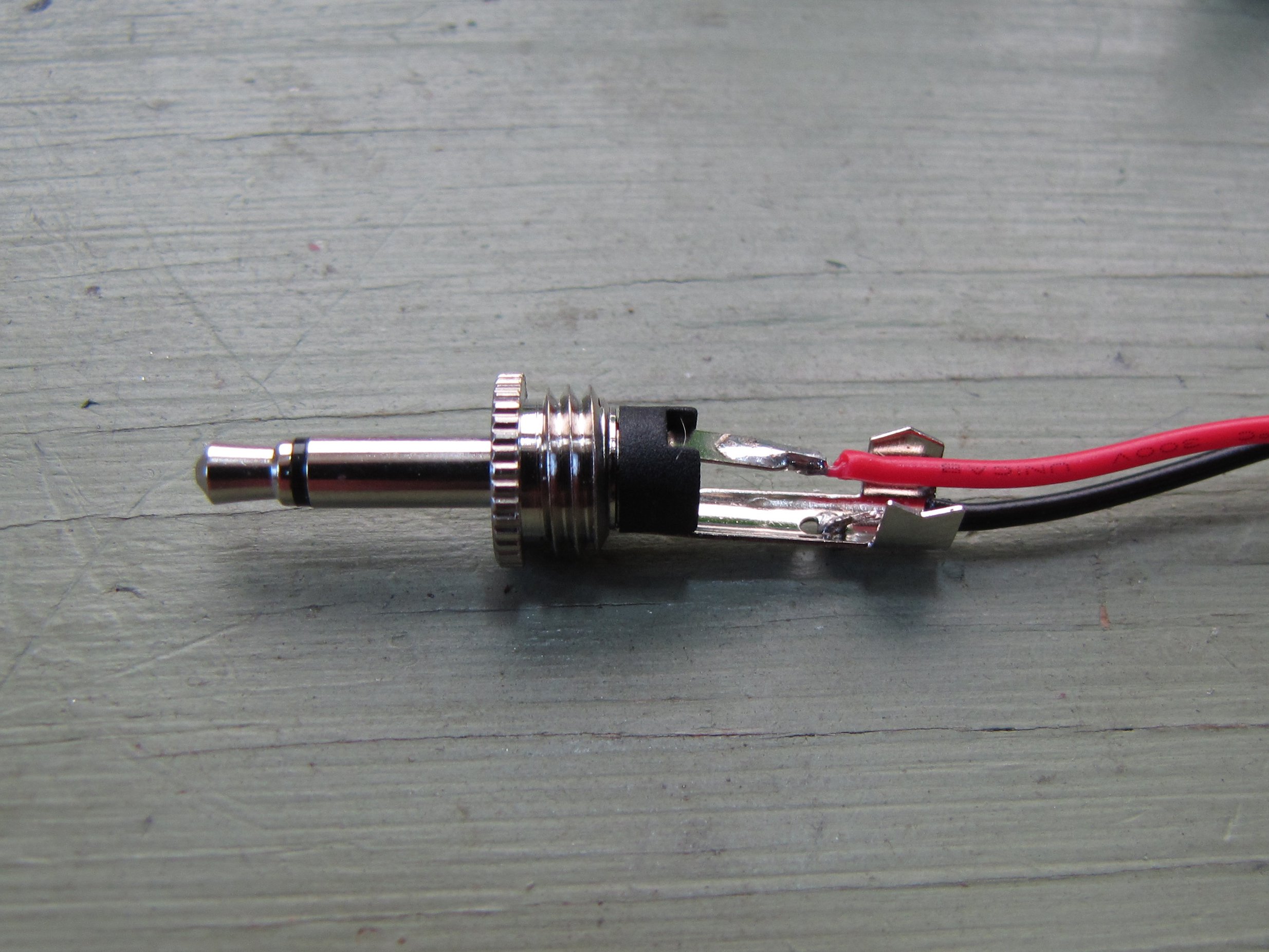
**Layer 1-Inputs and Outputs**

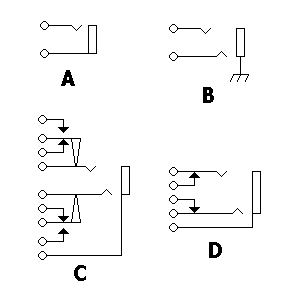
**Inputs**

-Buttons/switch

This type of input is a signal(pulse). Τhe signal can be 1V-5V.

This applies to the buttons/switches with a mono phone plug output. The tip send the signal and the sleeve is the ground. The circuit is shown below:

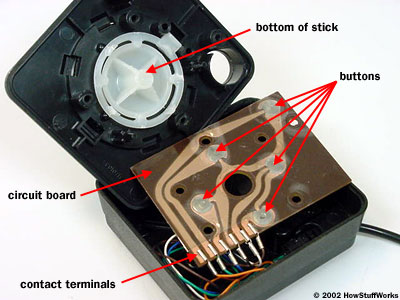




Left the mono phone plug. Right the A the input.

- joystick

The types of joysticks a many. The most common one is the type that works like a combination of buttons.



The button type switch circuit

The output of this is either a usb device or some kind of connector.

-Camera

If a camera is going to be the input, we need some process in this level. Also we need the help of the software in a next level. Also that means that the input layer must have a driver to support a camera.

-Sensors

The type of signal we are going to have depends on the sensor we are going to use. In general is pretty easy to handle the signal of a sensor with a microcontroller.

The signal from from the sensor depends on the type of sensor we are going to use. In general it is from some mVs to some Volts. This signal must be processed

Some sensors are:

**pressure sensor**

A pressure sensor is generally a variable resistor that changes resistance based on how hard it is pressed. They are also known as Force Sensing Resistors or FSRs. They come in many shapes and sizes, from tiny little dots to long strips. They're pretty flexible, and can usually be cut to size.

Often you use an FSR as the variable resistor in a voltage divider circuit. By using one FSR and one fixed resistor and hooking Vout to an A/D converter you can give the microcontroller some idea of how much pressure is being put on the FSR. If you don't have an A/D converter handy, there's a simple RCtime circuit that you can use to do more-or-less the same thing.

FSRs can be used in many places that a button is used, with the advantage that you can not only tell that the button is being pressed, but also how hard. They're used a lot in things like velocity-sensitive keyboards, but are also good for use in floor mats, on the sides of instruments, on finger tips...Basically anywhere there's going to be a force that you want to measure.

**light sensor**

The most common light sensor is the CDC photo sensor. It works like a variable resistor with a value that changes based on how much light it receives. Other types are photo-transistors and photo-diodes.

Photo sensors are used in much the same way as pressure sensors. If you hook one up to a microcontroller via a voltage divider (or RCtime) circuit, the microcontroller can sense how much light is falling on the device.

**beam breaker**

A beam breaker is a non-physical switch or trigger. The simplest way to make one is to aim a cheap laser pointer at a photo-sensor. When the beam is broken, the resistance of the photo-resistor changes. If the photo resistor is hooked up to a microcontroller, it can sense whether the beam is intact or broken. So simple!

**proximity sensor**

There are a variety of ways to implement proximity sensing. The most common are infrared and ultrasound. They're both usually used as packaged devices that hook up to a microcontroller via one or more digital i/o lines.

An infrared sensor works by sending out pulses of infrared (invisible) light. It then tries to detect reflections of that light from nearby objects. If it detects a reflection then it assumes that there's an object nearby and puts out a digital "1". Otherwise it puts out a "0". There are some tricks to get a rough distance measurement out of an infrared detector, but most often they're used for simple object detection.

An ultrasonic proximity detector works by putting out ultrasonic (inaudible) pulses of sound. It then measures the length of time it takes those pulses to hit nearby objects and return as echos. The longer the time, the farther away the object. Ultrasonic proximity detectors are good for measuring short distances (maybe a couple yards). It's usually up to the microcontroller to do the counting and distance computations; the detector takes care of sending and detecting the ultrasonic pulses.

**accelerometer**

Accelerometers are cool. They're small ICs with tiny little suspended weights embedded in them. By measuring the force of gravity/inertia on those weights the device can emit a digital signal that tells you about its orientation in space.

Like proximity sensors, accelerometers are hooked up to a microcontroller via a few digital i/o lines.

**microphone**

Most microcontrollers aren't fast enough to be useful for recording sound via a mic. But they can do some basic things like listen for suddens sounds or listen for silence. They can also be used to control sound recording devices.

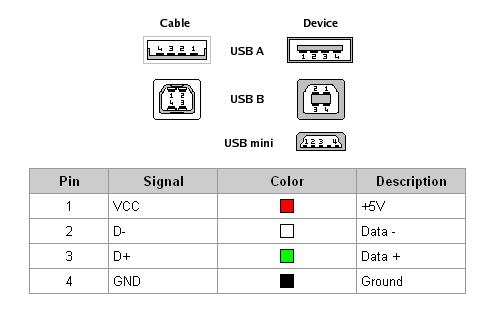
**contact mic**

These are made from piezo elements. A piezo element is made from a curious type of ceramic material that gives off electricity when it's deformed (this is called the piezo-electric effect). The voltage created is (roughly) proportional to how much the element is deformed. This property lets it be used as a contact mic. You can scream all you want into a contact mic and it won't hear you. But rub it against something, and it will wail! There are other types of piezo materials, like piezo rubber.

**Output**

-cable

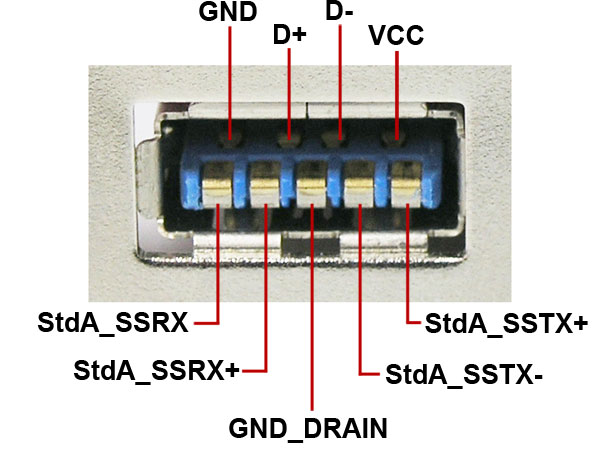
usb 2.0

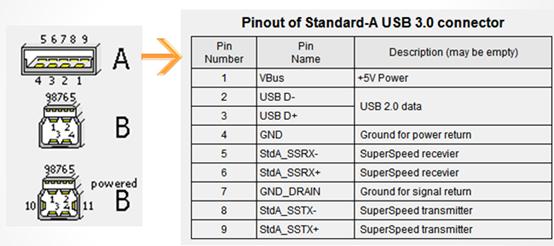


As we can see there are four pins for the usb . Two pins VCC 5v,Ground and two pins for the data (D+,D-).

Max signaling rate 480Mbit/s.

usb 3.0





Throughput 4Gbits/sec.

In conclusion using a usb cable we need a signal in (pulse) and 5V power supply.

-Bluetooth

Bluetooth is a wireless technology standard for exchanging data over short distances (using short-wavelength UHF radio waves in the ISM band from 2.4 to 2.485 GHz[]](https://en.wikipedia.org/wiki/Bluetooth#cite_note-4)) from fixed and mobile devices, and building personal area networks (PANs).

Bluetooth operates at frequencies between 2400 and 2483.5 MHz (including guard bands of 2 MHz at the bottom end and 3.5 MHz at the top).This is in the globally unlicensed (but not unregulated) Industrial, Scientific and Medical (ISM) 2.4 GHz short-range radio frequency band. Bluetooth uses a radio technology called frequency-hopping spread spectrum. Bluetooth divides transmitted data into packets, and transmits each packet on one of 79 designated Bluetooth channels. Each channel has a bandwidth of 1 MHz. Bluetooth 4.0 uses 2 MHz spacing, which accommodates 40 channels. The first channel starts at 2402 MHz and continues up to 2480 MHz in 1 MHz steps. It usually performs 1600 hops per second, with Adaptive Frequency-Hopping (AFH) enabled.

Originally, Gaussian frequency-shift keying (GFSK) modulation was the only modulation scheme available. Since the introduction of Bluetooth 2.0+EDR, π/4-DQPSK (Differential Quadrature Phase Shift Keying) and 8DPSK modulation may also be used between compatible devices. Devices functioning with GFSK are said to be operating in basic rate (BR) mode where an instantaneous data rate of 1 Mbit/s is possible. The term Enhanced Data Rate (EDR) is used to describe π/4-DPSK and 8DPSK schemes, each giving 2 and 3 Mbit/s respectively. The combination of these (BR and EDR) modes in Bluetooth radio technology is classified as a "BR/EDR radio".

Bluetooth is a packet-based protocol with a master-slave structure. One master may communicate with up to seven slaves in a piconet. All devices share the master's clock. Packet exchange is based on the basic clock, defined by the master, which ticks at 312.5 µs intervals. Two clock ticks make up a slot of 625 µs, and two slots make up a slot pair of 1250 µs. In the simple case of single-slot packets the master transmits in even slots and receives in odd slots. The slave, conversely, receives in even slots and transmits in odd slots. Packets may be 1, 3 or 5 slots long, but in all cases the master's transmission begins in even slots and the slave's in odd slots.

The above is valid for "classic" BT. Bluetooth Low Energy, introduced in the 4.0 specification, uses the same spectrum but somewhat differently

Officially Class 3 radios have a range of up to 1 metre (3 ft), Class 2, most commonly found in mobile devices, 10 metres (33 ft), and Class 1, primarily for industrial use cases,100 metres (300 ft). Bluetooth Marketing qualifies that Class 1 range is in most cases 20–30 metres (66–98 ft), and Class 2 range 5–10 metres (16–33 ft).

Table of power and range per class

|  |  |  |  |
| --- | --- | --- | --- |
| Class | Max permitted power  (mW) | Max permitted power  (dB) | Range  (m) |
| 1 | 100 | 20 | 100 |
| 2 | 2.5 | 4 | 10 |
| 3 | 1 | 0 | 1 |

Table of data rate and throughput per Version

|  |  |  |
| --- | --- | --- |
| Version | Data rate | Max application throughput |
| 1.2 | 1 Mbit/s | >80 kbit/s |
| 2.0+EDR | 3 Mbit/s | >80 kbit/s |
| 3.0+HS | 24 Mbit/s | See [Version 3.0 + HS](https://en.wikipedia.org/wiki/Bluetooth#Bluetooth_v3.0_.2B_HS) |
| 4.0 | 24 Mbit/s | See [Version 4.0 LE](https://en.wikipedia.org/wiki/Bluetooth#Bluetooth_v4.0) |

* ZigBee

ZigBee is poised to become the global control/sensor network standard. It has been designed to provide the following features:

* Low power consumption, simply implemented
  + Users expect batteries to last many months to years! Consider that a typical single family house has about 6 smoke/CO detectors. If the batteries for each one only lasted six months, the home owner would be replacing batteries every month!
* In contrast to Bluetooth, which has many different modes and states depending upon your latency and power requirements, ZigBee/IEEE 802.15.4 has two major states: active (transmit/receive) or sleep. The application software needs to focus on the application, not on which power mode is optimum for each aspect of operation.
* Even mains powered equipment needs to be conscious of energy. ZigBee devices will be more ecological than their predecessors saving megawatts at it full deployment. Consider a future home that has 100 wireless control/sensor devices,
  + Case 1: 802.11 Rx power is 667 mW (always on)@ 100 devices/home & 50,000 homes/city = 3.33 megawatts
  + Case 2: 802.15.4 Rx power is 30 mW (always on)@ 100 devices/home & 50,000 homes/city = 150 kilowatts
  + Case 3: 802.15.4 power cycled at .1% (typical duty cycle) = 150 watts
* Low cost to the users means low device cost, low installation cost and low maintenance.
  + ZigBee devices allow batteries to last up to years using primary cells (low cost) without any chargers (low cost and easy installation). ZigBee's simplicity allows for inherent configuration and redundancy of network devices provides low maintenance.
* High density of nodes per network
  + ZigBee's use of the IEEE 802.15.4 PHY and MAC allows networks to handle any number of devices. This attribute is critical for massive sensor arrays and control networks.
* Simple protocol, global implementation
  + ZigBee's protocol code stack is estimated to be about 1/4th of Bluetooth's or 802.11's. Simplicity is essential to cost, interoperability, and maintenance. The IEEE 802.15.4 PHY adopted by ZigBee has been designed for the 868 MHz band in Europe, the 915 MHz band in N America, Australia, etc; and the 2.4 GHz band is now recognized to be a global band accepted in almost all countries.

**ZigBee/IEEE 802.15.4 - General Characteristics**

* Dual PHY (2.4GHz and 868/915 MHz)
  + Data rates of 250 kbps (@2.4 GHz), 40 kbps (@ 915 MHz), and 20 kbps (@868 MHz)
  + Optimized for low duty-cycle applications (<0.1%)
  + CSMA-CA channel access
  + - Yields high throughput and low latency for low duty cycle devices like sensors and controls
  + Low power (battery life multi-month to years)
  + Multiple topologies: star, peer-to-peer, mesh
  + Addressing space of up to:
  + - 18,450,000,000,000,000,000 devices (64 bit IEEE address)
  + - 65,535 networks
  + Optional guaranteed time slot for applications requiring low latency
  + Fully hand-shaked protocol for transfer reliability
  + Range: 50m typical (5-500m based on environment)