Hands-On Guide: Raspberry Pi with Drowsiness & Yawn Detection

1. Installing the Operating System on the Raspberry Pi

- Download the Raspberry Pi Imager from the official website: https://downloads.raspberrypi.org/imager/imager_latest.exe
- Insert the SD card into the computer and use the Imager to select Raspberry Pi 4, the recommended OS, and the SD card.
- In the 'OS Customization Settings' menu, set hostname, username, password, and Wi-Fi credentials.
- Enable SSH and VNC.
- Save the settings and write them to the SD card.
- Insert the SD card into the Raspberry Pi, connect Mini-HDMI, keyboard, mouse, and power via USB-C.
- Run 'sudo raspi-config' and enable VNC.
- Access via RealVNC Viewer or SSH using Windows PowerShell.

2. Installing Required Libraries

- Open a terminal (VNC or SSH).
- Update and upgrade the system: sudo apt update && sudo apt upgrade
- Check Python version: python --version
- Create a virtual environment: python3 -m venv drowsiness --system-site-packages
- Activate the environment: source drowsiness/bin/activate
- Install required libraries: numpy, scipy, scikit-image, cmake, dlib, opencv-python, imutils, argparse
- Increase swap memory to 1024 MB, perform installation, then revert to 512 MB.

3. Running the Program

- Download files from GitHub (except drowsiness_yawn.py) and save them in the home directory.
- Save drowsiness_yawn.py separately.
- Activate the virtual environment and run the program: source drowsiness/bin/activate && python drowsiness_yawn.py
- Quit with the 'q' key.

4. Connecting and Using LEDs for Traffic Light Indication

- Connect LEDs: Green (GPIO 17), Yellow (GPIO 27), Red (GPIO 22).
- Copy required files into the home directory.
- Start the program: source drowsiness/bin/activate && python drowsiness_yawn_2.py
- Quit with 'q'.

5. Auto-start Program on Boot

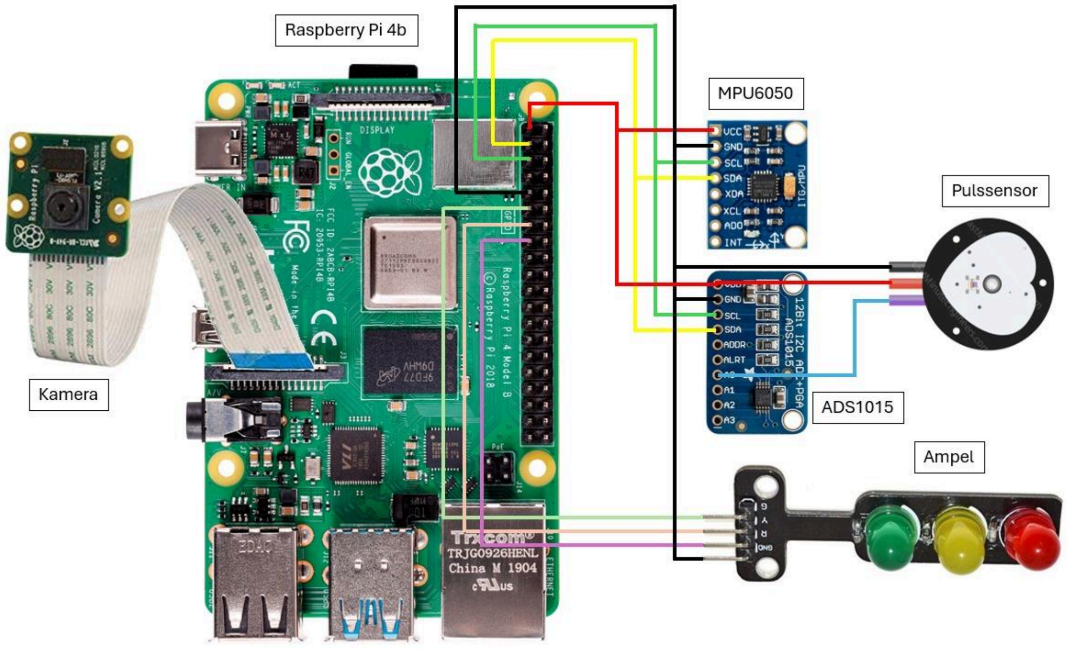
- Copy the program to /home/USERNAME and make it executable: chmod +x drowsiness_yawn.py
- Create a systemd service file: /etc/systemd/system/yawn_monitor.service
- Adjust the service file content (replace USERNAME).
- Enable and start the service: sudo systemctl enable yawn_monitor.service && sudo systemctl start yawn_monitor.service
- Reboot the Raspberry Pi.

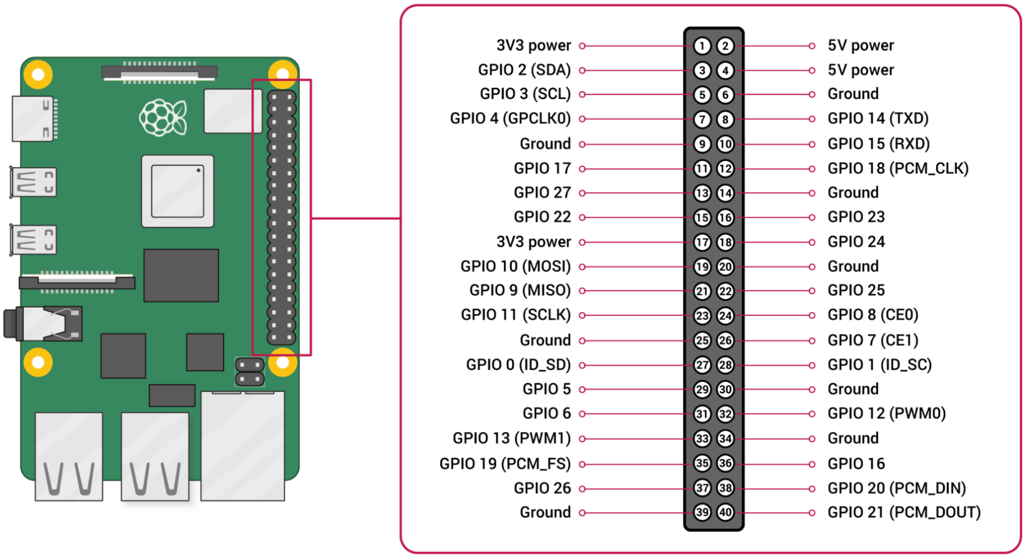
6. Setting up the MPU6050 Sensor

- Wiring: SDA → GPIO 2, SCL → GPIO 3, VCC → 3.3 V, GND → GND.
- Install libraries: sudo apt-get install python3-smbus i2c-tools && pip3 install mpu6050-raspberrypi
- Enable I^{2}C on the Raspberry Pi.
- Check connection: i2cdetect -y 1 (address 68 should appear).
- Run the test program (head movements detected).

7. Setting up the Pulse Sensor

- Connect pulse sensor to ADS1015: Red → VCC, Black → GND, White → A0.
- Connect ADS1015 to Raspberry Pi: VCC → 3.3 V, SCL → GPIO 3, SDA → GPIO 2, GND → GND.
- Run test program to display heart rate.
- Combine MPU6050 and ADS1015 and run extended code.


Raspberry Pi 4b
MPU6050
VCC
GND
SCL
SDA
XDA
XCL
ADO
INT
Pulssensor
GND
SCL
SDA
ADDR
ALRT
A0
A1
A2
A3
ADS1015
Kamera
Ampel


3V3 power
1
2
5V power
GPIO 2 (SDA)
3
4
5V power
GPIO 3 (SCL)
5
6
Ground
GPIO 4 (GPCLK0)
7
8
GPIO 14 (TXD)
Ground
9
10
GPIO 15 (RXD)
GPIO 17
11
12
GPIO 18 (PCM_CLK)
GPIO 27
13
14
Ground
GPIO 22
15
16
GPIO 23
3V3 power
17
18
GPIO 24
GPIO 10 (MOSI)
19
20
Ground
GPIO 9 (MISO)
21
22
GPIO 25
GPIO 11 (SCLK)
23
24
GPIO 8 (CE0)
Ground
25
26
GPIO 7 (CE1)
GPIO 0 (ID_SD)
27
28
GPIO 1 (ID_SC)
GPIO 5
29
30
Ground
GPIO 6
31
32
GPIO 12 (PWM0)
GPIO 13 (PWM1)
33
34
Ground
GPIO 19 (PCM_FS)
35
36
GPIO 16
GPIO 26
37
38
GPIO 20 (PCM_DIN)
Ground
39
40
GPIO 21 (PCM_DOUT)