Gesture PID tuning

The gesture based PID tuning allows the pilot to change the acro PIDs at the field, without the use of a computer. The new PIDs can be saved, and will be loaded next time the quad powers up. If not saved by performing the appropriate gesture, they will be discarded when the quad is powered off.

Silverware branches of BWHOOP, H8BLUE (both STM MCU), H101/H8S and H8Green (GigaDevice MCU) now have the possibility to tune the PIDs via Tx gestures (thanks to RCGroups user eitama for adding it!).

New options in config.h:

```
// Comment out to disable pid tuning gestures
#define PID_GESTURE_TUNING
#define COMBINE_PITCH_ROLL_PID_TUNING
```

Make sure you have commented in

```
#define GESTURES2_ENABLE
```

on STM Silverwares, it won't work otherwise!

Basic instructions:

The PIDs can be changed in the order you find them in pid.c

example:

```
// Kp                            ROLL       PITCH     YAW
float pidkp_flash[PIDNUMBER] = { 12.0e-2, 12.0e-2, 4e-1 };
// Ki
float pidki_flash[PIDNUMBER] = { 6.5e-1, 6.5e-1, 50e-1 };
// Kd
float pidkd_flash[PIDNUMBER] = { 6.05e-1, 6.05e-1, 4e-1 };
```

If you plug in the battery, The “cursor” stands on PID value “P” on ROLL axis

Gestures are as follows:

- Up - Down - Up (further called UDU) (means right stick Up-Center-Down-Center-Up, others work similar)
- Up - Down - Down (further called UDD)
- Up - Down - Left (further called UDL)
- Up - Down - Right (further called UDR)
- Down - Down - Down (further called DDD)

UDU: Cycle the Cursor to the next Row (Confirmed by LED Blinks, 1x Blink = P, 2x Blink = I, 3x Blink = D)

UDD: Cycle the Cursor to the next Column (ROLL –> PITCH –> YAW, also confirmed by the LED blinks as above)

note: If you selected #define COMBINE_PITCH_ROLL_PID_TUNING, you won't get the 2 blinks while changing the Column, as ROLL/PITCH are tuned simultaneously

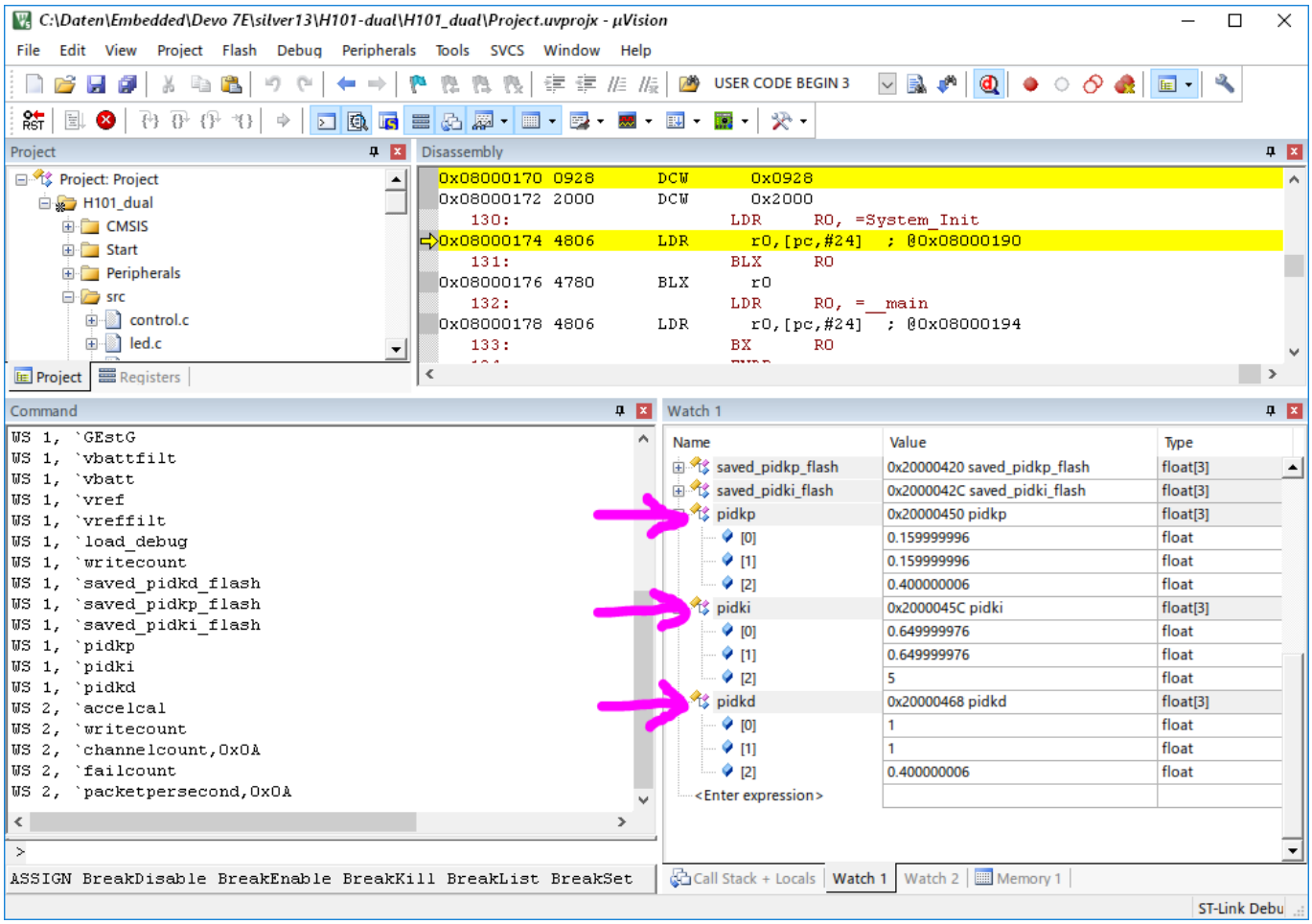
UDL: Decrease the selected Value (Where the Cursor stands) by 10%

UDR: Increase the selected Value (Where the Cursor stands) by 10%

DDD: Save The selected PIDs to the Quad. (Note: this performs an ACC calibration if PIDs are unchanged. If UDL/UDR is performed at least once, the ACC calibration is skipped and only the PIDs are saved.)

For PID tuning, it's highly recommended to use SilverVISE Android app, where you can see the cursor and also see the PID numbers / changes. For all infos how to use the app please click at the link above.

For all that don't own an Android device and are interested in the actual numbers, debug mode can be used (with Keil), where the PIDs can be seen (as decimal numbers). Just add pidkp, pidki, and pidkd to the Watch1 window:



Important:

Once the PIDs got changed and saved via gestures, they will stay even when you reflash the firmware, given that you *did not change the PIDs in PID.c.* Once these are changed, these will overwrite the PIDs in flash, so take care while using Debug Mode

Entering debug will automatically flash code unless a setting is turned off.

To use debug mode without reflashing the quad, you can do the following:

In the last tab of the settings (utilities) uncheck “update target before debugging”

Also go to stlink - settings , where the algorithms are, and set “do not erase” , also uncheck program and verify. This way it can't program it even if it tries to. This is just for added safety, the first option should do it but some things one really don't want erased even if it's unlikely. Be aware that this procedure only works if you use the exactly same Silverware version you flashed at the quad, otherwise you may get crappy values (because the values shown are stored in a particular memory section, and only the compiler knows where they are. Therefor, it's standard to reflash the code before using debug mode.