

```
//@version=3
```

```
study(title="SerkanEdge", overlay=true)
```

```
// Source
```

```
src = input(defval=close, title="Source")
```

```
// Sampling Period
```

```
// Settings for 5min chart, BTCUSDC. For Other coin, change the paremeters
```

```
per = input(defval=100, minval=1, title="Sampling Period")
```

```
// Range Multiplier
```

```
mult = input(defval=3.0, minval=0.1, title="Range Multiplier")
```

```
// Smooth Average Range
```

```
smoothrng(x, t, m)=>
```

```
  wper  = (t*2) - 1
```

```
  avrng  = ema(abs(x - x[1]), t)
```

```
  smoothrng = ema(avrng, wper)*m
```

```
  smoothrng
```

```
smrng = smoothrng(src, per, mult)
```

```
// Range Filter
```

```
rngfilt(x, r)=>
```

```
  rngfilt = x
```

```

    rngfilt := x > nz(rngfilt[1]) ? ((x - r) < nz(rngfilt[1]) ? nz(rngfilt[1]) : (x - r)) : ((x + r) > nz(rngfilt[1]) ?
nz(rngfilt[1]) : (x + r))
    rngfilt
    filt = rngfilt(src, smrng)

// Filter Direction

upward = 0.0
upward := filt > filt[1] ? nz(upward[1]) + 1 : filt < filt[1] ? 0 : nz(upward[1])
downward = 0.0
downward := filt < filt[1] ? nz(downward[1]) + 1 : filt > filt[1] ? 0 : nz(downward[1])

// Target Bands

hband = filt + smrng
lband = filt - smrng

// Colors

filtcolor = upward > 0 ? lime : downward > 0 ? red : orange
barcolor = (src > filt) and (src > src[1]) and (upward > 0) ? lime : (src > filt) and (src < src[1]) and
(upward > 0) ? green :
    (src < filt) and (src < src[1]) and (downward > 0) ? red : (src < filt) and (src > src[1]) and (downward >
0) ? maroon : orange

filtplot = plot(filt, color=filtcolor, linewidth=3, title="Range Filter")

// Target

hbandplot = plot(hband, color=aqua, transp=100, title="High Target")
lbandplot = plot(lband, color=fuchsia, transp=100, title="Low Target")

```

```
// Fills
```

```
fill(hbandplot, filtplot, color=aqua, title="High Target Range")
```

```
fill(lbandplot, filtplot, color=fuchsia, title="Low Target Range")
```

```
// Bar Color
```

```
barcolor(barcolor)
```

```
// Break Outs
```

```
longCond = na
```

```
shortCond = na
```

```
longCond := ((src > filt) and (src > src[1]) and (upward > 0)) or ((src > filt) and (src < src[1]) and  
(upward > 0))
```

```
shortCond := ((src < filt) and (src < src[1]) and (downward > 0)) or ((src < filt) and (src > src[1]) and  
(downward > 0))
```

```
CondIni = 0
```

```
CondIni := longCond ? 1 : shortCond ? -1 : CondIni[1]
```

```
longCondition = longCond and CondIni[1] == -1
```

```
shortCondition = shortCond and CondIni[1] == 1
```

```
//Alerts
```

```
plotshape(longCondition, title = "Buy Signal", text = "Long", textcolor = white, style=shape.labelup,  
size = size.small, location=location.belowbar, color = green, transp = 0)
```

```
plotshape(shortCondition, title = "Sell Signal", text = "Short", textcolor = white,  
style=shape.labeldown, size = size.small, location=location.abovebar, color = red, transp = 0)
```

```
alertcondition(longCondition, title="Buy Alert", message = "Short")
```

```
alertcondition(longCondition, title="Buy Alert", message = "Short")
```

alertcondition(longCondition, title="Buy Alert", message = "Short")

alertcondition(shortCondition, title="Sell Alert", message = "Long")