UNIVERSIDAD NACIONAL DE MAR DEL PLATA
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA

Sistema robótico de control y monitoreo remoto, de aplicación en la industria del Oil&Gas.

Autor: Emilio Menna

Directores: Dr. Roberto M. Hidalgo
Ing. Walter A. Gemin

Carrera: Ingeniería Electrónica

Año: 2014

Índice

Resumen

Este proyecto consiste en el diseño y construcción del sistema de control de un mecanismo robótico para medición de longitud y diámetro interno de tubos utilizados en la industria petrolera. El dispositivo robótico se introduce en un extremo del tubo, lo recorre hasta el extremo opuesto, y vuelve entregando el valor de las mediciones. Para impulsarse cuenta con un motor de corriente continua. Posee tres juegos de sensores, uno para la medición del diámetro, y dos para detectar su posición dentro del tubo. Además cuenta con un display para mostrar datos al usuario, un teclado para poder ingresar datos, y conectividad USB para realizar la comunicación con la computadora. Para realizar la automatización del dispositivo mecánico se utilizó un microcontrolador 18F4550 de la firma Microchip, el cual se programó en lenguaje C utilizando el entorno CCS de PICC. El objetivo final de este proyecto es controlar el robot para lograr automatizar la medición de longitud y diámetro interno y tener un registro de las mediciones realizadas en lotes de tuberías, a la vez que brinda una interfaz de manejo sencilla.

I. Introducción

En la industria petrolera se utilizan tubos llamados *casing*, durante operaciones de perforación, producción y reparación de pozos. Su función es mantener la estabilidad del pozo, evitar la contaminación del suelo, aislar el agua y controlar la presión [8]. Los materiales utilizados en su fabricación y dimensiones, varían según las características de la perforación.

Antes de su instalación es necesario medir las longitudes y diámetros internos de los mismos debido a que se debe conocer de antemano si los tubos están en condiciones de ser utilizados.

Los métodos que se utilizan habitualmente para estas mediciones son los siguientes:

- Para la medición de longitud se pueden utilizar dos tipos de herramientas. La primera es la cinta métrica de acero correctamente calibrada [9]. La segunda es el instrumento de medición de distancia por láser. La utilización de cualquiera de los métodos es indiferente. La distancia debe ser medida en milímetros. La medición y registro la realizan al menos dos operarios, los cuales deben entregar una planilla manuscrita con los resultados de las mediciones al usuario final. Luego, esta planilla es transcripta a una computadora por dicho usuario. Este proceso de medición se denomina *tally*.

Figura I.1 - Medición de la longitud de los tubos con cinta de acero.

- La medición del diámetro interno del tubo consiste en corroborar que este permanezca mayor a un valor determinado. Para esto se hace circular por el interior del tubo una bala cilíndrica fabricada de metal, cuyo diámetro coincide con el valor mínimo de diámetro aceptado. Esta medición la realizan dos operarios, los cuales se posicionan en los extremos del tubo, y hacen circular la bala tirando de una soga atada a la misma. Este proceso de medición se denomina calibración. En caso de que el tubo no cumpla con las condiciones de calibración, es desechado.

Figura I.2 - *Drift* utilizado para la calibración de los tubos.

I.1 Problemáticas Observadas

En ocasiones, se malinterpretan los datos de las planillas manuscritas realizadas por los operarios durante la medición de la longitud de los tubos al ser transcriptas a una planilla digital en la computadora por el usuario final. Esto ha generado un error en el cálculo de la cantidad de caños a colocar en el orificio de extracción de petróleo, provocando que tengan que retirarse caños, y teniendo como consecuencia una pérdida muy grande de tiempo y dinero para la empresa.

En cuanto al calibrado, en algunos casos, se observó que para evitar el agotamiento que produce el método actual de medición, se coloca la bala en un extremo, y se le aplica aire comprimido para impulsarla a lo largo del caño. Esto ahorra tiempo y alivia la fatiga, pero se han registrado casos en los cuales la bala impactó sobre el operario al otro lado del caño, por lo que se considera un método de alto riesgo.

I.2 Origen del Proyecto

En el año 2011, la empresa DIGIMAGE ELECTÓNICA, situada en la localidad de Mar del Plata, Partido de Gral. Pueyrredón, tomó conocimiento de las diferentes problemáticas observadas.

La empresa comenzó el desarrollo de un dispositivo robótico capaz de solucionar estos problemas, llagando a construir un prototipo mecánicamente funcional. Sin embargo, se vieron limitados para realizar el desarrollo del sistema de comando y control.

Por tal motivo, se solicitó a la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata, el diseño y construcción de un sistema de control para el dispositivo construido, con el objetivo de poder comenzar con las pruebas en campo del mismo.

Una vez comunicada la inquietud por parte de la empresa se abocó a la tarea de analizar detalladamente los sistemas de medición actuales. Asimismo, fue necesario un estudio minucioso del dispositivo robótico existente con el fin de realizar un control eficiente sobre el mismo. Luego de completar estas tareas, se discutió sobre el uso que tendría el producto final, y los objetivos solicitados por la empresa. De esta manera se pudo obtener una visión general del trabajo a realizar y sus posibles formas de llevarlo a cabo.

I.2.1 Descripción del dispositivo robótico

El prototipo desarrollado por empresa, encargado de realizar las tareas de calibración y *tally*, se muestra en la Figura I.3. Este se desplaza por el interior del caño, utilizando un motor ubicado en el eje trasero (1).

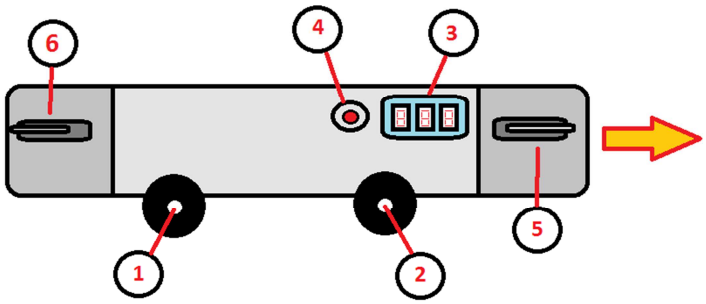

Figura I.3 - Modelo esquemático del dispositivo robótico existente.

En el eje delantero (2) tiene acoplado un *encoder* mecánico que genera pulsos eléctricos a medida que el dispositivo avanza. Estos pulsos son contados y desplegados en tres *display* 7 segmentos (3). Para reiniciar la cuenta, el dispositivo posee con un botón de *Reset* (4), que al presionarlo vuelve el valor mostrado en los *displays* a "000".

Los finales de carrera dispuestos en el extremo delantero (5) y trasero (6) del dispositivo le permiten detectar cuando se encuentra dentro o fuera del caño.

I.2.2 Funcionamiento del dispositivo robótico

Para comprender el funcionamiento del dispositivo, primero se realizó un diagrama en bloques del mismo, el cual se muestra en la Figura I.4.

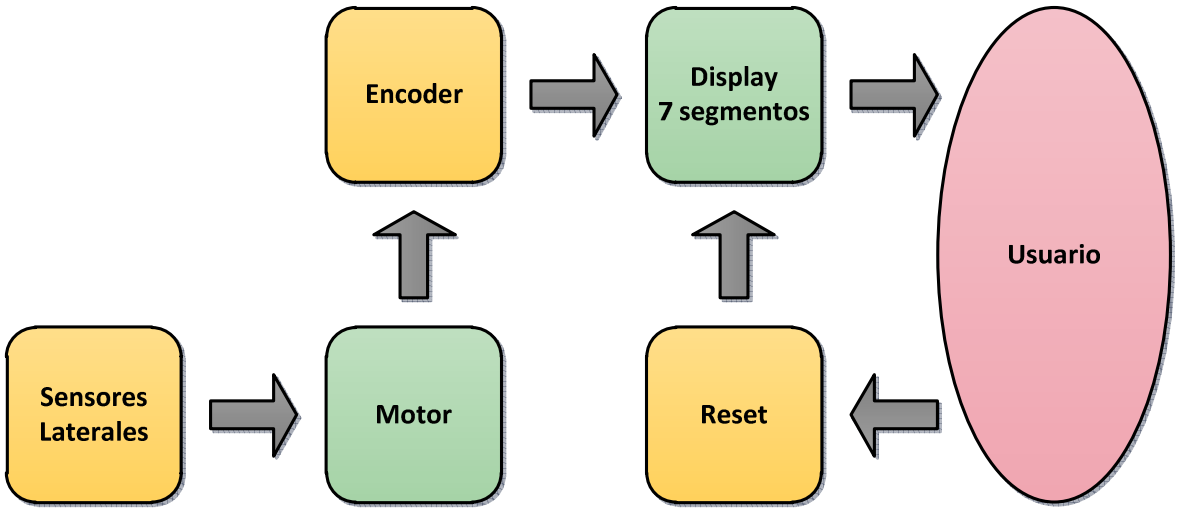

Figura I.4 - Diagrama en bloques del dispositivo robótico construido por la empresa DIGIMAGE ELECTRONICA.

El modo de operación es el siguiente: En primer lugar debe ser introducido en un extremo del caño. Utilizando los sensores laterales, el dispositivo puede reconocer cuando se encuentra completamente dentro, en cuyo caso activa automáticamente el motor para comenzar a desplazarse. En caso de que alguno de los sensores no se encuentre presionado, el robot interpreta que se encuentra fuera del caño, por lo que su activación no es posible.

Una vez dentro del tubo, el dispositivo lo recorre por su interior hasta el extremo opuesto, mientras cuenta los pulsos entregados por el *encoder*. Cuando llega al final del

recorrido, frena automáticamente indicando en los *displays* 7 segmentos el resultado de la medición.

Para realizar una nueva medición, se presiona el botón *Reset*, que lleva los *displays* a 0, quedando preparado para empezar a contar nuevamente.

El dispositivo cuenta con una interfaz de manejo sencilla pero poco intuitiva, lo que requiere que el operario deba estar muy familiarizado con la herramienta. No realiza una vuelta automática, evitando el ahorro de personal. Además, el dato desplegado indica la cantidad de pulsos contados por el *encoder*, por lo que luego se debe realizar un cálculo para obtener la longitud verdadera del tubo. Por último, no permite detectar anomalías en el diámetro a lo largo del tubo, impidiendo reemplazar el engorroso método actual de medición.

Por lo tanto, si bien cuenta con una interfaz sencilla para el usuario y evita la fatiga del operario, no soluciona la totalidad de los problemas observados.

I.3 Especificaciones preliminares

El objetivo del proyecto es realizar un sistema de control que le permita al dispositivo antes descripto cumplir con las siguientes especificaciones principales:

- Realizar la ida y vuelta por el tubo.
- Medir la longitud del tubo.
- Detectar anomalías en el diámetro e indicar a que distancia se encuentran.
- Presentar una interfaz sencilla y amigable para el operario.

Además, existen objetivos secundarios que, si bien no son indispensables, son deseables para obtener un sistema con mayor versatilidad:

- Almacenar la información de cada uno de los tubos de un lote.
- Identificar a que pozo pertenece cada lote.
- Identificar el operario que realizo la medición.
- Permitir el almacenamiento de la información de varios lotes (pertenecientes a distintos pozos).

- Descargar los datos en forma de planillas en una computadora.
- Minimizar los costos de implementación.

De esta manera, se logra la integración de dos tareas, calibrado y *tally*, utilizando un solo operario, y tratando de minimizar los errores de interpretación e incorrecta manipulación.

II. Anteproyecto

Una vez estudiada la problemática y fijados los objetivos principales, se abocó a la tarea de decidir de qué manera se implementaría la solución deseada. Esto incluyó tomar decisiones sobre el tipo de tecnología a utilizar, el software con el que se desarrolló el proyecto, y los componentes que se utilizaron.

II.1 Opciones de Control

II.1.1 Controlador Lógico Programable (PLC)

Una de las opciones por excelencia utilizadas en la industria para realizar automatizaciones, son los módulos PLC (*Programmable Logic Controller*). Estos dispositivos poseen una gran inmunidad al ruido eléctrico, por los que permiten crear un sistema de control robusto en entornos industriales. Admiten controlar múltiples entradas y salidas, y existe una gran diversidad de módulos pre-fabricados que permiten incorporar interfaces y sensores rápidamente. Su programación se realiza en software especializado, provisto por la marca de cada dispositivo particular. La mayor desventaja es su elevado costo.

II.1.2 Microcontroladores

Otra opción son los microcontroladores. Muchas empresas dedicadas a la construcción de semiconductores producen los suyos propios, por lo que el mercado es inmenso. El valor de un microcontrolador es mucho menor que el de un PLC. En el caso de interfaces y periféricos se requiere un esfuerzo adicional para desarrollar los circuitos correspondientes, pero las opciones son ilimitadas, lo que genera mayor flexibilidad. La robustez eléctrica depende del diseño final. En este proyecto se opta por trabajar con microcontroladores llamados PIC (*Peripheral Interface Controller*), de la firma Microchip, debido a que los mismos disponen de abundante información, y su programación se puede realizar con diferentes lenguajes (*software*) de fácil acceso comercialmente.

II.2 Lenguaje de programación

Dentro de los lenguajes en los que se programan rutinas para PIC existen tres posibilidades:

- Lenguaje BASIC
- Lenguaje Ensamblador (*assembler*)
- Lenguaje C

A continuación se describen brevemente cada una de las posibilidades enumeradas anteriormente, cada una con sus ventajas y desventajas.

II.2.1 Lenguaje BASIC

Aunque es un lenguaje muy simple y con instrucciones fácilmente legibles, incluso por no expertos, nunca se logra tener el control del programa en cuanto a tiempos de ejecución y control de registros bit a bit.

Es muy complicado el manejo de interrupciones simultáneas y tiene limitaciones cuando se genera el archivo que se carga al microcontrolador, ya que no optimiza su tamaño, por lo que ocupa más espacio en la memoria de programa del PIC. La mayoría de compiladores para este lenguaje pueden utilizarse únicamente bajo sistema operativo Windows.

II.2.2 Lenguaje Ensamblador

Es el lenguaje de bajo nivel natural de la línea PIC, tanto para gama baja, media o alta. Con él se tiene un aprovechamiento eficiente de sus recursos. Se pueden crear macros con este lenguaje, para después simplificar el código en diferentes desarrollos, como así también controlar los tiempos y los registros bit a bit. Resulta excelente para manejar interrupciones simultáneas. Cuando se genera el archivo que se carga dentro del microcontrolador, éste resulta completamente optimizado. A pesar de sus ventajas, los

programas suelen ser extensos, resulta difícil detectar errores, y la programación se vuelve engorrosa al usar periféricos complejos.

II.2.3 Lenguaje C

Es un lenguaje de nivel medio. En él se pueden construir rutinas matemáticas fácilmente y admite ser combinado con lenguaje Ensamblador, sobre todo en la gama alta. Se pueden crear macros para después simplificar el código en diferentes desarrollos. Es aceptado por la empresa fabricante Microchip, e incluso esta brinda algunos compiladores C. La desventaja es que los programas al compilarlos pueden resultar un poco extensos y pesados, por lo que debe tenerse en cuenta la capacidad de memoria de programa del PIC a utilizar.

Para este lenguaje existen también varias empresas que producen software y compiladores, entre ellas las más importantes son:

- CCS
- Microchip
- Hi-Tech

En este caso se decidió programarlos en lenguaje C, ya que es un lenguaje versátil y soportado perfectamente por los microcontroladores de Microchip, permitiendo que trabaje eficientemente. El entorno de programación utilizado fue el CCS.

II.3 Selección de componentes

A continuación se explican los criterios utilizados en la selección de los componentes que forman parte del proyecto.

1- Microcontrolador

Microchip ofrece una gran cantidad de microcontroladores con diversas funcionalidades. Debido a la cantidad y características de los periféricos a controlar se

decidió trabajar con la mayor cantidad de entradas/salidas posibles. Los microcontroladores de 40 pines son los más grandes que se encuentra en formato DIP (*Dual in-line package*), el cual permite trabajar en una placa experimental y realizar su soldadura con herramientas convencionales. Además, el hecho de necesitar de una interfaz USB, llevo a la decisión de utilizar el PIC18F4550, el cual está preparado para esta función. Igualmente, luego se propuso implementar la comunicación a través de la EUSART (*Enhaced Universal Synchronous Asynchronous Receiver-Transmitter*) que el mismo dispone, utilizando una interfaz externa a USB apropiada, no usando esta función.

2- Teclado

Para poder ingresar datos e interactuar con los menús, se incorporó un teclado matricial de 3x4. El mismo se muestra en la Figura II.1. En el teclado se muestran los números del 0 al 9, y además dos teclas para aceptar y cancelar las opciones desplegadas. Un teclado más chico no hubiera permitido mostrar todos los números a la vez, y uno más grande hubiera sido difícil de incorporar al robot debido al reducido espacio disponible. Esta manera de ingresar datos se consideró la más intuitiva para el usuario, con una complejidad relativamente baja de implementación y costo.

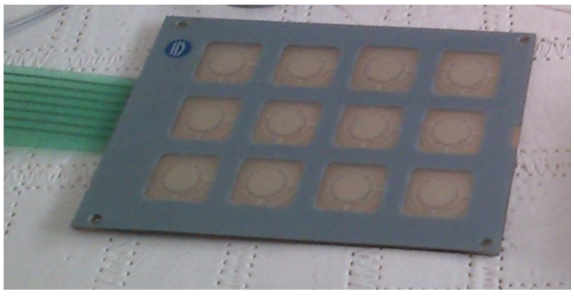

Figura II.1 Teclado matricial de 3x4.

3- Display

En el mercado existen una gran variedad de *displays*, desde simples *display* 7 segmentos, hasta alfanuméricos de 2 y 4 líneas. Se decidió finalmente incorporar al diseño final un *display* GLCD (*Graphic Liquid Crystal Display)* de 128x64 pixeles, ya permite desplegar tanto texto como imágenes cómodamente. El mismo de muestra en la Figura

II.2. Este cuenta con el controlador KS0108, el cual se seleccionó debido a la disponibilidad de librerías para manejarlo, lo cual ahorra tiempo al momento de la programación en el entorno CCS. Este dispositivo junto con el teclado, utilizan la mayor cantidad de pines del microcontrolador.

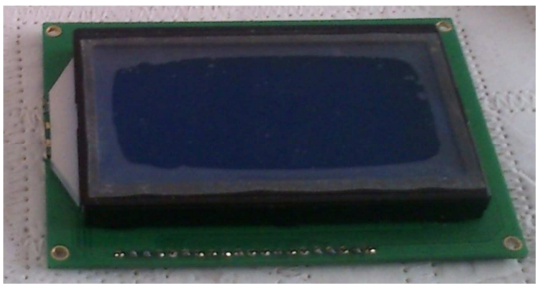

Figura II.2 - Display GLCD de 128x64 pixeles.

4- Dispositivos I^2C

Debido a la necesidad de almacenar información, se decidió incorporar una memoria EEPROM. Como no se disponía de gran cantidad de pines del microcontrolador libres, se optó por utilizar una memoria con protocolo de comunicación I^2C. Este protocolo permite conectar varios dispositivos a un mismo bus de dos líneas, en las cuales una controla la señal de sincronismo, y la otra se utiliza para entablar una comunicación tipo serie.

Utilizando la flexibilidad que provee, se incorporó también un RTC (*Real Time Clock*) con I^2C, que permite en cualquier momento contar con información sobre la fecha y hora con un consumo mínimo de batería. Estos datos se consideraron importantes para desplegar información sobre la medición realizada en el dispositivo final.

5- Driver H

Para manejar el motor se utilizó un circuito integrado llamado *Driver H*, el cual permite controlar cuatro motores de corriente continua en un solo sentido de giro, o dos en ambos sentidos de giro. En este caso no se utilizó la totalidad de la capacidad, ya que el dispositivo solo cuenta con un motor, el cual puede girar en los dos sentidos.

Se midió la corriente de consumo del motor incorporado en el dispositivo para elegir un *Driver H* que soporte dicha carga. Debido a que el motor podría cambiarse en un futuro este bloque se diseñó por separado, como se verá en secciones siguientes.

6- Encoder

Por cuestiones de disponibilidad en el mercado, se utilizó un *encoder* mecánico de 24 pulsos por vuelta. Esto dispositivo actúa como transductor entre las giros de la rueda del dispositivo y la distancia recorrida por el mismo.

7- Comunicación

Como se explica en el inciso (1-), se tomó la decisión final de realizar la comunicación utilizando la USART del PIC18F4550. Aunque en principio se pensó en utilizar la funcionalidad USB que provee el PIC18F4550, esto utilizaba una cantidad considerable de memoria de programa, y requería realizar un driver específico para que el dispositivo sea compatible con los distintos sistemas operativos disponibles en las computadoras personales (PC). A pesar de descartar esa opción, se utilizó una interfaz externa de USART a USB. La interfaz utilizada es un UART2USB del fabricante CIKA, y permite trabajar con alimentación de 3V y 5V. La interfaz está basada en el circuito integrado FT232BL, para el cual se actualizan los drivers periódicamente, lo que evitará que la comunicación de la placa quede obsoleta.

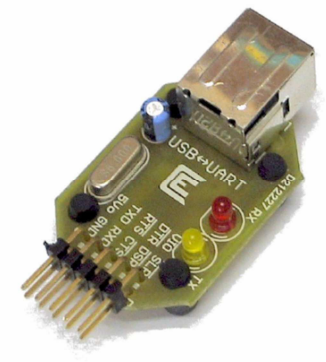

Figura II.3 - Interfaz conversora USB a UART TTL (5V) o LV-TTL (3V).

8- Conexionado de Alimentación

Para la alimentación de la placa se toma tensión de una pack de baterías de 12V, ya incluidas en el diseño del dispositivo robótico existente. Estos 12V alimentan al motor a través del *Driver H*, y a la placa a través de un regulador de 5V.

III. Proyecto

III.1 Controlador propuesto

A continuación en la Figura III.1 se proporciona el diagrama en bloques del controlador propuesto. En este se incluyen todos los periféricos descriptos anteriormente, como así también el personal humano involucrado en el proceso de medición.

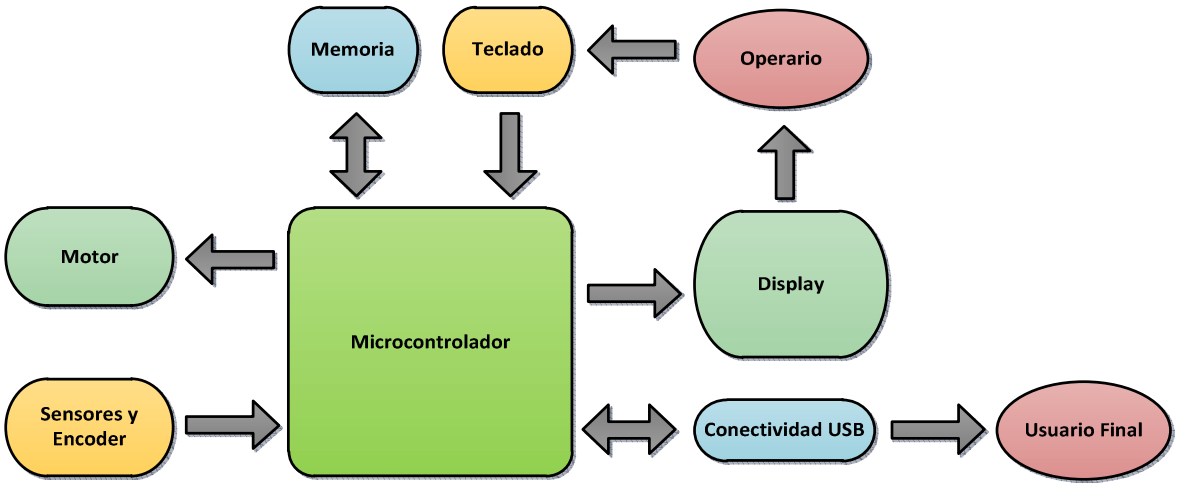

Figura III.1 - Diagrama en bloques del controlador propuesto.

Cabe aclarar que el operario es el encargado de operar el equipo al momento de realizar la medición. En cambio, el usuario final solo se encarga de recibir la información sobre las mediciones realizadas y acondicionarlas para su posterior procesamiento y utilización. A partir de este diagrama en bloques se comenzó con una etapa de simulación, seguida del diseño y construcción de la placa controladora motivo de este proyecto.

III.1.1 Simulación

En orden de corroborar el funcionamiento de los componentes seleccionados para la placa controladora, sin efectuar una inversión económica significativa, se utilizó el software Proteus para realizar simulaciones previas. Este cuenta con un entorno de simulación a nivel circuital, llamado ISIS, y un entorno para el diseño de PCB (*Printed Circuit Board*) llamado ARES.

En la Figura III.2, se muestra el banco de pruebas virtual construido en el ISIS, al cual se llegó luego de varias simulaciones tratando de optimizar conexionado.

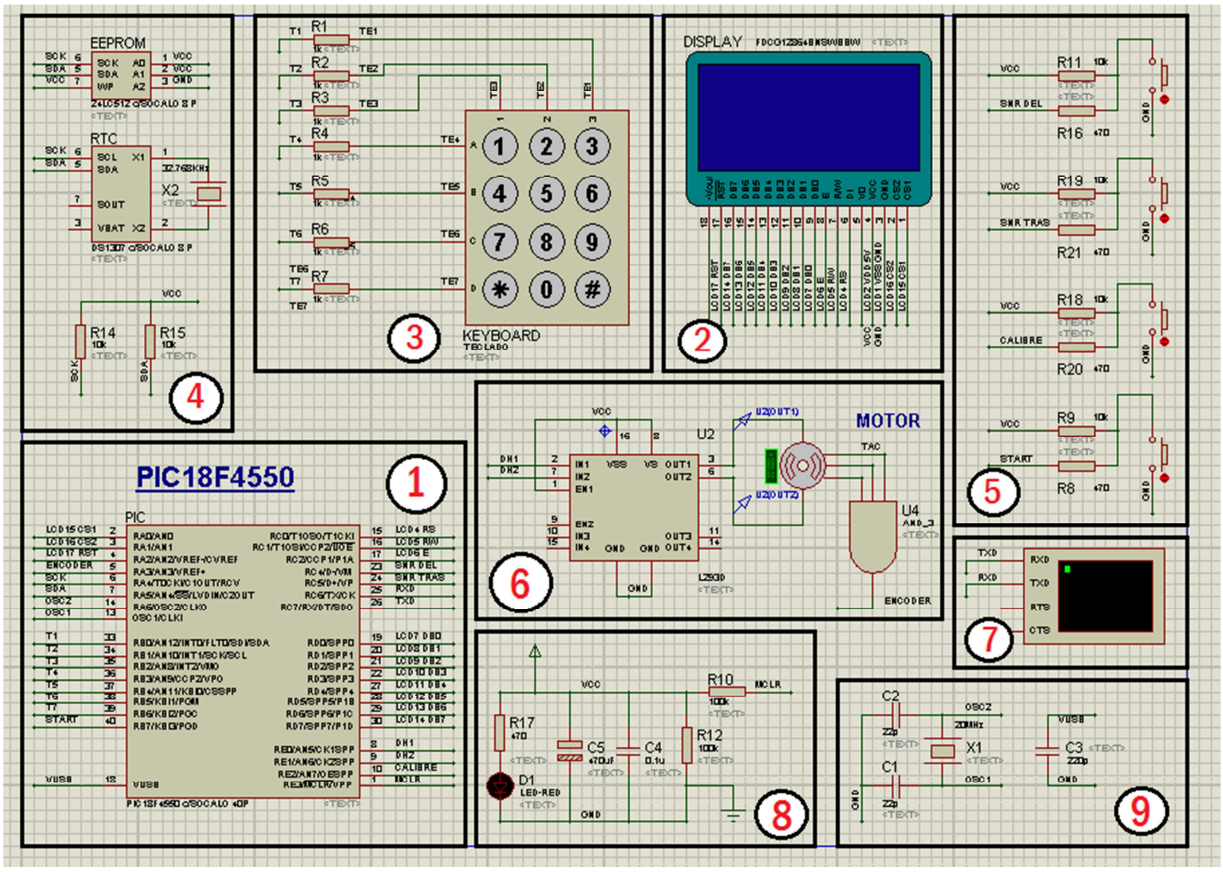

Figura III.2 - Prototipo virtual de la placa controladora en ISIS.

En este banco virtual, se verificó el comportamiento de los distintos dispositivos que fueron probados a lo largo del desarrollo del proyecto. Las pruebas se realizaron generando pequeños programas en C, y ejecutándolos desde el entorno ISIS dentro de un microcontrolador virtual. Si bien estas simulaciones en ocasiones difieren levemente del comportamiento eléctrico real de los dispositivos, en líneas generales se puede anticipar si el desempeño de los mismos cumplirá con las expectativas requeridas. Además, dentro de las simulaciones, es posible detener la ejecución del programa del microcontrolador y verificar, por ejemplo, que las variables se comporten como era esperado, y los espacios de almacenamientos de las memorias internas y externas del PIC, estén guardando la información correctamente.

III.2 Diseño de la placa controladora

Una vez seleccionados los componentes a utilizar, se comenzó con el diseño y construcción de la placa controladora. En primera instancia, luego de conseguir todos los componentes seleccionados, se realizaron pruebas sobre una placa de pruebas genérica y una *protoboard*. Esto permitió depurar errores eléctricos en el conexionado que no pudieron notarse durante la etapa de simulación. Luego, se construyeron módulos que agrupaban los periféricos de características similares y se probaron nuevamente en la placa de pruebas para observar su comportamiento. Por último, se diseñó y construyó la placa final que se coloca dentro del dispositivo mecánico. A lo largo del diseño se tuvo en cuenta la disposición de los componentes para su conexionado, y las dimensiones de la placa, ya que el espacio dentro del robot es reducido.

III.2.1 Pruebas Preliminares

Antes de comenzar directamente con el desarrollo de la placa final que se encuentra dentro del dispositivo, se consideró apropiado hacer pruebas con los periféricos en una placa de entrenamiento. Por este motivo se diseñó y construyó la placa de pruebas que se puede observar en la Figura III.3.

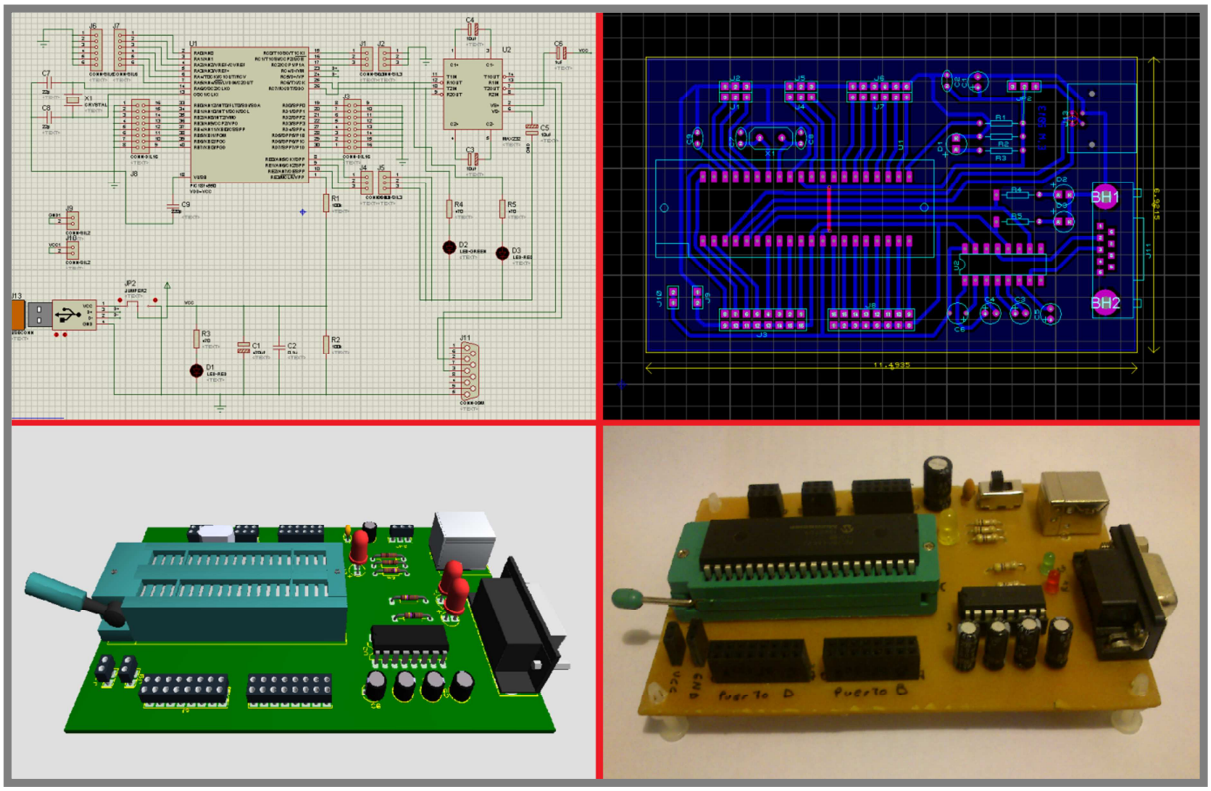

Figura III.3 - Placa de Pruebas.

La placa cuenta con todas las salidas del PIC disponibles, excepto por los pines dedicados a la comunicación de la EUSART y USB. La EUSART está conectada a un MAX232, circuito integrado que genera una interfaz EUSART-RS232, lo que permite entablar una comunicación serie con la computadora. Además, la conexión USB provee la alimentación de la placa.

Al igual que en la etapa de simulación, en esta instancia se programaron pequeñas rutinas en C para probar los periféricos. La programación se realizó cargando previamente en el microcontrolador un *firmware* llamado *BootLoader*. Este *firmware* funciona de la siguiente manera: cuando se alimenta el microcontrolador, este ejecuta el *BootLoader*, el cual espera unos segundos a que llegue un nuevo programa a través de la EUSART, para cargar en el microcontrolador. Si esto no ocurre, ejecuta el último programa que se haya cargado en su memoria. Esto permite la programación sin retirar el microcontrolador de la placa en la que está siendo utilizado (programación *In-Circuit*) utilizando solo 2 pines, aunque no permite la depuración de errores en esta condición (*debug In-Circuit*).

EL teclado matricial se conectó de la manera habitual, pudiendo lograr resultados un poco tiempo.

En la caso de la pantalla se tuvo el siguiente inconveniente. A la frecuencia de trabajo del PIC, que en este caso es de 48MHz, la librería provista por el entorno de programación no funcionó correctamente. El problema radicaba en que no eran respetados los tiempos que se muestran en la Figura III.4, lo que imposibilitaba la correcta escritura de los datos.

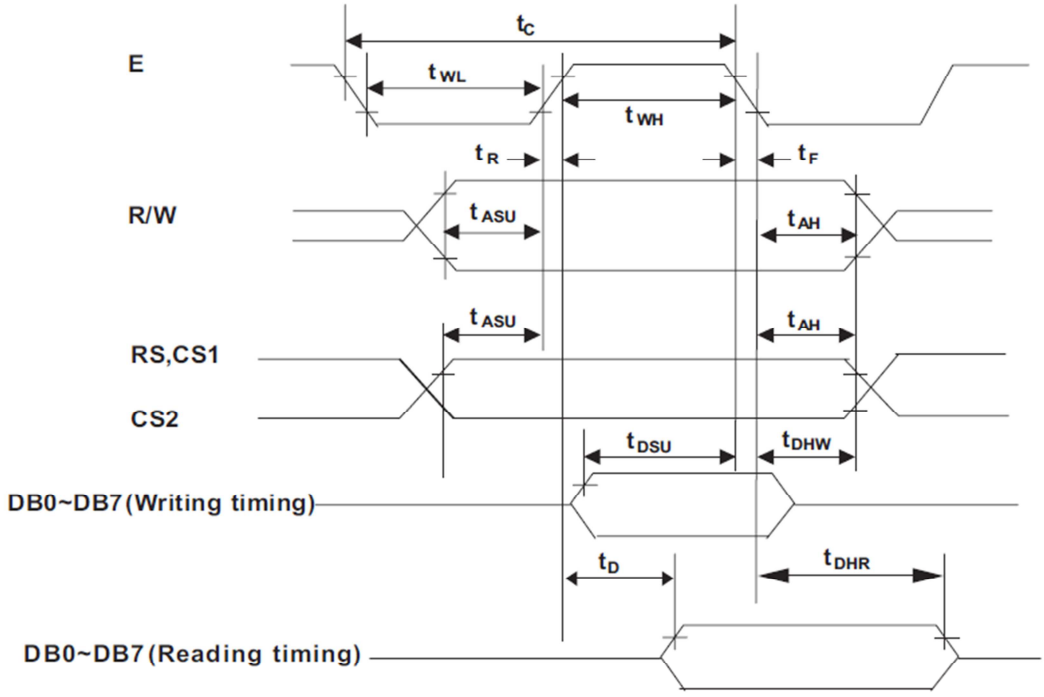

Figura III.4 - Diagrama de tiempos del GLCD "FDCG12864B".

En la librería original, todos los tiempos están en función de la frecuencia de trabajo del PIC. En las simulaciones a baja frecuencia, el *display* funcionaba correctamente, pero no lo hacía para frecuencias de trabajo por encima de los 20MHz. Una vez corregidos los tiempos, se logró el correcto funcionamiento del *display*.

El RTC y la memoria EEPROM se pudieron incorporar a la placa de pruebas sin ningún problema relevante. Únicamente se tuvo en cuenta el direccionamiento de ambos dispositivos para su correcta comunicación a través del bus I^2C.

En el caso del *encoder*, si bien el funcionamiento teórico es sencillo, no se contaba con información sobre la conexión de sus pines. Dada esta situación, se utilizaron las instalaciones de la Facultad de Ingeniería con el fin de realizar las mediciones pertinentes para corroborar sus características. El instrumental solicitado consistió de un osciloscopio de dos canales y una fuente de tensión variable. Se ensayó la alimentación de los pines

alternativamente, midiendo de la misma manera sus salidas, hasta obtener en una de ellas el tren de pulsos característico de este dispositivo. Se pudo identificar que para su correcto funcionamiento demanda una resistencia de *pull-up* a su salida. De esta manera se conocieron las características del *encoder* utilizado.

La interfaz USBtoUART mostrada en la Figura II.3 también funcionó perfectamente. Incluso se pudo realizar la programación del PIC utilizando una PC con conectividad USB.

El *driver H* se pudo probar luego de realizar una placa específica para el mismo, ya que este requiere de una superficie de disipación de calor. En el diseño de esta placa se dejó lugar para el manejo de un segundo motor, en caso de que la empresa desee incluirlo en un futuro. Esta etapa fue considerada como de potencia, ya que la alimentación se realizaba con 12 Volts, a diferencia de los 5 Volts que alimentaban el resto de los circuitos.

III.2.2 Disposición Física

Una vez que se tuvo funcionando todos los dispositivos correctamente, se comenzó con el diseño y construcción de la placa controladora. Esta etapa fue de suma importancia para la correcta incorporación de la placa al dispositivo. Se planteó con el fin de que la misma cumpla simultáneamente con las siguientes características:

- Inclusión de todos los componentes utilizados.
- Dimisiones de la placa.
- Facilidad en el conexionado de componentes externos.

Esto resultó dificultoso ya componentes como el display y el teclado cuentan con *buses* de datos de hasta ocho pines, además de contar con conectores especiales. Esto sumado al espacio reducido que se disponía para distribuir los componentes en el PCB, implico un gran desafío de diseño. Conjuntamente, se dispusieron los conectores para los sensores externos de manera que el montaje de la placa final sea sencillo. Esto provee además comodidad al momento de desarmar el equipo y reemplazar componentes en casa de que se encuentren defectuosos o se requiera su mantenimiento.

En el transcurso de tiempo que llevó llegar a esta etapa, la empresa construyó un nuevo prototipo más robusto, para el cual está destinada la placa controladora. En la Figura III.5 se puede observar un modelo en 3D que se realizó del nuevo prototipo.

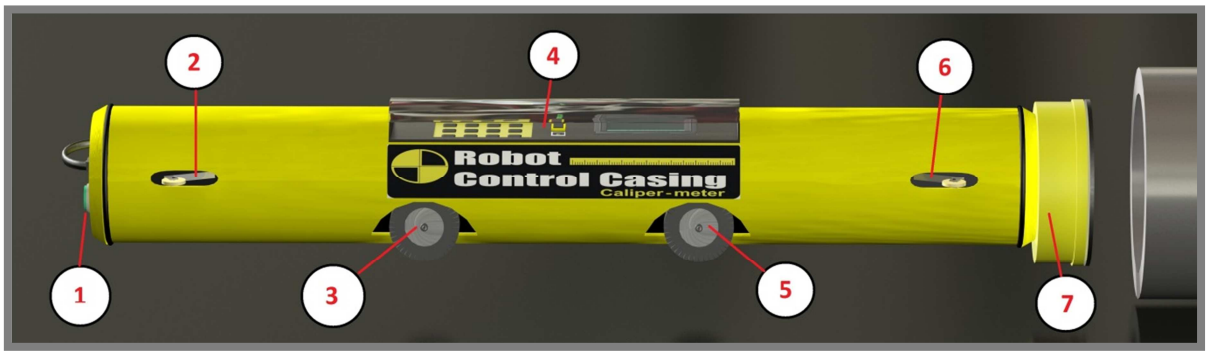

Figura III.5 - Modelo 3D del nuevo prototipo robótico.

En dicha figura se pueden observar los conjuntos de sensores en los extremos del robot, (2 y 6). En el extremo delantero (7) se encuentra el sensor de calibración. El motor se encuentra conectado al eje de las ruedas trasera (3) bajo del panel de control del dispositivo (4). El panel incluye la pantalla, el teclado, la conectividad USB y el interruptor de encendido en su parte frontal. En su reverso se colocó la placa controladora. En el eje de las ruedas delanteras (5) se colocó el *encoder* mecánico, del cual se extrae la información sobre el avance del robot. Sobre el extremo trasero, en el interior, está dispuesto el pack de baterías del cual se obtiene la alimentación. El controlador del motor, *Driver H*, se encuentra sobre el extremo delantero. Sobre la cara trasera (1) se encuentra el botón de *Start*, el cual permite activar el dispositivo cuando se encuentra dentro del caño.

III.2.3 Diseño Modular

Por comodidad en el conexionado y en la detección de errores, se procuró realizar un diseño modular. En este caso se realizaron dos módulos, uno de control, y uno de potencia. El módulo de control, se ocupa de ejecutar el programa, leer las entradas, controlar las salidas lógicas y almacenar la información. El módulo de potencia se encarga de convertir las salidas lógicas en salidas de potencia para controlar el motor. Además, se diseñó y construyó de un módulo para el control de los sensores laterales.

III.2.3.a Placa principal

En la Figura III.6, se puede observar la placa controladora final. Ésta incluye todos los periféricos que se describieron en la sección II.3, integrados en una sola placa.

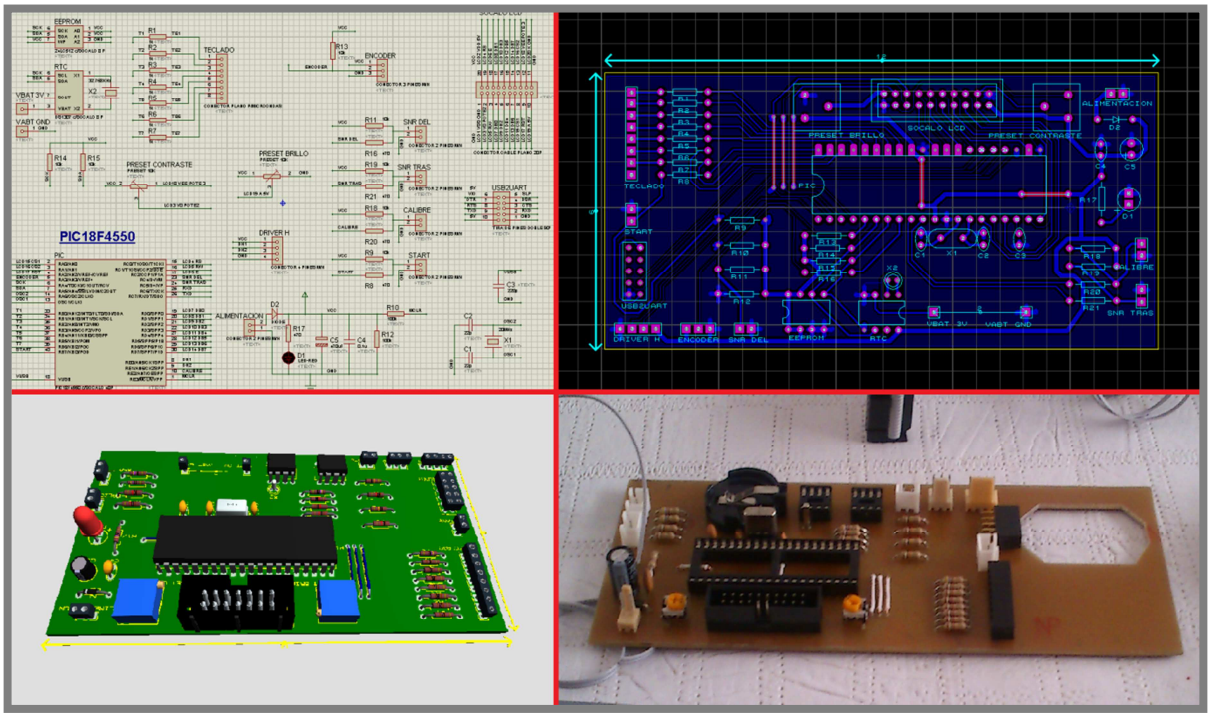

Figura III.6 - Diseño y construcción de la Placa Controladora.

El diseño del diagrama se realizó en ISIS, y luego en el ARES se dio forma al PCB. Los conectores se disponen en los laterales de la placa para que la conexión de los componentes se pueda realizar fácilmente. Esto permite montar el dispositivo de manera rápida y poder desarmarlo también rápidamente en caso de necesitar, por ejemplo, realizar el intercambio de un componente defectuoso, con una dificultad mínima.

Además se ajustó el tamaño para coincidir con el panel externo, de manera de remover toda la parte electrónica de control del dispositivo en un solo paso, como se ve en la Figura III.7.

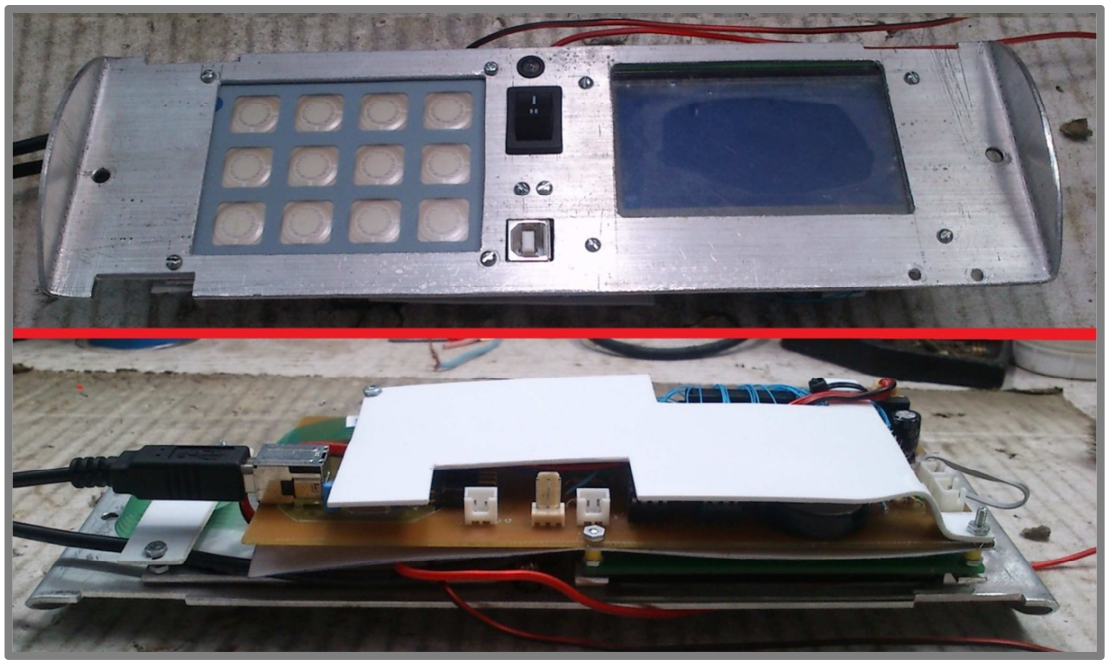

Figura III.7 - Panel de control del robot (superior) y placa controladora (inferior).

III.2.3.b Controlador del Motor

El controlador del motor se realizó en un módulo separado, ya que se consideró como una etapa de potencia. El mismo se puede observar en la Figura III.8.

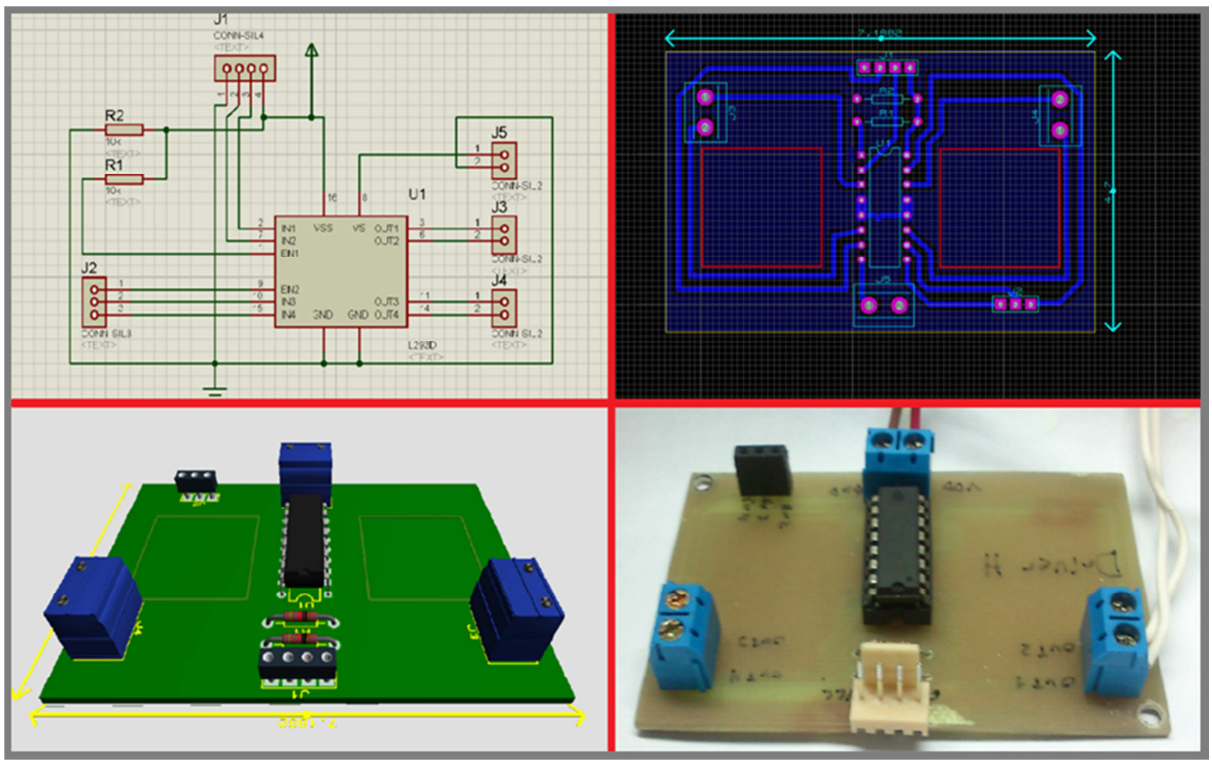

Figura III.8 – Diseño y Construcción del Módulo de Potencia

Este cuenta con 2 alimentaciones, una para la parte lógica para la cual se utilizan 5 Volts, y una para la parte de potencia que es variable, ambas con su correspondiente masa. La placa cuenta también con 4 entradas lógicas y 4 salidas de potencia. Al utilizar un solo motor, se utilizan solo un par entrada-salida. Controlando las dos entradas, se puede alternar el valor de las dos salidas de potencia correspondientes. Al conectar un motor de corriente continua a estas dos salidas, se puede conseguir que el mismo gire en ambos sentidos, como se muestra en la

Tabla 1.

Motor	**Entrada 1**	**Entrada 2**
Quieto	0	0
Avanza	1	0
Retrocede	0	1
Quieto	1	1

Tabla 1 – Comportamiento del motor con respecto a las entradas del *Driver H*.

En el caso del prototipo, se alimenta con 12V la etapa de salida, pero el módulo permite ser alimentado con una tensión de hasta 36V, en caso de que se quiera reemplazar el motor por uno de mayor potencia. Además, el módulo está preparado para incluir un segundo motor, en caso de que la empresa lo requiera en un futuro, aunque esto implicaría un cambio en la etapa de control. Cabe agregar que en ese caso, la superficie de disipación podría requerir un aumento, por lo que se dejó preparado un espacio a la medida de disipadores estándar tipo U, que al incluirlos aumentan dicha superficie considerablemente.

III.2.3.c Sensores Laterales

Si bien no se trabajó específicamente sobre los sensores, se colaboró con la empresa en el diseño y construcción de una placa que controla los nuevos sensores incluidos en el dispositivo. La misma se puede ver en la Figura III.9.

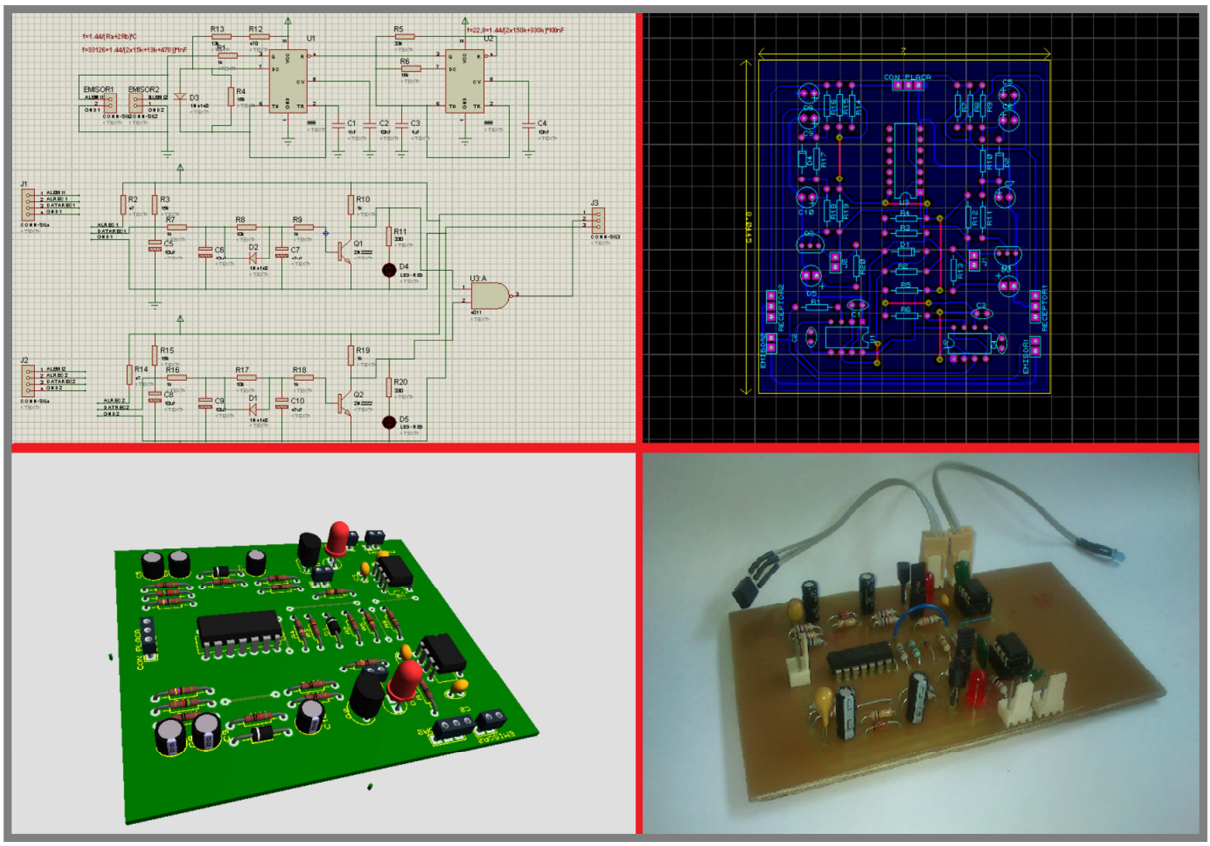

Figura III.9 - Diseño y construcción de la Placa de Sensores Laterales.

Esta placa reemplaza los sensores de contacto actuales por sensores ópticos, los cuales se consideran más adecuados para el uso que se le da al dispositivo. Estos sensores funcionan con tecnología infrarroja modulada. De esta manera se elimina el rose y desgaste propio de los sensores de contacto.

III.3 Desarrollo del *firmware*

Una vez que se tuvo una placa completamente funcional, se comenzó con el desarrollo del *firmware*. Este genera una interfaz amigable para el operario o usuario final que manipule el dispositivo, a la vez que automatiza las tareas que debe realizar el robot.

III.3.1 Objetivos

El objetivo del *firmware* es brindar una interfaz sencilla e intuitiva para el usuario, que permita controlar el dispositivo de manera simple, además de automatizar todas las funciones del robot. Esas funciones son:

- Realizar la puesta a cero del robot en cada ciclo de medición.
- Desplazarse por el caño a medir, detectar el final del caño o un error de calibre y volver automáticamente al extremo de donde se envió.
- Indicar si el caño cumple o no con las especificaciones del calibrado.
- Calcular la longitud del caño, o en su defecto, la distancia a la que se detectó un error de calibración.
- Almacenar la longitud del caño en el banco de memoria.

Como objetivos secundarios se procura cumplir con las siguientes tareas:

- Identificar a qué pozo pertenecen las mediciones.
- Identificar qué operario realizó las mediciones.
- Poder almacenar las mediciones de varios pozos antes de descargar la información.
- Poder descargar la información a una computadora.

Considerando la última función, es necesario realizar un *software* para PC que permita comunicar a esta última con el dispositivo, y descargar las mediciones en un formato que permita su posterior procesamiento. De esta manera se obtienen planillas automáticas, proveyendo al usuario final de información clara y ordenada, y evitando el error por la equívoca interpretación de planillas manuscritas.

III.3.2 Desarrollo

El *firmware* se desarrolló completamente en lenguaje C, utilizando el compilador CCS. Este compilador se integra perfectamente con el software de simulación PROTEUS y con el entorno que provee el fabricante del microcontrolador, el MPLab IDE.

A continuación, en la Figura III.10, se muestra un diagrama en bloques general del *firmware*.

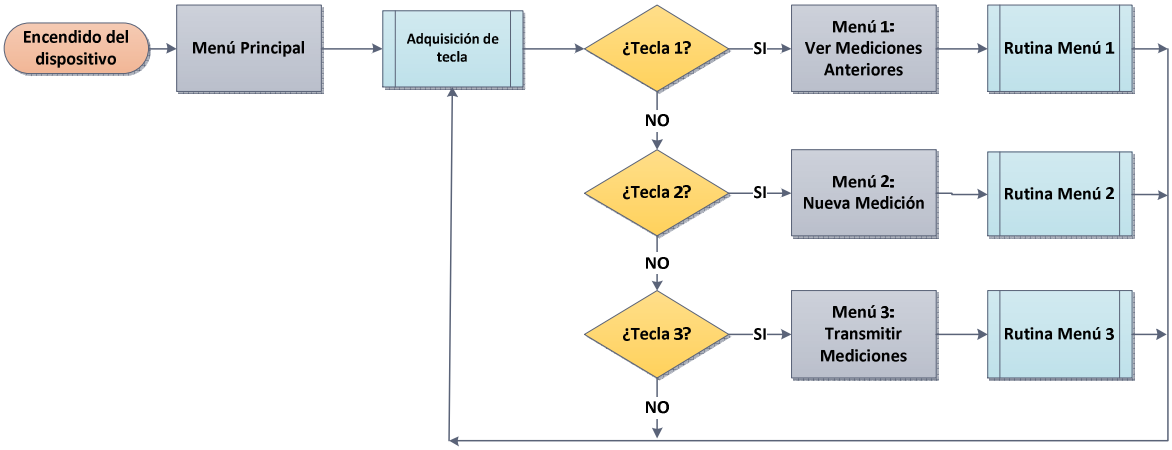

Figura III.10 - Diagrama en bloques del firmware

Este diagrama muestra la forma general en la que se desarrolló el programa. Se pensó el programa en forma de menús con pocas opciones, para no dificultar su manejo. Todas las opciones se despliegan en la pantalla de una sola vez, por lo que no hace falta realizar una navegación por los menús. No se muestra información sobre los cálculos en el *display*, sólo los resultados finales, y algunas variables para seguir el funcionamiento del dispositivo en caso de que éste presente algún error. Cada sub-etapa tiene también su propio diagrama. A continuación se muestran los diagramas en bloques del “Menú 1” y “Menú 2” en la Figura III.11 y Figura III.12 respectivamente.

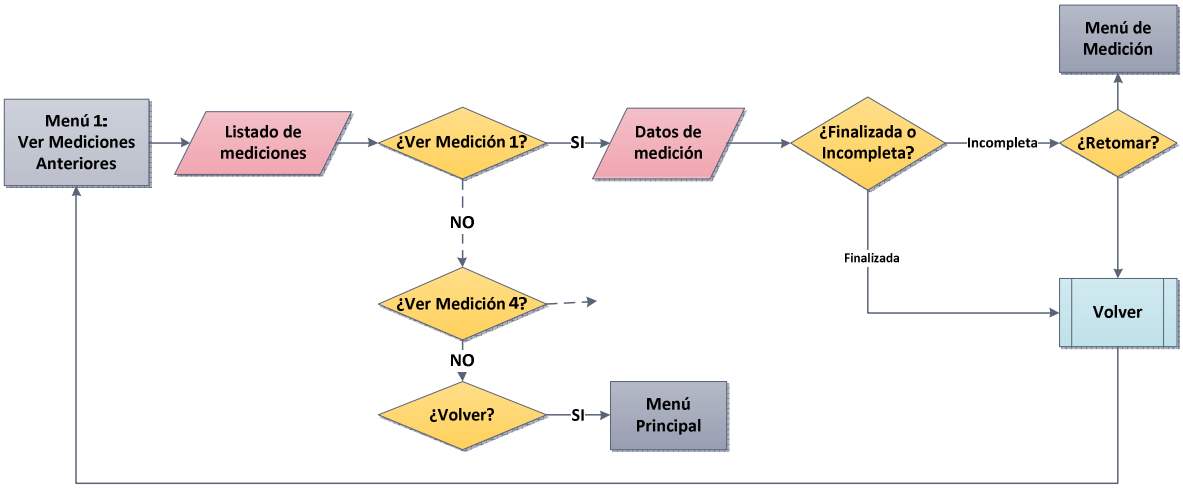

Figura III.11 - Diagrama en bloques del Menú 1.

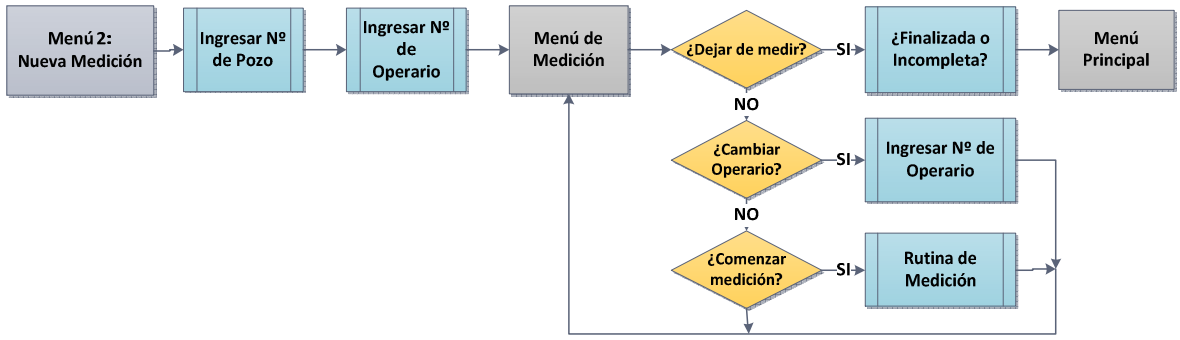

Figura III.12 - Diagrama en bloques del Menú 2.

El “Menú 3” que corresponde a la etapa de transmisión de datos no fue desarrollado, aunque se dejó preparado al dispositivo para poder realizar la misma. Esta etapa formaría parte de un desarrollo posterior.

Además de la organización del programa, un aspecto importante a desarrollar fue la organización de la memoria. Se optó por dividir la memoria en cuatro partes iguales, cada una de las cuales almacena toda la información correspondiente a una medición. Para hacer más organizada la búsqueda de la información almacenada se decidió crear una cabecera, o *header,* para caracterizar los datos importantes de las mediciones. De esta manera se buscó simplificar la obtención de la información correspondiente a cada medición. En la Figura III.13 se muestra una imagen de la distribución de la memoria.

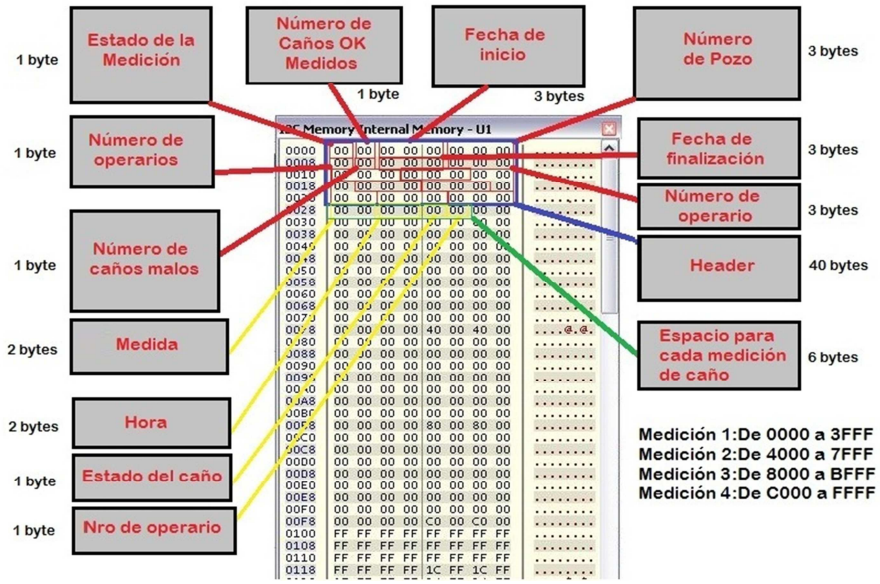

Figura III.13 - Mapeo de la memoria dentro de la EEPROM.

En la imagen se puede ver claramente la organización de la memoria dentro de la EEPROM. El recuadro azul simboliza el *header*. En él se encuentra la siguiente información:

1- **<u>Estado de la Medición:</u>** El firmware prevé dos estados posibles, finalizado o incompleto. En hexadecimal se simbolizaron como F1 y 1C respectivamente. Si bien este estado podría haberse simbolizado con un solo bit, los demás datos a medir requieren más memoria, por lo que igual se requería un byte extra.
2- **<u>Número de Caños OK Medidos:</u>** Este byte está dedicado a indicar la cantidad de caños útiles del lote. Al ser un solo byte, el firmware permite medir hasta 255 caños. Aunque esta cantidad es suficiente para la aplicación, en caso de requerir más cantidad de almacenamiento puede modificarse la distribución del *header*

para incluir un byte extra. De esta manera la cantidad se ve ampliada de 255 a 65535.

3- **Fecha de Inicio:** El firmware reserva tres bytes para colocar la fecha. De esta manera se guarda el día, el mes y el año en cada byte.
4- **Número de Pozo:** Se reservaron tres bytes para indicar el número de pozo.
5- **Número de Operarios:** En este byte se almacena la cantidad de operarios que intervinieron en la medición. Aunque en teoría se pueden almacenar hasta 255 operarios, luego se limitó por firmware que no puedan intervenir más de 5 operarios en un lote de medición.
6- **Número de Caños Malos:** Este espacio indica la cantidad de caños con anormalidades en su diámetro.
7- **Fecha de Finalización:** al igual que para la Fecha de Inicio, se reservan 3 bytes para almacenar la fecha.
8- **Número de Operario:** En este caso, se reservan 30 bytes para almacenar en el *header* datos sobre los operarios que intervinieron en la medición. Se utilizan 6 bytes para cada operario, 3 para indicar el número de operario, y 3 para almacenar información sobre la fecha y hora que dicho operario comenzó a utilizar el dispositivo. En total se puede guardar información sobre 5 operarios, aunque modificando la longitud del *header* podría ampliarse a más.

Luego del *header* comienzan los datos de la medición propiamente dicha. La medición de cada caño individual ocupa un total de 6 bytes. Los primeros 2 bytes se encargan de almacenar la cantidad de pulsos registrados por el *encoder*. La distancia se calcula luego para no perder precisión. A continuación se colocan en los siguientes 2 bytes información sobre la hora y el minuto en que se midió el caño. Este dato, además de servir para control del ente que realice la medición tiene la utilidad de poder generar estadísticas sobre los tiempos que se tardan en realizarlas. Esto podría utilizarse en un futuro para mejorar el dispositivo. El siguiente byte se utiliza para indicar el número de operario que realizó la medición. Cabe aclarar que no se trata del número real de operario, sino un número relativo a esta medición particular, es decir, indicar si fue el primer operario, el segundo operario, etc. En el último byte se indica si el caño tiene alguna anomalía en su

diámetro o no. Las mediciones de los sucesivos caños se almacenan en el siguiente espacio libre.

III.3.3 Funcionamiento

Al encender el dispositivo, luego de mostrar el logo de la empresa y del producto, se muestra el menú principal que se ve en la Figura III.14.

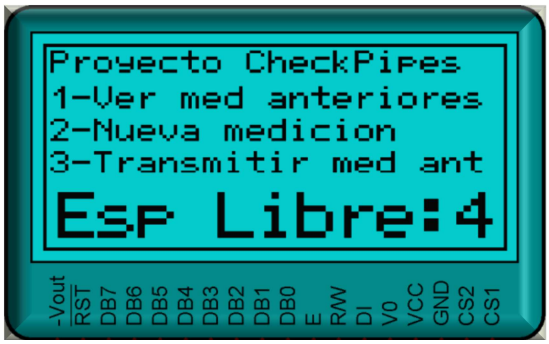

Figura III.14 - Menú Principal.

Utilizando el teclado pueden seleccionarse cualquiera de las 3 opciones. Además se despliega el indicador de espacio libre, el cual indica cuantos espacios están disponibles para iniciar una nueva medición. Esto es importante ya que si el dispositivo no tiene espacio para almacenar una nueva medición, queda inutilizado. Es por eso que el indicador se muestra en forma ampliada.

Si el operador selecciona la opción “1- Ver mediciones anteriores”, el dispositivo entra al correspondiente menú, el cual se muestra en la Figura III.15.

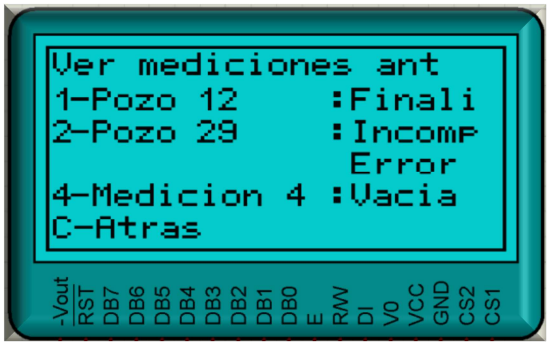

Figura III.15 - Ver mediciones anteriores.

Este menú permite ingresar a una medición en particular y conocer más datos sobre la misma. A su vez, indica anticipadamente si la medición se encuentra finalizada o incompleta, y a que pozo pertenece. Al mismo tiempo permite seleccionar cualquiera de las mediciones utilizando el teclado numérico, y obtener más datos sobre ella.

En el caso de la Figura III.15, se muestran los cuatro casos que pueden darse en este menú. El primero es una medición normal que se encuentra finalizada. Entrando a la opción correspondiente a esta medición (en este caso la opción 1) se ingresa a una pantalla donde se despliegan los datos la misma, como se muestra en la Figura III.16.

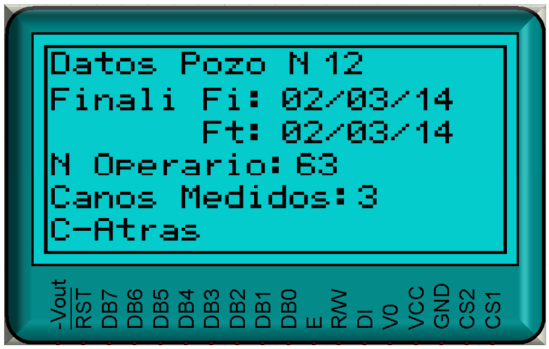

Figura III.16 - Medición Finalizada.

En dicha pantalla se muestra el número de pozo al que pertenece la medición, el estado (Finalizada), la fecha de inicio de la medición (Fi), la fecha de finalización (Ft), el número del operario que realizó la medición, y la cantidad de caños que se midieron. En caso de que más de un operario haya realizado la medición, en lugar de mostrar el número de operario, se escribe la palabra “varios”.

Presionando la tecla “C” se vuelve al menú “Ver Mediciones Anteriores” de la Figura III.15. Si el operario seleccionara la opción 2, entraría a los datos de la segunda medición, que en el caso del ejemplo es la medición incompleta del pozo 29.

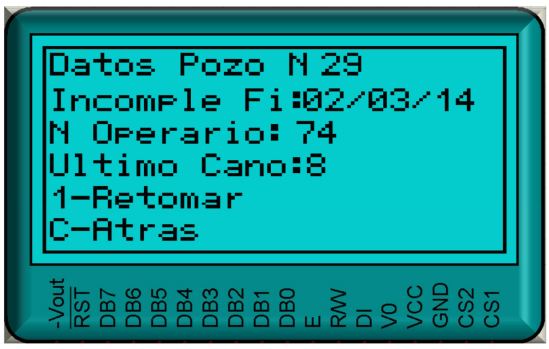

Figura III.17 - Medición Incompleta.

En la Figura III.17 pueden ver los datos antes mencionados. A diferencia de la medición finalizada, en este caso no se muestra la fecha de finalización (Fi), y se reemplaza el total de caños medidos por la información del último caño que se midió. Además se incluye una nueva opción que permite retomar la medición. Este sistema de incluir una opción para dejar incompletas mediciones para luego poder retomarlas se consideró importante. De esta manera permite al operario poder detener el proceso de medición, sea por finalización de la jornada laboral o por inconvenientes climáticos entre otros factores, para luego poder retomarlo desde donde se dejó. Al presionar la opción de retomar, se ingresa al "Menú de Medición", que se mostrará más adelante.

Para los casos restantes de medición vacía y errónea, al entrar se muestra un cartel indicando que se trata de este tipo de mediciones, como se muestra en la Figura III.19 y Figura III.18 respectivamente.

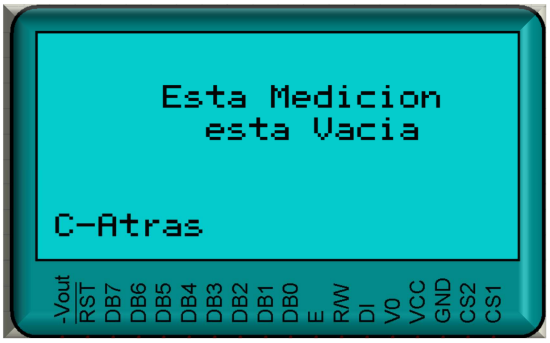

Figura III.18 - Medición Vacía.

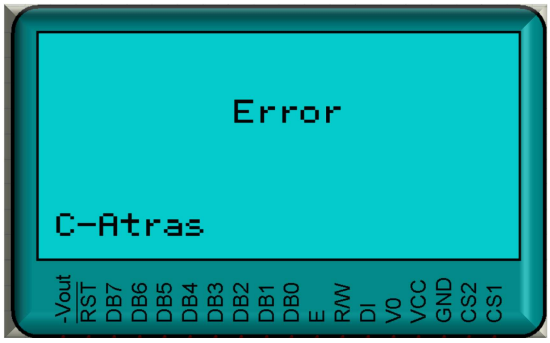

Figura III.19 - Medición Errónea.

Cuando se despliega el cartel de error quiere decir que el *header* se encuentra escrito de manera errónea, por lo que los datos de la medición no se pueden leer correctamente. Esto sirve a la hora de detectar errores de funcionamiento del dispositivo. El caso de la medición vacía significa que el espacio de memoria se encuentra disponible para comenzar una nueva medición.

Volviendo al "Menú Principal" de la Figura III.14, si se selecciona la opción "2-Nueva Medición", el dispositivo comienza la rutina de medición. Se asigna automáticamente el siguiente espacio libre para la nueva medición. Luego de seleccionar la opción, el dispositivo pregunta en primer lugar a que número de pozo corresponde la medición, como se muestra en la Figura III.20.

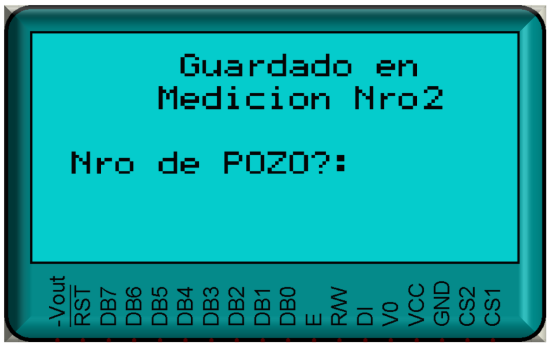

Figura III.20 - Número de Pozo.

El número a ingresar debe ser de 6 dígitos como máximo. Si se quiere ingresar un número de menos caracteres basta con presionar la tecla “Aceptar” luego de ingresar los dígitos correspondientes. Además se puede observar en la imagen que en la pantalla se indica en que espacio de memoria se está guardando la medición.

Luego de ingresar el número de pozo se debe ingresar el número de operario. El procedimiento es exactamente el mismo que para el número de pozo. Esta pantalla se puede ver en la Figura III.21.

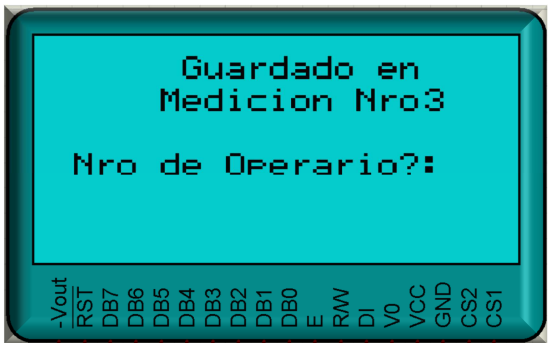

Figura III.21 - Número de Operario.

Una vez ingresado estos dos datos se despliega el menú de medición. Este menú, que se muestra en la Figura III.22, es el que será más utilizado por el operario.

En esta pantalla se despliega el número del último caño que se midió, además de permitir otras opciones para la realización de la medición. Para el caso del ejemplo, la medición recién comienza, por lo que el último caño medido es “Ninguno”. La primera opción es “1-Dejar de medir”.

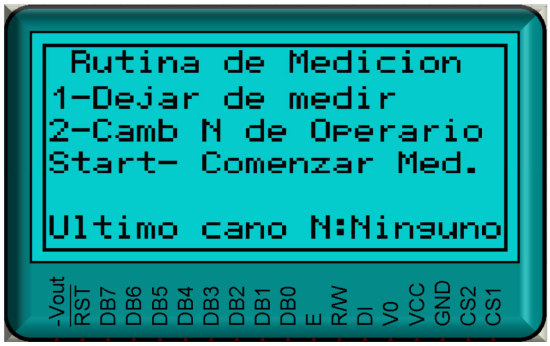

Figura III.22- Menú de Medición.

Si se selecciona esta opción el dispositivo pregunta si la medición se finalizó o quedo incompleta. Esta pantalla se muestra en la Figura III.23.

Figura III.23 - Decisión al dejar de medir.

Como se explicaba antes, esta opción se pensó para el caso en que la medición se detenga, sea por que se terminó de medir el lote de caños correspondiente, o bien porque algún evento externo requiere que se suspenda momentáneamente, para retomarla en un momento futuro.

En el caso de la opción “2-Cambiar Numero de Operario”, el dispositivo vuelve a la pantalla de la Figura III.21, donde permite ingresar un nuevo operario, el cual continuará con la medición. Esta opción se pensó con la intención de poder hacer relevos de personal fácilmente, además de tener un control sobre quien estaba a cargo de la herramienta en cada momento de la etapa de medición. El dispositivo permite cambiar hasta seis veces de operario con el *header* actual, ya que normalmente la medición la realizan entre uno y dos grupos de trabajo. Igualmente este parámetro puede ser modificado fácilmente modificando la longitud del *header* y haciendo una revisión del programa principal.

Por último se encuentra la opción “Start- Comenzar Medición”. Esta opción es la que tuvo aparejada la mayor complejidad en la programación, ya que conjuga el uso de los

actuadores, sensores, accesos y escrituras en memoria, entre las tareas principales. Antes de continuar con la explicación de esta función, es útil recordar la disposición de los sensores y el botón de *Start* en el dispositivo, los cuales se muestran en la Figura III.24.

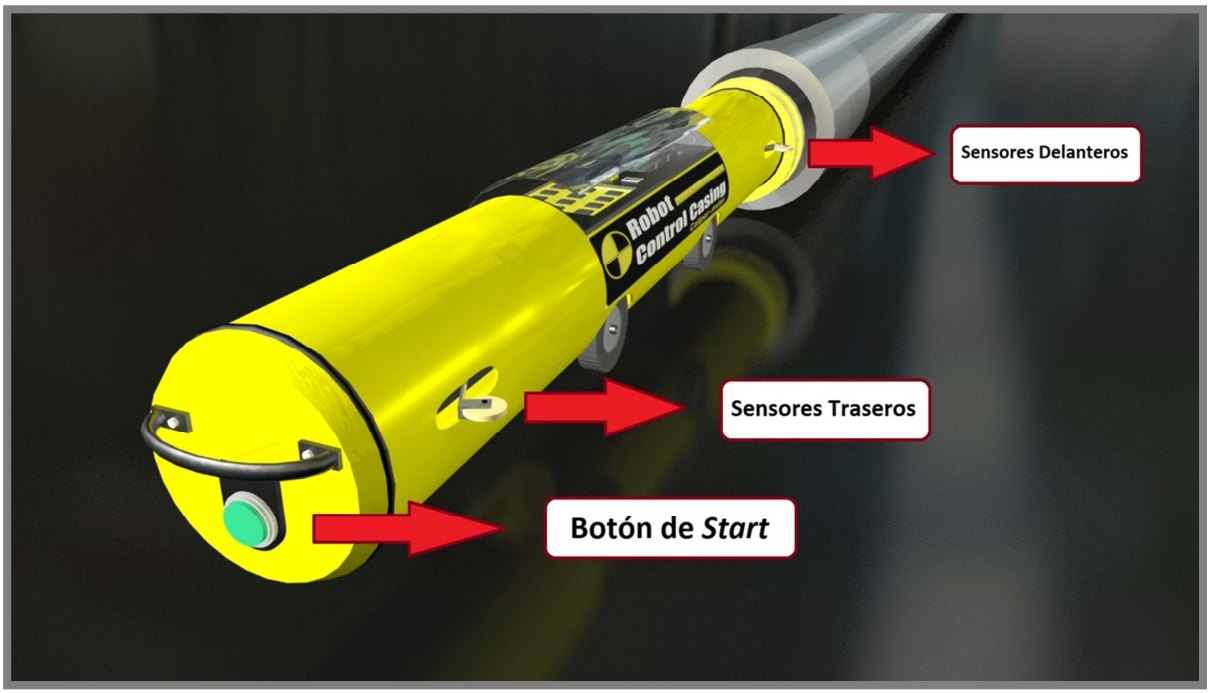

Figura III.24 - Ubicación de componentes en el dispositivo robótico.

Para que el dispositivo comience a medir, no basta con presionar el botón *Start*, sino que además, los sensores laterales traseros y delanteros tienen que estar todos presionados (en el caso del prototipo son *micro-switches*). Al estar en esta situación se considera que el dispositivo se encuentra completamente dentro del caño. Una vez que se encuentra dentro del caño, al presionar el botón *Start* el mismo comienza una etapa de "puesta a cero". El motivo de esta etapa es que no se conoce *a priori* cuan adentro del caño se encuentra el dispositivo. Por lo tanto, simplemente se activa el motor hacia atrás hasta que los sensores de la parte trasera salen del caño por el mismo extremo que se introdujo.

Una vez realizado esto, el robot sabe que se encuentra exactamente en un extremo del caño, por lo que pone en marcha el motor hacia delante, y comienza a contar los pulsos que entrega el *encoder*. Mientras realiza esta acción muestra en la pantalla el resultado parcial de la medición, como se ve en la Figura III.25.

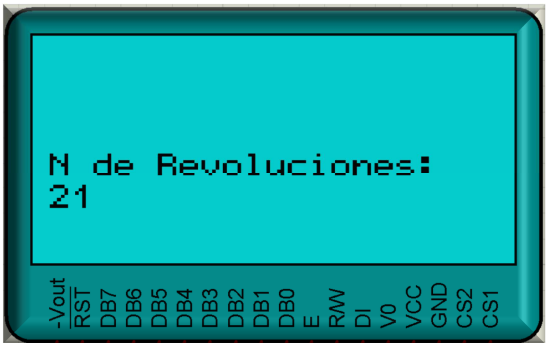

Figura III.25 - Resultado parcial de la medición.

Aunque este dato no puede ser visualizado por el usuario fue útil para la depuración de errores durante el desarrollo. La imagen es ilustrativa, ya que los datos desplegados en esta etapa fueron variando dependiendo de lo que se quiso probar en cada momento. La distancia recorrida se calcula con las siguientes fórmulas:

$$D_p = \frac{d.\pi.\rho}{24}$$

$$D_F = D_p + L_d + L_t - ML$$

Dónde:

- D_p: Distancia parcial; distancia recorrida por la rueda acoplada al *encoder*.
- d: Diámetro de la rueda acoplada al *encoder*.
- π: Número pi.
- ρ: Cantidad de pulsos contados por el *encoder.*
- L_d: Distancia de la rueda del *encoder* hasta el sensor delantero.
- L_t: Distancia de la rueda del *encoder* hasta el sensor trasero.
- ML: Pérdida por enrosque (make-up loss).
- D_F: Longitud total del caño.

De esta manera se calcula la longitud del caño utilizando el perímetro de la rueda y la cantidad de pulsos que detectó el *encoder*. El factor 24 se debe justamente a que el *encoder* genera 24 pulsos por vuelta. Además, para el cálculo de la longitud final se pueden observar los factores de corrección L_t y L_d. Estos deben ser incluidos ya que de otra manera

no se estaría considerando los fragmentos de caño que no recorre la rueda del *encoder* debido a su ubicación en el dispositivo robótico.

Además, según especifica el Manual de Uso de *Casing* y *Tubing* (Tenaris 2007), para calcular la longitud efectiva del *casing* se debe restar la pérdida por enrosque (ML) como se ve en la Figura III.26, ya que ese fragmento de caño estará superpuesto al siguiente caño.

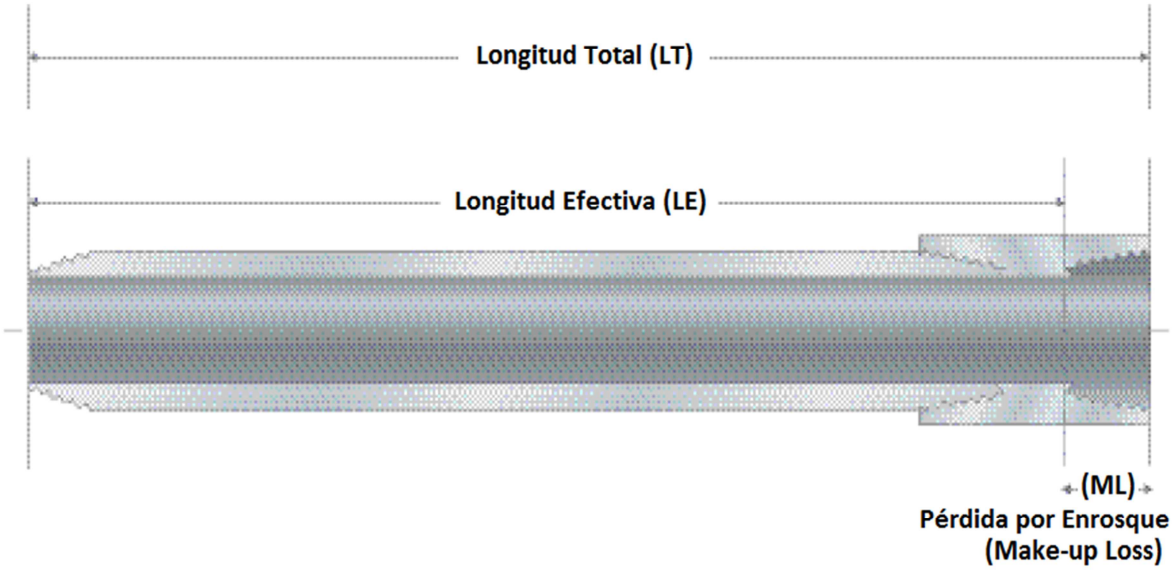

Figura III.26 - Pérdida por Enrosque y Longitud Efectiva.

Durante un proceso de medición normal en el que el caño mantenga las especificaciones del diámetro, el robot va y vuelve por el caño, realizando la medición de longitud en ambos casos. Si las longitudes de ida y vuelta difieren en más de cierto valor, la medición se considera errónea y pide al operario que repita la medición. Si estas se mantienen dentro de cierto margen la medición se considera correcta, se almacena, y se aumenta en uno el indicador de "Último Caño" en el Menú de Medición (Figura III.22).

En caso de que el caño no cumpla con las especificaciones del diámetro, el dispositivo informa al operario de esta situación, en cuyo caso el caño debe ser marcado con una X para no ser utilizado. En esta situación el indicador "Último Caño" no avanza, ya que ese caño no va a ser utilizado posteriormente. Igualmente, el robot almacena la distancia a la que se encontró la anormalidad, en caso de que la empresa propietaria desee cortar y roscar el caño averiado para su reutilización. Por último, en el Menú Principal de la

Figura III.14, se puede observar una última opción “3- Transmitir Mediciones”, que al seleccionarla se despliega en la pantalla el mensaje que se puede ver en la Figura III.27.

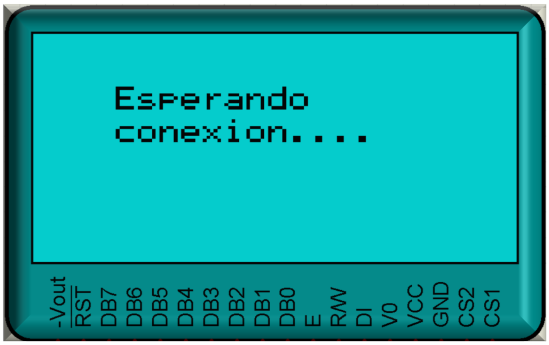

Figura III.27 - Mensaje al seleccionar: Trasmitir Mediciones.

Luego envía todos los datos almacenados sobre los tubos a través de la EUSART. Se realizaron pruebas sobre la comunicación, obteniendo resultados exitosos, pudiendo descargar los datos a la computadora, utilizando el *software* DockLigth, el cual permite enviar y recibir datos por los puertos tipo COM.

IV. Mejoras Futuras

A lo largo del desarrollo del controlador, se observaron ciertos aspectos que, aunque escapaban al alcance del proyecto, se consideraron interesantes para ser desarrollados posteriormente. A continuación se nombran los tres más importantes:

1- Se propone realizar un interfaz para la computadora que mejore la experiencia del usuario al momento de descargar los datos a una computadora.

2- Resultaría útil contar con una función que indique el nivel de batería del equipo en el *display*, de modo que el operario cuente con información adicional al momento de utilizar el dispositivo.

3- Se tuvo en cuenta la opción de incorporar una interfaz serie *bluetooth*, manteniendo la comunicación serie que provee el instrumento, pero eliminando la necesidad de cable.

V. Conclusión

Luego de la implementación del sistema de control del dispositivo robótico para medición de longitud y diámetro interno de tuberías, se llegó a las siguientes conclusiones finales:

- ✓ Integración de tareas:

Ahora, en lugar de realizar dos operaciones de medición por separado, se cuenta con un dispositivo que las realiza de manera conjunta.

- ✓ Reutilización de Material:

Al realizar la medición de longitud y calibrado simultáneamente, permite conocer la distancia a la que se detecta una anomalía en el diámetro, posibilitando la reparación de los tubos.

- ✓ Mayor calidad de trabajo:

Al reemplazar el método tradicional de calibración, evita los riesgos inherentes al mismo y alivia la fatiga del operario.

- ✓ Reducción de tiempos y personal:

El dispositivo realiza de manera ágil las tareas de medición, asistido únicamente por un operario.

- ✓ Minimización de errores:

Al realizar automáticamente las planillas de datos al descargar las mediciones a una computadora, disminuyen los errores por interpretación errónea de la información.

Por lo tanto, se considera que el producto final motivo de este proyecto resulta en una solución para las problemáticas observadas, posibilitando un potencial ahorro de personal, tiempo y dinero para la industria petrolera, a la vez de mejorar la seguridad de los operarios y minimizar errores por la mala manipulación de la información.

VI. Bibliografía

1. American Petroleum Institute. *Specification for Casing and Tubing, API Specification 5CT.* ISO 11960:2004, Petroleum and natural gas, 2006.
2. Benchimol, Daniel. *Microcontroladores.* Buenos Aires: Users, 2011.
3. Cervantes de Anda, Ismael, y Juan Carlos Martínez Díaz. «PIC16F874/877 Fundamentos - Diseño - Programación.» *Club Saber Electrónica*, 2009.
4. Cuenca, Andres. *Foros de Electrónica.* 20 de 3 de 2005. http://www.forosdeelectronica.com/f16/encoders-informacion-tecnica-25/ (último acceso: 26 de 02 de 2014).
5. García, Javier. *Aprenda lenguaje ANSI C como si estuviera en primero.* San Sebastián: Universidad de Navarra, 1998.
6. López, Andrés Cánovas, y Víctor Dorado. *Manual de Usuario del Compilador PCW de CCS.* s.f.
7. MCBtec. *MCBtec.* 2004. http://www.mcbtec.com/Funcionamiento_Encoder.pdf (último acceso: 26 de 2 de 2014).
8. Society of Petroleum Engineers. *http://www.spe.org/.* s.f. http://petrowiki.spe.org/Casing_and_tubing (último acceso: 25 de 7 de 2013).
9. Tenaris. *Manual de Uso de Casing y Tubing.* Campana: Tenaris, 2007.
10. Zabala, Gonzalo. *Robótica, Guía Teórica y Práctica.* Buenos Aires: Users, s.f.