	Comment éclairer les rues des villes uniquement lorsqu'il y en a besoin ?	SNT
		Thème : Informatique embarquée et IoT
		Maquette lampadaires
Capacités attendues	Informatique embarquée	<input checked="" type="checkbox"/> Identifier des algorithmes de contrôle des comportements physiques à travers les données des capteurs, l'IHM et les actions des actionneurs dans des systèmes courants. <input type="checkbox"/> Réaliser une IHM simple d'un objet connecté. <input checked="" type="checkbox"/> Écrire des programmes simples d'acquisition de données ou de commande d'un actionneur.
	Localisation	<input type="checkbox"/> Décoder une trame NMEA pour trouver des coordonnées géographiques.
	Données structurée	<input type="checkbox"/> Identifier les principaux formats et représentations de données. <input checked="" type="checkbox"/> Distinguer la valeur d'une donnée de son descripteur. <input type="checkbox"/> Réaliser des opérations de recherche, filtre, tri ou calcul sur une ou plusieurs tables.

CONTEXTE

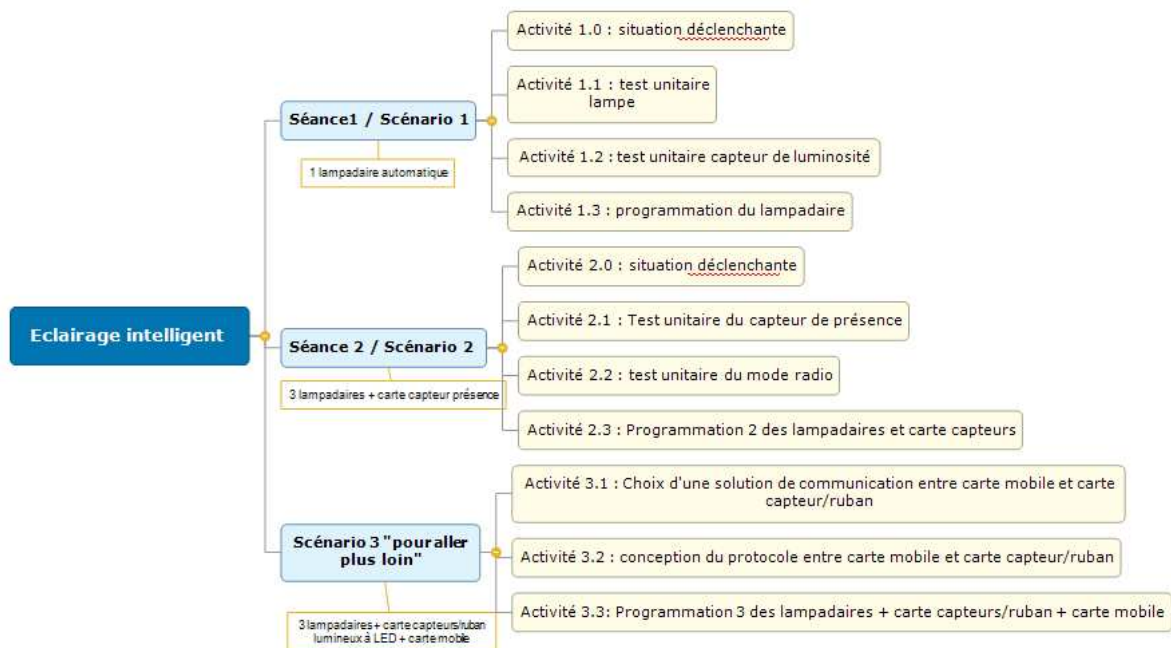
Dans le cadre de la transition énergétique et à l'heure des économies d'énergie, les villes se rendent compte que l'éclairage constant des rues la nuit, alors même que personne ne s'y trouve, engendre des dépenses énergétiques inutiles et une forte pollution lumineuse.

Votre ville souhaite équiper les rues de lampadaires intelligents à détection de passage.

Développeur informatique, vous avez en charge la programmation de l'allumage automatique de lampadaires de rue. Vous allez programmer une maquette afin de vérifier le fonctionnement prévu au cahier des charges et de présenter un prototype à votre ville, en vue d'une installation réelle à grande échelle.

DÉROULÉ

La séquence sur la maquette des lampadaires est proposée sur 2 séances de 1h30.



SITUATION DÉCLENCHANTE

Pour introduire la séquence, les élèves pourront visionner la vidéo suivante :

Exemple : <https://vimeo.com/134928585>



SEANCE 1

Question directrice : comment programmer en Python l'allumage d'un lampadaire maquetisé en fonction de la luminosité.

Activité 1.0 : Situation déclenchante

Répondez au questionnaire :

Q1 Quel est le phénomène qui doit être pris en compte pour allumer la lampe du lampadaire

Rep : la luminosité

Q2 : Donner l'action que le lampadaire doit effectuer

Rep : allumer sa lampe

Activité 1.1 : Test unitaire LAMPE

Ecrire un programme simple permettant d'allumer la lampe pendant ½ seconde puis de l'éteindre pendant ½ seconde dans une itération infinie

1.1.1/Travail préparatoire : donnez la broche (« pin ») permettant de commander la lampe.

1.1.2/ Conception : Proposer un algorithme

Rq : nous pourrions proposer de compléter l'algorithme ci-dessous :

```

Tant que ( toujours vrai) faire
Début
    AllumerLampe()
    Temporiser(500 ms)
    .....EteindreLampe()
    Temporiser(500 ms)
Fin
  
```

La fonction de déclaration de l'utilisation de la lampe (broche 0) est : `S_lampe = pin0`

La fonction permettant d'éteindre la lampe est : `S_lampe.write_digital(0)`

1.1.3/ Proposer la ligne de code en python permettant d'allumer la lampe.

1.1.4/ Ecrivez le programme, issu de votre algorithme, en langage python avec le logiciel MU.
Télécharger votre programme sur la carte microbit

1.1.4/ TESTs : Proposer une procédure permettant de montrer que votre programme répond bien au besoin.

1.1.5/ Effectuer les tests donnés dans votre procédure. Conclure

Activité 1.2 : Test unitaire CAPTEUR LUMINOSITE

Ecrire un programme simple permettant d'afficher la valeur fournie par le capteur de luminosité en pleine lumière puis en cachant le capteur.

1.2.1/Travail préparatoire : donnez la broche (« pin ») reliée au capteur de luminosité.

1.2.2/ Conception : Compléter l'algorithme ci-dessous :

```
Tant que ( toujours vrai) faire
Début
    Luminosité <- lireEntréeAnalogique(0)
    Afficher (luminosité)
    Temporiser(500 ms)
Fin
```

La fonction de déclaration de l'utilisation du capteur de luminosité (broche 1) est : `E_lum = pin1`

La fonction permettant de lire la quantité de luminosité est : `E_lum.read_analog()`

1.2.3/ Ecrivez le programme, issu de votre algorithme, en langage python avec le logiciel MU.

Télécharger votre programme sur la carte microbit

1.2.4/ TESTs : Proposer une procédure permettant de montrer que votre programme répond bien au besoin.

1.2.5/ Effectuer les tests donnés dans votre procédure. Conclure

Q Donner une valeur correspondante à la luminosité ambiante de votre salle de cours, nous prendrons cette valeur comme limite pour commander l'éclairage.

Activité 1.3 : Programmation 1 lampadaire

Capacité visée : Écrire des programmes simples d'acquisition de données ou de commande d'un actionneur.

1.3.1/ Attendus : Reportez vous à la vidéo de présentation du matériel et des attendus de votre programme

1.3.2/ Conception : En vous aidant des algorithmes précédents et en utilisant une structure algorithmique alternative (si ...alors... sinon...), proposer un algorithme permettant de répondre au besoin d'un lampadaire automatique.

1.3.3/ Ecrivez le programme, issu de votre algorithme, en langage python avec le logiciel MU.

Télécharger votre programme sur la carte microbit

1.3.4/ TESTs : Proposer une procédure permettant de montrer que votre programme répond bien au besoin.

1.2.5/ Effectuer les tests donnés dans votre procédure. Conclure

SEANCE 2

Question directrice : comment programmer en Python l'allumage d'un lampadaire maquettisé en fonction de la luminosité et en présence d'un véhicule.

Activité 2.0 : Situation déclenchante

Répondez au questionnaire :

Quel est le nouveau phénomène qui doit être pris en compte pour allumer la lampe du lampadaire en plus de celui mis en place durant l'activité précédente ?

Rep : la présence d'un véhicule

Les capteurs de présence sont reliés à une carte de communication supportant une carte microbit

Activité 2.1 : Test unitaire CAPTEUR PRESENCE

Ecrire un programme simple permettant d'afficher l'état du capteur de présence en fonction de la présence ou non d'un véhicule devant ce capteur.

2.1.1/Travail préparatoire : donnez la broche (« pin ») relié au capteur N°1

2.1.2/ Conception : Proposer un algorithme

2.1.3/ Ecrivez le programme, issu de votre algorithme, en langage python avec le logiciel MU.
Télécharger votre programme sur la carte microbit

2.1.4/ TESTs : Proposer une procédure permettant de montrer que votre programme répond bien au besoin.

2.1.5/ Effectuer les tests donnés dans votre procédure. Conclure

Activité 2.2 : Test unitaire COMMUNICATION RADIO

Ecrire un programme sur une carte microbit qui envoie en mode radio un message (« CODE ») sur une autre carte microbit.

2.2.1/ Conception : Proposer un algorithme de la carte émettrice qui envoie en boucle le mot « CODE »

2.2.2/ Conception : Proposer un algorithme de la carte réceptrice qui affiche le mot reçu.

2.2.3/ Ecrivez le programme, issu de votre algorithme, en langage python avec le logiciel MU.
Télécharger votre programme sur la carte microbit

2.2.4/ TESTs : Proposer une procédure permettant de montrer que votre programme répond bien au besoin.

2.2.5/ Effectuer les tests donnés dans votre procédure. Conclure

Activité 2.3 : Programmation 2 lampadaire

Capacité visée : Écrire des programmes simples d'acquisition de données ou de commande d'un actionneur.

2.3.1/ Attendus : Reportez vous à la vidéo de présentation du matériel et des attendus de votre programme (SCENARIO 2)

2.3.2/ **Question :** comment communiquent les lampadaires avec la carte connectée aux capteurs ?

Remarque : vous devez créer un protocole de communication entre les cartes

2.3.3/ **Question :** Proposer un schéma montrant les communications entre les différentes cartes.

Elaboration du protocole :

2.3.4/ **Question :** Proposer des phrases permettant de notifier les commandes à envoyer entre les différentes cartes

2.3.5/ **Conception :** Proposer un algorithme du programme sur la carte « CAPTEURS » qui doit détecter la présence du véhicule et qui envoie la phrase d'allumage du bon lampadaire. Cet algorithme doit permettre de gérer les 3 lampadaires.

2.3.6/ **Conception :** Proposer un algorithme du programme sur la carte lampadaire qui doit gérer le lampadaire en automatique et qui reçoit la phrase de la carte CAPTEURS.

2.3.7/ **Ecrivez les programmes,** issu de vos algorithmes, en langage python avec le logiciel MU. Téléchargez vos programmes sur les cartes microbit.

2.3.8/ **TESTs :** Proposer une procédure permettant de montrer que votre programme répond bien au besoin.

2.3.9/ **Effectuer les tests** donnés dans votre procédure. Conclure

SEANCE 3 « Pour aller plus loin »

Question directrice : comment programmer en Python l'allumage d'un lampadaire maquettisé en fonction de la luminosité et en présence du véhicule de Madame X qui porte une carte microbit.

Capacité visée : Écrire des programmes simples d'acquisition de données ou de commande d'un actionneur.

Attendus : Reportez vous à la vidéo de présentation du matériel et des attendus de votre programme (SCENARIO 3)

Q3.0/ **Question** : Proposer un schéma montrant les communications connues entre les différentes cartes du système.

Activité 3.1 : choix d'une solution de communication

3.1.1/ Proposer une solution de communication afin que le système puisse connaître la présence de Mme X.

Rep : solution radio

3.1.2/ Donner le nom de la carte connectée au ruban lumineux à LED présent sur la propriété de Mme X ?

Rep : carte capteurs

Activité 3.2 : Conception du protocole

Une solution est d'utiliser la communication radio entre la carte mobile et la carte capteur

Elaboration du protocole :

3.2.1/ Question : Proposer un message envoyé par la carte de Mme X

Activité 3.3 : Conception des programmes

Aide : Il existe dans le mode radio plusieurs canaux de communication (« channel »).

3.3.1/ Conception : Modifier l'algorithme du programme sur la carte « CAPTEURS »

3.3.2/ Conception : Proposer un algorithme du programme sur la carte de Mme X.

3.3.3/ Ecrivez les programmes, issu de vos algorithmes, en langage python avec le logiciel MU. Télécharger vos programmes sur les cartes microbit.

3.3.4/ TESTs : Proposer une procédure permettant de montrer que votre programme répond bien au besoin.

3.3.5/ Effectuer les tests donnés dans votre procédure. Conclure

BILAN – SYNTHÈSE

...

+ Synthèse version visuelle (schéma, carte mentale...).

ÉVALUATION

...

Evaluations formatives, sommatives, autonomes, QCM...