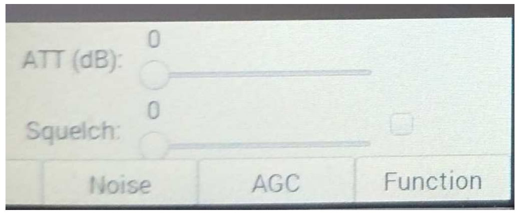
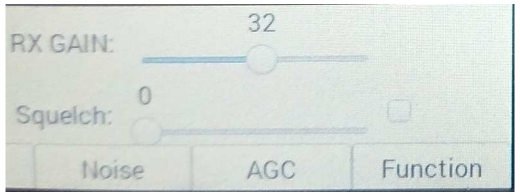
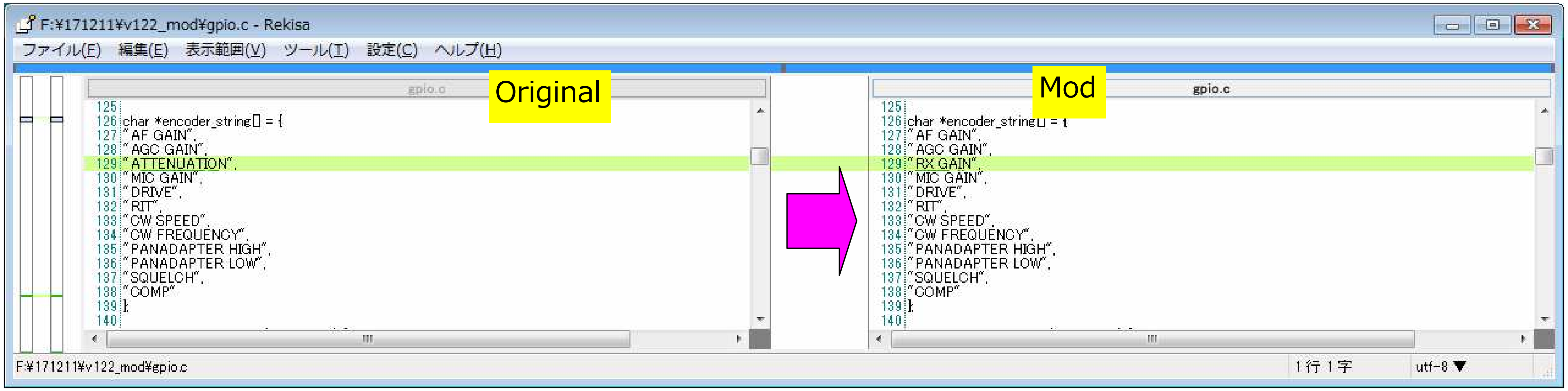
◆ Slider function : Attenuator → RX GAIN

10 Dec. 2017 JI1UDD





Source file : gpio.c (v1.2.2)


F:¥171211¥v122_mod¥gpio.c - Rekisa
ファイル(F) 編集(E) 表示範囲(V) ツール(T) 設定(C) ヘルプ(H)
gpio.c
gpio.c
1行 1字
utf-8 ▼

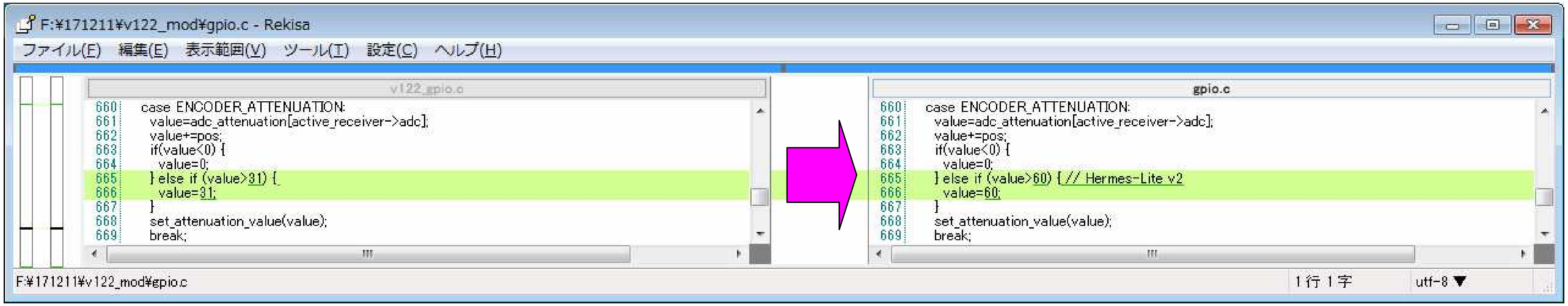
Original

```
char *encoder_string[] = {
"AF GAIN",
"AGC GAIN",
"ATTENUATION",
"MIC GAIN",
"DRIVE",
"RIT",
"CW SPEED",
"CW FREQUENCY",
"PANADAPTER HIGH",
"PANADAPTER LOW",
"SQUELCH",
"COMP"
};
```

Mod

```
char *encoder_string[] = {
"AF GAIN",
"AGC GAIN",
"RX GAIN",
"MIC GAIN",
"DRIVE",
"RIT",
"CW SPEED",
"CW FREQUENCY",
"PANADAPTER HIGH",
"PANADAPTER LOW",
"SQUELCH",
"COMP"
};
```

F:¥171211¥v122_mod¥gpio.c

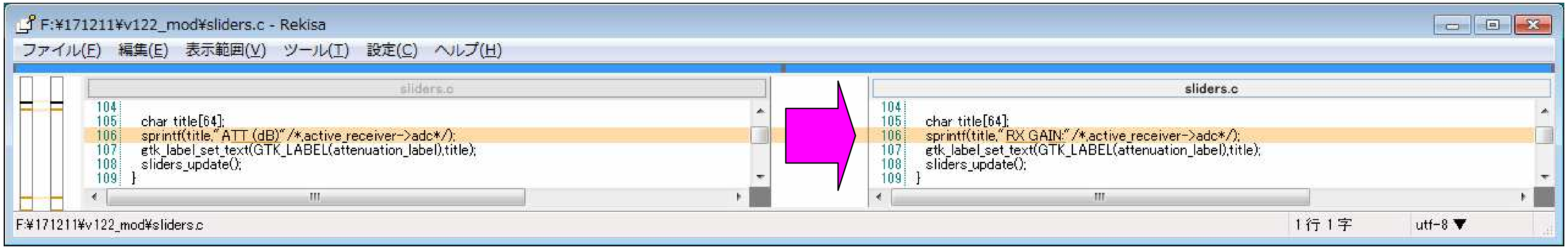

F:¥171211¥v122_mod¥gpio.c - Rekisa
ファイル(F) 編集(E) 表示範囲(V) ツール(T) 設定(C) ヘルプ(H)
v122_gpio.c
gpio.c
1行 1字
utf-8 ▼

```
    case ENCODER_ATTENUATION:
      value=adc_attenuation[active_receiver->adc];
      value+=pos;
      if(value<0) {
        value=0;
      } else if (value>31) {
        value=31;
      }
      set_attenuation_value(value);
      break;
```

```
    case ENCODER_ATTENUATION:
      value=adc_attenuation[active_receiver->adc];
      value+=pos;
      if(value<0) {
        value=0;
      } else if (value>60) { // Hermes-Lite v2
        value=60;
      }
      set_attenuation_value(value);
      break;
```

F:¥171211¥v122_mod¥gpio.c

Source file : sliders.c (v1.2.2)



F:¥171211¥v122_mod¥sliders.c - Rekisa
ファイル(F) 編集(E) 表示範囲(V) ツール(T) 設定(C) ヘルプ(H)

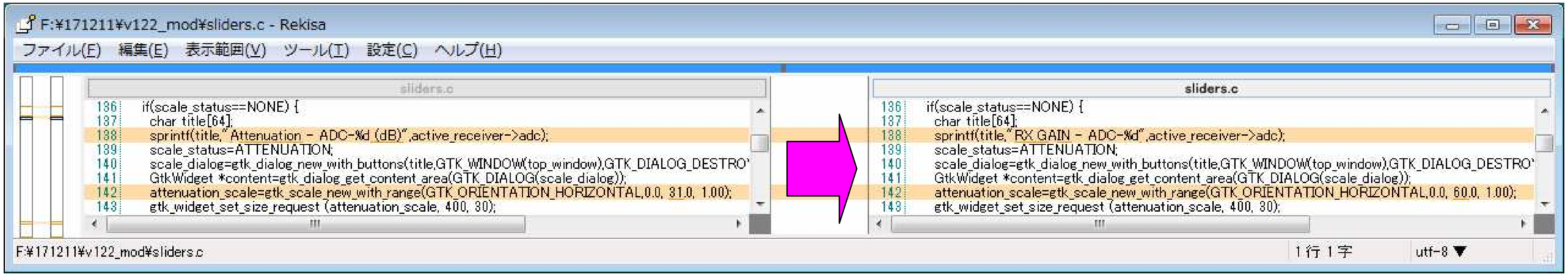
sliders.c

```c

  char title[64];
  sprintf(title,"ATT (dB)"/*,active_receiver->adc*/);
  gtk_label_set_text(GTK_LABEL(attenuation_label),title);
  sliders_update();
}
```

sliders.c

```c

  char title[64];
  sprintf(title,"RX GAIN:"/*,active_receiver->adc*/);
  gtk_label_set_text(GTK_LABEL(attenuation_label),title);
  sliders_update();
}
```

F:¥171211¥v122_mod¥sliders.c



F:¥171211¥v122_mod¥sliders.c - Rekisa
ファイル(F) 編集(E) 表示範囲(V) ツール(T) 設定(C) ヘルプ(H)

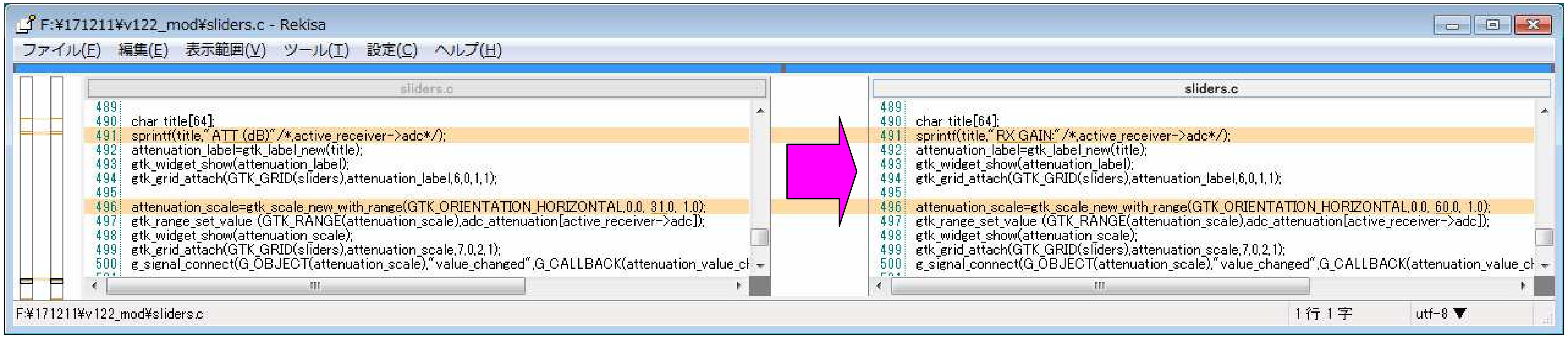
sliders.c

```c
  if(scale_status==NONE) {
    char title[64];
    sprintf(title,"Attenuation - ADC-%d (dB)",active_receiver->adc);
    scale_status=ATTENUATION;
    scale_dialog=gtk_dialog_new_with_buttons(title,GTK_WINDOW(top_window),GTK_DIALOG_DESTRO
    GtkWidget *content=gtk_dialog_get_content_area(GTK_DIALOG(scale_dialog));
    attenuation_scale=gtk_scale_new_with_range(GTK_ORIENTATION_HORIZONTAL,0.0, 31.0, 1.00);
    gtk_widget_set_size_request (attenuation_scale, 400, 30);
```

sliders.c

```c
  if(scale_status==NONE) {
    char title[64];
    sprintf(title,"RX GAIN - ADC-%d",active_receiver->adc);
    scale_status=ATTENUATION;
    scale_dialog=gtk_dialog_new_with_buttons(title,GTK_WINDOW(top_window),GTK_DIALOG_DESTRO
    GtkWidget *content=gtk_dialog_get_content_area(GTK_DIALOG(scale_dialog));
    attenuation_scale=gtk_scale_new_with_range(GTK_ORIENTATION_HORIZONTAL,0.0, 60.0, 1.00);
    gtk_widget_set_size_request (attenuation_scale, 400, 30);
```

F:¥171211¥v122_mod¥sliders.c



F:¥171211¥v122_mod¥sliders.c - Rekisa
ファイル(F) 編集(E) 表示範囲(V) ツール(T) 設定(C) ヘルプ(H)

sliders.c

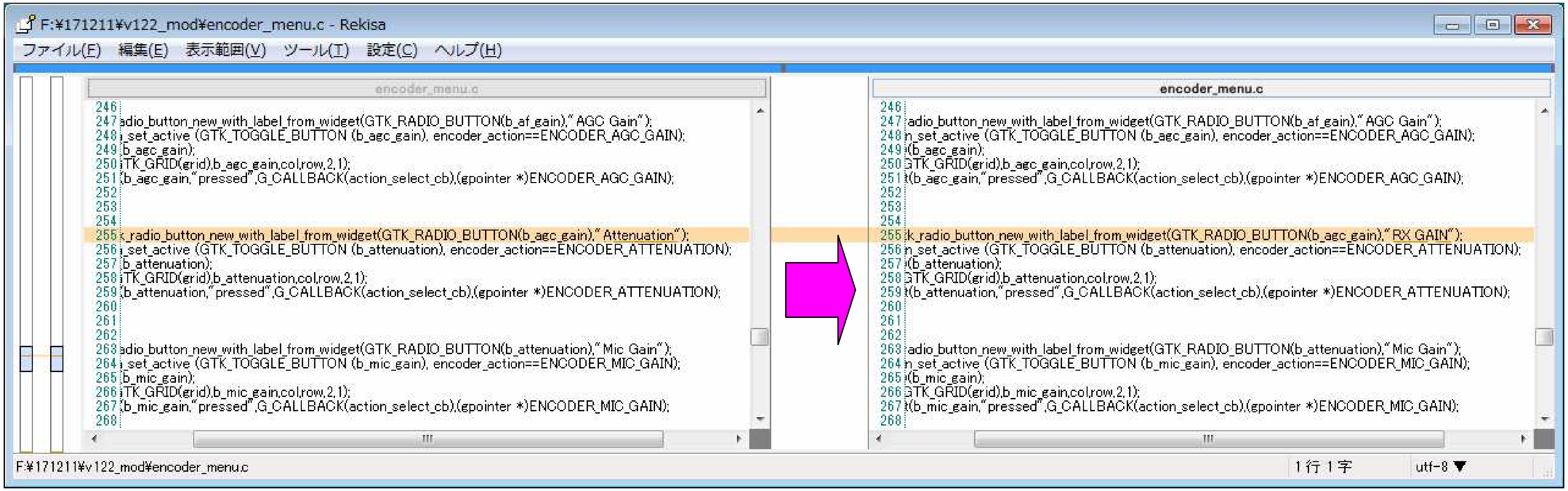
```c

char title[64];
sprintf(title,"ATT (dB)"/*,active_receiver->adc*/);
attenuation_label=gtk_label_new(title);
gtk_widget_show(attenuation_label);
gtk_grid_attach(GTK_GRID(sliders),attenuation_label,6,0,1,1);

attenuation_scale=gtk_scale_new_with_range(GTK_ORIENTATION_HORIZONTAL,0.0, 31.0, 1.0);
gtk_range_set_value (GTK_RANGE(attenuation_scale),adc_attenuation[active_receiver->adc]);
gtk_widget_show(attenuation_scale);
gtk_grid_attach(GTK_GRID(sliders),attenuation_scale,7,0,2,1);
g_signal_connect(G_OBJECT(attenuation_scale),"value_changed",G_CALLBACK(attenuation_value_cl
```

sliders.c

```c

char title[64];
sprintf(title,"RX GAIN:"/*,active_receiver->adc*/);
attenuation_label=gtk_label_new(title);
gtk_widget_show(attenuation_label);
gtk_grid_attach(GTK_GRID(sliders),attenuation_label,6,0,1,1);

attenuation_scale=gtk_scale_new_with_range(GTK_ORIENTATION_HORIZONTAL,0.0, 60.0, 1.0);
gtk_range_set_value (GTK_RANGE(attenuation_scale),adc_attenuation[active_receiver->adc]);
gtk_widget_show(attenuation_scale);
gtk_grid_attach(GTK_GRID(sliders),attenuation_scale,7,0,2,1);
g_signal_connect(G_OBJECT(attenuation_scale),"value_changed",G_CALLBACK(attenuation_value_cl
```

F:¥171211¥v122_mod¥sliders.c

Source file：encoder_menu.c (v1.2.2)



F:¥171211¥v122_mod¥encoder_menu.c - Rekisa

encoder_menu.c

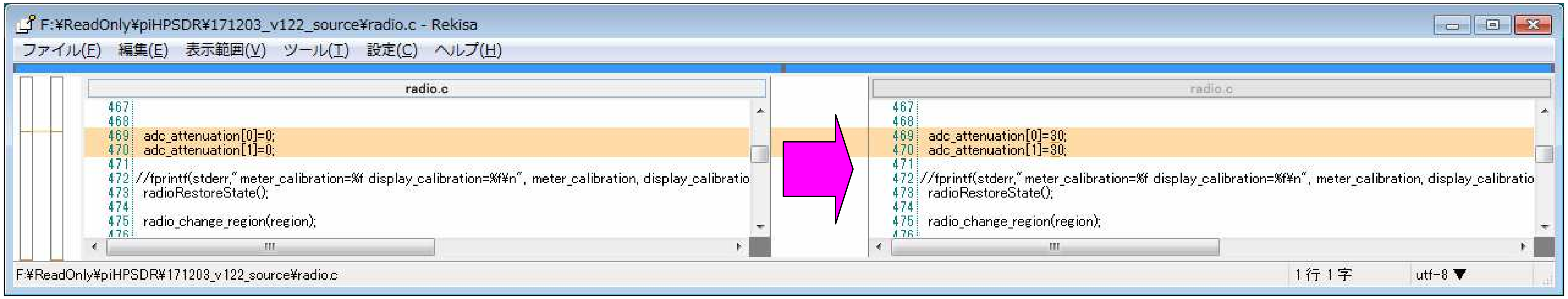
```
adio_button_new_with_label_from_widget(GTK_RADIO_BUTTON(b_af_gain),"AGC Gain");
_set_active (GTK_TOGGLE_BUTTON (b_agc_gain), encoder_action==ENCODER_AGC_GAIN);
b_agc_gain);
TK_GRID(grid),b_agc_gain,col,row,2,1);
(b_agc_gain,"pressed",G_CALLBACK(action_select_cb),(gpointer *)ENCODER_AGC_GAIN);



_radio_button_new_with_label_from_widget(GTK_RADIO_BUTTON(b_agc_gain),"Attenuation");
_set_active (GTK_TOGGLE_BUTTON (b_attenuation), encoder_action==ENCODER_ATTENUATION);
b_attenuation);
TK_GRID(grid),b_attenuation,col,row,2,1);
(b_attenuation,"pressed",G_CALLBACK(action_select_cb),(gpointer *)ENCODER_ATTENUATION);



adio_button_new_with_label_from_widget(GTK_RADIO_BUTTON(b_attenuation),"Mic Gain");
_set_active (GTK_TOGGLE_BUTTON (b_mic_gain), encoder_action==ENCODER_MIC_GAIN);
b_mic_gain);
TK_GRID(grid),b_mic_gain,col,row,2,1);
(b_mic_gain,"pressed",G_CALLBACK(action_select_cb),(gpointer *)ENCODER_MIC_GAIN);
```

encoder_menu.c

```
adio_button_new_with_label_from_widget(GTK_RADIO_BUTTON(b_af_gain),"AGC Gain");
n_set_active (GTK_TOGGLE_BUTTON (b_agc_gain), encoder_action==ENCODER_AGC_GAIN);
(b_agc_gain);
GTK_GRID(grid),b_agc_gain,col,row,2,1);
t(b_agc_gain,"pressed",G_CALLBACK(action_select_cb),(gpointer *)ENCODER_AGC_GAIN);



k_radio_button_new_with_label_from_widget(GTK_RADIO_BUTTON(b_agc_gain),"RX GAIN");
n_set_active (GTK_TOGGLE_BUTTON (b_attenuation), encoder_action==ENCODER_ATTENUATION);
(b_attenuation);
GTK_GRID(grid),b_attenuation,col,row,2,1);
t(b_attenuation,"pressed",G_CALLBACK(action_select_cb),(gpointer *)ENCODER_ATTENUATION);



adio_button_new_with_label_from_widget(GTK_RADIO_BUTTON(b_attenuation),"Mic Gain");
n_set_active (GTK_TOGGLE_BUTTON (b_mic_gain), encoder_action==ENCODER_MIC_GAIN);
(b_mic_gain);
GTK_GRID(grid),b_mic_gain,col,row,2,1);
t(b_mic_gain,"pressed",G_CALLBACK(action_select_cb),(gpointer *)ENCODER_MIC_GAIN);
```

F:¥171211¥v122_mod¥encoder_menu.c　1行 1字　utf-8 ▼

Source file：radio.c (v1.2.2)



F:¥ReadOnly¥piHPSDR¥171203_v122_source¥radio.c - Rekisa

radio.c

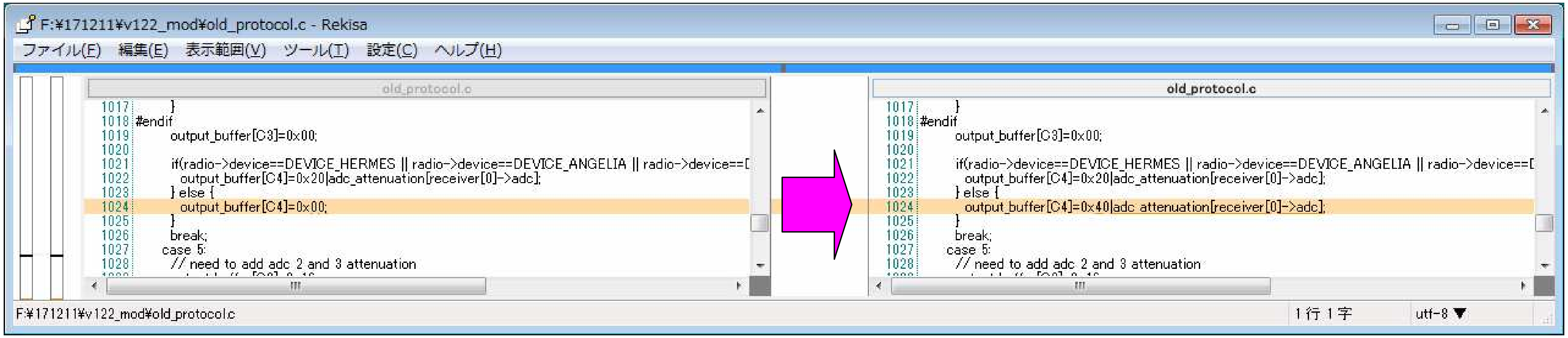
```

  adc_attenuation[0]=0;
  adc_attenuation[1]=0;

//fprintf(stderr," meter_calibration=%f display_calibration=%f¥n", meter_calibration, display_calibratio
  radioRestoreState();

  radio_change_region(region);
```

radio.c

```

  adc_attenuation[0]=30;
  adc_attenuation[1]=30;

//fprintf(stderr," meter_calibration=%f display_calibration=%f¥n", meter_calibration, display_calibratio
  radioRestoreState();

  radio_change_region(region);
```

F:¥ReadOnly¥piHPSDR¥171203_v122_source¥radio.c　1行 1字　utf-8 ▼

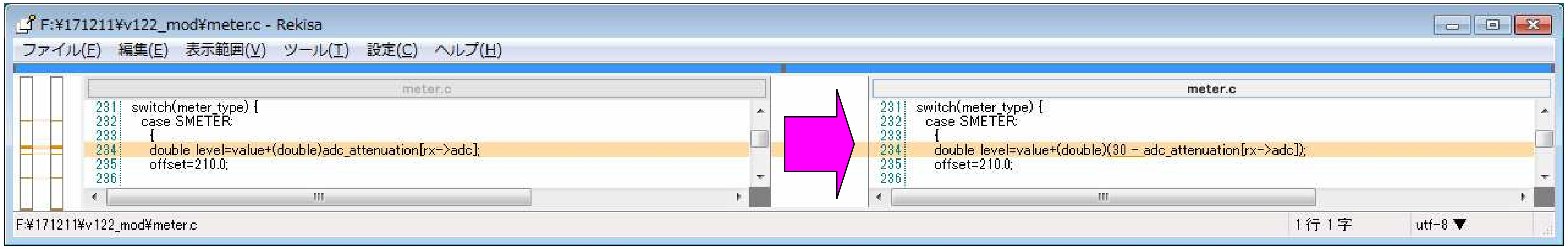
Source file：old_protocol.c (v1.2.2)


F:¥171211¥v122_mod¥old_protocol.c - Rekisa
ファイル(F) 編集(E) 表示範囲(V) ツール(T) 設定(C) ヘルプ(H)
old_protocol.c
old_protocol.c
F:¥171211¥v122_mod¥old_protocol.c
1行 1字
utf-8 ▼

```
        }
#endif
        output_buffer[C3]=0x00;

        if(radio->device==DEVICE_HERMES || radio->device==DEVICE_ANGELIA || radio->device==[
          output_buffer[C4]=0x20|adc_attenuation[receiver[0]->adc];
        } else {
          output_buffer[C4]=0x00;
        }
        break;
      case 5:
        // need to add adc 2 and 3 attenuation
```

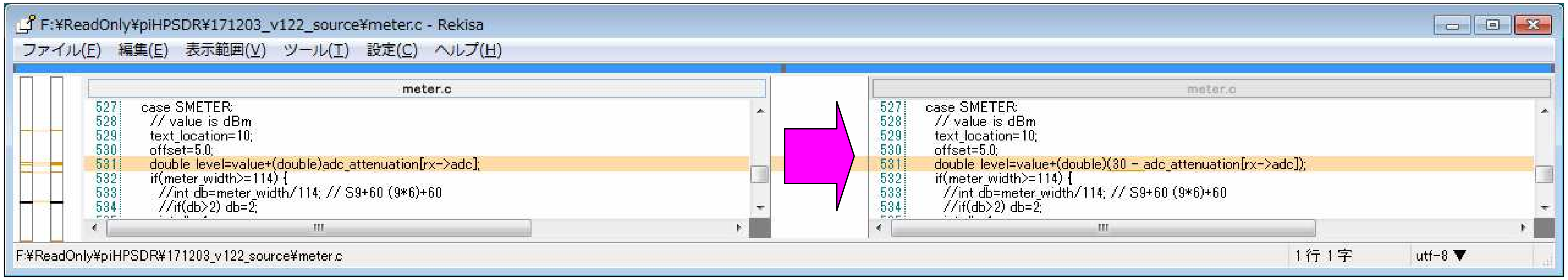
```
        }
#endif
        output_buffer[C3]=0x00;

        if(radio->device==DEVICE_HERMES || radio->device==DEVICE_ANGELIA || radio->device==[
          output_buffer[C4]=0x20|adc_attenuation[receiver[0]->adc];
        } else {
          output_buffer[C4]=0x40|adc_attenuation[receiver[0]->adc];
        }
        break;
      case 5:
        // need to add adc 2 and 3 attenuation
```

Source file：meter.c (v1.2.2)


F:¥171211¥v122_mod¥meter.c - Rekisa
ファイル(F) 編集(E) 表示範囲(V) ツール(T) 設定(C) ヘルプ(H)
meter.c
meter.c
F:¥171211¥v122_mod¥meter.c
1行 1字
utf-8 ▼

```
  switch(meter_type) {
    case SMETER:
      {
      double level=value+(double)adc_attenuation[rx->adc];
      offset=210.0;

```

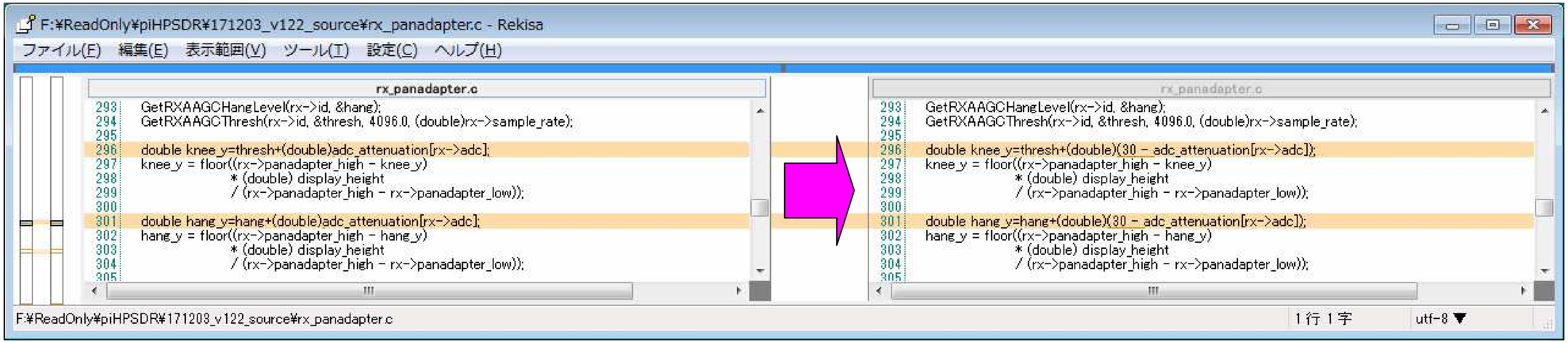
```
  switch(meter_type) {
    case SMETER:
      {
      double level=value+(double)(30 - adc_attenuation[rx->adc]);
      offset=210.0;

```


F:¥ReadOnly¥piHPSDR¥171203_v122_source¥meter.c - Rekisa
ファイル(F) 編集(E) 表示範囲(V) ツール(T) 設定(C) ヘルプ(H)
meter.c
meter.c
F:¥ReadOnly¥piHPSDR¥171203_v122_source¥meter.c
1行 1字
utf-8 ▼

```
    case SMETER:
      // value is dBm
      text_location=10;
      offset=5.0;
      double level=value+(double)adc_attenuation[rx->adc];
      if(meter_width>=114) {
        //int db=meter_width/114; // S9+60 (9*6)+60
        //if(db>2) db=2;
```

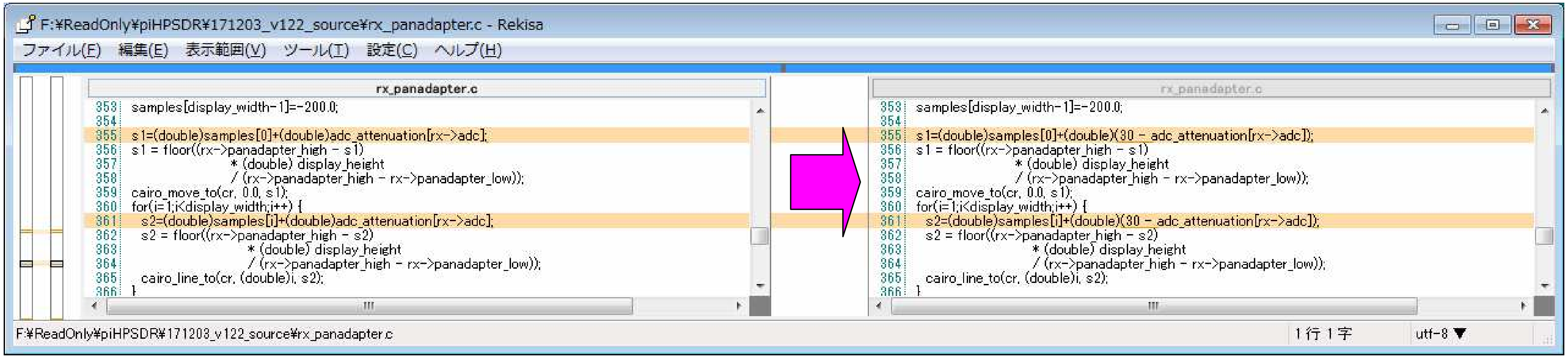
```
    case SMETER:
      // value is dBm
      text_location=10;
      offset=5.0;
      double level=value+(double)(30 - adc_attenuation[rx->adc]);
      if(meter_width>=114) {
        //int db=meter_width/114; // S9+60 (9*6)+60
        //if(db>2) db=2;
```

Source file：rx_panadapter.c (v1.2.2)


F:¥ReadOnly¥piHPSDR¥171203_v122_source¥rx_panadapter.c - Rekisa
ファイル(F) 編集(E) 表示範囲(V) ツール(T) 設定(C) ヘルプ(H)
rx_panadapter.c
rx_panadapter.c
F:¥ReadOnly¥piHPSDR¥171203_v122_source¥rx_panadapter.c
1行 1字
utf-8 ▼

```
  GetRXAAGCHangLevel(rx->id, &hang);
  GetRXAAGCThresh(rx->id, &thresh, 4096.0, (double)rx->sample_rate);

  double knee_y=thresh+(double)adc_attenuation[rx->adc];
  knee_y = floor((rx->panadapter_high - knee_y)
                  * (double) display_height
                  / (rx->panadapter_high - rx->panadapter_low));

  double hang_y=hang+(double)adc_attenuation[rx->adc];
  hang_y = floor((rx->panadapter_high - hang_y)
                  * (double) display_height
                  / (rx->panadapter_high - rx->panadapter_low));
```

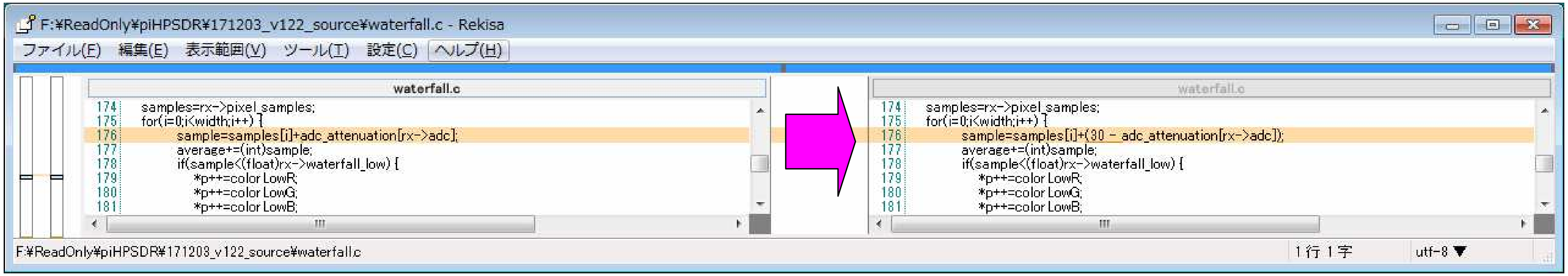
```
  GetRXAAGCHangLevel(rx->id, &hang);
  GetRXAAGCThresh(rx->id, &thresh, 4096.0, (double)rx->sample_rate);

  double knee_y=thresh+(double)(30 - adc_attenuation[rx->adc]);
  knee_y = floor((rx->panadapter_high - knee_y)
                  * (double) display_height
                  / (rx->panadapter_high - rx->panadapter_low));

  double hang_y=hang+(double)(30 - adc_attenuation[rx->adc]);
  hang_y = floor((rx->panadapter_high - hang_y)
                  * (double) display_height
                  / (rx->panadapter_high - rx->panadapter_low));
```


F:¥ReadOnly¥piHPSDR¥171203_v122_source¥rx_panadapter.c - Rekisa
ファイル(F) 編集(E) 表示範囲(V) ツール(T) 設定(C) ヘルプ(H)
rx_panadapter.c
rx_panadapter.c
F:¥ReadOnly¥piHPSDR¥171203_v122_source¥rx_panadapter.c
1行 1字
utf-8 ▼

```
samples[display_width-1]=-200.0;

s1=(double)samples[0]+(double)adc_attenuation[rx->adc];
s1 = floor((rx->panadapter_high - s1)
                * (double) display_height
                / (rx->panadapter_high - rx->panadapter_low));
cairo_move_to(cr, 0.0, s1);
for(i=1;i<display_width;i++) {
  s2=(double)samples[i]+(double)adc_attenuation[rx->adc];
  s2 = floor((rx->panadapter_high - s2)
                  * (double) display_height
                  / (rx->panadapter_high - rx->panadapter_low));
  cairo_line_to(cr, (double)i, s2);
}
```

```
samples[display_width-1]=-200.0;

s1=(double)samples[0]+(double)(30 - adc_attenuation[rx->adc]);
s1 = floor((rx->panadapter_high - s1)
                * (double) display_height
                / (rx->panadapter_high - rx->panadapter_low));
cairo_move_to(cr, 0.0, s1);
for(i=1;i<display_width;i++) {
  s2=(double)samples[i]+(double)(30 - adc_attenuation[rx->adc]);
  s2 = floor((rx->panadapter_high - s2)
                  * (double) display_height
                  / (rx->panadapter_high - rx->panadapter_low));
  cairo_line_to(cr, (double)i, s2);
}
```

Source file：waterfall.c (v1.2.2)


F:¥ReadOnly¥piHPSDR¥171203_v122_source¥waterfall.c - Rekisa
ファイル(F) 編集(E) 表示範囲(V) ツール(T) 設定(C) ヘルプ(H)
waterfall.c
waterfall.c
F:¥ReadOnly¥piHPSDR¥171203_v122_source¥waterfall.c
1行 1字
utf-8 ▼

```
  samples=rx->pixel_samples;
  for(i=0;i<width;i++) {
        sample=samples[i]+adc_attenuation[rx->adc];
        average+=(int)sample;
        if(sample<(float)rx->waterfall_low) {
            *p++=colorLowR;
            *p++=colorLowG;
            *p++=colorLowB;
```

```
  samples=rx->pixel_samples;
  for(i=0;i<width;i++) {
        sample=samples[i]+(30 - adc_attenuation[rx->adc]);
        average+=(int)sample;
        if(sample<(float)rx->waterfall_low) {
            *p++=colorLowR;
            *p++=colorLowG;
            *p++=colorLowB;
```