

**PROTOTIPO DE UN DISPOSITIVO DE RECONOCIMIENTO DE CARACTERES
APLICADO A LOS CONTADORES MONOFÁSICOS DE ENERGÍA ELÉCTRICA
TIPO CO-U449MT MARCA SKATEKS**

JUAN CAMILO GONZÁLEZ ROJAS
051018
Antares_colombia@gmail.com

ALEX ENRIQUE VELÁSQUEZ RIVAS
042078
Elementswar@gmail.com

**FUNDACIÓN UNIVERSITARIA SAN MARTÍN
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE ELECTRÓNICA Y TELECOMUNICACIONES
BOGOTÁ, COLOMBIA
Mayo de 2009**

**PROTOTIPO DE UN DISPOSITIVO DE RECONOCIMIENTO DE CARACTERES
APLICADO A LOS CONTADORES MONOFÁSICOS DE ENERGÍA ELÉCTRICA
TIPO CO-U449MT MARCA SKATEKS**

JUAN CAMILO GONZÁLEZ ROJAS
051018
Antares_colombia@gmail.com

ALEX ENRIQUE VELÁSQUEZ RIVAS
042078
Elementswar@gmail.com

MONOGRAFÍA DE GRADO

ASESOR TÉCNICO
ING. FREUD ABNER ROMERO TORRES
DSP

FUNDACIÓN UNIVERSITARIA SAN MARTÍN
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE ELECTRÓNICA Y TELECOMUNICACIONES
BOGOTÁ, COLOMBIA
Mayo de 2009

Nota de Aceptación:

**Freud Abner Romero
Asesor**

**Ramsés Martínez
Jurado**

**Rafael Cubillos
Jurado**

Bogotá, Mayo de 2009

AGRADECIMIENTOS

Queremos agradecer a aquellas personas que de una forma u otra estuvieron presentes en el desarrollo de este trabajo, brindando valiosas opiniones para guiarnos de la mejor manera posible en la elaboración de este prototipo.

A nuestro asesor Freud Abner Romero Torres, por sus constantes sugerencias y opiniones para el correcto desempeño del proyecto.

A nuestras familias, amigos y profesores.

CONTENIDO

	Pág.
1. RESUMEN	12
2. INTRODUCCIÓN	13
3. OBJETIVOS	14
3.1 OBJETIVO GENERAL	14
3.2 OBJETIVOS ESPECÍFICOS	14
4. MARCO REFERENCIAL	15
4.1 ANTECEDENTES	15
4.2 MARCO CONCEPTUAL	16
4.2.1 Contador Domiciliario	16
4.2.2 Sistema de Reconocimiento de Caracteres (OCR)	16
4.2.3 Estándar RS-232	17
4.3 MARCO TEÓRICO	17
4.3.1 Algoritmos de OCR	18
4.3.2 DSP (Digital Signal Processor)	22
4.3.3 Estándar RS-232	36
4.3.4 Módulo de Cámara VGA CMOS C328R	39
4.4 ESTADO DEL ARTE	42
5. METODOLOGÍA	43
5.1 Soporte e iluminación para el módulo CMOS C328R	43
5.2 Comunicación serial entre el módulo de cámara CMOS C328R y MATLAB	44
5.3 Desarrollo del algoritmo OCR en MATLAB	44
5.4 Comunicación serial entre el DSP y MATLAB	45
5.5 Sintetizar el algoritmo de OCR en el DSP	45
5.6 Comunicación entre el DSP y el módulo CMOS C328R	45
5.7 Visualización del resultado del OCR en una interfaz externa (LCD)	46

6. DESARROLLO	47
6.1 Soporte e iluminación del módulo CMOS C328R.	47
6.2 Comunicación serial entre el módulo de cámara CMOS C328R y MATLAB.	48
6.2.1 Conexión con el módulo CMOS C328R.	49
6.2.2 Rutinas de configuración del módulo CMOS.	54
6.3 Desarrollo del algoritmo OCR en MATLAB	57
6.3.1 Captura de imágenes con cámara web	58
6.3.2 Lectura de imágenes en MATLAB.	58
6.3.3 Recorte de la imagen.	60
6.3.4 Conversión de la imagen a blanco y negro (Binarización)	60
6.3.5 Ubicación de caracteres	61
6.3.6 Creación de Plantillas	61
6.3.7 Comparación con plantilla	61
6.3.8 Análisis especial del último dígito	62
6.3.9 Transmisión y recepción de datos mediante el puerto serial en MATLAB	63
6.3.10 Transmisión y recepción de datos seriales en el DSP	63
6.4 Sintetizar el algoritmo de OCR en el DSP	65
6.4.1 Almacenamiento de la imagen al DSP, binarización y recortado	65
6.4.2 Comparación de la plantilla con la imagen del contador	66
6.5 Comunicación entre el DSP y el módulo CMOS C328R	68
6.5.1 Configuración del módulo UART en el DSP	68
6.5.2 Configuración del módulo CMOS desde el DSP	68
6.6 Visualización del resultado del OCR en una interfaz externa (LCD)	69
6.6.1 Rutina de inicialización y configuración de la LCD	71
6.6.2 Rutinas de escritura en la LCD	74
7. PRUEBAS Y RESULTADOS	76
7.1 Captura de imágenes del módulo CMOS	76
7.2 Conversión de imágenes en escala de grises a blanco y negro	77
7.3 Calibración de los valores de referencia	78
7.4 Resultados del algoritmo de OCR en el DSP	80
8. CONCLUSIONES	82
9. RECOMENDACIONES	84
10. TRABAJO FUTURO	85
11. GLOSARIO	86

12.	BIBLIOGRAFIA	88
12.1	Referencias Bibliográficas	88
12.2	Referencias de Internet	88
13.	ANEXOS	95
13.1	Anexo 1. Especificaciones del módulo CMOS C328R	95
13.2	Anexo 2. Manual del Usuario	96
	MANUAL DEL USUARIO	96
13.3	Anexo 3. Circuito Impreso para el DSP y el módulo CMOS C328R	99
13.4	Anexo 4: Tabla con las pruebas de luminancia –porcentaje de similitud basada en 50 lecturas realizadas.	100
13.5	Anexo 5: Porcentaje de error cuantitativo de las lecturas.	108

LISTA DE FIGURAS

	Pág.
Figura 1. Segmentación y normalización.	19
Figura 2. Ejemplo del Método de Matriz de Concordancia (OCR) .	21
Figura 3. Ejemplo del Método de Extracción de Características (OCR) .	22
Figura 4. Una señal digital.	23
Figura 5. Señal Analógica en un osciloscopio.	24
Figura 6. Proceso de conversión analógica a digital .	24
Figura 7. Etapas básicas del PDS.	25
Figura 8. Tipos de arquitectura de memoria entre un Microcontrolador (arquitectura Von Neumann) y un DSP (arquitectura Harvard).	29
Figura 9. dsPIC30F4011.	35
Figura 10. Disposición de pines del conector RS-232.	37
Figura 11. Conectores DB-25 y DB-9.	37
Figura 12. Circuito integrado MAX232.	39
Figura 13. Conexión de pines para un cable Null-Modem.	39
Figura 14. Módulo de cámara VGA CMOS C328R.	40
Figura 15. Imagen capturada con el módulo C328R con resolución de 320x240 pixeles.	41
Figura 16. Metodología en espiral.	43
Figura 17. Etapas de la metodología en espiral, aplicada al proyecto.	44
Figura 18. Vistas: lateral, superior y frontal del soporte para el módulo CMOS.	47
Figura 19. Vista lateral e interior del soporte para el módulo CMOS.	48
Figura 20. Módulo de cámara VGA de Toshiba TCM8230MD.	49
Figura 21. Módulo de cámara CMOS C328R, placa y conectores.	50
Figura 22. Diseño de fuente de voltaje con salida de 5v y 3.3v.	50
Figura 23. Ejemplo de conversión TTL a RS-232.	51
Figura 24. Circuito de operación típico del MAX232.	52
Figura 25. Circuito conversor de niveles de voltaje CMOS a TTL.	52
Figura 26. Conversión de niveles de voltaje TTL a CMOS.	53
Figura 27. Entorno de programación de MATLAB.	53
Figura 28. Conexión entre el puerto serial y el MAX232.	54
Figura 29. Ejemplo de comando del módulo CMOS.	54
Figura 30. Ejemplo de sincronización entre el host y el módulo CMOS.	55
Figura 31. Ejemplo del proceso de captura de una imagen.	56
Figura 32. Una Imagen obtenida con el módulo CMOS en MATLAB.	58
Figura 33. Diagrama de flujo del proceso de OCR en MATLAB.	59
Figura 34. Imágenes obtenidas con cámara web.	59
Figura 35. Imagen antes y después del proceso de recortado.	60
Figura 36. Imagen convertida a blanco y negro.	60
Figura 37. Visualización de los caracteres separados según análisis de su ubicación.	61

Figura 38. Plantilla creada mediante unión de otras imágenes.	62
Figura 39. Proceso de comparación de imagen del contador y plantilla	63
Figura 40. Cambio de número en el último dígito del contador.	62
Figura 41. Caso especial del último dígito.	64
Figura 42. Vista de las ventanas utilizadas en MPLAB IDE v8.1.	64
Figura 43. Creación de la plantilla almacenada en el DSP.	66
Figura 44. Diagrama de flujo del proceso de OCR en el DSP.	67
Figura 45. Conversión de la imagen a blanco y negro y recortado de la misma para compresión de datos (8:1).	68
Figura 46. Diagrama de flujo de comunicación entre el DSP y el módulo CMOS C328R.	69
Figura 47. Carácter de una LCD.	70
Figura 48. Diagrama de bloques de una LCD TC1602A.	71
Figura 49. Comando "Clear Display".	72
Figura 50. Comando "Entry Mode Set".	72
Figura 51. Comando "Display ON/OFF Control".	72
Figura 52. Rutina de configuración de la LCD.	73
Figura 53. Comando "Entry Mode Set". Nota: X = no importa.	74
Figura 54. Comando "Write Data to CG or DD RAM".	74
Figura 55. Diagrama de flujo de la rutina de 'escritura'.	75
Figura 56. Proceso de organización de datos obtenidos por el módulo CMOS.	76
Figura 57. Capturas en resolución 120x160 variando posición, condiciones de iluminación y giro del lente de la cámara para mejor enfoque.	77
Figura 58. Imagen original e imágenes en blanco y negro con variación del promedio de luminancia.	78
Figura 59. Imagen original e imágenes en blanco y negro en el DSP.	78
Figura 60. Graficas que representa las pruebas de calibración realizadas	79
Figura 61. Resultado del proceso de OCR.	80
Figura 62. Porcentajes de error en el reconocimiento de caracteres.	81

LISTA DE TABLAS

	Pág.
Tabla 1. Especificaciones del dsPIC30F4011.	35
Tabla 2. Valores de voltaje asignados a los niveles lógicos.	37
Tabla 3. Funciones de los comandos de configuración del módulo C328R.	55

LISTA DE ANEXOS

	Pág.
Anexo 1. Especificaciones del módulo CMOS C328R.	95
Anexo 2. Manual del Usuario.	96
Anexo 3. Circuito Impreso para el DSP y el módulo CMOS C328R .	99
Anexo 4: Tablas con las pruebas de luminancia y porcentaje de similitud basada en 50 lecturas realizadas.	100
Anexo 5: Porcentaje de error cuantitativo de las lecturas.	108

1. RESUMEN

En este trabajo se diseña un sistema prototipo para el reconocimiento de caracteres en una imagen capturada a través de una cámara, el cual está enfocado a los contadores monofásicos de energía eléctrica. Este permitiría reducir el costo y el tiempo empleado en realizar labores de lectura para estos contadores.

El prototipo se encarga de capturar la imagen mediante una cámara fotográfica, reconocer los caracteres que se encuentran en la imagen, y mostrar el resultado a través de una interfaz que para este trabajo es una pantalla LCD.

El reconocimiento de caracteres se realiza mediante un algoritmo implementado en un DSP (Digital Signal Processor), el cual permite extraer para el caso de los contadores monofásicos de energía eléctrica los que indican el consumo del servicio público utilizado.

Asimismo, el prototipo posee una conexión adicional para exportar los resultados obtenidos hacia otros dispositivos que procesan datos, como lo puede ser un computador.

2. INTRODUCCIÓN

Actualmente existen diferentes procesos para realizar la lectura y supervisión de algunos elementos en los cuales no pueden aparecer errores de medición, pero muchos de estos procesos implican un alto costo de fabricación y funcionamiento.

En el caso de los servicios públicos, los errores de medición se han vuelto muy frecuentes y la forma de solucionarlos requiere una segunda lectura, la cual representa bastante tiempo.

Muchas empresas han optado por cambiar su línea de contadores, pero hay otras que no poseen el capital necesario para llevar a cabo esta labor, y han optado por colocar sistemas que intervienen físicamente con el funcionamiento del contador, lo que provoca en algunas ocasiones daños irreversibles para el mismo.

Es por esto que se ha diseñado una solución al problema donde el prototipo establecido no interviene el funcionamiento de los contadores, y además de ser de muy bajo costo, permite obtener datos reales y fiables en cualquier momento con el uso de fotografías.

De igual manera la utilización de este sistema prototipo puede ser expandido a diferentes productos en donde se vea la necesidad de realizar mediciones frecuentes.

Cabe mencionar que la característica más importante de este sistema prototipo es la utilización de un DSP (Digital Signal Processor), ya que además de su bajo costo y su reducido tamaño, se encarga de llevar a cabo todo el procesamiento requerido para la imagen.

3. OBJETIVOS

3.1 OBJETIVO GENERAL

Diseñar e implementar un sistema prototipo que permita hacer la lectura de contadores monofásicos de energía eléctrica Tipo CO-U449MT Marca SKATEKS usando un algoritmo de reconocimiento de caracteres sintetizado en un DSP.

3.2 OBJETIVOS ESPECÍFICOS

- Elegir una cámara externa que pueda capturar imágenes de baja resolución a corta distancia para realizar la lectura de contadores domiciliarios.
- Seleccionar un Dispositivo DSP (Digital Signal Processor) que permita el almacenamiento y procesamiento de imágenes de baja resolución.
- Sintetizar un algoritmo OCR (Optical Character Recognition) en el dispositivo DSP que permita identificar los caracteres de la imagen.
- Interconectar el dispositivo DSP y la cámara externa usando una comunicación serial bajo el estándar RS-232.
- Diseñar una interfaz que permita interpretar los datos resultantes arrojados por el DSP

4. MARCO REFERENCIAL

4.1 ANTECEDENTES

Con la aparición de los servicios públicos, las empresas prestadoras de éstos han tratado de tarifar el costo del mismo. Antiguamente no se hacía un conteo del consumo domiciliario como tal, simplemente se cobraba un cargo fijo por la simple contratación del servicio. En la medida en que las poblaciones fueron creciendo, surgió la necesidad de establecer con precisión el consumo por usuario, de esta forma apareció el contador. Actualmente la lectura de los contadores que muestran en forma numérica el consumo de energía de un usuario, se hace con el método de lectura manual.

Con las grandes poblaciones crecientes se requiere de un sistema más confiable, rápido, y económico. Los más relevantes se han centrado en la forma en cómo se podrían enviar los datos desde un sistema remoto de lectura hasta una central, tal y como es el caso de un proyecto de grado de la escuela profesional de ingeniería electrónica en Lima - Perú, en el cual se desarrolló en el año 2000, un sistema de monitoreo remoto de procesos a través del canal de voz de teléfonos celulares GSM, donde se propone usar dicho sistema como uno de monitoreo remoto de contadores. En este proyecto no se plantea la forma de hacer la lectura del conteo del medidor. En Colombia se ha trabajado en el área, en la pontificia universidad javeriana, en conjunto con su facultad de ingeniería electrónica se planteó el desarrollo de un “sistema telemedición de servicios de públicos en Santafé de Bogotá” en donde se propone básicamente el monitoreo remoto de medidores a través de cualquier medio inalámbrico posible, como satélites y/o la red telefónica celular, nuevamente se parte de que el contador entrega las lecturas de forma electrónica, contrastando con la realidad, ya que los contadores instalados actualmente en Bogotá no tienen estas características [UPCE2007] , [PUJE2000].

Asimismo se ha dado una solución a la manera en que se realiza la adquisición de los datos mostrados por el contador, donde la manera más eficiente de realizarlo consiste en la implantación de un sistema de reconocimiento de caracteres (OCR), como es el caso del proyecto de grado realizado en la Universidad Autónoma de Barcelona enfocado hacia el desarrollo de sistemas de adquisición de datos por medio de un programa de OCR aplicado sobre una FPGA. El sistema es bastante confiable pero presenta el problema de ausencia de memoria para almacenar los datos procesados, de igual manera al desarrollarse este sistema sobre una FPGA es dispendioso realizar algún tipo de modificación, problema que no se presenta cuando se trabaja con algún tipo de microprocesador como lo son los DSP.

4.2 MARCO CONCEPTUAL

En el mundo se han dado muchas soluciones al tema de la medición de los contadores de servicios públicos, dado que las empresas buscan reducir costos de operación y problemas de medición. Muchas soluciones involucran complejos sistemas de medición satelital, o cambios tecnológicos en estos contadores.

Cuando se habla de un “Prototipo de un Dispositivo de Reconocimiento de Caracteres Aplicado a los Contadores de Servicios Domiciliarios” es necesario comentar las partes y componentes que se manejan en este prototipo para entender de una mejor manera el funcionamiento y ventaja del mismo.

4.2.1 Contador Domiciliario

Es un dispositivo que mide el consumo de energía eléctrica suministrada a una residencia o negocio. Actualmente los contadores de servicios públicos se dividen en mecánicos y electrónicos. Los mecánicos son los más usados en las principales ciudades colombianas, donde su funcionamiento se basa en la rotación de un disco de aluminio. Esta rotación permite que un sistema de numeración mecánico avance visualizando un número determinado.

Los contadores electrónicos consisten en un dispositivo que calcula la cantidad de corriente que pasa a través de él para luego representarla en una pantalla que puede encontrarse a varios metros del lugar donde se encuentra el dispositivo. Este tipo de contador es mucho más costoso e implica una mayor inversión por parte de las empresas en términos de seguridad [ANSW2008].

4.2.2 Sistema de Reconocimiento de Caracteres (OCR)

El OCR (*Optical Character Recognition*) es un algoritmo de reconocimiento de texto que saca de una imagen el texto que contiene y lo transforma en cadenas de caracteres para guardarlos en un formato que se pueda utilizar en programas de edición de texto.

Cuando se tiene una imagen (ya sea una fotografía o un documento) que se haya escaneado, el texto que contiene forma parte de esa imagen, al igual que cualquier elemento de la misma (un dibujo o un esquema). Si se necesita extraer dicho texto para poder editarlo, es necesario un programa de OCR que reconozca y lo transforme en una cadena de caracteres para posteriormente copiar esta cadena a un programa de edición y poder trabajar con ella, con el consiguiente ahorro de tiempo al no tener que teclear este texto.

4.2.3 Estándar RS-232

Es un estándar diseñado para la conexión serial de señales de datos binarios entre un DTE (Equipo terminal de datos) y un DCE (Equipo de terminación del circuito de datos).

El estándar RS-232 tiene un conector tipo DB-9 (9 pines), aunque también es posible encontrar conectores tipo DB-25 (de 25 pines) en equipos antiguos. Estos conectores DB-9 se encuentran en la gran mayoría de computadores siendo empleados para la transmisión y recepción de datos.

Algunas limitaciones de este estándar, es que los cables no pueden tener longitudes mayores a 15 metros, y no se recomienda realizar transmisiones a velocidades mayores de 20 KB/s [PIPE2009].

4.3 MARCO TEÓRICO

Actualmente se ha incrementado la tendencia hacia el mejoramiento de la adquisición de datos del sistema de contadores de servicios públicos domiciliarios, provocando el auge y desarrollo de nuevas tecnologías y dispositivos que ayudan a solucionar en gran parte los problemas que se presentan con el sistema actual de mediciones. En los últimos años se ha optado por rediseñar los contadores domiciliarios para incorporarles opciones de lectura remota, seguridad en mediciones, lecturas múltiples, reducción de costos y tiempo en el proceso de registro de datos.

Estas soluciones han sido de gran ayuda para la lectura remota de contadores, pero exigen una modificación del contador que de igual manera provoca una intervención del servicio público prestado. Por tal razón se ha decidido implementar un sistema prototipo que permita realizar las mismas tareas de los sistemas de medición modernos, pero sin la desventaja de tener que intervenir en el sistema de suministro del servicio público.

El sistema de lectura radica en la captura y procesamiento de la imagen que muestra el contador del servicio público. Esta imagen se obtiene a través de una cámara digital, y su procesamiento es realizado por un algoritmo de OCR. En la actualidad existen distintos tipos de software para realizar OCR desarrollados en ambientes de programación como lo son JAVA o C++. Una vez ejecutado el software de OCR, la información de los caracteres reconocidos es transmitida hacia un dispositivo externo que sería el encargado de visualizar un posible resultado.

Actualmente se han desarrollado dispositivos que permiten un alto poder de procesamiento a muy bajo costo, como lo son los DSP (Digital Signal Processor), mediante los cuales se pueden realizar una serie de tareas programables en un ambiente específico, para que en este caso sigan instrucciones de comunicación, procesamiento y transmisión de datos, e implantación de un algoritmo de OCR.

Cada uno de los procesos antes mencionados se encarga de una función en especial (captura, reconocimiento, transmisión) que es imprescindible para lograr el funcionamiento del dispositivo como tal y de esta forma darle solución al problema propuesto. Por esta razón se hace necesaria una descripción detallada de los procesos ejecutados en la lectura y en el funcionamiento de cada una de las partes involucradas en el sistema.

4.3.1 Algoritmos de OCR

OCR (Optical Character Recognition) es un software de reconocimiento óptico de caracteres, el cual de una imagen extrae los caracteres que ésta contiene para almacenarlos y procesarlos con programas de edición de texto.

En todo sistema de reconocimiento óptico de caracteres (OCR) se distinguen al menos estas 4 etapas:

- Adecuación de la imagen (pre-proceso).
- Selección de la zona de interés (segmentación).
- Representación digital de la imagen (extracción de características).
- Distinción del carácter contenido en la imagen (reconocimiento).

Y para cada una de las cuatro etapas es posible aplicar multitud de técnicas ya existentes o desarrollar alguna específica en función de las condiciones en las que se presentan los datos de entrada, que en el caso de OCR se puede traducir por las imágenes de entrada. A continuación se describen algunos de los métodos que se pueden aplicar para la realización de OCR. [UPVE2005]

4.3.1.1 Pre-proceso

Las técnicas de OCR son útiles para digitalizar textos de algún libro (caracteres impresos) o texto escrito manualmente (caracteres manuscritos), donde la extracción de los caracteres es mucho más sencilla para el caso de los caracteres impresos, ya que para el segundo caso es necesaria la aplicación de técnicas de pre-proceso y segmentación más complejas.

En esta fase de pre-proceso (o adecuación de la imagen) el objetivo que se busca es eliminar de la imagen cualquier tipo de ruido o imperfección que no pertenezca al carácter, así como normalizar el tamaño del mismo.

Para la eliminación del ruido que puede aparecer en una imagen digital provocado por manchas reales, escritura imperfecta o defectos técnicos en la adquisición o binarizado de la imagen, se utilizan diversos algoritmos:

- Etiquetado: para la división de la imagen en regiones de componentes conectadas.
- Eliminación / expansión: para la eliminación de pequeños grupos de píxeles.
- Umbralizado de histograma: para eliminar/seleccionar los objetos más brillantes o más oscuros de la imagen.

4.3.1.2 Segmentación

El proceso de segmentación consiste en separar los caracteres para que sean procesados individualmente y no como una sola palabra.

Cuando se trabaja con caracteres impresos, es mucho más fácil realizar este proceso de segmentación debido a que la impresión de los caracteres se desarrolla en condiciones en las que se predetermina el espacio entre los mismos, mientras que en el caso de caracteres manuscritos, se debe llevar a cabo este proceso para identificar con claridad cada carácter. [UPVE2005]

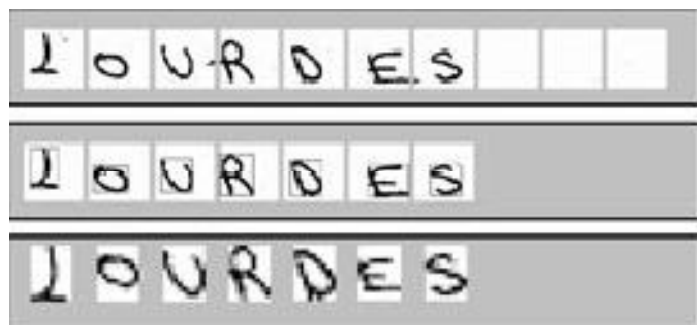


Figura 1. Segmentación y normalización. [UPVE2005]

En la Figura 1 se muestra el proceso de segmentación y normalización mostrando la imagen original dividida por las marcas de un campo de un formulario, y de igual manera se visualiza la eliminación de ruido, la detección de la caja mínima de inclusión, y la normalización del tamaño de los caracteres.

Sin embargo, para el caso de texto continuo se requiere la explotación de alguna característica del mismo, como puede ser la longitud de los caracteres (en el caso más sencillo), los valles de separación entre letras o números distintos, las proyecciones del texto sobre líneas imaginarias y el posterior análisis de los perfiles obtenidos, entre otros. La utilización de estos métodos está sujeta a las características concretas del texto que se desea segmentar, por lo tanto el uso de los mismos o de otros distintos se decide tras un análisis de los datos con los que se debe trabajar.

4.3.1.3 Extracción de Características

Una vez realizada la segmentación, se dispone de una imagen normalizada en la que se encuentra la información dispuesta a ser “reconocida”. La información representada por una matriz bidimensional de valores binarios, niveles de gris o color RGB, no codifica de la mejor forma las características más discriminativas del objeto al que representa.

Desde el punto de vista del reconocimiento de formas, la matriz bidimensional se ve como un vector de tantas dimensiones como componentes tiene la matriz. La dimensión de estos vectores (el número de componentes) es normalmente elevado, lo que supone un gran coste computacional a la hora de procesar el mismo. Más importante aún, es que está comprobado que al intentar clasificar (“reconocer”) vectores de este tamaño aparece un efecto, que provoca que los resultados, independientemente del método de clasificación utilizado, sean malos. Por ello se han desarrollado multitud de técnicas, denominadas “técnicas de selección y extracción de características”, mediante las cuales es posible obtener una representación del objeto a reconocer más eficiente.

Eficiencia, en este caso, significa que con una representación más compacta se consigue un poder discriminativo igual o superior al que se tenía con la representación original. Esto no es solo importante por el ahorro de espacio en el almacenamiento de las muestras, sino que durante el proceso de reconocimiento reduce los costos computacionales, debido a la reducción en el volumen de información procesada.

En el campo de investigación del reconocimiento de formas se utilizan algunos métodos de extracción de características basados en transformaciones del espacio de representación de las muestras. Ejemplos de estos métodos son:

- PCA (Principal Component Analysis).
- LDA (Linear Discriminant Analysis).
- ICA (Independent Component Analysis).
- NDA (Non-linear Discriminant Analysis).

Uno de los más usados es el método PCA. El objetivo de esta técnica es definir una transformación lineal desde el espacio de representación original a un nuevo espacio en el que las distintas clases de las muestras quedan mejor separadas. Además, esta transformación permite reducir la dimensión del nuevo espacio sin perjudicar sensiblemente la capacidad discriminativa de la nueva representación.

4.3.1.4 Reconocimiento

En esta última etapa es donde el reconocimiento (o clasificación) de los caracteres se realiza. El problema planteado en esta etapa consiste en desarrollar algún método que sea capaz distinguir la clase a la que pertenece un caracter entre un conjunto limitado de posibles opciones. [UPVE2005].

Existen dos métodos básicos utilizados para el reconocimiento de OCR: Matriz de concordancia y extracción de características. La matriz de concordancia es la más simple y la más común, ya que comprara las imágenes de caracteres escaneados con una librería de matrices o plantillas de caracteres. Cuando una imagen coincide con una de estas matrices en un cierto nivel de similitud, el software arroja el correspondiente ASCII del caracter escaneado, tal como se puede observar en la Figura 2.

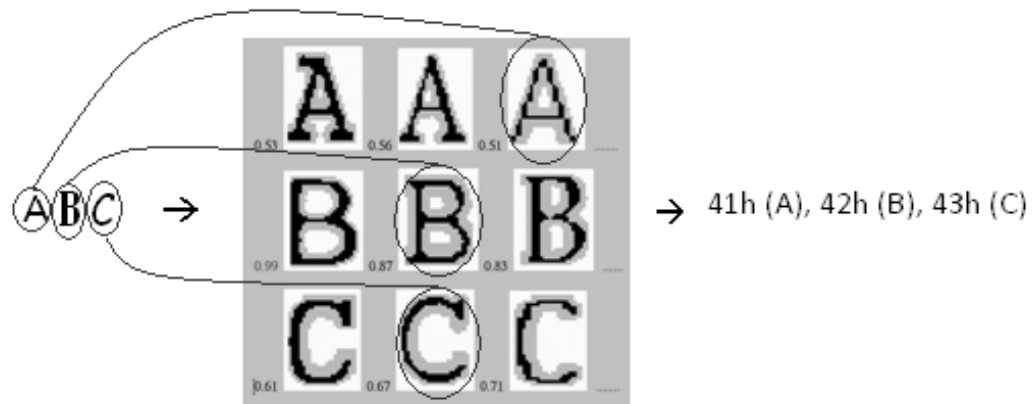


Figura 2. Ejemplo del Método de Matriz de Concordancia (OCR) [UPVE2005]

El método de *Extracción de Características* consiste en un software un poco más robusto (dependiendo del fabricante), que busca las características generales de los caracteres (Figura 3), como lo son las curvas, líneas cerradas o abiertas, líneas diagonales, entre otras.

Este método es muy utilizado para encontrar caracteres ante diferentes estilos de escritura, aspecto muy limitado en el método de matriz de concordancia, pero al

mismo tiempo la matriz de concordancia presenta una gran ventaja ante su contraparte cuando los caracteres son escritos en un solo estilo o tipo de letra, ya que se agiliza mucho el procesamiento de la imagen al analizarse solo un rango de caracteres [DATA2008]. Es por esto que la adquisición de los caracteres que aparecerán en la imagen tomada del contador de energía eléctrica, serán procesadas mediante un software que implemente el método de matriz de concordancia.

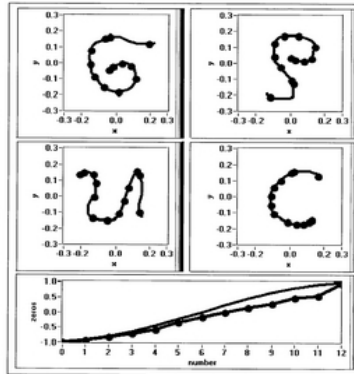


Figura 3. Ejemplo del Método de Extracción de Características (OCR) [DATA2008].

El software empleado para la ejecución del OCR es implementado en el lenguaje de programación C++ mediante la utilización de comandos, y con el apoyo de librerías de uso libre que permitan el análisis de los caracteres que se encuentran en una imagen. La razón de la utilización de este lenguaje de programación, se debe a la facilidad de programación de los DSP con este lenguaje.

4.3.2 DSP (Digital Signal Processor)

Para hablar de lo que es un DSP, es necesario entender algunos conceptos básicos relacionados con el mismo.

4.3.2.1 Señal

Es un símbolo, un gesto u otro tipo de signo que informa o avisa de algo. La señal sustituye por lo tanto a la palabra escrita o al lenguaje. Ellas obedecen a convenciones, por lo que son fácilmente interpretadas.

Cuando se trata de símbolos, las señales están colocadas en lugares visibles y están realizadas normalmente en diversos colores y formas. En el caso de los

gestos, son hechas por las personas mediante las manos y los brazos. También hay indicaciones consistentes en banderas, utilizadas sobre todo en la navegación marítima, y señales luminosas, como las de los faros en las costas.

Así mismo, una señal puede ser también la variación de una corriente eléctrica u otra magnitud física que se utiliza para transmitir información. Por ejemplo, en telefonía existen diferentes señales, que consisten en un tono continuo o intermitente, en una frecuencia característica, que permite conocer al usuario en qué situación se encuentra la llamada. [NAVA2008]

4.3.2.2 Señales analógicas y digitales

Una señal digital es simplemente una sucesión de impulsos eléctricos que pueden interpretarse únicamente como valores altos (1) o valores bajos (0) tal como se observa en la figura 4. [CALI2008]

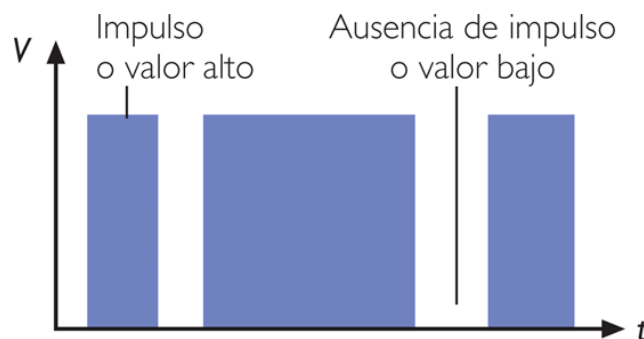


Figura 4: Una señal digital [CALI2008]

Por otra parte, una señal analógica es una señal que varía de forma continua a lo largo del tiempo. La mayoría de las señales que representan una magnitud física (temperatura, luminosidad, humedad, etc.) son señales analógicas, donde estas últimas pueden tomar todos los valores posibles de un intervalo, apreciándose mejor en la Figura 5 [CALI2008].

4.3.2.3 Conversión de señales analógicas a digitales.

Muchas veces se hace necesario interpretar señales analógicas como si fueran señales digitales para poder modificarlas o procesarlas según requerimientos específicos, y esto se logra aplicando la conversión analógico-digital. Esta conversión se basa en tres aspectos principales que consisten en el muestreo, cuantificación y codificación de la señal.

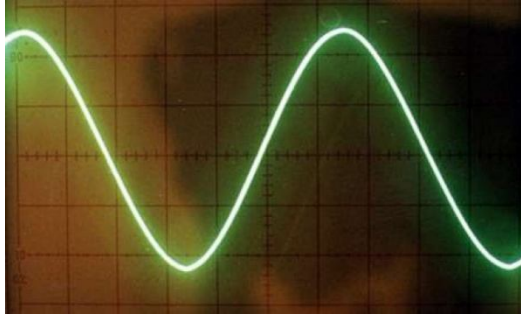


Figura 5. Señal Analógica en un osciloscopio

El muestreo consiste en tomar muestras periódicas de la señal analógica con respecto a su amplitud. La velocidad con que se toma la muestra (número de muestras por segundo) es lo que se conoce como frecuencia de muestreo [CALI2008]. La cuantificación es un proceso que mide el nivel de voltaje de cada una de las muestras con respecto a unos valores preestablecidos, asignándole a cada muestra un valor dependiendo de la amplitud que alcanza la señal analógica en ese instante. La codificación consiste en traducir a código binario los valores obtenidos en el proceso de cuantificación, aunque también es posible “traducirlos” a otro tipo de códigos. Un ejemplo de conversión de una señal analógica a digital se puede apreciar en la Figura 6.

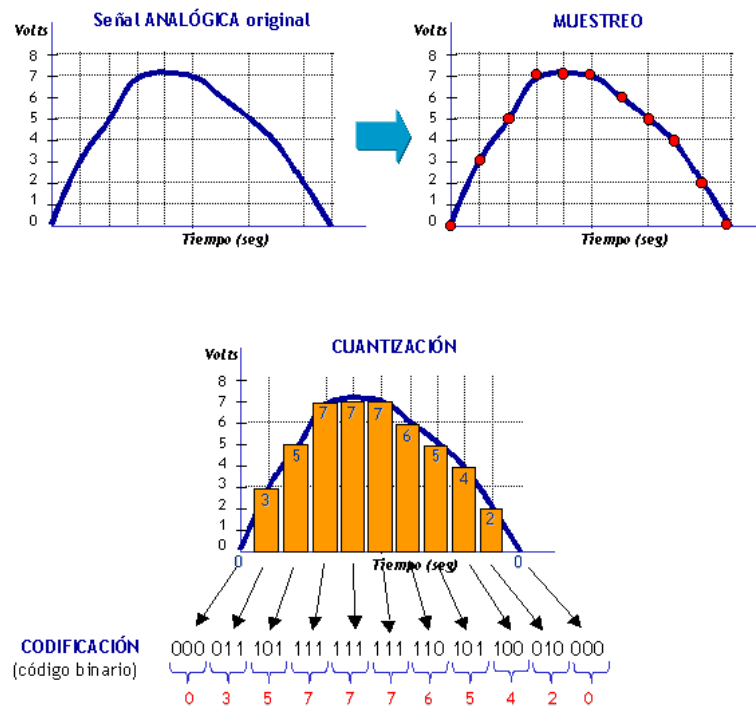


Figura 6. Proceso de conversión analógica a digital. [EVEL2009]

4.3.2.4 Procesamiento digital de señales [CONS2007]

El Procesamiento Digital de Señales (PDS) o DSP (Digital Signal Processing) es un área de la ingeniería que se dedica al análisis y procesamiento de señales digitales de audio, voz, imágenes y video. El procesamiento se hace en señales digitales por diferentes razones:

Una señal digital es más fácil de procesar que una analógica. Las señales que son convertidas a formato digital, además de facilitar la transmisión de estas o su almacenamiento; hacen más barato la implementación del hardware necesario para procesarla que en el procesamiento analógico.

Además las señales digitales requieren usualmente menos ancho de banda y pueden ser comprimidas (es decir reducir la cantidad de información que representa dicha señal, sin pérdida alguna). Sin embargo, hay pérdida de datos durante la cuantificación inherente de información al convertir la información continua (analógica) en discreta (digital); y aún más si las muestras se toman a frecuencias muy bajas.

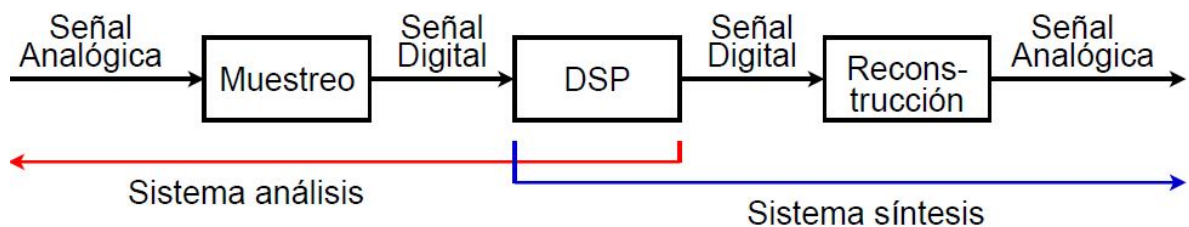


Figura 7. Etapas básicas del PDS

En síntesis, las etapas del procesamiento digital de señales se pueden observar en la Figura 7. Una señal analógica es cuantificada mediante un proceso de conversión digital (Muestreo) luego un DSP (Digital Signal Processor) procesa la señal de la manera deseada, ya sea para hacer un filtro de voz, o compresión de la señal, etc. Este último arroja la información resultante como señal digital, y nuevamente es transformada a su forma natural como señal analógica mediante un proceso inverso al muestreo.

4.3.2.5 Procesadores y microprocesadores

El procesador, también como CPU (Central Unit Process o Unidad Central de proceso) es el componente principal en una computadora digital. Este se encarga de “Pensar”, es decir, interpreta las instrucciones y procesa los datos contenidos en los programas de la computadora. Se conoce como microprocesador a la CPU

que es manufacturada con circuitos integrados orientados a aplicaciones dedicadas. Con los microprocesadores le es posible a una computadora común, delegar tareas que no requieren demasiada capacidad de procesamiento, como lo es ejecutar el programa de impresión en una impresora, o recibir datos en un módem. Los microprocesadores le ahorran a la CPU central una gran cantidad de procesos haciendo más eficiente el uso de las características principales de una CPU como unidad central. [PUBL1999]

4.3.2.6 Introducción a los DSP's [UTFS2004]

Un Procesador Digital de Señales (DSP, sigla en inglés de *Digital Signal Processor*) es un tipo de microprocesador rápido, poderoso y único, porque procesa señales en tiempo real, capacidad de procesamiento necesaria para aplicaciones que no toleran ningún retardo o desfase temporal, como lo son las transmisiones de video (en la TV digital), reproducción de música en formatos digitales (un Compact Disc) o una llamada telefónica por ejemplo.

Cuando se termina el proceso de conversión, los datos ahora digitales, son entregados al DSP. Eventualmente el DSP deberá devolver los datos, pero no sin antes haberlos procesado. Por ejemplo, una señal de audio puede ser adquirida y filtrada para eliminar en gran medida ruido, crujidos de estática, amplificar ciertas frecuencias de interés, eliminar otras, etc. Un DSP permite modificar la información adquirida de la forma en que se desee, y gracias a esto son posibles aplicaciones como: la eliminación del eco en las líneas de comunicaciones, la optimización de imágenes de órganos internos en los equipos de diagnóstico médico, el cifrado de conversaciones en teléfonos celulares para mantener privacidad o el análisis de datos sísmicos para encontrar nuevas reservas de petróleo, entre muchas otras.

4.3.2.7 Clasificación de los DSP's [ARIT2007]

Una forma de clasificar los DSP's y aplicaciones es a través de su rango dinámico, que es un conjunto de datos, muy variantes en su cantidad, que deben ser procesados en el curso de una aplicación. El DSP debe ser capaz de manejar los números generados tanto en la transformación análoga – digital como durante los cálculos (multiplicaciones, sumas, divisiones) con dicha señal.

La capacidad del DSP está en función del ancho de datos (el número de bits manipulados) y el tipo de aritmética que posee (punto fijo o flotante). Un bit es la unidad básica lógica de medida de datos digitales, por ejemplo, una letra representada digitalmente, tendría un ancho 8 bits. La aritmética de punto flotante o coma es un método de representación de números reales que se puede adaptar

al orden de magnitud del valor a representar, es decir, se pueden representar números negativos o positivos de forma binaria. De esta forma, con un número dado de dígitos representativos se obtiene mayor precisión que con la coma fija, debido a que esta no puede representar números negativos. Debido a esta adaptación, permite representar un rango mucho mayor de números (determinado por los valores límite que puede tomar el exponente).

Un procesador puede tener desde 8 bits de ancho de datos. Los rangos varían según la aplicación o los requerimientos de precisión en el procesamiento de la información. Los procesadores de 8 bits son utilizados para tareas que no requieren grandes velocidades de procesamiento, ni demasiada precisión en los datos procesados; el uso de procesadores de 8 bits es común en aplicaciones como control de dispositivos periféricos o de monitoreo.

Los de 16 bits, pueden manejar mucha más información que los de 8 bits. Estos procesadores son comúnmente usados en aplicaciones de sonido básicas, como reproductores de voz, filtros digitales, y ecualizadores. El procesamiento de imágenes simples requiere de un ancho de bits adicional, es decir 24 bits, ya que para este tipo de procesos es prioridad la definición de los datos. Finalmente se tienen los procesadores de 32bits, que son capaces de procesar gráficos 3D y simulaciones científicas. Estas últimas necesitan un rango dinámico mucho mayor y por lo tanto requieren el uso de punto flotante de 32 bits y ADC's de 24 bits. [UTFS2004]

4.3.2.8 Arquitectura de los DSP [UTFS2004]

Las arquitecturas de los computadores actuales son clasificadas como RISC's (*Reduced Instruction Set Computers*) y CISC's (*Complex Instruction Set Computers*). Estos últimos tienen un gran número de instrucciones sumamente poderosas, mientras que la arquitectura RISC posee pocas instrucciones y realiza movimientos de datos entre registros en un ciclo de máquina. Debido a que con la arquitectura RISC se puede alcanzar un más alto rendimiento por medio del uso de un eficiente compilador, la arquitectura CISC's está en desuso.

Es por esto que los DSP's estándar tienen mucho rasgos de una arquitectura tipo RISC, pero ellos son procesadores de propósitos específicos cuya arquitectura es especialmente diseñada para operar en ambientes de alta necesidad de cálculo. Un DSP estándar ejecuta varias operaciones en paralelo mientras que un RISC usa unidades funcionales altamente eficientes que pueden iniciar y completar una instrucción simple en uno o dos ciclos de reloj.

Los DSP's típicos se caracterizan generalmente porque tienen una unidad funcional rápida que puede multiplicar y acumular en un ciclo de instrucción. Un

ciclo de instrucción puede durar generalmente 1 ó 2 ciclos de reloj ya sean DSP's de punto fijo y flotante. Además tienen varias unidades funcionales que realizan operaciones en paralelo, incluyendo accesos a memoria y cálculo de direcciones. Estas unidades funcionales poseen típicamente una unidad principal aritmético-lógica (ALU, Arimethic Logic Unit) que es la encargada de realizar la mayoría de operaciones lógicas básicas, como sumas y restas. Todo esto junto con dos o más unidades de generación de direcciones (AGU). Todas las unidades funcionales poseen su propio conjunto de registros y muchas instrucciones se realizan en un solo ciclo de instrucción. Los DSP tienen varias unidades de memoria en el mismo chip usadas para almacenar instrucciones, datos o tablas. Cada unidad de memoria puede ser accesible una vez en cada ciclo de instrucción.

4.3.2.8.1 Arquitectura Neumann y Harvard

En la arquitectura Neumann la ALU y la unidad de control están conectadas a una sola unidad de memoria que almacena tanto instrucciones de programa como datos. Durante la ejecución de un programa, una instrucción es leída desde la memoria y decodificada, los operandos necesarios son obtenidos (fetched) desde la memoria, y, finalmente, la instrucción es ejecutada. La principal desventaja de esta arquitectura es que la memoria se transforma en el cuello de botella de esa arquitectura debido a que no se pueden hacer proceso de lectura de memoria y ejecución de programa en forma paralela limitando mucho las funciones de acumulación y multiplicación necesarias en la mayoría de los procesos.

La instrucción que con más frecuencia realiza un DSP estándar es la multiplicación y acumulación. Ésta debe ser realizada con eficiencia, y para ello debería ser completada en un ciclo de instrucción. Esto implica que dos valores deben ser leídos desde memoria y (dependiendo de la organización) un valor debe ser escrito, o dos o más registros de direcciones deben ser actualizados, en ese ciclo. Por lo tanto, una longitud grande en la memoria es tan importante como la operación de multiplicación–acumulación.

Varios buses y memorias incluidas en el chip son utilizadas de forma que lecturas y escrituras a diferentes unidades de memoria pueden ser hechas a la vez. Dos memorias son utilizadas en la arquitectura Harvard clásica. Una de ellas es utilizada exclusivamente para datos, mientras que la otra es utilizada para instrucciones. Esta arquitectura alcanza un alto grado de concurrencia (lecturas y escrituras simultáneas) y es por esto que la arquitectura Harvard es la más utilizada hoy en día. En la figura 8 se ilustran ambos tipos de arquitectura

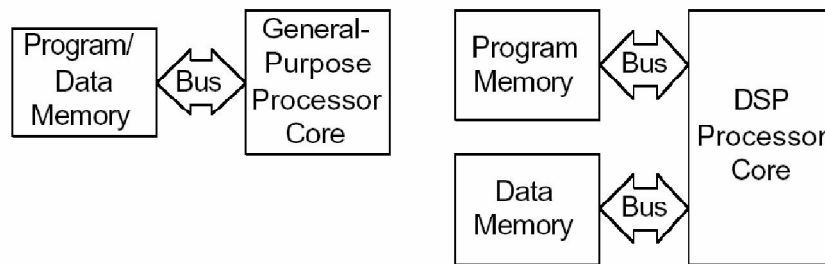


Figura 8: Tipos de arquitectura de memoria entre un Microcontrolador (arquitectura von Neumann) y un DSP (arquitectura Harvard).

4.3.2.9 La memoria en un DSP [ROMM2008]

En los procesadores con arquitectura Harvard hay varios tipos de memoria, y aunque todas se encargan de almacenar datos, cada una tiene un propósito en específico que marcan la diferencia entre estas, tanto en hardware como en software.

4.3.2.9.1 La memoria ROM

La memoria ROM, también conocida como *firmware*, es un circuito integrado programado con unos datos específicos cuando es fabricado. Los chips de características ROM no solo se usan en ordenadores, sino en muchos otros componentes electrónicos también. Los datos que se almacenan en estos chips son no volátiles, lo cual significa que no se pierden cuando se apaga el equipo. Tampoco pueden ser modificados, y en tal caso, necesitan de alguna operación especial para lograr este tipo de operación.

4.3.2.9.1.1 Funcionamiento ROM

Los chips ROM contienen una hilera de filas y columnas, aunque la manera en que interactúan es bastante diferente. Mientras que RAM usualmente utiliza transistores para dar paso a un capacitor en cada intersección, ROM usa un diodo para conectar las líneas si el valor es igual a 1. Por el contrario, si el valor es 0, las líneas no se conectan en absoluto.

Un diodo normalmente permite el flujo eléctrico en un sentido y tiene un umbral determinado, que indica cuanto flujo eléctrico será necesario para dejarlo pasar. Normalmente, la manera en que trabaja un chip ROM necesita la perfecta programación y todos los datos necesarios cuando es creado. No se puede variar una vez que está creado. Si algo es incorrecto o hay que actualizar algo, hay que descartarlo y empezar con uno nuevo. Crear la plantilla original de un chip ROM

es normalmente laborioso dando bastantes problemas, pero una vez terminado, los beneficios son grandes. Una vez terminada la plantilla, los siguientes chips pueden costar cantidades ridículas.

El consumo de energía de estos circuitos integrados es bastante bajo, son bastante fiables, y pueden llevar toda la programación para controlar el dispositivo en cuestión. Los ejemplos más cercanos los tenemos en algunos juguetes infantiles los cuales hacen actos repetitivos y continuos.

4.3.2.9.2 EEPROM y memorias FLASH

EEPROM (*Electrically Erasable Programmable Read-Only Memory*), son un gran paso sobre las ROM iniciales, en términos de utilidad. Los chips no tienen que ser retirados para sobre escribirse, ni se tiene que borrar el chip por completo para cambiar una porción del mismo. En este tipo de memorias se utilizan campos eléctricos para volver a incluir información en las celdas de datos que componen circuitos del chip.

El problema con las memorias EEPROM, es que, aunque son muy versátiles, también pueden ser lentos con algunos productos lo cuales deben realizar cambios rápidos a los datos almacenados en el chip. Los fabricantes respondieron a esta limitación con la memoria flash, un tipo de EEPROM que utiliza un “cableado” interno que puede aplicar un campo eléctrico para borrar todo el chip, o simplemente zonas predeterminadas llamadas *bloques*.

4.3.2.9.3 La memoria RAM [RAMM2006]

RAM son las siglas de *random access memory*, un tipo de memoria de ordenador a la que se puede acceder aleatoriamente; es decir, se puede acceder a cualquier byte de memoria sin acceder a los bytes precedentes. La memoria RAM es el tipo de memoria más común en ordenadores y otros dispositivos como impresoras.

Hay dos tipos básicos de memoria RAM: RAM dinámica (DRAM) y RAM estática (SRAM). Los dos tipos de memoria RAM se diferencian en la tecnología que utilizan para guardar los datos, la memoria RAM dinámica es la más común. La memoria RAM dinámica necesita actualizarse miles de veces por segundo, mientras que la memoria RAM estática no necesita actualizarse, por lo que es más rápida, aunque también más cara. Ambos tipos de memoria RAM son volátiles, es decir, que pierden su contenido cuando se apaga el equipo.

4.3.2.10 Fabricantes de DSP y Familias (general) [UJAE2006]

Actualmente el mercado de los DSP está centrado en cuatro grandes familias que son: Texas Instruments, Motorola, Analog Devices y Microchip, donde cada una posee una serie de características especiales en cada uno de sus DSP.

4.3.2.10.1 Texas instruments

Texas Instruments engloba su conjunto de DSP's mediante el prefijo TMS320. Fue el primer fabricante de estos nuevos dispositivos en el año 1.984. Sus productos están ampliamente consolidados en el mercado y ocupan una posición de privilegio respecto a sus competidores. Sin embargo, en algunos ámbitos de aplicación el mercado lo tienen copado otros fabricantes.

Las ventajas de los DSP's de Texas respecto a los de la competencia son:

- Gama de productos más amplia.
- Líderes en soluciones de mercado.
- Fabricación consolidada y compromiso de entrega.
- Amplia variedad en opciones de encapsulado.
- Mejor soporte técnico desde la aparición de la idea hasta la consecución del producto.
- Bajo coste en sistemas de desarrollo y módulos de evaluación.
- Simuladores hardware fiables.
- Existencia de compiladores cruzados para lenguajes de alto nivel.
- Existencia de Debuggers.
- Existencia de Emuladores en tiempo real.
- Amplia librería de software para aplicaciones.
- Servicio hotline de asistencia técnica, también accesible desde Internet
- Para algunos de sus productos, hay otros posibles fabricantes por lo que la dependencia tecnológica es menor.

4.3.2.10.2 Analog Devices

Los DSP's de la familia ADSP-2100 proporcionan un juego de instrucciones que requieren un único ciclo de instrucción cuando se implementan algoritmos de DSP.

El procesador puede realizar varias operaciones por ciclo. La unidad multiplicador–acumulador (MAC), la ALU y los registros de desplazamiento están separados pero no pueden trabajar en paralelo. Otros registros secundarios siguen cada ejecución de los registros de unidad, permitiendo un rápido cambio de contexto para el procesamiento de instrucciones. Si necesitamos precisión

extendida podemos usar el acumulador de 40 bits de la unidad MAC (incluidos los 8 bits para extensión de signo) como dos registros de 16 bits y uno de ocho, y copiar individualmente la información que contienen en otros registros.

El registro de desplazamiento mueve una entrada de 16 bits a la izquierda o la derecha dentro de un registro de 32. El hardware también permite la detección de exponente y normalización de bloques en coma flotante para incrementar la precisión del DSP de 16 bits. Una aplicación usa el registro de desplazamiento para conversiones entre números en coma fija y en coma flotante.

Los miembros de la familia ADSP-2100 tienen generadores de direcciones X e Y y buses de datos y programa. Cuando en la ejecución utilizamos memoria interna, los buses facilitan los datos X e Y para cada ciclo de MAC. También podemos usar la memoria del programa como memoria de datos para almacenar constantes durante la realización de la instrucción MAC. La memoria de programa de doble puerto permite dos accesos a memoria en un solo ciclo. Para accesos a la memoria externa, el ADSP-2100 tiene un generador de estados de espera programable desde 0 hasta 7 estados posibles.

Los diseñadores de Analog Devices optaron por una palabra de datos de 16 bits y una palabra de instrucción de 24. El tamaño de la palabra de instrucción permite al dispositivo usar más instrucciones complejas y ofrece mayor flexibilidad que la que da un código de operación de 16 bits. La diferencia entre el tamaño de la palabra de código y la palabra de datos requiere una arquitectura Harvard con dos espacios de memoria. Estas arquitecturas, con espacios de memoria separados, son frecuentes en la mayoría de los DSPs, permitiendo realizar instrucciones en paralelo con operaciones de ciclo único MAC. Para diseños con memoria externa, los diferentes anchos de memoria conllevan que si se distribuye la memoria de datos y de programa en tres chips de memoria de 8 bits de palabra, se pierde uno de cada 3 bytes en el área de memoria de datos.

4.3.2.10.2.1 Modo de direccionamiento

Los ADSP-2100 permiten direccionamiento inmediato, directo, indirecto e indexado. Cada generador de direcciones soporta como máximo 4 buffers circulares con 3 registros cada uno. Los registros definen el final, la longitud y las direcciones de acceso. Un generador de direcciones proporciona direccionamiento bit reverse sólo para datos.

4.3.2.10.2 Instrucciones especiales

El ADSP-2100 puede ejecutar condicionalmente la mayoría de las instrucciones. El comando do-until establece una secuencia de instrucciones que pueden ser de longitud arbitraria. Debido a que el ADSP-2100 es un dispositivo 'nonpipelined' no incurre en penalizaciones por ejecutar saltos y llamadas a subrutinas.

4.3.2.10.2.3 Soporte

Analog Devices proporciona un compilador ANSI C, un ensamblador, un linkador y un simulador interactivo. Las tarjetas de evaluación están disponibles para la mayoría de los DSP's de esta familia. Los emuladores en circuito están disponibles para el debug de tarjetas hardware.

4.3.2.10.3 Motorola

Motorola, aunque entró a competir en el mercado de los DSP's relativamente tarde (1987), ha conseguido tener un peso relativamente importante en este mercado. El primer DSP que sacó al mercado (DSP56000/56001) era de coma fija pero de 24 bits. Este valor de precisión ocupa un valor intermedio entre la precisión que aporta la coma fija de 16 bits con la coma flotante de 32 bits. Este DSP desde su aparición tuvo una elevada aceptación sobre todo en lo que se refiere a aplicaciones con audio.

Actualmente la familia de DSP's que Motorola ofrece son:

- Familia de coma fija de 16 bits (DSP561xx)
- Familia de coma fija de 24 bits (DSP5600x)
- Familia de coma flotante de 32 bits (DSP96002)

4.3.2.10.4 Microchip

Microchip se ha caracterizado por ofrecer una gran gama de microcontroladores a gusto de las necesidades de sus clientes, y debido a esto a partir del año 2004 inicio su fabricación en masa de la línea de DSP que son llamados bajo el nombre de dsPIC. Manejan un bus de datos de 16 bits y presentan muchas instrucciones adicionales que no se encuentran en los microcontroladores convencionales de esta misma familia.

Estos dsPIC son ofrecidos a un precio menor que los DSP de otras marcas y ofrecen un amplio rango de características según las necesidades del usuario.

4.3.2.11 ¿Por qué Microchip? [MICR2004]

En Colombia, las familias de DSP con mayor factibilidad de adquisición son la DSP5600x de Motorola y DSP30Fx de Microchip.

Pese a que Motorola ha distribuido kits de desarrollo donde se pueden llevar a cabo pruebas con diferentes tipos de DSP Motorola, sus DSP son de muy baja memoria, y los que poseen especificaciones de mayor nivel son de difícil adquisición.

Por su parte Microchip se ha encargado de fabricar DSP's con características de alto nivel con facilidad de programación y adquisición en el mercado. Además, sus DSP's pueden ser leídos y escritos mediante circuitos programadores convencionales, y debido a su gran velocidad de procesamiento, es posible implementar aplicaciones para procesar imágenes que no necesiten procesamiento en tiempo real.

4.3.2.12 ¿Por qué el dsPIC 30F4011? [USER2004]

El dsPIC30F4011 posee las siguientes especificaciones mostradas en la Tabla 1:

Architecture	16-bit
CPU Speed (MIPS)	30
Memory Type	Flash
Program Memory (KB)	48
RAM Bytes	2,048
Temperature Range C	-40 to 125
Operating Voltage Range (V)	2.5 to 5.5
I/O Pins	30
Pin Count	40
System Management Features	PBOR, LVD
Internal Oscillator	7.37 MHz, 512 kHz
nanoWatt Features	Fast Wake/Fast Control
Digital Communication Peripherals	2-UART, 1-SPI, 1-I2C
Analog Peripherals	1-A/D 9x10-bit @ 1000(kcps)
Comparators	0
CAN (#, type)	1 CAN
Capture/Compare/PWM Peripherals	4/4

Motor Control PWM Channels	6
Quatrature Encoder Interface (QEI)	1
Timers	5 x 16-bit 2 x 32-bit
Parallel Port	GPIO

Tabla 1. Especificaciones del dsPIC30F4011

El dsPIC30F4011 posee una gran ventaja para su manipulación la cual se basa en su tamaño y en la característica de su empaquetado (Dip40) que permite llevar a cabo pruebas en protoboard sin necesidad de sockets o elementos adicionales. Este DSP se muestra en la siguiente figura:



Figura 9: dsPIC30F4011

El estilo de programación y conjunto de instrucciones de este DSP son rígidos y permiten el generado compacto eficiente del código programa principal, debido a que usa compiladores basados en el lenguaje C.

La familia 30Fx cuenta con una gran gama de dispositivos periféricos internos resumidos a continuación:

- Un puerto CAN (Controller Area Network) que le permite al DSP comunicarse a gran velocidad con otros dispositivos similares en un mismo prototipo
- Un conversor analógico digital (ADC Analog-to-Digital Converter) para digitalizar señales externar al DSP (como la Voz Humana), con una resolución de 12 bits para mayor precisión de los datos adquiridos.
- Un decodificador de cuadratura para manejar señales moduladas
- Un modulador de ancho de pulso PWM Pulse Width Modulator Module (PWM)
- Dos módulos UART con buffer FIFO.
- Un módulo I2C™ con 7 a 10 bits de direccionamiento.
- Un temporizador de cuadratura (Quad Timer Module) para modular señales.

Todas estas características hacen del dsPIC30F4011 casi ideal para los posibles requerimientos de hardware necesarios para realizar rutinas de OCR sintetizadas.

4.3.3 Estándar RS-232

El estándar RS-232 designa una norma para el intercambio serial de datos binarios entre un DTE (Equipo terminal de datos) y un DCE (Data Communication Equipment, Equipo de Comunicación de datos), aunque existen otras situaciones en las que también se utiliza la interfaz RS-232.

El estándar RS-232 permite la transmisión síncrona y asíncrona donde la transmisión asíncrona es la más frecuente. El conector original es del tipo DB - 25 (25 pines), aunque el más común es el de 9 pines (DE-9), que es más barato y más extendido para algunos periféricos como módems, mouse o teclados [PIPE2009].

4.3.3.1 Normalización de Aspectos del Estándar RS-232

La normalización de los aspectos corresponde a cada una de las características mecánicas, eléctricas o funcionales necesarias para llevar a cabo una transmisión de datos de forma serial.

4.3.3.1.1 Mecánico

El aspecto mecánico trata de las características físicas del estándar como por ejemplo el hecho de que RS-232 utiliza un conector DB-9 macho para el DTE y hembra para el DCE.

La conexión entre DTE y DCE es simple. Cada pin debe conectarse con su par equivalente (el 1 con el 1 o el N con el N). Cada pin tiene asignado una función tal y como se muestra en la figura 10. Los nombres de las líneas están puestos desde el punto de vista del DTE. Así, el pin 2 es la línea TxD (transmisión de datos) pero obviamente eso no es cierto en ambos equipos, sólo en el DTE. En el DCE, por el contrario es la línea por la que recibe los datos del DTE [ACWB2009].

Generalmente el macho es el DTE y la hembra el DCE, pero muchas veces esta norma, y otras de la RS-232, no siempre son seguidas por los fabricantes, y por esta razón no siempre es fácil manejar esta interface. La Figura 11 muestra un conjunto de conectores DB-25 machos (arriba-izquierda) y hembras (arriba-centro). También pueden observarse conectores DB-9 (mouse) y dispositivos adaptadores DB-25 a DB-9 (Parte inferior de la imagen) [PIPE2009].

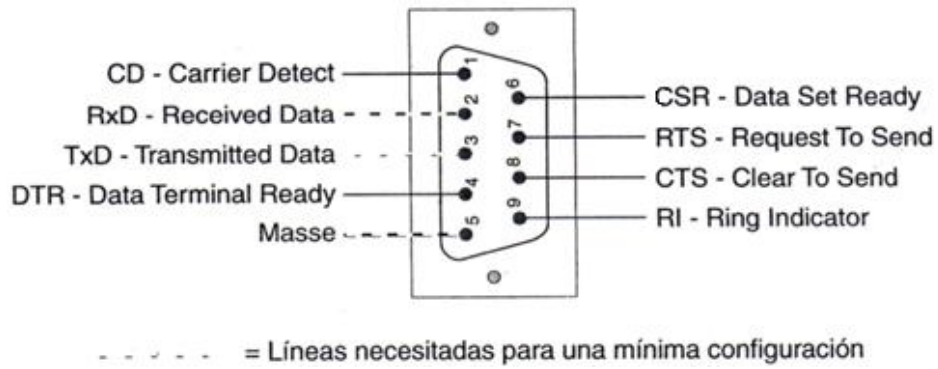


Figura 10. Disposición de pines del conector RS-232



Figura 11. Conectores DB-25 y DB-9

La longitud máxima del cable entre DTE y DCE depende de la calidad de éste y de la velocidad de transmisión utilizada. En principio la norma recomienda que no sea superior a 15 metros para una velocidad de 20Kbits/seg.

4.3.3.1.2 Eléctrico

La subnorma eléctrica de la RS-232 es la V28. La norma fija una transmisión en modo común (cada circuito tienen una referencia a tierra y esta es común para todos los circuitos). Los circuitos son punto a punto, es decir, un driver con un sólo receptor de la señal [PIPE2009].

La señal es bipolar con lógica invertida utilizando los valores de la Tabla 2:

Voltaje	Señal	Nivel Lógico	Control
+3 a +15	Espacio	0	On
-3 a -15	Marca	1	Off

Tabla 2. Valores de voltaje asignados a los niveles lógicos.

Los valores de voltaje se invierten desde los valores lógicos. Por ejemplo, el valor lógico más positivo corresponde al voltaje más negativo. También un 0 lógico corresponde a la señal de valor verdadero o activada. Por ejemplo si la línea DTR está al valor 0 lógico, se encuentra en la gama de voltaje que va desde +3 a +15 V, entonces DTR está listo (*ready*). La ausencia de señal (0 voltios) queda diferenciado del 0 y 1 lógicos.

La tensión máxima de operación es ± 25 voltios y la carga máxima es de 3Kohm a 7Kohm, con una corriente máxima de 500mA. En adición a los estados lógicos y los niveles de voltaje empleados en la transmisión RS-232, también existen algunos dispositivos que permiten convertir los voltajes a niveles TTL.

El MAX232 es un circuito integrado que convierte los niveles de las líneas de un puerto serie RS232 a niveles TTL y viceversa. Lo interesante es que sólo necesita una alimentación de 5V, ya que genera internamente algunas tensiones que son necesarias para el estándar RS232. Otros integrados que manejan las líneas RS232 requieren dos voltajes, +12V y -12V. Sus parámetros importantes y la disposición de los pines se pueden apreciar en la figura 12 [ROBO2009].

El MAX232 soluciona la conexión necesaria para lograr comunicación entre el puerto serie de una PC y cualquier otro circuito con funcionamiento en base a señales de nivel TTL/CMOS. El circuito integrado posee dos convertidores de nivel TTL a RS232 y otros dos que, a la inversa, convierten de RS232 a TTL. Estos convertidores son suficientes para manejar las cuatro señales más utilizadas del puerto serie del PC, que son TX, RX, RTS y CTS, donde TX es la señal de transmisión de datos, RX es la de recepción, y RTS y CTS se utilizan para establecer el protocolo para el envío y recepción de los datos.

4.3.3.1.3 Funcional

Estas características están asignadas para la norma asíncrona conformada por nueve líneas. El tipo de cable utilizado para la transmisión es denominado cable null-módem, el cual es un cable de comunicaciones serie con una modificación en su configuración de hilos, que permite a dos computadores comunicarse en modo asíncrono directamente.

Para lograr esto, los hilos de señal de transmisión y recepción se cruzan, de modo que la salida de transmisión en un extremo del cable se conecta con la entrada de recepción en el otro extremo, y viceversa. Esto mismo se hace también con las líneas de control de módem presentes en la interfaz RS-232, de tal manera que el funcionamiento del computador remoto de cara al ordenador local reproduzca el comportamiento de un módem.

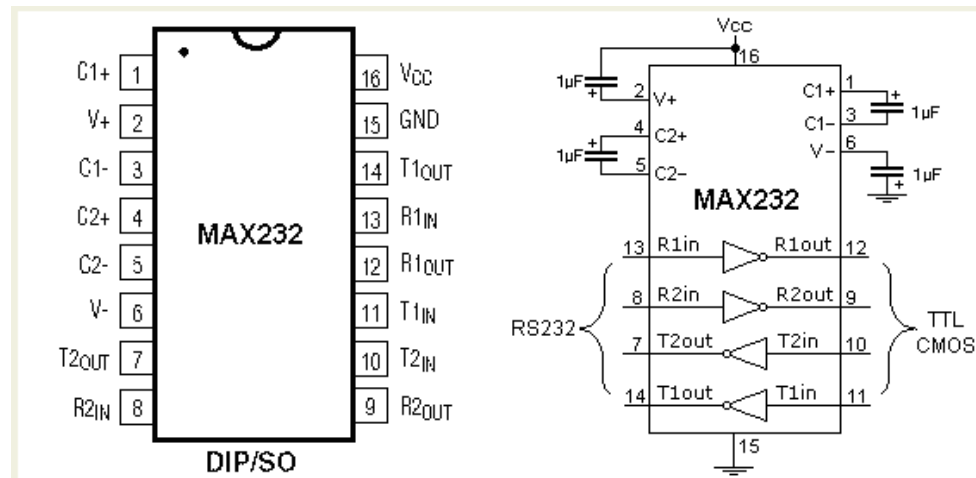


Figura 12. Circuito integrado MAX232 [ROBO2009]

Las diferentes formas de conectar el cable para este tipo de comunicación se muestran en la Figura 13 [UCOL2009].

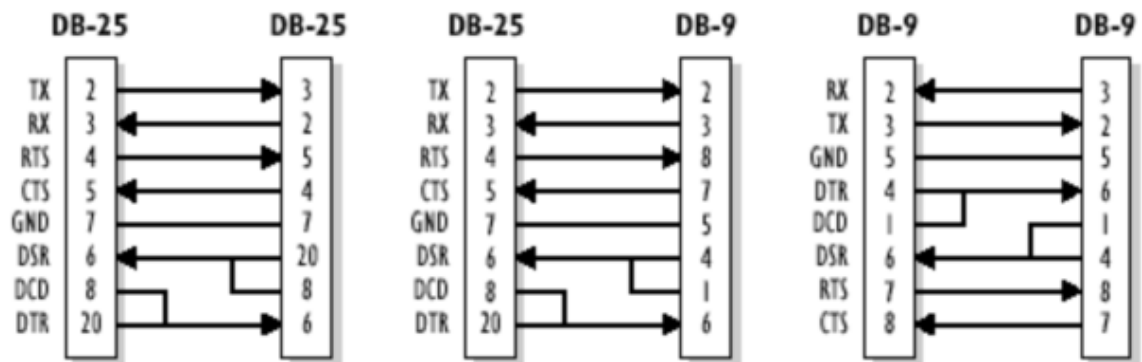


Figura 13. Conexión de pines para un cable Null-Modem. [UCOL2009].

4.3.4 Módulo de Cámara VGA CMOS C328R

Este módulo permite realizar la captura de imágenes en diferentes formatos y resoluciones, permitiendo de igual manera la realización de imágenes en formato JPEG con una resolución de hasta 640x480 pixeles.

Una de sus ventajas es su diminuto tamaño, tal como se ve en la Figura 14, ya que sus dimensiones son de 20x28 mm. Además, en comparación con otros

módulos de cámaras VGA, este posee dos pines para comunicación vía serial, uno para transmisión y uno para recepción, lo que permite una mejora en la comunicación con un computador o con diferentes componentes electrónicos que permitan el envío y recepción de datos serialmente.

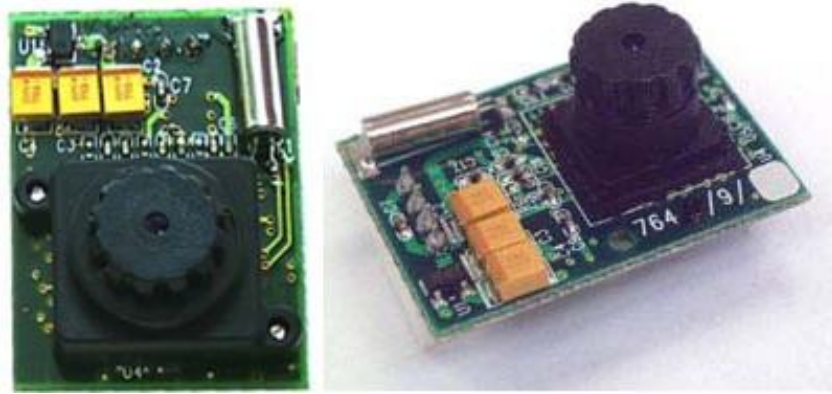


Figura 14. Módulo de cámara VGA CMOS C328R.

Algunas de las características de este módulo son

- Tamaño reducido 20x28mm.
- Resolución QVGA o VGA.
- Voltaje de operación de 3.3v
- Bajo consumo de corriente, 60 mA.
- Uso de comandos amigables para controlar el módulo.
- Interface de comunicación UART de hasta 115,2 Kbps.
- Auto detección del baudrate para establecer comunicación con el host.
- Posibilidad de cambiar el lente.

Como se menciona anteriormente, la gran ventaja de este módulo es su comunicación vía serial, ya que aunque existen diferentes módulos que también permiten la comunicación serial, el C328R es el de menor costo y funcionamiento con la misma calidad de imagen, tal como se ve en la Figura 15.



Figura 15. Imagen capturada con el módulo C328R con resolución de 320x240 píxeles.

4.4 ESTADO DEL ARTE

Muchas soluciones se han dado al problema de la lectura remota de contadores, donde en muchos casos se han diseñado nuevos modelos de contadores electrónicos, o se han implantado sistemas interviniendo en los contadores existentes.

Actualmente en España la empresa CONTAZARA provee una solución de contadores digitales para el servicio de medición de agua. El dispositivo es de aproximadamente 10 cm³ (el más pequeño), en el cual se debe cambiar el contador anterior para colocar uno nuevo y adicionalmente conectarlo a la red eléctrica para la alimentación del dispositivo. La lectura se realiza mediante una turbina interna que calcula la cantidad de agua que pasa por ella. Provee lecturas en tiempo real a través de internet, servicio GSM o vía telefónica. El costo (según usuarios del producto) es de aproximadamente 180 euros (\$486.000).

De igual manera se han diseñado diversos tipos de contadores de energía eléctrica que permiten aparte de captar el consumo en tiempo real, proveen un sistema de alarma que informa al usuario que está consumiendo demasiado rápido mucha energía o que ya ha superado un límite establecido. Estos contadores funcionan mediante un dispositivo inalámbrico que se agrega a las conexiones del contador tradicional y vía inalámbrica informa al usuario los datos obtenidos. Su precio está alrededor de los U\$70.

La empresa GUTSON de Colombia S.A. ofrece dispositivos de lectura remota de todo tipo de contadores usando módulos que envían la información a través de radio frecuencias, redes celulares o vía satélite. También ofrecen una arquitectura de concentración de lectura mediante nodos re-transmisores subestaciones. La gran desventaja radica en que los contadores deben ser intervenidos físicamente para que entreguen lecturas de forma compatible con el resto del hardware, limitándose a algunos modelos de contadores modernos.

En Colombia, la principal modalidad de lectura de contadores se realiza de forma visual, donde un grupo de personas que trabajan para una empresa prestadora de servicios públicos, debe recorrer a pie las calles de las ciudades para ver que lectura muestra el contador para registrar ese dato en una planilla.

5. METODOLOGÍA

La metodología en espiral, es la empleada para el desarrollo del sistema prototipo. En esta, a medida que se avanza y se realizan pruebas, el sistema es retroalimentado y reevaluado continuamente para obtener un resultado lo más eficiente posible, esto se puede entender de una mejor manera con ayuda de la Figura 16.



Figura 16. Metodología en espiral.

Para completar el proyecto, es necesario establecer las posibles etapas o fases de todo el proceso llevado a cabo, y en cada una de ellas se deben aplicar cuatro pasos básicos (Requerimientos, Definición de la solución, Desarrollo y Pruebas) como requisito para continuar con la siguiente etapa. Si todo el proceso se pudiera ver representado en una imagen, seguramente luciría como una 'espiral' tal y como se muestra en la Figura 17. A continuación se explica detalladamente cada etapa.

5.1 Soporte e iluminación para el módulo CMOS C328R

Uno de los requisitos principales es colocar el módulo CMOS sobre el contador monofásico de energía eléctrica para capturar las imágenes necesarias. Esta tarea se requiere un sistema de iluminación adicional que brinde la cantidad de luz suficiente para lograr imágenes con la mayor nitidez posible.



Figura 17. Etapas de la metodología en espiral, aplicada al proyecto.

Esta etapa debe desarrollarse simultáneamente con la etapa de comunicación serial entre el módulo CMOS y MATLAB, ya que las imágenes deben visualizarse en este programa para determinar la mejor posición del módulo sobre el contador y corregir los posibles errores de iluminación que puedan surgir. Esta tarea se lleva a cabo en aproximadamente 1 semana.

5.2 Comunicación serial entre el módulo de cámara CMOS C328R y MATLAB

Debido a que el módulo CMOS permite comunicación únicamente de forma serial, se hace necesario el desarrollo de un algoritmo en MATLAB que permita enviar los comandos necesarios para adquirir y almacenar las imágenes provenientes de este módulo para desarrollar posteriormente el algoritmo de OCR. El tiempo necesario para llevar a cabo este proceso es de 3 semanas.

5.3 Desarrollo del algoritmo OCR en MATLAB

El desarrollo de este algoritmo es el que más tiempo consume debido a que debe realizarse teniendo en cuenta que solo las instrucciones más simples que maneja este programa pueden ser ejecutadas de la misma manera en el DSP, por lo tanto

el programa desarrollado solo debe contener los comandos más básicos y al mismo tiempo debe responder a las requerimientos deseados.

A continuación y partiendo de este hecho, el programa debe poder tomar fotos de prueba del contador monofásico de energía eléctrica y una vez se tienen almacenadas las imágenes, se procede a realizar el algoritmo OCR en MATLAB, proceso que tarda entre 3 y 4 semanas.

5.4 Comunicación serial entre el DSP y MATLAB

Esta etapa es muy importante, ya que permite transferir imágenes de muestra hacia el DSP. Además, una vez adecuado el módulo UART en el DSP, se puede lograr de una manera más eficaz la comunicación del DSP con otros dispositivos externos, como lo pueden ser memorias de datos externas o el módulo CMOS. Esta labor se demora aproximadamente 3 semanas.

5.5 Sintetizar el algoritmo de OCR en el DSP

Cuando el programa diseñado en MATLAB para ejecutar la rutina de OCR ha sido finalmente terminado, se debe proceder a sintetizarlo sobre el DSP. La gran ventaja de este proceso es que muchos de los comandos utilizados en MATLAB también se pueden ejecutar en el DSP. Esta labor se desarrolla lentamente, realizando para cada bloque de código la comprobación respectiva, hasta terminar completamente el programa. Para esta etapa es necesario emplear 3 semanas, ya que se deben realizar gran cantidad de pruebas logrando la perfecta ejecución del algoritmo de OCR en el DSP.

5.6 Comunicación entre el DSP y el módulo CMOS C328R

Una vez realizado el programa tanto en MATLAB como en el DSP para llevar a cabo la comunicación serial, se necesita comunicar el módulo CMOS con el DSP diseñando comandos que permitan sincronizar el módulo, elegir la velocidad de comunicación como factor importante, para de esta manera poder realizar captura de imágenes en el DSP.

Esta etapa consume una gran cantidad de tiempo, ya que las imágenes que se almacenen en el DSP se deben acomodar de la mejor manera posible, por lo cual se hace necesario un periodo de 3 semanas para completar la etapa.

5.7 Visualización del resultado del OCR en una interfaz externa (LCD)

Una vez que el algoritmo de OCR ha sido sintetizado en el DSP, el resultado final que arroja este algoritmo debe visualizarse de alguna manera en una interfaz externa que para este caso se compone de una pantalla de cristal líquido o *liquid cristal display* (LCD). Esta etapa no lleva mucho tiempo, pues la comunicación entre el DSP y esta LCD no es muy complicada, razón por la cual solo se dedica una semana para su desarrollo.

6. DESARROLLO

Una vez definida la metodología a seguir, es mucho más fácil observar la forma en que debe plantearse el desarrollo del proyecto. Por lo tanto es necesario documentar el proceso implementado y explicar de manera clara cada uno de los aspectos que se tuvieron en cuenta en la solución de cada etapa:

6.1 Soporte e iluminación del módulo CMOS C328R.

En el diseño del soporte donde se situó el módulo, se tuvieron en cuenta varios criterios:

- ✓ El tamaño que ocupa el módulo CMOS, incluyendo el lente de enfoque y los cables de alimentación y comunicación.
- ✓ La distancia mínima requerida para lograr un enfoque óptimo de la imagen.
- ✓ El tamaño de la cubierta transparente del contador.
- ✓ El posible tamaño del sistema de iluminación y sus cables de alimentación.
- ✓ Una cubierta o tapa removible para instalar la cámara
- ✓ Un conector RJ11 para extender el cableado interno hacia el circuito de control de la cámara.

Un bosquejo del diseño se apreciar en la figura 18:

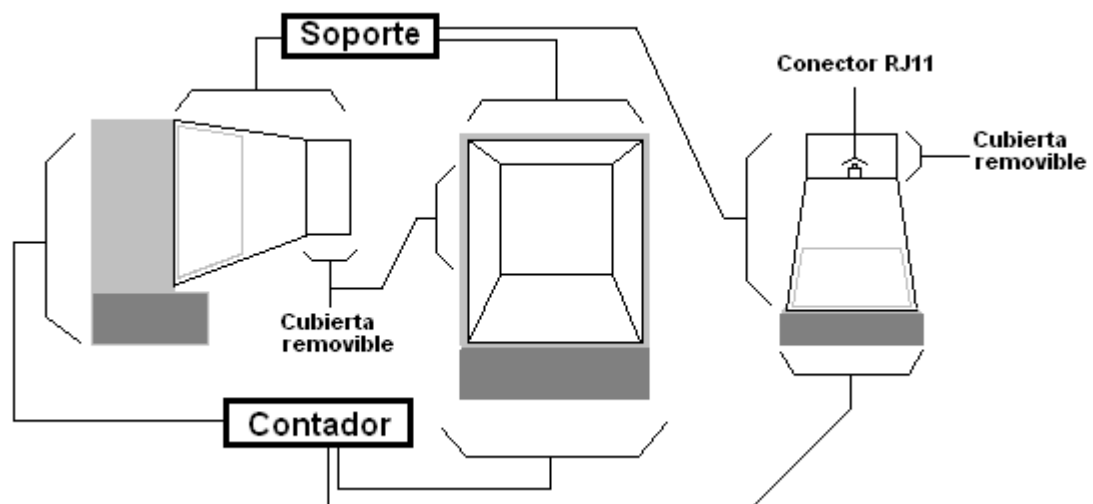


Figura 18. Vistas: lateral, superior y frontal del soporte para el módulo CMOS.

Una vez diseñada la base que contendrá el módulo CMOS era necesario colocar en el interior de ésta el sistema de iluminación de luz blanca. Este debía situarse de tal manera que la zona de interés del contador tuviera una iluminación lo más

uniforme posible. La solución fue colocar una fila de seis LED's (Diodo Emisor de Luz) blancos de alto brillo en la pared interior del lado inferior de la base, de forma inclinada para no iluminar directamente la cubierta transparente del contador (para evitar haces de luz en las imágenes) tal como se puede apreciar en la figura 19.

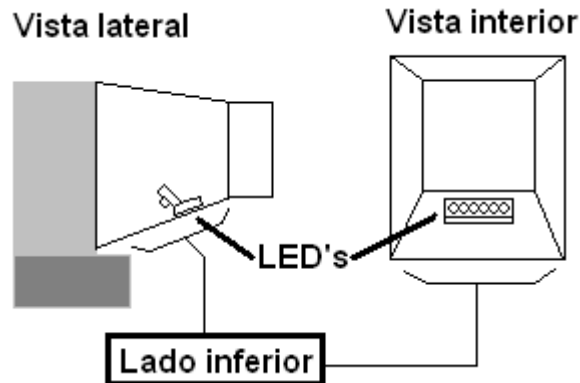


Figura 19. Vista lateral e interior del soporte para el módulo CMOS.

6.2 Comunicación serial entre el módulo de cámara CMOS C328R y MATLAB.

Para poder realizar la captura de imágenes, primero se realizó un estudio acerca del tipo de cámara a utilizar, tomando como factor importante el tamaño que debía tener la misma.

En un principio surgió la idea de utilizar una cámara web, porque debido a su reducido tamaño, resultaba ideal para adaptarla al soporte y así poder realizar más fácilmente la captura de imágenes. Además las imágenes que se pueden obtener desde este dispositivo son nítidas, sin mucho ruido y tienen la resolución suficiente para procesarlas..

El gran problema al querer utilizar las cámaras Web con propósitos es que las empresas fabricantes no brindan documentación libre sobre la manera de manipular estos dispositivos. Este inconveniente replantea el uso de la cámara web.

Entonces se optó por utilizar módulos de cámaras digitales que puedan ser utilizados para proyectos de electrónica, que fueran fáciles de manipular y configurar. Con esta idea se realizó una búsqueda minuciosa del modelo de cámara a elegir, ya que existen muchos en el mercado, y su precio varía según la aplicación requerida.

La primera opción de fue la cámara Toshiba TCM8230MD, que con su reducido tamaño y características permite la captura de imágenes con resolución de hasta 640x480 pixeles. Este módulo se puede apreciar en la Figura 20.

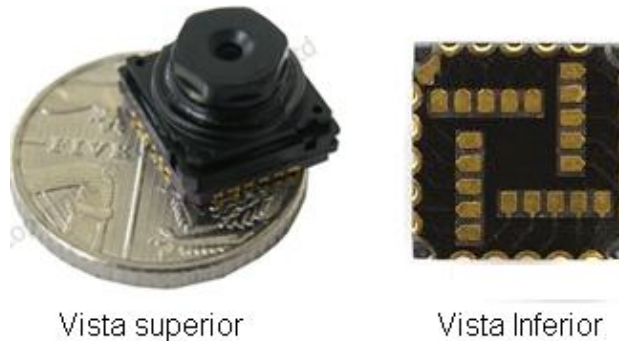


Figura 20. Módulo de cámara VGA de Toshiba TCM8230MD [TOSH2009].

La forma de configuración y la adquisición de datos de éste dispositivo se realiza en forma paralela, esto demandaba el diseño de un circuito de control y conversión paralelo-serie para poder establecer una comunicación bajo el estándar RS-232. Además, éste módulo no posee ningún conector directo, en cambio tiene 20 puntos de salida de dimensiones muy pequeñas; entonces para poder utilizar la cámara se necesitaba fabricar una tarjeta o placa en donde ésta se pudiese colocar.

Como se necesitaban muchos elementos adicionales para su funcionamiento, esta opción fue descartada. Por todo esto, la mejor opción encontrada fue la cámara CMOS C328R, que es de bajo costo y posee una placa de soporte con terminales de transmisión, recepción y alimentación listos para su uso. Este módulo se puede apreciar con más detalle en la Figura 21.

6.2.1 Conexión con el módulo CMOS C328R.

Para llevar a cabo la configuración y manipulación de esta cámara, fue necesario conocer sus especificaciones técnicas y rutinas de configuración. Un resumen de éstas pueden observarse en el Anexo 1.

Una vez conocido que el rango de voltajes usados es de 0v a 3.3V y que la comunicación de la cámara se hace mediante una interfaz serial asíncrona, se diseñó el circuito adecuado para hacer funcionar al C328R.

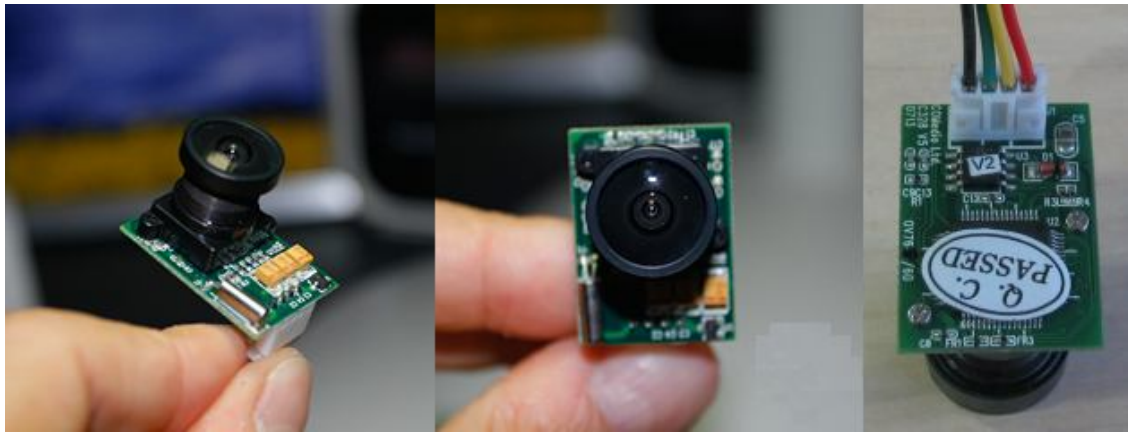


Figura 21. Módulo de cámara CMOS C328R, placa y conectores [OMNI2007]

6.2.1.1 Circuito de alimentación.

Teniendo en cuenta que la mayoría de dispositivos empleados para el desarrollo del proyecto (DSP, MAX232, LCD) utilizan lógica TTL (5V), y que la cámara debía alimentarse con 3.3V, se implementó el circuito mostrado en la Figura 22:

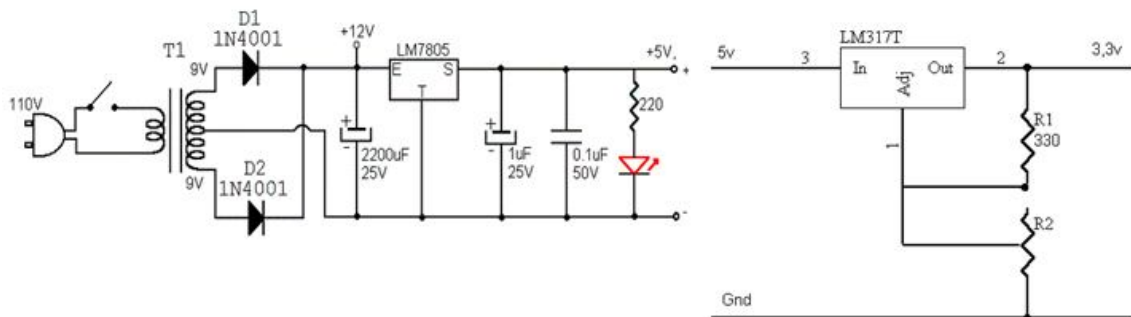


Figura 22. Diseño de fuente de voltaje con salida de 5V y 3.3V.

La fuente de voltaje diseñada entrega una salida de 5V constantes con una corriente máxima de 1.5A. A partir de esta tensión, se obtuvo fácilmente con un regulador ajustable, los 3.3V requeridos para el funcionamiento del C328R.

6.2.1.2 Comunicación del Módulo CMOS C328R con dispositivos externos bajo el estándar RS-232.

La forma de transferencia de datos desde el módulo es serial asíncrona. Bajo esta premisa, se pensó en comunicar el módulo CMOS con cualquier otro dispositivo

utilizando el estándar RS-232. Este estándar maneja unos niveles particulares de voltaje, ya que un 0 lógico en TTL (5V) se representa por un voltaje entre 3V y 15V en RS-232, y un 1 lógico se representa por un voltaje entre -3V y -15V [EIAT1993]. Un ejemplo de esta conversión se puede apreciar en la Figura 23.

Como los datos que se transmiten bajo el estándar RS-232 desde un dispositivo TTL o CMOS deben sufrir una conversión en los niveles de voltaje, Era necesario usar circuito que realizase esta tarea, y el circuito integrado MAX232 viene diseñado para ello.

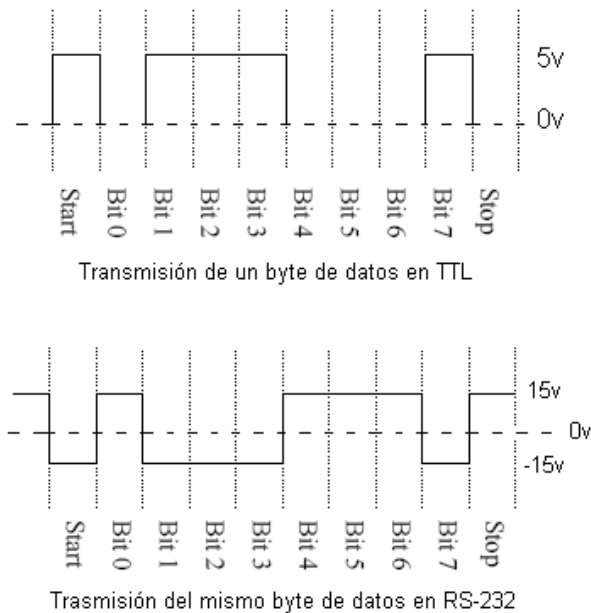


Figura 23. Ejemplo de conversión TTL a RS-232[EIAT2005].

La razón principal por la cual se usó el estándar RS232, se centra en que los datos que se recibiesen o se envasen al módulo CMOS se debían manipular con la ayuda del entorno MATLAB (ya que en este entorno se puede manejar fácilmente un puerto serial de comunicaciones de un ordenador común), con el fin de facilitar las pruebas que permitieran realizar la configuración adecuada del módulo. La hoja de especificaciones del MAX232 provee una configuración que se muestra en la figura 24.

Como el rango de voltajes TTL de entrada/salida es 0V a 5Vdel MAX232, y el de la cámara es de 0V 3.3V, entonces fue necesario realizar un circuito adicional que permitiese acoplar el C328R al MAX232.

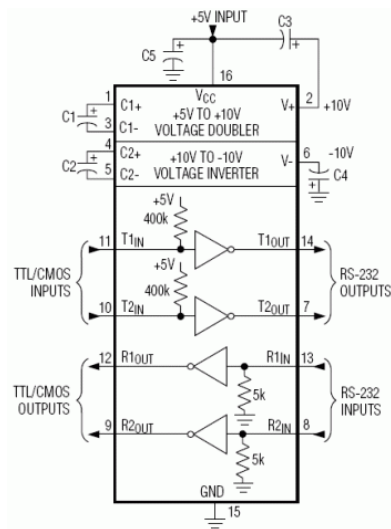


Figura 24. Circuito de operación típico del MAX232 [MAXI2006].

En la transmisión de datos desde el módulo CMOS (convertir 3.3V en 5V), se utilizó el circuito integrado 74LS14, el cual es un arreglo de un compuertas NOT con disparador Trigger Schmitt, garantizando el paso por la región prohibida de voltajes TTL (cerca a los 3.3V), lo que implica 'negar' dos veces la señal de entrada para no modificar su lógica. El esquema de conexión entre el C328R y el 74LS14 se puede apreciar en la figura 25.

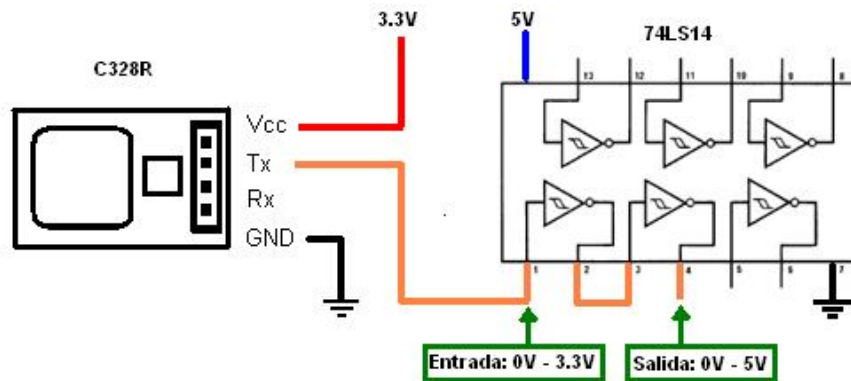


Figura 25. Circuito convertidor de niveles de voltaje CMOS a TTL.

En la recepción de datos implementó el siguiente circuito que permite convertir 5V a 3.3V:

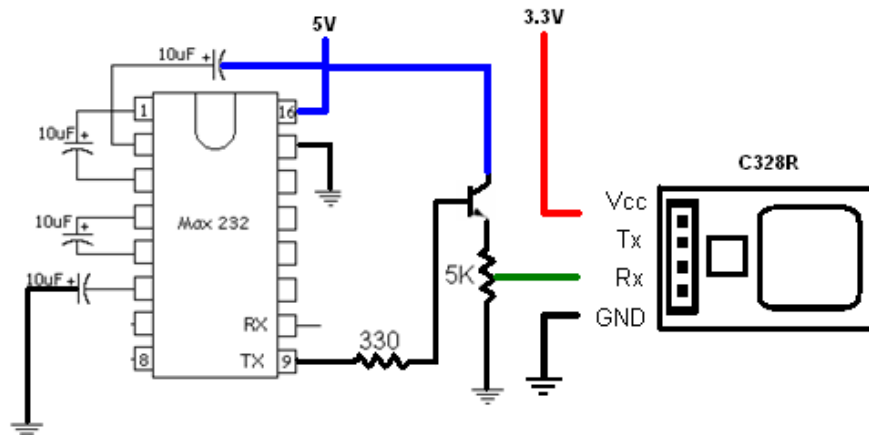


Figura 26. Conversión de niveles de voltaje TTL a CMOS.

Como se puede apreciar en la figura anterior, la salida TTL del MAX232 se conecta a un transistor que funciona como interruptor y junto con un reóstato conectado entre el colector y tierra. Con el fin de obtener exactamente 3.3v en la entrada 'Rx' del C328R, el reóstato fue ajustado consecuentemente.

Con la implementación de los circuitos anteriores, fue posible realizar pruebas del funcionamiento de la cámara para obtener las imágenes. Para llevar a cabo dichas pruebas, se creó una rutina en el entorno de programación de MATLAB (Trial Version), la cual era capaz de recibir y enviar datos hacia el módulo, mediante el puerto serial de un ordenador. En la figura 27 se muestra el entorno de programación que posee MATLAB.

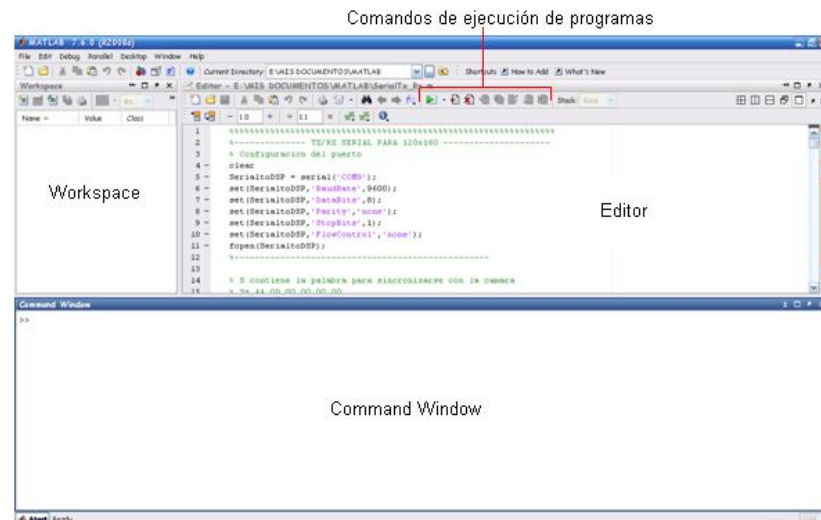


Figura 27. Entorno de programación de MATLAB [MATL2009].

El 'Workspace' muestra cada una de las variables que se utilizan en el 'Editor', como en las rutinas que se ejecuten en el 'Command Window'. El 'Editor' es el espacio en donde se elaboran los programas. Por último el 'Command Window' permite visualizar los resultados de los programas que se ejecuten desde el 'Editor'.

Por último fue necesario realizar una conexión física ente el puerto serial del ordenador y el MAX232 tal como se ve en la Figura 28.

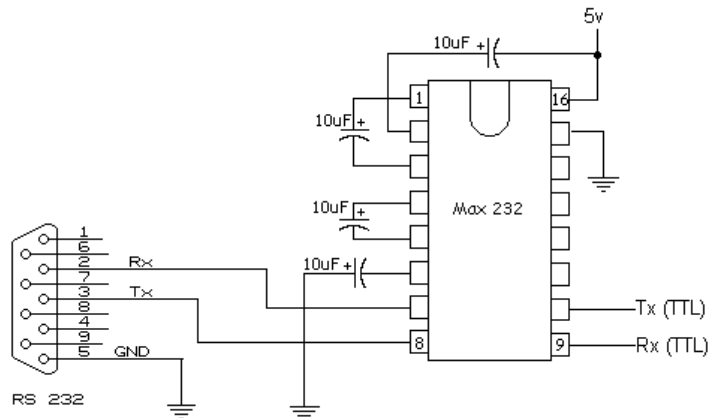


Figura 28. Conexión entre el puerto serial y el MAX232

6.2.2 Rutinas de configuración del módulo CMOS.

Cada comando que se envía o se recibe desde la cámara consta de 6 bytes. Un ejemplo de esto se puede apreciar en la figura 29.

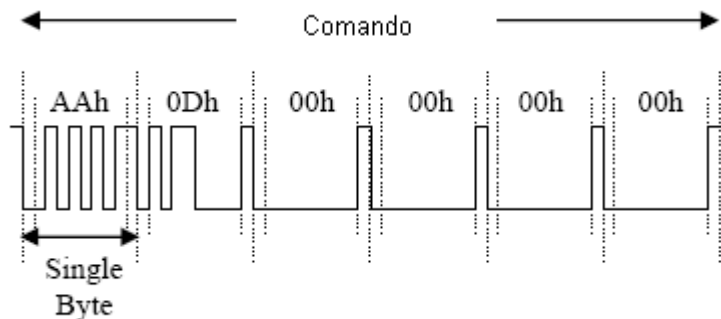


Figura 29. Ejemplo de comando del módulo CMOS

Ya conociendo la estructura de los comandos de configuración, fue posible analizar las funciones de cada uno de ellos, apoyándose en la información provista en el manual del usuario del C328R, los cuales se encuentran en la Tabla 3.

Command	ID Number	Parameter1	Parameter2	Parameter3	Parameter4
Initial	AA01h	00h	Color Type	RAW Resolution (Still image only)	JPEG Resolution
Get Picture	AA04h	Picture Type	00h	00h	00h
Snapshot	AA05h	Snapshot Type	Skip Frame Low Byte	Skip Frame High Byte	00h
Set Package Size	AA06h	08h	Package Size Low Byte	Package Size High Byte	00h
Set Baudrate	AA07h	1st Divider	2nd Divider	00h	00h
Reset	AA08h	Reset Type	00h	00h	xxh*
Power Off	AA09h	00h	00h	00h	00h
Data	AA0Ah	Data Type	Length Byte 0	Length Byte 1	Length Byte 2
SYNC	AA0Dh	00h	00h	00h	00h
ACK	AA0Eh	Command ID	ACK counter	00h / Package ID Byte 0	00h / Package ID Byte 1
NAK	AA0Fh	00h	NAK counter	Error Number	00h
Light Frequency	AA13h	Frequency Type	00h	00h	00h

Tabla 3. Funciones de los comandos de configuración del módulo C328R [SING2007].

Para realizar la captura de imágenes en la resolución y gamma de colores apropiada, se desarrolló un programa en MATLAB que se encarga de sincronizar y configurar el módulo, regido por el protocolo que recomienda el fabricante de la cámara. La sincronización entre MATLAB y el módulo, se logra enviando el comando [AA 0D 00 00 00 00] denominado 'SYNC', hasta que se reciba un comando como respuesta desde el módulo tal y como se muestra en la figura 30.

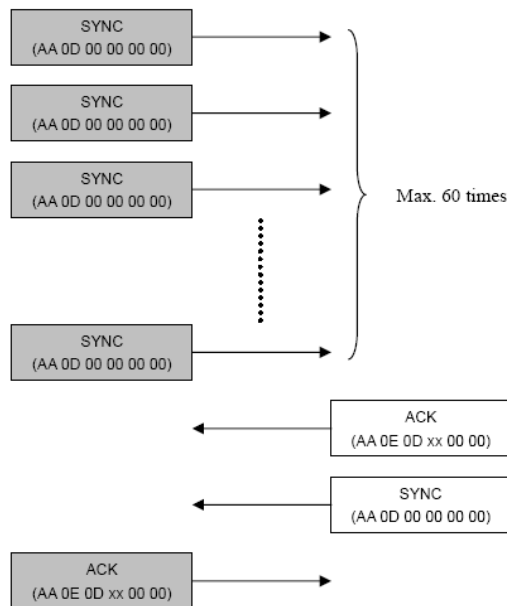


Figura 30. Ejemplo de sincronización entre el host y el módulo CMOS

Cuando la comunicación con la cámara estuvo sido establecida, fue posible enviar los comandos de configuración, en los cuales se establecieron los valores de resolución, paleta de colores y formato de la imagen. El esquema de lo anterior se puede apreciar en la Figura 31.

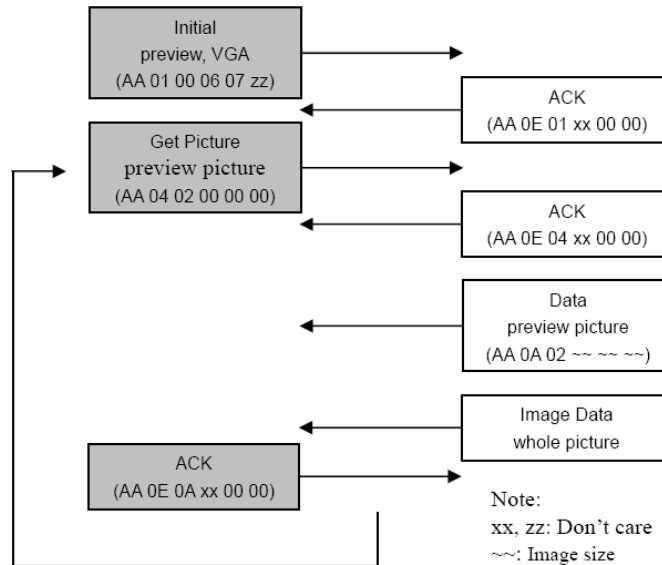


Figura 31. Ejemplo del proceso de captura de una imagen

Las imágenes a utilizar en el proyecto debían ser de baja resolución y en escala de grises. Para fijar dichas características se realizaron los siguientes cambios en los comandos de configuración:

6.2.2.1 Comando 'Initial'

Este comando permite configurar el tamaño de la imagen y el tipo de color. Con ayuda de la tabla 3, se determinaron los parámetros de este comando:

Tipo de Color (Parámetro 2)

2 bit en escala de grises	01h
4 bit en escala de grises	02h
8 bit en escala de grises	03h
12 bit en color	05h
16 bit en color	06h
JPEG	07h

Resolución de visualización (Parámetro 3)

80x60	01h
160x120	03h

Resolución JPEG (Parámetro 4)

80x64	01h
160x128	03h
320x240	05h
640x480	07h

Entonces, si se desean obtener imágenes en escala de grises de 8 bits, con una resolución de 160x120, la palabra de configuración inicial es: [AA 01 00 03 03 03]

6.2.2.2 Comando 'Get Picture'

Get Picture le dice a la cámara que capture una imagen y que la envíe. Los formatos en que se pueden obtener una imagen están descritos a continuación:

Tipo de Imagen (Parámetro 1)

Snapshot Picture	01h
Preview Picture	02h
JPEG	05h

Como los comandos anteriores se definieron para obtener imágenes en 'Preview Picture', la palabra de configuración 'Get Picture' es: [AA 04 02 00 00 00]. Con todos los comandos definidos, llega el momento de seguir los pasos que se observan en la figura 33. El total de Bytes a recibir es de 19200, ya que las dimensiones de la imagen es $160 \times 120 = 19200$ pixeles. Una vez almacenados todos estos datos, es posible visualizar la imagen obtenida (Figura 32).

6.3 Desarrollo del algoritmo OCR en MATLAB

El desarrollo del algoritmo OCR es uno de los procesos de más duración de dentro de la solución de este trabajo. Éste se probó inicialmente utilizando imágenes capturadas una cámara Web Genius Messenger 310 en la resolución de 640x480 pixeles.

El proceso que sigue el algoritmo comienza con la captura y lectura de imágenes desde MATLAB. Luego éstas son recortadas dejando sólo las regiones de interés. Después se convierte la nueva imagen a blanco y negro, se compara con una

imagen e referencia previamente creada (plantilla), y finalmente se haya el porcentaje de similitud entra ambas imágenes para determinar cuáles son los dígitos que presentes en el contador. Este proceso se ve más claramente con el diagrama de flujo (Figura 33) y la explicación de cada paso que sigue a continuación.

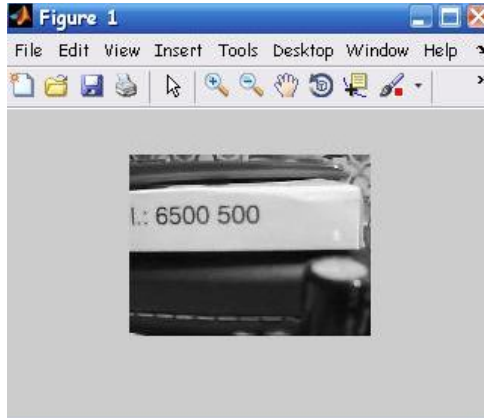


Figura 32. Una imagen obtenida con el módulo CMOS en MATLAB

6.3.1 Captura de imágenes con cámara web

La primera parte de la solución del algoritmo es la captura de diferentes tipos de imágenes desde una cámara web para definir la mejor posición a la que debe colocarse la misma, para realizar esto fue posible utilizar el soporte del módulo CMOS. Las imágenes se obtuvieron con ayuda del software que entrega el fabricante de la cámara (Genius Messenger 310 v2.01), de las cuales algunas capturas se muestran en la Figura 34.

6.3.2 Lectura de imágenes en MATLAB.

Una vez que se obtiene las imagen del contador, el siguiente paso es llevar a cabo la lectura de ésta en MATLAB. Para esto es necesario utilizar el comando *imread*. Por ejemplo si se requiere leer en una imagen de 640x480 pixeles a color, el comando *imread* arroja una matriz de 640x480 posiciones de tres planos diferentes, uno para cada componente de color (rojo, verde y azul). Como la iluminación del contador es con luz blanca, y ésta misma una combinación de todos los colores, solo fue necesario manipular solo uno de los planos de la matriz.

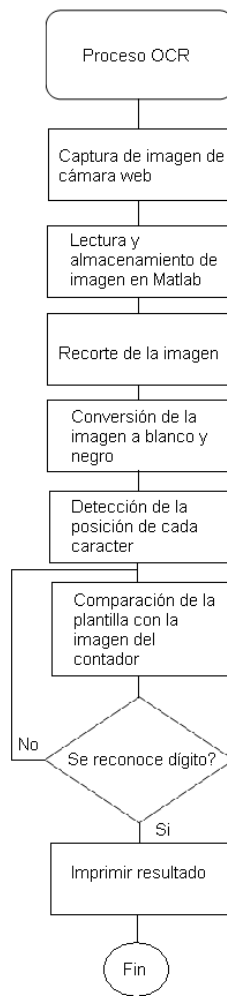


Figura 33. Diagrama de flujo del proceso de OCR en MATLAB



Figura 34. Imágenes obtenidas con cámara web

6.3.3 Recorte de la imagen.

La imagen leída posee regiones en donde no hay dígitos a reconocer. Dichas regiones son eliminadas, con una rutina que recorta éstas áreas lo más cerca posible a los dígitos. Un ejemplo de una imagen recortada se puede apreciar en la Figura 35.

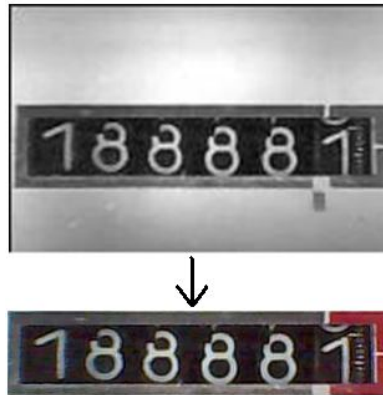


Figura 35. Imagen antes y después del proceso de recortado

6.3.4 Conversión de la imagen a blanco y negro (Binarización)

Con la imagen recortada el siguiente paso es convertir ésta última a valores que representen los colores blanco y negro, los cuales son '1' y '0' respectivamente. Esta conversión se realiza comparando el valor de cada pixel de la matriz con una referencia denominada 'pixel promedio' (que no es más que el valor medio entre los pixeles con la luminancia máxima y mínima de toda la imagen), Si es menor que el dicho promedio, se le asigna '0', de lo contrario se le asigna un '255'. Un ejemplo de dicha conversión se muestra en la Figura 36.



Figura 36. Imagen convertida a blanco y negro.

6.3.5 Ubicación de caracteres

El siguiente proceso que se realiza corresponde a la extracción de cada uno de los caracteres que contiene la imagen con el propósito de agilizar la comparación entre la imagen y la plantilla.

La rutina recorre la imagen verticalmente de arriba hacia abajo y de derecha a izquierda comprobando que exista una columna continua de píxeles negros. Luego busca donde hay otra línea similar, estableciendo la distancia entre líneas como '*un posible dígito*'. Para todos los caracteres se desarrolla el mismo proceso hasta tener almacenado en un vector las posiciones de inicio y fin de cada uno de ellos. Un ejemplo del resultado puede ser apreciado en la figura 37.



Figura 37. Visualización de los caracteres separados según análisis de su ubicación

6.3.6 Creación de Plantillas

Para poder identificar un dígito, es necesario tener una imagen de referencia de cada uno de estos dígitos, a esto se le denomina plantilla. Fue necesario crear la plantilla a partir de imágenes reales del contador, (esto con el fin de crear caracteres lo más parecidos posible). El resultado es una imagen (figura 38) que contiene todos los caracteres a reconocer orientados de forma vertical, debido a que precisamente éste es el movimiento que realizan estos símbolos en el contador.

6.3.7 Comparación con plantilla

Esta se realiza desplazando la plantilla verticalmente y horizontalmente sobre la imagen del contador, determinando si el porcentaje de similitud entre ambas en cada posición posible, era superior a un porcentaje de referencia. La figura 39 ilustra como ocurre este proceso.

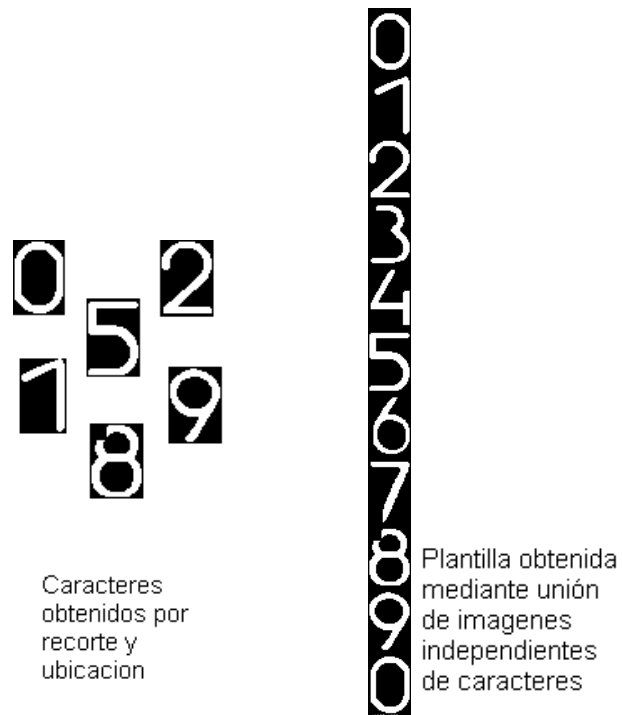


Figura 38. Plantilla creada mediante unión de otras imágenes

6.3.8 Análisis especial del último dígito

En el espacio correspondiente al último dígito del contador, se puede hallar dos números al mismo tiempo tal como se muestra en la figura 40.



Figura 40. Cambio de numero en el último dígito del contador

Cuando esto ocurre, se realiza el mismo proceso de comparación, solo que al momento de decidir cuál fue el carácter encontrado, se tiene en cuenta el rango de coordenadas exacto en la plantilla y con base en esto se determina el resultado. Un ejemplo de lo anterior se puede observar en la figura 41.

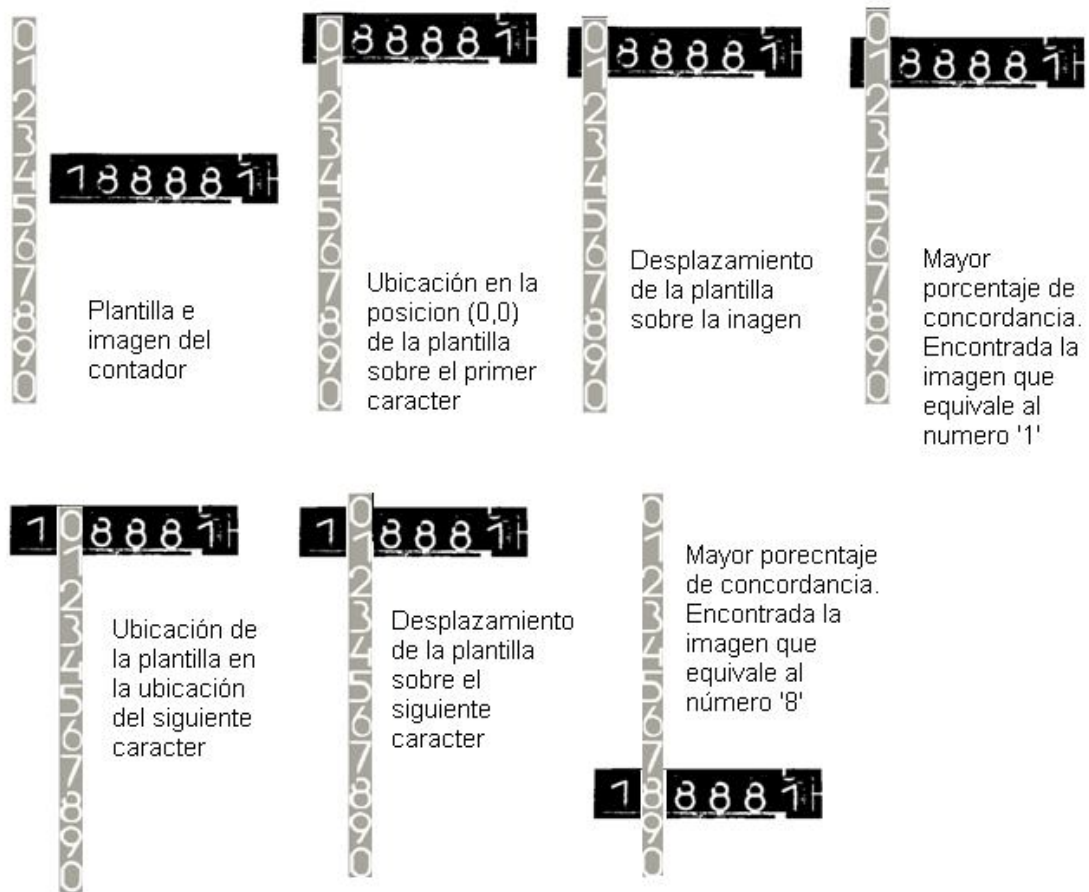


Figura 39. Proceso de comparación de imagen del contador y plantilla

6.3.9 Transmisión y recepción de datos mediante el puerto serial en MATLAB

Una vez configurado el puerto serial, es posible enviar y recibir datos hacia cualquier dispositivo serial (que para este caso es el DSP), con la ayuda de las instrucciones “fwrite” y “fread” respectivamente.

6.3.10 Transmisión y recepción de datos seriales en el DSP

Para el diseño de los algoritmos en el DSP, se utilizaron dos programas diferentes que son: MPLAB IDE v8.1 el cual permite crear y compilar rutinas en lenguaje y PICkit 2 V2.52 (junto con un programador) el cual permite ‘grabar’ en el DSP el programa creado. La apariencia básica de MPLAB IDE v8.1 se pueden apreciar en la figura 42.

Para llevar a cabo las rutinas de envío y recepción de datos del DSP en MPLAB, se debe configurar primero el módulo UART del DSP, y luego de tener configuradas la recepción como la transmisión de datos en el DSP se realiza la prueba de comunicación completa con MATLAB, donde es necesario realizar un programa que envía un dato hacia el DSP y este se encarga de guardarlo y devolverlo nuevamente hacia MATLAB. Para ello se requiere que tanto MATLAB como el DSP este configurados con el mismo Baudrate.

6.4 Sintetizar el algoritmo de OCR en el DSP

Para que el algoritmo de OCR funcione, es necesario tener una plantilla disponible, éstas se crean en MATLAB, a partir de imágenes reales del contador capturadas con el módulo C328R. La plantilla final se muestra en la Figura 43.

Luego de crear y almacenar la plantilla en la EEPROM interna del DSP, se simplificaron cada una de las rutinas elaboradas en MATLAB relacionadas al algoritmo de OCR, el cual es explicado en la figura 44.

El primer paso a seguir en el proceso, consiste en la recepción de la imagen desde el módulo CMOS, imagen que proviene en escala de grises de 8 bits. A medida que los datos de la imagen son recibidos en el DSP, este último debe realizar dos procesos simultáneos que son la conversión a blanco y negro de los datos y la compresión de los mismos en relación 8:1.

6.4.1 Almacenamiento de la imagen al DSP, binarización y recortado

Una imagen capturada real en escala de grises posee 19200 pixeles de un Byte cada uno (resolución de la imagen: 120x160), pero debido a la limitada capacidad de almacenamiento del DSP, no es posible guardarla en su totalidad. Las imágenes manejan 8 bits de datos, pero al binarizarla solo se necesita un bit para representar cada pixel de éstas, por lo que cada 8 pixeles de la imagen pueden ser almacenados en un solo byte de datos.

Sabiendo esto fue posible implementar una conversión y compresión simultánea de los datos entrantes para así reducir el espacio en memoria necesario para almacenar la imagen capturada, pero aún así eran necesarios 2400 Bytes para lograr dicho proceso. Como solo una región de la imagen es la que contiene los caracteres a reconocer, sólo dicha región es almacenada, la cual tiene 28x160 pixeles. Con esto finalmente es posible reducir los 19.2KBytes iniciales a solo 560

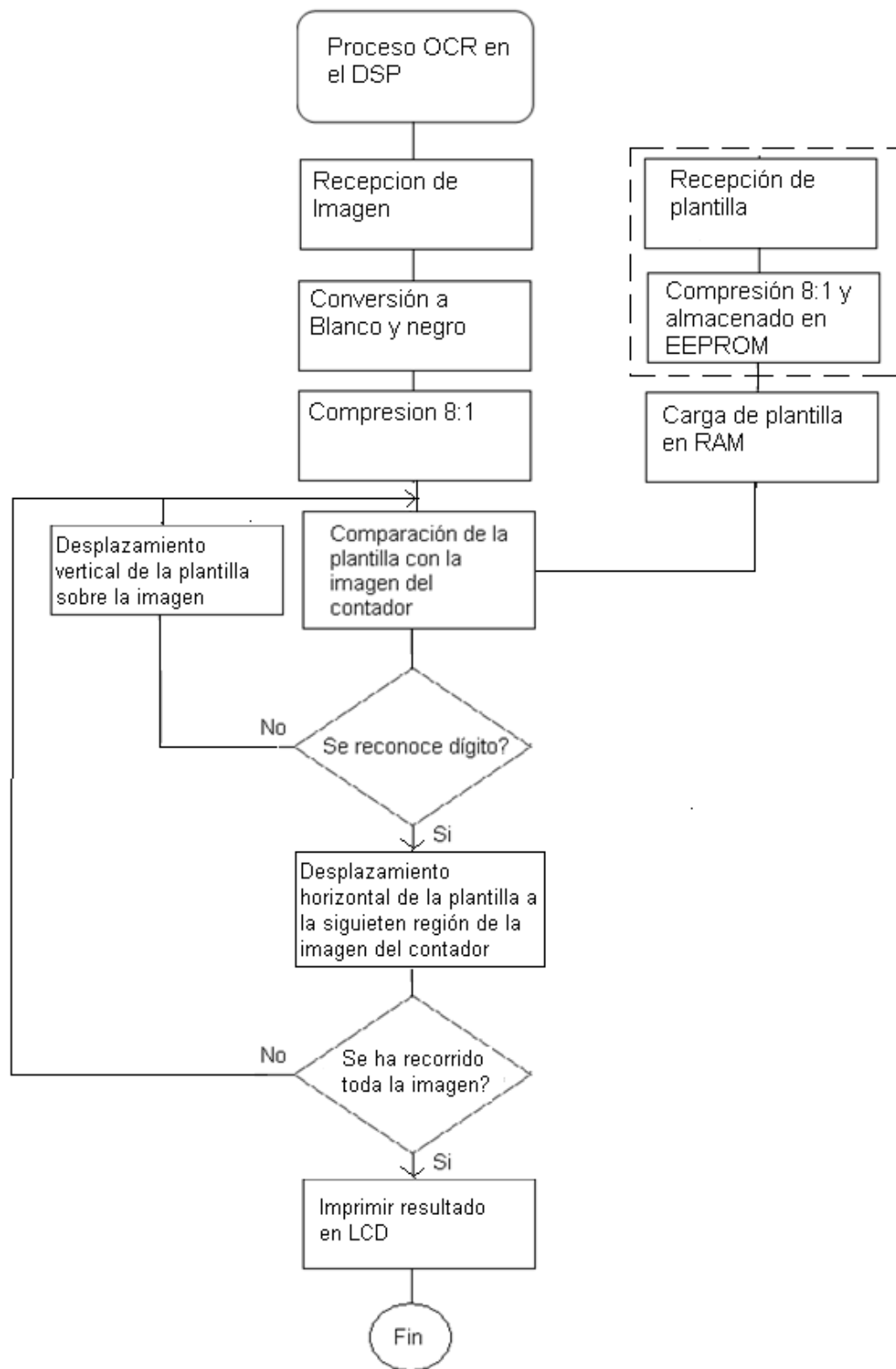


Figura 44. Diagrama de flujo del proceso de OCR en el DSP

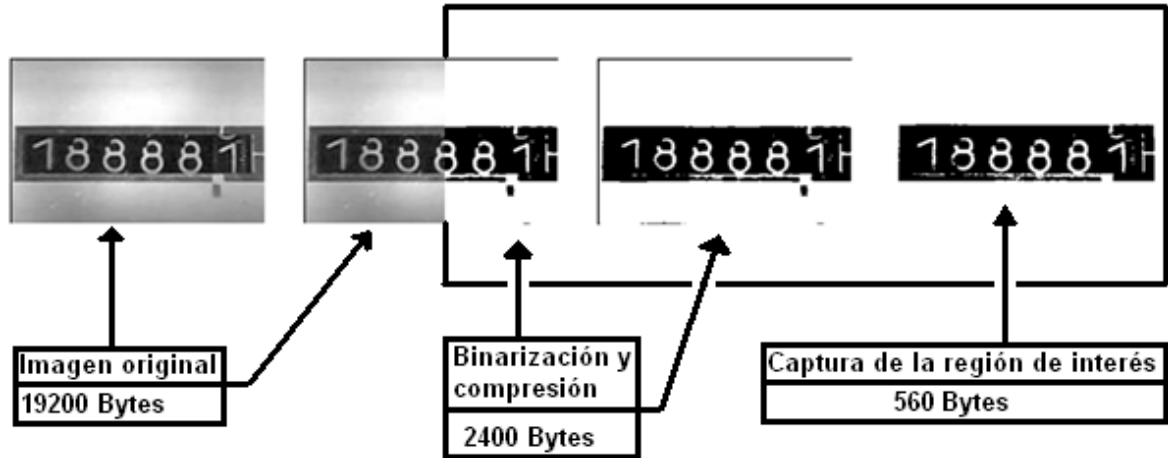


Figura 45. Conversión de la imagen a blanco y negro y recortado de la misma para compresión de datos (8:1)

6.5 Comunicación entre el DSP y el módulo CMOS C328R

Para establecer la comunicación entre el DSP y el módulo CMOS es necesario seguir la misma secuencia que se observa en las figuras 31 y 32, adaptando los comandos e instrucciones que se realizaron en MATLAB a lenguaje que entienda el DSP. El proceso a seguir comprende desde la configuración de la UART del DSP hasta la visualización por medio de un led del estado de la cámara. Este proceso se puede apreciar en el diagrama de flujo de la figura 46.

6.5.1 Configuración del módulo UART en el DSP

La configuración del módulo UART en el DSP se hace de la misma forma realizada en el numeral 6.4.2 teniendo presente que la tasa de transferencia de datos debe estar en el rango de operación del módulo CMOS (Ver Anexo 1).

6.5.2 Configuración del módulo CMOS desde el DSP

Después de configurar el módulo UART del DSP se debe crear la rutina capaz de sincronizarse con el módulo CMOS semejante a la implementada en MATLAB con la pequeña diferencia de que hubo que generar un retardo aproximado de 3 milisegundos después del envío de cada comando para que la cámara pudiera dar respuesta a cualquier acción.

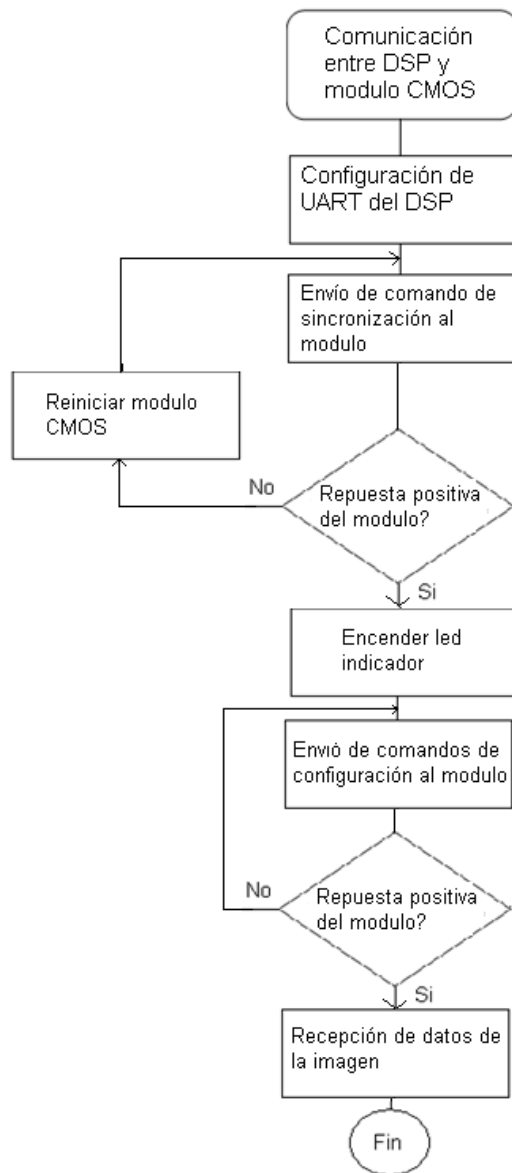


Figura 46. Diagrama de flujo de comunicación entre el DSP y el módulo CMOS C328R

6.6 Visualización del resultado del OCR en una interfaz externa (LCD)

Para visualizar el resultado final del OCR, el DSP utiliza una pantalla de cristal líquido (LCD). Ésta última está formada con matrices de puntos de cristal líquido, organizados en filas de varias posiciones cada una. Para el proyecto se usó la

LCD TC1602A con luz trasera, modelo que cuenta con 2 filas de 16 matrices cada una. Cada matriz tiene 7 puntos de cristal de alto, por 5 de ancho, junto con una línea de cursor de un píxel de alto por 5 píxeles de ancho (Ver figura 47), siendo así una resolución suficiente para visualizar cualquier carácter ASCII, pudiendo así mostrar un máximo de 32 símbolos a la vez.

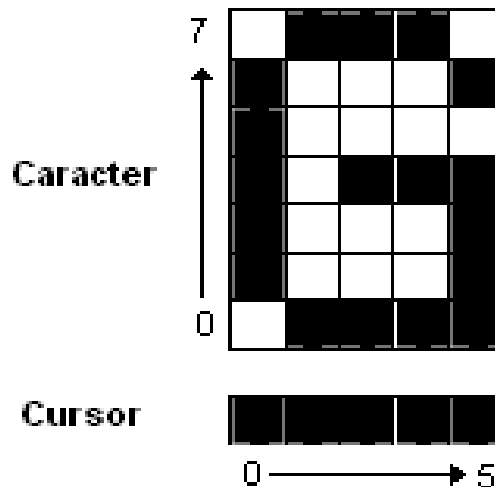


Figura 47. Carácter de una LCD [OPTR2000]

La utilización del módulo TC1602A, es sencilla, ya que solamente hay que configurarlo en el modo de visualización deseado, enviarle el código ASCII correspondiente al carácter que se deba mostrar y el módulo realiza el proceso faltante. El modo de funcionamiento de la LCD es simple: recibe datos de 4 u 8 bits de ancho, tiene 2 memorias internas; una para guardar los datos correspondientes a los símbolos que se van publicar (DDRAM) y otra para crear símbolos nuevos (CGRAM). En este caso solo se utilizó la memoria DDRAM con un bus de datos de 8 bits.

La LCD tiene 16 conectores que permiten realizar la adecuada alimentación, sincronización, visualización y transferencia de datos; mostrados brevemente en la Figura 48.

En donde 'DBX' son los pines correspondientes a los 8 bits del bus de datos, RS es la señal de control de comando/datos, 'R/W' corresponde a la línea de control de escritura y lectura de datos y 'E' es el pin de activación/desactivación del dispositivo. Además de 'Vcc' (que es la línea de alimentación de sistema), hay 3 pines de adicionales que son 'GND' que la 'tierra' del sistema, 'Vo' que es el voltaje de contraste del módulo y Vi que es el pin de alimentación de la luz trasera o 'Backlighth'.

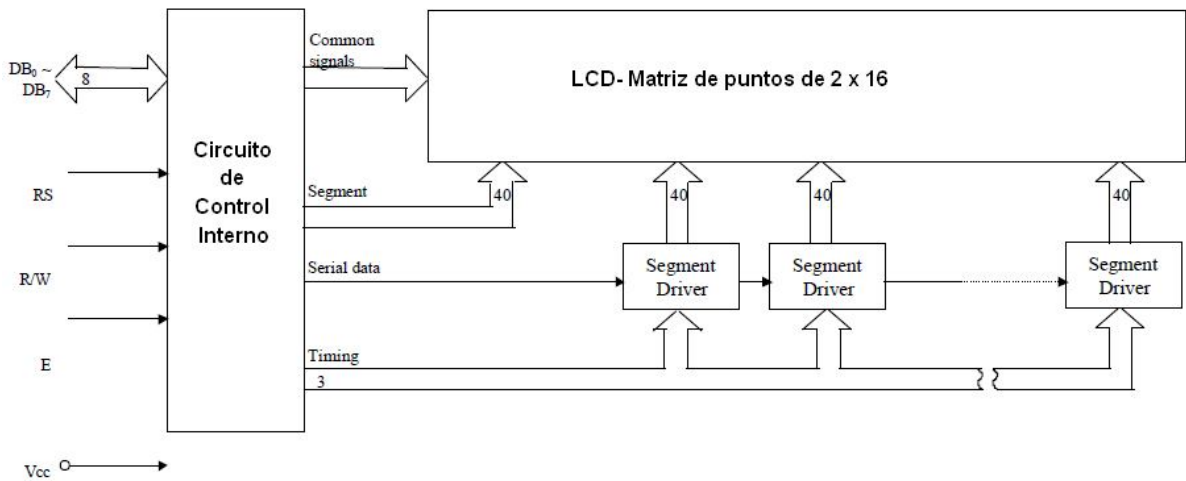


Figura 48. Diagrama de bloques de una LCD TC1602A [OPTC2000]

6.6.1 Rutina de inicialización y configuración de la LCD

Para empezar a publicar símbolos en la LCD, primero se enciende la LCD conectado la alimentación en el pin 'Vcc', luego, se debe esperar a que el proceso de inicialización interno del módulo termine, haciendo un retardo de al menos 15ms. Éste último proceso se repite 2 veces más haciendo retardos de 4.1ms y 100us de forma correspondiente.

Luego de los primeros 3 comandos, el módulo está listo para recibir la configuración por defecto con la que funcionará siempre y cuando no ocurra un reinicio del sistema (desconexión de fuente de alimentación). A continuación se envían los comandos necesarios que especifican el formato de los datos que va a recibir, la forma en que se van a mostrar, etc. Durante este proceso, es necesario determinar si la LCD aún está procesando el comando enviado anteriormente, esto es posible detectando la bandera de ocupado o "Busy Flag" antes de enviar una nueva orden. Todo lo anterior se explica de una mejor manera en el diagrama de flujo de 'inicialización' (Ver Figura 49) proporcionado por el fabricante. Los siguientes son los comandos usados en el proceso de inicialización [OPTO2000]:

6.6.1.1 "Clear Display"

Escribe el código ASCII de ('20' Hexadecimal) en todas las direcciones de memoria DDRAM. Esto ocasiona que el cursor reinicie su posición al borde superior izquierdo (línea 1, carácter 1). En la figura 49 se puede apreciar la estructura de este comando.

	RS	R/W	DB ₇	DB ₆	...	DB ₁	DB ₀
Code	0	0	0	0	0	0	1

Figura 49. Comando "Clear Display".

6.6.1.2 "Entry mode set"

Establece el modo de "Incremento/Decremento" en el sentido deseado:

I/D: Incrementa (I/D = 1) o decrementa (ID = 0) la dirección DDRAM en factor de cuando un carácter es escrito o leído, esto repercute en la posición del cursor en la pantalla, que efectivamente se desplaza en la dirección establecida anteriormente.

S: Desplaza la pantalla en la dirección determinada por I/D. Si S = 0; no desplaza el display, y si S = 1 Desplaza el display.

En la figura 50 se puede apreciar la estructura de este comando.

	RS	S/W	DB ₇	DB ₆	...	DB ₁	DB ₀
Code	0	0	0	0	0	1	I/D S

Figura 50. Comando "Entry Mode Set".

6.6.1.3 "Display ON/OFF Control"

Controla el estado de "encendido/apagado" de todo lo visualizado en la pantalla. Si D = 1, la pantalla es encendida y Si D = 0, la pantalla es apagada pero los datos escritos anteriormente en la DDRAM permanecen intactos. El cursor "C" es mostrado si C = 1 y no aparece cuando C = 0. El titilar o "blinking" del cursor es controlado por el bit "B", si B = 1, el cursor titila, de lo contrario no lo hace. En la figura 51 se puede apreciar la estructura de este comando

	RS	R/W	DB ₇	DB ₆	...	DB ₁	DB ₀
Code	0	0	0	0	0	1	D C B

Figura 51. Comando "Display ON/OFF Control".

6.6.1.4 "Function Set"

Con este comando se establecen el ancho del bus de datos, la fuente de los caracteres y el número de líneas que posee el display. En la figura 53 se puede apreciar la estructura de este comando.

INICIALIZACION

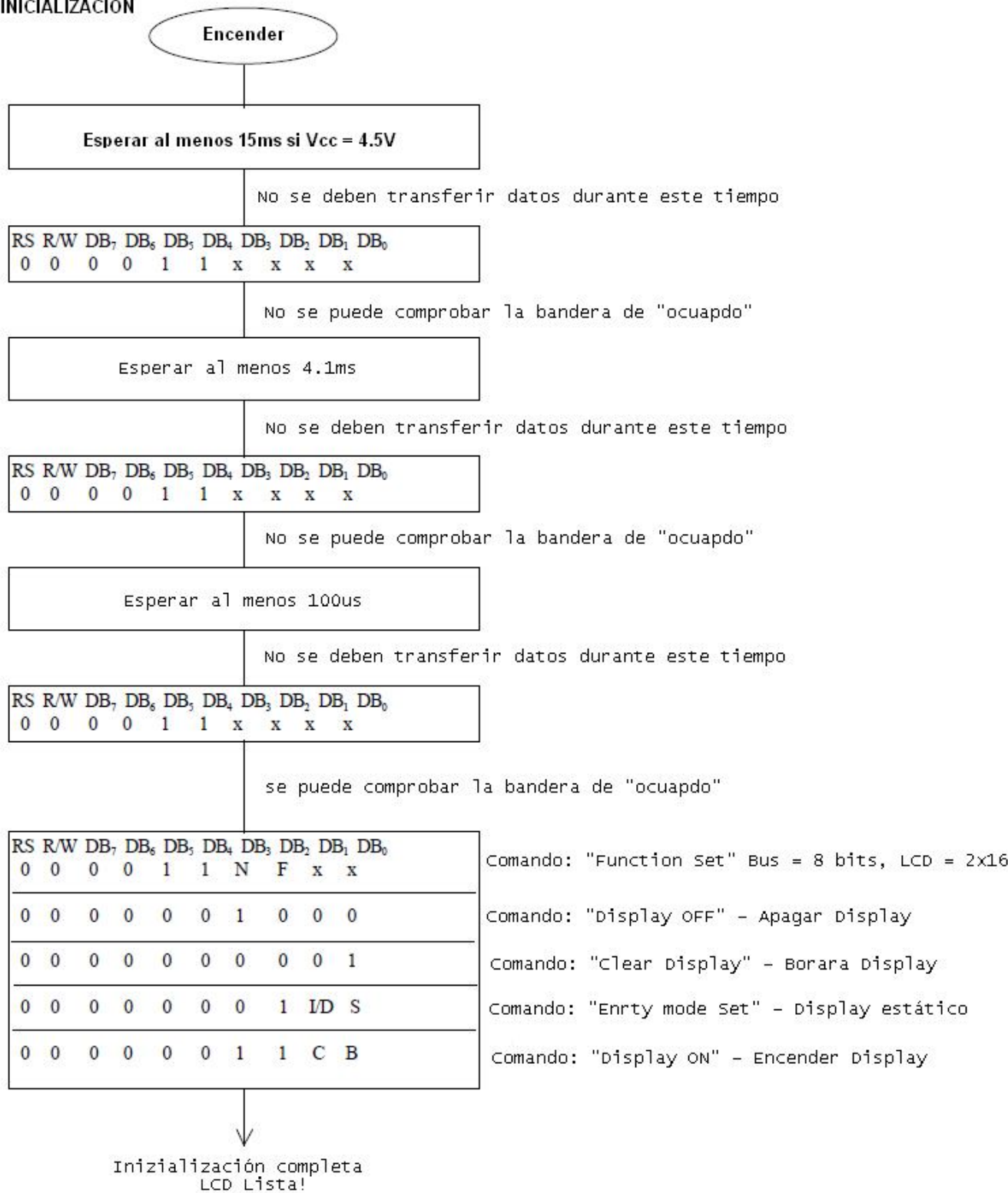


Figura 52. Rutina de configuración de la LCD [OPTP2000]

DL: Ancho del bus de datos. Si DL = “1” entonces el bus es de 8 bits y si D = 0 el bus es de 4 bits y las líneas D. Si se usa el modo de datos de 8 bits, cada proceso de escritura debe hacerse 2 veces seguidas, debido a que internamente la LCD lee un dato de 8 bits.

N: Número de líneas del Display

Si N = “0”, entonces el módulo tiene una pantalla que solo visualiza 16 caracteres (1 línea).

Si N = “1”, entonces el módulo tiene una visualiza 32 caracteres (2 líneas).

F: Fuente de los caracteres.

Si F = “1”, indica que los caracteres tienen dimensiones de 5 x 10 píxeles.

Si F = “0”, indica que los caracteres tienen dimensiones de 5 x 7 píxeles.

	RS	R/W	DB ₇	DB ₆	...	DB ₁	DB ₀			
Code	0	0	0	0	1	DL	N	F	x	x

Figura 53. Comando “Entry Mode Set”. Nota: X = no importa.

6.6.2 Rutinas de escritura en la LCD

Para ordenarle al módulo LCD que “muestre” un carácter, o que “cambie” algún parámetro del display, se usa la rutina de escritura, ésta funciona solamente si previamente el módulo ha sido inicializado.

Después de inicializar, el comando “Entry mode set” es enviado, luego se inicia la escritura (“Write data to XXRAM”) de cada carácter de forma secuencial. El display muestra el carácter recibido en la posición un tiempo después y automáticamente apunta a la siguiente posición en espera de que un nuevo carácter sea escrito. En la figura 55 se muestra esquema del proceso de escritura en la LCD:

El comando que “escribe” en la LCD es **“Write Data to CG or DD RAM”**, este escribe datos de 8 bits en la memoria DDRAM o CGRAM del módulo. Cada escritura incrementa automáticamente la dirección de escritura actual en una posición. En la figura 54 se puede apreciar la estructura de este comando.

	RS	R/W	DB ₇	DB ₆	...	DB ₁	DB ₀		
Code	1	0	D	D	D	D	D	D	D
	← Bits de mayor peso			Bits de menor peso →					

Figura 54. Comando “Write Data to CG or DD RAM”.

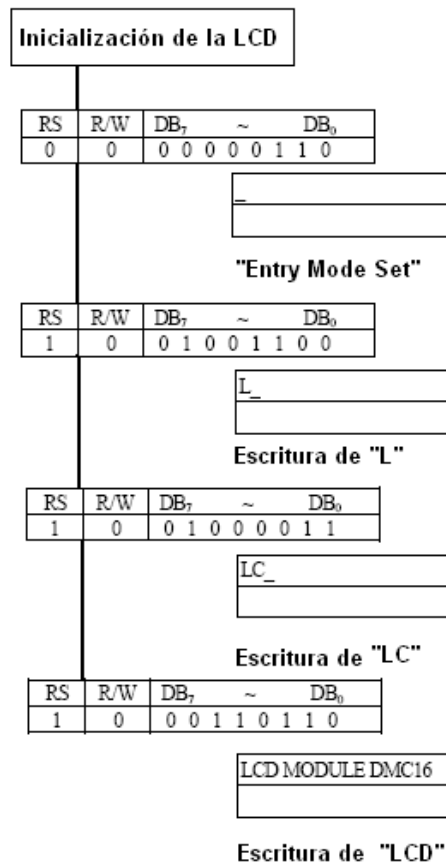


Figura 55. Diagrama de flujo de la rutina de 'escritura' [OPTD2000]

7. PRUEBAS Y RESULTADOS

7.1 Captura de imágenes del módulo CMOS

Entre las pruebas realizadas, las de mayor importancia tienen que ver con la comunicación tanto de la cámara como del DSP con MATLAB, ya que así fue posible conocer la forma correcta del funcionamiento y comunicación de cada dispositivo. Adicionalmente ha sido utilizada la herramienta *Hyperterminal* de *Windows XP SP2* con el fin de verificar la información que se transmiten entre los dispositivos.

La captura de imágenes del módulo implica la realización de bastantes tomas por el hecho de que no se conocía el orden en que se envían los datos desde la cámara. En una prueba se almacenaron todos los datos de la imagen obtenida en una matriz y manipulándola se pudo determinar dicho orden. La cámara envía la imagen de derecha a izquierda y de abajo hacia arriba tal como se observa en la figura 56. En estas mismas pruebas, la cámara se bloqueó en determinadas ocasiones y no arrojó ningún dato; este hecho ocurría después de que se intentaban tomar entre 10 y 13 imágenes consecutivas. La única solución que se halló para de evitar este problema es reiniciar el módulo después de cada captura de imagen.

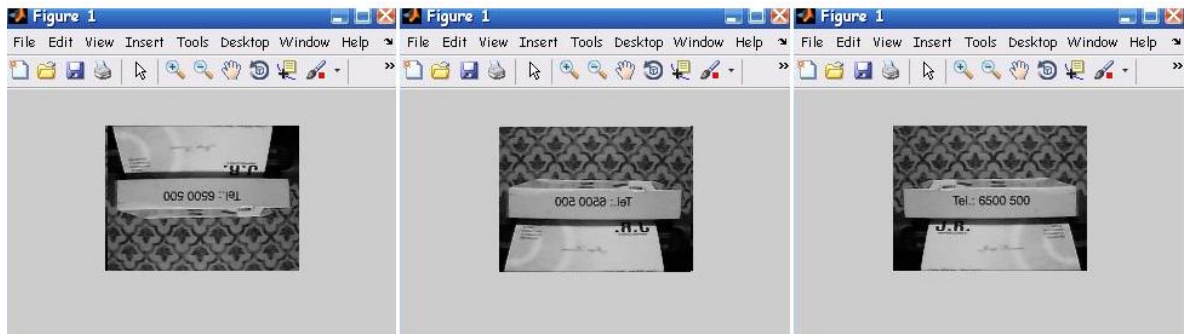


Figura 56. Proceso de organización de datos obtenidos por el módulo CMOS

Adicional a lo anterior, fue necesario hacer un ajuste minucioso del enfoque de la cámara para garantizar que la imagen obtenida fuese lo más nítida posible, el resultado de esta calibración se muestra en la figura 57:

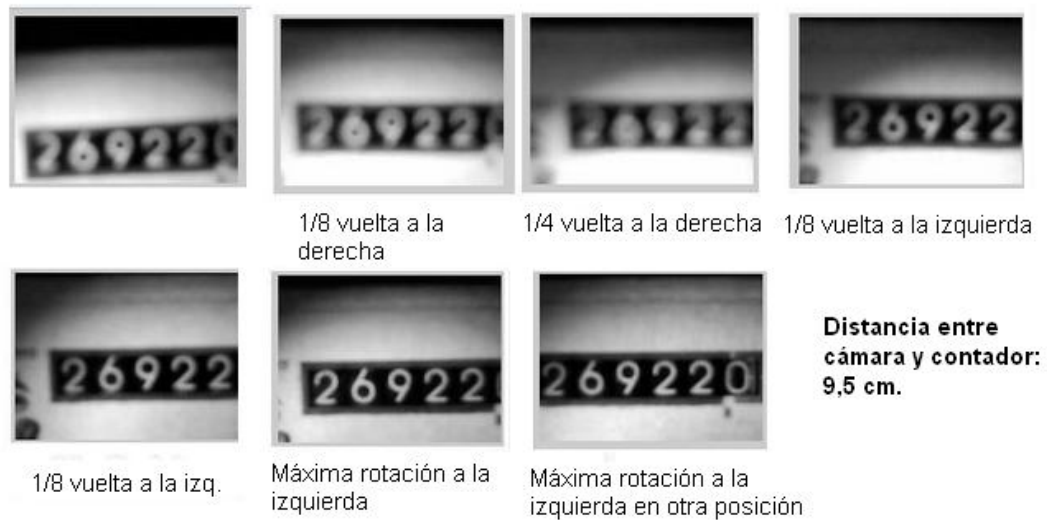


Figura 57. Capturas en resolución 120x160 variando posición, condiciones de iluminación y giro el lente de la cámara para mejor enfoque.

7.2 Conversión de imágenes en escala de grises a blanco y negro

La conversión de imágenes a blanco y negro se analizó en dos escenarios diferentes: en MATLAB y en el DSP. Para el primero, las pruebas consistieron en modificar moderadamente el valor de luminancia promedio hasta que la imagen convertida fuese lo suficientemente clara y legible para el DSP, tal como se puede apreciar en la figura 58.

Con base en estos resultados, fue posible determinar un valor preliminar de luminancia con el que se debía binarizar la imagen, el cual fue de 60.

En el DSP, se realizó el mismo procedimiento, pero a diferencia del entorno MATLAB, con el DSP no era fácil saber si la imagen estaba siendo convertida correctamente. Entonces para comprobar esto, se utilizó el mismo programa que configura la cámara, pero evaluando y adaptando la parte de recepción y almacenamiento de la imagen para que el DSP pudiese transmitir la imagen. El resultado obtenido se puede observar en la figura 59.

Los resultados de esta prueba permitieron comprobar que tanto la conversión a blanco y negro como el almacenamiento de la imagen dentro del DSP se desarrollaban de forma satisfactoria.

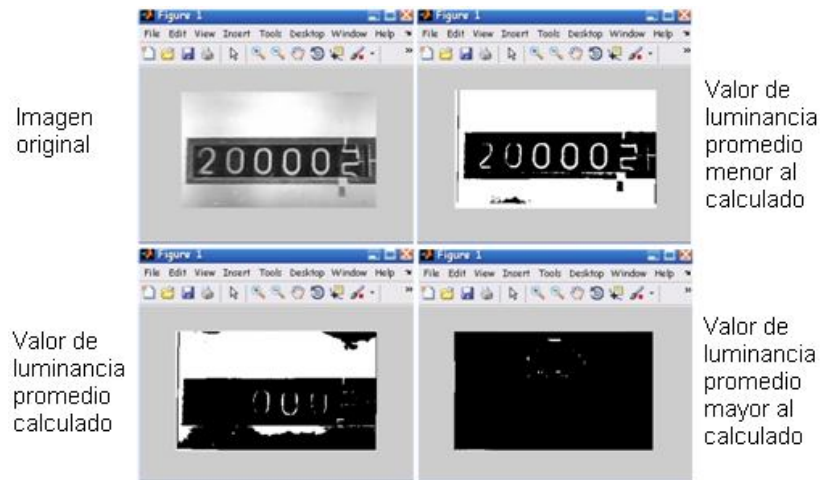


Figura 58. Imagen original e imágenes en blanco y negro con variación del promedio de luminancia

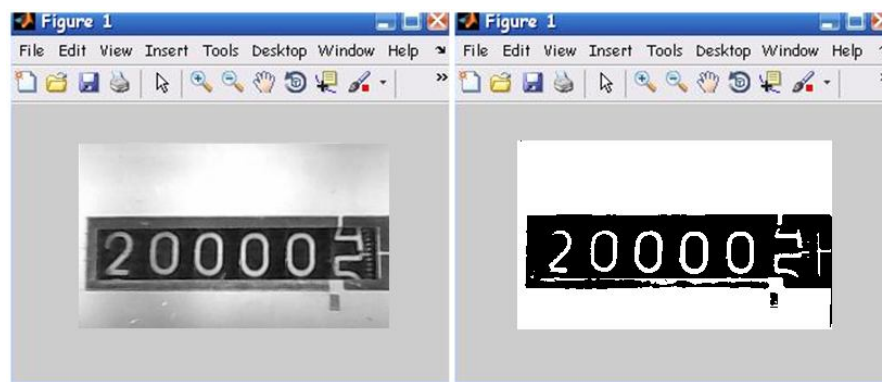


Figura 59. Imagen original e imágenes en blanco y negro en el DSP

7.3 Calibración de los valores de referencia

El algoritmo OCR tiene dos valores de referencia cruciales: el porcentaje de comparación y la luminancia promedio. El primero se usa para determinar si un carácter está presente en la imagen y el segundo para realizar la binarización de ésta última. Durante las pruebas del algoritmo OCR se presentó un inconveniente importante: se descubrió una relación entre ambos valores de referencia, es decir, que con un valor muy bajo o muy alto de luminancia, no era posible identificar dígitos debido a que ninguna comparación igualaba (o superaba) el porcentaje de comparación.

Para encontrar una solución a éste problema, se realizaron 50 lecturas con 4 valores distintos de luminancia, y se tabularon los resultados (Ver anexo 4). Esto

con el objetivo de determinar los valores óptimos de ambos parámetros que permitiesen diferenciar claramente los dígitos de la imagen. En la figura 55 se puede observar una representación grafica de dichas pruebas.

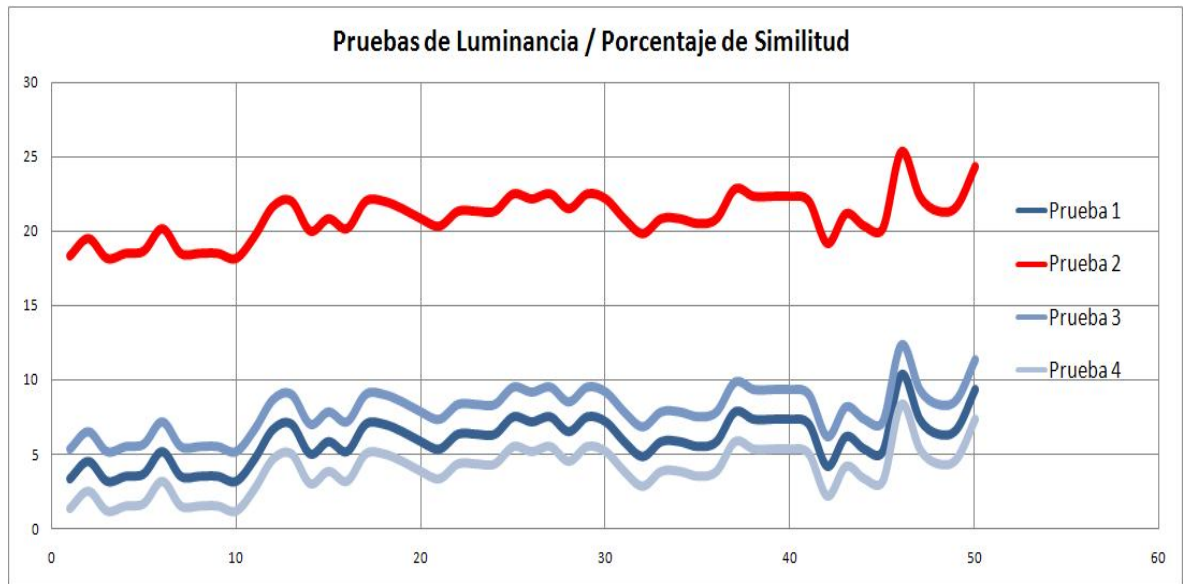


Figura 60: Graficas que representa las pruebas de calibración realizadas.

En la gráfica anterior se observan 4 curvas, cada una representa una prueba de 50 lecturas realizadas. En cada una se estimaron los porcentajes de similitud de cada dígito hallado y se le halló la diferencia con el porcentaje de 'no' similitud (entiéndase este último como el valor de comparación hallado cuando no se identifican dígitos) Las pruebas 1, 2, 3, y 4 se hicieron con 25, 45, 65 y 127 como valores de luminancia de referencia.

Como se puede observar, en la prueba 2 hay un mayor diferencial que en las otras curvas, esto quiere decir que los dígitos contenidos en una imagen capturada con esta luminancia son mucho más claros que con otros valores, en consecuencia éstos son más fáciles de identificar y el resultado del algoritmo con dichos parámetros es más confiable.

En las tablas que respaldan a la figura 60, se observó que todas las coincidencias de los primeros 5 dígitos encontrados superaban el 84% de similitud; para el sexto carácter éste valor no tuvo una tendencia clara. Finalmente, el valor de luminancia promedio más apropiado para obtener resultados confiables es 45, porque las pruebas realizadas revelaron que con éste parámetro los posibles caracteres en una imagen son totalmente reconocibles.

7.4 Resultados del algoritmo de OCR en el DSP

Tras varias pruebas del prototipo funcionando completamente, se obtuvieron lecturas del contador, en las cuales, en consecuentes ocasiones se presentaron errores, más precisamente en el análisis del último dígito. Esto se debe a que en dicha posición correspondiente hay líneas adicionales que interfieren al momento de hallar el porcentaje de similitud, reduciéndolo considerablemente. En consecuencia el resultado mostrado en la LCD solía ser erróneo. A continuación, en la figura 61 se muestran capturas reales de la imagen original del contador y sus respectivas lectura.

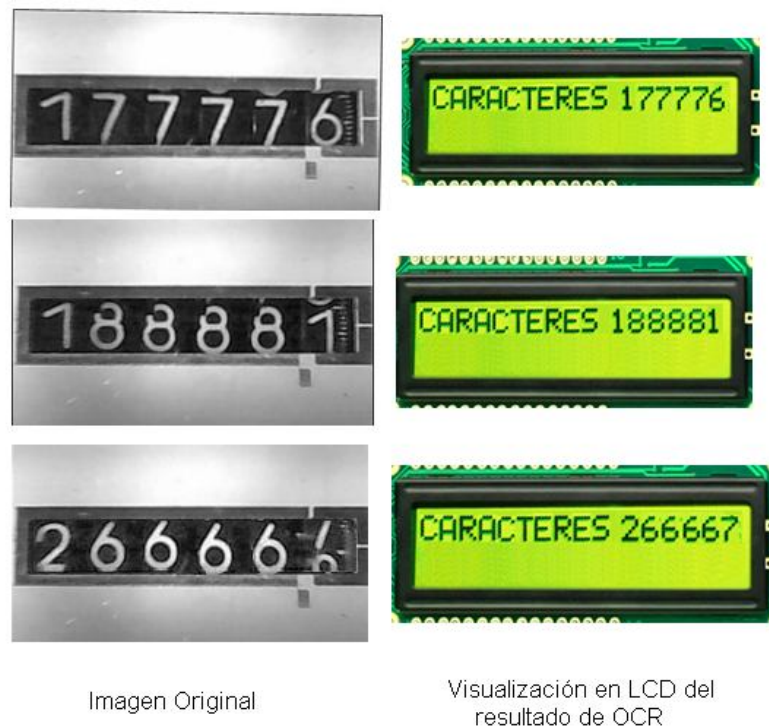


Figura 61. Resultado del proceso de OCR.

Sabiendo cual es la causa del error, se hicieron modificaciones leves en el algoritmo OCR, con el objetivo de darle un tratamiento especial a ésta área de la imagen. Estos cambios consistieron en eliminar la condición de comparación con el porcentaje de referencia cuando se estuviese reconociendo el último dígito. Ésta medida redujo la posibilidad de error notablemente a excepción del caso en donde el número 8 y 9 están presentes debido a que el algoritmo no siempre reconoce el dígito correcto, porque ambos dígitos forman dos círculos consecutivos que, para el DSP, concuerda con casi todos los caracteres de la plantilla.

Sin embargo, fue necesario realizar un sondeo más profundo de cuan eficiente era el algoritmo ya habiéndole hecho todas las calibraciones finales. Se hicieron 50 lecturas de las cuales 4 resultaron erróneas (En el anexo 5, se muestran todos los resultados de dichas pruebas), esto equivale al 8% total de las pruebas realizadas y si se tiene en cuenta que de 6 dígitos, solo uno será incorrecto, se puede concluir que la probabilidad de obtener una lectura errónea es del 1.3% y sólo se centra en el caso específico ya mencionado (transición entre el 8 y el 9). La figura 62 muestra la distribución gráfica de lo anterior.

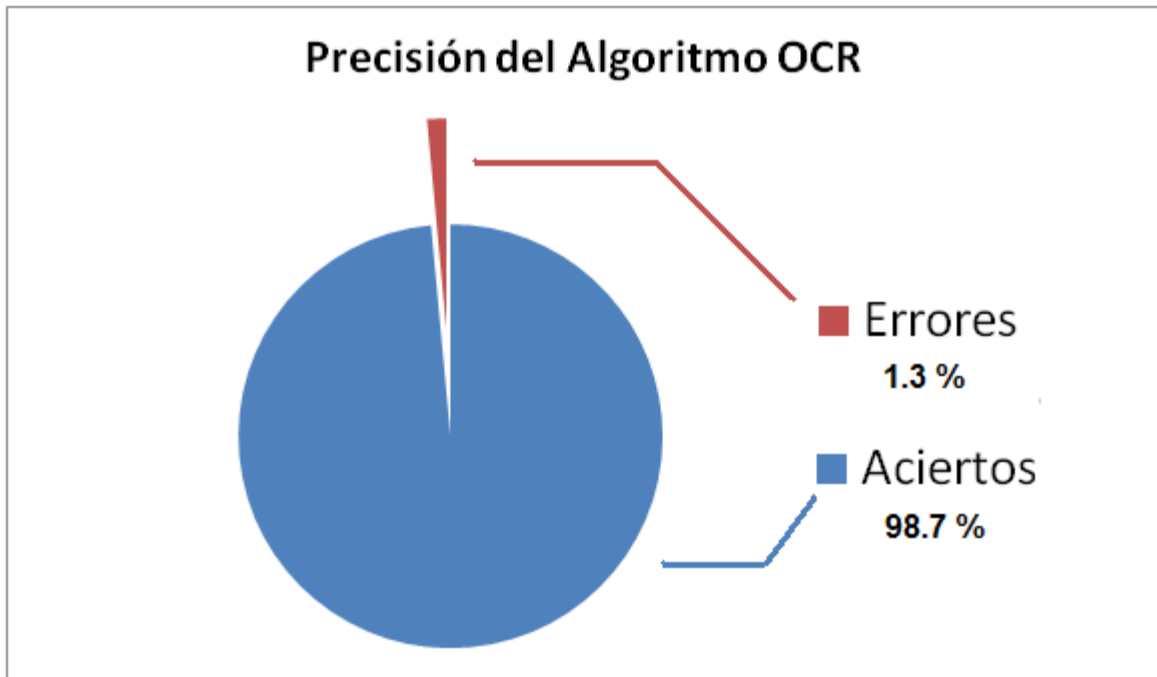


Figura 62. Porcentajes de error en el reconocimiento de caracteres.

8. CONCLUSIONES

En general, el algoritmo de OCR desarrollado para el dsPIC resultó ser muy rígido en algunos aspectos, y en otros, muy flexible. La eficiencia del algoritmo es regular, debido a que al ejecutar el código, le toma mucho tiempo al DSP llegar a un resultado coherente haciéndolo “lento”, y solo funciona en condiciones con luminosidad controlada, con un tamaño de imagen predeterminado, y fuente de caracteres fijo.

En cuanto a la eficacia y viéndolo desde otro punto de vista, las mismas desventajas anteriores vuelven flexible a dicho algoritmo, porque al ser un código de comparación simple, se puede implementar en un microprocesador con prestaciones medias, sin la necesidad de una memoria externa para almacenar imágenes, haciendo posible la idea de lo que se puede decir sería un “OCR on a Chip”.

Teniendo un tal dispositivo, “lento” pero “seguro”, sería posible hacer lecturas no solo de contadores eléctricos, sino de dispositivos que usen interfaces gráficas mecánicas (como lo son válvulas y medidores de paso) en los que una medición con imágenes simple, barata y automatizada es la única una solución considerable.

En el principio del proceso de desarrollo, la cantidad de memoria necesaria para almacenar imágenes en escala de grises en el entorno MATLAB fuera demasiado alta desde el punto de vista del DSP. Esto ocasionaba que, al elegir el circuito integrado DSP adecuado para realizar la captura y procesamiento de imágenes, se incrementase el costo del mismo, y solo por la memoria de datos que éste debía poseer. Al implementar la sintetización del algoritmo OCR, se observó que al binarizar (convertir a blanco y negro) las imágenes a procesar, no era necesario destinar un Byte por cada píxel de éstas (cada bit representa un píxel); es por eso que se diseñó una técnica de compresión sencilla, la cual empaqueta en un Byte 8 píxeles, y en consecuencia “comprime” el costo del DSP.

Para hacer OCR es necesario hacer que el procesador “vea”, esta cualidad puede resultar una herramienta útil en otros sistemas que procesan imágenes, ya que al hacer OCR, en el DSP todo que se captura es convertido/comprimido y se puede transmitir a otros dispositivos logrando un ahorro de recursos (tiempo, bus de datos, memoria) considerable para éstos últimos.

Al final del desarrollo del proyecto, fue posible reducir la rutina de binarización/compresión de datos desde la cámara CMOS, de tal forma que podía ser ejecutada durante la recepción de dichos datos (proceso durante el cual el DSP debía ocuparse solamente de ésta tarea). A partir de esto, y basándose en la

linealidad en la ejecución del algoritmo, se puede decir que el tiempo de obtención de resultados se puede reducir considerablemente aprovechando los tiempos “muertos” (de no procesamiento de datos) para realizar otras tareas.

Generalmente para obtener y procesar imágenes desde dispositivos externos, es necesario dedicar bastantes recursos para lograr la sincronización de los módulos de captura. El módulo CMOS C328R resultó ser bastante flexible (más de lo esperado) al momento de capturar imágenes, ya que la interfaz simple que posee y el diseño compacto, permiten adaptarlo casi a cualquier sistema que requiera la obtención rápida de imágenes de resolución aceptable utilizando pocos recursos.

Al tratar de reducir aún más el tiempo de ejecución del programa, el compilador de Microchip Inc. MPLAB C30 (Student Version) pudo reducir de forma relevante el consumo de memoria de programa necesario, y a su vez el tiempo de ejecución del algoritmo OCR. Esto sin mencionar el ambiente bastante “amigable” en el momento de diseñar las rutinas de configuración comunicación y sobre todo, las rutinas de procesamiento, debido a que con el entorno que ofrece este compilador, se pueden implementar matrices de datos, que permiten organizar de una manera más “intuitiva” las imágenes recibidas.

Durante las pruebas del proyecto, la cualidad en “espiral” de la metodología usada permitió hacer reajustes dinámicos durante el desarrollo sin rediseñar totalmente el sistema, haciendo que las etapas anteriores a las más recientes fueran compatibles sin hacer mayores cambios. De esta manera, se pudieron superar inconvenientes que a primera vista parecían insalvables porque afectan las fases del proyecto que estaban dadas por finalizadas.

En definitiva, el algoritmo OCR implementado en este proyecto, (que fue diseñado solo para contadores eléctricos) resulta ser adecuado para realizar reconocimiento de imágenes en ambientes controlados, en los cuales el tiempo de obtención de resultados nos sea relevante, pero la fiabilidad de éstos si sea de gran importancia.

9. RECOMENDACIONES

Al momento de implementar comunicaciones seriales bajo el estándar RS-232 que deban interactuar con dispositivos CMOS o TTL, es recomendable la utilización de circuitos cambiadores de nivel con transistores u otros métodos y no el integrado MAX-232, debido a que este último presenta gran cantidad de inconvenientes de ruido en la transmisión de los datos, generando errores inesperados.

Durante el desarrollo de éste tipo de proyectos, en los que se encestan obtener imágenes, es importante controlar las condiciones de luz del objetivo, debido a que facilita el análisis de los datos capturados en distintas ocasiones.

Es importante anotar, que para verificar las comunicaciones seriales, es de gran utilidad usar interfaces (como MATLAB o el Hyperterminal de Windows) que permitan observar (Snifing) y verificar los datos transmitidos. De esta forma es posible identificar posible errores de lógica en la comunicación. También puede ser de gran utilidad hacer “snifing” con LED’s indicadores que se marquen el estado actual de la ejecución de un algoritmo, para así detectar fallos que son difíciles de detectar con simuladores

10. TRABAJO FUTURO

Uno de las posibles mejoras a realizar en este proyecto es la incorporación de un algoritmo adicional en el DSP que se encargue de realizar la descompresión de imágenes en formato JPEG, permitiendo el uso de imágenes de mayor resolución y la utilización de una gama más amplia de módulos de cámaras.

Asimismo es posible implantar un sistema de envío de los datos resultantes hacia un dispositivo externo como lo puede ser un teléfono celular u otro sistema con comunicación inalámbrica para así poder extender el rango de uso del prototipo.

Al momento de sintetizar un algoritmo de OCR en un DSP, el método utilizado en este trabajo se basa en la comparación mediante Matriz de Concordancia, por lo tanto es posible aumentar la fidelidad de los datos procesados si se utiliza otro método de comparación como lo es la Extracción de Características teniendo en cuenta que el nivel de procesamiento requerido es mayor.

11. GLOSARIO

CÁMARA WEB: es una cámara pequeña digital, generalmente de baja resolución que es conectada a un computador va USB [WEBH2008].

CHARACTER: es cualquier signo tipográfico que puede ser letra, número, símbolo, signo de puntuación o espacio. [DEFI2008]

CIFRAR: es una operación en la que se convierte un texto claro a uno codificado [ZOOF2005].

CONTADOR DOMICILIARIO DE ENERGÍA ELÉCTRICA: Dispositivo mecánico encargado de realizar el conteo del consumo de un servicio público de energía eléctrica [ANSW2008].

DESCIFRAR: recuperar un texto normal que había sido cifrado mediante alguna clase de claves [ZOOF2005].

DSP: Digital Signal Processor o Procesador Digital de Señal, se encarga de procesar señales análogas (audio, voz, imagen, video) para tratarlas digitalmente [CASA2008].

ENCRIPCIÓN: es un proceso que como medida de seguridad para información delicada, la vuelve ilegible a través de claves o parámetros [ALEG2008].

FREESCALE: es un fabricante estadounidense de semiconductores que ofrece soluciones a problemas con la utilización de dispositivos como DSP's, microchips y procesadores. [FREE2004]

FUENTE DE VOLTAJE: dispositivo electrónico que crea una diferencia de potencial [ROBE2008]

GSM: es el sistema de telefonía móvil digital más utilizado en el mundo. [MASA2008]

JAVA: es un software donde los programas creados en él pueden ejecutarse sin cambios en diferentes tipos de arquitecturas y dispositivos computacionales. [WEBT2008]

MANOS LIBRES: es un dispositivo que permite realizar una comunicación a distancia ente un celular y el usuario. [YOTE2008]

MATLAB: es un entorno de computación y desarrollo de aplicaciones totalmente integrado y orientado para llevar a cabo proyectos en donde se encuentren implicados complejos cálculos matemáticos y la visualización gráfica de los mismos. [MONO2007]

MENSAJE DE TEXTO: es un servicio disponible en teléfonos móviles para enviar mensajes cortos entre teléfonos y dispositivos móviles. [DEFI2008]

OBTURADOR: es el dispositivo que controla el tiempo durante el cual llega la luz al elemento sensible de una cámara para tomar una fotografía. [GRAN2007]

OCR: Software encargado del reconocimiento óptico de caracteres provenientes de una imagen para que puedan ser editados como texto. [ORGA2008]

RUTINA: es un procedimiento o conjunto de código de un programa que son utilizados constantemente. [PORT2007]

SIM CARD: es una tarjeta que se utiliza en los teléfonos móviles en la que se almacena de forma segura la información del usuario del teléfono necesaria para identificarse en una red. [PHON2007]

TELÉFONO CELULAR: es un teléfono portátil sin hilos, conectado a una red celular que permite que el usuario lo utilice en cualquier lugar cubierto por la red. [PRTE2008]

USB 2.0: es una interfaz entre la computadora y ciertos dispositivos para envío de datos con velocidades de hasta 480 Mbps. [SHOP2008]

12. BIBLIOGRAFÌA

12.1 Referencias Bibliogràficas

[SHEL1995] Sheldon Tom. Encyclopedia LAN Times de Redes, McGraw-Hill. 1995.

12.2 Referencias de Internet

[ACWB2009] Universidad del Pais Vasco: Cableado, España 2008. www.sc.ehu.es/acwbecae/ikasgaiak/CDA/Lab2.html (Fecha de consulta: en Enero 15 de 2009)

[ALEG2008] Diccionario informàtico: definiciones a conceptos informàticos. Argentina, 2008. <http://www.alegsa.com.ar/Dic/encryptacion.php> (Fecha de consulta: en Septiembre 14 de 2008)

[ANSW2008] Free Online Encyclopedia: Electricity Meter. United States, 2008. <http://www.answers.com/topic/electricity-meter> (Fecha de consulta: en Septiembre 21 de 2008)

[ANSW2008] Portal de respuestas: dispositivos de mediciones de energìa elèctrica. United States, 2008. <http://www.answers.com/topic/electricity-meter> (Fecha de consulta: Septiembre 15 de 2008).

[ARIT2007] Aritmética en coma Flotante: Ampliación de Estructura de Computadores, Daniel Mozos. Marcos Sánchez-Elez, José Luis Risco Facultad de Informática http://www.fdi.ucm.es/profesor/mozos/AEC/aritm_pf.PDF (Fecha de consulta 1 de noviembre de 2008)

[CALI2008] Kalipedia: Las seàales analògicas y las seàales digitales. Madrid, 2008. http://co.kalipedia.com/tecnologia/tema/electronica/senales-analogicas-senales-digitales.html?x=20070822klpingtcn_137.Kes (Fecha de consulta: 2 de Noviembre de 2008).

[CARS2008] Planet Carsoft: USB: Universal Serial Bus. Argentina, 2008. http://www.forpas.us.es/aula/hardware/dia2_USB.pdf (Fecha de consulta: 2 de Noviembre de 2008).

[CASA2008] Casa domo: El portal del hogar digital. España, 2008. www.casadomo.com/diccionario.aspx (Fecha de consulta: Septiembre 14 de 2008)

[CIBE2007] Ciberaula Java. J2ME: Introducción al J2ME. España, 2007. http://java.ciberaula.com/articulo/introduccion_j2me/ (Fecha de consulta: en Septiembre 21 de 2008)

[CONS2007] Consumer Eroski: Qué es un micorprocesador? España, 2007. <http://www.consumer.es/web/es/tecnologia/hardware/2005/03/17/140483.php> (Fecha de consulta: 2 de Noviembre de 2008).

[DAMI2007] Damián Martínez: Sistemas de procesamiento de señal en tiempo real: Estudio comparativo de los DSP. España, 2007. <http://www4.ujaen.es/~damian/spstr/Estudio%20DSPs.pdf> (Fecha de consulta: en Septiembre 21 de 2008)

[DATA2008] Data Identification: What is OCR? United States, 2008. <http://www.dataid.com/aboutocr.htm> (Fecha de consulta: en Septiembre 21 de 2008)

[DEFI2008] Definición: Diccionario de definiciones online. 2008, www.definicion.org/diccionario/90 (Fecha de consulta: Septiembre 18 de 2008)

[EDNA1996] Access for Design, By Design. USB: a neat package with a few loose ends. *Richard A Quinnell* United States 1996. http://www.edn.com/archives/1996/102496/df_01.htm#USB%20fundamentals (Fecha de consulta: 2 de Noviembre de 2008).

[EIAT1993] EIA/TIA: Estándar RS232 1993: http://www.camiresearch.com/Data_Com_Basics/RS232_standard.html#anchor1154232 (Fecha de consulta: 16 de junio de 2009).

[EIAT2005] EIA/TIA: Niveles lógicos RS232 2005: <http://www.best-microcontroller-projects.com/how-rs232-works.html> (Fecha de consulta: 14 de junio de 2009).

[EVEL2009] Conversión Analógico a Digital. España, 2009. <http://www.eveliux.com/mx/conversion-analogico-digital-adc.php> (Fecha de consulta 15 de Junio de 2009)

[FREE2005] Freescale Semiconductor: User guide 56F800. United States, 2005. http://www.freescale.com/files/dsp/doc/user_guide/DSP56F800DBUM.pdf (Fecha de consulta: en Septiembre 21 de 2008)

[GRAN2007] Gran Angular: Diccionario de términos Fotográficos. Argentina, 2008. www.gran-angular.net/diccionario-de-terminos-fotograficos/2008/04/25/ (Fecha de consulta: Septiembre 14 de 2008)

[INTR2007] Java: Introducción a MIDP2.0. España, 2007. <http://leo.ugr.es/J2ME/MIDP/intro.htm> (Fecha de consulta: en Septiembre 21 de 2008)

[MASA2008] Mas adelante: Diseño de páginas web e información de tecnología. España, 2008. <http://www.masadelante.com/faq-gsm.htm> (Fecha de consulta: en Septiembre 16 de 2008)

[MATL2009] MATLAB: descripción del entorno 2009: <http://www.mathworks.com/products/matlab/description1.html> (Fecha de consulta: 16 de junio de 2009).

[MAXI2006] Maxim: MAX232 +5V-Powered, Multichannel RS-232Driver/Receiver 2006: <http://datasheets.maxim-ic.com/en/ds/MAX220-MAX249.pdf> (Fecha de consulta: 16 de junio de 2009).

[MICR2004] Microchip: fabricación Características del dsPIC30F4011. United States, 2004. <http://www.microchip.com> (Fecha de consulta: en Septiembre 14 de 2008)

[MONO2007] Monografías. MATLAB: Introducción a MATLAB. México, 2007. <http://www.monografias.com/trabajos5/matlab/matlab.shtml#intro> (Fecha de consulta: Septiembre 15 de 2008)

[NAVA2008] Navactiva: Portal para empresas de Navarra. España, 2008. <http://www.navactiva.com/web/es/atic/doc/glosario/internet/?letra=S> (Fecha de consulta: 2 de Noviembre de 2008).

[OMNI2007]: OmniVision: Single Chip Camera-to-Serial Bridge C328R 2007: http://www.davidus.sk/web/main/index/article_id/6, <http://bansky.net/blog/2008/03/jpeg-camera-and-micro-framework/> (Fecha de consulta 16 de junio de 2009).

[OPTC2000] Optrex Corporation: Dot Matrix Character LCD module User's manual 2000, página 13: http://www.ee.bilkent.edu.tr/~ee212/Dmcman_full.pdf (Fecha de consulta: 16 de junio de 2009).

[OPTD2000] Optrex Corporation: Dot Matrix Character LCD module User's manual 2000, página 44: http://www.ee.bilkent.edu.tr/~ee212/Dmcman_full.pdf (Fecha de consulta: 16 de junio de 2009).

[OPTO2000] Optrex Corporation: Dot Matrix Character LCD module User's manual 2000, páginas 34 a 28: http://www.ee.bilkent.edu.tr/~ee212/Dmcman_full.pdf (Fecha de consulta: 16 de junio de 2009).

[OPTP2000] Optrex Corporation: Dot Matrix Character LCD module User's manual 2000, página 32: http://www.ee.bilkent.edu.tr/~ee212/Dmcman_full.pdf (Fecha de consulta: 16 de junio de 2009).

[OPTR2000] Optrex Corporation: Dot Matrix Character LCD module User's manual 2000, página 35: http://www.ee.bilkent.edu.tr/~ee212/Dmcman_full.pdf (Fecha de consulta: 16 de junio de 2009).

[ORGA2008] Organización Internacional del Trabajo. Biblioteca digital: Glosario. Costa Rica, 2008. www.oit.or.cr/bidiped/Glosario.html (Fecha de consulta: Septiembre 14 de 2008)

[PHON2007] Phone Packages. Telephone warehouse: Glossary. Minneapolis, 2007. www.phonepackages.com/shop/glossary.html (Fecha de consulta: Septiembre 19 de 2008)

[PIPE2009] JAL Ingenieros en Instrumentación y Control: Redes de Datos. Juan Felipe García Sepúlveda: Estándar RS-232 2008. <http://pipelara20.tripod.com/rs232.htm> (Fecha de consulta: 12 de Enero de 2009).

[PORT2007] Portal Uralde. Informática: Diccionario Informático. Costa Rica, 2007. <http://www.portal-uralde.com/dicr.htm> (Fecha de consulta: Septiembre 19 de 2008)

[PRTE2008] Puerto Rico Telephone. Portal educativo: Glosario de Telecomunicaciones. Puerto Rico, 2008. www.prteeducativo.com/jovenes/glosario.htm (Fecha de consulta: Septiembre 16 de 2008)

[PUBL1999] Super tutoriales: Glosario de palabras claves. España, 1999. http://www.google.com/search?hl=es&lr=&defl=es&q=define:Procesador&sa=X&oi=glossary_definition&ct=title (Fecha de consulta: 2 de Noviembre de 2008).

[PUJE2000] Pontificia universidad javeriana. Carrera de ingeniería electrónica. Gilbert Alexander López Ovalle, Fabián Andrés López Rengifo, Jorge Alberto Velásquez Montaña, Mario Plata García: Sistema de telemedición de servicios públicos en Santafé de Bogotá 2000. <http://triton.javeriana.edu.co/carrera/tgrado/2000-2/telemedicion.PDF> (Fecha de consulta: 17 de Septiembre de 2008).

[RAMM2006] Que es la memoria ROM? Definición de la memoria ROM. Alicante, España, 2006. <http://www.masadelante.com/faq-memoria-rom.htm> (Fecha de consulta 1 de Noviembre de 2008)

[ROBE2008] Roberto E: Corriente Eléctrica. México, 2008. <http://www.angelfire.com/empire/seigfrid/Portada.html> (Fecha de consulta: en Septiembre 14 de 2008)

[ROBO2009] Robótica en Argentina. Comunicación MAX232: Conversor TTI-RS232. Argentina, 2008. http://robots-argentina.com.ar/Comunicacion_max232.htm (Fecha de consulta 16 de Enero de 2009)

[ROMM2008] Como funciona la memoria ROM?, Ordenadores y portátiles. <http://www.ordenadores-y-portatiles.com/memoria-rom.html> (Fecha de consulta 1 de noviembre de 2008)

[SHDB2008] Secretaria distrital de hacienda. Bogotá en datos: demografía, Número de viviendas por localidad - Demografía y vivienda (Bogotá 1951-2010) 2008. http://www.shd.gov.co/portal/page?_pageid=237,3593522&_dad=portal&_sc_hema=PORTAL (Fecha de consulta: 12 de Septiembre de 2008).

[SHOP2008] Shopping Online. Technology: Glosario. Argentina, 2008. <http://www.alien-tech.com.ar/glosario/u.htm> (Fecha de consulta: 20 Septiembre de 2008)

[SING2007] OmniVision: Single Chip Camera-to-Serial Bridge C328R User's manual 2007: http://ssdl.stanford.edu/ssdl/images/stories/AA236/0708A/Lab/Rover/Parts/cam_c328.pdf (Fecha de consulta: 16 de junio de 2009).

[TODO2007] TodoHard: Conectores, Cables y Circuitos. Argentina, 2007. <http://todohard.awardspace.com/docs/ConectorUSB/> (Fecha de consulta: 2 de Noviembre de 2008).

[TOSH2009] Toshiba: TCM8230MD CMOS Camera - 640x480 2004: http://www.sparkfun.com/commerce/product_info.php?products_id=8667 , <https://specialcomp.com/calao/dab/TCM8230MD.pdf> (Fecha de consulta: 16 de junio de 2009).

[UCOL2009] Universidad de Colima: Transmisión de datos: Estándares y diseño de cables. México, 2007. http://alumno.ucol.mx/~al026137/PracticaS_archivos/prac6.htm (Fecha de consulta 16 de Enero de 2009)

[UJAE2006] Estudio Comparativo de los DSP: Damian Martínez Muñoz, Universidad de Jaen. España, 2006. <http://www4.ujaen.es/~damian/spstr/Estudio%20DSPs.pdf> (Fecha de consulta 1 de Noviembre de 2008)

[UPCE2007] Universidad Peruana de Ciencias Aplicadas (UPC). Facultad de Ingeniería – Escuela Profesional de Ingeniería Electrónica, Procesamiento digital de señales. Luis Armando Boderó Bullón, Diego Alonso Sarmiento Pastor: Monitoreo remoto de procesos a través del canal de voz de teléfonos celulares gsm 2007. <http://www.upc.edu.pe/RepositorioAPS/0/13/JER/PARTICIPACIONENCONGRESOS/Monitoreo-Procesos-canal-GSM.PDF> (Fecha de consulta: 16 de Septiembre de 2008).

[UPVE2005] Universidad Politécnica de Valencia. Grupo de Investigación del ITI. Javier Cano y Juan Carlos Pérez: OCR (Optical Character Recognition) 2005. <http://web.iti.upv.es/actualidadtic/2003/11/2003-11-ocr.pdf> (Fecha de consulta: 2 de Noviembre de 2008).

[USER2004] Microchip: User Manual dsPIC30F4011. United States, 2004. <http://ww1.microchip.com/downloads/en/devicedoc/70135C.pdf> (Fecha de consulta 1 de noviembre de 2008)

[UTFS2004] Universidad Técnica Federico Santa María, Departamento de Electrónica: Introducción a los DSP. Valparaíso, 2004. <http://www.elo.utfsm.cl/~elo385/Documentos/Intro-DSP.pdf> (Fecha de consulta: 2 de Noviembre de 2008).

[WEBH2008] Web Hosting en México, Xpress-Hosting: servicio de Hosting para páginas web. México, 2008. www.xpress.com.mx/glosario_c.php (Fecha de consulta: en Septiembre 14 de 2008)

[WEBT2008] Web Taller. Lenguajes de Programación: Que es JAVA. México, 2008. <http://www.webtaller.com/construccion/lenguajes/java/lecciones/que-es-java.php> (Fecha de consulta: 21 de Septiembre de 2008).

[YOTE2008] Guía de ayuda documentada. Yoteca: glosario de móviles. México, 2008. www.yoteca.com/pg/glosario-de-moviles.asp (Fecha de consulta: Septiembre de 2008)

[ZOOOF2005] Blog informativo: portal informativo de noticias actuales. España, Abril de 2005. <http://zooglea.blogspot.com/2005/04/apuntes-sobre-criptografa-i-conceptos.html> (Fecha de consulta: en Septiembre de 15 2008)

13. ANEXOS

13.1 Anexo 1. Especificaciones del módulo CMOS C328R

Features

- Small in size, 20x28mm
- VGA resolution, down sample to QVGA or CIF
- 3.3V operation
- Low power consumption 60mA
- User friendly commands to control the module
- UART interface of up to 115.2Kbps
- Auto detect baud rate and make connection to the host
- Power saving mode
- Various lens options

System Configuration

1. Camera Sensor

The C328R module uses OmniVision VGA color digital CameraChips with an 8-bit YCbCr interface.

2. OV528 Serial Bridge

The OV528 Serial Bridge is a JPEG CODEC embedded controller chip that can compress and transfer image data from CameraChips to external device. The camera interface synchronizes with input video data and performs down sampling, clamping and windowing functions with desired resolution, as well as color conversion that is requested by the user through serial bus host commands. The JPEG CODEC can achieve higher compression ratio and better image quality for various image resolutions.

3. Program EEPROM

A serial type program memory is built-in for C328R to provide a set of user-friendly command interfacing to external host.

13.2 Anexo 2. Manual del Usuario

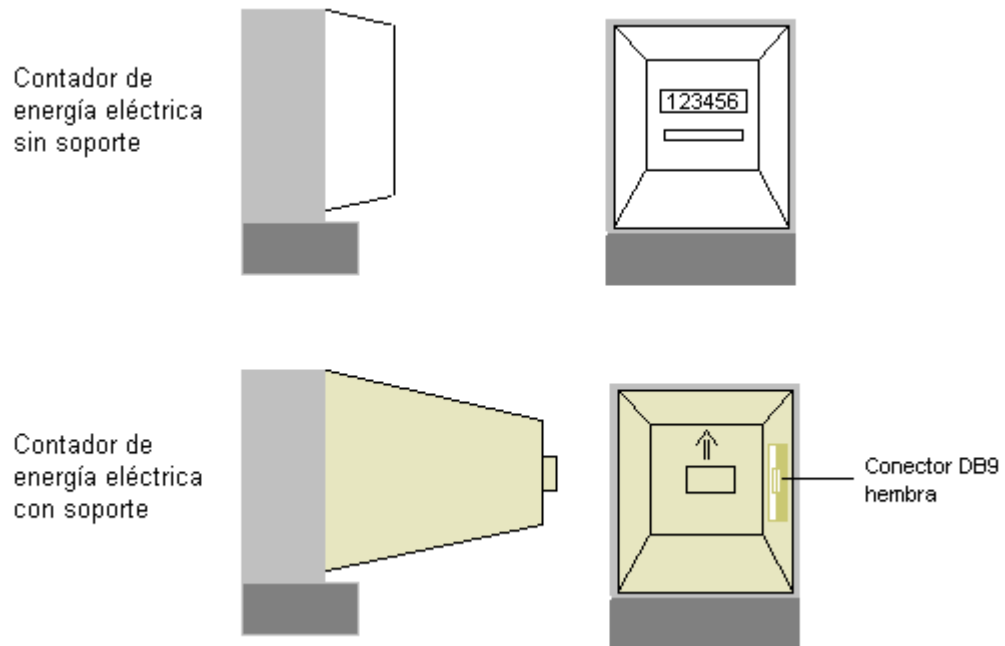
MANUAL DEL USUARIO

El sistema prototipo se encuentra listo para utilizarse sobre contadores monofásicos de energía eléctrica marca Skateks tipo CO-U449MT e incluye los siguientes componentes:

- Soporte con módulo CMOS C328R incorporado.
- Sistema de procesamiento y visualización de datos.
- Cable DB9 macho-macho.
- Fuente de alimentación de 5v.

SOPORTE CON MÓDULO CMOS C328R

El soporte posee una flecha negra impresa indicando la orientación como debe colocarse sobre el contador de energía eléctrica, tal como se puede apreciar en el dibujo.

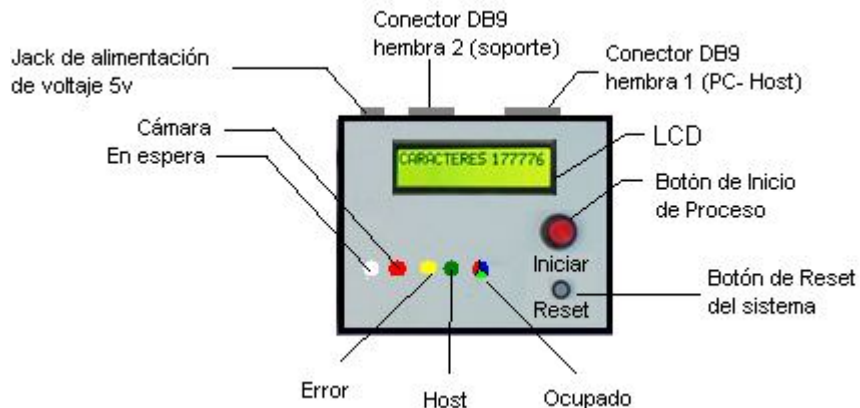


Este soporte es elaborado en madera de bajo peso con recubrimiento de yeso para brindar mayor dureza y estética de diseño. Posee un conector lateral DB9 hembra para comunicarse con el resto de dispositivos.

SISTEMA DE PROCESAMIENTO Y VISUALIZACIÓN DE DATOS (SPVD)

Este sistema muestra a través de una pantalla los datos de la lectura del contador. Posee las siguientes características:

- Conector DB9 Hembra 1: este conector de 9 pines permite conectar el sistema a un dispositivo externo como un computador u otro componente configurador previamente para hacerlo.
- Conector DB9 hembra 2: este conector de 9 pines permite conectar el sistema con el soporte que se coloca sobre el contador.
- Jack de alimentación de voltaje (5v): este conector posee la entrada de alimentación del sistema, la cual debe ser de 5v.
- LCD: esta pantalla de cristal líquido visualiza los resultados enviados desde el soporte con módulo CMOS C328R.
- Botón de inicio de proceso: es el botón que da inicio al proceso de lectura del contador.
- Botón de reset del sistema: cuando un fallo ocurre en el proceso de lectura, este botón reinicia el prototipo para ejecutar la lectura nuevamente.
- Leds indicadores: muestran el estado del sistema donde cada led presenta un estado diferente.
 - En espera: indica que el dispositivo se ha encendido esta esperando que se inicie el proceso de lectura.
 - Cámara: indica si la conexión con el soporte con módulo CMOS C328R se ha establecido.
 - Error: se enciende cuando algún error ha ocurrido en el proceso de lectura del contador.
 - Host: indica si la conexión con un pc u otro dispositivo externo diferente al soporte C328R ha sido establecida.
 - Ocupado: se enciende cuando el sistema inicia el proceso de lectura y visualización de datos.



FUNCIONAMIENTO

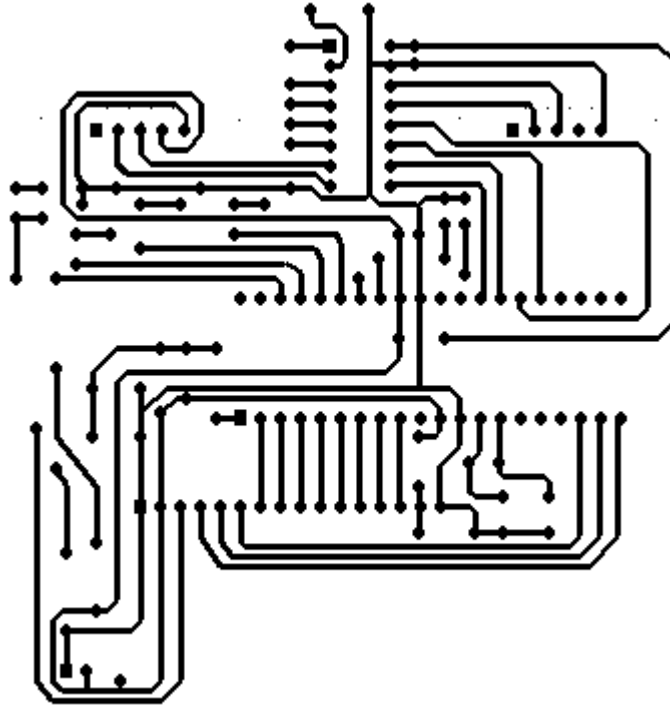
Los pasos a seguir para el correcto funcionamiento del sistema son:

1. Colocar el soporte con módulo CMOS C328R sobre el contador verificando que la flecha negra se encuentre apuntando al norte del contador.
2. Conectar un extremo del cable DB9 macho-macho en el conector DB9 hembra del soporte, y el otro extremo en el conector DB9 hembra 2 del sistema de procesamiento y visualización de datos (SPVD).
3. Conectar la alimentación de 5v en el jack de alimentación del SPVD.
4. Se encenderá el led indicador de Cámara.
5. Se encenderá el led indicador "En Espera".
6. Presionar el botón "Iniciar".
7. Se encenderá el led indicador de "Ocupado".
8. Se visualiza el resultado en la LCD.
9. Si se desea realizar una nueva lectura, presionar el botón reset y continuar desde el paso 4.

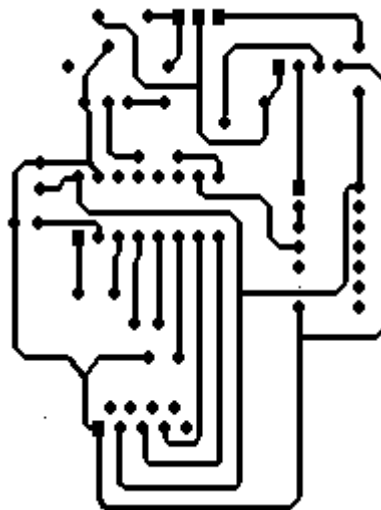
Nota: En caso de que se presente un error se encenderá el led indicador "Error", por lo tanto debe presionarse el botón reset para iniciar nuevamente.

Si después de conectar la alimentación el led "Cámara" no se enciende y por el contrario se enciende el led "Error", quiere decir que el cable de comunicación DB9 macho-macho no se encuentra bien conectado.

13.3 Anexo 3. Circuito Impreso para el DSP y el módulo CMOS C328R



Circuito impreso DSP



Circuito Impreso modulo CMOS C328R

13.4 Anexo 4: Tabla con las pruebas de luminancia – porcentaje de similitud basada en 50 lecturas realizadas.

		Prueba 1 - Luminancia promedio: 25							
		Porcentaje de similitud por posición.							
#	contador	I	II	III	IV	V	VI	Interm.	Dif
1	00112-0	69	68	72	73	70	58	65	3,3333
2	00112-1	69	68	72	73	70	59	64	4,5
3	00112-2	69	68	72	73	70	57	65	3,1667
4	00112-3	69	68	72	73	70	59	65	3,5
5	00112-4	69	68	72	73	70	60	65	3,6667
6	00112-5	69	68	72	73	70	57	63	5,1667
7	00112-6	69	68	72	73	70	59	65	3,5
8	00112-7	69	68	72	73	70	59	65	3,5
9	00112-8	69	68	72	73	70	59	65	3,5
10	00112-9	69	68	72	73	70	57	65	3,1667
11	23344-0	68	69	71	72	68	58	63	4,6667
12	23344-1	68	69	71	72	68	58	61	6,6667
13	23344-2	68	69	71	72	68	60	61	7
14	23344-3	68	69	71	72	68	60	63	5
15	23344-4	68	69	71	72	68	59	62	5,8333
16	23344-5	68	69	71	72	68	61	63	5,1667
17	23344-6	68	69	71	72	68	60	61	7
18	23344-7	68	69	71	72	68	60	61	7
19	23344-8	68	69	71	72	68	57	61	6,5
20	23344-9	68	69	71	72	68	59	62	5,8333
21	55667-0	70	67	69	71	69	58	62	5,3333
22	55667-1	70	67	69	71	69	58	61	6,3333
23	55667-2	70	67	69	71	69	58	61	6,3333
24	55667-3	70	67	69	71	69	58	61	6,3333
25	55667-4	70	67	69	71	69	59	60	7,5
26	55667-5	70	67	69	71	69	57	60	7,1667
27	55667-6	70	67	69	71	69	59	60	7,5
28	55667-7	70	67	69	71	69	59	61	6,5
29	55667-8	70	67	69	71	69	59	60	7,5
30	55667-9	70	67	69	71	69	57	60	7,1667
31	78899-0	68	70	72	69	68	60	62	5,8333

32	78899-1	68	70	72	69	68	60	63	4,8333
33	78899-2	68	70	72	69	68	60	62	5,8333
34	78899-3	68	70	72	69	68	60	62	5,8333
35	78899-4	68	70	72	69	68	58	62	5,5
36	78899-5	68	70	72	69	68	60	62	5,8333
37	78899-6	68	70	72	69	68	60	60	7,8333
38	78899-7	68	70	72	69	68	57	60	7,3333
39	78899-8	68	70	72	69	68	57	60	7,3333
40	78899-9	68	70	72	69	68	57	60	7,3333
41	12345-6	67	68	71	69	71	56	60	7
42	78901-2	68	68	75	71	70	57	64	4,1667
43	34567-8	68	69	73	72	70	57	62	6,1667
44	90123-4	69	69	72	73	71	56	63	5,3333
45	56789-0	68	71	72	74	70	60	64	5,1667
46	00000-0	69	68	71	72	71	59	58	10,333
47	89898-9	67	71	77	72	70	59	62	7,3333
48	38383-8	67	70	72	74	69	58	62	6,3333
49	27272-7	69	70	70	73	72	58	62	6,6667
50	99999-9	67	69	73	69	69	57	58	9,3333

Prueba 2 - Luminancia promedio: 45

Porcentaje por posición.

#	contador	I	II	III	IV	V	VI	Interm.	Dif
1	00112-0	86	85	89	90	87	75	67	18,333
2	00112-1	86	85	89	90	87	76	66	19,5
3	00112-2	86	85	89	90	87	74	67	18,167
4	00112-3	86	85	89	90	87	76	67	18,5
5	00112-4	86	85	89	90	87	77	67	18,667
6	00112-5	86	85	89	90	87	74	65	20,167
7	00112-6	86	85	89	90	87	76	67	18,5
8	00112-7	86	85	89	90	87	76	67	18,5
9	00112-8	86	85	89	90	87	76	67	18,5
10	00112-9	86	85	89	90	87	74	67	18,167
11	23344-0	85	86	88	89	85	75	65	19,667
12	23344-1	85	86	88	89	85	75	63	21,667
13	23344-2	85	86	88	89	85	77	63	22
14	23344-3	85	86	88	89	85	77	65	20
15	23344-4	85	86	88	89	85	76	64	20,833
16	23344-5	85	86	88	89	85	78	65	20,167
17	23344-6	85	86	88	89	85	77	63	22
18	23344-7	85	86	88	89	85	77	63	22
19	23344-8	85	86	88	89	85	74	63	21,5
20	23344-9	85	86	88	89	85	76	64	20,833
21	55667-0	87	84	86	88	86	75	64	20,333
22	55667-1	87	84	86	88	86	75	63	21,333
23	55667-2	87	84	86	88	86	75	63	21,333
24	55667-3	87	84	86	88	86	75	63	21,333
25	55667-4	87	84	86	88	86	76	62	22,5
26	55667-5	87	84	86	88	86	74	62	22,167
27	55667-6	87	84	86	88	86	76	62	22,5
28	55667-7	87	84	86	88	86	76	63	21,5
29	55667-8	87	84	86	88	86	76	62	22,5
30	55667-9	87	84	86	88	86	74	62	22,167
31	78899-0	85	87	89	86	85	77	64	20,833
32	78899-1	85	87	89	86	85	77	65	19,833
33	78899-2	85	87	89	86	85	77	64	20,833
34	78899-3	85	87	89	86	85	77	64	20,833
35	78899-4	85	87	89	86	85	75	64	20,5
36	78899-5	85	87	89	86	85	77	64	20,833

37	78899-6	85	87	89	86	85	77	62	22,833
38	78899-7	85	87	89	86	85	74	62	22,333
39	78899-8	85	87	89	86	85	74	62	22,333
40	78899-9	85	87	89	86	85	74	62	22,333
41	12345-6	84	85	88	86	88	73	62	22
42	78901-2	85	85	92	88	87	74	66	19,167
43	34567-8	85	86	90	89	87	74	64	21,167
44	90123-4	86	86	89	90	88	73	65	20,333
45	56789-0	85	88	89	91	87	77	66	20,167
46	00000-0	86	85	88	89	88	76	60	25,333
47	89898-9	84	88	94	89	87	76	64	22,333
48	38383-8	84	87	89	91	86	75	64	21,333
49	27272-7	86	87	87	90	89	75	64	21,667
50	99999-9	84	86	90	86	86	74	60	24,333

Prueba 2 - Luminancia promedio: 65									
Porcentaje por posición.									
#	contador	I	II	III	IV	V	VI	Interm.	Dif
1	00112-0	72	71	75	76	73	61	66	5,3333
2	00112-1	72	71	75	76	73	62	65	6,5
3	00112-2	72	71	75	76	73	60	66	5,1667
4	00112-3	72	71	75	76	73	62	66	5,5
5	00112-4	72	71	75	76	73	63	66	5,6667
6	00112-5	72	71	75	76	73	60	64	7,1667
7	00112-6	72	71	75	76	73	62	66	5,5
8	00112-7	72	71	75	76	73	62	66	5,5
9	00112-8	72	71	75	76	73	62	66	5,5
10	00112-9	72	71	75	76	73	60	66	5,1667
11	23344-0	71	72	74	75	71	61	64	6,6667
12	23344-1	71	72	74	75	71	61	62	8,6667
13	23344-2	71	72	74	75	71	63	62	9
14	23344-3	71	72	74	75	71	63	64	7
15	23344-4	71	72	74	75	71	62	63	7,8333
16	23344-5	71	72	74	75	71	64	64	7,1667
17	23344-6	71	72	74	75	71	63	62	9
18	23344-7	71	72	74	75	71	63	62	9
19	23344-8	71	72	74	75	71	60	62	8,5
20	23344-9	71	72	74	75	71	62	63	7,8333
21	55667-0	73	70	72	74	72	61	63	7,3333
22	55667-1	73	70	72	74	72	61	62	8,3333
23	55667-2	73	70	72	74	72	61	62	8,3333
24	55667-3	73	70	72	74	72	61	62	8,3333
25	55667-4	73	70	72	74	72	62	61	9,5
26	55667-5	73	70	72	74	72	60	61	9,1667
27	55667-6	73	70	72	74	72	62	61	9,5
28	55667-7	73	70	72	74	72	62	62	8,5
29	55667-8	73	70	72	74	72	62	61	9,5
30	55667-9	73	70	72	74	72	60	61	9,1667
31	78899-0	71	73	75	72	71	63	63	7,8333
32	78899-1	71	73	75	72	71	63	64	6,8333
33	78899-2	71	73	75	72	71	63	63	7,8333
34	78899-3	71	73	75	72	71	63	63	7,8333
35	78899-4	71	73	75	72	71	61	63	7,5
36	78899-5	71	73	75	72	71	63	63	7,8333

37	78899-6	71	73	75	72	71	63	61	9,8333
38	78899-7	71	73	75	72	71	60	61	9,3333
39	78899-8	71	73	75	72	71	60	61	9,3333
40	78899-9	71	73	75	72	71	60	61	9,3333
41	12345-6	70	71	74	72	74	59	61	9
42	78901-2	71	71	78	74	73	60	65	6,1667
43	34567-8	71	72	76	75	73	60	63	8,1667
44	90123-4	72	72	75	76	74	59	64	7,3333
45	56789-0	71	74	75	77	73	63	65	7,1667
46	00000-0	72	71	74	75	74	62	59	12,333
47	89898-9	70	74	80	75	73	62	63	9,3333
48	38383-8	70	73	75	77	72	61	63	8,3333
49	27272-7	72	73	73	76	75	61	63	8,6667
50	99999-9	70	72	76	72	72	60	59	11,333

Prueba 2 - Luminancia promedio: 127

Porcentaje por posición.

#	contador	I	II	III	IV	V	VI	Interm.	Dif
1	00112-0	54	53	57	58	55	43	52	1,33333333
2	00112-1	54	53	57	58	55	44	51	2,5
3	00112-2	54	53	57	58	55	42	52	1,16666667
4	00112-3	54	53	57	58	55	44	52	1,5
5	00112-4	54	53	57	58	55	45	52	1,66666667
6	00112-5	54	53	57	58	55	42	50	3,16666667
7	00112-6	54	53	57	58	55	44	52	1,5
8	00112-7	54	53	57	58	55	44	52	1,5
9	00112-8	54	53	57	58	55	44	52	1,5
10	00112-9	54	53	57	58	55	42	52	1,16666667
11	23344-0	53	54	56	57	53	43	50	2,66666667
12	23344-1	53	54	56	57	53	43	48	4,66666667
13	23344-2	53	54	56	57	53	45	48	5
14	23344-3	53	54	56	57	53	45	50	3
15	23344-4	53	54	56	57	53	44	49	3,83333333
16	23344-5	53	54	56	57	53	46	50	3,16666667
17	23344-6	53	54	56	57	53	45	48	5
18	23344-7	53	54	56	57	53	45	48	5
19	23344-8	53	54	56	57	53	42	48	4,5
20	23344-9	53	54	56	57	53	44	49	3,83333333
21	55667-0	55	52	54	56	54	43	49	3,33333333
22	55667-1	55	52	54	56	54	43	48	4,33333333
23	55667-2	55	52	54	56	54	43	48	4,33333333
24	55667-3	55	52	54	56	54	43	48	4,33333333
25	55667-4	55	52	54	56	54	44	47	5,5
26	55667-5	55	52	54	56	54	42	47	5,16666667
27	55667-6	55	52	54	56	54	44	47	5,5
28	55667-7	55	52	54	56	54	44	48	4,5
29	55667-8	55	52	54	56	54	44	47	5,5
30	55667-9	55	52	54	56	54	42	47	5,16666667
31	78899-0	53	55	57	54	53	45	49	3,83333333
32	78899-1	53	55	57	54	53	45	50	2,83333333
33	78899-2	53	55	57	54	53	45	49	3,83333333
34	78899-3	53	55	57	54	53	45	49	3,83333333
35	78899-4	53	55	57	54	53	43	49	3,5
36	78899-5	53	55	57	54	53	45	49	3,83333333

37	78899-6	53	55	57	54	53	45	47	5,83333333
38	78899-7	53	55	57	54	53	42	47	5,33333333
39	78899-8	53	55	57	54	53	42	47	5,33333333
40	78899-9	53	55	57	54	53	42	47	5,33333333
41	12345-6	52	53	56	54	56	41	47	5
42	78901-2	53	53	60	56	55	42	51	2,16666667
43	34567-8	53	54	58	57	55	42	49	4,16666667
44	90123-4	54	54	57	58	56	41	50	3,33333333
45	56789-0	53	56	57	59	55	45	51	3,16666667
46	00000-0	54	53	56	57	56	44	45	8,33333333
47	89898-9	52	56	62	57	55	44	49	5,33333333
48	38383-8	52	55	57	59	54	43	49	4,33333333
49	27272-7	54	55	55	58	57	43	49	4,66666667
50	99999-9	52	54	58	54	54	42	45	7,33333333

13.5 Anexo 5: Porcentaje de error cuantitativo de las lecturas.

Luminancia promedio: 45			
#	Contador	DSP	Error
1	00112-0	00112-0	0
2	00112-1	00112-1	0
3	00112-2,5	00112-3	0
4	00112-3	00112-3	0
5	00112-4	00112-4	0
6	00112-5,5	00112-6	0
7	00112-6	00112-6	0
8	00112-7	00112-7	0
9	00112-8,5	00112-8	0
10	00112-9	00112-9	0
11	23344-0	23344-0	0
12	23344-1,5	23344-2	0
13	23344-2	23344-2	0
14	23344-3	23344-3	0
15	23344-4,5	23344-5	0
16	23344-5	23344-5	0
17	23344-6	23344-6	0
18	23344-7,5	23344-8	0
19	23344-8	23344-8	0
20	23344-9	23344-9	0
21	55667-0,5	55667-1	0
22	55667-1	55667-1	0
23	55667-2	55667-2	0
24	55667-3,5	55667-4	0
25	55667-4	55667-4	0
26	55667-5	55667-5	0
27	55667-6,5	55667-7	0
28	55667-7	55667-7	0
29	55667-8	55667-8	0
30	55667-9,5	55667-0	0
31	78899-0	78899-0	0
32	78899-1	78899-1	0
33	78899-2,5	78899-3	0

34	78899-3	78899-3	0	
35	78899-4	78899-4	0	
36	78899-5,5	78899-6	0	
37	78899-6	78899-6	0	
38	78899-7	78899-7	0	
39	78899-8,5	78899-0	1	
40	78899-9	78899-9	0	
41	12345-6	12345-6	0	
42	78901-2,5	78901-2	0	
43	34567-8	34567-8	0	
44	90123-4	90123-4	0	
45	56789-0,5	56789-1	0	
46	00000-0	00000-0	0	
47	89898-8,5	89898-3	1	
48	38383-8,5	38383-8	1	
49	27272-7	27272-2	1	
50	99999-9	99999-9	0	
			Total Errores	4
Porcentaje de Error (%)		8		