# Microcontroller programming

AVR Studio Simulator Stimuli

# Stimuli

- Provide very basic external input
- “Timed” IO register setting
- Logging

- Evaluated *between* instruction execution
  ⇨ Delay may be a bit longer due to multi-cycle instructions

- Debug / Set Stimulifile (*.stim)
  or
  Project Properties / Tool / Select Stimuli File for Simulator
- Debug / Execute Stimulifile
- Output goes to Output / FileStimuliProvider
  - Timestamped (time = clock cycles)
  - Appears only when started
  - Retains previous output, needs to be explicitly cleared

# Very basic features

- Delay
- Assignment
- Directive
- Comment

# Delay

#<num>

- Clock cycles
- Commands not separated by delay are executed simultaneously

# Assignment

target op value

- Target: any IO register (address or tinyAVR and megaAVR register name)
- Operators: `=` `|=` `&=` `^=` (must be surrounded by space)
- Value: numerical constant, `*`register

*Fun fact: For devices with complex I/O structure (XMEGA®, UC3, SAM) it is for now recommended to use addresses. The easiest way to determine the address is to bring up the I/O view and select the desired register. The address can be copied from the I/O view (select the desired register, right-click, and select "Copy Address").*

# Directives

| | |
|---|---|
| **$stimulate** <file> | include that file |
| **$break** | break debugging |
| **$repeat** <num> … **$endrep** | loop num times |
| **$log** <register> [<mask>] | log reg if changed |
| **$unlog** <register> [<mask>] | stop logging reg |
| **$startlog** <file> [**a**\|**o**] | start dumping to file |
| **$stoplog** | stop dumping to file |
| **$fuse** <address> <value> | set fuse byte |
| **$reset p\|e\|b\|s** | reset device |
| **$memload** <file> **s**\|**f**\|**e**\|**i** [nocheck] | load memory from file |
| **$memdump** <file> <address> <size> [**s**\|**f**\|**e**\|**i**] | save memory to file |

# Comment

```
// single-line
```

- (block comments are not supported)

# Example

```
// loop 5 times
$repeat 5
   #45
   PINA = *PORTB
   #5
   PINA &= 0b11100111
$endrep
```

# Issues

- The mapping between register names and addresses only work reliably for devices with flat I/O structure with unique register names. Dotted notation does not work, use numeric addresses instead.
- Logging of some I/O registers on 32-bit devices may be unsupported. This will be documented on a per-device basis.
- In assignments, the operator (=, etc) must be surrounded by spaces.
- The stimuli interpreter will fail if the last line of the stimuli input file is not terminated by a newline.
- On 8-bit devices, it is not possible to assign values to 16- or 32-bit register tuples, e.g., to assign to ADC one must assign to ADCL and ADCH separately. See example in Example Stimuli Session
- Error reporting leaves a lot to be desired.
- The timing of stimuli can be a cycle or two off compared to delay specification because stimuli files are evaluated only between CPU single-steps in the current implementation.
- Sharing violation if attempting to edit a stimuli file while open.

- Cannot be aborted (only by aborting whole debug session)
- No absolute timing