CAREL – Confidential



**REQUIREMENTS SPECIFICATION**

CloudGateMini

Step 2

rev. See history

Index

[Index 2](#_Toc62111250)

[1 Revision 3](#_Toc62111251)

[2 Introduction 4](#_Toc62111252)

[2.1 Scope of the document 4](#_Toc62111253)

[2.2 Definitions, acronyms and abbreviations 4](#_Toc62111254)

[3 Upload of log data read from the connected device 5](#_Toc62111255)

[3.1 Preface 5](#_Toc62111256)

[3.2 File contents 5](#_Toc62111257)

[3.3 Flow of a full file transfer 7](#_Toc62111258)

[3.4 Flow of a partial file transfer 8](#_Toc62111259)

[3.5 Error management 9](#_Toc62111260)

[4 Background transfer 10](#_Toc62111261)

[5 ERROR CODES 11](#_Toc62111262)

[6 Useful links 12](#_Toc62111263)

1. Revision

|  |  |  |  |
| --- | --- | --- | --- |
| Rev. | Rev. date | Author | Note |
| *0.1* | xx/12/2020 | A.Bilato | 1st draft |
| 0.2 | 15/12/2020 | A.Bilato | Added useful link |
| 0.3 | 16/12/2020 | A.Bilato | Added log contents |
| 0.4 | 20/01/2021 | A.Bilato | Revised the content according to the meeting with IoT |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Introduction
   1. Scope of the document

This document refers to the realization of some enhancement to the current FW of the CloudGateMini.

* 1. Definitions, acronyms and abbreviations

AP = WiFi access point to connect to

CGM\_AP = gateway that act as an access point  
CGM\_WIFI = CloudGateMini WiFi

CGM = CloudGateMini  
GSM = refer to 2G/4G/NB IoT connection  
FW = firmware

IoT = Internet of Things

MonDev = the device connected to the GME through the RS485 interface

OTA = Over The Air

SW = software  
MFT = Modbus File Transfer

“model file” = a binary file that contains the data needed by the GME to read and write via ModBus the connected device.

MODEL\_TABLE = table defining the meaning of Modbus registers build from “model file”

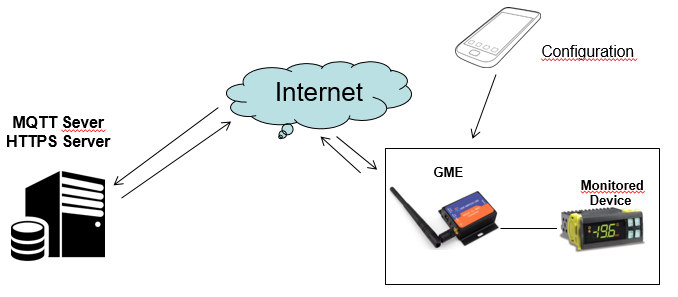
GTW000MWT0 = the GME-WiFi model

GTW000MGP0 = the GME-2G model with CAREL SIM installed

GTW000MGT0 = the GME-2G model without SIM

GTW000M2W0 = the GME-2G+WiFi model without SIM

1. Upload of log data read from   
   the connected device

  
Fig.1

* 1. Preface

Some of the CAREL controllers are able to store some logs and transfer it to a supervisory system through the Modbus File Transfer (MFT) protocol.

To do that with a CGM some modification are requested and are listed below, some must be implemented also server side.  
The block diagram of the system is explained on Fig.1.

To fully understand this document please refer also to the documentation of the other command of the CGM.

* 1. File contents

The file is transferred “as is” the CGM don’t take into account the means of the transferred bytes.  
The only elaboration done by the CGM is related to the interpretation of the   
“Compression Header” [[1]](#UL_1), this header is on top of the file and contain the total size of the file itself.   
This size is used to:

* calculate how many file chunks are needed
* is sent as part of the message [payload-upload\_file\_values.cbor](file:///C:\hwfwdept_proj\c780_carel_cloud_engine_binary\c780_carel_cloud_engine_binary_Step_2\Documents\specs\payload_examples\payload-upload_file_values.cbor)

Take into account that :

1. the Compression header is sent as part of the file.  
   Is a choice of the cloud to discard the “compression header” content on the fly or store it as part of the file.
2. The CGM does not check the file index (field “fid”) sent from the cloud, this means that is possible to read virtually any file index, the only check that the CGM perform is on the format and consistency of the “Compressor Header”.   
   Take into account that all the files must have the compression header on top.  
   Just for reference the file index currently in use are the ones you will find on [[2].](#UL_2)  
   If you send a command with a not existing “fid” or the file don’t have a valid “compression header” the CGM return an invalid “compression header” error.
3. the field “fsz” in the [payload-upload\_file\_values.cbor](file:///C:\hwfwdept_proj\c780_carel_cloud_engine_binary\c780_carel_cloud_engine_binary_Step_2\Documents\specs\payload_examples\payload-upload_file_values.cbor) is equal to

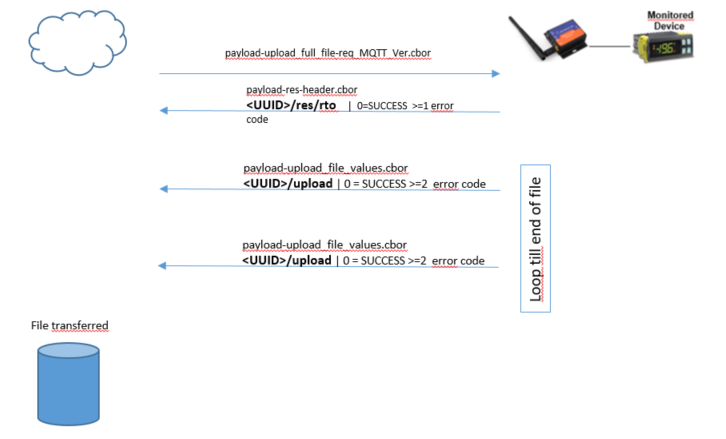
“fsz” = "compression header size" + "file CompressedSize"

or in other word the real file is a little bit small, just "compression header size" less.

1. the field “fsz” in the [payload-upload\_range\_file-req\_MQTT\_Ver.cbor](file:///C:\hwfwdept_proj\c780_carel_cloud_engine_binary\c780_carel_cloud_engine_binary_Step_2\Documents\specs\payload_examples\payload-upload_range_file-req_MQTT_Ver.cbor) is equal to  
      
    “fsz” = “fle”  
     
   this because this commend don’t trigger the CGM to read the “compressed header”.

* 1. Flow of a full file transfer

Below the flow to transfer a file from the monitored device to the cloud.

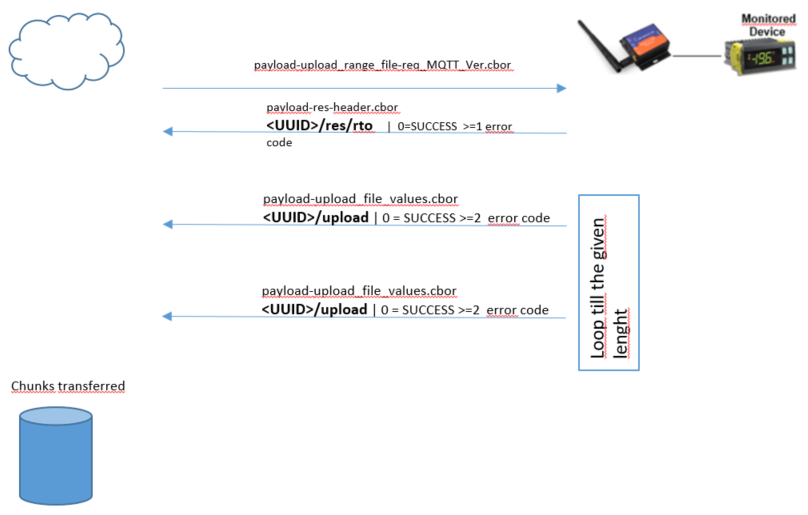


* [payload-upload\_range\_file-req\_MQTT\_Ver.cbor](payload_examples/payload-upload_range_file-req_MQTT_Ver.cbor)   
  in this example the request for the full download of the specified file.  
  This is the command Id 19
* [payload-res-header.cbor](payload_examples/payload-res-header.cbor)  
  this is the standard response header of every command.  
  Note that the CGM respond immediately OK/FAIL to this command and if OK start the transfer.
* [payload-upload\_file\_values.cbor](payload_examples/payload-upload_file_values.cbor)  
  this is the response that transport a chunk of file, so that to transport the whole file the CGM sent some packets of this type.  
  The “fsz” returnend is equal to "compression header size" + "file CompressedSize"

Note  
If a transfer command is already running and the CGM receive again the same command the CGM return error “5”

* 1. Flow of a partial file transfer

Below the flow to transfer a file from the monitored device to the cloud.



* [payload-upload\_range\_file-req\_MQTT\_Ver.cbor](payload_examples/payload-upload_range_file-req_MQTT_Ver.cbor)   
  in this example the request for the full download of the specified file.  
  This is the command Id 20.
* [payload-res-header.cbor](payload_examples/payload-res-header.cbor)  
  this is the standard response header of every command.  
  Note that the CGM respond immediately OK/FAIL to this command and if OK start the transfer.
* [payload-upload\_file\_values.cbor](payload_examples/payload-upload_file_values.cbor)  
  this is the response that transport a chunk of file, so that to transport the whole file the CGM sent some packets of this type.  
  The “fsz” returned is equal to “fle”.

The elaboration of the fragmentation of the file requested by the current implementation of the Modbus File Transfer is in charge to the CGM, this means that the “fst” parameter is always referred to the start of the “fid” file index passed.

An example of this :

“fid” = 1000

“fst” = 23004

“fle” = 500

This means that the CGM will try to read the data from the controller starting from

File index 1001, because the “fst” exceed 20000, and request the data from 3004 to 3504.

* 1. Error management

The errors are always returned in the field “res”.

There are some cases to take into consideration:

1. Power failure of the CGM during the transfer;   
   the cloud could detect it because receive a “hello” message immediately after the reboot. If the situation persists for some time the MQTT broker, emit also a LWT message.
2. Power failure of the device during the transfer;  
   this case is covered because the Modbus return timeout, and the CGM return a error code 4.
3. Modbus transfer error / exception / timeout;  
   if for unknown reason the device produce an error this case is covered because the Modbus return timeout or exception after the retry.  
   The CGM return an error code 4.
4. Internet connection lost;  
   E’ questa è una bella storia perchè se dura più di timeout del broker quest’ultimo manderà un LWT e il cloud considererà abortito il trasferimento, d’altro canto il CGM appena ritorna la connessione potrebbe continuare il trasferimento

1. Background transfer

The background transfer is possible, but this feature require defining a new way for the sampling of the alarms.  
Currently the alarms are sampled best-effort this means that the Modbus data flow is continuous without any interruption, so that, this method don’t permit to insert a new message to retrieve a data chunk on the middle.

After some research we have jointly assumed that is reasonable to **poll the alarms** every **5 sec** (see email), this free some time to get the data from the controller and upload it to the cloud.

Anyway take into account that the total transfer time of the log in background increase compared to an exclusive transfer.

1. ERROR CODES

This is the error code returned by the function

|  |  |
| --- | --- |
| **Code** | **Description** |
| -1 | Invalid command, possible if the CGM mount an old version of the FW |
| 0 | Success |
| 1 | Error |
| 2 | Compression header not found or not valid |
| 3 | Compression header version not supported; provided for future back compatibility |
| 4 | Transfer aborted due to Modbus error during the get of a chunk; the CGM try 3 retry, and after that return this error and abort the file transfer |
| 5 | Error Upload command already running |
|  |  |

1. Useful links

[1]Log File Format

<https://docs.google.com/document/d/19gSBbhbCRT1G5XsI6V0dhK3mYm31KwSa/edit>

Log format Marzolla vecchia versione non aggiornato

[https://docs.google.com/spreadsheets/d/1f\_rMX1lBmSu6YCM6COguO6nWo2AgXItDHF2cLJrEVGw/edit#gid=759605612](https://docs.google.com/spreadsheets/d/1f_rMX1lBmSu6YCM6COguO6nWo2AgXItDHF2cLJrEVGw/edit%23gid=759605612)

[2] Strategia Integrazione

<https://drive.google.com/file/d/1CrJS0_9Qj3wN0NtKzLcX3lbx9s_iJSVm/view?usp=sharing>

[3]STc

[https://docs.google.com/document/d/1DoFhnghlsaDkonmvz3HgNPmnK6CfHsbW/edit#](https://docs.google.com/document/d/1DoFhnghlsaDkonmvz3HgNPmnK6CfHsbW/edit)

[4] CAREL Design Standard Modbus Protocol

[https://svncarel.carel.com/svn/designstandards/trunk/Modbus Protocol](https://svncarel.carel.com/svn/designstandards/trunk/Modbus%20Protocol)