Experiment No:04

Aim: Study of Connectivity and configuration of Raspberry-Pi /Arduino circuit with basic peripherals, LEDS. Understanding GPIO and its use in program.

Understanding GPIO:

One powerful feature of the Raspberry Pi is the row of GPIO (general purpose input/output) pins along the top edge of the board.These pins are a physical interface between the Pi and the outside world. At the simplest level, you can think of them as switches that you can turn on or off (input) or that the Pi can turn on or off (output). Of the 40 pins, 26 are GPIO pins and the others are power or ground pins.There are eight ground pins and two +5V pins and three +3.3V pins, which are not programmable. There are other dedicated pins too. Most of the pins have alternative functions, as shown in the figure. A majority of the pins are directly connected to the SoC; so while connecting circuits or components, one should be careful to avoid wrong wiring and short circuits. It is always good to have a descriptive pinout diagram printed out for quick reference as well as a multimeter on the work desk.

GPIO#	2nd func.	Pin#	Pin#	2nd func.	GPIO#
	+3.3 V	1	2	+5 V	
2	SDA1 (I²C)	3	4	+5 V	
3	SCL1 (I²C)	5	6	GND	
4	GCLK	7	8	TXD0 (UART)	14
	GND	9	10	RXD0 (UART)	15
17	GEN0	11	12	GEN1	18
27	GEN2	13	14	GND	
22	GEN3	15	16	GEN4	23
	+3.3 V	17	18	GEN5	24
10	MOSI (SPI)	19	20	GND	
9	MISO (SPI)	21	22	GEN6	25
11	SCLK (SPI)	23	24	CE0_N (SPI)	8
	GND	25	26	CE1_N (SPI)	7
(Pi 1 Models A and B stop here)					
EEPROM	ID_SD	27	28	ID_SC	EEPROM
5	N/A	29	30	GND	
6	N/A	31	32		12
13	N/A	33	34	GND	
19	N/A	35	36	N/A	16
26	N/A	37	38	Digital IN	20
	GND	39	40	Digital OUT	21

Example :

```
import RPi.GPIO as GPIO
#for the sleep method
import time
led = 8
#set numbering mode for the program
GPIO.setmode(GPIO.BOARD)
#setup led(pin 8) as output pin
GPIO.setup(led, GPIO.OUT,initial=0)
try:
#turn on and off the led in intervals of 1 second
while(True):
#turn on, set as HIGH or 1
GPIO.output(led,GPIO.HIGH)
print("ON")
time.sleep(1)
#turn off, set as LOW or 0
GPIO.output(led, GPIO.LOW)
print("OFF")
time.sleep(1)
except KeyboardInterrupt:
#cleanup GPIO settings before exiting
GPIO.cleanup()
print("Exiting...")
```

This code will print *ON* and *OFF* alternatively on the screen, in sync with when the LED is turned on and off. The *Ctrl+C* key combination can be used to terminate the execution of the program. The *except Keyboard Interrupt:* mechanism is used to detect the *Ctrl+C* keypress. The Sleep method will make the process wait for the given amount of time, which is one second here.

Conclusion:

Question :

1) Explain the Pin configuration of Raspberry-pi.

2) Write the features of Raspberry pi.

3) Write the code for alarm in Python.