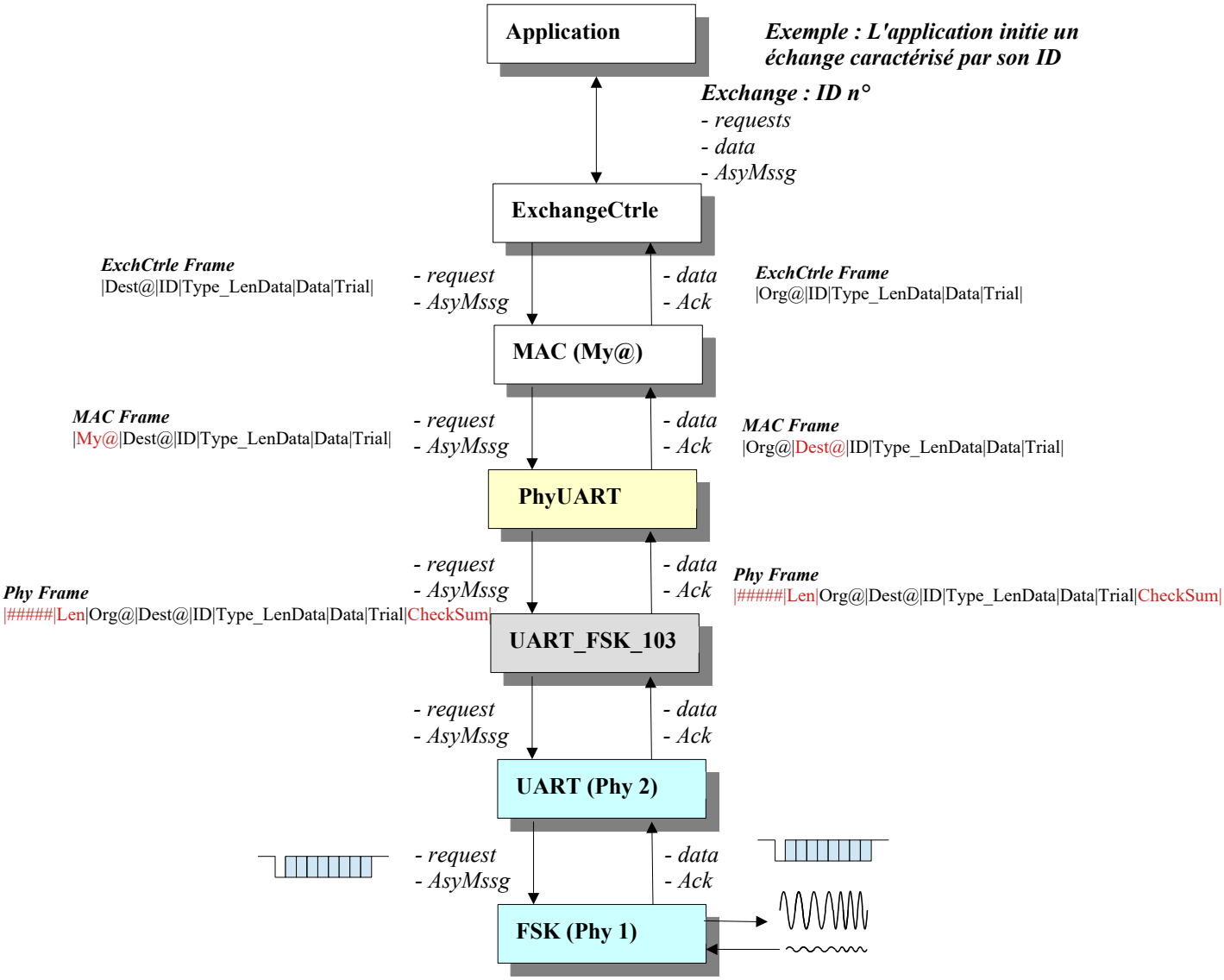
# User Guide
# FSK_COM_STACK

## Table des matières

1.Architecture de la pile de communication....1
2.Couches physique....2
2.1.Vue d'ensemble....2
2.2.Description des fonctions....3
2.3.Le flux de données....6

# 1. Architecture de la pile de communication

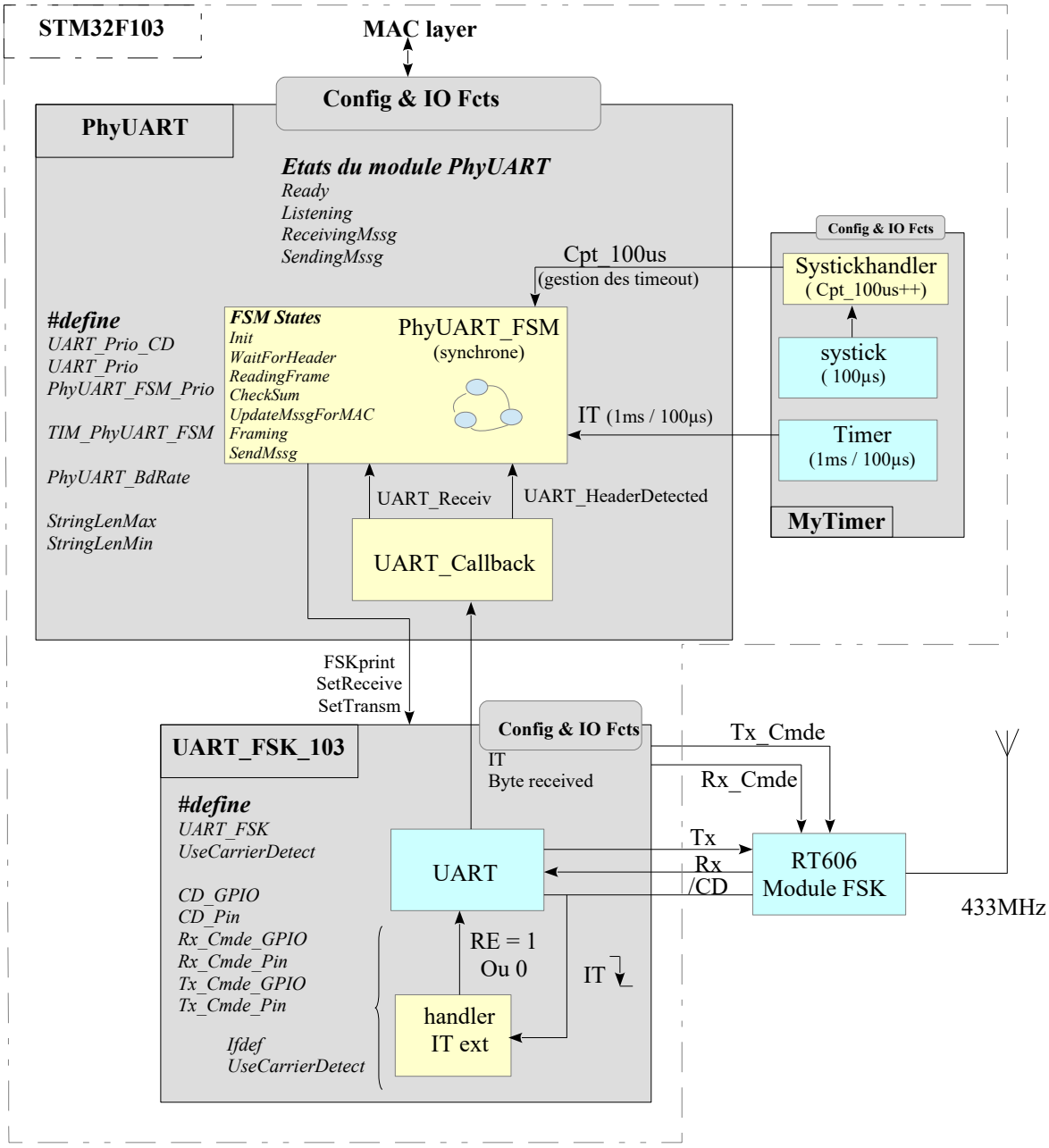


# 2. Couches physique

## 2.1. Vue d'ensemble

La couche physique regroupe :

- FSK (Phy 1) , le module FSK RT606 = matériel hertzien
- UART (Phy 2), l'UART du µC = matériel
- UART_FSK_103 , module driver UART hardware dépendant = logiciel bas niveau
- Phy_UART, module de plus haut niveau de la couche physique = logiciel



## 2.2. Description des fonctions

### *2.2.1. UART_FSK_103.c/.h*

Le module dépend du hardware.

Deux interruptions sont utilisées :

- réception octet,
- détection de porteuse (optionnelle). Si le *UseCarrierDetect* est défini, l'UART est bloqué en réception en absence de porteuse. Cela permet d'éviter d'avoir des interruptions UART qui arrivent constamment en présence de bruit.

***Les broches :***

CD, RxCmde et Tx Cmde sont définies dans le .h.

Tx et Rx sont définies automatiquement, et configurées dans le .c en fonction de l'USART choisie (USART1 , 2 ou 3).

***Les fonctions de configurations***

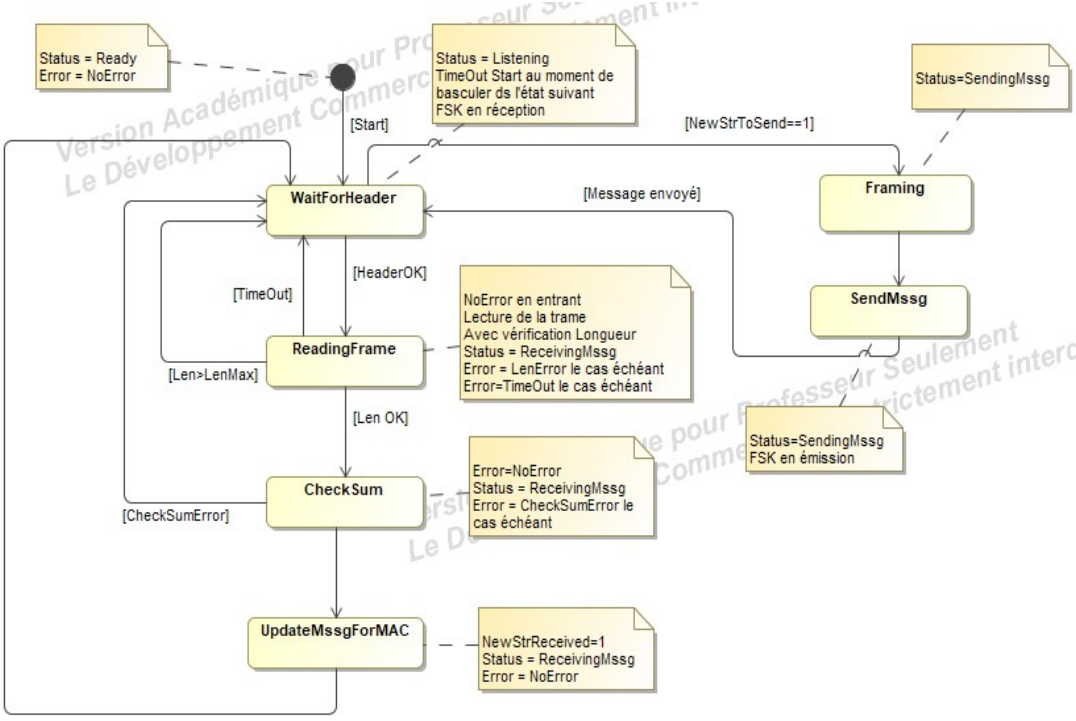
```
/*****************************************************************************************************
 Configure l'UART spécifiée en #define, avec le débit indiqué (en Baud), la priorité utilsée pour
 la détection de porteuse, la priorité d'interrution de réception, et enfin, on précise la fonction
 callback associée à l'interruption.
 Exemple :
 USART_FSK_Init(PhyUART_BdRate,UART_Prio_CD,UART_Prio,UART_Callback);
*****************************************************************************************************/
void USART_FSK_Init(int Baud_Rate_bits_par_Sec, char Prio_USART_CD, char Prio_USART, void (*IT_function) (void));
```

***Les fonctions IO***

```
/*****************************************************************************************************
 Renvoie le dernier caractère lu par l'UART
 Exemple :
 if (USART_FSK_GetByte()==HeaderCar)
*****************************************************************************************************/
char USART_FSK_GetByte(void);

/*****************************************************************************************************
 Envoie caractère par caractère la chaîne dont l'adresse est spécifiée. Le nombre d'octet à émettre
 est spécifiée par le paramètre Len.
 Exemple :
 USART_FSK_Print("1234",4);
*****************************************************************************************************/
void USART_FSK_Print(char* Mssg, int Len);

/*****************************************************************************************************
 Fixe le module FSK en position de réception
*****************************************************************************************************/
void USART_FSK_SetReceiveAntenna(void);

/*****************************************************************************************************
 Fixe le module FSK en position d'émission
*****************************************************************************************************/
void USART_FSK_SetTransmAntenna(void);
```

### *2.2.2. PhyUART.c/.h*

La FSM associée est synchrone. Elle est basée sur timer dont la périodicité des interruptions peut prendre deux valeurs, selon le mode dans lequel se trouve la FSM (cf ci-dessous) :




Dans l'état *WaitForHeader* (majorité du temps), l'échantillonnage de la FSM est de 1ms tant que le *Header* n'est pas détecté. Dès que celui-ci est détecté (dans le callback UART), le timing descend à 100µs ainsi que pour tous les autres états. Le retour à l'état *WaitForHeader* se fait toujours après avoir modifié le timing à 1ms.

***Les #define***

*#define StringLenMax 30*

*#define StringLenMin 7*

Ce sont les longueurs qui définissent la chaîne de réception et la celle d'émission.

***Les fonctions***

```
/*****************************************************************************************************
 Configure le module PhyUART (Timer, UART, état initial de la FSM...
*****************************************************************************************************/
void PhyUART_Init(void);

/*****************************************************************************************************
 Lance la FSM en validant la transition vers l'état WaitForHeader
*****************************************************************************************************/
void PhyUART_StartFSM(void);


char PhyUART_IsNewMssg(void);
PhyUART_StatusType PhyUART_Get_Status(void);
PhyUART_ErrorType PhyUART_Get_Error(void);

/*****************************************************************************************************
 Permet de recopier la donnée validée par la couche PhyUART.
 Len est la longueur Maximale pour cette recopie.
 ATTENTION ! la chaîne samplée par l'UART est entièrement recopiée à partir de l'adresse pointée.
 Un test est donc fait pour savoir si la longueur du tableau recevant est suffisant.
 La fonction répond -1 en cas d'erreur sur ce test ou 0 si tout est OK
*****************************************************************************************************/
int PhyUART_GetNewMssg (char * AdrString, int Len);
int PhyUART_SendNewMssg (char * AdrString, int Len);
```

# 2.3. Le flux de données