

AVRDUDE(FormerlyAVRPROG)

AVRDownloader/UploaDEr

- ThelatestversionofAVRDUDEis4.2.0,whichisavailablefor [downloadhere:](#)
[avrdude-4.2.0.tar.gz](#)
- Viewthe [NEWS](#)fileforasummaryofchangessincethelastversion.
- Forthelatestprojectinformation,seetheAVRDUDEprojec [tat savannah.nongnu.org.](#)

NOTE:AVRDUDEistheFreeBSDUnixsoftwareformerlyknown asAVRPROG.Thenamewas changedsoasnottocounteractwithAtmel'sownprogramm ersoftwarecalledAVRPROG.EXE.Currently, aneffortisunderwaytoportAVRDUDEtoLinuxandWindow s.ThenameAVRPROGwouldsurely causeconfusionwithAtmel'ssoftware,thusIfeltthe bestwaytohandlethisistojustchangethename.

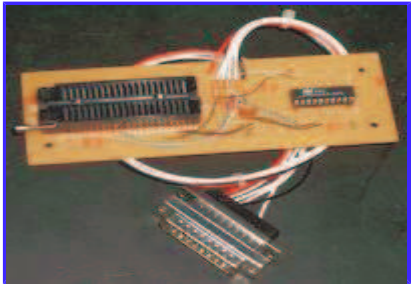
AVRDUDEisafullfeaturedFreeBSDUnixprogramforprogra mmingAtmel'sAVRCPU's.Itcan programtheFlashandEEPROM,andwheresupportedbythese rialprogrammingprotocol,itcan programfuseandlockbits.AVRDUDEalsosuppliesadirecti nstructionmodeallowingonetoinstallany programminginstructiontotheAVRchipregardlessowhet herAVRDUDEimplements thatspecific featureofaparticularchip.

AVRDUDEcanbeusedeffectivelyviathecommandlinet oreadorwriteallchipmemorytypes(eeprom, flash,fusebits,lockbits,signaturebytes)orviaan interactive(terminal)mode.UsingAVRDUDEfrom thecommandlineworkswellforprogrammingtheentirem emoryofthechipfromthecontentsofafile, whileinteractivemodeisusefulforexploringmemoryco ntents,modifyingindividualbytesofeeprom, programmingfuse/lockbits,etc.

AVRDUDEsupportstwobasicprogrammertypes:Atmel'sSTK500 andthePPI(parallelportinterface). PPIrepresentsaclassofsimpleprogrammerswherethe programminglinesaredirectlyconnectedtothe PCparallelport,whiletheSTK500usestheserialportto communicatewiththePCandcontains on-boardlogictocontroltheprogrammingofthetargetde vice.Severalpinconfigurationsexistforthe variousPPIprogrammersthatexist,andAVRDUDEcanbe beconfiguredtoworkwiththembyeither specifyingtheappropriateprogrammeronthecommandlineo rbycreatinganewentryinits configurationfile.Allthat'susuallyrequiredforanew entryistotellAVRDUDEwhichpinstousefor eachprogrammingfunction.

Thesimplestpossibleprogrammerisshownbelowandrequir esonlyafewconnectionsandonlyone additionalpart(crystalresonator).Init'ssimplestf orm,justmakethefollowingconnectionsfromthe parallelportpinstotheAtmelAVRdevice:

ParallelPort	ProgrammerFunction
Pins2-5	Vcc(seenotebelow)
Pin7	AVR/RESET
Pin8	AVRSCK(clockinput)
Pin9	AVRMOSI(instructionin



Pin10	AVRMISO(dataout)
Pin18	SignalGround

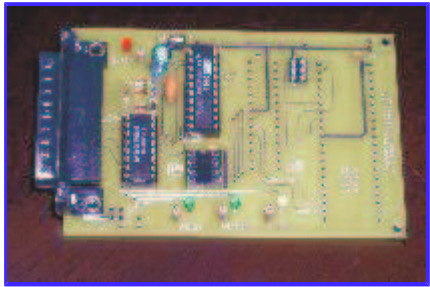


The above photo is the board I put together. Quick and dirty, but functional. Don't forget to hook up a crystal or resonator to the AVR chip's supply clock from some other source to XTAL1.

NOTE regarding Vcc connection: make sure your parallel port can supply an adequate amount of current to power your device. 6-10mA per pin is common for parallel port signal lines, but is not guaranteed, especially for notebook computers. For additional power, use multiple pins tied together with Schottky diodes. If in doubt, don't risk damaging your parallel port, use an external power supply.

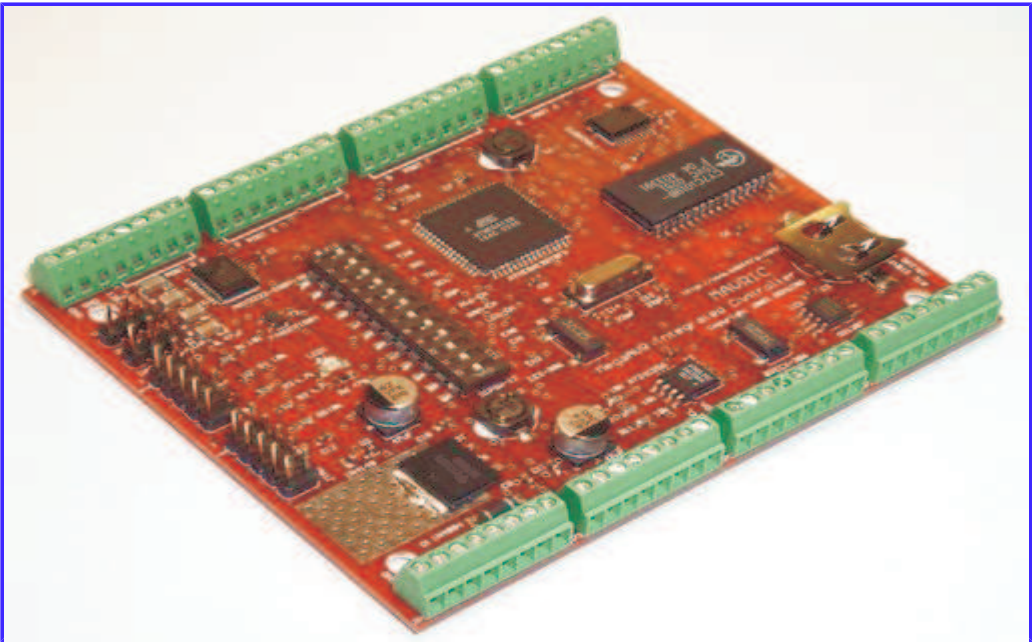
Additionally, the following pins may be connected which provide the ability to buffer the parallel port from the AVR device (allowing for in-circuit programming), and for status LEDs. All LEDs should be wired active low.

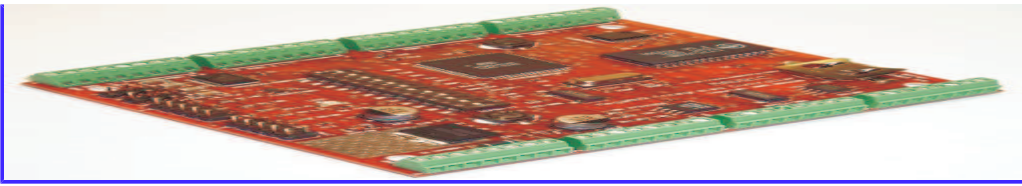
ParallelPort	ProgrammerFunction
Pins1	STATUSLED(program or verify error)
Pin6	/ENABLE(74367 bus driver)
Pin14	STATUSLED(ready)
Pin16	STATUSLED(programming)
Pin17	STATUSLED(verifying)



The above photo is of a board that Tony Friebe made to use with AVRDUDE. This board utilizes all the features of AVRDUDE including buffered signals using a 74367 as well as LED indicators. Here is a [schematic](#) drawn by Joerg Wunsch which shows how to hook everything up.

Need a microcontroller board?





If so, you might be interested in my [MAVRIC](#) board, an Atmel ATmega128 based microcontroller that I make. I like to have a few of these on hand to pull out of the parts cabinet when I need to put together a control solution. You can buy it as a bare PCB, a Kit, or assembled and tested. It's simple to put together, has lots of features, and is very flexible.