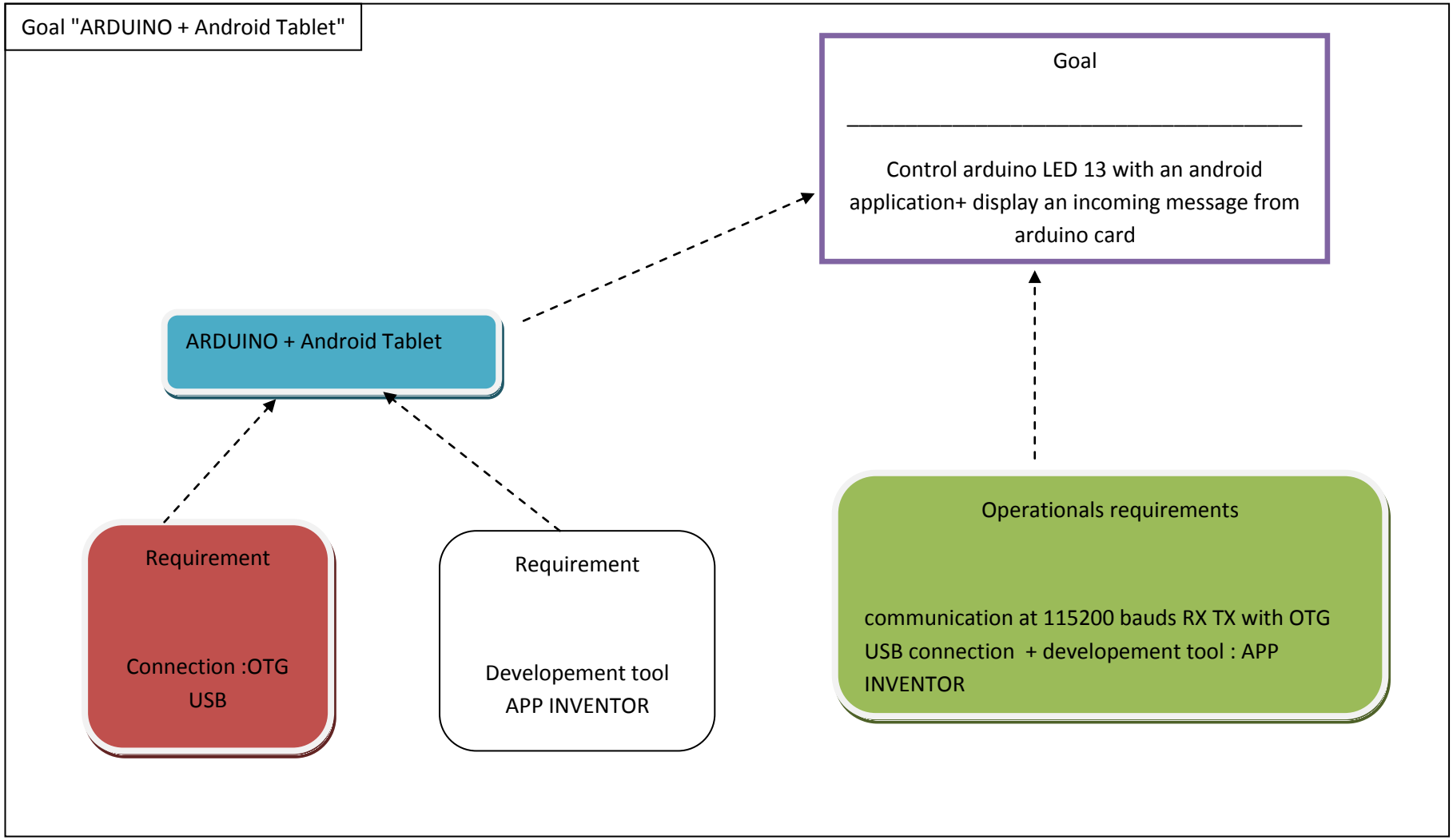
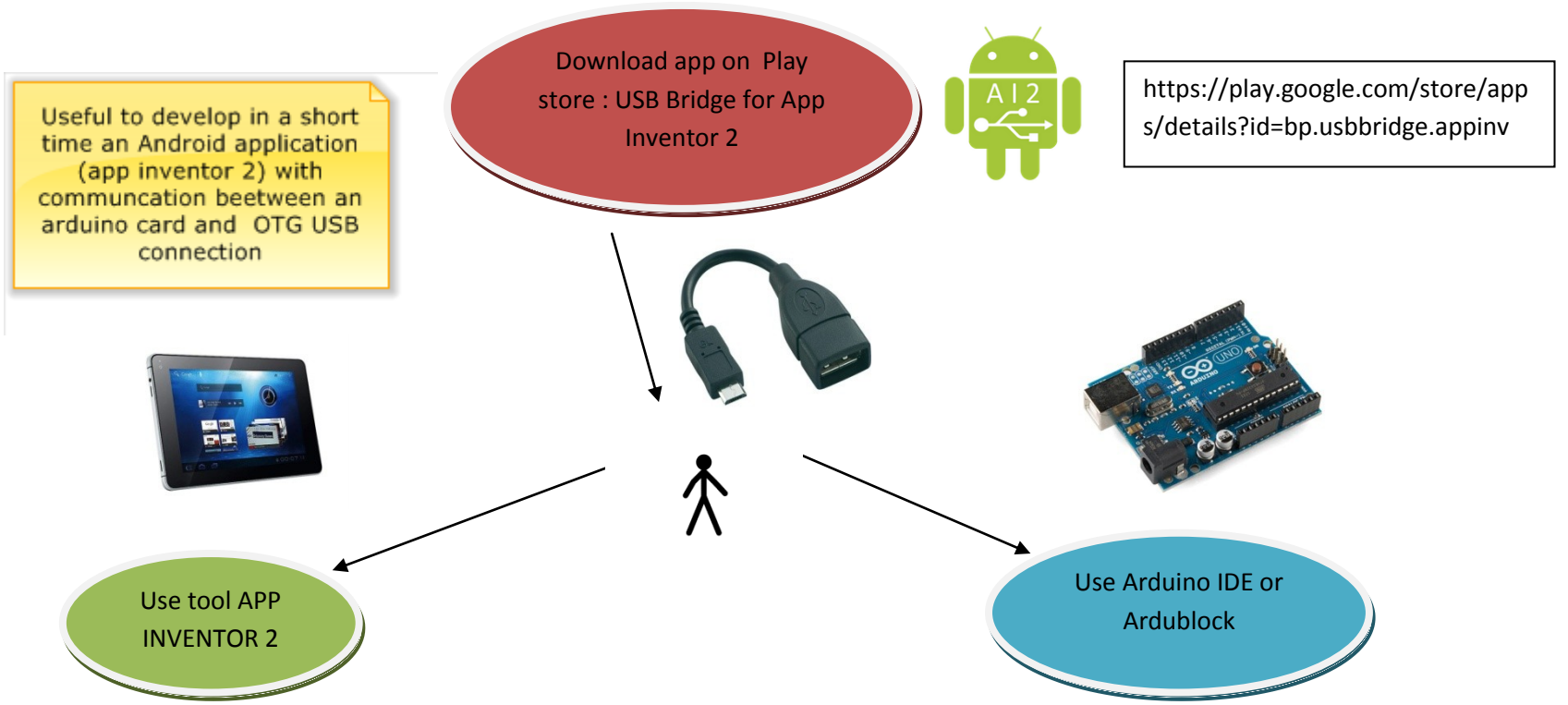
[COMMUNICATION : APP INVENTOR <-USB->UART]


Goal "ARDUINO + Android Tablet"
Goal
Control arduino LED 13 with an android application+ display an incoming message from arduino card
ARDUINO + Android Tablet
Requirement
Connection :OTG USB
Requirement
Developement tool APP INVENTOR
Operationals requirements
communication at 115200 bauds RX TX with OTG USB connection + developement tool : APP INVENTOR

Bdd : context system

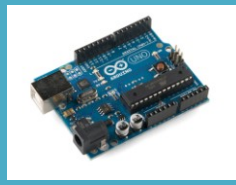


Use Case Diagram: Software

Use tool APP INVENTOR 2

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

Use Arduino IDE or Ardublock

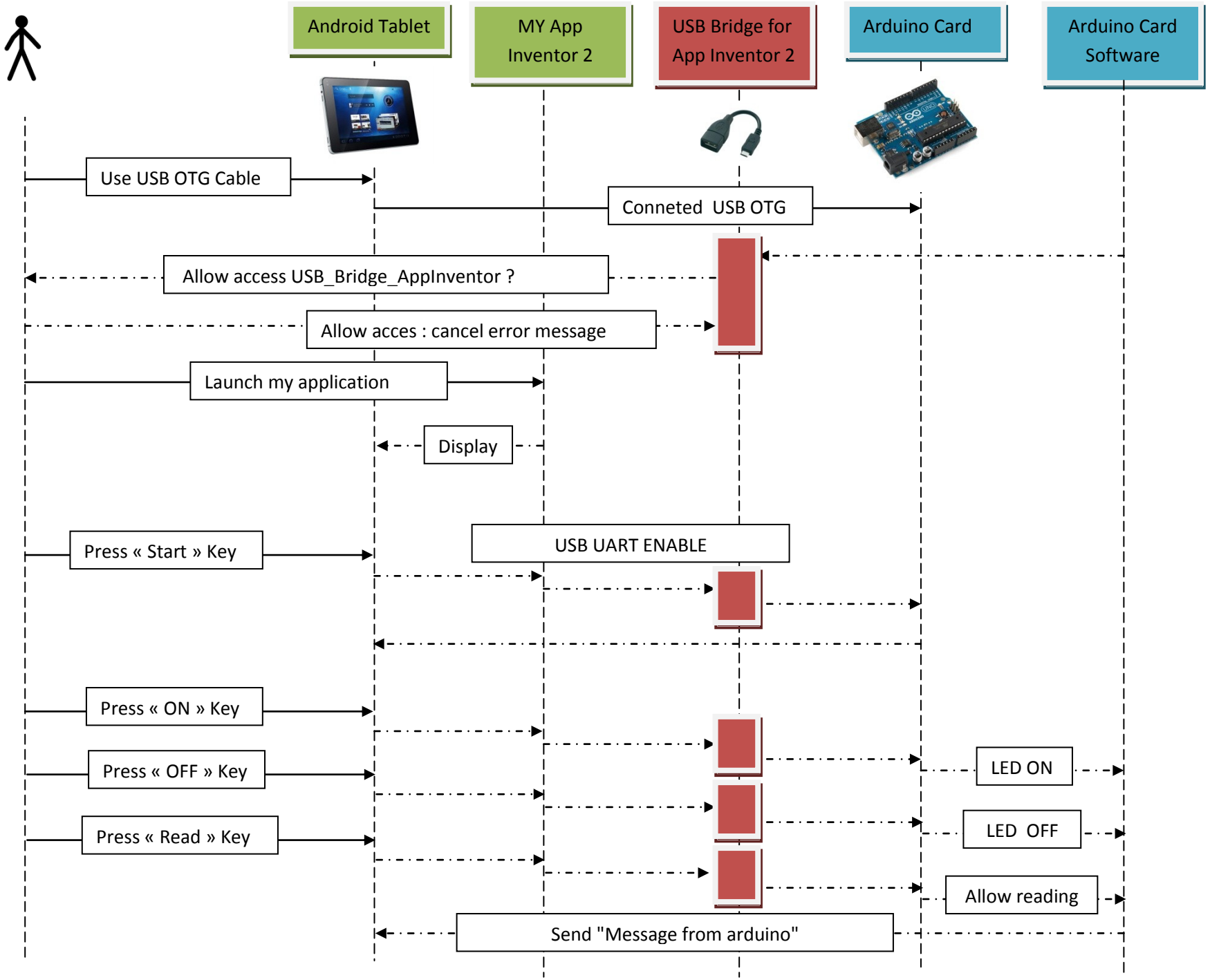


Possibility used :

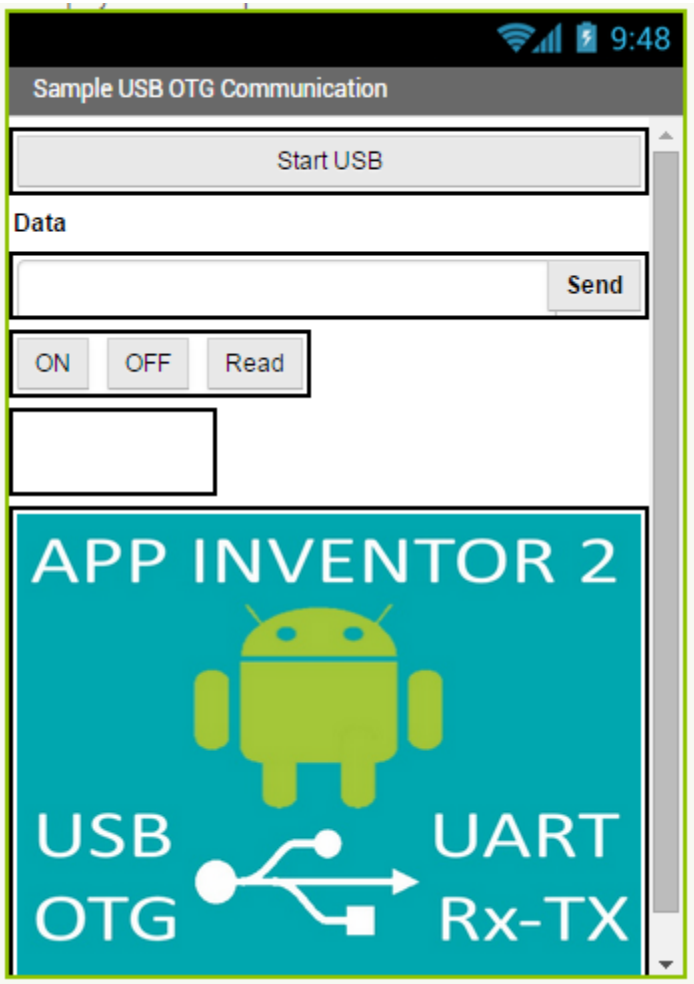
1: Develop an Android application with App Inventor 2

2: Develop an Arduino application (115200 Bauds)

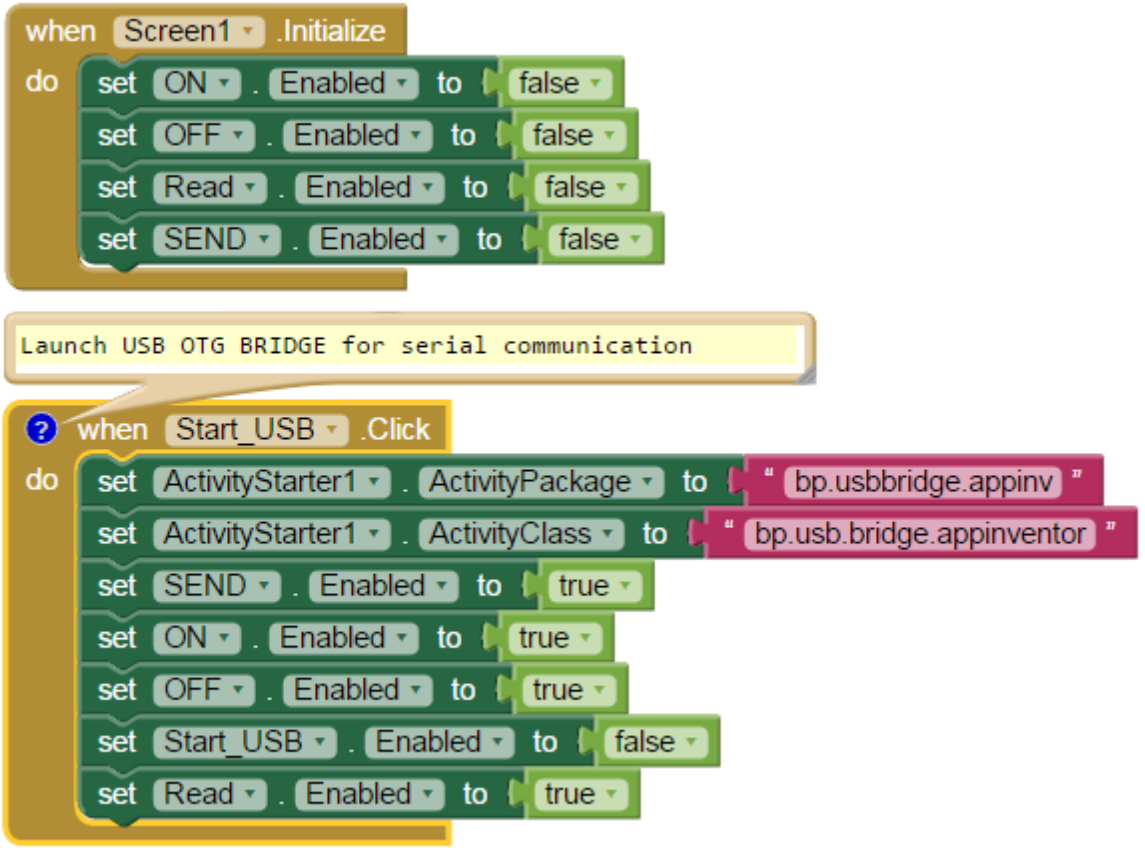
3: Download and Use USb_Bridge_AppInventor to allow communication

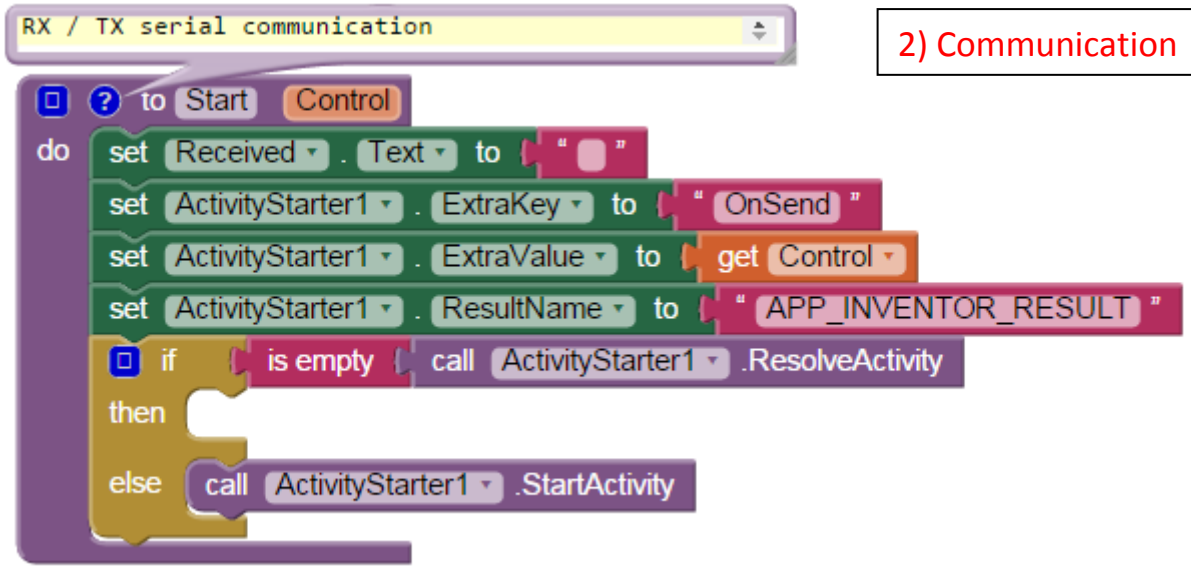

Android Tablet
MY App Inventor 2
USB Bridge for App Inventor 2
Arduino Card
Arduino Card Software
Use USB OTG Cable
Conneted USB OTG
Allow access USB_Bridge_AppInventor ?
Allow acces : cancel error message
Launch my application
Display
Press « Start » Key
USB UART ENABLE
Press « ON » Key
Press « OFF » Key
Press « Read » Key
LED ON
LED OFF
Allow reading
Send "Message from arduino"

Android Tablet APP INVENTOR 2



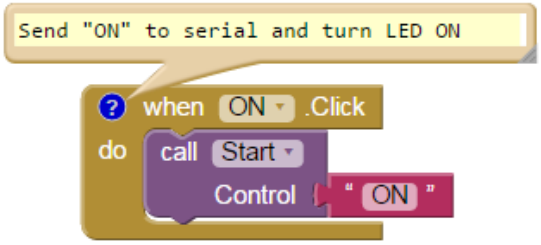
1) Initialisation

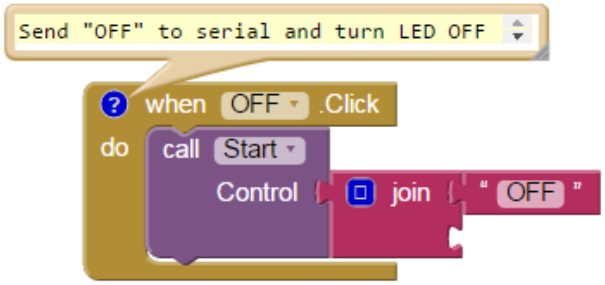


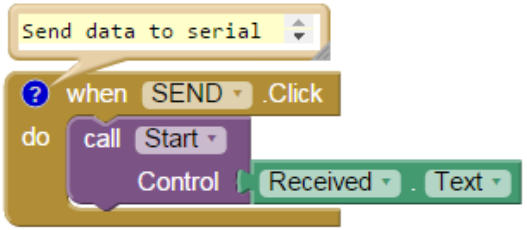


2) Communication

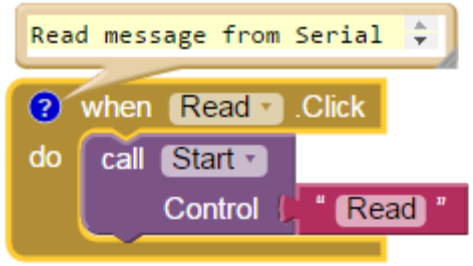
3) Write Command

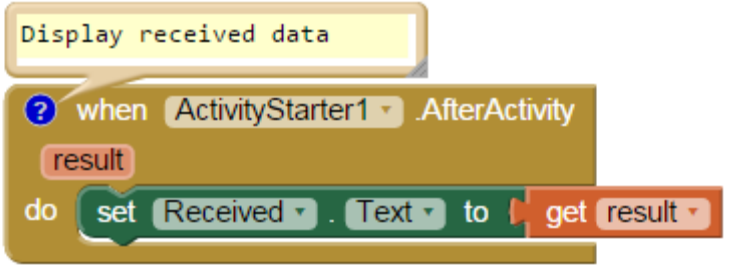






4) Read Command





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 incomming 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 Invetor 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/