Hello Friends, Welcome to the video tutorial on the Raspberry Pi.

In this tutorial we will learn how to access the GPIO pins of the Raspberry pi.

Now the first question is what is GPIO. GPIO stands for General Purpose Input Output. Where external hardware can be connected. The external hardware can be the input or output device. Different input devices are switch keyboard sensors etc. and the different output devices are LED, LCD and buzzer.

Starting with the pins available on the Raspberry pi. In this tutorial I am using the Raspberry Pi 1 model B+. There are 40 pins available on the Raspberry pi as shown in Figure. Different pins have different functions. Total we have 7 different functions. First function is power source. There are two power sources 5v and 3.3 v available on the raspberry pi. 2 5v pins are shown by red dots and 2 3.3v pins are shown by the orange dots. Gnd pins are also available which are indicated by black dots. Next is the GPIO pins which are the simple input output pins. In model 1 B+ there are 26 GPIO pins. The other functions available are I2C interface , SPI interface and UART interface for serial communication.

Last one is

ID\_SD and ID\_SC PINS:  
These pins are reserved for ID EEPROM.

First experiment:

Interfacing an LED with the Raspberry Pi.

Hardware required for the experiment are

1. Breadboard : On which connections have to be made.
2. LED : Light Emitting Diode which is an output device
3. 330ohm resistor: through which LED will be connected to Raspberry pi. It is used to limit the current passing through led.

Anode of the LED is connected to pin no. 35 (which is nothing but the GPIO 19 pin) Cathode of the LED is connected to a resistor(330 ohms) which is in turn connected to GND pin on R-Pi.

Figure shows the connections of LED and Raspberry Pi.

First define the problem statement

Turn on the LED for 1 second and then turn it off for 1 second and repeat the process continuously.

After connections next part is code.

Raspbian comes preloaded with Python, the official programming language of the Raspberry Pi and IDLE 3, a Python Integrated Development Environment. And hence we can directly program the Pi using Python. Although we can even program the Pi using C language (but i will be

using Python language in this tutorial).

In order to access GPIO pins we need to use the Rpi.GPIO package which is usually present in the Python libraries. (but if you are using an R-Pi please ensure that the version of this package is greater than 0.5.10 )

########### Import the libraries ###############

Import RPi.GPIO as GPIO # module to control the GPIO pins

Import time # used for delay generation

In pin diagram you must have noticed that the board contains pins named as GPIO (that are used for interfacing input and output devices) and hence in order to refer to the R-Pi pins there exists two modes:

1. **BCM mode**: Referring the pins with the GPIO number

2. **Board mode**: Referring the pins using the IC pin numbers.

# to use Raspberry Pi in BCM mode

GPIO. setmode (GPIO.BCM)

# to use Raspberry Pi board pin numbers

GPIO. setmode (GPIO.BOARD)

# s e t up GPIO output channel i . e . on pin 35

GPIO. setup (12 , GPIO.OUT)

If you are using the BCM mode then you have to write GPIO.setup (19, GPIO.OUT)

While (1):

GPIO. output ( pin ,GPIO.HIGH) # to turn on the LED

Time.sleep(1) # delay of one second

GPIO. output ( pin ,GPIO.LOW) # to turn off the LED

Time.sleep(1) # delay of one second

#to clean up all the ports used

GPIO.cleanup()

We are ready with the code. Save it in the /home/pi directory and run it just by hitting f5. You can see LED is blinking continuously with the 1 second delay.

**Exercise**

Now that you know how to write simple programs to access GPIO pins of an R-Pi, try your hand at the following problems

1. controlling a led using a push button.

2. Generating a PWM signal on a GPIO pin to drive an LED