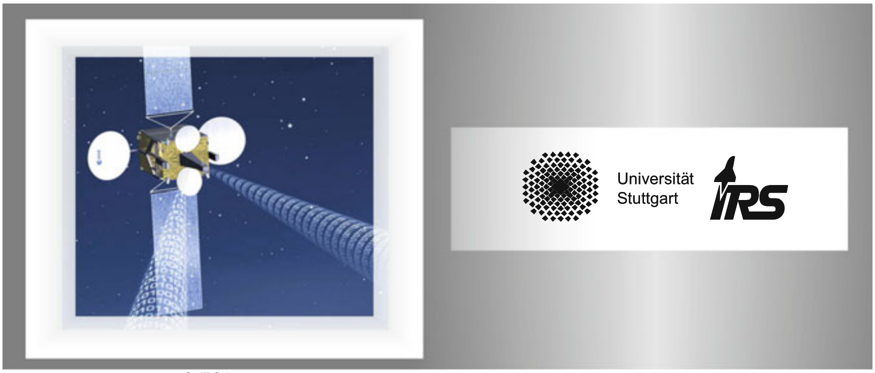
Chapter 2
The FLP Platform Operability

Kai-Sören Klemich and Jens Eickhoff



© ESA

Abstract This chapter provides the basic overview on how an FLP-based spacecraft is operated. It explains the basic system modes and commanded transitions as well as fallback transitions. It explains the table-driven specification of mode-transitions. And it finally details all the command/control services which the platform provides—both standard PUS services, according to ECSS as well as defined private services. Command addressing and telemetry source definition via Virtual Channels and APIDs are also addressed.

Keywords Spacecraft modes · Attitude control subsystem modes · Mode transitions and control tables · Telecommand packet definition · Telemetry packet definition · Virtual channels · Application IDs · ECSS standard PUS services · Private PUS services

K.-S. Klemich (✉)
Institute of Space Systems, University of Stuttgart, Stuttgart, Germany
e-mail: klemich@irs.uni-stuttgart.de

J. Eickhoff
Airbus DS GmbH, Friedrichshafen, Germany
e-mail: jens.eickhoff@airbus.com

J. Eickhoff (ed.), *The FLP Microsatellite Platform*,
Springer Aerospace Technology, DOI 10.1007/978-3-319-23503-5_2

2.1 Spacecraft Configuration Handling Concept

In this section the spacecraft configuration management mechanisms are explained. The configuration comprises the information on

- which devices
  - are active on and are
  - commanded into which mode and
- which software processes
  - are in which mode.

Thus the configuration defines the operational mode of the entire spacecraft. In order to better understand the configuration handling, some concepts of the OBSW are introduced hereafter.

2.1.1 System Mode Changes Using the OBSW Object Hierarchy

Before explaining the platform modes in more detail, the principle of mode transitions and transition commanding shall be explained. See Sect. 3.2 then for further details. The FLP OBSW is based on the object-oriented paradigm and features a hierarchy of "objects". In this hierarchy the *System* object is the top-level one. The modes of the spacecraft correspond 1:1 to the modes of the *System* object.

Below the *System* object (see Fig. 3.3 and Sect. 3.2.6), there exist the subsystem objects (see Sect. 3.2.5), *ACS_Subsystem* (Attitude Control), *PSS_Subsystem* (Power Supply), *TCS_Subsystem* (Thermal Control), *TTC_Subsystem* (Telemetry/ Telecommand), *Payload*. *System* and subsystem objects also are called "high level" objects. Below these subsystem objects are the "low level" objects, which are

- device handlers (see Sect. 3.2.2) which are configuring and communicating with the hardware components,
- assemblies (see Sect. 3.2.3) which manage redundancies of device handlers in case of redundant devices like the RWL and
- controllers (see Sect. 3.2.4) which are software objects using data from the devices to calculate control variables, e.g. the *ACS_Controller* which performs the attitude control.

All of these objects have different operational modes, which are defined by a combination of a mode (ID) and a submode (ID). For device handlers the available modes are defined by the device's internal modes, for controllers the modes are defined by different control strategies, for high level objects the modes are defined as lists of modes of their sub-objects (explained later in Sect. 2.1.12).

If *System* is commanded to a certain mode (e.g. Idle Mode) it commands its sub-objects (i.e. the subsystem objects) to the necessary sub-object modes. For example *ACS* would be commanded to the ACS Idle Mode. Also *PSS* will be commanded to the required submode and similarly *TCS* etc.

The subsystems themselves then command their sub-objects, i.e. device handlers, assemblies and controllers to their necessary modes, e.g. for ACS Idle Mode the reaction wheels must be active and the *ACS_Controller* must be set to a mode in which it applies the Idle Mode control strategy. Following this concept, commanding the *System* object to a mode triggers mode changes of all lower level objects, i.e. of all objects.

2.1.2 Overview on the FLP Operational Modes

The diagram Fig. 2.1 shows the mode transitions of the spacecraft—in the OBSW represented by the modes of the *System* object. All of its lower modes, up to Idle Mode, are closely coupled to the respective *ACS_Subsystem* modes and the *ACS_Controller* modes which share the same mode names. The ACS modes are

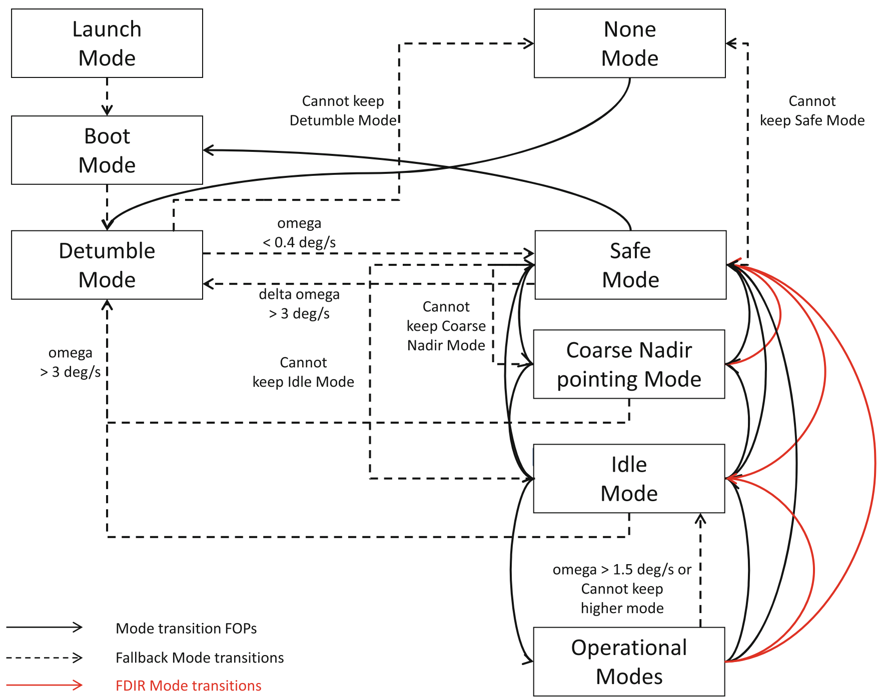


Fig. 2.1 FLP spacecraft modes and transitions. © IRS, University of Stuttgart

illustrated in Fig. 2.2. The different operational modes of the *System* object are explained in the following sections.

The “operational modes” cited in Figs. 2.1 and 2.2 include all payload modes as well as modes where no payload is activated but ACS is in a dedicated pointing mode e.g. target pointing for TM/TC ground station contact. There exist three kinds of mode transitions depicted in the diagram:

- The black continuous arrows show mode transitions which are available as “ Flight Procedures” (see Chap. 15). Note that technically the mode machine in the OBSW does not restrict the modes which are allowed to be commanded. This restriction is induced by the availability of corresponding Flight Procedures. There are procedures to transit from lower to higher modes step by step, but also from Safe Mode to Idle Mode, and to from higher to lower modes in arbitrary steps, i.e. also from the highest modes to Safe Mode.
- The black dashed arrows show fallback mode transitions which are automatically triggered by the mode machine as fallback modes in case where a higher mode cannot be kept. Such a fallback transition is triggered in two cases:
  - A mode cannot be kept or cannot be reached, e.g. because the “Failure Detection, Isolation and Recovery” controller *FDIR* has marked a vital device for this mode as faulty or a vital device goes into an error mode or an

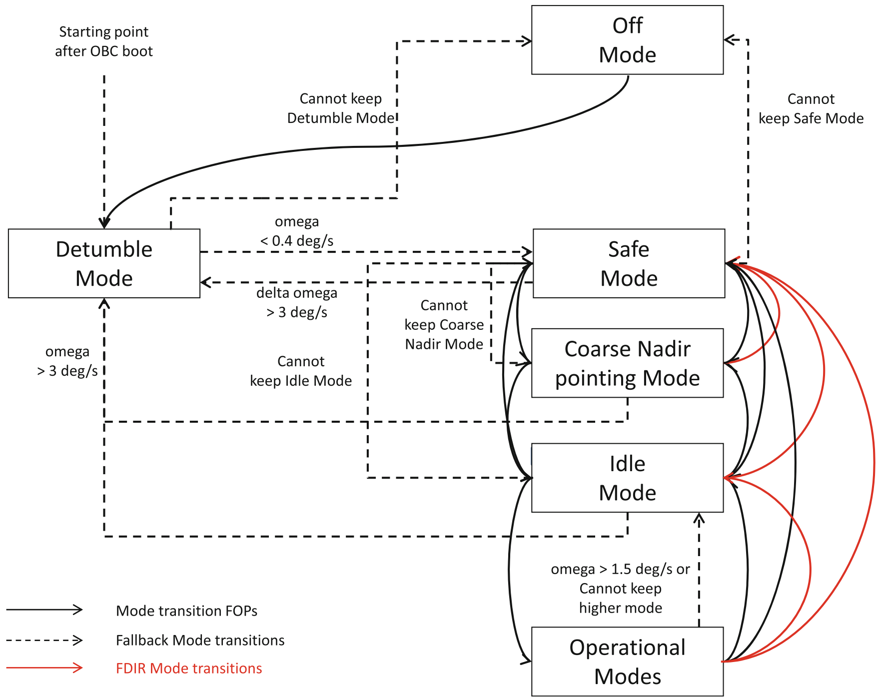


Fig. 2.2 FLP ACS modes and transitions. © IRS, University of Stuttgart

assembly (a logical group of identical devices—like for the 4 reaction wheels) cannot reach its target mode.
  - The *ACS_Controller* detects that the rotational rates or mispointings are too large for the respective mode. In such case, the *ACS_Controller* will notify the *ACS_Subsystem* that it cannot keep its mode which will trigger the fall-back transition of the *ACS_Subsystem*, which in turn will trigger the depicted fallback transition of the System object (i.e. of the entire spacecraft).

- The red arrows show additional transitions triggered by FDIR from higher modes to a lower mode. These are induced in critical cases—for example when the battery State-of-Charge drops below certain limits (see Sect. 10.3).

The FLP is designed for satellites which are launched as secondary passengers and which fulfill the requirement of being entirely switched off until after separation from launcher.

Starting from Launch Mode, where the system is completely switched off, the OBSW performs its boot sequence (Boot Mode) and then starts to detumble the satellite (Detumble Mode). Once the overall rotational rate drops below ca. 0.4°/s, it orients the satellite's solar panels to the sun and spins the satellite about the sun axis (Safe Mode). If the estimated rotational rate diverges from the target rotational rate of 2°/s by more than 3°/s (named "delta omega" in the diagram), the system is set back to Detumble mode.

Higher modes than Safe Mode have to be commanded from ground, which is generally possible in both directions, i.e. from high to low and from low to high. In Coarse Nadir Pointing and Idle mode, the satellite system falls back to Detumble mode when the total rotational rate exceeds 3°/s. The diverse modes are explained in more detail from Sect. 2.1.3 onwards.

Once the satellite has reached an operational mode with a pointing control strategy, Idle Mode serves as an additional fallback option to avoid a transition to the Safe Mode in case of a failure with low severity. This fallback is triggered when the total rotational rate exceeds 1.5°/s or the higher mode cannot be kept, e.g. due to a failure in a payload device. If the failure is more severe, another fallback from Idle to Safe mode will be triggered.

As the flight modes of the satellite are closely related to the corresponding ACS control strategies, the ACS modes and transition diagram as illustrated in Fig. 2.2 looks rather similar.

2.1.3 Launch Mode

The Launch Mode is included in Fig. 2.1 although it does not really exist as operational mode. In the Launch Mode the PCDU is still off—except for the launcher separation detection circuit—and the Onboard Computer (OBC) is still off. Upon detection of the separation from the launcher, the spacecraft transits through the Boot Mode and switches to the Detumble/Safe Mode automatically.

2.1.4 Boot Mode

The Boot Mode in Fig. 2.1 also is no real mode but just an operational transition. It can however be commanded in case where a reboot of the satellite system becomes necessary. As soon as the PCDU's launcher separation circuit detects separation, the PCDU boots and thereafter initiates power-up of diverse equipment for the Safe Mode, namely OBC and Transceivers. The OBC/OBSW then perform the final steps to reach Detumble/Safe Mode. The overall satellite system boot at launcher separation is explained in more detail later in Sect. 4.4.

During boot the PCDU switches on both TTC receivers and both OBC CCSDS-Boards in hot redundancy to avoid single point failures. In addition it activates the power lines for magnetotorquers and magnetometers.

2.1.5 Detumble/Safe Mode

After separation from the launcher, the satellite tumbles with random rates. The actual rates depend on the used launch vehicle and the separation mechanism. During Launch and Early Orbit Phase (LEOP) and if the satellite starts tumbling during nominal operations, the orientation of the solar panels to the Sun and thus the generated energy cannot be predicted and changes very quickly. Therefore the active devices are reduced to a minimum. Only the magnetotorquers are used as actuators to detumble the satellite. Magnetometers and Sun sensors are used as sensor equipment in this mode. In order to achieve a sufficient torque, both, the nominal and redundant coils of the magnetotorquers are used. The solar panel deployment mechanisms will be activated for a maximum of 1 min during LEOP. More details are provided in Sect. 5.3 and in Sect. 11.6.1.

As soon as the total angular rate of the satellite decreases below 0.4°/s, the satellite ACS changes the control strategy from rate damping to Sun orientation (see Sect. 2.3) and spin-up around the −z axis up to a rate of approximately 2°/s (see Figs. 1.9 and 2.3).

The angle between the panel normal and the principal axis of inertia is approximately 8°. The accuracy of the measurement of the direction of the sun by the Sun sensors depends on the direction to the Earth. The maximum error is approximately 20°. Thus the maximum pointing error is in the range of 28°. However due to the spin its mean value is significantly smaller.

Besides the ACS components, only the most vital components are switched on, in order to avoid unnecessary power consumption. Thus, the battery can be recharged and the satellite is kept safely in operation even in case of a loss of control by ground. This mode is also entered in case of a contingency and can be automatically initiated by the FDIR or can be commanded from ground (see Fig. 2.1).

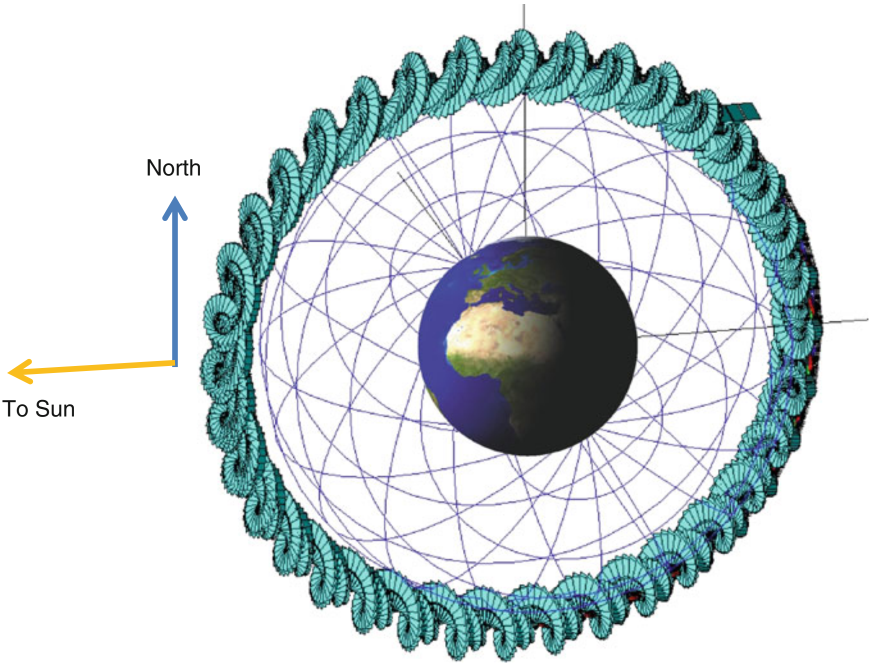


Fig. 2.3 Satellite safe mode. © IRS, University of Stuttgart

From Safe Mode upwards the normal transition is to Idle Mode. The Coarse Nadir Pointing Mode which is illustrated in Fig. 2.1 is used only during the spacecraft Commissioning Phase or during failure debugging (see Sect. 11.7).

2.1.6 Idle Mode

If the satellite operates nominally in Safe Mode and the battery is sufficiently recharged, the Idle Mode can be entered—commanded by telecommand (TC) from ground. In this mode, the solar panels are accurately oriented to the sun using the full capabilities of the ACS, using reaction wheels as actuators and GPS and STRs as sensors. Note that the attitude control algorithms avoid blinding of the STR by the Earth and the Sun at all times. In order to achieve this, the satellite is rotated once per orbit (see Fig. 2.4). No payload equipment is activated in Idle Mode.

From this state, the satellite can be brought into the operational modes like for Earth Observation (Nadir and Target Pointing Mode) or Inertial Pointing Mode for observation of celestial bodies. These operational modes are mission specific.

If an observation or data downlink period is finished and the battery needs to be recharged, the satellite will always be commanded back to Idle Mode.

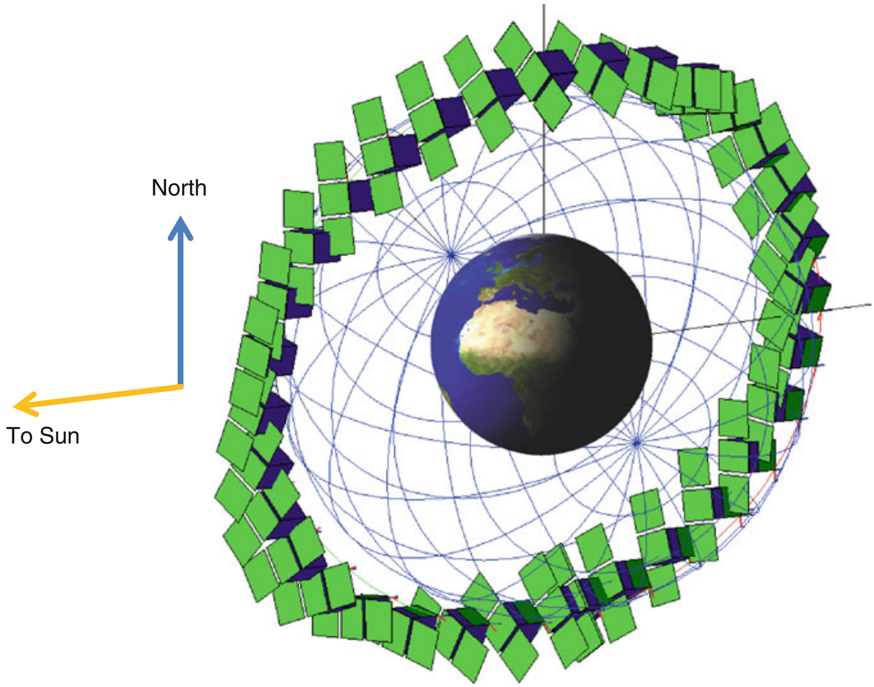


Fig. 2.4 Satellite idle mode. © IRS, University of Stuttgart

2.1.7 Coarse Nadir Pointing Mode

The Coarse Nadir Pointing Mode is an intermediate mode which is used during the spacecraft platform Commissioning Phase. Its control strategy is nadir pointing as this allows to achieve longer ground station contact times as in Safe Mode. However, in Coarse Nadir Pointing Mode the Earth orientation is achieved purely based on magnetotorquer control. The RWs are off when entering the mode from Safe Mode. The Coarse Nadir Pointing Mode is used to safely take the RWs into operations for the first time in the Commissioning Phase—see also Sect. 11.7.

The Coarse Nadir Pointing Mode can also be used for deeper spacecraft inspection in contingency operations to perform ground based FDIR or to perform Earth observation in case the RW cannot be used anymore.

2.1.8 Operational Modes—Ground Contact

In this mode the FLP points to and tracks a ground station antenna by target pointing. On S-band contact establishment the satellite starts to downlink life telemetry (TM), playback telemetry and Event telemetry via its diverse TM virtual channels (for details see Sect. 2.2.1). The satellite can also be commanded by ground via telecommands.

Payload data is transmitted to ground using the science “Data Downlink System” (DDS) of the satellite which is mission specific and can be S-band (as for the first FLP mission of the University), X-Band (as foreseen for FLP Generation 2) or even Ka-Band.

With respect to ACS operation this mode represents a standard ACS target pointing mode as also used for target pointing payload operations—see next section.

2.1.9 Operational Modes—Payload Operations

In these modes, the full ACS capabilities are required to perform platform operations with payload pointing and mission product generation. The PLOC is switched on and it is possible to activate the other payload components according to mission timeline or life manual command. The Data Downlink System (DDS) is not active. As part of this mode, several different sub-modes are possible which depend on the dedicated mission. The FLP as platform supports by default:

- Earth observation in nadir pointing
- Earth observation in target pointing
- Inertial pointing for astronomical observations etc.

The activation of the necessary devices and the switching of the different required ACS modes is controlled by the OBC taking into account the required ACS mode transitions (see Sect. 2.1.12 and following). If the satellite starts spinning with a rate exceeding 1.5°/s during payload operations, it is automatically switched back to the Idle Mode (see Fig. 2.1).

2.1.10 None Mode

The None Mode is a fallback mode for extreme failure situations where the OBSW is not able to control the spacecraft anymore. Only the bare minimum equipment is kept operational which are PCDU, OBC and TTC Receivers and Transmitter. The heater strategy is reduced to battery thermal conditioning. No ACS control is performed. All payloads are disabled. The spacecraft attends ground support. If battery state of charge is decreasing further the PCDU will even shut off TTC and OBC and finally itself. For more details see Sects. 10.2 and 10.3.

2.1.11 Open Mode Concept

On the FLP a so-called open mode concept (as explained in [140]) is applied, which means that the devices and software functions turned on during a specific mode can be modified by telecommands from the ground station. The actual modification can be done for the system and subsystem objects—like the ACS in the example in Sect. 2.1.2. The implemented concept is very flexible and leads to a design, where no PUS[1] (see Sect. 2.4) Service 18 (On-board Control Procedures) is mandatory and no Event-Action Service 19.

For every system or subsystem mode a standard set of mode/submode combinations of the lower level objects is preprogrammed into the OBSW image in the OBC memory. For system level modes, these sets consist of subsystem modes and for subsystem modes, these sets consist of assembly, controller and device handler modes. This way, different configurations of the complete spacecraft are saved. Changes of these predefined mode sets as well as new mode definitions can be commanded at all times via loadable tables (see the following section).

Note that assemblies manage device handlers of redundant devices. For such devices, only the assembly mode is defined as part of the a.m. mode tables and the device modes are controlled by the assembly. This way, switching between redundant units can be performed by setting health flags rather than changing any predefined modes. If a nominal unit is set to faulty, its assembly will automatically switch it off and switch on the redundant unit.

To save the health statuses of all devices a Spacecraft State Vector is stored in dedicated memory areas on the I/O-Boards of the OBC (see Sects. 4.3.2 and 10.2.11). At launch, no component should be defective, so all flags are set to “HEALTHY”. These can be changed automatically to “FAULTY” by the FDIR upon a detection of an error or by a telecommand from ground. However, to change the status of a device back to “HEALTHY”, this is only possible by telecommand from ground—see also Sect. 2.4.16.

2.1.12 Mode Tables and Sequences Control Service

The satellite modes are defined in the OBSW by tables which contain mode/ submode combinations and by sequences which contain sets of tables. In order to manipulate these tables and sequences, a dedicated on-board Packet Utilization Standard (PUS) Service is used, which is referred to as Service 202—see later in Sect. 2.4.17 for details.

[1]ESA Packet Utilization Standard (PUS)—see [13].

Satellite mode target tables are defined for the spacecraft, one for Safe Mode, one for Idle Mode etc. There also may be tables defined for mission specific modes. Mode table comprise n ≥ 0 mode entries. Each entry (a dedicated line in the table) consists of an object-ID as well as the object's mode ID and submode ID.

As example the Idle Mode target table contains mode/submode entries for the STR, the RW—defining these devices to be in On mode. The entry for the Safe Mode target table of the spacecraft will also comprise entry lines for STR and RWs —however defining these devices to be in Off mode.

The concept of mode transitions is explained at hand of Fig. 2.5 and Table 2.1:

A mode transition sequence includes one target table and one or multiple so-called sequence tables. Figure 2.5 depicts the principle of an overall layout as sketch—not for readability. Table 2.1 further below depicts the detailed table for the satellite system modes and transitions. Each subsystem chapter later in this book contains such a table set for the subsystem modes and transitions—see for example Table 6.12. For typesetting layout purposes in this book here the sequence tables are placed below the target table.

- The first table in a sequence—the target table—contains the system/subsystem target mode—i.e. into which status the system shall be brought—plus the intended modes of the child objects.
- A sequence table has in principal the same logic and layout as a target table, but corresponds to an intermediate stage only. Several sequence tables make up a switch sequence, and define each an intermediate mode status of the devices/ assemblies/controller.
- The mode transitions of the next sequence table are started when the transitions of the previous table are finished, plus a configurable time offset (WaitTime). For example the transition of the system to Boot Mode in Table 2.1 shows that 3 sequence tables are to be processed one by one when the system enters transition to boot mode. These sequence tables are executed in the defined order until the target mode settings are reached.
- One target table plus one or more sequence tables corresponds exactly to one mode of the controlled system/subsystem. E.g.—on satellite system level—there will be one such arrangement for Detumble/Safe Mode, one for Idle Mode etc.
- The actual spacecraft/subsystem mode is regularly checked against the target table once the mode transition sequence to the target has finished.

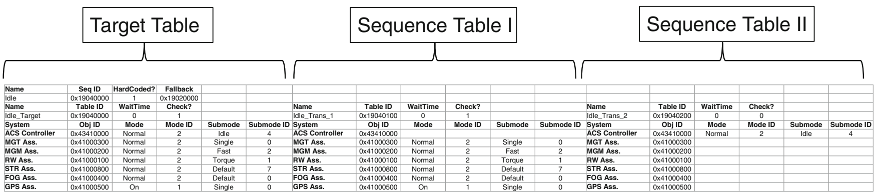


Fig. 2.5 Mode switch tables—sequence sketch. © IRS, University of Stuttgart

Table 2.1 FLP system level modes and transition table

	System Modes	**0x01000100**				
Commanding to None Mode						
#						
Sequence	**Name**	**Seq ID**	**HardCoded?**	**Fallback**		
	None	0x79000000	1			
	Name	**Table ID**	**WaitTime**	**Check?**		
	None_Target	0x79000000	0	0		
	System	**Obj ID**	**Mode**	**Mode ID**	**Submode**	**Submode ID**
	ACS Subsystem	**0x01000200**				
	Power Subsystem	**0x01000300**				
	TTC Subsystem	**0x01000400**				
	TCS Subsystem	**0x01000500**				
	Payload Subsystem	**0x01005000**				
#						
Achieved by transitions						
	Name	**Table ID**	**WaitTime**	**Check?**		
	None_Trans1	0x79000100	0	0		
	System	**Obj ID**	**Mode**	**Mode ID**	**Submode**	**Submode ID**
	ACS Subsystem	**0x01000200**	Off	0x19000000	None	0
	Power Subsystem	**0x01000300**	Off	0x49000000	None	0
	TTC Subsystem	**0x01000400**	Off	0x69000000	None	0
	TCS Subsystem	**0x01000500**	Off	0x59000000	None	0
	Payload Subsystem	**0x01005000**	Off	0x39000000	None	0

(continued)

Table 2.1 (continued)

Commanding to Boot Mode						
#						
Sequence	**Name**	**Seq ID**	**HardCoded?**	**Fallback**		
	Boot	0x79010000	1	0x79000000		
	Name	**Table ID**	**WaitTime**	**Check?**		
	Boot_Target	0x79010000	0	0		
	System	**Obj ID**	**Mode**	**Mode ID**	**Submode**	**Submode ID**
	ACS Subsystem	**0x01000200**				
	Power Subsystem	**0x01000300**				
	TTC Subsystem	**0x01000400**				
	TCS Subsystem	**0x01000500**				
	Payload Subsystem	**0x01005000**				
#						
Achieved by transitions						
	1					
	Name	**Table ID**	**WaitTime**	**Check?**		
	Boot_Trans1	0x79010100	0	1		
	System	**Obj ID**	**Mode**	**Mode ID**	**Submode**	**Submode ID**
	ACS Subsystem	**0x01000200**	Off	0x19000000	None	0
	Power Subsystem	**0x01000300**	Off	0x49000000	None	0
	TTC Subsystem	**0x01000400**	Off	0x69000000	None	0
	TCS Subsystem	**0x01000500**	Off	0x59000000	None	0
	Payload Subsystem	**0x01005000**	Off	0x39000000	None	0

(continued)

Table 2.1 (continued)

	2					
	Name	**Table ID**	**WaitTime**	**Check?**		
	Boot_Trans2	0x79010200	0	1		
	System	**Obj ID**	**Mode**	**Mode ID**	**Submode**	**Submode ID**
	ACS Subsystem	**0x01000200**				
	Power Subsystem	**0x01000300**	Boot	0x49020000	None	0
	TTC Subsystem	**0x01000400**				
	TCS Subsystem	**0x01000500**				
	Payload Subsystem	**0x01005000**				
	3					
	Name	**Table ID**	**WaitTime**	**Check?**		
	Boot_Trans3	0x79010300	0	1		
	System	**Obj ID**	**Mode**	**Mode ID**	**Submode**	**Submode ID**
	ACS Subsystem	**0x01000200**				
	Power Subsystem	**0x01000300**	Off	0x49000000	None	0
	TTC Subsystem	**0x01000400**				
	TCS Subsystem	**0x01000500**				
	Payload Subsystem	**0x01005000**				
#						
Commanding to Detumble/Safe Mode						
#						
Sequence	**Name**	**Seq ID**	**HardCoded?**	**Fallback**		
	Detumble	0x79020000	1	0x79000000		

(continued)

Table 2.1 (continued)

	Name	**Table ID**	**WaitTime**	**Check?**		
	Detumble_Target	0x79020000	0	1		
	System	**Obj ID**	**Mode**	**Mode ID**	**Submode**	**Submode ID**
	ACS Subsystem	**0x01000200**	Detumble	0x19010000	None	0
	Power Subsystem	**0x01000300**	Default	0x49010000	None	0
	TTC Subsystem	**0x01000400**				
	TCS Subsystem	**0x01000500**	Survival	0x59020000	None	0
	Payload Subsystem	**0x01005000**	Off	0x39000000	None	0
#						
Achieved by transitions						
	1					
	Name	**Table ID**	**WaitTime**	**Check?**		
	Detumble_Trans1	0x79020100	0	0		
	System	**Obj ID**	**Mode**	**Mode ID**	**Submode**	**Submode ID**
	ACS Subsystem	**0x01000200**	Detumble	0x19010000	None	0
	Power Subsystem	**0x01000300**	Default	0x49010000	None	0
	TTC Subsystem	**0x01000400**	Standby	0x69020000	None	0
	TCS Subsystem	**0x01000500**	Survival	0x59020000	None	0
	Payload Subsystem	**0x01005000**	Off	0x39000000	None	0
#						
Commanding to Coarse Nadir Mode						
#						
Sequence	**Name**	**Seq ID**	**HardCoded?**	**Fallback**		
	CoarseNadir	0x79040000	1	0x79030000		

(continued)

Table 2.1 (continued)

	Name	**Table ID**	**WaitTime**	**Check?**		
	CoarseNadir_Target	0x79040000	0	1		
	System	**Obj ID**	**Mode**	**Mode ID**	**Submode**	**Submode ID**
	ACS Subsystem	**0x01000200**	CoarseNadir	0x19030000	None	0
	Power Subsystem	**0x01000300**	Default	0x49010000	None	0
	TTC Subsystem	**0x01000400**				
	TCS Subsystem	**0x01000500**	Default	0x59010000	None	0
	Payload Subsystem	**0x01005000**				
#						
Achieved by transitions						
	1					
	Name	**Table ID**	**WaitTime**	**Check?**		
	CoarseNadir_Trans1	0x79040100	0	0		
	System	**Obj ID**	**Mode**	**Mode ID**	**Submode**	**Submode ID**
	ACS Subsystem	**0x01000200**	CoarseNadir	0x69020000	None	0
	Power Subsystem	**0x01000300**	Default	0	None	0
	TTC Subsystem	**0x01000400**	Standby	0	None	0
	TCS Subsystem	**0x01000500**	Default	0	None	0
	Payload Subsystem	**0x01005000**	Off	0	None	0
#						
Commanding to Idle Mode						
#						
Sequence	**Name**	**Seq ID**	**HardCoded?**	**Fallback**		
	Idle	0x79050000	1	0x79030000		

(continued)

Table 2.1 (continued)

	Name	**Table ID**	**WaitTime**	**Check?**		
	Idle_Target	0x79050000	0	0		
	System	**Obj ID**	**Mode**	**Mode ID**	**Submode**	**Submode ID**
	ACS Subsystem	**0x01000200**	Idle	0x19040000	None	0
	Power Subsystem	**0x01000300**	Default	0x49010000	None	0
	TTC Subsystem	**0x01000400**				
	TCS Subsystem	**0x01000500**	Default	0x59010000	None	0
	Payload Subsystem	**0x01005000**				
#						
Achieved by transitions						
	1					
	Name	**Table ID**	**WaitTime**	**Check?**		
	Idle_Trans1	0x79050100	0	0		
	System	**Obj ID**	**Mode**	**Mode ID**	**Submode**	**Submode ID**
	ACS Subsystem	**0x01000200**	Idle	0x19040000	None	0
	Power Subsystem	**0x01000300**	Default	0x49010000	None	0
	TTC Subsystem	**0x01000400**	Standby	0x69020000	None	0
	TCS Subsystem	**0x01000500**	Default	0x59010000	None	0
	Payload Subsystem	**0x01005000**	Off	0x39000000	None	0
#						
Commanding to Ground Contact Mode						
#						
Sequence	**Name**	**Seq ID**	**HardCoded?**	**Fallback**		
	TargetPt_GS	0x79060000	1	0x79050000		

(continued)

Table 2.1 (continued)

	Name	**Table ID**	**WaitTime**	**Check?**		
	TargetPt_GS	0x79060000	0	1		
	System	**Obj ID**	**Mode**	**Mode ID**	**Submode**	**Submode ID**
	ACS Subsystem	**0x01000200**	NadirPt	0x19060000	None	0
	Power Subsystem	**0x01000300**	Default	0x49010000	None	0
	TTC Subsystem	**0x01000400**				
	TCS Subsystem	**0x01000500**	Default	0x59010000	None	0
	Payload Subsystem	**0x01005000**				
#						
Achieved by transitions						
	1					
	Name	**Table ID**	**WaitTime**	**Check?**		
	TargetPt_GS_Trans1	0x79030100	0	0		
	System	**Obj ID**	**Mode**	**Mode ID**	**Submode**	**Submode ID**
	ACS Subsystem	**0x01000200**	TargetPt	0x19060000	None	0
	Power Subsystem	**0x01000300**	Default	0x49010000	None	0
	TTC Subsystem	**0x01000400**	Comm	0x69010000	None	0
	TCS Subsystem	**0x01000500**	Default	0x59010000	None	0
	Payload Subsystem	**0x01005000**	Off	0x39000000	None	0
#						
Commanding to Nadir Pointing EO Mode						
#						
Sequence	**Name**	**Seq ID**	**HardCoded?**	**Fallback**		
	NadirPt	0x790B0000	1	0x79050000		

(continued)

Table 2.1 (continued)

	Name	**Table ID**	**WaitTime**	**Check?**		
	NadirPt_Target	0x790B0000	0	1		
	System	**Obj ID**	**Mode**	**Mode ID**	**Submode**	**Submode ID**
	ACS Subsystem	**0x01000200**	NadirPt	0x19050000	None	0
	Power Subsystem	**0x01000300**	Default	0x49010000	None	0
	TTC Subsystem	**0x01000400**				
	TCS Subsystem	**0x01000500**	Default	0x59010000	None	0
	Payload Subsystem	**0x01005000**				
#						
Achieved by transitions						
	1					
	Name	**Table ID**	**WaitTime**	**Check?**		
	NadirPt_Trans1	0x790B0100	0	0		
	System	**Obj ID**	**Mode**	**Mode ID**	**Submode**	**Submode ID**
	ACS Subsystem	**0x01000200**	NadirPt	0x19050000	None	0
	Power Subsystem	**0x01000300**	Default	0x49010000	None	0
	TTC Subsystem	**0x01000400**	Standby	0x69020000	None	0
	TCS Subsystem	**0x01000500**	Default	0x59010000	None	0
	Payload Subsystem	**0x01005000**			None	0
#						
Commanding to Target Pointing EO Mode						
#						
Sequence	**Name**	**Seq ID**	**HardCoded?**	**Fallback**		
	TargetPt	0x79130000	1	0x79050000		

(continued)

Table 2.1 (continued)

	Name	**Table ID**	**WaitTime**	**Check?**		
	TargetPt_Target	0x79130000	0	1		
	System	**Obj ID**	**Mode**	**Mode ID**	**Submode**	**Submode ID**
	ACS Subsystem	**0x01000200**	TargetPt	0x19060000	None	0
	Power Subsystem	**0x01000300**	Default	0x49010000	None	0
	TTC Subsystem	**0x01000400**				
	TCS Subsystem	**0x01000500**	Default	0x59010000	None	0
	Payload Subsystem	**0x01005000**				
#						
Achieved by transitions						
	1					
	Name	**Table ID**	**WaitTime**	**Check?**		
	TargetPt_Trans1	0x79130100	0	0		
	System	**Obj ID**	**Mode**	**Mode ID**	**Submode**	**Submode ID**
	ACS Subsystem	**0x01000200**	TargetPt	0x19060000	None	0
	Power Subsystem	**0x01000300**	Default	0x49010000	None	0
	TTC Subsystem	**0x01000400**	Standby	0x69020000	None	0
	TCS Subsystem	**0x01000500**	Default	0x59010000	None	0
	Payload Subsystem	**0x01005000**			None	0
#						
Commanding to Inertial Pointing Mode						
#						
Sequence	**Name**	**Seq ID**	**HardCoded?**	**Fallback**		
	InertialPt	0x79070000	1	0x79050000		

(continued)

Table 2.1 (continued)

	Name	**Table ID**	**WaitTime**	**Check?**		
	InertialPt_Target	0x79030000	0	1		
	System	**Obj ID**	**Mode**	**Mode ID**	**Submode**	**Submode ID**
	ACS Subsystem	**0x01000200**	InertialPt	0x19070000	None	0
	Power Subsystem	**0x01000300**	Default	0x49010000	None	0
	TTC Subsystem	**0x01000400**				
	TCS Subsystem	**0x01000500**	Default	0x59010000	None	0
	Payload Subsystem	**0x01005000**				
#						
Achieved by transitions						
	1					
	Name	**Table ID**	**WaitTime**	**Check?**		
	InertialPt_Trans1	0x79070100	0	0		
	System	**Obj ID**	**Mode**	**Mode ID**	**Submode**	**Submode ID**
	ACS Subsystem	**0x01000200**	InertialPt	0x19070000	None	0
	Power Subsystem	**0x01000300**	Default	0x49010000	None	0
	TTC Subsystem	**0x01000400**	Standby	0x69020000	None	0
	TCS Subsystem	**0x01000500**	Default	0x59010000	None	0
	Payload Subsystem	**0x01005000**			None	0
#						

- Thus a mode transition to a new target mode—by ground command, by fallback or triggered by FDIR—starts with the first sequence table of the mode. All necessary equipment/subsystem switching for such a sequence table happens semi-parallel. If a subsystem/equipment already is in the prescribed mode, no switching is necessary. If the CheckFlag is set, the status of all items is checked against the sequence table and if the check is successful, the next sequence table is processed. In a failure case the fallback is triggered. The target table defines the final state to be reached during the mode transition.
- All of these target/sequence tables for the satellite system level mode transitions are included in Table 2.1 at the end of this section.
- As an exception, on system level (Table 2.1), the subsystems are defined as to be commanded "subunits". In the standard subsystem tables like Tables 4.5, 5.8, 6.12, 7.16 and 8.6 the subitems are the subsystem controller and the assemblies and, for those elements where no assembly is available, the devices. Where assemblies are available, only the assemblies are commanded. They command their controlled lower level devices into the right mode.
- Each table has a "WaitTime" in seconds associated, which indicates how long the OBSW shall wait after a sequence table has been commanded until the next one is commanded. In addition, a check flag indicating whether or not the OBSW shall check the successful commanding of all objects in a sequence table can be set. The WaitTime and Check Flag are ignored for target tables. The "HardCoded" flag indicates whether the sequence shall be stored in the OBC safeguard memory to be available after reboot, or whether it s only a handcoded test sequence which will not be used later.
- Also, fallback modes may be given in Table 2.1, which correspond to the fallback transitions depicted in Fig. 2.1. Similarly fallbacks for ACS in the mode transition table of ACS (Table 7.16) correspond to the fallbacks in Fig. 2.2. A fallback in fact indicates a transition to another mode, which means another target/sequence table arrangement. In Table 2.1 for example the fallback of the boot mode is 0x09000000 which is the None Mode. The lowest operational mode of system or a subsystem has no further fallback. The transition to a fallback mode is triggered when an object reports that it cannot keep its mode any longer, e.g. due to a vital device marked faulty, or when a mode transition fails, e.g. due to a device handler not reaching its commanded mode.
- If the status of a subsystem or controller/assembly/device shall not be checked in a certain mode (so that Ground can switch it to arbitrary settings during failure inspection without violating cyclical status checks by the OBSW) the item simply is left out in the mode/sequence table. This feature is also used during platform commissioning, e.g. to test higher ACS devices in Safe Mode without violating any ACS subsystem mode target table mode definitions.

- If the CheckFlag is set, the status of all items is checked against the sequence table and if the check is successful, the next sequence table is processed. In a failure case the fallback is triggered. The target table defines the final state to be reached during the mode transition.
- A transition sequence then can be defined by means of Srv. (202,2) as for example provided in Table 2.1 further below for the satellite system level or Table 6.14 for the power subsystem.

It is important to note that in this processing scheme it may not be a concern from which mode the system switches to a new table. E.g. the **sequence** for the transition to Safe Mode should be designed in a way, that it is of no concern whether the system enters Safe Mode from a higher operational mode by FDIR detected failure, by ground command or by an automatic boot up transition during LEOP.

However, mode transitions may also be designed in such a way that they can only be used in one “direction”, i.e. from higher to lower or vice versa. This becomes necessary when devices like GPS and STR are activated, which need a longer time to get a first measurement. A wrong usage of such transitions is avoided by the mission planning system in which the system mode is tracked and the respective Flight Procedures are provided (see Chaps. 14 and 15).

For such system mode transitions a large number of transitions on subsystem, assembly, controller and device level is performed (also in nominal cases) for which the OBSW generates corresponding Event TM. By this means ground can easily track all the transitions occurring on board.

Note that the TTC subsystem is left out of all system mode target tables in Table 2.1. This is because the system is planned to be operated in such a way that system and TTC subsystem modes are commanded separately. This avoids duplicate system mode definitions for Safe, Idle, Coarse Nadir Pointing and other modes, which can be used both with and without ground station contact. Leaving the TTC subsystem out of the target table allows commanding it independently from the system mode.

2.1.13 Spacecraft Equipment Operation Versus Modes

Table 2.2 (part 1–3) contains a detailed illustration of the diverse satellite subsystems and devices being off or in their diverse operational modes related to the spacecraft modes.

Table 2.2 FLP system modes

System	Object ID	None			0x79000000	Boot			0x79010000	Safe			0x79030000
	(hex)	Mode	#	Submode	#	Mode	#	Submode	#	Mode	#	Submode	#
FDIR Controller	**0x43460000**	Off	0	Off	0	Off	0	Default	0	Normal	2	Default	0
ACS Subsystem	**0x01000200**	Off	0x19000000			Off	0x19000000			Safe	0x19020000		
ACS Controller	**0x43410000**	Off	0	Off	0	Off	0	Off	0	Normal	2	Safe	2
MGT Ass.	**0x41000300**	Off	0	Off	0	Off	0	Off	0	Normal	2	Multi	1
MGM Ass.	**0x41000200**	Off	0	Off	0	Off	0	Off	0	Normal	2	Fast	2
RW Ass.	**0x41000100**	Off	0	Off	0	Off	0	Off	0	Off	0	Off	0
STR Ass.	**0x41000800**	Off	0	Off	0	Off	0	Off	0	Off	0	Off	0
FOG Ass.	**0x41000400**	Off	0	Off	0	Off	0	Off	0	Off	0	Off	0
GPS Ass.	**0x41000500**	Off	0	Off	0	Off	0	Off	0	Off	0	Off	0
CDH Subsystem	**0x01000600**	Default	0x29000000			Default	0x29000000			Default	0x29000000		
I/O-Board Ass.	**0x41010000**	On	1	Single	0	On	1	Single	0	On	1	Single	0
Power Subsystem	**0x01000300**	Off	0x49000000			Off	0x49000000			Default	0x49010000		
Power Controller	**0x43500000**	Off	0	Off	0	Off	0	Off	0	Normal	2	Default	0
PCDU	**0x44003200**	Normal	2	Idle	0	Normal	2	Idle	0	Normal	2	Idle	0
TTC Subsystem	**0x01000400**	Off	0x69000000			Off	0x69000000			Standby	0x69020000		
Communic. Controller	**0x43520000**	Off	0	Off	0	Off	0	Off	0	Normal	2	Standby	1
Rx Assembly	**0x41000600**	On	1	Default	0	On	1	Default	0	Normal	2	Default	0
Tx Assembly	**0x41000700**	Off	0	Off	0	Off	0	Off	0	Off	0	Off	0
CCSDS-Board Ass.	**0x41100000**	On	1	Passive	0	On	1	Passive	0	On	1	Active	1
TCS Subsystem	**0x01000500**	Off	0x59000000			Off	0x59000000			Default	0x59010000		
TCS Controller	**0x43540000**	Off	0	Off	0	Off	0	Off	0	Normal	2	Auto	0
Payload Subsystem	**0x01005000**	Off	0x39000000			Off	0x39000000			Off	0x39000000		
Payload Controller	**0x43700000**	Off	0	Off	0	Off	0	Off	0	Off	0	Off	0
PL1	**0x44501000**	Off	0	Off	0	Off	0	Off	0	Off	0	Off	0
PL2	**0x44500F00**	Off	0	Off	0	Off	0	Off	0	Off	0	Off	0
PL3	**0x44500B00**	Off	0	Off	0	Off	0	Off	0	Off	0	Off	0
PL4	**0x44500E00**	Off	0	Off	0	Off	0	Off	0	Off	0	Off	0
DDS Ass.	**0x41010200**	Off	0	Off	0	Off	0	Off	0	Off	0	Off	0
PLOC Ass.	**0x41010100**	Off	0	Off	0	Off	0	Off	0	Off	0	Off	0

(continued)

Table 2.2 (continued)

System	Object ID	CoarseNadir			0x79040000	Idle			0x79050000	TargetPt_GS			0x79060000
	(hex)	Mode	#	Submode	#	Mode	#	Submode	#	Mode	#	Submode	#
FDIR Controller	**0x43460000**	Normal	2	Default	0	Normal	2	Default	0	Normal	2	Default	0
ACS Subsystem	**0x01000200**	CoarseNadir	0x19030000			Idle	0x19040000			TargetPt	0x19060000		
ACS Controller	**0x43410000**	Normal	2	CoarseNadir	3	Normal	2	Idle	4	Normal	2	TargetPt	6
MGT Ass.	**0x41000300**	Normal	2	Single	0	Normal	2	Single	0	Normal	2	Single	0
MGM Ass.	**0x41000200**	Normal	2	Fast	2	Normal	2	Fast	2	Normal	2	Fast	2
RW Ass.	**0x41000100**	Off	0	Off	0	Normal	2	Torque	1	Normal	2	Torque	1
STR Ass.	**0x41000800**	Normal	2	Default	7	Normal	2	Default	7	Normal	2	Default	7
FOG Ass.	**0x41000400**	Off	0	Off	0	Normal	2	Default	0	Normal	2	Default	0
GPS Ass.	**0x41000500**	On	1	Single	0	On	1	Single	0	On	1	Single	0
CDH Subsystem	**0x01000600**	Default	0x29000000			Default	0x29000000			Default	0x29000000		
I/O-Board Ass.	**0x41010000**	On	1	Single	0	On	1	Single	0	On	1	Single	0
Power Subsystem	**0x01000300**	Default	0x49010000			Default	0x49010000			Default	0x49010000		
Power Controller	**0x43500000**	Normal	2	Default	0	Normal	2	Default	0	Normal	2	Default	0
PCDU	**0x44003200**	Normal	2	Idle	0	Normal	2	Idle	0	Normal	2	Idle	0
TTC Subsystem	**0x01000400**	Standby	0x69020000			Standby	0x69020000			Comm	0x69010000		
Communic. Controller	**0x43520000**	Normal	2	Standby	1	Normal	2	Standby	1	Normal	2	Comm	2
Rx Assembly	**0x41000600**	Normal	2	Default	0	Normal	2	Default	0	Normal	2	Default	0
Tx Assembly	**0x41000700**	Off	0	Off	0	Off	0	Off	0	On	1	Default	0
CCSDS-Board Ass.	**0x41100000**	On	1	Active	1	On	1	Active	1	On	1	Active	1
TCS Subsystem	**0x01000500**	Default	0x59010000			Default	0x59010000			Default	0x59010000		
TCS Controller	**0x43540000**	Normal	2	Auto	0	Normal	2	Auto	0	Normal	2	Auto	0
Payload Subsystem	**0x01005000**	Off	0x39000000			Off	0x39000000			Off	0x39000000		
Payload Controller	**0x43700000**	Off	0	Off	0	Off	0	Off	0	Off	0	Off	0
PL1	**0x44501000**	Off	0	Off	0	Off	0	Off	0	Off	0	Off	0
PL2	**0x44500F00**	Off	0	Off	0	Off	0	Off	0	Off	0	Off	0
PL3	**0x44500B00**	Off	0	Off	0	Off	0	Off	0	Off	0	Off	0
PL4	**0x44500E00**	Off	0	Off	0	Off	0	Off	0	Off	0	Off	0
DDS Ass.	**0x41010200**	Off	0	Off	0	Off	0	Off	0	Off	0	Off	0
PLOC Ass.	**0x41010100**	Off	0	Off	0	Off	0	Off	0	Off	0	Off	0

(continued)

Table 2.2 (continued)

System	Object ID	InertialPT			0x79070000	NadirPt			0x790B0000	TargetPt			0x79130000
	(hex)	Mode	#	Submode	#	Mode	#	Submode	#	Mode	#	Submode	#
FDIR Controller	**0x43460000**	Normal	2	Default	0	Normal	2	Default	0	Normal	2	Default	0
ACS Subsystem	**0x01000200**	InertialPt	0x19080000			NadirPt	0x19050000			TargetPt	0x19060000		
ACS Controller	**0x43410000**	Normal	2	InertialPt	7	Normal	2	NadirPt	5	Normal	2	TargetPt	6
MGT Ass.	**0x41000300**	Normal	2	Single	0	Normal	2	Single	0	Normal	2	Single	0
MGM Ass.	**0x41000200**	Normal	2	Fast	2	Normal	2	Fast	2	Normal	2	Fast	2
RW Ass.	**0x41000100**	Normal	2	Torque	1	Normal	2	Torque	1	Normal	2	Torque	1
STR Ass.	**0x41000800**	Normal	2	NEO	71	Normal	2	Default	7	Normal	2	Default	7
FOG Ass.	**0x41000400**	Normal	2	Default	0	Normal	2	Default	0	Normal	2	Default	0
GPS Ass.	**0x41000500**	On	1	Single	0	On	1	Single	0	On	1	Single	0
CDH Subsystem	**0x01000600**	Default	0x29000000			Default	0x29000000			Default	0x29000000		
I/O-Board Ass.	**0x41010000**	On	1	Single	0	On	1	Single	0	On	1	Single	0
Power Subsystem	**0x01000300**	Default	0x49010000			Default	0x49010000			Default	0x49010000		
Power Controller	**0x43500000**	Normal	2	Default	0	Normal	2	Default	0	Normal	2	Default	0
PCDU	**0x44003200**	Normal	2	Idle	0	Normal	2	Idle	0	Normal	2	Idle	0
TTC Subsystem	**0x01000400**	Standby	0x69020000			Standby	0x69020000			Standby	0x69020000		
Communic. Controller	**0x43520000**	Normal	2	Standby	1	Normal	2	Standby	1	Normal	2	Standby	1
Rx Assembly	**0x41000600**	Normal	2	Default	0	Normal	2	Default	0	Normal	2	Default	0
Tx Assembly	**0x41000700**	Off	0	Off	0	Off	0	Off	0	Off	0	Off	0
CCSDS-Board Ass.	**0x41100000**	On	1	Active	1	On	1	Active	1	On	1	Active	1
TCS Subsystem	**0x01000500**	Default	0x59010000			Default	0x59010000			Default	0x59010000		
TCS Controller	**0x43540000**	Normal	2	Auto	0	Normal	2	Auto	0	Normal	2	Auto	0
Payload Subsystem	**0x01005000**	InertialPt	0x39090000			NadirPt	0x39010000			TargetPt	0x39070000		
Payload Controller	**0x43700000**	Normal	2	Default	0	Normal	2	Default	0	Normal	2	Default	0
PL1	**0x44501000**	Off	0	Off	0	Off	0	Default	0	Normal	2	Default	0
PL2	**0x44500F00**	Off	0	Off	0	Normal	2	Default	0	Off	0	Default	0
PL3	**0x44500B00**	Off	0	Off	0	Normal	2	Default	0	Off	0	Default	0
PL4	**0x44500E00**	Normal	2	Default	0	Off	0	Default	0	Off	0	Default	0
DDS Ass.	**0x41010200**	Off	0	Off	0	Off	0	Default	0	Off	0	Default	0
PLOC Ass.	**0x41010100**	Off	0	Off	0	Normal	2	Default	0	Normal	2	Default	0

2.2 Spacecraft Telecommand and Telemetry Structure

Figures 2.6 and 2.7 illustrate the structure of the FLP telecommands and telemetry respectively from packet layer to transmission layer according to the CCSDS standards [15–27]. The exact Virtual Channel settings, Packet length settings, CRC usage etc. can be taken from these figures.

2.2.1 CCSDS Protocol Addressing

As defined in the CCSDS standards [18, 27], several different IDs are defined to identify the channel and the origin of a TM packet or the destination of a TC packet (please also refer to the Figs. 2.6 and 2.7). The highest level of the channels is the physical channel, identified by its name. Physical channels are typically used for different ground/space links, i.e. for missions with multiple spacecraft and/or multiple ground stations.

For the FLP mission “Flying Laptop” there is only one spacecraft and only one ground station involved at a time (either Weilheim or Stuttgart) so there is no need to assign a specific ID to the Physical Channel.

The physical channel consists of several Master Channels identified by the Master Channel ID (MCID). The MCID consists of the Transfer Frame Version Number (TFVN, see Sect. 2.5) and the Spacecraft ID (SCID, see Sect. 2.2.2), both of which are given in the Transfer Frame Header. As for the FLP platform the TFVN is always “00” (in binary).

A Master Channel consists of several Virtual Channels, identified by the Global Virtual Channel ID (GVCID). The GVCID consists of the MCID and the Virtual Channel ID (VCID). As mentioned before, the MCID is fix for the “Flying Laptop” mission. Different VCID sets exist for uplink and downlink:

Uplink

For the uplink the VCID determines whether the command is a High Priority Command (HPC) or not and to which OBC CCSDS-Board the command is directed (nominal or redundant). The FLP platform uplink VCID definitions are provided in Table 2.3.

For the uplink only—according to the CCSDS standard—a Virtual Channel can consist of several Multiplexer Access Point (MAP) channels, identified by a Global Multiplexer Access Point ID (GMAPID) according to [27]. The GMAPID consists of the GVCID and the MAP ID.

For the FLP platform, there are no MAPs defined, so the MAPID is fixed to 0 for all commands. The TC routing is performed through the APID mechanism provided by the Packet Utilization Standard (PUS)—see Sect. 2.4. The APID definitions are provided in Sect. 2.3.

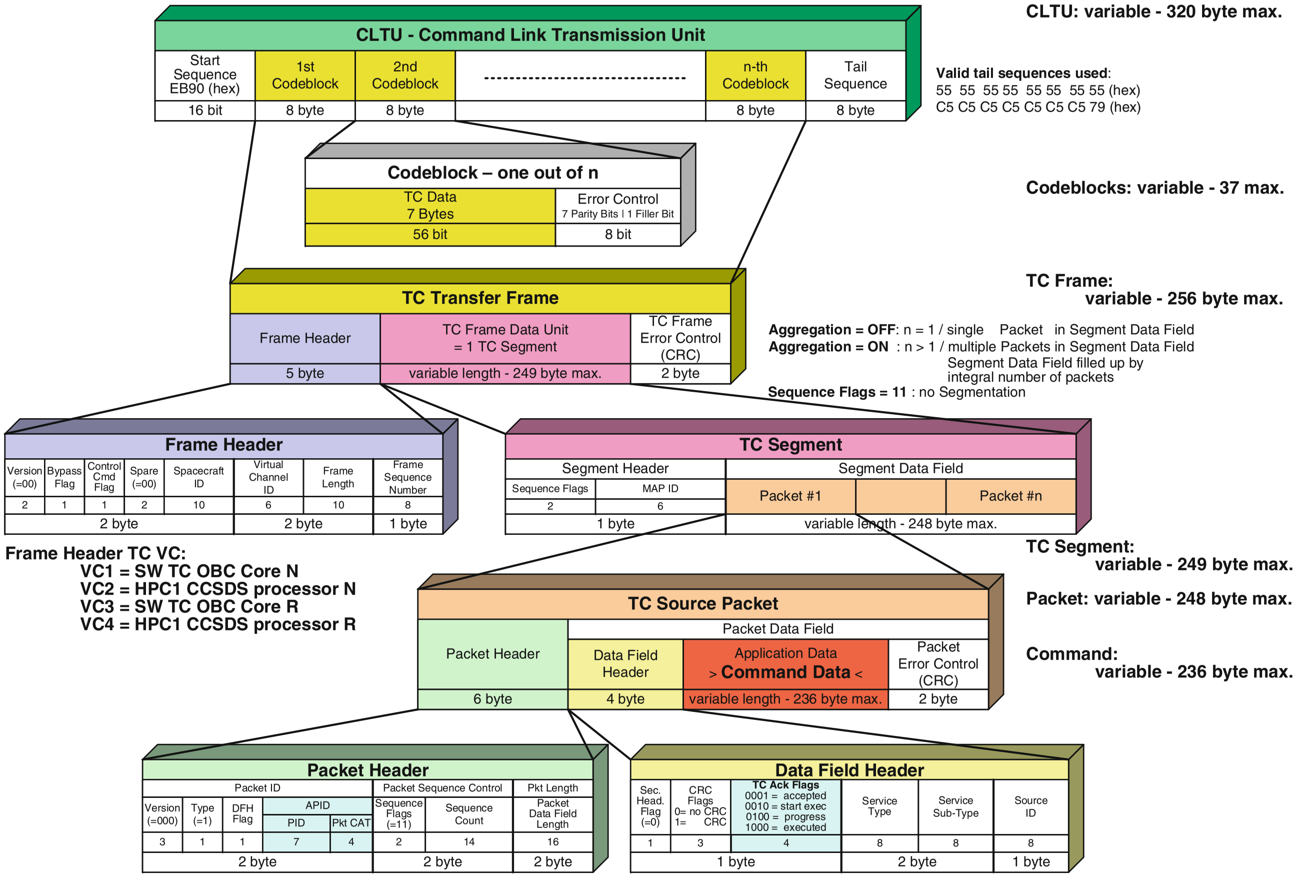


Fig. 2.6 Telecommand packet definition. © IRS, University of Stuttgart, Template © Airbus DS

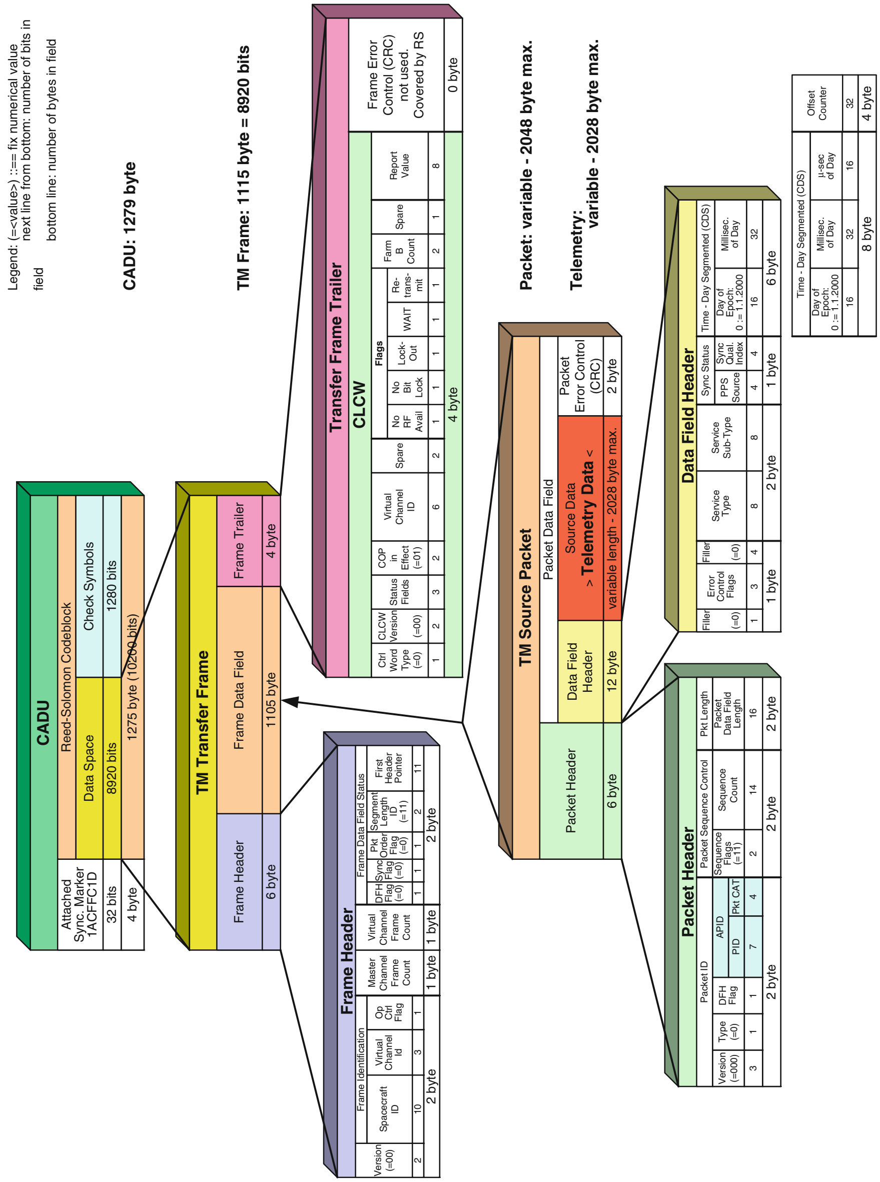


Fig. 2.7 Telemetry packet definition. © IRS, University of Stuttgart, Template © Airbus DS

Table 2.3 Uplink VCIDs overview

TC VCIDs	CCSDS-Board	Command Taype
VC1	CCSDS N	Nominal High Priority TC
VC2		Nominal standard TC
VC3	CCSDS R	Redundant High Priority TC
VC4		Redundant standard TC

Table 2.4 Downlink VCIDs overview

TM VCIDs	Command Type
VC0	Live TM
VC1	Stored TM: Srv 3 Housekeeping/Diagnistic packets
VC2	Stored TM: Srv 5 Event packets
VC3	Stored TM: all other packets

Downlink

For the downlink the VCID determines the origin of the TM, i.e. whether it is live TM or not and if not, from which packet store it originates (see Sects. 2.4.11 and 4.3.2). Thus the VC also determines the priority of the submitted TM frames and thus the routing priority in the CCSDS-Board multiplexer. Four Virtual Channels can be determined according to the origin of the TM, as summarized in Table 2.4.

2.2.2 *Spacecraft ID*

Each spacecraft transmitting CCSDS protocol in the S-band range reserved for space communication, needs an official registered spacecraft ID number. The official CCSDS Spacecraft ID (SCID) assigned to the FLP mssion "Flying Laptop" is "001001011101" in binary or "25D" in hexadecimal numbers. It will appear in every TM- and TC-Frame Header. It was requested by M. Pilgram of DLR/GSOC and was initially assigned on August 1st, 2011. The entry in the official database of registered spacecraft Space Assigned Number Authority (SANA) [46] is provided in Table 2.5:

Table 2.5 Spacecraft ID overview

Spacecraft Name	Channel	Version	Id (hex)	Requestor Name	Requestor Affiliation	Requestor Affiliation Country	Assigned Date	Assignor	Status
FLYINGLAPTOP	TC, TLM	1	25D	M. Pilgram	DLR	DE	2011-08-01	JGG	Assigned

2.2.3 *System Authentication*

The FLP core data handling subsystem (Core DHS) provides a simplified authentication concept to prevent satellite hijacking. This is developer intellectual property and is not subject to this published FOM release.

2.3 Application Process ID Definitions (APIDs)

The CCSDS Space Packet Protocol Standard [20] defines the lower protocol structure, namely the TM or TC packets (see Sect. 2.5). Within this standard, the Application Process ID (APID) is defined as the main ID to define the origin of a TM packet respectively the target of a TC packet. The APID range 2040–2047 is however reserved by certain standards (see [21]).

The APID can be used to address different parts of the OBSW or different devices. It needs to be noted that a device which has its own APID must be a PUS terminal. Thus it has to be able to receive/interprete and to send PUS compatible packets. This requires a certain integrated "intelligence" in such a device. For the FLP only the STR has this capability and thus its own APID. All other devices are controlled by the OBSW and addressed using the OBSW APID.

Inside the FLP OBSW the diverse processes do not have individual APIDs—so there exist no diverse Process IDs (PRID) and Categories (PCAT). The entire OBSW has a single APID. For the defined APIDs see Table 2.6:

Table 2.6 FLP APIDs definition

Process/Hardware	APID (dez)	APID (bin)	Remarks
Idle Packet	2047	11111111111	Defined in [20]
OBSW general	53	00000110101	
Star Tracker	1937	11110010001	
Simulator	1804	11100001100	Fixed, as compiled in simulator kernel (source code IP of Airbus DS)
TMTCFE Remote Control	1540	11000000100	Configurable

2.4 PUS Tailoring Concept

The ECSS Standard ECSS-E-70-41A "Ground systems and operations—Telemetry and telecommand packet utilization" [13] defines the so-called "Packet Utilization Standard" (PUS) Services. Within these PUS Services generic functionalities are defined on how telecommands and telemetry can be utilized for spacecraft control, and for on-board and on-ground data handling, telecommand verification etc. Each service includes several subservices which can either be a request to a service provider, i.e. an on-board application process, or a report from the provider. Often a report is defined as a "reply" to a service request, the request being a telecommand (TC) which is uplinked and the report being a telemetry (TM) packet which is downlinked.

The ECSS standard defined services are numbered from 1 to 19, with the services 7, 10 and 16 currently being reserved but unused [13]. Not all services and subservices defined by the ECSS need to be implemented for a dedicated mission. Additional services may be defined by a spacecraft manufacturer for special mission purposes with service numbers above 127. For self-defined services and for standard services also additional subservices may be defined by a spacecraft manufacturer. Several of these additional services are defined for the FLP. For each service a "minimum capability" set of subservices is defined, which are considered mandatory if the service itself is implemented.

Due to the generic nature of the PUS standard, there are no dedicated services for specific equipment operations, but standard PUS Services can be extended for such purposes. Examples for such extensions are specific payload functions, specific failure handling functions or specific onboard data handling functions.

For these reasons of mission specific services and mission specific additional subservices the PUS standard needs to be tailored for each mission. In the following sections all services and all corresponding subservices are described which are implemented in the FLP Core Data Handling Subystem (Core DHS). For each service used on the FLP, a table of all subservices is provided at the end of each section. The implementation of the telecommand and telemetry packets and variables in the Flight Control System's database—the so-called Mission Information Base (MIB)—is covered in detail in Chap. 16.

The FLP Onboard Software by default supports the following standard PUS Services:

- Service 1: Acknowledge Service
- Service 2: Device Commanding Service
- Service 3: Housekeeping Service
- Service 5: Event Reporting Service
- Service 6: Memory Management Service
- Service 8: Function Service
- Service 9: Time Management Service
- Service 11: On-board Operations Scheduling Service
- Service 12: On-Board Monitoring Service

- Service 15: On-board Storage and Retrieval Service
- Service 17: Test Service
- Service 20x: Self-defined services for mode commanding.

The following sections describe the FLP PUS implementation. For the detailed TC/TM/Event lists refer to the annex Sect. 17.3.

2.4.1 PUS-Service 1—Telecommand Verification

Service 1 defines the reporting of telecommand reception and execution in various stages (see also Table 2.7). In general, telecommands shall be verified upon reception on board the satellite. When a command is received, a Telecommand Acceptance Report is sent to Ground—Service (1,1) for successful reception—respectively (1,2) in case of failed reception or corrupted command. After a command has been successfully executed or in case where an error occurred during the execution, a Telecommand Execution Completed Report—Service (1,7) respectively (1,8)—is be generated. The execution of more complex commands can be acknowledged by sending execution start and execution progress success and failure reports—Service (1,3)—(1,6). All of these report types are implemented in the FLP Core DHS.

Table 2.7 PUS-Service 1—telecommand verification

Nr.	Description	TM/TC
(1,1)	Telecommand Acceptance Report—Success	TM
(1,2)	Telecommand Acceptance Report—Failure	TM
(1,3)	Telecommand Execution Started Report—Success	TM
(1,4)	Telecommand Execution Started Report—Failure	TM
(1,5)	Telecommand Execution Progress Report—Success	TM
(1,6)	Telecommand Execution Progress Report—Failure	TM
(1,7)	Telecommand Execution Completed Report—Success	TM
(1,8)	Telecommand Execution Completed Report—Failure	TM

2.4.2 PUS-Service 2—Device Command Distribution

PUS-Service 2—see Table 2.8—is used for direct commanding of on-board devices (equipment—like a star tracker, a reaction wheel etc.). It is used mainly for testing, internal OBSW commanding and contingency situations. During nominal operations, Service 2 commands can be used e.g. to request and report settings or other data from devices, which is not part of the housekeeping TM or to trigger certain rarely used functions:

Table 2.8 PUS-Service 2—device command distribution

Nr.	Description	TM/TC
(2,128)	Distribute Raw Device Commands	TC
(2,129)	Raw Device Command Report	TM
(2,130)	Distribute Direct Device Commands	TC
(2,132)	Distribute Channel Switch Command	TC
(2,133)	Device Wiretapping Request	TC
(2,134)	Device Wiretapping TC Report	TM
(2,135)	Direct Device Command Report	TM

Subservice (2,128) is used to command a device (on-board equipment) using a command structure which is understood by the device itself ("raw commanding"). The command is directly routed to the device by the device handler, if the handler is in raw mode (see Sect. 3.2.2). Subservice (2,129) is the report answer to the (2,128) request, containing the actual answer packet from the commanded device. It is also directly routed to the ground by the device handler without any interpretation.

Due to the complexity of the PCDU, this functionality is mainly used during the PCDU verification, which is why only the Service (2,129) reports from the PCDU are interpreted by the Mission Control System (MCS).

Subservice (2,130) is used to command a device by a simpler command, which is "known" by the OBSW ("direct commanding"). Thus, it is not in a format understood by the target device, but is interpreted by the device handler in the OBSW. Furthermore, the reply packet of the device is also interpreted by the device handler. To use this Subservice, the device handler must be in On or Normal mode (see Sect. 3.2.2). Note that not all commands of a given device are necessarily implemented in the OBSW so that they can be sent by Service (2,130). For the other commands, Service (2,128) must be used.

TM answers to direct commands of type (2,130) containing requests of TM from devices are downlinked as Subservice (2,135). This can be used to downlink TM parameters, which are not requested by the device handlers on a regular basis. This includes mainly status variables which do not change frequently (as for example bi-stable relays states in the PCDU) and which are not used by any on-board controller. All values requested by service (2,130) are not written into the OBSW *datapool* (see Sect. 4.3) and cannot be downlinked using PUS Service 3 (Housekeeping).

In general, Service 2 is mainly used for troubleshooting during testing and possibly on orbit. However, using Service (2,130) commands to receive (2,135) TM is part of nominal operations.

Subservice (2,133) is used to enable and disable a so-called "wiretapping mode" of a device handler. In this mode, the device handler downlinks all communication between a device and the device handler. Commands from the handler to the device are downlinked as Service (2,134) packets and answer packets from the device as Service (2,129) (see above). This service is intended to be used for debugging purposes during tests and possibly in orbit. Wiretapping can be activated in any device handler mode (see Sect. 3.2.2). However, it needs to be used carefully as large amounts of data might be generated, possibly spamming the downlink channel, especially when the handler is in on or normal mode. It can be useful to enable the service shortly before a to be debugged command is sent, and to disable it shortly after command execution and error recording. Generally, the packets are however not interpreted within the Mission Control System but their TM (see service (2,135) explained below) needs to be interpreted manually.

Service (2,3) is a subservice which is used on board of standard spacecraft for emergency commanding. They are used to send pulse commands to the so-called Command Pulse Distribution Unit (CPDU) to switch units on or off, directly from ground (see also [140]). Due to the special Combined Data and Power Management Infrastructure (CDPI) on the FLP here the High Priority Commands (HPC) here are distinguished by their virtual channels, provide more sophisticated features and are not PUS compliant. Thus there is no need for implementation of Service (2,3).

2.4.3 PUS-Service 3—Housekeeping and Diagnostic Reporting

This service provides the possibility to report housekeeping (HK) and diagnostic data to ground. The term HK data refers to nominal telemetry in the PUS, whereas diagnostic data is telemetry sent in case of anomalies occurring within the HK data.

Both the nominal HK and diagnostic data can be downlinked in periodic or filtered mode. In periodic mode, the packets are generated at a predefined sampling rate, whereas in filtered mode, packets are only generated if predefined thresholds are exceeded. However, the filtered mode is not used on the FLP.

Within the service a list of HK and diagnostic parameter reports is maintained (see Table 2.9), in which sampling rates and parameters to be sampled are defined. The first eight subservices of this service will be used on the FLP in order to define new (3,1)/(3,2) or to erase old (3,3)/(3,4) HK or diagnostic parameter reports and to enable (3,5)/(3,7) or to disable (3,6)/(3,8) the generation of HK and diagnostic parameter reports.

Subservices (3,9) to (3,12) provide the possibility to downlink the HK and diagnostic parameter report definitions in order to double check the definitions actually used on board.

Subservices (3,25)/(3,26) define reports for HK and diagnostic parameters. These subservices are the minimum capability of the service.

There exist predefined housekeeping packets for the FLP, containing the *datapool* parameters of a single subsystem or device. Every *datapool* variable can be found in at least one (3,25) packet.

Table 2.9 PUS Service 3—housekeeping and diagnostics data handling

(3,1)	Define new Housekeeping Parameter Report	TC
(3,2)	Define New Diagnostic Parameter Report	TC
(3,3)	Clear Housekeeping Parameter Report Definitions	TC
(3,4)	Clear Diagnostic Parameter Report Definitions	TC
(3,5)	Enable Housekeeping Parameter Report Generation	TC
(3,6)	Disable Housekeeping Parameter Report Generation	TC
(3,7)	Enable Diagnostic Parameter Report Generation	TC
(3,8)	Disable Diagnostic Parameter Report Generation	TC
(3,9)	Report Housekeeping Parameter Report Definitions	TC
(3,10)	Housekeeping Parameter Report Definitions Report	TM
(3,11)	Report Diagnostic Parameter Report Definitions	TC
(3,12)	Diagnostic Parameter Report Definitions Report	TM
(3,25)	Housekeeping Parameter Report	TM
(3,26)	Diagnostic Parameter Report	TM

2.4.4 PUS-Service 4—Parameter Statistics Reporting

This service provides the possibility to report only statistics of certain TM (minimum, maximum, mean values, standard deviation) in order to reduce the amount of data to be downlinked. This service is not implemented in the Core DHS of the FLP.

2.4.5 PUS-Service 5—Event Reporting

This service is used to report event TM to Ground. Event TM can be generated in case of anomalies and errors or if certain autonomous on-board actions have been carried out or milestones have been reached. There are four different kinds of event reports defined—subservices (5,1) to (5,4) for

- normal/progress reports for autonomous on-board actions or milestones—Service (5,1) and for
- error/anomaly reports with three different levels of severity, namely low, medium and high—Services (5,2) to (5,4).

Subservices (5,5) and (5,6) are used to activate/deactivate the Event generation on board in flight—individually for the diverse report types by their Report-ID (RID). This is useful in case of a device sending corrupted TM or when priority of certain reports change during flight.

For the FLP it needs to be noted that when an Event is disabled by its RID, it is disabled for all on-board objects, which are able to generate it.

Table 2.10 PUS Service 5—Event reporting service

(5,1)	Normal/Progress Report	TM
(5,2)	Error/Anomaly Report - Low Severity	TM
(5,3)	Error/Anomaly Report - Medium Severity	TM
(5,4)	Error/Anomaly Report - High Severity	TM
(5,5)	Enable Event Report Generation	TC
(5,6)	Disable Event Report Generation	TC
(5,128)	Inject Event for Test Purposes	TC

All Service 5 subservices are used on the FLP (see Table 2.10).

The private subservice Service (5,128) can be used to trigger reporting of a certain event. This is used exclusively for test purposes.

2.4.6 PUS-Service 6—Memory Management Service

This service is used to directly access different parts of the on-board memory. It is commonly used in case of OBSW updates/patches or STR star-map patches or in contingency situations, in which certain parts of the memory of a PUS commandable device/software needs to be dumped or overwritten directly.

However, for the FLP this service was extended to enable higher level data access. For example, it is possible to set configuration variables of controllers—like the *ACS_Controller* with its control laws for steering the satellite or the power supply subsystem's *PSS_Controller* with its variables for battery State-of-Charge computation. For more background on the controllers see Sect. 3.2.8.

This access to variables is done by using the controller object-ID as a memory ID and the ID of the parameter as the memory address. This allows performing high level data access operations within the standard PUS framework.

In general, there are two different ways to access the memory, either by an absolute memory address or by using a base address plus an offset. All subservices are defined for both ways. For FLP, only the absolute addressing is used.

The handling of memory management is somewhat similar to the access of device specific information via Service 2. Service 6 however is not used for accessing the OBSW device handlers and the devices, but the assemblies and controllers. The similarity is depicted below (see also Table 2.11):

- TC Service to set a value of a device: Srv. (2,130)
- TC Service to query a value of a device: Srv. (2,130)
- TM Service with the requested values of a device: Srv. (2,135)
- TC Service to set a value of a controller: Srv. (6,132)
- TC Service to query a value of a controller: Srv. (6,135)
- TM Service with the requested values of a controller: Srv. (6,136)

Parameter setting in Service 6 is for one parameter at a time. Setting of multiple parameters requires multiple individual Service 6 TCs.

Table 2.11 PUS-Service 6—memory management service

(6,2)	Load Memory using Absolute Addresses	TC
(6,5)	Dump Memory using Absolute Addresses	TC
(6,6)	Memory Dump using Absolute Addresses Report	TM
(6,9)	Check Memory using Absolute Addresses	TC
(6,10)	Memory Check using Absolute Addresses Report	TM
(6,132)	Load Memory using Absolute Addresses customized	TC
(6,135)	Dump Memory using Absolute Addresses customized	TC
(6,136)	Memory Dump using Absolute Addresses Report customized	TM

2.4.7 PUS-Service 8—Function Management Service

This service has only one subservice defined in the PUS called "Perform Function" (see Table 2.12). It is used to invoke all sorts of non-standard functions, e.g. payload commanding, mode switches etc.

Table 2.12 PUS-Service 8—function management service

(8,1)	Perform Function	TC
(8,129)	Request Data Function Report	TM

Service (8,1) is used to set groups of often used parameters in controllers, e.g. all parameters for the battery State-of-Charge estimation with a single command, and to query the data. The answers to request commands are downlinked using Service (8,129).

As the FLP uses the mode switching concept based on the mode tables and sequence tables described in Sect. 2.1.12 it does not need a full implementation of the classic PUS Service 8.

2.4.8 PUS-Service 9—Time Management Service

This service can be used to report the on-board time to the ground to coordinate it with UTC (9,2) and to change the rate with which this report is generated (9,1).

On the FLP a clock supplied by the OBC operating system counts the seconds from a defined starting point (OBC boot). This service is used to link this seconds counter with UTC by uplinking the current UTC as Service (9,128) TC—see Table 2.13. Once the GPS system is on, an on-board time in UTC format is

Table 2.13 PUS Service 9—time management service

(9,1)	Change Time Report Generation Rate	TC
(9,128)	Set Time Request	TC

available from the GPS time packets on board and GPS time is used. This time code is more precise than any OBC OS seconds counter based on-board time.

When the GPS system is switched off (safe mode, detumble mode), the correlation between the on-board time counter and UTC can still be done using any live TM packet containing a time stamp. Thus the standard subservices (9,1) and (9,2) are not essential and are not implemented in the FLP Generation 1 OBSW. In the PUS it is explicitly stated that the concepts for missions using a GPS system (like FLP spacecraft) are not covered.

2.4.9 PUS-Service 11—On-board Operations Scheduling Service

Within this PUS Service a command schedule is maintained, which consists of loadable telecommands to be executed at a specified time. According to the ECSS standard [14] such a schedule can be divided into subschedules, which each can contain a dedicated commands or Flight Procedures for example. For the first FLP mission at the University of Stuttgart, however subschedules are not supported by the older version of the Mission Control System (MCS).

Via PUS Service 11 it is also possible to downlink the contents of the schedule to double check its integrity. Within the standard, the possibilities for summarized and detailed reports of the contents of the schedule are provided.

The service—as implemented for the FLP—consists of subservices to enable (11,1) and to disable (11,2) the release of commands from the schedule, to reset the entire schedule (11,3) and to insert (11,4) commands into the schedule. These are all part of the minimum capability set and are available on the FLP—see Table 2.14.

It is only possible to enable the schedule if the system time has been set. When inserting new telecommands into the mission timeline, the following rules apply:

- New commands may be inserted both when the schedule is enabled and disabled.
- Telecommands do not need to be ordered according time tag during uplink/load. The scheduling service orders them by time-tag.
- If a system time is set, the service rejects TCs which are already outdated at time of insertion. Otherwise, insertion is possible.

Table 2.14 PUS-Service 11—On-board operations scheduling service

(11,1)	Enable Release of Telecommands	TC
(11,2)	Disable Release of Telecommands	TC
(11,3)	Reset Command Schedule	TC
(11,4)	Insert Telecommands in Command Schedule	TC

In the FLP Generation 1 OBSW the schedule is stored in the OBC RAM, therefore it is lost in case of a system reboot. Resetting the schedule (11,3) works as specified in the ECSS standard. The service disables the schedule in case of a time jump, as defined in the PUS standard.

More sophisticated subservices for reporting, time shifting of commands etc. are not yet implemented—also due to the fact that the payload management in FLP Generation 1 via the PLOC is somewhat limited compared to the architecture later targeted for FLP Generation 2—see Sect. 9.3.

2.4.10 PUS-Service 12—On-board Monitoring Service

This Service is used to monitor on-board variables in order to generate Event TM or to notify the *FDIR* in the OBSW if specified parameter limits are exceeded. The generated Monitoring Reports contain details on all limit transitions since the last time the same Report was issued, including how long and how often limits were exceeded. The service also maintains a list of parameters to be monitored and the applicable limits.

PUS Service 12 implementation is also minimized and tailored for its implementation on FLP Generation 1. As the parameter monitoring is performed already in the device handlers and controllers of the object-oriented OBSW (see Sect. 3.2), it is sufficient to provide the modification service and transition report (12,12)—see Table 2.15. The only other standard service is the definition of the maximum reporting delay via Service (12,3).

Besides the PUS Service to modify the parameter checking information (=set limits) (12,7), there is a custom subservice (12,137) to be used for specific objects with additional limits, e.g. the battery strings which have a maximum spatial gradient limit.

Monitoring of parameters in a specific object, e.g. a controller, can also be enabled and disabled by Services (12,131) and (12,132) respectively.

Table 2.15 PUS-Service 12—On-board monitoring service

(12,3)	Change maximum reporting delay	TC
(12,7)	Modify Parameter Checking Information	TC
(12,12)	Check Transition Report	TM
(12,131)	Enable Monitoring of Parameters within one object	TC
(12,132)	Disable Monitoring of Parameters within one object	TC
(12,137)	Modify Parameter Checking Information for objects with additional limits	TC

2.4.11 PUS-Service 15—On-board Storage and Retrieval Service

This service maintains a list which TM packet is to be stored in which memory area (packet store). There exist three packet stores in the OBC I/O-Board, one for Service 3 HK and diagnostic packets, one for Service 5 event TM and one for all other TM (see Sects. 2.4.11 and 4.3.2). The service is used to organize storage of the packets during times of no ground station access and to retrieve the packets in order to downlink them—see also Table 2.16:

Table 2.16 PUS_Service 15—On-board storage and retrieval service

(15,1)	Enable Storage in Packet Stores	TC
(15,2)	Disable Storage in Packet Stores	TC
(15,3)	Add Packets to Storage Selection Definition	TC
(15,4)	Remove Packets from Storage Selection Definition	TC
(15,5)	Report Storage Selection Definition	TC
(15,6)	Storage Selection Definition Report	TM
(15,7)	Downlink Packet Store Contents for Packet Range	TC
(15,10)	Delete Packet Stores Contents up to Specified Packets	TC
(15,12)	Report Catalogues for Selected Packet Stores	TC
(15,13)	Packet Store Catalogue Report	TM

2.4.12 PUS-Service 17—Test Service

This service is a simple “ping” service, designed to test the connection between the ground station and the spacecraft during tests, during LEOP and potentially at the beginning of each ground station contact. It consists of a request to perform a connection test (17,1) and a connection test report (17,2) as a reply to the request (see Table 2.17). If both packets are received correctly by each side, the connection is established successfully. Both subservices are implemented on in the FLP Core DHS.

Table 2.17 PUS-Service 17—test service

(17,1)	Perform Connection Test (“Ping”)	TC
(17,2)	Connection Test Report	TM

2.4.13 PUS-Service 18—On-board Control Procedures Service

This service can be used to define On-board Control Procedures by means of a predefined scripting language interpreter in the OBSW. Such a feature is not directly provided by the FLP OBSW.

A similar functionality as what OBCPs provide w.r.t. sequence control is provided by the custom service 202 Sequences Control Service (Sect. 2.4.17).

2.4.14 PUS-Service 19—Event-Action Service

Due to the object-oriented OBSW design and the encapsulation of functions the reaction chains on certain symptom occurrences are not implemented via Event-Action chains but by handling through the next higher OBSW instance.

On symptom occurrence Events are generated as in a classic procedural software design for information of the flight operators and for function triggering in the next higher object instance.

Therefore the classic Event-Action Service is not needed for the FLP OBSW.

2.4.15 PUS Service 200—Mode Control Service

The PUS service 200 is custom defined for the FLP mode management and thus replaces certain functionalities of a classical PUS Service 8. As shown in Table 2.18, Service (200,1) is used to command any "object" of the satellite system, which has different modes (see Sect. 3.1.2), in one of its modes—identified by a combination of mode and submode.

An object can be any software or hardware component which is identified by an object-ID (see Sect. 3.2) and has different modes, like

- device handlers,
- assemblies,
- controllers,
- subsystems or
- the satellite system object.

Subservices (200,2) and (200,3) are used to query the mode of an object or an object and all of its children recursively, e.g. all ACS objects for the ACS

Table 2.18 PUS-Service 200—mode control service

(200,1)	Set Object Mode	TC
(200,2)	Request Object Mode	TC
(200,3)	Request Object Mode Recursively	TC

subsystem. There are no report answers defined for these subservices, because the replies are provided in the form of Event TM via Service 5 (see Sect. 2.4.5).

2.4.16 PUS-Service 201—Health Flag Control Service

The PUS service 201 is tailored for the FLP to control the health states of the devices on-board the satellite, which have an associated health flag (for more details see Sect. 3.1.2). As shown in Table 2.19, there is a subservice to set an object's health flag (201,1). The subservice (201,4) is used to query the health flag of an object.

The TM of this request is returned in form of Service 5 Event TM (see Sect. 2.4.5), similar to services (200,2) and (200,3). There exist also subservices to request and report the health flags of all objects having a health flag assigned.

Table 2.19 PUS-Service 201—health flag control service

(201,1)	Set Object Health Flag	TC
(201,2)	Request All Object Health Flags	TC
(201,3)	Report All Object Health Flags	TM
(201,4)	Request Single Object Health Flag	TC

2.4.17 PUS-Service 202—Mode Tables and Sequences Control Service

The PUS Service 202 is tailored for managing the FLP spacecraft high level object modes. These comprise the

- satellite *System* object modes (which correspond 1:1 to the spacecraft modes) and
- the subsystem modes of all the spacecraft subsystems. The latter are also represented by a control object in the hierarchy of the OBSW (see later in Sect. 3.1.1).

This custom PUS Service is used to control the tables and sequences that are used to change the S/C system or a subsystem level mode and which were introduced already in Sect. 2.1.12.

The definitions of subsystem/assembly/device modes versus S/C modes—see Table 2.2—are defined in internal "Mode Tables" which are preloaded in the OBSW ROM image and are reloadable/modifiable via this service from ground.

In general, these tables contain several entries of mode commands for invoking the system transition from one mode to the next—similar to Service (200,1) requests. Functional sequences (sequence tables) on board can contain a number of such mode commands in a table to be executed sequentially and to achieve a mode change on subsystem or even system level.

Table 2.20 PUS-Service 202—mode tables and sequences service

(202,1)	Add Mode Change Table	TC
(202,2)	Add Mode Change Sequence	TC
(202,3)	Delete Mode Change Table	TC
(202,4)	Delete Mode Change Sequence	TC
(202,5)	Request Mode Change Tables Available	TC
(202,6)	Report Mode Change Tables Available	TM
(202,7)	Request Mode Change Sequences Available	TC
(202,8)	Report Mode Change Sequences Available	TM
(202,9)	Request Mode Change Table Data	TC
(202,10)	Report Mode Change Table Data	TM
(202,11)	Request Mode Change Sequence Data	TC
(202,12)	Report Mode Change Sequence Data	TM
(202,13)	Request Available Memory Slots for Mode Change Tables and Sequences	TC
(202,14)	Report Available Memory Slots for Mode Change Tables and Sequences	TM
(202,15)	Request Available Memory Slots for Mode Change Tables of an Object	TC
(202,16)	Report Available Memory Slots for Mode Change Tables of an Object	TM
(202,17)	Request Available Memory Slots for Mode Change Sequences of an Object	TC
(202,18)	Report Available Memory Slots for Mode Change Sequences of an Object	TM

As listed in Table 2.20, there are subservices implemented to add such tables[2] and sequences[3] (202,1), (202,2), to delete them (202,3), (202,4), to request and report lists of available tables and sequences (202,5) to (202,8). Furthermore, there exist subservices

- to request and report the data contained within these tables (202,9) to (202,12) and
- to request and report the available memory slots for tables and sequences (202,13) to (202,14)
- and to request and report the available memory slots for tables or sequences of a single object via the subservices (202,15) to (202,18).

2.4.18 *PUS-Service 203—Payload Control Service*

The FLP OBSW by default provides a dedicated PUS Service for interfacing a payload controller if implemented in the spacecraft. In the case of the first FLP

[2]target tables in the context described in Sect. 2.1.12.

[3]sequence tables in the context described in Sect. 2.1.12.

based satellite based—the “Flying Laptop”—this custom Service is used for controlling the Payload Control Computer (PLOC).

Details on the specific implementation of payload commands please are to be defined in a mission specific Payload Flight Operations Manual.

2.5 Spacecraft Commandability and Observability

As mentioned the communication between spacecraft and ground station applies the CCSDS standards. Telecommand packets are segmented and framed and converted to Command Link Transmission Units (CLTU) in the ground segment. The CLTUs are modulated, transferred to the spacecraft and demodulated on board by the receivers. The demodulated bitstream is forwarded to the CCSDS-Boards. The CCSDS-Boards extract the TC frames from the CLTUs by performing all the necessary decoding and unpacking. The on-board software cyclically acquires the TC frames from the locked CCSDS-Board (see Chap. 6) and unpacks the TC packets to process them.

As explained in more detail in Sect. 6.1 the FLP is equipped with two CCSDS-Boards, a nominal and a redundant one. The CCSDS-Boards are operated in hot redundancy. When ground contact is achieved the first board reporting receiver lock and carrier lock status to the OBSW will be used to downlink TM unless being marked non-healthy by *FDIR* or by ground. The board used to process the uplinked commands is selected by ground via the TC Virtual Channel (see Sect. 2.2.1).

As the transmitter itself is operated in cold redundancy, there will never be two different signals sent at the same time. During nominal operations, the nominal CCSDS-Board is used. When no TM is received on ground, it is possible that the CCSDS-Board in use is marked as non-healthy or is not working correctly. A switchover has to be commanded.

Spacecraft telemetry also follows the CCSDS standards and transmission is again performed via the CCSDS-Boards to the active transmitter (see Fig. 6.1). TM frames including the TM packets are submitted to the CCSDS-Boards by the OBSW via separate TM Virtual Channels for life TM, and different kinds of playback TM (see Sect. 2.2.1).

The active CCSDS-Board performs the diverse encoding down to Channel Acquisition Data Units (CADU) which are forwarded to the active transmitter and which are modulated there. When ground connection is established and currently no TM is downlinked by the OBSW the CCSDS-Board generates Idle Frames to keep the data connection to Ground alive.

The diverse PUS services for TC and TM that are supported by the FLP have been explained in the previous chapter.

High Priority Commands of class 1 (HPC1)—which bypass the OBC and OBSW—can be sent to the spacecraft and are identified in the CCSDS-Board—by Virtual Channel ID, not by MAP-ID (see Sect. 2.2.1). They are sent to the PCDU directly from the CCSDS-Boards and directly processed by the Combined-Controller in the PCDU—see Sect. 4.2 and Figs. 4.1 and 4.2.

The FLP Generation 1 does not provide High Priority TM replies for these HPC1 commands.

The mode control service Srv.(200) and the mode table management via Srv.(202) allow a flexible adaptation of the spacecraft transition sequences by upload of new mode tables also in flight. This obsoletes a complex OBCP interpreter engine in the OBSW and supports the concept of an easy to use low-cost platform.

With this set of features the commandability and observability of the spacecraft platform in nominal and contingency situations is sufficiently complete.

2.6 Spacecraft On-board Time Management

On-board time management is performed via PUS Service 9 (Sect. 2.4.8). In the FLP platform, the On-Board Computer (OBC), the GPS system and the Star Tracker (STR) share time information.

The FLP platform does not utilize a dedicated real-time clock which keeps time and date information available at any time, but relies on the time information delivered by ground or by the GPS system. There exist two independent pulse-per-second (PPS) lines, one from the GPS to the OBC and one from the OBC to the STR (see Fig. 2.8). In fact electrically they all are redundant but only one of each pair of redundant lines is active at a time.

A *Time_Manager* object in the OBSW is responsible for collection and distribution of timestamps. A time tag is allocated to the incoming pulse from the GPS, which, together with the absolute time information in the GPS time packet, can be used to set the on-board time in the OBC to an absolute value.

Similarly, the Time_Manager forwards the absolute time of the outgoing pulse to the STR unit, which automatically uses this information to compute and report its attitude information with the OBC on-board time.

The Time Manager in the OBC handles two different time domains:

- After reboot, it uses the CPU uptime as time reference. To maintain a monotonic clock, the current uptime value is stored in the OBC PROM (see Sect. 4.3.1) in regular intervals. It is retrieved at boot time.
  By this means a system restart with a runtime smaller than the previous one is avoided.

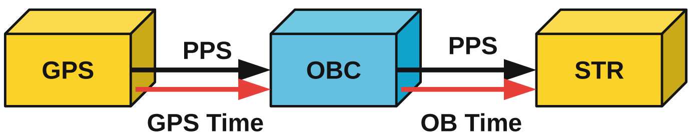


Fig. 2.8 Time distribution in the spacecraft

- An absolute time in (CCSDS CDS format [26]) is used as soon as the time is set by ground command or has been acquired from GPS.

A precise synchronization (in the range of a few ms) to the GPS pulse-per-second is possible with this set-up, but is subject to later implementation in the FLP Generation 2. The current implementation implies jumps in the on-board time when switching from uplinked UTC time to GPS time.

http://www.springer.com/978-3-319-23502-8

The FLP Microsatellite Platform
Flight Operations Manual
Eickhoff, J. (Ed.)
2016, XXVI, 680 p. 266 illus., 56 illus. in color.,
Hardcover
ISBN: 978-3-319-23502-8