

Múltiples Entradas y Salidas en Sistemas Embebidos

Multiple Inputs and Outputs in Embedded Systems

Eduardo Daniel Cohen¹, Esteban Volentini², Matías Giori³

Recibido: Mayo 2015

Aceptado: Julio 2015

Resumen.- El ingreso del estado de múltiples contactos mecánicos a un Sistema Embebido, así como el caso de un número alto de salidas a través de leds indicadores, configuran un desafío importante para los sistemas embebidos, dada la cantidad limitada de puertos de Entrada/Salida (E/S) de los Microcontroladores. Por otra parte, si bien eliminar el rebote mecánico de un contacto cuyo estado debe ingresar al sistema es razonablemente simple, ello se complica cuando se trata de múltiples contactos, cuyos tiempos de estabilización son en general diferentes entre sí. Existen diversos métodos para E/S con múltiples contactos. El objetivo de este trabajo es reunir, analizar y explicar los mismos de manera integrada, para luego proponer una variante novedosa que integra múltiples entradas y salidas empleando tres puertos. Se presentan y discuten dos métodos para evitar los efectos del rebote mecánico de múltiples contactos. Por último se propone un esquema de cableado modular para el caso de que los contactos pudieran estar distribuidos en diversas localizaciones a lo largo de una planta.

Palabras Clave: Entrada/Salida; Conexión Matricial; Charlieplexing; Rebote.

Summary.- *Input of many states from multiple mechanical contacts to an Embedded System, as well as outputs to many leds, pose a challenge for the design of an embedded system due to the small number of Input/Output (I/O) pins that most Microcontrollers have. On the other hand, although it is quite simple to debounce a single mechanical contact, the problem becomes more complex in the case of multiple contacts, which usually have different bouncing times. There are many known I/O methods that deal with multiple contacts. This work aims to explain and analyse them in an integrated manner. It also proposes a different way, which provides the possibility of connecting many inputs and outputs, with only three ports. Two methods to debounce multiple mechanical contacts are discussed. Finally, an innovative modular way of cabling many contacts along a house, or enterprise, is presented.*

Keywords: *Input/Output; Matrix Connection; Charlieplexing; Debouncing.*

1. Introducción.- Cuando se requiere diseñar un sistema embebido con múltiples entradas mecánicas y salidas, por ejemplo un simple sistema de alarma domiciliario, uno de los primeros desafíos que se presenta consiste en leer el estado de múltiples sensores – diseminados a lo largo de una planta.

Por otra parte, si el sistema necesita informar el estado de cada sensor mediante leds. Se requerirá conectar un número grande de salidas, similar al número de entradas.

Es necesario contemplar la manera en que se va a eliminar el efecto rebote de los sensores mecánicos, ya sea contactos fijos que cambian de estado y/o pulsadores. Las llaves y/o

1 Universidad Nacional de Tucumán, Tucumán, Argentina, dcohen@herrera.unt.edu.ar

2 Universidad Nacional de Tucumán, Tucumán, Argentina, evolentini@herrera.unt.edu.ar

3 Universidad Nacional de Tucumán, Tucumán, Argentina, matias.giori@domotech.com.ar

pulsadores no presentan idénticos tiempos de rebote, por el contrario, se verifica una amplia variación en los mismos [1].

El problema del cableado de sensores a lo largo de una superficie razonablemente extendida constituye un desafío que requiere una manera sistemática de atacarlo.

Un objetivo de diseño adicional consiste en que futuras variaciones al sistema de E/S en la cantidad de sensores y leds indicadores, no requieran modificaciones a la aplicación que se ejecuta en el Microcontrolador (MCU). Las mismas pueden ocasionarse por cambios en la misma planta o porque se requiera instalar sistemas similares en otras.

Si bien varios métodos que se presentan son conocidos, la mayoría de ellos se encuentra publicado en sitios web [1] o en libros de texto [2], muchas veces con una descripción superficial o parcial. El presente trabajo incorpora nuevos aspectos que profundizan su análisis, el que se realiza de una manera integral, permitiendo una clara comprensión y profundización. Se proponen nuevas variantes para diseñar un sistema matricial de Entrada/Salida (E/S) como así también para el cableado de las mismas a lo largo de una planta.

2. Métodos de Conexión de Múltiples Entradas.

2.1. Conexión con dos puertos.- La figura I muestra la forma en que se conectan 64 entradas a un MCU empleando únicamente dos puertos, P1 y P2, de E/S.

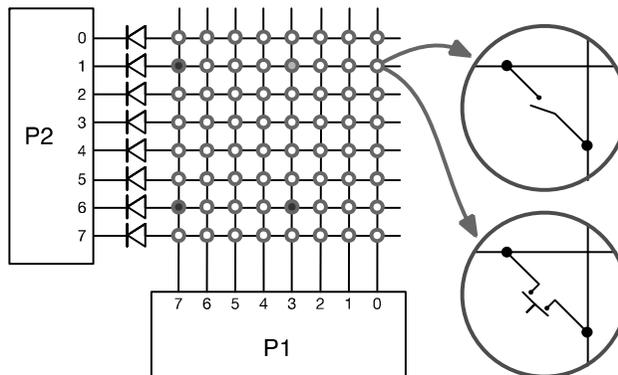


Figura I.- Conexión de 64 entradas a dos puertos, P1 y P2

Para detectar el estado de los contactos, es necesario configurar inicialmente a los puertos P1 y P2 de la siguiente manera:

- P1 como entrada en “pull-up”, es decir, entradas conectadas a fuente por intermedio de una resistencia – en el interior del MCU.
- P2 como salida, todas en ‘1’.

Las salidas de P2 se denominan filas mientras que las entradas en P1, columnas.

Es claro que si a cualquier fila i , F_i , se la configura como salida con el valor “0”, todo contacto activo en cualquier columna “ j ” de la misma, C_j , será leído en P1 como un “0”, mientras que para las otras columnas el valor será “1”. Los pares ordenados salida-entrada (F_i , C_j) con el valor (0,0) identifican unívocamente a los sensores en contacto. A partir de esta observación, surge un

sencillo algoritmo, SCAN, que resulta en una matriz con el estado de todos los contactos. Se trata de una matriz de 8×8 bits, compuesta por 8 bytes, M , donde $M(B)$ es la identificación de un byte correspondiente a una fila de esta matriz, con $0 < B < 7$. Sea C el byte que contiene a los bits que se leen de $P1$.

SCAN

- a. $i=0$
- b. $F_i = 0$
- c. $M(i)=C$
- d. $F_i = 1$
- e. Si $i=7$ terminar
- f. $i=i+1$
- g. Volver a (b)

Los diodos en figura I se colocan con el objeto de evitar un cortocircuito entre dos salidas, F_i y F_j , que se produciría si dos contactos de una misma fila, k , (i,k) y (j,k) , estuvieran cerrados. Los diodos pueden eliminarse con solo configurar inicialmente a todos los pines de $P2$ como entradas sin pull-up, es decir en estado de alta impedancia. El punto (a) de la configuración inicial debe modificarse para que $P1$ quede configurado como entrada sin pull-up. El punto “d” del algoritmo SCAN debe, claro está, cambiarse para que F_i retorne a su configuración inicial.

La conexión presenta el problema de que cuando se verifican más de dos contactos activos, es posible que una configuración de contactos determinada provoque la detección de contactos inexistentes, o “fantasmas”. Por ejemplo, en el caso de figura I, al leer $P1$, para $i=1$ se detectan los pares (F_1, C_7) y (F_1, C_3) como contactos cerrados. Es claro que el estado del segundo contacto no es tal, sino que el valor 0 en C_3 surge de la conexión de (F_1, C_7) que se traslada a través de los contactos (F_6, C_7) y (F_6, C_3) , que es en realidad el que se lee.

En el caso en que todos los contactos fueran pulsadores, tal el caso de un teclado, el problema se podría resolver obligando a que solo un único pulsador sea oprimido para que el sistema registre un cambio de estado. Si el sistema detectara más de un contacto activo, no reaccionaría mientras durara tal situación. Esta solución, claro está, no es de aplicación cuando los contactos son llaves cuyos cambios de estados requieran identificación, puesto que la situación de una única llave en contacto es una excepción y no una regla.

Para solucionar el problema de las lecturas “fantasmas” se requiere conectar diodos en todo contacto, tal como se muestra en figura II, a fin de evitar conducción hacia $(F1,C2)$ como el lector podrá fácilmente comprobar. En este caso, todas las filas podrían configurarse inicialmente ya sea en alta impedancia, o como salidas en ‘1’ sin pull-up (los diodos impiden cortocircuitos entre dos filas distintas, ambas con contactos activos en la misma columna).

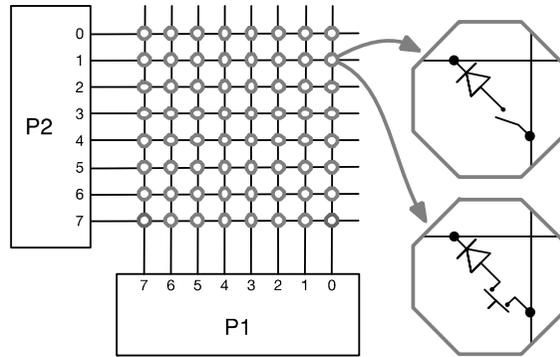


Figura II.- Conexión de Diodos en todo contacto

El método de conexión permite hasta un máximo de 64 entradas – considerando puertos de 8 bits. Si se deseara un número mayor de entradas, es posible extender el número de filas y columnas. Se podría, por ejemplo, incorporar un único puerto para duplicar el número de filas, y lograr así el doble en la cantidad máxima de contactos posibles. Otras alternativas de crecimiento, ya sea en cantidad de filas o columnas, son también viables. Es importante notar que el máximo de contactos se logrará siempre que la ampliación se realice con igual número de filas y columnas. Por ejemplo, una matriz de 12 filas por 12 columnas daría un total de 144 contactos, un 12,5% más que los 128 que se obtendrían de una configuración de 16 filas por 8 columnas – en ambos casos se emplea el mismo número de pines de E/S.

Para posibilitar que el firmware de la aplicación sea invariante a variaciones futuras en la cantidad de contactos, ya sea por nuevos requerimientos o bien a instalaciones con distinta cantidad de contactos, se debe dimensionar la matriz para contener un número máximo que contemple estas alternativas.

En todos los casos solo se conectarán las entradas en uso, permaneciendo el resto de los contactos inactivos, los que al leerse no deben provocar reacción alguna, asegurando por tanto que no generen cambios de estado. Una forma de lograr que los contactos que no están en uso permanezcan invariantes se tratará más adelante, conjuntamente con los métodos para evitar los efectos del rebote mecánico.

La aplicación se debe diseñar para reaccionar al número máximo de contactos, estén en uso o no. Si se supone que los contactos pueden agruparse en conjuntos para los cuales el sistema reacciona de manera similar, se organizará cada grupo de tal forma que queden contactos libres cercanos para su crecimiento futuro. El incremento de contactos solo requiere incorporar los mismos en los lugares que se dejaron libres en la matriz de conexión, sin necesidad de modificar el firmware del sistema.

2.2. Conexión con un único puerto.- El método de conexión con un único puerto, denominado “Charlie Multiplexing” o también “Charlieplexing” [3], permite incorporar hasta 56 conexiones externas a un único puerto, de 8 bits, de E/S de un MCU, en total 8 contactos menos que con el método matricial de dos registros presentado previamente.

Los 56 contactos resultan de conectar cada uno de los 8 pines del puerto con los 7 restantes. El único puerto P, presentado para conectar las columnas, se replica a las filas, mediante la conexión permanente que se muestra en la diagonal ascendente en figura III. Es claro que en todos los cruces se puede conectar contactos, con excepción de la mencionada diagonal.

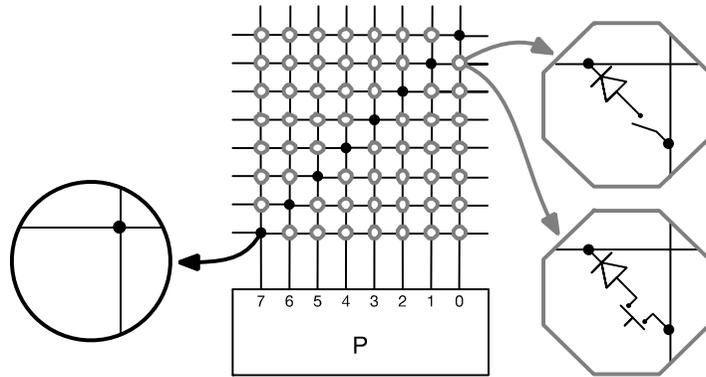


Figura III.- 56 contactos en un único puerto P

El algoritmo para leer la matriz de entrada, SCAN_CM, es muy parecido a SCAN. La lectura se efectúa en una matriz de 8x8 por simplicidad. La diagonal ascendente de la matriz no debe ser considerada por la aplicación.

Cada columna C_i de P genera la fila F_i de la matriz resultante. Si bien siempre se verifica que $C_i=F_i$, esta denominación ayuda a entender mejor el funcionamiento del algoritmo de identificación de los contactos, SCAN_CM, que se presenta a continuación:

SCAN_CM

- a) Configurar P para que sus ocho pines sean entradas con pull-up (valor en 1)
- b) $i = 0$
- c) Configurar F_i como salida, $F_i = 0$
- d) $M(i)=C$
- e) $F_i = 1$ configuración pull-up
- f) Si $i=7$ terminar
- g) $i=i+1$
- h) Volver a (c)

Es claro que si no se instalaran los diodos que se muestran en figura III, en caso de que se cerraran al menos dos contactos de una misma columna, se estaría produciendo una lectura incorrecta. Por ejemplo, el lector podrá comprobar fácilmente que si únicamente los contactos (F2,C3) y (F5, C3) estuvieran cerrados, dado $F5=C5$ por conexión, se generaría una lectura fantasma en (F2, C5) al intentar leer la fila 2.

3. El problema del rebote.- El rebote de un único contacto mecánico, tal como se muestra en figura IV, produce variaciones de tensión. Se requiere evitar la posibilidad de que el sistema interprete sus efectos como sucesivos cambios en la información de entrada.

Los rebotes mecánicos, no presentan en general un tiempo determinado de duración [1]. El mismo depende del tipo, el estado de conservación del switch o pulsador, el circuito al que esté conectado, entre otras variables. Existen diversas alternativas de solución a este problema, los que están ampliamente explicados en la bibliografía [1].

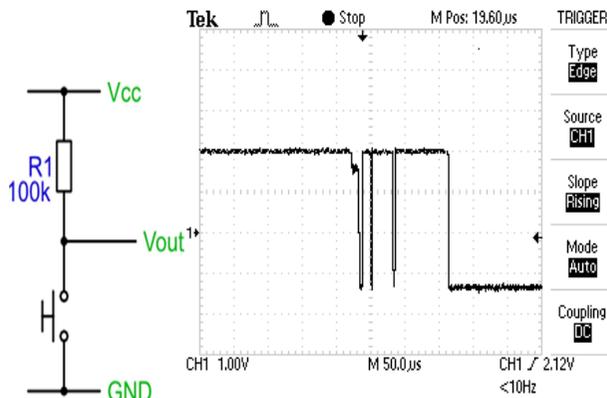


Figura IV.- Rebote Mecánico de un pulsador

Uno de los principales métodos para eliminar el efecto de rebote consiste en asegurar que el valor de entrada se mantenga estable durante un intervalo de tiempo determinado, para recién entonces realizar el ingreso del dato. La ventaja de este método es que se puede implementar por software y es de tipo adaptivo, por lo tanto funcionará correctamente para diversos contactos sin necesidad de determinar de antemano un tiempo máximo de duración del rebote.

Cuando se dispone de múltiples contactos, es necesario contar con un método que pueda resolver el problema de manera global. Al respecto, se presentan a continuación dos soluciones alternativas, que se pueden aplicar a las configuraciones matriciales desarrolladas.

3.1. Detección de múltiples contactos estables.- Para detectar valores estables en múltiples entradas, conectadas o no matricialmente, se identificará a aquellas que no varíen durante un intervalo de tiempo de espera determinado, t_e , asegurando de esta manera que no estén rebotando. El lapso mínimo entre cambios de estado de un contacto, en una aplicación determinada, debe ser superior a la suma del tiempo de rebote, más el de espera, más el que requiera la identificación del contacto y la reacción del sistema. Para tomar un ejemplo, desarrollado para un sistema de alarmas por intrusión, los tiempos entre cambios de estado en sensores son altos. Los menores tiempos tendrán lugar cuando el usuario opere un teclado de activación. Se fija un tiempo de espera de 50 ms. que se considera adecuado tanto para el sistema del ejemplo, como así también para la percepción del usuario al operarlo.

Se remarca la ventaja de que este valor sea independiente del tiempo de rebote, ya que se controla que la entrada se mantenga constante durante un lapso de tiempo, posterior a la finalización del mismo.

3.1.1. Operaciones Matriciales.- A fin de aplicar el método para múltiples contactos dispuestos matricialmente, se tomarán muestras del estado de sus valores en forma periódica. Cada muestra de 64 contactos se representa como una matriz M_i de 64 bits, que contiene los estados de todas las entradas. Esta matriz se acomoda en 8 bytes, con cada fila dispuesta en un byte.

Las muestras se toman continuamente, mediante interrupciones a intervalos regulares, y se almacenan en una cola circular. El sistema debe reaccionar a cambios que se detecten, para lo cual será necesario establecer el valor de las entradas que no rebotan y comparar las mismas con su estado estable anterior, que se supone almacenado en la matriz M_E , para poder determinar si hubo cambios.

Las siguientes ecuaciones matriciales (1) y (2) permiten determinar las entradas estables, durante n muestras. Los valores estables '1' y '0', se registran en las matrices E_1 y E_0 , respectivamente.

$$E_1 = M_0 \wedge M_1 \wedge M_2 \dots M_{n-1} \quad (1)$$

$$E_0 = M_0 \vee M_1 \vee M_2 \dots M_{n-1} \quad (2)$$

Los valores de las entradas estables son aquellas que coinciden en las dos matrices. Aquellos que difieren se encuentran en proceso de cambio y no deben producir reacción.

Las entradas estables se representan por un '1' en la posición correspondiente de la matriz estable M_S , que se obtiene de la siguiente ecuación:

$$M_S = \neg (E_0 \oplus E_1) \quad (3)$$

Las entradas que han cambiado en relación al estado anterior, toman el valor '1' en la matriz de cambios, M_C , que resulta de la siguiente ecuación

$$M_C = (M_E \oplus E_1) \wedge M_S \quad (4)$$

La primera operación XOR resulta en 1 para todo cambio (incluyendo '0's que no son estables). La segunda operación AND pone a cero todo 1 que proviene de un valor inestable '0' en E_1 .

Si se deseara ignorar algunas entradas, por ejemplo la diagonal principal en la conexión Charlieplexing, u otros contactos que no se usen, se define una matriz de percepción, M_P , con un 1 en las entradas a considerar y un 0 en las que se obviarán. La siguiente operación permite enmascarar todos los contactos que no se empleen, poniéndolos a cero.

$$M_C = M_C \wedge M_P \quad (5)$$

Queda únicamente por calcular el nuevo estado, de la siguiente manera:

$$M_E = M_E \oplus M_C \quad (6)$$

La obtención de M_C y M_E son los resultados más importantes, a partir de los cuales se puede generar la información que requiere el sistema para realizar la correspondiente reacción ante cambios en las variables de entrada.

Las operaciones matriciales lógicas requieren muy poco tiempo de procesamiento. Una operación entre dos matrices se realiza mediante 8 operaciones lógicas en un procesador de 8 bits y en 2 operaciones con uno de 32 bits. También es posible calcular en un mismo lazo, las ecuaciones (3) a (6), con lo cual se logrará mayor rapidez aún, evitando también la necesidad de almacenar la matrices intermedias E_0 , E_1 y M_S . En cuanto a las necesidades de memoria, si se considera 8 muestras, se requeriría almacenar un total de 11 matrices, en total 88 Bytes, cantidad prácticamente no significativa en un MCU.

Contemplando la posibilidad de que el valor de 50 ms pudiera no ser apropiado para un sistema determinado, se sugiere que el mismo se mantenga como un parámetro que pueda ser cambiado con facilidad por el desarrollador del firmware. Se debe tener en cuenta que, en el lapso de 50 ms, se debe realizar una toma completa de muestras, por ejemplo 8, lo que requiere igual número de interrupciones. Si el intervalo elegido fuera menor, la cantidad de interrupciones por unidad de tiempo sería mayor, con la posibilidad de resentir la velocidad de reacción del sistema. Por ello es importante definir un valor de compromiso entre las necesidades de entrada de datos y la rapidez del sistema.

En cuanto a la frecuencia de muestreo, se recomienda asegurar que la misma no sea múltiplo ni de 50 Hz ni de 60 Hz, con el objeto de evitar el ingreso de ruido electromagnético, que en gran medida proviene de la red eléctrica y sus armónicos, en cualquier región del mundo en que se instale.

Es importante tener en cuenta que el cálculo de las ecuaciones (1) a (6) podría realizarse en cada interrupción que tome una muestra. Sin embargo, conviene desarrollar un sistema embebido de modo tal que sus interrupciones sean lo más cortas posible. Se recomienda, por tanto, diseñar un módulo de procesamiento de muestras (MPM), que sería convocado por la aplicación principal, para identificar las entradas cuyo valor se ha modificado y así reaccionar acorde a las mismas. El procesamiento puede aligerarse si el módulo en cuestión revisa que la matriz de Cambios (M_C) esté completamente en 0, en cuyo caso no hay nada que para hacer.

Se debe tener especial cuidado de que la interrupción no modifique las muestras mientras MPM se encuentre realizando el cálculo de las ecuaciones (1) y (2). Se trata de una típica relación “productor-consumidor” [4], para la cual existen diversas técnicas que pueden aplicarse, por ejemplo dedicar un semáforo, S_1 , que indique si la cola se encuentra en uso para cálculo o no [5]. También se puede optimizar la sincronización entre interrupciones y módulo de cálculo, mediante un segundo semáforo, S_2 , que indique al módulo si existen muestras a procesar o no.

3.1.2. Contadores.- Una segunda alternativa para identificar los cambios de estado en las variables de entrada se implementa mediante contadores.

Al igual que el método anterior, se realiza un número fijo de interrupciones cada 50 ms, para tomar – en cada una de ellas – una muestra que se guarda en una cola circular.

Se provee un contador por cada variable de entrada, con un total de 64 contadores para una matriz de 8x8.

A fin de determinar qué valor tiene una entrada, se procede a registrar mediante el contador “i”, el número de veces que la entrada “i” tomó el valor ‘1’. Si la cuenta supera el 80% del número total de muestras (NM), se determina que su valor es ‘1’. Los ceros corresponden a aquellas variables cuyos contadores no superen el 20% de NM. Todas aquellas variables cuyos contadores se encuentren entre un 20% y un 80% de NM, no tienen un valor determinado, y se encuentran en transición.

El estado de los contadores se determina con un margen de tolerancia a ruidos o colas de rebote de hasta un 20%. En caso de que el entorno fuera más ruidoso aún, las tolerancias se podrían extender, por ejemplo a un 30%, considerando ‘0’ al estado de aquellos contactos cuya cuenta sea inferior al 30% de NM y ‘1’ a los que superan el 70% de ese valor.

Para determinar qué cambios se produjeron en las variables estables que se detectan, se procede a compararlas con el valor del estado anterior. El sistema reaccionará en función de cada cambio que se detecte.

Al igual que para el método anterior, se recomienda diseñar un MPM que será convocado por la aplicación principal. Asimismo, son de aplicación las consideraciones para sincronizar MPM con la rutina de interrupción a fin de asegurar que las muestras permanezcan inalteradas mientras se las procesa como así también para evitar procesamiento cuando no existan nuevas muestras.

El método de contadores emplea más tiempo de procesamiento. Ello puede provocar que el sistema reaccione más lentamente, lo que podría representar un problema dependiendo del

sistema embebido en cuestión. Por otra parte, el filtrado mejora con mayor cantidad de muestras, lo que requiere también mayor carga de procesamiento.

En contraposición a esta desventaja, este sistema trabaja muy bien en ambientes ruidosos, en los cuales es posible que el sistema con ecuaciones lógicas no funcione pues los ruidos impedirían lograr una señal estable, idéntica en todas las muestras que se procesan.

4. Conexión de Múltiples Salidas (Indicadores Leds).

4.1 Conexión Matricial.- Considérese el caso de que se requiera indicar el estado de múltiples entradas mediante el encendido o no de sus correspondientes leds indicadores.

La conexión de leds se realiza de manera matricial, empleando dos puertos del MCU, tal como se muestra en figura V.

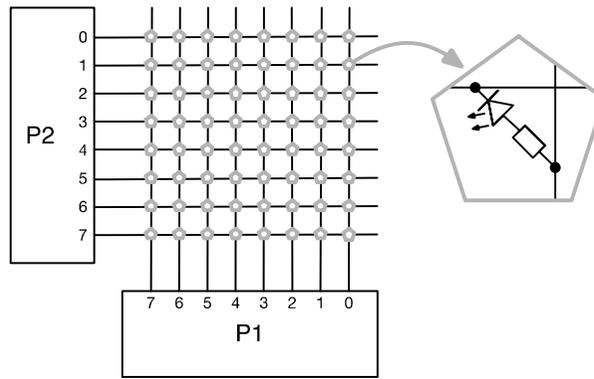


Figura V.- Conexión de hasta 64 leds.

La conexión funciona multiplexando las salidas de cada fila en el tiempo. En realidad, cada led encendido recibe un pulso periódico, debido a la persistencia óptica se percibe el valor medio de esta onda, siempre que su frecuencia supere un mínimo determinado (al menos 25 Hz). Se elige una frecuencia de trabajo (FT) superior a la mínima.

La frecuencia de las interrupciones surge de multiplicar FT por la cantidad de filas. Cada interrupción refresca una fila completa. Para ello el pin F_i de P2, correspondiente a la fila i que se refresca en una interrupción, se configura como salida en '0', el resto de las filas se colocan como salida en '1' o bien en estado de alta impedancia. Las columnas de los leds a encender se configuran en '1' sin pull up, el resto en '0' o alta impedancia.

Por ejemplo si se refrescara la fila 7, $F_7=0$, en la cual se deben encender los leds correspondientes a las intersecciones (7,1) y (7,3), las columnas deben tomar el estado $C_1=1$ y $C_3=1$, permaneciendo el resto de ellas en '0' o alta impedancia.

Se hace notar que la interrupción es muy corta y consume muy poco tiempo de procesamiento. Para que los leds reflejen el estado de las entradas, bastará con enviar los valores de la fila en cuestión de la matriz de estado M_E al puerto P1. Todo valor '1' en la fila de M_E se representará por un led encendido en la posición correspondiente.

A fin de que las filas sean capaces de soportar el máximo valor de consumo, que se produce cuando se encienden sus 8 diodos, deben contar, cada una de ellas, con un transistor que amplifica la salida, tal como se muestra en la figura VI, y una resistencia que asegure que no se supere la máxima corriente que permite el diodo. Considerando que el transistor invierte la señal,

será necesario invertir los valores a la salida de P2 cuando se produce el multiplexado de las filas de salida.

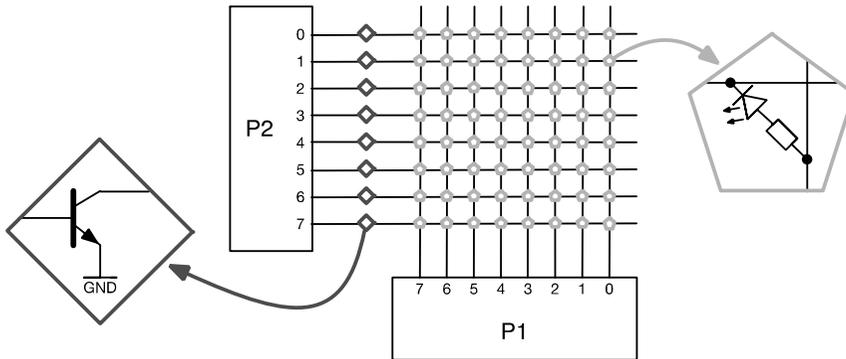


Figura VI.- Amplificación en las Filas.

4.2. Conexión con único puerto.- La conexión con un único puerto se denomina “Charlie Multiplexing” o “Charlieplexing” y es similar a la conexión del apartado (2.2), pero para salidas. Al igual que en el caso de salida matricial con dos puertos, para asegurar la persistencia óptica es necesario refrescar todas las filas, con una frecuencia de al menos 25 Hz.

Las filas se generan a partir de las columnas conectadas, vía transistor, al único puerto P, tal como se muestra en figura VII. Si bien $F_i=C_i$, la denominación de filas y columnas permite entender el funcionamiento por similitud al caso de una matriz con dos puertos. Los transistores aseguran que cada fila F_i posea la capacidad de encender los 7 diodos conectados a la misma. Las salidas se toman del emisor para asegurar alta impedancia cuando su entrada sea ‘0’.

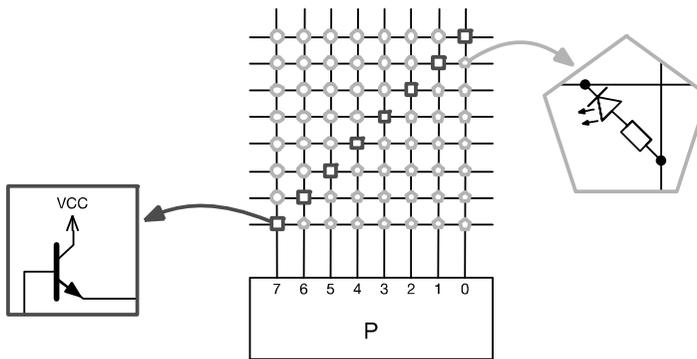


Figura VII.- Amplificación en Filas.

Se hace notar que en este caso la configuración del transistor no invierte la señal. Para refrescar una fila, la misma se configura como salida en el valor ‘0’ y las columnas correspondientes a los leds a encender, como salida en ‘1’, el resto de los pines deben permanecer en estado de alta impedancia.

Tanto la explicación de funcionamiento como la determinación de la frecuencia de interrupciones es similar al presentado en el caso de una matriz de leds con dos puertos.

5. Conexión de múltiples entradas y salidas de leds.- En figura VIII se presenta un esquema de conexión que emplea tres puertos, de 8 bits c/u, para un máximo de 64 entradas y 64 salidas de

leds. El sistema trabaja como una combinación de los sistemas de entrada y de salida explicados en los apartados (2.1) y (4.1), solo que comparten el puerto de las columnas.

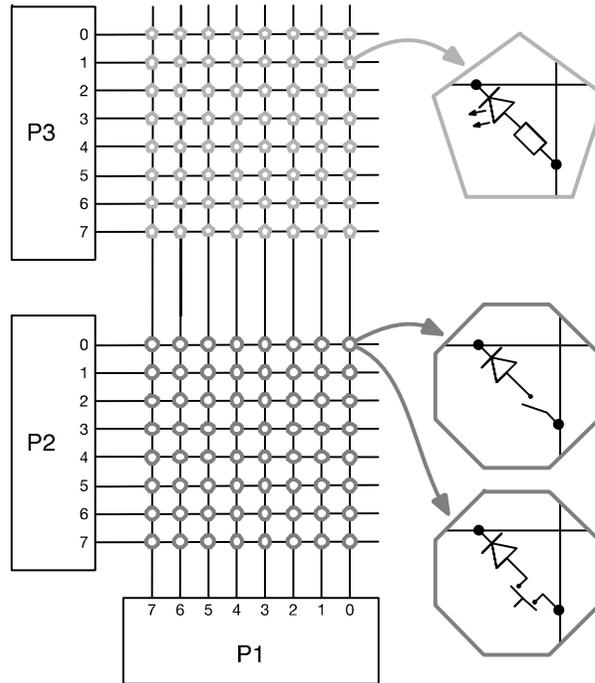


Figura VIII.- Esquema de conexión de E/S con 3 puertos.

El precio que se paga para ahorrar un puerto consiste en que cada vez que se realiza una interrupción para toma de muestras, en la matriz inferior, debe desactivarse la matriz superior, y por tanto disminuye el valor medio de la onda que refresca la correspondiente fila de leds.

Con el objeto de mantener el mismo valor medio de onda para cada fila de leds, la frecuencia de interrupciones para muestreo de la entrada debe ser un múltiplo, N , de la frecuencia de refrescado de los leds, ya que, como ya se señaló, durante la toma de muestras debe interrumpirse la alimentación de una fila de leds. Por idéntica razón, cada operación de muestreo debe durar un tiempo similar. En este sistema coexisten dos tipos de tareas periódicas: las de refrescado de leds y las de muestreo del estado de las entradas. Por simplicidad, conviene diseñar una única interrupción que se encargue de ambas funciones, ejecutando el refresco de una fila de leds luego de N tomas de muestras. Las interrupciones deben realizarse con una frecuencia fija en el tiempo, para asegurar ondas de refresco periódicas en los leds. En caso de que alguna de estas recomendaciones no se cumpla, las filas de leds presentarán un brillo variable.

Para que el valor medio de onda no decaiga demasiado, las interrupciones para muestreo de entradas deben durar lo menos posible. En tal sentido, es de aplicación lo señalado en el apartado (3.1.1), en cuanto a que el procesamiento de las muestras no debe realizarse durante las interrupciones, sino mediante un módulo del programa principal, cuya prioridad sea inferior a las interrupciones de muestreo.

Para el caso de ambientes muy ruidosos, que hagan necesario emplear el método de contadores, cada interrupción de muestreo debe cortar la alimentación de una fila de leds solo el tiempo que lleve tomar una muestra de las entradas en una matriz.

Si el ruido no es un problema, siempre será conveniente realizar el procesamiento mediante operaciones lógicas entre matrices, para lograr una menor carga de procesamiento.

6. Consideraciones de cableado. El ingreso de múltiples entradas a partir de contactos que pueden estar distribuidos en una planta determinada, requiere el diseño de un sistema de cableado para conectar los mismos al MCU.

Para lograr una identificación de cada sensor resulta necesario instalar un par de conductores por cada contacto. Una opción muy difundida consiste en emplear un cable multifilar. Una primera alternativa sería emplear un multifilar de 128 hilos, que distribuya tanto entradas como salidas. Esta opción resulta cara y dificultosa para instalar y mantener – por la cantidad de hilos.

Para minimizar el número de conductores que deben cablearse a contactos concentrados en un área remota de la planta, los mismos deberían conectarse a sub-áreas cuadradas de la matriz de entrada. Por ejemplo, para el caso de 16 contactos, en lugar un tendido a partir de dos filas con 8 columnas, con un total de 10 cables, se llevaría los conductores de una sub-área de 4 filas por 4 columnas, resultando en un ahorro de 25% en su cantidad.

Siguiendo con la idea anterior, un buen compromiso costo-beneficio podría ser partir la matriz de 8x8 en 4 matrices de 4x4, resultando así en 4 cables de 8 conductores c/u que podrían distribuirse en distintas áreas de la planta.

Se sugiere, en consecuencia, emplear cables normalizados con menor número de conductores, 8 en este ejemplo, y conectar los sensores (pulsadores o switches) mediante cajas de conexión modulares.

Una caja de conexión modular se muestra en figura IX. Presenta un conector del cable multifilar, tanto de entrada como de salida. En su interior, una placa permite seleccionar al par que se conectará al contacto. Ello se realiza mediante una operación de configuración – mediante un equipo de muy sencillo diseño - consistente en quemar todos los fusibles que no correspondan al contacto en cuestión.

El equipo de configuración se puede diseñar para realizar también la prueba de una caja, logrando así saber si la misma funciona correctamente y cuál es el par de conductores asignados al contacto a conectar. También es posible diseñar una rutina de testeo como parte del sistema instalado, brindando así la capacidad de autodiagnóstico.

Estas cajas, por ser todas iguales, se pueden producir en escala, minimizando así las cantidades requeridas en stock, obteniendo un costo menor, el que seguramente compensará aquellos de instalación y mantenimiento de alternativas manuales que requieren soldaduras. Cables multifilares de igual cantidad de hilos también permiten menores costos.

Los diodos, de la conexión matricial, se instalan en el interior de las cajas modulares de contactos, con lo cual no se requiere instalarlas en la placa del MCU.

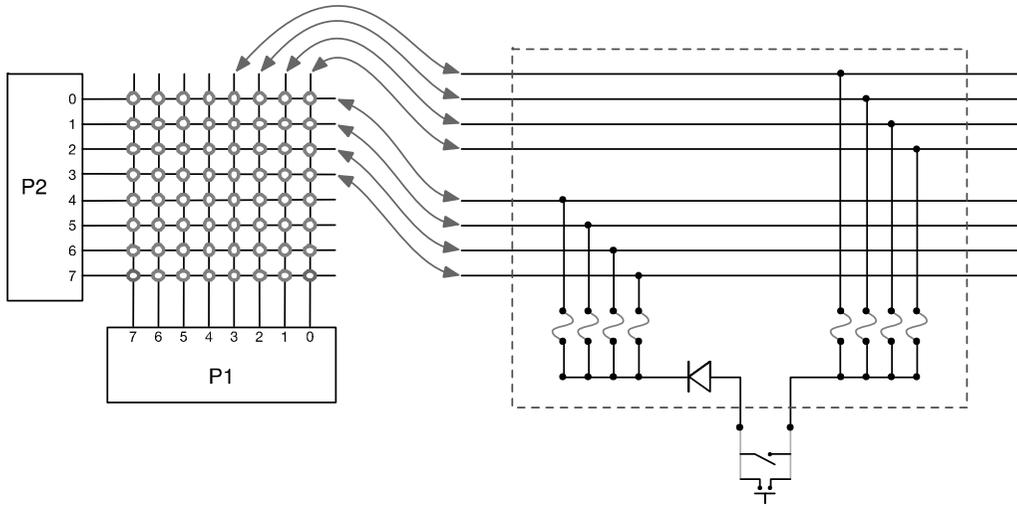


Figura IX: Esquema de una caja de conexión para un contacto de entrada.

Existen otras alternativas de cableado en árbol, por ejemplo mediante la instalación de concentradores activos, dotados de un MCU, dedicados a tomar los contactos de una zona determinada y enviar los mismos a un concentrador más cercano al sistema central o directamente al mismo.

La conexión entre concentradores, o entre concentradores y el sistema central se realiza de manera serial, no siendo de aplicación el sistema matricial.

La alternativa anterior muestra que los sistemas de E/S pueden variar de instalación en instalación o bien en nuevas versiones del sistema central. También es claro que, para el caso de conexiones matriciales, se verifican distintos criterios para la ubicación de las entradas correspondientes a los sensores: por funcionalidad, como se explica en el apartado (2.1), o para economizar cableado.

Ante estas modificaciones y requerimientos dispares para E/S, se plantea como objetivo que el diseño de la aplicación principal sea capaz de contemplarlos sin requerir cambios. Para obtener este resultado, surge la necesidad de lograr que el sistema de E/S sea transparente a la aplicación. En otras palabras, la visión lógica de entradas y salidas debe ser invariante a las mismas. A tal efecto, se debe desarrollar un driver para adaptar cada configuración de E/S a esta visión lógica, el que se expondrá en un próximo artículo.

7. Conclusiones.- En cuanto a los dos métodos de entrada de múltiples contactos presentados, la conexión “Charlieplexing” trabaja con el mínimo número de puertos, permitiendo un número máximo de 56 entradas – 8 menos que con el método matricial tradicional para el caso de puertos de 8 bits.

Se presentaron dos formas de conexión de múltiples leds indicadores. En este caso el análisis es similar al párrafo anterior.

En el caso de que se requiera la entrada y salida de un número alto de contactos y leds indicadores, podría emplearse una conexión matricial para la entrada y otra para la salida. Ello requiere 4 puertos para 64 entradas y 64 salidas o bien dos puertos para 56 entradas y 56 salidas. Se propuso y analizó una conexión intermedia que para 64 entradas y 64 salidas, emplea 3

puertos. La última propuesta presenta un buen compromiso entre complejidad y uso de puertos, con puertos de 8 bits, para un máximo de 64 conexiones de E/S, si bien exige ser cuidadoso en el diseño del firmware para mantener equilibrado el valor medio de alimentación de las salidas.

Se analizaron dos propuestas para solucionar el problema de rebote en caso de múltiples entradas. Ambas propuestas se caracterizan por ser de tipo “adaptivas”, con lo cual la respuesta se obtiene independientemente de los tiempos de rebote de los contactos individuales, que podrían ser muy diferentes entre sí, y sin penalizar a los más rápidos en función de los más lentos. Una de las alternativas de diseño emplea el filtrado estadístico de ruido mediante contadores. Constituye la elección de rigor en casos de ambientes muy ruidosos. En el resto de los casos, se sugiere el empleo de la alternativa que usa ecuaciones lógicas, debido a su menor carga de procesamiento.

Con el objetivo de mantener el firmware de la aplicación principal invariante ante cambios futuros en la cantidad de contactos de E/S, se propuso dimensionar la cantidad de conexiones para un máximo que cubra todas las instalaciones posibles del sistema, incluyendo crecimiento futuro por necesidades en cada una de ellas. La aplicación deberá diseñarse para reaccionar ante este número máximo, asegurando así su invariancia.

La extensión del objetivo previo, a cambios futuros en la estructura del sistema de E/S, o a requerimientos contrapuestos de diseño, introduce directamente en el concepto de drivers, lo cual se desarrollará en un próximo artículo.

La cantidad de aplicaciones de Sistemas Embebidos crece aceleradamente, se emplean métodos que quedan implícitos en las soluciones y muchos de ellos no se discuten apropiadamente o se publican solo parcialmente. Los autores entienden que si bien las aplicaciones son importantes, tanto o más importantes son las metodologías e ideas presentes en las mismas y por tanto este trabajo avanza en esa dirección.

7. Referencias

- [1] Ganssle, J., *A Guide to Debouncing*, <http://www.ganssle.com/debouncing.htm>, Ganssle Group Website, 2014.
- [2] Verle, M., *PIC Microcontrollers*, mikroElektronika, 2008.
- [3] Gupta, S., and Dhananjay V, *Multiplexing technique yields a reduced-pin-count LED display*. EDN Network, 2009. 2009-06-25, 2009: p. 58.
- [4] Valvano, J. W., *Embedded Systems: Real-Time Operating Systems for ARM Cortex-M Microcontrollers*, CreateSpace Independent Publishing Platform, 2014.
- [5] Tanenbaum, H. B., *Modern Operating Systems (fourth edition)*, Prentice Hall Publishers, 2014