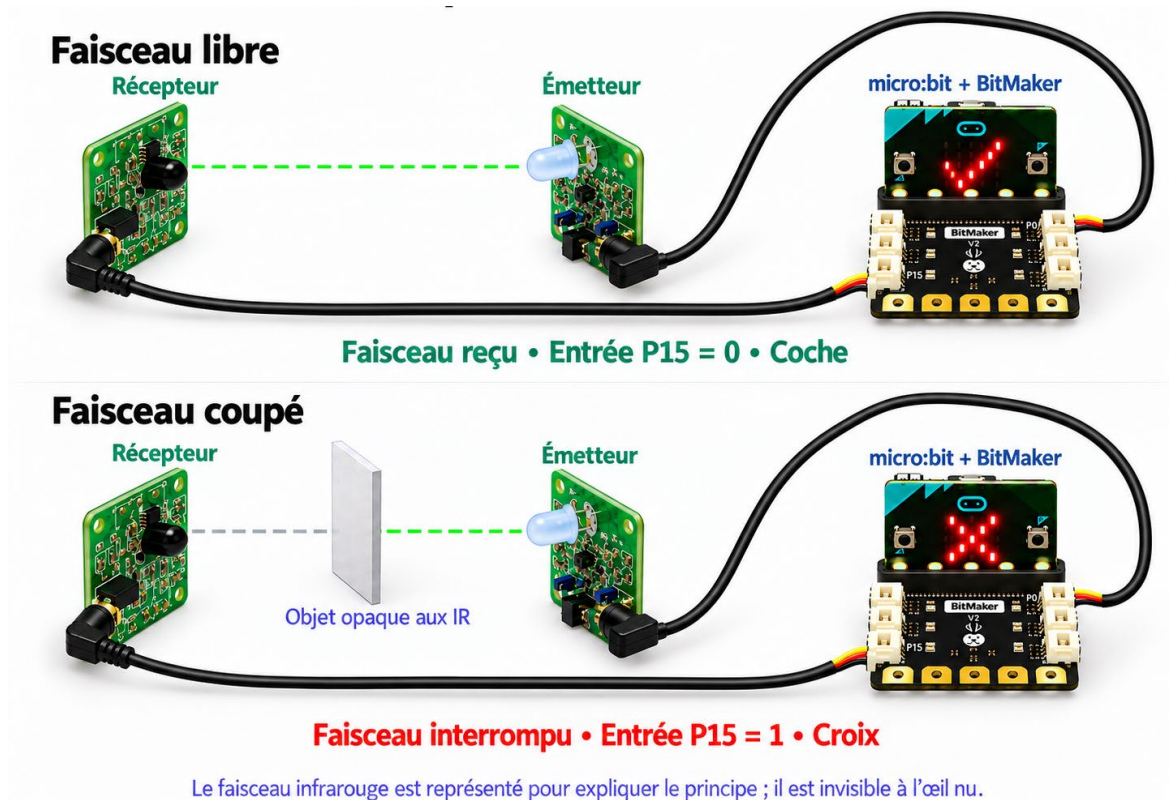


Kit universel de barrière infrarouge

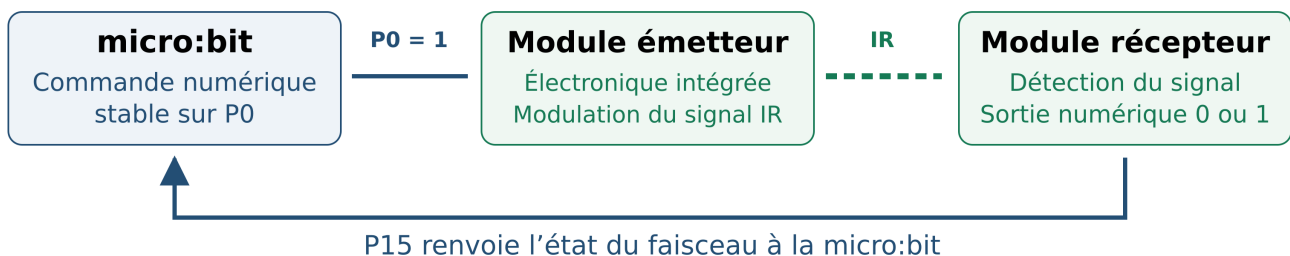
Cette notice présente l'installation du kit MI-BAR-IR avec micro:bit et un shield Grove, Arduino et un shield Grove, ou micro:bit et un shield DFR1216. Elle explique le réglage de l'émetteur et les tests de détection dans MakeCode et l'IDE Arduino.

Comprendre la détection



Lorsque l'émetteur et le récepteur sont alignés, le récepteur indique que le faisceau est présent. Si un objet opaque aux infrarouges s'interpose, la réception cesse et l'entrée change d'état. Il s'agit d'une détection par coupure de faisceau : les deux modules se placent de part et d'autre de la zone à surveiller.

Le rôle des modules AutoProg



Exemple avec micro:bit. Le même fonctionnement s'applique à Arduino : les modules gèrent le signal infrarouge.

La modulation fait varier rapidement l'émission lumineuse selon un rythme adapté au récepteur. L'électronique de l'émetteur s'en charge. Le programme micro:bit ou Arduino maintient une commande numérique active et lit l'état du récepteur, sans générer de porteuse infrarouge.

Composition et configurations compatibles

Le kit comprend deux modules IR AutoProg et deux lots de câbles. Utiliser une paire de câbles adaptée au shield choisi. La carte, le shield et leur alimentation sont à prévoir séparément.

Qté	Élément
1	Module émetteur IR AutoProg monté (K-AP-MEBIR-M)
1	Module récepteur IR AutoProg monté (K-AP-MRIR-M)
2	Câbles jack AutoProg vers Grove
2	Câbles jack AutoProg vers connecteur type servo à 3 points

Choisir les câbles et les broches

Configuration	Deux câbles à utiliser	Émetteur	Récepteur
micro:bit + shield Grove	Jack vers Grove	P0	P15
Arduino + shield Grove	Jack vers Grove	D2	D4
micro:bit + shield DFR1216	Jack vers type servo	P0	P15

Les broches ci-dessus correspondent aux exemples de cette notice. Les deux modules utilisent des broches distinctes : une sortie numérique pour l'émetteur, une entrée numérique pour le récepteur.

micro:bit avec shield Grove

Brancher un jack sur chaque module et les connecteurs Grove sur les ports numériques P0 et P15 du shield. Vérifier que le signal du câble correspond bien à la broche indiquée sur le port. Si P15 est indisponible, choisir une autre entrée et adapter le programme.

Arduino avec shield Grove

L'exemple utilise un Arduino Uno avec l'émetteur sur le port Grove D2 et le récepteur sur D4. Avec un Base Shield V2, placer le sélecteur d'alimentation sur 5 V. Le programme utilise les fonctions numériques standard de l'IDE Arduino.

micro:bit avec shield DFR1216

Utiliser les câbles jack vers type servo sur les connecteurs GPIO P0 et P15 du DFR1216. Pour chaque connecteur, faire correspondre le signal à la rangée P, l'alimentation à 3V3 et la masse à GND. Respecter ce sens de branchement.

Le terme « type servo » décrit le connecteur à trois points du câble. Utiliser les GPIO repérés P0 et P15, reliés à la micro:bit. Les ports S0 à S5 sont des sorties servomoteurs dédiées ; les ports C0 à C5 sont gérés par le contrôleur du shield et ne conviennent pas au test numérique présenté ici.

Alimenter les modules en 3,3 V avec micro:bit et le DFR1216. Une alimentation de 5 V convient avec l'Arduino Uno. Avec tout autre shield ou carte, vérifier la compatibilité des signaux ; les câbles ne réalisent aucune adaptation de niveau électrique.

Régler et brancher les modules

Effectuer les branchements hors tension, avec les câbles correspondant au shield choisi. Placer l'émetteur et le récepteur face à face, puis régler l'émetteur en mode B (Barrière infrarouge) et CODE 127.

Réglages à choisir sur l'émetteur

MODE
B • Barrière infrarouge

CODE
127 • Émission si P0 = 1

Repérer MODE et CODE sur la sérigraphie de la carte.

Choisir les positions des cavaliers indiquées par les repères MODE et CODE sur la carte de l'émetteur.

Avec CODE 127, une commande à 1 active l'émission, sur P0 avec micro:bit ou D2 avec Arduino dans les exemples proposés. CODE 126 inverse cette commande. Le mode T correspond à un autre usage, la télécommande infrarouge.



Les deux câbles transmettent aussi l'alimentation et la masse.

Exemple de raccordement avec micro:bit et un shield Grove. Avec le DFR1216, conserver P0 et P15 en utilisant les câbles jack vers type servo. Avec Arduino, utiliser D2 et D4 comme indiqué à la page précédente.

Module	Broche utilisée	Fonction
Émetteur	P0	Sortie numérique maintenue à 1
Récepteur	P15	Entrée numérique lue par le programme

Alimentation : le schéma illustre une liaison Grove. Avec le DFR1216, utiliser les rangées 3V3 et GND des GPIO. Avec l'Arduino Uno, utiliser l'alimentation 5 V du shield Grove.

Mise en service : commencer avec un écartement de 5 à 30 cm, aligner les modules et lancer le test. Le récepteur renvoie 0 quand le faisceau est reçu et 1 quand il est coupé ou absent. Un mauvais alignement produit donc également une valeur 1.

Programmer la micro:bit dans MakeCode

Utiliser le fichier HEX de test

Si le fichier microbit-BARRIERE-IR-AutoPROG.hex est disponible avec vos ressources, ouvrir MakeCode pour micro:bit, choisir Importer et le sélectionner. On peut aussi le copier sur le lecteur MICROBIT. Le code ci-dessous permet de créer le test sans ce fichier.

Le fichier HEX d'origine affiche un cœur au démarrage, active l'émetteur sur P0 et lit le récepteur sur P15. Ce raccordement convient aux deux shields micro:bit présentés. La croix correspond à une valeur 1 ; la coche correspond à une valeur 0.

État du faisceau	Valeur lue sur P15	Affichage
Reçu	0	Coche
Coupé ou absent	1	Croix

Créer le programme de test

Dans MakeCode, créer un projet, ouvrir JavaScript et coller le code ci-dessous. Il convient au shield Grove et aux GPIO P0 et P15 du DFR1216. Il actualise la matrice uniquement quand l'état change. Aucune extension n'est nécessaire.

```
// MI-BAR-IR : émetteur P0, récepteur P15.  
// Émetteur réglé sur MODE B et CODE 127.  
pins.digitalWritePin(DigitalPin.P0, 1)  
let etatPrecedent = -1  
  
basic.forever(function () {  
    let coupure = pins.digitalReadPin(DigitalPin.P15)  
    if (coupure != etatPrecedent) {  
        etatPrecedent = coupure  
        if (coupure == 1) {  
            basic.showIcon(IconNames.No, 0)  
        } else {  
            basic.showIcon(IconNames.Yes, 0)  
        }  
    }  
})
```

Les trois opérations à retenir

Maintenir P0 à 1, lire P15 et choisir l'affichage selon la valeur reçue. Avec une autre entrée, remplacer DigitalPin.P15 dans le code par la broche réellement utilisée. L'émission reste active ; une lecture régulière permet de détecter les coupures.

Programmer Arduino avec un shield Grove

Cet exemple utilise un Arduino Uno, un shield Grove alimenté en 5 V, l'émetteur sur D2 et le récepteur sur D4. Régler l'émetteur sur MODE B et CODE 127. Aucune bibliothèque complémentaire n'est nécessaire.

Créer et transférer le programme

Ouvrir l'IDE Arduino, créer un croquis et coller le code ci-dessous. Sélectionner la carte et son port USB, puis téléverser le programme. Ouvrir le moniteur série à 9 600 bauds.

```
// MI-BAR-IR : emetteur D2, recepteur D4.
// Emetteur regle sur MODE B et CODE 127.
const int BROCHE_EMETTEUR = 2;
const int BROCHE_RECEPTEUR = 4;
int etatPrecedent = -1;

void setup() {
  pinMode(BROCHE_EMETTEUR, OUTPUT);
  pinMode(BROCHE_RECEPTEUR, INPUT);
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(BROCHE_EMETTEUR, HIGH);
  Serial.begin(9600);
}

void loop() {
  int coupure = digitalRead(BROCHE_RECEPTEUR);
  if (coupure != etatPrecedent) {
    etatPrecedent = coupure;
    digitalWrite(LED_BUILTIN, coupure);
    if (coupure == HIGH) {
      Serial.println("Faisceau coupe ou absent : 1");
    } else {
      Serial.println("Faisceau recu : 0");
    }
  }
}
```

Vérifier la détection

État du faisceau	Valeur lue sur D4	LED intégrée de l'Uno
Reçu	0 (LOW)	Éteinte
Coupé ou absent	1 (HIGH)	Allumée

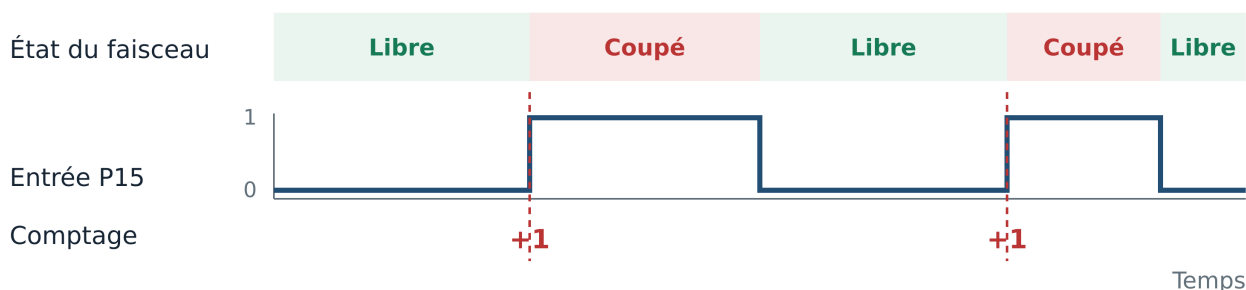
Passer un objet opaque entre les modules. La LED s'allume et le moniteur série signale la coupure. Retirer l'objet : la LED s'éteint. Pour utiliser d'autres ports numériques, modifier les deux constantes au début du code et le raccordement correspondant.

Exploiter la détection dans une activité

Compter un passage

Pendant une coupure, l'entrée peut rester à 1 pendant plusieurs lectures. Pour compter une seule fois, repérer le passage de 0 à 1. Attendre que le faisceau soit de nouveau reçu avant de compter une nouvelle coupure. Initialiser l'état précédent avec la valeur réellement lue pour éviter un comptage artificiel au démarrage.

Un niveau indique une coupure qui dure



Chronogramme illustré avec P15 sur micro:bit. Le même principe s'applique à D4 sur Arduino : chaque début de coupure peut incrémenter le compteur une seule fois.

Pistes pédagogiques

- Observer les deux états d'une entrée numérique et construire une condition dans MakeCode ou l'IDE Arduino.
- Compter des pièces ou des passages, afficher le total ou le transmettre par radio avec micro:bit, ou par la liaison série avec Arduino.
- Déclencher une signalisation ou un actionneur dans une maquette automatisée.

Intégrer le kit dans un programme plus complet

La modulation est produite sur le module émetteur. Maintenir sa sortie de commande active et lire régulièrement le récepteur, avec micro:bit comme avec Arduino. Une coupure plus courte que l'intervalle entre deux lectures peut être manquée.

Utiliser ensemble l'émetteur et le récepteur IR AutoProg du kit. Les câbles Grove permettent de raccorder ces modules aux shields Grove. L'emploi d'autres modules IR peut nécessiter un programme différent. Le kit ne détermine pas le sens du passage et ne mesure pas une distance.

Documents et références

Référence du kit : MI-BAR-IR. Composition : modules K-AP-MEBIR-M et K-AP-MRIR-M, deux câbles jack vers Grove et deux câbles jack vers type servo. Les illustrations présentent l'exemple micro:bit avec shield Grove.

[Module émetteur AutoProg monté](#)

[Module récepteur AutoProg monté](#)

[Câble Grove vers AutoProg version simple](#)

[Documentation A4 du module émetteur et de ses réglages](#)

[Documentation A4 du module récepteur](#)

[Documentation DFRobot du shield DFR1216 et de ses connecteurs](#)

[Documentation Seeed Studio du Grove Base Shield V2 pour Arduino](#)