

EQUIPO PARA CONTROLAR EL USO DE MAQUINARIA

Aureliano F Núñez Mendoza

Estudiante de Ingeniería Técnica en Telecomunicaciones – EPSEVG UPC

Resumen

El presente artículo describe la elaboración de un equipo electrónico ideado con la finalidad de ser ubicado en maquinaria, cuyo uso debe ser restringido y debe quedar registrado.

El proyecto parte de la necesidad de limitar y registrar el uso dado a los transportadores portuarios. Tras su análisis se establecen los requerimientos a cumplir, los cuales llevan a la definición del equipo a realizar.

En el diseño del circuito electrónico, se han seleccionado componentes altamente integrados, de fácil utilización y que resuelven perfectamente la función asignada. El diseño constructivo se ha realizado procurando que el dispositivo ocupe el menor espacio posible. También se ha construido un prototipo del equipo y se le ha dotado de una plataforma para realizar pruebas. Se ha desarrollado el software del dispositivo y se ha evaluado su funcionamiento. Finalmente, se han propuesto una serie de mejoras a incorporar en el equipo realizado.

Summary

The following article describes the development of an electronic system designed to be embedded in machinery whose use must be restricted and logged.

The project motivation is the need to limit and register the use of straddle carriers in maritime ports. After analyzing this need, the system requirements are established which lead to the definition of the equipment to be developed.

In the design of the electronic circuitry, highly integrated, easy to use components that perform perfectly the required function have been used. The physical design has tried to reduce the equipment size as much as possible. In addition, a prototype of the electronic system, including a testing platform, has been built. The software required by the equipment has been developed and evaluated. Finally, a number of improvements of the developed equipment have been proposed.

1. Introducción

En las terminales de contenedores se utilizan carretillas pórtico, “straddle carrier” en inglés, para trasladar y apilar los contenedores de mercancías. El manejo y conducción es realizado por un solo operario desde el interior de una cabina, situada en lo alto del transportador. Esta cabina, de pequeñas dimensiones, dispone de un pequeño cuadro de mando que aloja las botoneras e indicadores que permiten la interacción con la carretilla.

En ocasiones se producen incidentes de diversa gravedad, como pueden ser el extravío o deterioro de contenedores, averías provocadas por mal uso de las carretillas, accidentes incluso mortales, desvío de cargas como consecuencia de contrabando o tráfico de estupefacientes, entre otros.

La correspondiente investigación para intentar esclarecer los incidentes acaecidos, crea la necesidad de disponer de un registro fiable de quién y cuándo ha hecho uso de una determinada carretilla. En ocasiones, incluso, este tipo de registro ha sido solicitado por la policía como consecuencia de sus propias investigaciones.

En el presente proyecto se pretende desarrollar un equipo electrónico que proporcione los recursos necesarios para generar un registro que informe de qué conductores, y en qué momento, han utilizado una determinada carretilla pórtico. El equipo también registrará ciertos eventos de interés, conocidos mediante entradas digitales conectadas al cuadro de mando de la carretilla.

2. Requerimientos del equipo a desarrollar

La identificación de los usuarios del transportador, se realizará mediante tarjetas de banda magnética. El equipo estará ubicado en la cabina de conducción y dispondrá de un lector de tarjetas. Mediante una tabla software, el equipo relacionará la tarjeta introducida con un usuario y un tipo de uso permitido. Se definen los tipos de uso siguientes:

- Sin autorización: No se permite el uso del vehículo.
- Mantenimiento: Permite la puesta en marcha, pero no la conducción.
- Conducción: Permite la utilización al completo.

El equipo contará con dos salidas por relé conectadas al cuadro de mando del vehículo. Una habilitará su puesta en marcha y la otra habilitará su conducción.

El lector de tarjetas será motorizado y retendrá la tarjeta mientras el transportador esté en marcha, estado que se conoce mediante una entrada digital. La retención de la tarjeta asegura que quede perfectamente reflejado el instante inicial y final de uso. Si la tarjeta no quedase capturada, podría efectuarse un cambio de conductor sin que este hecho quedase registrado, bastaría con mantener en marcha el transportador durante el cambio de conductor.

La entrada que indica que el vehículo está en marcha tendrá asociado un contador de horas de funcionamiento. Éste podrá utilizarse para planificar acciones de mantenimiento preventivo.

Las carretillas suelen disponer de una señal de advertencia que, a partir de unos acelerómetros, avisa en caso de que se

efectúen maniobras bruscas durante la conducción. Esta señal quedará registrada en el equipo.

Se contará con una entrada que informará si el vehículo se encuentra en movimiento y se registrará su conexión y su desconexión. Durante el desplazamiento, se activará una salida que pondrá en marcha un inhibidor de telefonía móvil ubicado en la cabina de conducción.

Otra entrada indicará si la carretilla lleva carga o no. Esta entrada, junto con la de transportador en circulación, proporcionará información relativa a los instantes en los que el conductor ha efectuado paradas y maniobras de carga y descarga. Disponer de esta información, puede resultar muy útil en las investigaciones sobre casos de contrabando.

También se contará con una entrada digital proveniente del indicador de nivel mínimo de combustible y otra conectada a la señal de aviso de incidencia en el transportador.

Finalmente, se precisa que el equipo posea una interfaz Ethernet, que permita la conexión de un ordenador remoto desde el que realizar tareas de configuración y consulta de la información almacenada. Cabe señalar que es habitual que las carretillas dispongan de conexión LAN a través de la infraestructura WiFi de la terminal.

A partir de los requerimientos descritos, confeccionamos una relación de las características hardware que deberá poseer el equipo a diseñar.

- Núcleo microprocesador programable
- Memoria remanente
- Reloj calendario
- Puerto Ethernet
- Lector de banda magnética motorizado
- Pulsador de extracción de tarjeta
- Micropulsador de reset de parámetros
- 4 leds de señalización
- 6 Entradas digitales
- 3 Salidas por relé
- Alimentación y entradas a 24 Vdc
- Compacto y tamaño reducido

Como núcleo central del equipo se utilizará un sistema embebido, que incorpore un microcontrolador con puertos I/O e interfaz Ethernet. El módulo microcontrolador irá conectado a una placa de circuito impreso que le aportará las interfaces de entrada y salida, las alimentaciones y los elementos adicionales que se precisen.

3. Componentes principales del equipo

En los siguientes apartados se describirán los elementos que por su importancia o complejidad requieren una mención especial. El resto de componentes se analizarán cuando se aborde el diseño del circuito.

3.1. Lector de banda magnética

Para la realización de nuestro equipo, se ha seleccionado el lector MTM-770T-1R-0101 del fabricante japonés Neuron. Se trata de un lector OEM motorizado, con cabezal de sólo lectura para la pista 2 y con interfaz TTL. [1]

La interfaz de este lector es muy similar a las de otras marcas y, por tanto, la descripción que se hará sobre él puede considerarse como un estándar para este tipo de lectores.

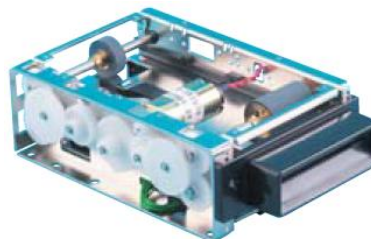


Fig. 1. Lector de tarjetas MTM-770T-1R-0101 de Neuron

Para interactuar con la sección motorizada se dispone de una interfaz TTL compuesta por 5 salidas digitales (denominadas PSS1, PSS2, PSS3, PSS4 y PSS5) y por 2 entradas digitales (denominadas MDSE y MDDR). Todas estas señales trabajan en lógica negativa. Las salidas de PSS1 a PSS5 (card position) provienen de unos detectores que determinan la posición de la tarjeta, la entrada MDSE (motor driving control signal) permite poner en marcha el motor de arrastre de la tarjeta, mediante la entrada MDDR (motor driving direction control signal) se puede escoger la dirección de arrastre.

La interfaz con el cabezal de lectura cuenta con 1 entrada, denominada DCD, y con 3 salidas, denominadas CLS, RDP y RCP. La señal DCD (decode stop signal) permite activar el funcionamiento del circuito decodificador del cabezal, la señal CLS (card loading signal) informa de que se está pasando una tarjeta por el cabezal de lectura, la señal RDP (read data pulse) contiene los datos y la señal RCP (read clock pulse) es un sincronismo que se debe utilizar para validar el dato presente en RDP.

A continuación, se muestra el cronograma de éstas señales. Del paso 1 hasta el 6 se corresponde con la inserción de una tarjeta, del paso 7 al 10 con la expulsión y el paso 11 es la extracción manual.

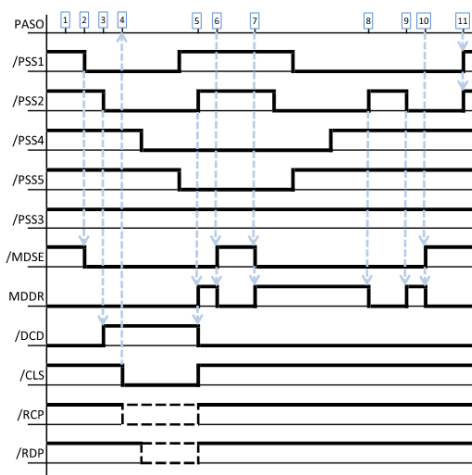


Fig. 2. Cronograma de una secuencia de inserción y extracción

Para codificar los caracteres se utilizan 4 bits. Además, se añade un bit de paridad impar; es decir, los bits del carácter, junto con el bit de paridad, tienen una cantidad impar de

valores a 1. Los bytes se graban en la pista comenzando por el bit menos significativo y finalizando por el bit de paridad. Los caracteres pueden convertirse a ASCII sumando 30H.

La trama grabada en la pista, comienza con un carácter reservado para marcar el inicio (“Start Sentinel”) y termina con otro carácter reservado para marcar el final (“End Sentinel”). A continuación de la trama se añade un byte de comprobación (“LRC”). Éste se calcula realizando la función XOR de todos los bytes de la trama, sin incluir los bits de paridad.

3.2. Sistema embebido

Como núcleo procesador se escogió el módulo RabbitCore[®] RCM3200 del fabricante Digi International. Éste cuenta con Ethernet 10/100 Base-T, 512 KB flash para programa, 256 KB RAM para datos y reloj calendario RTC. Tanto el reloj como la RAM pueden estar conectados a una batería externa de respaldo. [2]

También dispone de 52 puertos I/O, 4 puertos SPI, 11 timers, 4 PWM y watchdog. Se programa con el entorno de desarrollo Dynamic C y dispone de gran cantidad de librerías libres de royalties.

Aunque se alimenta a 3,3 voltios, la mayoría de sus puertos soportan 5 voltios y sus umbrales de conmutación son TTL compatible, por tanto, puede ser conectado directamente a señales TTL.



Fig. 3. RabbitCore[®] RCM3200 de Digi International

3.3. Memoria de almacenamiento

Para almacenar los eventos registrados se utilizará una memoria flash adicional. Ésta aportará gran cantidad de espacio de almacenamiento y remanencia aunque se agote la batería.

Las memorias flash se borran por sectores y tienen un número máximo de grabaciones. Estas limitaciones no suponen un problema a la hora de almacenar un histórico, pues éste tiene un comportamiento de tipo FIFO, es decir, la información más antigua es la que primero se elimina.

Se selecciona la memoria SST25VF080B de Microchip Technology. Tiene interfaz SPI, 1 MB de capacidad y sus sectores son de 4 KB. [3]

En la comunicación SPI se utilizan 3 entradas, denominadas CE, SCK y SI, y 1 salida, etiquetada como SO. La entrada CE (chip enable) sirve para habilitar el dispositivo, la entrada SCK (serial clock) es por donde recibe la señal de sincronismo, la entrada SI (serial data input) se utiliza para enviar información al dispositivo y, finalmente, mediante la salida SO (serial data output) obtenemos la información

proveniente del dispositivo. Tanto para la entrada SI, como para la salida SO, el dato se pone en el flanco negativo de SCK, y puede consultarse su valor en el flanco positivo de SCK. Toda la información que se transmite mediante el canal serie, comienza por el bit más significativo (MSB).

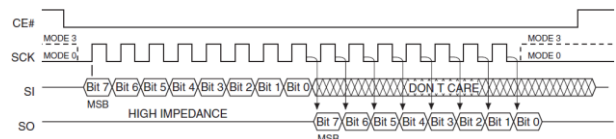


Fig. 4. Cronograma de una comunicación SPI

4. Diseño del circuito electrónico

El módulo RCM3200 tan solo necesita ser alimentado para poderlo utilizar. A partir de este punto, el diseño consistirá en ir implementando la conexión entre este módulo y los otros elementos del equipo. En base a la funcionalidad que requiera el elemento a conectar, se escogerán pines de los puertos del RCM3200 y se diseñará el circuito de interfaz que se precise para la interconexión.

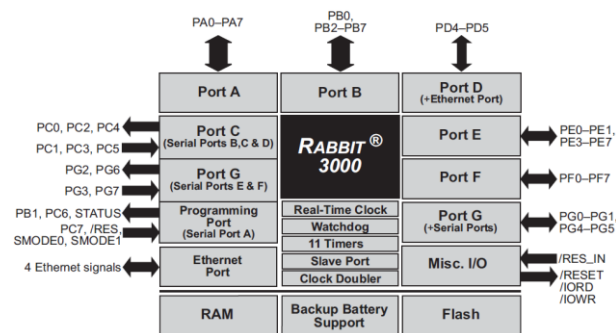


Fig. 5. Puertos del módulo Rabbit RCM3200

4.1. Conexión del lector de tarjetas

Puesto que las señales del lector son TTL y el módulo Rabbit es TTL compatible, la interconexión es directa, es decir, sin necesidad de búferes intermedios.

Todas las entradas y salidas se conectarán a pines I/O generales, salvo la entrada de sincronismo (RCP) que se conectará a una entrada de interrupción, pues requiere atención inmediata. Justo en el instante de producirse el flanco de sincronismo, el programa se desviará a una rutina de interrupción que almacenará el dato recibido mediante la señal RDP. Todos los bits recibidos y almacenados, se procesarán posteriormente en el ciclo normal de ejecución.

4.2. Conexión de la memoria flash

La memoria dispone de una interfaz SPI que se deberá conectar a uno de los canales SPI del módulo Rabbit. Se escoge utilizar el puerto serie B del citado módulo. Esto supone conectar, de forma directa, el pin PB2 (CLKB) con la señal de reloj de la memoria (SCK), el pin PC4 (TXB) con la entrada de datos (SI) y el pin PC5 (RXB) con la salida de datos (SO).

Para la señal de habilitación de la memoria se utilizará un pin de un puerto I/O, que por software se configurará como salida. Se escoge el pin PB2, pues dispone de un pull-up incorporado, y se conectará con el pin CS de la memoria.

Los otros 2 pines con los que cuenta la memoria, no se necesitan para nuestra aplicación, por lo que se mantendrán deshabilitados, aplicándoles VCC.

4.3. Circuito de la batería de respaldo

Para mantener el funcionamiento del reloj y proporcionar remanencia a la memoria RAM, en caso de ausencia de la alimentación del equipo, se utilizará la batería recargable MT2032-T25 de 65 mAh. Con el circuito de carga diseñado se conseguirá una autonomía de 226 días y un tiempo de recarga de 36 horas.

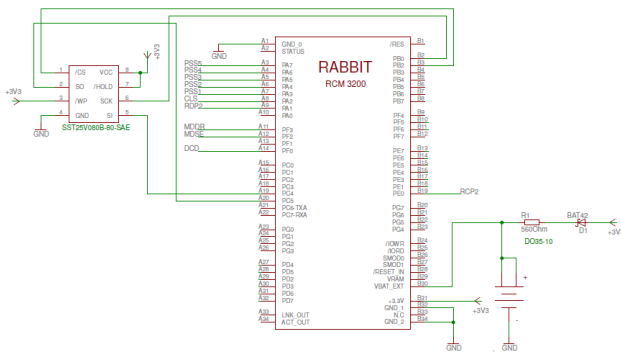


Fig. 6. Conexión de la lectora, la memoria flash y la batería

4.4. Circuito de las entradas digitales

Los pulsadores del equipo se conectarán directamente a dos entradas que disponen de pull-up interno. Las entradas externas pasarán por un optoacoplador para conseguir aislamiento galvánico. Se utilizará un optoacoplador de tipo bidireccional para permitir conexiones PNP o NPN.

Para limitar la corriente que circula por el diodo led de entrada del optoacoplador, se utilizará un divisor de tensión en lugar de una sola resistencia en serie. De esta forma se puede centrar el nivel de conmutación, con lo que se mejora la inmunidad frente al ruido eléctrico. Con los valores calculados, las entradas admitirán una tensión de entrada entre 11 y 30 Vdc y consumirán 7 mA por canal.

El colector de salida del optoacoplador se conectará a 5 voltios y el emisor a las entradas del microcontrolador. Para forzar un valor 0 en la entrada, cuando el transistor esté en corte, se colocarán pull-downs y se utilizarán puertos que no tengan pull-up interno.

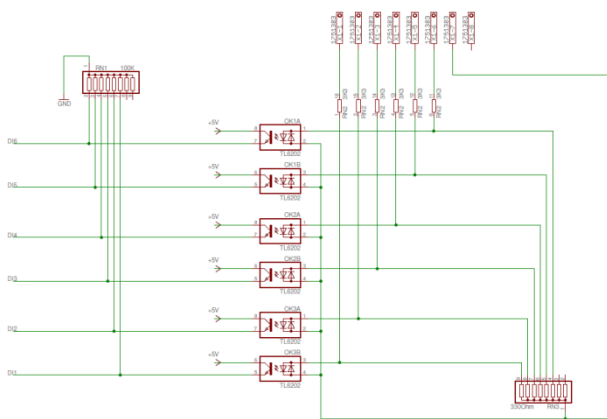


Fig. 7. Circuito de las entradas digitales externas

4.5. Circuito de las salidas digitales

Como búfer de salida de los puertos del microcontrolador, se utilizará el chip ULN2803A. Éste incorpora 8 arrays de transistores Darlington. Como su salida es de tipo NPN, las cargas deberán estar referidas al positivo de la alimentación.

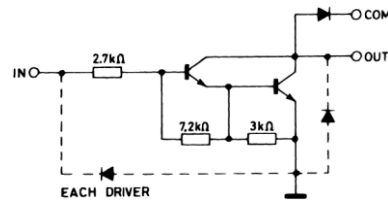


Fig. 8. Esquema interno de un canal del integrado ULN283A

Puesto que dispone de diodo de protección a la salida, la conexión con los relés es directa y no precisa de ningún componente adicional. El relé utilizado es el G5V-2 que dispone de dos contactos conmutados. Los contactos se conectarán en paralelo, con lo que soportarán hasta 4 A a 30 Vdc o 1 A a 125 Vac.

Para la señalización se utilizarán cuatro leds tricolores, aunque dos de ellos sólo lucirán en color verde. Para limitar su corriente se colocarán resistencias en serie con la salida del ULN2803A.

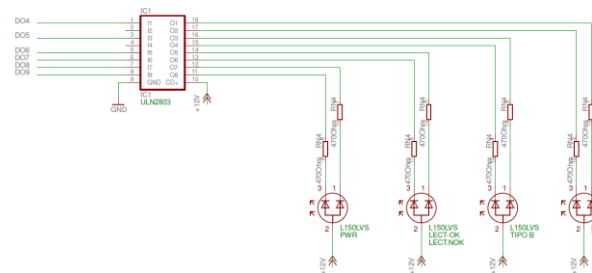


Fig. 9. Circuito de los pilotos de señalización

4.6. Circuitos de alimentación

El módulo Rabbit precisa una alimentación de 3,3 Vdc, los optoacopladores de 5 Vdc y el lector de tarjetas de 12 Vdc. Para la alimentación de 3,3 Vdc se utilizará el regulador lineal LM1117. Éste es capaz de suministrar 800 mA, suficientes para cubrir el consumo estimado de 335 mA. El regulador obtendrá la alimentación de la línea de 5 Vdc, con lo que se reducirán las pérdidas al máximo.

Para optimizar el consumo y reducir las pérdidas térmicas, las alimentaciones de 5 y 12 V se implementarán mediante convertidores Buck, utilizando reguladores integrados.

Para la línea de 5 Vdc se utilizará el regulador LM2575. El consumo estimado en esta línea es de 500 mA y el integrado puede proporcionar hasta 1 A.

En la alimentación de 12 Vdc se utilizará un SI-8120S que soporta hasta 3 A, valor que cubre los 1,2 A requeridos en esta línea.

La entrada de alimentación del equipo cuenta con un diodo de protección contra inversión de polaridad. Se estima que el equipo podrá ser alimentado con una tensión en el rango de 16 a 35 Vdc y tendrá un consumo típico de 8 W y un máximo de 18 W (con el motor de la lectora en marcha).

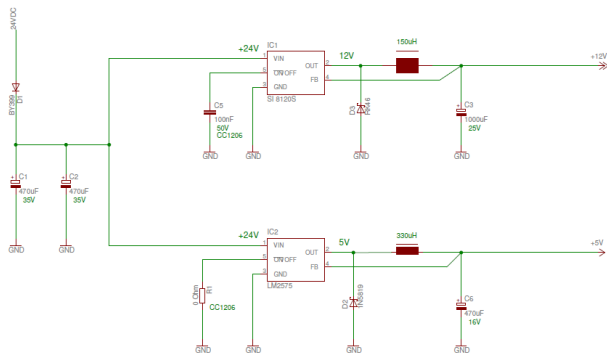


Fig. 10. Fuentes de alimentación de 12 y 5 Vdc

5. Construcción del equipo

El equipo está ubicado en una caja para instrumentación de 96 x 96 x 140 mm. El panel delantero y el posterior están mecanizados para dar salida a los elementos que deben ser accesibles desde el exterior. Adheridos sobre los paneles se han colocado adhesivos con las leyendas de los elementos.

El circuito electrónico tuvo que dividirse en dos circuitos impresos, unidos mediante un cable plano y situados uno sobre el otro. Para separar la potencia del control, en el circuito inferior se alojaron los relés y las fuentes de 12 y 5 voltios.

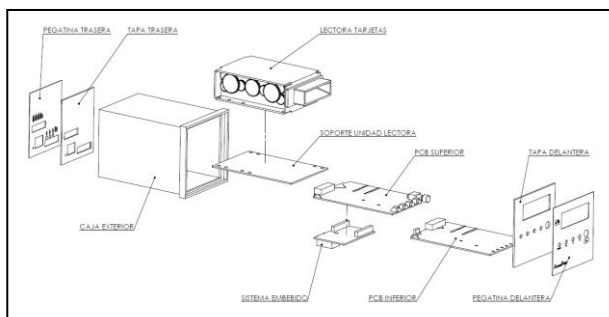


Fig. 11. Vista explosionada del equipo

Para disponer de una plataforma en la que realizar pruebas durante el desarrollo software, se construyó una caja con 6 interruptores y 3 pilotos que se conectan a las entradas y salidas del equipo. Esta caja también suministra los 24 voltios el equipo.



Fig. 12. Plataforma para la realización de pruebas

6. Desarrollo software

Para programar el módulo RCM3200, se necesita un cable de comunicación específico y el entorno de desarrollo Dynamic C [4]. En éste y en Internet existen multitud de ejemplos y librerías cuya utilización está libre de royalties.

A partir de los requerimientos expuestos en el apartado 2, elaboramos una lista de las funcionalidades que debe proporcionar la aplicación a desarrollar.

- Identificación de usuarios mediante tarjetas
- Registro de eventos a partir de entradas digitales
- Control de salidas por relé
- Gestión de leds de señalización
- Administración de la tabla de usuarios
- Edición de los parámetros de configuración
- Consulta del histórico de los eventos registrados

Por el espacio que requiere y por su característica FIFO, el histórico de eventos se alojará en la memoria flash. Los parámetros de configuración y la tabla de usuarios se almacenarán en RAM con batería.

Para resolver la interfaz HMI se implementará un servidor web, accesible a través del puerto Ethernet. Mediante páginas HTML, un ordenador remoto podrá consultar y modificar las tablas de datos contenidas en el equipo.

La aplicación estará compuesta por un programa principal y diversas librerías que le permitirán interactuar con los bloques funcionales definidos en el diagrama siguiente.

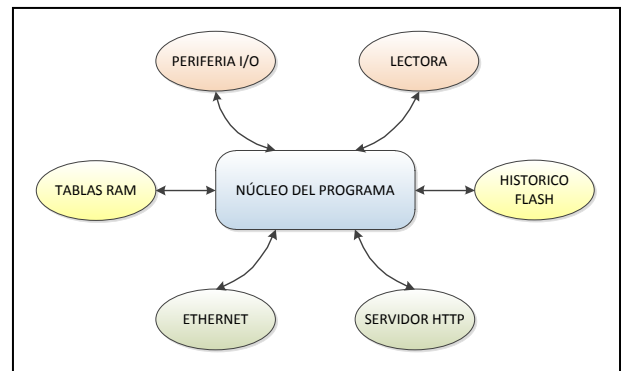


Fig. 13. Diagrama de los bloques funcionales de la aplicación

6.1. Programa principal

Comienza con una inicialización de todos los elementos del sistema y, a continuación, pasa a una sección cíclica. En esta sección, el hilo de ejecución entra en un bucle infinito en el que se realizan llamadas a funciones que, tras ejecutar breves secuencias de instrucciones, siempre devuelven el control al programa principal. Si alguna de estas funciones necesita realizar una secuencia de larga duración, ésta se efectuará implementando una máquina de estados, es decir, mediante una ejecución dividida en etapas. De esta forma, se consigue simular una ejecución casi paralela, con un sólo hilo de ejecución.

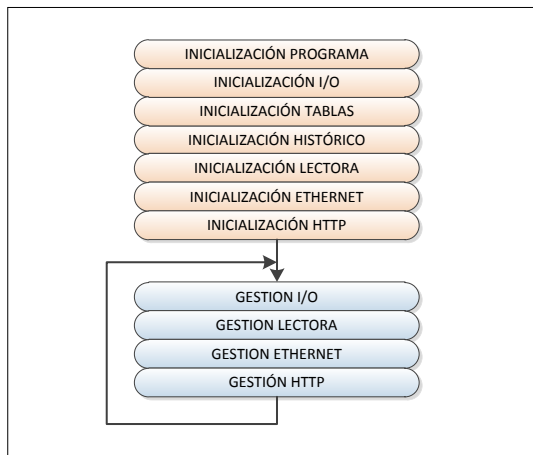


Fig. 14. Diagrama de flujo del programa principal

6.2. Librería de tablas de configuración

En esta librería se definen las variables remanentes de la aplicación. Éstas almacenan parámetros de configuración, acumulados y apuntadores que interesa conservar. También contienen la tabla de usuarios. Durante la inicialización se comprueba si el contenido se ha dañado, en cuyo caso se asignan unos valores por defecto.

Cada tarjeta tiene grabado un número que no se repite. Mediante la tabla de usuarios se establece la relación entre el operario y el número de tarjeta que tiene asignado. Si el usuario extravía su tarjeta, se le proporcionará una nueva con otro número. El nuevo número se pondrá en su ficha, con lo que la anterior tarjeta quedará sin acceso.

La librería proporciona una primitiva que, a partir de un número de tarjeta, localiza el usuario asociado. Cuando se introduce una tarjeta en el lector, se utiliza esta función para identificar al usuario que ha accedido al transportador.

6.3. Librería de histórico de eventos

Proporciona las primitivas públicas de inicialización, grabación y consulta de eventos. Estas primitivas se apoyan, a nivel interno, en funciones auxiliares que resuelven el acceso a la memoria flash y la gestión de la cola FIFO.

La cola FIFO cuenta con un puntero al apunte más antiguo y otro al más reciente. Los punteros se guardan en RAM y en caso de pérdida se inspecciona la memoria flash para recuperar su valor.

La función de grabación de evento añade un apunte al final de la cola e incrementa el puntero final. Puesto que la memoria flash se borra en segmentos de 4KB, antes de realizar la grabación, se comprueba si la dirección destino se corresponde con el inicio de un segmento. En ese caso, se realiza un borrado del sector y, posteriormente, se graba el apunte. Esto proporciona un borrado automático de los apuntes más antiguos, ya que al llegar al final de la memoria se continúa reescribiendo desde el principio. La capacidad máxima es de 16320 apuntes.

La función de consulta de evento recupera el apunte solicitado mediante un índice. Esta función permite acceder al más antiguo, al anterior, al solicitado, al siguiente y al más reciente.

6.4. Librería de periferia I/O

Esta librería contiene las funciones de configuración, acceso, procesamiento y control de las entradas y salidas del equipo.

La función de inicialización se encarga de configurar todos los puertos del módulo microprocesador y, además, de poner a cero el mapa I/O.

La función de ejecución cíclica realiza el intercambio entre las entradas físicas y el mapa de memoria, genera los eventos relacionados con las entradas digitales, gestiona el micropulsador de reset de parámetros, mantiene el valor del cuenta horas asociado a la entrada de “transportador en marcha”, controla la salida para activar el inhibidor de móvil y gestiona el led de estado del equipo.

6.5. Librería del lector de tarjetas

La función cíclica ejecuta las secuencias de inserción y expulsión de tarjeta. Puesto que ciertos pasos requieren esperas prolongadas, el secuenciador se ha implementado mediante una máquina de estados. Ésta coordina el movimiento del motor y la decodificación de la banda magnética

La señal de sincronismo del lector está conectada a una entrada de interrupción. La rutina de interrupción se encarga de ir almacenando la información recibida en un búfer, para que posteriormente sea procesa en el paso de secuencia definido para ello.

Durante el procesamiento del búfer de recepción, se descartan los bytes que están antes y después del carácter de inicio y fin de trama. También se comprueba la integridad mediante el bit de paridad y el LRC. El led de resultado de lectura se actualiza de acuerdo a la comprobación efectuada.

Habitualmente, los datos contenidos en la tarjeta tienen una parte que identifica el ámbito de uso y otra parte que identifica a la tarjeta en sí. El identificador de ámbito se compara con un valor preestablecido y, si ambos coinciden, se toma el número de la tarjeta como dato útil. Este número se localiza en la tabla de usuarios y, en función del tipo, se enciende el led y la salida del relé correspondiente.

6.6. Librería de Ethernet y servidor HTTP

Las librerías del servidor web operan directamente con los servicios Ethernet y TCP, sin que sea necesario intervenir en la capa de comunicaciones. Tan solo se requiere inicializar el puerto Ethernet y llamar cíclicamente a una función que gestiona los servicios TCP/IP.

Al crear las páginas HTML se identifican los botones y los campos del formulario mediante etiquetas. Estas páginas se cargarán en la memoria de programa al compilar. En la función de inicialización del servidor HTTP, se deben vincular los campos de formulario con variables de nuestra aplicación y, a su vez, se deben asociar los botones con funciones del programa.

La función de ejecución cíclica del servidor gestiona todos los servicios relacionados con el envío de páginas HTML y con la recepción de peticiones desde el navegador. El servidor soporta CGI y SSI, lo cual permite añadir contenido dinámico a las páginas HTML. La interacción

entre nuestra aplicación y las páginas HTML, se realiza mediante las variables y las funciones que se vincularon durante la inicialización.

Cuando un navegador solicita una página, el servidor HTTP sustituye todas las etiquetas por el valor de la variable vinculada y, entonces, envía la página resultante. Antes de efectuar las sustituciones, el servidor realiza una llamada a una función asociada en la inicialización. Normalmente, en esta función se actualizan los valores de las variables a mostrar.

La recepción de un formulario enviado desde el navegador es muy similar. A partir de las asociaciones entre etiquetas y variables, el servidor copia los valores recibidos en las variables asociadas y ejecuta una función previamente vinculada que, habitualmente, realiza comprobaciones sobre los datos recibidos

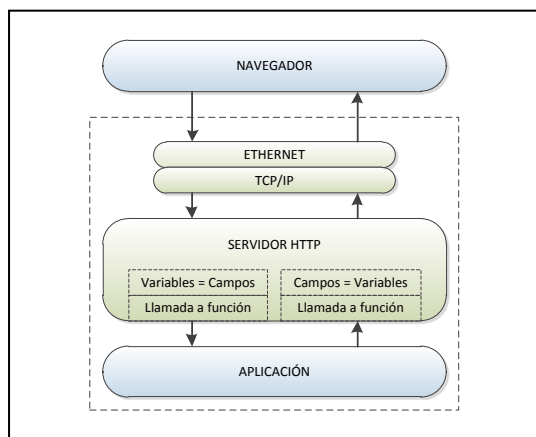


Fig. 15. Diagrama de flujo del servicio web

7. Evaluación del resultado obtenido

Una vez completado el desarrollo del equipo, se verifica que éste cumple con todos los requerimientos planteados en el apartado 2 y, por tanto, se puede concluir que satisface los objetivos del presente proyecto.

Sin embargo, se han detectado ciertas limitaciones que convendría resolver, antes de liberar el equipo para su utilización en un entorno productivo. Existe una limitación importante en cuanto a la cantidad de usuarios que se pueden declarar. Se debería ampliar la tabla de usuarios y mejorar su utilidad de administración. Asimismo, la utilidad para consultar los apuntes del histórico no está ideada para manejar cantidades elevadas de apuntes y, además, no cuenta con herramientas de búsqueda y filtrado.

Como nueva funcionalidad, se propone la implementación de una interfaz M2M a través del puerto Ethernet. Esta interfaz dispondría de comandos que permitirían realizar tareas de administración y consulta de toda la información contenida en el equipo. Gracias a esta interfaz, se podría desarrollar una aplicación para que, desde un ordenador remoto, se dispusiese de potentes utilidades para la edición y consulta de las tablas de datos, para rellenar automáticamente la tabla de usuarios y para almacenar gran cantidad de apuntes, en un fichero histórico ubicado en el ordenador. Si la conexión Ethernet es ininterrumpida, se podría monitorizar, en tiempo real, el uso del transportador y se podrían disparar alarmas en caso de incidencias

consideradas de interés. Finalmente, la aplicación se comunicaría con todos los transportadores existentes y presentaría la información, obtenida de todos ellos, de forma consolidada.

Con la mejora expuesta, se podría eliminar del equipo la utilidad de administración de la tabla de usuarios y la de consulta del histórico, ya que, la aplicación del ordenador proporcionaría mejores herramientas.

Otra mejora, de mucho interés, sería implementar otra interfaz M2M con un equipo GPS externo, dotado de puerto Ethernet. De esta forma, se podría conocer la localización del transportador y se podría incorporar, esa información, en los apuntes registrados en el histórico de eventos. Por tanto, el histórico proporcionaría información de cuándo, dónde y por quién se ha producido un determinado suceso. Analizando esa información, se podrían identificar lugares que disparan frecuentemente la señal de conducción brusca, conocer en qué lugares, y durante cuánto tiempo, ha estado parado el vehículo, determinar los lugares en los que se ha recogido o depositado una carga, etcétera.

Es importante resaltar que todos los cambios propuestos no requieren modificaciones en el diseño hardware del equipo.

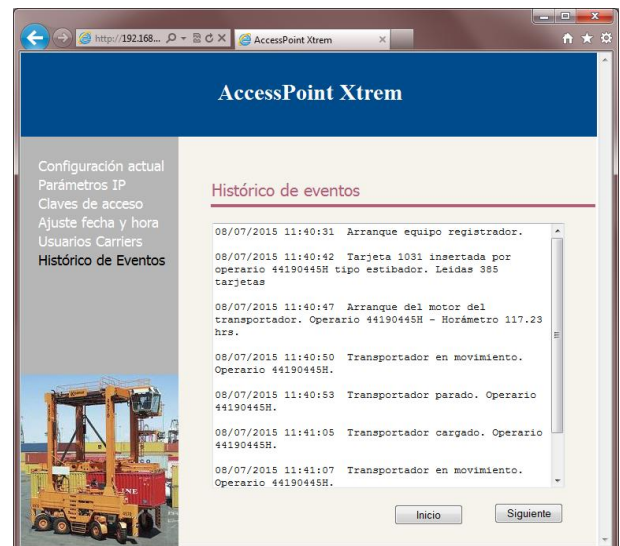


Fig. 16. Utilidad para consultar el histórico de eventos

8. Conclusiones

El equipo resultante ha cubierto las necesidades para las que fue ideado, pero también ha aportado el hecho de disponer de un equipo compacto que incorpora un lector de tarjetas motorizado, entradas, salidas, memoria remanente, puerto Ethernet y un sistema microprocesador programable. Además, se han desarrollado un conjunto de librerías software que resuelven la utilización de todos estos elementos.

Puesto que se trata de un equipo programable, las posibilidades de utilizarlo en la resolución de otras situaciones, son amplísimas. Sin modificar el software, o con pequeñas adaptaciones, este equipo puede ser utilizado para controlar el uso de otro tipo de maquinaria como podrían ser grúas, excavadoras, prensas e, incluso, líneas de fabricación.

Referencias

- [1] A. Utsui, "MTM-7x0T-1R-0101 & MTM-7x0T-1R-0102 Motor drive magnetic card reader", Neuron[®] Corporation, 2003
- [2] Rabbit Semiconductor Inc., "RabbitCore[®] RCM3200 User's Manual", Digi International[®] Inc., 2008. Part Number 019-0118
http://ftp1.digi.com/support/documentation/0190118_n.pdf
- [3] Silicon Storage Technology Inc, "8 Mbit SPI Serial Flash SST25VF080B", Microchip Technology Company, 2011. ISBN 978-1-61341-648-8
<http://ww1.microchip.com/downloads/en/DeviceDoc/25045A.pdf>
- [4] Rabbit[®] Semiconductor Inc., "Dynamic C User's Manual", Digi International[®] Inc., 2011. Part Number 019-0125
<http://ftp1.digi.com/support/documentation/019-0125K.pdf>