Politecnico di Torino

Department of Electronics and Telecommunications

Neuronica Lab

The Axle

Francesco Rugiano - 2013

Customer requirements

- The purpose of this study is to create a control and sensing system for a new and innovative military axle that was developed in order to overcome the limits of today vehicles.
- The main features of the system are:
  - Two wheeled robot, driven by DC or brushless motors.
  - The body should be able to stand horizontal even without a foothold (inverted pendulum).
  - It must communicate wirelessly with the base station (PC) and must be remotely controlled.
  - It should have different sensors (e.g. GAS).

Technical specifications

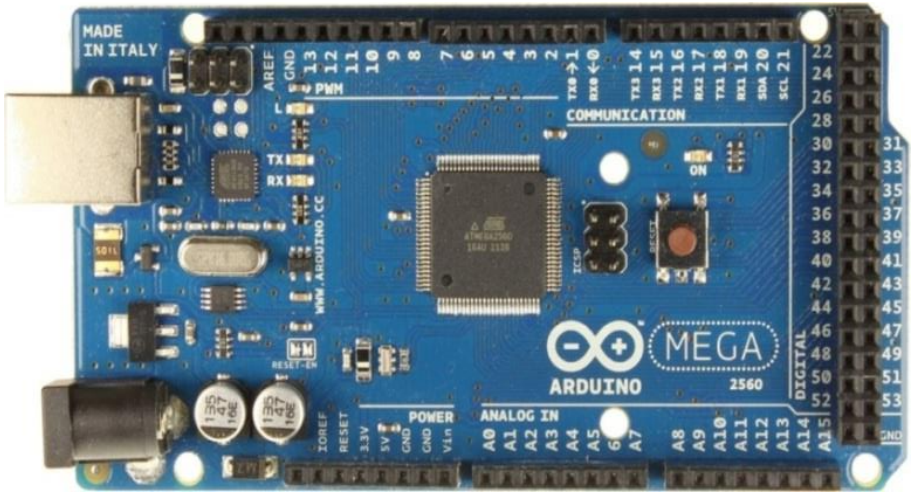
- Since the mechanical part is being developed in another department, we will focus on the electronic control system.
- The inverted pendulum part requires using at least a 3-axial accelerometer or a gyroscope.
- Since the inverted pendulum algorithm is quite well known but depends on the mechanical body, it won't be implemented in the prototype but, once the mechanical part is developed, the electronic part will be able to use the algorithm to balance it.

Technical specifications

- As for the motors, usually big DC motors and brushless motors are driven through a dedicated controller (which can be controlled by a PWM signal). So the motor interface should be a PWM signal.
- The wireless communication will not use Bluetooth or other complex interfaces, but just a 433MHz GFSK modulation, which is simpler to handle and consume less power.
- The test sensors will be a gas sensor (Methane, Butane, LPG, smoke) and a pressure and temperature sensor.

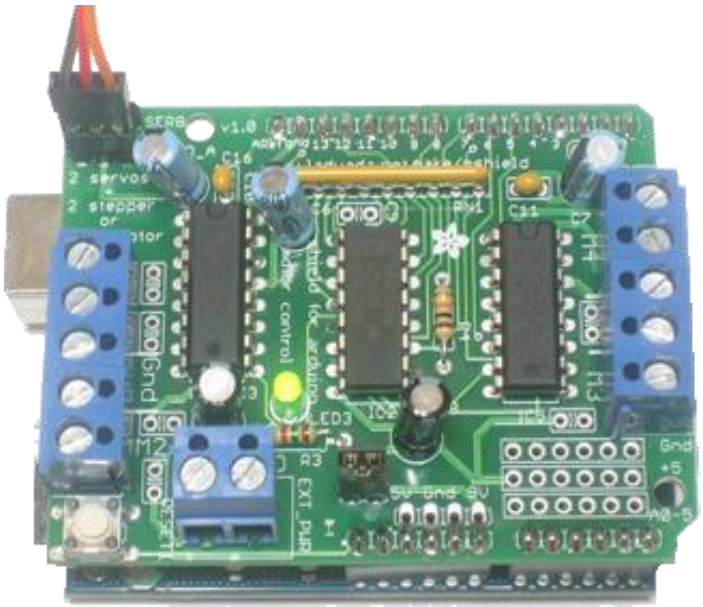
Components choice

- The core of the robot is a Arduino Mega 2560. This has been chosen because it is more powerful than the Arduino UNO and has got more IO ports. Moreover it has four serial connections (instead of just one), which simplifies the wireless communications.



Components choice

- The motor controlling circuit is not built from scratch, but it is a pre-assembled circuit (the Adafruit Motor Shield).
- The wireless interface is a APC220, which exposes a serial interface and can set a 1 km long link.



Components choice

- For balancing, a 9DOF sensors module was used; it incorporates an accelerometer, a gyroscope and a magnetometer (all of them are 3-axial).
- The gas sensor is a MQ-2.
- An HC-SR04 module (ultrasonic distance meter) was also added.
- As for the pressure and temperature, the 9DOF module incorporates also a sensor.

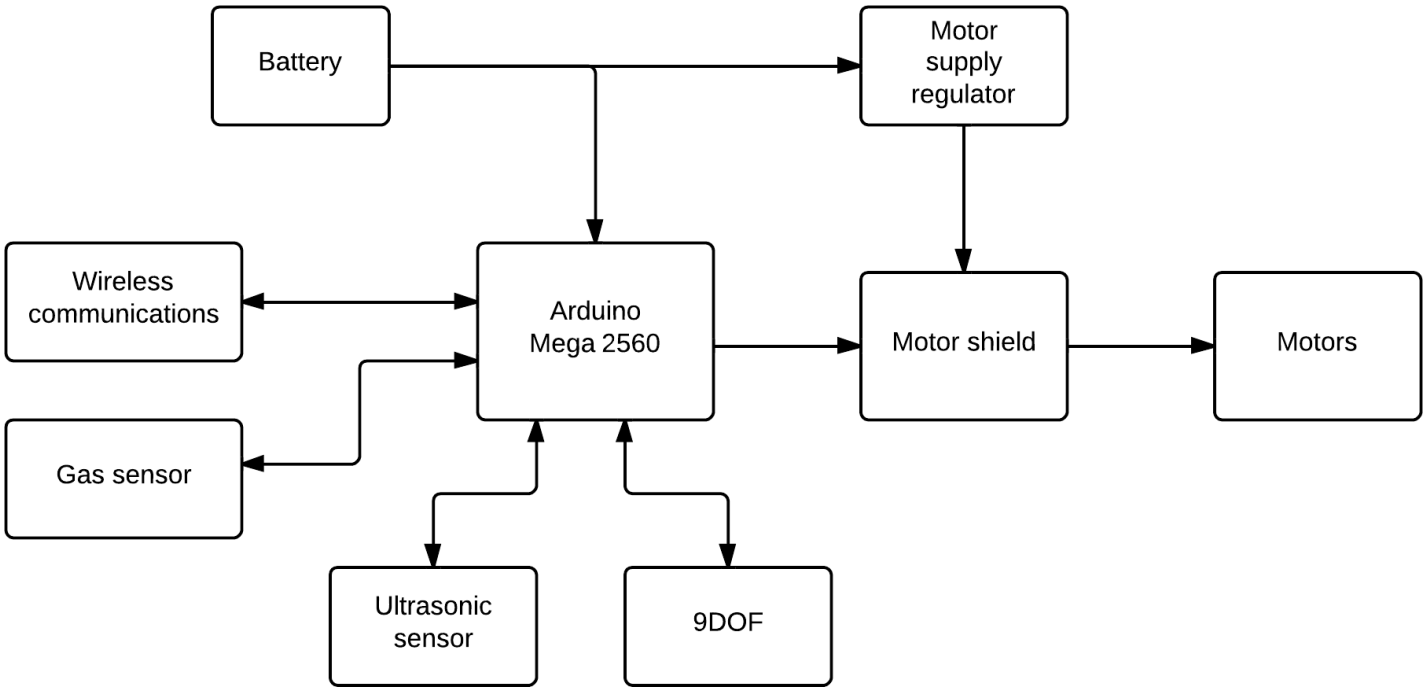
Components choice

- As for the power supply, a 9V battery was chosen.
- The Arduino has got a power supply onboard, so we can connect the battery directly to the board. It then power the sensors.
- The motors, however, can't be powered by the onboard supply; for this reason a separate power supply (5V, using a 7805) was made.

Components choice

- Since the body is not available and since having a big prototype is not needed, a small chassis was chosen, so that also the motor shield can be tested.
- Among the others, the Dagu Magician chassis was chosen; it is moved by two small DC motors.

Block diagram



The firmware

- The firmware is divided in a library, containing the "drivers" for the sensors (i.e. the code to read or write data) and the motors, and a main sketch.
- The sketch is pretty straightforward; it reads the serial buffer, than executes the action according to the command received: if it is a movement command, sets the motors so that it can move in the appropriate direction; if it is the "read sensors" command, it starts reading all the sensors and then sends them through the serial port

The firmware

```
void loop(){
  inData = Serial1.read();
  if ((inData >= '0') && (inData <= '9')) { // Motor speed
    pwm = inData - '0';
    motorLeft.SetMotorSpeed(pwm*255/9);
    motorRight.SetMotorSpeed(pwm*255/9);
  }
  if (inData == 'w') go_forward();
  if (inData == 'x') go_reverse();
  if (inData == 'd') go_right();
  if (inData == 'a') go_left();
  if (inData == 's') stop_car();
  if (inData == 'S') { // Get sensor data
    sens10DOF.GetTempPress(&temperature, &pressure);
    Serial1.print("T");
    Serial1.print(temperature);
    Serial1.print("P");
    Serial1.print(pressure);
    distance = HCSR04.GetDistance();
    Serial1.print("D");
    Serial1.print(distance);
    MQ2.GetGasPpm(&GL, &GC, &GS);
    Serial1.print("GL"); //LPG Liquefied petroleum gas,
    Serial1.print(GL);
    Serial1.print("GC"); //CO
    Serial1.print(GC);
    Serial1.print("GS"); //SMOKE
    Serial1.print(GS);
    ...
```

The firmware

- The library contains four different modules:
  - Motor, which controls one of the motors;
  - Sensor10DOF, which communicates with the accelerometer/gyroscope/magnetometer;
  - SensorGas, which handles the MQ2 sensor;
  - SensorUltrasonic, which calculates the distance using the HC-SR04 sensor.
- Moreover there is a class (ErrorCodes) which handles the communication codes of the modules.

The software

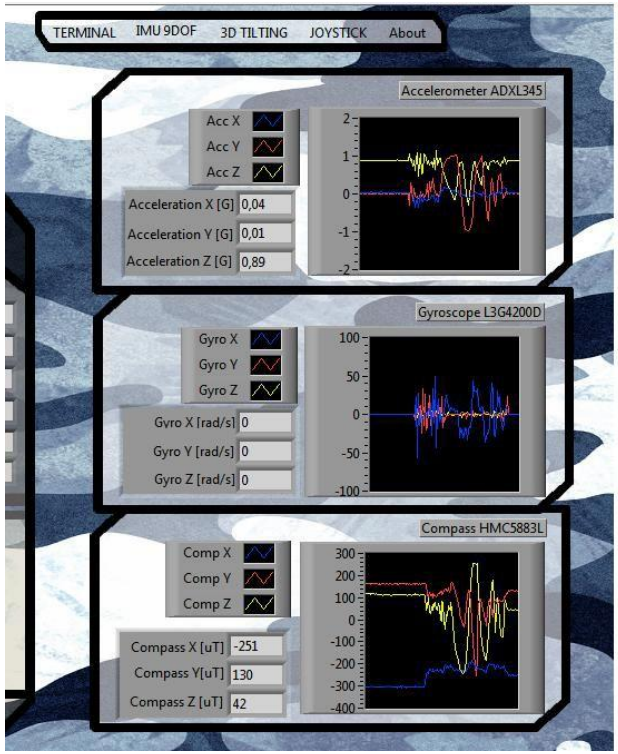
- Since the robot should communicate with a PC, a Labview program was developed.

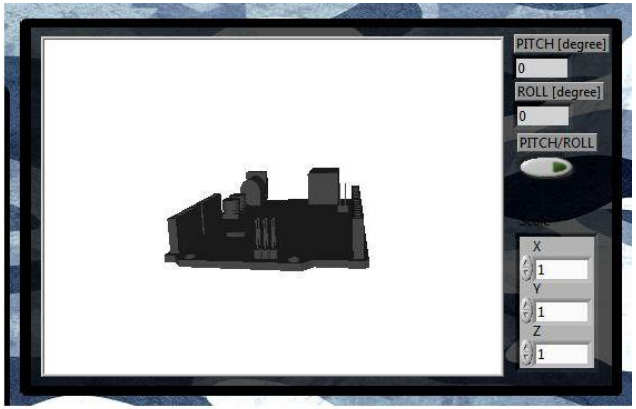
The software

- This virtual instruments allows the user to read the sensors data (gas concentration, pressure, temperature and distance from the nearest object) and view the graphs for the accelerations, angular velocities and magnetic fields; using this data it then reconstructs the robot orientation and shows it to the user using a 3D model.
- The robot can be controlled through the buttons on the interface or using a joypad.

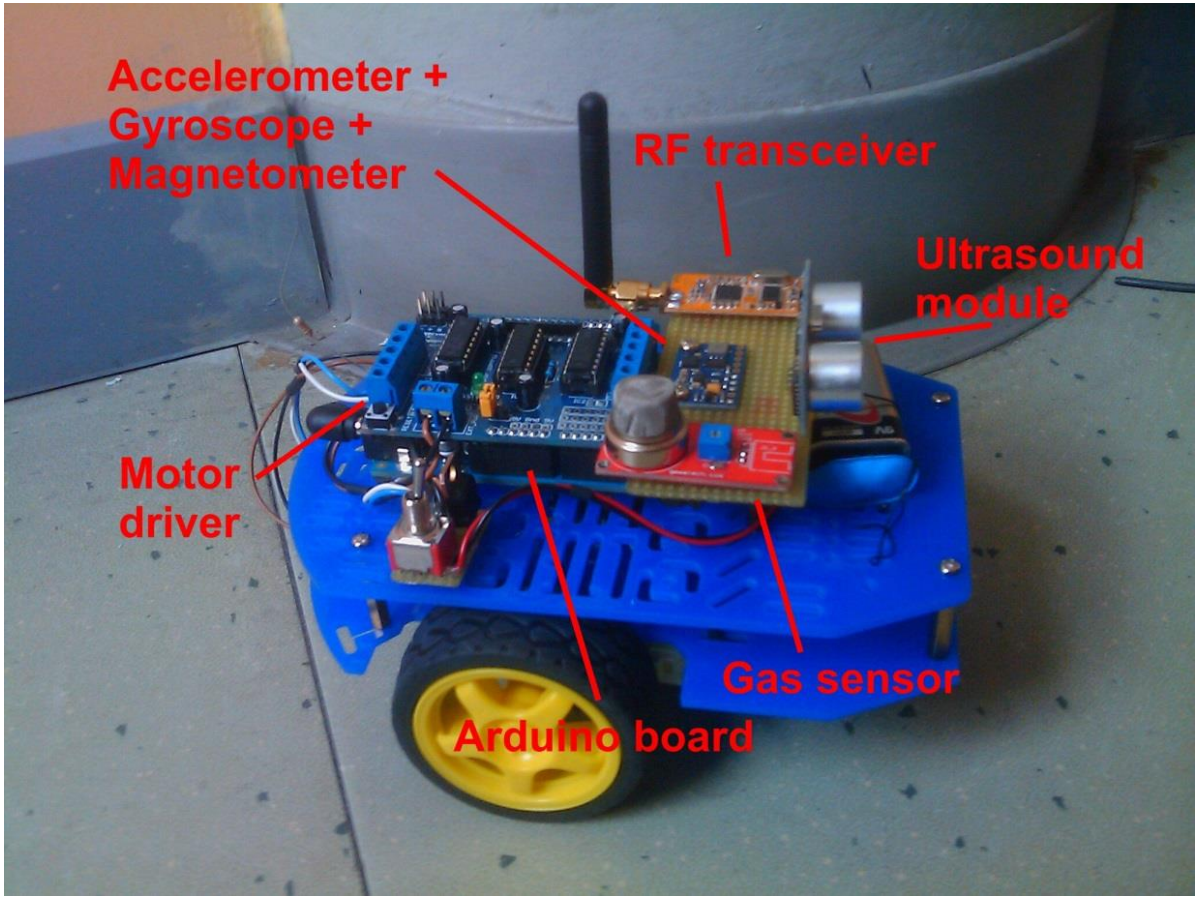
The software

- These are the graphs and the 3D reconstruction.





The final prototype



Prototype costs

Name	Unit cost	Quantity	Total
Chassis	34€	1	34€
Arduino mega 2560	38€	1	38€
Motor shield	13€	1	13€
APC220	34€	1	34€
9DOF sensor	26€	1	26€
Gas sensor	10€	1	10€
Ultrasonic module	6€	1	6€

Total: 161 €