EC2x-QuecOpen

Matrix Keypad Guidelines

LTE Standard Module Series

Rev. EC2x-QuecOpen_Matrix_Keypad_Guidelines_V1.1

Date: 2019-04-16

Status: Preliminary

Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:

Quectel Wireless Solutions Co., Ltd.
7th Floor, Hongye Building, No.1801 Hongmei Road, Xuhui District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

Or our local office. For more information, please visit:
http://www.quectel.com/support/sales.htm

For technical support, or to report documentation errors, please visit:
http://www.quectel.com/support/technical.htm
Or email to: support@quectel.com

GENERAL NOTES

QUECTEL OFFERS THE INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. QUECTEL MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. QUECTEL DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

COPYRIGHT

About the Document

History

Revision	Date	Author	Description
1.0	2018-09-30	Gale GAO	Initial
1.1	2019-04-16	Juson ZHANG	Updated Row and Column Pin Description

Contents

About the Document 1

Contents 2

1 Introduction 3

2 Matrix Keypad Hardware Design 4

3 Driver and Device Tree 5

3.1. Add Matrix Keypad Configuration to Device Tree 5

3.2. Enable Matrix Keypad Kernel Option 6

4 Application Example 8

1 Introduction

Independent Keypad: Simple programming but wasting IO interface;

Matrix Keypad: Complex programming but saving IO interface;

Matrix Keypad is recommended when the number of keys over 6;

In order to meet the customer's requirement for Matrix Keypad, this document provides driver, device tree debugging guidance and user-layer program example.

2 Matrix Keypad Hardware Design

Common reference design as the following 4*4 matrix keypad;

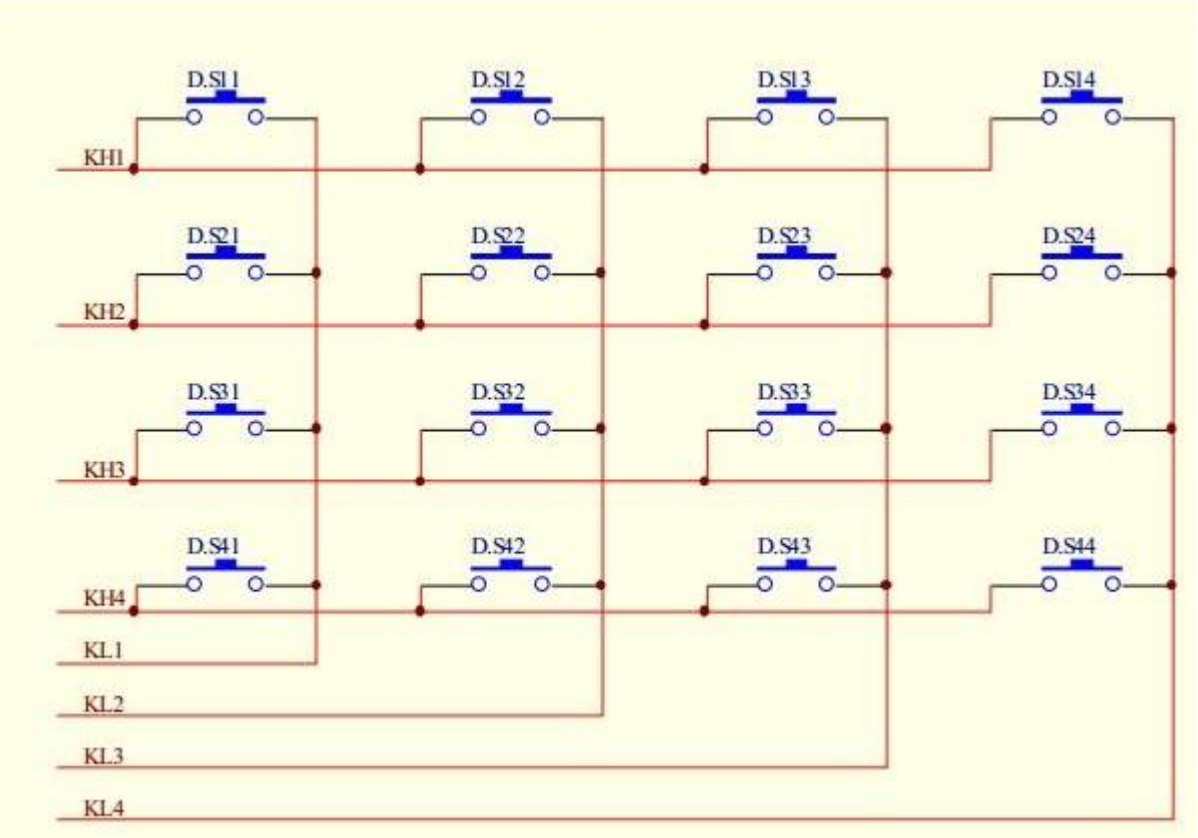


Figure 1：4*4 Matrix Keypad

3 Driver and Device Tree

3.1. Add Matrix Keypad Configuration to Device Tree

Add following Matrix Keypad configuration under the file: *mdm9607.dtsi.*

- Attribute explanation:
  debounce-delay-ms: Time for key anti-shake;
  col-scan-delay-us: Column scan delay after the key is triggered;
  linux,wakeup: Support key wake up module;
  row-gpios: GPIO number used in ROW;
  col-gpios: GPIO number used in COLUMN;
  linux-keymap: Keycodes for row and column and also for reporting to APP;

  Reference to device tree confirguration:
  ql-ol-kernel/msm-3.18/Documentation/devicetree/bindings/input/gpio-matrix-keypad.txt

- Function description:
  As input pin, columns select IO that in support of interrupting wake up and keep default internal pull down;
  As output pin, rows drive default configuration and output high-level
  When edge is triggered, both rising edge and falling edge will be reported;

  Driver source code:
  ql-ol-kernel/msm-3.18/drivers/input/keypad/matrix_keypad.c

  User-layer program refers to ***chapter 4.***

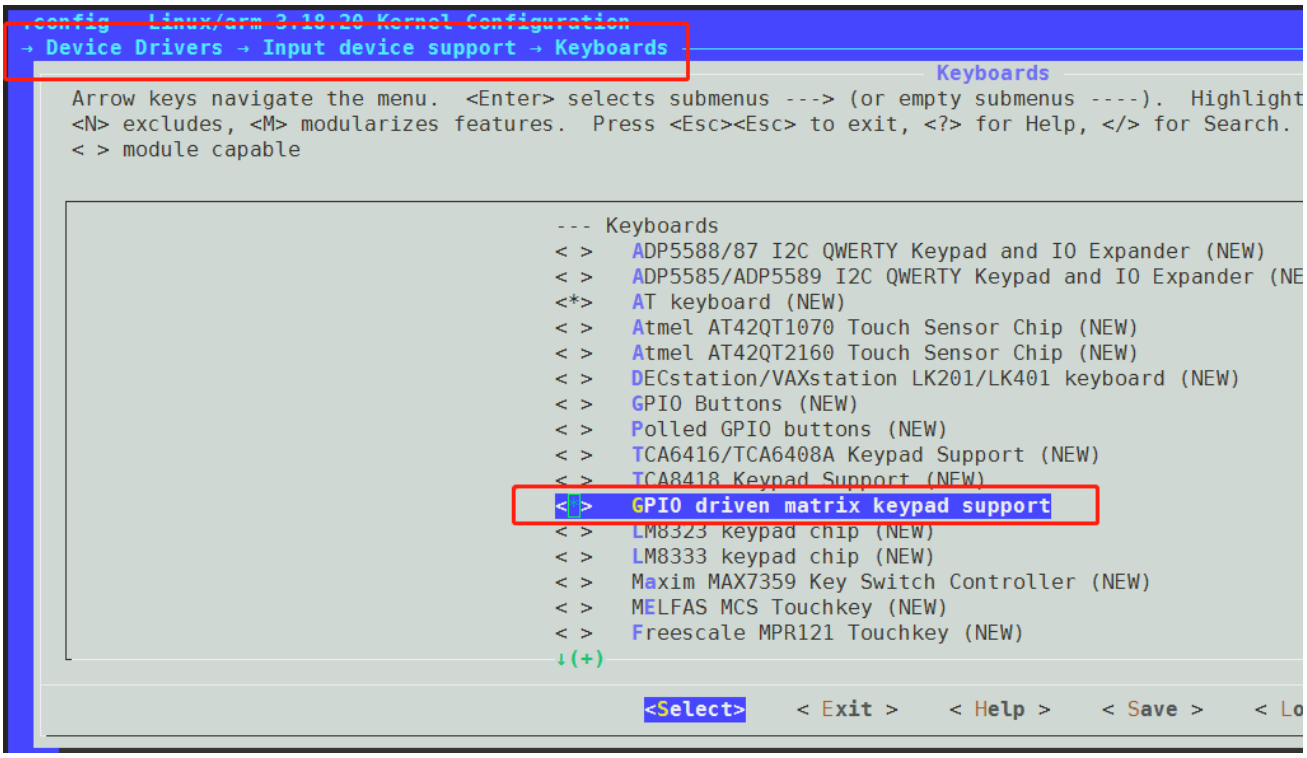
```
--- ql-ol-kernel-orig/arch/arm/boot/dts/qcom/mdm9607.dtsi   2018-05-12 19:02:48.000000000 +0800
+++ ql-ol-kernel/arch/arm/boot/dts/qcom/mdm9607.dtsi    2018-06-26 09:33:51.915118760 +0800
@@ -1849,6 +1849,65 @@
         interrupts = <0x0 0xA1 0x0>; /* PMD9607 MPP 2 */
         interrupt-names = "vbus_det_irq";
     };
+
+    matrix-keypad {
+        compatible = "gpio-matrix-keypad";
+        debounce-delay-ms = <20>;
+        col-scan-delay-us = <2>;
+        linux,no-autorepeat;
+        /*linux,clustered_irq = <88>;
+        linux,clustered_irq_flags = <0x1>;*/
+        linux,wakeup;
+
+        row-gpios = <&tlmm_pinmux 3 0
+                 &tlmm_pinmux 5 0
+                 &tlmm_pinmux 34 0
+                 &tlmm_pinmux 42 0
+                 &tlmm_pinmux 75 0>;
+
+        col-gpios = <&tlmm_pinmux 0 0
+                 &tlmm_pinmux 2 0
+                 &tlmm_pinmux 4 0
+                 &tlmm_pinmux 10 0
+                 &tlmm_pinmux 24 0
+                 &tlmm_pinmux 25 0>;
+
+        linux,keymap = <0x00000000  /* 1st col */
+                0x01000001
+                0x02000002
+                0x03000003
+                0x04000004
+
+                0x00010005  /* 2nd col */
+                0x01010006
+                0x02010007
+                0x03010008
+                0x04010009
+
+                0x0002000a
+                0x0102000b
+                0x0202000c
+                0x0302000d
+                0x0402000e
+
+                0x0003000f
+                0x01030010
+                0x02030011
+                0x03030012
+                0x04030013
+
+                0x00040014
+                0x01040015
+                0x02040016
+                0x03040017
+                0x04040018
+
+                0x00050019
+                0x0105001a
+                0x0205001b
+                /*0x0305001c
+                0x0405001d*/>;
+    };
 };

 #include "mdm9607-rpm-regulator.dtsi"
```

3.2. Enable Matrix Keypad Kernel Option

ql-ol-sdk$ make kernel_menuconfig

ql-ol-sdk$ make kernel

Download Mirroring



4 Application Example

矩阵键盘-支持唤醒

The Matrix Keypad directory in the attachment above needs to be copied under QL-OL-EXTSDK for compilation, and then download the executable program to the module for testing; When the key is triggered, users can see the key code is reported, as well as under sleep state.