## Sensor Data Manager (SDM)

The Sensor Data Manager is the abstract interface for the sensors. It provides a getter method to get the sensor data.  
A single sensor value isn’t meaningful because it is heavily influenced by the vibration jitter. Therefore the values which are provided by the getter method are simply pre-filtered with averaging. It’s averaging 20 values that have bin measured with high frequency.  
The SDM is running in a separate task, this makes is easy to encapsulate all the functionality of the Sensors in one component.

## Logging

Logging is an important part when looking for debugging. The target of the logging on the NIOS II system is to provide all useful data to the developer, additionally it gives the opportunity to display those information’s nicely while flying.   
However, as task on the system the logging it has a very low priority because the logging should not use calculation power which actually would be needed by the flight controlling. So the Logging Task is running as a filler between the more important tasks.   
The data which needs to be logged are produced in the main controlling loop so the end of such a loop is the perfect place to start the logging procedure. But the Logging needs to be encapsulated from the control loop. In the controlling loop all data which should be logged, is pushed into a queue. If there is computing time left (no task is running) the logger task starts running and reading data out of the queue and sends it to the ARM/Linux system via MCAPI

## Interfaces between Components

In the table below (Ref Table In- and output ranges of Components) shows the interfaces between the components.  
Notes:

\* These values are not determined jet, however they not necessarily need to be. The α-β-γ-Filter uses the input and produces the Euler Angles as long as the sensor values, especially the gyro and Accelerometer, are correctly calibrated. For more information see (Ref zur überschrift mit Calibration Accelometer und/oder Gyrometer).  
\*\* These values depend on the P, I and D values in the PID’s. Since these weren’t finally set jet, the value ranges aren’t defined. For more information see (Ref zur Problemsection über das PID)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Component | | | Input Range | | Output Range | |
| Min Value | Max Value | Min Value | Max Value |
| Sensor Data Manager | | | - | - | \* | \* |
| RC Receiver | Roll & Yaw | | - | - | -180 | +180 |
| Pitch | | - | - | -90 | +90 |
| Throttle | | - | - | 0 | 100 |
| α-β-γ-Filter | | | \* | \* | -180 | +180 |
| PIDs | Pitch | Filter | -180 | +180 | \*\* | \*\* |
| Rc | -90 | +90 |
| Roll & Yaw | Filter | -180 | +180 | \*\* | \*\* |
| Rc | -180 | +180 |
| MotorMapper | | | \*\* | \*\* | 0 | 100 |

(Ref Table x “In- and outputranges of Components”)

## PWM-Signals

There are three important different PWM-Signals related to the xCopter project.

The first is the common PWM-signal (cPWM), its range is from 0.0% to 100.0%. It is possible to send a low (0.0%) or a high signal (100%). This signal is normally produced by the xCopter. The driver provides the functionality to send every step between 0 and 255.

The second is a PWM-Signal usually used in model building (mbPWM). Usually it’s running on a 50 kHz frequency (20ms period length). The range is from 1ms high signal to 2ms high signal. It is not possible to create a constant high or low signal.

The third and maybe the most important is the one that was created for the xCopter application. It was needed because the ESC’s require a non high/low signal like the mbPWM and the cPWM doesn’t fulfill this requirement. It’s possible to emulate the mbPWM with the cPWM, but this would decrease the resolution. For example the usable range that would be left over would be between 25 and 50. This would force the signal to stay between 1ms and 2ms. But this is not an acceptable resolution. But it is possible to calibrate the ESC’s so we made them working in a range from 8 to 218. Now the resolutions is fine enough and the ESC’s are working well.