

Nume studenți:

Cruț Andreea-Diana

Hâncu Anda-Ioana



Mail:

[andreea-diana.crut@student.tuiasi.ro](mailto:andreea-diana.crut@student.tuiasi.ro)

[anda-ioana.hancu@student.tuiasi.ro](mailto:anda-ioana.hancu@student.tuiasi.ro)

Proiect:

Press To Switch

Hackster: <https://www.hackster.io/346949/press-to-switch-9c0460>

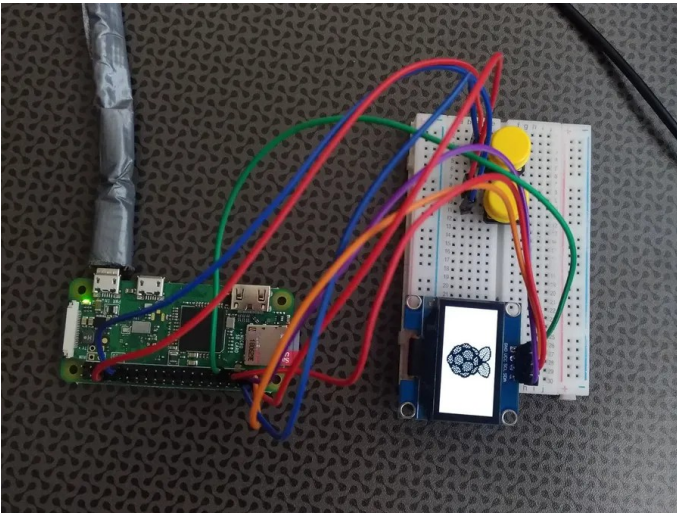
Video: <https://youtu.be/5w9bcW3yKX4>

## Echipa nr. 3: Press To Switch

Elevator Pitch

The design consists of displaying different screens by pressing the attached buttons, both forward and backwards.

## Cover Image

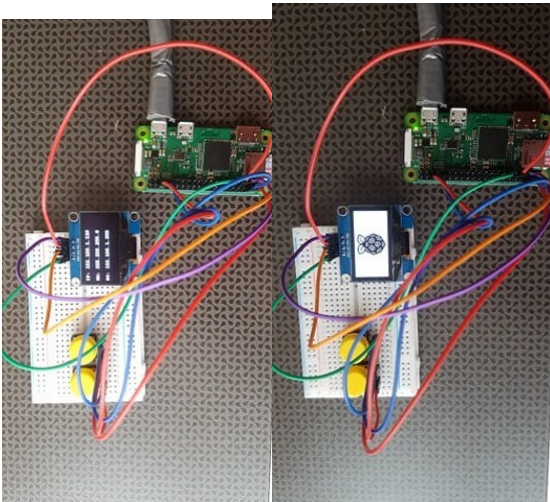
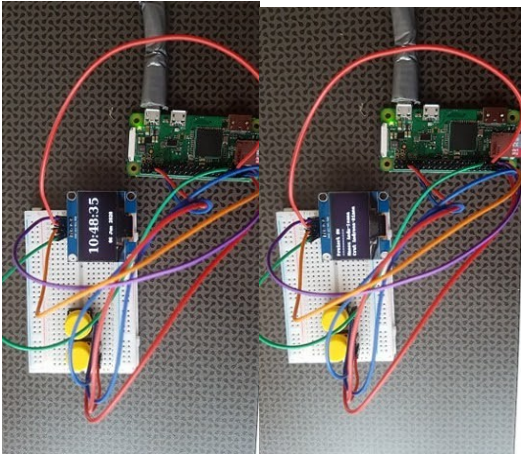


## Story

Having to do a project in a college course and not having experience working with the Raspberry Pi, we decided to make a simple project using an OLED display and 2 push buttons.

This project shows different displays, by pushing the 2 existing buttons, one for going forward and the other one for going backwards.

We implemented 4 types of displays for screening different information. In the first one we introduced ourselves and project title. The second one contains a RaspberryPi Logo. Continuing with the third, it displays the current date and time with updates. At last, we decided to show some information about the local address, showing the IP, with the subnet mask and the broadcast. Everytime we want to switch between displays, we just press the buttons for skipping or going back to the one we desire.



For this project you will need Raspberry Pi Zero Wifi, an OLED 1.3" 128x64 pixel display and two push buttons 12x12x7.3. The OLED we are using for this project has white pixels and uses an I2C interface which only requires four wires to be connected to the Pi.

## Setup

This kind of display has four pins, two are power (VCC and GND) and two are for the I2C interface (SDA and SCL). If you bought something similar to the one in the photo above, you may need to solder the pins before you get started.

Also, before starting the project, make sure you have an SD card(min. 8GB) with the latest Raspbian image.

The four pins of the display are connected to the Raspberry Pi through a breadboard, as it is described below:

OLED GPIO

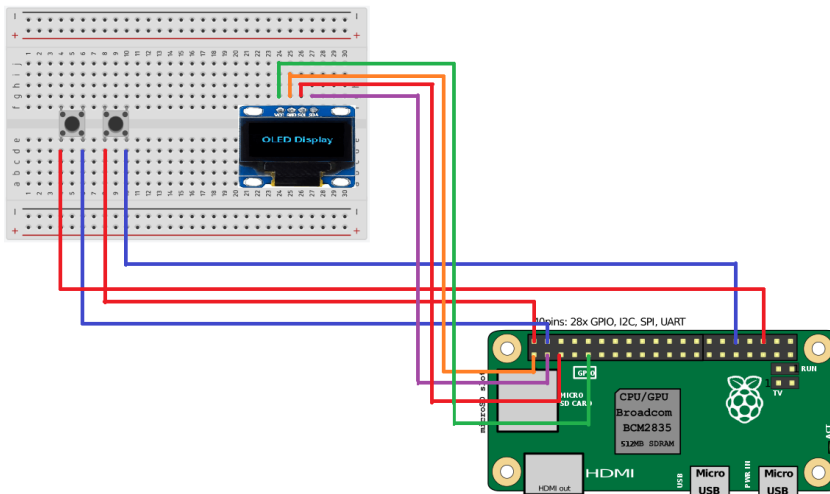
GND 9 (ground)

VCC 1 (power)

SCL GPIO3 (SCL I2C)

SDA GPIO2(SDA I2C)

If the I2C interface is disabled by default you need to enable it by running : `sudo raspi-config` . With the I2C library installed, use the following command to find the module on the I2C bus: `i2cdetect -y 1`

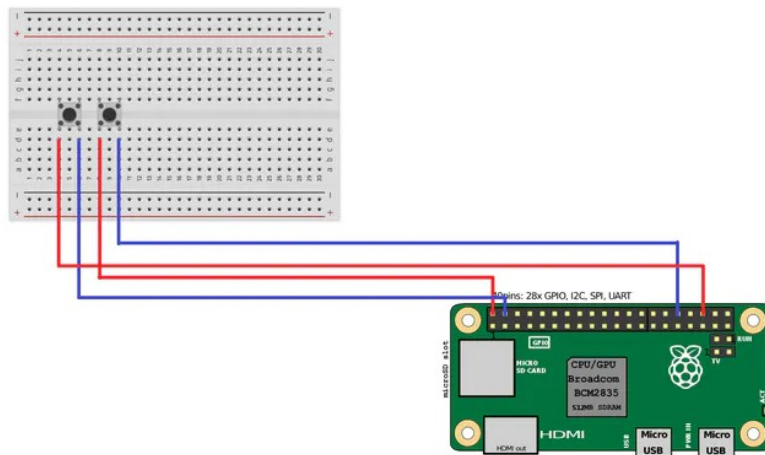


## Push Buttons Connection

Each button has one pin connected to power on Raspberry Pi and the other Pin connected to one GPIO pin.






The button in the left has one pin connected to power and the other one connected to GPIO16.

The button in the right has one pin connected to power and the other one connected to GPIO12.



## Components and apps


### Hardware components

	Raspberry Pi Zero Wireless	× 1
	ElectroPeak 0.96" OLED 64x128 Display Module	× 1
	SparkFun Pushbutton switch 12mm	× 2
	Male/Female Jumper Wires	× 1
	Breadboard (generic)	× 1

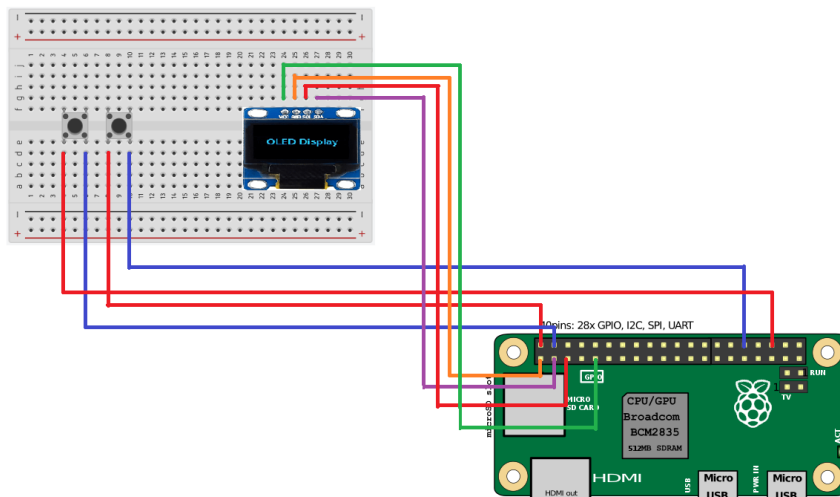
### Software apps and online services

	Raspberry Pi Raspbian
---	-----------------------

### Hand tools and fabrication machines

	Soldering iron (generic)
---	--------------------------

## Schematics



## Code

```
import RPi.GPIO as GPIO

from PIL import Image, ImageFont

from datetime import datetime

from luma.core.render import canvas

from luma.core.interface.serial import i2c

from luma.core.virtual import terminal

from luma.oled.device import sh1106

import re, os, time

GPIO.setmode(GPIO.BCM)

GPIO.setup(12, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

GPIO.setup(16, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

display=-1

def display_time():

    today_last_time = "Unknown"

    now = datetime.now().time()

    today_date = now.strftime("%d %b %Y")
```

```
today_time = now.strftime("%H:%M:%S")

if today_time != today_last_time:

    today_last_time = today_time

    with canvas(device) as draw:

        now = datetime.now()

        today_date = now.strftime("%d %b %Y")

        font=ImageFont.truetype('DejaVuSerif-Bold.ttf',24)

        x_pos=(device.width/2)-58

        y_pos=2+(device.height-12)/2-(35/2)

        draw.text((x_pos,y_pos),today_time,font=font,fill=255)

        x_pos=(device.width/2)-33

        y_pos=device.height-10

        draw.text((x_pos,y_pos),today_date,fill=255)

time.sleep(0.1)
```

```
def welcome():

    img_path = os.path.abspath(os.path.join(os.path.dirname(__file__),'pi_logo.png'))

    logo = Image.open(img_path).convert("RGBA")

    fff = Image.new(logo.mode, logo.size, (255,) * 4)

    background = Image.new("RGBA", device.size, "white")

    posn = ((device.width - logo.width) // 2, 0)

    background.paste(logo, posn)

    device.display(background.convert(device.mode))
```

```
def prezentare():

    term=terminal(device)

    term.println("Proiect SM")

    term.println("-----")

    term.println("Hancu Anda-Ioana")

    term.println("Crut Andreea-Diana")


def display_network():

    ipaddress=os.popen("ifconfig wlan0 | grep 'inet ' | awk -F: '{print $1}'").read()

    ipaddress=re.sub(' +',' ',ipaddress)

    ipaddress=ipaddress.split(' ')

    inet="IP:"+" "+ipaddress[2]

    subnet="NM:"+" "+ipaddress[4]

    broadcast="BC:"+" "+ipaddress[6]

    with canvas(device) as draw:

        draw.text((0,0),inet,fill=255)

        draw.text((0,20),subnet,fill=255)

        draw.text((0,40),broadcast,fill=255)

    serial=i2c(port=1,address=0x3C)

    device=sh1106(serial,rotate=0)

    device.clear()

    while True:
```



```
if(GPIO.input(12)):  
    display+=1  
    if(display>4):  
        display=0  
elif(GPIO.input(16)):  
    display-=1  
if(display==0):  
    display+=1  
    prezentare()  
elif(display ==1):  
    welcome()  
elif(display==2):  
    display_time()  
elif(display==3):  
    display_network()  
time.sleep(0.1)
```

Contributions:

Cruț Andreea-Diana:

- implementare funcție pentru afișarea timpului + datei
- implementare funcție pentru afișare logo
- conectare display+butoane cu RaspberryPi

Hâncu Anda-Ioana:

- implementare logică butoane pentru setarea stării display-ului în funcție de apăsare

- implementare funcție pentru afișarea prezentării
- implementare funcție pentru afișarea informațiilor despre rețea