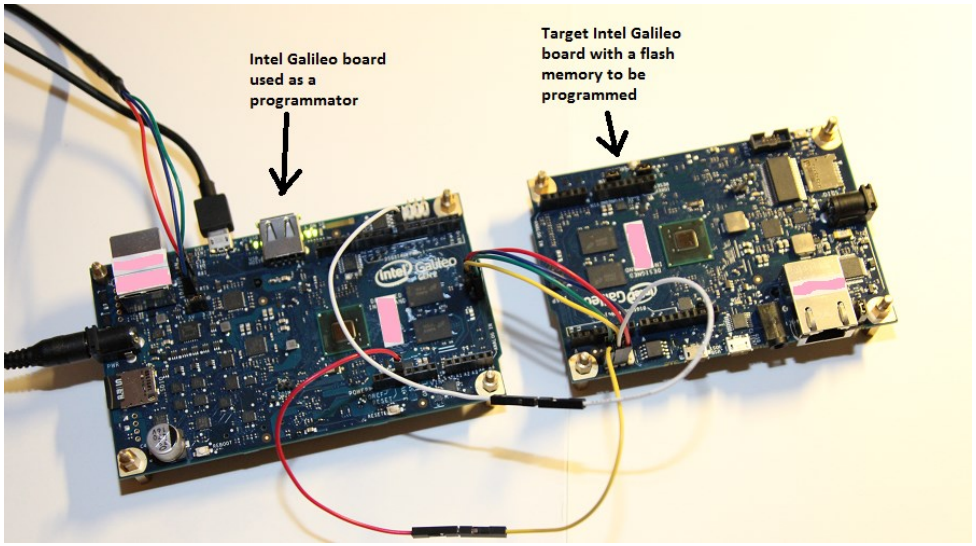
GaliProg v1.1

15.02.2015
xbolshe
pub@relvarsoft.com

1. GaliProg… What is it? It is a tool (sketch) which allows to read/program/erase/verify SPI flash memory image on Intel® Galileo board. Galiprog may help in a situation when Galileo board is bricked after a failed firmware upgrade.

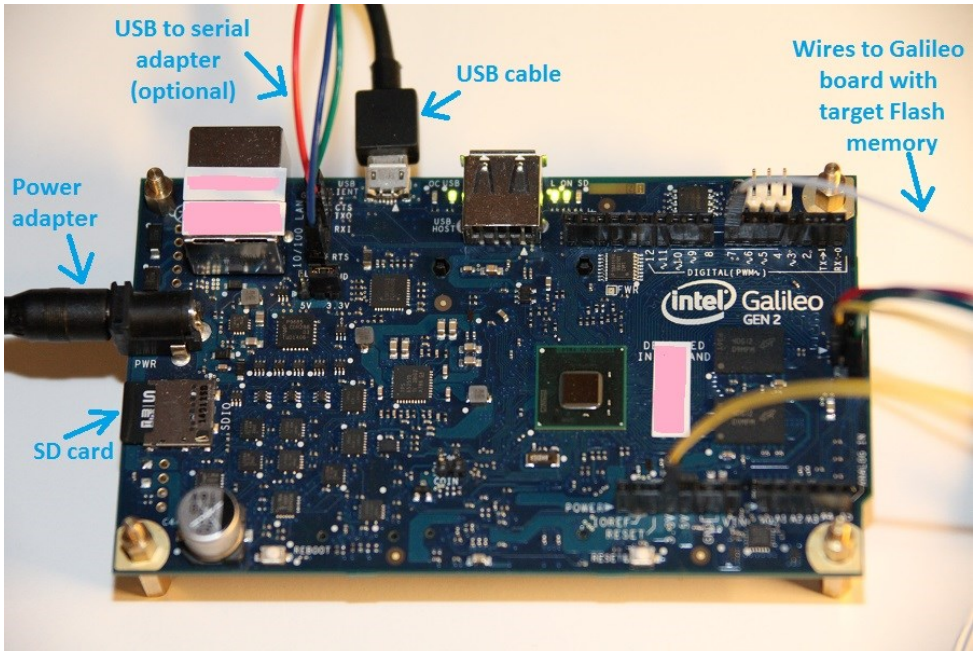
   NOTE: it was tested only in two configurations:
   a) when Galileo Gen 2 used as programmator and Galileo Gen 1 used as target board;
   b) when Galileo Gen 1 used as programmator and Galileo Gen 2 used as target board.



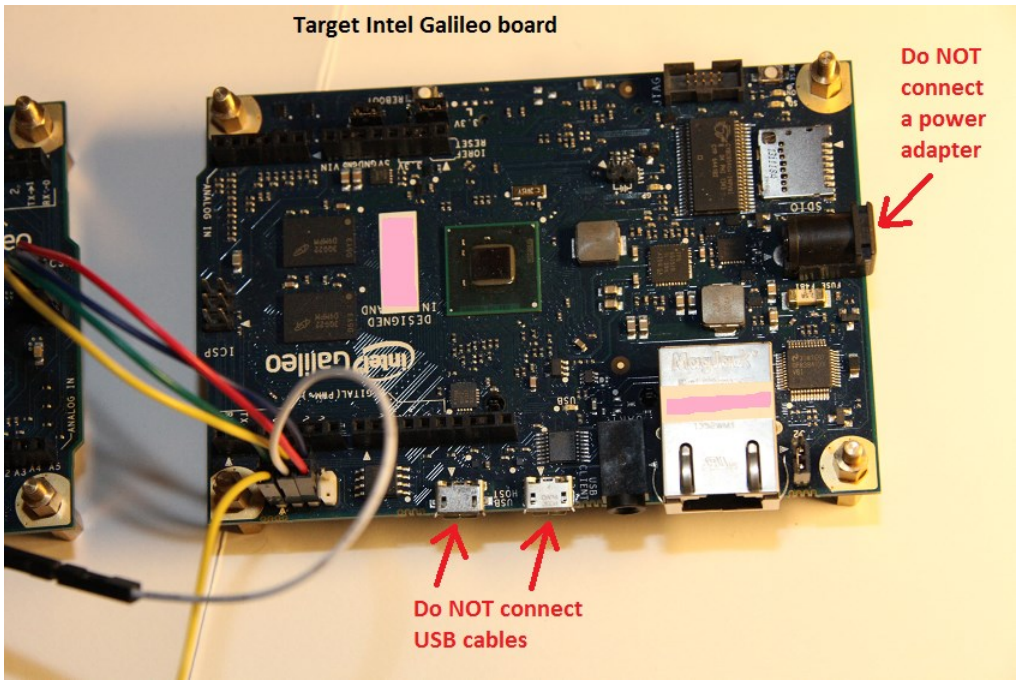
2. Required hardware

Need to have the following items to program SPI flash memory:

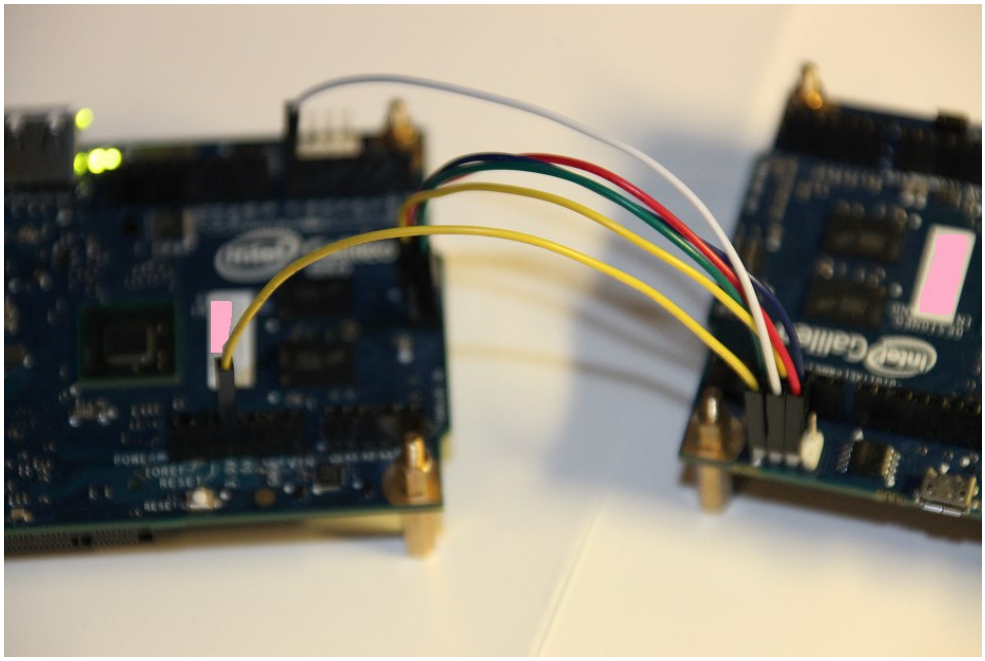
a) Properly worked Intel Galileo board with USB cable, micro SD card and power adapter. It will be used as used as a programmator.



b) Galileo board with target Flash memory



c) wires to connect Galileo boards



d) PC with installed Intel Arduino Software 1.5.3

e) Micro SD card reader

f) Two 10k Ohm resistors (in case when Galileo Gen 1 used as programmator and Galileo Gen 2 used as target board)

3. Required software

a) Intel Arduino Software (IDE) 1.5.3 created for Intel Galileo board
Link to download: https://communities.intel.com/docs/DOC-22226

b) SD-Card Linux Image
Link to download: https://communities.intel.com/docs/DOC-22226

c) SPI flash image

Select a way to get SPI flash image from described below:

Official way:

1) Flash Missing PDAT Release (.bin file)
   Link to download: https://communities.intel.com/docs/DOC-22226

2) BSP Patches and Build Instructions
   Link to download: https://communities.intel.com/docs/DOC-22226

Following the instruction above need to patch .bin file with a required platform configuration. Next need to rename a resulting file 'Flash+PlatformData.bin' to 'galiprog_flash_write.bin'.

Simplified way:

If you do not want to read documentation, compile and patch a firmware, I recommend you to use this way.

1) Galileo SPI Binary Pack 1.0.4
   This minimized pack contains all required tools and data to create SPI flash image. It is based on BSP 1.0.4. Need just to enter MAC address and Platform type to create a flash image with name 'galiprog_flash_write.bin'.

   See Annex 1 below for license information.

   Link to download: https://github.com/xbolshe/galiprog

Clone way:

If you have a problem with generation of SPI flash image with your MAC address, it is possible to copy SPI flash image from one board (use same Gen !) and copy it to another board. Need just to rename 'galiprog_flash_dump.bin' to 'galiprog_flash_write.bin'.

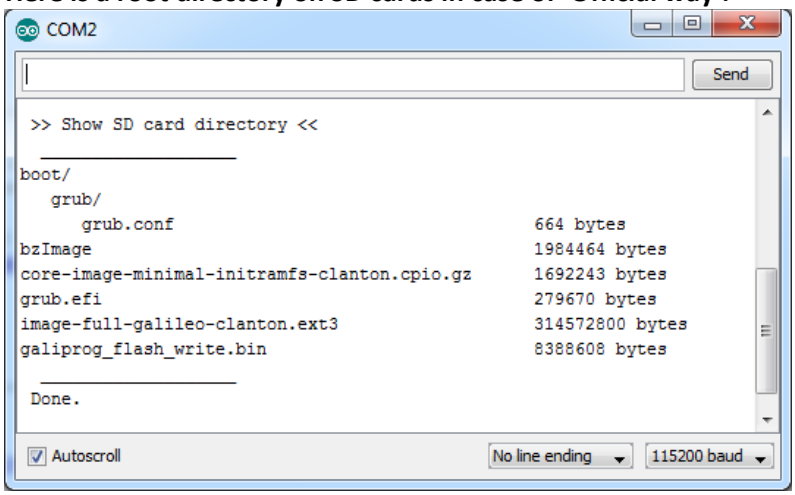
d) Galiprog (galiprog.ino)
This is a flash programing tool.

Link to download: https://github.com/xbolshe/galiprog

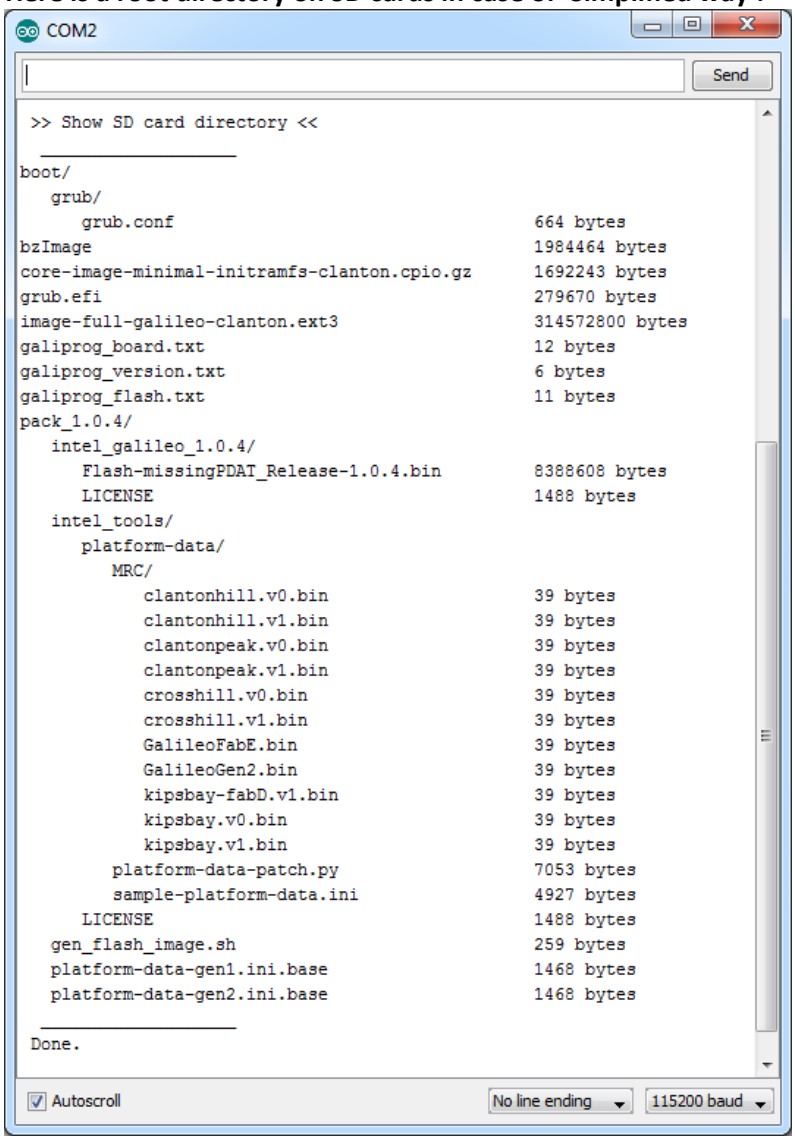
4. **Prepare a data on SD card**

   a) **Format SD card**
   b) **Unpack SD-Card Linux Image to the root of SD card**
   c) **Copy 'Flash+PlatformData.bin' as 'galiprog_flash_write.bin', if you selected Official way.**

Here is a root directory on SD cards in case of 'Official way':



Here is a root directory on SD cards in case of 'Simplified way':



5. Connections between Galileo boards and hardware settings

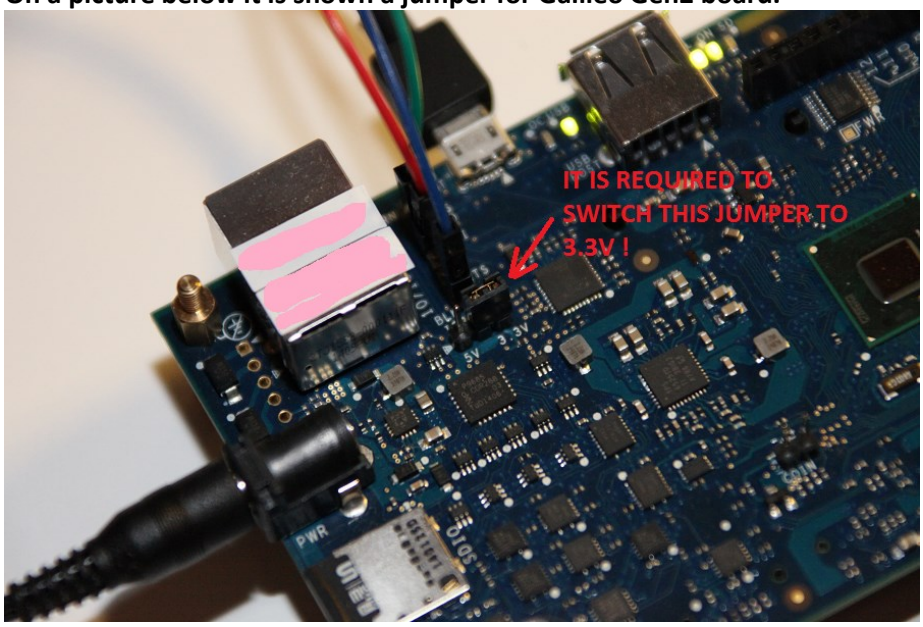
a) Configure Galileo board which works as a programmator

A SPI flash memory works with 3.3V lines. So, it is required to switch Galileo board - programmator to 3.3V.

NOTE: providing 5V may damage your Galileo board! Be careful with connecting boards and selecting a jumper setting.

Need to switch a jumper shown on a picture below to 3.3V option.

On a picture below it is shown a jumper for Galileo Gen2 board.



In case of Galileo Gen1 a jumper is located near with REBOOT button.

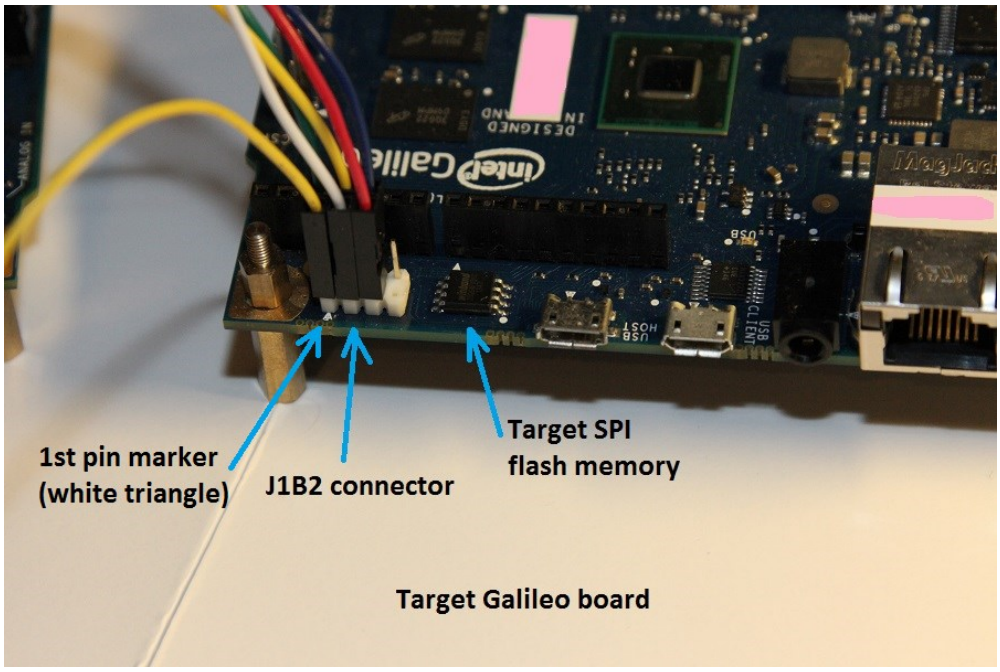
b) Wire connections

№	Galileo Gen2 board - programmator	Signal role	Galileo Gen1 board - target
1	3.3V	VCC	J1B2 – pin 1
2	Digital IO7	Slave selection	J1B2 – pin 3
3	ICSP – pin 4	MOSI	J1B2 – pin 6
4	ICSP – pin 1	MISO	J1B2 – pin 5
5	ICSP – pin 3	SCK	J1B2 – pin 4
6	ICSP – pin 6	Ground	J1B2 – pin 2

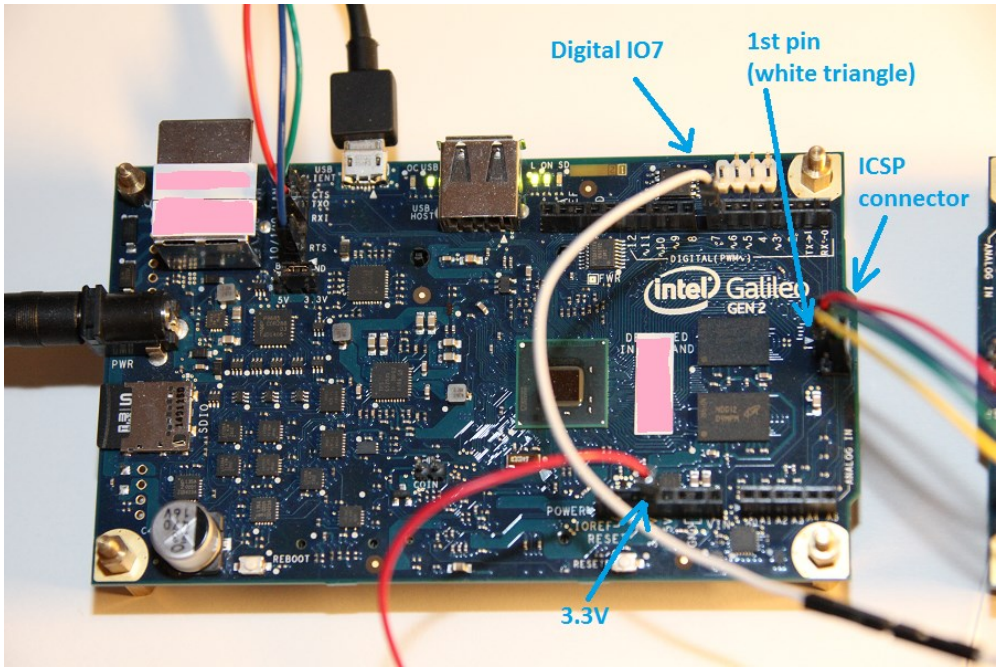
№	Galileo Gen1 board - programmator	Signal role	Galileo Gen2 board - target
1	3.3V	VCC	J1B2 – pin 1
2	Digital IO7	Slave selection	J1B2 – pin 3
3	ICSP – pin 4	MOSI, pull-up resistor 10kOhm	J1B2 – pin 6
4	ICSP – pin 1	MISO, pull-up resistor 10kOhm	J1B2 – pin 5

5	Digital IO13	SCK	J1B2 – pin 4
6	ICSP – pin 6	Ground	J1B2 – pin 2

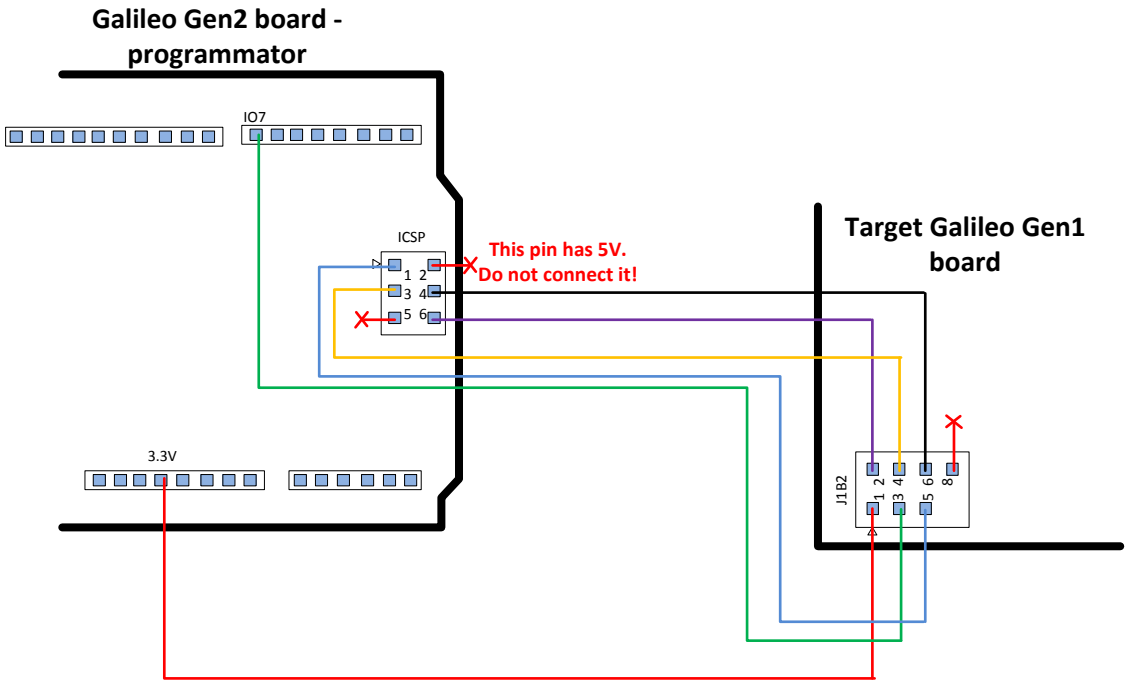
A location of J1B2 connector is shown on a picture below:

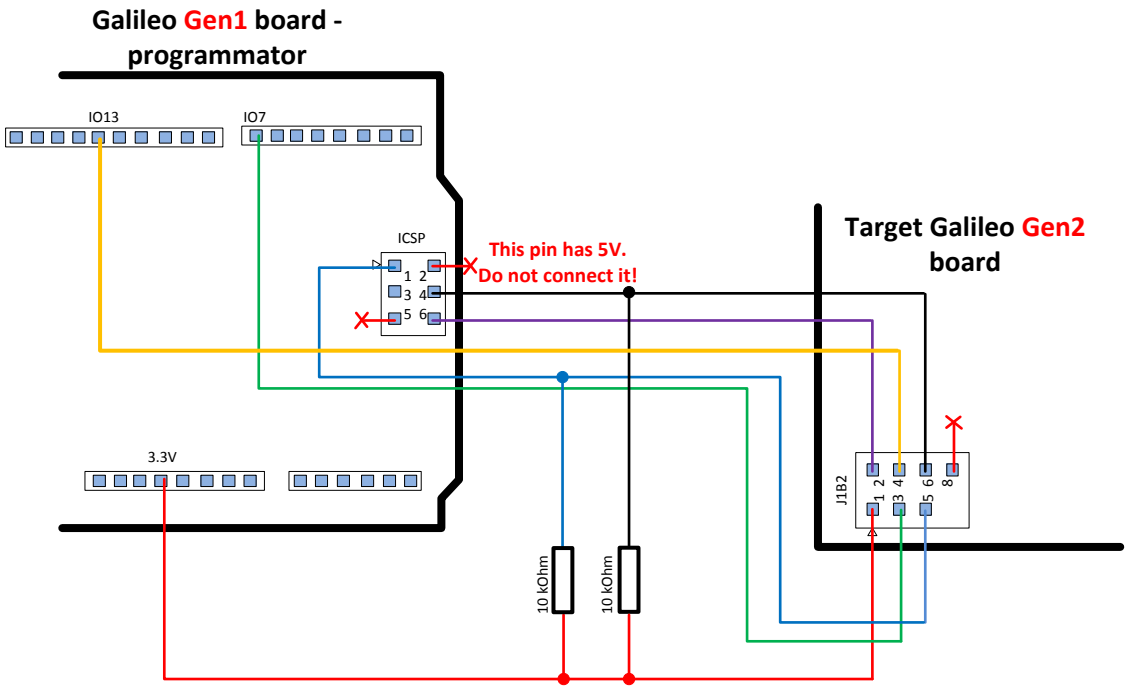


A location of ICSP connector is shown on a picture below:



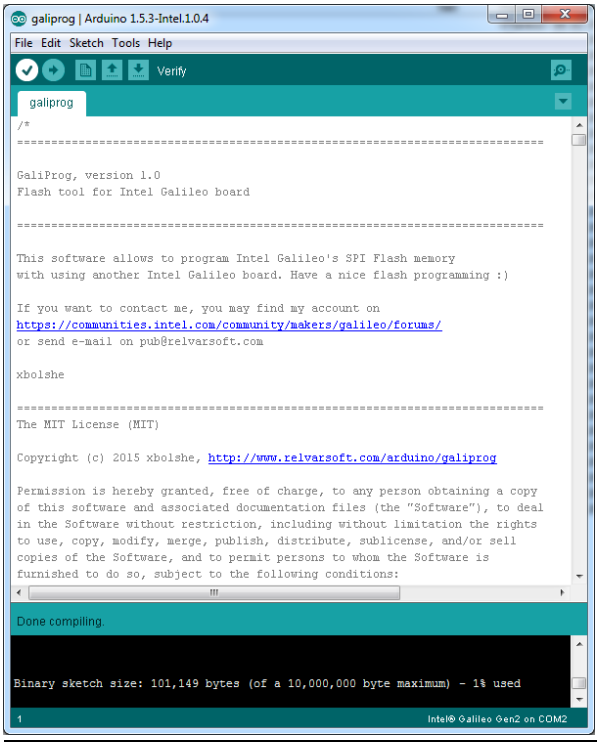
Select a scheme below for your boards.

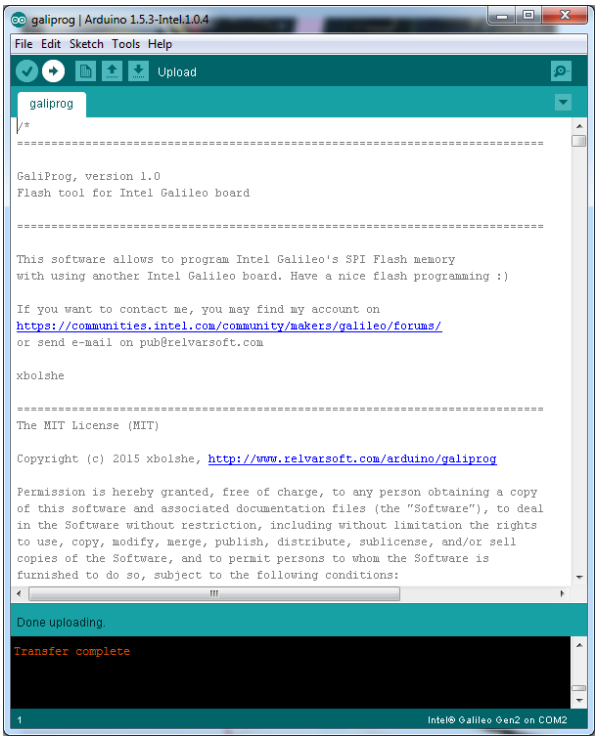




6. Compiling Galiprog sketch

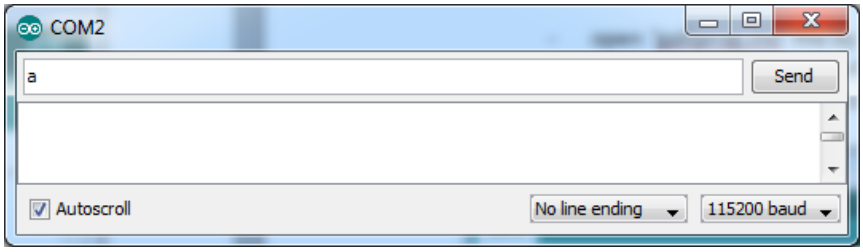
- open ‘galiprog.ino’ file by Intel Arduino Software (IDE) 1.5.3
- compile it with using ‘Verify’ button
- upload it to Galileo Board with using ‘Upload’ button



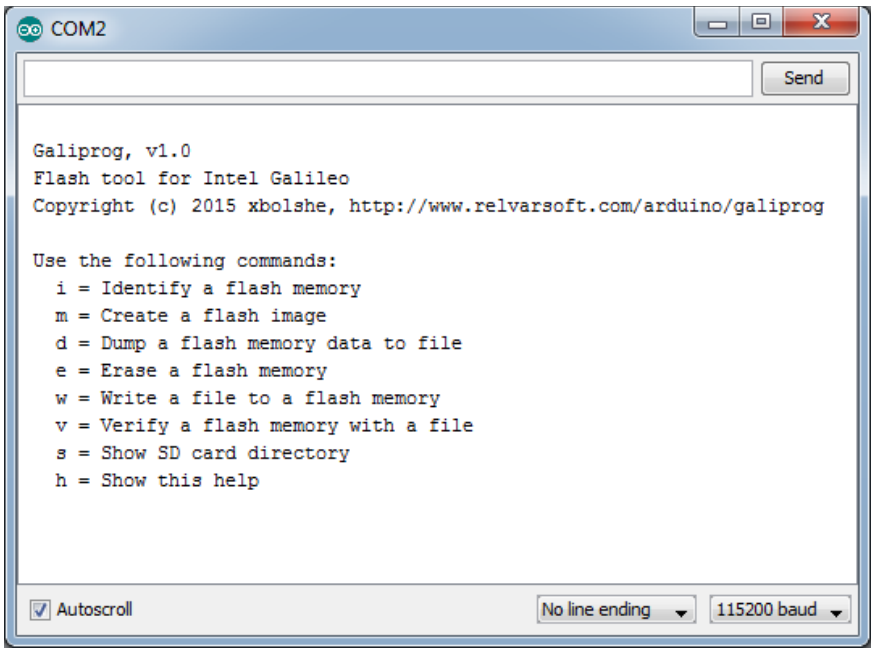


7. Galiprog commands

- when galiprog is uploaded to Galileo board, select Tools -> Serial Monitor
- type any character and push 'Send' button



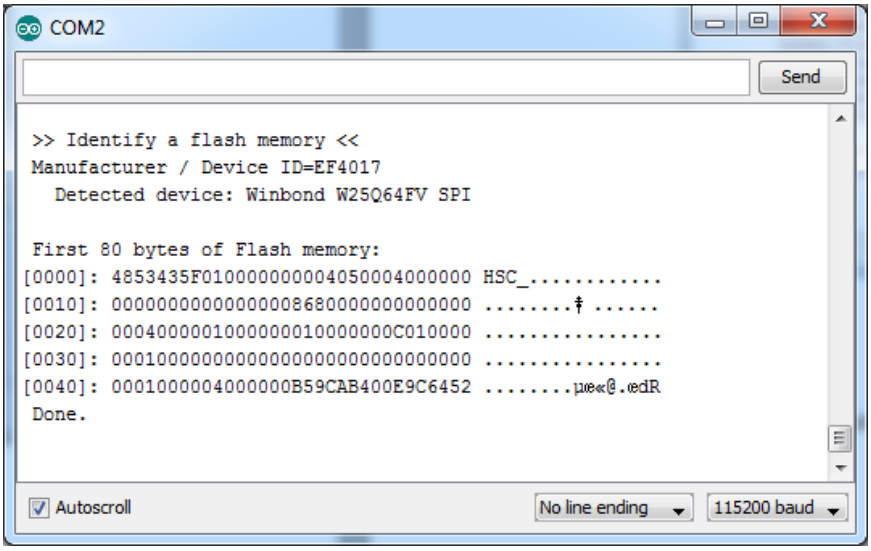
- a command list will be shown



To select menu item type a letter and push ‘Send’ button.

1. Identify a flash memory

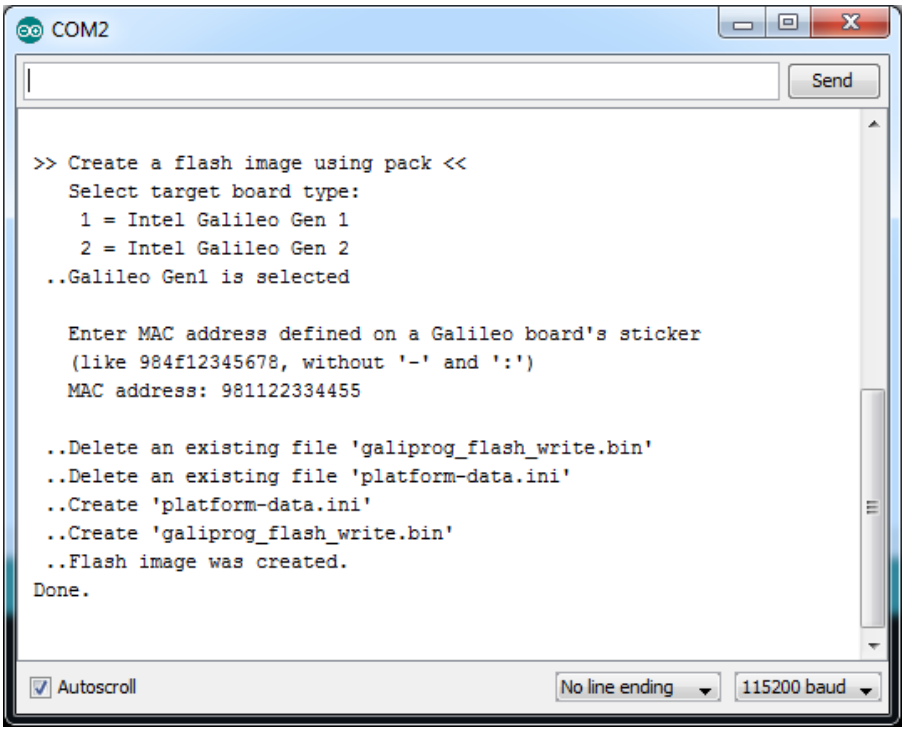
   This menu item allows to check that a connection with a target Galileo board is correct.
   It is recommended to use it before operations.

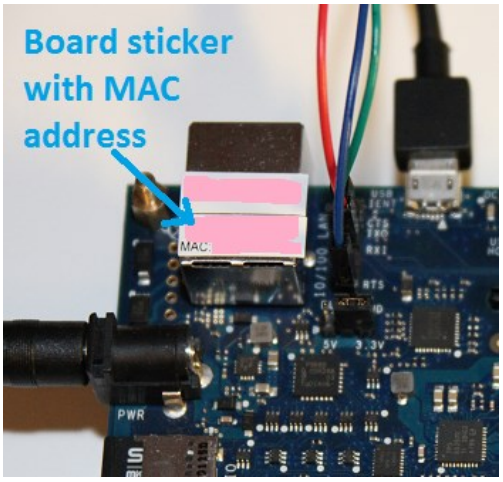


2. Create a flash image

   This menu item is available only when ‘Simplified way’ is used (Pack 1.0.4 is installed on SD card).

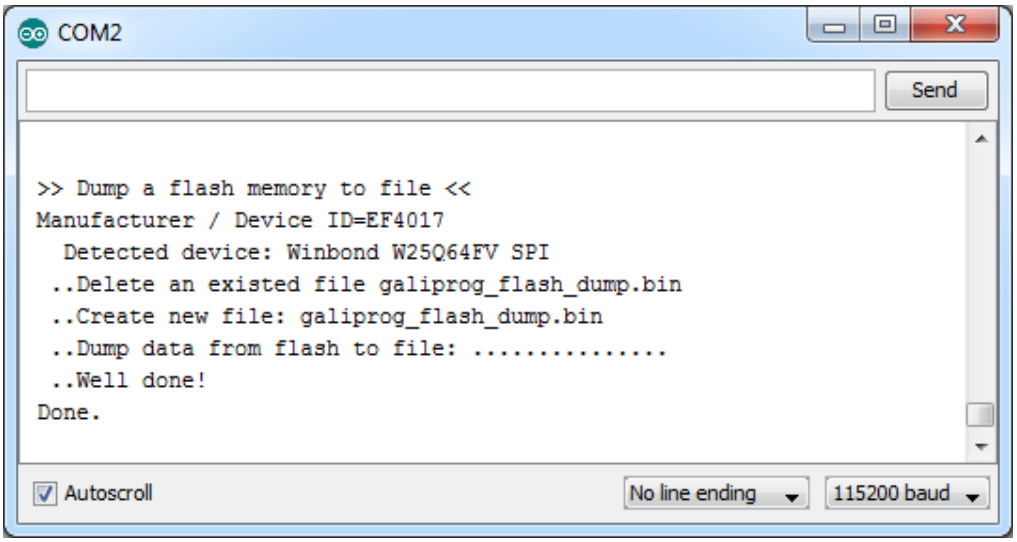
   Type ‘1’ or ‘2’ and push ‘Send’ button to select a board type.
   Enter MAC address shown on a board sticker and push ‘Send’.
   A file ‘galiprog_flash_write.bin’ will be created in the root of SD card.





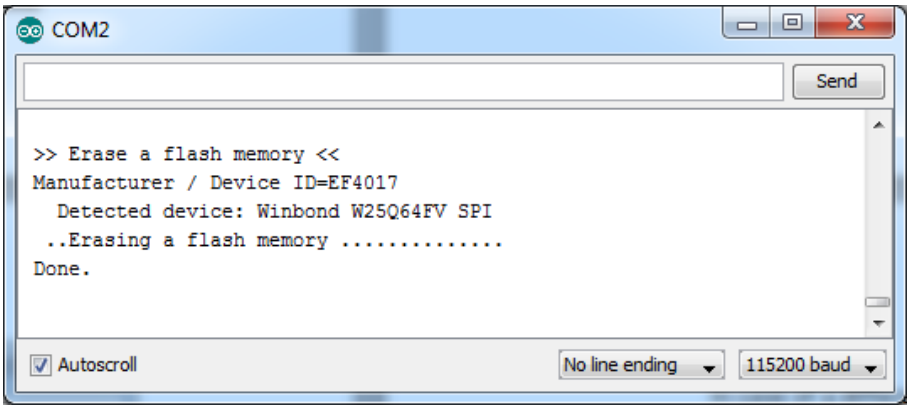
3. Dump a flash memory data to file

   This menu item allows to read all data (8 Megabytes) from SPI flash memory to a file with name ‘galiprog_flash_dump.bin’ (located in the root of SD card).



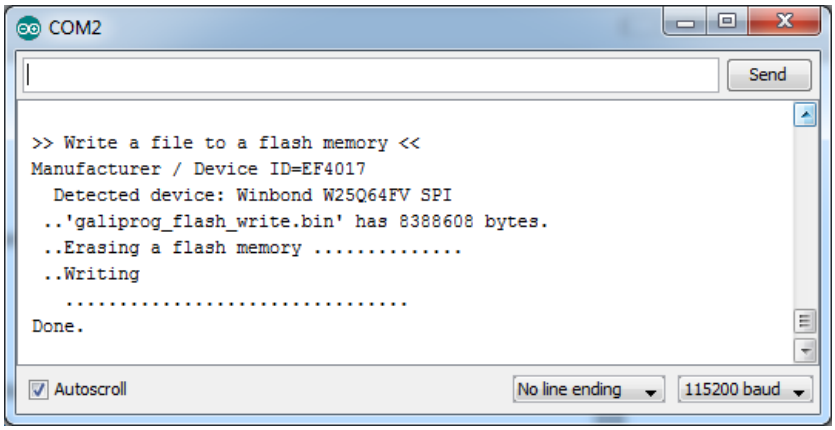
4. Erase a flash memory

   This menu item erases all SPI flash memory (fill it by 0xFF).



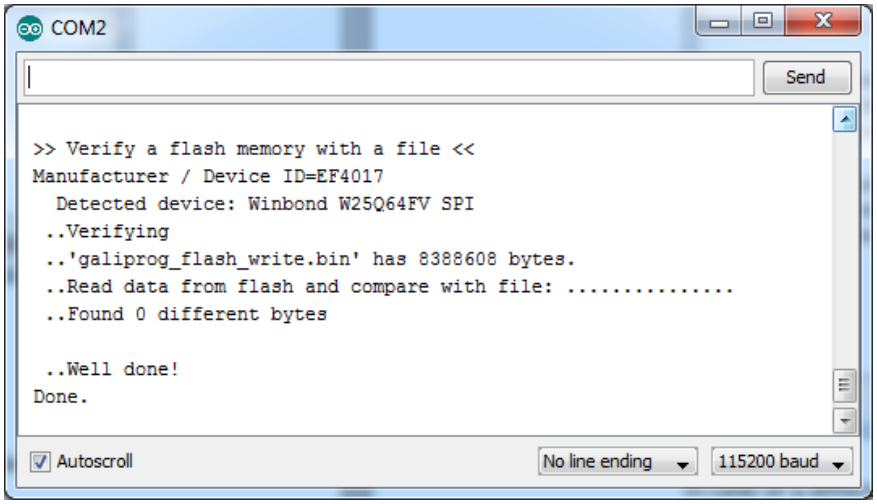
5. Write a file to a flash memory

   This menu item erases all SPI flash memory (fill it by 0xFF) and writes a data from a file with name ‘galiprog_flash_write.bin’ to a flash memory.

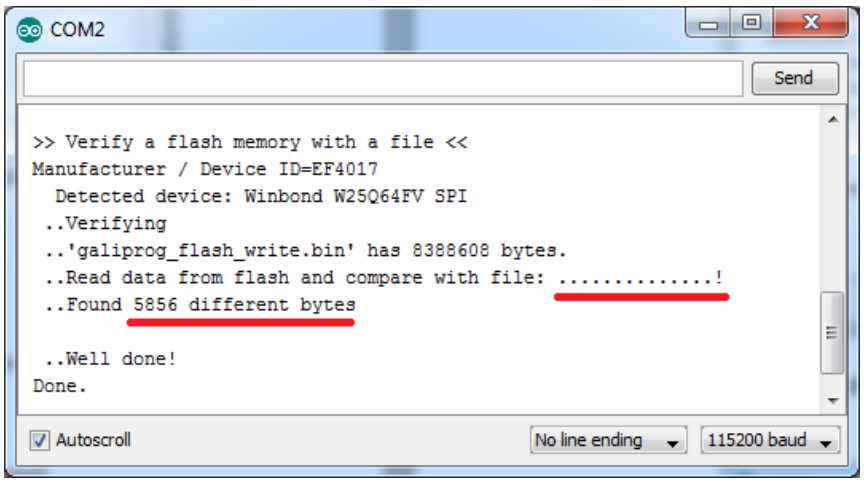


6. Verify a flash memory with a file

   This menu item reads all SPI flash memory and compares with a data from a file with name ‘galiprog_flash_write.bin’.

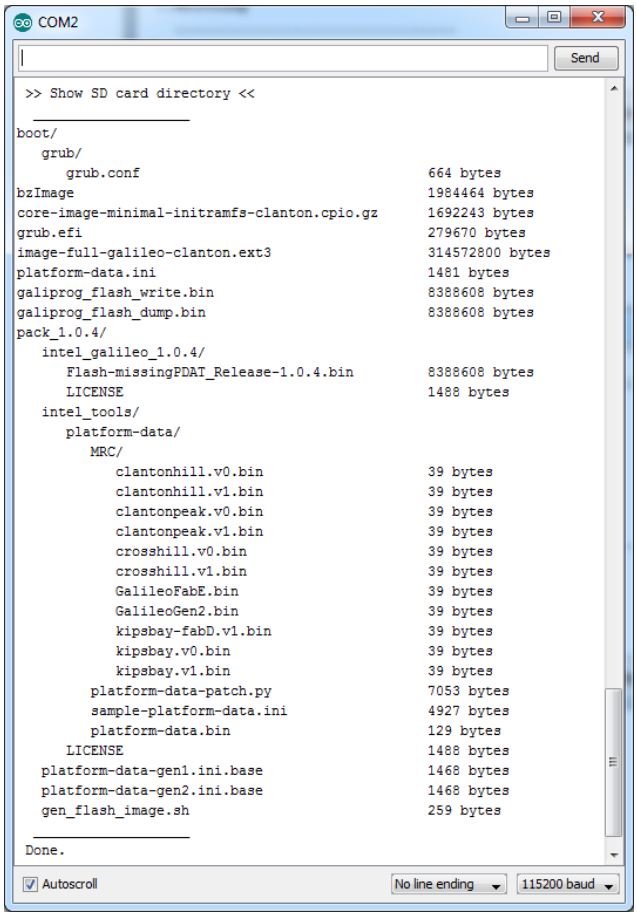


   In case of a difference between the file and flash memory the following information will be shown:



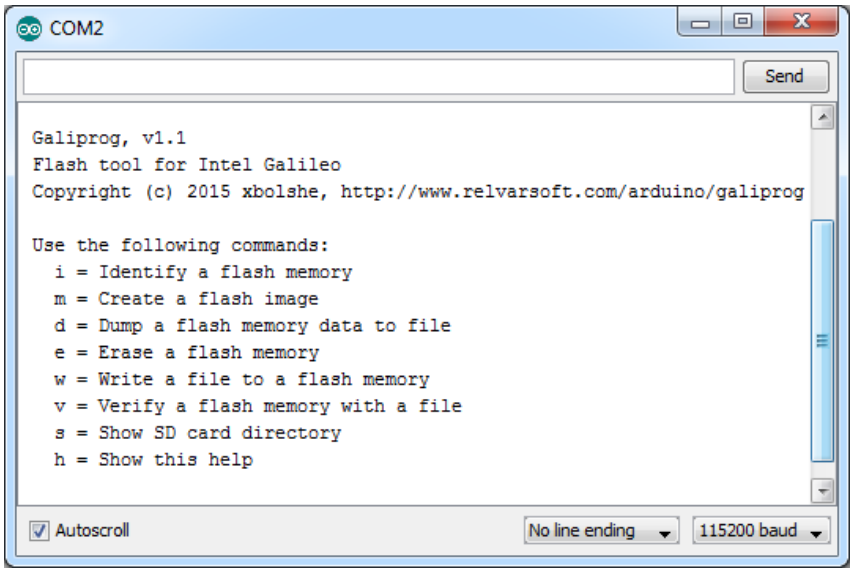
7. Show SD card directory

   This menu item prints a current list of files on SD card.



8. Show this help

   This menu item shows help screen like shown below:



8. Questions

1) What I need to execute to restore broken image in SPI flash memory?

Answer:

- Identify a flash memory
- Dump a flash memory data to file (optional)
- Write a file to a flash memory
- Verify a flash memory with a file

2) How to check a stability of data read/write?

Answer:

Use «Verify a flash memory with a file» 5 times.

If a difference is the same all the times, then a processing is stable. You may write a data to flash memory.

If a difference (in bytes) is not the same even one time compare with others, DO NOT WRITE a data to SPI flash memory! Need to fix a reason of this problem before writing a data.

3) More questions or comments? Write me e-mail.

Annex 1. About Pack 1.0.4

Pack 1.0.4 contains «Flash-missingPDAT_Release-1.0.4.bin» (original source: https://communities.intel.com/docs/DOC-22226) and a part of «spi-flash-tools-v1.0.1» (original source: https://downloadcenter.intel.com/Detail_Desc.aspx?DwnldID=23197) under the following license:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
