CANopen object dictionary of an injector

This article describes the functionality and the possible relationships of the CANopen objects specified for injectors in CiA 425-2.

The object dictionary (OD) implemented on a CANopen-connected injector is a list of objects existing at run-time and accessible by a scanner via the CANopen network. Each object serves a specific purpose. The OD as a whole defines the injector's application functionality. It determines the injector's operation aspects e.g., features, capabilities, communication parameters, injection protocol parameters, and expected behaviors in case of communication loss. The scanner, which controls the injector, learns those aspects by reading the objects from the injector's OD. The scanner can also influence the injector's behavior by writing new values to the objects, if allowed by the injector. An object can provide a "read-only" (ro), "write-only" (wo), or "read-and-write" (rw) access. An injector may restrict the object's access permission based on its own control measures or safety requirements.

An object in the OD can be one of the two object types: simple and complex. A simple object (variable), contains a single data piece with the data-type (Boolean, Unsigned8, etc.) as specified in CiA 301. A complex object (array or record) contains multiple pieces of data. These pieces may have the same data type (array) or different data-types (record). Each object in the OD is addressed with a 16-bit index and an 8-bit sub-index (00_h for variables; 00_h to FE_h for arrays or records). Each object can be accessed by a scanner using the SDO (service data object) service.

As mentioned above, the OD exists only at the run-time. Once the injector starts up, its OD, with all objects populated with default values, becomes accessible to the scanner. As some of the objects (e.g. 1016_h, 6070_h) have invalid default values, the scanner has to configure them via SDO. When the injector shuts down, all object values (including the configured) are lost, as object 1010_h (store parameters) may not be supported by the injector. Every time the injector starts up (or resets), the scanner must re-configure certain objects. An electronic data sheet (EDS) is a file that lists all the objects (and their default values) supported by an injector. Using the injector's EDS file, the scanner knows which objects the injector supports, and which objects have to be configured.

The OD is divided into communication profile area (objects 1000_h to $1FFF_h$, e.g. heartbeat), manufacturer-specific profile area (2000_h to $5FFF_h$, not specified objects), and standardized profile area (6000_h to $67FF_h$). The standardized profile area objects in CiA 425-2 specify the common application functionality of an injector (see table 1). The objects of the category "conditional" may have to be implemented depending on the injector compliance class.

Table 1: Standardized profile area objects for injectors (Source: CiA 425-2)

Index	Object Name	Object/Data Type	Category*
Device Identity and Profile Version Objects			
6070_h	Scanner identity	RECORD	M
6073_h	Profile Version	UNSIGNED32	M
Injector FSA-related Objects			
6000_h	Control word	UNSIGNED32	M
6001_h	Status word	UNSIGNED32	M
6006_h	Communication lost	UNSIGNED8	M
Injector Function Objects			
6007_h	Functions supported	UNSIGNED32	M
6008_h	Global attributes support	ARRAY	M
Injector Capability Objects			
6002_h	Injection mode	UNSIGNED8	M
$600D_h$	Maximum configurable volume	ARRAY	O
$601A_h$	Maximum configurable pressure	UNSIGNED16	O
6028_h	Maximum configurable total flow rate	ARRAY	O
6050_h	Configured piston content	ARRAY	C
6051_h	Injector capabilities	ARRAY	C
Injection Protocol Objects			
6005_h	Examination delay	ARRAY	C
6019_h	Configured pressure limit	UNSIGNED16	C
6020_h	Configured phase type	ARRAY	C
6024_h	Configured total flow rate	ARRAY	C
6025_h	Configured total volume	ARRAY	C
6027_h	Configured delay duration	ARRAY	O
6031_h – 6038_h	Configured piston ratio 1 – Configured piston ratio 8	ARRAY	C
$603E_h$	Configuration check command	UNSIGNED8	*reserved*
$603F_h$	Configuration error list	ARRAY	C
Dynamic Injection Objects			
6009_h	Current injected total volume	UNSIGNED16	C
$600A_h$	Current pressure	UNSIGNED16	C
$600B_h$	Current total flow rate	UNSIGNED16	C
$600C_h$	Current volume remaining	ARRAY	C
Achieved Injection Objects			
6021_h	Achieved average total flow rate	ARRAY	*reserved*
6022_h	Achieved total volume	ARRAY	C
6023_h	Achieved duration	ARRAY	C
6029_h	Achieved peak flow rate	ARRAY	C
6039_h	Start phase timestamp	ARRAY	O
$603A_h$	End phase timestamp	ARRAY	C
Unit and Increment Definition Objects			
6041_h	Time unit	UNSIGNED32	C
6042_h	Flow rate unit	RECORD	C
6043_h	Pressure limit unit	RECORD	C
6044_h	Volume unit	RECORD	C
6045_h	Piston ratio unit	ARRAY	O
$604A_h$	Display increment unit	ARRAY	O

* M – mandatory; O – optional; C – conditional

Device identity and profile version

The scanner identity object (6070_h) includes the scanner's CANopen vendor-ID, product code, revision number, and serial number. After the injector starts up or resets, the scanner has to identify itself to the injector by providing this object with appropriate data. Otherwise, the injector will not communicate with the scanner for security reasons. The injector's identity (same object type as 6070_h) is specified in object 1018_h. Upon the injector's startup, the scanner can read this object to determine whether it should provide its own identity to the injector.

Object 6073_h specifies the CiA 425-1 and CiA 425-2 versions supported by the injector. A scanner reads this object during configuration to interpret the injector objects correctly, as their definitions could have been changed between versions (e.g. 6028_h).

Injector FSA-related objects

Object 6000_h contains the scanner command, which the injector receives via RPDO 1 (receive process data object). Among others, it causes a transition from the current injector FSA (finite state automaton) state to a new state. On a successful state transition, the injector stores a status word (with the new state) in object 6001_h and transmits the object value to the scanner via TPDO 1 (transmit PDO). If the injector fails to execute the state transition, it sends an emergency message. Then, it sends a copy of the status word still containing the current state. Object 6006_h specifies whether (and which) state transition an injector will perform in case of communication loss during an injection. The scanner reads this object during configuration, so it knows what to expect when communication loss occurs.

Injector function objects

Object 6007_h provides the functions supported by an injector. This 32-bit object currently defines 7 bit (see table 2).

Table 2: Bit field specification of 6007_h (Source: CiA 425-2)

Bit		Description
0	rearm	Whether remote arming (from scanner) is supported
1	*reserved*	
2	ecg	Whether ECG (electrocardiogram) is supported
3	*reserved*	
4	skip	Whether "skip forward to phase" command is supported
5	hc	Whether injector state of hold configuration is supported
6	rc	Whether injector state of ready configuration is supported
7	drb[1]	How a delay phase should be handled after hold state
8	mix[2]	Whether mixed flow is supported
9-31	*reserved*	

Note:

1. Bit 7 indicates how an injector handles a remaining delay phase after being put on hold. Assuming that the injector is currently executing a delay phase with a duration of 15 s. When the duration counts down to 5 s, the operator decides to push the hold button (state transition from procedure executing to hold). After a while (not defined how long), the operator ends the hold by pushing the hold-resume button (state transition from hold to procedure executing). What should happen to the remaining 5 s of the delay phase? If bit 7 is set to 0, the injector will terminate the delay phase immediately, and move on to the next phase (if any). If bit 7 is 1, the injector will resume counting down the remaining 5 s.
2. Dual-flow is a special and the most common case of the mixed flow, where two syringes are active in an injection phase.

Some injector functions (global attributes) can be controlled remotely by the scanner via the "set global attributes" command (see 6000_h).

Object 6008_h is an array with two 16-bit sub-indices (see table 3). Sub-index 1 indicates which global attributes the injector has implemented (bit = 0), meaning that they can be controlled locally by the injector. Since these are ▷

the injector's implementation details, sub-index 1 has “ro” as access. Sub-index 2 indicates which global attributes (from those implemented by the injector) are further granted permission to be also controlled remotely by the scanner (bit = 1). However, there exist disagreement among the SIG (special interest group) contrast media injector members about which device should grant the permission for the remote control. If it is the injector, sub-index 2 should have the “ro” access type. But if it is the scanner, sub-index 2 should be “rw”. In the current version of CiA 425-2, sub-index 2 is “rw”. The author thinks that sub-index 2 should be “ro”. In other words, the injector should decide which global attributes can be remotely controlled. For example, an injector may support an XDS (extravasation detection system) as an accessory (sub-index 1: bit 3 = 0 i.e. implemented), but it may not allow the scanner to remotely activate it due to the injector's own safety requirements (sub-index 2: bit 3 = 0 i.e. not supported). In such cases, the injector will not delegate the permission to the scanner.

Table 3: Bit field specification of 6008_h sub-indices (Source: CiA 425-2)

Bit	6000_h Global Attribute Bit	
0-2	*reserved*	
3	11	Activate/stop XDS
4	12	Start/stop KVO (keep veins open)
5	13	Lock/unlock local Arm button
6	14	Lock/unlock local Start button
7	15	Lock/unlock local Hold-Resume button
8	16	Lock/unlock local configuration
9	17	Transition 8* automatic/commanded
10	18	Provide phase number/type
11	22	Provide examination delay status
12	23	Provide injector reconfiguration status
13	19	Provide configuration check status
14-15	*reserved*	

* Injection completed to procedure completed

Injector capability objects

Object 6002_h indicates what kind of injection an injector can perform, such as CT, CV, or MR injection. In addition, bit 0 indicates whether remote arming is allowed, which requires that the injector supports remote arming (6007_h: bit 0 = 1, see table 2). Interestingly, this object has the “rw” access permission in the idle state and in any of the configuration states. This implies that a scanner is potentially able to configure this object so that, for example, a CT injector can be turned into an MR injector. This is unlikely to happen. So, this object is usually implemented as “ro” in all injector states.

Object $600D_h$ indicates the maximum configurable volume of an injection protocol to use for each fluid type (see 6050_h below). This is the total capacity of all syringes filled with the same fluid type. Each sub-index of this array represents one syringe (or piston) type. Object $601A_h$ indicates the maximum allowed pressure for the syringes installed on the injector. It applies to all injection phases.

Originally, object 6028_h indicated the maximum configurable total flow rate for each injection protocol phase. For a dual-flow phase, it was the total of the two involved flow rates. Since CiA 425 version 2.2.0 the meaning of this object has changed. Now it means the maximum flow rate for each syringe (or piston). Therefore, the word "total" in the object's name is no longer appropriate.

Object 6050_h indicates the fluid type (e.g. contrast media, saline) in each syringe (represented by a sub-index). The highest bit of each sub-index value also indicates whether that syringe is in service. Object 6051_h (array) indicates the maximum number of possible phases (sub-index 1), the maximum number of injection phases (sub-index 2) and the maximum number of delay phases (sub-index 3, see also 6020_h).

Injection protocol objects

The injection protocol configuration objects can only be configured (written to) when the injector is in one of the configuration states. Object 6005_h specifies an examination delay (sub-index 1), which must be within the minimum value (sub-index 2) and the maximum value (sub-index 3), or 0 (no delay). When an injector enters the procedure executing state, the injection starts. At the same time the examination delay countdown starts as well. In this sense, the examination delay (also called scan delay) has no direct impact on the ongoing injection. However, the injector cannot enter the procedure completed state until the examination delay has counted down to 0.

Object 6019_h specifies the pressure limit for the current injection protocol, which must be less than or equal to the value of object $601A_h$ (maximum configurable pressure). If the pressure during an injection exceeds the value of 6019_h, the injection may be aborted by the injector.

Object 6020_h specifies the four phase types for each injection protocol phase: injection (1), delay (2), wait (3), and test bolus (4). The delay phase has a configured duration. The wait phase is infinite until it is terminated by a local action or a remote command that is not part of the injection protocol.

Object 6024_h specifies the total flow rate for each injection protocol phase. For a dual-flow phase, it is the total of the both involved flow rates. The actual flow rate allocated to each fluid is based on the fluid ratio (see 6031_h to 6038_h). Object 6025_h specifies the total volume for each injection protocol phase. For a dual-flow phase, it is the total of the two volumes of both involved fluids. The actual volume allocated to each fluid is based on the fluid ratio (see 6031_h to 6038_h). Object 6027_h specifies the delay duration for each injection protocol phase. This is meaningful only for a delay phase (see 6020_h), otherwise it is always 0. Objects 6031_h to 6038_h specify the fluid ratio in syringes 1 to 8 for each injection protocol phase. For a single flow phase, a ratio of 100 (in %) must be set for the corresponding syringe. For a dual (or more) flow phase, the total ratio of the two (or more) syringes must be 100 (%). In this case, the specific ratio for each syringe is controlled by object 6045_h (see unit and increment definition objects).

Previously with object $603E_h$, the injector received (by SDO) the configuration check command from the scanner. If the object value is 1, the injector first clears the configuration errors in object $603F_h$ (see below), sets sub-index 0 of $603F_h$ to 0, and executes the configuration check. If ▷

the check succeeds, sub-index 0 of $603F_h$ is set to 1, and sub-index 1 is set to 0 (no error). But if the check fails, sub-index 0 of $603F_h$ is set to the number of errors, and the subsequent sub-indices are filled with the appropriate error codes (e.g. volume too high). If the value of object $603E_h$ is 0, only the configuration errors are cleared, no configuration check itself takes place. Since the scanner sends this command via SDO, there is no mechanism for the injector to inform the scanner about the completed configuration check. To find out the check status, the scanner needs to periodically read object $603F_h$, sub-index 0 until it gets a non-zero value. Because of this inefficiency, object $603E_h$ has been deprecated since CiA 425-2 version 2.2.0. Instead, a new "execute configuration check" command has been added to the control word 6000_h. Since the control word is received via RPDO 1, the injector must transmit a status word via TPDO 1 with the result of the configuration check. This is much more efficient.

Object $603F_h$ contains a list of pre-defined configuration check error codes that have resulted from the last configuration check command. Before the first configuration check or after a successful one, sub-index 0 is 1, and sub-index 1 is 0 (no error). This is one of the few cases where the value of sub-index 0 is not equal to the highest sub-index number of the object.

Dynamic injection objects

“Dynamic” means here that the object values are periodically updated by the injector during an injection. The object values are also periodically sent to the scanner via TPDO 2 to TPDO 4 with a rate determined by the scanner. Object 6009_h provides the total volume (from all active syringes) delivered so far during the current injection. Object $600A_h$ provides the pressure that is being measured during the current injection. Object $600B_h$ provides the total measured flow rate during the current injection. If the current phase is a dual-flow phase, it is the total of the two flow rates. Values of 6009_h, $600A_h$, and $600B_h$ are sent to the scanner via TPDO 2.

Object $600C_h$ provides volumes that are remaining in all active syringes. This array represents a syringe in each sub-index. Values of this object are sent to the scanner via TPDO 3 (sub-indices 1 to 4) and TPDO 4 (sub-indices 5 to 8). If an injector has four or less syringes, TPDO 4 is disabled.

Achieved injection objects

The actually achieved values for each injection phase are placed into these objects right after the phase is completed. The scanner is then notified via the injector status word and can therefore read those values if required. The achieved values are kept until the injector enters the system ready state for the next injection.

Object 6021_h holds the average total flow rate for each injection phase. For a dual-flow phase, it is the total of the two flow rates. Since it is an average value, the actual phase duration (see 6023_h) will impact it. For example, if a phase is put on hold for a period of time, the phase duration will become longer. However, CiA 425-2 is silent on ▷

whether the hold time should be included in calculating the average flow rate for the phase. If the hold time is included, the average value could be well below the configured value, which could confuse the end users. For this reason, CiA 425-2 version 2.2.0 has added object 6029_h to replace 6021_h. Object 6029_h provides the peak (highest) flow rate for each injection phase, which should be very close (if not equal) to the phase's configured flow rate value.

Object 6022_h holds the total volume for each injection phase. For a dual-flow phase, it is the total of both fluid volumes. Object 6023_h provides the actual injection duration for each phase. Objects 6039_h and $603A_h$ hold the start and end timestamp for each injection phase, respectively. The difference between the two values should match the actual injection phase duration (6023_h).

Unit and increment definition objects

These objects define simple and complex units for the injection parameters. The latter include the unit (sub-index 1), the low limit (sub-index 2) and the high limit (sub-index 3) constraining the unit's value range. A unit is an unsigned32 value with the structure specified in the object 0080_h, sub-index 1 (see figure 1).

numerator	denominator	prefix	00_h
MSB			LSB

Figure 1: Unit value structure (Source: CiA 425-2)

The unit representation is interpreted as: prefix * numerator/denominator. For units without a denominator (e.g. time), the denominator byte is 00_h. The full list of codes for the numerator, denominator, and prefix is specified in CiA 303-2[3]. If the LSB (least significant byte) has a non-zero value, the 4-byte unsigned value is interpreted as an increment (see figure 2). An increment specifies a unit's resolution used for input or output purposes. In this case, the unit value is interpreted as: increment * prefix * numerator/denominator.

numerator	denominator	prefix	increment
MSB			LSB

Figure 2: Unit value structure with an increment (Source: CiA 425-2)

Object 6041_h defines the time unit used by objects representing time (e.g. 6005_h, 6027_h). Objects 6042_h, 6043_h and 6044_h define the complex units for flow rate, pressure limit, and volume, respectively. These units are used by injection protocol, dynamic injection, and achieved injection objects. The chosen low and high limits must be wide enough to represent all possible values for the corresponding parameters.

Object 6045_h defines the minimum piston ratio increment and its limits. It applies only to dual-flow (or mixed flow) injection phases. For such phases, each value of objects 6031_h to 6038_h must be a multiple of the ratio increment (sub-index 1) and within the defined range (between sub-index 2 and sub-index 3). Furthermore, the total of the mixed-flow values involved in a phase must be 100 (%).

For values of time, volume, pressure, piston ratio and flow rate that are displayed on the injector's user interface, object $604A_h$ defines the increments for these parameters. The difference between sub-index 4 of $604A_h$ (piston ratio) and sub-index 1 of 6045_h (scalable increment) is that the former is used for output and the latter for input (configuration) purposes. ◀

References

[1] CiA 301: CANopen application layer and communication profile, version 4.2.0, 2011

[2] CiA 425-2: CANopen application profile for medical diagnostic add-on modules, Part 2: Injector, version 2.3.4, 2017

[3] CiA 303-2: CANopen recommendation, Part 2: Representation of SI units and prefixes, version 1.5.0, 2012

Author

Ron Kong
Bayer US LLC
ron.kong@bayer.com
www.bayer.us