```
/*! \class Microscope
Interaction with the USB 5 MP microscope
*/

#pragma once
#define MICROSCOPE_VERSION 1 /*!< Version of the class*/

#define MIN_BRIGHTNESS -64
#define MAX_BRIGHTNESS 64
#define MIN_CONTRAST 0
#define MAX_CONTRAST 64
#define MIN_SATURATION 0
#define MAX_SATURATION 128
#define MIN_HUE -40
#define MAX_HUE 40
#define MIN_GAMMA 40
#define MAX_GAMMA 500
#define MIN_SHARPNESS 1
#define MAX_SHARPNESS 25

#include "stdint.h"
#include <vector>

#include "USB.h"

#include <opencv2/photo.hpp>
#include <opencv2/imgcodecs.hpp>
#include <opencv2/opencv.hpp>
#include <opencv/highgui.h>

namespace Hardware{
    class Microscope
    {
    public:
        /*! Struct that represent the Resolution that is used */
        struct Resolution
        {
        public:
            uint16_t Width;     /*!< Width of the image*/
            uint16_t Height;    /*!< Height of the image*/
        };

        uint8_t FrameDelayTrigger;  /*!< Delay in seconds */
        cv::Mat LastFrame;              /*!< Last grabbed and processed frame */
        Resolution Dimensions;      /*!< Dimensions of the frame */

        Microscope();
        Microscope(uint8_t frameDelayTrigger, Resolution dimensions = Resolution{ 2592, 1944 });
        //Microscope(uint8_t frameDelayTrigger = 3, Resolution dimensions = Resolution{ 1944, 2592 });
        ~Microscope();

        void GetFrame(cv::Mat &dst);
        void GetHDRFrame(cv::Mat &dst, uint32_t noframes = 5);

        bool IsOpened();
        void Release();

    private:

        cv::VideoCapture captureDevice; /*!< An openCV instance of the capture device*/
        void openCam();

        std::vector<cv::Mat> HDRframes;
        std::vector<float> times;
    };
}

```