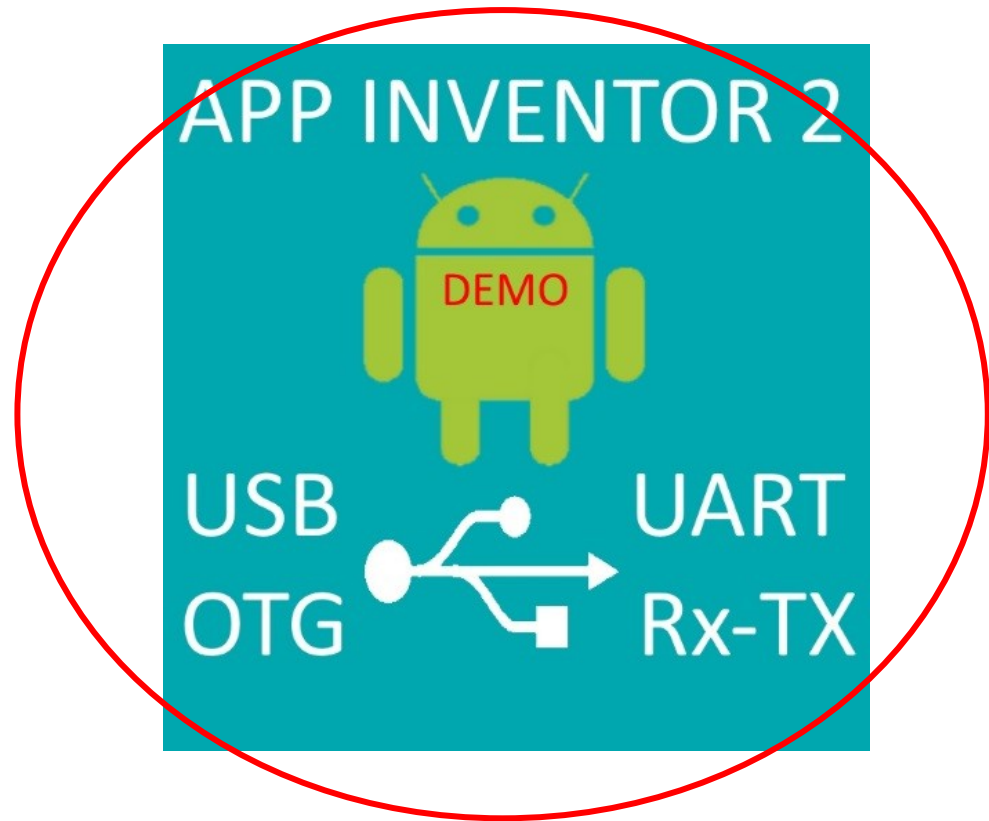
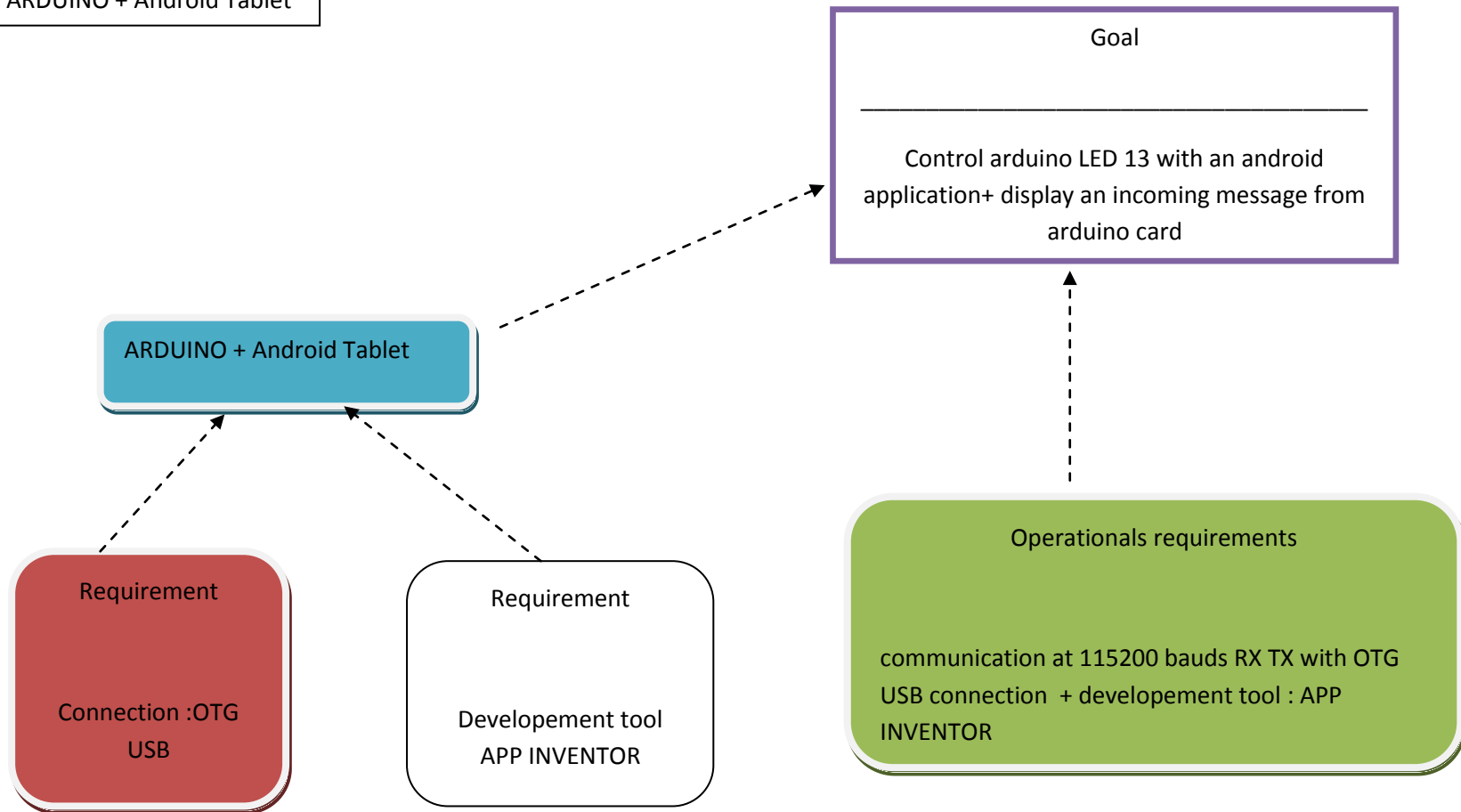


HELP FOR DEMO



[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.usbbridge.appinv>

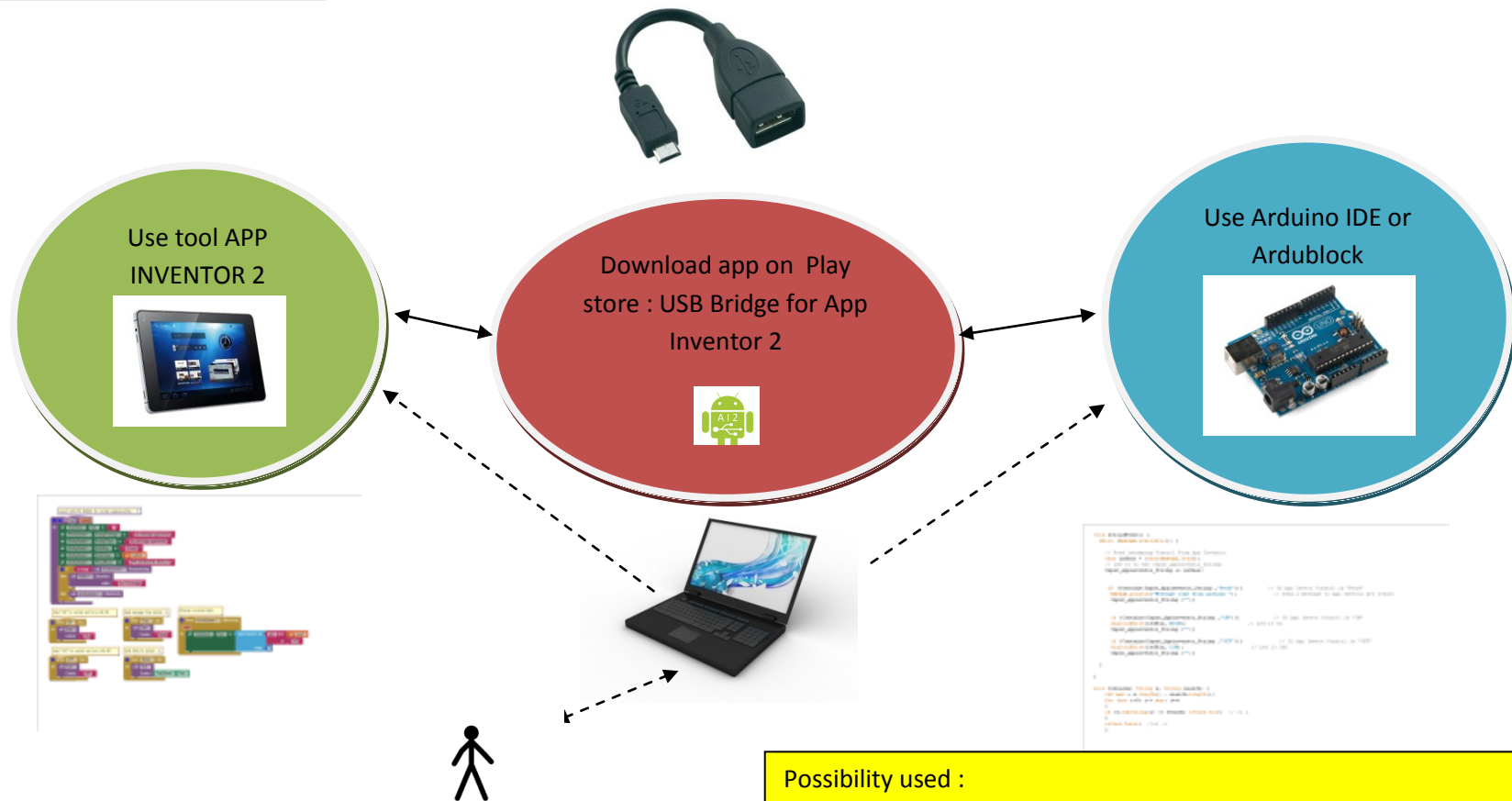


Use tool APP INVENTOR 2



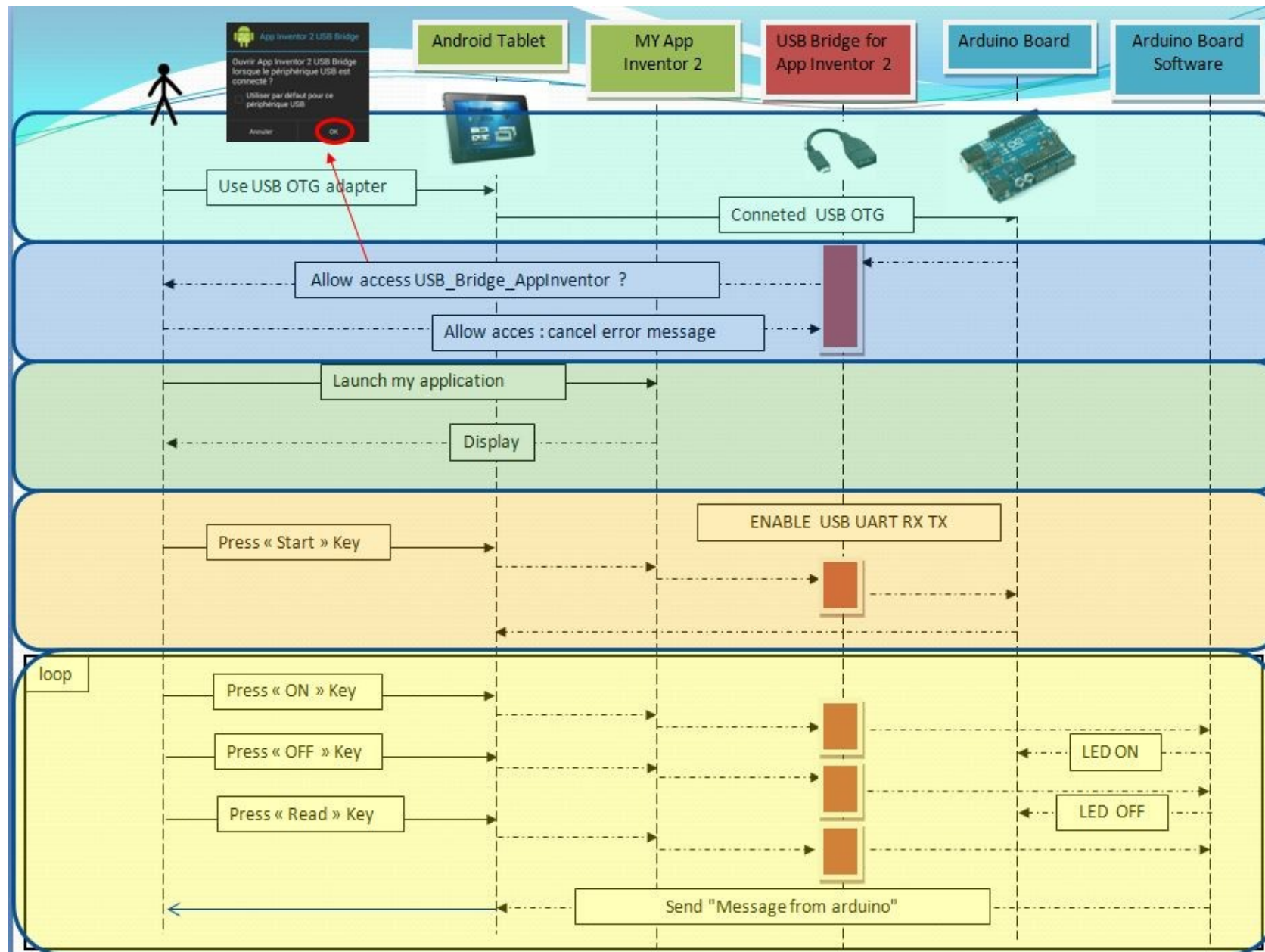
Use Arduino IDE or Ardublock

Use Case Diagram: Software

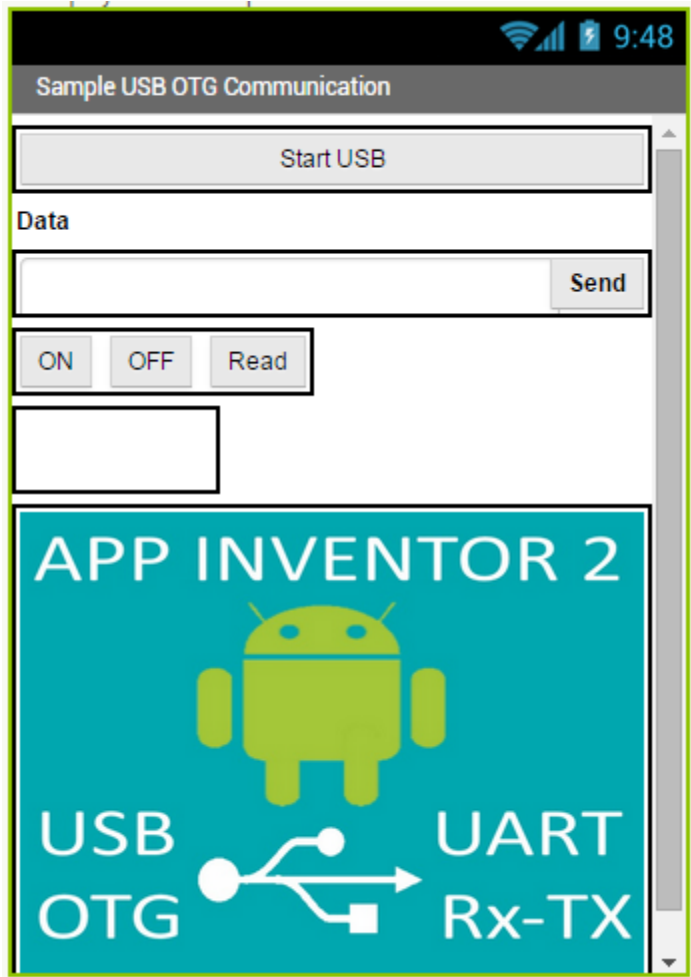


Possibility used :

- 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

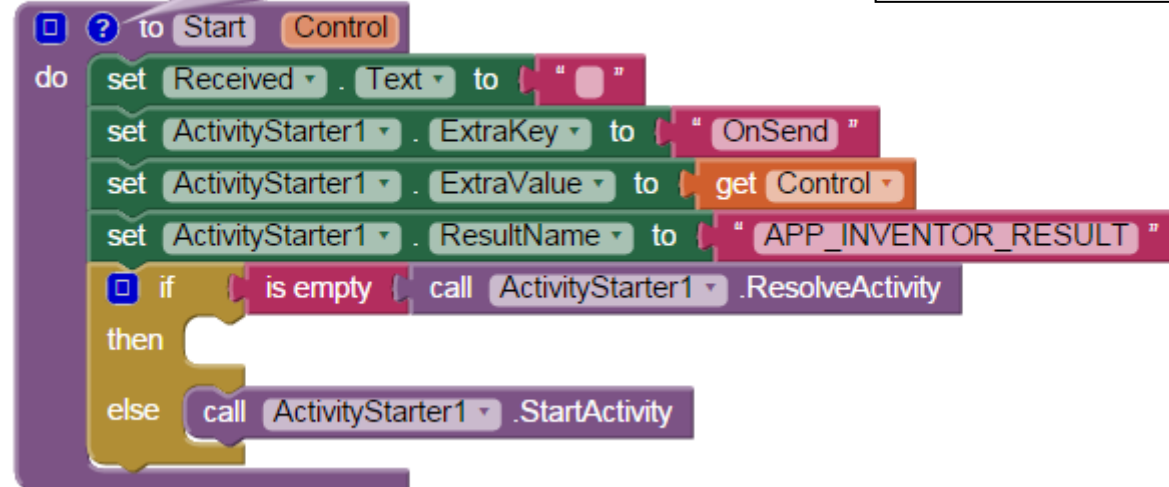
? when Start_USB.Click

do

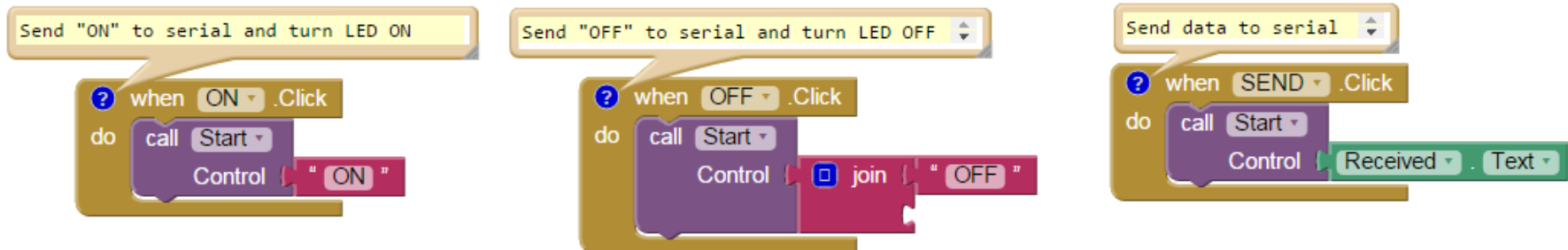
- set ActivityStarter1.ActivityPackage to "bp.usbbridge.appinv"
- set ActivityStarter1.ActivityClass to "bp.usb.bridge.appinventor"
- set SEND.Enabled to true
- set ON.Enabled to true
- set OFF.Enabled to true
- set Start_USB.Enabled to false
- set Read.Enabled to true

RX / TX serial communication

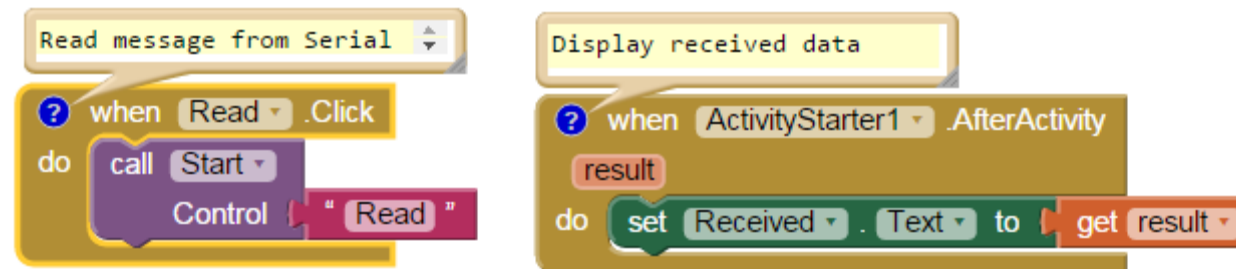
2) Communication



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 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 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/>

