

Đọc giá trị ADC trên STM8S hiển thị lên LCD : mình sẽ học cách sử dụng tính năng ADC trên Vi điều khiển STM8S103F3P6 của mình. ADC là một thiết bị ngoại vi rất hữu ích trên vi điều khiển thường được các nhà lập trình nhúng sử dụng để đo các đơn vị luôn thay đổi như điện áp, dòng điện, nhiệt độ, độ ẩm thay đổi, v.v.

Như mình biết “mình đang sống trong thế giới tương tự với các thiết bị kỹ thuật số”, nghĩa là mọi thứ xung quanh mình như tốc độ gió, cường độ ánh sáng, nhiệt độ và mọi thứ mình xử lý như tốc độ, vận tốc, áp suất, v.v. đều có bản chất tương tự. Nhưng bộ vi điều khiển và bộ vi xử lý của mình là thiết bị kỹ thuật số và chúng sẽ không thể đo các thông số này nếu không có một thiết bị ngoại vi quan trọng được gọi là Bộ chuyển đổi tương tự sang số (ADC) . Vì vậy, trong bài viết này, mình cùng tìm hiểu cách sử dụng **ADC trên Vi điều khiển STM8S** với trình biên dịch COMIC C.

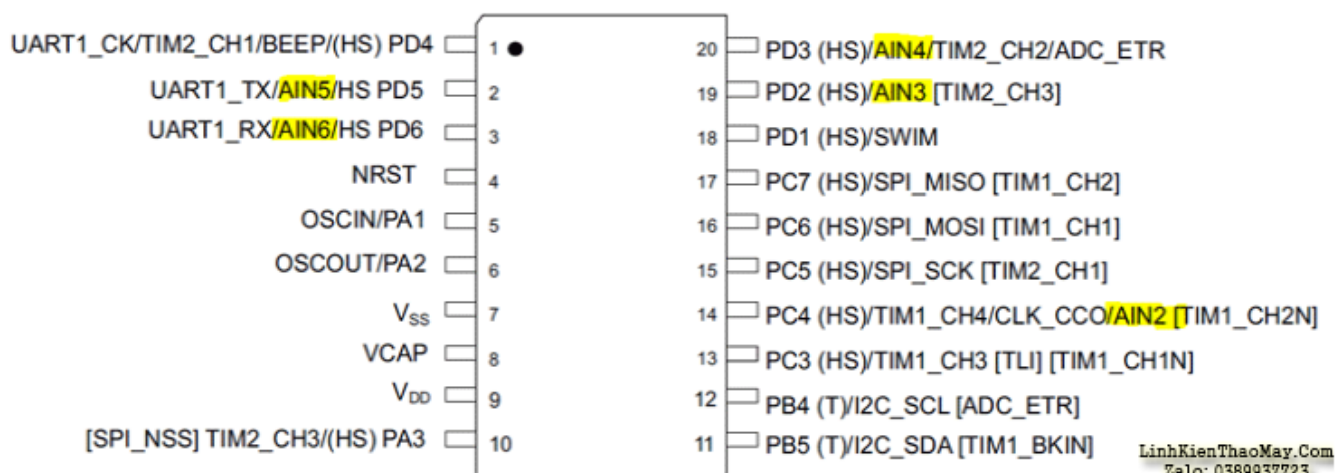
Linh kiện cần có

Trong bài viết này, mình sẽ đọc hai giá trị điện áp tương tự từ hai chiết áp và hiển thị giá trị ADC của nó trên màn hình LCD 16×2. Để làm điều này, mình sẽ cần các linh kiện sau.

- Bảng phát triển STM8S103F3P6
- Lập trình viên ST-Link V2
- LCD 16×2
- Chiết áp
- Kết nối dây
- Điện trở 1k

ADC trên STM8S103F3P6

Có nhiều loại ADC và mỗi vi điều khiển có thông số kỹ thuật riêng. Trên STM8S103F3P6, mình có một bộ ADC với **độ phân giải 5 kênh và 10 bit**; với độ phân giải 10-bit, mình sẽ có thể đo giá trị số từ 0 đến 1024 và ADC 5 kênh cho biết rằng mình có 5 chân trên Vi điều khiển có thể hỗ trợ ADC, 5 chân này được đánh dấu trong hình dưới đây.



Như bạn có thể thấy, tất cả năm chân này (AIN2, AIN3, AIN4, AIN5 và AIN6) được ghép với các thiết bị ngoại vi khác, có nghĩa là ngoài việc chỉ hoạt động như một chân ADC, các chân này cũng có thể được sử dụng để thực hiện các giao tiếp khác như ví dụ , chân 2 và 3 (AIN5

Tài liệu này được tải từ website: <http://linhkienthaomay.com>. Zalo hỗ trợ: 0389937723

và AIN 6) không chỉ có thể được sử dụng cho ADC mà còn có thể được sử dụng cho giao tiếp nối tiếp và các chức năng GPIO. Lưu ý rằng sẽ không thể sử dụng cùng một chân cho cả ba mục đích, vì vậy nếu mình đang sử dụng hai chân này cho ADC, thì mình sẽ không thể thực hiện giao tiếp nối tiếp. Các **đặc điểm ADC** quan trọng khác **cho STM8S103P36** có thể được tìm thấy trong bảng dưới đây lấy từ biểu dữ liệu.

ADC characteristics

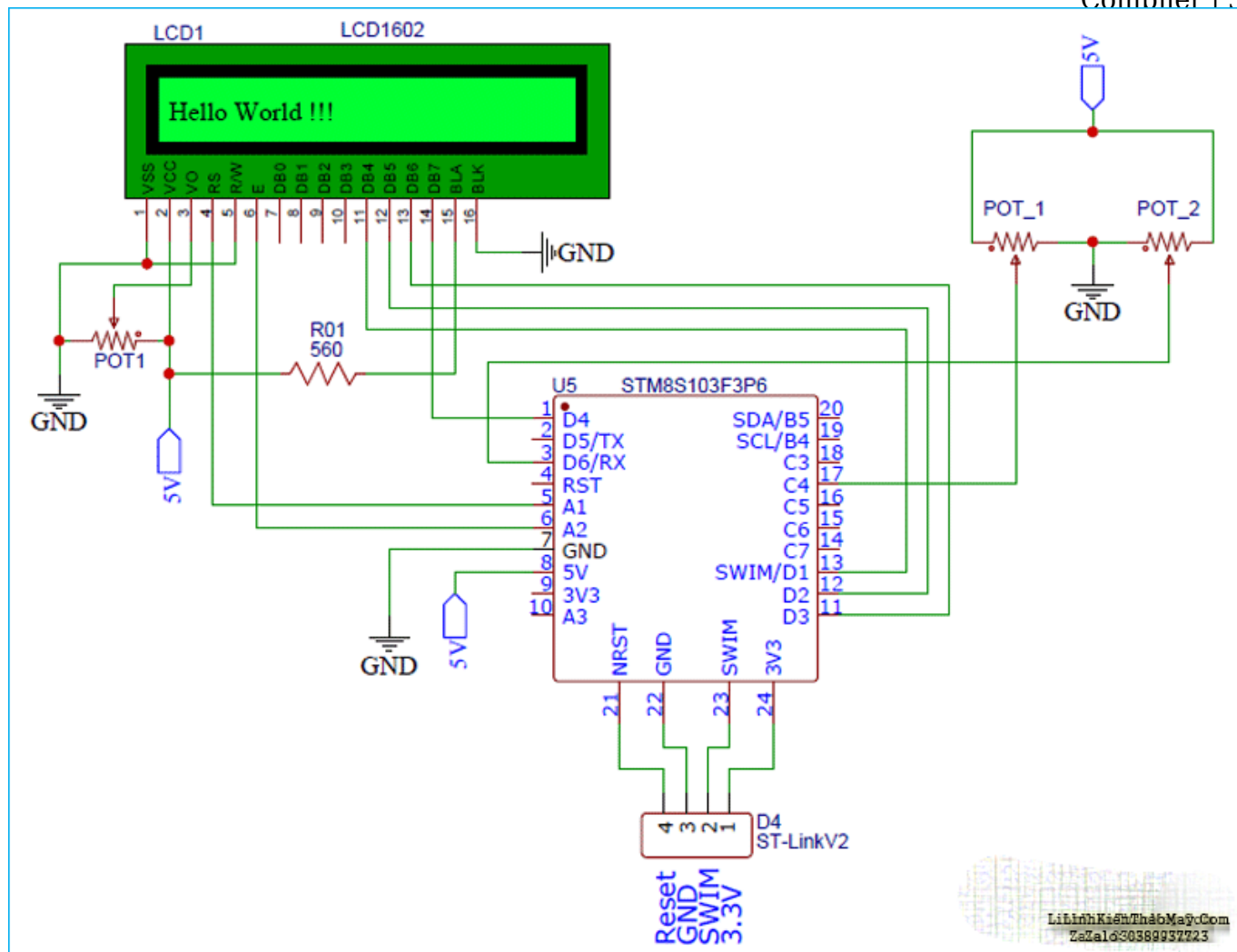
Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{ADC}	ADC clock frequency	$V_{DD} = 2.95 \text{ to } 5.5 \text{ V}$	1	-	4	MHz
		$V_{DD} = 4.5 \text{ to } 5.5 \text{ V}$	1	-	6	
V_{AIN}	Conversion voltage range ⁽¹⁾	-	V_{SS}	-	V_{DD}	V
C_{ADC}	Internal sample and hold capacitor	-	-	3	-	pF
$t_S^{(1)}$	Minimum sampling time	$f_{ADC} = 4 \text{ MHz}$	-	0.75	-	μs
		$f_{ADC} = 6 \text{ MHz}$	-	0.5	-	
t_{STAB}	Wakeup time from standby	-	-	7.0	-	μs
t_{CONV}	Minimum total conversion time (including sampling time, 10-bit resolution)	$f_{ADC} = 4 \text{ MHz}$	3.5			μs
		$f_{ADC} = 6 \text{ MHz}$	2.33			μs
		-	14			μs

LinhKienThaoMay.Com
Zalo: 0389937723

Trong bảng trên, Vdd đại diện cho điện áp hoạt động và Vss đại diện cho đất. Vì vậy, trong trường hợp của mình trên bảng phát triển của mình, mình có bộ vi điều khiển hoạt động trên 3.3V, bạn có thể kiểm tra sơ đồ mạch bảng phát triển ngay từ đầu với hướng dẫn STM8S. Với điện áp hoạt động là 3,3V, tần số đồng hồ ADC của mình có thể được đặt từ 1 đến 4MHz và phạm vi điện áp chuyển đổi của mình là từ 0V đến 3,3V. Điều này có nghĩa là ADC 10-bit của mình sẽ đọc 0 khi 0V (Vss) được cung cấp và sẽ đọc tối đa 1024 khi cung cấp 3,3V (Vdd). mình có thể dễ dàng thay đổi 0-5V này bằng cách thay đổi điện áp hoạt động của MCU nếu được yêu cầu.

Sơ đồ mạch để đọc giá trị ADC trên STM8S và hiển thị trên màn hình LCD

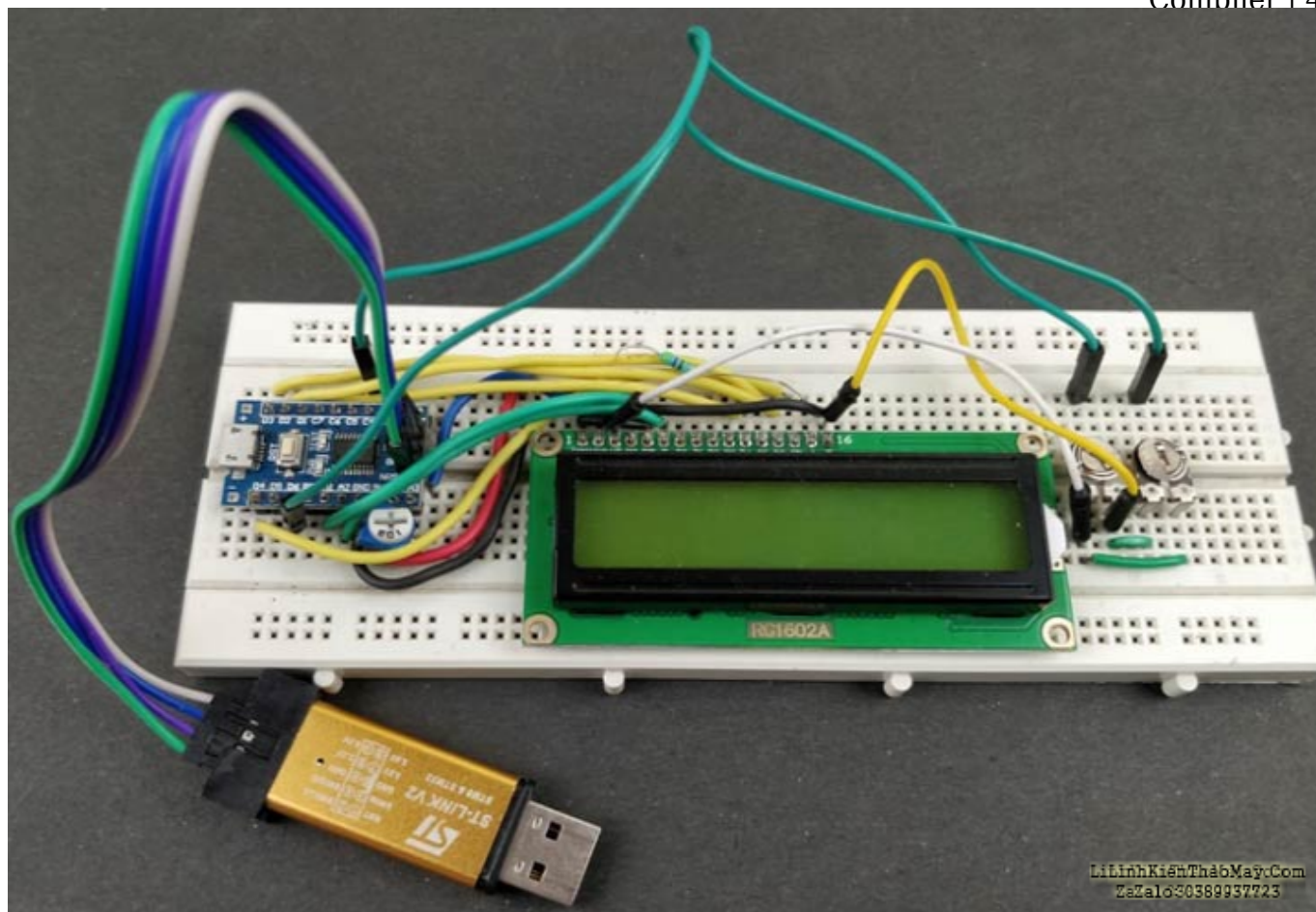
Sơ đồ mạch hoàn chỉnh được sử dụng trong Project này được đưa ra dưới đây, nó rất giống với hướng dẫn về LCD STM8S mà mình đã thảo luận trước đây.



Như bạn có thể thấy, các linh kiện bổ sung duy nhất ngoài màn hình LCD là hai chiết áp *POT_1* và *POT_2*. Các chiết này được kết nối với các cổng PC4 và PD6, là các chân ANI2 và ANI6 như đã thảo luận trên hình ảnh sơ đồ trước đó.

Chiết áp được kết nối theo cách mà khi mình thay đổi nó, mình sẽ thấy 0-5 V trên các chân tương tự của mình. mình sẽ lập trình bộ điều khiển của mình để đọc điện áp tương tự này ở giá trị kỹ thuật số (0 đến 1024) và hiển thị nó trên màn hình LCD. Sau đó, mình cũng sẽ tính toán giá trị điện áp tương đương và hiển thị nó trên màn hình LCD, hãy nhớ rằng bộ điều khiển của mình được cung cấp bởi 3.3V, vì vậy ngay cả khi mình cung cấp 5V cho chân ADC, nó sẽ chỉ có thể đọc từ 0V đến 3.3V.

Sau khi kết nối xong, phần cứng của mình trông giống như hình dưới đây. Bạn có thể thấy hai chiết áp ở bên phải và bộ lập trình liên kết ST ở bên trái.

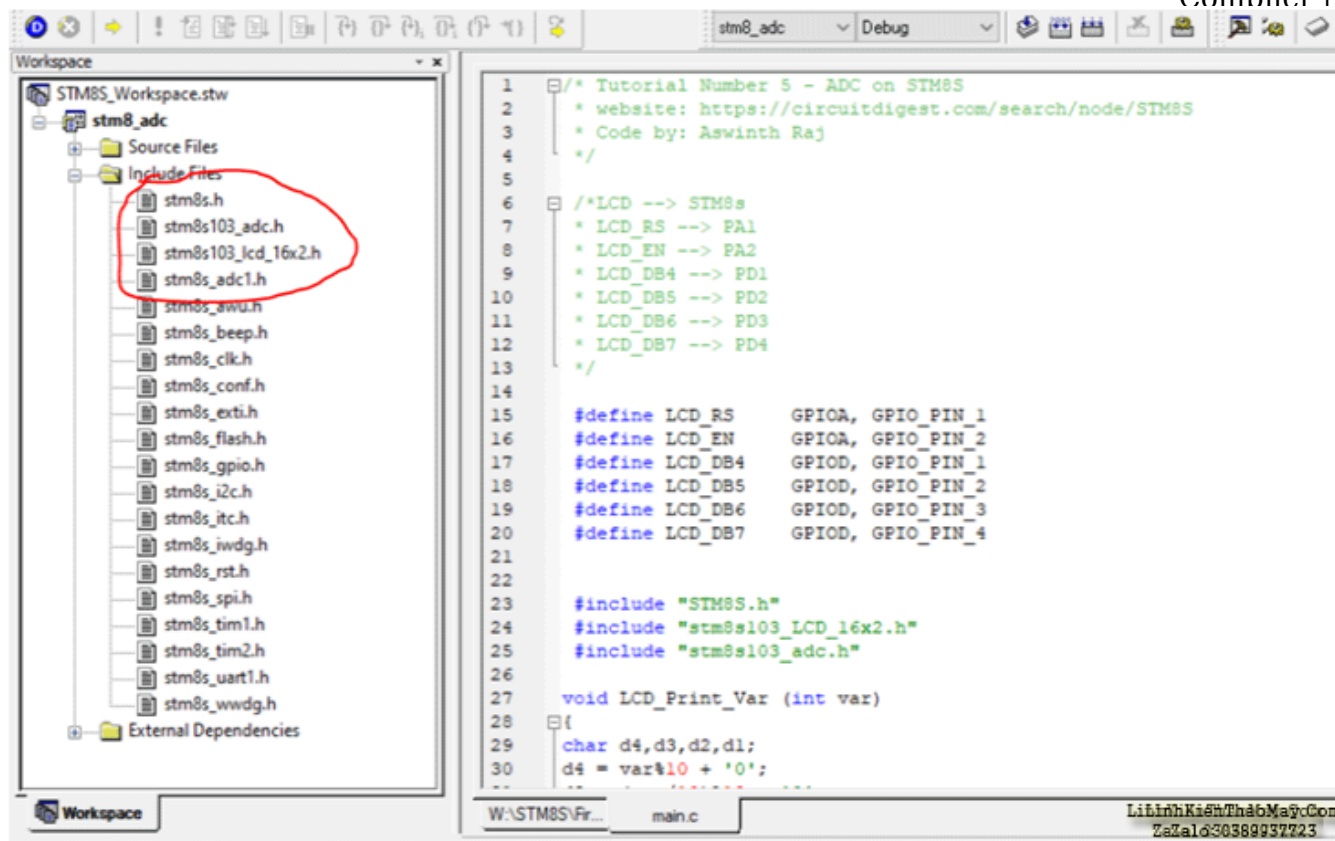


Thư viện ADC cho STM8S103F3P6

Để lập trình cho các chức năng ADC trên STM8S, mình sẽ sử dụng trình biên dịch Cosmic C cùng với các thư viện SPL. Nhưng để làm cho các quy trình dễ dàng hơn, mình đã tạo một tệp tiêu đề khác có thể được tìm thấy trên GitHub bằng liên kết bên dưới.

Thư viện ADC cho STM8S103F3P6 :
https://github.com/CircuitDigest/STM8S103F3_SPL/blob/master/inc/stm8s103_ADC.h

Nếu bạn biết mình đang làm gì, bạn có thể tạo tệp tiêu đề bằng cách sử dụng mã trên và thêm tệp đó vào thư mục “bao gồm tệp” trên trang Project của bạn. Ngoài ra, hãy làm theo phần bắt đầu với hướng dẫn STM8S để biết cách thiết lập môi trường lập trình và trình biên dịch của bạn. Khi thiết lập của bạn đã sẵn sàng, IDE của bạn phải có các tệp tiêu đề sau, ít nhất là những tệp được bao quanh bằng màu đỏ.



Tập tiêu đề trên bao gồm một hàm có tên là `ADC_Read ()`. Hàm này có thể được gọi trong chương trình chính của bạn để lấy giá trị ADC tại các chân nào. Ví dụ: `ADC_Read (AN2)` sẽ trả về giá trị ADC trên chân AN2. Chức năng được hiển thị bên dưới.

```
unsigned int ADC_Read (ADC_CHANNEL_TypeDef ADC_Channel_Number)
{
    unsigned int result = 0;
    ADC1_DeInit ();
    ADC1_Init (ADC1_CONVERSIONMODE_CONTINUOUS,
               ADC_Channel_Number,
               ADC1_PRESSEL_FCPU_D18,
               ADC1_EXTTRIG_TIM,
               DISABLE,
               ADC1_ALIGN_RIGHT,
               ADC1_SCHMITTTRIG_ALL,
               DISABLE);
    ADC1_Cmd (BẬT);
    ADC1_StartConversion ();
    while (ADC1_GetFlagStatus (ADC1_FLAG_EOC) == FALSE);
    k_t qu = ADC1_GetConversionValue ();
    ADC1_ClearFlag (ADC1_FLAG_EOC);
    ADC1_DeInit ();
}
```

Như bạn thấy, mình có thể chuyển tám tham số cho hàm này và điều này xác định cách ADC được cấu hình. Trong mã thư viện của mình ở trên, mình đã đặt chế độ chuyển đổi thành liên tục và sau đó nhận số kênh được truyền vào một tham số. Và sau đó, mình phải đặt tần số

CPU của bộ điều khiển của mình, theo mặc định (nếu bạn chưa kết nối tinh thể bên ngoài), STM8S của bạn sẽ hoạt động với bộ dao động nội bộ 16Mhz. Vì vậy, mình đã đề cập "ADC1_PRESSEL_FCPU_D18" làm giá trị tỷ lệ trước. Bên trong hàm này, mình đang sử dụng các phương thức khác được định nghĩa bởi tệp tiêu đề SPL *stm8s_adc1.h*. mình bắt đầu bằng cách Khởi tạo các chân ADC và sau đó là *ADC1_Init()* để khởi tạo ngoại vi ADC. Định nghĩa của chức năng này từ hướng dẫn sử dụng SPL được hiển thị bên dưới.

```
void ADC1_Init ( ADC1_ConvMode_TypeDef ADC1_ConversionMode,
                 ADC1_Channel_TypeDef ADC1_Channel,
                 ADC1_PresSel_TypeDef ADC1_PrescalerSelection,
                 ADC1_ExtTrig_TypeDef ADC1_ExtTrigger,
                 FunctionalState ADC1_ExtTriggerState,
                 ADC1_Align_TypeDef ADC1_Align,
                 ADC1_SchmittTrigq_TypeDef ADC1_SchmittTriggerChannel,
                 FunctionalState ADC1_SchmittTriggerState
               )
```

Initializes the ADC1 peripheral according to the specified parameters.

Parameters:

ADC1_ConvMode,:	specifies the conversion mode can be one of the values of ADC1_ConvMode_TypeDef .
ADC1_Channel,:	specifies the channel to convert can be one of the values of ADC1_Channel_TypeDef .
ADC1_PrescalerSelection,:	specifies the ADC1 prescaler can be one of the values of ADC1_PresSel_TypeDef .
ADC1_ExtTrigger,:	specifies the external trigger can be one of the values of ADC1_ExtTrig_TypeDef .
ADC1_ExtTriggerState,:	specifies the external trigger new state can be one of the values of FunctionalState .
ADC1_Align,:	specifies the converted data alignment can be one of the values of ADC1_Align_TypeDef .
ADC1_SchmittTriggerChannel,:	specifies the schmitt trigger channel can be one of the values of ADC1_SchmittTrigq_TypeDef .
ADC1_SchmittTriggerState,:	specifies the schmitt trigger state can be one of the values of FunctionalState .

Return values:

None

LinhKienThaoMay.Com
Zalo: 0389937723

Tiếp theo, mình đặt trình kích hoạt bên ngoài bằng cách sử dụng bộ hẹn giờ và tắt trình kích hoạt bên ngoài vì mình sẽ không sử dụng nó ở đây. Và sau đó, mình có căn chỉnh được đặt ở bên phải và hai tham số cuối cùng được sử dụng để đặt kích hoạt Schmitt, nhưng mình sẽ tắt nó trong hướng dẫn này. Vì vậy, nói ngắn gọn, mình sẽ ADC của mình làm việc ở chế độ chuyển đổi liên tục trên chân ADC được yêu cầu với trình kích hoạt bên ngoài và trình kích hoạt Schmitt bị vô hiệu hóa. Bạn có thể kiểm tra biểu dữ liệu nếu bạn cần thêm thông tin về cách sử dụng tùy chọn trình kích hoạt bên ngoài hoặc trình kích hoạt Schmitt, mình sẽ không thảo luận về điều đó trong hướng dẫn này.

Code STM8S Đọc giá trị ADC trên STM8S

Bạn có thể tìm thấy mã hoàn chỉnh được sử dụng trong tệp *main.c* ở cuối trang này. Sau khi thêm các tệp tiêu đề và tệp nguồn bắt buộc, bạn sẽ có thể biên dịch trực tiếp tệp chính. Giải thích về mã trong tệp chính như sau. mình sẽ không giải thích chương trình LCD STM8S vì mình đã thảo luận về điều đó trong hướng dẫn trước.

Mục đích của mã sẽ là đọc các giá trị ADC từ hai chân và chuyển đổi nó thành giá trị điện áp. mình cũng sẽ hiển thị cả giá trị ADC và giá trị Điện áp trên màn hình LCD. Vì vậy, mình đã sử dụng một hàm có tên *LCD_Print* Var nhận một biến ở định dạng số nguyên và chuyển đổi nó thành một ký tự để hiển thị trên màn hình LCD. mình đã sử dụng các toán tử mô đun đơn giản (%) và chia (/) để lấy từng chữ số từ biến và đặt vào các biến như d1, d2, d3 và d4

như hình dưới đây. Sau đó, mình có thể sử dụng hàm *LCD_Print_Char* để hiển thị các ký tự này trên màn hình LCD.

```
void LCD_Print_Var (int var)
{
char d4, d3, d2, d1;
d4 = var% 10 + '0';
d3 = (var / 10)% 10 + '0';
d2 = (var / 100)% 10 + '0';
d1 = (var / 1000) + '0';
Lcd_Print_Char (d1);
Lcd_Print_Char (d2);
Lcd_Print_Char (d3);
Lcd_Print_Char (d4);
}
```

Tiếp theo trong hàm main, mình có bốn biến được khai báo. Hai trong số chúng được sử dụng để lưu giá trị ADC (0 đến 1024) và hai trong số đó được sử dụng để lấy giá trị điện áp thực tế.

```
unsigned int ADC_value_1 = 0;
unsigned int ADC_value_2 = 0;
int ADC_voltage_1 = 0;
int ADC_voltage_2 = 0;
```

Tiếp theo, mình phải chuẩn bị các chân GPIO và cấu hình đồng hồ để đọc điện áp tương tự. Ở đây mình sẽ đọc điện áp tương tự từ các chân AIN2 và AIN6 lần lượt là các chân PC4 và PD6. mình phải xác định các pin này ở trạng thái thả nổi như hình dưới đây. mình cũng sẽ kích hoạt thiết bị ngoại vi đồng hồ cho ADC.

```
CLK_PeripheralClockConfig(CLK_PERIPHERAL_ADC,  ENABLE);  //Enable
Peripheral Clock for ADC
GPIO_Init (GPIOC, GPIO_PIN_4, GPIO_MODE_IN_FL_IT);
GPIO_Init (GPIOC, GPIO_PIN_4, GPIO_MODE_IN_FL_IT);
```

Bây giờ các chân đã sẵn sàng, mình phải đi vào vòng lặp while vô hạn để đọc điện áp tương tự. Vì mình có tệp tiêu đề của mình, mình có thể dễ dàng đọc điện áp tương tự từ các chân AIN2 và AIN 6 bằng cách sử dụng các dòng dưới đây.

```
ADC_value_1 = ADC_Read (AIN2);
ADC_value_2 = ADC_Read (AIN6);
```

Bước tiếp theo là chuyển đổi số đọc ADC này (0 đến 1023) thành điện áp tương tự. Bằng cách này, mình có thể hiển thị giá trị điện áp chính xác được cung cấp cho chân AIN2 và AIN6. Các công thức tính điện áp tương tự có thể được đưa ra bởi-

Analog Voltage = ADC Reading * (3300/1023)

Trong trường hợp của mình trên bộ điều khiển STM8S103F3, mình có ADC với độ phân giải 10-bit, vì vậy mình đã sử dụng $1023 (2^10)$. Cũng trong quá trình phát triển của mình, bộ điều khiển có 3.3V là 3300, vì vậy mình chia 3300 cho 1023 trong công thức trên. Khoảng $3300/1023$ sẽ cho mình 3,226, vì vậy trong chương trình của mình, mình có các dòng sau để đo điện áp ADC thực tế bằng cách sử dụng điện áp ADC.

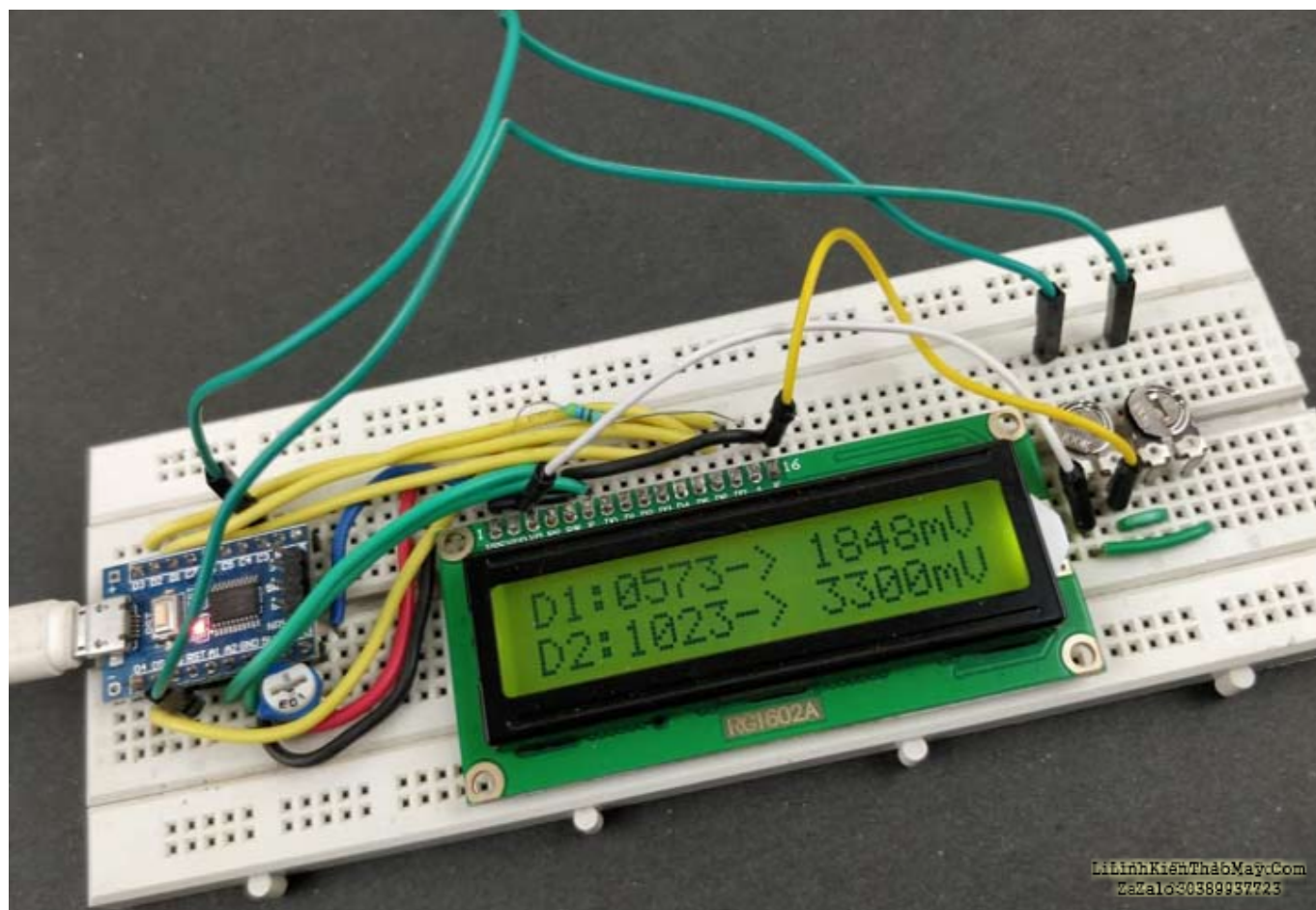
```
ADC_voltage_1 = ADC_value_1 * (3.226); //(3300/1023 =~ 3.226)convert  
ADC value 1 to 0 to 3300mV
```

```
ADC_voltage_2 = ADC_value_2 * (3.226); //convert ADC value 1 to 0 to  
3300mV
```

Phần còn lại của mã chỉ được sử dụng để hiển thị bốn giá trị này trên màn hình LCD. mình cũng có độ trễ 500ms để màn hình LCD được cập nhật sau mỗi 500mS. Bạn có thể giảm điều này hơn nữa nếu bạn cần cập nhật nhanh hơn.

Đọc điện áp tương tự từ hai chiết áp bằng STM8S

Biên dịch mã và tải nó lên bảng phát triển của bạn. Nếu bạn gặp các lỗi biên dịch nào, hãy đảm bảo rằng bạn đã thêm tất cả các tệp tiêu đề và tệp nguồn như đã thảo luận trước đó. Sau khi mã được tải lên, bạn sẽ thấy một thông báo chào mừng nhỏ có nội dung “ADC trên STM8S” và sau đó bạn sẽ thấy màn hình bên dưới.



MR. XÔ - 0901.679.359 - 80 Võ Thị Sáu, Phường Quảng Thuận, tx Ba Đồn, tỉnh Quảng Bình

GIÁ RẺ

NHANH CHÓNG

LINH KIỆN CHÍNH HÃNG

SANYO ELEC MSUNG
Panasonic TOSHIBA BISHI

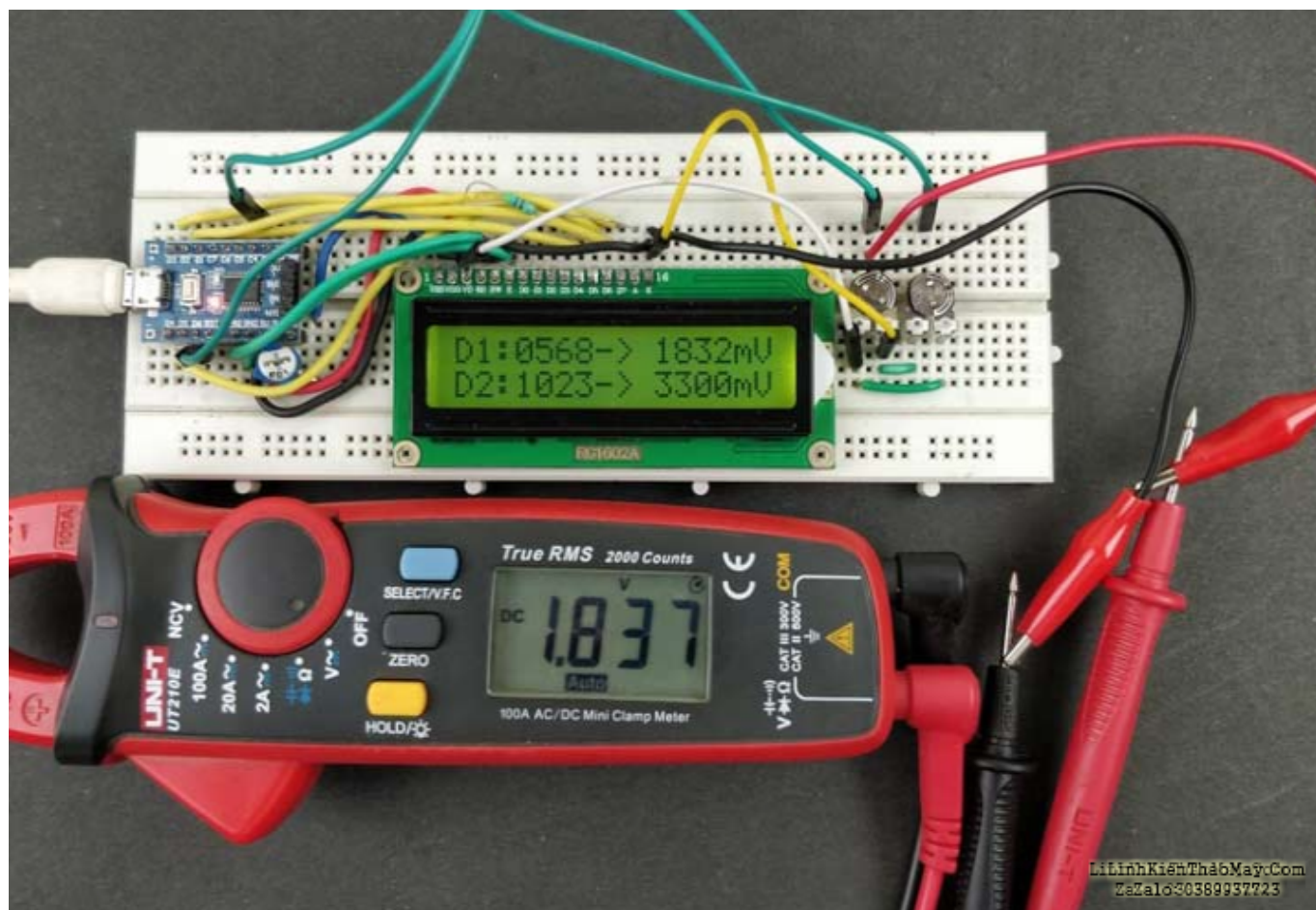


TRUNG TÂM SỬA CHỮA ĐIỆN TỬ XÔ NGUYỄN

- Dịch vụ sửa chữa điện tử tại nhà
- Cung cấp linh kiện điện tử
- Tư vấn lắp đặt nhà thông minh

Đc: Quảng Thuận. tx Ba Đồn,
tỉnh Quảng Bình - 0901.679.359

Giá trị D1 và D2 cho biết giá trị ADC từ chân Ain2 và AIN6 tương ứng. Ở phía bên phải, mình cũng có các giá trị điện áp tương đương được hiển thị. Giá trị này phải bằng điện áp xuất hiện trên chân AIN2 và AIN6 tương ứng. mình có thể kiểm tra giống nhau bằng cách sử dụng đồng hồ vạn năng, mình cũng có thể thay đổi chiết áp để kiểm tra xem giá trị điện áp cũng thay đổi tương ứng.



Hoàn thành công việc cũng có thể được tìm thấy trong video dưới đây. Hy vọng bạn thích hướng dẫn và học được điều gì đó hữu ích, nếu bạn có các câu hỏi nào, hãy để lại chúng trong phần bình luận bên dưới. Bạn cũng có thể sử dụng diễn đàn của mình để bắt đầu thảo

Tài liệu này được tải từ website: <http://linhkienthaomay.com>. Zalo hỗ trợ: 0389937723

luận hoặc đăng các câu hỏi kỹ thuật khác.

CODE

```
/* Tutorial Number 5 - ADC on STM8S
 * website: mobitool.net
 * Code by: Aswinth Raj
 */
/*LCD --> STM8s
 * LCD_RS --> PA1
 * LCD_EN --> PA2
 * LCD_DB4 --> PD1
 * LCD_DB5 --> PD2
 * LCD_DB6 --> PD3
 * LCD_DB7 --> PD4
 */
#define LCD_RS      GPIOA, GPIO_PIN_1
#define LCD_EN      GPIOA, GPIO_PIN_2
#define LCD_DB4     GPIOD, GPIO_PIN_1
#define LCD_DB5     GPIOD, GPIO_PIN_2
#define LCD_DB6     GPIOD, GPIO_PIN_3
#define LCD_DB7     GPIOD, GPIO_PIN_4
#include "STM8S.h"
#include "stm8s103_LCD_16x2.h"
#include "stm8s103_adc.h"
void LCD_Print_Var (int var)
{
char d4,d3,d2,d1;
d4 = var%10 + '0';
d3 = (var/10)%10 + '0';
d2 = (var/100)%10 + '0';
d1 = (var/1000) + '0';
Lcd_Print_Char(d1);
Lcd_Print_Char(d2);
Lcd_Print_Char(d3);
Lcd_Print_Char(d4);
}
void main()
{
//Variable declarations
unsigned int ADC_value_1 = 0;
unsigned int ADC_value_2 = 0;
int ADC_voltage_1 = 0;
int ADC_voltage_2 = 0;
Lcd_Begin();
Lcd_Clear();
```

```
Lcd_Set_Cursor(1,1);
Lcd_Print_String("ADC on STM8S");
CLK_PeripheralClockConfig(CLK_PERIPHERAL_ADC,  ENABLE);  //Enable
Peripheral Clock for ADC
GPIO_Init (GPIOC, GPIO_PIN_4, GPIO_MODE_IN_FL_IT);
delay_ms(5000);
Lcd_Clear();
Lcd_Set_Cursor(1,1);
Lcd_Print_String("D1:      ->      mV");
Lcd_Set_Cursor(2,1);
Lcd_Print_String("D2:      ->      mV");
while (1)
{
ADC_value_1 = ADC_Read (AIN2);
ADC_value_2 = ADC_Read (AIN6);
ADC_voltage_1 = ADC_value_1 * (3.226); //(3300/1023 =~ 3.226)convert
ADC value 1 to 0 to 3300mV
ADC_voltage_2 = ADC_value_2 * (3.226); //convert ADC value 1 to 0 to
3300mV
Lcd_Set_Cursor(1,4);
LCD_Print_Var (ADC_value_1);
Lcd_Set_Cursor(1,11);
LCD_Print_Var (ADC_voltage_1);
Lcd_Set_Cursor(2,4);
LCD_Print_Var (ADC_value_2);
Lcd_Set_Cursor(2,11);
LCD_Print_Var (ADC_voltage_2);
delay_ms(500);
}
}
```

Các bài viết tương tự:

1. [Giá trị True RMS, AVG, Peak là gì?](#)
2. [Cách đọc giá trị điện trở thường qua mã màu](#)
3. [dạ em có con quạt hơi nước hiện tượng các nút ok riêng nút nguồn ko hư hỏng bấm ko tác dụng,,,khi bấm nút tắt ko tác dụng bấm nút này đèn led hiển thị của các nút yếu đi,,,mạch in dẫn tới nút ăn thẳng vào vì sử lý ko qua trở,,,,em chưa kiểm tra nguồn - laoj quạt này\(quạt hơi nước\) cắm nguồn bấm nút chức năng số\(tốc độ\),hoặc quay hoặc hẹn giờ hoặc tạo âm vãn bình thường riêng nút tắt ko tắt dc,,,nguyên bản là tắt dc nhưng giờ là ko tắt dc](#)
4. [đầu DVD calinew - tình hình rất chi là tình hình mấy a c, giúp e voi, hiện tượng ko đọc đc đĩa DVD chỉ đọc đc VCD, thay mất từ đầu đang chạy sang ko đc, đi chạy lại rôm thì thẳng quai dở kia chạy kiểu j hết lên lun, lắp mất đang chạy vào máy cũ đang chạy thì cũng ko đọc đc DVD lun, vậy là thành hư cả 2 lun bó tay chưa sửa xong cho khách mà](#)

hư lun của mình

5. Đầu dvd xe tải. - Đợt trước làm con này bị ko đọc usb, ko đọc đĩa, thay tổng đã chạy tốt. Chiều nax khách đến lấy thử cho khách thì usb đọc, đĩa thì ko nhận đưa đĩa vào nó ko kéo đĩa vào ko nhận lệnh .huhu chán wá
6. Điều hoà Daikin 2 chiều Inverter gas R32 - Cánh đảo gió khi mới hoạt động thì dừng ở đúng vị trí cài đặt, chạy một lúc cánh tự chuyển lên trên (vị trí cánh gió giống chức năng Auto comfort) mặc dù mình không cài đặt thêm gì
7. Làm thế nào để đọc giá trị tụ điện?
8. lcd acer v173 - khi e cắm nguồn vào thì vẫn hiện logo,nhưng màn hình chỉ hiện thị khoảng mấy giây rồi tắt,khi rút cáp tín hiệu thì màn hình lại hiện thị không có tín hiệu được kết nối,e không kết nối cáp tín hiệu thì để cả tiếng không vấn đề gì và vẫn hiện thị không có thiết bị được kết nối
9. Màn lcd LG - Màn của em bật lên dùng dc khoảng 1-2h là xuất hiện chớp trắng rồi sau đó màn hình đen sì luôn không thấy gì nữa.đèn nguồn vẫn còn.Sau đó dù có tắt đi mở lại cũng ko lên nữa,khoảng vài h sau bật thì màn hình lại lên bình thường ,nhưng lại chỉ dc 1-2h sau thì lại đen sì.Tháo ra nhìn sơ qua thì linh kiên thấy vẫn còn mới ko thấy tụ nào bị phồng
10. Máy in cp1025 - Lỗi "Printer Mispick" - In test trên máy thì ok nhưng in xong thì nó báo đèn nhấp nháy đèn vàng và đèn xanh tắt. In trên PC thì in được lệnh đầu tiên in xong thì nó báo nhấp nháy đèn vàng và đèn xanh vẫn sáng, trên PC thì báo "Printer Mispick" và hỏi mình có "Resumed" không, mình ấn thì nó in lại lệnh trc đó, mình nhấn nút hủy lệnh nó cũng lại lệnh đó, đến khi mình tắt máy bật lên lại mới in lệnh tiếp theo được nhưng in xong cũng bị trường hợp trên, và khi mình in thì giấy vẫn còn trong khay.
11. Tea2025b sử dụng với mạch stereo - Tự nhiên 1 bên của e k còn nghe thấy nữa e đã ktra kĩ hết đầu input ổn cả lúc sau thử thì cả 2 bên đều k thấy rì cả e đã thay 2 con 16v450uf nhưng vẫn bị.
12. Tivi ASANO CS55DU3000 - cắm điện máy chạy bình thường, dùng đktx tắt chờ bật lên mất hiển thị (rút điện ra để 10 phút, cắm điện bật lên lại chạy bình thường, tắt chờ bật lên mất hiển thị)