



**IJIRCCCE**

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

**Volume 9, Issue 7, July 2021**

**ISSN** INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA

**Impact Factor: 7.542**



9940 572 462



6381 907 438



ijircce@gmail.com



www.ijircce.com

# IoT (Internet of Things) Based Smart Alarm Clock

Shreshth Shukla, Anshu Jha, Deepak Singh, Dr. A.A Shinde

Department of Electronics Engineering, Bharati Vidyapeeth (Deemed to-be University), College of Engineering,  
Pune, India

**ABSTRACT:** An application that enables user to set alarm in mobile phone that triggers the arduino system to turn on the rooms light, and play pre-defined music in external speakers, switches on the water heater, turns off fan beforehand of the wake-up time in an ascending style. The application is set up with mathematical calculation function to turn off the alarm. The proposed prototype is a smart alarm, which triggers the alarm tone in an ascending style 20 minutes prior to the actual wake-up time. It glows the rooms light in an ascending fashion 10 minutes beforehand the actual wake up time. It switches off fan in the room and triggers on the water heater 5 minutes prior to the wake up time, and after the system completes its process, our system hands over the control of entire electric circuit from arduino to the main power supply. It makes people to wake up from their bed in an enthusiastic way rather than either getting a bang wake up or end up sleeping without waking up. The module to turn off the alarm requires the user to solve a mathematical calculation which triggers the brain and boost it so that the user gets out of the sleep.

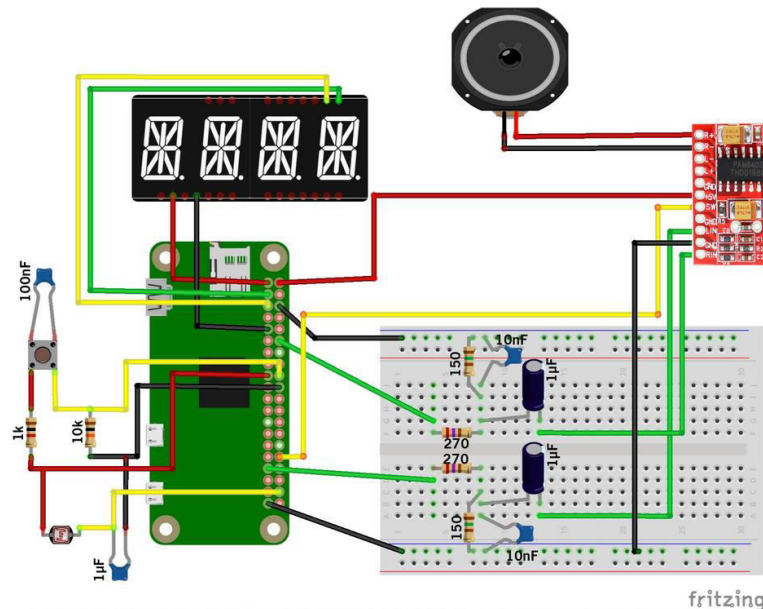
## I. INTRODUCTION

The smart alarm clock is an open-source project about building and programming an Internet of Things alarm clock. The following project provides all information you need to build the smart alarm clock. The IoT Smart Alarm Clock has text to speech synthesizer, three ways of wake-up sound, set alarm via smartphone or any other computer, running apache2 server, automatic, display brightness adjustment, audio amplifier volume control



## II. OBJECTIVE

The purpose of this project is to build an IoT Smart Alarm Clock. The project is designed to help students develop practical IoT device knowledge.



## III. APPLICATION

Digital Logic Design is the representation of signals and sequences of a digital circuit through numbers. It is the basis for digital computing and provides a fundamental understanding on how circuits and hardware communicate within a computer. Digital logic is typically implanted into most electronic devices, including calculators, computers, video games, and watches. This project introduces how to connect IoT (internet of things) devices to circuitry. This may relate to a professional engineering job by an engineer may design smart devices. Many engineers may get in this field with the evolution of smart products, such as smart lights, smart homes, smart appliances, smart televisions, smart phones and more

## IV. COMPONENTS

- Raspberry Pi Zero (with minor changes you might as well use a different Raspberry Pi)
- Keyboard, Mouse and Screen (for initial setup)
- Micro SD Card (8GB or more is recommended)
- USB-WiFi-Stick (for example: EDIMAX EW-7811UN Wireless USB Adapter)
- Audio Amplifier (PAM8403)
- Hole Grid Board (70mm x 50mm x 1.2mm, 24x18 holes, plus one hole in each corner, can be found easily online)
- Button (Tactile Switch)
- Photoresistor
- Speaker (3 Watt 4 Ohm 40mm diameter)
- 14-Segment Alphanumeric Display
- Resistors, Capacitors, Wires

## V. PROCEDURE

### 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

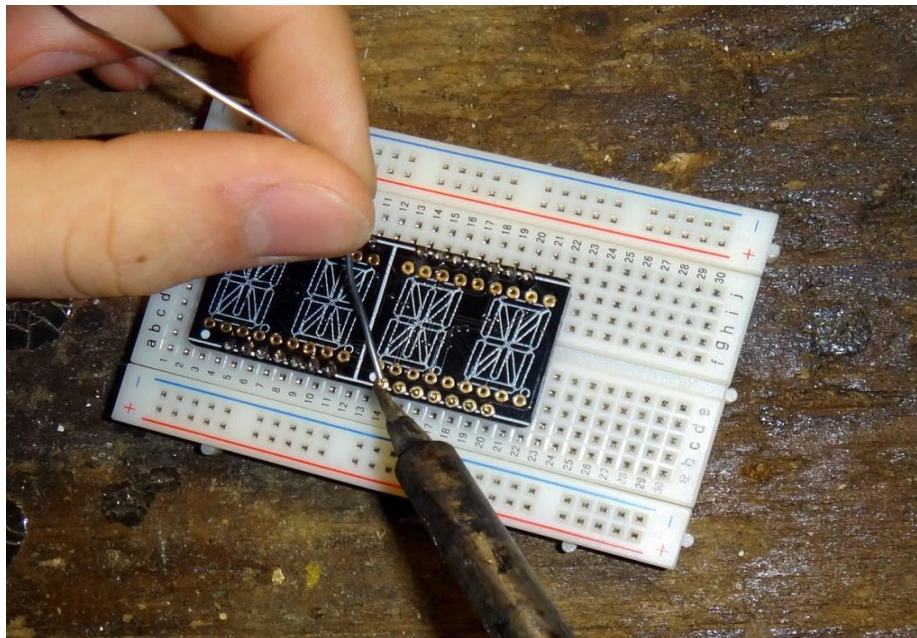
system somewhat. For example, if you want to use Raspberry Pi 3 you don't have to buy and use an additional WiFi-stick, might just use the on-board audio output, but might not be able to use the given 3D-printable case. So by changing some parts of this project you got to keep in mind that this instruction is based on using Raspberry Pi zero

## Raspberry Pi GPIO Configuration

Pin#	NAME	NAME	Pin#	
01	3.3v DC Power	DC Power 5v	02	Amplifier Power
03	GPIO02 (SDA1 , I <sup>2</sup> C)	DC Power 5v	04	
05	GPIO03 (SCL1 , I <sup>2</sup> C)	Ground	06	PWM Audio GND1
07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08	
09	Ground	(RXD0) GPIO15	10	
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12	PWM Audio right
13	GPIO27 (GPIO_GEN2)	Ground	14	
15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16	
17	3.3v DC Power	(GPIO_GEN5) GPIO24	18	Button Input
19	GPIO10 (SPI_MOSI)	Ground	20	Button + Photocell GND
21	GPIO09 (SPI_MISO)	(GPIO_GEN6) GPIO25	22	
23	GPIO11 (SPI_CLK)	(SPI_CE0_N) GPIO08	24	
25	Ground	(SPI_CE1_N) GPIO07	26	
27	ID_SD (I <sup>2</sup> C ID EEPROM)	(I <sup>2</sup> C ID EEPROM) ID_SC	28	
29	GPIO05	Ground	30	
31	GPIO06	GPIO12	32	Amplifier Switch
33	GPIO13	Ground	34	
35	GPIO19	GPIO16	36	
37	GPIO26	GPIO20	38	Photocell Input
39	Ground	GPIO21	40	

### Display

This step contains the soldering and configuration of the "Alphanumeric Display". Again you might choose a different display .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 I<sup>2</sup>C 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.



### Install Display Library

Once you are done assembling you need to install the Adafruit\_LED\_Backpack library. First prepare your system by typing:

```
sudo apt-get update
sudo apt-get install build-essential python-dev
```

also you'll need to install the python-smbus and python-imaging libraries:

```
sudo apt-get install python-smbus python-imaging
```

In order to clone the Adafruit github repository to your Pi. Unzip the file, navigate inside the library's directory and execute the setup.py file:

```
sudo python setup.py install
```

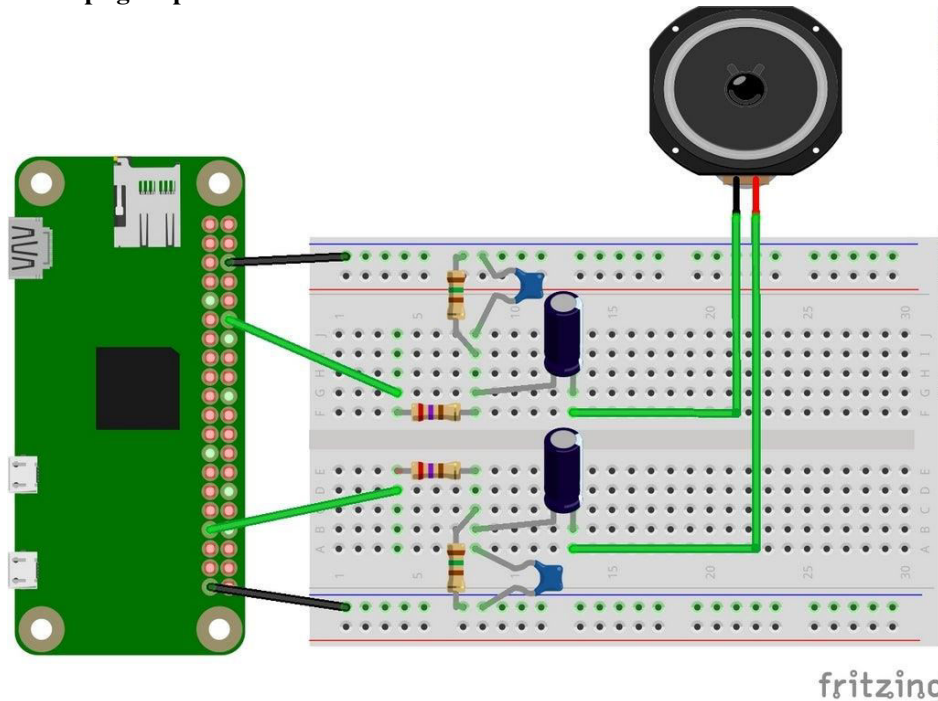
### Enable Audio

#### Low-Pass-Filter Circuit

You might have noticed the Raspberry Pi Zeros lack of an audio output. Building an alarm clock without any chance of playing sounds might cause a bunch of oversleeps and angry bosses. But don't worry some skillful people already found a hack to cover the lacking audio output. All you need to do is following the adafruit tutorial of how to Add a Basic Audio Output to Raspberry Pi Zero. Stick to Option 2. Manually Assigning PWM pins for and detailed instruction of how to configure your Raspberry Pi Zero. Here is a short summary of what you need to do.

First update your Pi to the latest version:

```
sudo apt-get update
```



#### Install pygame

You also need to install the pygame python library in order to play audio files via your python script. Doing so you need to

```
sudo apt-get install python-pygame
```

### Install Python Text to Speech

As well as .mp3 files we want to get the alarm clock speaking. Therefore, we need to install the python text to speech package (pyttsx). But first we have to install pip:

**sudo apt-get install python-pip**

Again test your system by running

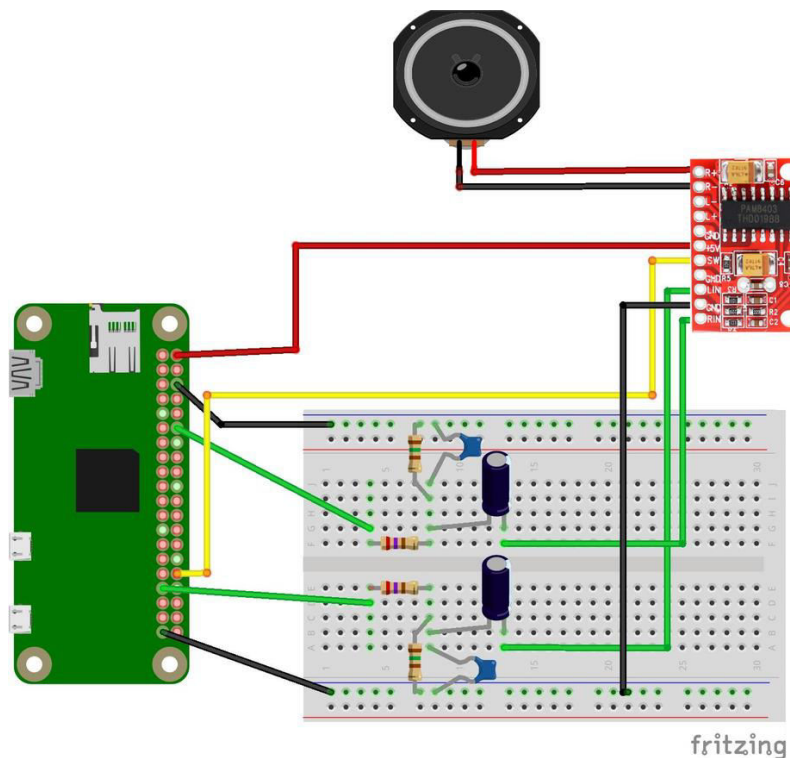
**: python smart\_alarm/tests/say\_text.py**

Because we don't just want to play mp3 files which we have locally on the Pis sdcard, we need to install the Music Player Client & Daemon. Using the music player client (mpc) we are able to listen to tons of different internet radio stations. Since this alarm clock is a Internet of Things alarm clock, we for sure want to enable listening to internet radio stations. So go ahead and install

**sudo apt-get install mpc**

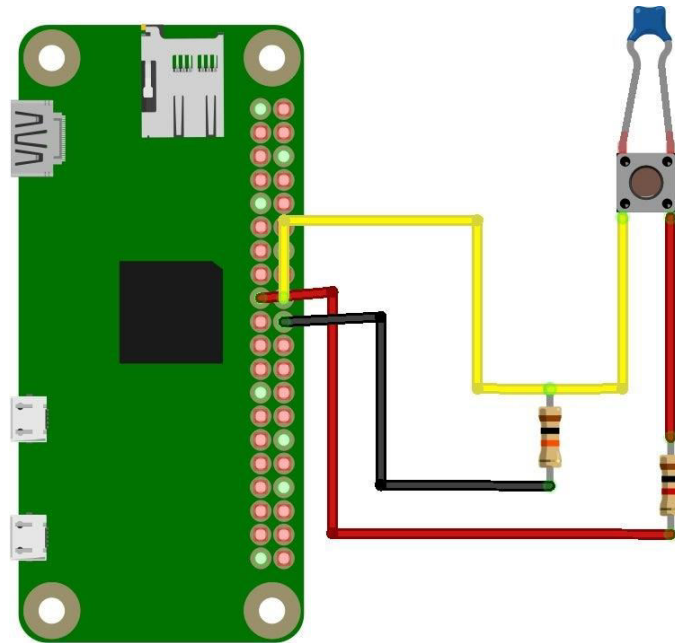
### Connecting the Amplifier

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.



### Connecting the Button

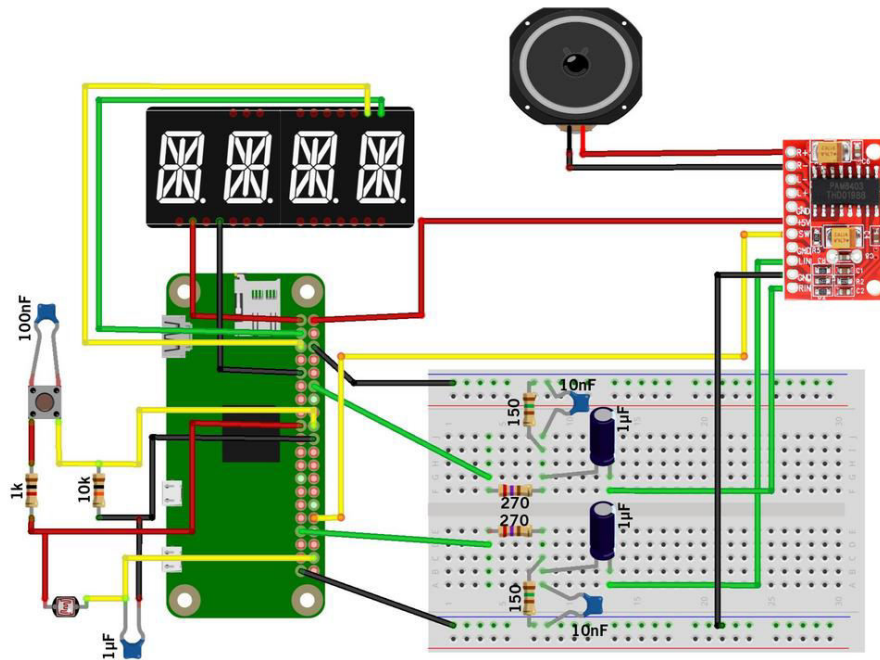
You need two resistors: 1x 1k Ohm and 1x 10k Ohm, one capacitor with 100nF and some wires. Connect everything like shown in the image above. The 1k Ohm resistor is connected between the two red wires, the 10k Ohm resistor is connected between the yellow and black wires. Your soldering on the hole grid board should like the following picture.



fritzing

### Photocell

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:

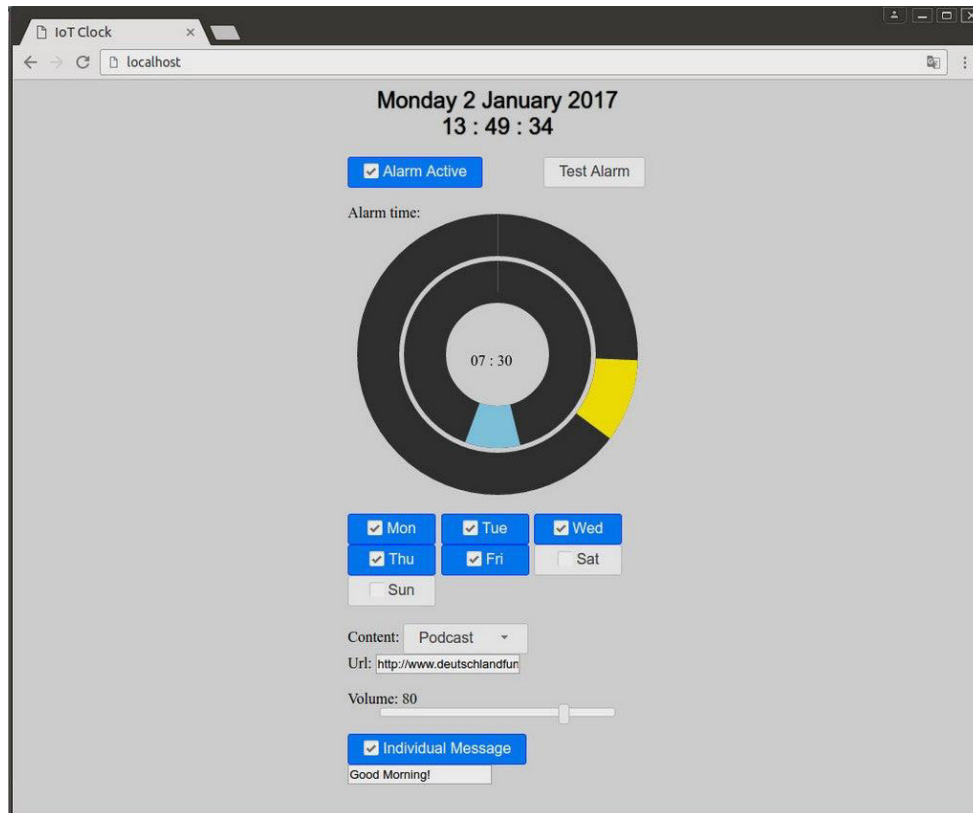


fritzing

### Setup and install Apache server

To install the apache server on the raspberry pi zero please follow the steps in this tutorial. The plain installation command is:

```
sudo apt-get install apache2 -y
sudo apt-get install libapache2-mod-wsgi
sudo a2enmod wsgi
```



### Change Hostname

We want to establish a connection to our Raspberry Pi Zero as easy as possible. Therefore we need to change the Pis hostname. Doing so we are able to simply type the Pis hostname in our smartphones (or any other computers) browser. In order to change your Pis hostname we recommend following this tutorial. First you need to edit the following file by typing:

```
sudo nano /etc/hosts
sudo nano /etc/hostname
sudo /etc/init.d/hostname.sh
```

### Change Password

Changing your user password is easy. Even though this is optional, we recommend doing it. Just type:

```
sudo raspi-config
```

### Configure WiFi

Plug in your WiFi stick and establish a connection to your local network. Once this is done type

```
Ifconfig
sudo raspi-config
```



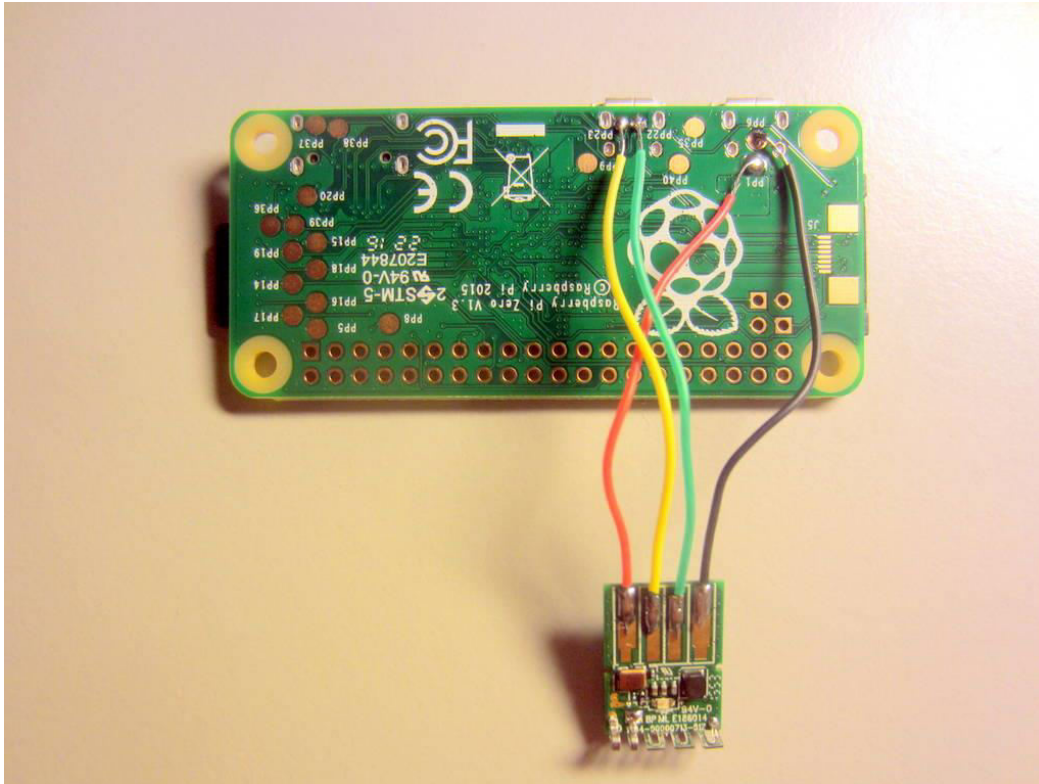
navigate to the Advanced Options menu and enable the ssh connection.

**sudo reboot**

your Pi in order to apply the changes. Now try to connect to your Raspberry Pi using your computer and ssh, like this:

**ssh pi@ip-address**

Replace 'ip-address' in this command by the Pis ip-address or its hostname.



### Configuring Multiple WiFi Connections

If you plan to use your device in multiple networks you'll need to configure them first. By just using one WiFi connection you might just skip this step. Adding additional network connections you need to modify the file

**/etc/wpa\_supplicant/wpa\_supplicant.conf.**

Therefore type:

**sudo nano /etc/wpa\_supplicant/wpa\_supplicant.conf**

### VI. CONCLUSION

Current wireless technologies, such as wireless body area networks and wireless personal area networks, provide promising applications in medical monitoring systems to measure specified physiological data and also provide location-based information, if required. With the increasing sophistication of wearable and implantable medical devices and their integration with wireless sensors, an ever-expanding range of therapeutic and diagnostic applications is being pursued by research and commercial organizations. This project aims to provide a comprehensive review of recent developments in wireless sensor technology for monitoring behavior related to human physiological responses. It presents background information on the use of wireless technology and sensors to develop a wireless physiological measurement system. A generic miniature platform and other available technologies for wireless sensors have been studied in terms of hardware and software structural requirements for a low-cost, low-power, non-invasive and unobtrusive system.



## VII. FUTURE ENHANCEMENT

During project building there are many changes from the point of learning environments and discussion among colleagues. It can directly increase the dedication and rational attitude toward myself. I conclude that the project program has provided many benefits to students even if there are minor flaws that are somewhat disfiguring condition, so that this weakness can be rectified in the future. We can conclude that this industry is through training I received a lot of exposure in the computing world.

## REFERENCES

- [1] Instructables. "IoT Smart Alarm Clock [Open Source Project]." Instructables, Instructables, 21 Sept. 2017, [www.instructables.com/id/IoT-Smart-Alarm-Clock-OpenSource-Project/](http://www.instructables.com/id/IoT-Smart-Alarm-Clock-OpenSource-Project/). Website: <https://www.instructables.com/id/IoT-Smart-Alarm-Clock-Open-Source-Project>.
- [2] Gary Scott, Jeannette Chin, A DIY approach to pervasive computing for the Internet of Things: A smart alarm clock, 5th Computer Science and Electronic Engineering Conference (CEEC), 2013.
- [3] Mohammad Ayyub, Dr. S.Kishor Reddy, Dr. G.Sai Kumar, Iot Based Intelligent Home Using Smart Devices, Complexity International Journal (CIJ), Volume 23, Issue 2, July-August 2019



**INNO**  **SPACE**  
SJIF Scientific Journal Impact Factor  
**Impact Factor: 7.542**



**ISSN** INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
**INDIA**



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

 **9940 572 462**  **6381 907 438**  **ijircce@gmail.com**



[www.ijircce.com](http://www.ijircce.com)

Scan to save the contact details