

Manual de Usuario Librería SITAU "Phased Array"



DASEL, SL
Avda. de Madrid, 84
Arganda del Rey
28500 MADRID
www.daselsistemas.com

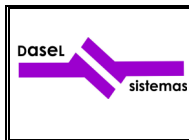
Edición: 2.1 - 29/05/2009

Revisión: 2

Fecha Revisión: 19/06/2009

ÍNDICE

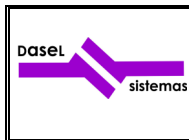
1	CONTROL DE VERSIONES	5
2	FUNCIONES DEL TIPO 'SET'. CONFIGURACIÓN DE PARÁMETROS DE ACCESO DIRECTO	7
2.1	PROGRAMACIÓN DE LOS PARÁMETROS DE ADQUISICIÓN	7
	<i>INT ST_SetGain (FLOAT data)</i>	<i>7</i>
	<i>INT ST_SetOffsetTime (FLOAT data)</i>	<i>7</i>
	<i>INT ST_SetAcqMode (INT data)</i>	<i>7</i>
	<i>INT ST_SetSignalTest (INT data)</i>	<i>7</i>
	<i>INT ST_SetTXMode (INT data)</i>	<i>8</i>
2.2	PROGRAMACIÓN DE PARÁMETROS DEL "PULSER"	8
	<i>INT ST_SetPulser (INT npulses, FLOAT wpulse)</i>	<i>8</i>
2.3	PROGRAMACIÓN DE PARÁMETROS DEL PROCESAMIENTO DE SEÑAL	8
	<i>INT ST_SetRFFilterEnable (INT data)</i>	<i>8</i>
	<i>INT ST_SetRFFilter (FLOAT low, FLOAT high)</i>	<i>8</i>
	<i>INT ST_SetReductionFactor (INT data)</i>	<i>9</i>
2.4	PROGRAMACIÓN DE LOS PARÁMETROS DE CONFIGURACIÓN DE LOS ENCODERS	10
	<i>INT ST_SetEncoderConfig (INT data, INT encoder)</i>	<i>10</i>
	<i>INT ST_SetEncoderZero (INT encoder)</i>	<i>10</i>
	<i>INT ST_SetEncoderSign (INT encoder)</i>	<i>10</i>
	<i>INT ST_SetEncoderTrigger (INT data)</i>	<i>10</i>
	<i>INT ST_SetEncoderCalibration (INT data, INT encoder)</i>	<i>11</i>
2.5	PROGRAMACIÓN DE LA CURVA TGC	11
	<i>INT ST_SetCATEnable (INT data)</i>	<i>11</i>
	<i>INT ST_SetCAT (FLOAT *gain, FLOAT *time, USHORT length)</i>	<i>11</i>
2.6	PROGRAMACIÓN DE LAS PUERTAS	12
	<i>INT ST_SetStartGate (FLOAT data)</i>	<i>12</i>
	<i>INT ST_SetEndGate (FLOAT data)</i>	<i>12</i>
	<i>INT ST_SetThrGate (FLOAT data)</i>	<i>13</i>
	<i>INT ST_SetGateEnable (INT data)</i>	<i>13</i>
3	FUNCIONES DEL TIPO 'LOAD'. CONFIGURACIÓN DE PARÁMETROS DE ACCESO DIFERIDO	13
3.1	PROGRAMACIÓN DE LOS PARÁMETROS DE ADQUISICIÓN	13
	<i>INT ST_LoadAcqConfig (FLOAT startdist, FLOAT enddist)</i>	<i>13</i>
	<i>INT ST_LoadPRF (INT data)</i>	<i>14</i>
3.2	PROGRAMACIÓN DE LAS LEYES FOCALES	14
	<i>INT ST_LoadLinearScanConfig (USHORT step, USHORT aperture, USHORT startelement, USHORT endelement, FLOAT angle)</i>	<i>14</i>
	<i>INT ST_LoadSectorialScanConfig (FLOAT step, WORD aperture, WORD startelement, FLOAT startangle, FLOAT endangle)</i>	<i>15</i>
	<i>INT ST_LoadFocalPoint (FLOAT emission, FLOAT reception)</i>	<i>15</i>
	<i>INT ST_LoadFocalPointEnable (INT data)</i>	<i>15</i>
	<i>INT ST_LoadFocalLaw (CHAR *dle, CHAR *dlr, CHAR *nmz, CHAR *ap)</i>	<i>16</i>
	<i>INT ST_LoadMaterialVelocity (INT data)</i>	<i>16</i>
3.3	PROGRAMACIÓN DE LOS PARÁMETROS DE CONFIGURACIÓN DEL SENSOR PA	17
	<i>INT ST_LoadPAConfig (INT nelement, FLOAT pitch, FLOAT frequency, FLOAT distance, INT autofreq, INT autopulse)</i>	<i>17</i>
3.4	PROGRAMACIÓN DE LOS PARÁMETROS DE CONFIGURACIÓN DE LA ZAPATA	18
	<i>INT ST_LoadWedgeConfig (FLOAT height, FLOAT width, FLOAT padist, FLOAT angle, FLOAT velocity)</i>	<i>18</i>
	<i>INT ST_LoadWedgeEnable (INT data)</i>	<i>18</i>
4	FUNCIONES DEL TIPO 'GET'. ACCESO A LOS PARÁMETROS DE CONFIGURACIÓN	19
4.1	PARÁMETROS DE ADQUISICIÓN	19
	<i>FLOAT ST_GetGain (VOID)</i>	<i>19</i>
	<i>FLOAT ST_GetAcqStartDepth (VOID)</i>	<i>19</i>
	<i>FLOAT ST_GetAcqEndDepth (VOID)</i>	<i>19</i>



Manual de Usuario Librería SITAU "Phased Array"

Versión:	2.1 29/05/2009
Revisión:	2
Fecha Revisión:	19/06/2009

FLOAT ST_GetAcqStartTime (VOID)	19
FLOAT ST_GetAcqEndTime (VOID)	19
FLOAT ST_GetAcqTime (VOID)	20
INT ST_GetTriggerSource (VOID)	20
FLOAT ST_GetSamplingFreq (VOID)	20
INT ST_GetAcqMode (VOID)	20
INT ST_GetPRF (VOID)	20
FLOAT ST_GetOffsetTime (VOID)	21
INT ST_GetLinesNumber (VOID)	21
ULONG ST_GetAscanDataNumber (VOID)	21
ULONG ST_GetImageSizeAcq (ULONG *nmu, ULONG *nlin)	21
4.2 PARÁMETROS DE CONFIGURACIÓN PARA EL CÁLCULO DE LAS LEYES FOCALES	21
FLOAT ST_GetFocalPointEmission (VOID)	21
FLOAT ST_GetFocalPointReception (VOID)	22
INT ST_GetFocalPointEnable (VOID)	22
INT ST_GetFocalLawMaterialEntrance (FLOAT *xe, FLOAT *ze, FLOAT *angle)	22
INT ST_GetMaterialVelocity (VOID)	22
4.3 PARÁMETROS DE CONFIGURACIÓN DEL BARRIDO	22
INT ST_GetAperture (VOID)	22
INT ST_GetScanStartElement (VOID)	23
INT ST_GetLinearScanStep (VOID)	23
INT ST_GetLinearScanStartElement (VOID)	23
INT ST_GetLinearScanEndElement (VOID)	23
FLOAT ST_GetSectorialScanStartAngle (VOID)	23
FLOAT ST_GetSectorialScanEndAngle (VOID)	23
FLOAT ST_GetSectorialScanStepAngle (VOID)	24
FLOAT ST_GetImageType (VOID)	24
4.4 PARÁMETROS DE CONFIGURACIÓN DE LA ZAPATA	24
INT ST_GetWedgeEnable (VOID)	24
FLOAT ST_GetWedgeHeight (VOID)	24
FLOAT ST_GetWedgeWidth (VOID)	24
FLOAT ST_GetWedgePADistance (VOID)	24
FLOAT ST_GetWedgeAngle (VOID)	25
FLOAT ST_GetWedgeVelocity (VOID)	25
4.5 PROGRAMACIÓN DE PARÁMETROS DEL "PULSER"	25
INT ST_GetPulsesNumber (VOID)	25
FLOAT ST_GetPulseWidth (VOID)	25
4.6 PROGRAMACIÓN DE LOS PARÁMETROS DE CONFIGURACIÓN DE LOS ENCODERS	25
INT ST_GetEncoderCalibration (INT data)	25
4.7 PROGRAMACIÓN DE LA CURVA TGC	26
INT ST_GetCATSize (VOID)	26
INT ST_GetCAT (FLOAT *gain, FLOAT *time)	26
INT ST_GetCATRange (VOID)	26
4.8 PARÁMETROS DE CONFIGURACIÓN DEL SENSOR PA	26
FLOAT ST_GetPAPitch (VOID)	26
INT ST_GetPAElementNumber (VOID)	27
FLOAT ST_GetPAFrequency (VOID)	27
FLOLAT ST_GetPADistance (VOID)	27
4.9 PARÁMETROS DE CONFIGURACIÓN PARA PROCESAMIENTO DE SEÑAL	27
FLOAT ST_GetRFFilterLow (VOID)	27
FLOAT ST_GetRFFilterHigh (VOID)	27
INT ST_GetRFFilterEnable (VOID)	27
INT ST_GetReductionFactor (VOID)	28
4.10 PARÁMETROS DE CONFIGURACIÓN DE LA PUERTA	28
FLOAT ST_GetStartGate (VOID)	28
FLOAT ST_GetEndGate (VOID)	28
FLOAT ST_GetThrGate (VOID)	28
INT ST_GetGateEnable (VOID)	28
5 FUNCIONES DEL TIPO 'RD'. ACCESO DIRECTO A LOS REGISTROS DEL HARDWARE	29



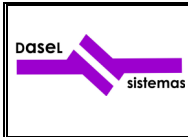
Manual de Usuario Librería SITAU "Phased Array"

Versión: 2.1
29/05/2009
Revisión: 2
Fecha Revisión: 19/06/2009

	<i>INT ST_RDTemp (BYTE *data)</i>	29
	<i>INT ST_RDEncoderValue (LONG *x, LONG *y, LONG *z)</i>	29
6	'GETINFO'. INFORMACIÓN DEL EQUIPO	29
	<i>VOID ST_InfoLibraryVersion (CHAR * data)</i>	29
	<i>WORD ST_InfoACTChannelsNumber (VOID)</i>	29
	<i>WORD ST_InfoMUXChannelsNumber (VOID)</i>	30
	<i>WORD ST_InfoBASENumber (VOID)</i>	30
	<i>WORD ST_InfoMODNumber (VOID)</i>	30
	<i>WORD ST_InfoEncodersNumber (VOID)</i>	30
	<i>VOID ST_InfoUCIFirmware (CHAR *date, CHAR *time, CHAR *name, CHAR *device)</i>	30
	<i>VOID ST_InfoBASEFirmware (CHAR *date, CHAR *time, CHAR *name, CHAR *device)</i>	31
	<i>VOID ST_InfoMODFirmware (CHAR *date, CHAR *time, CHAR *name, CHAR *device)</i>	31
7	OTRAS FUNCIONES	31
7.1	FUNCIONES DE ACCESO AL SISTEMA	31
	<i>INT ST_OpenSys (CHAR * path, USHORT num_channel, USHORT num_channel_mux)</i>	31
	<i>INT ST_CloseSys (void)</i>	32
7.2	FUNCIONES DE ADQUISICIÓN DE DATOS	32
	<i>LONG ST_GetBuffer (SHORT INT *bscan, SHORT INT *cscan, SHORT INT *dscan, LONG *encx, LONG *ency)</i>	32
7.3	FUNCIONES DE CONFIGURACIÓN	32
	<i>INT ST_PRG (VOID)</i>	32
	<i>INT ST_SaveConfig (CHAR *data)</i>	33
	<i>LONG ST_LoadConfig (CHAR *file, CHAR *date)</i>	33
7.4	FUNCIONES DE CONTROL DEL PROCESAMIENTO	33
	<i>INT ST_Run (INT wait)</i>	33
	<i>INT ST_RunScanSW (INT time)</i>	34
	<i>INT ST_RunScanTRG (VOID)</i>	35
	<i>INT ST_RunScanENC (VOID)</i>	35
	<i>INT ST_Stop (VOID)</i>	36
8	CONFIGURACIÓN INICIAL POR DEFECTO	37
9	TIPOS DE DATOS Y ACCESO A LAS FUNCIONES DE LA LIBRERÍA	38
10	CÓDIGOS DE ERROR	39
11	ESQUEMA DE PROGRAMACIÓN	41
11.1	DISPARO POR SOFTWARE - ST_RUN()	41
11.2	SCAN CON PROCESAMIENTO CONCURRENTENTE MULTI-THREAD (ST_RUNSCANSW(), ST_RUNSCANENC(), ST_RUNSCANTRG())	42

1 Control de Versiones

Versión Software	Revisión Documento	Actualización
2.1 29/05/2009	2 19/06/2009	- Se eliminan de este documento las características técnicas.
2.1 29/05/2009	1 29/05/2009	<ul style="list-style-type: none"> - Incluye función de configuración de adquisición por transmisión: ST_SetTXMode(). - Resuelve los problemas de la carga de la configuración por defecto. - Cambio ST_SetEncoderCalibration(), el parámetro que se pasa es el número de pulsos/mm - Se incrementa el conjunto de funciones: <ul style="list-style-type: none"> - ST_GetEncoderCalibration() - ST_GetEncoderTrigger() - Cambio en las siguientes funciones: <ul style="list-style-type: none"> - ST_SaveConfig() - ST_LoadConfig()
2.0 30/03/2009	1 30/03/2009	<p>Incluye funciones de configuración de zapata con ángulo de inclinación y superficie de contacto plana.</p> <p>Los cambios afectan a las siguientes funciones:</p> <ul style="list-style-type: none"> - ST_LoadWedgeConfig() - ST_GetBuffer() <p>Se incrementa el conjunto de funciones:</p> <ul style="list-style-type: none"> - ST_GetWedgeVelocity() - ST_GetWedgeHeight() - ST_GetWedgeWidth() - ST_GetWedgeEnable() - ST_GetWedgeAngle() - ST_GetWedgePADistance() <p>Cambio de nombre de funciones:</p> <ul style="list-style-type: none"> - ST_GetInfoVersion() → ST_GetInfoLibraryVersion() - STGetPAFreq → ST_GetPAFrequency
1.1.0 03/12/2008	1 03/12/2008	<p>Incluye funciones de interpolación sectorial</p> <p>Los cambios afectan a las siguientes funciones:</p> <ul style="list-style-type: none"> - ST_LoadLinearScanConfig() - ST_LoadFocalLaw() - ST_LoadWedgeConfig() - ST_RDEncoderValue() - ST_SetEncoderSign() <p>Se incrementa el conjunto de funciones:</p> <ul style="list-style-type: none"> - ST_GetInfoACTChannelsNumber() - ST_GetInfoMUXChannelsNumber() - ST_GetInfoBASENumber() - ST_GetInfoMODNumber() - ST_GetInfoEncodersNumber() - ST_GetInfoUCIFirmware() - ST_GetInfoBASEFirmware() - ST_GetInfoMODFirmware() - ST_GetScanStartElement() - ST_LoadWedgeEnable() - ST_GetFocalLawMaterialEntrance() - ST_GetPADistance - ST_LoadSectorialScanConfig() - ST_GetAcqStartTime() - ST_GetAcqEndTime() <p>Cambio de nombre de funciones:</p> <ul style="list-style-type: none"> - ST_GetInfoVersion() → ST_GetInfoLibraryVersion() - STGetPAFreq → ST_GetPAfrequency



Manual de Usuario Librería SITAU "Phased Array"

Versión: 2.1
29/05/2009
Revisión: 2
Fecha Revisión: 19/06/2009

1.0.0 30/09/2008	1 30/09/2008	Versión inicial de la documentación.

2 Funciones del tipo 'SET'. Configuración de parámetros de acceso directo

Este tipo de funciones configuran en el hardware del equipo valores de los parámetros de configuración de acceso directo, sin necesidad de ejecutar ninguna función posterior.

2.1 Programación de los parámetros de adquisición

INT ST_SetGain (FLOAT data)	
Configura el valor de la ganancia.	
Parámetros	
data (dB)	Ganancia. Rango → [0 dB .. 60 dB], el paso es de 0,2 dB.
Devuelve (INT)	
(Ver códigos de error)	
Dependencias	
Variaciones en la ganancia pueden provocar cambios en el rango de la curva CAT. Si ganancia > 48 dB → El rango de la curva CAT es (12 dB .. 60 dB) Si ganancia < 48 dB → El rango de la curva CAT es (0 dB .. 48 dB)	

INT ST_SetOffsetTime (FLOAT data)	
Configura el tiempo de retardo, que se espera antes de iniciar la adquisición de la imagen B-Scan. Este retardo es común a todos los elementos de la apertura.	
Parámetros	
data (µs)	Tiempo de retardo desde el inicio del procesamiento. Rango → [0 µs .. 1600 µs]
Devuelve (INT)	
(Ver códigos de error)	

INT ST_SetAcqMode (INT data)	
Configura el sistema el modo en el que se adquiere la señal.	
Parámetros	
data	0 → Modo radiofrecuencia. 1 → Modo vídeo (detección de envolvente mediante procesamiento digital). 2 → Rectificación de media onda (positiva) 3 → Rectificación de media onda (negativa) 4 → Rectificación de onda completa
Devuelve (INT)	
(Ver códigos de error)	

INT ST_SetSignalTest (INT data)	
El sistema genera una señal sintética de test en forma de diente de sierra.	
Parámetros	
data	0 → Señal de test deshabilitada. 1 → Señal de test habilitada.
Devuelve (INT)	
(Ver códigos de error)	

INT ST_SetTXMode (INT data)	
Configura el sistema para trabajar en pulso-eco, emite y recibe por el mismo conector E/R (EMITTER/RECEIVER), ó en transmisión, emite por el conector E/R y recibe por el conector R (RECEIVER).	
Parámetros	
data	0 → El sistema adquiere en pulso-eco, emite y recibe por el mismo conector E/R. 1 → El sistema adquiere en transmisión, emite por el conector E/R y recibe por el conector R.
Devuelve (INT)	
(Ver códigos de error)	

2.2 Programación de parámetros del “pulser”

INT ST_SetPulser (INT npulses, FLOAT wpulse)	
Configura los parámetros que definen el pulso de emisión.	
Parámetros	
npulses	Número de veces que se repite la secuencia de excitación definida con ST_SetTXCode() . Por defecto la secuencia de excitación es un único pulso. Rango → [1 .. 255]
wpulse (µs)	Ancho de los pulsos de excitación. Los posibles valores dentro del rango permitido son discretos y múltiplos de 6.25 ns, si el valor programado no es múltiplo de este valor, se produce un redondeo para ajustarlo al valor más próximo permitido. Rango → [50 µs .. 1500 µs] en pasos de 6,25 ns
Devuelve (INT)	
(Ver códigos de error)	

2.3 Programación de parámetros del procesamiento de señal

INT ST_SetRFFilterEnable (INT data)	
Habilita / Deshabilita Activa el filtro FIR paso banda de orden 63 de procesado de señal.	
Parámetros	
data	0 → Deshabilita el filtro digital FIR. 1 → habilita el filtro digital FIR.
Devuelve (INT)	
(Ver códigos de error)	

INT ST_SetRFFilter (FLOAT low, FLOAT high)	
Configura las frecuencias de corte del filtro digital FIR. Filtro FIR de 63 coeficientes con una frecuencia de muestreo de 40 MHz.	
Parámetros	
low (KHz)	Frecuencia de corte inferior del filtro. Rango → [0 KHz .. 40000 KHz]
high (KHz)	Frecuencia de corte superior del filtro. Rango → [0 KHz .. 40000 KHz]
Devuelve (INT)	
(Ver códigos de error)	

INT ST_SetReductionFactor (INT data)	
Configura el reductor de datos. La reducción es de (data+2) datos a 2 datos, ya que por cada intervalo de tamaño data+2 se generan 2 muestras.	
La cantidad de datos a la salida se calcula como: $N_{SAL} = 2 \left\lfloor \frac{N_{ENT}}{data + 2} \right\rfloor$	
La cantidad de datos a la salida siempre es par, incluso en el caso data = 0 (sin reducción).	
Parámetros	
data	Factor de reducción de datos. 0 → No se aplica reducción de datos. Rango → [0 .. 255]
Devuelve (INT)	
(Ver códigos de error)	

El sistema realiza la reducción de datos por un factor programable, R, de 8 bits. La reducción obtenida es de (R+2):2 (compresión), con $0 \leq R \leq 255$. Así, por ejemplo, si R=0, la reducción es de 2:2 (equivalente a 1:1, esto es, no hay compresión); si R=6 la reducción es de 8:2 (equivalente a 4:1); si R=7, la reducción es de 9:2, esto es, una muestra de salida por cada 4.5 muestras de entrada.

$\lfloor \cdot \rfloor$ Representa el truncado al entero más próximo hacia cero del argumento. Así, si la traza de entrada tiene 1000 muestras y la reducción es de R=6 (8:2), la salida tiene 250 muestras (la cuarta parte), pero si la traza de entrada tiene 999 muestras, la salida tiene 248 muestras ya que $\lfloor 999/8 \rfloor = 124$.

Para cada intervalo de largo (R+2) el reductor entrega 2 datos correspondientes al Máximo y Mínimo de la señal dentro del intervalo, teniendo en cuenta el formato de los datos (con o sin signo) y el orden de aparición.

- El reductor de datos siempre entrega una cantidad par de muestras. Incluso cuando se trabaja sin reductor (R=0, reducción 2:2), la cantidad de datos generados por el equipo será par. El redondeo es hacia abajo, por ejemplo, si se programan 1001 muestras con R=0, se recibirán 1000.
- Este operador *NO ES LINEAL*, y por lo tanto no debe utilizarse si el posterior procesamiento de las señales requiere esta propiedad (ej. Cálculo de la FFT)

2.4 Programación de los parámetros de configuración de los encoders

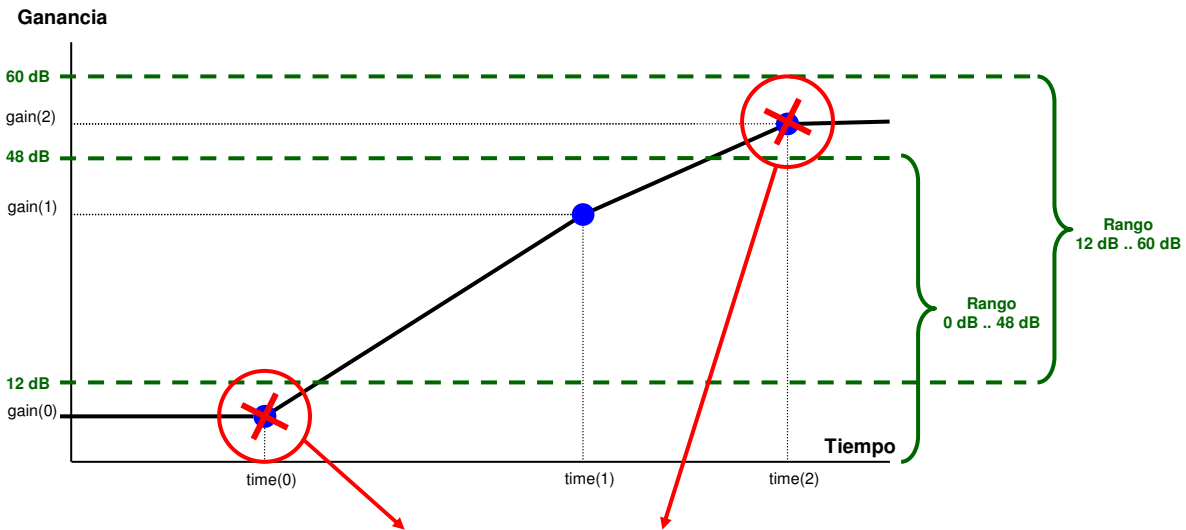
INT ST_SetEncoderConfig (INT data, INT encoder)	
Programa el número de pulsos de encoder que transcurren entre disparo y disparo, cuando el equipo se haya en modo de disparo por encoder. Este modo se configura mediante la función ST_SetScanENC() .	
Parámetros	
data (pasos)	Mínimo número de pulsos del encoder entre cada adquisición de una imagen. Rango → [1 .. 65535]
encoder	Encoder que se desea configurar. Rango → [0 .. 2]
Devuelve (INT)	
(Ver códigos de error)	
INT ST_SetEncoderZero (INT encoder)	
Resetea el valor del encoder, que se lee mediante la función ST_RDEncoderValue() .	
Parámetros	
encoder	Encoder que se desea poner a cero. Rango → [0 .. 2]
Devuelve (INT)	
(Ver códigos de error)	
INT ST_SetEncoderSign (INT encoder)	
Cambia el signo del encoder.	
Parámetros	
encoder	Encoder del que se desea cambiar el signo. Rango → [0 .. 2]
Devuelve (INT)	
(Ver códigos de error)	
INT ST_SetEncoderTrigger (INT data)	
Configura el encoder con el que se desea que dispare el sistema.	
Parámetros	
data	Encoder con el que se dispara el procesamiento y la adquisición de la imagen. Rango → [0 .. 2]
Devuelve (INT)	
(Ver códigos de error)	

INT ST_SetEncoderCalibration (INT data, INT encoder)	
Calibra el encoder a partir del recorrido de una distancia conocida. Esta función configura número de pasos del encoder por milímetro recorrido.	
Parámetros	
data	Número de pulsos por milímetro.
encoder	Encoder que se desea calibrar. Rango → [0 .. 2]
Devuelve (INT)	
(Ver códigos de error)	

2.5 Programación de la curva CAT

INT ST_SetCATEnable (INT data)	
Función para la habilitación de la curva CAT (Curva de atenuación a lo largo del tiempo de adquisición).	
Parámetros	
data	0 → Curva CAT deshabilitada. 1 → Curva CAT habilitada.
Devuelve (INT)	
(Ver códigos de error)	

INT ST_SetCAT (FLOAT *gain, FLOAT *time, USHORT length)	
Carga la curva CAT de en la memoria del equipo. El valor para la curva CAT en t=0 es el primer valor de ganancia facilitado por el usuario, independientemente del tiempo especificado para el mismo. El funcionamiento de la curva CAT es necesario habilitarlo mediante la función ST_SetCATEnable()	
Parámetros	
*gain (dB)	Vector que contiene los valores de la ganancia que el usuario desea en los instantes indicados en el vector 'time'. Los valores del vector 'gain' han de ir en el rango de 12 a 60 dB o bien de 0 a 48 dB.
*time (µs)	Vector de tiempos que indica en qué instantes se desean los valores de ganancia del vector 'gain'. Todos los elementos han de ser positivos e ir en orden creciente. Rango → [0 .. Tiempo de adquisición]
length	Longitud de los vectores 'gain' y 'time'. Debe ser menor de 2047.
Devuelve (INT)	
(Ver códigos de error)	



Puntos NO válidos.
En una curva CAT no pueden existir puntos que pertenecen a rangos distintos

Fig. 1.- Rangos de ganancia de la curva CAT

2.6 Programación de las puertas

INT ST_SetStartGate (FLOAT data)	
Configura la distancia inicial de la puerta.	
Parámetros	
data (mm)	Distancia inicial de la puerta. - Tiene que ser mayor que el inicio de la adquisición. - Tiene que ser menor que el final de la adquisición. - Tiene que ser menor que el final de la puerta.
Devuelve (INT)	
(Ver códigos de error)	

INT ST_SetEndGate (FLOAT data)	
Configura la distancia final de la puerta.	
Parámetros	
data (mm)	Distancia final de la puerta. - Tiene que ser mayor que el inicio de la adquisición. - Tiene que ser menor que el final de la adquisición. - Tiene que ser mayor que el inicio de la puerta.
Devuelve (INT)	
(Ver códigos de error)	

INT ST_SetThrGate (FLOAT data)	
Configura el umbral de la puerta.	
Parámetros	
data (%)	Umbral de la puerta. - Si el modo de adquisición es RF Rango → [-50 .. 50] - Si el modo de adquisición es distinto de RF: Rango → [0 .. 100]
Devuelve (INT)	
(Ver códigos de error)	

INT ST_SetGateEnable (INT data)	
Habilita / deshabilita la puerta.	
Parámetros	
data	0 → Deshabilita la puerta. 1 → Habilita la puerta.
Devuelve (INT)	
(Ver códigos de error)	

3 Funciones del tipo 'LOAD'. Configuración de parámetros de acceso diferido

Este tipo de funciones cargan en memoria los valores de los parámetros de configuración de acceso diferido, pero no acceden al equipo, tan solo almacena su valor en la memoria del PC, para su posterior programación en el equipo con la función **ST_PRG()**.

3.1 Programación de los parámetros de adquisición

INT ST_LoadAcqConfig (FLOAT startdist, FLOAT enddist)	
Carga en memoria los parámetros de configuración del rango de la inspección, es decir, la distancia inicial en la que comienza la adquisición y la distancia final a la que finaliza la adquisición, estas distancias se miden desde la entrada del material (0 mm). Estos parámetros no se configuran en el hardware del sistema SITAU, para configurarlos en el sistema es necesario ejecutar después la función ST_PRG()	
Parámetros	
startdist (mm)	Distancia inicial a partir de la cual comienza la adquisición, desde la entrada del material.
enddist (mm)	Distancia en la que finaliza la adquisición, desde la entrada del material.
Devuelve (INT)	
(Ver códigos de error)	

INT ST_LoadPRF (INT data)	
<p>Carga en el sistema el valor de la frecuencia de repetición de disparos para la adquisición de las líneas (A-Scan) que componen la imagen B-Scan, PRF (Pulse Repetition Frequency), es el inverso del tiempo mínimo que ha de pasar entre dos disparos del equipo, teniendo en cuenta que con cada disparo se obtiene una línea (A-Scan) de la imagen B-Scan, este parámetro representa el tiempo mínimo entre adquisiciones de líneas en una imagen B-Scan, el tiempo mínimo transcurrido entre adquisiciones de imágenes (B-Scan) será este tiempo multiplicado por el número de líneas que tiene la imagen. Según esto, la frecuencia de repetición de imágenes es:</p> $PRF_{imagen} = PRF_{línea} / N_{líneas}$ <p>La frecuencia de disparo nunca será mayor que la PRF, pero puede ser menor en función del tiempo de adquisición o de la velocidad a la que se desplaza el encoder.</p> <p>Estos parámetros no se configuran en el hardware del sistema SITAU, para configurarlos en el sistema es necesario ejecutar después la función ST_PRG().</p>	
Parámetros	
data (Hz)	Frecuencia de repetición de adquisición de líneas. Rango → [350 Hz .. 10000 Hz]
Devuelve (INT)	
(Ver códigos de error)	

3.2 Programación de las leyes focales

INT ST_LoadLinearScanConfig (USHORT step, USHORT aperture, USHORT startelement, USHORT endelement, FLOAT angle)	
<p>Carga en memoria los parámetros de configuración para realizar un barrido lineal.</p> <p>Estos parámetros no se configuran en el hardware del sistema SITAU, para configurarlos en el sistema es necesario ejecutar después la función ST_PRG()</p>	
Parámetros	
step	Número de elementos que se desplaza la apertura en cada disparo del barrido lineal. Debe estar entre 0 y <i>endelement-aperture</i>
aperture	Número de elementos activos en cada disparo. La apertura se corresponde con los elementos que se disparan según la ley focal en emisión y por los que se recibe la señal según la ley focal en recepción. Debe estar entre 0 y <i>nº de canales activos</i> .
startelement	Elemento inicial del barrido lineal. Su valor está comprendido entre 1 y el número máximo de elementos del array. Siempre ha de ser menor que <i>endelement-aperture</i>
endelement	Elemento final del barrido lineal. Su valor está comprendido entre 1 y el Número máximo de elementos de array. Siempre ha de ser mayor que <i>startelement+aperture</i>
angle (°)	Ángulo del barrido.
Devuelve (INT)	
(Ver códigos de error)	

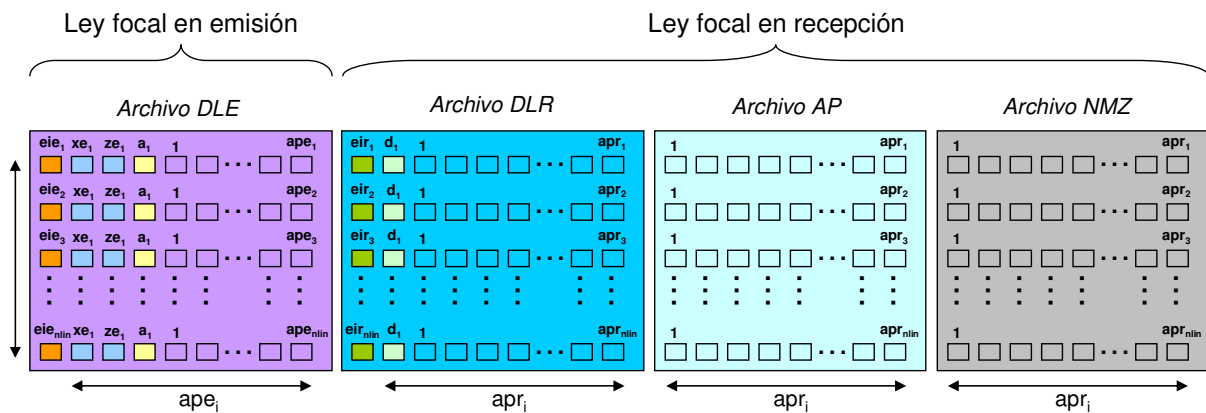
INT ST_LoadSectorialScanConfig (FLOAT step, WORD aperture, WORD startelement, FLOAT startangle, FLOAT endangle)	
Carga en memoria los parámetros de configuración para realizar un barrido angular (sectorial). Estos parámetros no se configuran en el hardware del sistema SITAU, para configurarlos en el sistema es necesario ejecutar después la función ST_PRG()	
Parámetros	
step (°)	Incremento angular que se realiza entre cada línea del barrido angular.
aperture	Número de elementos activos en cada disparo. La apertura se corresponde con los elementos que se disparan según la ley focal en emisión y por los que se recibe la señal según la ley focal en recepción. Debe estar entre 0 y n° de canales activos.
startelement	Elemento inicial de la apertura. Su valor está comprendido entre 1 y el número máximo de elementos del array menos el tamaño de la apertura.
startangle (°)	Angulo inicial del barrido.
endangle (°)	Angulo final del barrido.
Devuelve (INT)	
(Ver códigos de error)	

INT ST_LoadFocalPoint (FLOAT emission, FLOAT reception)	
Carga en memoria los parámetros de configuración de las distancias focales en emisión y en recepción. Estas distancias se miden desde la entrada del material (0 mm). Estos parámetros no se configuran en el hardware del sistema SITAU, para configurarlos en el sistema es necesario ejecutar después la función ST_PRG() . Para que se realice la focalización según estas distancias, debe habilitarse la focalización en foco fijo mediante la función ST_LoadFocalPointEnable() .	
Parámetros	
emission (mm)	Distancia focal en emisión, desde la entrada del material.
reception (mm)	Distancia focal en recepción, desde la entrada del material.
Devuelve (INT)	
(Ver códigos de error)	

INT ST_LoadFocalPointEnable (INT data)	
Habilita la focalización con foco fijo según las distancias focales configuradas mediante la función ST_LoadFocalPoint() . Estos parámetros no se configuran en el hardware del sistema SITAU, para configurarlos en el sistema es necesario ejecutar después la función ST_PRG() .	
Parámetros	
data	0 -> Deshabilita la focalización con foco fijo. Es equivalente a situar los focos en el infinito. 1 -> Habilita la focalización con foco fijo. La configuración de las distancias focales se realiza mediante la función ST_LoadFocalPoint() .
Devuelve (INT)	
(Ver códigos de error)	

INT ST_LoadFocalLaw (CHAR *dle, CHAR *dlr, CHAR *nmz, CHAR *ap)	
Carga las leyes focales generadas por el usuario. Estos parámetros no se configuran en el hardware del sistema SITAU, para configurarlos en el sistema es necesario ejecutar después la función ST_PRG() .	
Parámetros	
*dle	Archivo que contiene las leyes focales en emisión.
*dlr	Archivo que contiene las leyes focales en recepción.
*nmz	Archivo que contiene el número de muestras nulas para cada ley focal en recepción.
*ap	Archivo que contiene los factores de multiplicación para cada elemento de la apertura en recepción (apodización).
Devuelve (INT)	
(Ver códigos de error)	

- ⇒ ape_i : Número de elementos de la apertura en emisión para la línea i del barrido.
- ⇒ apr_i : Número de elementos de la apertura en recepción para la línea i del barrido.
- ⇒ eie_i : Elemento inicial de la apertura en emisión, para la línea i del barrido.
- ⇒ eir_i : Elemento inicial de la apertura en recepción, para la línea i del barrido.
- ⇒ $nlin$: Número de líneas del barrido.
- ⇒ xe_i (mm): Coordenada x del punto de entrada al material.
- ⇒ ze_i (mm): Coordenada z del punto de entrada al material.
- ⇒ a_i (rad): Angulo de entrada de cada haz al material.



$$\min (eie, eir) + \max (ape, apr) - \min (eie, eir) \leq N^o \text{ Canales activos del sistema}$$

Fig. 2.- Formato de las matrices de configuración de las leyes focales.

INT ST_LoadMaterialVelocity (INT data)	
Carga en memoria la Estos parámetros no se configuran en el hardware del sistema SITAU, para configurarlos en el sistema es necesario ejecutar después la función ST_PRG() .	
Parámetros	
data (m/s)	Velocidad de propagación del ultrasonido.
Devuelve (INT)	
(Ver códigos de error)	

3.3 Programación de los parámetros de configuración del sensor PA

INT ST_LoadPAConfig (INT nelement, FLOAT pitch, FLOAT frequency, FLOAT distance, INT autofreq, INT autopulse)	
Carga en memoria los parámetros de configuración del sensor PA (<i>Phased Array Probe</i>). Estos parámetros no se configuran en el hardware del sistema SITAU, para configurarlos en el sistema es necesario ejecutar después la función ST_PRG()	
Parámetros	
nelement	Número total de elementos del sensor PA.
pitch (mm)	Separación entre dos elementos anexos del sensor PA.
frequency (MHz)	Frecuencia nominal del sensor PA.
distance (mm)	Distancia desde el borde del PA hasta el primer elemento.
autofreq	Cálculo automático de la frecuencia de muestreo del sistema a partir de la frecuencia nominal del sensor PA. 0 → desactivado el cálculo automático de la frecuencia de muestreo. 1 → activado el cálculo automático de la frecuencia de muestreo. $\text{Frecuencia de muestreo} ; \text{Frequency} \times 4 \text{ [MHz]}$
autopulse	Cálculo automático del ancho del pulso de excitación a partir de la frecuencia nominal del sensor PA. 0 → desactivado el cálculo automático del ancho del pulso. 1 → activado el cálculo automático del ancho del pulso. $\text{Ancho de pulso} = \frac{1}{2 \times \text{Frequency}} \text{ [ns]}$
Devuelve (INT)	
(Ver códigos de error)	

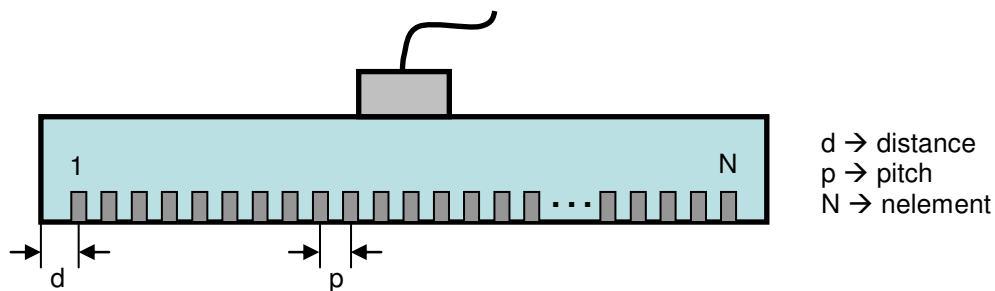


Fig. 3.- Parámetros de configuración del transductor "Phased Array".

3.4 Programación de los parámetros de configuración de la zapata

<i>INT ST_LoadWedgeConfig (FLOAT height, FLOAT width, FLOAT padist, FLOAT angle, FLOAT velocity)</i>	
Carga en memoria los parámetros de configuración de la zapata o material intermedio entre el sensor PA y el material. Estos parámetros no se configuran en el hardware del sistema SITAU, para configurarlos en el sistema es necesario ejecutar después la función ST_PRG()	
Parámetros	
height (mm)	Altura de la zapata en lado más alto.
width (mm)	Ancho de la zapata, superficie de contacto con el material.
padist (mm)	Distancia desde el borde inferior de la zapata hasta el borde del sensor PA.
angle (°)	Ángulo de inclinación de la zapata.
velocity (m/s)	Velocidad de propagación del sonido en el interior de la zapata.
Devuelve (INT)	
(Ver códigos de error)	

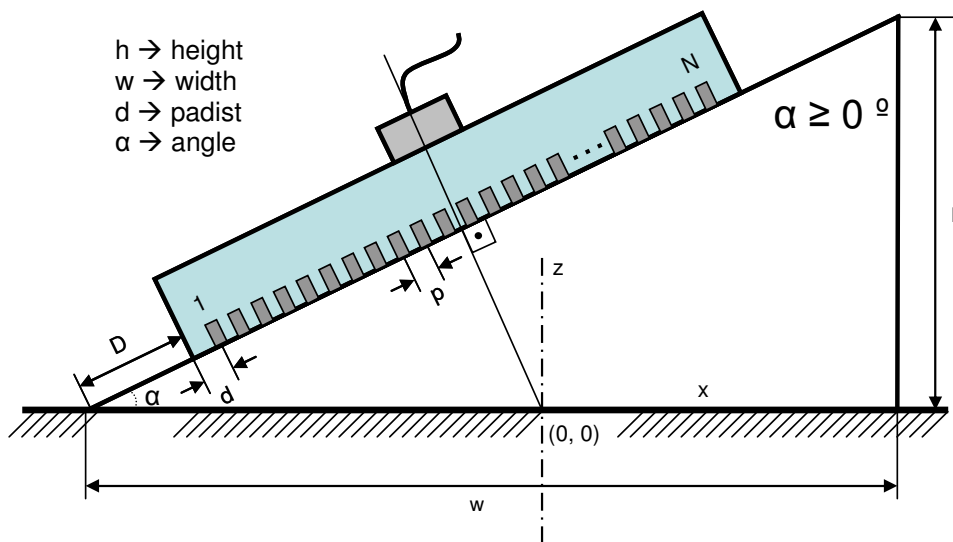


Fig. 4.- Parámetros dimensionales de la zapata.

<i>INT ST_LoadWedgeEnable (INT data)</i>	
Carga en memoria los parámetros de configuración para la habilitar / deshabilitar la zapata. Estos parámetros no se configuran en el hardware del sistema SITAU, para configurarlos en el sistema es necesario ejecutar después la función ST_PRG()	
Parámetros	
data	0 \rightarrow Deshabilita la zapata, el cálculo de las leyes focales se realiza sin tener en cuenta la zapata. 1 \rightarrow Habilita la zapata, el cálculo de las leyes focales se realiza teniendo en cuenta la zapata según los parámetros con los que ha sido configurada con la función ST_LoadWedgeConfig() .
Devuelve (INT)	
(Ver códigos de error)	

4 Funciones del tipo 'GET'. Acceso a los parámetros de configuración.

Este tipo de funciones devuelve el valor programado en el sistema para cada uno de los parámetros de configuración.

4.1 Parámetros de adquisición

FLOAT ST_GetGain (VOID)
Devuelve el valor de la ganancia programado en el sistema. En el caso de que la curva CAT esté habilitada devuelve el valor de la ganancia en t=0. Este valor se configura mediante la función ST_SetGain() .
Devuelve (FLOAT)
(dB) Devuelve el valor de la ganancia programado en el sistema.
FLOAT ST_GetAcqStartDepth (VOID)
Devuelve la distancia inicial de la adquisición, desde la entrada de material (0 mm). Este valor se configura con el primer parámetro de la función ST_LoadAcqConfig() .
Devuelve (FLOAT)
(mm) Distancia inicial de la adquisición medida desde le entrada del material (0 mm).
FLOAT ST_GetAcqEndDepth (VOID)
Devuelve la distancia final de la adquisición, desde la entrada de material (0 mm). Este valor se configura con el segundo parámetro de la función ST_LoadAcqConfig() .
Devuelve (FLOAT)
(mm) Distancia final de la adquisición medida desde le entrada del material (0 mm).
FLOAT ST_GetAcqStartTime (VOID)
Devuelve el tiempo inicial de la adquisición, desde la entrada de material (0 mm). Este valor se configura con el segundo parámetro de la función ST_LoadAcqConfig() .
Devuelve (FLOAT)
(µs) Tiempo inicial de la adquisición medida desde le entrada del material (0 mm).
FLOAT ST_GetAcqEndTime (VOID)
Devuelve el tiempo final de la adquisición, desde la entrada de material (0 mm). Este valor se configura con el segundo parámetro de la función ST_LoadAcqConfig() .
Devuelve (FLOAT)
(µs) Tiempo final de la adquisición medida desde le entrada del material (0 mm).

FLOAT ST_GetAcqTime (VOID)
Devuelve el tiempo de adquisición. Es la diferencia entre el tiempo final y el tiempo inicial de adquisición. Este valor depende de la velocidad de propagación del sonido en el material.
$t_{ADQ} = \frac{d_{FINAL} - d_{INICIAL}}{v_{MATERIAL}}$
Devuelve (FLOAT)
(µs) Tiempo de la adquisición.
INT ST_GetTriggerSource (VOID)
Devuelve el modo de disparo en el que se encuentra el sistema. Este valor se configura al ejecutar las funciones ST_Run() , ST_RunScanSW() , ST_RunScanENC() ó ST_RunScanTRG()
Devuelve (INT)
0 → Modo de disparo por software, en este modo el disparo se realiza con la función ST_Run() 1 → Modo de disparo por encoder, este modo de disparo se inicia con la función ST_RunENC() 2 → Modo de disparo por software continuo, este modo de disparo se inicia con la función ST_RunSW() 3 → Modo de disparo por evento externo, este modo de disparo se inicia con la función ST_RunTRG()
FLOAT ST_GetSamplingFreq (VOID)
Devuelve la frecuencia de muestreo de la señal. Este valor se configura de forma automática con la función ST_LoadPAConfig() cuando el último parámetro ('autofreq') es 1. En ese caso la relación entre la frecuencia de muestreo y la frecuencia nominal del sensor PA es:
Frecuencia de muestreo ; Frecuencia PA (freq) × 4
Devuelve (FLOAT)
(MHz) Frecuencia de muestreo de la señal.
INT ST_GetAcqMode (VOID)
Devuelve el modo de adquisición de la señal. Este valor se configura con el segundo parámetro de la función ST_SetAcqMode() .
Devuelve (INT)
0 → Modo radiofrecuencia. 1 → Modo vídeo (detección de envoltente mediante procesamiento digital). 2 → Rectificación de media onda (positiva) 3 → Rectificación de media onda (negativa) 4 → Rectificación de onda completa
INT ST_GetPRF (VOID)
Devuelve la frecuencia de repetición de disparos para la adquisición de las líneas (A-Scan) que componen la imagen B-Scan, PRF (Pulse Repetition Frequency). Este valor se configura con la función ST_LoadPRF() .
Devuelve (INT)
(Hz) Frecuencia PRF.

<i>FLOAT ST_GetOffsetTime (VOID)</i>
Devuelve el tiempo de retardo, que se espera antes de iniciar la adquisición de la imagen B-Scan. Este retardo es común a todos los elementos de la apertura. Este valor se configura con la función ST_SetOffsetTime() .
Devuelve (FLOAT)
(µs) Tiempo de retardo desde el inicio del procesamiento.

<i>INT ST_GetLinesNumber (VOID)</i>
Devuelve el número de líneas (A-Scan) que compone la imagen B-Scan.
Devuelve (INT)
Número de líneas que componen la imagen B-Scan.

<i>ULONG ST_GetAscanDataNumber (VOID)</i>
Devuelve el número de muestras que componen cada línea (A-Scan) de la imagen.
Devuelve (ULONG)
Número de muestras que componen cada línea de la imagen.

<i>ULONG ST_GetImageSizeAcq (ULONG *nmu, ULONG *nlin)</i>	
Devuelve la cantidad de datos adquiridos en la próxima imagen pendiente de leer del buffer. La imagen se debe leer del buffer con la función ST_GetBuffer()	
Parámetros	
*nmu	Número de muestras por línea (A-Scan) de la próxima imagen pendiente de leer del buffer.
*nlin	Número de líneas (A-Scan) de la próxima imagen pendiente de leer del buffer.
Devuelve (ULONG)	
Número de datos (SHORT) que ocupa la imagen B-Scan. Este valor debe ser utilizado para reservar la memoria dinámica cuyo puntero se pasa como primer parámetro a la función ST_GetBuffer() . El valor devuelto es igual a 'nmu' x 'nlin'.	

4.2 Parámetros de configuración para el cálculo de las leyes focales

<i>FLOAT ST_GetFocalPointEmission (VOID)</i>
Devuelve la distancia focal en emisión, para la focalización con foco fijo. Este valor ha sido configurado con el primer parámetro de la función ST_LoadFocalPoint()
Devuelve (FLOAT)
(mm) Distancia focal en emisión.

<i>FLOAT ST_GetFocalPointReception (VOID)</i>
Devuelve la distancia focal en recepción, para la focalización con foco fijo. Este valor se configura con el segundo parámetro de la función ST_LoadFocalPoint()
Devuelve (FLOAT)
(mm) Distancia focal en recepción.

<i>INT ST_GetFocalPointEnable (VOID)</i>
Permite al usuario conocer si está habilitada o deshabilitada la focalización con foco fijo según la configuración realizada con la función ST_LoadFocalPoint() .
Devuelve (FLOAT)
0 → La focalización con foco fijo está deshabilitada. Es equivalente a configurar los focos en misión y recepción en el infinito. 1 → La focalización con foco fijo está habilitada.

<i>INT ST_GetFocalLawMaterialEntrance (FLOAT *xe, FLOAT *ze, FLOAT *angle)</i>	
Devuelve los vectores de las coordenadas (x,y) de los puntos de entrada al material y los ángulos de incidencia de cada línea del barrido.	
*xe (mm)	Puntero al vector de las coordenadas x de los puntos de entrada de los haces en el material.
*ze (mm)	Puntero al vector de las coordenadas y de los puntos de entrada de los haces en el material.
*angle (radianes)	Puntero al vector de los ángulos de incidencia de los haces en el material.
Devuelve (INT)	
- (Ver códigos de error)	

<i>INT ST_GetMaterialVelocity (VOID)</i>
Devuelve la velocidad de propagación del sonido en el material. Este valor se configura con la función ST_SetMaterialVelocity() .
Devuelve (ULONG)
(m/s) Velocidad de propagación del sonido en el material.

4.3 Parámetros de configuración del barrido

<i>INT ST_GetAperture (VOID)</i>
Devuelve el número de elementos activos que forman la apertura. La apertura se corresponde con los elementos que se disparan según la ley focal en emisión y por los que se recibe la señal según la ley focal en recepción. Este valor se configura con la función ST_LoadLinearScanConfig()
Devuelve (ULONG)
Número de elementos de la apertura.

INT ST_GetScanStartElement (VOID)
Devuelve el valor configurado para el elemento inicial del barrido. Este valor se configura con las funciones de configuración de barrido ST_LoadLinearScanConfig() y ST_LoadSectorialScanConfig() .
Devuelve (INT)
- Elemento inicial del barrido. - (Ver códigos de error)
INT ST_GetLinearScanStep (VOID)
Devuelve el paso (número de elementos que se desplaza la apertura entre A-Scan y A-Scan de la imagen B-Scan) cuando se realiza un barrido lineal. Este valor se configura mediante la función ST_LoadLinearScanConfig() .
Devuelve (INT)
Número de elementos que se desplaza la apertura entre A-Scan y A-Scan de la imagen B-Scan.
INT ST_GetLinearScanStartElement (VOID)
Devuelve el elemento inicial de la apertura para la adquisición de la primera línea (A-Scan) de la imagen B-Scan. Este valor se configura mediante la función ST_LoadLinearScanConfig() .
Devuelve (INT)
Devuelve el elemento inicial de la apertura.
INT ST_GetLinearScanEndElement (VOID)
Devuelve el elemento final de la apertura, cuando se realiza un barrido lineal, para la adquisición de la última línea (A-Scan) de la imagen B-Scan. Este valor se configura mediante la función ST_LoadLinearScanConfig() .
Devuelve (INT)
Devuelve el elemento final de la apertura de la última línea cuando se realiza un barrido lineal.
FLOAT ST_GetSectorialScanStartAngle (VOID)
Devuelve el ángulo inicial para el barrido angular. Este valor se configura mediante la función ST_LoadSectorialScanConfig() .
Devuelve (FLOAT)
(°) Devuelve el ángulo inicial del barrido angular.
FLOAT ST_GetSectorialScanEndAngle (VOID)
Devuelve el ángulo final para el barrido angular. Este valor se configura mediante la función ST_LoadSectorialScanConfig() .
Devuelve (FLOAT)
(°) Devuelve el ángulo final del barrido angular.

FLOAT ST_GetSectorialScanStepAngle (VOID)
Devuelve el incremento angular para realizar un barrido angular. Este valor se configura mediante la función ST_LoadSectorialScanConfig() .
Devuelve (FLOAT)
(°) Devuelve el incremento angular del barrido.

FLOAT ST_GetImageType (VOID)
Devuelve el tipo de barrido que se esta realizando. Este valor se configura como barrido lineal mediante la función ST_LoadLinearScanConfig() , y como barrido sectorial lineal mediante la función ST_LoadSectorialScanConfig() .
Devuelve (FLOAT)
Devuelve el tipo de barrido. 0 → Barrido Lineal 1 → Barrido Sectorial o Angular.

4.4 Parámetros de configuración de la zapata

INT ST_GetWedgeEnable (VOID)
Permite al usuario conocer si está habilitada o deshabilitada la zapata. Este valor se configura mediante la función ST_LoadWedgeEnable() .
Devuelve (INT)
Devuelve el tipo de barrido. 0 → La zapata está deshabilitada. El contacto con el material se realiza directamente con el sensor PA. 1 → La zapata está Habilitada. El contacto con el material se realiza mediante la zapata.

FLOAT ST_GetWedgeHeight (VOID)
Devuelve la altura de la zapata. Este valor se configura mediante la función ST_LoadWedgeConfig() .
Devuelve (FLOAT)
(mm) Devuelve la altura de la zapata.

FLOAT ST_GetWedgeWidth (VOID)
Devuelve el ancho de la zapata. Este valor se configura mediante la función ST_LoadWedgeConfig() .
Devuelve (FLOAT)
(mm) Devuelve el ancho de la zapata.

FLOAT ST_GetWedgePADistance (VOID)
Devuelve la distancia que hay entre el borde del sensor PA y el borde de la zapata. Este valor se configura mediante la función ST_LoadWedgeConfig() .
Devuelve (FLOAT)
(mm) Devuelve la distancia desde el borde del sensor PA hasta el borde de la zapata.

FLOAT ST_GetWedgeAngle (VOID)
Devuelve el ángulo de inclinación de la zapata. Este valor se configura mediante la función ST_LoadWedgeConfig() .
Devuelve (FLOAT)
(°) Devuelve el ángulo de la zapata.

FLOAT ST_GetWedgeVelocity (VOID)
Devuelve la velocidad de propagación del sonido en el interior del material de la zapata. Este valor se configura mediante la función ST_LoadWedgeConfig() .
Devuelve (FLOAT)
(m/s) Devuelve el ángulo de la zapata.

4.5 Programación de parámetros del “pulser”

INT ST_GetPulsesNumber (VOID)
Devuelve el número de pulsos de excitación del pulser. Este valor se configura mediante la función ST_SetPulser() .
Devuelve (INT)
Devuelve el número de pulsos del pulser.

FLOAT ST_GetPulseWidth (VOID)
Devuelve el ancho del pulso de excitación del pulser. Este valor se configura mediante la función ST_SetPulser() .
Devuelve (FLOAT)
(ns) Devuelve el ancho del pulso de excitación.

4.6 Programación de los parámetros de configuración de los encoders

INT ST_GetEncoderCalibration (INT data)		
Devuelve el número de pulsos por milímetros configurados en el encoder. Este valor se configura mediante la función ST_SetEncoderCalibration() .		
Parámetros		
<table border="1"> <tr> <td>data</td> <td>Encoder del que se desea conocer su configuración. 0 → Encoder A 1 → Encoder B 2 → Encoder C 3 → Encogger D</td> </tr> </table>	data	Encoder del que se desea conocer su configuración. 0 → Encoder A 1 → Encoder B 2 → Encoder C 3 → Encogger D
data	Encoder del que se desea conocer su configuración. 0 → Encoder A 1 → Encoder B 2 → Encoder C 3 → Encogger D	
Devuelve (INT)		
Devuelve el número de pulsos por milímetro configurados en el encoder.		

INT ST_GetEncoderTrigger (VOID)

Devuelve el encoger con el que se realiza la adquisición.
Este valor se configura mediante la función **ST_SetEncoderTrigger()**.

Devuelve (INT)

Encoder del que se realiza la adquisición.
0 → Encoder A
1 → Encoder B
2 → Encoder C
3 → Encoger D

4.7 Programación de la curva CAT

INT ST_GetCATSize (VOID)

Devuelve el número de puntos de la curva de compensación Atenuación-Tiempo CAT.
Este valor se configura mediante la función **ST_SetCAT()**.

Devuelve (INT)

Devuelve el número de puntos de la curva de compensación Atenuación-Tiempo CAT.

*INT ST_GetCAT (FLOAT *gain, FLOAT *time)*

Devuelve los puntos que forman la curva de compensación Atenuación-Tiempo CAT.
Estos puntos se configuran mediante la función **ST_SetCAT()**.

Parámetros

*gain (dB)	Devuelve los valores de la ganancia correspondiente a los puntos de la curva CAT. El tamaño de la memoria que se debe reservar es el correspondiente al número de puntos: ST_GetCATSize()
*time (µs)	Devuelve los valores del tiempo correspondiente a los puntos de la curva CAT. El tamaño de la memoria que se debe reservar es el correspondiente al número de puntos: ST_GetCATSize()

Devuelve (INT)

Devuelve el número de pulsos de la curva de compensación Atenuación-Tiempo CAT.

INT ST_GetCATRange (VOID)

Permite al usuario conocer el rango de la curva CAT. Este valor se configura de forma automática en función de los valores de ganancia correspondientes a los puntos programados mediante la función **ST_SetCAT()**.

Devuelve (INT)

0 → El Rango de la curva CAT es [0 dB .. 48 dB]
1 → El rango de la curva CAT es [12 dB .. 60 dB]

4.8 Parámetros de configuración del sensor PA

FLOAT ST_GetPAPitch (VOID)

Devuelve el pitch del sensor Phased Array (PA), es decir, la separación entre elementos contiguos del array.
Este valor se configura con la función **ST_LoadPAConfig()**.

Devuelve (FLOAT)

(mm) Separación entre dos elementos contiguos del sensor PA.

INT ST_GetPAElementNumber (VOID)
Devuelve el número de elementos del sensor Phased Array (PA). Este valor se configura con la función ST_LoadPAConfig() .
Devuelve (INT)
Número de elementos del sensor PA.
FLOAT ST_GetPAFrequency (VOID)
Devuelve la frecuencia nominal del sensor Phased Array (PA). Este valor se configura con la función ST_LoadPAConfig() .
Devuelve (FLOAT)
(MHz) Frecuencia nominal del sensor PA.

FLOAT ST_GetPADistance (VOID)
Devuelve la distancia que hay entre el borde del sensor PA y el primer elemento. Este valor se configura con la función ST_LoadPAConfig() .
Devuelve (FLOAT)
(mm) Distancia del borde del PA hasta el primer elemento.

4.9 Parámetros de configuración para procesamiento de señal

FLOAT ST_GetRFFilterLow (VOID)
Devuelve la frecuencia de corte inferior del filtro FIR paso banda. Este valor se configura mediante la función ST_SetRFFilter() .
Devuelve (FLOAT)
(KHz) Devuelve la frecuencia de corte inferior del filtro FIR paso banda.

FLOAT ST_GetRFFilterHigh (VOID)
Devuelve la frecuencia de corte superior del filtro FIR paso banda. Este valor se configura mediante la función ST_SetRFFilter() .
Devuelve (FLOAT)
(KHz) Devuelve la frecuencia de corte superior del filtro FIR paso banda.

INT ST_GetRFFilterEnable (VOID)
Devuelve si el filtro digital FIR está habilitado o deshabilitado. Este valor se configura con la función ST_SetRFFilterEnable() .
Devuelve (INT)
0 → El filtro digital FIR está deshabilitado 1 → El filtro digital FIR está habilitado

INT ST_GetReductionFactor (VOID)
Devuelve el valor configurado para el reductor de datos. Este valor se configura con el segundo parámetro de la función ST_SetReductionFactor() .
Devuelve (INT)
- Valor del reductor de datos: La reducción es de (data+2) datos a 2 datos, ya que por cada intervalo de tamaño data+2 se generan 2 muestras. La cantidad de datos a la salida se calcula como: $N_{SAL} = 2 \left\lfloor \frac{N_{ENT}}{data + 2} \right\rfloor$ La cantidad de datos a la salida siempre es par, incluso en el caso data = 0 (sin reducción). - 0 → No se aplica reducción de datos. - (Ver códigos de error)

4.10 Parámetros de configuración de la puerta

FLOAT ST_GetStartGate (VOID)
Devuelve la distancia inicial de la puerta. Este valor se configura con la función ST_SetStartGate() .
Devuelve (INT)
(mm) Distancia inicial de la puerta.

FLOAT ST_GetEndGate (VOID)
Devuelve la distancia final de la puerta. Este valor se configura con la función ST_SetEndGate() .
Devuelve (INT)
(mm) Distancia final de la puerta.

FLOAT ST_GetThrGate (VOID)
Devuelve el umbral de la puerta. Este valor se configura con la función ST_SetThrGate() .
Devuelve (INT)
(% Pantalla) Umbral de la puerta.

INT ST_GetGateEnable (VOID)
Permite al usuario conocer si está habilitada o deshabilitada la puerta. Este valor se configura con la función ST_SetGateEnable() .
Devuelve (INT)
0 → La puerta está deshabilitada. 1 → La puerta está habilitada.

5 Funciones del tipo 'RD'. Acceso directo a los registros del hardware.

INT ST_RDTemp (BYTE *data)	
Lee la temperatura máxima del sistema.	
Parámetros	
*data (°C)	Puntero a la variable donde se devuelve la temperatura máxima del sistema.
Devuelve (INT)	
(Ver códigos de error)	

INT ST_RDEncoderValue (LONG *x, LONG *y, LONG *z)	
Devuelve la posición actual de los dos encoder expresada en número de pulsos.	
Parámetros	
*x (pasos)	Posición actual del encoder X.
*y (pasos)	Posición actual del encoder Y.
*z (pasos)	Posición actual del encoder Z.
Devuelve (INT)	
(Ver códigos de error)	

6 'GETINFO'. Información del equipo.

VOID ST_InfoLibraryVersion (CHAR * data)	
Versión de la librería.	
Parámetros	
*data	Puntero a una cadena de caracteres donde se devuelve la versión de la librería. Su longitud mínima debe ser 20.

WORD ST_InfoACTChannelsNumber (VOID)	
Devuelve el número máximo de canales activos (canales paralelos para la emisión / recepción) que tiene el sistema. Se corresponde con el parámetro que ha sido configurado previamente con la función ST_OpenSys()	
Devuelve (INT)	
Número máximo de canales activos.	

WORD ST_InfoMUXChannelsNumber (VOID)
Devuelve el número máximo de canales multiplexados que tiene el sistema. Si este valor es 0, no tiene ningún canal multiplexado, todos los canales son activos (paralelos). Se corresponde con el parámetro que ha sido configurado previamente con la función ST_OpenSys()
Devuelve (INT)
Número máximo de canales multiplexados.

WORD ST_InfoBASENumber (VOID)
Devuelve el número de bases que tiene el sistema.
Devuelve (INT)
Número de bases.

WORD ST_InfoMODNumber (VOID)
Devuelve el número de módulos por base que tiene el sistema.
Devuelve (INT)
Número de módulos por base.

WORD ST_InfoEncodersNumber (VOID)
Devuelve el número de encoders que tiene el sistema.
Devuelve (INT)
Número de encoders.

VOID ST_InfoUCIFirmware (CHAR *date, CHAR *time, CHAR *name, CHAR *device)	
Devuelve la información del firmware del dispositivo UCI.	
Parámetros	
*date	Puntero a una cadena de caracteres donde se devuelve la fecha de compilación del firmware. Su longitud mínima debe ser 50.
*time	Puntero a una cadena de caracteres donde se devuelve la hora de compilación del firmware. Su longitud mínima debe ser 50.
*name	Puntero a una cadena de caracteres donde se devuelve el nombre del firmware. Su longitud mínima debe ser 50.
*device	Puntero a una cadena de caracteres donde se devuelve el tipo de dispositivo para el que se generó el firmware. Su longitud mínima debe ser 50.

VOID ST_InfoBASEFirmware (CHAR *date, CHAR *time, CHAR *name, CHAR *device)

Devuelve la información del firmware del dispositivo BASE.

Parámetros

*date	Puntero a una cadena de caracteres donde se devuelve la fecha de compilación del firmware. Su longitud mínima debe ser 50.
*time	Puntero a una cadena de caracteres donde se devuelve la hora de compilación del firmware. Su longitud mínima debe ser 50.
*name	Puntero a una cadena de caracteres donde se devuelve el nombre del firmware. Su longitud mínima debe ser 50.
*device	Puntero a una cadena de caracteres donde se devuelve el tipo de dispositivo para el que se generó el firmware. Su longitud mínima debe ser 50.

VOID ST_InfoMODFirmware (CHAR *date, CHAR *time, CHAR *name, CHAR *device)

Devuelve la información del firmware del dispositivo MODULO.

Parámetros

*date	Puntero a una cadena de caracteres donde se devuelve la fecha de compilación del firmware. Su longitud mínima debe ser 50.
*time	Puntero a una cadena de caracteres donde se devuelve la hora de compilación del firmware. Su longitud mínima debe ser 50.
*name	Puntero a una cadena de caracteres donde se devuelve el nombre del firmware. Su longitud mínima debe ser 50.
*device	Puntero a una cadena de caracteres donde se devuelve el tipo de dispositivo para el que se generó el firmware. Su longitud mínima debe ser 50.

7 Otras funciones

7.1 Funciones de acceso al sistema

INT ST_OpenSys (CHAR * path, USHORT num_channel, USHORT num_channel_mux)

Abre la comunicación con el sistema, carga el firmware y configura el sistema con los valores por defecto.

Parámetros

*path	Path donde se encuentran los archivos de inicialización: CYPRESS.BIX, UCI.BIT, FOC8_V2R2.BIT, BASE_V2R2.BIT. La barra inclinada '\' que separa los directorios debe ser doble "\", y el path siempre debe finalizar con estas barras. Por ejemplo ("..\BIN\", ó "C:\FILES\BIN\"). Si estos archivos se encuentran en el mismo path de la aplicación se debe pasar como parámetro ".".
num_channel	Número de canales de la apertura del sistema. Debe ser múltiplo de 32.
num_channel_mux	Número de canales multiplexados del sistema. Si no es multiplexado el valor será 0.

Devuelve (INT)

(Ver códigos de error)

INT ST_CloseSys (void)
Cierra la comunicación con el sistema y libera los recursos reservados por la librería.
Devuelve (INT)
(Ver códigos de error)

7.2 Funciones de adquisición de datos

LONG ST_GetBuffer (SHORT INT *bscan, SHORT INT *cscan, SHORT INT *dscan, LONG *encx, LONG *ency)	
Esta función devuelve la próxima imagen B-Scan pendiente de leer que se encuentra en el buffer del PC.	
Parámetros	
*bscan	Puntero a la zona de memoria donde se devuelve la imagen B-Scan. El tamaño de la memoria necesaria para recibir la imagen B-Scan se obtiene con la función ST_GetImageSizeAcq() .
*cscan	Puntero a la zona de memoria donde se devuelve la línea la imagen C-Scan correspondiente a esa adquisición. El tamaño de la memoria necesaria para recibir estos datos es el número de líneas de la imagen, y se obtiene en el segundo parámetro de la función ST_GetImageSizeAcq() La línea de la imagen C-Scan se corresponde con el valor de pico para cada A-Scan de la imagen, según la configuración de la puerta.
*dscan	Puntero a la zona de memoria donde se devuelve la línea la imagen D-Scan correspondiente a esa adquisición. El tamaño de la memoria necesaria para recibir estos datos es el número de líneas de la imagen, y se obtiene en el segundo parámetro de la función ST_GetImageSizeAcq() . La línea de la imagen D-Scan se corresponde con el número muestra en la que se detecta el pico para cada A-Scan de la imagen, según la configuración de la puerta.
*encx	Puntero a la variable donde se devuelve la posición del encoder A para esa adquisición. La posición del encoder se devuelve en número de pulsos.
*ency	Puntero a la variable donde se devuelve la posición del encoder B para esa adquisición. La posición del encoder se devuelve en número de pulsos.
Devuelve (INT)	
<ul style="list-style-type: none"> - Número de datos de la imagen devuelta en el parámetro 'bscan'. - (Ver códigos de error) 	

7.3 Funciones de configuración

INT ST_PRG (VOID)
Esta función programa en el sistema los parámetros cargados en memoria con las funciones del tipo LOAD.
Devuelve (INT)
Ver códigos de error

INT ST_SaveConfig (CHAR *data, LONG filepos)	
Guarda la configuración actual del sistema para poder ser cargada posteriormente con la función ST_LoadConfig() .	
Parámetros	
*data	Cadena de caracteres con el nombre del archivo en el que se desea guardar la configuración actual del equipo.
filepos	Posición, dentro del archivo, en la que se desea guardar la configuración.
Devuelve (INT)	
(Ver códigos de error)	

LONG ST_LoadConfig (CHAR *file, CHAR *date, LONG filepos)	
Carga en el sistema una configuración que ha sido guardada previamente con la función ST_SaveConfig() .	
Parámetros	
*file	Cadena de caracteres con el nombre del fichero de configuración que se desea cargar en el equipo.
*date	Puntero a una cadena de caracteres en la que se devuelve la fecha en la que se guardó la configuración que ahora se carga en el equipo. La longitud mínima reservada debe ser 15 bytes.
filepos	Posición, dentro del archivo, en la que se encuentra la configuración.
Devuelve (LONG)	
- Devuelve la posición del último valor leído del archivo de configuración. - (Ver códigos de error)	

7.4 Funciones de control del procesamiento

INT ST_Run (INT wait)	
Inicia el ciclo de procesamiento para la adquisición de una imagen B-Scan. Si existe alguna imagen en la memoria interna del equipo la lee y la almacena en el buffer de la memoria del PC para su posterior lectura con la función ST_GetBuffer() . En cada ejecución de esta función se inicia la adquisición una única imagen B-Scan.	
Parámetros	
wait	0 → (No bloqueante) No se espera el fin de procesamiento por parte del equipo. La función inicia el ciclo de procesamiento, si existe alguna imagen en la memoria interna del equipo la lee y la almacena en el buffer del PC, y finaliza su ejecución. 1 → (Bloqueante) Espera el fin de procesamiento por parte del equipo. La función inicia el ciclo de procesamiento, espera a que este finalice, si existe alguna imagen en la memoria interna del equipo la lee y la almacena en el buffer del PC, y finaliza su ejecución.
Devuelve (INT)	
(Ver códigos de error)	

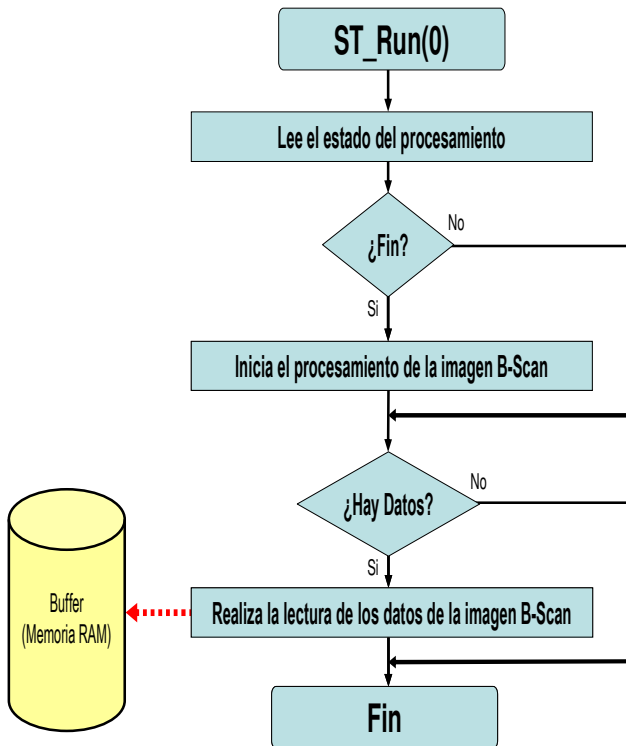


Fig. 5.- Diagrama de flujo de la ejecución de la función ST_Run(0)

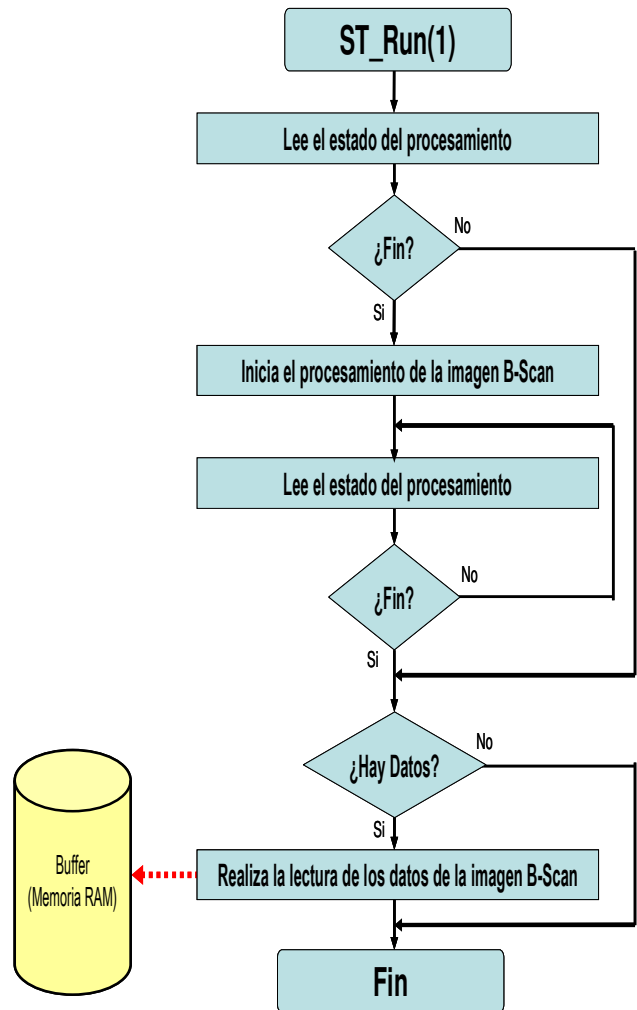


Fig. 6.- Diagrama de flujo de la ejecución de la función ST_Run(1)

INT ST_RunScanSW (INT time)	
Activa el estado de adquisición de imágenes B-Scan, en modo disparo por software continuo. Se inicia un hilo de procesamiento concurrente cuya ejecución es un bucle continuo en el que se ejecuta la función ST_Run(1) de forma continua con un tiempo de retardo entre una ejecución y la siguiente. Para salir del bucle es necesario ejecutar ST_Stop() . Mientras el sistema se encuentre en este estado no se podrá acceder a la programación de los parámetros de acceso directo o diferido. El estado en el que se encuentra el sistema se puede consultar con la función ST_GetTriggerSource() .	
Parámetros	
time (ms)	Tiempo de espera del bucle. Es el tiempo de espera entre una ejecución ST_Run(1) y la siguiente.
Devuelve (INT)	
(Ver códigos de error)	

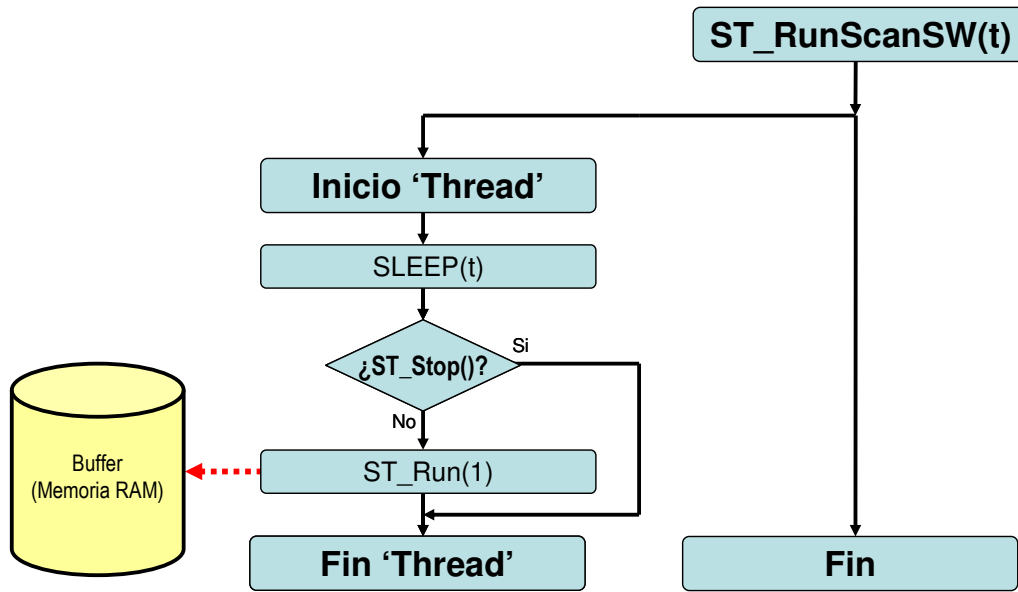


Fig. 7.- Diagrama de flujo de la ejecución de la función ST_RunScanSW().

INT ST_RunScanTRG (VOID)

Activa el estado de adquisición de imágenes B-Scan, en modo continuo con disparo por evento externo (TRIGGER IN). Se inicia un hilo de procesamiento concurrente cuya ejecución es un bucle continuo en el que se realiza la lectura de datos de la memoria interna del equipo de forma continua. Para salir del bucle es necesario ejecutar **ST_Stop()**. Mientras el sistema se encuentre en este estado no se podrá acceder a la programación de los parámetros de acceso directo o diferido. El estado en el que se encuentra el sistema se puede consultar con la función **ST_GetTriggerSource()**.

Devuelve (INT)

(Ver códigos de error)

INT ST_RunScanENC (VOID)

Activa el estado de adquisición de imágenes B-Scan, en modo continuo con disparo por encoder. Se inicia un hilo de procesamiento concurrente cuya ejecución es un bucle continuo en el que se realiza la lectura de datos de la memoria interna del equipo de forma continua. Para salir del bucle es necesario ejecutar **ST_Stop()**. Mientras el sistema se encuentre en este estado no se podrá acceder a la programación de los parámetros de acceso directo o diferido. El estado en el que se encuentra el sistema se puede consultar con la función **ST_GetTriggerSource()**.

Devuelve (INT)

(Ver códigos de error)

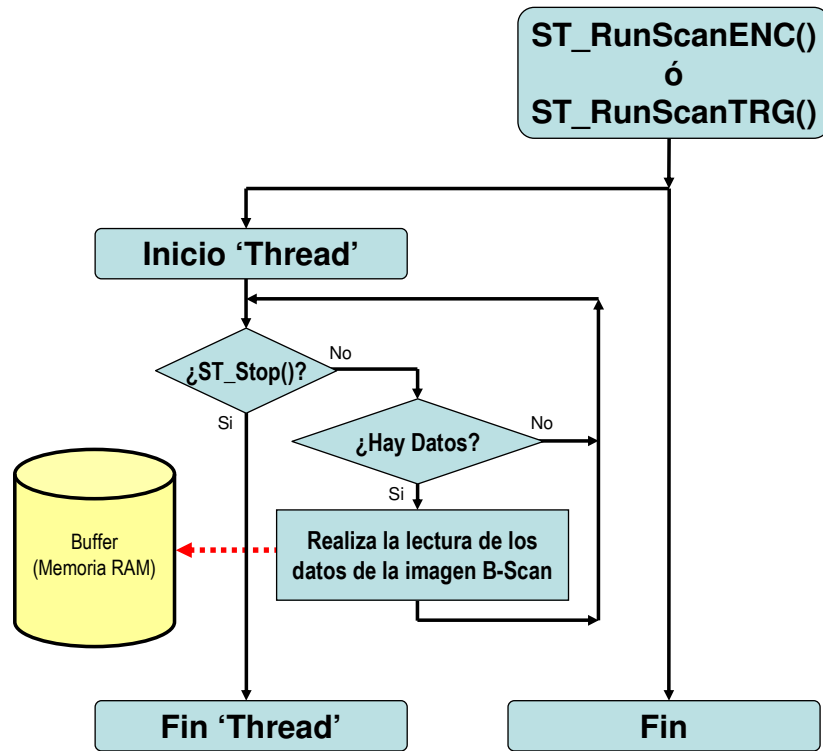


Fig. 8.- Diagrama de flujo de la ejecución de la función *ST_RunScanENC()* ó *ST_RunScanTRG()*.

INT ST_Stop (VOID)
Detiene la adquisición continua de imágenes B-Scan, que se produce al configurar el sistema con las funciones ST_RunScanSW() , ST_RunScanENC() y ST_RunScanTRG()
Devuelve (INT)
(Ver códigos de error)

8 Configuración inicial por defecto

Después de abrir la conexión con el SITAU y cargar el firmware (**ST_OpenSys()**), se carga una configuración por defecto que tiene los siguientes valores:

Ancho de pulso	100 ns
Número de pulsos	1
Frecuencia de muestreo	40 MHz
Velocidad del material	5.146 m/s
Ganancia	25 dB
PRF	3.500 Hz
Modo de adquisición	VIDEO (Envolvente)
'Phased Array Probe' PA	
Frecuencia	5 MHz
Pitch	0.50 mm
Nº Elementos	128 elementos
Distancia hasta el primer elemento	0 mm
Puerta	
Habilitada	
Inicio	10 mm
Fin	50 mm
Umbral	20 %
Filtro digital FIR	
Deshabilitado	
Frecuencia de corte inferior	0 MHz
Frecuencia de corte superior	20 MHz
Adquisición	
Inicio	0 mm
Fin	100 mm
Focalización	
Habilitada con foco fijo	
Distancia focal en emisión	18 mm
Distancia focal en recepción	18 mm
Scan	
Habilitado barrido lineal	

Apertura	32 elementos
Elemento inicial	1
Elemento final	128
Paso de barrido (desplazamiento de la apertura entre disparos)	1 elemento
Encoder	
Nº de pasos entre disparos	50

9 Tipos de datos y acceso a las funciones de la librería

Tipo	Tamaño (Bytes)	Rango
UCHAR	1	0 ... 255
CHAR	1	-128 ... 127
SHORT	2	-32.768 ... 32.767
UINT	4	0 ... 4.294.967.295
INT	4	-2.147.483.648 ... 2.147.483.647
ULONG	4	0 ... 4.294.967.295
LONG	4	-2.147.483.648 ... 2.147.483.647
FLOAT	4	$\pm 1,18 \cdot 10^{-38} \dots \pm 3,40 \cdot 10^{38}$
DOUBLE	8	$\pm 2,23 \times 10^{-308} \dots \pm 1,79 \times 10^{308}$
LONG DOUBLE	10	$\pm 3,37 \times 10^{-4932} \dots \pm 1,18 \times 10^{4932}$

IMPORTANTE

Todas las funciones están compiladas como "Standard call" en la convención de llamada a las funciones, por lo tanto el proyecto en el que se incluya la librería 'stlib.dll' debe estar configurado del mismo modo.

10 Códigos de error

Código	Definición
0	OK, resultado de la operación correcto, no se ha producido ningún error.
-1	Handle no nulo en OpenUsb.
-2	Nombre de fichero inválido en LoadFirmware_FX2.
-3	No se puede abrir el fichero del firmware en LoadFirmware_FX2.
-4	No se pudo cerrar el fichero en LoadFirmware_FX2.
-5	Nombre de fichero inválido en LoadFirmware_UCI.
-6	No se puede abrir el fichero del .bit de la UCI en LoadFirmware_UCI.
-7	No se pudo cerrar el fichero en LoadFirmware_UCI.
-8	Argumentos no válidos en inportBloque.
-9	Falla la cuenta de words_restantes en inportBloque.
-10	Argumentos no válidos en outportBloque.
-11	Argumentos no válidos en Ready.
-12	Argumentos no válidos en is_cypress_on.
-13	No se pudo reservar memoria para bufferdev en is_cypress_on.
-14	No se pudo abrir el fichero de log.
-15	Error en la llamada a CreateFile.
-16	Error en la llamada a CloseHandle.
-17	Error en la llamada a DeviceIoControl.
-18	No se ha encontrado ningún dispositivo USB válido.
-19	Error en el número de puerto del dispositivo.
-101	No se pudo reservar memoria en Escribir.
-102	Argumentos no válidos en Escribir.
-103	Argumentos no válidos en Leer.
-104	Argumentos no válidos en LoadFirmware.
-105	No se pudo abrir el fichero .bit en LoadFirmware.
-106	No se pudo cerrar el fichero .bit en LoadFirmware.
-201	Módulo incorrecto en Test_UCI.
-202	Argumentos no válidos en InsertarPrograma.
-203	El programa de la UCI es demasiado largo (InsertarEnPrograma).
-204	Argumentos no válidos en InicioSubrutina.

-205	No se puede iniciar otra subrutina, se ha alcanzado ya el máximo (InicioSubrutina).
-206	Argumentos no válidos en FijarInicioPrograma.
-207	Argumentos no válidos en CrearEscrituraMultipleRegistro.
-208	Argumentos no válidos en PRG_jump_set.
-209	Argumentos no válidos en CMD_sub_set.
-210	Argumentos no válidos en PRG_jump_cond_set.
-211	Argumentos no válidos en PRG_jump_c_get.
-212	Argumentos no válidos en PRG_sub_c_get.
-213	Argumentos no válidos en PRG_sub_rtn.
-214	Argumentos no válidos en PRG_set_wait.
-215	Argumentos no válidos en PRG_set_pro.
-301	Error de archivo, acceso al archivo incorrecto.
-302	Error de rango, la variable pasada como parámetro está fuera de rango.
-303	Error de número de módulo, el módulo al que se desea acceder esta fuera de rango.
-304	Error de número de canal, el canal al que se desea acceder esta fuera de rango.
-305	Error en la reserva de memoria.
-306	Error de desbordamiento de la memoria de parámetros.
-401	Error de archivo, acceso al archivo incorrecto.
-402	Error de rango, la variable pasada como parámetro esta fuera de rango.
-403	Error de número de base, la base a la que se desea acceder esta fuera de rango.
-404	Error de número de coeficiente del filtro, el número de coeficiente al que se desea acceder está fuera de rango.
-405	Error de memoria dinámica, memoria insuficiente.
-501	Error: Una base no se ha configurado correctamente.
-502	Error: Un módulo no se ha configurado correctamente.
-503	Error: Se ha llamado a ST_ConfigSys sin una llamada previa a ST_OpenSys.
-504	Error: Se ha llamado a alguna función sin una llamada previa a ST_ConfigSys.
-505	Error: Rango de datos no válido.
-506	Todo el buffer leído está compuesto por 0xCACA, implica que aún no se han adquirido datos válidos se ha leído demasiado pronto. Sólo es útil para la rutina de adquisición ST_Acq.
-507	Se ha salido de un bucle por timeout.
-508	Error número de canales, el numero de canales es distinto al configurado por la función ST_OpenSys.
-509	Error en la reserva de memoria.
-510	Error intento de disparo con cero muestras programado.
-511	Error trama incorrecta en el buffer del canal, la trama leída del buffer no tiene la palabra de control inicial.

-512	Error trama incorrecta en el buffer del canal, la trama leída del buffer no tiene la palabra de control final.
-513	Error desbordamiento de la memoria interna del equipo.
-514	Error trama incorrecta en el buffer del canal, la trama leída del buffer del canal indica que contiene un número de datos superior al tamaño de la trama.
-515	Error trama incorrecta en el buffer del canal, la trama leída del buffer del canal indica que contiene un número de datos distinto al programado.
-516	Error: Se ha llenado el buffer de adquisición.
-517	Error desbordamiento de buffer (durante la adquisición), el numero de datos leídos es superior al tamaño del buffer temporal de adquisición.
-518	Error de archivo, acceso al archivo incorrecto.
-519	Se va a intentar dividir por cero. Se aborta la operación.
-520	Intento de lectura de Buffer vacío.
-521	Error: Se ha llamado a alguna función sin una llamada previa a ST_ConfigSys()
-522	Error de Disparo: El sistema se encuentra en un estado de barrido por encoder ST_RunScanENC()
-523	Error de Programación: Intento de programación mientras el sistema encuentra en un estado de barrido configurado con ST_RunScanSW() o ST_RunScanENC()

11 Esquema de programación

11.1 Disparo por software - ST_Run()

Los pasos para programar utilizando la librería SITAU (stlib.dll) son los siguientes:

1. Abrimos la conexión con el equipo y la carga del firmware:

ST_OpenSys (path, canales_activos, canales_mux)

2. Se programan los parámetros de configuración necesarios para la adquisición con las funciones del tipo:

ST_SetXXXXX ()
 ó
ST_LoadConfig ()
ST_PRG ()
 ó
ST_LoadConfig ()

3. Comienza el bucle de adquisición:

- 3.1. Se realiza el inicio del procesamiento de la imagen y la transferencia de todos los datos que se encuentran en la memoria interna del SITAU al buffer de la memoria RAM del PC:

ST_Run (1)

3.2. Se comprueba si hay alguna imagen para leer en el buffer de la memoria RAM del PC:

```
size = ST_GetImageSizeAcq()
```

3.3. Si existen datos en el buffer los leemos:

```
if (size > 0) ST_GetBuffer()
```

3.4. Volvemos a ejecutar el inicio del bucle de adquisición 3.1

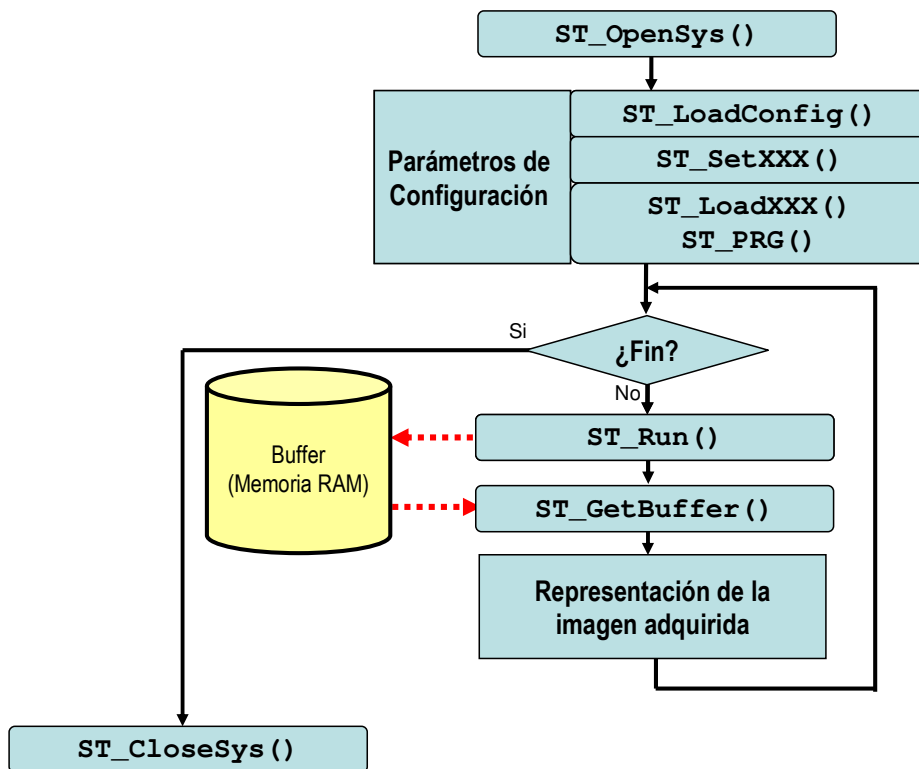


Fig. 9.- Diagrama de flujo de una aplicación en la que el disparo se realiza por software ST_Run().

11.2 Scan con procesamiento concurrente Multi-Thread

(ST_RunScanSW(), ST_RunScanENC(), ST_RunScanTRG())

Los pasos para programar utilizando la librería SITAU (stlib.dll) son los siguientes:

4. Abrimos la conexión con el equipo y la carga del firmware:

```
ST_OpenSys(path, canales_activos, canales_mux)
```

5. Se programan los parámetros de configuración necesarios para la adquisición con las funciones del tipo:

```
ST_SetXXXXX ()  
ó  
ST_LoadConfig ()  
ST_PRG ()  
ó  
ST_LoadConfig ()
```

6. Se inicia el hilo de ejecución para la realización del procesamiento y la adquisición de la imagen en paralelo con la lectura y la representación de la misma:

```
ST_RunScanSW (delay)  
ó  
ST_RunScanENC ()  
ó  
ST_RunScanTRG ()
```

7. Comienza el bucle de lectura de la imagen:

- 7.1. Se comprueba si hay alguna imagen para leer en el buffer de la memoria RAM del PC:

```
size = ST_GetImageSizeAcq()
```

- 7.2. Si existen datos en el buffer los leemos:

```
if (size > 0) ST_GetBuffer()
```

- 7.3. Representación de los datos leídos

- 7.4. Volvemos a ejecutar el inicio del bucle de adquisición 3.1

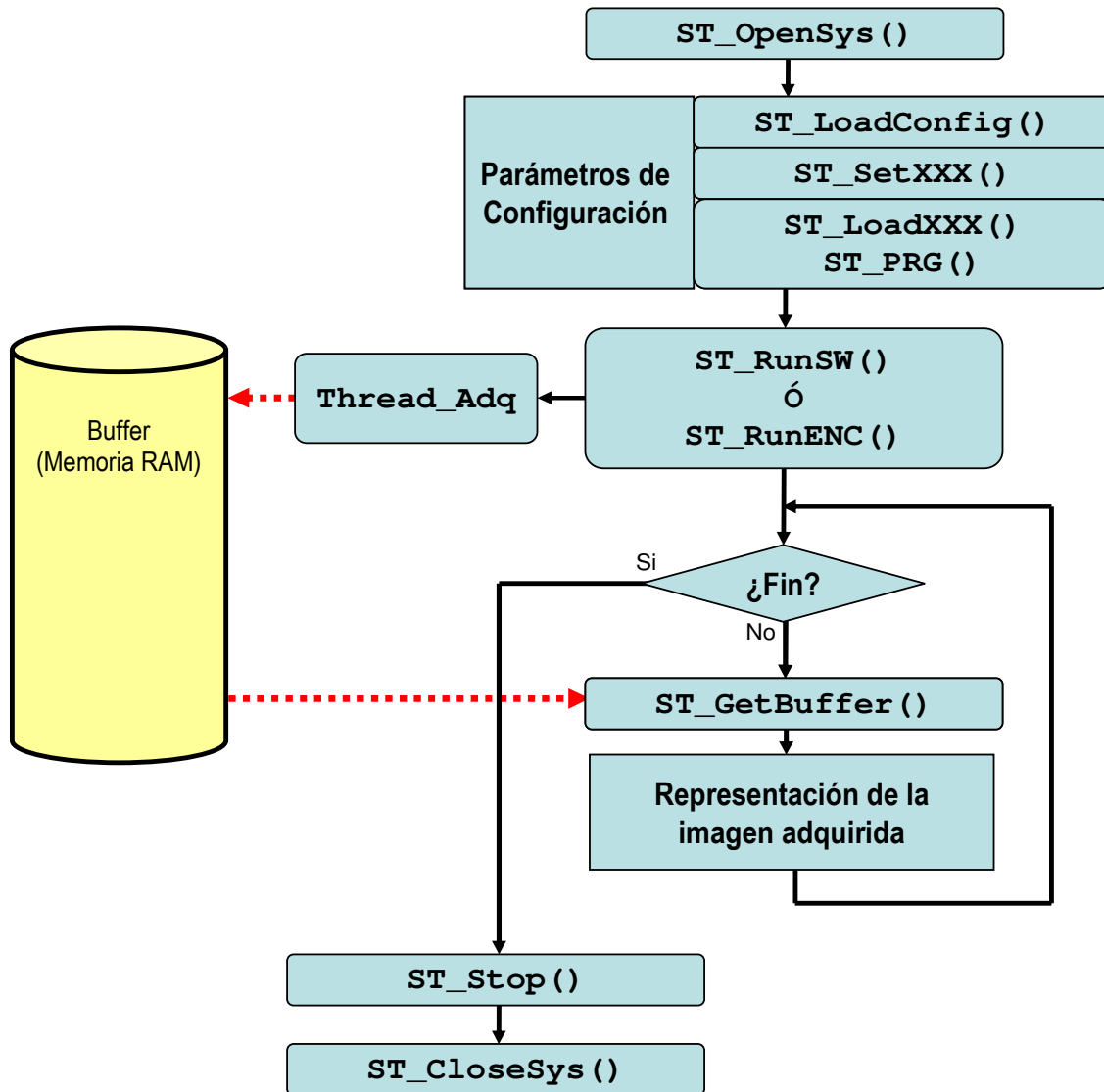


Fig. 10.- Diagrama de flujo de una aplicación en la que la adquisición se realiza en un hilo de ejecución concurrente con la representación de los datos - `ST_RunSW()` ó `ST_RunENC()`