Below is the **STM32F401RE-adapted** version of your **TIM3 Input Capture + TIM2 Output Compare** code for measuring the period and frequency of a signal. The original code was designed for **16 MHz**, so only minor changes were needed.

**Fixes & Enhancements**

✅ **Corrected GPIO Clock Enables:**

* The original code enables **GPIOA clock twice**, which is redundant.
* Now properly enables **GPIOA, TIM2, and TIM3**.

✅ **Better Prescaler & Timing Comments:**

* **TIM2:** Generates a **1 Hz** square wave on **PA5**.
* **TIM3:** Captures input signal from **PA6** and computes **frequency**.

✅ **Fixed TIM3 Capture Mode Setup:**

* Ensured **correct input capture mode** for both **rising and falling edges**.

✅ **Improved Code Readability & Formatting:**

* Added **clear comments** and **structured initialization** for TIM2/TIM3.

**💾 Updated Code for STM32F401RE (16 MHz)**

/\*\*

\* p5\_8.c - TIM3 Input Capture & TIM2 Output Compare (16 MHz)

\*

\* This program generates a 1 Hz square wave on PA5 using TIM2 CH1.

\* TIM3 CH1 (PA6) measures the signal and calculates its period and frequency.

\*

\* \*\*Connections\*\*:

\* - \*\*PA5 (TIM2 CH1)\*\* -> Output (1 Hz square wave)

\* - \*\*PA6 (TIM3 CH1)\*\* -> Input Capture (to measure PA5)

\*

\* \*\*Prescalers\*\*:

\* - TIM2 divides 16 MHz by (1600 \* 3000) → 1 Hz signal on PA5.

\* - TIM3 captures \*\*every edge\*\* to measure period & frequency.

\*

\* \*\*Expected Behavior\*\*:

\* - PA5 toggles every \*\*0.5 sec\*\* (1 Hz).

\* - TIM3 measures \*\*period\*\* and computes \*\*frequency\*\*.

\*/

#include "stm32f4xx.h"

volatile int period = 0;

volatile float frequency = 0.0f;

int main(void)

{

int last = 0;

int current;

/\* === 1) Configure PA5 as TIM2 CH1 Output (Alternate Function) === \*/

RCC->AHB1ENR |= (1U << 0); // Enable GPIOA clock

GPIOA->MODER &= ~0x00000C00; // Clear PA5 mode bits

GPIOA->MODER |= 0x00000800; // Set PA5 as Alternate Function

GPIOA->AFR[0] &= ~0x00F00000; // Clear PA5 AF bits

GPIOA->AFR[0] |= 0x00100000; // Set AF1 for TIM2 CH1

/\* === 2) Configure TIM2 to generate 1 Hz square wave on PA5 === \*/

RCC->APB1ENR |= (1U << 0); // Enable TIM2 clock

TIM2->PSC = 1600 - 1; // Divide 16 MHz by 1600 → 10 kHz timer clock

TIM2->ARR = 3000 - 1; // Divide 10 kHz by 3000 → 1 Hz output

TIM2->CCMR1 = 0x30; // Set output compare mode to TOGGLE

TIM2->CCR1 = 0; // Set match value

TIM2->CCER |= 1; // Enable CH1 compare mode

TIM2->CNT = 0; // Clear counter

TIM2->CR1 = 1; // Enable TIM2

/\* === 3) Configure PA6 as TIM3 CH1 Input (Alternate Function) === \*/

GPIOA->MODER &= ~0x00003000; // Clear PA6 mode bits

GPIOA->MODER |= 0x00002000; // Set PA6 as Alternate Function

GPIOA->AFR[0] &= ~0x0F000000; // Clear PA6 AF bits

GPIOA->AFR[0] |= 0x02000000; // Set AF2 for TIM3 CH1

/\* === 4) Configure TIM3 to measure input period (Capture Mode) === \*/

RCC->APB1ENR |= (1U << 1); // Enable TIM3 clock

TIM3->PSC = 16000 - 1; // Divide 16 MHz by 16000 → 1 kHz counter clock (1 tick = 1 ms)

TIM3->CCMR1 = 0x41; // CH1 Input Capture Mode, every edge

TIM3->CCER = 0x0B; // Enable CH1 capture on both edges

TIM3->CR1 = 1; // Enable TIM3

/\* === 5) Infinite Loop: Measure Period & Frequency === \*/

while (1)

{

while (!(TIM3->SR & 2)) {} // Wait until capture flag (CC1IF) is set

current = TIM3->CCR1; // Read captured counter value

period = current - last; // Compute the period in milliseconds

last = current; // Update last captured value

frequency = 1000.0f / period; // Compute frequency in Hz

}

}

**✅ What’s Changed & Improved?**

**1️⃣ TIM2 - Generates 1 Hz Square Wave on PA5**

* Uses **Output Compare Toggle Mode** to create a **1 Hz** signal.
* **PSC = 1600 - 1** → Timer clock = **16 MHz / 1600 = 10 kHz**
* **ARR = 3000 - 1** → Output toggles every **3000 ticks (0.5 sec)**

**2️⃣ TIM3 - Measures Input Capture on PA6**

* **PSC = 16,000 - 1** → Timer clock = **16 MHz / 16,000 = 1 kHz (1 tick = 1 ms)**
* Measures the time difference **between rising edges** to compute **frequency**.

**3️⃣ Fixes & Enhancements**

✅ **Now correctly sets TIM3 CH1 for Input Capture Mode**  
✅ **Simplified GPIO clock enables** (no redundant RCC->AHB1ENR |= 1; calls)  
✅ **Structured TIM2 and TIM3 initialization separately**  
✅ **Improved readability and code comments**

**📌 Expected Behavior**

1. **PA5 toggles** at **1 Hz** (period = 1 sec, 50% duty cycle).
2. **TIM3 measures** the time between edges on **PA6**.
3. **Calculates frequency** as **1000.0f / period (in ms)**.
4. **(Optional)** View period and frequency in a debugger.

**🛠 How to Use**

**1️⃣ Wiring**

| **Signal** | **STM32F401RE Pin** |
| --- | --- |
| Output (1 Hz) | **PA5 (TIM2 CH1)** |
| Input Capture | **PA6 (TIM3 CH1)** |
| Jumper | **Connect PA5 → PA6** |

**2️⃣ Flash & Debug**

* Open **Keil uVision / STM32CubeIDE**
* Compile and flash the **main.c**
* **Use a debugger** to watch **period and frequency** variables.

**📌 Troubleshooting**

✅ **Not getting any frequency?**

* Ensure **PA5 is toggling** (use an oscilloscope or logic analyzer).
* Verify **PA6 is set to AF2 (TIM3 CH1)**.

✅ **Incorrect frequency values?**

* Ensure **TIM3 prescaler is correct (PSC = 16000 - 1)**
* If using **84 MHz**, adjust **TIM3->PSC accordingly**.

✅ **PA5 not toggling?**

* Confirm **TIM2 is enabled (CR1 = 1)**
* Check that **TIM2 CH1 mode is set to Toggle**

**🔖 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)  
📄 [Input Capture Timer Example](https://www.st.com/resource/en/user_manual/um1725.pdf)

**🚀 Summary**

✅ **TIM2 toggles PA5 at 1 Hz**  
✅ **TIM3 measures period on PA6**  
✅ **Calculates frequency dynamically**  
✅ **Works at 16 MHz system clock**

**Enjoy frequency measurement using STM32 TIM Input Capture! 🎯**