PROTOCOLE DE CONTROLE RESEAU MASTERCLOCK ®
pour le CONTROLE A DISTANCE des afficheurs d'horloge numerique NTDSXX

PARAMETRES DE COMMUNICATION :

• Multicast, IP par defaut=239.252.0.0, port=6168 et configurable a l'aide des commandes TELNET ou des fonctions administratives WinDiscovery (Figure 1)

• Paquets de datagrammes UDP FORMAT DU TAMPON DE PAQUET : <HDR1><HDR2><RSRV1><DEVICE><FAMILY><RSRV2><ZEROS><RSRV3><CTRLCODE><H><M><S>* Ou :

<HDR1>	4 octets	=	(2381D765 16)
<HDR2>	4 octets	=	(10B32FE1 16)
<RSRV1>	2 octets	=	Rempli de zeros
<DEVICE>	2 octets	=	Les deux octets de poids faible de l'adresse MAC du peripherique de controle a distance -ou- ID de la source de controle auxiliaire
<FAMILY>	4 octets	=	(00000080_{16})
<RSRV2>	3 octets	=	Rempli de zeros
<ZEROS>	1 octet	=	(00 16) : Zeros de tete ON (01 16) : Zeros de tete OFF
<RSRV3>	24 octets	=	Rempli de zeros
<CTRLCODE>	1 octet	=	(00 16) : Liberer le controle a distance, horloge reglee sur l'affichage normal (01(02 16) : L'horloge affiche les valeurs <H><M><S> 16) : L'horloge affiche des blancs (02 16) : L'horloge affiche les valeurs <H><M><S> (03 16) : L'horloge affiche des tirets
<H>	1 octet	=	Affichage des heures (en hexadecimal)
<M>	1 octet	=	Affichage des minutes (en hexadecimal)
<S>	1 octet	=	Affichage des secondes (en hexadecimal)
<DAYS>	2 octets	=	Jours (1 – 366) (en hexadecimal) (note : fonction d'affichage non prise en charge par toutes les horloges)
<CHANNEL>	1 octet	=	Canal source (A, B ou C) (en ASCII) (note : regler sur 0 (en hexadecimal) si non necessaire)

*NOTE : Ce tampon de paquet UDP est chiffre avant la transmission, le code C de cet algorithme est fourni dans l'Annexe A.

**NOTE : Les peripheriques de controle a distance Masterclock® comprennent les modeles : RC1000, RC600 et RC500.

NOTES D'UTILISATION :

• Pendant le controle a distance (c'est-a-dire que le NTDSxx n'a pas recu de commande de fin de controle <CTRLCODE> = 0), l'horloge s'attend a recevoir des donnees a un debit minimal d'une transmission par seconde, meme lorsque l'affichage est en pause ou arrete (dans ces cas, les donnees sont simplement retransmises en continu.) Si ces transmissions sont interrompues pendant plus de trois secondes, les deux-points clignotent pour indiquer une perte de communication, et l'affichage se fige avec la derniere valeur mise a jour. L'horloge est configurable pour revenir au mode normal apres une periode de temporisation predefinie a l'aide des commandes TELNET ou des fonctions administratives WinDiscovery (Figure 1).

• Pendant une session de controle, le NTDSxx ne fournit aucune fonction de synchronisation interne et aucune verification d'erreur sur les valeurs qu'il affiche. Il ne synchronise pas non plus les mises a jour de l'affichage sur le debut de seconde, bien que la synchronisation temporelle interne soit maintenue et reprenne apres la liberation du controle a distance.

• Il est possible d'envoyer des donnees a un debit superieur a une fois par seconde, et le NTDSxx devrait repondre a ce debit plus rapide, bien que la limite de ce debit n'ait pas ete testee.

• Aucune disposition n'est prevue pour controler les deux-points ou les decimales du NTDSxx ; les deux-points restent allumes et les decimales eteintes pendant une communication de controle a distance ininterrompue.

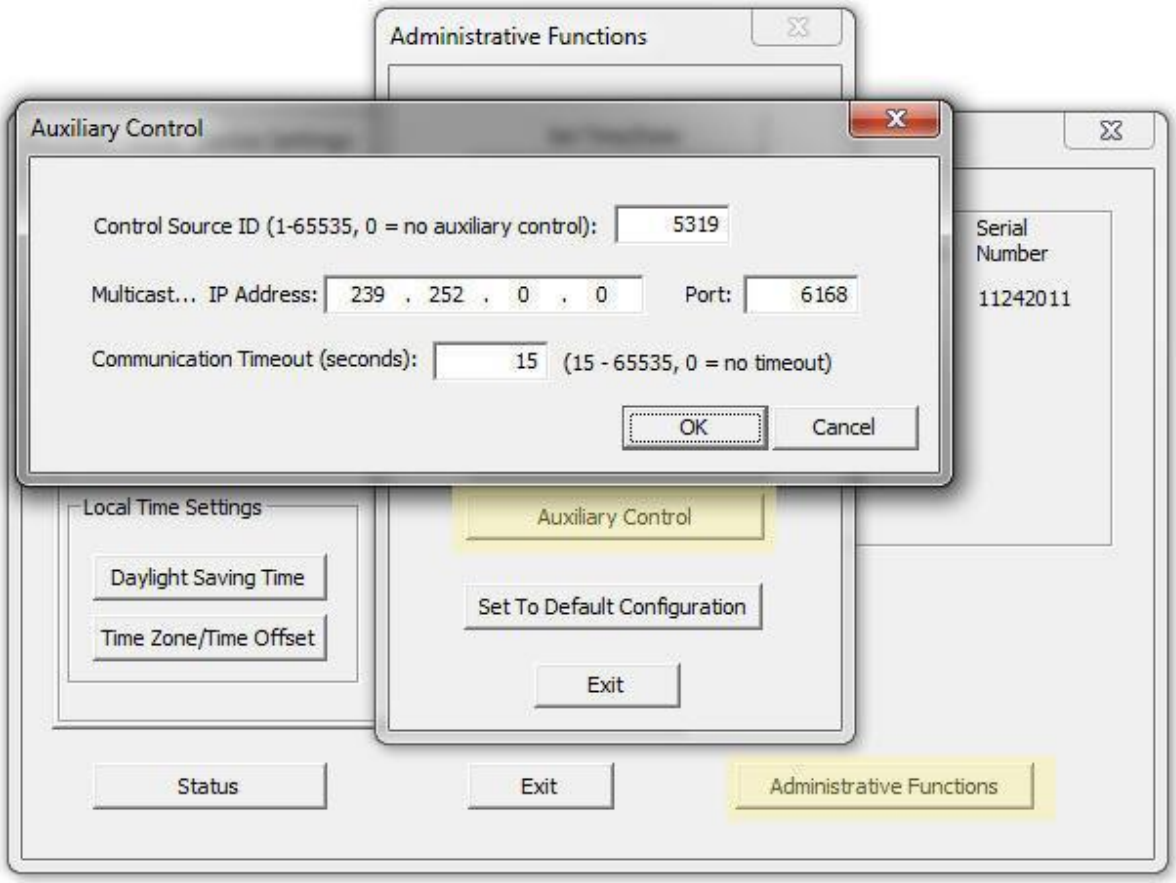
• La charge utile de donnees <H><M><S> d'un paquet <CTRLCODE> = 0 est arbitraire (c'est-a-dire non affichee sur l'horloge) et generalement reglee sur des zeros.

• Pour les horloges NTDSxx controlees par un peripherique autre qu'un peripherique de controle a distance Masterclock, la valeur <DEVICE> peut etre attribuee arbitrairement ; cependant, l'horloge doit etre configuree via les commandes TELNET ou les fonctions administratives WinDiscovery (Figure1) pour accepter le controle depuis cette adresse <DEVICE>.

• Les paquets de controle de datagramme UDP doivent etre transmis dans l'ordre des octets reseau (en code C, cela est realise a l'aide des fonctions htonl() et htons() sur les types de donnees UINT32 et UINT16, respectivement.)

• La nature du multicast exige que les peripheriques « rejoignent » et « quittent » des groupes en fonction d'une combinaison adresse IP/port attribuee. Il est important de noter que sur certains routeurs, le multicast n'est pas active par defaut et doit etre configure pour autoriser cette communication.

FIGURE 1 :


Administrative Functions
Auxiliary Control
Control Source ID (1-65535, 0 = no auxiliary control):
5319
Multicast... IP Address:
239 . 252 . 0 . 0
Port:
6168
Communication Timeout (seconds):
15
(15 - 65535, 0 = no timeout)
OK
Cancel
Serial Number
11242011
Local Time Settings
Daylight Saving Time
Time Zone/Time Offset
Auxiliary Control
Set To Default Configuration
Exit
Status
Exit
Administrative Functions

EXEMPLE : Pour une horloge a l'ecoute du peripherique (14C7 16) sur l'IP multicast 239.252.0.0:6168 afin d'afficher « 12:34:56 », la procedure suivante serait utilisee :

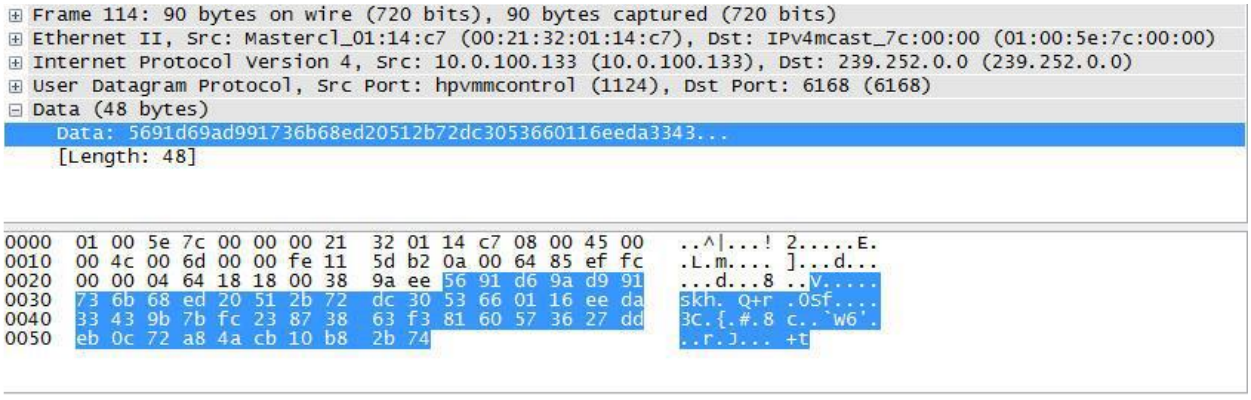
• Fill a 48-element array with the hexadecimal values (note **<H><M><S>** are also hexadecimal.)

23 81 D7 65 10 B3 2F E1 00 00 14 C7 00 00 00 80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02 0C 22 38

• Execute the *crypt()* function on this array as outlined in Appendix A to yield

56 91 D6 9A D9 91 73 6B 68 ED 20 51 2B 72 DC 30 53 66 01 16 EE DA 33 43 9B 7B FC 23 87 38 63 F3 81 60 57 36 27 DD EB 0C 72 A8 4A CB 10 B8 2B 74

• Diffuser ce paquet en multicast sur 239.252.0.0:6168. Une capture d'ecran WireShark montre le datagramme UDP complet pour ce paquet de controle, avec la partie de donnees mise en surbrillance…



ANNEXE A :

```
#define
KEY_SIZE                17

const UINT8 _key[KEY_SIZE] =
        { 0x74, 0x12, 0x02, 0xfb, 0xcc, 0x24, 0x5b, 0x82,
        0x61, 0xe7, 0x3f, 0x9a, 0x26, 0x7c, 0xd3, 0xa0, 0x42 };

void crypt(UINT8 *buf, UINT32 bsize);

void crypt(UINT8 *buf, UINT32 bsize)

{

        UINT8 padcnt, keycnt, *p;

        UINT32                  t;

padcnt = 1;
keycnt = 0;
t = 0;
p = buf;

while(t < bsize)

{

        *p = *p ^ padcnt ^ _key[keycnt];

        if (++keycnt == KEY_SIZE) keycnt = 0;
        if (++padcnt == 254) padcnt = 1;


        p++;

        t++;

        }

}
```