

## Requête pour récupérer le temps de départ :

SELECT minuts, seconds FROM timer ORDER BY id DESC LIMIT 1

### Requête pour récupérer la séquence de courses :

SELECT firstRace, secondRace, thirdRace FROM races Sequence ORDER BY id DESC LIMIT 1

# Requête pour enregistrer le temps de départ :

INSERT INTO timer(minuts, seconds)
VALUES(minuts, seconds)

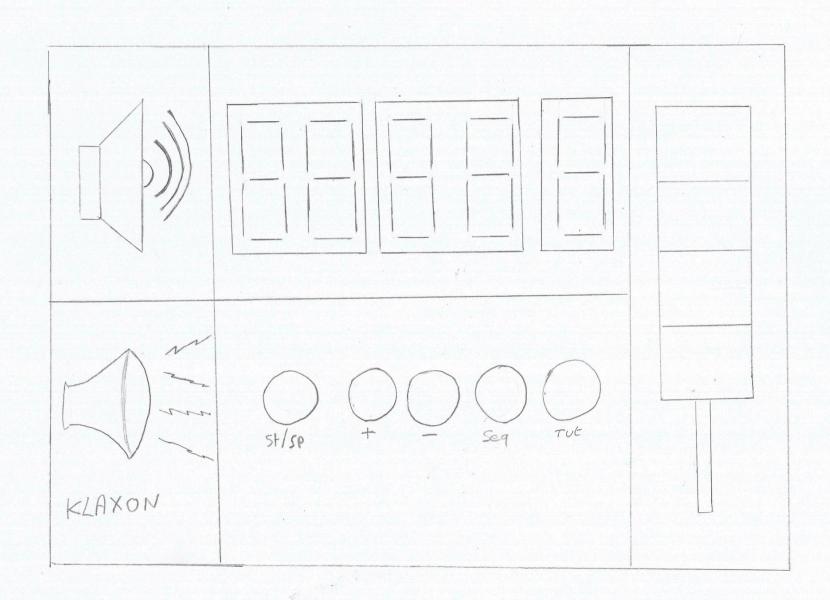
# Requête pour enregistrer la séquence de courses :

INSERT INTO racesSequence(firstRace, secondRace, thirdRace)
VALUES(firstRace, secondRace, thirdRace)

# Protocole de communication Bluetooth

De	Vers	Rôle	forme	Quand
Application	Raspberry	Demander le test des boutons	Un entier (1)	À l'appui sur le bouton « test des boutons » de l'application
Application	Raspberry	Demander le test de l'afficheur	Un entier (2)	À l'appui sur le bouton « test de l'afficheur » de l'application
Application	Raspberry	Demander le test des leds	Un entier (3)	À l'appui sur le bouton « test des leds » de l'application
Application	Raspberry	Demander le test des hauts parleurs	Un entier (4)	À l'appui sur le bouton « test des hauts-parleurs » de l'application
Application	Raspberry	Demander le test du klaxon	Un entier (5)	À l'appui sur le bouton « test du klaxon » de l'application
Raspberry	Application	Indiquer la fin du test	Un entier (0)	À la fin de chaque test

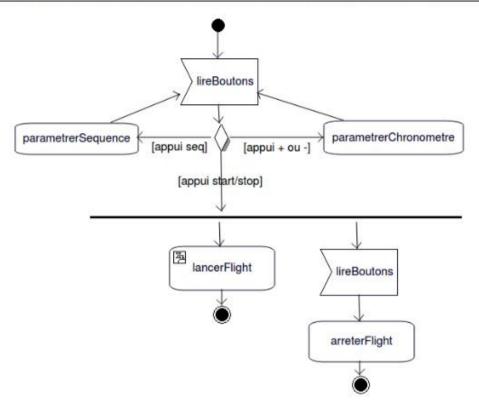
# IHM APPLICATION TEST DU BOÎTIER SALF



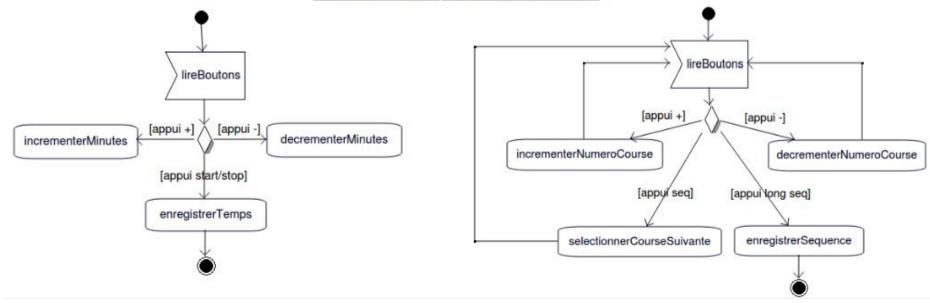
#### **DESCRIPTION DU CAS D'UTILISATION « LANCER »**

- Le système initialise le boîtier SALF
- 2) le comité de course appuie sur le bouton start/stop pour lancer le flight
- 3) le système détecte l'appui
- le système récupère dans sa base de données les informations relatives au flight à lancer (chronomètre de départ, ordre des courses)
- 5) le système démarre le lancement du flight (et l'arrête si le comité appuie de nouveau sur start/stop)
- lancement du chronomètre sur l'afficheur
- pour chaque course : affichage du numéro de course et envoi des différents signaux visuels
- changement du numéro de course

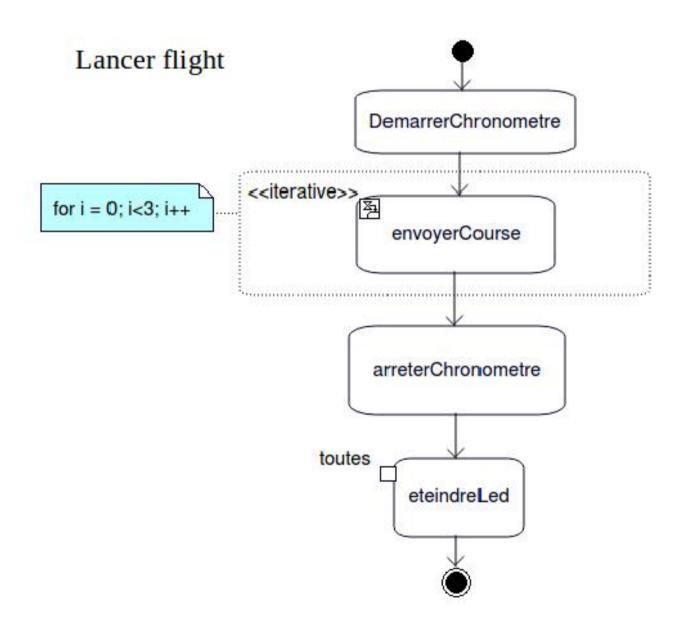
#### DIAGRAMME D'ACTIVITÉ DU CAS D'UTILISATION : LANCER



#### **DÉTAILS DES PARAMÉTRAGES**

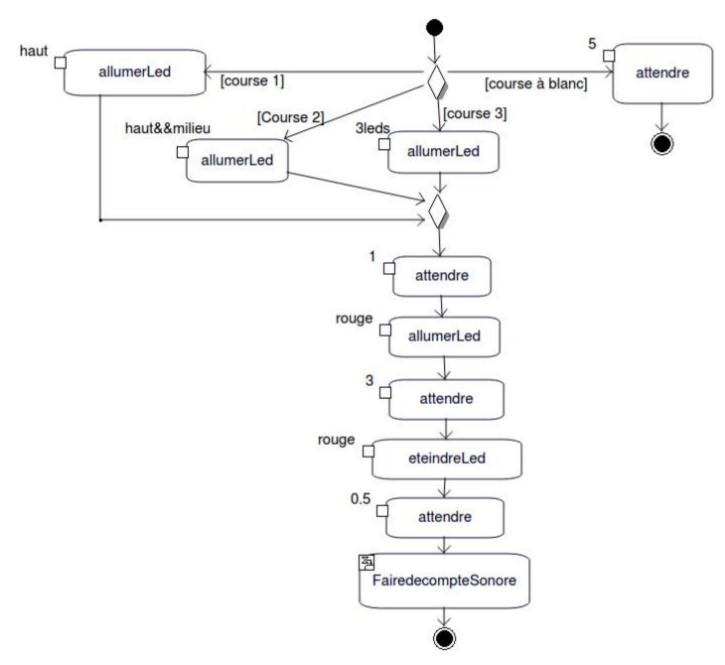


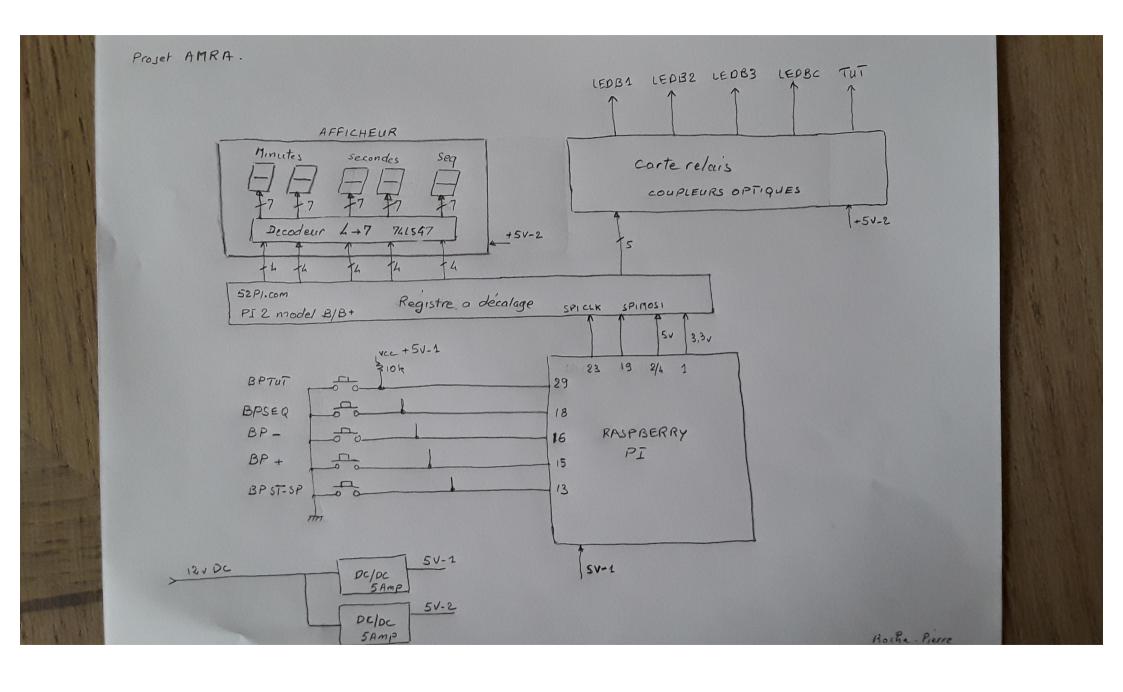
# **UC LANCER: DÉTAILS DU LANCEMENT**



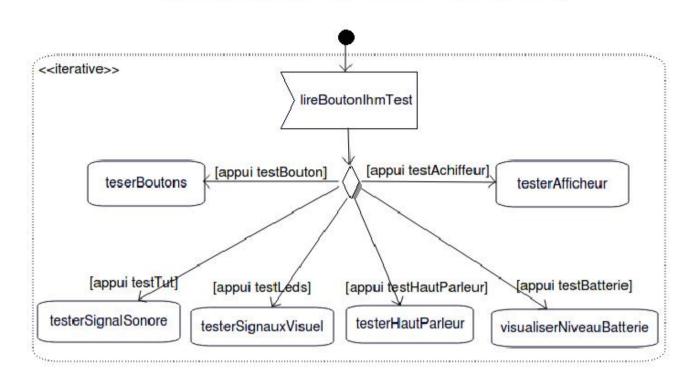
# <u>UC LANCER</u>: <u>DÉTAILS DU LANCEMENT</u>

#### Envoyer course

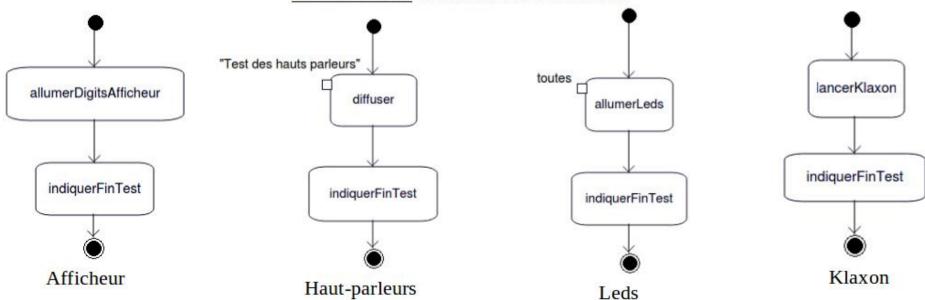




#### **DIAGRAMME D'ACTIVITÉ: UC TESTER:**



#### **UC TESTER: DÉTAILS DES TESTS**



# Speaker voiceLanguage: string + Speaker(in voiceLanguage : string = "en") + play(in textToPlay : string) : void + isPlaying(): bool speaker SalfBoxlhm + IO EXTENDER SIZE : int= 4 {readOnly} + COMPTEUR : long= 2 {readOnly} + SEQUENCE : long= 4 {readOnly} inverseDigits(in nb : uint8 t) : uint8 t inverseDigitsBack(in nb : uint8 t) : uint8 t buttonTest(in pin : ButtonGpio) : void + setCompteur(in minuts : uint8 t, in seconds : uint8 t) : void + setSequence(in sequence : uint8 t) : void + setLeds(in positionLed : LedColor, in tutEnd : bool = false) : void

keySetBox: key t

kevGetBox : kev t

etatLeds: int

buttonState: int

+ ioExtender : char

+ init() : void

spi: std::ofstream

ioExtenderFileId: int

ioExtenderFileGet : int

+ LEDS : long= 3 {readOnly}

initializeButtonsMap(): void

+ updateloExtender(): void

+ getloExtender(): void

+ getSeconds() : char

+ getSequence() : char

+ getMinuts() : char

+ getEtatLed(): int

+ doTut(): void

+ doSeq(): void

+ doMinus(): void

+ doStartStop(): void

+ doPlus(): void

+ clearloExtender(): void

+ getloExtenderFileGet(): int

```
Database
           zErrMsg: char
           sqlRequest: string
           filename: string

    Database(in filename : string = "Salf.db")

         + ~Database()
         + executeRequest(in sqlRequest : string, in isSelectRequest : bool = false, inout data : std::vector<std::vector<std::string> > = NULL) : void
         + prepareRequest(): int
                                                                    -database
                                                                                     -database
                                  Flight
                                                                                      nbAppuiSeq: int
  isActive : bool
                                                                                      symbolPosition: int
 stopNow: bool
                                                                                     currentRace: int
 currentRace : char
                                                                                     buttonCombinaison: int
 completeSequence : char
                                                                                     minuts: int
 minuts: int
                                                                                      seconds: int
 seconds: int
                                                                                      idFileButtons: int
+ Flight()
                                                                                      sequence : char
+ ~Flight()
                                                                                      selectedSymbol: char
+ initializeMapRaceLeds(): void
                                                                                      started: bool
+ start(): void
                                                                                      sequenceConfiguring : bool
+ stop(): void
                                                                                     timerConfiguring: bool
+ playSeconds(): void
                                                                                     onPress : bool
+ executeStart(inout context : void) : void
                                                                                     segModified: bool
                                                                             -flight -
+ executePlaySeconds(inout context : void) : void
                                                                                      <<:vector>> symbols : char
+ setSeconds(in seconds : int) : void
                                                                                      executeIncrementDecrement(inout context : void) : void
+ getIsActive(): bool
                                                                                      executeSetUpSequence(inout context : void) : void
+ setMinuts(in minuts : int) : void
                                                                                      executeStartStopFlightLaunch(inout context : void) : void
+ setCompleteSequence(in first : char, in second : char, in third : char) : void
                                                                                      executeTut(inout context : void) : void
+ setSequence(in sequence : char) : void
                                                                                      startStopFlightLaunch(): void
+ checkMinuts(in m : int) : void
                                                                                      increment(): void
+ doFirstRace(in m : int) : void
                                                                                      setUpSequence(): void
+ doSecondRace(in m : int) : void
                                                                                     tut(): void
+ doThirdRace(in m : int) : void
                                                                                      resetSymbols(): void
+ getSequenceFromDb(in sequence : string) : void
                                                                                      seq(): void
+ getTimerFromDb(): int
                                                                                      plusMinus(): void
                                                                                     startStop(): void
                                                                                     clearTimer(): void
                                                                                     manageButtons(): void
                                                                                      getAvailableSymbol(): void
```

ButtonController

checkSequenceConfiguration(): bool + ButtonController(in idFileButtons : int)

+ ~ButtonController() + getButtonsState(): void