Description

CambridgeIC's Central Tracking Unit (CTU) chips work with sensors built from PCBs to measure the position of contactless targets using resonant inductive sensing technology.

CTU chips are designed to be embedded inside electromechanical products, and communicate with a host system processor over an SPI interface.

To assist demonstration, evaluation and development, CambridgeIC provides hardware and software that works with a PC. A CTU Adapter enables a PC to communicate with CTU chips by converting between SPI and USB interfaces. This document is a User Guide for the software.

Applications

- ***CTU Demo*** for displaying measurements
- ***CTU Repeat Average*** for extra precision and stats
- ***Save to File*** for exporting data
- ***Update Adapter Firmware***
- ***Update CTU Firmware***
- ***CTU Control Panel*** for configuring, manipulating and saving CTU settings

System Requirements

- Windows 10

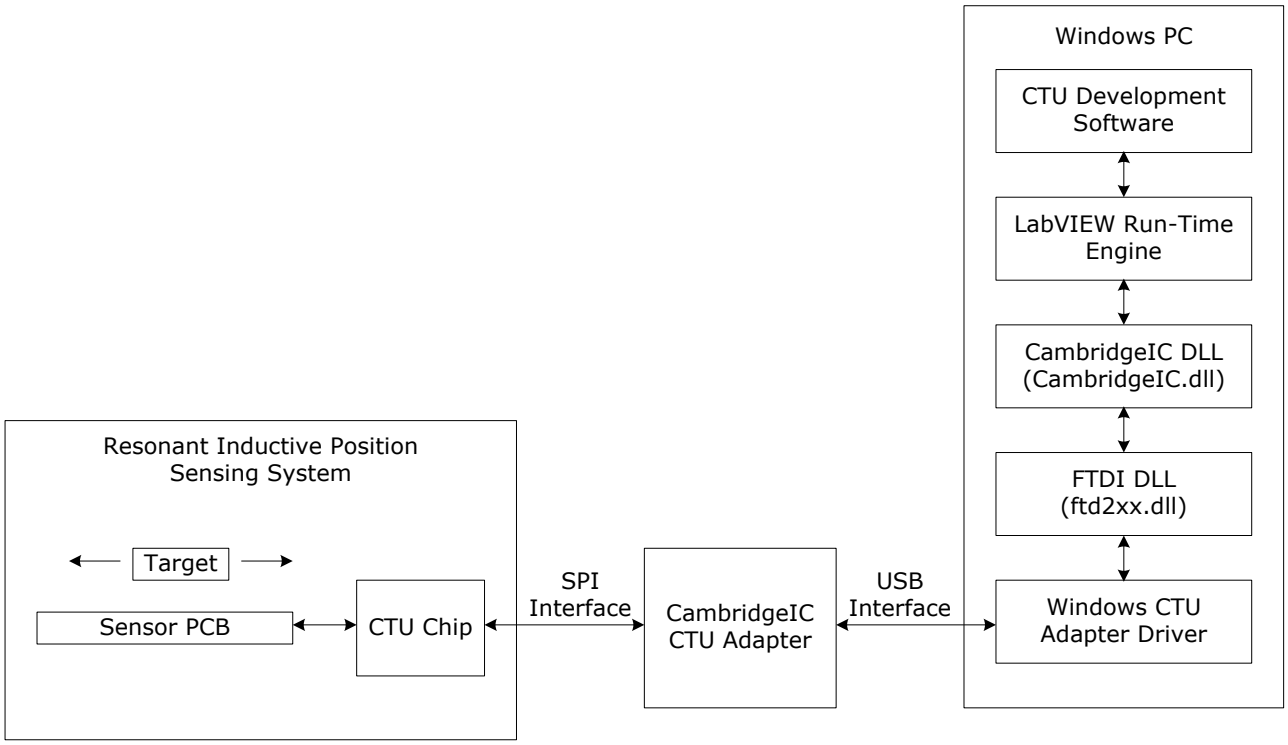


Figure 1 System block diagram

© Cambridge Integrated Circuits Ltd 2009-2022

1 Background

1.1 Software Platform

The software architecture is illustrated in the Windows PC box of Figure 1.

The applications described in this document are coded using National Instruments' LabVIEW graphical programming language. They are built as stand-alone applications, so that it is not necessary to have LabVIEW installed to run them. They require only the Run-Time Engine, which is included in the installer (section 3).

The LabVIEW Virtual Instruments (VIs) used to build the applications are also available for customers who have LabVIEW. This enables a customer to rapidly develop their own demonstration applications and to integrate with their own test equipment in a single programming environment. These Vis are documented in the CTU LabVIEW VI User Guide.

The applications use two DLLs for communicating with the CTU Adapter. The higher level one is ***CambridgeIC.DLL***, and this contains all of the functions required to send data to and from a CTU chip over its SPI interface, together with functions for controlling the Adapter itself. The Adapter contains a chip for USB communication from FTDI. This has its own DLL, ***ftd2xx.DLL***, which must also be loaded on the PC to enable CambridgeIC.DLL to communicate with the Adapter. They are also included in the installer.

If a customer wishes to develop their own PC applications for communicating with the CTU through the CTU Adapter, using programming languages such as Visual Basic or C, the DLLs can be used for communication. Please refer to the CTU Adapter Class Library User Manual for documentation on this approach. If available, it is recommended to use LabVIEW, since the CTU LabVIEW VIs allow programming at a higher level.

The Adapter requires a Windows driver, whose installation is described in the next main section.

1.2 Document Conventions

The names of CTU software applications and DLLs are highlighted like this: ***CTU Demo***.

The names of window controls and indicators are shown like this: **Next**.

© Cambridge Integrated Circuits Ltd 2009-2022

2 Windows CTU Adapter Driver

This section describes how to install the Windows Adapter Driver required for the PC to communicate with the CTU through the CTU Adapter. There are actually two drivers: one used by the applications themselves and one that exposes a virtual COM port (VCP) for debug purposes. It is convenient to install them together, although the VCP driver is not required for normal operation.

2.1 Automatic delivery by Windows Update

The current CTU Adapter Driver is available from Windows Update. However it is not configured to install automatically. If a customer has an old version of the CTU Adapter Driver installed and it is working OK, there is no need to update it.

2.2 Manual delivery by Windows Update

The latest version of the CTU Adapter Driver may be obtained using the manual delivery from Windows Update. This section details installation steps. The PC must be connected to the internet.

Open Device Manager, for example by entering "Device Manager" in the Windows "Type here to search" box. Locate and expand "other devices". If a CTU Adapter Driver is not already installed then it will appear as in Figure 2.

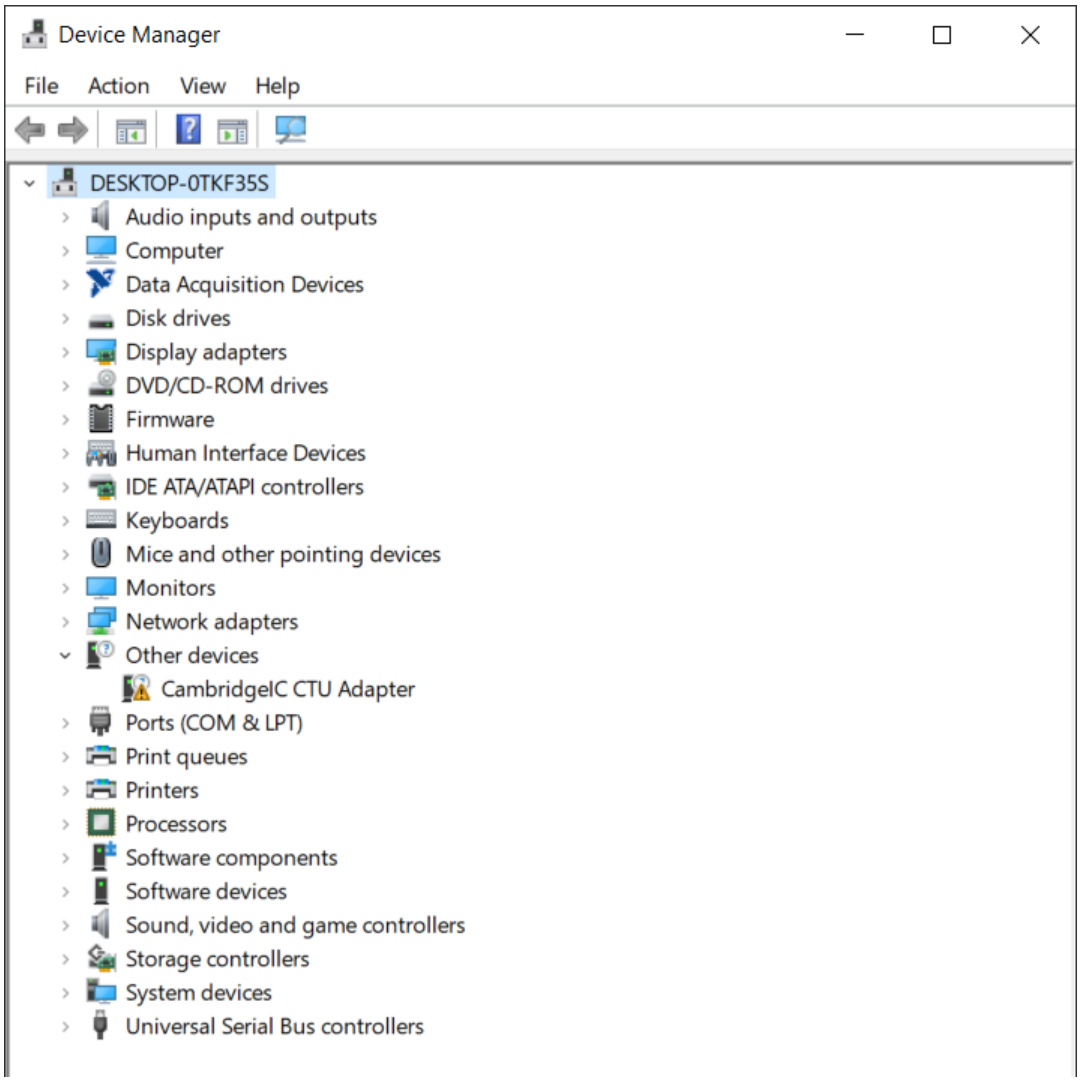


Figure 2 Device Manager showing CTU Adapter driver is not installed

Right click on "CambridgeIC CTU Adapter" and select "Update driver". The window shown in Figure 3 should appear.

© Cambridge Integrated Circuits Ltd 2009-2022

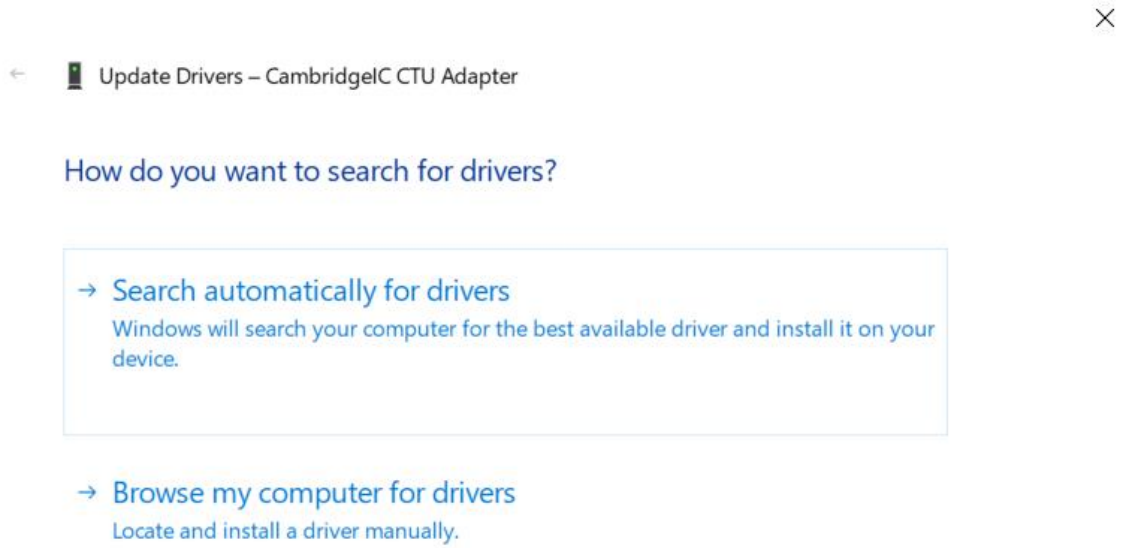


Figure 3 How to search

Select "Search automatically for drivers". Windows will not automatically install the driver and instead the window shown in Figure 4 will appear.

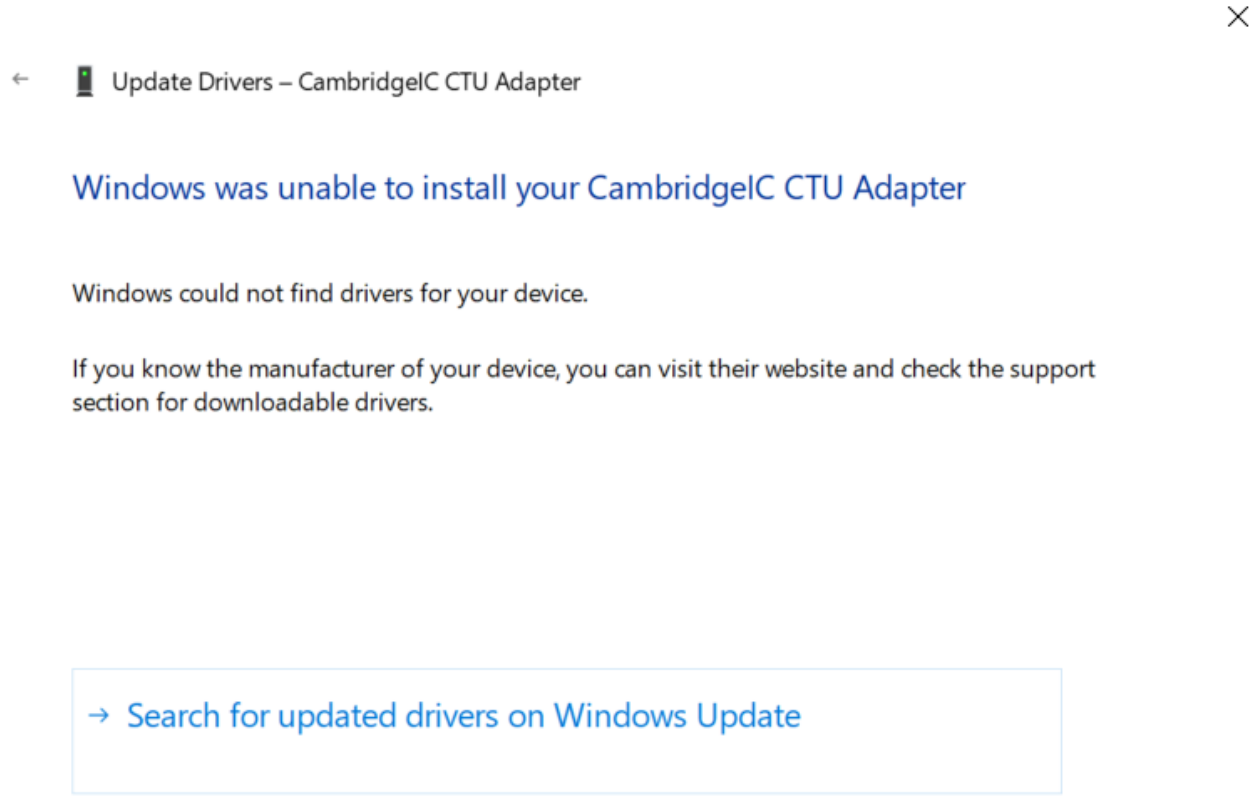


Figure 4 Unable to install

© Cambridge Integrated Circuits Ltd 2009-2022

Select "Search for updated drivers on Windows Update". The window shown in Figure 5 will appear.

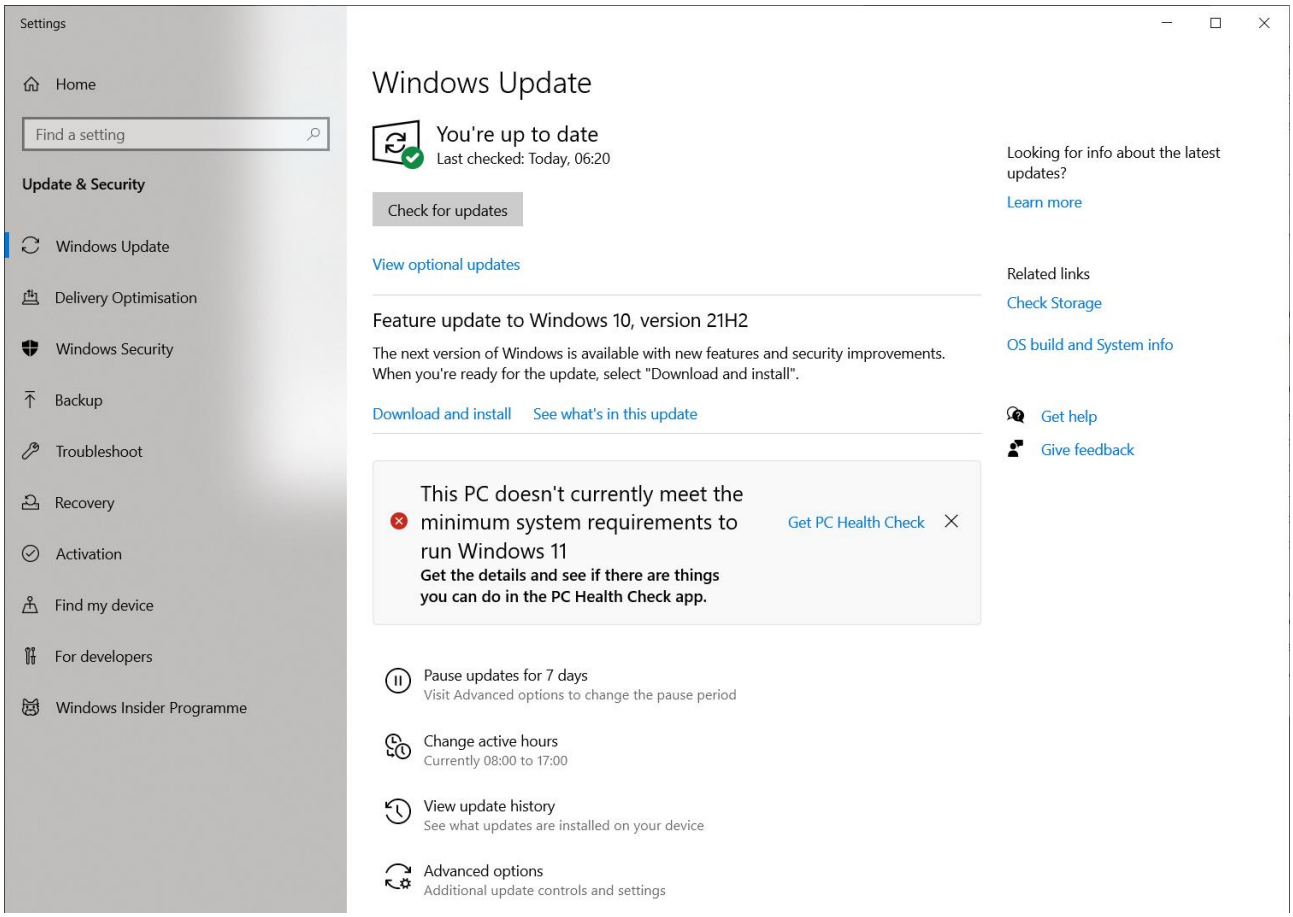


Figure 5 Windows Update screen

Click on "Check for updates" and wait for Windows to check for updates over the internet. Then click on "View optional updates" and expand "Driver updates". The list of available updates should now include "CambridgeIC – USB", and possibly also "CambridgeIC – Ports", as shown in Figure 6. Select one or both boxes and Click "Download and install".

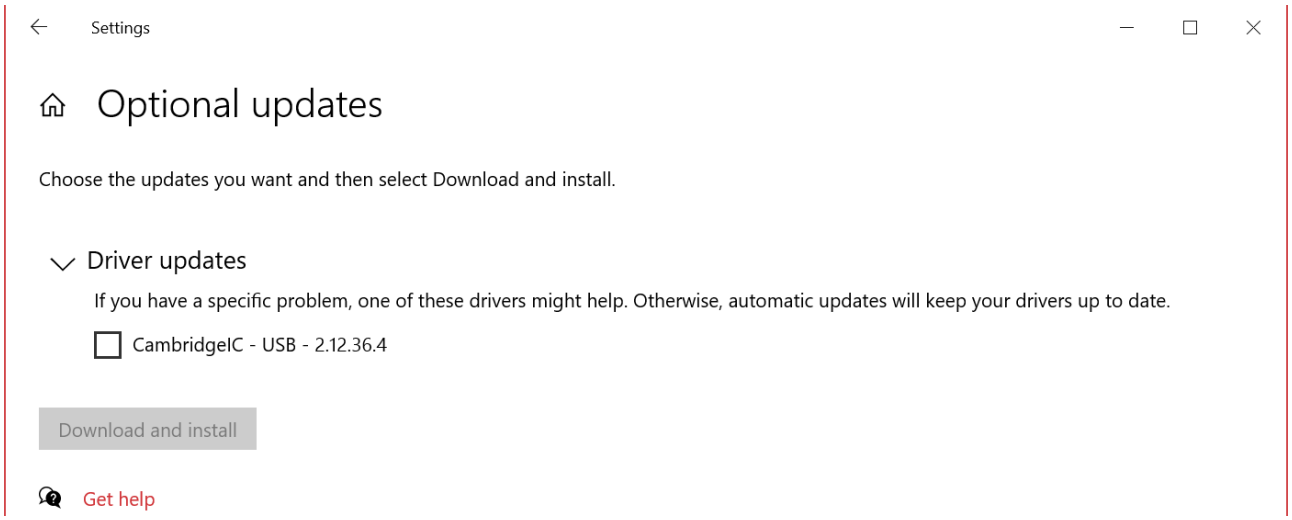


Figure 6 Optional updates window

The "Windows update" window should then display again, including "Status: Downloading" for both drivers. Note that the "0%" progress indication may not necessarily change.

© Cambridge Integrated Circuits Ltd 2009-2022

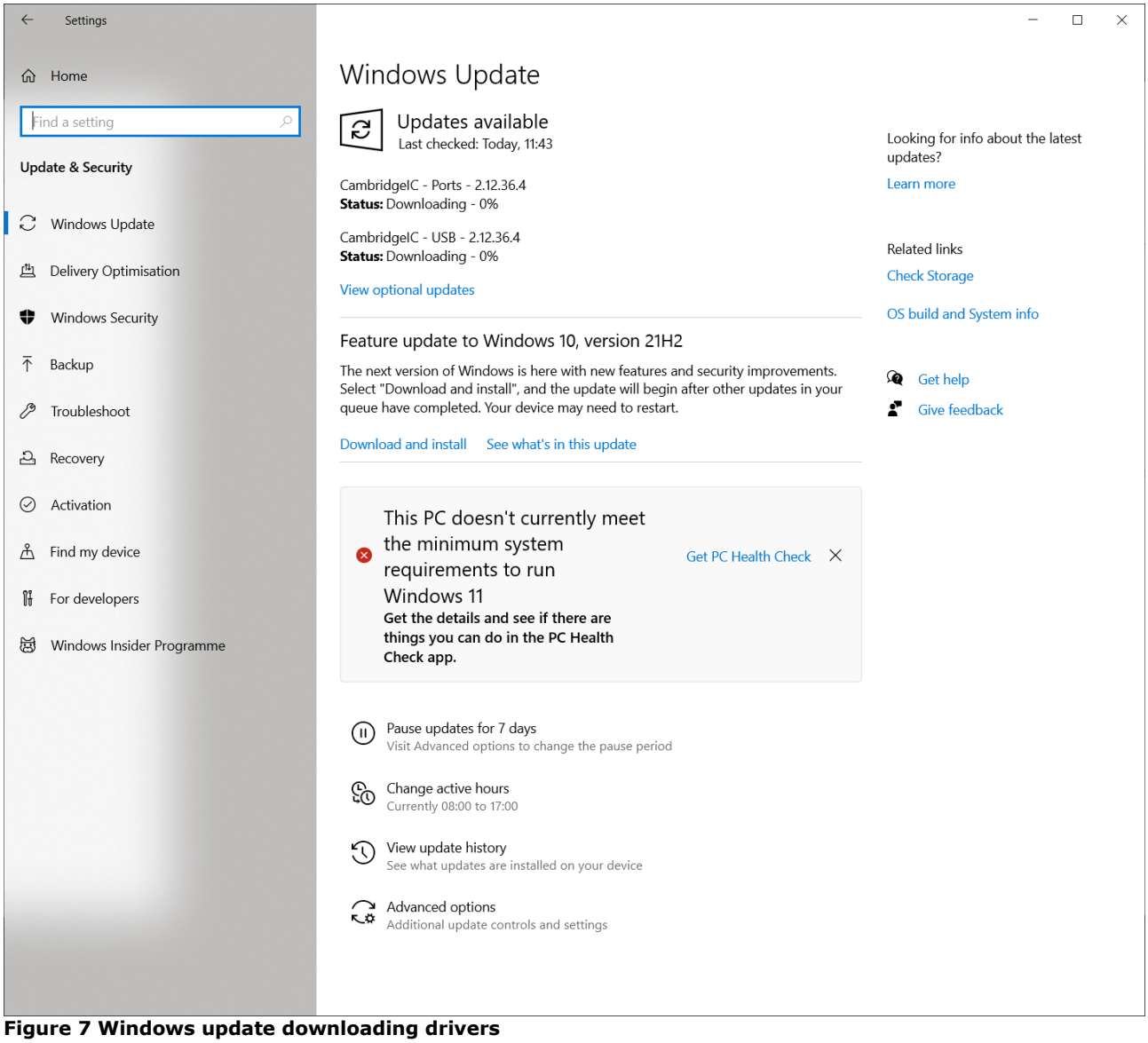


Figure 7 Windows update downloading drivers

After a few seconds the Device Manager window should refresh to include "CambridgeIC CTU Adapter" in two places: under "ports" and "Universal Serial Bus controllers", as shown in Figure 8.

© Cambridge Integrated Circuits Ltd 2009-2022

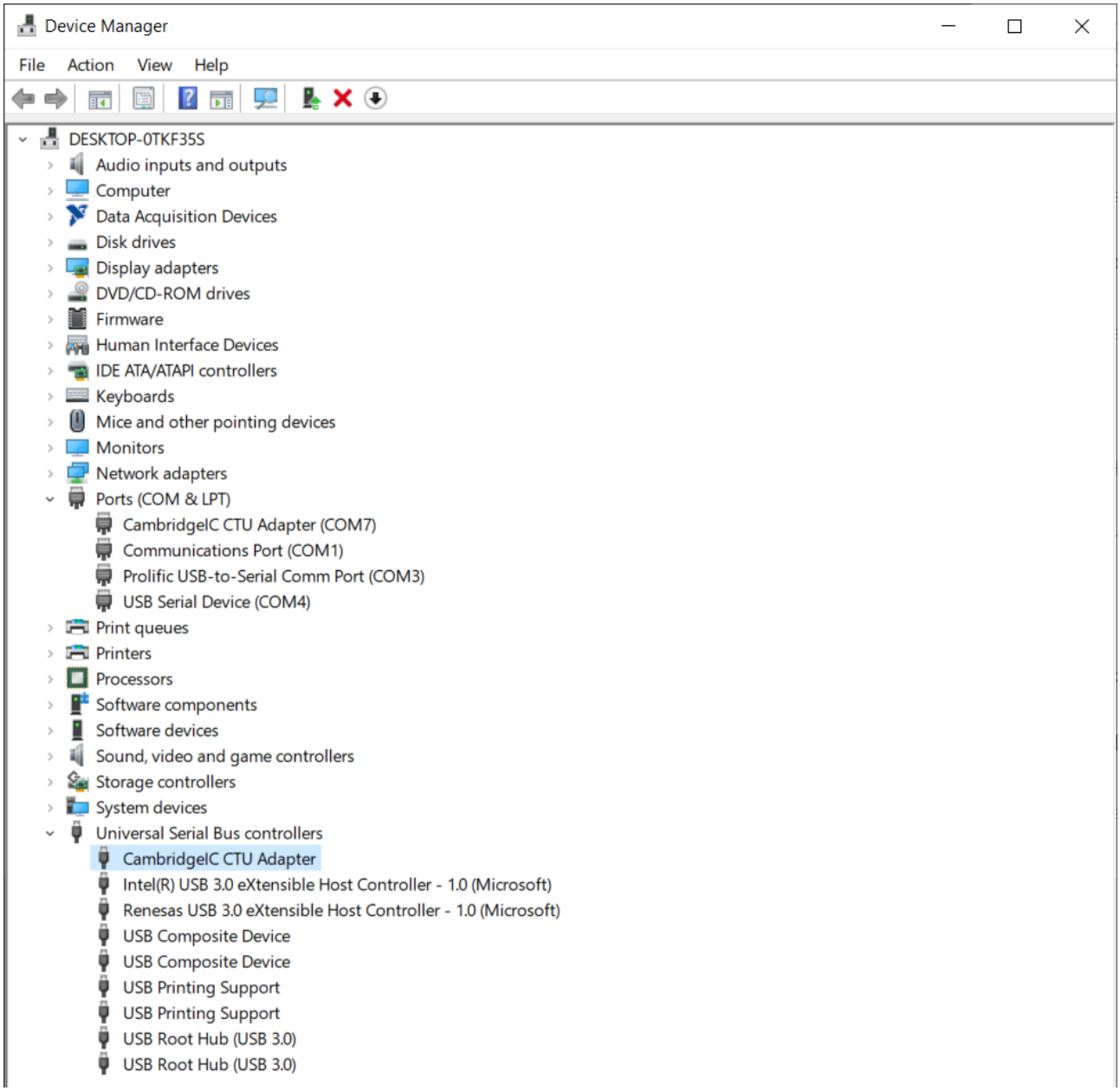


Figure 8 Device Manager, CambridgeIC CTU Adapter installed successfully

Once Device Manager shows "CambridgeIC CTU Adapter" in both locations as in Figure 8, driver installation is successful. CambridgeIC CTU Software may now be used.

© Cambridge Integrated Circuits Ltd 2009-2022

3 CambridgeIC CTU Software Installation

3.1 Installing

The files required for installation are available as a zip file from the download section of www.cambridgeic.com, or in the appropriately named directory on the CD supplied with a CTU Development Kit. Start by saving these files to an appropriate directory on the target PC.

The PC's security settings may prevent files from being copied out of the zip file until it has been unblocked. If this is encountered, right click on the zip file in a file browser and click on **unblock**.

It is recommended to shut all other programs before installation. If an older version of CambridgeIC CTU Software is already installed, this should be uninstalled with the procedure of section 3.2.

Locate and launch the setup.exe program. Follow the on-screen prompts to complete the installation.

Once completed, the applications require a restart of the PC for correct operation.

Microsoft Vista may display the warning illustrated in Figure 9 during installation, because the CTU software is not registered with Microsoft.



Figure 9 Vista PC warning

3.2 Uninstalling

Select the **Control Panel** from the **start** menu on the PC's desktop. Double click on **Add or Remove Programs**. Scroll down the list of installed applications and click on CambridgeIC CTU Software. Click the **Remove** button to remove the software.

3.3 Microsoft .NET Framework 3.5

CambridgeIC Software requires Microsoft .NET Framework 3.5 on a PC to run successfully. This is often already installed. If it is missing then CambridgeIC CTU Software applications may display a broken arrow illustrated in Figure 10 in place of the run button of Figure 11.

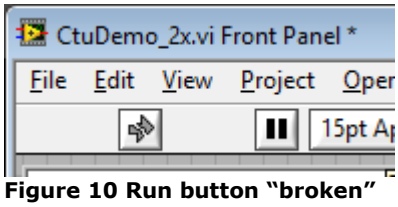


Figure 10 Run button "broken"

In in this case Microsoft .NET Framework 3.5 must be installed. Visit www.microsoft.com and search their web site for ".net 3.5 download".

© Cambridge Integrated Circuits Ltd 2009-2022

4 Operation of Applications

4.1 Starting and Stopping

The CambridgeIC CTU Software appears as shortcuts from the PC's **start** → **All Programs** menu when installed with default settings.

When launched, the applications will run automatically. They may be stopped by the **STOP** button and re-started by clicking on the **Run** arrow, as illustrated in Figure 11.

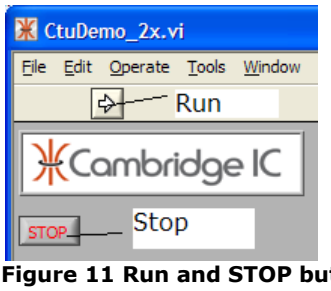


Figure 11 Run and STOP buttons

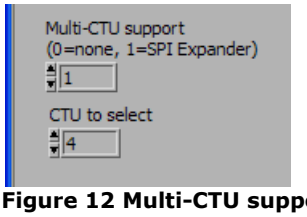


Figure 12 Multi-CTU support

4.2 Changing Settings

The CTU Applications include controls which modify the system's 9ehavior. Some of the changes take effect immediately when the system is running, for example changes to **PWRDN** in ***CTU Demo***.

Other controls used for initial system configuration only take effect when the application is run. To make changes, stop the application, make the change and run it again.

Some applications launch a second window, for example ***CTU Demo*** launches ***Display Type 1 (or 2) Results***. Depending on the PC, it may be necessary to stop the applications to make changes to controls in the original window. This will be apparent if clicking on the control has no effect.

4.3 Checking Software Version Number

To check the version number of applications, click on Help and select About. The About window includes the version number of the CambridgeIC CTU Software, as illustrated in Figure 13.

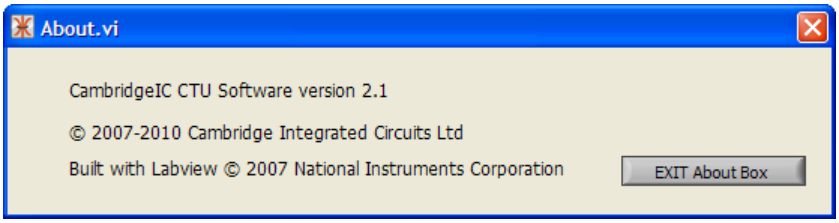


Figure 13 checking software version number

4.4 Selecting a CTU in a Multi-CTU System

Several CTU chips may be connected to the CTU Adapter's SPI bus, using additional circuitry to allow the CTU Adapter to switch between them. In this case, the applications which communicate with CTU chips have provision to select which one they communicate with. The relevant controls are illustrated in Figure 12, and may be located by scrolling down below the default window contents. The **Multi-CTU Support** control selects how multiple CTU chips are supported. When a single CTU chip is connected directly to the CTU Adapter, this should remain at the default value of 0. If an SPI Expander is used, with CTU selection using I2C, the control should be changed to 1. The **CTU to Select** control then selects which CTU chip to communicate with. Both controls only take effect once when first run, so the application must be stopped and re-started for changes to take effect.

© Cambridge Integrated Circuits Ltd 2009-2022

5 CTU Demo

This application reads and displays measurements taken by a CTU chip. It is controlled by the window illustrated in Figure 14. The info boxes contain CTU, Adapter and DLL version information.

Measurement results are displayed in a separate window. Figure 15 illustrates the window for a Type 1 sensor as shipped in the CTU Development Kit. Change the **Sensor Type** control if different sensor(s) are attached to the CTU chip. To make changes to settings, press **STOP** then press the **Run** button to start again.

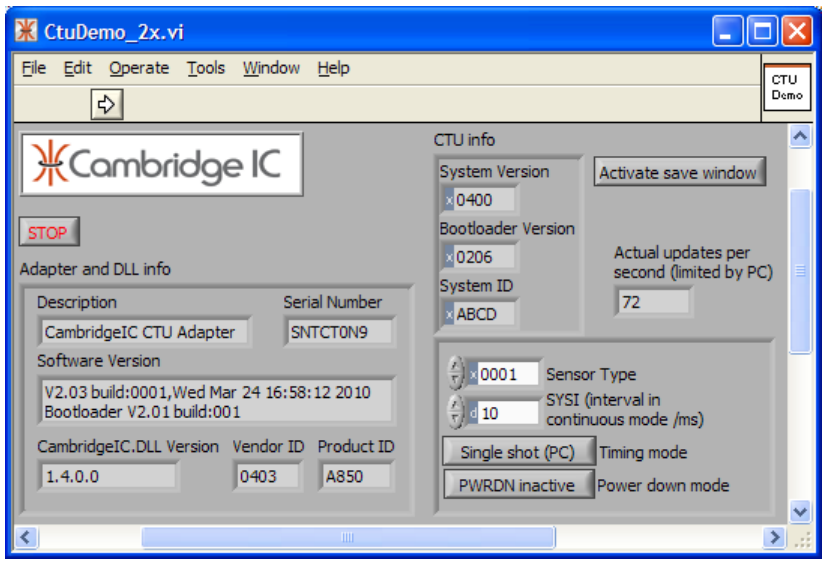


Figure 14 CTU Demo screen shot

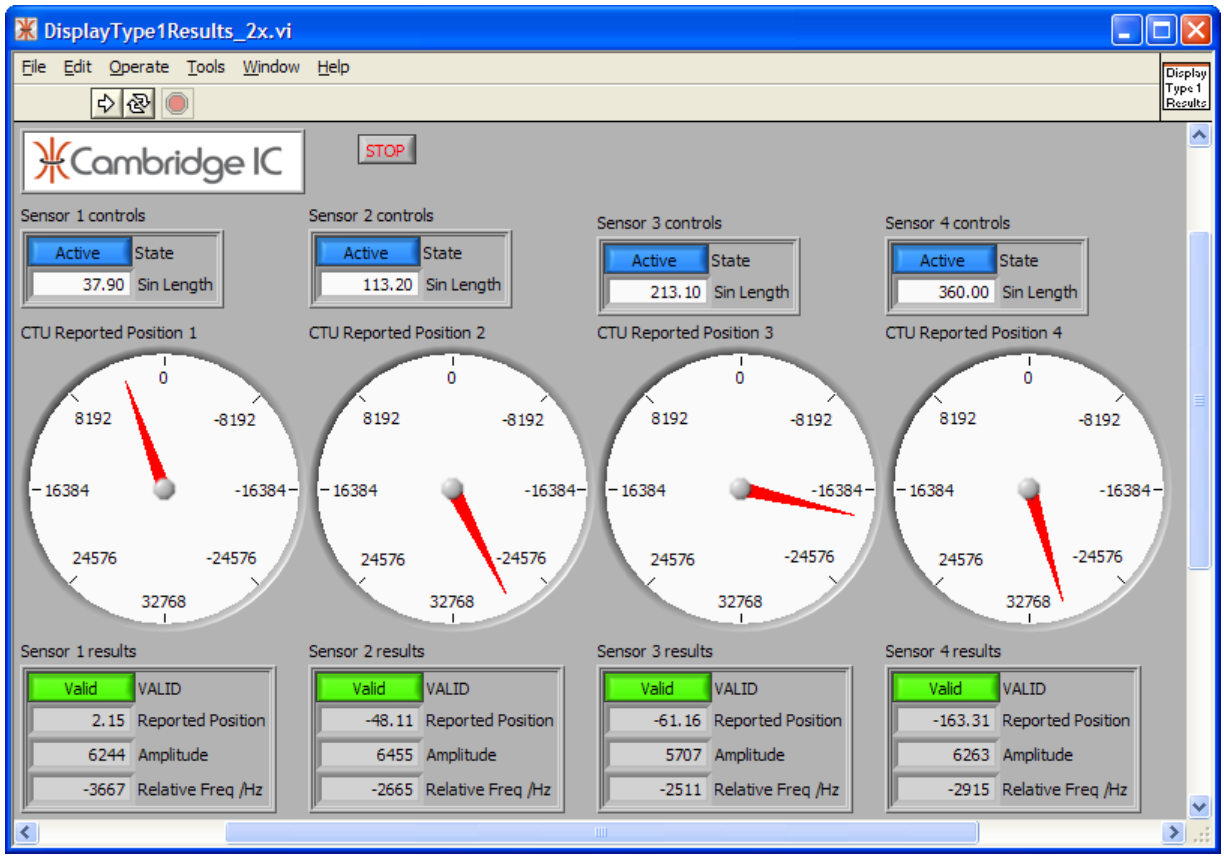


Figure 15 Display Type 1 Results screen shot

© Cambridge Integrated Circuits Ltd 2009-2022

The CTU Demo window displays the PC's estimate of the number of position updates per second for the set of active sensors. This figure is typically 250 updates per second when one sensor is active. When integrated into a typical host system without PC timing, much faster rates are achievable.

By default **Single Shot (PC)** is active: each measurement is initiated by the PC. This yields the greatest number of updates per second. However PC timing is erratic. Clicking on this control changes to **CONT (CTU)** and makes the CTU operate in its continuous mode with the interval between measurements more accurately controlled by the CTU's **SYSI** register. **SYSI** should be set to at least 10ms (1 sensor active) or 20ms (4 sensors active) so that the PC does not interrupt measurements in progress.

Click on the **Power down mode** button to put the CTU into its power down mode. The application STOP button only works when this control is inactive. Click on **Activate save window** to launch the ***Save to File*** window, which is described in section 7.

5.1 Type 1 Results Window

The CTU Development Kit is supplied with Type 1 sensors. The CAM204 CTU chip can process up to 4 Type 1 sensors. The display window illustrated in Figure 15 includes a set of displays and controls for each sensor.

Each sensor's **State** control may be independently set to **Active** or **Inactive**. Setting unused sensors to **Inactive** increases the maximum sample rate for the others.

The raw position output for each sensor is **CTU Reported Position**, and this is a signed 16-bit word for Type 1 sensors (-32768 to +32767). It is displayed on the circular dial. **Sin Length** controls how this value is scaled to Reported Position…

$$Reported\ Position = SinLength \times \frac{CTU\ Reported\ Position}{65536}$$

Equation 1

The application's default values of **Sin Length** are for the CTU Development Kit as shipped, resulting in Reported Position in mm for the three linear sensors and degrees for the rotary sensor. Change **Sin Length** for other sensors and operating conditions, and when sensors are plugged into different CTU inputs. Values are quoted in the relevant sensor's datasheet. It is also possible to view position measurements as a percentage of Sin Length by setting Sin Length = 100, or in CTU units by setting Sin Length = 65536.

The **VALID** indicator reflects the state of the sensor's **VALID** flag, indicating a target in range. **Amplitude** is a measure of signal quality and depends mainly on the gap between target and sensor. **Relative Frequency** is the CTU's approximate measurement of the target's resonant frequency in Hz, relative to the nominal centre frequency.

5.2 Type 2 Results Window

The CAM204 CTU chip can process one Type 2 sensor. Connect a Type 2 sensor to the CTU and run CTU Demo with **Sensor Type** set to 2. The window illustrated in Figure 16 displays measurement results.

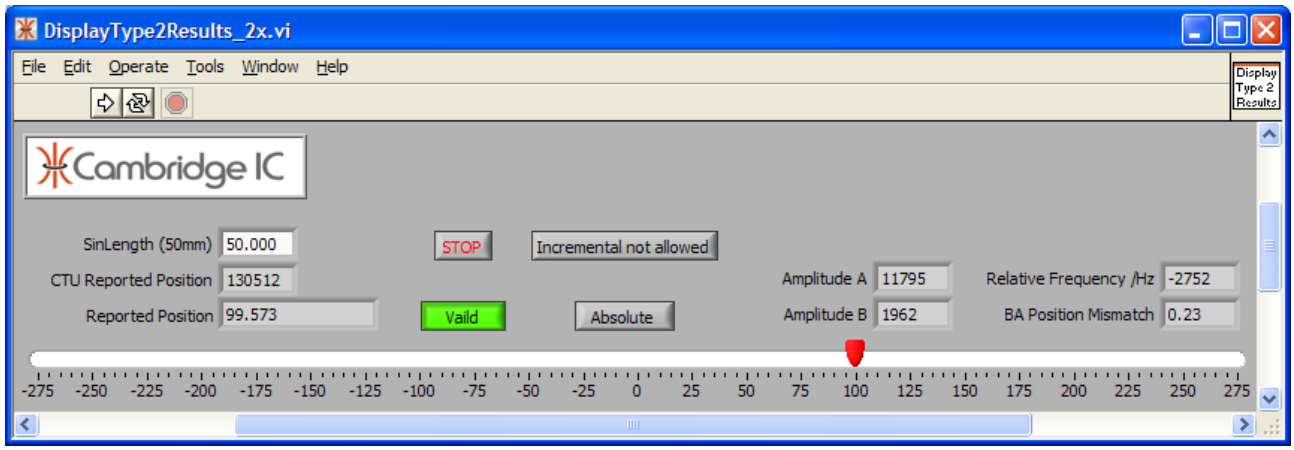


Figure 16 Display Type 2 Results screen shot

Sin Length scales from **CTU Reported Position** units to physical units (mm). It applies to the Type 2 sensor's "Fine Pitch A" parameter, usually 50mm. **Reported Position** is calculated from **CTU Reported Position** with Equation 1.

© Cambridge Integrated Circuits Ltd 2009-2022

5.3 Type 6 Results Window

The CAM204 chip can process either one or two Type 6 sensors. Connect the sensor(s) to the CTU and run ***CTU Demo*** with **Sensor Type** set to 6. The window illustrated in Figure 17 will launch and displays measurement results.

Type 6 sensors have a **Subtype**. For example a Type 6.3 sensor is Type 6, Subtype 3. The Subtype is the number of fine sensor coil periods per circle. The correct value for each sensor must be entered in the **Num Fine Periods (Subtype)** control. The **SinLength** parameter is the pitch angle of the fine sensor coils, and its value in degrees is calculated automatically for each sensor. Table 1 shows the expected values. Subtype may be changed while CTU Demo is running; new values are sent to the CTU chip automatically when required.

Each sensor's **State** control may be independently set to **Active** or **Inactive**. Setting unused sensors to **Inactive** increases the maximum sample rate for the others. The CAM502 CTU chip only supports one Type 6 sensor, and sensor 2 must be set to **Inactive** when operating with a CAM502 chip otherwise the system will fail to respond correctly.

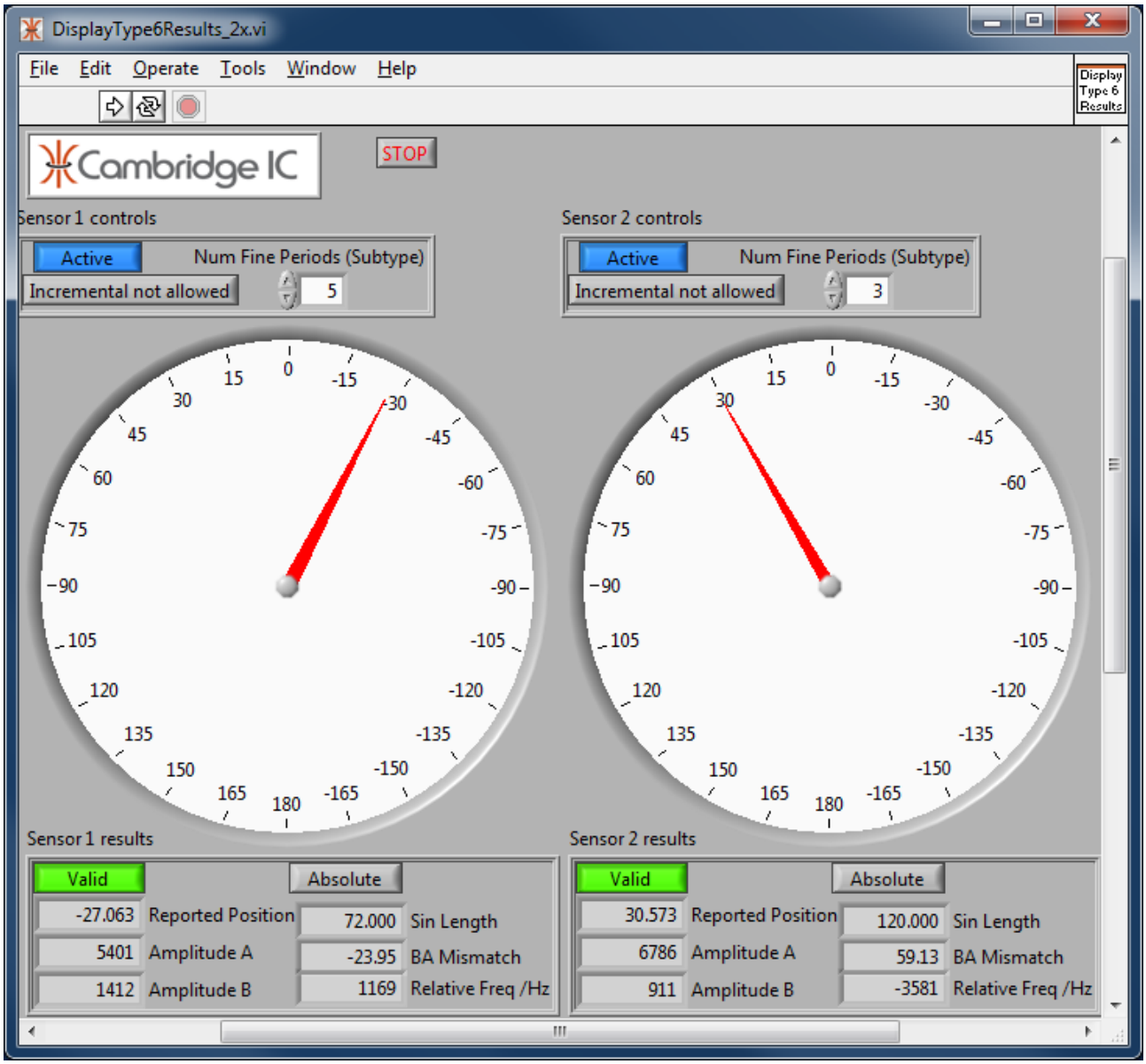


Figure 17 Display Type 6 Results screen shot

Each sensor may be operated in absolute mode (**Incremental not allowed**) or incremental mode (**Incremental allowed**). When operating in incremental mode, the CTU chip uses absolute measurements to determine angular position until Valid, then takes measurements only from the fine sensor coils to update absolute position each sample. This mode is only suitable when the angle change per measurement is less than half the SinLength angle, otherwise

© Cambridge Integrated Circuits Ltd 2009-2022

the system may skip a fine period resulting in an angle error that is only cleared on the next absolute measurement. PCs do not have well controlled timing and may leave long gaps between measurements, so incremental operation is not recommended for normal testing with a PC.

When running, the ***Display Type 6 Results*** window displays measurement results for each sensor.

The **Valid** indicator shows when the target is in range.

Reported Position is indicated in degrees with a digital display and dial indicator.

Two amplitude readings are displayed. **Amplitude A** is determined from the fine sensor coils, and **Amplitude B** from the coarse. References to Amplitude alone relate to Amplitude A.

In absolute mode **BA Mismatch** displays the difference in reported angle between fine and coarse sensors, and is an indication of system health. In incremental mode BA Mismatch displays the difference in angle between successive measurements. In each case, healthy values are less than 40% of SinLength.

Relative Freq /Hz displays the frequency difference between the CTU chip's nominal centre frequency (187.5kHz for the CAM204) and its measurement of the target's resonant frequency.

Table 1 SinLength values for Type 6 sensors

Subtype	SinLength
2	180°
3	120°
4	90°
5	72°
6	60°

5.4 Advanced Controls for the CAM502 CTU Chip

CTU Demo and other CambridgeIC CTU Software applications also work with the CAM502 CTU chip, since it has a very similar SPI interface and on-chip registers.

CTU Demo includes advanced controls for accessing additional features of the CAM502. These appear when the main window is expanded downwards as illustrated in Figure 18. To access the features described below, change **CTU (CAM204=0, CAM502=1)** to 1. The settings described below should be made before running CTU Demo. Some controls such as FCSEL are updated while the software is running, but it is advisable to stop and restart to ensure other changes are transferred to the CAM502.

The CAM502 can measure positon at a sample rate of up to 5.1kHz (195µs measurement interval). CTU Demo can not collect samples at this rate due to the limited speed of a PC and the USB interface. Instead, it is possible to configure the CAM502 for continuous operation triggered by an on-chip timer. That way, it does not rely on an SPI transaction initiated by CTU Demo to trigger each sample.

For the highest possible continuous sample rate, the CAM502 includes pipeline mode (**PIE** set to **Active**). **SYSI** controls the interval between measurements, and is in multiples of 0.1µs when PIE is active. In this case if SYSI is set to 2000 the measurement interval will be 200µs and the sample rate 5kHz. Otherwise SYSI is in 100µs multiples.

Since CTU Demo is unable to display such high sample rates, the benefits of a high sample rate can instead be observed by configuring the CAM502 to perform on-chip filtering, using the **FCSEL** control. This is an array, with sensor 1's FCSEL listed first, then sensor 2's and so on. Since the CAM502 only supports sensor 1, only the upper figure should be used. When FCSEL is set to 0, no filtering is applied, and each position displayed by CTU Demo is a new, independent measurement. As the value of FCSEL is increased up to its maximum of 15, progressively more filtering is applied. This delivers approximately 0.23 bits of increase in noise free resolution for each bit of FCSEL. At the same time the Group Delay between actual and reported position increases. For example at FCSEL=10, the delay is ~6ms for a sample rate of 5kHz.

When operating in pipeline mode (PIE active) and with **Incremental Not Allowed** set in the Display Type 2 or 6 results window, the CAM502 performs occasional coarse measurements from the sensor's coarse coils as a regular check of absolute position. Its **ABSSEL** control determines how often, with 2^ABSSEL fine samples between each coarse. ABSSEL is displayed as an array, but only the top value (sensor 1) is used for the CAM502. A value of 8 is recommended (256 fine measurements between each coarse).

© Cambridge Integrated Circuits Ltd 2009-2022

If CTU Demo is to be used with a Type 2 sensor, set the top entry of the **Sensor Subtypes** Control to 0.

The CAM502 supports connection to either a Type 2 or a Type 6 sensor, and **Sensor Type** must be set accordingly. When run, CTU Demo will launch the appropriate display window.

For a Type 6 sensor, Display Type 6 Results launces (Figure 17). Sensor 2 must be set to inactive for correct operation with a CAM502, and sensor 1's **Num Fine Periods (Subtype)** must be changed to match the sensor's subtype.

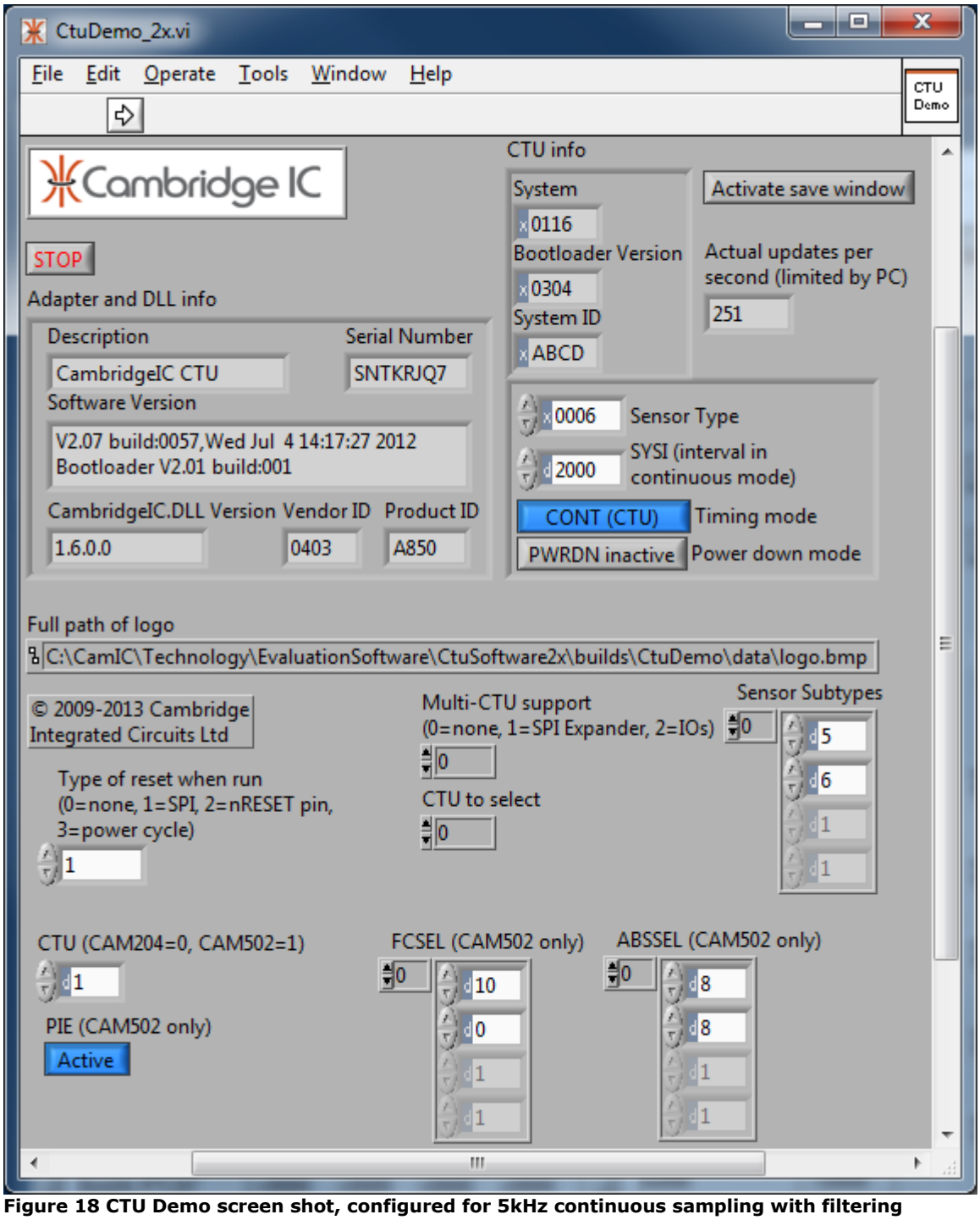


Figure 18 CTU Demo screen shot, configured for 5kHz continuous sampling with filtering

© Cambridge Integrated Circuits Ltd 2009-2022

6 CTU Repeat Average

In this application, the CTU take a number of measurements on a selected sensor, and displays average readings. Averaging improves resolution and reduces noise in the averaged output. This application demonstrates the improvement in resolution. It may be used to take higher resolution measurements than CTU Demo. The **Requested Num VALID** control selects how many CTU measurements to base the average on. A larger number results in improved resolution and lower noise in the **Position Average** output at the expense of update rate.

In addition to average values, the application also displays statistics based on the set of samples acquired for each averaged measurement. For example **Position Std Dev** is the standard deviation of the incoming reported positions, and **Position Pk Dev From Average** is the peak deviation from the average value. These are useful measures of the noise present in the CTU's output, and hence of system performance. For meaningful noise statistics, it is recommended to set **Requested Num Valid** to 64 or more. Note that peak and standard deviation measurements are based on incoming data and are not measures of noise in the averaged output. Neither *improves* as Requested num VALID increases, although they do become less variable themselves.

CTU Repeat Average is designed to work with different sensor types. Set the **Sensor Type** control before running so that the CTU chip and PC software performs the correct operations. Some sensor Types including Type 6 have different Subtypes, and in this case the **Subtype** control must also be set to the correct value for the sensor. Otherwise it must be left at its default value of 0.

As with CTU Demo, **Sin Length** scales from the CTU's reported position data to physical units such as mm or degrees used for reporting position measurements. SinLength must be entered, it is not calculated automatically. Values for any sensor are provided in its datasheet. The SinLength value of Type 6 sensors is determined by their Subtype, see Table 1.

Select **Rotary** from the **Linear/Rotary** button when operating a rotary sensor to ensure the apparent discontinuity at ±180° is dealt with properly. For a rotary sensor, the average of +175° and -177° is +179°, not -1°!

The CAM204 CTU chip supports up to 4 Type 1 sensors, or up to 2 Type 4 or 6 sensors. Select the one to measure with the **Sensor number** control. **Sensor number** must be set to 1 for a Type 2 sensor.

For operation with a CAM502 CTU Chip, the CTU Repeat Average window may be extended downwards to reveal advanced controls, which operate in the same way as with CTU Demo (section 5.4).

© Cambridge Integrated Circuits Ltd 2009-2022

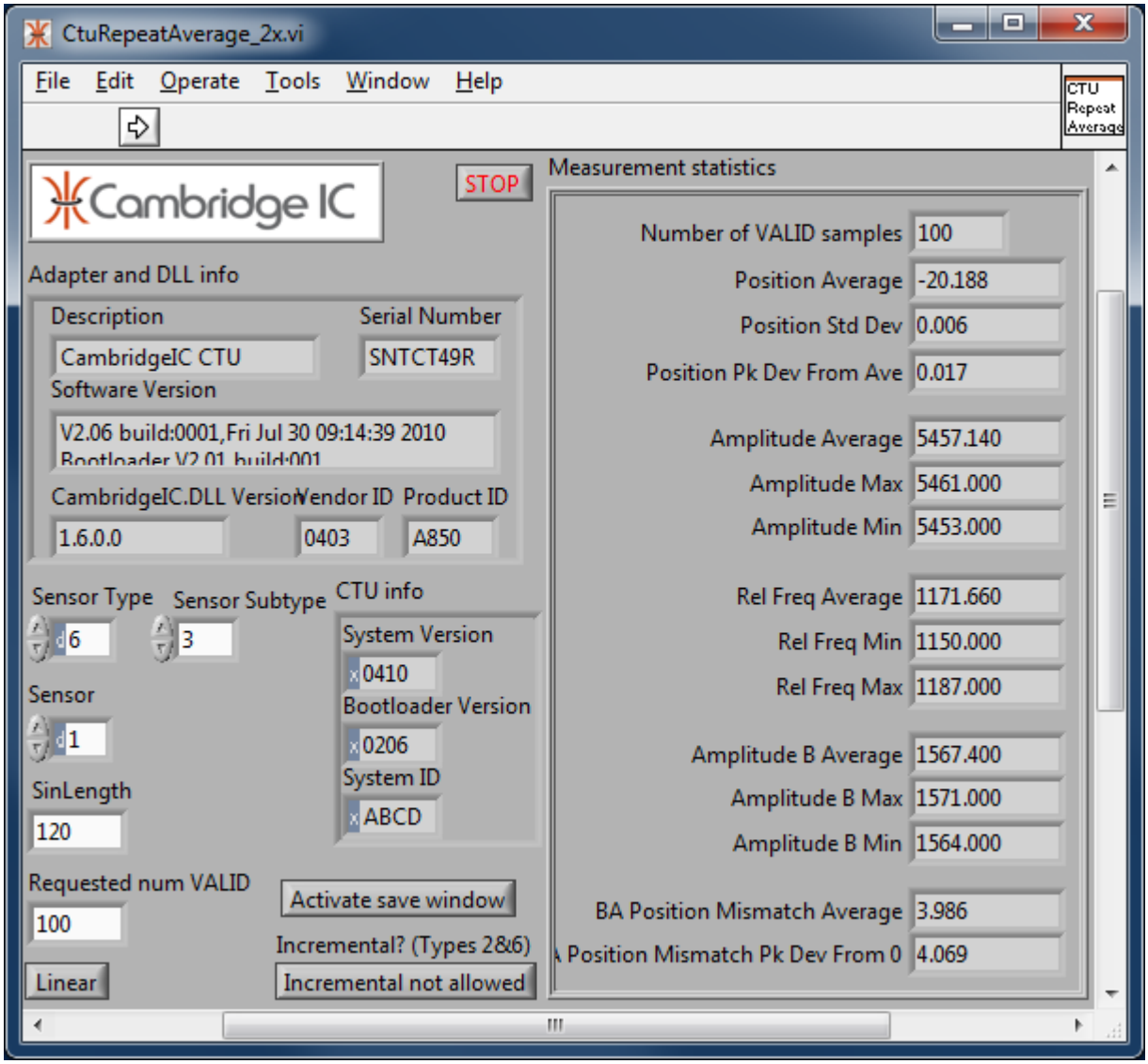


Figure 19 CTU Repeat Average screen shot

Click on **Activate save window** to launch the ***Save to File*** window, which is described in section 7.

© Cambridge Integrated Circuits Ltd 2009-2022

7 Save to File

This window is activated from either ***CTU Demo*** or ***CTU Repeat Average***, and is illustrated in Figure 20. It allows CTU measurements to be saved to a file as tab delimited text. The file that is generated may be opened as a text file by Microsoft Excel, Mathcad or other applications for processing and presenting the data.

When launched, the application prompts the user for a results file name and location. This may be a new or existing file. The file can be changed by pressing the **New results file** button. It is recommended to use the file suffix .txt for saved data. Every time a new file is selected the application writes a header row to the file which labels each column of data. The **Write header row** button may also be activated to write a header row next time data is saved.

The **Text to save** table is updated by the source application. Header information is on the left, and data on the right. There are two saving modes. When active, **Save continuously** streams data to file as it arrives. **Save selected number** saves the number selected in the **Number to save** control. The default is 1 sample, so that each time the **Save selected number** button is pressed a single data row is added to the file.

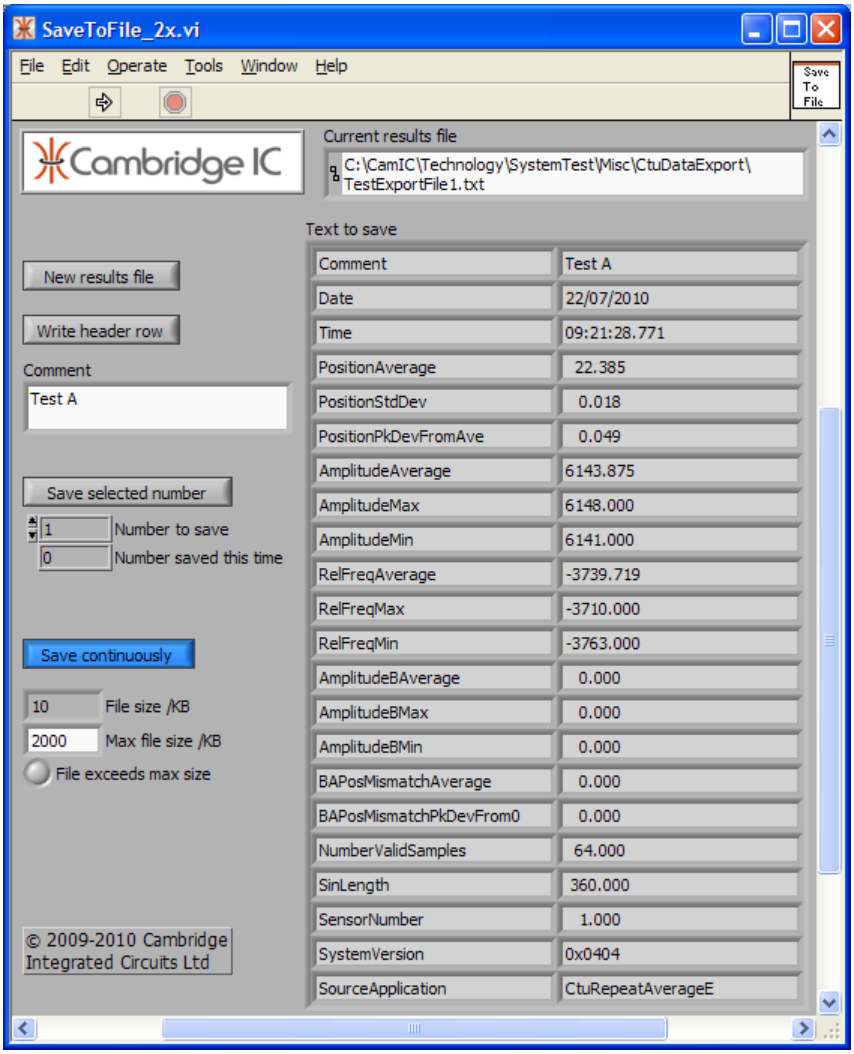


Figure 20 Save to File window screen shot

When used with ***CTU Demo***, the **Text to save** table displays the results of the most recent measurements on each sensor. In this case the update rate is controlled by ***CTU Demo***. If **Single shot** mode is active the update rate is the maximum the PC can achieve for the number of sensors active. If **CONT** is active, the CTU chip times the approximate interval between samples on all active sensors, and the interval between samples controlled with **SYSI**. ***CTU Demo*** saves data from all active sensors.

When used with ***CTU Repeat Average***, individual measurements are taken from a single selected sensor at the maximum rate. They are processed in batches to determine Position Average and other statistics which are passed to ***Save to File***. Change ***CTU Repeat Average***'s **Requested num VALID control** to change the update rate.

© Cambridge Integrated Circuits Ltd 2009-2022

Data is always appended to the selected file, and can not be removed within the ***Save to File*** application. If data needs to be removed, this can be done using a text editor or in Microsoft Excel.

CTU Demo can generate up to 40kB data per second at maximum rate, so file sizes can quickly become excessive. The **Max file size** control sets the maximum allowed size in kB, and can be changed by the user to suit their needs. If **File size** exceeds **Max file size** then saving is disabled to protect against accidentally generating large files.

In addition to the data originating from the CTU, ***Save to File*** includes a comment, the date and the time at the beginning of each data row, and the CTU system version and source application that generated the data at the end of the row. The comment data comes from the **Comment** box and enables the user to add data to each row, for example temperature, actual position(s) or other configuration data. The time is in 24 hour clock format and includes seconds and thousandths of a second: hours:minutes:seconds.xxx. The best resolution available from the PC's internal clock is typically 16ms.

When choosing a file, the dialog box gives the option to **cancel**. If this is pressed the next window will give the option to **continue** or **stop**. Always press **continue**, otherwise the application will not close communication with the Adapter correctly and an error will occur next time a CTU Application is run.

7.1 Importing Data into Microsoft Excel

Microsoft Excel is useful for processing data and displaying the results in graphs.

The following description is for Microsoft Excel 2007. The procedure for earlier versions is similar.

There are two methods of importing saved data into Microsoft Excel:

The first method is to select **Open** from within Excel then change **Files of type** to **.txt** (or the file extension used when saving the data), then click **Open**. Change the file to Excel format when saving to preserve any non-text additions made, or copy and paste the data in to an Excel workbook. It is also possible to right click on the text file from a file browser and select **Open With** then **Microsoft Office Excel**.

The second method imports data directly into an open Excel worksheet. Select **Get External Data** from the Data tab and click on **From Text**. Change the **Files of type** control if necessary. Select the file and click **Import**. Click **Delimited** then click **Next** and check the **tab** box in the **Delimiters** box. Click **Next** then **Finish**, then select the top right cell where the data is required.

Excel may fail to include thousandths of seconds when interpreting the time. Thousandths of seconds are usually important for the time difference between two cells. This can be calculated in Excel using…

$$Time\ difference\ in\ seconds\ between\ cell\ B\ and\ cell\ A\ = 24\ \times 60\ \times 60\ \times (cell\ B - cell\ A)$$

The result should be formatted as a number in Excel, not a time. Click on the equation result and select **number** from the **home** tab and change to **number** format with the required number of decimal places.

7.2 Importing Data into Mathcad

Mathcad can also be used to read data generated by ***Save to File***. Its READPRN function only imports numbers. Recent versions of Mathcad (e.g. version 14) can accept files that include a mixture of numbers and text. The result is typically a data array that excludes header information, the Source Application column and the comments column if it includes non-numeric text.

Mathcad may import the first digits of date and time information, but not the complete field. If the time field is required, it is recommended that the file is opened using Excel and the time field is converted to a number using Excel as described above, before re-saving the file as text.

© Cambridge Integrated Circuits Ltd 2009-2022

8 CTU Control Panel

CTU Control Panel is for the manipulation of CTU register settings. These can be saved to file, retrieved from file, viewed and manipulated on-screen.

CTU Control Panel is particularly useful for configuring CTU chips for autonomous operation. It includes controls for transferring settings to the CTU's non-volatile memory, so that they become the new *Configurable Defaults*. These allow the CTU to operate autonomously without a host device connected to the SPI interface. Typical applications include driving an external DAC to generate an analog output, and generating position triggers.

Figure 21 shows a screen shot of the left hand side of the application's window. Sensor Register and System Control Register settings are listed to the left and bottom respectively. The state of each individual register depends on the settings of controls located in the tabs on the right hand side of the application. The rows highlighted with a blue indicator show the registers that are controlled by the tab currently activated. The right hand side of the application's window is devoted to a set of tabs controlling individual CTU functions, and these are described in the following subsections.

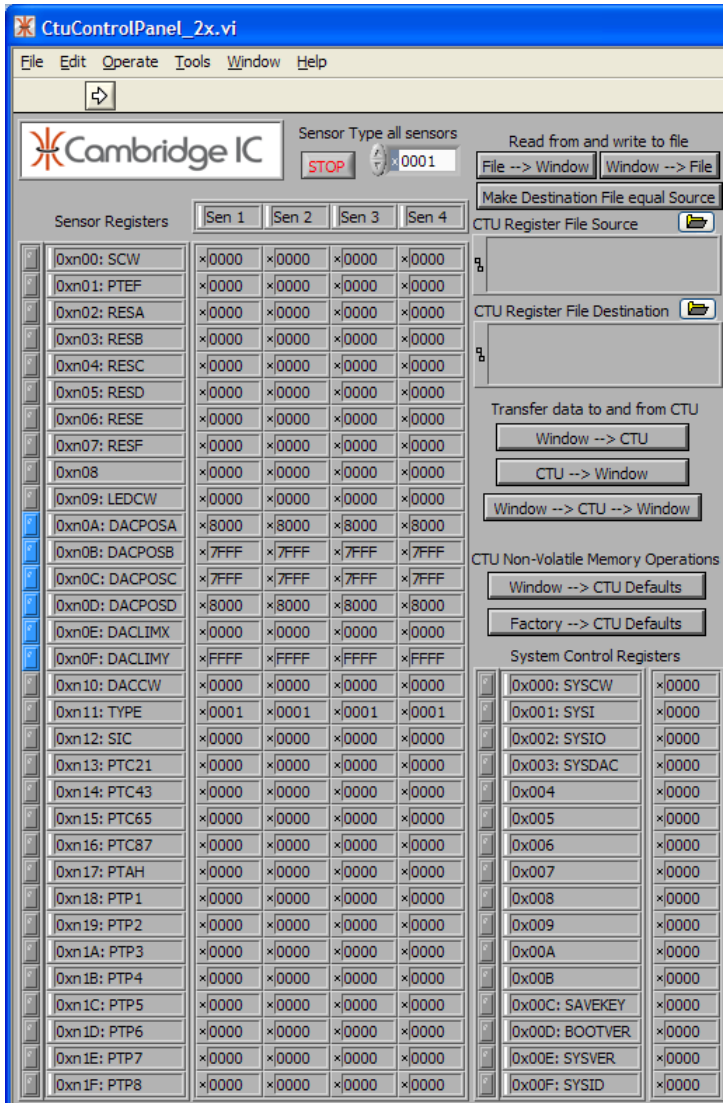


Figure 21 CTU Control Panel screen shot, left hand side

The controls to the top right of Figure 21 are for transferring settings between files, the CTU and the application's window. To change the files used for retrieving and saving settings, click on the **browse** button to the top left of the **CTU Register File Source** and **CTU Register File Destination** controls respectively.

File→ Window updates the application's window with data from a CTU Register File (.crf). **Window → File** saves the data in the window to file. **Window → CTU** transfers all of the register settings shown in the window to the CTU.

© Cambridge Integrated Circuits Ltd 2009-2022

CTU → Window reads the CTU's register contents and updates the window accordingly. **Window → CTU → Window** transfers data from the window to the CTU, and then reads back the CTU register contents and places them in the window.

Window → CTU Defaults takes the settings from the window and writes them to the CTU. It then makes the CTU perform a save operation which results in these settings becoming the new *Configurable Defaults*. The CTU will then reset itself. After this reset, and each subsequent one, the CTU will load these Configurable Defaults into its registers. To configure CTU chips for autonomous operation from file, first use **File → Window** to transfer the file contents to the window, then press **Window → CTU Defaults** to save them to the CTU. Note that CTU chips allow a large number of such updates to internal non-volatile memory, but not limitless. Please see the CTU datasheet for more information

Pressing the **Factory → CTU** button causes the CTU to revert to factory default register settings on each subsequent reset. As with the **Window → CTU Defaults** setting, this operation may be performed on CTU chips a large but not limitless number of times.

8.1 CTU Control Panel: Info Tab

The **Info** tab includes version information for the CTU Adapter and CTU chip. It also displays the text header of any CTU Register File that has been loaded.

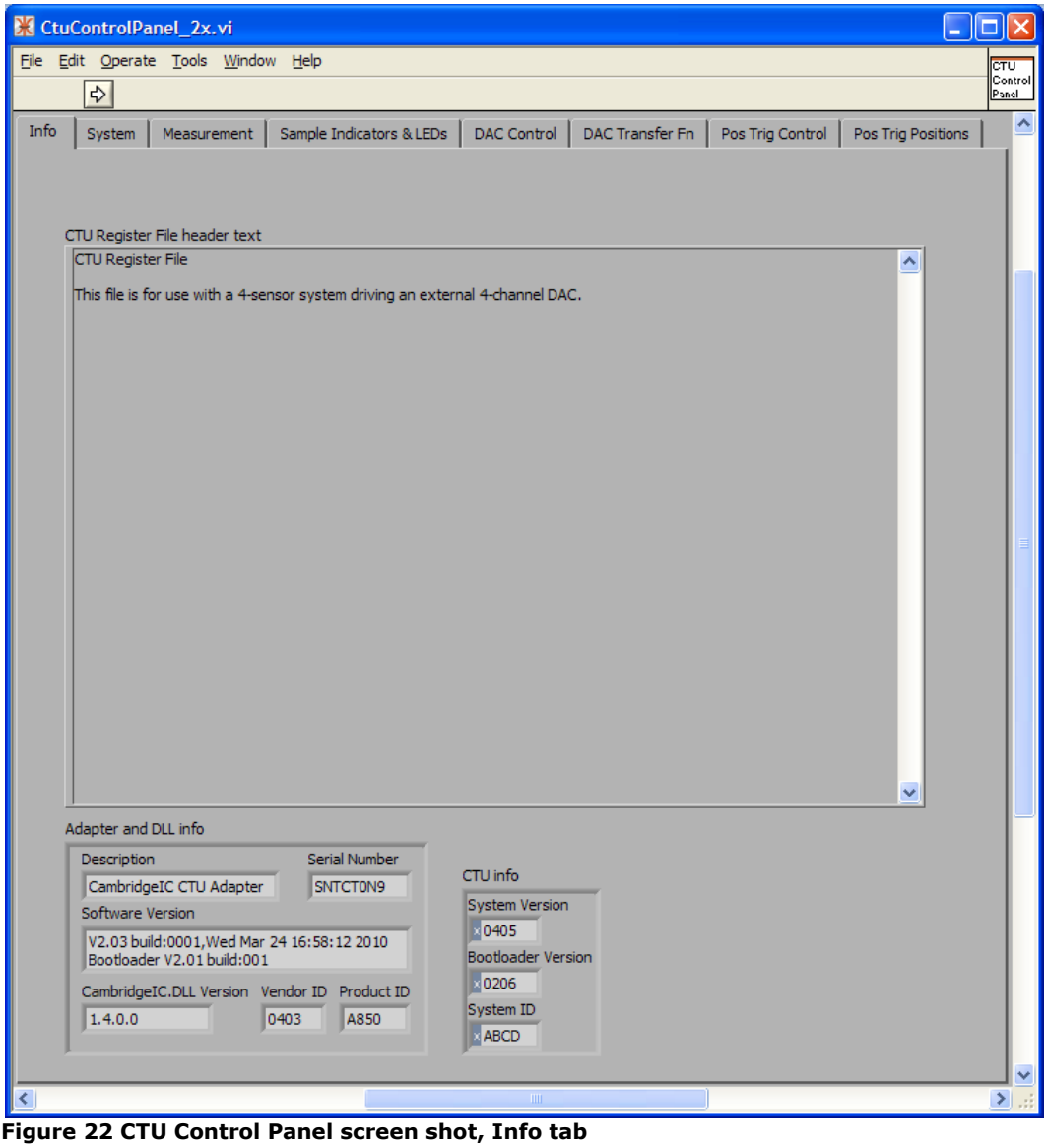


Figure 22 CTU Control Panel screen shot, Info tab

© Cambridge Integrated Circuits Ltd 2009-2022

8.2 CTU Control Panel: System Tab

The **System** tab includes system level CTU settings, as well as provision for resetting the CTU. Its contents are shown in Figure 23 below.

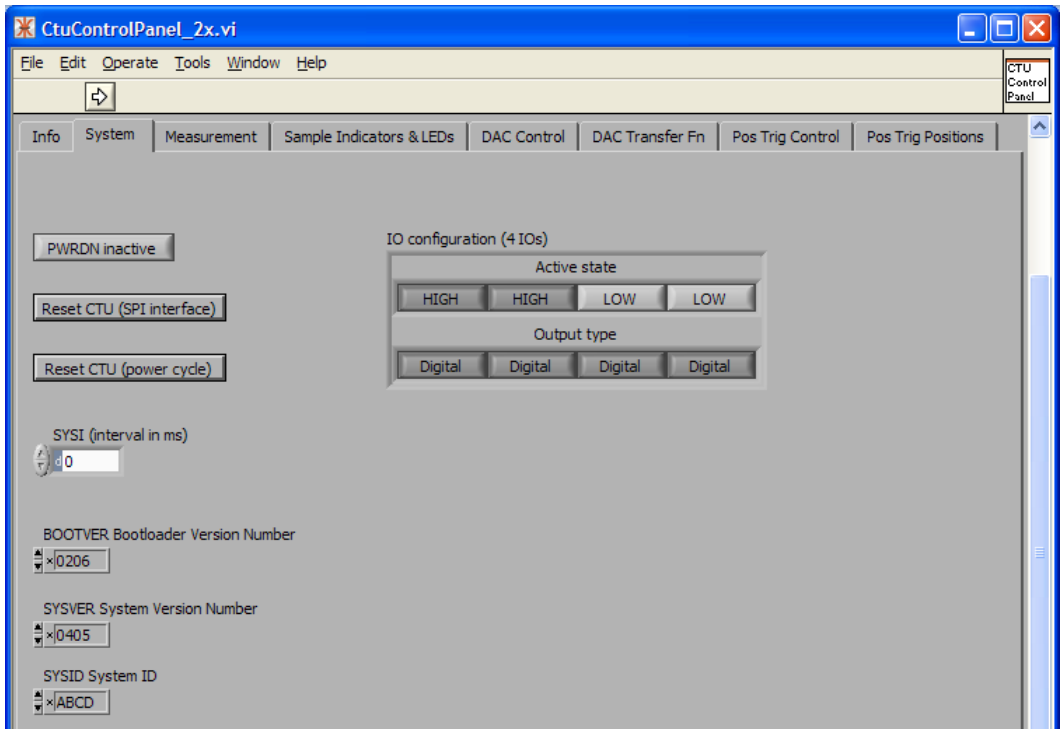


Figure 23 CTU Control Panel screen shot, System tab

The **PWRDN** button controls the state of the CTU's power down bit, which puts the CTU into a low power mode. This control must be returned to inactive for access to other functions.

The **Reset CTU (SPI Interface)** button sends a reset command to the CTU over its SPI interface. **Reset CTU (power cycle)** resets the CTU by turning its power supply off then on again. This control will only work if the CTU's power supply is controlled by the CTU Adapter. If not, **Reset CTU (SPI Interface)** button should be used instead.

The **SYSI** control is the interval between measurements when the CTU is set to continuous mode. The interval is measured by the CTU. The value is approximate, and is in milliseconds.

BOOTVER and **SYSVER** report the version numbers of the CTU's bootloader and application firmware respectively. **SYSID** is a user-configurable System ID. It is the first word read out of the CTU over the SPI interface each transaction, and may be used to verify the identity of individual CTU chips when there are multiple devices on the same SPI bus.

The **IO Configuration** control cluster determines the active state and type of output for each of the CTU's User Ios. The left hand controls correspond to IO1.

© Cambridge Integrated Circuits Ltd 2009-2022

8.3 CTU Control Panel: Measurement Tab

The **Measurement** tab is for controlling CTU measurements and checking measurement results. Its contents are shown in Figure 24.



Figure 24 CTU Control Panel: Measurement tab

The top two clusters of controls relate to the CTU's Sensor Control Word registers (SCW). **SCW Read from CTU** is updated when registers are read from the CTU by the **CTU → Window** and **Window → CTU → Window** buttons. Each row of buttons corresponds to one bit within the SCW register. Each column corresponds to one sensor, with sensor 1 on the left. To the right is the **SCW Controlling CTU** control. This includes the same set of bits for each sensor. This control may be updated by the user, or by the **File → Window** button. Its contents are written to the CTU when the **CTU → Window**, **Window → CTU → Window** or **Window → CTU Defaults** buttons are pressed.

Within the SCW controls, each sensor's **GO** bit is set to either **Active** to kick off a measurement, or to **Inactive** to not kick off a measurement. The **GO** and **CONT** bits should both be set to **Active** for continuous measurement, or just the **GO** bit for a single-shot measurement. Each sensor's **TRIG** bits enable Position Triggers for that sensor. The CTU sets a sensor's **NEW** bit each time it takes a measurement. The **NEW** bit controls on the right hand side should be set to **Clear** to ensure the indicators on the left are accurate.

Sample Indicators are digital outputs from the CTU that indicate the completion of a measurement, and/or the status of the last measurement. Each sensor has an enable bit **SIE** for sample indicators, and a flag indicating whether or not they have been activated, **SIF**. Sample Indicators are configured with the **Sample Indicators and LEDs** tab, see section 8.4.

© Cambridge Integrated Circuits Ltd 2009-2022

For sensor Types that support the feature, such as Type 2 and 6, the **INCE** bit selects incremental measurement mode. The **Sensor Subtype** control is an array for defining the subtype of each sensor connected, sensor 1 uppermost.

The **VALID** indicators show the state of each sensor's **VALID** bit, which indicates whether the last measurement on that sensor was valid or not. To be valid, an appropriate target must be in range of the sensor, and the CTU must have performed a sufficient number of measurements to lock onto the target's resonant frequency.

SinLength scales the raw position output from the CTU (**CTU Reported Position**) into physical units such as degrees or millimeters (**Reported Position**), see Equation 1 in section 5.1. The SinLength control is presented as a two-dimensional array. Values only need be entered for the Sensor Type (row number) required, and in the column number(s) corresponding to the sensor(s) connected.

The **Results** indicators at the bottom of the Measurements tab show measurement results, based on readings taken from the CTU's results registers.

8.4 CTU Control Panel: Sample Indicators and LEDs

The Sample Indicators and LEDs tab is shown in Figure 25.

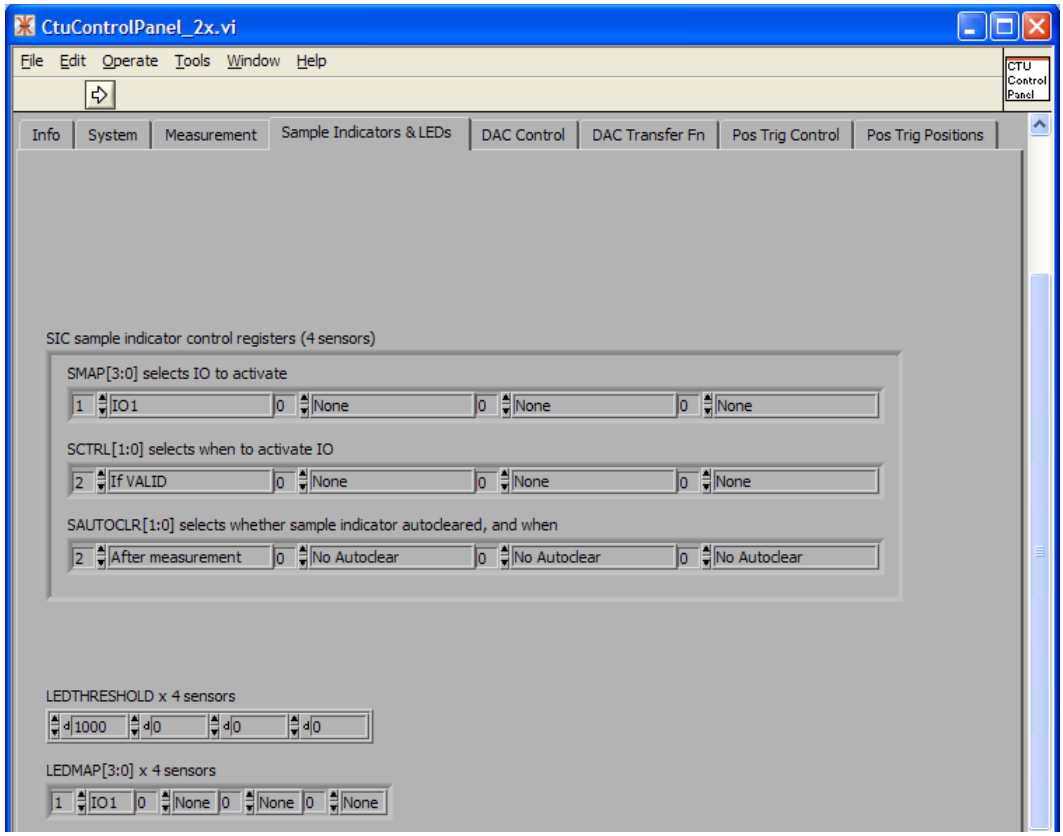


Figure 25 CTU Control Panel: Sample Indicators and LEDs tab

The Sample Indicator Control Register (SIC) controls Sample Indicators. There are controls for up to 4 sensors, arranged from left (sensor 1) to right (sensor 4). SMAP determines which of the CTU's User Ios the sensor's sample indicator function activates. SCTRL determines whether sample indicators are to occur on each new position, or only on VALID positions. SAUTOCLR determines whether the sample indicator is cleared automatically by the CTU, and when the clearing takes place. Set this to **No Autoclear** to clear sample indicators over the SPI interface by clearing the SIE bit in the SCW register, to Before Measurement to ensure the IO toggles each measurement or to After Measurement when using sample indicators to output a stable indication of a sensor's VALID bit.

The CTU can operate LEDs that are controlled by its user Ios. An LED mapped to a sensor with the **LEDMAP** control is turned on when the last measurement's Amplitude reading exceeded its **LEDTHRESHOLD** value. It is turned off when it is below the minimum Amplitude setting for the CTU. Finally, it flashes when the Amplitude is between the two thresholds, to indicate that a target is present but Amplitude is not yet sufficient. LEDTHRESHOLD is application-specific, and depends on the performance required. Performance generally improves with increasing Amplitude.

© Cambridge Integrated Circuits Ltd 2009-2022

8.5 CTU Control Panel: DAC Control

The CTU can be configured to drive an external Digital to Analog Converter, providing it is compatible with one of the supported formats. Please refer to the datasheet for the CTU device for more details.

The **DAC Control** tab is illustrated in Figure 26. There are two groups of controls: an upper set that define system-level 24ehavior applying to all sensors, and a lower set that define how measurements from each sensor update DAC outputs.

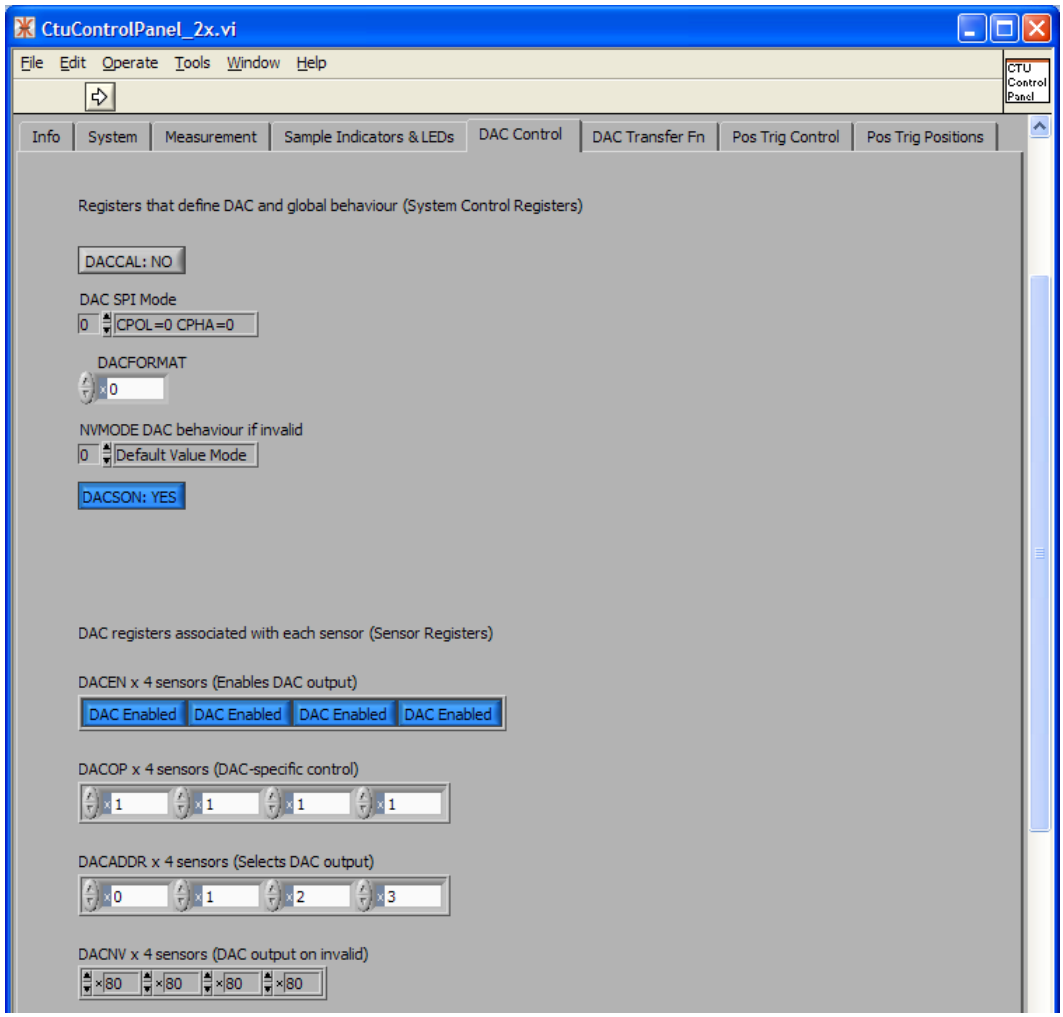


Figure 26 CTU Control Panel: DAC Control tab

Among the system registers, the **CACCAL** control is for DAC calibration and should normally be set to **NO**. The correct setting for the **DAC SPI Mode** and **DACFORMAT** controls depend on the type of DAC connected. Please refer to the CTU datasheet for the correct settings, or load a **CTU Register File** appropriate to that DAC. The **NVMODE** control determines the DAC output when sensor measurements are not VALID. The **DACSON** button must be set to **NO** when no DAC is connected, and to **YES** when DAC output is required.

The **DACEN** bits are sensor-specific. When a sensor's **DACEN** bit is enabled, the DAC will be updated with values based on that sensor. The function of the **DACOP** control is specific to each type of DAC. Settings generally include a low-power mode, and provision to synchronise the outputs of DACs having multiple outputs. **DACADDR** applies to DACs with multiple outputs, and defines which of those outputs each sensor's data is addressed to. **DACNV** controls the DAC output when sensor measurements are not VALID, in conjunction with **NVMODE**.

© Cambridge Integrated Circuits Ltd 2009-2022

8.6 CTU Control Panel: DAC Transfer Function

When an external DAC is connected to the CTU, the **DAC Transfer Function** describes the relationship between sensor measurements and the DAC output(s). Its tab is shown in Figure 27. This is arranged in 4 columns, each representing one sensor connected to the CTU.

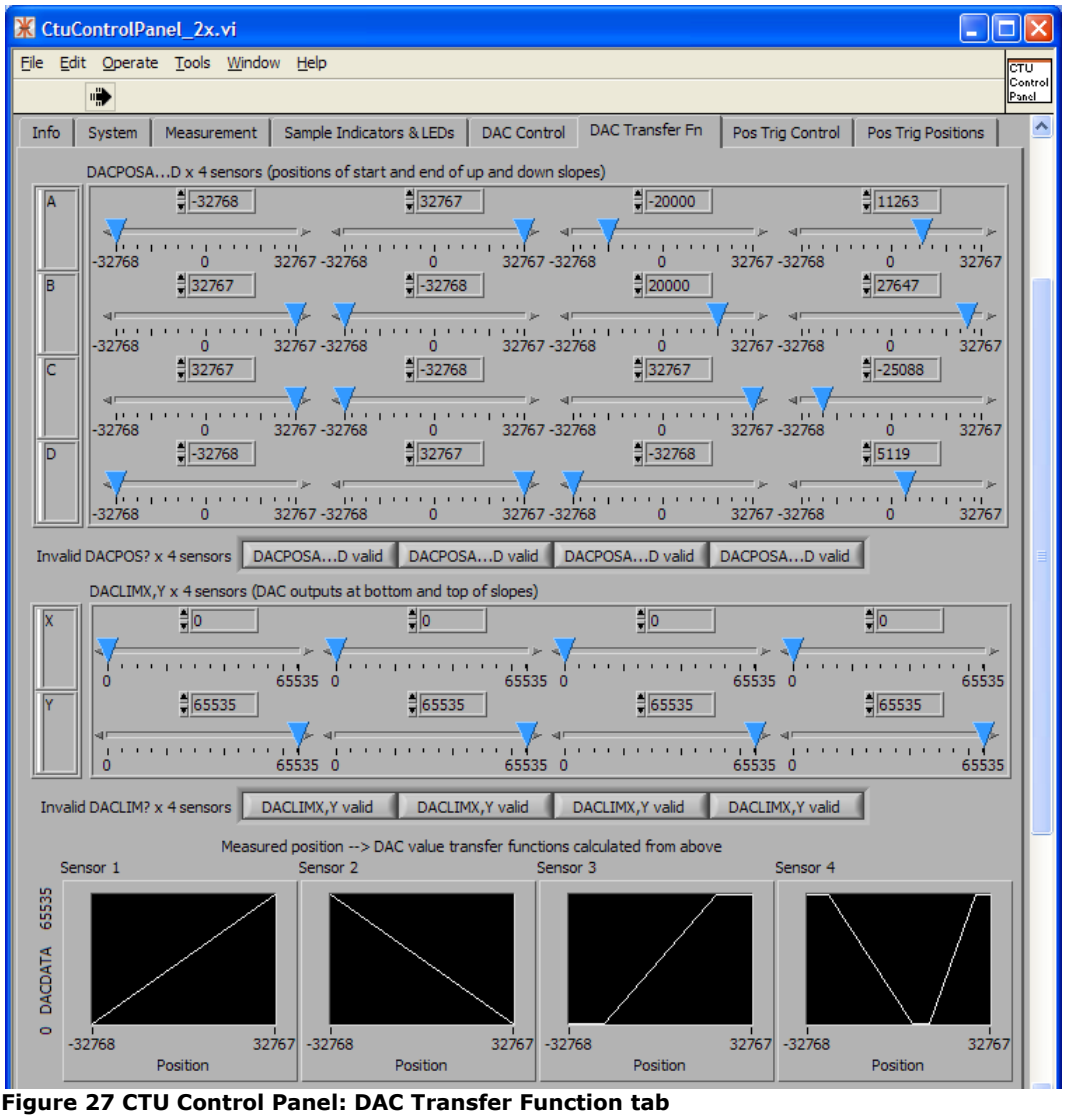


Figure 27 CTU Control Panel: DAC Transfer Function tab

The 4 upper rows of controls labeled A… D represent the CTU's **DACPOSA...D** registers. These define the start and end-points of the "up" and "down" slopes of the transfer function. Below these controls there is a row of **Invalid DACPOS?** Indicators, one for each sensor. They indicate whether the settings above are valid or not. If not, the DAC function will not work properly. For settings to be valid, **DACPOSA...D** must be ordered correctly, as set out in the CTU datasheet.

The next two rows of controls correspond to the CTU's DACLIMX and Y registers. These define the lower and uppermost DAC outputs respectively. Once again there is a set of 4 indicators below indicating whether the settings are valid, **Invalid DACLIM?** In this case, **DACLIMY** must be greater to or equal to **DACLIMX**.

The 4 graphs at the bottom of the window are plots of the **DAC Transfer Function**, based on the settings above. Like the other indicators, these will update when the application is running. The x-axis of each graph corresponds to the range of possible sensor positions. The y-axis is the DAC data that will be output by the CTU at each position. The value 0 generally corresponds to the low-end of the DAC's output range, and 65535 to its high-end.

© Cambridge Integrated Circuits Ltd 2009-2022

8.7 CTU Control Panel: Position Trigger Control

The CTU can be configured to generate Position Triggers when a sensor's measured position passes pre-set values. This sub-section describes the CTU Control Panel controls associated with Position Triggers, illustrated in Figure 28. The next section, 8.8, describes controls for the Position Trigger Positions themselves.

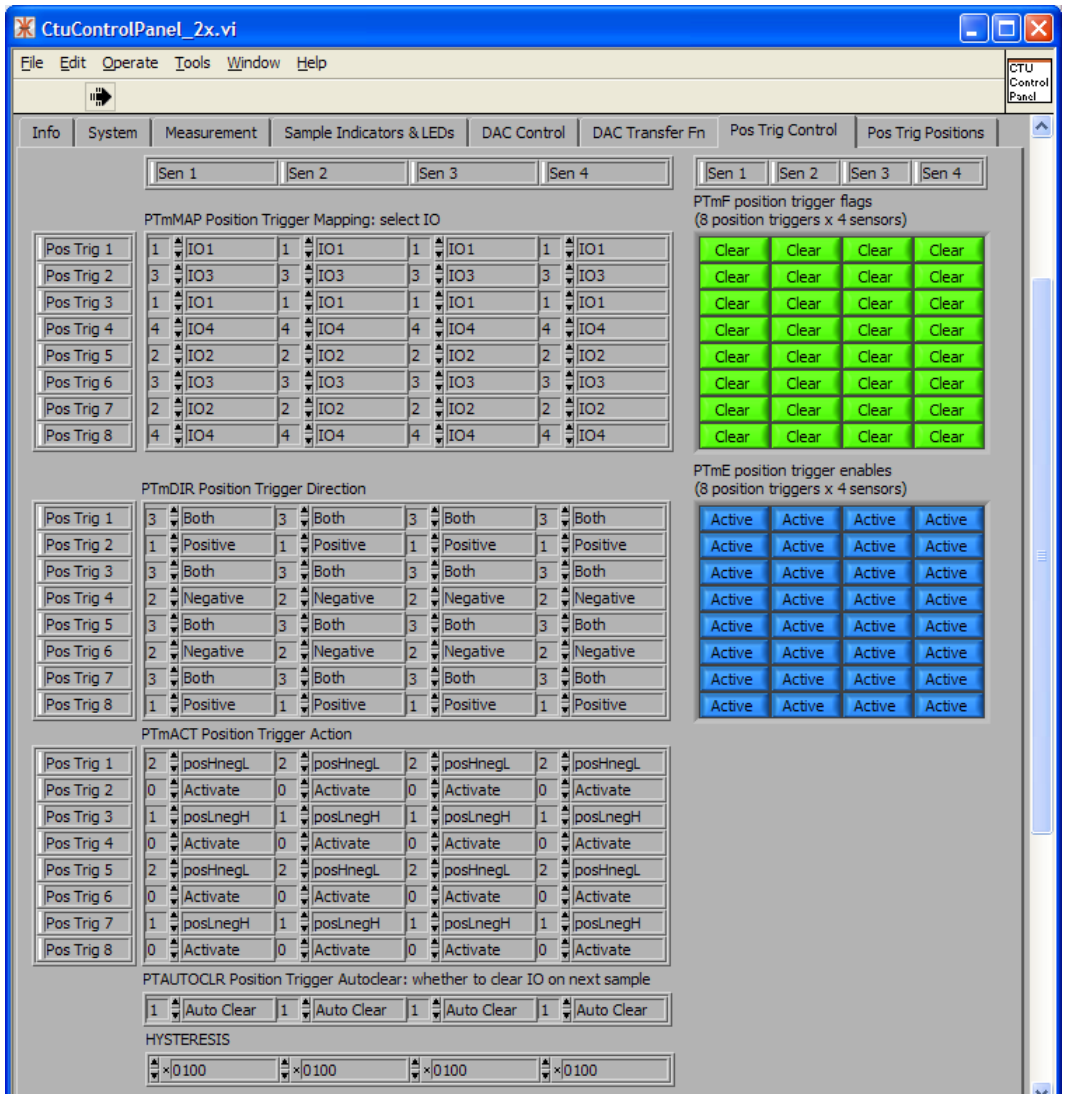


Figure 28 CTU Control Panel: Position Trigger Control tab

The controls are all arranged in 4 columns, one for each sensor. The 5 upper controls are arranged in 8 rows, one for each of 8 available Position Triggers per sensor.

PTmMAP controls which of the CTU's user Ios each Position Trigger activates. **PTmDIR** controls whether the Position Trigger is activated when measured position passes the Position Trigger Position in a **Positive** direction, **Negative** direction or **Both**.

PTmACT controls the Action that the CTU takes on a Position Trigger event. The options are **Activate**, which pulses the IO active, **PosLnegH** which sets the IO low if the trigger was positive or high if negative, **PosHnegL** which has the opposite effect, or **Toggle**, which toggles the Ios state between high and low. **PosLnegH** and **PosHnegL** are usually used in pairs to set and reset Ios as a function of position.

PtmE is a block of enables for each Position Trigger and each sensor. **PTmF** is a set of flags reflecting whether or not Position Triggers have occurred. To clear a Position Trigger over the SPI interface set the appropriate button to **Clear**.

Position Triggers can also be cleared by the CTU by setting the **PTAUTOCLR** bit to **Auto Clear**. This is usually appropriate when the CTU is to generate IO pulses on each Position Trigger.

© Cambridge Integrated Circuits Ltd 2009-2022

The **HYSTERESIS** control is used to prevent multiple activations of Position Triggers due to noise, which can happen when the measured position coincides with a **Position Trigger Position**. A small value, such as 0x0100 for a Type 1 sensor, is usually sufficient. **HYSTERESIS** also applies to the DAC output, and it is recommended that it be set to 0 when the CTU is configured to control and external DAC, unless the application specifically requires different.

8.8 CTU Control Panel: Position Trigger Positions

The previous section, 8.7, was about how to use CTU Control Panel software to configure Position Trigger 27ehavior. This section is about setting the **Position Trigger Positions** themselves.

Figure 29 shows the **Position Trigger Positions** tab. Each column of controls represents one sensor, and each row represents one of the 8 available Position Triggers.

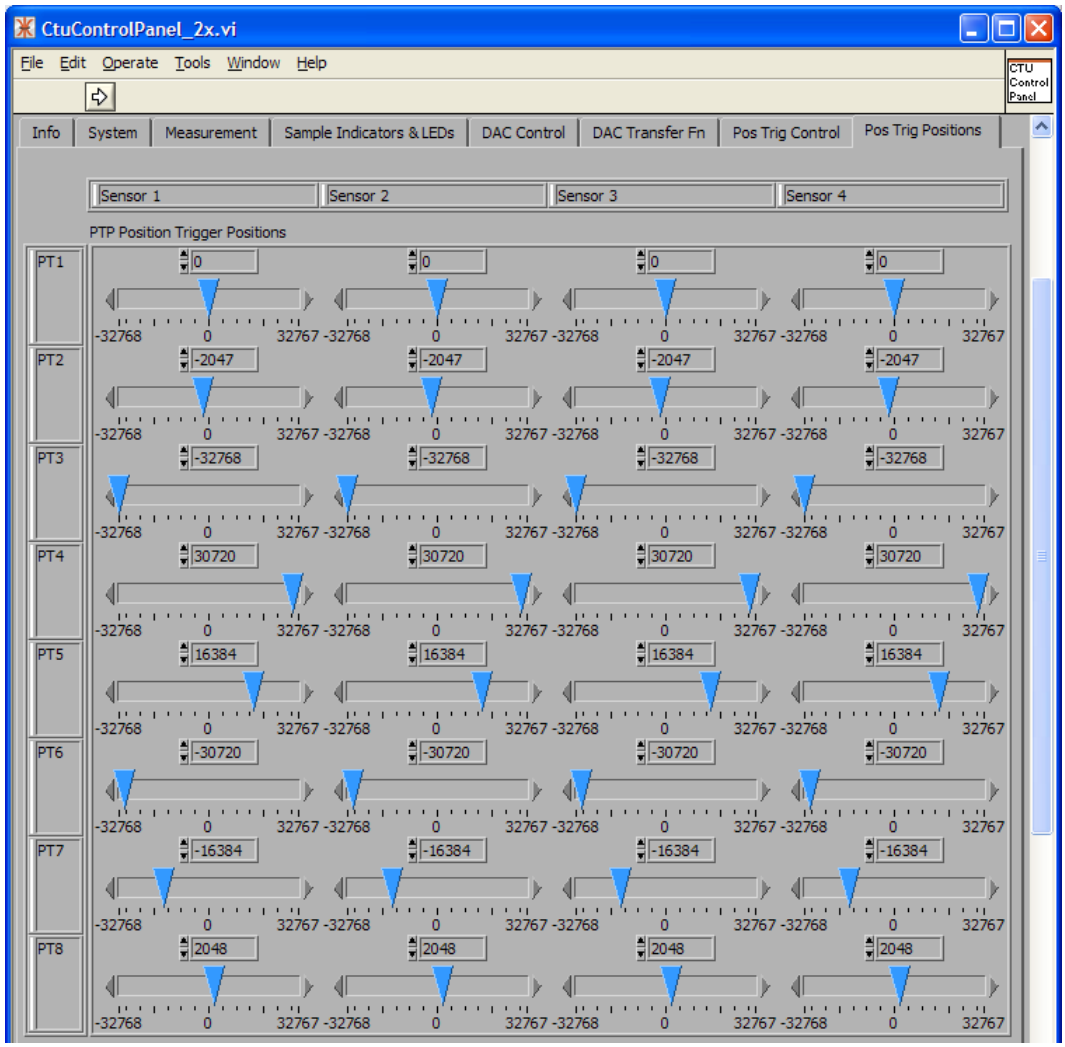


Figure 29 CTU Control Panel: Position Trigger Positions tab

Position Trigger Positions are signed 16-bit integers (in the range -32768 to +32767). This corresponds to position as reported by the CTU when sensor Type is set to 1 or 4. Its relationship to physical units such as degrees or millimeters is described by Equation 1 in section 5.1. The relationship is scaled to allow 16-bit register representation for other sensor Types. Please see the CTU's datasheet for details on this scaling.

8.9 Operation with the CAM502 CTU Chip

To operate with a CAM502 chip correctly, the **CTU (CAM204=0, CAM502=1)** control must be set to 1. This control is visible when the main window is expanded downwards. **FCSEL** and **ABSSEL** controls are provided on the measurement tab (Figure 24), and these operate as described in section 5.4.

© Cambridge Integrated Circuits Ltd 2009-2022

9 Update Adapter Firmware

This application updates the CTU Adapter firmware. This enables new features or bug fixes to be uploaded.

Figure 30 illustrates the application's window. Press the **Run** button if the application is not already running. The application prompts for the location of the file to program. The file and location may also be changed by pressing the **Change Adapter Firmware File** button.

The **Adapter Firmware File comments** box displays the file's header information, including version information.

Press the **Update Adapter with firmware** button to start the upload process. The process usually takes less than 5 seconds. If successful, the **Adapter and DLL info** box show the updated version information.

The **Software Version** box includes version information for the newly loaded firmware. Once correctly programmed, this should match the version information in the Adapter Firmware File's comments section. The update process does not change the bootloader.

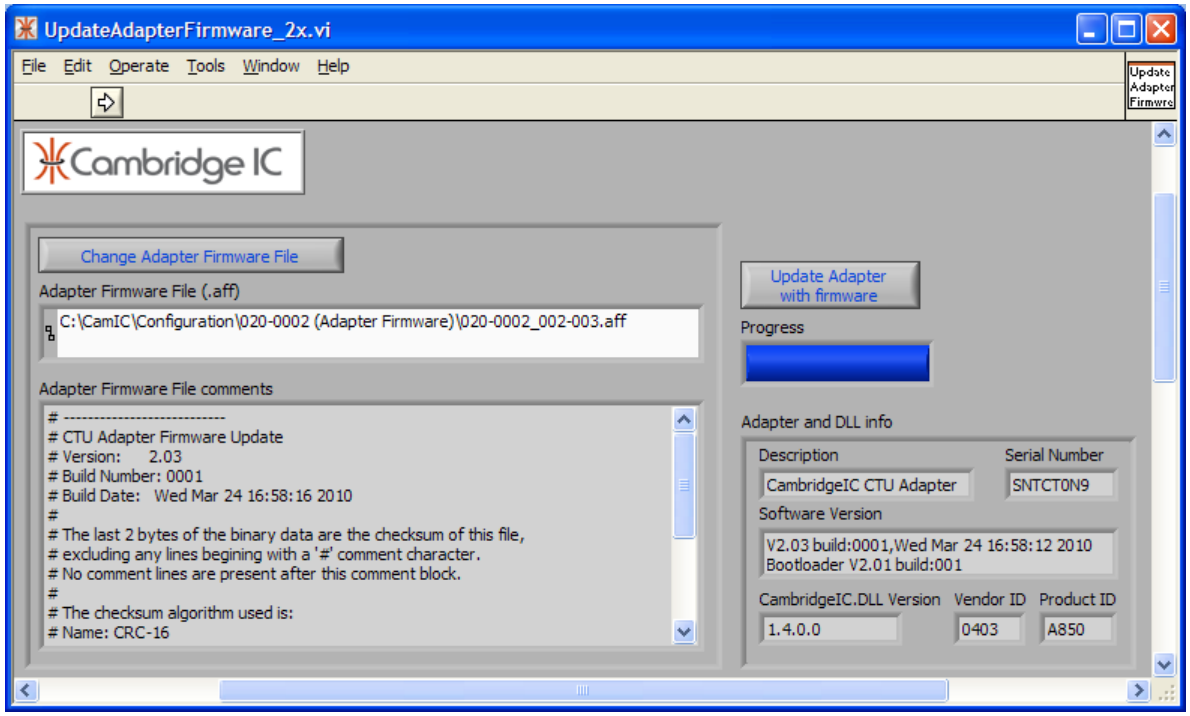


Figure 30 Update Adapter Firmware screen shot

© Cambridge Integrated Circuits Ltd 2009-2022

10 Update CTU Firmware

This application updates the firmware that runs inside CTU chips. This enables new features or bug fixes to be uploaded to the CTU using a PC connected to an Adapter.

Figure 31 illustrates the application's window. Press the **Run** button if the application is not already running. The application prompts for the location of the file to program. The file and location may also be changed by pressing the **Change CTU Firmware File** button.

The **CTU Firmware File comments** box displays the file's header information, including version information.

The button labeled **CTU contains .cff already** determines the method that the Adapter uses to initiate firmware upload to the CTU. Set this to **Yes** if the CTU contains valid firmware already (for example, if works with the CTU Demo application). Set it to **No** if the CTU does not contain valid firmware, for example because a previous firmware upload was stopped before completion. When set to **No**, the upload process can only begin immediately after the CTU is reset. The Adapter tries to reset the CTU itself by toggling its nRESET output and by dropping its power output to zero briefly. In some cases these may not be connected to the CTU, for example if the Adapter is being used to upload firmware to a CTU chip that is integrated with a customer's product with its own supply and no nRESET connection. If this is the case, the host system must reset the CTU and not attempt any SPI communication with it before upload.

Press the **Update CTU with firmware** button to start the upload process. The process usually takes less than 10 seconds. If successful, the **CTU info** box shows updated version information. The **System Version** should match the version number quoted in the CTU Firmware File comments box. The bootloader is fixed, and the update process does not change the **Bootloader Version**.

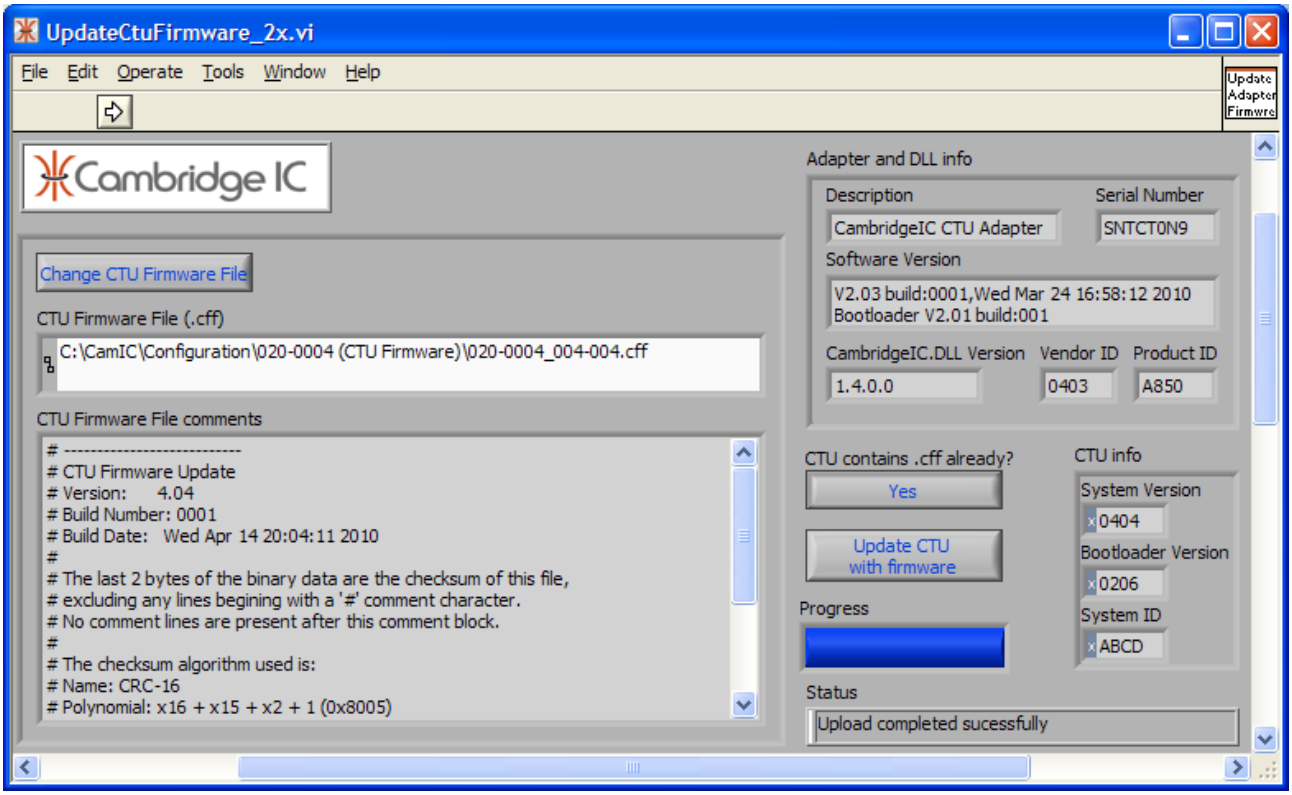


Figure 31 Update CTU Firmware screen shot

The upload process is performed over the CTU's SPI interface and the process is documented in its datasheet. It is therefore also possible to implement firmware upload through the host processor in a customer's own product.

© Cambridge Integrated Circuits Ltd 2009-2022

11 Troubleshooting Guide

Symptom	Possible cause	Rectification
Error 5001: DEVICE NOT FOUND	Adapter not plugged into USB port	Plug it in
	Adapter driver not installed	Install driver (see section **Error! Reference s ource not found.**)
	Adapter driver disabled	Re-Enable driver (see section **Error! Reference s ource not found.**)
	Adapter not yet enumerated	Wait 5 seconds between plugging in the Adapter and running any application that uses it
Error 5004: Read timeout…	Adapter power selector link missing	Check there is a link selecting between 2V7, 3V3 and 3V6 fitted through the Adapter's rear panel.
Error 5008: Adapter Error	USB communications with the Adapter were not properly closed before it was stopped.	Close down the application and run it again.
		Unplug the Adapter from the PC's USB port then plug it back in again and wait 5 seconds
Error 5012: Did not get expected SYSID from CTU	Adapter to CTU SPI interface not properly connected	Check SPI connections
	CTU chip not powered	Check power is supplied to the CTU by the Adapter or other means
	SPI Expander not working	Check SPI Expander, if used
	SPI Expander not present	Turn off multi-CTU support (see section 4.4)
Error 5013: SYSID = 0x10AD: CTU firmware missing	CTU firmware update was interrupted while in progress	Try uploading CTU firmware again
Error 5107: Failed mid-stream		Upload CTU firmware
CTU does not report VALID when a target is present	A sensor or excitation coil connection is missing or shorted	Check sensor and its connections
	A Type 2 sensor is being used with Sensor Type set to 1	Change **Sensor Type** to match the sensor type connected
	The CTU is measuring a different Type 1 sensor	Check that **Sensor number** matches the CTU input number that the sensor is connected to
	The CTU is measuring an invalid Type 2 sensor number	Check that **Sensor number** is set to 1 for a Type 2 sensor
	CTU's excitation circuit not working	Check the circuit conforms to the CTU datasheet
	The target is not working	Try a different target
	The target's frequency has been changed by nearby metal beyond the CTU's tuning range	Check for metal closer to target and sensor than specified in their datasheets
	Misaligned target	Check sensor datasheet for correct alignment
Reported position does not change smoothly with target position	A sensor coil connection is missing or shorted	Check sensor and its connections
	A Type 1 sensor is being used with Sensor Type set to 2	Change **Sensor Type** to match the sensor type connected
Reported position incorrect	Incorrect value of Sin Length for the sensor	Check that the value used matches the appropriate value from the sensor's datasheet
Other errors	Try closing and the software then opening and running it again.	
	Try unplugging the Adapter from the USB port and plugging it back in again.	

© Cambridge Integrated Circuits Ltd 2009-2022

12 Document History

Revision	Date	Description
0001	12 August 2009	First draft
0002	9 September 2009	Added Save to File
0003	14 September 2009	Revised to reflect minor updates to software
0004	5 November 2009	Revised to reflect software name change
0005	28 January 2010	Updated logo and style
0006	21 July 2010	Changed to reflect CTU Software version 2.x
0007	11 August 2010	Updated supported Windows versions
0008	15 March 2012	Updated now that the driver is "signed", and supports Windows 7
0009	13 November 2012	Updated with details of operation with Type 6 sensors
0010	18 December 2014	Updated with advanced controls for CAM502 operation (version 2.7 on)
0011	7 February 2019	Added note on requirement for Microsoft .NET Framework 3.5
0012	20 January 2022	Updated driver install process: now manual process using Windows Update

13 Contact Information

Cambridge Integrated Circuits Ltd
21 Sedley Taylor Road
Cambridge
CB2 8PW
UK

Tel: +44 (0) 1223 413500

info@cambridgeic.com

14 Legal

This document is © 2009-2022 Cambridge Integrated Circuits Ltd (CambridgeIC). It may not be reproduced, in whole or part, either in written or electronic form, without the consent of CambridgeIC. This document is subject to change without notice. It, and the products described in it ("Products"), are supplied on an as-is basis, and no warranty as to their suitability for any particular purpose is either made or implied. CambridgeIC will not accept any claim for damages as a result of the failure of the Products. The Products are not intended for use in medical applications, or other applications where their failure might reasonably be expected to result in personal injury. The publication of this document does not imply any license to use patents or other intellectual property rights.