Hittite Journal of Science and Engineering, 2018, 5 (4) 317-320
ISSN NUMBER: 2148-4171
DOI: 10.17350/HJSE19030000110

Low-cost IoT Design and Implementation of a Remote Food and Water Control System for Pet Owners

Ugur Baran Asaner **Armagan Elibol**
Yildiz Technical University, Department of Mathematical Engineering, 34220, Istanbul, Turkey

Article History:
Received: 2017/11/30
Accepted: 2018/08/03
Online: 2018/12/31

***Correspondence to**: Ugur Baran Asaner*
Netaş, Kurtkoy, Istanbul
E-Mail:*ugurasaner@gmail.com*

ABSTRACT

The Internet of things (IoT) concept basically aims to connect any device, vehicle or other items to transfer data between these subsystems. IoT applications make people's lives easier and more efficient in many ways. One of the areas where IoT can be useful is that monitoring food and water supply for pets that are left unattended for either a short or a long time. The main purpose of this paper is to state and detail an instance of low-cost IoT by designing a remote food and water control system for pet owners. The whole system consists of three subsystems; the performing unit, server, and mobile application. Each subsystem has developed by using different open-source programming languages and frameworks.

Keywords:
Internet of things, remote monitoring, open-source development

INTRODUCTION

The Internet of things (IoT) is commonly used to name a set of objects that are directly connected to the Internet. One prediction is that the number of connected things in the world will have a thirty-fold increase between 2009 and 2020, thus by 2020, there will be 26 billion things that are connected to the Internet [1, 2]. Typical application areas for the IoT are home automation [3], personal health monitoring [4, 5], building automation [6], industrial automation [7] and smart cities [8]. The purpose of this paper is to implement a design of low-cost IoT concept, which is a food and water control system for pet owners. There are previous studies on the context of pet feeding: Some of them state an automated system without the remote control [9], while some of them require additional components to be bonded to pets [10]. These studies introduce remote scheduled or automated systems. However, most of these studies do not provide instant access to food or water amounts, which may be important for some pet owners. There are also commercial products aiming to provide automated [11] or remote-controlled [12] pet feeding systems with higher costs. Our proposal is easy to use, low-cost and does not require precise calibration and installation work. Our project consists of three subsystems; performing unit, server, and mobile application. The subsystems and the communication between them have detailed in Section Design and Implementation.

DESIGN AND IMPLEMENTATION

Our IoT system is designed to provide food and water control for pets and it is controlled by a mobile application interface. The user can observe and monitor the amounts of food and water remotely by using the interface. The user can also refill the food or water if needed. Our system is composed of three main components: Performing unit, server and mobile application respectively. The server has a connecting role between other components as it manages the requests and responses between the performing unit and mobile applications. It also provides authentication to use the system. The information coming from the sensors is stored in the database component, which only accessed by the server. The performing unit has sensors to measure the amounts and motors to refill and the mobile application provides a Graphical User Interface (GUI) for the user to access the server functions. Thus, the system does not allow the mobile application to contact with neither the performing unit nor the database directly. The overview of the system components can be seen in Fig. 1.

Performing Unit

Performing Unit is the major part of our IoT system. It is responsible for core processes such as measuring current food and water amounts and performing refill operation separately for both food and water. The main component of the unit is a single-board computer (SBC) whose role is to control the sensors and motors using some specific drivers. This SBC is

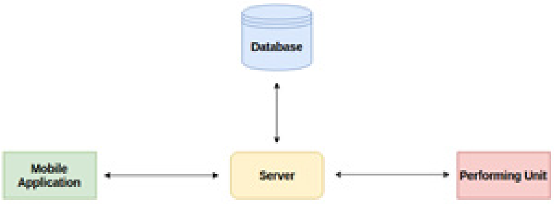

Figure 1. Component relation diagram of the designed IoT.

connected to the internet so that it can receive requests remotely and transmit sensor data to the server at a pre-defined time interval (i.e., hourly). The overview of the unit is given in Fig. 2.

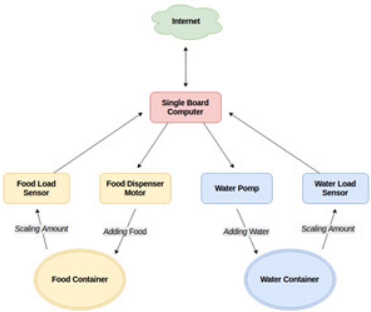

Figure 2. Performing Unit components and flow diagram of the designed IoT.

The unit has a food dispenser and a water tank as storage. There is a motor to refill the food container and water pump to refill water container. Both containers are attached to suitable sensors to measure the amounts. The design sketch of the performing unit is illustrated in Fig. 3.

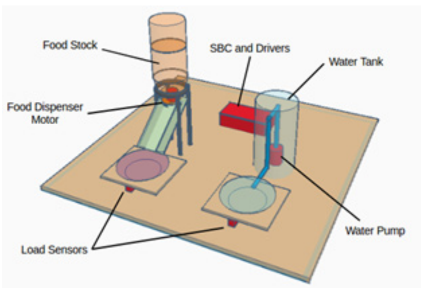

Figure 3. Design sketch of the Performing Unit.

The technical details of the performing unit are detailed in two parts, hardware and software respectively. Each of them will be detailed separately.

Performing Unit Hardware

Performing unit hardware is composed of an SBC, a DC and a Stepper motor, and two load sensors. We used Raspberry Pi B+ model as an SBC and it has General Purpose Input / Output (GPIO) pins, which allow the computer to send and receive signals. There are forty pins available on the SBC and seventeen of them are used by three different components as illustrated in Fig. 4.

Used For	Function	Pin No	Pin No	Function	Used For
	3.3 V	1	2	5 V	Load Sensors 1-2
	GPIO 2	3	4	5 V	DC Motor
	GPIO 3	5	6	GND	DC Motor
	GPIO 4	7	8	GPIO 14	DC Motor
Load Sensor 1	GND	9	10	GPIO 15	DC Motor
Load Sensor 1	GPIO 17	11	12	GPIO 18	DC Motor
Load Sensor 1	GPIO 27	13	14	GND	
	GPIO 22	15	16	GPIO 23	
	3.3 V	17	18	GPIO 24	
	GPIO 10	19	20	GND	
	GPIO 9	21	22	GPIO 25	
	GPIO 11	23	24	GPIO 8	
	GND	25	26	GPIO 7	
	DNC	27	28	DNC	
	GPIO 5	29	30	GND	
	GPIO 6	31	32	GPIO 12	Stepper Motor
	GPIO 13	33	34	GND	Stepper Motor
Load Sensor 2	GPIO 19	35	36	GPIO 16	Stepper Motor
Load Sensor 2	GPIO 26	37	38	GPIO20	Stepper Motor
Load Sensor 2	GND	39	40	GPIO 21	Stepper Motor

Figure 4. SBC Pins usage diagram

The DC motor is used for designing a water pump to refill water container from the water tank. This motor is controlled by an H-bridge driver named 'l293b'. The stepper motor is responsible for refilling the food container from the food tank. The working principle of the motor is based on a vector-based position declaration. This allows defining the rotation with a specific angle. Two load sensors provide weight data and the data is read by SBC using 'hx711' driver. The range of the sensor is from 1gr. to 5kg. The sensor does not require an external power supply to work.

Performing Unit Software

The software side of the performing unit accomplishes two main tasks: The first task is to control electronic components and the second one is to make the performing unit accessible over the internet to handle the remote requests. Python programming language is preferred for the first task as there are many sources available and there are also included libraries to control the pins on the SBC. For a second task, open-source web framework Django is used to develop a web server. The performing unit includes a database to store some basic configurations, which makes the development easier and adjustable to different implementations. This database stores API paths, GPIO pin numbers, server IP address, the current amount of food and water in the containers, tare and threshold values for scaling, and the step size of the stepper motor.

The SBC has the operating system called 'raspbian', which is an optimized version of Debian for low-performance architecture. The performing unit sends the current amount of data to the server hourly, 'cron' is preferred to use on this task, which is a time-based job scheduling software utility in Unix-like operating systems. Cron is used by defining a cronjob, which could be any task on command line interface. On the other hand, Django has an external package to manage cronjobs in the project, which is called Django-cron [18]. The package works on the operating system's conjob utility and allows defining classes in Python language, which is suitable for our project. The cronjob checks the defined classes every five minutes and the classes send the data of current amounts to the server, if necessary.

A current control class in the GPIO library is used to control the DC motor. This class allows running the motor for five seconds, which pumps approximately 50ml. of water from the tank to the container. The motor starts running when the appropriate pin set as high current and stops when the pin set as low current.

The stepper motor is also controlled by a class in GPIO library. Besides the pin numbers, the number of steps is obtained from the internal database of the performing unit and provided for the class.

The load sensors require an additional library with the same name as the driver chip 'hx711'. Load sensors used are prone to return inconsistent sensor readings sometimes. These outliers should be filtered out. To do this, several readings from the sensor are made and extreme readings (as minimum and maximum) are removed. The arithmetic mean of the remaining readings is computed and returned as data. The library called the same name with the driver chip, hx711 is used.

Server

The server is running on a Virtual Private Server (VPS) that has an Ubuntu Server operating system. Laravel web framework is preferred to develop the software on the server as it provides easy configuration and management. The server has an API role between the performing unit and the mobile application. There are two main tasks handled by the server; tracking the amount of food and water and the authentication of the mobile application. The amount track is composed of communicating with the performing unit and receiving/requesting the current amount of food and water in the containers. The server maintains a web server software that keeps track of the requests coming from the mobile application and authentication related data between performing unit and the mobile application.

Mobile Application

The main purpose of the mobile application is to provide a GUI for the IoT. The mobile application is developed for the Android platform. The main functions are showing the amount of food and water for the last ten hours and refilling the food and water. There are two line graphs showing the food and water amounts on the y-axis and time on the x-axis. The previously specified minimum amount of food and water are also shown in these graphs as flat lines. The function of refilling food and water is

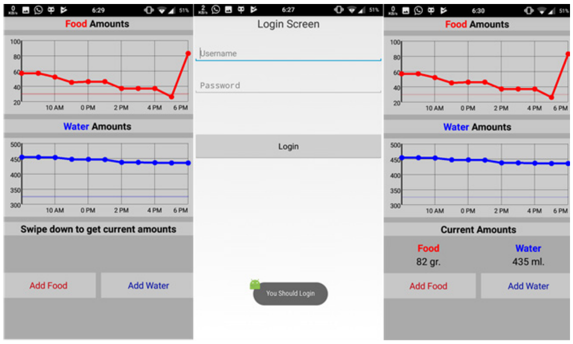

Figure 5. Mobile Application main screens. The graphs are for monitoring the current amounts of food and water over a pre-defined time period

accessed through two buttons. Once the user wishes to perform refill operation, the application asks for the confirmation. The user needs to swipe down the screen while running the application to obtain the current amounts. The screens of the mobile applications are given in Fig. 5. The demonstration video of our IoT prototype can be seen at https://goo.gl/4z59Aq

CONCLUSION

The Internet has been continuously changing our daily life over two decades. Recently, not only computers but also mobile platforms (e.g., phones, tablets etc.) make internet access possible. Lately, the concept of the Internet of Things (IoT) has been developing rapidly. It aims to create smart tools by connecting everything to the internet. As they make life easier for humans, the need for them has been increasing and the new application areas have been expanding day and day. One of these emerging areas is home automation and/or monitoring in the context of smart homes.

Pets are usually left unattended during a day-time of working days. They are also left unattended when their owners are in travel. Therefore, it would be useful to have a system to feed them remotely. In this study, from top to toe, a low-cost IoT system is presented for pet owners to monitor food and water supply and feed their pets remotely. The system is composed of performing unit, server, and mobile application. The performing unit is mainly for controlling the amounts of food and water and refilling. The server plays a bridge role between performing unit and mobile application with suitably arranged security measures. The mobile application is a GUI of the whole system. It provides the current amounts of food and water and allows the user to perform a refill operation. All software parts of the subsystems are developed by using open-source programming environments with relevant security measures and its sensor suite was designed in a minimalistic way.

REFERENCES

1. http://www.faz.net/aktuell/wirtschaft/diginomics/grosse-internationale-allianz-gegen-cyber-attacken-15451953-p2.html?printPagedArticle=true#pageIndex_1, Retrieved on May 21, 2018
2. Nordrum, Amy (18 August 2016). "Popular Internet of Things Forecast of 50 Billion Devices by 2020 Is Outdated". IEEE.
3. Pavithra, D., & Balakrishnan, R. (2015, April). IoT based monitoring and control system for home automation. In Communication Technologies (GCCT), 2015 Global Conference on (pp. 169-173). IEEE.
4. Hassanalieragh, M., Page, A., Soyata, T., Sharma, G., Aktas, M., Mateos, G., ... & Andreescu, S. (2015, June). Health monitoring and management using Internet-of-Things (IoT) sensing with cloud-based processing: Opportunities and challenges. In Services Computing (SCC), 2015 IEEE International Conference on (pp. 285-292). IEEE.
5. Kirci, P. and Kurt, G., Long Term and Remote Health Monitoring with Smartphone, International Journal of Intelligent Systems and Applications in Engineering, Vol.4, No. 4, 2016
6. Kastner, W., Kofler, M., Jung, M., Gridling, G., & Weidinger, J. (2014, September). Building Automation Systems Integration into the Internet of Things The IoT6 approach, its realization and validation. In Emerging Technology and Factory Automation (ETFA), 2014 IEEE (pp. 1-9). IEEE.
7. Da Xu, L., He, W., & Li, S. (2014). Internet of things in industries: A survey. IEEE Transactions on industrial informatics, 10(4), 2233-2243.
8. Zanella, A., Bui, N., Castellani, A., Vangelista, L., & Zorzi, M. (2014). Internet of things for smart cities. IEEE Internet of Things journal, 1(1), 22-32.
9. Manoj M, "Automatic Pet Feeder", International Journal of Advances in Science Engineering and Technology, ISSN: 2321-9009, Volume-3, Issue-3, Special Issue Sept.-2015 http://www.iraj.in/journal/journal_file/journal_pdf/6-185-14414356149-11.pdf
10. Own, C. M., Shin, H. Y., & Teng, C. Y. (2013). The study and application of the IOT in pet systems. Advances in Internet of Things, 3(01),
11. Healty Pet Simply Feed Programmable Digital Pet Feeder, Petsafe, https://store.intl.petsafe.net/en-gb/healthy-pet-simply-feed-programmable-digital-pet-feeder, Retrieved on May 26, 2018
12. Smart Pet Feeder, Petnet, https://petnet.io/smartfeeder, Retrieved on May 26, 2018
13. Overview of the Internet of things,. International Telecomunications Union. http://handle.itu.int/11.1002/1000/11559, Retrieved on May 26, 2018.
14. Whitmore, A., Agarwal, A., & Da Xu, L. (2015). The Internet of Things—A survey of topics and trends. Information Systems Frontiers, 17(2), 261-274.
15. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future generation computer systems, 29(7), 1645-1660.
16. 16. Govindraj, V., Sathiyanarayanan, M., & Abubakar, B. (2017, August). Customary homes to smart homes using Internet of Things (IoT) and mobile application. In Smart Technologies For Smart Nation (SmartTechCon), 2017 International Conference On (pp. 1059-1063). IEEE.
17. Wunderlich, S., Cabrera, J. A., Fitzek, F. H., & Reisslein, M. (2017). Network coding in heterogeneous multicore IoT nodes with DAG scheduling of parallel matrix block operations. IEEE Internet of Things Journal, 4(4), 917-933.
18. https://djangopackages.org/packages/p/django_cron/, Retrieved on August 21, 2017