EXPERIMENT N0 3

Experiment Title:

Understanding and connectivity of Raspberry-Pi /Beagle board with camera. Write an application to capture and store the image

Theory:

The **Raspberry Pi** is a versatile single-board computer widely used in **IoT and embedded systems**. One of its essential features is the ability to **connect with external devices via physical (wired) interfaces**. These wired connections are crucial for data transmission, device control, and integration with hardware components.

Types of Wired Connectivity in Raspberry Pi:

1. Power Supply Connection:

- The Raspberry Pi is powered using a **USB Type-C (for Raspberry Pi 4)** or **micro-USB (for older models)** cable connected to a 5V power adapter.
- Proper and stable power is essential for functioning and peripheral operation.

2. HDMI Port (Display Connection):

- Raspberry Pi includes one or two **micro-HDMI ports** (depending on the model) for connecting to a **monitor or TV** using an HDMI cable.
- Enables GUI-based development and debugging.

3. Ethernet Port (Wired Internet):

- Raspberry Pi (excluding Zero models) has an **RJ-45 Ethernet port** for wired network connectivity.
- Provides reliable internet access for headless setups, SSH, software updates, and data transfer.

4. USB Ports (Peripheral Connection):

- The board has multiple **USB 2.0 and USB 3.0** ports.
- These are used to connect **keyboard, mouse, USB storage, USB-to-serial devices, Wi-Fi dongles** (for older models), or other peripherals.

5. GPIO Pins (General Purpose Input/Output):

- Raspberry Pi provides a **40-pin GPIO header** that supports both power and data signal lines.
- Through these pins, it connects **wired sensors, LEDs, relays, motors**, and **other embedded components**.
- GPIO supports various wired protocols:

- **I2C (Inter-Integrated Circuit):** For connecting sensors and displays over two wires (SDA and SCL).
- **SPI (Serial Peripheral Interface):** High-speed data exchange with ADCs, displays, etc.
- **UART (Universal Asynchronous Receiver/Transmitter):** For serial communication with modules like GSM, GPS, and microcontrollers (e.g., Arduino).

6. Camera and Display Interfaces:

- **CSI (Camera Serial Interface):** For connecting official Raspberry Pi camera module using a ribbon cable.
- **DSI (Display Serial Interface):** For connecting Raspberry Pi touch displays via flat ribbon cable.

7. Audio Jack:

- Older models include a **3.5mm audio jack** for analog audio output, which can also function as a composite video output via AV cable.

Importance of Wired Connectivity:

- **Stable and fast communication**, especially for data-intensive tasks.
- **Low latency** compared to wireless connections.
- **Secure and interference-free** data transfer.
- Essential in **industrial IoT applications** where reliable communication is critical.

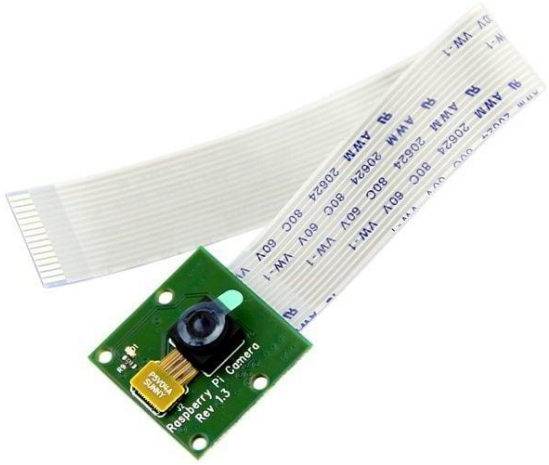

Fig - Pi Camera Module (v1.3)

The pi Camera module is a camera that can be used to take pictures and high definition video.

Raspberry Pi Board has CSI (Camera Serial Interface) interface to which we can attach the PiCamera module directly.

This Pi Camera module can attach to the Raspberry Pi's CSI port using a 15-pin ribbon cable.

Features of Pi Camera

Here, we have used Pi camera v1.3. Its features are listed below,

- Resolution – 5 MP
- HD Video recording – 1080p @30fps, 720p @60fps, 960p @45fps and so on.
- It Can capture wide, still (motionless) images of a resolution 2592x1944 pixels
- CSI Interface enabled.

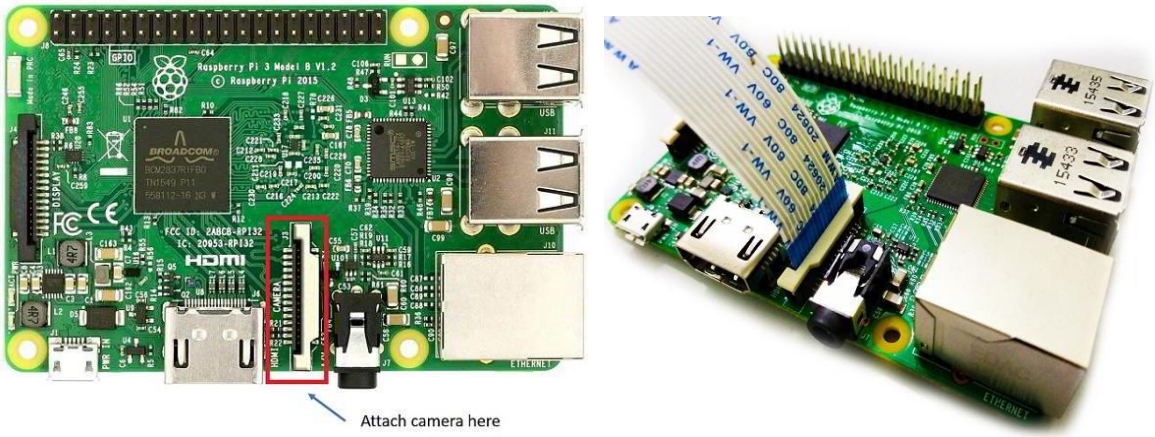

Fig – connection of camera to raspberry pi

Now, we can use Pi Camera for capturing images and videos using Raspberry Pi.

Before using Pi Camera, we need to enable camera for its working.

How to Enable Camera functionality on Raspberry Pi

For enabling the camera in Raspberry Pi, open the raspberry pi configuration using the following command,

sudo raspi-config

then select **Interfacing options** in which select the **camera** option to enable its functionality.

reboot Raspberry Pi.

Now we can access the camera on Raspberry Pi.

Now we can capture images and videos using Pi Camera on Raspberry Pi.

Source code

Pi Camera Python Program for Image Capture

```
import picamera
from time import sleep
#create object for PiCamera class
camera = picamera.PiCamera()
#set resolution
camera.resolution = (1024, 768)
camera.brightness = 60
camera.start_preview()
#add text on image
camera.annotate_text = 'Hi Pi User'
sleep(5)
#store image
camera.capture('image1.jpeg')
camera.stop_preview()
```

Functions Used

To use picamera python based library we have to include it in our program as given below

```
import picamera
```

This picamera library has PiCamera class for the camera module. So, we have to create an object for **PiCamera** class.

PiCamera Class

To use Pi Camera in Python on Raspberry Pi, we can use PiCamera class which has different APIs for camera functionality. We need to create object for PiCamera class.

E.g. Camera = picamera.PiCamera()

The above PiCamera class has different member variables and functions which we can access by simply inserting a dot (.) in between object name and member name.

E.g. Camera.resolution = (1080, 648)

capture()

It is used to capture images using Pi Camera.

E.g. Camera.capture("/home/pi/image.jpeg")

The capture() function has different parameters which we can pass for different operations like resize, format, use_video_port, etc.

E.g. Camera.capture("/home/pi/image.jpeg", resize=(720, 480))

resolution= (width,height)

It sets the resolution of the camera at which image captures, video records, and previews will display. The resolution can be specified as **(width, height)** tuple, as a string formatted **WIDTHxHEIGHT**, or as a string containing commonly recognized display resolution names e.g. "HD", "VGA", "1080p", etc.

E.g.

Camera.resolution = (720, 480)

Camera.resolution = "720 x 480"

Camera.resolution = "720p"

Camera.resolution = "HD"

Annotate_text = "Text"

It is used to add text on images, videos, etc.

E.g. Camera.annotate_text = "Hi Pi User"

start_preview()

It displays the preview overlay of the default or specified resolution.

E.g. Camera.start_preview()

stop_preview()

It is used to close the preview overlay.

E.g. Camera.stop_preview()

Conclusion:

Wired connectivity in Raspberry Pi allows for stable and secure communication with various peripherals and networks. Understanding the use of physical ports like GPIO, USB, Ethernet, HDMI, and serial interfaces is vital for building reliable and efficient embedded and IoT systems.