



 MarathonTP[®]

\\Référence du protocole

 NERTIA
SYSTEMES

\\Copyrights

Inertia Systèmes
73 rue des Colombes
Ange-Gardien, Québec
J0E 1E0
<http://inertiasystemes.com>

\\Avertissements

Ce manuel de référence est à titre informatif uniquement et est fourni « tel quel ». Les informations énoncées dans le présent document sont présentées comme un guide et non comme un processus étape par étape. Nous recommandons fortement que vous engagiez une expertise supplémentaire afin d'évaluer davantage les conditions requises pour votre environnement spécifique.

INERTIA SYSTÈMES SE RÉSERVE LE DROIT D'APPORTER DES MODIFICATIONS AUX SPÉCIFICATIONS DU PROTOCOLE À TOUT MOMENT SANS PRÉAVIS.

LES UTILISATEURS DOIVENT ASSUMER L'ENTIÈRE RESPONSABILITÉ POUR L'APPLICATION DES RÈGLES MENTIONNÉS DANS LES PRÉSENTES. INERTIA SYSTÈMES N'ASSUME AUCUNE RESPONSABILITÉ, ET EXCLUT TOUTE GARANTIE, EXPRESSE OU IMPLICITE, RELATIVE À L'APPLICATION DES RÈGLES DANS LES PRÉSENTES.

L'obtention de ce document ne vous donne aucun droit de licence pour les brevets, marques, droits d'auteur ou autre propriété intellectuelle.

\\Table des matières

\\Copyrights.....	2
\\Avertissements.....	3
\\Liste des figures.....	6
\\Liste des tableaux.....	7
1 \\Introduction.....	8
2 \\Types de données.....	9
2.1 Représentation des nombres.....	10
2.2 Notes sur les chaînes de texte.....	10
3 \\Structure de message.....	11
3.1 Définition des éléments.....	11
3.1.1 Les marqueurs de début et de fin.....	11
3.1.2 Les séparateurs de champs.....	11
3.1.3 Le descripteur.....	12
3.1.4 La charge utile.....	13
4 \\Commandes MarathonTP.....	14
4.1 Lecture de variable.....	14
4.1.1 Format de la requête.....	14
4.1.2 Format de réponse.....	15
4.2 Écriture de variable.....	16
4.2.1 Format de la requête.....	16
4.2.2 Format de la réponse.....	17
4.3 Codes de réponse.....	17
5 \\Organigramme des messages.....	18
5.1 Émission d'un message.....	18
5.2 Réception d'un message.....	19
5.3 Recyclage des messages.....	20
6 \\Listes d'échange.....	21
6.1 Précision des index réservés.....	21
6.1.1 Index 0 : Ping.....	22
6.1.2 Index 1 : Device Serial.....	22
6.1.3 Index 2 : Device IS Identifier.....	22

6.1.4	Index 10 : Sended Count	22
6.1.5	Index 11 : Received Count.....	22
6.1.6	Index 12 : Failed Count.....	22
6.1.7	Index 13 : Retried Count	23
6.1.8	Index 14 : Successful Per Second	23
7	\\Spécifications du protocole.....	24
7.1	Port de communication.....	24

\\Liste des figures

Figure 1 : Modèle OSI	8
Figure 2 : Format d'un paquet MarathonTP.....	11
Figure 3 : Liste d'objet triple	13
Figure 4 : Organigramme d'émission.....	18
Figure 5 : Organigramme de réception.....	19
Figure 6 : Organigramme de recyclage.....	20

\\Liste des tableaux

Tableau 1 : Type de données du protocole	9
Tableau 2 : Caractères réservés	10
Tableau 3 : Champs de descripteur	12
Tableau 4 : Codes d'erreur MarathonTP	17
Tableau 5 : Répartition des plages d'index	21

1 \\Introduction

MarathonTP est un protocole de communication machine à machine (M2M). Il est conçu pour être simple d'implémentation et pouvant s'adapter à une multitude de plateforme capable de gérer des chaînes de caractère et la couche de transport UDP. Il permet l'échange fiable de message entre dispositifs de ressources limitées. Ces dispositifs utilisent souvent des microcontrôleurs 8 à 32 bits bon marché.

MarathonTP se distingue par le fait qu'il est dit « humainement lisible ». En effet, les messages transmis sont en fait des chaînes de caractère de format [UTF-8](#). Ainsi, lors de son implémentation, il est aisé de confirmer la structure des messages transmis avec des outils d'analyse réseau comme [WireShark](#).

Selon le modèle OSI (Open Source Interconnection), MarathonTP est un protocole de la couche 7, soit la couche d'application. Du point de vue du modèle OSI, la couche d'application est responsable de la représentation des données à l'utilisateur, de leur codage et du contrôle du dialogue entre les systèmes. MarathonTP défini à la fois une structure standardisé de message ainsi qu'une méthodologie de transfert fiable. Ce document ne couvre pas l'établissement des connections entre les machines.

Le nom du protocole a été choisi pour décrire les buts recherchés lors de son développement. Le marathon représente une épreuve d'endurance et l'acronyme « TP » signifie « Transport Protocol ». MarathonTP est donc un protocole de transport d'information fiable et durable.



Figure 1 : Modèle OSI

2 \Types de données

Le protocole prévoit l'échange d'information sous plusieurs types. À chacun de ces types correspond donc un identificateur unique. Celui-ci permet à l'interpréteur du message de convertir la chaîne de caractère reçu dans le bon format. Le tableau suivant fait la synthèse des types que le protocole supporte :

Type de donnée	Description	Identificateur	Valeurs possibles
BOOLEAN	Contient des valeurs qui ne peuvent être que vrai ou faux. Les mots clés True et False correspondent aux deux États de variables booléennes.	« Bo »	« True » « False »
INTEGER	Contient des entiers signés de 32 bits (4 octets).	« In »	Toutes valeurs entières entre : « -2147483648 » et « 2147483647 ».
SHORT	Contient des entiers signés de 16 bits (2 octets).	« Sh »	Toutes valeurs entières entre : « -32768 » et « 32767 ».
USHORT	Contient des entiers non signés de 16 bits (2 octets).	« USh »	Toutes valeurs entières entre : « 0 » et « 65535 ».
LONG	Contient des entiers signés de 64 bits (8 octets).	« Lo »	Toutes valeurs entières entre : « -922337236854775808 » et « 922337236854775807 ».
SINGLE	Contient des nombres signés IEEE 32 bits à virgule flottante de simple précision (4 octets). Les nombres en simple précision stockent une approximation d'un nombre réel.	« Si »	Pour les nombres négatifs : de « - 3,4028235E+38 » à « - 1,401298E-45 ». Pour les nombres positifs : de « 1,401298E-45 » à « 3,4028235E+38 ».
DOUBLE	Contient des nombres signés IEEE 64 bits à virgule flottante de double précision (8 octets). Les nombres en double précision stockent une approximation d'un nombre réel.	« Do »	Pour les nombres négatifs : de « -1.79769313486231570E+308 » à « -4.94065645841246544E-324 ». Pour les nombres positifs : de « 4.94065645841246544E-324 » à « 1.79769313486231570E+308 ».
BYTE	Contient des entiers non signés de 8 bits (1 octet).	« By »	Toutes valeurs entières entre : « 0 » et « 255 ».
STRING	Représente une chaîne de texte constituée de caractères Unicode.	« St »	Tous les caractères UTF-8 à l'exception des caractères réservés « { », « } » et « : ».
NUL	Représente une donnée de valeur nulle. Ce type est utilisé en cas d'erreur.	« Nil »	La chaîne de caractère « Nil ».

Tableau 1 : Type de données du protocole

2.1 Représentation des nombres

MarathonTP supporte les nombres sous la notation à point fixe ou scientifique. Par contre, le protocole ne définit pas de règle précise sur la représentation de l'une ou l'autre des notations. Les exemples suivants sont tous des représentations valides :

Exemples :

```
0.000135569887426
1.35569887426E-05
220000000000000000
2.2E17
```

2.2 Notes sur les chaînes de texte

MarathonTP utilise trois caractères UTF-8 spécifique dans son implémentation. L'utilisation de ces caractères dans une chaîne de texte occasionnera des exceptions au niveau des interpréteurs de message. Si ces caractères doivent malgré tout être transférés d'une application à l'autre un mécanisme doit être imaginé par le développeur.

Caractère UTF-8	Fonction
{ (Hex 7B)	Marqueur de début de message.
} (Hex 7D)	Marqueur de fin de message.
: (Hex 3A)	Séparateur de champs.

Tableau 2 : Caractères réservés

3 \\Structure de message

Le protocole MarathonTP définit un modèle de message précis permettant aux dispositifs compatibles d'échanger des informations entre eux.

Un message MarathonTP, aussi appelé « paquet », se compose de quatre éléments fondamentaux.

1. Les marqueurs de début et de fin de message
2. Les séparateurs de champs
3. Le descripteur
4. La charge utile



Figure 2 : Format d'un paquet MarathonTP

3.1 Définition des éléments

3.1.1 Les marqueurs de début et de fin

Les messages MarathonTP débutent par le caractère UTF-8 « { », Hex 7B et se terminent par le caractère UTF-8 « } » Hex 7D.

Il s'agit de deux caractères réservés par le protocole qui ne peuvent en aucun cas faire partie du message en lui-même. (Voir le point 2.2 pour la gestion d'exception)

3.1.2 Les séparateurs de champs

Les séparateurs de champs délimitent les différents éléments constitutifs d'un paquet. Ils ne sont présents que dans le descripteur et la charge utile. Ils sont représentés par le caractère UTF-8 « : », Hex 3A.

Il s'agit d'un caractère réservé par le protocole qui ne peut en aucun cas faire partie du message en lui-même. (Voir le point 2.2 pour la gestion d'exception)

3.1.3 Le descripteur

Le descripteur sert à identifier la nature du message. Il contient 4 éléments descriptifs obligatoires et, selon le cas, un élément complémentaire. Le tableau suivant fait l'énumération des ces éléments :

Éléments	Description	Condition	Valeurs possibles
VER	Version du protocole.	Requis	« 1.0 »
RA	Type de message « requête » ou « réponse ».	Requis	« R » ou « A »
TNS	Numéro de transaction.	Requis	« 0-65535 »
CMD	Commande du message.	Requis	« 0-255 »

Tableau 3 : Champs de descripteur

VER : La version du protocole permet l'évolution de celui-ci. Ainsi, un système particulier pourrait répondre à l'une ou l'autre des versions disponibles selon les fonctions utilisées. Ce champ assure l'interopérabilité des systèmes.

RA : Le type de message. Pour chaque commande il y a deux types de message. La requête « R » et la réponse « A ».

TNS : La notion de transaction permet la traçabilité d'un message entre deux systèmes. Tous les messages actifs doivent impérativement avoir un numéro de transaction unique. Comme la quantité de numéro est limitée à 65536, il est possible de reprendre des numéros déjà utilisés. Pour autant qu'est respectée la règle d'unicité.

CMD : Il existe deux types de message MarathonTP. Ceux-ci sont traités au point 4.

3.1.4 La charge utile

La charge utile est le cœur du message. C'est à ce niveau que se retrouve toutes les données d'échange entre les machines. La charge utile se présente sous la forme d'une liste d'objet. Le format de ces objets varie selon la nature du message échangé. Il y a trois types d'objet. Les objets simple, double et triple qui contiennent respectivement 1, 2 et 3 éléments. La figure suivante illustre l'exemple d'une liste d'objet triple :

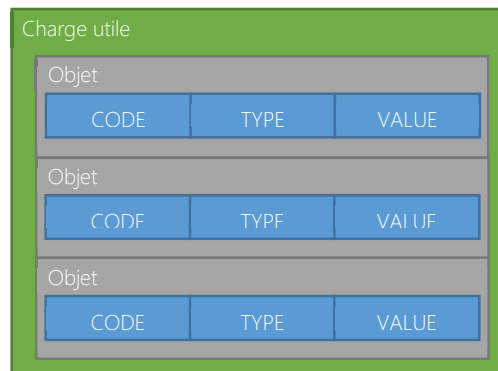


Figure 3 : Liste d'objet triple

4 \\Commandes MarathonTP

MarathonTP suit un modèle de transaction du type requête/réponse. Ce modèle contient deux types de commande. Chacune de ces commandes définit une tâche bien précise à exécuter par le système qui la reçoit.

Comme il a été vu à la section 3.1.3, la commande de message est le 4e paramètre du descripteur. Il s'agit d'un élément obligatoire. Le protocole est prévu pour définir jusqu'à 255 commandes différentes. À l'heure actuelle, MarathonTP définit deux commandes distinctes. La structure de message associée à chacune de ces commandes suit un format bien précis. La conformité d'une implémentation MarathonTP dépend grandement du respect de ce formatage.

Puisque le protocole est de type requête/réponse, on retrouve deux formats de requête et deux formats de réponse. Pour un total de 4 formats différents.

4.1 Lecture de variable

Il s'agit de la commande « 1 ». La lecture de variable permet d'obtenir la valeur d'un ou plusieurs éléments de la liste d'échange du système visé. La liste d'échange est traitée au point 6. La quantité maximale d'élément pouvant être lu avec la même requête est limitée à 10. Combiner plusieurs éléments en une seule requête permet, dans de nombreux scénarios, de maximiser la bande passante du réseau sous-jacent.

4.1.1 Format de la requête

Le tableau suivant énumère les différents éléments de la requête.

Éléments	Description	Valeurs possibles
VER	Version du protocole.	« 1.0 »
RA	Type de message « requête » ou « réponse ».	« R »
TNS	Numéro de transaction.	« 0-65535 »
CMD	Commande du message.	« 1 »
...	Début d'énumération.	
ELE	Élément de la liste d'échange.	« 0 » et « 65535 »
...	Fin d'énumération.	

Exemple de message :

1.0	R	25693	1	0	1
VER	RA	TNS	CMD	ELE	ELE

Paquet : {1.0:R:25693:2:0:1}

4.1.2 Format de réponse

L'énumération de réponse doit correspondre à l'énumération de la requête correspondante. Chaque élément est accompagné d'un code de réponse tel que défini au point 4.3.

Pour tout code de réponse différent de 0, il est obligatoire d'utiliser le type d'élément « NUL » du tableau 1, accompagné d'une valeur de 0.

Éléments	Description	Valeurs possibles
VER	Version du protocole.	« 1.0 »
RA	Type de message « requête » ou « réponse ».	« A »
TNS	Numéro de transaction.	« 0-65535 »
CMD	Commande du message.	« 1 »
...	Début d'énumération.	
CODE	Code de réponse.	Voir le tableau 4
TYP	Type de l'élément	Voir le tableau 1
VALUE	Valeur de l'élément	Voir le point 2.1
...	Fin d'énumération.	

Exemple 1 : Message résultant de la requête du point 4.1.1. :

1.0	A	25693	1	0	Si	84.83	0	Do	8.936E+10
VER	RA	TNS	CMD	CODE	TYPE	VALUE	CODE	TYPE	VALUE

Paquet : {1.0:A:25693:2:0:Si:84.83:0:Do:8.936E+10}

Exemple 2 : Message résultant de la requête du point 4.1.1, mais dont l'élément « 1 » ne fait pas partie de la liste d'échange du récepteur. :

1.0	A	25693	1	0	Si	84.83	1	Nil	0
VER	RA	TNS	CMD	CODE	TYPE	VALUE	CODE	TYPE	VALUE

Paquet : {1.0:A:25693:1:0:Si:84.83:1:Nil:0}

4.2 Écriture de variable

Il s'agit de la commande « 2 ». L'écriture de variable permet de mettre à jour la valeur d'un ou plusieurs éléments de la liste d'échange du système visé. La liste d'échange est traitée au point 6. La quantité maximale d'élément pouvant être mise à jour avec la même requête est limitée à 10. Combiner plusieurs éléments en une seule requête permet, dans de nombreux scénarios, de maximiser la bande passante du réseau sous-jacent.

4.2.1 Format de la requête

Le tableau suivant énumère les différents éléments de la requête.

Éléments	Description	Valeurs possibles
VER	Version du protocole.	« 1.0 »
RA	Type de message « requête » ou « réponse ».	« R »
TNS	Numéro de transaction.	« 0-65535 »
CMD	Commande du message.	« 2 »
...	Début d'énumération.	
ELE	Élément de la liste d'échange.	« 0 » et « 65535 »
VALUE	Valeur à inscrire à l'élément.	Voir le point 2.1
...	Fin d'énumération.	

Exemple de message :

1.0	R	25693	2	0	25.6	1	8.15698563
VER	RA	TNS	CMD	ELE	VALUE	ELE	VALUE

Paquet : {1.0:R:25693:3:0:25.6:1:8.156985631}

4.2.2 Format de la réponse

Une écriture de variable ne retourne pas de valeur. Simplement un code de réponse. L'ordre dans lequel sont envoyé les code de réponse est le même que celui du message de requête correspondant.

Éléments	Description	Valeurs possibles
VER	Version du protocole.	« 1.0 »
RA	Type de message « requête » ou « réponse ».	« R »
TNS	Numéro de transaction.	« 0-65535 »
CMD	Commande du message.	« 2 »
...	Début d'énumération.	
CODE	Code de réponse	« »
...	Fin d'énumération.	

Exemple de message résultant de la requête du point 4.2.1 :

1.0	A	25693	2	0	1
VER	RA	TNS	CMD	CODE	CODE

Paquet : {1.0:A:25693:3:0:1}

4.3 Codes de réponse

Toutes les réponses MarathonTP sont accompagnées de code définissant l'état des transactions. Il y a des circonstances où les requêtes MarathonTP occasionneront des erreurs dans les systèmes qui les exécuteront. Le protocole prévoit quatre types d'erreur défini selon un code numérique. Le tableau ci-dessous en fait l'énumération :

Code d'erreur	Description
0	L'opération s'est correctement déroulée.
1	L'élément est introuvable.
2	Le type de donnée est incompatible. Cette erreur survient lors de l'écriture de variable.
3	Index en dehors des limites de la liste d'échange.

Tableau 4 : Codes d'erreur MarathonTP

5 \\Organigramme des messages

Les messages MarathonTP sont de type requête/réponse. Cette méthodologie implique que l'envoi d'un message vers un destinataire doit impérativement se terminer par la réception de la réponse au niveau de l'émetteur du message. Le tout dans un délai prédéfini. Si ces conditions ne sont pas rencontrées alors le message est considéré perdu et doit être envoyé de nouveau au besoin.

5.1 Émission d'un message

L'organigramme suivant illustre la séquence d'opération pour l'émission d'un message de requête MarathonTP.

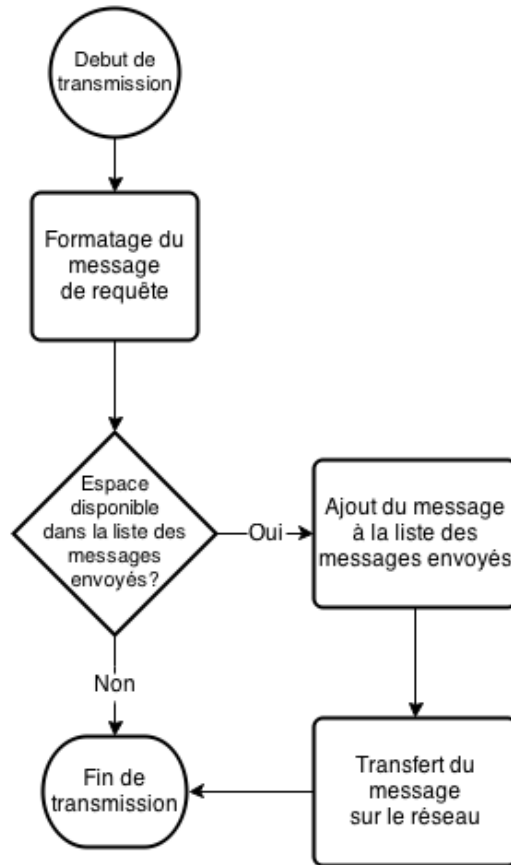


Figure 4 : Organigramme d'émission

5.2 Réception d'un message

L'organigramme suivant illustre la séquence d'opération de la réception des messages MarathonTP.

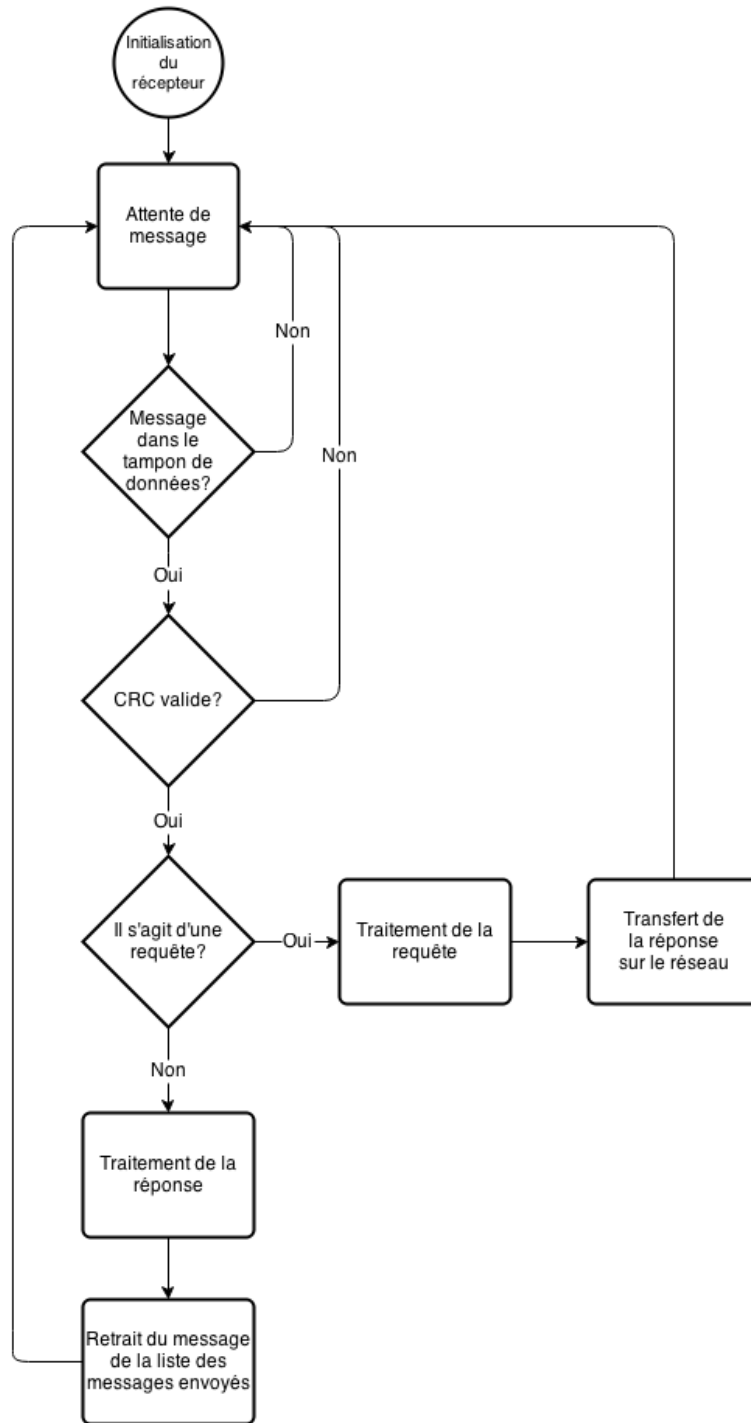


Figure 5 : Organigramme de réception

5.3 Recyclage des messages

L'organigramme suivant illustre la séquence d'opération d'un recycleur de message MarathonTP.

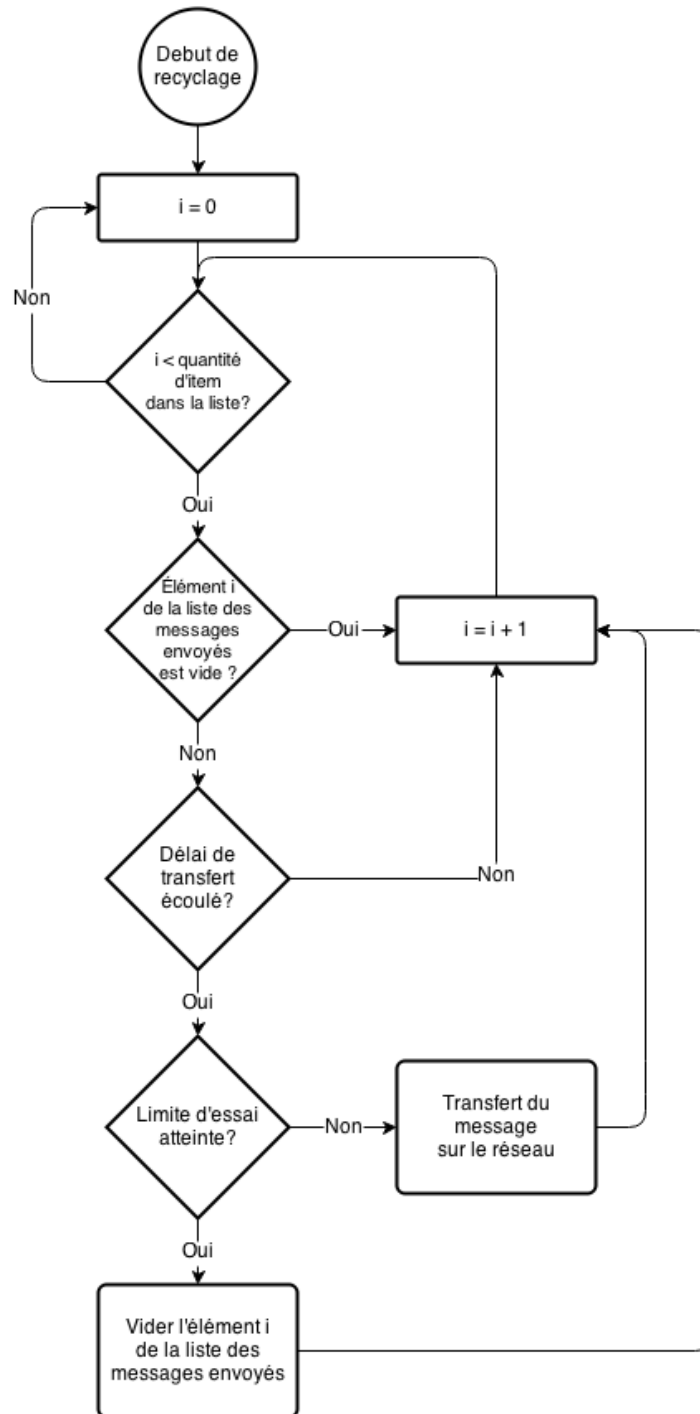


Figure 6 : Organigramme de recyclage

6 \\Listes d'échange

Les accès aux données dans MarathoTP se font par le biais d'index. Ces index sont de type « UShort ». Il y a donc 65536 données distinctes qui peuvent être transférées entre deux systèmes MarathoTP dont 65436 utilisables par les tiers.

Une certaine quantité de ces données sont réservées par le protocole. Le tableau suivant illustre la répartition des plages d'index.

Index	Élément	Type	Description
0	Ping	Boolean	Réponse Ping Maratho TP.
1	Device Serial	String	Numéro de série du dispositif.
2	Device IS Identifier	String	Numéro d'identification Inertia Systèmes du dispositif.
3 à 9			Réservé MarathoTP.
10	Sended Count	Integer	Compte de message envoyé.
11	Received Count	Integer	Compte de message reçu.
12	Failed Count	Integer	Compte de message en échec.
13	Retried Count	Integer	Compte de renvoi de message.
14	Successful Per Second	UShort	Compte de message réussi par seconde.
15 à 99			Réservé MarathoTP.
100 à 65535			Utilisable par les tiers.

Tableau 5 : Répartition des plages d'index.

Il est du rôle du concepteur du dispositif de fournir un document précisant la fonction des 65436 index utilisables pour ses besoins. C'est ce document qui constitue la liste d'échange. La liste d'échange doit être accessible à tout utilisateur du dispositif.

6.1 Précision des index réservés

Les points suivants décrivent avec plus de précision la nature et la fonction des index réservés par le protocole.

6.1.1 Index 0 : Ping

Le « Ping » retourne simplement au client une réponse de type booléenne dont la valeur est vraie. La réception d'une réponse vraie par le client signale l'existence d'un serveur actif à l'adresse IP interrogée.

6.1.2 Index 1 : Device Serial

Le numéro de série unique du dispositif, présentée sous forme textuelle. Le numéro de série est propre au fabricant du dispositif.

6.1.3 Index 2 : Device IS Identifier

Identificateur unique de dispositif fourni par Inertia Systèmes lors de l'approbation de la conformité MarathonTP. Cet identificateur textuel permet aux services et logiciels de consulter la base de données de dispositif d'Inertia Systèmes et d'obtenir les caractéristiques du produit.

6.1.4 Index 10 : Sended Count

Nombre incrémental représentant la quantité de message envoyé par le dispositif. Ce paramètre est utile à des fins d'analyse et de statistique. Le dispositif doit gérer le retour à 0 lors du dépassement de capacité d'un « Integer ».

6.1.5 Index 11 : Received Count

Nombre incrémental représentant la quantité de message reçu par le dispositif. Ce paramètre est utile à des fins d'analyse et de statistique. Le dispositif doit gérer le retour à 0 lors du dépassement de capacité d'un « Integer ».

6.1.6 Index 12 : Failed Count

Nombre incrémental représentant la quantité de message qui n'ont jamais été interprétés par le receveur. Ce paramètre est utile à des fins d'analyse et de statistique. Le dispositif doit gérer le retour à 0 lors du dépassement de capacité d'un « Integer ».

6.1.7 Index 13 : Retried Count

Nombre incrémental représentant la quantité de renvoi de message par le dispositif. Ce paramètre est utile à des fins d'analyse et de statistique. Le dispositif doit gérer le retour à 0 lors du dépassement de capacité d'un « Integer ».

6.1.8 Index 14 : Successful Per Second

Nombre représentant la quantité de message correctement échangés entre deux dispositifs MarathonTP à chaque seconde. Ce nombre est mis à jour aux secondes. Ce paramètre est utile à des fins d'analyse et de statistique.

7 \\Spécifications du protocole

Pour assurer la compatibilité du protocole MarathonTP entre des systèmes de technologies variée, certaines caractéristiques fondamentales doivent être respectées.

7.1 Port de communication

MarathonTP n'a pas encore de port UDP officiel. Pour l'instant, le port non-officiel à utiliser est le port 50000.