

Application Vidéo Client Serveur Bluetooth

Sommaire

1	Test Application Vidéo	2
2	Pré-requis.....	2
3	Déclaration du projet CameraFrame sous Eclipse Luna	2
3.1	Création du projet.....	2
3.2	Création Class Stream	3
3.3	Création Class CameraFrame	3
4	Modification du code	4
4.1	Modification du code de CameraFrame	4
4.2	Modification du code de Stream	4
5	Exécution.....	5
5.1	Connexion en bluetooth de la brick EV3 et du PC	5
5.1.1	Afficher les périphériques bluetooth	5
5.1.2	Ajouter un périphérique	5
5.2	Configuration CameraFrame.....	7
5.3	Exécution du programme Stream coté brick (Serveur)	7
5.4	Exécution du programme CameraFrame coté PC (Client)	8

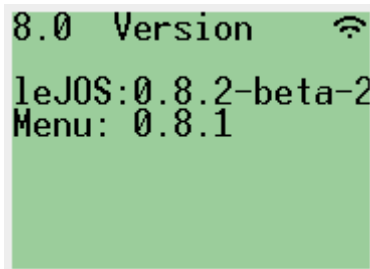
1 Test Application Vidéo

Le code utilisé se trouve sur ce lien :

➤ <http://lejosnews.wordpress.com/2014/09/04/webcam-streaming/>

2 Pré-requis

Brick en Lejos 0.8.2

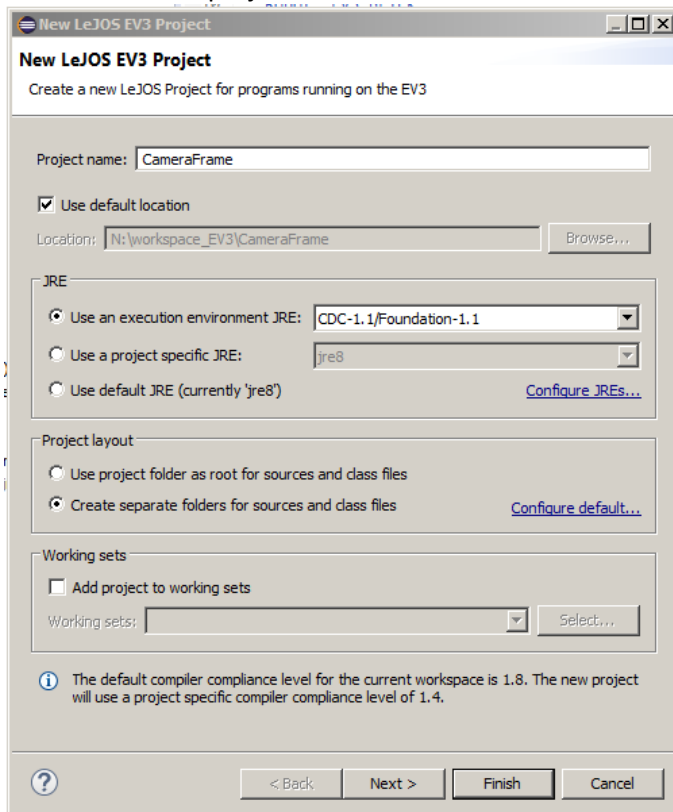


3 Déclaration du projet CameraFrame sous Eclipse Luna

3.1 Création du projet

File → New → Lejos EV3 Project

Entrer le nom du projet : CameraFrame

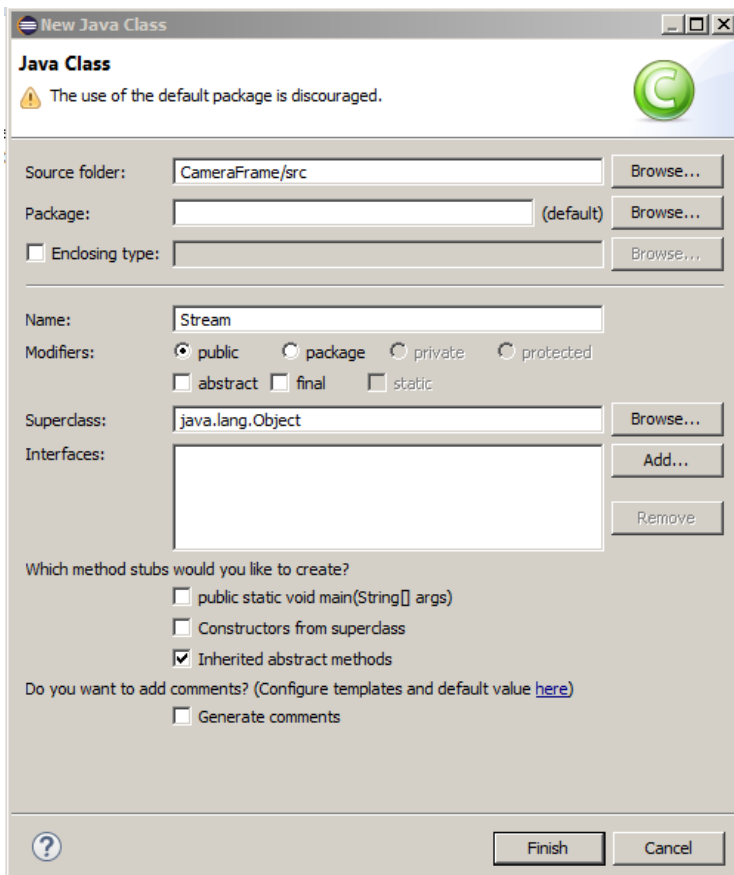


Finish

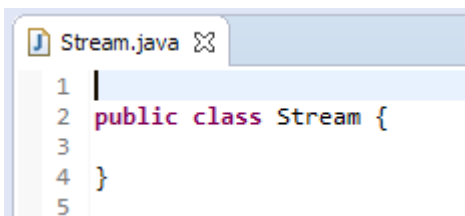
3.2 Création Class Stream

File → New → Class

Entrer le nom de la Class : Stream



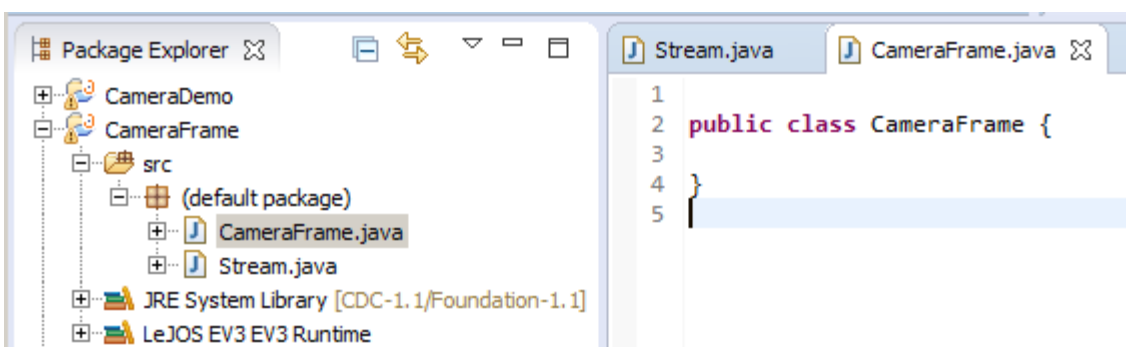
Finish



3.3 Création Class CameraFrame

File → New → Class

Entrer le nom de la Class :CameraFrame



Copier les codes correspondants du lien ci-dessus dans chaque classes.

4 Modification du code

4.1 Modification du code de CameraFrame

```
public class CameraFrame {
    private static final int WIDTH = 160;
    private static final int HEIGHT = 120;
    private static final int NUM_PIXELS = WIDTH * HEIGHT;
    private static final int BUFFER_SIZE = NUM_PIXELS * 2;
    private static final int PORT = 1234;

    private ServerSocket ss;
    private Socket sock;
    private byte[] buffer = new byte[BUFFER_SIZE];
    private BufferedInputStream bis;
    private BufferedImage image;
    private CameraPanel panel = new CameraPanel();
    private JFrame frame;

    public CameraFrame() {
        try {
            // ss = new ServerSocket(PORT);
            // sock = ss.accept();
            Socket sock = new Socket("10.0.1.1", 1234);
            System.out.println("Connected");
            bis = new BufferedInputStream(sock.getInputStream());
        }
    }
}
```

Enlever la ligne d'entête définissant le package

Commenter tous les @Override

```
frame.addWindowListener(new WindowAdapter() {
    // @Override
    public void windowClosed() {
    }
});
```

4.2 Modification du code de Stream

```
private static final String HOST = "10.0.1.1";
private static final int PORT = 1234;

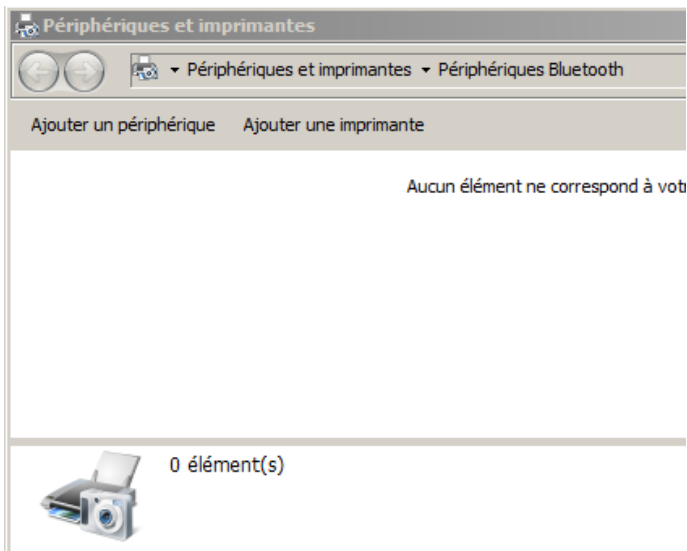
public static void main(String[] args) throws IOException {
    EV3 ev3 = (EV3) BrickFinder.getLocal();
    Video video = ev3.getVideo();
    video.open(WIDTH, HEIGHT);
    byte[] frame = video.createFrame();
    ServerSocket server = new ServerSocket(PORT);
    Socket sock = server.accept();
    // Socket sock = new Socket(HOST, PORT);
    System.out.println("CONNECTED");
}
```

Enlever la ligne d'entête définissant le package

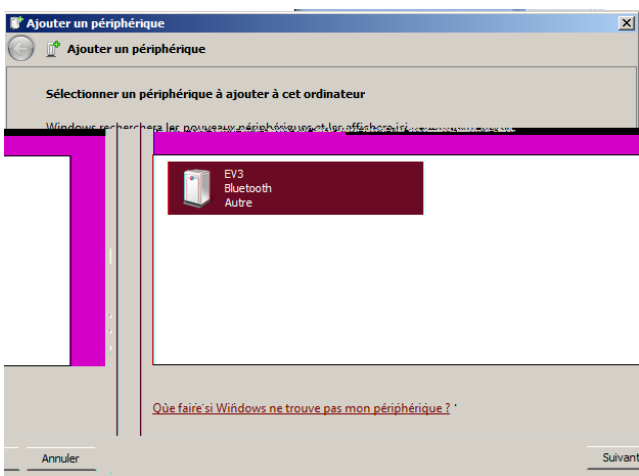
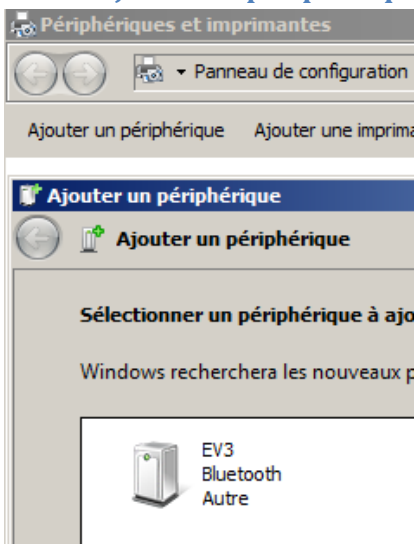
5 Exécution

5.1 Connexion en bluetooth de la brick EV3 et du PC

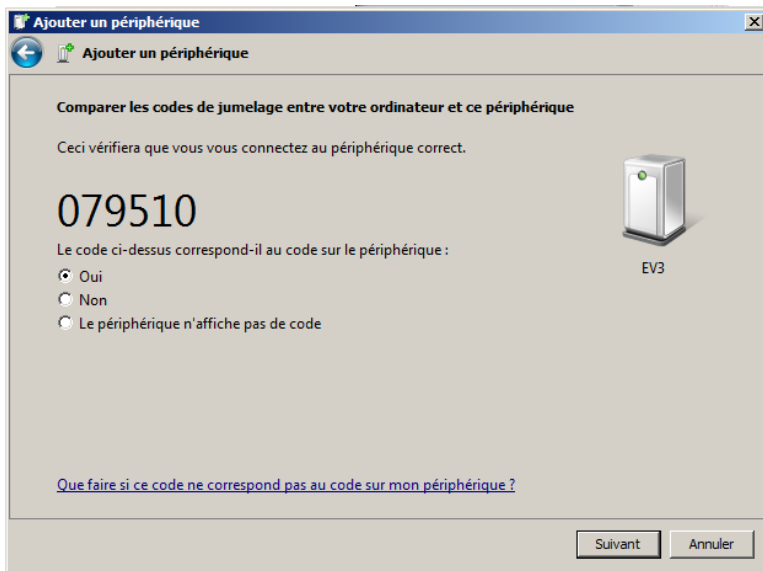
5.1.1 Afficher les périphériques bluetooth



5.1.2 Ajouter un périphérique



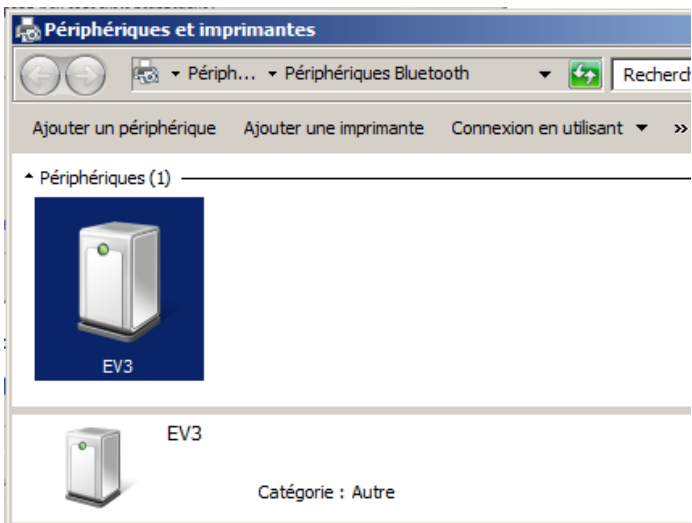
Suivant



Laisser coché **Oui**, **Suivant** (ne pas s'occuper du code)



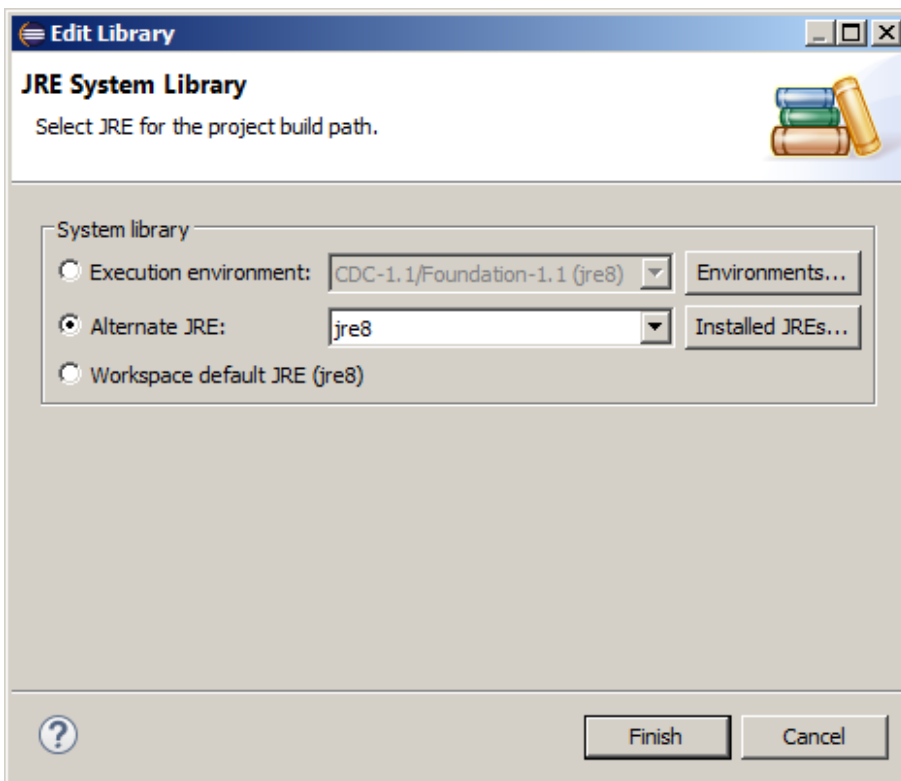
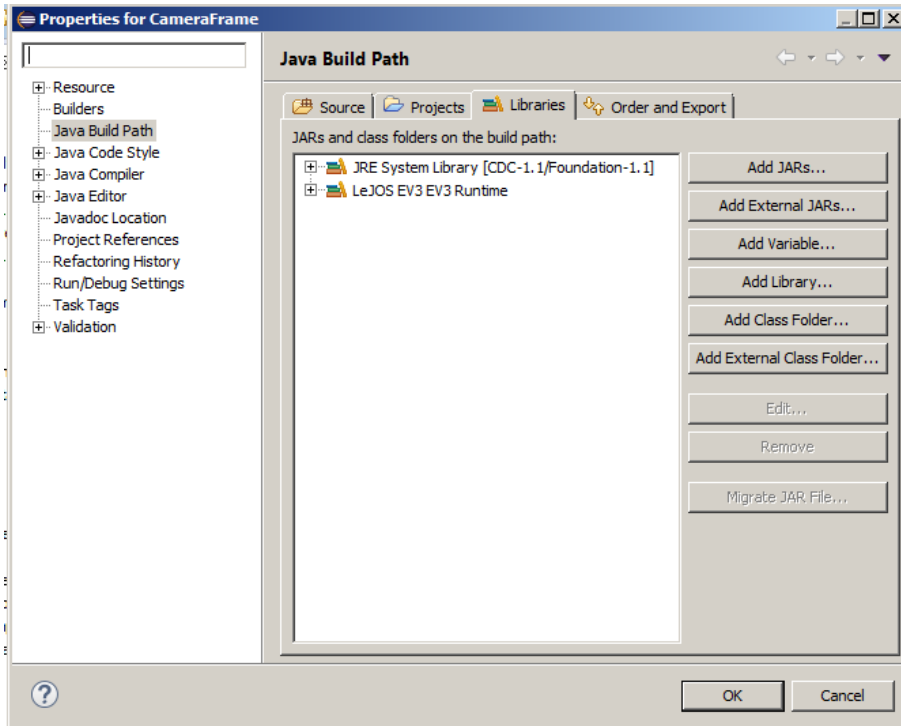
Fermer



Cliquer sur **Connexion utilisant**, choisir **Point d'accès**

5.2 Configuration CameraFrame

Cliquer sur Projet CameraFrame, Click droit **Properties**



Finish

5.3 Exécution du programme Stream coté brick (Serveur)

Click droit sur Stream → Run as → Lejos EV3 program

5.4 Exécution du programme CameraFrame coté PC (Client)

Click droit sur CameraFrame → Run as → Java Application

