ArdPicProg

User's Guide

Version 2.5
dated 11/03/15

Gregor Schlechtriem

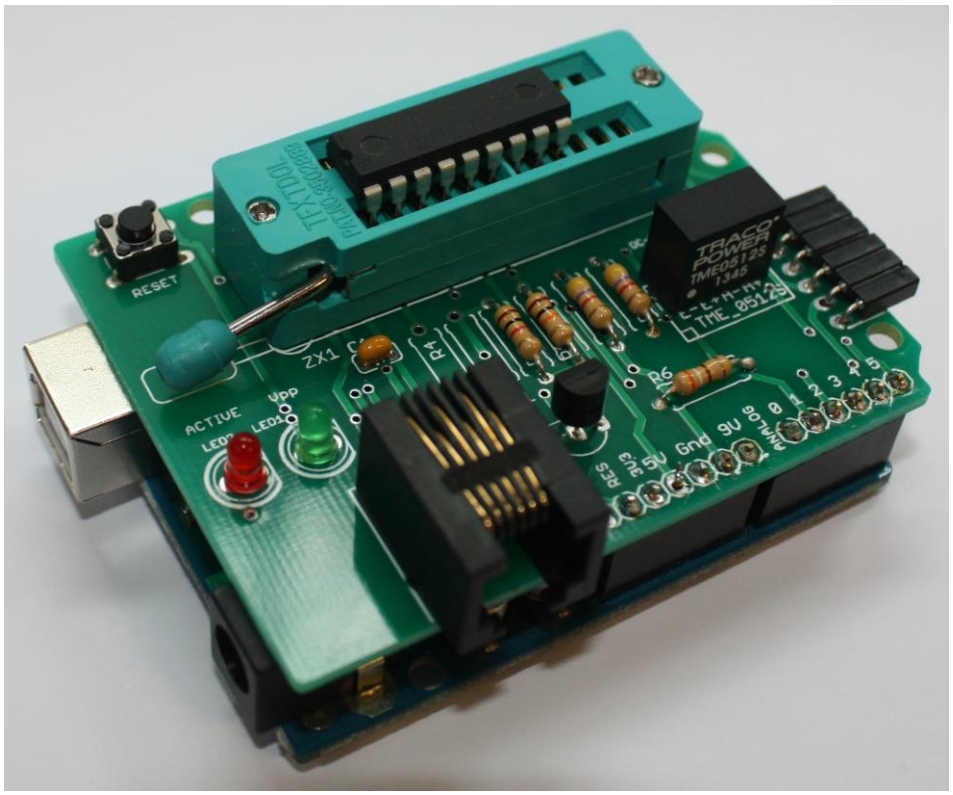
Content

1

Overview

If you would search the internet you will find quite some DIY PIC programmers. However, those designs often either require a true serial or parallel port instead of an easily available USB port or are designed around a pre-programmed controller assuming access to a programmer.

A compelling alternative would be to employ an Arduino as controller and add a limited number of additional components on a prototype shield. This makes an ArdPicProg: the Arduino Pic Programmer.



ArdPicProg has been originally designed by Rhys Weatherley (please refer to rweather.github.io/ardpicprog/index.html for more information). I made several improvements to the original solution.

- I do not like the concept of the power supply for the programming voltage provided by additional batteries or an external device. Therefore, I am deploying a readily available dc-dc converter to step up the available 5V to the required 12V.

- I did eliminate the jumper used to determine the pic type in the original design which is actually not required.

- I designed an Arduino shield seen on the photo above which simplifies building your own Arduino Pic Programmer.

- The host program user interface is line oriented and more geared towards an IT expert than an everyday user. Therefore, I developed an additional host program with a graphical user interface for Windows computers.

I also added features to the original Arduino Sketch "ProgramPic" such as supporting additional PIC controllers. These sketches will identify themselves by a version number > 1.0. Following general industry guidelines the sketch would be fully backwards compatible to the original design as long as the main version number is still 1.

With this, the complete solution is comprised of the following three components:

1. A circuit build on a prototype board for the programming logic
2. A sketch called "ProgramPic" that is loaded into an Arduino to directly interface with the device during programming. The sketch implements a simple serial protocol for interfacing with the host and can be downloaded.
3. A host program called "ArdPicProgHost" which provides a Windows-GUI, implements the serial protocol to the programmer and controls the programming process on the computer side. This software can also be downloaded (alternatively, you could download the original host software "PicProgHost.exe").

This User's Guide covers the hardware construction, the complete hard- and software setup and will illustrate the programming of a controller using the graphical UI. Additionally, the programming with the original line oriented host software is covered.

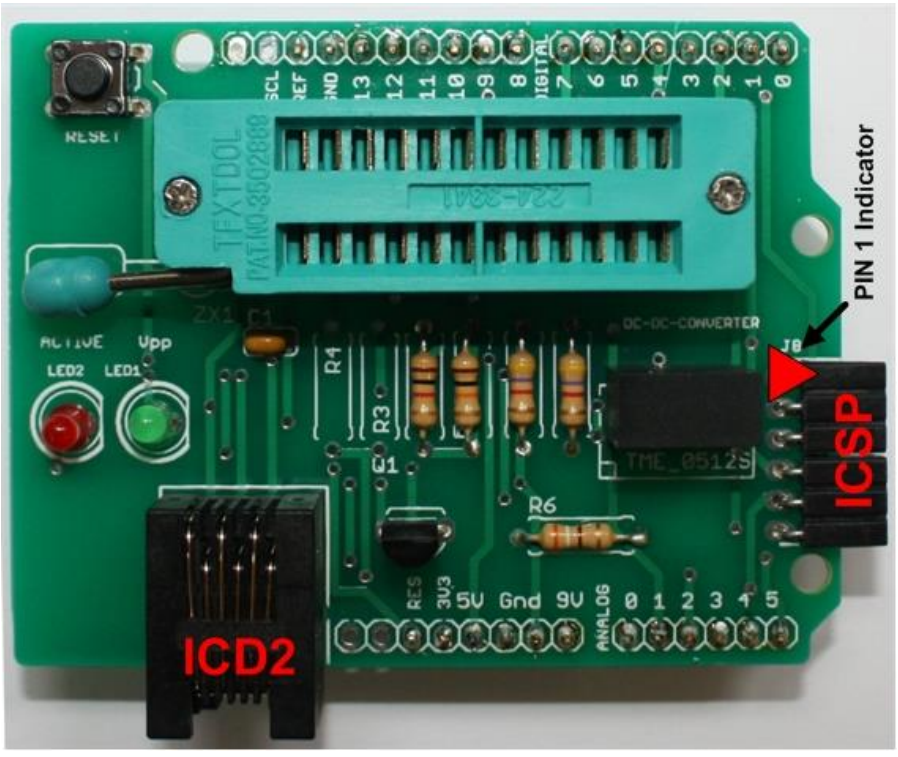
This User's Guide is based on the most recent hard- and firmware available for the ArdPicProg. Please check for updated information and the latest software releases on www.pikoder.com.

Please share with me any comments, improvement ideas or errors you will find or encounter in working with your ArdPicProg. I can be reached at webmaster@pikoder.com. Thank you very much!

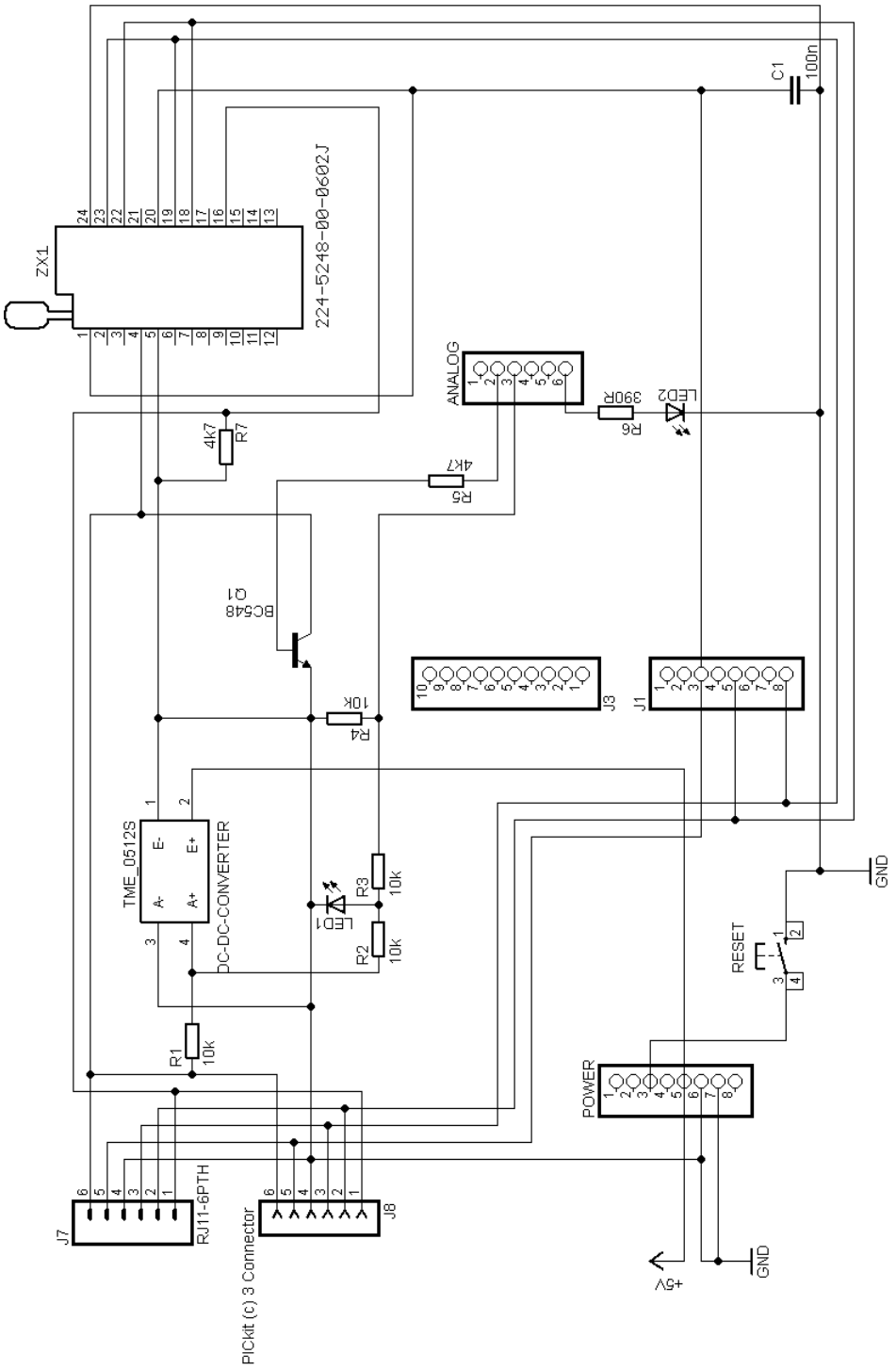
2

ArdPicProg Shield

The circuit with the programming logic is built using the ArdPicProg Shield PCB and is straight forward. Alternatively, you can also build the programmer on a prototype shield. The shield features a ZIF-programming socket, an ICD2-(RJ-11) jack and also an ICSP connector with a PIKKIT™ 3 pinnout allowing you to address a variety of programming scenarios.



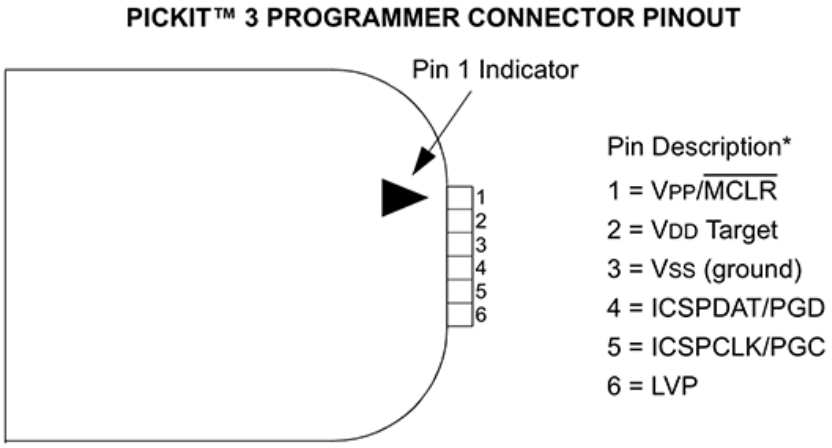
Schematics



The schematic is presented above. By employing a DC-DC-converter the complete programmer fits on a single Arduino shield. Please note that the schematic is showing two different versions. The basic kit version features the green LED1 for monitoring the programming voltage. In this version R2 becomes 1k and R3 and R4 are not required.

ICSP Pinout

The ICSP connector features the PIKKIT™ 3 pinout shown below.



* The 6-pin header (0.100" spacing) accepts 0.025" square pins.

- This page is left blank intentionally -

3

Software Overview

The next step towards programming pic controllers would be the download of the required software modules.

The latest version of the Arduino Sketch is made available for download through the ProgramPic repository on Github. This sketch supports additional PIC controllers and is fully backwards compatible to Rhys' original version. Copy the source code and paste it into the Arduino IDE, translate and upload the executable into your Arduino. Then follow the steps which Rhys is describing on his web page for commissioning your ArdPicProg.

Then you would have to setup the host (PC-) application. You can either use the Windows application "ArdPicProgHost" available at the www.pikoder.com page for downloading or apply the original host program "PicProgHost". The Windows application would be described in the following section 4.

If you decide to proceed with the original program then you could either download the source code at the github project repository and build the application yourself or you can download the executable for Windows 7 on the pikoder-webpage. The program handling would be described in section 5.

- This page is left blank intentionally -

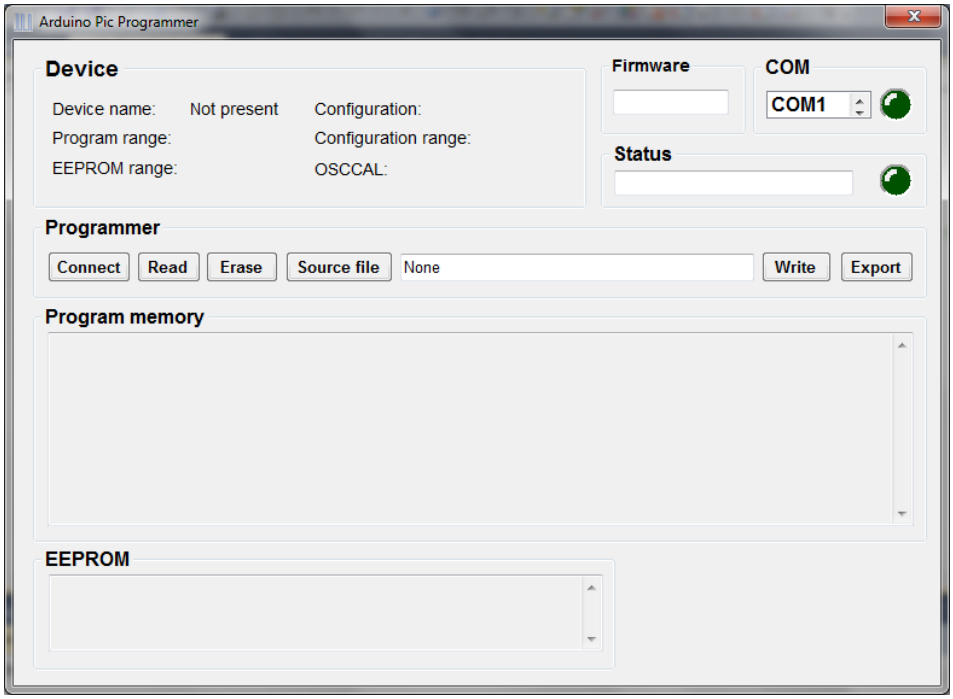
4

ArdPicProgHost application

This section describes the installation and use of the ArdPicPorgHost application.

Please download the executable application from the www.pikoder.com website. Unzip the program file. There is no installation required and the program can be started right away.

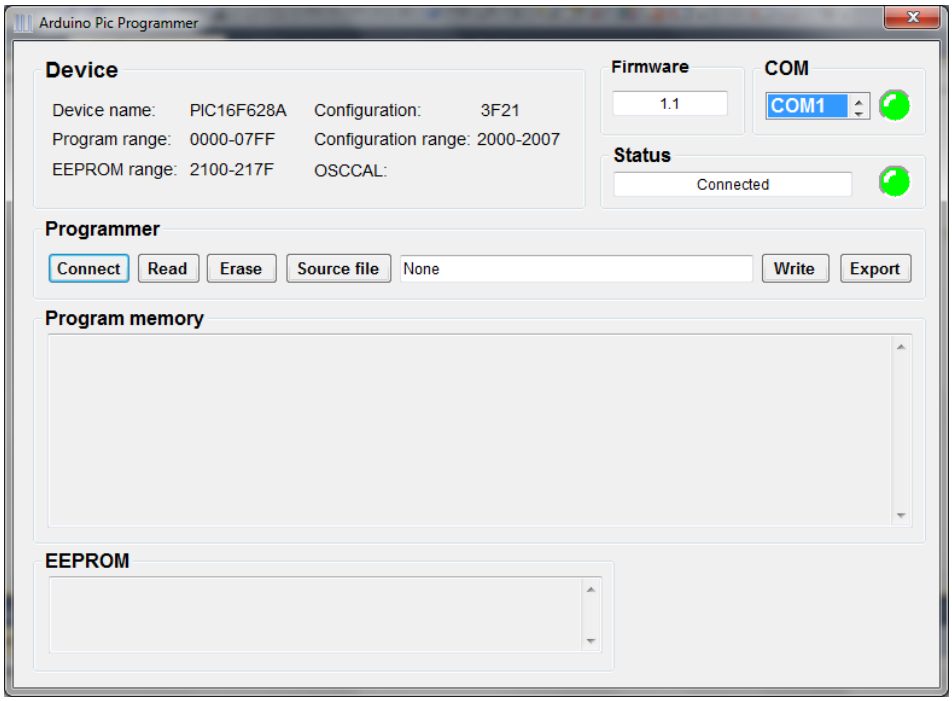
If your computer would offer COM-ports then the "Arduino Pic Programmer" would show the following screen. In this example COM1 would be available.



Click on the COM-Port and the application will attempt to connect to the programmer. Please make sure that your Arduino is loaded with the pro-

grammer sketch and connected to the COM port. Also, make sure that the little jumper on the ArdPicProg board is in the appropriate position to support the pic controller you want to program.

After establishing the communication with the ArdPicProg the following screen would be presented.



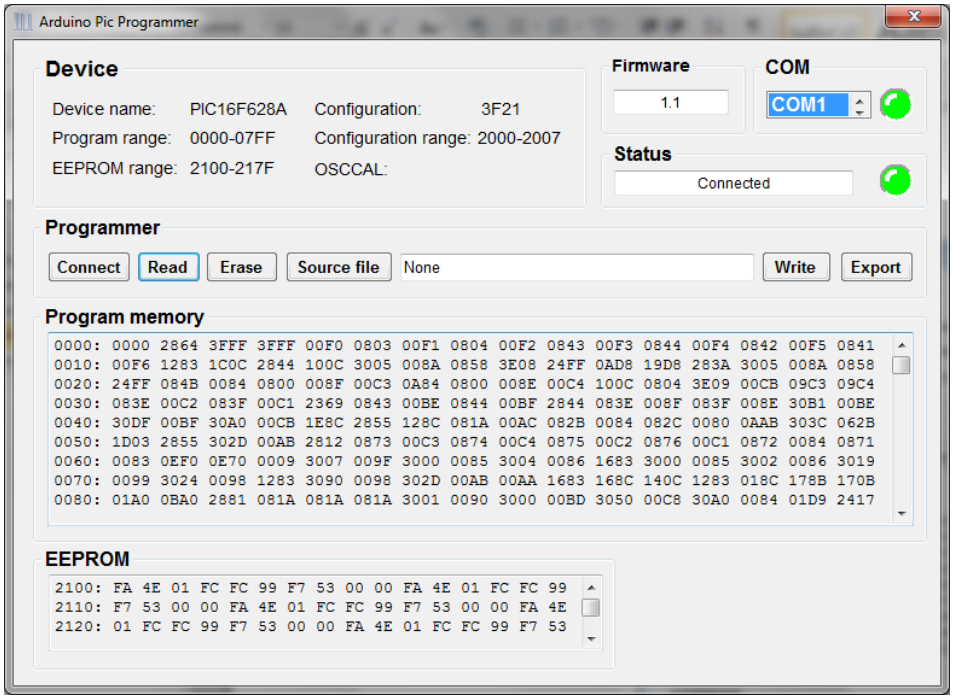
The top green led indicates the connection to the programmer and the firmware version is shown. The second status window indicates that a valid device has been found in the programming socket. The device data are shown is the respective window.

If no device was found then the device data field remains empty, the lower led is off and the status field is empty. Insert a device and hit the "Connect" button and the programmer will retrieve the device information then. Afterwards the window should appear similar to the image above.

If you wanted to read the pic's program and data then hit the "Read" button. The (physical) green led on the ArdPicProg will start to flash and the lower green led on the screen will turn red and start blinking. This indicates programmer activity and that it is currently not save to remove the controller from the socket. After completing the operation this led will turn back into green.

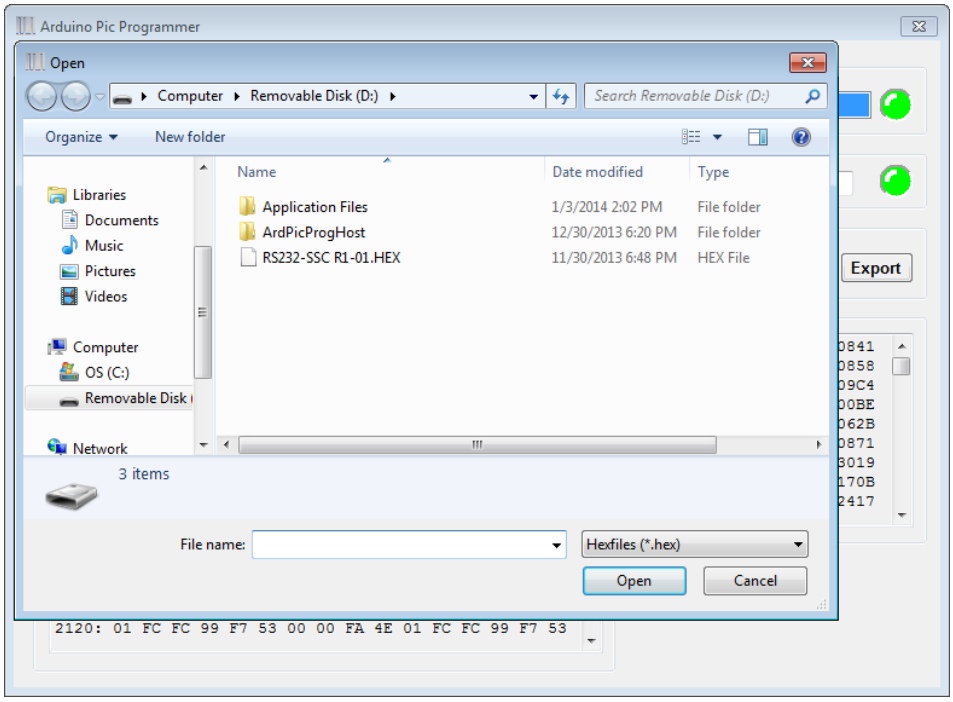
Also, the current reading address would be displayed in the status window. Please note that depending on the memory size the reading may take some time and that reading data would be only possible if the code protection of the controller has not been enabled.

After reading the controller the screen may look similar to the image on the following page.



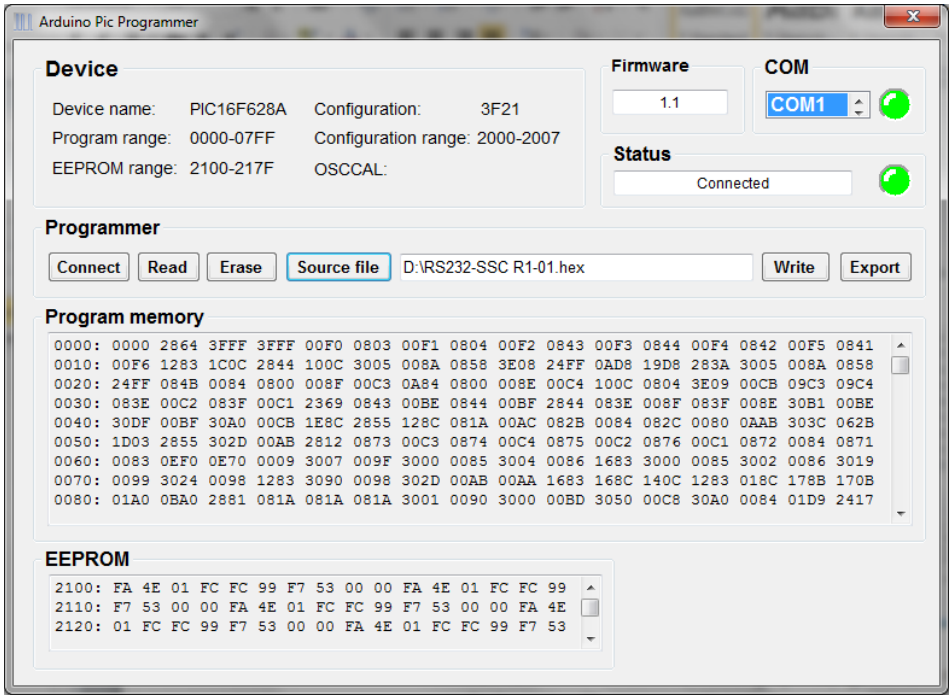
You could export the controller data in an Intel Hex file format by hitting the "Export" button.

For writing a controller you have to import the file with the firmware. Hit the "Source file" button and a file open window comes up.



Select the source file and the file would be imported. Please note that the source file has to be in Intel hex format as generated by the Microchip IDE. The source file name would be displayed in the text field next to the

"Source file" button. Now your programmer would be ready to program the controller.



Hit the "Write"-button. The (physical) green led on the ArdPicProg will start to flash again and the lower led on the screen will turn to red and blink also. Please note that the programming may take some time depending on the controller's memory size.

5

PicProgHost terminal application

Follow the steps described in the previous sections. Make sure that your Arduino is loaded with the programmer sketch and take a note of the COM port the Arduino is using. This information can be obtained through the Arduino IDE. For this example the COM port would be "COM4"

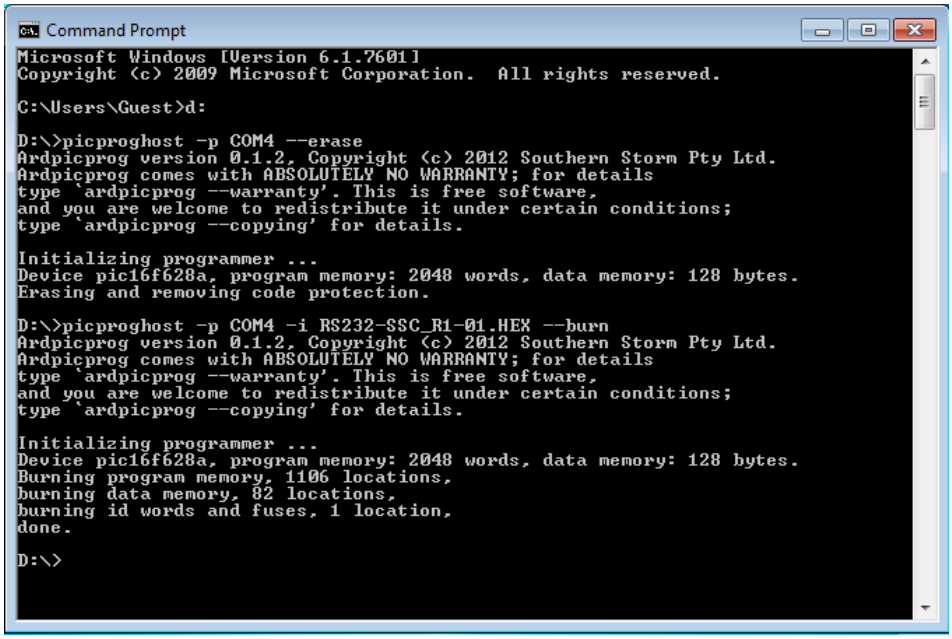
The host program is executed in Windows console mode. Please refer to the github-webpage for the program parameters. The easiest approach to programming a controller would be to copy the host application together with the hex-file (the file name would be "RS232-SSC_R1-01.HEX") for the pic controller on a USB stick (which would be drive "D" in this document) and start the application from there. This avoids dealing with extensive path names.

Make sure that the jumper on your programmer is located correctly for programming a PIC 16F628A (position for 18 pin controllers) and the red led should be on to indicate Vpp availability.

The host software operates in the "Command Prompt" window. This window can be found under "All Programs" > "Accessories" > "Command Prompt". After clicking the application a traditional terminal window with a command line will appear on the screen. Please refer to the picture on the following page.

Switch to the USB stick drive by entering "d:". Now you are able to invoke the host application right away. It is probably a good idea to erase the pic controller first. The command string would be: "picproghost –p COM4 –-erase".

Your next step would be to program the controller with the hex-file downloaded earlier by typing: "picproghost –p COM4 –i RS232-SSC_R1-01.HEX --burn". This step completes the controller programming.


Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\Guest>d:
D:\>picproghost -p COM4 --erase
Ardpicprog version 0.1.2, Copyright (c) 2012 Southern Storm Pty Ltd.
Ardpicprog comes with ABSOLUTELY NO WARRANTY; for details
type `ardpicprog --warranty'. This is free software,
and you are welcome to redistribute it under certain conditions;
type `ardpicprog --copying' for details.
Initializing programmer ...
Device pic16f628a, program memory: 2048 words, data memory: 128 bytes.
Erasing and removing code protection.
D:\>picproghost -p COM4 -i RS232-SSC_R1-01.HEX --burn
Ardpicprog version 0.1.2, Copyright (c) 2012 Southern Storm Pty Ltd.
Ardpicprog comes with ABSOLUTELY NO WARRANTY; for details
type `ardpicprog --warranty'. This is free software,
and you are welcome to redistribute it under certain conditions;
type `ardpicprog --copying' for details.
Initializing programmer ...
Device pic16f628a, program memory: 2048 words, data memory: 128 bytes.
Burning program memory, 1106 locations,
burning data memory, 82 locations,
burning id words and fuses, 1 location,
done.
D:\>

A

List of supported devices

The following devices are supported by the ArdPicProg:

- PIC12F629
- PIC12F675
- PIC16F630
- PIC16F676
- PIC16F84
- PIC16F84A
- PIC16F87
- PIC16F88
- PIC16F627
- PIC16F627A
- PIC16F628
- PIC16F628A
- PIC16F648A
- PIC16F684
- PIC16F690
- PIC16F882
- PIC16F883
- PIC16F884
- PIC16F886
- PIC16F887