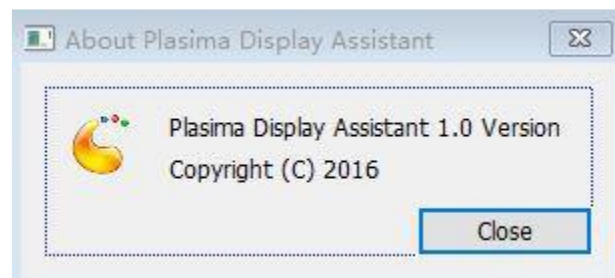# Plasimo Display Assistant User Guide

Junming Zhang

April 26, 2017

Interface installation

OpenCV configuration is the first step of installation. The program cannot build successfully without it because the image command is developed based on OpenCV.

Step 1:

Open the opencv-2.4.10.exe and release the OpenCV files to hard disk D which is shown in Figure 1.
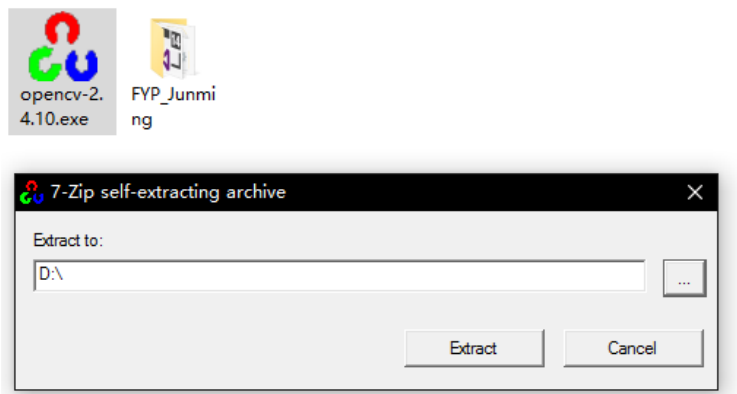

Figure 1. The configuration step 1 of OpenCV

Step 2:

Go to hard disk D, open the "bin" folder followed by clicking D:\opencv\build\x64\vc10\bin. Finally copy all files in this folder to system folder at C:\Windows\System32 and C:\Windows\ SysWOW64 which is shown in Figure 2.
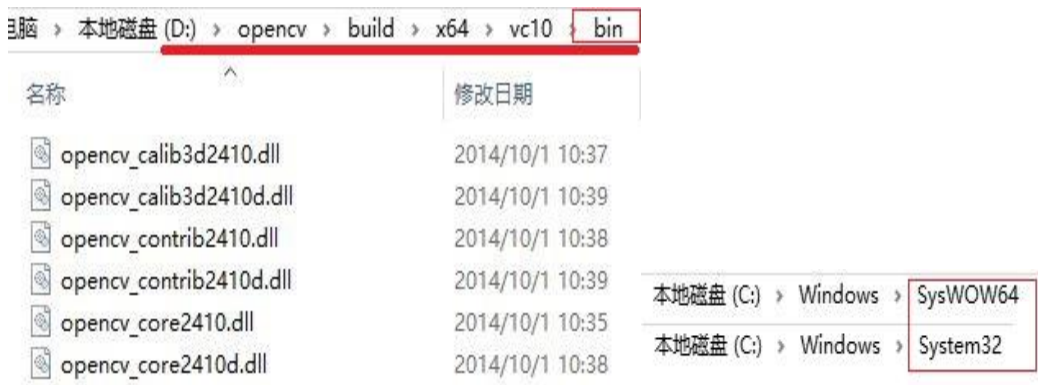

Figure 2. The configuration step 2 of OpenCV

Step 3:

After that, open the project file "Junming_Zhang_FYP.sln" by using Visual Studio 2015 which is shown in Figure 3.
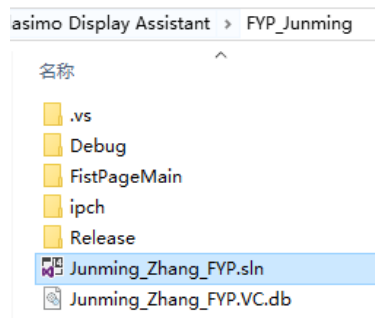
Figure 3. The configuration step 3 of OpenCV

Step 4:

Right-click the project name and then open the properties setting which is shown in Figure 4.
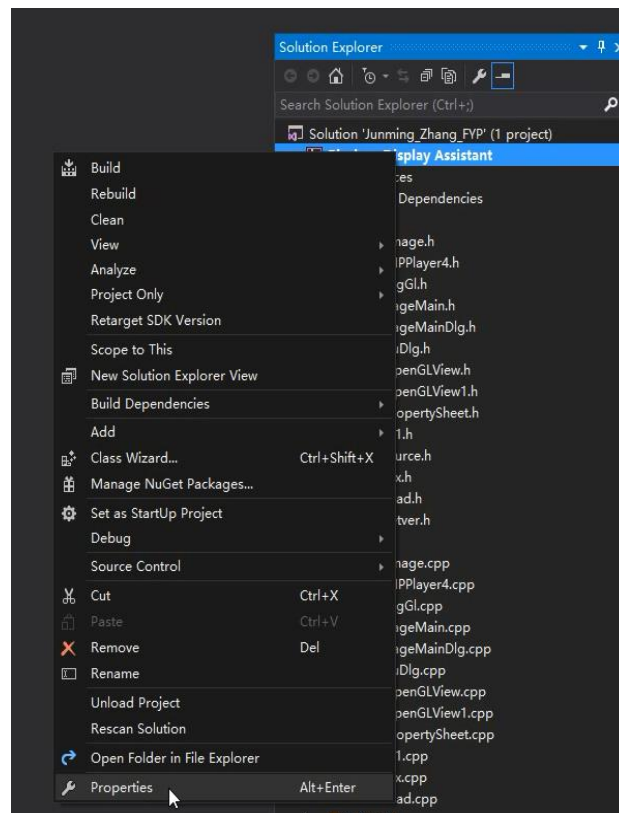

Figure 4. The configuration step 4 of OpenCV

Step 5:

Find the Include Directories of VC++ Directories in the Configuration Properties and then type the following address into it which is shown in Figure 5 and 6.

D:\opencv\build\include\opencv

D:\opencv\build\include\opencv2
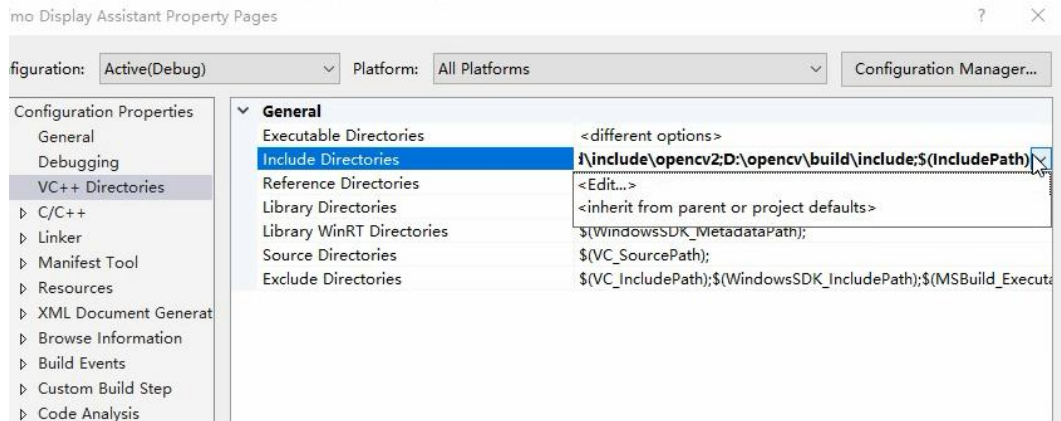
D:\opencv\build\include
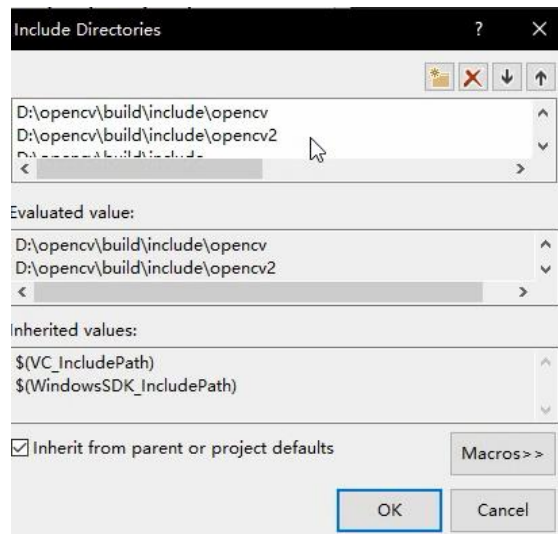
Figure 5. The configuration step 5 of OpenCV

Figure 6. The configuration step 5 of OpenCV

Step 6:

It is similar to the previous step, find the Library Directories of VC++ Directories in the Configuration Properties and then type the following address into it which is shown in Figure 7 and 8.
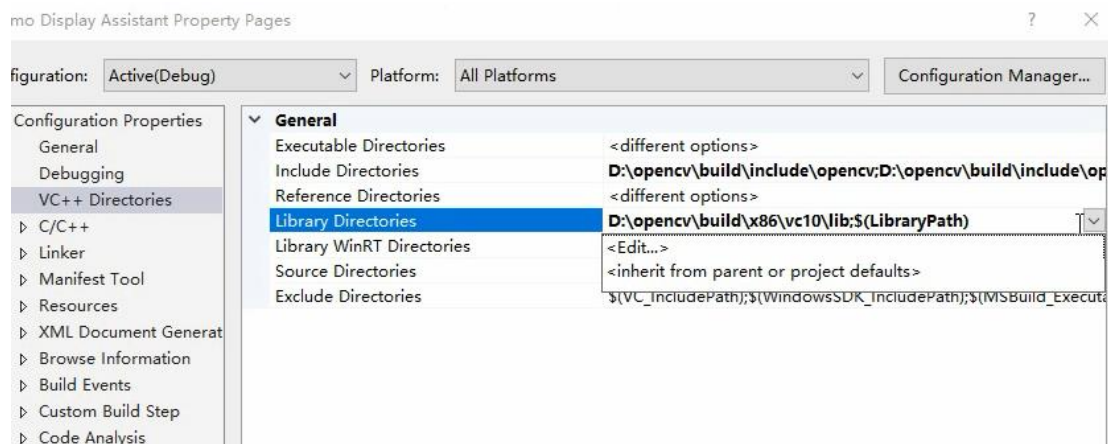
D:\opencv\build\x86\vc10\lib

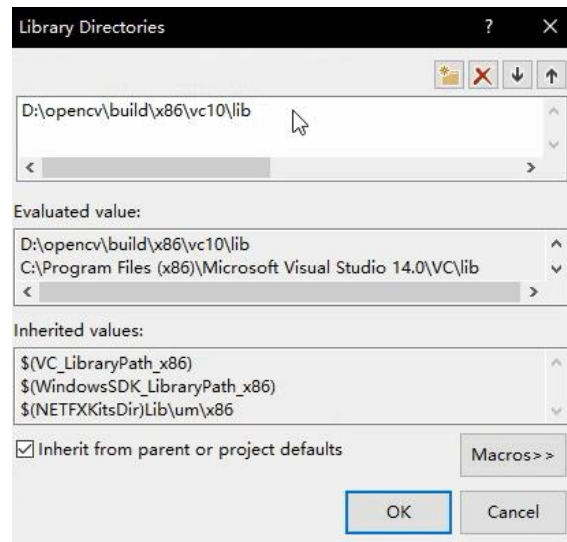Figure 7. The configuration step 6 of OpenCV

Figure 8. The configuration step 6 of OpenCV

Step 7:

The last step is to find the Additional Dependencies of Input in the Linker and then type the following names of library file into it which is shown in Figure 9 and 10.

opencv_ml2410d.lib
opencv_calib3d2410d.lib
opencv_contrib2410d.lib
opencv_core2410d.lib
opencv_features2d2410d.lib
opencv_flann2410d.lib
opencv_gpu2410d.lib
opencv_highgui2410d.lib
opencv_imgproc2410d.lib
opencv_legacy2410d.lib
opencv_objdetect2410d.lib
opencv_ts2410d.lib
opencv_video2410d.lib
opencv_nonfree2410d.lib
opencv_ocl2410d.lib
opencv_photo2410d.lib
opencv_stitching2410d.lib
opencv_superres2410d.lib
opencv_videostab2410d.lib
opencv_objdetect2410.lib
opencv_ts2410.lib
opencv_video2410.lib
opencv_nonfree2410.lib
opencv_ocl2410.lib
opencv_photo2410.lib
opencv_stitching2410.lib

opencv_superres2410.lib

opencv_videostab2410.lib

opencv_calib3d2410.lib

opencv_contrib2410.lib

opencv_core2410.lib

opencv_features2d2410.lib

opencv_flann2410.lib

opencv_gpu2410.lib

opencv_highgui2410.lib

opencv_imgproc2410.lib
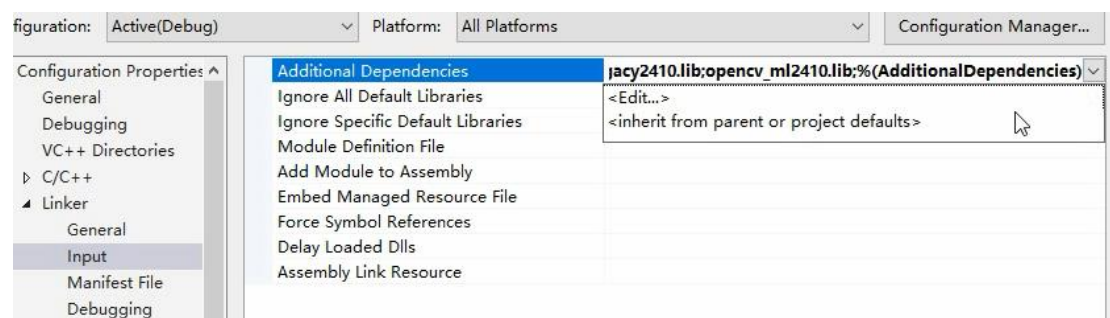
opencv_legacy2410.lib

opencv_ml2410.lib

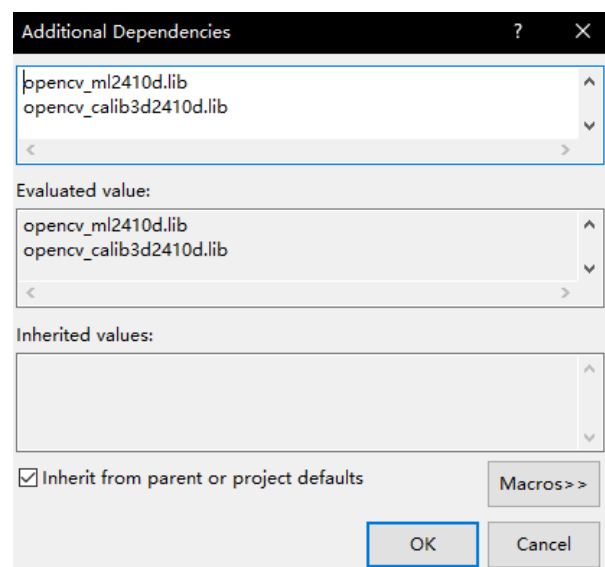
Figure 9. The configuration step 7 of OpenCV

Figure 10. The configuration step 7 of OpenCV

The program can be built now by Visual Studio 2015. Click Local Windows Debugger to obtain the executable program which is shown in Figure 11.

Figure 11. Build process of program

Extension

In this project, pdp_md2d is the target model. However, it is just one model of the Plasimo. According to the investigate of Plasimo, it could found that different model will generate different files with a different name. Therefore, the addition exploitation will focus on the extensional function of this project.

The matched result of files and corresponding properties will stored in a CSV file in the NameManagement folder. For example, the pdp_md2d model generates 68 output files. However, there are 87 properties of this model. Some output file may be lost because the original code of pdp_md2d model limited these files.

The CSV file can quick manage the matching result. Figure 80 display the basic theory of matching. The sub menu is marked by number 1 to 87 because there are 87 properties in pdp_md2d model, The CSV file will store the corresponding text file name based on the number 1 to 87. The CSV reading command will process theses data and use ShowFile command to display required data. ShowFile(0) is corresponding to the first number of CSV and sub menu.
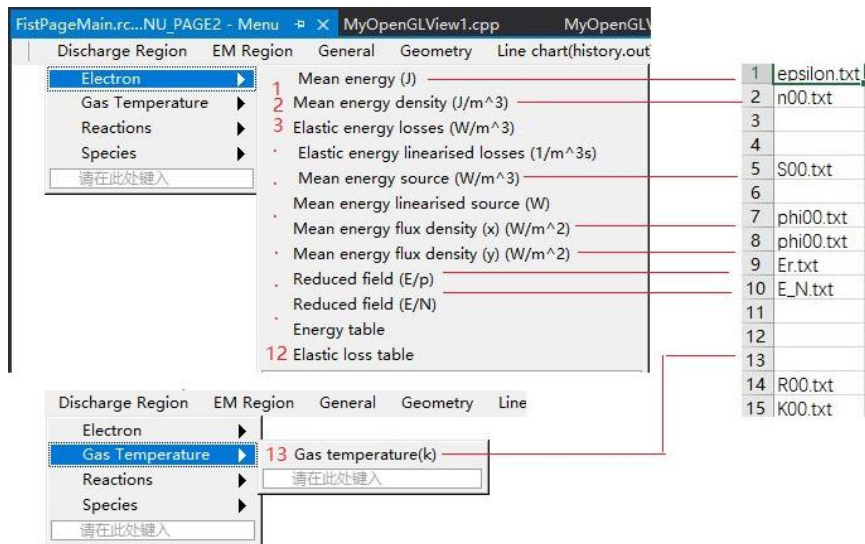

Figure 12. The basic theory of matching data and properties

```cpp
CString ** CSVRead::ReadCsvFile(CString filepath, char divided,int *row,int *col)//Read csv file, get related data and save into an arrray
{
    CStringArray * Array = NULL;

    if (!PathFileExists(filepath))
        return NULL;
    CSVRead read(filepath);

    int Rowcount = read.FileRowCount();
    Array = read.CSVInf(divided);
    int Columcount = Array->GetSize();

    CString **p = new CString *[Rowcount];
    for (int i = 0; i < Rowcount; i++)
        p[i] = new CString[Columcount];
    for (int i = 0; i < Columcount; i++)
        p[0][i] = Array->GetAt(i);

    *row = Rowcount;
    *col = Columcount;
    for (int i = 1; i<Rowcount ; i++)
    {
        Array = read.CSVInf(divided);
        for (int j = 0; j < Columcount; j++)
        {
            CString a = Array->GetAt(j);
            p[i][j] = Array->GetAt(j);
        }
    }

    return p;
}
```
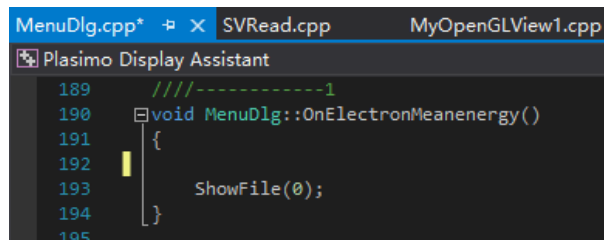
Figure 13. The code of reading csv file function

Figure 14. Display corresponding data when click sub-menu