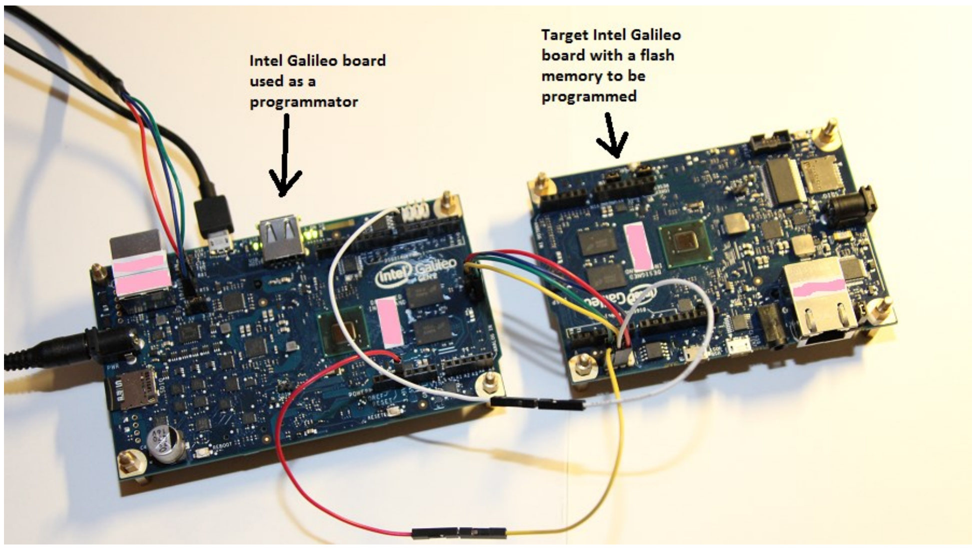
GaliProg v1.2

10.10.2015
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.

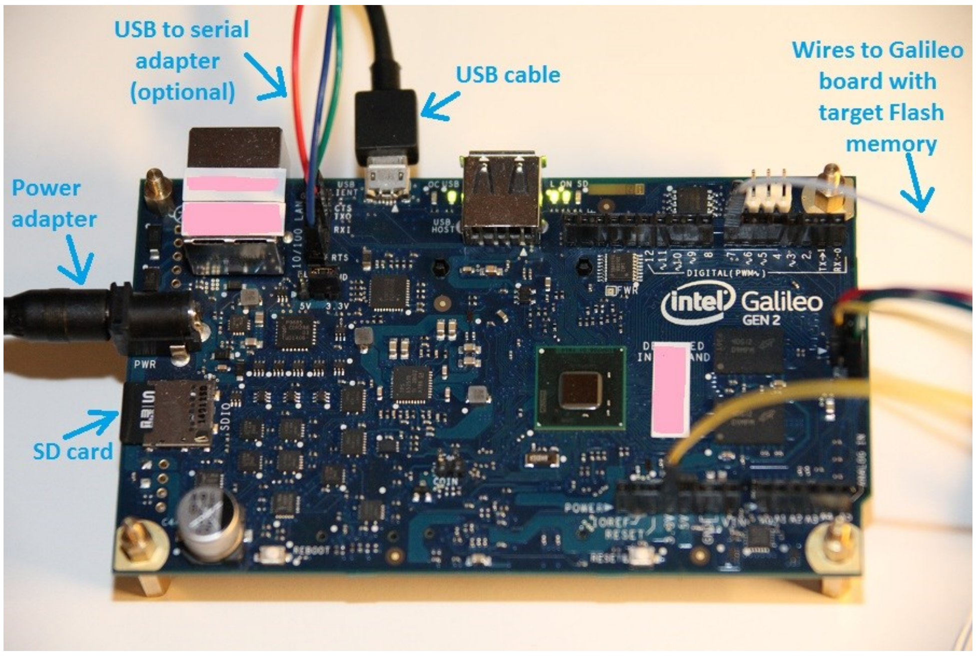
№	Programmator board	Target board with SPI on-board flash memory to be programmed	Was it tested?	Notes
1	Galileo Gen1	Galileo Gen1	no	
2	Galileo Gen1	Galileo Gen2	yes	
2	Galileo Gen2	Galileo Gen1	yes	
3	Galileo Gen2	Galileo Gen2	no	
4	Edison Arduino	Galileo Gen1	yes	Edison FW: need to use edison-image-ww05-15.zip
5	Edison Arduino	Galileo Gen2	yes	Edison FW: need to use edison-image-ww05-15.zip


Intel Galileo board used as a programmator
Target Intel Galileo board with a flash memory to be programmed

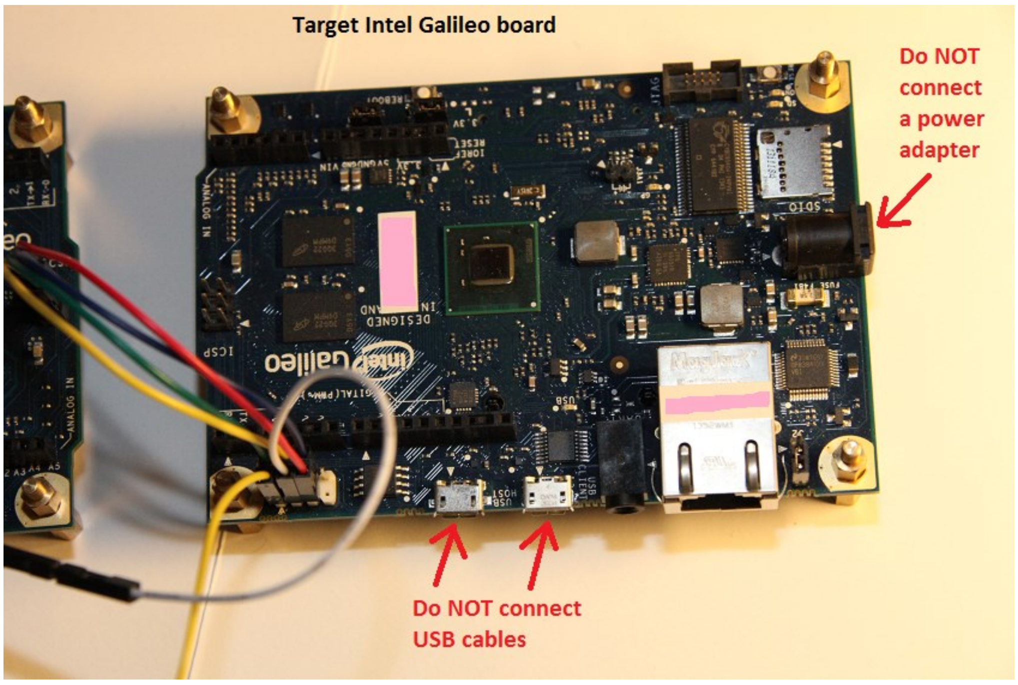
2. Required hardware

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

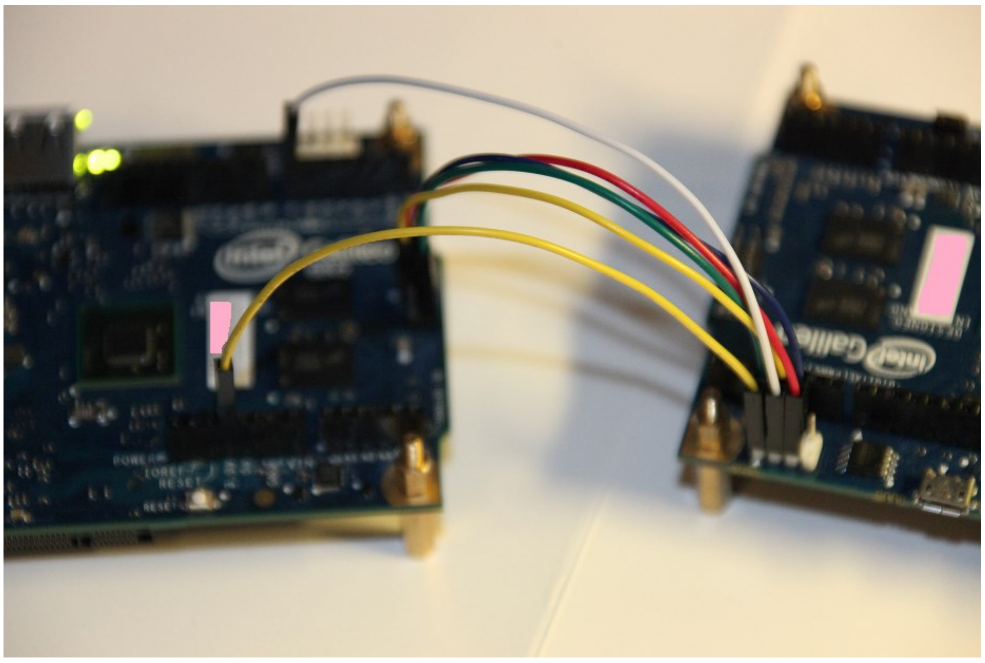
a) Properly worked Intel Galileo board with USB cable, micro SD card and power adapter. It will be used as used as a programmator.


USB to serial adapter (optional)
USB cable
Wires to Galileo board with target Flash memory
Power adapter
SD card

b) Galileo board with target Flash memory


Target Intel Galileo board
Do NOT connect a power adapter
Do NOT connect USB cables

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 for Intel Galileo board
Link to download: https://communities.intel.com/docs/DOC-22226

Intel Arduino Software (IDE) 1.6.4 for Intel Edison board
Links to download:
https://software.intel.com/iot/downloads
http://downloadmirror.intel.com/25028/eng/iotdk_win_installer.exe

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

Firmware for Edison Arduino board
Because of SPI bus problems the following Edison firmware will not work:
edison-image-ww18-15.zip
edison-image-ww25.5-15.zip

Please use this firmware: edison-image-ww05-15.zip
Link to download: http://downloadmirror.intel.com/24909/eng/edison-image-ww05-15.zip
Use Flash Tool Lite: https://software.intel.com/ru-ru/iot/hardware/edison/downloads

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 1.0.4v2
   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'.

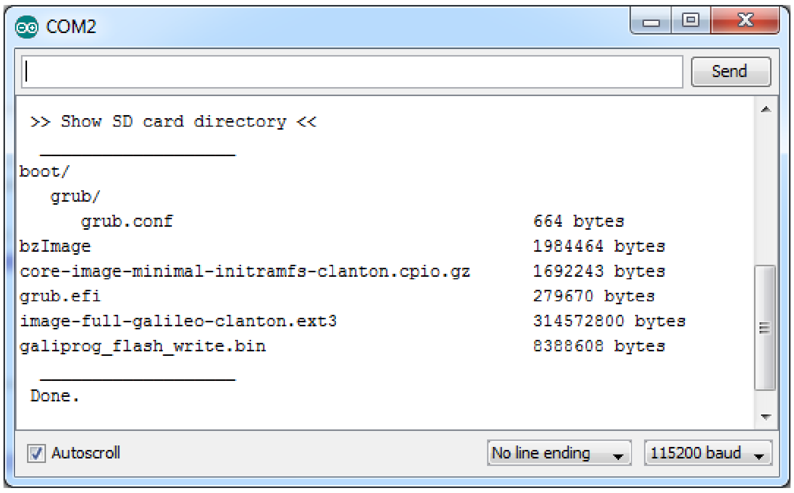
d) Galiprog (galiprog.ino)
This is a flash programing tool.

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

4. Prepare a data on SD card

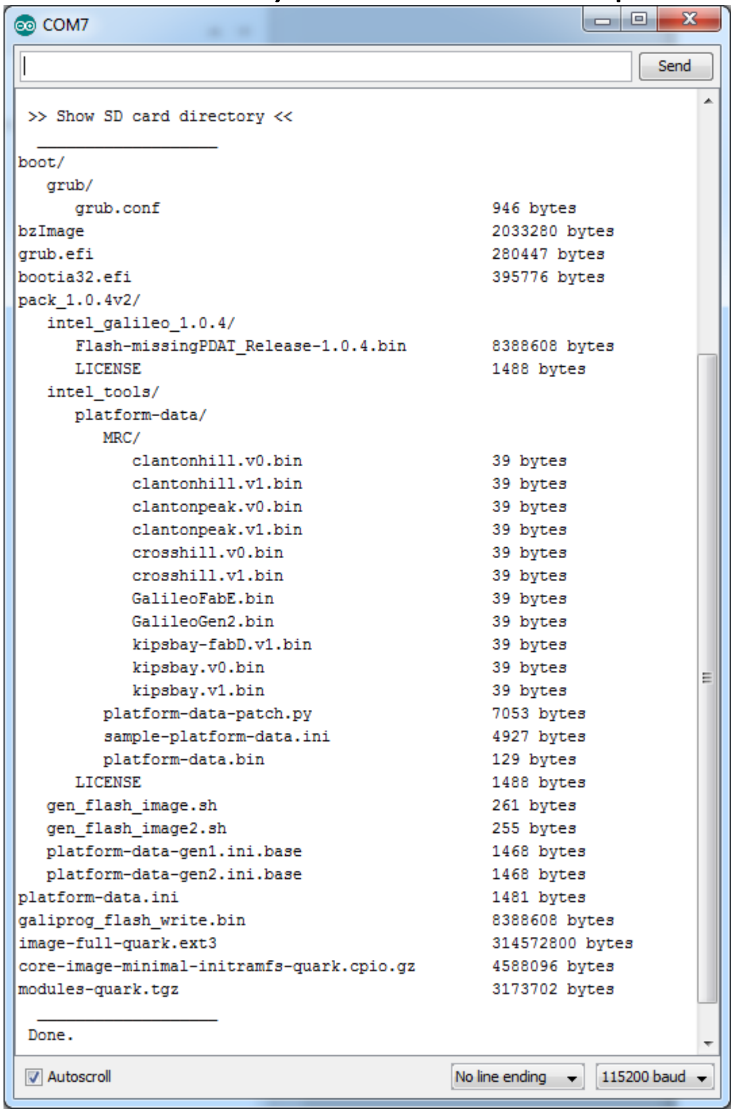
a) Format SD card
b) Unpack SD-Card Linux Image to the root of SD card
c) Copy 'Flash+PlatformData.bin' as 'galiprog_flash_write.bin', if you selected Official way.

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



```
>> Show SD card directory <<

 ________________
boot/
   grub/
      grub.conf                                  664 bytes
bzImage                                          1984464 bytes
core-image-minimal-initramfs-clanton.cpio.gz     1692243 bytes
grub.efi                                         279670 bytes
image-full-galileo-clanton.ext3                  314572800 bytes
galiprog_flash_write.bin                         8388608 bytes

 ________________
 Done.
```

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



```
>> Show SD card directory <<

 ________________
boot/
   grub/
      grub.conf                                  946 bytes
bzImage                                          2033280 bytes
grub.efi                                         280447 bytes
bootia32.efi                                     395776 bytes
pack_1.0.4v2/
   intel_galileo_1.0.4/
      Flash-missingPDAT_Release-1.0.4.bin        8388608 bytes
      LICENSE                                    1488 bytes
   intel_tools/
      platform-data/
         MRC/
            clantonhill.v0.bin                   39 bytes
            clantonhill.v1.bin                   39 bytes
            clantonpeak.v0.bin                   39 bytes
            clantonpeak.v1.bin                   39 bytes
            crosshill.v0.bin                     39 bytes
            crosshill.v1.bin                     39 bytes
            GalileoFabE.bin                      39 bytes
            GalileoGen2.bin                      39 bytes
            kipsbay-fabD.v1.bin                  39 bytes
            kipsbay.v0.bin                       39 bytes
            kipsbay.v1.bin                       39 bytes
         platform-data-patch.py                  7053 bytes
         sample-platform-data.ini                4927 bytes
         platform-data.bin                       129 bytes
      LICENSE                                    1488 bytes
   gen_flash_image.sh                            261 bytes
   gen_flash_image2.sh                           255 bytes
   platform-data-gen1.ini.base                   1468 bytes
   platform-data-gen2.ini.base                   1468 bytes
platform-data.ini                                1481 bytes
galiprog_flash_write.bin                         8388608 bytes
image-full-quark.ext3                            314572800 bytes
core-image-minimal-initramfs-quark.cpio.gz       4588096 bytes
modules-quark.tgz                                3173702 bytes

 ________________
 Done.
```

5. Connections between Galileo boards and hardware settings

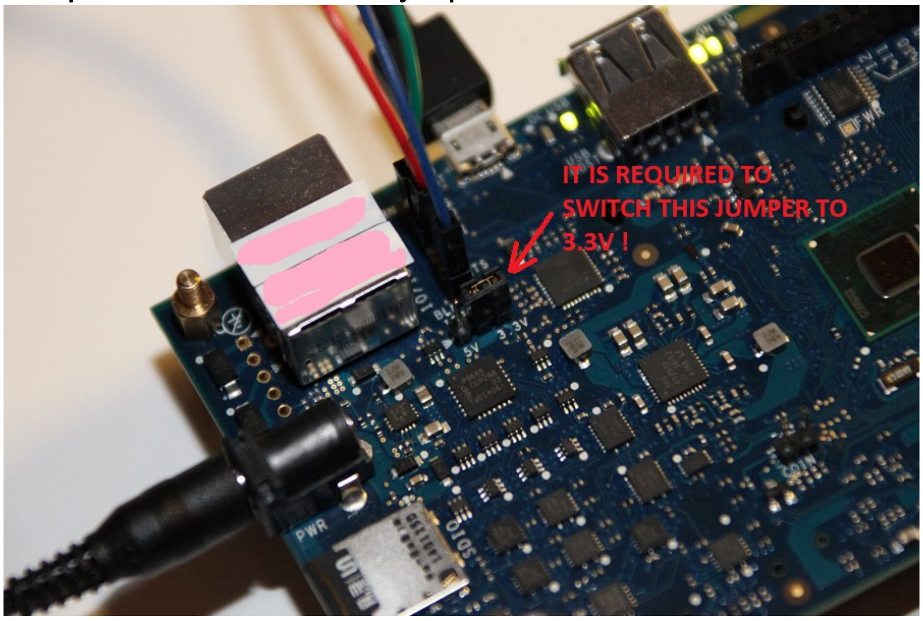
a) Configure Galileo board which works as a programmator

A SPI flash memory works with 3.3V lines. So, it is required to switch Galileo/Edison 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.

In case of Edison board need to connect pins 2 and 3 of J9.

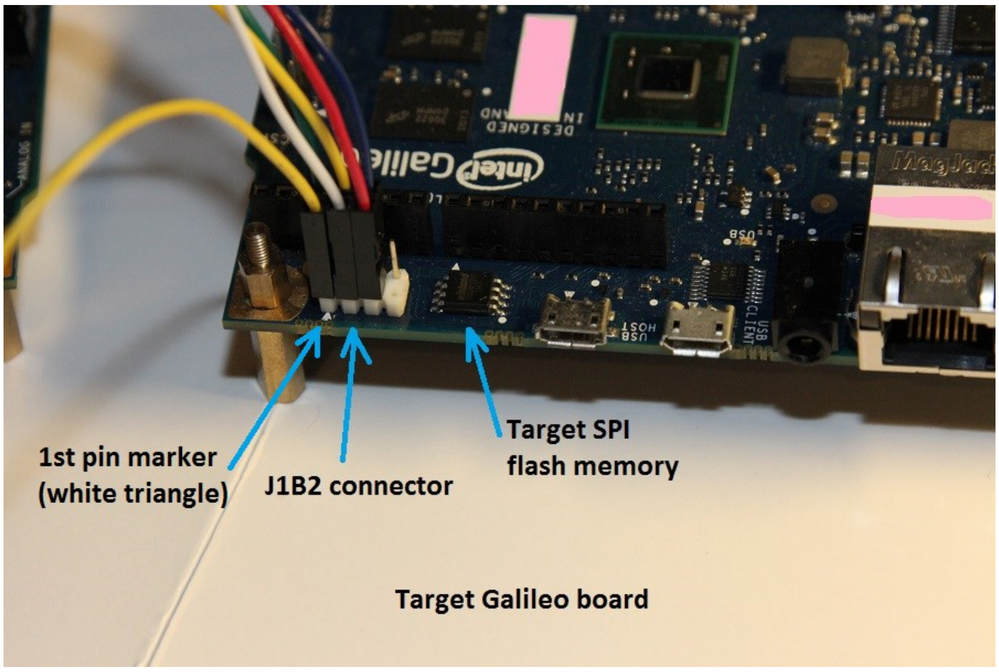
b) Wire connections

№	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

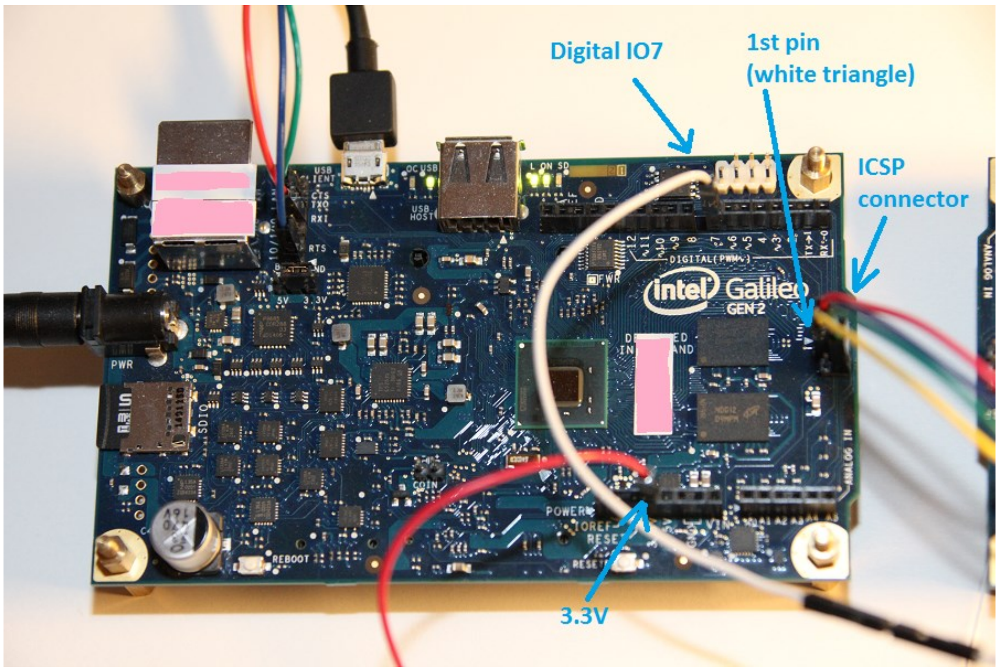
№	Galileo Gen1 board - programmator	Signal role	Galileo Gen2 board - target
1	3.3V	VCC	J1B2 – pin 1
2	Digital IO7	Slave selection	J1B2 – pin 3
3	ICSP – pin 4	MOSI, pull-up resistor 10kOhm	J1B2 – pin 6
4	ICSP – pin 1	MISO, pull-up resistor 10kOhm	J1B2 – pin 5
5	Digital IO13	SCK	J1B2 – pin 4
6	ICSP – pin 6	Ground	J1B2 – pin 2

№	Edison Arduino board - programmator	Signal role	Galileo Gen1/Gen2 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

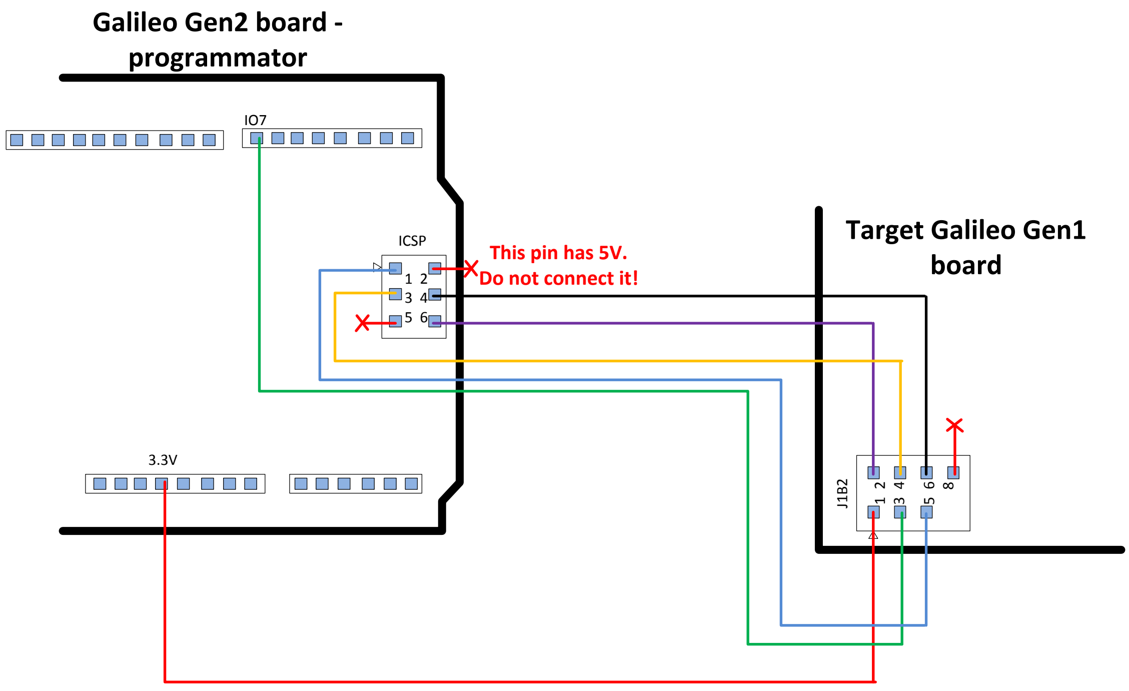
A location of J1B2 connector is shown on a picture below:

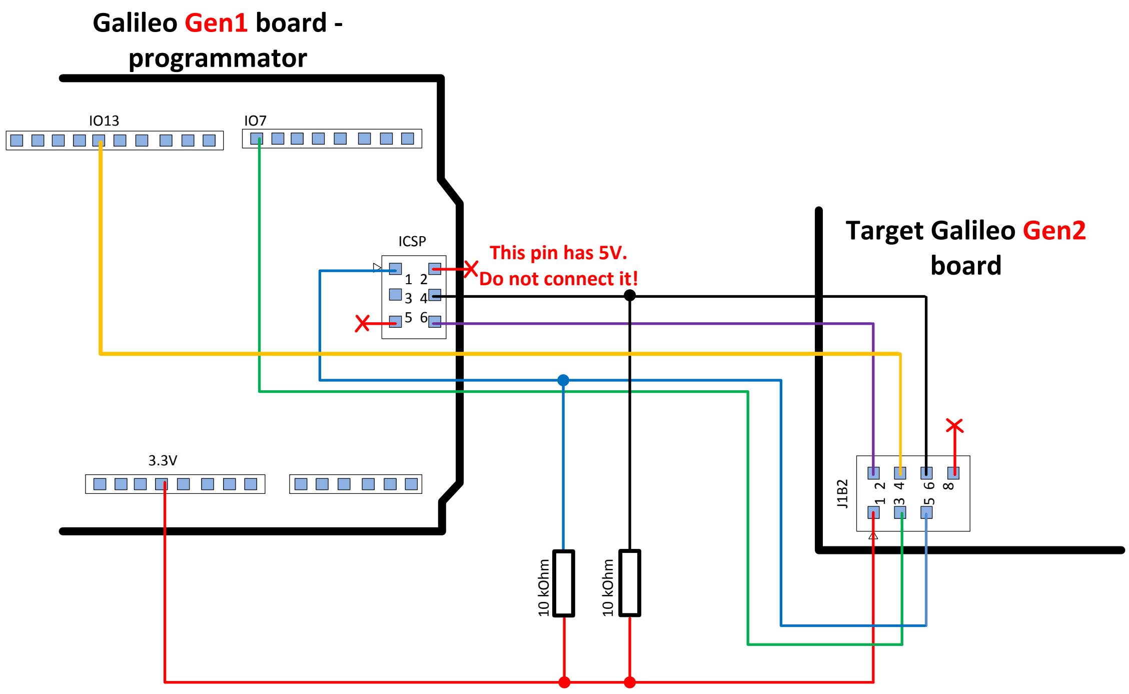


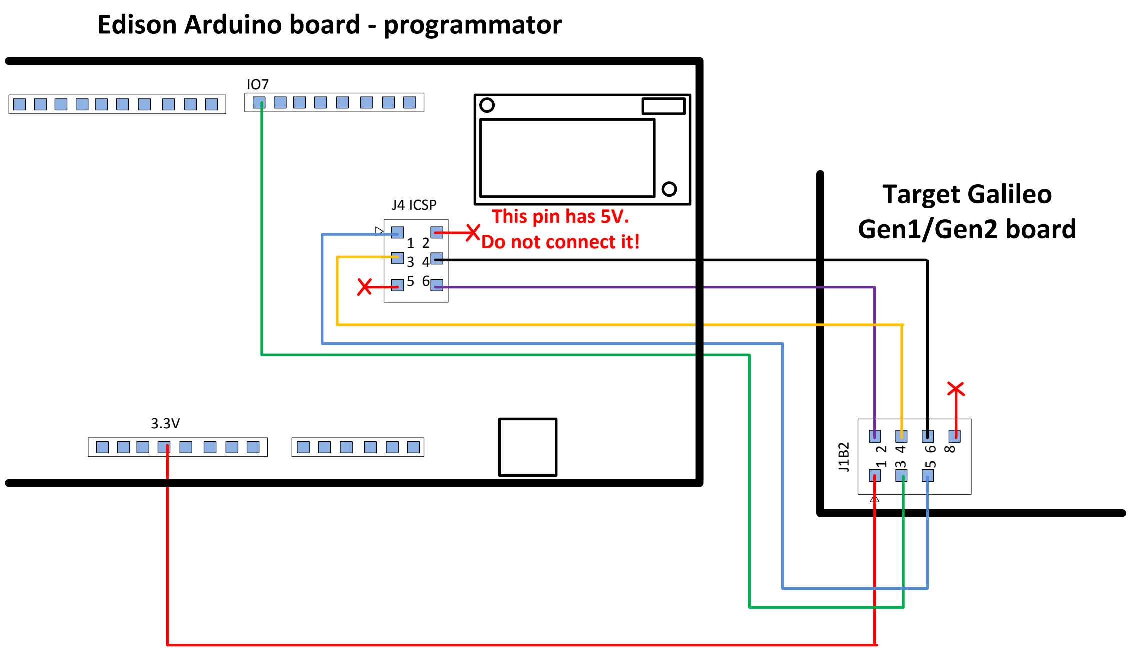
A location of ICSP connector is shown on a picture below:



Select a scheme below for your boards.

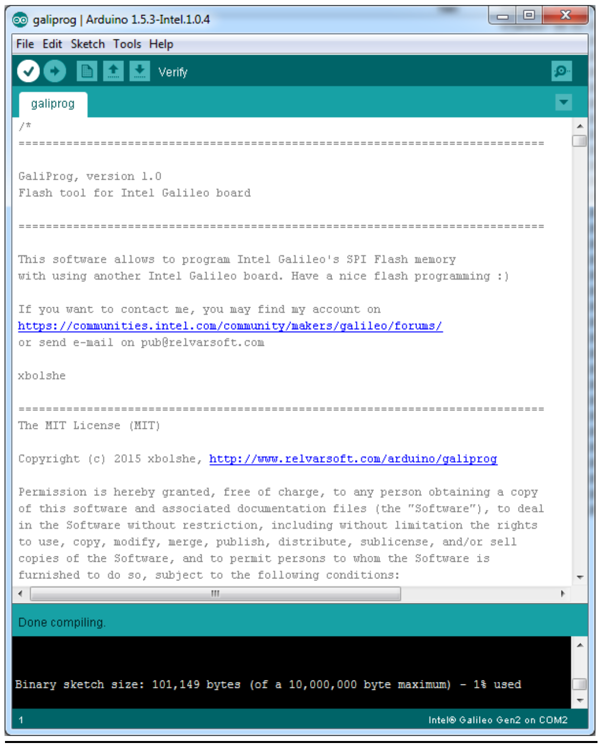






6. Compiling Galiprog sketch

- 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

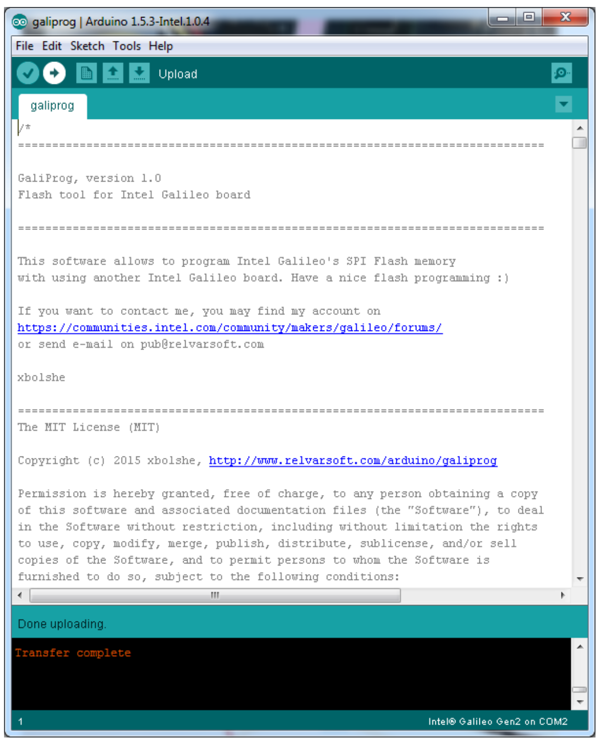

galiprog | Arduino 1.5.3-Intel.1.0.4
File Edit Sketch Tools Help
Verify
galiprog

```
/*
==============================================================================

GaliProg, version 1.0
Flash tool for Intel Galileo board

==============================================================================

This software allows to program Intel Galileo's SPI Flash memory
with using another Intel Galileo board. Have a nice flash programming :)

If you want to contact me, you may find my account on
https://communities.intel.com/community/makers/galileo/forums/
or send e-mail on pub@relvarsoft.com

xbolshe

==============================================================================
The MIT License (MIT)

Copyright (c) 2015 xbolshe, http://www.relvarsoft.com/arduino/galiprog

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
```

Done compiling.

```
Binary sketch size: 101,149 bytes (of a 10,000,000 byte maximum) - 1% used
```

1 Intel® Galileo Gen2 on COM2


galiprog | Arduino 1.5.3-Intel.1.0.4
File Edit Sketch Tools Help
Upload
galiprog

```
/*
==============================================================================

GaliProg, version 1.0
Flash tool for Intel Galileo board

==============================================================================

This software allows to program Intel Galileo's SPI Flash memory
with using another Intel Galileo board. Have a nice flash programming :)

If you want to contact me, you may find my account on
https://communities.intel.com/community/makers/galileo/forums/
or send e-mail on pub@relvarsoft.com

xbolshe

==============================================================================
The MIT License (MIT)

Copyright (c) 2015 xbolshe, http://www.relvarsoft.com/arduino/galiprog

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
```

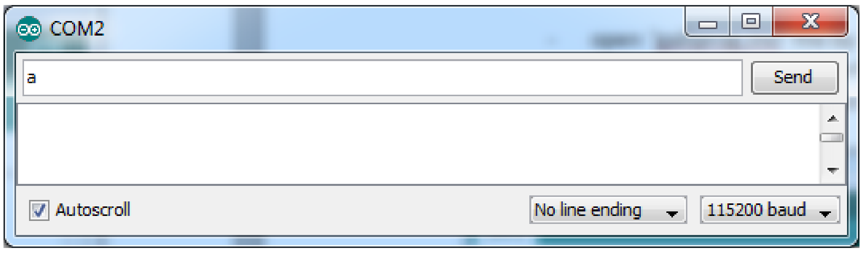
Done uploading.

```
Transfer complete
```

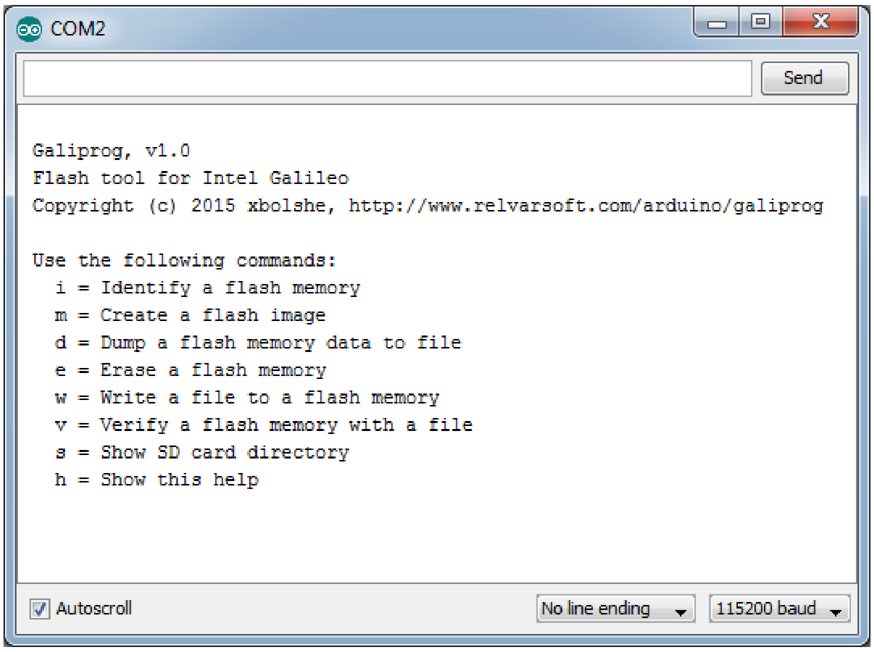
1 Intel® Galileo Gen2 on COM2

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

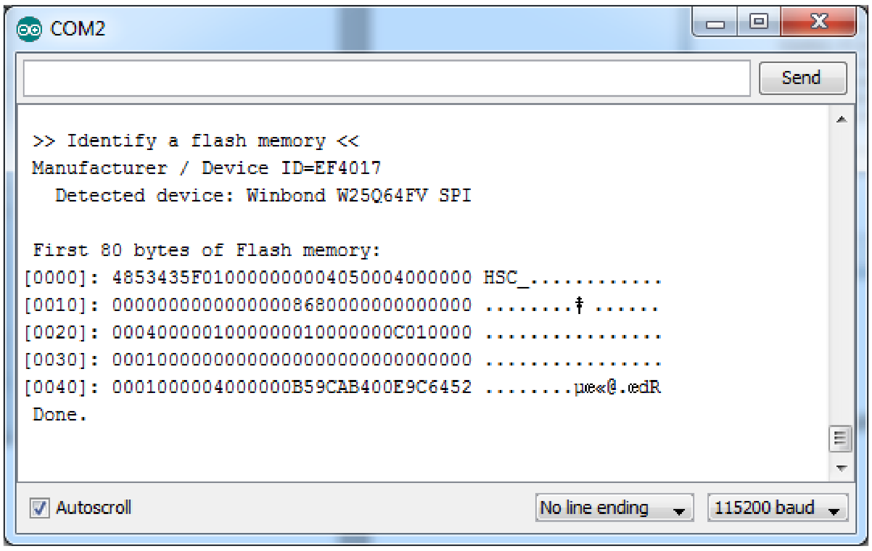


```
Galiprog, v1.0
Flash tool for Intel Galileo
Copyright (c) 2015 xbolshe, http://www.relvarsoft.com/arduino/galiprog

Use the following commands:
  i = Identify a flash memory
  m = Create a flash image
  d = Dump a flash memory data to file
  e = Erase a flash memory
  w = Write a file to a flash memory
  v = Verify a flash memory with a file
  s = Show SD card directory
  h = Show this help
```

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.

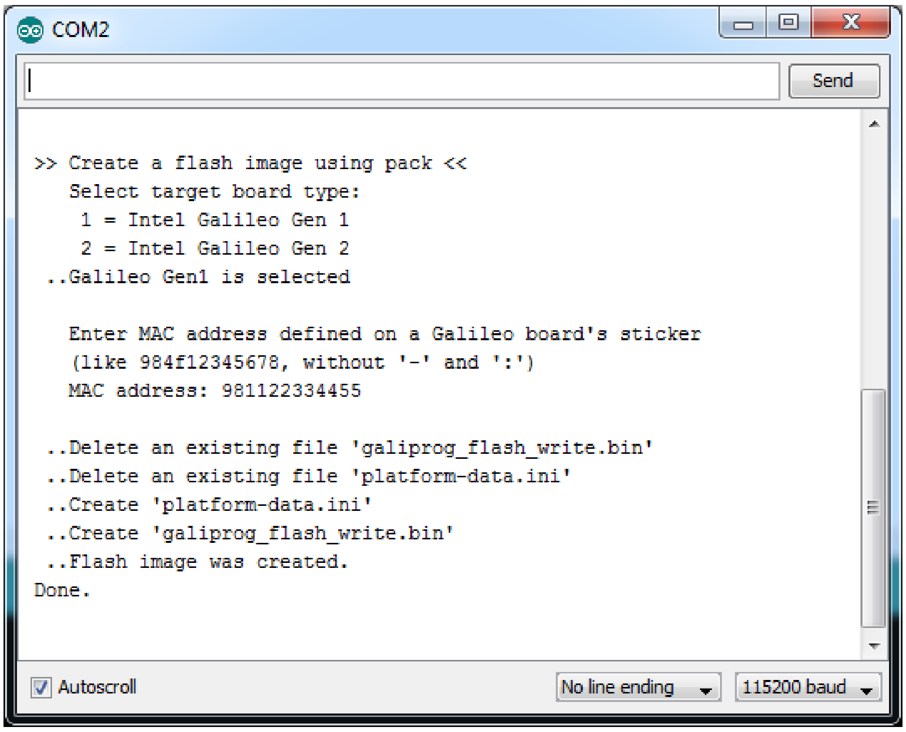


```
 >> Identify a flash memory <<
 Manufacturer / Device ID=EF4017
   Detected device: Winbond W25Q64FV SPI

 First 80 bytes of Flash memory:
[0000]: 4853435F010000000000405000400000 HSC_............
[0010]: 00000000000000000086800000000000 .........‡ ......
[0020]: 0004000001000000010000000C010000 ................
[0030]: 00010000000000000000000000000000 ................
[0040]: 0001000004000000B59CAB400E9C6452 ........µœ«@.œdR
 Done.
```

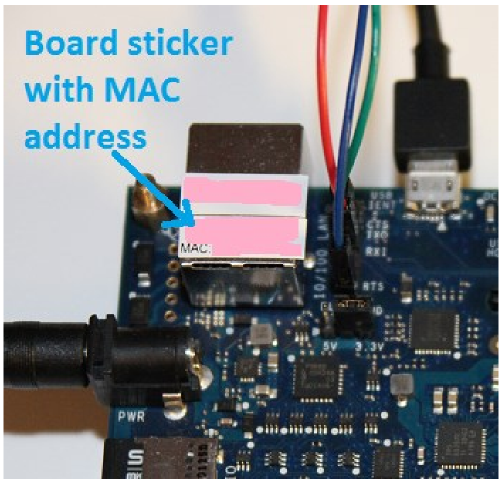
2. Create a flash image

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.

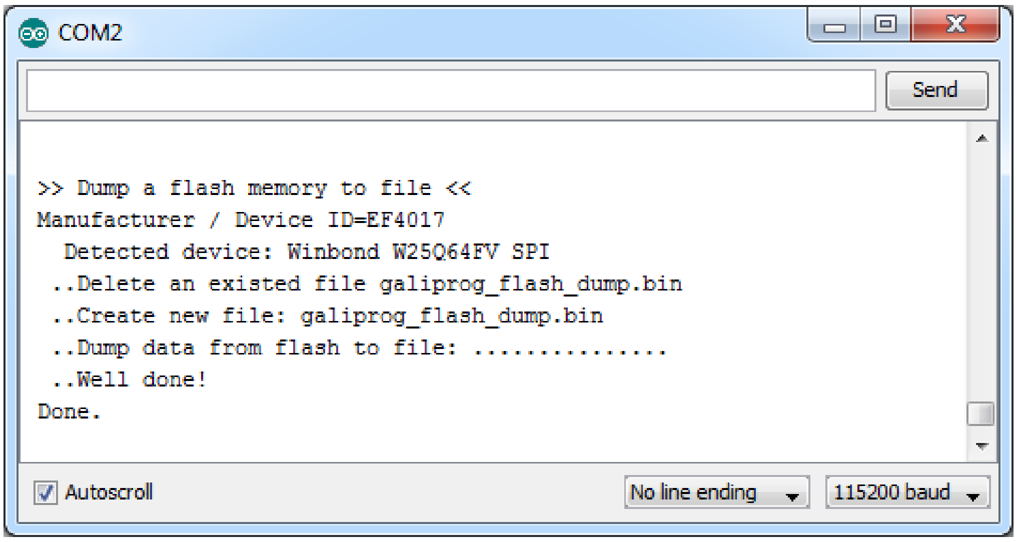


```
>> Create a flash image using pack <<
   Select target board type:
    1 = Intel Galileo Gen 1
    2 = Intel Galileo Gen 2
 ..Galileo Gen1 is selected

   Enter MAC address defined on a Galileo board's sticker
   (like 984f12345678, without '-' and ':')
   MAC address: 981122334455

 ..Delete an existing file 'galiprog_flash_write.bin'
 ..Delete an existing file 'platform-data.ini'
 ..Create 'platform-data.ini'
 ..Create 'galiprog_flash_write.bin'
 ..Flash image was created.
Done.
```



3. Dump a flash memory data to file

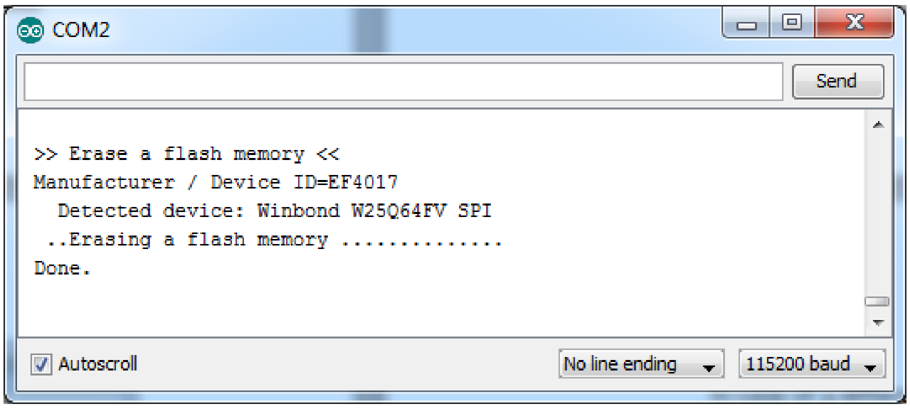
   This menu item allows to read all data (8 Megabytes) from SPI flash memory to a file with name 'galiprog_flash_dump.bin' (located in the root of SD card).



```
>> Dump a flash memory to file <<
Manufacturer / Device ID=EF4017
  Detected device: Winbond W25Q64FV SPI
 ..Delete an existed file galiprog_flash_dump.bin
 ..Create new file: galiprog_flash_dump.bin
 ..Dump data from flash to file: ...............
 ..Well done!
Done.
```

4. Erase a flash memory

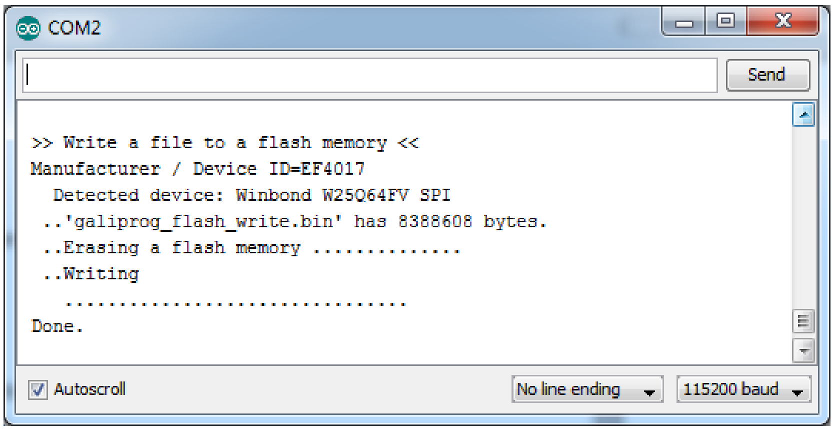
   This menu item erases all SPI flash memory (fill it by 0xFF).



```
>> Erase a flash memory <<
Manufacturer / Device ID=EF4017
   Detected device: Winbond W25Q64FV SPI
 ..Erasing a flash memory ..............
Done.
```

5. Write a file to a flash memory

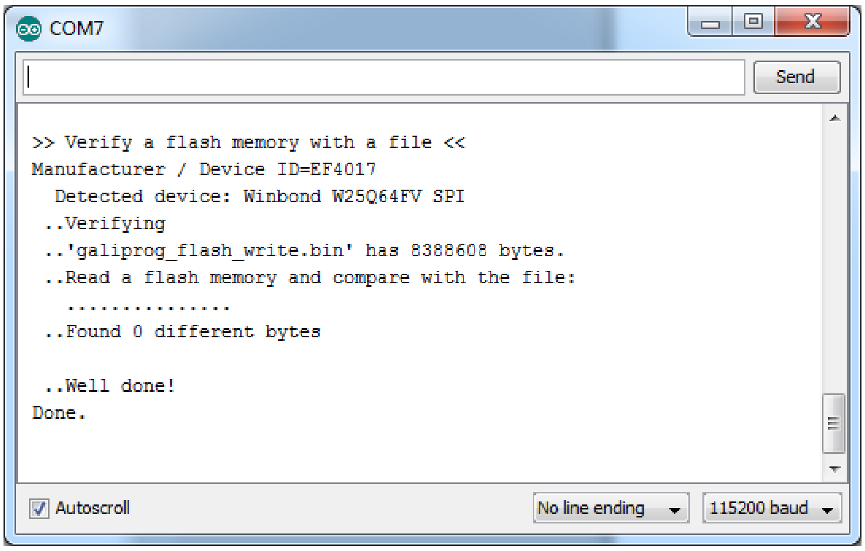
   This menu item erases all SPI flash memory (fill it by 0xFF) and writes a data from a file with name ‘galiprog_flash_write.bin’ to a flash memory.



```
>> Write a file to a flash memory <<
Manufacturer / Device ID=EF4017
   Detected device: Winbond W25Q64FV SPI
 ..'galiprog_flash_write.bin' has 8388608 bytes.
 ..Erasing a flash memory .............
 ..Writing
   ................................
Done.
```

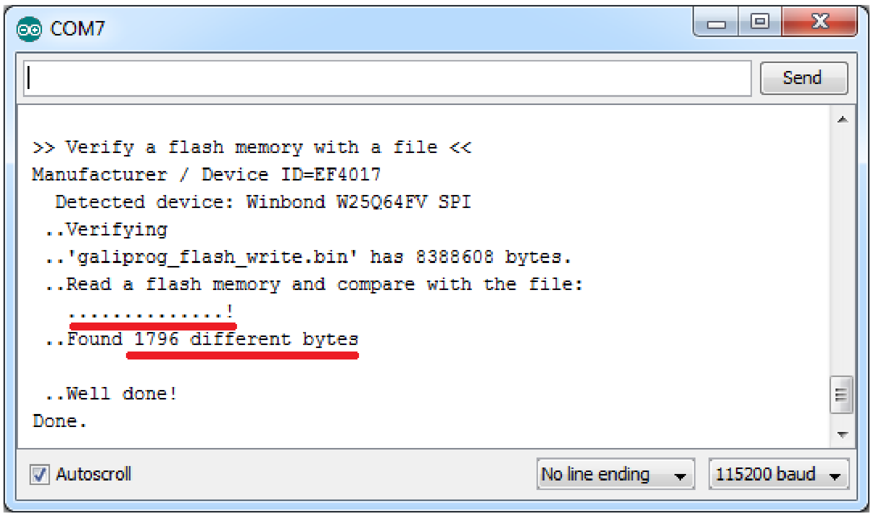
6. Verify a flash memory with a file

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



```
>> Verify a flash memory with a file <<
Manufacturer / Device ID=EF4017
  Detected device: Winbond W25Q64FV SPI
 ..Verifying
 ..'galiprog_flash_write.bin' has 8388608 bytes.
 ..Read a flash memory and compare with the file:
   ..............
 ..Found 0 different bytes

 ..Well done!
Done.
```

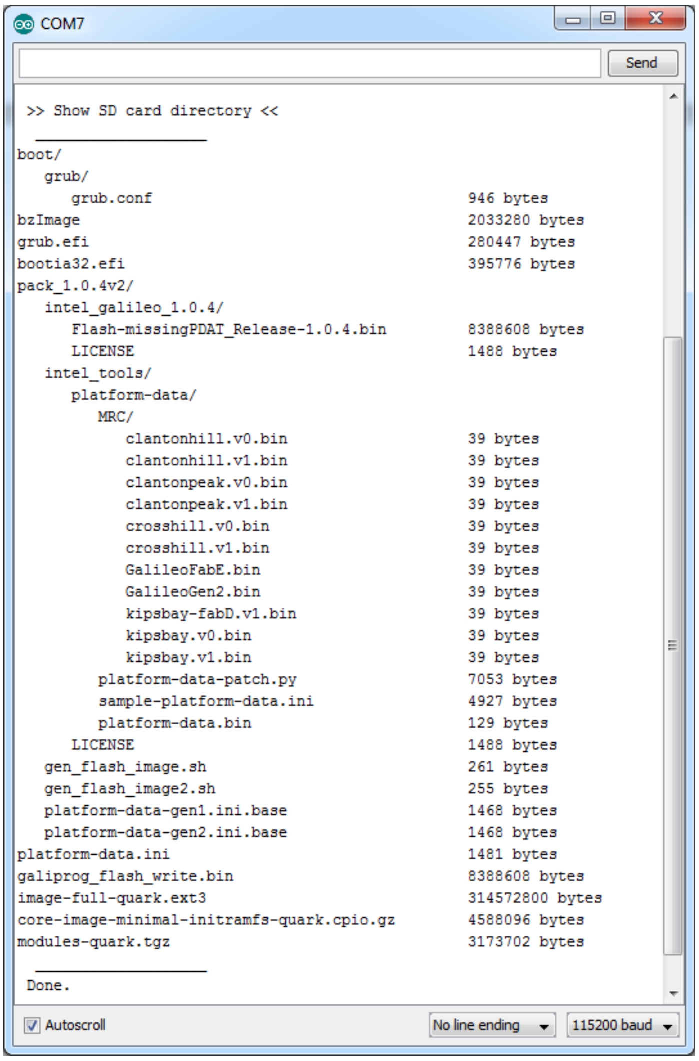
   In case of a difference between the file and flash memory the following information will be shown:



```
>> Verify a flash memory with a file <<
Manufacturer / Device ID=EF4017
  Detected device: Winbond W25Q64FV SPI
 ..Verifying
 ..'galiprog_flash_write.bin' has 8388608 bytes.
 ..Read a flash memory and compare with the file:
   ..............!
 ..Found 1796 different bytes

 ..Well done!
Done.
```

7. Show SD card directory

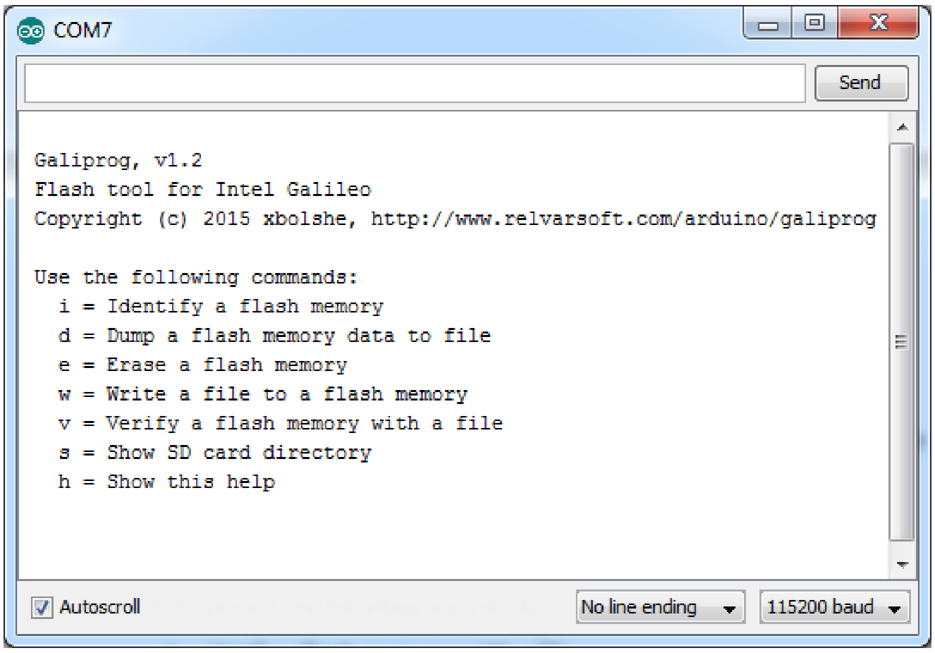
   This menu item prints a current list of files on SD card.



```
>> Show SD card directory <<
 __________________
boot/
   grub/
      grub.conf                                    946 bytes
bzImage                                            2033280 bytes
grub.efi                                           280447 bytes
bootia32.efi                                       395776 bytes
pack_1.0.4v2/
   intel_galileo_1.0.4/
      Flash-missingPDAT_Release-1.0.4.bin          8388608 bytes
      LICENSE                                      1488 bytes
   intel_tools/
      platform-data/
         MRC/
            clantonhill.v0.bin                     39 bytes
            clantonhill.v1.bin                     39 bytes
            clantonpeak.v0.bin                     39 bytes
            clantonpeak.v1.bin                     39 bytes
            crosshill.v0.bin                       39 bytes
            crosshill.v1.bin                       39 bytes
            GalileoFabE.bin                        39 bytes
            GalileoGen2.bin                        39 bytes
            kipsbay-fabD.v1.bin                    39 bytes
            kipsbay.v0.bin                         39 bytes
            kipsbay.v1.bin                         39 bytes
         platform-data-patch.py                    7053 bytes
         sample-platform-data.ini                  4927 bytes
         platform-data.bin                         129 bytes
      LICENSE                                      1488 bytes
   gen_flash_image.sh                              261 bytes
   gen_flash_image2.sh                             255 bytes
   platform-data-gen1.ini.base                     1468 bytes
   platform-data-gen2.ini.base                     1468 bytes
platform-data.ini                                  1481 bytes
galiprog_flash_write.bin                           8388608 bytes
image-full-quark.ext3                              314572800 bytes
core-image-minimal-initramfs-quark.cpio.gz         4588096 bytes
modules-quark.tgz                                  3173702 bytes
 __________________
 Done.
```

8. Show this help

This menu item shows help screen like shown below:



```
Galiprog, v1.2
Flash tool for Intel Galileo
Copyright (c) 2015 xbolshe, http://www.relvarsoft.com/arduino/galiprog

Use the following commands:
  i = Identify a flash memory
  d = Dump a flash memory data to file
  e = Erase a flash memory
  w = Write a file to a flash memory
  v = Verify a flash memory with a file
  s = Show SD card directory
  h = Show this help
```

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: pub@relvarsoft.com

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: