

Robot Chow: Automatic Animal Feeding with Intelligent Interface to Monitor Pets

Rogério Nogueira, Humberto Araújo, David Prata

Department of PPGMCS, Federal University of Tocantins, Brazil

Abstract— According to the latest IBGE (Brazilian Institute of Geography and Statistics) survey, there are more than 74.3 million dogs and cats in Brazil. Moreover, the highest cost for raising these animals is pet feed, which should be served at appropriate times and quantities. In contrast, the daily lives of pet breeders have often unforeseen events such as meetings, last minute trips, traffic jams, among others. With the purpose of helping the difficulties faced by small domestic animal breeders, this work presents the implementation of a solution called RobotChow, capable of serving granulated feed and water exchange at pre-scheduled times and volumes for pets; in such a way that the user can monitor if the animal is feeding. For this, the customer can use instant messaging on the mobile, with a solution interface that controls the features only interacting ("chatting") with the application. In addition to providing commands, the solution also send photos or even notify the user if something is unusual to happen with the pet. The user can monitor his/her pet regardless of space-time, just accessing the internet. The entire solution was developed in Raspberry pi, using Python, and it is integrated with the Watson Assistant and Telegram cloud services.

Keywords— Pet Feeder, Artificial Intelligence, Raspberry, Chatbot, Pet Breeders.

I. INTRODUCTION

According to a survey from the Brazilian Institute of Geography and Statistics (IBGE), around 63% of all households in Brazil have at least one dog or cat, totaling 52.2 million dogs and 22.1 million cats in Brazil country [1]. In another survey carried out by the National Confederation of Shopkeepers (CNDL), in conjunction with the Credit Protection Service (SPC Brazil), 52% of the respondents buy premium pet feed (higher digestibility), and these data show that the Brazilians are worried about the quality of life of their pets.

As discussed above, quality food is one of the major concerns of Brazilian animal breeders. Besides the quality, the control of the amount of feed per day is fundamental for the maintenance of the body weight of the animal and must be calculated based on the

metabolizable energy of the food due to the estimated energy requirement of the animal, according to the Brazilian Association of Animal Estimation Products [2]. In addition to controlling the amount of food, it is imperative that the breeder could monitor whether the animal has been eating. Otherwise, this fact may be a strong indicator of problems for the animal or in the food served to it.

Another important aspect is the feeding times of the animal, which may vary according to race, size and age. The fact is, every animal should be fed at least once a day. Ensuring this happens is a basic and fundamental function for any pet breeder.

Faced with these challenges, pet breeders often find it difficult to overcome, for a variety of reasons, such as an unexpected trip or simply being stuck in traffic jam. Or even, commitments arising from a frenetic daily life.

Currently on the market, we can basically find two product groups that serve food to animals. The first one counts on products that function as prescheduled food dispensers, such as the feeders of manufacturers Amicus [3], Eyenimal [4] and Hoison [5]. The second group is focused on giving snacks to the animal seeking an iteration with the same, as the products developed by Pawbo [6], Petcube [7] and Furbo [8]. We observed that none of them has the monitoring features, with notifications and automatic feeding in the same device.

The solution we present here aims to make life easier for dog and cat breeders; by automating the serving of pet foods and water exchange, with controlled quantity and prescheduled times. In addition, the application allows to visually monitoring the animal through photos taken throughout the day. It can happens by automatic detection of animals' presence in strategic locations.

All the management of the solution is through an intelligent interface, which makes use of a conversational agent that interacts with the user through natural language processing techniques and artificial intelligence, known as chatbot [9]. The application seeks to help the user by answering questions, giving the impression of talking to a person, not a machine.

The physical part of the solution consists of low cost or reused equipment such as PVC pipes, solenoid valves,

proximity sensor, DC motor, 5 megapixel camera and 4 channel relay module, all connected to a Raspberry pi 3 microprocessor.

The purpose of this work is to present the development of a solution capable to automatic monitoring the feeding of dogs and cats, making use of intelligent and mobile interface. This application can allow: i) to control the meal dosage; ii) schedule meals online; iii) automatically change the water; iv) photographically record the animal being fed; v) detect the presence of the animal; and vi) create a double track line between the solution and the user.

II. DEVELOPMENT

The development of the solution was basically divided into mechanics, electronics and software.

A. MECHANICS

The mechanical part consists on the system of dispersion of granulated solid foods. Where it uses a pipe of PVC with 100 millimeters of diameter and one meter of height, working like a warehouse of foods. At the lower end of the barrel, it was added a rotary valve, made with PVC shaft and six 4 mm thick EVA fins. It should be noted that this choice is due to the malleability of the material. Many granulated foods are somewhat resistant and may cause valve locking when the fin presses the grain against the wall of the barrel.

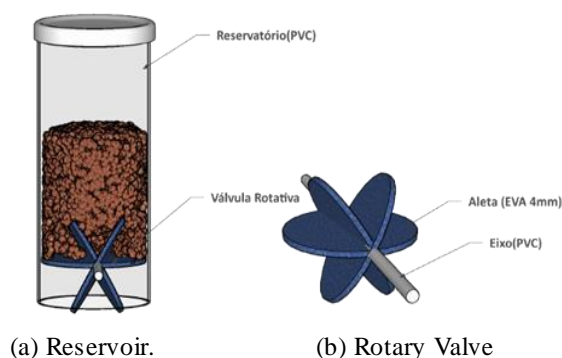


Fig. 1. Rotary valve and PVC food tank.

In order to prevent the animal from carrying shock, as well as gauging greater resistance to the components, the motor was coated with a wooden box. All wiring has been coupled and protected at the back of the equipment, as well as the solenoid valve.

B. ELETRONICS

To turn the rotary valve, we use an AKYAMA motor, with a speed of 5 RPM. It has been chosen because of a torque of 70Kgf.cm with a low cost. When the motor is energized with a voltage of 220V [10], it starts to rotate its axis which is connected to the valve axis by means of a flexible coupling of 8 mm. When rotating its axis, the

food that is between the fins fall in the food container of the animal in a moderate form. Therefore, the amount of food is given by the control of time that the engine is energized.

An electromechanical valve, known as a solenoid valve, is triggered to control the flow of liquids. When resting, the plunger inside the valve closes the duct for passing the liquid. When energized, the piston changes position to unclog the orifice, allowing the liquid to flow [11].

In this project, we used a manual tap connected to the water box of the residence. When open, the tap dispenses water directly into the animal's container by the solenoid valve that controls the water flow through the application of 220v voltage at its poles.

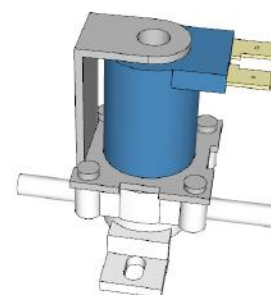


Fig.2: Solenoid Valve

Source: [12]

Relays are used to control the energization of the motor and solenoid valve. The relays are eletromechanical devices that are able to close the circuit when energized [13]. A 4-channel relay module is used, where each channel, when energized with 5 volts, is capable of closing a circuit of 10 amperes and voltage of 250v.

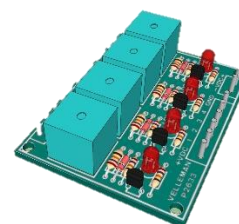


Fig. 3. 4-channel relay module.

Source: [12]

To find out if the animal is near the food container, we put an ultrasonic sensor, developed to measure the distance between where the sensor is and where the animal is. The sensor uses ultrasonic waves, which are inaudible by humans. The sensor used in this project is the HC-SR04, which measures the distance to a shield by calculating the timepiece between the emission of ultrasonic waves and the return of the waves after reaching the target [14].



Fig. 4. PVC box with proximity sensor and camera.

When the presence of the animal is detected, the system takes pictures to visually confirm if it is feeding. For this, we use a 5MP camera, which connects directly to the Raspberry microprocessor through the CSI (camera serial interface) port. This is a small and very accessible computer that can be used to learn programming [15]. It is a low cost microprocessor that is ideal for generic machines, control systems and units which need to generate less heat and spend less energy [16].

The version used in this project is the Raspberry pi 3 b +, which features a Broadcom BCM2837B0 64bits ARM Cortex-A53 Quad-Core processor, 1.4 Ghz speed, 1 GB of RAM and Bluetooth adapter. This component is responsible for controlling all the mentioned devices, as well as housing and running the software that make up this solution.

A powerful feature presented in the Raspberry pi 3 is the GPIO line, a general-purpose one, consisting of 40 pins for input and output. This line acts as interface for the control of the devices [17].

In the diagram below (Fig. 5) the connections between the raspberry and the other equipments are represented.

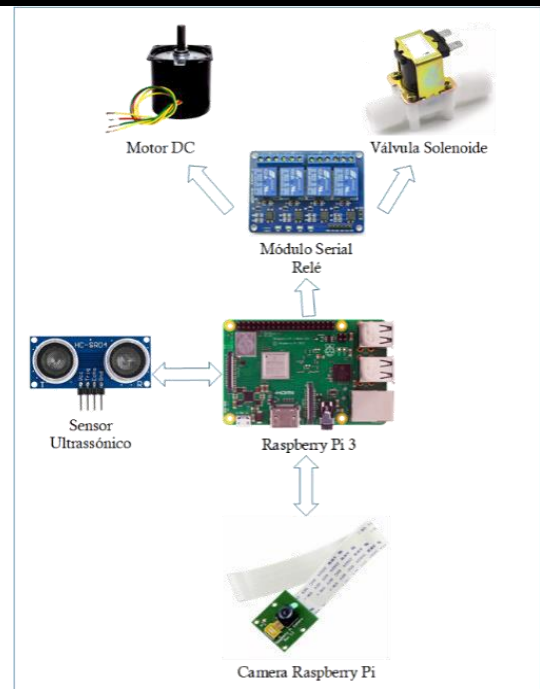


Fig. 5. PVC box with proximity sensor and camera.

Table I shows the connections between the Raspberry Pi GPIO ports and the device connections.

C. SOFTWARE

The Operating System (OS) used is Raspbian 4.14 kernel version, based on Debian and optimized for Raspberry. The software is completely free, counting on more than 35 thousand packages of precompiled systems. The software is being maintained by a community of free software, with emphasis on stability and performance [18]. This system supports the entire software framework that makes up the solution, as we can see in figure 6.

Raspberry has memory and processing limitations. Therefore, all the technologies employed in this project were thought in the light of these restrictions. In this sense, we opted for a micro server, light and agile, called bottle in version 0.12 (stable), without dependencies; except the language module used, capable of being executed from a single file [19].

The language used in the development of the applications that make up the solution is Python, in version 3.7. Because it is a high-level, object-oriented language, Python can be used across platforms because it is interpreted [20]. This language has been shown to be a good choice for speed of development and maintenance; and has established itself as one of the most popular languages of scientific computing [21].

Table I: Conections' Map

GPIO Port	Device Port	Device
2	VCC	Relay Module
4	VCC	Proximity Sensor
6	GND	Proximity Sensor
16	TRIG	Proximity Sensor
18	Echo	Proximity Sensor
38	IN2	Relay Module
39	GND	Relay Module
40	IN1	Relay Module

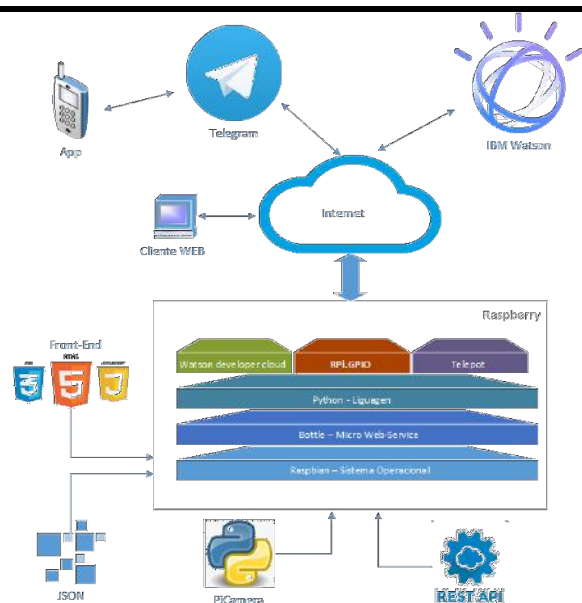


Fig. 6. Software Architecture.

All quoted scripts are initialized from `service.py`, which makes use of the threading module to be triggered independently. Modules are Python files, which can contain definitions, functions, and classes. Modules are initialized the first time they are imported (called) within the application [22]. Often, modules are used to abstract objects, being used in this project.

- RPi.GPIO provides a class to promote GPIO control of raspberry pi [23]; the version used in the project is 0.6.4.

- Telepot is used to help the communication of the application with the Telegram API [24]; the version used in the project is 12.6.

- Watson Develop Cloud: package used for interaction between the application and the IBM Watson API; used in this work version 2.4.0. Watson is a tool created by IBM, which makes use of artificial intelligence techniques such as machine learning and natural language processing. Watson is also used to support the construction of autonomous applications that simulate human speech, called bots or chatbots [25].

- Schedule: this module has the function of executing tasks in pre-scheduled periods by executing functions in python [26]. We opted for this module in version 0.4.3, due to its simple and friendly syntax, as well as its ability to run multiple scripts in parallel.

- PiCamera: A package that acts as interface between the Raspberry Pi Camera Module and the applications developed in Python [27]; version 1.13.

- Threading: Python module used to execute multiple processes in parallel [28]; you can activate and disable multiple scripts (applications) at the same time.

- JSON: As the name suggests, the module facilitates interaction with data in JSON format [29]. The format is

used to save system configuration information, thus avoiding the need to install a database.

D. Interface development

Seeking to promote a more comfortable experience for users, the system has two interfaces. One via web and another via the cloud-based instant messaging service, known as Telegram [30].

The Telegram provides the API Bot, fully open to developers, and customizable. It is an API that is able to connect bots to the telegram through special accounts; thus serving as a conversation interface between the application and the user [31].

To integrate Watson into the Telegram, the `chatbot.py` script was created, which connects to the Watson API and authenticates the previously registered user. The script also connects to the Telegram through the token specially created for the use of bots. Subsequently, the script enters a loop, forwarding messages from the telegram users to the Watson service. Watson interprets and responds to messages based on a question-and-answer database previously discussed. Such an answer is subsequently returned to the Telegram chat. During this message routing, the script analyzes them, checking if there are no commands to execute such as: taking a picture, serving a meal, exchanging water and others. For security reasons, only previously registered Telegram users can interact with the bot created on the platform.

The second interface, now in web format, is intended to make it easier to schedule daily tasks, such as serving food and changing water through forms. In addition, this interface can also pass commands to perform the tasks mentioned in real time. Another feature of the web interface is the visualization of the gallery of photos taken during the monitoring of the animal. The predefined structures of the bottle dynamize and promote the reuse of the HTML, JavaScript and CSS codes used in the development of these pages.

The scheduling data made by the web interface is saved in JSON format in the `config.json` file. Later, this data is read by the `schedule.py` script, which uses the schedule module to perform scheduling of the serving meal and / or serving water functions; depending on the configured parameters, according to the flow diagram of figure 7.

The `sensorprox.py` script has the function of taking a photographic record every time the animal eats. Thus, the user is certified that the animal has fed. A proximity sensor installed above the feeder detects the distance variation between it and the nearest bulkhead. When this distance reaches 10 cm, the script sends the command for photographic registration. The camera starts shooting every 3 seconds. After recording 5 photos, the system suspends monitoring operations for 60 minutes, saving memory space.

In determining a preset time, the system checks if there are records of the animal on that day. Otherwise, the breeder receives an alert in the Telegram informing that the animal did not go to the feeder. This can be interpreted by the creator as a hint of a problem.

The photos can be viewed either by request via Telegram or on the "gallery" page of the WEB interface.

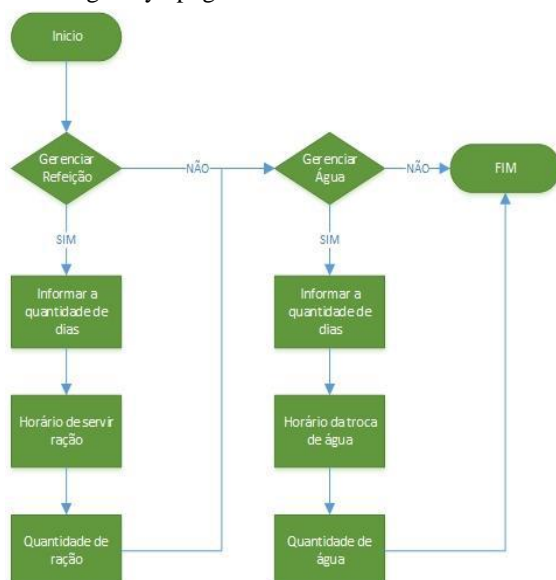


Fig. 7. Scheduling flowchart.

III. CONCLUSIONS

During the development of this solution, we realize not only the importance of animal's health for breeders, but also their satisfaction and disposition to invest much of their income in the quality of animal's life.

The solution is economically viable, with a cost of production orbiting around \$150.00 dollars. The pet feeders cited in this research, which have access by mobile applications, can cost up to three times more. In addition, to making use of low-cost materials, we have chosen an open technology framework using free cloud services. And the technology used here can still be replaced. For example, instead of using Telegram, one can opt for any instant messaging tool like Facebook Messenger or WhatsApp. The Watson Assistant can be replaced by the services of Dialogflow, Wit.ai, Luis.ai and others.

The constant speed of 5 revolutions per minute of the chosen motor facilitated the control of the amount of pet food to be served. But, to know the exact ideal quantity, some tests should still be carried out with the ration chosen by the person in charge of the animal. Since each ration has its own density and granulation.

The use of an instant messaging interface has several advantages, such as, the need to install a new application and the creation of a single channel to receive notifications. The interface also receive photos of the

animal directly on the mobile device, making it easier to share the image on social networks.

At the end of the work, we can conclude that the solution has a reasonable and flexible cost, able to provide food and water for pets. The solution also allows to monitor the animals through photographic records and notification of possible problems. The application makes use of an interface with the ability to interact with the user through natural language processing techniques, sending and receiving instant messages. As a result, the application can provide greater comfort and convenience to the dog and cat breeders.

ACKNOWLEDGEMENTS

We acknowledge ESMAT (Superior School of Magistrates of Tocantins) for financial support.

REFERENCES

- [1] "Sistema IBGE de Recuperação Automática - SIDRA". [Online]. Available in: <https://sidra.ibge.gov.br/pesquisa/pns/volume-2>. [Accessed: 03-nov-2018].
- [2] "9ª Edição". [Online]. Available in: <http://www.mflip.com.br/pub/abinpet/index3/#page/25>. [Acessado: 03-nov-2018].
- [3] "Alimentador Eletrônico - Amicus". [Online]. Available in: <http://www.amicus.com.br/alimentador-eletronico.html>. [Acessado: 22-nov-2018].
- [4] "Electronic Pet Feeder". [Online]. Available in: <http://www.eyenimal.com/en/health-and-wellness/16-electronic-pet-feeder.html>. [Acessado: 22-nov-2018].
- [5] "Alimentador Inteligente p/ Pet", Hoison. [Online]. Available in: <http://hoison.com.br/produto/alimentador-inteligente-para-pet>. [Acessado: 22-nov-2018].
- [6] "Pawbo". [Online]. Available in: <https://www.pawbo.com/tw/>. [Acessado: 22-nov-2018].
- [7] "Petcube Bites - HD Pet Treat Camera That Monitors And Dispenses Treats", Petcube web-site. [Online]. Available in: <https://petcube.com/bites/>. [Acessado: 22-nov-2018].
- [8] "Furbo 丟零食狗狗攝影機 | Furbo智慧寵物攝影機, HD畫質+紅外線夜視, 讓你能隨時觀看無死角, 還能和毛孩說話, 遠端丟零食互動!". [Online]. Available in: <https://shoptw.furbo.com/>. [Acessado: 22-nov-2018].
- [9] "ALICE Chatbot: Trials and Outputs". [Online]. Available in: http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1405-55462015000400625&lng=es&nrm=iso&tlng=en. [Accessed: 15-nov-2018].

- [10] R. S. P. Portela, “Disponibilizamos os Desenhos Técnicos 2D e 3D no site www.neoyama.com.br”, p. 2.
- [11] “Solenoid valve type differences - Tameson.com”. [Online]. Available in: <https://tameson.com/solenoid-valve-types.html>. [Accessed: 16-nov-2018].
- [12] “3D Warehouse”. [Online]. Available in: <https://3dwarehouse.sketchup.com/>. [Accessado: 27-nov-2018].
- [13] “Relê (Relay) - Definição de Relê (Relay)”. [Online]. Available in: <https://www.hardware.com.br/termos/rele-relay>. [Accessado: 16-nov-2018].
- [14] “Interfacing HC-SR04 Ultrasonic Sensor with Raspberry Pi”, electroSome, 30-jan-2015. .
- [15] R. P. Foundation, “Raspberry Pi — Teach, Learn, and Make with Raspberry Pi”, Raspberry Pi. [Online]. Available in: <https://www.raspberrypi.org>. [Accessado: 16-nov-2018].
- [16] “Como funciona o Raspberry Pi? Entenda a tecnologia e sua aplicabilidade”, TechTudo. [Online]. Available in: <https://www.techtudo.com.br/noticias/noticia/2014/11/como-funciona-o-raspberry-pi-entenda-tecnologia-e-sua-aplicabilidade.html>. [Accessed: 16-nov-2018].
- [17] “GPIO - Raspberry Pi Documentation”. [Online]. Available in: <https://www.raspberrypi.org/documentation/usage/gpio/>. [Accessed: 17-nov-2018].
- [18] “FrontPage - Raspbian”. [Online]. Available in: <https://www.raspbian.org/>. [Accessado: 18-nov-2018].
- [19] “Bottle: Python Web Framework — Bottle 0.13-dev documentation”. [Online]. Available in: <http://bottlepy.org/docs/dev/>. [Accessado: 18-nov-2018].
- [20] “The Python Tutorial - Documentação do Python 3.7.1”. [Online]. Disponível em: <https://docs.python.org/3/tutorial/index.html>. [Accessed: 18-nov-2018].
- [21] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python”, J. Mach. Learn. Res., vol. 12, no Oct, p. 2825–2830, 2011.
- [22] [22] “6. Módulos — Python v2.7.2 documentation”. [Online]. Disponível em: <http://turing.com.br/pydoc/2.7/tutorial/modules.html>. [Accessado: 18-nov-2018].
- [23] B. Croston, RPi.GPIO: A module to control Raspberry Pi GPIO channels. .
- [24] “Introduction — telepot 12.7 documentation”. [Online]. Disponível em: <https://telepot.readthedocs.io/en/latest/>. [Accessado: 18-nov-2018].
- [25] “O que é chatbot? Entenda como funciona o robô que conversa com você”, TechTudo. [Online]. Disponível em: <https://www.techtudo.com.br/noticias/2018/03/o-que-e-chatbot-entenda-como-funciona-o-robo-que-conversa-com-voce.ghtml>. [Accessado: 18-nov-2018].
- [26] “schedule — schedule 0.4.0 documentation”. [Online]. Available in: <https://schedule.readthedocs.io/en/stable/>. [Accessado: 18-nov-2018].
- [27] “picamera | Read the Docs”. [Online]. Available in: <https://readthedocs.org/projects/picamera/>. [Accessado: 18-nov-2018].
- [28] “16.2. threading — Higher-level threading interface — Python 2.7.15 documentation”. [Online]. Available in: <https://docs.python.org/2/library/threading.html>. [Accessado: 19-nov-2018].
- [29] “json — JSON encoder and decoder — Python 3.7.1 documentation”. [Online]. Available in: <https://docs.python.org/3/library/json.html>. [Accessado: 18-nov-2018].
- [30] “Telegram – Apps no Google Play”. [Online]. Available in: https://play.google.com/store/apps/details?id=org.telegram.messenger&hl=pt_BR. [Accessado: 18-nov-2018].
- [31] “Telegram APIs”. [Online]. Available in: <https://core.telegram.org/>. [Accessado: 18-nov-2018].