

Capteur de Force, Jauges

Bluetooth

USB

GPS

Ethernet

Alim 12V

Potentiomètre

Bus I2C

Poussoir

Servo-moteur Variateur CC Brushless

CAN

Mesure courant ± 5A

MOS N

Relais 2RT

Moto-réducteurs (codeur incrémental)

Collecteurs ouverts

Ecran tactile

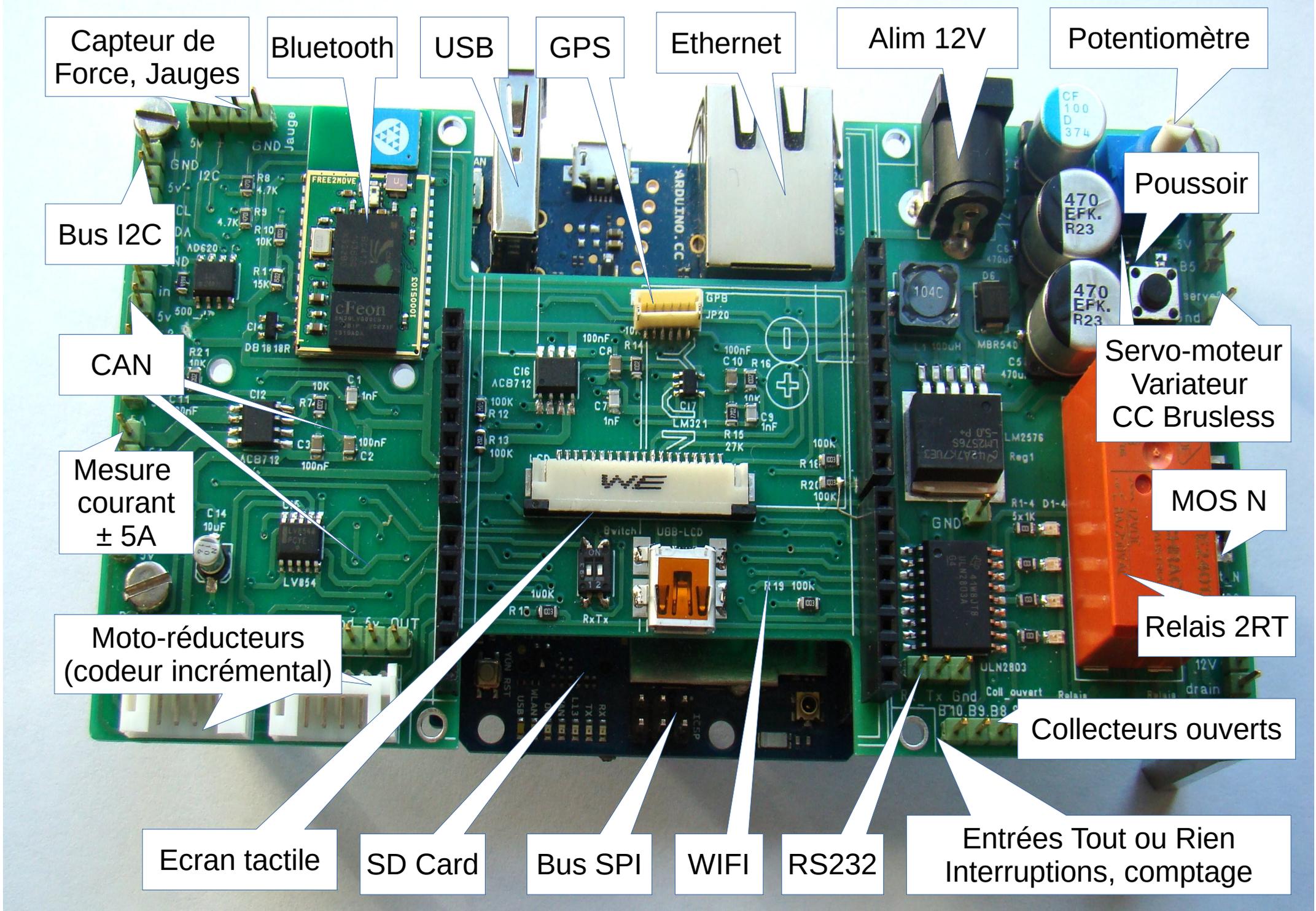
SD Card

Bus SPI

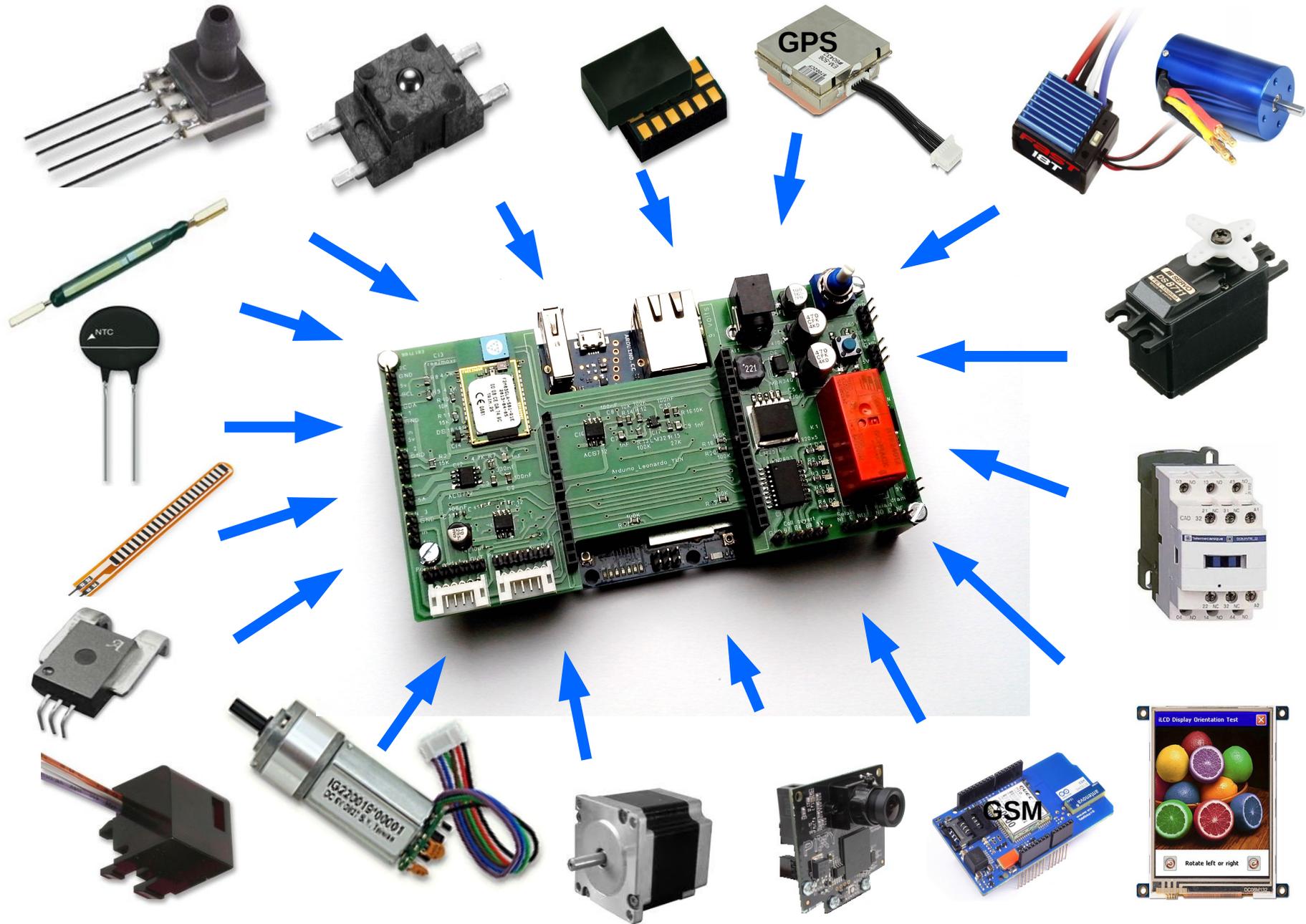
WIFI

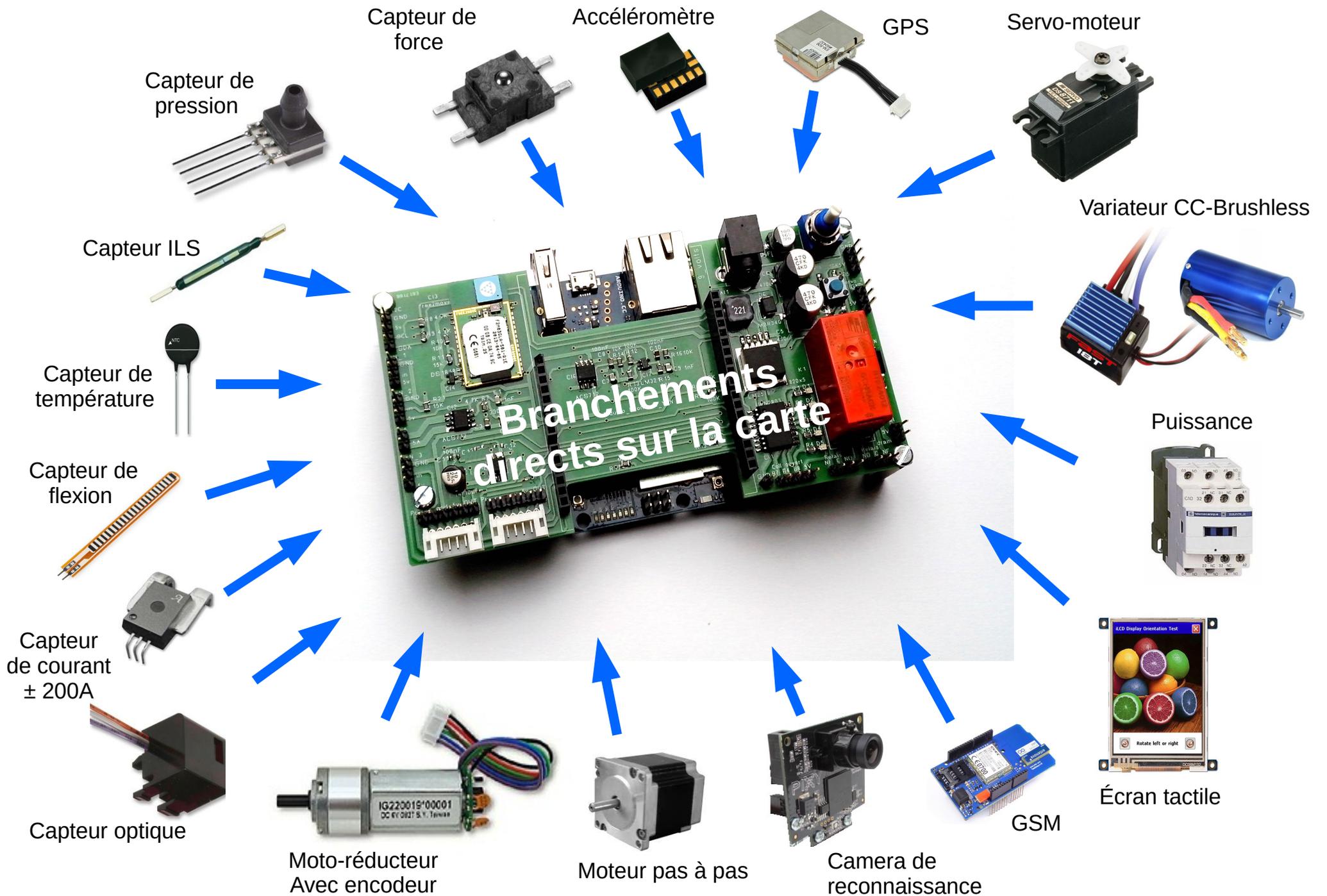
RS232

Entrées Tout ou Rien Interruptions, comptage

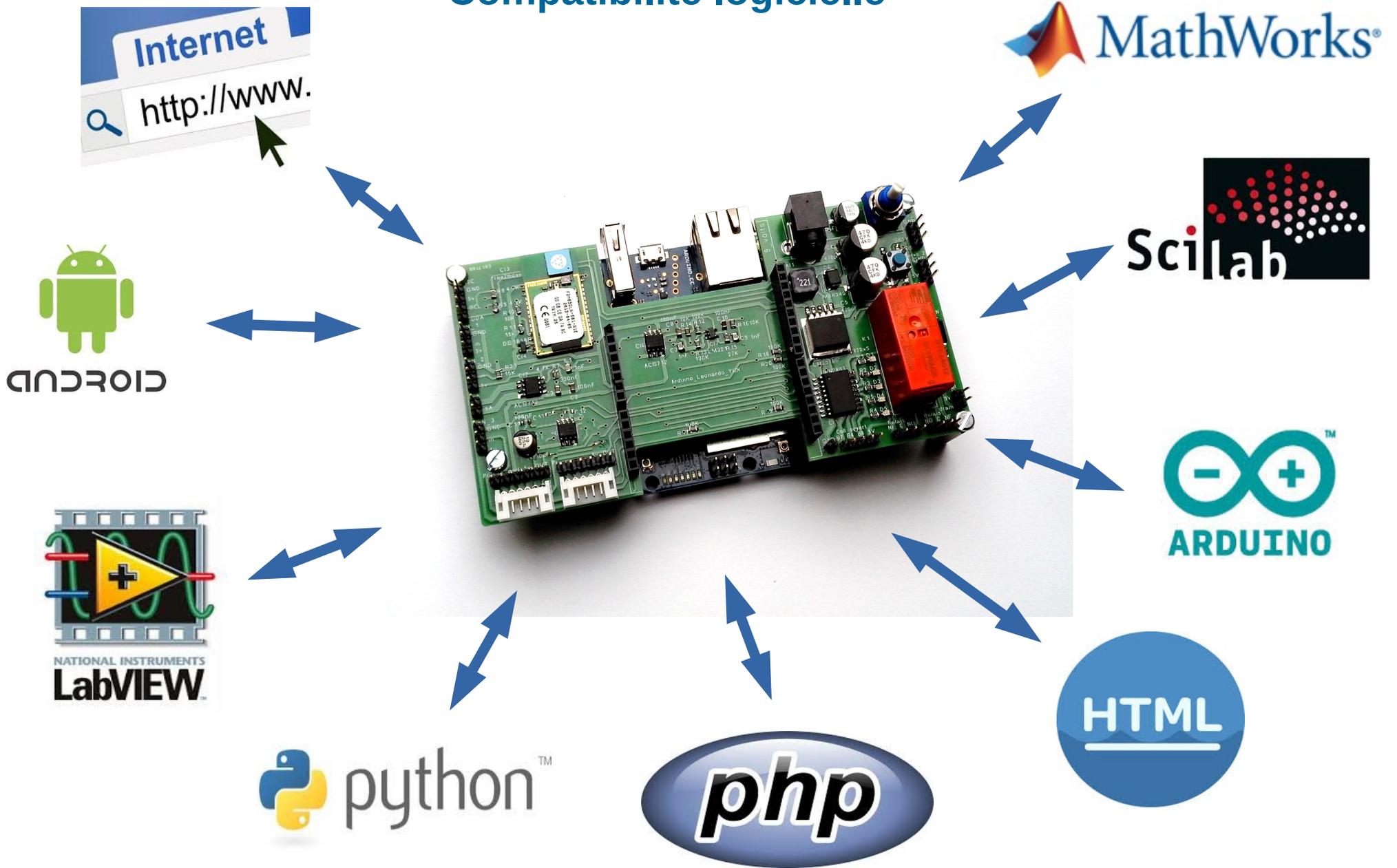


Branchement direct sur la carte

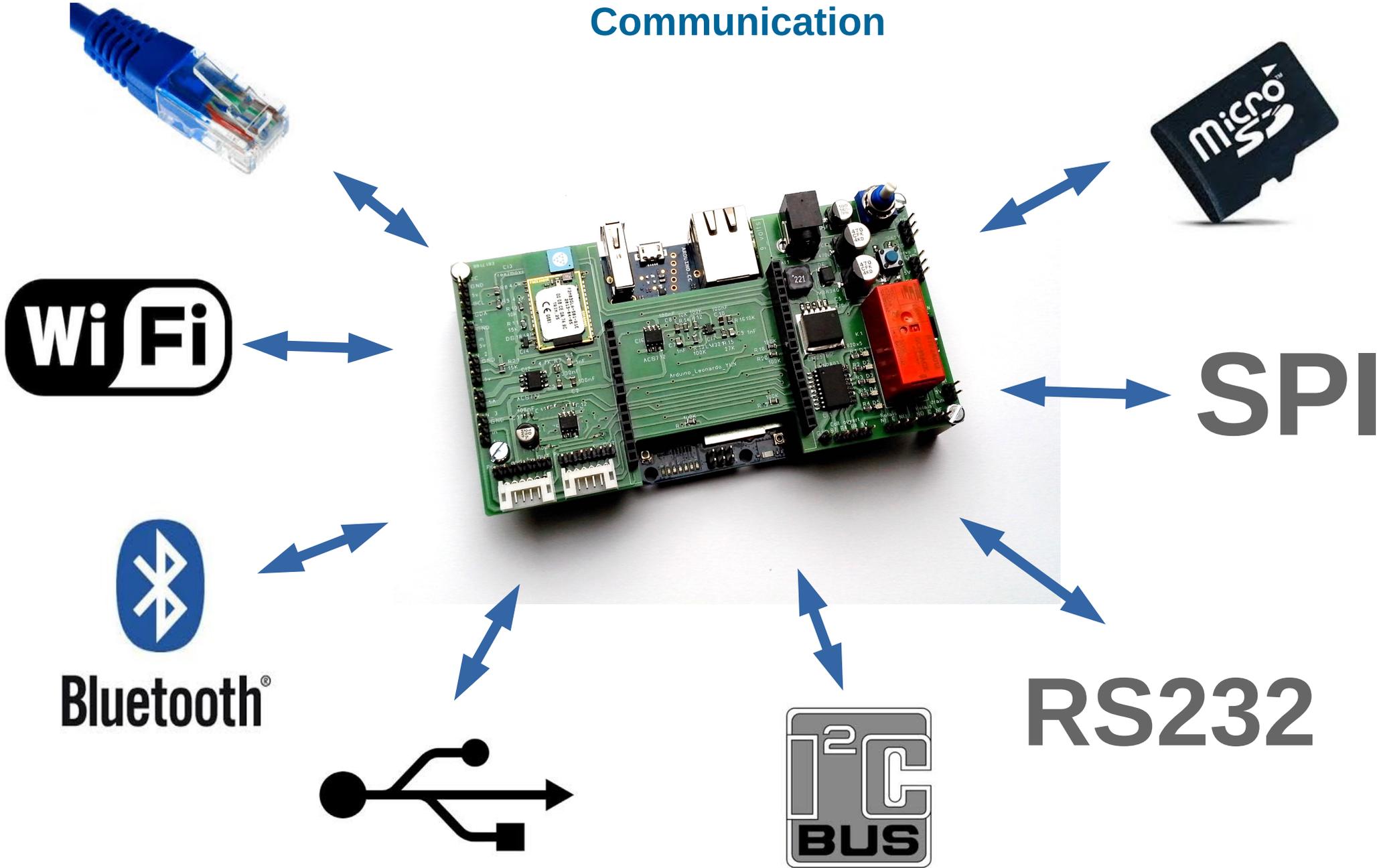




Compatibilité logicielle



Communication



Fonctionnalités de la carte

La carte CREArduino a deux objectifs pédagogiques.

- Fournir un outil simple pour apprendre la programmation des micro-systèmes avec un grand choix d'applications
- Faciliter la mise en œuvre des projets par les élèves. Elle doit permettre de répondre à toutes les problématiques du projet sans avoir à fabriquer des cartes électroniques spécifiques.

Un point fort de cette carte est un ensemble de fiches pédagogiques simples qui décrivent les principaux cas d'utilisation et qui vont permettre aux élèves de mettre facilement en œuvre un dispositif de mesure et de contrôle sur leur projet.

Le second point fort est de pouvoir connecter directement les composants standard du commerce (capteurs, moteurs, afficheur, GPS etc...)

On peut directement y connecter les composants du commerce sans avoir besoin de circuits complémentaires,

Elle peut communiquer avec des interfaces logicielles sur le PC comme Labview, Matlab ou Scilab pour visualiser les mesures et piloter le système étudié. Il est ainsi facile de faire un asservissement de vitesse avec le moteur fourni.

Elle peut communiquer avec des applications Android par la liaison Bluetooth.

Elle fonctionne avec un Arduino Leonardo, Yun ou Mega, Equipée d'un Arduino Yun, elle peut être reliée au réseau par Ethernet ou par WIFI, elle peut aussi servir de point d'accès WIFI autonome. Elle peut alors héberger des applications Web qui exploitent les entrées et sorties.

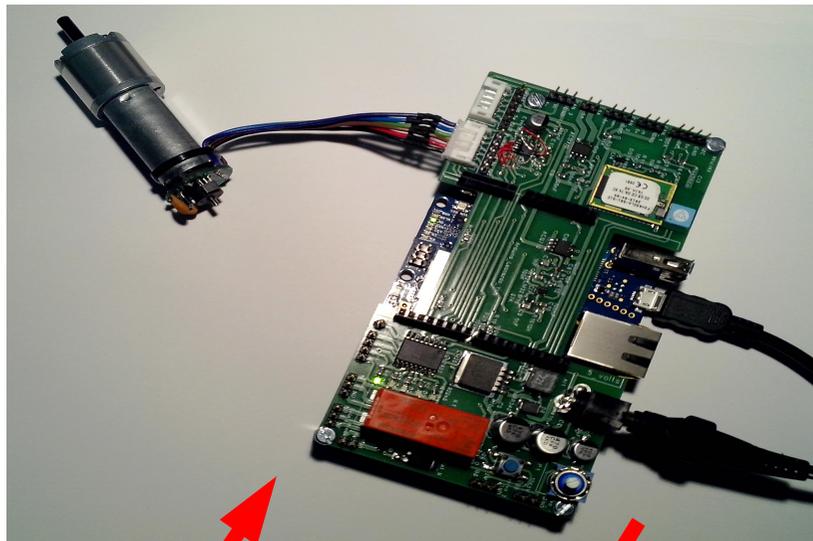
La carte Arduino est programmée en C++, la partie Linux peut être programmée en Python ou en PHP (YUN).

Elle peut être équipée d'un afficheur couleur tactile programmable avec des commandes simples.

Les connexions possibles sont multiples et des exemples sont fournis (attention, toutes ne sont pas exploitables simultanément).

- Un potentiomètre et un bouton poussoir permanents sur la carte
- Des entrées logiques tout ou rien avec comptage et interruptions
- Des entrées analogiques pour tout capteur alimenté en 0 – 5V
- Une entrée pour capteur résistif (thermistance, flexomètre, photo-résistance ...)
- Une entrée différentielle pour jauge de contrainte (capteur d'effort)
- Trois sorties à collecteurs ouverts pour faibles courants, quatre diodes.
- Une sortie relais avec double contacts repos travail
- Un capteur de courant $\pm 5A$ pour un circuit externe
- Deux branchements pour moto-réducteur avec encodeur et mesure de courant
- Une sortie à transistor MOS N pour charge quelconque (moteur, résistance, lampe ...)

Exemple d'asservissement de vitesse avec Labview



Labview gère l'asservissement

La carte Arduino pilote le moteur avec la consigne demandée et retourne la vitesse mesurée,



Fiches pédagogiques

Fiche N°1 : chenillard à LED

Cette activité consiste à...

Fiche N°11 : Commande de servo-moteur

Cette activité consiste à alimenter un servo-moteur utilisé le plus souvent pour obtenir simplement une variation angulaire d'un mécanisme. Le schéma de la structure utilisée. Chaque servo-moteur est simplement relié sur la sortie dédiée avec une alimentation 5V et une sortie PWM. La consigne de position est fournie ici par le potentiomètre.

Le programme Arduino

```
#include <Servo.h>
Servo myservo; // crée le servo
int potPin = A0; // pin de lecture du potentiomètre
int servoPin = 9; // pin de commande du servo

void setup() {
  myservo.attach(servoPin); // attache le servo
  pinMode(potPin, INPUT); // configure le pin du potentiomètre
}

void loop() {
  int val = analogRead(potPin); // lit la valeur du potentiomètre
  float potVal = val / 1023.0; // convertit en valeur de 0 à 1
  float servoVal = map(potVal, 0, 1, 0, 180); // convertit en angle de 0 à 180 degrés
  myservo.write(servoVal); // écrit l'angle sur le servo
  delay(10); // pause de 10 ms
}
```

CREAproject

Fiche N° : Mesures de pression

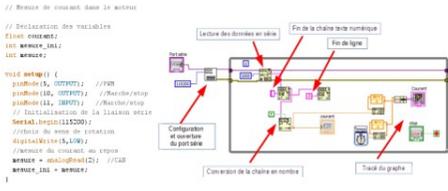
Le capteur utilisé ici est un capteur de pression absolu

CREAproject

Fiche N°10 : Mesure du courant dans Labview

Le programme Labview permet de tracer l'évolution du courant dans le moteur. Le bouton poussoir est ici utilisé pour démarrer et arrêter le moteur.

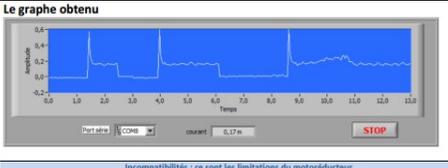
Les programmes Arduino et Labview



```
// Mesure de courant dans le moteur
// Déclaration des variables
float mesure;
int mesure_val;
int mesure;

void setup() {
  pinMode(4, OUTPUT); //PWR
  pinMode(10, OUTPUT); //Moteur stop
  pinMode(12, OUTPUT); //Moteur stop
  // Initialisation de la liaison série
  Serial.begin(115200);
  // Choix du mode de lecture
  digitalWrite(5, LOW);
  // Choix du courant de sortie
  pinMode(12, OUTPUT); //PWR
  pinMode(10, OUTPUT); //Moteur stop
}

void loop() {
  // Lecture de la valeur du courant
  mesure = analogRead(A0); //PWR
  mesure_val = mesure * 27 / 1023.0; //Quantité 100mA, 0V, 10 bits
  Serial.print(mesure_val);
  delay(10);
}
```



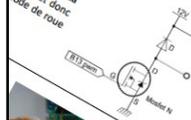
Incompatibilités : ce sont les limitations du motoreducteur.

CREAproject

Fiche N°9 : Commande moteur à courant continu

Cette activité consiste à...

... un moteur à courant continu avec la sortie MOS. La consigne de vitesse est fournie par le potentiomètre.



Une rotation de 180° est obtenue en inversant les bornes IN+ et IN- du moteur.

CREAproject

Fiche N°10 : Commande d'un moteur pas à pas

Cette activité consiste à...



Une rotation de 180° est obtenue en inversant les bornes IN+ et IN- du moteur.

Attention, dans cet exemple...