### DOCUMENT RÉPONSES – Domaine Professionnel À RENDRE AVEC LA COPIE

Réponse aux questions Q1 et Q2



Réponse à la question Q3 et Q4



Réponse aux questions Q6 et Q7

Réseau APN Privé 172.16.88.0

|  |  |  |
| --- | --- | --- |
| **Masque de sous réseau** | **Notation CIDR** | **Notation décimale pointée** |
|  |  |
| **Nombre d’hôtes par****sous réseau** |  |

Réponse à la question Q8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Adresse sous réseau** | **1ère adresse hôte** | **Dernière adresse hôte** | **Adresse de diffusion** |
| **1èr sous réseau** |  |  |  |  |
| **2èmesous réseau** |  |  |  |  |
| **…** | … | … | … | … |
| **64ème sous réseau** |  |  |  |  |

Réponse à la question Q11

|  |  |
| --- | --- |
| **Ordre d’envoi** | **Syntaxe** |
| **1er SMS** | **MT1 = . . . . . . . . . . . . . . . . . .** |
| **2ème SMS** | **. . . . . = . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .** |

Réponse à la question Q12

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Champs** | **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** |
| **Taille en****Octets** |  |  |  |  |  |  | 0 à 255 |  |

Rôle du champ A :

Réponse à la question Q13

Trame :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Champs** | **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** |
| **Valeur en Hexadécimal** | **0x00****0x00****0x00****0x00****0x00** |  | **0x5A** |  |  |  | **Fermé, bacs plein, réouverture le 15** |  |

Réponse à la question Q14

Fichier panneau.h

#ifndef \_PANNEAU\_H #define \_PANNEAU\_H

#include "protocole.h"

class CPanneau : public . . . .

{

private:

CProtocole protocole;

. . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . .

public:

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

};

#endif

Réponse à la question Q15

Fichier protocole.cpp

#include "protocole.h"

#define NUL 0 //

#define SOH 1 // Début entête

#define STX 2 // Début texte

#define ETX 3 // Fin texte

#define EOT 4 // Fin de transmission #define ACK 6

#define NAK 21

#define ESC 27 //(0x1B) Mode field

#define ALL\_SIGNS 0x5A // Type Code = All Signs

#define WRITE\_TEXT\_FILE 0x41 // Command Code = Write TEXT file

CProtocole::CProtocole(void){} CProtocole::~CProtocole(void){}

//méthodes de la classe non détaillées

void CProtocole::ajouterTexte(string texte){…} //ajoute le message à la trame void CProtocole::ajouterControle(){…} //ajoute la fin de trame EOT

// Création de la trame à envoyer pour une transmission standard au panneau

// En entrée: num: numéro du panneau destinataire (1 à 99). 0 pour diffuser à tous les panneaux

// texte: texte du message à envoyer.

void CProtocole::fabriquerTrame(int num, string texte)

{

. . . . . . . . . . . . . . . . . . . . . . . . . . . . // Ajout des caractères de

. . . . . . . . . . . . . . . . . . . . . . . . . . . . // synchronisation (Champ A)

trame[5]= SOH ; . . . . . . . . . . . . . . . . . . . . // Champ B

. . . . . . . . . . . . . . . . . . . . . . . . . . . . // Champ C

. . . . . . . . . . . . . . . . . . . . . . . . . . . . // Champ D

. . . . . . . . . . . . . . . . . . . . . . . . . . . . // Champ E

. . . . . . . . . . . . . . . . . . . . . . . . . . . . // Champ F

ajouterTexte(texte) ; // Champ G

ajouterControle(); // Champ H

}

Réponse à la question Q16

Fichier protocole.cpp (suite)

// Ecriture de la trame sur le flux d'E/S f.

// Si le numéro de panneau num est 0, tous

// les panneaux reçoivent la trame, mais aucun ne renvoie d'acquittement.

// Si num est différent de 0, un panneau particulier est destinataire de la trame et

// renvoie un caractère d'acquittement (ACK ou NAK)

// Valeur de retour : false si acquittement négatif reçu (NAK)

// true dans tous les autres cas.

// Note: fonction bloquante en cas de non réponse du panneau. Aucun timeout n'est

// mis en place dans cette version du logiciel.

bool CProtocole::envoyerTrame (int num, string texte)

{

int i = 0;

fabriquerTrame(num,texte);

//compose la trame à envoyer vers le panneau

while(trame[i] != EOT)

f.put(trame[i++] ;

//écriture de la trame sur le flux de communication

if ( )

{

char acquittement;

//pas de broadcast : un seul panneau destinataire

. . . . . . . . . . . . . . . . . //Lecture du caractère d’acquittement sur le flux de

// communication

. . . . . . . . . . . . . . . . . //Si pas d’acquittement

. . . . . . . . . . . . . . . . . //retourner faux

}

else return true;

}

Réponse à la question Q17

