

Séquence S4 <i>Comment fonctionne une montre intelligente ?</i>	Fiche Activité – Séance N°2	Cycle 4 3ème
Compétences développées en activités : CT 4.2	Comment programmer les différentes fonctions d'une smartwatch ?	
Nom:	Prénom	Classe:



Une **smartwatch**, littéralement « montre intelligente », est une montre bracelet informatisée avec des fonctionnalités allant au-delà du simple affichage de l'heure et du chronométrage. Elle possède **plusieurs fonctionnalités**. Pour chaque fonction, une **solution technique** est choisie.

Nous allons apprendre à programmer les fonctions techniques d'une smartwatch. A partir des algorithmes ci-dessous, tu vas écrire les programmes pour la carte micro:bit, enfin de simuler ces fonctions :

a) Affichage de la température

Regarde la vidéo sur la carte micro:bit avec son capteur de température

(https://www.youtube.com/watch?v=IHhbPRVyn_c&hl=fr&cc_lang) : Clique sur paramètres, sous-titres, traduire automatiquement et choisis le français.

- 1) Clique sur l'éditeur de micro:bit pour commencer à construire ton **programme1** d'après la vidéo ci-dessus qui commence à 3:25.
- 2) Injecte ton programme1 en cliquant sur Télécharger (situé en bas et à gauche) dans la carte micro: bit.
- 3) A partir de l'algorithme ci-dessous, tu vas écrire un **programme2**

Présentation de l'algorithme de l'affichage de la température :

Si le bouton A est pressé alors la température s'affiche

Si le bouton B est pressé et :

Si la température est inférieure à 10 alors un message « couvrez-vous » s'affiche.

Si la température est comprise entre 10 et 25 alors un message « Bonne journée » s'affiche.

Si la température est supérieure à 25 alors un message « hydratez-vous » s'affiche.

Clique sur l'éditeur de micro:bit pour commencer à construire ton programme2.

- 4) Injecte ton programme2 en cliquant sur Télécharger (situé en bas et à gauche) dans la carte micro: bit.

b) Affichage de la boussole

- 1) A partir de l'algorithme ci-dessous, tu vas écrire un **programme3** pour la carte micro:bit capable d'afficher la boussole.

Présentation de l'algorithme de l'affichage de la boussole :

Si la direction de la boussole est comprise entre 0 et 45 alors afficher la flèche Nord.

Séquence S4 <i>Comment fonctionne une montre intelligente ?</i>	Fiche Activité – Séance N°2	Cycle 4
Compétences développées en activités : CT 4.2	Comment programmer les différentes fonctions d'une smartwatch ?	3ème
Nom:	Prénom	Classe:

Si la direction de la boussole est comprise entre 45 et 90 alors afficher la flèche Nord-Ouest.
Si la direction de la boussole est comprise entre 90 et 135 alors afficher la flèche Ouest.
Si la direction de la boussole est comprise entre 135 et 180 alors afficher la flèche Sud-Ouest.
Si la direction de la boussole est comprise entre 180 et 225 alors afficher la flèche Sud.
Si la direction de la boussole est comprise entre 225 et 270 alors afficher la flèche Sud-Est.
Si la direction de la boussole est comprise entre 270 et 315 alors afficher la flèche Est.
Si la direction de la boussole est comprise entre 315 et 360 alors afficher la flèche Nord-Est.

Par exemple en mettant l'orientation au nord (0°), on obtient

- 2) Injecte ton programme3 en cliquant sur Télécharger (situé en bas et à gauche) dans la carte micro: bit.

c) Affichage du compte à rebours

- 1) A partir de cet algorithme ci-dessous, tu vas écrire un **programme4** pour la carte micro:bit capable d'afficher un compte à rebours.

Présentation de l'algorithme de l'affichage du compte à rebours :

Fixer la variable « nbre_depart » à 10.

Si le bouton A est pressé alors, tant que la variable « nbre_depart » est supérieure à zéro, décrémenter la variable de 1 chaque seconde.

Clique sur l'[éditeur de micro:bit](#) pour commencer à construire ton programme4.

- 2) Injecte ton programme4 en cliquant sur Télécharger (situé en bas et à gauche) dans la carte micro: bit.
- 3) A partir de cet algorithme ci-dessous, tu vas écrire un **programme5** pour la carte micro:bit capable d'afficher un compte à rebours (pour aller plus loin).

Présentation de l'algorithme de l'affichage du compte à rebours :

Fixer la variable « nbre_depart » à 0.

Si le bouton A est pressé alors, tant que la variable « nbre_depart » est supérieure à zéro, décrémenter la variable de 1 chaque seconde.

Si le bouton B est pressé alors la variable « nbre_depart » s'incrémente de 1.

Si les boutons A et B sont pressés alors la variable « nbre_depart » est remise à 0.

Clique sur l'[éditeur de micro:bit](#) pour commencer à construire ton programme5.

- 4) Injecte ton programme5 en cliquant sur Télécharger (situé en bas et à gauche) dans la carte micro: bit.

d) Le chronomètre

- 1) A partir de cet algorithme ci-dessous, tu vas écrire un **programme6** pour la carte micro:bit capable d'afficher un compte à rebours.

Séquence S4 <i>Comment fonctionne une montre intelligente ?</i>	Fiche Activité – Séance N°2	Cycle 4
Compétences développées en activités : CT 4.2	Comment programmer les différentes fonctions d'une smartwatch ?	3ème
Nom:	Prénom	Classe:

Présentation de l'algorithme de l'affichage du chronomètre :

Fixer la variable « tps_chrono » à zéro.

Tant que le bouton A est pressé, la variable « tps_chrono » s'incrémente de 0,1 toutes les 100ms. Une fois le bouton A relâché, la variable « tps_chrono » s'affiche.

Si le bouton B est pressé, la variable « tps_chrono » est remise à zéro.

Clique sur l'[éditeur de micro:bit](#) pour commencer à construire ton programme6.

- 2) Injecte ton programme6 en cliquant sur Télécharger (situé en bas et à gauche) dans la carte micro: bit.

e) Le podomètre

- 1) A partir de cet algorithme ci-dessous, tu vas écrire un **programme7** pour la carte micro:bit capable d'afficher un podomètre.

Présentation de l'algorithme de l'affichage du podomètre :

Si le bouton A est pressé, le nombre de pas s'affiche.

Si la montre est secouée ou a la tête en bas alors la variable « pas » s'incrémente de 1.

Si les boutons A et B sont pressés, la variable « pas » est remise à zéro.

Clique sur l'[éditeur de micro:bit](#) pour commencer à construire ton programme7.

- 2) Injecte ton programme7 en cliquant sur Télécharger (situé en bas et à gauche) dans la carte micro: bit.
- 3) A partir de cet algorithme ci-dessous, tu vas écrire un **programme8** pour la carte micro:bit capable d'afficher un podomètre (pour aller plus loin).

Présentation de l'algorithme de l'affichage du un podomètre :

Si le bouton A est pressé, le nombre de pas s'affiche.

Si la montre est secouée ou a la tête en bas alors la variable « pas » s'incrémente de 1.

Si les boutons A et B sont pressés, la variable « pas » est remise à zéro.

Si le bouton B est pressé et si le nombre de pas est inférieur à 8000 alors un message s'affiche sinon un message affiche.

Clique sur l'[éditeur de micro:bit](#) pour commencer à construire ton programme8.

- 4) Injecte ton programme8 en cliquant sur Télécharger (situé en bas et à gauche) dans la carte micro: bit.

f) Distance parcourue

- 1) A partir de cet algorithme ci-dessous, tu vas écrire un **programme9** pour la carte micro:bit capable d'afficher la distance parcourue.

Présentation de l'algorithme de l'affichage de la distance parcourue :

Séquence S4 <i>Comment fonctionne une montre intelligente ?</i>	Fiche Activité – Séance N°2	Cycle 4
Compétences développées en activités : CT 4.2	Comment programmer les différentes fonctions d'une smartwatch ?	3ème
Nom:	Prénom	Classe:

Si le bouton A est pressé, la distance s'affiche.

Si la montre est secouée ou a la tête en bas alors la variable « distance » s'incrémente de 0,25.

Si les boutons A et B sont pressés, la variable « distance » est remise à zéro.

Clique sur l'[éditeur de micro:bit](#) pour commencer à construire ton programme9.

2) Injecte ton programme9 en cliquant sur Télécharger (situé en bas et à gauche) dans la carte micro: bit.

3) A partir de cet algorithme ci-dessous, tu vas écrire un programme10 pour la carte micro:bit capable d'afficher la distance parcourue (pour aller plus loin).

Présentation de l'algorithme de l'affichage de la distance parcourue :

Si le bouton A est pressé, la distance s'affiche.

Si la montre est secouée ou a la tête en bas alors la variable « distance » s'incrémente de 0,25.

Si les boutons A et B sont pressés, la variable « distance » est remise à zéro.

Si le bouton B est pressé alors la longueur du pas s'incrémente de 0.1 avec une longueur maximale de 1,2.

Si la distance est inférieure à 1000, alors la distance s'affiche en mètres sinon en kilomètres.

Clique sur l'[éditeur de micro:bit](#) pour commencer à construire ton programme10.

4) Injecte ton programme8 en cliquant sur Télécharger (situé en bas et à gauche) dans la carte micro: bit.

g) Gestion affichage

1) A partir de cet algorithme ci-dessous, tu vas écrire un programme11 pour la carte micro:bit capable de gérer l'affichage.

Présentation de l'algorithme de l'affichage du podomètre :

Si la montre est orientée vers le haut alors la variable « mode » s'incrémente de 1.

Si la variable=1 alors le message « heure » s'affiche.

Si la variable=2 alors le message « pas » s'affiche.

Si la variable=3 alors le message « distance » s'affiche.

Si la variable=4 alors le message « Température » s'affiche.

Si la variable=5 alors le message « boussole » s'affiche.

Clique sur l'[éditeur de micro:bit](#) pour commencer à construire ton programme7.

2) Injecte ton programme11 en cliquant sur Télécharger (situé en bas et à gauche) dans la carte micro: bit.

Conclusion :

A partir de l'que le professeur nous a donné, nous avons réalisé..... des fonctions avec Makecode pour Micro:bit.

J'ai appris que pour enregistrer des valeurs, il faut créer On peut faire des calculs avec les variables, ou

Séquence S4 <i>Comment fonctionne une montre intelligente ?</i>	Fiche Activité – Séance N°2	Cycle 4 3ème
Compétences développées en activités : CT 4.2	Comment programmer les différentes fonctions d'une smartwatch ?	
Nom:	Prénom	Classe: