Carte d'extension

http://developer.android.com/tools/adk/index.html

ADK

Le kit de développement d'accessoires (ADK) est une API qui permet de communiquer avec un carte électronique avec un micro-contrôleur qui est compatible avec le protocole accessory.

Disponible depuis mai 2011 sur Android, Google a initialisé le développement avec la carte Arduino Google Arduino Mega2560 en fournissant tout le code en open source. Par la suite de nombreuses cartes avec micro-contrôleur ont également suivi le même chemin en offrant une API compatible avec l'ADK Google.

Accessory utilise Android Open Accessory (AOA) protocole pour communiquer avec les périphériques Android, via un câble USB ou via une connexion Bluetooth.

Il existe deux kits de développement :

ADK 2012 Guide
ADK 2011 Guide

http://developer.android.com/tools/adk/index.html

ADK

La carte ATMEGA2560 dispose :

- d'une mémoire flash interne de 256 Ko
- un processeur 8-bits et il fonctionne à 16MHz.
- Il fournit 8 Ko de SRAM et EEPROM 4KB.

Les 16 ports d'entrée-sorties analogiques de la carte fournissent une résolution sur 10 bits de 0 à 5 v. La puce dispose de 54 broches numériques avec 14 d'entre elles étant PWM (modulation de largeur d'impulsion).

lib ADK --> https://dl-ssl.google.com/android/adk/adk_release_0512.zip

ADK

ADK Boards	Google ADK	Arduino ADK	Seeeduino ADK	Sparkfun IOIO
Processor	ATmega2560 ATmega	2560	ATmega2560 PIC	24FJ256
CPU clock speed	16 MHz	16 MHz	16 MHz	32 MHz
Flash memory	256 Kbytes	256 Kbytes	256 Kbytes	256 Kbytes
RAM	8 Kbytes	8 Kbytes	8 Kbytes	96 Kbytes
Digital IO pins	54 (14 PWM)	54 (14 PWM)	56 (14 PWM)	48 (28 PWM)
Analog input pins	16	16	16	16
Input voltage	5.5V - 16V	5.5V - 16V	6V - 18V	5V - 15V
Connectors DC	power USB A-type USB micro B-type	DC power USB A-type USB B-type	DC power USB A-type USB micro B-type	USB A-type

ADK

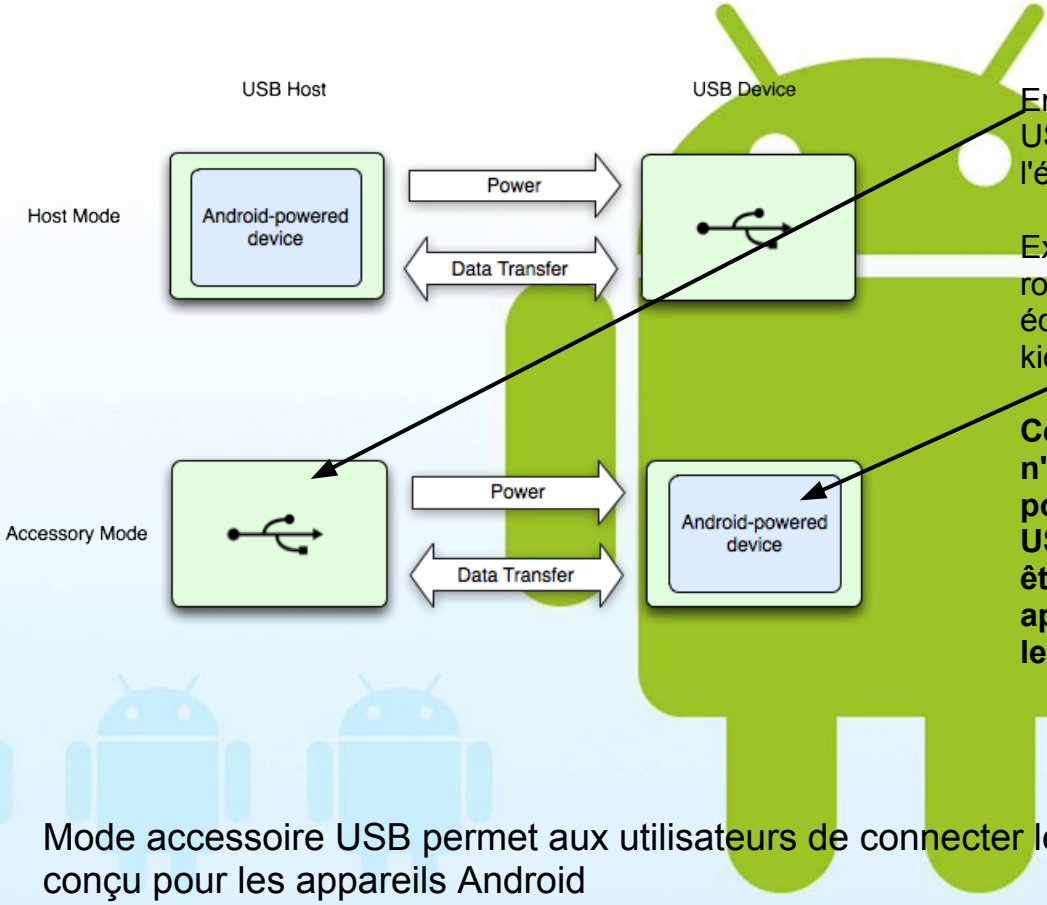
Le protocole veut que l'accessoire suive quatre étapes de base pour créer un communication avec l'appareil Android:

1. L'accessoire est en état d'attente et tente de détecter tous les périphériques connectés.
2. L'accessoire de support pour vérifier le mode de l'appareil accessoire.
3. L'accessoire tente de mettre l'appareil en mode accessoire si cela est nécessaire.
4. Si le périphérique prend en charge le protocole accessoire Android, l'accessoire établit la communication

L'accessory mode permet un appareil Android qui n'a pas la capacité USB host de communiquer avec un matériel externe, qui à son tour agit en tant que partie USB host.

La spécification de la norme d'accessoiry stipule que l'hôte USB doit fournir de l'énergie pour le bus USB et peut énumérer les périphériques connectés.

Mode accessory



En mode USB accessory, le matériel USB externe agit comme USB host et l'équipement Android en USB device.

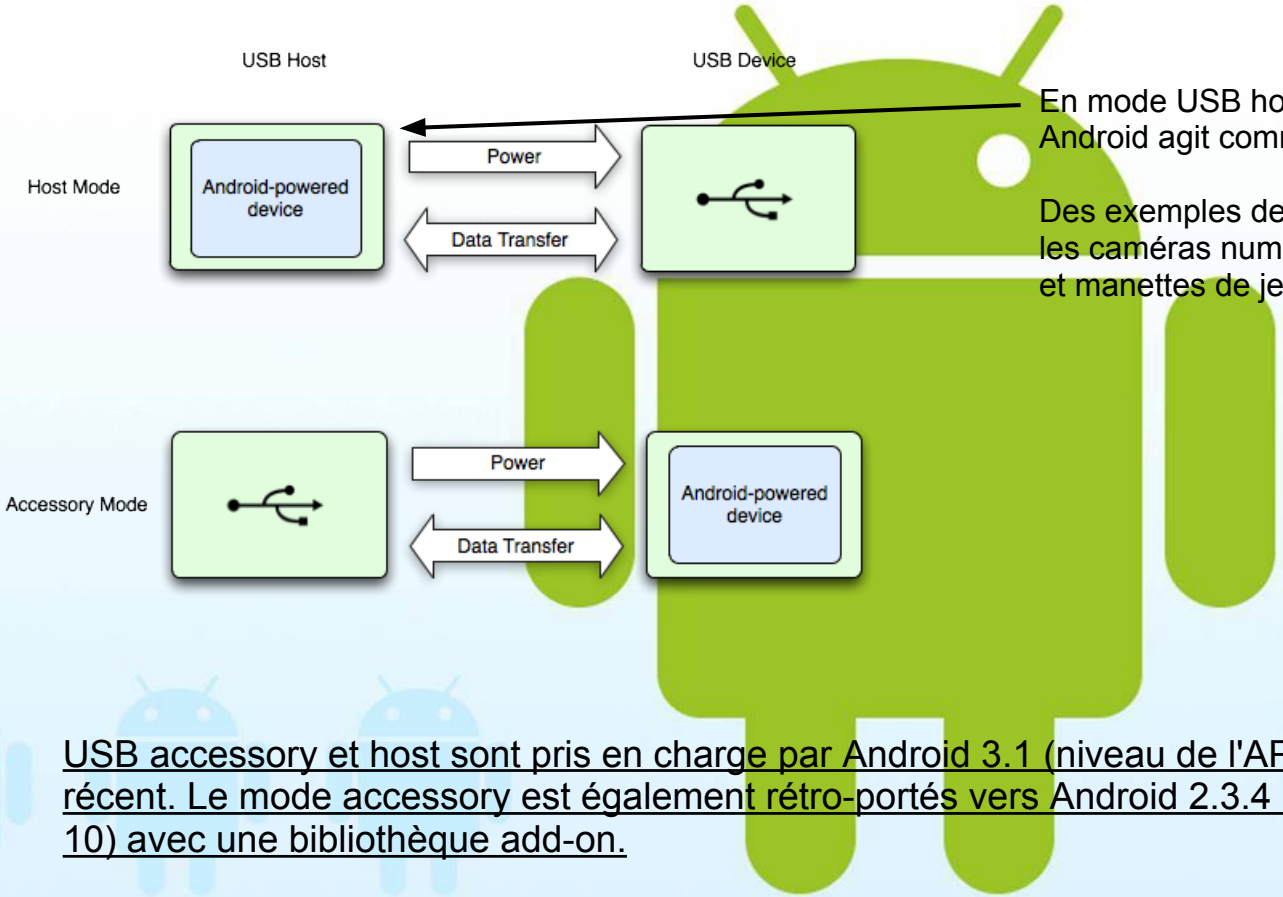
Exemples d'accessoires : les contrôleurs robotiques, une station d'accueil, des équipements de diagnostic et musicales; kiosques; lecteurs de cartes.

Cela permet aux appareils Android qui n'ont pas les capacités USB host la possibilité d'interagir avec le matériel USB. Les accessoires USB doivent être conçus pour fonctionner avec les appareils Android et doivent respecter le protocole de communication ADK.

Mode accessoire USB permet aux utilisateurs de connecter le matériel hôte USB spécialement conçu pour les appareils Android

ADK

Mode USB host

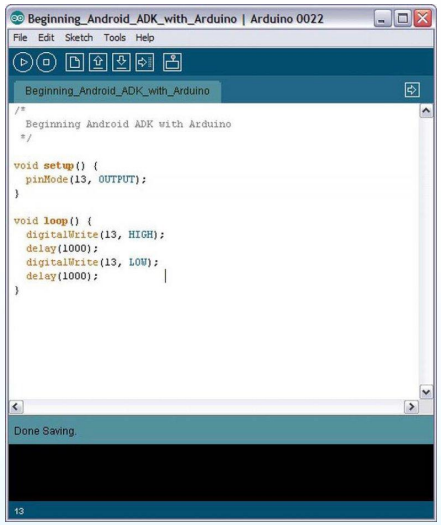


En mode USB host, le périphérique sous Android agit comme host.

Des exemples de dispositifs comprennent les caméras numériques, claviers, souris et manettes de jeu.

USB accessory et host sont pris en charge par Android 3.1 (niveau de l'API 12) ou plus récent. Le mode accessory est également rétro-portés vers Android 2.3.4 (API de niveau 10) avec une bibliothèque add-on.

ADK



The Arduino IDE http://arduino.cc/en/Main/Software

Pour installer votre environnement de développement :

1 : installer l'IDE en fonction du système
www.arduino.cc/playground/Learning/Linux

Exemple sur Fedora :

sudo yum install arduino
yum install uisp avr-libc avr-gcc-c++ rxtx avrdude

2 : Placer la lib ADK dans le répertoire librarie de votre projet, redémarrer l'IDE et vérifier que dans Sketch->import library vote llib est bien présente.
La lib = les deux répertoires qui se trouve dans l'archive dans /ADK_release_0512/firmware/arduino_libs/ (la lib --> https://dl-ssl.google.com/android/adk/adk_release_0512.zip)

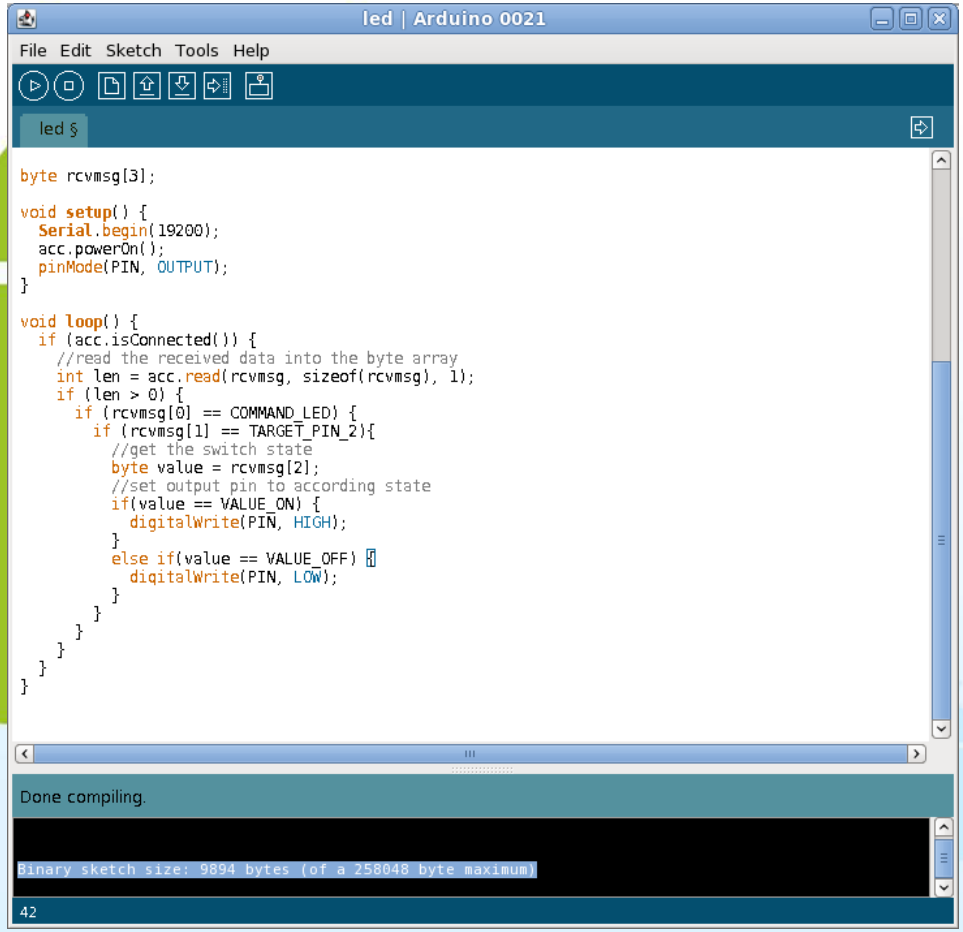
3 Lancer une compilation ...

Sur Ubuntu : http://playground.arduino.cc//Linux/Ubuntu

ADK

```
#include <Usb.h>
#include <AndroidAccessory.h>
#define COMMAND_LED 0x2
#define TARGET_PIN_2 0x2
#define VALUE_ON 0x1
#define VALUE_OFF 0x0
#define PIN 2

AndroidAccessory acc("Manufacturer",
"Project01",
"Description",
"Version",
"URI",
"Serial");

byte rcvmsg[3];

void setup() {
  Serial.begin(19200);
  acc.powerOn();
  pinMode(PIN, OUTPUT);
}

void loop() {
  if (acc.isConnected()) {
    //read the received data into the byte array
    int len = acc.read(rcvmsg, sizeof(rcvmsg), 1);
    if (len > 0) {
      if (rcvmsg[0] == COMMAND_LED) {
        if (rcvmsg[1] == TARGET_PIN_2){
          //get the switch state
          byte value = rcvmsg[2];
          //set output pin to according state
          if(value == VALUE_ON) {
            digitalWrite(PIN, HIGH);
          }
          else if(value == VALUE_OFF) {
            digitalWrite(PIN, LOW);
          }
        }
      }
    }
  }
}
```

Exemple : piloter une LED à partir d'Android



ADK

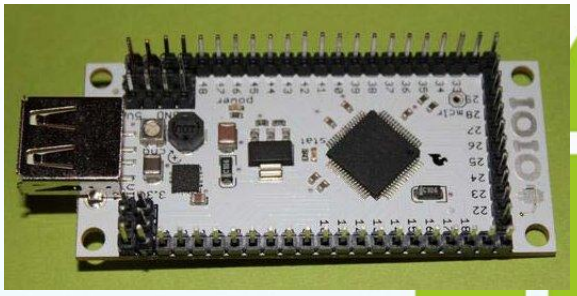
TD Arduino ADK

- mini Projet 1 : Piloter la carte depuis Android

- mini Projet2 : Piloter Android depuis une carte

IOIO

Sparkfun IOIO board / Microcontroleur PIC24



- PIC24
- 48 pins d’entrées/sorties
- 16 entrées analogiques (10-bits)
- 9 sorties PWM
- jusqu’à 4 liaisons séries UART
- jusqu’à 3 liaisons SPI
- jusqu’à 3 liaisons TWI (I2C-compatible)
- V3.04

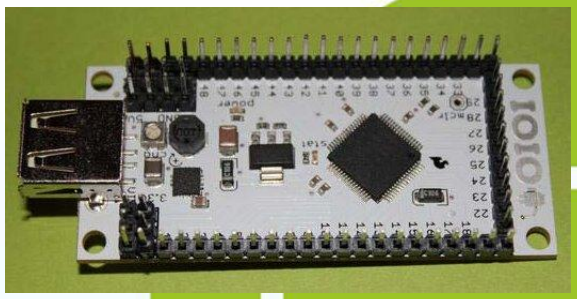
La platine IOIO-OTG (prononcé "yo-yo-O-T-G") est un module d'interface spécialement conçu pour être piloté via un dispositif Android (avec version d'OS 1.5 et sup.) par le biais de son port USB. Ce pilotage s'effectue via des API JAVA sans avoir recours à une programmation embarquée, ni au moindre programmateurs externe. Le processeur embarqué du module IOIO interpréte ainsi les commandes issues de l'application Android.

La carte IOIO agit comme un hôte USB et se connecte à la plupart des appareils Android USB esclave.

https://www.sparkfun.com/tutorials/280

IOIO

Principe de base



```
code Java
{
 xxxxxxxxxxxx
 xxxxxxxxxxxx
 xxxxxxxxxxxx
}
```

Interprétation du code Java par le microcontrôleur.
Commande par USB ou par librairie bluetooth.

IOIO

Principe de base

```
ioio.waitForConnect();
AnalogInput input = ioio.openAnalogInput(40);
PwmOutput pwmOutput = ioio.openPwmOutput(12, 100);  // 100Hz

while (true)
{
  float reading = input.read();
  pwmOutput.setPulseWidth(1000 + Math.round(1000 * reading));
  sleep(10);
}
```

Environnement de développement

ADK

TD IOIO

- mini Projet 1 : Actionner une led de la carte

- mini Projet2 : Relever de température à distance avec une communication bluetooth