JOURNAL OF OPTOELECTRONICS AND ADVANCED MATERIALS Vol. 25, No. 11-12, November - December 2023, p. 563 - 571

Sampling the travel distance of a vehicle through an unconventional method for data acquisition

M. I. RUSU[1,*], V. SAVU[1,*], D. SAVASTRU[1], C. H. GANDESCU[2], A. STAN[2], D. M. COTOROBAI[2]
[1]National Institute of R&D for Optoelectronics INOE 2000, 409 Atomistilor, Magurele, PO Box MG-5, 77125, Ilfov, Romania
[2] S.C. Lightning-Net SRL, 24C Blandesti, District 4, Bucharest, PO Box 042077, Bucharest, Romania

The improvement of transport on the road arteries presupposes, first of all, the best possible quality of the asphalt carpet. In this sense, they developed standards regarding the quality of practical asphalt carpets on road arteries and created a series of systems and devices placed on vehicles to determine the asphalt carpets' compliance with these standards. These systems are made, more recently, with lasers that illuminate the asphalt carpet, in the form of lines perpendicular to the direction of movement of the vehicles intended to control the quality of the asphalt carpets, and with the help of cameras, the images reflected by the asphalt carpet are applied, following a further processing of the images. Selected A problem, very important in the development of this process, is the triggering of the time-lapse cameras, determined by the speed of movement of the control vehicles, which means a very precise sampling of the distances of the road arteries. In this, we do not propose to implement a dedicated software in a microcontroller, theoretically study and practically determine a non-conventional search camera triggering system at the very well-established time interval with an error of less than 10^{-4}.

(Received November 10, 2023; accepted November 24, 2023)

Keywords: Sampling, Accelerometer, Speed, Automatic Trigger, Camera, Road bumps

1. Introduction

The systems for determining the quality of road arteries are equipped with devices, which are triggered at very well determining time intervals, in order to acquire information related to their technical characterization. Arterial roads can be single-lane or multi-lane. All these categories of road arteries are of particular importance from the point of view of their quality. The quality of road arteries is very important for the flow of traffic, people's health, freight transport, military transport in the case of strategic roads, etc. [1]. Also, the technical quality of road arteries is also important for reducing fuel or electricity consumption (electric vehicles) [2]. The determination of the quality of road arteries is carried out by sampling them, with the help of systems developed for this purpose, at distances in accordance with the standards in force, followed by data processing in the laboratory, in order to put these results into practice and correct these road arteries to obtain their quality in accordance with current standards.

These systems are commanded, for the road samples in order to acquire the information taken from the road arteries, by different triggering devices: odometers [3], [4] GPS, accelerometers, etc.

Odometers can introduce errors into the sampling systems, due to accumulation of unwanted particles on the odometer system [5], increasing the diameter of the odometer reading system, which represents errors in the triggering system. In the case of GPS driven trigger systems, errors may occur at times of loss of GPS signal (under bridges, tunnels, etc.) and will result in errors in these situations.

Modern systems use accelerometers on all axes and calculating the area under the speed graph gives the distance travelled [6]. Accelerometers are used to determine the accelerations, speeds and distances travelled by motor vehicles, and knowing the acceleration as the second derivative of space with respect to time, it follows that if we use an integrator, as an electronic circuit, we will obtain the speed of movement of the motor vehicle and continuing with a second integrator, after integration, we will obtain the space travelled by the vehicle [7].

The integrative systems, made in hardware, are based on operational amplifiers that have capacitive elements on the negative feedback loop, which after a period of time can change their capacity value, for certain reasons (temperature, mechanical oscillations and vibrations, aging, etc.) and they will introduce errors into the integrative system, resulting in errors into the triggering system.

The system that we propose in this paper is an accelerometric system that extracts the information of the distance travelled by the vehicle [8] intended to determine the quality of the road arteries, based on a software implemented in the data processing board (ARDUINO UNO) [9] coming from the accelerometer of type MPU-6050 [10].

The data from the accelerometer are transferred serially on an I^2C Bus to the ARDUINO UNO type software processing board using the ATMEGA328P microcontroller [11, 12] and from here on to the acquisition camera. The implemented software eliminates the offset errors of the accelerometer by calibrating the software, calculates the speed of the vehicle, determines the space by applying the physical formulas of movement and transmits the trigger moments to the optical system.

By using the software implemented in the proposed system, errors due to vibrations and mechanical oscillations are eliminated (the MPU-6050 accelerometer includes gyroscopes on the x, y, z axes), errors due to temperature (the MPU-6050 accelerometer also determines the temperature), etc.

The software program is stored in a non-volatile memory of the microcontroller and can be accessed whenever needed, thus eliminating system aging errors.

In the experimental part, the speed determined with the help of the oscilloscope was approximated to the speed determined by the software with an error close to 10^{-4}.

The experimental part is described in section 2, and the results obtained from both oscilloscope and software determinations are detailed and discussed in section 3. Section 4 summarizes the conclusions of this study.

2. Experimental

To determine the quality of road arteries, systems are used to acquire them, by sampling them at distances according to the standards in force. For the acquisition of the data [13] corresponding to the determined samples, visible red lasers [14] are used that illuminate the asphalt carpet and with the help of some cameras this information is acquired [15]. Image acquisition cameras [16] require triggering at times corresponding to these samples. The triggering of these cameras, most precisely and to eliminate errors [17], is done with accelerometers. The accelerometer I used in this work is MPU-6050 type and determines the acceleration on the 3 spatial axes, the rotation around these axes and the temperature of the environment where it works.

This accelerometer has the following technical data:

- Supply voltage: 3.3V - 5V (LDO regulator included);
- I^2C bus voltage: 3.3V (MAX);
- Current: 5mA;
- Accelerometer programmable range: ±2g, ±4g, ±8g, ±16g;
- Maximum I2C frequency: 400kHz
- Three accelerometric axes: x, y, z.
- Three gyroscopic axes of rotation: x, y, z.
- It determines the temperature of the environment where it acts.

Communication between the MPU-6050 type accelerometer module and the ARDUINO module with the ATMEGA328P microcontroller is carried out serially through the I^2C bus. This bas realizes bidirectional communication between MPU-6050 accelerometer board and ARDUINO UNO board with ATMEGA328P microcontroller. This type of communication is carried out only at the request of one of the parties, making it possible for the microcontroller to function and for other operations that are necessary in the software. In this situation the routine for calculating the speed, distance, and sending the camera trigger signal does not delay the transmission of data on this bas.

The whole system occupies very small dimensions, being composed of two boards (Fig. 3a), having the dimensions of the MPU-6050 boards of 20mm x 15mm and respectively the ARDUINO board with dimensions of 58mm x 76mm.

The MPU-6050 accelerometer board is shown in Fig. 1, where the positive x and y axes of accelerometer displacement can be observed and also the positive gyroscopic directions for the x and y axes.

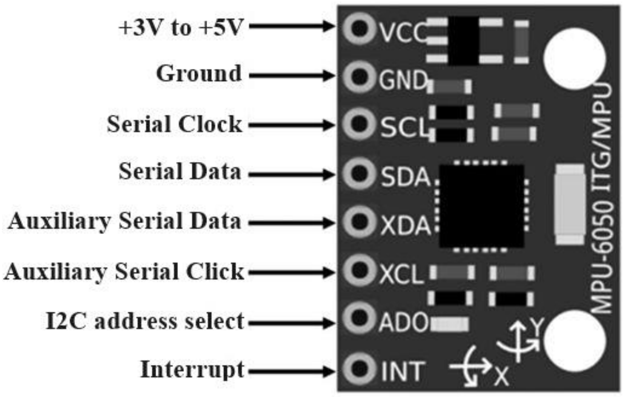


Fig. 1. Accelerometer module type MPU-6050. [18] (color online)

Fig. 2 shows the axes of positive action of the acceleration and the rotation around the axes in the positive direction. The axis of interest for our application is +X, the others are neglected.

It should be noted that the other axes are neglected only for the calculation of the traveled distance (sampling distance of 0.25m) and otherwise they are used to eliminate mechanical oscillations and vibrations and the determination of temperature is used to eliminate the influence of the environmental temperature on the determination of accelerations.

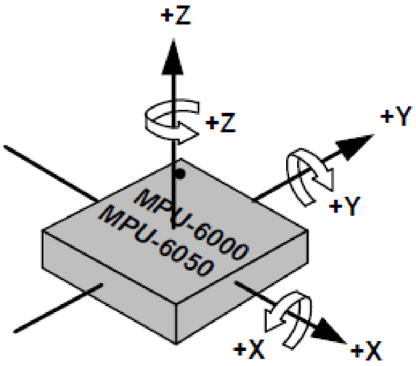


Fig. 2. Presentation of axes of accelerations and gyroscopes.[19]

The module contains an MPU-6050 integrated circuit with accelerometer, gyroscope and temperature sensor. It communicates over the I^{2}C interface, needing only 2 connections. The MPU6050 module uses the I^{2}C protocol to communicate with the ARDUINO board.

The board in Fig. 3b is programmed on the computer through the USB port with a dedicated software applicable to the MPU-6050 accelerometer. After writing the software in the non-volatile internal memory of the ATMEGA328P microcontroller, the system can be applied in the field keeping the data written by programming.

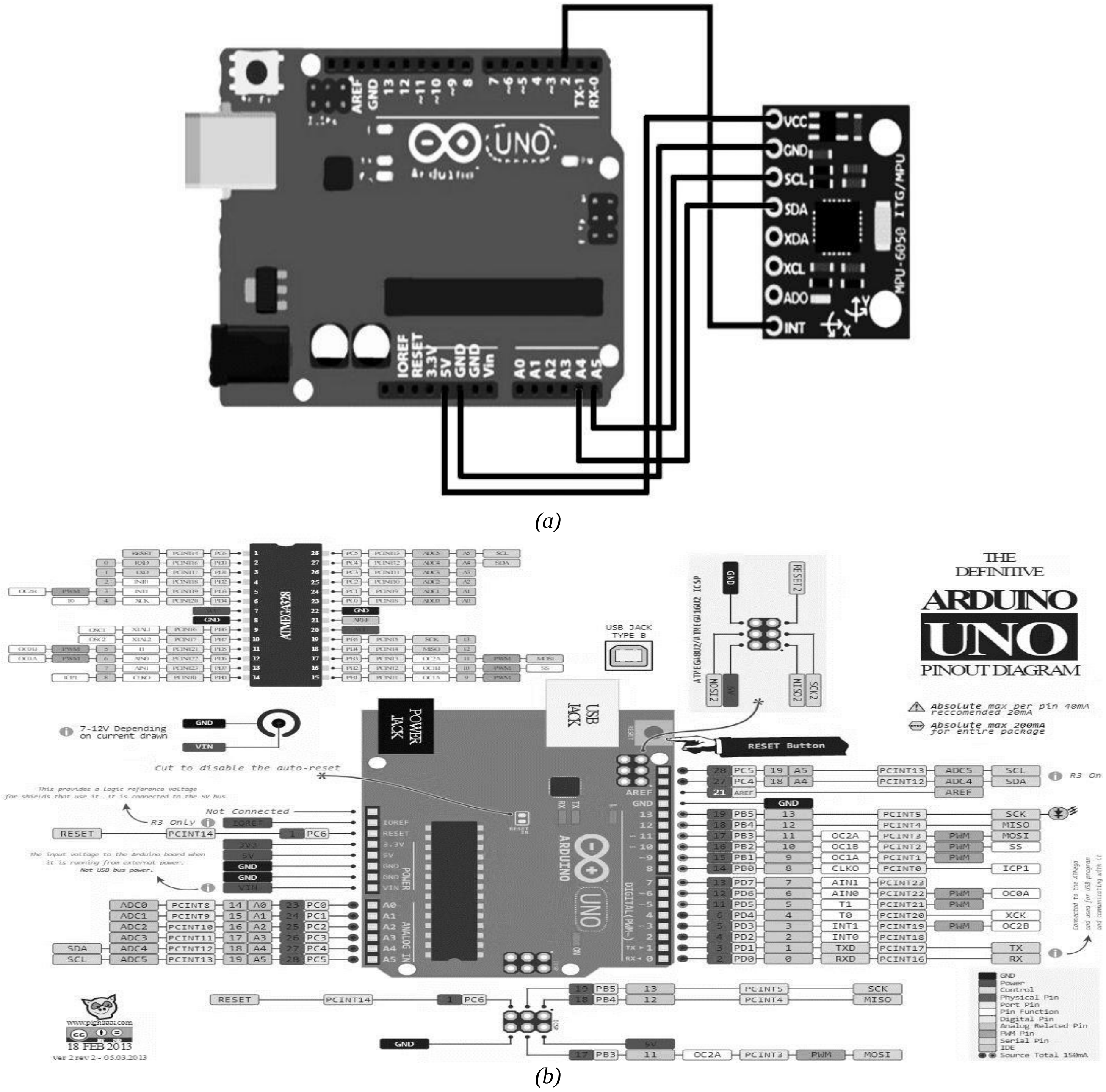
Fig. 3a shows the overall electronic diagram of the triggering transmission system of the cameras for acquiring images of road artery samples.

The MPU 6050 accelerometer module is powered from the ARDUINO UNO board, which means that a separate power supply for the accelerometer module is not required.

The communication between the accelerometer module and the ARDUINO UNO board is done through the I^{2}C bus, where the serial data and address bus (SDA) connects to address A4 and the serial clock bus (SCL) connects to address A5.

The ATMEGA328P microcontroller has these addresses programmed (by software) for I^{2}C serial communication (SCL, SDA).

The communication between the PC and the ARDUINO UNO board is carried out through the serial universal bass port (USB type B), through which the ATMEGA328P microcontroller is programmed (transferred) and also the output data from the ARDUINO UNO board is transferred to the PC.



(a)

(b)

Fig. 3. Connecting the MPU6050 module to the Arduino board (a), Arduino board connectors (b). [20,21] (color online)

The software implemented for sampling the travel distance in order to trigger the camera is briefly presented below:

```
    #include <Adafruit_MPU6050.h>
    #include <Adafruit_Sensor.h>
    #include <Wire.h>
    #include <Adafruit_BusIO_Register.h>
    Adafruit_MPU6050 mpu;
    void setup(void) {
     Serial.begin(115200);
     if (!mpu.begin()) {
      Serial.println("Failed to find MPU6050 chip");
      while (1) {
       delay(10);
      }
     }
     Serial.println("MPU6050 Found!");

mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);
     delay(100);
    }

     float v=0;
     float d=0;
     float a0=0;
     float a1=0;
     float t=0.01167;
     int TTL1 = 2;
     int con1 = 25;
    void loop() {
     /* Get new sensor events with the readings */
     sensors_event_t a, g, temp;
     mpu.getEvent(&a, &g, &temp);
     Serial.print("Acceleration X: ");
     Serial.print(a.acceleration.x);
     if (a1==0){
     a1 = a.acceleration.x;
     }else{
     a0 = a.acceleration.x - a1;
      v = v + (a0*t);
      if (v<0)
      {
      v=o;
      }
      d = d + (v*t);
      if (d>=25) {
      digitalWrite(TTL1,HIGH);
      delay(0.25); // the time to keep TTL on High in
milliseconds
      digitalWrite(TTL1,LOW);
      d = 0;
       }
     }
      Serial.println(", Distance d: ");
      Serial.println(d);
      Serial.println(", V: ");
      Serial.println(v);

      Serial.println(", a0: ");
      Serial.println(a0);
      Serial.println("a1: ");
      Serial.println(a1);

      delay(10);
    }
```

This software is implemented in the non-volatile memory of the ATMEGA328P microcontroller of the ARDUINO UNO board.

With this software, the speed of the vehicle intended to determine the quality of the road arteries is determined. At the beginning, the libraries necessary to run the program are loaded into the non-volatile memory of the microcontroller, then the accelerometer is initialized, following the description of the software variables, and finally the software subroutines are put into operation, displaying the trigger distance at the time of the trigger, and the speed determined by the software will also be displayed.

3. Results and discussion

The calibration of the sampling system of the distance travelled by a vehicle through an unconventional method for data acquisition, we achieved it by determining the time between two consecutive triggering intervals of a camera, which transmits images at the moments of triggering to an acquisition unit, following a further processing of the data, in order to determine the inclusion of the quality of roads in the standards in force.

This time interval was determined with the help of an oscilloscope that is connected to the PD2 port of the ARDUINO board. The assembly was connected to a PC. Knowing the speed shown on the PC display and the time determined by the oscilloscope, the corrections were made so that the sampling distance we proposed of 25cm was determined as correctly as possible.

This operation was done in several consecutive steps similar to a recurrence relation and continued until the error was of the order of 10^{-4}.

In the program we started from a time t = 0.01167s and by successive introductions of the time determined by the oscilloscope in the program we obtained a difference between the speed displayed on the PC display and the one calculated with the time determined by the oscilloscope of 10^{-4}.

The procedure for determining the recurrence was carried out in several steps, briefly presented below.

Initial data:

d_p = 0,25m - represents the distance proposed and introduced in the past.

t_{p0} = 0.01167s – represents the initial time entered in the software

Step 0 (initial)

$$V_{p_0} = a_0 \times t_{p0} \tag{1}$$

where, V_{p_0} – represents the initial speed determined with the help of the software;

a_0 – represents the initial acceleration determined by the MPU 6050 accelerometer board;

Next, we determined the time measured on the TEKTRONIX DTS 3032 oscilloscope;

t_{o_0} – represents the initial time determined on the oscilloscope.

$$t_{o_0} \times V_{p_0} = d_{c_0} \tag{2}$$

where, d_{c_0} is the initially calculated distance.

$$d_{c_0} - d_p = \Delta d_0 \tag{3}$$

If Δd_0 is negative, the time is reduced, or if it is positive, the time is increased to the value

$$t_{p_1} = {d_{c_0}}\Big/{V_{p_0}} \tag{4}$$

$$V_{p_1} = a_0 \times t_{p1} \tag{5}$$

Calculate $V_{p_0} - V_{p_1} = \Delta V_{p_0}$, which must have a value lower than 10^{-4}.

Step 1

Calling relation 5, we determine the time measured on the oscilloscope in step 1, t_{o_1}

t_{o_1} – represents the time determined on the oscilloscope.

$$t_{o_1} \times V_{p_1} = d_{c_1} \tag{6}$$

where d_{c_1} is the distance calculated in step 1.

$$d_{c_1} - d_p = \Delta d_1 \tag{7}$$

If Δd_0 is negative, the time is reduced, or if it is positive, the time is increased to the value

$$t_{p_2} = {d_{c_1}}\Big/{V_{p_1}} \tag{8}$$

$$V_{p_2} = a_0 \times t_{p2} \tag{9}$$

Calculate $V_{p_1} - V_{p_2} = \Delta V_{p_1}$, which must have a value lower than 10^{-4}.

This algorithm is continued until a value smaller than 10^{-4} or very close to it is obtained for ΔV_{p_n}.

In the software we introduced a routine to calibrate the MPU 6050 accelerometer and Fig. 4 shows the shape of the signal after calibrating the MPU 6050 accelerometer, when the entire trigger system is in Standby.

This fact is marked in white on a black background at the 101.526ms software time instant of the MPU6050 accelerometer with zero (0) acceleration value and is shown in Fig. 4.

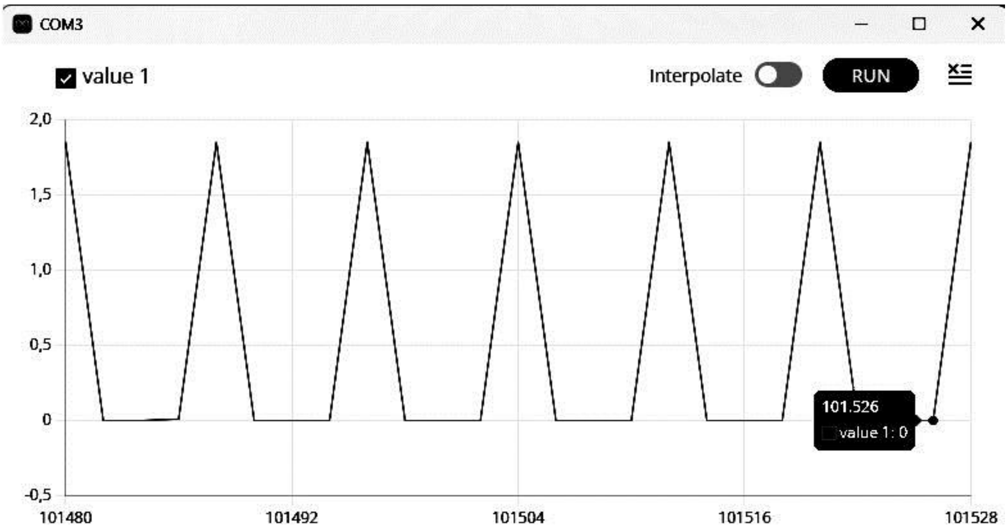


Fig. 4. Standby signal generated of accelerator (color online)

The MPU6050 accelerometer calibration is also reported in the software implemented in the acceleration processing board (ARDUINO UNO with ATMEGA328P microcontroller) and can be seen in the software response sequence at time 11:56:01.734 on a grey background.

Data sequence after calibration with the system in standby.

```
11:56:01.619 -> MPU6050 Found!
11:56:01.734 -> Acceleration X: 2.29, Distance d:
11:56:01.734 -> 0.00
11:56:01.734 -> , V:
11:56:01.734 -> 0.00
11:56:01.734 -> , a0:
11:56:01.734 -> 0.00
11:56:01.734 -> a1:
11:56:01.734 -> 2.29
```

After the MPU6050 accelerometer calibration process, we moved the accelerometer to obtain an acceleration and Fig. 5 shows the shape of the signal generated by the moving accelerometer. The acceleration is marked in white on a black background and at time 108.316ms it has a value of 1.79m/s^2.

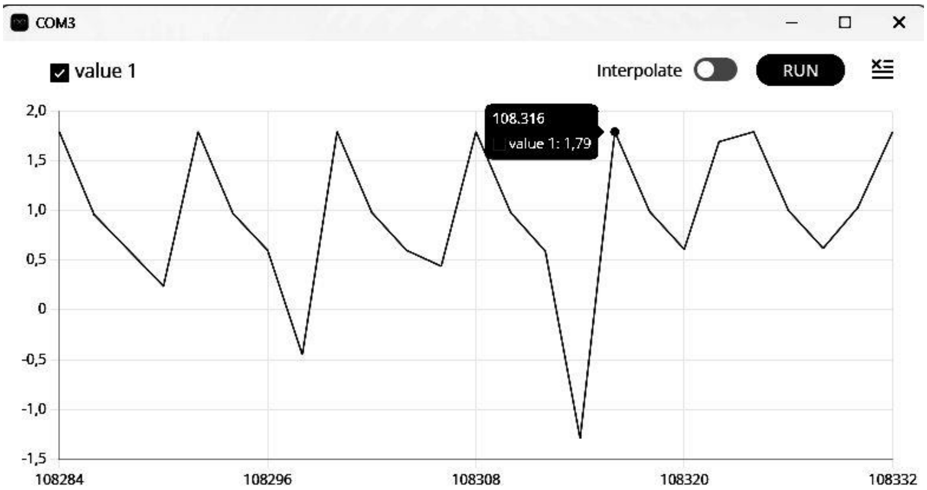


Fig. 5. The signal generated in motion

In the software sequence presented below, a braking process is signalled where the acceleration at time 12:01:01.668 has the value -16.18m/s^2 and is marked on a grey background.

Flow of data in motion.

```
12:01:01.668 -> Acceleration X: -15.41, Distance d:
12:01:01.668 -> 1.09
12:01:01.668 -> , V:
12:01:01.668 -> 1.46
12:01:01.668 -> , a0:
12:01:01.668 -> -16.18
12:01:01.668 -> a1:
12:01:01.668 -> 0.77
```

Following the process of correcting the speed calculated by the software with the one determined with the oscilloscope, we obtained the deviation values entered in the software normalized to the value of 10^{-4} and the following table shows these values.

Table 1. Normalized deviation values for speed correction (soft and practical)

Step	p_0	p_1	p_2	p_3	p_4	p_5	p_6
Normalized deviation	0	1.45	0.68	1.24	0.85	1.09	0.99

Making an analysis between the data measured with the oscilloscope and those generated by the software, I concluded that the approximation process in the steps described above, manifests itself as an oscillating system and can be appreciated with equation (10) which I presented graphically in the Fig. 6.

Here it is observed that at step p6 the approximation of the speed in the software came very close to the speed determined with the help of the oscilloscope.

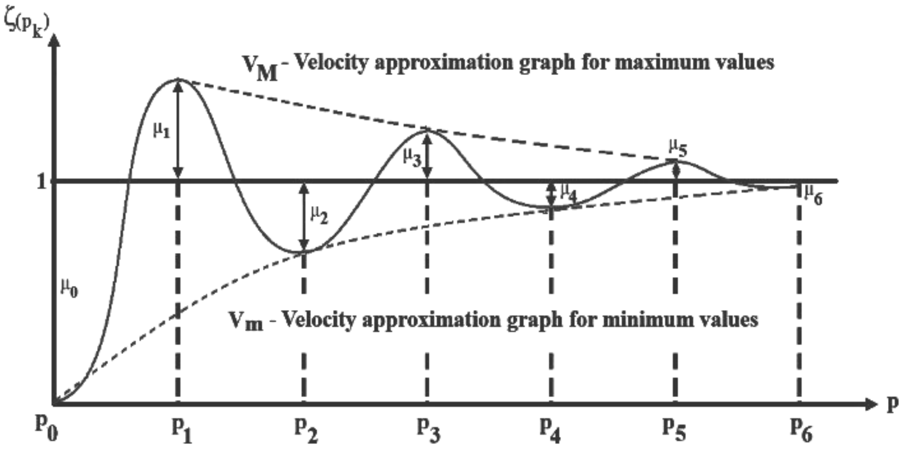


Fig. 6. Approximation error correction plot of speed determined by software compared to speed determined practically with the oscilloscope (color online)

In the graph in Fig. 6, the method of approximating the speed determined by the software compared to the one determined practically (with the oscilloscope) is presented. Here, at the initial step (p_0) the acceleration calibration was done, the initial velocity is 0m/s and after determining the value of the first acceleration, the recurrent process of velocity approximation in the software was started.

The process was continued until the software-determined speed approached the oscilloscope-determined speed to the fourth decimal place (10^{-4}). The recursion steps are shown on the abscissa and the software speed approximation is shown on the ordinate. The blue color graph represents the software speed approximation. The dotted and brown graph represents the variation of the approximate maximum software speed values. The dotted and green graph represents the variation of the minimum approximate software speed values.

$$\zeta_{(p_k)} = 1 - (-1)^k \cdot e^{-k\pi \cdot ctg(\mu)} \qquad (10)$$

where: ζ_{p_k} represents the speed approximation at step k;

$k \in N$ and represents the step order;

p represents the step in the procedure for determining the recurrence;

μ represents the value of the deviation entered in the software from the normalized value.

The ideal normed approximation is $\zeta_n = 1$ (ζ_n the normed approximation) and has the expression:

$$\zeta_n = \frac{\zeta_{(p_k)}}{\zeta_d} \qquad (11)$$

where ζ_d represents the desired approximation (10^{-4}).

The time measured on the oscilloscope is given by the difference between the steps:

$$p_{k+1} - p_k = t_k \qquad (12)$$

where p_{k+1} represents the next step;

p_k represents the current step;

t_k represents the current time.

The first step after calibration (p_0) will have the initial time ($t_{o_0} = 0$), the following times will be from the step difference determined on the oscilloscope at steps: p_1, p_2, p_3, p_4, etc. resulting in times: t_{p1}, t_{p2}, t_{p3}, t_{p4}, etc. in the form: $p_1 - p_0 = t_{p1}$;

$p_2 - p_1 = t_{p2}$; and so on. On the oscilloscope we determine the time at step k and at step $k+1$ and by the difference of the times of the two consecutive steps we determine the time at which a distance of 25cm was travelled, thus we determine the speed and enter the new speed in the software, resulting in a new time constant value. It continues until the difference between the speeds measured on the oscilloscope and the one determined by the software has a difference to 4 decimal places. As can be seen from the graph in Fig. 6, the approximation of the speed calculated by the software compared to the one determined practically with the help of the oscilloscope, came close to the error we proposed, of 10^{-4}.

The oscilloscope was connected to port 2 of the ARDUINO UNO board, then the recursion steps were followed and the speed was corrected to the approximation of 10^{-4}.

To trigger the acquisition cameras, the ARDUINO UNO board generated every 0.25m the signal shown in Fig. 7, with the maximum amplitude of 4V and the average duration of 120µs.

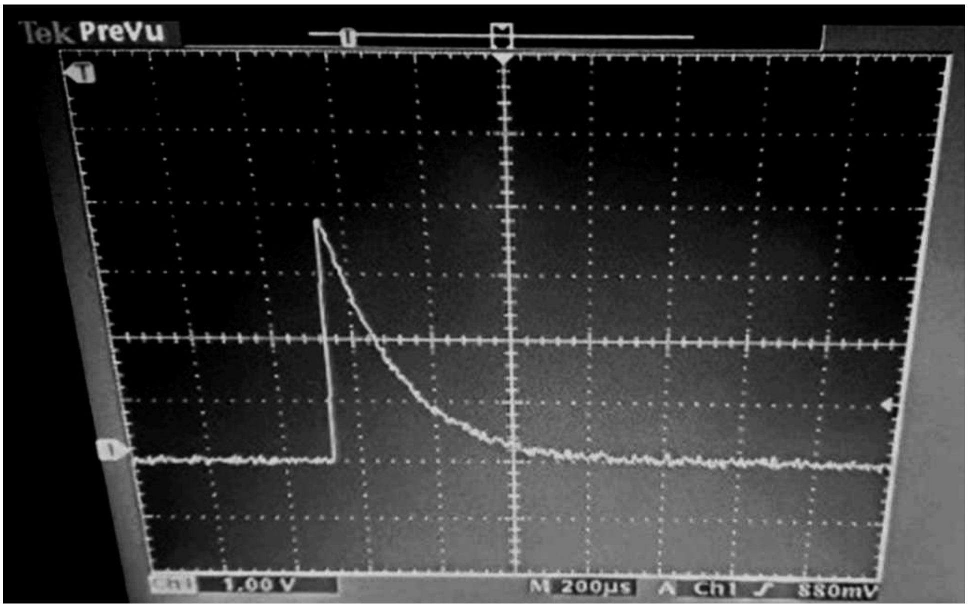


Fig. 7. Generation of acquisition camera trigger pulse (color online)

Fig. 8 shows the method of determining the practical time measured with the oscilloscope and entered into the software in order to correct the software speed. As can be seen on the oscilloscope, the time between two successive triggers is 15.6ms. Making an approximate calculation we deduce a software speed of 16.0256m/s or 57.6923077Km/h (the speed value is given to the level of hundreds of µm).

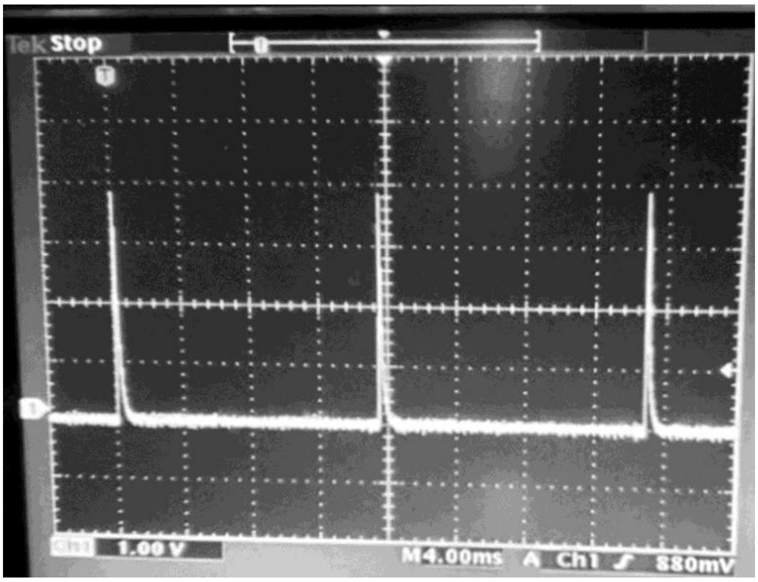


Fig. 8. Generating the trigger signal in a sequence of three pulses to determine the vehicle's travel time for software speed correction (color online)

4. Conclusions

Determining the quality of road arteries as accurately as possible is dependent on the accuracy with which they are sampled. The sampling of road arteries is dependent on the correctness of the transmission of the trigger pulses, at the time points, which are dependent on the standardized distance between two consecutive samples. The most accurate triggering, in terms of the distance between two consecutive samples, of the cameras, is transmitted from the ARDUINO UNO board.

These trigger pulses are dependent on the speed of the road test vehicle. The acquisition cameras were triggered by the ARDUINO UNO board using the software presented in the article.

The correction of the speed calculated with the help of the software compared to the real one, was achieved by designing a recursive calculation system and using the oscilloscope.

Analysing the error correction graph of the speed approximation determined by the software compared to the speed determined practically, with the oscilloscope, it was found that at the sixth step the speed approximation is 10-4 and also the speed approximation is close to a damped oscillating system.

The speed correction was made with an approximation of the real speed of the vehicle, lower than 10^{-4}. With this approximation, for a given velocity measured in m/s, an approximation of the sampling distance of ±100µm results

Acknowledgments

This work was supported by the Core Program within the National Research Development and Innovation Plan 2022-2027, carried out with the support of MCID, project no. PN 23 05, the Romanian Ministry of Research, Innovation and Digitalization, through Program 1-Development of the national research-development system, Subprogram 1.2 - Institutional performance - Projects to finance the excellent RDI, Contract no. 18PFE/30.12.2021 and a grant of the Ministry of Research, Innovation and digitalization, CNSC-UEFISCDI, project number PN-III-P4-PCE-2021-0585, within PNCDI III. This re-search was funded by the Romanian Ministry of European Investment and Projects & Romanian Ministry of Research, Innovation and Digitalization, contract no.8/1.2.1 PTI ap.2/17.02.2023.

References

[1] C. Mc. Andrews, K. M. Pollack, D. Berrigan, A. L. Dannenberg, J. Christopher, Am. J. Public Health **107**(8), 1278 (2017).
[2] Y. Xu, K. Wang, Advances in Computer Science Research **80**, 504 (2018).

[3] D. Nemec, M. Hrubos, A. Janota, R. Pirnik, M. Gregor, IEEE Sensors Journal **21**(14), 15687 (2021).

[4] A. A. Youssef , N. Al-Subaie, N. El-Sheimy, M. Elhabiby, Sensors **21**, 1327 (2021).

[5] M. I. Rusu, Y. Addab, C. Martin, C. Pardanaud, V. Savu, I. I. Lancranjan, D. Tenciu, Optoelectron. Adv. Mat. **17**(3-4), 159 (2023).

[6] Michael J. Ruiz, Phys. Educ. **56**, 055035 (2021).

[7] M. N. Tautan, S. Micloș, A. Stoica, D. Savastru, R. Savastru, Patent RO130784A2·2015-12-30.

[8] V. Savu, M. I. Rusu, D. Tenciu, L. Baschir, Romanian Association for Nonconventional Technologies **XVIII**(3), 116 (2014).

[9] https://www.farnell.com/datasheets/3969706.pdf

[10] https://www.allaboutcircuits.com/electronic-components/datasheet/MPU-6050--InvenSense/

[11] https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf

[12] V. Savu, Cherenkov cone detection in saline environment, Editura Tehnopress Iași, 2019.

[13] M. S. Gomezdiaz, D. A. Gutiérrez-Hernandez, F. J. Casillas-Rodriguez, M. Mora-Gonzalez, C. I. Medel-Ruiz, J. Muñoz-Maciel, J. Optoelectron. Adv. M. **25**(7-8), 340 (2023).

[14] M. I. Rusu, Materiale nanostructurate pentru dispozitive optoelectronice: Aplicații, Ed. Tehnopress, Iasi, Romania, 1 (2019).

[15] N. Bhan, S. Kaur, N. Nair, J. Optoelectron. Adv. M. **24**(5-6), 221 (2022).

[16] P. Mihailovic, M. Barjaktarovic, S. Petricevic, Optoelectron. Adv. Mat. **15(**9-10), 433 (2021).

[17] B. Guo, Z. Yan Wang, Optoelectron. Adv. Mat. **15**(7-8), 364 (2021).

[18] https://components101.com/sensors/mpu6050-module

[19] https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf

[20] https://www.optimusdigital.ro/ro/senzori-senzori-inertiali/96-modul-senzor-triaxial-mpu-6050.html

[21] https://www.circuito.io/blog/arduino-uno-pinout/

---

[*] Corresponding author: madalin@inoe.ro;
savuv@inoe.ro