

Informe de Resultados

Proyecto de Innovación Tecnológica

Proyecto ANR 2001

FONTAR-Agencia de Promoción Científica y Tecnológica.

***"Incorporación de la tecnología de procesadores especializados
en tratamiento de señal (Digital Signal Processors, DSP)
a una línea de detectores de metal industriales"***

Índice

1. Diseño de Hardware y Algoritmos

- 1.1 Consideraciones sobre el hardware.....
- 1.2 Consideraciones sobre los algoritmos.....
- 1.3 Algoritmo de detección.....
- 1.4 Pasos del algoritmo de detección.....

2. Programa principal detector de Metales

Línea Penta FLEX-DSP

- 2.1 Bloque 1. Inicialización.....
 - 2.1.1 Tipos de Interrupciones y Servicios.....
- 2.2 Bloque 2. Adquisición y Procesamiento de Datos.....
 - 2.2.1 Algoritmo del Timer C0.....
- 2.3 Bloque 3. Procesamiento de Buffers Circulares y Buffers Display.....
- 2.4 Bloque 4. Procesamiento de Modos de Operación.....
 - 2.4.1. Modo 0. Presetup.....
 - 2.4.2. Modo 1. Entrenamiento Producto.....
 - 2.4.3. Modo 2. Entrenamiento Probeta.....
 - 2.4.4. Modo 3. Entrenamiento Umbrales.....
 - 2.4.5. Modo 4. Detección.....
- 2.5 Bloque 5. Atención y Procesamiento de Comandos Externos.....
 - 2.5.1 Comandos de Debug (Procesamiento con Interfaz Matlab).....
 - 2.5.2 Comandos de Operación.....
 - 2.5.3. Comandos de Control de Hardware.....

3. Apéndices

- 3.1 Apéndice A. Algoritmo de inicialización.....**
- 3.2 Apéndice B. Algoritmo de adquisición y procesamiento de datos.....**
 - Apéndice B.1 Verificación de presencia de producto.....**
 - Apéndice B.2 Control de ganancia.....**
- 3.3 Apéndice C. Algoritmo Habilitación Proceso de Modos y Arma buffer Display.....**
- 3.4 Apéndice D. Algoritmos de Procesamiento Modos de Operación.....**
 - Apéndice D.1 Algoritmos Modo 0.....**
 - Apéndice D.2 Algoritmos Modo 1.....**
 - Apéndice D.3 Algoritmos Modo 2.....**
 - Apéndice D.4 Algoritmos Modo 3.....**
 - Apéndice D.5 Algoritmos Modo 4.....**
- 3.5 Apéndice E. Atención y Procesamiento de Comandos.....**
 - Apéndice E.1 Comandos de Debug (Procesamiento con Interfaz Matlab).....**
 - Apéndice E.2 Comandos de Operación.....**
 - Apéndice E.3 Comandos de Control de Hardware.....**
 - Apéndice E.4 Servicio de Interrupción puertos SCI0 y SCI1.....**

1. Diseño de Hardware y Algoritmos

De acuerdo a los trabajos de investigación y desarrollo llevados a cabo en este proyecto en conjunto con la empresa Penta S.A. y los ensayos con diferentes muestras de envoltorios y los efectos producidos por estos en los detectores que la Penta S.A. comercializa actualmente hemos arribado a las conclusiones de los incisos 1.1 y 1.2 .

1.1 Consideraciones sobre el Hardware

El hardware de procesamiento de señales mas apropiado para la tarea a realizar, tanto por sus prestaciones como el software desarrollo y precio, resultó ser el microprocesador de Motorola DSP-56F807.

La motivación principal para su elección, y el descarte de la opción inicial (el Motorola DSP56824), fue su alto nivel de integración de dispositivos: puertos de comunicaciones serie síncronos y asíncronos, conversores AD de múltiples entradas, salidas dedicadas con funciones PWM, suficientes líneas entrada/salida multipropósito, memoria RAM, memoria ROM y flash EPROM.

1.2 Consideraciones sobre los algoritmos

El mejor algoritmo para la detección de partículas metálicas dentro de envoltorios con aluminio en función de las señales analógicas disponibles resultó ser, a nuestro criterio, una combinación de diferentes técnicas de filtrado que incluyen: el filtrado por filtro acoplado (correlación), filtrado pasabajos (promedio) y resta predictiva con entrenamiento continuo (cancelamiento efecto producto). Los filtrados selectivos en la banda de la señal fueron estudiados y posteriormente descartados pues distorsionaban, por problemas de resolución de palabra del procesador DSP, la señal en los niveles pequeños donde se manifiestan los efectos que queremos resaltar (resolución señal probeta- detección por correlación).

1.3 Algoritmo de Detección

Básicamente el algoritmo diseñado para la detección consta de:

1^{er}) Una etapa que realiza una disminución del “efecto producto” mediante una resta entre el “efecto producto” que pasa por el detector con una muestra del mismo producto promediada estadísticamente adquirida durante una etapa de entrenamiento.

2^{do}) La etapa de detección propiamente dicha que realiza la detección por dos métodos: umbral de amplitud y umbral de correlación, ambos calculados sobre las señal resultado de la resta “efecto producto” pasante y el promedio del “efecto producto” en memoria.

1.4 Pasos del algoritmo de detección.

Los pasos para la detección son los siguientes:

1) El efecto de inducción que genera un producto con envoltorio de aluminio o del tipo aluminizado es medido estadísticamente (se toman varias muestras del mismo producto) por el detector y se guarda una versión del promedio de muestras en memoria.

2) Luego se guarda la señal que genera una probeta tipo del metal que deseamos detectar (“efecto probeta”). Al igual que en el paso anterior se guarda en memoria una versión promedio.

El caso óptimo en el sentido de detección se obtiene pasando el producto en forma transversal con la probeta en el centro geométrico del producto.

Debido a los efectos no lineales de la suma entre el “efecto probeta” y el “efecto producto” (la inducción generada por uno y otro difiere en función del tamaño), y como deseamos saber de la existencia señal “efecto probeta” dentro de la señal “efecto producto”, la señal “efecto probeta” que se guarda en memoria es el resultado de la medición que obtenemos al pasar el producto en conjunto con la probeta en cuestión con el detector cancelando el “efecto producto”.

Resumiendo, la señal que denominamos “efecto probeta” es el resultado de restar el “efecto producto” al “efecto producto” que se genera pasándolo en conjunto con una probeta.

Esta señal difiere de la que se genera al pasar una probeta sola por el detector pero es la señal óptima en términos de detección por correlación.

3) Una vez adquiridas las dos señales anteriores (“efecto producto” y “efecto probeta”) el detector debe calcular los umbrales de amplitud y correlación.

Estos se calculan pasando el producto varias veces por el detector y realizando operaciones sobre la señal resultado de la resta producto pasante con “efecto producto” promedio guardado en memoria.

3.1) Los umbrales de amplitud son los valores máximos y mínimos encontrados a lo largo de la señal “resta efecto producto”. Para obtener los umbrales de amplitud en la “resta efecto producto”, la misma, es dividida en bloque obteniéndose un umbral de amplitud por bloque. Cada umbral por bloque es el resultado de promediar las muestras dentro del bloque para minimizar los efectos del ruido.

Debido a que en cada bloque se buscan máximos y mínimos lo correcto es hablar de bandas de umbrales por amplitud.

El largo del bloque es un compromiso entre la velocidad de muestreo (cantidad de muestras por producto) y la resolución deseada en la detección por amplitud.

3.2) Los umbrales de correlación se calculan efectuando la operación correlación cruzada entre la señal “resta efecto producto” y la señal “efecto probeta” guardada en memoria durante el entrenamiento.

La cantidad de umbrales de correlación depende del compromiso entre la velocidad de muestreo y la resolución de detección deseada.

A diferencia de los umbrales de amplitud aquí solo se calculan máximos. Estos máximos son valores discretos tomados de la curva de correlación cruzada.

El hecho de haber elegido como umbral valores de la curva de correlación cruzada entre “resta efecto producto”-“efecto probeta” en lugar de la correlación “efecto probeta”-“efecto probeta”, se debe, a que “resta efecto producto” contiene componentes en frecuencia similares a “efecto probeta” por lo tanto la forma de distinguir si los máximos encontrados corresponden al producto o a la probeta es calculando el máximo valor de correlación cruzada de este modo cuando superamos el umbral estamos seguros de que se trata de un efecto tipo probeta (contaminación).

4) Con los pasos previos completados el detector esta en condiciones de trabajar como detector de metales propiamente dicho.

La detección se realiza procesando la señal “resta efecto producto”, es decir, cada vez que pasa un producto por el detector se le va restando el bloque correspondiente a la señal “efecto producto” guardada en memoria y se calculan valores de amplitud y correlación para ser comparados con umbrales calculados en pasos anteriores.

El algoritmo procesa esta señal por bloques. El tamaño del bloque es el mismo con el que se calcularon los umbrales de amplitud y correlación. La decisión de trabajar por bloque se tomo principalmente en función de la limitación en tiempo para procesar la correlación. Otro aspecto tenido en cuenta en esta decisión es que al trabajar por bloques tenemos toda la información necesaria ni bien el producto sale de la ventana de detección. Esto permite reducir el largo de la cinta, pudiendo eyectar antes un producto contaminado.

2. Programa principal detector de Metales Línea PENTA *FLEX*-DSP

EL programa principal del detector de metales PENTA *FLEX*-DSP se encarga principalmente de digitalizar y procesar las señales analógicas, (asociadas a las perturbaciones en las bobinas del detector), con los algoritmos mencionados en el inciso anterior. De esta manera se pretende extraer la información de detección deseada.

Además, administra y coordina los recursos internos y externos del microprocesador Motorola DSP 56F807, para que este pueda interactuar con los dispositivos externos como Teclado, Display, Internet, Pistones Actuadores, Luces Indicadoras, etc.

Las señales analógicas que digitaliza y procesa son dos: una es una versión amplificada en fase de la perturbación generada en las bobinas del detector y la otra es una versión en cuadratura respecto de la anterior. Ambas señales forman un marco de referencia cartesiano donde la relación módulo-fase entre ambas esta íntimamente relacionada con las características físicas del metal (tamaño y tipo material).

Dichas señales analógicas ingresan al procesador DSP por los puertos internos de conversión AD que posee. Una vez digitalizadas, a una velocidad de muestreo relacionada con la velocidad de cinta y resolución de detección, estas señales serán denominadas “Canal Modulo” y “Canal Fase”.

Las secciones siguientes describen detalladamente la operación del programa principal y los modos de operación y funciones que este implementa.

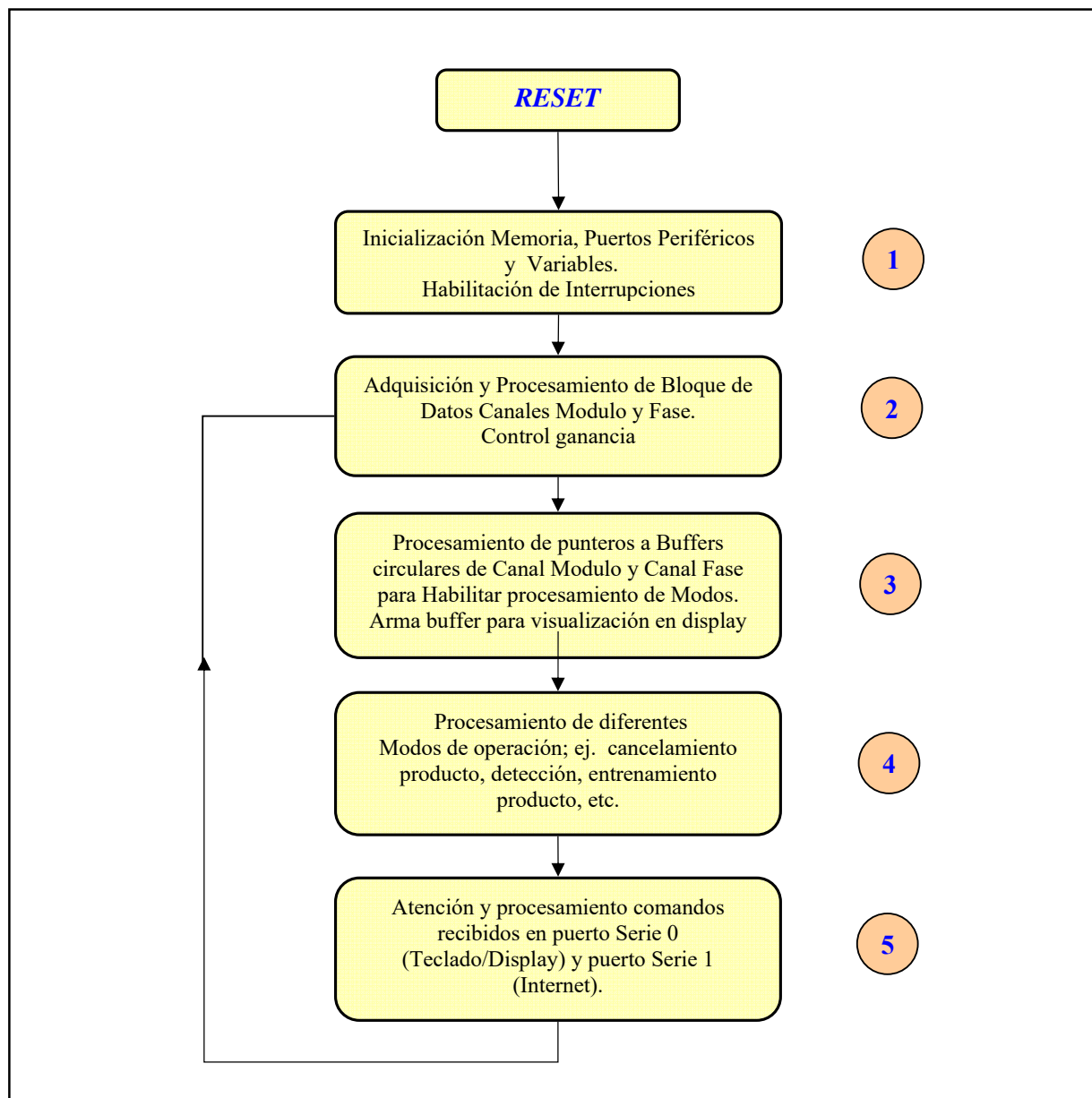


Figura 1. Programa Principal

Para su mejor comprensión al programa principal lo dividiremos en cinco bloques (Figura 1). Inicialización, Adquisición y Procesamiento de Datos, Procesamiento de Buffers Circulares y Buffers Display, Procesamiento de Diferentes Modos de Operación y por Último Atención y Procesamiento de Comandos Externos.

2.1 Bloque 1. Inicialización.

Una vez que el ciclo comienza luego del RESET del microprocesador Motorola DSP 56F807, el **Bloque 1** se encarga de limpiar la memoria RAM, inicializar variables, inicializar los diferentes puertos y configurar la líneas multipropósito de entrada/salida acorde a las funciones requeridas.

Terminado este proceso asigna las prioridades a los diferentes servicios de interrupción y por último habilita las interrupciones.

A continuación creemos conveniente hacer hincapié en el tipo de interrupciones que atiende y que servicios de interrupción ejecuta cada una.

2.1.1 Tipos de Interrupciones y Servicios

Las interrupciones que básicamente atiende el programa son cuatro:

- 1) Timer C0. El periodo del mismo determina la velocidad de muestreo del sistema. La rutina que atiende este servicio de interrupción principalmente se encarga de adquirir las muestras de los canales Modulo y Fase y procesarlas. Otras funciones son efectuadas pero se verán en detalle en la descripción del bloque 2.
- 2) Timer D0-D1. Estos dos timers en conjunto se utilizan para el temporizado de la función `pre_setup()` durante la fase previa al entrenamiento del producto. El valor de estos timers dependerá principalmente del retardo de los amplificadores en alcanzar la estabilidad una vez cambiados sus valores de ganancia.
- 3) Timer D2. El periodo del mismo es de 60 segundos. La rutina que atiende esta interrupción se encarga de verificar los niveles normales de alimentación y controlar la estabilidad de la relación de fase entre amplificadores Modulo y Fase.
- 4) Buffer de recepción lleno puerto SCI0. La rutina que atiende el servicio de interrupción se encarga de adquirir datos provenientes del Teclado/Display, chequear la integridad del dato recibido, verificar las reglas del protocolo de comunicación y generar variables de estado (flags) acorde a los resultados de las operaciones mencionadas anteriormente.
- 5) Buffer de recepción lleno puerto SCI0. La rutina que atiende el servicio de interrupción se encarga de adquirir datos provenientes de Internet, chequear la integridad del dato recibido, verificar las reglas del protocolo de comunicación y generar variables de estado (flags) acorde a los resultados de las operaciones mencionadas anteriormente.

La descripción detallada del algoritmo que implementa este bloque se encuentra en el Anexo A.

2.2 Bloque 2. Adquisición y Procesamiento de Datos.

El **bloque 2** básicamente es un lazo de espera en el programa principal. El programa permanecerá en este estado hasta que se cumpla con la condición de espera. Dicha condición es actualizada muestra a muestra por el servicio de interrupción del timer C0.

La condición es que se complete la adquisición de un bloque de datos, de largo N, del canal Modulo y un bloque de igual tamaño del canal Fase para su posterior procesamiento.

El servicio de interrupción del timer C0 es el encargado de adquirir los datos, llevar la cuenta (indexar una variable) y procesar o acondicionar los datos adquiridos. Una vez alcanzado N el programa sale del lazo de espera y procesa los **bloques 3, 4 y 5**.

El valor N (cantidad de muestras por bloque) es determinado por la velocidad de pasada del producto, velocidad de muestreo y la resolución deseada en detección.

2.2.1 Algoritmo del Timer C0.

El funcionamiento del algoritmo lo describiremos dividido en los siguientes pasos. En el Anexo B encontraremos el diagrama de flujo relacionado con estos pasos.

- 1) Inicia el proceso de conversión de los conversores AD para cada canal. Luego queda a la espera de la finalización de este ciclo. Finalmente detiene la conversión.
- 2) Según las condiciones determinadas por el funcionamiento del programa principal calcula el valor medio de las señales “Canal Modulo” y “Canal Fase”. Esta media esta relacionada con el “offset” de los amplificadores analógicos.
- 3) Lee los valores de “Canal Modulo” y “Canal Fase” de los registros de los conversores AD y los procesa.
Este procesamiento implica para ambos canales: restar el valor medio correspondiente y multiplicar este valor por una constante relacionada con el valor de amplificación de la última etapa analógica (resolución digital).
- 4) Guarda los datos en sendos buffers circulares transitorios.
- 5) Calcula la energía de cada canal mediante un integrador con perdidas. Estos valores se utilizaran luego para detectar la presencia de producto en cinta de transporte.
- 6) En función de las condiciones del programa principal verifica, por medio de los valores de energía calculados en el paso anterior, si esta ingresando un producto al detector de metales.
- 7) En función de las condiciones del programa principal realiza funciones de control automático de ganancia.
- 8) Por último indexa la variable de cuenta de largo de bloque y verifica final de buffer circular para posterior ajuste de punteros.

Los buffers transitorios mencionados en el paso 4 son del tipo circular, quiere decir que cuando encontramos en ellos una señal para procesar debemos procesar los datos a una velocidad que evite que la señal se sobrescriba.

Precisamente esta limitación es una de las razones para procesar las señales por bloque.

La descripción detallada de los algoritmos que interactúan en este bloque se encuentra en el Anexo B.

2.3 Bloque 3. Procesamiento de buffers circulares y buffers display.

El bloque 3 del programa principal se encarga de poner a cero la cuenta de largo de bloque, procesar los punteros a buffer para verificar si se puede procesar un bloque de señales de entrada (cancelamiento, entrenamiento, etc.) y armar un buffer de muestras para poder enviarlas y visualizarlas en el Display LCD que posee el detector de metales.

Armar el buffer de display implica decimar el buffer de señal de entrada (un bloque de tiempo igual al tiempo de refresco del display) y convertirlo a uno de 64 muestras (cantidad de muestras que puede imprimir el display por cada refresco).

Las muestras del buffer de display deben además ser convertidas de 12 bits a nuevas muestras de 8 bits.

La descripción detallada del algoritmo de este bloque y sus líneas de código se encuentran en el Apéndice C.

2.4 Bloque 4. Procesamiento de diferentes modos de operación.

El bloque 4 del programa principal se encarga de procesar las señales de entrada en función de que operación se deba realizar sobre esta (modo de operación).

El estudio de este bloque lo realizaremos analizando el funcionamiento de cada uno de estos modos.

Los mismos están relacionados con los pasos del algoritmo de detección de descriptores en la sección 1.4.

El funcionamiento del programa principal en uno u otro modo se define externamente con los comandos enviados por el Teclado/Display o por Internet.

El programa principal en sí mismo no tiene facultades para cambiar el modo en que está operando.

A continuación describiremos brevemente qué función realiza cada modo y por qué se implementaron de esa manera.

En el apéndice D encontraremos los detalles de diagramas de flujo, funciones y líneas de código que intervienen en cada modo.

2.4.1. Modo 0. Presetup

Cada vez que necesitamos guardar las características de un nuevo producto en memoria, el programa principal debe ingresar primero en este modo.

El modo 0 es el primer paso en el proceso de entrenamiento de producto y probeta de contraste para la correlación. Una vez finalizado se habrán calculado y almacenado en variables de memoria las energías de ruido de cada canal y los valores óptimos de ganancia para el producto en cuestión.

Una vez ingresado mediante el comando Presetup, el programa permanecerá en este modo hasta que se cumplan los cuatro pasos temporizados que el modo implementa, excepto que, se emita un comando que fuerce el programa principal a salir.

Los pasos son:

- 1) Inicialización variables de producto y amplificadores.
- 2) Medición de rango optimo de ganancia. El operador debe pasar uno o varios productos por el detector para medir la ganancia.
- 3) Configura amplificadores a la ganancia hallada en paso anterior.
- 4) Calcula máxima energía de ruido y lo guarda en las variables correspondientes en memoria RAM.

Cuando se completan los pasos anteriores sale automáticamente del modo 0.

En el apéndice D.1 encontremos los detalles de los algoritmos y líneas de código que intervienen en este modo.

2.4.2. Modo 1. Entrenamiento Producto

Para poder cancelar o disminuir el efecto que el producto genera en el detector de metales el algoritmo de detección descrito en la sección 1.3 necesita una muestra de este efecto para poder lograr su cometido.

La forma en la que adquiere esta información es midiendo el efecto producto sin ninguna otra interferencia.

El modo 1 es el modo en el que ingresa el programa principal para grabar en memoria RAM la forma de onda del efecto producto.

El entrenamiento producto puede ser único, medir y grabar el efecto de un solo producto, o estadístico, se promedia y graba la medición de N productos pasados por el detector. Esta última es la forma recomendada para mejorar el cancelamiento del efecto producto.

Debemos destacar que este modo es el complemento del modo 0 ya que en este último es donde se encuentran los valores de ganancia para el producto en cuestión.

Cada vez que se quiera grabar un nuevo producto debemos comenzar siempre por el modo anterior.

El programa principal saldrá automáticamente de este modo cuando se cumpla la pasada de los N productos por el portal. El valor N se asigna por comando.

En el apéndice D.1 encontremos los detalles de los algoritmos y líneas de código que intervienen en este modo.

2.4.3. Modo 2. Entrenamiento Probeta

Uno de los datos necesario que necesita el algoritmo de la sección 1.3 para poder detectar pequeñas partículas de un determinado metal dentro de un envoltorio metálico, es la medida de la interferencia que genera una probeta patrón del metal buscado dentro del producto con envoltorio metalizado que entrenamos en el modo anterior.

Para poder obtener esa información el programa principal debe ingresar en este modo.

Al igual que en el modo anterior se pueden grabar una única probeta o el promedio estadístico de la pasada de N probetas de igual característica.

Como ya mencionamos en la sección 1.4 el la forma de onda que se grabara en memoria será el resultado de la resta entre: la forma de onda que se genera al pasar por el portal del detector un producto (el entrenado en el modo anterior) con una probeta incluida y la forma de onda grabada en el modo anterior (efecto producto).

El programa principal saldrá automáticamente de este modo cuando se cumpla la pasada de los N productos por el portal. El valor N se asigna por comando.

En el apéndice D.2 encontremos los detalles de los algoritmos y líneas de código que intervienen en este modo.

2.4.4. Modo 3. Entrenamiento Umbrales

Otros de los datos que necesita el algoritmo de la sección 1.3 son los valores de los umbrales contra los cuales debe contrastar los datos adquiridos mientras funciona en modo 4 (detección) y así por contraste de umbrales determinar si existe detección o no.

El programa principal debe ingresar en este modo para calcular los umbrales.

Los umbrales en cuestión son dos: umbrales de amplitud y umbrales de correlación. Cada canal tiene sus propios umbrales.

El cálculo de los umbrales es flexible. Las funciones que implementa este modo permiten grabar umbrales de amplitud solamente, umbrales de correlación solamente o grabar ambos umbrales.

Esto le brinda al programa principal la flexibilidad de trabajar en el modo de detección mixto, correlación y amplitud, o elegir uno solo de estos dos.

La posibilidad de selección entre detectar por amplitud o correlación se fija en función de la variedad de productos con la que trabajara el detector. Algunos de ellos presentan características optimas de detección por un método, otros al revés y algunos otros lo mejor es aplicarle ambos métodos a la vez.

Recordemos que el entrenamiento de umbrales se realiza pasando productos por la cinta y guardando los valores máximos del residuo (efecto producto pasante- producto en memoria), para el caso amplitud, y guardando los valores máximos de crosacorrelacion entre residuo y efecto probeta guardado en memoria.

El programa principal sale de este modo solamente por comando. No existe un numero N prefijado de pasadas de producto para calcular los umbrales. De todos modos el sentido común indica que cuanto mayor sea la muestra para calcular los umbrales se obtendrá un mejor desempeño en el sentido de falsa alarma.

En el apéndice D.3 encontremos los detalles de los algoritmos y líneas de código que intervienen en este modo.

2.4.5. Modo 4. Detección

Una vez que se cuenta con los valores de efecto producto, efecto probeta y umbrales el detector de metales esta en condiciones de pasar a este modo.

Este será el modo en el que deberá permanecer trabajado el detecto hasta que se decida entrenar algún nuevo producto.

La detección por amplitud se realizará bloque a bloque contrastando el valor de umbral del bloque en cuestión con el valor calculado durante la detección para ese bloque. Por lo tanto las alarmas de detección por amplitud (luces, accionamiento de pistón, etc.) se podrán dar en forma casi instantánea. El tiempo de alarma menor a $2 \times$ tiempo de bloque.

En el caso de detección por correlación se necesitan adquirir una cierta cantidad N de bloques para poder hacer el filtro acoplar. Esta cantidad de bloques esta asociada con la diferencia de largo entre la señal que genera el producto y la señal que genera una probeta.

El programa principal saldrá de este modo mediante un comando.

En el apéndice D.4 encontremos los detalles de los algoritmos y líneas de código que intervienen en este modo.

2.5 Bloque 5. Atención y procesamiento de comandos externos.

El bloque 5 del programa principal es el ultimo en la línea de ejecución y se encarga de procesar los comandos recibidos por alguno de los puertos series de los que dispone. Recordemos que el puerto 0 corresponde al Teclado/Display y el puerto 1 corresponde a la interfaz con Internet.

El procesamiento de los comandos se llevará a cabo si la variable que indica nuevo comando esta en habilitada. Esta variable se habilitara si alguno de los servicios de interrupción de los puertos de comunicación serie verifican que se ha recibido una serie de datos que se corresponden con el formato de un comando perteneciente al protocolo de comunicación.

Una vez que el comando sea procesado, la variable que indica nuevo comando se deshabilitara permitiéndole al programa principal procesar un nuevo comando, si es que existe uno. De este modo solo se puede procesar un solo comando a la vez.

Los comandos se dividieron en subgrupos para poder procesarlos mas rápido y dejarle al programa principal el mayor slot de tiempo posible y una mayor prioridad al procesamiento de señales.

El número de identificación del comando esta determinado por el protocolo de comunicación.

A continuación detallaremos los tres grupos de comandos que se diseñaron para el desarrollo y funcionamiento del detector.

En el apéndice E encontremos los detalles de los algoritmos y líneas de código que intervienen en este modo.

2.5.1 Comandos de Debug (Procesamiento con Interfaz Matlab)

Los comandos de debug fueron diseñados específicamente para propósitos de depuración de programa. Son los comandos que trabajan en conjunto con el software de debug desarrollado en el lenguaje Matlab.

Principalmente estos comandos se utilizan para ver las señales que se encuentran en la memoria del procesador DSP (efecto producto, resta efecto producto, curva de correlación, umbrales de amplitud, osciloscopio digital en tiempo diferido, diagrama de fase producto, fft, etc.). De este modo aceleramos el proceso de desarrollo estudiando las señales y los resultados de operaciones sobre estas señales que se hacen visibles en una interfaz grafica cuando las requerimos con algún comando de este tipo.

Los comandos de debug tienen en mismo formato que el resto de los comando, es decir cumplen con el protocolo diseñado para la comunicación de procesador DSP con el Teclado/Display e Internet.

Para procesar la mayoría de estos comandos se debe detener la adquisición de datos (detener el timer C0) porque el procesamiento y envío al software Matlab es lento en relación a la velocidad de procesamiento de bloque.

En el apéndice E.1 se pueden encontrar detalles y ejemplos de este tipo de comandos.

2.5.2 Comandos de Operación

Los comandos de operación están relacionados con los modos de operación.

A excepción del comando de presetup, todos funcionan de manera similar. Cada vez que se ejecuta un comando de este tipo se inicializan las variables y flags que tienen que ver con el control de bloque y sensado de producto. También se inicializan los punteros y buffers de señal que intervienen en las operaciones de

Cada vez que se completa el procesamiento del comando se transmite un comando de recepción ok .

En el apéndice E.2 se pueden encontrar detalles y ejemplos de este tipo de comandos.

2.5.3. Comandos de Control de Hardware

En este grupo se ha colocado aquel tipo de comando que esta relacionado con la configuración de hardware.

Una de las ventajas de agruparlos de esta manera es que el grupo 3 de comandos es la última opción que evalúa el programa principal cuando procesa comandos, entonces como estos comandos son poco frecuentes durante la operación normal del equipo, se disminuye el tiempo promedio de procesamiento de comandos.

En el apéndice E.3 se pueden encontrar detalles y ejemplos de este tipo de comandos.