



Protocole MQTT

Cette page liste la manière dont se connecte l'application mobile avec l'Arduino via des sujets MQTT

Topics MQTT

Statut

État des capteurs/actionneurs

Les sujets (topics) sont prefixés de `ordurex/`, (ex: `status/trash-0/collect-requested` devient `ordurex/status/trash-0/collect-requested`)

Sujet	Publicateur	Abonné	Commentaire	Type de donnée	QoS*	Rétention	Implém (Public)
status/trash-0/collect-requested	Arduino	Android	En cas de demande de ramassage**	N/A	1	N	✗
status/trash-1/collect-requested	Arduino	Android	En cas de demande de ramassage**	N/A	1	N	✗
status/trash-2/collect-requested	Arduino	Android	En cas de demande de ramassage**	N/A	1	N	✗
status/trash-0/invalid-code	Arduino	Android	Saisie du mauvais code de ramassage	client_id	1	N	✗
status/trash-1/invalid-code	Arduino	Android	Saisie du mauvais code de ramassage	client_id pas de client_id si code du capteur RFID invalide	1	N	✗
status/trash-2/invalid-code	Arduino	Android	Saisie du mauvais code de ramassage	client_id pas de client_id si code du digicode invalide	1	N	✗
status/trash-1/burning	Arduino	Android	État du capteur de flammes	Bool 0 : normal 1 : en feu	1	N	✗
status/trash-0/lid	Arduino	Android	Statut du couvercle	Bool 0 : fermé 1 : ouvert	1	O	✗
status/trash-2/lid	Arduino	Android	Statut du couvercle	Bool 0 : fermé 1 : ouvert	1	O	✗
status/simulation	Arduino	Android	La simulation est initiée par le client donné	Statut 0 : Prêt (inactif) 1 : Occupé (actif) Client (si statut	1	O	✗

Sujet	Publicateur	Abonné	Commentaire	Type de donnée	QoS*	Rétention	Implém (Public:
				= 0) UUIDv4			
status/truck	Android (maître)	Android (autres)	Position en 2D du camion poubelle	Position X Position Y	0	N	✗

QoS* (Quality of Service) :

0 : Au mieux 1 fois (risque de non réception - connexion stable)

1 : Au moins 1 fois (abonné peut recevoir plusieurs fois - connexion instable)

2 : Exactement 1 fois

En cas de demande de ramassage** :

Le status `status/trash-x/collect-requested` existe suite à une demande de ramassage dans les cas suivants

Pour la poubelle 0 :

- Requête de ramassage via l'application et le bon code (`action/trash-0/request-collect`)

Pour la poubelle 1 :

- TRequête de ramassage via l'application et le bon code (`action/trash-1/request-collect`)
- Saisie du bon code via le pavé numérique de la poubelle

Pour la poubelle 2 :

- Requête de ramassage via l'application et le bon code (`action/trash-2/request-collect`)
- Passage d'une carte valide sur le lecteur RFID

Actions

Actions contrôlées par l'application mobile

Sujet	Publicateur	Abonné	Commentaire	Type de donnée	QoS*	Rétention	Implém (Public:
action/trash-0/lid	Android	Arduino	Contrôle l'ouverture ou fermeture du couvercle	Bool 0 : fermeture 1 : ouverture	1	N	✓
action/trash-2/lid	Android	Arduino	Contrôle l'ouverture ou fermeture du couvercle	Bool 0 : fermeture 1 : ouverture	1	N	✓
action/trash-1/buzzer	Android	Arduino	Joue une musique	Int 0 : Musique 1 1 : Musique 2 2 : ...	2	N	✓
action/trash-2/display	Android	Arduino	Affiche du texte sur le LCD	String	2	N	✓
action/trash-0/request-collect	Android	Arduino	Demande de ramassage de la poubelle 0.	client_id code secret (String)	1	N	✓
action/trash-1/request-collect	Android	Arduino	Demande de ramassage de la poubelle 1	client_id code secret (String)	1	N	✓

action/trash-2/request-collect	Android	Arduino	Demande de ramassage de la poubelle 2	client_id code secret (String)	1	N	✓
action/simulation	Android	Arduino	Demande de connexion	Action 0 : arrêter 1 : lancer 2 : pauser ID Client (si action=0) UUIDv4	2	N	✓

QoS* (Quality of Service) :

0 : Au mieux 1 fois (risque de non réception - connexion stable)

1 : Au moins 1 fois (abonné peut recevoir plusieurs fois - connexion instable)

2 : Exactement 1 fois

Échanges TCP/IP

L'arduino envoie l'état des capteurs tous les x ms.

Côté technique pour l'arduino :

- nombre d'octets envoyés/reçus est précisé dans les commandes AT
 - Requette d'envoi `AT+CIPSEND=__longueur_chaine__`
 - Retour de commande :


```
Recv __longueur_chaine__ bytes
SEND OK
```
 - Réception : `+IPD,__longueur_chaine__:__contenu__`
- Utiliser exceptionnellement `Serial.print` plutôt que `Serial.println` pour omettre `\r\n` contrairement aux autres commandes AT

Format de données :

Ordre	Commentaire	Type de donnée
1	Nombre de status à envoyer	octet
3	Liste de status	status[]

Type status :

Ordre	Commentaire	Type de donnée
1	Id de requêtes mqtt	Enum de status : 0 : status/trash-0/collect-requested 1 : status/trash-1/collect-requested 2 : status/trash-2/collect-requested 3 : status/trash-0/invalid-code 4 : status/trash-1/invalid-code 5 : status/trash-2/invalid-code 6 : status/trash-1/burning 7 : status/trash-0/lid 8 : status/trash-2/lid 9 : status/simulation Enum d'actions : 0 : action/trash-0/lid 1 : action/trash-2/lid 2 : action/trash-1/buzzer 3 : action/trash-2/display 4 : action/trash-0/request-collect 5 : action/trash-1/request-collect

		6 : action/trash-2/request-collect 7 : action/simulation
2 (optionnel)	Longueur de chaîne de caractère si type chaîne de caractère	octet
3	Contenu de la requête	Dépend de l'id de la requête mqtt

Exemples :

- Arduino envoi (en binaire) : 323112233445566778899aabbccddeeff0070
 - 3 : 3 status à envoyer
 - 1er status
 - Sujet 2 : status/trash-2/collect-requested
 - Donnée : aucune
 - 2ème status
 - Sujet 3 : status/trash-0/invalid-code
 - Donnée : client_id (16 octets)
 - 3ème status
 - Sujet 7 : status/trash-0/lid
 - Donnée : bool (1 octet)
- Android envoi (en binaire) : 30135salut69112233445566778899aabbccddeeff00motdepass
 - 3 : 3 status à envoyer
 - 1er status
 - Sujet 0 : action/trash-0/lid
 - Donnée : bool (1 octet)
 - 2ème status
 - Sujet 3 : action/trash-2/display
 - Longueur de chaîne de caractère : 5 (octet)
 - Donnée : chaîne de caractères
 - 3ème status
 - Sujet 6 : action/trash-2/request-collect
 - Longueur de chaîne de caractère : 9 (octet)
 - Donnée : client_id (16 octet), chaîne de caractères