



# PROTOCOLE I2C RUSHBALL

Ce protocole présente la communication entre l'unité centrale Raspberry PI et les panneaux de cibles.

## STS SN session 2020

Ce protocole répond à la nécessité de :

- Récupérer l'état "touché" de chaque cible d'un panneau.
- Fixer la couleur de chaque cible d'un panneau.

Chaque panneau dispose d'un microcontrôleur esclave I2C qui gère 3 cibles.

Pour le moment, le choix de l'adresse de l'esclave, donc du panneau, est définie de façon logicielle.

**Communication** : Les données écrites et reçues seront succédées d'un octet de contrôle (CRC).

**Etat "Touché" d'une cible** : 4 interrupteurs sont câblés en parallèle. Une seule information binaire est fournie au microcontrôleur. Un seul octet permet de transmettre l'état des 3 cibles d'un panneau.

**Couleur d'une cible** : 7 couleurs possibles. 1 octet par cible pour choisir la couleur. Les couleurs sont numérotés de 1 à 7. 0 signifie cible éteinte. La couleur 7 est la couleur joker. Voir tableau de correspondance ci-dessous pour les couleurs.

Code couleur	Valeur RVB de couleur (hex)	Info
0	0-0-0	Cible éteinte
1	FF-0-0	Rouge
2	0-FF-0	Vert
3	0-0-FF	Bleu
4	FF-0-FF	Magenta
5	0-FF-FF	Cyan
6	FF-FF-FF	Blanc
7	FF-FF-0	Jaune ( <i>joker</i> )



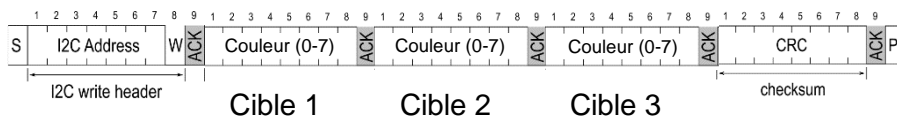
## Suivi des modifications

Date	Modification	Auteur
12/05/2020	v1.0 Document initial livré par l'enseignant.	PhA
13/05/2020	V1.1 Simplification du protocole suite à la précision du fonctionnement de la commande de l'éclairage d'une cible.	CH, PhA
15/05/2020	Limitation à 7 couleurs pour une cible. La couleur 7 est la couleur bonus.	PhA
04/06/2020	Définition des couleurs	PhA



## Trame I2C d'écriture

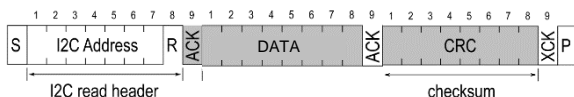
Cette trame permet de fixer les couleurs de chaque cible d'un panneau.



**Note :** Les bits de couleur sombre sont émis par l'esclave.

## Trame I2C de lecture

Cette trame permet de lire l'état "touché" de chaque cible du panneau.



**Note :** Les bits de couleur sombre sont émis par l'esclave.

### Détail de l'octet data :

- Bit 0 (LSB) : état 1 signifie Cible 1 touchée. 0 non touchée.
- Bit 1 (LSB) : état 1 signifie Cible 2 touchée. 0 non touchée.
- Bit 2 (LSB) : état 1 signifie Cible 3 touchée. 0 non touchée.

## Méthode C++ de codage du CRC

```
#define CRC_INIT 0xFF
#define CRC_POLY 0x31

uint8_t CMAclasse::generateCRC(uint8_t *data, uint8_t datalen) {
    // calculates 8-Bit checksum with given polynomial
    uint8_t crc = CRC_INIT;

    for (uint8_t i = 0; i < datalen; i++) {
        crc ^= data[i];
        for (uint8_t b = 0; b < 8; b++) {
            if (crc & 0x80)
                crc = static_cast<uint8_t>((crc << 1) ^ CRC_POLY);
            else
                crc <<= 1;
        }
    }
    return crc;
}
```