Software Design Elements of HMS

Brian Eng, Mary Ann Antonioli, Peter Bonneau, Tanest Chinwanawich, Michael Ferguson,
Tony Madany, Marc McMullen, Armenak Stepanyan, Werth Teachey and Amrit Yegneswaran

Physics Division, Thomas Jefferson National Accelerator Facility, Newport News, Va 23606
January 5, 2004

HMS Monitoring System (HMS), which monitors the electronic instrumentation of the Continuous Electron Beam Accelerator Facility's Large Acceptance Spectrometer (CLAS) at Jefferson Laboratory uses a National Instruments' (NI) developer suite software, LabVIEW, whose files called *Virtual Instruments (VI)* consist of *front-panels* that provide interface-elements such as indicators, buttons and graphs and *block diagrams* that outline data-flow. An overview of software design elements and descriptions of the front panel and the block diagram elements are presented in this note.

Figure 0, shows the monitoring flow chart of HMS[1].

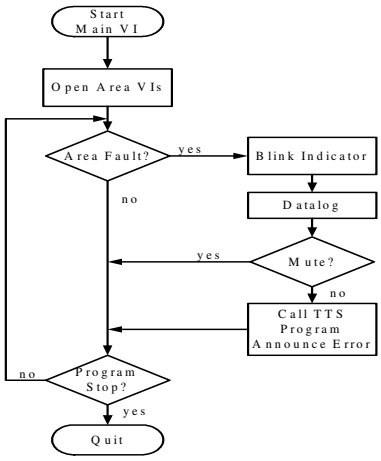

Figure 0: Program Flowchart.

Main.VI calls in the code's ***Open Area VIs*** section ***Area VIs*** that check the performance-indicator signals in specified areas, *e.g.*, ***Area.VI***: ***SFL1N.VI*** monitors Space Frame Level 1 North. If an ***Area.VI*** detects a signal fault, signal out of operational limits, then the information is broadcasted to the ***Main VI*** and concomitantly the software's ***Blink Indicator*** part makes that area's indicator display a signal fault status. Subsequently the ***Datalog*** section of the code logs date, time, error address and error information in an EXCEL spreadsheet. After which ***Call Text To Speech (TTS)*** issues an audio-visual alarm to alert personnel, provided the mute feature for the CDR that detected the error is not selected. Finally the choice to quit program is presented by the ***Program Stop*** section which if not opted permits ***Main VI*** to begin the loop again and which if opted causes ***Main VI*** to signal ***Area VIs*** to close and wait for all *sub-VIs* to terminate before halting execution.

Figure 1 shows the three levels of the monitoring program's hierarchy, Main, Area and Individual Crates. The notation for the different areas is:

FCD# ≙ Forward Carriage Deck #; (# ε1, 2, 3, 4)

SFL#N/S ≙ Space Frame Level # North/South; (# ε1, 2, 3)

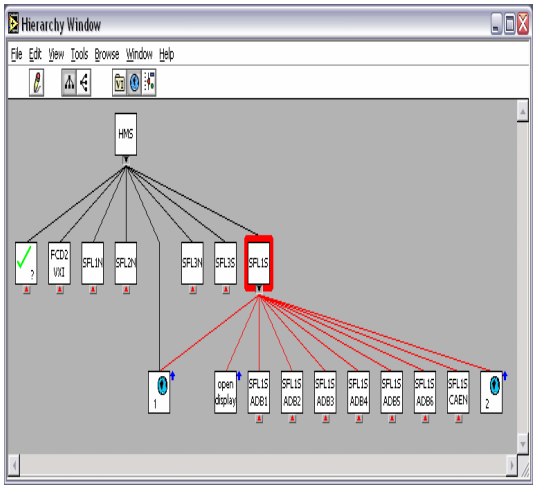

Figure 1: Software Hierarchy.
Only SFL1S area's Crate VIs are shown.

In the main front panel monitored areas are arranged by location in the end station, Figure 2. Green and ghost buttons indicate active and inactive areas respectively. In the event of a fault, crate-parameter signal-value out of operational limit, the green buttons flash red. Also the GUI has a circular status light in the lower right hand corner, which indicates the status of the HMS crates.

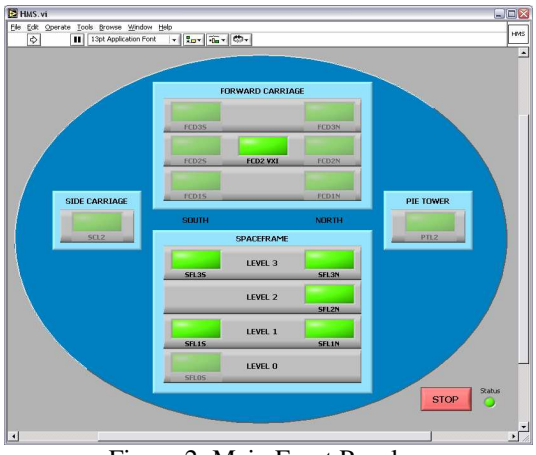

Figure 2: Main Front Panel.

[1] Antonioli *et.al* A Novel Monitoring System for Electronic Instrumentation. **CLAS-NOTE 2003-018**

Area Front Panels, Figure 3, show the status of individual crates that are being monitored. The area front-panel screen is displayed when the button for an area on the Main Front Panel is clicked (all areas are displayed similarly). The buttons are green when the performance indicator signal-values are within operational limits, flash red in the event of a fault. Checking the boxes below the status indicators mutes the Text To Speech error message for that crate. The CLOSE button in the lower right corner closes the Area-Front-Panel but does not stop program execution

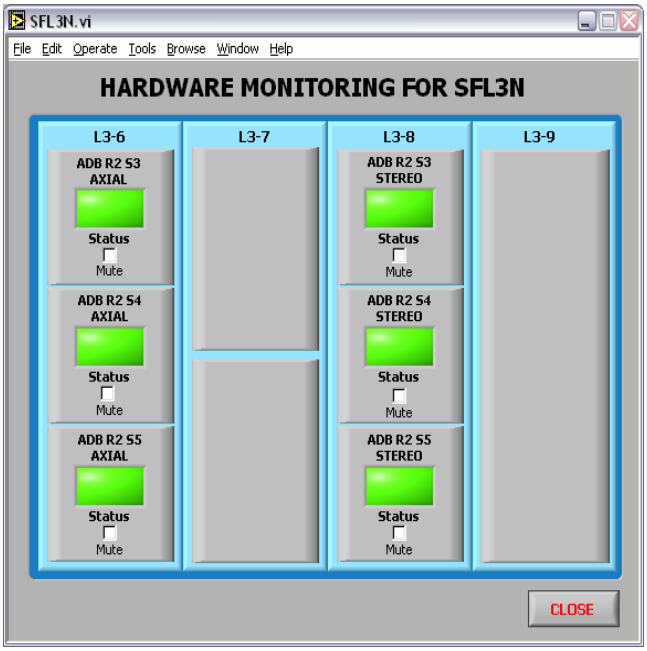

Figure 3: Area Panel of Space Frame Level 3 North.

Clicking on a crate button in the Area-Front-Panel opens the Individual-Crate-Front-Panel which displays performance indicator signals monitored for that crate; Figure 4 shows details of the Region three Sector three ADB crate.

Figure 4: Example of Individual-Crate-Front-Panel

Figure 5 shows an error condition – the DC power failure on the crate is indicated by the red button.

Figure 5: Individual-Crate-Front-Panel with an error

The AC ON button changes to reflect AC Power status for that crate and automatically reverts to the AC ON position once reset, which is accomplished by clicking the AC ON button and cycling the AC power delivered to the crate. An additional step is needed when Level 2 Trigger is being used. The Level 2 (L2) Trigger must be reset manually from the front panel of the L2 card, as it is unaffected by an AC Reset.

When the stop button is pressed a dialog box pops up requiring confirmation, Figure 6.

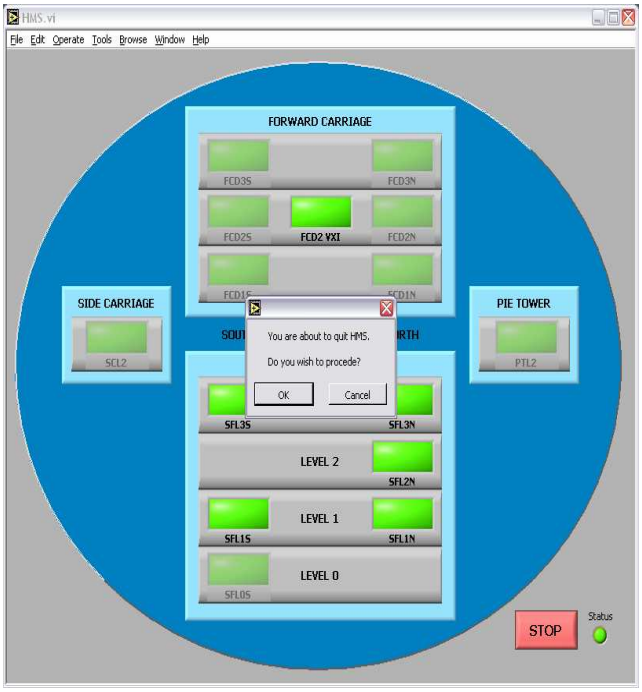

Figure 6: Main Front Panel. After STOP is Pressed

If the OK button is pressed the program will begin to shut down monitoring, logging and annunciation of errors – a process, which takes several minutes

The colored *block diagram* of the Main VI shows the functions performed.

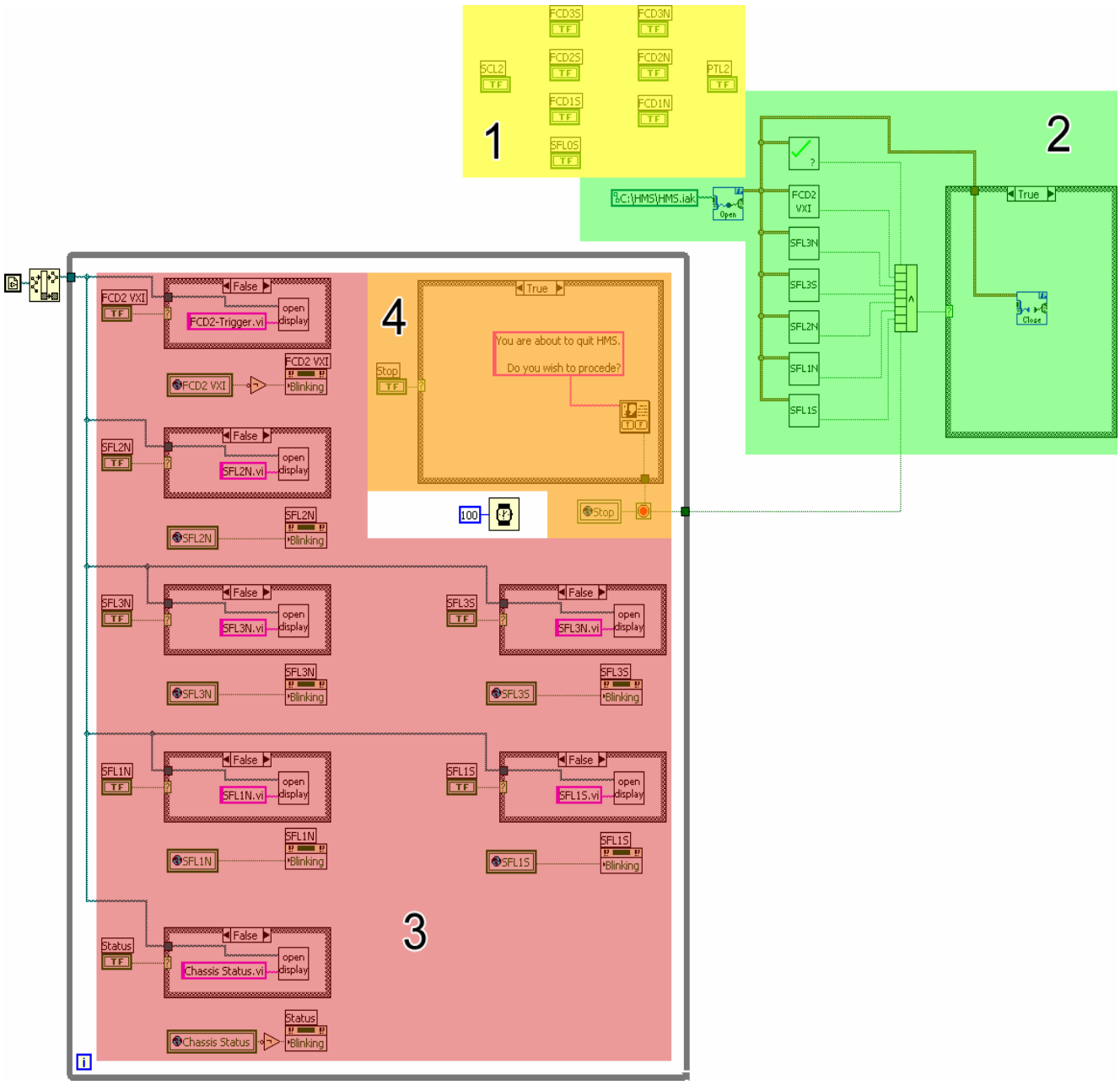

Figure 7: Block Diagram – Main VI

1. The yellow part shows indicators on the front panel that haven't been activated.
2. The green section is the area where connection to FieldPoint Explorer is opened and information is passed to the area's virtual instruments.
3. The red portion indicates the part where mouse clicks are detected and the associated areas' virtual instruments are opened.
4. The orange area shows the stop button part needed to halt execution of the program.

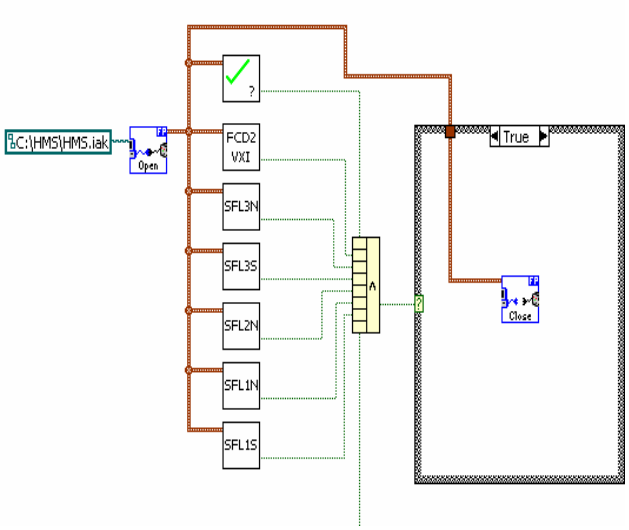

Figure 8: Block Diagram – Main VI
FieldPoint Connection

In Figure 8 the reference generated from a FieldPoint Open call is passed to the Area-VIs. After the Area-VIs have completed execution a boolean AND is performed on the outputs generated. Once the AND function is completed FieldPoint Close is called.

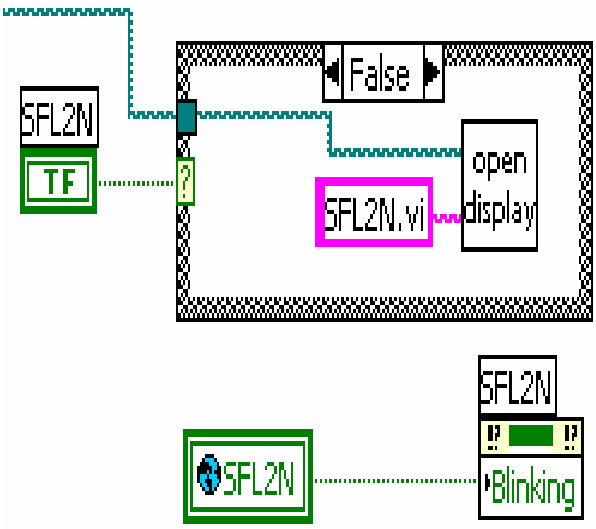

Figure 9: Block Diagram – Main VI
Blink detail

Figure 9 details the code used to handle the indicator for the various areas, in this case SFL2N. The code inside the rectangle (Open Display, which is detailed in Figure 10) is executed once the button has been pressed. The code on the bottom causes the indicator to blink on an error.

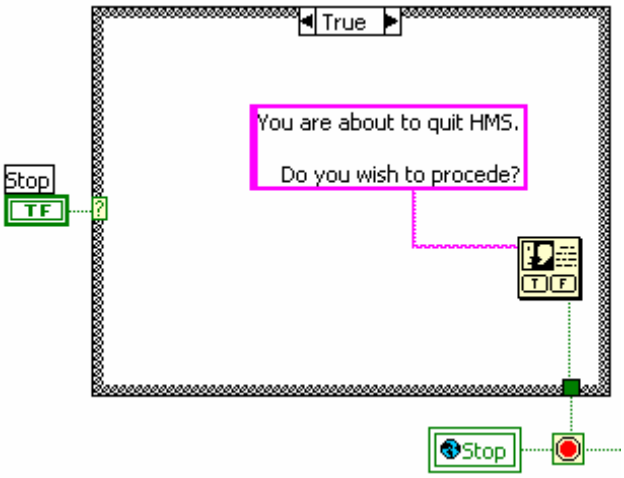

Figure 10: Block Diagram – Main VI
FieldPoint Close detail

Figure 10 shows that when the user presses the STOP button and it becomes true a dialog is presented, which is passed to the Stop variable that is globally accessible to all HMS VIs.

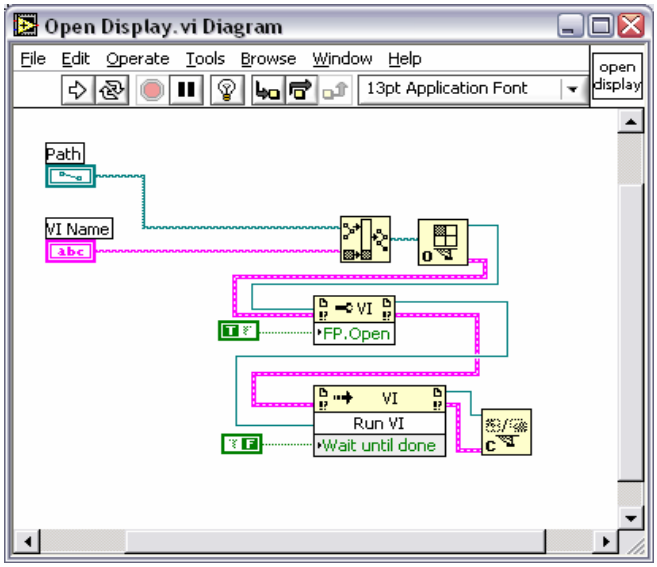

Figure 11: Block Diagram – Main VI
Open Display VI

The Open Display VI , Figure 11, accepts a file path and VI-Name and combines the two so that the requested VI front panel will be opened.

The Area-VIs are one level below the Main-VI as seen in Figure 12. The Area-VIs group individual crates by area into a single VI, which has a layout similar to the Main-VI. This example shown is for SFL3N.

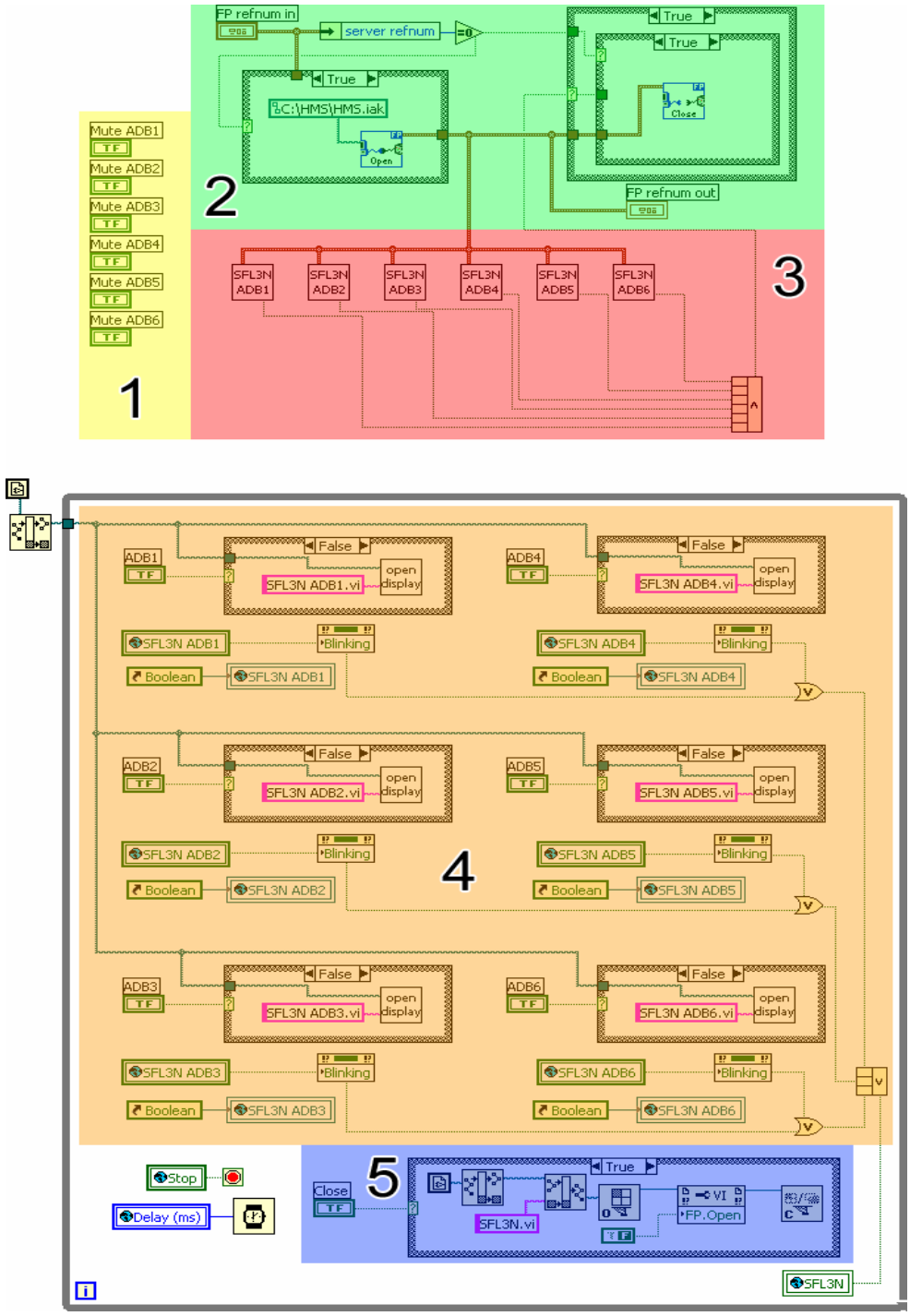

Figure 12: Area VI

1. The yellow section contains the mute buttons for each crate.
2. The green area is where the FieldPoint connection is opened (if needed) and closed (if needed).
3. The red area contains all the individual Crate-VIs that is monitored in this particular Area VI.
4. The red portion detects mouse clicks and opens the associated area VI. This is exactly the same as in the Main VI
5. The blue portion of the block diagram detects when the CLOSE button has been pressed and closes the Area VI front panel.

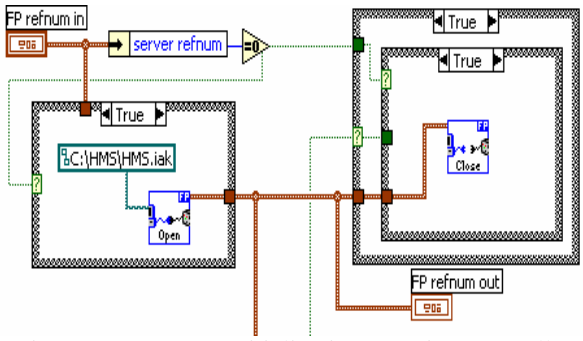

Figure 13: Area VI Initialization & Exit – No Caller

Figure 13 shows the steps when a FieldPoint reference (*FP refnum in*) is not supplied, more precisely the Area VI is being run as the top level VI (not being called by the Main HMS VI). When run as a top level VI, the Area VI will open and close the connection to FieldPoint.

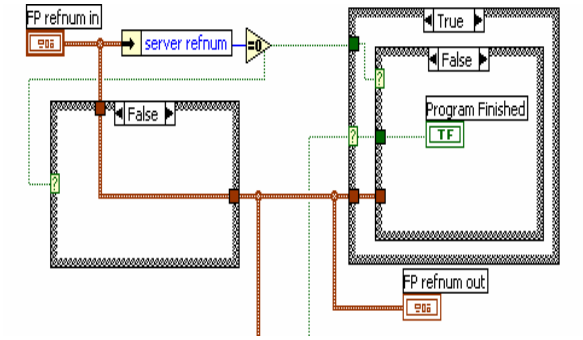

Figure 14: Area VI Initialization & Exit – Called

Figure 14 is the case when the Area VI has been called by another VI and passed a *FP refnum*. When this happens the Area VI uses the passed *refnum* and returns a boolean called Program Finished to the calling VI to signal it has run to completion.

Figure 15 details the portion of the Area VI that closes the front panel after the CLOSE button has been pressed. It is important to note that doing that only closes the front panel window but does not stop execution of the VI.

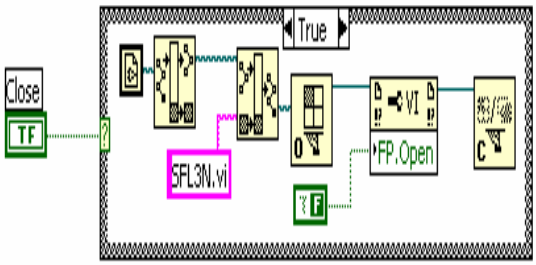

Figure 15: Close Front Panel Detail

The Individual-Crate VIs are responsible for monitoring the individual channels of each crate. While the signals monitored vary from crate to crate execution style is similar. Figure 16 shows the block diagram for an ADB crate.

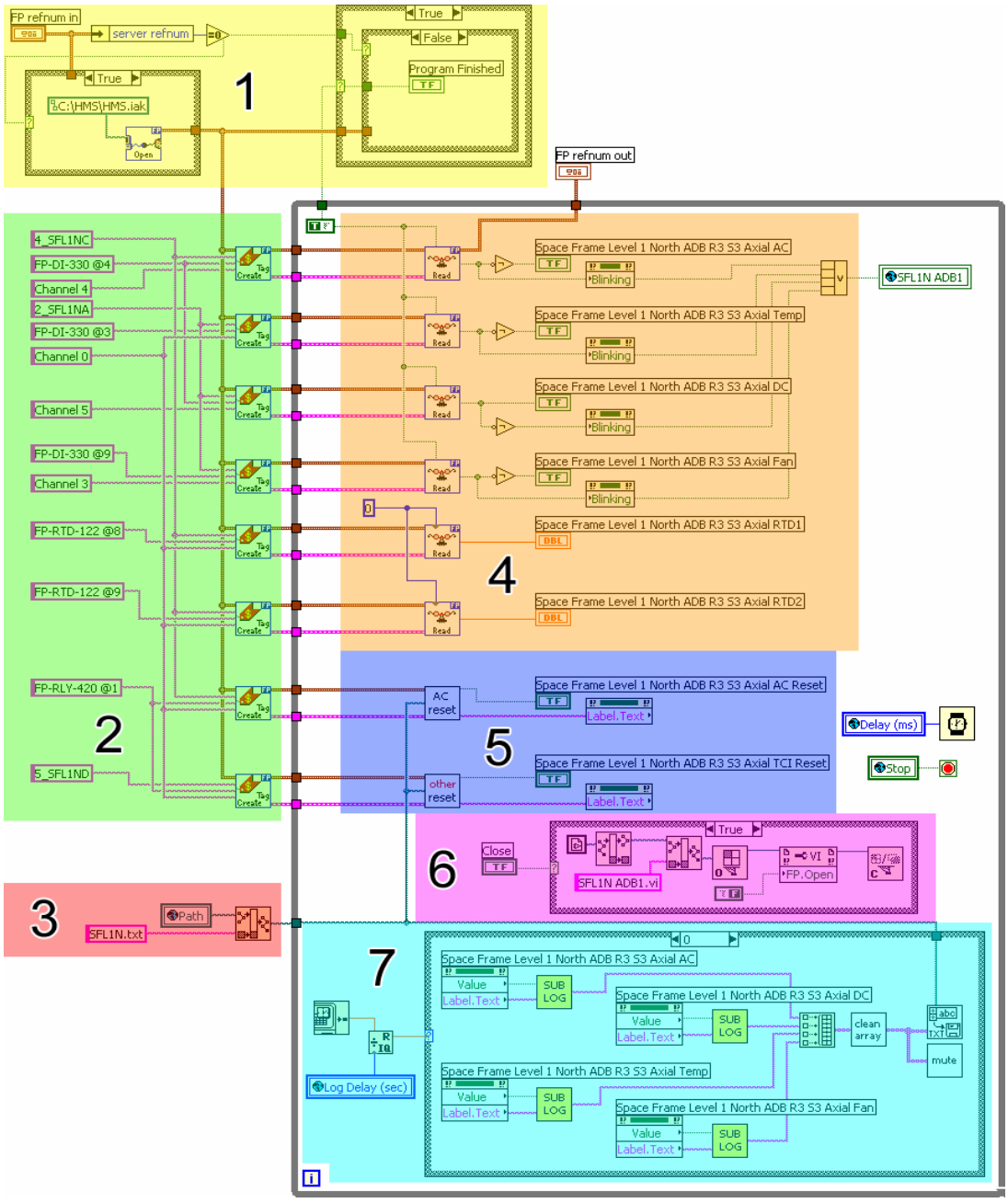

Figure 16: Crate VI

1. The yellow area is where the FieldPoint connection is opened (if needed) and closed (if needed).
2. In the green block is where the FieldPoint tags for each channel are created, a channel corresponds to something (a voltage, current, temperature, etc) being monitored.
3. The file where errors are logged to is set in the red section
4. Inside the orange block is where the channels are read and if in error causes the corresponding indicator to blink. A single error in any of the channels causes the entire crate to report an error status.
5. The blue area is where the resets are performed.
6. The purple section closes the front panel.
7. Finally, the light blue box contains the error logging and text-to-speech (TTS) functions the details of which can be seen in Figure 26.

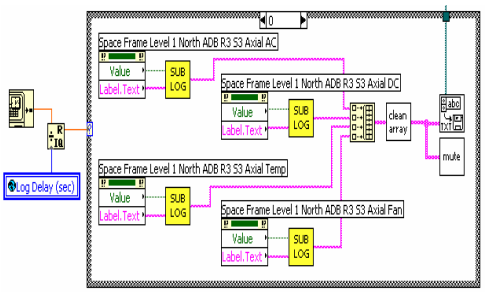

Figure 17: Crate VI error logging & TTS

On the left is the portion of code that delays the logging so that it will only write to file and annunciate errors based on what Log Delay (sec) is set to, by default it is 60 seconds. Once the specified time has elapsed, the code in the case statement is executed. The Sub Log VI, detailed in Figure 17, converts a channel's text based on its value into a string. These strings are then added to an array and passed to the Clean Array VI, which removes any empty cells from the array. Finally, the stripped array is written to a file and sent to the TTS portion of HMS.

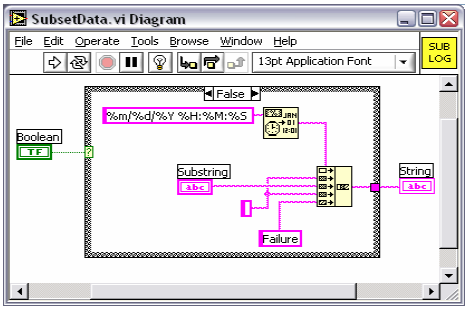

Figure 18: Sub Log VI on error

Figure 18 illustrates what happens when the Sub Log VI gets passed a channel that has an error. A string is created that contains the time and the name of the channel that was bad.

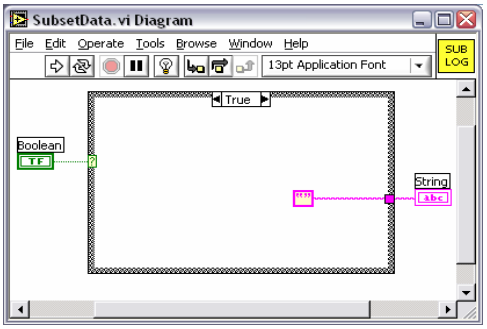

Figure 19: Sub Log VI with no error

In Figure 19 the steps performed when there is no error can be seen. In this case the Sub Log VI returns an empty string.

Figures layout the steps in performing the various resets used in HMS. The ones below show an AC Reset which is used to reset the BiRa AC Reset Modules. The other resets follow the exact same steps with the only difference being the value of the case statement; the AC Reset VI executes on False while the other resets run on True.

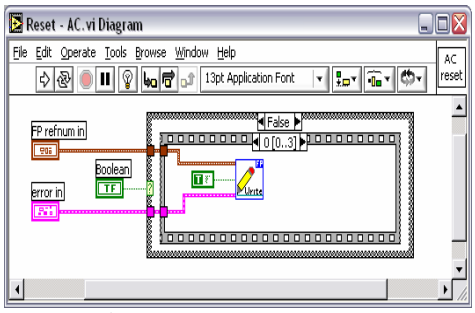

Figure 20: AC Reset step 1

The first step of the VI is to write a TRUE to the FP-RLY-420 module.

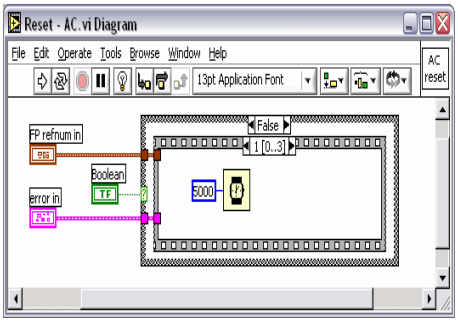

Figure 21: AC Reset step 2

Next the VI pauses for 5 seconds to ensure that the value has had time to propagate across the network and that it has been set for sufficient time.

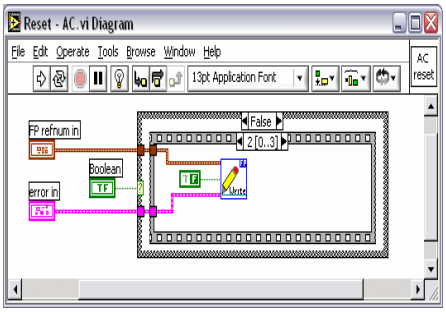

Figure 23: AC Reset step 3

After pausing for 5 seconds the VI writes a FALSE to the module, which is the default value.

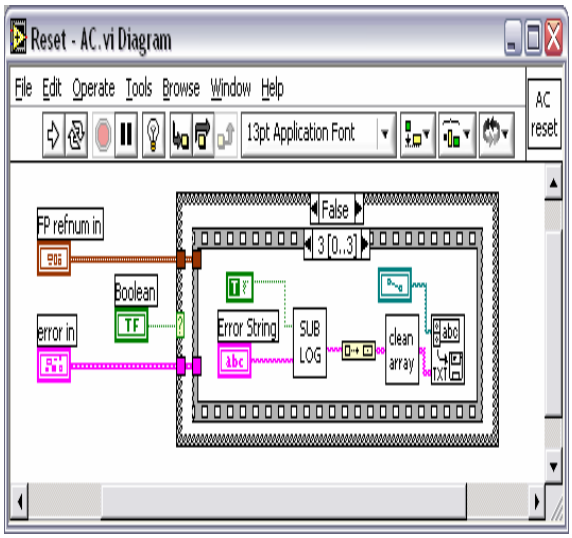

Figure 24: AC Reset step 4

The last step is to write to the log file that a reset has occurred; this is done with identical steps to those shown in Figure 24.

The TTS portion of HMS has several VIs that are used to determine which errors are annunciated. Unfortunately the code for them is much too large to be displayed in a single window so only the main portion will be shown. The part that has been omitted simply initializes each mute button by determining if it has been selected and which crate is to be muted.

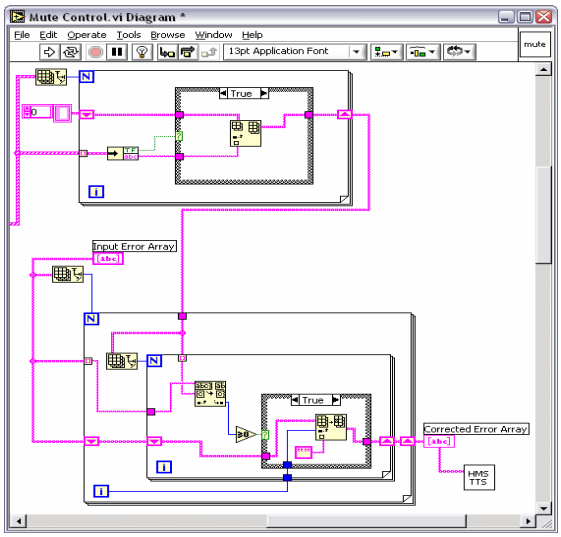

Figure 25: Mute Control

The loop on the top creates an array of all the crates that have their mute buttons enabled. The second loop compares these values with the input error array. If there is a match then that particular crate will be omitted from the array passed to HMS TTS and will not be annunciated.

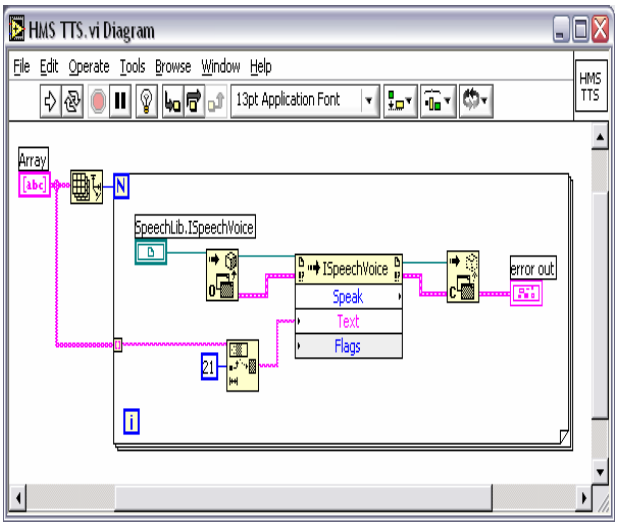

Figure 26: Text To Speech (TTS) VI

An automation refnum is passed to an automation open which creates the link between LabVIEW and the AT&T TTS software. An invoke node passes the string that is to be annunciated and finally an automation close is called to prevent memory leaks.

In summary, the software has been developed in such a way that it can be easily enlarged to incorporate additional monitoring units without significant increase in monitoring time. As implemented all elements of the program are working well.