ENGINEERING
TOMORROW

Feature Catalogue

Ally™ electronic Radiator Thermostat (eTRV)

ENGINEERING TOMORROW

1. Basic Features

Needed for the eTRV to work correctly

1.1 Radio interface configuration

1.2 Time Synchronization

1.3 Display Rotation

1.4 eTRV Orientation

1.5 Diagnostics

1.6 Valve adaptation

1.1 Radio interface configuration

Since the eTRV is a battery-powered end-device, the radio communication requested will have a significant impact on battery lifetime.

The reporting configuration should not be configured to intervals below the default, in order to have a maximum of 650 radio communications per day (average), and the GW must not send unnecessary requests to the eTRV.

The check-in interval is by default 5 min. It can be configured to a time between 1 – 7208960 quarter seconds. In order to comply to the conditions needed to achieve the estimated battery lifetime, 5 min radio communication wake-up interval is used.

The long poll interval can be increased up to the check-in interval, using the "Set long poll interval" command. Increasing the long poll interval will preserve battery but reduces the device ability to respond fast to Zigbee communication.

Cluster	Attribute	Name	Function
0x0020	0x0000	Check-in Interval	Radio control wakeup interval. Default 1200, means 5 min in quarter seconds (1200/4 = 300/60 = 5). Check-in interval is the time between intervals.
0x0020	0x0001	Long Poll Interval	A 28 quarter seconds interval, where the device can receive data from a controller.

1.2 GW to set Time in Time Cluster – recommendation

The eTRV has no knowledge about current time and expects it to be set from a GW.

Weekly Schedule will not work if time is not set in the eTRV, also Valve Exercise does not happen the correct time and the Valve Adaptation algorithm does happen at the correct time.

Attributes mandatory to be set:

Cluster	Attribute	Name	Value
0x000A	0x0000	Time	Seconds since 0 hrs 0 mins 0 sec on 1st January 2000 UTC
0x000A	0x0001	TimeStatus	Write required with only bit 1 set required eg 0x02
0x000A	0x0002	TimeZone	According to timezone that could be taken from the app. Value in seconds.
0x000A	0x0003	DstStart	DstStart value according to current year and adjusted for installation country.
0x000A	0x0004	DstEnd	DstEnd value according to current year and adjusted for installation country.
0x000A	0x0005	DstShift.	This value is normally 3600 for 1 hour or 0. adjusted for installation country.

1.2 GW to set Time in Time Cluster

eTRV will loose time information at battery change, OTA, hardware resets, watchdog error, etc.

Action required by GW:

When eTRV joins gateway or rejoins

- Time status shall be checked in Diagnostics Cluster Attribute 0x4000.
- if Bit 10 (0x0b05 0x4000 ”SW Error code”) indicates that Time is lost, the GW shall write the 6 attributes specified.

Action required by GW (mostly if running schedule):

At least once every year

- DstStart and DstEnd needs to be adjusted for the current year before enter DST period.

Recommended action by GW:

At least once every week (mostly important if the schedule functionality is used)

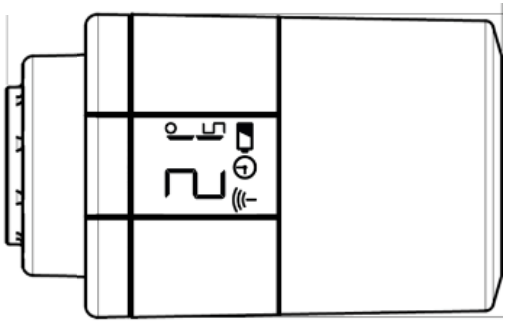
- GW to set time in eTRV to adjust timing of the eTRV to have a precise running clock.
- GW to write all 6 attributes specified on earlier slide.

1.3 Display Rotation setting

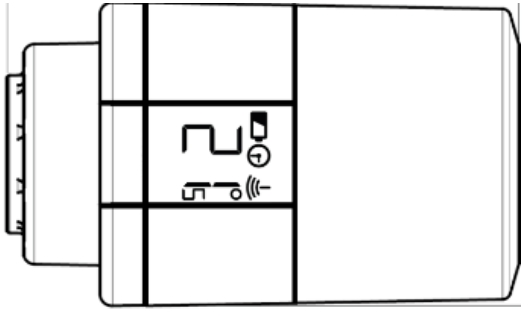
Viewing Direction is only the direction in which the LCD shows information, there will be no impact in actual temperature measure/estimation or heating control behaviour. The feature can be controlled via the attribute:

Cluster	Attribute	Name	Function
0x0204	0x4000	Viewing Direction	Range: 0 to 1 0x00 = viewing direction 1 0x01 = viewing direction 2 Default is 0

Default (0):



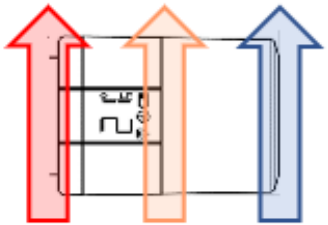
Inverted (1):



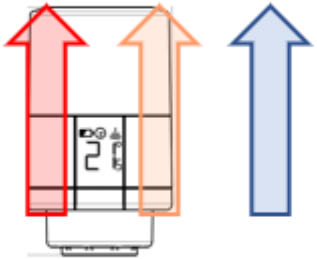
1.4 eTRV Orientation

Orientation is the actual direction in which the eTRV is mounted:

- When mounted horizontally, the eTRV will measure temperature and calculate an estimated room temperature according to one algorithm considering the typical temperature gradient in proximity of the radiator.



- When mounted vertically, the eTRV will measure temperature and calculate an estimated room temperature using a different algorithm that compensates the measure because of the "column" of hot air running through the eTRV in this position.



The orientation will impact the actual heating function of the eTRV and can be controlled via the attribute:

Cluster	Attribute	Name	Function
0x0201	0x4014	eTRV orientation	0x00: Horizontal (Default) 0x01: Vertical

1.5 Error States

Error and fault states will be sent by the eTRV via the SW Error Code attribute:

Cluster	Attribute	Name	Function
0x0b05	0x4000	SW Error code	A bit map signalling the error state of the eTRV.

Each active error corresponds to a bit in the following bitmask

Bit index	(msb) 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Set =	Reserved	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1

1.5 Error States

Code	Description	Description
E1	Top PCB sensor error	Device faulty
E2	Side PCB sensor error	Device faulty
E3	NVM error	Device faulty
E4	Unknown HW error	Device faulty
E5	N/A	N/A
E6	Motor error	Motor jammed, try re-start, try reboot, try re-mount, if the error persists the device is faulty
E7	N/A	N/A
E8	Invalid communication	Invalid internal communication, try reboot (re insert batteries), if the error persists the device is faulty (if not persistent, the error should disappear after 10 minutes)
E9	N/A	N/A
E10	Invalid clock information (need resynchronization)	You have recently replaced battery and lost time synchronization.
E11	N/A	N/A
E12	Radio communication error	Communication with GW lost (will be active for 2 minutes after re-connection for collecting diagnostic if needed)
E13	Encoder Jammed	N/A
E14	Low Battery	Low Battery, the device will soon run out of batteries, replace ASAP
E15	Critical Low Battery	Batteries Empty, Replace immediately, The eTRV cannot control the valve longer and has opened the valve to prevent potential frost damages to the system

1.5 Other Diagnostics/Status Information

Cluster	Attribute	Name	Function
0x0B05	0x011C	LastMessageLQI	Indicates the link quality for last received messages
0x0B05	0x4010	Motor step counter	A value holding the accumulated motor steps over the lifetime of the eTRV.
0x0001	0x0021	BatteryPercentageRemaining	0-200
0x0201	0x4000	eTRV Open Window Detection	0x00: Quarantine 0x01: Windows are closed 0x02: Hold ,Windows are maybe about to open 0x03: Open window detected 0x04: In window open state from external, but detected closed locally
0x0201	0x4012	Mounting mode active	0x00: Mounted 0x01: Not mounted (after factory reset) Default is 0, but overwritten to actual status at Init. UI possibility: Notify “your eTRV has been fitted/mounted to a valve and is ready to work”
0x0201	0x404D	Adaptation run status	bit0=adaptation run in progress bit1=adaptation run successful, Valve Characteristic found bit2=Valve Characteristic lost/invalid UI possibility: Notify, “your eTRV has completed an Adaptation cycle successfully”
0x0201	0x0008	PIHeatingDemand	% of valve opening
0x0201	0x404A	Load on this radiator	Parameter used for load balancing

1.5 Battery level – example

Full battery:
Green Icon

Cluster	Attribute	Name	Value
0x0B05	0x4000	SW Error code	Bit13=FALSE & Bit14=FALSE

Low battery:
Yellow Icon +
Notification

Cluster	Attribute	Name	Value
0x0B05	0x4000	SW Error code	Bit13=TRUE & Bit14=FALSE

Critical Low battery:
Red Icon +
Notification

Cluster	Attribute	Name	Value
0x0B05	0x4000	SW Error code	Bit14=TRUE

Alternatively, use Battery Percentage where:

Low Battery = Below 25%

Critical Low Battery = Below 12,5%

1.6 Adaptation Run* Control and UI

The eTRV needs to estimate the Valve characteristic in order to work at full performance.

This information is important for the eTRV to achieve energy efficiency, comfort and prompt response.

In order to do this the eTRV runs an "Adaptation Run".

The "Adaptation Run" takes 1 to 3 hours (typically 1-2 hours): at a point during this period the radiator will suddenly get warm for circa 15-30 minutes. The **heating must be on during adaptation run**.

The adaptation data is refreshed every time there is a "cold" period followed by an active heating period e.g. when running heating schedule (the eTRV runs adaptive algorithms and statistics with these data to adapt to changes and get more and more precise during the whole lifetime).

Since the first Adaptation Run needs heating, it could be annoying **for users with gas boilers** if run during the night **(default setting).**

For these users we suggest to implement a UI to guide the user through this procedure or disable the automatic feature (see next slide).

* The term "Adaptation Run" comes from the EN15500 standard and from eu.bac certification rules, some users may find it a weird term to understand. "Measure", "Calibration", "Fit", "Adjustment"… might be more intuitive terms.

1.6 Adaptation Run Control Recommended UI example

Customer Journey:

1. Gateway already installed.
2. Unbox eTRV.
3. Join the eTRV Via AppUI.
4. User prompted to configure eTRV (room – temperatures etc.).
5. User alarmed by e.g. banner or red dot "push for more info".
6. User informed "Your Ally™ eTRV needs to learn/adapt to your radiator valve in order to react promptly and be energy efficient. It will do it automatically during the next few days, but you may experience lower performance until done. You can avoid waiting for full performance by starting the adaptation run now. Do you want to start an Adaptation Run now?" At this point or in FAQ there could be reference to situations of unwanted heat (your heating went on during the night because of automatic-adaptation run) or slow start of radiator (it took up to 1 hour before your radiator got warm at schedule change).
7. If started, Adaptation Run takes 1 to 3 hours (typically 1-2 hours): at a point during this period the radiator will suddenly get warm for circa 15-30 minutes.
8. In the eTRV UI on the app the "Adaptation Run Status" could be shown as an alarm, but **only after first installation**. During the product lifetime, the adaptation value may become inconsistent (e.g. something changes mechanically… eTRV hit by a door or other impact, thermal deformation, tightened on adaptor again, tap crawl etc…) therefore the eTRV might repeat the adaptation procedure after e.g. one year (extremely rare occasions). This is normal behaviour and should not alarm the user.

NOTE: User interaction (setpoint command type 1) cancels the Adaptation Run!!!

1.6 Adaptation Run Controllable via Zigbee

Attribute "Adaptation Run Control"
ENUM8, not NVM saved, only RAM
1=Initiate Adaptation run
2=cancel Adaptation run

Cluster:	(0x0201) Thermostat			
Attribute ID	Name	Data Type	R/W	Reporting
0x404C	Adaptation Run Control	enum8	R/W	No

Attribute "Adaptation Run Status"
8Bit, not NVM saved, only RAM
bit0=adaptation run in progress
bit1=adaptation run Successful
Reporting default at change

Cluster:	(0x0201) Thermostat			
Attribute ID	Name	Data Type	R/W	Reporting
0x404D	Adaptation Run Status	bitmap8	R	Yes

Attribute "Adaptation Run Settings"
8Bit, NVM saved,
1=Automatic adaptation run enabled
(runs during the night)

Cluster:	(0x0201) Thermostat			
Attribute ID	Name	Data Type	R/W	Reporting
0x404E	Adaptation Run Settings	bitmap8	R/W	No

Automatic adaptation run will be default ON, it can be canceled via "Adaptation Run Control".

When the unit receives the command "do the adaptation run NOW!" the adaptation run will start "immediately" (starting by closing the valve) unless the Valve characteristic is already found! (note: the unit can be set in Mounting Mode to repeat the adaptation run).

At any time the GW can set the "Enable automatic adaptation run", the eTRV will run the automatic adaptation run during the next coming night as in current implementation, the feature will run only if the Valve characteristic is not yet found.

Adaptation Run can be canceled by entering Mounting Mode or by turning the dial on the thermostat.

ENGINEERING TOMORROW

2. Basic Comfort Features

Needed for the eTRV to perform in Energy Efficiency and Comfort

2.1 Setpoint change/control

2.2 Load Balancing

2.3 Room Sensor Integration

2.4 Room Sensor Integration with Covered Radiators

2.5 Temperature Setpoint Offset

2.6 Boiler Integration

2.1 Setpoint Change

Temperature setpoint can be adjusted between the limits stated in the table below.

Setpoint adjustment is done at the TRV by turning the handle. Temperature can be set from the App/Gateway via the Zigbee interface, by changing the value of the occupied heating setpoint.

In order to replicate the behavior during a user interaction at the TRV dial, a manufacturer specific command has been implemented as well to handle the different situations.

Cluster	Command	Name	Function
0x0201	0x40	Setpoint Command	The GW sends SetpointCommand+Type(8bit)+HeatingSetpoint(16bit) 0: "Schedule Change" Just changes occupied heating setpoint. No special behavior, the PID control setpoint will be update with the new setpoint**. This is equivalent to sending "OccupiedHeatingSetpoint" to the eTRV directly** **1: "User Interaction" Changes occupied heating setpoint and triggers an aggressive reaction of the actuator as soon as control SW runs, to replicate the behavior of turning the dial on the eTRV.** 2: "PreHeat" changes only internal setpoint used in the eTRV control and the change is not directly visible for the user, that is, the display is not updated with the transmitted setpoint.

When the temperature is adjusted the TRV will send information about the setpoint source to help the GW to Coordinate.
Relevant Attributes:

Cluster	Attribute	Name	Function
0x0201	0x0003	Abs Min Heating Setpoint Limit	A **read only** value for the lowest possible setpoint
0x0201	0x0004	Abs Max Heating Setpoint Limit	A **read only** value for the highest possible setpoint
0x0201	0x0012	Occupied heating setpoint	The active setpoint in the eTRV
0x0201	0x0015	Min Heating Setpoint Limit	An adjustable limit for restricting how low a user can adjust the setpoint on the eTRV
0x0201	0x0016	Max Heating Setpoint Limit	An adjustable limit for restricting how high a user can adjust the setpoint on the eTRV
0x0201	0x0030	Setpoint Change Source	0x00: Manual, user-initiated setpoint change 0x01: Schedule, program-initiated setpoint change 0x02: Externally initiated Setpoint change

2.1 Reaction aggressiveness at user interaction

Bigger actuator movement at user interaction

During a User Interaction (setpoint change by turning the dial or setpoint change via ”SetpointCommand + UI=1”) the motor jumps to a much different position **regardless of limitations for battery saving and energy efficiency**.

NOTE: The aggressive command (type 1) must not be used all the time for schedule changes!!! If done, it could reduce comfort, energy efficiency and battery lifetime!

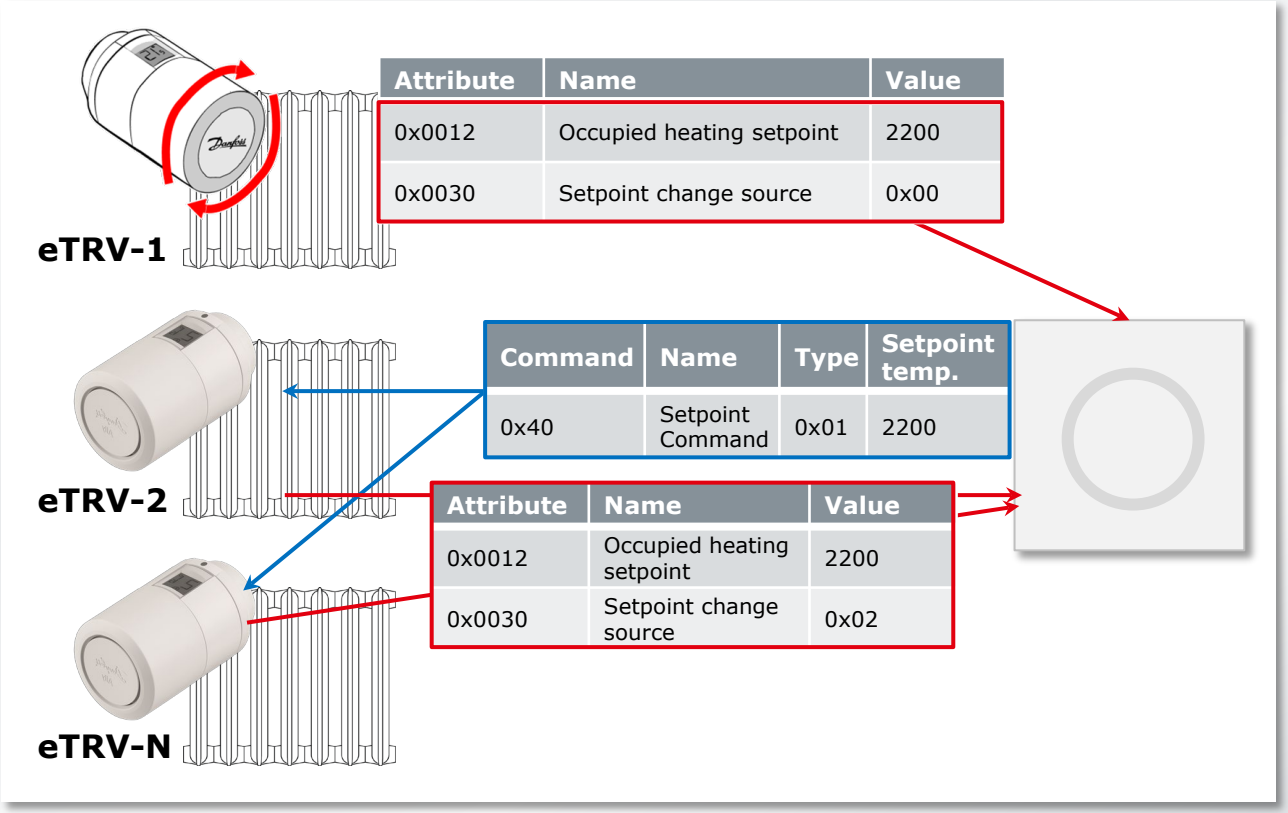
Setpoint command sends: UserInteraction (#VALUE) + HeatingSetpoint (16bit).

Thermostat Cluster (0x0201)		
Command Id	Command Name	Direction
0x40	Setpoint Command	client->server

2.1 Use case example – User Interaction at eTRV

Adjusting temperature at eTRV1 when there are 2 or more eTRV in the room previously at 21 degrees setpoint. That is, if Setpoint Change source (0x0030) is 0x00 on an eTRV the setpoint needs to be transmitted to the other eTRV as a user interaction.

The App UI could show a banner to the user saying "temporary setpoint", (if possible clickable banner with explanation) to be shown until the eTRV sends a new scheduled temperature information (see use case – scheduled setpoint change at eTRV).

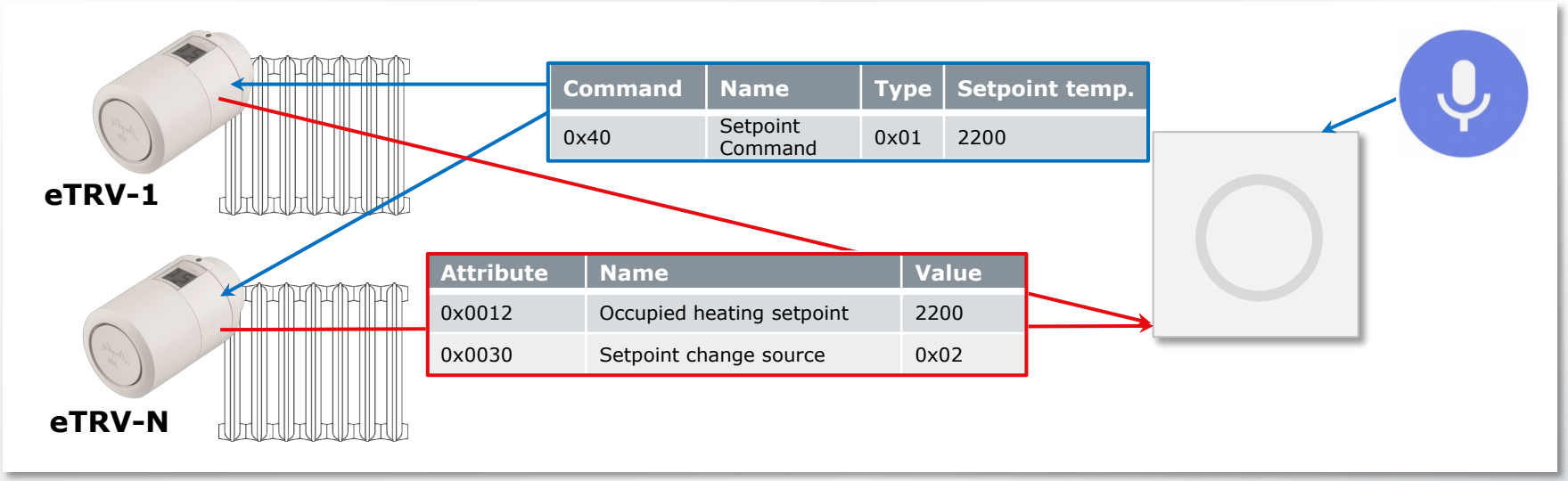


2.1 Use case example – User Interaction via voice control

The user asks to set a new setpoint (e.g. 22°C) in a room or device via voice control.

The GW sends a setpoint command to each eTRV in the room.

If accessed, the App UI could show a banner to the user saying “temporary setpoint”, (if possible clickable banner with explanation) to be shown until the eTRV sends a new scheduled temperature information (see use case – scheduled setpoint change at eTRV).



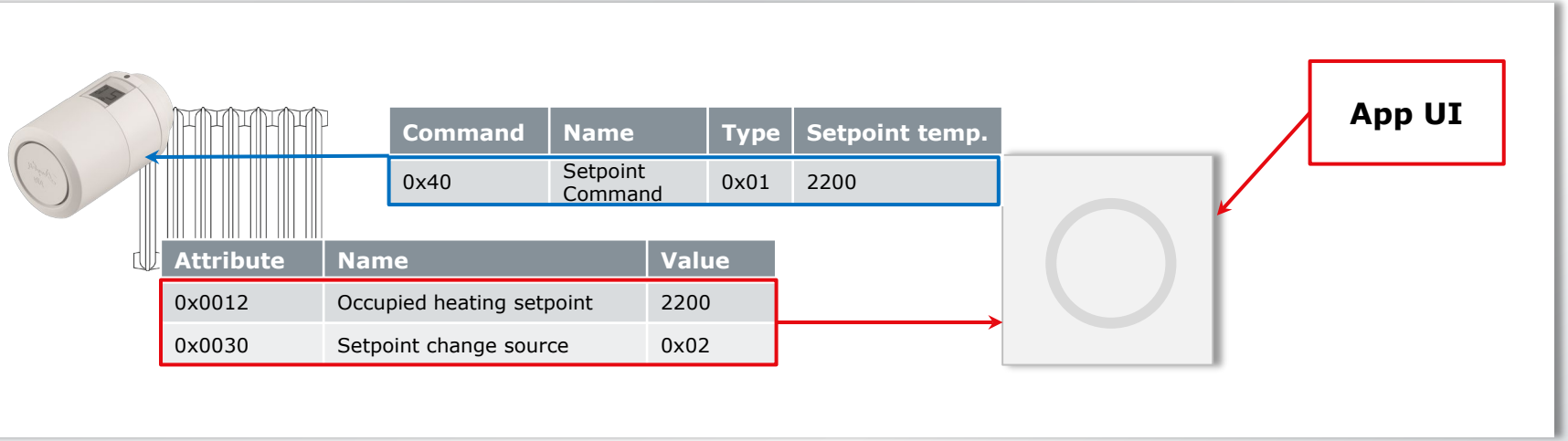
2.1 Use case example – User Inter. via App in Manual Mode

The user changes the current setpoint in manual mode.

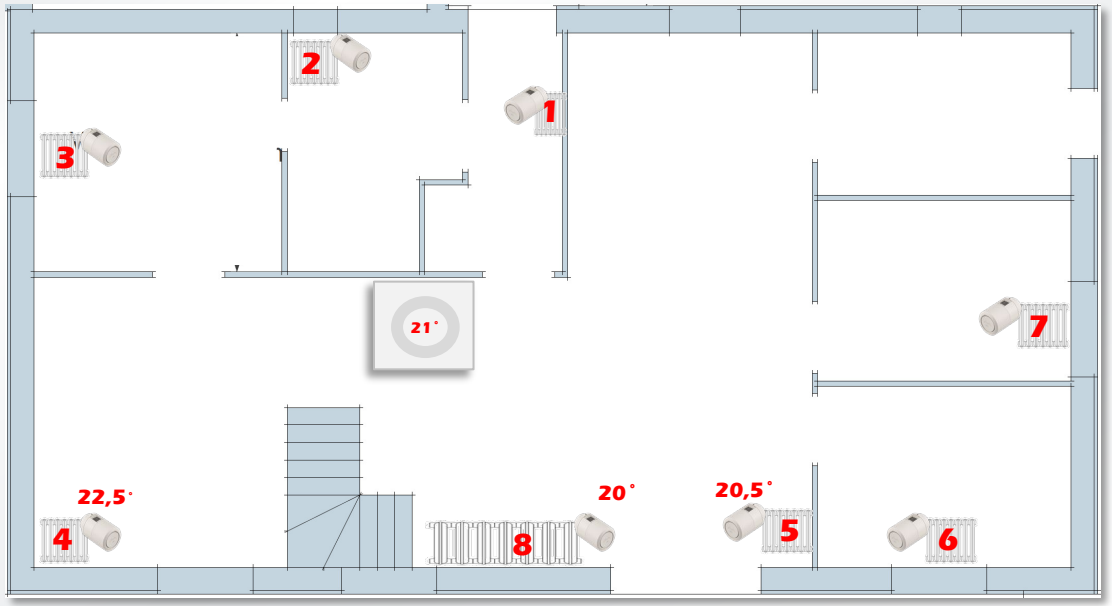
The system should react as soon as possible to the change to satisfy the user's need.

The communication below is repeated for all the eTRVs in the same room.

The system should avoid to attempt "real time" update of the temperature setpoint and only send the last setting from the App. (e.g do not send 1 temperature setpoint every 0,5 degrees when adjusting temp).



2.2 Load Balancing



In this example shown radiator 8 is bigger than 5 and it can heat up also the area around 5.

Each eTRV is an independent controller, so, if 8 can heat up the whole room by itself, it will.

In this case 5 is not needed to heat the room but a user might see this as a malfunction.

A dominant radiator or a radiator placed in a cold area (e.g. less insulated or exposed wall) will "kill" the other ones in the room and they will turn cold.

This situation can be avoided with an additional function called "Load Balancing" that ensures that all radiators contribute to heating the same air space.

2.2 Load Balancing

For **Load Balancing** is intended a feature aimed to distribute the room need for heat (load) between 2 or more radiators in the same room.

Each eTRV in the room will report its own load level to the gateway via the attribute 0x404A "Load estimate on this radiator".

The gateway calculates an average of the load for all the radiators in the room and distributes it to all the eTRVs in the room via the attribute 0x4040 "Load Radiator Room Mean" **every 15 minutes**.

The Gateway must discard all the values below -500 (too low) down to -8000 (invalid/inactive) and values older than 90 minutes. The average must then be calculated with the values from the other eTRVs in the room.

The feature is enabled by default on the eTRV. The eTRV will start reacting to the information from the gateway as soon as the average room load information is received.

To disable the feature the GW has 2 possibilities:

1 - set the attribute "0x4032 Load Balancing Enable" enable to "FALSE".

2 - Send the value -8000 (Invalid) via the attribute 0x4040 "Load Radiator Room Mean".

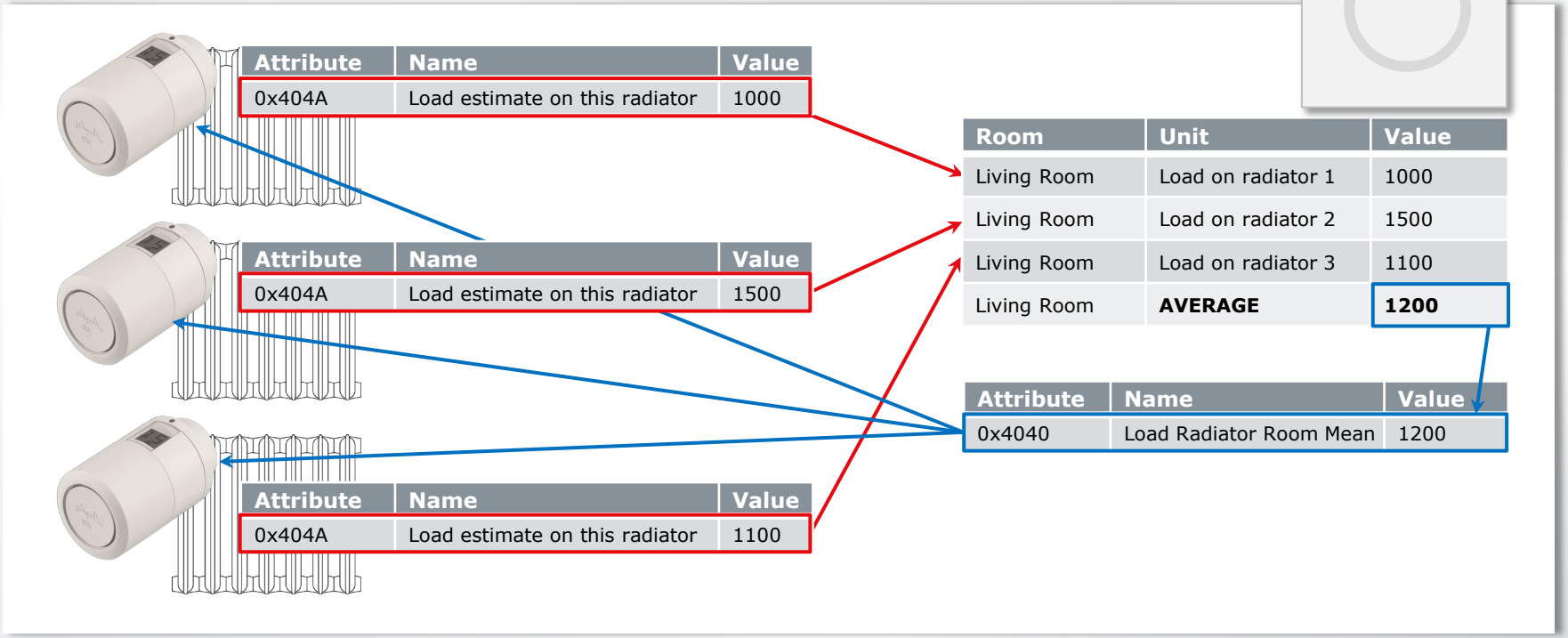
Notes:

- This control **must not be used in rooms with only 1 eTRV.**
- If the load radiator Room Mean information is not sent by the gateway for more than 90 minutes the eTRV will go back to normal mode.

Cluster	Attribute	Name	Type	Default
0x0201	0x404A	Load estimate on this radiator	Int16	0xE0C0 (-8000)
0x0201	0x4040	Load Radiator Room Mean	Int16	0xE0C0 (-8000)
0x0201	0x4032	Load Balancing Enable	boolean	TRUE

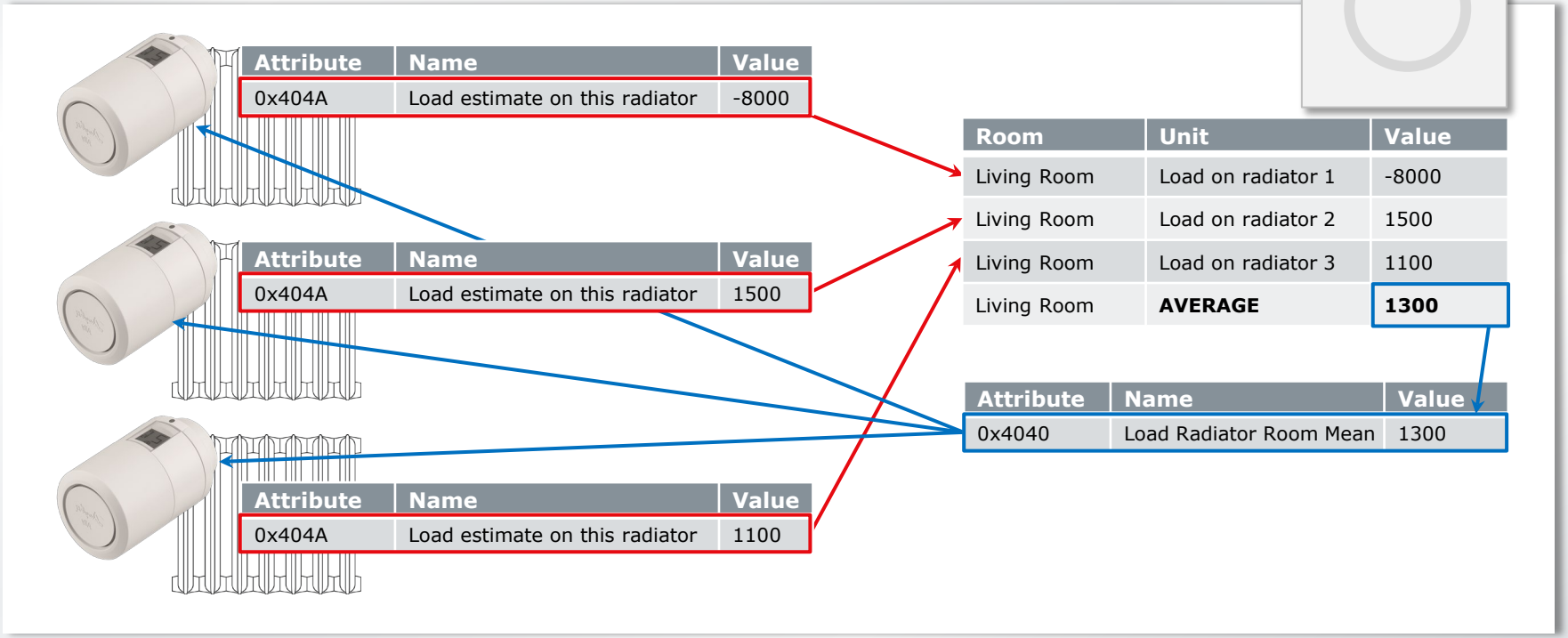
2.2 Load Balancing Examples and exceptions 1/3

Normal Situation all the TRVs are running.



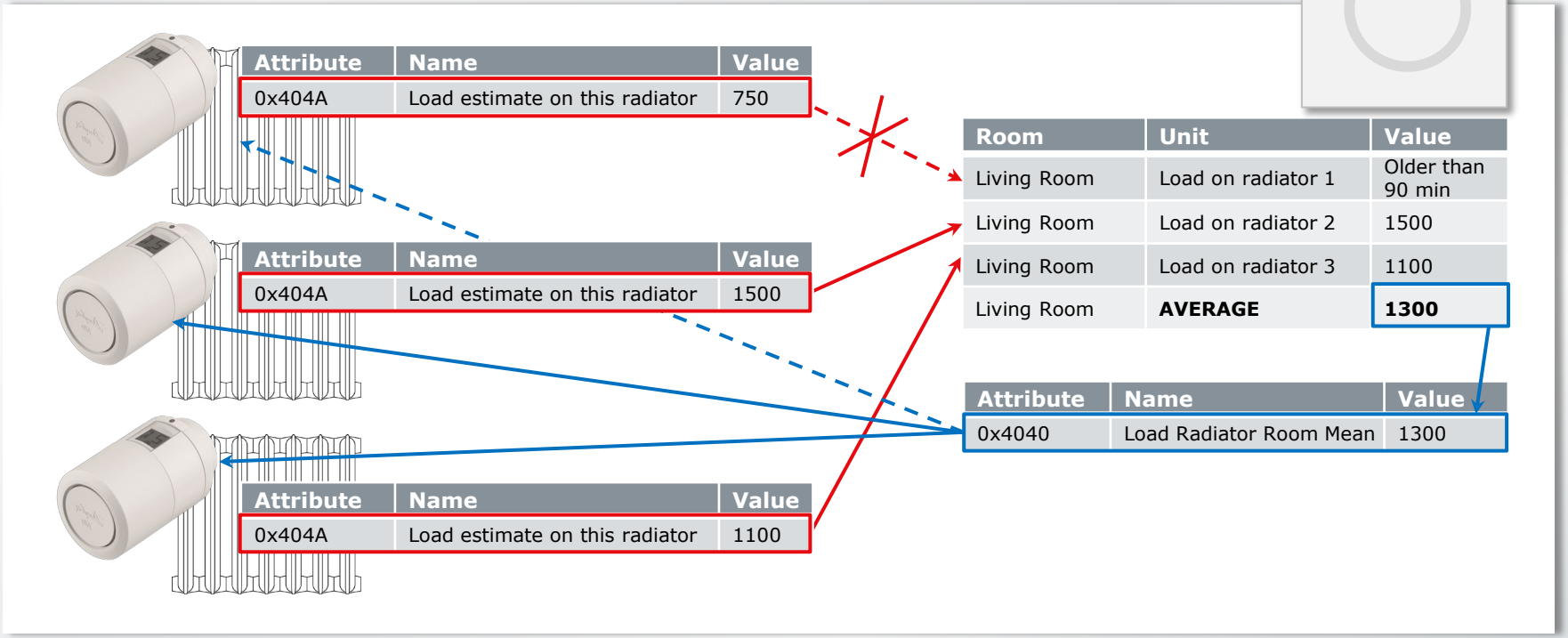
2.2 Load Balancing Examples and exceptions 2/3

Normal Situation one TRV busy/inactive (e.g. during adaptation run or window open).



2.2 Load Balancing Examples and exceptions 3/3

Fault Situation one TRV off-network.



2.3 External room sensor integration – Automatic Offset

In the example shown the GW system is in the center of an air space to which radiators 4, 5 and 8 contribute heating.

The temperature measured at the GW device should be sent "as measured" to all the 3 eTRVs at least every 3 hours but not more often than every 30 minutes @ every 0,1K change.

The radiators will automatically offset to +1,5, -0,5, 0 (assumed steady state condition in the example).

Each eTRV will offset it's setpoint to match as good as possible. Since every eTRV will be in different corners of the same room their temperature will be "wrong" or at least not representative for the temperature in the GW position due to natural temperature gradient, influence of cold wall behind GW sensor etc…

All the other eTRVs must act as standalone and not be fed with any "External Measured Room Sensor".

2.3 Using Automatic Offset Feature External room sensor Temperature

Depending on where the eTRv is placed in the room, furniture covering it, or a cold wall, the estimation will be off. To compensate for this, the setpoint for the controller can be adjusted via radio with an automatic and adaptive offset function that corrects the temperature thanks to the information from a room/wall sensor (correction range +/- 4K).

The adjustment will not be visible for the user on the eTRV display.

As soon as the Gateway sends the temperature measured by a wall sensor, the Ally™ eTRV will use it to adjust its own temperature by adding a dynamic offset up to +/-4K in order to compensate for the relative position of the measuring device in the room.

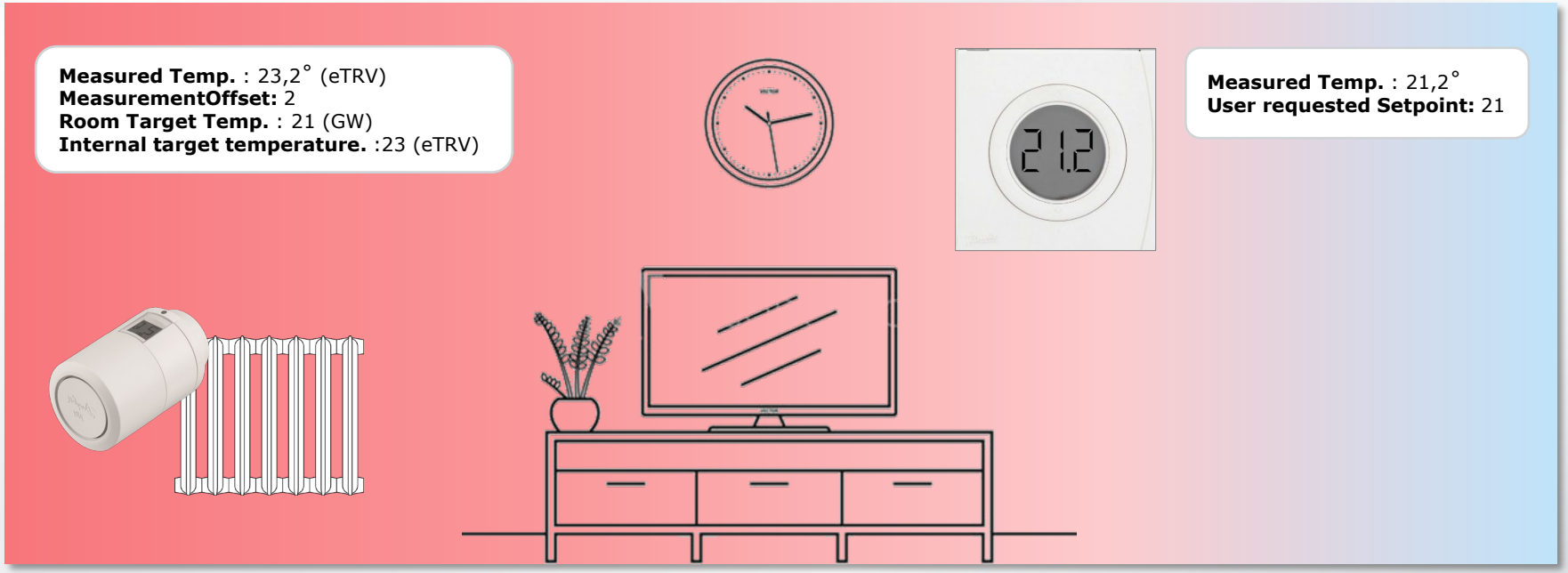
This feature is not suitable for covered radiators and the room sensor/external sensor must be in the same room with the eTRV (e.g. room sensor in the hallway + eTRV in the rooms is NOT OK).

0x0201	0x4016	Radiator Covered	FALSE = Exposed Radiators (Automatic setpoint Offset feature).
0x0201	0x4015	External Measured Room Sensor	Recommended to be received from Gateway at least every 3 hours but not more often than every 30 minutes @ every 0,1K change. After 3 hours the function is disabled and goes back to standard mode The value -8000 disables the function.

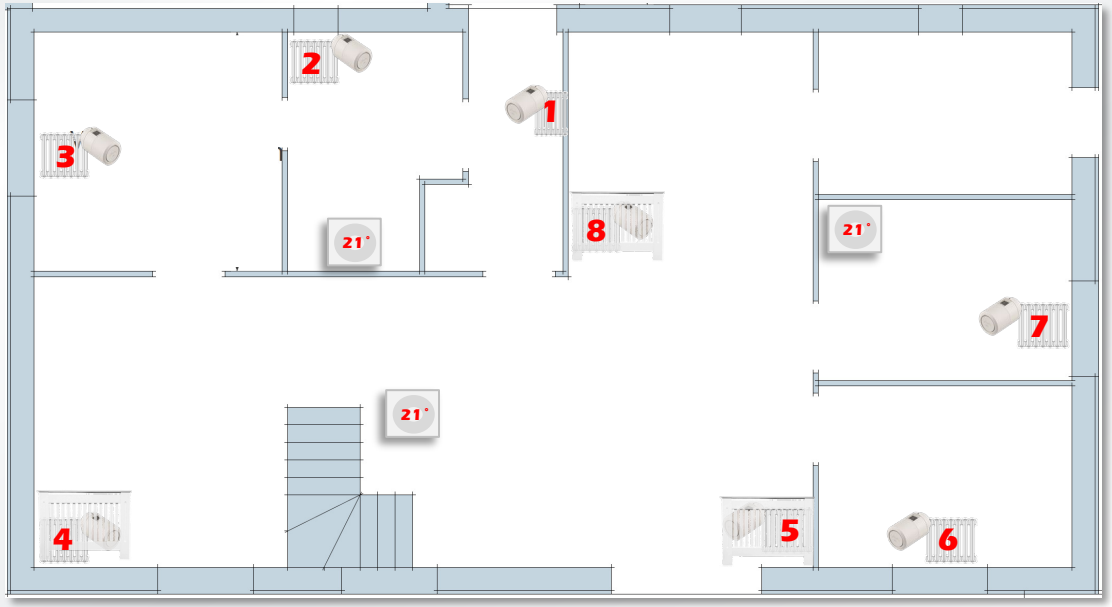
2.3 Room Sensor Application

In this example we use the Room Sensor to decide what is the right room temperature, e.g. if the radiator is located in another location than where the temperature setpoint is desired. The eTRV is provided the temperature measurement from the Room Sensor and internally in the eTRV this is used to derive a temperature measurement offset (the offset is limited to +/- 4 °C).

This functionality can be used with a combination of supply and return mounted eTRVs but not covered radiators.



2.4 External room sensor integration – Covered Radiators



In the example the system has some rooms where an extra room sensor is mounted, this could be because the radiators in those room are covered or just because temperature control is desired in a precise place in the room (e.g. thermostat close to the floor will not control temperature realistically at 1,5m height).

The temperature measured at the room sensors device should be sent “as measured” to all the devices that are in the same room at least every 30 minutes but not more often than every 5 minutes @ every 0,1K change.

The radiator thermostats will use the information in their control algorithm in order to achieve the temperature at the room sensor.

For rooms with multiple radiators the use of the “Load Balancing” feature is recommended.

2.4 Using Covered Radiator Feature External room sensor Temperature

When a radiator is concealed behind a radiator cover or, in some situations, behind a heavy curtain or furniture, it is not possible for the eTRV to estimate the room temperature correctly.

When sufficiently enclosed, the temperature inside the enclosure is not simply an offset due to temperature gradient in the room and the Automatic Offset Feature is no longer sufficient to control temperature and achieve a good comfort in the room.

The eTRV can be configured to use the information from an outside sensor instead of its own to control the temperature with a separate control algorithm.

This feature can be used for exposed radiators as well, in any case the room sensor/external sensor must be in the same room where the eTRV is installed (e.g. room sensor in the hallway + eTRV in the rooms is NOT OK).

Note: The Window open detection feature is disabled in Covered Radiator Mode.

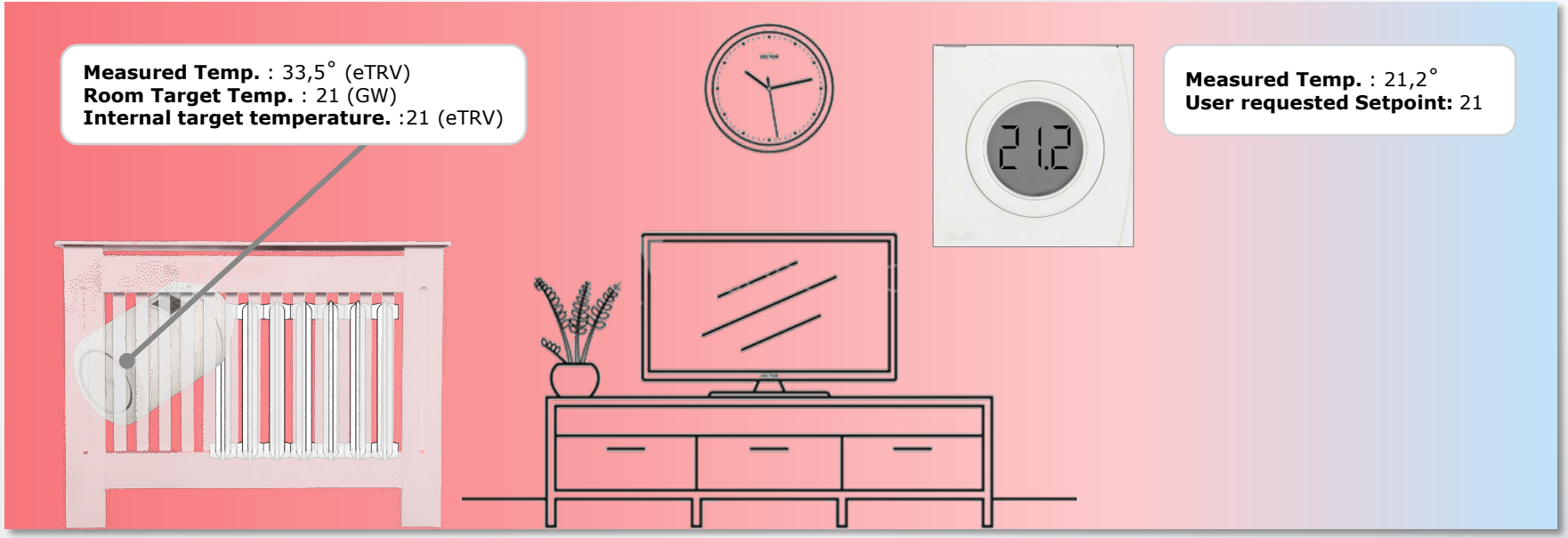
0x0201	0x4016	Radiator Covered	TRUE = Covered Radiators Mode
0x0201	0x4015	External Measured Room Sensor	At least every 30 minutes but not more often than every 5minutes @ every 0,1K change for **covered radiators** (after 35 minutes the function is disabled and goes back to standard mode) The value -8000 disables the function

2.4 Room Sensor use in Covered Radiator Mode

In this example we use the Room Sensor to feed the eTRV with the room temperature.

The eTRV is provided the temperature measurement from the Room Sensor and internally in the eTRV this is used to drive the control algorithm. As shown, the temperatures inside the cover/at the radiator are most dependent on other factors (like water temperature or room gradient) and might not have any evident correlation with the room temperature.

This functionality can be used both with covered radiators and exposed radiators.



2.5 Temperature Setpoint Offset

Depending on where the eTRv is placed in the room, furniture covering it, or a cold wall, the estimation will be off. To compensate for this, the setpoint for the PID controller can be adjusted via radio, +/- 2,5K.

The adjustment will not be visible for the user on the eTRV display.

Example of usages: The user has requested 21C degrees. Since the eTRV is slightly covered, it will always measure a bit higher than the real room temperature, therefore the Local temperature calibration is set to 2K (20). This will be added to the user set point, consequently the PID setpoint will be 21+2 = 23 C.

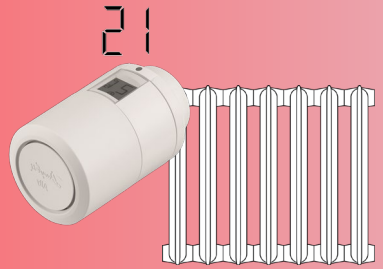
0x0201	0x404B	Regulation SetPoint Offset	An offset to the internal temperature control. The range is limited -25 to +25 -25 (dec) means that the setpoint for the PID controller is reduced by 2,5 C

2.5 Gradient in the room Room Thermostat vs. eTRV

Depending on the eTRv placement, the current load of the room and room physics, there will be a gradient in the room, in this example the area close to the radiator is approximately 2 degrees warmer than the wall where the Room Thermostat is placed.

This in case the temperature is controlled solely by the eTRV and the Room Sensor is used only to show the measured temperature. The LCD on the eTRV shows the current room setpoint = 21.

Measured Temp. : 21,2°
Room Target Temp. : 21
Offset: 0
Internal Target Temp. :21



Measured Temp. : 19,2°
Room Target Temp. : 21

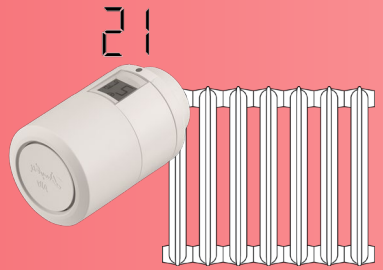
The eTRV always and only shows the target room temperature setpoint.
The room thermostat might show the actual room temperature and will display always something different from the target temperature, in this case the user will notice a difference between the target temperature and the measured temperature. The user might be able to notice that the temperature is ”typically” 2 degrees Off what he/she selected.

2.5 Gradient in the room Room Thermostat vs. eTRV

In this example we use the Room Thermostat to decide what should be the right setpoint (target temperature) for the eTRV.

The system decides that, with the current load, the eTRV should aim to a target temperature 2 degrees higher than the actual setpoint and sends a message via the Zigbee attribute 0x0201/0x404B/”Regulation SetPoint Offset” to set the offset to +2 degrees. The LCD on the eTRV still shows the current room setpoint = 21.

Measured Temp. : 23,2°
Room Target Temp. : 21 (LCD)
Offset: +2
Internal Target Temp. :23



Measured Temp. : 21,2°
Room Target Temp. : 21

The eTRV always and only shows the target room temperature setpoint.
The display will show 21, but due to the offset, the PID control will aim for 23 degrees.
If the UI on the app allows the user to see the measured temperature for all the devices, the user might notice that the actual measured temperature at the eTRV is 2 degrees higher than the room setpoint.

2.6 Inhibit Heat Request in small boiler systems e.g. individual gas boilers

In systems with gas Boilers the activation of the boiler might be noticed by the user due to noise during ignition of the gas burner.

Since the eTRV is a PID controller and due to other advanced features, the eTRV might request the boiler to start even if the comfort temperature is reached (e.g. heating to prevent over-cooling if the room temperature is decreasing rapidly… like braking with a car *before* hitting the actual obstacle).

This feature might not be understood and seem anti-intuitive to the user. The perceived behavior could be "the system is requesting heat when not needed".

In order to provide a more intuitive behavior the Heat request can be suppressed under the following conditions:

Ignore heat request from eTRV when all of the below is true

- RoomTemperature > RoomSetpoint (if possible, add some hysteresis on this to prevent frequent toggle).
- AdaptationRunStatus (0x404D) <> 1).

RoomTemperature could be the LocalTemperature (0x0000) measured at the eTRV or the one measured at the room sensor. Room setpoint could be OccupiedHeatingSetpoint (0x0012) at the eTRV or the room setpoint in room sensor application.

Please Note:

Inhibiting the Heat Request deprives the PID control of some authority in controlling the room's temperature, this can result in a less accurate temperature control and eventually reduce energy efficiency.

3. Optional Features

Needed for the eTRV to perform in Energy Efficiency and Comfort

3.1 Schedule

3.2 Adaptive Pre-Heating

3.3 Window Open

3.4 Min-Max Temperature setpoint limitation

3.5 Childlock (Keypad Lockout)

3.6 Identify Me

3.7 Changing Valve Exercise Settings

3.8 Changing Heating Control Scaling

3.9 OTA Update of eTRV FW

3.1 Schedule

The schedule functionality can be configured by the GW via the following interface.

The gateway need to refresh the schedule periodically to ensure that other changes than the routine weekly schedule are operated by the thermostat (e.g away or at home events).

At first join, the eTRV has no default schedule (and time is not synchronized/valid).

By checking the Time invalid attribute in the Zigbee Time cluster, the GW can verify if a power cycle event has occurred and need to refresh the Schedule in the eTRV.

Cluster	Command	Name	Function
0x0201	0x01	SetWeeklySchedule	Vacation day is not used, the schedule is set according to Zigbee Specifications (please refer to https://zigbeealliance.org/wp-content/uploads/2019/12/07-5123-06-zigbee-cluster-library-specification.pdf section 6) The events within one day must be ordered chronologically
0x0201	0x02	GetWeeklySchedule	Can be used to verify that the schedule is stored in the eTRV (the eTRV does not modify the schedule itself) **Note! The schedule information is lost after power cycle or OTA**
0x0201	0x03	ClearWeeklySchedule	Deletes all schedule events

Relevant Attributes:

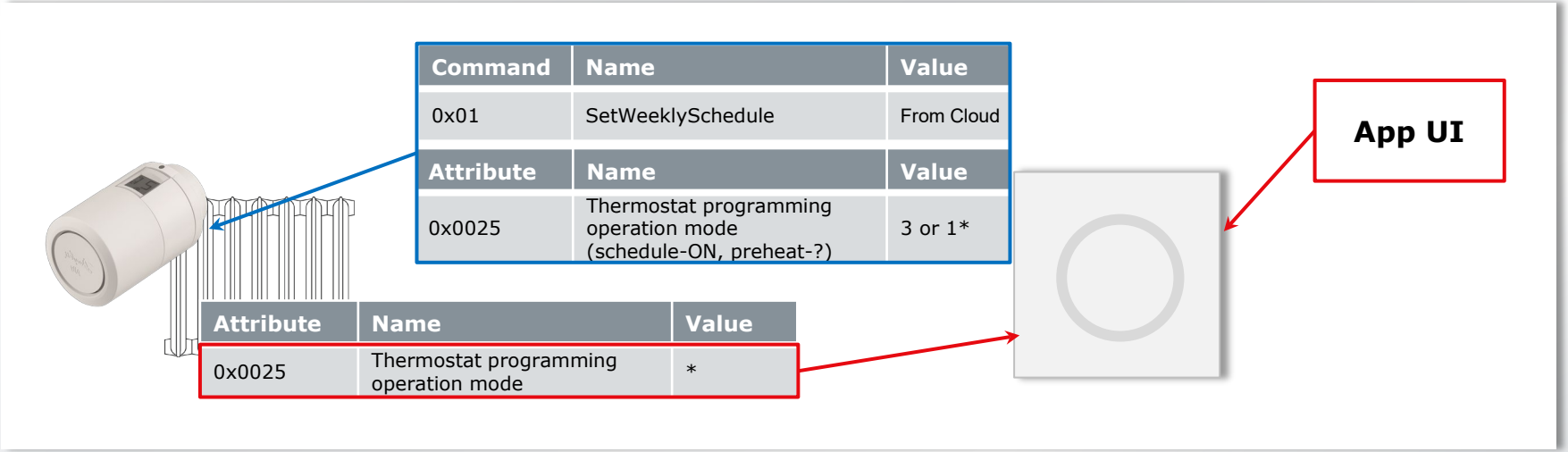
Cluster	Attribute	Name	Function
0x0201	0x0020	Start of Week	Has no impact for behavior, hardcoded, read only
0x0201	0x0021	Number of Weekly transitions	42 – hardcoded, read only
0x020	0x0022	Number of Daily transitions	6 – hardcoded, read only
0x0201	0x0025	Thermostat programming operation mode	Toggles the schedule function ▪ Bit0 toggles the schedule function (1= enabled; 0=manual, the eTRV will aim at the temperature in OccupiedHeatingSetpoint) ▪ Bit1 toggles the PreHeat function (1= enabled; 0=disabled)

3.1 Use case example – Changing to Schedule

The system should react as soon as possible to the change to satisfy the user's need.

Whenever the user has made edits to the Weekly Schedule page (and if schedule is enabled) the updated settings should be transmitted to the eTRV when the user leaves the Weekly Schedule page.

The communication below is repeated for all the eTRVs in the same room.



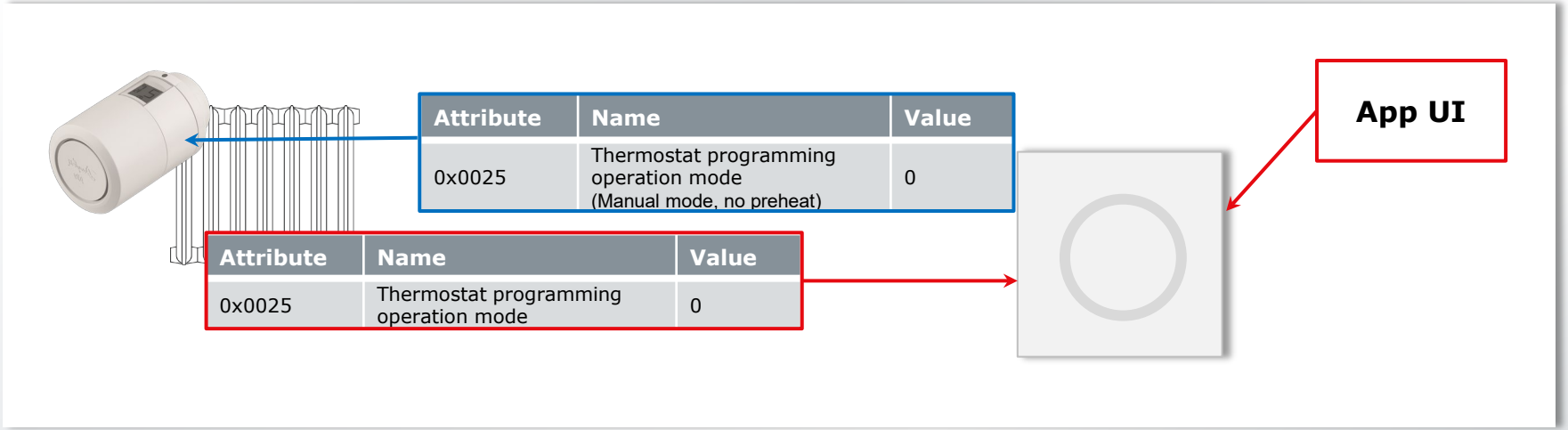
* The value depends on whether PreHeat Setting is TRUE or FALSE (see Use case – Preheat Control in App). The value of Bit0 (0-indexing) of 0x0025 is determined by the schedule/manual mode state.

3.1 Use case – Changing to Manual mode

The system should react as soon as possible to the change to satisfy the user's need.

The communication below is repeated for all the eTRVs in the same room.

The eTRV will aim at the temperature as in the current schedule period (last reported Occupied Heating setpoint).



Attribute	Name	Value
0x0025	Thermostat programming operation mode (Manual mode, no preheat)	0

Attribute	Name	Value
0x0025	Thermostat programming operation mode	0

3.1 Use case – Schedule handling and maintaining

- Midnight crossing
  - The schedule in the eTRV consists of up to 6 events per day, however, the UI consists of up to three at home periods each day. To ensure enough events for three at home periods and a consistent user experience it is necessary to remove 23:59 event if a new at home periods follows the next day at 00:00.
- Example:
  - Away setpoint = 17 degrees.
  - At Home setpoint = 21 degrees.

Monday	Tuesday	Wednesday
06:00 -> 21 08:00 -> 17 16:00 -> 21 23:59 -> 17	06:00 -> 21 08:00 -> 17 16:30 -> 21 17:30 -> 17 20:00 -> 21	08:00 -> 17 16:00 -> 21 22:30 -> 17

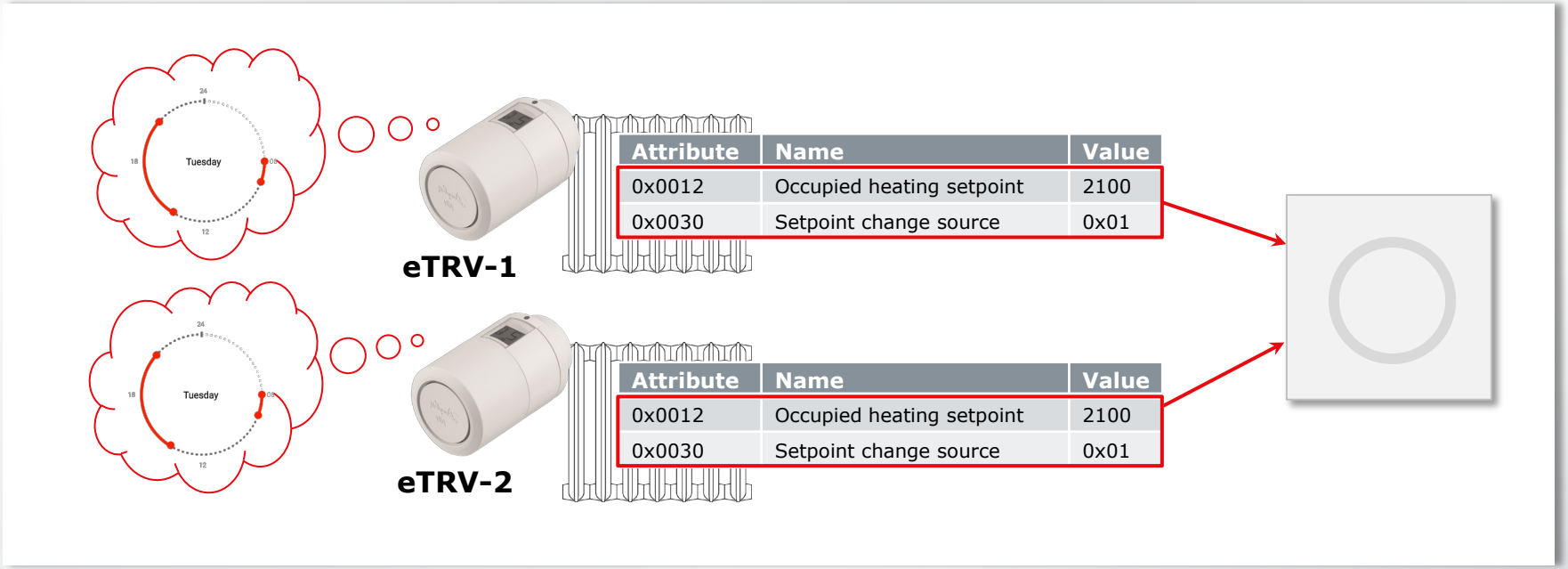
Note that if changes are performed to the schedule for Tuesday it may result in updates for Monday and Wednesday as well. E.g. if the start time for the first period Tuesday is changed to 00:00 this will result in removing the change at 23:59 Monday and not change should be performed Tuesday at 00:00.

Similar if the end time for the last period Tuesday is changed to 23:00 this will require a setpoint change Wednesday at 00:00.

Monday	Tuesday	Wednesday
06:00 -> 21 08:00 -> 17 16:00 -> 21 ~~23:59 -> 17~~	~~00:00 -> 21~~ 08:00 -> 17 16:30 -> 21 17:30 -> 17 20:00 -> 21 23:00 -> 17	00:00 -> 21 08:00 -> 17 16:00 -> 21 22:30 -> 17

3.1 Use case – Scheduled setpoint change at eTRV

When the eTRV runs its own schedule, it informs the GW that a scheduled setpoint change has occurred, GW/cloud changes the currently shown Temperature setpoint for the device and for the room.



3.2 Pre-Heat

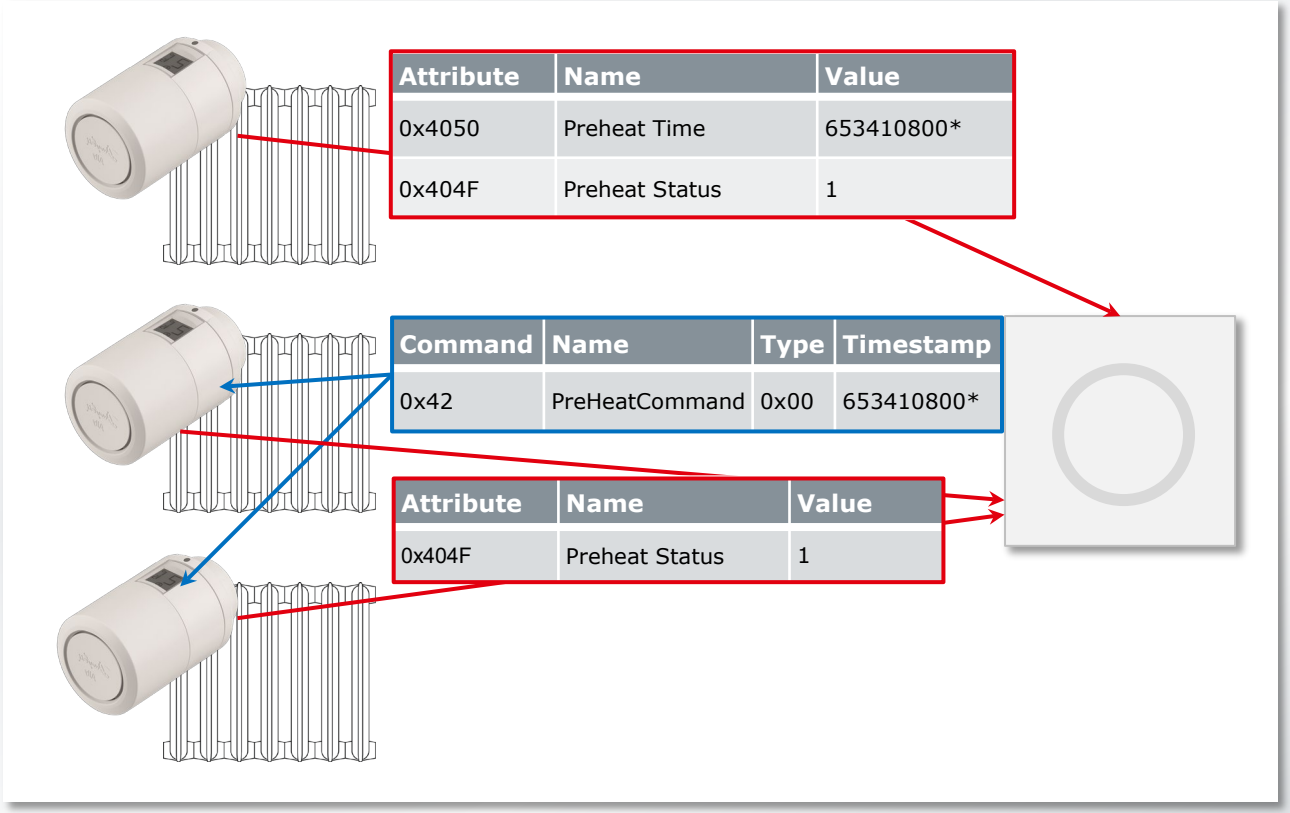
The Pre Heat functionality can be configured by the GW via the following interface. Furthermore, for rooms with multiple eTRVs there is a need for synchronizing Preheat such that all eTRVs provide heat at the correct time. Otherwise, it could happen that the radiator controlled by the eTRV in the coldest spot of the room will provide all the heat during preheat. See "Use case – Preheat coordination" for further details on the coordination and "Use case – Preheat status in App" and "Use case – Preheat control in App" for details on information used in the App.

Cluster	Command	Name	Function
0x0201	0x42	PreHeatCommand	Request eTRV to enter pre-heat if in schedule mode and if other eTRV in same room has triggered pre-heat. command needs two parameter enum8 = 0 = force preheat. Other values for future needs. Second parameter uint32 is timestamp received from other eTRV in the same room that went into preheat. This command is used for synchronization of Preheat in rooms with multiple eTRVs

Cluster	Attribute	Name	Function
0x0201	0x404F	Preheat Status	Preheat during schedule. 1=preheating. 0=no preheat is running. This should be used in the App to show preheat symbol.
0x0201	0x4050	Preheat Time	Time stamp of the scheduled setpoint which is currently being Preheat to. This is used to synchronize Preheat in rooms with multiple eTRVs.
0x0201	0x0025	Thermostat programming operation mode.	Bit 1 = 0: Preheat off, Bit 1 = 1: Preheat on.

3.2 Use case – Preheat coordination

To avoid one eTRV delivers all the heat during preheating it is necessary to coordinate preheat within a room. This is done by distributing the active preheat setpoint time between eTRVs, i.e., the eTRV will notify the setpoint time which it is currently preheating to. This information should be shared between eTRVs in the same room to force all eTRVs to preheat to the same setpoint.



* Same time stamp sent to all eTRVs in the Room.

3.3 Window Open

The Window Open functionality can be configured and operated by the GW via the following interface. The Window Open use cases will be explained in the following slides.

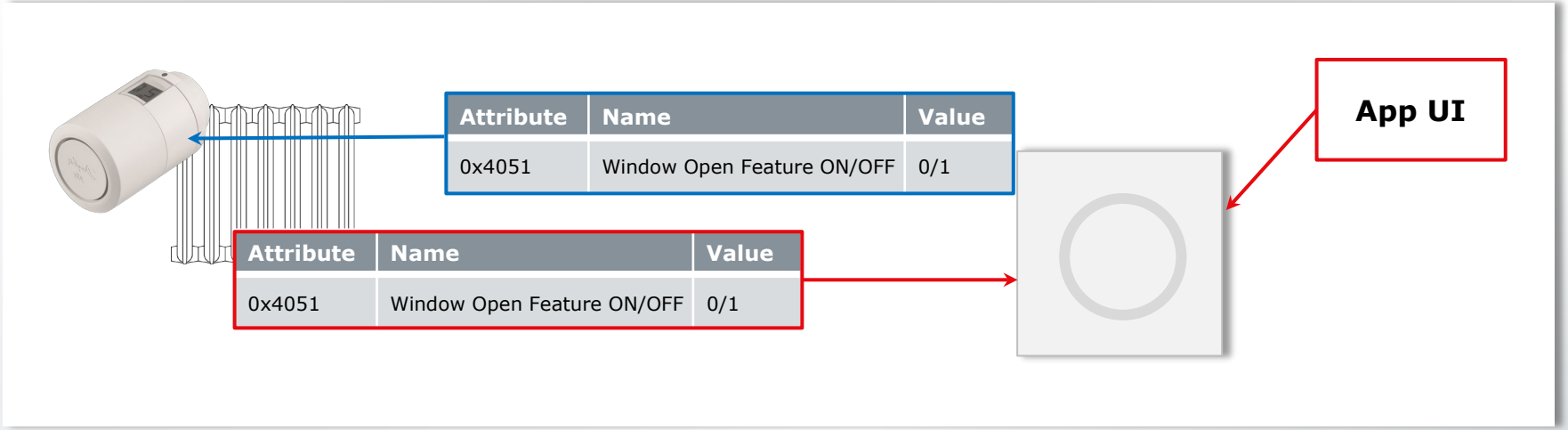
Cluster	Attribute	Name	Function
0x0201	0x4000	eTRV Open Window Detection	0x00: Quarantine 0x01: Windows are closed 0x02: Hold, Windows are maybe about to open 0x03: Open window detected 0x04: In window open state from external, but detected closed locally
0x0201	0x4003	External Open Window Detected	0x00: Windows are closed 0x01: Windows are opened
0x0201	0x4051	Window Open Feature ON/OFF	0x00: window open feature OFF. 0x01: window open feature ON.

3.3 Use case example – Window Open Control in App

Window Open Detection can be controlled by **"Window Open Feature ON/OFF"** (cluster 0x0201 Attribute 0x4051).

The feature is default "ON" in the eTRV.

The communication below is repeated for all the eTRVs in the same room.

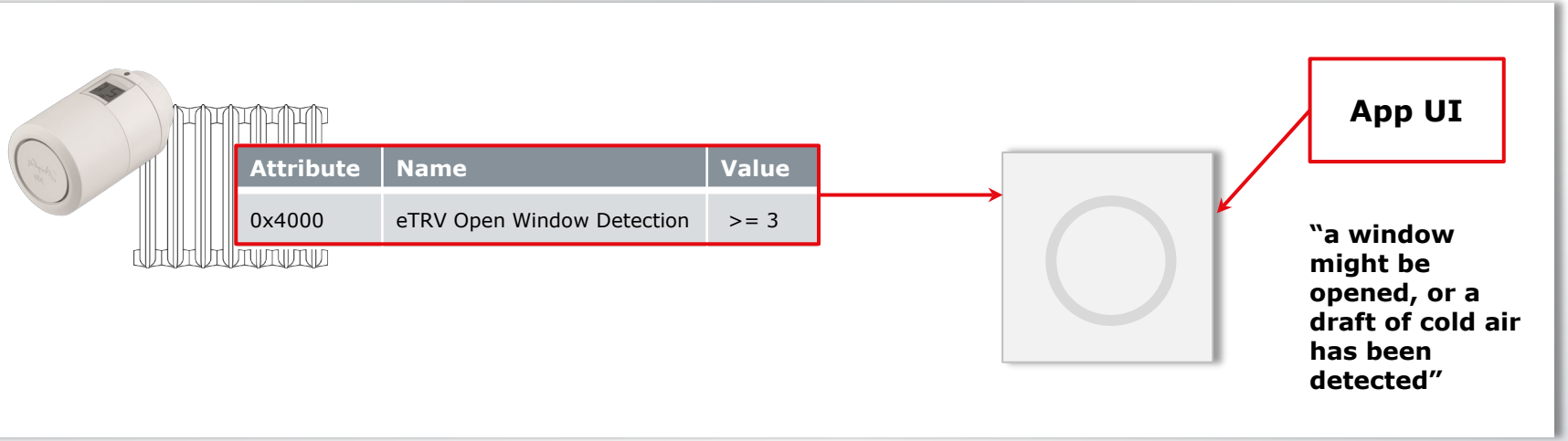


3.3 Use case example – Window Open Status

The Window Open status can be retrieved if needed in a remote UI via the attribute “eTRV Open Window Detection” (cluster 0x201 attribute 0x4000).

If the value is 3 or above the Window is considered opened by the eTRV (valve is closed).

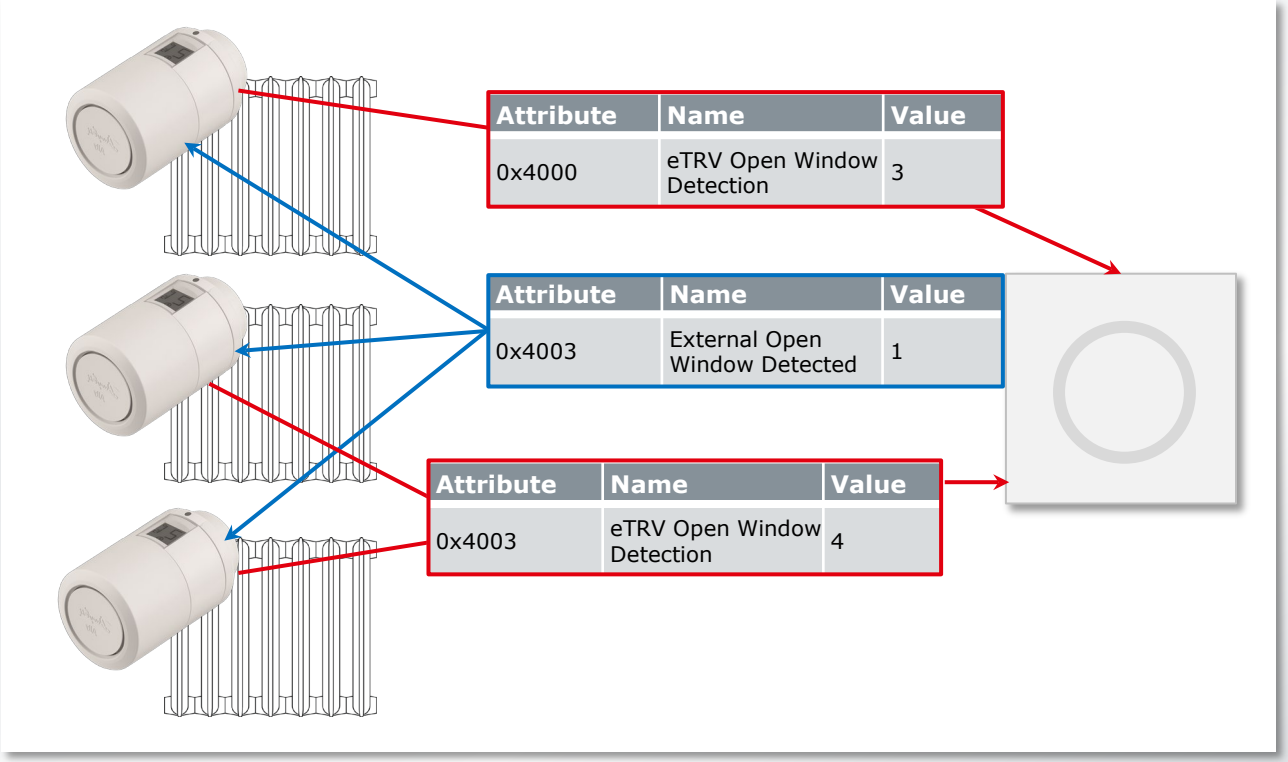
If the value is 2 or below the Window is considered closed by the eTRV (valve is opened).



3.3 Use case example – Window Open coordination (Activation)

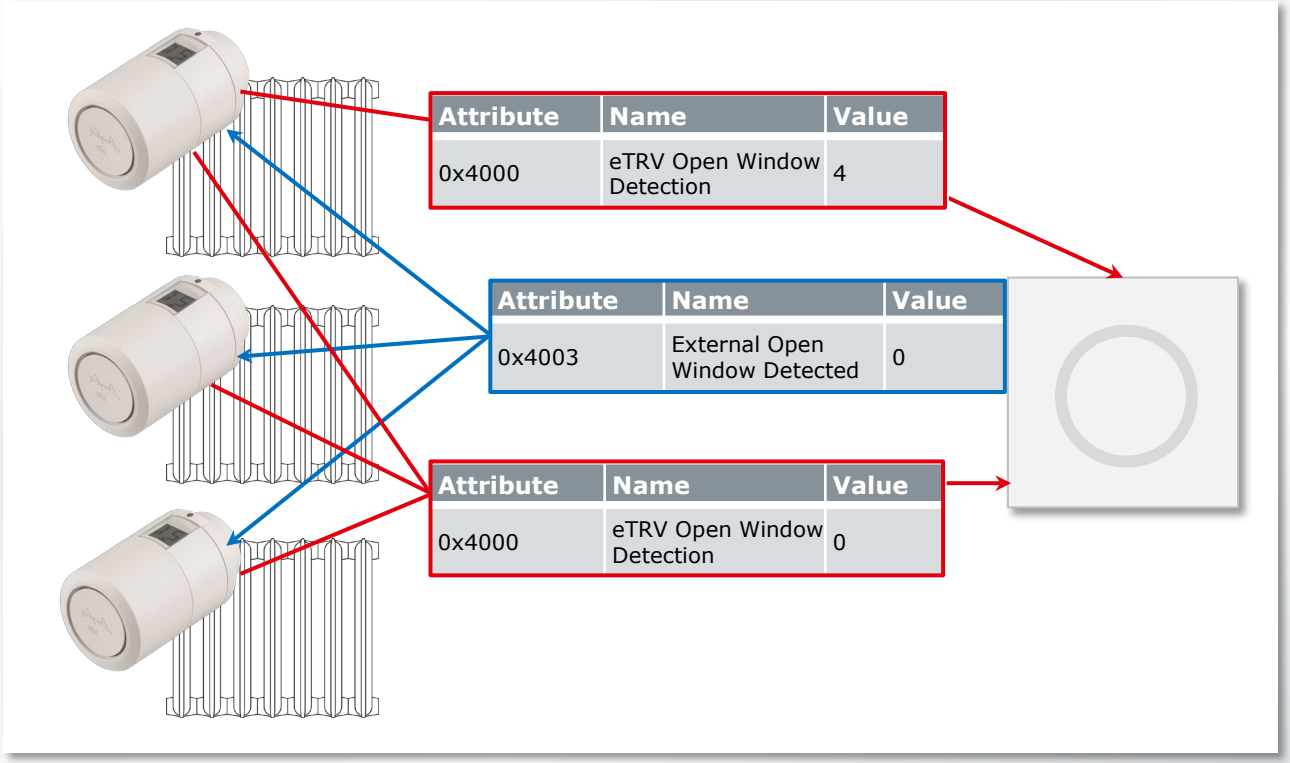
The GW has the possibility to coordinate and synchronize all the eTRVs in the same room or area so that they react simultaneously.

When the first eTRV (or any other device e.g. a window contact) detects a window opened event, the GW can force the other eTRVs in "external window open", the eTRVs will respond by changing their status to "4".



3.3 Use case example – Window Open coord. (Deactivation)

When all the eTRVs (including the one that detected the first time) that were forced to external open window change to "4" (In window open state from external, but detected closed locally) the GW can deactivate the External window opened.



3.4 Min.& Max settings

eTRV Min.& Max temperature setpoint setting attributes can be set by the GW in a range within absMin and absMax.

If the user tries to select a temperature outside the range via turning the dial, the display will blink.

If the OccupiedHeatingSetpoint is set to a temperature outside the range it will be rejected.

Cluster	Attribute	Name	Function
0x0201	0x0003	absMinHeatSetpointLimit	READ ONLY (absolute temperature limit = 5)
0x0201	0x0004	absMaxHeatSetpointLimit	READ ONLY (absolute temperature limit = 35)
0x0201	0x0015	MinHeatSetpointLimit	Range: 0x0003 absMinHeatSetpointLimit to 0x0016 MaxHeatSetpointLimit
0x0201	0x0016	MaxHeatSetpointLimit	Range: 0x0015 MinHeatSetpointLimit to 0x0004 absMaxHeatSetpointLimit

3.5 Childlock

The feature can be enabled via the following attribute and has only 1 functional level (the behavior will be the same for all settings 0x01 to 0x05):

Cluster	Attribute	Name	Function
0x0204	0x0001	KeypadLockout	Range: 0 to 5 0x00 = no lockout 0x01 to 0x05 = lockout (child lock)

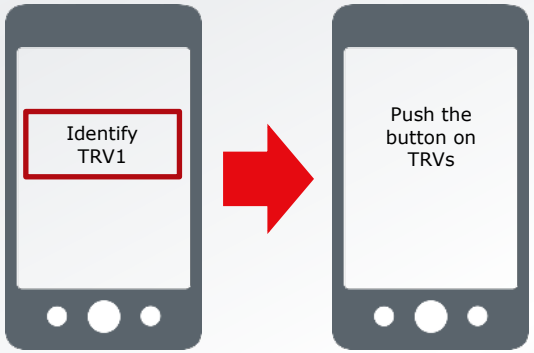
3.6 Zigbee Identify me feature

Description

- The GW sends the command "Identify yourself for n seconds!" to the TRV.
- The TRV starts a countdown timer where it stay in Identify status for n seconds. (radio icon blink + backlight steady on).

Since the eTRV is a battery driven device it only activates its LCD display every 60 second unless a user interaction occurs (turn dial or push a button) therefore the Gateway has the responsibility to send an identify request longer than 60-90 seconds, otherwise the device might never turn on for identification.

In order to speed up the identification the App could ask the user to push the button on the eTRVs, only the TRV which is under identification will remain steady on.



- During Identify UI no other UI is active.
- Heating control does not run during UI.

3.7 Changing Valve exercise settings

The Valve is exercised every week (Thursday at 11:00)

If another trigger time is needed it can be set via the following attributes:

Cluster	Attribute	Name	Function
0x0201	0x4010	Exercise day of the week	Range 0-7 0 = Sunday, 1 = Monday, … 6 = Saturday, 7 = undefined Default = 4 (Thursday)
0x0201	0x4011	Exercise trigger time	Range 0 to 1439 Minutes since midnight Default = 660 (11:00)

3.8 Changing Heating Control Scaling

The feature can be enabled via the following attribute and has only 1 functional level (the behavior will be the same for all settings 0x01 to 0x05):

Cluster	Attribute	Name	Function
0x0204	0x4020	Control algorithm scale factor	Range 1-10 Scale factor of setpoint filter timeconstant ("aggressiveness" of control algorithm) 1=5min(Quick) ... 5=30min(Moderate) ... 10=80min(Slow).

3.9 OTA update of eTRV FW

The Danfoss Ally eTRV support OTA via Zigbee. The OTA cluster is described in the Zigbee ZCL specification as example in Section 11.3:

https://zigbeealliance.org/wp-content/uploads/2019/12/07-5123-06-zigbee-cluster-library-specification.pdf

Example:

OTA file for the new 01.08 firmware: 1246-0100-01080108.0002_(DF4ECCE1).ota

What is required to know from a Zigbee gateway is given from within the OTA file itself but there are some details that might be needed to setup on the gateway:

Manufacturer ID:	0x1246
Image Type:	0x0100
Firmware Version:	0x00000108
Total Image Size:	501899 bytes

NOTE: SW Downgrade is not allowed, The TRV will refuse an OTA image that carries a Firmware version number smaller or equal than its own.

ENGINEERING
TOMORROW