ARD4	Structurer un programme	Durée : 3h.

Matériel nécessaire : Un kit ESP32 – Un module SQ200

1 - Introduction

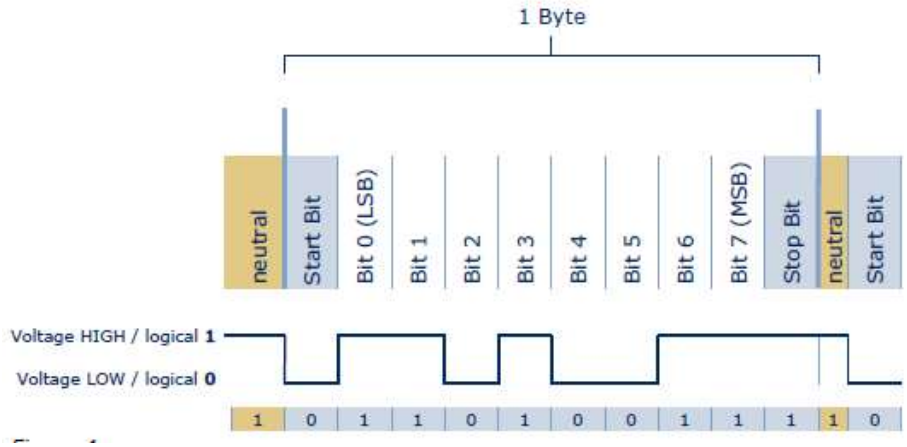
Pour structurer un programme, il faut :
- Décomposer en fonctions logicielles élémentaires.
- Enregistrer les fonctions dans une bibliothèque pour qu'elles puissent être réutiliser dans différents projets.
- Ecrire l'algorithme pour les fonctions un peu complexe, comme le programme principal. L'écrire en commentaire dans le programme.

Pour illustrer cette façon de faire, nous allons prendre l'exemple du dialogue entre le µP et un module Enocean. Ce dialogue est assuré par une liaison UART, utilisant l'UART2 de l'ESP32.

Le format est donné ci-dessous, le débit est de 57600 bauds.

1.4 UART framing

The UART of the EnOcean module has the framing: 8 data bits, no parity bit, one start bit (logical 0), one stop bit (logical 1). The line idle (≙neutral) is logical 1 (standard).

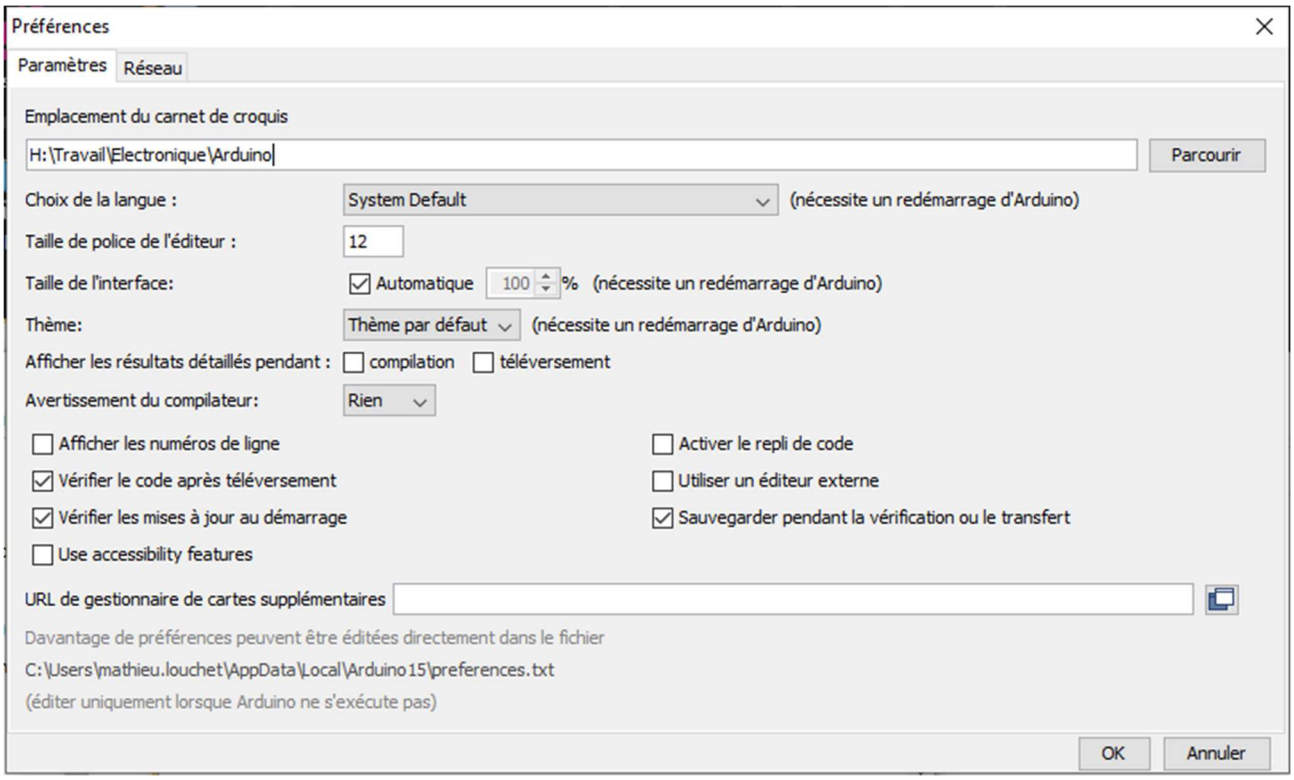


2 – Organiser les répertoires de travail

⇨ Dans le répertoire de travail sur kwartz (H:), créer un répertoire **Arduino** (ou Arduino_Esp32) et dans ce répertoire créer 2 sous répertoires **libraries** et **module_enocean.**

Dans le répertoire libraries seront placées les différentes bibliothèques personnelles et dans le répertoire module_enocean seront placés les différents exemples développés pour tester le module enocean.

⇨ Sous le logiciel Arduino, déclarer le répertoire de travail (se trouvant sur H:) dans les préférences (Fichiers/Préférences)



⇨ Dans le répertoire libraries (sur H), créer un sous répertoire Enocean_Cmd et y placer les fichiers, donnés en annexe, Enocean_Cmd.cpp et Enocean.h.

⇨ Sous le logiciel Arduino, créer un nouveau fichier et y copier le programme ci-dessous.

```
// le port série utilisé est déclaré dans Enocean_Cmd.h
#include <Enocean_Cmd.h>

void setup() {
  //Initialiser la liaison série Enocean
  enocean_setup();
}

//Toutes les 2 secondes on transmet la trame de mise en veille
void loop() {
  //Mettre en veille le module Enocean pendant 1 seconde
  enocean_veille(100);
  //Attendre 2 secondes
  delay(2000);

}
```

⇨ L'enregistrer dans le répertoire module_enocean, sous le nom E1_veille_enocean. (E1 pour exemple 1).

3 – Explication des fonctions logicielles fournies et test de la transmission.

⇨ Ouvrir les fichiers Enocean_Cmd.cpp et Enocean_Cmd.h avec Notepad++.

Dans le fichier Enocean_Cmd.h (fichier d'entête):

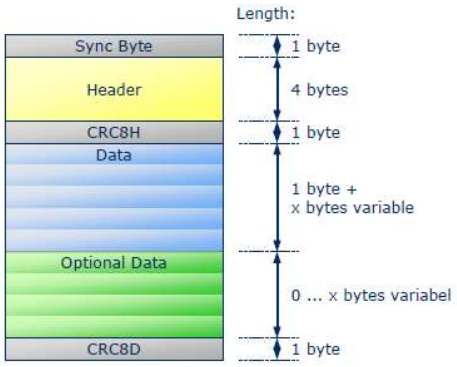
```
#define Serial_Enocean Serial2

unsigned char calculCRC(unsigned char* data, unsigned int DataSize);
void enocean_setup(void);
bool enocean_veille(unsigned long dixmilli);
```

• L'étiquette Serial_Enocean est affectée à Serial2. Il faudra ici relier le module à l'UART2.
• La fonction calculCRC permet de calculer le CRC (code de contrôle d'intégrité de la trame). Si le module détecte un CRC non correct, la trame est ignorée.
En entrée : Elle reçoit l'adresse du premier octet de la trame et le nombre d'octet à traiter.
En sortie : Elle retourne la valeur du CRC
• La fonction enocean_setup permet de paramétrer la liaison série entre ESP32 et module ENOCEAN.
• La fonction enocean_veille permet de mettre en veille le module ENOCEAN:
En entrée : la durée de la mise en veille en centième de seconde.
En sortie : retourne true ou false si la fonction s'est bien déroulée.

Le format de transmission Enocean est le suivant :

Format général de transmission



Mise en veille du module

2.5.3 Code 01: CO_WR_SLEEP

Function: Order to enter the energy saving mode.

Group	Offset	Size	Field	Value hex	Description
-	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0005	5 bytes
	3	1	Optional Length	0x00	0 byte
	4	1	Packet Type	0x05	COMMON_COMMAND = 5
-	5	1	CRC8H	0xnn	
Data	6	1	COMMAND Code	0x01	CO_WR_SLEEP = 1
	7	4	Deep sleep period	0x00nnnnnn	Period in 10 ms units 00000000 = default max. value = max. data range 00 FF FF FF (~ 46h); After waking up, the module generate an internal hardware reset
-	11	1	CRC8D	0xnn	

Table 22

In this case, the following **RESPONSE** message gives only the return codes:
00 RET_OK
02 RET_NOT_SUPPORTED
Since no additional data are included which require description the standard RESPONSE structure is detailed in chapter 2.2.3

CRC8H : CRC sur les 4 octets d'entête (Header)
CRC8D : CRC sur les données.

Dans le fichier Enocean_Cmd.cpp, les sources des fonctions logicielles.
• Le tableau unsigned char u8CRC8Table[256] est utilisé pour le calcul du CRC
• Le tableau VEILLE est initialisé avec les valeurs connues à transmettre au module.

```
unsigned char VEILLE[12]={0x55,0x00,0x05,0x00,0x05,0x00,0x01,0x00,0x00,0x00,0x00,0x00};
```

• La fonction enocean_veille place les 3 octets de la durée de veille, calcule les CRC, transmet la trame et ici, pour simplifier, retourne true sans tester l'accusé de réception (on se laisse la possibilité d'améliorer la fonction plus tard, en testant le CRC).

```
bool enocean_veille(unsigned long dixmilli){
    //METTRE LES 8 BITS DE POIDS FAIBLE DANS VEILLE[10]
    VEILLE[10]=dixmilli;
    //DECALER DE 8 BITS VERS LA DROITE POUR RECUPERE LES 8 BITS SUIVANTS
    dixmilli=dixmilli>>8;
    //METTRE LES 8 BITS DE POIDS FAIBLE DANS VEILLE[9]
    VEILLE[9]=dixmilli;
    //DECALER DE 8 BITS VERS LA DROITE POUR RECUPERE LES 8 BITS SUIVANTS
    dixmilli=dixmilli>>8;
    //METTRE LES 8 BITS DE POIDS FAIBLE DANS VEILLE[8]
    VEILLE[8]=dixmilli;
    //CALCULER ET MEMORISER LE CRC8H
    VEILLE[5]=calculCRC(&VEILLE[1],4);
    //CALCULER ET MEMORISER LE CRC8D
    VEILLE[11]=calculCRC(&VEILLE[6],5);
    //TRANSMETTRE LA TRAME AU MODULE ENOCEAN
    Serial_Enocean.write(VEILLE,12);
    //RETOURNER TRUE SANS TESTER L'ACCUSE DE RECEPTION
    return true;
}
```

⇨ Compiler et transférer le programme dans un module ESP32 et tester la transmission avec un module SQ200 (faire vérifier le câblage avant toute chose).

⇨ Relever les données transmises et identifier les différentes valeurs de la trame.

4 – Ajouter la fonction RESET du module

La commande reset du module correspond au descriptif suivant :

2.5.4 Code 02: CO_WR_RESET

Function: Order to reset the device.

Group	Offset	Size	Field	Value hex	Description
-	0	1	Sync. Byte	0x55	
Header	1	2	Data Length	0x0001	1 byte
	3	1	Optional Length	0x00	0 byte
	4	1	Packet Type	0x05	COMMON_COMMAND = 5
-	5	1	CRC8H	0xnn	
Data	6	1	COMMAND Code	0x02	CO_WR_RESET = 2
-	7	1	CRC8D	0xnn	

Table 23

In this case, the following **RESPONSE** message gives only the return codes:
00 RET_OK
01 RET_ERROR
02 RET_NOT_SUPPORTED
Since no additional data are included which require description the standard RESPONSE structure is detailed in chapter 2.2.3

⇨ Ajouter la fonction **bool enocean_reset(void)** aux fichiers Enocean_Cmd.cpp et Enocean_Cmd.h et tester la transmission comme la fonction veille précédente.

Remarque : les réponses du module ne sont pas gérées ici, par soucis de simplicité.