

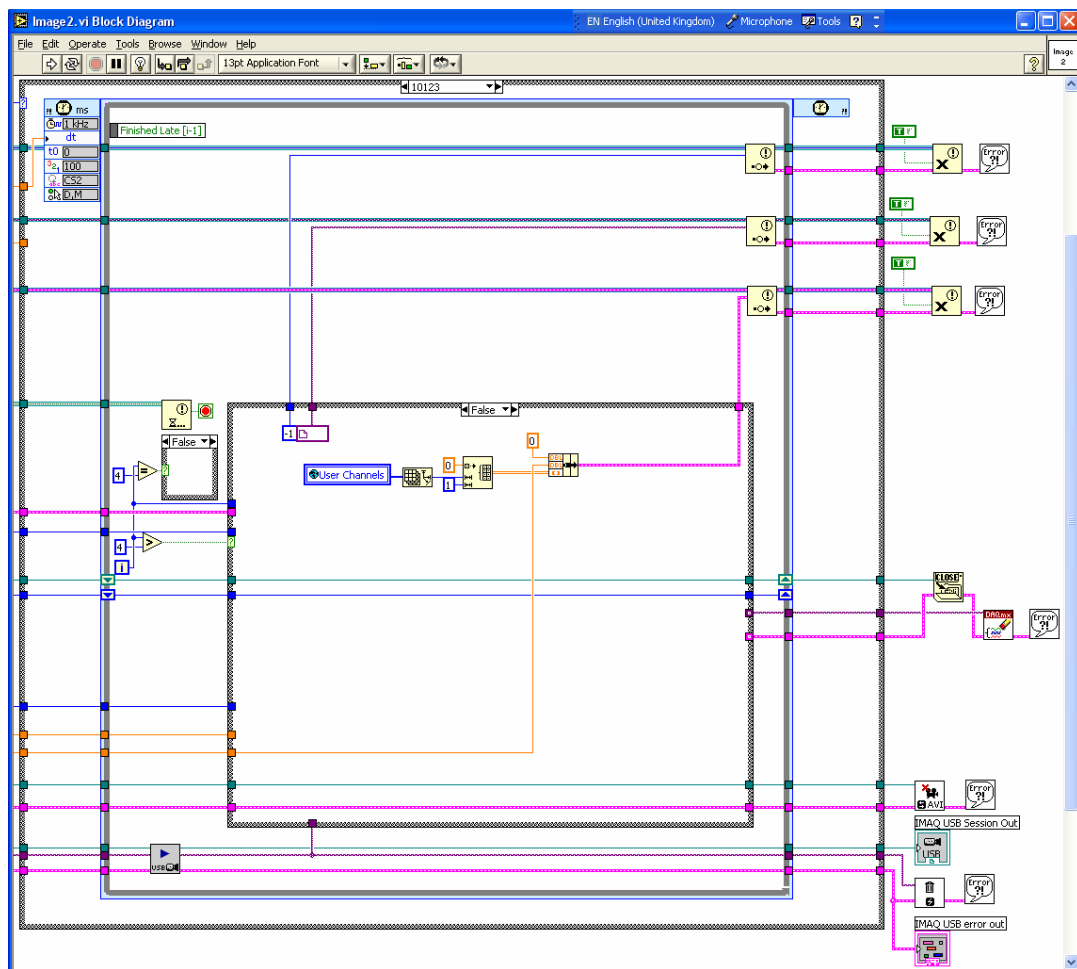


Figure 3.140: Illustration of the code of the function "Image2" that waits for the fifth USB camera frame to start the data acquisition

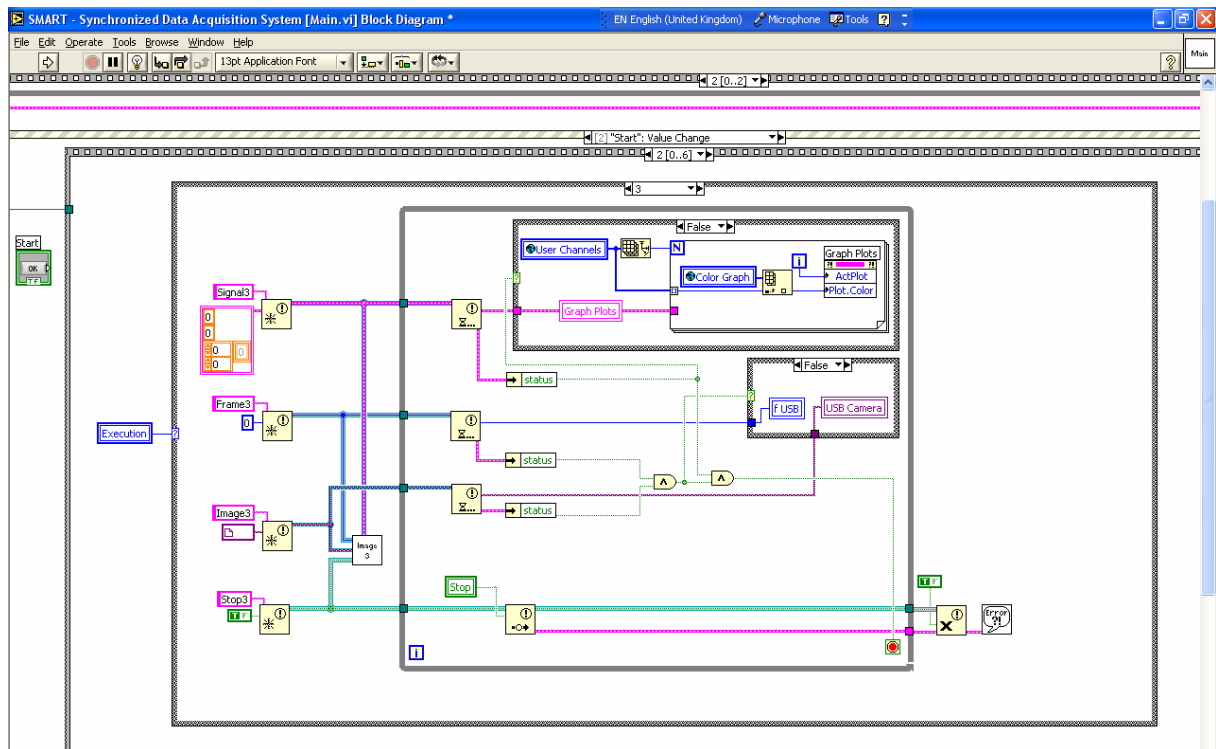


Figure 3.141: Illustration of the forth frame of the event “Start” that calls the function “Image3” and plots the acquired data

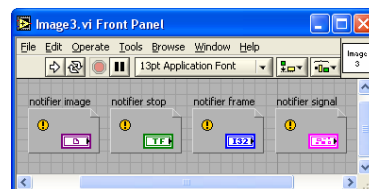


Figure 3.142: Illustration of the front panel of the function "Image3"

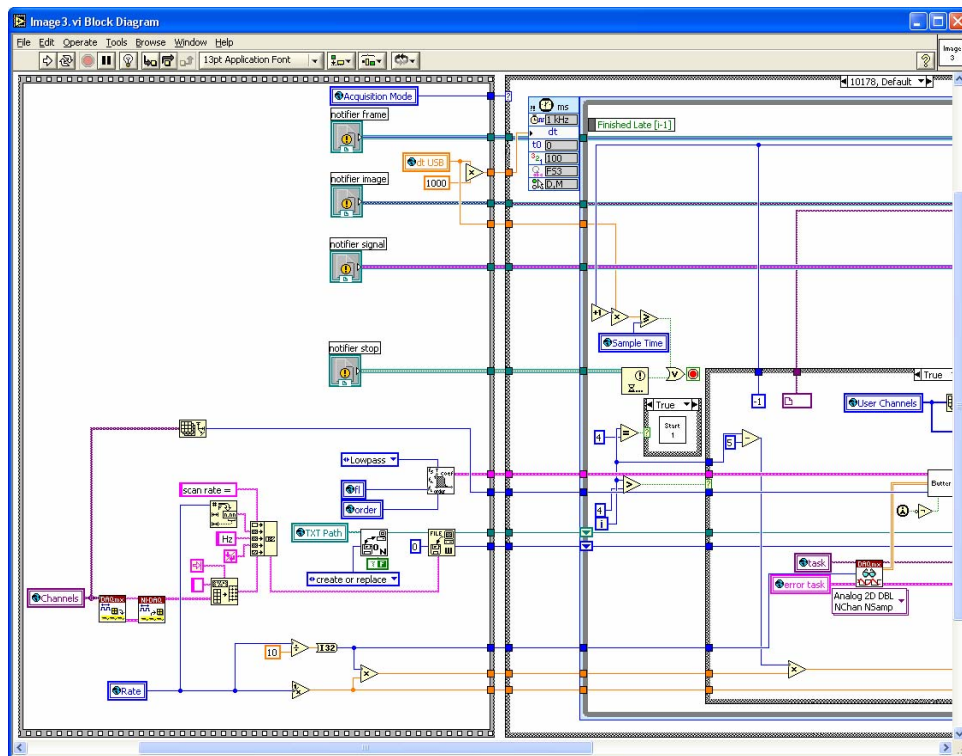


Figure 3.143: Illustration of the code of the function "Image3" that sets the inputs to be used by the functions during the data acquisition

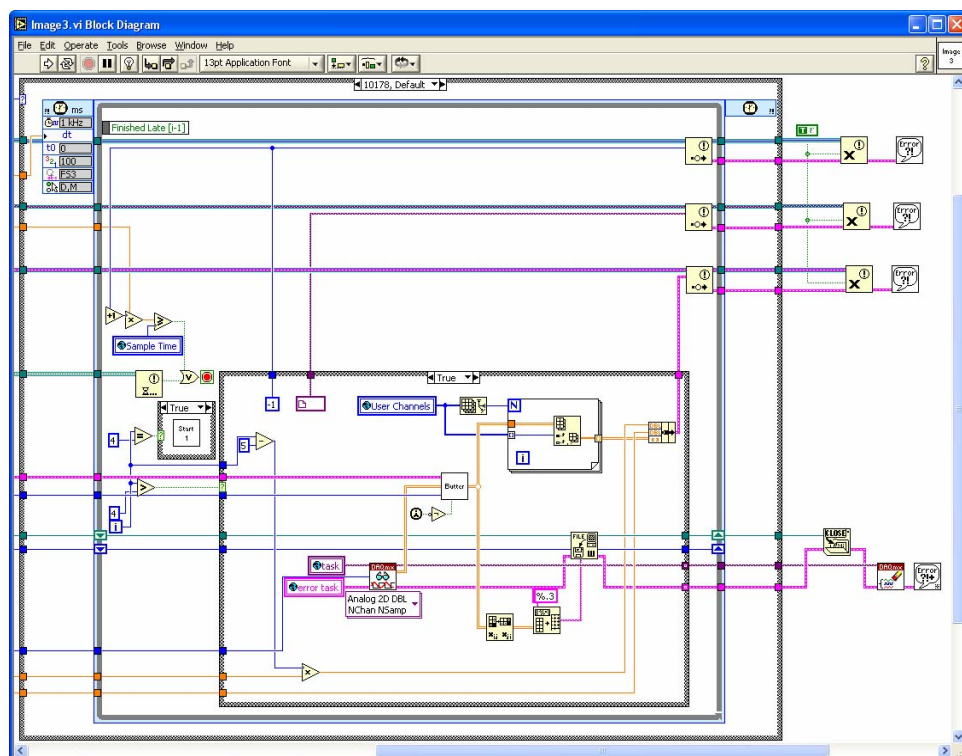


Figure 3.144: Illustration of the code of the function "Image3" that calls functions "Start1" and "Butterworth" and acquires data during a determined interval of time

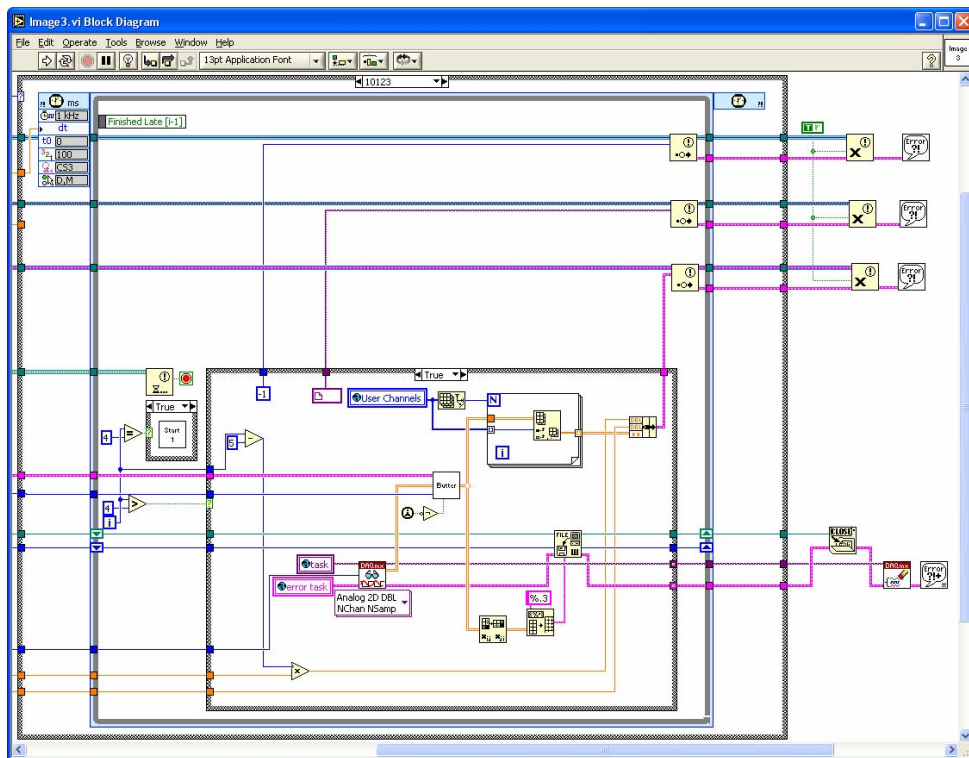


Figure 3.145: Illustration of the code of the function "Image3" that calls functions "Start1" and "Butterworth" and acquires data until the button Stop of the main window is pressed

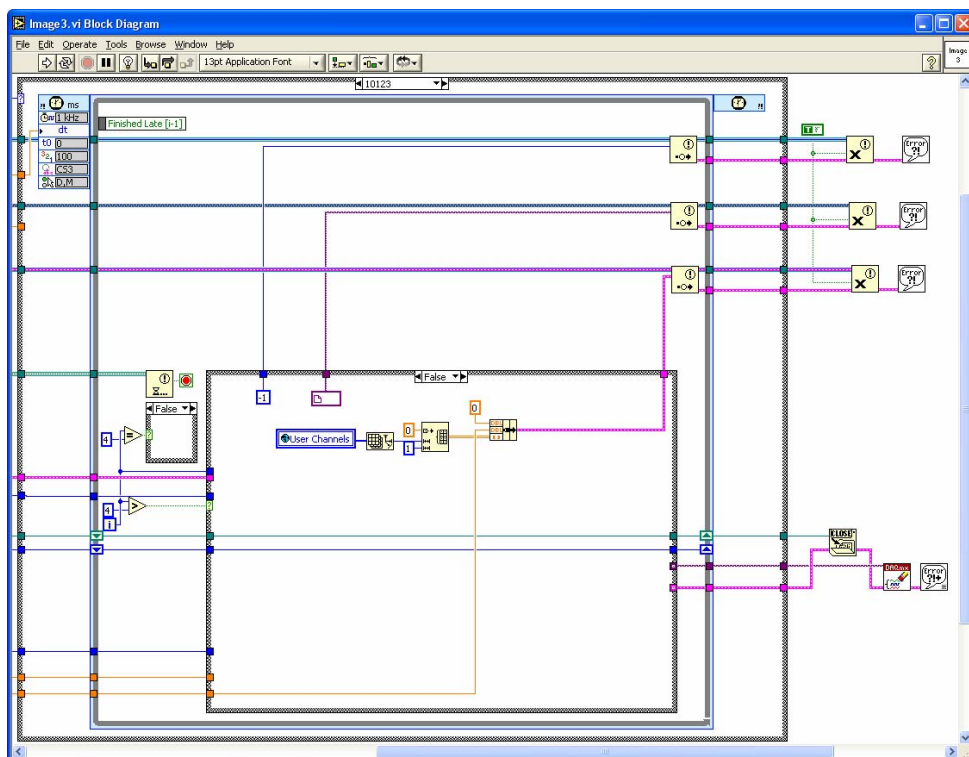


Figure 3.146: Illustration of the code of the function "Image3" that waits for the fifth USB camera frame to start the data acquisition

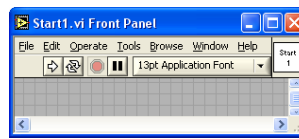


Figure 3.147: Illustration of the front panel of the function "Start1"

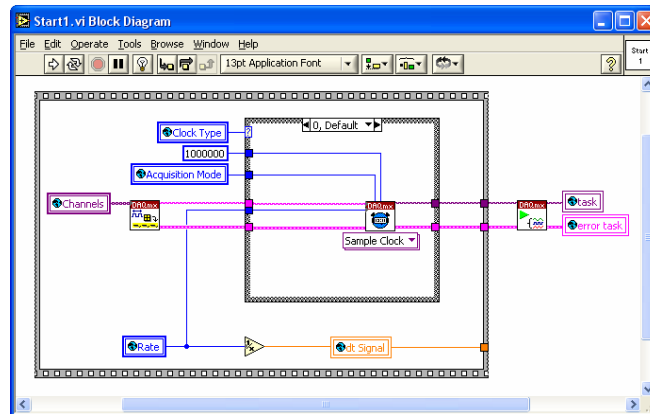


Figure 3.148: Illustration of the code of the function "Start1" that starts reading the inputs of the acquisition board (internal clock case)

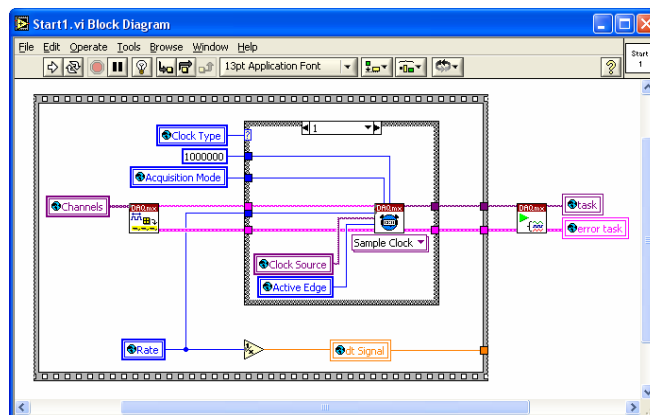


Figure 3.149: Illustration of the code of the function "Start1" that starts reading the inputs of the acquisition board (external clock case)

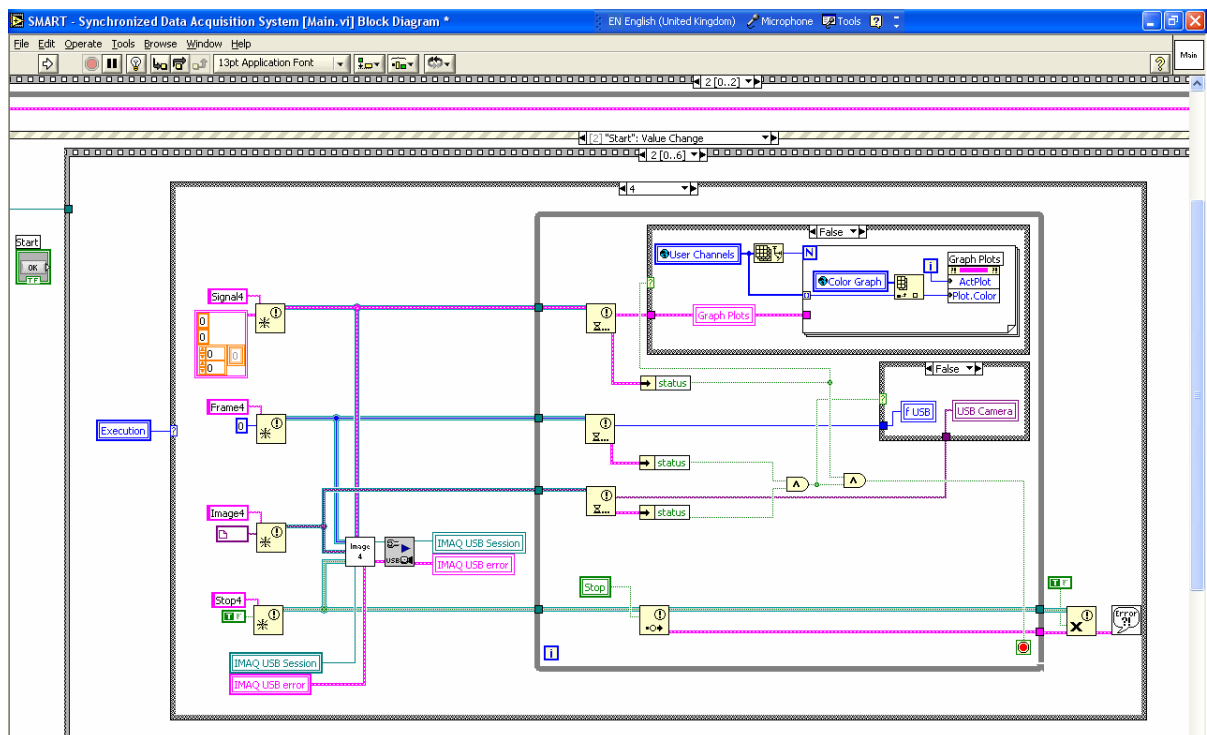


Figure 3.150: Illustration of the fifth frame of the event "Start" that calls the function "Image4" and plots the acquired data

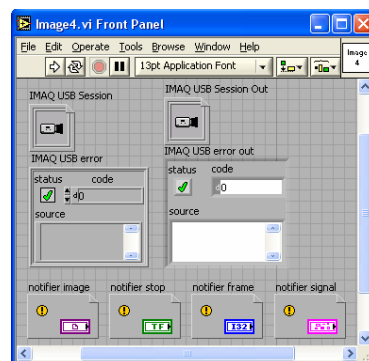


Figure 3.151: Illustration of the front panel of the function "Image4"

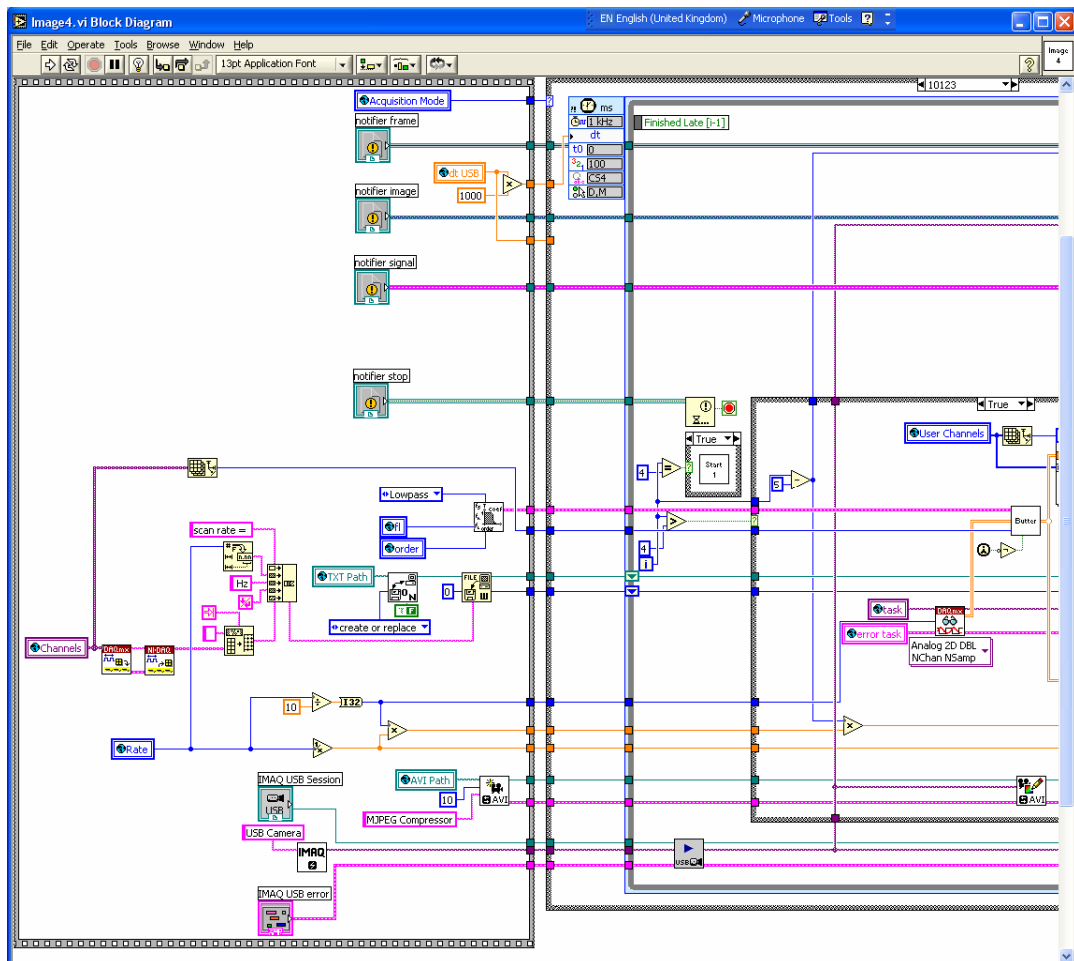


Figure 3.152: Illustration of the code of the function "Image4" that sets the inputs to be used by the functions during the data acquisition

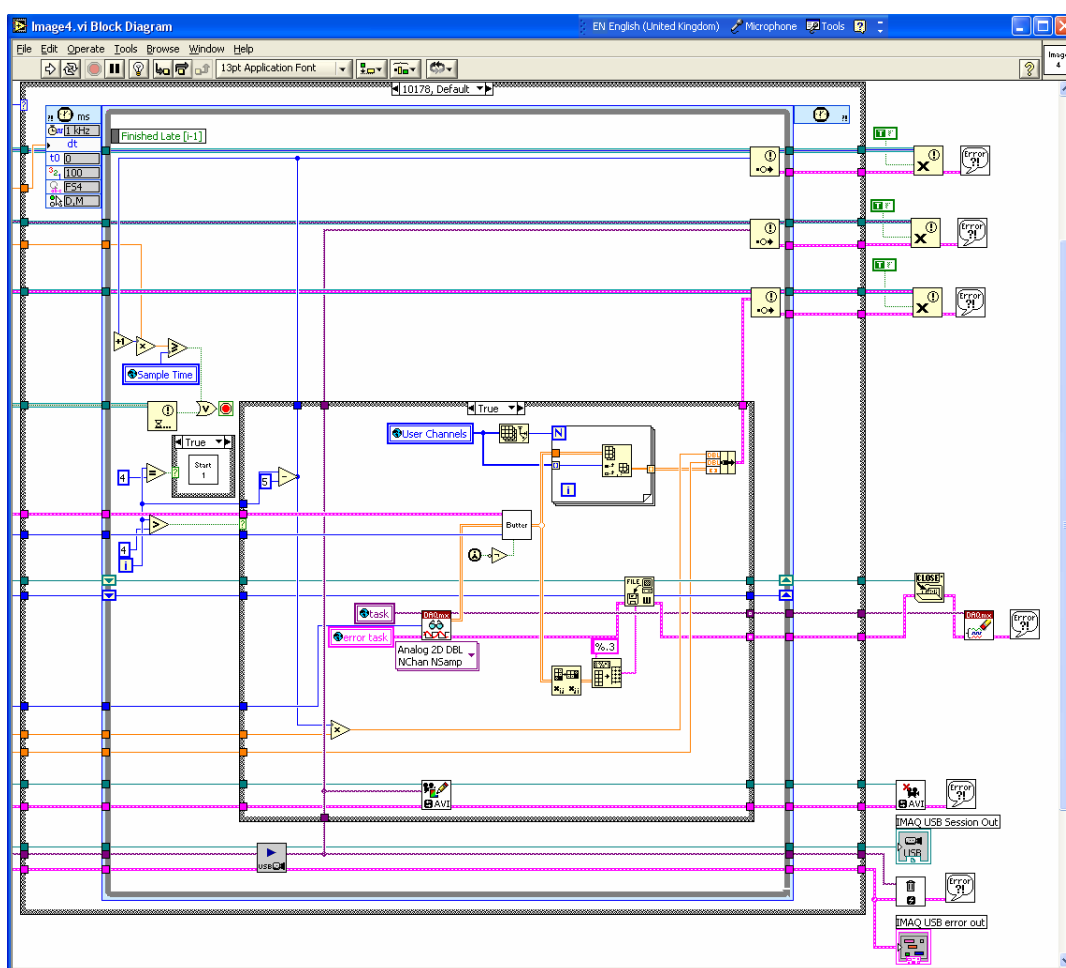


Figure 3.153: Illustration of the code of the function "Image4" that calls functions "Start1" and "Butterworth" and acquires data during a determined interval of time

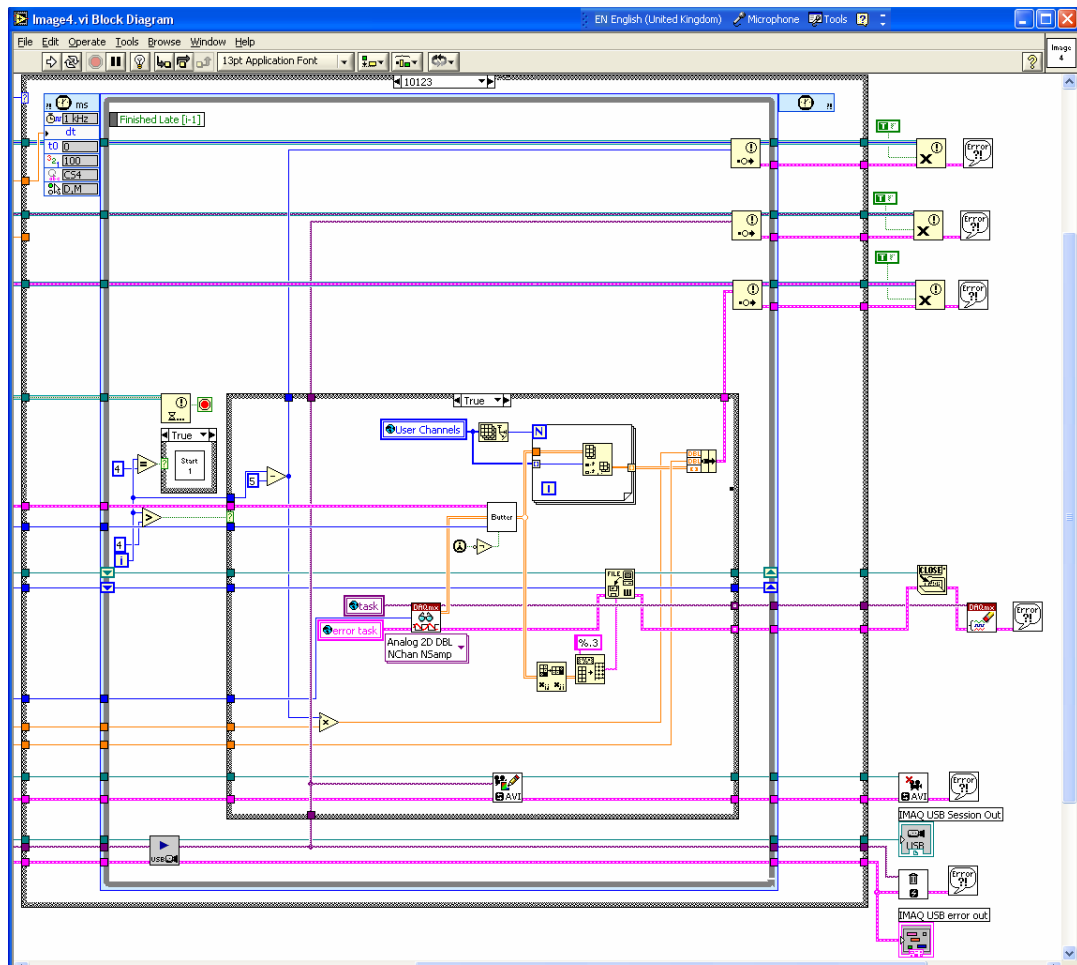


Figure 3.154: Illustration of the code of the function "Image4" that calls functions "Start1" and "Butterworth" and acquires data until the button Stop of the main window is pressed

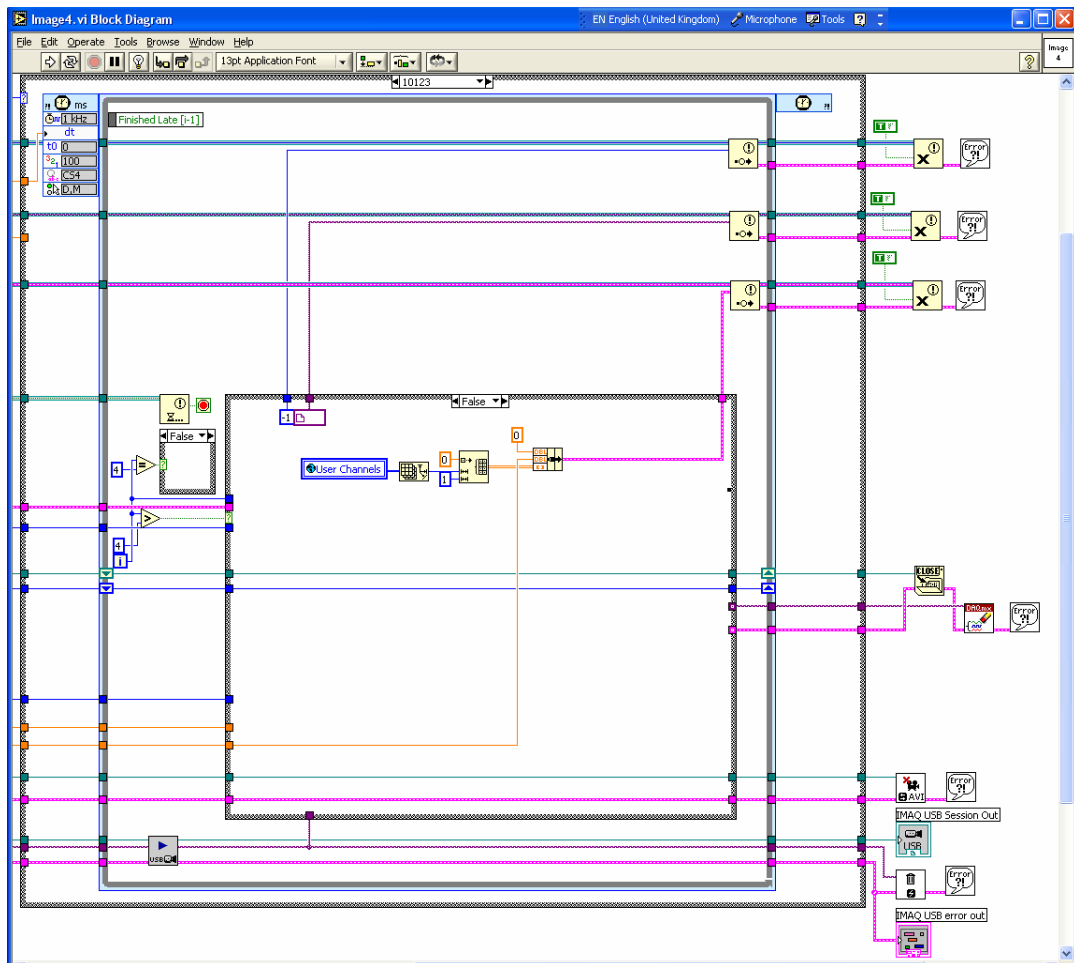


Figure 3.155: Illustration of the code of the function "Image4" that waits for the fifth USB camera frame to start the data acquisition

After doing the data acquisition, the button "Stop" is disabled. Then, the system is prepared to convert the acquired data, which is saved into a txt file, to an xls file. For this purpose, a dialog box asks if this conversion is desirable. In a positive case, a toolbar, on the main window, shows a progress bar indicating how the conversion goes. This procedure is shown by Figure 3.156. If any xls file already exists it is deleted by function "Delete", as Figure 3.157 shows. Eventually the conversion starts, as Figure 3.158 illustrates. At the end, the items of the menu and the other objects on the window "Main" are set with the previous values, as shown by Figure 3.159.

After acquiring the signals and images and opening the correlated files, a frame-by-frame analysis can be done. Pressing the button "forward HS", it is added 1 frame to the variable "f HS", as Figure 3.160 shows. If this value is greater than the quantity of acquired frames, nothing happens. Otherwise, the next HS camera frame is shown as well as the correspondent signal buffer and USB camera frame.



Figure 3.160: Illustration of the code of the event “forward HS” that adds 1 frame to the variable “f HS” (case -1)

The variables “dF HS Signal” and “dF HS USB” are the quantity of samples and USB camera frames, respectively, that have to be considered for each HS camera frame. The variables “P HS Signal” and “P HS USB” decide how to calculate which signal buffer and USB camera frame have to be plotted, respectively. If these variables are equal to -1 (Figure 3.160), nothing happens. Otherwise, if they are equal to 0 (Figure 3.161), it means that the sampling rate of the HS camera frames is less than the sampling rate of the signals and the USB camera images. In this case, the variable “f HS” is divided by variables “dF HS Signal” and “dF HS USB”. If they are equal to 1 (Figure 3.164), the sampling rate of the HS camera frames is greater than the sampling rate of the signals and the USB camera images. In this case, the variable “f HS” is multiplied by the variables “dF HS Signal” and “dF HS USB”.

The variable “DSignal” is always equal to zero, because the signals samples are not delayed. Function “Camera USB” (Figure 3.162) is responsible for plotting the calculated USB camera frame if it is equal or greater than zero and less than the quantity of acquired frames, as Figure 3.163 shows. Eventually, function “Camera HS” (Figure 3.166 and Figure 3.167) plots the solicited HS camera frame, as shown by Figure 3.165.

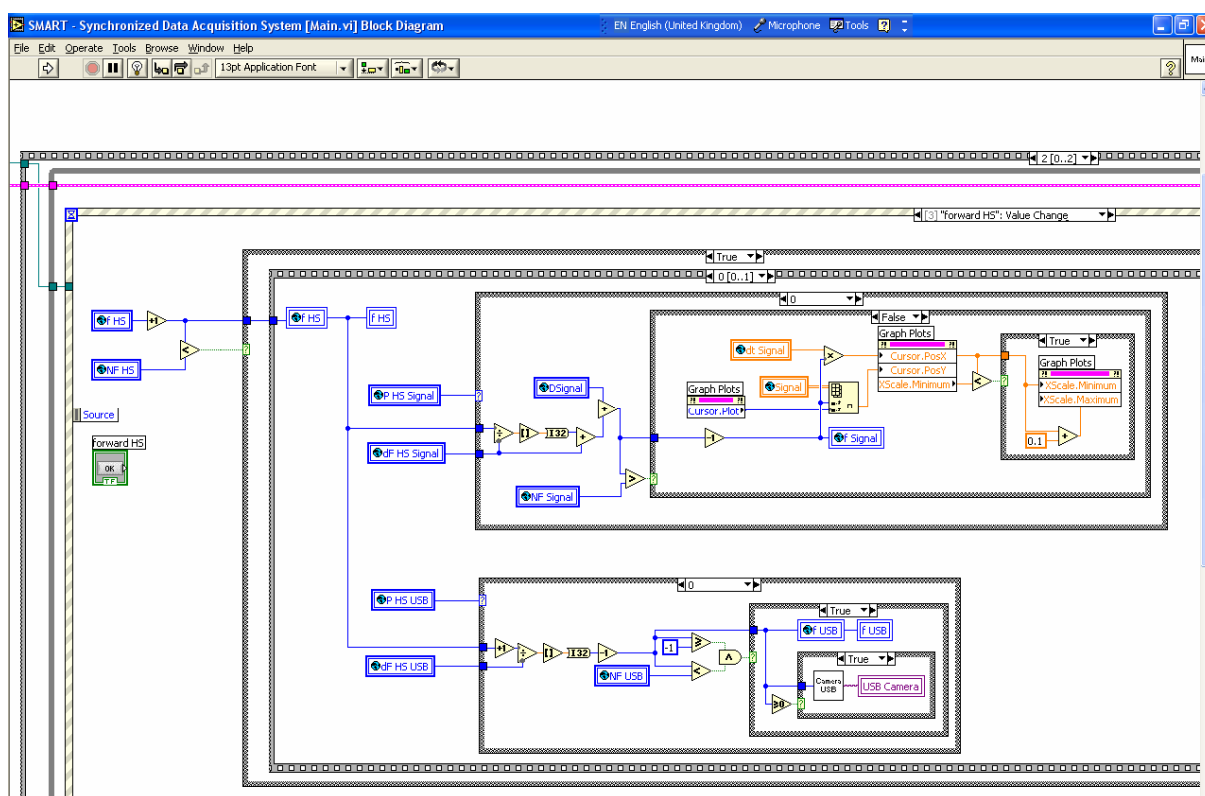


Figure 3.161: Illustration of the code of the event “forward HS” that adds 1 frame to the variable “f HS” (case 0)



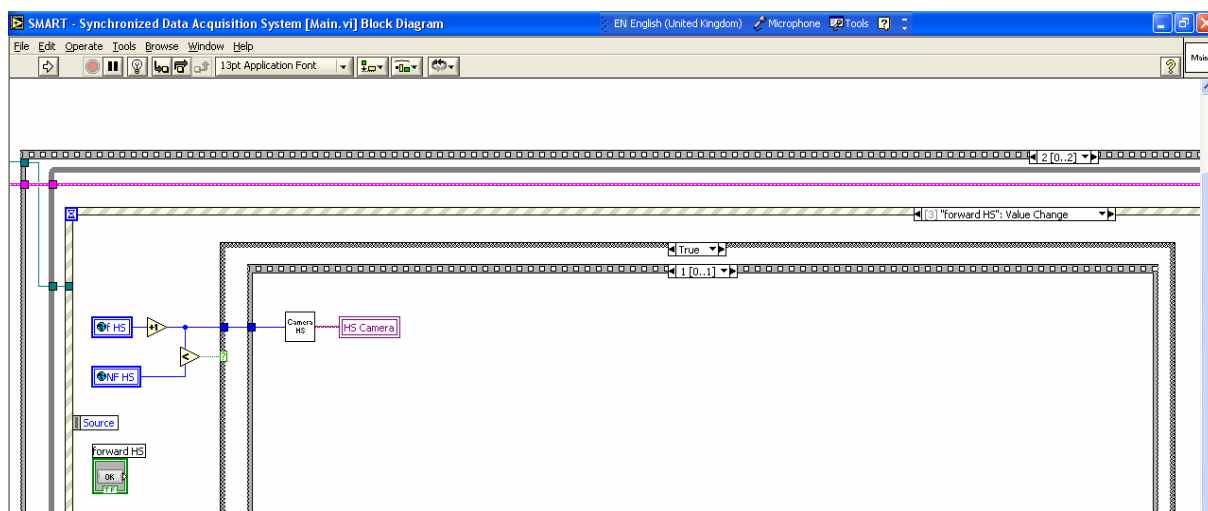


Figure 3.165: Illustration of the code of the event “forward HS” that calls the function “Camera HS”

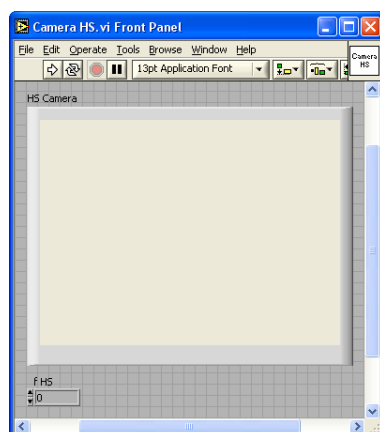


Figure 3.166: Illustration of the front panel of the function "Camera HS"

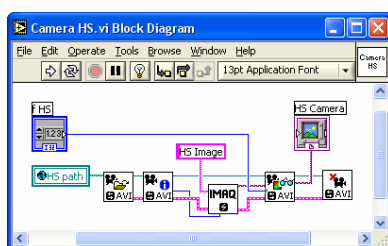


Figure 3.167: Illustration of the code of the function "Camera HS" that plots a HS camera frame

Pressing the button “backward HS”, it is decreased 1 frame from the variable “f HS”. If this value is equal or less than -1, nothing happens. Otherwise, the next HS camera frame is shown as well as the correspondent signal buffer and USB camera frame. For this purpose,

the same procedure previously explained is executed, as illustrated by Figure 3.168 and Figure 3.169.

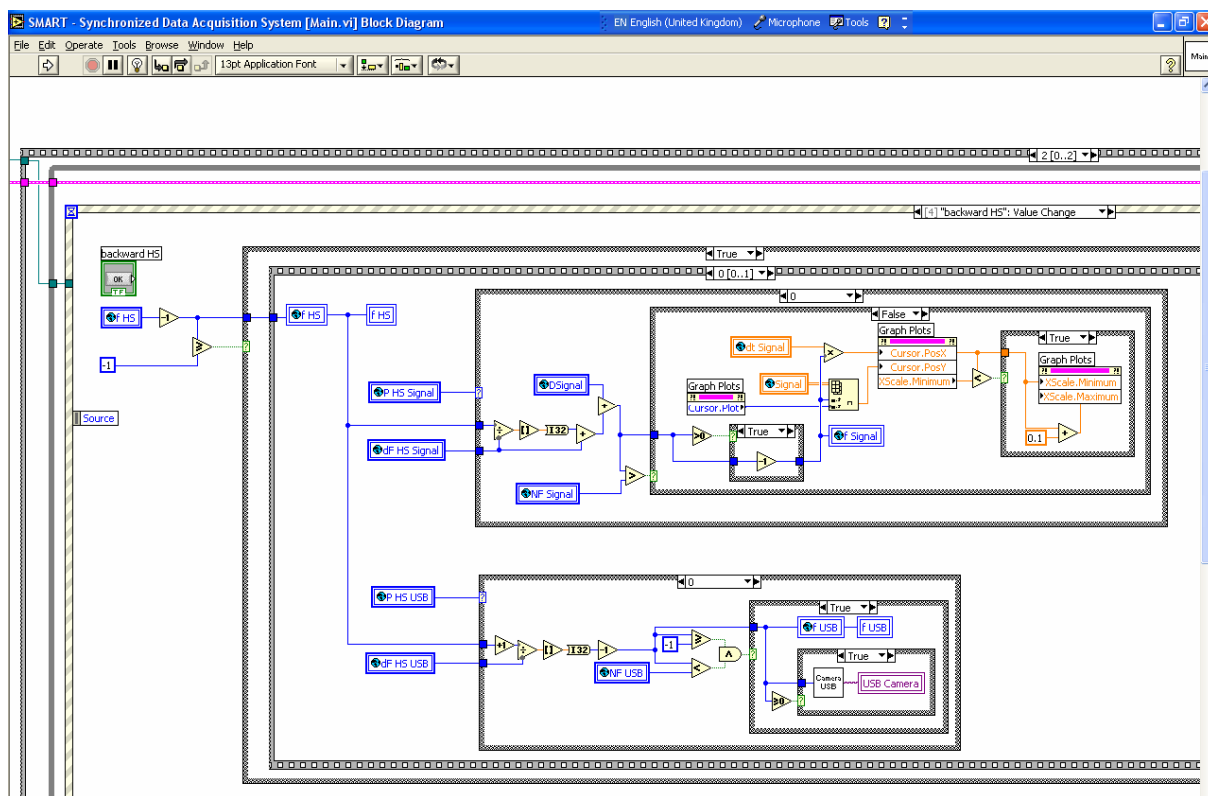


Figure 3.168: Illustration of the code of the event “backward HS” that decreases 1 frame of the variable “f HS” (case 0)

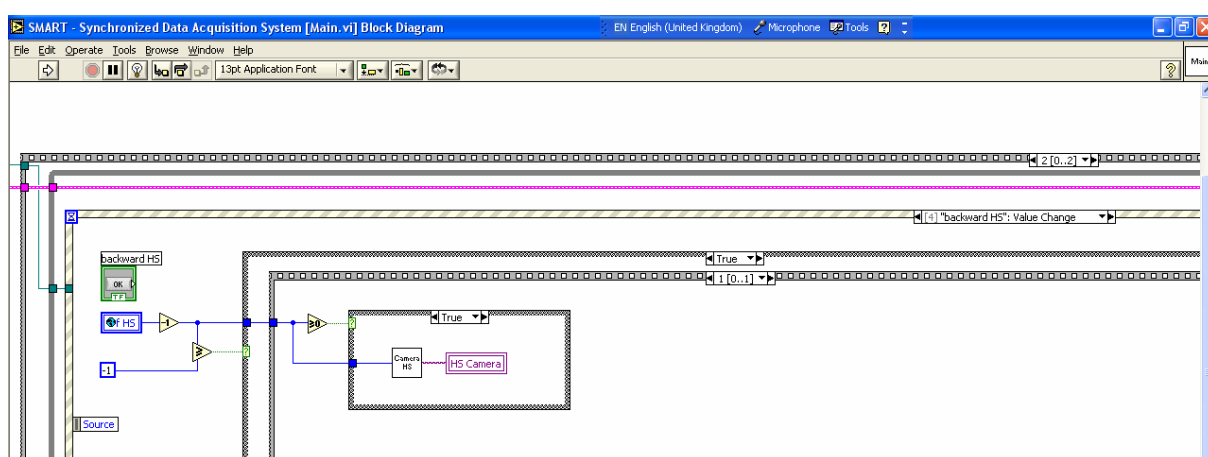


Figure 3.169: Illustration of the code of the event “backward HS” that calls the function “Camera HS”

Pressing the button “forward USB”, it is added 1 frame to the variable “f USB”. If this value is greater than the quantity of acquired frames, nothing happens. Otherwise, the next

USB camera frame is shown as well as the correspondent signal buffer and HS camera frame.

The variables “dF USB Signal” and “dF USB HS” are the quantity of samples and HS camera frames, respectively, that have to be considered for each USB camera frame. The variables “P USB Signal” and “P USB HS” decide how to calculate which signal buffer and HS camera frame have to be plotted, respectively. If these variables are equal to -1, nothing happens. Otherwise, if they are equal to 0 (Figure 3.170), the sampling rate of the USB camera images is less than the sampling rate of the signals and HS camera images. In this case, the variable “f USB” is divided by variables “dF USB Signal” and “dF USB HS”. If they are equal to 1 (Figure 3.171), the sampling rate of the USB camera images is greater than the sampling rate of the signals and HS camera images. In this case, the variable “f USB” is multiplied by variables “dF USB Signal” and “dF USB HS”.

Variable “DHS” represents the delay mentioned before, in terms of frames, and it is used for shifting the HS camera frames. Function “Camera HS” is responsible for plotting the calculated HS camera frame, if it is equal or greater than zero and less than the quantity of acquired frames. Eventually, function “Camera USB” plots the USB camera frame.

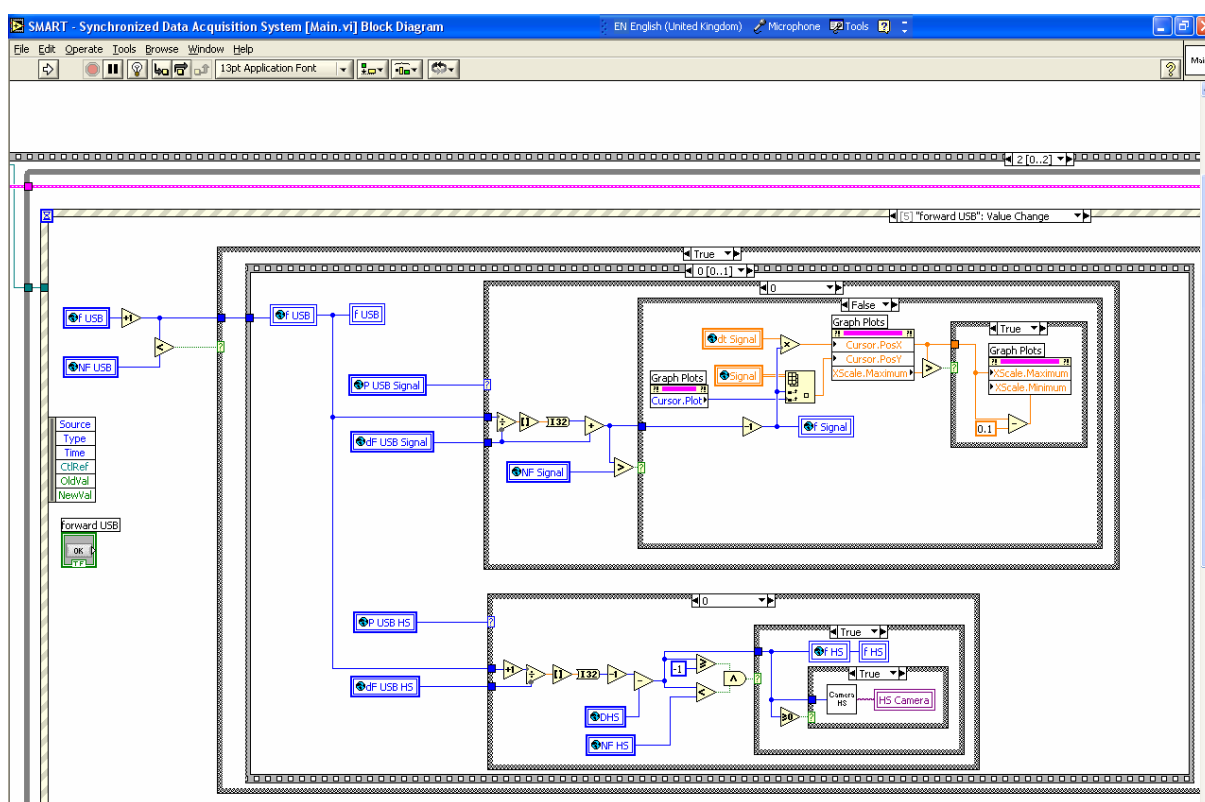


Figure 3.170: Illustration of the code of the event “forward USB” that adds 1 frame to the variable “f USB” (case 0)

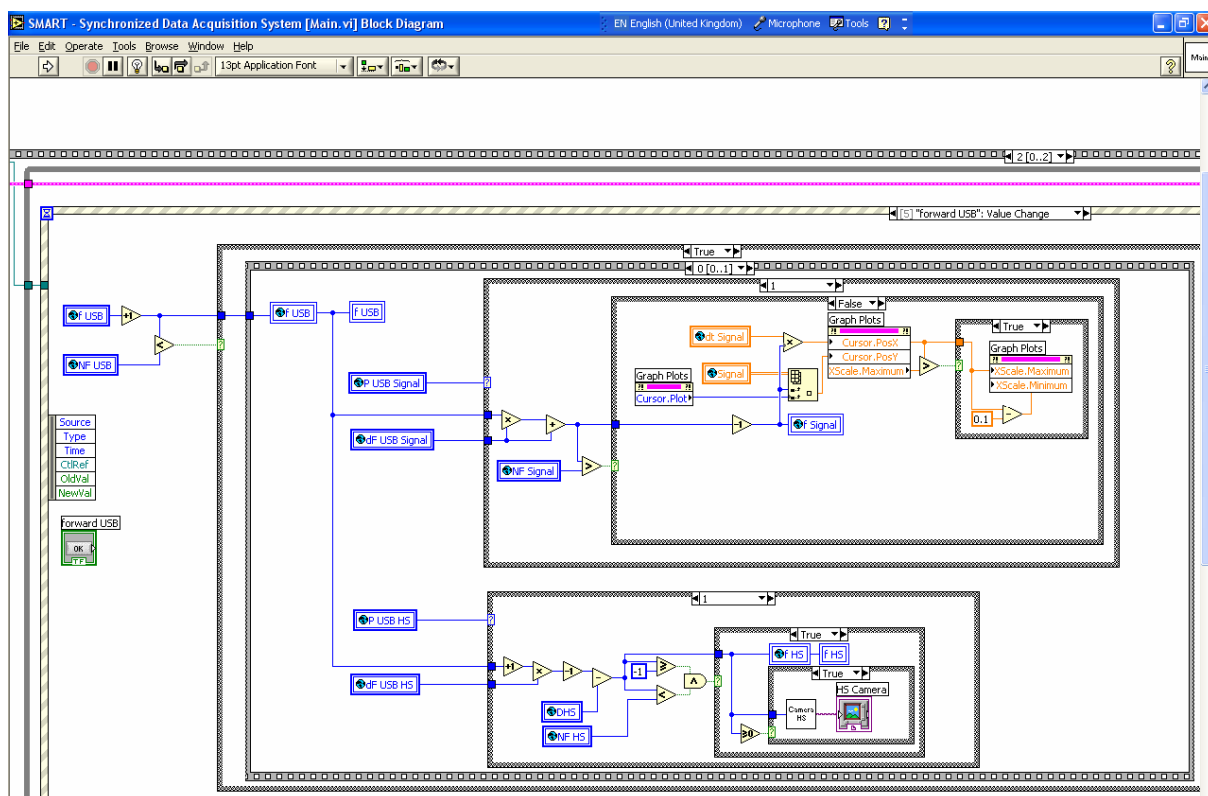


Figure 3.171: Illustration of the code of the event “forward USB” that adds 1 frame to the variable “f USB” (case 1)

Pressing the button “backward USB”, it is decreased 1 frame from the variable “f HS”. If the result is equal or less than -1, nothing happens. Otherwise, the next USB camera frame is shown as well as the correspondent signal buffer and HS camera frame. For this purpose, the same procedure previously explained is executed, as illustrated by Figure 3.172.

Typing the desired USB frame in the box “f USB”, it is checked if the typed value is into the acceptable limits. In a negative case, the previous value is shown in the box “f USB”, as Figure 3.173 illustrates. Otherwise, it is executed the same procedure explained when the button “forward USB” is pressed, as shown by Figure 3.174.

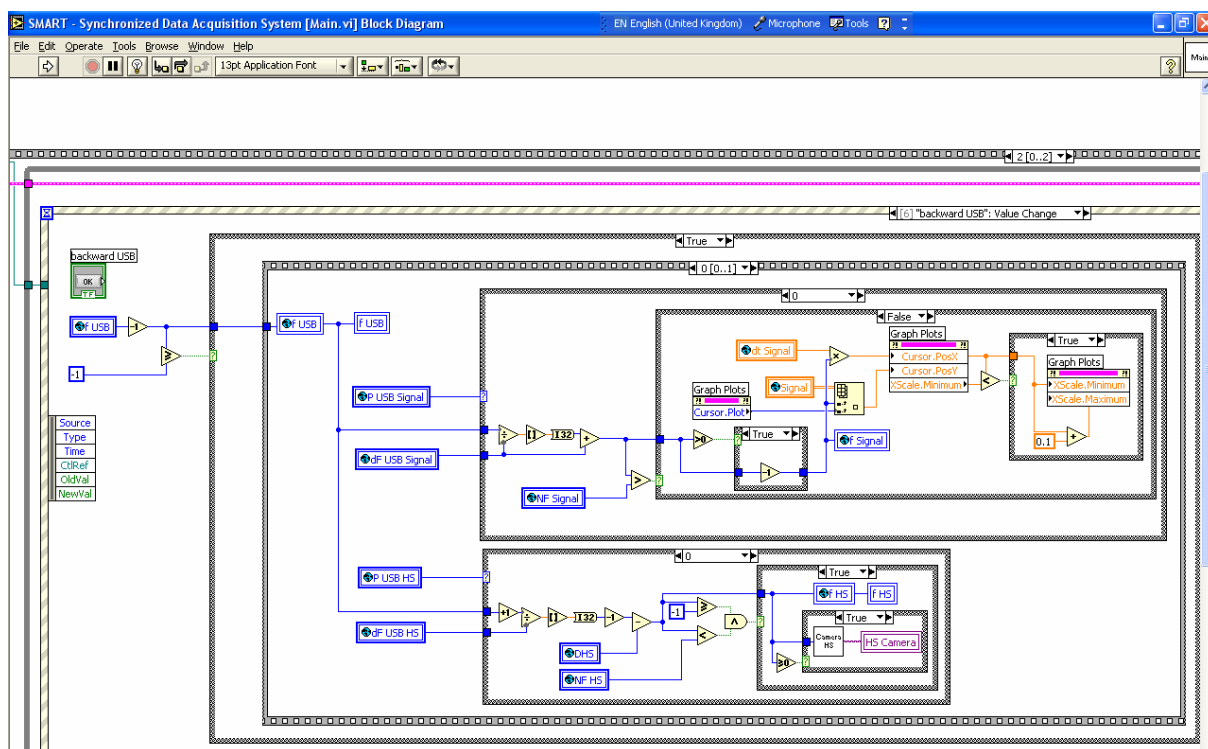


Figure 3.172: Illustration of the code of the event “backward USB” that decreases 1 frame of the variable “f USB” (case 0)



Figure 3.173: Illustration of the code of the event “f USB” that keeps the same USB camera frame if the typed frame is not under the acceptable limits