实验三：ADC 与 DMA 实验

一、 实验目的

(1) 了解 STM32-F1 系列处理器定时器 +ADC+DMA 工作原理及使用方法。

(2) 对 ADC 数据进行简单的处理、计算。

二、 实验内容

通过定时器 3 定时启动 ADC，自 DMA 缓冲区读取 ADC1.1 结果，保存、计算。

三、 目标要求

(1) 设定恰当采样率（如 2499+ 学号个位 7），以此采样率得到的 ADC 采样结果（PA1 通道），陆续保存至循环缓冲区 md.ADdat[0-7], 并在定时中断处理程序计算 8 点数据平均值保存到 ADdat[15]。

(2) 调整电位器，观察实验结果。

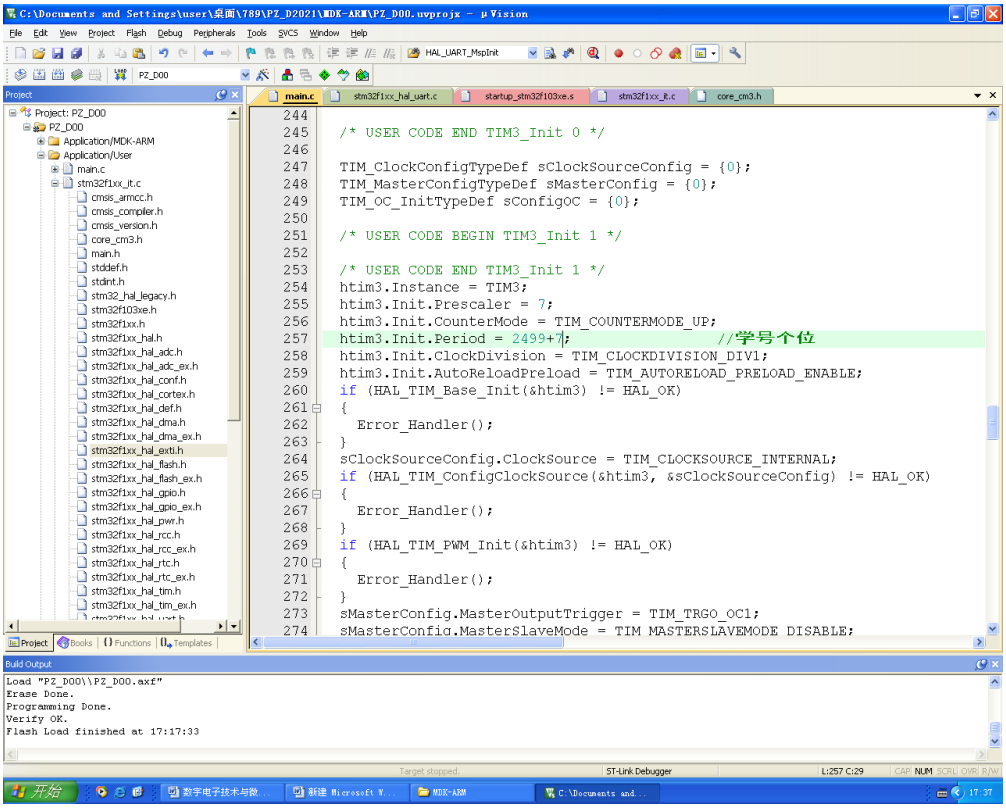

图 1: 采样率预设

四、 原理简述

ADC 模数转换模块是一种将模拟信号转换为数字信号进行处理的模块。在存储或传输时，模数转换器几乎必不可少。

ADC 经常和 DMA 共同使用，用于模拟到数字信号的获取和转换。

配置 ADC：对于 ADC 通道，每个 ADC 通道对应一个 GPIO 引脚端口，ADC 的输入需要使用模拟信号输入，所以，把 ADC1 的通道使用的 GPIO 引脚配置成模拟输入模式，GPIO 的引脚在设为模拟输入模式后可用于模拟电压的输入。

配置 DMA：内存直接访问。使用 DMA 的通道 1，数据从 ADC 的数据寄存器（ADC1_DR_Address）转移到内存（ADC_ConvertedValue 变量）当中, 内存外设地址都固定，使用 DMA 循环传输模式。

ADC 设置成连续转换模式，对应的 DMA 通道开启循环模式，ADC 就一直在进行数据采集然后通过 DMA 把数据搬运至内存。

加入定时中断，来定时读取内存中的数据。设置好定时器的触发间隔，就能实现 ADC 定时采样转换。

五、 实验程序

```
static void MX_TIM3_Init(void)
{
 /* USER CODE BEGIN TIM3_Init 0 */
 /* USER CODE END TIM3_Init 0 */
 TIM_ClockConfigTypeDef sClockSourceConfig = {0};
 TIM_MasterConfigTypeDef sMasterConfig = {0};
 TIM_OC_InitTypeDef sConfigOC = {0};
 /* USER CODE BEGIN TIM3_Init 1 */
 /* USER CODE END TIM3_Init 1 */
 htim3.Instance = TIM3;
 htim3.Init.Prescaler = 7;
 htim3.Init.CounterMode = TIM_COUNTERMODE_UP;
 htim3.Init.Period = 2499+7;
 htim3.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
 htim3.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_ENABLE;
 if (HAL_TIM_Base_Init(&htim3) != HAL_OK)
 {
 Error_Handler();
 }
 sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
 if (HAL_TIM_ConfigClockSource(&htim3, &sClockSourceConfig) != HAL_OK)
 {
 Error_Handler();
 }
 if (HAL_TIM_PWM_Init(&htim3) != HAL_OK)
 {
 Error_Handler();
 }
 sMasterConfig.MasterOutputTrigger = TIM_TRGO_OC1;
```

```
sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
if (HAL_TIMEx_MasterConfigSynchronization(&htim3, &sMasterConfig) != HAL_OK)
{
Error_Handler();
}
sConfigOC.OCMode = TIM_OCMODE_PWM1;
sConfigOC.Pulse = 6;
sConfigOC.OCPolarity = TIM_OCPOLARITY_HIGH;
sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
if (HAL_TIM_PWM_ConfigChannel(&htim3, &sConfigOC, TIM_CHANNEL_2) != HAL_OK)
{
Error_Handler();
}
/* USER CODE BEGIN TIM3_Init 2 */
/* USER CODE END TIM3_Init 2 */
HAL_TIM_MspPostInit(&htim3);
}
```

六、 实验结果

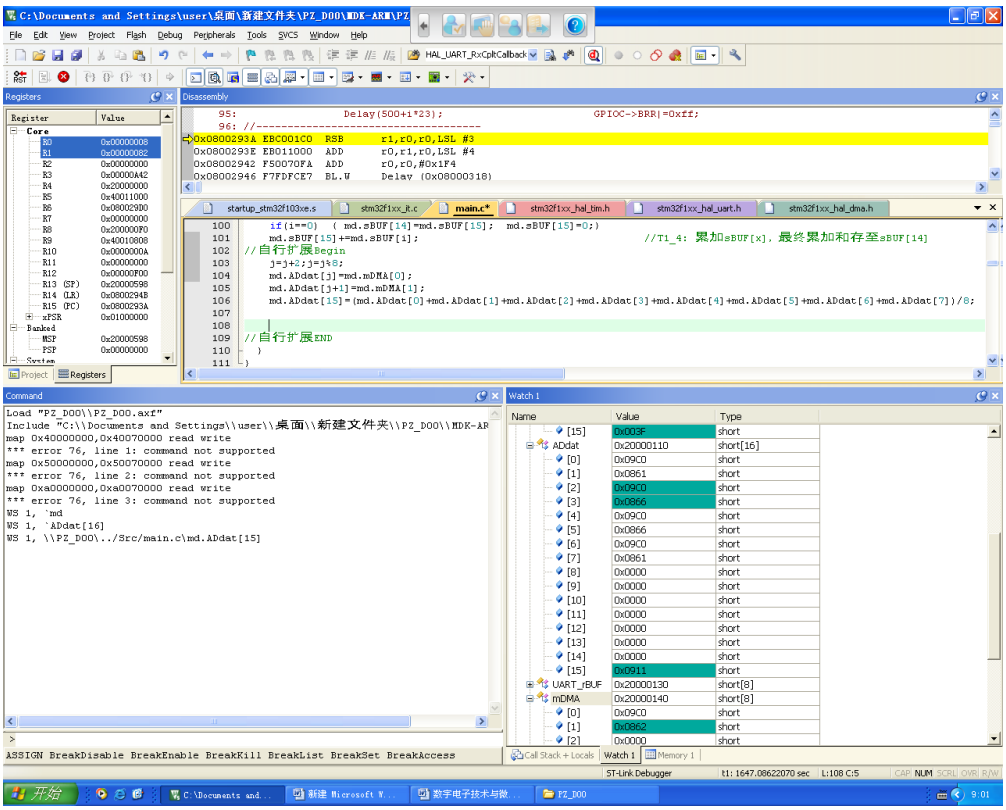

图 2: DMA 数据搬运调试