void HAL\_GPIO\_EXTI\_Callback(uint16\_t pin) {

if (pin == GPIO\_PIN\_2) { // Proximity sensor interrupt

uint8\_t int\_src = 0;

// Read interrupt source register

sx1509\_read(REG\_INTERRUPT\_SOURCE\_A, &int\_src);

// Check which sensor triggered the interrupt

if (int\_src & 0x01) {

printf("Proximity sensor 1 triggered the interrupt\n");

}

if (int\_src & 0x02) {

printf("Proximity sensor 2 triggered the interrupt\n");

}

if (int\_src & 0x04) {

printf("Proximity sensor 3 triggered the interrupt\n");

}

if (int\_src & 0x08) {

printf("Proximity sensor 4 triggered the interrupt\n");

}

if (int\_src & 0x10) {

printf("Proximity sensor 5 triggered the interrupt\n");

}

if (int\_src & 0x20) {

printf("Proximity sensor 6 triggered the interrupt\n");

}

// Clear interrupt source register

sx1509\_write(REG\_INTERRUPT\_SOURCE\_A, 0xFF);

} else if (pin == GPIO\_PIN\_4) { // Keypad interrupt

uint8\_t key\_data\_1 = 0, key\_data\_2 = 0;

// Read keypad data registers

sx1509\_read(REG\_KEY\_DATA\_1, &key\_data\_1);

sx1509\_read(REG\_KEY\_DATA\_2, &key\_data\_2);

// Check which key was pressed

for (int i = 0; i < 4; i++) {

for (int j = 0; j < 4; j++) {

if (key\_data\_1 & (1 << j) && key\_data\_2 & (1 << i)) {

printf("Key %d pressed\n", i\*4 + j + 1);

}

}

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Online C Compiler.

Code, Compile, Run and Debug C program online.

Write your code in this editor and press "Run" button to compile and execute it.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <stdint.h>

int reg2index(uint8\_t ind)

{

uint8\_t temp = ind;

temp = ~temp;

temp = temp & 0x0F;

temp = temp-1;

int count = 0;

while (temp) {

count += temp & 1;

temp >>= 1;

}

return count;

}

void printer(uint16\_t val)

{

uint16\_t x = val;

for (int i = 0; i < 16; i++) {

printf("%d", (x & 0x8000) >> 15);

x <<= 1;

}

printf("\n");

}

int countSetBits(uint8\_t n)

{

int count = 0;

while (n) {

count += n & 1;

n >>= 1;

}

return count;

}

#define val 0xA0

void main()

{

uint8\_t z = val & 0xFF;

printer(z);

z = ~z;

printer(z);

z = z & 0x0F;

//printer(z);

z = z-1;

printer(z);

printf("Original: %d\n\n", countSetBits(z));

uint8\_t t = val & 0x0F;

printf("Mod: %d\n", reg2index(t));

}

#define INITIAL\_WAIT\_TIME 400;

int actual\_wait = INITIAL\_WAIT\_TIME;

// EASY WAY

int frequency = 10;

actual\_wait = 1.0 / frequency;

INSIDE THE MAIN

/\* USER CODEWHILE \*/ // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CODE to make Led blinking\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

if(actual\_wait <= 0 )

actual\_wait = INITIAL\_WAIT\_TIME;

HAL\_GPIO\_WritePin(GPIOB, LD2\_Pin, GPIO\_PIN\_SET);

HAL\_Delay(actual\_wait);

HAL\_GPIO\_WritePin(GPIOB, LD2\_Pin, GPIO\_PIN\_RESET);

HAL\_Delay(actual\_wait);