



Programmation bare metal



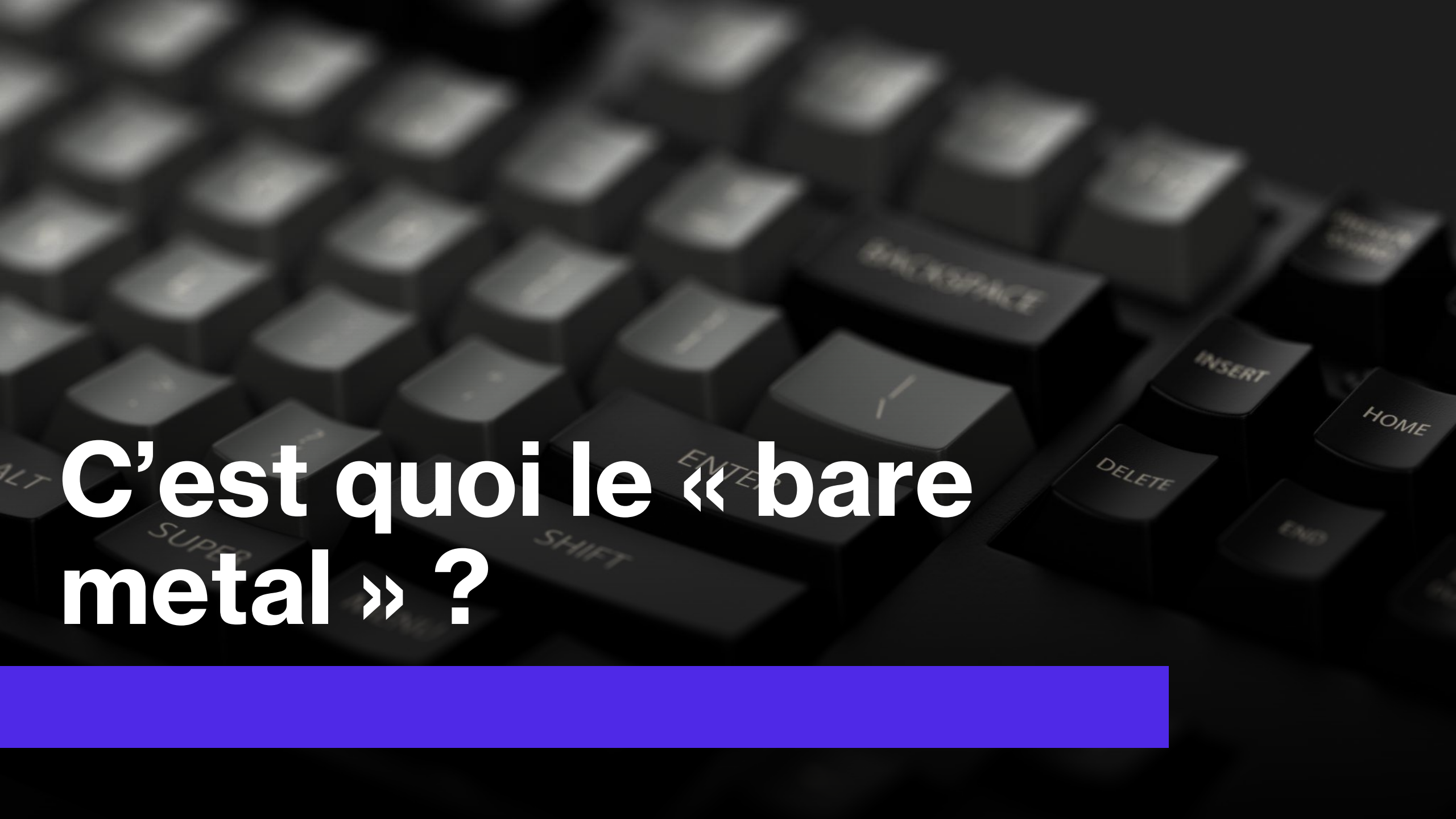
Sommaire

I – Qu'est-ce que le « bare metal » ?

II – Objectifs du projet

III – Réalisations

Conclusion



C'est quoi le « bare metal » ?

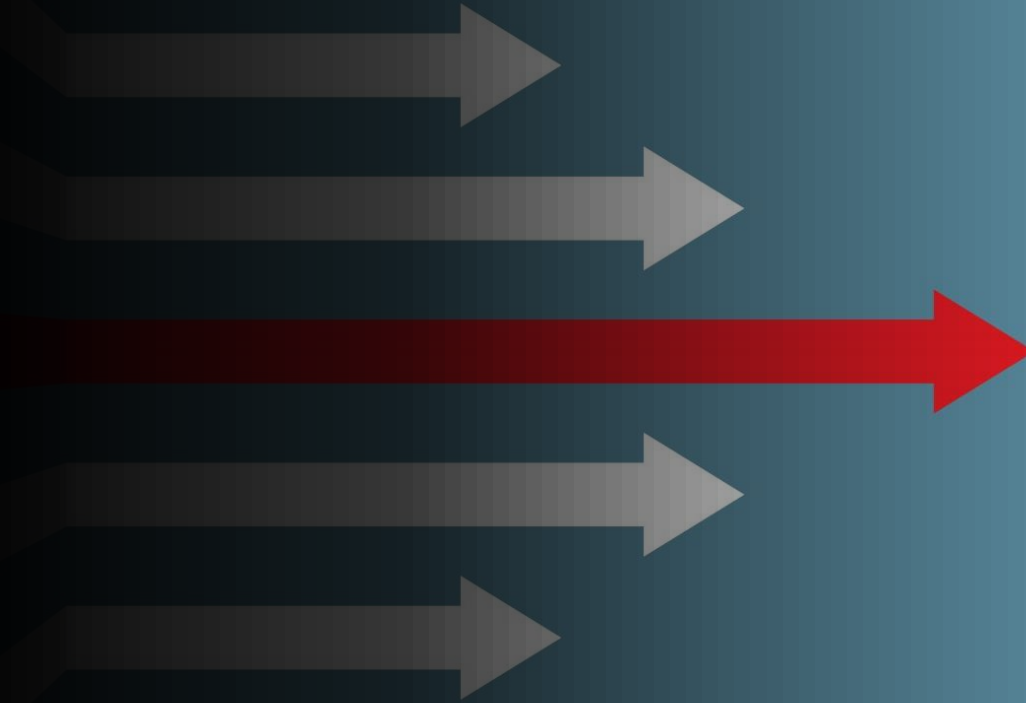


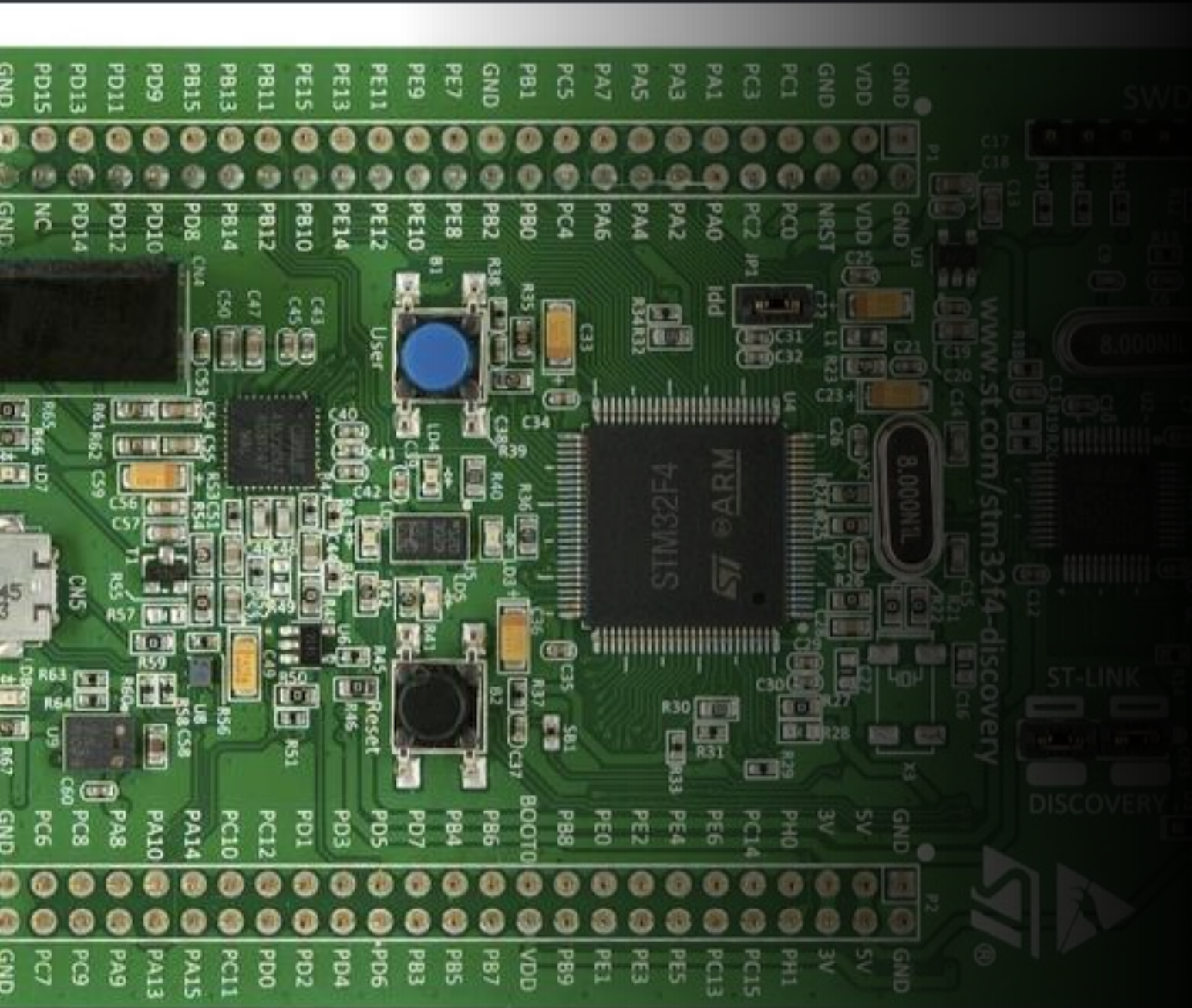


**On appelle 'bare metal',
ou 'programmation bare
metal', un système
informatique matériel
utilisé sans système
d'exploitation complexe
mais directement par
un programme,
généralement à l'aide
d'une bibliothèque
spécialisée. (Wikipédia)**

Objectifs

- Comprendre le mécanisme de compilation d'un programme C et l'architecture d'un exécutable
- Faire exécuter ce programme sur un microprocesseur arm et observer son fonctionnement





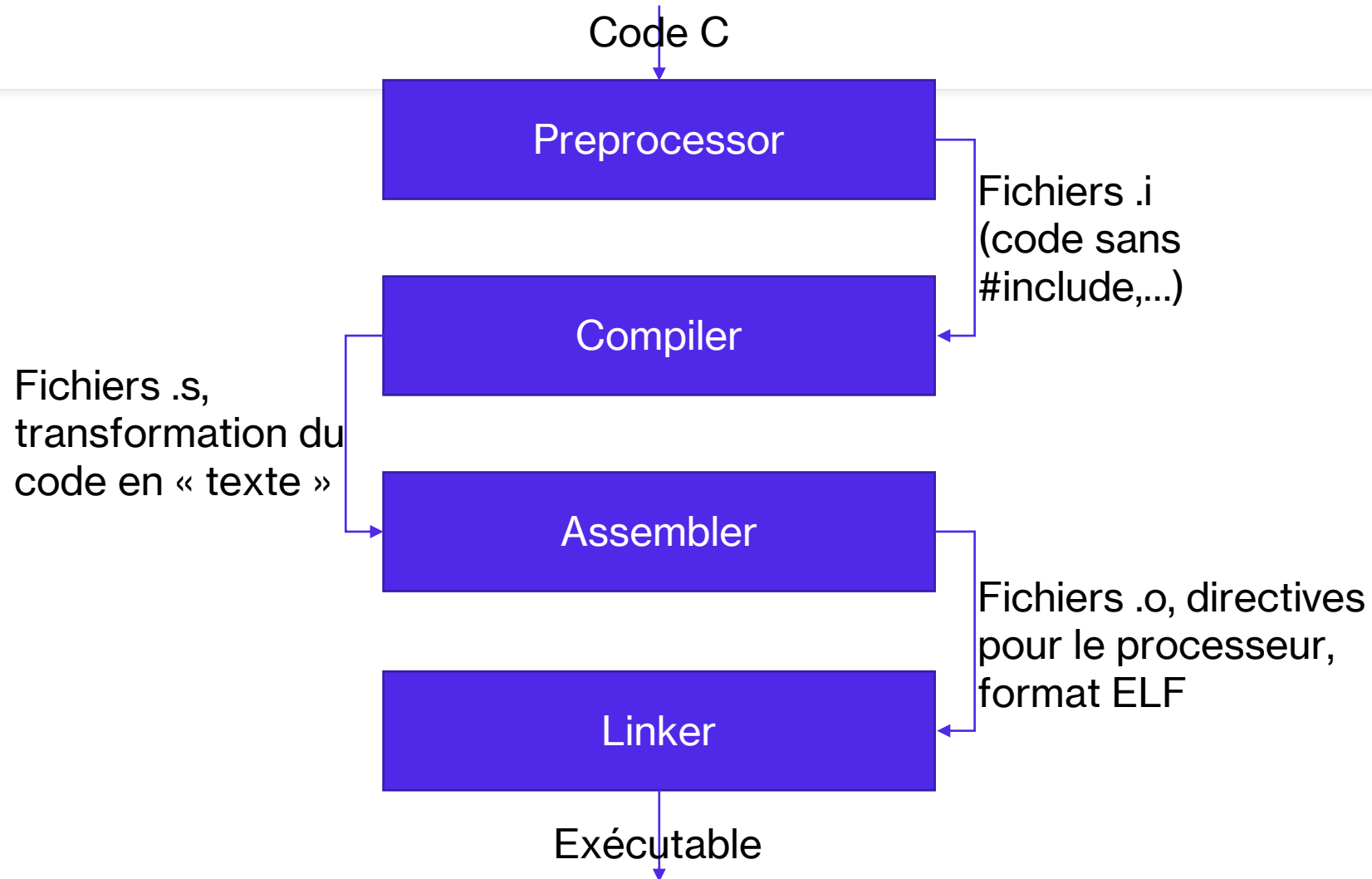
Outils utilisés

- La bibliothèque gdb-multiarch
- Un exécutable pour émuler la carte qemu-arm-gnuarmeclipse



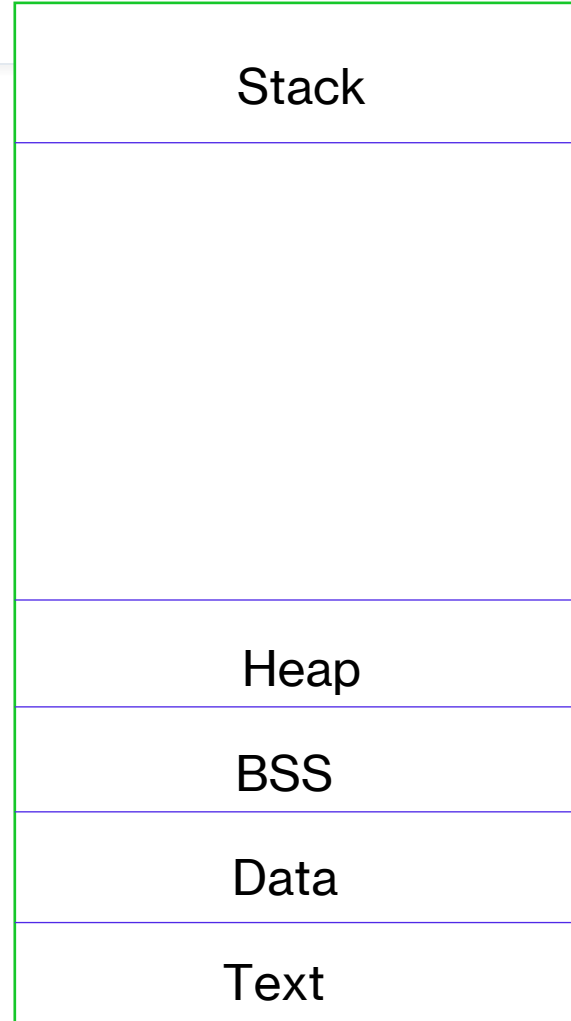
Réalisations

Compilation d'un programme C



Format ELF (Executable and Linkable Format)

Rodata :
constantes



Variables nulles

Autres variables

Code



Linker script et crt0

```
MEMORY
{
    flash (rx) : ORIGIN = 0x00000000, LENGTH = 1M
    sram1 (wx) : ORIGIN = 0x20000000, LENGTH = 112K
    sram2 (wx) : ORIGIN = 0x2001C000, LENGTH = 16K
}

ENTRY(_start)

_sp_init = ORIGIN(sram2) + LENGTH(sram2);

SECTIONS
{
    .vector_table : {
        *(.vector_table)
    }>flash

    .crt0 : {
        crt0.o(*)
        *(.crt0)
    }>flash

    .text : {
        _tstart = .;
        *(.text)
        _tend = .;
    } >sram1 AT> flash
    _loadTextA = LOADADDR(.text);

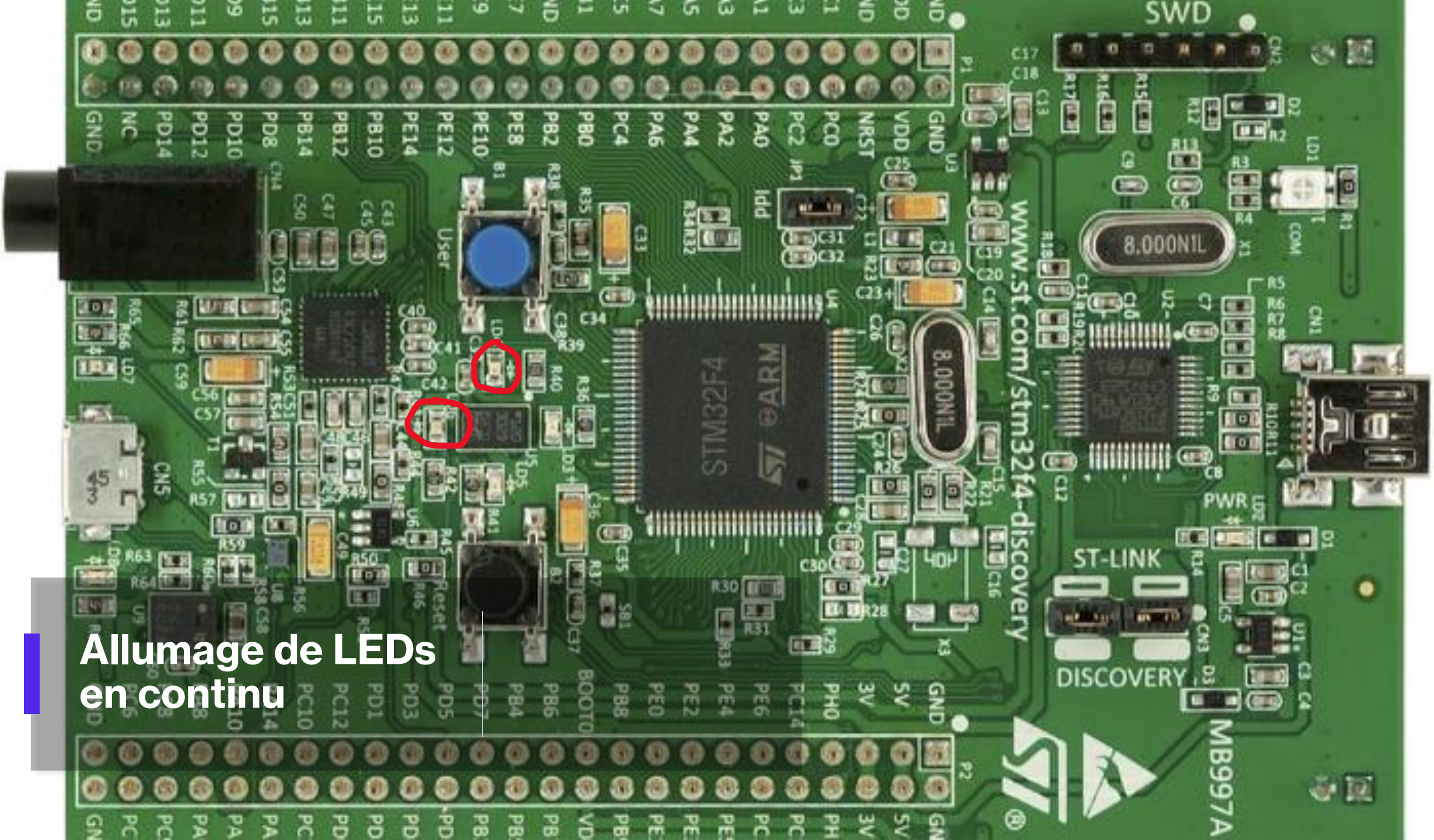
    .rodata :
    {
        *(.rodata)
    }>flash

    .data :
    {
        _dstart = .;
        *(.data)
        _dend = .;
    }
```

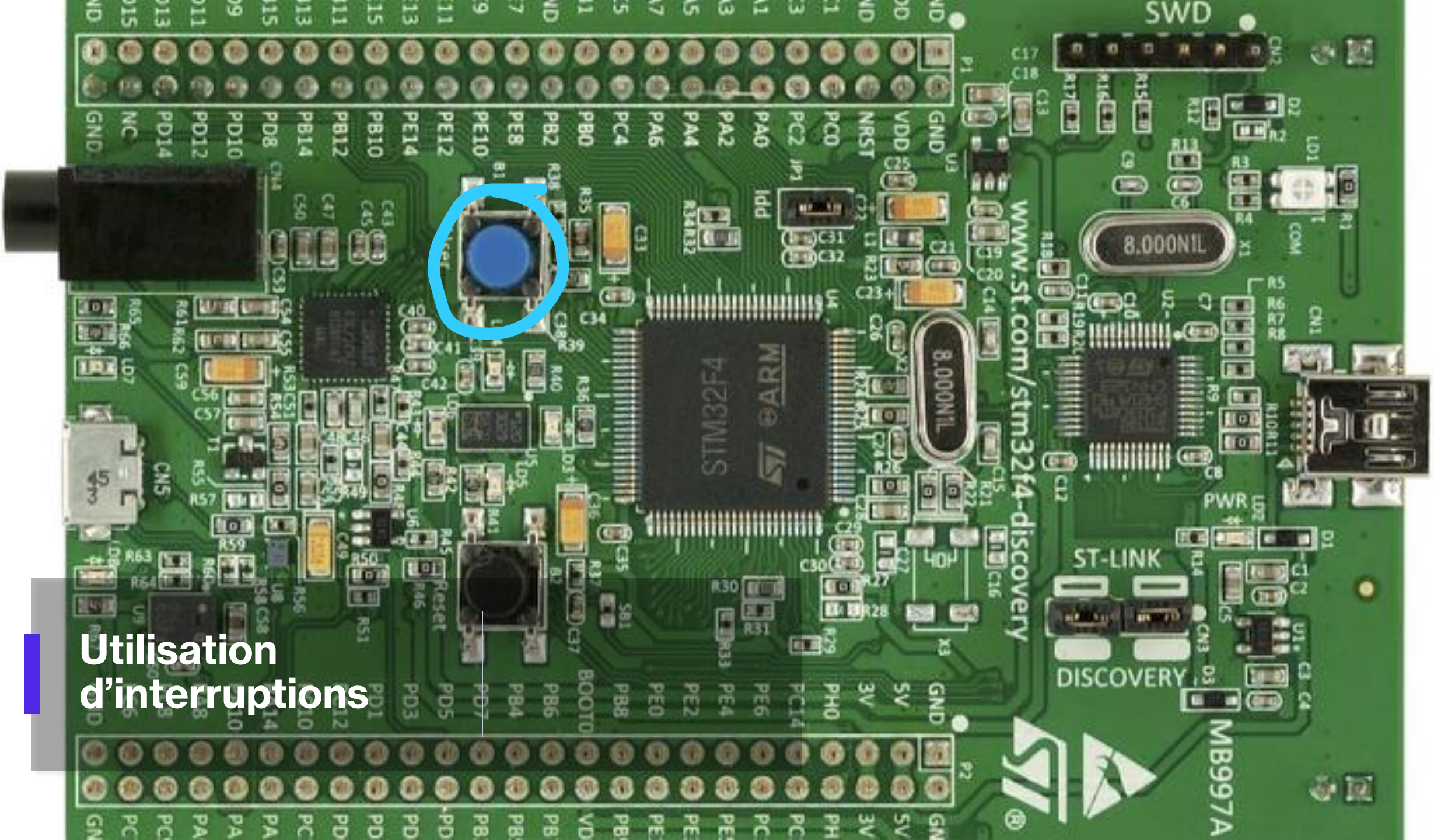
```
.global _start
.syntax unified
.thumb
.thumb_func
_start:
    ldr sp,=_sp_init
    bl copyData
    bl fillZeroBSS
    bl main

loop:
    b loop
```

Allumage de LEDs
en continu



Utilisation d'interruptions





Conclusion



**Merci de
m'avoir
écouté.
Posez vos
questions**

