

- PB5(13) en sortie (LED Arduino),
 - PD2-PD5 en entrée avec pull-up interne,
 - PC0(A0) en sortie,
 - PC1(A1) en entrée.
3. Avec une configuration correcte, la fonction loop suivante doit conduire à ce que la LED rouge s'allume quand le bouton bleu est enfoncé, et éteinte sinon. Tester.

```
void loop()
{
  if((PIND&0x08)==0) PORTB|=1;
  else PORTB&=~0x01;
}
```

Exercice 2 (Interruptions)

1. Ajouter une fonction ISR (poly 6.1 page 14) sur l'interruption INT1(PD3) pour qu'elle ait lieu sur front négatif. Dans cette fonction, on inversera l'état de la led rouge.
2. Assurer que cette fonction ISR est exécutée de façon asynchrone en mettant le code suivant dans la fonction loop.

```
void loop()
{
  PORTB^=0x20; // bascule la sortie PB5(13) (led arduino)
  delay(500);
}
```

On doit observer que peu importe quand l'on appuie sur le bouton jaune, la fonction ISR est appelée. Ce n'est pas du tout synchrone avec la fonction loop.

3. Ajouter une fonction ISR sur l'interruption INT0(PD2) pour qu'elle ait lieu sur front négatif, modifier les ISR pour que l'appui sur le bouton violet allume la led rouge et que le bouton jaune éteigne la led rouge.
4. Pinchange est un autre mécanisme d'interruption. N'importe quelle broche du microcontrôleur peut l'utiliser. L'idée est d'associer une ISR à chaque changement d'état d'une ou de plusieurs entrées. Mettre en place l'ISR liée à l'interruption PinChange (cf p16) pour les entrée PD2-PD5 (les 4 BP). Dans cette ISR, basculer l'état de la led rouge si au moins un des 4 BP est enfoncé.

Note : vous devrez, pour cette question, retirer les ISR INT0/INT1.

Exercice 3 (Mesure de distance)

Le capteur SR04 réalise des mesures de distance. Pour faire une mesure, on doit créer une impulsion de quelques microsecondes sur l'entrée trigger du capteur. Ensuite, le capteur fait une mesure et restitue la mesure sous forme d'une impulsion sur la broche echo. La durée de l'impulsion est proportionnelle à la distance mesurée (vous pouvez voir l'impulsion sur l'oscillo). Il est possible de modifier l'obstacle du capteur SR04.

```
void Trigger(){ // fonction pour générer le trigger
PORTC|=0x01; // début trigger
delayMicroseconds(10);
PORTC&=~0x01; // fin trigger
}
```

1. Utiliser l'interruption PinChange sur l'entrée PC1 pour détecter le début et la fin de l'impulsion echo.
2. Mesurer la durée de l'impulsion en utilisant les fonctions de chronomètre d'Arduino (millis())
3. Modifier le programme pour qu'à chaque appui sur l'un des 4 BP, une mesure de distance soit faite. Si un obstacle est à moins de 1m, allumer la led rouge.

Note : Proportionnalité mesure SR04 :
durée impulsion (micros) / 58 = distance (cm).