digitalSTROM Basic Concepts

digitalSTROM

Version: v1.3-branch*

August 19, 2015

*Revision: 3c451f5c0c98db7edb9555940b5215106499d5d1

digitalSTROM Alliance
Building Technology Park Zurich
Brandstrasse 33
CH-8952 Schlieren
Switzerland

Contents

1 Introduction

1.1 Introduction to the digitalSTROM basics

This document describes the basic system concepts of the digitalSTROM system. Intended audience are Hardware and Software developers that want to create applications that interact with digitalSTROM installations. Sample applications can be:

- Smartphone applications that control digitalSTROM installations
- Scripts running in the digitalSTROM server that extend the functionality of the system or interface to cloud based services
- Integration of 3rd party hardware with digitalSTROM

The document assumes that the user has experience using the digitalSTROM components, either in an installation or through a development kit. The document only covers the software aspects of the digitalSTROM technology.

Information about planning, installation and usage can be found in the following link:
http://www.digitalstrom.com/Partner/Support/Infocenter/?.

1.2 Organization of this document

The information in this document is presented in the following chapters:

- "Structure Objects" describes the topology of an electrical installation from the digitalSTROM point of view, see 2
- "Event Objects" presents the different layers where events occur in the system and their handling, see 3
- "Scenes and Activities" introduces the set of commands available within the digitalSTROM system, and describes their specific scopes, see 4
- "Communication Objects" details the basic communication concepts within a digitalSTROM installation, see 7
- "Actions" List the commands that are understood by all digitalSTROM enabled devices, see 5
- The glossary and appendixes provide detailed information about parts of the system, see 10
- "Certification rules" List the rules that all applications or devices have to comply to be "digitalSTROM ready", see 11

2 Structure Objects

The following sections describe the structural basis of a digitalSTROM installation. The structure objects need to be seen from a logical point of view. Although these objects represent real parts of the building they are defined to have a simple model of the installation. This is the basis for ease of use and Plug'n'Play mechanisms.

2.1 Circuit

The power circuit is the natural basic structure element of an installation. The circuit is the physical connection between the circuit breaker on one end and each electric device connected on this line on the other end.

2.2 Zone

A zone is the logical representation of the rooms, halls and other structural works of the building. In digitalSTROM installations a zone is a logical combination of one or more power circuits or of one more subsets of a power circuit.

In the simple case a power circuit completely covers solely one room, in this scenario there is a single zone object defined for this room which consists of the whole circuit and its electric devices. A power circuits often spans over multiple rooms, then the physical circuit can be split into multiple logical zones corresponding to the users usage behavior.

2.3 Group

A group is the application class of devices. digitalSTROM defines application groups to reflect the common usage of devices in buildings. Those predefined groups allow to have a simple system setup as all digitalSTROM Ready Devices have their standard group preconfigured.

All devices of the same application within the same zone are operated together per default.

Some applications are not existing in the context of a single zone, but only in the context of the whole house or apartment, e.g. ringing door bells.

Group ID	Name	Color	Application
1	Lights	Yellow	Room lights
2	Blinds	Gray	Blinds or shades outside
12	Curtains	Gray	Curtains and blinds inside
3	Heating	Blue	Heating
9	Cooling	Blue	Cooling
10	Ventilation	Blue	Ventilation
11	Window	Blue	Windows
48	Temperature Control	Blue	Single room temperature control
4	Audio	Cyan	Playing music or radio
5	Video	Magenta	TV, Video
8	Joker	Black	Configurable
n/a	Single Device	White	Various, individual per device
n/a	Security	Red	Security related functions, Alarms
n/a	Access	Green	Access related functions, door bell

Table 1: digitalSTROM application groups and their colors

Devices assigned to the function "Joker" are customizable and in most cases the hardware allows them to be used with different consumer devices. For example the relay adapter plug or a wall push button input line can be assigned by the user to that application that they are used for.

Devices of the category "Single Device" are not grouped in general. There are devices for which a group behavior and collective operation is not wanted anyhow. Common examples are vacuum cleaner, coffee machine, or the garage door.

The security and access related functions are global apartment-wide applications and therefore do not have a status or relation to a particular zone.

2.4 Device

A digitalSTROM Device is an device with well defined system behavior and with digitalSTROM communication capabilities. The digitalSTROM Device is the combination of a terminal block with an attached native device. The digitalSTROM Device is from the users point of view the complete piece of equipment, including all visible output values and input capabilities. This definition includes that a digitalSTROM Device may be build out of multiple terminal blocks and components.

The individual functions of digitalSTROM Device are called Submodule. The Submodules are logically related to the functional groups depending on their use. Each submodule of a digitalSTROM Device belongs exactly to one functional group and zone.

A digitalSTROM Ready Device is a device which complies to all rules of the digitalSTROM concept.

Rule 1 A digitalSTROM Ready Device has to be preconfigured in the right application group. This is essential to ensure that all electrical devices in one application group can be orchestrated together.

Rule 2 A digitalSTROM Ready Device must be configured for exactly one digitalSTROM application group. The assigned application group must be non-ambiguous and is part of the static device configuration.

Rule 3 The function of a devices output is the basis of its group membership. For devices without output function the target function of the switch button decides about the group membership.

The concept of digitalSTROM Devices is independent from the underlying technology of the communication hardware. Thus the digitalSTROM system represents any kind of hardware equipment with the same methods, on condition that the hardware has communication capabilities and can be uniquely identified. For example technologies and products like the digitalSTROM Powerline Terminal Blocks, TCP/IP network equipment, radio equipment, or in general components of other bus communication systems can be seamlessly integrated with the appropriate gateway functionality.

2.5 Device Identification

digitalSTROM Terminal Blocks are uniquely identified by a 32 bit serial number. This number is stored in non-volatile memory on the device and can be read programmatically by the system.

Terminal Blocks itself are described by a global trade item number (GTIN, see glossary, 10.10). If the digitalSTROM Device is combined with the Native Device this new article shall get a new GTIN, additionally to that one of the Native Device. This new GTIN can be stored on and read from the Terminal Block.

The serial number of the Terminal Block and the GTIN of the article are then combined to build a so-called serialized global trade item number (SGTIN). The GTIN and SGTIN are standardized formats for identifying trade items, this standard is written by and under control of the GS1 organization.

Following the SGTIN-96 standard the digitalSTROM system identifies any digitalSTROM Device with an unique number called dSID (see glossary 10.4).

2.6 Area

An area is built by a subset of devices within a zone. This is a user definable set of devices. This is achieved by individual device configuration and is not based on an addressing mechanism.

2.7 Cluster

A cluster is built by a subset of devices of the same group from any zone. This is a user definable set of devices. All devices in a cluster share something common, for example Clusters can be defined for:

- all shades in the first floor
- the shades with orientation to the south
- all lights that are part of the christmas illumination

The devices of a Cluster are still part of their respective zone and are under control of the room group operations.

2.8 Server

The digitalSTROM Server is the appliance where user specific extensions and adjustments to the digitalSTROM system behavior can be implemented. These adjustments are individual to each user and therefore not part of the system behavior. More details of the user customization is in chapter 3.3. For example the digitalSTROM Server Addon *User Defined Actions* gives the user advanced control over certain events and how his digitalSTROM installation should behave.

Furthermore the server appliance allows the user to have advanced automation and personalized functionality.

There is exactly one instance of server per digitalSTROM installation. The server is acting as a gateway between the digitalSTROM components and the network or Internet and makes the digitalSTROM system accessible from virtually anywhere.

2.9 Apartment

The apartment is the logical totality of all digitalSTROM components connected together in one installation.

2.10 Topology

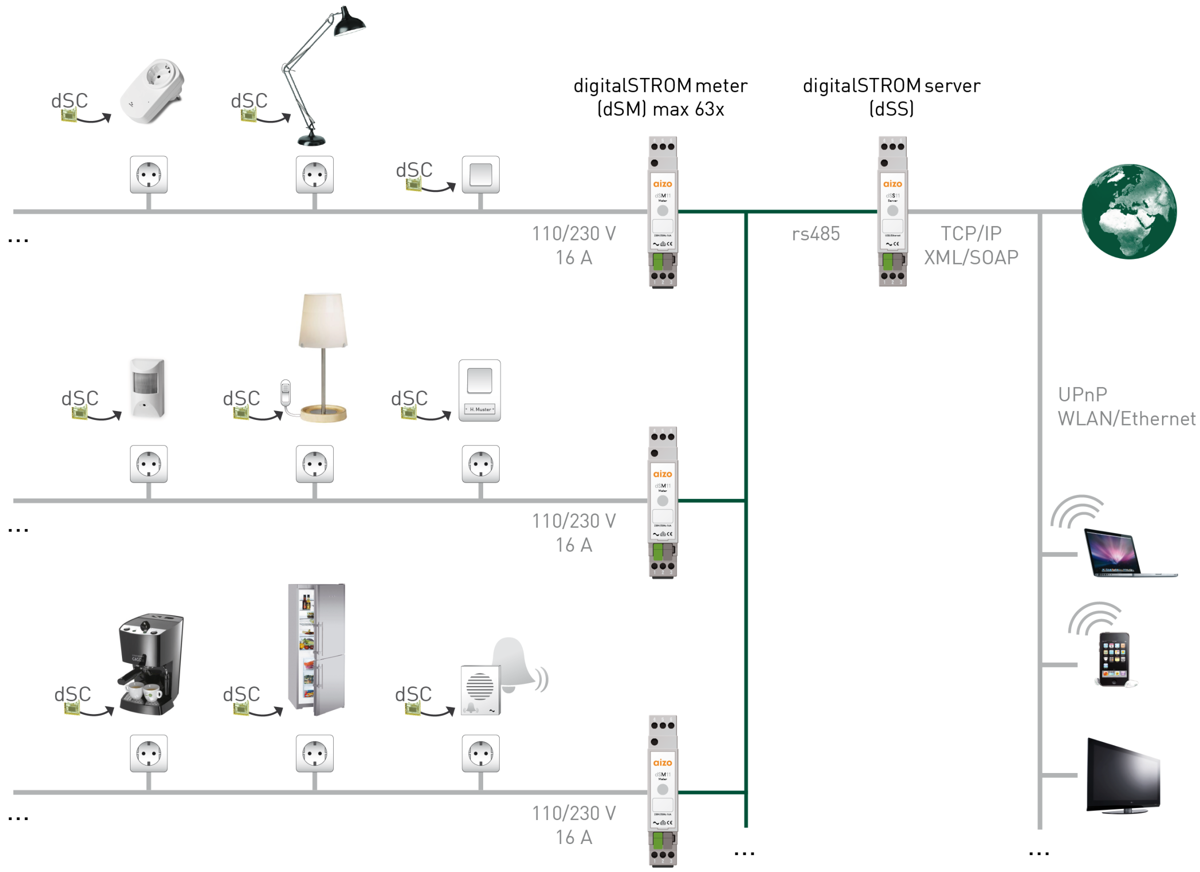


Figure 1: digitalSTROM Topology

3 Event Objects

The digitalSTROM concept is based on a hierarchical architecture. Three context levels communicate bidirectional using event objects on different abstraction levels. The *Device* level is represented by digitalSTROM Devices, the *System* layer has a logical context in the zone and is physically represented by a digitalSTROM Meter, and the *User* level which is present on the digitalSTROM Server and its system customization through user definable actions and automatisms.

3.1 Device Level

On the device level digitalSTROM generates Low-Level-Events to signal and forward events to the system. Low-Level-Events are caused by sensor input data that is preprocessed through device specific decision engines and state machines. Device level events do not have a context, they are not bound to a zone or functional group.

Examples for Low-Level-Events are the events generated if a wall switch is pressed and a device on-off switch is toggled.

3.2 System Level

The system layer processes Low-Level-Events, adds the context information of the event, and generates System-Level-Events. The event context consists of the corresponding zone of the device where the event occurred, the target function and color group of the device and the current state of the functional group in the specific zone. System-Level-Events are sent downstream as system actions and are forwarded upstream. Low-Level-Events without relevance on the system layer may bypass the system level processing and will only be forwarded to upper layers.

For example the commands to turn the lights in a room on and to open the blinds are System-Level-Events.

3.3 High Level

The highest layer receives System-Level-Events and provides them and additionally the system state for further processing and for external applications. The System-Level-Events are evaluated by user level applications and enables them to generate appropriate response actions taking user behavior and user defined parameters into account.

The user level descriptions *Turn light on in living room* and *Breakfast* are examples for High-Level-Events. With digitalSTROM Server modules the user is able to define his own personalized event descriptions and thus to create a customized system behavior.

For a few High-Level-Events it is very likely that any user has the same understanding of the event meaning. Those High-Level-Events with high standardization potential are mapped directly to System-Level-Events, for example the door bell signal.

3.4 Information Flow

The following graph illustrates the information flow between the three layers.

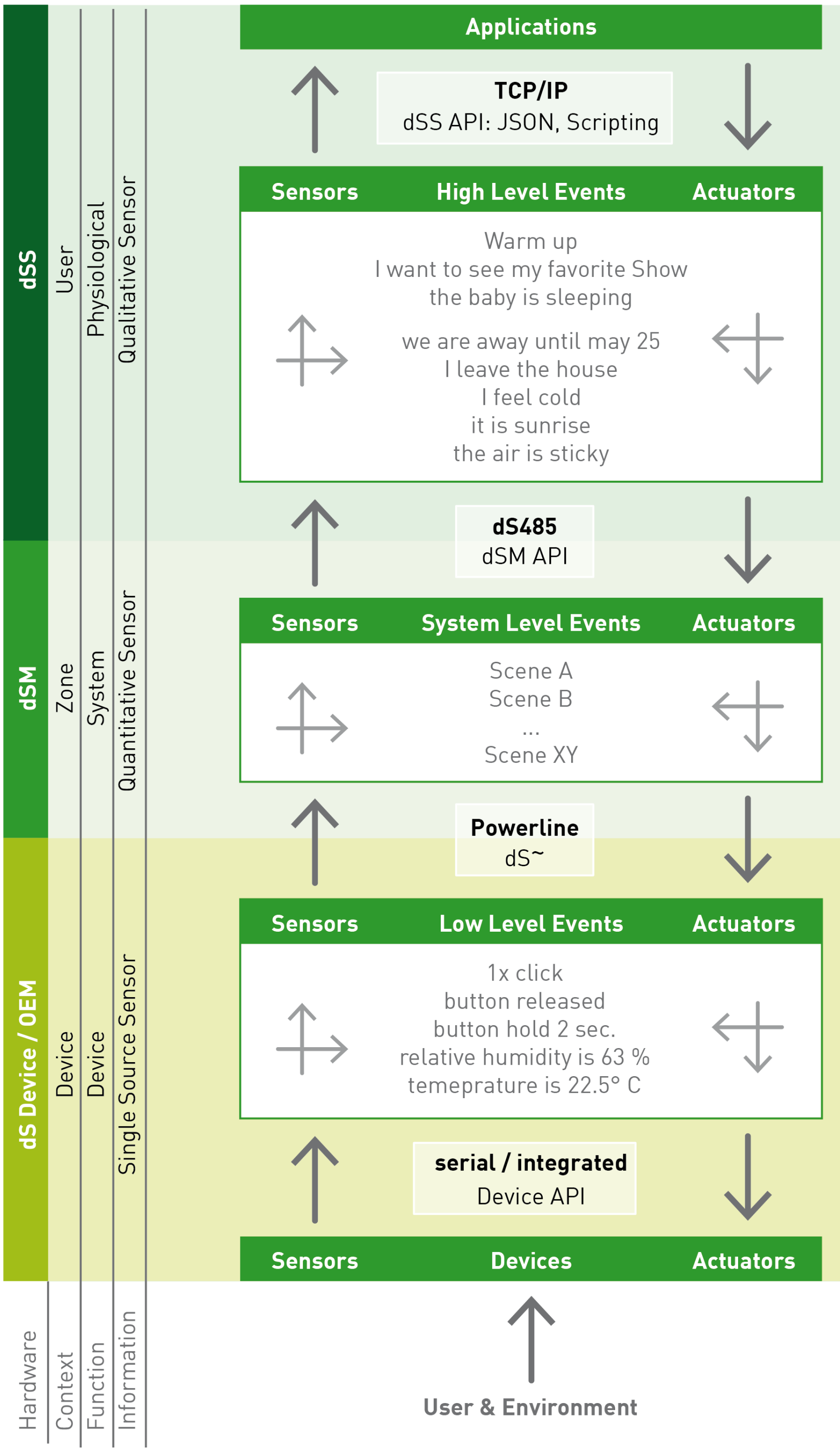


Figure 2: digitalSTROM Information Flow

3.5 Operating Concept

The functional groups and the zone are the central objects of the digitalSTROM operating concept (6). Calling presets for groups in a zone is the basic method to control digitalSTROM devices. Therefore this standard defines the default preset values for devices in each group to guarantee the well defined system behavior. Those default presets are subject of the particular functional group documents.

From users point of view the room wall switches are the direct interface to the operating concept. For example a traditional light switch is expected to turn the lights in a room on and off. A digitalSTROM wall switch behaves exactly like that, if the underlaying digitalSTROM pusbbutton device and the corresponding digitalSTROM light actuators belong to the light group. This Plug'n'Play mechanism uses an autodetection of device configuration and capabilities (F).

3.6 Distributed Intelligence

A central aspect of the digitalSTROM topology is the distributed intelligence. On device layer any digitalSTROM Device has the knowledge how to react to System-Level-Events (see 4), and on the system layer the logic is split into the electrical circuits.

Due to the distributed logic there is no need for a central communication management and the configuration complexity is reduced. Furthermore the overall system reliability is inherently increased.

4 Scenes, Activities and Presets

A scene is the command that is sent to a group of devices and results in changes of output actuator states. Each digitalSTROM Device keeps a table of preset values locally and knows how to react to scene commands.

Every digitalSTROM Ready Device has a well defined default scene configuration depending on its functional group. These standardized values ensure system compatibility and guarantee that future products and devices will behave harmonically in the existing infrastructure.

The actuator change depends on the individual device configuration. Though there are the preset values defined, this configuration can be modified by the user to adopt device behavior to his needs. This personalization is the basis for creating different preconfigured scenarios which can be activated by a single scene command.

Scenes directly correspond to System-Level-Events. An activity is the process of posting a scene command, in the API documentation also referred as *call scene*.

On device level scene commands may have different effects:

- setting an absolute output value for each output channel
- relative output value changes
- stopping an output value change process, applies to actions that take longer time to complete
- ignoring the command and not to react on a specific scene
- temporary changing the output state for signalling or identification purposes

Details of the scene configuration are explained in appendix Device Scene Table.

The digitalSTROM system implements 128 scene commands. The first half of 64 scenes is used for group specific actions, the second half are used in the context of a zone or for system wide functions.

The group related scenes may have different meanings depending on the functionality of the group. For example the activities "Light On" and "Open Blinds" share the same scene command.

Rule 4 digitalSTROM Devices have to implement a default behavior for all 128 scene commands. The system behavior and default values are defined in the particular documents for each functional group.

4.1 Function and group related scenes

The first 64 scene commands are related to the functional group. The intended behavior and the resulting preset is group dependent. The command is transferred using a multicast addressing scheme and is executed by every device that is part of the target group (addressing detailed in).

On the system layer the zone and group state machines track the activities and keep the status of the currently active scene. Thereby the system layer is able to implement state dependent behavior 6.

Rule 5 When applications send a scene command to a set of digitalSTROM Devices with more than one target device they have to use scene calls directed to a group, splitting into multiple calls to single devices has to be avoided due to latency and statemachine consistency issues.

Section B defines the details of the terms and values.

4.1.1 Basic Presets

digitalSTROM defines 5 sets of basic presets per functional group. Each of these sets has different scene commands and separate configuration values in the digitalSTROM Devices.

For example the presets for room lights are defined as follows:

Activity	Description
Preset 0	Lights off
Preset 1	Lights on
Preset 2	Lights on, Brightness 75%
Preset 3	Lights on, Brightness 50%
Preset 4	Lights on, Brightness 25%
Auto Off	Lights fading slowly off

Table 2: Light Preset 0...4

4.1.2 Stepping

digitalSTROM defines scene commands for stepping or dimming operations. Those commands increment or decrement the output value of actuators by a defined delta value.

Rule 6 digitalSTROM Ready Devices must ignore stepping commands if their output value is zero.

4.1.3 Area scenes

digitalSTROM defines 4 subsets of a group in a zone, called areas. The system implements distinct area commands for "On/Off" and stepping operations.

4.1.4 Local push button

digitalSTROM defines special scenes commands to reflect the usage with a device switch operated locally. These operations include "Local On", "Local Off" and "Local Stop".

4.1.5 Special scenes

digitalSTROM defines special scene commands for forcing the device to set the output value to maximum and minimum value. An additional scene command forces an output value change process to be stopped immediately.

4.1.6 Temperature control scenes

The room temperature control group uses its own set of system-level-events to control the system behavior of the control algorithm. Each control scene is related to a predefined behavior and is associated with setpoint values for the temperature.

4.2 Group independent scenes

The second half of 64 scene commands have a group independent meaning. Some of them are standardised High-Level-Events (e.g. Wakeup or Sleeping). Others are used for system wide functions (e.g. Absent, Door Bell), which are either a signal or an apartment state.

Section B.7 defines the details of the terms and values.

4.2.1 Deep Off

The activity *Deep Off* indicates that a zone is inactive and the user won't need the digitalSTROM Devices in this single zone for a longer time. digitalSTROM Ready Devices shall turn their power off. Devices with an integrated LED should indicate this state and turn it off.

4.2.2 Standby

The activity *Standby* indicates that a zone is inactive but the user might return in a short term. digitalSTROM Ready Devices shall switch to a standby mode.

4.2.3 Zone Active

The activity *Zone Active* indicates that a zone will become active in a short term and devices should be prepared to be powered on.

4.2.4 Auto Standby

The activity *Auto Standby* indicates that a zone is inactive. This signal is mainly provided to be generated automatically.

4.2.5 Absent

The activity *Absent* indicates that the residents left their home. Devices which do not have to be necessarily on should turn their power off. This is an apartment state and will be reset by the *Present* scene command.

4.2.6 Present

The activity *Present* indicates that the residents came back to their home. Devices that the user expects to be immediately availabe should turn their power on or go into standby mode.

4.2.7 Sleeping

The activity *Sleeping* indicates that the apartment switches to night operation and the residents go to sleep.

4.2.8 Wakeup

The activity *Wakeup* indicates that the apartment switches to daytime and the residents wake up. Devices which are likely to be used in the morning should be prepared to be powered on.

4.2.9 Door Bell

The activity *Door Bell* indicates that someone is standing in front of the house door and wants to get in. This is a signal and does not change any group or zone states.

4.2.10 Panic

The activity *Panic* indicates that the user is scared and believes there is someone in the apartment who should not be there. digitalSTROM Ready Devices shall switch to a protection state. This is an apartment state and will be reset by the associated undo scene command (see 5.1.2).

4.2.11 Fire

The activity *Fire* indicates an active fire alarm and is meant as an additional alarm signal for the residents. This is an apartment state and will be reset by the associated undo scene command (see 5.1.2).

4.2.12 Alarm

The activities *Alarms [1..4]* indicate an active alarm and is meant as an user defineable alarm signal for the residents. This is an apartment state and will be reset by the associated undo scene command (see 5.1.2).

4.2.13 Wind

The activity *Wind* is a signal to protect equipment and hardware of being damaged by high wind force or gust. This is an apartment state but can be used in a cluster as well. The state is reset by the associated *No Wind* activity.

4.2.14 Rain

The activity *Rain* is a signal to protect equipment and hardware of being damaged by rain and water. This is an apartment state. The state is reset by the associated *No Rain* activity.

4.2.15 Hail

The activity *Hail* is a signal to protect equipment and hardware of being damaged by a hail storm. This is an apartment state but can be used in a cluster as well. The state is reset by the associated *No Hail* activity.

4.3 Local Priority

The local priority is a special device state where usually scene commands with group relation are ignored. This state is triggered by local push button presses on the device itself or by a particular device action command which is used by the zone state machine for area scene commands (see 5.1.4).

The intended behavior for the local priority is to allow the device to keep its locally requested state even if the zone/group state is changed by the user.

This state is reset by turning the device locally off or by specific scene commands that force to change the output state regardless of the local priority. For example the system wide function *Absent* shall have an effect on room lights regardless if they were turned on locally.

5 Actions

A digitalSTROM Device must understand the actions detailed in this section.

5.1 Scenes

The scene actions can be sent to:

- a group of devices in a zone
- all devices in a zone
- all devices of a group in any zone
- a single device

5.1.1 Calling a Scene

The digitalSTROM Device executes the given scene command and changes the output state according to its scene table value and configuration flags. The scene and state is valid until another scene command is executed.

If the device is in the local priority state the scene is only executed if the *IgnoreLocalPrio* flag is set in the scene table for the called scene. The execution of this command can optionally be forced and regardless of current local priority state.

5.1.2 Undo a Scene

The digitalSTROM Device reverts all effects of the currently executed scene and executes the previously active scene. Only one previous scene command has to be stored. If no previous scene command is available this call shall be ignored.

The action can be constrained to only revert a specific scene command.

5.1.3 Store Current Value

Persistently store the current value of the digitalSTROM Device's output in the scene table at the given scene command.

5.1.4 Activate Local Priority

If the digitalSTROM Device is currently executing the given scene, activate the local priority state. This action remotely sets the local priority state for a group of devices.

5.1.5 Set to Lowest Stepping Step

Prepare the digitalSTROM Device to start stepping from the off state. If the scene table contains a non zero value at the given scene number execute the scene according to the scene table but set the output value to the minimal stepping value instead of the value in the scene table.

This action is necessary for stepping operations when the state of the addressed group is "Off". Compare to the rule for scene commands Stepping, where stepping commands are to be ignored for devices in the *Off* state.

5.2 Programming Mode

The programming mode actions are used internally by the digitalSTROM Meter to control the wall switch scene programming operation.

5.2.1 Programming Mode Start

This action indicates the start of a programming operation. Operate the digitalSTROM Device's output in order to identify the device in the installation. The operation must complete within 4 seconds and the device must return to the previous state afterwards. Also prepare the device for scene table programming.

Rule 7 digitalSTROM Device have to complete the identification action on the command *Programming Mode Start* within 4 seconds.

5.2.2 Programming Mode Finish

This command indicates the end of a programming operation. Operate the digitalSTROM Device's output in order to identify the device in the installation. The operation must complete within 4 seconds and the device must return to the previous state afterwards.

5.3 Configuration

5.3.1 Read and Write Device Parameters

For configuration purposes a digitalSTROM Device must support the reading and writing of device parameters. The device parameters are detailed in appendix sections C and D.

Rule 8 Application processes that do automatic cyclic reads or writes of device parameters are subject to a request limit: at maximum one request per minute and circuit is allowed.

5.4 Sensor Values

5.4.1 Read Device Sensor Values

digitalSTROM Devices with sensor equipment provide a table with entries for each available sensor. The table entries and the corresponding measurements are detailed in C.2.8 and E.

Rule 9 Application processes that do automatic cyclic reads of measured values are subject to a request limit: at maximum one request per minute and circuit is allowed.

5.4.2 Distribute Sensor Value

The command distributes a single numerical measured value. The measurement can be sent to:

- a group of devices in a zone
- all devices in a zone
- all devices of a group in any zone

digitalSTROM Devices may change their output state depending on their type and configuration.

5.5 Miscellaneous

5.5.1 Device Identification (a.k.a. "Blink")

Operate the digitalSTROM Device output in order to identify the device in the installation. For example for a light device: on-off-on-off or a shade device: up-down-up-down.

5.5.2 Directly Set Output Value

Unconditionally set the digitalSTROM Device's output value. The current local priority state is ignored.

Rule 10 The action command "Set Output Value" must not be used for other than device configuration purposes.

6 Function and group state machines

The Low-Level-Events generated by devices are received and interpreted by the digitalSTROM Meter. The digitalSTROM Meter translates these events (7.2) into actions (5). This translation process takes several parameters into account, like type and group membership of the originating device.

The translation is done according to function specific state machines (SM). These state machines define the user visible behavior of a digitalSTROM system. The SMs each have a state per context appropriate to the function (e.g. per function and zone, per function in an installation, per installation). The state machines are also responsible for the handling of STOP actions in functions where it is required.

For a detailed description of the behavior refer to the function specific state machine documents.

7 Communication Objects

7.1 Device Level Physical Layer

The transfer direction of events from device to system level is called upstream, the system to device event transfer direction is the downstream direction.

7.1.1 Downstream

Downstream data is sent by the digitalSTROM Meter and received by any digitalSTROM Device on the same power circuit. The downstream commands are received by all devices but only processed by previously selected devices and ignored by all others.

7.1.2 Upstream

Upstream data is sent by the digitalSTROM Device and received by the digitalSTROM Meter on the same power circuit. Other digitalSTROM Devices on the same power circuit do not receive the upstream event data.

7.1.3 Automatic Repeat Query (ARQ)

In case of parallel transmissions of upstream data by two or more digitalSTROM Devices the event data may not be received by the digitalSTROM Meter. To guarantee delivery of event data the system implements an automatic repeat query mechanism. An automatic retransmission of the event data takes place if the upstream reception is not acknowledged within a defined timeout period.

Generally there is an explicit acknowledge command necessary, but several actions can make use of an implicit acknowledge by reception of downstream scene commands.

7.1.4 Latency

Event transfer over the power line takes noticeable time, which increases in case of retransmissions. The overall latency of typical actions is shown in table 3.

Description	Latency in ms
Pushbutton tip to resulting system level event reception by devices	600
Query of device parameter to reception of value	1200
Downstream transmission of zone/group scene command to reception by devices	250

Table 3: Latency of typical operations

7.1.5 Addressing

The downstream communication implements three different levels of addressing.

Broadcast Broadcast addressing is defined both for zones and groups. A zone number of zero indicates that a downstream command is directed to all devices regardless of their zone membership. A group number of zero indicates group broadcast addressing and the downstream command is directed to all devices regardless of their group membership.

Group and Cluster Sending commands to a specific group or cluster within a zone is known as multicast addressing. This method is preferred for scene commands.

Device Unicast addressing to a single device is the slowest transfer mechanism. This method is mainly used for configuration purposes.

7.2 Device Level Events

Events on device level are generally created due to any sensor input data being available. The sensor data is then evaluated and preprocessed by device specific decision engines and state machines.

If the pushbutton configuration is *local device mode* the device decides about local actions it has to take. If the device is about to change its output state it has to send this information upstream.

If the pushbutton configuration is *system mode* the event information has to be sent upstream. System mode includes zone pushbuttons, area pushbuttons, application mode, any mode which is not exclusive processed locally in the device.

7.2.1 Pushbutton

Pushbutton event data include a button index and the type of key press. Details of this event are defined in section 8.

Pushbutton events are used as direct input to the system state machines (6).

7.2.2 Binary Input Event

Binary input events are generated by level triggered input lines. They have the binary states *on* or *off*.

Binary input events are used as direct input to the system state machines (18).

7.2.3 Status Event

digitalSTROM Devices can generate events depending on status changes of internal applications. This includes malfunctions, error conditions and device specific notifications.

7.2.4 Sensor Event Table

digitalSTROM Devices may implement a generic preprocessing algorithm which allows generation of events depending on thresholds of sensor input data. The thresholds and limits for the preprocessing algorithm are stored in well defined table format in the digitalSTROM Device. Details of the event table format are defined in the appendix Class 6 - Sensor Event Table.

The event data includes the index of the triggering table row. This index is unique only per device. These table index events are device specific and therefore not system relevant and bypass the system state machines. They are signalled on the user level.

7.2.5 Limits

digitalSTROM Ready Devices must ensure that they do not monopolize the upstream data channel. They must have a transmission control algorithm that controls the upstream event data rate.

Rule 11 digitalSTROM Ready Devices must not send upstream events contինously and must stop sending Low-Level-Event data even if the event is still or repeatedly valid. Transmission of pushbutton events must be abondoned after a maximum time of 2.5 minutes. Automatically genereated events must not exceed a rate limit of 10 events per 5 minutes.

7.3 System level communication

digitalSTROM uses a central communication controller digitalSTROM Meter (dSM) on each power circuit placed directly behind the circuit breaker. The dSM's are connected among themselves and with the digitalSTROM Server over a RS485 line and communicate using the dS485 protocol.

7.3.1 dS485 Protocol

The dS485 protocol is a data link layer communication protocol running over a shared medium. The protocol implements token passing medium access control and thus guarantees a fair scheduling of transmission requests.

7.3.2 dSM-API

The dSM-API is the application layer protocol to interface with the digitalSTROM components. From external application side the dSM-API is an internal interface, enclosed in the digitalSTROM Server and should not be used directly.

7.3.3 Webservices

The digitalSTROM Server provides webservice interfaces for external applications. This interface includes a more abstract representation of the digitalSTROM components and a convenient data model.

Rule 12 Applications shall use the digitalSTROM Server webservice interface for communication with the digitalSTROM system. Directly interfacing the dSM-API shall be avoided because it is an internal interface and its API may change in the future.

7.3.4 Limits

Applications must ensure that they do not monopolize the downstream data channel. Therefore they have to take that scene commands are not executed at a arbitrary rate.

Rule 13 Applications that automatically generate *Call Scene* action commands (see 5.1.1) must not execute the action commands at a rate faster than one request per second.

8 Pushbuttons

The term pushbutton refers to the input line of a digitalSTROM terminal block for wall switches or the on-off switch of electrical devices. digitalSTROM requires to use pushbuttons instead of switches to be able to enhance the simple on-off functionality. digitalSTROM Devices are able to detect in which manner the user activated the button, for example a single tip, a hold operation, or fast sequence of consecutive tips.

8.1 Input Processing

8.1.1 Timing

Depending on the duration and pulse duty of pushbutton tips different events are generated. The following table defines the timing requirements of the pushbutton evaluation. A "H" (High) means the button input is Active, a "L" (Low) means the input is inactive.

Pushbutton Event	**Timing**
Single Tip	140ms <= H < 500ms
Double Tip	140ms <= H < 500ms, L < 800ms, 140ms <= H < 500ms
Triple Tip	140ms <= H < 500ms, L < 800ms, 140ms <= H < 500ms, L < 800ms, 140ms <= H < 500ms
Quadruple Tip	140ms <= H < 500ms, L < 800ms, 140ms <= H < 500ms, L < 800ms, 140ms <= H < 500ms, L < 800ms, 140ms <= H < 500ms [a]
Hold Start	H >= 500ms [b]
Single Click	H < 140ms
Double Click	H < 140ms, L < 140ms, H < 140ms
Triple Click	H < 140ms, L < 140ms, H < 140ms, L < 140ms, H < 140ms
Short-Long	H < 140ms, L < 140ms, H>= 500ms
Short-Short-Long	H < 140ms, L < 140ms, H < 140ms, L < 140ms, H >= 2500ms

[a] Subsequent tips start again at generating Double-Tip signals and wrap around each 3 tips: Single Tip - Double Tip - Triple Tip - Quadruple Tip - Double Tip - Triple Tip - ...

[b] After initial "Hold Start" detection each second a "Hold Repeat" signal is generated. After the button input is released a "Hold End" signal is generated.

Table 4: Pushbutton timing

8.2 Low-Level-Events

A pushbutton operates either in device or in zone mode, and it may have one (1-way) or two input lines (2-way). Depending on this configuration the pushbutton has a different system behavior. In device mode some operations are executed only locally.

The pushbutton input events are encoded using a *Click-Type* and a *Key-Number*. The tupel of Click-Type and Key-Number identifies the kind of Low-Level-Event. Where Click-Type encodes the manner the button has been activated, the Key-Number is an index for the operating mode of the pushbutton depending on 1-way, 2-way and device or zone mode.

The following sections describe the different Low-Level-Events generated by pushbutton clicks in dependence of the operation mode. The Low-Level-Event names listed in the tables below are used as input signal names in the state machine documentation of the particular functional groups.

> **Notice** The short-short-long sequence is always reserved for device internal use and configuration purposes.

8.2.1 1-way device pushbutton

In device pushbutton mode the *Single Tip* events are processed locally and toggle the output state of the device. If the device output is in a slow change operation a *Single Tip* stops the change process.

The *Hold Start* event is processed locally if the device output is active and step the output value alternately down and up. If the output is inactive the button hold operation sequence is signaled upstream with an initial *Hold Start*, consecutive *Hold Repeat* and a final *Hold End* event.

The short-long sequence is processed locally and enables local configuration of the default scene values for Preset1 to Preset4.

If digitalSTROM Devices do not have an output the local pushbutton mode is not supported.

Pushbutton Event	Click type	Key Number	Low Level Event Name
Single Tip	11	4	LOCAL_ON
Hold Start	4	4	HOLD_START
Hold Repeat	5	4	HOLD_REPEAT
Hold End	6	4	HOLD_END

Table 5: Output off or inactive

Pushbutton Event	Click type	Key Number	Low Level Event Name
Single Tip	10	4	LOCAL_OFF

Table 6: Output on or active

Pushbutton Event	Click type	Key Number	Low Level Event Name
Single Tip	14	4	LOCAL_STOP

Table 7: Output is currently changing

Pushbutton Event	Click type	Key Number	Low Level Event Name
Double Tip	1	4	TIP_2X
Triple Tip	2	4	TIP_3X
Quadruple Tip	3	4	TIP_4X
Single Click	7	4	CLICK_1X
Double Click	8	4	CLICK_2X
Triple Click	9	4	CLICK_3X

Table 8: Output in any state

8.2.2 1-way zone or area button

There is no local processing of input events in zone or area mode. All events are forwarded upstream regardless of the device output state.

Pushbutton Event	Click type	Key Number	Low Level Event Name
Single Tip	0	0	TIP_1X
Double Tip	1	0	TIP_2X
Triple Tip	2	0	TIP_3X
Quadruple Tip	3	0	TIP_4X
Single Click	7	0	CLICK_1X
Double Click	8	0	CLICK_2X
Triple Click	9	0	CLICK_3X
Hold Start	4	0	HOLD_START
Hold Repeat	5	0	HOLD_REPEAT
Hold End	6	0	HOLD_END

Table 9: Zone pushbutton Low-Level-Events

8.2.3 2-way device pushbutton

Pushbuttons with two input lines have designated *Up* and *Down* functions. Input on these buttons generate different Low-Level-Events than the 1-way button.

In device pushbutton mode the *Single Tip* events on both inputs are processed locally, tips on the *Up* input turns the device output on, tips on the *Down* input turn it off.

The *Hold Start* event on both input lines is processed locally and step the output value accordingly up and down. Holding the *Up* button if the output is inactive turns the output on, if the output is capable of stepping with the lowest possible output value otherwise with the maximum output value. Holding the *Down* button if the output is active steps the output value down:

- if the device output is capable of stepping the output value is reduced down to the minimum value but not to the off state,
- switched mode outputs do ignore *Hold Start* events.

If the output is inactive the button hold operation sequence on the *Down* input is signaled upstream with an initial *Hold Start*, consecutive *Hold Repeat* and a final *Hold End* event.

The short-long sequence is processed locally and enables local configuration of the default scene values for Preset 0...4 to Preset 40...44.

If digitalSTROM Devices do not have an output the local pushbutton mode is not supported.

Pushbutton Event	Click type	Key Number	Low Level Event Name
Single Tip	11	5	LOCAL_OFF
Hold Start	4	5	HOLD_START_DOWN
Hold Repeat	5	5	HOLD_REPEAT_DOWN
Hold End	6	5	HOLD_END_DOWN

Table 10: Down input, output off or inactive

Pushbutton Event	Click type	Key Number	Low Level Event Name
Single Tip	11	5	LOCAL_OFF

Table 11: Down input, output on or active

Pushbutton Event	Click type	Key Number	Low Level Event Name
Double Tip	1	5	TIP_2X_DOWN
Triple Tip	2	5	TIP_3X_DOWN
Quadruple Tip	3	5	TIP_4X_DOWN
Single Click	7	5	CLICK_1X_DOWN
Double Click	8	5	CLICK_2X_DOWN
Triple Click	9	5	CLICK_3X_DOWN

Table 12: Down input, output in any state

Pushbutton Event	Click type	Key Number	Low Level Event Name
Single Tip	12	6	LOCAL_ON
Double Tip	1	6	TIP_2X_UP
Triple Tip	2	6	TIP_3X_UP
Quadruple Tip	3	6	TIP_4X_UP
Single Click	7	6	CLICK_1X_UP
Double Click	8	6	CLICK_2X_UP
Triple Click	9	6	CLICK_3X_UP

Table 13: Up input, output in any state

8.2.4 2-way zone or area button

There is no local processing of input events in zone or area mode. All events are forwarded upstream regardless of the device output state.

Pushbutton Event	Click type	Key Number	Low Level Event Name
Single Tip	0	1	TIP_1X_DOWN
Double Tip	1	1	TIP_2X_DOWN
Triple Tip	2	1	TIP_3X_DOWN
Quadruple Tip	3	1	TIP_4X_DOWN
Single Click	7	1	CLICK_1X_DOWN
Double Click	8	1	CLICK_2X_DOWN
Triple Click	9	1	CLICK_3X_DOWN
Hold Start	4	1	HOLD_START_DOWN
Hold Repeat	5	1	HOLD_REPEAT_DOWN
Hold End	6	1	HOLD_END_DOWN

Table 14: Down input, 2-way zone pushbutton

Pushbutton Event	Click type	Key Number	Low Level Event Name
Single Tip	0	2	TIP_1X_UP
Double Tip	1	2	TIP_2X_UP
Triple Tip	2	2	TIP_3X_UP
Quadruple Tip	3	2	TIP_4X_UP
Single Click	7	2	CLICK_1X_UP
Double Click	8	2	CLICK_2X_UP
Triple Click	9	2	CLICK_3X_UP
Hold Start	4	2	HOLD_START_UP
Hold Repeat	5	2	HOLD_REPEAT_UP
Hold End	6	2	HOLD_END_UP

Table 15: Up input, 2-way zone pushbutton

8.3 System Interfaces

8.3.1 Settings

digitalSTROM Devices may have one pushbutton which can operate in different modes as described above. Beside the one system pushbutton there may be additional buttons with local functionality only.

> **Notice** digitalSTROM supports only one pushbutton per digitalSTROM Device with system functionality.

Although used by the upper layer state machines the pushbutton operating mode parameters are stored locally on the device. That way the system can retrieve the mode parameters from the device and synchronize the settings.

8.3.2 Event Limits

digitalSTROM Devices have to take care about pushbutton or input malfunctions and may not generate any number of events, see 7.2.5.

8.3.3 Configuration

The pushbutton parameters that are relevant for the system behavior and the state machine operations are stored on the digitalSTROM Device itself. This ensures that a device operates in the same mode regardless of the circuit and zone where it is plugged in.

Operating Mode This 4-Bit value *ButtonId* is stored in the device register LTNUMGRP0 (see C.3.2).

Target Group This 4-Bit value *ButtonGroup* is stored in the device register LTNUMGRP0 (see C.3.2).

Button Input Mode This 8-Bit value *ButtonInputMode* is stored in the device register LTMODE (see C.3.6).

9 Binary Inputs

The term *binary input* refers to an input line of a digitalSTROM Device which is connected to an corresponding binary status output signal of an electrical device. This input type is used for automation purposes and provides an interface to different kinds of status outputs.

For example smoke detectors, wind monitors or motion detectors provide a binary status signal.

9.1 Low-Level-Events

Each transition on the input line is signaled with an appropriate upstream event. Additionally, the state of a binary input line must be available at any time for synchronization, e.g. on system startup.

The binary input events are encoded using a binary data type, the data may be either 0 (inactive) or 1 (active).

The status of an input line is always available as a status value on the digitalSTROM Server. The status change of an input line can be used as trigger for user defined activities.

digitalSTROM Devices have to take care about event rate limits input and may not generate an arbitrary number of events, see 7.2.5.

9.2 Input Types

Each binary input line of digitalSTROM Terminal Block has an associated type. This type is determined by the device registration process and later on evaluated for each binary input event received from this digitalSTROM Device.

Input Type	Assigned Index	Natural Device and Description
Presence	1	Presence detector
Brightness	2	
Presence in darkness	3	Presence detector with activated internal twilight sensor
Twilight	4	Twilight sensor
Motion	5	Motion detector
Motion in darkness	6	Motion detector with activated internal twilight sensor
Smoke	7	Smoke Detector
Wind strength above limit	8	Wind monitor with user-adjusted wind strength threshold
Rain	9	Rain monitor
Sun radiation	10	Sun light above threshold
Temperature below limit	11	Room thermostat with used-adjusted temperature threshold
Battery status is low	12	electric battery is running out of power
Window is open	13	Window contact
Door is open	14	Door contact
Window is tilted	15	Window handle; window is tilted instead of fully opened
Garage door is open	16	Garage door contact
Sun protection	17	Protect against too much sun light
Frost	18	Frost detector

Table 16: Binary input types

Addionally each input line has an assigned target group which specifies which target function will be triggered by input signals. Currently only the Joker function is supported.

Target Function	Assigned Index
Joker	8

Table 17: Binary input target functions

9.3 System Behavior

The digitalSTROM system automatically generates a system level events upon reception of binary events from Terminal Blocks. Depending on the current corresponding state, the input type and the target group an appropriate event is generated. Further system actions can be implemented by system add-ons.

State Change	System Activity
Smoke detector active	Call fire alarm in apartment
Wind monitor active	Call wind alarm in apartment or cluster
Wind monitor inactive	Call no-wind activity
Rain monitor active	Call rain alarm in apartment or cluster
Rain monitor inactive	Call no-rain activity
Presence active	Set presence state in corresponding zone
Presence inactive	Clear presence state in corresponding zone
Motion active	Set motion state in corresponding zone
Motion inactive	Clear motion state in corresponding zone
Room thermostat active	Call On activity in heating group of the corresponding zone
Room thermostat inactive	Call Off activity in heating group of the corresponding zone

Table 18: Binary input activities

A state is cleared if all previously active inputs switched to inactive again. Logically the input signals are OR'ed. For example the motion state in a zone is set to active when any motion detector in the same zone triggers, but the state will be cleared only if all motion detectors in the same zone are inactive.

9.4 Input Processing

The input signal of a binary input line may be preprocessed by the digitalSTROM Device. digitalSTROM Terminal Blocks use an 8-Bit value *BinaryInputMode* that is stored in the device register LTMODE (see **??**).

10 Glossary

10.1 digitalSTROM Device

A digitalSTROM Device is an device with well defined system behavior and with digitalSTROM communication capabilities. The digitalSTROM Device is from the users point of view the complete piece of equipment, including all visible output, sensor and input functions. This definition includes that a digitalSTROM Device may be build out of multiple Terminal Blocks.

10.2 digitalSTROM Ready Device

A digitalSTROM Ready Device is a device which complies to all rules of the digitalSTROM concept.

10.3 dSUID - dS Unique Identifier

The dSUID is a 136 bit unique id uses a format that is based on Universal Unique Identifiers (RFC 4122). The dSUID was introduced for such cases where a GTIN and/or serial number is unknown or not available for a component or product, or where multiple components are build upon the same basis Id.

The dSUID has to be generated according to the following prioritized rules. The preferred methods allow to regenerate the dSUID and thus always recreate the same key. For the last-choice Random-dSUID it is an important requirement that the dSUID does not change over the lifetime. Modules that generate such a random dSUID have to store it persistently.

1. SGTIN-96 is available for a device
   → Use this SGTIN-96
2. A GTIN and serial number of device
   → Combine GTIN and serial number to form a SGTIN-128 with Application Identifier 21: "(01)<GTIN>(21)<serial number>" and use the resulting string to generate a UUIDv5 in the GS1-128 name space
3. An existing UUID of a device is available
   → Use the existing UUID
4. Another unique ID of the device is available within the kind of the device
   → Generate name based UUIDv5 with unique ID in the relevant name space; pre-defined name spaces are existing, e.g. for EnOcean, IEEE MAC
5. Nothing from the above
   → Only a locally unique ID can be generated, generate a random UUIDv4

10.4 dSID - SGTIN-96

The dSID is a 96 bit unique identifier and follows the SGTIN-96 (Serialized Global Trade Item Number) format defined by the GS1 standard organization.

The SGTIN-96 number consist of 4 parts:
8 Bit Header: fixed value, 0x30
3 Bit Filter Value:
3 Bit Partition Value: indicates length of company prefix
40-20 Bit Company Prefix
4-24 Bit Item Reference
38 Bit Serial Number

10.5 SGTIN-128

The SGTIN-128 (Serialized Global Trade Item Number) format is defined by the GS1 standard organization. The SGTIN-128 is a two part data object that consists of two application identifier and the related data, represented by e.g. the text string "(01)<GTIN>(21)<serial number>".

10.6 GID-96

GID-96 is the legacy format of the dSID. The structure of the GID-96 Global Identifier standard was defined by EPCGlobal Inc.

To keep compatibility with existing systems the legacy format of a dSID is furthermore supported. The SGTIN and GID-96 have a unique header prefixes and therefore can be used in parallel without overlap. The legacy dSID number is automatically generated for digitalSTROM Devices with firmware revision prior to 3.5.5.

The GID-96 number consist of 4 parts:
8 Bit Header: fixed value, 0x35
28 Bit Manager Number: EPCGlobal uniquely assigned number, 0x04175FE
24 Bit Object Class: digitalSTROM defined system objects
36 Bit Serial Number: digitalSTROM product serial number

The following Object Class values are defined by digitalSTROM:

Value	Object Class
0	digitalSTROM Device
1	digitalSTROM Meter
2 .. 0xFEFFFF	Reserved
0xFF0000 .. 0xFFFFFF	Devices with Ethernet MAC Address

Table 19: dSID Object Class

Devices with Ethernet interface can be mapped into the aizo dSID address space. For example the digitalSTROM Server is uniquely identified by a generated dSID using its Ethernet MAC Address. For example the MAC Address 12:34:56:78:90:AB is translated to the dSID "3504175FEFF12340567890AB".

10.7 Function-ID

The Function-ID is a 16 bit device configuration value that describes basic capabilities of digitalSTROM devices. This value contains the standard group of a device, basic information about the functionality and the encoding of upstream data messages.

The standard group of a digitalSTROM Device corresponds to its default color group.

10.8 Vendor-ID

The Vendor-ID is 16 bit device configuration value that describes the vendor company of the digitalSTROM device. The following Vendor-ID's are defined:

Value	Vendor-ID
1	digitalStrom GmbH, Germany
2	digitalStrom AG, Switzerland
3	ONE Smart Control, Belgium

Table 20: Vendor ID

10.9 Product-ID

The Product-ID is a 16 bit device configuration value describing the product family and hardware platform.

The Vendor-ID, Product-ID and standard color group together are inputs for looking up the GTIN and Product Code.

The Product-Code is used by the digitalSTROM Server and Server User Interface (Configurator) to decode product capabilities.

10.10 GTIN

Global Trade Item Number (GTIN) is a format to uniquely identify trade items following the standards defined by the GS1 organization.

As digitalSTROM Devices do not have their full GTIN stored on the device, this number is generated by the digitalSTROM Server using lookup tables like **??**.

The digitalSTROM Meter has its GTIN stored in the configuration area of the device, and a digitalSTROM Server has a GTIN as well.

Product Code	GTIN	Standard Color	Product-ID	Vendor-ID
GE-KM200	4290046000010	1	0x00C8	1
GE-TKM210	4290046000027	1	0x04D2	1
GE-SDM200	4290046000034	1	0x08C8	1
GE-SDS200-CW	7640156790221	1	0x18C8	2
GE-SDS200-CS	7640156790238	1	0x18C9	2
GE-SDS220-CT	7640156790214	1	0x18DC	2
GE-TKM220	4290046000201	1	0x04DC	1
GE-TKM230	4290046000218	1	0x04E6	1
GE-KL200	4290046000195	1	0x0CC8	1
GN-KM200	4290046000041	7	0x00C8	1
GN-TKM200	4290046000065	7	0x04C8	1
GN-TKM210	4290046000058	7	0x04D2	1
GR-TKM200	4290046000638	2	0x04C8	1
GR-TKM210	4290046000645	2	0x04D2	1
GR-KL200	4290046000607	2	0x0CC8	1
GR-KL210	4290046000614	2	0x0CD2	1
GR-KL220	4290046000621	2	0x0CDC	1
RT-TKM200	4290046000072	6	0x04C8	1
RT-SDM200	4290046000089	6	0x08C8	1
SW-TKM200	4290046000904	8	0x04C8	1
SW-TKM210	4290046000911	8	0x04D2	1
SW-AKM200	7640156790405	8	0x20C8	2
SW-AKM210	7640156790412	8	0x20D2	2
SW-AKM220	7640156790429	8	0x20DC	2
SW-KL200	4290046000959	8	0x0CC8	1
SW-ZWS200-J	4290046000935	8	0x14C8	1
SW-ZWS200-F	4290046000942	8	0x14C9	1
SW-ZWS200-E+F	7640156790481	8	0x14CA	2
SW-SDS200-CW	7640156790221	8	0x18C8	2
SW-SDS200-CS	7640156790238	8	0x18C9	2
SW-SDS220-CT	7640156790214	8	0x18DC	2

Table 21: aizo Product Codes and GTIN's

11 Certification Rules

Rule 1 A digitalSTROM Ready Device has to be preconfigured in the right application group. This is essential to ensure that all electrical devices in one application group can be orchestrated together.

Rule 2 A digitalSTROM Ready Device must be configured for exactly one digitalSTROM application group. The assigned application group must be non-ambiguous and is part of the static device configuration.

Rule 3 The function of a devices output is the basis of its group membership. For devices without output function the target function of the switch button decides about the group membership.

Rule 4 digitalSTROM Devices have to implement a default behavior for all 128 scene commands. The system behavior and default values are defined in the particular documents for each functional group.

Rule 5 When applications send a scene command to a set of digitalSTROM Devices with more than one target device they have to use scene calls directed to a group, splitting into multiple calls to single devices has to be avoided due to latency and statemachine consistency issues.

Rule 6 digitalSTROM Ready Devices must ignore stepping commands if their output value is zero.

Rule 7 digitalSTROM Device have to complete the identification action on the command *Programming Mode Start* within 4 seconds.

Rule 8 Application processes that do automatic cyclic reads or writes of device parameters are subject to a request limit: at maximum one request per minute and circuit is allowed.

Rule 9 Application processes that do automatic cyclic reads of measured values are subject to a request limit: at maximum one request per minute and circuit is allowed.

Rule 10 The action command "Set Output Value" must not be used for other than device configuration purposes.

Rule 11 digitalSTROM Ready Devices must not send upstream events continously and must stop sending Low-Level-Event data even if the event is still or repeatedly valid. Transmission of pushbutton events must be abondoned after a maximum time of 2.5 minutes. Automatically genereated events must not exceed a rate limit of 10 events per 5 minutes.

Rule 12 Applications shall use the digitalSTROM Server webservice interface for communication with the digitalSTROM system. Directly interfacing the dSM-API shall be avoided because it is an internal interface and its API may change in the future.

Rule 13 Applications that automatically generate *Call Scene* action commands (see 5.1.1) must not execute the action commands at a rate faster than one request per second.

A Structure Reference

A.1 Circuit

The maximum supported cable length on one circuit is limited to 50 meters. The maximum number of digitalSTROM components connected on the RS485 line is 63.

A.2 Zone

The maximum number of zones per circuit is limited to 15.

A.3 Group

The maximum number of groups per zone is limited to 63. The first 16 groups have a special meaning and cover all functional classes. The remaining 47 groups are reserved for user defined configurations.

A.4 Device

The maximum number of devices per circuit is limited to 128.

B Scene Command Reference

Scene command indices not mentioned here are reserved and must not be used.

B.1 Presets

Activity	Scene Command	Description
Preset 0	0	Set output value to Preset 0 (Default: Off)
Preset 1	5	Set output value to Preset 1 (Default: On)
Preset 2	17	Set output value to Preset 2
Preset 3	18	Set output value to Preset 3
Preset 4	19	Set output value to Preset 4

Table 22: Preset 0...4

Activity	Scene Command	Description
Preset 10	32	Set output value to Preset 10 (Default: Off)
Preset 11	33	Set output value to Preset 11 (Default: On)
Preset 12	20	Set output value to Preset 12
Preset 13	21	Set output value to Preset 13
Preset 14	22	Set output value to Preset 14

Table 23: Preset 10...14

Activity	Scene Command	Description
Preset 20	34	Set output value to Preset 20 (Default: Off)
Preset 21	35	Set output value to Preset 21 (Default: On)
Preset 22	23	Set output value to Preset 22
Preset 23	24	Set output value to Preset 23
Preset 24	25	Set output value to Preset 24

Table 24: Preset 20...24

Activity	Scene Command	Description
Preset 30	36	Set output value to Preset 30 (Default: Off)
Preset 31	37	Set output value to Preset 31 (Default: On)
Preset 32	26	Set output value to Preset 32
Preset 33	27	Set output value to Preset 33
Preset 34	28	Set output value to Preset 34

Table 25: Preset 30...34

Activity	Scene Command	Description
Preset 40	38	Set output value to Preset 40 (Default: Off)
Preset 41	39	Set output value to Preset 41 (Default: On)
Preset 42	29	Set output value to Preset 42
Preset 43	30	Set output value to Preset 43
Preset 44	31	Set output value to Preset 44

Table 26: Preset 40...44

B.2 Stepping

Activity	Scene Command	Description
Increment	11	Increment output value
Decrement	12	Decrement output value

Table 27: Stepping scenes

B.3 Area scenes

Activity	Scene Command	Description
Area 1 Off	1	Set output value to Preset Area 1 Off (Default: Off)
Area 1 On	6	Set output value to Preset Area 1 On (Default: On)
Area 1 Increment	43	Initial command to increment output value
Area 1 Decrement	42	Initial command to decrement output value
Area 1 Stop	52	Stop output value change at current position
Area Stepping Continue	10	Next step to increment or decrement

Table 28: Area 1

Activity	Scene Command	Description
Area 2 Off	2	Set output value to Area 2 Off (Default: Off)
Area 2 On	7	Set output value to Area 2 On (Default: On)
Area 2 Increment	45	Initial command to increment output value
Area 2 Decrement	44	Initial command to decrement output value
Area 2 Stop	53	Stop output value change at current position
Area Stepping Continue	10	Next step to increment or decrement

Table 29: Area 2

Activity	Scene Command	Description
Area 3 Off	3	Set output value to Area 3 Off (Default: Off)
Area 3 On	8	Set output value to Area 3 On (Default: On)
Area 3 Increment	47	Initial command to increment output value
Area 3 Decrement	46	Initial command to decrement output value
Area 3 Stop	54	Stop output value change at current position
Area Stepping Continue	10	Next step to increment or decrement

Table 30: Area 3

Activity	Scene Command	Description
Area 4 Off	4	Set output value to Area 4 Off (Default: Off)
Area 4 On	9	Set output value to Area 4 On (Default: On)
Area 4 Increment	49	Initial command to increment output value
Area 4 Decrement	48	Initial command to decrement output value
Area 4 Stop	55	Stop output value change at current position
Area Stepping Continue	10	Next step to increment or decrement

Table 31: Area 4

B.4 Local pushbutton

Activity	Scene Command	Description
DeviceOn	51	Local on
DeviceOff	50	Local off
DeviceStop	15	Stop output value change at current position

Table 32: Device scenes

B.5 Special scenes

Activity	Scene Command	Description
Minimum	13	Minimum output value
Maximum	14	Maximum output value
Stop	15	Stop output value change at current position
Auto-Off	40	Slowly fade down to off value

Table 33: Special scenes

B.6 Temperature control scenes

Activity	Scene Command	Description
Off	0	
Comfort	1	
Economy	2	
Not Used	3	
Night	4	
Holiday	5	

Table 34: Temperature control scenes

B.7 Group independent scenes

Activity	Scene Command
Deep Off	68
Standby	67
Zone Active	75
Auto Standby	64
Absent	72
Present	71
Sleeping	69
Wakeup	70
Door Bell	73
Panic	65
Fire	76
Alarm-1	74
Alarm-2	83
Alarm-3	84
Alarm-4	85
Wind	86
No-Wind	87
Rain	88
No-Rain	89
Hail	90
No-Hail	91

Table 35: Group independent activities and scene command values

C Device Parameters

digitalSTROM Devices have configuration parameters of different categories. The system relevant parameters described below are mandatory for digitalSTROM Ready Devices.

C.1 Class 0 - Communication Specific Parameters

C.1.1 Local Programming Mode

Offset 0x08, Length 8 Bit - PROGEN: Enable Local Programming Mode
This parameter controls the local programming mode functions of a digitalSTROM Device.

Value	Description
0	local programming mode disabled
1	only short-long allowed
2	only short-short-long allowed
3	both short-long and short-short-long allowed

C.2 Class 1 - digitalSTROM Device Specific Parameters

Parameters in this class are read-only and are not allowed to be modified by user operations.

C.2.1 Firmware Version

Offset 0x00, Length 16 bit - VER: Firmware Version
This is the firmware version of the digitalSTROM Device. 0x0321 reads as 3.2.1.

C.2.2 Serial Numer

Offset 0x02, Length 32 bit - DSID: digitalSTROM-ID
This parameter contains the serial number, the lower 32 bits of the unique 96-bit dSID **??**.

C.2.3 Function ID

Offset 0x06, Length 16 bit - FID: Function-ID
The Function ID defines the basic capabilities of the digitalSTROM Device.

C.2.4 Vendor ID

Offset 0x08, Length 16 bit - VID: Vendor-ID
Here the Vendor ID is defined.

C.2.5 Product ID

Offset 0x0a, Length 16 bit - PID: Product-ID
Here the Product ID is defined.

C.2.6 Structure and adressing

Offset 0x0c, Length 16 bit - ADDR: Short Address
The dynamic device short address assigned by the digitalSTROM Meter during device registration process F.

Offset 0x0e, Length 16 bit - VC: Virtual Circuit
The bit field indicating the zone membership of a device.

Offset 0x10, Length 64 bit - GRP: Group Mask
The bit field indicating the group membership of a device.

C.2.7 OEM/GTIN

Offset 0x1c, Length 16 bit - OEM_SERIAL: OEM Serial Number
Device manufacturers may store a serial number of their bundled Native Device in the OEM_SERIAL register.

Offset 0x1e, Length 8 bit - OEM_PARTNO: OEM Part Number
Device manufacturers may store a part number if a complex digitalSTROM Device is built of more than one Terminal Block.

C.2.8 Sensor Equipment

Offset 0x20, Length variable - ST: Sensor Table
A table containing an entry for each available sensor in the device.

Register C.1: Sensor Table Entry

Type	Reserved	Conversion	Context	Polling
15 – 8	7 – 5	4	3	2 – 0

Polling Default polling interval

Value	Description
0	disabled
1	60 seconds
2	5 minutes
3	30 minutes
4	1 hour
5	6 hours
6	12 hours
7	24 hours

Context Zone or global context

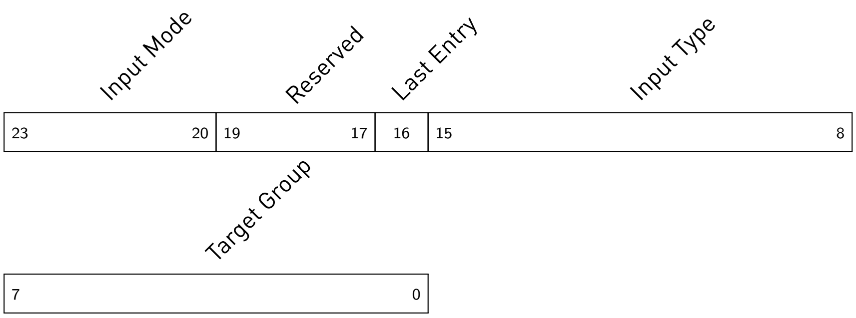
Conversion 12 to 10 bit conversion, either high or low bits important

Type Sensor Type (see E)

C.2.9 Binary Input Equipment

Offset 0x40, Length variable - BIT: Binary Input Table
A table containing an entry for each available binary input in the device.

Register C.2: Binary Input Table Entry



Target Group Id of the target functional

Input Type Binary input type, see table 16

Last Entry 0 = more entries to follow, 1 = this is the last input entry

Input Mode Input mode Id

C.3 Class 3 - Function Specific Parameters

C.3.1 Output Mode

Offset 0x00, 8 bit - MODE: Output Mode
This parameter describes, how the output is used, e.g. as dimmer or switch. This parameter is digitalSTROM Device and product specific, depending on the hardware equipment only selected values are supported by a device.

Output Mode	Description
0	No output or output disabled
16	Switched
17	RMS (root mean square) dimmer
18	RMS dimmer with characteristic curve
19	Phase control dimmer
20	Phase control dimmer with characteristic curve
21	Reverse phase control dimmer
22	Reverse phase control dimmer with characteristic curve
23	PWM (pulse width modulation)
24	PWM with characteristic curve
33	Positioning control
39	Relay with switched mode scene table configuration
40	Relay with wiped mode scene table configuration
41	Relay with saving mode scene table configuration
42	Positioning control for uncalibrated shutter

Table 36: Output Mode Register

C.3.2 Pushbutton Configuration

Offset 0x01, 8 bit - LTNUMGRP0: Pushbutton Configuration
This parameter controls the function of the pushbutton input. The parameter is divided into 2 parts: The lower 4 bit define the function and the upper 4 bit define the target group. For target group *Joker* special functions are available and the bush button can be operated as the corresponding panic, leave home or door bell Terminal Blocks.

Higher 4 bit	Target group
0	Reserved
1	Light
2	Blinds
3	Climate
4	Audio
5	Video
6	Reserved
7	Reserved
8	Joker
9 -15	Reserved

Table 37: Button Input Groups

Lower 4 bit	Description
0	local pushbutton (local + presets 2-4)
1	area 1 pushbutton (area 1 + presets 2-4)
2	area 2 pushbutton (area 2 + presets 2-4)
3	area 3 pushbutton (area 3 + presets 2-4)
4	area 4 pushbutton (area 4 + presets 2-4)
5	room pushbutton (presets 0-4)
6	extended 1 pushbutton (presets 10-14)
7	extended 2 pushbutton (presets 20-24)
8	extended 3 pushbutton (presets 30-34)
9	extended 4 pushbutton (presets 40-44)
10	extended area 1 pushbutton (area 1 + presets 12-14)
11	extended area 2 pushbutton (area 2 + presets 22-24)
12	extended area 3 pushbutton (area 3 + presets 32-34)
13	extended area 4 pushbutton (area 4 + presets 42-44)
14	apartment pushbutton
15	app pushbutton

Table 38: Button Input Id's - Group 1 .. 7

Lower 4 bit	Description
0	reserved
1	alarm
2	panic
3	leave/come Home
4	reserved
5	door bell
6 .. 14	reserved
15	app pushbutton

Table 39: Button Input Id's - Group Joker

C.3.3 Dimm Time

Offset 0x06, 8 bit - DIMTIME0_UP: Dimm Time 0 Up

This transition time is used when dimming up. The formula for time calculation is:
$T = (100ms \cdot 2^{exp}) - \frac{(100ms \cdot 2^{exp}) \cdot (15 - lin)}{32}$ whereas exp = Bit 7..4 and lin = Bit 3..0.
Examples: 0x0F = 100ms, 0x1F=200ms, 0x27 = 300ms, 0x2F = 400ms, 0x37 = 600ms.

Offset 0x07, 8 bit - DIMTIME0_DOWN: Dimm Time 0 Down

This transition time is used when dimming down. See DIMTIME0_UP for time calculation.

Offset 0x08, 8 bit - DIMTIME1_UP: Dimm Time 1 Up

This transition time is used when dimming down. See DIMTIME0_UP for time calculation.

Offset 0x09, 8 bit - DIMTIME1_DOWN: Dimm Time 1 Down

This transition time is used when dimming down. See DIMTIME0_UP for time calculation.

Offset 0x0a, 8 bit - DIMTIME2_UP: Dimm Time 2 Up

This transition time is used when dimming down. See DIMTIME0_UP for time calculation.

Offset 0x0b, 8 bit - DIMTIME2_DOWN: Dimm Time 2 Down

This transition time is used when dimming down. See DIMTIME0_UP for time calculation.

C.3.4 Flashing Mode

Offset 0x0e, 8 bit - FOFFVAL0: Off Value for Programming Mode Flashing

The digitalSTROM Device flashes when entering or exiting the programming mode. The flashing starts at the current output value, then there are FCOUNT0 flash cycles. Each cycle transitions to FONVAL0, then to FOFF-VAL0. After the last cycle, the initial output value is restored.

Offset 0x0f, 8 bit - FONVAL0: On Value for Programming Mode Flashing
This is the on value for the flashing cycles. See FOFFVAL0.

Offset 0x10, 8 bit - FOFFTIME0: Off Time for Programming Mode Flashing
The off value while flashing is held for this time. See OFFVAL0. Each step means 33ms. Example: 20 = 660ms.

Offset 0x11, 8 bit - FONTIME0: On Time for Programming Mode Flashing
The on value while flashing is held for this time. See OFFVAL0. Each step means 33ms. Example: 20 = 660ms.

Offset 0x12, 8 bit - FCOUNT0: Flash Cycle Count for Programming Mode Flashing
See OFFVAL0. 0 disables flashing, 255 flashes infinitely.

Offset 0x13, 8 bit - FOFFVAL1: Off Value for Scene Flag Flashing
The digitalSTROM Device flashes when a scene is called and the corresponding bit is set in scene configuration. The flashing starts at the current output value, then there are FCOUNT1 flash cycles. Each cycle transitions to FONVAL1, then to FOFFVAL1. After the last cycle, the initial output value is restored (if DC bit is set in scene configuration) or a transition to the called scene value takes place.

Offset 0x14, 8 bit - FONVAL1: On Value for Scene Flag Flashing
This is the on value for the flashing cycles. See FOFFVAL1.

Offset 0x15, 8 bit - FOFFTIME1: Off Time for Scene Flag Flashing
The off value while flashing is held for this time. See OFFVAL1. Each step means 33ms. Example: 20 = 660ms.

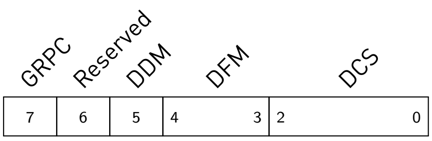
Offset 0x16, 8 bit - FONTIME1: On Time for Scene Flag Flashing
The on value while flashing is held for this time. See OFFVAL1. Each step means 33ms. Example: 20 = 660ms.

Offset 0x17, 8 bit - FCOUNT1: Flash Cycle Count for Scene Flag Flashing
See OFFVAL1. 0 disables flashing, 255 flashes infinitly.

C.3.5 LED Configuration

Offset 0x18, 8 bit - LEDCON0: LED Configuration 0

Register C.3: LED Configuration



GRPC: LED Color Mode Normally the LED shows automatically the color for the user selected group (e.g. using short-short-long). But it is also possible to overwrite this color for some scenes using the scene configuration flags and this register with GRPC=0.

GRPC	Description
0	use DCS register for specific color
1	automatic color for user programmed group

Reserved Should always be 0.

DDM: LED Brightness Select how the LED brightness is controlled. This flag is only used if GRPC is set to 1.

DDM	Description
0	full brightness
1	brightness depends on output value (higher value = brighter)

DFM: LED Flash Mode Select when the LED flashes.

DFM	Description
0 0	always on, flash during transition
0 1	always on, flash when output on
1 0	always on
1 1	always flashing

DCS: Direct Color Selection Directly select a specific color.

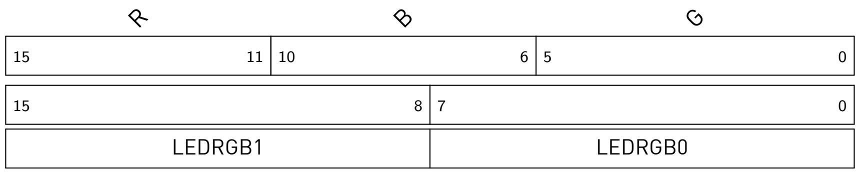
DCS	Description
0 0 0	use LEDRGB0..1 registers
0 0 1	blue
0 1 0	green
0 1 1	cyan
1 0 0	red
1 0 1	magenta
1 1 0	yellow
1 1 1	white

Offset 0x19, 8 bit - LEDCON1: LED Configuration 1
See LEDCON0.

Offset 0x1a, 8 bit - LEDCON1: LED Configuration 2
See LEDCON0.

Offset 0x1b, 8 bit - LEDRGB0: LED RGB Register 0
Offset 0x1c, 8 bit - LEDRGB1: LED RGB Register 1

Register C.4: LED RGB Register



R: LED Brightness Red 5 bit LED brightness for color red

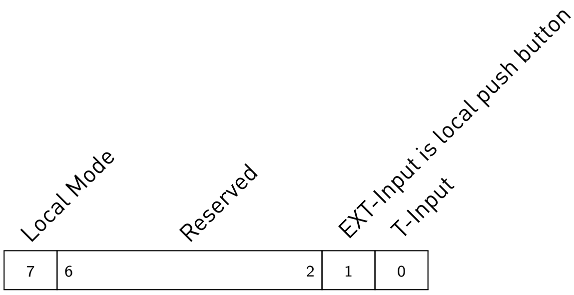
B: LED Brightness Blue 5 bit LED brightness for color blue

G: LED Brightness Green 6 bit LED brightness for color green

C.3.6 Local Pushbutton

Offset 0x1d, 8 bit - LTCON: Local Pushbutton Configuration

Register C.5: LTCON



Local Mode Push button events only local

EXT-Input EXT-Input is local push button

T-Input T-Input is local push button

Offset 0x1e, 8 bit - LTMODE: Local Pushbutton Mode

Value	Description
0	Standard button
1	Turbo button: limited operation, hold, click and programming click disabled
2 .. 4	Reserved
5	2-way up, paired with input 1
6	2-way up, paired with input 2
7	2-way up, paired with input 3
8	2-way up, paired with input 4
9	2-way down, paired with input 1
10	2-way down, paired with input 2
11	2-way down, paired with input 3
12	2-way down, paired with input 4
13	2-way up/down
14 ..15	Reserved
16	Standard binary input
17	Inverted binary input
18	Binary input, rising edge 1
19	Binary input, falling edge 1
20	Binary input, rising edge 0
21	Binary input, falling edge 0
22	Binary input, toggle mode on rising edge
23	Binary input, toggle mode on falling edge

C.3.7 Configuration Flags

Offset 0x1f, 8 bit - DEVICECFG: Device Configuration Flags

Register C.6: DEVICECFG

Protect Configuration Device configuration is locked

C.3.8 OEM/GTIN

Offset 0x2a, 48 bit - GTIN: Global Trade Item Number

Device manufacturers store the GTIN of their bundled Native Device in the GTIN register. This register is used by the digitalSTROM Server to get extended device information like descriptive text, icons, website and service URL's (see 2.4).

C.4 Class 6 - Sensor Event Table

digitalSTROM Devices may implement the sensor event table, see 7.2.4. This table can hold up to 16 entries. Each entry consists of 6 bytes. The first entry starts at offset 0x00, the second at offset 0x06, the third at offset 0x0C and so on.

Each entry consists of:

- Sensor-ID to compare
- relational operator
- threshold value to compare with
- hysteresis
- action

If an entry matches the conditions, the action is executed. The action normally is: sending the event to user level. But the digitalSTROM Device could also request the digitalSTROM Meter to poll the value of the affected sensor. It is also possible to let the digitalSTROM Device send a pushbutton event instead.

To disable a table entry, write all bytes to 0xff.

Register C.7: Sensor Event Table Offset 0

Sensor-ID	RelOp	Action
7 … 4	3 … 2	1 … 0

Sensor-ID If the digitalSTROM Device gets a new value from an internal sensor with this Sensor-ID, it will be compared.

RelOp: Relational operator 0:=, 1:<, 2:>, 3=reserved

Action 0:send push-event with entry-number, 1:request polling of Sensor-ID value, 2:send pushbutton event, 3:reserved

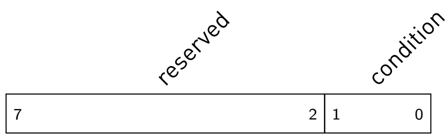
Register C.8: Sensor Event Table Offset 1..3

Threshold (offset 1, bit 11..4 & offset 2, bit 3..0) The value to compare with

Hysteresis (offset 2, bit 11..8 & offset 3, bit 7..0) The hysteresis works like this:

- For the relational operator $=$:
  - If the sensor value equals the threshold value to compare with, the action will be executed
  - Additional events for this table entry will only be executed, if the sensor value drops below $threshold - hysteresis$ or rises above $threshold + hysteresis$
- For the relational operator $<$:
  - If the sensor value is lower than the threshold value to compare with, the action will be executed
  - Additional events for this table entry will only be executed, if the sensor value rises above $threshold + hysteresis$
- For the relational operator $>$:
  - If the sensor value is higher than the threshold value to compare with, the action will be executed
  - Additional events for this table entry will only be executed, if the sensor drops below $threshold - hysteresis$

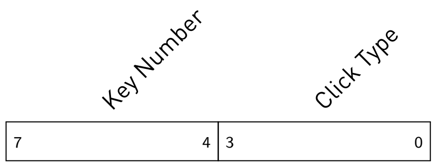
Register C.9: Sensor Event Table Offset 4



reserved reserved, write 0

condition 0:execute action always, 1:execute action only if $outputvalue = 0$, 2:execute action only if $outputvalue > 0$, 3=execute action never

Register C.10: Sensor Event Table Offset 5



Key Number only used if $Action = 2$. See 9.1.

Click Type only used if $Action = 2$. See 9.1.

C.5 Class 64 - Output status

C.5.1 Current output status

The actual device output value of the first output channel can be read back for configuration purposes from offset 0.

D Device Scene Table

The scene table contains values and settings for 128 presets. Each entry consists of a value for each output channel and configuration bits.

D.1 Scene Value

After a scene call a transition takes place from the current value to the new scene value. The output transition depends on the scene configuration bits and the internal local priority state.

The bitlength of the scene value is not set to a fixed value.

D.2 Configuration Bits

There are 8 configuration bits for each scene.

Register D.1: Scene Configuration

DT		LC		FLASH	reserved	LPRIO	DC
7	6	5	4	3	2	1	0

DT: Dim Time Control the transition time between the current value and the new scene value.

DT	Description
0 0	use last dimtime
0 1	use dimtime from register DIMTIME0
1 0	use dimtime from register DIMTIME1
1 1	use dimtime from register DIMTIME2

LC: LED Configuration Control the behavior of the LED.

LC	Description
0 0	use last LED configuration
0 1	use LED configuration from register LEDCON0
1 0	use LED configuration from register LEDCON1
1 1	use LED configuration from register LEDCON2

FLASH: Flash Configuration If this bit is set, the output flashes (as configured with *FOFFVAL1*,*FONVAL1*, *FOFFTIME1*, *FONTIME1* and *FCOUNT1*). After flashing the output transistion takes place (if *DC* is not set and internal local priority state is not set or *LPRIO* is set).

reserved Must not be modified and written back as read.

LPRIO: Ignore Local Priority If this bit is set, the output transition takes place even if the internal local priority state is set. The new local priority state depends on the newly called scene.

DC: Output Don't Care Flag If this bit is set, the output value will not be changed. The output transition will not take place.

E Device Sensors

digitalSTROM Devices optionally include up to 15 sensor objects. The list of possible sensors include internal device specific measurements and externally supplied values. The table of implemented sensor objects per device is defined per digitalSTROM Device parameters (see C.2.8).

Each of the sensor objects has a well defined type. The following table defines the sensor types used in the system. Unused sensor type numbers are reserved. All sensor types can be sent upstream, sensor types 0..63 can also be sent downstream to digitalSTROM Devices.

Sensor Type	Description	Unit	Min	12 Bit Max	12 Bit Resolution
4	Active power	Watts (W)	0	4095	1
5	Output current	Ampere (mA)	0	4095	1
6	Electric meter	Watthours (kWh)	0	40,95	0,01
9	Temperature indoors	Kelvin (K)	230	332,375	0,25
10	Temperature outdoors	Kelvin (K)	230	332,375	0,25
11	Brightness indoors	Lux (Lx)	1	131446,795	logarithmic: $lx = 10^{\frac{x}{800}}, x = 800 * \log(lx)$
12	Brightness outdoors	Lux (Lx)	1	131446,795	logarithmic: $lx = 10^{\frac{x}{800}}, x = 800 * \log(lx)$
13	Relative humidity indoors	Percent (%)	0	102,375	0,025
14	Relative humidity outdoors	Percent (%)	0	102,375	0,025
15	Air pressure	Pascal (hPa)	200	1223,75	0,25
18	Wind speed	m/s	0	102,375	0,025
19	Wind direction	degrees	0	511,875	$\frac{0,5}{4}$
20	Precipitation	mm/m2	0	102,375	0,025
21	Carbon Dioxide	ppm	1	131446,795	logarithmic: $ppm = 10^{\frac{x}{800}}, x = 800 * \log(ppm)$
25	Sound pressure level	dB	0	255,938	$\frac{0,25}{4}$
50	Room temperature set point	Kelvin (K)	230	332,375	0,025
51	Room temperature control variable	Percent (%)	0	102,375	0,025
64	Output current (H)	Ampere (mA)	0	16380	4
65	Power consumption	Volt-Ampere (VA)	0	4095	1

Table 40: Sensor Types

F Device Registration

A device needs to be registered at the digitalSTROM Meter who also tracks the device availability state. If a new device is connected to the circuit, the device sends its unique device ID (dSID) to the digitalSTROM Meter. The digitalSTROM Meter then assigns a circuit-wide, unique short address to the device. If the device was seen in the circuit before, it will get the same short address as before. At registration time some parameters are synchronized with the digitalSTROM Meter. The parameters to read depend on the Function-ID which is read first. More parameters are: pushbutton configuration (room/area/...), output mode (switchable or dimmable) and Product-ID.

If the digitalSTROM Meter is powered on, there is no need to register all devices again. The digitalSTROM Meter sends its own unique ID to the devices and only devices which were not registered at this digitalSTROM Meter before will then request a registration.

The digitalSTROM Meter detects if there are more than one devices which needs to be registered at the same time. It will then switch to a slower registration mode which allows to register up to 128 new devices at the same time.

The digitalSTROM Meter checks the device availability every 24 hours. Each known device is contacted by the digitalSTROM Meter and if the device does not respond, the state will be set to inactive.