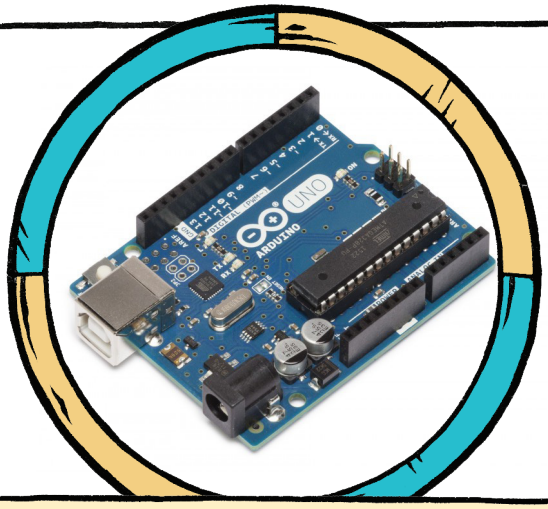
OUTIL



FICHES ASSOCIÉES

- entrées analogiques

FAIRE DES MESURES RAPIDES

Jusqu'à quelle fréquence peut-on faire des mesures ?

Taux d'échantillonnage : 1 kEch/s ; 9 kEch/s ; 55 kEch/s
Difficulté d'utilisation : moyen à compliqué
Utilisations possibles : enregistrer des données rapides

MÉTHODE CONTINUE

On veut des mesures en continu, et les transférer le plus vite possible à l'ordinateur. La méthode la plus simple est d'utiliser le câble USB pour ce transfert, mais la communication entre la carte et l'ordinateur va prendre du temps, et diminuer la fréquence maximale de travail. Le moniteur série peut travailler à différentes vitesses de communication, il faut choisir des vitesses rapides si un taux d'échantillonnage élevé est nécessaire.

```
void setup() {
  Serial.begin(115200) ;
}
```

La carte Arduino doit être configurée pour utiliser la même vitesse de communication que le moniteur série.

```
void loop() {
  Serial.print(millis()) ;  // on mesure et on transmet le temps
  Serial.print("\t") ;      // caractère tabulation pour séparer les données
  Serial.println(analogRead(A0)) ;  // on mesure et on transmet la tension
}                                   // appliquée sur le port A0
```

Ce programme peut tourner indéfiniment, et envoie les résultats au fur et à mesure sur le port série. Le taux d'acquisition est de l'ordre de 1,2 kEch/s quand on utilise une vitesse de communication de 115200 bauds. En utilisant la vitesse de communication standard de 9600 bauds, la fréquence de travail descend à 90 Ech/s.

MÉTHODE PAR PAQUETS

Quand les mesures sont immédiatement transmises à l'ordinateur, le temps de transfert ralentit la fréquence de travail. On peut économiser ce temps de transfert en faisant d'abord les mesures, puis en les transférant par paquets ensuite. On gagne alors en vitesse, mais on perd la continuité des mesures. La taille réduite de la mémoire des cartes Arduino Uno devient alors limitante.

Le programme ci-dessous permet de travailler à 9 kHz environ. La limite devient donc la durée que dure le paquet de mesure. En effet, la mémoire de l'arduino Uno est limitée, et au-delà de 800 integers sauvegardés, on risque de déborder. Dans le programme ci-dessus, on a 400 échantillons à environ 9 kHz, soit environ un temps de mesure de 45 ms.

```
const int NombreDeMesures = 400 ;
int Mesures[NombreDeMesures] ;  // tableau de mesures
int Temps[NombreDeMesures] ;    // tableau de temps

void setup() {
  Serial.begin(9600) ;        // la vitesse de transfert peu être basse
}

void loop() {
  for(int i = 0 ; i < NombreDeMesures ; i++ ){ // on fait les mesures
    Mesure1[i] = analogRead(A0) ;
    Temps[i] = millis() ;
  }
  for(int i = 0 ; i < NombreDeMesures ; i++ ){ // on transfert les mesures
    Serial.print(millis()) ;
    Serial.print("\t") ;
    Serial.println(analogRead(A0)) ;
  }
}
```

- Certaines cartes Arduino ont des taille de mémoires plus importantes ;
- On peut économiser la place occupée par les mesures de temps en n'enregistrant pas le temps à chaque mesure, mais uniquement le temps initial et le temps final.

HACK DU CAN

Avec la méthode précédente, la limite est la vitesse de travail du convertisseur analogue-numérique de la carte. Cette vitesse peut être augmentée au prix d'une légère dégradation de la qualité de la mesure. Le lien suivant décrit très bien la méthode :
https://sites.google.com/site/measuringstuff/the-arduino
Le programe suivant exécute 800 mesures à 55 kEch/s

```
#define FASTADC 1
#ifndef cbi
#define cbi(sfr, bit) (_SFR_BYTE(sfr) &= ~_BV(bit))
#endif
#ifndef sbi
#define sbi(sfr, bit) (_SFR_BYTE(sfr) |= ~_BV(bit))
#endif
const int NombreDeMesures = 800 ;
int Mesures[NombreDeMesures] ;  // tableau de mesures
void setup() {
  Serial.begin(9600) ;        // la vitesse de transfert peu être basse
#if FASTADC
  sbi(ADCSRA,ADPS2) ;
  cbi(ADCSRA,ADPS1) ;
  cbi(ADCSRA,ADPS0) ;
#endif
}
void loop() {
  unsigned long TempsInitial = micros() ;
  for(int i = 0 ; i < NombreDeMesures ; i++ ){ // on fait les mesures
    Mesure1[i] = analogRead(A0) ;
  }
  unsigned long TempsFinal = micros() ;
  for(int i = 0 ; i < NombreDeMesures ; i++ ){ // on transfert les mesures
    Serial.println(analogRead(A0)) ;
  }
  Serial.println(TempsFinal - TempsInitial) ; // durée des 800 mesures
}
```