```
//
// Programmer:    Craig Stuart Sapp <craig@ccrma.stanford.edu>
// Creation Date: Tue May  9 05:24:33 PDT 2006
// Last Modified: Sun May 21 00:03:39 PDT 2006 (parameter control added)
// Filename:      MzChronogram.h
// URL:           http://sv.mazurka.org.uk/include/MzChronogram.h
// Documentation: http://sv.mazurka.org.uk/MzChronogram
// Syntax:        ANSI99 C++; vamp 0.9 plugin
//
// Description:   Display audio signal in two dimensions.
//

#ifndef _MZCHRONOGRAM_H_INCLUDED
#define _MZCHRONOGRAM_H_INCLUDED

#include "MazurkaPlugin.h"  // Mazurka plugin interface for Sonic Visualiser


class MzChronogram : public MazurkaPlugin {

   public:

   // plugin interface functions:

                    MzChronogram            (float samplerate);
      virtual      ~MzChronogram            ();

      // required polymorphic functions inherited from PluginBase:
      std::string   getName                 (void) const;
      std::string   getMaker                (void) const;
      std::string   getCopyright            (void) const;
      std::string   getDescription          (void) const;
      int           getPluginVersion        (void) const;

      // optional parameter interface functions
      ParameterList getParameterDescriptors (void) const;

      // required polymorphic functions inherited from Plugin:
      InputDomain   getInputDomain          (void) const;
      OutputList    getOutputDescriptors    (void) const;
      bool          initialise              (size_t channels,
                                             size_t stepsize,
                                             size_t blocksize);
      FeatureSet    process                 (float **inputbufs,
                                             Vamp::RealTime timestamp);
      FeatureSet    getRemainingFeatures    (void);
      void          reset                   (void);

      // optional polymorphic functions from Plugin:
      size_t        getPreferredStepSize    (void) const;
      size_t        getPreferredBlockSize   (void) const;
      // size_t     getMinChannelCount      (void) const { return 1; }
      size_t        getMaxChannelCount      (void) const { return 17; }
                                            // can handle up to septendecaphonic

   // non-interface functions and variables:

      static void   buildLookupTable (float* buffer,int size,float sensitivity);

   private:

      int    mz_whichchannel;  // which channel to display (-1 for all)
      int    mz_diffB;         // which channel to use for stereo diff
      float *mz_lookup;        // used with the sensitivity scaling factor

      // input parameters:
      //
      //    "verticalperiod"   -- number of samples on vertical axis
      //    "frequency"        -- base frequency of vertical axis
      //    "chroma"           -- chroma of a base frequency
      //    "octave"           -- octave number of a base frequency
      //    "channelview"      -- which channel to display
      //    "sensitivity"      -- control high/low amplitude contrast

};


#endif // _MZCHRONOGRAM_H_INCLUDED
```