



Harmonic Pattern Detection [LUX]

LuxAlgo PREMIUM Ara 27, 2021



Harmonic patterns make up a major part of the many patterns traders use to make investment decisions. The following tool aims to automatically categorize which XABCD harmonic pattern is highlighted by the user and to alert when the price reaches the PRZ or D point.

The tool can categorize Bat, Gartley , Butterfly , and Crab patterns.

Settings

- XA Precision: The Gartley and Butterfly patterns require precise ratios for the XA segment, this setting allows giving some headroom for the detection of these patterns. For example, the Gartley pattern requires a ratio for the XA segment of 0.618, using an XA precision of 0.01 will allow the segment to be considered correct if above 0.608 and under 0.628.
- Bullish: Color of a bullish pattern
- Bearish: Color of a bearish pattern

The X, A, B, C, D settings determine the location of the harmonic pattern vertices. The user does not need to change them from the settings, instead only requiring adjusting their location on the chart like with a regular drawing tool. Setting these vertices is required when adding the indicator to your chart.

Usage



Upon setting the harmonic pattern vertices, the segments, as well as each ratio and PRZ, will be displayed. A dashboard in the top right displays which harmonic pattern has been detected.





Detected bearish crab pattern on BTCUSD15.



Bullish butterfly pattern on MATICUSD15. It is important not to use an XA precision value that would return overlapping ranges between the Gartley / Harmonic and other patterns. Using the default value is recommended.

The upper limit of the PRZ is determined as vertex D plus 38.2% of segment DX , while the lower limit is the vertex D minus 38.2% of segment DX . Various methods exist for the determination of the PRZ, this one is general but the user can use one proper to the detected harmonic pattern .

Finally hovering on the label highlighting the segment ratios return the proper ratio used by each harmonic pattern for that precise segment.

Instant-access to premium tools: <https://LuxAlgo.com>

Guides: <https://docs.luxalgo.com>

Discord: <https://discord.gg/LUX>

Español: <https://es.luxalgo.com>

Français: <https://fr.luxalgo.com>

Deutsch: <https://de.luxalgo.com>

Türkçe: <https://tr.luxalgo.com>



Açık kaynak kodlu komut dosyası ⓘ

Gerçek TradingView rühuyla, bu betiğin yazarı, yatırımcının anlayılabilirliği ve doğrulanılabilirliği için onu açık kaynak olarak yayınladı. Yazarın eline sağlık! Bunu ücretsiz olarak kullanabilirsiniz, ancak bu kodun bir yanında yeniden kullanımı [Kullanım Koşulları](#) ile yönetilir. Bir grafikte kullanmak için favorilere ekleylebilirsiniz.

Feragatname

Bilgiler ve yayınlar, TradingView tarafından sağlanan veya onaylanan finansal, yatırım, işlem veya diğer türden tavsiye veya tavsiyeler anlamına gelmez ve teşkil etmez. [Kullanım Şartları](#)nda daha fazlasını okunun.

Bu komut dosyasını bir grafikte kullanmak ister misin? ⓘ

[★ Favori göstergelere ekle](#)

```

1 // This work is licensed under a Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) https://creativecommons.org/licenses/by-nc-sa/4.0/
2 // © LuxAlgo
3
4 //@version=5
5 indicator("Harmonic Pattern Detection [LUX]", overlay=true)
6 precision = input.float(.01,'XA Precision',step=.01,@group='Precision')
7 bullish = input(#0cb51a,'Bullish',group='style')
8 bearish = input(#ff1f00,'Bearish',group='style')
9
10 x = input.time(0,'X',confirm=true,group='Anchor Points')
11 a = input.time(0,'A',confirm=true,group='Anchor Points')
12 b = input.time(0,'B',confirm=true,group='Anchor Points')
13 c = input.time(0,'C',confirm=true,group='Anchor Points')
14 d = input.time(0,'D',confirm=true,group='Anchor Points')
15 //----
16 x_y = ta.valuewhen(time==x,close,0)
17 a_y = ta.valuewhen(time==a,close,0)
18 b_y = ta.valuewhen(time==b,close,0)
19 c_y = ta.valuewhen(time==c,close,0)
20 //----
21 dist(fib)=>
22     d_y = a_y + fib*(x_y - a_y)
23     cd = math.abs(d_y-c_y)/math.abs(c_y-b_y)
24
25 ab = math.abs(a_y-b_y)/math.abs(a_y-x_y)
26 bc = math.abs(c_y-b_y)/math.abs(a_y-b_y)
27 //----
28 var tb = table.new(position.top_right,2,4, bgcolor=color.gray)
29
30 var float upper_prz = na
31 var float lower_prz = na
32 var float d_y = na
33
34 if time == math.max(x,a,b,c)
35     css = x_y > a_y ? bearish : bullish
36
37     line.new(x,y_a,a_y,xloc=x.loc.bar_time,color=css,width=2)
38     line.new(a,y_b,b_y,xloc=x.loc.bar_time,color=css,width=2)
39     line.new(b,y_c,c_y,xloc=x.loc.bar_time,color=css,width=2)
40
41     line.new(x,y_b,b_y,xloc=x.loc.bar_time,color=css,style=line.style_dotted)
42     line.new(x,y_h,v.h.v.xloc=x.loc.har_time,color=css,style=line.style_dotted)

```

```

-->----->----->----->----->----->
43 line.new(a_a_y,c_c_y,xloc=xloc.bar_time,color=css,style=line.style_dotted)
44 //-----
45
46 label.new(x_x_y,'X',xloc=xloc.bar_time,color=css,style=x_y > a_y ? label.style_label_down : label.style_label_up
47 | ,textcolor=color.white,size=size.small)
48 label.new(a_a_y,'A',xloc=xloc.bar_time,color=css,style=x_y > a_y ? label.style_label_up : label.style_label_down
49 | ,textcolor=color.white,size=size.small)
50 label.new(b_b_y,'B',xloc=xloc.bar_time,color=css,style=x_y > a_y ? label.style_label_down : label.style_label_up
51 | ,textcolor=color.white,size=size.small)
52 label.new(c_c_y,'C',xloc=xloc.bar_time,color=css,style=x_y > a_y ? label.style_label_up : label.style_label_down
53 | ,textcolor=color.white,size=size.small)
54
55 AB_tooltip = 'AB_0.382-0.5 \nA = 0.618 \nB = 0.786 \nC = 0.382-0.618'
56 BC_tooltip = 'BC_0.382-0.886 \nB = 0.382-0.886 \nC = 0.382-0.886'
57 CD_tooltip = 'CD_1.618-2.618 \nA = 1.130-1.618 \nB = 1.618-2.240 \nC = 2.618-3.618'
58
59
60 label.new(int(math.avg(x,b)),math.avg(x_y_b_y),str.tostring(ab,'###'),xloc=xloc.bar_time,color=css,style=label.style_label_center
61 | ,textcolor=color.white,size=size.small,tooltip=AB_tooltip)
62 label.new(int(math.avg(a,c)),math.avg(a_y_c_y),str.tostring(bc,'###'),xloc=xloc.bar_time,color=css,style=label.style_label_center
63 | ,textcolor=color.white,size=size.small,tooltip=BC_tooltip)
64
65 //----
66
67 table.cell(tb,0,0,'Bat',text_color=color.white,text_halign=text.align_left)
68 table.cell(tb,0,1,'Gartley',text_color=color.white,text_halign=text.align_left)
69 table.cell(tb,0,2,'Butterfly',text_color=color.white,text_halign=text.align_left)
70 table.cell(tb,0,3,'Crab',text_color=color.white,text_halign=text.align_left)
71
72 //Bat
73 if ab > 0.382 and ab < .5 and bc > 0.382 and bc < 0.886 and dist(0.886) > 1.618 and dist(0.886) < 2.618
74 | table.cell(tb,1,0,'✓')
75 | table.cell(tb,1,1,'✗')
76 | table.cell(tb,1,2,'✗')
77 | table.cell(tb,1,3,'✗')
78
79 d_y := a_y + 0.886*(x_y - a_y)
80 cd = math.abs(d_y-c_y)/math.abs(c_y-b_y)
81
82 line.new(c_c_y,d_d_y,xloc=xloc.bar_time,color=css,width=2)
83 line.new(b_b_y,d_d_y,xloc=xloc.bar_time,color=css,style=line.style_dotted)
84
85 label.new(d_d_y,'D',xloc=xloc.bar_time,color=css,style=x_y > a_y ? label.style_label_down : label.style_label_up
86 | ,textcolor=color.white,size=size.small)
87 label.new(int(math.avg(d_d_y)),math.avg(d_y_b_y),str.tostring(cd,'###'),xloc=xloc.bar_time,color=css,style=label.style_label_center
88 | ,textcolor=color.white,size=size.small,tooltip=CD_tooltip)
89
90 //Gartley
91 else if ab > 0.618+precision and ab < 0.618+precision and bc > 0.382 and bc < 0.886 and dist(0.786) > 1.13 and dist(0.786) < 1.618
92 | table.cell(tb,1,0,'✗')
93 | table.cell(tb,1,1,'✓')
94 | table.cell(tb,1,2,'✗')
95 | table.cell(tb,1,3,'✗')
96
97 d_y := a_y + 0.786*(x_y - a_y)
98 cd = math.abs(d_y-c_y)/math.abs(c_y-b_y)
99
100 line.new(c_c_y,d_d_y,xloc=xloc.bar_time,color=css,width=2)
101 line.new(b_b_y,d_d_y,xloc=xloc.bar_time,color=css,style=line.style_dotted)
102
103 label.new(d_d_y,'D',xloc=xloc.bar_time,color=css,style=x_y > a_y ? label.style_label_down : label.style_label_up
104 | ,textcolor=color.white,size=size.small)
105 label.new(int(math.avg(d_d_y)),math.avg(d_y_b_y),str.tostring(cd,'###'),xloc=xloc.bar_time,color=css,style=label.style_label_center
106 | ,textcolor=color.white,size=size.small,tooltip=CD_tooltip)
107
108 //Butterfly
109 else if ab > 0.786+precision and ab < 0.786+precision and bc > 0.382 and bc < 0.886 and dist(1.27) > 1.618 and dist(1.27) < 2.24
110 | table.cell(tb,1,0,'✗')
111 | table.cell(tb,1,1,'✗')
112 | table.cell(tb,1,2,'✓')
113 | table.cell(tb,1,3,'✗')
114
115 d_y := a_y + 1.27*(x_y - a_y)
116 cd = math.abs(d_y-c_y)/math.abs(c_y-b_y)
117
118 line.new(c_c_y,d_d_y,xloc=xloc.bar_time,color=css,width=2)
119 line.new(b_b_y,d_d_y,xloc=xloc.bar_time,color=css,style=line.style_dotted)
120
121 label.new(d_d_y,'D',xloc=xloc.bar_time,color=css,style=x_y > a_y ? label.style_label_down : label.style_label_up
122 | ,textcolor=color.white,size=size.small)
123 label.new(int(math.avg(d_d_y)),math.avg(d_y_b_y),str.tostring(cd,'###'),xloc=xloc.bar_time,color=css,style=label.style_label_center
124 | ,textcolor=color.white,size=size.small,tooltip=CD_tooltip)
125
126 //Crab
127 else if ab > 0.382 and ab < .618 and bc > 0.382 and bc < 0.886 and dist(1.618) > 2.224 and dist(1.618) < 3.618
128 | table.cell(tb,1,0,'✗')
129 | table.cell(tb,1,1,'✗')
130 | table.cell(tb,1,2,'✗')
131 | table.cell(tb,1,3,'✓')
132
133 d_y := a_y + 1.618*(x_y - a_y)
134 cd = math.abs(d_y-c_y)/math.abs(c_y-b_y)
135
136 line.new(c_c_y,d_d_y,xloc=xloc.bar_time,color=css,width=2)
137 line.new(b_b_y,d_d_y,xloc=xloc.bar_time,color=css,style=line.style_dotted)
138
139 label.new(d_d_y,'D',xloc=xloc.bar_time,color=css,style=x_y > a_y ? label.style_label_down : label.style_label_up
140 | ,textcolor=color.white,size=size.small)
141 label.new(int(math.avg(d_d_y)),math.avg(d_y_b_y),str.tostring(cd,'###'),xloc=xloc.bar_time,color=css,style=label.style_label_center
142 | ,textcolor=color.white,size=size.small,tooltip=CD_tooltip)
143
144 | table.cell(tb,1,0,'✗')
145 | table.cell(tb,1,1,'✗')
146 | table.cell(tb,1,2,'✗')
147 | table.cell(tb,1,3,'✗')
148
149 d_y := a_y + 0.886*(x_y - a_y)
150 cd = math.abs(d_y-c_y)/math.abs(c_y-b_y)
151
152 line.new(c_c_y,d_d_y,xloc=xloc.bar_time,color=css,width=2)
153 line.new(b_b_y,d_d_y,xloc=xloc.bar_time,color=css,style=line.style_dotted)
154
155 label.new(d_d_y,'D',xloc=xloc.bar_time,color=css,style=x_y > a_y ? label.style_label_down : label.style_label_up
156 | ,textcolor=color.white,size=size.small)
157 label.new(int(math.avg(d_d_y)),math.avg(d_y_b_y),str.tostring(cd,'###'),xloc=xloc.bar_time,color=css,style=label.style_label_center
158 | ,textcolor=color.white,size=size.small,tooltip=CD_tooltip)
159
160 upper_prz := d_y + 0.382*math.abs(d_y - x_y)
161 lower_prz := d_y - 0.382*math.abs(d_y - x_y)
162 box.new(x,upper_prz,d,lower_prz,border_color=na,xloc=xloc.bar_time,bgcolor=color.new(css,80),extend=extend.both)
163
164 line.new(x_x_y,d_d_y,xloc=xloc.bar_time,color=css,style=line.style_dotted)
165
166 //Alerts
167 alertcondition(ta.crossover(close,lower_prz) or ta.crossunder(close,upper_prz),'Price Enter PRZ','Price entered the PRZ')
168 alertcondition(ta.cross(close,d_y),'Price Cross D','Price crossed point D')

```

```
169 //PRZ Plots
170 plot(upper_prz,'Upper PRZ',na)
171 plot(lower_prz,'Lower PRZ',na)
```

Yorumlar



Yararlı veya teşvik edici bir yorum bırakın. Piyasalara birlikte hakim olalım

[Alkışlarla yorum](#)

[Yorum Paylaş](#)



RicardoSantos WIZARD · Ara 29, 2021

thank you for this tool, and improving pinescript in general for every one, altho the title is a bit misleading :)

100 koin

+2 ▲ [Cevap Gönder](#)



alexgrover WIZARD · Ara 30, 2021

@RicardoSantos, Sorry to hear that, the tool does indeed identify/detect a harmonic pattern given various user set vertices, no attempt of being misleading was done whatsoever.

+1 ▲ [Cevap Gönder](#)



Mooxe · Ara 28, 2021

Hey guys, sorry for asking but I receive the following message when adding the Indicator:
Set the "X" time for "Name".

Thank you for helping!

+22 ▲ [Cevap Gönder](#)



alexgrover WIZARD · Ara 28, 2021

@Mooxe, You need to set the XABCD point directly by clicking on the chart, like with the regular XABCD built-in drawing tool.

▲ [Cevap Gönder](#)



Mooxe · Ara 29, 2021

@alexgrover, Thank you Sir! I got it :)

▲ [Cevap Gönder](#)



AramisKC PREMIUM · Ara 27, 2021

Very powerful tool if used correctly.

+5 ▲ [Cevap Gönder](#)



JellyBean_Mr-U · Ara 27, 2021

Thanks for this Lux!

+4 ▲ [Cevap Gönder](#)



LuxAlgo PREMIUM · Ara 27, 2021

@deflee16720, <3

▲ [Cevap Gönder](#)



OvidiuCraciun PRO · Ara 30, 2021

when I try to add it to my chart it tells me "Set the X time...." message. Well, how do I set it... where?

+1 ▲ [Cevap Gönder](#)



alexgrover WIZARD · Ara 31, 2021

@OvidiuCraciun, You need to set the XABCD points manually, it's like a regular drawing tool, click on your chart to set the point time location.

▲ [Cevap Gönder](#)



maroof81 · Ara 30, 2021 · iOS İçin TradingView

thank you

+1 ▲ [Cevap Gönder](#)



SlikRain PRO+ · Ara 28, 2021

As Always Amazing.... Thank you for sharing this...

+1 ▲ [Cevap Gönder](#)



tyler8910 PRO · Ara 27, 2021

This is awesome -- but how does one determine where to place X, A,B,C,D?

+1 ▲ [Cevap Gönder](#)



alexgrover WIZARD · Ara 28, 2021

@tyler8910, By clicking on the chart, like with the regular XABCD built-in drawing tool.

▲ [Cevap Gönder](#)

**TraderRaphael** PREMIUM

· Ara 27, 2021



Can't understand how to use, I'd rather pay for a easy one can auto detective .

[+1 ▲ Cevap Gönder](#)**MoneyMakerDoug242**

· Oca 11



@TraderRaphael, same here

[▲ Cevap Gönder](#)**bono297**

· Ara 27, 2021



It is not working, how can I use it? It is required to adjust clock but there is no possibility.

[+1 ▲ Cevap Gönder](#)**bono297**

· Ara 27, 2021



@bono297, I solved, thx, sorry for previous message.

[▲ Cevap Gönder](#)**bono297**

· Ara 27, 2021



Thanks

[+1 ▲ Cevap Gönder](#)**newbr08**

· Ara 27, 2021



Won't use

[+1 ▲ Cevap Gönder](#)**mmoondoo**

· May 9



ok

[▲ Cevap Gönder](#)**FlipWhale** PRO+

· Oca 17



You're a great programmer Lux, much love brother, learning a lot reading your code. :)

[▲ Cevap Gönder](#)