enviar 0x00 (dirección interna del registro de comando).
- 4. Enviar 0x51 (el comando para comenzar la medición del SRF08).
- 5. Enviar la secuencia de parada.

LECTURA DESDE UN DISPOSITIVO ESCLAVO

Esta operación es algo más complicada, pero no demasiado. Antes de leer datos desde el dispositivo esclavo, primero se le debe informar desde cuál de sus direcciones internas se va a leer. De manera que una lectura desde un dispositivo esclavo en realidad comienza con una operación de escritura en él. Es igual a cuando se desea escribir en él: Se envía la secuencia de inicio, la dirección de dispositivo con el bit de lectura/escritura en bajo y el registro interno desde el que se desea leer. Ahora se envía otra secuencia de inicio nuevamente con la dirección de dispositivo, pero esta vez con el bit de lectura/escritura en alto. Luego se leen todos los bytes necesarios y se termina la transacción con una secuencia de parada.

Volviendo al ejemplo del módulo de brújula CMPS03, veamos cómo se lee el registro de ángulo:

- 1. Enviar una secuencia de inicio.

- 2. Enviar 0xC0 (La dirección de dispositivo del CMPS03 con el bit de lectura/escritura en bajo).
- 3. Enviar 0x01 (dirección interna del registro de ángulo en valor 0-255) .
- 4. Enviar una secuencia de inicio (inicio reiterado).
- 5. Enviar 0xC1 (La dirección de dispositivo del CMPS03 con el bit de lectura/escritura en alto).
- 6. Leer un byte de dato desde el CMPS03.
- 7. Enviar la secuencia de parada.

La secuencia se verá así:

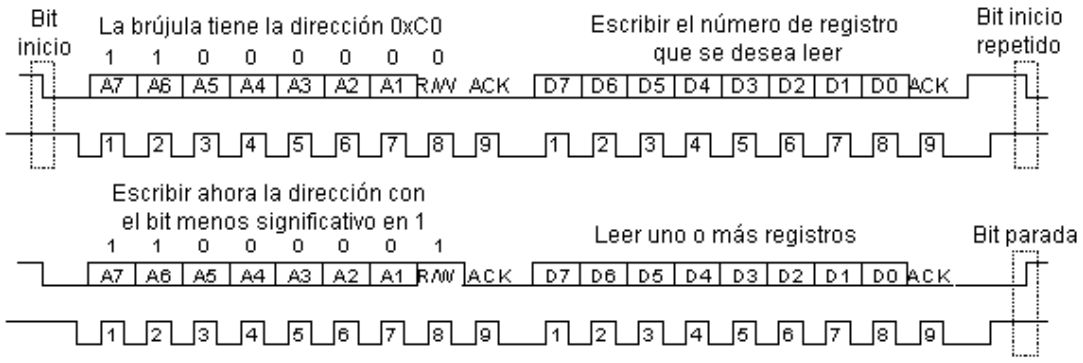

Figura 5: Ejemplo de secuencia de lectura desde un dispositivo esclavo

UN CASO UN POCO MÁS COMPLICADO

Esto es todo cuando se trata de comunicaciones simples, pero debemos considerar una posible complicación: Cuando el dispositivo maestro está leyendo desde el esclavo, quien pone los datos en la línea SDA del bus es el dispositivo esclavo, y el maestro es el que controla el pulso de reloj. ¿Qué pasa si el esclavo no está listo para enviar un dato? Con dispositivos como una EEPROMs esto no sería problema, pero si el dispositivo esclavo es un microprocesador, que tiene otras tareas que realizar, pueden surgir inconvenientes.

Para atender la transacción, el microprocesador debe pasar a una rutina de interrupción, guardar sus registros de trabajo, determinar qué dirección desea leer el dispositivo

maestro, obtener el dato y ponerlo en su registro de transmisión. Esto puede llevar varios microsegundos, lo que implica que el dispositivo maestro podría estar enviando pulsos de reloj ciegamente por la línea SCL sin que el dispositivo esclavo pueda responderle. El protocolo I2C ofrece una solución para esto: el esclavo puede mantener la línea SCL en bajo. A esto se le llama estiramiento del reloj. Cuando el esclavo recibe el comando de lectura lo primero que hace es poner la línea de reloj en bajo. Entonces sí, obtiene el dato solicitado, lo pone en el registro de transmisión, y recién entonces libera la línea de reloj, que pasará de inmediato a alto debido al nivel que aporta el resistor de polarización.

Desde el punto de vista del dispositivo maestro, éste tratará de enviar el primer pulso de reloj para la lectura de datos liberando la línea SCL para que pase a alto, pero antes de continuar comprobará que ésta realmente haya ido al nivel lógico 1. Si la línea SCL permanece en bajo, el dispositivo maestro interpreta que el esclavo la mantiene así y espera a que SCL vaya a alto antes de continuar. Por suerte, la mayoría de los puertos I2C de los microprocesadores manejan esto de manera automática.

Sin embargo, a veces el manejo de I2C en el dispositivo maestro no está implementado por circuito, sino que es un juego de subrutinas que maneja dos líneas de un puerto. Algunas implementaciones ignoran este estiramiento del reloj. Estas soluciones trabajarán bien con dispositivos tales como las EEPROM, pero no podrán intercambiar datos correctamente con microprocesadores esclavos que utilizan el estiramiento del pulso de reloj. Como resultado, se obtendrán datos erróneos.

ANEXO D. SENSORES APLICADOS

El objetivo del presente anexo es explicar las características esenciales de los sensores aplicados al interfaz. Estas características determinan las variables almacenadas en el datasheet electrónico del sensor y los valores de ganancia y offset que debe aplicar el bloque de control para maximizar la precisión del interfaz.

LDR

Una LDR es un sensor resistivo cuyo valor de resistencia depende de la luz incidente, haciéndose muy grande en la oscuridad (varios megaohmios) y con valores pequeños (del orden de los ohmios) para intensidad luminosa alta. La dependencia de la resistencia con la iluminación es de tipo exponencial como se muestra en la ecuación 1.

$$R = R_o \cdot \left(\frac{L_0}{L}\right)^{\alpha} \quad (1)$$

Donde L_0 [lux] es el nivel de luz de referencia, L[lux] es la iluminación incidente, R_o [Ω] es la resistencia a L_0 y α es una constante que depende del material. Otra de sus características fundamentales es que tiene tiempos de respuesta lentos ante cambios bruscos de intensidad luminosa.

La LDR elegida es NSL-19M51 de Silonex. Su pico de absorción espectral está sobre los 550 nm, dentro del espectro visible. El circuito de acondicionamiento básico es un divisor resistivo (figura 1).

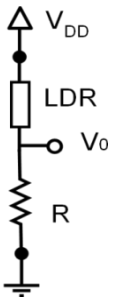

Figura 1: Circuito de acondicionamiento de la LDR

A pesar de que la dependencia entre la resistencia y la iluminación es conocida, el fabricante no proporciona los valores exactos de las constantes L_o, α y R_o, por lo que fue necesario realizar una caracterización del sensor. Así tomé una serie de datos del valor de resistencia entre 10 y 1000 lux, haciendo un ajuste de acuerdo a la ecuación 2.

$$L = A \cdot R^B \tag{2}$$

Las medidas de intensidad de luz fueron realizadas con un luxómetro de Chauvin Arnaux C.A. 813, y las de resistencia con un multímetro digital de 6 ½ dígitos HP 34401ª. Los valores obtenidos fueron de A= 4747800 lux/ Ω^B y B=-1.1771.

Con los valores obtenidos se determinaron los campos que emplearía el datasheet electrónico de la LDR, así como sus valores:

- Tipo de sensor = 2.
- Rango de medidas: intensidad máxima=1000 lux; intensidad mínima=10 lux.
- Constantes de la función de transferencia: A= 4747800 lux/ Ω^B; B=-1.1771.

Los valores de la ganancia y el offset aplicados en la etapa de amplificación de la LDR son: ganancia=1.1795, offset=0.2870.

SENSOR DE HUMEDAD

El sensor de humedad es el H25K5A de Sencera. Es un sensor resistivo cuya resistencia depende de la humedad relativa. Su respuesta es fuertemente dependiente de la temperatura, por lo que es necesario su compensación. Las variaciones de la resistencia son bastante elevadas, yendo desde los kiloohmios hasta los megaohmios. El adecuado acondicionamiento del sensor de humedad necesita incorporar una NTC en el divisor resistivo, como se observa en figura 2.

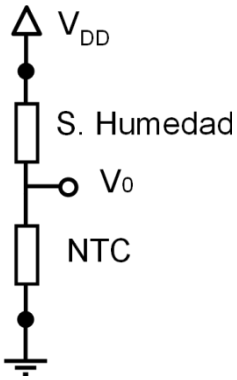

Figura 2: Circuito de acondicionamiento del sensor de humedad

Debido a la dificultad de obtener una expresión matemática que defina correctamente el comportamiento del sensor, el fabricante proporciona una tabla con los valores de resistencia asociados a una determinada temperatura y humedad relativa. Por ello, resulta más conveniente que el PC que recibe toda la información de la red desde el nodo coordinador posea un fichero con estos datos. El interfaz guarda en el *datasheet* electrónico el nombre de dicho fichero. De esta forma, cuando el PC reciba el nombre del archivo y el valor de la medida puede determinar el valor de humedad relativa.

Como el circuito de acondicionado tiene una NTC, también es necesario que se guarden los parámetros de la NTC en el interfaz. El *datasheet* del sensor de humedad estará formado por:

- Tipo de sensor= 4
- Rango de medidas: Humedad relativa mínima= 30 %, máxima= 85 %.
- Nombre del fichero con la tabla de datos de 4 caracteres.
- Datos de la NTC: Constante B=4050 K, resistencia de referencia=50 kΩ, temperatura de referencia=25 *ºC*.

El acondicionado de este sensor precisa programar una ganancia de 1.1036, no siendo necesario el empleo de la etapa de offset.

NTC

La NTC es un sensor de temperatura integrado dentro de la plataforma. En el apartado 2.1.1 se explicaron sus características básicas, así como su circuito de acondicionado. Su datasheet electrónico está formado por:

- Rangos de medida: Temperatura mínima= -20ºC, máxima= 80^{o}C.
- Temperatura de referencia = 25^{o}C.
- Resistencia a temperatura de referencia = 4700 Ω.
- Constante B = 4300 K.
- Resistencia de acondicionamiento = 2700 Ω.

Para la NTC la etapa de amplificación tiene una ganancia de 1.158 y un offset de 0.024 voltios.

SENSOR HALL

El sensor Hall escogido es A1391 de Allegro, un circuito integrado que utiliza el efecto Hall para la medida de pequeñas distancias. El funcionamiento básico de este sensor consiste en colocar un imán a una pequeña distancia del sensor, mediante el efecto Hall el imán induce un voltaje en el sensor proporcional a la distancia. Puede ser alimentado entre 2.5 y 3.5 voltios y su consumo es pequeño. Posee dos entradas, una de ellas para referenciar el voltaje de salida y otra con la que se habilita un modo de bajo consumo. Una característica fundamental de este sensor es que tiene una precisión del orden de micrómetros; además su salida es muy lineal con la distancia. Después de una caracterización del sensor, la dependencia entre distancia y voltaje de salida viene dada por la ecuación 3.

$$d = A \cdot V_0 + B \tag{3}$$

Donde d[m] es la distancia entre el imán y el sensor, V_0 es el voltaje de salida del sensor y A= 226.6 m/v y B= -3.748 m son dos constantes que determinan la curva. Como he comentado su rango de medida de distancias es relativamente pequeño entre 1.78 y 1.9

cm. La aplicación de este sensor dentro del proyecto, es para el estudio del estrés hídrico de las plantas. El sensor se coloca en el tronco de la planta, de esta forma las variaciones en el grosor del tronco, que son del orden de micrómetros, son detectadas por el sensor. Estas variaciones en el tronco, están relacionadas con las variaciones hídricas que sufre la planta, así se puede detectar cuando la planta sufre estrés hídrico.

En su *datasheet* electrónico se guarda la siguiente información:

- Tipo de sensor = 0.
- Rango de medida: distancia mínima = 0.0178 *m*, máxima 0.019 *m.*
- Constantes de la curva característica: A= 226.6 *m/v* y B= -3.748 *m.*

Para el sensor hall, el bloque de amplificación se configura con una ganancia de 6.8114 y un valor de offset de 0.1880 voltios.

TERMOPAR

El termopar es un sensor de temperatura con salida de voltaje diferencial. Simplemente es la unión de dos metales distintos que producen una diferencia de potencial (potencial de contacto) entre ellos, que depende de la temperatura en el extremo donde se produce la unión de los dos metales (unión de medida) y el otro extremo (unión de referencia).

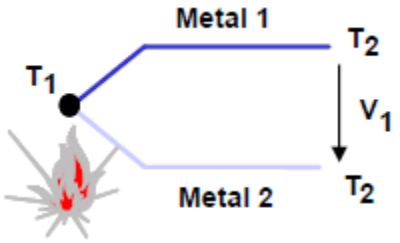

Figura 3: Termopar

En la figura 3 se representa el esquema de medida de un termopar, donde T_1 es la temperatura de la unión de medida, T_2 es la temperatura de la unión de referencia y V_1 es la diferencia de tensión creada. Los valores de la diferencia de tensión en función de la temperatura están normalizados por el NIST (*National Institute of Standars and*

Technology) y son proporcionados en forma de tablas o coeficientes del polinomio de ajuste. Estas tablas o coeficientes vienen dados para una temperatura en la unión referencia de 0ºC, por lo que es necesario realizar la compensación de la temperatura ambiente. Ésta se realiza con la ecuación 4.

$$V_{T_1 0} = V_{T_1 T_2} + V_{T_2 0} \quad (4)$$

La tensión $V_{T_1 T_2}$ es la diferencia de tensión que se obtiene del termopar. Si conocemos la temperatura de referencia T_2 (temperatura ambiente), a partir de las tablas o polinomio de aproximación podemos conocer $V_{T_2 0}$. Ahora ya podemos calcular $V_{T_1 0}$ y utilizando los polinomios de aproximación podemos obtener la temperatura de medida, T_1. Este tipo de sensores se emplean habitualmente para la medida de temperaturas muy altas, algo que no se puede conseguir con otros sensores de temperatura como por ejemplo la NTC.

El termopar no necesita de un circuito de acondicionamiento específico, solamente una amplificación diferencial, ya que los valores de la diferencia de tensión entre metales son muy pequeños, del orden de milivoltios.

El termopar elegido es un termopar tipo N, formado por los metales Nicrosil(Ni-Cr-Si) y Nisil(Ni-Si). Su datasheet electrónico contiene los siguientes campos de información:

- Tipo de sensor = 1
- Tipo de termopar = 1
- Rango de medida: Temperatura mínima = 10^{o}C, máxima = 300^{o}C.
- Coeficientes del polinomio de aproximación: C_0= -0.036934 *mv/ºC*, C_1 = 38.742 *mv/ºC*, C_2=-1.1084 *mv/ºC*, C_3= 0.054095 *mv/ºC*, C_4= -0.0012116 *mv/ºC*.

El NIST proporciona 10 coeficientes de aproximación, pero para nuestra aplicación los errores cometidos son suficientemente pequeños con 5, reduciendo así la memoria necesaria en nuestro *datasheet* electrónico.

En el caso del termopar, su señal de salida no pasa por la etapa de restado de offset. En la etapa de amplificación se configura una ganancia de 291.

FOTODIODO

Un fotodiodo es un semiconductor que polarizado a la inversa produce una corriente de circulación cuando es excitado por la luz. Los fotodiodos pueden trabajar en el espectro visible o en el infrarrojo y producen corrientes bastante débiles. La figura 4 muestra su esquema básico de acondicionamiento:

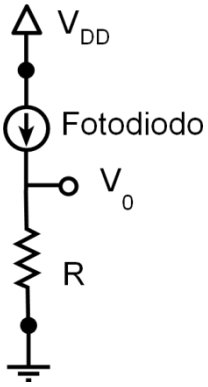

Figura 4: Circuito de acondicionamiento del fotodiodo

El fotodiodo utilizado es el S8265 Hamatsu Photonics. Está especialmente diseñado para realizar medidas en exteriores, su pico de absorción está en torno a 540 nm, en el espectro visible. A pesar de que el fabricante nos asegura que el fotodiodo es bastante lineal, no nos proporciona los valores de la curva que representa la dependencia entre intensidad luminosa y corriente generada. Por ello realicé una caracterización del sensor con el esquema de acondicionamiento propuesto y aproximé los datos obtenidos a la siguiente relación:

$$L = A \cdot V_0 + B \tag{5}$$

Donde L es la iluminación, V_0 es la salida del circuito de acondicionado y las constantes que definen las curva son A = 3036.1 *lux/v* y B = 0.0476 *lux*. La información que contiene el datasheet electrónico del fotodiodo es la siguiente:

- Tipo de sensor = 2;
- Rango de medidas: intensidad máxima = 2000 lux, mínima = 100 lux.
- Constantes de la curva de caracterización: A = 3036.1 *lux/v* y B = 0.0476 *lux.*

La salida del circuito de acondicionamiento del fotodiodo no pasa por la etapa de offset, y la etapa de amplificación se configura con una ganancia de 4.7708.

ANEXO E: TIPOS DE VARIABLES

El objetivo de este anexo es explicar los tipos de variables que se utilizan para guardar los parámetros del *datasheet* electrónico y el ahorro en memoria que suponen respecto a las definiciones de los tipos en el lenguaje C. Hay que remarcar que el ahorro de memoria no sólo supone tener una memoria más pequeña en el interfaz, también supone un ahorro en el consumo ya que la información contenida en la memoria debe ser transmitida al nodo WSN. Los tipos de datos definidos son los siguientes:

- Tipo *Char*: Es el mismo tipo que en C, ocupa un byte en memoria. En el interfaz se usa para almacenar:

    - Palabras de configuración de los registros.
    - Palabras de programación de los potenciómetros.
    - Comandos del sistema.
    - Variables especiales como tipo de sensor y tipo de termopar.

- Tipo *Short*: Es el mismo tipo que en C(dos bytes), el uso típico es para variables enteras cuyo valor máximo es pequeño. Su rango de valores es está entre -32767 y 32767. En la memoria del interfaz se usa para almacenar:

    - Temperaturas.

- Tipo *Int*: Son variables que de 3 bytes, representan números enteros cuyo rango valor máximo es relativamente grande. Su rango de valores está entre -8388607 y 8388607. En C este tipo de variable está definido como 4 bytes de memoria, sin embargo para las variables que son necesarias guardar en el interfaz es suficiente con 3 bytes. En el interfaz se utiliza para guardar:

    - Niveles de iluminación.
    - Resistencias.
    - Constantes enteras.

- Tipo *Real*: Son variables de 3 bytes que representan números reales. Siguen el método de representación de números en coma flotante, en el que el número se

representa a través de una mantisa y un exponente. Los 24 bits de los que se compone el número se descomponen en:

- 1 bit de signo para el exponente.
- 5 bit para el valor del exponente.
- 1 bit de signo para la mantisa.
- 17 bits para valor de la mantisa.

De esta forma podemos representar exponentes en un rango entre -31 y 31, y valores de mantisa entre -131071 y 131071. Así el número se construye como $mantisa \cdot 10^{exponenete}$. En C las variables reales ocupan entre 4 y 8 bytes de memoria, por lo que el ahorro es bastante considerable manteniendo una precisión suficiente. En el interfaz el tipo real se utiliza para guardar:

- Parámetros reales.
- Coeficientes de polinomios de aproximación.

ANEXO F. FORMATO DE LA TRAMA EN LA WSN

El estándar 802.15.4 de Redes inalámbricas de área personal (WPAN) define el nivel físico, el nivel de enlace y MAC, pero deja abierta las especificaciones de niveles superiores para compatibilizarlo con un mayor número de aplicaciones. En la WSN desarrollada por el GDE se utilizan transceptores Xbee, que cumple el estándar y que a niveles superiores permite cumplir diferentes especificaciones, permitiendo por ejemplo comunicaciones ZigBee. Los transceptores son programados con un firmware de Digimesh que permite crear redes con topología en malla, en la que todos los dispositivos menos el coordinador tienen la misma jerarquía, con un algoritmo de enrutamiento que se deriva del AODV(Ad-*hoc On-demand Distance Vector*). Como se ha comentado en la memoria, cada nodo tiene un transceptor, un microcontrolador y un conjunto de sensores. El microcontrolador se encarga de obtener los datos provenientes de los sensores y del interfaz, y posteriormente envía estos datos al transceptor XBee que genera una trama del tipo *receive packet* y la transmite hacia el nodo coordinador.

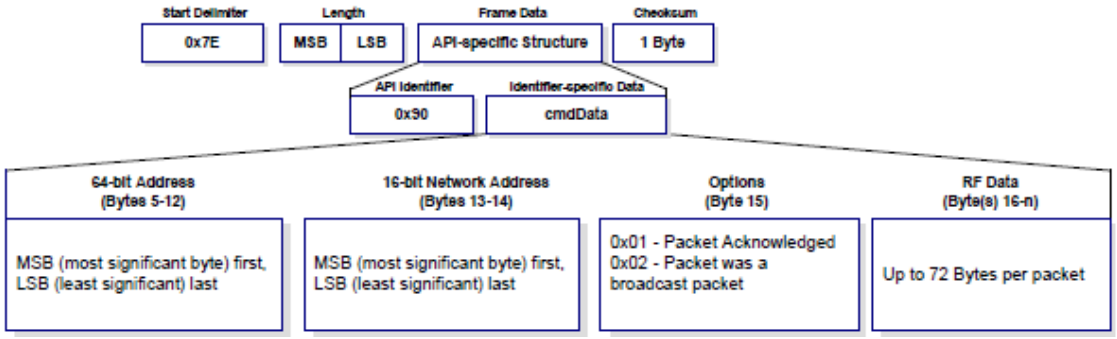

Figura 1: Formato de la trama *receive packet*

En la figura 1 se observa el formato de la trama de nivel de red en la comunicación entre el nodo sensor y el nodo coordinador. En la trama se pueden distinguir los siguientes campos:

- *Start Delimiter*. Este campo marca el inicio de la trama, cualquier dato que sea recibido antes del *Start Delimiter* es descartado.

- *Length*. Está compuesto de dos bytes que representan la longitud total de la trama.
- *Frame Data*. Está formado por el *API identifier* que indica el tipo de trama(en nuestro caso *receive packet*) y *Identifier-specific Data* que contiene los datos de los sensores. El campo *Identifier-specific Data* es generado por el microcontolador del nodo sensor, una vez que ha tomado las medidas de los sensores. En él podemos distinguir:
  - *Address*. Está formado por 8 bytes que indican la dirección del nodo sensor que envía los datos.
  - *Network Address*. Son 4 bytes que determinan la dirección de red destino de los datos.
  - *Options.* Es 1 byte que indica si el paquete es o no de *broadcast*.
  - *RF Data*. Contiene los datos que el nodo sensor quiere transmitir al nodo coordinador, tiene un tamaño máximo de 72 bytes. En nuestra caso está formado por:
    - 6 bytes que contienen las palabras digitales con las medidas de los sensores analógicos del nodo sensor. Estas medidas se digitalizan a través del conversor A/D del microcontrolador.
    - 6 bytes con las palabras digitales correspondientes a las medidas de los sensores digitales del nodo sensor.
    - Datos provenientes del interfaz. En el primer ciclo de trabajo con el interfaz conectado contiene tanto los parámetros como las lecturas de la NTC y del sensor del interfaz, lo que supone 41 bytes. En el resto de ciclos de trabajo sólo se envían los 4 bytes correspondientes a las palabras digitales de las lecturas de la NTC y del sensor.
    - 4 bytes de control. Con el primero y el segundo se indica cuáles son los sensores analógicos y digitales conectados, el tercero indica si los datos del interfaz son 41 o 4 bytes y el último es el número de trama.
- *Checksum*. Son los 8 bits menos significativos de la suma de todos los bytes de la trama, exceptuando el *Start Delimiter*. Su objetivo es verificar de manera sencilla que la trama no está corrupta.