

Hold the #1 Spot

Example Picture:

Order books Open orders Closed orders Wallets								
1 of 10 Prev Next 50			\$469,409.98 3 Decimals \$1,169,906.63			1 of 1 Prev Next All		
Agg. total	Total	Quantity	Price	Price	Quantity	Total	Agg. total	
1,455.6139	1,455.6139	28,727.332	0.05067 ^	0.05081 v	39,187.719	1,991.1280	1,991.1280	
3,451.1177	1,995.5037	39,390.125	0.05066 ^	0.05082 v	128,555.497	6,533.1904	8,524.3184	
5,445.3786	1,994.2609	39,373.365	0.05065 ^	0.05083 v	39,377.805	2,001.5738	10,525.8922	
7,886.2244	2,440.8458	48,219.000	0.05062 ^	0.05089 v	84,860.000	4,318.5254	14,844.4176	
12,388.9596	4,502.7352	88,969.279	0.05061 ^	0.05092 v	288,020.000	14,665.9784	29,510.3960	
12,488.9596	100.0000	1,982.554	0.05044 ^	0.05093 v	331.746	16.8958	29,527.2918	
12,604.9256	115.9660	2,300.000	0.05042 ^	0.05095 v	144.270	7.3506	29,534.6424	
12,610.4286	5.5030	109.425	0.05029 ^	0.05097 v	117.236	5.9755	29,540.6179	
12,615.9316	5.5030	109.796	0.05012 ^	0.05099 v	109.057	5.5608	29,546.1787	
17,257.9031	4,641.9715	92,635.632	0.05011 ^	0.05100 v	858,352.352	43,775.9700	73,322.1487	
17,263.8048	5.9017	117.868	0.05007 ^	0.05113 v	2,378,240.000	121,599.4112	194,921.5599	

The above is an example order book at Bittrex. Below is a random example of the Bid/Ask Prices

If the goal is to “buy” with a “limit order” holding the #1 spot in the order book, then the buy order should set the price to the smallest increment above the current highest buy price:

$$0.05067 + 0.00001 = 0.05068$$

Bid	Ask
0.05067	0.05082
0.05066	0.05083
0.05065	0.05085
0.05061	0.05086
0.05044	0.05092
0.05042	0.05093
0.05029	0.05095
0.05012	0.05097
0.05011	0.05099

If the goal is to “sell” with a “limit order” holding the #1 spot in the order book, then the buy order should set the price to the smallest increment below the current highest buy price:

$$0.05082 - 0.00001 = 0.05081$$

If a new order comes in above our current order, we will want to cancel the previous order and put in a new “buy limit order” to the smallest increment above the current highest buy price

$$0.05071 + 0.00001 = 0.05072$$

Bid	Ask
0.05071	0.05077
0.05068	0.05079
0.05067	0.05081
0.05066	0.05082
0.05065	0.05083
0.05061	0.05085
0.05044	0.05086
0.05042	0.05092
0.05029	0.05099

If a new order comes in above our current order, we will want to cancel the previous order and put in a new “buy limit order” to the smallest increment below the current highest buy price

$$0.05077 - 0.00001 = 0.05076$$

If orders “under” our order start to “hollow out” or cancel” and the gap between our #1 spot and the next spot is greater than the lowest increment, then change the order to the smallest increment above the highest price

IF Level 1 – Level 2 > the smallest increment, THEN cancel order AND REPLACE

$0.05072 - 0.05061 > 0.00001$ THEN

$0.05061 + 0.00001 = 0.05062$

Bid	Ask
0.05072	0.05076
	0.05083
0.05061	0.05085
0.05044	0.05086
0.05042	0.05092
0.05029	0.05099

If orders “under” our order start to “hollow out” or cancel” and the gap between our #1 spot and the next spot is greater than the lowest increment, then change the order to the smallest increment below the highest price

IF Level 1 – Level 2 > the smallest increment, THEN cancel order AND REPLACE

$0.05083 - 0.00001 = 0.05082$

If the Spread (Bid – Ask) is equal to the lowest increment, then the buy order would be a tie for first place.

$0.05072 - 0.05071 = 0.00001$

Smallest Increment = 0.00001

THEN Buy Order Price = 0.05072

Bid	Ask
0.05071	0.05072
0.05068	0.05074
0.05065	0.05077
0.05061	0.05079
0.05044	0.05092
0.05042	0.05093
0.05029	0.05095
0.05012	0.05097
0.05011	0.05099

If the Spread (Bid – Ask) is equal to the lowest increment, then the sell order would be a tie for first place.

$0.05072 - 0.05071 = 0.00001$

Smallest Increment = 0.00001

THEN Sell Order Price = 0.05072

NOTE: Maybe obvious, but this logic stops when the buy/sell limit order is filled

NOTE: This final condition should not be in the first initial release

Bot War Condition:

If our bot is at war with another bot for the #1 spot in the order book. We can determine this by a “leap frog” affect, where we have to change the price to regain the #1 spot every XXXX milliseconds. I do not know the speed of bots, or even the speed of our bot, but this value can be determined through experimentation in the future.

If we have 3 price changes in $6 * XXXX$ milliseconds, then on the 4th price change, attempt a tie. Do not go one increment above/below, but instead tie for first place. If the other bot accepts a tie, then stay at that price. If the other bot does not accept the tie, then continue the war for 1st place until the 10th increment. At the 10th increment, utilize the rules for 2nd place to reset the price, and then if the other bot follows, restart the war again. If the other bot does not follow, then utilize there will be a choice whether to continue to follow, or to use the logic to stay in 2nd place. The future strategy will dictate which choice to utilize, so the code should accept an “alternate hold 2nd place” in “bot war condition”, when the action strategy allows for a 2nd place hold.