ASSIGNMENT NUMBER: B1

TITLE Connectivity of Raspberry Pi /Beagle board circuit with InfraRed (IR) sensor.
PROBLEM STATEMENT /DEFINITION Understanding the connectivity of Raspberry-Pi /Beagle board circuit with IR sensor. Write an application to detect obstacle and notify user using LEDs.

OBJECTIVE

 Understanding the connectivity of Raspberry Pi /Beagle board circuit with IR sensor.

S/W PACKAGES AND HARDWARE APPARATUS USED Raspberry pi board/ BBB, IR sensor, LED Raspbian (OS)

Aim: Connectivity of Raspberry Pi /Beagle board circuit with IR sensor to detect obstacle and notify user using LEDs.

Pre-requisite:

Basic knowledge of GPIO of Raspberry pi/BBB

Basic knowledge of Python programming.

Working and connections of sensors.

Learning Objectives:

 Understanding the connectivity of Raspberry Pi /Beagle board circuit with IR sensor.

Learning Outcomes:

The students will be able to

 To interface IR sensor to Raspberry pi.

 Detect the obstacle with IR sensor.

 Can perform actuation.

H/W AND S/W Requirements:

Raspberry Pi/Beagle board Development Boards

PC / Monitor/Keyboard

IR (Infrared) Sensor, 1 LED, 1 Resistor (330 Ω)

Few jumper cables,1 Breadboard

Raspbian (OS), Debian LINUX and Python

Theory:

Introduction:

The Raspberry Pi is a series of credit card-sized single-board computers developed in the

United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. It is a capable little computer which can be used in electronics projects, and for many of the things that your desktop PC does, like spreadsheets, word processing, browsing the internet, and playing games. It also plays high-definition video.

The Raspberry Pi is open hardware, with the exception of the primary chip on the Raspberry Pi, the Broadcom SoC (System on a Chip), which runs many of the main components of the board–CPU, graphics, memory, the USB controller, etc. Many of the projects made with a

Raspberry Pi are open and well-documented as well and are things you can build and modify yourself.

The Raspberry Pi was designed for the Linux operating system, and many Linux distributions now have a version optimized for the Raspberry Pi.

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 pinsand the others are power or ground pins (plus two ID EEPROM pins which you should not play with unless you know your stuff!)

You can program the pins to interact in amazing ways with the real world. Inputs don't have to come from a physical switch; it could be input from a sensor or a signal from another computer or device, for example. The output can also do anything, from turning on an LED to sending a signal or data to another device. If the Raspberry Pi is on a network, you can control devices that are attached to it from anywhere and those devices can send data back. Connectivity and control of physical devices over the internet is a powerful and exciting thing, and the Raspberry Pi is ideal for this.

Raspberry Pi Board with GPIO:

Technical Specification:

 Broadcom BCM2837 64bit ARMv7 Quad Core Processor powered Single Board

 Computer running at 1.2GHz

 1GB RAM

 BCM43143 WiFi on board

 Bluetooth Low Energy (BLE) on board

 40pin extended GPIO

 4 x USB 2 ports

 4 pole Stereo output and Composite video port

 Full size HDMI

 CSI camera port for connecting the Raspberry Pi camera

 DSI display port for connecting the Raspberry Pi touch screen display

 Micro SD port for loading your operating system and storing data

 Upgraded switched Micro USB power source (now supports up to 2.4 Amps)

 Expected to have the same form factor has the Pi 2 Model B, however the LEDs will

Change position

InfraRed (IR) Sensor:

IR (Infrared) Sensor works by emitting infrared signal/radiation and receiving of the signal when the signal bounces back from any obstacle. In other words, the IR Sensor works by continuously sending signal (in a direction) and continuously receive signal, if comes back by bouncing on any obstacle in the way.

Components: IR Sensor

1. Emitter: This component continuously emits the infrared signal

2. Receiver: It waits for the signal which is bounced back by obstacle

3. Indicator: On board LED to signal if obstacle is deducted by the sensor

4. Output: Could be used as Input for further processing of the signal

5. Ground: Ground/Negative point of the circuit

6. Voltage: Input 3.3V

IR Sensor has 3 pins, viz VCC, GND and OUT. We will use GPIO 17 (do not get confused with pin number 17) for receiving input from the sensor.

Connecting IR Sensor

1. Connect GPIO 17 from the Raspberry Pi to Breadboard

2. Connect OUT pin of the sensor with the Breadboard This will send input received from sensor to GPIO 17, which will be processed further.

3. Connect GND (any pin from board will work) with negative line on left side of the breadboard

4. Connect GND of the IR Sensor to Breadboard

5. Connect GND from Step 3 to Breadboard

6. Connect VCC of the IR Sensor to Breadboard

7. Connect 3v3 (Pin #1) to positive line on left side of the breadboard

8. Connect 3v3 (connected in Step 7) to the Breadboard

Connecting LED

Objective is to turn on the LED when obstacle is detected.

1. Connect GPIO 4 from the board to the Breadboard

2. Connect positive point of the LED (longer pin of the LED) to the Breadboard

3. Connect negative point of the LED (smaller pin of the LED) to the Breadboard

4. Use resistor (330 Ω) to connect negative to the negative point of the LED

Python code to detect obstacle

```
from gpiozero import LED
from signal import pause
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
LED_PIN = 27
IR_PIN = 17
indicator = LED(LED_PIN)
GPIO.setup(IR_PIN, GPIO.IN)
count = 1
while True:
got_something = GPIO.input(IR_PIN)
if got_something:
indicator.on()
print("{:>3} Got something".format(count))
else:
indicator.off()
print("{:>3} Nothing detected".format(count))
count += 1
time.sleep(0.2)
```

Conclusion: Hence Experiment was carried out successfully.