CAREL – Confidential



**How to install the GME FW**

*Gateway Middle End*

rev. See revision table

DRAFT

Index

[Index 2](#_Toc38292135)

[Revisione 3](#_Toc38292136)

[1 Instructions 4](#_Toc38292137)

[1.1 Prerequisites 4](#_Toc38292138)

[1.1 Connect the HW 4](#_Toc38292139)

[1.2 Install the FW 5](#_Toc38292140)

[1.3 Basic HW/FW check for the WiFi model 6](#_Toc38292141)

[Test TTL Interface with an CAREL IR33 6](#_Toc38292142)

[1.4 Basic HW/FW check for the 2G model 7](#_Toc38292143)

[GPIO34 (pin to detect if the HW is 2G or WiFi only) 8](#_Toc38292144)

[GPIO22 & GPIO23 (Power supply e Power Key) 8](#_Toc38292145)

[GSM-TX and GSM-RX pin 8](#_Toc38292146)

[SIM card 8](#_Toc38292147)

[Other possible situation. 8](#_Toc38292148)

[1.5 Additional informations 9](#_Toc38292149)

[Erase the ESP32 Flash 9](#_Toc38292150)

[Speed up the programming time 9](#_Toc38292151)

Revisione

|  |  |  |  |
| --- | --- | --- | --- |
| Rev. | Rev. date | Author | Note |
| *0.01* | *30/03/2020* | *A.Bilato* | 1° draft |
| 0.02 | 09/04/2020 | A.Bilato | Added 2G section |
| 0.03 | 20/04/2020 | A.Bilato | Added the IR33 test section |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Instructions

Here below some brief instruction to program the ESP32 inside the Gateway Middle End (GME).

**WARNING!**

The given FW is not official, so something probably will be adjusted, ie. in the batch files probably some number will change but the rest remain the same.

* 1. Prerequisites

To run the FW uploader you need to install in your Windows computer the following :

1. Python 2.7.x
2. Make sure that Python is present in “PATH”
3. Install pySerial V.3.0 or newer
4. Dezip the CAREL GME\_Production\_Line\_Batch.zip file in a folder   
   ie. C:\ GME\_Production\_Line\_Batch

If you have some doubt refer to

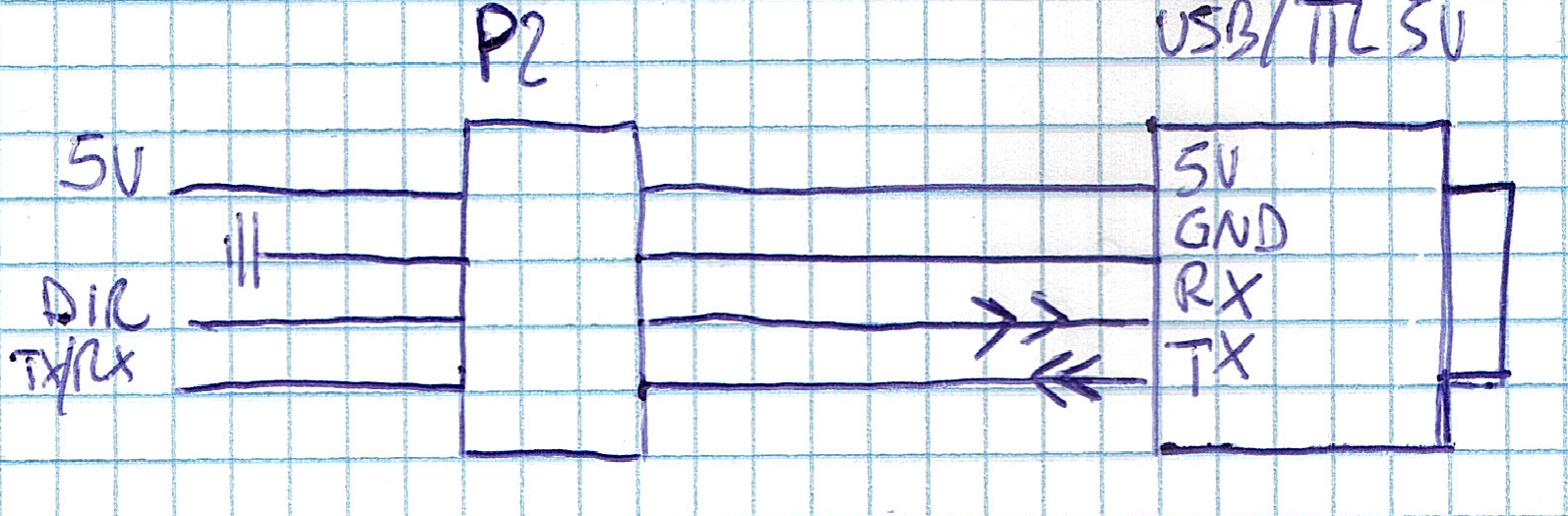
<https://github.com/espressif/esptool>

where there are the instructions to install the programming tool.

You need also a :

1. USB/TTL 5V serial converter
2. Mobus Slave simulator like Modscan32.
3. A terminal emulator, we suggest Putty.
   1. Connect the HW

You need to connect the USB/TTL 5V serial converter to the TTL serial port of the GME,  
the wiring in the below schema (Fig.1).

  
Fig.1

* 1. Install the FW

1. To install the FW open a Windows command prompt window (cmd.exe),   
   change the working directory to the one where you have unzipped the CAREL files (ie. C:\ GME\_Production\_Line\_Batch).
2. Press the GME button, and apply the power supply
3. Run the batch   
   GME\_ESP32\_Upload.bat *COMxx* 115200  
   where COMxx is the serial port Windows assigned to you USB/TTL 5V serial converter, 115200 is not the maximum speed but useful to initially test the GME.
4. After a while you will see something similar to the content of Fig.2

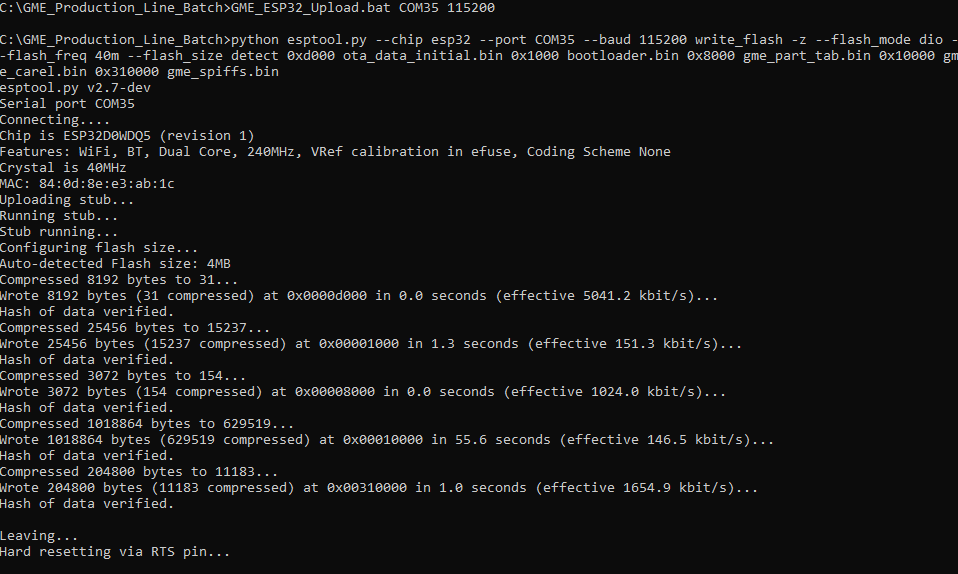


Fig.2

1. Power off the GME
   1. Basic HW/FW check for the WiFi model
2. Power on the GME, take a look to the leds, at the beginning you will see 3 sequence of green/red blinking.
3. After some second the green led slowly blink
4. Use a smartphone and search an AP called “CGATEM\_xxyyzz” where xxyyzz are the last 3 digits of the MAC address of the GME, and connect to it.
5. Open a browser (ie. CHROME) and open the URL <http://10.10.100.254>
6. At first time the GME ask for a new Username and password   
   type “Admin” and “12345678” and press “Submit”

The GME store the new credential of the configuration page and prompt for login.

1. The GME ask again for Username and Password use “Admin” and “12345678”
2. The GME prompt now, for some data, to connect to a router or an AP to access the Internet.  
   Select “Scan” and select an AP SSID a suitable to connect to Internet.

Insert also the Password of the AP and press Submit.

1. At this point the GME reboot itself and connect to selected AP, and try to connect to CAREL server.
2. The green led is ON. (Note A)
3. The test FW of GME will try to read via RS485 some coils and registry.   
   Use a SW like ModSim32 to emulate on a PC, 100 Holding register and 100 Coils to respond to the GME enquiry.

Note A

If you give to CAREL the MAC address of the GME under test, we provide to register it to our server. After a successful registration on the server the led green go on.

Test TTL Interface with an CAREL IR33

On normal usage, the TTL port is dedicated to interfacing directly, a CAREL IR33 controller with the GME. To test the HW compatibility of this port we need to try to connect directly an IR33 to the GME, unfortunately we have not any other additional serial port suitable to output debugging informations, so that, the only way is to obtain this informations indirectly.  
To do that you need to load a special version of the FW nameoffile.zip, the instructions are the same described in chapter 1.2 above.

This FW version do the following:

1. Load a model file for the IR33.  
   This model monitor only 2 COILS and 2 HR, this to try to avoid problems with different versions of IR33.
2. The bicolor led of the GME now use only the red led to show you if the   
   TTL communication is right or not, in short if the led is on the TTL communication is OK otherwise something don’t work correctly.
3. The communication with the cloud is active, so is possible to see a little bit more indirectly, but you need to give us the MAC address of the device to enable it on the cloud server, otherwise it don’t work.

To test the device you need:

1. Load the test FW
2. Connect the GME WiFi to an IR33 through the TTL port (the TTL cable is the one already in use with the TTL/RS485 adapter of the IR33 sorry I don’t have the part number here)
3. Power on the system, the GME show the led test as usual
4. Configure the GME to access the Internet
5. Wait at least 1 minute
6. The green led is ON  
   1. Basic HW/FW check for the 2G model

The below information are preliminary this because the FW is currently under construction.

1. Insert a SIM able to connect to the network and to the Internet.
2. Power on the GME\_2G, take a look to the led.  
   The power led will light.  
      
     
   The led below will light sequentially on and off, escluding .  
     
       
     
   After that, the FW will power on the M95 module, if everything will go right

the led  start to blinking slow, obviously a SIM card must be present.  
  
At the end the situation (could be) this  
   
  
  
If this happen, it means that   
- The I/O pin GPIO34 is OK.  
  
the M95 module is correctly driven  
- The I/O pin GPIO23 that drive PWR-KEY is OK.  
- The I/O pin GPIO22 that drive V\_BAT is OK.  
- The GSM-TX and GSM-RX pin works right.

If something was wrong (hope no) do the following:

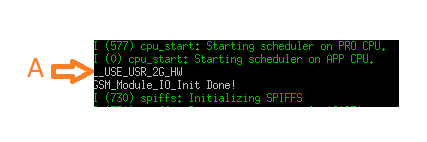
1. In the debugging FW we have left active the output of the debugging message through the TTL serial port.

First step is to open a terminal emulator, ie. Putty and connect with the USB/TTL adapter as in Fig.1.  
Activate the logging of the session in Putty, so that is possible for you to send back to us all the messages that appear.

Here below, some points, useful to find some trouble if will happen, are listed in chronological order.

GPIO34 (pin to detect if the HW is 2G or WiFi only)

In the terminal at very beginning you could see \_\_USE\_USR\_2G\_HW, if not the GPIO34 is not recognized well.

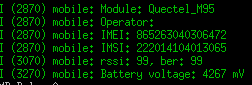


GPIO22 & GPIO23 (Power supply e Power Key)

If you reach this point  the power supply of M95 is ON and the power pulse to PWR\_KEY performed.

GSM-TX and GSM-RX pin

If you will find something like this, the communication with the module is ok.



SIM card

If you will find something like this, the SIM card is OK and you are connected to the Internet.

****

Other possible situation.

GME\_2G continuously rebooting – case 1.  
If this happen >5 times there are, probably, a trouble with GPIO22 & GPIO23.  
Check with an oscilloscope the state of the pins.

GME\_2G continuously rebooting – case 2.  
If the previous point is OK, another possibility is a FW trouble, where GSM-TX and GSM-RX are swapped. In this case try the FW version. …RXTX\_Swap… and try again.

* 1. Additional informations

Here below some additional informations

Erase the ESP32 Flash

If you want to erase the entire ESP32 Flash memory and reset also the area where the login data are stored you simply launch the batch

GME\_ESP32\_eraseflash.bat COMxx

Where COMxx is the same as above.

Speed up the programming time

If you will find that the batch GME\_ESP32\_Upload.bat work well at 115200 bps, you could try to use the upper baud rate of 921600 bps.

If doesn’t work well try 460800 or 230400.

Determinate the maximum speed is useful to understand the programming time in production line.