IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

ASSEMBLY OF AN IOT BASED ALARM CLOCK

[1]Fazal Rahim Naeemkhel, [2]Mr. Dushynat Singh, [3]Mr. Sitaram Gupta

[1]Research Scholar, [2]Assistant Professor, [3]HoD of CS department

[1]Department of Computer Science & Engineering

[1]Vivekananda Global University, Jaipur, India

Abstract: This smart alarm clock assembly project provides an easy to build Internet of Things alarm clock. It is designed, such that you can adapt your needs and improve it easily. The smart alarm clock project is based on the Raspberry Pi Zero. A WiFi stick soldered to the system provides easy access to the server running on the Pi. Assembling your alarm clock and setting the alarm time makes fun and is possible through each and every computers browser. The case is 3d-printable and has some dedicated slot to insert a tactile switch to. Therefore the whole setup is controllable by just one button.

***Index Terms* - Smart Alarm Clock, Assembly, Internet of Things, Raspberry Pi, Wifi Stick, 3d Printing.**

I. INTRODUCTION

Alarm clocks have been being used for a considerable length of time since they take care of a genuine issue guaranteeing we wake up on time. Their utility is obvious, and a considerable lot of us would most likely experience serious difficulties making it to work each day without one. This has to be one of the smoothest ways of waking up. The alarm triggers a light that will slowly get brighter and trigger hormones in your body, getting you out of bed without a jolt. There's always a backup alarm, in case you worry about waking up just with a light, it even turns ON the heater beforehand and switches OFF the fan corresponding to the alarm time, Even though most people have turned to their cell phones to act as their main alarm clocks, in our perspective there is still a whole room available for a smart alarm. The trouble is that you need something that has quite a few features and that will perform well compared to a traditional alarm, alongside the issue of space. You don't require a whole room for this smart setup. This makes user to save their time and make them get up from the bed in a energetic way, and our pre-defined mathematical calculations triggers a boost in the users brain and make the user active throughout the day. Individuals have been utilizing cautions of sorts to wake up a long time before timekeepers were even a thing. The morning sun was the first prompt to rise and sparkle, and chickens have likely been crowing sleepers wakeful for ages, many said to have woken up early with a clever device that used draining water to signal a whistle as the contraption filled, supposedly spurring the development of later mechanical timekeepers. Church bells have awoken communities for morning service for centuries, and by the 1300s, we can find descriptions of chiming clock towers in Europe designed to inform residents of the time.

Later, booming industrial factories in the 18th century relied on an on-time workforce, and would sound morning whistles to rouse their workers living nearby. Some cities and companies even employed a "knocker- up", designated people that would go door to door delivering wake up calls. Albeit existing as right on time as the 1500s, it wasn't until the point that the 1870s that wake up timers began to end up a typical thing in private homes with the coming of the mechanical breeze up variant. After some time, this apparatus established its situation in the room, joining extra highlights like radio 3 collectors, cuckoos, rest catches and that's just the beginning. Present day wake up timers keep on advancing, including light-based alerts to timekeepers that keep running over the room, and even telephone applications. With employments, school and different duties, awakening by a set time remains an unquestionable requirement for the lion's share of us. So normally,
numerous individuals rely upon wake up timers to begin the day. Be that as it may, would they say they are truly helping us out, or is this dependence on wake up timers unfavorable?

II. LITRATURE REVIEW

Thorat, S. S., Ashwini, M., Kelshikar, A., Londhe, S., & Choudhary, M. (2017) in their study stated that with the exponential increase in the number of vehicles and world population day by day, vehicle availability and usage on the road in recent years, finding a space for parking the bike is becoming more and more difficult with resulting in the number of conflicts such as traffic problems. This is about creating a reliable system that takes over the task of identifying free slots in a parking area and keeping the record of vehicles parked very systematic manner. This project lessens human effort at the parking area to a great extent such as in case of searching of free slots by the driver and calculating the payment for each vehicle using parking area. The various steps involved in this operation are vehicle identification using RFID tags, free slot detection using IR sensors and payment calculation is done on the basis of period of parking and this is done with the help of real time clock.

Scott, G., & Chin, J. (2013, September) wrote a paper which explored how the recent development of low- cost System on a Chip (SoC) boards can be used by the Internet of Things (IoT) DIY community to assist the process of smart object innovation through the example of an intelligent alarm clock. The alarm clock will combine existing traffic and weather web services with local temperature sensor readings to provide a suitable alarm time for the user. Included is a brief review of currently available IoT components and state-of-the-art alarm clocks offering augmented features. Provided is a description and justification of both the hardware and software components of the alarm clock. CPU and memory resource testing demonstrate the computational suitability of the SoC device in the context of an intelligent alarm clock. User feedback regarding the features of the alarm clock provides suggestions for further development.

Celia, L., & Cungang, Y. (2018, July) in their study stated that the Internet of Things (IoT) provides transparent and seamless incorporation of heterogeneous and different end systems. It has been widely used in many applications such as smart homes. However, people may resist the IOT as long as there is no public confidence that it will not cause any serious threats to their privacy. Effective secure key management for things authentication is the prerequisite of security operations. In this paper, we present an interactive key management protocol and a non-interactive key management protocol to minimize the communication cost of the things. The security analysis show that the proposed schemes are resilient to various types of attacks.

Kodali, R. K., & Sahu, A. (2016, December) in their study states that the Internet of Things (IOT) describes the interconnection of devices and people through the traditional internet and social networks for various day- to-day applications like weather monitoring, healthcare systems, smart cities, irrigation field, and smart lifestyle. IOT is the new revolution of today's internet world which monitors live streaming of the entire world's status like temperature, humidity, thunderstorm, earthquake, floods etc. that can stagger an alarm to human life. This paper proposes a low-cost weather monitoring system which retrieves the weather condition of any location from the cloud database management system and shows the output on an OLED display. The proposed system uses an ESP8266-EX microcontroller based Wemos D1 board and it is implemented on Arduino platform which is used to retrieve the data from the cloud. The main objective of this paper is to view weather conditions of any location and allows to access the current data of any station.

Kanuparthi, A., Karri, R., & Addepalli, S. (2013, November) in their study stated that Internet of Things (IoT) is the interconnection of a large number of resource-constrained devices such as sensors, actuators, and nodes that generate large volumes of data which are then processed into useful actions in areas such as home and building automation, intelligent transportation and connected vehicles, industrial automation, smart healthcare, smart cities, and others. Important challenges remain to fulfill the IoT vision including data provenance and integrity, trust management, identity management, and privacy. We describe how embedded and hardware security approaches can be the basis to address these security challenges.

Hodges, S., Taylor, S., Villar, N., Scott, J., Bial, D., & Fischer, P. T. (2012) in their study stated that tools like Microsoft .NET Gadgeteer offer the ability to quickly prototype, test, and deploy connected devices, providing a key element that will accelerate our understanding of the challenges in realizing the Internet of Things vision. Gadgeteer is a general-purpose device development platform where the key elements includes rapid construction and reconfiguration of electronic device hardware, ease of programming and debugging, and the ability to leverage online Web services for additional storage, communication, and processing.

Hurlburt, G. F., Voas, J., & Miller, K. W. (2012) in their study stated that today, a new Internet player is rowing more important: things-that is, inanimate objects that can be programmed to communicate, sense, and interact with other things. But will an increasingly fragile ecosystem be able to sustain the amount of power necessary to run all these gadgets? And what other challenges must we overcome to realize a productive and reliable Internet of Things?

Restuccia, F., D'Oro, S., & Melodia, T. (2018) in their study stated that the Internet of Things (IoT) realizes a vision where billions of interconnected devices are deployed just about everywhere, from inside our bodies to the most remote areas of the globe. As the IoT will soon pervade every aspect of our lives and will be accessible from anywhere, addressing critical IoT security threats is now more important than ever. Traditional approaches where security is applied as an afterthought and as a "patch" against known attacks are insufficient. Indeed, next-generation IoT challenges will require a new secure-by-design vision, where threats are addressed proactively and IoT devices learn to dynamically adapt to different threats. To this end, machine learning (ML) and software-defined networking (SDN) will be key to provide both reconfigurability and intelligence to the IoT devices. In this paper, we first provide a taxonomy and survey the state of the art in IoT security research, and offer a roadmap of concrete research challenges related to the application of ML and SDN to address existing and next-generation IoT security threats.

Madakam, S. (2015) in her study stated that research study on the Internet of Things and Smart Things has been going on for more than a decade and reaches back to Mark Weiser's original dream of ubiquitous computing. Bruce Sterling recently popularized the idea of Smart Objects and the IoT. Smart Things is another paradigm shift in IT world. Smart Things are the things that are having embedding smartness or intelligence, identification, automation, monitoring and controlling calibre. Smart Things are assisting human life a lot, nowadays without their applications life is becoming cumbersome. This paper exhibits systematically on Internet, Things, and then explores on Internet of Things and finally Smart Things from researchers', and corporate's perspective. Moreover, this article focuses on the state of Smart Things and its applications. This in turn would help the new researchers, who want to do research in this IoT domain.

Yun, J., Ahn, I. Y., Choi, S. C., & Kim, J. (2016) in their study stated that the Internet of Things allows things in the world to be connected to each other and enables them to automate daily tasks without human intervention, eventually building smart spaces. This article demonstrates a prototype service based on the

Internet of Things, TTEO (Things Talk to Each Other). We present the full details on the system architecture and the software platforms for IoT servers and devices, called Mobius and &Cube, respectively, complying with the globally-applicable IoT standards, oneM2M, a unique identification scheme for a huge number of IoT devices, and service scenarios with an intuitive smartphone app. We hope that our approach will help developers and lead users for IoT devices and application services to establish an emerging IoT ecosystem, just like the ecosystem for smartphones and mobile applications.

III. ASSEMBLY

Raspberry Pi Zero Running Raspbian

In order to complete this project you need to set up a Raspberry Pi running Raspbian (Lite), the Debian version of Raspberry Pi. Sure you might use different Raspberry Pis and a different OS, but therefore you might need to modify your system somewhat. For example if you want to use Raspberry Pi 3 you don't have to buy and use an additional WiFi-stick, might just use the on-board audio output, but might not be able to use the given 3D-printable case. So by changing some parts of this project you got to keep in mind that this instruction is based on using Raspberry Pi Zero.

Display

This step contains the soldering and configuration of the "Alphanumeric Display". Again you might choose a different display. Doing so you have to apply small changes when it comes to coding as well as using the 3d- printable case. The display, we are using for this project, goes by the name "Adafruit 0.54" Quad Alphanumeric FeatherWing Display" and is provided by adafruit. This model comes together with the 14- Segment Alphanumeric LED FeatherWing containing the HT16K33 driver chip, which handles the multiplexing of the commands. Therefore we are able to control the display using the Raspberry Pi and I²C protocol. And of course adafruit also provide a ready-to-use library, which enables us of simply sending all kinds of strings to the display. But first lets care about the soldering.

Display Assembly

Following instructions given by ad fruit for information about the assembly of the display and the Feather Wing. First you need to solder some pins to the Feather Wing.

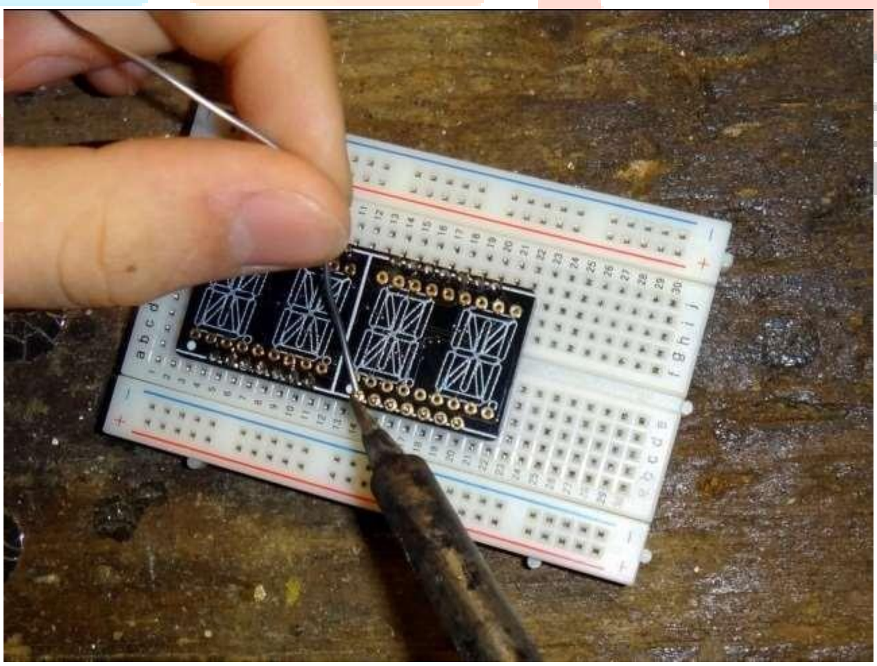

Figure 1: Soldering pins to the Feather Wing

Next you need to plug the display into the FeatherWing board and solder all the pins to the board. Again check the adafruit tutorial for details and orientation of the displays assembly.

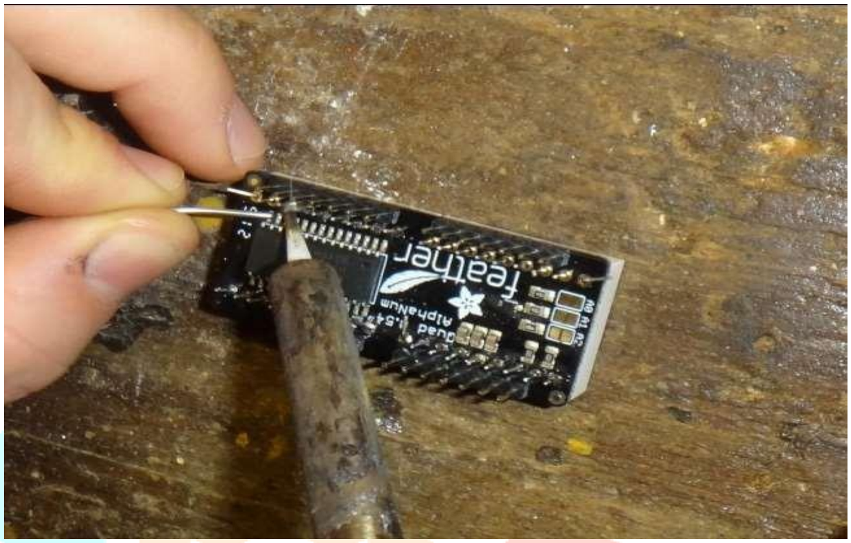

Figure 2: Plugin the display into the Feather Wing

In order to check the display you need to connect it to the Raspberry Pi. Therefore you might want to grab some wires, preferably multiple colors. But before connecting everything together, we recommend to first take the hole grid board and cut out a hole in the lower center. The hole grid board which we are using is 70mm times 50mm and fits perfectly into the provided case. Use the cordless mill to cut a hole into the board, like shown in the pictures below.

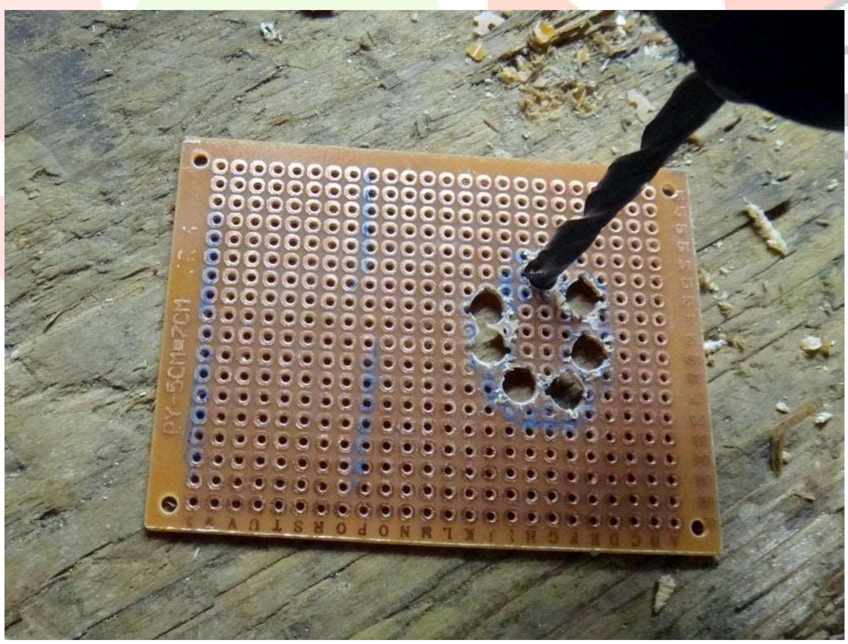

Figure 3: Cutting a hole into the board

If you already have printed the case you might take the chance to see if it fits. Put the speaker on the provided cut-out and the hole grid board on top of it. Since those boards are not the highest quality, the position of the four holes at the edges of the board may vary a bit. Don't worry if two of the holes are not fitting exactly to the corresponding screw holes, everything will be squeezed together in the end, such that the use of two screws will be perfectly fine.

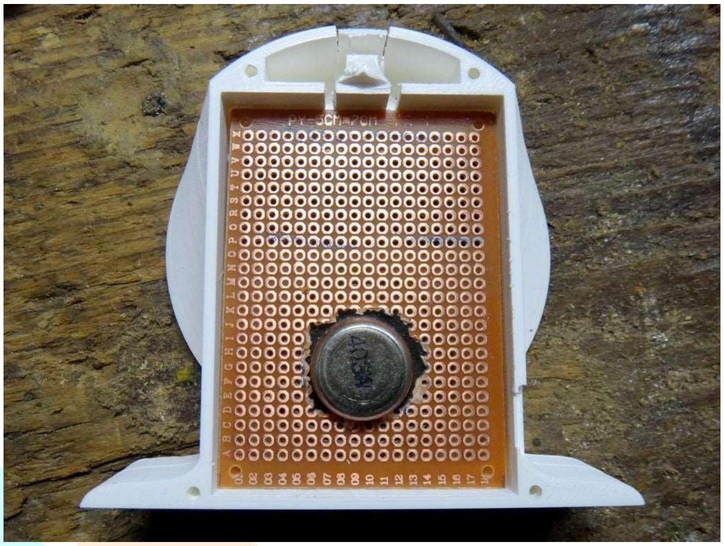

Figure 4: Putting the speaker on the provided cut-out

Once you are done preparing the hole grid board, soldering the display to the Pi is up next. Grab your colored wires and cut them to a length of about 10cm. Put the display centered into the top row of holes. Check the adafruit tutorials pinout page for details about the wiring. You need to connect four wires: 3.3V; GND and the two I²C pins: SDA and SCL. Consider the schematic diagram concerning the wiring of the display and the Raspberry Pi

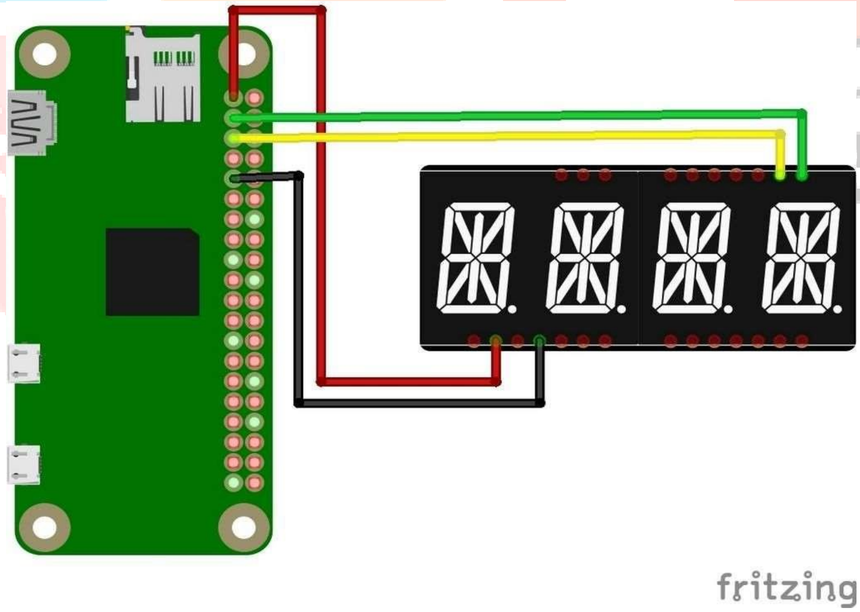

Figure 5: Wiring of the display and the Raspberry Pi

Having the display and the raspberry pi successfully soldered and mounted on the hole grid board, your setup should look something like the last picture from above.

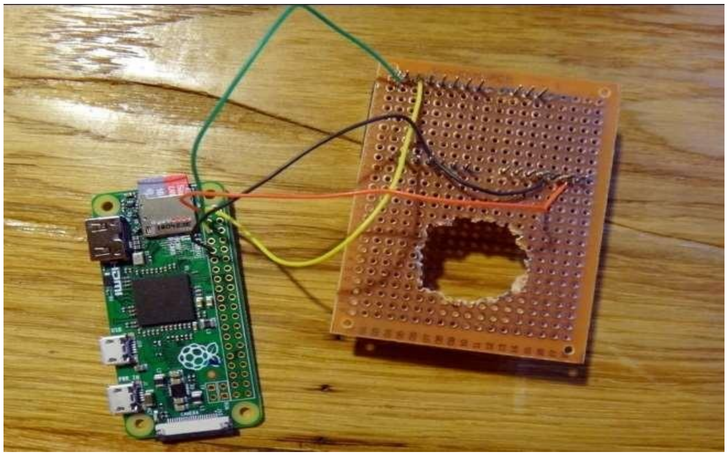

Figure 6: Display and the raspberry pi soldered and mounted on the hole grid board

Now continue installing the corresponding Adafruit library.

Audio

Using the alternate function of pins 13 and 18 enables the usage of pulse-width-modulation. Together with a small low-pass-filter circuit we are able to generate a basic audio output. Consider the following schematic diagram for the wiring of the circuit.

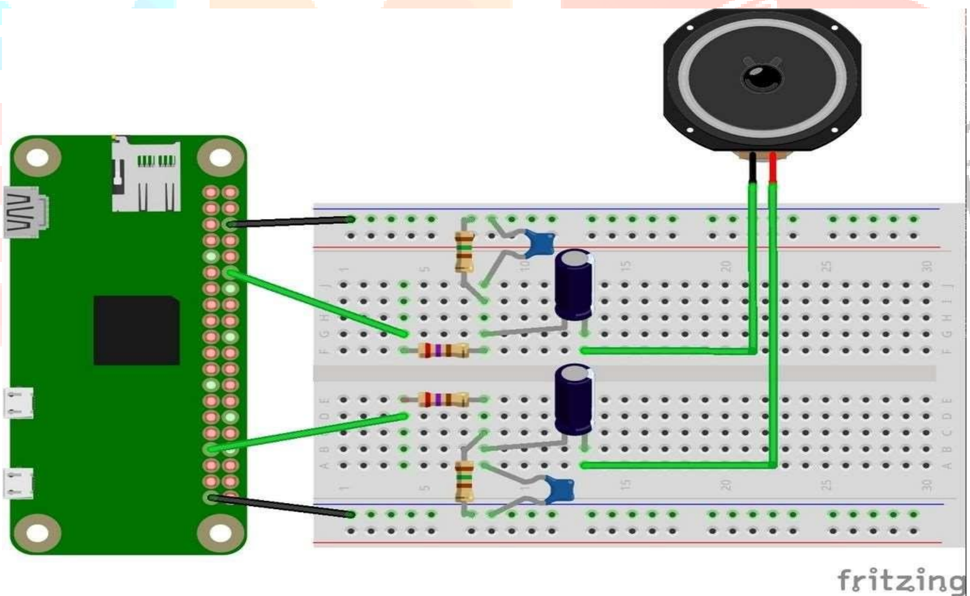

Figure 7: Wiring of the circuit

You'll need some resistors and capacitors to build the mentioned low-pass-filter. Of course you might just start soldering the circuit to your hole grid board, but we recommend to first us a breadboard to make sure you connect everything right and test the filter. Note that there are two low-pass-filter, each for one audio output, namely right and left. There is only a need to have one audio output for this project, since there is just one speaker. On the next couple of pictures you will recognize two low-pass-filters, though. This is just because we had some earlier plans of adding two speakers. So don't worry about it and choose what you need for your project. If you are going to build the alarm clock with just one speaker, one filter (=one audio output) is enough. The resistors you need for the low-pass-filter are 150 Ohm and 270 Ohm. The capacitors need to have

10nF and 1µF. Try to find some space on your hole grid board where the resistors and capacitors don't contact the speaker, like shown in the next picture:

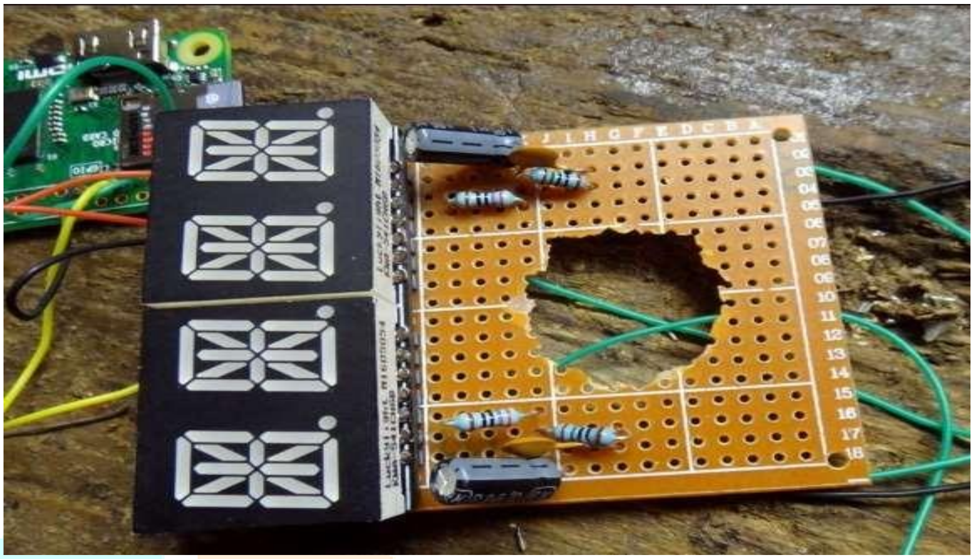

Figure 8: Resistors and capacitors

Solder the wires following the fritzing connection diagram above. Once you are done, your circuit should look something like the next picture. The green wires at the bottom are the one coming from Raspberry Pi Zero pins 13 and 18, the black wires at the top of the image are ground and coming from Pi as well. The green wires at the edge of the board are the audio output coming from the low-pass-filter and will be connected to the speaker and later on to the amplifier in order to increase the volume.

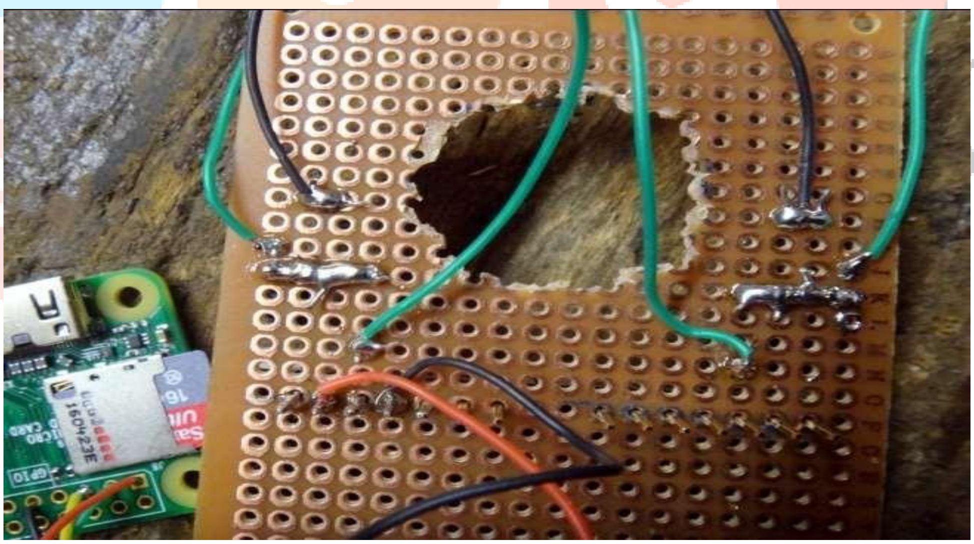

Figure 9: Green wires are the audio output

Again note, that there is no need for both low-pass-filter, controlling just one speaker requires only one filter.

Audio Amplifier

This section cares about increasing the volume of the system. Again we don't want to cause oversleeping. So adding an amplifier will solve this problem and give your project enough power to even use it like a small jukebox. We choose to use the Stereo Audio Amplifier PAM8403 Chip since it is cheap and easy to use.

Connecting the Amplifier

Consider the following fritzing schematic for wiring information:

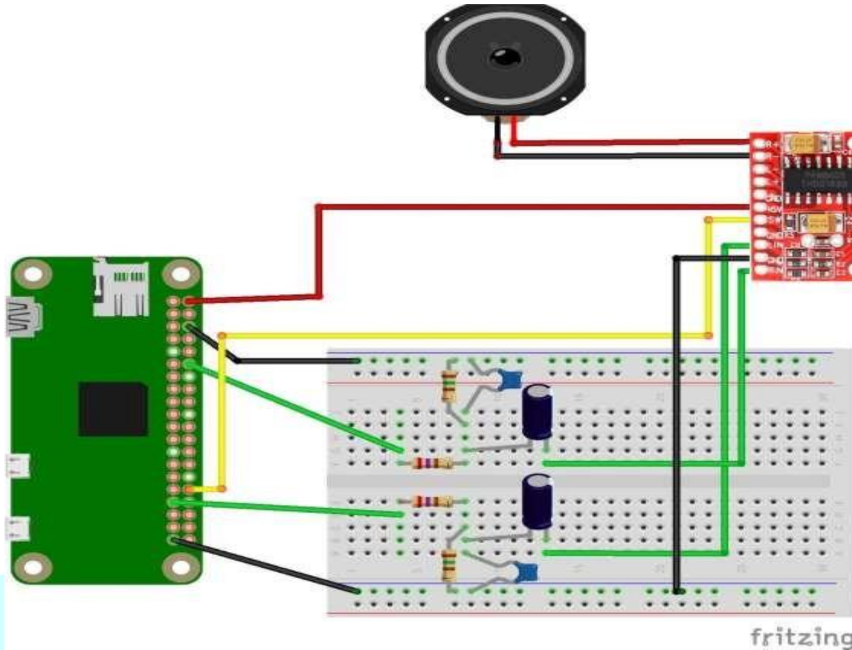

Figure 10: Connecting the Amplifier

As you can see we are using another Raspberry Pi pin for controlling the amplifier. The yellow wire coming from gpio 12 is connected to the amplifiers switch input pin. Therefore we need to add another line of code to all future python scripts (don't worry we've already done this!) which enables the amplifier by just sending a logical one for turn on and a zero for turn off vice versa. This appears to be handy since you might have noticed the noise produced by the Raspberry Pis pulse width modulation. Once you have connected everything your setup should look similar to the following one:

Figure 11: Module after connecting amplifier

Buttons

How to stop sounds? - Sure by pushing buttons! Once you woke up you need to stop the alarm. Therefore we need to add a button to our project.

Connecting the Button

All of you who have been working with buttons already might know about bouncing and debouncing. If you do not know about it, you can read some tutorials by Raspi.TV. If you don't want to read too much, skip the tutorials, grab your button and start wiring. Consider the following fritzing for the wiring:

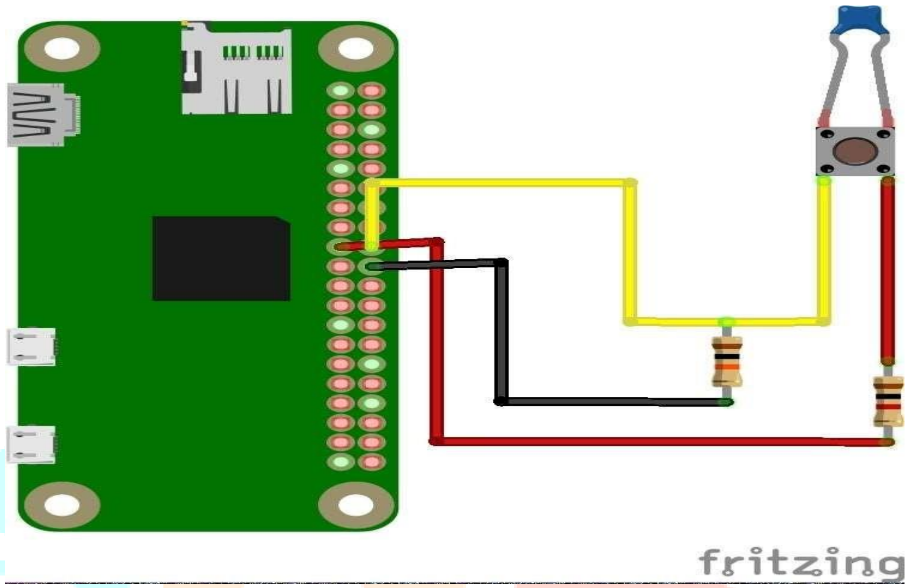

Figure 12: Connecting the button

You might choose some space at the bottom of your hole grid board to solder the connections to. Make sure not to get too close to the speaker. You need two resistors: 1x 1k Ohm and 1x 10k Ohm, one capacitor with 100nF and some wires. Connect everything like shown in the image above. The 1k Ohm resistor is connected between the two red wires, the 10k Ohm resistor is connected between the yellow and black wires. Your soldering on the hole grid board should like the following picture. On the front:

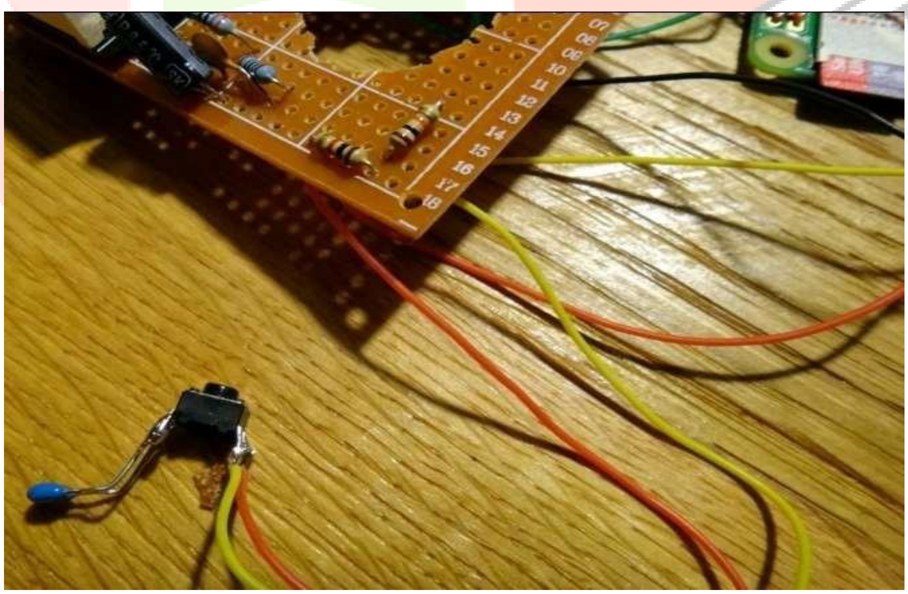

Figure 13: Soldering on the hole grid board (front)

And on the back:

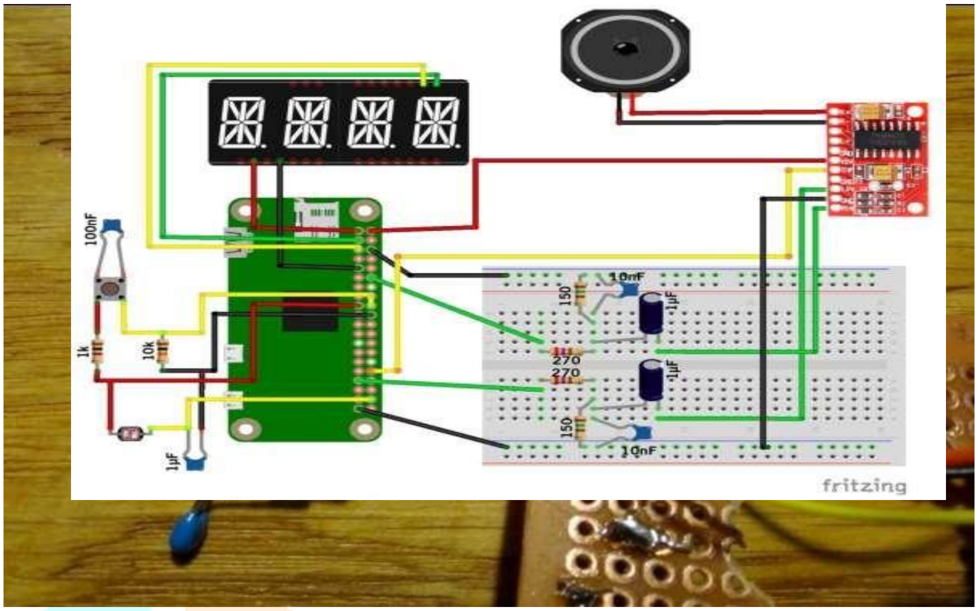

Figure 14: Soldering on the hole grid board (back)

If you got your case already printed you might want to check if the button fits into the provided slot like this:

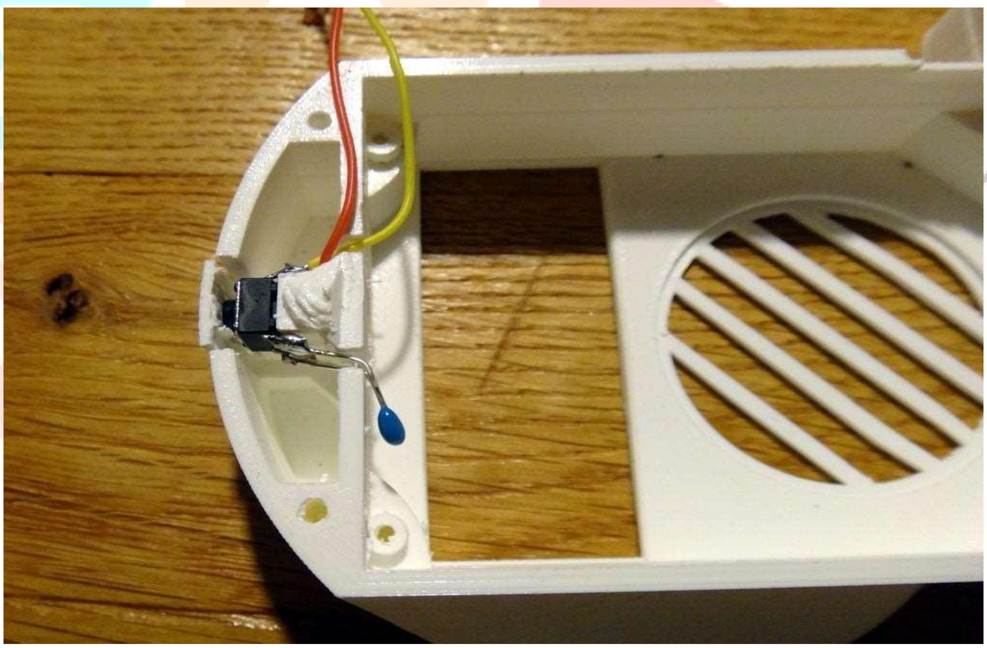

Figure 15: Fitting button in case slot

Photocell

Consider the following fritzing schematic for connecting the photocell to your setup. Note that this graphic is the complete and final wiring of the overall smart alarm project.

Since both photocell and button do not consume much power at all, connecting both to the same power supply pin and ground pin of the Raspberry Pi Zero is possible. You can do this by connecting the photocell wires to the soldering joint at the button on the lower edge of the hole grid board. Your connections should look similar to this one:

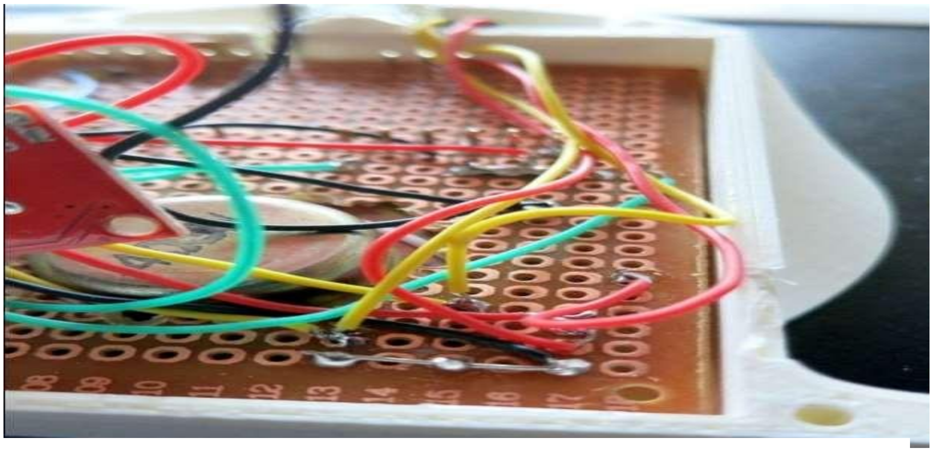

Figure 16: Connections after connecting the photocell

Make sure to use wires with a length of about 15cm in order to install the photocell in the top of the case. If the photocell does not fit into the provided hole you might use a sharp knife to slightly enlarge the holes. Take a pen (or some other thin tool) and press the photocell into the flat laying case. The wires can be arranged through the provided slit in the wall between top and body of the case. In the end your installation and wiring should look similar to the following picture:

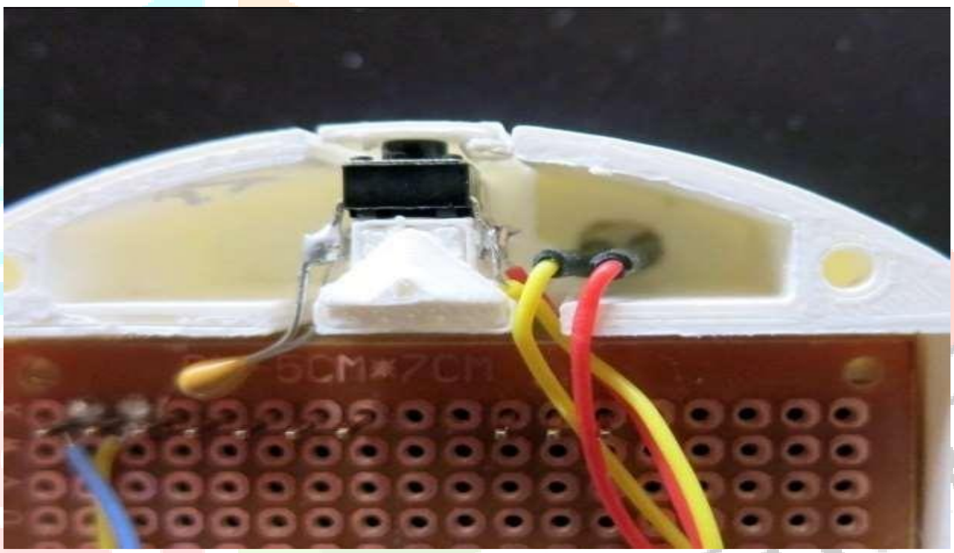

Figure 17: Wiring after connecting the photocell

Wifi Stick

Take Apart WiFi Stick

Like shown on intro of this instructable, this project uses the "EDIMAX EW-7811UN Wireless USB Adapter". Of course it should be possible to use just any kind of WiFi USB Stick in this project, but keep in mind that we just tested the one described. Now take apart the WiFi-stick by slightly lifting the plastic case at the flat side of the usb plug. You should be able to remove everything, such that you have the bare wifi board laying in front of you, like shown in the picture

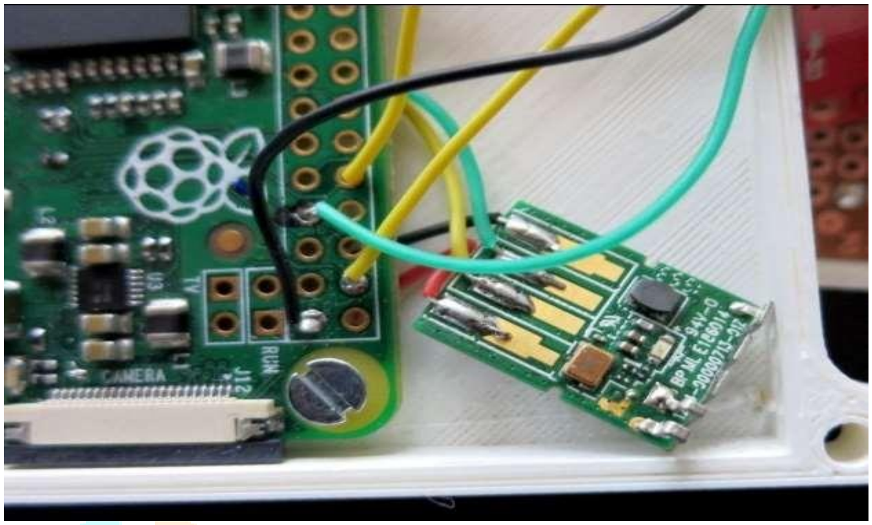

Figure 18: Soldering to Pis Test Pads

Soldering to Pis Test Pads

After taking apart the WiFi-stick you'll need to cut four wires to a length of about 4cm. Solder the USB connections to the Raspberry Pi Zero test pads like shown in the picture

Again, keep in mind, that it will not be possible to connect additional USB devices to the Pis micro USB connection after using the USB test pads. Once the soldering is done, plug in the Pis power supply and check if the connection is working. The WiFi-stick should start blinking a few seconds after plugging in the power supply. Wait until booting is done and establish a WiFi ssh connection. All further steps will be based on using a ssh connection like this. In order to check your internet connection and update your Pi to the latest Raspbian version, type:

```
sudo apt-get update<br>sudo apt-get upgrade
```

3D Printed Case

Once you are done building and configuring your setup you are ready to install it to the 3D printed case. The design of the case has been evaluated many times, such that you do not need to print support material at all and in the end all hardware parts fit into the body of the case perfectly.

Download and Print the 3D-Model

See the following link to the 3D model at thing verse: http://www.thingiverse.com/thing:2009740

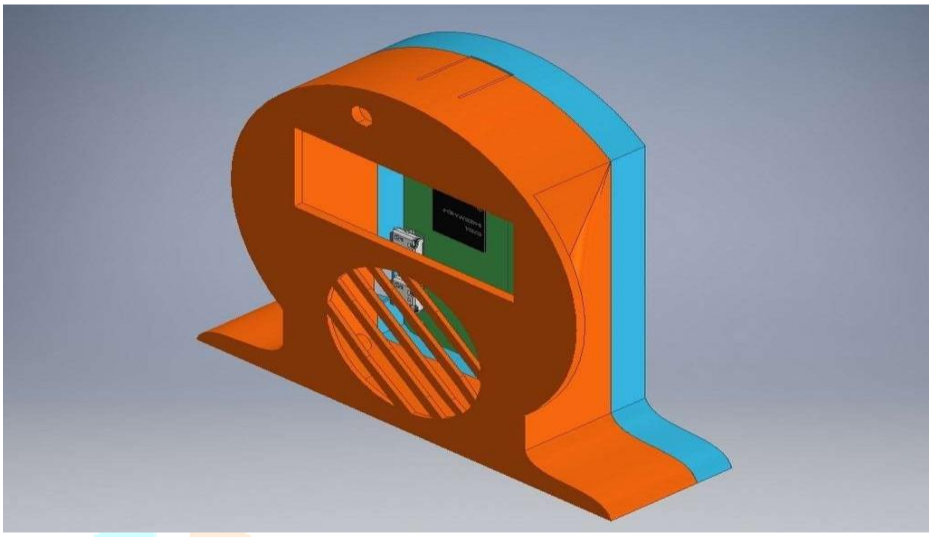

Figure 19: Case 3D Model

Assembling the Case

For the installation of the hardware parts in the case and the assembly of the case you need the following screws:

- 4 x M2 screws length: 6mm (hole grid board)
- 4 x M2.5 screws length: 6mm (Raspberry Pi Zero)
- 4 x M2.5 screws length: 16mm (case itself)

After printing the case you might need to remove overlapping material in order to achieve a better fitting. First put the speaker on the provided circular hole. Then try to fit the hole grid board together with the display into the provided slot. You might need to move the display slightly to all sides until it will 'click' into the case. Now put the hole grid board on top of everything. Make sure to not bend any wires too much when tightening the screws for the hole grid board. Slightly insert the button and the photocell. Afterwards lay down the back part next to the front part and find the right position for the Raspberry Pi Zero. Use the M2.5 screws and mount the Pi Zero to the back of the case. Before closing the case make sure the WiFi board and amplifier are positioned on the side, so you are able to close the case. In the end use the long (16mm) M2.5 screws to tighten the back to the front. Finally you are done building the smart alarm project and may start applying the program code to your needs.

Conclusion

Plug in your power supply and watch your alarm clock booting up. The *autostart.sh* script should be started automatically on each boot. Therefore the apache2 server and the *smart_alarm.py* (with the current version number in it) script should be started as well. Now you are able to set your alarm via your mobile phone, laptop or any other computer connected to your home wifi. Open your browser and type: `http://ip-address-of-your-pi`. Once you completed step 9.2 Change Hostname you are as well able to simply type: `your-pis-hostname`
in your browser. If your browser now starts your search-engine to search the web for your pis hostname you need to put *http://* front of the hostname. If your apache server is set up and configured to point to the *python_server.py* script you should be able to see the graphical user interface of the smart alarm project. Start playing around a little to get familiar with all the options and possibilities.

REFERENCES

[1] Thorat, S. S., Ashwini, M., Kelshikar, A., Londhe, S., & Choudhary, M. (2017). IoT based smart parking system using rfid. *International Journal of Computer Engineering In Research Trends, 4*(1), 9-12.

[2] Hodges, S., Taylor, S., Villar, N., Scott, J., Bial, D., & Fischer, P. T. (2012). Prototyping connected devices for the internet of things. *Computer, 46*(2), 26-34.

[3] Hurlburt, G. F., Voas, J., & Miller, K. W. (2012). The internet of things: A reality check. *IT Professional, 14*(3), 56-59.

[4] Restuccia, F., D'Oro, S., & Melodia, T. (2018). Securing the internet of things in the age of machine learning and software-defined networking. *IEEE Internet of Things Journal, 5*(6), 4829-4842.

[5] Madakam, S. (2015). Internet of things: smart things. *International journal of future computer and communication, 4*(4), 250.

[6] Yun, J., Ahn, I. Y., Choi, S. C., & Kim, J. (2016). TTEO (Things Talk to Each Other): Programming smart spaces based on IoT systems. *Sensors, 16*(4), 467.