

STM32Cube高效开发教程（基础篇）

第5章 STM32CubeMonitor的使用

王维波

中国石油大学（华东）控制科学与工程学院

STM32Cube高效开发教程（基础篇）

作者：王维波，鄢志丹，王钊

人民邮电出版社

2021年9月出版

如果有读者需要本书课件的PPT版本用于备课，可以给作者发邮件免费获取，并可加入专门的教学和技术交流QQ群

邮箱：wangwb@upc.edu.cn



第5章 STM32CubeMonitor的使用

5.1 STM32CubeMonitor功能简介

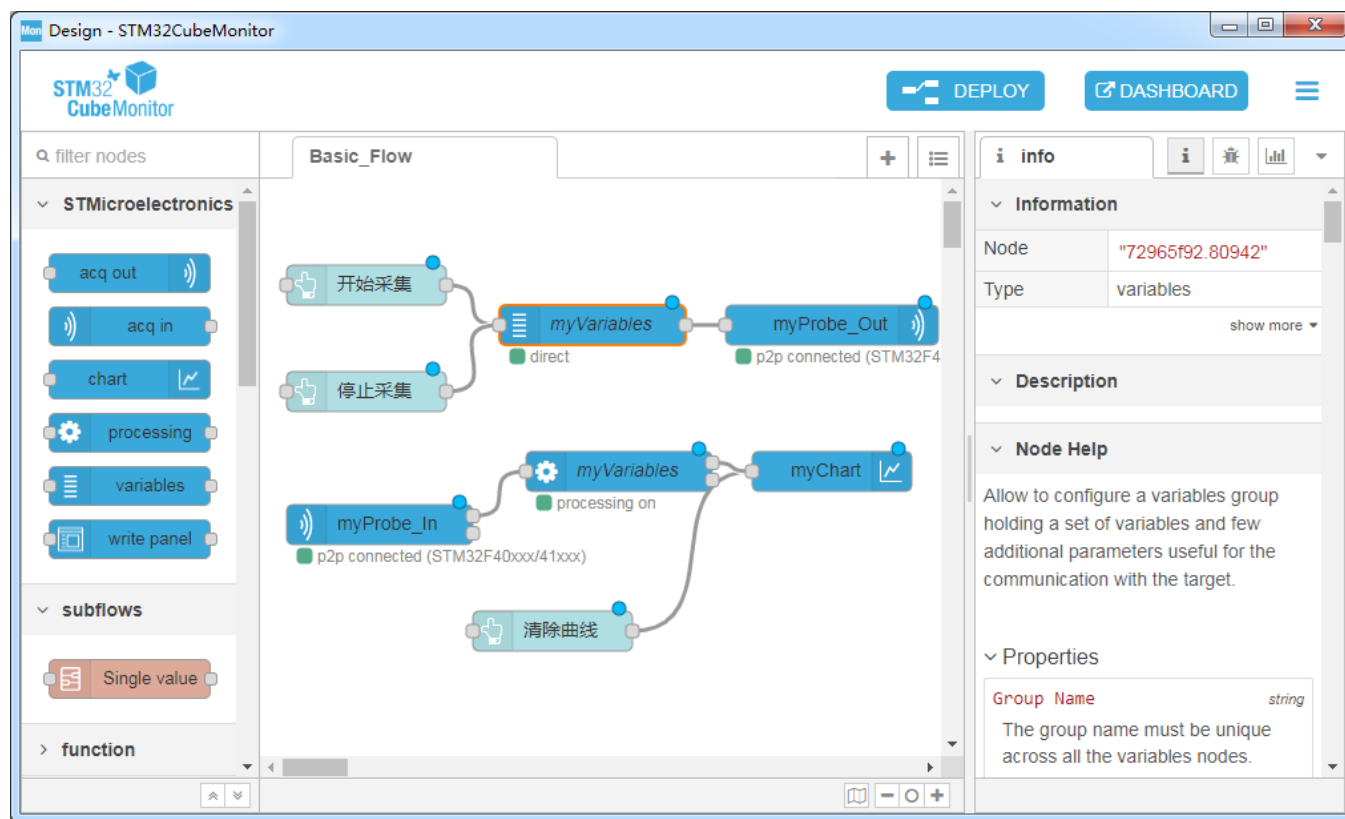
5.2 CubeMonitor基本操作

5.3 CubeMonitor基本功能使用示例

5.4 CubeMonitor的使用小结

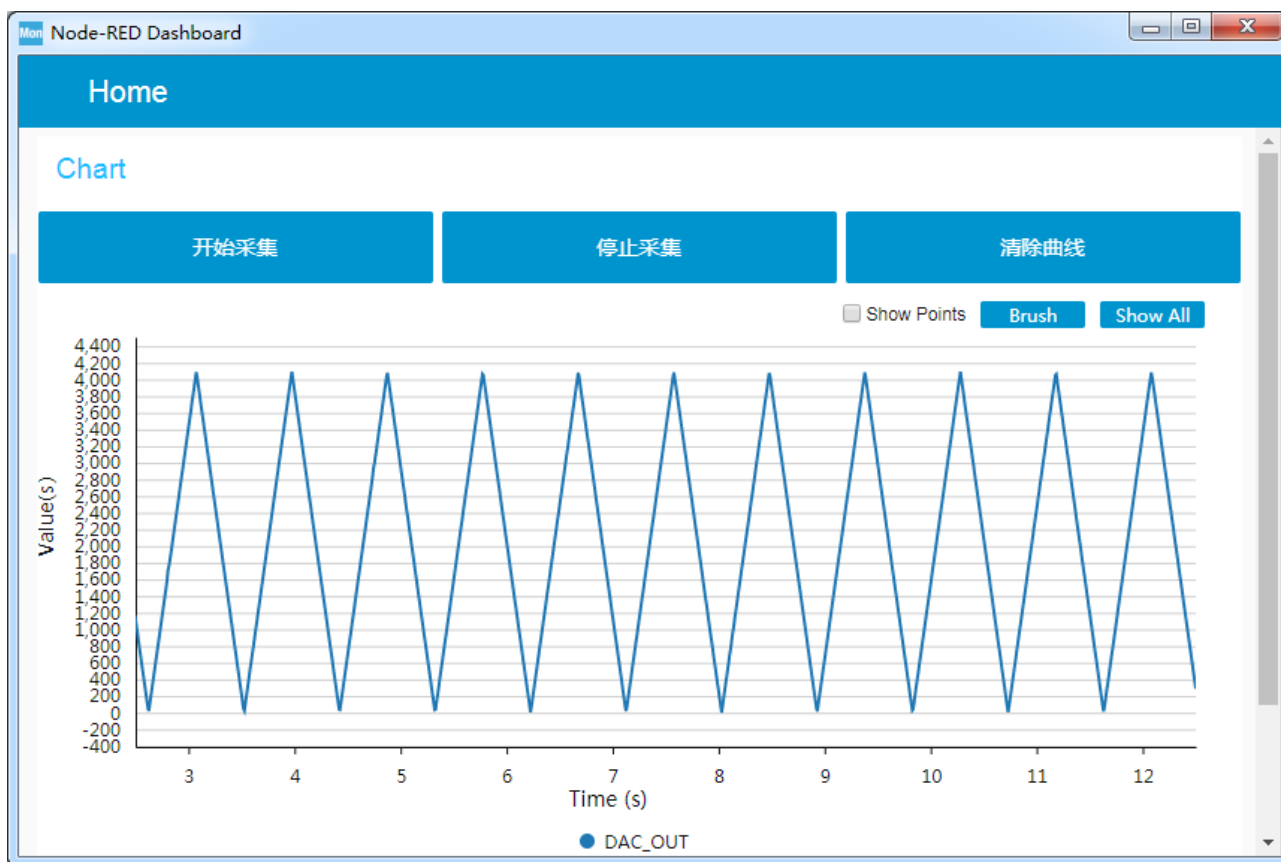
5.1 STM32CubeMonitor功能简介

CubeMonitor是基于流程（Flow）的图形化编程工具，类似于LabView，或MATLAB中的SimuLink



设计模式界面

CubeMonitor能使用图形化编程方式设计监测程序，通过ST-LINK仿真器连接STM32系统后，就可以实时监测和显示所监测的变量或外设寄存器的值。



数据监测的Dashboard窗口，监测DAC输出的三角波

CubeMonitor具有如下的一些功能和特性：

- 基于流程的图形化编辑器，无需编程。
- 目前只支持ST-LINK仿真器。
- 在STM32上的程序全速运行时，CubeMonitor可以即时读取或修改STM32内存中的变量或外设寄存器的值。
- 可以设置触发条件开始数据采集。
- 可以将监测的数据存储到文件，以便后期分析。
- 具有数据可视化显示组件，如曲线、gauge、柱状图等。
- 在同一个局域网内的其他电脑、手机或平板电脑上，通过浏览器就可以实现远程监测。
- 可以通过公用云平台和MQTT协议实现远程网络监测。

5.2 CubeMonitor基本操作

5.2.1 Node-RED中的一些基本概念

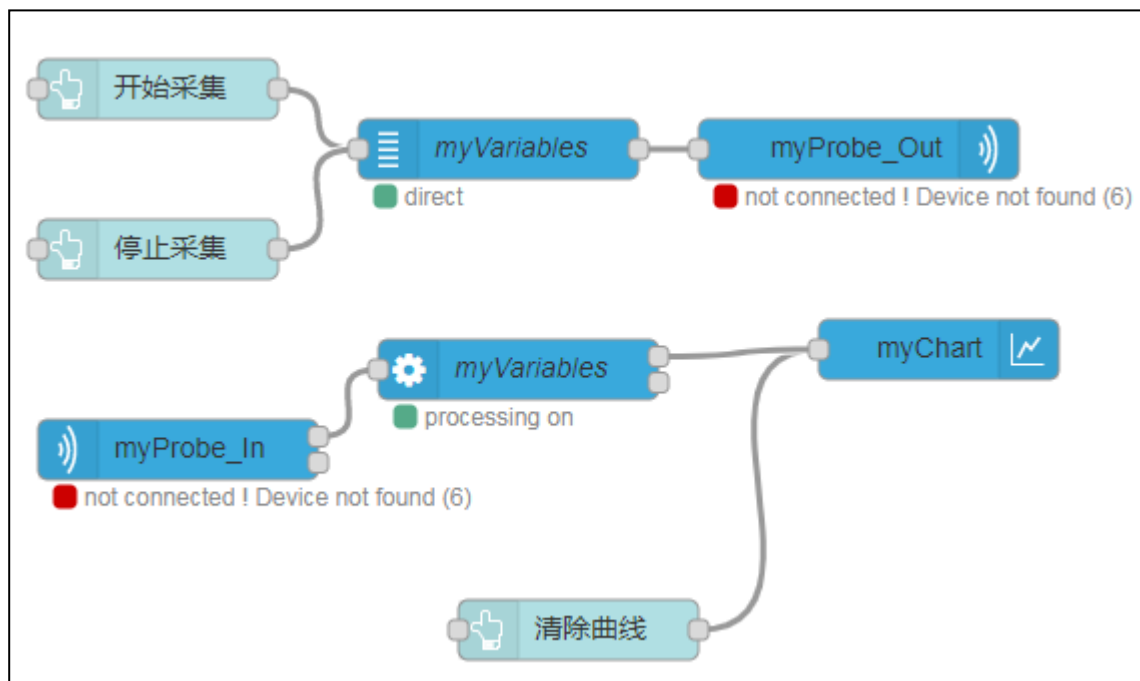
5.2.2 设计模式界面和基本操作

5.2.3 程序部署和Dashboard界面

5.2.1 Node-RED中的一些基本概念

操作软件解释以下概念

1. Node（节点）
2. Configuration node（配置节点）
3. Flow（流程）
4. Context（上下文）
5. Message（消息）
6. Subflow（子流程）
7. Wire（连线）



5.2.2 设计模式界面和基本操作

1. 设计模式界面组成

节点面板

工作区

主工具栏

侧边栏

The screenshot displays the STM32CubeMonitor Design interface. The main workspace contains a flowchart titled 'Basic_Flow' with nodes including 'START Acquisition', 'STOP Acquisition', 'myVariables', 'myProbe_In', 'myProbe_Out', 'myChart', and 'Clear Graphs'. The left sidebar, labeled '节点面板' (Node Panel), lists various node types under 'STMicroelectronics' and 'common'. The top toolbar, labeled '主工具栏' (Main Toolbar), includes 'DEPLOY' and 'DASHBOARD' buttons. The right sidebar, labeled '侧边栏' (Sidebar), shows the 'info' panel for the selected 'myProbe_Out' node, displaying its ID, name, and type, along with a description and node help.

Information	
Node	"731c3f91.24b62"
Name	myProbe_Out
Type	acquisition out

show more

Description

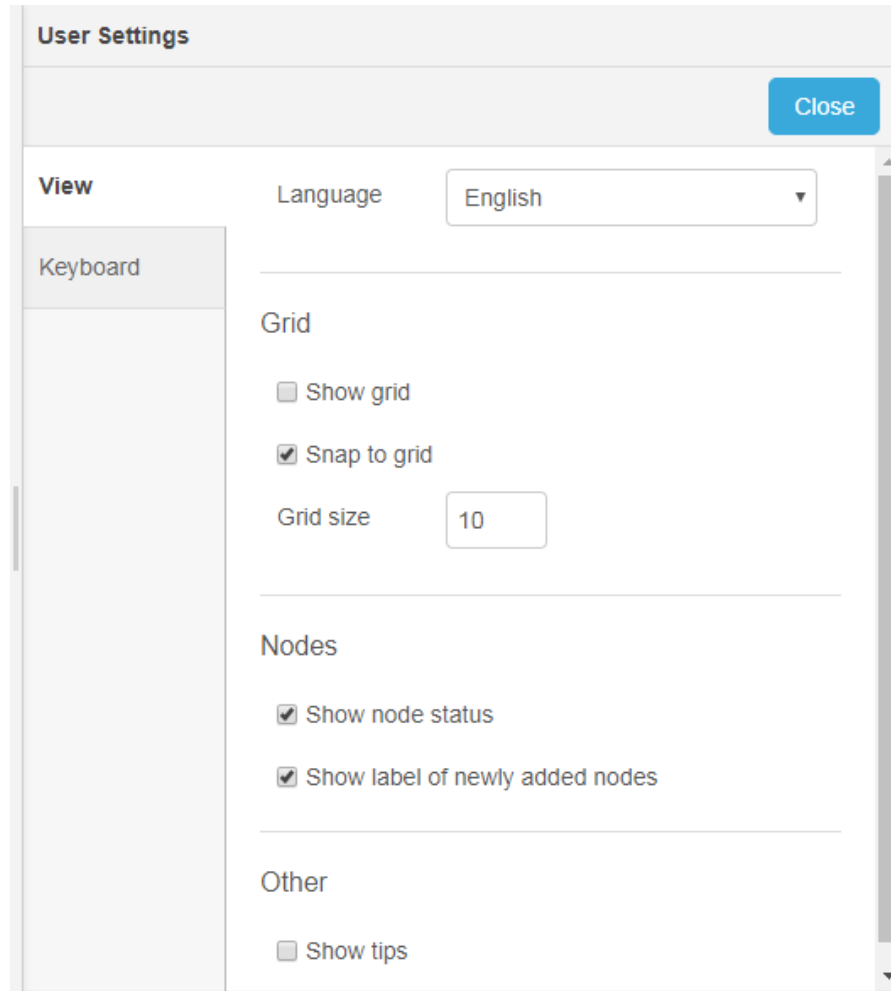
Node Help

Allow to define and/or select a probe configuration (probe name, protocol and frequency), open and close the connection and

ctrl click in the workspace to open the quick-add dialog

2. 软件设置

点击主菜单中的“Settings”项，可以打开软件设置的对话框



3. 各种节点

节点面板分组管理所有可以用于流程设计的节点。STMicroelectronics分组是ST公司为实现CubeMonitor的功能设计的一些专用节点



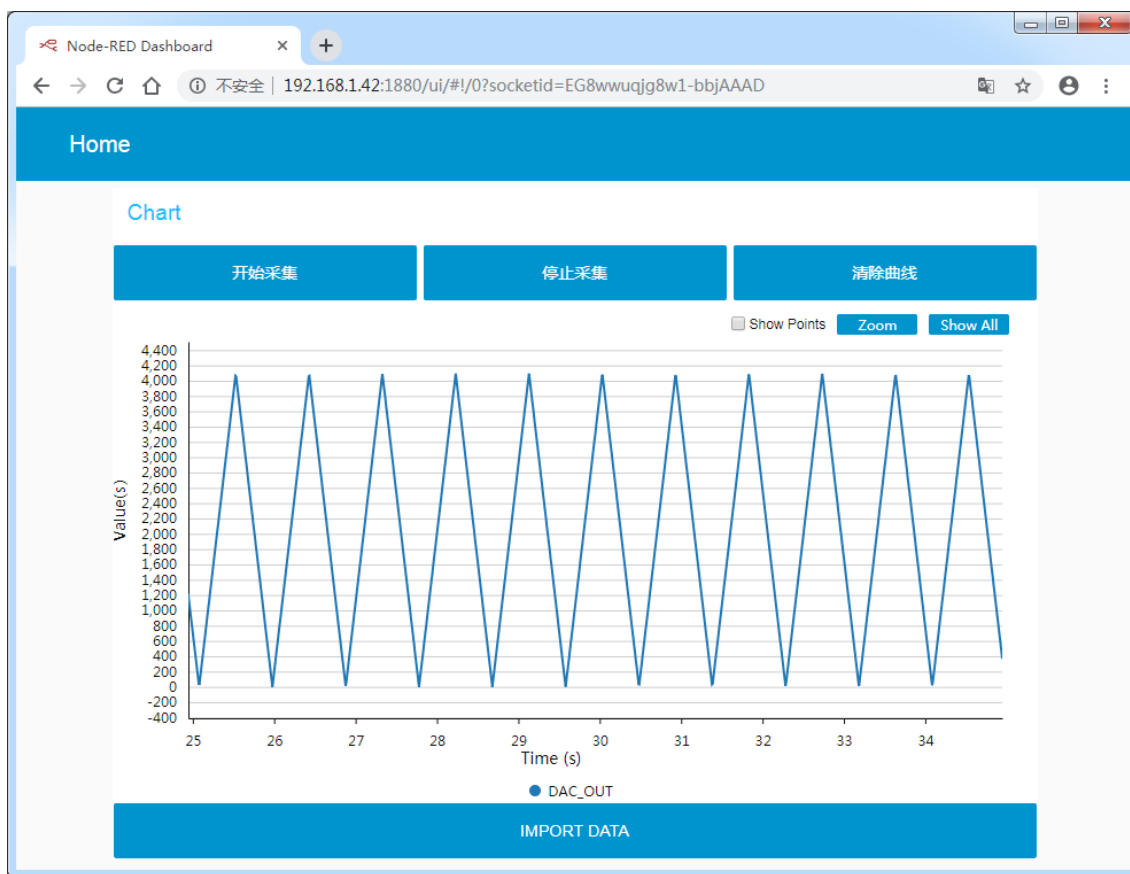
5.2.3 程序部署和Dashboard界面

1. 程序部署

在设计完流程图后，需要部署后才可以进行变量监测和显示。

2. 远程访问

可以通过局域网内的其他电脑、手机或平板电脑的浏览器访问CubeMonitor，只需要主机的IP地址和1880端口



5.3 CubeMonitor基本功能使用示例

5.3.1 STM32 MCU项目

5.3.2 变量监测的基本操作

5.3.3 监测外设寄存器的值

5.3.4 监测变量的数值显示

5.3.5 修改变量的值

5.3.1 STM32 MCU项目

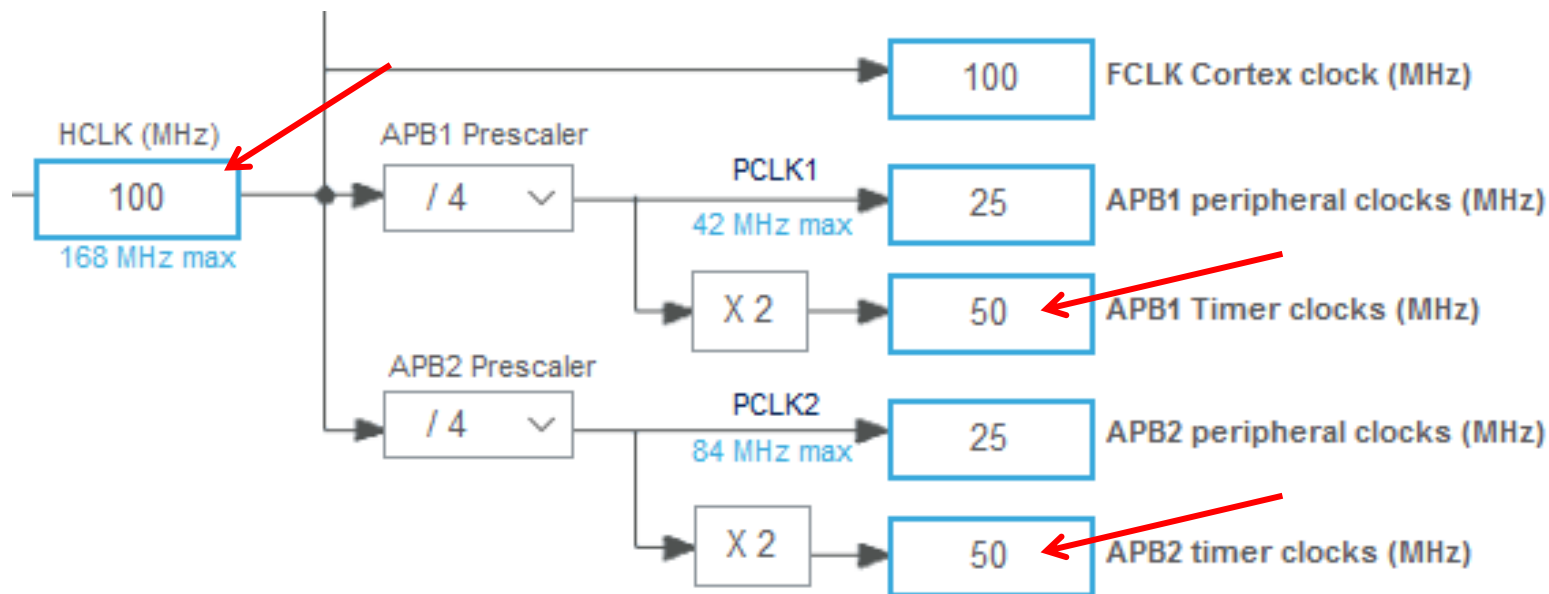
本节示例的STM32项目Demo5_1ADC从第14章的项目Demo14_2TimTrigger复制而来，用定时器TIM3触发ADC1进行数据采集。

在示例Demo14_2TimTrigger中，只能在LCD上显示当前采样点的值，而在本示例中，可以通过CubeMonitor监测ADC1采集数据的曲线，而且可以改变TIM3的定时周期，从而改变ADC1的采样率。

如果对ADC、DAC、定时器等不熟悉也应该是能看懂CubeMonitor的功能和基本操作的，但要完全理解其中一些设置的原理，还是需要学习过后面的一些章节后再来看本章。

1. TIM3的设置

在时钟树上，设置HCLK为100MHz，APB1和APB2的定时器时钟信号频率都设置为50MHz



TIM3的设置

设置TIM3的定时周期
就是100ms。

TRGO信号源设置为
Update Event，就是用
UEV事件信号作为TRGO
信号

TIM3 Mode and Configuration	
Mode	
Slave Mode	Disable
Trigger Source	Disable
Clock Source	Internal Clock
Channel1	Disable
Channel2	Disable
Configuration	
Reset Configuration	
User Constants	
NVIC Settings	
DMA Settings	
Parameter Settings	
Search (Ctrl+F)	
Counter Settings	
Prescaler (PSC - 16 bits value)	49999
Counter Mode	Up
Counter Period (AutoReload Register - 16 bits value)	100
Internal Clock Division (CKD)	No Division
auto-reload preload	Disable
Trigger Output (TRGO) Parameters	
Master/Slave Mode (MSM bit)	Disable (Trigger)
Trigger Event Selection	Update Event

2. ADC1的设置

模式设置中使用通道IN5。12位精度，右对齐格式，参数 External Trigger Conversion Source（外部触发转换源）选择为Timer 3 Trigger Out event。开启ADC1的全局中断

✓ Parameter Settings	✓ User Constants	✓ NVIC Settings	✓ DMA Settings	✓ GPIO Settings
✓ ADCs_Common_Settings				
Mode	Independent mode			
✓ ADC_Settings				
Clock Prescaler	PCLK2 divided by 2			
Resolution	12 bits (15 ADC Clock cycles)			
Data Alignment	Right alignment			
Scan Conversion Mode	Disabled			
Continuous Conversion Mode	Disabled			
Discontinuous Conversion Mode	Disabled			
DMA Continuous Requests	Disabled			
End Of Conversion Selection	EOC flag at the end of single channel conversion			
✓ ADC_Regular_ConversionMode				
Number Of Conversion	1			
External Trigger Conversion Source	Timer 3 Trigger Out event			
External Trigger Conversion Edge	Trigger detection on the rising edge			
✓ Rank				
Channel	Channel 5			
Sampling Time	28 Cycles			

3. 程序代码

完整程序看示例源代码

在ADC1的中断里读取转换结果数据，并在LCD上显示

```
/* ADC的转换完成EOC事件中断回调函数 */
void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc)
{
    if (hadc->Instance == ADC1)
    {
        ADCValue=HAL_ADC_GetValue(hadc);        //读取转换结果
        LCD_ShowUintX(orgX,orgY,ADCValue, 5);

        ADCVoltage=3300*ADCValue;                //mV
        ADCVoltage = ADCVoltage >>12;            //除以2^12
        LCD_ShowUintX(voltX,voltY, ADCVoltage, 4);
    }
}
```

5.3 CubeMonitor基本功能使用示例

5.3.1 STM32 MCU项目

5.3.2 变量监测的基本操作

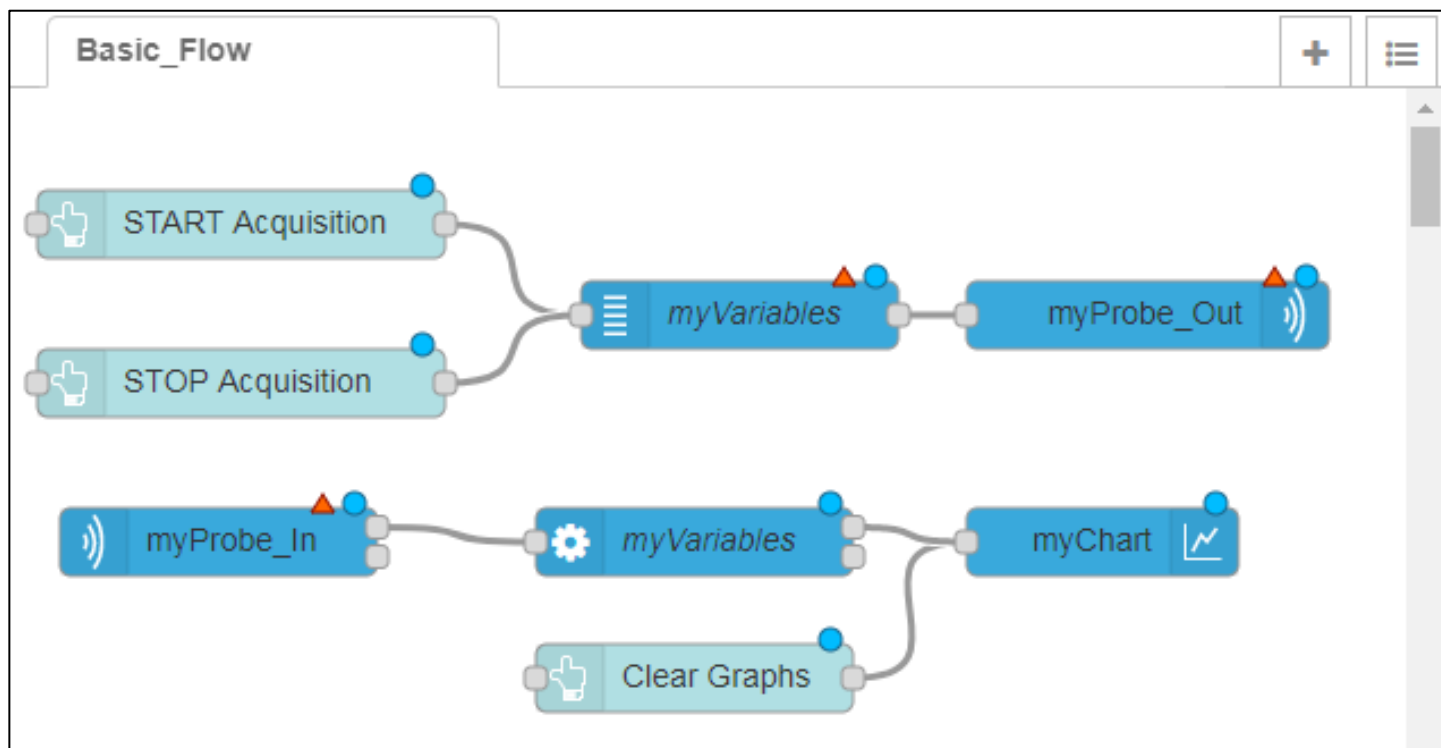
5.3.3 监测外设寄存器的值

5.3.4 监测变量的数值显示

5.3.5 修改变量的值

5.3.2 变量监测的基本操作

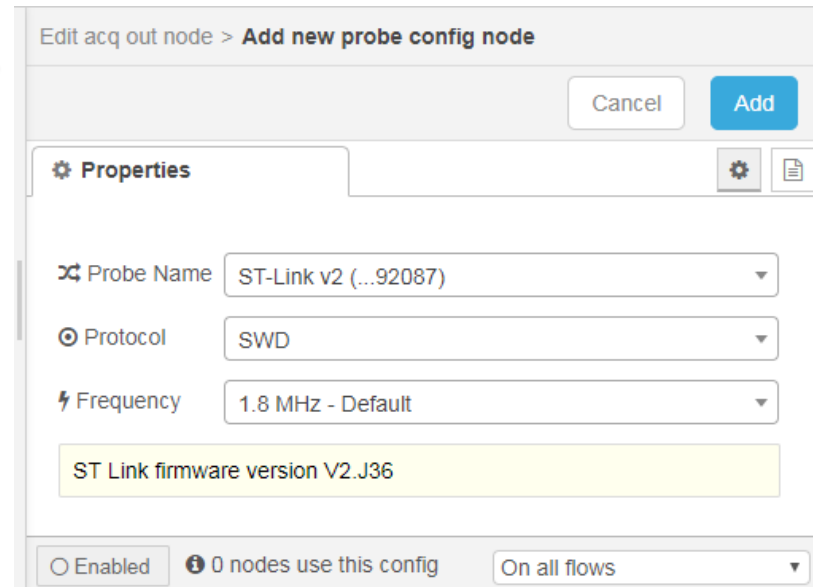
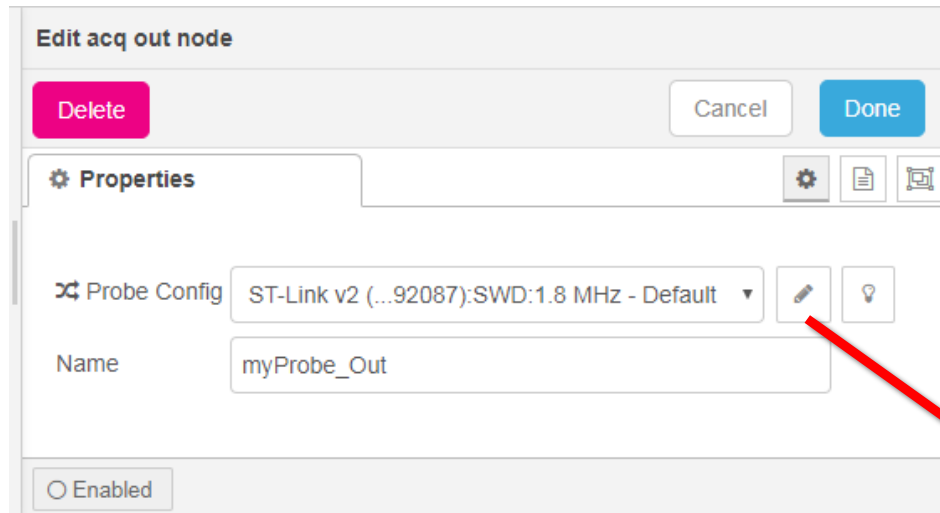
1. Basic_Flow流程图



打开软件操作讲解

2. acq out和acq in节点配置

设置acq out节点，配置所连接的ST-LINK仿真器。



打开软件操作讲解

3. variables节点配置

variables节点用于定义一个变量组。从STM32可执行文件，即编译后的elf文件中选择需要监测的变量

打开软件操作讲解

Edit variables node

Delete Cancel Done

Properties

Group Name myVariables

Access point 0

Executable Demo5_1EXE (Demo5_1ADC.elf)

Variable list

Name	Start Address	Type
ADCValue	0x20000158	Unsigned 32-bit
ADCVoltage	0x2000014c	Unsigned 32-bit

+ Add custom variable - Remove all custom variables

Acquisition parameters

Sampling frequency sequential loop

Acquisition mode direct

Trigger start mode off

Trigger name ADCVoltage

Trigger threshold 30000

Enabled

Edit variables node > Add new exe-config config node

Cancel Add

Properties

Name Demo5_1EXE

Folder D:\CubeDemo\Part1_Environment\Chap05CubeMonitor\Demo5_1ADC\Debug

File Demo5_1ADC.elf

Expand Variable List

Variable List

Select	Name	Start Address	Type
<input checked="" type="checkbox"/>	ADCValue	0x20000158	Unsigned 32-bit
<input checked="" type="checkbox"/>	ADCVoltage	0x2000014c	Unsigned 32-bit
<input type="checkbox"/>	AHBPrescTable[0]	0x08003ebc	Unsigned 8-bit
<input type="checkbox"/>	FSMC_Initialized	0x20000094	Unsigned 32-bit
<input type="checkbox"/>	hadc1.DMA_Handle	0x200000e8	Unsigned 32-bit
<input type="checkbox"/>	hadc1.ErrorCode	0x200000f4	Unsigned 32-bit
<input type="checkbox"/>	hadc1.Init.ClockPrescaler	0x200000b4	Unsigned 32-bit

Select All Deselect All Filter on variable name

Enabled 0 nodes use this config On all flows

4. processing节点配置

processing节点的输入是acq in节点的输出，也就是ST-LINK仿真器采集到的数据。processing节点每50ms输出一次，可能包含一次采样或多次采样的数据。

processing节点的输出作为chart节点的输入，也就是在曲线上显示。

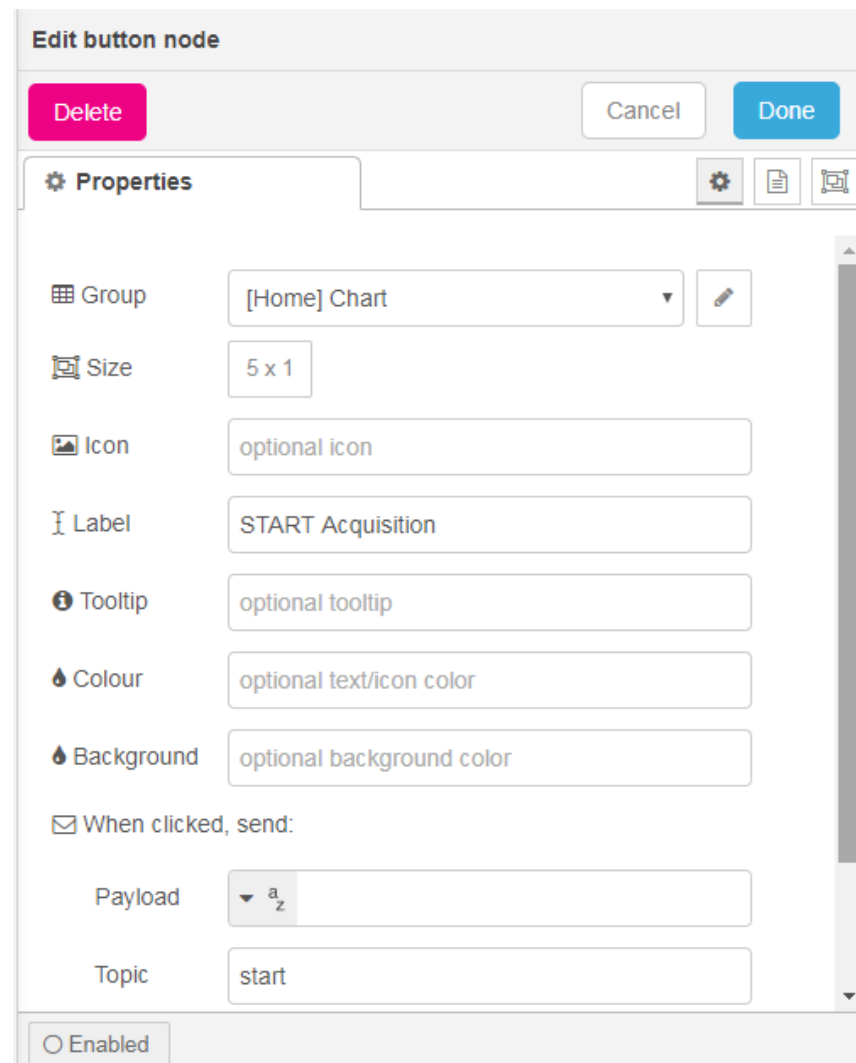
The screenshot shows the 'Edit processing node' dialog box. It has a title bar 'Edit processing node' and three buttons: 'Delete' (pink), 'Cancel' (white), and 'Done' (blue). Below the buttons is a 'Properties' section with a gear icon and three sub-sections: 'Group Name' with a dropdown menu showing 'myVariables', 'Log option' with a dropdown menu showing 'No log', and 'Outputs' with a list of two items: 'ADCValue' and 'ADCVoltage'. Each item has a right arrow and a close button. Below the 'Outputs' section is a 'Post-processing' section with a dropdown menu showing 'Expression' and 'Statistic'. At the bottom, there is a checkbox labeled 'Enabled'.

打开软件操作讲解

5. button节点配置

当按钮被点击后会发出一个msg，有msg.payload和msg.topic两个属性。按钮的payload自动设置为按钮节点的ID，按钮的msg.topic表示了按钮的作用类型，流程图中3个按钮的msg.topic分别是start、stop和clear。

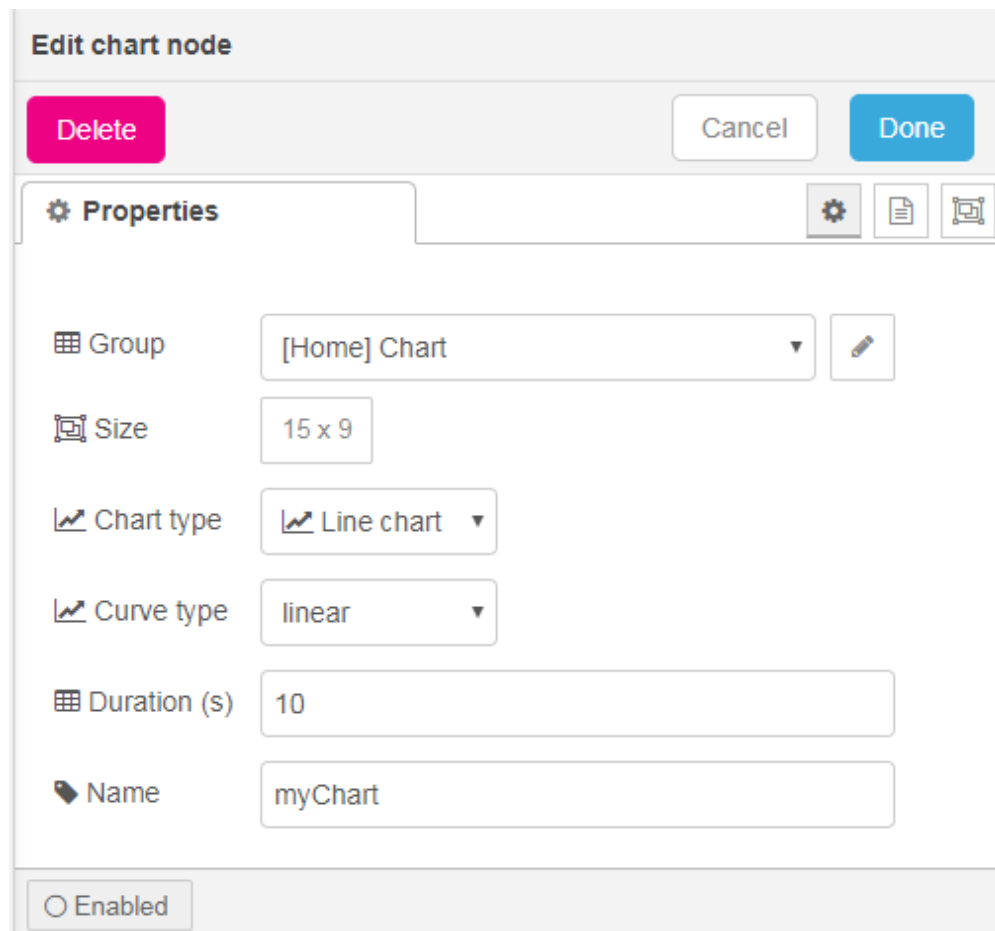
打开软件操作讲解



The screenshot shows the 'Edit button node' configuration window. It has a title bar 'Edit button node' and three buttons: 'Delete' (red), 'Cancel' (grey), and 'Done' (blue). Below the buttons is a 'Properties' section with a gear icon and three sub-panels: 'Group', 'Size', and 'Icon'. The 'Group' panel shows a dropdown menu with '[Home] Chart' selected. The 'Size' panel shows a text input with '5 x 1'. The 'Icon' panel shows a text input with 'optional icon'. Below these are several other properties: 'Label' (text input with 'START Acquisition'), 'Tooltip' (text input with 'optional tooltip'), 'Colour' (text input with 'optional text/icon color'), and 'Background' (text input with 'optional background color'). At the bottom, there is a checkbox 'When clicked, send:' which is checked. Below this checkbox are two text inputs: 'Payload' (with a dropdown menu showing 'a_z') and 'Topic' (with 'start'). At the very bottom, there is a radio button 'Enabled' which is selected.

6. chart节点配置

chart节点的输入是processing节点的输出，就是用图表显示监测的数据。



The screenshot shows the 'Edit chart node' configuration window. At the top, there are three buttons: 'Delete' (pink), 'Cancel' (white), and 'Done' (blue). Below these is a 'Properties' tab with a gear icon. The main area contains several configuration options:

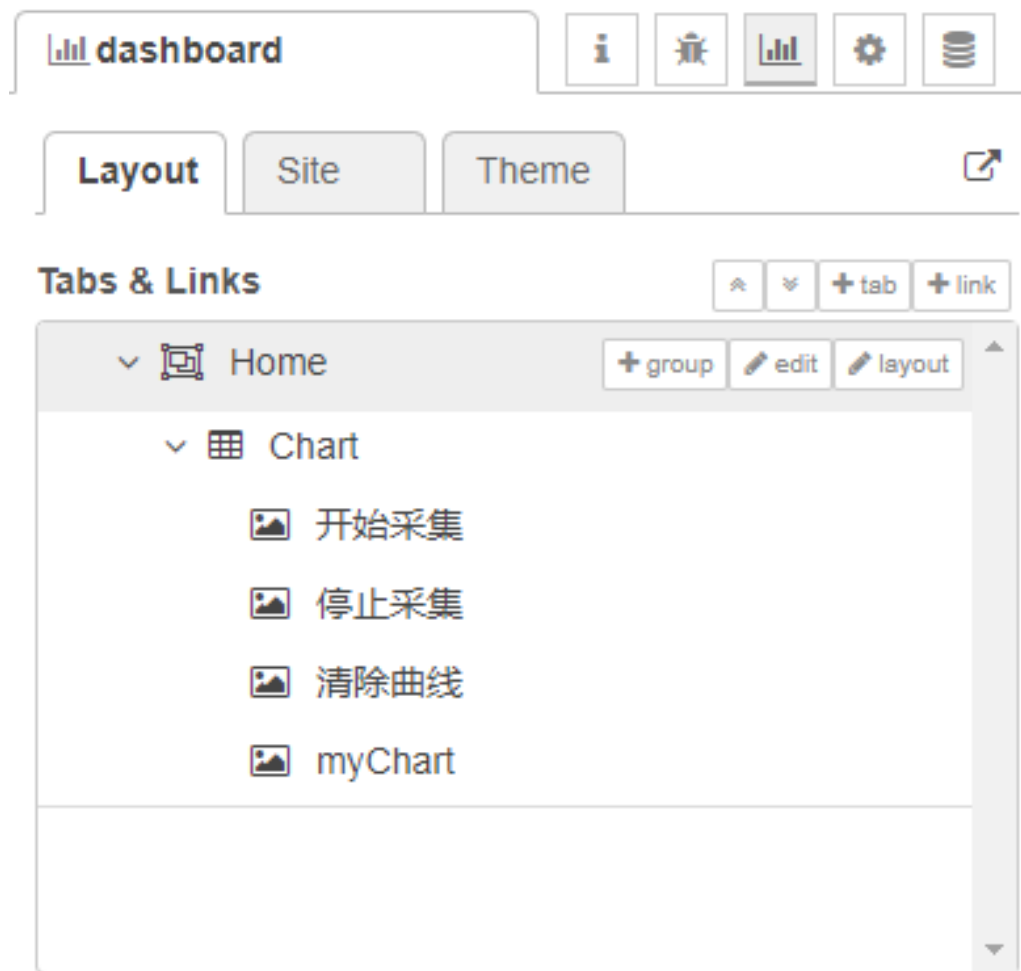
- Group:** A dropdown menu showing '[Home] Chart' with a pencil icon to its right.
- Size:** A text input field containing '15 x 9'.
- Chart type:** A dropdown menu showing 'Line chart' with a line graph icon.
- Curve type:** A dropdown menu showing 'linear'.
- Duration (s):** A text input field containing '10'.
- Name:** A text input field containing 'myChart'.

At the bottom of the window, there is a checkbox labeled 'Enabled' which is currently unchecked.

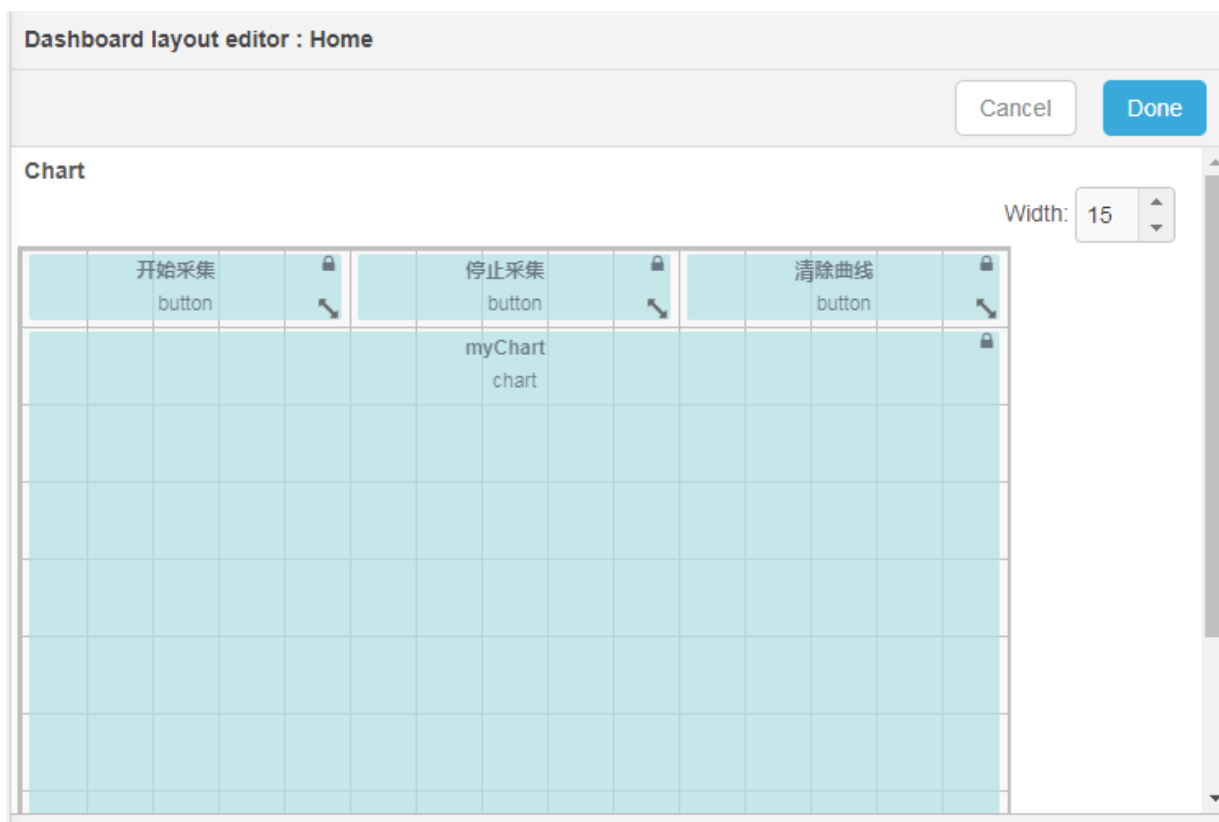
打开软件操作讲解

7. Dashboard设计

Dashboard就是显示监测结果的UI界面，可以在设计模式下设置Dashboard界面。在侧边栏的“Dashboard”页面显示了Dashboard界面上的节点组成，如图所示。



下图是本示例中3个按钮和1个图表的界面布局，它们在一个Chart界面分组里。注意到界面上有网格，Dashboard上界面元素的大小就是用网格个数定义的。



打开软件操作讲解

8. 部署和运行

点击主工具栏上的“DEPLOY”按钮进行部署，部署就是要保存当前的设计并发布给CubeMonitor的运行环境。

完成部署后再点击主窗口工具栏上的“DASHBOARD”按钮就可以进入Dashboard模式，显示监测界面

打开软件操作讲解

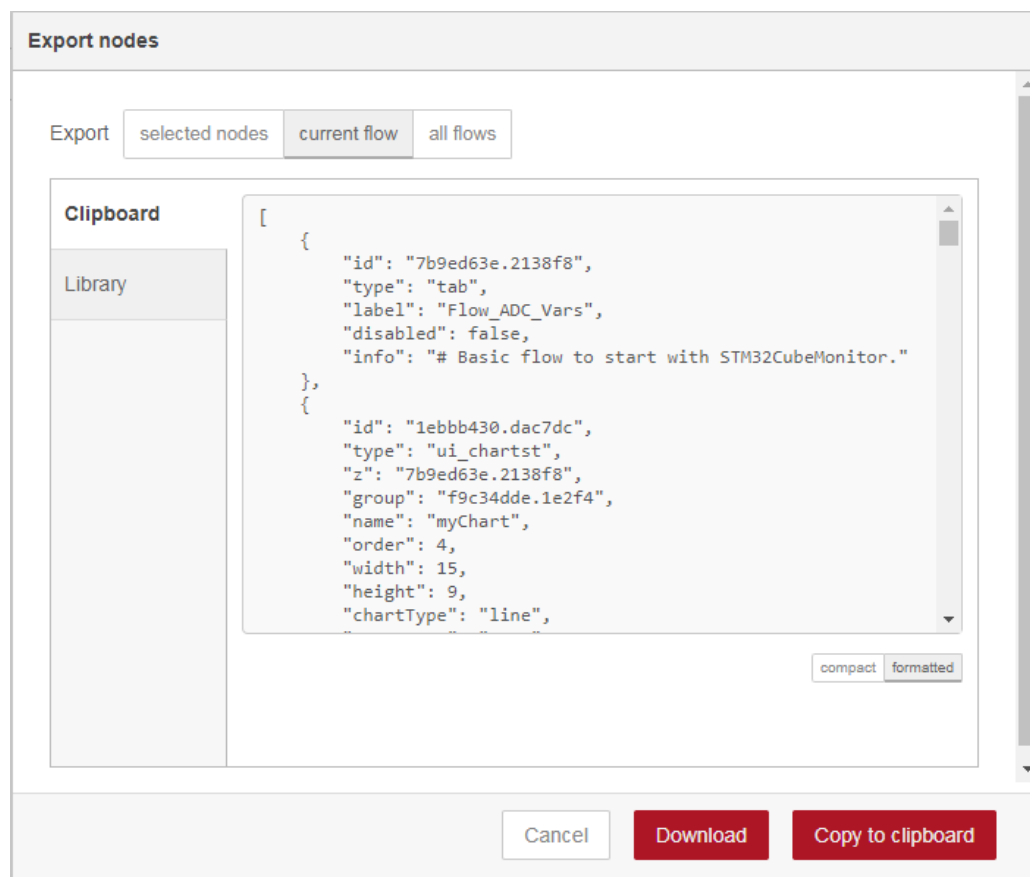


9. 保存流程

设计好一个流程图，并运行测试无误后可以将当前流程图导出为一个文件。点击主菜单中的“Export”菜单项。

节点和流程是用JSON格式定义的，点击“Download”按钮就可以将当前页面上的定义保存为一个JSON文件。

打开软件操作讲解



5.3 CubeMonitor基本功能使用示例

5.3.1 STM32 MCU项目

5.3.2 变量监测的基本操作

5.3.3 监测外设寄存器的值

5.3.4 监测变量的数值显示

5.3.5 修改变量的值

5.3.3 监测外设寄存器的值

前一示例中对STM32 MCU中的全局变量进行监测，为此将ADCValue和ADCVoltage定义为全局变量。

必须注意：MCU程序如果修改后重新编译，全局变量的地址可能会改变，需要重新配置变量组。

还可以对外设寄存器进行监测，因为外设寄存器相当于就是有固定地址的全局变量。

1. variables节点配置

配置variables节点，添加对寄存器ADC_DR的监测

Edit variables node

Delete Cancel Done

Properties

Group Name: myVariables

Access point: 0

Executable: Demo5_1EXE (Demo5_1ADC.elf)

Variable list

Name	Start Address	Type
ADC_Reg	0x4001204C	Unsigned 32-bit

+ Add custom variable - Remove all custom variables

> Acquisition parameters

Enabled

外设初始化代码中的Instance指向外设ADC1的基地址，MCU的参考手册中有寄存器ADC_DR内部偏移地址，从而可确定ADC_DR的全局地址

```
27 ADC_HandleTypeDef hadc1;
28
29 /* ADC1 init function */
30 void MX_ADC1_Init(void)
31 {
32     ADC_ChannelConfTypeDef sConfig = {0};
33
34     /** Configure the global features of the ADC (Clock, Resolution, Data Alignment)
35     */
36     hadc1.Instance = ADC1;
37     hadc1.Init.ClockPrescaler = 0;
38     hadc1.Init.Resolution = ((ADC_TypeDef *) ((0x40000000UL + 0x00010000UL) + 0x2000UL));
39     hadc1.Init.ScanConvMode = 0;
40     hadc1.Init.ContinuousConvMode = DISABLE;
41     hadc1.Init.DiscontinuousConvMode = DISABLE;
42     hadc1.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_RISING;
43     hadc1.Init.ExternalTrigConv = ADC_EXTERNALTRIGCONV_T3_TRGO;
```

打开软件操作讲解

2. processing节点配置

在processing节点配置中，通过对ADC_Reg的计算得到变量ADC_mV，即电压的毫伏值

The screenshot shows the 'Edit processing node' dialog box. It has a title bar 'Edit processing node' and three buttons: 'Delete' (pink), 'Cancel' (light blue), and 'Done' (blue). Below the buttons is a 'Properties' tab with a gear icon. The 'Expression' tab is selected, showing an 'Expression name' field with 'ADC_mV' and a 'Formula' field with '(3300*ADC_Reg)>>12|'. There are '+ add' and 'clear' buttons below the formula field. A list of 'Possible variables in formula (click to insert)' shows 'ADC_Reg'. On the right, the 'x2 Operations' dropdown is set to 'Mathematical', showing a list of operators: '+', '-', '*', '/', '%', 'mod', '^', and '!'.

Edit processing node

Delete Cancel Done

Properties

Expression Statistic

Expression name

ADC_mV

Formula

(3300*ADC_Reg)>>12|

+ add clear

Possible variables in formula (click to insert)

ADC_Reg

x2 Operations

Mathematical

+
-
*
/
%
mod
^
!

Enabled

打开软件操作讲解

3. 部署和运行

对Dashboard界面不做任何修改，完成设置后部署流程图，然后启动Dashboard模式。数据监测的界面与图5-24相似，只是本示例显示的是变量ADC_Reg和ADC_mV的数据曲线。

5.3 CubeMonitor基本功能使用示例

5.3.1 STM32 MCU项目

5.3.2 变量监测的基本操作

5.3.3 监测外设寄存器的值

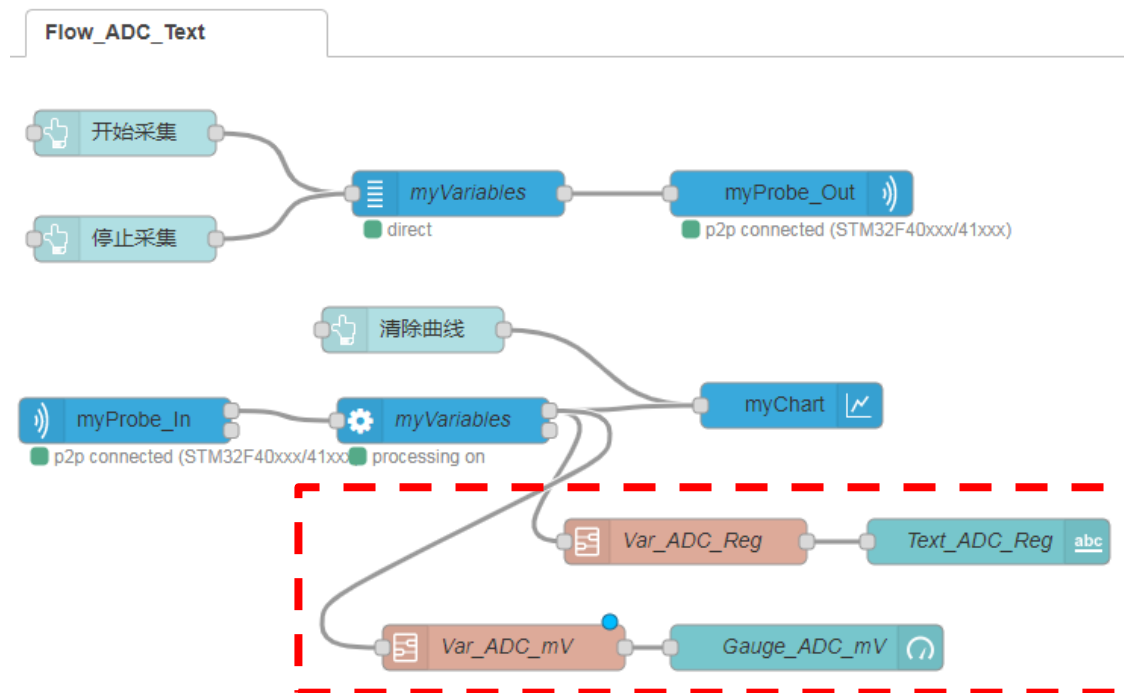
5.3.4 监测变量的数值显示

5.3.5 修改变量的值

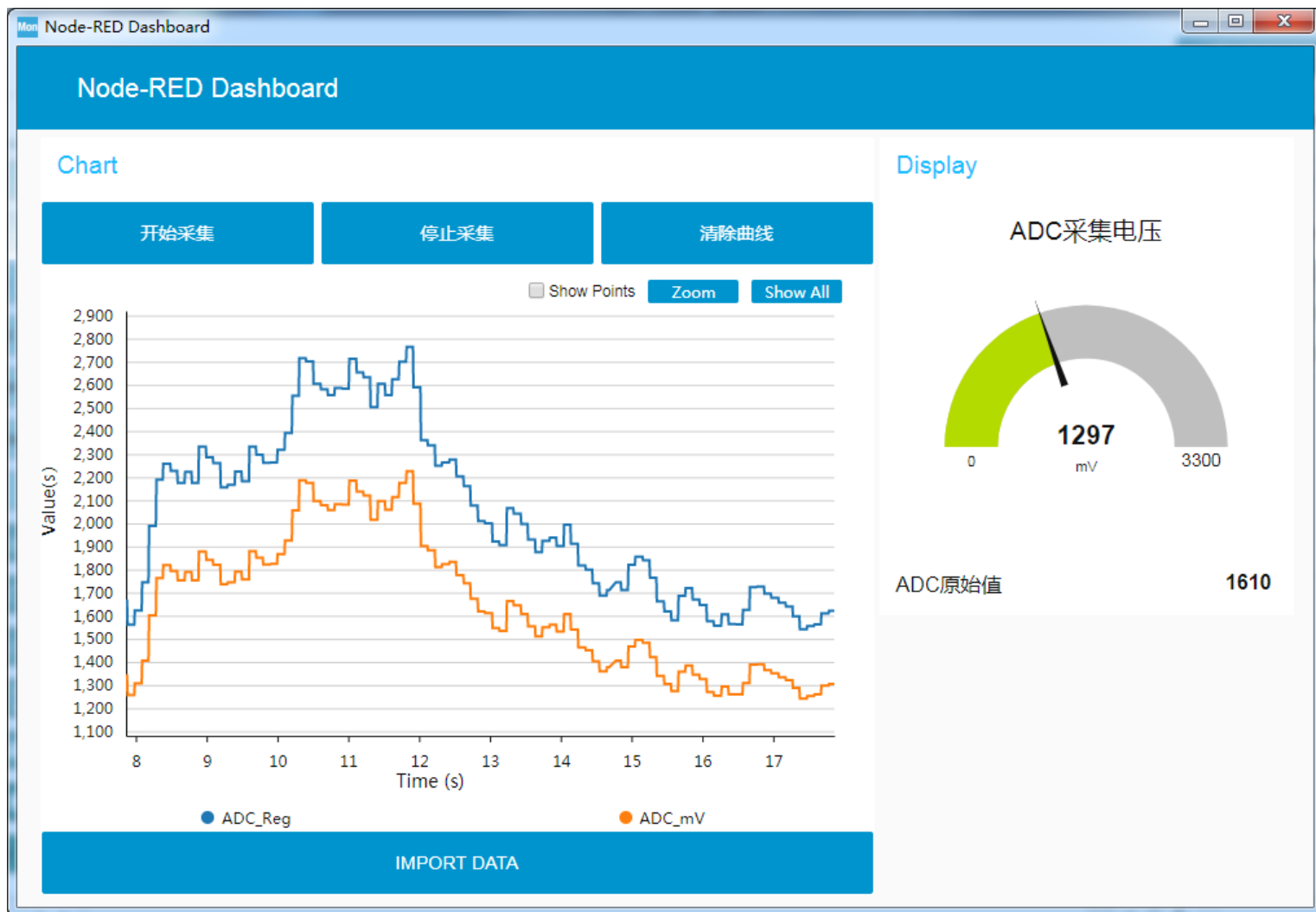
5.3.4 监测变量的数值显示

1. 示例功能和运行效果

在前一示例流程图基础上添加单个变量的显示功能，本节完成的流程图如图所示，添加了虚线框中的几个节点。



打开软件操作讲解



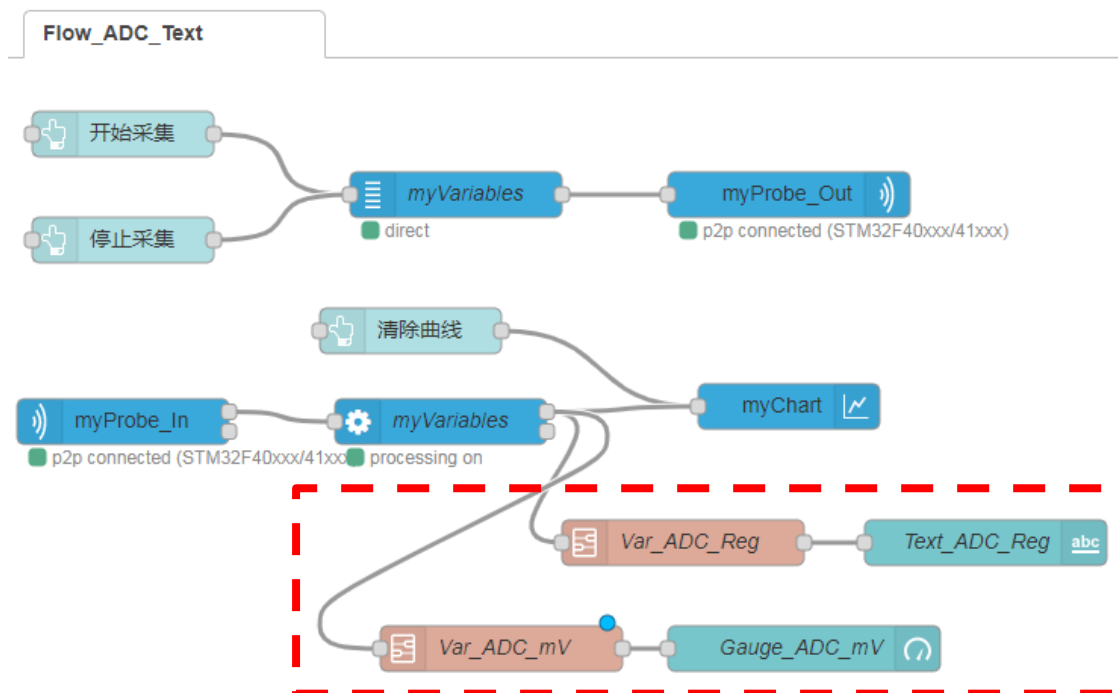
本节示例的Dashboard模式界面

打开软件操作讲解

2. 流程图设计

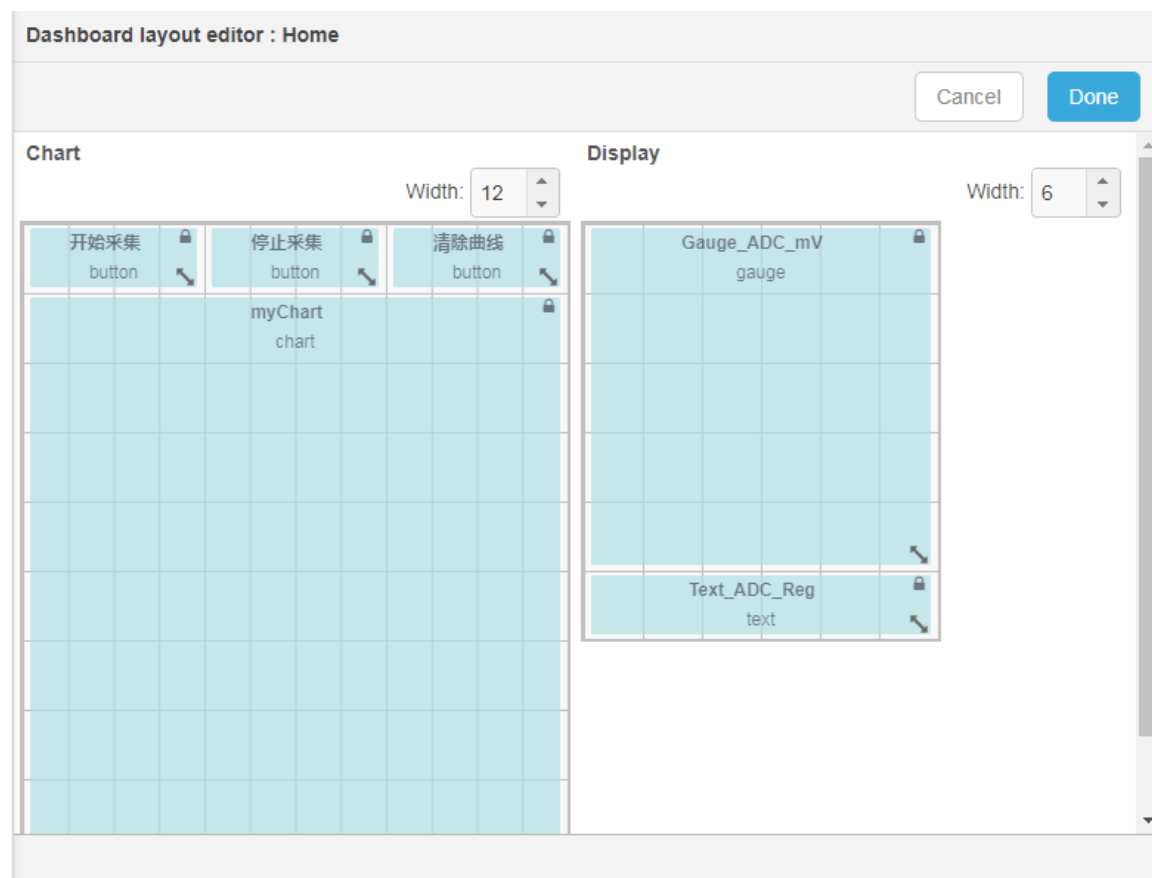
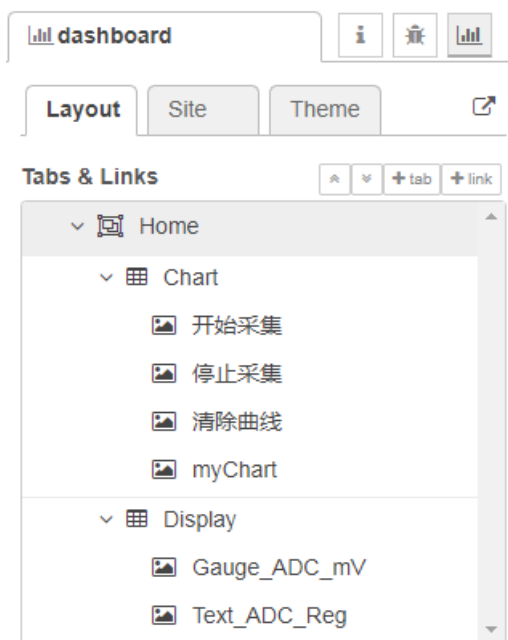
流程图中新增的部分是图5-31中虚线框中的几个节点，包括2个single value节点，1个text节点和1个gauge节点。

详细设计过程见教材，打开软件操作讲解



3. Dashboard布局设计

在Home页面上新增了一个名称为“Display”的分组，这个界面分组的宽度设置为6个网格。将text节点和gauge节点都放在Display分组里。



5.3 CubeMonitor基本功能使用示例

5.3.1 STM32 MCU项目

5.3.2 变量监测的基本操作

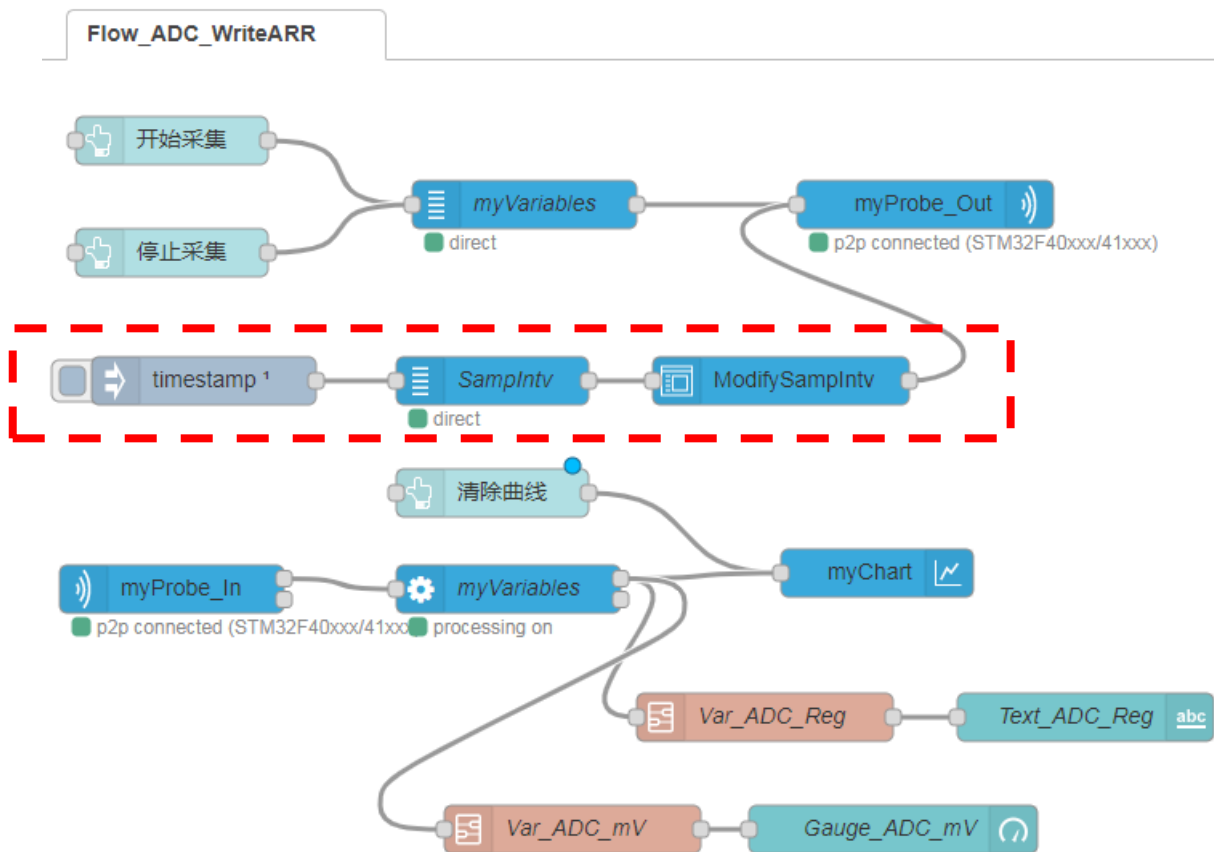
5.3.3 监测外设寄存器的值

5.3.4 监测变量的数值显示

5.3.5 修改变量的值

1. 示例功能和运行效果

使用CubeMonitor还可以修改变量的值。在前一示例的基础上修改流程图，在Dashboard模式下可以修改TIM3_ARR寄存器的值，也就是修改ADC1的采样频率。



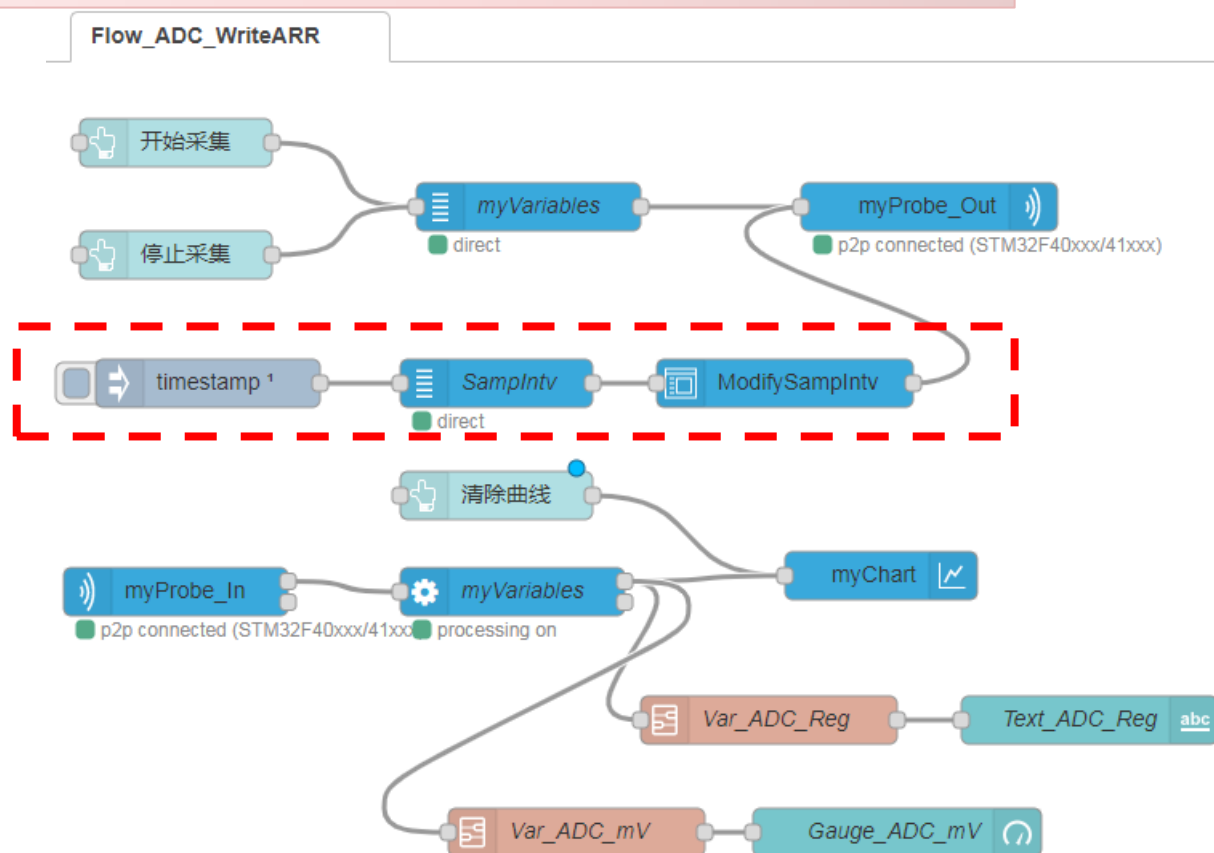
本示例Dashboard模式运行时界面，可修改采样周期



2. 流程图设计

流程图中新增的部分是图中虚线框中的几个节点，包括1个inject节点，1个variables节点和1个write panel节点。

详细设计过程见教材，打开软件操作讲解



第5章 STM32CubeMonitor的使用

5.1 STM32CubeMonitor功能简介

5.2 CubeMonitor基本操作

5.3 CubeMonitor基本功能使用示例

5.4 CubeMonitor的使用小结

CubeMonitor有两种采样模式，**Direct**模式和**Snapshot**模式。本章的示例使用的都是Direct模式，即非侵入模式，不需要对MCU程序做任何修改。Snapshot模式可以实现比较精确的采样频率，但是需要在MCU程序中嵌入代码，是一种侵入式监测模式。

CubeMonitor的采样率是比较低的，官方资料说单变量监测时可以达到1000Hz，但是实际中基本达不到。所以，使用CubeMonitor不能监测信号频率比较高的变量。

参考资料

- Node-RED官网, <https://nodered.org/docs/>
- STM32CubeMonitor介绍,
[https://www.stmcu.com.cn/ecosystem/Cube/STM32Cube
Monitor](https://www.stmcu.com.cn/ecosystem/Cube/STM32CubeMonitor)