

2. 제어 시스템의 기초 수학

2.7 실험 실습: Lab 2

2.7.1 A/D 컨버터

피드백 제어 시스템을 구성하기 위해서는 출력 신호를 측정해서 피드백 하는 센서가 필요하다. 센서의 출력은 아날로그 신호 또는 디지털 신호이다. 최근에, 디지털 신호를 출력하는 센서를 많이 사용하지만, 아날로그 신호를 출력하는 센서도 현재 많이 사용되고 있다. 마이크로컨트롤러가 아날로그 신호를 출력하는 센서의 신호를 받아들이기 위해서는 A/D(analog-to-digital) 컨버터가 필요하다. A/D 컨버터는 아날로그 전압을 읽어서 디지털 숫자로 변환하는 동작을 한다. STM32F429 Discovery 보드의 마이크로컨트롤러는 3개의 12비트 A/D 컨버터가 포함되어 있다. 또한, 각 A/D 컨버터의 입력에는 16개 채널을 입력할 수 있는 멀티플렉서(multiplexer) 회로가 있다. 그러나, 마이크로컨트롤러의 핀 개수의 제약으로 인해서 이러한 입력 채널을 모두 사용할 수 있는 것은 아니다.

A/D 컨버터의 프로그래밍에 대한 설명 이전에, 전압의 크기를 변환할 수 있는 OP 앰프 회로에 대한 설명이 필요하다. 실습용 보드의 마이크로컨트롤러에 인가되는 전압이 DC 3볼트이므로, A/D 컨버터와 D/A 컨버터가 입력 또는 출력할 수 있는 전압의 범위는 0~3 볼트이다. 그러나, 제어 시스템을 구성하기 위해서는 다양한 전압 범위의 신호를 입력하거나 출력하는 것이 필요할 수 있다. 특히, 어떤 경우에는 음의 전압과 양의 전압 사이의 전압 신호를 입력하거나 출력할 필요가 있다.

먼저, A/D 컨버터에 사용될 전압 변환 회로를 고려한다. 이 책에서는 -10V~10V의 전압 신호를 입력할 수 있는 회로를 구성해서 사용한다. 즉, 아래의 그림과 같이 -10~10V의 전압을 0~3V의 전압으로 변환하는 회로를 구성한다.

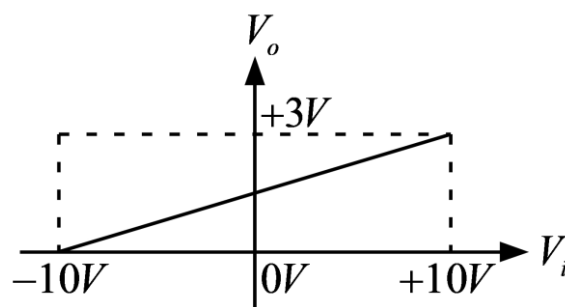


그림 2.15 A/D 컨버터를 위한 전압 변환

위의 그래프의 입출력 관계를 식으로 나타내면 다음과 같다.

$$V_o = \frac{3}{20}V_i + \frac{3}{2} \quad (2.227)$$

위의 식을 구현하는 회로는 여러 가지가 있을 수 있으며, 그림 2-16은 한가지 예를 보여준다. 이 회로에서 레퍼런스 전압 10볼트의 생성을 위해서는 Texas Instruments의 REF102와 같은 IC를 사용할 수 있다. 또한, 0 볼트가 정확하게 맞지 않는 경우를 대비해서, 0 볼트를 조절할 수 있는 가변 저항을 사용할 수 있다. 예를 들어서, 10볼트 전원에 연결된 20K옴의 저항 대신에 18K옴의 저항과 5K옴의 가변 저항을 연결하면 0 볼트의 전압을 조절하는 기능을 가지게 된다. 또한, 필요한 경우에는, 전압의 기울기를 조절할 수 있는 OP 앰프의 피드백 저항에도 가변 저항을 연결할 수 있다.

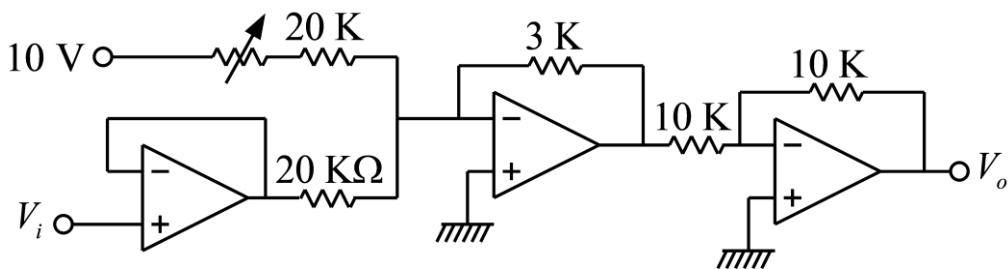


그림 2.16 A/D 컨버터를 위한 전압 변환 회로

다음으로, D/A 컨버터를 위한 전압 변환 회로를 고려해 본다. 이 경우에는, A/D 컨버터 전압 변환 회로와 반대로, 그림 2-17과 같이 0~3V의 전압이 입력되어 -10~10V 전압을 출력하는 회로를 구성한다.

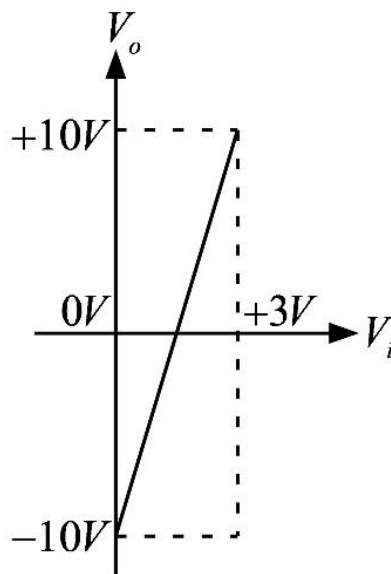


그림 2.17 D/A를 위한 전압 변환

위의 그래프의 입출력 관계를 식으로 나타내면 다음과 같다.

$$V_o = \frac{20}{3}V_i - 10 \quad (2.228)$$

위의 식을 구현하는 회로는 여러 가지가 있을 수 있으며, 그림 2-18은 한가지 예를 보여준다. 이 회로에서 레퍼런스 전압 10볼트는 A/D 컨버터의 레퍼런스 전압을 공유할 수 있다. 또한, 0 볼트가 정확하게 맞지 않는 경우를 대비해서, 0 볼트를 조절할 수 있는 가변 저항을 사용할 수 있다. 예를 들어서, 10볼트 전원에 연결된 10K옴의 저항 대신에 9K옴의 저항과 2K옴의 가변 저항을 연결하면 0 볼트의 전압을 조절하는 기능을 가지게 된다. 또한, 필요한 경우에는, 전압의 기울기를 조절할 수 있는 OP 앰프의 피드백 저항에도 가변 저항을 연결할 수 있다.

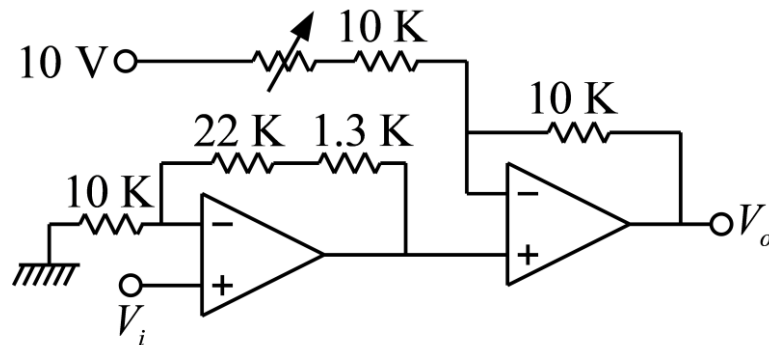


그림 2-18 D/A 컨버터를 위한 전압 변환 회로

위와 같은 회로를 구성한 후, 그림 2-19와 같이 A/D 컨버터의 입력 회로에는 함수 발생기를, D/A 컨버터의 출력 회로에는 오실로스코프를 연결한다.

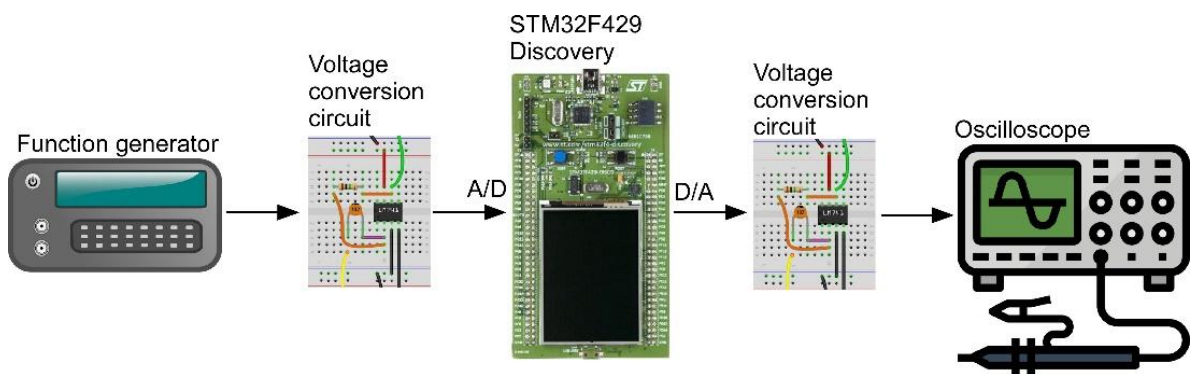


그림 2-19 A/D 컨버터와 D/A 컨버터의 시험

위에서 설명한 전압 변환 회로는 비교적 간단한 회로이므로 브레드 보드 등을 이용해서 구성할 수 있다. 그러나, 브레드 보드에서 구성한 회로는 접촉 불량 등의 문제가 발생할 수 있으므로, 안정적인 회로의 동작을 위해서는 가급적 PCB를 제작해서 구현하는 것을 권장한다. 아래의 사진은 저자가 이 책의 실습을 위해서 제작한 PCB를 보여준다. 이 PCB에는 위에서 설명한 전압 변환 회로 이외에 뒤의 장에서 사용하게 될 아날로그 다이내믹 시뮬레이터 회로도 포함하고 있다.



그림 2-20 Discovery 보드가 장착된 실습용 PCB

STM32CubeIDE에서 새로운 프로젝트를 만들고 프로젝트 이름을 ADDAconversion이라고 정한다. 새로운 프로젝트를 만드는 과정은 1장의 Lab1 실습 내용과 같으므로 반복하지 않는다. Lab1의 경우와 마찬가지로 FREERTOS를 비활성화 한다.

이 실습에서는 1장 Lab1의 **1.4.3 타이머 인터럽트와 D/A 변환**의 설정을 그대로 사용하므로, 1.4.3절에서 코드 생성하기 전까지 실시했던 과정을 그대로 반복한다. Lab1의 1.4.3절의 설정 이외에 이 실습에서는 추가로 A/D 컨버터를 사용하므로 A/D 컨버터의 설정이 필요하다. 그림 2.21은 A/D 컨버터의 설정 화면을 보여준다. 아날로그 입력을 위해서 ADC1의 채널 13을 사용하며, 입력 핀은 PC3이 지정된다.

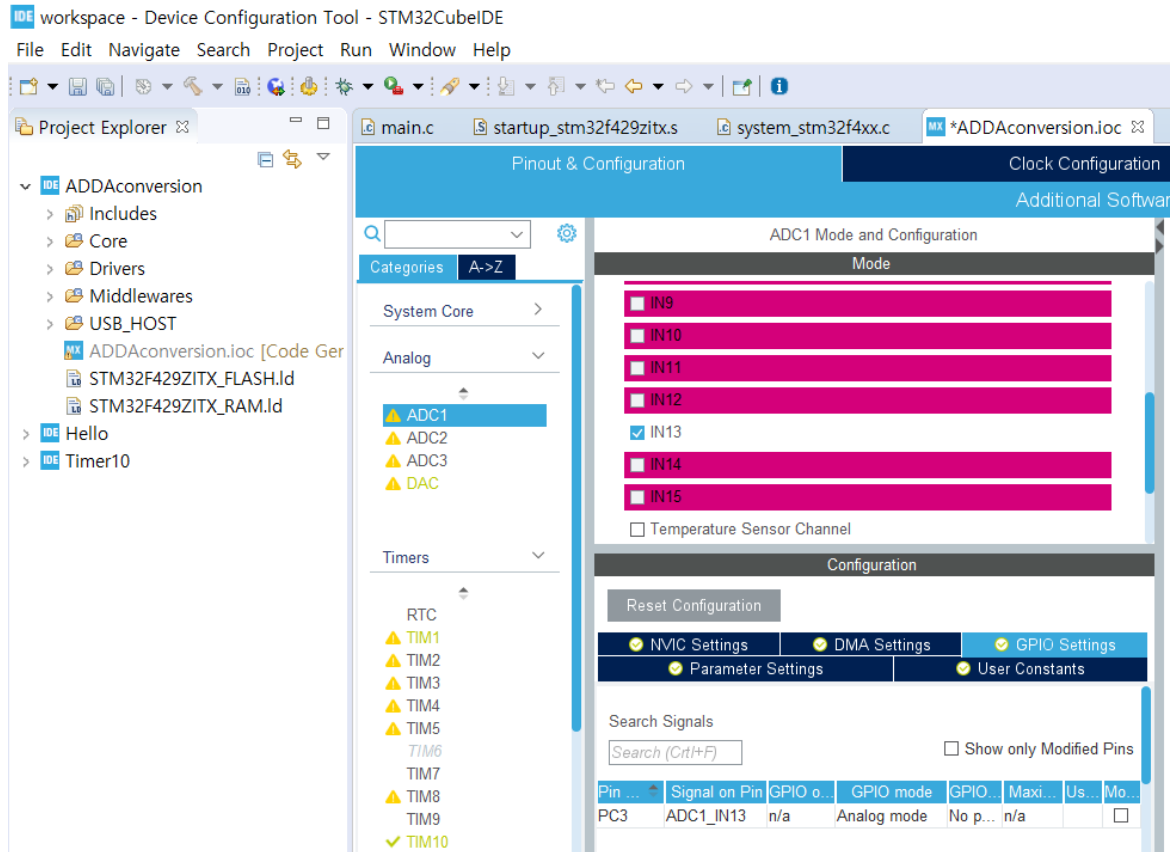


그림 2.21 A/D 변환기 설정

코드 생성(code generation)을 진행한 후, main.c에서 /* USER CODE BEGIN ... */ 과 /* USER CODE END ... */ 사이의 구간을 찾아서 아래의 코드 2.1과 코드 2.2를 입력한다. Lab1의 1.4.3절과 마찬가지로 타이머 인터럽트의 주기가 1msec 이며, 아래의 코드 2.2는 타이머 인터럽트 주기마다 불리는 callback 함수이다. 이 callback 함수에서 A/D 컨버터로 읽은 아날로그 입력 값을 그대로 D/A 컨버터로 출력한다.

코드 2.1

```
/* USER CODE BEGIN 2 */
HAL_TIM_Base_Start_IT(&htim10);
HAL_DAC_Start(&hdac, DAC_CHANNEL_2);
/* USER CODE END 2 */
```

코드 2.2

```
/* USER CODE BEGIN Callback 0 */
int da_value, ad_value;
if (htim->Instance == TIM10) {
    HAL_ADC_Start(&hadc1);
    if (HAL_ADC_PollForConversion(&hadc1, 10000) == HAL_OK)
    {
        ad_value = HAL_ADC_GetValue(&hadc1);
    }
    da_value = ad_value;
    HAL_DAC_SetValue(&hdac, DAC_CHANNEL_2, DAC_ALIGN_12B_R, (uint32_t)(da_value));
}
/* USER CODE END Callback 0 */
```

함수 발생기에서 주파수가 100 Hz, peak-to-peak 전압이 2V, offset이 0V인 정현파가 발생하도록 설정한다. 오실로스코프의 2개 채널을 이용해서 채널1은 A/D 컨버터에 입력되는 신호, 채널2는 D/A 컨버터에서 출력되는 신호를 관찰한다. 그림 2.22는 오실로스코프 화면을 보여주며, 100 Hz의 정현파를 1초에 1000번 샘플링 하므로 정현파 1주기 마다 10번 샘플링이 되는 것을 관찰 할 수 있다.

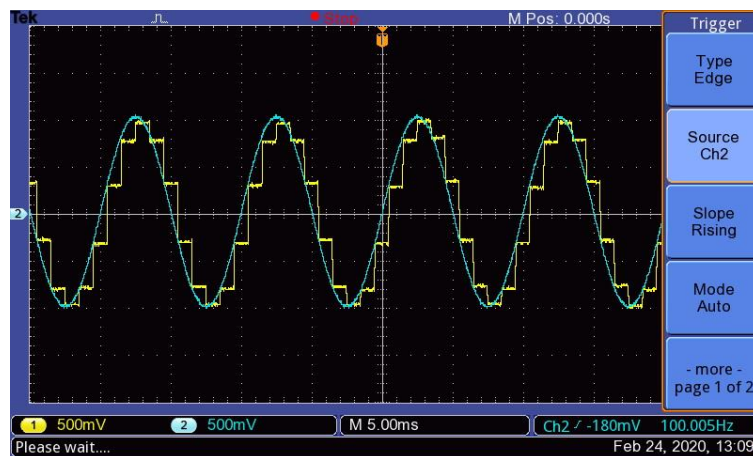


그림 2.22 오실로스코프 화면

실습 연습문제 2.1

D/A 변환기에서 출력되는 전압 신호가 입력 신호의 2배 크기 신호가 되는 프로그램을 작성하는 경우를 가정해 본다. 이를 위해서

```
da_value = ad_value;
```

의 코드를

```
da_value = 2*ad_value;
```

으로 바꾸어서 실행해 보고 결과를 관찰한다. 이와 같이 변경하면 출력 신호는 입력 신호의 2배 크기의 신호가 되지 않는 것을 관찰할 수 있다. 왜 그런지 이유를 생각해 보고 문제의 해결 방안에 대해서 생각해 본다.

실습 연습문제 2.1의 결과에서 A/D 컨버터로 입력된 수치는 산술 계산으로 사용할 수 없다는 것을 알 수 있다. 표 2.1은 A/D 컨버터 입력 전압과 수치의 관계를 보여준다. 여기에서 입력 전압은 마이크로컨트롤러의 A/D 변환 입력 단자가 아닌 전압 변환 회로에 입력되는 전압으로 -10V ~ 10V의 전압이다. A/D 컨버터의 종류마다 다를 수 있지만, 일반적으로 가장 낮은 전압에 가장 작은 숫자를 배정하고 가장 높은 전압에 가장 높은 숫자를 배정한다. 12비트 A/D 컨버터에서 가능한 숫자의 범위는 0~0xFFF 또는 0~4095 이므로 가장 낮은 전압에 0을 가장 높은 전압에 0xFFF 또는 4095를 배정한다. 이와 같이 A/D 컨버터에서 얻은 수치를 이용해서 산술 계산을 하기 위해서는 음의 전압에 대해서 음의 수치를 배정해야 하므로 A/D 컨버터로 얻은 수치를 2의 보수(two's complement)로 변환을 해야 한다. 표 2.1에서 볼 수 있듯이, 2의 보수로 변환을 하면 입력 전압 0V에 대해서 0의 숫자가 배정되고, 양(+)의 전압에 대해서는 양의 숫자를, 음(-)의 전압에 대해서는 음의 숫자가 배정된다. 2의 보수에서 가장 높은 자리의 비트는 부호 비트(sign bit)임을 기억한다. A/D 컨버터로 얻은 숫자를 2의 보수로 변환하는 과정은 간단하게 A/D 컨버터에서 읽은 숫자에서 2048을 빼면 2의 보수로 변환이 된다.

D/A 컨버터의 전압과 수치의 관계도 A/D 컨버터의 경우와 동일하다. 따라서 A/D 컨버터의 입력 수치를 2의 보수로 변환한 후에 산술 계산을 실행하고 D/A 컨버터로 출력하기 전에는 A/D 컨버터의 경우와 반대로 계산 결과 값에 2048을 더해준다.

표 2.1 입력 전압과 A/D 컨버터 입력 수치

Input Voltage(V)	A/D input number		Two's Complement	
	Hexadecimal	Decimal	Hexadecimal	Decimal
$(2047/2048)*10$	0x0fff	4095	0x07ff	2047
...
$(1/2048)*10$	0x0801	2049	0x0001	1
0	0x0800	2048	0x0000	0
$-(1/2048)*10$	0x07ff	2047	0xffff	-1
...
-10	0x0000	0	0xf800	-2048

실습 연습문제 2.2

D/A 컨버터에서 출력되는 신호의 크기가 A/D 컨버터에 입력되는 신호 크기의 2배가 되는 프로그램을 작성한다. 즉, A/D 변환기에서 입력된 숫자에서 2048을 빼서 2의 보수 숫자로 변환한다. 다음 2의 보수 숫자로 나타낸 숫자에 2를 곱한 후, 이 숫자에 2048를 더해서 D/A 변환기로 출력한다. 이와 같이 작성된 프로그램을 실행한 상태에서 A/D 컨버터에 정현파를 입력하고 D/A 컨버터의 출력 신호의 크기가 입력 신호 크기의 2배인지 확인한다.

2.7.2 1차 동적 시스템의 실시간 시뮬레이션

동적 시스템의 특성을 예측하기 위해서 컴퓨터를 이용한 시뮬레이션을 실시한다. MATLAB은 제어 시스템의 시뮬레이션을 할 수 있는 프로그램중에서 가장 많이 사용되는 프로그램이다. MATLAB과 같은 종류의 프로그램을 이용해서 시뮬레이션을 실행할 때, 대부분의 경우 시뮬레이션 프로그램이 동작하는 시간이 실제 시간과 일치하지 않으므로 실시간 시뮬레이션이 아니다. 시뮬레이션의 시간이 실제 시간과 일치하도록 하는 시뮬레이션이 실시간 시뮬레이션이며, 제어 시스템을 설계하고 제작하는 과정에서 실시간 시뮬레이션이 필요한 경우가 있을 수 있다.

이 실험에서는 마이크컨트롤러를 이용해서 간단한 시스템의 실시간 시뮬레이션을 실행해 본다. 다음과 같은 1차 미분 방정식의 동적 특성을 가지는 시스템의 실시간 시뮬레이션을 고려해 본다.

$$\tau \frac{dx(t)}{dt} + x(t) = u(t) \quad (2.229)$$

위의 식에서 $x(t)$ 는 출력, $u(t)$ 는 입력으로 정의한다. τ 는 시스템 응답 속도를 결정하는 시정수(time-constant)이다. 위의 식은 연속 시간 시스템(continuous-time system)이므로, 디지털 컴퓨터에서 실행할 수 있는 식의 형태로 변환해야 한다. 변환하는 방법에는 여러 가지 방법이 있지만, 다음 식과 같이 가장 쉽고 간단한 오일러(Euler)의 방법을 이용한다.

$$\frac{dx(t)}{dt} \approx \frac{x(t) - x(t - \Delta t)}{\Delta t} \quad (2.230)$$

위의 식은 미분을 근사적으로 계산하는 식이다. Δt 의 값이 충분히 작다면, 위의 식을 이용해서 미분의 근사값을 구할 수 있다. 식 (2.230)를 식 (2.229)에 대입하면 다음의 식을 얻을 수 있다.

$$\tau \frac{x(t) - x(t - \Delta t)}{\Delta t} + x(t) = u(t) \quad (2.231)$$

위의 식을 정리해서 다음과 같은 디지털 구현 식을 얻을 수 있다.

$$x(t) = \frac{1}{\tau + \Delta t} (\tau \cdot x(t - \Delta t) + \Delta t \cdot u(t)) \quad (2.232)$$

STM32CubeIDE에서 앞의 2.7.1절의 프로젝트와 동일한 방법으로 설정해서 코드 생성을 진행한 후, 아래에 주어진 코드를 main.c에 입력한다. 아래의 코드에서 샘플링 주기는 $\Delta t = 0.001$, 시정수는 $\tau = 0.025$ 이다.

코드 2.3

```
/* USER CODE BEGIN PV */
float x,x_old,u,tau,delt;
/* USER CODE END PV */
```

코드 2.4

```
/* USER CODE BEGIN 1 */
    tau = 0.025;
    delt = 0.001;
/* USER CODE END 1 */
```

코드 2.5

```
/* USER CODE BEGIN 2 */
    HAL_TIM_Base_Start_IT(&htim10);
    HAL_DAC_Start(&hdac, DAC_CHANNEL_2);
/* USER CODE END 2 */
```

코드 2.6

```
/* USER CODE BEGIN Callback 0 */
int da_value,ad_value;
if (htim->Instance == TIM10) {
    HAL_ADC_Start(&hadc1);
    if (HAL_ADC_PollForConversion(&hadc1, 10000) == HAL_OK)
    {
        ad_value = HAL_ADC_GetValue(&hadc1);
    }
    u = ad_value - 2048.0;
    x = (tau*x_old + delt*u)/(tau+delt);
    x_old = x;
    da_value = x + 2048.0;
    HAL_DAC_SetValue(&hdac, DAC_CHANNEL_2, DAC_ALIGN_12B_R, (uint32_t)(da_value));
}
/* USER CODE END Callback 0 */
```

함수 발생기에서 주파수가 4 Hz, peak-to-peak 전압이 2 V인 구형파(square wave)가 발생되도록 설정한다. 오실로스코프를 이용해서 A/D 변환기 입력과 D/A 변환기 출력을 오실로스코프로 관찰하면 그림 2.23과 같은 계단 응답(step response)를 볼 수 있다. 시정수가 25msec 임을 확인할 수 있다.

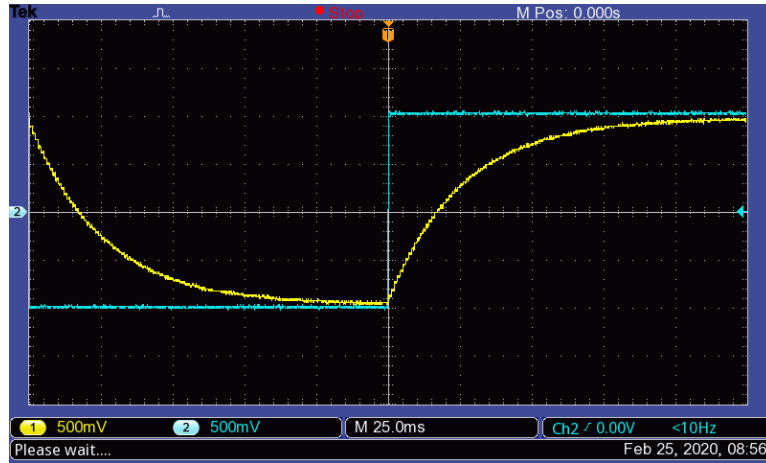


그림 2.23 실시간 시뮬레이션의 오실로스코프 화면

실습 연습문제 2.3

- (1) 위의 실험에서 시정수를 $\tau = 0.0125$ 로 변경해서 실행해 본다.
- (2) 동적 특성이 다음과 같은 시스템의 실시간 시뮬레이션을 실행해 본다.

$$\tau \frac{dx(t)}{dt} + 2x(t) = u(t) \quad (2.233)$$