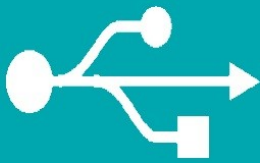


APP INVENTOR 2



USB
OTG

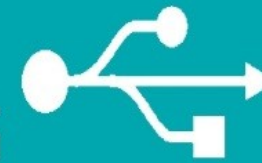


UART
Rx-TX

APP INVENTOR 2



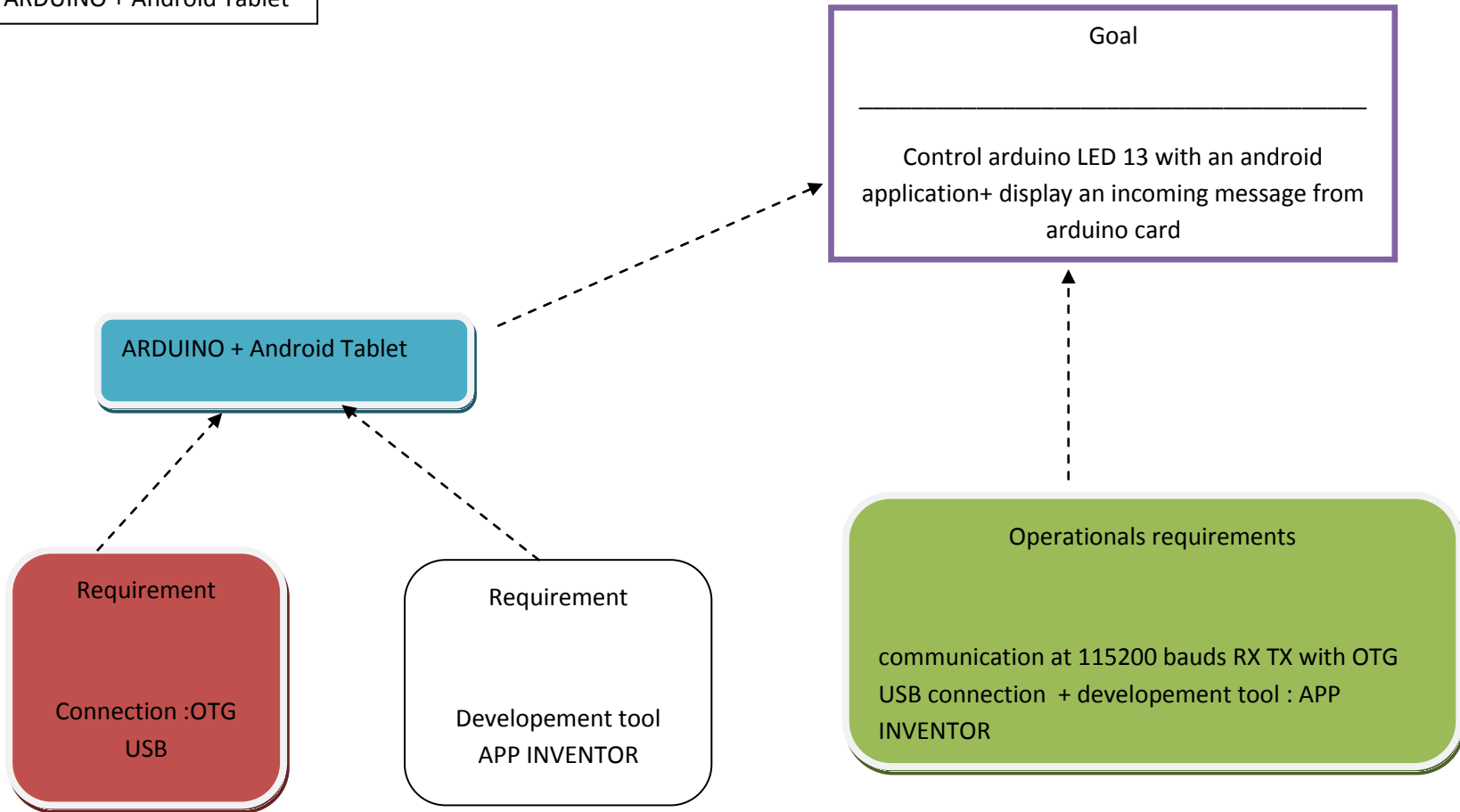
USB
OTG



UART
Rx-TX

[COMMUNICATION : APP INVENTOR <-USB->UART]

Goal "ARDUINO + Android Tablet"



Bdd : context system

Useful to develop in a short time an Android application (app inventor 2) with communication between an arduino card and OTG USB connection

Download app on Play store : USB Bridge for App Inventor 2

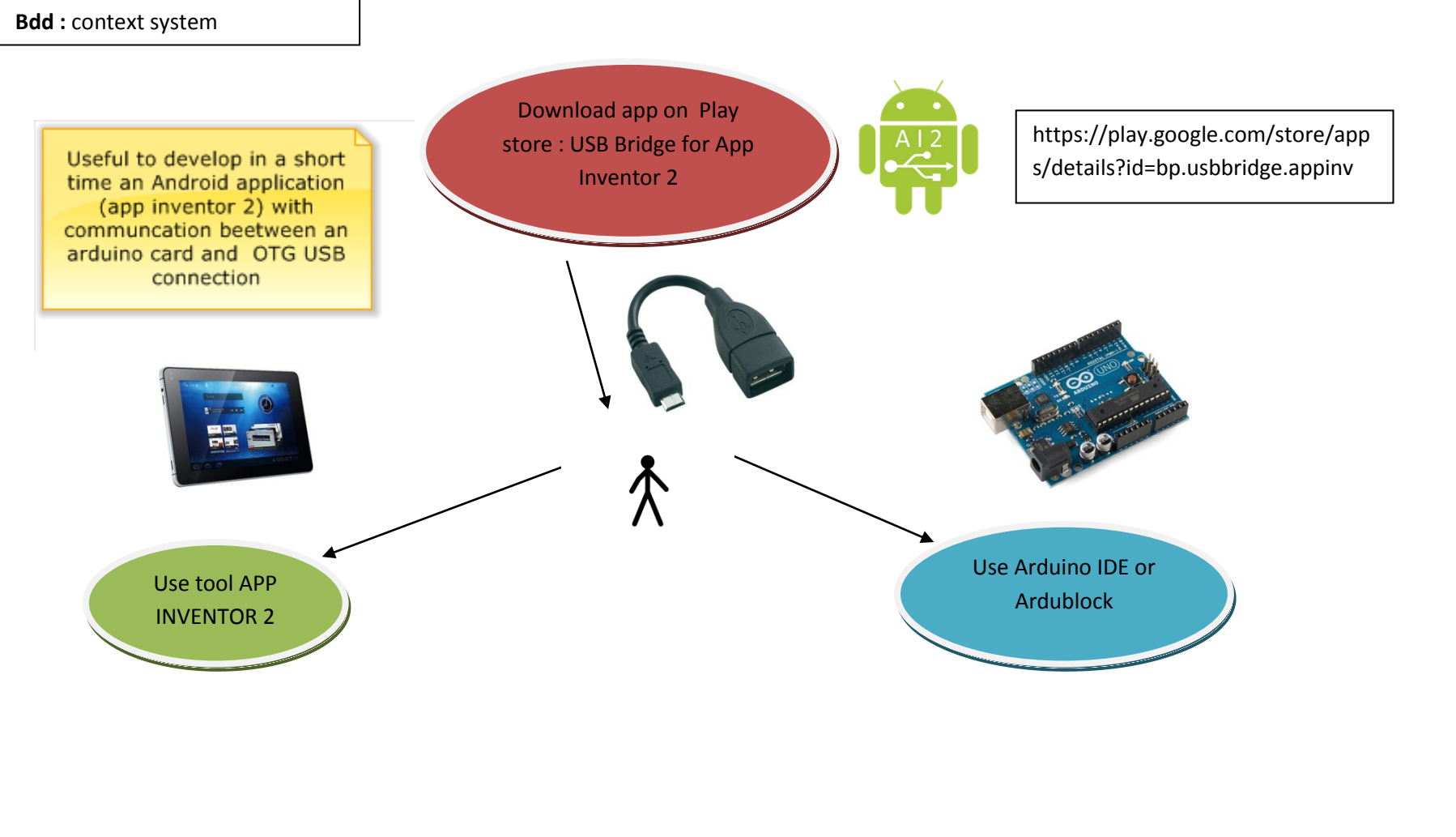


<https://play.google.com/store/apps/details?id=bp.usbridge.appinv>

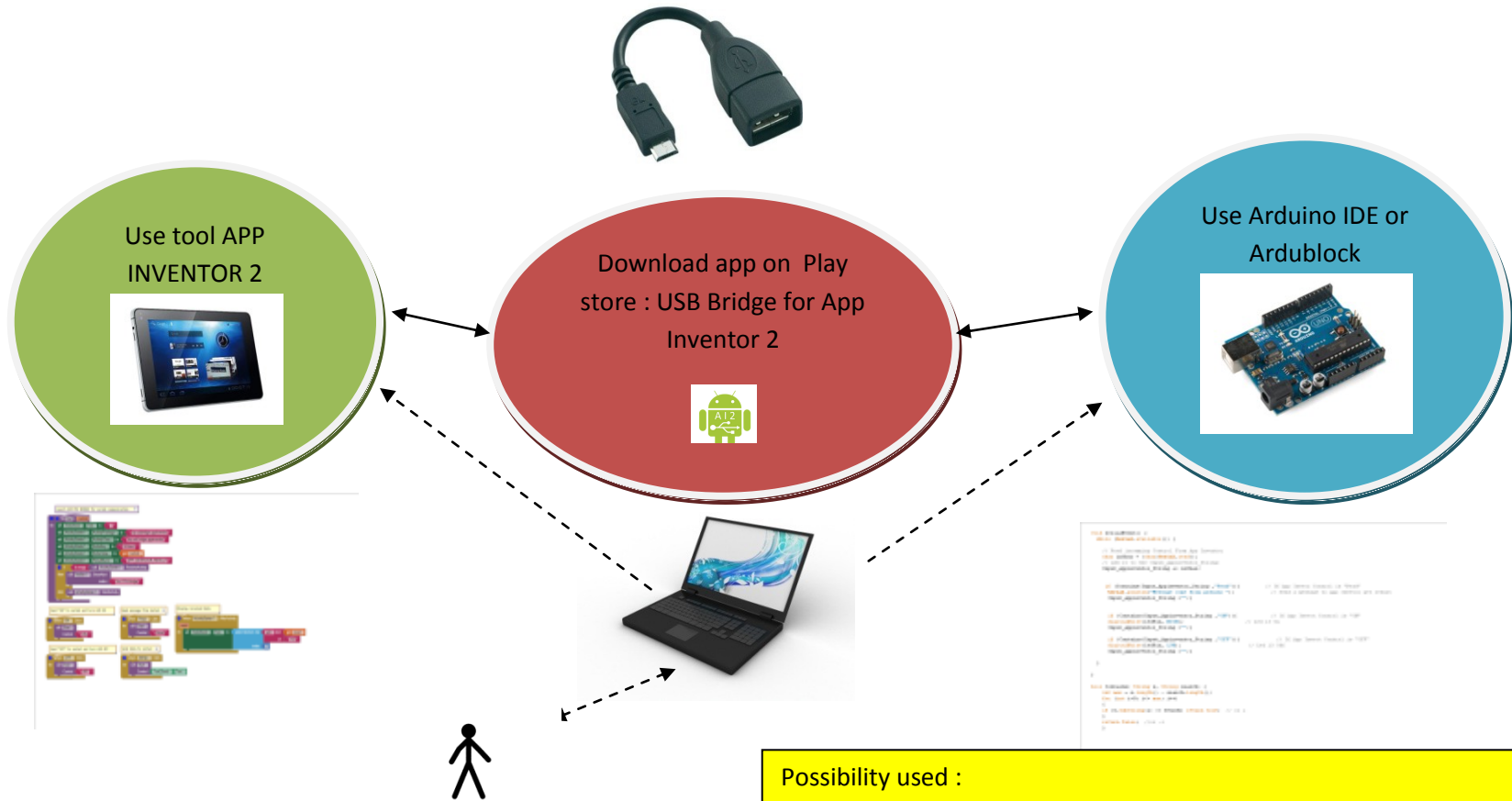


Use tool APP INVENTOR 2

Use Arduino IDE or Ardublock

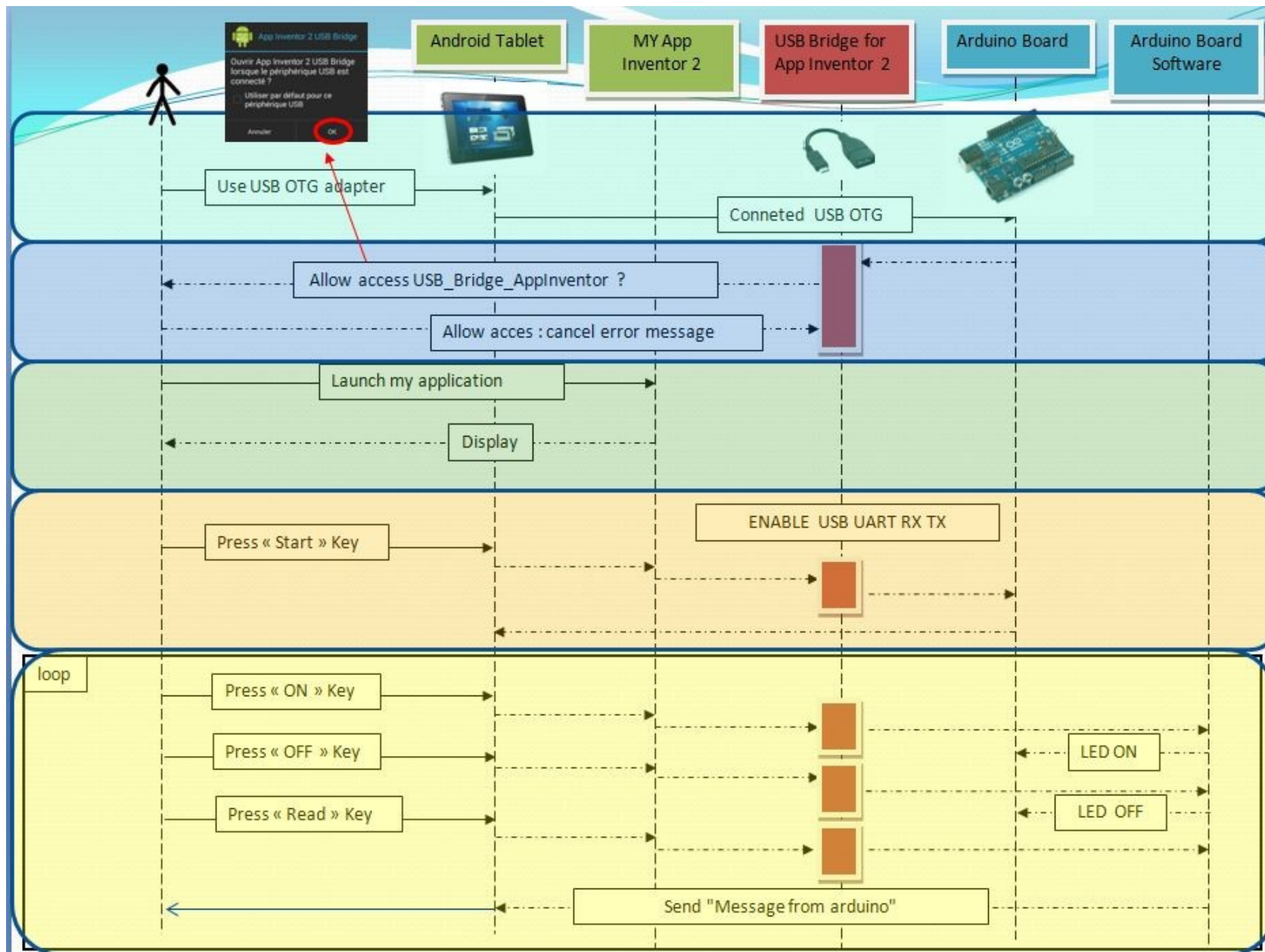


Use Case Diagram: Software

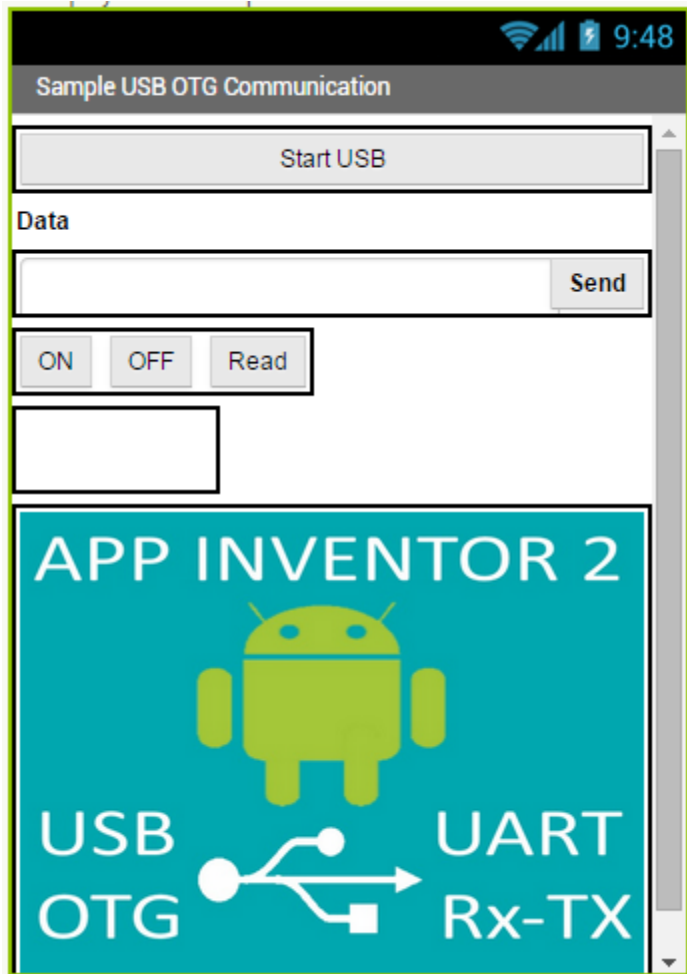


Possibility used :

- 1: Develop an Android application with App Inventor 2
- 2: Develop an Arduino application (115200 Bauds)
- 3: Download and Use USB_Bridge_AppInventor to allow communication



Android Tablet APP INVENTOR 2



1) Initialisation

when Screen1.Initialize

```
do
  set ON.Enabled to false
  set OFF.Enabled to false
  set Read.Enabled to false
  set SEND.Enabled to false
```

Launch USB OTG BRIDGE for serial communication

? when Start_USB.Click

```
do
  set ActivityStarter1.ActivityPackage to "bp.usbbridge.appinv"
  set ActivityStarter1.ActivityClass to "bp.usb.bridge.appinventor"
  set SEND.Enabled to true
  set ON.Enabled to true
  set OFF.Enabled to true
  set Start_USB.Enabled to false
  set Read.Enabled to true
```

RX / TX serial communication

2) Communication

```
do
  set Received . Text to ""
  set ActivityStarter1 . ExtraKey to "OnSend"
  set ActivityStarter1 . ExtraValue to get Control
  set ActivityStarter1 . ResultName to "APP_INVENTOR_RESULT"
  if is empty call ActivityStarter1 . ResolveActivity
  then
  else call ActivityStarter1 . StartActivity
```

3) Write Command

Send "ON" to serial and turn LED ON

```
when ON . Click
do
  call Start
  Control "ON"
```

Send "OFF" to serial and turn LED OFF

```
when OFF . Click
do
  call Start
  Control join "OFF"
```

Send data to serial

```
when SEND . Click
do
  call Start
  Control Received . Text
```

4) Read Command

Read message from Serial

```
when Read . Click
do
  call Start
  Control "Read"
```

Display received data

```
when ActivityStarter1 . AfterActivity
result
do
  set Received . Text to get result
```

ARDUINO Card

```
// Sample Arduino sketch for use with usb-serial-for-android OTG and APP INVENTOR 2
```

```
// 09/02/2015
```

```
// LED ON (13) if received "ON"
```

```
// LED OFF (13) If received "OFF"
```

```
// Send a message if received "Read"
```

```
String Input_Appinventor_String = "";           // String to hold incoming App inventor Data
```

```
const int ledPin = 13;
```

```
int Valeur = 0 ;
```

```
void setup() {
```

```
  Serial.begin(115200);
```

```
  // Pin 13 has an LED connected on most Arduino boards:
```

```
  pinMode(ledPin, OUTPUT);
```

```
  Input_Appinventor_String.reserve(200);
```

```
  delay(2000);
```

```
}
```

```
void loop() {
```

```
  delay(10);
```

```
}
```



```

void serialEvent() {
  while (Serial.available()) {

    // Read incoming Control From App Inventor
    char inChar = (char)Serial.read();
    // add it to the Input_Appinventor_String:
    Input_Appinventor_String += inChar;

    if (Contains(Input_Appinventor_String , "Read")){           // If App Inveor Control is "Read"
      Serial.println("Message come from Arduino ");           // Send a message to App Inveor get result
      Input_Appinventor_String = "";
    }

    if (Contains(Input_Appinventor_String , "ON")){           // If App Inveot Control is "ON"
      digitalWrite(ledPin, HIGH);                             // Led 13 On
      Input_Appinventor_String = "";
    }

    if (Contains(Input_Appinventor_String , "OFF")){          // If App Inveot Control is "OFF"
      digitalWrite(ledPin, LOW);                               // Led 13 Off
      Input_Appinventor_String = "";
    }

  }
}

bool Contains( String s, String search) {
  int max = s.length() - search.length();
  for (int i=0; i<= max; i++)
  {
    if (s.substring(i) == search) return true; // or i
  }
  return false; //or -1
}

```

Program Arduino with ARDUBLOCK : ardublock-beta-20140828.jar

<http://sourceforge.net/projects/ardublock/files/>

The screenshot displays the Ardublock software interface with two main code blocks: 'programme' and 'serialEvent'. The 'programme' block contains a 'Programme d'installation (setup)' section with a 'set String variable' block for 'InputAppinventor String' and a 'Serial.begin(115200);' block. It also includes a 'Tête du programme (head)' section with a 'bool Contains(String s, String search) {' function definition. The 'serialEvent' block contains a 'Boucle' section with a 'while (Serial.available()) {' loop, a 'char inChar = (char)Serial.read();' block, and a 'Boucle' section with an 'ABVAR 1 InputAppinventor String += inChar;' block. The 'serialEvent' block also includes 'Read', 'ON', and 'OFF' blocks. The 'Read' block contains a 'Boucle' section with a 'while' loop and a 'Serial.println("Message from Arduino);' block. The 'ON' block contains a 'Boucle' section with a 'while' loop and a 'Fixe la sortie numerique au niveau # 13 HIGH' block. The 'OFF' block contains a 'Boucle' section with a 'while' loop and a 'Fixe la sortie numerique au niveau # 13 BAS' block. The interface also features several 'Teste' blocks and 'Boucle' blocks for conditional execution. A 'Main' block is visible in the bottom right corner. A yellow callout box at the top right says 'Don't change this part, but you can add own setup informations'. A yellow callout box at the bottom left says 'When a serial event appear, this function read incomming message and can compare it. You can just change parts Read, ON, OFF or add new parts as you want'. A yellow callout box at the bottom center says 'Copy this function and Add new Commands to control servo...'. The interface is in French.

```
programme
Programme d'installation (setup)
  set String variable InputAppinventor String
  Serial.begin(115200);
Tête du programme (head)
  bool Contains( String s, String search) {
  int max = s.length() - search.length();
  for (int i=0; i<= max; i++){
  if (s.substring(i) == search) return true; } return false; }
  définir une variable numerique Test
  Valeur FAUX
  delay MILLIS Millisecondes 200
  Boucle
  delay MILLIS Millisecondes 10

serialEvent
Boucle
  while (Serial.available()) {
  Boucle
  char inChar = (char)Serial.read();
  Boucle
  ABVAR 1 InputAppinventor String += inChar;
  Read
  ON
  OFF
  Boucle
  }
  set String variable InputAppinventor String
  Valeur ""

Read
Boucle
  while (Serial.available()) {
  Boucle
  char inChar = (char)Serial.read();
  Boucle
  ABVAR 1 InputAppinventor String += inChar;
  Read
  ON
  OFF
  Boucle
  }
  set String variable InputAppinventor String
  Valeur ""

ON
Boucle
  while (Serial.available()) {
  Boucle
  char inChar = (char)Serial.read();
  Boucle
  ABVAR 1 InputAppinventor String += inChar;
  Read
  ON
  OFF
  Boucle
  }
  set String variable InputAppinventor String
  Valeur ""

OFF
Boucle
  while (Serial.available()) {
  Boucle
  char inChar = (char)Serial.read();
  Boucle
  ABVAR 1 InputAppinventor String += inChar;
  Read
  ON
  OFF
  Boucle
  }
  set String variable InputAppinventor String
  Valeur ""
```