DPU

Experiment no 04

Aim:- Study of connectivety and configuration of Raspberry-Pi / Arduino circuits with basic peripherals, LED's, understanding GPIO and its use in program.

Understanding GPIO :-

One powerfull feature of Raspberry-Pi is the row of GPIO pins along the top edge of the board. At the simplest level you can think of them as switches that you can turn on or off or that the Pi can turn on or off. 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 +3.3 V pins which are not programmable. There are other dedicated pins too. Most of the pins have alternative functions, as shown in the figure.

To avoid wrong wiring and short circuits It is always good to have a descriptive print out diagram printed out for quick refeference as well as multimeter on the work desk.

Example :-

```
import RPi.GPIO as GPIO
# for the sleep method.
import time
led = 8
```

```
# Set the numbering mode for the program.
GPIO.setmode(GPIO.BOARD)
# Set up 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 existing
GPIO.cleanup()
print("Exiting")
```

This code will print ON and OFF alternatively on the screen, in sync with when the LED is turned ON or OFF. The Ctrl+C key combination can be used to terminate the execution of the program. The except keyboard Interrupts Mechanism is used to detect the Ctrl+C keypress. The sleep method will make the process wait for the

DPU

given amount of time., which is one second here.

Conclusion.

DPU

Experiment No 05

Title:

Write a program using Arduino to control LED. (one or more ON/OFF).

Aim:-

Develope an application using an Arduino to control LED.

Prerequisites:

> Programming in python / c++.
> Basics and logical concerns of LED lights.

Objectives:

> To earn basic concepts of Arduino.
> Learning to work with a Arduino.
> To program a LED light simulation on the kit.
> To observe the hardware interfacing of the Arduino with using suitable program and interfaces.

* Theorey:
* Aurdiuno circuit with an LED, and a button
* working:-

> To build the circuit you will need those components:-

> Arduino board (any board, if you don't have Uno can easily adapt by finding corresponding pins).
> BreadBoard.
> LED any colour.
> Push button.

DPU

> 220 ohm resistor for the LED. If you don't have this specific value, any resistor from 330 to 1 k ohm will do.
> 10.k ohm resistor for the push button. If you don't have, you can go until 20k -50k ohm.
> A bunch of male wires (including, if possible black, red and other colors).
> First, make sure to power off your Arduino -remove any USB cable.
> Plug the LED. You can notice that the LED has leg shorter than the other. plug this shorter leg to the ground (blue line here) of the circuit.
> Connect the longer leg of LED to digital pin (here pin no 8, you can change it). Add a 220 ohm registor in between to limit the current going through LED.
> Add the push button to the bread board, like in the picture.
> Add red wire between another leg of the button and VCC (5V).
> finally connect a leg of button (same side as the pull down resistor) to a digital pin. (here 7).

* Conclusion :-