   Wyatt Tack

Keanu Lam  
   EE 329-01 F'24

Group P  
   2024-Sept-28

**EE 329 A2**

This code is designed to validate the use of a keypad, using the same LED array used in A2. The code functions through polling to determine if a key is pressed, and if so, goes into a subroutine to determine which key is pressed by scrolling through each column, then each row. This program was helpful for organizing file structure in C, making various source and header files, as well as writing multiple #define statements to make code more legible.

**Link to YouTube Presentation:**

<https://youtu.be/oWzriSy2BYs>

**Obtained Data:**

Figure A2(a): 3x4 Keypad and 4xLED Wiring

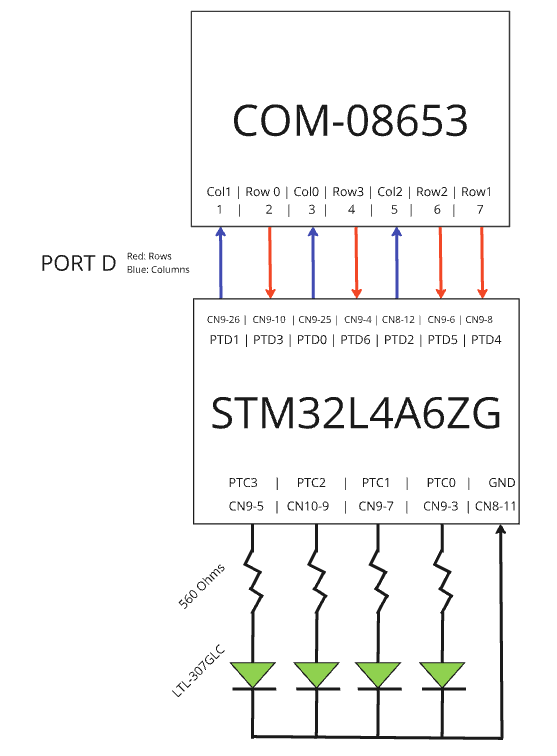


Table A2(a): Pull-Up/Pull-Down resistances found in pg.176 of STM32 Datasheet

|  |  |  |  |
| --- | --- | --- | --- |
| Resistance | **Min** | **Nominal** | **Max** |
| **Pull Up** | 25 kΩ | 40 kΩ | 55 kΩ |
| **Pull Down** | 25 kΩ | 40 kΩ | 55 kΩ |

**Psuedocode Flow Chart:**

**Main:**

initializeLEDGPIO()

initializeKeypadGPIO()

loop

ifKeyPressed

detectWhichKeyPressed

ifNotValid 🡪 display zero

ifValid 🡪 display key pressed

**ifKeyPressed:**

columns = high

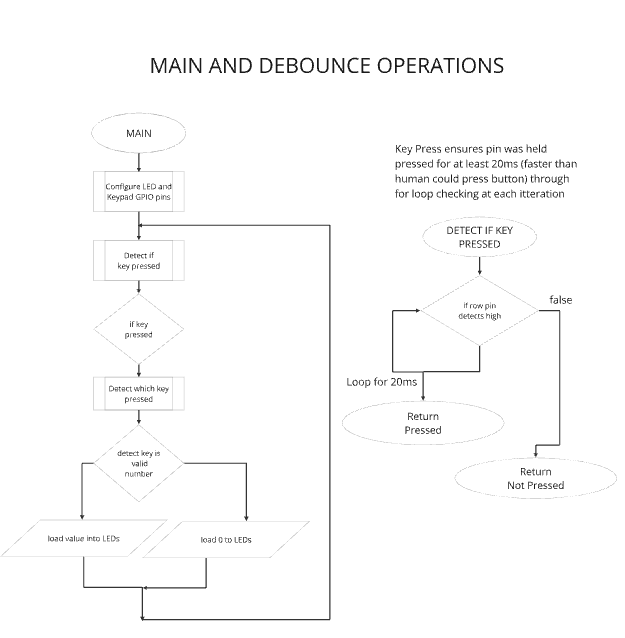
poll rows

if row high

read high for 10ms (debounce)

if high for 10ms 🡪 return pressed

else 🡪 return not pressed



**Formatted Source Code main.h:**

---------------------------------------------------------------------------------------

main.h

---------------------------------------------------------------------------------------

/\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* @file : main.h

\* project : EE329 Lab A2

\* author : Wyatt Tack (wwt) - wtack@calpoly.edu

\* date : 9/27/2024

\* firmware : ST-Link V1

\* @attention : Copyright (c) 2024 STMicroelectronics. All rights reserved.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*

\* main header for defines for C and stm32 headers/hal

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

#ifndef \_\_MAIN\_H

#define \_\_MAIN\_H

#ifdef \_\_cplusplus

extern "C" {

#endif

/\* Includes ------------------------------------------------------------------\*/

#include "stm32l4xx\_hal.h"

/\* Exported functions prototypes ---------------------------------------------\*/

void SystemClock\_Config(void);

void Error\_Handler(void);

#ifdef \_\_cplusplus

}

#endif

#endif

**Formatted Source Code main.c:**

---------------------------------------------------------------------------------------

main.c

---------------------------------------------------------------------------------------

/\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* @file : main.c

\* project : EE329 Lab A2

\* author : Wyatt Tack (wwt) - wtack@calpoly.edu

\* date : 9/27/2024

\* firmware : ST-Link V1

\* @attention : Copyright (c) 2024 STMicroelectronics. All rights reserved.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*

\* Takes in input from 3x4 (0-9 \*#) keypad, then writes the hex value of the

\* number to the LED array. LEDs display 0 if glitch, LEDs display 0xF for 0,

\* 0xA for \*, and 0xC for #

\*

\* Keypad defined at GPIOD Col PTC0-2, Row PTC3-6

\* LEDs defined at GPIOC PTC0-3

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

#include "main.h"

#include "keypad.h"

void Led\_Config(void);

int main(void)

{

// Initialize system clock, keypad registers, led registers, and held value

HAL\_Init();

SystemClock\_Config();

Keypad\_Config();

Led\_Config();

//uint8\_t currentKeyValue = 0;

//GPIOC->BSRR = GPIO\_PIN\_0;

while (1)

{

if (Keypad\_IsAnyKeyPressed())

{

if (Keypad\_WhichKeyIsPressed() > 0 && Keypad\_WhichKeyIsPressed() < 16){

//currentKeyValue = Keypad\_WhichKeyIsPressed();

//GPIOC->ODR = (currentKeyValue);

GPIOC->ODR = Keypad\_WhichKeyIsPressed();

}

else

GPIOC->ODR = (0x0);

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ledConfig \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Led\_Config(void)

{ // configure GPIO pins PC0-3 for:

// output mode, push-pull, no pull up or pull down, high speed

RCC->AHB2ENR |= (RCC\_AHB2ENR\_GPIOCEN);

GPIOC->MODER &= ~(GPIO\_MODER\_MODE0 | GPIO\_MODER\_MODE1

| GPIO\_MODER\_MODE2 | GPIO\_MODER\_MODE3);

GPIOC->MODER |= (GPIO\_MODER\_MODE0\_0 | GPIO\_MODER\_MODE1\_0

| GPIO\_MODER\_MODE2\_0 | GPIO\_MODER\_MODE3\_0);

GPIOC->OTYPER &= ~(GPIO\_OTYPER\_OT0 | GPIO\_OTYPER\_OT1

| GPIO\_OTYPER\_OT2 | GPIO\_OTYPER\_OT3);

GPIOC->PUPDR &= ~(GPIO\_PUPDR\_PUPD0 | GPIO\_PUPDR\_PUPD1

| GPIO\_PUPDR\_PUPD2 | GPIO\_PUPDR\_PUPD3);

GPIOC->OSPEEDR |= ((3 << GPIO\_OSPEEDR\_OSPEED0\_Pos) |

(3 << GPIO\_OSPEEDR\_OSPEED1\_Pos) |

(3 << GPIO\_OSPEEDR\_OSPEED2\_Pos) |

(3 << GPIO\_OSPEEDR\_OSPEED3\_Pos));

GPIOC->BRR = (GPIO\_PIN\_0 | GPIO\_PIN\_1

| GPIO\_PIN\_2 | GPIO\_PIN\_3);

}

// System

void SystemClock\_Config(void)

{

RCC\_OscInitTypeDef RCC\_OscInitStruct = {0};

RCC\_ClkInitTypeDef RCC\_ClkInitStruct = {0};

/\*\* Configure the main internal regulator output voltage

\*/

if (HAL\_PWREx\_ControlVoltageScaling(PWR\_REGULATOR\_VOLTAGE\_SCALE1) != HAL\_OK)

{

Error\_Handler();

}

/\*\* Initializes the RCC Oscillators according to the specified parameters

\* in the RCC\_OscInitTypeDef structure.

\*/

RCC\_OscInitStruct.OscillatorType = RCC\_OSCILLATORTYPE\_MSI;

RCC\_OscInitStruct.MSIState = RCC\_MSI\_ON;

RCC\_OscInitStruct.MSICalibrationValue = 0;

RCC\_OscInitStruct.MSIClockRange = RCC\_MSIRANGE\_6;

RCC\_OscInitStruct.PLL.PLLState = RCC\_PLL\_NONE;

if (HAL\_RCC\_OscConfig(&RCC\_OscInitStruct) != HAL\_OK)

{

Error\_Handler();

}

/\*\* Initializes the CPU, AHB and APB buses clocks

\*/

RCC\_ClkInitStruct.ClockType = RCC\_CLOCKTYPE\_HCLK|RCC\_CLOCKTYPE\_SYSCLK

|RCC\_CLOCKTYPE\_PCLK1|RCC\_CLOCKTYPE\_PCLK2;

RCC\_ClkInitStruct.SYSCLKSource = RCC\_SYSCLKSOURCE\_MSI;

RCC\_ClkInitStruct.AHBCLKDivider = RCC\_SYSCLK\_DIV1;

RCC\_ClkInitStruct.APB1CLKDivider = RCC\_HCLK\_DIV1;

RCC\_ClkInitStruct.APB2CLKDivider = RCC\_HCLK\_DIV1;

if (HAL\_RCC\_ClockConfig(&RCC\_ClkInitStruct, FLASH\_LATENCY\_0) != HAL\_OK)

{

Error\_Handler();

}

}

void Error\_Handler(void)

{

\_\_disable\_irq();

while (1)

{

}

}

#ifdef USE\_FULL\_ASSERT

void assert\_failed(uint8\_t \*file, uint32\_t line)

{

}

#endif

**Formatted Source Code keypad.h:**

---------------------------------------------------------------------------------------

keypad.h

---------------------------------------------------------------------------------------

/\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* @file : keypad.h

\* project : EE329 Lab A2

\* author : Wyatt Tack (wwt) - wtack@calpoly.edu

\* date : 9/27/2024

\* firmware : ST-Link V1

\* @attention : Copyright (c) 2024 STMicroelectronics. All rights reserved.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*

\* main header for defines for keypad.h

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

#ifndef INC\_KEYPAD\_H\_

#define INC\_KEYPAD\_H\_

#include "stm32l4xx\_hal.h"

//-------------------------------- KEYPAD Defines ------------------------------

#define COL\_PORT GPIOD

#define COL\_PINS (GPIO\_PIN\_0 | GPIO\_PIN\_1 | GPIO\_PIN\_2)

#define ROW\_PORT GPIOD

#define ROW\_PINS (GPIO\_PIN\_3 | GPIO\_PIN\_4 | GPIO\_PIN\_5 | GPIO\_PIN\_6)

#define SETTLE 1900 //defined as time for 20ms loop instruction (\*5.5us)

#define BIT0\_COL GPIO\_PIN\_0 //defined as first bit for columns

#define BIT0\_ROW GPIO\_PIN\_3 //defined as first bit for rows

#define NUM\_COLS 3

#define NUM\_ROWS 4

#define NO\_KEYPRESS 0x0

#define KEY\_ZERO 11

#define CODE\_ZERO 0xF

//--------------------------------- function prototypes ------------------------

void Keypad\_Config(void);

int Keypad\_IsAnyKeyPressed(void);

int Keypad\_WhichKeyIsPressed(void);

#endif /\* INC\_KEYPAD\_H\_ \*/

**Formatted Source Code keypad.c:**

---------------------------------------------------------------------------------------

keypad.c

---------------------------------------------------------------------------------------

/\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* @file : keypad.c

\* project : EE329 Lab A2

\* author : Wyatt Tack (wwt) - wtack@calpoly.edu

\* date : 9/27/2024

\* firmware : ST-Link V1

\* @attention : Copyright (c) 2024 STMicroelectronics. All rights reserved.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*

\* Keypad pulling function source file provided on behalf of the EE329 lab

\* manual. Adapted from EE329 lab manual.

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

#include "keypad.h"

// ------------------------------------- modified from excerpt from keypad.c ---

void Keypad\_Config(void){//must be manually changed if separate GPIO port is used

//set for port D as current config

//Port clock initialize

RCC->AHB2ENR |= (RCC\_AHB2ENR\_GPIODEN);

//Column pin initialize - Push Pull, no PU/PD, high speed

COL\_PORT->MODER &= ~(GPIO\_MODER\_MODE0 | GPIO\_MODER\_MODE1

| GPIO\_MODER\_MODE2);

COL\_PORT->MODER |= (GPIO\_MODER\_MODE0\_0 | GPIO\_MODER\_MODE1\_0

| GPIO\_MODER\_MODE2\_0);

COL\_PORT->OTYPER &= ~(GPIO\_OTYPER\_OT0 | GPIO\_OTYPER\_OT1

| GPIO\_OTYPER\_OT2 | GPIO\_OTYPER\_OT3);

COL\_PORT->PUPDR &= ~(GPIO\_PUPDR\_PUPD0 | GPIO\_PUPDR\_PUPD1

| GPIO\_PUPDR\_PUPD2 | GPIO\_PUPDR\_PUPD3);

COL\_PORT->OSPEEDR |= ((GPIO\_OSPEEDR\_OSPEED0) |

(GPIO\_OSPEEDR\_OSPEED1) |

(GPIO\_OSPEEDR\_OSPEED2));

//Row pin initialize - Input, pull down

ROW\_PORT->MODER &= ~(GPIO\_MODER\_MODE3 | GPIO\_MODER\_MODE4

| GPIO\_MODER\_MODE5 | GPIO\_MODER\_MODE6);

ROW\_PORT->PUPDR &= ~(GPIO\_PUPDR\_PUPD3 | GPIO\_PUPDR\_PUPD4

| GPIO\_PUPDR\_PUPD5 | GPIO\_PUPDR\_PUPD6);

ROW\_PORT->PUPDR |= (GPIO\_PUPDR\_PUPD3\_1 | GPIO\_PUPDR\_PUPD4\_1

| GPIO\_PUPDR\_PUPD5\_1 | GPIO\_PUPDR\_PUPD6\_1);

}

// -----------------------------------------------------------------------------

int Keypad\_IsAnyKeyPressed(void) {

// drive all COLUMNS HI; see if any ROWS are HI

// return true if a key is pressed, false if not

COL\_PORT->BSRR = COL\_PINS; // set all columns HI

for ( uint16\_t idx=0; idx<SETTLE; idx++ ) // let it settle

;

if ((ROW\_PORT->IDR & ROW\_PINS) != 0 ) { // got a keypress!

for ( uint16\_t idx=0; idx < SETTLE; idx++ ){

if ((ROW\_PORT->IDR & ROW\_PINS) == 0 ) return( 0 );

} // if key held for 20ms then return 1 (debounce)

return( 1 );

}

else

return( 0 ); // nope.

}

// -----------------------------------------------------------------------------

int Keypad\_WhichKeyIsPressed(void) {

// detect and encode a pressed key at {row,col}

// assumes a previous call to Keypad\_IsAnyKeyPressed() returned TRUE

// verifies the Keypad\_IsAnyKeyPressed() result (no debounce here),

// determines which key is pressed and returns the encoded key ID

int8\_t iRow=0, iCol=0, iKey=0; // keypad row & col index, key ID result

int8\_t bGotKey = 0; // bool for keypress, 0 = no press

COL\_PORT->BSRR = COL\_PINS; // set all columns HI

for ( iRow = 0; iRow < NUM\_ROWS; iRow++ ) { // check all ROWS

if ( ROW\_PORT->IDR & (BIT0\_ROW << iRow) ) { // keypress in iRow!!

COL\_PORT->BRR = ( COL\_PINS ); // set all cols LO

for ( iCol = 0; iCol < NUM\_COLS; iCol++ ) { // 1 col at a time

COL\_PORT->BSRR = ( BIT0\_COL << (iCol) ); // set this col HI

if ( ROW\_PORT->IDR & (BIT0\_ROW << iRow) ) { // keypress in iCol!!

bGotKey = 1;

break; // exit for iCol loop

}

}

if ( bGotKey )

break;

}

}

// encode {iRow,iCol} into LED word : row 1-3 : numeric, ‘1’-’9’

// row 4 : ‘\*’=10, ‘0’=15, ‘#’=12

// no press: send NO\_KEYPRESS

if ( bGotKey ) {

iKey = ( iRow \* NUM\_COLS ) + iCol + 1; // handle numeric keys ...

if ( iKey == KEY\_ZERO ) // works for ‘\*’, ‘#’ too

iKey = CODE\_ZERO;

return( iKey ); // return encoded keypress

}

return( NO\_KEYPRESS ); // unable to verify keypress

}