FSST 3. Jahrgang

Qt Projekt 2014

Signalanalyse

HTL St. Pölten EL

Signalanalyse

ausgeführt von

Hofstätter Alexander 3BHEL

am 6. Juni 2014

1 Aufgabenstellung

Einlesen von Messdaten von einer Datei,
mit grafischer Darstellung der Messdaten und
Berechnung von Mittelwert, Uss und RMS.

2 Programmbeschreibung

Beim Programmstart wird die Einggabedatei eingelesen (CSV-Datei mit allen Messdaten direkt vom Oszilloskop)

Anschließend werden die Daten analysiert und visualisiert.

Auf der rechten Fensterseite sind diverse Analysewerte zu finden (RMS, Mittel und Spitze-Spitze)

3 Programmübersicht

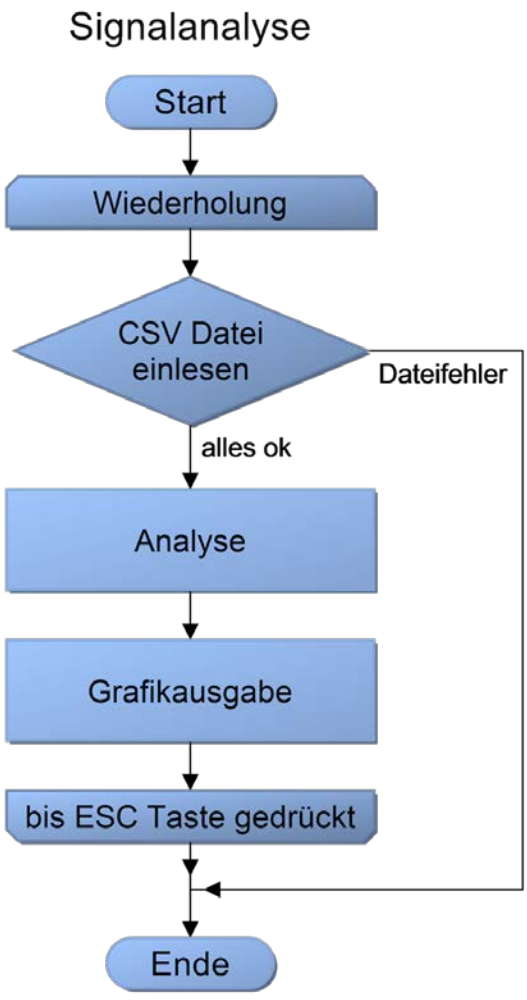
Alle verwendeten Methoden und Klassen findet man in grafik.h sowie mainwindow.h und den dazugehörigen Source-files.

Die CSV-Datei folgt folgenden Format Regeln:

Zeit,Wert-Kanal-1,Wert-Kanal-2

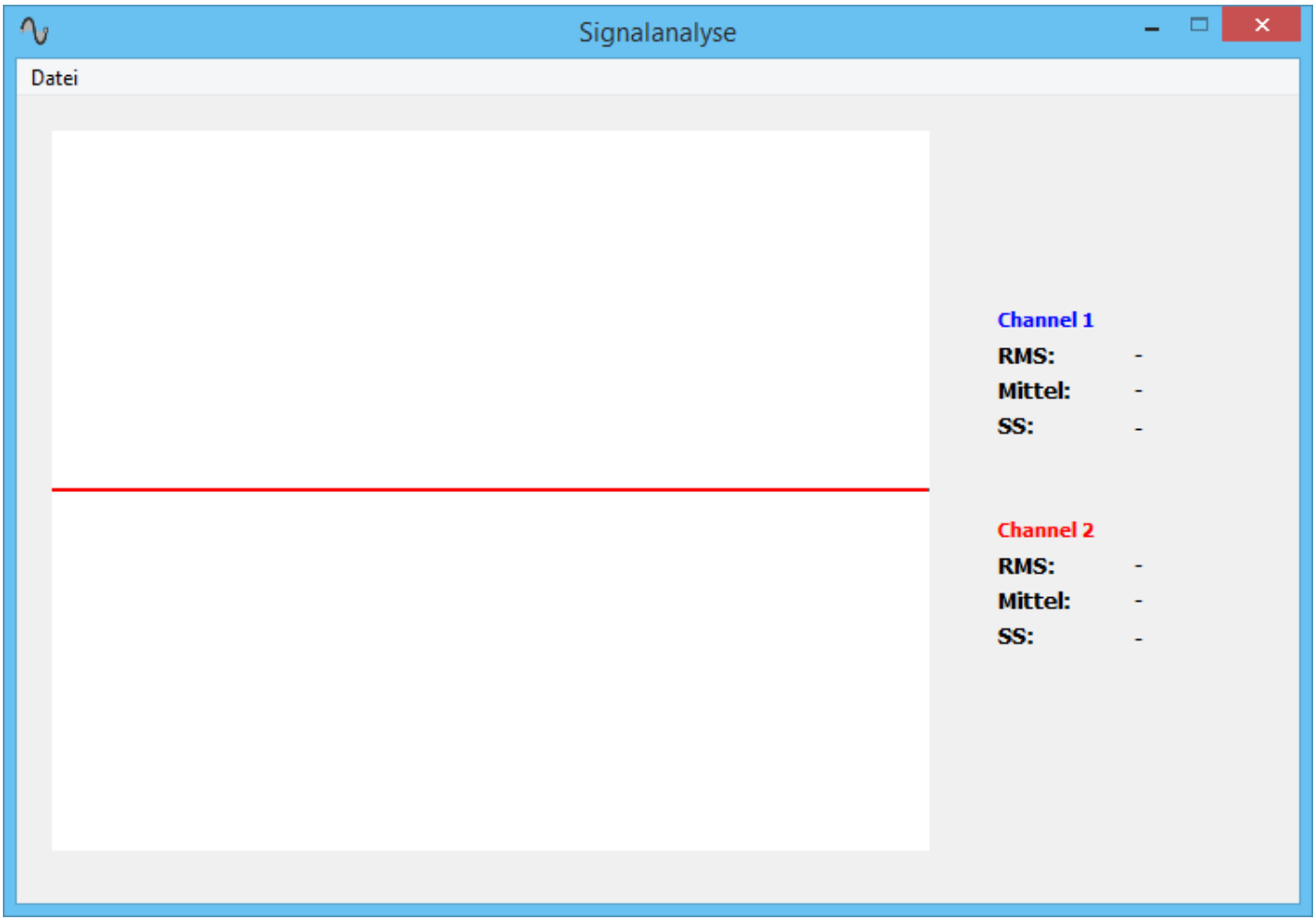
x-axis,1,2
second,Volt,Volt
-2.800000E-03,+2.226130597E+00,+1.594057798E+00
-2.797200E-03,+2.226130597E+00,+1.594057798E+00
-2.794400E-03,+2.301507481E+00,+1.594057798E+00
-2.791600E-03,+2.306532606E+00,+1.594057798E+00
-2.788800E-03,+2.306532606E+00,+1.594057798E+00
-2.786000E-03,+2.306532606E+00,+1.678479910E+00
...
...
...

Das Oszilloskope liefert CSV-Dateien mit ca. 2000 Werten. Es können aber beliebig viele Werte übermittelt werden, da das Array dynamisch alloziert wird.

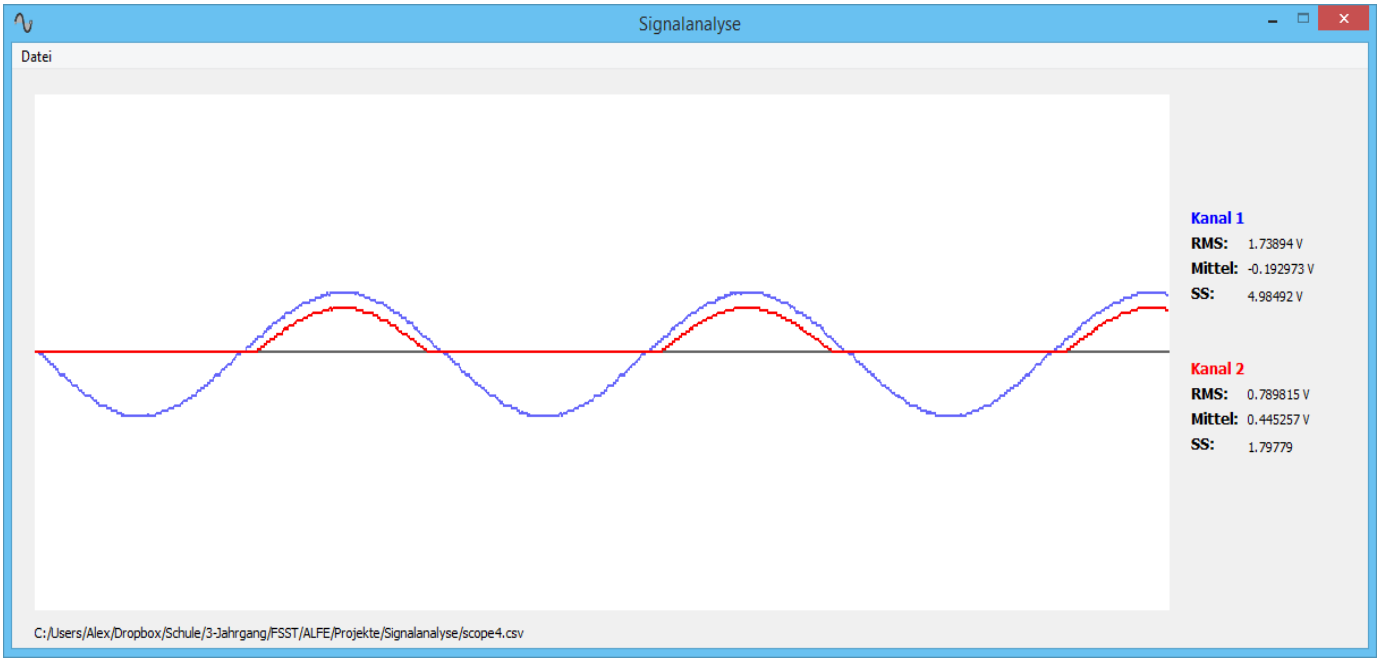


4 Benutzerhandbuch

Direkt nach dem Programmstart erscheint ein leeres Grafikfenster. Nach laden der CSV Datei über den Menüpunkt: „Datei -> Datei laden“ erscheint ein visualisiertes Signal Array inlusive aller Analysewerte (RMS, Mittel und Spitze-Spitze)



Je nach CSV File sind 1-2 Kanäle vorhanden. Kanal 1 wird in blau und Kanal 2 in rot dargestellt. Rechts außen werden Mittelwerte, RMS und Spitze-Spitze dargestellt. Unten wird der absolute Pfad zur CSV Datei eingeblendet.



5 Klassenübersicht

5.1 Klasse „grafik“

```
class grafik : public QWidget
{
protected:
    static const int N=1000;
    float samples[N];
    float samples2[N];

public:
    grafik(QWidget *w);
    virtual void paintEvent( QPaintEvent * );
    int getN()
    {  return N;
    }
    float* getsamples()
    {  return samples;
    }

    float* getsamples2()
    {  return samples2;
    }
};
```

5.2 Klasse „MainWindow“

```
class MainWindow : public QMainWindow
{
    Q_OBJECT

private:
    Ui::MainWindow *ui;

    int NMAX;

    void getAm();
    void getFre();
    void getPhi();
    void getSu();

public:
    explicit MainWindow(QWidget *parent = 0);
    ~MainWindow();

    QString filename;

    float am,fr,phi,su;
    float *samples;
    float *samples2;

    float array_max ( float a[], int n1 , int n2 );
    float array_min ( float a[], int n1 , int n2 );
    float array_mittel(float a[], int n1, int n2);
    float array_ss ( float a[], int n1 , int n2 );
    float array_rms ( float a[], int n1, int n2 );

private slots:
    void on_actionDatei_laden_triggered();

};
```