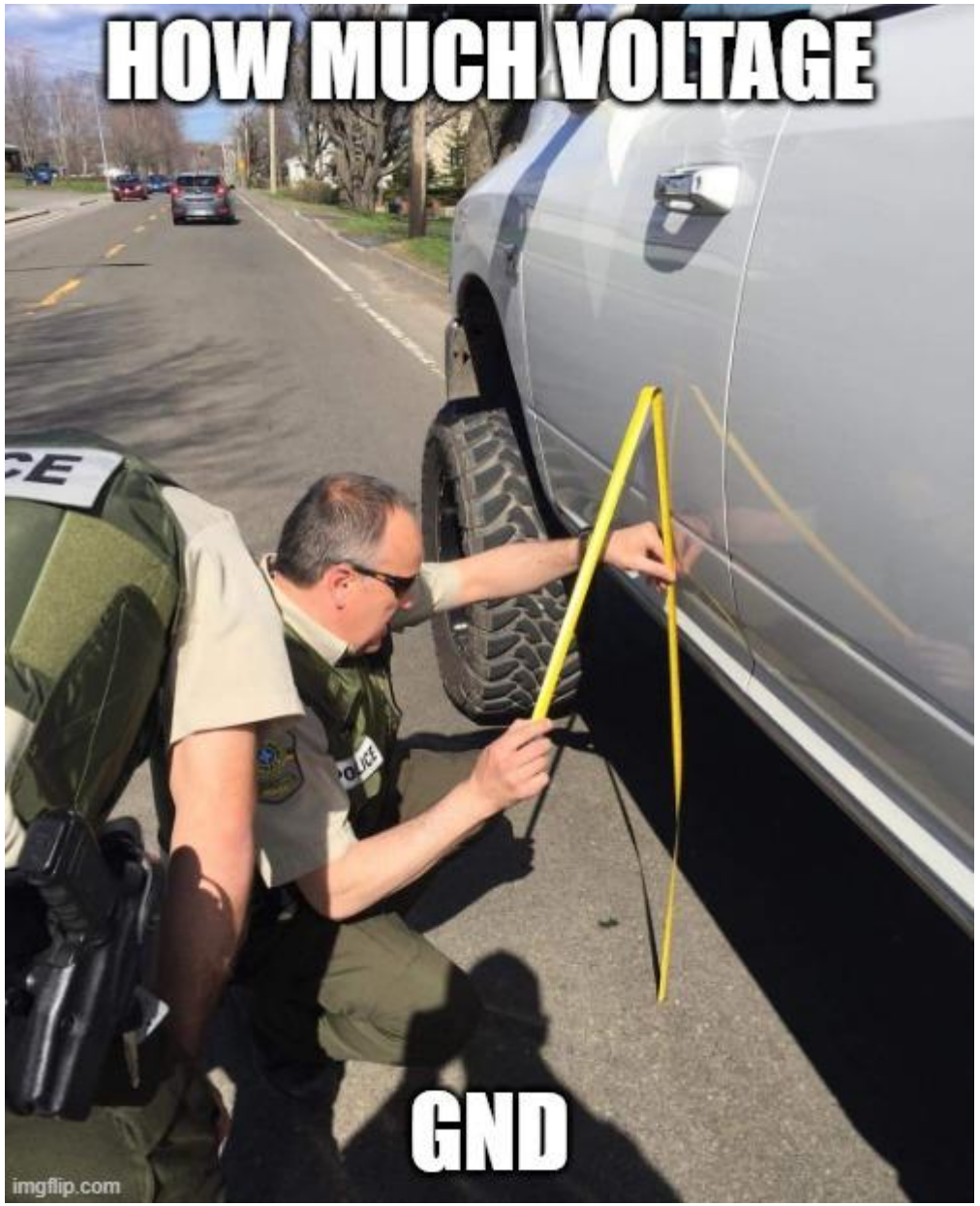
3.ADC (analog to digital convertor)

DR.SOMSIN THONGKRAIRAT


HOW MUCH VOLTAGE
GND
imgflip.com

What is Analog? (ปรัชญา)

Analog คือปริมาณหรือค่าต่างๆ ที่มีความต่อเนื่อง หรือปริมาณที่อยู่ในธรรมชาติ

เช่น

เวลา -> แปบนึง , ก่อนหน้านี้

ปริมาณน้ำฝนที่ตก -> ฝนตกพร้ำๆ , ฝนริน

ตำแหน่งของวัตถุต่างๆ -> ตรงนี้ , ห่างมากๆ

What is Digital (ปรัชญา)

การนำค่าที่อยู่ในธรรมชาติต่างๆ มาตีความ โดยใส่ กฎเกณฑ์ หรือ วิธีการในการ ตีความเข้าไป

เช่น

เวลา -> 1 นาที , 3 เดือน , 1 เทอม

ปริมาณน้ำฝนที่ตก -> ฝนตกหนัก , ฝนตกทำให้น้ำขัง 2 เมตร

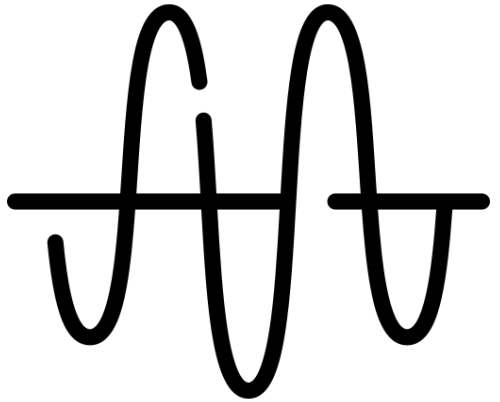
ตำแหน่งของวัตถุต่างๆ -> 1 cm , 2 km

What is Analog?

สัญญาณที่ยังไม่ผ่านการตีความให้เป็น logic 1 หรือ 0 โดยส่วนมากจะอยู่ในรูปแบบของ voltage (V) เนื่องจาก hardware สร้างได้ง่าย

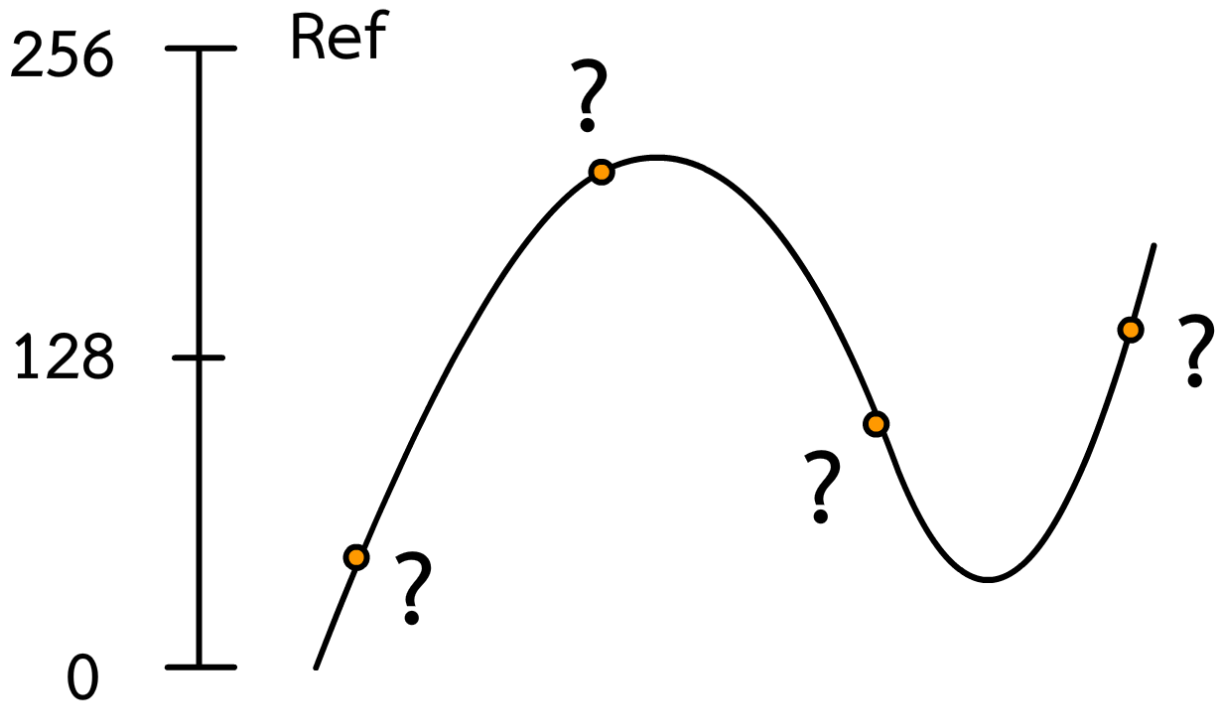
* หากต้องการวัด A หรือ R โดยมากจะแปลงให้เป็น V แล้วค่อยวัด V

ดังนั้น! ข้อมูลที่ถูกตีความด้วยกฎแล้ว เช่น logic 1 0



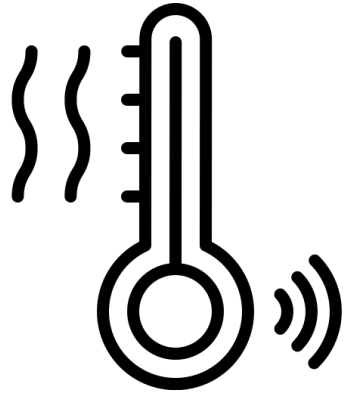
What is ADC

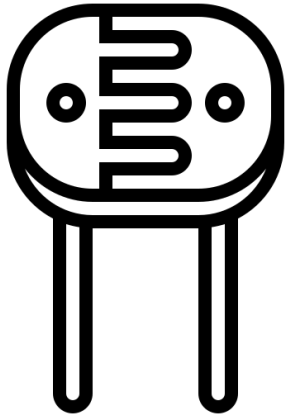
การตีความจากระดับแรงดัน (V) ให้กลายเป็นข้อมูลในระบบโดยมีจุดอ้างอิง (Vref)

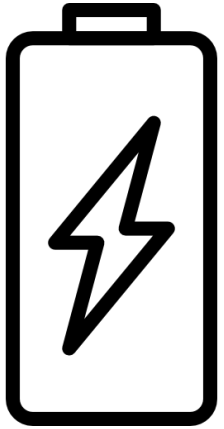


Why ADC?

- most of sensor and transducer provide analog value
- analog sensor is cheaper than digital



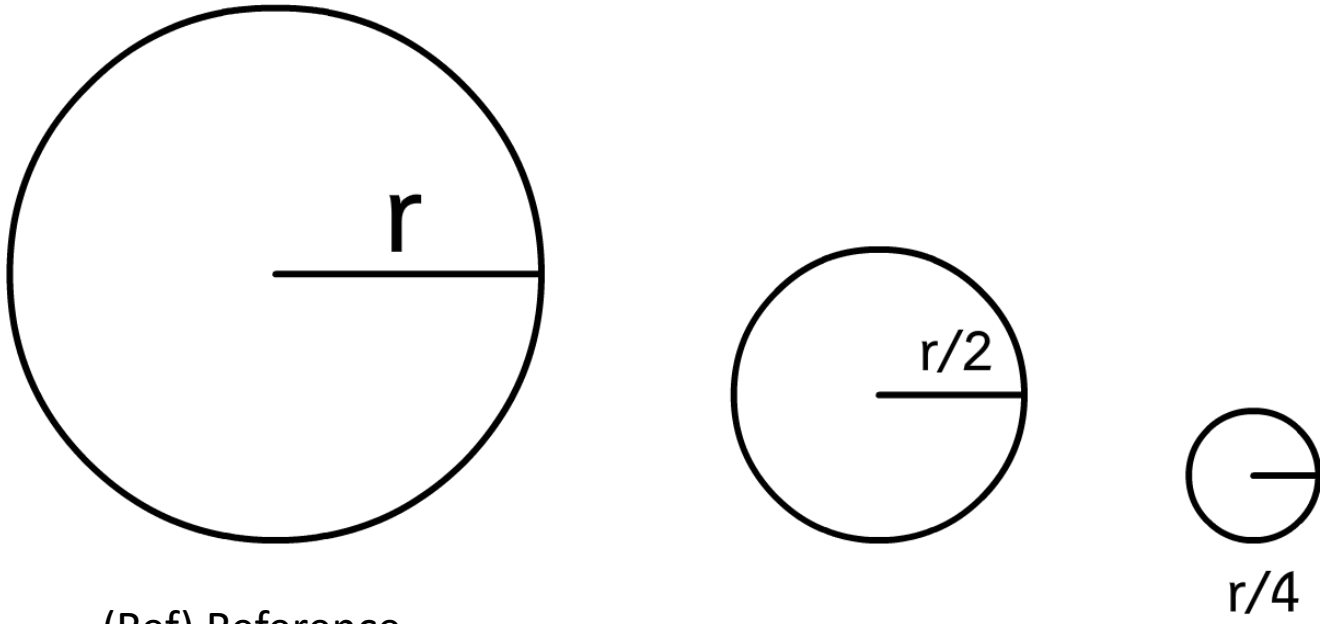




ADC in Microcontroller

- measure Voltage using Vref to reference maximum value

- วัด Voltage โดยใช้ Vref เพื่ออ้างอิงค่าที่มากที่สุด



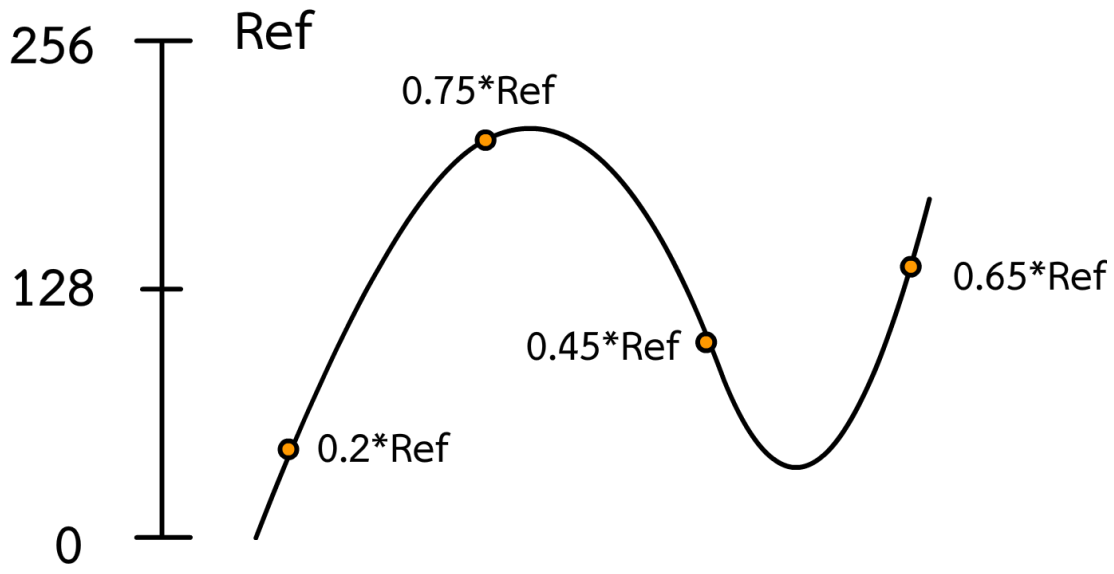
(Ref) Reference

*why not just measure true Voltage
Ans: theoretically impossible

ADC in Microcontroller

- Output measured value is in fraction of Reference format

- ค่าที่ได้ออกมาจะเป็นเศษส่วนที่อ้างอิงกับ Reference



How ADC works?

Varies of method

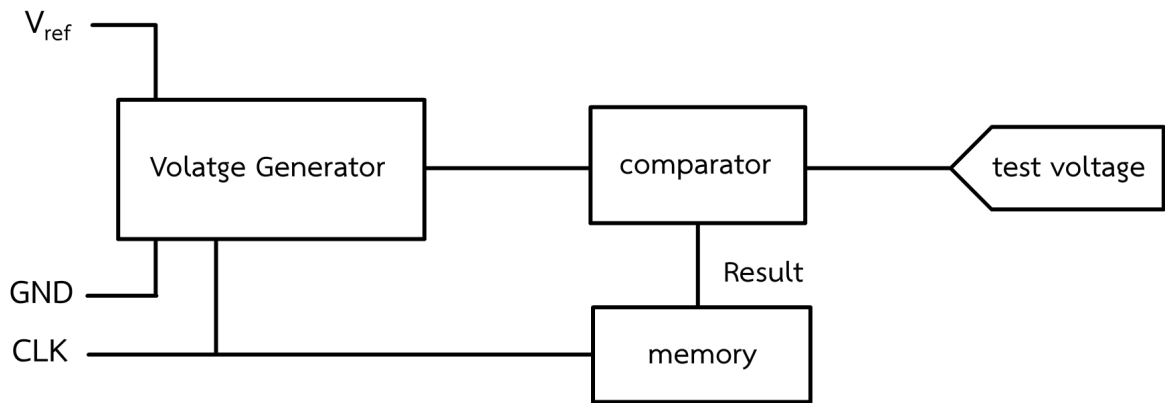
- R to R ladder
- OP-AMP array
- Sigma delta
- SAR: Successive Approximation Register (STM32 and most of MCU)

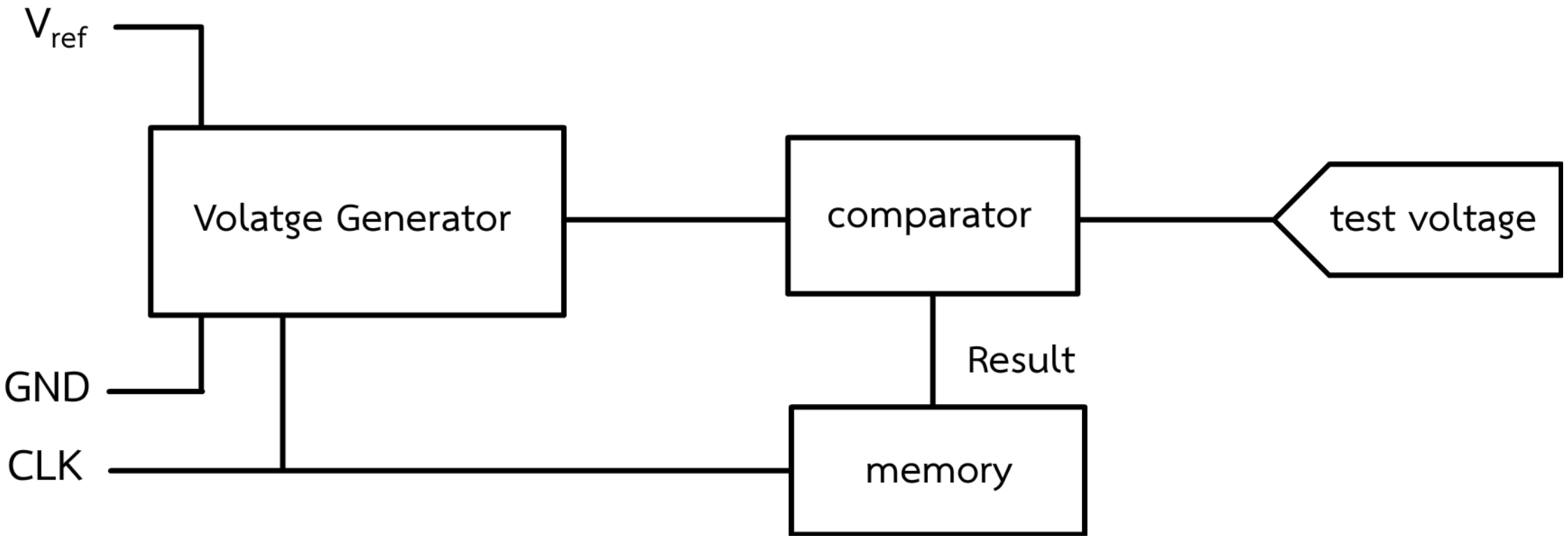
SAR: Successive Approximation Register

low-cost , small-size , medium resolution

ทำงานโดยการเปรียบเทียบสัญญาณทดสอบกับ voltage generator ไปเรื่อย ๆ ตามความละเอียดที่ต้องการ

- voltage generator (ตัวสร้างแรงดัน)
- comparator (ตัวเปรียบเทียบแรงดัน)
- memory (หน่วยความจำ)

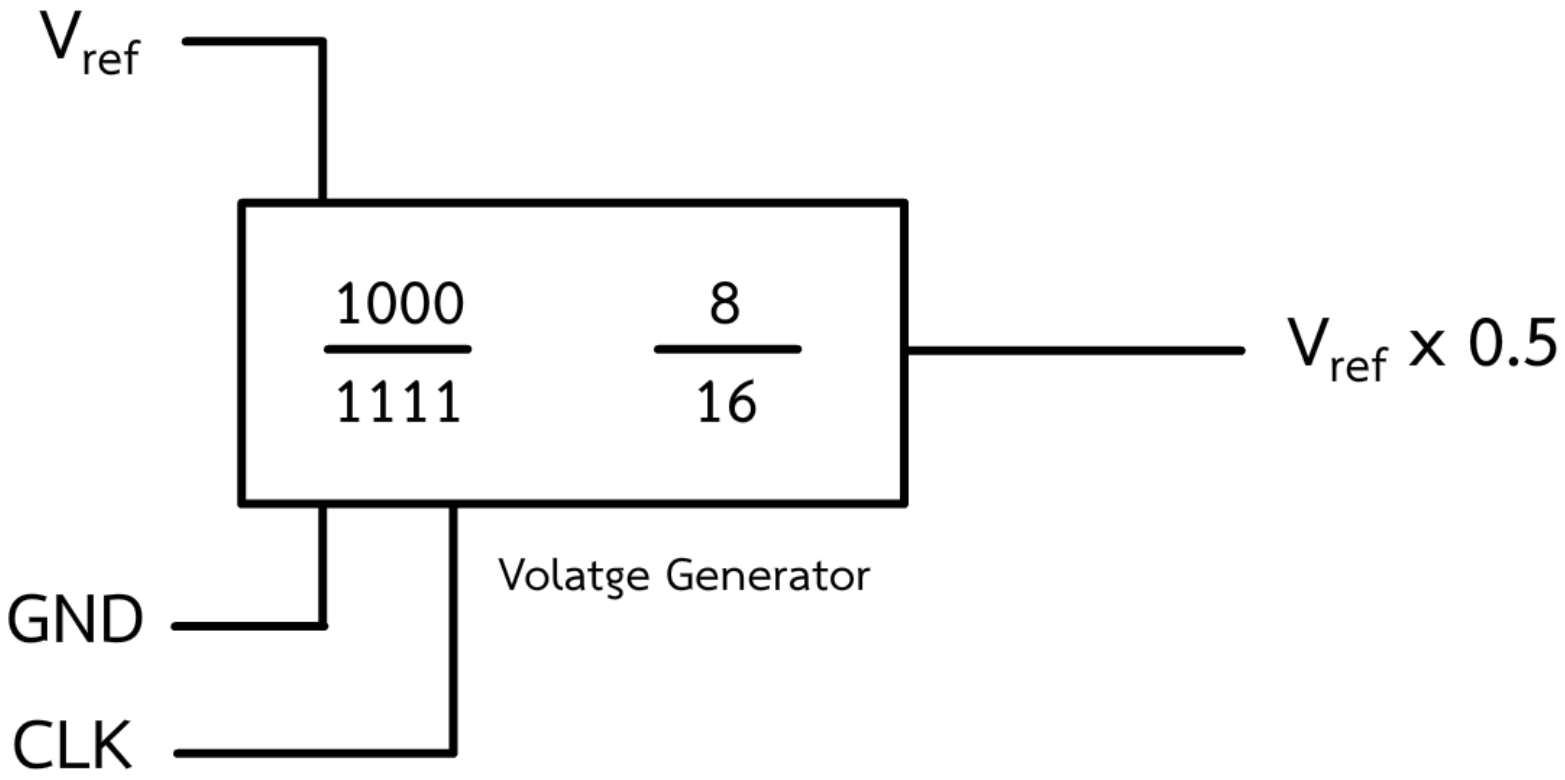


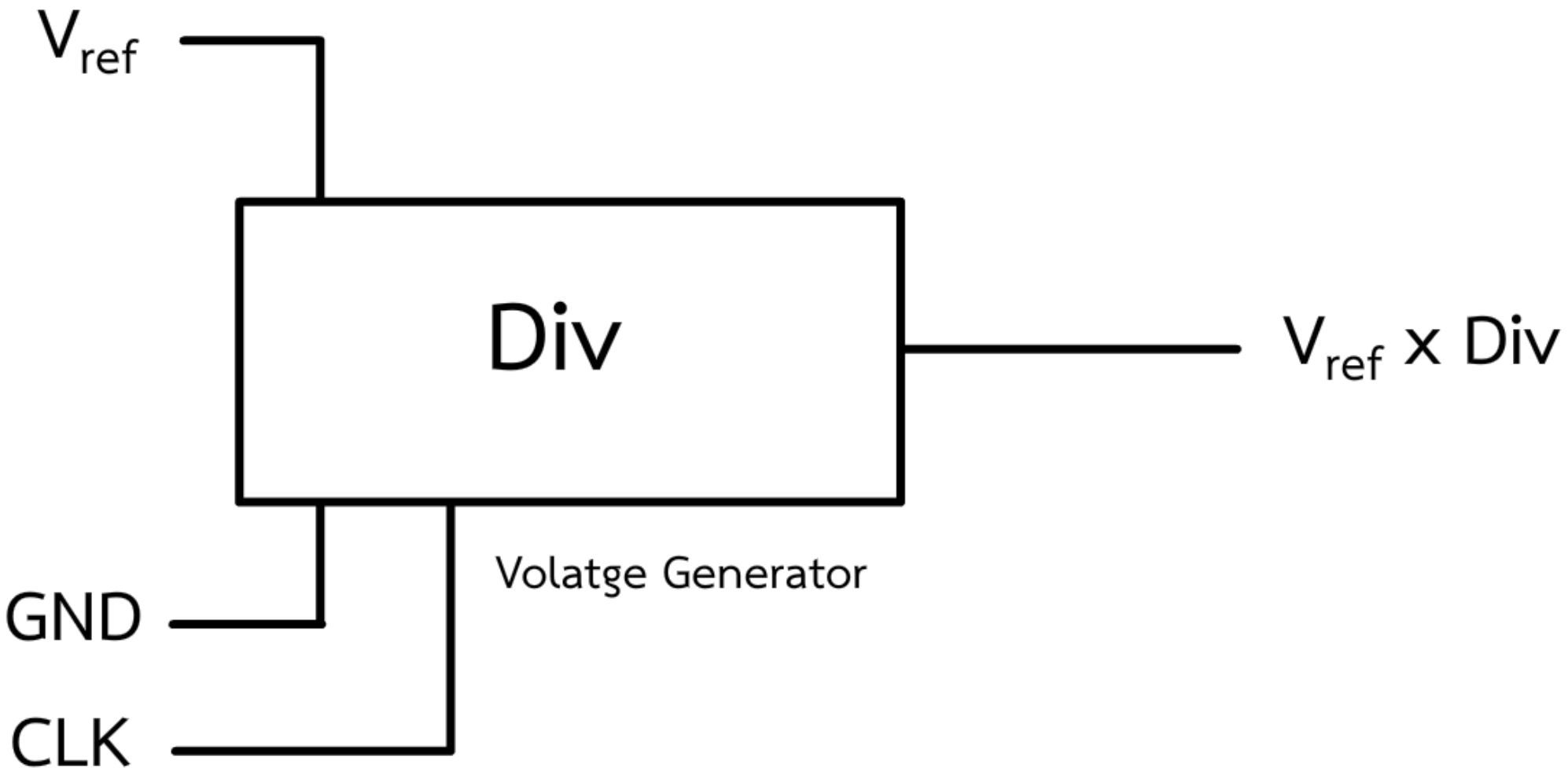

V_{ref}
Volatge Generator
comparator
test voltage
GND
Result
CLK
memory

Voltage generator (DAC)

Control output voltage by divide V_{ref}

ควบคุม V output ด้วยวงจรหาร V_{ref}

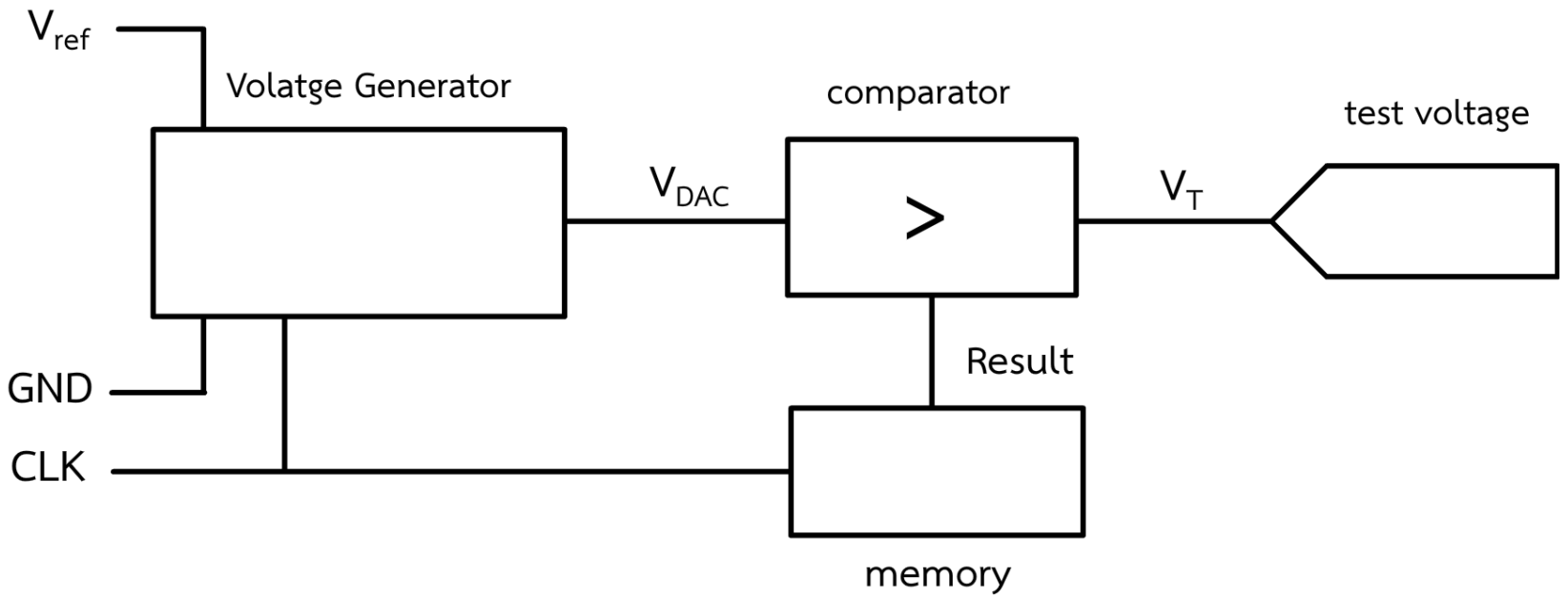



V_{ref}
Div
V_{ref} x Div
Volatge Generator
GND
CLK

Comparator

Compare voltage from Voltage generator (DAC) And test Voltage then store result into memory

ทำการเปรียบเทียบแรงดันจาก Voltage generator (DAC) และ test Voltage จากนั้นจะผลลัพธ์ไว้ใน memory



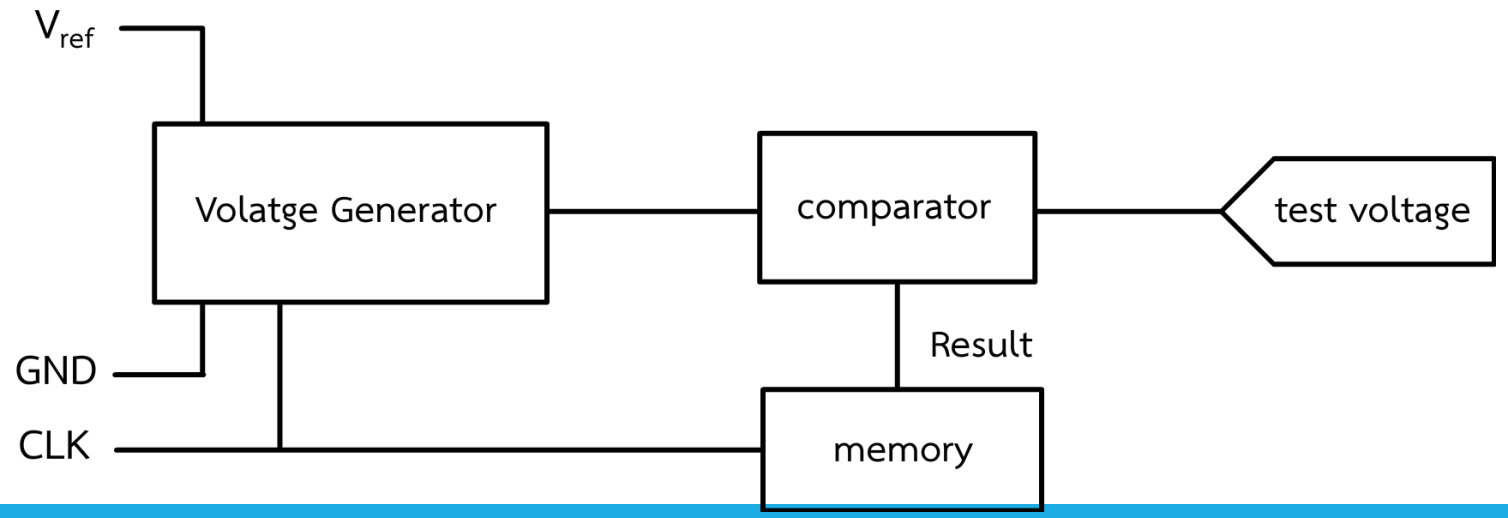
SAR

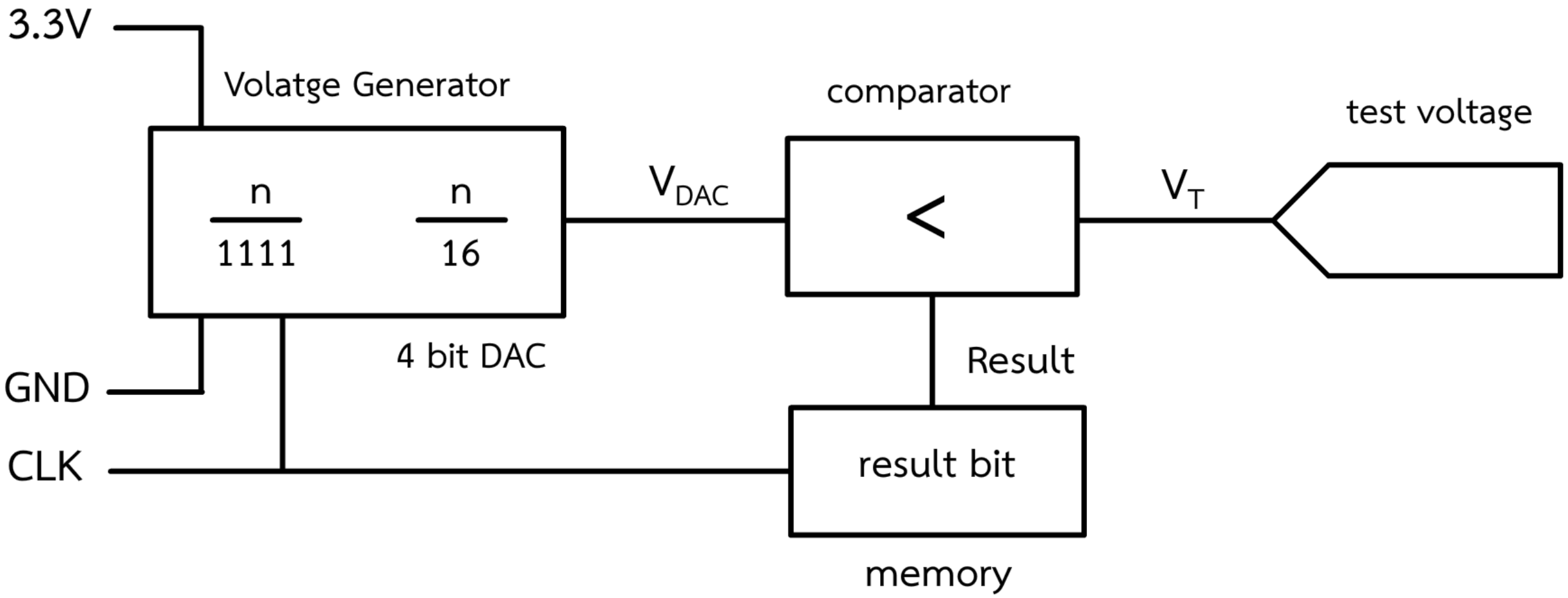
1. set 1 to start bit

2. compare voltage generator with test voltage

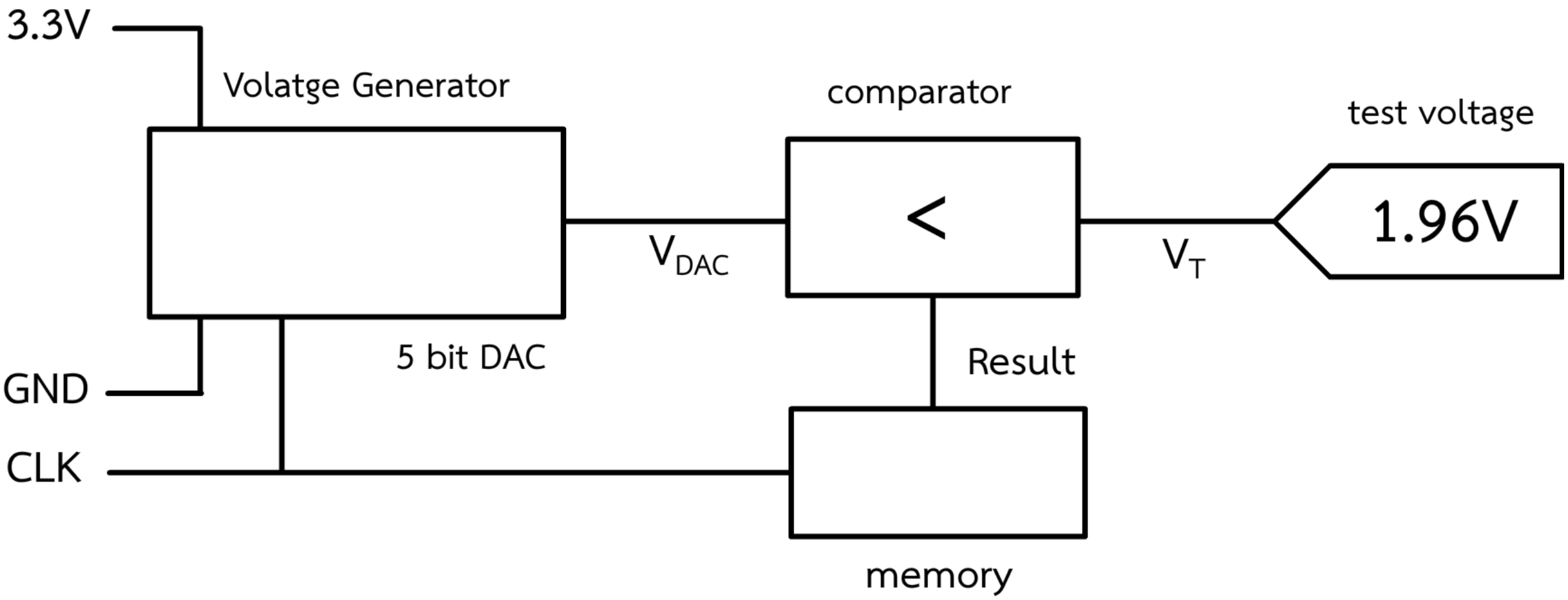
3. store data to memory and set bit to voltage generator

4. repeat until meet resolution (number of bit)




3.3V
Volatge Generator
n
1111
n
16
4 bit DAC
GND
CLK
V_{DAC}
comparator
<
V_T
test voltage
Result
result bit
memory

5 bit ADC


3.3V
Volatge Generator
comparator
test voltage
<
1.96V
V_{DAC}
V_T
5 bit DAC
GND
Result
CLK
memory

ADC Resolution

$$Read\ Voltage = \frac{Value_{ADC}}{MAX_{ADC}} \times V_{Ref};\ MAX_{ADC} = 2^{Resolution}$$

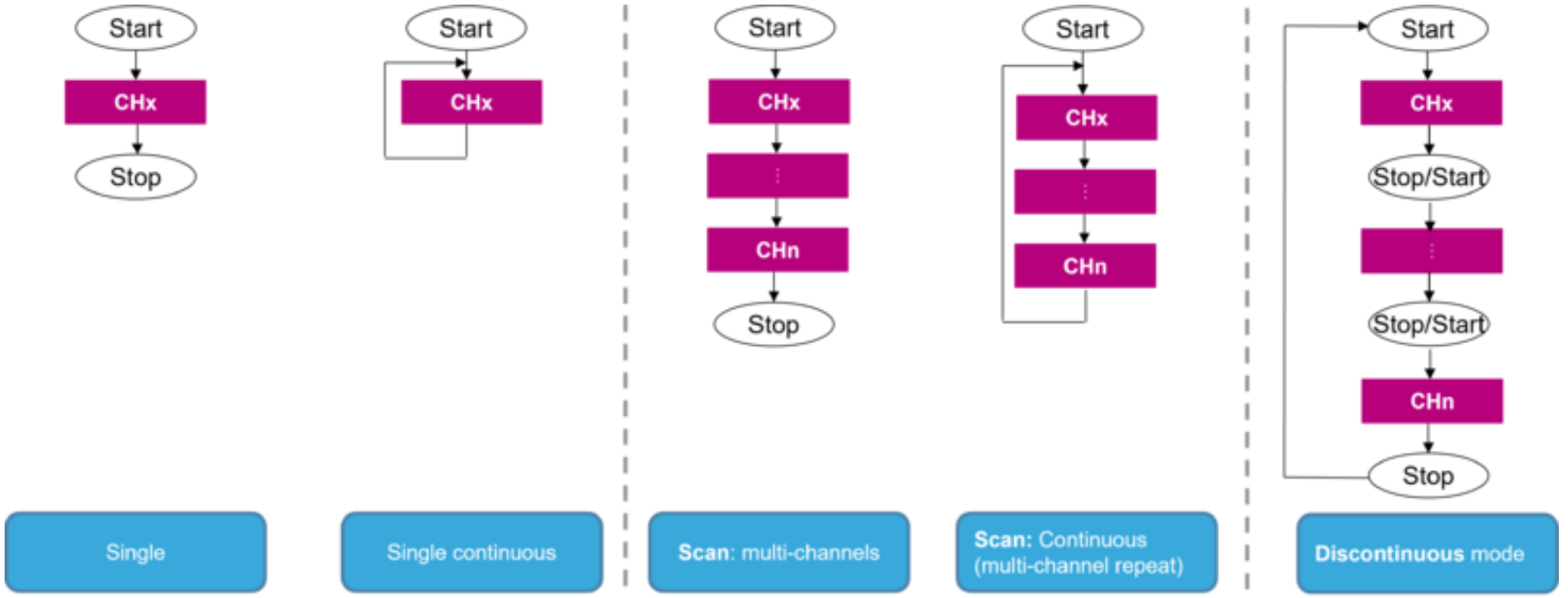
EX1. ADC Value = 203 , Resolution = 9 bit , V_{ref} = 5V what is voltage

Ans = $\frac{203}{2^9} \times 5V = \frac{203}{512} \times 5V = 1.9824V$

EX2. ADC Value = 622 , Resolution = 10 bit , V_{ref} = 3.3V what is voltage

Ans = $\frac{622}{2^{10}} \times 3.3V = \frac{622}{1024} \times 3.3V = 2V$

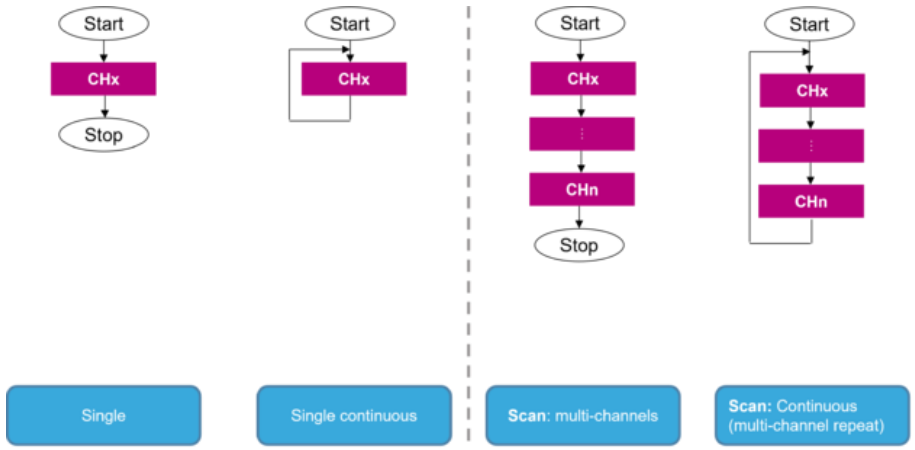
STM32 ADC mode



*https://wiki.st.com/stm32mcu/wiki/Getting_started_with_ADC

STM32 ADC mode

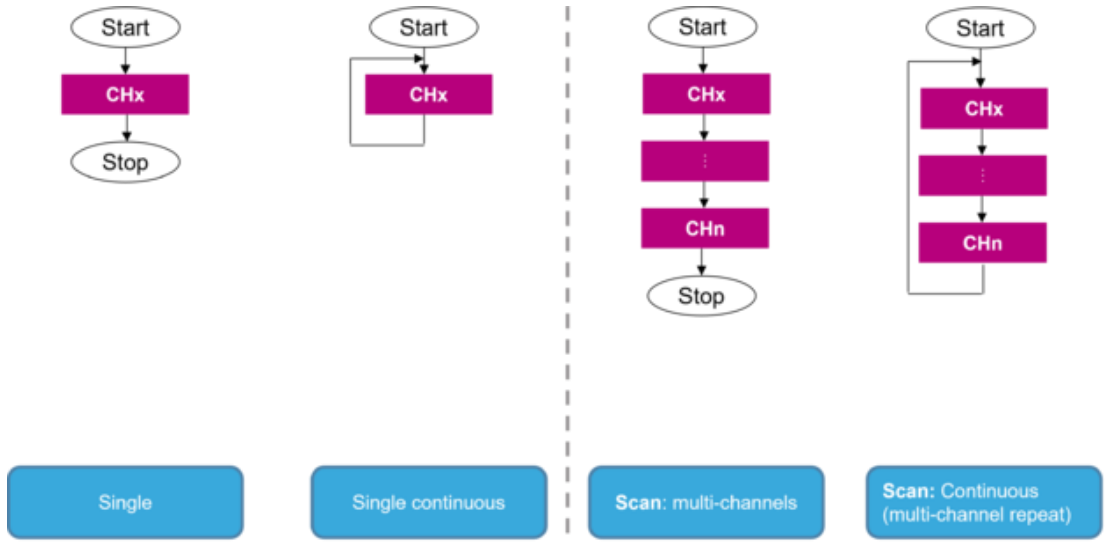
- single and multiple channel mode
  - Indicate that one ADC peripheral connected on multiple PIN or not
  - ระบบุว่า ADC peripheral 1 อันนั้นต่อกับ PIN หลาย PIN หรือไม่
- continuous mode or signal conversion mode
  - Let MCU looping conversion or not
  - ระบุว่าให้ MCU ทำการ convert ไปเรื่อย ๆ หรือไม่



*https://wiki.st.com/stm32mcu/wiki/Getting_started_with_ADC

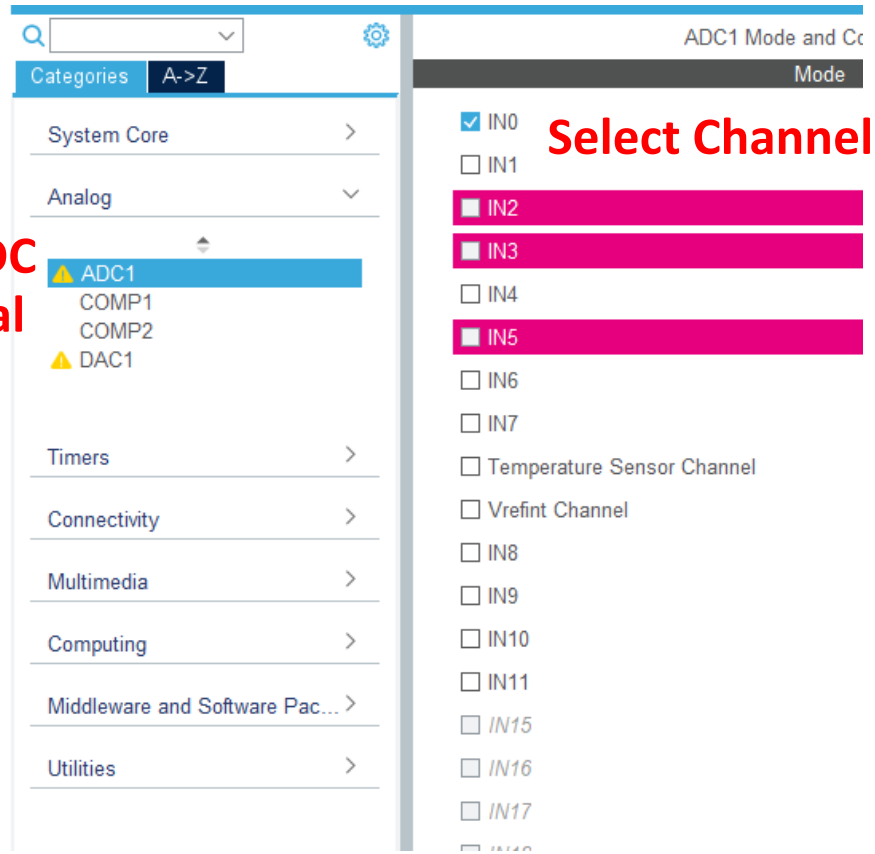
STM32 ADC mode

- single channel single conversion
- single channel continuous conversion
- multiple channel single conversion
- multiple channel continuous conversion

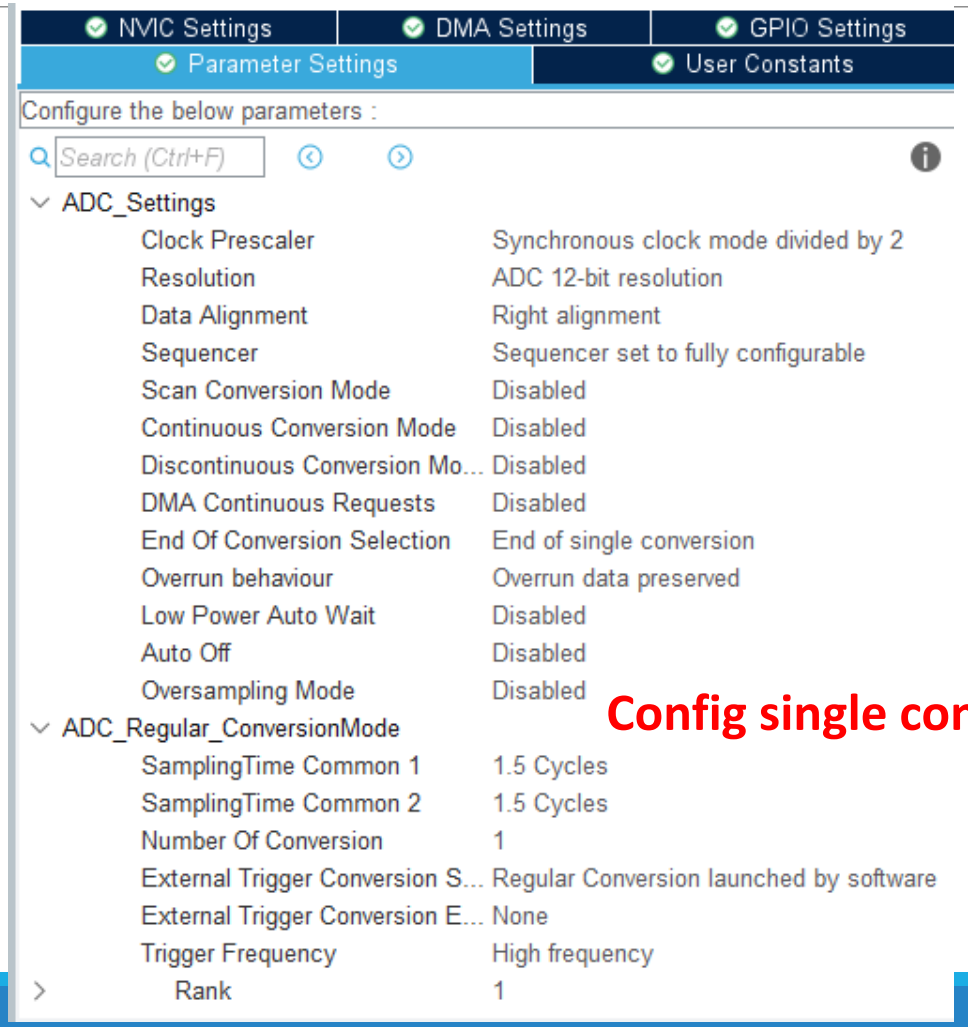


single channel single conversion

Select ADC Peripheral



Select Channel



Config single conversion

coding

```
92   /* Initialize all configured peripherals */
93   MX_GPIO_Init();
94   MX_USART2_UART_Init();
95   MX_ADC1_Init();
96   /* USER CODE BEGIN 2 */
97   HAL_ADCEx_Calibration_Start(&hadc1); // Calibrate ADC
98   HAL_UART_Transmit(&huart2, (uint8_t *)"haruhi\r\n", 8, 1000);
99   /* USER CODE END 2 */
100
101  /* Infinite loop */
102  /* USER CODE BEGIN WHILE */
103  while (1)
104  {
105    uint16_t adc_value = 0;
106    char adc_value_string[20];
107    int adc_value_string_size = 0;
108    HAL_ADC_Start(&hadc1); // start conversion
109    HAL_ADC_PollForConversion(&hadc1, 1000); // wait until conversion finished
110    adc_value = HAL_ADC_GetValue(&hadc1); // get ADC result value
111    HAL_ADC_Stop(&hadc1); // stop conversion
112
113    adc_value_string_size = sprintf(adc_value_string,"%d\r\n",adc_value);
114    HAL_UART_Transmit(&huart2, adc_value_string, adc_value_string_size, 1000);
115
116    HAL_Delay (1000);
117    /* USER CODE END WHILE */
118
119    /* USER CODE BEGIN 3 */
120  }
121  /* USER CODE END 3 */
122 }
```

```
/* USER CODE BEGIN 2 */
HAL_ADCEx_Calibration_Start(&hadc1);
HAL_UART_Transmit(&huart2, (uint8_t *)"haruhi\r\n", 8, 1000);
/* USER CODE END 2 */
```

```
/* USER CODE BEGIN WHILE */
while (1)
{
uint16_t adc_value = 0;
char adc_value_string[20];
int adc_value_string_size = 0;
HAL_ADC_Start(&hadc1); // start conversion
HAL_ADC_PollForConversion(&hadc1, 1000); // wait until conversion finished
adc_value = HAL_ADC_GetValue(&hadc1); // get ADC result value
HAL_ADC_Stop(&hadc1); // stop conversion


adc_value_string_size = sprintf(adc_value_string,"%d\r\n",adc_value);
HAL_UART_Transmit(&huart2, adc_value_string, adc_value_string_size, 1000);


HAL_Delay (1000);
/* USER CODE END WHILE */
```

Wiring

โปรดใช้ความระมัดระวังก่อนจ่ายไฟ!!!!
จ่ายไฟเกิน (V) board ไหม้นะครับ

ระวังกันมากๆ นะครับ ตอนปรับ V

Power Supply

0.1 A limit !!!

0 – 3.3 V !!!

\-

\+

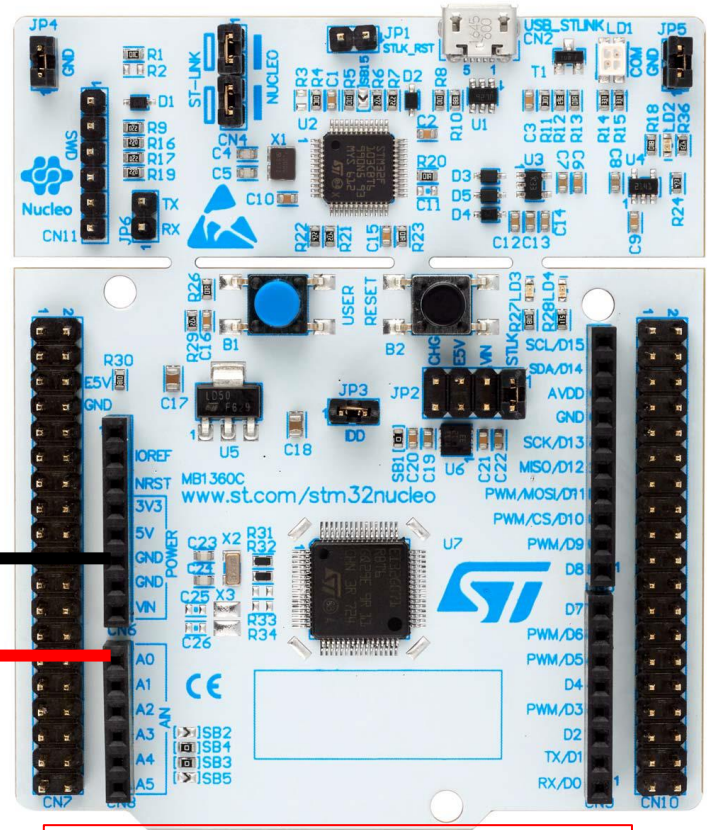
ระวังมากๆๆๆๆๆๆๆ นะครับ

ห้ามต่อกลับขั้วนะครับ

เช็ค PIN ให้ดีๆ นะครับ

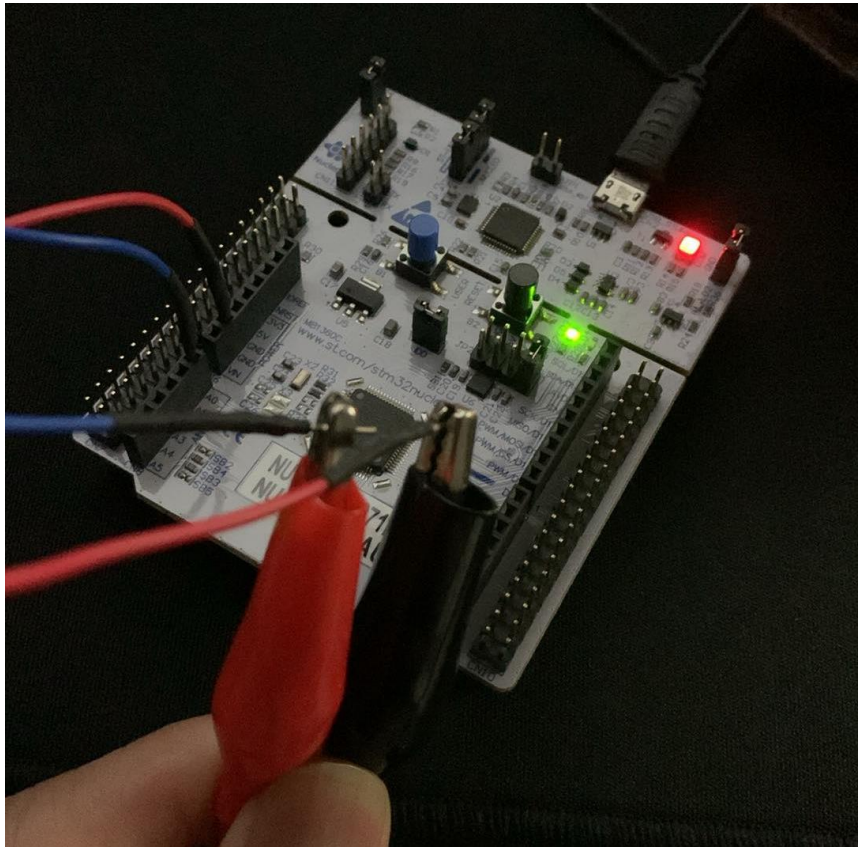
GND

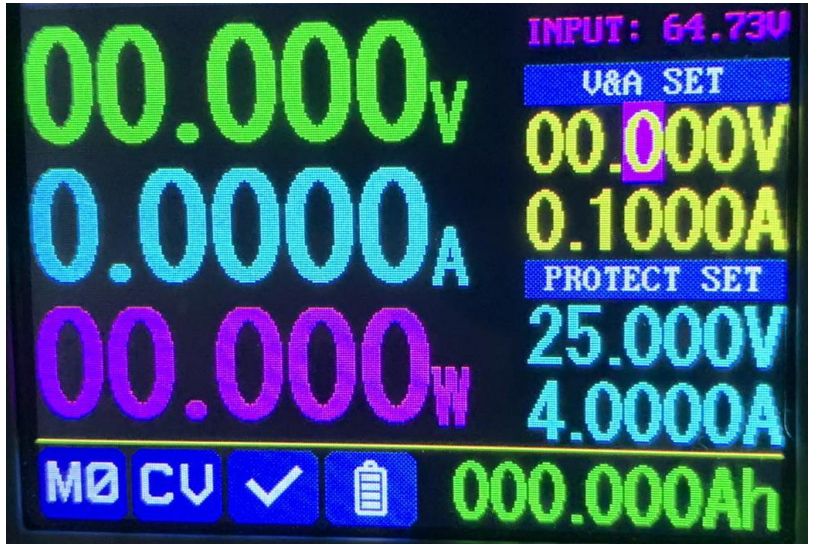
A0(Arduino pin) for A0 channel



บอร์ดพังแล้วพังเลยนะครับ

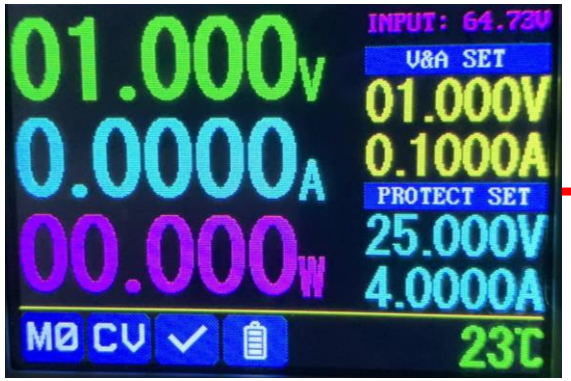
Setup





Limit Current

STM32 ADC Result



haruhi
0
13
15
366
988
1242
1252
1258
1242
1313
2970
2954
2963
2966
2530
2978

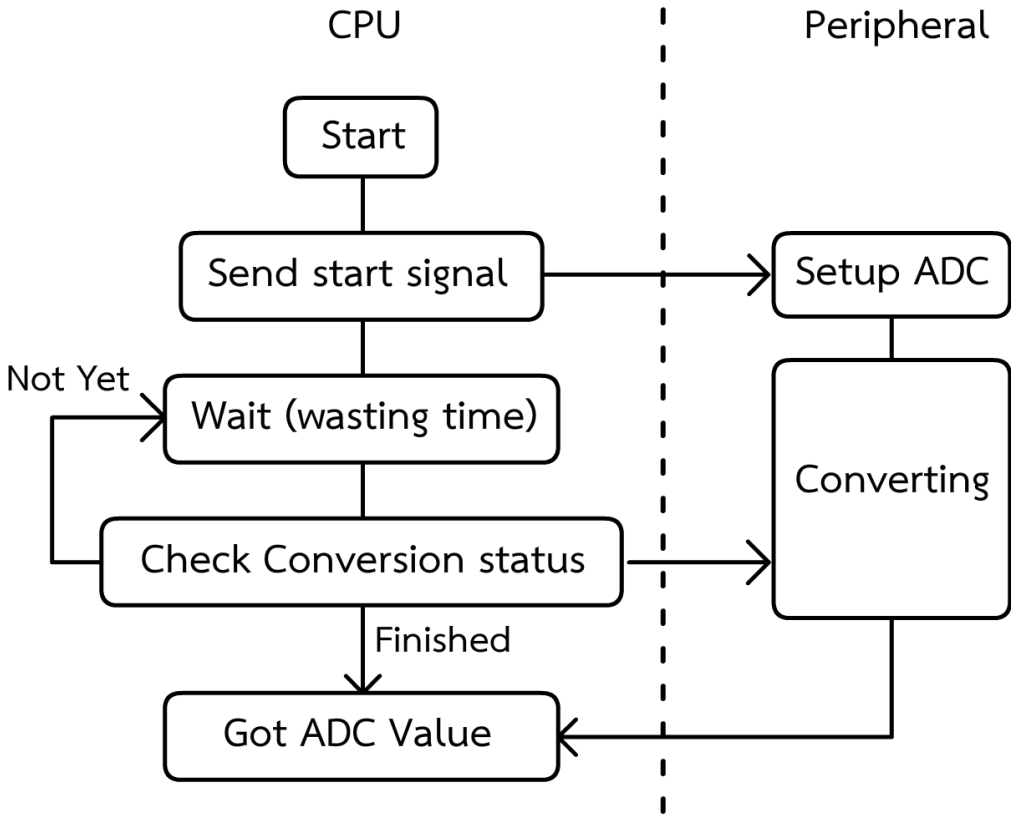
Restart

0V

$$3.3 \times \frac{1242}{4096} = 1.0006347V$$

$$3.3 \times \frac{2945}{4096} = 2.3799316V$$

Single Conversion mode

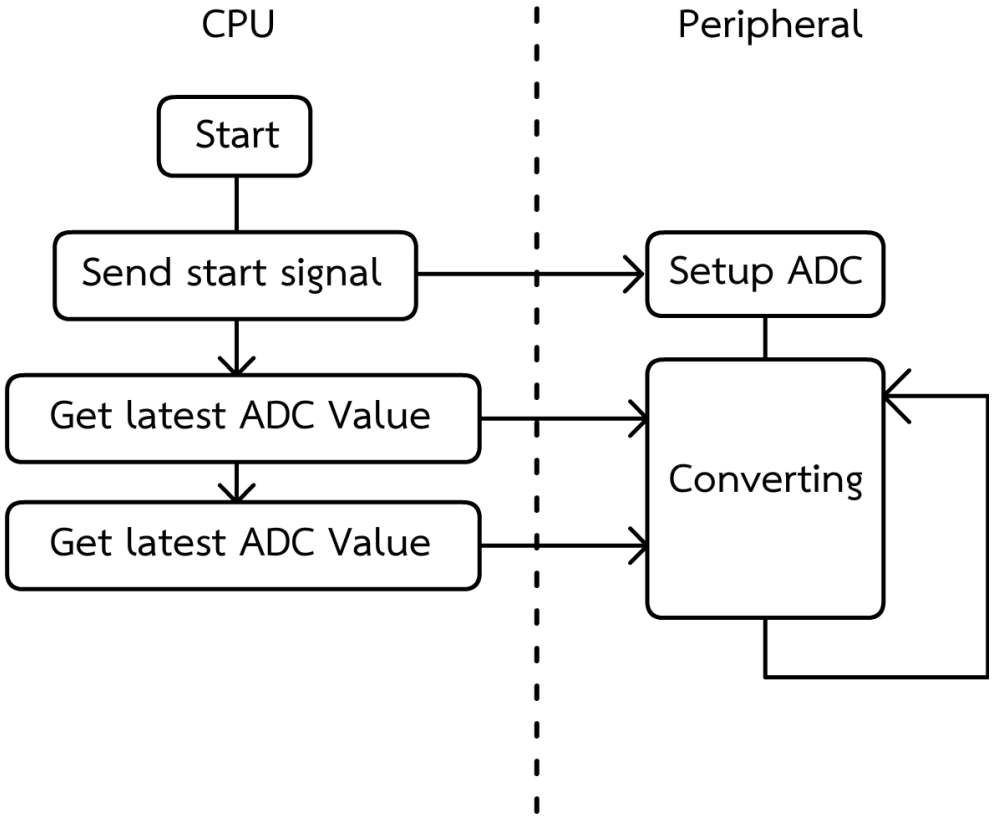


1. Send start signal to ADC peripheral
2. Polling for finish conversion status (waste of time)
3. Get ADC value

Continuous Conversion mode

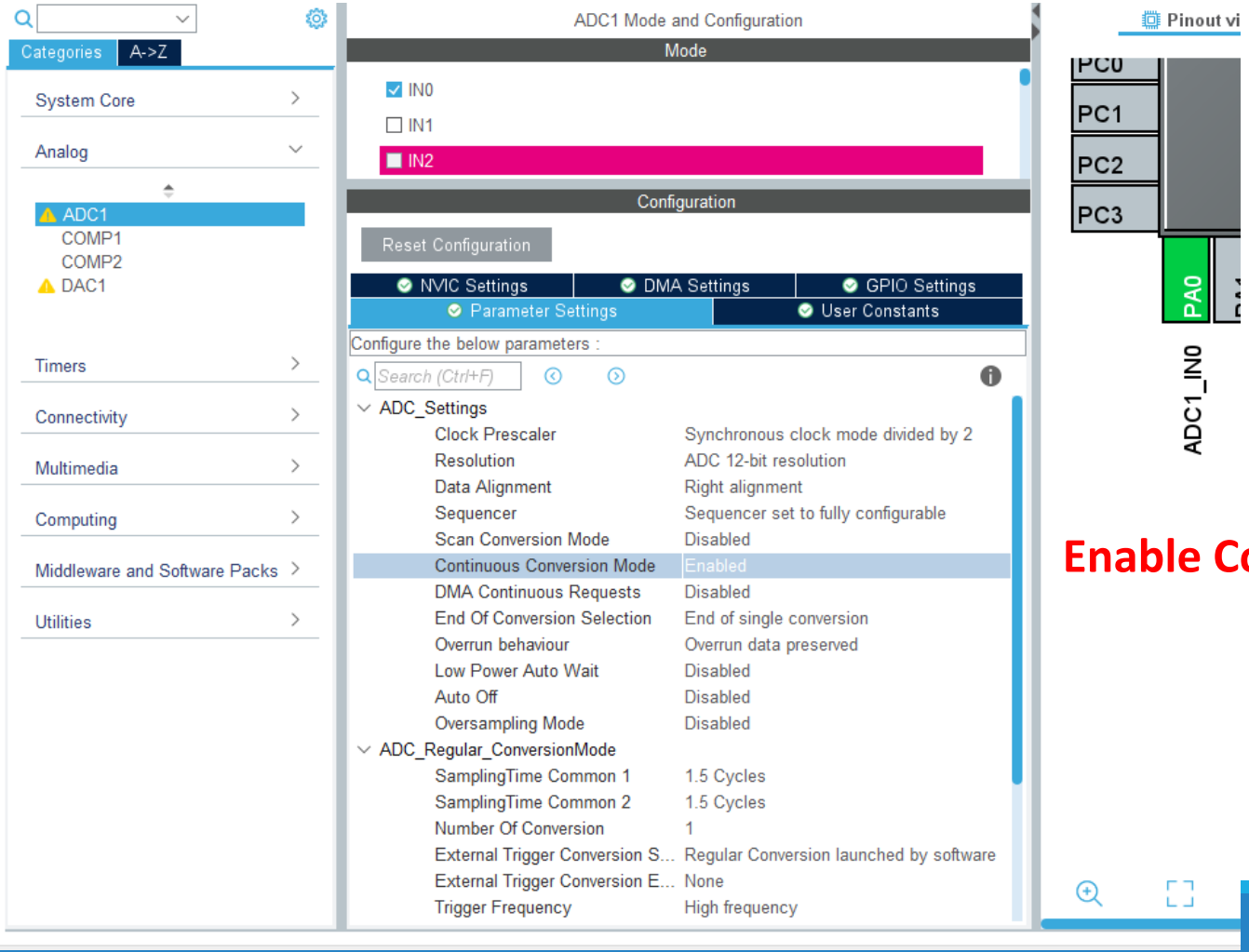
1. Send start signal to ADC peripheral
2. Get lasted data

No waiting time!!
ADC continue convert value for CPU
Consume more energy



Continuous Conversion mode setup

Select ADC Peripheral



Enable Continuous Mode

coding

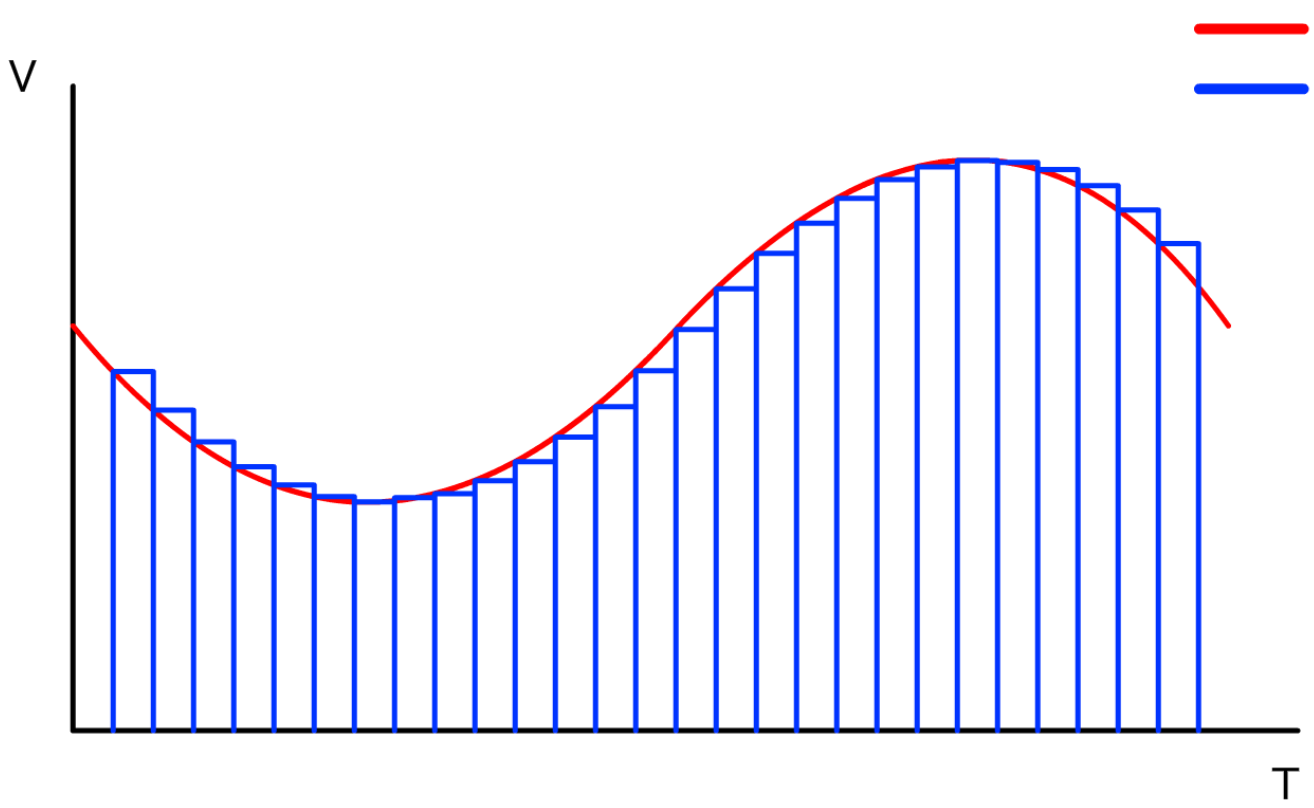
Same Result?

```
 96  /* USER CODE BEGIN 2 */
 97  HAL_ADCEx_Calibration_Start(&hadc1); // Calibrate ADC
 98  HAL_UART_Transmit(&huart2, (uint8_t *)"haruhi2\r\n", 9, 1000);
 99  HAL_ADC_Start(&hadc1); // start conversion
100  /* USER CODE END 2 */
101
102  /* Infinite loop */
103  /* USER CODE BEGIN WHILE */
104  while (1)
105  {
106    uint16_t adc_value = 0;
107    char adc_value_string[20];
108    int adc_value_string_size = 0;
109
110    adc_value = HAL_ADC_GetValue(&hadc1); // get ADC result value
111
112    adc_value_string_size = sprintf(adc_value_string,"%d\r\n",adc_value);
113    HAL_UART_Transmit(&huart2, adc_value_string, adc_value_string_size, 1000);
114
115    HAL_Delay (1000);
116    /* USER CODE END WHILE */
117
118    /* USER CODE BEGIN 3 */
119  }
```

```
/* USER CODE BEGIN 2 */
HAL_ADCEx_Calibration_Start(&hadc1); // Calibrate ADC
HAL_UART_Transmit(&huart2, (uint8_t *)"haruhi2\r\n", 9, 1000);
HAL_ADC_Start(&hadc1); // start conversion
/* USER CODE END 2 */
```

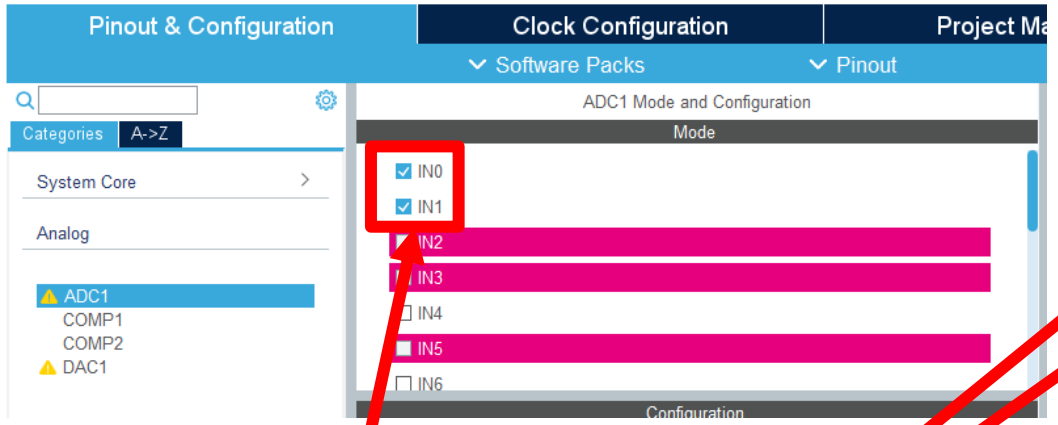
```
/* USER CODE BEGIN WHILE */
while (1)
{
uint16_t adc_value = 0;
char adc_value_string[20];
int adc_value_string_size = 0;


adc_value = HAL_ADC_GetValue(&hadc1); // get ADC result value


adc_value_string_size = sprintf(adc_value_string,"%d\r\n",adc_value);
HAL_UART_Transmit(&huart2, adc_value_string, adc_value_string_size, 1000);


HAL_Delay (1000);
/* USER CODE END WHILE */
```

Measure sinewave



Output are in digital format (zero-hold)

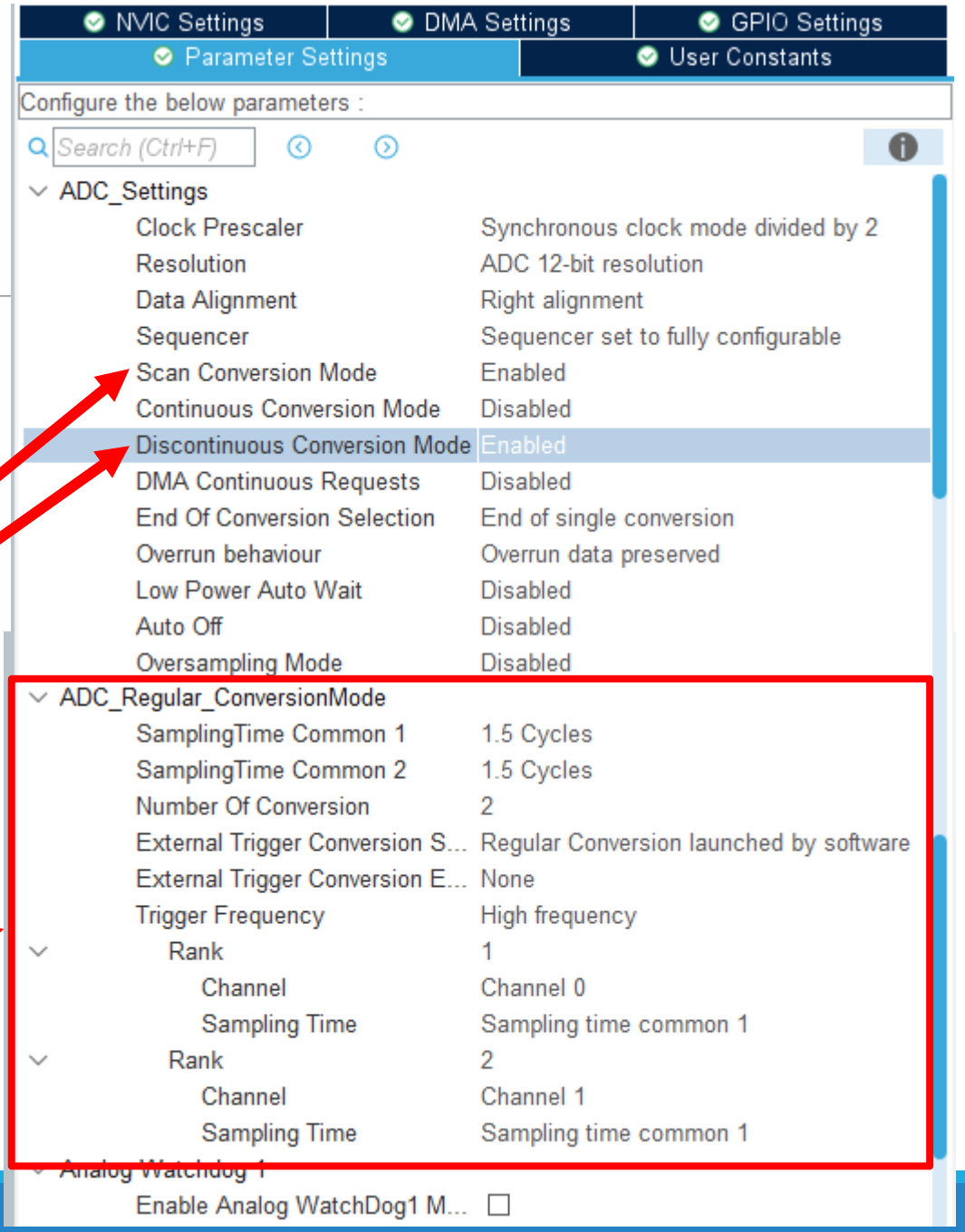
Multi Channel



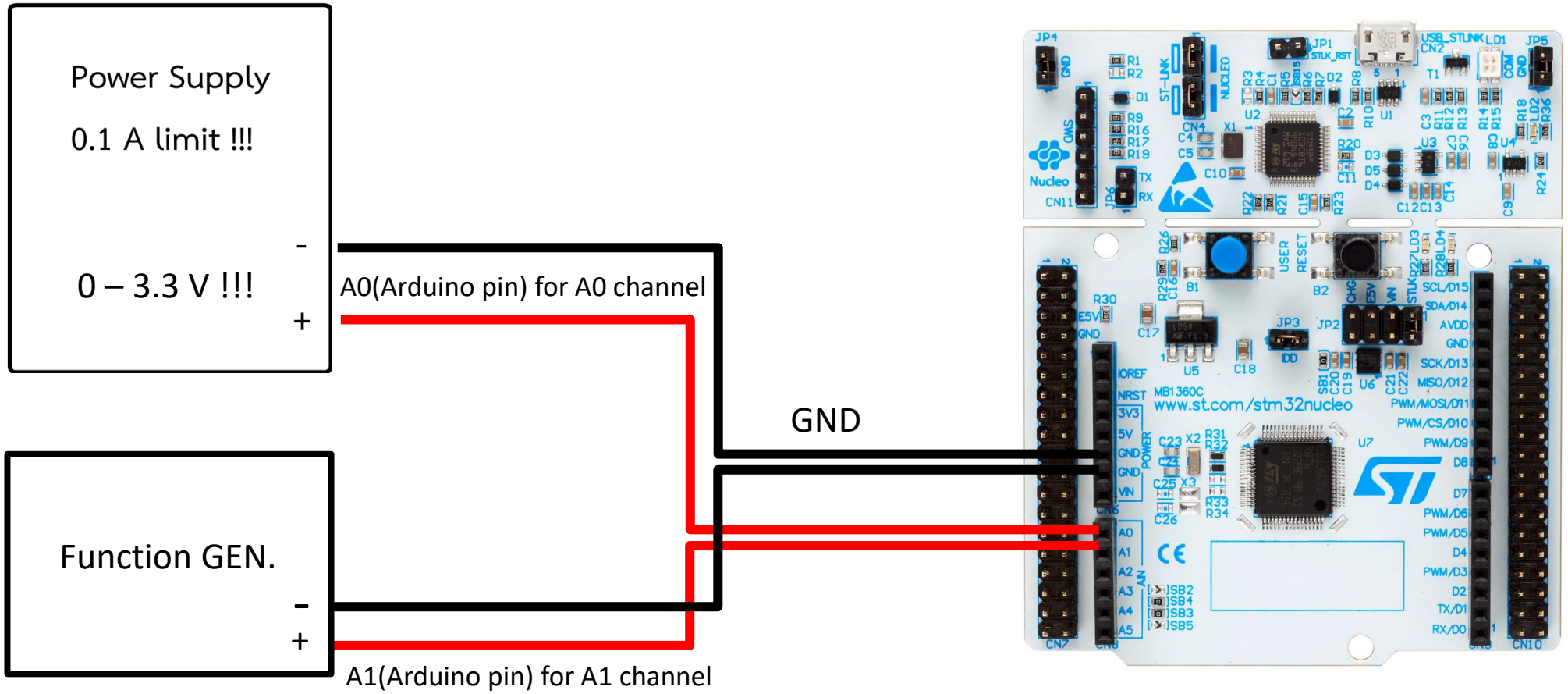
Select Channel

Select Scan mode

Setup Channel



Wiring


Power Supply
0.1 A limit !!!
0 – 3.3 V !!!
-
+
A0(Arduino pin) for A0 channel
GND
Function GEN.
-
+
A1(Arduino pin) for A1 channel
MB1360C
www.st.com/stm32nucleo

coding

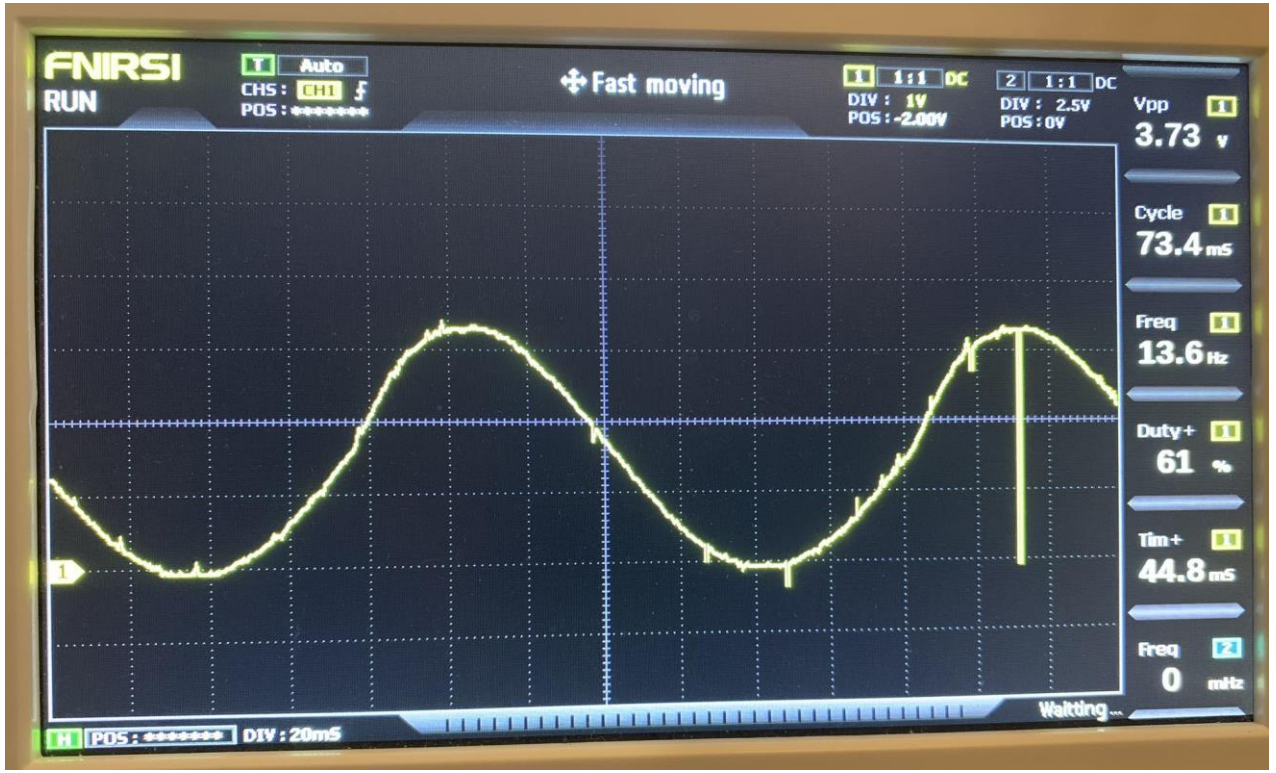
```
96  /* USER CODE BEGIN 2 */
97  HAL_ADCEx_Calibration_Start(&hadc1); // Calibrate ADC
98  HAL_UART_Transmit(&huart2, (uint8_t *)"haruhi3\r\n", 9, 1000);
99  /* USER CODE END 2 */
100
101 /* Infinite loop */
102 /* USER CODE BEGIN WHILE */
103 while (1)
104 {
105   uint16_t adc_value[0] = {0,0};
106   char adc_value_string[30];
107   int adc_value_string_size = 0;
108   HAL_ADC_Start(&hadc1); // start conversion (CH0)
109   HAL_ADC_PollForConversion(&hadc1, 1000); // wait for conversion
110   adc_value[0] = HAL_ADC_GetValue(&hadc1); // get ADC CH0 value
111   HAL_ADC_Start(&hadc1); // start conversion (CH1)
112   HAL_ADC_PollForConversion(&hadc1, 1000); // wait for conversion
113   adc_value[1] = HAL_ADC_GetValue(&hadc1); // get ADC CH1 value
114   HAL_ADC_Stop(&hadc1); // Reset Conversion
115
116   adc_value_string_size = sprintf(adc_value_string,"%d,%d\r\n",adc_value[0],adc_value[1]);
117   HAL_UART_Transmit(&huart2, adc_value_string, adc_value_string_size, 1000);
118
119   HAL_Delay (1000);
120   /* USER CODE END WHILE */
```

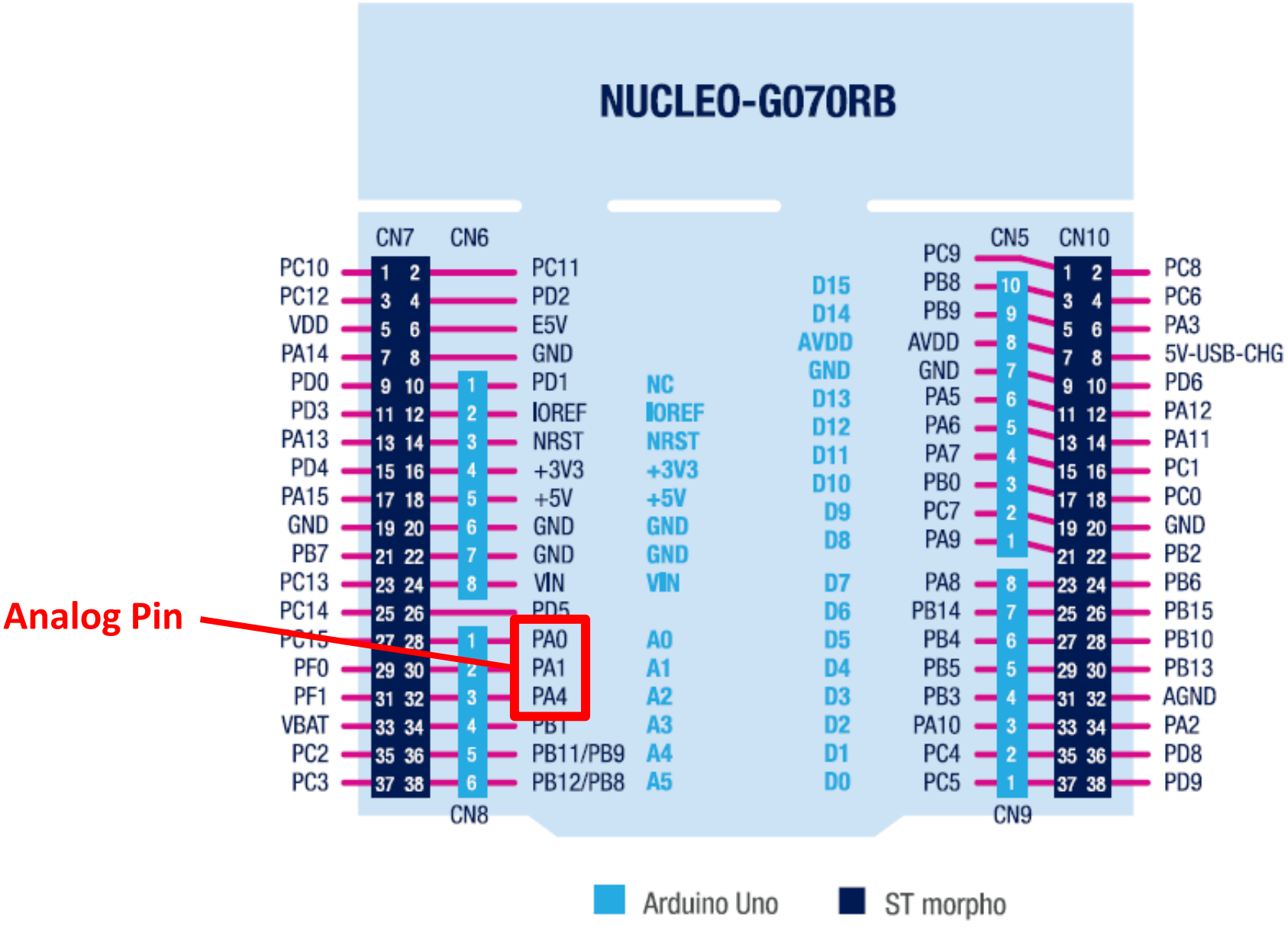
```
/* USER CODE BEGIN 2 */
HAL_ADCEx_Calibration_Start(&hadc1); // Calibrate ADC
HAL_UART_Transmit(&huart2, (uint8_t *)"haruhi3\r\n", 9, 1000);
/* USER CODE END 2 */
```

```
/* USER CODE BEGIN WHILE */
while (1)
{
uint16_t adc_value[0] = {0,0};
char adc_value_string[30];
int adc_value_string_size = 0;
HAL_ADC_Start(&hadc1); // start conversion (CH0)
HAL_ADC_PollForConversion(&hadc1, 1000); // wait for conversion
adc_value[0] = HAL_ADC_GetValue(&hadc1); // get ADC CH0 value
HAL_ADC_Start(&hadc1); // start conversion (CH1)
HAL_ADC_PollForConversion(&hadc1, 1000); // wait for conversion
adc_value[1] = HAL_ADC_GetValue(&hadc1); // get ADC CH1 value
HAL_ADC_Stop(&hadc1); // Reset Conversion


adc_value_string_size =
sprintf(adc_value_string,"%d,%d\r\n",adc_value[0],adc_value[1]);
HAL_UART_Transmit(&huart2, adc_value_string, adc_value_string_size, 1000);


HAL_Delay (1000);
/* USER CODE END WHILE */
```

What about DAC?

- Digital to analog converter
- Generate voltage signal
- Generate analog waveform




NUCLEO-G070RB
CN7
CN6
CN5
CN10
CN8
CN9
PC10
PC12
VDD
PA14
PD0
PD3
PA13
PD4
PA15
GND
PB7
PC13
PC14
PF0
PF1
VBAT
PC2
PC3
PC11
PD2
E5V
GND
PD1
IOREF
NRST
+3V3
+5V
GND
GND
VIN
PD5
PA0
PA1
PA4
PB1
PB11/PB9
PB12/PB8
NC
IOREF
NRST
+3V3
+5V
GND
GND
VIN
A0
A1
A2
A3
A4
A5
D15
D14
AVDD
GND
D13
D12
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
PC9
PB8
PB9
AVDD
GND
PA5
PA6
PA7
PB0
PC7
PA9
PA8
PB14
PB4
PB5
PB3
PA10
PC4
PC5
PC8
PC6
PA3
5V-USB-CHG
PD6
PA12
PA11
PC1
PC0
GND
PB2
PB6
PB15
PB10
PB13
AGND
PA2
PD8
PD9
Arduino Uno
ST morpho

Analog Pin

LAB