http://www.edutechkits.com.br

curso de programação

EdutechKits

arduino

Rua Francisco Negrão, 175 - Curitiba - Paraná - (41) 3079 8752

Arduino

Fernando Michelotti

4 de fevereiro de 2011

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Iniciado como tradução de (ITP, 2010)

Utilizando arduino (ARDUINO, 2010)

Utilizando wiring (WIRING, 2010)

Capítulo 2

Arduino

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments.

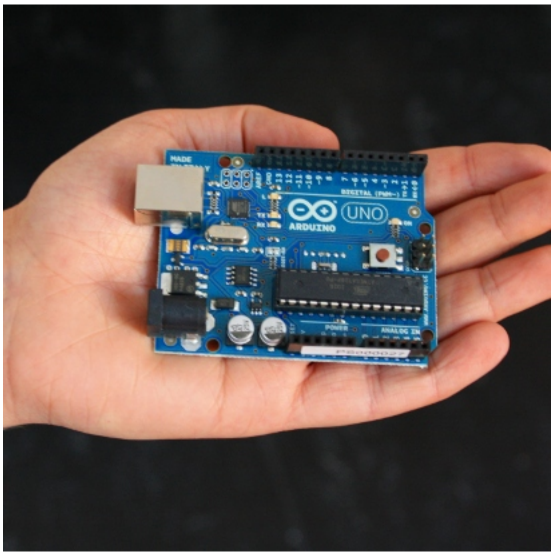

Figura 2.1: Arduino

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

The boards can be built by hand or purchased preassembled; the software can be downloaded for free. The hardware reference designs (CAD files) are available under an open-source license, you are free to adapt them to your needs.

Arduino received an Honorary Mention in the Digital Communities section of the 2006 Ars Electronica Prix. The Arduino team is: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, and David Mellis. Credits

2.1 Programação para micro controladores

2.2 Arduino

Arduino is an open-source electronics prototyping platform, designed to make the process of using electronics in multidisciplinary projects more accessible. The hardware consists of a simple open hardware design for the Arduino board with an Atmel AVR processor and on-board I/O support. The software consists of a standard programming language and the boot loader that runs on the board.

Arduino hardware is programmed using a Wiring-based language (syntax + libraries), similar to C++ with some simplifications and modifications, and a Processing-based IDE.[1]

Currently shipping versions can be purchased pre-assembled; hardware design information is available for those who would like to assemble an Arduino by hand. Additionally, variations of the Italian-made Arduino—with varying levels of compatibility—have been released by third parties.

The Arduino project received an honorary mention in the Digital Communities category at the 2006 Prix Ars Electronica.[2][3]

The project began in Ivrea, Italy in 2005 to make a device for controlling student-built interaction design projects less expensively than other prototyping systems available at the time. As of February 2010 more than 120,000 Arduino boards had been shipped.[4] Founders Massimo Banzi and David Cuartielles named the project after a local bar named Arduino.[5] The name is an Italian masculine first name, meaning "strong friend". The English pronunciation is "Hardwin", a namesake of Arduino of Ivrea

2.3 Platform

2.3.1 Hardware

An official Arduino Duemilanove (rev 2009b).

An Arduino board consists of an 8-bit Atmel AVR microcontroller with complementary components to facilitate programming and incorporation into other circuits. An important aspect of the Arduino is the standard way that connectors are exposed, allowing the CPU board to be connected to a variety of interchangeable add-on modules (known as shields). Official Arduinos have used the megaAVR series of chips, specifically the ATmega8, ATmega168, ATmega328, and ATmega1280. A handful of other processors have been used by Arduino compatibles. Most boards include a 5 volt linear regulator and a 16 MHz crystal oscillator (or ceramic resonator in some variants), although some designs such as the LilyPad run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions. An Arduino's microcontroller is also pre-programmed with a bootloader that simplifies uploading of programs to the on-chip flash memory, compared with other devices that typically need an external chip programmer.

At a conceptual level, when using the Arduino software stack, all boards are programmed over an RS-232 serial connection, but the way this is implemented varies by hardware version. Serial Arduino boards contain a simple inverter circuit to convert between RS-232-level and TTL-level signals. Current Arduino boards are programmed via USB, implemented using USB-to-serial adapter chips such as the FTDI FT232. Some variants, such as the Arduino Mini and the unofficial Boarduino, use a detachable USB-to-serial adapter board or cable, Bluetooth or other methods. (When used with traditional microcontroller tools instead of the Arduino IDE, standard AVR ISP programming is used.)

The Arduino board exposes most of the microcontroller's I/O pins for use by other circuits. The Diecimila, now superseded by the Duemilanove, for example, provides 14 digital I/O pins, six of which can produce PWM signals, and six analog inputs. These pins are on the top of the board, via female 0.1 inch headers. Several plug-in application "shields"are also commercially available.

The Arduino Nano, and Arduino-compatible Bare Bones Board and Boarduino boards provide male header pins on the underside of the board to be plugged into solderless breadboards.

Sortable table Arduino Processor Flash KiB EEPROM KiB SRAM KiB Digital I/O pins ...with PWM Analog input pins Dimensions Diecimila ATmega168 16 0.5 1 14 6 6 2.7"x2.1"Duemilanove ATmega328 32 1 2 14 6 6 2.7"x2.1"Uno ATmega328 32 1 2 14 6 6 2.7"x2.1"Mega ATmega1280 128 4 8 54 14 16 4"x2.1"Fio ATmega328P 32 1 2 14 6 8 1.1"x1.6"Mega2560 ATmega2560 256 4 8 54 14 16 4"x2.1"

2.3.2 Software

The Arduino IDE is a cross-platform application written in Java, and is derived from the IDE for the Processing programming language and the Wiring project. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click. There is typically no need to edit makefiles or run programs on the command line.

The Arduino IDE comes with a C/C++ library called "Wiring"(from the project of the same name), which makes many common input/output operations much easier. Arduino programs are written in C/C++, although users only need define two functions to make a runnable program:

setup() – a function run once at the start of a program that can initialize settings

loop() – a function called repeatedly until the board powers off

A typical first program for a microcontroller simply blinks a LED (light-emitting diode) on and off. In the Arduino environment, the user might write a program like this:

```
#define LED_PIN 13

void setup () {
    pinMode (LED_PIN, OUTPUT);     // enable pin 13 for digital output
}

void loop () {
    digitalWrite (LED_PIN, HIGH);  // turn on the LED
    delay (1000);                  // wait one second (1000 milliseconds)
    digitalWrite (LED_PIN, LOW);   // turn off the LED
    delay (1000);                  // wait one second
}
```

[7] The above code would not be seen by a standard C++ compiler as a valid program, so when the user clicks the "Upload to I/O board"button in the IDE, a copy of the code is written to a temporary file with an extra include header at the top and a very simple main() function at the bottom, to make it a valid C++ program. The Arduino IDE uses the GNU toolchain and AVR Libc to compile programs, and uses avrdude to upload programs to the board. Official hardware

The Arduino Mega2560, uses a surface-mounted ATmega2560, bringing the total memory to 256 kB. It also incorporates the new ATmega8U2 USB chipset.

2.3.3 Open hardware and open source

2.3.4 Main article: Open source hardware

The Arduino hardware reference designs are distributed under a Creative Commons Attribution Share-Alike 2.5 license and are available on the Arduino Web site. Layout and production files for some versions of the Arduino hardware are also available. The source code for the IDE and the on-board library are available and released under the GPLv2 license.[1]

2.3.5 Accessory hardware

2.3.6 A prototyping shield, mounted on an Arduino

Arduino and Arduino-compatible boards make use of shields, which are printed circuit boards that sit atop an Arduino, and plug into the normally supplied pin-headers. These are expansions to the base Arduino. There are many functions of shields, from motor controls, to breadboarding (prototyping).[1]

For example:

- Arduino Ethernet Shield
- XBee Shield
- TouchShield from Liquidware
- Datalog Shield: RTC, SD card storage, temperature sensing, etc. From NuElectronics
- USB Host Shield from Circuits@Home
- Cosmo WiFi Connect from JT5
- A list of Arduino-compatible shields is maintained at the Arduino Shield List website.

Capítulo 3

Componentes

3.1 Voltage Regulator

Voltage regulators take a range of DC voltage and convert it to a constant voltage. For example, this regulator, a 7805 regulator, takes a range of 8 - 15 volts DC input and converts it to a constant 5-volt output.

Note the label on the regulator that reads ”7805”. Check the label on every component. This physical form factor, called the package, is used by many different components, and not all of them are voltage regulators. This is a TO-220 package.

The 7800 series regulators come in many different voltages. 7805 is a 5-volt regulator. 7809 is a 9-volt regulator. 7812 is a 12-volt regulator. All the regulators of this family have the same pin connections. In the image above, the left leg is connected to the input voltage. The middle leg is connected to ground. The right leg is the output voltage.

7805 datasheet

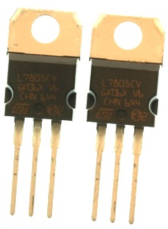

Figura 3.1: 5V voltage regulator

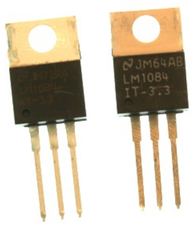

Figura 3.2: 3.3V voltage regulator

3.3V regulators are also common. Note that these ones don't have the same pin configuration as the 7805 regulators!

3.2 LED

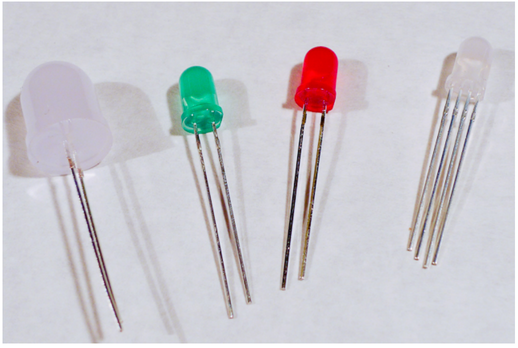

Figura 3.3: LEDs

LEDs, or Light Emitting Diodes, are diodes that emit light when given the correct voltage. Like all diodes, they are polarized, meaning that they only operate when oriented correctly in the circuit. The anode of the LED connects to voltage, and the cathode connects to ground. The anode in the LEDs in this photo is the longer leg on each LED. LEDs come in many diferent packages. The packages above have built-in lenses. These LEDs are the cheapest you can buy, and they're not very bright. You can get superbright LEDs as well, which are much brighter. If you're working on applications that need very small light sources, you can also get LEDs in a surface mount package. LEDs can only handle a limited amount of current and voltage. The details should be covered in each LED's datasheet, but if not, here's a link to a handy LED current calculator. For most common LEDs running at 5 volts, a resistor between 220 and 1K ohms will do the job.

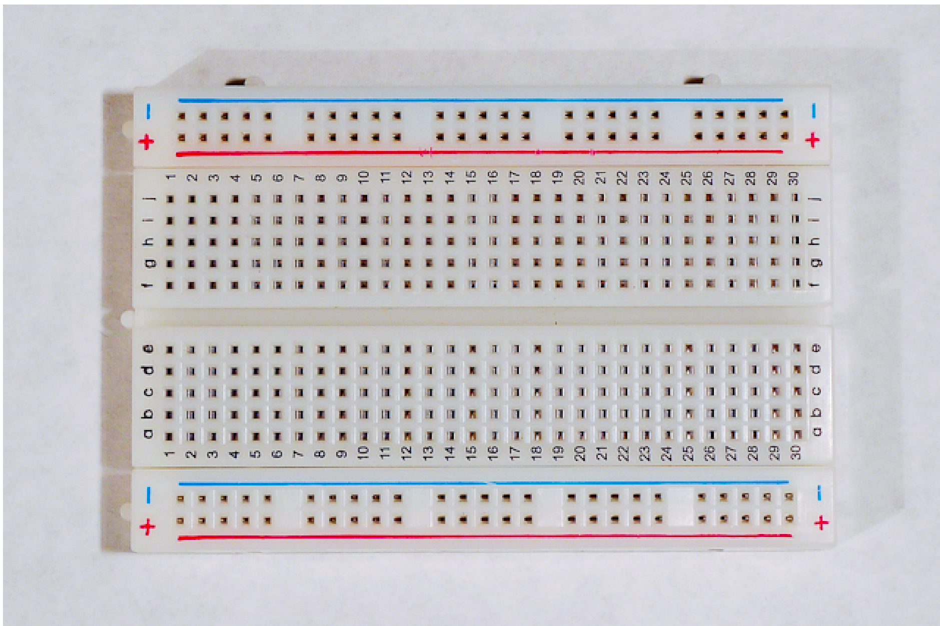

Figura 3.4: solderless breadboard

3.3 Resistors

Resistors resist the flow of electrical current. When placed in series, they reduce the voltage, and limit the current. The bands on a resistor indicate the resistor's value. Here's a handy resistor color code calculator.

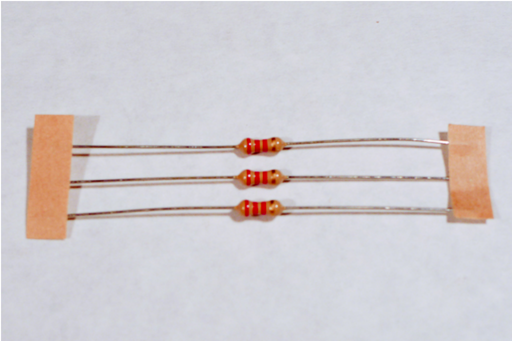

Figura 3.5: resistors

3.4 Potentiometers

Potentiometers are variable resistors. The two outside terminals act as a fixed resistor. A movable contact called the wiper moves across the resistor, producing a variable resistance between the center terminal and either of the two sides.

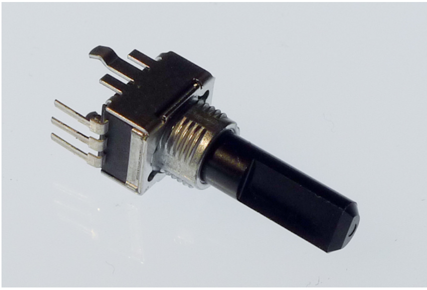

Figura 3.6: potentiometer

3.4.1 trimmer potentiometers

Trimmer potentiometers are designed to be mounted on a circuit board, difficult to turn, so you can use them to adjust a circuit. They're handy to use as physical variables, to tune your project.

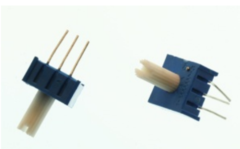

Figura 3.7: trimmer potentiometer

3.5 Switches

Switches are one form of digital input. There are many kinds of switches. The two most useful caategories are **momentary switches**, which remain closed only when you press them, and **toggle switches**, which stay in place after you switch them.

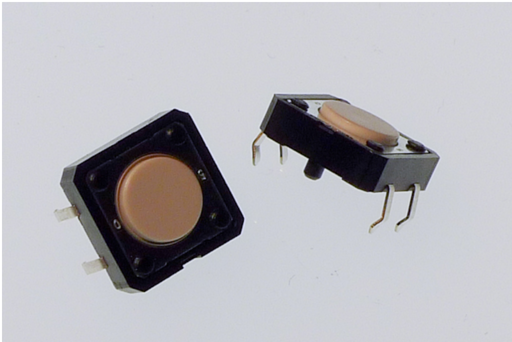

Figura 3.8: momentary switches

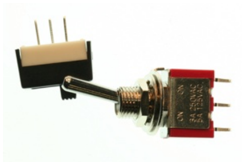

Figura 3.9: toggle switches

3.6 Photocells

Photocells are variable resistors whose resistance changes as the light hitting them changes.

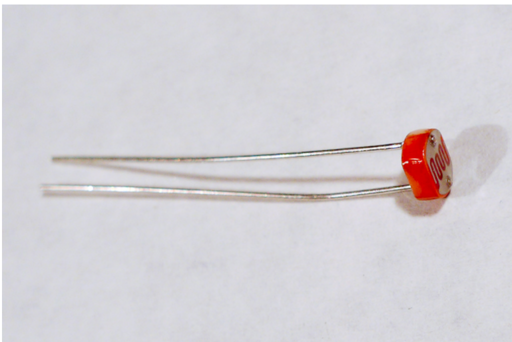

Figura 3.10: photocell

3.7 Thermistors

Thermistors are variable resistors whose resistance changes as the temperature changes.

Figura 3.11: thermistor

3.8 Capacitors

Capacitors store electrical energy while there's energy coming in, and release it when the incoming energy stops. They have a variety of uses. One common use is to smooth out the dips and spikes in an electrical supply. This use is called decoupling.

Figura 3.12: ceramic capacitors

LEDs, or Light Emitting Diodes, are diodes that emit light when given the correct voltage. Like all diodes, they are polarized, meaning that they only operate when oriented correctly in the circuit. The anode of the LED connects to voltage, and the cathode connects to ground. The anode in the LEDs in this photo is the longer leg on each LED. LEDs come in many diferent packages. The packages above have built-in lenses.

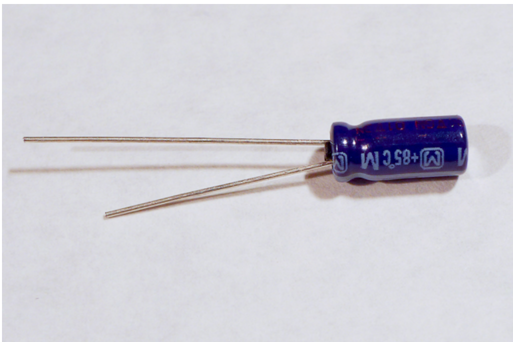

Figura 3.13: electrolytic capacitors

Ceramic capacitors are cheap and unpolarized. They generally have very small capaacitance values. They're useful decoupling caps in a low-current circuit. You often see them used to decouple the power going into a microcontroller or other integrated circuit. The number on a ceramic cap gives you its value and order of magnitude. For example, 104 indicates a 0.1 microfarad (uF) cap. 103 indicates a 0.001 microfarad cap.

Figura 3.14: electrolytic capacitor detail

Electrolytic capacitors can generally store more charge than ceramic caps, and are longer lasting and more expensive. They're usually polarized, meaning that they have a positive leg and a negative leg. This is because current flows more efficiently through them one way than the other.

An electrolytic cap will have a + or - on one side, as shown here.

3.9 Diodes

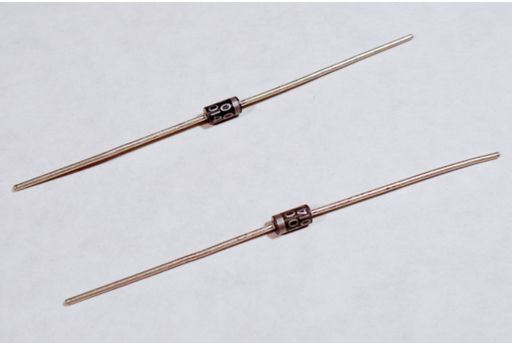

Figura 3.15: 1N4001 diodes

Diodes permit voltage to flow in one direction and block it in the other direction. LEDs are a type of diode, as are the 1N4001 diodes shown here. They're useful for stopping voltage from going somewhere you don't want it to go.

Figura 3.16: zener diodes

Zener diodes have a breakdown voltage past which they allow current to flow in both directions. They're used to chop off excess voltage from a part of a circuit.

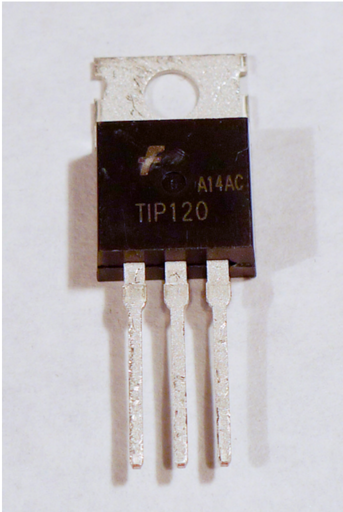

Figura 3.17: transistors

3.10 Transistors

Transistors act as electronic switches. When you put a small voltage across the base and emitter, the transistor allows a larger current and voltage to flow from the collector to the emitter.

3.11 Power Jacks

Figura 3.18: DC power jack, disassembled

Figura 3.19: DC power jack

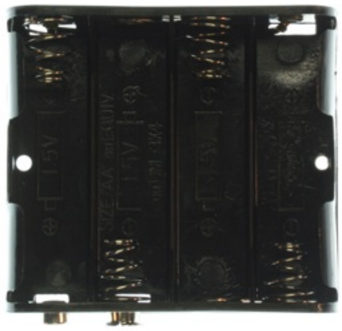

Figura 3.20: AA battery holder

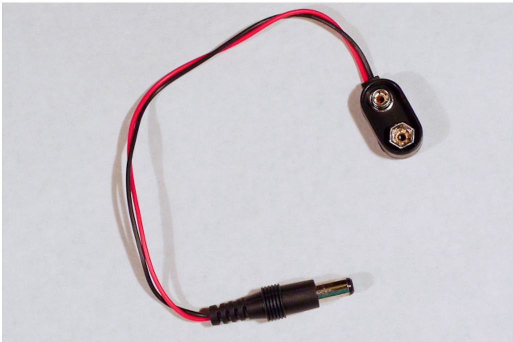

Figura 3.21: 9V battery snap

3.12 Battery Holders

3.13 Motors

3.13.1 Servo Motor

3.13.2 DC Motor

3.14 Gear Kit

3.15 H-Bridge

3.16 Reed Relay

3.17 Screw Terminal

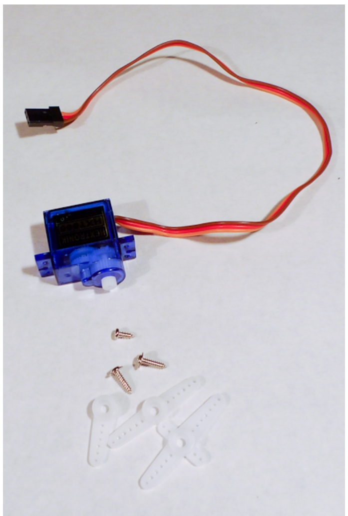

Figura 3.22: servomotor

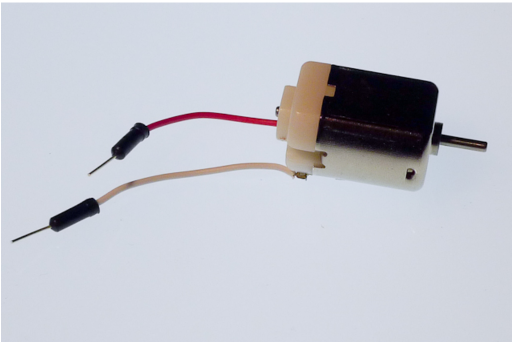

Figura 3.23: DC motor

Figura 3.24: gearbox kit

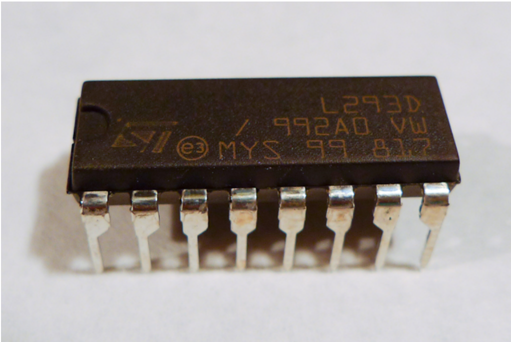

Figura 3.25: H-bridge

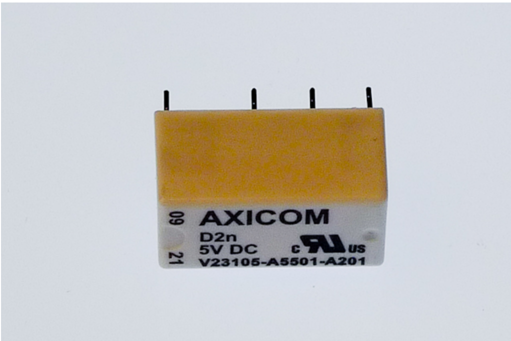

Figura 3.26: reed relay

Figura 3.27: screw terminals

Capítulo 4

Set Up

Capítulo 5

Setting up a breadboard

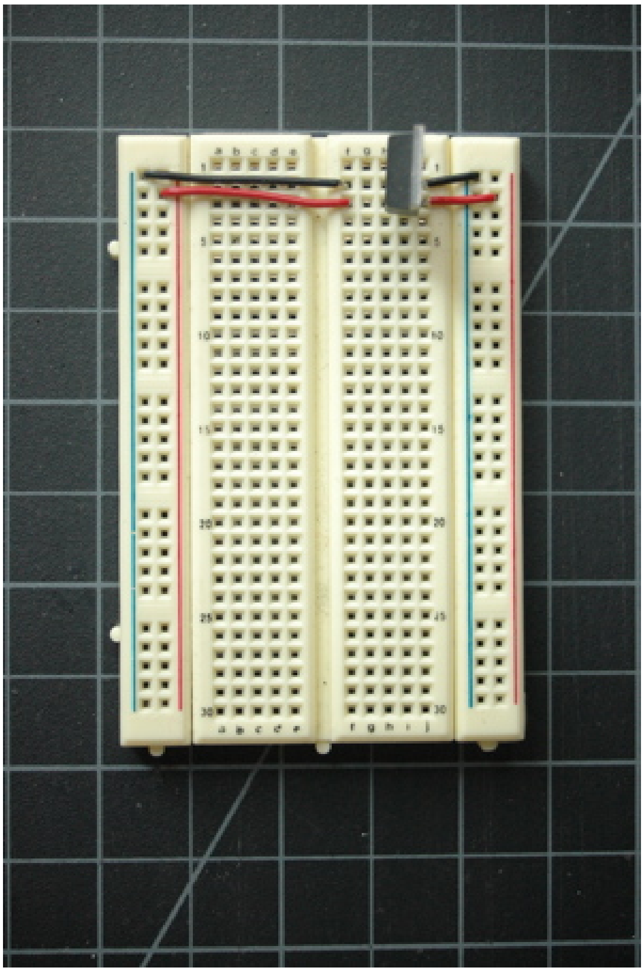

Figura 5.1: breadboard

Solderless beadboards are the quickest tools for prototyping a new circuit. For a detailed description of a breadboard, check these notes.

The picture at left shows a typical breadboard with a 7805 5-volt voltage regulator mounted on it.

There are several rows of holes for components. The holes on the breadboard are separated by 0.1-inch spaces, and are organized in many short rows in the center, and in two long rows down each side of the board. The short horizontal rows in the middle are separated by a center divider. The pattern varies from model to model; some breadboards have only one strip down each side , others have multiple side rows, and some have no side rows.

On each side of the board are two long rows of holes, with a blue or a red line next to each row. All the holes in each of these lines are connected together with a strip of metal in the back. In the center are several short rows of holes separated by a central divider. All of the five holes in each row in the center are connected with a metal strip as well. This allows you to use the holes in any given row to connect components together. To see which holes are connected to which, take a multimeter and a couple of wires, set the multimeter to measure continuity, stick the two wires in two holes, and measure them with the multimeter. If the meter indicates continuity, then the two holes in question are connected.

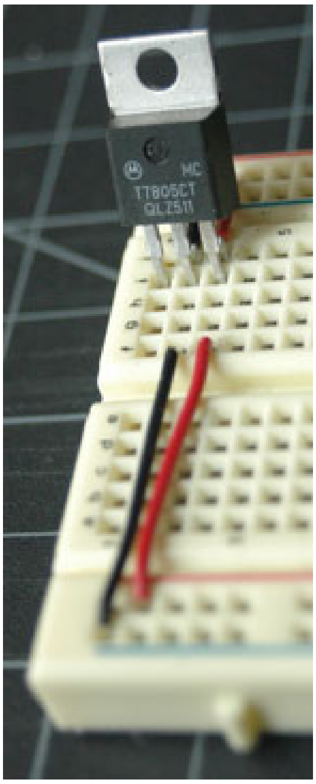

Figura 5.2: breadboard

The reason for the center divider is so that we can mount integrated circuit chips, like a microprocessor, on the breadboard. IC chips typically have two rows of pins that we need to connect other components to. The center row isolates the two rows from each other, and gives us several holes connected to each pin, so we can connect other components.

When you start to put components on your breadboard, avoid adding, removing, or changing components on a breadboard whenever the board is powered. You risk shocking yourself and damaging your components.

The regulator in the picture above is there to supply 5 volts to the two red side rows. The two blue side rows are connected to ground. These will be your power and ground bus rows. They give you lots of convenient places to connect to power or ground as needed. The red and black wires (red for power, black for ground) connect the bus rows to the rows where the regulator's ground and output pins are plugged in. The image at right shows a closeup on the connections to the regulator pins' rows. With your

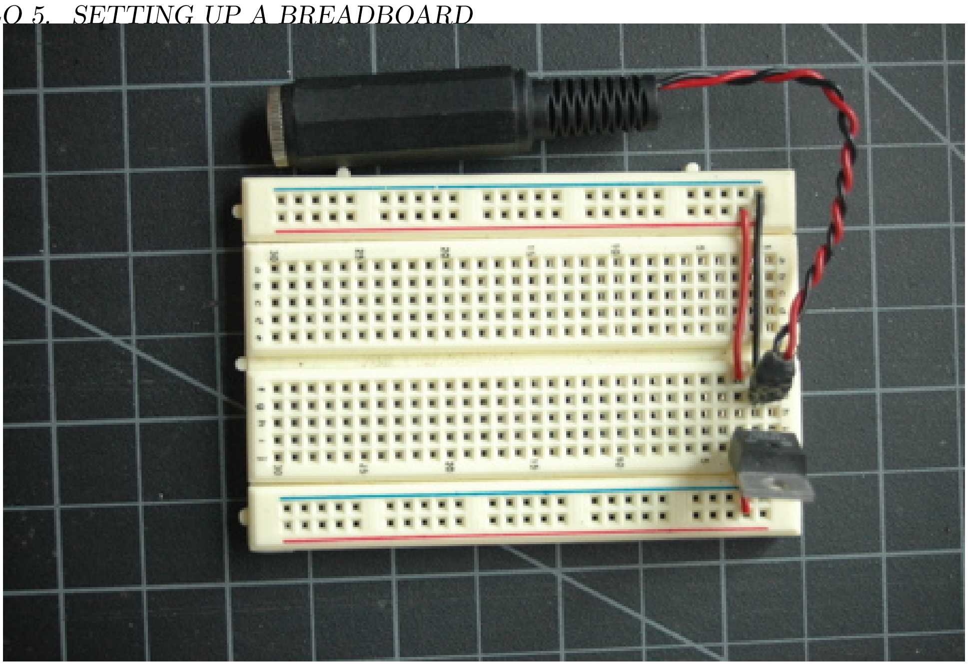

Figura 5.3: breadboard

board connected like this, you'll be able to build many different 5-volt circuits on the board. The last thing you need to add is a power connector to connect 8 - 12 volts DC to supply power for the voltage regulator. The image below shows a power connector connected to the input and ground pins of the voltage regulator.

Capítulo 6

Soldering

6.1 Overview

If you're going to do any electronics work, you're going to have to do some soldering. Many people fear it before they've done it, but it's really pretty simple to do. This exercise will show you how to solder a DC power connector to a set of wires for a breadboard.

6.1.1 Parts

To do it, you'll need:

Figura 6.1: DC Power Jack

Figura 6.2: Header Pins

Figura 6.3: Helping Hands

6.1.2 Preparing the parts

Cut a red and a black wire about four inches in length. Strip the ends back about 1/4 of an inch. Bend a hook on one end of each wire. Unscrew the power jack. Connect the red wire to hole in the the center

Figura 6.4: Soldering Iron

Figura 6.5: Solder

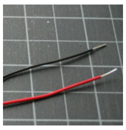

Figura 6.6: 22-AWG hookup wire, in red and black

Figura 6.7: Hot Glue Gun and Hot Glue

Figura 6.8: Wire Strippers

tab of the power, and the black to the hole in the outside tab. Use pliers to crimp the hooked wires to their connections on the jack. If the connections aren't soldered, you'll get an inconsistent connection at best, and a short circuit at worst.

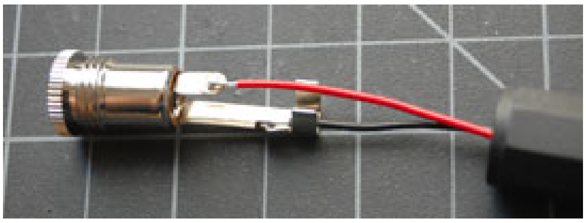

Figura 6.9: power connector unsoldered

6.1.3 Soldering

Touch the iron to the joint between the wire and the metal to heat the joint. Then touch the solder to the joint (NOT to the iron) until it melts. This should make a clean solder with a small blob of solder. Twist the wires together and thread them through the sleeve of the jack.

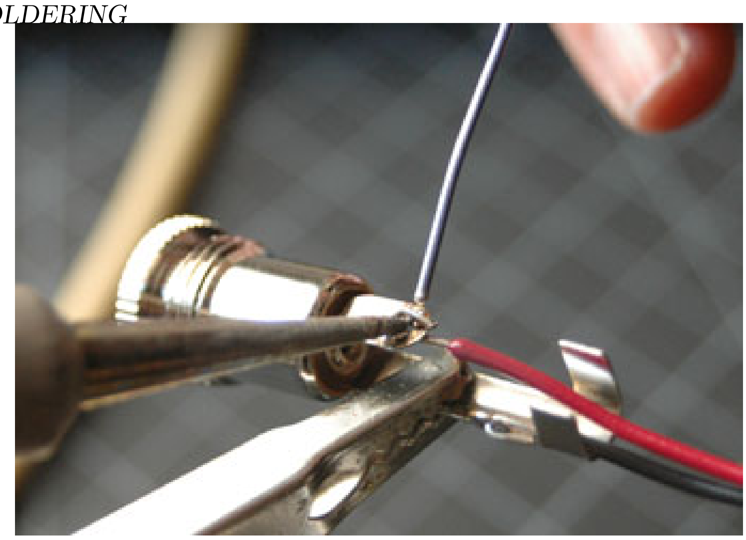

Figura 6.10: soldering power connector

The result should look like this:

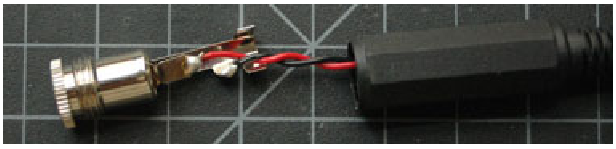

Figura 6.11: power connector inside

Trim the ends of the wires to the same length, and strip them back to about 1/8th of an inch. Break off two header pins and hold them in one clip of the helping hands. Clip the wires in the other clip, and align them with the header pins like so:

Figura 6.12: soldering headers

When you're done soldering, you should have two separate blobs like this:

If you can't see space between the two solder joints, de-solder them and do it again. You need to be sure there's no connection between these wires that can cause a short circuit, and the best way to do that is to leave space. Take a hot glue gun and surround the bare connections to protect them and provide some strain relief. When you're done, you should have a connection like this:

Make sure you can tell which pin is connected to the red wire and which is connected to the black.

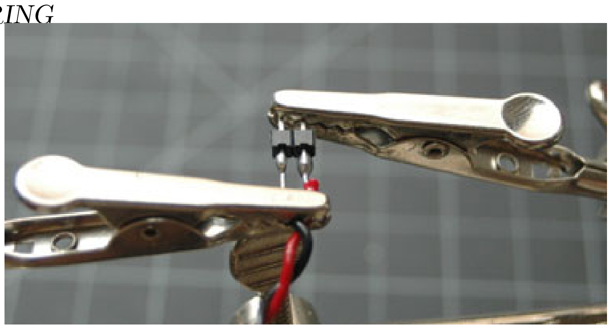

Figura 6.13: soldering headers 2

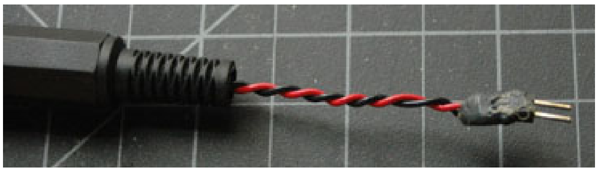

Figura 6.14: power connector headers

6.1.4 Testing

Before you connect it to power, take a meter and check the connections for continuity. The center pin should be connected to the red wire, and the outer rim should be connected to black. When that's good, you're ready to use it.

Capítulo 7

Introduction to electronics

Referências Bibliográficas

ARDUINO. *Arduino.cc*. 2010. http://www.arduino.cc. [Online; accessed 25/11/2010].

ITP. *ITP Physical Computing*. 2010. http://itp.nyu.edu/physcomp/Tutorials/Tutorials. [Online; accessed 25/11/2010].

WIRING. *Wiring.org.co*. 2010. http://www.wiring.org.co. [Online; accessed 25/11/2010].