

```

#
# Study: FW_SnapBack
#
# Copyright 2009,2010 FirstwaveDevelopersGroup
# http://groups.google.com/group/firstwave-developers
# All rights reserved.
# This code is for the exclusive and individual use of members of the
# David Elliott chat room on ToS. It may not be posted or shared
# without written permission.
#
# Written by ToS User: davidinc (studiessupport@pc3i.com)
# See revision history and embedded comments for the names of other
# contributors.
# If you find any problems please let me know and refer to the
# revision number and date. Thanks.
#
# User Notes:
# Stoch Modes:
# All:
#   Stochastics: white short dashed (2,1); white solid (2,2);
#   yellow solid (5,3); green solid (8,5); red solid (17,5);
#   cyan solid (48,3).
#   RSIs: dark_orange solid (RSI_EMA, 2 prd); magenta solid
#   (RSI_SMA, 2 prd).
#
# SOAP:
#   Stochastics: yellow solid (5,3), green solid (8,5).
#
# SuperSOAP:
#   Stochastics: yellow solid (5,3); green solid (8,5) and turquoise solid
#   (12,3).
#
# STAMP:
#   Stochastics: yellow solid (5,3); green solid (8,5); red solid (17,5).
#
# Revision history:
# Rev Date      Change Description
# -----
# 00 2009-01-06 Original
# 01 2009-01-12 Updated RSI section with code provided by kmktroy.
# 02 2009-01-12 Updated RSI section with code provided by Sidewinder747
# 03           unpublished.
# 04 2009-02-19 Add horizontal reference line at 50.
# 05 2009-04-07 Code cleanup. Added SOAP only option.
# 06           unpublished.
# 07 2009-06-05 Added the SuperSOAP and STAMP indicators FW describes in his
#               Advanced Equity Training CDs. These options are turned off
#               by default since David doesn't use them in his screenshow
#               on Tos.
#               Added SOAPwRSI mode.
# 08 2009-09-11 Added UserDefined stochastic.
#               Added RSIOonly mode.
#               Changed default mode to SOAP.
# 09 2010-07-15 Rearranged code to improve performance.
#               Removed Exponential smoothing option for stochastics.
#               Set defaults for User-Defined stochastic to 34,2 (cyan).
#               Added a RSI minus Stoch08x5 breadth plot & flags.
#               Added a RSI minus StochUDef breadth plot & flags.

```

```
# Adjusted the a vertical displacement input so that
# stochastics, RSI and breadth can oscillate around 0 or 50.
# Added the arrows from the SOAPCounty study by Managematics.
```

```
declare lower;
```

```
#hint Stoch_Mode: The mode names correspond to FirstWave's usage in his CDs.
#hint UDef_Stoch: Use 'yes' to display the user-defined stochastic.
#hint UDef_FastK: The user-defined stochastic FastK setting.
#hint UDef_Slowing: The user-defined stochastic Slowing period setting.
#hint RSI_Mode: Yes to turn the RSI plot on. No to turn it off.
#hint showBreadth_RSI_Stoch8x5: Yes to plot the breadth (difference) between RSI and the 8,5 stochastic.
#hint showFlags_RSI_Stoch8x5: Yes to plot the limes. The limes display when breadth (difference) between RSI and the
8,5 stochastic exceeds the FlagTriggerLevel (sensitivy). This is the same plot as showFlags_RSI_Stoch8x5. When the
FlagTriggerLevel is exceeded (limes are displayed) there is high probability of SnapBack.
#hint showBreadth_RSI_StochUDef: Yes to plot the breadth (difference) between RSI and the user-defined stochastic.
#hint showFlags_RSI_StochUDef: Yes to plot the blueberries. The blueberries display when breadth (difference) between
RSI and the user-defined stochastic exceeds the FlagTriggerLevel (sensitivy). This is the same plot as
showFlags_RSI_StochUDEF. When the FlagTriggerLevel is exceeded there is high probability of SnapBack.
#hint FlagTriggerLevel: Adjust sensitivity of flags. Larger numbers are less sensitive and give fewer signals.
#hint arrow_location: Used to vertically compress the graph so that arrows are more fully displayed. Pick one that works
for you.
#hint VertDisp: Use -50 (default) to oscillate around 0, use 0 to oscillate around 50 (standard stochastic setting). This
adjustment is necessary as stochastics oscillate around 50 (bounded by 0 and 100) while breadth (SnapBacks) oscillate
around 0 (bounded by -200 and +200).
```

```
input Stoch_Mode      = {All, default SOAP, SuperSOAP, STAMP};
input UDef_Stoch      = {default "Yes", "No"};
input UDef_FastK      = 34;
input UDef_Slowing    = 2;
input RSI_Mode        = {default "Yes", "No"};
input RSI_Length      = 2;
input showBreadth_RSI_Stoch8x5 = no;
input showFlags_RSI_Stoch8x5  = Yes;
input showBreadth_RSI_StochUDef = no;
input showFlags_RSI_StochUDef  = Yes;
input FlagTriggerLevel    = 45;
input arrow_location = {default "Partially_Onscreen", "Always_Onscreen",
    "Possibly_Offscreen", "On_CenterLine", "Inside_Lines",
    "OBOS_Lines"};
input VertDisp         = -50;
```

```
def priceH = high;
def priceL = low;
def priceC = close;
```

```
# plot the horizontal reference lines
plot Line20 = 20 + VertDisp;
Line20.SetDefaultColor(Color.Light_Red);
Line20.HideBubble();
```

```
plot Line50 = 50 + VertDisp;
Line50.SetDefaultColor(Color.White);
Line50.HideBubble();
```

```
plot Line80 = 80 + VertDisp;
Line80.SetDefaultColor(Color.Light_Red);
Line80.HideBubble();
```

```

plot TriggerUp = if showBreadth_RSI_Stoch8x5 or showBreadth_RSI_StochUDef then
  FlagTriggerLevel + 50 + VertDisp else double.nan;
TriggerUp.SetDefaultColor(Color.Gray);
TriggerUp.SetStyle(Curve.Short_Dash);
TriggerUp.HideBubble();

plot TriggerDn = if showbreadth_RSI_Stoch8x5 or showBreadth_RSI_StochUDef then
  - FlagTriggerLevel + 50 + VertDisp else double.nan;
TriggerDn.SetDefaultColor(Color.Gray);
TriggerDn.SetStyle(Curve.Short_Dash);
TriggerDn.HideBubble();

# plot the fast stochastics (%K)
plot K_UDef;
plot K_48x3;
plot K_17x5;
plot K_12x3;
plot K_08x5;
plot K_05x3;
plot K_02x2;
plot K_02x1;

switch (Stoch_Mode) {
case All:
  K_02x1 = StochasticFull(KPeriod = 2, slowing_period = 1) + VertDisp;
  K_02x2 = StochasticFull(KPeriod = 2, slowing_period = 2) + VertDisp;
  K_05x3 = StochasticFull(KPeriod = 5, slowing_period = 3) + VertDisp;
  K_08x5 = StochasticFull(KPeriod = 8, slowing_period = 5) + VertDisp;
  K_12x3 = Double.nan;
  K_17x5 = StochasticFull(KPeriod = 17, slowing_period = 5) + VertDisp;
  K_48x3 = StochasticFull(KPeriod = 48, slowing_period = 3) + VertDisp;
case SOAP:
  K_02x1 = Double.nan;
  K_02x2 = Double.nan;
  K_05x3 = StochasticFull(KPeriod = 5, slowing_period = 3) + VertDisp;
  K_08x5 = StochasticFull(KPeriod = 8, slowing_period = 5) + VertDisp;
  K_12x3 = Double.nan;
  K_17x5 = Double.nan;
  K_48x3 = Double.nan;
case SuperSOAP:
  K_02x1 = Double.nan;
  K_02x2 = Double.nan;
  K_05x3 = StochasticFull(KPeriod = 5, slowing_period = 3) + VertDisp;
  K_08x5 = StochasticFull(KPeriod = 8, slowing_period = 5) + VertDisp;
  K_12x3 = StochasticFull(KPeriod = 12, slowing_period = 3) + VertDisp;
  K_17x5 = Double.nan;
  K_48x3 = Double.nan;
case STAMP:
  K_02x1 = Double.nan;
  K_02x2 = Double.nan;
  K_05x3 = StochasticFull(KPeriod = 5, slowing_period = 3) + VertDisp;
  K_08x5 = StochasticFull(KPeriod = 8, slowing_period = 5) + VertDisp;
  K_12x3 = Double.nan;
  K_17x5 = StochasticFull(KPeriod = 17, slowing_period = 5) + VertDisp;
  K_48x3 = Double.nan;
} #end of switch (Mode)
K_02x1.SetDefaultColor(Color.White);

```

```

K_02x1.SetLineWeight(1);
K_02x1.SetStyle(Curve.Short_Dash);
K_02x2.SetDefaultColor(Color.White);
K_02x2.SetLineWeight(1);
K_05x3.SetDefaultColor(Color.Yellow);
K_05x3.SetLineWeight(1);
K_08x5.SetDefaultColor(Color.Green);
K_08x5.SetLineWeight(1);
K_12x3.SetDefaultColor(Color.Red);
K_12x3.SetLineWeight(1);
K_12x3.SetDefaultColor(CreateColor(0, 154, 104));
K_12x3.SetLineWeight(1);
K_17x5.SetDefaultColor(Color.Red);
K_17x5.SetLineWeight(1);
K_48x3.SetDefaultColor(Color.Cyan);
K_48x3.SetLineWeight(1);

switch (UDef_Stoch) {
case "Yes":
    K_UDef = StochasticFull(KPeriod = UDef_FastK , slowing_period = UDef_Slowing) + VertDisp;
case "No":
    K_UDef = Double.nan; }
K_UDef.SetDefaultColor(Color.Cyan);
K_UDef.SetLineWeight(1);

### Start of SOAPCount - this code adapted from the SOAPCount study by Managematics.
Def CountChg;
REC SOAPCount;
Def distance;
Def multiplier;
def offset;
def margin;

switch (arrow_location) {
case Always_Onscreen:  offset = 0; distance = 1; multiplier = 1; margin = 15;
case Partially_Onscreen: offset = 0; distance = 1; multiplier = 1; margin = 5;
case Possibly_Offscreen: offset = 0; distance = 1; multiplier = 1; margin = 0;
case On_CenterLine:    offset = 50; distance = 0; multiplier = 0; margin = 0;
case Inside_Lines:     offset = 0; distance = -20; multiplier = 1; margin = 0;
case OBOS_Lines:      offset = 50; distance = 20; multiplier = 0; margin = 0; }

def k1v = K_05x3;
def k2v = K_08x5;
If k2v > 50 + VertDisp
then {
    CountChg = If k1v <= K2v and k1v[1] > k2v[1] and k2v[1] > 50 + VertDisp Then -1 else 0;
    SOAPCount = compoundvalue(1, Min (0, SOAPCount[1]) + CountChg, 0);}
Else {
    CountChg = If k1v >= K2v and k1v[1] < k2v[1] and k2v[1] <= 50 + VertDisp Then 1 else 0;
    SOAPCount = compoundvalue(1, Max (0, SOAPCount[1]) + CountChg, 0);}

Plot SoapSell =
if k2v > 50 + VertDisp and k1v <= K2v and k1v[1] > k2v[1] then
    K2v * multiplier + offset + distance
else Double.Nan;
SoapSell.SetPaintingStrategy(PaintingStrategy.Arrow_Down);
SoapSell.HideBubble();

```

```

Plot SoapBuy =
  if k2v < 50 + VertDisp and k1v >= K2v and k1v[1] < k2v[1] then
    K2v * multiplier + offset - distance
  else Double.NaN;
SoapBuy.SetPaintingStrategy(PaintingStrategy.Arrow_Up);
SoapBuy.HideBubble();

```

```

SoapBuy.DefineColor("Template Buy", Color.Yellow);
SoapBuy.DefineColor("SOAP Buy", Color.Green);
SoapSell.DefineColor("Template Sell", Color.Yellow);
SoapSell.DefineColor("SOAP Sell", Color.RED);

```

```

SoapSell.AssignValueColor(
  if SoapCount >= -1 then SoapSell.color("Template Sell")
  else SoapSell.color("SOAP Sell"));
SoapBuy.AssignValueColor(
  if SoapCount <= 1 then SoapBuy.color("Template Buy")
  else SoapBuy.color("SOAP Buy"));

```

```

Plot dummydot =
  if margin == 0 then double.NaN
  else
    if !isNan(SoapBuy) then SoapBuy-margin
    else
      if !isNaN(SoapSell) then SoapSell+margin
      else double.NaN;

```

```

dummydot.setLineWeight(1);
dummydot.setPaintingStrategy(PaintingStrategy.POINTS);
dummydot.AssignValueColor(
  if SoapCount > 1 then SoapBuy.color("SOAP Buy")
  else
    if SoapCount == 1 then SoapBuy.color("Template Buy")
    else
      if SoapCount == -1 then SoapSell.color("Template Sell")
      else SoapSell.color("SOAP Sell"));
dummydot.HideBubble();
### end of SOAPCount

```

```

### RSI snapback studies
plot RSI;
switch (RSI_Mode) {
  case "yes": RSI = RSIWilder (length = RSI_Length) + VertDisp;
  case "no" : RSI = double.NaN; }
RSI.SetDefaultColor(Color.Magenta);
RSI.SetLineWeight(1);
### end of snapback studies

```

```

# Breadth plot for RSI - Stochastic (8,5) Breadth
def Diff_RSI_S8x5 = RSI - K_08x5;
plot RSI_S8x5Breadth =
  if showbreadth_RSI_Stoch8x5 then
    Diff_RSI_S8x5 + 50 + VertDisp
  else double.nan;
RSI_S8x5Breadth.SetDefaultColor(Color.Green);
RSI_S8x5Breadth.SetStyle(Curve.Short_Dash);
#
# cherries for RSI - Stochastic (8,5) Breadth
plot Flag_RSI_S8x5_Up =

```

```

if showFlags_RSI_Stoch8x5 and Diff_RSI_S8x5 > FlagTriggerLevel then
  Diff_RSI_S8x5 + 50 + VertDisp
else double.nan;
Flag_RSI_S8x5_Up.SetDefaultColor(Color.Green);
Flag_RSI_S8x5_Up.SetPaintingStrategy(PaintingStrategy.Line_vs_Points);
Flag_RSI_S8x5_Up.HideBubble();

plot Flag_RSI_S8x5_Dn =
  if showFlags_RSI_Stoch8x5 and Diff_RSI_S8x5 < -FlagTriggerLevel then
    Diff_RSI_S8x5 + 50 + VertDisp
  else double.nan;
Flag_RSI_S8x5_Dn.SetDefaultColor(Color.Green);
Flag_RSI_S8x5_Dn.SetPaintingStrategy(PaintingStrategy.Line_vs_Points);
Flag_RSI_S8x5_Dn.HideBubble();
#
# Breadth for RSI - Stochastic (UserDefined) Breadth
def Diff_RSI_SUDef = RSI - K_UDef;
plot RSI_SUDefBreadth =
  if showbreadth_RSI_StochUDef then
    Diff_RSI_SUDef + 50 + VertDisp
  else double.nan;
RSI_SUDefBreadth.SetDefaultColor(Color.Cyan);
RSI_SUDefBreadth.SetStyle(Curve.Short_Dash);
#
# blueberries for RSI - Stochastic (User Defined) Breadth
plot Flag_RSI_SUDef_Up =
  if showFlags_RSI_StochUDef and Diff_RSI_SUDef > FlagTriggerLevel then
    Diff_RSI_SUDef + 50 + VertDisp
  else double.nan;
Flag_RSI_SUDef_Up.SetDefaultColor(Color.Cyan);
Flag_RSI_SUDef_Up.SetPaintingStrategy(PaintingStrategy.Line_vs_Points);
Flag_RSI_SUDef_Up.HideBubble();

plot Flag_RSI_SUDef_Dn =
  if showFlags_RSI_StochUDef and Diff_RSI_SUDef < -FlagTriggerLevel then
    Diff_RSI_SUDef + 50 + VertDisp
  else double.nan;
Flag_RSI_SUDef_Dn.SetDefaultColor(Color.Cyan);
Flag_RSI_SUDef_Dn.SetPaintingStrategy(PaintingStrategy.Line_vs_Points);
Flag_RSI_SUDef_Dn.HideBubble();
# end of Breadth plots

```