SilF4ware

- RCGroups thread
- Github Repository

Supported Hardware

For wiring receiver modules, see this page.

Flight Controller	Weight	Notes	Board Pinout
Omnibus	3.2g		Image
NOXE (first version)	3.4g (?)		Image
NOXE v1	4.3g (?)	untested	Image
F4-XSD	4.2g	RCGroups Post	Image

Receiver	Weight	Notes	Source
NRF24 Mini	0.5g	Howto add an wire antenna	use “nrf24l01 mini” as search term (Banggood)
NRF24 PA LNA (GT-24)	1.1g	Howto add an U.FL antenna	use “GT-24” as search term (Banggood)
XN297			Can be harvested from toy transmitters like H101, H8, ...
XN297 DIY PCB			DIY PCB
XN297		untested	Banggood

Configuration

Overview

Configuration Type	Filename
Main	config.h
Battery	battery.c
Radio	radio_config.txt (Github)
Dshot Driver Selection	hardware.h
Dshot Driver Configuration	depends on driver selection: drv_dshot_bidir.c (RPM Filter), drv_dshot_dma.c (DMA), drv_dshot.c (Conventional Dshot)
PID	Basic PIDs are set in the main config file
PID Advanced Configuration	pid.c
PID Level Mode Configuration	angle_pid.c

Basic Configuration

Battery Cell Count

Default setup is configured for 4S batteries. Make sure to adjust CELL_COUNT_UNSCALED in battery.c

if needed, e.g. for a 2S setup:

```
#define CELL_COUNT_UNSCALED 2 // Voltage divider, idle_offset, and PID
values tuned for 4S.
```

Receiver

The default configuration is setup for NRF24 modules. If using a NRF24 module with PA LNA, it is recommended to adjust TX_POWER in SilF4ware/config.h:

```
#define TX_POWER 1 // 0 .. 3 (use 1 when using an nRF24L01 PA LNA module)
```

If using a XN297 module, see radio_config.txt file for configuration notes.

Dshot

Default setup is configured for using Dshot with RPM Filter. If using this configuration, make sure that the number of magnets on the motor bell is set correct (drv_dshot_bidir.c):

```
#define MOTOR_POLE_COUNT 14 // usually on 22xx motors and above
// #define MOTOR_POLE_COUNT 12 // usually on 18xx motors and below
```

Other Dshot drivers can be selected in hardware.h:

```
#define DSHOT_DMA_BIDIR // needed for RPM_FILTER, 4k loop frequency max
// #define DSHOT_DMA_DRIVER // conventional Dshot, consumes less cycles,
works for 8k loop frequency
// #define DSHOT_DRIVER // delay version
```

2D/3D Flying

3D flying is enabled by default. If using a 2D setup, following changes are needed:

In the main config file (config.h):

```
//#define INVERTED_ENABLE
#define FN_INVERTED CH_OFF
//#define LEVEL_MODE_INVERTED_ENABLE // be careful when enabling this
```

In the dshot configuration file (e.g. when using RPM Filter: drv_dshot_bidir.c):

```
// Enable this for 3D. The 'Motor Direction' setting in BLHeliSuite must be
set to 'Bidirectional' (or 'Bidirectional Rev.') accordingly:
//#define BIDIRECTIONAL
```

Misc

- Props out configuration is enabled by default (comment INVERT_YAW_PID to disable it)
- Default rates are very high, adjust them if needed

Compiling & Flashing

It is recommended to use STM32CubeIDE for compiling SilF4ware. Keil uVision can also be used, but it produces some non-working binary when using lot's of RAM. Change FFT_SIZE in fft.h to 2048 when using Keil.

Flashing can be done with betaflight-configurator. After compiling, the target files can be found in the “Release” directory.

Using

Gestures

- PID Tuning: some gestures have been swapped in comparison to other silverware branches: UDD switches to the next column and UDU to the next row
- LRU: reboot flight controller (nice when otherwise one would unplug and replug the battery)
- LRD: switch to DFU mode (nice if the BOOT button is mechanically hard to reach after installing the FC into a quad)

Motors test mode

With default setup, when using LLU stick gesture (Left, Left, Up) SilF4ware switches into motor test mode (MOTORS_TO_THROTTLE). It can be used to verify that the configured motor order is correct, but also to check for bad/noisy props.

In motor test mode, push the stick in the corresponding direction, e.g. left up will make the motor spin which is configured as front left.

LLD stick gesture (Left, Left, Down) turns this mode off again.

If you are used to other silverware branches, please note that with SilF4ware it is not needed to adjust the idle offset to make sure that only one motor spins.

Devo TX

A Devo 7E build and model file which is able to display PID values can be found in this post. Note that it will only display PIDs set via gestures, not when set via analog aux channels.

Advanced Features

Analog Aux Channels

The variables aux_analog[0] and aux_analog[1] hold a value between 0.0 and 2.0 which can be used in various places in the code. Per default they are used to tweak Kp and Kd respectively. This is done in SilF4ware/pid.c:

```
#define AA_pidkp ( x <2 ? pdScaleValue * aux_analog[ 0 ] : 1.0f ) // Scale
Kp and Kd only for roll and pitch.
#define AA_pidki 1.0f
#define AA_pidkd ( x <2 ? pdScaleValue * aux_analog[ 1 ] : 1.0f ) // Scale
Kp and Kd only for roll and pitch.
```

If you want to use them for something else, change the define for AA_pidkp and AA_pidkd to look similar to the one for AA_pidki:

```
#define AA_pidkp 1.0f
#define AA_pidki 1.0f
#define AA_pidkd 1.0f
```

Now you could use aux_analog[0] and aux_analog[1] for example to tune the filter frequency by adding it to config.h like this:

```
#define GYRO_LPF_2ND_HZ_BASE 400 * aux_analog[ 0 ]
#define GYRO_LPF_2ND_HZ_MAX 400 * aux_analog[ 1 ]
```

(Original post)

Blackbox Logging

Blackbox logging is possible with an external logging device. See details here