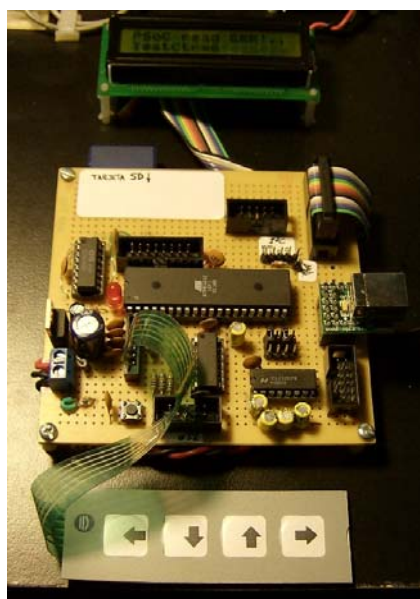


Agencia Nacional de Promoción Científica y Tecnológica
Fondo Tecnológico Argentino - FONTAR - CONVOCATORIAS REGIONALES DE
PROYECTOS DE INNOVACION TECNOLOGICA

ANR N° SC002/2003

“Sistema de Adquisición de Datos (Data Logger) de Bajo Costo y Arquitectura Abierta para Aplicaciones Ambientales y de Energía”

TERCER INFORME TÉCNICO –FINAL – AGOSTO 2009
REVISION 1.2



CONTENIDO:

- 1) INTRODUCCION
- 2) ETAPAS CUMPLIDAS
- 3) TRABAJOS REALIZADOS SOBRE SISTEMA DL51/AVR
- 4) APLICACIONES IMPLEMENTADAS
- 5) JUSTIFICACION DE LAS COMPRAS REALIZADAS
- 6) CONCLUSIONES

1. INTRODUCCIÓN

El presente proyecto tuvo como objetivo el desarrollo de un sistema de adquisición y almacenamiento de datos (Data Logger) para aplicaciones ambientales y energéticas, con características de bajo costo, facilidad de uso y arquitectura abierta, y con la posibilidad de realizar y almacenar mediciones de temperatura, humedad, presión atmosférica, velocidad y dirección de viento, radiación y parámetros eléctricos de sistemas de energía renovable (tensión, corriente, potencia), con períodos de almacenamiento y funciones estadísticas configurables. La memoria de almacenamiento es de tipo no volátil. Se desarrolló un primer sistema que trabaja con almacenamiento EEPROM serial que se ha probado exhaustivamente, y se culminó el desarrollo de un sistema más avanzado que utiliza almacenamiento en tarjetas MMC/SD (Secure Disk) o miniSD comerciales.

El mercado principal que se busca cubrir es el académico-científico y el industrial. La arquitectura abierta y modular le otorga un valor educativo y permite una mejor evaluación previa de la compra a los principales interesados. La salida principal del proyecto fue la construcción de dos prototipos: uno de ellos con procesador de la familia genérica 8051 ó Atmel AVR (derivado del 8051), de tecnología ampliamente probada y sencilla comprensión; y otro basado en tecnología PSoC (Cypress Semiconductor). Por cuestiones de costo y necesidad estuvo lista, probada y con circuitos impresos en producción antes la variante Cypress, que integra funciones programables de hardware como así también memoria reprogramable en un sólo chip de bajo costo. En el interín, la familia 8C2 de Cypress evolucionó hacia controladores más avanzados, respecto a los primeros 26443 que se mostraron en el informe de octubre 2004, incluyendo la línea 27xxx y luego la 29xxx. Por las especiales características y la mayor cantidad de memoria y módulos configurables, se eligió trabajar exclusivamente con la línea 29466 de costo levemente mayor, que es la que equipa los sistemas DLCy actuales (Figura 1.1).



Figura 1.1 izq- Vista de Placa DL-Cy para Freno/1 y sistemas EM/12 (PERMER,der.) con DLCy

Para los equipos más avanzados que utilizan tarjetas de memoria SD, se ha llevado adelante la tarea de diseño de la placa utilizando la arquitectura Atmel AVR, específicamente con el controlador ATmega32 ó ATmega644P (Figura 1.2). Se eligió esta variante por el reducido costo de sus herramientas de software, tecnología más adaptada a lenguajes de alto nivel (C) y una amplia y muy activa comunidad de usuarios a nivel internacional. Asimismo, se cuenta con importante experiencia de programación de módulos basados en ATmega128 que utilizan bibliotecas comerciales de acceso a tarjetas de memoria SD, y que son aplicables en gran medida a los sistemas mas reducidos ATmega644P (ya están en etapa preliminar los integrados ATmega1284P, compatibles con los 644P, que duplican su memoria).

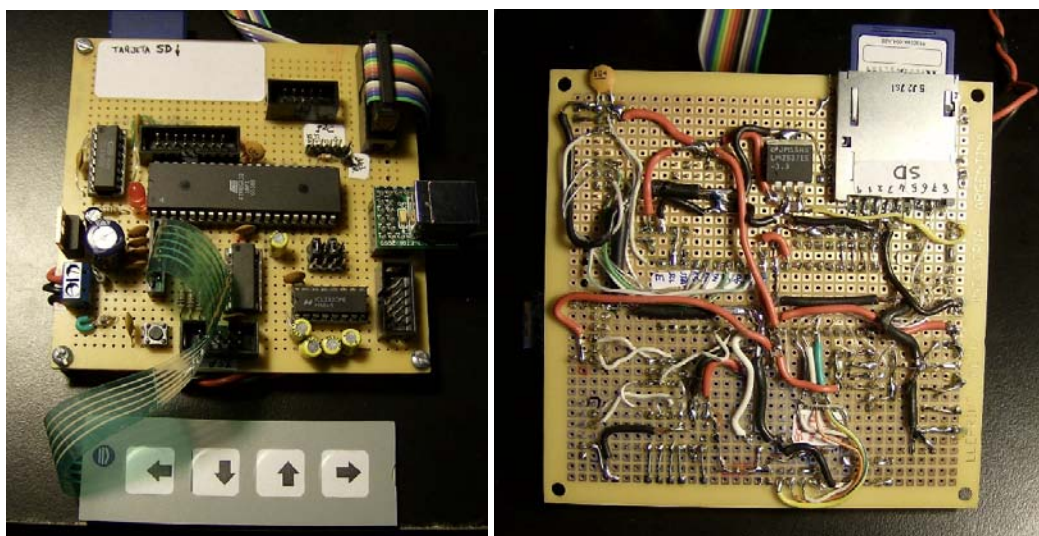


Figura 1.2 – Vistas Frontal y Posterior del Prototipo CL2b ensayado

Esta placa tiene la particularidad de poder utilizar la CPU DLCy existente como “front-end” analógico o de E/S discreta (Figura 1.3, izq), recibiendo los datos ya digitalizados a través de un enlace serial interno y funcionando como unidad de almacenamiento e interfase con el usuario. Además permite la conexión USB con una PC (Figura 1.3, der). El programa interno de ambos prototipos es un sistema de módulos escritos en lenguaje C, que junto con la documentación en CD y el software para PC de manejo serán las salidas comerciales del proyecto.

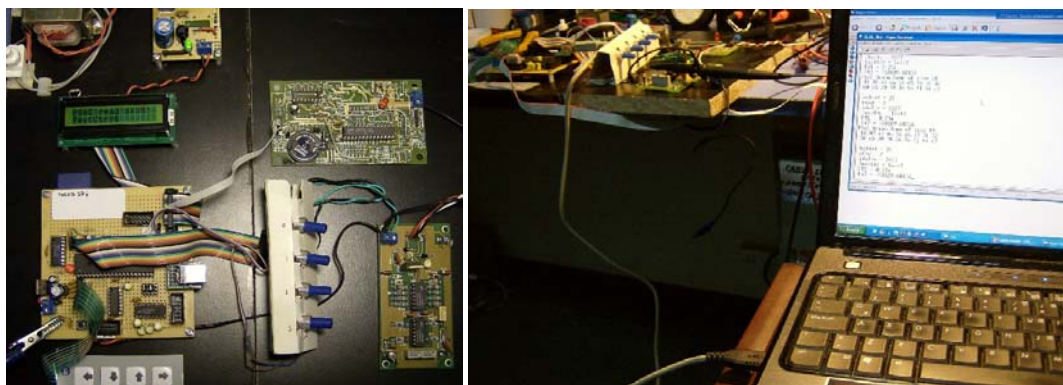


Figura 1.3 – Ensayos del Prototipo CL2b en conexión a DLCy y medición Temperatura, Enlace USB con PC

2. ETAPAS CUMPLIDAS

Las etapas planteadas en la formulación inicial del Proyecto fueron las siguientes:

ETAPA N°	DESCRIPCION
A	Diseño inicial de los módulos de Software Interno, Ensayos Nivel 1 de PSoC, con ICE ya adquirido Diseño Preliminar DL51/AVR, Ensayos con protoboard de DL51/AVR
B	Ensayo de los módulos de Software Interno, continuación ensayos protoboard DL51/AVR, Inicia Placa diseñada para el DL51/AVR (CL2b).
C	Port del Software Interno a particularidades del PSoC (Cypress). Continuación Placa CAD, armado placa prototipo DL51/AVR, Inicio ensayo DL51/AVR con sensores, Inicio Software para Host PC.
D	Ensayo del Port en PSoC. Placa diseñada por CAD para DLCy, envío a fabricante, armado y prueba con sensores. Finalización Software Host PC.
E	Test del sistema completo. Documentación e Informe.

Como se indicó en (1.), dichas etapas sufrieron cambios impulsados principalmente por el requerimiento de productos basados en PsoC / Cypress de menor costo, y que se complementaron en gran medida con solicitudes de equipos que integraban CPUs comerciales de mayor tamaño (ATMega128) con capacidad de almacenamiento en tarjetas SD. De allí surgió la importancia de que el prototipo DL51/AVR (finalmente llamado **CL2b**) contara con capacidad de almacenamiento en tarjetas SD/MMC, lo cual finalmente se pudo implementar. Un detalle de las tareas cumplidas en cada etapa se resume a continuación:

- A) Cumplida en todo lo referido a PsoC. Se ha trabajado intensamente en los ensayos y distintas variantes de la placa impresa (DLCy-F) y se ha fabricado asimismo una placa DLCy-AUX que realiza las interfases más comunes para los tres sistemas que la utilizan (Freno / LoggerC3x – Temperatura, medidor de energía en CC - EM/12). Para el cuarto sistema (Meteo-Front/End) que utiliza la CPU/DLCy como convertidor y terminal de transmisión de datos RS485, se produjo una placa especial MetAnlg (Informe N°2). Se ha concluido el diseño y producción de la placa DL51/AVR, denominada **CL2b**, que incorpora almacenamiento SD y comunicación I2C con una DLCy esclava, y finalmente con una plaqueta para montaje en gabinetes DIN. (grado finalización: 100%)
- B) Los módulos de software interno tienen una madurez considerable para todo lo relacionado con los sistemas PSoC, especialmente para los productos terminados y en pleno funcionamiento Freno, Meteo Fr/End y equipo EM/12. Los módulos de software para DL51/AVR se han ensayado exhaustivamente en la placa **CL2b** utilizando un procesador Mega32. Las unidades de producción con la placa final utilizarán un procesador Mega644P que difiere en la mayor capacidad de memoria y módulos, pero es 100% compatible en cuanto a software. (grado finalización: 100%)
- C) Esta etapa, se encuentra completa en todo lo relativo a Cypress según ya se mencionó en B), y también finalizada en lo relativo al equipo DL51/AVR. Se han realizado pruebas exhaustivas de almacenamiento SD/MMC como equipo independiente y con un DLCy como esclavo, en un sistema conectado por bus I2C. En cuanto al software HOST, se desarrolló en forma completa utilizando Borland C++ Builder (grado finalización: 100%)
- D) Todo lo que se refiere al PSoC / Cypress se encuentra con un buen grado de madurez debido a su aplicación en distintas variantes de conexión, utilización de módulos internos y rutinas especiales en C. Las primeras placas DLCy se entregaron a mediados de 2005 y las auxiliares AuxDLCy se empezaron a utilizar en agosto de 2006, ambas realizadas con TangoPCB y fabricadas por el proveedor habitual,

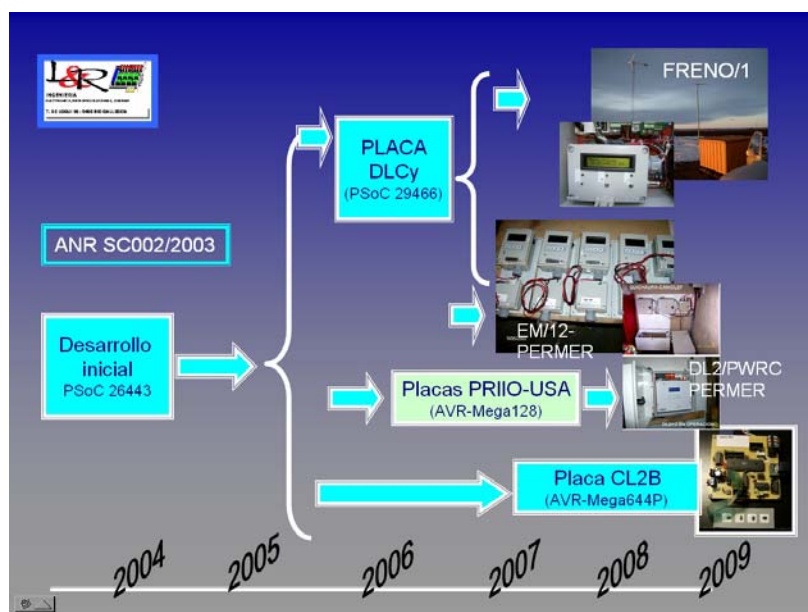
Ernesto Mayer S.A. de Buenos Aires. Para las placas posteriores a 2007 se utilizó el software Protel. Asimismo, se concluyó un programa de configuración más amigable bajo Windows en Borland C++ Builder. (grado finalización: 100%)

- E) El sistema se considera completo con el desarrollo y prueba de la variante DL51/AVR basado en la versión de mayor capacidad Mega32/Mega644P de AVR. Esta es la etapa en la que ha habido más trabajo en el período reciente 2008 y 2009. El prototipo CL2b es capaz de almacenar en tarjetas SD/MMC series de tiempo de variables analógicas en forma confiable, incluyendo un TS (TimeStamp), en archivos de formato CSV legibles directamente en una PC o notebook con lector de tarjetas flash convencional. Se indican las mejoras que se utilizarán en las variantes comerciales, no implementadas en la fase de prototipo. (Grado de finalización 100%).

En todo lo posible, se ha utilizado la metodología indicada en los Informes 1 y 2, con énfasis en lo planteado por [O'Cull 2002]¹, para el **Desarrollo de un Proyecto con Microcontroladores**, utilizando las distintas Fases:

- I.) Fase de Definición.
- II.) Fase de Diseño (Hardware y software)
- III.) Fase de Definición de Tests
- IV.) Fase de Construcción del Prototipo y Prueba inicial
- V.) Fase de Integración y Desarrollo del Software.
- VI.) Fase de System Test.

Evolución gráfica del Proyecto e hitos:



¹ "Embedded C Programming and the Atmel AVR" - Barnet, R.; Cox, S.; O'Cull, L. - Thomson/Delmar Learning 2002, ISBN 1-4018-1206-6

3. TRABAJOS REALIZADOS SOBRE SISTEMA DL51/AVR (CL2b)

3.1 Placa Prototipo CL2b - DL/51AVR.

La placa Prototipo CL2b (CL2b-Proto) se utilizó para validar las funciones diseñadas para el logger DL51/AVR según la concepción de tener un registrador mas avanzado con amplia capacidad de almacenamiento a través de tarjetas extraíbles tipo MMC/SD. Dichas funciones son sobre todo las relativas a la captura de datos analógicos, la comunicación a través del puerto serie RS232 o a través de USB, el funcionamiento del teclado, el funcionamiento del reloj de tiempo real (RTC), el display LCD y el almacenamiento confiable en tarjetas MMC/SD. Las pruebas realizadas en el marco del presente proyecto han permitido determinar evaluar estas capacidades y sentar las bases para un modelo comercial con unos pocos agregados. En la Figura 3.1 puede verse un diagrama en bloques de esta versión.

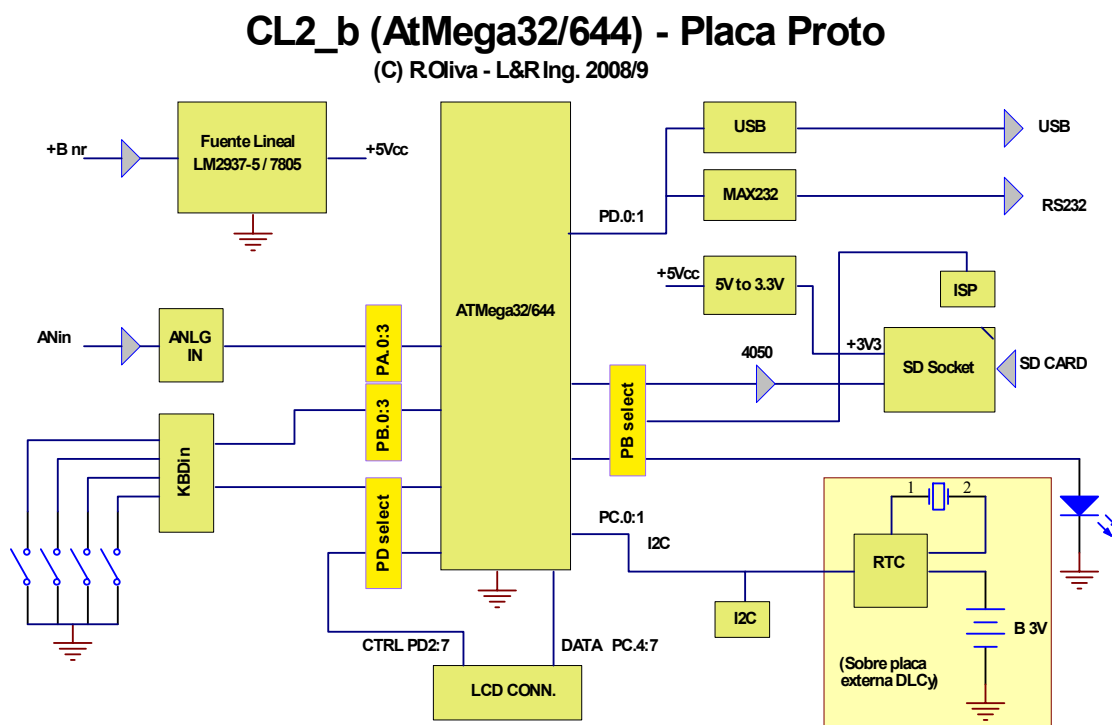


Figura 3.1 – Circuito Principal de CL2b-Proto

La versión ensayada para el presente proyecto contaba con un procesador ATmega32-16PI, el cual cuenta con 32KB de memoria Flash y un solo puerto serie. Es pin a pin compatible con la versión ATmega644P, que cuenta con el doble de memoria interna y un puerto serie adicional, además de tener un consumo menor.

Los componentes de dicho circuito se analizarán por partes, en la siguiente secuencia:

- 3.1.a) Fuente de Alimentación
- 3.1.b) Entradas Analógicas
- 3.1.c) Interfase para teclado de membrana
- 3.1.d) Display LCD
- 3.1.e) Interfase I2C y Reloj de tiempo real.
- 3.1.f) Interfase a tarjetas SD/MMC
- 3.1.g) Interfase Serie RS232 y USB.

3.1.a) Fuente de Alimentación

La fuente de alimentación debe proveer los +5V regulados para la operación de la placa (VCC), y además proveer una salida filtrada (AVCC) para la alimentación del conversor A/D incluido en el microcontrolador. Se privilegió para el prototipo la simplicidad y se utilizó un regulador del tipo lineal convencional (ó LDO) para obtener la tensión VCC estabilizada (Figura 3.2). La salida AVCC proviene de un simple filtro RC de primer orden.

CL2-B ProtoBoard - (c) 2009 / LyR Ingenieria

Power Supply

rev.8 08-2009 / (c) R.Oliva

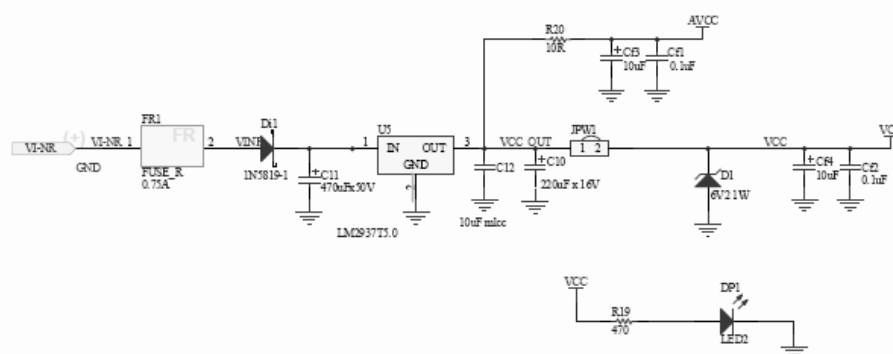


Figura 3.2 – Fuente de alimentación para CL2b – Proto

3.1.b) Entradas Analógicas

Para el sistema en cuestión se decidió dejar previsto un conector para las 8 entradas PA.7/PA.0 sin agregar filtrado o amplificación, que se deja como agregados posibles para una placa analógica exterior. El conversor A/D es de 10bits, por lo cual los rangos son de 0-1023 enteros. En el sistema implementado, el conversor funciona por interrupción, y la rutina de servicio reinicializa la conversión una vez que se ha almacenado el valor leído. Se cuenta además con un multiplexor interno al AVR que permite seleccionar el canal correspondiente a la conversión en curso (Figura 3.3).

3.1.c) Interfase para teclado de membrana

Se utilizó un teclado de 4 llaves de producción nacional (de Microteclados S.A., Buenos Aires), que permiten resolver con simplicidad el manejo de menús en displays alfanuméricos típicos. Los teclados se proveen con su conector, y el circuito consiste de una resistencia de pull-up más un capacitor a 0V cuya función es reducir los rebotes, para cada contacto. Los 4 contactos se conectan a las entradas PB.0-PB.3. Los pines pueden definirse como entidades separadas en el compilador HPInfotech (Codevision) utilizado, por lo cual la tarea de realizar la rutina de software que verifica el estado de las teclas no tiene grandes dificultades. Se presenta un detalle del sistema en la Figura 3.4 (der.)

CL2B Proto Board - (c) 2009 / LyR Ingenieria
rev.5 - 08-2009 / (c) R.Oliva
CPU+Peripheral

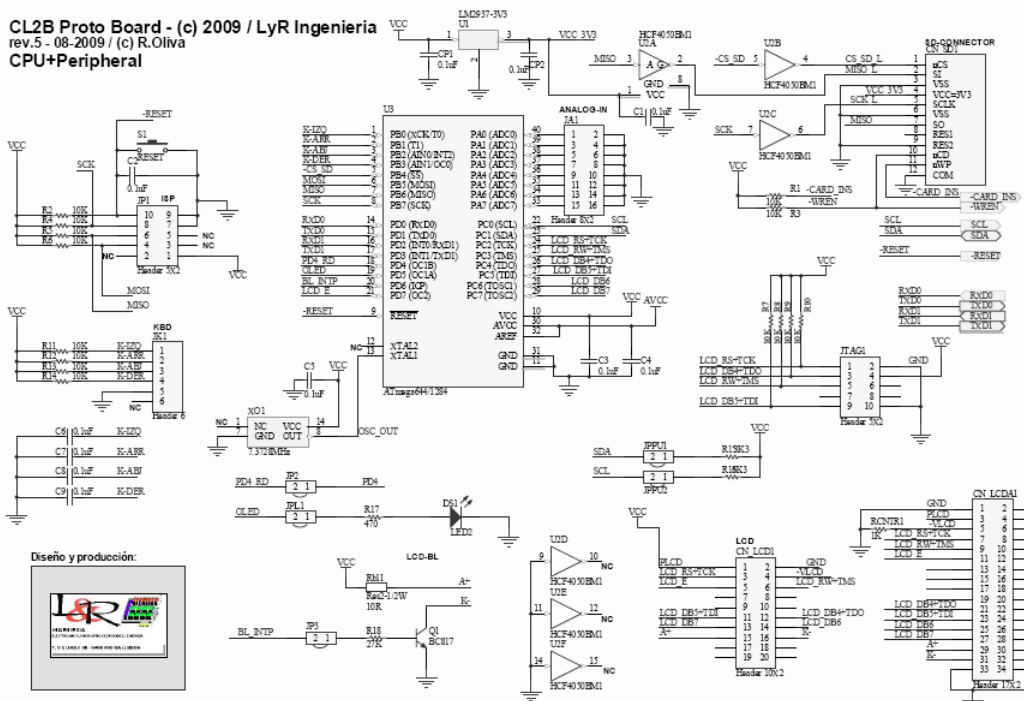


Figura 3.3 – Diagrama CPU+Periféricos para CL2b – Proto

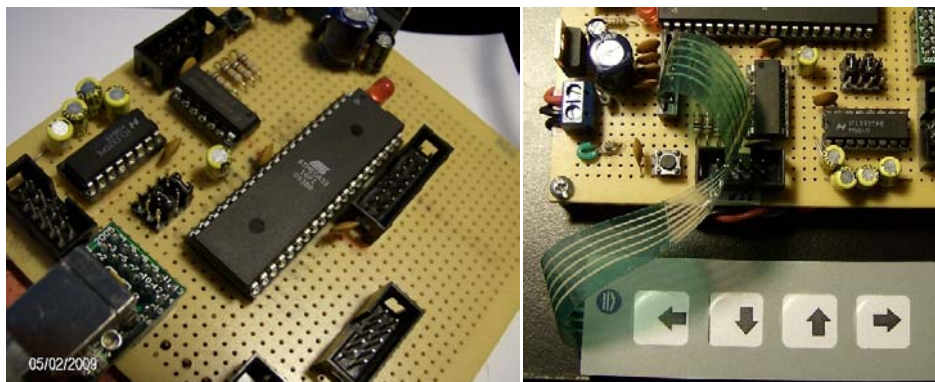


Figura 3.4 – (izq) Detalle de la CPU CL2b – Proto en primeras pruebas, luego con teclado instalado (der)

3.1.d) Display LCD

Se diseñó el circuito para poder utilizar displays alfanuméricos con la interfase usual de 4 líneas de datos y 3 de control, en tamaños desde 1x16 a 4x40. La interfase se implementa con líneas del controlador. Al ser prioritaria la utilización del puerto alto B para la interfase SPI que implementan las tarjetas SD, y asimismo importante implementar la interfase I2C (o TWI según la denominación utilizada por Atmel para no incurrir en problemas de patente con Philips) por hardware (PC.0/PC.1) se debió utilizar un conjunto combinado de líneas de los puertos C y D (Figura 3.3), lo cual hizo algo dificultosa la adaptación de los “drivers” existentes. Una vez realizado esto, el display se comportó a la perfección (Figura 3.5).

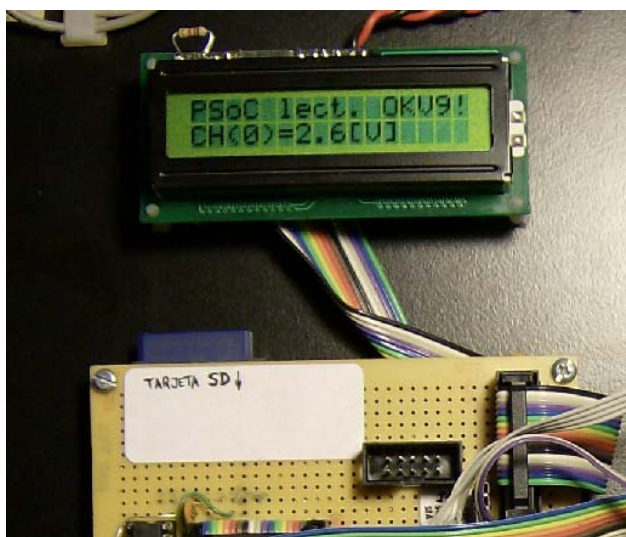


Figura 3.5 – Interfase a Display LCD

3.1.e) Interfase I²C y Reloj de tiempo real.

La interfase I²C de Philips (o TWI para Atmel) otorga una gran flexibilidad para la conexión de dispositivos de bajo nivel, con un robusto formato de dos líneas y retorno. Hay una gran cantidad de dispositivos que utilizan este sistema que consume escasos recursos y energía, y proporciona una conexión confiable. El “lado oscuro” es la complejidad de programación, algo que ya se relató en los primeros informes sobre el DLCy que describieron la comunicación del PSoC con el PCF8563 (Reloj de tiempo real o RTC) y la memoria serial LC256, ambas de interfase I²C. En nuestro caso, la reimplementación del código para el RTC no presentaba mayor dificultad, salvo por el hecho de que se eligió utilizar una placa DLCy (Figura 3.6) para conectar dicho dispositivo al CL2b, y de paso utilizar la función “I2C-Hardware-Slave” del PSoC. Se encontraron problemas en la conexión (el camino de GND entre ambas placas debe ser de baja resistencia), y en el software. En un caso, se trató de un “bug” presente en un código comercial suministrado para las placas mas grandes Mega128 por la firma PRIIO.com, que funcionaba en un 99% de los casos pero tenía desactivado un “timer de seguridad” (nunca se incrementaba). Se reutilizó este código, y una de las rutinas de menor nivel, *twi_step()* es la que se transcribe a continuación y presentaba un error aleatorio dejando al bus colgado, por ejemplo ante ruido o desconexión de la segunda placa. Al realizar la corrección desapareció el error. Esta rutina hace que actúe el CL2b como Maestro en la comunicación I²C.

```
17.5.09 Modify options.h to access twi-cl2-2.c/.h,
Add label TIMERMAT for TWI Timer in TWIStep() routine, and increment this timer.
/*****
** Sends a two wire interface command - para CL2b a 184kHz
**
** Parameters: twcr_mask, Command to be sent
**             status, Flag to see if we need to wait for the status
**
** Returns: 0 - Command sent successfully
**          1 - An error occurred
**
** Modified 17.5.09 Added TWI_STEP_TIMER_MAX label, made = 100 as before, and
** make TWITimer active (there were no increments!!)
*****/
int8 twi_step(uint8 twcr_mask, uint8 status)
{
    uint8 twi_status;

    TWCR = twcr_mask | TWI_ENABLE;
    if(status != TWI_NO_WAIT)
    {
```

```

TWITimer = 0;
while(((TCCR & 0x80) == 0) && (++TWITimer < TWI_STEP_TIMER_MAX)) //17.5.09 Increment
// TWITimer..
;

twi_status = (TWSR & 0xFC);

if((status != TWI_IGNORE_STATUS) && ((twi_status != status) || ((TCCR & 0x80) == 0)))
{
    TCCR = TWI_SEND_STOP | TWI_ENABLE;
    TWITimer = 0;
    while(((TCCR & 0x80) == 0) && (++TWITimer < TWI_STEP_TIMER_MAX)) // 17.5.09
    // Increment TWITimer..
    ;
    return(TWI_ERROR);
}
}
return(TWI_SUCCESS);
}

```

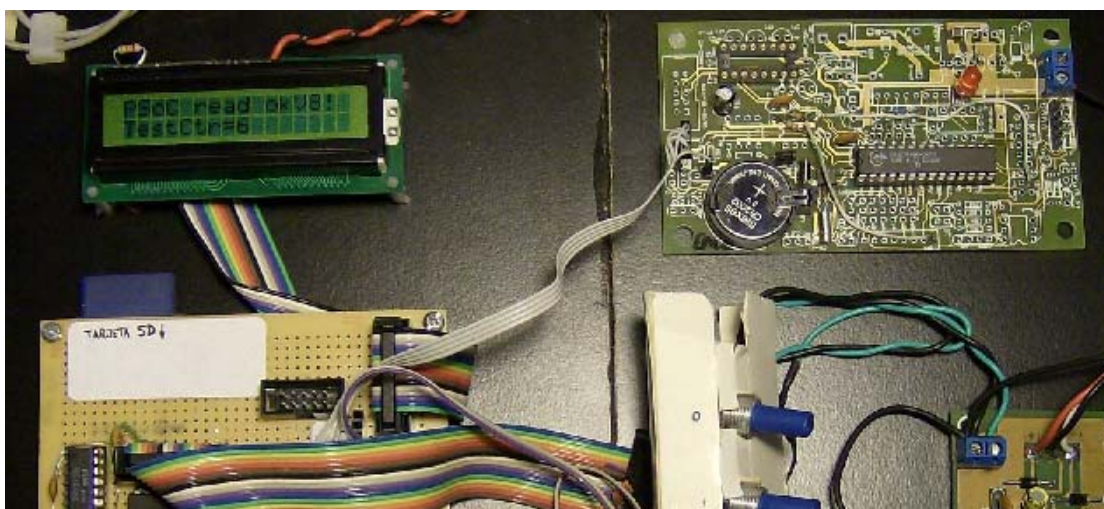


Figura 3.6a – Interfase I2C a Placa DLCy comunicándose con RTC y PSoC

La interfase I²C conectada al reloj permite mantener una hora medianamente precisa con el cristal de 32.768kHz (se observó deriva a lo largo de varios meses en productos reales, por lo cual la placa CL2b de producción incorpora la opción de un TXCO, oscilador compensado por temperatura).

CL2B Proto Board - (c) 2009 / LyR Ingenieria
I2C Peripherals
rev.8 08-09 / (c) R.Oliva

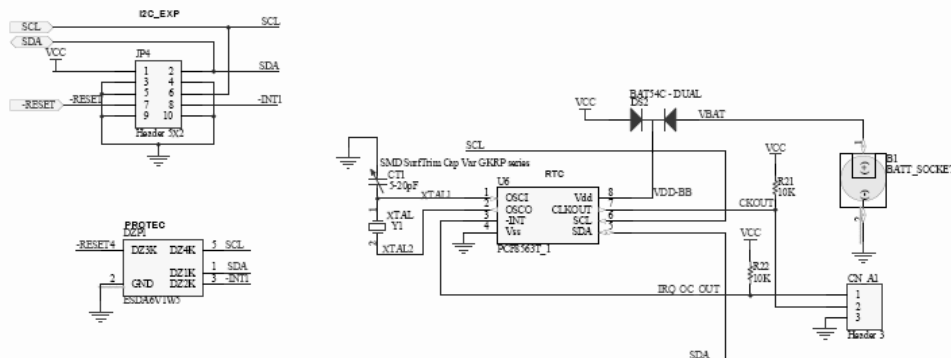


Figura 3.6b – Interfase I2C a RTC

Dicha interfase I²C se implementó también en el PSoC, que presenta un interesante potencial como “periférico configurable” a través del módulo Ez-HwI2C / esclavo. Por ejemplo, una de las primeras versiones de lo que se programó en este módulo PSoC / DLCy tenía el siguiente aspecto:

```
// 5.4.2009 Restart with Application "test" on DLCy, Similar to an
// I2C slave standing at address 0x05 (RAM) and 0x45 (ROM) of the I2C space.
// ** The ram values are arranged in a structure as follows:

struct I2C_Regs { // Example I2C interface structure
  BYTE bStat;
  BYTE bCmd;
  int iVolts;
  char cStr[6]; // Read only string
} MyI2C_Regs;

/** ROM Values
const BYTE DESC[] = "Hello I2C Master";

/** Main program running on PSoC..
void main()
{

  // Set up RAM buffer
  EzI2Cs_1_SetRamBuffer(sizeof(MyI2C_Regs), 4, (BYTE *) &MyI2C_Regs);
  EzI2Cs_1_SetRomBuffer(sizeof(DESC), DESC); // Set up ROM buffer
  M8C_EnableGInt; // Turn on interrupts
  EzI2Cs_1_Start(); // Turn on I2C
  // EzI2Cs_1_SetAddr(5); // Change address to 5 - En options del modulo,
  // lo pusimos como "static" en 5 (si es Dynamic, habilita esta funcion)

  // Initialization Added 5.4.09
  MyI2C_Regs.bStat = 1;
  MyI2C_Regs.bCmd = 2;
  MyI2C_Regs.iVolts = 2627;
  strncpy(MyI2C_Regs.cStr, "Test", 5);

  while(1) {
    // Place user code here to update and read structure data.
    if(MyI2C_Regs.bStat<255) MyI2C_Regs.bStat += 1;
    MyI2C_Regs.bCmd = 2;
    MyI2C_Regs.iVolts = 2627;
  }
}
```

Con esta herramienta, el procesador PSoC puede “publicar” en el espacio I²C una serie de registros que dependen del usuario y la aplicación. Se pueden así crear dispositivos I²C a medida para distintas aplicaciones.

3.1.f) Interfase a tarjetas SD/MMC

Esta interfase es la que permite comunicarse con las tarjetas de memoria flash de formatos en continua reducción física (SD, miniSD, microSD..) e incremento de capacidad. Esencialmente se comporta como un dispositivo SPI de mayor velocidad que el I2C pero requiriendo el uso de 4 líneas. Los drivers actuales (de PRIO.com) permiten la utilización segura de tarjetas Kingston SD/MMC de cualquier capacidad incluyendo la de 1GB (no se han ensayado mayores). Otras implementaciones de SD (ej.Sandisk) dan resultados de error en grabación por lo cual se recomienda mantenerse con dicha marca para el software utilizado. Según se vio en la figura 3.3, se requiere una alimentación y buffers de 3.3V para el uso de estas tarjetas. En el circuito de prueba se utilizó un buffer 4049 (inversor) en lugar del 4050 indicado en el esquemático, pero solo se requieren 3 de las 6. En los ensayos se utilizó una tarjeta Kingston de 64MB del 2004. Los zócalos utilizados son tipo Hirose y contienen switches indicadores de inserción y escritura habilitada, aunque estos se utilizarán sólo en la versión de producción.

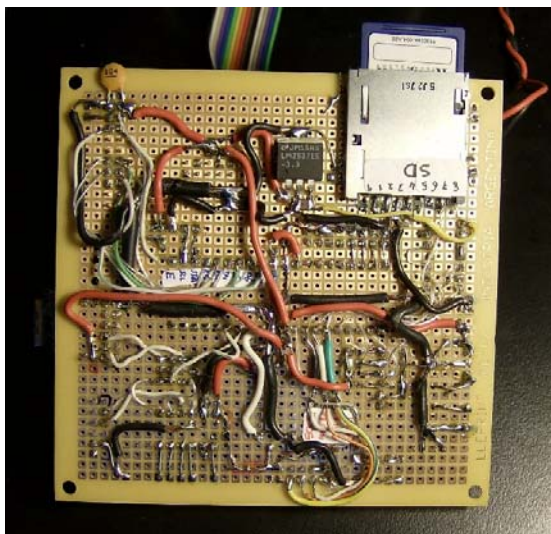


Figura 3.7 – Interfase a tarjetas SD/MMC

Para ejemplificar la interfase en C que presenta la combinación de este dispositivo con las librerías utilizadas, se incluye a continuación un extracto del código fuente a través del cual se inicializa la tarjeta SD:

```
// Variable FILE similar a las de Disco..
FILE *pntr1;

// -----
printf("3.Inicializacion..\n\r");
// Rutina que inicializa el puerto SPI y a la tarjeta SD/MMC...
while(initialize_media() == 0)
{
    set_LCD_cur(0,0);
    disp_cstr("Espera SD..  ");
    delay_ms(200);
    set_LCD_cur(0,0);
    disp_cstr("Espera SD.*  ");
    printf("* ");
    delay_ms(200);
    if(++SDCount>20) goto MAIN_LOOP; // Exit if no SD card. (Saving_to_SD_Flag is 0)
};

/* Create File */
printf("Creando File..\n\r");
set_LCD_cur(0,0);
disp_cstr("4.Creacion File.");
// Similarmente a un archivo de disco, se crea con fcreate()
pntr1 = fcreatec(_FF_FNAME, 0);
n = 0;
while(pntr1 == 0)
{
    if(pntr1 == 0)
    {
        _FF_printf(Fail_str, _FF_error);
        n++;
        if(n & 0xF0)
            return;
    }
    pntr1 = fcreatec(_FF_FNAME, 0);
}
}
```

Una vez creado el archivo, se escribe y lee en forma muy similar a como se hace en C con las computadoras personales, es decir con la rutina *fprintf()*. Por ejemplo, para escribir 10 columnas:


```
for(c = 0; c < 10; c++){
    fprintf(pntr1, column_d, c);
}
fprintf(pntr1, _FF_CRLFstr);
```

Una vez escrito el archivo, que declaramos con la extensión .CSV para leerlo en Excel, el mismo puede ser leído en una PC utilizando un lector de tarjetas convencional. Por ejemplo, un archivo de las pruebas recientes tiene el aspecto:

```
"TestCL2b.csv"
CL2 ArchivoPrueba - R.Oliva 2009
"08/09/2009, 02:25:17"
Fecha/Hora, CH0[V], CH1[V], CH2[V], Temp[C], fv1(PSoC)"
08/09/2009, 02:25:19, 2.5, 3.1, 2.9, 18.7, 0.234,
08/09/2009, 02:25:22, 2.5, 3.1, 2.9, 18.7, 0.234,
08/09/2009, 02:25:26, 2.5, 3.1, 2.9, 18.7, 0.234,
08/09/2009, 02:25:29, 2.5, 3.1, 2.9, 18.7, 0.234,
08/09/2009, 02:25:33, 2.5, 3.1, 2.9, 18.7, 0.234,
08/09/2009, 02:25:37, 2.5, 3.1, 2.9, 18.7, 0.234,
08/09/2009, 02:25:40, 2.5, 3.1, 2.9, 18.7, 0.234,
08/09/2009, 02:25:44, 2.5, 3.1, 2.9, 18.7, 0.234,
08/09/2009, 02:25:47, 2.5, 3.1, 2.9, 18.7, 0.234,
08/09/2009, 02:25:51, 2.5, 3.1, 2.9, 18.8, 0.234,
08/09/2009, 02:25:55, 2.5, 3.1, 2.9, 18.8, 0.234,
08/09/2009, 02:25:58, 2.5, 3.1, 2.9, 18.8, 0.234,
08/09/2009, 02:26:02, 2.5, 3.1, 2.9, 18.7, 0.234,
08/09/2009, 02:26:05, 2.5, 3.1, 2.9, 18.7, 0.234,
08/09/2009, 02:26:09, 2.5, 3.1, 2.9, 18.7, 0.234,
08/09/2009, 02:26:13, 2.5, 3.1, 2.9, 18.8, 0.234,
08/09/2009, 02:26:16, 2.5, 3.1, 2.9, 18.7, 0.234,
08/09/2009, 02:26:20, 2.5, 3.1, 2.9, 18.7, 0.234,
08/09/2009, 02:26:23, 2.5, 3.1, 2.9, 18.8, 0.234,
08/09/2009, 02:26:27, 2.5, 3.1, 2.9, 18.7, 0.234,
```

3.1.g) Interfase Serie RS232 y USB.

En el caso de las comunicaciones seriales con la PC, el CL2b permite implementarlas a través del un tanto relegado puerto RS232, o a través de un módulo USB que se conecta emulando un puerto COM. Si se utiliza un ATmega32 se cuenta con un solo puerto, mientras que con los nuevos tipos (compatibles) ATmega324P o ATmega644P (incluso a futuro ATmega1284P) se cuenta con dos puertos serie.

CL2B Proto Board - (c) 2009 / LyR Ingenieria
COMM CIRCUITS
rev.7 08-09 / (c) R.Oliva

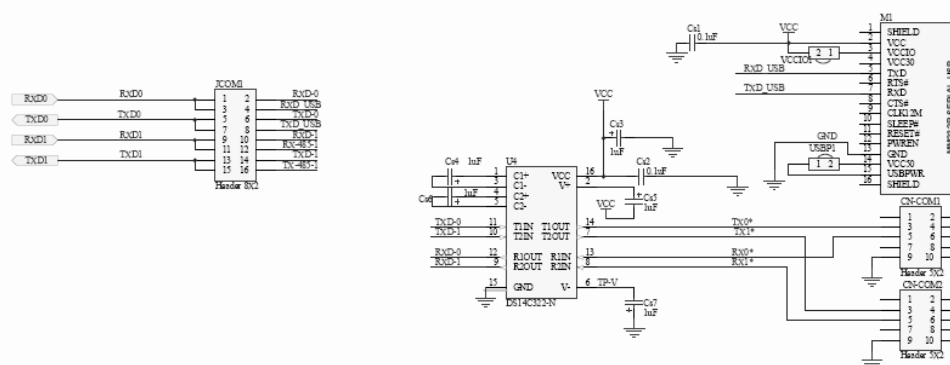


Figura 3.8 – Interfase Puertos serie RS232 y USB

En los CL2b-Proto se deja una salida adicional para RS485, una interfase convencional que se implementa en la placa de producción. Así, los switches de JCOM1 permiten que el COM0 se conecte con RS232 (aquí con un DS14C322, pinout compatible con el MAX232) o el modulo MM232-USB, y que para el COM1 se seleccione entre otro RS232 al mismo DS14C322 o al mencionado puerto RS485, que implementa "enable" automático por hardware o programable con un pin del controlador. A pesar del costo relativamente elevado (USD18 FOB) del módulo MM232 de FTDIChips, resulta conveniente que sea removible para los casos en que no se requiera esta interfase.

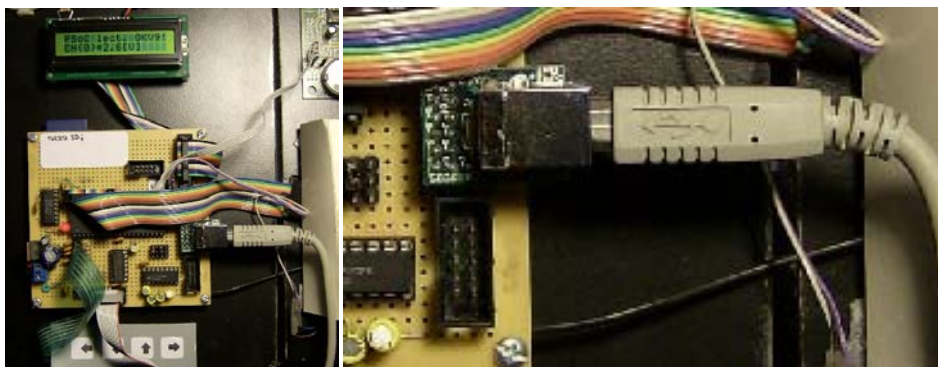


Figura 3.9 – Puertos serie RS232 y USB – Detalle del módulo FT232/USB (der)

Como nota relativa, la modificación del programa realizado con el compilador CodeVision se realiza a través del puerto ISP (en figura 3.9, izq, cable de 10conductores) a una interfase "TheCable-AVR", conectada al puerto COM1 de la PC de desarrollo.

En los circuitos de prueba CL2b se implementa el puerto (COM0) a través de USB, como en la Figura 3.9. Si se conecta a un puerto de la PC esta lo detecta (en general, bajo Windows XP) como un driver de tipo COM y le asigna un número (en la prueba COM9), aunque puede requerirse instalar el driver de FTDI (www.ftdi.com, gratuito). Abriendo un programa de comunicación como Hyperterminal se puede ver la secuencia de arranque a 9600, N,8,1 sin protocolo:

```
Probando v0.9 CL2-Prototipo 9-8-2009
LCD Access testing..
-TestCtr=1 -TestCtr=2 -TestCtr=3 -TestCtr=4 -TestCtr=5 -TestCtr=6
Ahora RTC test..
RTC inicializando..
"08/09/2009, 08:50:32"
Ahora PSoC-2 prueba de acceso..
PSoCReadStatus_in = 0
PSoC Union Dump of size 18..
08 02 43 0a 54 65 73 74 32
00 c5 20 70 3e 9a 01 65 c7

PSoCReadStatus_after = 0
bsStat = 8
bCmd = 2
iVolts = 2627
TestStr = Test2
FV1 = 0.234
FV2 = -58625.60156

Ahora test acceso a SD...

3.Inicializacion..
FlashFile v3.0
Boot_Sec: [EB 0 90] [55] [AA]
Part Address: 4E00
BPB_BytsPerSec: 200
BPB_SecPerClus: 20
```

```
BPB_RsvdSecCnt:      1
BPB_NumFATS:        2
BPB_RootEntCnt:      200
BPB_FATSize16:       C
BPB_TotSec16:       1E4D9
BPB_FATType:         FAT12
ClusterCnt:          F25
ROOT_ADDR:           8000
FAT2_ADDR:           6800
RootDirSectors:      F25
FirstDataSector:     39Creando File..
```

```
PSoC Union Dump of size 18..
08 02 43 0a 54 65 73 74 32
00 c5 20 70 3e 9a 01 65 c7
```

```
bsStat = 8
bCmd = 2
ivolts = 2627
TestStr = Test2
FV1 = 0.234
FV2 = -58625.60156
```

```
PSoC Union Dump of size 18..
09 02 43 0a 54 65 73 74 32
00 c5 20 70 3e 9a 00 65 c7
```

```
bsStat = 9
bCmd = 2
ivolts = 2627
TestStr = Test2
FV1 = 0.234
FV2 = -58624.60156
```

```
PSoC Union Dump of size 18..
09 02 43 0a 54 65 73 74 32
```

Como se observa, el bloque al que escribe inicialmente es el LCD, luego el RTC y el PSoC – conectado a través del puerto I2C, y finalmente en (3.) se ocupa de inicializar la tarjeta SD si se encuentra presente. Las cadenas siguientes son para depurar la comunicación I2C con el PSoC, y pueden eliminarse para dejar activa la interfase de comandos, utilizando el programa HOST CL2b (C++Builder), que también trabaja a través de un COM asignado (figura 3.10).

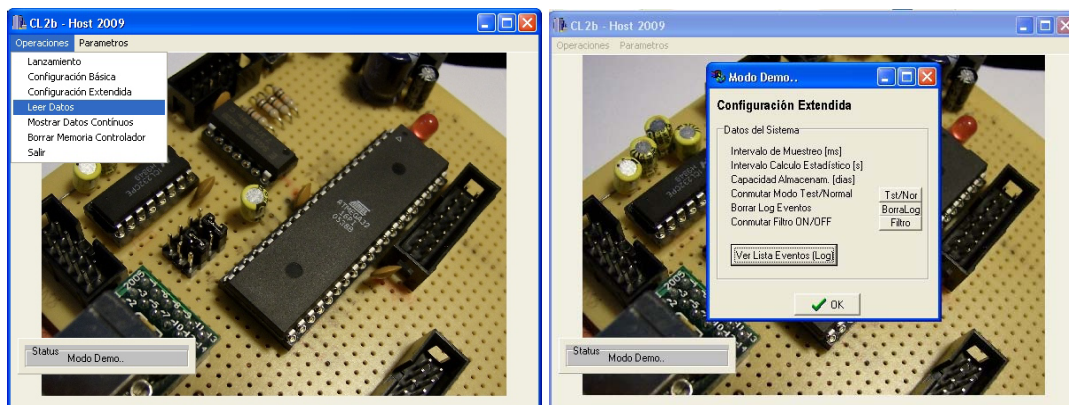


Figura 3.10 – Software HostCL2b en C++Builder

3.2 Comparación Placa Prototipo CL2b – a placa de producción.

Como se indicó en 2., la variante ensayada cumple con todos los requerimientos de funcionamiento de un sistema comercial, aunque para determinadas aplicaciones pueden

requerirse agregados o características especiales que conviene dejarlas previstas o implementadas.

Algunas de estas características pueden ser:

- 3.2.a) Consumo extra-reducido de energía y administración de dispositivos.
- 3.2.b) Precisión extra del RTC, sin implementar interfase GPS.
- 3.2.c) Mayor capacidad de E/S.
- 3.2.d) Conexión a redes RS485

A tal efecto, se presenta en lo que sigue el diagrama en bloques del sistema expandido denominado simplemente CL2b / Producción, en la Figura 3.11.

CL2_b (AtMega32/644) - Placa Produccion

(C) R.Oliva - L&R Ing. 2008/9

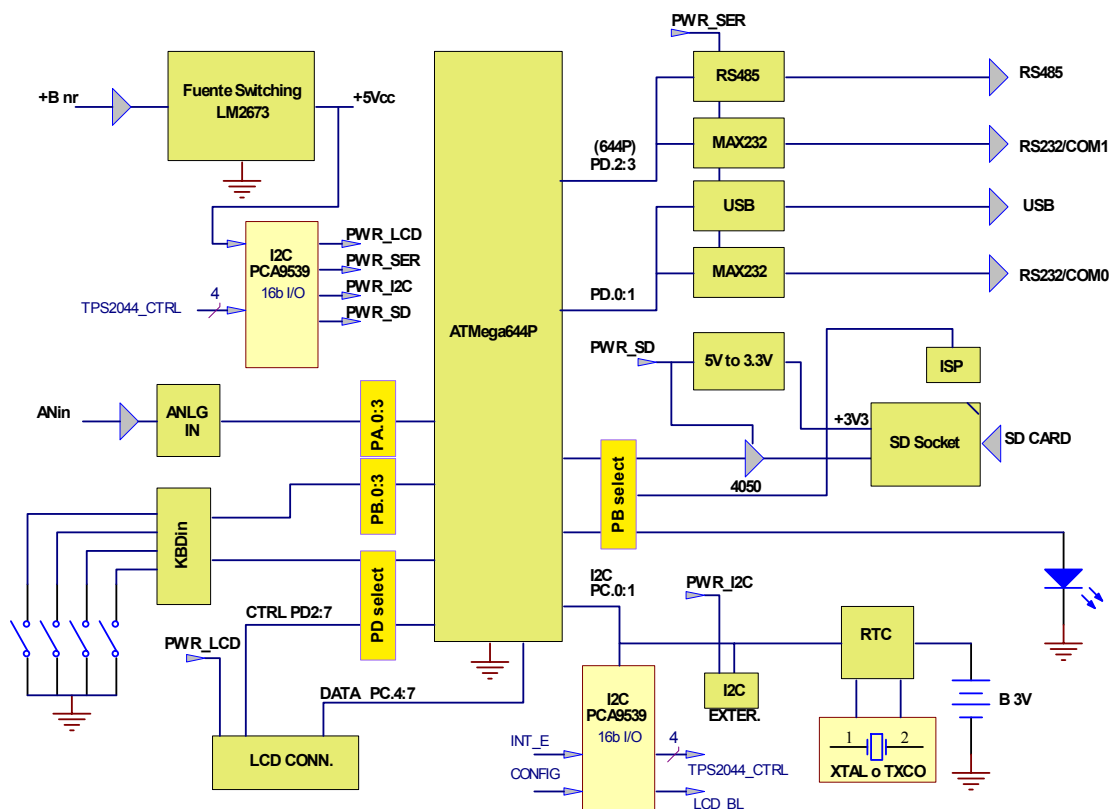


Figura 3.11 – Sistema CL2b expandido

Según se observa, es muy similar al sistema Prototipo pero tiene previstas dos unidades expandidas que lo hacen más poderoso. En primer lugar, la fuente que convierte la tensión de entrada puede recibir tensiones de hasta 40Vcc, y convertir con alta eficiencia a +5V. Esto reduce la disipación de potencia y ahorra energía. Asimismo, se incluye un administrador de potencia para dispositivos de 4 canales (TPS2044) que permite encender o detener por software dispositivos de consumo relativo alto como la interfase SD, el LCD, los puertos de comunicación o dispositivos externos I2C. El control de este dispositivo, así como la expansión de E/S indicada se logra a través de un integrado PCA9539, E/S de 16bits de última generación con interfase I2C.

Los circuitos que se modifican implican una penalidad en costo y complejidad, por lo cual esta opción dependerá de las características que se requiera. En la Figura 3.12 se puede verificar el agregado del integrado de E/S, como así también del TXCO (DS32kHz de Dallas) que mejora sensiblemente la precisión del RTC manteniendo el cristal interno de 32kHz a temperatura constante. El costo de este integrado (U10) es de USD8,50 FOB, por lo cual sólo se justifica en determinadas aplicaciones. En la Figura 3.13 se puede observar la nueva fuente switching y el integrado de administración de energía. Los Jumpers PWS1-4 permiten regresar a la configuración original si una o más características no resultan necesarias.

CL2B Board - (c) 2009 / LyR Ingenieria
I2C Peripherals
rev.7B 17-7-09 / (c) R.Oliva

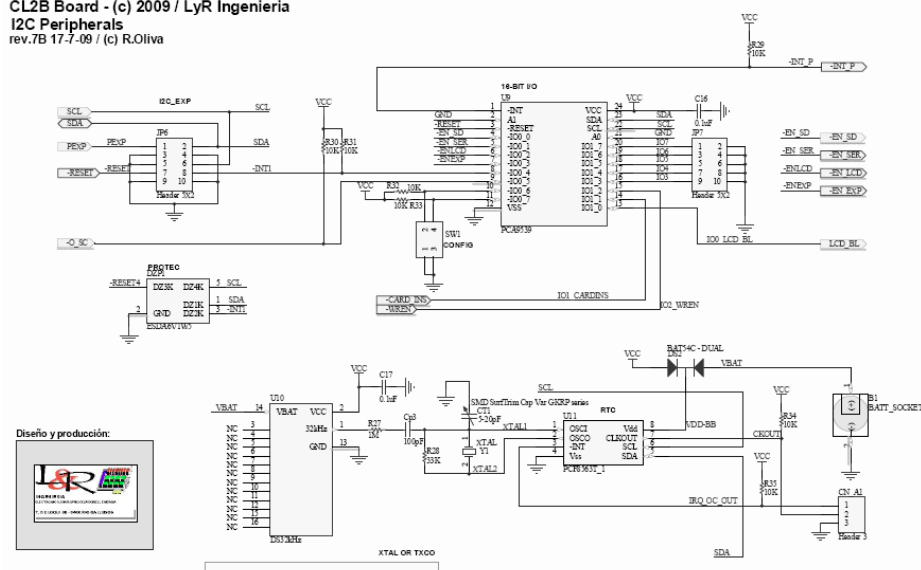


Figura 3.12 – Sistema CL2b – subsistema I2C con expansion

CL2-B Board - (c) 2009 / LyR Ingenieria

Power Supply
rev.7d 4-8-2009 / (c) R.Oliva

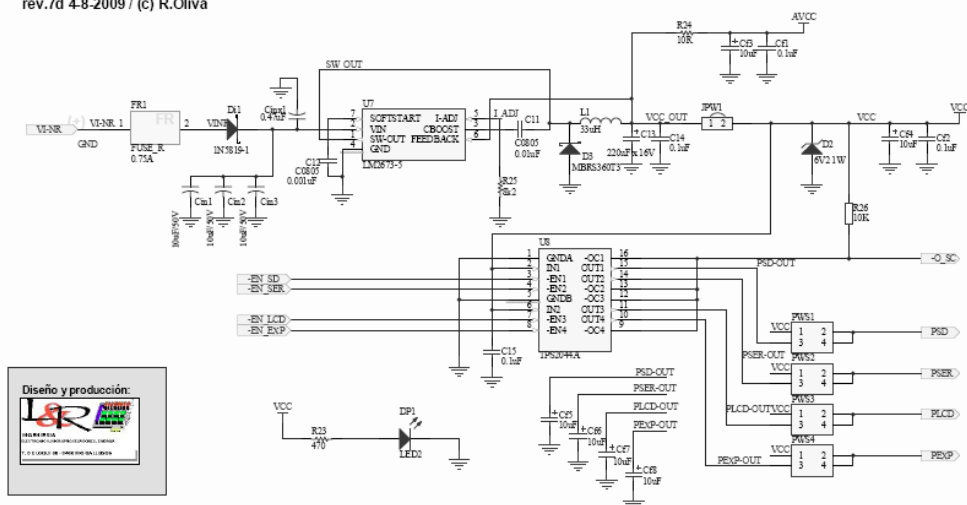


Figura 3.13 – Sistema CL2b – subsistema Fuente de alimentación

A efectos de incorporar la capacidad 3.2.d), se incluye en el módulo de comunicaciones un subsistema ampliamente difundido (R.E.Smith) de conexión a RS485, frecuentemente ensayado en otras implementaciones (Figura 3.14). Este subsistema puede ser configurado con J1 para Auto-Enable o Enable vía software, en nuestro caso utilizando el pin PD.4.

CL2B Board - (c) 2009 / LyR Ingenieria

COMM CIRCUITS

rev,6 b 18-07-09 / (c) R.Oliva

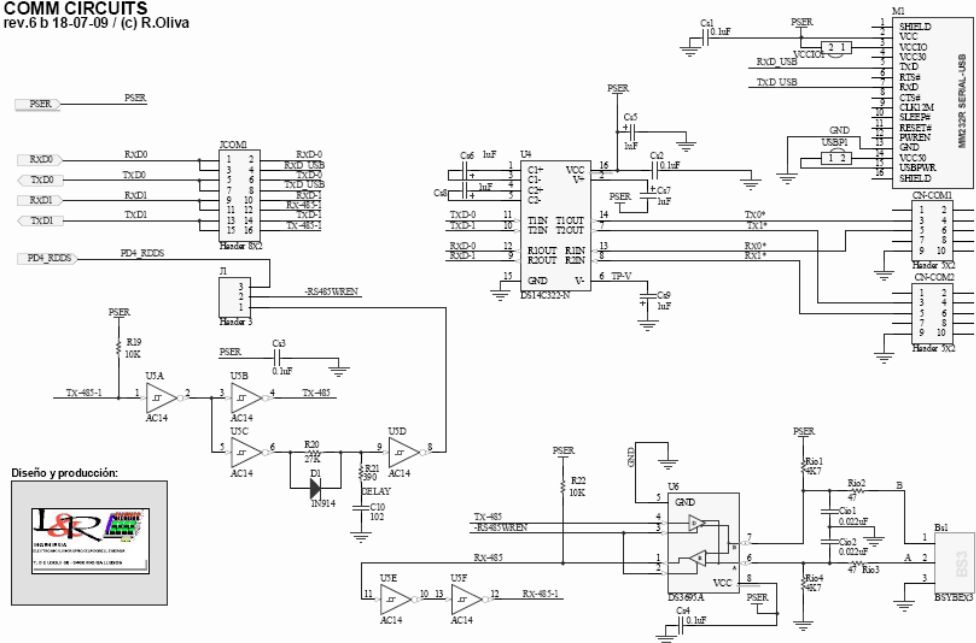


Figura 3.14 – Sistema CL2b – Serie + RS485

La implementación de este tipo de placa resulta altamente compleja para el armado a mano, por lo cual se encuentra en gestión la utilización de una firma de Buenos Aires con sistema de soldadura por reflow y autoposicionador de componentes. El circuito en su *placing* preliminar puede observarse en la Figura 3.15.

Las dimensiones de la placa se adaptan a las de los gabinetes del tipo SLB utilizados para la mayoría de los sistemas que implementan DLCy (Figura 1.1der), por lo cual se puede utilizar fácilmente incluso con riel DIN a través de soportes adecuados.

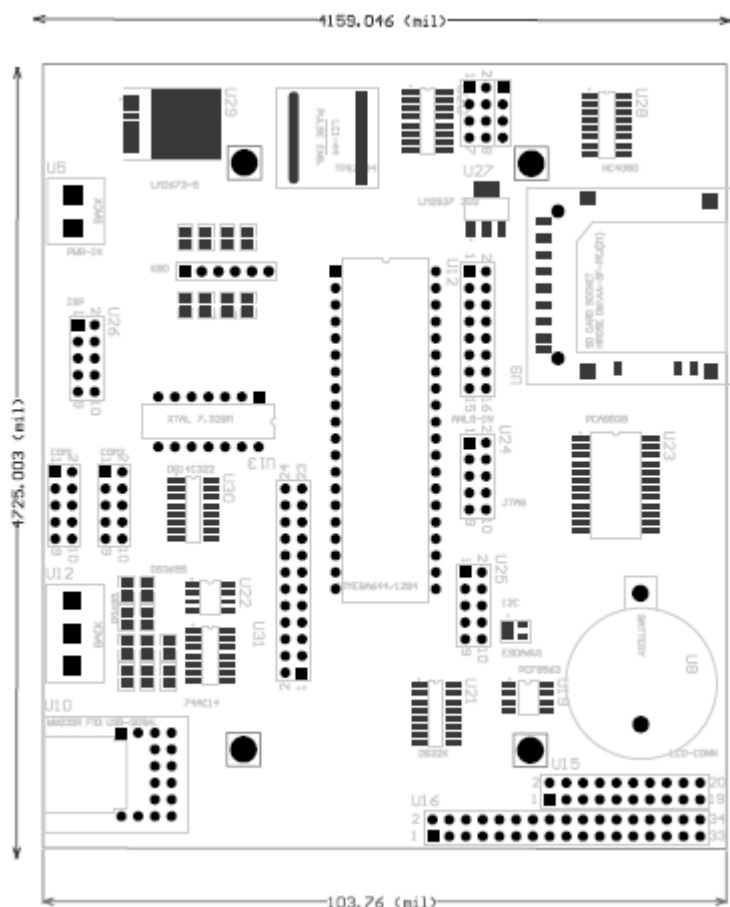


Figura 3.15 – Sistema CL2b – Placing preliminar

3.3 Ensayo de Sistema CL2b Proto para medición de temperatura y logeo en SD.

Para verificar el correcto funcionamiento del sistema se procedió a repetir un ensayo que se había realizado con ítems separados, esta vez en conjunto incluyendo un sensor de temperatura, potenciómetros para ajuste de canales y la posibilidad de almacenar en una tarjeta SD de 64MB (Figuras 3.16 / 3.17).

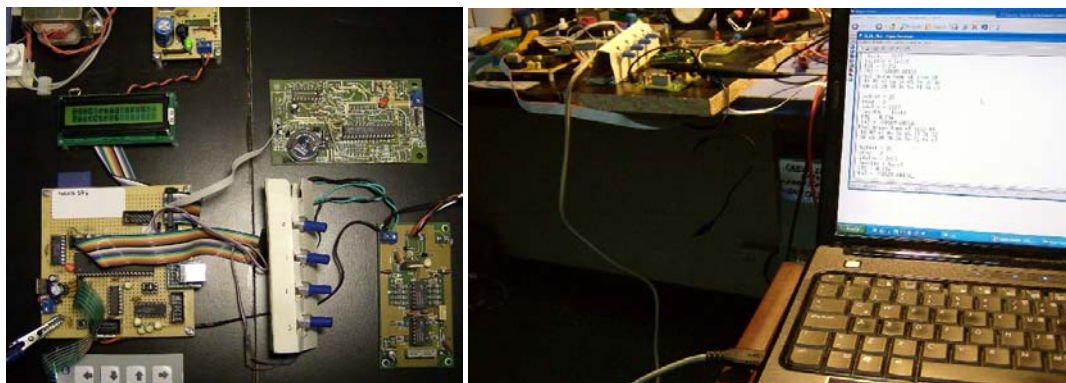


Figura 3.16 (repite 1.3) – Ensayos del Prototipo CL2b + DLCy y medición Temperatura, Enlace USB con PC

ENSAYO CON CL2B / TEMP / DLCy - 2009

(C) R.Oliva - L&R Ing. 2008/9

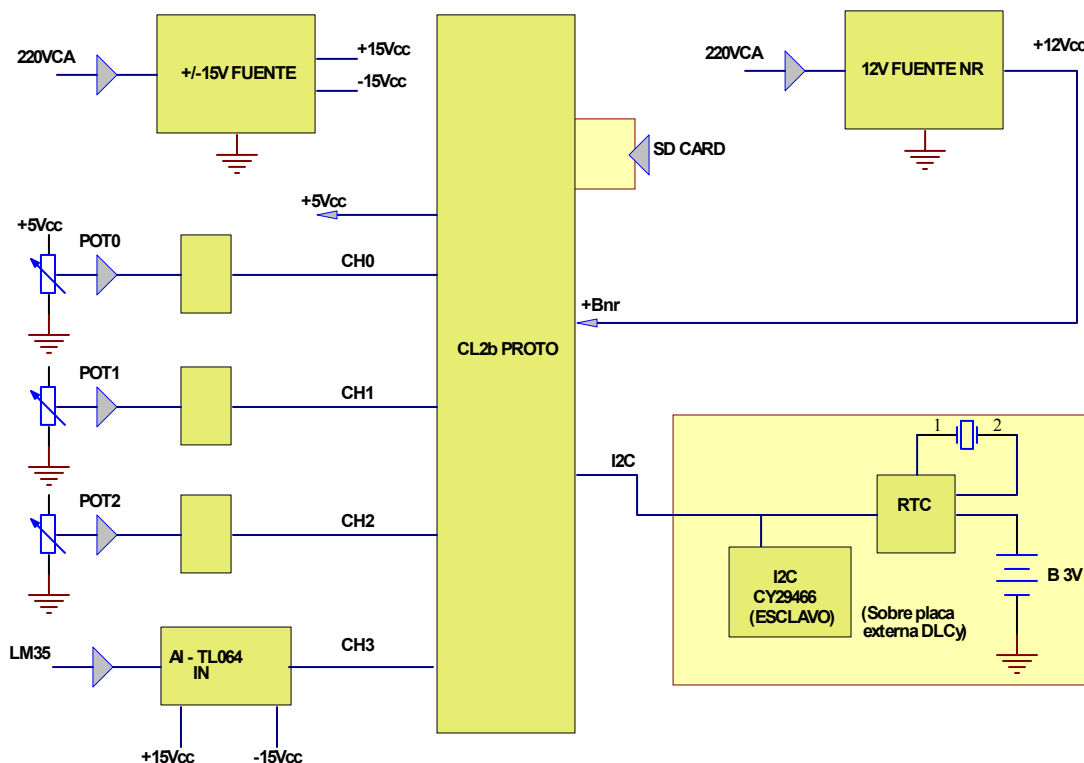


Figura 3.17 Diagrama en bloques del ensayo del Prototipo CL2b + DLCy y medición Temperatura

En el extracto que sigue se toman datos de temperatura y de los otros 3 canales (constantes) para el archivo TESTCL2B.CSV, en solo una porción de los 5600 registros:

TestCL2b.csv							
CL2 ArchivoPrueba - R.Oliva 2009							
08/09/2009, 02:45:25							
Fecha/Hora	CH0[V]	CH1[V]	CH2[V]	Temp[c]	fv1(PSoc)"		
08/09/2009 07:47:25	2.5	3.1	2.9	17.8	0.234		
08/09/2009 07:47:28	2.5	3.1	2.9	17.8	0.234		
08/09/2009 07:47:32	2.5	3.1	2.9	17.8	0.234		
08/09/2009 07:47:36	2.5	3.1	2.9	17.8	0.234		
08/09/2009 07:47:39	2.5	3.1	2.9	17.8	0.234		
08/09/2009 07:47:43	2.5	3.1	2.9	17.8	0.234		
08/09/2009 07:47:46	2.5	3.1	2.9	17.8	0.234		
08/09/2009 07:47:50	2.5	3.1	2.9	17.8	0.234		
08/09/2009 07:47:54	2.5	3.1	2.9	17.8	0.234		
08/09/2009 07:47:57	2.5	3.1	2.9	17.7	0.234		
08/09/2009 07:48:01	2.5	3.1	2.9	17.7	0.234		
08/09/2009 07:48:04	2.5	3.1	2.9	17.8	0.234		
08/09/2009 07:48:08	2.5	3.1	2.9	17.8	0.234		
08/09/2009 07:48:11	2.5	3.1	2.9	17.7	0.234		
08/09/2009 07:48:15	2.5	3.1	2.9	17.7	0.234		
08/09/2009 07:48:19	2.5	3.1	2.9	17.8	0.234		
08/09/2009 07:48:22	2.5	3.1	2.9	17.7	0.234		
08/09/2009 07:48:26	2.5	3.1	2.9	17.8	0.234		
08/09/2009 07:48:29	2.5	3.1	2.9	17.8	0.234		
08/09/2009 07:48:33	2.5	3.1	2.9	17.8	0.234		
08/09/2009 07:48:37	2.5	3.1	2.9	17.8	0.234		
08/09/2009 07:48:40	2.5	3.1	2.9	17.8	0.234		
08/09/2009 07:48:44	2.5	3.1	2.9	17.8	0.234		
08/09/2009 07:48:47	2.5	3.1	2.9	17.8	0.234		
08/09/2009 07:48:51	2.5	3.1	2.9	17.8	0.234		
08/09/2009 07:48:55	2.5	3.1	2.9	17.8	0.234		



INGENIERIA
ELECTRONICA, MICROPROCESADORES, ENERGIA

T. DE LOQUI 58 - 9400 RIO GALLEGOS

Informe N°3-Final ANR SC002/2003 - Página 21 de 30

08/09/2009	07:48:58	2.5	3.1	2.9	17.8	0.234
08/09/2009	07:49:02	2.5	3.1	2.9	17.8	0.234
08/09/2009	07:49:05	2.5	3.1	2.9	17.8	0.234
08/09/2009	07:49:09	2.5	3.1	2.9	17.8	0.234
08/09/2009	07:49:13	2.5	3.1	2.9	17.7	0.234
08/09/2009	07:49:16	2.5	3.1	2.9	17.8	0.234
08/09/2009	07:49:20	2.5	3.1	2.9	17.8	0.234
08/09/2009	07:49:23	2.5	3.1	2.9	17.7	0.234
08/09/2009	07:49:27	2.5	3.1	2.9	17.7	0.234
08/09/2009	07:49:31	2.5	3.1	2.9	17.8	0.234
08/09/2009	07:49:34	2.5	3.1	2.9	17.8	0.234
08/09/2009	07:49:38	2.5	3.1	2.9	17.8	0.234
08/09/2009	07:49:41	2.5	3.1	2.9	17.8	0.234
08/09/2009	07:49:45	2.5	3.1	2.9	17.8	0.234
08/09/2009	07:49:48	2.5	3.1	2.9	17.8	0.234
08/09/2009	07:49:52	2.5	3.1	2.9	17.7	0.234
08/09/2009	07:49:56	2.5	3.1	2.9	17.8	0.234
08/09/2009	07:49:59	2.5	3.1	2.9	17.7	0.234
08/09/2009	07:50:03	2.5	3.1	2.9	17.8	0.234
08/09/2009	07:50:06	2.5	3.1	2.9	17.7	0.234
08/09/2009	07:50:10	2.5	3.1	2.9	17.8	0.234
08/09/2009	07:50:14	2.5	3.1	2.9	17.8	0.234
08/09/2009	07:50:17	2.5	3.1	2.9	17.7	0.234
08/09/2009	07:50:21	2.5	3.1	2.9	17.7	0.234
08/09/2009	07:50:25	2.5	3.1	2.9	17.7	0.234
08/09/2009	07:50:28	2.5	3.1	2.9	17.8	0.234
08/09/2009	07:50:32	2.5	3.1	2.9	17.8	0.234
08/09/2009	07:50:35	2.5	3.1	2.9	17.7	0.234
08/09/2009	07:50:39	2.5	3.1	2.9	18.6	0.234
08/09/2009	07:50:43	2.5	3.1	2.9	19.6	0.234
08/09/2009	07:50:46	2.5	3.1	2.9	20.2	0.234
08/09/2009	07:50:50	2.5	3.1	2.9	20.5	0.234
08/09/2009	07:50:53	2.5	3.1	2.9	20.8	0.234
08/09/2009	07:50:57	2.5	3.1	2.9	20.9	0.234
08/09/2009	07:51:01	2.5	3.1	2.9	21	0.234
08/09/2009	07:51:04	2.5	3.1	2.9	21	0.234
08/09/2009	07:51:08	2.5	3.1	2.9	21.1	0.234
08/09/2009	07:51:11	2.5	3.1	2.9	21.1	0.234
08/09/2009	07:51:15	2.5	3.1	2.9	21.1	0.234
08/09/2009	07:51:19	2.5	3.1	2.9	21.2	0.234
08/09/2009	07:51:22	2.5	3.1	2.9	21	0.234
08/09/2009	07:51:26	2.5	3.1	2.9	20.8	0.234
08/09/2009	07:51:29	2.5	3.1	2.9	20.5	0.234
08/09/2009	07:51:33	2.5	3.1	2.9	20.3	0.234
08/09/2009	07:51:37	2.5	3.1	2.9	20.1	0.234
08/09/2009	07:51:40	2.5	3.1	2.9	20.9	0.234
08/09/2009	07:51:44	2.5	3.1	2.9	21.5	0.234
08/09/2009	07:51:47	2.5	3.1	2.9	22	0.234
08/09/2009	07:51:51	2.5	3.1	2.9	22.3	0.234
08/09/2009	07:51:55	2.5	3.1	2.9	22.6	0.234
08/09/2009	07:51:58	2.5	3.1	2.9	22.8	0.234
08/09/2009	07:52:02	2.5	3	2.9	22.9	0.234
08/09/2009	07:52:05	2.5	3	2.9	23.2	0.234
08/09/2009	07:52:09	2.5	3	2.9	24	0.234
08/09/2009	07:52:12	2.5	3	2.9	24.7	0.234
08/09/2009	07:52:16	2.5	3	2.9	25	0.234
08/09/2009	07:52:20	2.5	3	2.9	25.3	0.234
08/09/2009	07:52:23	2.5	3	2.9	25.6	0.234
08/09/2009	07:52:27	2.5	3	2.9	25.6	0.234
08/09/2009	07:52:30	2.5	3	2.9	25.1	0.234
08/09/2009	07:52:34	2.5	3	2.9	24.5	0.234
08/09/2009	07:52:38	2.5	3	2.9	24	0.234
08/09/2009	07:52:41	2.5	3	2.9	23.6	0.234
08/09/2009	07:52:45	2.5	3	2.9	23.2	0.234
08/09/2009	07:52:48	2.5	3	2.9	22.8	0.234
08/09/2009	07:52:52	2.5	3.1	2.9	22.4	0.234
08/09/2009	07:52:56	2.5	3.1	2.9	22	0.234
08/09/2009	07:52:59	2.5	3.1	2.9	21.7	0.234
08/09/2009	07:53:03	2.5	3.1	2.9	21.4	0.234
08/09/2009	07:53:06	2.5	3.1	2.9	21.2	0.234
08/09/2009	07:53:10	2.5	3.1	2.9	20.9	0.234
08/09/2009	07:53:14	2.5	3.1	2.9	20.8	0.234
08/09/2009	07:53:17	2.5	3.1	2.9	20.6	0.234
08/09/2009	07:53:21	2.5	3.1	2.9	20.4	0.234
08/09/2009	07:53:24	2.5	3.1	2.9	20.2	0.234
08/09/2009	07:53:28	2.5	3.1	2.9	20	0.234
08/09/2009	07:53:32	2.5	3.1	2.9	19.9	0.234
08/09/2009	07:53:35	2.5	3.1	2.9	19.8	0.234
08/09/2009	07:53:39	2.5	3.1	2.9	19.6	0.234
08/09/2009	07:53:42	2.5	3.1	2.9	19.5	0.234
08/09/2009	07:53:46	2.5	3.1	2.9	19.4	0.234
08/09/2009	07:53:50	2.5	3.1	2.9	19.3	0.234
08/09/2009	07:53:53	2.5	3.1	2.9	19.2	0.234
08/09/2009	07:53:57	2.5	3.1	2.9	19.1	0.234
08/09/2009	07:54:00	2.5	3.1	2.9	19	0.234
08/09/2009	07:54:04	2.5	3.1	2.9	19	0.234
08/09/2009	07:54:08	2.5	3.1	2.9	18.9	0.234

08/09/2009	07:54:11	2.5	3.1	2.9	18.8	0.234
08/09/2009	07:54:15	2.5	3.1	2.9	18.8	0.234
08/09/2009	07:54:18	2.5	3.1	2.9	18.7	0.234

En este extracto se puede observar el sector que en la Figura 3.18 (también un sector reducido de 600 registros, del registro total) se indica como (a) ampliado, y se debe al hecho de apoyar el dedo sobre el sensor que está habitualmente en contacto con la temperatura ambiente. El resto de las variables, incluida la lectura vía I2C al PSoC, prácticamente no fluctúan. El espaciamiento no es perfectamente constante pero oscila entre 3 y 4 segundos ya que no está controlado en forma precisa en este caso.

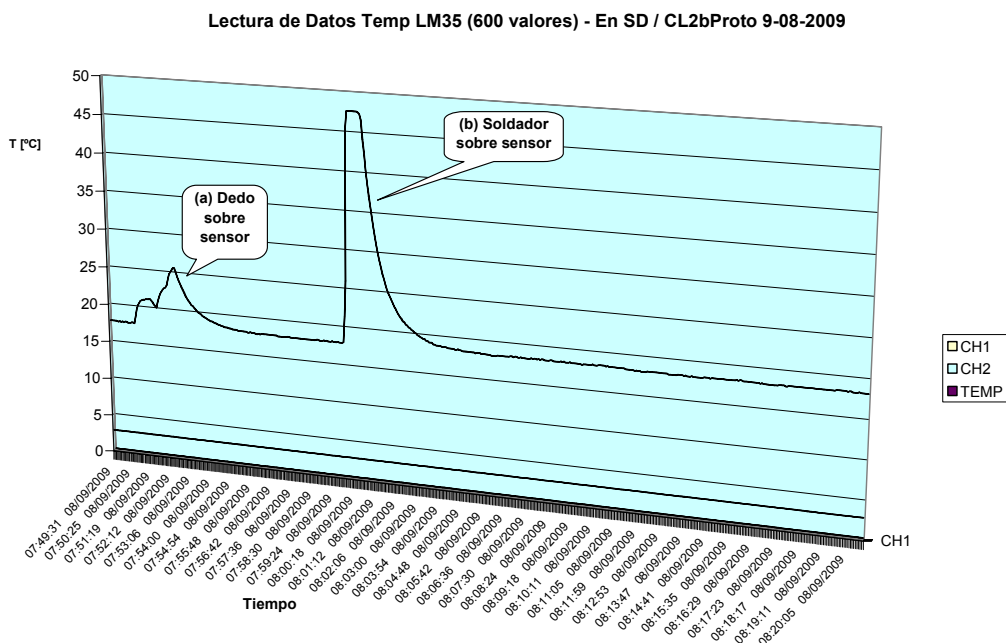


Figura 3.18 Medición – grafica de últimos 600 valores y aumentos por contacto (a) dedo y (b) soldador

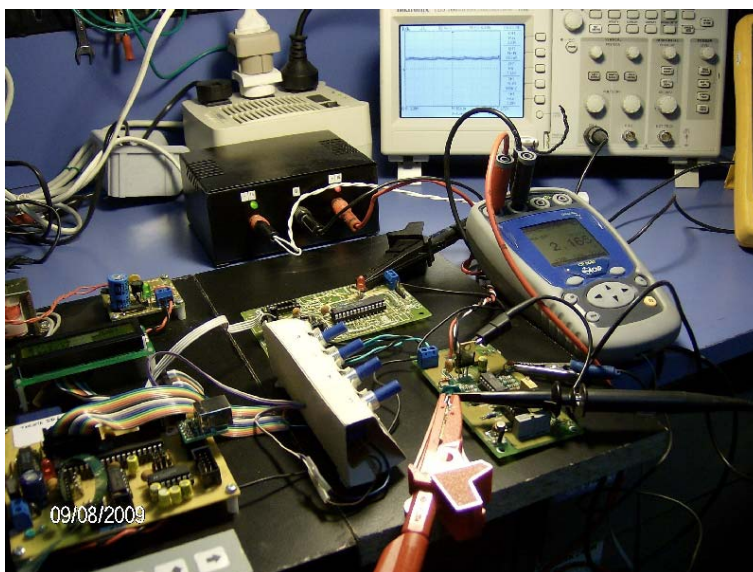


Figura 3.19 Medición – Uso de instrumental Calibrador AOIP y TEK

4. APLICACIONES IMPLEMENTADAS

Los sistemas desarrollados en base a los controladores PSoC (DLCy) y la experiencia ganada en los sistemas mayores AVR adquiridos como placas terminadas (basadas en ATmega128) han permitido una serie de avances importantes como para entrar en la producción propia de placas de mayor capacidad (CL2b) y variantes de la placa DLCy basada en PSoC. El dominio de la tecnología PSoC es una ventaja de escalabilidad a futuro, al anunciarse recientemente desde Cypress la implementación de sistemas de capacidad media PSoC3 (arquitectura 8051) y alta capacidad PSoC5 (arquitectura ARM Cortex de 32bits), manteniendo las capacidades mixtas analógica / digitales de la familia PSoC.

4.1 Monitoreo y control de pequeños aerogeneradores

Los sistemas DLCy han encontrado aplicación en sistemas diversos, en particular como controles para freno de aerogenerador combinados con sistema de monitoreo (Freno/1) según Informe 1, o como el que se implementa para DM Comunicaciones y Petrobras en Puesto Peter, Santa Cruz, (2007/2009) en el suministro y control de bombas dosificadoras.



Figura 4.1 Sistema de suministro de energía Eolico-Diesel, Freno (con DLCy), PLC y trans. Modbus RF Petrobras

DATOS DE OPERACIÓN - P38

Sistema Pbras- Pozo38 Agosto 2008
 (ix) Eventos + Muestras cada 10min

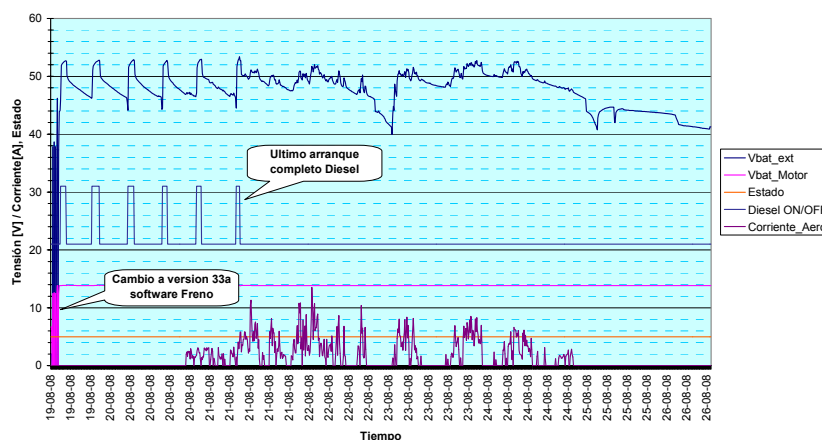


Figura 4.2 Datos relevados por DLCy en sistema PP38 - Petrobras

4.2 EM/12 -Aplicación al monitoreo de Energía en sistemas de baterías

Los equipos DLCy se han utilizado también en sistemas EM/12 para monitoreo de corriente y tensión continua, en aerogeneradores domiciliarios pequeños como los instalados a través del programa PERMER en Chubut (2007). Se trata un equipo de medición de energía que funciona con sistemas de 12V en corriente continua, con características que permiten un sencillo reemplazo y reubicación. Su diagrama general puede observarse en la Figura 1. Esta configuración consta de un **medidor básico EM2/12**, que cumple las características requeridas utilizando un DLCy, un display alfanumérico de 2 líneas x 16 caracteres y teclado de cuatro teclas.

En el tablero donde se montan el Regulador, el Rectificador y las protecciones que son provisión standard de los Aerogeneradores se reserva un espacio para ubicar una **Placa de Sensor**. Alternativamente (dado que el diseño del tablero original resultó muy comprimido), se instala la placa del sensor sobre un mismo panel junto con el EM12, realizándose una extensión de los cables que llevan la suma de la corriente de carga (tomas + iluminación) hasta dicha placa ubicada en el interior una caja separadora. Dicha placa contiene un sensor de Efecto Hall de alta corriente (consumo) y sensor de tensión. La Placa se monta sobre borneras tipo Fournas con capacidad 100A. Alternativamente, si el sistema tiene corriente de carga limitada por llaves térmicas, se utilizan borneras de 60A . En caso de removerse el sistema para prueba en otro aerogenerador, la placa se reemplaza por un puente de cobre con capacidad suficiente de carga.

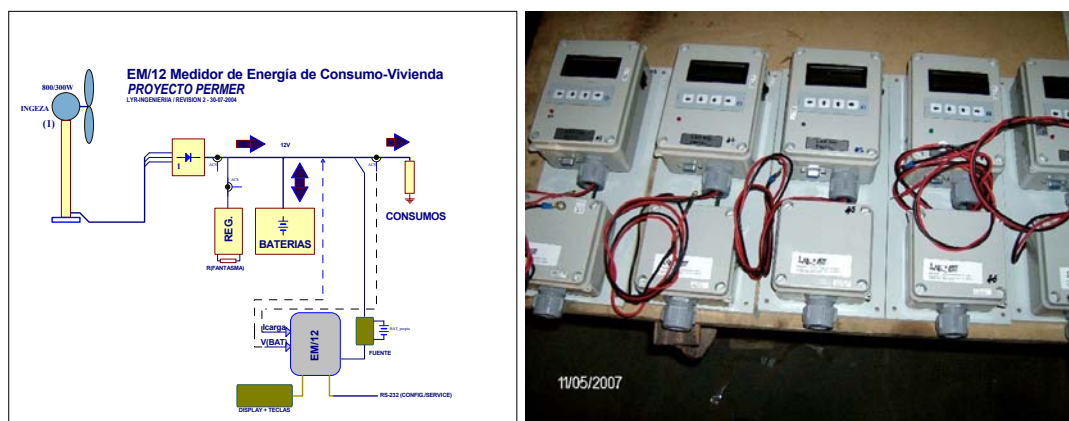


Figura 4.3 Sistema de Medición de Energía EM/12

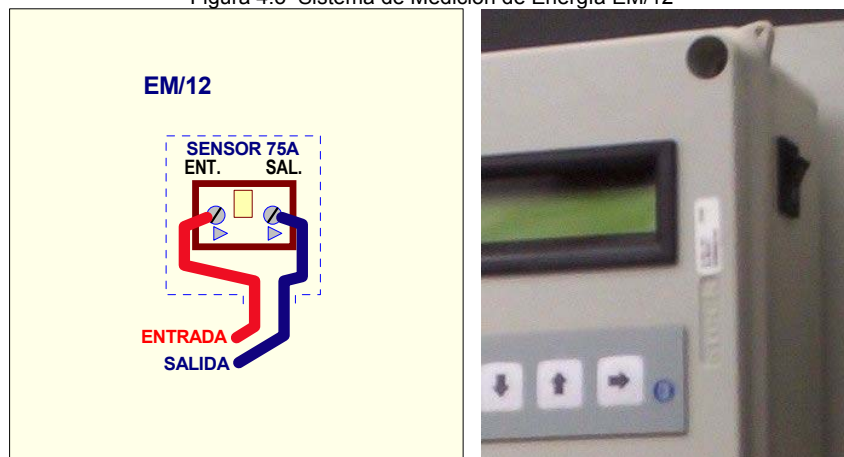


Figura 4.4 – Conexión a placa sensora de corriente EM/12 (izq.) y Llave de encendido (der.)

El sistema cuenta con una doble interfase con el usuario: a través del conjunto Teclado – Display LCD en el frente del Equipo (Figura 4.5), o a través de un puerto USB o COM, y una notebook PC corriendo el programa Config_EM12 provisto (Figura 4.6) La interfase LCD es informativa, la configuración exhaustiva debe realizarse a través de una notebook y el programa de configuración.

HHHHHH:mm:ss de operación (hasta 999.999 horas – 114 años)
Timestamp (Fecha/hora) de inicio operación
Energía total generada (kWh) – Valor mínimo en LCD: 0.1kWh
Pmax potencia máxima [W].
E(total), E(mes).
Histograma de potencias (opcional)
Eventos (Puede incluirse el evento muestra de Datos de tensión y corriente cada 10min/30min/no habilitado)
1. Apagado/Encendido 2. Ingreso a Setup 3. Re-arranque (RESET) 4. Evento Dato 5. Evento de prueba ('7')

Detalle de parámetros de configuración

Tipo de Evento de muestra de Datos (10min/30min/no habilitado)
String de Sitio hasta 20 caracteres (Fijo en v8)
String de ID/Medición hasta 20 caracteres (Fijo en v8)



Figura 4.5 – Interfase de Teclado y Display EM12



Figura 4.6 – Programa de Configuración para Equipos EM-12 (Data Entry Sistemas)

4.3 Sistemas OpenDLogger

Como evolución de los sistemas DLCy, el Proyecto Open DLogger (en evolución) involucra la publicación de un sistema abierto tanto en el diseño del hardware como del código fuente interno en lenguaje C, para la realización de un registrador de datos de aplicaciones múltiples, orientado fundamentalmente a proyectos de ciencia y técnica que requieran un equipo flexible, de precisión aceptable y presupuesto acotado. El sistema básico se basa en la placa DLCy – PSoC aunque se espera que, para evitar múltiples placas, se conciba un modelo DLCy-2 con canales analógicos de entrada, por ejemplo 4 canales de entrada 0-5V o 0-10V, conversión A/D en 13 bits, con los siguientes componentes:

- CPU PSoC 29466/32K memoria flash – con módulos analógicos y digitales configurables
- Reloj de tiempo real PCF8563 con batería CR3032
- Memoria 24LC256 /512
- LCD alfanumérico de 2x16 hasta 4x20
- Placa auxiliar para conexión de teclado de membrana
- Puertos serie RS232 o módulo USB
- Opción: fuente exterior con LM2575 y opción de conexión a panel fotovoltaico y batería exterior, funcionando como mini-UPS.

Otros elementos se encuentran en fase de definición: En cuanto al software interno, se basaría en una modificación del software para EM/12. Se publica source code explicado y reformateado del sistema base. Para el Software de comunicación se considera VB2008 para que sea de libre acceso, alternativo a la operación como Terminal. Debe poder generar un CSV. Se espera una planilla de Excel y en lo posible una con interfase VBA para integrar los XLS a partir del CSV.

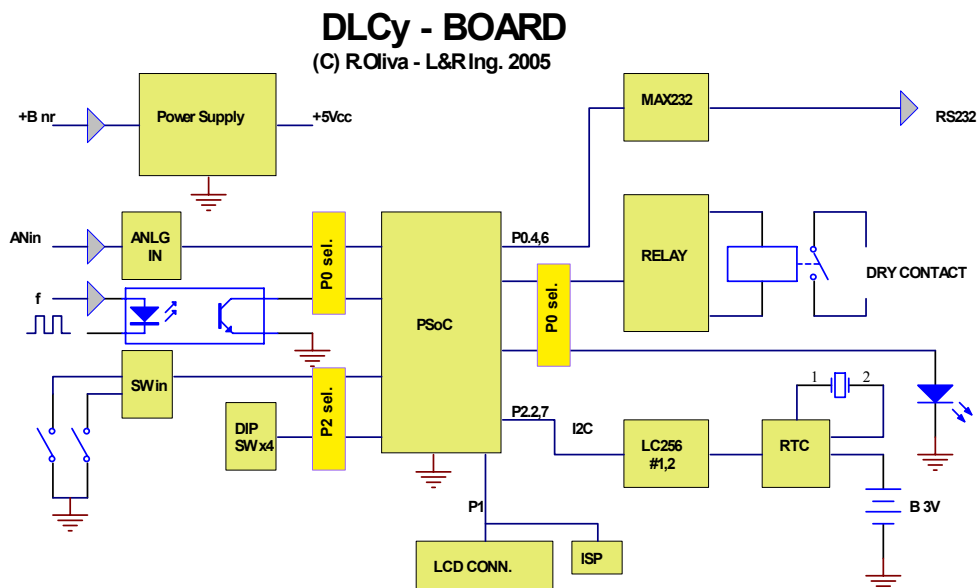


Figura 4.7 – OpenDlogger se basaría en los sistemas DLCy tendiendo hacia una versión DLCy-2

5. JUSTIFICACION DE LOS GASTOS REALIZADOS:

En la Tabla “3ra Certificación Contable sobre Rendición de Cuentas” (adjunta al presente informe), correspondiente al ANR N° SC-02/2003, se detallan gastos realizados en el marco de la tercera y última parte del presente proyecto, y se han mantenido las proporciones en los casos en que existen esperables diferencias entre los valores supuestos y los reales. Los principales ítems adquiridos son los siguientes:

5.1 Rubro 1 – Bienes de Capital

a) Maquinaria, equipos e instrumentos.

a.1 Aquí se incluye el ítem indicado como “I1) Osciloscopio DSO Tektronix TDS200 60MHz”. Para éste elemento, se eligió finalmente el modelo TDS1002B de 60MHz, equipo que suplantó al TDS200 con mejores características. El mismo se adquirió a la firma COASIN SA de Buenos Aires por un valor total de \$4703,29 y fue recibido en Julio de 2007. El sobrecosto respecto al presupuesto original de \$4000 ha sido de 17%.



Figura 5.1 – Osciloscopio TEKtronix adquirido TDS1002B / 60MHz y certif de calibración

a.2 Para el ítem “**S2) Anemómetro y Veleta NRG / USA**”, se adquirieron los modelos NRG #40C (Anemómetro Calibrado) y #200P (WindVane / 10K), al representante de NRG en Argentina (Energía Renovable Argentina, de Rosario). Debido a que se había discontinuado el modelo #40 presupuestado originalmente y al aumento de costos, el costo final de ambos sensores (\$1972,18) fue más del doble de lo estimado originalmente (\$895).



Figura 5.2 – Sensores NRG – Anemómetro y Veleta

a.3 Para el ítem “**I3) Calibrador 4-1/2 Digit 0.015%, (Calibrators-INC, DVC350A)**”, al momento de realizar la compra el mismo se encontraba discontinuado. Se solicitó cotización por equipos similares Fluke (EEUU) y AOIP (Francia) a COASIN y a SOLTEC, ambas firmas de Buenos Aires. Finalmente se adquirió el equipo AOIP CP6632, por reunir las características similares al DVC350A e incluir 5 modos de funcionamiento: de medición (tensión y corriente), de emisión (tensión y corriente) y de prueba de continuidad. El costo final del equipo adquirido a SOLTEC fue sin embargo más elevado de lo previsto, quedando en \$3865,39 o sea un 75% más de los \$2200 presupuestados originalmente.



Figura 5.3 – Calibrador AOIP CP6632 en funcionamiento

b) Otros. En la presente rendición se incluyen los siguientes elementos indicados como “**S3) Herramental especial para SMT/Soldadura/Conexion/Prueba**”. Estos elementos son complementarios a los adquiridos en la primera etapa (en ese momento por un total de \$1251,85), y se trata de una rotuladora electrónica con conexión a PC para el marcado de

elementos en gabinetes (Brother PT-1950P) adquirido a Amazon.com por un equivalente de \$99,57 + derechos de importación (sumados en el ítem 4.13). Además se adquirió un equipo de soldado SMT con aire caliente "Full Energy 908 – SMD Rework soldering station" y Kit de estaño Heraeus, a la firma Electro Tools de Buenos Aires por \$760.90.



Figura 5.4 – Estación de Soldado Full Energy

5.2 Rubro 2 – Recursos Humanos

En éste ítem se incluye el costo de las tareas llevadas adelante por el programador Prof. Rodolfo Vallejos, titular de Data Entry Sistemas, que consistió en software de procesamiento de los datos producidos por los loggers, basado en Delphi7 y construido sobre formato de bases de datos DbaseIV. El costo fue de \$850 para esta etapa, un 9% inferior a lo estimado en el presupuesto inicial del Proyecto.

5.3 Materiales e insumos. Aquí se han incorporado elementos componentes como microprocesadores, memorias, semiconductores diversos, reguladores, supresores de transitorios, y gabinetes .

5.4 Otros Costos. Aquí se han incluido gastos relacionados con la construcción de los circuitos impresos complementarios a las tareas del proyecto y sistemas auxiliares. La proporción es del 100% a cargo de contraparte, de acuerdo a lo pautado en la Reformulación.

En el caso de la adquisición del software Gimpel / C (PC-Lint) la disponibilidad de una versión gratuita de los code-checkers hacia 2007 hizo redundante este gasto de u\$s389. Se reemplazó por la compra del compilador CodeVision 2.043 (€150), versión 2009 que soporta los nuevos controladores de la línea AVR 324P,644P,1284P.

6. CONCLUSIONES

El Proyecto se encuentra finalizado en cuanto a los desarrollos planteados como objetivo, a pesar de los atrasos iniciales. Los cambios surgidos en cuestiones técnicas y de mercado han obligado a adaptaciones pero se muestra una evolución interesante de los equipos diseñados con varias aplicaciones en utilización. Esto concluye con los requerimientos de ejecución del proyecto.

Ing. Rafael Oliva
Río Gallegos, agosto de 2009