

Echipa 6:

Student: Ceuca Roxana **Student:**Iftimescu Marina

E-mail:roxana.ceuca@student.tuiasi.ro **E-mail:** marina.iftimescu@student.tuiasi.ro

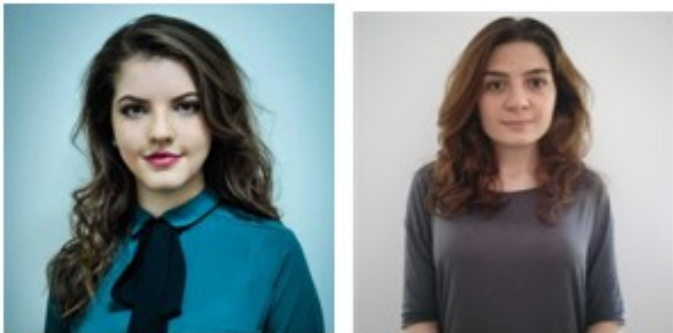
Picture: Picture:



Student: Pavel Alina-Elena **Student:**Tudose Georgiana

E-mail: alina-elena.pavel@student.tuiasi.ro **E-mail:**viorica_tudose10@yahoo.com

Picture: Picture:



Project name: 🍷 PiCoffeeAlarm 🍷

Hackster.io link for more details about project:

<https://www.hackster.io/350713/picoffeealarm-cb1a04>

Code: <https://github.com/georgiana-viorica/PiCoffeeAlarm-SM>

Github link for the code:

<https://github.com/georgiana-viorica/PiCoffeeAlarm-SM>

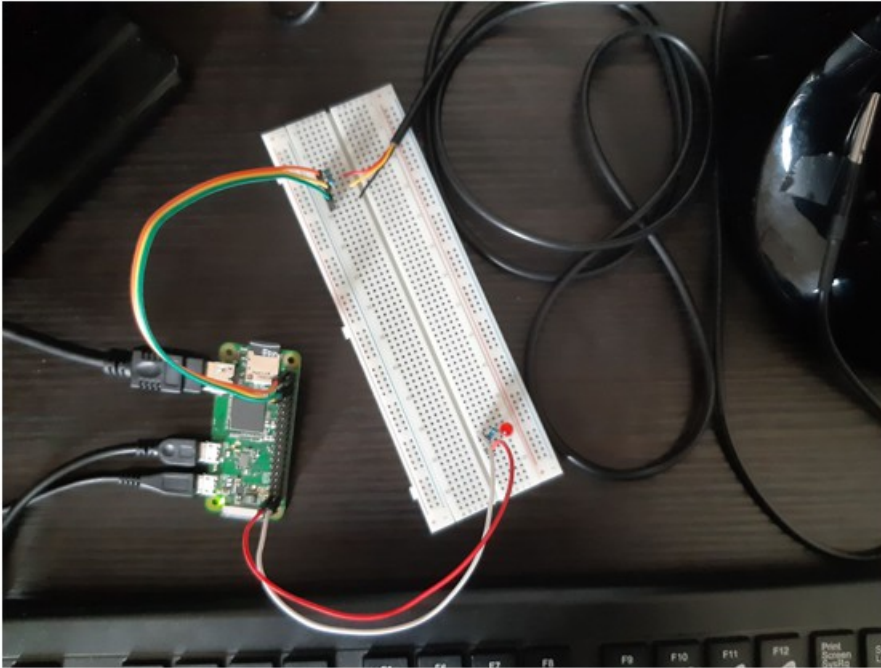
YouTube link for demo:

<https://www.youtube.com/watch?v=sOZVyqiLZ2I&feature=youtu.be>

Elevator Pitch: The project contains WEB development: Back-End (Node.js, Java

Script) and programming on a RaspberryPi running Raspbian (Linux).

Cover project:



Story:

The presented project, „PiCoffeeAlarm”, is meant for company employees who love coffee. In order to stay beside the coffee machine, you can use the assigned time for another types of activities because the drinker will be announced through a Short Message Service when the coffee is ready to serve. The user of the coffee machine needs to create an account through the Twilio Application, by registering its phone number. As well, we can note the fact that the messages can be sent simultaneously to several employees or customers.



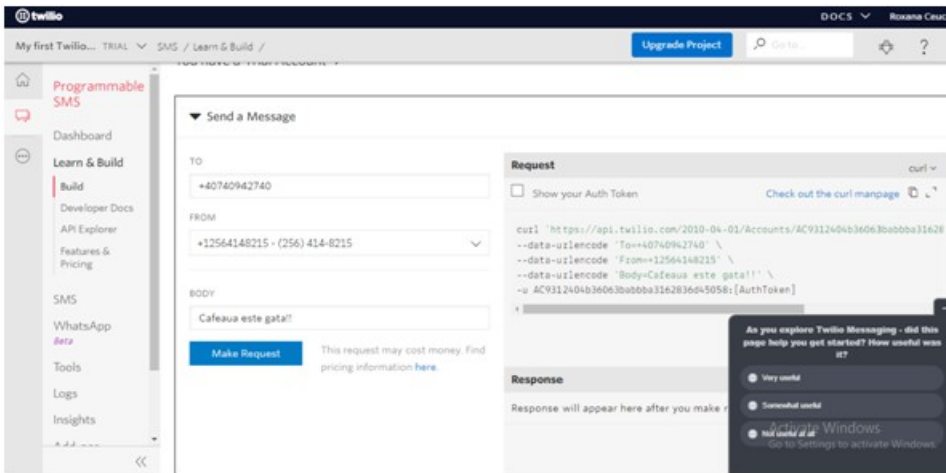
The Twilio Application

Twilio is a developer platform for communications. Software teams use Twilio APIs to add capabilities like voice, video, and messaging to their applications. This enables businesses to provide the right communications experience for their customers. Behind Twilio APIs is a Super Network, a software layer that connects and optimizes communications networks around the world. This is what allows your users to reliably call and message anyone anywhere.

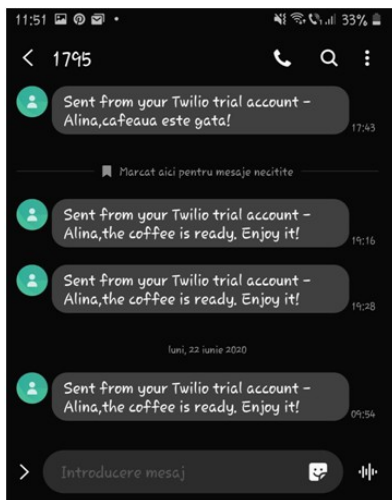
A TwiML app can be configured on a Twilio phone number for handling incoming calls, faxes, or messages. For complete instructions, please see the appropriate article:

- [Configuring Phone Numbers to Receive Calls](#)
- [Configuring Phone Numbers to Receive and Respond to SMS and MMS Messages](#)

The customer should create an account in order to receive or send messages.



When running the application, we will monitor the temperature of the coffee, and when it reaches the temperature of 93 degrees Fahrenheit, the consumer will receive a message on the phone with the text **"User's name,the coffee is ready.Enjoy it! "**, only if he has created an account on the Twilio platform .



Components and apps:

Hardware Components:

- RaspberryPi Zero W;
- 32 GB SD Card with NOOBs;
- 1 m HDMI-miniHDMI cable;
- 5V, 3A USB Type C power supply;
- 4 Digit 7 Segment Common Cathode Display
- 1 5 mm red LED;
- 2 resistors: 10K;
- Breadboard 830p;

- 5 male-female jumper wires for breadboard;
- DS18B20 Temperature Sensor;
- Coffee Machine.

DS18B20 Temperature Sensor

DS18B20 is a regular transistor, there is actually quite a lot going on inside.

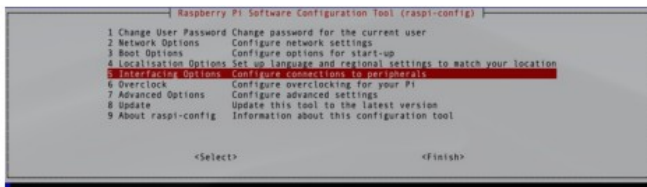
The chip includes the special 1-wire serial interface as well as control logic and the temperature sensor itself. Its output pin sends digital messages and Raspbian includes an interface to read those messages.

Wire using the 'raspi-config' tool on your Raspberry Pi the correct kernel modules will be loaded on subsequent reboots.

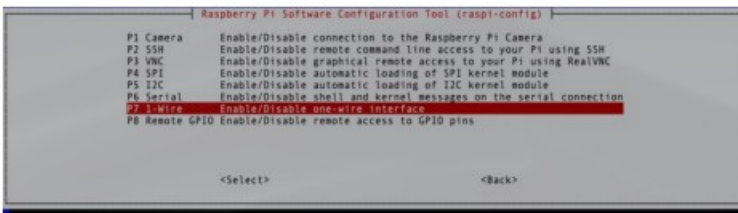
Enable 1-Wire

Execute: `sudo raspi-config`

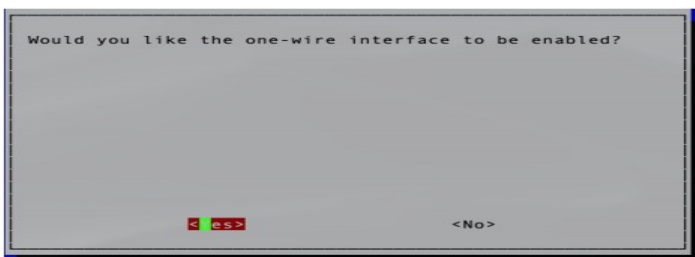
Select Interfacing Options:



Select 1-Wire:



Select Yes:



Exit out of raspi-config and reboot Raspberry Pi so the 1-wire kernel modules load up.

```
sudo reboot
```

Verify that the 1-Wire kernel modules have loaded on the next boot. Will appear something like the below output when running the lsmod command.

```
lsmod | grep -i wl_
```

```
wl_therm      16384  0
wl_gpio      16384  0
wire         45056  2 wl_gpio,wl_therm
```

Software applications:

- Raspberry Pi Raspbian
- Node.js
 - npm
 - ejs
 - body-parser
 - Twilio Application
- pigpio library
- RPi.GPIO library

To implement the server, we used Node.js having the language code which imports the standard FS library that allows us to interact with the filesystem. It also imports the PI GPIO libraries so that we can control the GPIO pins. We use Twilio to send a text message and we set up the LED by using the PI GPIO library to make pin 21 an output pin. The Celsius temperature is converted to Fahrenheit and also, there are some functions which verify the status of the preparation process (ready).

Initialize the Node.js dependencies

- npm init
- npm install ejs --save
- npm install body-parse --save
- npm install twilio-client --save
- npm install gpio

-npm install node-fetch --save

- **Spawn pigpio library daemon (you will probably need root permission)**

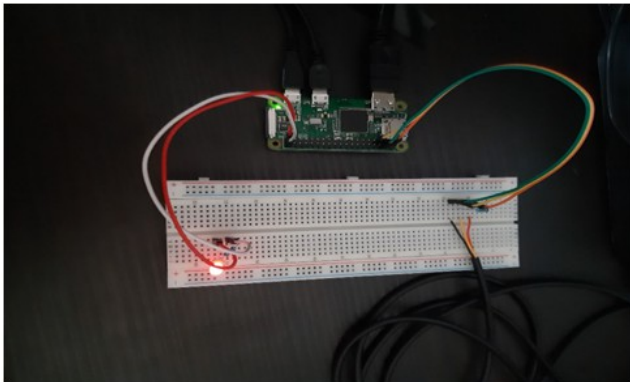
-sudo pigpiod

Our PiCoffeeAlarm

We're gonna be making our office coffee maker smart by building a coffee alert system that sends notifications when someone brews a fresh pot of coffee.

We're gonna need to continuously monitor the temperature of the coffee machine. To do this, we went with **DS18B20 Temperature Sensor** and a **Raspberry Pi Zero W**.

To see how hot it gets when it's not brewing, we imported these readings into Text File . We needed to make the temperature probe connect to **the breadboard** without falling out so we soldered the **positive and negative leads** from a **USB cable**. **The LED** will come on and blink when coffee is being brewed.



Schematics:

