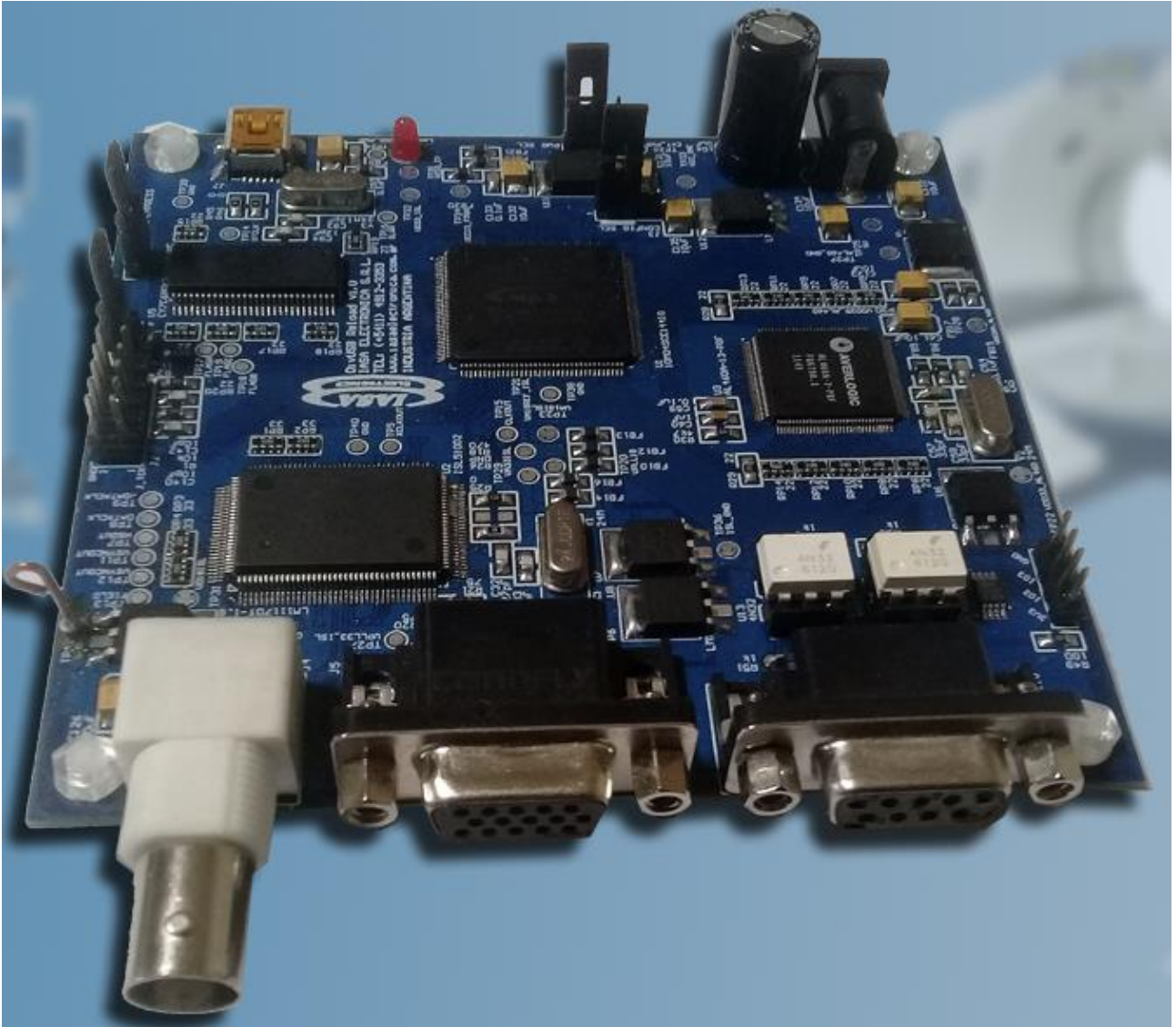


DivUSB-RE



MANUAL DE USO





ÍNDICE

<i>DivUSB-RE</i>	1
<i>ÍNDICE</i>	2
DESCRIPCIÓN	3
ESPECIFICACIONES.....	4
CONECTORES	5
ENTRADA USB.....	5
ALIMENTACIÓN.....	5
ENTRADA DE VIDEO COMPUESTA (BNC).....	5
ENTRADA VGA (DB15-HD)	6
ENTRADAS OPTOACOPLADAS (DB9)[OPCIONAL]	6
SDK	7
LIBRERIA DIV USB RELOAD - API	7
DRIVER	14
PROGRAMA EJEMPLO	14



DESCRIPCIÓN

Placa de captura de video multisincrónica por puerto USB 2.0 apta para captura de imágenes y videos en equipos de RX tales como: Arcos en C, Mesas Telecomandadas, Seriógrafos, etc.

También es ideal para construir Sistemas de Captura Secundaria DICOM en equipos de TC y RMI.

Permite adquirir imágenes digitalizando la señal de video compuesto o VGA del equipo y almacenarlas en el disco rígido de la PC.

Es una placa analógica de alta resolución de gran flexibilidad que permite adaptarla a cualquier sistema de TV, de alta o baja resolución, con barrido entrelazado o progresivo.

Tiene un amplio rango de sincronización, aceptando señales de video con sincronismo horizontal desde 60 uSeg hasta 13 uSeg y vertical desde 50 hz modo entrelazado hasta 100 hz modo progresivo.

Se puede conectar a cualquier PC bajo Windows por medio de un puerto USB 2.0 con alimentación interna (según la PC) o externa (recomendado).

Permite adquirir imágenes estáticas y videos con resoluciones de hasta 1920x1080 x 8 bits(FULL HD).

Tiene 2 puertos de comando con entradas optoacopladas para el manejo de pedales, botones o recepción de señales de control de los equipos.

El paquete OEM incluye los drivers para Windows XP, Windows 7, Windows 8 y Windows 10, mas programas de demostración en C#, C++ y VB.Net.

La placa de captura DIVUSB-RE es totalmente compatible con los sistemas de captura y procesamiento digital de imágenes RCAP V6.0 para equipos de CT y RMI y MEMORAD Q-A para equipos de RX.

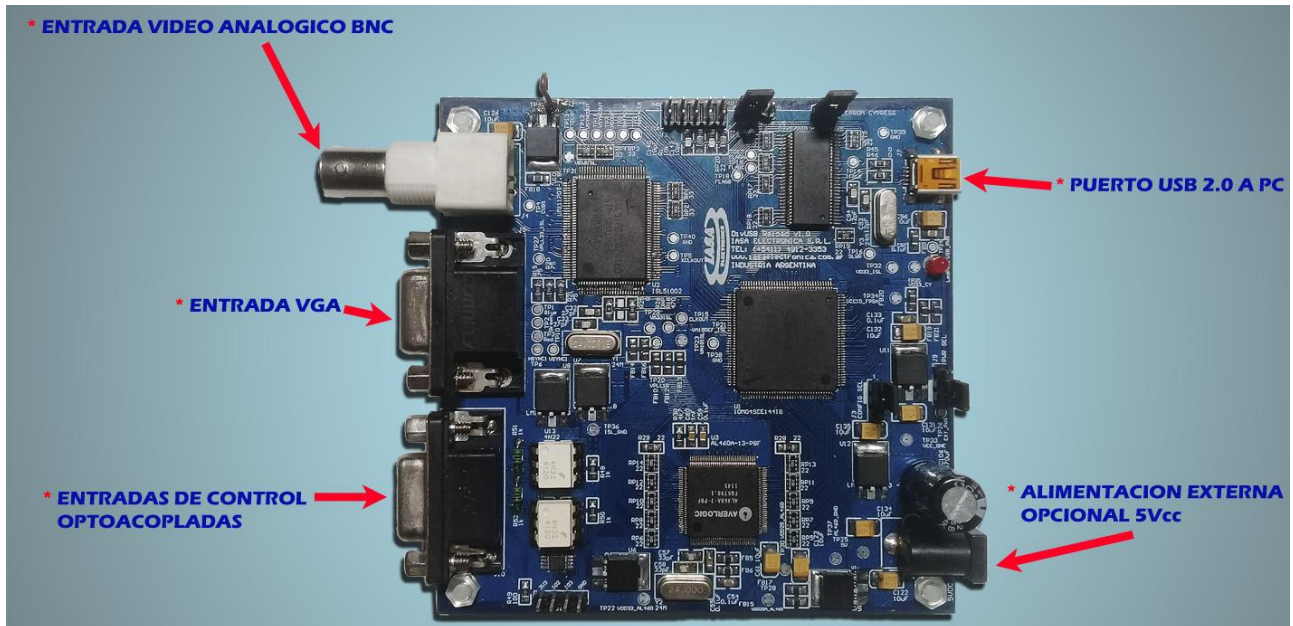
Nota: El software y hardware adicional para implementar estos sistemas utilizando esta placa no están incluidos en el paquete OEM y deben ser adquiridos por separado.



ESPECIFICACIONES

Característica	Especificación
Resolución de Conversión:	8 bits, 256 niveles de gris
Matriz de captura máxima:	1920 x 1080 x 8 bits:
Reloj de Píxel máximo:	165 MHz
Normas de Video de entrada:	Video entrelazado o progresivo hasta 165 MHz de reloj de píxel, sincronismo compuesto o separado
Matriz de adquisición máxima:	1920 x 1080 x 8 bits
Entradas de Video	Una entrada de Video compuesto analógico tipo BNC y una entrada tipo VGA; terminadas en 75 ohms
Memoria de almacenamiento	8 MB x 16 bits FIFO
Restauración de nivel de continua:	automático o programable
Ganancia y offset de entrada:	Programable
Interface a PC:	Puerto USB 2.0.
Rango de frecuencia horizontal admitida:	De 10 khz a 100 khz
Rango de frecuencia vertical admitida:	Hasta 100 hz en modo progresivo
Resolución de conversión:	8 bits, 165 Mhz
Nivel de video de entrada máximo:	2Vpp
Velocidad de transferencia PC	30 MB/s
Velocidad de adquisición de imágenes en norma CCIR 640x574x8, 50 Hz, modo entrelazado:	25FPS
Velocidad de adquisición de imágenes en Full HD 1920x1080x8, 60 Hz, modo progresivo:	8FPS
Puertos de control de entrada:	2 puertos optoacoplados para control de grabación de imágenes fijas y videos.
Alimentación:	500 mA. Fuente externa de 5 VCC 1A(incluida) o con la fuente interna del USB.
Dimensiones:	100mm x 100mm.
Peso:	50 Gr.

CONECTORES



ENTRADA USB

Requiere un cable mini-USB (tipo B) a USB (tipo A) para conectar la capturadora a la PC.

ALIMENTACIÓN

La placa necesita de una fuente de alimentación de 5V@1000mA conector Jack DC 2,5mm.



ENTRADA DE VIDEO COMPLETA (BNC)

Utilizada para una señal de video compuesta con una impedancia intrínseca de 75ohms. Señales típicas (entrelazadas):

- CCIR: 640x576@25Hz
- RS170: 640x496@30Hz
- HR1249E: 1024x1024@25Hz

Otras señales que cumplan con los límites absolutos de la capturadora descritos previamente en las [especificaciones](#), serán compatibles, incluso aquellas que sean progresivas.

Nota: la resolución a utilizar deberá ser múltiplo de 512. Si la señal posee una resolución no divisible por 512, se deberá reducir o aumentar el tamaño de la resolución para que el resultado cumpla con este requisito.

ENTRADA VGA (DB15-HD)

Utilizada para una señal de video VGA (RGB + Vsync + Hsync) con una impedancia intrínseca de 75ohms. Señales típicas (progresivas):

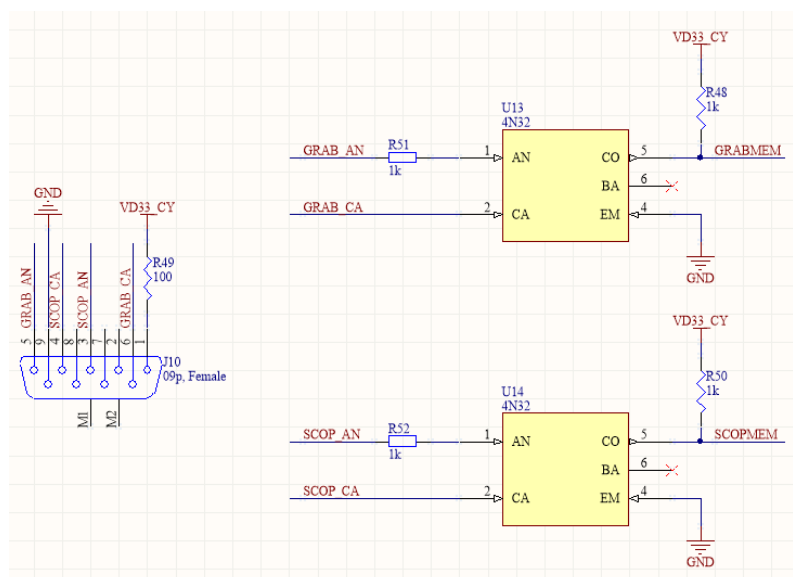
- 640x480@60Hz
- 800x600@60~85Hz
- 1024x768@60~85Hz
- 1280x720@60Hz
- 1280x768@60Hz
- 1280x800@60Hz
- 1280x960@60~85Hz
- 1280x1024@60~85Hz
- 1366x768@60Hz
- 1400x1050@60Hz
- 1440x900@60Hz
- 1600x900@60Hz
- 1600x1200@60Hz
- 1680x1050@60Hz
- 1920x1080@60Hz

Otras señales que cumplan con los límites absolutos de la capturadora descritos previamente en las [especificaciones](#), serán compatibles.

Nota: la resolución a utilizar deberá ser múltiplo de 512. Si la señal posee una resolución no divisible por 512, se deberá reducir o aumentar el tamaño de la resolución para que el resultado cumpla con este requisito. Ej.: si la resolución de la señal fuera de 1440x900 se podría adoptar otra de 1440x912.

ENTRADAS OPTOACOPLADAS (DB9) [OPCIONAL]

Utilizadas para la interfaz de usuario mediante una pedalera o un switch de control.



SDK

Se incluye:

- **LibreriaDivUSBReload:** librería que permite manipular la capturadora mediante una DLL compatible para C# y Visual Basic.NET.
- **DivUSBReloadDemoCS:** ejemplo del uso de la capturadora usando C#.
- **DivUSBReloadDemoVB:** ejemplo del uso de la capturadora usando Visual Basic.NET.
- **DivUSBReloadDemoMFC:** ejemplo del uso de la capturadora usando C++ mediante MFC.

Se requieren las siguientes librerías:

- **ITEX41:** necesaria para mostrar imagen en pantalla. **Incluida**
- **OpenCv:** necesaria para procesar imagen (se utiliza la versión 2.49). **No incluida**

LIBRERIA DIV USB RELOAD - API

▪ `double` **GetVersion()**

Descripción: obtiene la versión de la librería.

Argumentos: ninguno.

Devuelve: la versión de la librería.

▪ `int` **InicializarDivUSBReload()**

Descripción: inicializa la capturadora.

Argumentos: ninguno.

Devuelve:

- 1: Inicialización correcta.
- -1 o -2: Error al reservar memoria.
- -3: No se encontró la capturadora.

▪ `void` **CerrarDivUSBReload()**

Descripción: cierra la capturadora.

Argumentos: ninguno.

Devuelve: nada.

▪ `int` **LeerRegistroPCA9536()**

Descripción: obtiene el estado de las entradas optoacopladas.

Argumentos: ninguno.

Devuelve: el estado de los puertos. Los puertos son activos bajos.

- Bit 0: ScopMem
- Bit 1: GrabMem

▪ `int LeerRegistroSL51002(BYTE registro)`

Descripción: obtiene el estado de un registro del conversor A/D de video.

Argumentos: el registro a leer.

Devuelve: el estado del registro.

▪ `int GetEstadoSL()`

Descripción: obtiene el estado de los registros más importantes del conversor A/D de video. Es necesario su ejecución, previo a obtener los estados del conversor en forma individual.

Argumentos: ninguno.

Devuelve:

- 1: lectura de estado correcta.
- -1: error al obtener el estado del conversor A/D de video.

▪ `double GetTiempoHorizontalSL()`

Descripción: obtiene el tiempo del sincronismo horizontal de la señal de video.

Requiere previamente ejecutar `GetEstadoSL()`.

Argumentos: ninguno.

Devuelve: el tiempo del sincronismo horizontal en uS.

▪ `double GetTiempoVerticalSL()`

Descripción: obtiene el tiempo del sincronismo vertical de la señal de video.

Requiere previamente ejecutar `GetEstadoSL()`.

Argumentos: ninguno.

Devuelve: el tiempo del sincronismo horizontal en mS.

▪ `unsigned int GetGreenGainRegSL()`

Descripción: obtiene el registro de la ganancia del color verde (usado como B/W).

Requiere previamente ejecutar `GetEstadoSL()`.

Argumentos: ninguno.

Devuelve: el valor del registro.

▪ `double GetGreenGainISL()`

Descripción: obtiene la ganancia del color verde (usado como B/W) en unidades dB. Requiere previamente ejecutar `GetEstadoSL()`.

Argumentos: ninguno.

Devuelve: el valor del registro convertido a valores en dB.

▪ `unsigned int GetGreenOffsetISL()`

Descripción: obtiene el offset del color verde (usado como B/W). Requiere previamente ejecutar `GetEstadoSL()`.

Argumentos: ninguno.

Devuelve: el valor del registro.

- `unsigned int GetPixelClockISL()`

Descripción: obtiene el pixel clock del conversor A/D de video. Requiere previamente ejecutar `GetEstadoISL()`.

Argumentos: ninguno.

Devuelve: el valor del registro en píxeles.

- `unsigned int GetDeStartISL()`

Descripción: obtiene el back porch horizontal del conversor A/D de video. Requiere previamente ejecutar `GetEstadoISL()`.

Argumentos: ninguno.

Devuelve: el valor del registro en píxeles.

- `unsigned int GetDeWidthISL()`

Descripción: obtiene el tamaño de la zona activa horizontal del conversor A/D de video. Requiere previamente ejecutar `GetEstadoISL()`.

Argumentos: ninguno.

Devuelve: el valor del registro en píxeles.

- `unsigned int GetLineStartISL()`

Descripción: obtiene el back porch vertical del conversor A/D de video. Requiere previamente ejecutar `GetEstadoISL()`.

Argumentos: ninguno.

Devuelve: el valor del registro en píxeles.

- `unsigned int GetLineWidthISL()`

Descripción: obtiene el tamaño de la zona activa vertical del conversor A/D de video. Requiere previamente ejecutar `GetEstadoISL()`.

Argumentos: ninguno.

Devuelve: el valor del registro en píxeles.

- `unsigned int GetEntrelazadoISL()`

Descripción: averigua si la señal de video de entrada es entrelazada o progresiva. Requiere previamente ejecutar `GetEstadoISL()`.

Argumentos: ninguno.

Devuelve: el valor del registro en píxeles.

- 1: señal entrelazada.
- 0: señal progresiva.

- `unsigned int GetVideoStatusISL()`

Descripción: obtiene el estado de la señal de video de entrada. Requiere previamente ejecutar `GetEstadoISL()`.

Argumentos: ninguno.

Devuelve: el valor del registro.

▪ `unsigned int GetVideoActivityISL()`

Descripción: obtiene la presencia de señal de video en la entrada BNC y VGA. Requiere previamente ejecutar `GetEstadoISL()`.

Argumentos: ninguno.

Devuelve: el valor del registro.

- Nibble bajo: BNC.
- Nibble alto: VGA.

▪ `int EscribirRegistroISL51002(BYTE registro, BYTE dato)`

Descripción: escribe un registro del conversor A/D de video.

Argumentos:

- `BYTE registro`: registro a escribir.
- `BYTE dato`: dato a escribir.

Devuelve: el valor del registro.

- 1: escritura de registro correcta.
- -1: error al escribir el registro.

▪ `int SetISL()`

Descripción: setea los registros más importantes del conversor A/D de video. Es necesario su ejecución, luego de setear los registros del conversor en forma individual.

Argumentos: ninguno.

Devuelve:

- 1: escritura de registros correcta.
- -1: escritura de registros errónea.

▪ `void SetInputVideo(unsigned char input)`

Descripción: setea la entrada del conversor A/D de video. Requiere la ejecución de `SetISL()` luego del seteo.

Argumentos: selección de la entrada.

- 0: BNC.
- 1: VGA.

Devuelve: nada.

▪ `void SetSyncSource(unsigned char sync_source)`

Descripción: setea la fuente de sincronismo del conversor A/D de video. Requiere la ejecución de `SetISL()` luego del seteo.

Argumentos: selección de la fuente de sincronismo.

- Bit 0: automático (0)/manual (1).
- Bit 1: HSYNC (0)/SOG (1).
- Bit 2: VSYNC (0)/ Sync Separator Output(1).

Devuelve: nada.

- `void SetGreenGainReg(unsigned int green_gain_reg)`

Descripción: setea el registro de la ganancia de color verde (usado como B/W) del conversor A/D de video. Requiere la ejecución de SetISL() luego del seteo.

Argumentos: valor de la ganancia (0x00xx-0xffff). Requiere un desplazamiento de 8 bits.

Devuelve: nada.

- `void SetGreenOffset(unsigned int green_offset)`

Descripción: setea el registro del offset de color verde (usado como B/W) del conversor A/D de video. Requiere la ejecución de SetISL() luego del seteo.

Argumentos: valor del offset (0x00xx-0xffff). Requiere un desplazamiento de 8 bits.

Devuelve: nada.

- `void SetPixelClock(unsigned int pixel_clock)`

Descripción: setea el registro del pixel clock del conversor A/D de video. Requiere la ejecución de SetISL() luego del seteo.

Argumentos: valor del pixel clock en píxeles.

Devuelve: nada.

- `void SetDeStart(unsigned int de_start)`

Descripción: setea el registro del back porch horizontal del conversor A/D de video. Requiere la ejecución de SetISL() luego del seteo.

Argumentos: valor del back porch horizontal en píxeles.

Devuelve: nada.

- `void SetDeWidth(unsigned int de_width)`

Descripción: setea el registro de la zona activa horizontal del conversor A/D de video. Requiere la ejecución de SetISL() luego del seteo.

Argumentos: valor de la zona activa horizontal en píxeles.

Devuelve: nada.

- `void SetLineStart(unsigned int line_start)`

Descripción: setea el registro del back porch vertical del conversor A/D de video. Requiere la ejecución de SetISL() luego del seteo.

Argumentos: valor del back porch vertical en píxeles.

Devuelve: nada.

- `void SetLineWidth(unsigned int line_width)`

Descripción: setea el registro de la zona activa vertical del conversor A/D de video. Requiere la ejecución de SetISL() luego del seteo.

Argumentos: valor de la zona activa vertical en píxeles.

Devuelve: nada.

▪ `void SetAfeBW(unsigned char afe_bw)`

Descripción: setea el registro del ancho de banda del conversor A/D de video.
Requiere la ejecución de SetISL() luego del seteo.
Argumentos: valor del ancho de banda (0-15).
Devuelve: nada.

▪ `void SetCampoImpar(unsigned char campo_impar)`

Descripción: setea la inversión de campo durante el entrelazado.
Argumentos:

- 0: Sin inversión.
- 1: Con inversión.

Devuelve: nada.

▪ `void SetHandleVentana(int* puntero_ventana)`

Descripción: setea el handler del elemento gráfico a utilizar para mostrar la imagen en pantalla.
Argumentos: handler del elemento gráfico.
Devuelve: nada.

▪ `void SetHandleVentanaVB(int puntero_ventana)`

Descripción: setea la dirección del elemento gráfico a utilizar para mostrar la imagen en pantalla. **(Utilizar solo en VB)**
Argumentos: dirección del elemento gráfico.
Devuelve: nada.

▪ `int MostrarImagenDivUSBReload(int tipo_video, int flag_grabar_video, int width_in, int height_in, int width_out, int height_out)`

Descripción: permite mostrar en pantalla la imagen capturada en vivo.
Argumentos:

- `int tipo_video`:
 - 0: Tomar imagen entrelazada.
 - 1: Tomar imagen progresiva.
 - 2: Tomar imagen progresiva en HD. **(usado en señales cuyo pixel clock es igual o mayor a 96MHz)**
- `int flag_grabar_video`: indica si se está grabando un video mientras se muestra en pantalla **(grabación previamente inicializada)**.
- `int width_in`: ancho de la imagen capturada.
- `int height_in`: alto de la imagen capturada.
- `int width_out`: ancho de la imagen de salida.
- `int height_out`: alto de la imagen de salida.

Devuelve:

- 1: la imagen se muestra correctamente.
- -1: error al mostrar imagen.

▪ **int GuardarUltimaImagenDivUSBReload(char nombre_archivo[])**

Descripción: permite almacenar la última imagen capturada.

Argumentos: directorio y nombre de la imagen a guardar.

Devuelve:

- 1: la imagen se guardó correctamente.
- -1: error al guardar imagen.

▪ **int GuardarUltimaImagenDivUSBReloadVB()**

Descripción: permite almacenar la última imagen capturada en el directorio del ejecutable con el nombre "Imagen_temp.bmp". **(Usar en VB)**

Argumentos: ninguno.

Devuelve:

- 1: la imagen se guardó correctamente.
- -1: error al guardar imagen.

▪ **int InicializarVideoWriter(char nombre_archivo[], float fps)**

Descripción: permite inicializar la grabación de video.

Argumentos:

- **char nombre_archivo[]**: directorio y nombre del video a guardar.
- **float fps**: la cantidad de cuadros por segundos que tendrá el video.

Devuelve:

- 1: la inicialización se completó correctamente.
- -1: error al inicializar la grabación.

▪ **int InicializarVideoWriterVB(float fps)**

Descripción: permite inicializar la grabación de video en el directorio del ejecutable con el nombre "Video_temp.avi". **(Usar en VB)**

Argumentos:

- **float fps**: la cantidad de cuadros por segundos que tendrá el video.

Devuelve:

- 1: la inicialización se completó correctamente.
- -1: error al inicializar la grabación.

▪ **void CerrarGrabacion()**

Descripción: cierra el video.

Argumentos: ninguno.

Devuelve: nada.

DRIVER

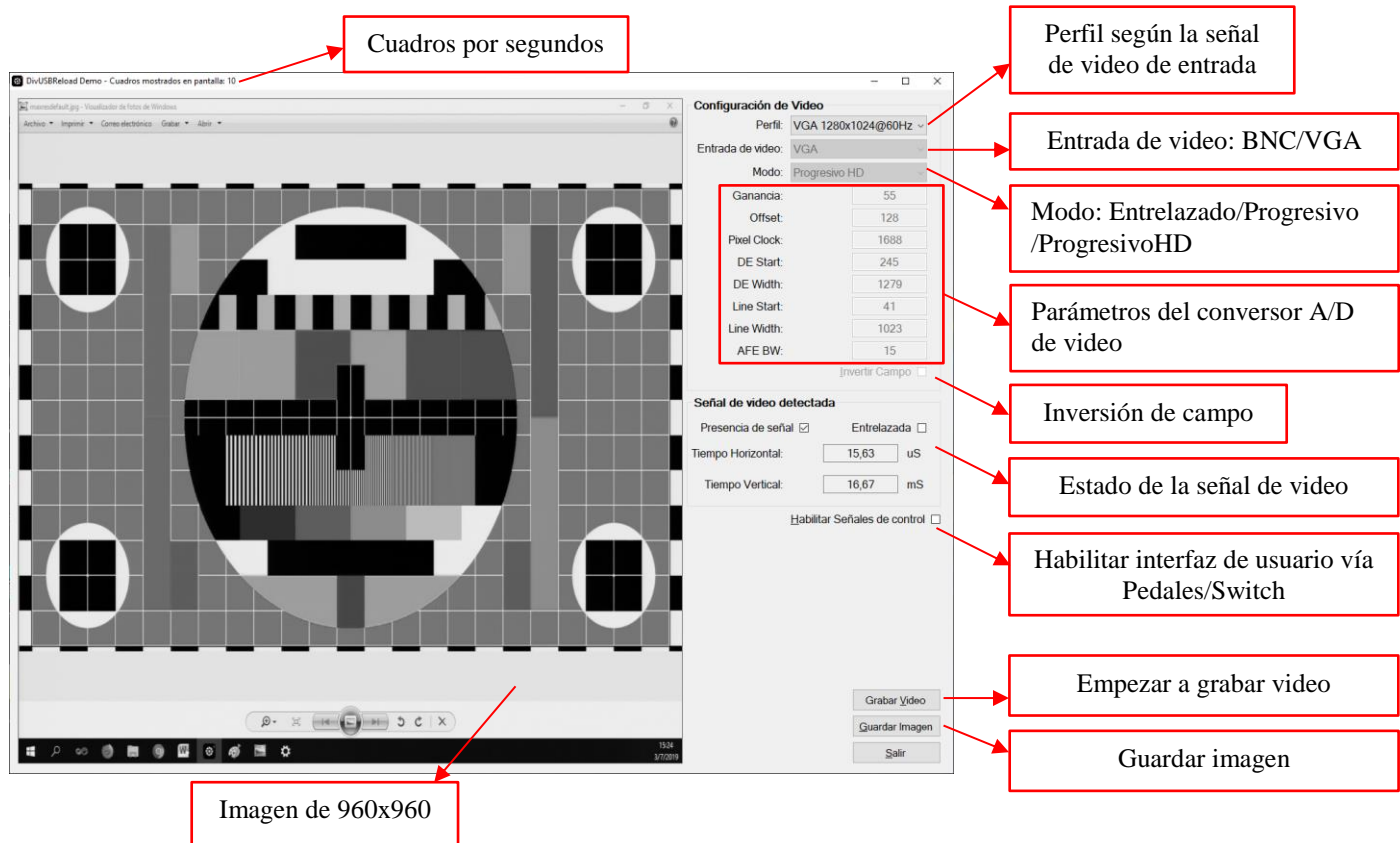
La capturadora previamente a ser utilizada requiere instalar el driver “cyusb3.inf”. Los pasos de instalación son los siguientes:

1. Ir al administrador de dispositivos.
2. Presionar el botón derecho del mouse sobre el dispositivo desconocido y seleccionar “Instalar/Actualizar driver”.
3. Seleccionar “Buscar software de controlador en el equipo”.
4. Elegir una lista de controladores disponibles en el equipo.
5. Mostrar todos los dispositivos.
6. Utilizar Disco.
7. Examinar y buscar el archivo “cyusb3.inf” según la versión del SO instalada.

PROGRAMA EJEMPLO

Hecho a partir de la “[LibreriaDivUSBReload](#)”, se entrega el programa ejemplo “DivUSBReloadDemo.exe” que permite:

- Configurar la capturadora con diferentes perfiles o en forma manual.
- Ver el estado de la señal de video según la entrada seleccionada.
- Mostrar imagen en pantalla.
- Guardar última imagen.
- Grabar un video.
- Usar las señales de control optoacopladas para mostrar imagen (ScopMem) y guardarla (GrabMem).



The screenshot shows the DivUSBReload Demo application interface. It features a central video preview window displaying a test pattern. To the right is a configuration panel with various settings. Red boxes and arrows highlight specific features and settings:

- Cuadros por segundos**: Points to the frame rate control in the video preview window.
- Perfil según la señal de video de entrada**: Points to the profile selection dropdown in the configuration panel.
- Entrada de video: BNC/VGA**: Points to the video input selection dropdown.
- Modo: Entrelazado/Progresivo /ProgresivoHD**: Points to the video mode selection dropdown.
- Parámetros del convertor A/D de video**: Points to the A/D converter parameters table.
- Inversión de campo**: Points to the field inversion checkbox.
- Estado de la señal de video**: Points to the signal status section.
- Habilitar interfaz de usuario vía Pedales/Switch**: Points to the control interface enable checkbox.
- Empezar a grabar video**: Points to the 'Grabar Video' button.
- Guardar imagen**: Points to the 'Guardar Imagen' button.
- Imagen de 960x960**: Points to the video preview window.

Configuración de Video	
Perfil:	VGA 1280x1024@60Hz
Entrada de video:	VGA
Modo:	Progresivo HD
Ganancia:	55
Offset:	128
Pixel Clock:	1688
DE Start:	245
DE Width:	1279
Line Start:	41
Line Width:	1023
AFE BW:	15