Below is a **README** document for your 4×4 Matrix Keypad project on the **STM32F401RE**. You can copy and paste it into a file named README.md, or you can open it in Microsoft Word and save it as README.docx if you want a Word document. Feel free to adjust formatting, add images, or include any additional details that suit your project.

**4×4 Matrix Keypad Interface on STM32F401RE**

This project demonstrates how to interface a **4×4 matrix keypad** with an **STM32F401RE** (e.g., a Nucleo-F401RE board) using **direct register access**. When a key is pressed on the keypad, an on-board LED blinks a number of times corresponding to the key.

**Table of Contents**

1. [Overview](https://chatgpt.com/c/67ac9336-5850-8000-b005-1be99daeb42b#overview)
2. [Hardware Setup](https://chatgpt.com/c/67ac9336-5850-8000-b005-1be99daeb42b#hardware-setup)
3. [How It Works](https://chatgpt.com/c/67ac9336-5850-8000-b005-1be99daeb42b#how-it-works)
4. [Project Structure](https://chatgpt.com/c/67ac9336-5850-8000-b005-1be99daeb42b#project-structure)
5. [Building and Running](https://chatgpt.com/c/67ac9336-5850-8000-b005-1be99daeb42b#building-and-running)
6. [Customization](https://chatgpt.com/c/67ac9336-5850-8000-b005-1be99daeb42b#customization)
7. [Troubleshooting](https://chatgpt.com/c/67ac9336-5850-8000-b005-1be99daeb42b#troubleshooting)
8. [License](https://chatgpt.com/c/67ac9336-5850-8000-b005-1be99daeb42b#license)
9. [References](https://chatgpt.com/c/67ac9336-5850-8000-b005-1be99daeb42b#references)

**1. Overview**

* **MCU**: STM32F401RE (e.g., Nucleo-F401RE board)
* **Keypad**: 4×4 matrix keypad
* **GPIO Usage**:
  + **PC0–PC3** = Keypad columns (inputs with pull-ups)
  + **PC4–PC7** = Keypad rows (driven one at a time)
  + **PA5** = On-board LED (LD2 on Nucleo-F401RE)
* **Clock**: 84 MHz system clock (default on the Nucleo-F401RE)
* **Language**: C (direct register access via CMSIS)

**Behavior**: When you press a key, the code calculates a key number (from **1** to **16**) and blinks the on-board LED that many times. If no key is pressed, the LED remains off.

**2. Hardware Setup**

Below is a sample wiring guide. Make sure your **4×4** keypad pins match the corresponding MCU pins (or vice versa).

| **Keypad Pin** | **MCU Pin** | **Direction** | **Notes** |
| --- | --- | --- | --- |
| Col0 | PC0 | Input (pull-up) |  |
| Col1 | PC1 | Input (pull-up) |  |
| Col2 | PC2 | Input (pull-up) |  |
| Col3 | PC3 | Input (pull-up) |  |
| Row0 | PC4 | Output (driven low) | Driven one at a time |
| Row1 | PC5 | Output (driven low) |  |
| Row2 | PC6 | Output (driven low) |  |
| Row3 | PC7 | Output (driven low) |  |
| (LED) | PA5 | Output to LED | On-board LED (LD2 on Nucleo-F401RE) |

**Power**: The keypad itself usually doesn’t need a separate power pin. The rows/columns lines are driven by the MCU pins. Make sure you also have the Nucleo board connected via USB for power and programming.

**3. How It Works**

1. **Initialization**:
   * **keypad\_init()**: Configures **PC0..PC3** as inputs with internal pull-ups and **PC4..PC7** as outputs (though they are only driven one row at a time).
   * **LED\_init()**: Configures **PA5** as output to drive the on-board LED.
2. **Key Scanning**:
   * **keypad\_getkey()**:
     1. It drives **all rows** (PC4..PC7) low and reads the columns (PC0..PC3). If all columns are high (0xF), no key is pressed.
     2. If any column is low, it activates each row one by one to see which column becomes low. This identifies exactly which key is pressed (row × 4 + column).
   * Returns a unique number (1–16) depending on which key is pressed, or 0 if none.
3. **LED Blinking**:
   * **LED\_blink()**: Blinks the on-board LED **N** times, where **N** is the key’s numeric code (value). For example, if the key code is **5**, the LED will blink 5 times.
4. **Delays**:
   * **delayMs()** is a simple software loop. The code is calibrated for an ~84 MHz system clock by using ~16,800 loops per ms.
   * **delay()** is a very short delay to let signals stabilize during row/column driving.

**4. Project Structure**

A minimal directory layout could be:

.

├─ Inc/

│ └─ stm32f4xx.h // or stm32f401xe.h, included by default in CMSIS

│

├─ Src/

│ └─ main.c // contains keypad code & main program

│

└─ README.md (this file)

If you’re using an IDE like **Keil uVision** or **STM32CubeIDE**, you may have auto-generated folders (e.g., Core, Drivers, etc.). Place **main.c** in the appropriate Src folder, and ensure the system clock is set to 84 MHz.

**5. Building and Running**

1. **Open** your STM32 project in Keil uVision, STM32CubeIDE, or another CMSIS-compatible environment.
2. **Add** or replace the provided main.c to the Src directory.
3. **Check** your **Options for Target** / **Project Settings** to confirm the clock is 84 MHz, or to ensure the default startup file configures it correctly.
4. **Compile** the project.
5. **Connect** the Nucleo-F401RE board via USB and **flash** the code.
6. **Open** a power source or keep USB connected (the board should be powered).

Now, pressing a key on the 4×4 keypad should cause the on-board LED to blink a number of times corresponding to that key code.

**6. Customization**

* **Key Mappings**:  
  By default, the example returns **row \* 4 + column** (1-based). You can remap or interpret these numbers to match a typical keypad layout (e.g., 1,2,3,A / 4,5,6,B / etc.).
* **LED Behavior**:  
  You could print the key code via UART instead of blinking an LED, or drive additional LEDs.
* **Delay Accuracy**:  
  If you need more accurate timing, consider using **SysTick** or a hardware timer rather than a software loop.
* **Clock Frequency**:  
  If your board runs at a different clock speed, adjust the delayMs() function accordingly.

**7. Troubleshooting**

1. **No response when pressing keys**:
   * Check the **pull-ups** are enabled on the column pins.
   * Verify rows (PC4..PC7) are driven one at a time.
   * Ensure the physical wiring from the keypad to the MCU pins is correct.
2. **Incorrect or multiple keys read**:
   * Make sure each keypad pin matches the correct MCU pin.
   * Add more settling delay in delay() if the lines take longer to stabilize.
3. **LED never blinks**:
   * Check if **PA5** is properly set to output.
   * Look at the define for the LED pin in your code and confirm the Nucleo-F401RE on-board LED is indeed on PA5.
4. **Timing issues**:
   * If your clock is not 84 MHz, the delayMs() might be too short or too long. Adjust loops\_per\_ms in the code or use hardware timers.

**8. License**

This project is available under the [MIT License](https://opensource.org/licenses/MIT). You are free to modify, use, and distribute it in both commercial and non-commercial applications.

**9. References**

* **STM32F401RE**:
  + [Reference Manual (RM0368)](https://www.st.com/resource/en/reference_manual/dm00096844.pdf)
  + [Data Sheet & Nucleo-F401RE Board Info](https://www.st.com/en/evaluation-tools/nucleo-f401re.html)
* **Keypad**:
  + Typical 4×4 matrix documentation from electronics suppliers
* **Direct Register Access**:
  + [CMSIS / STM32F4 Standard Peripheral Registers](https://www.arm.com/why-arm/technologies/cmsis)
* **HD44780 LCD** (if referencing from previous labs) or other peripherals

**Enjoy your 4×4 matrix keypad project with the STM32F401RE!**