**EEE 313 INTRODUCTION TO EMBEDDED SYSTEMS**

**INTERM APPLICATION HOMEWORKS**

**(25P) Q2.**  Perform an application by using two buttons and one 7-segment display with your Rpi, STM32 or Nano 33 board. The groups with 3 students will use Common Anode 7-segment display while others will use Common Cathode one.

**Purpose of Application:** Implement a counter that counts forward from 0 to 9 when one button is pressed, and from 9 to 0 when the other button is pressed. Clearly state the intended purpose (i.e, what you are trying to do for each situation) in your video capture and in this document.

**Circuit Diagram:** You are free to build your circuit for application. Draw your circuit in **Fritzing.**

**Restriction:** Neither the Arduino IDE nor the Arduino programming language will be used when performing this application. Use only Thonny IDE with MicroPhython/CircuitPython or STM32CubeIDE with C/C++.

**Homework Submission:** Record a videowith all the team members for your application. In your video content; explain your program codes, show your program to be compiled successfully, show your program to be uploaded to your board, show your circuit to be run successfully for each case. Note that no simulation study is requested.

The following files need to be uploaded to Teams.

1. This word document by completing the ANSWERS section (do not upload as pdf)
2. Your video file (Will be talked in English)
3. Fritzing circuit file
4. Application project folder created by IDE software. Include your source file

------------------------------------------------ANSWERS-----------------------------------------------

**Project Team:** Ata Güneş Suat Deniz

**Your Board:** Nvidia Jetson Nano Developer Kit

**Your 7-segment Display Type:** Common Cathode.

**Your Software IDE:** Python (In text file)

**Your Programming Language:** Python

**Application Purpose:**

**Hardware Components:**

1. **7-Segment Display Pins:**
   * A, B, C, D, E, F, G: Pins corresponding to the segments of the 7-segment display.
   * DP: Decimal point pin of the 7-segment display.
2. **Switch Pins:**
   * **switchUpPin**: Pin connected to a push-button switch for incrementing a counter.
   * **switchDownPin**: Pin connected to a push-button switch for decrementing a counter.

**Functionality:**

1. **Counter Display:**
   * The script initializes a counter variable (**counter**) and sets up GPIO pins for the 7-segment display and the two push-button switches.
   * The **changeNumber** function is defined to control which segments of the 7-segment display are illuminated based on the digit to be displayed.
2. **Button Input Handling:**
   * The main loop continuously monitors the state of the push-button switches (**switchUpPin** and **switchDownPin**).
   * If the "up" button is pressed, the counter is incremented (up to a maximum value of 9) and the display is updated accordingly.
   * If the "down" button is pressed, the counter is decremented (down to a minimum value of 0) and the display is updated accordingly.
3. **Display Update:**
   * The **changeNumber** function is called to update the 7-segment display based on the current value of the counter.
4. **Cleanup:**
   * The GPIO pins are cleaned up using **GPIO.cleanup()** in the **finally** block, ensuring that the resources are released when the script is interrupted or terminated.

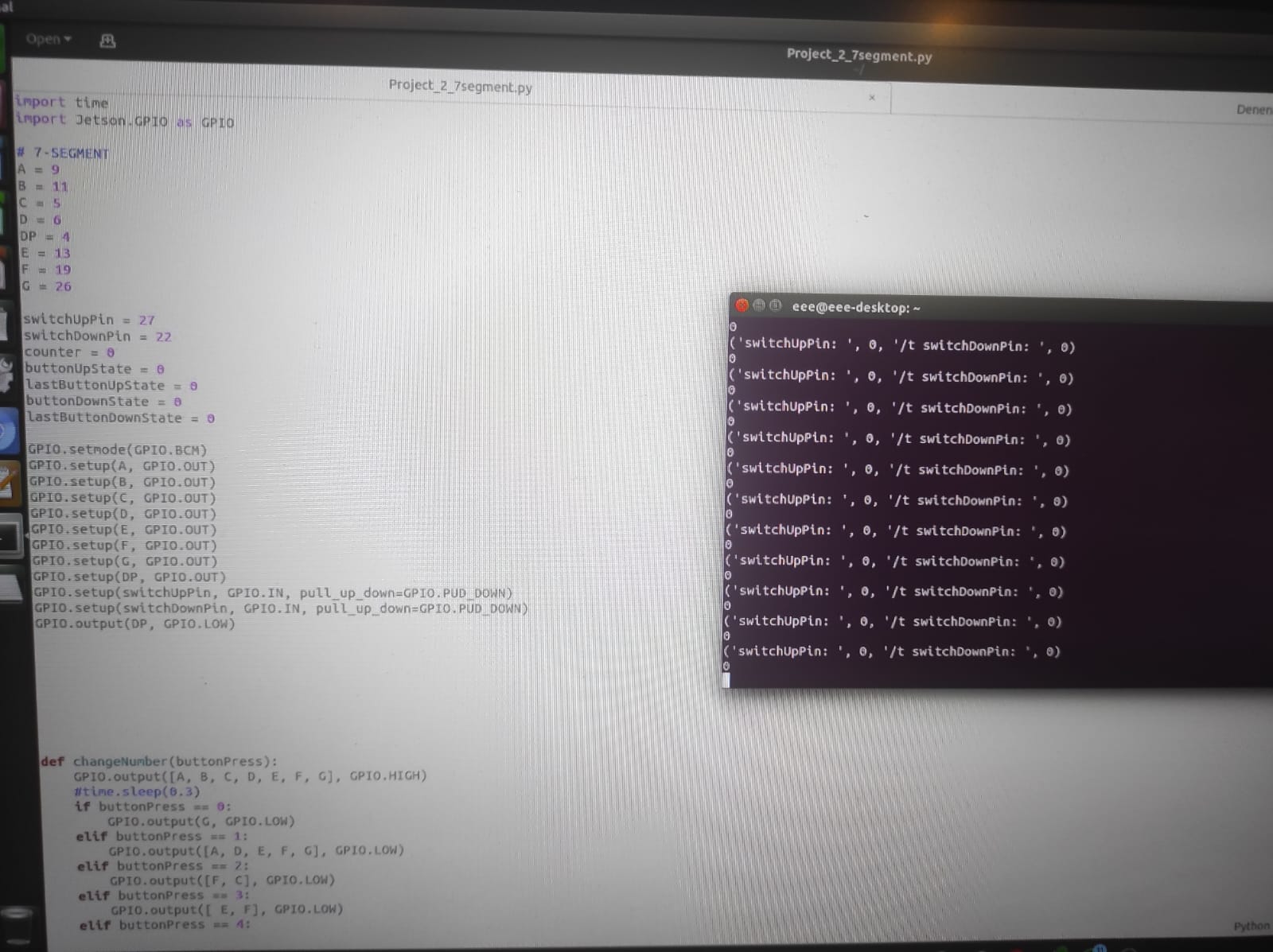
**Notes:**

* The script uses the **Jetson.GPIO** library, which is likely specific to NVIDIA Jetson platforms. If you are using a different platform (e.g., Raspberry Pi), you might need to replace it with the appropriate GPIO library (e.g., **RPi.GPIO**).
* The script includes print statements for debugging purposes, providing information about the state of the push-button switches and the current counter value.

**Fritzing Circuit Diagram:**

A circuit board with wires

Description automatically generated**Program codes:**

 **A screen shot of a computer screen

Description automatically generated**

**Photo for your circuit (only 1 photo):**

A close-up of a circuit board

Description automatically generated