Fuzzy Logic-based IoT enabled thermostat

Candale Andrei
Department of Computer Science
Babeş-Bolyai University
1 M. Kogalniceanu Street, 400084, Cluj-Napoca, Romania
Email: caie1378@scs.ubbcluj.ro

***Abstract*—This paper presents an accessible approach to an efficient, intelligent, Internet of Things (IoT) enabled thermostat to be used, instead of the usual, naive thermostats that are currently in a reasonable price range on the market. The current means by which most of the population control their house temperature is to use a simple, straight-forward thermostat that is simply reactive to only the temperature it records and the temperature set point of the user. The anticipated outcome of this approach is to create a new kind of thermostat that is first of all affordable, making the most of the new technologies we have, accessible through the infrastructure which the Internet provides and intelligent, giving the user an improved experience and greater energy management control.**

I. Introduction

The evolution of modern technologies has lead to an increased number of Do-It-Yourself (DIY) possiblities in the field of electronics and microchips; thus it has become easy and affordable to incorporate such technologies in mundane, house-hold objects that can present themselves with some data that may be relevant. This new concept turned into a movement now called the Internet of Things (IoT).

We use the term Internet of Things to refer to the general idea of things, especially everyday objects, that are readable, recognizable, locatable, addressable, and/or controllable via the Internet, wireless LAN, wide-area network, or other means. Everyday objects includes not only the electronic devices we encounter everyday, and not only the products of higher technological development such as vehicles and equipment, but things that we do not ordinarily think of as electronic at all, such as food, clothing, and shelter; materials, arts, and subassemblies; commodities and luxury items;landmarks, boundaries, and monuments; and all the miscellany of commerce and culture. [1]

In order for this concept to grow, technologies had to be made available to the general public, meaning that the cost at which they came had to be affordable. Fortunately, the current technological context lead to the development of cheap and powerful components which can now be used to build the infrastructure of the IoT.

With the now given possibility to acquire powerful, yet cheap, means by which one can control and gather data from different contexts, various applications can be developed, like the one described by this paper. The chosen hardware for the proposed thermostat consist of 3 microcontrollers, temperature and humidity sensors and radio transceiver, all coming to a price comparable to a normal thermostat. The main processing unit is an ESP8266 wireless module. An in depth description of the architecture will be presented in section IV.

This paper is organized as follows: Section II describes the motivation to build such a thermostat, Section III presents information relevant for the understanding of the chapter that follows, Section IV explains the proposed approach, Section V outlines the hardware architecture and human interaction and Section VI presents the conclusion that were drawn.

II. Motivation

The motivation behind this proposal is three-fold: affordability, efficiency and connectivity.

The time we are living in presents itself with rapidly occurring technological breakthroughs that help human kind in setting up for themselves a better, more comfortable and efficient world. The greater problem concerning this is that it does not come cheap. Advanced electronics were not affordable to the general public so the vast majority of people were unable to get access to them, let alone develop "things" having them as a platform. This situation has changed in the past few years and now we have easy access to such devices enabling the contribution of everybody to the Internet of Things.

Having access to diverse pieces of technology lets anybody create mechanisms through which data is gathered from the environment, analysed and used to make existing processes more efficient. The Internet of Things enables the world to research itself and find patterns that were otherwise invisible, thus leading to a greater understanding of the context we are living in, which in turn lets us control that context in a more efficient manner.

All these connected devices means that data can be shared and used in a collaborative manner and the possibility to control processes from afar arises. For the application in discussion, this fact enables one to monitor and control one's house temperature setting from anywhere thus increasing the comfort level. For example, going away from home, one would turn the temperature down but when coming back, it is desirable to have the house already at a comfortable temperature level. This can be achieved by being connected to your home thermostat.

III. Theoretical Background

As stated before, the proposed thermostat is using fuzzy logic for the decision engine because to model the problem using mathematics would be much to complex for a simple microcontroller to handle and specific data about each environment would have to be known.

Fuzzy logic is an extension of Boolean logic coined by Lotfi Zadef. Fuzzy logic is build on the theory of fuzzy sets which is a generalization of the classical set theory. He introduced the notion of degree in the verification of a condition such that it could be in a state different from true or false. This provides a valuable extended flexibility for reasoning which makes it possible to take into account inaccurate data. One advantage of fuzzy logic is that the rules by which the reasoning is made can be expressed in natural language making it easy to take advantage of a human expert and the knowledge he/she posses.

A. *Definitions*

In the rest of the chapter, the concepts surrounding fuzzy logic will be explained.

1) Fuzzy sets: A fuzzy set is a class of objects with a continuum of grades of membership[2].

Definition 3.1: Let X be a set. A $fuzzy$ subset A of X is characterised by a **membership function**

$$f^a : X \to [0, 1] \tag{1}$$

Note: This membership function is equivalent to the identity function of a classical set.

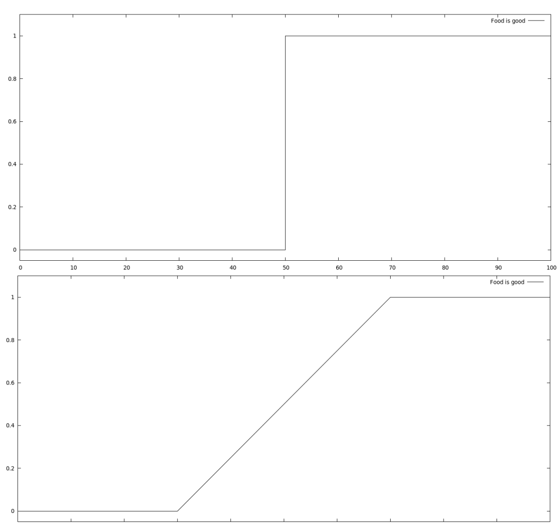

Fig. 1. Comparison between a identity function of a conventional set and a membership function of fuzzy set.

2) Linguistic variables and values:

Definition 3.2: Let V be a variable (temperature, humidity etc), X the range of values of the variable and T_V a finite or infinite set of fuzzy sets. A **linguistic variable** corresponds to the triplet (V, X, T_V).

The concept of membership function discussed above allows us to define fuzzy systems in a natural language, thus we can label different ranges of values for an input. For example, having a linguistic variable called “music” we have the linguistic value “silent”, “moderate” and “loud”. The membership functions for these values may describe a trapezium, a triangle, or anything else depending on the problem in question.

3) Fuzzy operators: In order to properly manipulate fuzzy sets, the operators from classical set theory are redefined to fit the membership function values that are in rage $[0, 1]$. These operators can be chosen, having to kinds: Zadeh or probabilistic operators. Zadeh operators are in the folowing form:

$$AND = min(\mu_A(x),\mu_B(x)) \tag{2}$$

$$OR = min(\mu_A(x),\mu_B(x)) \tag{3}$$

$$NOT = 1 - \mu_A(x) \tag{4}$$

while probabilistic operators:

$$AND = \mu_A(x) \times \mu_B(x) \tag{5}$$

$$OR = \mu_A(x) + \mu_B(x) - \mu_A(x) \times \mu_B(x) \tag{6}$$

$$NOT = 1 - \mu_A(x) \tag{7}$$

4) Fuzzy logic reasoning: In fuzzy logic, reasoning is based on **fuzzy rules** that are expressed in natural language using the above described linguistic variables. A fuzzy rules has the form:

$$ifx \in A \quad and \quad y \in B \quad or \quad z \in C \quad then \quad d \in U \tag{8}$$

For example:

“if food is good and treatment is nice then tip is great”.

After computing the degree of membership of the linguistic variables “food” and “treatment” to their respective linguistic values from the antecedent, “good” and “nice”, the binary fuzzy logic operator “and” will be applied which will get the minimum of the two degrees (Figure 2). The variable “tip” belongs to the linguistic value “great” to the degree of validity of the premise.

The next step in arriving at a result is to determine the fuzzy implication of the antecedent. There are several fuzzy implication patterns out there but the most commonly known are Mamdani fuzzy implication and Larsen fuzzy implication. Mamdani fuzzy implication takes the minimum between the resulting antecedent membership function and the consequent membership function

$$Mamdani \quad min(f_a, f_b) \tag{9}$$

while Larse fuzzy implication model takes the product of the two

$$Larsen \quad f_a \times f_b \tag{10}$$

The result will be the membership function of the consequent or a clipped version of it.

The case can present itself when we have multiple rules for which the consequent is identical.

For example:

“if food is bad and treatment is bad then tip is low”

and

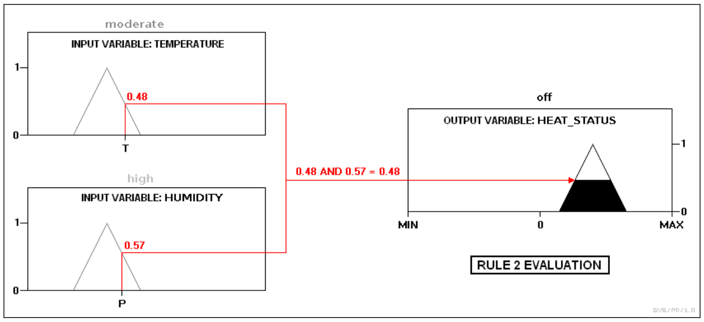

Fig. 2. Rule evaluation result.

"if temperature is good and treatment is bad then tip is low"

In such a case, the value that is chosen is the one for which the resulting membership function is the greatest. For example if the resulting membership function for the first rule is 0.45 and the resulting membership function for the second rule is 0.7, for the linguistic value "off" of variable heating_status will be 0.7.

After having the membership functions of all the consequents, a final result can be achieved by defuzzifying them. This is done by finding the abscissa of the center of gravity corresponding to the polygon described by the membership functions of all the output linguistic variables.

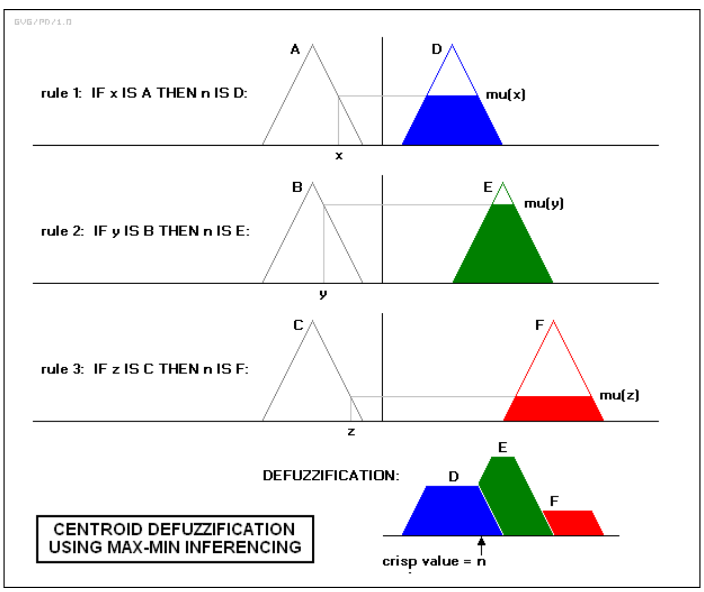

Fig. 3. Resulting polygon. Image source [3]

IV. Architecture

This section will present an in-depth look at the architecture used for the thermostat: each of them individually and then as an assembly Pursuing the first key of the motivation for this approach, the chosen microcontrollers for this application are in the affordable category, but nonetheless powerful enough to carry out the task.

The first microcontroller used is the ESP8266 wireless module created by Espressif (Figure 4).

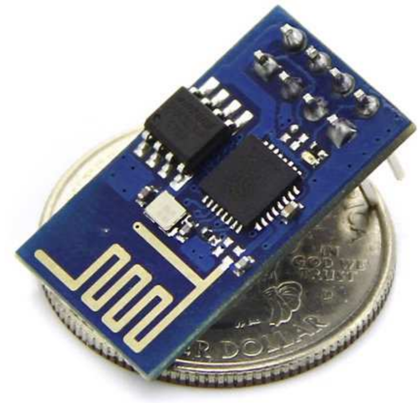

Fig. 4. ESP8266 Wireless Module. Image source: [4].

ESP8266 is a highly integrated chip designed for the needs of a new connected world. It offers a complete and self-contained WiFi networking solution, allowing it to either host the application or to offload all WiFi networking functions from another application processor.[5] The microcontroller is build onto ARM platform, it has 200 kB of ROM, 32 kB of SRAM and 80 kB of DRAM. Given its relatively great storage capacity and RAM, the ESP8266 is the main processing unit of the thermostat where the fuzzy logic algorithm will run. This module also provides, as the name suggests, the connectivity capability. It supports 802.11 b,g and n standards and has TCP and UDP stack implemented on it.

Since its release, the open-source community has developed a number of programming languages support, including lua[6] and javascript[7] but, while testing, it has been found that currently the C SDK that the producer has released is the most stable and it is the one used.

The SDK comes with a documented API that exposes low-level functions that perform basic tasks like initializing a WiFi Access Point (AP), making a Http request and creating a server that listens and fetches the requests that the module receives.

The second microcontroller used is the Atmel Attiny85 (Figure 5). [8]

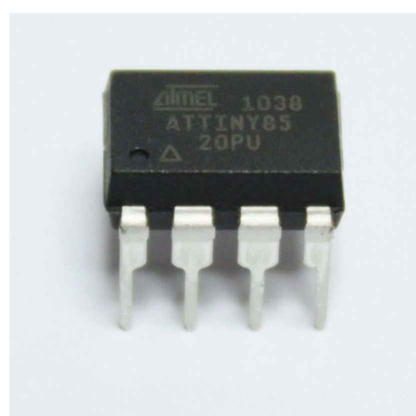

Fig. 5. ESP8266 Wireless Module. Image source: [4].

This is a small AVR processor with 8 kB of ISP flash memory, 512-Byte SRAM and 6 general purpose I/O lines. It can be programmed using an Arduino and the Arduino IDE by writing C or C++ code [9].

The thermostat setup consists of at least two modules: a main module, that lies near the thermal heating unit and one or more smaller modules that can be spread around the environment. The former records the temperature and sends it to the main module to be processed.

The main module (Figure 6) is made up of the ESP8266 wireless module, an Attiny85 and a 433 MHz radio receiver

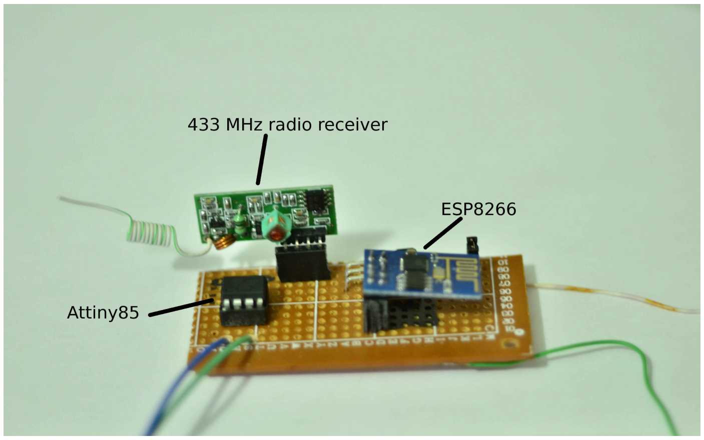

Fig. 6. Main module prototype.

and the second module (Figure 7) (that we will call reporter) is made up of a temperature and humidity sensor (DHT22), an Attiny85 and a 433 MHz radio transmitter. This will run on battery.

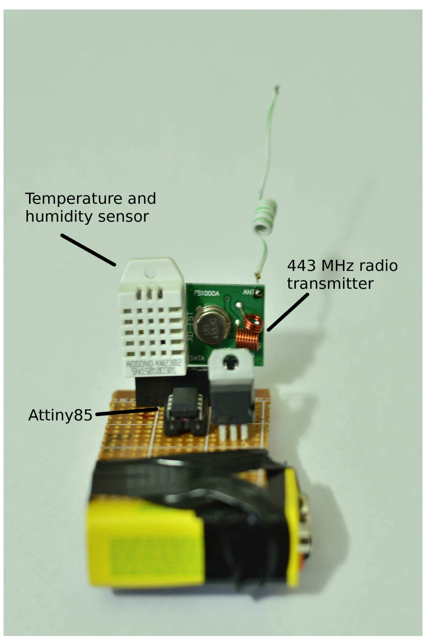

Fig. 7. Second module prototype.

The way it works is that that the reporter reads the temperature form the sensor and sends it over the 433 MHz radio. This happens at an established interval such that the battery will last as long as possible and the data will still be relevant for heating control. The Attiny85 on the main module reads the data from the 433 HMz radio receiver and stores it until the ESP8266 sends a request over serial communication and asks for the readings. The data is then processed and a decision is taken (turn the heating on or off) that is communicated by the ESP8266 to the Attiny85. Besides the above mentioned components, the main module will also hold a relay that is controlled by the AVR microcontroller in response to the decision that was taken. The radio communication between the modules is done using the Manchester encoding library [10].

The user may interact with the thermostat in a couple of ways. At first start-up, the WiFi module creates an AP (Access Point) and a webserver that listens to the default Http port 80 on IP address 192.168.4.1. A small, lightweight webserver framework was designed in order to facilitate the processing of requests. If the page /wifi_setup is accessed, the user will be presented with a form where he should enter the SSID of his home wireless access point and the password with which the module can connect to it. After doing that, in approximately twenty seconds the module will connect to the wireless router and the AP it created will disappear. The ESP8266 will remember the SSID and password so the next time it starts-up, the same steps won't be necessary. After having connected and the AP disappeared, the ESP8266 will make use of its mDNS feature to publish itself on the intranet he just gained access to. After that, the user, while being connected to his home router, will be able to access the thermostat by going to http://thermostat.local.

If the thermostat has internet access while connected to the wireless router, it will publish its reading of temperature, humidity and computed values (rate of heating, rate of cooling) to a remote server.

V. Proposed approach

The proposed thermostat uses a Mamdani fuzzy logic engine to reach a conclusion, based on 4 inputs, about whether or not is the heating supposed to be on. There are a couple of factors and the relationship amongst them that are taken into consideration when trying to take this decision. First of all is the difference between the temperature that the user desires and the temperature that the sensors are currently reading. The larger this difference, the greater the time is in which the heating should be on. Also, one might set a higher than needed point in order to get to a desired temperature in a shorter period of time which means that the time the heating should be on is greater. In contrast, multiple, smaller changes brought to the target temperature, by the user, might mean that the desire is to achieve a more accurate temperature control.

The second factor is the relation between the temperature and the relative humidity. The higher the relative humidity is, the greater the felt temperature will be. Taken this into consideration, rules are constructed in such a way that heating control is "aware" that comfort level in high humidity is low.

Another factor taken into consideration is the rate at which the sensed environment heats up or cools down. This information helps take pre-emptive measures or avoid heating a room in which the temperature is steadily rising.

5) *FuzzyEngine:* Having described the factors that are taken into consideration, these are the inputs that the fuzzy engine takes:

- Input 1: temperature error; the difference between the set temperature and the actual temperature of the room
- Input 2: humidity; the relative humidity from the environment
- Input 3: rate of heating; temperature trend in a given time frame; more on this below
- Input 4: rate of cooling; similar to rate of heating
- Output: heating status; this is an output fuzzy set and represents the decision at which the engine has arrived based on all the inputs described above

The inputs temperature and humidity are read from the sensors but rate of cooling and heating has to be calculated in order to get the trend of the temperature. This was done by storing and analysing data from a certain time interval in the past. Several approaches were considered for getting a useful value from the data points:

- the difference between the last and first data point from the dataset

$$Rateofcooling = t_m - t_1 \tag{11}$$

- computing the average temperature difference at every measurement

$$Rateofcooling = \frac{\sum_{i=1}^{m}(t_{i+1} - t_i)}{(m-1)} \tag{12}$$

- using least squares regression line method to approximate a function that best describes the dataset and use that function's slope as input for the engine

$$Rateofcooling = \frac{\sum_{i=1}^{m}(x_i - \overline{X})(y_i - \overline{Y})}{\sum_{i=1}^{m}(x_i - \overline{X})^2} \tag{13}$$

where

$$\overline{X} = \frac{\sum_{i=1}^{m} x_i}{m} \tag{14}$$

and

$$\overline{Y} = \frac{\sum_{i=1}^{m} x_i}{m} \tag{15}$$

Experiments (Figure 8) showed that the last method, least squares, proved to be the most steady and relevant with the fluctuations of temperature. Even with extreme differences, the values drop back to normal in a fair amount of time.

All the linguistic variables described above have associated with them three linguistic values: '"low", "moderate" and "high". The engine uses a combination of triangular and trapezium membership functions (Figure 9).

The fuzzy engine is triggered at an established interval of time, usually a minute, and the decision is sent over serial communication to the on-board Attiny85 which controls the relay.

Fig. 8. Values of slope with regard to temperature change.

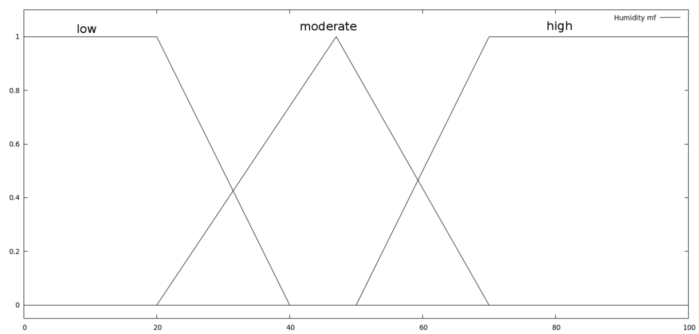

Fig. 9. Humidity membership functions.

VI. Results and discussions

Experiments have shown that the chosen hardware and software setup is well fit to accomplish the task of running a fuzzy engine and handle the communication with the user and with itself. The main concern that was present while developing was if the 32 kB memory of the ESP8266 will be enough to run the fuzzy engine and the first encountered challenge was the proper memory management and avoidance of memory leaks. While testing it was found that the "infrastructure" code, which is responsible for the interaction with the user, is consuming about 4 kB of SRAM. The fuzzy engine is initialized at system start-up and it won't be freed from memory since it consumes about 10 kB of code. When running, the memory overhead is of approximately 3 kB.

Testing was done to see how the ESP8266 behaves in time. While testing the responsiveness, a loss of 14% of the packets sent was recorded. The module behaves well except the times when a CPU intensive activity is going on like running the fuzzy engine or publishing the data on the remote server.

In the event of a system crash, the ESP8266 recovers immediately and resumes normal operation, including connection to wireless router and mDNS publishing.

Experiments to test the fuzzy engine's output and how it manages heating are yet to be performed.

VII. Conclusion

As seen from the experiments, the proposed solution is a promising one, offering enough computational power to run a fairly complex algorithm, sufficiently advanced hardware to accomplish the needs for a IoT enabled thermostat and cheap enough to be in a price range that can surpass the current, main-stream thermostats.

References

[1] "Six technologies with potential impacts on us interests out to 2025," in *Disruptive Civil Technologies*. National Intelligence Council, 2008.

[2] L. A. Zadeh, "Fuzzy sets," *Fuzzy Logic*, 1965.

[3] Defuzzification, "http://commons.wikimedia.org/."

[4] ESP8266, "http://is.gd/jiv6jk."

[5] ——, "http://espressif.com/en/products/esp8266/."

[6] NodeMCU, "http://www.nodemcu.com/."

[7] Espruino, "http://www.espruino.com/."

[8] Attiny85, "http://is.gd/wv8aia."

[9] H. to program an Attiny85, "http://highlowtech.org/?p=1695."

[10] Manchester, "http://mchr3k.github.io/arduino-libs-manchester/."