

**Project made by: (Group 6)**

B.Revanth

B.Sumanth

B.Venkataramana Reddy

B.Karthik Reddy

**Project Aim:**

The aim of this project is to design a QR and barcode scanning system using a Raspberry Pi instead of a Raspberry Pi Pico, along with a camera module. The system will take images of QR and barcode data, decode them, and then show the results through a web interface, making the scanned data accessible via any internet-enabled device.

**Problem Statement:**

Due to increased usage of QR codes and barcodes in transactions, tracking products, and sharing information in everyday life, low-cost scanning systems for such codes are very important. Most of the existing solutions are either proprietary or costly.

This project aims to develop an accessible, cost-effective, and customizable QR and barcode scanner system using open-source hardware, Raspberry Pi, and open-source software that can be integrated into different applications, such as inventory management systems, authentication systems, or information sharing platforms.

**Solution:**

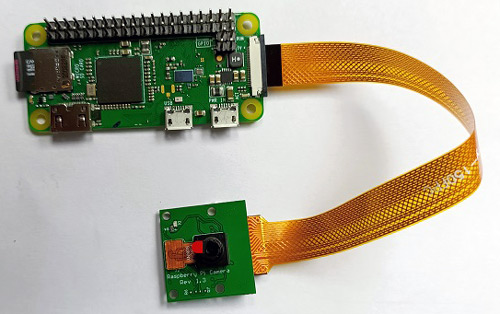
This project would aim at solving the problem by developing a low-cost, flexible, and scalable QR and barcode scanner implemented on a Raspberry Pi. The system will:

* Capture Images: By means of a Camera Module attached to the Raspberry Pi.
* Decode QR code and barcodes: Use of the pyzbar library to decode the information from the captured images.
* It should also include an accompanying web interface with scanned QR or barcode data, accessible via any internet-enabled device.

This makes the system prove to be an easy and efficient tool for anyone who might want to actually implement barcode or QR code scanning in virtually any application.

**Project Design Requirements:**

* ***Hardware:***
  + Raspberry Pi model: Should be a Raspberry Pi 3, 4 or greater with Wi-Fi functionality.
  + Camera Module: A camera module compatible with the Raspberry Pi, such as the official Raspberry Pi Camera Module.
  + Wi-Fi Connection: Raspberry Pi must be connected to a local Wi-Fi network for web interface access.
* ***Software:***
  + OS: Raspberry Pi OS - or any OS compatible.
* ***Libraries:***
  + pyzbar: For decoding QR codes and barcodes.
  + picamera: Interacts with the camera module.
  + Pillow (Python Imaging Library): Handling image data.
  + Web Server: An HTTP server, usually minimalistic, hosting the web interface.
  + Web Interface: Output decoded QR/barcode data on a webpage accessible from any internet-enabled device.



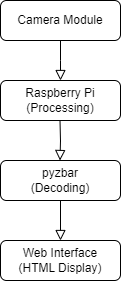
**System Features:**

* Capture the image through the camera module.
* Automatically decode from the QR/barcode with pyzbar. Display the decoded data in real-time onto an HTML page hosted online. It should allow users to access the scanner's output from any browser connected to the internet.

**Project Architecture:**

1. Camera Module: The camera captures an image of the QR code or barcode, which is processed by the Raspberry Pi.
2. Image Capture and Decoding: The captured image is then passed on to the pyzbar library that decodes any QR/barcode present in the image.
3. Web Server: A simple socket-based server that delivers any browser over the internet with an HTML page containing the scanned data from the QR/barcode.
4. Data Display: The decrypted message is shown on the webpage.

**Block Diagram:**



**Components Working Principles/Functionality:**

1. Raspberry Pi: The core component which processes the QR code information and allows a web server, it actually makes a connection with the camera module and is responsible for the image processing and decoding. It works through the operational mechanism: runs a Python script that retrieves images, reads QR codes, and releases information over the web interface.
2. Camera Module: Picture Capture of QR codes. The module's camera sensor captures the image, and then it sends that captured image to Raspberry Pi for further processing.
3. PyZbar Library: Decodes captured image QR codes. Scans the visual, detects QR codes, and retrieves information (text or URL).
4. Pillow Library: It processes images into recognizable formats to decode. Open, resize, or adjust the captured image so that it can be decoded by PyZbar.
5. Web Server (Using the Socket Library): Hosts the web interface to display decoded QR code data. Monitors for incoming connections and subsequently transmits HTML pages that exhibit the QR code information.
6. Built-in Wi-Fi Module in Raspberry Pi: It connects the Raspberry Pi to the internet or local network. It enables a network linkage that permits distant access to the web interface.
7. Power Source: Powers the Raspberry Pi and connected peripherals. It supplies the required voltage and current supply to Pi and the camera module for working.

**Introduction to the QR Code Reader Program Using Micropython for Raspberry Pi: (Check main.py file)**

***Utilized Libraries:***

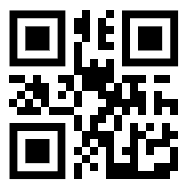
* Network: This library allows the Raspberry Pi to connect to a Wi-Fi network. Using the network.WLAN() function, an internet connection for the Raspberry Pi can be established. This is very important to make the web server communicate with its clients.
* Socket: This is used to set up the web server. It is built to listen and respond to requests coming from clients, including web browsers, and sends back data, like the decoded QR code, in response.
* picamera: The picamera library offers access to the camera module of the Raspberry Pi. This is used to capture images that are then analyzed to detect QR codes.
* pyzbar: It decodes QR codes from images captured using the camera from the phone's library. It scans the image, identifies any QR codes possibly present in the viewfinder, and then extracts data stored in them.
* PIL (Python Imaging Library): We handle image data using PIL. The camera will capture the image after which the library opens and processes it to decode the QR code.

***Main Functions:***

* Wi-Fi connectivity: The function connect\_to\_wifi() connects the Raspberry Pi to the local Wi-Fi network. This is an important process for allowing the Raspberry Pi to communicate with clients over the internet.
* Web Server: The start\_web\_server() function sets up the web server to wait for incoming requests originating from web browsers, gets an image from the server, searches for potentially present QR codes within that image, and returns the result as a reply to the client through a web page.
* QR Code Scanning: The capture\_qr\_code() function captures an image with the camera, and any QR code contained within that image is decoded using the library pyzbar. It returns the data in the QR code.
* Making HTML: The generate\_html() function develops a very simplistic web page with the data decoded from the detected QR code. It reports if nothing can be found with a message saying "No QR code detected."

**Project output:**

***Example QR code scanned:***

******

***Hosted website output:***

