

12-Digit Keypad Guide for BeagleBone Green

By Team HAP: Henry, Alex and Piercson
Last update: April 12, 2022

This document guides the user through

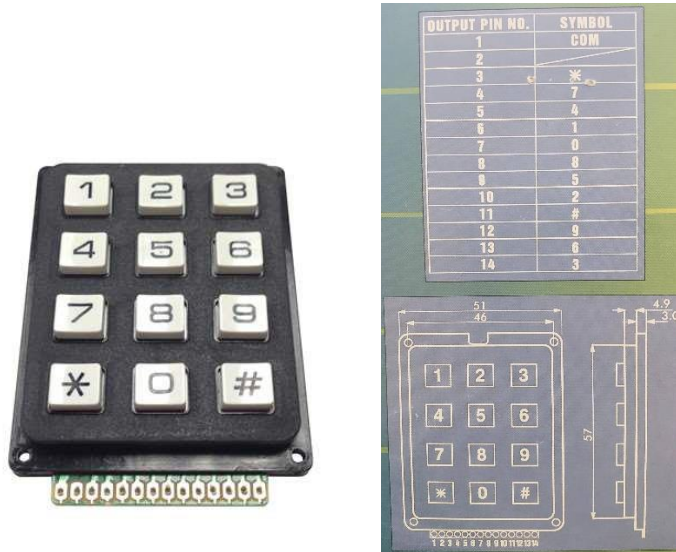
- 1. Wiring a 12-digit keypad to the BeagleBone Green
- 2. Using a C program to read user inputs

Table of Contents

- 1. Introduction 2
- 2. Wiring the 12-digit keypad to the BeagleBone Green 3
- 3. C code..... 5
- 4. Reference 6

1. introduction

This guide will provide step-by-step instructions to wire a [“KEYBOARD 12 KEYS” from velleman](#) with the BeagleBone Green. Below is a picture of the keypad and its specifications.



A mapping of the pins from left to right to the buttons is shown below.

1	2	3	4	5	6	7	8	9	10	11	12	13	14
COM	N/A	*	7	4	1	0	8	5	2	#	9	6	3

2. Wiring the 12-digit keypad to the BeagleBone Green

The keypad is rated for up to 24 volts and 20 milliamperes. For our circuit, we used one of the 3.3-volt VDDs as the circuit's power source, one of the ground pins as the circuit's ground, and 12 GPIO pins as inputs. Below is a picture of the Zencape's extension headers on BeagleBone Green (1).

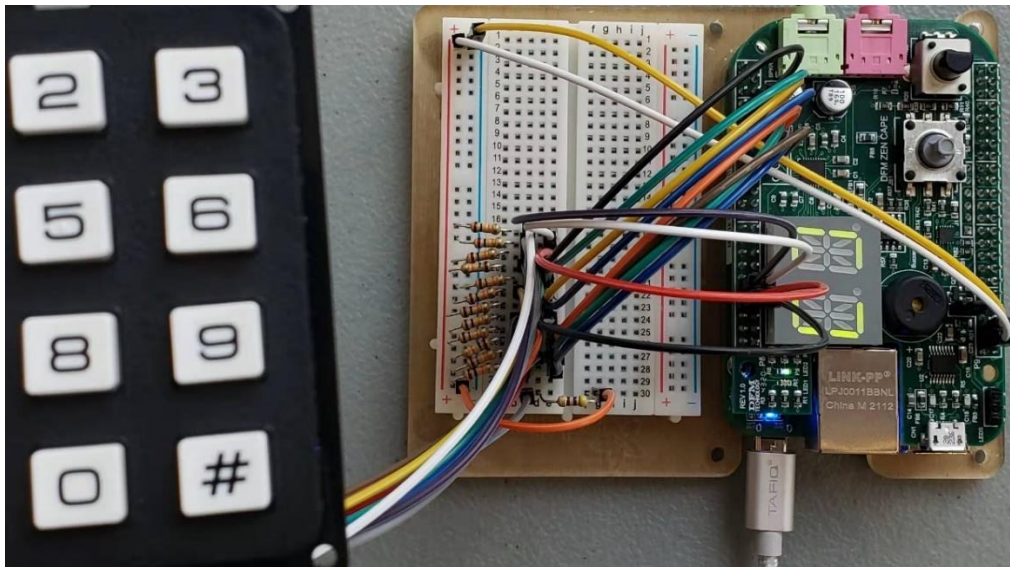
65 possible digital I/Os

P9				P8			
DDND	1	2	DDND	DDND	1	2	DDND
VDD_3V3	3	4	VDD_3V3	GPIO_38	3	4	GPIO_39
VDD_5V	5	8	VDD_5V	GPIO_34	5	6	GPIO_35
5V5_5V	7	8	5V5_5V	GPIO_66	7	8	GPIO_67
PWR_BTN	9	10	SW_RESETN	GPIO_69	9	10	GPIO_68
GPIO_30	11	12	GPIO_60	GPIO_45	11	12	GPIO_44
GPIO_31	13	14	GPIO_50	GPIO_23	13	14	GPIO_26
GPIO_48	15	16	GPIO_51	GPIO_47	15	16	GPIO_46
GPIO_5	17	18	GPIO_4	GPIO_27	17	18	GPIO_65
IOCS_SDA	19	20	IOCS_SDA	GPIO_22	19	20	GPIO_63
GPIO_3	21	22	GPIO_2	GPIO_62	21	22	GPIO_37
GPIO_49	23	24	GPIO_15	GPIO_36	23	24	GPIO_33
GPIO_117	25	26	GPIO_14	GPIO_32	25	26	GPIO_61
GPIO_115	27	28	GPIO_113	GPIO_86	27	28	GPIO_88
GPIO_111	29	30	GPIO_112	GPIO_87	29	30	GPIO_89
GPIO_110	31	32	VDD_ADC	GPIO_10	31	32	GPIO_11
AIN4	33	34	GNDA_ADC	GPIO_9	33	34	GPIO_81
AIN6	35	36	AIN5	GPIO_8	35	36	GPIO_80
AIN2	37	38	AIN3	GPIO_78	37	38	GPIO_79
AIN0	39	40	AIN1	GPIO_76	39	40	GPIO_77
GPIO_20	41	42	GPIO_7	GPIO_74	41	42	GPIO_75
DDND	43	44	DDND	GPIO_72	43	44	GPIO_73
DDND	45	46	DDND	GPIO_70	45	46	GPIO_71

For our circuit, we used:

- **P9_2** for the ground
- **P9_4** for the circuit's power source
- **12** Through-Hole Resistors(10k)
- **1** 470ohm current limiting resistor (connect from d to f at the bottom in the breadboard below)

Below is a picture of the circuit:



IMPORTANT:

In the beginning, we used **P8_31, P8_41, P8_43 and P8_45** were used as GPIO input. However, our group had some problems booting up our BeagleBone Green when those pins are used as GPIO input. So, it is not recommended to use the above 4 pins as GPIO input.

Below are the pins we used to connect to the 12key keypad.

Ground	Power source
P9_2	P9_4

0	1	2	3	4	5	6	7	8	9	*	#
P8_7	P8_9	P8_7	P8_0	P8_7	P8_5	P8_9	P8_0	P8_3	P8_8	P8_8	P8_9

3. C code

Below is the code that demonstrates how to initialize all the GPIO pins. When a button is pressed on the keypad, its corresponding GPIO pin will read 1, 0 otherwise. There are many ways you can read and store user input. One way is to loop through and read all the values for all GPIO pins and see if it reads 1. The actual implementation depends on you.

Note: the **export_to_gpio()** function is a helper function we defined that will write the pin number to the export file. Same as executing the command “echo 88 > /sys/class/gpio/export”. 88 can be replaced with any pin number.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
#include "../utility/utility.h"
#include "keypad.h"

#define GPIO_BASE_DIR "/sys/class/gpio"
#define NUM_KEYPAD_GPIO 12
#define MAX_BUFFER_SIZE 256
#define SLEEP_TIME_MS 300

static const char *KEYPAD_GPIO[NUM_KEYPAD_GPIO] = {
    "88", "89", "86", "87", "66", "9",
    "8", "78", "76", "67", "69", "68"
};

typedef enum keypad_element {
    star, seven, four, one, zero, eight,
    five, two, hash, nine, six, three
} keypad_element;

static const int KEYPAD_KEYS[NUM_KEYPAD_GPIO] = {
    star, 7, 4, 1, 0, 8, 5, 2, hash, 9, 6, 3
};

void Keypad_init(void) {
    for (int i = 0; i < NUM_KEYPAD_GPIO; i++) {
        export_to_gpio(KEYPAD_GPIO[i]);
        my_sleep_ms(SLEEP_TIME_MS);
    }

    my_sleep_ms(SLEEP_TIME_MS);

    for (int i = 0; i < NUM_KEYPAD_GPIO; i++) {
        // set direction
        char tempString[MAX_BUFFER_SIZE];
        snprintf(tempString, MAX_BUFFER_SIZE - 1, "/sys/class/gpio/gpio%s/direction", KEYPAD_GPIO[i]);
        file_write(tempString, "in");
    }

    printf("Keypad initialized\n");
}
```

4. Reference

1. <https://beagleboard.org/support/bone101>
2. <https://opencoursehub.cs.sfu.ca/bfraser/grav-cms/cmpt433/links/files/2017-student-howtos/12-DigitKeypadGuide.pdf>