Word Count: 1298

Section 5, Physical Computing with Raspberry Pi 4

In this section, we will cover the following topics

● Getting started with physical computing on the Raspberry Pi 4

● Introduction to Hardware Interfacing on the Raspberry Pi 4

● Working with LEDs & Buttons

● Working with LDR & PIR Sensors

● Working with Relays

● Working with Ultrasonic Sensor & DC Motors

● Working with UART, I2C & SPI Protocols

Video, Getting started with Physical computing on the Raspberry Pi 4

In this video, we will learn about physical computing, it’s working principle, and finally get a deep understanding of the GPIO pins.

Physical Computing is the place where Computer Science meets Electronics. In layman’s terms, Physical computing refers to the creation of devices that interact with the world around them. Such a device would be able to sense its environment, process the data, and perform some action. Thus “sense - think -act” cycle is the core principle of physical computing. Physical computing opened up space for learners to explore computational ideas in a creative and interactive way. This approach of interdisciplinary learning offered by physical computing is one of the key factors for innovations in fields like Internet of Things and Smartphones

You may have seen some really cool Raspberry Pi hardware projects, but when you take a closer look, they seem impossibly complex. It’s all rather daunting but fear not. It’s simpler than it looks. The functions of any physical computing project can be split in three ways. The computer measures inputs, it uses software to think about what it will do with those measurements, then it makes something happen with the result. If you see a project you’d like to make, then splitting it into those three parts makes it much simpler to understand. Taking a look at those inputs, just how does a computer measure something? The answer is that it asks a sensor connected to one of its inputs, and receives the reading. That sensor can be as simple as a push-button, a temperature sensor, or a Pi Camera Module. It doesn’t even need to be physical hardware. For instance, a feed of the temperature in your city from the internet or the current time from the Pi’s clock could both be inputs. The information they return can then be fed to the software for the thinking stage. Finally, with the inputs read and the decisions made by the software, the computer must do something. For that, it needs outputs, things that transfer from the software to the real world. These can be as simple as an LED, a servo in a robot or even a mechanism of a 3D printer. Again, they don’t even need to be physical – for example, a piece of software can send a Tweet as its output. What type of project you create is up to you, be it just for fun or to solve a practical problem.Thus the “sense - think -act” cycles becomes the core working principle of physical computation.

Until now, this course has focused on the Raspberry Pi's software side, but now we'll get into the hardware. Take your Raspberry Pi 4, and on the top edge, you can see a row of pins. This is the most powerful feature of the Raspberry Pi 4. These pins are called General Purpose Input/Output pins or GPIO in short. The Raspberry Pi 4 is just like a brain in a jar. We have to Frankenstein our way to make it alive. The GPIO pins are a physical getaway between the Pi 4 and the outside world. Understanding the functions of these pins are crucial for anyone working with the Raspberry Pi 4.

If you count, you can see that there are 40 pins on the Raspberry Pi 4, split equally across two rows. Please download the following pinout diagram given in the resources and take a color printout, before we learn the functions of each pin. While programming on the Pi 4, if you need access to the pinout diagram right on the terminal, you can just enter “pinout” on the terminal, and you will see the pinout.

The Pi’s 40-way GPIO expansion header has many more possibilities than it has pins. Basically, there are two types of pins on the Pi, they are GPIO pins and Power Pins. The power pins include 2 5 Volts pins, 2 3.3 Volts pins, and 8 Ground pins.

The 5V pins give direct access to the 5V supply coming from your power supply. It can be used to power other 5V devices. When using these pins directly, be careful and check your voltages before making a connection because they bypass any safety features, such as the voltage regulator and fuse, which are there to protect your Pi. Bypass these with a higher voltage, and you could render your Pi inoperable. Similarly, many peripherals work on 3.3 Volt power, and thus the Pi provides that too.

Now, If you subtract the power pins from the total pins, you will get the number of GPIO pins, which is 28 in number.

These GPIOs are on-off pins that you can either use as outputs or inputs, sending or receiving logic 1 or logic 0 voltage levels. Use one anywhere a simple ON or OFF condition is required, for example, to turn on or off an LED, or to sense whether a button has been pressed.

Sometimes you need more than just switching on & off the pins. Most of the GPIO pins comes with secondary functions that allow them to interface with different kinds of devices like I2C, SPI or UART protocols. We will discuss these various protocols in detail with hardware interfacing to make you understand the concept much more. For now, just keep in mind the following things in mind before we explain the pinout

1. To implement I2C protocol, you will need 2 GPIO pins
2. To implement UART protocol, you will need 2 GPIO pins
3. To implement SPI protocol, you will need a minimum of 4 GPIO pins

Now keeping that in mind if you take a look at this functional pinout diagram, it's clear that

We have

1. 6 I2C Bus implementations

2. 5 UART Bus Implementations &

3. 6 SPI Bus implementations on the Pi 4

I have provided a detailed spreadsheet in the resources, as shown here, which will give you a much clearer idea of what all features each GPIO pin is capable of doing. You might use these buses to interface an LCD display, a temperature sensor, analogue-to-digital converter and many more sensors and actuators. The way they packed in many functionalities to each GPIO pins is by using the concept of multiplexing. It allows switching between functions supported by a GPIO. At a time, a GPIO can be configured to attach to different pins of the main processor, so that a specific function can be enabled. The only disadvantage of multiplexed GPIO is that at a time, only one function can be used.

Please remember that all of those active GPIO pins lead directly into the CPU on your Pi 4, and thus if you apply too much voltage or current to them, you can damage it. They normally work at 3.3 V and 10 mA, so precautions, such as using a resistor with an LED, should be taken. To use your Pi with a 5 V device such as an Arduino, special addons are available to protect your Pi and give you a buffered set of GPIOs to safely work with.

Summary

In this video, we have covered the following

● What is Physical Computation

● What is the working principle of Physical Computation

● A detailed overview of the GPIO pins

In the next video, we will start working on simple hardware interfacing using Python.