```c
/*
 * CE2812 - 021
 * Winter 2016
 * Lab 4 - KEYPAD API
 * Name: Yahui Liang
 * Created: 01/05/2017
 */

/* Header files */
#include <inttypes.h>
#include "keypad_api.h"
#include "reg_struct.h"

static volatile GPIOx* gpioc = (GPIOx*) GPIOC_BASE;

/**
 * Enable the clock for GPIOC, and pull up rows.
 */
void key_init() {
	/* enable the clock */
	volatile uint32_t* ahb1enr = (uint32_t*) RCC_AHB1ENR;
	*ahb1enr |= 1 << 2;
	/* sets mode for pins */
	gpioc -> MODER &= ~0b1111111111111111;
	gpioc -> MODER |= 0b01010101;
	/* pull up rows */
	gpioc -> PUPDR &= ~0b1111111111111111; // make sure bits are set to be
zero.
	gpioc -> PUPDR |= (0b01010101 << 8); // set rows to be 1.
}

/**
 * Returns the value of the key which is pressed. If no key is
 * pressed, 0 will be returned.
 */
int key_getkey_noblock() {
	int four_set_bits = 0b1111;
	int clear_bit = 1;
	int pressed = 0;
	int key_value = 0;
	/* The algorithm for determining which key is being pressed. */
	for (int col = 1; col <= 4 && pressed == 0; col++) {
		int col_logic = four_set_bits & ~(clear_bit << (col - 1));
		gpioc -> BSRR |= 0b1111 << 16; // clear previous odr logic.
		gpioc -> BSRR |= col_logic; // set new logic.
		for (int row = 1; row <= 4 && pressed == 0; row++) {
			int row_logic = (gpioc -> IDR >> 4) & ~0xFFF0; // get the
row logic for comparing.
			/* determine if the current row is being pressed. */
			if (row_logic == (four_set_bits & ~(clear_bit << (row -
1)))) {
				pressed = 1;
				key_value = (row - 1) * 4 + col;
			}
		}
	}
	return key_value;
```

```
}

/**
 * Returns the value of the key which is pressed. However, the method blocks
if no key is
 * pressed.
 */
int key_getkey() {
	int key_value = 0;
	while (key_value == 0) {
		key_value = key_getkey_noblock();
		int row_logic = 0;
		/* repeat determining if the key has been released. */
		while (!(row_logic == 0b1111)) {
			row_logic = (gpioc -> IDR >> 4) & ~0xFFF0;
		}
	}
	return key_value;
}

/**
 * The method returns the character of the key which is pressed.
 */
char key_getchar() {
	const char keys[] = {'1', '2', '3', 'A', '4', '5', '6', 'B', '7', '8',
'9', 'C', '*', '0', '#', 'D'};
	return keys[key_getkey() - 1];
}
```