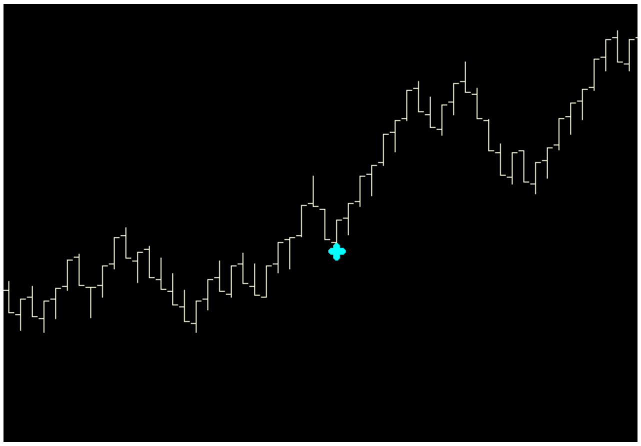
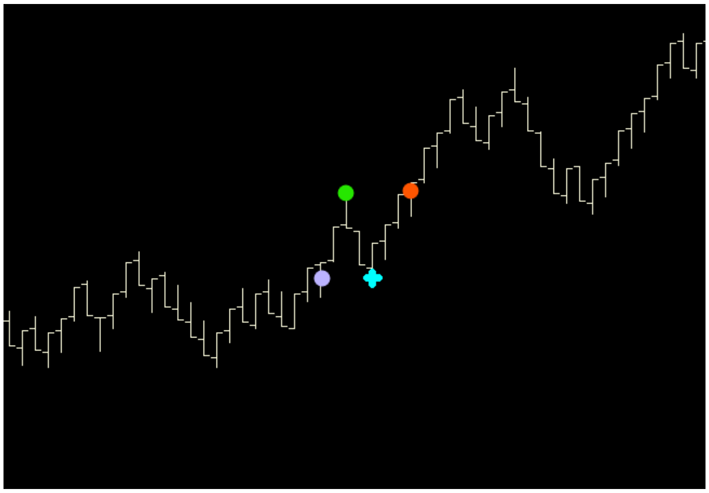
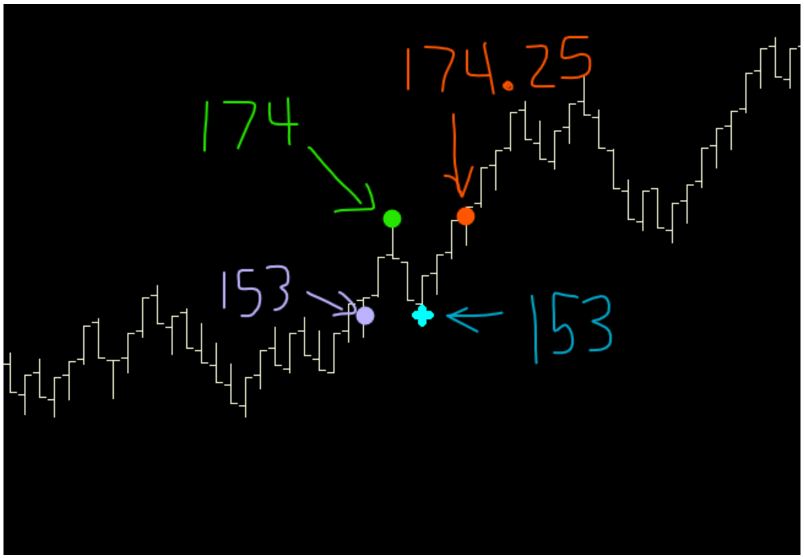
I have an indicator that gives a cyan alert, always at the bottom of a (range) bar, like this:



As represented in the screen shot below, I want to write code that will respond to that alert by placing an entry order (represented by an orange dot) at a breach of the highest price has been (previous pivot high represented by a green dot) since price last crossed the same level (purple dot) as when the alert fired (the original cyan alert)



So for example (as labeled in the screen shot below) if the cyan alert fires at the low of the close of the AlertBar at a price of 153, I want to know the highest price has been (174) since the last time it was at 153 in order to place an order for entry at the breach of that previous high at 174.25.



It seems to me that in order to determine that most recent pivot high, I'd need to use some logic like the following. In this attempt to outline my thinking, I'm assuming the bar with the cyan alert on it (AlertBar) has just completed and is also the PreviousBar:

```
        //I imagine I'd start with a variable called something like BarBeingEvaluated, which will look back, bar by bar, until it finds a bar equal to or less than the AlertBar. The first line below sets that variable initially to -1, meaning the last bar that finished:

BarBeingEvaluated = PreviousBar/AlertBar = -1

        //This next line determines whether two rules for placing an order for entry have been met: 1) an alert has fired, and 2) the AlertBar is lower than the bar before it:

If [AlertFired = True] and [AlertBar came from low of previous bar = True] Then Begin

        //And now is when I imagine I'd need some kind of loop to find the last time price was at the AlertPrice of 153, in order to find the highest it's been (174) since then:

Repeat

        //Each time 1 is subtracted, we look back one more bar:

(TheNew/Next)BarBeingEvaluated = (TheCurrnet/Previous)BarBeingEvaluated - 1

        //And capture that bar's high, adding it to a data set:

Add high of BarBeingEvaluated to something like a [BarHighsDataSet]

        //Until we've gone far enough back that we've found the last time price was at the same level as it was when the alert fired:

Until [low of BarBeingEvaluated] ≤ [low of AlertBar] = True

        //Then we can go back and determine the highest value of the [BarHighsDataSet] to determine the previous pivot high:

Pivot High = MaxValue of [BarHighsDataSet]

        //Now we just need to add a quarter point to that PivotHigh to determine the entry price:

PlaceEntryOrder for when price = [PivotHigh + 0.25]

End
```