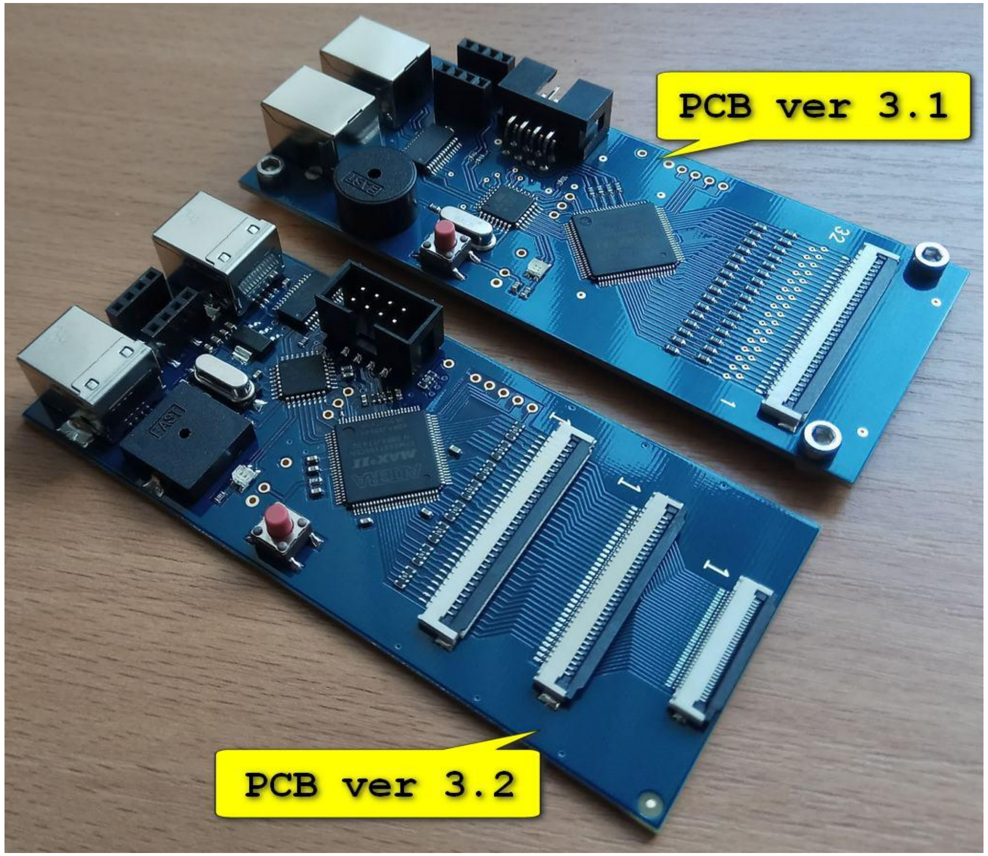
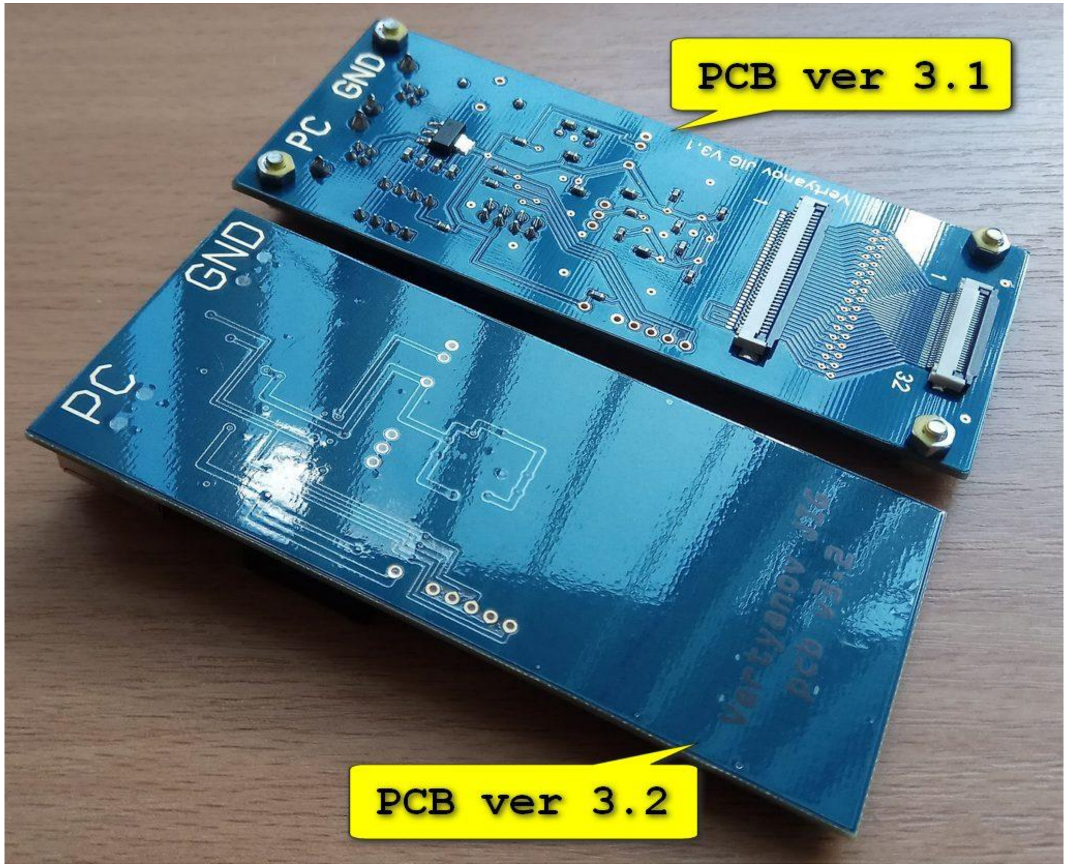
INSTRUCTION FOR WORKING WITH THE VERTYANOV JIG PROGRAMMER VER.3

redaction from 14/10/2018





Features of the programmer:

1 Programming NPCE288N / NPCE388N through the keyboard connector and on the external adapter.
2 Programming KB9010 / 9012/9016/9022/9028/902
3 Program EEPROM LCD panels (EDID matrices).
4 Programming is available IT8386E - 192KB IT8580 / 8585/8586/8587/8985 / 8987- at 128KB. Other ITEs simply were not checked.
5 Programming is available for MEC1609 / 1619 / 1633L / 5075/5085. The others were not checked by me.
6 Can check in AUTONOM mode keyboard for laptops pin spacing of 1mm / 05mm / 08mm.
(sound signaling of button presses and errors (buzzer).)
7 You can check the keyboards through the shell on the PC.

8 EC from Explore EPF011 / 021/035/036/037 using the interface module.

For work you need to have two USB B cords - square like on printers, scanners, etc. One cable to connect to the PC, and the other to connect the ground (mass) of the programmer and the laptop card. The shell works on all OS Windows XP / 7/8 / 8.1 / 10 32/64 bit. We do not have any activations. You buy the programmer and download all available software in free access. Free support on well-known forums.

At the moment there are two revisions of the programmer of the third version. Revision 3.1 and 3.2 are absolutely identical in their capabilities, except for some details. Revision 3.2, assembled completely on one side of the PCB, all components in smd execution. On the underside of the programmer is an insulating film. All other differences are visible in the pictures above. Both revisions can be in both a blue and a black mask.

1. Installing the drivers.

The very first and important point! MANDATORY it is necessary to check / install drivers independently, and not rely on the WINDOWS service. On this depends further correct work.

Connect the programmer to the PC and two new devices appear in the device manager, for which you need to install the drivers before you start. For communication with the PC, a USB-UART FT232RL chip from FTDI is used. The drivers for it are in the archive along with the program's shell. You can also download a more recent version of the drivers on the FTDI website by link http://www.ftdichip.com/Drivers/D2XX.htm Deeper into the details and subtleties of driver installation will not, all this is described in more detail in the FTDI documents on the page http://www.ftdichip.com/Support/Documents/InstallGuides.htm

Windows 10 Installation Guide

Windows 8 Installation Guide

Windows 7 Installation Guide

2. Configure program files for the shell version 6.5 and higher.

In one folder with the program are the following files (EC = embedded controller):
- MB connection ENE.txt (settings for motherboards with EC ENE)
- MB connection ITE.txt (settings for motherboards with EC ITE)
- MB connection NUVOTON.txt (settings DIFFERENT from "standard." Read below)
- MEC settings.txt (settings under MEC, FLASH memory size)
- NUVOTON.vss (service settings for working with NUVOTON EC on JTAG)
- Vertyanov JIG.exe
- UPDATE.txt (empty file, read below in the software update section)

Structure of the setup files for ENE, ITE, NUVOTON.

In each line of .txt files there is a setting for one motherboard. One setting is one line and no transfer to a new line is allowed. Different manufacturers of multi-controller (EC), different control signals for working with their built-in flash memory and entering the in-circuit programming mode.

ENE - **KSO3, KSI4, KSI5, KSI6, KSI7**

ITE - **PD0-PD7, BUSY, STB, AFD, INIT, SLIN, KSI4, KSI5**

NUVOTON - **JENK, TCK, TMS, TDI, TDO**

Since when working with these manufacturers EC connections occur through the keyboard connector, the following settings are needed for easy connection and use of various FFC (FFC - **F**lat **F**lexible **C**able)

KB_pins number of pins for keyboard connector (24/25/26/28/30/32/34)

LEFT or **RIGHT**. If, when installing the FFC from itself to the connector, the first contact on the left, then in the line we write LEFT. In another case- RIGHT. **UP** or **DOWN** - the contacts in the keyboard connector are located at the top or bottom. If there is no UP in the settings line for MB, then the default is DOWN.

05MM or **08MM** or **1MM** - the step between the pins in the keyboard connector. **This setting is relevant for revision 3.1**

If such parameters are present in the setup line, then you can use any FFC, with the same number of contacts, and with fewer contacts than in the keyboard connector. The step between the contacts in the FFC and in the connector, of course, should be one. Thus, in 99% of cases, you can manage three FFCs for all motherboards. It is enough to have FFC for 24 contacts with a pitch of 1mm, 26 contacts with a pitch of 0.8mm and for 30 contacts with a pitch of 0.5mm. The shell automatically configures the signals for the desired pins, based on the input parameters for MB and FFC and displays a picture- hint on the connection. If there are no parameters in the settings line about the number of contacts of the keyboard connector (KB_pins) and the orientation of the first output (LEFT / RIGHT), only FFC with the same number of contacts can be used.

Any settings in the MB line can be in any order, but I recommend sticking to the same style for all ECs. The number of the pin should be written immediately after the sign is equal, without spaces and the transfer of the settings to the next line is not allowed. You can add your own comments to the end of the line, but do not use the reserved words listed above KSO=, KSI=, TMS=, LEFT etc.

Examples of settings:

LA-7912P KB_pins=26 RIGHT UP 1MM **KSO3=4 KSI4=23 KSI7=26 KSI6=25 KSI5=24**

LA-C771P KB_pins=30 LEFT DOWN **JENK=7 TCK=19 TMS=20 TDI=21 TDO=10**

X555UJ KB_pins=24 LEFT DOWN **PD0=2 PD1=11 PD2=16 PD3=8 PD4=17 PD5=15 PD6=19 PD7=1 BUSY=5 STB=13 AFD=3 INIT=10 SLIN=12 KSI4=9 KSI5=7**

- the setting for X555UJ has been moved only for readability here

- DOWN setting is given for an example and it can be not written

For the EC of ITE, there is an additional setting for FLASH =. After the sign is equal to the size of FLASH memory in decimal. For example: FLASH = 56, FLASH = 192. If there is no configuration, the default FLASH size is 128KB.

3. Firmware update in the controller of the programmer.

The link to the latest version of the shell for working with the programmer, I post on the first page of my topics:

ONE and **TWO**

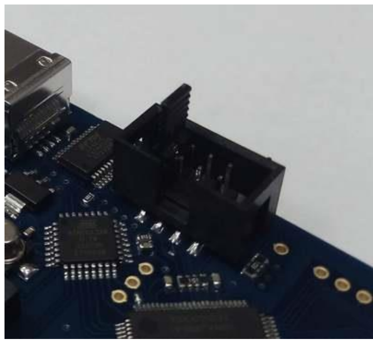
(the blue arrows in the picture are direct links to these forums and to the YouTube channel from the shell) Registration on these forums is optional, but it is better to ask questions on the application there, so I recommend registering. There is one important point in updating FW - disconnect all other devices from USB from the PC. This is especially true for devices on FTDI chips. If you forget it, it will be written in the log.

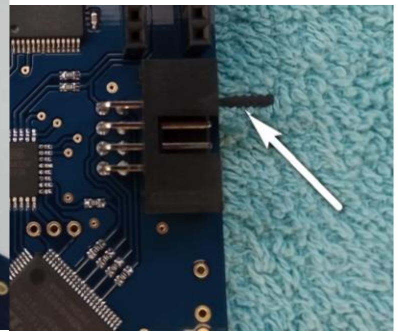
By running the shell and clicking on the Check connection button, the shell, if you need to update the firmware (FW) in the programmer, will prompt you to press the reset button. The reset button is located on the programmer. The FW of the programmer changes once, while you are working with this version of the shell. If you forget to do this before the first use of the new version, nothing terrible will happen - the shell will offer it to do it at the right time.

The second way is forcing the FW programmer to update. Start the shell and hold the reset button on the programmer. While holding the reset button, in the shell click Check connection. After the appearance of the window asking to press the reset button - release it. This method only works if there is a file named UPDATE.txt in the folder with the shell (it is always present by default for convenience). You do not need to update the FW of the programmer every time you start the shell "manic". If something does not work for you, it does not mean that it's the firmware of the programmer - look for the problem in another.

In the shell log everything will be written very simply, therefore, there is nothing more to comment here.

4. Offline checking of keyboards.





To enter the offline mode of checking the keyboards, without starting the shell on the PC, you must install the jumper on the 1 + 2 pins of the side connector of the programmer. You can power the programmer either from the PC or from any charger, such as a phone or tablet.

The programmer has 3 connectors:

- 32 contacts with 1 mm contact pitch.
- 34 contacts in 0.8 mm pitch.
- 32 contacts with a pitch of 0.5 mm.

In any of these connectors insert a ffc from the keyboard from 1 contact (on the board is signed). It is possible to connect from both the latter and the middle, but it is more advantageous from 1 because of the convenience of getting a clear contact between the connector and the keyboard ffc.

(When working with EC, we will always use the method of connecting the ffc from the first contact, so here it is better to do the same way to develop the right habit.)

The sequence of enabling the offline keypad check, I propose the following:

- Install the jumper into the socket of the programmer.
- Connect the keyboard for testing.
- Power the programmer from the PC or from the power supply.

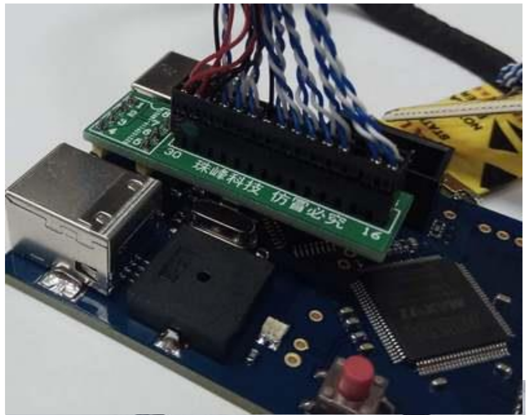
You can use another sequence but take for the rule - press the reset button on the programmer after all the connections. (The jumper is only checked when the power is applied or reset.) The jumper is not interrogated during operation.)

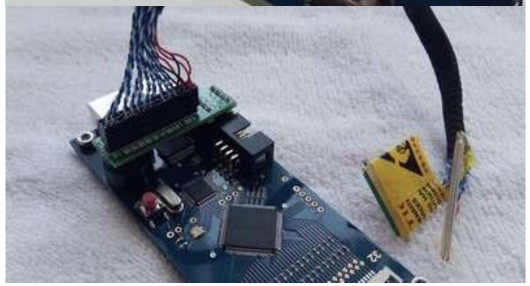
Further, in time with pressing the buttons on the keyboard, you will see a luminous red LED and a "tilim" beep. Remember that the pressed button is a closure of no more than two lines. Therefore, you will hear "tilim" only under this condition. If you press two or more buttons, or if you press the button, you will find a short circuit of more than two lines - you will hear "tadam". More precise information can be obtained by running the Keyboard test mode in the shell.

On some keyboards, the manufacturer has already made jumpers (often visible at the end of the loop), which close two to three pins on the ffc. The presence of such a jumper, at the first scanning pass by the programmer, will be indicated by a green LED and will not be considered in the testing. There may be a situation when a LED (in the CAPS or WIFI led button) with a low drop on the p-n junction will be received for the jumper - the Keyboard test mode in the shell will help to understand.

In most cases, offline mode is much easier and more convenient for quick testing of the performance of keyboards.

5. Read and write EDID LCD panels.

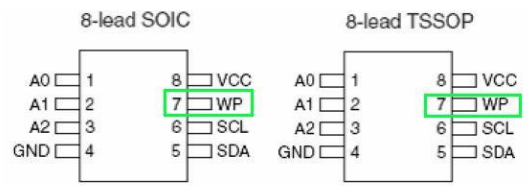




The picture shows the position of the EDID_cable for reading and writing EDID lcd panels. At the second end of the EDID cable are two connectors for 30 and 40 contacts. The cord is not designed to work with e-DP lcd!

On the lcd panels can be installed 2 types of EEPROM - analogue 24C02 and OTP_ROM. OTP_ROM is a one-time programmable memory that cannot be erased or modified in it. If you get this, you'll have to replace it with an ordinary one and then program it. Another difficulty is that the lcd board can be equipped with a firmware protection from the EEPROM 24C02 content upgrade. WP_pin name. If the 7 pin of the chip is connected to the log. 0 (ground) then it is possible to modify its contents.

If connected to log.1 - modification is prohibited. If you need to change the contents, then you need to provide log.0 on the 7th EEPROM pin. Reading the WP pin is not affected. The programmer's shell, in case of an unsuccessful write to the EEPROM, will write a log about the problem.

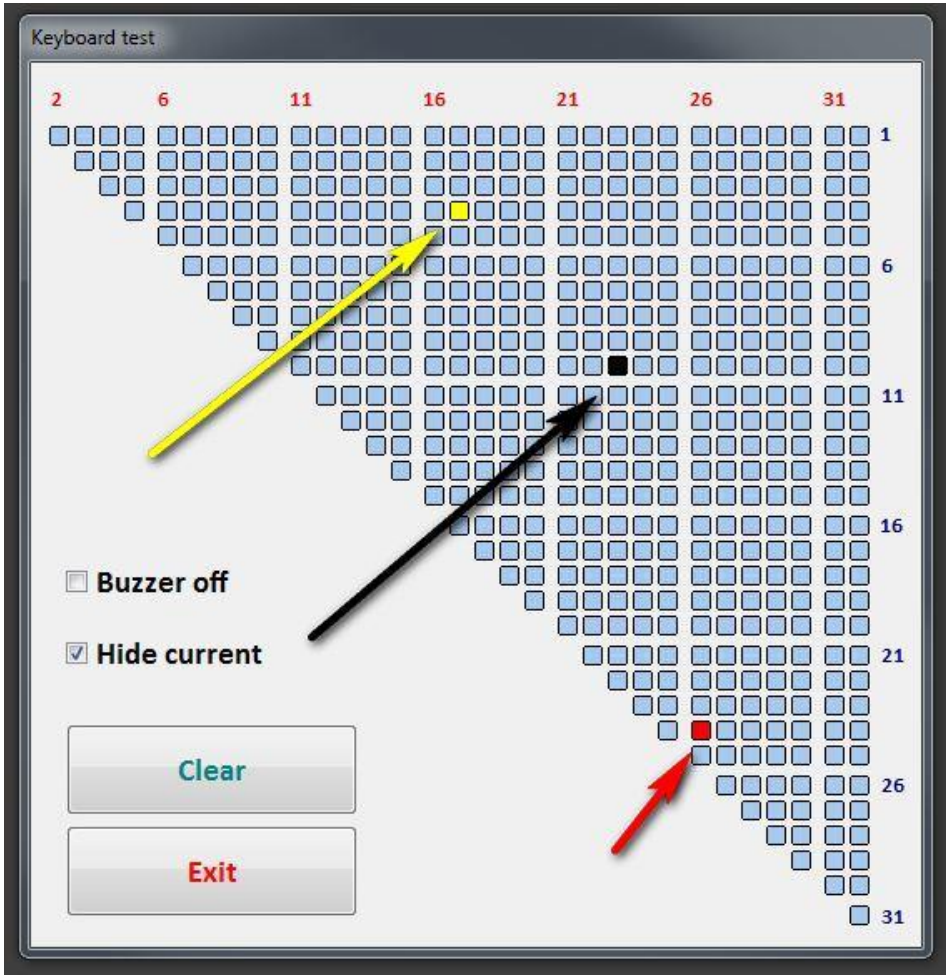


7	WP	I	Write Protect **WRITE PROTECT (WP):** The 24C02/24C04 has a Write Protect pin that provides hardware data protection. The Write Protect pin allows normal read/write operations when connected to ground (GND). When the Write Protect pin is connected to VCC, the write protection feature is enabled and operates as shown in the following Table 1.

6. Testing the keyboards in the program shell and checking the outputs of the ALTERA FPGA.

The colors of the squares can be of four colors. Light blue - there was no shortage.
Yellow or red - there were closures (the colors change alternately).
Black square - disguised, so as not to interfere and not distracted. You can disguise closed lines, that is, those squares that change their color alternately.
Now in more detail.
The numbers on the right are the lines to which the programmer exposes the log.0
Figures at the top - columns on which the scanning log is received.
From the picture on the right:
- Yellow square - 4 and 17 contacts are shorted.
- Red - 24 and 26 contacts.
- Black - disguised 10 and 23 contacts.

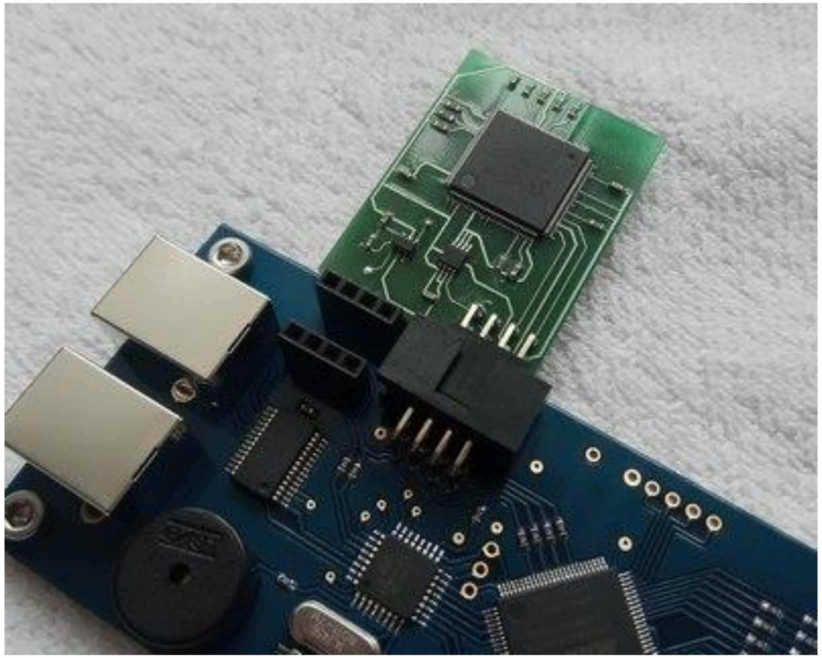
Thus, we have a cycle 'X' from 1 pin to 31 (which we set to log.0) and with each pass we follow the pin of 'X' + 1. This explains the field of triangle-shaped squares. Passing the scanning log. 0 is always done in one direction - from the first to the last.
You can exit the check of keyboards in the shell with the Exit button. At the request of users, the ability to exit the keypad check mode by pressing the reset button on the programmer or disconnecting the programmer from the PC is added. In other modes of the programmer you need to finish the work in regular ways - by clicking on the exit button or the cross in the upper right corner of the shell!



If you have a suspicion that you have damaged one or more outputs of the ALTERA FPGA with your sloppy actions, then this can be easily check.
It is necessary to disconnect all cables and ffc from the programmer, remove the jumper, and leave only the cable from the PC.
Then go to the mode of testing the keyboard. After the appearance of the window, as in the picture above, in case of problems, you will see blinking rows or columns.

If you do not observe such a picture, then there is no high probability of problems with the output pins of FPGAs.

7. Working with EC of Nuvoton through an adapter.



In the shell, this mode name is Nuvoton SPI. EC of the company Nuvoton, is the use of a small adapter, which soldered the chip to work with it. We work with EC on such an adapter via the SPI interface.
Now, the adapters are no longer produced, because when working with EC Nuvoton on motherboards, the JTAG interface is used.
Item Nuvoton SPI will soon be removed from the program's shell.

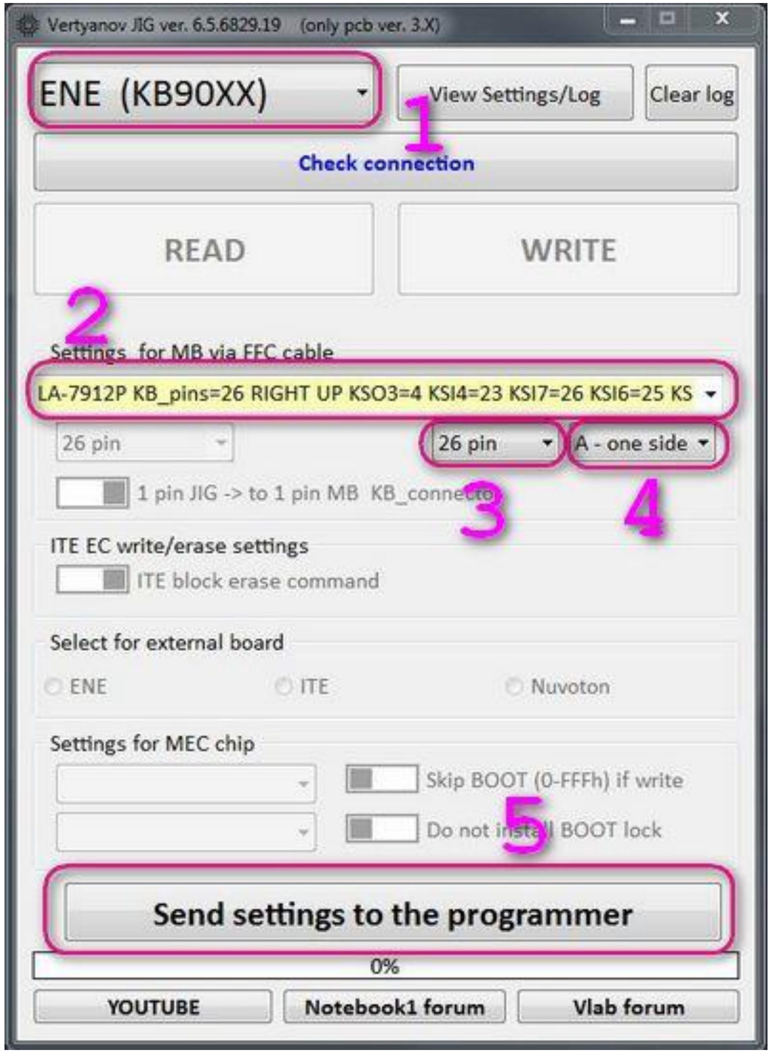
8. General rules and order of connections when working with EC ENE / ITE / NUVOTON with FFC cable through the keyboard connector (KB_connector)

Consider the option when in the setup line for MB there are parameters that indicate the location of the first pin of the keyboard connector- KB_pins, UP / DOWN, LEFT / RIGHT. (The assignment of signals is described in paragraph 2 of this instruction)

Procedure:

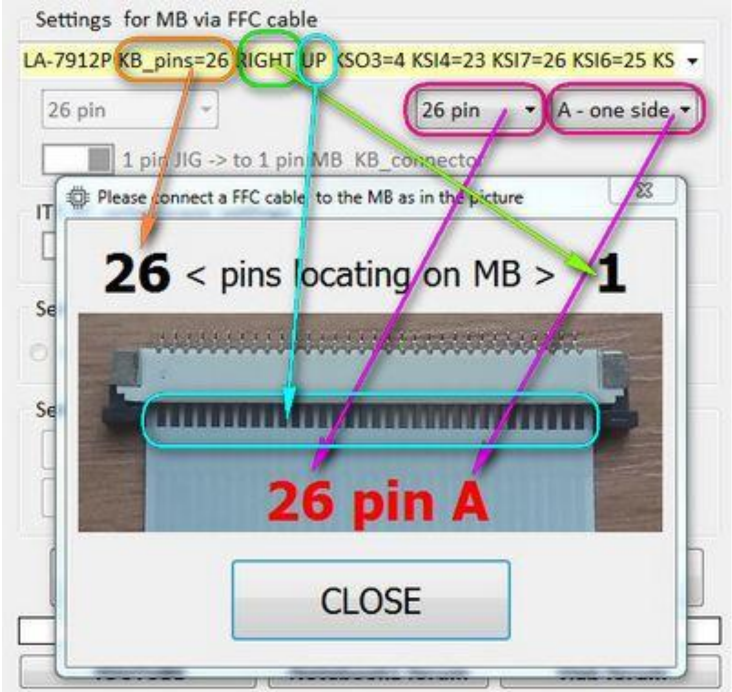
• Do not connect the power supply for the motherboard!
• Connect the programmer to the PC and start the shell.
• Connect the cable from the GND connector on the programmer to any free USB connector on the motherboard. So, we connect the ground of the programmer and the MB. The shorter will be this cable - the better and more reliable.
• In the shell we enter one of the modes, ENE / ITE / NUVOTON, based on the EC manufacturer installed on the MB. **(1)**
• Select MB from the drop-down list. **(2)**
• Choose a ffc from those that the shell offers in the drop-down list. If we work with EC of ENE firm with soldering, then we choose Soldering. **(3)**
• Take the ffc from the programmer's set with the same pitch (0.5/0.8/1mm) as KB_connector on MB, and the same number of contacts that were selected in the drop-down list **(3)** earlier and connect it to the programmer always to the first pin!
• Select the type of ffc that was set in the programmer - A or B. **(4)**
A is a direct ffc and the pins are on the same side of the FFC.
B - the reverse ffc and the contacts are on opposite sides.
• We send the settings to the programmer **(5)** and according to the prompt that appeared in the new window, connect the second side of the FFC to the MB. This can be a connection from the first or the last pin of the KB_connector. **On some MB, the FFC can be connected by contacts from MB upwards** - look at the picture carefully. (the UP parameter in the settings line)
• You can now power the MB from the power supply, but not from the battery!
• Take as a rule, the first thing to press the Check connection button in the shell, and then read or write.

Further actions with the programmer are described in the relevant sections under the specific EC manufacturer.

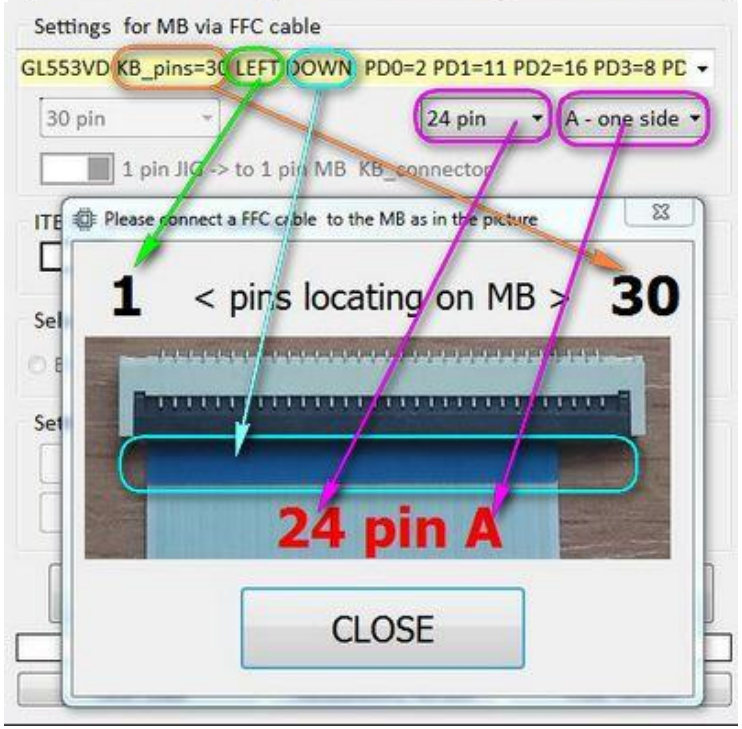


EXAMPLES

In the picture to the right, the setting for ENE EC on MB **LA-7912P** is selected. The KB_connector of this MB has a pitch of 1mm, the number of pins of connector 26, the pins in the connector are at the top.
The user selected a ffc equal to KB_connector size for 26 pins.
From the picture you can determine the location of the first pin on the MB - is on the right if you look at the connector from yourself.



Another example.
In the picture to the right, the setting for EC ITE and MB ASUS **GL553VD** is selected. The KB_connector of this MB has a pitch of 1mm, the number of pins in the connector is 30. Contacts in the connector are at the bottom.
The user has selected a ffc 24 pin, when as KB_connector has 30 pins.
After sending the settings to the programmer, the shell offers the end of the ffc to connect from the first pin to the MB, to block all the necessary signals with the selected ffc.
The first pin on MB, judging by the picture, is on the left if you look at the connector from yourself.

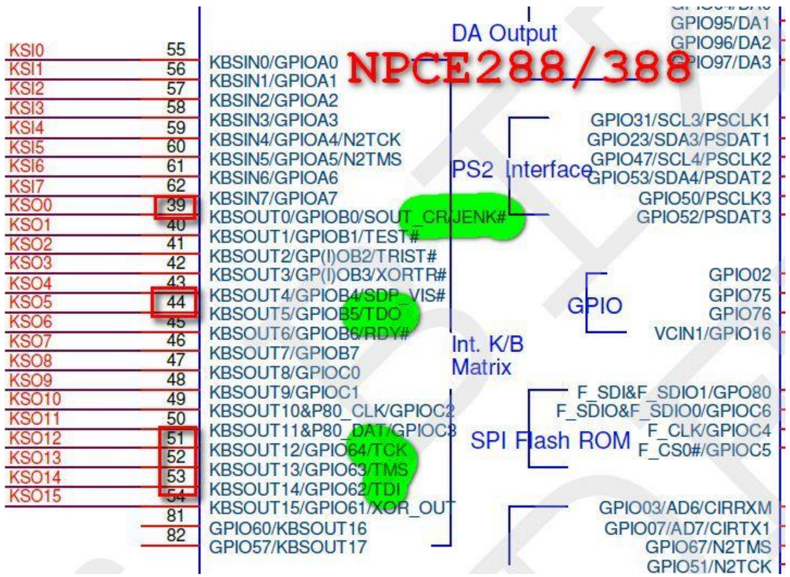


The second, old version of the connection is based on using a ffc with the number of contacts equal to the number of contacts in KB_connector. In view of the fact that ffc were added to the programmer's package arbitrarily and for one number of contacts there was either just a straight line or only a reverse cable, then there was a tuning for the conversion - the reverse of the contacts. If the first contact of the programmer on the ffc go to on the first contact of the keyboard connector, then the check mark (the switch was installed). If the first pin goes to on the last contact, the tick was removed.
This method was convenient and simple in the case when there was an FFC with the required number of contacts. If there was no suitable FFC, then you can get a complicated situation. The difficulty was to artificially change the position of the control contacts by upgrading the number of the pins. Mathematically, this problem was solved, but it caused inconvenience because of what the mechanism was developed and proposed as the main one.
Now, not all MB settings have been modified to use this technique. Not all setup lines have additional settings that describe the configuration of KB_connector. The name of these settings, I repeat, **KB_pins, UP / DOWN, LEFT / RIGHT.**
Users who did not have previous experience using the old technique may encounter some difficulties connecting to MB, which does not have these additional settings. In this situation, the easiest solution is to add these settings to the file, especially since MB will be in front of your eyes at this moment. After adding the parameters, the new method described earlier will be available to use. The second positive point of the new method with the use of additional settings is the elimination of errors related to determining the location of the first pin to MB.
If you add the settings to the lines, if the result of work with the EC is positive, then it would not be bad to share information with the team.

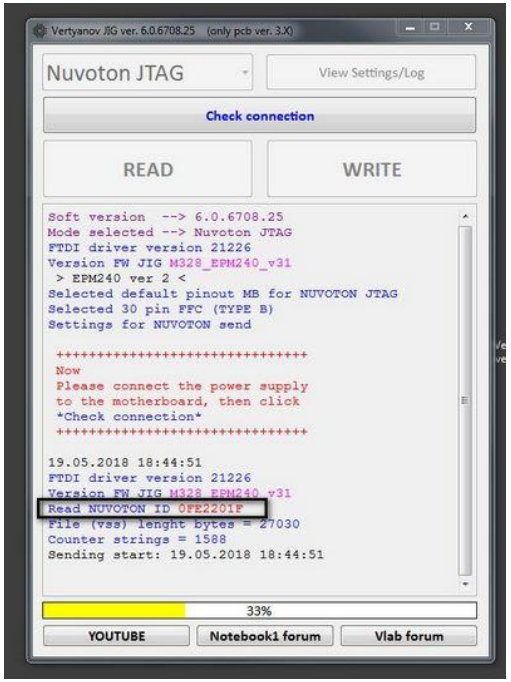
PS Let me remind you that on the programmer all connections to its FFC connectors are always from the FIRST pin! This applies to both the new method and the old one. This allows you to use one "wide" connector for each step of the outputs to “cover” all possible situations and not to install the same type of connector on the programmer for a different number of pins.

The picture on the right is for illustration.
In such a variety there is no practical benefit, and the downside is an additional lengthening of the communication line and the possibility of making problems with the contact.
Everything is solved by only three connectors with a pitch of 0.5/0.8/1mm already present on the programmer.

9. **Work with Nuvoton EC installed on MB, with FFC cable.**
NPCE288N and NPCE388N



When working with EC NUVOTON, it was noticed that the following pinout of the necessary signals is mainly used:
KB_pins=30 JENK=7 TCK=19 TMS=20 TDI=21 TDO=10.
The first sign that you most likely have a "standard" pinout is a 30-pin keyboard connector (KB_pins=30). The word "standard" is my proposal and by no means is it in a literal sense a kind of standard. It is entered only for convenience of work.
If on your MB with EC NUVOTON NPCE288N/388N keyboard connector is not for 30 pins, then you need to compose the line and enter it into the **MB connection file NUVOTON.txt**
The list of "non-standard MB", under which settings have already been made, can be seen in this file.
The settings line in this file has the following appearance:
LA-C771P KB_pins=30 LEFT DOWN JENK=7 TCK=19 TMS=20 TDI=21 TDO=10
the number of contacts of the keyboard connector, the orientation of the first pin, the location of the contacts, and five signaling signals. The number indicates the pin number of the keyboard connector to which this signal comes from the EC.
(The file structure is described in the second paragraph above.)
Signals **JENK/TCK/TMS/TDI/TDO** - JTAG interface. The exception is the JENK signal, and it is always in log 0 to activate the interface in EC Nuvoton.



To make a string of settings, if you do not have 30 pins keyboard on the MB, or the signals are not standard, or you do not see them in the list of cards, then you need to use the MB circuit, "board", or find the signals yourself using a multimeter. The resulting line is added to the file **MB connection NUVOTON.txt**

It would be very good, after a successful work with the EC, to share with the team the setting for a new MB!

The picture shows the EC Nuvoton, whose signals should be found on the keyboard connector (pins 39, 44, 51, 52, 53).
The procedure for connecting the FFC cable and the sequence has been described in detail in the general rules and the procedure for connecting with EC ENE/ITE/NUVOTON using the FFC cable through the keyboard connector (KB_connector)
In the shell, select NUVOTON JTAG. **In the motherboard selection field, select MB only if the settings for it are different from the standard settings.** If the settings are standard - do not touch the field.
If the settings and all connections are made correctly and there are all the necessary voltages on the EC power pins, ID EC 0FE2201Fh will be received and the service data will be loaded into the EC. Upon completion of the sending of the service data, EC content will be read and written. Reading the contents of the FLASH EC takes a little longer than writing.
After the end of reading or writing, first, you need to remove the power to MB, and then everything else.

10. **Work with EC of ENE, installed on MB, over FFC cable.**
KB9010 / 9012/9016/9022/9028/902

In the **MB connection ENE.txt** file there are settings for each MB. A lot of MB with matching settings, with different names of MB platforms. If your MB is not in the list, you can try to use the settings for the nearest by number.

To work with the EC of ENE, the following signals are needed: **KSO3, KSI4, KSI7, KSI6, KSI5.**

For example, for the Compal LA-9102P platform, the settings line looks like this:

LA-9102P KSO3=16 KSI4=27 KSI7=29 KSI6=28 KSI5=25

The name of the motherboard and five service signals. To compile a line of settings, if there is not a required list in the list of cards, you will need to use the MB scheme, the "board", or find the signals yourself using a multimeter. The resulting line is added to the file **MB connection ENE.txt**

Signals **KSO3/KSI4/KSI7/KSI6/KSI5** - SPI interface. The exception is the KSO3 signal, and it is always in the log 0 to activate the interface in the EC ENE.

It would be very good, after a successful work with the EC, to share with the team tuning to a new MB!

The picture shows EC ENE, whose signals should be found on the keyboard connector (pins 59,60,61,62,42)

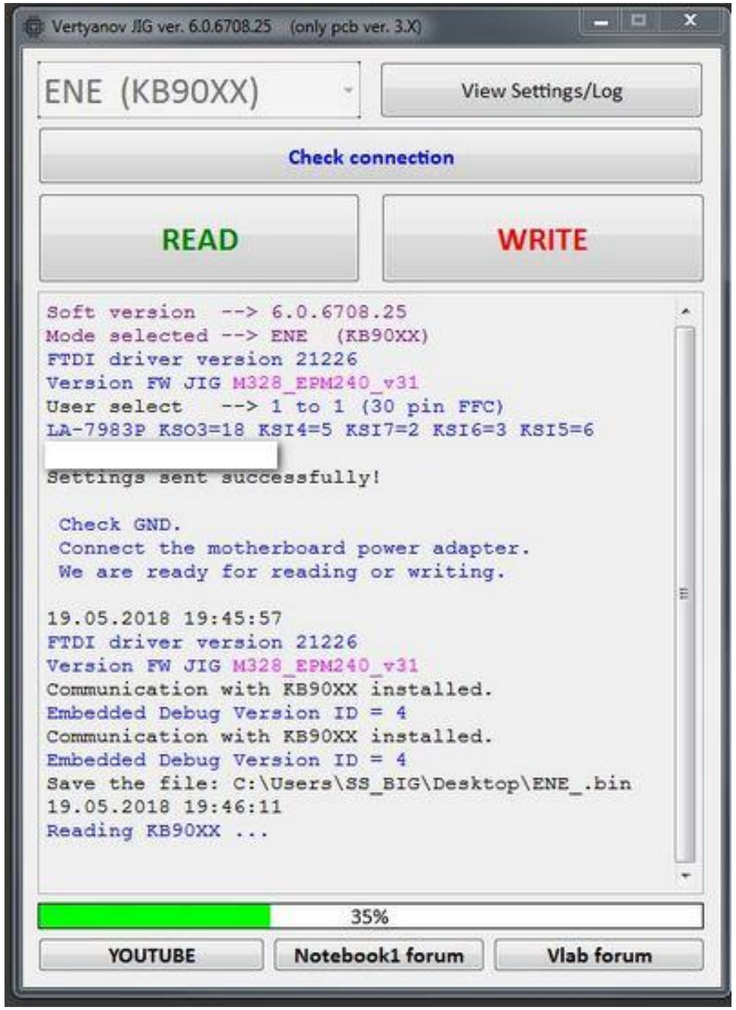
The procedure for connecting the FFC cable and the sequence was described in detail in general rules and order of connections when working with EC ENE/ITE/NUVOTON with FFC cable through the keyboard connector (KB_connector)

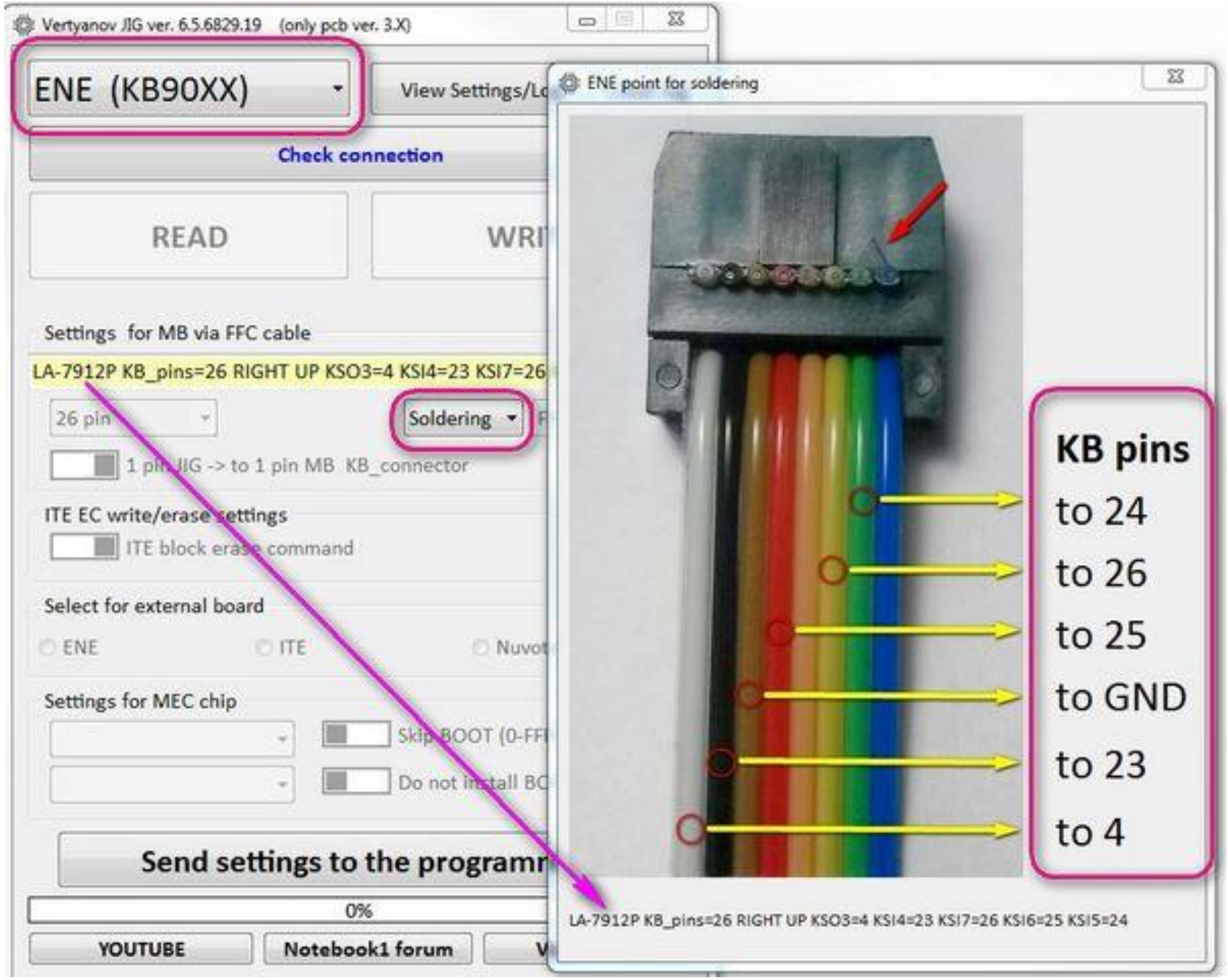
In the shell, select ENE (KB90XX).

If the settings and all connections are made correctly and there are all the necessary voltages on the EC power pins, then when the button is pressed, the communication check, read or write in the shell log will be written about the successful connection to ENE EC.

After the end of reading or writing, first, you need to remove the power to MB, and then everything else.

For EC ENE, there is another way to work by soldering a color cable from the programmer's package. The sequence is similar, only in the FFC drop-down list you need to select Soldering. A picture appears with a hint of cable soldering points on the MB keyboard connector. Do not forget to also send the settings to the programmer. The cable is connected to the side of the programmer. If there are no parameters describing KB_connector in the setup line for MB, then the drop-down list will not be active until you select any numerical value to the left of it. The value you select does not affect anything, it only opens access to the FFC list and soldering.


Vertyanov JIG ver. 6.0.6708.25 (only pcb ver. 3.X)
ENE (KB90XX)
View Settings/Log
Check connection
READ
WRITE
Soft version --> 6.0.6708.25
Mode selected --> ENE (KB90XX)
FTDI driver version 21226
Version FW JIG M328_EPM240_v31
User select --> 1 to 1 (30 pin FFC)
LA-7983P KSO3=18 KSI4=5 KSI7=2 KSI6=3 KSI5=6
Settings sent successfully!
Check GND.
Connect the motherboard power adapter.
We are ready for reading or writing.
19.05.2018 19:45:57
FTDI driver version 21226
Version FW JIG M328_EPM240_v31
Communication with KB90XX installed.
Embedded Debug Version ID = 4
Communication with KB90XX installed.
Embedded Debug Version ID = 4
Save the file: C:\Users\SS_BIG\Desktop\ENE_.bin
19.05.2018 19:46:11
Reading KB90XX ...
35%
YOUTUBE
Notebook1 forum
Vlab forum


Vertyanov JIG ver. 6.5.6829.19 (only pcb ver. 3.X)
ENE (KB90XX)
View Settings/Lo
Check connection
READ
WRI
Settings for MB via FFC cable
LA-7912P KB_pins=26 RIGHT UP KSO3=4 KSI4=23 KSI7=26
26 pin
Soldering
1 pin JIG -> to 1 pin MB KB_connector
ITE EC write/erase settings
ITE block erase command
Select for external board
ENE
ITE
Nuvot
Settings for MEC chip
Skip BOOT (0-FFF
Do not install BO
Send settings to the programm
0%
YOUTUBE
Notebook1 forum
ENE point for soldering
KB pins
to 24
to 26
to 25
to GND
to 23
to 4
LA-7912P KB_pins=26 RIGHT UP KSO3=4 KSI4=23 KSI7=26 KSI6=25 KSI5=24

11. Work with EC of ITE firm, installed on MB, on FFC cable.

IT8386 / 8580/8585/8586/8587/8886/8985/8987

In the **MB connection ITE.txt** file there are settings for each MB. A lot of MB with matching settings, with different names of MB platforms. If your MB is not in the list, you can try to use the settings for the nearest by number. To work with the EC ITE, you need the following signals: **PD0-PD7, BUSY, STB, AFD, INIT, SLIN, KSI4, KSI5.**

For example, for the NM-B321 platform, the settings line looks like this:

NM-B321 KB_pins=32 (JKB1) PD0=20 PD1=16 PD2=14 PD3=9 PD4=13 PD5=17 PD6=10 PD7=12 BUSY=23 STB=15 AFD=26 INIT=19 SLIN=18 KSI4= 22 KSI5=21

The name of the motherboard, the number of contacts in the keyboard connector, and 15 service signals. To compile a line of settings, if there is not a required list in the list of cards, you will need to use the MB scheme, the "board", or find the signals yourself using a multimeter. The resulting line is added to the file **MB connection ITE.txt**

The signals KSI4 and KSI5 are always in the log 0 for activating the ISP interface in the EC ITE.

It would be very good, after a successful work with the EC, to share with the team the setting for a new MB!

The picture shows EC ITE in the LQFP-128 case, the signals of which should be found on the keyboard connector (pins 58-63,36-43,45).

The 15 required service signals for EC IT8580/8585/8586/8587/8985/8987 are on the same pins of the chip. Also, on the picture IT8386VG in the case VFBGA128.

The procedure for connecting the FFC cable and the sequence has been described in detail in the general rules and the order of connections when working with EC ENE/ITE/NUVOTON with the FFC cable through the keyboard connector (KB_connector).

In the shell, select ITE (IT8XXX).

For some EC (IT8587/8987) there should be a switch for erasing FLASH memory blocks. For others, this setting does not matter.

If the settings and all connections are correct and there are all the necessary voltages on the EC power pins, then when you click the link check button in the shell log, you should see the name of the EC and its revision.

The EC name written on the chip must match the name in the log.

Revision is the letter: 1-A, 2-B, 3-C, 4-D, 5-E, 6-F.

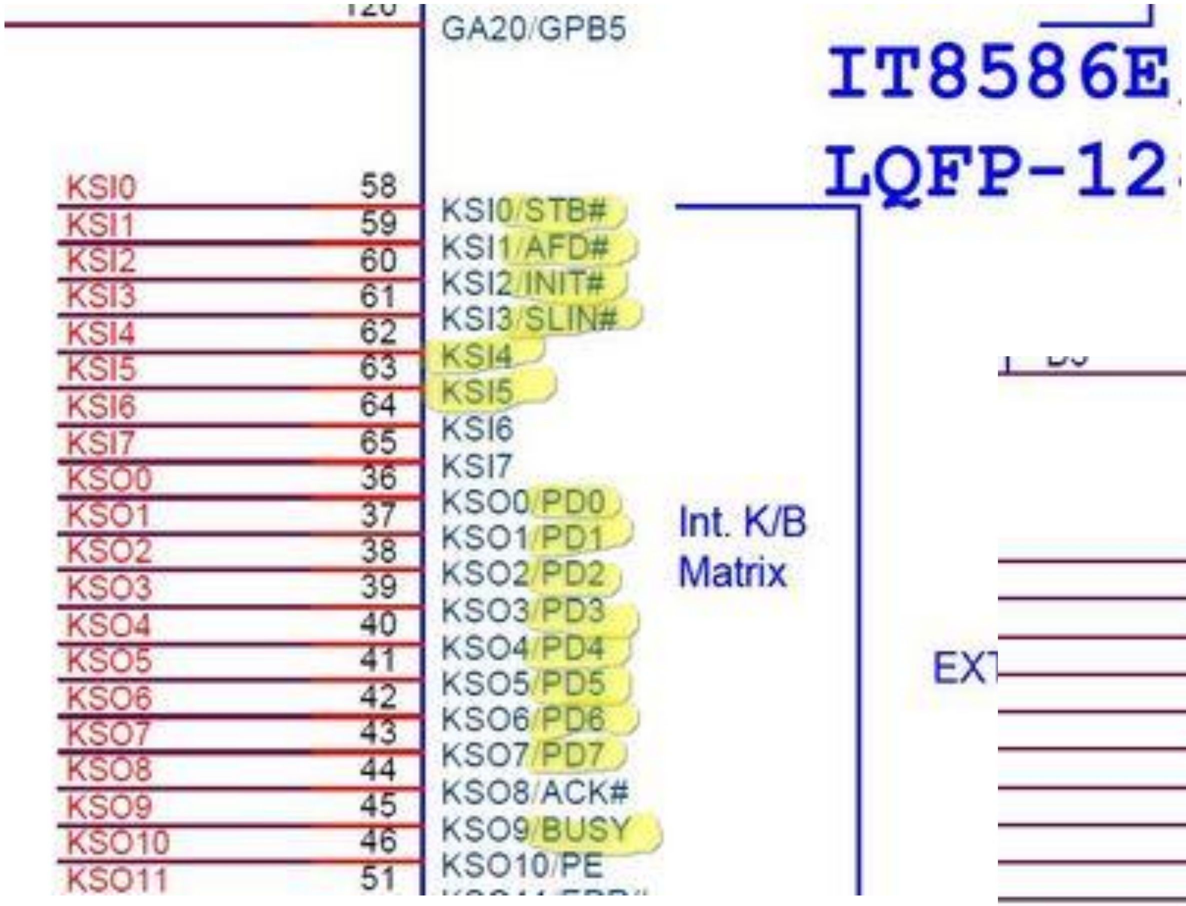
After the end of reading or writing, first, you need to remove the power to MB, and then everything else.

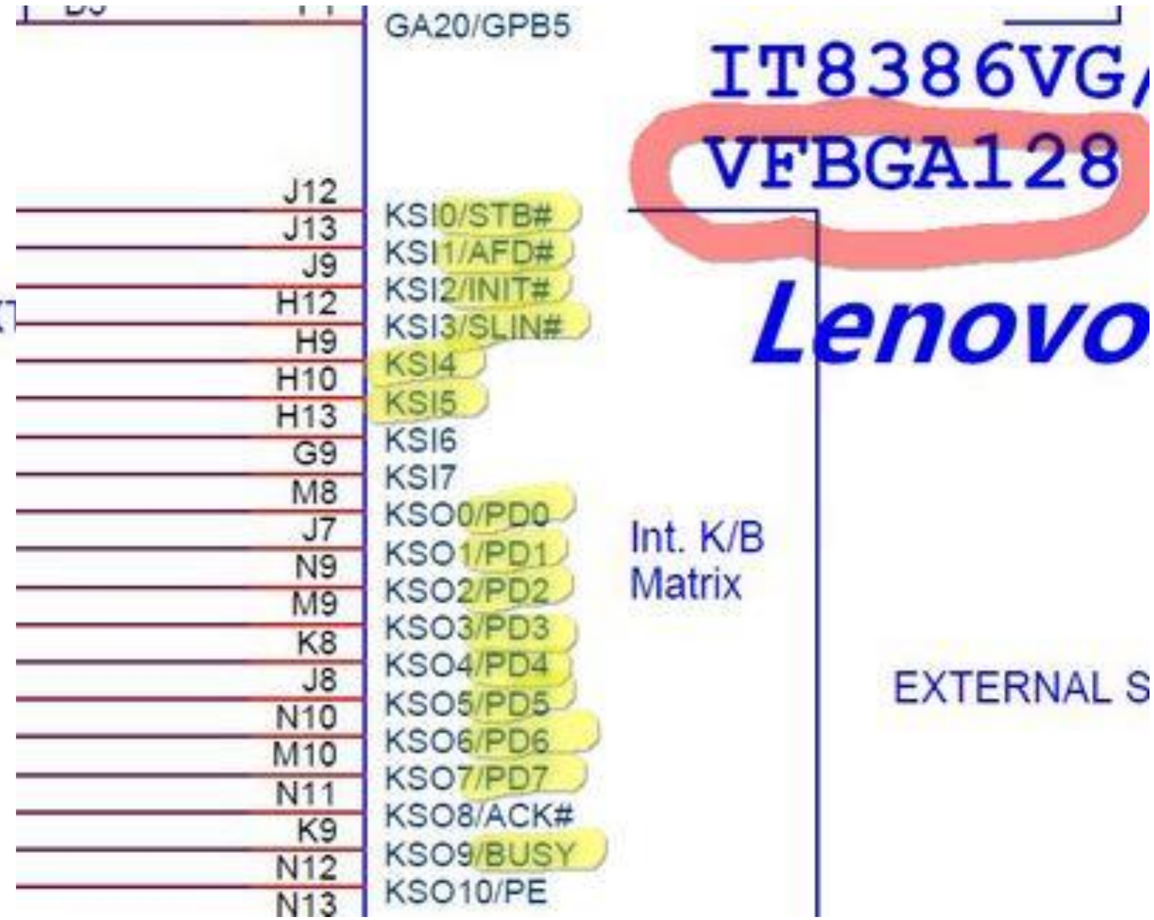
When writing ITE EC, the method of skipping bytes with byte value = FFh is implemented. This is since after erasing the EC chip, all FLASH memory contents contain FFh, and it makes no sense to write the same value. Thus, it is possible to reproach the recording process, and this explains the uneven progress of the bar at the bottom of the shell by the places. After writing to the EC, the shell will automatically verify the data.

```
Check connection ...
Received ITE ID  -> 8585 rev -> 06
21.05.2018 14:44:45
FTDI driver version 21226
Version FW JIG M328_EPM240_v31
 Now, as soon as possible,
 connect the adapter (5 sec)!
Check connection ...
Received ITE ID  -> 8585 rev -> 06
Save the file: C:\ITE_.bin
21.05.2018 14:45:00
Check connection ...
Received ITE ID  -> 8585 rev -> 06
Init ITE ...              complete.
Reading, please wait ... complete.
21.05.2018 14:45:11
```

```
Check connection ...
Received ITE ID  -> 8585 rev -> 06
21.05.2018 14:46:09
Write to ITE file: C:\ITE_.bin
Loaded from file to buffer 20000h bytes.
Check connection ...
Received ITE ID  -> 8585 rev -> 06
INIT complete.
ERASE ...
ERASE complete.
The number of bytes value of FFh -> 9400
Writing, please wait ... complete.
Verify ... complete.
The data is the same!
21.05.2018 14:48:26
```

```
Check connection ...
Received ITE ID  -> 8585 rev -> 06
```

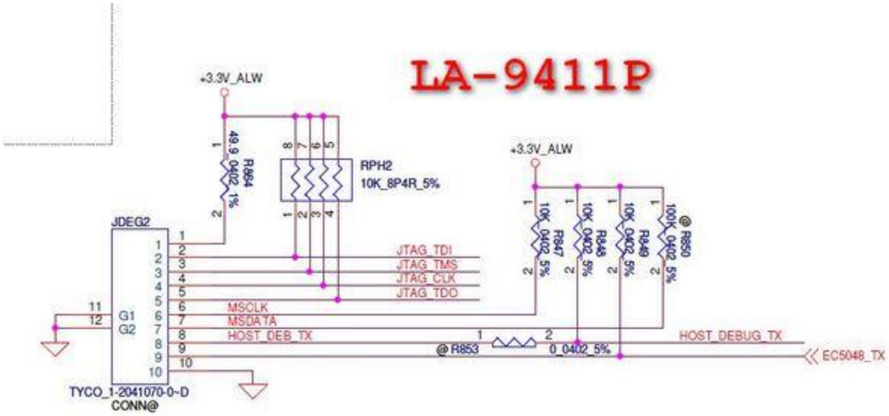

IT8586E
LQFP-12
GA20/GPB5
KSI0 58 KSI0/STB#
KSI1 59 KSI1/AFD#
KSI2 60 KSI2/INIT#
KSI3 61 KSI3/SLIN#
KSI4 62 KSI4
KSI5 63 KSI5
KSI6 64 KSI6
KSI7 65 KSI7
KSO0 36 KSO0/PD0
KSO1 37 KSO1/PD1
KSO2 38 KSO2/PD2
KSO3 39 KSO3/PD3
KSO4 40 KSO4/PD4
KSO5 41 KSO5/PD5
KSO6 42 KSO6/PD6
KSO7 43 KSO7/PD7
KSO8 44 KSO8/ACK#
KSO9 45 KSO9/BUSY
KSO10 46 KSO10/PE
KSO11 51
Int. K/B
Matrix
EX


IT8386VG
VFBGA128
Lenovo
GA20/GPB5
J12 KSI0/STB#
J13 KSI1/AFD#
J9 KSI2/INIT#
H12 KSI3/SLIN#
H9 KSI4
H10 KSI5
H13 KSI6
G9 KSI7
M8 KSO0/PD0
J7 KSO1/PD1
N9 KSO2/PD2
M9 KSO3/PD3
K8 KSO4/PD4
J8 KSO5/PD5
N10 KSO6/PD6
M10 KSO7/PD7
N11 KSO8/ACK#
K9 KSO9/BUSY
N12 KSO10/PE
N13
Int. K/B
Matrix
EXTERNAL S

12. Working with EC of Microchip.

MEC1609/1619/1633/1641/1650/1651/1653/5035/5045/5055/5075/5085 ...

To work with the EC company MICROCHIP (formerly SMSC), you need to find the MB points for connecting signals to the JTAG interface. Signal names: **TDI/TDO/TCK/TMS/JTAG_RESET#**. Do not forget about GND. Very often, these signals are output in one place to an unplugged connector.



Always ensure that the **JTAG_RESET#** signal is not pulled to the GND by a low-resistance resistor. On **JTAG_RESET#** in the inactive state there should be a log. 1. If you want to erase and write FLASH to EC, for example, MEC1609, then the log 1 to nFWP. For example, for MBX-237, set R412 from 1k to 3k. In some EC MEC, this pin may not be present. When working with EC, all power pins (VTR_) should be 3v3.

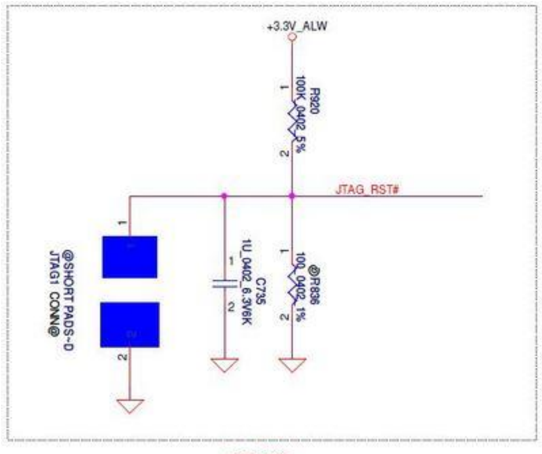
To activate the JTAG mode in EC, you must have a resistor on the 100-ohm-1k through the MSDATA line on GND. Check its availability!

In addition to the physical connection, data on the size of the FLASH memory of a specific EC is required. For this purpose, the **MEC settings.txt** file is in the folder with the shell. The file structure is very simple.

Let's look at a couple of examples:

CHIP=MEC5035 FLASH_start=0 FLASH_size_bytes=196608 INFO_size_bytes=2048

- MEC5035 EC name, which will be displayed in the shell
- FLASH_start=0 FLASH_size_bytes=196608 address of the beginning of the FLASH memory and the length of the memory in bytes in decimal.
- INFO_size_bytes=2048 size in bytes of the INFO area. Since the INFO area, if it exists in a EC, starts immediately after the FLASH area, then there is no point in specifying the start address.



The following example:

CHIP=MEC1650 FLASH_start=0 FLASH_size_bytes=294912 (EEPROM 2KB) EC_ID=02152445h

There is no INFO section in this EC, therefore there is no in line. It is not necessary in this case to write INFO_size_bytes=0!

(EEPROM 2KB) work with EEPROM is not implemented now and is left as a reminder.

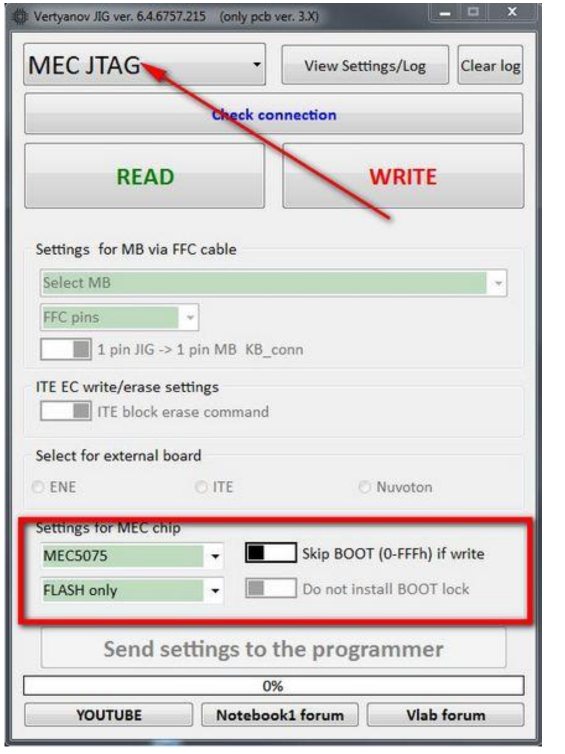
EC_ID=02152445h is the ID of a EC, in this case MEC1650. Also, while it is informative.

In the EC I distinguish two kinds of ID.

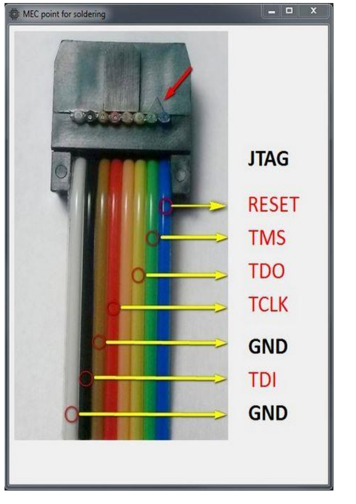
- JTAG_ID for all MECs will be one 200024B1h and get its standard JTAG command Ch
- EC_ID 02082445h, for example, for MEC5075 we get already access through registers in the EC itself.

Whether it is necessary to use EC_ID and how, I yet did not solve. Technically, by accumulating the database on EC_ID, you can abandon the configuration file and use auto-detection.

In the MEC EC, the BOOT block can be closed from reading and modification, and access to FLASH memory via the JTAG interface is blocked. The BOOT block is located at the beginning of the address space of the FLASH EC memory. If the BOOT block is closed, the first 4KB (BOOT size of the block) will not be read while reading - EC gives the FFh byte in the address range from 0 to FFFh. The shell will warn about this.
If the EC will be closed through JTAG, then nothing can be read from the FLASH memory, and nothing is obtained with the help of registers. When JTAG is closed, EC_ID, and we get it through registers, it's also impossible to read.
The protection of the BOOT unit and the protection against access via the JTAG interface can only be removed by a complete EC erase. I must warn that a full erasure of the EC will destroy the same data in the EEPROM EC. Now, the work with EEPROM did not even begin. If the EC has EEPROM and the manufacturer used it, then the data will be lost irrevocably. Before applying a full EC erase, you must find a full firmware with BOOT block. Otherwise, you will get a "brick".
On the forums are very often laid out EC MEC firmware in which the BOOT block is missing (the first 4KB contains 00h or FFh). For DELL laptops, the BIOS update file has a full EC image with a BOOT block. The company SONY, on popular models, BOOT block is not closed, and it can be considered. On TOSHIBA laptops JTAG access is closed and the update is not full image, but also cryptic. Remember, there must be a clear understanding of what and for what you want to do. Just because FLASH memory in EC cannot be damaged.
Do not need to create additional difficulties in repair.

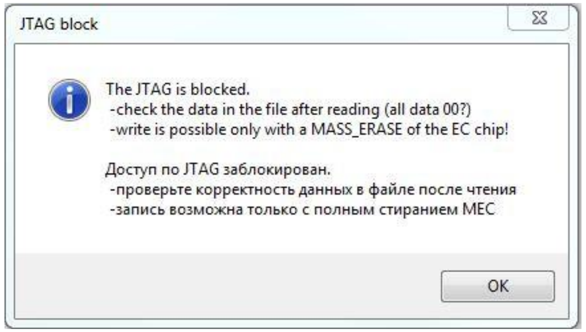


In the complete set with the programmer there is a color cable with a connector on the end. A pinout of this cable will always appear when you select **MEC JTAG** in the shell. GND can connect one of two, RESET is mandatory if you do a complete erasure when the BOOT or JTAG is closed. When working with MEC EC, the settings are shown in the screenshot.
In more detail, I'll talk about the switch positions using the MEC5075 as an example. The MEC5075 contains 288KB FLASH memory and INFO area it does not. I did not find information on EEPROM in it. When reading from the MEC, a 288KB file will be created. If the BOOT block is closed from reading, the shell will issue a warning, but will read the entire MEC. In this file, the first 2KB will be filled with the value of FFh since BOOT is closed from reading. If BOOT is not closed, the file contains the full contents of FLASH.



In the case of a write, you must have a file of 288KB in size. If you do NOT plan to overwrite the BOOT area and JTAG is not closed, then the upper slider in the shell to the left position -
Skip BOOT. In this case, the EC write will start immediately after the BOOT area (offset FFFh + 1), and the data will be taken from the file with the same offset. If you need to overwrite the BOOT block or if JTAG is closed, the EC erase, and

the position of the upper engine will be done as in the picture to the right. In this case, the BOOT block must be present at the addresses 0-FFFh, otherwise the positive result will not be accurate.
If the BOOT block is not closed, JTAG access is not denied, then in this position of the cursor (to the right) the blocks will be erased and the contents of the EEPROM will remain untouched. Of course, the BOOT area will be overwritten, and it should also be present in the firmware file. Thus, two types of memory erasure MEC - complete and blocks, affecting only the necessary places are realized.
The bottom engine is only active when we write to the BOOT area. It is responsible for setting the lock BOOT area.
When the shell starts, by default, the BOOT area is skipped on the engines and the BOOT area is not protected.
Documentation on EC Microchip began to spread on its website, but there are typos, copying pages from one EC to another and it is not entirely clear that this or that mechanism of operation of both protection and locking is described. According to scraps of information from different sources, through work and experiments, work was done with the EC to remove the locks.
A few words about the blocking of the BOOT area. MEC controllers are 32-bit. If the low bit in the last 32-bit word is log 0 then after blocking the **POR** (one of the reset signals), the BOOT area lock will be set. In other words, the EC lock can be installed not only by the engine, but also by the dump that you write. The minor bit that I'm talking about is at the FFCh address (0th bit) of the BOOT area. The engine we tell the shell "impudent" set the bit to zero and make a reset.
Below is a table that indirectly confirms the idea of blocking the BOOT area.



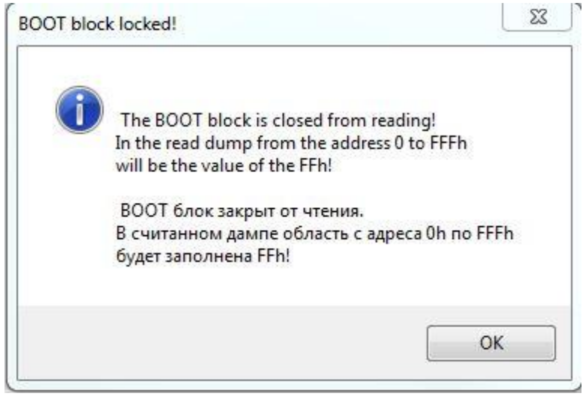


TABLE C-1: BOOT ROM CONTROL FLAGS

BITS *	VALUE	DESCRIPTION
0	0	JTAG accesses blocked except for mass erase.
	1	All JTAG accesses allowed.
1	0	Bypass Boot ROM and jump to 00_0000h.
	1	Enable Boot ROM host communication using UART.
3:2	00	20 msec timeout waiting for Host ACK.
	01	100 msec timeout waiting for Host ACK.
	10	200 msec timeout waiting for Host ACK.
	11	No timeout waiting for Host ACK (that is, wait forever).
4	0	Use external UART clock (GPIO025 = 1.8432 MHz).
	1	Use internal UART clock (MCLK/11).

* These bits apply to both the Flash Memory Array at address 00_0FFCh, and the Embedded Flash Initialization Register.

A small addition to the size of the file that we write to the EC.
The buffer size for FLASH and INFO is selected, according to the settings for the specific EC that the user chooses (the settings line of the **MEC settings.txt** file). The buffer under FLASH is filled with the value FFh, and INFO is filled with the value 00h. If you plan to write FLASH + INFO, then in the file, immediately after the FLASH area, the data for the INFO area should be located. Everything should be in one file.
For example, if the FLASH EC size is 288KB, the same amount will be allocated and filled with the value of FFh. Then a file for recording is loaded into the prepared buffer. If the file is shorter than necessary, then a message will be issued to the log, but all 288KB will be recorded. If the file is longer, the extra file will be discarded.
If the EC has an INFO area and a record and INFO is selected, and the file does not "lie" on the INFO area, it will be written 00h.

I will be grateful for any useful information or addition.

13. **Working with the modules TETRIS, SPIDER, JAWS.**

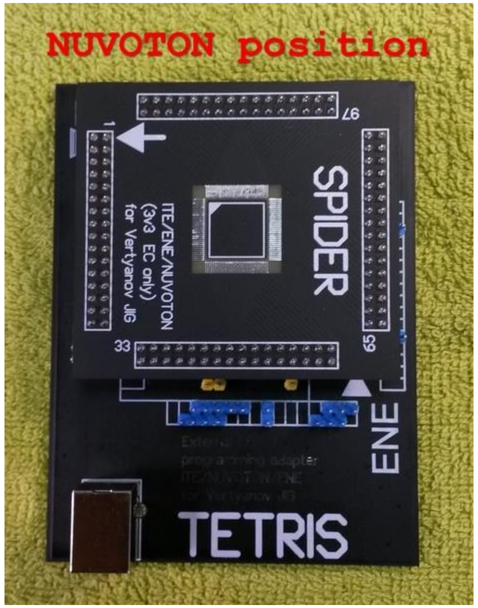
This kit is used to work with EC firms ITE/ENE/NUVOTON separately from MB.

The set consists of three parts:
- The main module of **TETRIS**
- The board on which the EC chip in the LQFP-128 case is soldered – **SPIDER.**
- Board with installed socket, into which EC is inserted in the LQFP-128 case – **JAWS.**

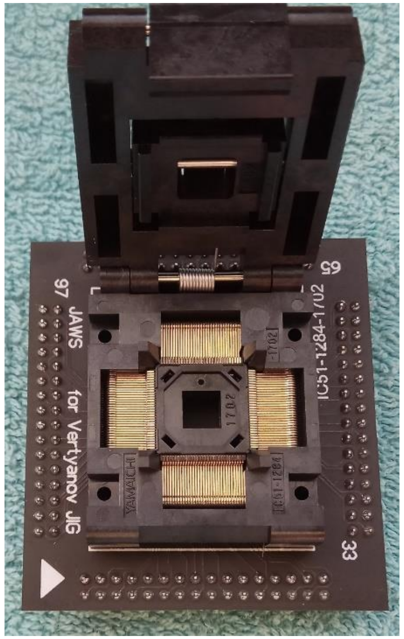
To work, you need two components - **TETRIS**+**SPIDER** or **TETRIS**+**JAWS.**
The ffc cable is supplied with 32 contacts for connection to the programmer.

The appearance of the main **TETRIS** board.
The **TETRIS** board has three options for installing **JAWS** and **SPIDER** cards. Each position corresponds to one of the manufacturers - ITE/ENE/NUVOTON. It is necessary that the programmer and the **TETRIS** board are connected to the same computer - a common ground wire. Otherwise, if **TETRIS** is powered by an external power supply, you need to connect both cards to ground. When working with the kit, you do not need to turn off the power from **TETRIS** - the power key is installed on the pcb.
The **TETRIS** board is equipped with:
- Red LED EC_power - will illuminates when power is supplied to the EC.
- Yellow LED CS_low - will illuminates only when working with EC ITE and with the log.0 on the pin **CS_FLASH.** The LED is informational in nature. It can determine the activity of the **CS_FLASH** signal.
- 8-pin socket for installing the FLASH memory on board for "special" cases with ITE EC.

Installation in their positions in the pictures.



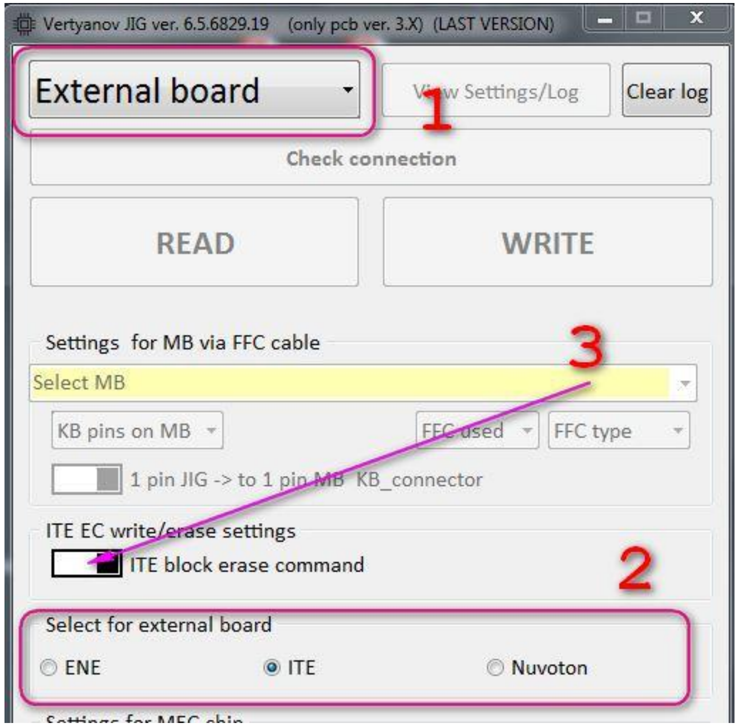
The appearance of the JAWS board.



I want to note that all three boards were manufactured at different times and at different factories, so they may differ in the color of the mask, but they are completely identical in functionality.

The procedure for connecting to the programmer and the actions in the shell:

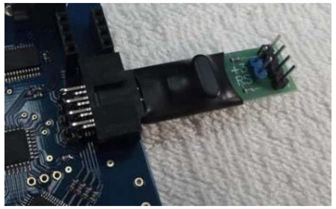
- Connect a ffc to the 32-pin TETRIS card and the programmer.
- Connect the programmer and TETRIS card to one PC with USB cables.
- In the program shell, select the **External board** item **(1)**.
- Choose an EC producer with which you will work **(2)**.
  (If you work with EC ITE, temporarily remove the SPIDER from the TETRIS card or open the JAWS socket to determine the latter correctly.)
- If you are working with IT8587/8987, a switch **(3)** must be set to right position!
  For other ITEs this parameter is not important.
- Send settings to the programmer.
- You can return SPIDER or JAWS and start working with EC.

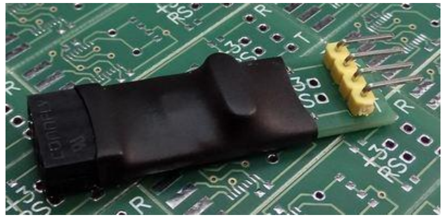


In other respects, working with EC does not differ from work on a ffc and is described in the relevant sections.

Modules **TETRIS** and **JAWS** allow you to check the chips from the supplier before transferring them to the warehouse of the enterprise. Almost all EC, sold under the guise of new ones, are shot with MB and prepared for sale. Among such EC, you can get into your hands frankly faulty ones, at least on power circuits, and in this case such a device will help to weed out the marriage before installing it on a laptop card. In the case when you are reliably aware of the need to replace the EC with a new one on the repaired MB, you can program it before installation. Incorrectly programmed EC, not by its firmware, on "special" MBs, or when the ffc cable solution does not help, this module will help you program the EC.

14. **Work with the EC company Explore** EPF035 (iROG), EPF036 / 021/011. To work with the EC of Explore, you need a pairing module - a EPF whistle. It is inserted into the side connector of the programmer.

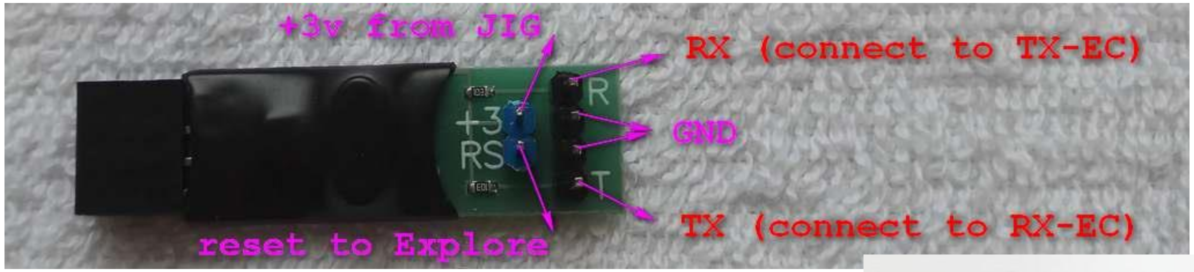




Below is a table of necessary signals for working with Explorer's EC, located outside the motherboard. If EC is on the MB, then at least three signals are used - **RXD_EC**, **TXD_EC**, **GND**. The **EC_OP_MODE** signal must be pulled to the log. 1 to enter the programming mode. Many ASUS MBs that use EPF035/036/037 have unused contacts for programming and can be found on the MB board. From the interface module, in addition, you can use the **RESET_EXPLORE** signal for complex cases, but DO NOT use + 3v when working with MB! (+3v is derived for convenience.)

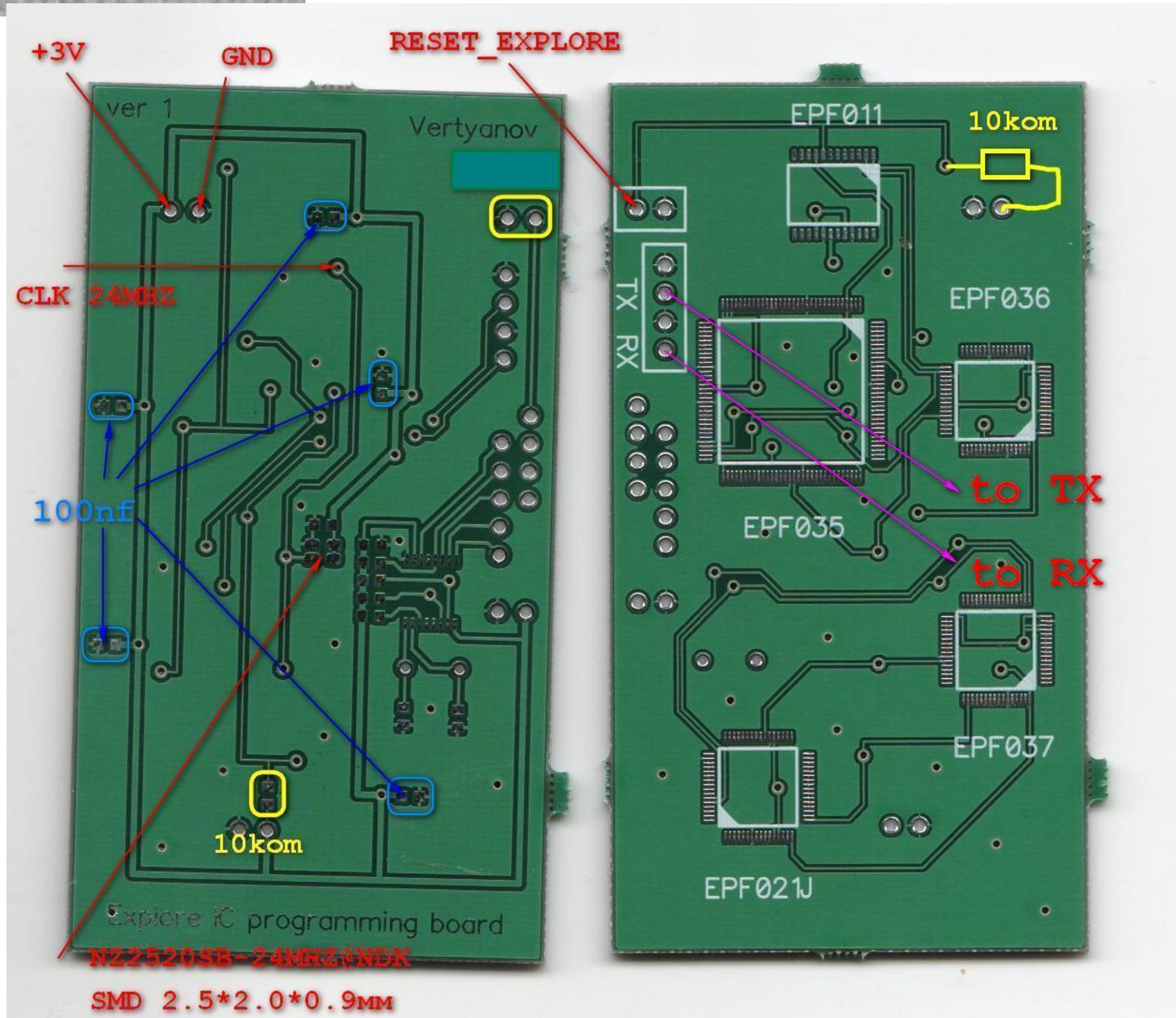
		Pin number				
NAME SIG.	Purpose	**EPF035 LQFP128**	**EPF037 LQFP-64**	**EPF036 LQFP-64**	**EPF011 TSOP24**	**EPF021J**
EC_OP_MODE	PULL_UP!	4	50	1	9	17
GND		13		6		7
3V3		30		15		12
3V3		31	13	16		13
3V3		32	14	17	16	
GND		43	15	26	17	
3V3		48	16			37
GND		51				38
GND		60				
RX_EC	data in	61	17	33	1	59
TX_EC	data out	62	18	34	2	60
3V3		65	33	35		
GND		80	42			50
3V3		83		46		
3V3		96			20	
GND		114	59	59	21	
RESET_EXPLORE	PULL UP	119	64	64	8	16
XTL_IN	24MHZ	81	43	44	18	39
XTL_OUT		82	44	45	19	40

Output signals on the interface module.



If you have a board as in the picture on the right, you need to install smd elements and you can use it to work with EC outside MB. (two R by 10kom, several C per 100n, a quartz oscillator at a frequency of 24 MHZ).

When working with such a board, it is necessary to use all five signals from the interface module. This board was planned to be used together with TETRIS, but later it was necessary to refuse.
To work with EC EXPLORE, you must select EXPLORE (EPFXXX) in the shell.
In each EC Explore there is a boot loader, which is activated with log 1 on the EC_OP_MODE pin every time the chip is turned on or reset. Once the programmer and EC are connected, you can read and write the contents of the memory. If the connection cannot be established, I recommend that the **RESET_EXPLORE** signal be sent from the interface module directly to the EC pin.

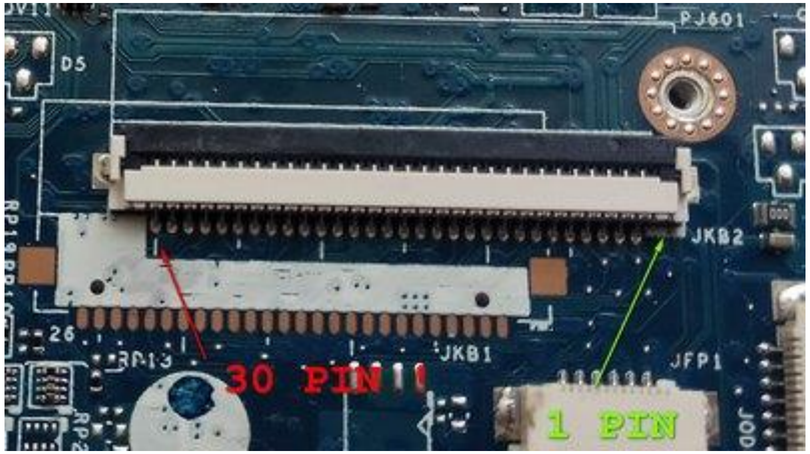


For beginners, be sure to read!

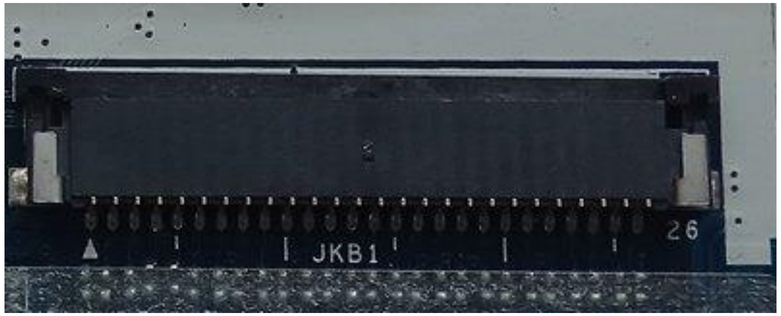
The most common step of the contacts of the keyboard connector is 1 mm. Less common, but can be - 0.8 mm and 0.5 mm.
If you find it difficult to determine the step of the contacts, and the loop plugged into the connector does not make it clear, the solution is as follows. **Take the metal ruler and measure the distance between the first and last pin of the connector in millimeters. This distance in millimeters, divide by the number of contacts in this connector minus one and get a step between the pins.**
For example, we have 30 contacts and a distance from 1 pin to 30 pin is approximately 29mm. We get 29/(30-1)=1mm.
Round off the result you need to the closest applicable "running" dimensions - 0.5 mm, 0.8 mm, 1 mm. If you get several 0.78, then apply a ffc with a step of 0.8 mm, etc. FFC with a pitch of 0.78 mm or 1.1 mm do not exist, and the reason is exactly your measuring ruler.

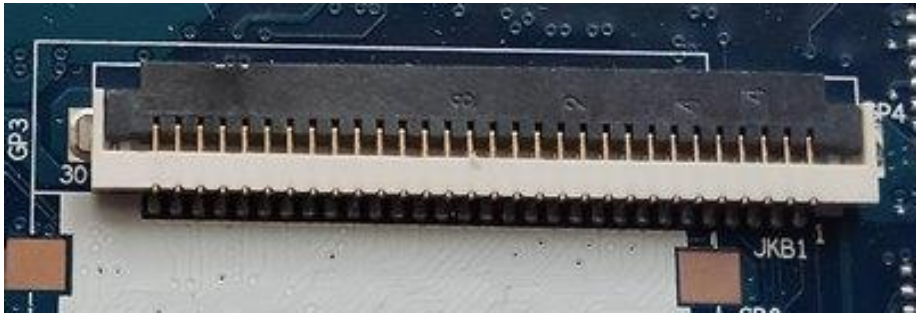
With the definition of the first output on the motherboard sometimes there are difficulties.

Very often for the first pin take this long white strip on the MB (red arrow). In fact, long strips denote tens (10,20,30, etc.), and short strips - five (5,15,25, etc.). Therefore, in this picture, the first pin is on the right! The green arrow points to the "key" of the connector and the first terminal is on the same side.

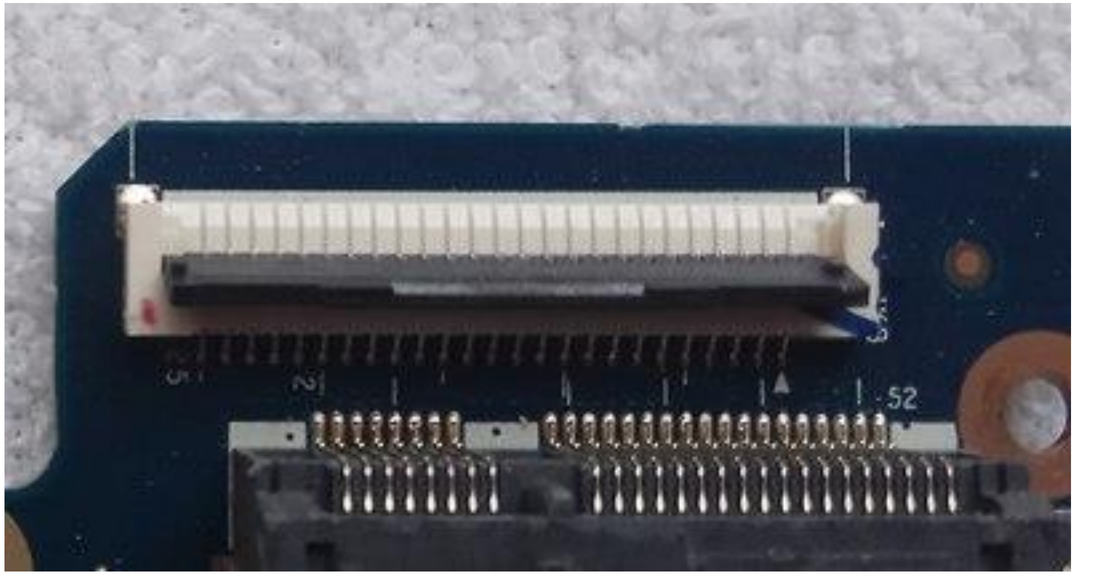


In the pictures below, everything is unambiguous and understandable about the position of the first conclusion.



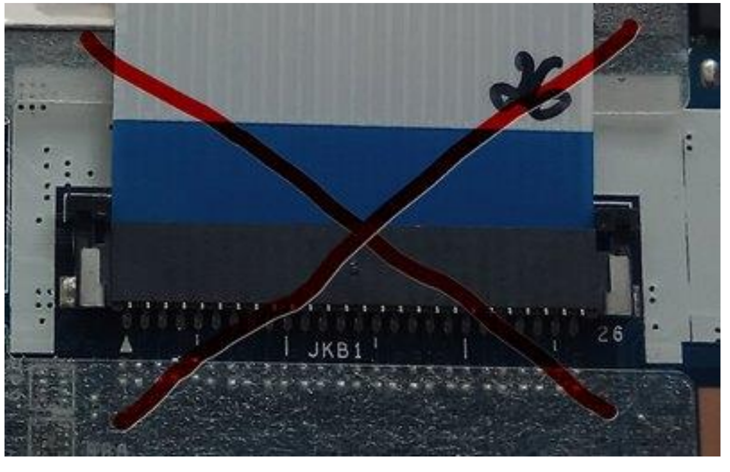


Another "not convenient" type of connectors that Samsung applied to MB from Compal.
A distinctive feature is the 25-pin connector under the keyboard cable with ears. The fact is that in the depth of this connector there is a slight narrowing and when the loop is installed to the end, the plume itself slides to the center slightly and there can be no contact. After installing the loop in such a connector, I recommend checking the pin contact with the terminals of the connector with a multimeter. It is enough to check 2-3 to determine.



The next common mistake is since on some (not flip) connectors the contacts can be located from the top and you need to insert a loop correctly in such connector.

NOT CORRECT!



CORRECT!