Assignment 3

Study of Raspberry pi board with its pin specifications

Shreya Pawaskar
Cno - C22018881961
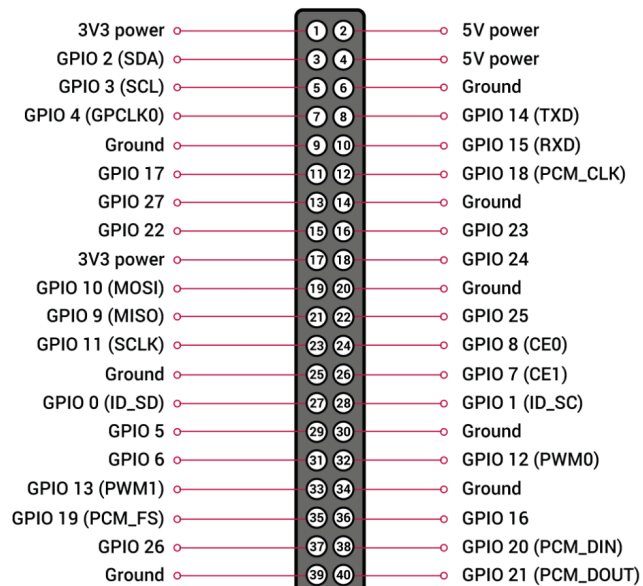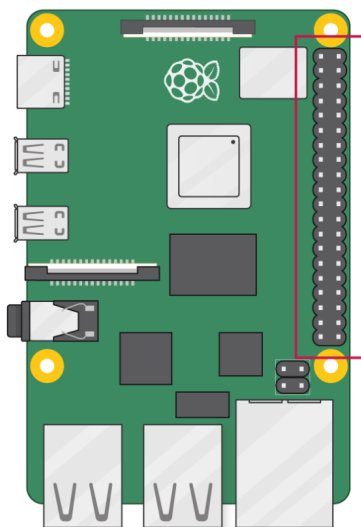Roll no - 4947
Division - C

Question 1: What is GPIO?
Answer 1:

- A general-purpose input/output is an uncommitted digital signal pin on an integrated circuit or electronic circuit board which may be used as an input or output, or both, and is controllable by the user at runtime.
- GPIOs have no predefined purpose and are unused by default.
- It can be used with sensors, diodes, displays, and System-on-Chip modules.
- It allows for providing external power supply, remote control of connected devices, broadcasting more contextual data, or defining contents for custom Bluetooth data packets.
- The General Purpose I/O IP provides up to 32 I/O ports that can be programmed individually for input, output, or bidirectional operation.
- Each port can be programmed to trigger the GPIO interrupt on level (high, low) or edge (rising, falling, any) events.
- GPIO can be used in three modes:
  - i. Input
  - ii. Output
  - iii. UART Interface

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)

Question 2: Difference between the versions of Raspberry-Pi boards.
Answer 2:

	Raspberry Pi 1 Model A	Raspberry Pi 1 Model A+	Raspberry Pi 1 Model B	Raspberry Pi 1 Model B+	Raspberry Pi 2 Model B	Raspberry Pi 3 Model B	Raspberry Pi Zero
USB 2.0 Ports	1	1	2	4	4	4	1 (Micro-USB)
Ethernet	None	None	10/100 Mbit/s	10/100 Mbit/s	10/100 Mbit/s	10/100 Mbit/s	None
Bluetooth	None	None	None	None	None	4.1	None
WiFi	None	None	None	None	None	802.11n	None
Audio In	I²S	I²S	I²S	I²S	I²S	I²S	I²S
Audio Out	I²S, analog (3.5mm jack), digital (HDMI)	I²S, analog (3.5mm jack), digital (HDMI)	I²S, analog (3.5mm jack), digital (HDMI)	I²S, analog (3.5mm jack), digital (HDMI)	I²S, analog (3.5mm jack), digital (HDMI)	I²S, analog (3.5mm jack), digital (HDMI)	Digital (mini-HDMI), analog GPIO PWM
Video In	CSI Camera Connector	CSI Camera Connector	CSI Camera Connector	CSI Camera Connector	CSI Camera Connector	CSI Camera Connector	None
Video Out	HDMI, Composite (RCA)	HDMI, Composite (TRRS)	HDMI, Composite (RCA)	HDMI, Composite (TRRS)	HDMI, Composite (TRRS)	HDMI, Composite (TRRS)	Mini-HDMI, GPIO Composite
External Storage	SD	MicroSD	SD	MicroSD	MicroSD	MicroSD	MicroSD

Question 3: Which are the different Operating Systems that can be installed on Raspberry-Pi?
Answer 3:

- ❖ Raspbian (general purpose OS)
- ❖ OSMC
- ❖ Open ELEC
- ❖ RISC OS
- ❖ Windows IOT Core
- ❖ Lakka
- ❖ Retro pie, etc.

Question 4: Difference between BeagleBone Black, Arduino and Raspberry- Pi 3 Model B board.
Answer 4:

Name	Arduino Uno	Raspberry Pi	BeagleBone
Model Tested	R3	Model B	Rev A5
Price	$29.95	$35	$89
Size	2.95"x2.10"	3.37"x2.125"	3.4"x2.1"
Processor	ATMega 328	ARM11	ARM Cortex-A8
Clock Speed	16MHz	700MHz	700MHz
RAM	2KB	256MB	256MB
Flash	32KB	(SD Card)	4GB(microSD)
EEPROM	1KB		
Input Voltage	7-12v	5v	5v
Min Power	42mA (.3W)	700mA (3.5W)	170mA (.85W)
Digital GPIO	14	8	66
Analog Input	6 10-bit	N/A	7 12-bit
PWM	6		8
TWI/I2C	2	1	2
SPI	1	1	1
UART	1	1	5
Dev IDE	Arduino Tool	IDLE, Scratch, Squeak/Linux	Python, Scratch, Squeak, Cloud9/Linux
Ethernet	N/A	10/100	10/100
USB Master	N/A	2 USB 2.0	1 USB 2.0
Video Out	N/A	HDMI, Composite	N/A
Audio Output	N/A	HDMI, Analog	Analog

Question 5: Different applications of BeagleBone Black, Arduino and Raspberry -Pi.
Answer 5:
  ★ Applications of BeagleBone Black Board:
    ○ 1. Development of Internet-connected systems (e.g., Internet of Things (IoT),
    ○ Cyber Physical Systems)
    ○ 2. Rich user-interface touchscreen applications
    ○ 3. IoT
    ○ 4. Cloud Computing
    ○ 5. Super Computer such as Cluster of Board's
    ○ 6. 2.4 GHz RC Planes etc.

  ★ Applications of Arduino Uno:
    ○ 1. Robot/Motor Control
    ○ 2. UAVs
    ○ 3. Sensor Networks

- ○ 4. Development of Automation System
- ○ 5. Designing of basic circuit designs
- ○ 6. In developing projects based on code-based control etc.

- ★ Applications of Raspberry -Pi:
  - ○ 1. Desktop PC can be made
  - ○ 2. Home Automation
  - ○ 3. Media Streamer
  - ○ 4. Stop Motion Camera
  - ○ 5. Game Server
  - ○ 6. For controlling Robots etc.

Question 6: List step by step process for installing Operating System on Raspberry-Pi.
Answer 6:
- ➢ The Operating System can be installed using an SD card in Raspberry-Pi.
- ➢ Here, the OS images are written to the SD card. Raspberry Pi operating systems are available as a disk image, either ISO or IMG format.
- ➢ These can be written on the SD card using several Tools like Raspberry Pi Imager, Etcher, etc.
- ➢ Steps-
  - ○ 1. Download Raspberry Pi Imager
  - ○ 2. Once installed on the computer, Under Operating System click Choose OS
  - ○ 3. Browse the list for your preferred OSs and select the one you want
  - ○ 4. Click Ctrl+Shift+X to preconfigure advanced options (see below)
  - ○ 5. Next, click Choose Storage to select the SD card
  - ○ 6. Click Write.

Question 7: How to use Python language for programming Embedded development boards.
Answer 7:
- ➔ By default, Raspbian (Stretch version April 2018 and earlier) uses Python 2.
- ➔ However, versions 2 and 3 come installed by default. We just have to make 1 minor change so that the Pi uses Python 3 whenever we type python into a terminal.
- ➔ In a terminal window, enter the following command:
  - ◆ - python –version
  - ◆ - Install pip

Full Desktop: If you are using the full desktop version of Raspbian, you should have pip already installed.

Headless: If you are using Raspbian Lite, the Python package manager, pip, does not come pre-installed.

As a result, you will need to install it with the commands:
- ◆ - sudo apt-get update
- ◆ - sudo apt-get install python3-pip

Press y when prompted.

Note that to use pip for Python 3, you will need to use the command pip3. However, we can modify the .bashrc file to use pip instead of pip3, as the rest of the tutorial will show examples using pip:

◆ - nano ~/. bashrc

Scroll down to the bottom, and add the following command to the file:

◆ - alias pip=pip3

From a terminal, enter the following command to start the Python interpreter:

◆ - Python

You should be presented with a different command prompt, consisting of 3 greater than signs >>>.

Type the following command:

◆ - Print ("Hello, World!")

Once you press enter, you should see the phrase Hello, World! repeated back to you.

Question 8: Explain use of RPi.GPIO library.
Answer 8:

☐ A powerful feature of the Raspberry Pi is the row of GPIO (general-purpose input/output) pins along the top edge of the board. The GPIO pins allow the Raspberry Pi to control and monitor the outside world by being connected to electronic circuits.

☐ The Pi is able to control LEDs, turning them on or off, run

☐ motors, and many other things. It's also able to detect whether a switch has been pressed, the temperature, and light. We refer to this as physical computing.

☐ This tool is provided by the GPIO Zero Python library, which is installed by default on the Raspberry Pi OS desktop image, but not on Raspberry Pi OS Lite.

Functions:

1. cleanup ()
- Cleans up the modules' running operations. It will set all pins configured before to input.

2. gpio_function (channel)
- Get the configuration of a pin.
o Parameters:
● channel channel/pin to be reported
o Returns:
● Pin configuration, being IN, OUT, I2C, PWM, SERIAL, SPI or UNKNOWN.

3. input (channel)
- Read the pin value. For pins configured as output, it returns the current output value.
o Parameters:
● channel channel/pin to be read
o Returns:
● Boolean true for a HIGH value, or false for a LOW value

4. output (channel, value)

    - Sets the output of a pin.

        o Parameters:

- channel channel/pin to be changed
- value (boolean) Use a truthy value to set the pin out to HIGH, or falsy to set to LOW. NOTE: a numeric '0' is also considered falsy! for compatibility with the original Python code.

5. setmode (mode)

    - Sets the pin numbering scheme to be used.

        o Parameters:

- mode (optional) either BCM (chip numbering) or BOARD (Rpi connector numbering)

        o Returns:

- currently set mode, being BCM, BOARD, or UNKNOWN.

6. setup_channel (channel, direction, pull_up_down, initial)

    - Sets a channel up on the GPIO interface.

        o Parameters:

- channel channel/pin to be setup
- direction Sets the direction of the pin, either IN or OUT
- pull_up_down (optional, only for inputs) Should the builtin pullup/down resistor be used. Either PUD_OFF, PUD_DOWN, or PUD_UP
- initial (boolean, optional, only for outputs) Should an initial value be set? set to truthy value to set the pin out to HIGH, or falsy to set to LOW.

        o NOTE: a numeric '0' is also considered falsy! for compatibility with the original Python code.

7. setwarnings (mode)

    - Turn warnings on or off.

        o Parameters:

- mode if nil or false turns warnings off, or on otherwise

Crossword:

Across	Down
2 raspi - config	1 SD Card
4 DHCP	3 Forty
8 Computer	5 Pico
9 Nano	6 True
10 Python	7 Raspbian
11 HDMI	