Bonrix Software Systems

DQ11 - Dynamic QR Code Display
Desktop Software Integration Guide

🌟 Introducing the Dynamic QR Displayer (Visual Display Unit) - Item Code-DQ11 🌟

Take charge of your payments with our cutting-edge 3.5 inch USB Dynamic QR Displayer! Display the QR Code with the amount in Rupees for every new transaction, all programmable with any software language.

💼 Specifications:

- *Processor*: Powerful 32-bits ARM for swift processing.
- *Display*: Vibrant 3.5" TFT LCD, 320*480 pixels that showcase clear, bright QR codes.
- *USB Cable*: Long 1.2m, Type A USB plug for easy connectivity.
- *Communication*: Reliable USB CDC Serial port for smooth data transfer.
- *Weight*: Light and portable at only 170g±10g.
- *Display Type*: Binary Image for crystal-clear QR representation.
- *Compatibility*: Fully compatible with Paytm DQR Display for seamless transactions.

💡 Features:

- *Dynamic QR Code*: Effortlessly show QR with the exact amount for each order.
- *Universal Compatibility*: Works with Paytm, PhonePe, BharatPe, GPay, and any bank QR.

- *Merchant Empowerment*: Full control over transaction amounts – no manual input by customers.
- *Real-Time Settlement*: Easy reconciliation with your billing system.
- *Instant Status*: Get payment confirmations in real-time across any UPI app.
- *Configurability*: Simple setup for any UPI; effortlessly change or upgrade as needed.

🏪 Perfect for:

- Grocery Stores
- Retail Outlets
- Various Ticketing Counters
- Hospitals, both Public and Private
- Billing Counters at LIC and Post Offices
- Fuel Stations
- Eateries and Hospitality Venues

✅ Advantages:

- Precise control over payments.
- Simplified customer experience with no amount entry needed.
- Unique QR for each order for error-free transactions.
- Synchronize your payment system with UPI settlements effortlessly.

📱 Kickstart your journey with our complimentary Desktop Software and Android App, tailor-made for BHIM UPI Payment Collection.

🚀 Elevate your transaction system with the Dynamic QR Displayer – your partner in the digital payment era!

#DynamicQRDisplayer #CashlessRevolution #UPIPayments #TechInRetail #SmartPayments

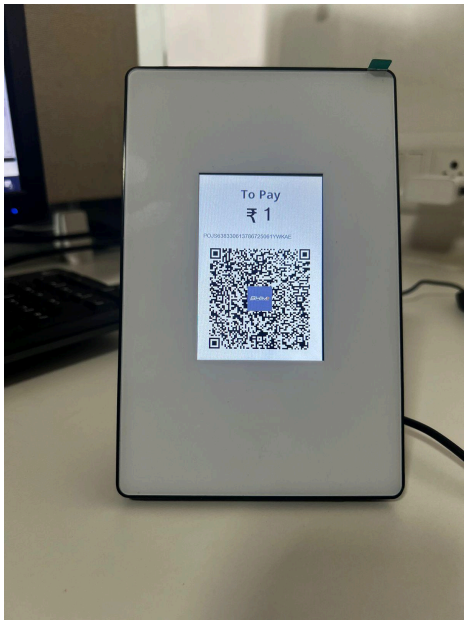
🔗 **Visit us at: https://www.bonrix.co.in**

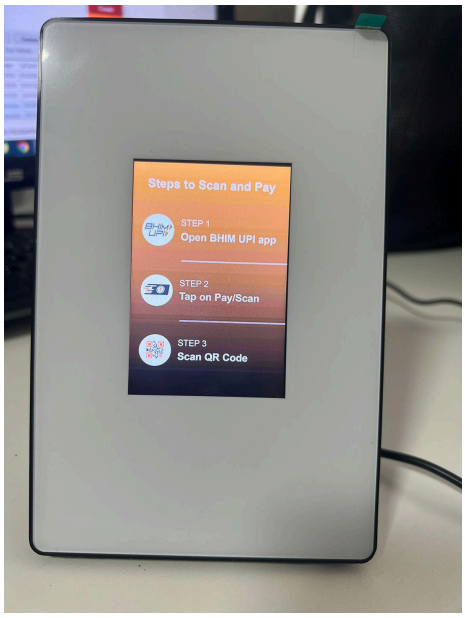
👍 **Like us on Facebook: https://www.facebook.com/BonrixSoftwareSystems**

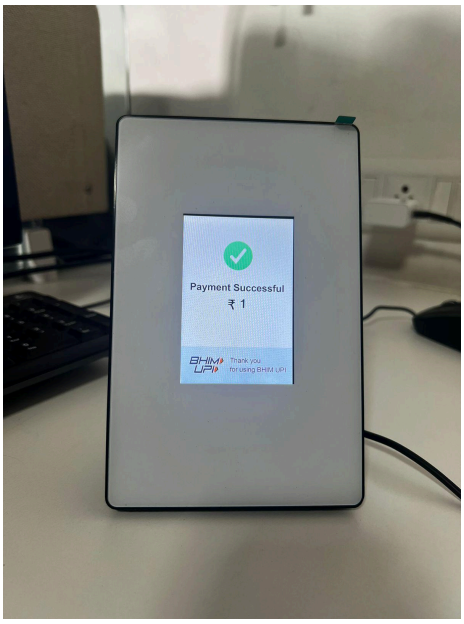
📞 **Contact us: +91-7624045500 | +91-9429045500 | +91-6352445500**

Bonrix Software Systems, where ideas meet reality. Join us in Ahmedabad (Gujarat) - INDIA for a demo!


Steps to Scan and Pay
STEP 1
Open BHIM UPI app
STEP 2
Tap on Pay/Scan
STEP 3
Scan QR Code


To Pay
₹ 1


Steps to Scan and Pay
STEP 1
Open BHIM UPI app
STEP 2
Tap on Pay/Scan
STEP 3
Scan QR Code


Payment Successful
₹ 1
Thank you
for using BHIM UPI

INTRODUCING
THE DYNAMIC QR DISPLAYER
(VISUAL DISPLAY UNIT)

ITEM CODE-DQ11

Take charge of your payments with our cutting-edge **3.5 inch USB Dynamic QR Displayer !**
Display the QR Code with the amount in Rupees for every new transaction, all programmable with any software language.

FEATURES

Dynamic QR Code, Configurability Universal Compatibility, Instant Status Merchant Empowerment, Real-Time Settlement

SPECIFICATIONS

- **PROCESSOR :** Powerful 32-bits ARM for swift processing.
- **DISPLAY :** Vibrant 3.5" TFT LCD, 320x480 pixels that showcaseclear, bright QR codes.
- **USB CABLE :** Long 1.2m, Type A USB plug for easy connectivity.
- **COMMUNICATION :** Reliable USB CDC Serial port for smooth data transfer.
- **WEIGHT :** Light and portable at only 170g±10g.
- **DISPLAY TYPE :** Binary Image for crystal-clear QR representation.
- **COMPATIBILITY :** Fully compatible with Paytm DQR Display for sea

FEATURES

1. **Precise control over payments.**
2. **Simplified customer experience with no amount entry needed.**
3. **Unique QR for each order for error-free transactions.**
4. **Synchronize your payment system with UPI**
5. **settlments effortlessly.**

- Kickstart your journey with our complimentary Desktop Software and Android App, tailor-made for **BHIM UPI** Payment Collection.
- Elevate your transaction system with the **Dynamic QR Displayer** your partner in the digital payment era!

www.bonrix.co.in facebook.com/BonrixSoftwareSystems

BONRIX SOFTWARE SYSTEMS
Where Ideas Map to Reality

+91 76240 45500 | +91 94290 45500 | +91 63524 45500
Bonrix Software Systems, Ahmedabad (Gujarat) - INDIA

Specification and Details:

Connectivity: USB

Display Type: Image Display

Image Size: 320 X 480 Pixels

USB - UART - Serial Port Terminal Tools for Testing:

1. PuTTY

2. Tera Term

3. RealTerm

4. Minicom

5. HyperTerminal

6. Screen (Linux command)

7. Picocom

8. CuteCom

9. GtkTerm

10. Serial Port Monitor by Eltima

11. SSCOM

Steps for Testing DQ-11 DQR Display with Tools:

Step 1: Download SSCOM -Serial port Terminal Software

Download Page:

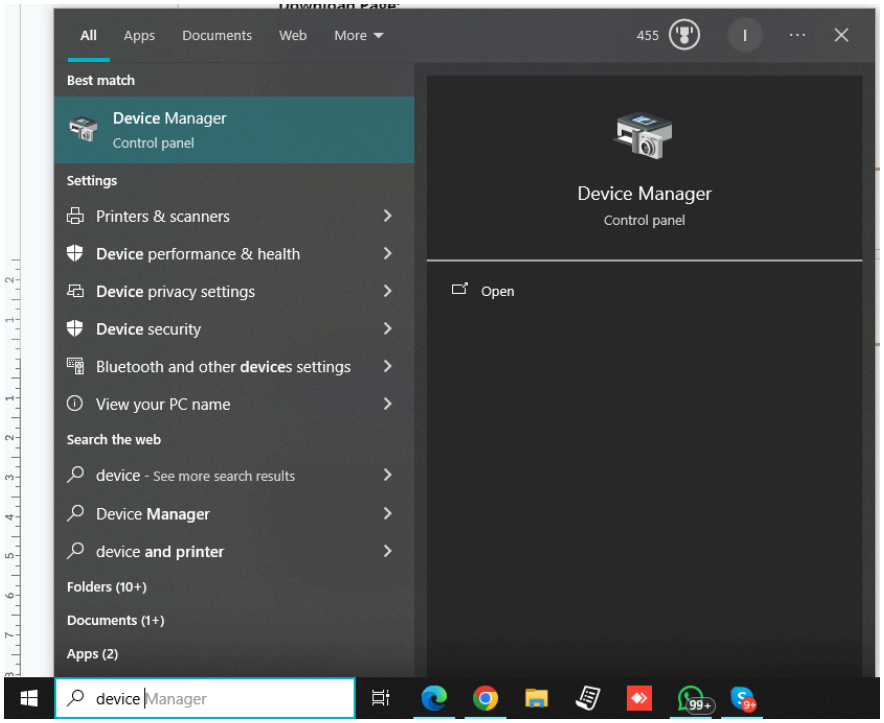
http://www.viewprotech.com/wap/index.php?ac=article&at=read&did=350

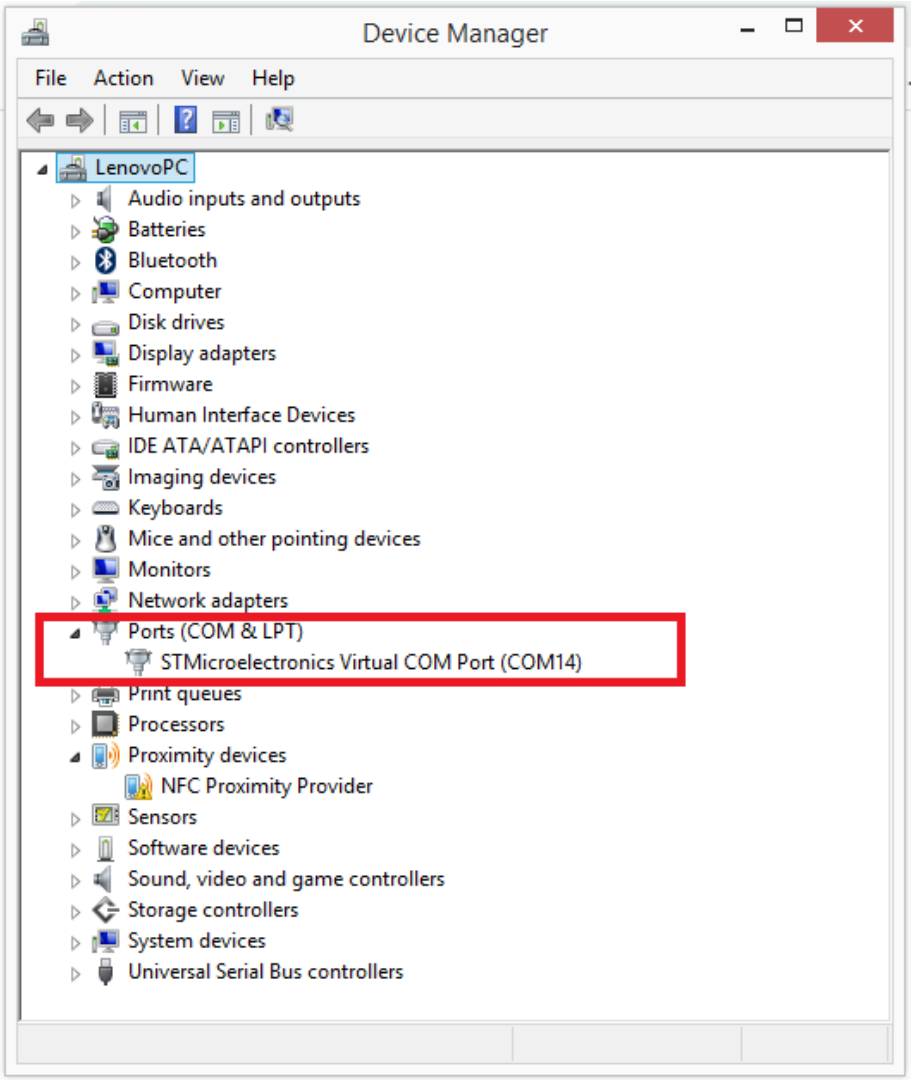
Download Link:

http://www.viewprotech.com/upfile/2019/06/20190603155148_810.rar

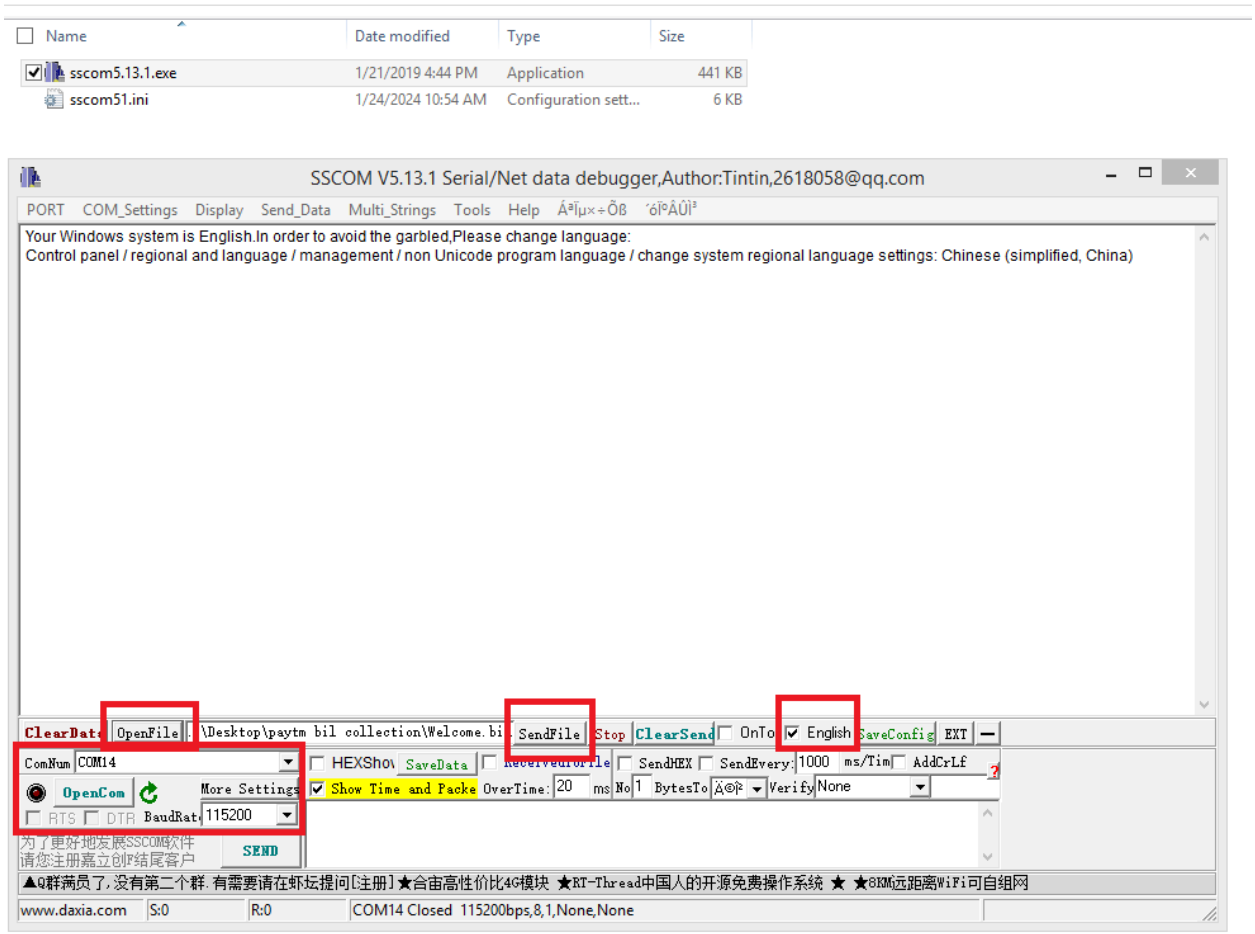
Step 2: Connect DQ11 Dynamic QR Code Device on USB Port

Step 3: Search your COM Port from Device Manager





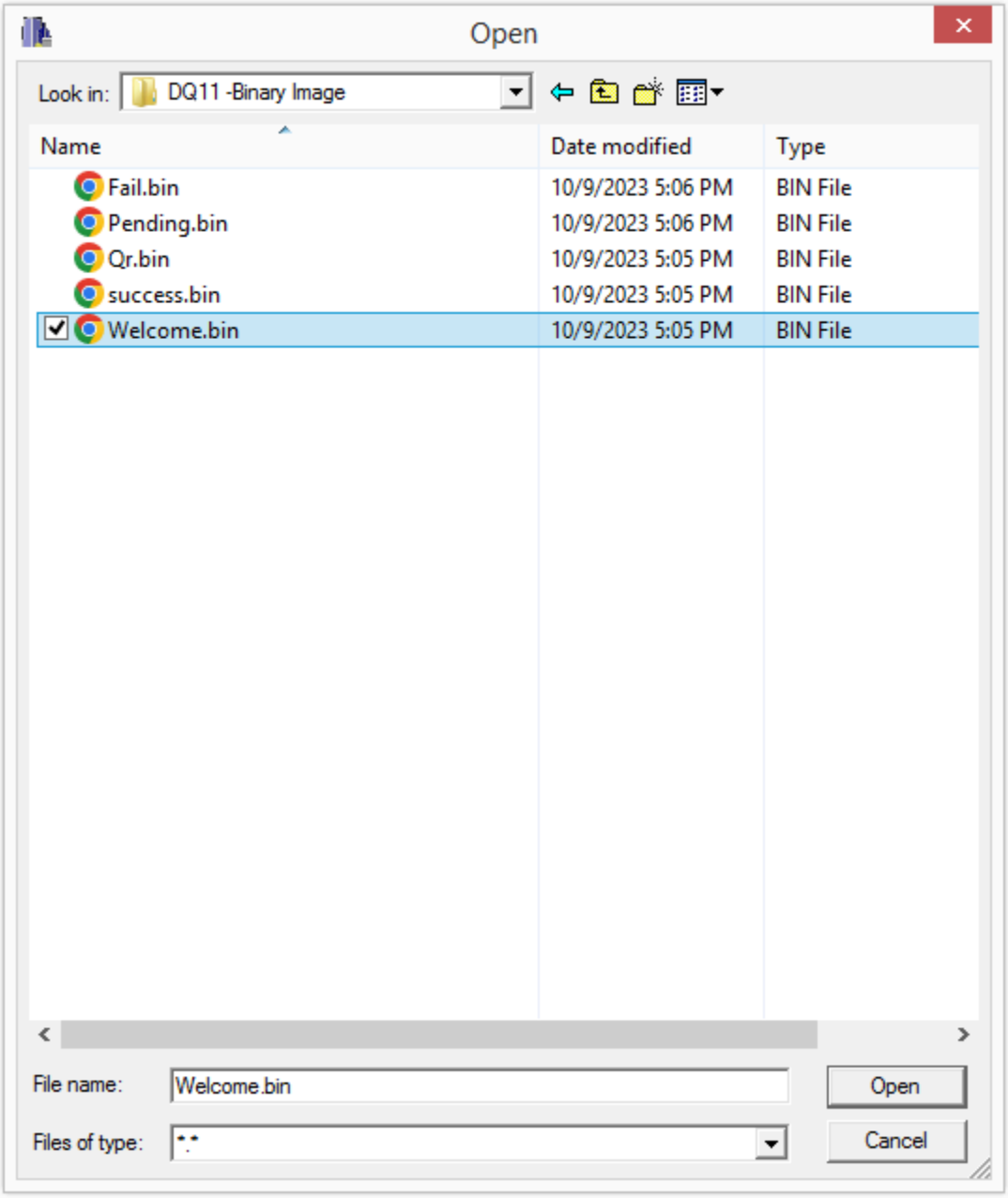
Step 3: Connect COM Port in SSCOM Serial Terminal and Set your language - English



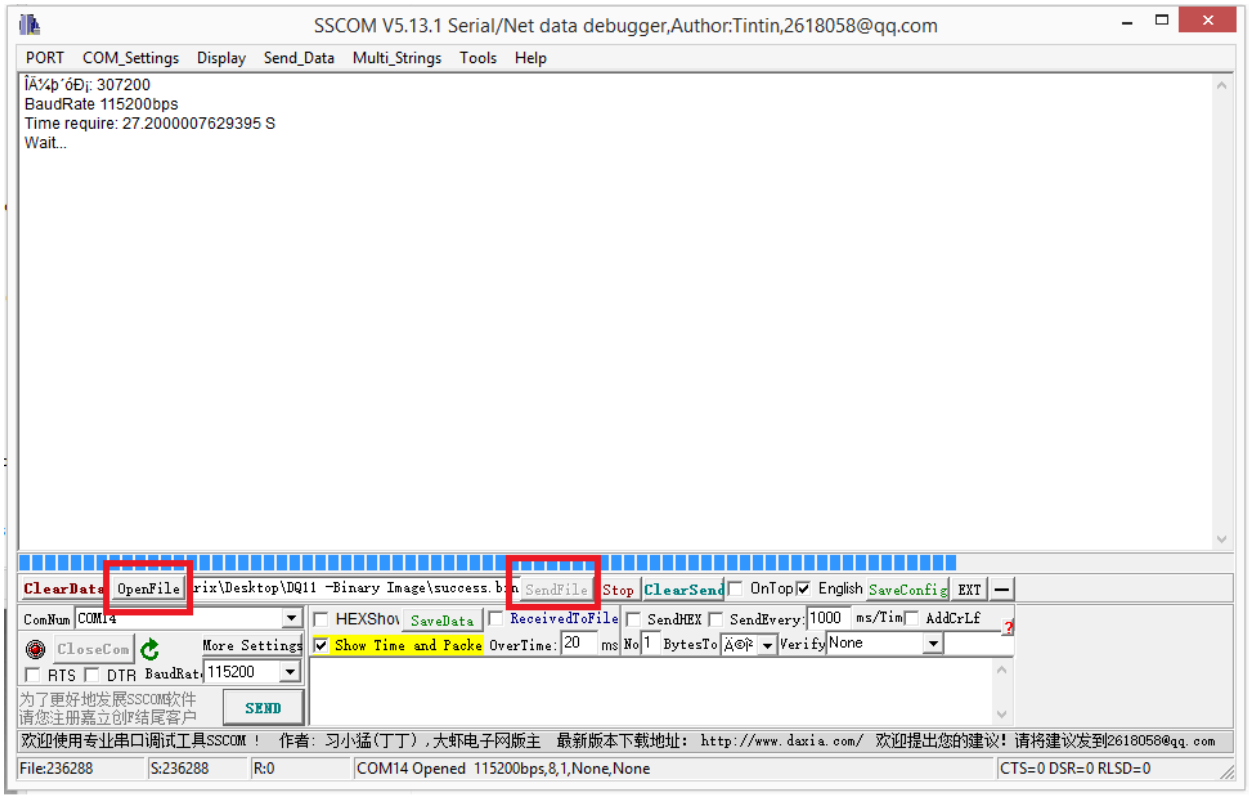
Step 4: Select your Binary image file that is 300KB File of RGB565 Format 320 X 480 X2 Bytes per pixel = 3,07,200 bytes

Download link:

https://download.rechargegrid.in/download/DQ11-BinaryImage.zip



Step 5: Click On Send file Button and Check your DQ11 Display for Image Draw



Steps for Testing DQ-11 with Programming Language:

Step 1: Take any image any image in JPEG/BMP/PNG Format

Step 2: Crop it to Pixel Size of 320 width X 480 Height size

Step 3: Convert or Save this image in BMP 24 bit Format

Step 4: Use available Image conversion algorithm for converting BMP 24 bit Format to RGB565 Binary Image format

Output bin file size - 320 X 480 X2 Bytes per pixel = 3,07,200 bytes

Step 5: You can use Canvas Drawing library to draw QR, write test using Font or include any clip art on canvas and then save it as BMP24 bit format and then convert it into RGB565 Format

Step 6: Now Send this bin file or ByteBuffer of size 300KB to Serial Port - COM3 / COM4 etc. Check it from the Device Manager.

Example Code For C#:

```
using System.IO;

using System.IO.Ports;

class Program
{
    static void Main()
    {
        string filePath = @"path_to_your_file\Compressed_success.bin"; // Adjust the file path accordingly
        using (SerialPort serialPort = new SerialPort("COM4", 921600))
        {
            serialPort.Open();
            using (FileStream fs = File.OpenRead(filePath))
            {
                byte[] buffer = new byte[fs.Length]; // Read the entire file if it's not too large
                fs.Read(buffer, 0, buffer.Length);
                serialPort.Write(buffer, 0, buffer.Length);
            }
            serialPort.Close();
        }
    }
}
```

csharp Copy code

```csharp
using System.IO;
using System.IO.Ports;

class Program
{
    static void Main()
    {
        string filePath = @"path_to_your_file\Compressed_success.bin";
        using (SerialPort serialPort = new SerialPort("COM4", 921600))
        {
            serialPort.Open();
            using (FileStream fs = File.OpenRead(filePath))
            {
                byte[] buffer = new byte[fs.Length]; // Read the entire
                fs.Read(buffer, 0, buffer.Length);
                serialPort.Write(buffer, 0, buffer.Length);
            }
            serialPort.Close();
        }
    }
}
```

Example Code in Java:

```java
import com.fazecast.jSerialComm.SerialPort;

import java.io.FileInputStream;
import java.io.IOException;

public class SerialPortFileTransfer {
    public static void main(String[] args) {
        String filePath = "path_to_your_file/Compressed_success.bin"; // Adjust the file path accordingly
        SerialPort serialPort = SerialPort.getCommPort("COM4");
        serialPort.setBaudRate(921600);

        try {
            serialPort.openPort();
            try (FileInputStream fileInputStream = new FileInputStream(filePath)) {
                byte[] buffer = new byte[fileInputStream.available()];
                fileInputStream.read(buffer);

                serialPort.writeBytes(buffer, buffer.length);
            } finally {
                serialPort.closePort();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

```java
import com.fazecast.jSerialComm.SerialPort;

import java.io.FileInputStream;
import java.io.IOException;

public class SerialPortFileTransfer {
    public static void main(String[] args) {
        String filePath = "path_to_your_file/Compressed_success.bin"; /
        SerialPort serialPort = SerialPort.getCommPort("COM4");
        serialPort.setBaudRate(921600);

        try {
            serialPort.openPort();
            try (FileInputStream fileInputStream = new FileInputStream(
                byte[] buffer = new byte[fileInputStream.available()];
                fileInputStream.read(buffer);

                serialPort.writeBytes(buffer, buffer.length);
            } finally {
                serialPort.closePort();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Example Code in Python:

```
import serial

def send_file_over_serial(port, baud_rate, file_path):
    try:
        # Open the serial port
        with serial.Serial(port, baud_rate, timeout=1) as ser:
            # Open the binary file
            with open(file_path, 'rb') as file:
                # Read the entire file
                data = file.read()
                # Send data over serial
                ser.write(data)
                print("File sent successfully.")
    except serial.SerialException as e:
        print(f"Error opening serial port: {e}")
    except FileNotFoundError:
        print("The specified file was not found.")
    except Exception as e:
        print(f"An error occurred: {e}")

# Example usage
file_path = "path_to_your_file/Compressed_success.bin"  # Adjust the file path accordingly
send_file_over_serial("COM4", 921600, file_path)
```

python Copy code

```python
import serial

def send_file_over_serial(port, baud_rate, file_path):
    try:
        # Open the serial port
        with serial.Serial(port, baud_rate, timeout=1) as ser:
            # Open the binary file
            with open(file_path, 'rb') as file:
                # Read the entire file
                data = file.read()
                # Send data over serial
                ser.write(data)
                print("File sent successfully.")
    except serial.SerialException as e:
        print(f"Error opening serial port: {e}")
    except FileNotFoundError:
        print("The specified file was not found.")
    except Exception as e:
        print(f"An error occurred: {e}")

# Example usage
file_path = "path_to_your_file/Compressed_success.bin"  # Adjust the fi
send_file_over_serial("COM4", 921600, file_path)
```

Example Code in Node.js:

```javascript
const SerialPort = require('serialport');
const fs = require('fs');

const portName = 'COM4'; // Adjust as needed
const baudRate = 921600;
const filePath = 'path_to_your_file/Compressed_success.bin'; // Adjust the file path accordingly

const port = new SerialPort(portName, { baudRate: baudRate }, (err) => {
  if (err) {
    return console.log('Error opening serial port:', err.message);
  }
});
```

```
// Open errors will be emitted as an error event
port.on('error', (err) => {
    console.log('Error:', err.message);
});

fs.readFile(filePath, (err, data) => {
    if (err) {
        return console.log('Error reading file:', err);
    }
    port.write(data, (err) => {
        if (err) {
            return console.log('Error writing to serial port:', err);
        }
        console.log('File sent successfully');
    });
});
```

javascript Copy code

```javascript
const SerialPort = require('serialport');
const fs = require('fs');

const portName = 'COM4'; // Adjust as needed
const baudRate = 921600;
const filePath = 'path_to_your_file/Compressed_success.bin'; // Adjust

const port = new SerialPort(portName, { baudRate: baudRate }, (err) =>
    if (err) {
        return console.log('Error opening serial port:', err.message);
    }
});

// Open errors will be emitted as an error event
port.on('error', (err) => {
    console.log('Error:', err.message);
});

fs.readFile(filePath, (err, data) => {
    if (err) {
        return console.log('Error reading file:', err);
    }
    port.write(data, (err) => {
        if (err) {
            return console.log('Error writing to serial port:', err);
        }
        console.log('File sent successfully');
    });
});
```