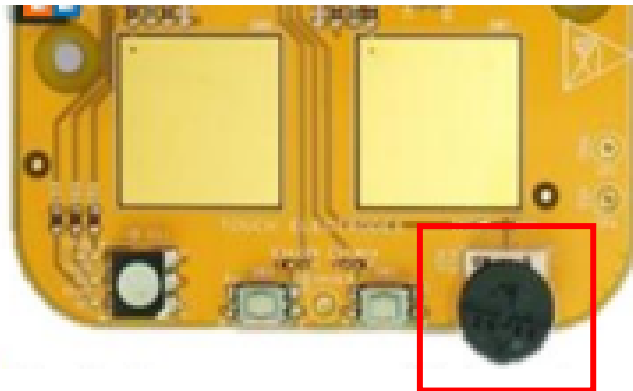


ADC 가변저항



Connect the pin for ADC

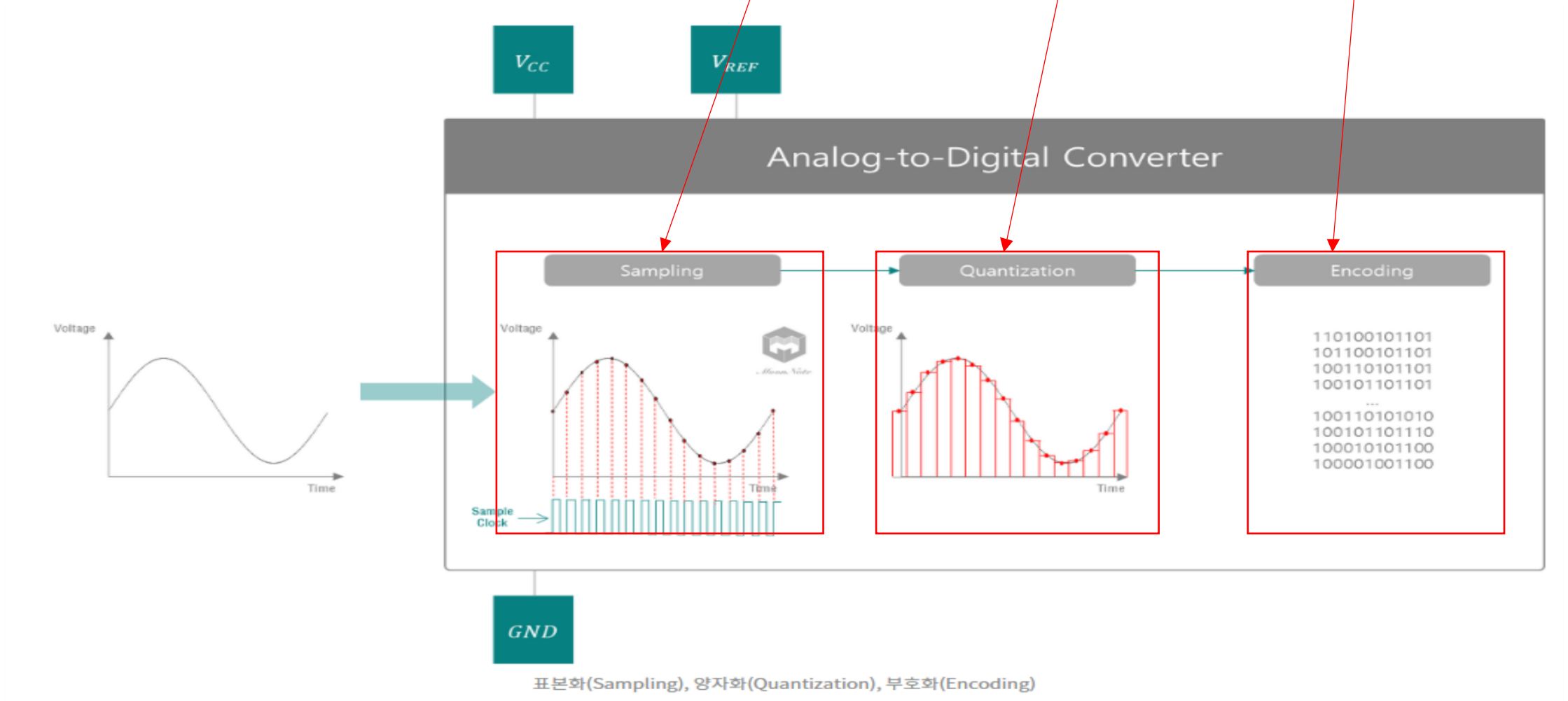
→ `convertAdcChan(13)` >> **PTC15**



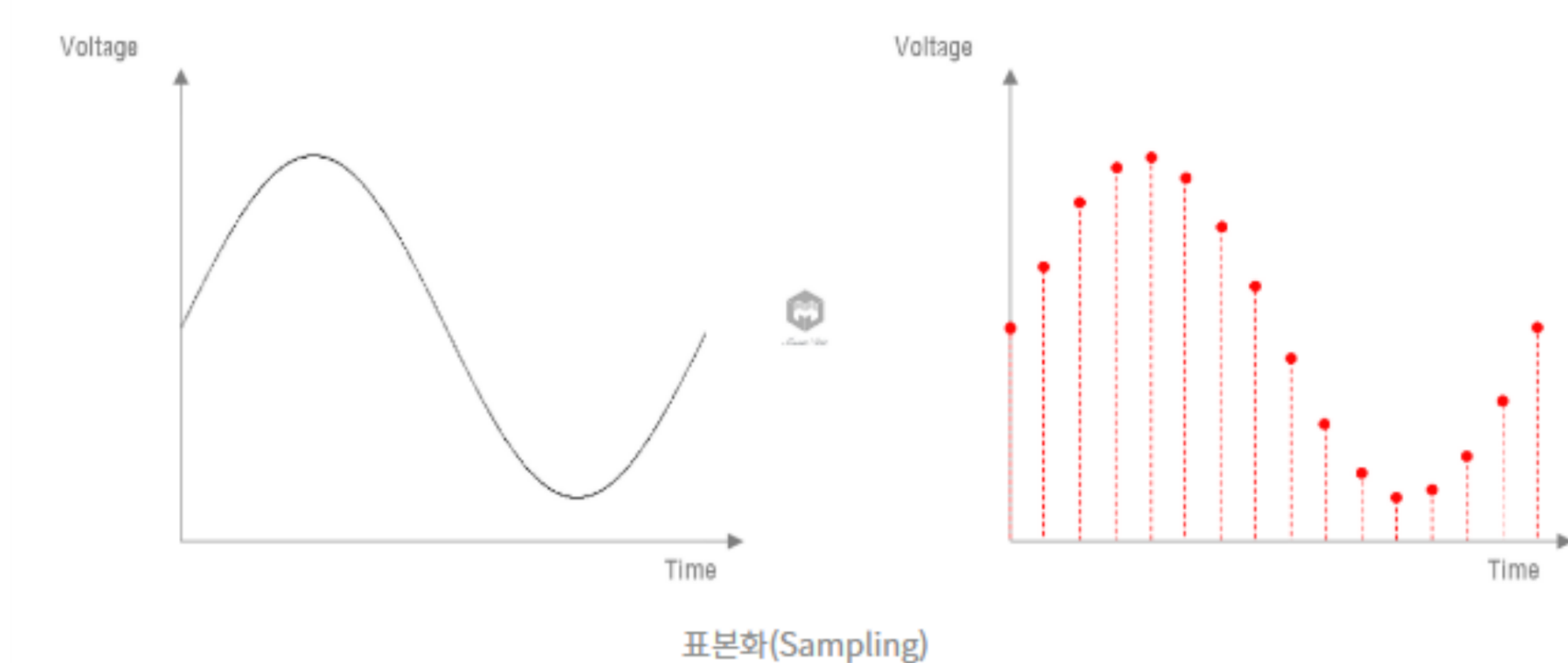
→ `convertAdcChan(12)` >> **None**

- MCU는 0v ~ 5v 사이의 값으로 다양한 전압 값을 전달 받음.
- MCU는 아날로그 값을 인식할 수 없기 때문에 디지털 값으로 변환해서 저장해주어야 함.
- 0v ~ 5v 의 아날로그 전압을 12비트 (2진수)로 인식하여 디지털 값으로는 0 ~ 4095 ($2^{12}-1$) 로 변환.
- Ex) 2.5v 의 디지털 값은 약 2047임을 확인가능.
 - -----> 산출법) $(2.5/5)*4095 = 2048$
- 여기서 비트가 클수록 더 많은 값의 표현이 가능해짐.
 - -----> 이는 분해능이 높다고 볼 수 있음.
- 예를 들어, 8비트 일 경우 0 ~ 255 (2^8-1) 의 범위로 12비트와 같이 세밀한 단위 표현이 어려움.
- (참고사항) 실제 특정 센서 활용 시, 해당 센서의 스펙으로 1v ~ 5v 의 작동 범위를 가지는 경우가 존재하는데, 12비트 기준으로 4095의 범위에서 1v 의 값을 버리도록 계산하여 실제로는 약 819.2 ~ 4096 의 범위가 된다고 볼 수 있다.

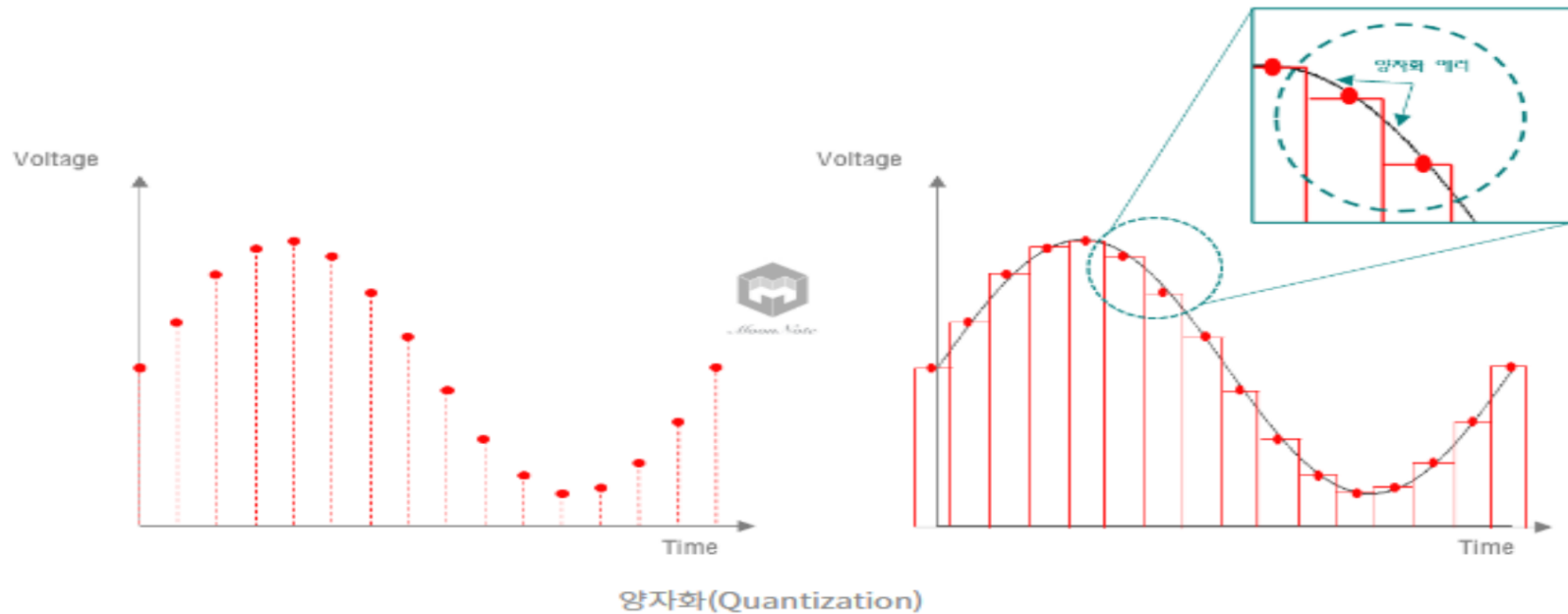
- ADC에서 아날로그 신호를 디지털화 하기 위해 **표본화(Sampling)**, **양자화(Quantization)**, **부호화(Encoding)** 으로 총 3단계 과정을 거침.



- **표본화 (Sampling)** – 아날로그 입력을 정기적으로 12비트 숫자로 변환하는 것
- 표본화는 아날로그 입력 신호를 일정한 간격으로 샘플링 하는 과정을 의미.
- 이 간격은 샘플링 주파수에 의해 결정.
- 고주파수 아날로그 신호를 낮은 주파수 디지털 데이터로 변환하려면 높은 샘플링 주파수가 필요.
- 샘플링 주파수는 나온 신호의 주파수보다 높아야 하며, 일반적으로 신호의 최고 주파수의 2배 이상이어야 함.



- **양자화 (Quantization)** – 각 샘플을 0부터 4095까지의 숫자 중 하나로 변환하는 것
- 양자화는 샘플링 된 아날로그 값들을 특정 범위 내의 디지털 값을 가지도록 변환하는 과정.
- 즉, 아날로그 신호를 연속적인 범위에서 디지털 값을 취하는 이산적인 값으로 변환하는 것을 의미.
- 양자화 수준은 ADC의 비트 해상도에 따라 결정되며, 더 높은 비트 해상도는 더 정밀한 측정을 의미.
- 양자화 과정에서 실제 값과 오차가 발생되게 되는데, 이를 양자화 에러라고 부름.



- **부호화 (Encoding)** – 각 양자화된 값이 부호화 되어 양수 또는 음수 값으로 표현하는 것
- 부호화는 양자화된 값들을 디지털 형식으로 표현하는 단계.
- 일반적으로, 부호화는 2의 보수 표현을 사용하여 부호화된 값을 나타냄.
- 부호 비트는 양수 또는 음수 값을 나타내며, 나머지 비트는 크기를 나타냄.

ADC.c file

```
void ADC_init(void)
```

- ADC0 의 초기화를 수행 부분 (ADC 의 설정 및 동작 구성)
- 'ADC0->CFG1', 'ADC0->CFG2', 'ADC0->SC2', 'ADC0->SC3' 등의 레지스터 설정을 통해 ADC 의 분해능, 샘플링 시간, 트리거 모드 등 정의

```
void ADC_init_HWTrigger(char Channel)
```

- 하드웨어 트리거 모드로 ADC 를 설정하는데 사용
- 트리거 모드에서 ADC 는 외부 이벤트에 의해 변환을 시작
- 특정 ADC 채널을 설정하고 하드웨어 트리거 모드 설정 후 ADC 변환 시작

사용 안되는 함수 부분

```
void convertAdcChan(uint16_t adcChan)
```

- ADC 채널에서 변환을 시작
- 'adcChan' 매개 변수로 주어진 ADC 채널에서 변환을 수행

```
uint8_t adc_complete(void)
```

- ADC 변환이 완료되었지 확인
- 'ADC0'의 'SC1[0]' 레지스터의 'COCO' 비트를 확인하여 변환 완료 시 1 반환, 그렇지 않으면 0 반환



- ADC0의 SC1[0] 레지스터의 COCO 비트란?
 - COCO "Conversion Over Complete"의 약자
 - 해당 비트는 현재 ADC 변환의 완료 여부를 나타내며, 소프트웨어가 ADC 변환의 상태 확인에 사용
 - `return ((ADC0->SC1[0] & ADC_SC1_COCO_MASK)>>ADC_SC1_COCO_SHIFT);`
 - 따라서 위 구문으로 변환 완료를 return < 완료 시 1 반환, 그렇지 않으면 0 반환 >

```
uint32_t read_adc_chx(void)
```

- ADC 변환 결과를 읽고 반환
- 변환 결과는 'ADC0'의 'R[0]' 레지스터에서 읽히며, 이 결과는 12비트 ADC 값으로 나타남
- 0부터 5000mV 범위로 변환 후 반환



- 5000을 ADC 결과 값에 곱하는 이유는 0 ~ 5V의 범위로 스케일링하는 것
- 즉, ADC의 결과는 0 ~ 4095의 범위에 있으며, 이를 다시 0 ~ 5V의 전압 범위로 변환한 것으로 결과적으로 mV 값을 얻고 보여주는 것

- SC1A, SC2, SC3, CFG1, CFG2 레지스터는 ADC (Analog-to-Digital Converter) 모듈의 레지스터
- 위 레지스터는 ADC 작업 및 설정을 제어하는데 사용

- **SC1A (Status and Control Register 1A)**

- SC1A는 ADC의 채널 선택 및 제어 레지스터.
- ADC 변환의 시작 및 제어에 사용.
- 주요 필드:
 - ADCH : 채널 선택 비트. ADC 모듈이 어떤 입력 채널에서 변환을 수행할지 지정.
 - AIEN : ADC 인터럽트 활성화 비트. 이 비트가 설정되면 변환이 완료될 때 인터럽트가 발생.

- **SC2 (Status and Control Register 2)**

- SC2는 ADC의 다양한 제어 기능을 제공하는 레지스터.
- 주요 필드:
 - ADTRG : ADC 트리거 선택 비트. 이 비트가 설정되면 하드웨어 또는 소프트웨어 트리거를 사용하여 변환이 시작.
 - ACFE : 비교 기능 활성화 비트. 설정하면 ADC 비교 기능을 활성화하고 비교된 결과를 얻을 수 있음.
 - ACFS : 비교 기능 상태 선택 비트. 비교 레지스터가 언제 업데이트 되는지 제어.
 - ACREN : 비교 범위 확장 비트. 설정하면 비교 결과가 주어진 범위 내에서 있는지 확인할 때 비트 범위가 확장.
 - DMAEN : DMA 활성화 비트. 설정하면 DMA 전송을 사용하여 ADC 데이터를 전송할 수 있음.

• SC3 (Status and Control Register 3)

- SC3는 ADC의 보정, 평균 및 다른 동작을 제어하는 레지스터.
- 주요 필드:
 - CAL : 보정 시작 비트. 이 비트를 설정하면 ADC 보정을 수행.
 - ADCO : 연속 운전 모드 활성화 비트. 설정하면 연속 변환이 실행.
 - AVGE : 평균 활성화 비트. 설정하면 하드웨어 평균 기능이 활성화.
 - AVGS : 평균 선택 비트. 평균 모드의 유형을 선택.

• CFG1 (Configuration Register 1)

- CFG1은 ADC의 입력 분주 및 작업 모드를 설정하는 레지스터.
- 주요 필드:
 - ADICLK : ADC 입력 클럭 선택 비트. 입력 클럭 소스를 선택.
 - ADIV : 입력 분주 비트. 입력 클럭을 나누는 분주 비율을 선택.
 - MODE : ADC 작업 모드 비트. ADC 변환이 8, 10 또는 12 비트 모드 중 어느 모드로 실행될지 선택.

• CFG2 (Configuration Register 2)

- CFG2는 ADC의 샘플링 시간을 설정하는 레지스터.
- 주요 필드:
 - SMPLTS : 샘플링 시간 비트. 샘플링 시간을 설정하여 변환 과정 중 입력 신호의 안정성을 제어.