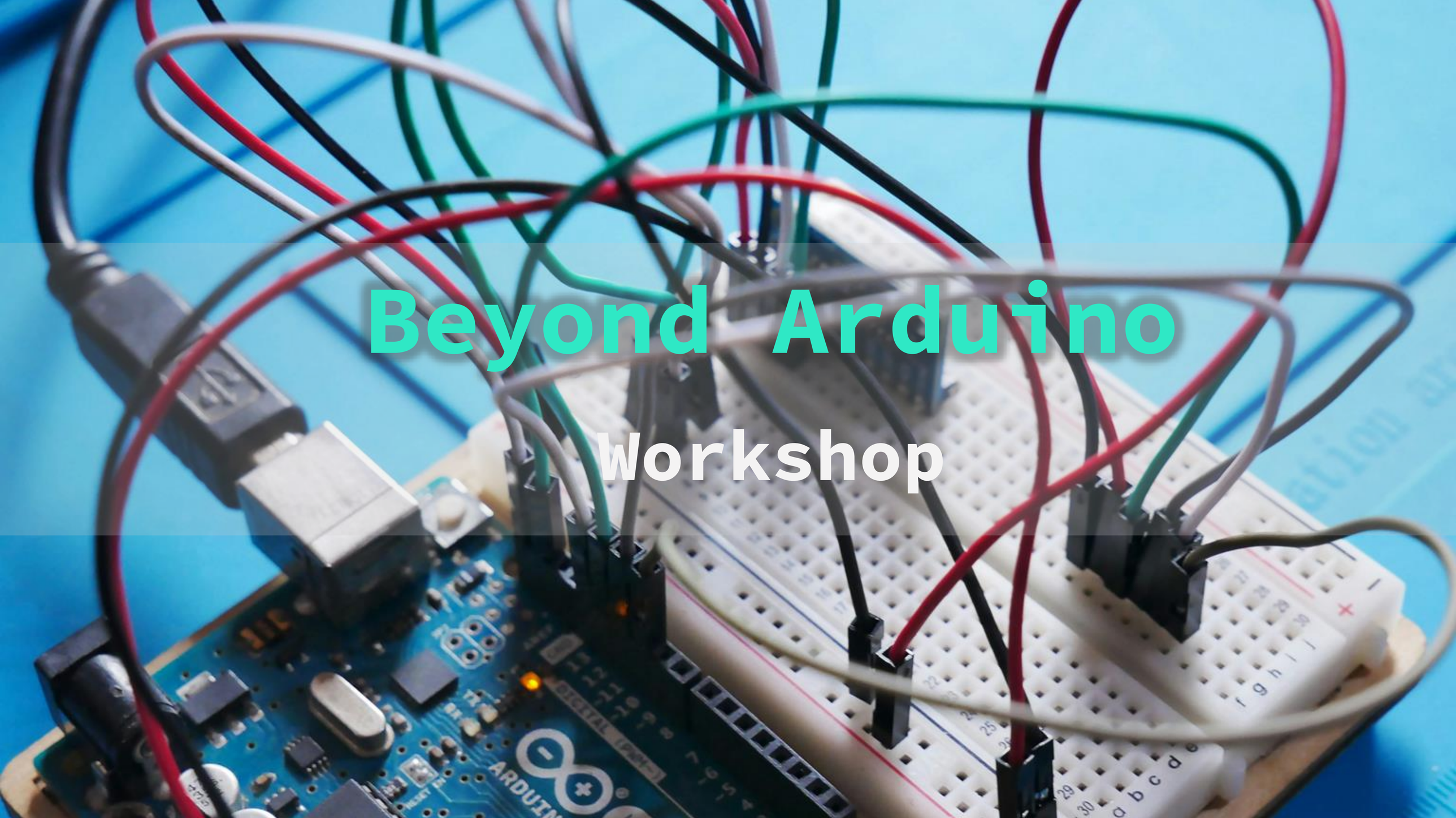


Beyond Arduino

Workshop

Beyond Arduino #3

01 Build process in C/C++

02 Uploading code to atmega328p

03 Hello, Makefiles!

Build Process in C/C++

How is the code flashed into your MCU built?

- **Pre-processing**: A **pre-processor** first takes your C file, expand all the includes and macros, resolve your conditional compilation directives, and finally passes the new C file to the compiler.

- **Compiling**: A **compiler** takes your C file and compile it into assembly code that is specific to the platform you’re working on.

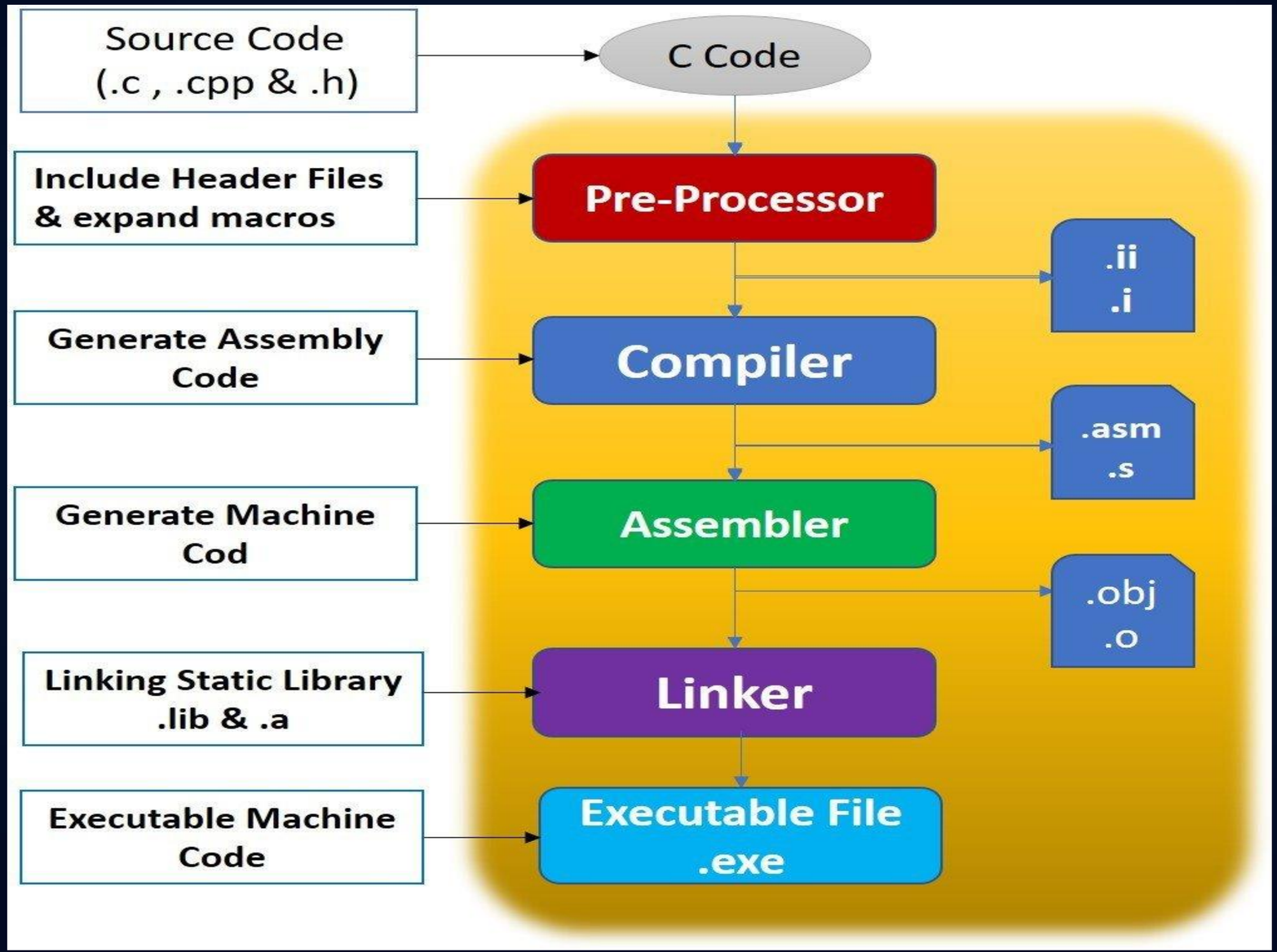
Build Process in C/C++

How is the code flashed into your MCU built?

- **Assembling**: An **assembler** takes that assembly code and turns it into object code which is almost pure binary (some symbols remain undefined until the linking phase).

- **Linking**: A **linker** takes object files (if multiple C files exist for a single project) and turns them into one executable.

Build Process in C/C++



Build Process in C/C++

How Arduino IDE does its job?

- The arduino IDE hides all compilation and uploading processes from you, but you can see them by choosing ‘verbose’ option in preferences.

- Arduino IDE uses the avr-gcc compiler to compile code and avr-dude to flash code into the MCU.

- By installing AVR compiler into your system, you install the compiler, the uploader, the binutils and the avr libc.

Build Process in C/C++

How Arduino IDE does its job?

- In order to compile a file into object format and link it later with other objects (libraries), we use this command:

```
avr-gcc -mmcu=atmega328p -c file.c -o file.o
```

avr-gcc the name of the compiler
-mmcu <Target MCU>
-c to compile and don’t link
-o to name the output as <name.o>

Build Process in C/C++

How Arduino IDE does its job?

- In order to make a binary file out of the object files:

```
avr-gcc -mmcu=atmega328p <LIST OBJECTS> -o file-name.bin
```

avr-gcc the name of the compiler
-o to name the output as <file.bin> (bin for binary)

Build Process in C/C++

How Arduino IDE does its job?

- AVR MCU expects an executable format called Intel Hex format:

```
avr-objcopy -O ihex -R .eeprom file.bin file.hex
```

avr-objcopy	copies binary file into hex file
-O	output
ihex	intel hex format
-R .eeprom	Remove eeprom section

Uploading Code into atmega328p

How Arduino IDE does its job?

- Avr-dude, the Downloader/UploaDEr (which Arduino IDE uses), can be used to flash our code into our MCU.

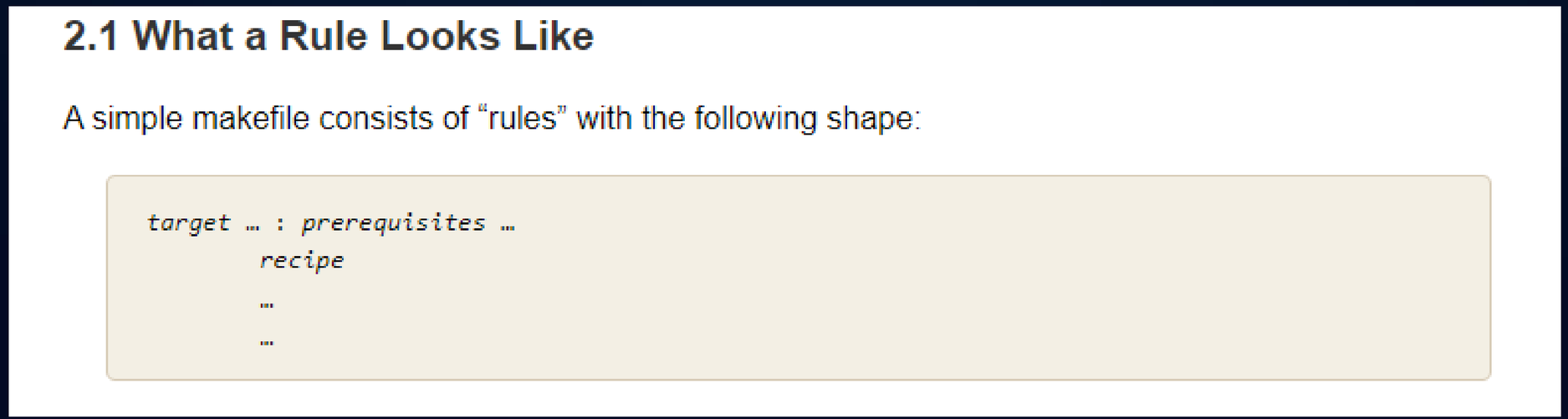
```
avrdude -F -V -c arduino -p ATMEGA328P -P <port> -b 115200 -U flash:w:file.hex
```

-c <PROGRAMMER-NAME>
-p <MCU-NAME>
-P <PORT>
-b <BAUD RATE>
-U what memory operation? -> a write (w) on flash memory (flash) the file (app.hex)

Hello, Makefiles!

Let’s build stuff with make!

- **Make is a build utility program. Its input is called a Makefile which describes recipes to build stuff. Makefiles are good at describing dependencies by knowing which program modules are dependent on which files, and examining the modification times of the various files.**



2.1 What a Rule Looks Like

A simple makefile consists of “rules” with the following shape:

```
target … : prerequisites …
        recipe
        …
        …
```

[GNU Make docs]

Beyond Arduino Workshop

Wameedh Scientific Club