-

R2CP.CAN RADIOLOGICAL ROOM CONTROL PROTOCOL

Software Protocol Specification

CONFIDENTIAL

Author: **Andujar, J.M.**  **Ingeniero De Sistemas Embebidos** **Date: 20/05/2021**	**Reviewed:** **Andujar, J.M.**  **Ingeniero De Sistemas Embebidos** **Date: 20/05/2021**	**Approved:** **Varo, Antonio**  **Team Leader Sistemas Embebidos** **Date: 20/05/2021**

⚠ *Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.*

Digitally signed by
Date: 2021.05.20
10:32:54 +02:00
Reason: Grupo Sedecal
Location: Algete - Madrid

Table of Updates

Document Version	Document Revision	Date	Changes / Remarks	Affected Sections	Author
0.0	A	14/03/2018	IS1632CA Rev. B document imported into EPDM as this release.	All	F. Sanchez
0.0	B	03/04/2018	Removed length field for pipe data. Extended description for *PACKAGE_VERSION* format.	All	F. Sanchez
0.0.	C	10/09/2018	Added Serial Number Set Added Interlock SubIndex Added New Node Status Added Warning SubIndex	6. Status Register 10. appendix 1	J. M. Andujar
1.0	D	19/11/2018	Updated IIS0023SFUN R2CP.CAN Version	2. Related Document	J. M. Andujar
1.0.	E	20/11/2018	Serial number string limits. Added document version.	10,11 Table of Updates	A.J.Varo
1.0.	F	17/01/2020	Added Compilation Date SubIndex Added Appendix 3: Compilation Date Serial number string limits. Added document version.	10,11,12 Table of Updates	J.M.Andujar
1.0.	G	13/10/2020	Added Set Protocol Version	10	J.M.Andujar
1.0	H	07/05/2021	Added System Versions Get/Answer	10	J.M.Andujar

CONFIDENTIAL

⚠ Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.

Table of Contents

CONFIDENTIAL

⚠ Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.

TABLES OF FIGURES

CONFIDENTIAL

⚠ Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.

1 Introduction

R2CP is a communication protocol and device profile specification for embedded systems. In terms of the OSI model, R2CP implements the layers above and including the network layer. R2CP consists of an addressing scheme, several small communication protocols and an application layer defined by a device profile. The communication protocol has support for network management, device monitoring and communication between one Master and one Node, including a simple transport layer for message segmentation/desegmentation of data blocks. The lower level protocol implementing the data link and physical layers is usually Controller Area Network (CAN), although devices using some other means of communication (such as serial port, Ethernet) could also implement the R2CP device profile.

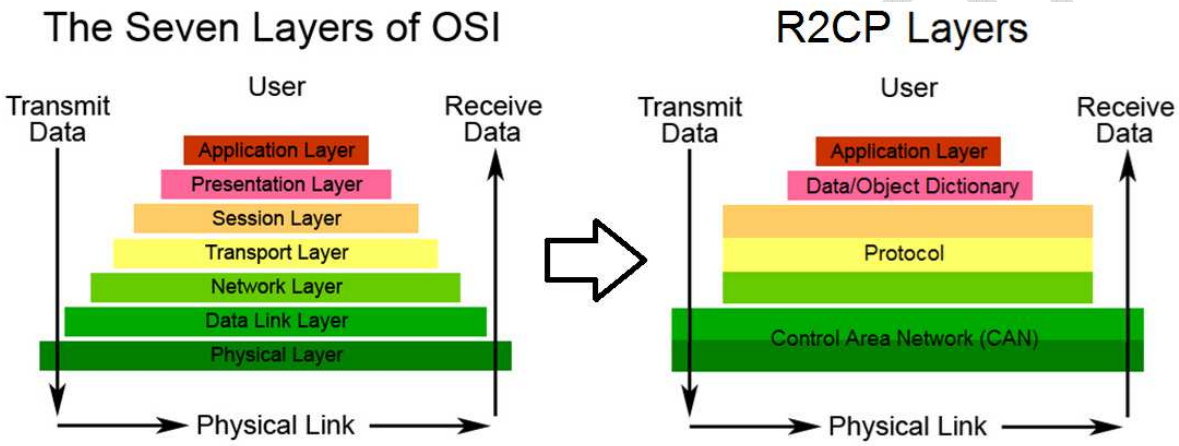


Figure 1 - R2CP Layers

Every R2CP user has to implement certain standard features in its controlling software.

- A communication unit implements the protocols for messaging with Master or Node in the network. Communication from Master can be developed to one or several nodes, but communication between nodes isn't allowed by protocol.

- Each node must assure transmission order. That means, a new message must wait to be sent for all pending messages.

- Each node must assure reception order. That means, a new message must be process according to reception order.

⚠ Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.

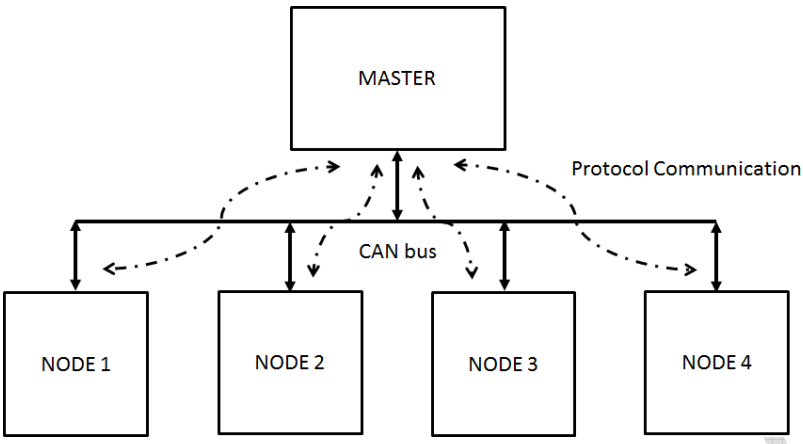


Figure 2 - R2CP Architecture

- Device status is deal via a state machine. It must contain the states Initialization, Ready, and also flags of Error, Booting reason and Heartbeat running. Transitions between states happen when device is ready to decode R2CP messages.

- The object dictionary is a table of variables and actions, addressable by an 8-bit Index and 8-bit Subindex. The variables and actions can be used to configure the protocol, board/node, and IOs, devices or/and peripherals connected to the board/node and reflect its environment, i.e. contain measurement data.

- The application part of the device actually performs the desired function of the device. The application is ordered by variables/actions through the object dictionary, and data are sent and received through the communication layer.

2 Related Documents

This revision of the document meets requirements of next documents and their versions specified bellow.

Document	Version
IIS0023SFUN R2CP.CAN Radiological Room Control Protocol Bootloader User Guide - Appendix B - Target IDs.docx	V1R0 Rev B

Figure 3 – Related Documents

⚠ Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.

CONFIDENTIAL

3 Object Dictionary

Every R2CP user must have an object dictionary, which is used for configuration and non-realtime communication with objects supported by the user. Each entry implemented in the node through its object dictionary is defined by:

- Index, the 8-bit address of the object in the dictionary.
- SubIndex, the 8-bit address of the parameter of the object in the dictionary.
- Access type, which gives information on the access rights for this entry, this can be read/write (getting/setting parameters), read-only (just getting information) or write-only (just setting information).
- Object dictionary table also describe the format of data and their meaning.
- Besides, it shows information about default values of parameters of the object.
- Additional information about the working of the object can be found on graphics way.

List of indexes reserved can be seen in next specification:

IIS0018SPRO R2CP.CAN Radiological Room Control Protocol - Appendix A - Indexes.docx

Object with Index=0 is defined for the main application and protocol use. This object gives access to Hardware and Software Versions, *6 Status Register, 7 Error Management, 8 Reset,* and software update implemented in the firmware.

For more details see *APPENDIX 1: R2CP Common Dictionary Object*.

CONFIDENTIAL

⚠ Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.

4 CAN Configuration

R2CP nodes must configure its CAN interface to receive and send message with next parameters.

CAN Configuration	
Baudrate	500Kbit/sec
Frame version	CAN 2.0B (Extended Frame)

Figure 4 - CAN Configuration

5 Datagram

Every R2CP message is based on the same datagram format/struct. The frame of a R2CP message is composed of a 29 bit Id, an 8 bit data length field and a maximum of 8 bytes of data which are sorted as **big endian system** (first byte is most significant byte, MSB).

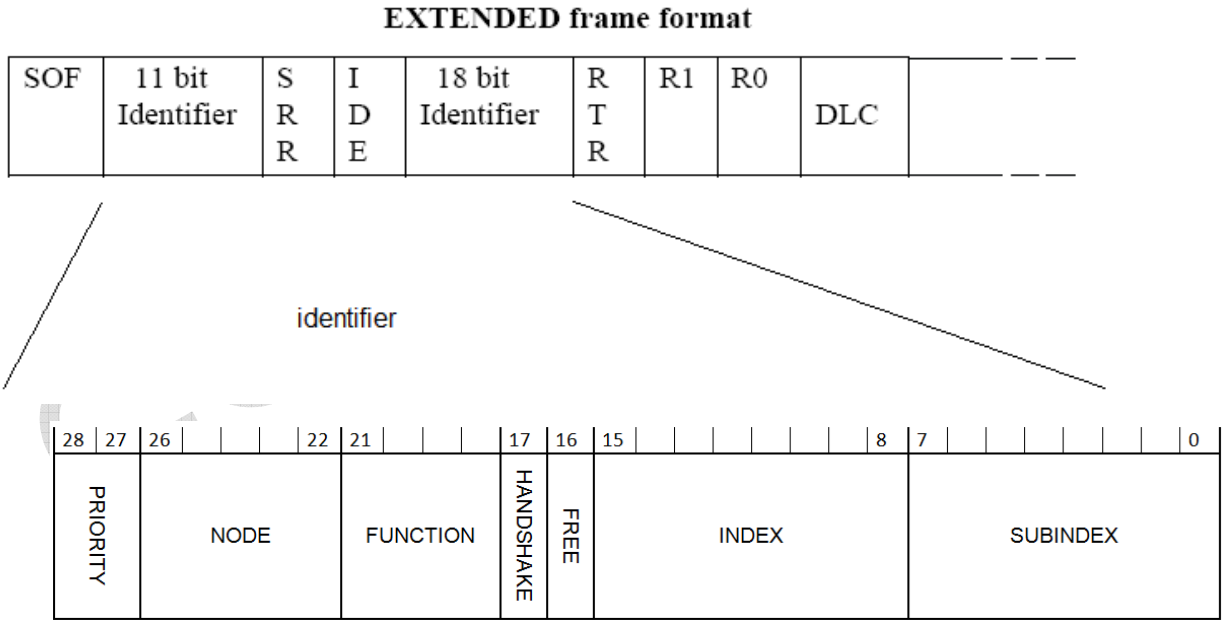


Figure 5 - R2CP Protocol ID

⚠ Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.

The format and description of the ID is as follows.

5.1 Priority

Two bits for four different levels of priority. Value 0 for the highest priority.

5.2 Node

Five bits where it's possible to read the addressee of the message, if the message is sent by Master, or origin node of the message, when message is sent from a node towards Master.

As Node field is composed of 5 bits, this makes support until 32 possible nodes in the same network.

Value 0 is reserved for broadcast use, and it will just filled by Master, when Master wishes to send a message to every node on the R2CP network.

List of Node IDs reserved can be seen in next specification:

IIS0019SPRO R2CP.CAN Radiological Room Control Protocol - Appendix B – Node IDs.docx

5.3 Handshake

It is used to request a confirmation as the message was received. If that's the case, the receptor will reply exactly the same message but clearing Handshake bit.

Next diagram shows an example of the handshake working for each function (functions supported are described later).

CONFIDENTIAL

⚠ Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.

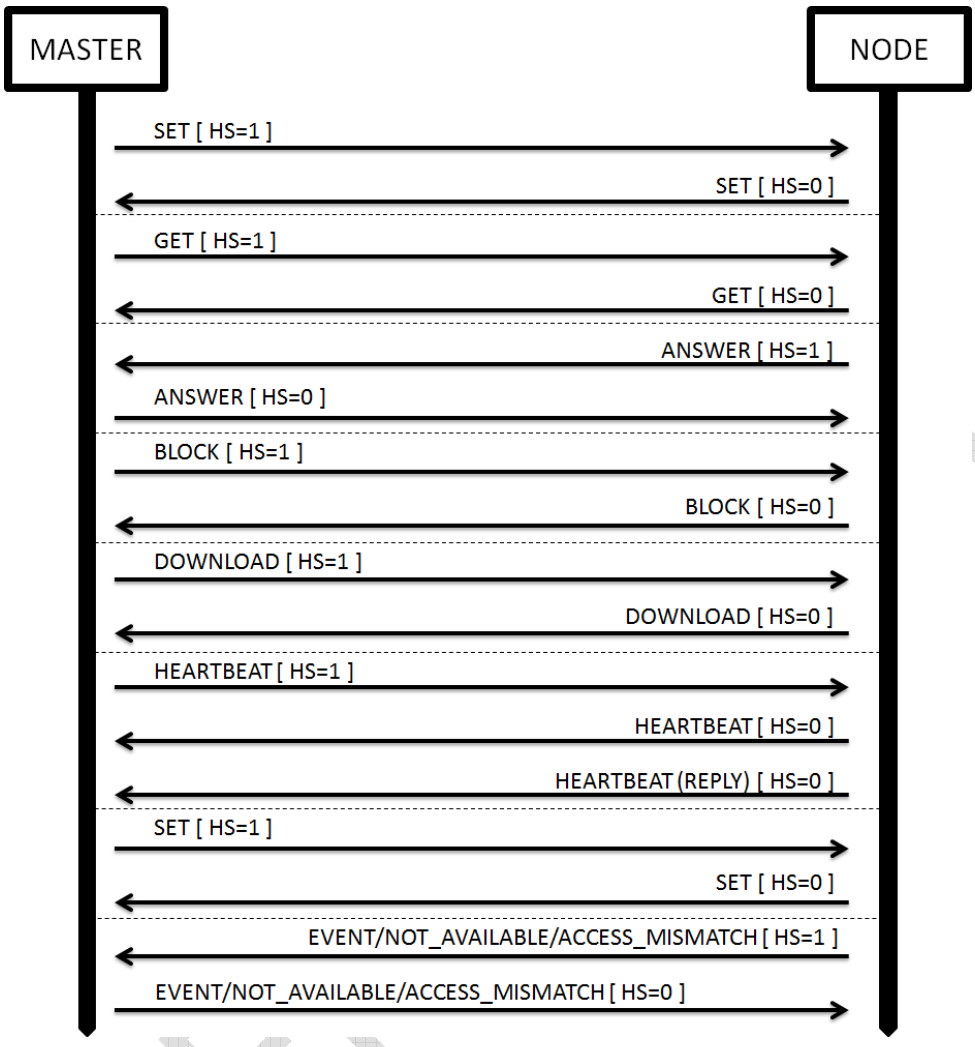


Figure 6 - Handshake Sequence Diagram

5.4 Function

Four bits to describe the meaning/functionality of the message. Pay attention that **value 0 is reserved for bootloader protocol**.

Next points enum supported ones at this protocol version.

5.4.1 SET

When function field values 1. It sets an action/parameter on data dictionary.

5.4.2 GET

When function field values 2. It gets an action configured/parameter from data dictionary.

⚠ Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.

5.4.3 ANSWER

When function field values 3. It's used to send an answer requested by a previous *GET* function.

5.4.4 EVENT

When function field values 4. Sending event information about a parameter/action configured on data dictionary.

5.4.5 BLOCK

When function field values 5. It's used to transmit a data block longer than 8 bytes. It's not necessary to implement if dictionary object of the node doesn't define a *BLOCK* function parameter.

Blocks are cut into several messages, which can carry up to 8 bytes maximum. Order and description of the messages which composed a block are as follows:

Message N	Data Bytes	Description
1st	Data[0]	0xFE. First message of the block.
	Data[1:2]	Data length of the block.
	Data[3]	Block function or type, same values as **Function** field of R2CP id.
	Data[4:7]	Reserved.
2nd to N - 1	Data[0]	Sequence number of the block, beginning from 0 until as much as necessary messages to carry every data. Maximum sequence number = 254. Sequence number from 0 to round up of data length / 7.
	Data[1:7]	Data.
N	Data[0]	0xFF. Last message of the block. Block completed, no data bytes pending to send.
	Data[1:7]	Reserved.

Figure 7 - Block Description

Next diagram shows an example of the sequence of messages to send a data block. As diagram shows, first byte of the data of the message, is the one which indicates if next data

⚠ Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.

CONFIDENTIAL

bytes carry information of the length of the data block, data of the block, or end of the data block.

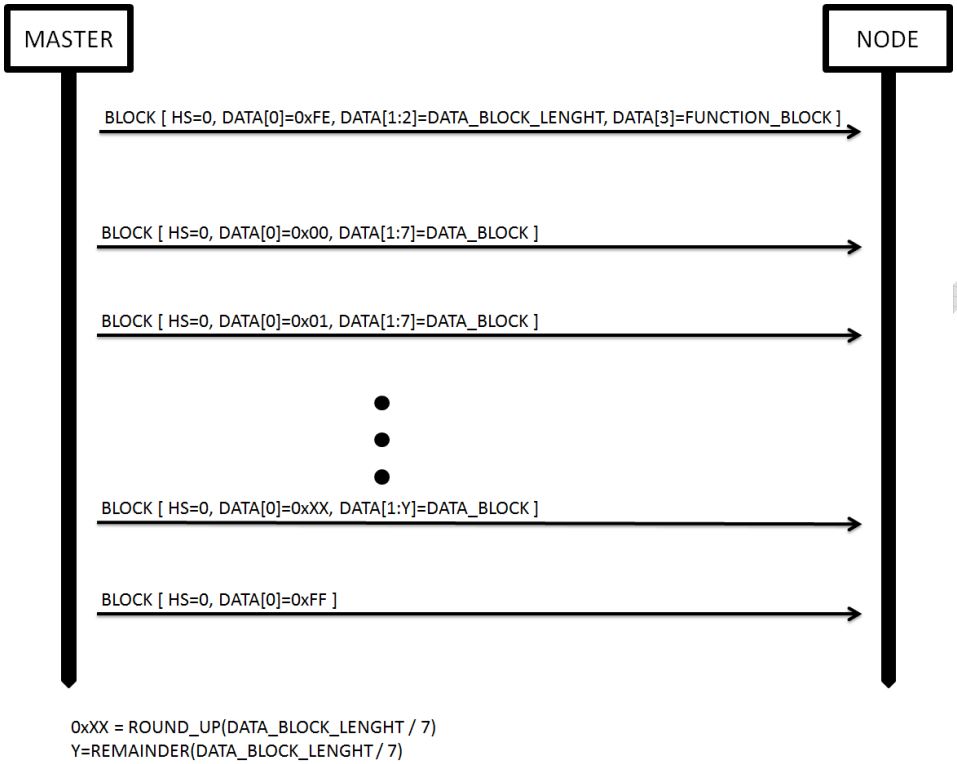


Figure 8 - Block Sequence Diagram

5.4.6 NOT_AVAILABLE

When function field values 6. It means that current R2CP User board doesn't support last command.

5.4.7 ACCESS_MISMATCH

When function field values 7. It indicates wrong kind of access/function in data dictionary.

5.4.8 HEARTBEAT/STATUS

When function field values 8. It is used to monitor status of the node, and it's also a mechanism for the node to realize whether Master continues alive or not.

For monitor use, it's not necessary to configure anything, just send the command and wait for the answer. In contrast, for the mechanism for checking if Master continues alive or not, it must be configured (heartbeat period) by one of the objects in dictionary (Index=0, SubIndex=5). Heartbeat flag inside *STATUS REGISTER (see point 5)* would be set then, indicating the mechanism is running.

⚠ Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.

If heartbeat timer expires, the node must notify error and transit to a safety state. Heartbeat mechanism will be necessary to configure again.

Next sequence diagram shows how heartbeat is working.

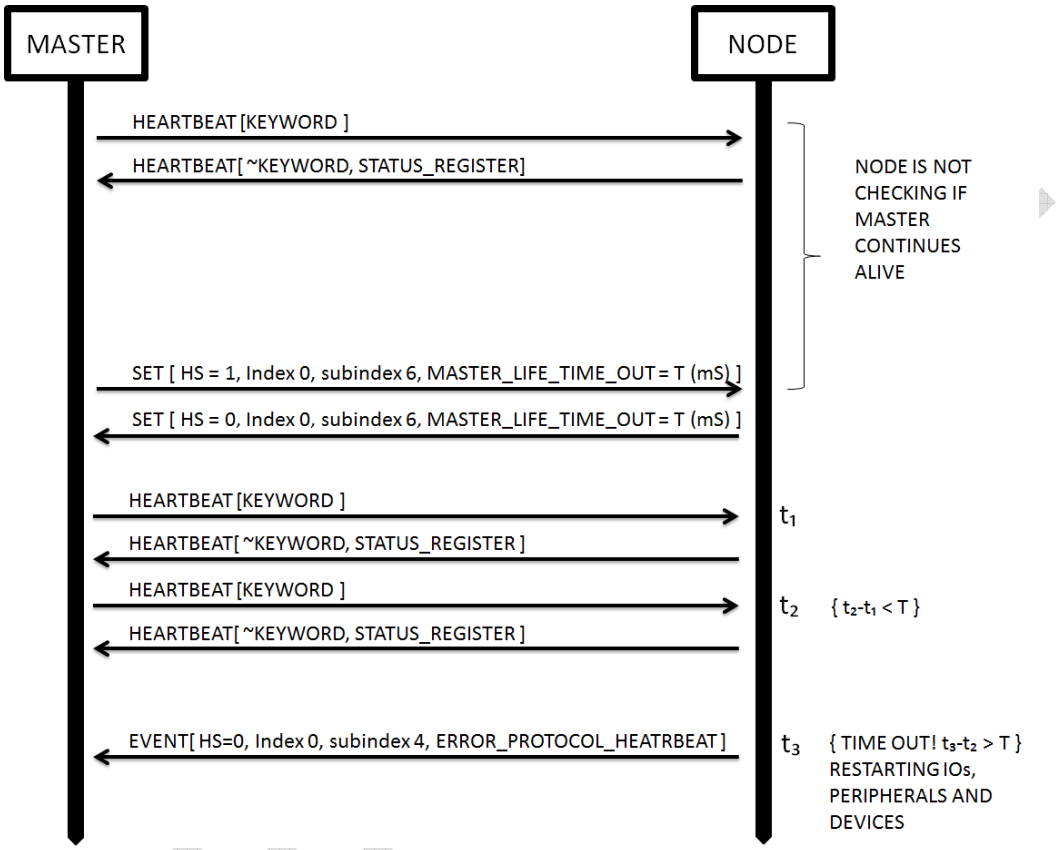


Figure 9 - Heartbeat Sequence Diagram

After heartbeat has been configured (*SET* command), the node realizes at t3 time Master doesn´t continue alive, so the node proceeds to indicate the error and configures itself on a safety state.

As it's possible to notice in *Figure 9 - Heartbeat Sequence Diagram*, this function changes the meaning of the datagram fields. When function field of the datagram is *HEARTBEAT*, Index and SubIndex fields have next meanings:

- Index. It becomes a key word. This keyword will be toggled for the reply when the node replies Master.

- SubIndex. It becomes node status (Index=0x0, SubIndex=0x2 in the object dictionary) just for the reply from the node.

⚠ Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.

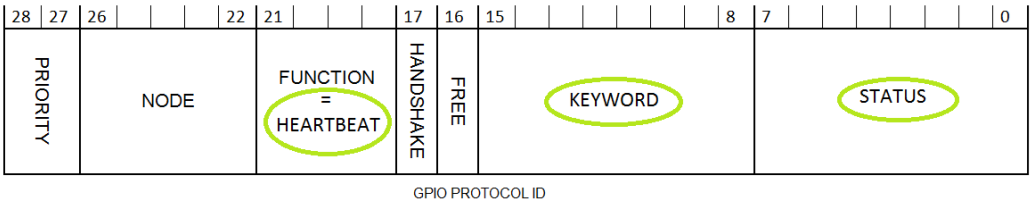


Figure 10 - R2CP Protocol ID for HEARTBEAT function

5.4.9 DOWNLOAD

When function field values 9. It's used to start a firmware download to upgrade the software. How to download the firmware and burn it will depend on each platform/application and its bootloader, but **that protocol must use FUNCTION field of the CAN bus Identifier with value of 0.**

When function field of the datagram is *DOWNLOAD*, Index and SubIndex fields become Target ID of the application project. Application, on reception, must use Target ID to check if binary to download fits its platform/project.

The Target Id is extracted from the content of the binary.

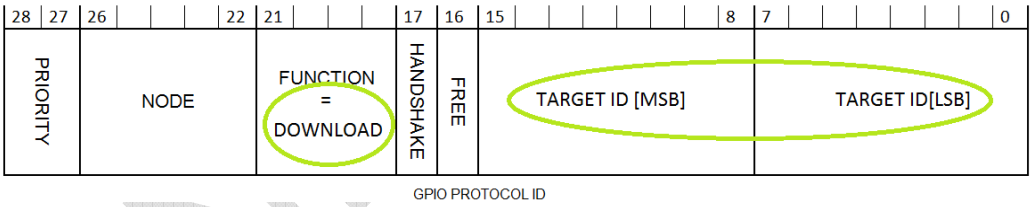


Figure 11 - R2CP Protocol ID for DOWNLOAD function

List of Target IDs already defined can be checked in IIS0023SFUN R2CP.CAN Radiological Room Control Protocol Bootloader User Guide - Appendix A - Target IDs.docx

Additionally, application updated also must check during every initialization if a software update was done, comparing version saved in non-volatile memory and its application version. In case it is different, an event will be sent indicating it by object dictionary Index=0, SubIndex=1.

5.4.9.1 Subnets

⚠ Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.

When, due to the topology of the architecture, the *DOWNLOAD* function needs to reach targets without direct access of communication (serve as an example next picture, where computer needs to update node 4), because there is one or several targets between the origin and destiny of the download (node 1 and node 3), then, each of these targets must work as bridges, receiving *DOWNLOAD* messages, and re-send them into each node connected to it, process until the target destiny of the message is reached.

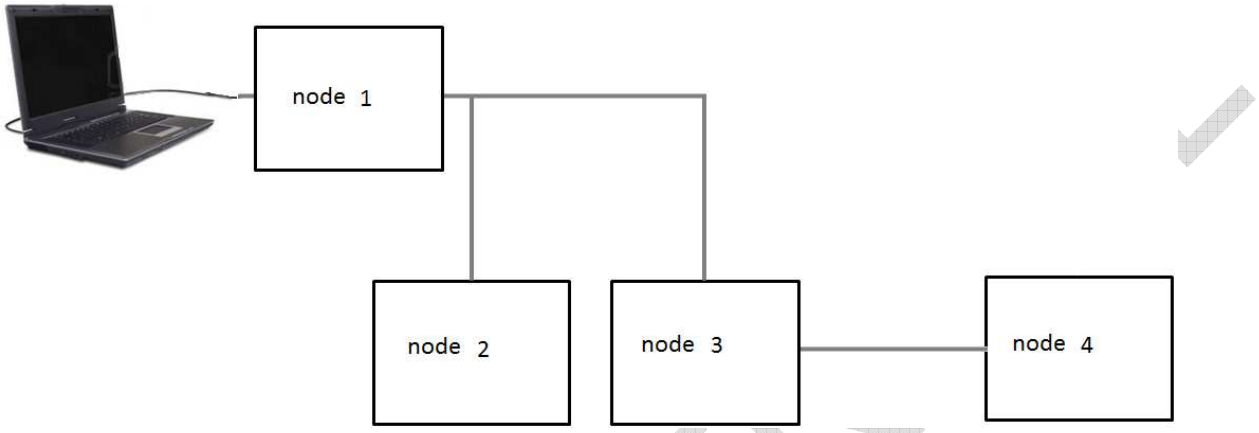


Figure 12 – Download through subnets

For these kinds of topologies, *DOWNLOAD* messages can carry data, where is detailed next node destiny of the message, until reaching the node to update.

Each data byte carries *NODE ID* of the next sub node in next subnet. First data byte must be node destiny of the message. Bridge must remove each data byte that is used to resend the message. Next figure shows example.

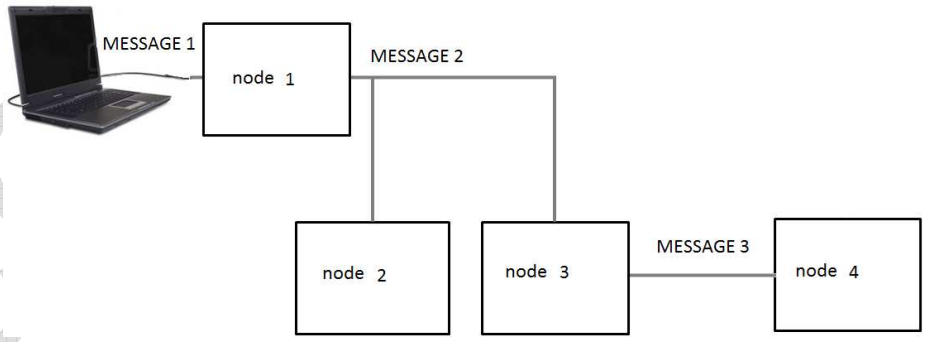


Figure 13 – Download example through subnets

Where Message 1 has next values:

⚠ Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.

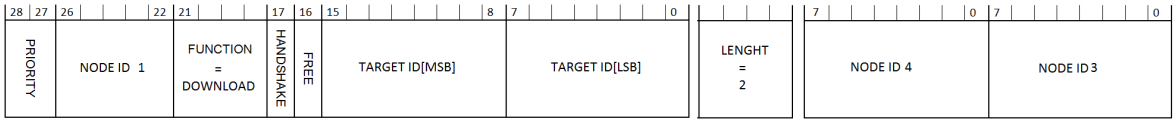


Figure 14 – Download example, Message 1

Message 2 after first bridge (node 1):

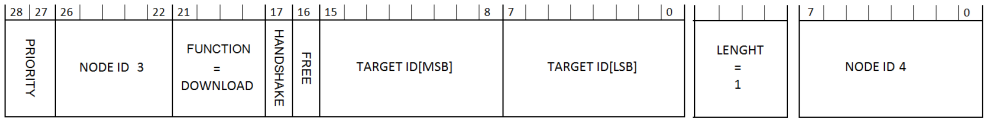


Figure 15 – Download example, Message 2

And Message 3 after second bridge (node 3):

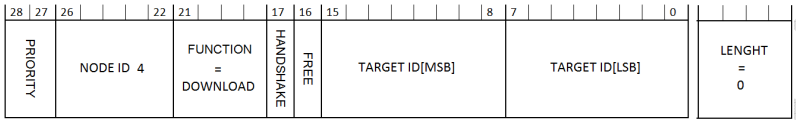


Figure 16 – Download example, Message 3

CONFIDENTIAL

⚠ Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.

5.4.10 MSG_PROCESSED

When function field values 10. It's used as a confirmation a message was processed. It's an optional message, so it will be defined for entries in dictionary objects. Then, it's not necessary to implement if dictionary object of the node doesn't define a *MSG_PROCESSED* function parameter.

This confirmation will not be sent until message to confirm took effect.

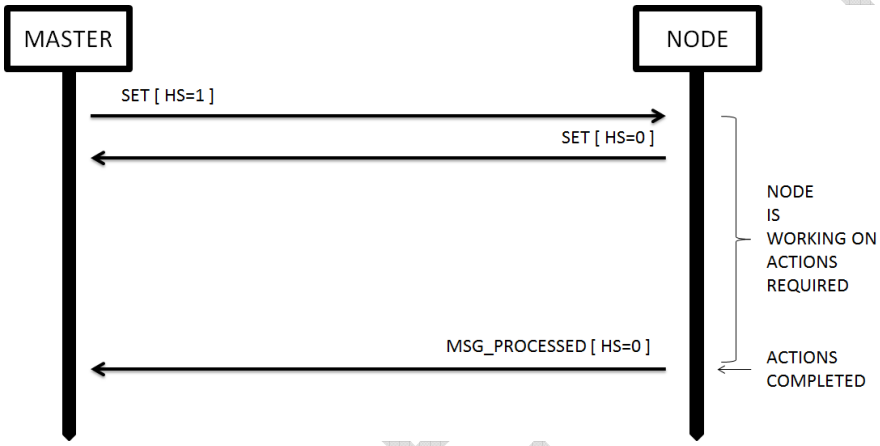


Figure 17 - Message Processed Sequence Diagram

5.5 Free.

One bit reserved for a future use.

5.6 Index

Eight bits address of the object in the dictionary.

5.7 Subindex

Eight bits address of the parameter/action of the object indicated by Index field in the dictionary.

CONFIDENTIAL

⚠ Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.

6 Status Register

The protocol has a register of 8 bits which indicates the status of the application, and several flags which indicate current working condition. Next table shows the status register format:

Bit	Field	Description
7 - MSB	Reserved	No used. Read as 0.
6	Boot	Boot reason: 0-Power on 1-Restarted
5	Error	Error flag. 0-No errors, or error queue was emptied. 1-At least an error happened. Check Error parameter.
4	HeartBeat	HeartBeat flag 0-HeartBeat has not been configured. 1-Have been configured
3	Reserved = 0	Reserved
[2:1]	Working Mode	Three different working modes supported at the moment**, each node must specify how the state machine between modes is**. -0x0: Normal operation. -0x1: Safety mode. The node will enter into this mode when determined conditions were found, **which must be specified for the node**. They are conditions that warn about a potential danger for the safety of the equipment/user. Some entries of the dictionary of the node may not be available. -0x2: Service mode. The node will enter into this mode when user requires not caring about a master, and working with different behavior, **which must be specified for the node**. 0x3: Interlock Mode: The node will enter into this mode when an interlock condition was found, **which must be specified for the node.** In case of an error happens during this mode, the node will enter in safety mode as long as it was needed. Some entries of the dictionary of the node may not be available.
0 - LSB	Status	Two different states supported at the moment: -0: Initializing. Initializing drivers and configurations. Application boots with this state. -1: Ready. When application is ready to decode R2CP messages.

Figure 18 - Status Register

CONFIDENTIAL

⚠ Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.

Application status can be always read from (Index=0, Subindex=2, see *APPENDIX 1: R2CP Common Dictionary Object* for more details) object dictionary, and it is also returned by heartbeat command.

After *READY* status updating with successful *INITIALIZATION*, the R2CP user can work on the object dictionary without problems.

7 Error Management

There is an error management included in the object dictionary (Index=0, SubIndex=4, see *APPENDIX 1: R2CP Common Dictionary Object* for more details), and an error code list defined for the specific node.

Application handles a volatile queue of errors, where they are stored. They can also be extracted, removing from the queue, one by one, when Master requests using a *SET* command of the same object. Pay attention that it's a queue, so it would extract first error notified and stored.

Besides, when any error is detected, Master is notified with an *EVENT* command of the same object, attaching the code which describes the error, and a field inside *6 STATUS REGISTER* is set indicating the new condition.

8 Reset

There are two different kinds of resets to implement for the application:

- By dictionary object. R2CP offers the option of resetting the board using a dictionary object (Index =0, SubIndex =3, see *APPENDIX 1: R2CP Common Dictionary Object* for more details).

- By watchdog. Application should implement a watchdog to prevent/guard the execution from uncontrolled errors.

Additionally, application can be restarted due to a software update, due to HW Reset Signal and other unexpected sources. A field inside status register has been added to indicate Master application node was restarted for any reason.

9 R2CP Node Requirements

⚠ Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.

CONFIDENTIAL

A R2CP Node must support the first index/object specified for any R2CP node, see details on *APPENDIX 1: R2CP Common Dictionary Object*.

Node id specified for each R2CP node must keep compatibility also with the rest of the nodes which will be connected to the same CAN bus. See list of Node IDs defined on IIS0019SPRO R2CP.CAN Radiological Room Control Protocol - Appendix B – Node IDs.docx

Next figure shows a short example about how the node (regarding R2CP actions) could start up and decode R2CP messages according to the specification mentioned. Pay special attention to node initialization events (status, software version updated, errors).

CONFIDENTIAL

⚠ Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.

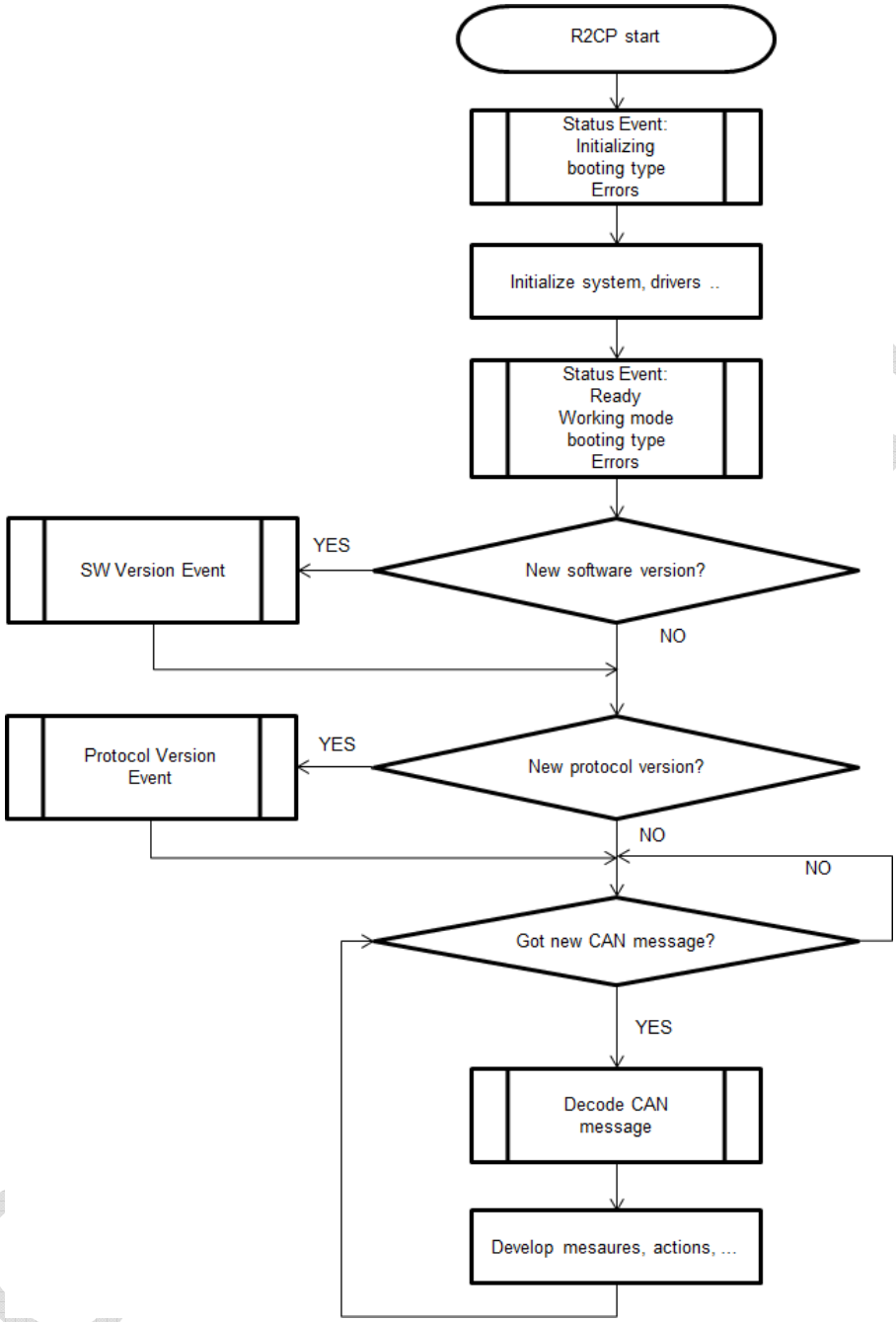


Figure 19 - R2CP Start Flow Chart Example

⚠ Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.

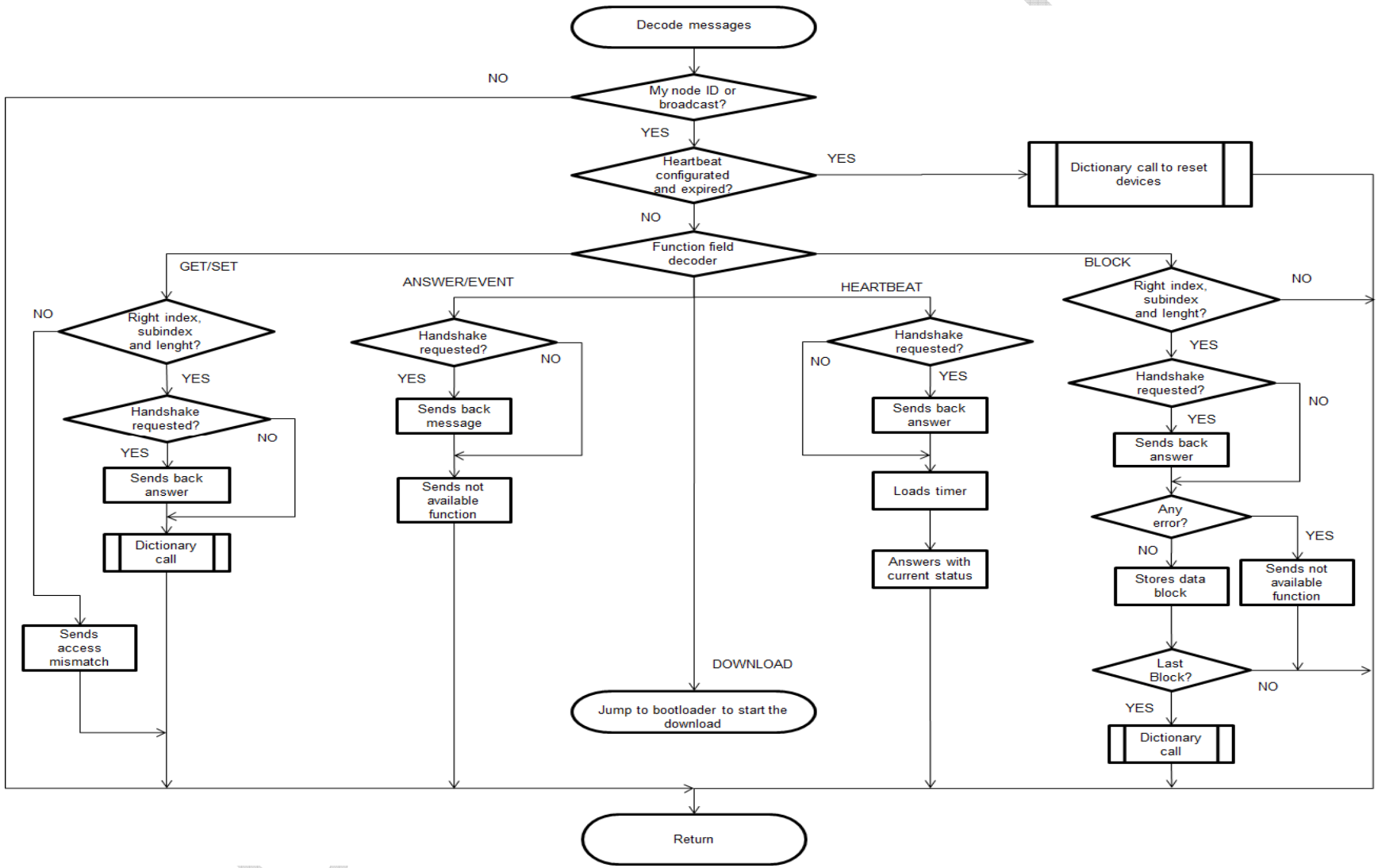


Figure 20 – R2CP Messages Parser Flow Chart Example

⚠ Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.

10 APPENDIX 1: R2CP Common Dictionary Object

Index	Sub Index	Device	Access Type	Functions	Format		Description	Default values	GRAPHICS
					Dlc	Data			
0x00	R2CP Common								
	0x00	HW VERSION	RD	GET	0		Request hardware version		
				ANSWER	4	MM MM VV RR	Format: AMMMM-VV-R *M*=Model. 2 bytes, decimal format *V*=Version. 1 byte, decimal format *R*=Revision. 1 byte, ascii format Ex: 0x0E 0x20 0x01 0x41 → A3616-01-A		
	0x01	SW VERSION	RD	GET	0		Request software version		
				ANSWER	3	VV RR SS	Format: vVrRR.S *V*=Version. 1byte, decimal format. It should be increased when there are big incompatibilities with previous version such as different hardware. *R*=Review. 1 byte, decimal format. It should be increased when there are big modifications in software. *S*=SubReview. 1 byte, decimal format. It should be increased when there is any modification in software. Ex: 0x01 0x0A 0x03 → V1R10.3		
				EVENT	3	VV RR SS	Software update done. Format: vVVrRR.SS		
	0x02	STATUS	WR/RD	GET	0		Request status of the node.	*S*=0	

CONFIDENTIAL

⚠ *Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.*

				ANSWER	1	Binary format: XBEHRWWS	S=Status of the node. 1bit. 0: Initializing board and drivers. 1: Ready to work as R2CP node. WW=Current working mode of the node. 2bits. 0: Normal operation. 1: Safety mode. 2: Service mode. 3: Interlock mode. R=Reserved. 1 bit. Value is set to 0x0. H=Heartbeat flag. 1 bit. 0: Heartbeat not running. 1: Heartbeat running. E= Error flag. 1 bit. 0: No errors have happened. 1: At least an error has happened; check ERROR subindex for more information. B=Booting type. 1bit. 0: Due to power on. 1: Due to any kind of reset such as: - SW Reset from heartbeat. - SW Reset from index 0x00 subindex 0x02. - SW Reset from DWL Function command (SW Update). - SW Reset from watchdog. - HW Reset. X=Reserved	W=0 R=0 B=0 E=0 H=0	
				EVENT	1	Binary format: XBEHRWWS	S=Status of the node. 1bit. 0: Initializing board and drivers. 1: Ready to work as R2CP node. WW=Working mode of the node. 2bits. 0: Normal operation. 1: Safety mode. 2: Service mode. 3. Interlock mode. R=Reserved. 1 bit. Value is set to 0x0. H=Heartbeat flag. 1 bit. 0: Heartbeat not running. 1: Heartbeat running. E= Error flag. 1 bit. 0: No errors have happened. 1: At least an error has happened; check ERROR subindex for more information. B=Booting type. 1bit. 0: Due to power on. 1: Due to any kind of reset such as:		

CONFIDENTIAL

⚠ Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.

						- SW Reset from heartbeat. - SW Reset from index 0x00 subindex 0x02. - SW Reset from DWL Function command (SW Update). - SW Reset from watchdog. - HW Reset. *X*=Reserved	
			SET	1	WW	Forced working mode. *WW*=Working mode of the node. 2bits. 0: Normal operation. 1: Safety mode. 2: Service mode.	
0x03	RESET	WR	SET	0		SW reset. After this command, application will be restarted. One booting application again, *STATUS* subindex is sent indicating Reset as booting type field.	
0x04	ERROR	WR/RD	SET	0		Removes current error showed. If there are not more errors pending to show, error flag in *STATUS* subindex will be clear. See list of errors for the specific node.	*EE*=0
			GET	0		Gets first error stored in a queue	
			ANSWER	1	EE	*EE*=Error code. See list of errors for the specific node.	
			EVENT	1	EE	*EE*=Error code. See list of errors for the specific node.	
0x05	MASTER_LIFE_TIME_OUT	WR/RD	SET	2	GG GG	*GG* = Master life time out. Units (10 mS) 0 means deactivated. If it's right configured, heartbeat flag field in *STATUS* subindex will be set. The master should polls each R2CP slave at regular time intervals, by using the command/function *HEARTBEAT*. If *MASTER_LIFE_TIME_OUT* is configured, and the node didn't get any *HEARTBEAT* for a *MASTER_LIFE_TIME_OUT* time, an *ERROR* event will be sent, and the node will be restarted. Once node boots again, heartbeat flag field in *STATUS* subindex will indicate it's not running any more. Master should configure it if it requires it.	*GG*=0
			GET	0		Request master life time out configured.	
			ANSWER	2	GG GG	*GG* = Master life time out. Units (10 mS) 0 means deactivated.	
			SET	3	VV SS RR	Set new protocol version. In case of new protocol version is supported, an event with the new protocol supported.	

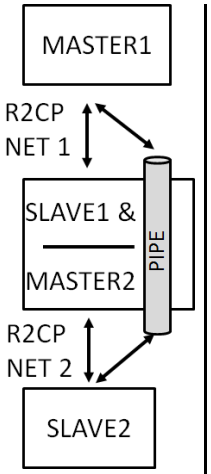
CONFIDENTIAL

⚠ Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.

	0x06	PROTOCOL_VERSION	RD				If the node not support the protocol version sent, it responds with a command processed message and response code 0x01. If the node not support protocol version changes, not available message should be sent. Format: vVV.sSS rRR *VV*=Version. 1 byte, decimal format *SS*=Subversion. 1 byte decimal format *RR*=Review. 1 byte, ASCII format		
				GET	0		Requests protocol version. This field must fit Radiological Room Control Protocol document version of the appropriated node.		
				ANSWER	3	VV SS RR	Format: vVV.sSS rRR *VV*=Version. 1 byte, decimal format *SS*=Subversion. 1 byte decimal format *RR*=Review. 1 byte, ASCII format Ex: *0x01 0x0A 0x41* → V1.10 A		
				EVENT	3	VV SS RR	After a software update, a new protocol version was found . Format: vVV.sSS rRR *VV*=Version. 1 byte, decimal format *SS*=Subversion. 1 byte decimal format *RR*=Review. 1 byte, ASCII format		
	0x07	BOOT_VERSION	RD	GET	0		Request bootloader version		
				ANSWER	3	VV RR SS	Format: vVrRR.S *V*=Version. 1byte, decimal format. It should be increased when there are big incompatibilities with previous version such as different hardware. *R*=Review. 1 byte, decimal format. It should be increased when there are big modifications in software. *S*=SubReview. 1 byte, decimal format. It should be increased when there is any modification in software. Ex: 0x01 0x0A 0x03 → V1R10.3		
	0x08	PACKAGE_VERSION	RD	GET	0		Requests package version. Package version can be implemented when node/slave integrates one/several subnodes/devices/modules with versions able to be got.		

CONFIDENTIAL

⚠ *Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.*

				ANSWER/ BLOCK (ANSWER)	X	STRING	String (ASCII characters) ending by NULL character. LF and CR characters can be used to send each version into a new line, and TAB for submodules. Example: *Module1:Module1Version\r\nModule2:Module2Version\r\n\tSubmodule1:Submodule1Version\r\n\tSubmodule2: Submodule2Version \r\n\0*		
				EVENT / BLOCK (EVENT)	X	STRING	String (ASCII characters) ending by NULL character. LF and CR characters can be used to send each version into a new line, and TAB for submodules. Example: *Module1:Module1Version\r\nModule2:Module2Version\r\n\tSubmodule1:Submodule1Version\r\n\tSubmodule2: Submodule2Version \r\n\0*		
	0x09	PIPE_CONFIG	WR/RD	SET	3	ON NN II	The pipe can be used to get access to nodes out of the master R2CP net, nodes which hang from the slave, but in a different R2CP net. *ON*=ON/OFF. 1 means ON. *NN*=SubNode Id, node id of the node which hangs from the slave, but in a different R2CP net. 0 means every node. *II*= Interface number, because the slave can have more than one R2CP NET hanging from it.	*ON=0* *NN=0*	
				GET	0		Gets pipe configuration.		
				ANSWER	3	ON NN II	*ON*=ON/OFF. 1 means ON. *NN*=SubNode Id, node id of the node which hangs from the slave, but in a different R2CP net. 0 means every node. *II*= Interface number, because the slave can have more than one R2CP NET hanging from it.		
	0x0A	PIPE	WR/RD	SET/ BLOCK (SET)	X	HH HH HH HH DD*	*HH*=Header of the message to send to the other side of the pipe configured. *DD*=Data (length specified by the message) of the message to send to the other side of the pipe configured.		
				EVENT/ BLOCK (EVENT)	X	HH HH HH HH DD*	Message got for the pipe configured *HH*=Header of the message got from the other side of the pipe configured. *DD*=Data (length specified by the message) of the message got from the other side of the pipe configured.		

CONFIDENTIAL

⚠ *Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.*

0x0B	SERIAL NUMBER (Appendix 2)	RD	SET	X	STRING	Sets a new serial number (Maximum string size is 10 characters) of the node/equipment ending by NULL character	
			EVENT BLOCK (EVENT)	X	STRING	Sends the new serial number (Maximum string size is 10 characters) updated ending by NULL character.	
			GET	0		Gets serial number (Maximum string size is 10 characters) of the node/equipment ending by NULL character. If the serial number has not been loaded, the node/equipment will send a string with "NA". However if the option is not available in the board will send a Not Available Response.	
			ANSWER BLOCK (ANSWER)	X	STRING	Sends the Serial number (Maximum string size is 10 characters) saved ending by NULL character.	
0x0C	DESCRIPTION	RD	GET	0		Gets a short description of a node.	
			ANSWER/ BLOCK (ANSWER)	X	STRING	Description. String ending by NULL character.	
0x0D	NODE ID CONFIG	WR/RD	SET	1	NN	Configures a different node id from the current one. *NN*=New node id. [1 to 31].	
			NOT_AVAILABLE	0		Event sent when value isn't allowed.	
			EVENT	1	NN	*NN*=New node id configured. [1 to 31]. This event is sent yet with the node id before the configuration. Ahead the node works with the node id configured	
0x0E	INTERLOCK	RD	EVENT	2	II SS	II: Interlock code. See list of interlock for the specific node. SS: 0 Deactivated. 1 Activated	
			GET	0			
			ANSWER	2	II SS	II: Interlock code. See list of interlock for the specific node. SS: 0 Deactivated. 1 Activated	
0x0F	WARNING	R	EVENT	1	W	W: Warning Code. See list of warning for the specific node.	
0x10	COMPILATION DATE (Appendix 3)	R	GET	0	STRING	Gets compilation Date of the node/equipment ending by NULL character. If the compilation has not been implemented, the node/equipment will send a string with "NA". However if the	

CONFIDENTIAL

⚠ Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.

						option is not available in the board will send a Not Available Response.	
			BLOCK (ANSWER)	24	STRING	Send the Compilation Date ending by NULL character.	
0x11	System versions	RD	GET	0		Request software system versions	
			BLOCK (ANSWER)	18	II MM MM VV RR SV SR SS BV BR BS HV HR HS GV GR GS AA	II Slaves/Master Identifier Format:AMMMM-VV-R *M*=Model. 2 bytes, decimal format *V*=Version. 1 byte, decimal format *R*=Revision. 1 byte, ascii format Ex: 0x0E 0x20 0x01 0x41 → A3616-01-A Format: vxVrxR.xS *xV*=Version. 1byte, decimal format. It should be increased when there are big incompatibilities with previous version such as different hardware. *xR*=Review. 1 byte, decimal format. It should be increased when there are big modifications in software. *xS*=SubReview. 1 byte, decimal format. It should be increased when there is any modification in software. Ex: 0x01 0x0A 0x03 → V1R10.3 Where: SV SR SS →Software Version BV BR BS →Bootloader Version HV HR HS →Hdl Version GV GR GS →Golden Version AA if the version is available: Bit 0: Hardware Version Apply Bit 1: Software Version Apply Bit 2: Boot Version Apply Bit 3: Hdl Version Apply Bit 4: Golden Version Apply Ex: Apply Only Hdl version → 0x08	

CONFIDENTIAL

⚠ *Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.*

Figure 21 – R2CP User Dictionary Object

CONFIDENTIAL

⚠ Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.

11 APPENDIX 2: SERIAL NUMBER

Codification of serial number requires 10 bytes memory space: N characters string followed by NULL character.

Currently, it consists of 9 ASCII characters:

2 bytes: Manufacturer codified in one/two characters, space/letter, (right justified, " A" - "ZZ").

2 bytes: Year ("00" – "99")

2 bytes: Week ("01" – "99")

3 letters: ("AAA" – "ZZZ")

Examples "␣A1824AAA"<NULL>

"ZZ1801AAA"<NULL>

This content may be transparent for the firmware. These figures are intended to be used just in display/records.

If no serial number saves, the node should send: "NA"<NULL>

CONFIDENTIAL

⚠ Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.

12 APPENDIX 3: COMPILATION DATE

The compilation date shows when the firmware has been compiled. It can be generated including the next string into the firmware code:

```
const char compileDate[] = __DATE__ " at " __TIME__;
```

It has to be auto-generated every time the code is compiled, it is necessary adding next configuration into project properties:

Build→Steps: ${CCS_INSTALL_ROOT}/utils/bin/gmake -B ./main.obj

CONFIDENTIAL

⚠ Confidentiality and privacy notice: All rights reserved. Unless otherwise specified and without written permission from SEDECAL, no part of this documentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, and may not be distributed or transmitted to third parties.