



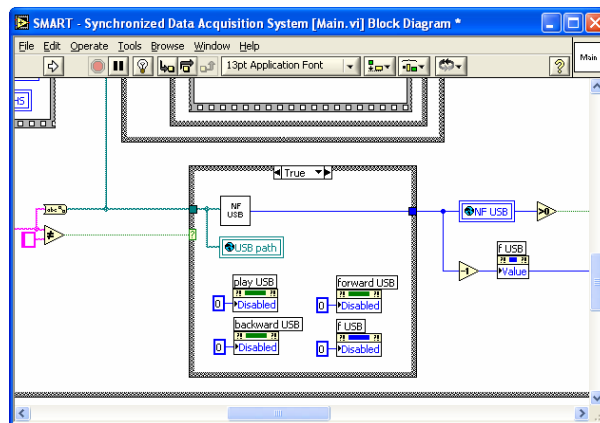


Figure 3.44: Illustration of the code that calls the function “NF USB”

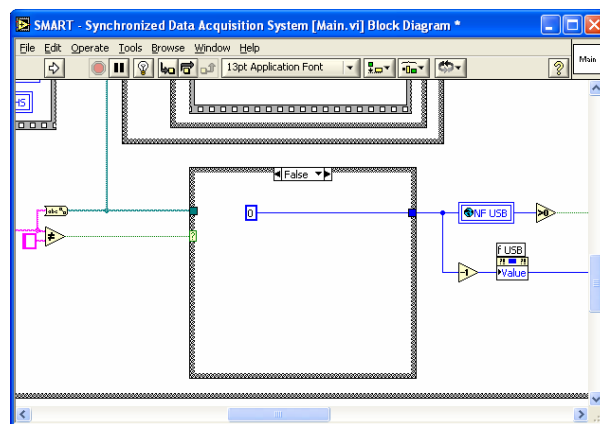


Figure 3.45: Illustration of the code that sets a default value for the variable “NF USB”

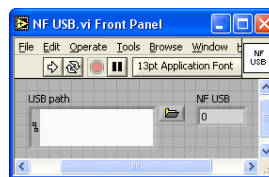


Figure 3.46: Illustration of the code of the function “NF USB”

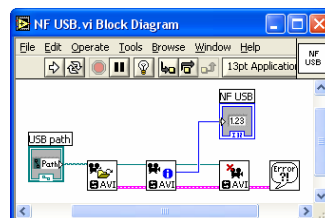


Figure 3.47: Illustration of the code of the function “NF USB” that reads the number of frames of the USB camera file

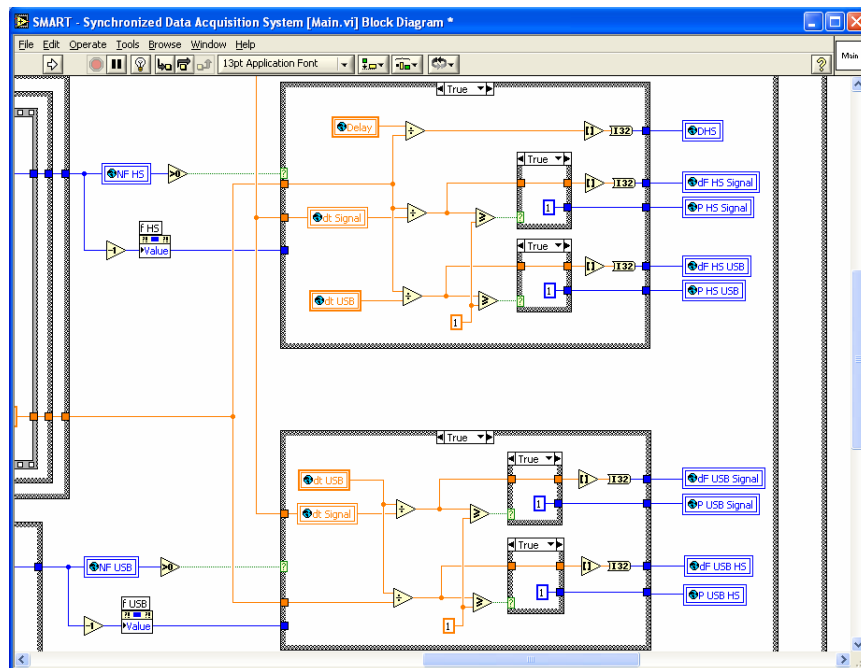


Figure 3.48: Illustration of the code that calculates the ratios among the frames of each camera and the signal samples

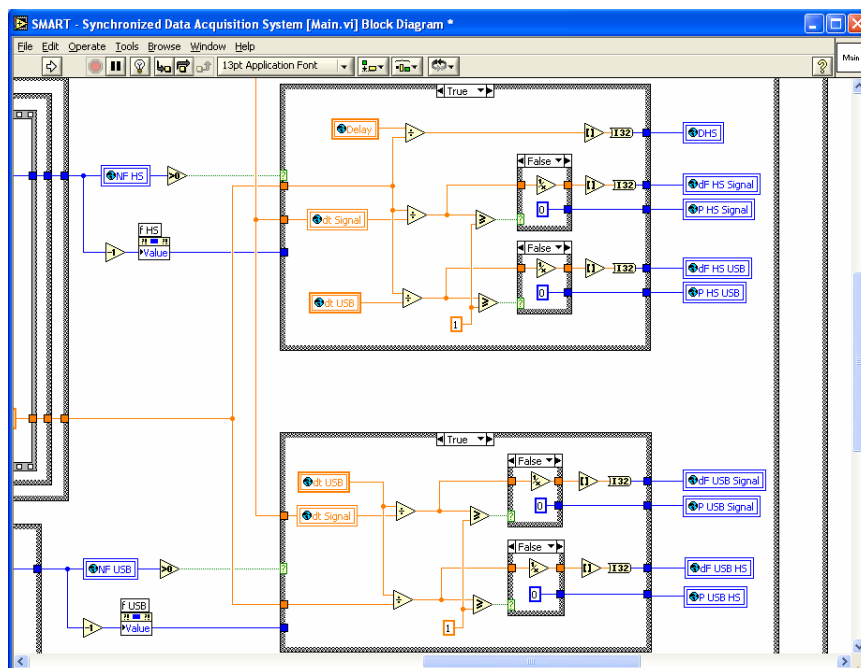


Figure 3.49: Illustration of the code that calculates the inverse ratios among the frames of each camera and the signal samples

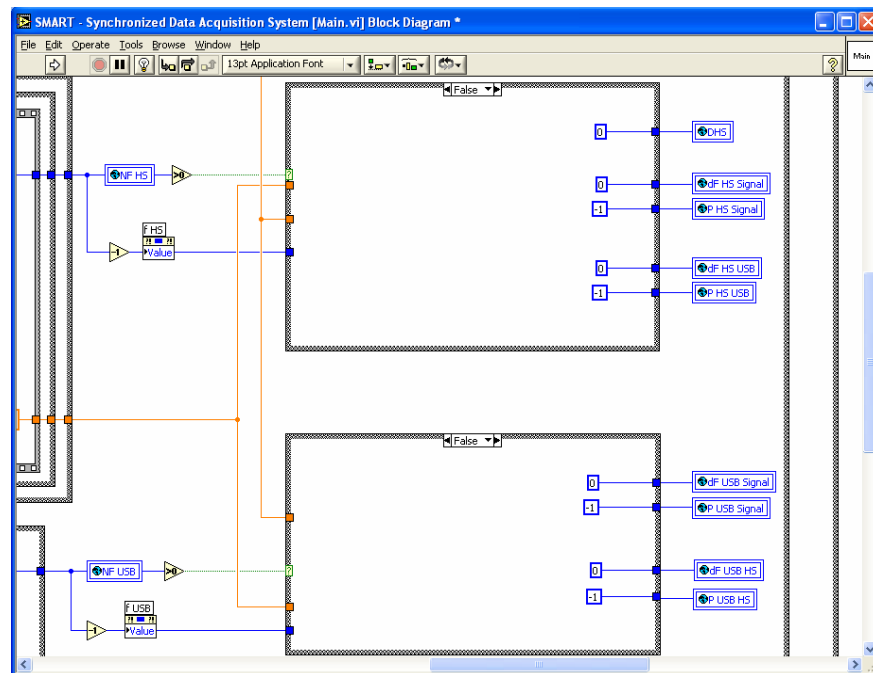


Figure 3.50: Illustration of the code that sets the variables to be used during the frame-by-frame analysis if no camera file is opened

The next “Menu Selection” frame is “Save as xls...”. This item converts a txt file into a xls file. First, the message “Saving .xls File...” appears on the toolbar of the main window and the progress bar is activated, as Figure 3.51 shows. Then, the function “Save XLS” (Figure 3.56, Figure 3.57 and Figure 3.58) asks the txt file to be converted and the function “Read txt File” (Figure 3.59, Figure 3.60 and Figure 3.61) creates an xls path. If it is not cancelled, the progress bar is set to zero, as Figure 3.52 shows. If a file with the created xls path already exists, it is deleted by function “Delete” (Figure 3.62 and Figure 3.63), as Figure 3.53 illustrates.

Afterwards, the conversion is done by function “xls” (Figure 3.64 and Figure 3.65), as Figure 3.54 shows. First, Excel is opened by function “Open Excel and Make Visible” (Figure 3.66 and Figure 3.67). Then, a workbook and a worksheet are created, by functions “Open Specific Workbook” (Figure 3.68 and Figure 3.69) and “Open Specific Worksheet” (Figure 3.70 and Figure 3.71), respectively. After that, function “Row Col To Range Format” (Figure 3.75 and Figure 3.76) converts the column and the row, which are numeric, into a range format of excel, that identifies the cell where a value is placed. Function “Set Cell Value with Range” (Figure 3.72, Figure 3.73 and Figure 3.74) places the values into the cells.

Eventually, the toolbar is cleaned and the progress bar is inactivated, as Figure 3.55 shows.

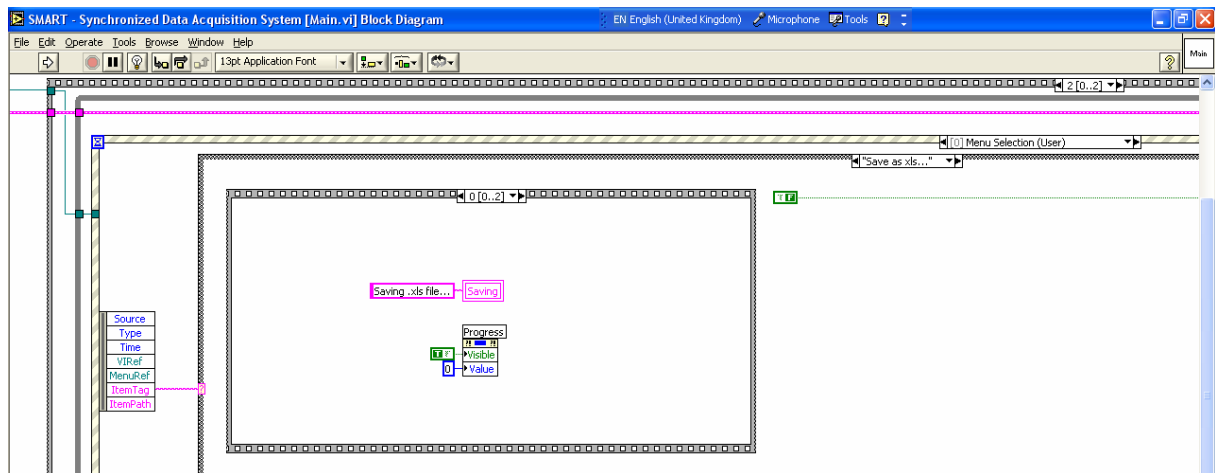


Figure 3.51: Illustration of the code that shows a message and a progress bar on the toolbar of the main window, before starting to convert a txt file into a xls file

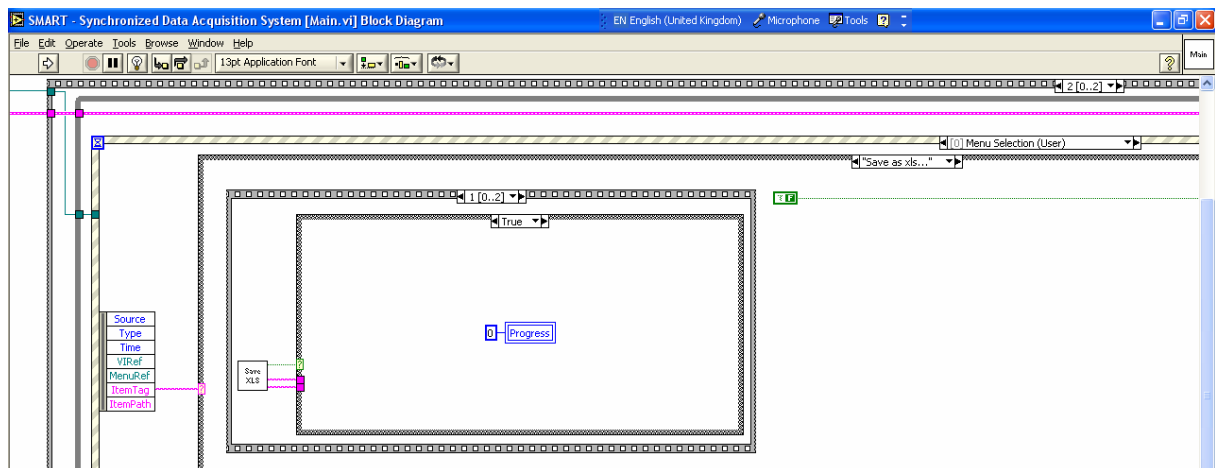


Figure 3.52: Illustration of the code that sets a progress bar on the toolbar of the main window

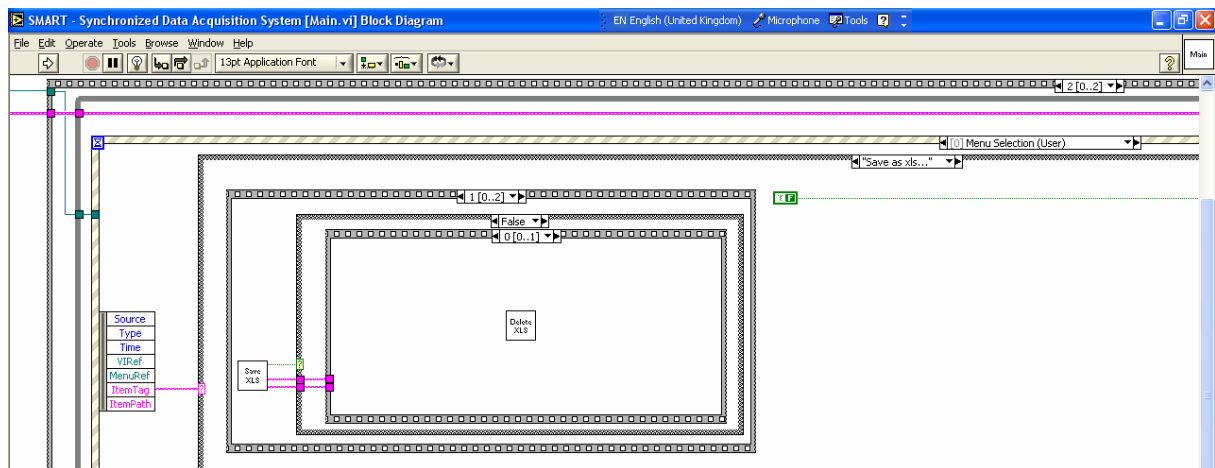


Figure 3.53: Illustration of the code that calls the function “Delete XLS”

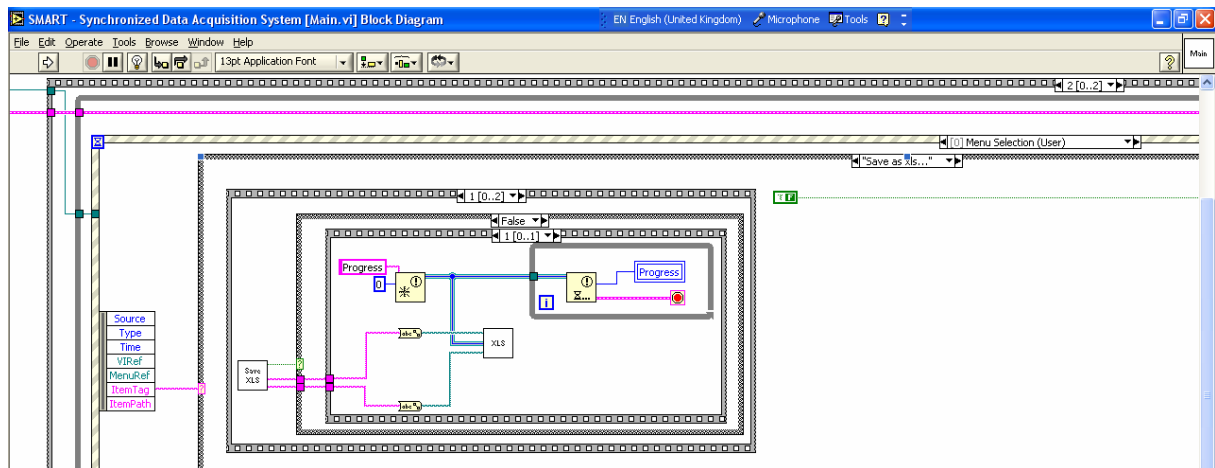


Figure 3.54: Illustration of the code that calls the function “XLS”

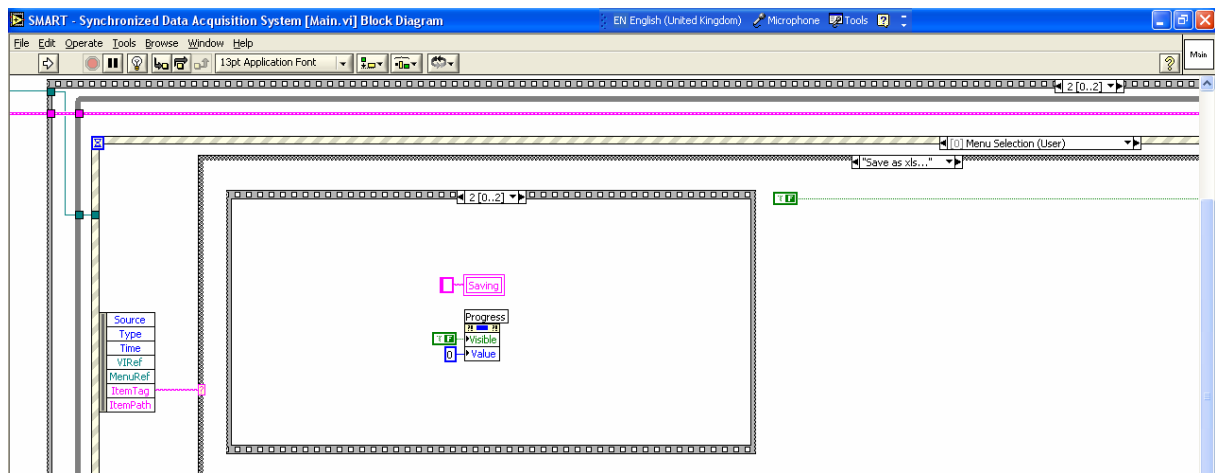


Figure 3.55: Illustration of the code that inactivates the progress bar and cleans the toolbar of the main window

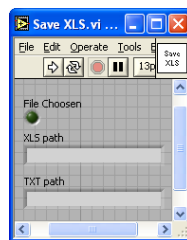


Figure 3.56: Illustration of the front panel of the function "Save XLS"

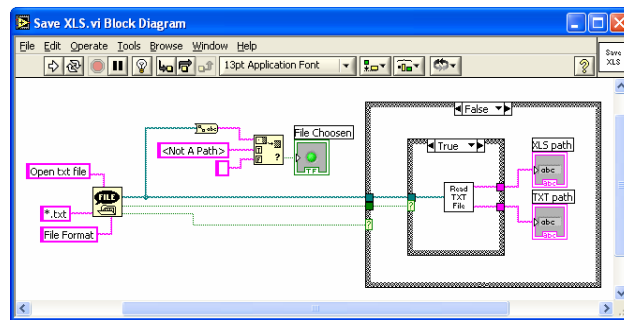


Figure 3.57: Illustration of the code of the function “Save XLS” that asks the txt file to be converted and calls the function “Read txt File”

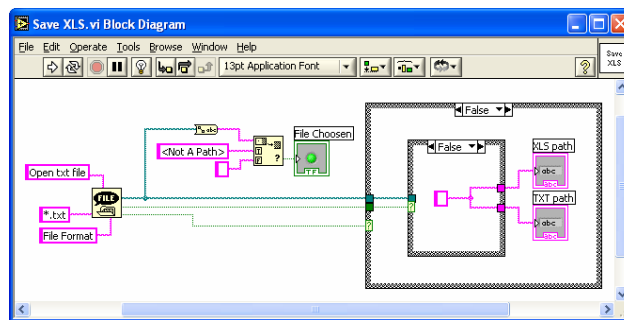


Figure 3.58: Illustration of the code of the function “Save XLS” that returns an empty string for the xls and txt files, case no txt file is opened

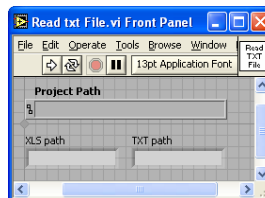


Figure 3.59: Illustration of the front panel of the function "Read txt File"

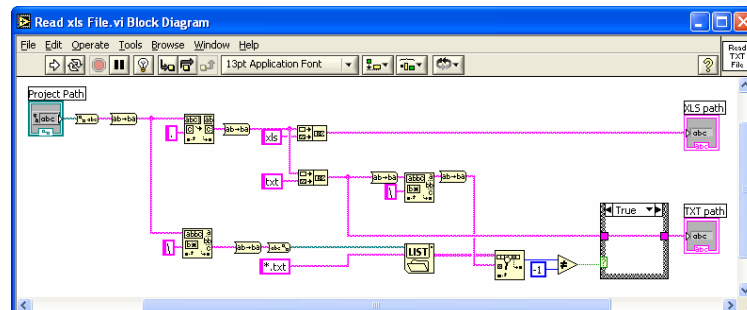


Figure 3.60: Illustration of the code of the function “Read txt File” that returns the paths of the xls and txt files

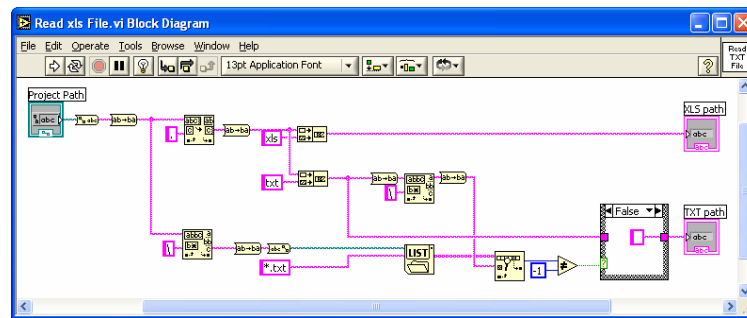


Figure 3.61: Illustration of the code of the function “Read txt File” that returns an empty string if no any txt file exists

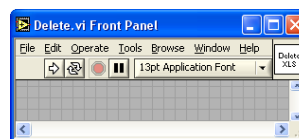


Figure 3.62: Illustration of the front panel of the function "Delete"

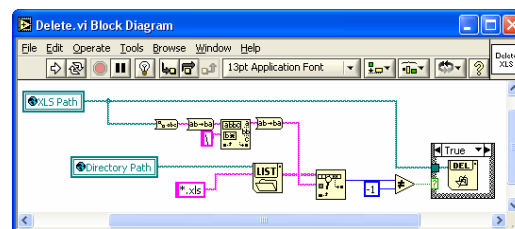


Figure 3.63: Illustration of the code of the function “Delete” that deletes an existent xls file

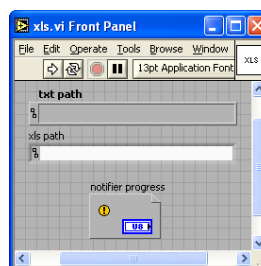


Figure 3.64: Illustration of the front panel of the function "xls"

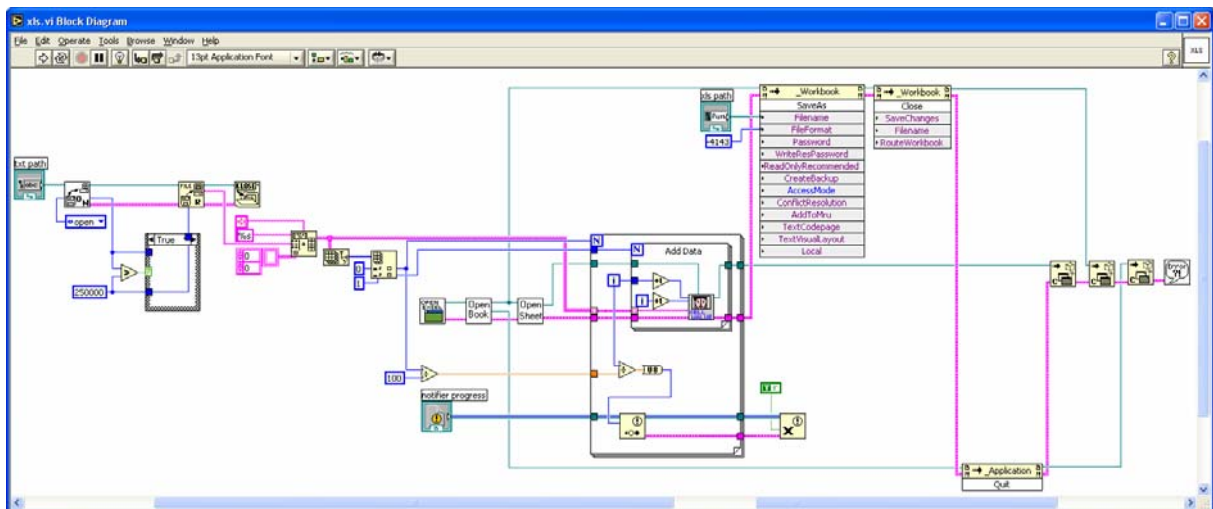


Figure 3.65: Illustration of the code of the function "xls" that converts a txt file into a xls file

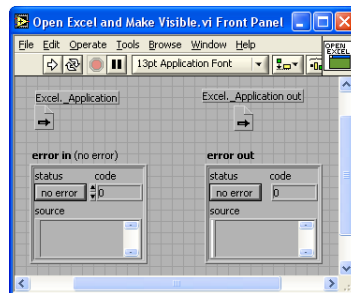


Figure 3.66: Illustration of the front panel of the function "Open Excel and Make Visible"

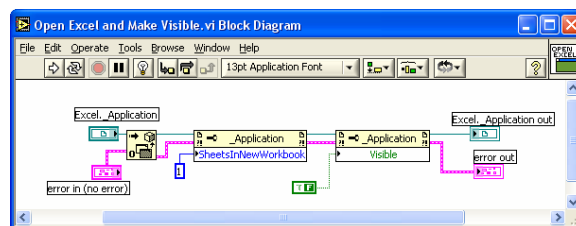


Figure 3.67: Illustration of the code of the function "Open Excel and Make Visible" that opens an excel application

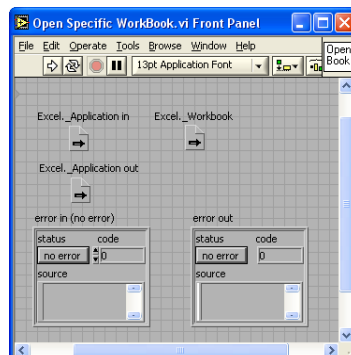


Figure 3.68: Illustration of the front panel of the function "Open Specific Workbook"

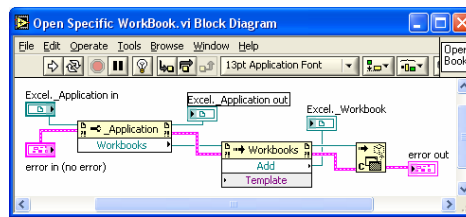


Figure 3.69 Illustration of the code of the function "Open Specific WorkBook" that opens a specific workbook of an excel application

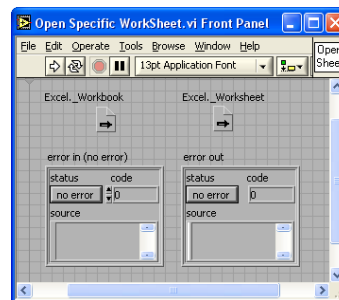


Figure 3.70: Illustration of the front panel of the function "Open Specific WorkSheet"

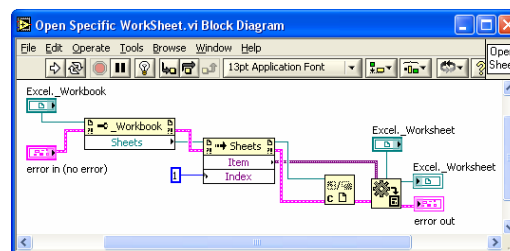


Figure 3.71: Illustration of the code of the function "Open Specific WorkSheet" that opens a specific worksheet of a workbook of an excel application

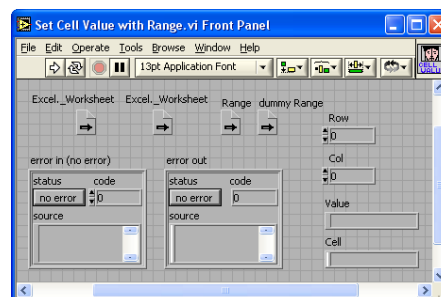


Figure 3.72: Illustration of the front panel of the function "Set Cell Value with Range"

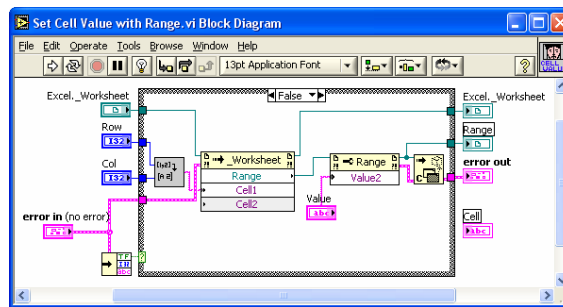


Figure 3.73: Illustration of the code of the function "Set Cell Value with Range" that writes a value into a excel cell

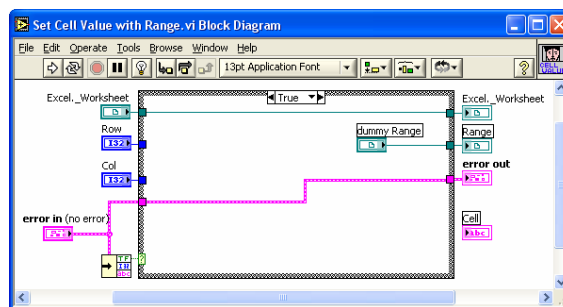


Figure 3.74: Illustration of the code of the function "Set Cell Value with Range" that does not write a value into an excel cell, if an error occurs

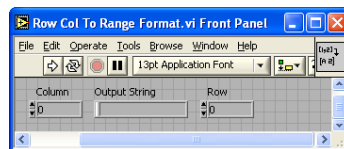


Figure 3.75: Illustration of the front panel of the function "Row Col To Range Format"

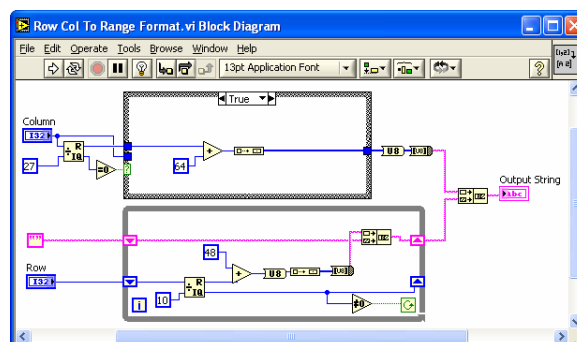


Figure 3.76: Illustration of the code of the function "Row Col To Range Format" that converts the column and the row, which are numeric, into a range format of excel

Figure 3.77 illustrates the item “Exit” of the “Menu Selection”. If the variable “Camera” is true, means that an USB camera session was previously opened and must be closed. Otherwise, this is not necessary. Then, the software is quit.

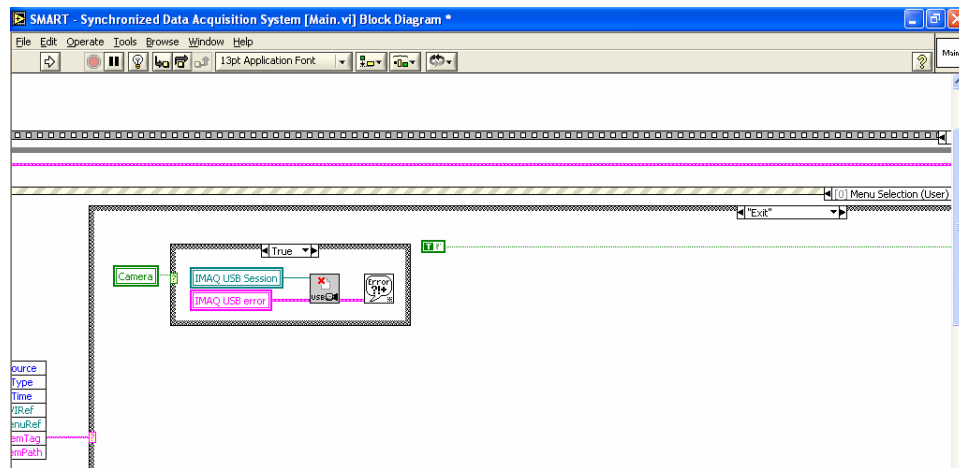


Figure 3.77: Illustration of the code that shows the item “Exit...” of the “Menu Selection” that closes an USB camera session and quits the software

The item “Setup Signal...” of the “Menu Selection” is shown by Figure 3.78. The window “Setup Signal” (Figure 3.79) shows up for setting the acquisition parameters: Channel, Clock Settings, Acquisition Mode and Advanced Clock Settings (actually, these ones are not used by the software). If the button OK is pressed, the global variables are upgraded. If the button Cancel is pressed, it does not happen. Pressing any of these buttons, this window is closed. Figure 3.80, Figure 3.81, Figure 3.82, Figure 3.83 and Figure 3.84 show how it works.

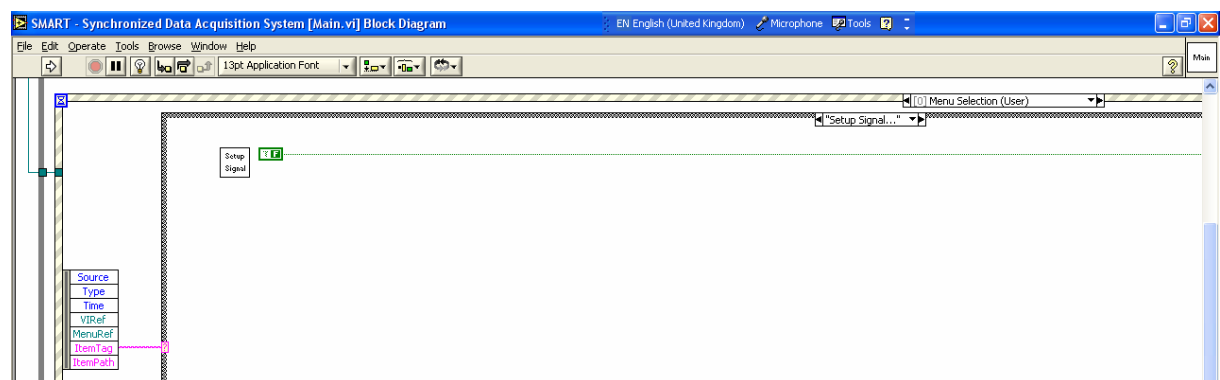


Figure 3.78: Illustration of the code that shows the item “Setup Signal...” of the “Menu Selection” that calls the function “Setup Signal”

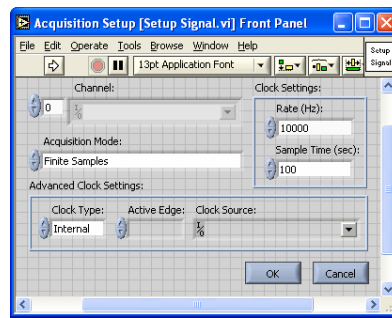


Figure 3.79: Illustration of the window "Acquisition Setup", referred in the project as function "Setup Signal"

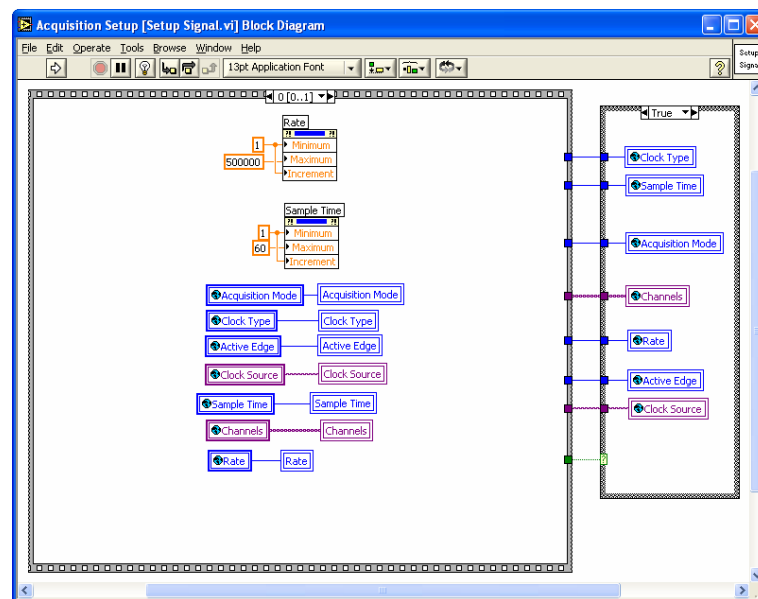


Figure 3.80: Illustration of the code of the function "Setup Signal" that sets the values of the objects, which appear on this window, with the values kept by global variables

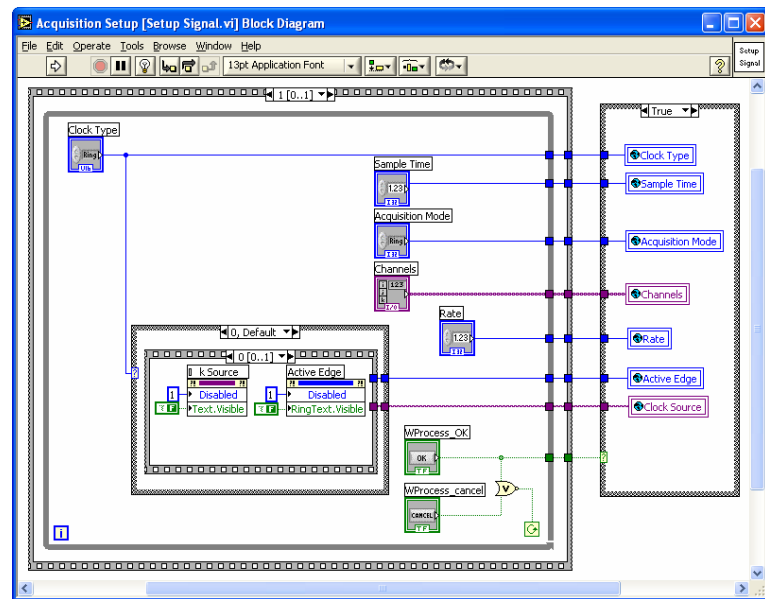


Figure 3.81: Illustration of the code of the function "Setup Signal" that disables the objects "Clock Source" and "Active Edge", case the object "Clock Type" is internal (value 0)

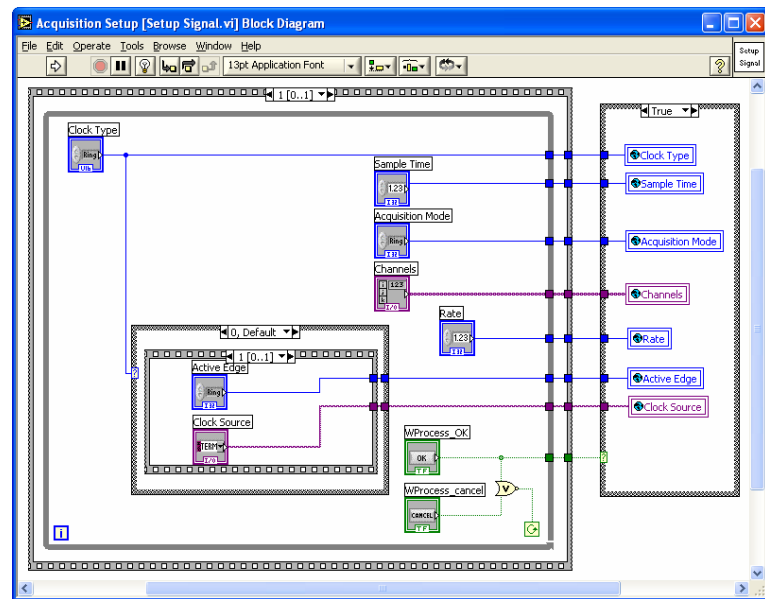


Figure 3.82: Illustration of the code of the function "Setup Signal" that upgrades the global variables "Clock Source" and "Active Edge", case the object "Clock Type" is internal (value 0)

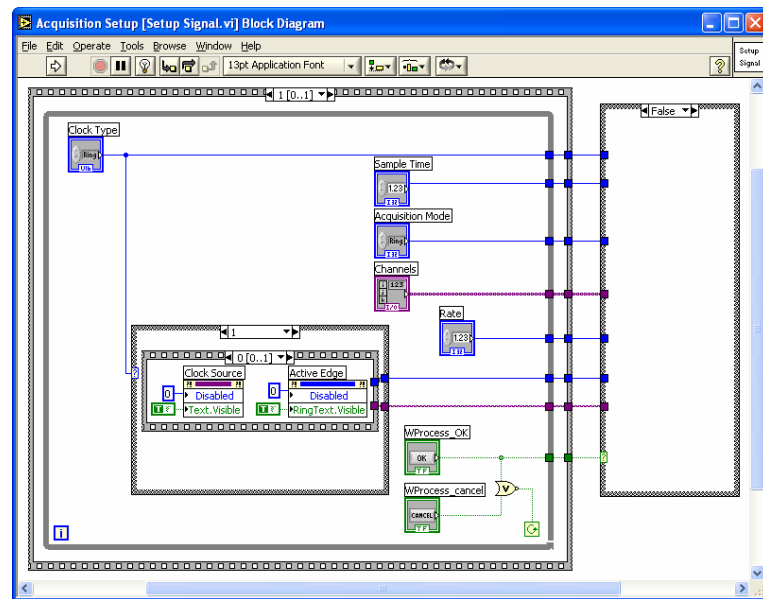


Figure 3.83: Illustration of the code of the function "Setup Signal" that enables the objects "Clock Source" and "Active Edge", case the object "Clock Type" is external (value 1)

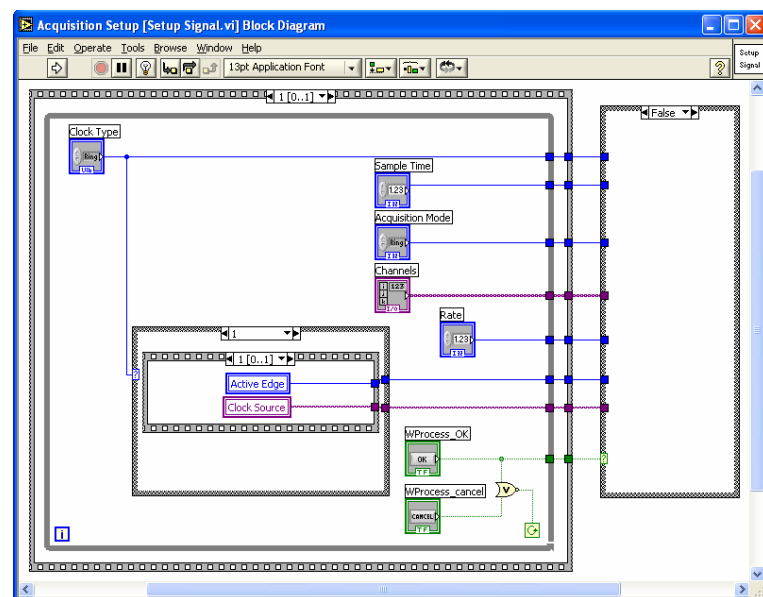


Figure 3.84: Illustration of the code of the function "Setup Signal" that upgrades the global variables "Clock Source" and "Active Edge", case the object "Clock Type" is external (value 1)

Figure 3.85 shows the item "Setup Image..." of the "Menu Selection". The window "Setup Image" (Figure 3.86) is used for selecting the trigger channel and the USB camera. The box "fps" means frames per second acquired by the USB camera. Normally, the webcams can acquire from 15 to 30fps. In order to avoid a repeated frame, the chosen value for this parameter is 10fps. Another reason, it is the quantity of information to be plotted



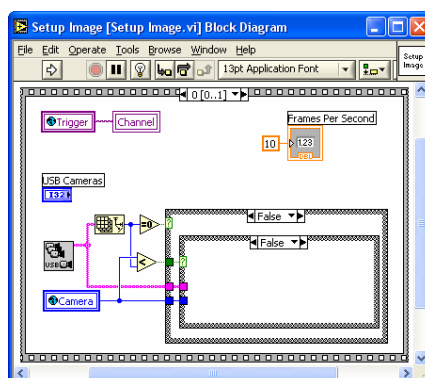


Figure 3.88: Illustration of the code of the function "Setup Image" that verifies if any USB camera is plugged to the computer and if any USB camera session was started (false case)

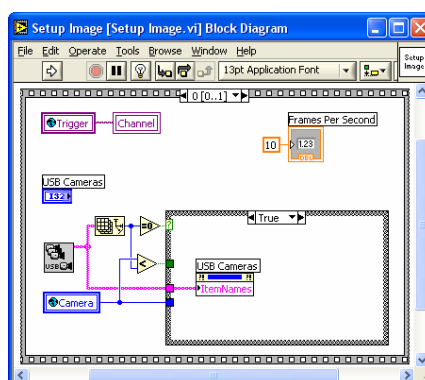


Figure 3.89: Illustration of the code of the function "Setup Image" that verifies if USB cameras are plugged to the computer and lists them

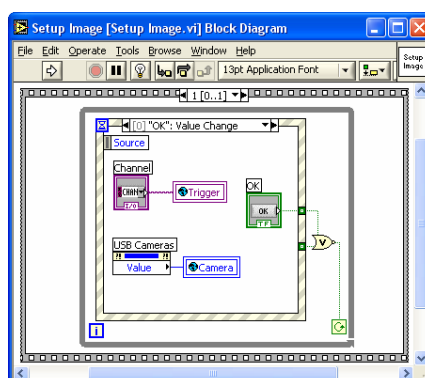


Figure 3.90: Illustration of the code of the function "Setup Image" that upgrades the global variables, if the button OK is pressed

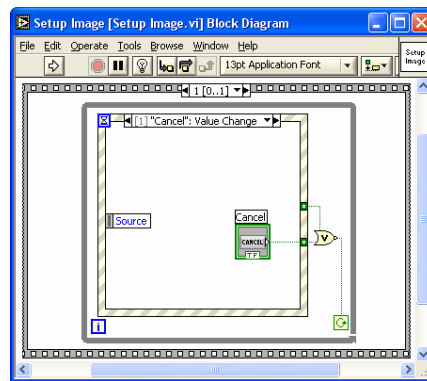


Figure 3.91: Illustration of the code of the function "Setup Image" that does not upgrade any global variable, if the button Cancel is pressed

Figure 3.92 shows the item "Filter..." of the "Menu Selection". The window "Filter" (Figure 3.93) asks the values for the cut-off frequency and the number of poles of the Butterworth low-pass digital filter. Figure 3.94, Figure 3.95 and Figure 3.96 show how it works.

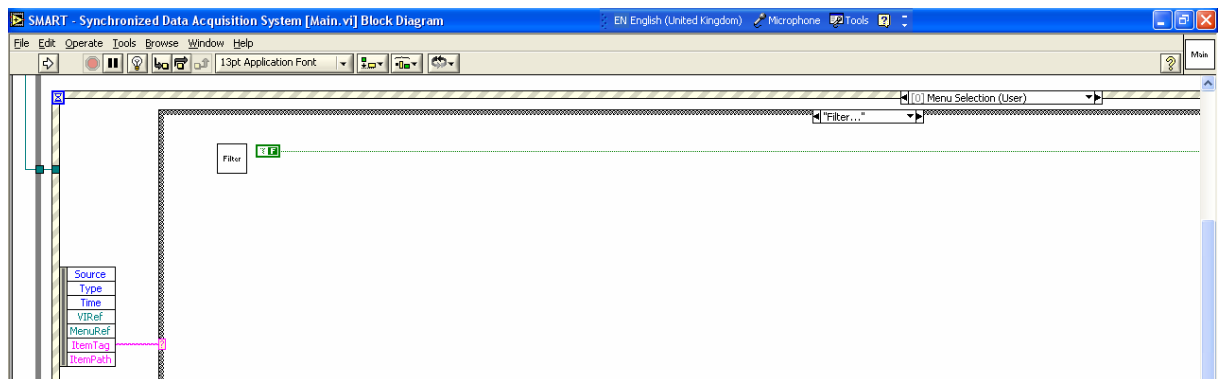


Figure 3.92: Illustration of the code that shows the item "Filter..." of the "Menu Selection" that calls the function "Filter"

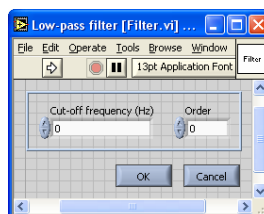


Figure 3.93: Illustration of the window "Low- pass filter", referred in the project as function "Filter"

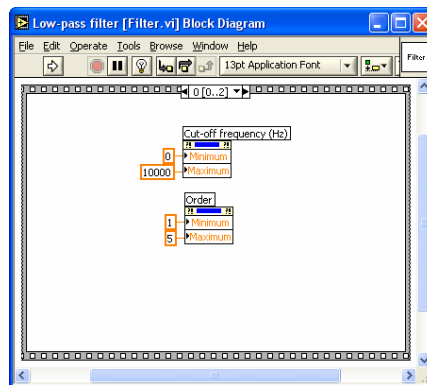


Figure 3.94: Illustration of the code of the function "Filter" that sets the limit values for the objects that appear on this window

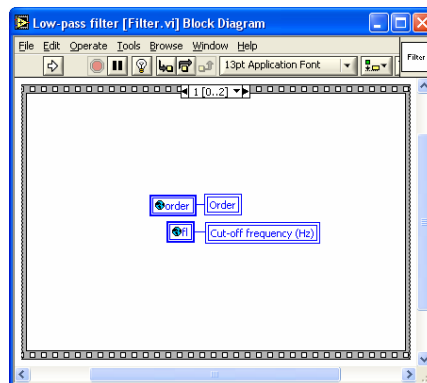


Figure 3.95: Illustration of the code of the function "Filter" that sets the values of the objects that appear on this window

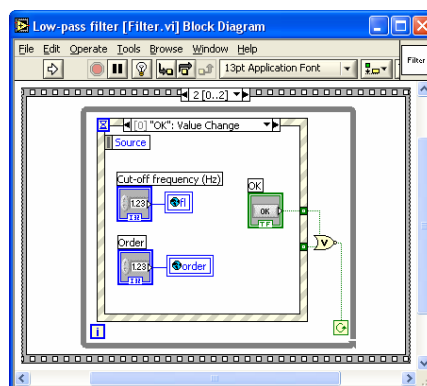


Figure 3.96: Illustration of the code of the function "Filter" that upgrades the global variables, if the button OK is pressed and does not upgrade anything if the button Cancel is pressed

The item "Settings..." of the "Menu Selection" (Figure 3.97) calls the window "Camera" (Figure 3.98), responsible for setting the USB camera parameters. Each webcam has different types of parameters which can be set. Thus, pushing the button "Image", it is

possible to set the image resolution, for instance. And pushing the button “Video”, it is possible to configure brightness, saturation, sharpness, gamma, mode and other parameters.

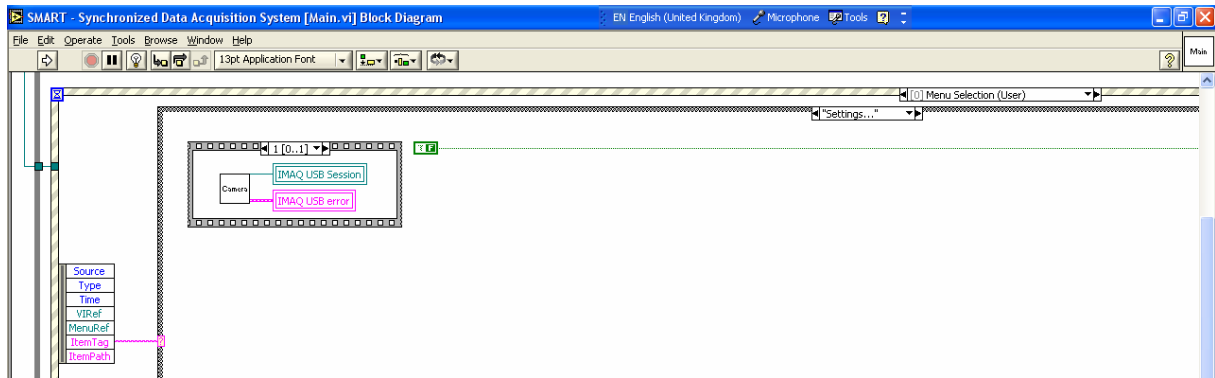


Figure 3.97: Illustration of the code that shows the item “Settings...” of the “Menu Selection” that calls the function “Camera”

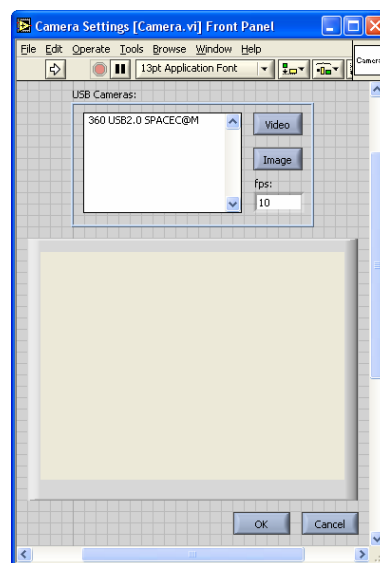


Figure 3.98: Illustration of the window "Camera Settings", referred in the project as function "Camera"

After doing this, those parameters are kept and the image acquisition of the USB camera will be done with the last configuration. Figure 3.99, Figure 3.100, Figure 3.101, Figure 3.102, Figure 3.103, Figure 3.104, Figure 3.105 and Figure 3.106 show how it works.

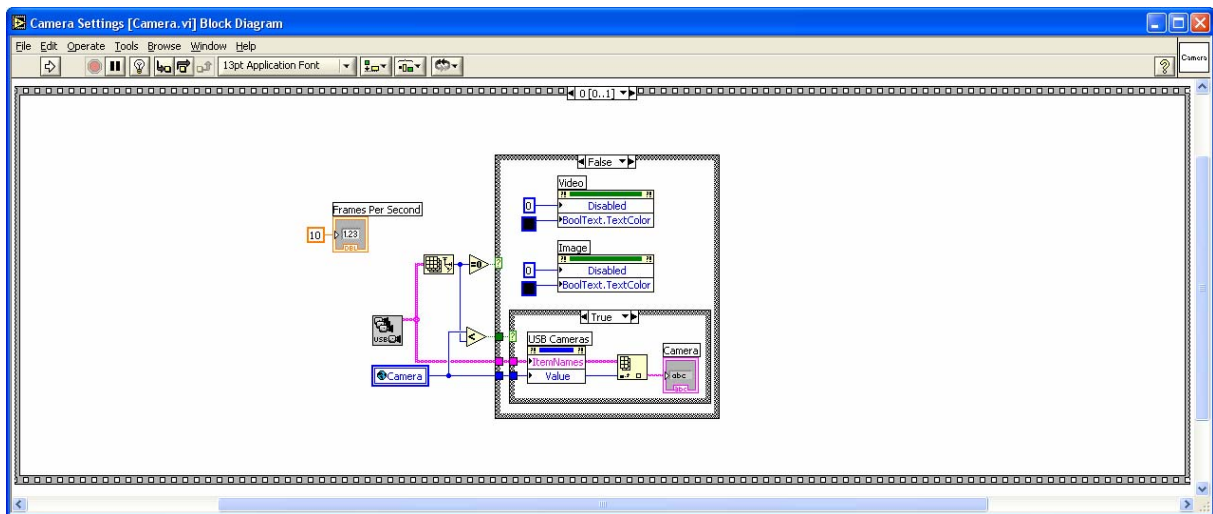


Figure 3.99: Illustration of the code of the function “Camera” that enables the buttons “Video” and “Image”, and shows the name of the USB camera indicated by the global variable “Camera”

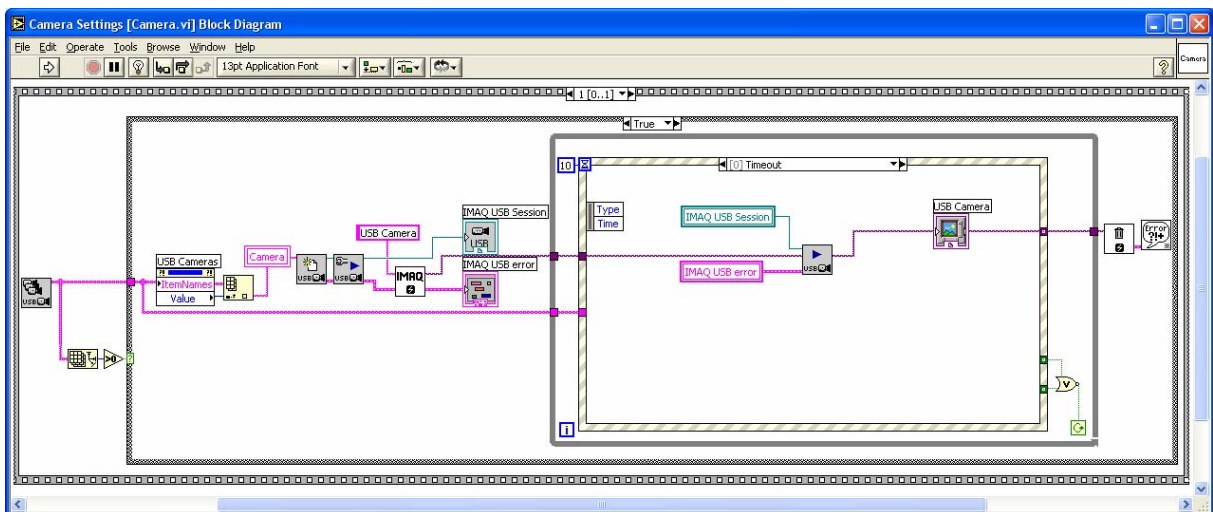


Figure 3.100: Illustration of the code of the function “Camera” that initializes a new USB camera session and shows the USB camera frames, if none of the objects on the window “Camera Settings” is changed in 10ms

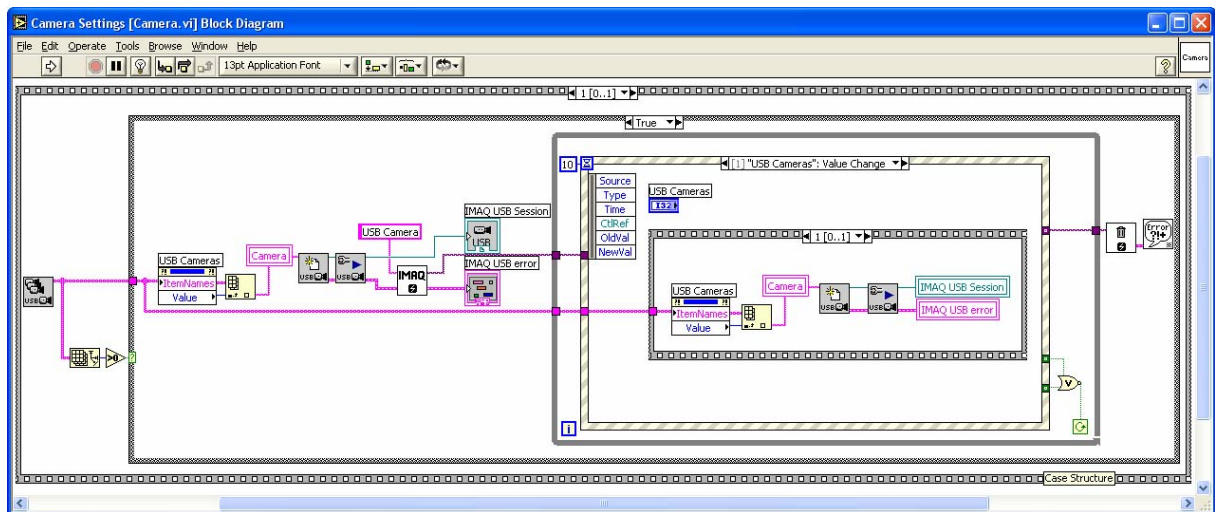


Figure 3.101: Illustration of the code of the function “Camera” that initializes a new USB camera session if the object “USB Cameras” changes, meaning that a new USB camera was chosen

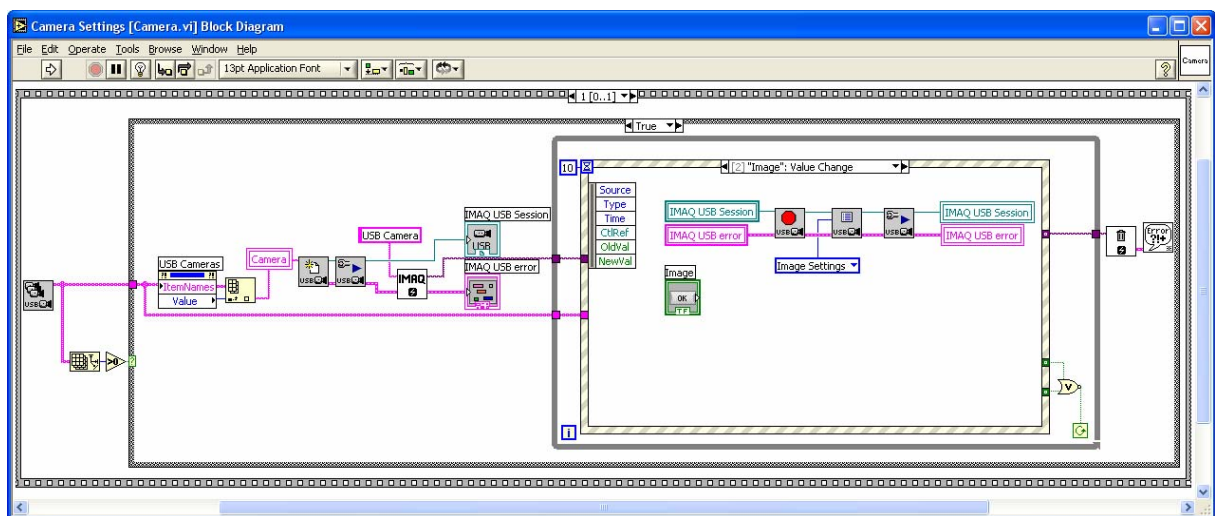


Figure 3.102: Illustration of the code of the function “Camera” that stops the current USB camera session, opens the “Image Settings” window of that camera and starts to play the frames again using the new configuration