

INSTITUTE
OF COMMUNICATION,
INFORMATION
AND PERCEPTION
TECHNOLOGIES



Scuola Superiore
Sant'Anna



ARTE: Arduino Real-Time Extension for Programming Multitasking Applications

Pasquale Buonocunto, Alessandro Biondi, Marco Pagani,
Mauro Marinoni, Giorgio Buttazzo

Scuola Superiore Sant'Anna, Pisa

Arduino Framework

“Arduino is a tool for making computers that can sense and control more of the physical world than your desktop computer.”

□ Very popular

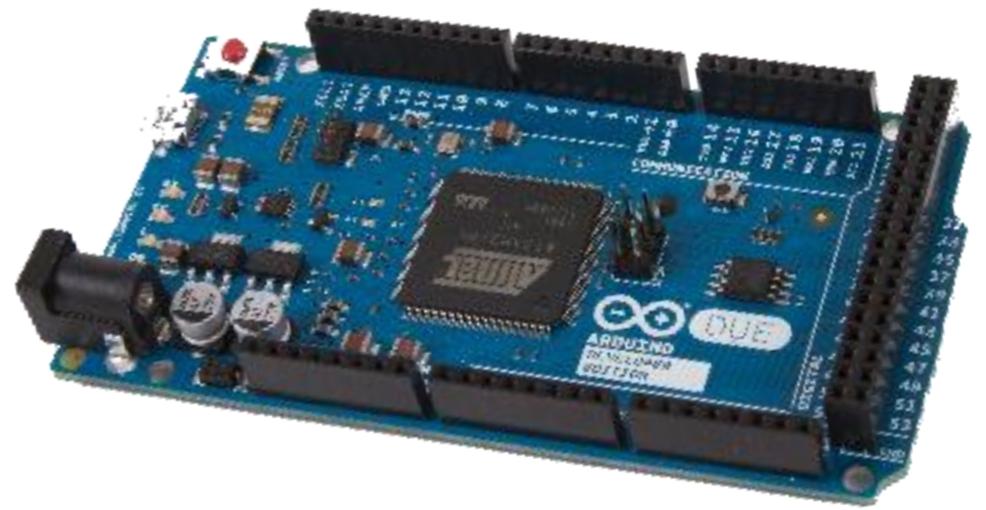
- 2013: 700,000+ official boards were in users' hands
(without counting clones..)

□ Very simple!

- Simple programming interface;
- Easy firmware loading.

□ Low cost

~20\$ official



Arduino Framework

```
void setup() {  
    <instructions here>  
}  
  
void loop() {  
    <instructions here>  
    delay(1000);  
}
```

One-shot execution
at startup

Cyclically executed
“*until power off*”!

Arduino Framework

A lot of libraries...

Standard Libraries

- [EEPROM](#)
- [Ethernet](#)
- [GSM](#)
- [LiquidCrystal](#)
- [SD](#)
- [Servo](#)
- [SPI](#)
- [SoftwareSerial](#)
- [Stepper](#)
- [TFT](#)
- [WiFi](#)
- [Wire](#)

Contributed Libraries

- [Messenger](#)
- [NewSoftSerial](#)
- [OneWire](#)
- [PS2Keyboard](#)
- [Simple Message System](#)
- [SSerial2Mobile](#)
- [Webduino](#)
- [X10](#)
- [XBee](#)
- [SerialControl](#)
- [Capacitive Sensing](#)
- [Debounce](#)
- [GLCD](#)

- [Improved LCD](#)
- [LedControl](#)
- [LedDisplay](#)
- [Matrix](#)
- [PCD8544](#)
- [Sprite](#)
- [ST7735](#)
- [FFT](#)
- [Tone](#)
- [TLC5940](#)
- [DateTime](#)
- [Metro](#)
- [MsTimer2](#)
- [PString](#)
- [Streaming](#)

Limitation of the Arduino Framework

- **No** support for concurrency;
- **No** execution period can be expressed;
- **No** precise timing can be enforced;
- Execution limited to **a single loop**.

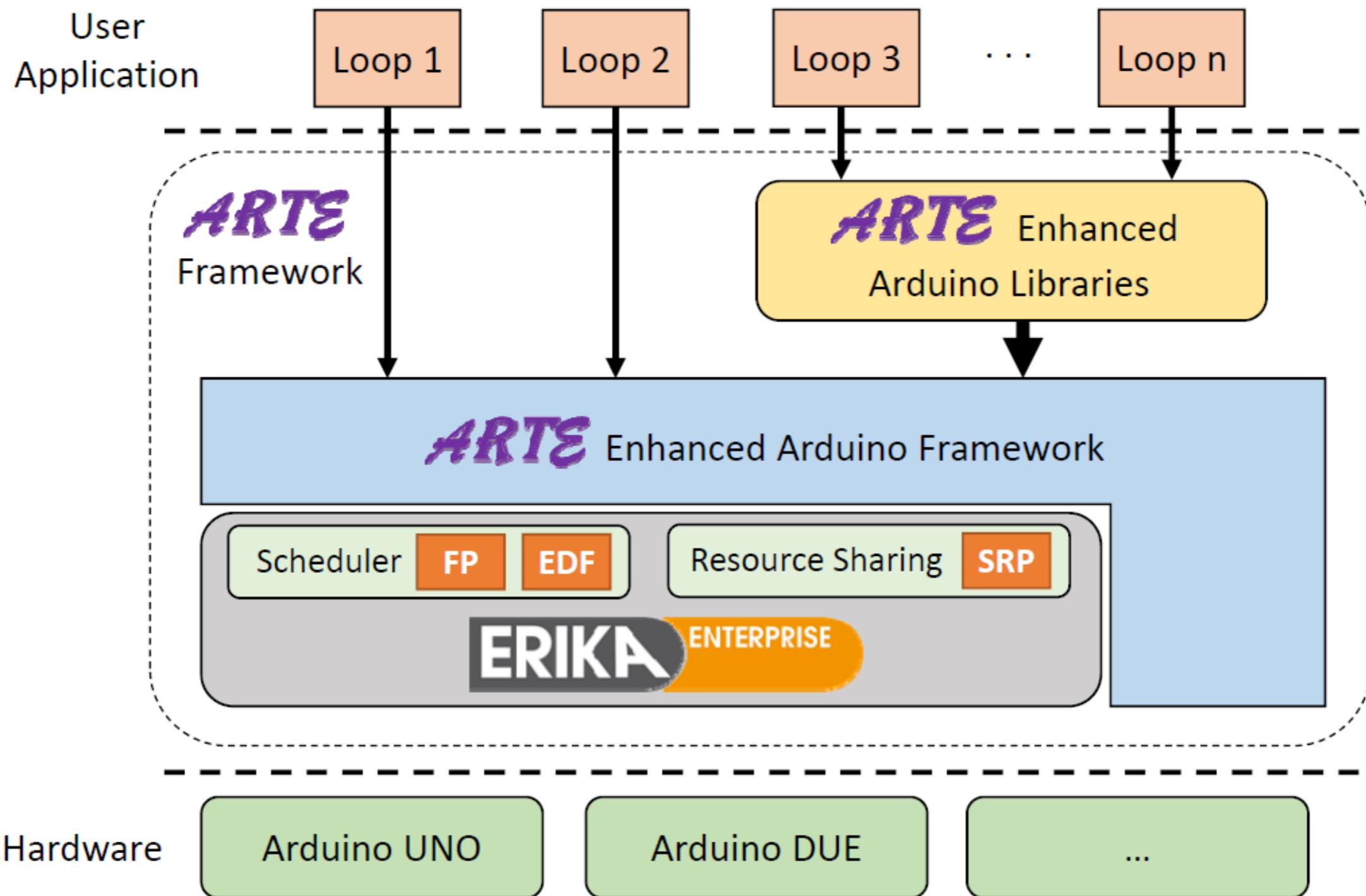
```
void loop() {  
  
    <instructions here>  
    delay(1000);  
}
```

ARTE: Arduino Real-Time Extension

GOAL

- Support for **multitasking**;
- Specification and enforcement of **real-time constraints (periodic activities)**;
- Maintain a **very simple programming interface** compliant with the Arduino philosophy;
- **Integration** with standard Arduino Libraries;
- **Minimal impact** on resources usage.

ARTE: Arduino Real-Time Extension



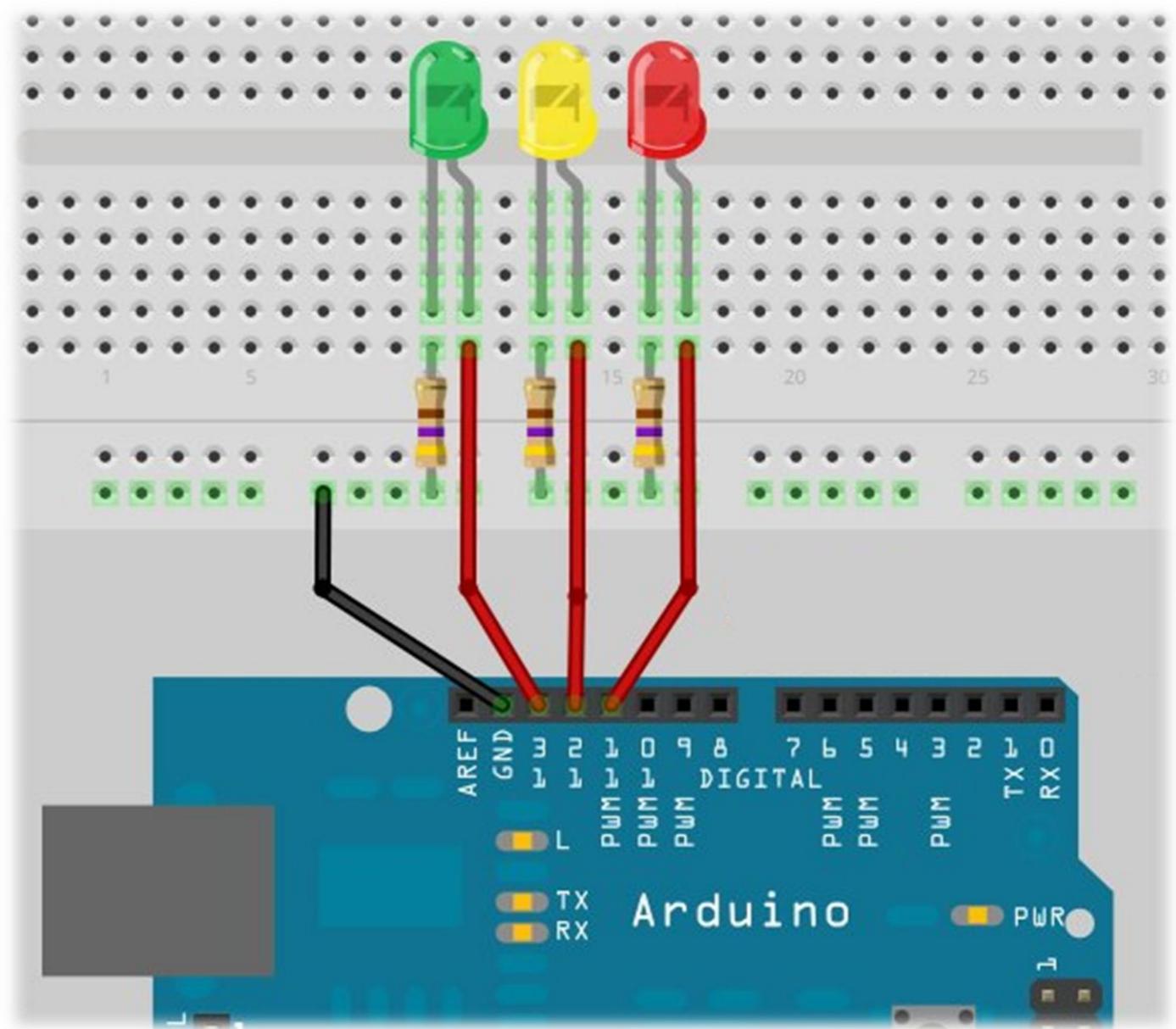
It is a **real-time kernel** for embedded platforms:

- **open-source** license;
- **efficient** (minimal footprint < 1 KB flash);
- **predictable** (fixed priority and EDF scheduling);
- runs on lot of platforms, including **multicore**;
- **certified** (compliant with **OSEK/VDX** standard);
- used by several **automotive** companies.

Example of use

Make blinking three different leds, each one at a different frequency.

- **Led1:** 3s
- **Led2:** 7s
- **Led3:** 11s



Robotmill.com

Example of use

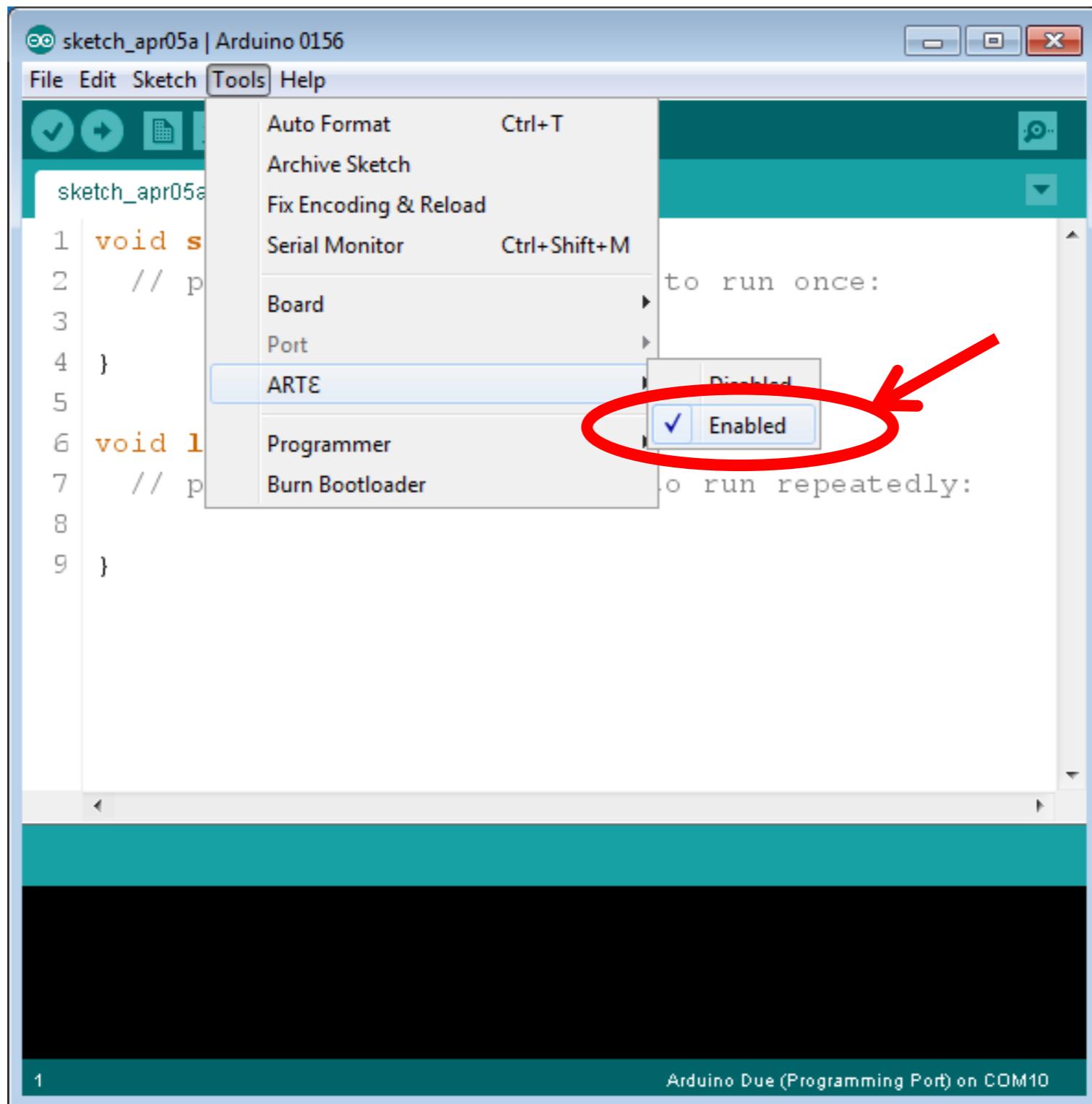
```
int led1 = 13, led2 = 14, led3 = 15;  
int count = 0;  
  
void loop() {  
    if (count%3 == 0)  
        digitalToggle(led1);  
  
    if (count%7 == 0)  
        digitalToggle(led2);  
  
    if (count%11 == 0)  
        digitalToggle(led3);  
  
    if (count == 3*7*11)  
        count = 0;  
    count++;  
    delay(1000);  
}
```

```
int led1 = 13;  
int led2 = 14;  
int led3 = 15;  
  
void loop1(3000) {  
    digitalToggle(led1);  
}  
  
void loop2(7000) {  
    digitalToggle(led2);  
}  
  
void loop3(11000) {  
    digitalToggle(led3);  
}
```

With classical **Arduino**
programming model

With **ARTE**

Example of use



Arduino Real-Time Extension

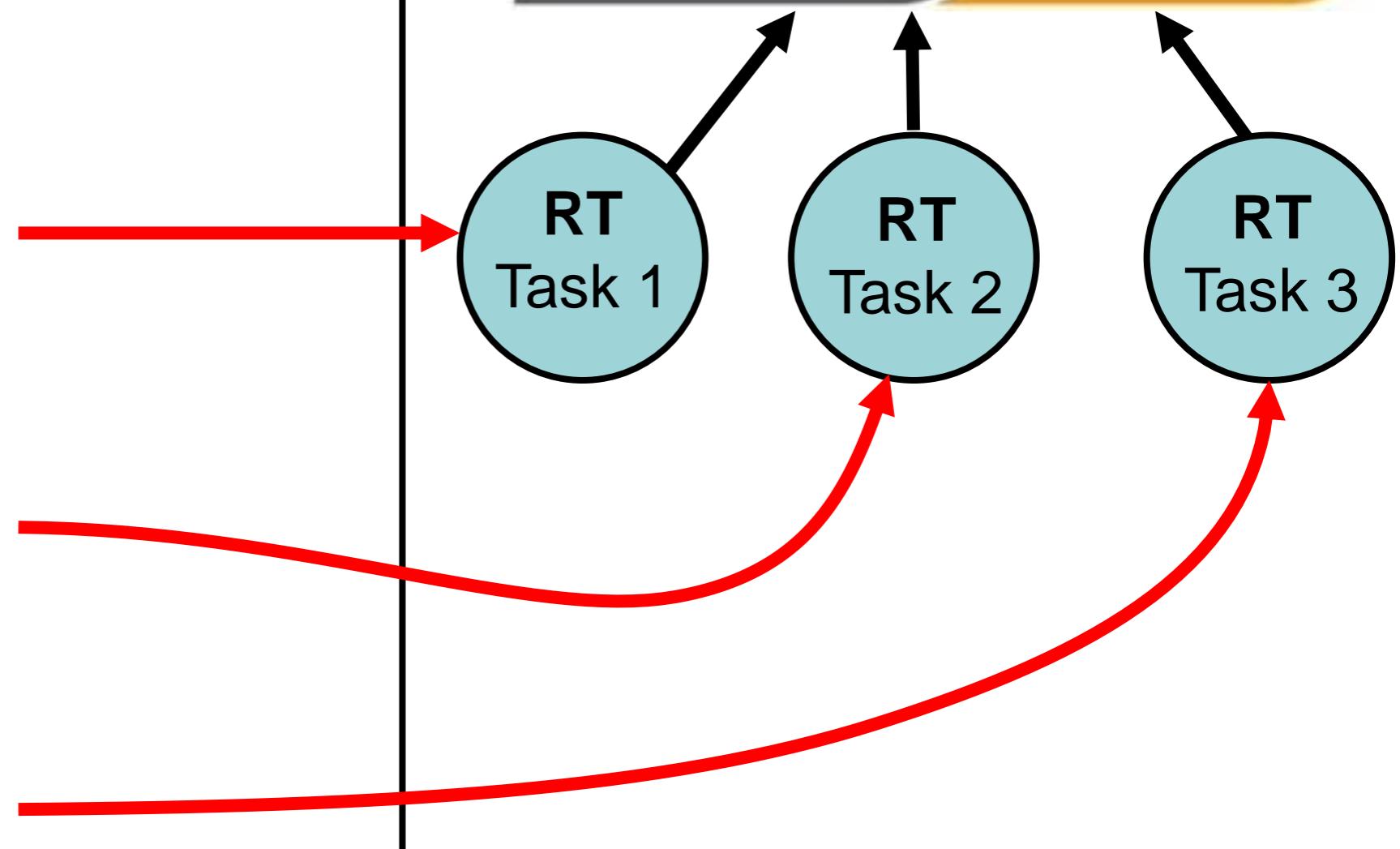
ARTE is a real extension to the **Arduino IDE**,
not an external library!



- It provides a **pre/post-processor** that avoids tiring and error-prone user configurations;
- **Automatic generation** of the application configuration and skeleton code;
- The user is focused on the application logic.

Example

```
int led1 = 13;  
int led2 = 14;  
int led3 = 15;  
  
void loop1(3000) {  
    digitalToggle(led1);  
}  
  
void loop2(7000) {  
    digitalToggle(led2);  
}  
  
void loop3(11000) {  
    digitalToggle(led3);  
}
```



ERIKA Enterprise is a **static** RTOS:

- All the RTOS configuration (*tasks, resources, counters, hardware configuration*) is decided at compile time;
- **Minimal** impact on the RAM and code memory;
- **OIL** language is used to configure the RTOS.

Example

□ Mapping to an OSEK application

```
int led1 = 13;  
int led2 = 14;  
int led3 = 15;  
  
void loop1(3000) {  
    digitalToggle(led1);  
}  
  
void loop2(7000) {  
    digitalToggle(led2);  
}  
  
void loop3(11000) {  
    digitalToggle(led3);  
}
```



OIL

```
TASK loop1 {  
    PRIORITY = 0x01;  
    SCHEDULE = FULL;  
    STACK = SHARED;  
};  
ALARM Alarmloop1 {  
    COUNTER = TaskCounter;  
    ACTION = ACTIVATETASK {  
        TASK = loop1;  
    }; };  
  
TASK loop2 {  
    PRIORITY = 0x02;  
    SCHEDULE = FULL;  
    STACK = SHARED;  
};  
ALARM Alarmloop2 {  
    COUNTER = TaskCounter;  
    ACTION = ACTIVATETASK {  
        TASK = loop2;  
    }; };  
  
TASK loop3 {  
    PRIORITY = 0x03;  
    SCHEDULE = FULL;  
    STACK = SHARED;  
};  
ALARM Alarmloop3 {  
    COUNTER = TaskCounter;  
    ACTION = ACTIVATETASK {  
        TASK = loop3;  
    }; };
```

Example

- Mapping to an OSEK application

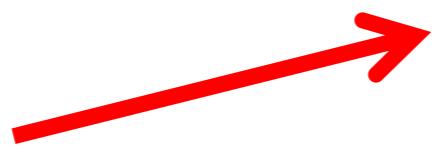


```
int led1 = 13;
int led2 = 14;
int led3 = 15;

void loop1(3000) {
    digitalToggle(led1);
}

void loop2(7000) {
    digitalToggle(led2);
}

void loop3(11000) {
    digitalToggle(led3);
}
```

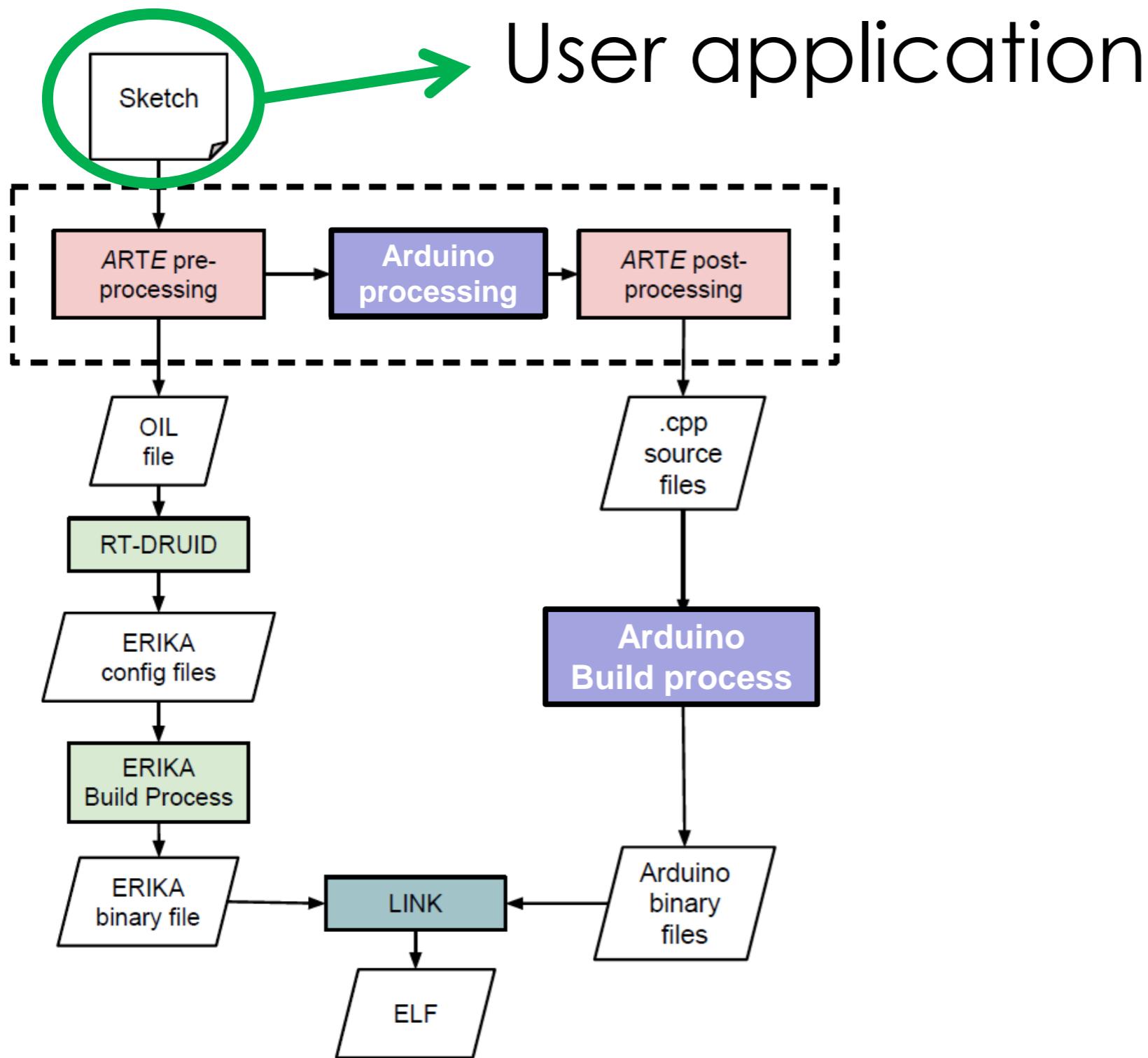


```
...
TASK loop2 {
    PRIORITY = 0x02;
    SCHEDULE = FULL;
    STACK = SHARED;
};

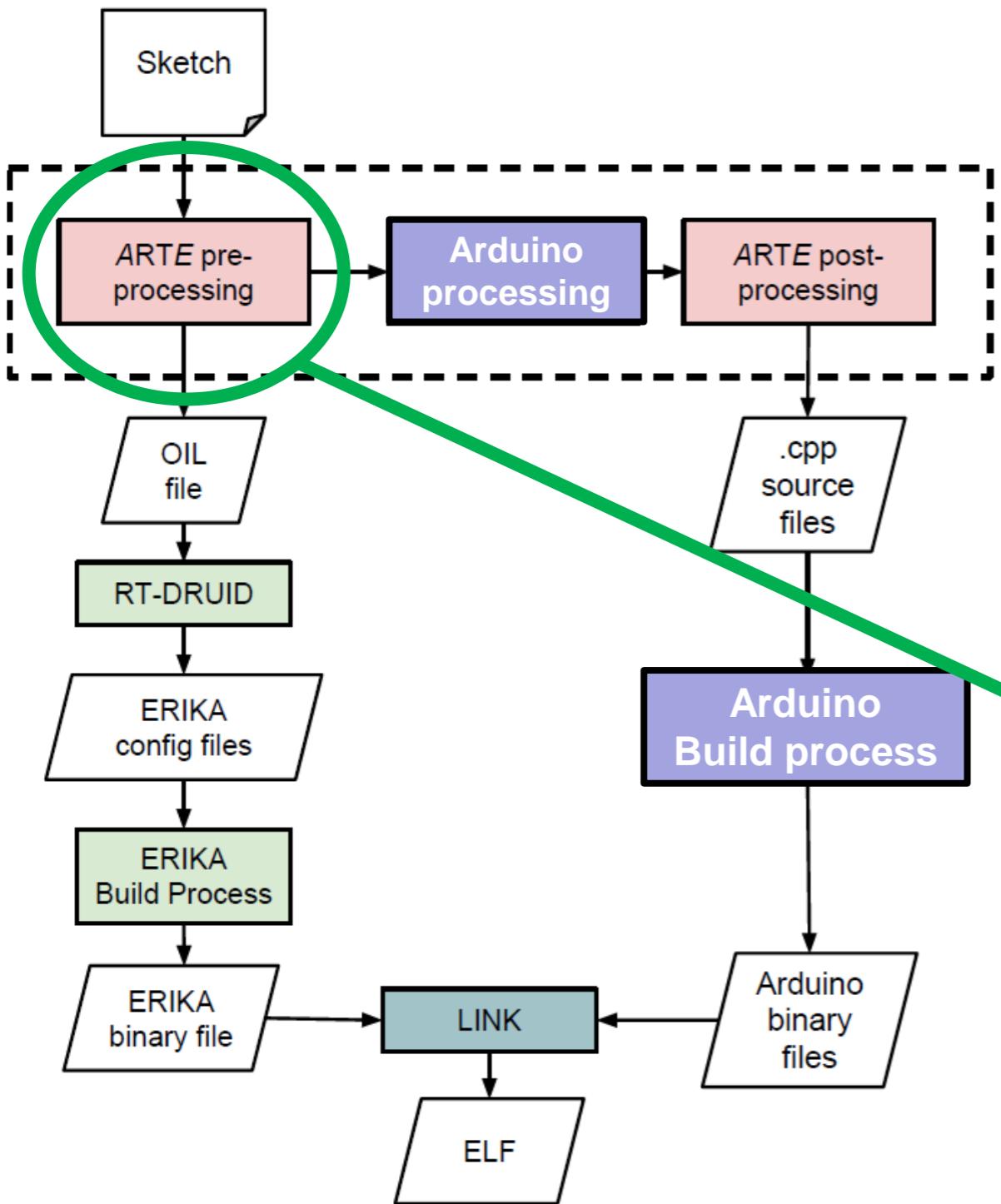
ALARM Alarmloop2 {
    COUNTER = TaskCounter;
    ACTION = ACTIVATETASK {
        TASK = loop2;
    }; };
...
```



Build Flow

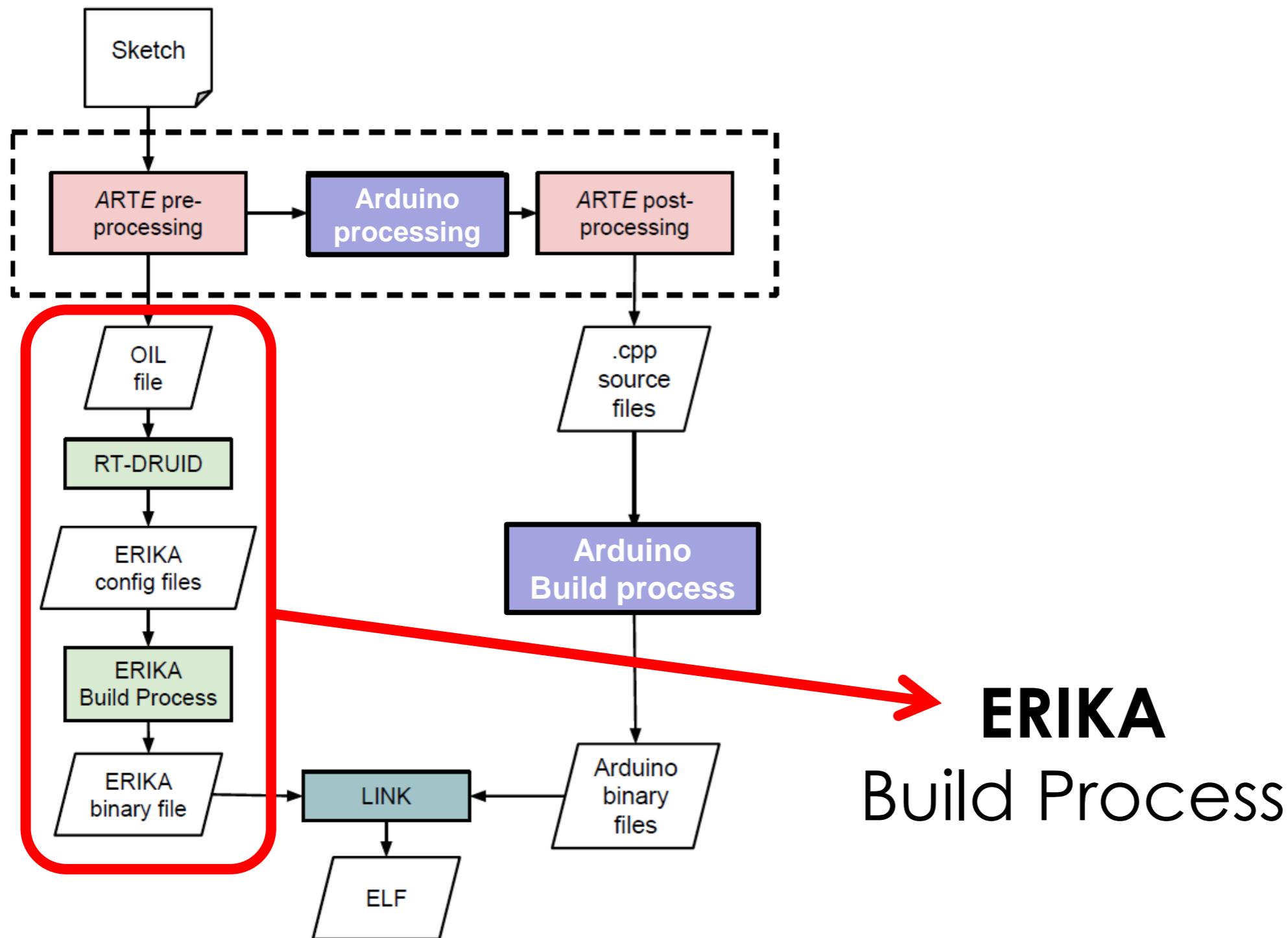


Build Flow



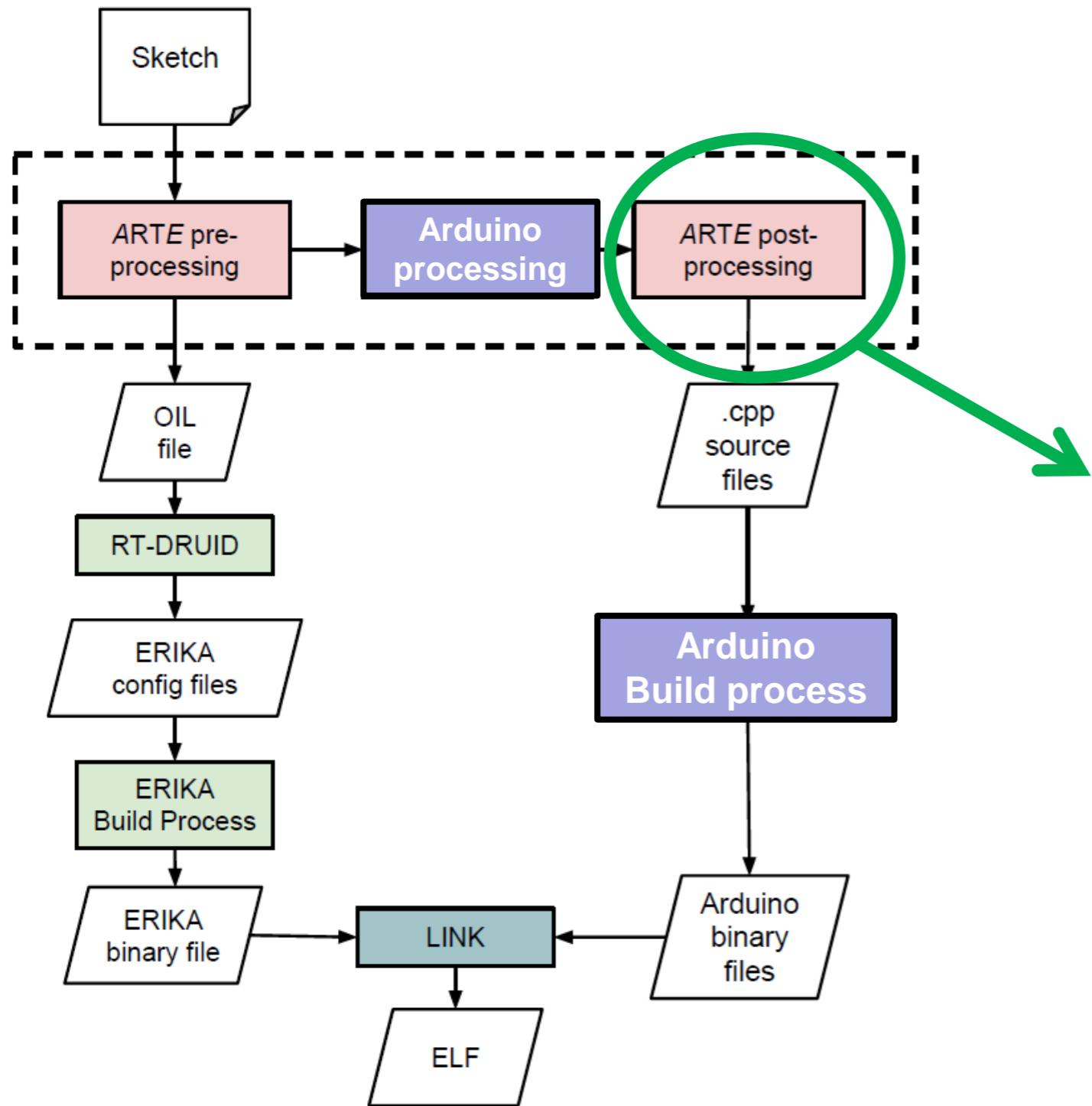
Generates the RTOS configuration

Build Flow



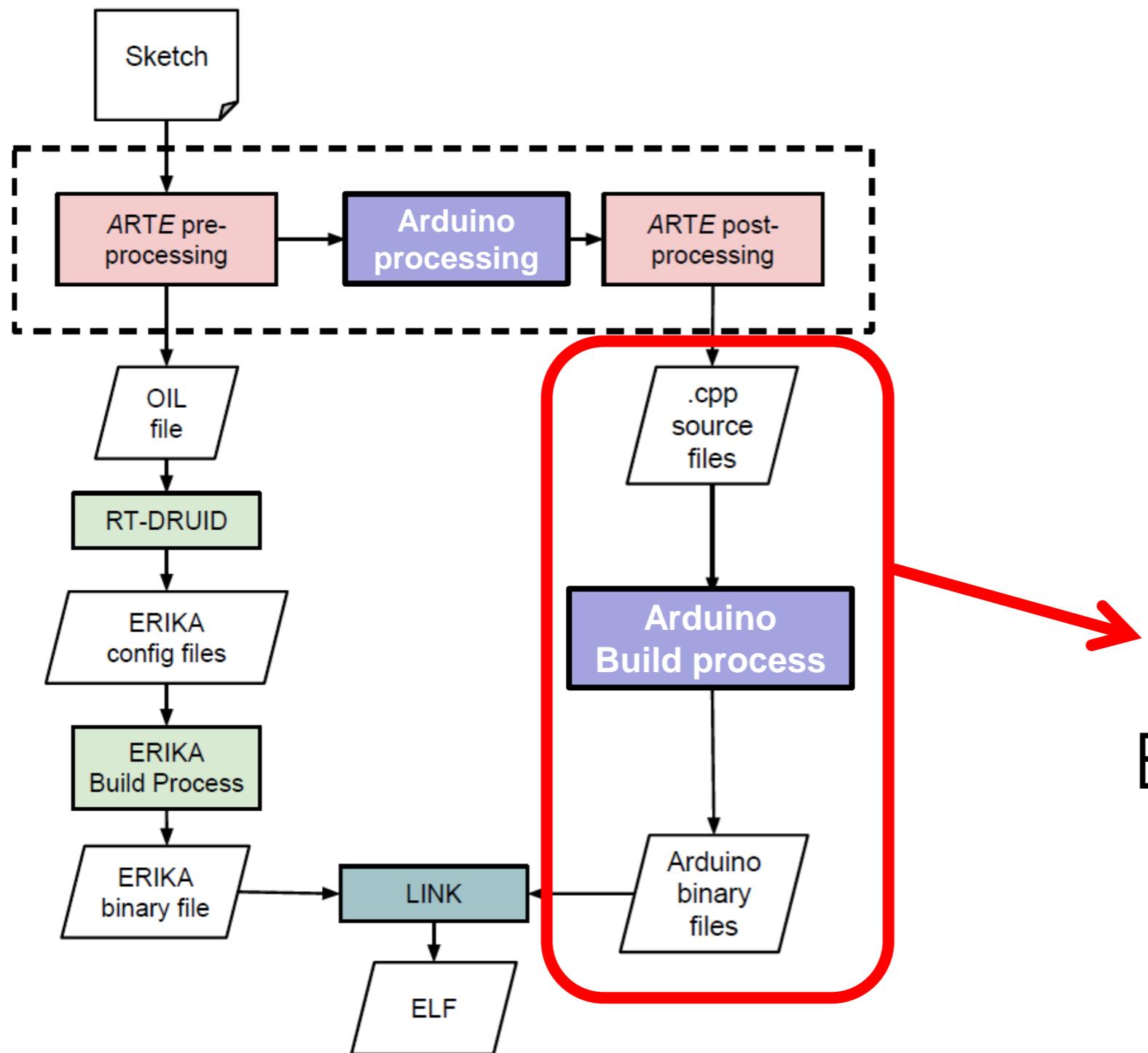
**ERIKA
Build Process**

Build Flow



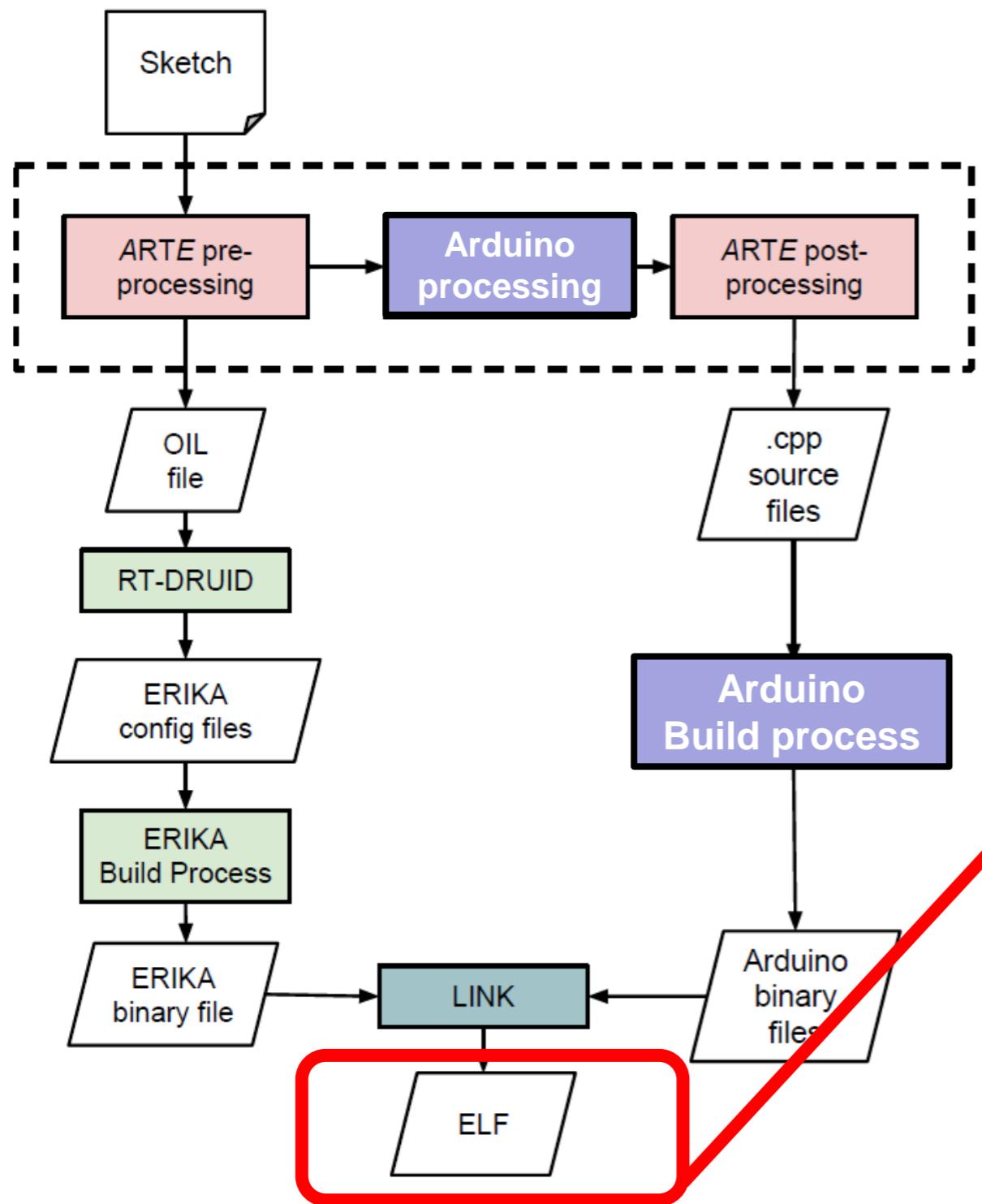
Generates an
ERIKA-compatible
application code

Build Flow

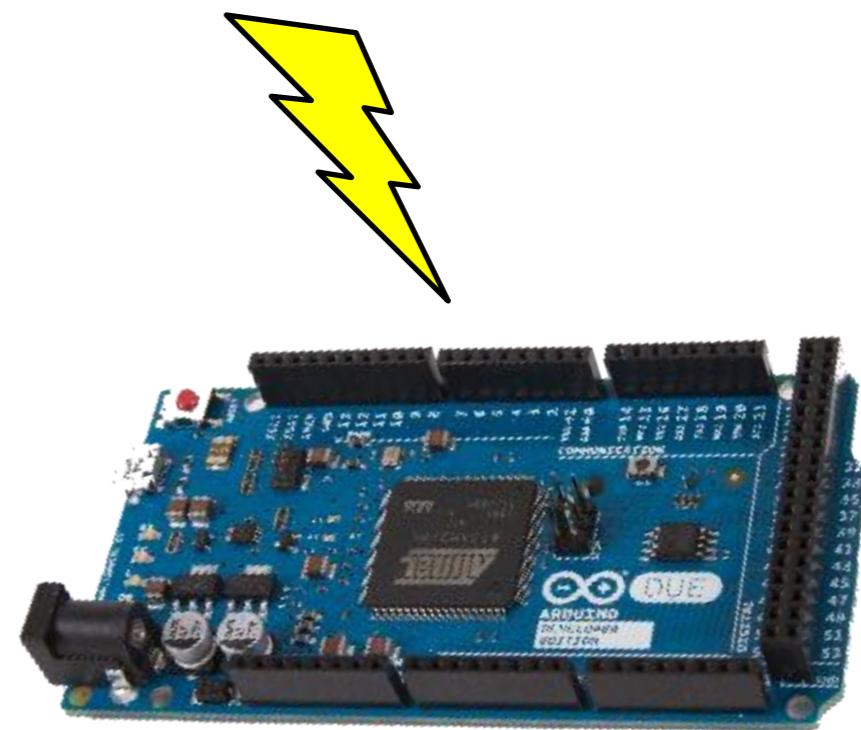


**Arduino
Build Process**

Build Flow



Flashing of the binary
on the Arduino board



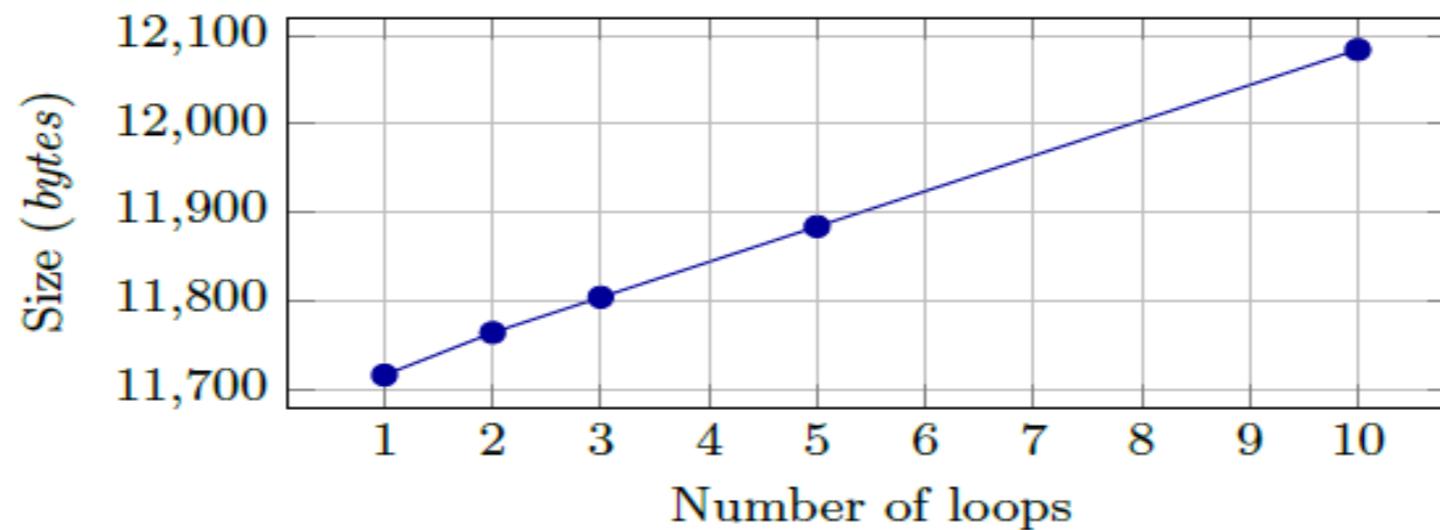
Footprint evaluation

Arduino

Code memory (bytes)	
UNO	DUE
466	10,500

Arduino + *ARTE*

# Loops	UNO	DUE
1	802	11,716
2	822	11,764
3	840	11,804
5	878	11,884
10	974	12,084



Conclusions

ARTe is an extension for the Arduino framework

- Multiprogramming (periodic threads)
- Real-time execution through a RTOS
- Simple programming interface with minimal impact on the Arduino programming model

Project Webpage:
<http://retis.sssup.it/ARTE>

Project Webpage:
<http://retis.sssup.it/ARTe>

Thank you!

Pasquale Buonocunto
p.buonocunto@sssup.it