IUT D'ANNECY
Département GEII
Parcours EME

Année 2022-2023
R3.13
TP n°2

Réseaux Spécialisés EME : EnOcean
Communication sans fil pour le bâtiment

Objectifs visés :

- ✓ *Le réseau sans fil **EnOcean***
- ✓ *Télégramme de type **RPM***

1. Le technologie EnOcean (www.enocean.com)

La technologie ***EnOcean*** est une ***technologie sans fil bidirectionnelle*** destinée aux bâtiments et aux ensembles industriels. Elle convertit l'énergie de notre environnement (*mouvement linéaire, pression, lumière, température, rotation et vibration*) pour l'employer à transmettre des informations provenant de capteurs. Elle est donc ***exempte d'entretien*** pour les capteurs qui répondent à cette norme puisque ces derniers sont ***autoalimentés***.

figure 1 : Principe de la technologie EnOcean

Afin de permettre la communication entre les capteurs et les différents systèmes de contrôle et/ou actionneurs, la norme ***EnOcean*** prévoie dans l'échange de données de mentionner les caractéristiques de l'émetteur. C'est la notion de profil (***E****nOcean* ***E****quipment* ***P****rofile* (*EEP : ORG FUNC TYPE*)) et il est indépendant du constructeur. Par ailleurs, cette norme offre un mode d'apprentissage afin d'établir les relations fonctionnelles entre les capteurs et les actionneurs.

1.1 Caractéristiques Radio Fréquence :

- ✓ *Fréquence d'émission : 868.3 Mhz EU*
  315 Mhz USA & Canada
- ✓ *Modulation d'amplitude ASK (<10mW)*
- ✓ *Vitesse de transmission : 125 kbps / 280 kHz*
- ✓ *Portée en extérieur/intérieur : 300 m / 40 m*

1.2 Télégramme :

Un message est ***une suite de n octets*** comprenant deux ***octets de synchronisation***, un ***entête*** (***header***) et des données se terminant par un octet de ***status*** puis de ***contrôle (cheksum)***.

Bit 7 **Bit 0**	Désignation
SYNC_BYTE1 (A5 Hex) **SYNC_BYTE0 (5A Hex)**	Synchronization bytes (0xA5 0x5A)
H_SEQ **LENGTH**	**H_SEQ** (3 bits) : Header identification: 0 : **R**eceive **R**adio **T**elegram (***RRT***) … 3 : **T**ransmit **R**adio **T**elegram (***TRT***) 4 : **R**eceive **M**essage **T**elegram (***RMT***) 5 : **T**ransmit **C**ommand **T**elegram (***TCT***) **LENGTH** (5 bits) : Number of octets following the header octet (always 11)
ORG	Type of telegram (*RPS, 1BS, 4BS, …*)
DATA_BYTE3 **DATA_BYTE2** **DATA_BYTE1** **DATA_BYTE0**	Data bytes 0..3 *(*)*
ID_BYTE3 **ID_BYTE2** **ID_BYTE1** **ID_BYTE0**	32-bit transmitter **ID** (*)
STATUS	Status field (*)
CHECKSUM	Checksum (Least Significant Byte from addition of all octets except synchronization bytes and checksum)

(*) La signification de ces octets est fonction du type de télégramme. Par exemple, pour un télégramme de type ***RPS*** *(ORG=0x05), l'octet de* **status** *s'interprète de la façon suivante :*

STATUS FIELD

7	5	4	0
Reserved	**T21**	**NU**	**RP_COUNTER**

- ✓ Reserved (2 bits) : For future use, default value 0
- ✓ T21 (1 bit) : T21=0 => telegram of type 1, T21=1 => telegram of type 2
- ✓ NU (1 bit) : NU=1 => N-message, NU=0 => U-message.
- ✓ RP_COUNTER (4 bits) : Repeater count

2. Mise en œuvre sur la cible Wago

Pour pouvoir disposer du réseau ***EnOcean*** sur la structure de commande ***Wago*** au profil ***Ethernet***, celle-ci doit être équipée d'une liaison ***série multi-points RS422/RS485*** programmable (module ***750-653***), ainsi que d'une passerelle ***RS485/EnOcean*** (***STC65-RS485***).

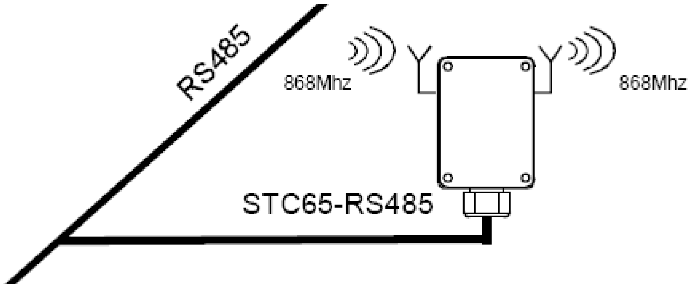


figure 2 : Passerelle EnOcean-RS485

D'un point de vue logiciel, on dispose de la bibliothèque «***Enocean_05.lib***» du constructeur ***Wago***. Elle permet :

- ✓ de récupérer sur l'interface ***RS485*** les télégrammes grâce au bloc fonctionnel ***FbThermokonSRC65_RS485_EVC***,

- ✓ d’offrir plusieurs blocs fonctionnels qui assurent ***l’interprétation*** des télégrammes reçus en fonction de leur ***type*** et des ***éléments qui les émettent***.

2.1 ***Réception des télégrammes*** *:*

Pour recevoir les télégrammes de la passerelle ***STC65-RS485*** connectée à la liaison série ***multi-points RS485*** du contrôleur, on dispose du bloc fonctionnel ***FbThermokonSRC65_RS485_EVC*** dont l’interface est donnée par :

```
    FUNCTION_BLOCK FbThermokonSRC65_RS485_EVC
        VAR_INPUT
            bCOM_PORT_NR: BYTE;     (* N° of the serial interface used :
                                        1 ⬄ Internal service port,
                                        2 ⬄ 1st connected serial module,
                                        3 ⬄ 2nd connected serial module *)
        END_VAR
        VAR_OUTPUT
            typEnocean: typEnocean;  (* Output data of the received radio telegram *)
            bError: BYTE;            (*error code :
                                        0x00 = no error,
                                        0x01 = illegal COM Port,
                                        0x0C = initialization error,
                                        0x0D = problem with hardware handshake,
                                        0x81 = faulty telegram (CRC-CHECSUM *)
        END_VAR
```

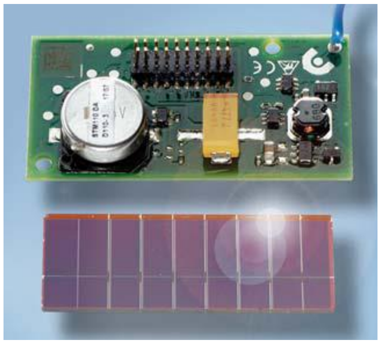
Le type ***typEnocean*** *est le reflet des informations constituant une trame* ***EnOcean*** *et est défini par :*

```
TYPE typEnocean :
    STRUCT
        MSG_Type     :     BYTE;      (* ORG: 5 -> RPS; 6 ->1BS; 7 -> 4BS; 8 -> HRC *)
        ID           :     DWORD;     (* Transmitter ID *)
        Data_Byte_0  :     BYTE;      (* DATA_BYTE0 *)
        Data_Byte_1  :     BYTE;      (* DATA_BYTE1 *)
        Data_Byte_2  :     BYTE;      (* DATA_BYTE2 *)
        Data_Byte_3  :     BYTE;      (* DATA_BYTE3 *)
        T21          :     BOOL;      (* PTM switch module of type 1 / 2 *)
        NU           :     BOOL;      (* U-message / N-message *)
        RP_Counter   :     BYTE;      (* Repeater level *)
        ToggelBit    :     BOOL;      (* Flag for telegram updates *)
        Status       :     BYTE;       (* Status byte *)
        xBusy        :     BOOL;      (* Sendeauftrag aktiv  *)
        xStartSend   :     BOOL;      (* Flag for send activation *)
        bAddress     :     BYTE;      (* Slave Address for multiple receiver *)
    END_STRUCT
END_TYPE
```

pascal.aygalinc@univ-smb.fr

*Etablir la classe **CEnoceanRcvRS485** qui permet une utilisation avancée de ce bloc fonctionnel **FbThermokonSRC65_RS485_EVC**. Un objet de cette classe permettra donc de recevoir les données du réseau **EnOcean** sur le **port de communication n°2** du contrôleur **API** dans notre cas.*

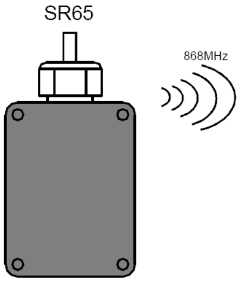
3. *Capteur de Température Extérieure : Structure de données et Utilisation*



Parmi les énergies de notre environnement qu'utilise la technologie ***EnOcean*** pour transmettre des informations provenant de capteurs, on dénombre le ***solaire***. Le module ***STM 110*** (***S****ensor* ***T****ransmitter* ***M****odule*) en est un exemple car il est doté d'une cellule photovoltaïque et d'un système de conservation d'énergie qui lui permet d'être ***autonome*** (*sans entretien*) et de fonctionner une ***cinquantaine d'heures dans l'obscurité totale***.
Parmi les capteurs qui intègrent ce système, on trouve des capteurs de luminosité, de présence, de température intérieure ou extérieure comme le ***SR65*** du constructeur ***Thermokon*** (*www.thermokon.de*).

Le capteur de température ***SR65*** est au profil ***A5/07-02-14 (TYPE=16#14)***. Sa configuration matérielle est établie à l'aide de ***2 cavaliers*** qui permettent de définir la périodicité des échanges (*radio telegram*) dont les contenus sont :

ORG 7 decimal Always (EnOcean module type “4BS”)
Data_byte1 Temperature -20...60°C, linear n=255...0
Data_byte0 Bit D3 Learn Button (0=Button pressed)
ID_Byte3 device identifier (Byte3)
ID_Byte2 device identifier (Byte2)
ID_Byte1 device identifier (Byte1)
ID_Byte0 device identifier (Byte0)



identifiant : ***16#000589F9***

Au niveau logiciel, on dispose du bloc fonctionnel ***FbA502xx_TemperatureSensor*** de la bibliothèque « ***Enocean_05.lib*** » qui permet d'en établir une gestion aisée. Il suffit de lui fournir la trame reçue, d'indiquer l'identifiant de son éventuel expéditeur avec son type pour savoir si c'est dernier qui l'a envoyé et d'en connaitre sa mesure. Son interface est donnée par :

```
FUNCTION_BLOCK FbA502xx_TemperatureSensor
        VAR_INPUT
                typEnocean : typEnocean;    (* the received radio telegram *)
                bTYPE : BYTE;               (*Type of device (TYPE) *)
                dwID : DWORD;               (* transmitter ID of the sensor *)
                tTimeOut : TIME;            (* Maximum interval between two telegrams,
                                            Preset value = T#60m (T#0s ⇔ desactivation) *)
        END_VAR
        VAR_OUTPUT
                rTemperature : REAL ;       (* Temperature measured by the sensor [°C] *)
                xError : BOOL;               (* No new telegram within timeout period *)
        END_VAR
```

*Etablir la structure de données **CSr65** qui permet une utilisation avancée de ce bloc fonctionnel **FbA502xx_TemperatureSensor** afin de récupérer la température émise par un capteur **SR65** de la salle.*

4. *Partie TP*

Ce ***tp*** aura pour objectif la mise en œuvre du réseau ***EnOcean*** afin de déterminer la température de la salle ***C264*** à partir d'un capteur ***SR65***.

4.1 Copier sous votre répertoire associé à ce module, le ***squelette de l'application*** *qui se trouve sous «U:\Documents\DUT\GEII\ModulesS3\RLI3.13\Squelettes\EnOcean ». Pour les* ***salles virtuelles 1*** *et* ***2****, prendre le fichier* ***EnOcean_750_881_sql.pro****, alors que pour les* ***salles virtuelles 3*** *et* ***4****, c'est le fichier* ***EnOcean_750_841_sql.pro****.*

4.2 Onglet ***Type*** *: déclarer les « 2 classes » conçues précédemment (****CEnoceanRcvRS485*** *et* ***CSr65****).*

4.3 Onglet ***Module*** *: compléter le programme* ***EnOcean***

4.3.1 Tout d'abord, initialiser ***partiellement*** *les 2 objets locaux déclarés :*

- ✓ ***enoceanRcvRS485*** *en fonction de la configuration matérielle de* ***l'API*** *(****port de communication n°2****)*

- ✓ ***sr65C264*** *en fonction des caractéristiques du capteur* ***SR65*** *de la salle* ***C264*** *(identifiant :* ***16#000589F9****, type=****16#14****)*

4.3.2 Au niveau du traitement, sa structure devra répondre à l'algorithme suivant :

- ✓ *Réception des trames* ***EnOcean*** *à l'aide de l'objet* ***enoceanRcvRS485****,*

- ✓ ***En cas de réception correcte d'une trame*** *:*
  - ○ *Analyser celle-ci via l'objet* ***sr65C264*** *pour connaitre la température de la salle* ***C264****,*
  - ○ *Afficher celle-ci à l'aide du* ***bargraph*** *en respectant les cahiers des charges suivants :*

« On considèrera que le pas de quantification est de ***3 degrés****, et que l'origine de l'affichage est de* ***0 °C****. Il en découle que la* ***led*** *la plus basse du* ***bargraph*** *(****qxBarGraph0****) sera allumée pour une température mesurée comprise entre* ***[0°C, 3°C [****. Dans le cas où la température mesurée n'est pas comprise dans la plage de visualisation (le* ***bargraph*** *ne disposant que de* ***8 leds****), le* ***bargraph*** *devra rester éteint (*). On proposera les 3 modes d'affichage suivants :*

- ○ *Affichage de type point : seule la led qui admet dans son intervalle la température mesurée doit être allumée, les autres resteront donc éteintes.*
- ○ *Affichage de type jauge : la led qui admet dans son intervalle la température mesurée ainsi que celles situées en dessous devront être allumées, les autres resteront donc éteintes.*
- ○ *(*) Reprendre l'affichage de type jauge mais lorsque la température mesurée n'est pas comprise dans la plage de visualisation, le bargraph aura une led sur 2 allumées.*

4.4 On souhaite réaliser une animation d'éclairage qui permettra de visualiser la température ambiante à l'aide d'une couleur (bleu ⇔ froid, rouge ⇔ chaud). Pour cela, on dispose des trois ***ballasts électroniques DALI*** *(ecg) commandant respectivement les composantes* ***Rouge****,* ***Vert*** *et* ***Bleu (ecgRouge, ecgVert, ecgBleu)***. Le tableau suivant donne les niveaux de lumière ***(en %*** *⇔ nombre compris entre 0 à 100)* pour établir l'animation sachant qu'on a pris comme convention *:* ***un pas de quantification*** *de* ***2 degrés*** *et* ***une origine de 0°C***.

N°	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bleu	0	100	85	70	55	40	25	10	0	0	0	0	0	0	0	0
Vert	0	0	10	25	40	55	70	85	100	85	70	55	40	25	10	0
Rouge	0	0	0	0	0	0	0	0	0	10	25	40	55	70	85	100
Tempé-rature		*[0, 2[*	*[2, 4[*	*[4, 6[*	*[6, 8[*	*[8, 10[*	*[10, 12[*	*[12, 14[*	*[14, 16[*	*[16, 18[*	*[18, 20[*	*[20, 22[*	*[22, 24[*	*[24, 26[*	*[26, 28[*	*[28, 30[*

Tableau 1 : Les spécifications *des composantes* ***RVB*** *pour l'animation*

Les ***15 dernières cases*** *[1 ..15]* permettent de passer d'une lumière bleue (***case n°1***) au rouge (***case n°15***) en passant par le vert (***case n°8***). La première (***case n°0***) permet l'extinction de toutes les composantes et sera utilisée lorsque la température mesurée n'est pas comprise dans l'intervalle ***[0°C, 30°C [.***

Effectuer l'animation demandée sachant qu'on peut modifier l'intensité lumineuse d'un objet de la classe **CEcg** *que si le champ* ***m_xStartDaliMaster*** *est dans l'état* **false**. *Que dans ce cas, il vous suffit* ***d'affecter le champ m_bCommandValue*** *avec la nouvelle valeur d'intensité (issue du tableau précédent) et de positionner le champ* ***m_xStartDaliMaster*** *à* ***true*** *pour émettre cette nouvelle valeur sur le réseau. Il est clair que l'on émettra des nouvelles commandes d'éclairage que si l'on détecte que l'animation en cours n'est plus d'actualité.*

Rappel : *Lors du* ***tp n°1****, les classes suivantes ont été établies pour une utilisation aisée des blocs fonctionnels* ***FbDALI_Joblist*** *et* ***FbDALI_Master.***

```
TYPE CDaliJobList:
    STRUCT
        (* données membres *)
        m_byModule_750_641          : BYTE;
        m_byFeedback                : BYTE;
        (* fonctions membres *)
        InstDALIMaster              : FbDALI_Master;
    END_STRUCT
END_TYPE
```

```
TYPE CEcg:
    STRUCT
        (* données membres *)
        m_byAddress                 : BYTE;
        m_iCommand                  : INT;
        m_byCommandValue            : BYTE;
        m_byModule_750_641          : BYTE;
        m_xStartDaliMaster          : BOOL;
        m_byQueryValue              : BYTE;
        m_byFeedback                : BYTE;
        (* fonctions membres *)
        InstDALIMaster              : FbDALI_Master;
    END_STRUCT
END_TYPE
```

```
TYPE TRvb:
     STRUCT
          m_byRouge      : BYTE;
          m_byVert  : BYTE;
          m_byBleu  : BYTE;
     END_STRUCT
END_TYPE
```

```
(* tableau des valeurs pour l’animation en fonction de la température *)
a16Rvb: ARRAY[0..15]OF TRvb:=(m_byRouge:=0,m_byVert:=0,   m_byBleu:=0),
                             (m_byRouge:=0,     m_byVert:=0,   m_byBleu:=100),
                             (m_byRouge:=0,     m_bVert:=10,   m_byBleu:=85),
                             (m_byRouge:=0,     m_bVert:=25,   m_byBleu:=70),
                             (m_byRouge:=0,     m_bVert:=40,   m_byBleu:=55),
                             (m_byRouge:=0,     m_bVert:=55,   m_byBleu:=40),
                             (m_byRouge:=0,     m_bVert:=70,   m_byBleu:=25),
                             (m_byRouge:=0,     m_byVert:=85,  m_byBleu:=10),
                             (m_byRouge:=0,     m_byVert:=100, m_byBleu :=0),
                             (m_byRouge:=10,    m_byVert:=85,  m_byBleu:=0),
                             (m_byRouge:=25,    m_byVert:=70,  m_byBleu:=0),
                             (m_byRouge:=40,    m_byVert:=55,  m_byBleu:=0),
                             (m_byRouge:=55,    m_byVert:=40,  m_byBleu:=0),
                             (m_byRouge:=70,    m_byVert:=25,  m_byBleu:=0),
                             (m_byRouge:=85,    m_byVert:=10,  m_byBleu:=0),
                             (m_byRouge:=100,   m_byVert:=0,   m_byBleu:=0) ;
```

```
(* Programme EnOcean *)
PROGRAM EnOcean
VAR
     enoceanRcvRS485 : CEnoceanRcvRS485 := (.....................................);
     sr65C264 : CSr65 := (   .............................................
                             ................................................
                             ................................................
                             ................................................
                             ...............................................);
     ...................................................................
     ...................................................................
     ...................................................................
     ...................................................................
     ...................................................................
     ...................................................................
     ...................................................................
     ...................................................................
     ...................................................................
END_VAR
```

```
(* Debut *)
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
```

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....