

Production de librairie statique et stratégie de débogage

Travaux pratiques 7 et 8

Laurent Bourgon

Mehdi Benouhoud

Ihsane Majdoubi

Catalina Andrea Araya Figueroa

Lundi 13 mars 2023

1 Description de la librairie

1.1 Classes

1.1.1 LED

Depuis le début du cours, nous utilisons un diode électro-luminescente située sur le circuit imprimé du robot. Cette diode est dite « bi-couleur », c'est-à-dire qu'elle peut s'allumer en rouge ou en vert, dépendamment du sens du courant qu'on lui transmet. En faisant clignoter rapidement la diode en alternant la couleur, on obtient une troisième couleur : ambrée.

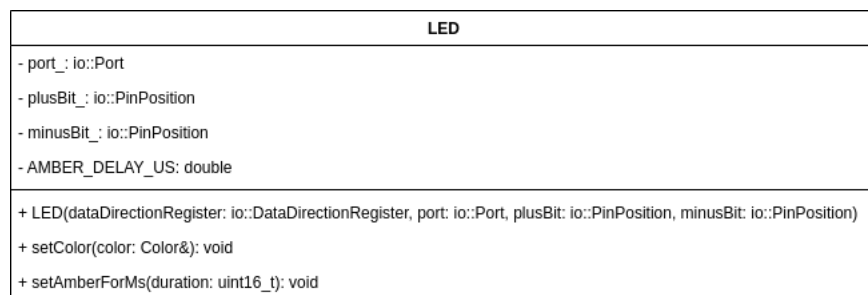


FIGURE 1 – Diagramme de classe UML de **LED**

La classe est dotée d'un constructeur, nous permettant d'indiquer où est branchée la diode.

```
LED(io::DataDirectionRegister dataDirectionRegister, io::Port port,
    const io::PinPosition plus, const io::PinPosition minus)
```

`dataDirectionRegister` indique le registre de direction des données où la diode est branchée, `port` indique son port et finalement `plus` et `minus` indique les broches sur lesquelles sont branchés la respectivement cathode et l'anode de la diode. Il est important de respecter ce sens afin de permettre à la méthode `setColor(const Color &color)` de fonctionner adéquatement.

La méthode `setColor(const Color &color)` permet d'allumer la diode en rouge ou en vert, en passant en paramètre `Color::RED` ou `Color::GREEN`.

La méthode `setAmberForMs(const uint16_t durationMs)` permet d'allumer la diode en ambrée. Nous devons passer une durée (en millisecondes) en paramètre.

En effet, puisque la diode doit alterner rapidement entre le rouge et le vert, il est impossible d'effectuer cette opération indéfiniment.

2 Modifications apportées au Makefile de départ