

KMK FW - HoldTap Keycodes

The HoldTap module lets keys do double duty: tap the key to do one thing, hold it longer than the configurable `tap_time` to do another.

HoldTap is often used with modifier keys. For example `KC.HT(KC.ESCAPE, KC.LCTRL)` configures a key that sends Escape when tapped and left control when held.

It can be used with regular keys as well like `KC.HT(KC.SPACE, KC.ENTER)` to send space on tap and enter on hold.

Simply import HoldTap and add it to the modules list.

This lets you use `KC.HT` actions like those below.

```
from kmk.modules.holdtap import HoldTap
holdtap = HoldTap()
# optional: set a custom tap timeout in ms
# holdtap.tap_time = 300
keyboard.modules.append(holdtap)
```

Keycodes

New Keycode	Description
<code>LCTRL = KC.HT(KC.SOMETHING, KC.LCTRL)</code>	<code>LCTRL</code> if held <code>kc</code> if tapped
<code>LSFT = KC.HT(KC.SOMETHING, KC.LSFT)</code>	<code>LSHIFT</code> if held <code>kc</code> if tapped
<code>LALT = KC.HT(KC.SOMETHING, KC.LALT)</code>	<code>LALT</code> if held <code>kc</code> if tapped
<code>LGUI = KC.HT(KC.SOMETHING, KC.LGUI)</code>	<code>LGUI</code> if held <code>kc</code> if tapped
<code>RCTRL = KC.HT(KC.SOMETHING, KC.RCTRL)</code>	<code>RCTRL</code> if held <code>kc</code> if tapped
<code>RSFT = KC.HT(KC.SOMETHING, KC.RSFT)</code>	<code>RSHIFT</code> if held <code>kc</code> if tapped
<code>RALT = KC.HT(KC.SOMETHING, KC.RALT)</code>	<code>RALT</code> if held <code>kc</code> if tapped
<code>RGUI = KC.HT(KC.SOMETHING, KC.RGUI)</code>	<code>RGUI</code> if held <code>kc</code> if tapped
<code>SGUI = KC.HT(KC.SOMETHING, KC.LSHFT(KC.LGUI))</code>	<code>LSHIFT</code> and <code>LGUI</code> if held <code>kc</code> if tapped
<code>LCA = KC.HT(KC.SOMETHING, KC.LCTRL(KC.LALT))</code>	<code>LCTRL</code> and <code>LALT</code> if held <code>kc</code> if tapped
<code>LCAG = KC.HT(KC.SOMETHING, KC.LCTRL(KC.LALT(KC.LGUI)))</code>	<code>LCTRL</code> and <code>LALT</code> and <code>LGUI</code> if held <code>kc</code> if tapped
<code>MEH = KC.HT(KC.SOMETHING, KC.LCTRL(KC.LSFT(KC.LALT)))</code>	<code>CTRL</code> and <code>LSHIFT</code> and <code>LALT</code> if held <code>kc</code> if tapped
<code>HYPR = KC.HT(KC.SOMETHING, KC.HYPR)</code>	<code>LCTRL</code> and <code>LSHIFT</code> and <code>LALT</code> and <code>LGUI</code> if held <code>kc</code> if tapped

Custom HoldTap Behavior

The full HoldTap signature is as follows:

```
KC.HT(KC.TAP, KC.HOLD, prefer_hold=True, tap_interrupted=False,  
tap_time=None, repeat=HoldTapRepeat.NONE)
```

- `prefer_hold`: decides which keycode the HoldTap key resolves to when another key is pressed before the timeout finishes. When `True` the hold keycode is chosen, the tap keycode when `False`.
- `tap_interrupted`: decides if the timeout will interrupt at the first other key press/down, or after the first other key up/release. Set to `True` for interrupt on release.
- `tap_time`: length of the tap timeout in milliseconds.
- `repeat`: decides how to interpret repeated presses if they happen within `tap_time` after a release.
- `TAP`: repeat tap action, if previous action was a tap.
- `HOLD`: repeat hold action, if previous action was a hold.
- `ALL`: repeat all of the above.
- `NONE`: no repeat action (default), everything works as expected. The `HoldTapRepeat` enum must be imported from `kmk.modules.holdtap`.

Each of these parameters can be set for every HoldTap key individually.