Subject: Re: Register a callback on "when window resized" ?
Posted by xrysf03 on Sun, 08 Dec 2019 13:12:24 GMT
View Forum Message <> Reply to Message

@slashupp: thanks for your response and for your care :)
Just for the record, the set_chart_grid() method goes like this:

```
void rtlsdr_skyline::set_chart_grid()
{
 if (display_in_calib_mode)
  return; // no modification needed

 if ((from_freq_rounded == 0) || (to_freq_rounded == 0))
  return;  // it's too early

 // grid spacing along axis X
 double x_tick = chart1_optimal_major_unit_x();

 // avoid unnecessary fractional digits in grid tick label,
 // by shifting the grid conveniently into alignment
 // with integer multiples of grid tick
 double x_ofs = (from_freq_rounded/1000000)
  - (ffloor(from_freq_rounded / 1000000 / x_tick) * x_tick);

 Chart1.SetXYMin(from_freq_rounded / 1000000, -100);
 Chart1.SetRange((to_freq_rounded - from_freq_rounded)/ 1000000, 100);
 Chart1.SetMinUnits(x_ofs,0);
 Chart1.SetMajorUnits(x_tick, 10);

 Chart1.SetLabels("MHz","dBFS");
 Chart1.ShowLegend(false);
}
```

...where chart1_optimal_major_unit_x() indeed polls the ScatterCtrl for its horizontal dimension in pixels, via GetPlotWidth() and calculates its output based on some "double" numbers in my own code.
It takes the frequency range, walks the exponential scale of a few decimal orders and looks for a nice spacing with a ratio of 1, 2 or 5 across the given range.

```
// the returned value is in MHz
double rtlsdr_skyline::chart1_optimal_major_unit_x()
{
 double retval = 1; // default response (1 MHz)
 double decimal_order = 0;
 int chart_horiz_pixels = Chart1.GetPlotWidth();
```

```
 if (chart_horiz_pixels < DIV_NO_SMALLER_THAN_PX)
  return retval; // prevent dv_by_zero. Return the default response.

 // we could also use "bandwidth" and "num_freqs" and whatnot...
 // but from_freq_rounded and to_freq_rounded are perhaps
 // the closest to the visual chart.
 double chart_bandwidth = (to_freq_rounded - from_freq_rounded) / 1000000;
 double mhz_per_optimal_div = chart_bandwidth
  / (chart_horiz_pixels / DIV_NO_SMALLER_THAN_PX);

 for (decimal_order = 0.001; decimal_order <= 1000; decimal_order *= 10)
 {
  if (decimal_order > mhz_per_optimal_div)
  {
   retval = decimal_order;
   break;
  }
  else if (decimal_order * 2 > mhz_per_optimal_div)
  {
   retval = decimal_order * 2;
   break;
  }
  else if (decimal_order * 5 > mhz_per_optimal_div)
  {
   retval = decimal_order * 5;
   break;
  }
  // else try another decimal order
 }

 return retval;
}
```

So it's really along the lines of what you suggest.

Nonetheless, I think I've caught wind of some other peculiarities in the inner workings of the GUI that I should pay attention to, with respect to flow of control, reentrancy of event handlers, calling Ctrl::ProcessEvents() etc. Vs. using threads etc. I'll start another topic on the subject.

Frank