GENERAL STRUCTURE:

The program consists of 9 independent programs linked via shm. Each program can be compiled inependently in its folder and uses ../Shm.h header for the SHM segments definitions. Programs can be also compiled all at once using ../Compile command. The main interface is the XRF_Scanner program which calls the other programs depending
on the needs.

EXTERNAL PROGRAMS (LINKED VIA SHM):

```
ADCXRF_Optical_Link            -> Optical link digitizer driver
ADCXRF_USB                     -> USB digitizer driver
Digitiser_interface            -> interface to set/change Digitiser parameters
OnLineMap                      -> displays realtime maps
rate                           -> rate meters for DAQ
ScreenDetector                 -> detects screen resolution for GUI_Creators
XrayTable                      -> Xray energy table
XRF                            -> Spectrum viewer
```

PROGRAMS FOLDERS TREE:

```
Digitiser_interface             -> contains files for digitiser executable
Digitiser_Optical_link           -> optical link drivers for CAEN 5780 digitiser
Digitiser_Usb                  -> usb link drivers for CAEN 5780 digitiser
OnLineMap                      -> contains files for real time map
rate                           -> contains files for ratemeter
ScreenDetector                 -> contains files for screen detection
Spectrum                       -> contains files for spectrum viewer
XrayTable                      -> contains files for energy element table
XRF_Program                    -> destination folder for all programs
XRF_Scanner                    -> contains files for the main program
```

FILES USED TO BUILD THE MAIN PORGRAMS (XRF_SCANNER):

USED FILES: (.CPP)

```
autofocus.cpp                  -> telemeter control and Z automatic positioning
Connectios_Creator.cpp         -> creates connections between GUI obj and
functions
export.cpp                     -> export map images
export_pymca.cpp               -> export data in PYMCA format
external_programs.cpp          -> manages other programs connected to this via
shm
GUI_Creator.cpp                -> creates GUI
laser.cpp                      -> manages lasers for alignement (not implemented)
main.cpp                       -> set basic ownership for ports, sets shm
parameters
mainwindow.cpp                 -> manages motors basic, scan basic, daq basic etc.
mainwindow_define_pixel.cpp    -> define pixel dimensions
mainwindow_loadSHM.cpp         -> load data for diplaying maps
mainwindow_mouse.cpp           -> controls mouse selection on maps
mainwindow_online.cpp          -> manages real time maps
mainwindow_showMap.cpp         -> manages static maps
menu.cpp                       -> creates mainwindow menu
move_motors.cpp                -> moves motors
resouces.qrc                   -> graphic resources
ScanYX_XY.cpp                  -> drives scans (YX and XY)
SHM_Creator.cpp                -> creates shm
Stage_selection.cpp            -> selects stages for the 3 axes
TTY_motors.cpp                 -> Inits serial communications with motors
X_Init.cpp                     -> Inits X stage (6 different stage available)
Y_Init.cpp                     -> Inits Y stage (6 different stage available)
```

```
Z_Init.cpp                      -> Inits Z stage (6 different stage available)


USED FILES: (.H)

mainwindows.h                   -> class definitions
../Header.h                     -> common header (used also in external programs)
../variables.h                  -> common variables (shared with external programs)
../Shm.h                        -> global definition of shm segments


OTHER FILES:

Resolution                      -> generated by ScreenDetector and use by programs
calibration.txt                 -> used by the spectrum viewer


INSTALL and COMPILE:

Each program can be compiled indide its folder and general compiler (Compile) is
present in the root directory. The command "./Compile" enters each of the 8
folders
compiles the program and copies it in the XRF_Program folder; at the end of the
loop launches XRF_Scanner.
The cmd "./install" operates in the same way but rebuilds also the environment.


INDEX OF THE FUNCTIONS IN MAIN PROGRAMS:


~MainWindow()                   -> mainwindow.cpp
Abort()                         -> mainwindow.cpp
AbortZ()                        -> mainwindow.cpp
Appartiene(int,int, struct)     -> mainwindow_DefinePixels.cpp
AssignACM()                     -> TTY_motors.cpp
AssignX()                       -> TTY_motors.cpp
AssignY()                       -> TTY_motors.cpp
AssignZ()                       -> TTY_motors.cpp
AutoFocusRunning()              -> autofocus.cpp
caenoscilloscope()              -> external_programs.cpp
char *read_Kanswer()            -> autofocus.cpp
CheckSegFault()                 -> mainwindow.cpp
CheckXOnTarget()                -> mainwindow.cpp
CheckYOnTarget()                -> mainwindow.cpp
CheckZOnTarget()                -> mainwindow.cpp
CONNECTIONS_CREATOR()           -> Connetions_Creator.cpp
CREATE_MENU()                   -> Menu.cpp
createActions()                 -> Menu.cpp
createStatusBar()               -> mainwindow.cpp
CutB()                          -> mainwindow.cpp
Define_Pixels()                 -> mainwindow_DefinePixels.cpp
Detector()                      -> external_programs.cpp
Digitiser2()                    -> external_programs.cpp
displayImage_SHM()              -> mainwindow_showMap.cpp
Enable_TabWidget_3_4_XY()       -> mainwindow.cpp
Enabling_Tabwidget()            -> TTY_motors.cpp
Exit()                          -> mainwindow.cpp
exportpymca()                   -> export_pymca.cpp
Focustimer()                    -> autofocus.cpp
GoOnLine()                      -> mainwindow_online.cpp
GUI_CREATOR()                   -> GUI_creator.cpp
Helium_interface()              -> external_programs.cpp
hideImage()                     -> mainwindow.cpp
Image_Export()                  -> export.cpp
InizializzazioneKeyence()       -> TTY_motors.cpp
InizializzazioneX()             -> X_Init.cpp
```

```
InizializzazioneY()             -> Y_Init.cpp
InizializzazioneZ()             -> Z_init.cpp
invia_comando_X(int,ch,ch)      -> mainwindow.cpp
invia_comando_Y(int,ch,ch)      -> mainwindow.cpp
invia_comando_Z(int,ch,ch)      -> mainwindow.cpp
Laser_interface()               -> laser.cpp
Laser_switching()               -> laser.cpp
LaserOff()                      -> laser.cpp
LaserOn()                       -> laser.cpp
LoadNewFile_SHM()               -> mainwindow.cpp
LoadNewFileWithCorrection_SHM() -> mainwindow_loadSHM.cpp
LoadNewFileWithNoCorrection_SHM -> mainwindow_loadSHM.cpp
LoadTxt()                       -> mainwindow.cpp
MapCorrection()                 -> mainwindow.cpp
MergeTxt()                      -> mainwindow.cpp
mousePressEvent(QMouseEvent)    -> mainwindow_mouse.cpp
mouseReleaseEvent(QMouseEvent)  -> mainwindow_mouse.cpp
Move_backward()                 -> move_motors.cpp
Move_down()                     -> move_motors.cpp
Move_forward()                  -> move_motors.cpp
Move_left()                     -> move_motors.cpp
Move_right()                    -> move_motors.cpp
Move_up()                       -> move_motors.cpp
MoveDoubleClick()               -> move_motors.cpp
MoveX_To()                      -> move_motors.cpp
MoveX(double)                   -> move_motors.cpp
MoveY(double)                   -> move_motors.cpp
MoveZ(double)                   -> move_motors.cpp
NameACM(int)                    -> TTY_motors.cpp
NameX(int)                      -> TTY_motors.cpp
NameY(int)                      -> TTY_motors.cpp
NameZ(int)                      -> TTY_motors.cpp
open_MAP()                      -> mainwindow.cpp
OPTICAL_DAQ()                   -> mainwindow.cpp
PassoX_Func(double)             -> mainwindow.cpp
PassoY_Func(double)             -> mainwindow.cpp
PassoZ_Func(double)             -> mainwindow.cpp
Pixel_BIG *Crea_PX(int,int,int) -> mainwindow_DefinePixels.cpp
PixelCrct()                     -> mainwindow.cpp
Pixels()                        -> mainwindow.cpp
RateMeter()                     -> external_programs.cpp
read_Xanswer() (char *)         -> mainwindow.cpp
read_Xanswer2() (string)        -> mainwindow.cpp
read_Yanswer() (char *)         -> mainwindow.cpp
read_Yanswer2()  (string)       -> mainwindow.cpp
read_Zanswer() (char *)         -> mainwindow.cpp
read_Zanswer2()  (string)       -> mainwindow.cpp
readKeyence()                   -> autofocus.cpp
SaveTxt()                       -> mainwindow.cpp
ScanXY()                        -> ScanYX_XY.cpp
ScanYX()                        -> ScanYX_XY.cpp
SelectChannels()                -> mainwindow.cpp
SelMeasTime()                   -> mainwindow.cpp
SetCurrentAction(QString)       -> mainwindow.cpp
SetSerialXName(int)             -> TTY_motors.cpp
SetSerialYName(int)             -> TTY_motors.cpp
SetSerialZName(int)             -> TTY_motors.cpp
SHM_CREATOR()                   -> SHM_Creator.cpp
ShowHistogram()                 -> external_programs.cpp
StartVme()                      -> external_programs.cpp
StartX()                        -> mainwindow.cpp
StartXYScan()                   -> mainwindow.cpp
StartXYScan()                   -> mainwindow.cpp
StartY()                        -> mainwindow.cpp
StartZ()                        -> mainwindow.cpp
Stop_Vme()                      -> external_programs.cpp
```

```
Stop()                          -> mainwindow.cpp
StopZ()                         -> mainwindow.cpp
string read_Kanswer2()          -> autofocus.cpp
timerEvent()                    -> mainwindow.cpp
TrackingON()                    -> autofocus.cpp
Treshold()                      -> mainwindow.cpp
USB_DAQ()                       -> mainwindow.cpp
Velocity(double)                -> mainwindow.cpp
VelocityZ(double)               -> mainwindow.cpp
VLC_interface()                 -> external_programs.cpp
WritePositionXY()               -> ScanYX_XY.cpp
WritePositionYX()               -> ScanYX_XY.cpp
X_Motor_selection(int)          -> Stage_selection.cpp
X_to(double)                    -> mainwindow.cpp
Xmassimo(double)                -> mainwindow.cpp
Xminimo(double)                 -> mainwindow.cpp
XrayTable()                     -> external_programs.cpp
Y_Motor_selection(int)          -> Stage_selection.cpp
Y_to(double)                    -> mainwindow.cpp
Ymassimo(double)                -> mainwindow.cpp
Yminimo(double)                 -> mainwindow.cpp
Z_Motor_selection(int)          -> Stage_selection.cpp
Z_to(double)                    -> mainwindow.cpp
Zmassimo(double)                -> mainwindow.cpp
Zminimo(double)                 -> mainwindow.cpp


THE OTHER PROGRAMS - FILES AND INDEX OF THE FUNCTIONS:

DIGITISER_INTERFACE:
FILES (.CPP): main.cpp, mainwindow.cpp, GuiCreator.cpp
FILES (.H): mainwindow.h
FUNCTIONS:
~MainWindow()                   -> mainwindow.cpp
Base_Line_Mean(int)             -> mainwindow.cpp
DC_Offset(int)                  -> mainwindow.cpp
Digi_range(int)                 -> mainwindow.cpp
Digi_treshold(int)              -> mainwindow.cpp
Digitiser_data_download_enable()-> mainwindow.cpp
Exit()                          -> mainwindow.cpp
Fall_Time(int)                  -> mainwindow.cpp
Flat_Top(int)                   -> mainwindow.cpp
GuiCreator()                    -> GuiCreator.cpp
HoldOff(int)                    -> mainwindow.cpp
Peak_Holdoff(int)               -> mainwindow.cpp
Peak_Mean(int)                  -> mainwindow.cpp
Peaking_Delay(int)              -> mainwindow.cpp
Rise_Time(int)                  -> mainwindow.cpp
Smoothing_Factor(int)           -> mainwindow.cpp
Trapezoid_Gain(int)             -> mainwindow.cpp


DIGITISER_OPTICAL_LINK:         -> (C program)
FILES (.C .CPP):  ADCXRF.c, Function.c, keyb.c
FILES (.H): CAENDPPLib.h, CAENDPPLibTypes.h, Functions.h,
            keyb.h
FUNCTIONS: reimplemented in ADCXRF modifying CAEN source


DIGITISER_USB:                  -> (C program)
FILES (.C .CPP): ADCXRF.c, Function.c, keyb.c
FILES (.H): CAENDPPLib.h, CAENDPPLibTypes.h, Functions.h,
            keyb.h
FUNCTIONS: reimplemented in ADCXRF modifying CAEN source
```

```
ON_LINE_MAP:
FILES (.CPP): main.cpp, mainwindow.cpp, GUI_CREATOR.cpp
FILES (.H): mainwindow.h
FUNCTIONS:
~MainWindow()                   -> mainwindow.cpp
displayImageOnLine()            -> mainwindow.cpp
Element_Ag()                    -> mainwindow.cpp
Element_Au()                    -> mainwindow.cpp
Element_Ba()                    -> mainwindow.cpp
Element_Ca()                    -> mainwindow.cpp
Element_Co()                    -> mainwindow.cpp
Element_Cr()                    -> mainwindow.cpp
Element_Cu()                    -> mainwindow.cpp
Element_Fe()                    -> mainwindow.cpp
Element_Hg()                    -> mainwindow.cpp
Element_K()                     -> mainwindow.cpp
Element_Pb()                    -> mainwindow.cpp
Element_Si()                    -> mainwindow.cpp
Element_Sn()                    -> mainwindow.cpp
Element_Ti()                    -> mainwindow.cpp
Element_Zn()                    -> mainwindow.cpp
enableOnLine(bool)              -> mainwindow.cpp
enableOnTop(bool)               -> mainwindow.cpp
Hi_Ch_Value(int)                -> mainwindow.cpp
Lo_Ch_Value(int)                -> mainwindow.cpp
quit()                          -> mainwindow.cpp


RATE:                           -> (QWT_6.1.2 program)
FILES (.CPP): main.cpp, dial.cpp, dialtab.cpp
FILES (.H): dial.h, dialtab.h
FUNCTIONS:
~DialBox()                      -> dial.cpp
createDial(int)                 -> dial.cpp
DialTab()                       -> dialtab.cpp
EventEnable()                   -> dial.cpp
setNeedleValue(int)             -> dial.cpp
setNum(double)                  -> dial.cpp
TimerEvent()                    -> dial.cpp


SCREENDETECTOR:
FILES  (.CPP): main.cpp
FUNCTIONS: No graphic. Basic program to get screen resolution


SPECTRUM:
FILES (.CPP): main.cpp, mainwindow.cpp, plot.cpp
FILES (.H): complexnumber.h, mainwindow.h, pixmaps.h, plot.h
FUNCTIONS:
~MainWindow()                   -> mainwindow.cpp
on_k4(bool)                     -> mainwindow.cpp
AutoCalibrate()                 -> mainwindow.cpp
Calibration(double)             -> plot.cpp
cancClicked()                   -> mainwindow.cpp
ch1(double)                     -> plot.cpp
ch2(double)                     -> plot.cpp
Check_SHM()                     -> plot.cpp
E1(double)                      -> plot.cpp
E2(double)                      -> plot.cpp
enableAutoScale(bool)           -> mainwindow.cpp
enableOnTop(bool)               -> mainwindow.cpp
enableRunMode(bool)             -> plot.cpp
enableZoomMode(  )              -> mainwindow.cpp
Energy_Channel_Choice(bool)     -> plot.cpp
exportTxt()                     -> mainwindow.cpp
```

```
howPixelHisto()                 -> plot.cpp
logSpace()                      -> plot.cpp
mousePressEvent(QMouseEvent)    -> mainwindow.cpp
moved(const QPoint &pos)        -> mainwindow.cpp
Offset(double)                  -> plot.cpp
okClicked()                     -> mainwindow.cpp
Open(  )                        -> plot.cpp
PreOpen()                       -> mainwindow.cpp
print()                         -> mainwindow.cpp
Retta_Cal()                     -> plot.cpp
selected(const QPolygon)        -> mainwindow.cpp
showData(double,double,int)     -> plot.cpp
showInfo()                      -> mainwindow.cpp
showInfo(QString)               -> mainwindow.cpp
timerRefreshEvent()             -> plot.cpp


XRAY_TABLE:
FILES (.CPP): main.cpp, mainwindow.cpp, GuiCreator.cpp
FILES (.H): mainwindow.h
FUNCTIONS:
~MainWindow()                   -> mainwindow.cpp
Digi_range(int)                 -> mainwindow.cpp
Exit()                          -> mainwindow.cpp
```