

### Fonctionnalité ajoutée :

Ajout de la possibilité de connaître l'état du distributeur de cartes (niveau des cartes, cartes coincées, etc.).

### Étapes d'implémentation :

1. Toutes les 2,5 secondes, envoyer la commande getstatus().
2. Recevoir la réponse et la parser à l'aide du délimiteur \x03.
3. Extraire les 4 derniers bits et les interpréter selon le tableau ci-dessous.
4. Soustraire la valeur obtenue à l'étape 3 des seuils décroissants.
5. Envoyer la liste des messages (dans l'événement “**cd\_status**”) ainsi qu'une variable booléenne indiquant s'il s'agit d'un état bloquant ou non.

### Exemple :

Message reçu de getstatus : “**151515SF1008**”

4 bits extraits : “**1008**”

### Analyse :

- $1008 - 1000 \geq 0 \rightarrow$  vrai  $\rightarrow$  “**bac vidé !**”
- $8 - 200 \geq 0 \rightarrow$  faux
- $8 - 100 \geq 0 \rightarrow$  faux
- $8 - 10 \geq 0 \rightarrow$  faux
- $8 - 8 \geq 0 \rightarrow$  vrai  $\rightarrow$  “**cartes confisquées vidées !**”

### Envoi via MQTT :

```
{  
  "device": "carddispenser",  
  "event": "cd_status",  
  "value": {  
    "message": ["back vidé !", "vidé cartes confisqué !"],  
    "erreur": false/true  
  }  
}
```

-Ahmed Douar-

HEX	BIN	STATUS
8000		(hold)
4000		(hold)
2000	0010 0000 0000 0000	(hold)
1000	0001 0000 0000 0000	Collect bin was full
0800	0000 1000 0000 0000	Dispensing card
0400	0000 0100 0000 0000	Collecting card
0200	0000 0010 0000 0000	Dispense card error
0100	0000 0001 0000 0000	Collect card error
0080	0000 0000 1000 0000	(hold)
0040	0000 0000 0100 0000	Card lap over
0020	0000 0000 0010 0000	Card Jam
0010	0000 0000 0001 0000	Card going to empty
0008	0000 0000 0000 1000	Card empty (no card)
0004	0000 0000 0000 0100	Light sensor 3 status
0002	0000 0000 0000 0010	Light sensor 2 status
0001	0000 0000 0000 0001	Light sensor 1 status