More Description Search projects, profiles ... Co Quonents O Logs 16 Instructions O Discussion 8 **HACKADAY.IO Projects** Pavapro - portable AVR programmer Pavapro is tiny programmer you can bring anywhere. You can load binary file into it and bring/use it as you wish. And a bit more than that. jaromir.sukuba Follow project Like project Join this project 🏖 13.1k 1.3k 84 followers comments likes **DESCRIPTION** Pavapro (Portable AVr PROgrammer) is tiny programmer one can bring and use anywhere. You can load binary file into it, get it at place of business and flash the binary into the AVR. That can be useful for resurrecting bricked AVR project somewhere in middle of nowhere or loading new firmware revision into remote AVR where computer would be annoying or impossible to bring. Just connect AVR, select your binary and here you go. This would be quite boring, so let's add something more. You can also write, assemble and load your own program into AVR, without using your PC or laptop. Yes, something like tiny micropower AVR IDE running on another AVR (trinket pro). I assume there will be some FLASH left unused, so crazy late night programming few hours before deadline will show what other functions (don't know, serial terminal maybe?) will be implemented. **DETAILS** Pavapro is relatively small, pocket-able device, allowing you to operate above microSD card, reading binary files (for now only .bin files are supported) and load those files into target AVR View Gallery device. In addition to this, fuses files (ASCII text files) can be loaded too and fuses written into AVR. Apart from that, pavapro allows you to open/edit/save text files, as well as there is also simple file viewer, when you don't want to mess with editor. As if it wasn't enough, simple ASCII serial terminal is included too, allowing to view incoming data on CMOS or RS232 levels. ● 13.1k ● 8 **1** 1.3k **3** 84 **TEAM (1)** 🎲 jaromir.sukuba Join this project's team project at GitHub Original resolution photos, including those ones **\* HARDWARE O COMPLETED PROJECT** Its dimensions are 118x71x19mm, runs on single Li-Ion or Li-Pol battery, consuming approximately **ASSEMBLER** PROGRAMMER PORTABLE 20mA when running - so, classic 600mAh type will allow to run approximately for 30 hours on TRINKET PRO **EDITOR** single charge. It can be charged from USB micro connector, serving as well for firmware update. The user IO is done via 16-keys keypad and 128x64 OLED display. The electrical IO is 6-pin AVR ISP header (alternatively it can serve as 5xADC input or 4xIO line) and USART on CMOS levels as well as RS232 levels, plus one universal CMOS IO on second 6-pin connector. Total price of components was around 20EUR + PCB price (could be anything, beginning from **RELATED LISTS** OEUR if you have good fellow hacker, thank you David!) **Trinket EDC Contest Winners** Atmel® AVR® Projects The develoment of pavapro was done on free (as freedom) or free (as beer for free) tools under AVR" Linux and all design source files are freely available on github - so anyone can take the project, build, alter and contribute. My main development machine runs on Xubuntu, software development was done using **Atmel Parts** Gedit/GCC/Arduino IDE, hardware development - Eagle for PCB design (in fact, the only exception from open-source tools) FreeCAD for 3D modelling and Slic3r/Pronterface for 3D printing. Though I'm running on FOSS tools, I understand that a lot of people runs windows OS, so I took care to select tools that have windows conterparts. THIS PROJECT IS SUBMITTED FOR The Trinket Everyday Carry Contest Myself, while working on pavapro. Want to build one? Here is how. Here is video of pavapro, ressurecting dead arduino pavapro - recovering bricked arduino And another video, showing other features of pavapro PROJECT LOGS ^ Pavapro 2 is alive It's been quiet here for a few months, as I had just a little time for hobbies, though the project was moving on, slowly, look: I made an expansion board for STM32L152RE Nucleo board. And it is stacked in usual boring way But it is alive! I'm thinking of changing display type. That OLED is nice, but its tiny compared to keyboard and has high power consumption. If I'd change display to something bigger, the OLED screen would take much more juice. Seems like little TFT displays could use less power - like the usual 160x128 pixels I used in #JJ Tricoder and #10\$ curve tracer projects. It seems to be happy with a few miliamps of power for background LEDs, having the biggest share in total display consumption. I considered also EADOGM128 monochromatic display, but the price seems to be quite high. The 160x128 TFT is on ebay for 3USD or thereabouts. What's next? Though Pavapro in its current form will not be not much developed, I strongly think of Pavapro 2. I looked into documentation and searched for FLASH programming of various MCU's. What I found it would be doable for: PIC16F1xxx PIC18F (including PIC18FxxKxx and PIC18FxxJxx) PIC24 PIC32 MSP430 AVR (it is currently done for AVRs with equal or less than 64kB) with unlimited FLASH and ATtiny10 and friends • AT89 - x51 MCUs In fact, the programmer part of pavapro wasn't the hardest one. The task would be further simplified by borrowing code from my previous projects: the 8-bit PIC from #1757. PP04 - Camel computer and AT89 from PIC89PROG I'm a bit reluctant about JTAG. I feel like it's not going to be rocket science, but anyway, implementing JTAG would allow interesting games, like • svf player for programming programmable logic various ARM support As a result of my love affair of 68HC08 MCUs (before Freescale obsoleted them in short time lesson learned: never use any MCU from Freescale again) I'm probably going to implement them too. So, HC08 More device support is possible, but not in first plan for now STM8 serial memories Above that all, Implementing not only programming from SD card, but also receiving the programming file from serial line (or USB) would allow to use pavapro2 as quite universal device programmer. Just send hex file into serial port from whatever with serial port (your PC, raspberry, phone, type it into ASR33...) and wait a moment. What I would definitely change is MCU in pavapro2. Honestly, AVRs are not that convenient to program as I'm used to (PIC, ARM). No debug support in Linux (AtmelStudio based on windowsonly Visual Studio, what a cruel joke in year 2015) and various little, but painful nuisances like "pgmspace.h" or annoying fuses (No RSTDISBL, no fun!), sometimes rendering your MCU into OTP part and your device into brick. In fact, not much of horsepower is required here - ATmega328 was sufficient, in this field. However lack of RAM and FLASH was showstopper for my development. 32k of FLASH was too small, 64kB would be better and 128kB or more is king size. 2kB of RAM was compromise, some 8kB would be nice, but I feel like 32kB or more would be great. I'm not sure whether I'm going to use PIC32 (my favourite PIC32MX795F512H or even new PIC32MZ with half a meg of RAM) or ARM MCU (venerable LPC1769, maybe something newer, like LPC11U68). Anyway, I want to keep the hardware not more complicated than it is now. I don't want to grow it into huge behemoth. I'm going to let this idea to mature and see how it turns out. Result of Trinket EDC Contest This project won second prize in Trinket EDC contest. Great! Thanks to all who followed/skulled this project, this feedback was valuable and helped me to put more effort in the project. View all 16 project logs **ENJOY THIS PROJECT? DISCUSSIONS** Log In or become a member to leave your comment Log In/Sign up to comment That One Guy wrote 03/28/2017 at 15:17 can you program a arduino w/ this? not sure what you mean by "recovering bricked arduino" helge wrote 09/13/2015 at 22:54 Awesome! This is something I wanted to build so badly when smartphones came up. I imagined myself sitting in a train or public place programming away and giving people that "yeah can your phone do \*this\*?"-look. You rock, man. jaromir.sukuba wrote 09/14/2015 at 05:52 Oh, thank you! I love to hear somebody has so similar thoughts. And now shameless plug: You may want take look at #PP04 - Camel computer Jasmine Brackett wrote 01/07/2015 at 22:52 Congratulations Jaromir on getting second place in the Trinket Everyday Carry Contest! We'll be sending out a Fluke 179 with a 6 piece industrial electronics tip kit soon. jaromir.sukuba wrote 01/0//2015 at 23:06 Jasmine, thank you, and all the HaD team for this great contest (and silently hope for more contests like this)! The second prize is fantastic credit for my work. davedarko wrote 01/07/2015 at 23:52 Congratulations! This was in my personal top three (like the first and third) - I'm happy for you and the other contestants, but also for being right this time - not like on

quite inclined to make pavapro mk2. **SIMILAR PROJECTS** 8051 tuner ATtiny10 resources MCU how-tos, reviews, rants shlonkin Ken Yap 훩 jaromir.sukuba

Your email address

Password

Does this project spark your interest? Become a member to follow this project and never miss any updates Become a member

o Worldill

● 3.6k **1** 10 **2** 9

TheHackadayPrice. Great work!

Wow, Jaromir... fantastic work! It's been a while since I checked in on this one and I'm shocked

I like the fact that there's a lot of supported chips. And the published code and well-

I tried to cover as much of AVR chips as I could by simple means. Having other

architectures would be quite challenging, but I love challenges, after all. The concept

wouldn't have to change that much, but more beefy MCU would be needed. Honestly, I'm

documented user interface would make it pretty easy for people to add other

Mike Szczys wrote 01/05/2015 at 21:56

at how polished the finished build is.

chips/architectures for their own uses.

Mike, thank you for your compliment.