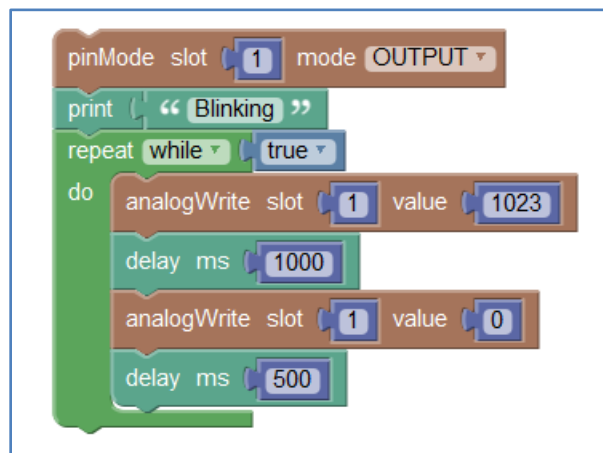
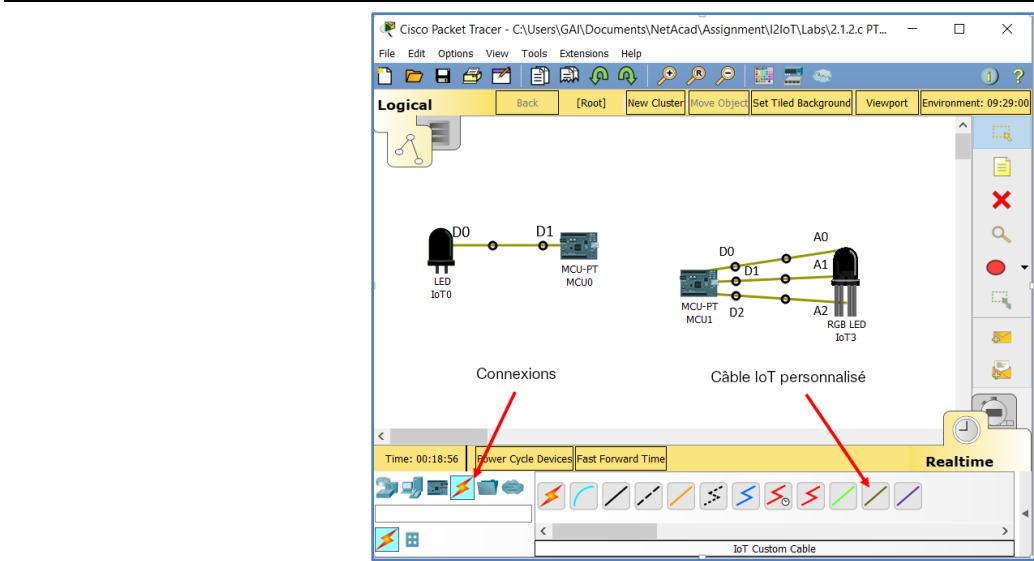


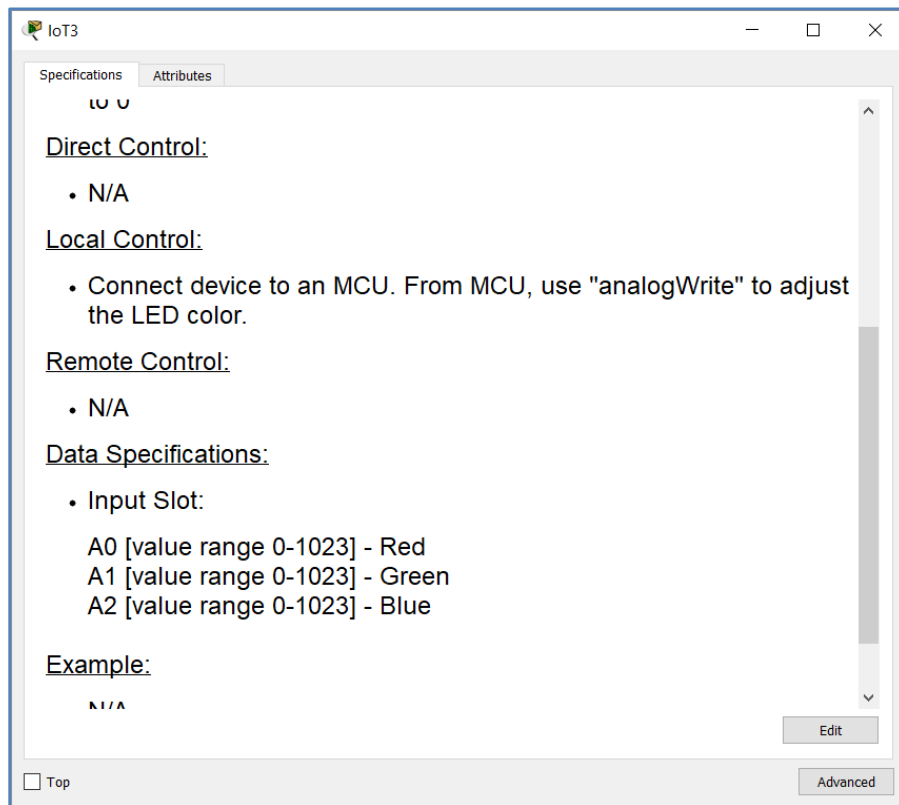
- i. Elle indique que nous pouvons utiliser « analogWrite » pour régler la luminosité de l'appareil. Développez le groupe **Pin Access** et utilisez le bloc **analogWrite** pour remplacer le bloc **digitalWrite**.



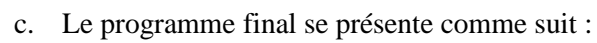
- j. À présent, modifiez les valeurs de premier (1023) et deuxième blocs (0) **analogWrite** et observez les différents niveaux de luminosité du voyant.



- c. Cliquez sur **RGB LED** et vérifiez ses caractéristiques. Notez que les entrées de broche différentes représentent différentes couleurs.



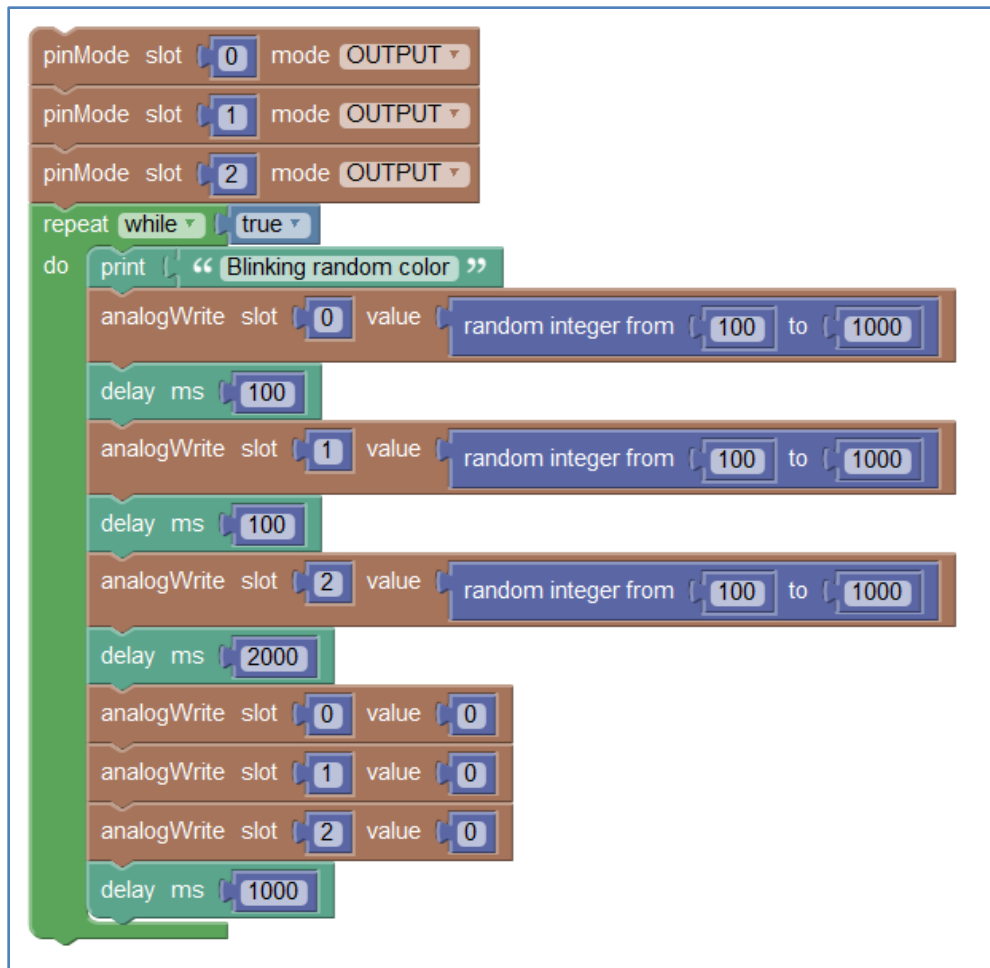
- d. Ouvrez le programme Blockly prégénéré. Cliquez sur MCU -> Programming. Sous la note **No Project Opened**, cliquez sur **New**. Dans la fenêtre **Create Project**, saisissez **Blinking RGB LED** comme nom du projet. Dans le menu déroulant Template, sélectionnez **Blink – Visual**.





- f. Faites la même chose que la dernière tâche, mais avec des valeurs différentes générées aléatoirement pour chaque logement

Voici un exemple :



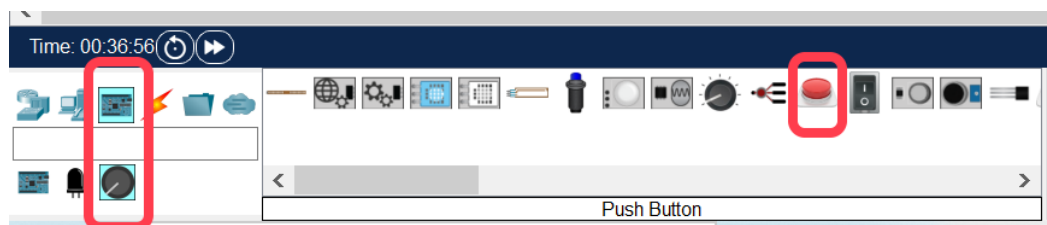
Partie 3 : Ajouter un bouton pour contrôler une LED RGB

Dans cette partie nous allons ajouter un objet IoT de type bouton pour contrôler la LED RGB que vous avez créé dans la partie précédente.

Étape 1 : Ajouter un bouton IoT

L'étape 1 consiste à créer un nouveau circuit en ajoutant un bouton et le connecter à la carte MCU

- Ajouter un nouveau **MCU**, une nouvelle **LED RGB**.
- Cliquez sur la catégorie **Components** puis **Sensors** puis **Push Button** comme il est illustré dans la figure suivante :



- c. Cliquez sur la catégorie **Connections**, sélectionnez un **IoT Custom Cables** pour lier le connecteur D3 de **MCU** avec le connecteur D0 du **Push Button**.



-
- c. Modifier ce programme pour pouvoir arrêter le clignotement immédiatement après avoir arrêté le bouton.
 - d. Pensez à d'autres scénarios