Arduino

Club de Robótica y Club de Informática:
Héctor David Menéndez Benito

March 11, 2011

Introduction

Arduino is defined as "an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software".

In our case, Arduino will be the brain of the robot. It will take the input information from the sensors and will send the output information to the motors. We are going to use the model Arduino Uno (see figure 1).
For more information about Arduino visit: `http://www.arduino.cc`

Hardware

Arduino Uno has the following features:

- USB connection with the computer.
- 6 Analog inputs.
- 14 Digital Inputs / Outputs where:
  - 6 of PWM.
  - 1 Led connected to the 13^{th} pin.
  - 7 pure digital leds.
- Input Voltage: 7 - 12V.
- Input limit: 6 - 20V.
- Flash memory: 32KB.
- SRAM: 2KB.
- EEPROM: 1KB.
- Clock Speed: 16 MHz.
- Output 5V.
- Output 3.3V
- 3 GND pins.
- Reset Pin.
- Reset button.
- AREF pin for reference pin.

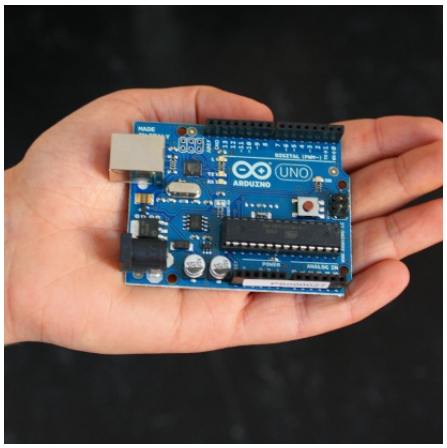


Figure 1: Arduino Uno

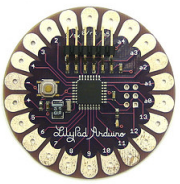
Types of Arduino

The others types of Arduinos that can de used are:

- Arduino Nano:

- Arduino Mega:

- Arduino Lilypad:



- Arduino Bluetooth:

- Arduino Mini:

- Arduino Fio:

Installation

If you want to install the Arduino software you need the following packages:

- avr-gcc-c++: compiles the programs you write for the Arduino.
- avr-gcc.
- avr-libc.
- sun-java6-jre: it is necessary for the user interface.

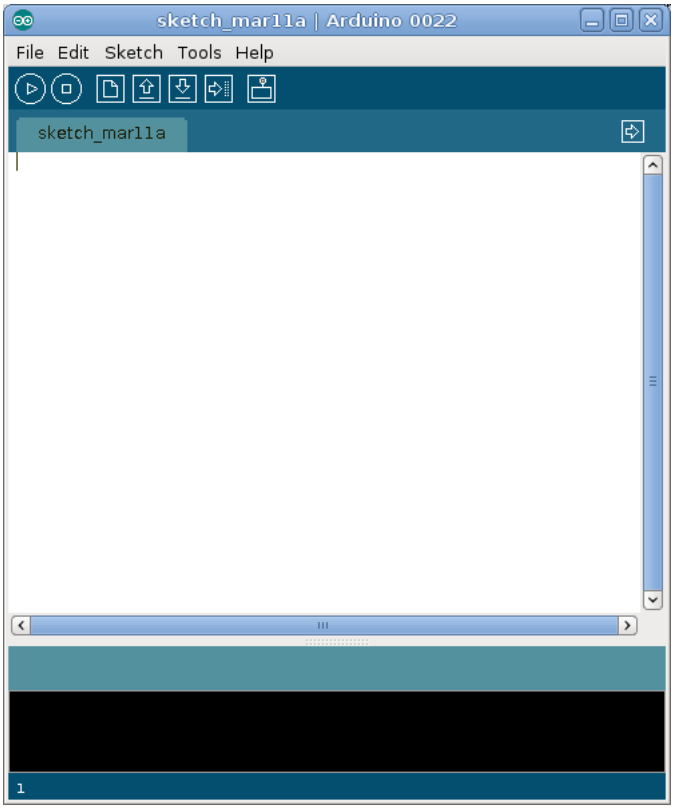
How to use the Software

The first step after download the Arduino software from the web is to run it as:

./arduino

Note: You need to have permission on the USB port.

Then, when you run the software you will find the following interface:



The buttons means, from left to right:

- Play: Verify the code.
- Stop: Stop verification.

- New: New file.
- Open: Open a file.
- Save: Save a file.
- Upload: Upload the program to the Arduino.
- Serial monitor: Use the serial monitor.

When you want to upload a program from the Arduino software to the Arduino Uno, you have to choose in "Tools → Board" Arduino Uno and load the USB device you are using in "Tools → Serial Port".

Programming in Arduino

Basics of Arduino

Arduino has two main functions:

- setup(): This function is used to load the pins, initialization of variables and start with libraries. It only runs once.
- loop(): It's a main function that uses an infinite loop.

Structured programming: Arduino is just C

Arduino use C for the software. You can use structured programming:

- if ... else if ... else; switch case;
- for; while; do ... while.

An non structured programming:

- break; continue; return; goto.

Another features of C syntax:
For C structures:

- # define.
- # include< $\cdots$ >.
- * pointer.
- & reference.

C types:

- void.
- boolean.
- char.
- byte.
- int.

- long.
- float.
- double.

Other:

- string.
- array.

Conversions:

- char().
- byte().
- int().
- long().
- float().

Arithmetic:

- = assignment.
- +. ++.
- -. –.
- *.
- /.
- %.

Comparison:

- == equality.
- !=.
- <. <=
- >. >=

Boolean:

- & &.
- ||.
- !

Arduino Functions:

- Digital I/O:

  - pinMode(int pin, constant mode): set the mode to a pin: INPUT or OUTPUT.
  - digitalWrite(int pin, constant value): set the pin HIGH or LOW.
  - digitalRead(int pin): reads the value of a pin.

- Analog I/O:
  - analogReference(constant type): the types are:
    * DEFAULT: the default analog reference of 5 volts (on 5V Arduino boards) or 3.3 volts (on 3.3V Arduino boards).
    * INTERNAL: an built-in reference.
    * EXTERNAL: the voltage applied to the AREF pin (0 to 5V only) is used as the reference.
  - analogRead(int pin): Reads a value between 0 and 1024. From 0V to 5V or AREF.
  - analogWrite(int pin, int value): Generates a PWM wave from values 0 to 255 depending of the wave.

- Advance I/O:
  - tone(int pin, int frequency, [optional] int duration): generates a square wave of the specific frequency.
  - noTone(int pin): stops a tone.
  - shiftOut(int dataPin,int clock,constant mode,int data):
  - pulseIn(): reads a pulse (time).

- Time:
  - millis(): Millisenconds of running since program starts.
  - micros(): Microseconds of running.
  - delay(int time): Delay.
  - delayMicroseconds(int time): Delay

- Math:
  - min(int x, int y): min between x and y.
  - max(int x, int y): man between x and y.
  - abs(int x): absolute valus of x.
  - constrain(int x, int a, int b): it says x if x is between a and b, a if x is less than a and b if x is bigger than b.
  - map(int value,int fromLow,int fromHigh,int toLow,int toHigh):

```
long map(long x, long in_min, long in_max, long out_min, long out_max)
{
  return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_m
}
```

  - pow(float value, float exponent):

  - sqrt(float value):
  - sin(float value): value is the angle in radians returns the sin.
  - cos(float value):
  - tan(float value):
  - randomSeed(int value): initializes the pseudo-random number generator.
  - random([optional] int min, int max): returns a random number.

- Bytes:
  - lowByte():
  - highByte():
  - bitRead():
  - bitWrite():
  - bitSet():
  - bitClear():
  - bit():

- External Interrupts:
  - attachInterrupt(int numberOfInterrupt,function callBack,constant mode): Set an interrupt: you set a number to the interrupt (numbers 0 (on digital pin 2) and 1 (on digital pin 3)), a function associated as a callback and a mode define with the following contants:
    * LOW: to trigger the interrupt whenever the pin is low.
    * CHANGE: to trigger the interrupt whenever the pin changes value.
    * RISING: to trigger when the pin goes from low to high.
    * FALLING: for when the pin goes from high to low.
  - detachInterrupt(int numberOfInterrupt): Turns off an interrupt

- Interrupts:
  - interrupts(): Enable interrupts.
  - noInterrupts(): Disable interrupts.

- Communication:
  - Serial: It used pins 0 (RX) and 1 (TX) and the USB to communicate with the computer or another board.

Example

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  int sensorValue = analogRead(A0);
  digitalWrite(13, HIGH);   // set the LED on
  delay(sensorValue*25);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(sensorValue*25);              // wait for a second
}
```