

The avremu Package

Christian Dietrich

stettberger@dokucode.de

<https://gitlab.brokenpipe.de/stettberger/avremu>

CTAN: <http://www.ctan.org/pkg/avremu>

Version v0.1 – 2014/10/09

```
#include <avr/io.h>

int
main(void)
{
    char *str = "Hello_World!";
    char *p = str;
    while (*p) {
        UDR = *p++;
    }
    asm volatile ("break;");
}

\avrloadc{hello-world.c}
\avrrun
UDR='avrUDR' in \avrinstcount\ instructions
```

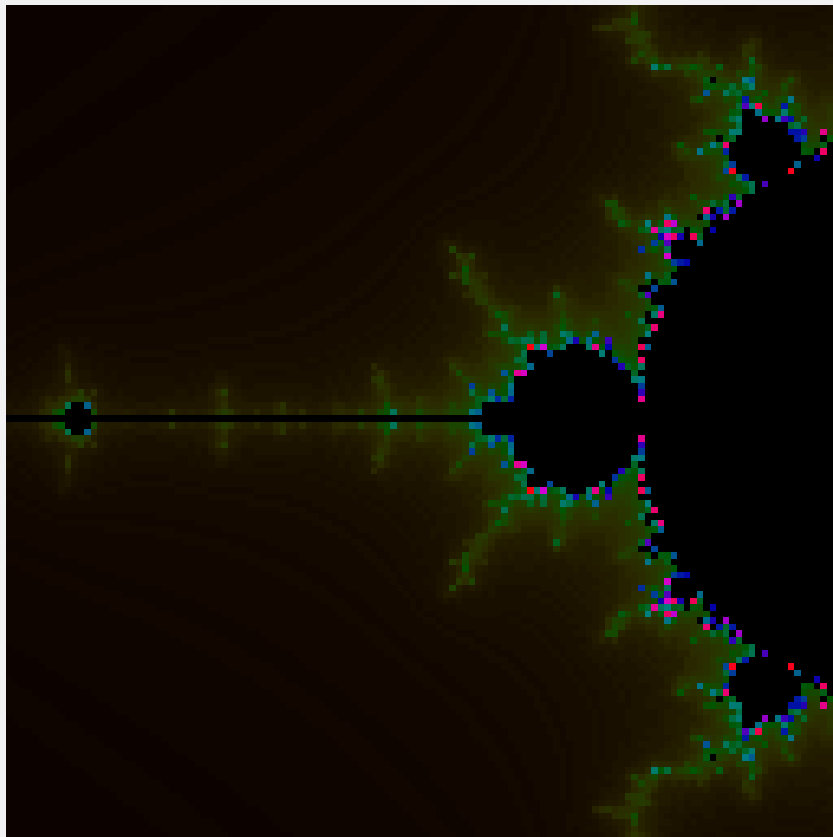
```
UDR='Hello World!' in 153 instructions
```

⌘TeX is known as a typesetting system. But the underlying TeX system is a powerful macro processor. In fact, TeX is a Turing-complete programming language. TeX can compute anything that is computable. Computability is a concept from theoretical computer science. After visiting a theoretical computer-science course, you will know that there are things that cannot be solved by a machine. Never. Look out for the halting problem.

This package does contain an *CPU emulator* for the 8-bit microcontroller platform Atmel AVR, more precisely it implements the instruction-set architecture of the ATmega8.

```
\avrloadc{mandelbrot.c}  
\avrrun  
  
\avrdrawppm{mandelbrot.ppm}  
\immediate\write18{convert mandelbrot.ppm mandelbrot.png}  
  
\includegraphics[width=\linewidth]{mandelbrot.png}
```

This picture (128x128) took 6 hours to render. The source code can be found in the test-suite directory under mandelbrot.c.



1 Provided Commands

`\avrloadihex{filename}`

Load an Intel HEX formatted image of the flash into the code memory of the AVR emulator. Additionally the state of the AVR emulator is set back to zero.

`\avrloadc [compiler options] {filename}`

Requires `--shell-escape`. Compiles C source code file with `avr-gcc` and the given compiler options. The default compiler option set is `-mmcu=atmega8 -Os`. The resulting `.elf` file is transformed to an Intel HEX file and loaded into the code memory of the emulator.

`\avrrun`

Run the emulator until a **break** instruction occurs.

`\avrstep [steps=1]`

Run the emulator for N instructions. The default is a single step. The stepping does automatically end, if a **break** instruction is executed.

`\avrinstrcount`

Expands to the number of executed instructions.

`\avrsinglestep`

Starts an interactive single-stepping mode, which was mainly used for implementing the emulator.

`\usravremulibrary{list of libraryies}`

1.1 Access to the Serial Console

If the program write to the UDR IO register, the emulator caught those characters in an internal buffer.

`\avrUDR`

Expands to the internal UDR buffer.

`\avrUDRclear`

Clears the internal UDR buffer.

1.2 Draw Library

`\useavremulibrary{avr.draw}`

See `source/test-suite/mandelbrot.c` for more details.

2 Implementation Details

Read the source.