

Soutenance POEI

Mini-projet Morse



Mercredi 19 avril 2023

© Groupe SII 2021 | Usage interne SII

8 rue des Pirogues de Bercy, 75012 PARIS

Intervenants



Djibril BAH



Lena SAVY-LARIGALDIE



Malik BERDIER

Plan

1. Introduction

Lena & Malik

2. Partie PC

Djibril

3. Partie Timers

Malik

4. Partie Communication

Lena

5. Démonstration

Djibril

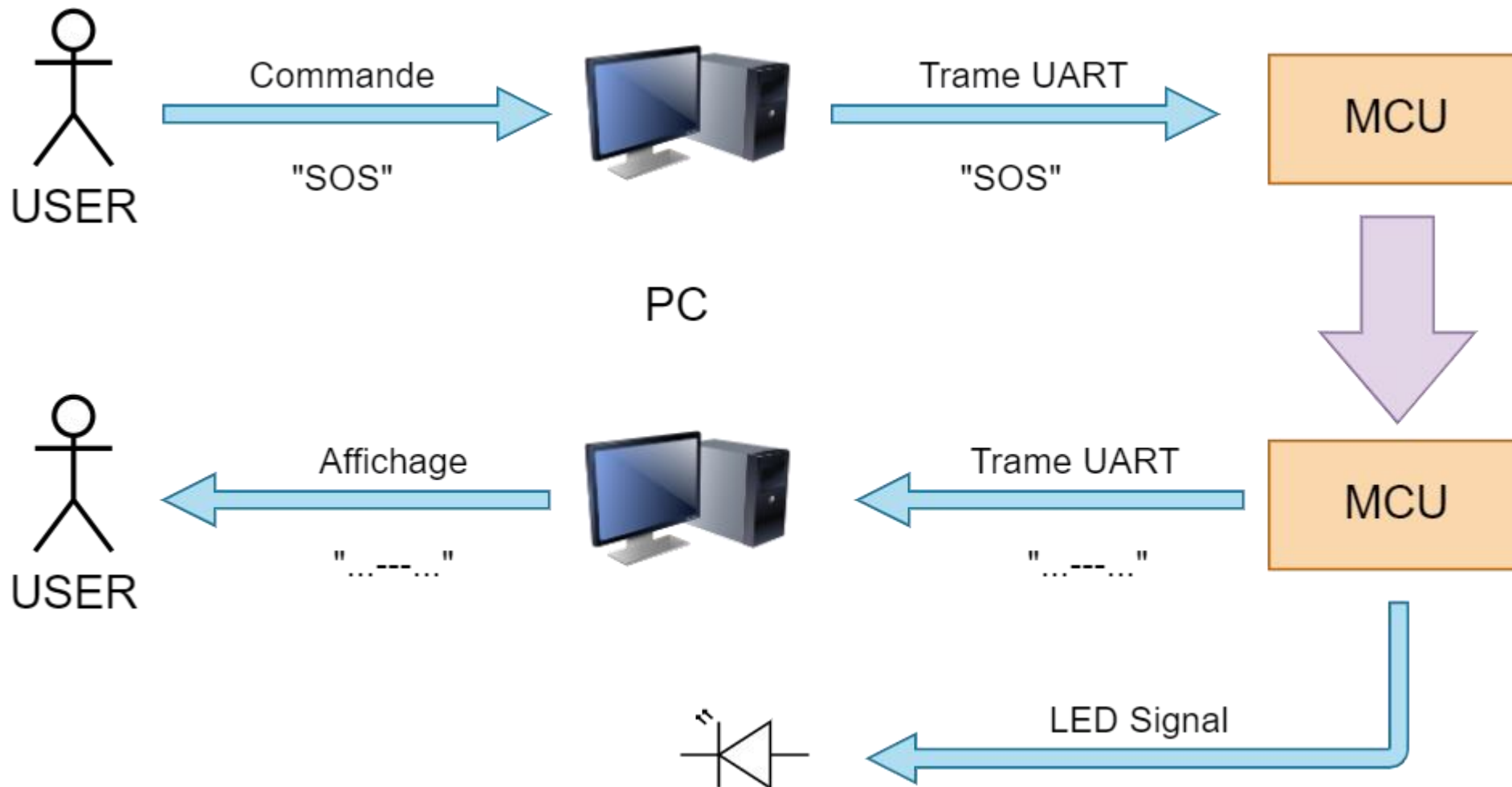
7. Remerciement

Djibril

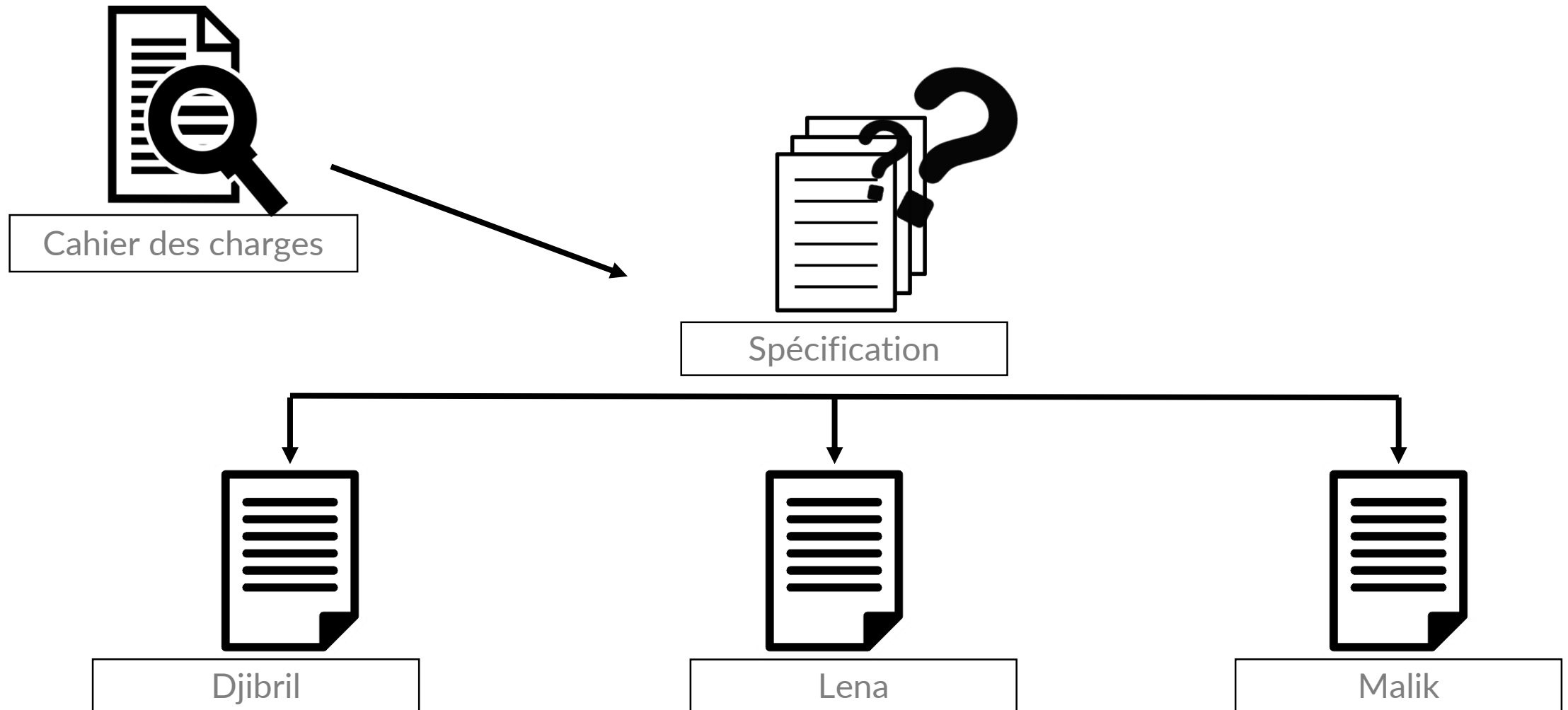
Introduction



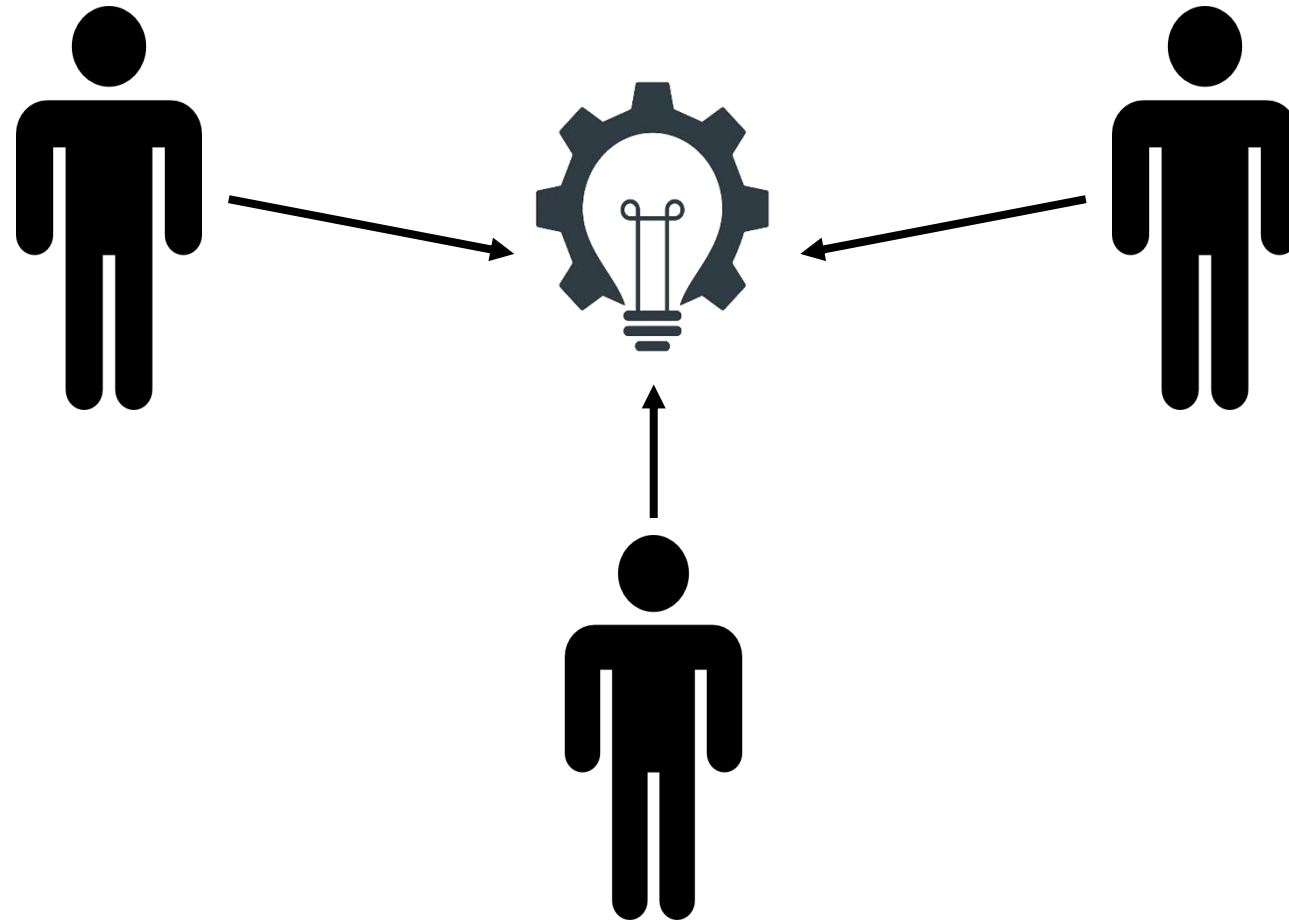
Contexte



Organisation



Organisation



Organisation

Exigences B - Modes

id	version	description
b001	v1.0	Il y a les modes <code>boucle</code> et <code>nombre</code>
b002	v1.0	Le mode <code>boucle</code> repete la transcription morse en boucle
b003	v1.0	Le mode <code>nombre</code> fait repeter la transcription morse un nombre de fois renseigne avec l'argument <code>-n</code>
b004	v1.0	Les modes <code>boucle</code> et <code>nombre</code> ne peuvent etre utilises ensembles

Exigences C - Commande PC

id	version	description
c001	v1.0	L'utilisateur du PC doit saisir avec l'argument <code>-m</code> le message textuel
c002	v1.0	L'utilisateur du PC peut saisir avec l'argument <code>-t</code> la duree des temps courts, moyens et longs du message morse
c003	v1.0	L'utilisateur du PC peut saisir l'argument <code>-b</code> qui transmet le mode <code>boucle</code>
c004	v1.0	L'utilisateur du PC peut saisir avec l'argument <code>-n</code> le mode <code>nombre</code>
c005	v1.0	L'utilisateur du PC peut saisir l'argument <code>-s</code> qui transmet l'arret de la transcription
c006	v1.0	L'utilisateur du PC peut saisir avec l'argument <code>-h</code> pour afficher l'aide de la commande

Organisation

Convention de nommage :

- Fonctions : SII_MODULE_NomFonction()
- Variables : sii_module_nomVariable
- Struct : SII_MODULE_NomStructure
- Types : SII_MODULE_NomType_t
- Enum : SII_MODULE_NomEnum
- Fichiers : sii_module.c/h

```
typedef enum { MODE_IDLE, MODE_LOOP , MODE_ITERATE} SII_MODES_Mode_t;

/* Select */
SII_MODES_Mode_t SII_MODES_GetMode(SII_DATA_Data_t* Data);
void SII_MODES_RunMode(SII_DATA_Data_t* Data, uint8_t* message, SII_MODES_Mode_t Mode);

/* Modes */
void SII_MODES_Loop(SII_DATA_Data_t* Data, uint8_t* message);
void SII_MODES_Iterate(SII_DATA_Data_t* Data, uint8_t* message);
```

Organisation

Djibil



PC

Lena



Messages & Signal

Malik



Timers

Organisation



Function SII_MODES_GetMode()

- Ref : b001 b002 b003 b004 (Specification.md)
- Input : SII_MODES_Data
- Output : SII_MODES_Mode_t

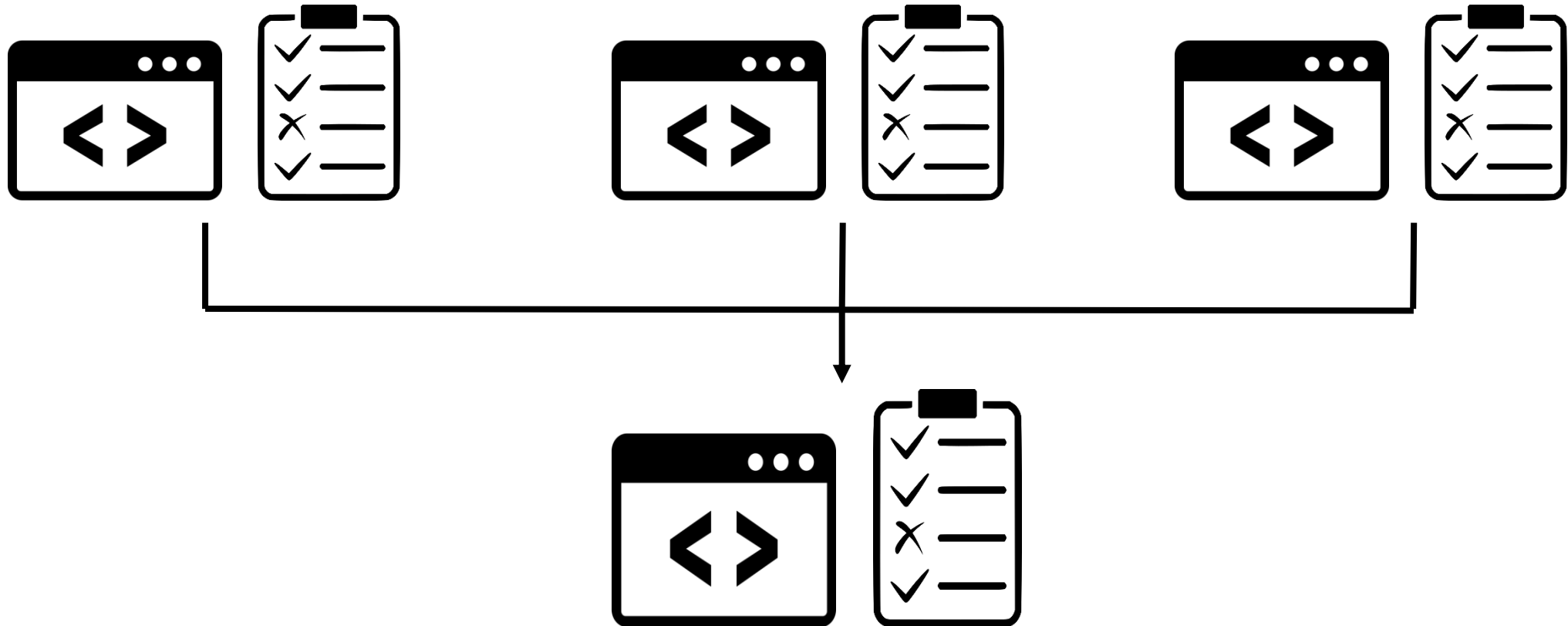
```
if SII_MODES_Data.loop == 1
|   return MODE_LOOP
else if SII_MODES_Data.interate >= 1
|   return MODE_INTERATE
```

Function SII_MODES_RunMode()

- Ref : b001 b002 b003 b004 (Specification.md)
- Input : sii_modes_mode
- Output : SII_MODES_Loop() , SII_MODES_Iterate()

```
if sii_modes_mode == MODE_LOOP
|   do SII_MODES_Loop()
else if sii_modes_mode == MODE_ITERATE
|   do SII_MODES_Iterate()
```

Organisation



Organisation

Test Intégration Function SII_MODES_RunMode()

Ref Spécification : b001 b002 b003 b004 (Specification.md)

Résumé du test :

Vérification de la sélection entre les différents modes '(MODE_ITERATE, MODE_LOOP)'

Etat initiale :

- Fin du test integration Test Intégration Function Main()

Procédure d'exécution :

1. Alimenter la carte
2. Un breakpoint est positionné sur l'appel de la fonction 'SII_MODES_RunMode()'

Vérifications nécessaire :

1. Si la variable `sii_modes_mode` vaut 'MODE_LOOP' lance l'appel à la fonction Function `SII_MODES_Loop()`
2. Si la variable `sii_modes_mode` vaut 'MODE_ITERATE' lance l'appel à la fonction Function `SII_MODES_ITERATE()`

Organisation

Test validation Function Mode()

Ref Spécification : b001 b002 b003 b004(Specification.md)

Resume du test :

Tester que le message boucle en continue ou par itération selon la saisie de l'utilisateur

Etat initial :

- La carte est démarrée
- Le terminale et la commande de lancement du script sont lancé, en attente de la saisie de l'utilisateur

Procédure d'exécution :

1. L'utilisateur saisit dans la ligne le commande `-b` pour envoyer en continue le message
2. L'utilisateur saisit dans la ligne le commande `-n` pour répéter n fois le message

Vérification nécessaire :

1. La LED répète en boucle le message morse.
2. La LED répète n fois le message morse.

Module Timer



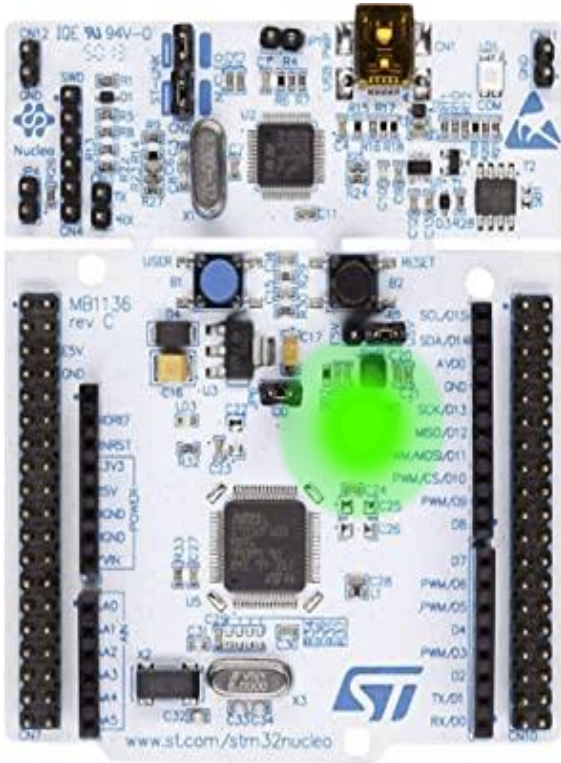
Module Timers

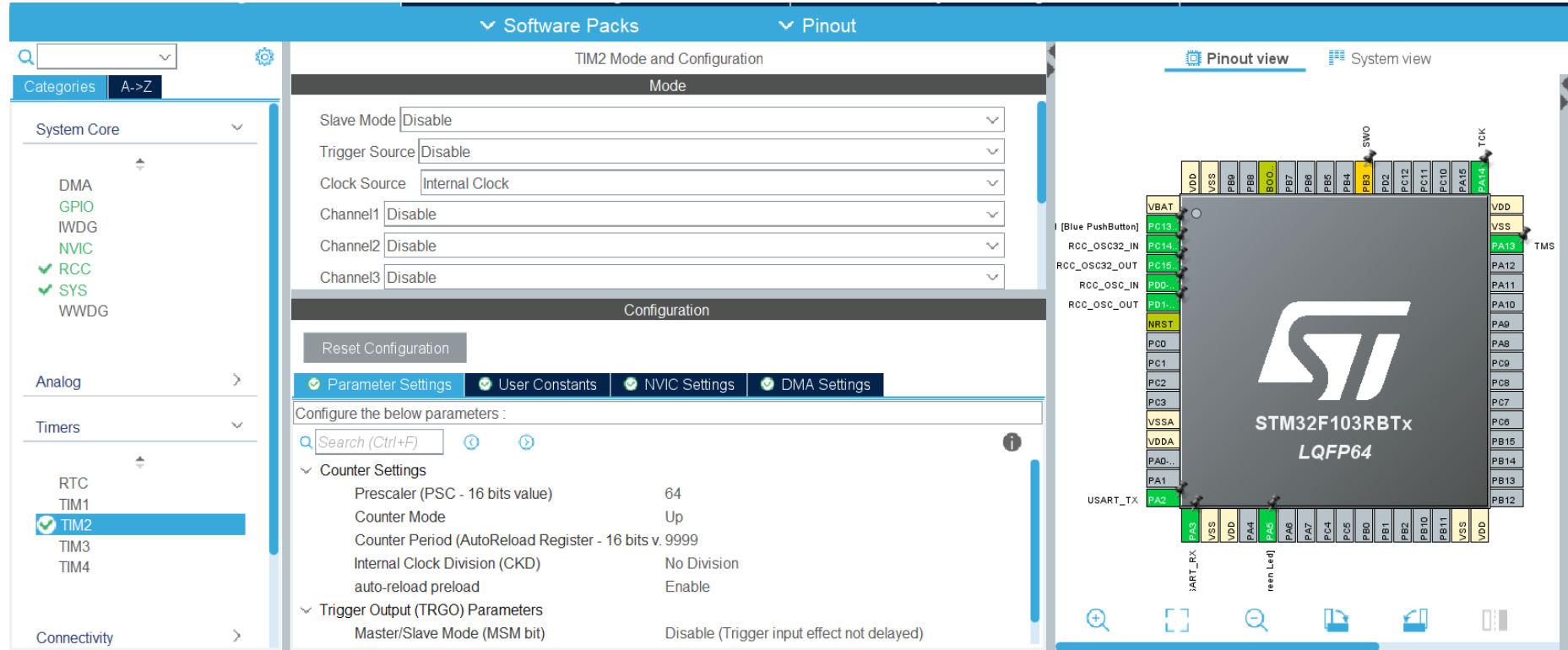
Objectif :

- Principe : compte les coups d'horloge
- Utilisation : Par dépassement d'une valeur seuil, compte une milliseconde

$$F_{Timer} = \frac{F_{clk}}{(1 + prescaler) \times (1 + periode)}$$

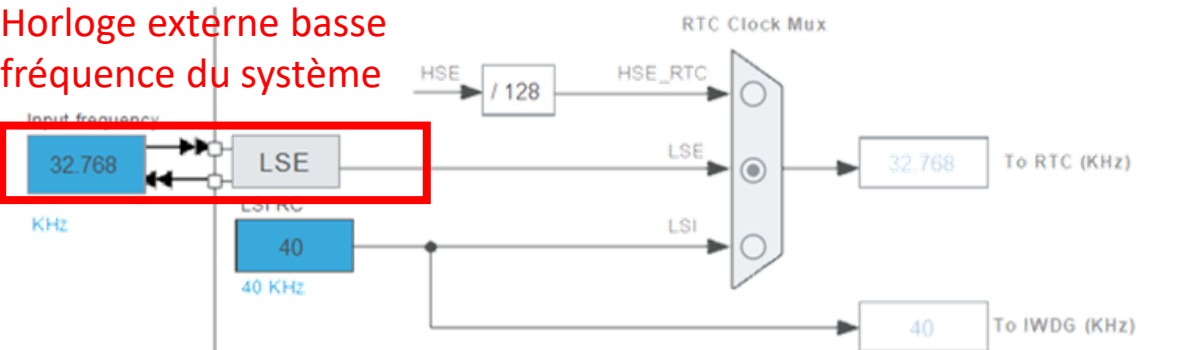
- F_{Timer} : fréquence de timer pour se réinitialiser
- F_{clk} : fréquence d'horloge
- **prescaler**: divise la fréquence d'horloge pour avoir la fréquence du timer voulue
- **periode** : valeur maximale du compteur du timer





Module Timers

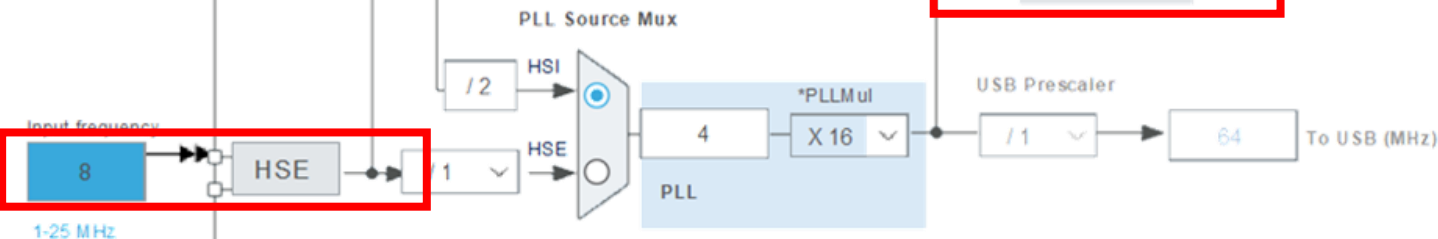
Horloge externe basse
fréquence du système



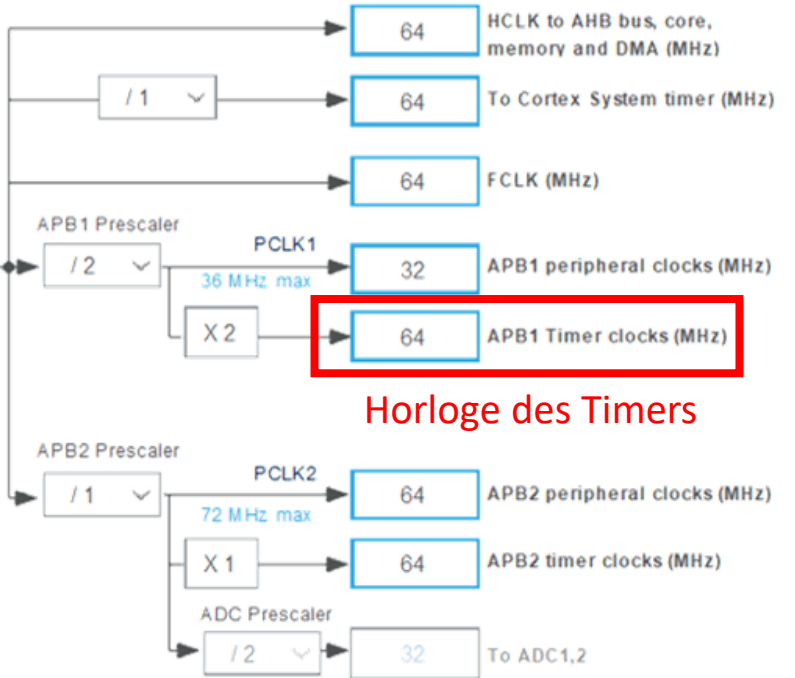
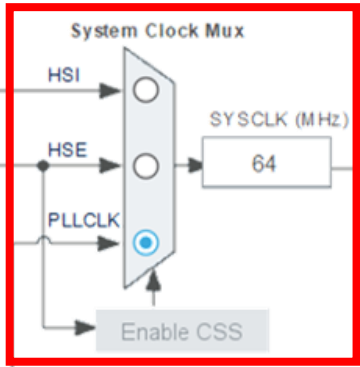
Horloge du système



Horloge interne
haute fréquence
du système



Horloge externe haute fréquence du système



Horloge des Timers

Commande PC



Les options de commande

Option	Arguments	Fonction
-m	"message"	Message a transcrire
-b		Mode boucle
-n	0-255	Mode iteratif
-t	"0-255 0-255 0-255"	Configuration des timers
-s		Arret de la transcription

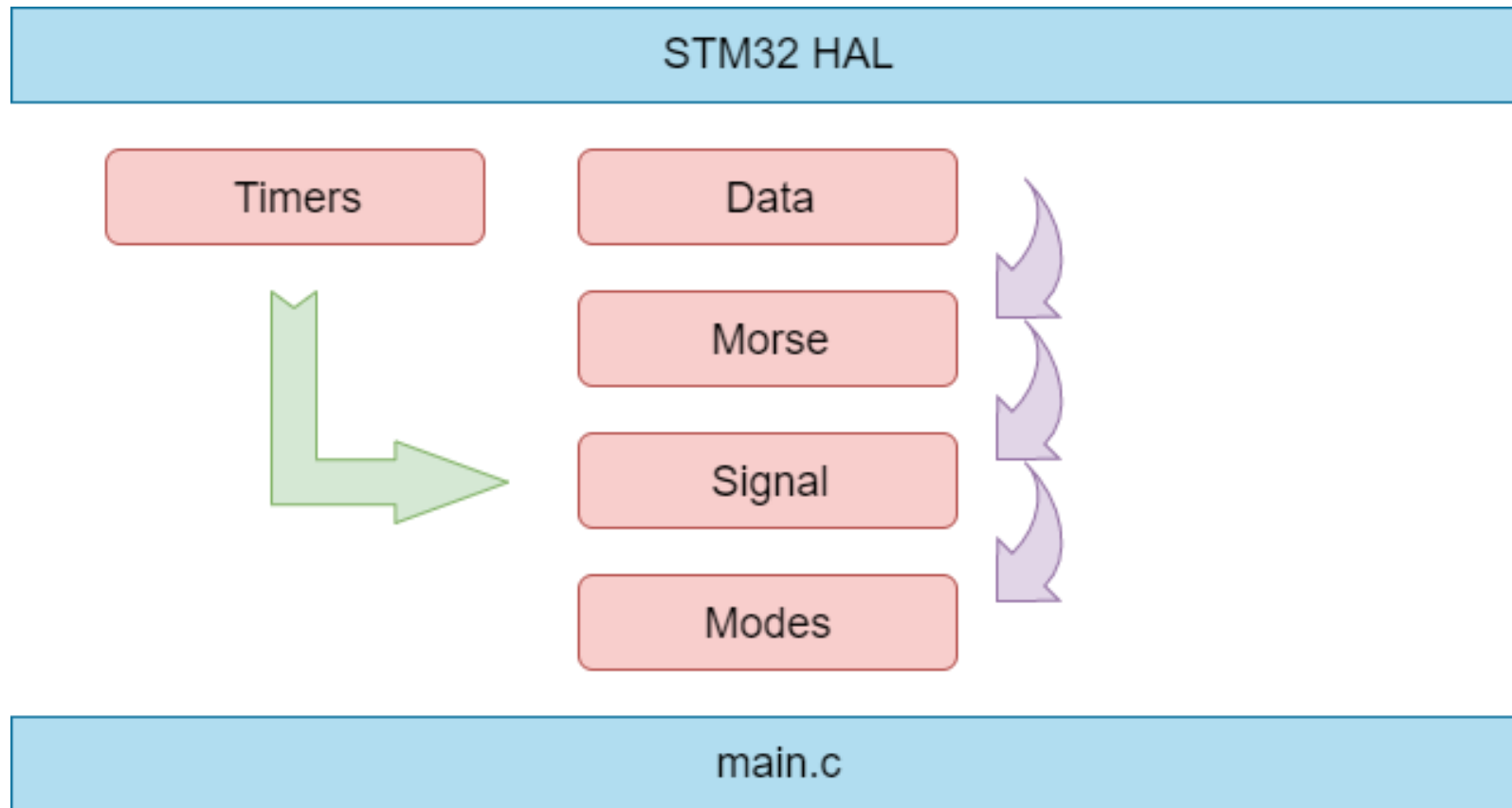
La trame

1	1	1	1	1	1	N	1
Mode Boucle	Nb Iterations	Timer1	Timer2	Timer3	Nombre de lettres	Lettres du Message	'\n'
0=desactive 1=Active	1-255	centieme de seconde	centieme de secondes	centieme de secondes	0-255	249 char max	'\n'

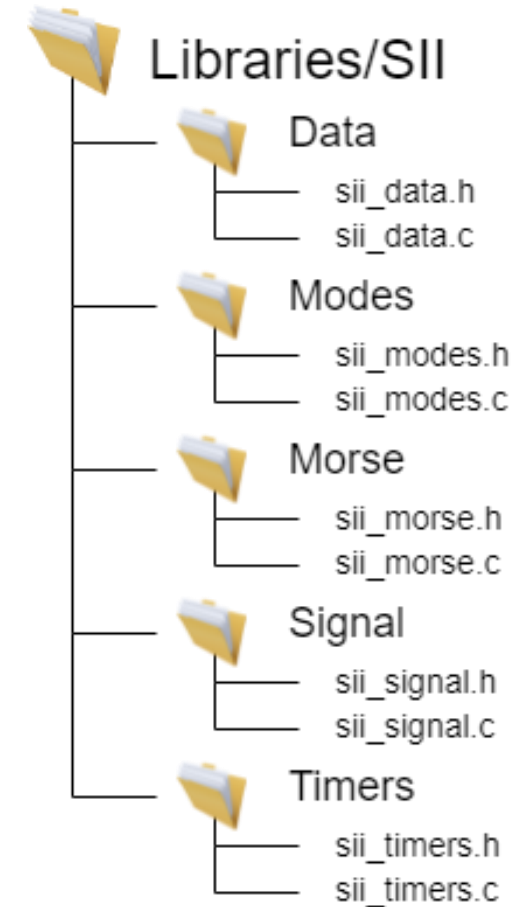
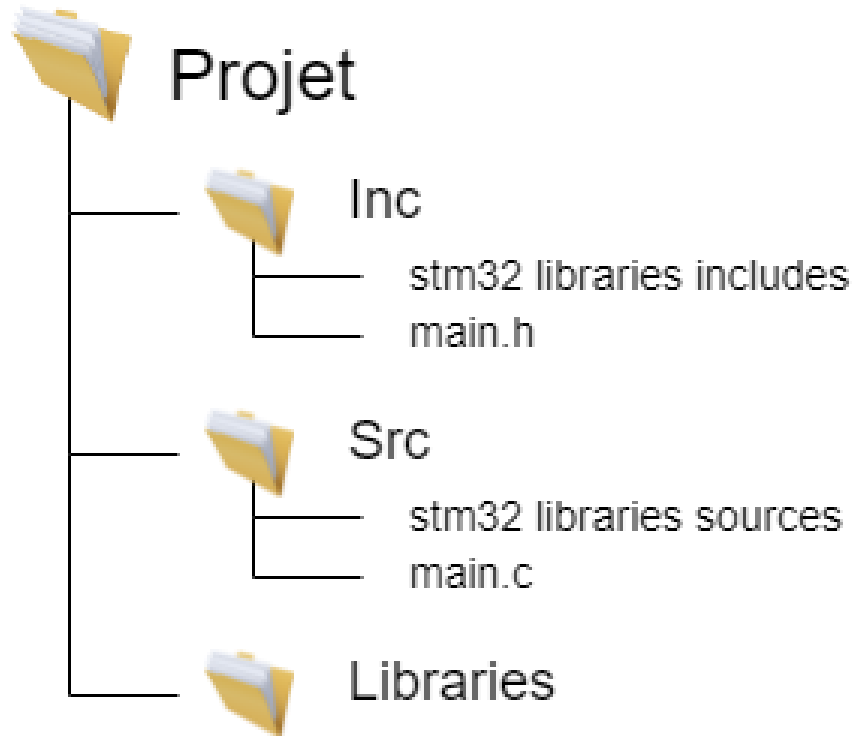
Communication



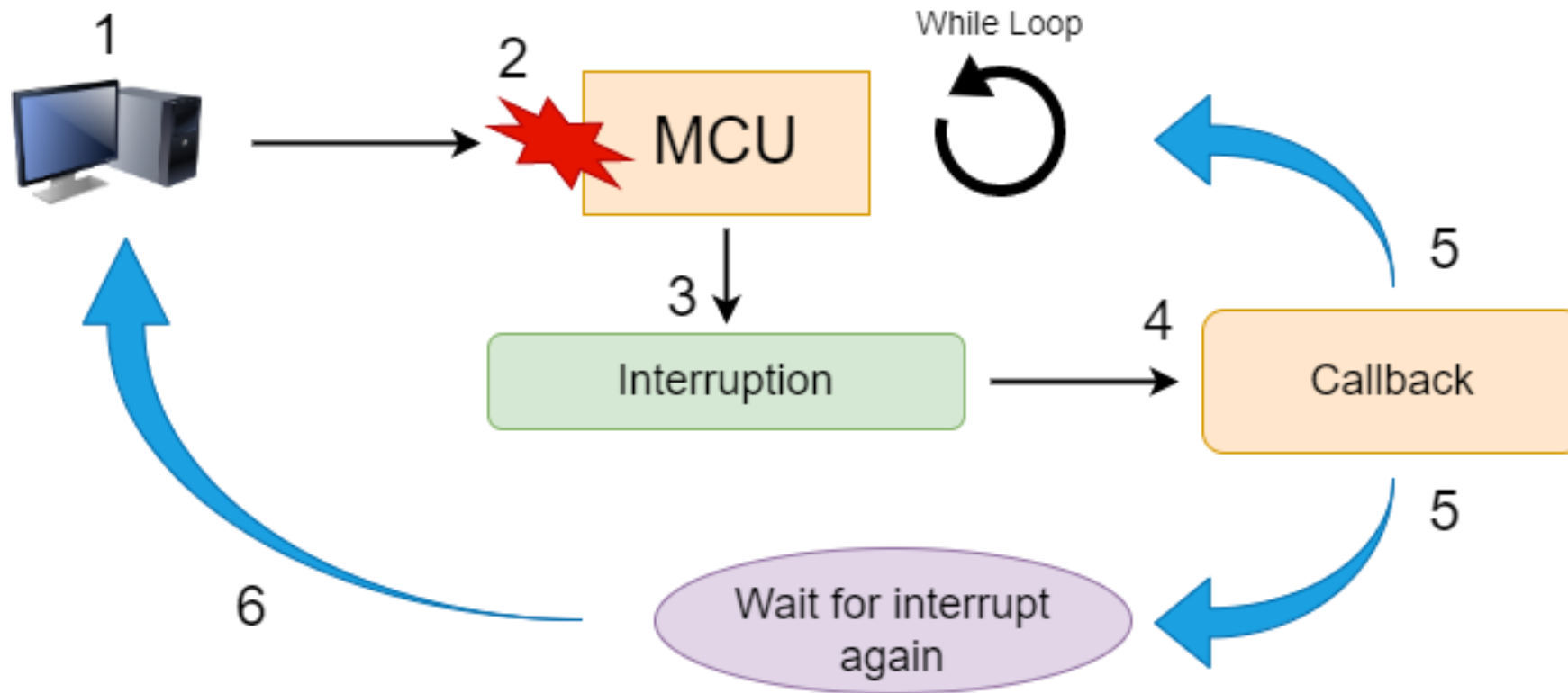
Architecture



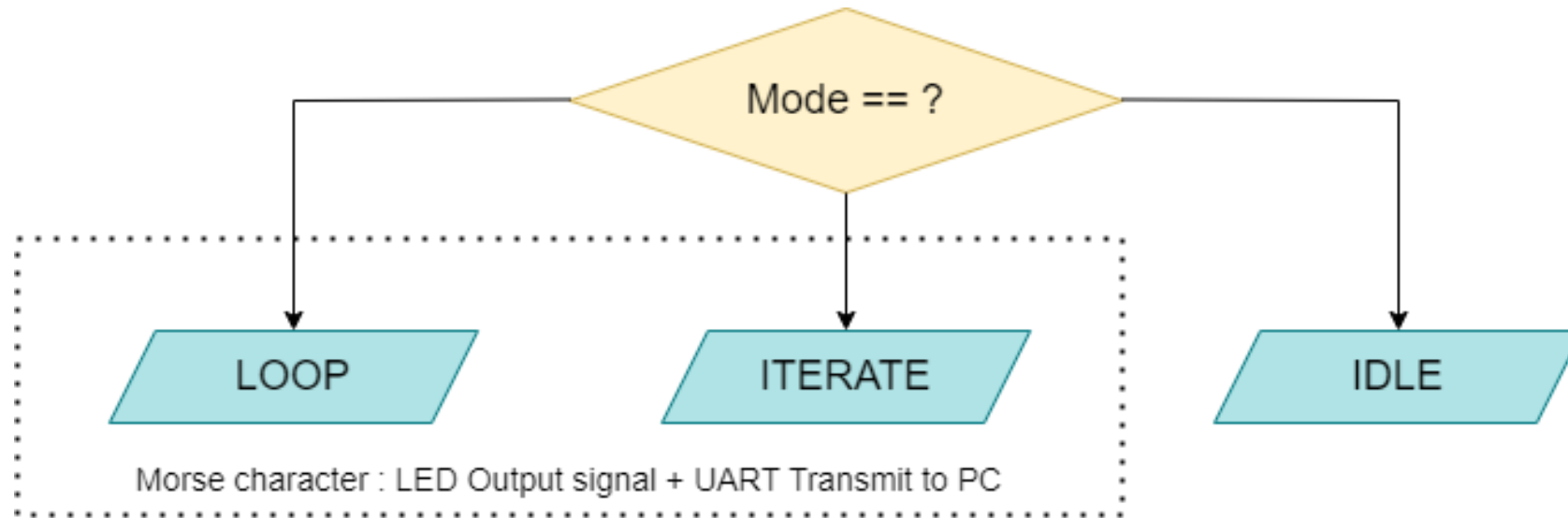
Arborescence



Réception de trame



Transmission du signal



Démonstration



Merci



Imagine the world
of tomorrow **with us**