**STM32F401RE: ADC Data Acquisition via DMA with Timer Trigger**

**Overview**

This project sets up **TIM2** to run at **1 kHz** and trigger **ADC1** conversions on **PA0** (ADC channel 0) in **8-bit** resolution. The **ADC** results are transferred to a memory buffer by **DMA2**. Once the buffer fills, the code prints the samples over **USART2** at **9600 baud**.

**Features**

1. **Timer-Triggered ADC (TIM2 → ADC1)**
2. **DMA2** to transfer ADC samples to adcbuf[] automatically
3. **8-bit** ADC data, each sample = 0..255
4. **Printing** over **USART2** (PA2) at **9600 baud**

**Hardware Setup**

**Pin Configuration**

| **Function** | **Pin** | **Description** |
| --- | --- | --- |
| **ADC1 Ch0** | **PA0** | Analog input, e.g. 0..3.3V |
| **TIM2 Ch2 Output** | **PB3 (AF1)** | 1 kHz PWM trigger to ADC |
| **USART2 Tx** | **PA2 (AF7)** | 9600 baud output to serial console |

**Connection Notes**

* **PA0** must be driven by an analog signal (e.g., pot) or left floating to read noise.
* **PB3** is used internally by the ADC as a trigger signal; can observe PWM if needed.
* **USART2 Tx** (PA2) can connect to a USB-Serial adapter (Rx) at 3.3V logic for console.

**Software Explanation**

**Sequence**

1. **DMA2\_init** & **TIM2\_init** & **ADC1\_init**.
2. **TIM2** runs at **1 kHz** → triggers **ADC1** conversions.
3. **ADC** results (8-bit) are **DMA**-transferred into adcbuf[64].
4. **DMA2\_Stream0\_IRQHandler** signals done → code prints them via USART2.

**Important Settings**

* **Timer**: PSC=160, ARR=100 → 1 kHz from 16 MHz.
* **ADC**: 8-bit resolution, triggered by **TIM2\_CH2** (rising edge).
* **DMA**: Stream0, Channel0 for **ADC1**, peripheral→memory, 64 samples.
* **USART2**: 9600 baud, 8N1 on **PA2**.

**Project Structure**

.

├── Inc/

│ └── stm32f4xx.h

├── Src/

│ └── main.c // Contains ADC+TIM2+DMA+USART code

└── README.md

**Building and Uploading**

**Using Keil uVision / STM32CubeIDE**

1. **Open your IDE** and create a new project for **STM32F401RE**.
2. **Copy** the main.c into Src/.
3. **Compile** and **flash** to your Nucleo-F401RE board.

**Usage**

**1️⃣ Flash the Code to STM32F401RE**

* In your IDE, build and program the board.

**2️⃣ Connect an Analog Signal**

* **PA0** can connect to a pot or some 0..3.3 V source.

**3️⃣ Open Serial Terminal at 9600 Baud**

* If hooking up a USB-Serial (Rx on PA2), or using ST-Link’s VCOM if bridging is done.
* Observe lines of ASCII containing the 64 samples, e.g. "012 045 123 ...".

**Troubleshooting**

**🔴 No Data?**

✅ **Ensure ADC input** on **PA0** is not floating (or do so to watch noise changes!). ✅ **Check console** at **9600 baud**.

**⚠️ Incorrect Values?**

✅ **Check** that your system clock is indeed 16 MHz. ✅ **Confirm** Timer PSC/ARR → 1 kHz. ✅ **Verify** 8-bit ADC range 0..255.

**License**

This project is licensed under the **MIT License**.

**References**

* [STM32F401RE Datasheet](https://www.st.com/en/microcontrollers-microprocessors/stm32f401re.html)
* [Reference Manual (RM0368)](https://www.st.com/resource/en/reference_manual/dm00096844.pdf)
* [ADC + TIM + DMA Info](https://www.st.com/resource/en/programming_manual/dm00245755.pdf)

**🚀 Summary**

* **Timer2** triggers **ADC** @1 kHz.
* **DMA2** collects 64 samples (8-bit each) into adcbuf[].
* **USART2** prints them at **9600** baud.

**Enjoy data acquisition with STM32F401RE!**