# Data Concept Telemetry AL19

The data from the MS6 is send in CAN messages. You can have multiple data points in one CAN message, so under one address. You must know how the MS6 “multiplexes” the data, so that the Software knows how to interpret the data packages.

The MS6 outputs the data by having a specified range for each variable in every Datagram. So basically, you can have 3 different Variables stored in the CAN ID #77A. The first one from Bit 0 to 16, the second one from Bit 16 to 32 and the last one for example with a length of 4 Bytes from Bit 32 to 64. If the Bytes are Little Endian or Big Endian can be configurated in the MS6 so it doesn’t matter.

The microcontroller receives time updates from the base station in specific intervals and just adds his offset to the latest received time update to generate a consistent timeline. After adding the time to the data package, the microcontroller adds the data with the timestamp to the SD card and the radio module. How the data is written on the SD card is not clear yet, but it should have the same format as before because when you want to get the data from the SD card with an ethernet connection you want to process it the same way you would with the “live” data.

As soon as the data gets onto the base station it will be packaged into single IDs, so that you can get independent packages out of the “multiplexed” CAN message. These packages get logged, so that the logfile consists of only the ID, the value and the timestamp. A configfile must be saved with the logfiles, so that you can always open the old data with a different “ID Setup”. With the help of the config file and your new ID you can add all the other information like the name of the variable, the standard unit and the multiplicator. This version of the data is generated every time a logfile gets loaded into the program. In the live version this is always done every time a data package reaches the program. The GUI gets the data it needs either from the generated Buffer in the offline mode or it gets them from a temporary buffer created the incoming live data.

The GUI is obviously able to choose the data it wants to see in the offline mode. In the online mode it can send the microcontroller a defined message which changes the IDs of the requested live data, so the microcontroller sends the new requested variable and stops sending the old one. Whether the microcontroller stops sending data depends on if there is a limit of variables he can send live or if there are variables that can’t be stopped like Tmot for example.

To conclude the data flow:

* There is a bidirectional communication of sending data to the base station and sending metadata like time updates and variable requests to the microcontroller
* The microcontroller only must calculate and add the time so that he is as fast as possible. The data is then stored into a SD card and some selected data is send over the radio module to the base station.
* The format only changes when the microcontroller is adding the timestamp. The data is logged and sent with the Timestamp. All other data enhancement is done on the base station.