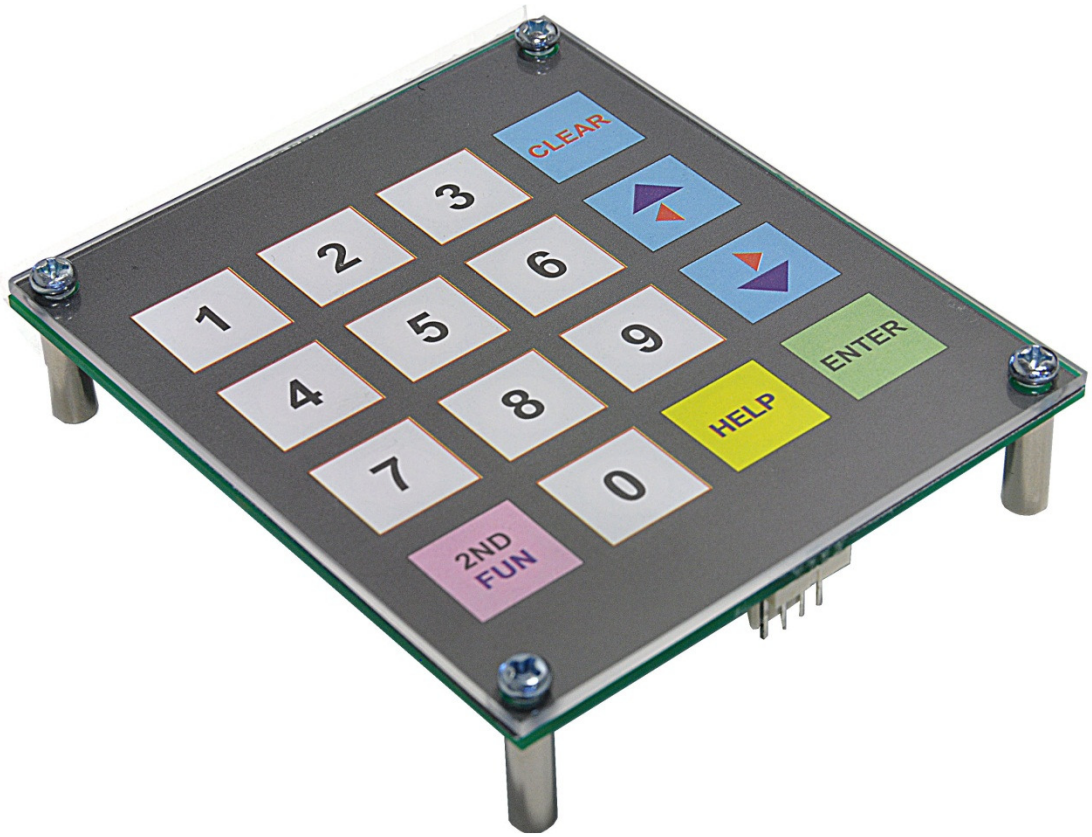
GRAVITECH.US



electronic experimental solutions

 WWW.GRAVITECH.US

Features

- 16 Keys, 4x4 touching pad (capacitive sensing technology)
- Compatible with 3.3V and 5.0V system
- Interfaces directly to any microcontroller or microprocessor
- LED indication and beeper sound for each individual keypress provide feedback to operator
- Running lights and sound at the start up
- Serial (UART-9600baud) and Parallel (BCD8421) outputs
- Include 2mm thick clear plastic face plate (other materials can be use)
- A special function (FUN) key can be used with another keypress to expand the number of keypresses (pressing two keys simultaneously)
- Automatically enter Sleep Mode after 8 sec. of no activity
- 2mA current consumption in Sleep Mode
- Cost effective

Module layout and functions

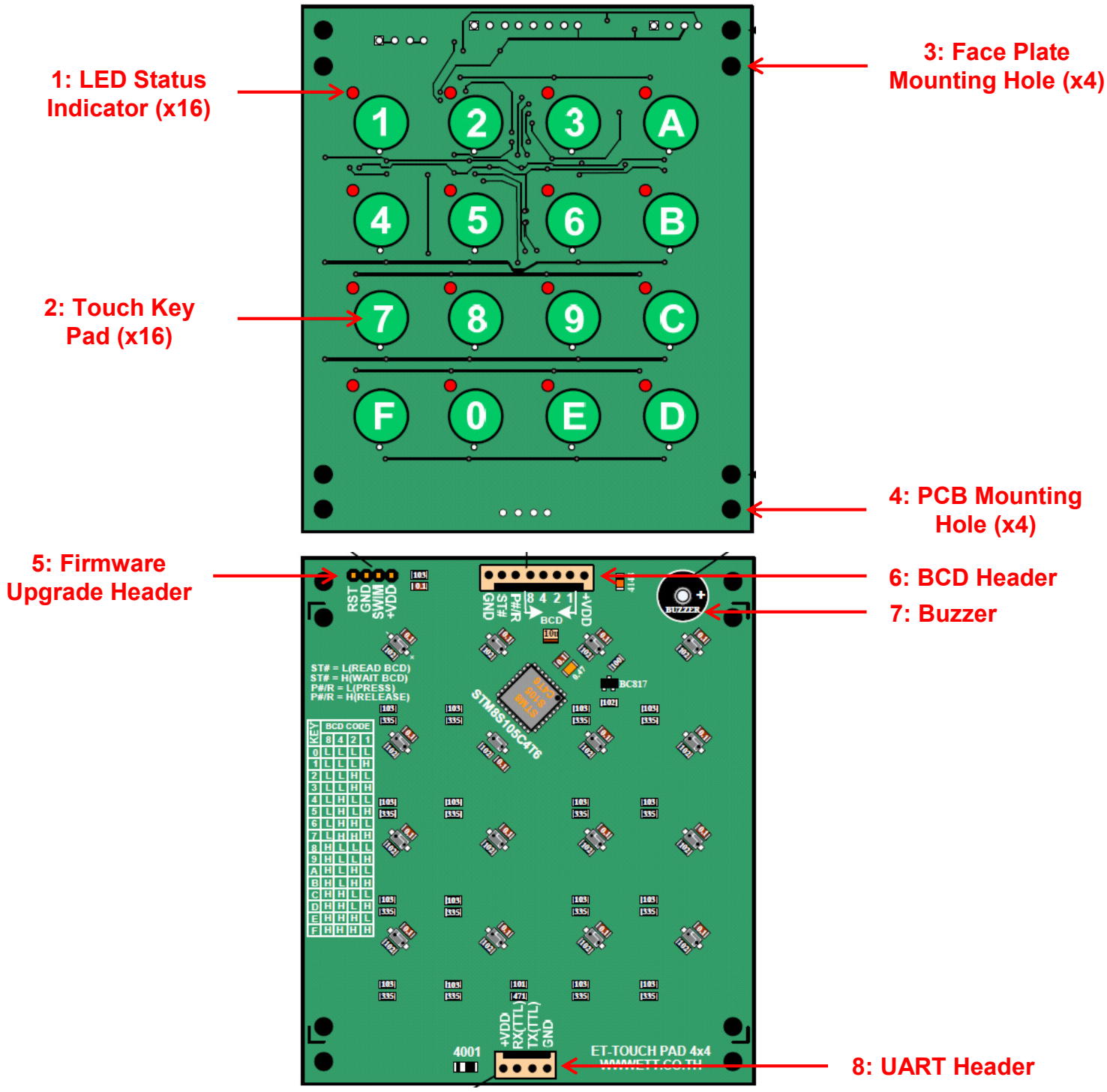


Fig. 1: PCB layout (top and bottom)

1. ***LED Status Indicator:*** It shows the status of a keypress by blinking when detected. All of the LED is illuminate at the start up in running sequence.

2. ***Touch Key Pad:*** It is an area where the keypress sensing active. It is not necessary to touch the print circuit board (PCB) directly like many other competitive modules due to a strong sensitivity of the module. The module also includes the 2mm clear plastic face plate so that the keypress label can be inserted. The label that comes with the module is for reference only. The user can change the label to suit your application. Please refer to Key Code table below.

3. ***Face Plate Mounting Hole:*** It is the hole us to attach fact plate to the PCB.

4. ***PCB Mounting Hole:*** It is the hole for user to attach the module to other structure.

5. ***Firmware Upgrade Header:*** It is use to upgrade module firmware.

6. ***BCD Header:*** It is 8-PIN 0.1” male connector use for power and sends out keypress data in Binary BCD8421 format. See figure 2 for the pin-out.

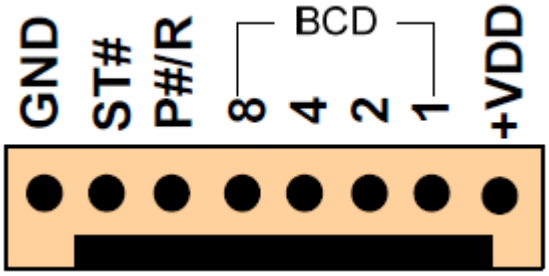


Fig. 2: BCD Header pin-out

BCD Header Pin-out:

+VDD, GND = Power and ground for the module. The module is accepting 5V (for 5V microcontroller, MCU) or 3.3V (for 3.3V MCU).

BCD8421 = 4-bit Key Code data for the keypress. BCD8 is the most significant bit (MSB). The data gets update every time there is a keypress. Otherwise remain the same for entire time (indicating the last keypress).

P#/R = Press/Release pin, indicating the keypress status. It is normally logic HIGH. It becomes logic LOW when keypress is detected and stay LOW until the keypress released.

ST# = Strobe pin, indicating the keypress status. It is normally logic HIGH. It sends out a 10mS pulse once the keypress is detected as show on the timing diagram figure 3.

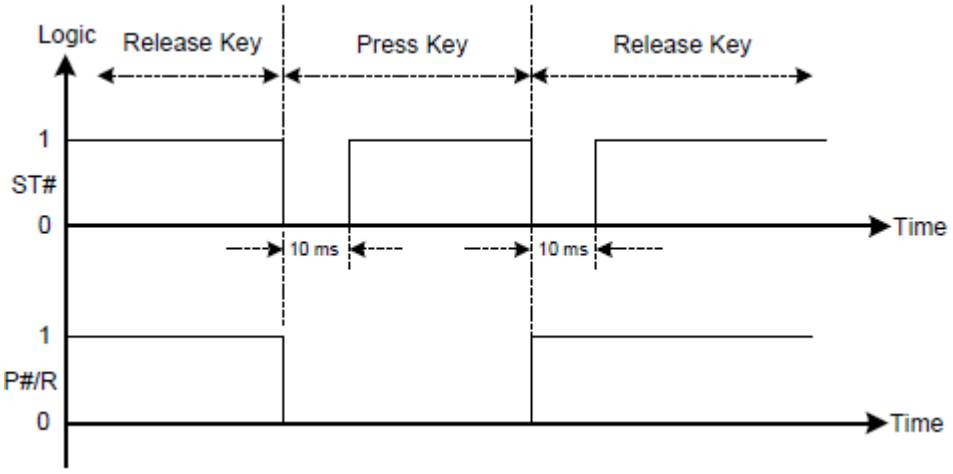


Fig. 3: P#/R and ST# timing diagram

NOTE: In a situation when the FUN key is use as a regular key (single key) the user can use either P#/R or ST# pin to check for the keypress status. However, when use FUN key simultaneously with other key it is necessary to check both of these pins for the status.

7. ***Buzzer:*** Sending the beep sound when keypress is detected. At the start up the buzzer beeps according to the LED Status Indicator.

8. ***UART Header:*** It is 4-PIN 0.1" male connector use for power and sends out keypress data in RS232-TTL (UART). The baud rate is fixed at 9600. TX and RX pins can be connecting directly to the MCU. **DO NOT** attaches this module directly to a computer serial port. You will need a RS232 level shifter circuitry. Doing so may damage the module or the computer. See figure 3 for the pin-out.

   Once the keypress is detected, the TX pin send out ASCII "P" (0x50) or upon keypress released the ASCII "R" (0x52). Then follow by the Key Code and 0x0D (total of 3 bytes).

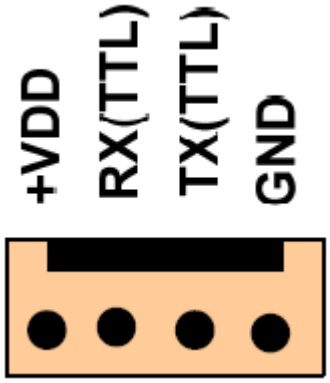


Fig. 4: UART Header pin-out

UART Header Pin-out:

+VDD, GND = Power and ground for the module. The module is accepting 5V (for 5V microcontroller, MCU) or 3.3V (for 3.3V MCU).

RX(TTL) = Receive serial UART data from user (not used).

TX(TTL) = Transmit serial UART keypress data.

	ASCII CODE		
Action/Data	Status Key (Byte 1)	Key Code (Byte2)	End Byte (Byte 3)
PRESS	“P” (0x50)	“0-9”, “A-F”	0x0D
RELEASE	“R” (0x52)	“0-9”, “A-F”	0x0D

Table 1: UART transmit data package

Example: The UART data package once key “5” press is 0x50, 0x35 and 0x0D. Once it release is 0x52, 0x35 and 0x0D.

KEY	FOR Binary MODE						FOR ASCII Mode (RS232 TTL)	
	BCD 8421 KEY CODE						ASCII KEY CODE	
	8	4	2	1	HEX		ASCII	HEX
1	0	0	0	1	0x01		'1'	0x31
2	0	0	1	0	0x02		'2'	0x32
3	0	0	1	1	0x03		'3'	0x33
4	0	1	0	0	0x04		'4'	0x34
5	0	1	0	1	0x05		'5'	0x35
6	0	1	1	0	0x06		'6'	0x36
7	0	1	1	1	0x07		'7'	0x37
8	1	0	0	0	0x08		'8'	0x38
9	1	0	0	1	0x09		'9'	0x39
0	0	0	0	0	0x00		'0'	0x30
A	1	0	1	0	0x0A		'A'	0x41
B	1	0	1	1	0x0B		'B'	0x42
C	1	1	0	0	0x0C		'C'	0x43
D	1	1	0	1	0x0D		'D'	0x44
E	1	1	1	0	0x0E		'E'	0x45
F	1	1	1	1	0x0F		'F'	0x46

Table 2: Key Code Table

Theory of Operations

After the start up sequence, the default status of each pin is show below. There is no data sending out on UART TX pin. Once there is a keypress, all other key pads are disabled (except the F key) until that keypress is release. As for the F key, another key can be pressed after or use as a standalone key. After each keypress, the individual LED illuminates momentary along with the beep sound from the buzzer.

Once the keypress or keyrelease detected the status of Key Code both in BCD and UART format along with *ST #* and *P#/R* pins get updated. Below are the details of each format.

1. ***Reading Key Code in BCD8421 format***

- **Default status:** *ST#* = 1, *P#/R* = 1,
  BCD8 = 0, *BCD4* = 0, *BCD2* = 0, *BCD1* = 0

- **Keypress detected:** Below is the sequence of the events on the BCD 8-pin connector.

  1) *ST#* change from logic 1 to 0 for 10mS then go back to logic 1 (Figure 3)
  2) *P#/R* change from logic 1 to 0 and stay at logic 0 (Figure 3)

3) *BCD8, 4, 2, 1* change according to Key Code table2

In case of using the F key along with other key, the same 3 steps above occur once the F keypress. If the module detect other keypress without releasing the F key, *ST#* pin follow the same step 1 and *P#/R* pin still at logic 0 (no change). Therefore to check for second keypress monitor the *ST#* pin. The *BCD8421* pins then change to the new Key Code according to the second keypress.

- **Keyrelease detected:** Below is the sequence of the events on the BCD 8-pin connector.

1) *ST#* change from logic 1 to 0 for 10mS then go back to logic 1 (Figure 3)
2) *P#/R* change from logic 0 to 1 and stay at logic 1 (Figure 3)
3) *BCD8, 4, 2, 1* change according to Key Code table2

In case of using the F key along with other key, once either one of the keyrelease the status of *ST#* pin change according to step 1 above and *P#/R* pin still remain at logic 0. This is because there is another key remain press. The *BCD8421* pins change according to the key that was released. Once the last

keypress release, the same 3 steps above occur. Therefore to check for the keyrelease monitor the *ST#* pin then check the BCD8421 to see which key was released. Then check the *P#/R* pin for every key has been released.

2. ***Reading Key Code in UART format***

TX and RX pins can be connecting directly to the MCU. The baud rate is fixed at 9600. **DO NOT** attaches this module directly to a computer serial port. You will need a RS232 level shifter circuitry. Doing so may damage the module or the computer.

- **Default status:** TX = 1

- **Keypress detected:** 3 bytes data package is sending out (see table1)

  In case of using the F key along with other key, the same 3 bytes is sending out (0x50, 0x46 and 0x0D). Once the second keypress without releasing the F key, the same data package structure is sending out (0x50, Key Code and 0x0D). Therefore, to detect this event just check for another keypress status “P” (0x50). Otherwise using F key in standalone mode you would see the “R” (0x52) at the beginning of the package.

- **Keyrelease detected:** 3 bytes data package is sending out (see table 1)

  In case of using the F key along with other key, whichever key release first the data package send out according to that key.

Examples Schematic Diagram and Source Code

The following are examples of the method to read Key Code in both formats BCD8421 and UART. There are 3 MCUs use in these examples ARM7 LPC2138 (using C-Keil Compiler), AVR MEGA 64/128 (using C-WIN AVR) and MCS51 AT89C51ED2 (using C-Keil Complier). Each MCU is showing in both formats BCD8421 and UART. The examples source code can be found on product page. Go to www.gravitech.us then search for 4x4-TOUCH-CAP.

Example of reading Key Code in BCD8421 format

EX1_Read_1Key_BCD: This is an example reading single key in BCD8421 format. The MCU is checking for the *ST#*, read the Key Code value then check for the *P#/R* for keyrelease. The MCU then send the Key Code value to the attached LEDs. Also, transmit the Key Code value to MCU serial port.

Below are the schematic diagrams for this example.

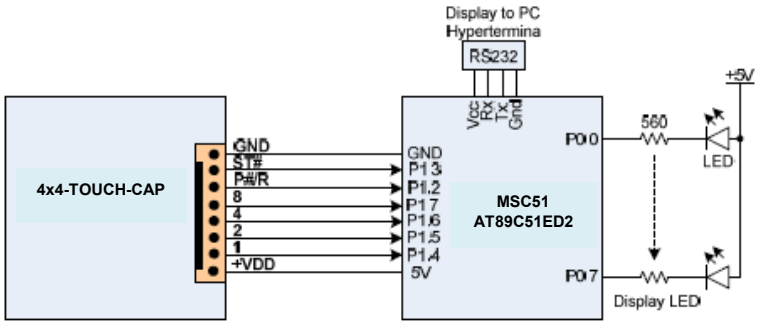


Fig. 5.1: Schematic for EX1-3 using MCS51 AT89C51ED2 MCU

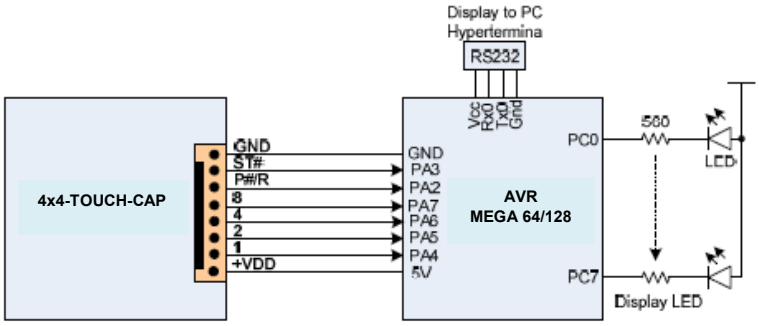


Fig. 5.2: Schematic for EX1-3 using AVR MEGA 64/128

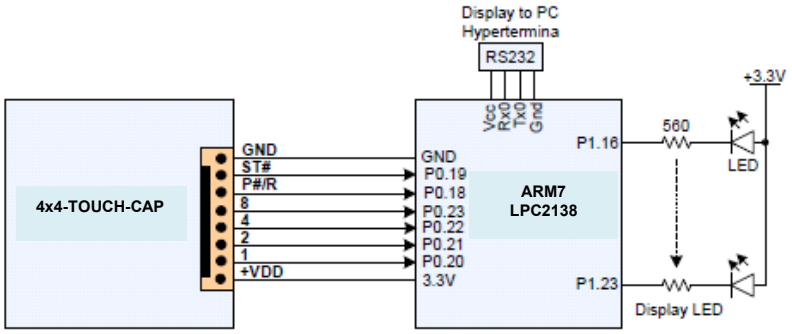


Fig. 5.3: Schematic for EX1-3 using ARM7 LPC2138

EX2_Read_2Key_BCD: This is an example reading F Key along with other key. The MCU is checking for the *ST#* for starting status, wait for the *ST#* to go back to logic 1, read the Key Code value, check and make sure the *P#/R* is still at logic 0 (holding down the key), testing the Key Code read for the F key, monitor the *ST#* for the next keypress, read the second Key Code value, double check the *P#/R* again then response to the F key. The MCU then send the Key Code value to the attached LEDs. Also, transmit the Key Code value to MCU serial port. If the F key still active, the program is looping for another keypress. The schematic diagrams are the same in Figure. 5.1-5.3.

EX3_Application_2Key_BCD: This example is the modify version of Example 2. While there is no keypress the MCU send out message “Not! Touch Key” to the MCU serial port. After a keypress detected the MCU send out message “Touch Key = [Key Code]”. If the F key is use along with other key the message “Touch Key = FUN+[Key Code] send. The schematic diagrams are the same in Figure. 5.1-5.3.

Example of reading Key Code in UART format

EX4_Read_1Key_ASCII: This is an example reading single key at a time in UART format. The MCU is looping/polling for the UART receive. Once the receiving data available, checking for “P”, read the Key Code, send the Key Code value to the attached LEDs then transmit the Key Code value to MCU serial port.

Below is the schematic diagram for this example.

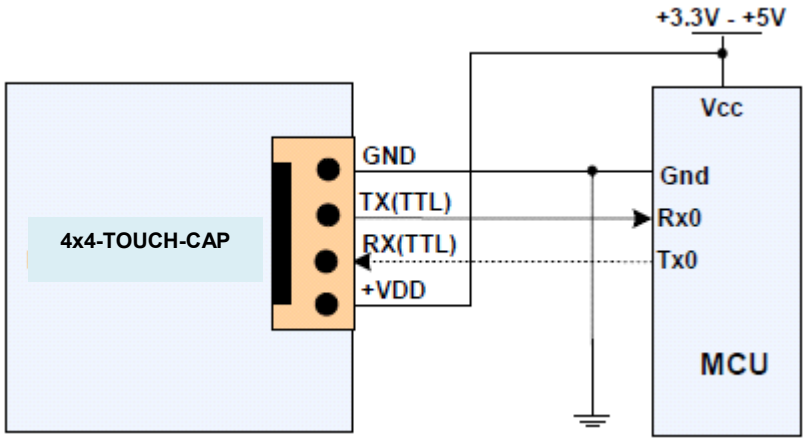
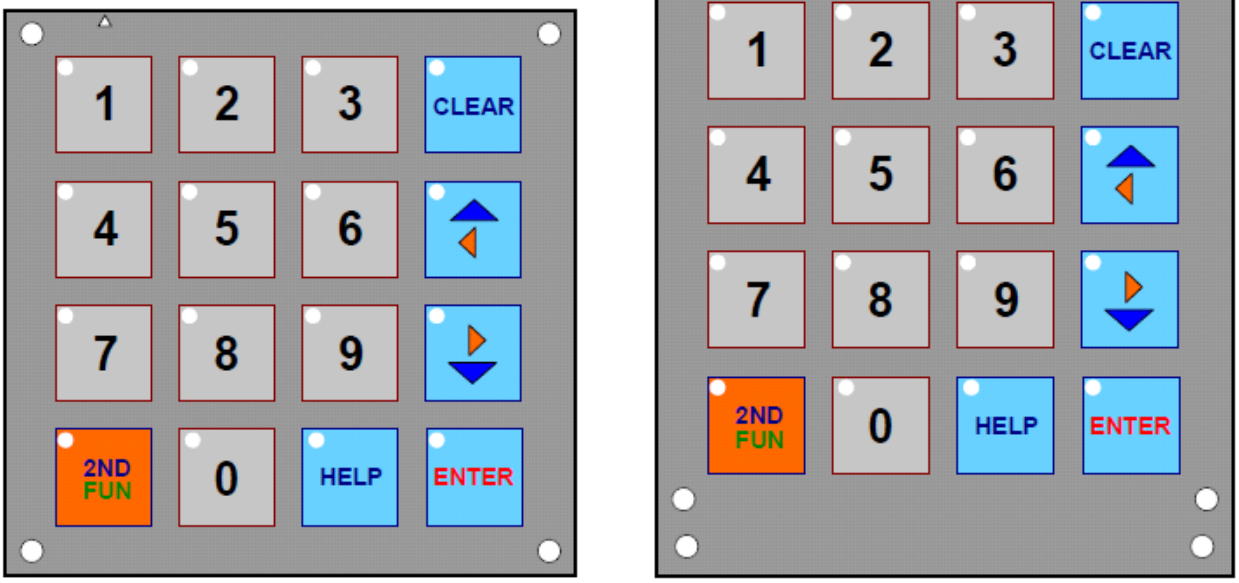


Fig. 6: Schematic for EX4-6 (reading Key Code in UART format)

EX5_Read_1Key_ASCII_INT: This example is the same as Example 4, but uses interrupt to read the Key Code. The schematic diagram is the same in Figure. 6.

EX6_Read_2Key_ASCII_INT: This is an example read the Key Code of F key along with other key. The schematic diagram is the same in Figure. 6.

Example of KEY labels (actual size)



PCB and label dimensions