Radare2 commands

Command line

`aaa` ..... autoanalyze all
`pd @ <address>` .....disassemble at address
`s <address>` ..... seek to address
`ii` ..... file information
`iz` .....strings in data section
`iZ` ..... all strings
`is` ..... symbols
`S` ..... section manipulation
`Ps <file>` ..... save project
`Po <file>` ..... open project
`Pl` ..... list projects
`Pi` .....project information
`/` ..... search
`/i` ..... search string (ignoring case)
`/c <instr>` ..... search for instruction
`/R <instr>` ..... search for ROP gadget
`~ or | grep` ..... filter results of command (grep)
`?` ..... help
`? <expr` ..... inline calculator (e.g. ? 0xa/2)
`?d <cmd>` ..... command descriptions (e.g. ?d mov)
`. <file>` ..... load commands (e.g. zignatures) file
`.z/ <beginaddr> <endaddr>` . search for zignature functions in file
`fs sign` ..... look into flagspaces (e.g. for zignatures)
`f` ..... display all flagspaces (e.g. show all zignature hits)
`axt` ..... display XREFs to current address
`af` .....analyze function
`afn` ..... rename function
`e asm.describe=true` ..... turn on autocommand descriptions
`e asm.emu=true` ..... turn on asm "emulation"
`e rop.comments=true` ..... turn on metadata in ROP search
`e asm.midflags=false`..turn off(true=on) forced alignment in code
`q` ..... quit

Visual mode

`V` ..... (from shell) Enter visual mode
`p/P` ..... rotate visual modes
`c` ..... toggle cursor
`q` ..... back to shell
`<Enter>` ..... Follow jump/call address
`d[f?]` ..... Define function, data, code
`o` ..... Jump to offset
`g/G` ..... go to beginning/end of file
`V` ..... (from visual) View graph
`u/U` ..... Undo/Redo seek
`:` .....run r2 command without quit to shell
`x` ..... show/follow XREFs
`e` ..... visual editor of r2 variables
`;<comment>` ..... add comment
`;-` .....remove comment
`T` .....browse analyzer info and comments
`v` ..... visual code analysis menu
`C` .....toggle colors
`R` ..... Randomize color palette

Command-line arguments

`r2 -a avr <firmware_file>` ..... open file
`r2 -a avr -c=H <firmware_file>`..... open file in Web GUI

Other tools

AVR- GDB

`target remote :<port>` ..... Connect to gdbserver @ <port>
`run` ..... run firmware
`continue` ..... continue execution
`ni <num>` ..... next instruction(s)
`b* $pc + <offset>` ..... set breakpoint at address (PC + offset)
`d` ..... delete specified or all breakpoints
`x/<num>i` ..... display <num> instruction(s) (x/20i -- show 20 instructions)
`x/<num>s <addr>` ..... display strings at <addr>
`x/<num>x <addr>` ..... display hex values at <addr>
`i r` ..... registers info

JTAGenum

Working pins: D2-D11
Connect: `screen /dev/ttyACM0 115200`
Search command: **s**

AVR Tools

`avr-objdump -D -b binary -m avr <file>` ..... plain disasm from GNU avr objdump
`avr-objcopy -I ihex -O binary blink.hex blink.bin` ..... convert between firmware formats
`avarice --mkI --jtag /dev/ttyUSB0 -p -e --file hello.hex` ..... flash hello.hex firmware through JTAG connected to port ttyUSB0
`avarice --mkI --jtag /dev/ttyUSB0 -d :4242` .. debug firmware through JTAG connected to port ttyUSB0 and with gdbserver on port 4242
`avarice --mkI --jtag /dev/ttyUSB0 -r –l` ..... read FUSEs and lockbits