StarFive 赛昉科技

# Using an LED Dot Matrix with VisionFive 2

with Python

Application Note

Version: 1.0

Date: 2022/11/30

Doc ID: VisionFive2-ANEN-003

# Legal Statements

Important legal notice before reading our documentation.

### PROPRIETARY NOTICE



Information in this document is provided "as is," with all faults. Contents may be periodically updated or revised due to the product development. Shanghai StarFive Technology Co., Ltd. (hereinafter "StarFive") reserves the right to make changes without further notice to any products herein.

StarFive expressly disclaims all warranties, representations, and conditions of any kind, whether express or implied, including, but not limited to, the implied warranties or conditions of merchantability, fitness for a particular purpose and non-infringement.

StarFive does not assume any liability rising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation indirect, incidental, special, exemplary, or consequential damages.

All material appearing in this document is protected by copyright and is the property of StarFive. You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. Information contained in this document may be used, at your sole risk, for any purposes. StarFive authorizes you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services.

### Contact Us

Address: Room 502, Building 2, No. 61 Shengxia Rd., China (Shanghai) Pilot Free Trade Zone, Shanghai, 201203, China

Website: http://www.starfivetech.com

Email: sales@starfivetech.com(sales) , support@starfivetech.com(support)

# Preface

About this guide and technical support information.

## About this document

This application note provides steps to use VisionFive 2's GPIO pins to make a MAX7219 Serial Dot Matrix display with StarFive logo through an example program with Python.

## Revision History

**Table 0-1 Revision History**

| Version | Released | Revision |
|---|---|---|
| 1.0 | 2022/11/30 | The first official release. |

## Notes and notices

The following notes and notices might appear in this guide:

- **Tip:**
  Suggests how to apply the information in a topic or step.
- **Note:**
  Explains a special case or expands on an important point.
- **Important:**
  Points out critical information concerning a topic or step.
- **CAUTION:**
  Indicates that an action or step can cause loss of data, security problems, or performance issues.
- **Warning:**
  Indicates that an action or step can result in physical harm or cause damage to hardware.

# Contents

# List of Tables

# List of Figures

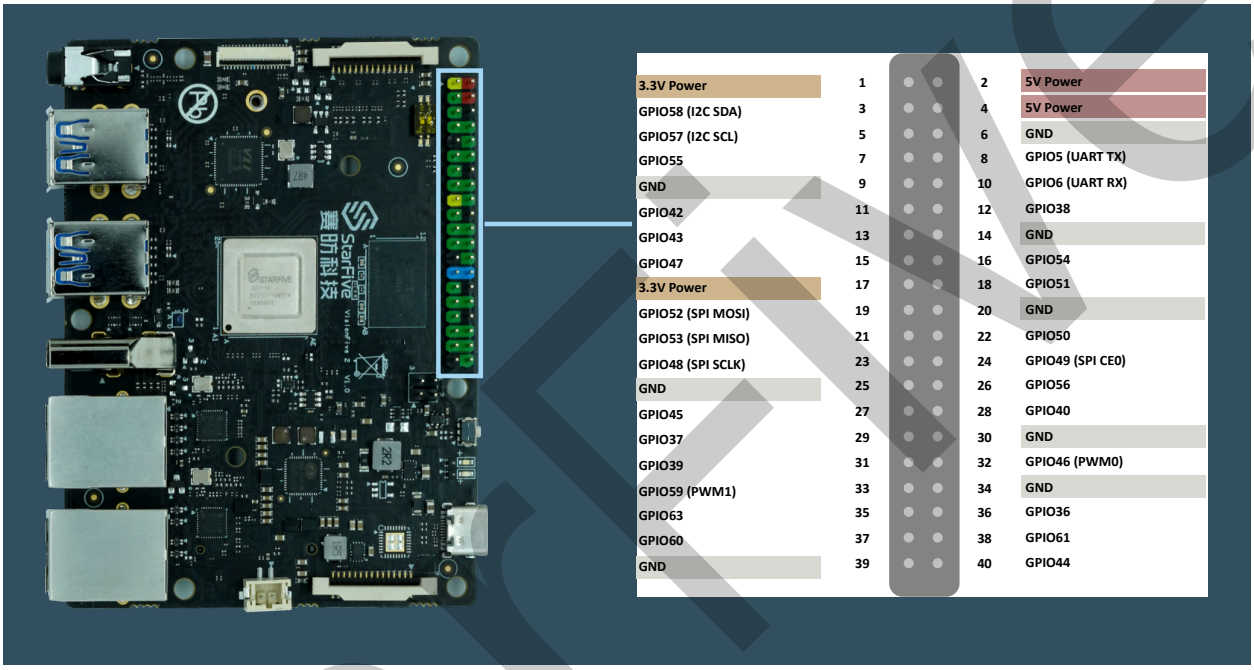
# 1. Introduction

This application note provides steps to use VisionFive 2's GPIO pins to make a MAX7219 Serial Dot Matrix display with StarFive logo through an example program with Python.

## 1.1. 40-Pin Header Definition

The following figure shows the location of the 40-pin header on VisionFive 2.

**Figure 1-1 40-Pin Definition**



# 2. Preparation

Before executing the demo program, make sure you prepare the following:

## 2.1. Environment Requirements

The environment requirements are as follows:

- Linux Kernel: Linux 5.15
- OS: Debian 12
- SBC: VisionFive 2
- SoC: JH7110

## 2.2. Preparing Hardware

Prepare the following hardware items before running the demo code:

**Table 2-1 Hardware Preparation**

| Type | M/O* | Item | Notes |
|---|---|---|---|
| General | M | VisionFive 2 single board computer | - |
| General | M | • 32 GB (or more) micro-SD card<br>• micro-SD card reader<br>• Computer (Windows/macOS/Linux)<br>• USB to serial converter (3.3 V I/O)<br>• Ethernet cable<br>• Power adapter (5 V / 3 A)<br>• USB Type-C Cable | These items are used for flashing Debian OS into a micro-SD card. |
| GPIO Demo (LED Matrix) | M | MAX7219 Serial Dot Matrix Display Module (with a 5-way female to female Dupont cable) | - |

**Note:**
*: M: Mandatory, O: Optional

### 2.2.1. Hardware Setup

The following table and figure describe how to connect MAX7219 to the 40-pin GPIO header:

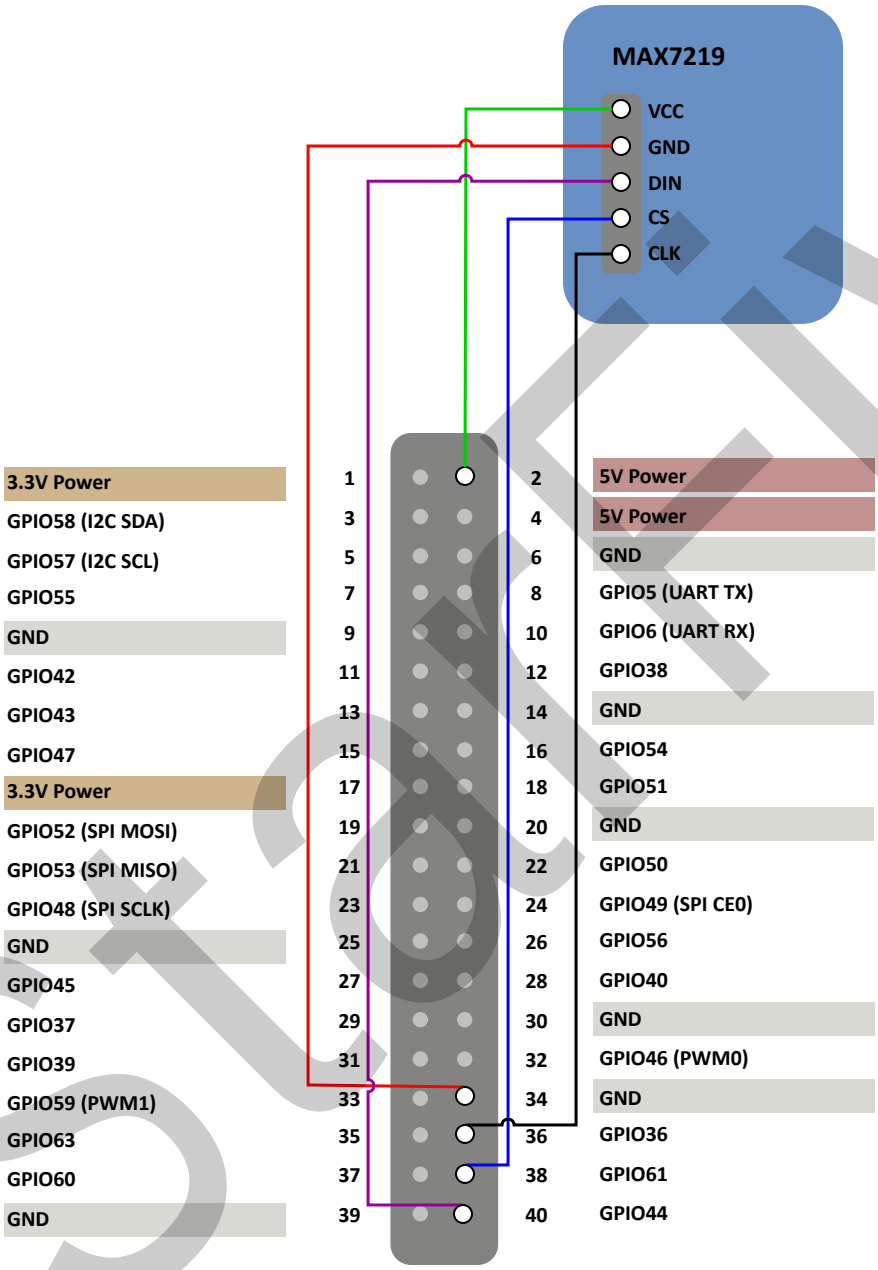
**Table 2-2 Connect MAX7219 to the 40-Pin Header**

| MAX7219 | 40-Pin GPIO Header | |
|---|---|---|
| | Pin Number | Pin Name |
| VCC | 2 | 5V Power |
| GND | 34 | GND |
| DIN | 40 | GPIO44 |

**Table 2-2 Connect MAX7219 to the 40-Pin Header (continued)**

| MAX7219 | 40-Pin GPIO Header | |
|---|---|---|
| | Pin Number | Pin Name |
| CS | 38 | GPIO61 |
| CLK | 36 | GPIO36 |

**Figure 2-1 Connect MAX7219 to the 40-Pin Header**




## 2.3. Preparing Software

Make sure the following procedures are performed:

**Note:**
The Python project, `VisionFive.gpio`, is applicable for VisionFive, VisionFive 2, and JH7110 EVB.

1. Flash Debian OS into a Micro-SD card as described in the *Flashing OS to a Micro-SD Card* section in the *VisionFive 2 Single Board Computer Quick Start Guide*.
2. Log into the Debian and make sure VisionFive 2 is connected to the Internet. For detailed instructions, refer to the *Using SSH over Ethernet* or *Using a USB to Serial Converter* section in the *VisionFive 2 Single Board Computer Quick Start Guide*.
3. Extend the partition on Debian as described in *Extend Partition* in the *VisionFive 2 Single Board Computer Quick Start Guide*.
4. Execute the following command to install PIP on Debian:

   ```
   apt-get install python3-pip
   ```

5. Execute the `pip` command on VisionFive 2 Debian to install the `VisionFive.gpio` package:

   ```
   sudo pip install VisionFive.gpio
   ```

   Alternatively, you can execute the following command:

   ```
   sudo pip3 install VisionFive.gpio
   ```

6. (Optional) If you copy the source code to the local directory under VisionFive 2 Debian, execute the following commands under the source code directory:

   **Tip:**
   The source code can be downloaded by clicking the following link: VisionFive.gpio.

   ```
   sudo apt-get install python3-dev
   sudo python setup.py install
   ```

   Alternatively, you can execute the following command:

   ```
   sudo python3 setup.py install
   ```

# 3. Running Demo Code

To run the demo code, perform the following on VisionFive 2 Debian:

1. Locate to the directory where the test code, `LED_Matrix.py`, exists:

    a. Execute the following command to get the directory where `VisionFive.gpio` exists:

    ```
    pip show VisionFive.gpio
    ```

    **Example Result:**

    ```
    Location: /usr/local/lib64/python3.9/site-packages
    ```

    **Note:**

    The actual output depends on how the application is installed.

    b. Execute the following to enter the directory, for example, `/usr/local/lib64/python3.9/site-packages` as indicated in the previous step output:

    ```
    cd /usr/local/lib64/python3.9/site-packages
    ```

    c. Execute the following command to enter the `sample-code` directory:

    ```
    cd ./VisionFive/sample-code/
    ```

2. Under the `sample-code` directory, execute the following command to run the demo code:

    ```
    sudo python LED_Matrix.py
    ```

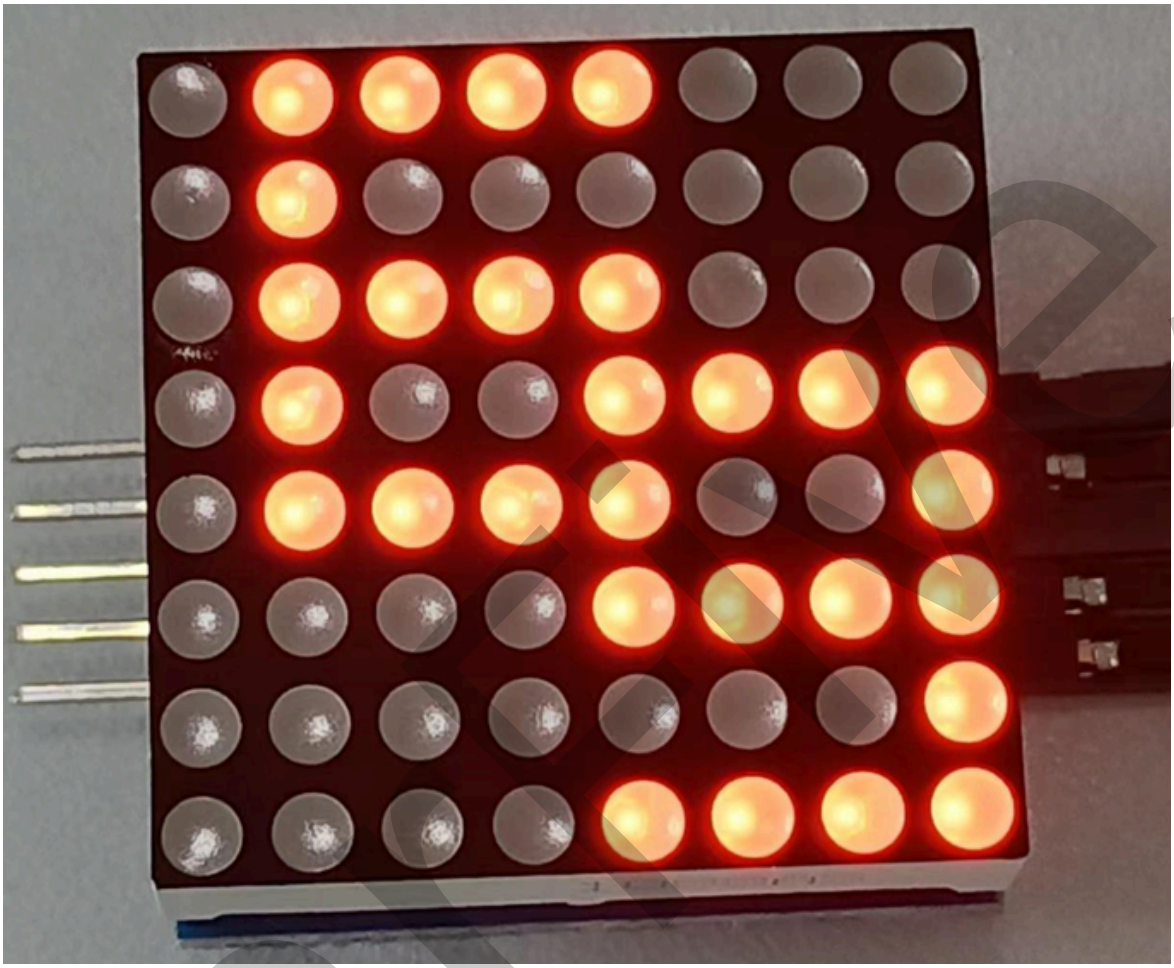
    Alternatively, you can execute the following command:

    ```
    sudo python3 LED_Matrix.py
    ```

    **Result:**

The Led Matrix module displays with the StarFive logo.

**Figure 3-1 Example Output**



# 4. Demo Source Code

The Python source code of this demo is provided for reference purpose only.

`LED_Matrix.py`:

```
'''
Please make sure the LED Dot Matrix is connected to the correct pins.
The following table describes how to connect LED Dot Matrix to the 40-pin header.
----------------------------------------
___MAX7219_______Pin Number_____Pin Name
    VCC             2           5V Power
    GND             34           GND
    DIN             40          GPIO44
    CS              38          GPIO61
    CLK             36          GPIO36
----------------------------------------
'''

import VisionFive.gpio as GPIO
import sys
import time

DIN = 40
CS  = 38
CLK = 36
#Configure the direction of DIN, CS, and CLK as out.
GPIO.setup(DIN, GPIO.OUT)
GPIO.setup(CS, GPIO.OUT)
GPIO.setup(CLK, GPIO.OUT)

#Display logo data.
buffer = ['01111000', '01000000', '01111000', '01001111', '01111001', '00001111', '00000001', '00001111']

#LED turn off data.
buffer_off = ['0', '0', '0', '0', '0', '0', '0', '0']

def sendbyte(bytedata):
    for bit in range(0, 8):
        if ((bytedata << bit) & 0x80):
            GPIO.output(DIN, GPIO.HIGH)
        else:
            GPIO.output(DIN, GPIO.LOW)

        #Configure the voltage level of CLK as high.
        GPIO.output(CLK, GPIO.HIGH)
        #Configure the voltage level of CLK as low.
        GPIO.output(CLK, GPIO.LOW)


def WriteToReg(regaddr, bytedata):
    #Configure the voltage level of cs as high.
    GPIO.output(CS, GPIO.HIGH)
    #Configure the voltage level of led_pin as low.
    GPIO.output(CS,GPIO.LOW)
    GPIO.output(CLK, GPIO.LOW)
    sendbyte(regaddr)
    sendbyte(bytedata)
    GPIO.output(CS, GPIO.HIGH)

def WriteALLReg():
    time.sleep(0.1)
    for i in range(0, 8):
        #Write data to register address. Finally the LED matrix displays StarFive logo.
        WriteToReg(i+1, int(buffer[i], 2))
    time.sleep(5)

    #Display logo.
    for i in range(0, 10):
        for j in range(0, 8):
```

```
            #Write data to the register address. Finally turn off the LED matrix.
            WriteToReg(j+1, int(buffer_off[j], 2))
        time.sleep(0.1)
        for j in range(0, 8):
            #Write data to the register address. Finally the LED matrix displays with StarFive logo.
            WriteToReg(j+1, int(buffer[j], 2))
        time.sleep(0.1)

def initData():
    WriteToReg(0x09, 0x00) #Set the decode mode.
    WriteToReg(0x0a, 0x03) #Set the brightness.
    WriteToReg(0x0b, 0x07) #Set the scan limitation.
    WriteToReg(0x0c, 0x01) #Set the power mode.
    WriteToReg(0x0f, 0x00)

def main():
    initData()
    while True:
        WriteALLReg()

if __name__ == "__main__":
    sys.exit(main())
```