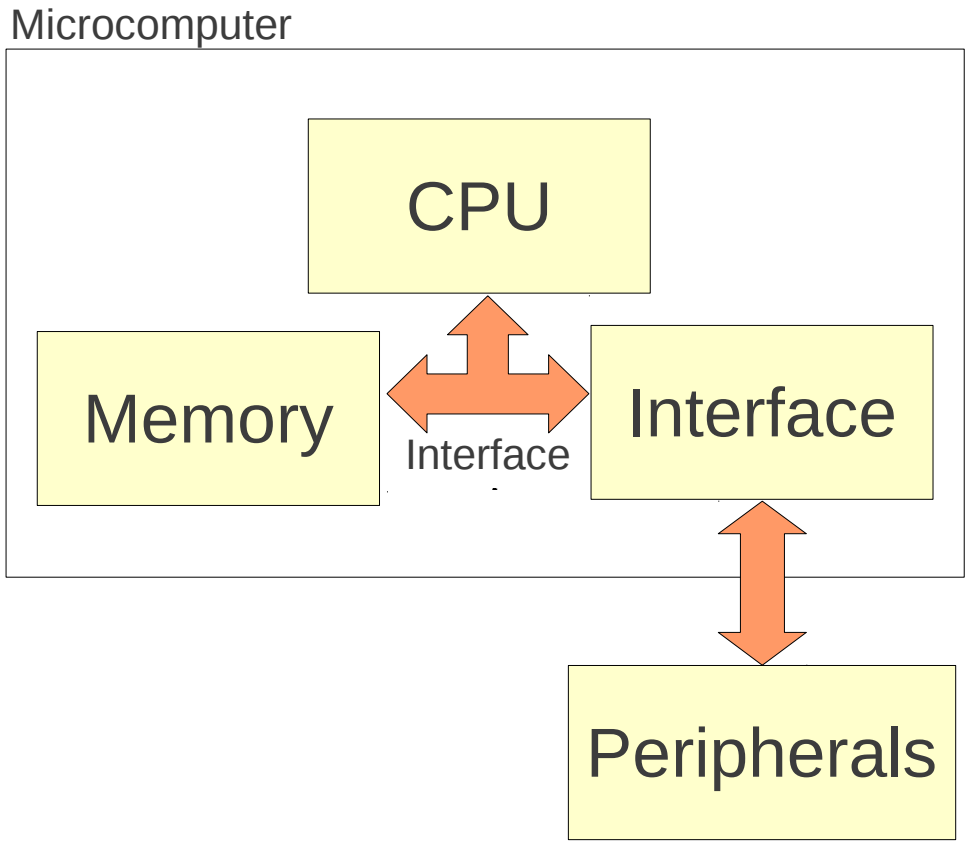
Christian Ackermann

Technische Universität München

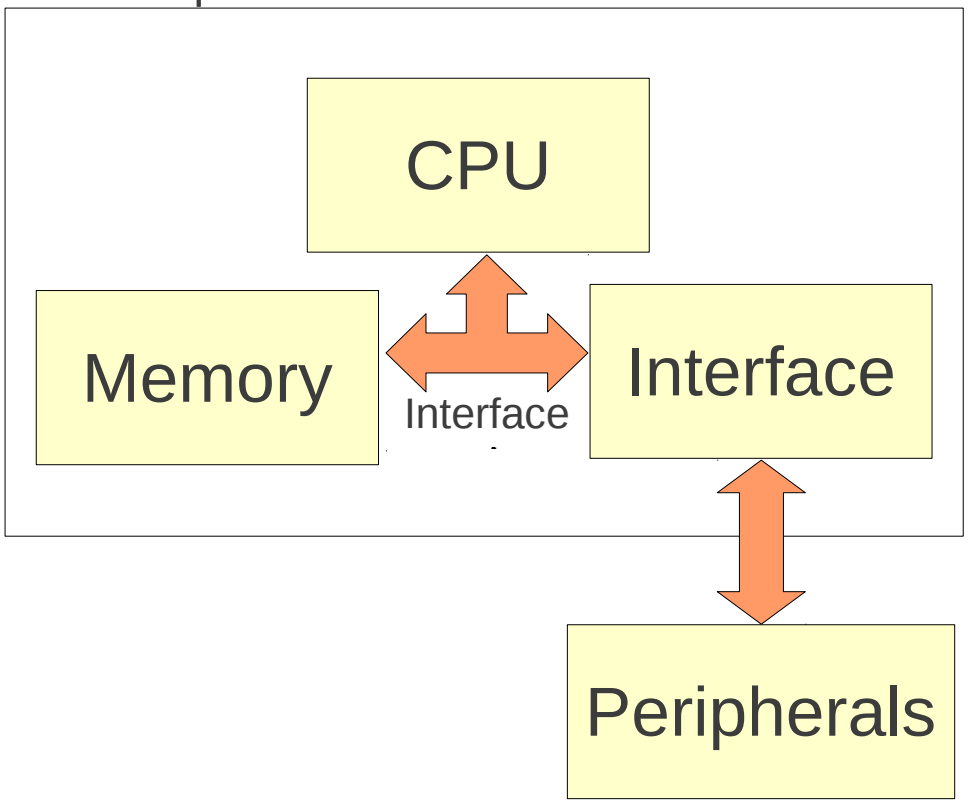
Institut für Astronomische und Physikalische Geodäsie

Differences between microcomputer and microcontroller



Differences between microcomputer and microcontroller

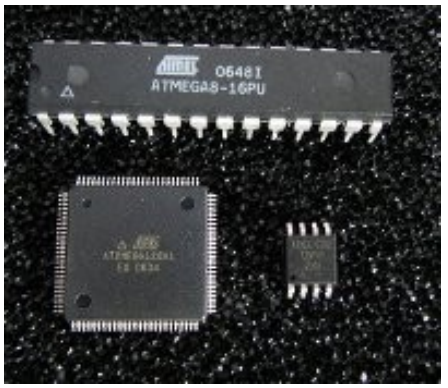
Microcomputer



Microcontroller

Is a semiconductor chip containing a processor core, memory and programmable input/output interfaces

For example some Atmel AVR microcontrollers:



Communication with the micocontroller

Hardware:

A single MC is not usable. One need a circuit in which the MC is embedded.
This circuit manages the communication between the peripherals and the MC.
E.g. the Arduino microcontroller boards

Software:

To program the MC one need a programming language, a compiler.

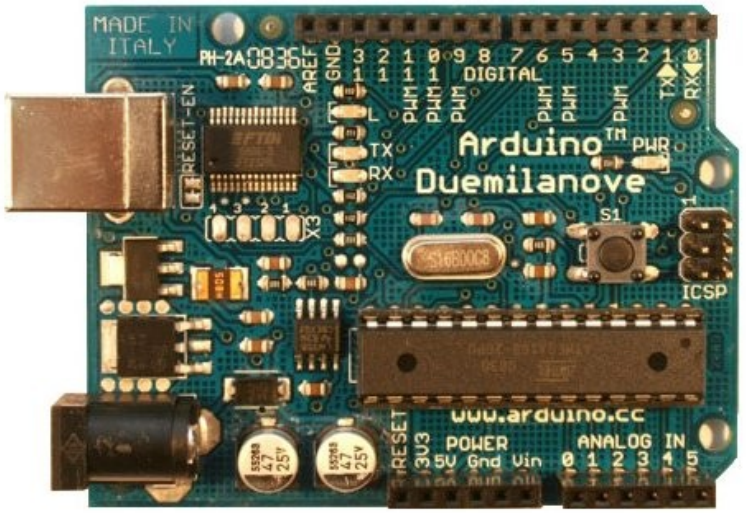
Possible languages are:

- Assembler
- C
- Pascal
- Basic
- Forth

We will use the Arduino development environment. The syntax for the integrated development environment is C like.

Arduino Duemilanove

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	2 KB (Atmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

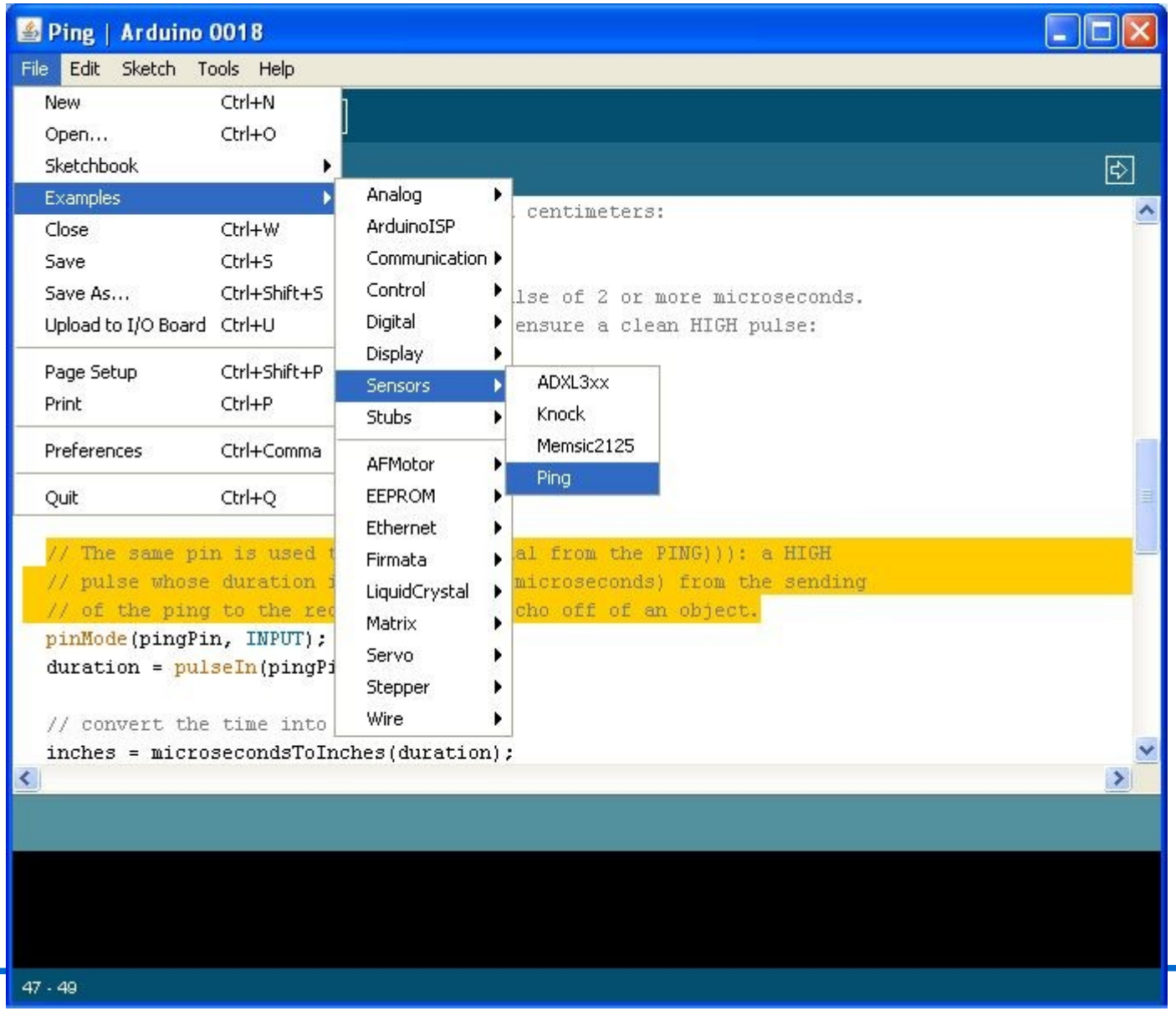


Programming

The Arduino Duemilanove can be programmed with the Arduino software.

One can upload new code to the MC without the use of an external hardware programmer. It communicates using the original STK500 protocol.

Ping programming



Ping programming

```
// ping declaration
const int pingPin = 13;
const int DurationInCm = 58;
long duration = 0;
long distance = 0;

 // initialize serial communication:
void setup() {
   Serial.begin(9600);
}
```

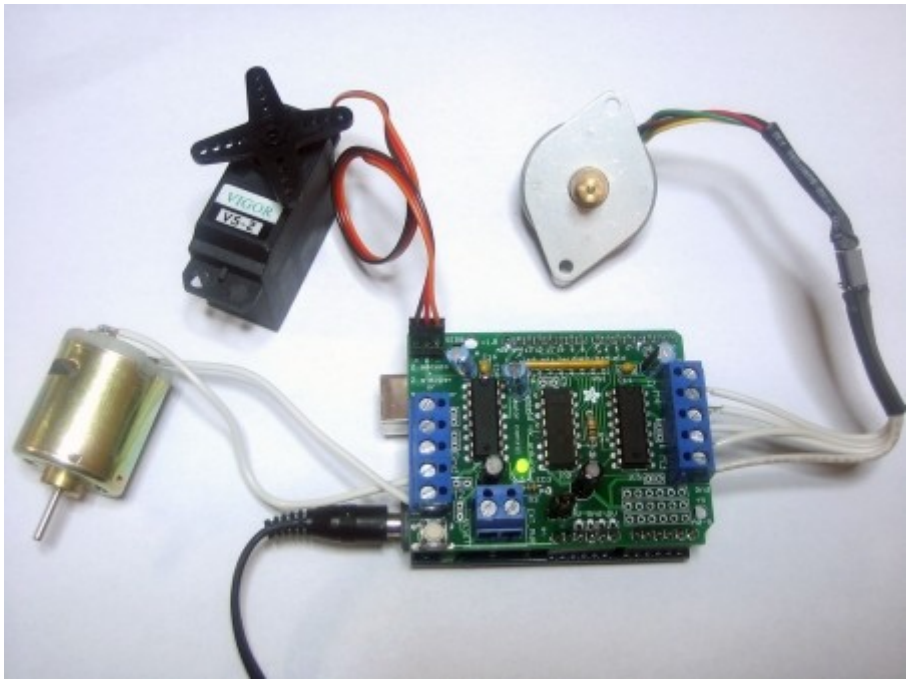
Ping programming

```
// ping usage
// The PING))) is triggered by a HIGH pulse of 2 or more microseconds.
// Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
void loop()   {
      pinMode(pingPin, OUTPUT);
      digitalWrite(pingPin, LOW);
      delayMicroseconds(2);
      digitalWrite(pingPin, HIGH);
      delayMicroseconds(10);
      digitalWrite(pingPin, LOW);

// The same pin is used to read the signal from the PING))): a HIGH
// pulse whose duration is the time (in microseconds) from the sending
// of the ping to the reception of its echo off of an object.
      pinMode(pingPin, INPUT);
      duration = pulseIn(pingPin, HIGH);
      distance = duration / DurationInCm;

      Serial.print(distance);
      Serial.println("cm");              }
```

Programming stepper motors



Motor Shield

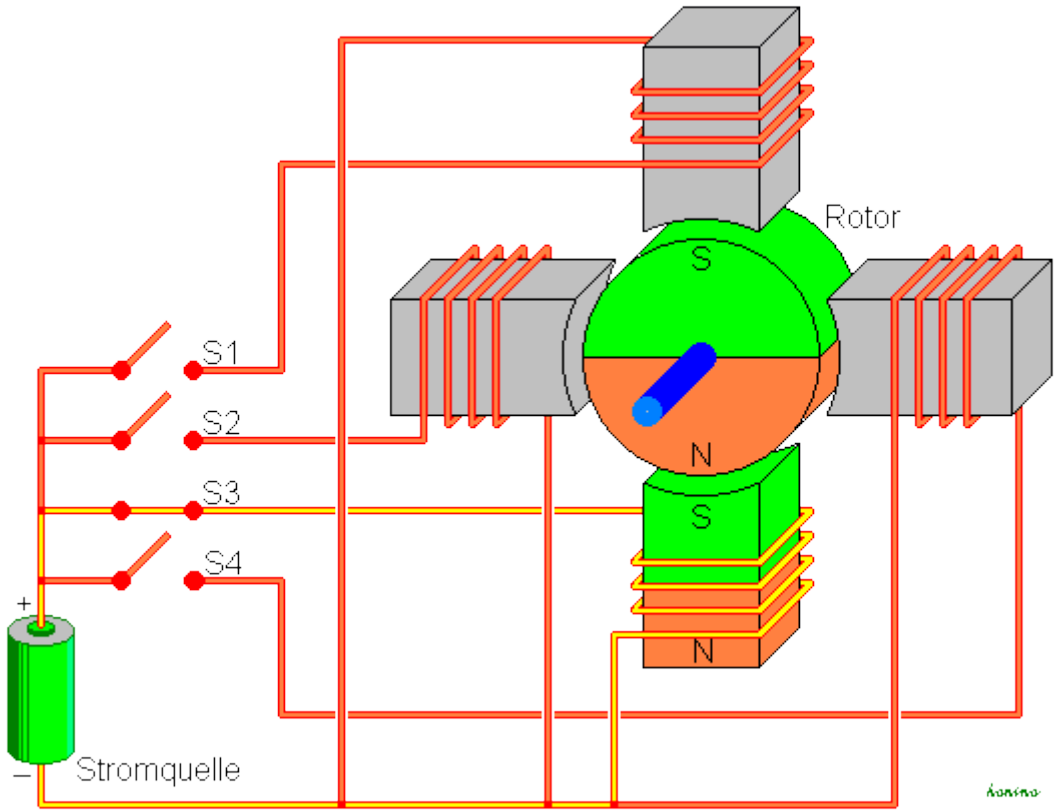
- 2 connections for servos
- 4 bi-directional DC motors
- 2 stepper motors

http://www.ladyada.net/make/mshield/use.html

Programming stepper motors

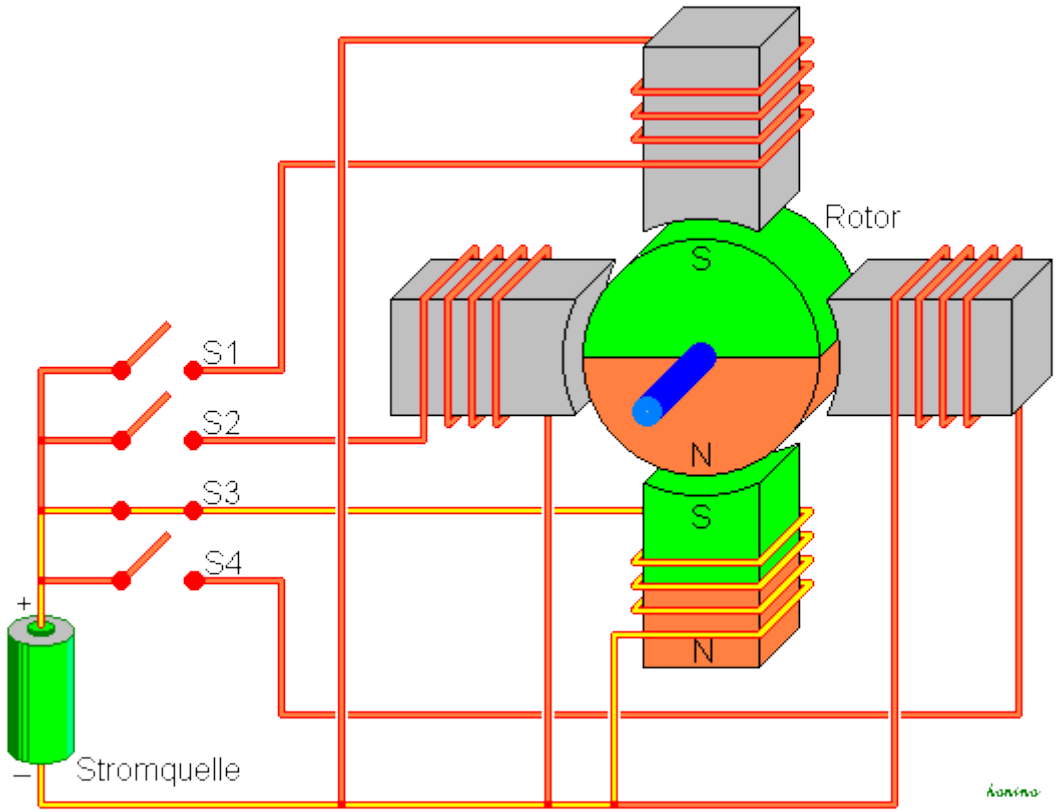
STEP ANGLE	1.8° (200 Steps/Rev.)
PHASES	2
VOLTAGE	12 V
CURRENT	0.33 A
WEIGHT	0.20 Kg

Programming stepper motors



Programming stepper motors

Programming stepper motors



Programming stepper motors

```
#include <AFMotor.h>

AF_Stepper motor(48, 2);                    // create a motor object
                                            // 48 are the steps per revolution
                                            // 2 is the port

void setup() {
  Serial.begin(9600);                       // set up Serial library at 9600 bps
  Serial.println("Stepper test!");

  motor.setSpeed(10);                       // revolution per minute

  motor.step(100, FORWARD, SINGLE);         // 100 = how many steps to take
                                            // move FORWARD or BACKWARD
                                            // SINGLE= step type
                                            // also available is DOUBLE, INTERLEAVE
                                            // or MICROSTEP

  delay(1000);
}
```

Programming stepper motors

```
// example void loop:
void loop() {
  motor.step(100, FORWARD, SINGLE);
  motor.step(100, BACKWARD, SINGLE);

  motor.step(100, FORWARD, DOUBLE);
  motor.step(100, BACKWARD, DOUBLE);

  motor.step(100, FORWARD, INTERLEAVE);
  motor.step(100, BACKWARD, INTERLEAVE);

  motor.step(100, FORWARD, MICROSTEP);
  motor.step(100, BACKWARD, MICROSTEP);
}
```