

Document réponse 1 – Montage 2

1. Traitement combinatoire permettant d'obtenir le fonctionnement attendu (voir le sujet)

led = bp

2. Programmation

Code du programme LED2 à implanter dans le microcontrôleur.

CORRECTION

```

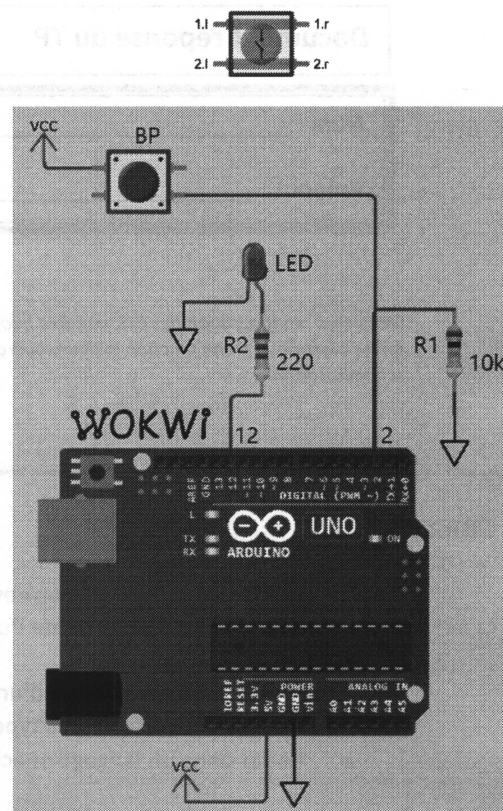
// Titre : LED2
// Fonctionnement attendu : la LED s'allume lorsqu'on presse le
// bouton-poussoir (BP), elle s'éteint lorsqu'on le relâche.

// Entrées, sorties
#define LED 12 // Sortie - Broche associée à la LED
#define BP 2 // Entrée - Broche associée au BP

void setup() // Ce code s'exécute une fois
{ // L'accolade ouvrante marque le Début du bloc de code
  // Configuration des E/S
  pinMode(LED, OUTPUT);
  pinMode(BP, INPUT);
} // L'accolade fermante marque la Fin du bloc de code

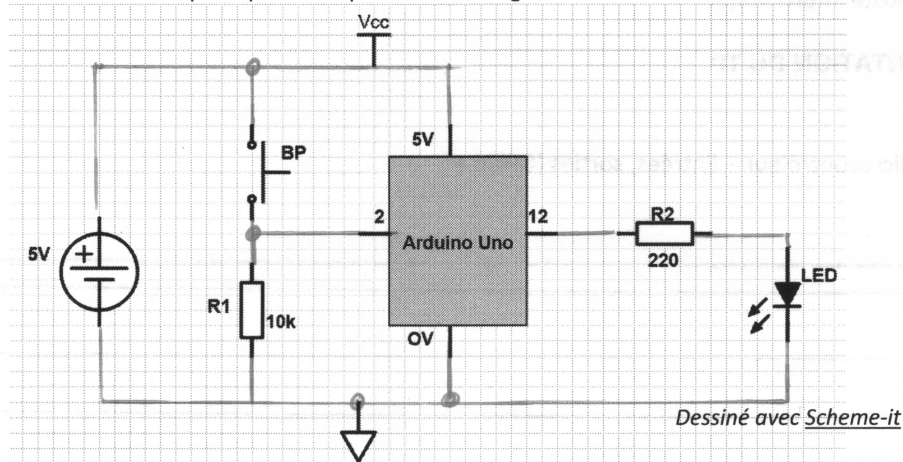
void loop() { // Ce code se répète indéfiniment
  // Lecture de l'entrée
  bool etat_bp = digitalRead(BP);
  // Ecriture de l'entrée sur la sortie
  digitalWrite(LED, etat_bp);
}

```



3. Schématisation

Complétez le schéma structurel ci-dessous pour qu'il corresponde au montage 2.



Quel nom donne-t-on à la résistance R1, quelle est son rôle ? Pull-down, force l'état logique 0 car si le BP n'est pas activé.

Document réponse 2 – Montage 3

1. Traitement combinatoire permettant d'obtenir le fonctionnement attendu (voir le sujet)

led = bp

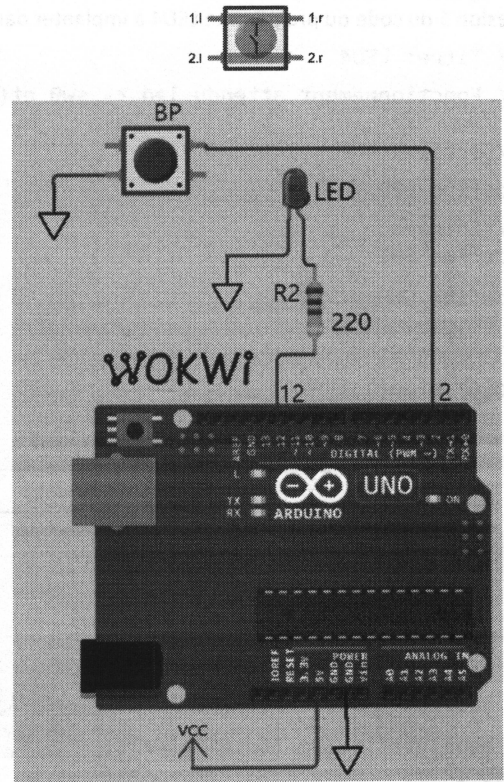
2. Programmation

Code du programme LED3 à implanter dans le microcontrôleur.

```
// Titre : LED3
// Fonctionnement attendu : la LED s'éclaire lorsqu'on presse
// le bouton-poussoir (BP), elle s'éteint lorsqu'on le relâche.
// Entrées, sorties
#define LED 12 // Sortie - Broche associée à la LED
#define BP 2 // Entrée - Broche associée au BP

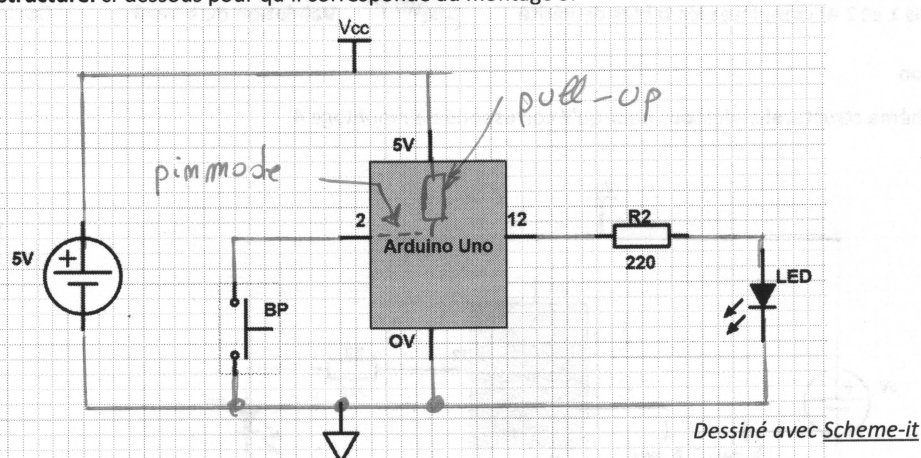
void setup()
{
  // Configuration des E/S
  pinMode(LED, OUTPUT);
  pinMode(BP, INPUT_PULLUP);
}

void loop() {
  // Lecture de l'entrée
  bool etat_bp = !digitalRead(BP);
  // Ecriture de l'entrée sur la sortie
  digitalWrite(LED, etat_bp);
}
```

**Remarque :** Il semble que ce montage ne réponde pas au fonctionnement attendu ;Que remarquez-vous ? le comportement du montage est inversé par rapport au montage 2**Solution :** placez un ! devant digitalRead. (Expliqué en cours)

3. Schématisation

Complétez le schéma structurel ci-dessous pour qu'il corresponde au montage 3.



Où se situe la résistance de pull-up, comment est-elle connectée au bouton-poussoir ? La résistance se situe dans le µC de la carte Arduino. Elle est connectée au bouton poussoir à l'aide du logiciel

Document réponse 3 – Montage 4

1. Traitement combinatoire permettant d'obtenir les fonctionnements attendus (voir le sujet)

Version 1 : led = SW0 ET SW1Version 2 : led = SW0 OU SW1

2. Programmation

Version 1 du code du programme LED4 à implanter dans le microcontrôleur.

```

// Titre: LED4
// Fonctionnement attendu led <- sw0 et(OU) sw1

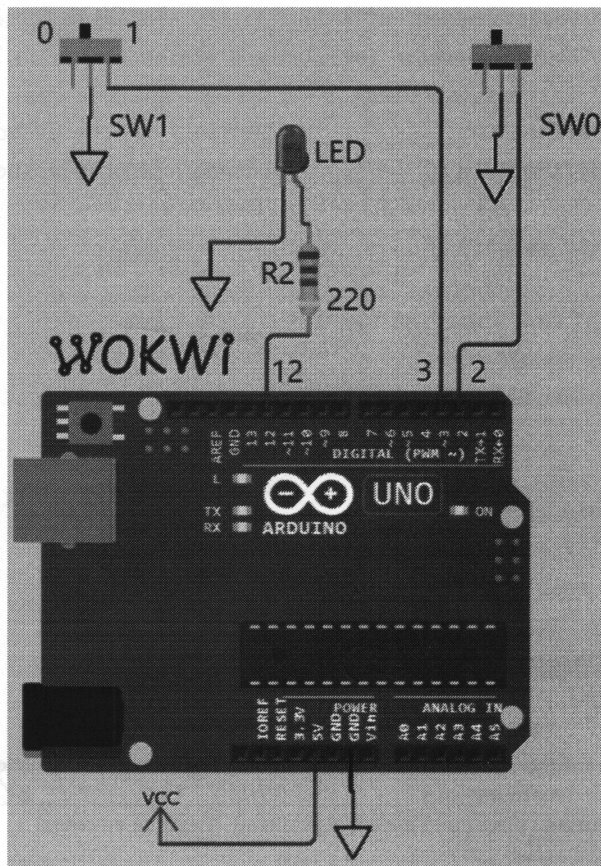
//Sortie
#define LED 12

// Entrées
#define SW1 3
#define SW0 2

void setup() {
  pinMode(LED, OUTPUT);
  pinMode(SW1, INPUT_PULLUP);
  pinMode(SW0, INPUT_PULLUP);
}

void loop() {
  // Lecture des entrées
  bool etat_sw0 = !(digitalRead(SW0));
  bool etat_sw1 = !(digitalRead(SW1));
  // Traitement
  bool etat_led = etat_sw0 && etat_sw1;
  // Ecriture sur la sortie
  digitalWrite(LED, etat_led);
}

```



Testez les versions 1 et 2 et complétez les tables de vérité

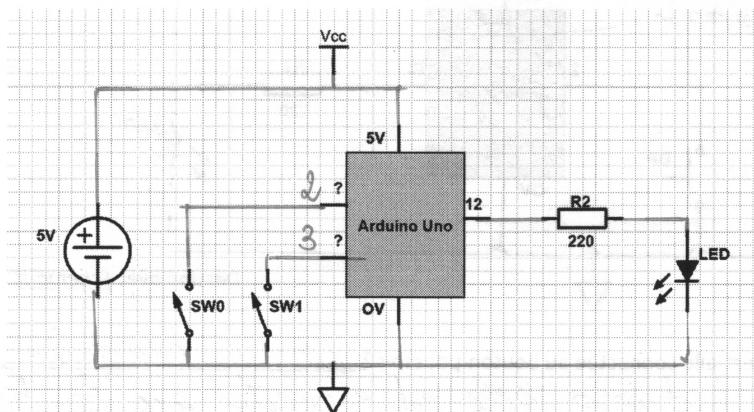
Appel prof

Validation du travail

Oui ☐ Partiel ☐ Non ☐

3. Schématisation

Complétez le schéma structurel ci-dessous pour qu'il corresponde au montage 4.

Dessiné avec Schema-it

Tables de vérité

Version 1

SW1	SW0	LED
0	0	0
0	1	0
1	0	0
1	1	1

Op. logique: ET

Version 2

SW1	SW0	LED
0	0	0
0	1	1
1	0	1
1	1	1

Op. logique: OU

Document réponse 4 – Accès de Parking (synthèse)

Description du fonctionnement attendu

Un parking souterrain est géré grâce à un gardien et à partir de capteurs de détection de véhicules.

Un capteur P dans le sol détecte la présence d'un véhicule à l'entrée du parking (p = "1").

Un capteur H en hauteur détecte la présence d'un véhicule de plus de 2 mètres (h = "1").

Pour une hauteur supérieure à 2 mètres l'entrée dans le parking est interdite.



De plus le gardien du parking a la possibilité de fermer un contact G (g = "1") si le parking est plein, pour ne pas autoriser l'entrée de véhicules supplémentaires.

Remarque : p, h et g sont les variables booléennes associées à P, H et G.

L'autorisation de pénétrer est visualisée sur un feu bicolore :

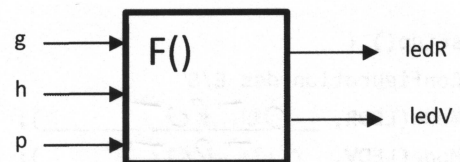
- Si le feu est vert, la barrière s'ouvre et le véhicule peut pénétrer dans le parking.
- Si le feu est rouge, la barrière reste fermée.

1. Traitement combinatoire

a. Complétez la table de vérité de la fonction logique à réaliser F().

Remarques

- « 1 » : feu allumé, « 0 » : feu éteint
- ledV : feu vert
- ledR : feu rouge



Pour les combinaisons matériellement impossibles ou si un véhicule n'est pas présent, le feu rouge restera allumé. Les cas impossibles seront notés X dans la table de vérité.

g	h	p	ledR	ledV
0	0	0	1	0
0	0	1	0	1
0	1	0	X	0
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	X	0
1	1	1	1	0

X = 0 0 1

b. En déduire ledR=f(g,h,p) et ledV=f(g,h,p).

$$\text{ledV} = \bar{g} \bar{h} p$$

$$\text{ledR} = \bar{g} \bar{h} \bar{p} + \bar{g} h \bar{p} + \bar{g} h p + g \bar{h} \bar{p} + g \bar{h} p + g h \bar{p} + g h p$$

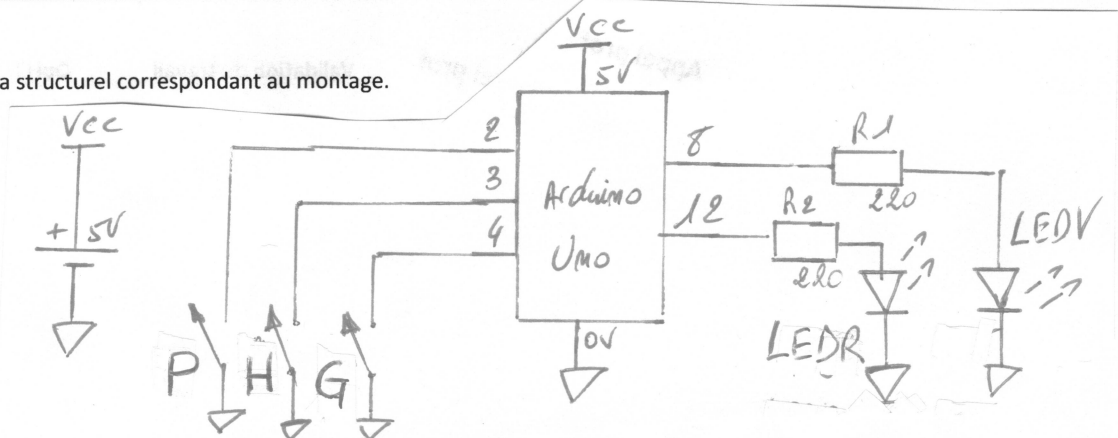
$$\text{ledR} = \bar{g} \bar{h} \bar{p} \Rightarrow \text{ledR} = \bar{g} \bar{h} \bar{p} = g + h + \bar{p}$$

Ouvrez le schéma créé sur Wokwi à partir du lien suivant

<https://bit.ly/3qFckIU>

2. Schématisation

Dessinez le schéma structurel correspondant au montage.



3. Programmation

Complétez le code du programme

```

/* Titre : Accès de parking
Matériels : Arduino Uno, Led RGB, 3 switchs sur Pull-Up
interne
*/

// Entrées, sorties
#define LEDR 12 // Feu s'éclaire en rouge
#define LEDV 8. // Feu s'éclaire en vert
#define G 4 // Parking plein
#define H 3 // Hauteur véhicule > 2m
#define P 2 // Présence véhicule

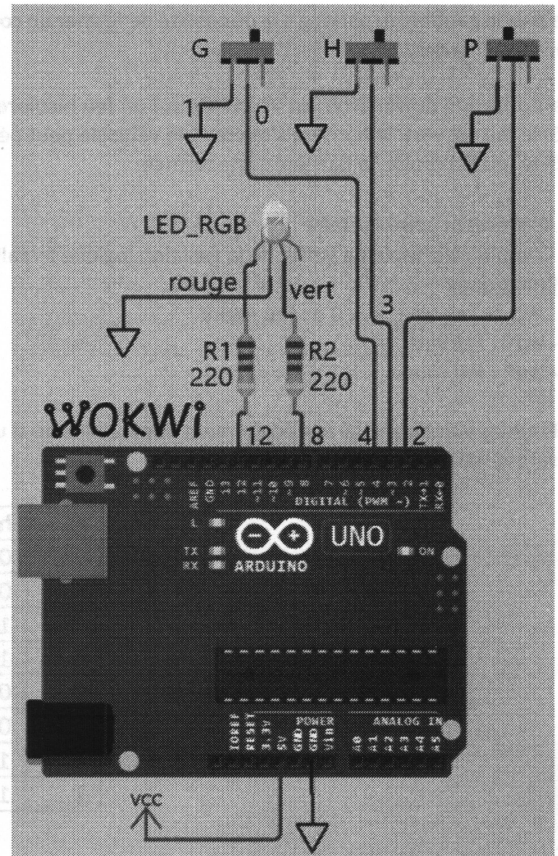
void setup() {
  // Configuration des E/S
  pinMode(LEDR, OUTPUT);
  pinMode(LEDV, OUTPUT);
  pinMode(G, INPUT_PULLUP);
  pinMode(H, INPUT_PULLUP);
  pinMode(P, INPUT_PULLUP);
}

void loop() {
  // Lecture des entrées
  bool g = !digitalRead(G);
  bool h = !digitalRead(H);
  bool p = !digitalRead(P);

  // Traitement
  bool ledV = !g && !h && p;
  bool ledR = !(!g && !h && p);

  // Ecriture sur les sorties
  digitalWrite(LEDR, ledR);
  digitalWrite(LEDV, ledV);
}

```



Appel prof

Validation du travail

Oui ☐ Partiel ☐ Non ☐