Here is a **README** file for your **STM32F401RE ADC1 Channel 1 (PA1) conversion** project. This file explains the project, setup, and usage in detail.

**ADC1 Channel 1 (PA1) Input Conversion & LED Control on STM32F401RE**

This project reads an **analog voltage from PA1** using **ADC1**, converts it to a **digital value**, and controls an **on-board LED (PA5, LD2)** based on the **8th bit** of the ADC result.

If **bit 8 of ADC result is set (>= 256 on a 12-bit scale)**:  
✅ **LED (PA5) turns ON**  
❌ Otherwise, **LED turns OFF**

**📌 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. [Software Explanation](https://chatgpt.com/c/67ac9336-5850-8000-b005-1be99daeb42b#software-explanation)
4. [Project Structure](https://chatgpt.com/c/67ac9336-5850-8000-b005-1be99daeb42b#project-structure)
5. [Building and Uploading](https://chatgpt.com/c/67ac9336-5850-8000-b005-1be99daeb42b#building-and-uploading)
6. [Usage](https://chatgpt.com/c/67ac9336-5850-8000-b005-1be99daeb42b#usage)
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 (Nucleo-F401RE)
* **ADC**: **ADC1 Channel 1 (PA1)**
* **Resolution**: 12-bit (0 to 4095)
* **Sampling Time**: Default (3 cycles)
* **Trigger Mode**: Software-triggered conversion
* **Clock Prescaler**: Default (divided by 2)

**Expected Behavior**

1. **ADC reads PA1 (0V to 3.3V) periodically**.
2. If **ADC result has bit 8 set (>= 256)**, LED (PA5) **turns ON**.
3. If **ADC result < 256**, LED **turns OFF**.

🔵 **Full-scale voltage range:**

* 0V → **ADC = 0** (LED OFF)
* 1.65V → **ADC ≈ 2048** (LED ON)
* 3.3V → **ADC = 4095** (LED ON for most readings)

**2. Hardware Setup**

| **Signal** | **STM32F401RE Pin** | **Description** |
| --- | --- | --- |
| **Analog Input** | **PA1 (ADC1 CH1)** | Connect 0-3.3V voltage |
| **LED Output** | **PA5 (LD2, On-board)** | Turns ON/OFF based on ADC reading |

**Connecting an External Signal**

* You can use **a potentiometer** (voltage divider) or **external sensor** to provide an analog voltage to **PA1**.
* Rotate the **potentiometer** to **change PA1 voltage** and observe LED behavior.

**3. Software Explanation**

**ADC Configuration**

* **PA1 set to analog mode** (GPIOA->MODER |= (3U << 2);).
* **Single-channel conversion (SQR3 = 1)**.
* **Software trigger (SWSTART)** is used.
* **Polling** EOC flag (while (!(ADC1->SR & (1U << 1))) {}) until conversion completes.

**LED Control Logic**

* **Reads ADC result**: result = ADC1->DR;
* **Checks bit 8 (0x100)** to decide LED state:
* if (result & 0x100)
* GPIOA->BSRR = (1U << 5); // Turn ON LED
* else
* GPIOA->BSRR = (1U << (5 + 16)); // Turn OFF LED

**4. Project Structure**

.

├── Inc/

│ └── stm32f4xx.h // CMSIS/Device headers

├── Src/

│ └── main.c // Contains ADC1 PA1 conversion & LED logic

└── README.md // This file

**5. Building and Uploading**

**Using Keil uVision / STM32CubeIDE**

1. Open **Keil uVision** or **STM32CubeIDE**.
2. Import or **create a new project** for STM32F401RE.
3. Copy main.c into the Src/ directory.
4. **Compile & Flash** the project to **Nucleo-F401RE**.

**6. Usage**

1. **Power the Nucleo-F401RE** via USB.
2. **Connect a potentiometer/sensor** to **PA1 (0-3.3V range).**
3. **Rotate the potentiometer** to vary PA1 voltage.
4. **Observe LED (LD2) behavior:**
   * **Low voltage (0-1V)** → LED **OFF**
   * **Mid voltage (~1.65V)** → LED **ON**
   * **High voltage (3.3V)** → LED **ON**

**7. Troubleshooting**

**🔴 LED Not Turning ON?**

✅ Check if **PA1 is receiving a valid analog voltage**.  
✅ Verify **ADC1 is enabled** (ADC1->CR2 |= 1;).  
✅ Ensure **PA1 is in analog mode** (GPIOA->MODER |= (3U << 2);).

**⚠️ Incorrect ADC Readings?**

✅ Confirm ADC **prescaler and sampling time** are correctly set.  
✅ **Try a different analog input voltage source**.  
✅ Add **debug print statements** to verify ADC results.

**⏳ Want Faster ADC Reads?**

✅ **Increase Sampling Time** (ADC1->SMPR2 or ADC1->SMPR1).  
✅ Use **DMA** to fetch ADC data without CPU polling.

**8. License**

This project is licensed under the **MIT License**.  
You are free to **modify, distribute, and use** the code.

**9. References**

📄 [STM32F401RE Datasheet](https://www.st.com/en/microcontrollers-microprocessors/stm32f401re.html)  
📄 [STM32 Reference Manual (RM0368)](https://www.st.com/resource/en/reference_manual/dm00096844.pdf)  
📄 [ADC Programming Guide](https://www.st.com/resource/en/programming_manual/dm00245755.pdf)

**🚀 Summary**

✅ **ADC1 reads PA1 (0-3.3V)**  
✅ **Bit 8 check (>= 256) controls LED ON/OFF**  
✅ **Works with potentiometer, sensors, or direct voltage input**

**🎯 Enjoy ADC programming on the STM32F401RE! 🎉**