JuPic Programmer

Emulator of PICSTART Plus programmer
Works with MPLAB IDE and PICP
http://ajpic.zonk.pl/

Programmer description

The programmer was build with a new processor **PIC16F87,** which has the self programming function. The device can work with Integrated Development Environment **MPLAB™ IDE** (works under operating systems like **Windows 98SE, Windows ME, Windows NT 4.0 SP6a WS, Windows 2000 SP2** or **Windows XP**). This combination makes very useful, efficient and professional tool for each user. Functionality of this pack makes this device easy to use with small, medium or advanced projects and is designed either for beginners or advanced users.

The device is compatible with the original **PICSTART Plus®** programmer and fully integrates with **MPLAB** environment. The programmer is designed to work with FLASH memory **"F"** microcontrollers and with EPROM memory **"C"** family microcontrollers:

PIC10F2XX, PIC12FXXX, PIC16F6XX,
PIC16F7X, PIC16F8X, PIC16F81X, PIC16F7X7,
PIC16F8XA, PIC16F8XX, PIC16F8XXA, PIC18FXXXX

PIC12C5XX, PIC12C6XX, 16C4XX, 16C5XX,
16C6X, 16C6XX, 16C7X, 16C7XX, 16C9XX

MPLAB IDE can be fully free downloaded from Microchip web site **http://www.microchip.com/**. The newest software is available in version "**6.xx**","**7.xx**", the programmer also works with older version "**5.70.40**".

MPLAB program is designed to work with single source files or whole **projects**. Grouping files with project gives a best way to design and control the application, it increase a comfort of work too.

Program can be use with many useful functions:

- ✓ creating and editing source files
- ✓ making the projects with files
- ✓ importing the Intel HEX files
- ✓ debugging the source code
- ✓ assembling, compiling and linking source code
- ✓ working with time critical signals
- ✓ watching the variables while the program is executing
- ✓ editing a memory
- ✓ simulating the program
- ✓ sending a code to processor
- ✓ debugging with ICD protocol
- ✓ exporting and importing data
- ✓ solving the problem with hand help

Programmer installation

1. Place the programmer on stable dielectric base
2. Plug the serial cable **RS232** to PC computer and programmer
3. Plug the power supply unit and next power cable **12V** to programmer
4. Place the processor in **DIP holder** or plug the external programming cable into **ICSP socket** (chapter **ICSP connector**)
5. Lunch **MPLAB IDE** program on PC computer (the program is free and can be downloaded from web site: **http://www.microchip.com/** or **http://ajpic.zonk.pl/,** documentation of program can be downloaded too from web site). User's interface depends on installed program version, JuPic programmer works either with old version of program 5.70.40 or with a newest version 6.xx or higher 7.xx
6. Before starting to work, the chapter **"Programmer configuration"** should be read

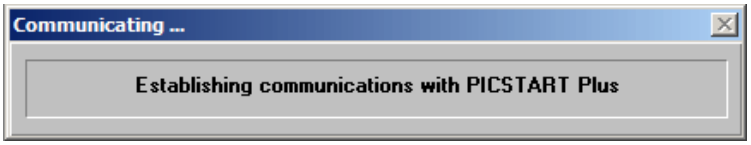
Working with MPLAB 5.70.40 version

The programmer activation should be done with options:

1. Set the type of programmer: Options → Programmer Options → Select Programmer… → **PICSTART Plus Dev. Programmer**
2. Set the port the programmer will be work with: Options → Programmer Options → Communication Port Setup… → **COMx**
3. Activate the programmer: PICSTART Plus → **Enable Programmer**

Points 1 and 2 should be done once when the first lunch of MPLAB program occurred because all the settings are saved. This settings can be change later if needed.

When the settings are completed **MPLAB** starts establishing a connection between program and programmer, this window should appear:



Pic. 1 Establishing communication with programmer

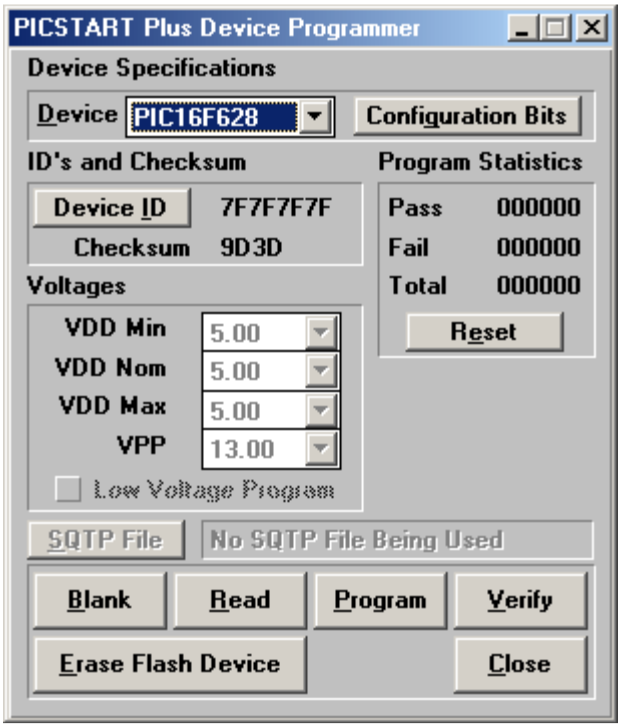
In case of any problems of transmission or configuration, MPLAB shows the error window with corresponding statement.

Next appear the firmware version of JuPic programmer.

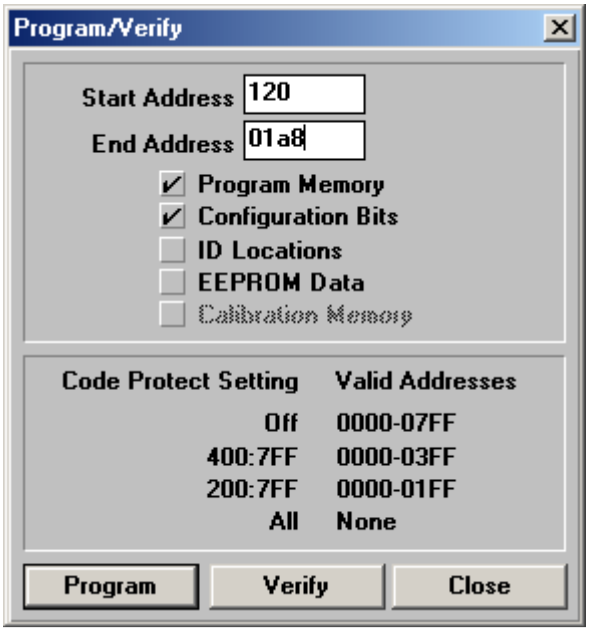
All descriptions which are displayed as PICSTART Plus programmer but refers to JuPic programmer of course.

Pic. 2 Getting the version of firmware

After establishing connection the user interface should be appear (Pic. 3) and all functions of programmer are ready to use:



Pic. 3 User interface



Pic. 4 Part memory programming window

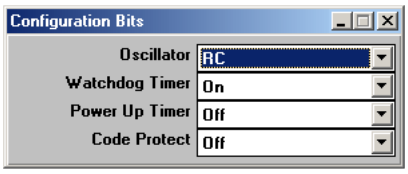
The buttons are:

- **Device — choosing the processor**
- **Device ID** — setting the ID bits
- **Configuration Bits** — setting the Configuration bits
- **Blank** — processor blank check
- **Read** — read the processor code
- **Program** — write the processor code
- **Verify** — code verification
- **Erase Flash Device** — erasing the processor, JuPic programmer has this function implemented in hardware (button **„ERASE"**) working independently from MPLAB.
- **Close** — closing the interface

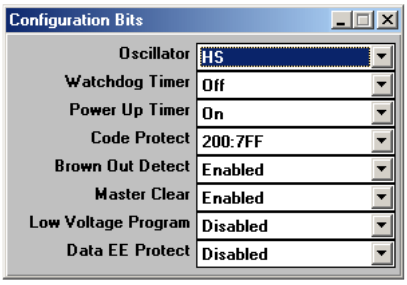
Processor can be program with "Program" button (Pic. 3), after this action all available processor memory will be changed: program memory (FLASH or EPROM), data memory (EEPROM), ID memory (ID Locations) and configuration memory (Configuration Bits).

If only a special part of memories must be program the range memory window should be choose Menu → PICSTART Plus → Program/Verify…, and next set the right parameters (Pic. 4). Program interface can be use to set program memory ranges too.

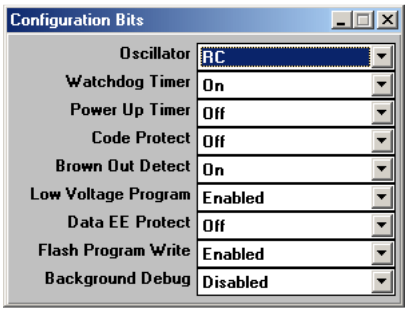
Very important stage of programming is preparing the configuration bits. That bits can be set in „Configuration Bits" memory window. Each processor has a different range of settings according to specification:



Pic. 5 PIC16F84

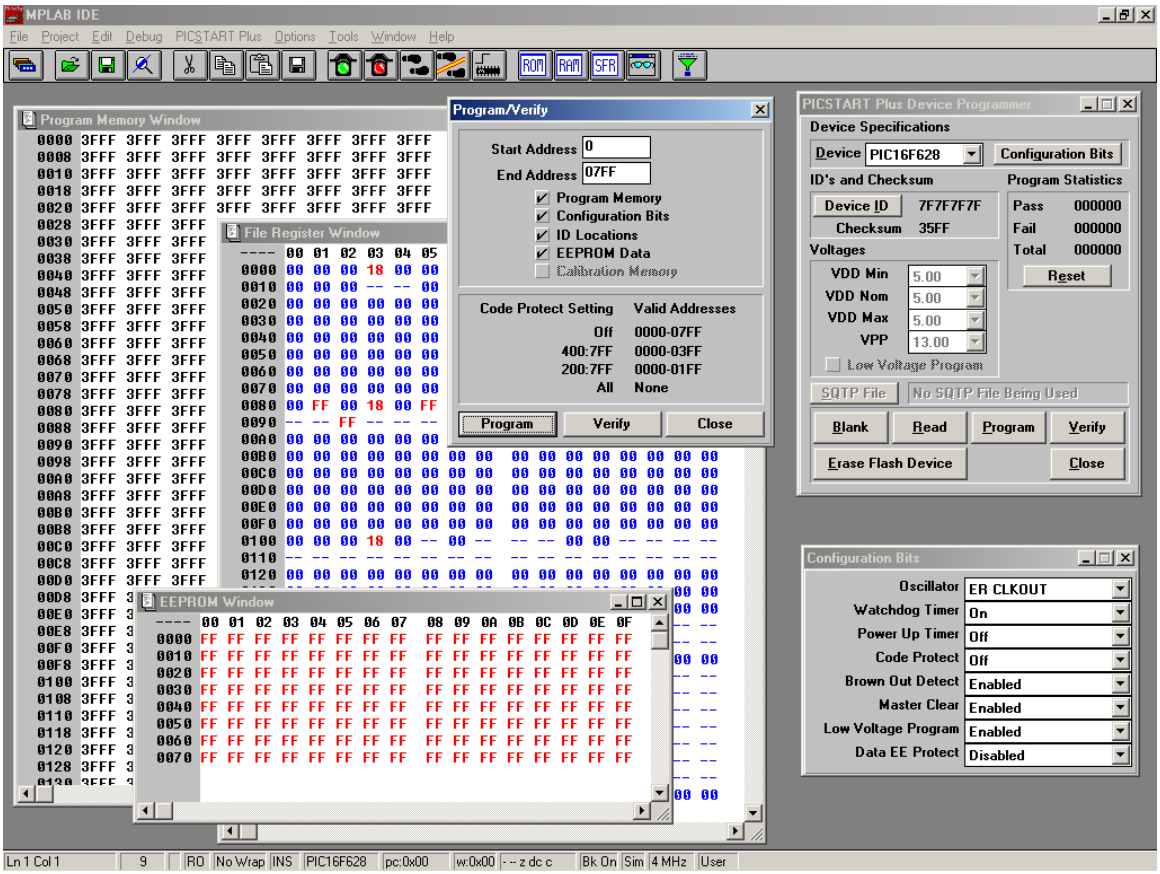


Pic. 6 PIC16F628



Pic. 7 PIC16F877

MPLAB environment is easy configurable and user can reach to all the functions and needs. This picture below shows the look of some project:



Pic. 8 Full user interface

Working with MPLAB 6.xx and 7.xx version

The programmer activation should be done with options:

1. Set the type of programmer: Programmer → Select Programmer → **PICSTART Plus**
2. Set the port of programmer: Programmer → Settings… → Communication → **COMx**
3. Activate the programmer: Programmer → **Enable Programmer**

That steps shown above (except enable) should be set only once at the first startup because MPLAB can remember all parameters. This settings can be also change later if needed.

When enable function is called **MPLAB** sends a request to **JuPic** and after a short while programmer is ready to use. Initialization is completed when device bar stays active:

Pic. 9 Programmer functions

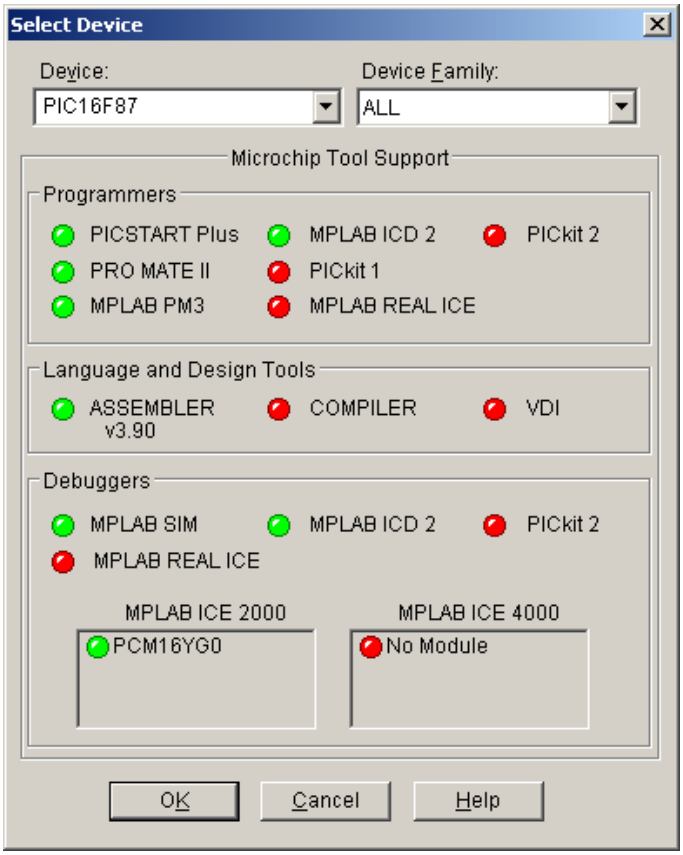
Calling the functions (from left side):

- **Blank Check** — processor blank check
- **Read** — read the processor code
- **Program** — write the processor code
- **Verify** — code verification
- **Erase Flash Device** — erasing the processor, JuPic programmer has this function implemented in hardware (button „**ERASE**") working independently from MPLAB.

The right side of abave bar shows a count of succeed writes (**Pass**), wrong writes (**Fail**) and total writes (**Total**) processor memories.

From this moment the programmer is ready to use and the "Workspace" can be build. Before programming the processor type must be set which is used in project: Configure → Select Device… → **Device** (Pic. 10)

A green led indicator (PICSTART Plus) shows if the processor can be use with this programmer (a list of processors work with JuPic programmer can be read below in last chapter), a yellow led shows limited capabilities with this part (beta support), and a red led shows that this processor can't be programmed by this tool.

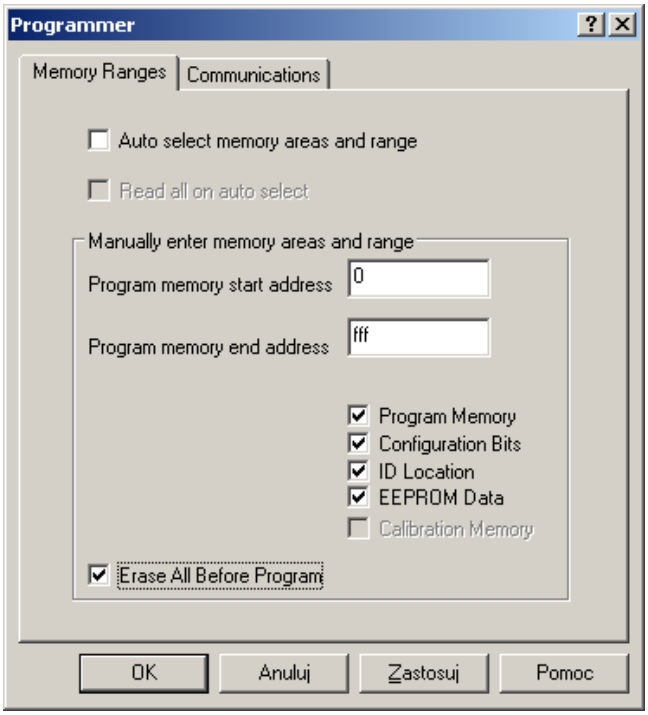


Pic. 10 Processor choosing window

After configuring this settings MPLAB is ready to program a connected processor.

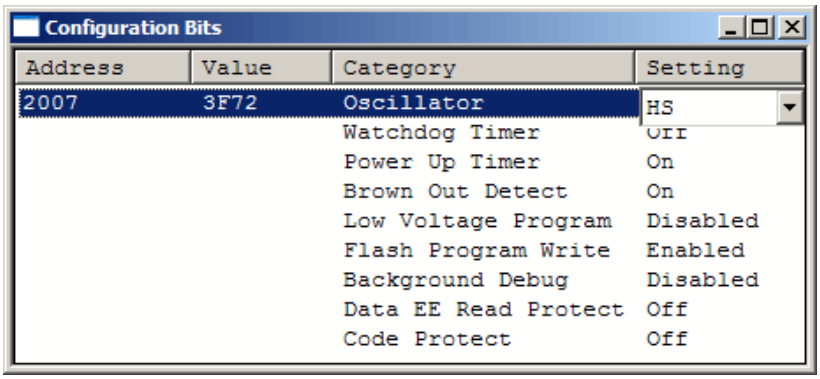
Processor can be program with "**Program**" button, after this action all available processor memory will be changed: program memory (FLASH or EPROM), data memory (EEPROM), ID memory (ID Locations) and configuration memory (Configuration Bits), depends on auto memory range calculation.

If only a special part of memories must be program the range memory window should be choose: Programmer → Settings... → **Memory Ranges** (Pic. 11) and set a required parameters. Program interface can be used to set program memory ranges too. All settings are saved in project and actived at next programming call.

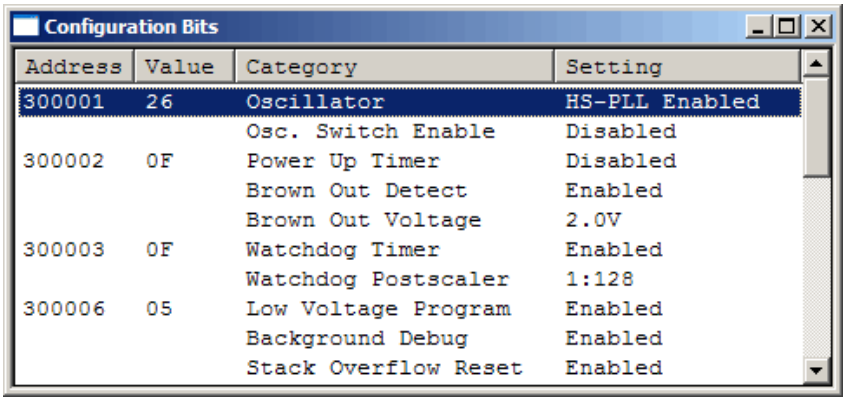


Pic. 11 Setting the range of memory programming

Before program the processor the configuration bits should be set: Configure → **Configuration Bits**... (Pic. 12 and Pic. 13)

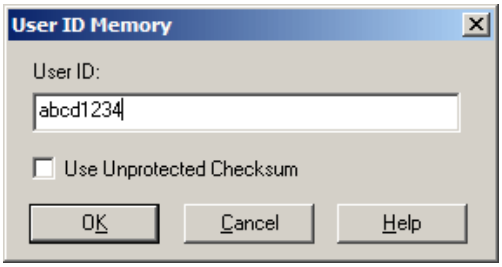


Pic. 12 Setting the configuration bits of 16F876



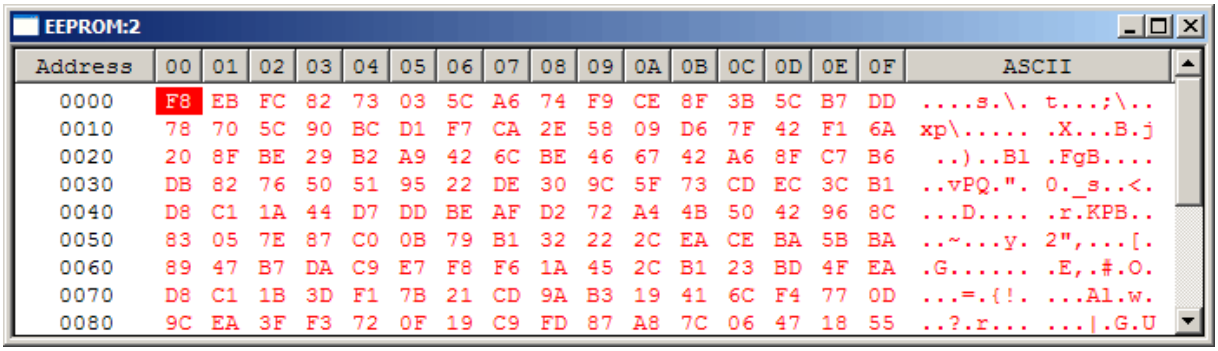
Pic. 13 Setting the configuration bits of 18F458 (part)

Settings of ID memory can be easy changed by calling menu window: Configure → **ID Memory**... (Pic. 14)



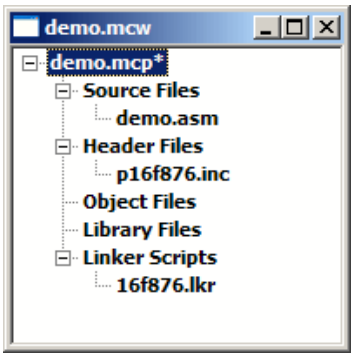
Pic. 14 Processor ID memory

The EEPROM memory can be easy modify by calling menu window: View → **EEPROM** (Pic. 15)



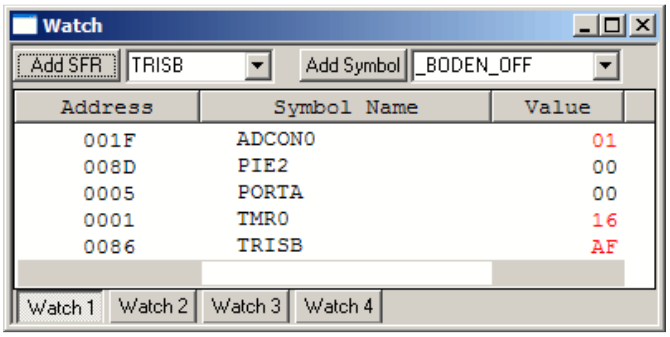
Pic. 15 Processor EEPROM memory

MPLAB is design to work with very advanced projects and it is the easiest way to build an application. The files in project's structure is very readable for user. A tree of project is shown below (Pic. 16):

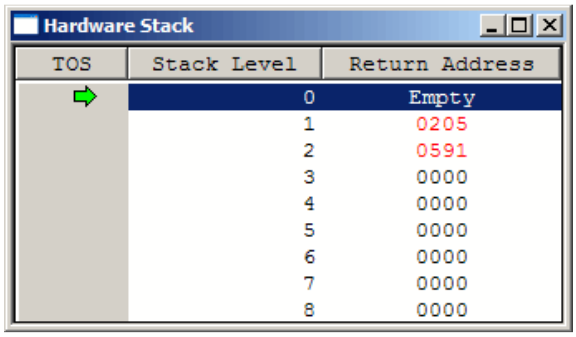


Pic. 16 Building the project

MPLAB supports very useful and helpful tools either for debugging and simulating program code Debugger → Select Tool → **MPLAB SIM** or watching the registers and stack pointer of program:

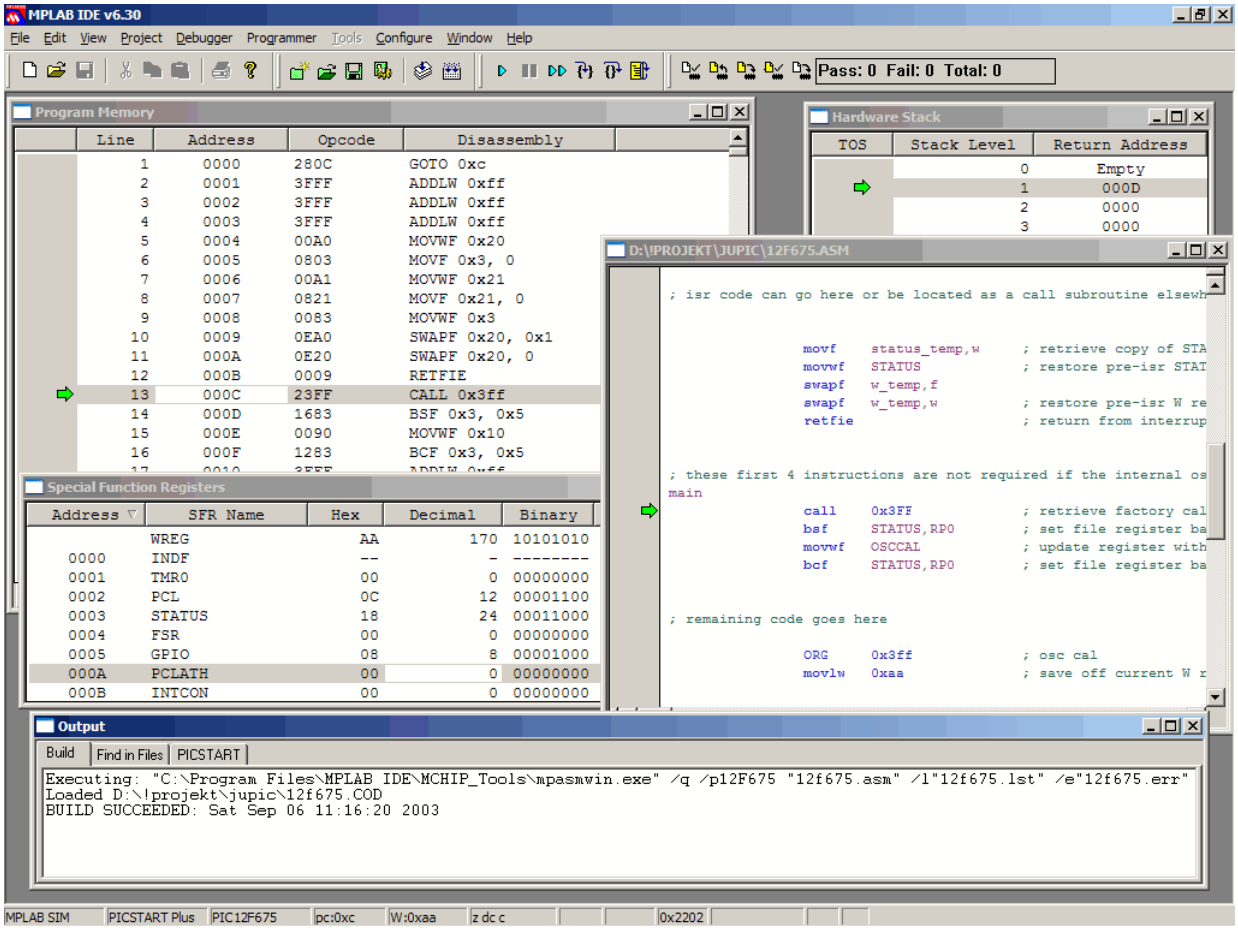


Pic. 17 Watching processor registers



Pic. 18 Watching program stack

MPLAB has build a new editor where syntax is highlighted, and the program code is more readable.



Pic. 19 Full user interface

Programmer functions

Programmer has many functions expanding its ability:

- ✓ device has build in an erase button, which can be use to erase the processor memories without lunching MPLAB and connecting programmer to computer (even Code Protection), to call this function an „**ERASE**" button must be pushed and hold down for 2 seconds.
- ✓ an information LED diode "**PROG**" shows a state of device:
  - two double short blink — power on an device initiatializing
  - diode is on — programmer is ready to use
  - diode is fast blinking — establishing communication with **MPLAB**
  - diode is slow blinking — data transmission is in progress with **MPLAB** (read/write)
  - diode is off — an "**ERASE**" button is pushed and hold down
  - three short blink — the processor was erased
  - diode generates periodically three short blink — the programmer is in trap state, there's no active code or the code is damaged and new firmware must be downloaded
- ✓ an external **ICSP** (In-Circuit Serial Programming) connector was build in, a processor can be programmed in designed board
- ✓ **LVP** mode can be activated and use
- ✓ programmer has build in a **DIP18** socket, which may be use to fast program a stand alone processor without any additional connecting cables. This socket is designed to work with **DIP18**, **DIP14** and **DIP8** compatible processors too (excluding PIC10FXXX).
- ✓ a configuration jumpers can easy suits the programmer to own hardware requirements
- ✓ programmer can be connected directly to serial computer port **COM** without serial cable
- ✓ ICs are place with additional sockets and can be easy replace with a new ones in case of damage.
- ✓ a write procedures was optimized and work faster now.
- ✓ a feature of upgrading new firmware through serial port was added.

Connecting serial cable RS-232

JuPic programmer can be connected with the computer through serial connector RS-232C and straight through cable (**modem**), which is the same for original PICSTART Plus programmer.

The cable for serial connection has one male plug **DB-9M** at programmer side and a female one **DB-9F** or **DB-25F** at computer side. This cable can be hand made; the signal used by programmer are listed below in table.

Signal	DB-25	DB-9	Direction	DB-9	Signal
	Female		PC — JuPic	Male	
TX	2	3	→	3	RX
RX	3	2	←	2	TX
DTR	20	4	→	4	Data Ready
GND	7	5	—	5	GND
DSR	6	6	←	6	pull up +5V
RTS	4	7	→	7	CTS
CTS	5	8	←	8	RTS

Table 1 Signals of serial cable for programmer

For serial connection a programmer is communicating with standard serial RS232 protocol speed **19200Kb/s** and **8N1** frame. Data sending is half duplex mode with hardware flow control handshaking by lines **RTS** and **CTS**. New firmware has impemented a new software FIFO which should resolve the problems with some notebooks and Windows XP low level serial port drivers.

Programmer configuration

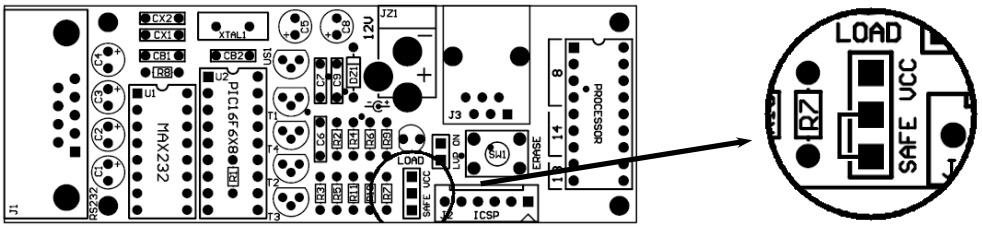
There are 2 modes for programming the processor:

- ✓ high voltage HVP (14V) — jumper „LVP ON" open
- ✓ low voltage LVP (5V) — jumper „LVP ON" short

There are 4 modes for power the device:

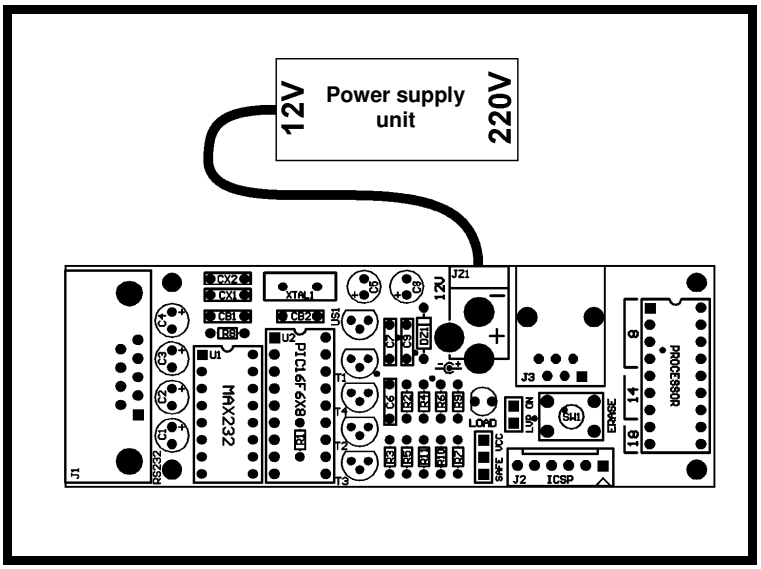
1 SAFE mode

This mode is activated when "SAFE" jumper is short



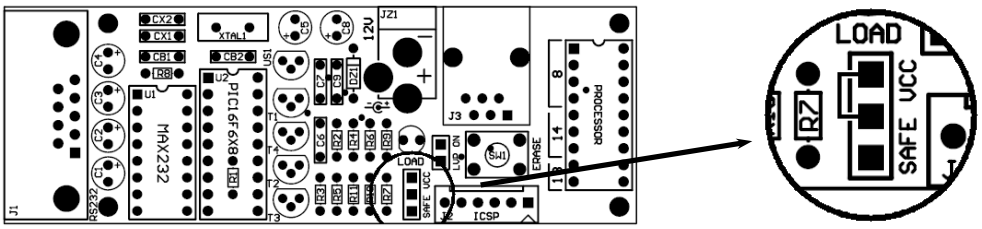
Pic. 20 SAFE mode configuration

This mode can be used for the most safe processor programming. The processor power is activate only while programming is in progress (read/write) and after transmission a power is cut off. It's not recommended to power an external circuit in this mode through **ICSP** connector because the transistor switch can damage this way.



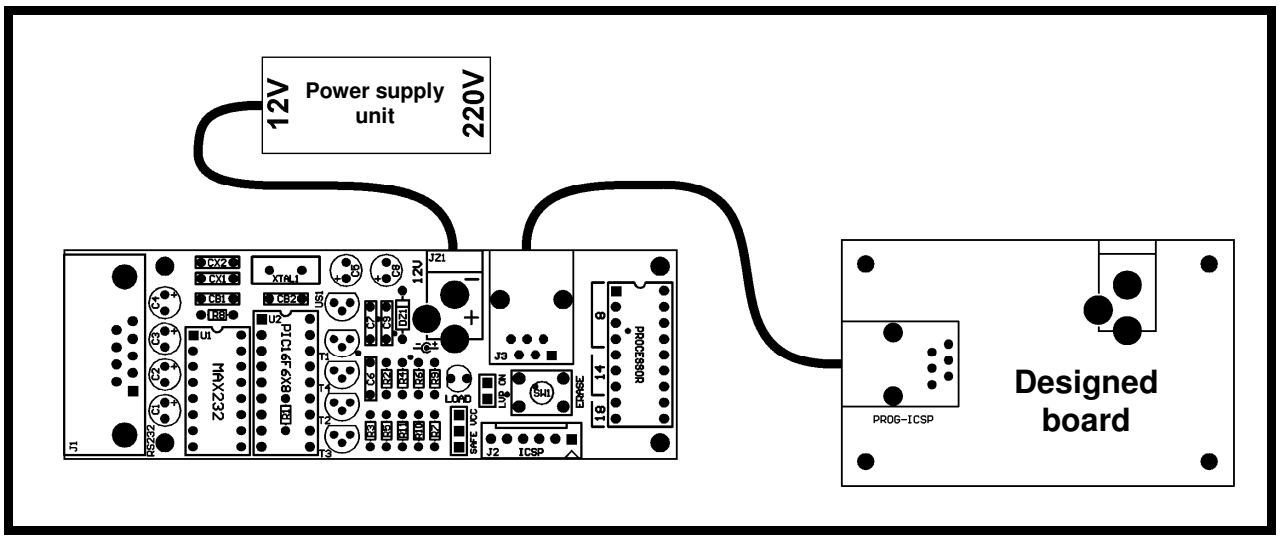
Pic. 21 SAFE mode connection

2 VCC mode — with directly power
This mode is activated when "VCC" jumper is short



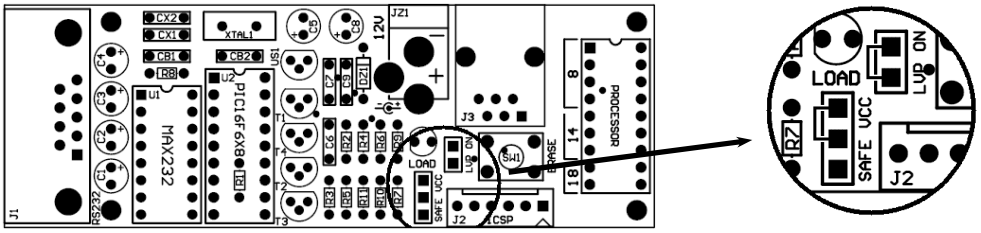
Pic. 22 VCC mode configuration

This mode can be use to power an external circuit or processor from programmer board. A voltage is present for all time through the ICSP connector. This way an external circuit don't need a additional power supply unit. There is a one very important restriction when use this mode because a current efficiency of programmer stabilizer in not to much (about 100mA) and only a low current circuits can be powered this way.



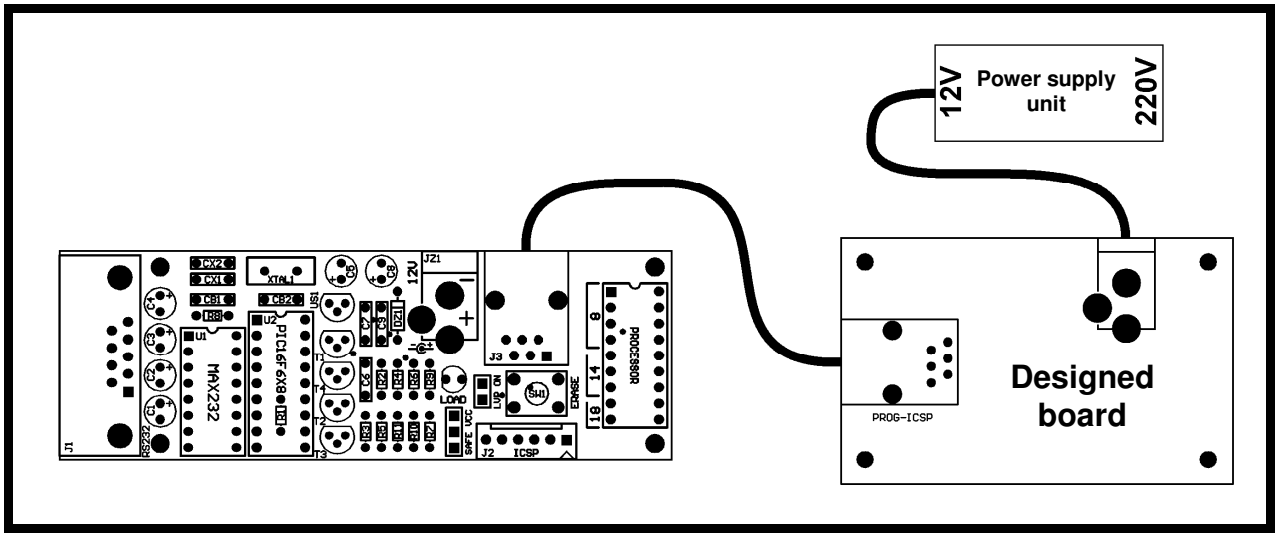
Pic. 23 VCC mode connection

3 Dependent Mode — without power supply unit
This mode is activated when "VCC" and "LVP ON" jumpers are short



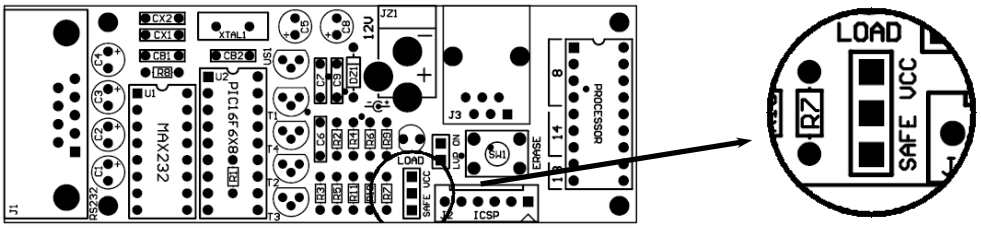
Pic. 24 Dependent mode configuration

This mode can be use to work programmer without connecting a power supply unit. The voltage for power the programmer (5V) is taken from designed board through **ICSP** connector. Using this mode is only designed for processors which can be program with low voltage (**LVP**) and also it's not allowed to power a programmer in this mode.



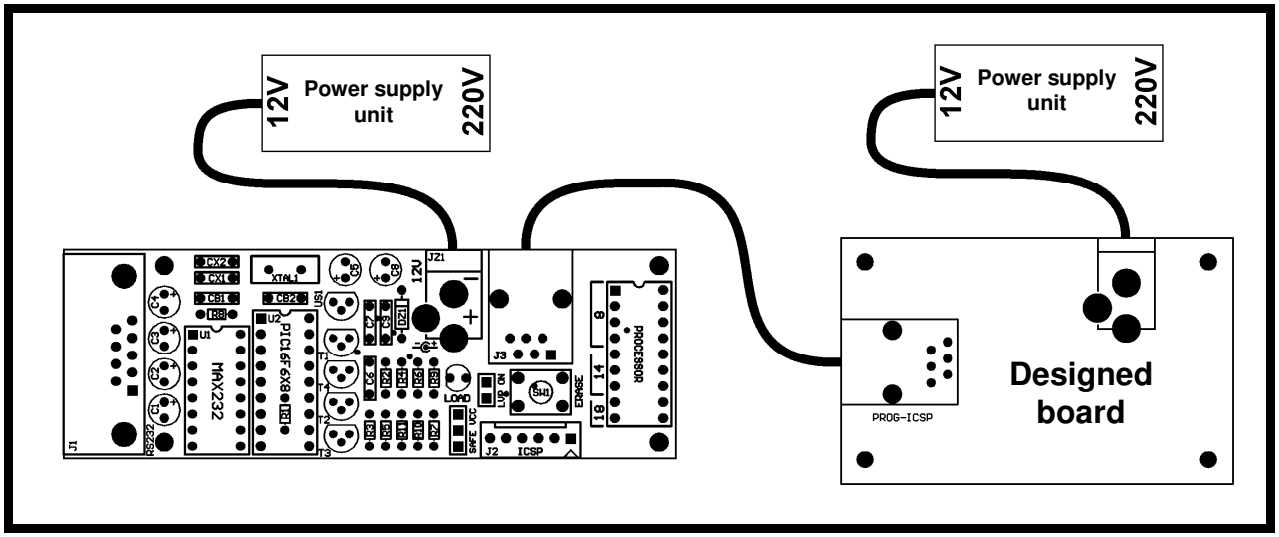
Pic. 25 Dependent mode connection

4 Independent Mode — with two power supplies units
This mode is activated when "SAFE" and "VCC" jumpers are open



Pic. 26 Independent mode configuration

This mode can be used when programmer and designed board are independent powered through different power supply units. A programmer is powered with its own stabilizer and the processor is powered from designed board. This mode is most often use with ICSP because this way a designed board has no restrictions for current load and others modules can be easy connected to designed system.

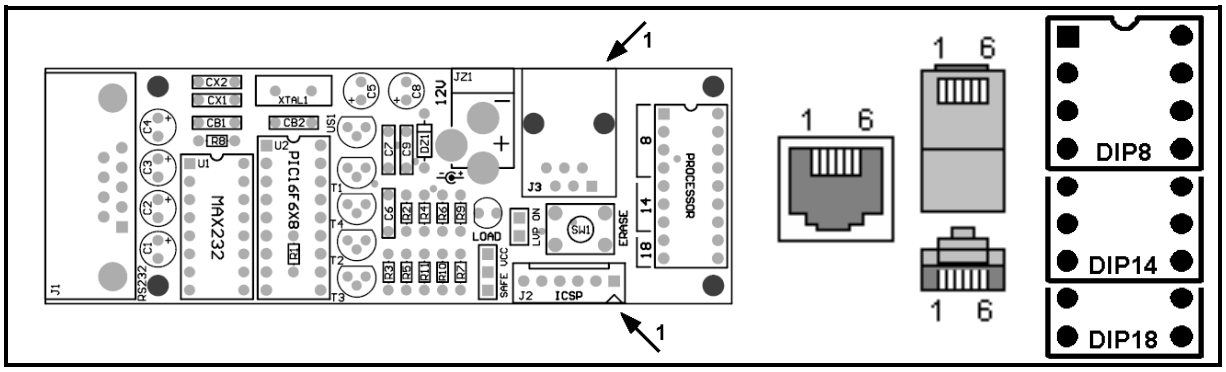


Pic. 27 Independent mode connection

ICSP connector

The programmer has two 6 pin **ICSP** connectors "**SIP6P**" and "**RJ-12**" type, which can be use to programming processor at the designed board without placing it to programming socket (Pic. 28).

There is also installed a precision 18 DIP socket which can be use for directly programming a stand alone processors: PIC12FXXX, PIC12CXXX, 14 pins: PIC16F6XX and 18 pins: PIC16CXXX, PIC16FXXX, PC18FXX20. The way of placing processors in socket is shown below Pic. 28. Detailed ICSP connector description is shown in Table 2. Programming a bigger processors can be done with external DIP adapter or programming through ICSP connector. It is possible to install a **ZIF** socket in place of standard precision socket too.

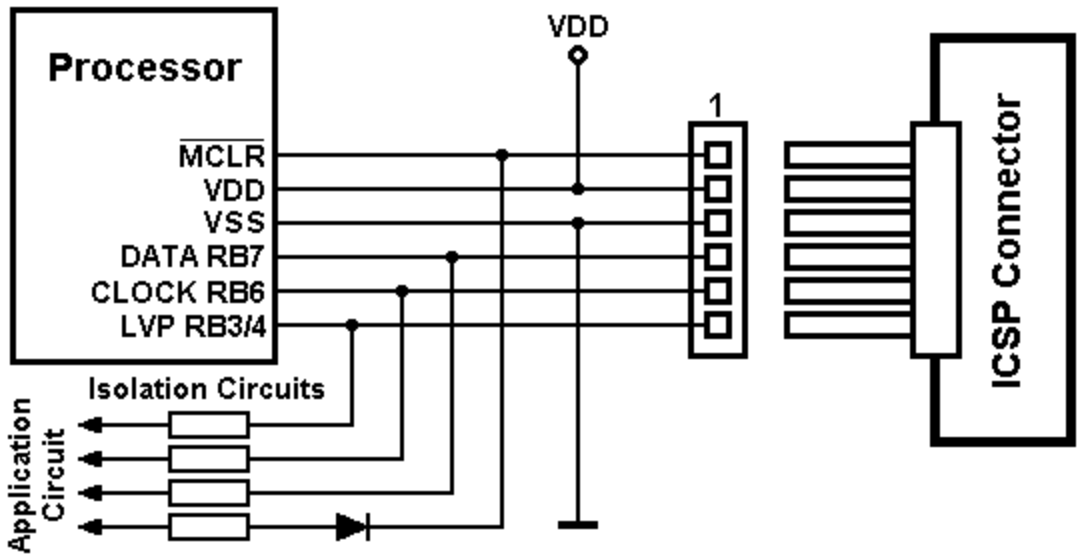


Pic. 28 ICSP connector description, pin 1.

While working with **ICSP** connector and designed board, the some important rules must be kept:

- ✓ **"MCLR"** signal should be connected directly to processor, if external power on reset is required (RC delaying circuit) this line must be cruelly separate with signal diode (e.g. 1N4148) or single resistor (1K) (Pic. 29) because in other case the processor can't be enter to programming mode.
- ✓ **"DATA"** line and **"CLOCK"** line should be also connected directly to processor. That lines can be use as usual I/O port if designed board won't input a noise signal. The most simple way in this case is connecting a two microswitches to lines **RB6** and **RB7**, which are normally open while the programming is in progress. When programming is completed this lines are cut off from board.

- ✓ **"LVP/PGM" line** can be use for programming the processor in low voltage mode. Port RB3, 4 or 5 (processor depended) and can't be the same use as usual I/O line.
- ✓ power supply modes are shown in "**Programmer configuration**" chapter.



Pic. 29 Connecting the programmer through ICSP connector

Pin	Signal	Port
1	VPP	MCLR
2	VCC	VDD
3	GND	VSS
4	DATA	RB7
5	CLOCK	RB6
6	PGM/LVP	RB3/4/5

Table 2 Pinout of ICSP connector (Microchip standard)

The programming protocol use in this device has implemented all time critical and electric parameters needed for regular processors operations and is restricted with **Microchip®** technical documentation.

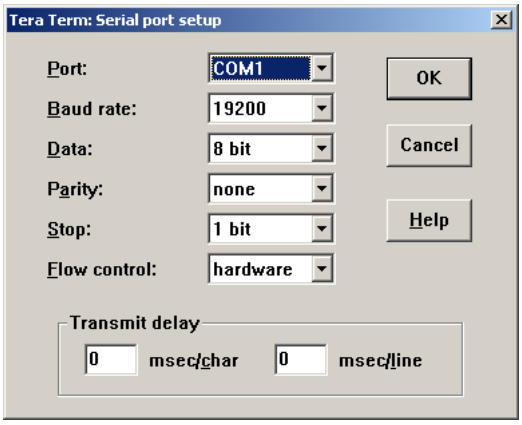
Full description of this programmer in PDF file can be downloaded from web site: **http://ajpic.zonk.pl/**

Upgrading the programmer

The programmer has built a bootloader module and can be easy upgrade with a new firmware version of code. There are two steps for reprogramming the processor.

Step 1 Preparing and configuring PC hardware and software

- ✓ Install a free serial terminal (e.g. **Tera term Pro**, can be downloaded from web site: **http://ajpic.zonk.pl/download/ttermp23.zip**)
- ✓ Launch the terminal and configure parameters of port COMx: Menu → Setup → Serial port… → **COMx, 19200, 8n1, hardware**.



- ✓ Save the settings: Menu → Setup → Save setup… → '**teraterm.ini**'

After configuring all of required parameters, a test of connection should be done:

1. Connect the programmer to computer
2. Power the programmer
3. Launch the serial terminal
4. Push „**s**" (serial) key on computer keyboard

The serial number should be displayed on terminal, if so, the connection is OK and the programmer can be upgraded in next step.

Programmer can be upgraded under **Linux** too, just configure the port and use a serial terminal (e.g. minicom –oL)

Step 2 Installation a new software to programmer.

1. Power on the programmer
2. Lunch the terminal
3. Push and hold „**ERASE**" button on programmer
4. Press "**u**" (upgrade) key on terminal
5. Entering into programming mode will be signalized by lighting up the „**PROG**" diode and displaying „**:**" character on terminal
6. Push „**l**" (load) key, active code in processor is erased, terminal displays prompt character „**>**" and programmer waits for a file.
7. Send file: Menu → File → Send file…→ '**jupic-x.xx.x.hex**', where **x.xx.x** is current version of firmware.

While the new code is loading the terminal displays a progress bar, and the LED „**PROG**" blinks on programmer. After succeeded operation, a new firmware is ready. „**OK!**" string appear on terminal and the programmer automatically starts with a new code. From this moment the upgrade is finished and the programmer is ready to use.

If something goes wrong while upgrading is in progress and some troubles take place (e.g. broke a transmission, power off, loading a wrong code and other bad things) and the programmer can't properly work — don't worry just use **Recovery mode** described below.

Programmer with no active code indicates this state with „**PROG**" diode with three cyclic blinks and terminal shows „**ERROR!**" string, when wrong hex file was sent the programmer can be unpredictable.

Recovery mode:

1. Power off the programmer
2. Lunch the terminal
3. Push and hold „**ERASE**" button on programmer
4. Power on the programmer
5. Programmer going to enter into regular upgrade mode (see point 5)

Listing of supported processors

PIC10F200
PIC10F202
PIC10F204
PIC10F206
PIC10F220
PIC10F222

PIC12F508
PIC12F509
PIC12F510
PIC12F609
PIC12F615
PIC12F629
PIC12F635
PIC12F675
PIC12F683

PIC16F505
PIC16F506
PIC16F54
PIC16F57
PIC16F610
PIC16F616
PIC16F627
PIC16F627A
PIC16F628
PIC16F628A
PIC16F630
PIC16F636
PIC16F639
PIC16F648A
PIC16F676
PIC16F684
PIC16F685
PIC16F687
PIC16F688
PIC16F689
PIC16F690
PIC16F716
PIC16F72
PIC16F73
PIC16F737
PIC16F74
PIC16F747
PIC16F76
PIC16F767
PIC16F77
PIC16F777
PIC16F785
PIC16F818
PIC16F819
PIC16F83
PIC16F84
PIC16F84A
PIC16F87
PIC16F870
PIC16F871
PIC16F872
PIC16F873
PIC16F873A
PIC16F874
PIC16F874A
PIC16F876
PIC16F876A
PIC16F877
PIC16F877A
PIC16F88
PIC16F883
PIC16F884
PIC16F886
PIC16F887
PIC16F913
PIC16F914
PIC16F916
PIC16F917
PIC16F946

PIC18F242
PIC18F248
PIC18F252
PIC18F258
PIC18F442
PIC18F448
PIC18F452
PIC18F458
PIC18F1220
PIC18F1230
PIC18F1320
PIC18F1330
PIC18F2220
PIC18F2221
PIC18F2320
PIC18F2321
PIC18F2331
PIC18F2410
PIC18F2420
PIC18F2423
PIC18F2431
PIC18F2450
PIC18F2455
PIC18F2480
PIC18F2510
PIC18F2515
PIC18F2520
PIC18F2523
PIC18F2525
PIC18F2550
PIC18F2580
PIC18F2585
PIC18F2610
PIC18F2620
PIC18F2680
PIC18F4220
PIC18F4221
PIC18F4320
PIC18F4321
PIC18F4331
PIC18F4410
PIC18F4420
PIC18F4423
PIC18F4431
PIC18F4450
PIC18F4455
PIC18F4480
PIC18F4510
PIC18F4515
PIC18F4520
PIC18F4523
PIC18F4525
PIC18F4550
PIC18F4580
PIC18F4585
PIC18F4610
PIC18F4620
PIC18F4680
PIC18F6620
PIC18F6720
PIC18F8620
PIC18F8720

PIC12C508
PIC12C508A
PIC12C509
PIC12C509A
PIC12C671
PIC12C672
PIC12CE518
PIC12CE519
PIC12CE673
PIC12CE674

PIC14C000

PIC16C505
PIC16C54C
PIC16C55
PIC16C554
PIC16C558
PIC16C55A
PIC16C56
PIC16C56A
PIC16C57
PIC16C57C
PIC16C58A
PIC16C58B
PIC16C620
PIC16C620A
PIC16C621
PIC16C621A
PIC16C622
PIC16C622A
PIC16C62A
PIC16C62B
PIC16C63
PIC16C63A
PIC16C642
PIC16C64A
PIC16C65A
PIC16C65B
PIC16C66
PIC16C662
PIC16C67
PIC16C71
PIC16C710
PIC16C711
PIC16C712
PIC16C715
PIC16C716
PIC16C717
PIC16C72
PIC16C72A
PIC16C73A
PIC16C73B
PIC16C745
PIC16C74A
PIC16C74B
PIC16C76
PIC16C765
PIC16C77
PIC16C770
PIC16C771
PIC16C773
PIC16C774
PIC16C781
PIC16C782
PIC16C923
PIC16C924
PIC16C925
PIC16C926
PIC16CE623
PIC16CE624
PIC16CE625

Notes