

Robot for tracking rectilinear motion

Stelică Timofte, Lenuța Cîndea*, Cornel Hațiegan

Abstract. *The aim of the research work is to deepen and acquire new practical and technical knowledge about how to work with the Arduino platform, the Android operating system and related technologies. Uniformly varied rectilinear motion monitoring can be done optimally using a robot via Bluetooth. The operation of the system can be easily modified and this mini robot can be used for a variety of purposes. Making adjustments in the design and body of the system will allow this vehicle to serve as a remote control robot to perform a wide range of operations.*

Keywords: *robot, straight motion, Arduino, monitoring.*

1. Introduction

The aim of this work was to develop a Line Follower device that involves the tracking of a black line on a white background (or vice versa) by means of light sensors, and its applicability makes it easier to study uniform rectilinear motion in laboratory conditions for the practical study of students in the Mechanical Engineering Specialization [1], [6], [7].

The use of a mobile phone during the application is an advantage to be able to control the robot, which would replace a conventional remote control by which it would normally have to be controlled. Another big advantage is the Android operating system which has helped to make the control interface a reality, as the Android operating system covers a wide range of devices and applications and is very widespread and new utilities are always emerging through its operation.

Arduino is an open source platform used for building electronic projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software or IDE (Integrated Development Environment) that runs on the initiator computer, used to write and upload source code to the physical board [1].



Arduino has been refined at the Ivrea Interaction Design Institute as an easy rapid prototyping tool for students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board began to change to adapt to new needs and challenges, differentiating its offerings from simple 8-bit boards to products for IoT, 3D printing and embedded environment applications. All Arduino boards are completely open, giving users the ability to build them independently and eventually tailor them to their specific needs. The software, too, is open-source, and is growing through user contributions from around the world.

The Arduino software is easy to use for beginners, but flexible enough for advanced users. It runs on Mac, Windows and Linux. Teachers and students use it to build inexpensive scientific instruments, prove principles of chemistry and physics, or get started with programming and robotics [3].

2. Choice of components. Hardware architecture

An early Arduino board with an RS-232 serial interface (top left) and an Atmel ATmega8 microcontroller (black, bottom right); the 14 digital I/O pins are at the top, the 6 analog input pins at the bottom right and the power connector at the bottom left. Most Arduino boards consist of an 8-bit ATM AVR microcontroller (ATmega8, ATmega168, ATmega328, ATmega1280, ATmega2560) with varying amounts of flash memory, pins and features. The 32-bit Arduino Due, based on the Atmel SAM3X8E, was introduced in 2012. The boards use single or double line pins or female headers that facilitate connections for programming and embedding in other circuits. They can connect with add-on modules called shields [2].

Multiple and possibly stacked shields can be individually addressed via an I2C serial bus. Most boards include a 5 V linear regulator and a 16 MHz crystal oscillator or ceramic resonator. Some models, such as the LilyPad, operate at 8 MHz and are taken off the on-board voltage regulator due to specific factory restrictions.

Arduino microcontrollers are pre-programmed with a bootloader that simplifies loading programs onto the chip's flash memory. The default bootloader of the Arduino UNO is the optiboot bootloader. The boards are loaded with program code via a serial connection to another computer. Some Arduino serial boards contain a level shift circuit to convert between RS-232 logic levels and transistor-transistor logic (TTL) level signals [4].

Current Arduino boards are programmed via the Universal Serial Bus (USB), implemented using USB-to-serial chips such as the FTDI FT232. Some boards, such as the latest generation Uno boards, replace the FTDI chip with a separate AVR chip containing USB-to-serial firmware, which is reprogrammable via its own ICSP header. Other variants, such as the unofficial Arduino Mini and Boarduino, use a USB adapter board or detachable cable, Bluetooth or other methods.

When used with traditional microcontroller tools, standard In-System Programming (ISP) is used instead of Arduino IDE programming [5].

2.1. Arduino UNO V3

It consists of a small platform (68mm/53mm - the most common variant) built around a signal processor and is capable of taking data from the environment through a series of sensors and performing actions on the environment through lights, motors, servomotors and other types of mechanical devices.

The processor is capable of running code written in a programming language that is very similar to the C++ language, the specifications of the Arduino development board are given in Tab. 1. [6].

Table 1. Development board specifications Arduino UNO V3

| | |
|-----------------------------|---|
| Microcontroller | ATmega328 |
| Working voltage | 5 V |
| Input voltage (recommended) | 7 – 12 V |
| Input voltage (limit) | 6 – 20 V |
| Digital pins | 14 (6 PWM output) |
| Analogue pins | 6 |
| Output intensity | 40 mA |
| Output current on 3.3 V | 50 mA |
| Flash memory | 30 KB (ATmega328) 0.5 KB for bootloader |
| SRAM | 2 KB (ATmega328) |
| EEPROM | 1 KB (ATmega328) |
| Clock speed | 16 MHz |

2.2. Engine drivers L298

It is used to control DC motors using the Arduino board. The Arduino is capable of putting out very little power on its ports, totally insufficient to turn a motor.

To avoid burning out the processor, an amplifier has been used to take power from the power supply (in the case of the battery) and transmit it to the motors via commands received from the Arduino board.

The amplifier is called a motor driver based on the L298 integrator, and is a mid-level driver in terms of power conducted. It can control motors requiring 2A or less.

The driver contains a jumper that allows the Arduino board to be powered directly from the power supply, as shown in Fig.1.

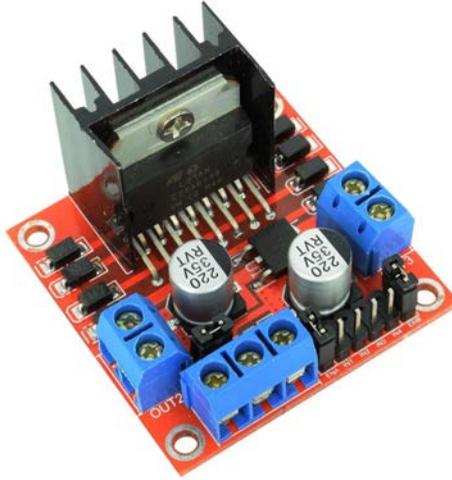


Figure 1. Engine drivers L298 [2]

The HC-05 bluetooth module is a high performance module and consumes very little power. It is also small in size. The product can be used in projects where data needs to be easily transmitted, but not by cable, over reasonable distances, Fig.2.

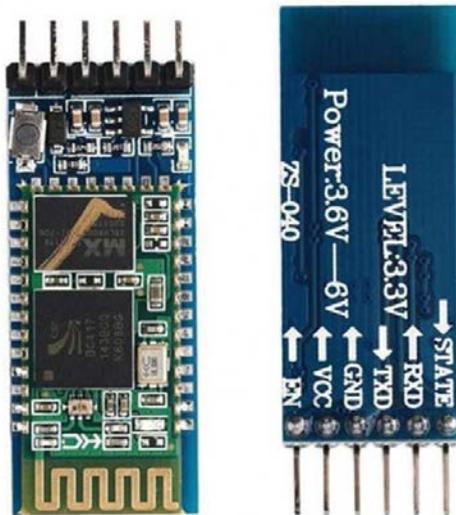


Figure 2. Modulul bluetooth HC-05 [2]

3. Finite element model of the robot chassis

3D simulation of the chassis, Fig. 4, was developed in the SOLIDWORKS 2016 software program, going through the simulation steps, of which the most relevant were selected: The material 2018 Alloy was chosen from the SOLIDWORKS 2016 material library.

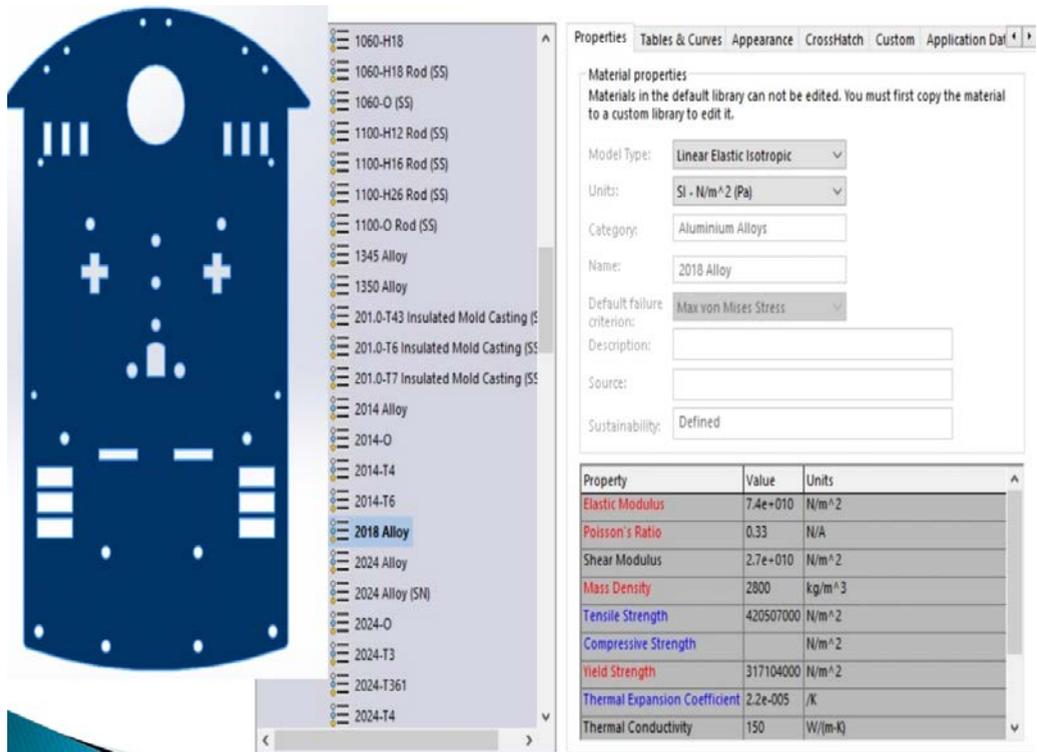


Figure 4. Choice of chassis material

The model was discretized in finite elements by applying Mesh for solid discretization, the chassis was fixed in the motor clamping area and a force of 100 N was applied in the driven wheel clamping area in order to carry out the analysis and simulation study on the mini robot chassis, Fig. 5 [4].

This stage of the simulation is part of the processing stage of the results, and we also set the regime to be static and dynamic by analysing the von Mises stresses.

Before starting the processing stage, the program transferred the boundary conditions and the loads applied on the geometrical elements to the nodes of the discretized model.

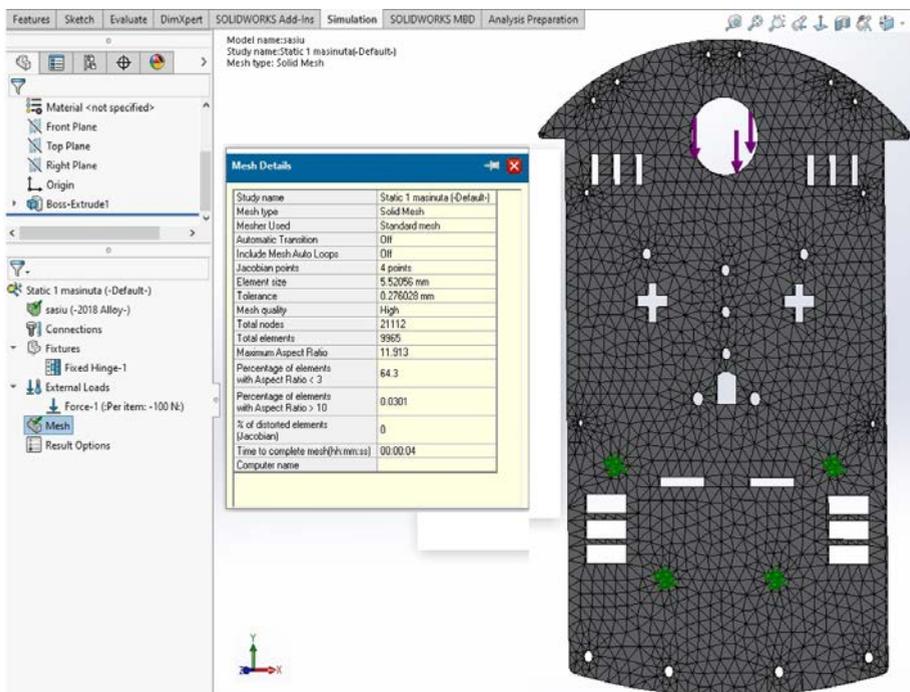


Figure 5. Discretizing the chassis

4. Usefulness of the robot for the study of uniform rectilinear motion

It is known that on the same trajectory, e.g. in a straight line, a mobile can travel at the same speed for several seconds or change speed.

In this case the robot plays the moving mobile. With the help of the apparatus in the Mechanics laboratory, made up of sliders that are placed along a track, the robot moves, under the control of the phone, over certain distances depending on the length of the track. The main condition is the perpendicularity of the light barriers equipped with an LED and the data acquisition device.

For distances travelled in 10 cm increments, the corresponding time is timed to cover the distance.

To save time, one can let the supplied trolley and the built robot pass through the two light barriers at the same distance on both speed settings one after the other.

Knowing the space covered, from 0.1m for each measurement, the time is timed at each step so that the time covered can be calculated for each measurement.

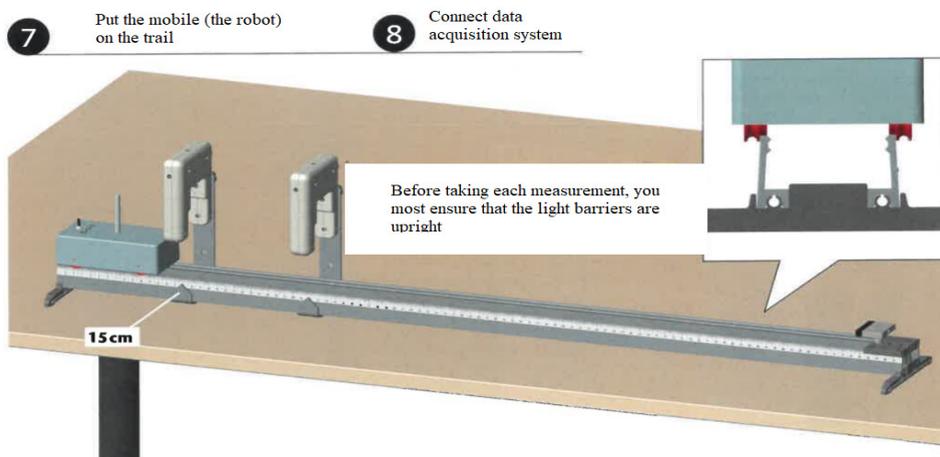


Figure 6. The experimental stand

For example, for the space $s=0.4\text{m}$, $t=3.44\text{s}$ was obtained, resulting in velocity:

$$v_4 = \frac{s}{t} = \frac{0,4}{3,44} = 0,116\text{m/s} \quad (1)$$

In this way the velocities are calculated for each value of the space travelled by both moving mobiles.

Both the robot and the motorised trolley move at constant speed on both speed settings. The actual speed may differ slightly from trolley to trolley, which means that the result of different groups is likely to differ between them.

For both speeds, the distance versus time plot should lead to a straight line through the too high degree of accuracy of the origin, Fig. 7.

A sample measurement of travel time between 25cm and 80cm at both speeds gave the following results:

$$t_1 = \frac{0,80\text{m} - 0,25\text{m}}{0,116\text{m/s}} \cong 4,74\text{s} \quad (2)$$

$$t_1 = \frac{0,80\text{m} - 0,25\text{m}}{0,153\text{m/s}} \cong 3,59\text{s} \quad (3)$$

The deviation between the two time measurements of 4.76 and 3.61 is about 0.5%.

Table 2. Measurement results

| The mobile | Δs [m] | 0,1 | 0,2 | 0,3 | 0,4 | 0,5 | 0,6 | 0,7 |
|----------------|----------------|------|------|------|------|------|------|------|
| Velocity v_1 | Δt [s] | 0,88 | 1,73 | 2,58 | 3,44 | 4,31 | 5,19 | 6,04 |
| Velocity v_2 | Δt [s] | 0,67 | 1,32 | 1,97 | 2,63 | 3,27 | 3,93 | 4,59 |

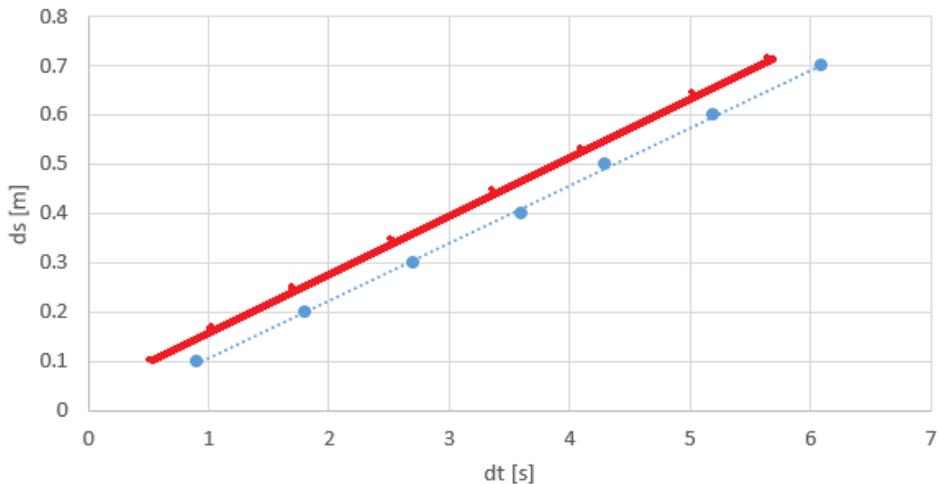


Figure 7. Distance versus time graph

5. Conclusion

The construction and testing of the robot involved three different phases: perception, processing and action.

Depending on the trajectory chosen to study a type of movement, it can be used in several applications, including monitoring different types of movements.

Studying rectilinear motion with this robot can improve students' ability to explain the phenomenon in practice. The construction of the robot is inexpensive and relatively fast and allows experiments to be carried out in the laboratory, comparing the measurement results with the analytical results given in the literature. The geometry of the trajectory is dependent on the time parameter in this type of application, and the errors resulting from comparisons are relatively small. The benefit of this type of robot is that it can be adapted to independently drivable vehicles. In this case, the sensors enable independent travel. Programming allows the robot to adapt to new requirements or to change certain parameters in a short time.

References

- [1] Stroia M.D., Derbac D., Hațiegan C., Cîndea L., Thermostat model with Arduino uno board for controlling a colling system, *Annals of the „Constantin Brancusi” University of Targu Jiu, Engineering Series*, 3, 2018, pp.73-77.
- [2] Doroftei I., Marta C., Hamat C., Suciuc L., Prisacaru Gh., A hybrid wheel-leg mobile robot, *Annals of DAAAM & Proceedings*, 2008, pp. 213.
- [3] Timofte S., Cîndea L., Hațiegan C., Control of a vehicle through a smartphone application, *Annals of the „Constantin Brancusi” University of Targu Jiu, Engineering Series*, 2, 2021, pp. 39-43.
- [4] Ward J.R.; Phillips M.J., Digitizer Technology: Performance Characteristics and the Effects on the User Interface, *IEEE Computer Graphics and Applications*. Spectrum.ieee.org. 2014.
- [5] Alfred A. (n.d.), How to Interface XBEE with Arduino., <http://www.engineersgarage.com/embedded/arduino/how-to-interface-xbee-witharduino-tutorial>.
- [6] Stroia M.-D., Hatiegan C., Popescu C., Virtual instrument designed for data acquisition, *Studia Universitatis Babeş-Bolyai Engineering*, 65(1), 2020. pp.179-186.
- [7] The Android Debug Bridge(ADB), Android Open Accessory Development Kit", <http://developer.android.com/guide/topics/usb/adk.html>, 2012.

Addresses:

- Ph.D. Stud. Eng. Stelică Timofte, Babeş-Bolyai University, Faculty of Engineering, Piața Traian Vuia, nr. 1-4, 320085, Reșița, Romania
stelică.timofte@ubbcluj.ro
- S.L. Dr. Eng. Lenuța Cîndea, Babeş-Bolyai University, Faculty of Engineering, Piața Traian Vuia, nr. 1-4, 320085, Reșița, Romania
(*corresponding author).
lenuta.cindea@ubbcluj.ro
- S.L. Dr. Eng. Cornel Hațiegan, Babeş-Bolyai University, Faculty of Engineering, Piața Traian Vuia, nr. 1-4, 320085, Reșița, Romania
cornel.hatiegan@ubbcluj.ro