



AVRPascalDbg

Manual

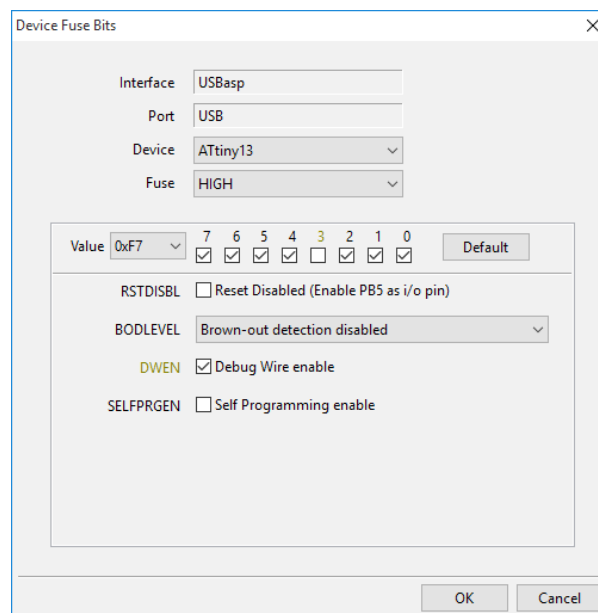
version 20/07/2024

1. Introduction

AVRPascalDbg is a debugger for AVR Pascal. It is distributed as a plug-in for the application (*avrpasdbg.dll* on Windows, *libavrpasdbg.so* on Linux) along with a license file (*.lic), placed in the system using a dedicated installer¹. These files should be located in the AVR Pascal binaries directory, i.e.:

- for Windows: *c:\Program Files\AVRPascal\bin\win64²*
- for Linux: */opt/AVRPascal/bin/linux64*

AVRPascalDbg uses the DebugWire (DW) protocol to communicate with microcontrollers. To work with AVR PascalDbg, the microcontroller must have the DWEN fuse bit enabled. This can be done, for example, using the UABasp programmer within AVR Pascal.



- 1 The debugger version must be compatible with the version of the installed AVR Pascal, i.e. the first two digits separated by a dot must be identical. In case of Linux systems, if both the AVR Pascal editor and the AVR PascalDbg debugger are installed and there is a need to install a new version of the editor, you must first uninstall the debugger, otherwise the installer will inform you about the violation of dependencies between the packages. In Windows systems, you can easily overwrite previously installed programs, keeping in mind the version compatibility principle.
- 2 The folder *c:\Program Files\AVRPascal* is only suggested during installation, the user can choose another one.

After turning on DWEN, the programmer should be disconnected³.

2. Adapter

AVRPascalDbg communicates with the microcontroller using a USB-UART converter, which must be installed in the operating system as a virtual serial port (VCP). AVRPascalDbg works with popular converters enabling the use of non-standard data transfer rates (baud-rates):

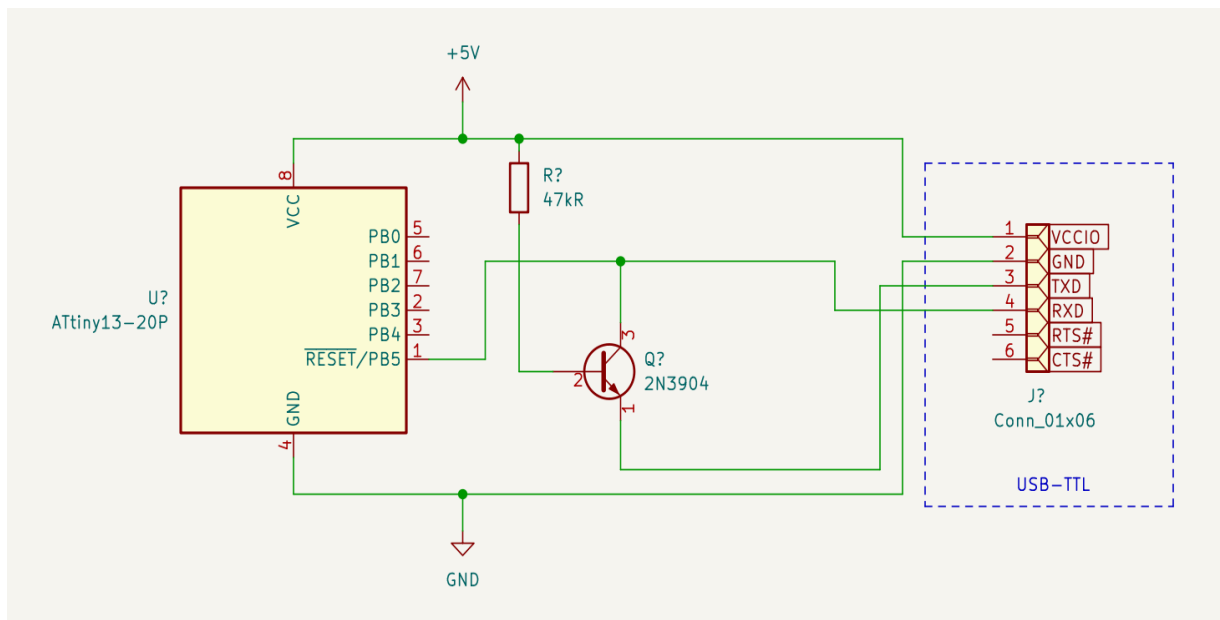
USB-UART Converter	Manufacturer	VID - PID
FT232	Waveshare	0x0403 - 0x6001
PL2303	Prolific	0x067b - 0x2303
CH340 (HL-340)	Quinheng	0x1a86 - 0x7523

For 64-bit Windows systems, using the above converters might require the installation of additional VCP drivers. These drivers are usually available for download from the manufacturers' websites:

Waveshare: <https://www.waveshare.com>

Prolific: <https://www.prolific.com.tw>

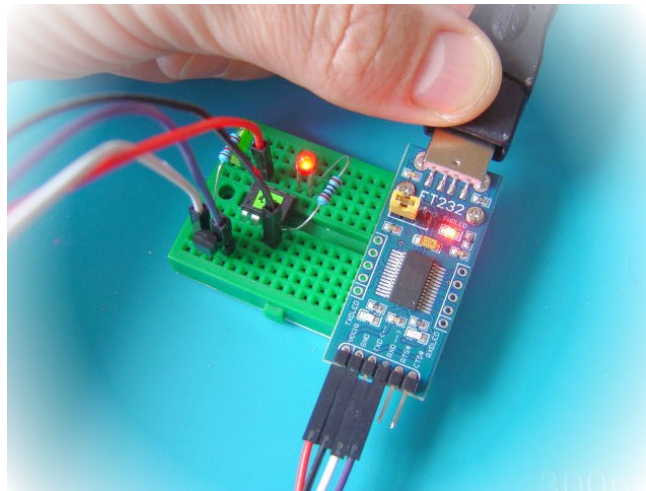
Quinheng: <https://wch-ic.com>




The VCC and GND outputs of the adapter should be connected to the corresponding VCC and GND pins of the microcontroller. The RXD output should be connected to the RESET pin through a 47 kΩ pull-up resistor, while the TXD output should be connected to the RESET pin through a 2N3904 transistor. The connection circuit (for example, with an

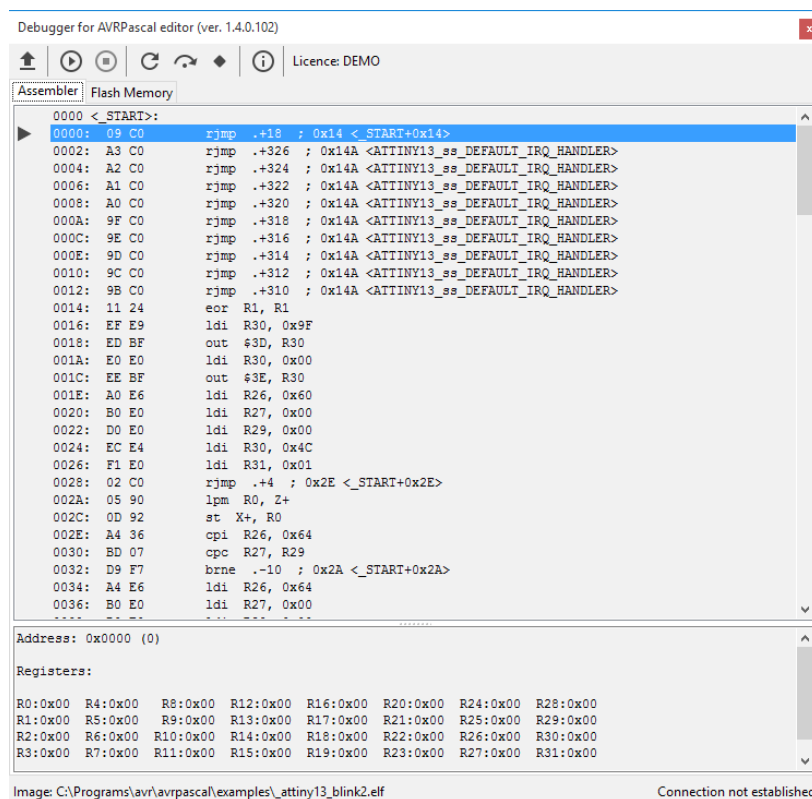
³ Enabling the DebugWire protocol using the DWEN fuse-bit prevents subsequent programming using USBasp. Therefore, dedicating a specific microcontroller only for debugging code seems like a good solution.

ATtiny13 microcontroller) is shown above. The connection can be made on a breadboard and expanded with other elements useful during debugging, such as an LED.

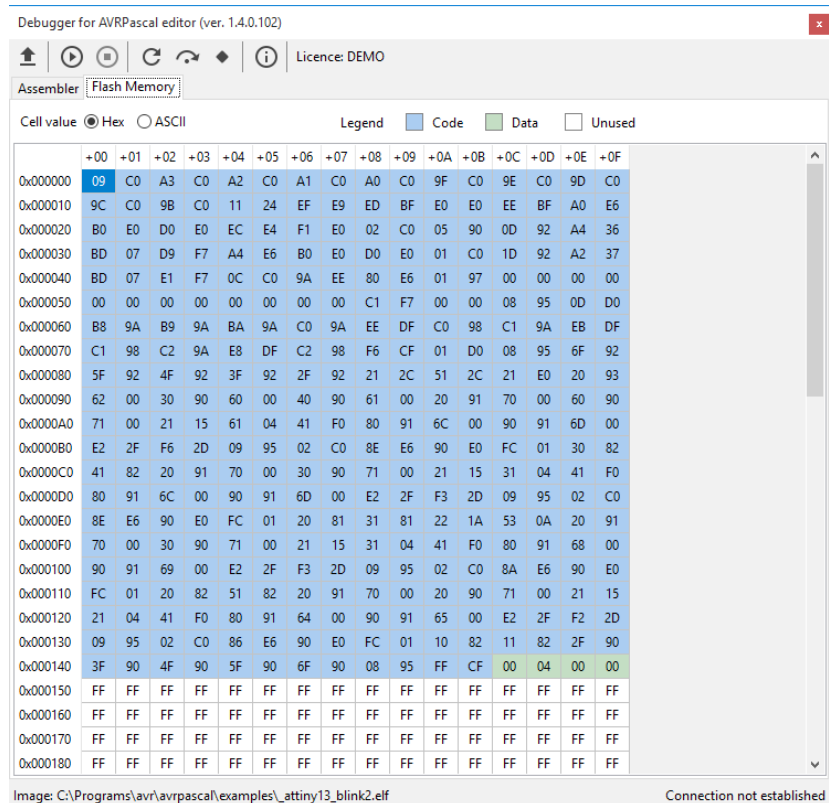


3. Debugger window

AVRPascal offers a debugger window accessible through the *View->Debug Window* menu. This window displays the loaded flash memory image from the *.elf file. The Assembler tab shows both the disassembled memory image and the current contents of the microcontroller registers. If a line of assembly code corresponds to a specific line of Pascal source code, an icon  will appear next to the corresponding address in the disassembled view.



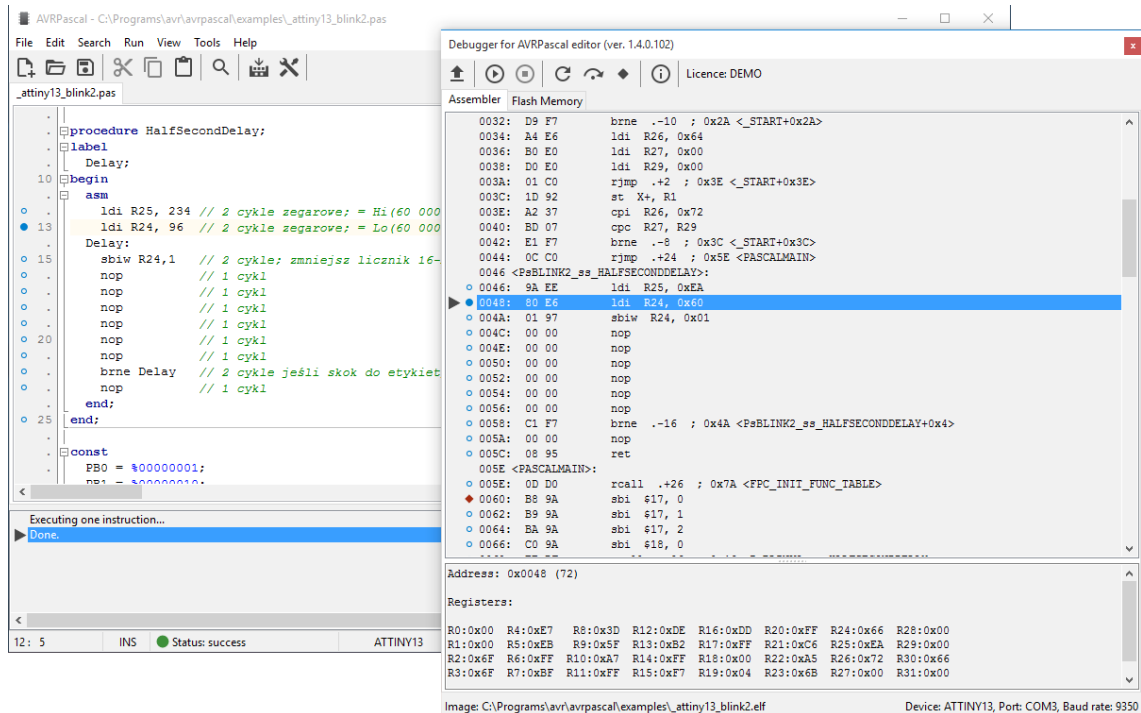
The *Flash Memory* tab displays the loaded flash memory image, differentiating code (blue background) and data (green background) sections. You can view memory bytes in either hexadecimal mode or as ASCII characters.



4. Debugging tools

The debugging tools are available in the toolbar and context menu. These tools include: *Upload* (⬆ - uploading the flash memory image into the microcontroller), *Run* (▶ - starting the debugger), *Stop* (⏹ - stopping the debugger), *Reset* (↺ - resetting the debugger), *One step* (⏪ - one step of the debugger), *Breakpoint* (◆ - breakpoint of execution) and *Config data* (ⓘ - microcontroller's configuration data). The results of each action and any error information are displayed in the AVR Pascal Messages window

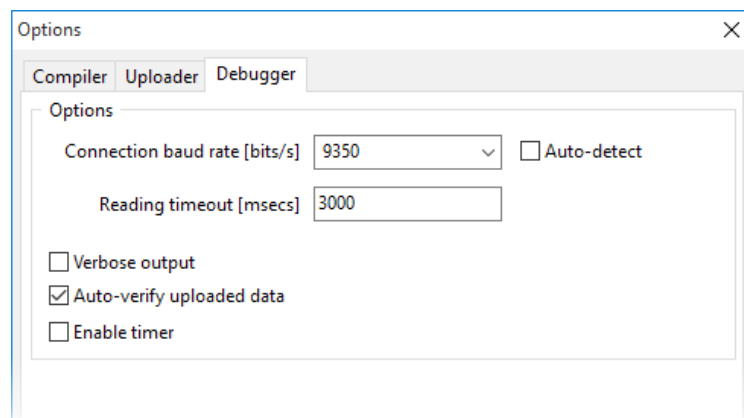
Before starting debugging, load the compiled program (referred to as the flash memory image) into the microcontroller using the *Upload* tool. This ensures that the microcontroller runs the exact code you compiled in AVR Pascal. Then, start the debugger using the *Run* tool. If no breakpoint is set, the program will run continuously until you press *Stop*. If a breakpoint is set using the *Breakpoint* tool, the program will pause execution at that specific memory location. Only one breakpoint can be active at a time, and it's marked by an icon in the memory listing by ◆ icon. To resume an interrupted program, you can either use *Run* again or use *One Step*. The *One Step* tool executes the next instruction, moving the program to the following memory address. If the current execution point corresponds to a line of Pascal source code, an icon ● will appear in both the memory listing and the AVR Pascal tab, next to the corresponding address.



The *Reset* tool is used to reset the microcontroller and return the program execution point to the beginning of the flash memory. In each case, the current content of the microcontroller registers is displayed in the lower panel of the Assembler tab. Finally, the *Config data* function shows the values of the microcontroller's fuse bits.

5. Debugger options

AVRPasDbg offers configuration options accessible in the *Debugger* tab of the AVR Pascal *Options* window. Here, you can configure the connection with the USB-UART adapter, indicating the baud-rate of the transfer (in bits/second) or its automatic detection (*Auto-detect*). Additionally, you can define a read timeout, i.e. the time after which the connection will be interrupted if there is no data transferred.



These parameters are important because an incorrectly determined data transfer rate may result in data corruption and, consequently, errors or the inability of the debugger to function correctly⁴.

The next parameters concern the ability of the microcontroller to use a timer during debugging (*Enable timer*), automatic verification of data loaded into the microcontroller's flash memory (*Auto-verify uploaded data*) and the level of detail of messages sent to the *Messages* window of AVRPascal (*Verbose output*).

6. DEMO version

AVRPasDbg is available on the website <http://akarowski.pl/> with a DEMO license (file *000B462305076250.lic*). This is a limited version of the debugger. Its only limitation is the maximum size of flash memory it can load into the microcontroller, which is 350 bytes.

Andrzej Karwowski

e-mail: ackarwow@gmail.com

⁴ The optimal baud-rate value is usually slightly higher than that determined automatically by the debugger.