Ardulink is a Java Open Source project. Please visit www.ardulink.org for more information.

Ardulink depends on some Java libraries and native code.

This is a detailed list:

Serial Connection

Serial Connection is the main connection method to Arduino boards. Ardulink SerialConnection uses serial RXTX library.

RXTX library site is http://rxtx.qbang.org
RXTX native code for windows systems (rxtxSerial.dll) is already in Ardulink distribution package. Run set32bitWindowsRXTX.bat or set64bitWindowsRXTX.bat regarding your windows system.
You can install RXTX for other systems reading instructions on RXTX site.
For linux systems you could use this command:

```
sudo apt-get install librxtx-java
```

For more info about linux systems read this article:
http://www.ardulink.org/how-to-install-arduino-pc-link-on-raspberry-pi/

Bluetooth Connection

Bluetooth Connection is intended for Arduino boards equipped with a bluetooth extension. This connection is developped and tested with an HC-06 board attached to an Arduino UNO. It uses bluecove 2.1.0 java library. In the Ardulink distribution you will find the bluecove jar but not the bluecove GPL jar intended for linux systems. If you need for it, you can download it from bluecove site.
Bluecove site is http://bluecove.org

USB Connection

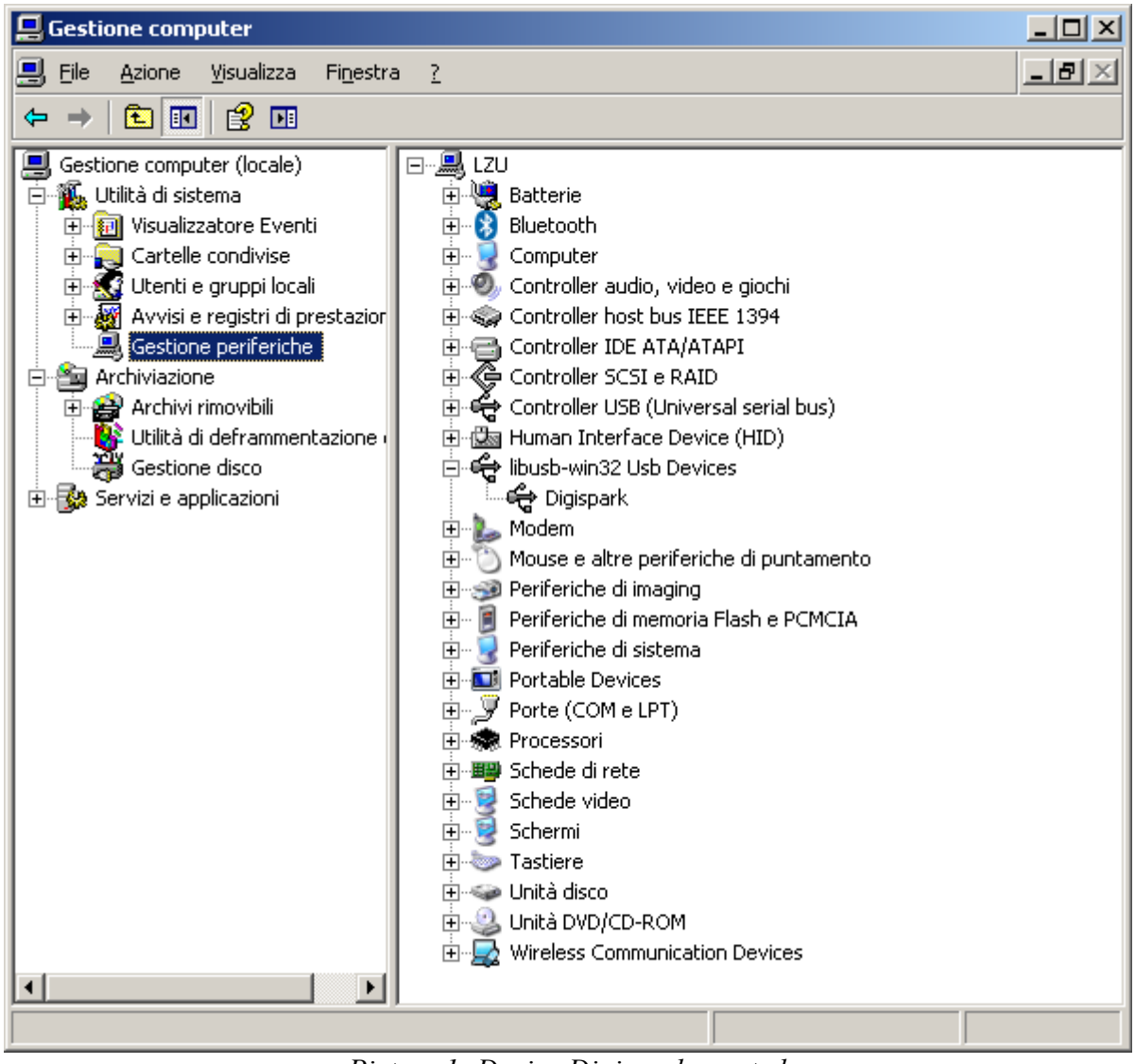
USB connection is intended for Arduino boards that don't have serial connection feature on USB. Actually Ardulink uses this connection for Digispark board (www.digistump.com) and PicoDuino board

(www.tindie.com/products/bobricius/picoduino).

Ardulink USB Connection uses Java libusb / libusb-win32 wrapper (http://libusbjava.sourceforge.net/wp/). This java library is a wrapper for libusb (0.1) (http://www.libusb.org/) and libusb-win32 (http://libusb-win32.sourceforge.net/) USB library.

Please check each library's site to install instructions.
For windows systems you should:

- extract libusb-win32-bin-1.2.6.0.zip
- run inf-wizard.exe
- select Arduino board and finish the wizard
- install USB device using .inf file created



Picture 1: Device Digispark created

Compatibility list

Ardulink is generic and should work with almost any Arduino boards and Arduino clone boards. However, I have tested Ardulink only with some versions of Arduino or Arduino clones.

Actually I tested, in chronological order, Ardulink with:

Board	Site	Ardulink Connection class	Ardulink GUI panel
Arduino UNO	www.arduino.cc	SerialConnection	SerialConnectionPanel
Digispark	digistump.com	DigisparkUSBConnection	DigisparkConnectionPanel (SimpleProtocol)
Arduino Micro	www.arduino.cc	SerialConnection	SerialConnectionPanel
Zigduino r2	logos-electro.com	SerialConnection	SerialConnectionPanel
PicoDuino	On tindie	DigisparkUSBConnection	DigisparkConnectionPanel (SimpleProtocol)
ChipKIT Uno32	chipkit.net	SerialConnection	SerialConnectionPanel
Crowduino With ATMega 328 V1.1	Elecrow	SerialConnection	SerialConnectionPanel
Crowduino Uno-SD V1.4	Elecrow	SerialConnection	SerialConnectionPanel
Bluno V2.0	DFRobot	SerialConnection	SerialConnectionPanel

Ardulink Core

Ardulink is a complete java open source solution to control and coordinate several Arduino boards. It has several software modules (JAR). The first one is Ardulink Core. This jar has the core Ardulink's classes and it have to be included in each java project that intends to use Ardulink technology.

Ardulink SWING

This module contains a collection of ready Ardulink SWING components. These components can be used to write a rich SWING GUI application able to interface with Arduino boards.

Ardulink Console

Ardulink Console is a java desktop application. This application contains all the SWING components from Ardulink SWING module. So Ardulink can be used from non programmer user too.

Ardulink Network Proxy

As explained here:

http://www.ardulink.org/how-control-arduino-from-network/

here:

http://www.ardulink.org/arduino-yun-alternative/

and here:

http://www.ardulink.org/control-and-coordinate-many-arduino-boards/

Ardulink Network Proxy Server is a command line application. It can be used to control Arduino boards over the net or multiplex messages from Arduino boards.

Ardulink Mail

This command line application waits for messages from an emal inbox. It analyzes they and take actions. Of course these actions can interact with Arduino boards. Ardulink Mail is a very flexible and configurable solution. So it can be used in a very wide range of use cases. However an user should understand security risks about email and be very carefull.

Ardulink MQTT

This command line application integrates Ardulink capabilities with an MQTT broker. So Arduino boards can be controlled with simple MQTT messages. There are a lot of ready applications for this protocol even for smart devices like Android and iOS.

Ardulink Sketches

Ardulink needs some code uploaded in Arduino board. Actually there isn't a specific library for Arduino IDE. In Ardulink distribution you can find several sketches that you can use to work with Ardulink. Of course they need for some modifies in order to accomplish your requirements but they are a good start point.

- **ArdulinkProtocol.ino** works with Arduino UNO boards
- **ArdulinkProtocol4Digispark.ino** works with Digispark and PicoDuino using Ardulink default protocol named: ALProtocol. Note: SWING component DigisparkConnectionPanel uses SimpleBinaryProtocol instead

of ALProtocol. I suggest you to use SimpleBinaryProtocol with these boards

- **ArdulinkProtocol4LeonardoAndMicro.ino** this is the most general sketch for Ardulink. It is equal to ArdulinkProtocol.ino but manage input in a different way. It is developed to work with Arduino Micro and Leonardo but you can test it with other Arduino based boards.

- **ArdulinkProtocol4Uno32.pde** this sketch is developed to work with chipKIT Uno32 boards

- **CustomMessagesChipKit.ino** this sketch is developed to work with chipKIT Uno32 boards. It manages custom messages, and of course is just an example. I've used it for this video: https://www.youtube.com/watch?v=PH2ejKGKaoM where Ardulink manages an OLED installed into the chipKIT Basic I/O Shield

- **JoystickCustomMessages.ino** this is the sketch I've developed to test the release v0.4.2 Top Gun. You can see a video here: https://www.youtube.com/watch?v=MErhEvy_NA8

- **JustReading4Uno32.pde** an example that sends messages to Ardulink reading a PIN

- **SimpleProtocol.ino** a sketch to manage the SimpleBinaryProtocol

- **SimpleProtocol4Digispark.ino** a sketch to manage the SimpleBinaryProtocol with a Digispark or PicoDuino