

# CONSTRUCTION AND CONTROL OF AN AUTONOMOUS MOBILE ROBOT FOR URBAN WASTE COLLECTION

Alexandru Stan<sup>1</sup>, Mihai Mărgăritescu<sup>1\*</sup>, Ana Maria Eulampia Rolea<sup>1</sup>, Andrei Cristian Dinu<sup>1</sup>,  
Victor-Marin Zafiu<sup>1</sup>, Diana Maria Cotorobai<sup>1</sup>

<sup>1</sup>The National Institute of Research and Development in Mechatronics and Measurement Technique (INCDMTM),  
6-8 Pantelimon Road, Bucharest, Romania

Email: [alexandrustan71.sa@gmail.com](mailto:alexandrustan71.sa@gmail.com), [mihai.margaritescu@gmail.com](mailto:mihai.margaritescu@gmail.com), [eulampia46@yahoo.com](mailto:eulampia46@yahoo.com),  
[victor.zaf@gmail.com](mailto:victor.zaf@gmail.com), [andrei.dinu931@gmail.com](mailto:andrei.dinu931@gmail.com), [diana.cotorobai22@gmail.com](mailto:diana.cotorobai22@gmail.com)

**Abstract** - The SIRAMAND project propose an autonomous mobile robot for collection of solid waste, specific to the urban environment, such as glass or plastic bottles, aluminium cans placed on alleys of parks, streets, etc. The mobile robot, following a certain programmable trajectory, detects the waste, collects and deposits it in a container. In case of damage, the robot can be remote controlled.

**Keywords:** Autonomous robots, Mobile robots, Waste collection.

## 1. Introduction

Nowadays, the waste pollution within recreational areas has become a major problem for the local authorities. Therefore, this paper aims to present the development, construction and testing of an autonomous ecological mobile robot. Its main objective is to take over the workload of local authorities by collecting waste, such as: small plastic/glass bottles, aluminium cans. Other types of waste collection robotic systems, but of different concept and structure are presented in [1], [2]. In addition to this, the motivation behind this activity is to interface different control units, each with its own particularities and implement intelligent collaborative algorithms for positioning and computer vision. In the following figure is represented the robot in functional state:



Figure 1.1 – Autonomous mobile robot

The main elements of the robots are: (1) The chassis – motor wheels and directional wheels; (2) Autonomous command module – CECC PLC, camera,  $\mu C$  Teensy, F405 autopilot, GPS module, buck/boost converters, multi-role PCB, sensors, cooling fans; (3) EXCM 3D positioning gantry; (4) Custom-made body. In the next figure is represented the robot with a side panel removed in order to see the inside of the robot:

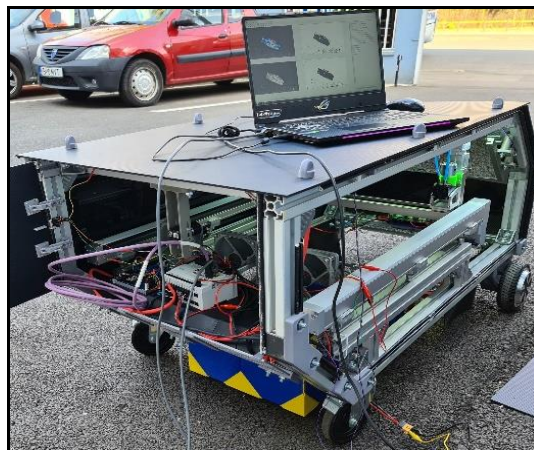


Figure 1.2 – Robot inside view

## 2. Work Concept

The working concept is described by the collaborative nature of the implemented algorithms. Connection work is described in [3-6]. The collaborative nature is given by the need of establishing a bidirectional connection between microcontrollers and a programmable logic controller.

By doing so, the algorithms are able to communicate no matter the logic level. In this particular case, the master controller is a Teensy  $\mu C$  and its slave controllers are a Festo CECC PLC and a F405 Autopilot.

Therefore, within the user must set-up the robot by following the work cycle: (1) After the power-on sequence, the initialization step has as primary objective the auto-calibration of the entire system and the acknowledgement of the bidirectional communication between Teensy  $\mu C$  and CECC PLC, hand-shake operation; (2) A connection between F405 Autopilot and a laptop is made in order to plan the trajectory of the robot by placing waypoints in the dedicated software, iNav; (3) Next, by using the dedicated remote monitoring controller, the user sends a "Go" command to autopilot through Teensy; (4) The robot commences the navigation phase and halts it when waste is detected by means of the onboard ultrasonic distance sensors; (5) The onboard camera snaps a picture of the waste and the centroid coordinate is calculated by recognizing the object and its form; (6) The calculated coordinate is sent to the Festo EXCM positioning gantry, and the gantry moves to the designated coordinate by Point-to-point method; (7) The Z axis grabs the waste by using the equipped vacuum gripper, and the gantry will place the grabbed waste object in the yellow-blue container; (8) The robot will resume to the navigation mode until the next detected waste object. In the next figure is presented the block diagram:

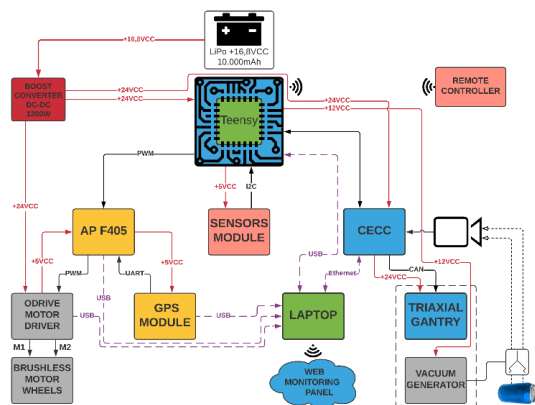


Figure 2.1 - Block diagram

### 3. Collaborative Algorithms

#### A) Teensy $\mu C$ control algorithm:

Teensy  $\mu C$  has been chosen to be the master controller for the presented robot for that fact that it delivers great computing power, being powered by an ARM Cortex-M4 at 120 MHz. Therefore, Teensy  $\mu C$  processes all the data received from the onboard sensors, such as three ultrasonic distance measurement sensors, infrared distance measurement sensor and a current sensing sensor.

After processing the data, specific digital signals must be sent to the PLC. Knowing the fact that Teensy  $\mu C$  has +3.3VDC TTL level and CECC PLC has a logic level of +24VDC, a logic level converter PCB must be designed and manufactured in order to ensure a bidirectional link for four channels. In the following figures are shown the PCB sectors of logic level conversion:

#### a) +24VDC $\rightarrow$ +3,3VDC

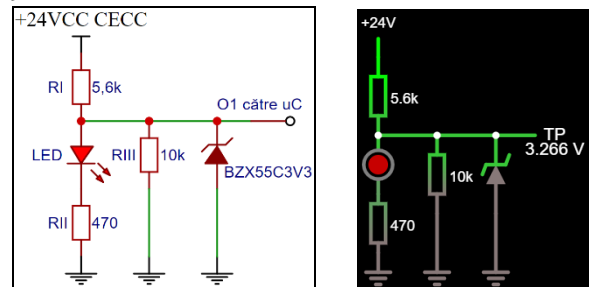


Figure 3.1 - Left - Electronic schematic; Right - Simulation

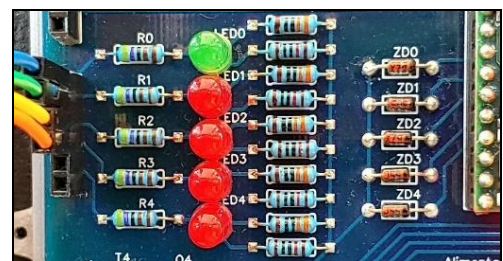


Figure 3.2 - Logic level converter

#### b) +3,3VDC $\rightarrow$ +24VDC

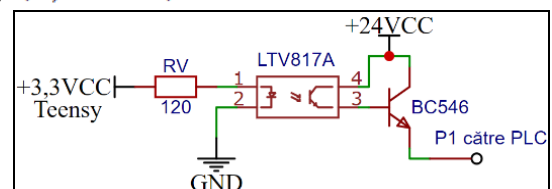


Figure 3.3 - Above - Electronic schematic; Below - Simulation

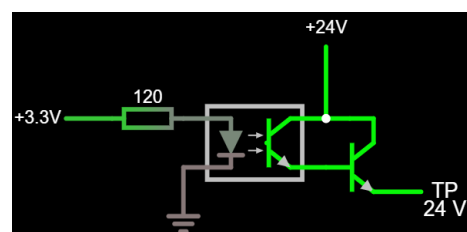


Figure 3.4 - Logic level converter

The designed PCB has auxiliary roles, such as: (1) Teensy  $\mu C$  install socket; (2) Extern access to I/O, UART,  $I^2C$ , SPI protocols; (3) NRF radio module install socket; (4) +5VDC and +3,3VDC power supplies for additional components if needed. The completed PCB is shown in the next figure:

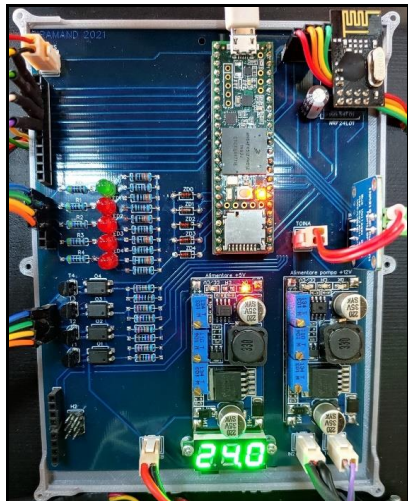


Figure 3.5 – Universal PCB

**B) Trajectory planning**

As stated before, the chosen autopilot to execute a predefined trajectory is *Mamba F405*. The trajectory is defined by connecting the autopilot to a laptop and accessing the dedicated software, iNav.

Firstly, the autopilot must undergo a calibration process: accelerometer, gyroscope and magnetometer calibration. Secondly, the autopilot must set to the right vehicle type, the rover, in order to use the correct control kinematics. Lastly, a trajectory is planned by using iNav implemented maps and placing blue waypoints across the alleys and alleys’ intersections. As it can be seen in Figure 3.6, the trajectory has a bigger resolution in intersections. This is needed in order to ensure that the robot will turn in the smoothest way possible.



Figure 3.6 – Trajectory planning

The autopilot operates in conjunction with a GPS module, C94-M8P and a magnetometer – Trajectory module, Figure 3.7:

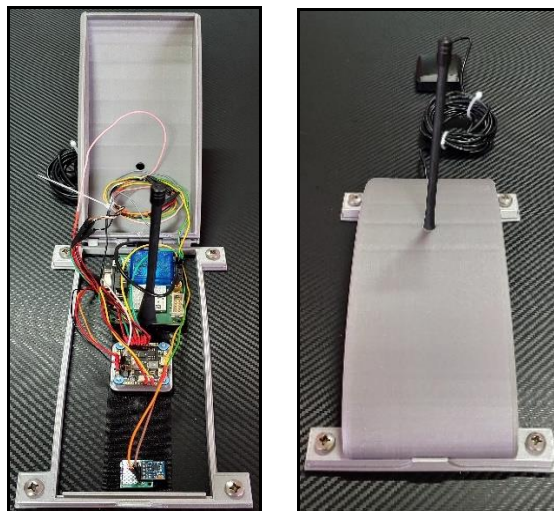


Figure 3.7 – Left – Trajectory module exposed; Right – Trajectory module covered

In order to enhance the trajectory accuracy, the GPS module must be also calibrated. Therefore, it must undertake a survey-in phase, where it samples satellite data in order to minimize the position deviation. The robot requires a minimum deviation of 1 meter. In the figure below, the survey-in process is completed, and the standard deviation is under 1 m, Figure 3.8.

Time Of Week:	286219.200 [s]
Status:	Successfully finished
Mean Position Valid:	Yes
Observation Time:	1158 [s]
Positions Used:	11577
Mean ECEF X:	4094807.6434 [m]
Mean ECEF Y:	2008395.0901 [m]
Mean ECEF Z:	4443869.9228 [m]
Mean 3D StdDev:	0.9176 [m]

Figure 3.8 – Survey-in process

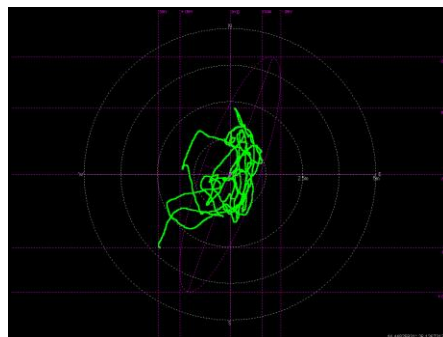


Figure 3.9 – Deviation map of the survey-in process

**C) EXCM Gantry control algorithm:**

The positioning system is represented in the following image, Figure 3.10:

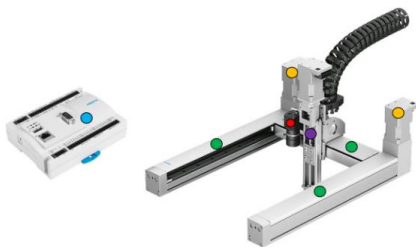


Figure 3.10 – EXCM Gantry [7]

Table 3.10 – EXCM Gantry legend

Color	Component	Code
Yellow/Green	Planar control gantry	EXCM-30-700-510-KF-SB-B1-5-EN
Purple	Z-Axis module	EGSK-26-200-2P
Blue	PLC	CECC-X-M1-MV
Red	Camera	SBPA-R2-C-U3-2E-2A-CS

In order to execute a precise and efficient 3D positioning motion, a motion control algorithm for the EXCM has been developed. The development platform used is Codesys, a development environment aiming industrial applications. The work-frame of the control algorithm is represented by the following diagram, Figure 3.11:

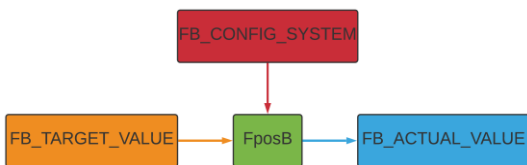


Figure 3.11 – Control algorithm work-frame

Where:

- FB\_CONFIG\_SYSTEM – the function block of containing configuration parameters;
- FB\_TARGET\_VALUE – the function block where the desired system state is set;
- FposB – the main function block that governates the entire algorithm;
- FB\_ACTUAL\_VALUE – the function block returns the actual system’s parameters.

The EXCM Gantry executes the positioning phase using the method *Point-to-Point*. That means that the Gantry needs a tridimensional cartesian coordinate that must be obtained from the object detection and centroid calculus. The detection algorithm sends the coordinate to the Point-to-point algorithm, Figure 3.12, and the EXCM is able to proceed to the positioning of the vacuum gripper.

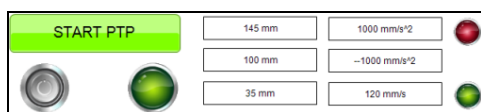


Figure 3.12 – Point-to-point control panel

The vacuum gripper is positioned in bidimensional plane described by the centroid coordinates. In order to execute a complete

tridimensional positioning, the PLC must know the vertical distance between the object and the vacuum gripper. In order to do so, a triangulation calculus is needed by using an infrared distance measurement sensor mounted behind the gripper. Figure 3.13:



Figure 3.13 – Infrared measurement sensor

### D) Object detection algorithm

In order to calculate the centroid coordinate of the waste object, an object detection algorithm must be developed. This is conducted by using two dedicated Festo computer vision software, CheckKon and CheckOpti, and following three steps, see Figure 3.14:

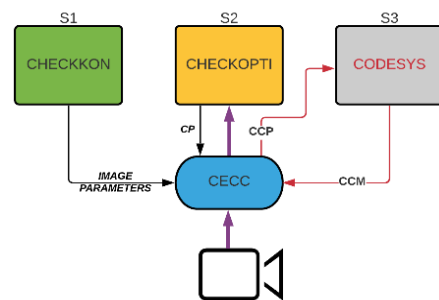


Figure 3.14 – Computer vision algorithms development steps

**Step 1 – S1** – This step is responsible with the software calibration of the camera and the preliminary pre-processing, such as image conversion from RGB to Grey-Scale color space, image segmentation, binary thresholding, morphological transformations, such as *Closing* or *Opening* [8]. These steps are carried on in the Festo provided software, CheckKon, Figure 3.15. This step is completed by saving the parameters to a dedicated file.

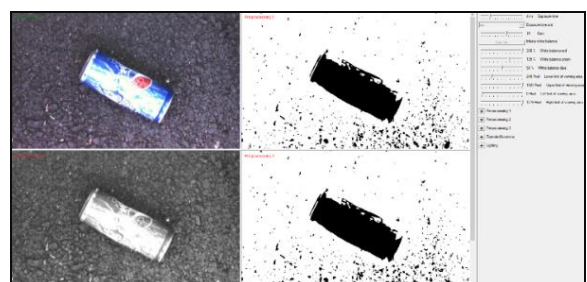


Figure 3.15 – CheckKon calibration panel

**Step 2 - S2** - After the previous step is completed, the next dedicated computer vision software is accessed, CheckOpti. Prior any related action, the saved parameters mentioned at **Step 1** must be loaded into CheckOpti's project. Next, in order to detect the object, a detection function must be used. In this case, the *Blob finder* function, or tool, is used. The *Blob finder* has the following features: calculus of the (x, y) coordinate in pixels of the blob relative to the camera's field of view and angle of inertia axis. Therefore, the relative position and orientation of the object can be obtained. In Figure 3.16, the orange square no. 1 indicates the detected blob along with its centroid coordinate,  $CC(579,796)$ , displayed with red color, and orientation  $\alpha = -89^\circ$ . The orange square no. 2 indicates the actual calculated values set in the program. In Figure 3.17, the data is: centroid coordinate  $CC(613,796)$ ,  $\alpha = 61,13^\circ$ .

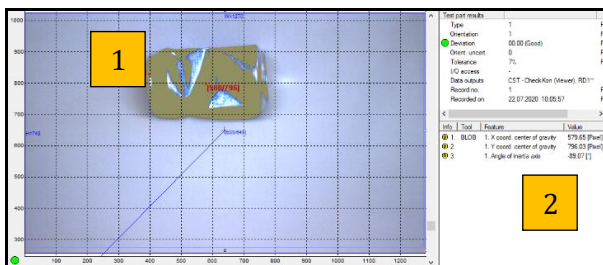


Figure 3.16 - Blob finder – laboratory test

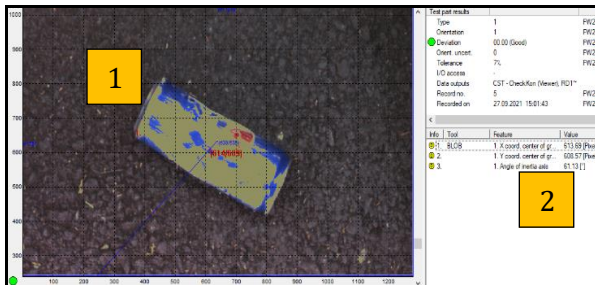


Figure 3.17.a)– Blob finder – real test conditions – parking lot



Figure 3.17.b)– Blob finder –real test conditions – parking lot – inside view

Next, the Checkprogram must be configured. A Checkprogram is a script that contains the function/tool set that the user needs to use within a specific computer vision situation.

Firstly, the data set needs a communication name in order to use within Codesys algorithm, see Figure 3.18.

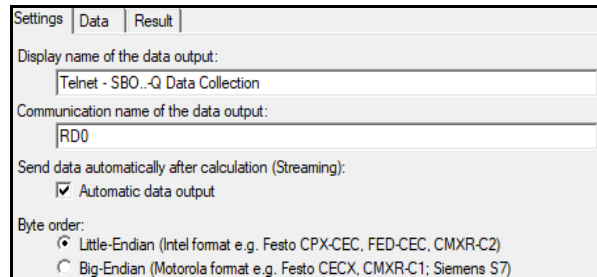


Figure 3.18 – Checkprogram – RDO

After the communication name is set, the data set must be set also. As stated before, the algorithm uses the Blob finder tool and centroid coordinates and angle of inertia axis calculus, Figure 3.19. After the Checkprogram is completed, it is uploaded directly to PLC's memory.

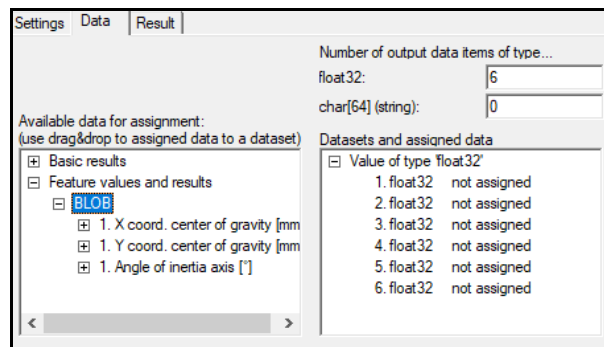


Figure 3.19 – Data set – RDO

**Step 3 - S3** - In this step the data from the previous step is called back and converted to millimeters so the EXCM Gantry to be able to use them for the planar positioning, Figure 3.20:

```

VisionSira(
    xEnable := TRUE,
    ulCheckProgram := 1,
    sNameOfDataSheet := 'RDO1');
v_state := 1;
IF VisionSira.eEvaluationMode = 0 AND
VisionSira.xTrigImage = TRUE THEN
CASE v_state OF
:
IF VisionSira.xTrigImage = TRUE AND
VisionSira.sImageDone = TRUE THEN
Arie := REAL_TO_INT(VisionSira.stRDO.value_REAL[0]/9);
DiX := REAL_TO_REAL((75+(75+(VisionSira.stRDO.value_REAL[1])/3.43)));
DiY := REAL_TO_REAL(((110-VisionSira.stRDO.value_REAL[1])/3.43)+55);
Unghtl := VisionSira.stRDO.value_REAL[3];
VisionSira.xTrigImage := FALSE;
TrpX := LREAL_TO_INT(DiX);
TrpY := LREAL_TO_INT(DiY);
v_state := 2;
END IF
: IF VisionSira.sImageDone = TRUE THEN
v_state := 1;
END IF
END_CASE
END IF
    
```

Figure 3.20 – VisionSira algorithm

The variables used for the planar positioning are the following:

- $DiX = VisionSira.stRDO.value\_REAL[2]$ ;
- $DiY = VisionSira.stRDO.value\_REAL[1]$ ;

These values are automatically sent the Point-to-point algorithm control panel, Figure 3.12.

### 4. Experimental Results

Next, object recognition experiments have been conducted in laboratory and outdoor conditions in order to determine the computer vision performance.

#### A) Laboratory:

These tests are conducted by placing an aluminum can in the camera's field of view in a total of 12 different positions, orientations and light conditions in order to determine the accuracy of the process. Next, a picture is taken for each position and the detection process is applied. The results are as follows, Figure 4.1 and Figure 4.2:

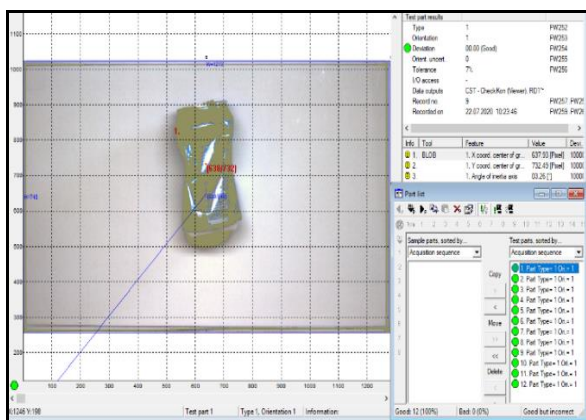


Figure 4.1 – CheckOpti screenshot

In Figure 4.2 are represented the 12 test images and CheckOpti indicates the fact that the recognition efficiency is 100%.

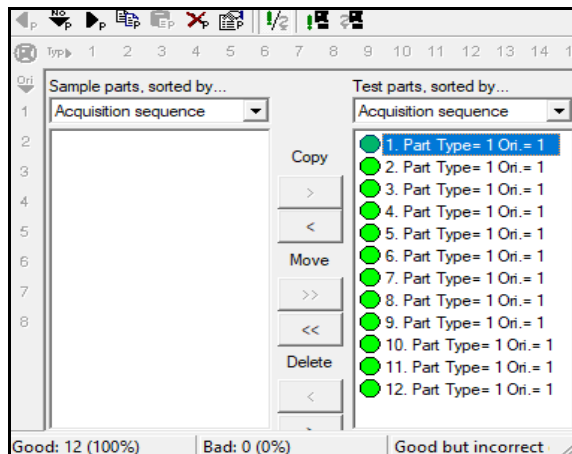
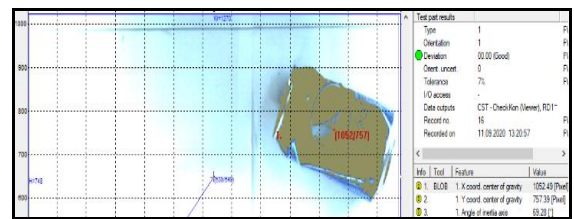
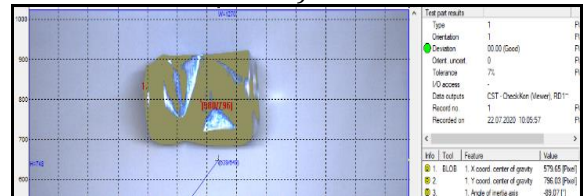


Figure 4.2 – Tests results

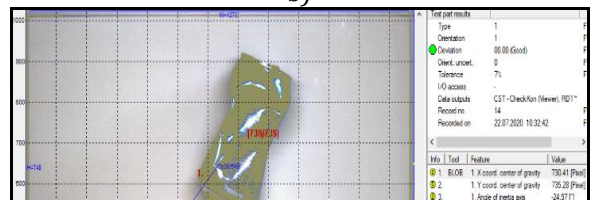
Furthermore, test images with various positions, orientations and light conditions are shown, Figure 4.3 – a), b), c), d):



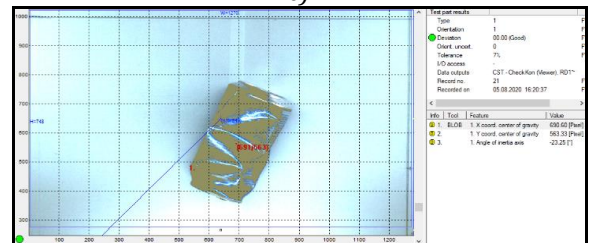
a)



b)



c)



d)

Figure 4.3 - a), b), c), d)

#### B) Outdoor

These tests are conducted by placing two different aluminium cans in the camera's field of view in a total of 12 different positions, orientations and light conditions, 6 for each can, in order to determine the accuracy of the process. See Figure 3.17 for exterior image during tests. Next, a picture is taken for each position and the detection process is applied. The results are as follows, Figure 4.4 and Figure 4.5:

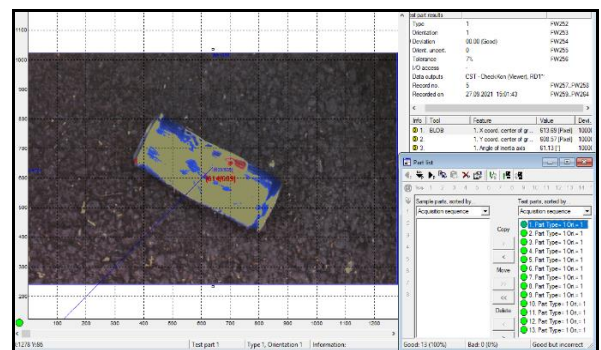
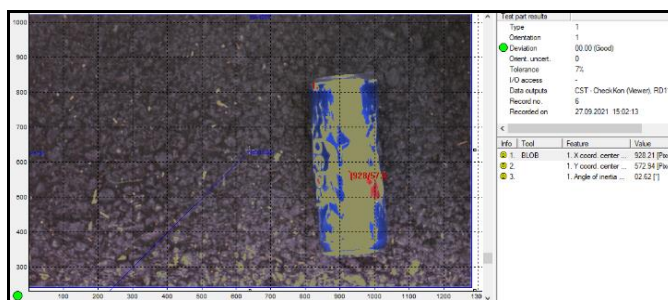


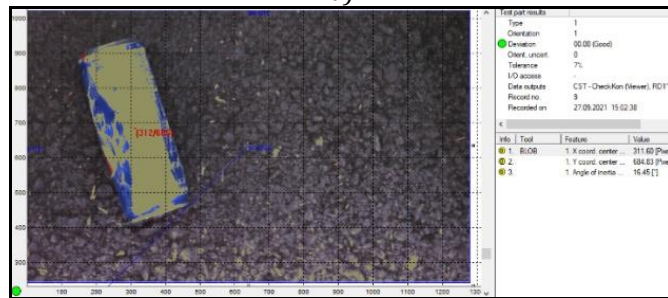
Figure 4.4 – CheckOpti screenshot

In Figure 4.5 and Figure 4.6 are represented the 12 test images and CheckOpti indicates the fact that

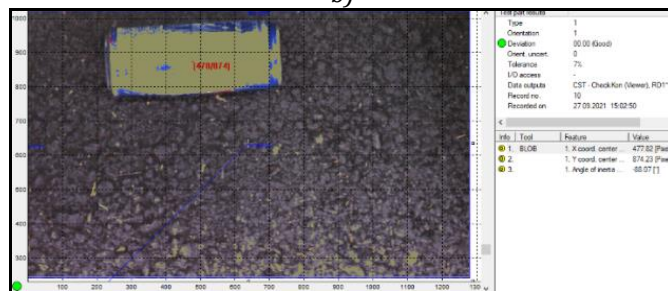
the recognition efficiency is 100% for both cans Blue can tests:



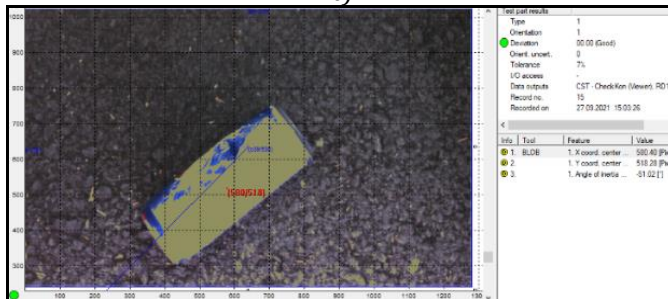
a)



b)



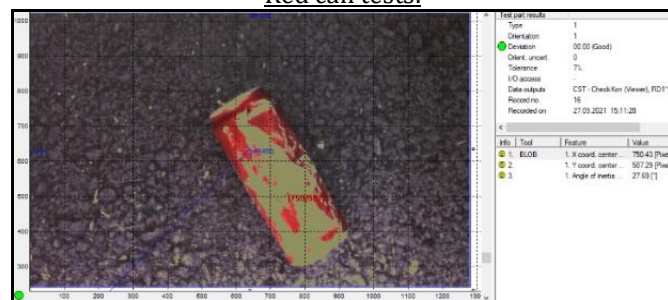
c)



d)

Figure 4.5 – a), b), c), d)

Red can tests:



a)

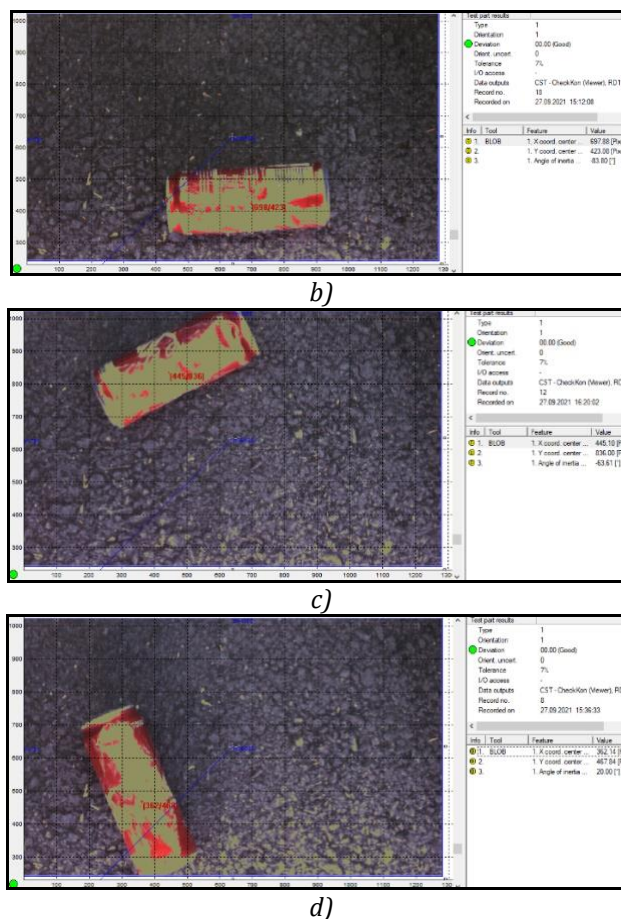


Figure 4.6 – a), b), c), d)

## 5. Conclusions

Based on an industrial Festo hardware and software, the SIRAMAND system has also proved to be efficient in a non-industrial environment, more precisely in an outdoor urban environment. Activities carried out in parallel with those described in this paper may lead in the near future to implementation of waste recognition and collection based on an artificial intelligence algorithm.

## Acknowledgements

This work was supported by a grant of the Romanian Ministry of Research and Innovation, CCCDI – UEFISCDI, project number PN-III-P1-1.2-PCCDI-2017-0086 / contract no. 22 PCCDI /2018, within PNCDI III.

## References

- [1] <http://www.theoldrobots.com/dustcar.html>
- [2] <https://www.youtube.com/watch?v=860Lxp8wfk>
- [3] M. Mărgăritescu, P. N. Ancuța , E. V. Canale, D. I. Stanciu, D. Dumitriu, C. M. Brișan - *Control of an Autonomous Mobile Waste Collection Robot* -

Proceedings of the International Conference of Mechatronics and Cyber-MixMechatronics – 2019

- [4] D. Dumitriu, M. Ionescu, D.O. Melinte - *New scissor-grasping tool for parallel grippers, ensuring stroke enhancement* - (Extended Abstracts) 36th Danubia Adria Symposium on Advances in Experimental Mechanics, pag. 131-132, Plzeň, Republica Cehă, 24-27 Septembrie 2019.
- [5] D. O. Melinte, A.-M. Travediu, D. Dumitriu - *Deep Convolutional Neural Networks object detector for real-time waste identification* - Applied Sciences, ISSN 2076-3417, vol. 10(20), 2020, special issue “Bio-inspired Computation and Applications”, ID articol 7301, <https://doi.org/10.3390/app10207301>, open access: <https://www.mdpi.com/2076-3417/10/20/7301>, IF 2.474 (2019).
- [6] A. Stan, D.M. Cotorobai, “Centroid and planar rotation calculus using parallel computational geometry for regular shaped objects” - International Journal of Mechatronics and Applied Mechanics, year, Issue 9, 2021
- [7] [https://www.festo.com/cms/ro\\_ro/20107.htm](https://www.festo.com/cms/ro_ro/20107.htm) accessed at:20.09.2021
- [8] <https://realpython.com/python-opencv-color-spaces/> accessed at: 20.09.2021