

Hệ nhúng (Embedded Systems)

IT4210

Đỗ Công Thuần

Khoa Kỹ thuật máy tính, Trường CNTT&TT

Đại học Bách khoa Hà Nội

Email: thuandc@soict.hust.edu.vn

ONE LOVE. ONE FUTURE.

Giới thiệu môn học

- Tên học phần: **Hệ nhúng**
- Mã học phần: **IT4210 (3-0-1-6)**
- Thời lượng:
 - 16.5 buổi lý thuyết (3 tiết/buổi)
 - 3 buổi thực hành (5 tiết/buổi)
- Yêu cầu kiến thức nền tảng:
 - Kiến trúc máy tính
 - Vi xử lý
 - Lập trình C

Mục tiêu môn học

- Hiểu được kiến trúc tổng quan, đặc điểm và hoạt động của một hệ nhúng
- Biết thiết kế hệ nhúng cơ bản (nguyên lý thiết kế mạch, ...)
- Hiểu được kiến trúc vi điều khiển (Intel, ARM)
- Lập trình vi điều khiển từ cơ bản đến nâng cao với các dòng vi điều khiển phổ biến
- Lập trình với hệ điều hành nhúng

Đánh giá học phần

1. Đánh giá quá trình: **40%**

- Bài tập về nhà
- Chuyên cần
- Các bài thực hành, nhóm **4 SV/nhóm**

2. Đánh giá cuối kỳ: **60%**

- Làm project cuối kỳ, nhóm **4 SV/nhóm**
- Yêu cầu sinh viên tự chọn nhóm và đăng kí đề tài.
Chú ý: danh sách đề tài sẽ được cập nhật sau!

Tài liệu tham khảo

- Textbook/Lecture notes:

- Peter Marwedel, ***Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems, and the Internet of Things***, Springer, 4th edition, 2021.
- Edward A. Lee and Sanjit A. Seshia, ***Introduction to Embedded Systems: A Cyber-Physical Systems Approach***, MIT Press, 2nd edition, 2017.
- Tammy Noergaard, ***Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers***, Elsevier, 2nd edition, 2013.
- Han-Way Huang, Leo Chartrand, ***Microcontroller: An Introduction to Software & Hardware Interfacing***, Cengage Learning, 2004.
- Lectures in Embedded Systems from *Univ. of Cincinnati (EECE 6017C)*, *Univ. of California, Berkeley (EECS 149)*, *Univ. of Pennsylvania (ESE 350)*, *Univ. of Kansas (EECS388)*.
- ...

- Manuals/Handbooks/Internet

- Atmel, Microchip, Texas Instruments, Keil...
- Keil ASM51
- Arduino IDE
- ...

Nội dung học phần

- Chương 1: Giới thiệu về Hệ nhúng
- Chương 2: Thiết kế phần cứng Hệ nhúng
- Chương 3: Lập trình với 8051
- Chương 4: Ghép nối ngoại vi với 8051
- Chương 5: Arduino
- Chương 6: Ghép nối nối tiếp
- Chương 7: Ghép nối với thế giới thực
- Chương 8: Kiến trúc ARM
- Chương 9: RTOS và FreeRTOS

Nội dung học phần

- Chương 1: Giới thiệu về Hệ nhúng
- Chương 2: Thiết kế phần cứng Hệ nhúng
- Chương 3: Lập trình với 8051
- Chương 4: Ghép nối ngoại vi với 8051
- Chương 5: Arduino
- Chương 6: Ghép nối nối tiếp
- **Chương 7: Ghép nối với thế giới thực**
- Chương 8: Kiến trúc ARM
- Chương 9: RTOS và FreeRTOS

Chương 7

Ghép nối với thế giới thực

Nội dung

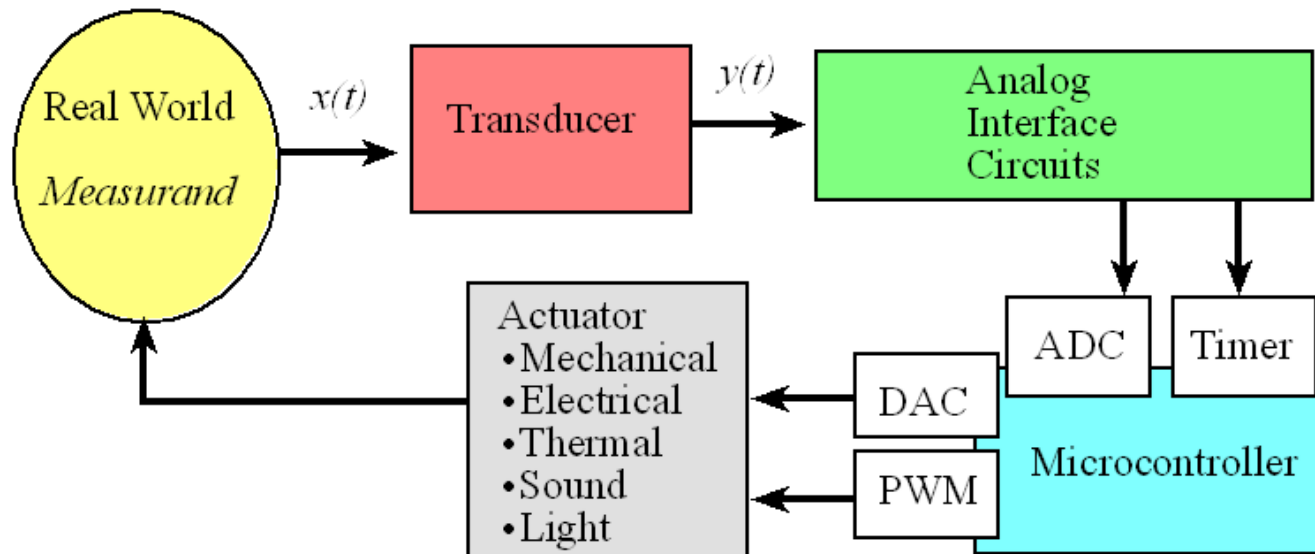
- Giới thiệu
- ADC/DAC
- Ghép nối ADC
- Ghép nối thiết bị công suất cao

Nội dung

- Giới thiệu
- ADC/DAC
- Ghép nối ADC
- Ghép nối thiết bị công suất cao

Giới thiệu

- Hệ nhúng là một thành phần quan trọng của hệ thống đo lường, điều khiển số
 - Đầu vào: các thông tin về đối tượng (nhiệt độ, độ ẩm, áp suất, độ pH, độ mặn,...) tùy thuộc bài toán.
 - Đầu ra: cơ cấu chấp hành tác động đến đối tượng.



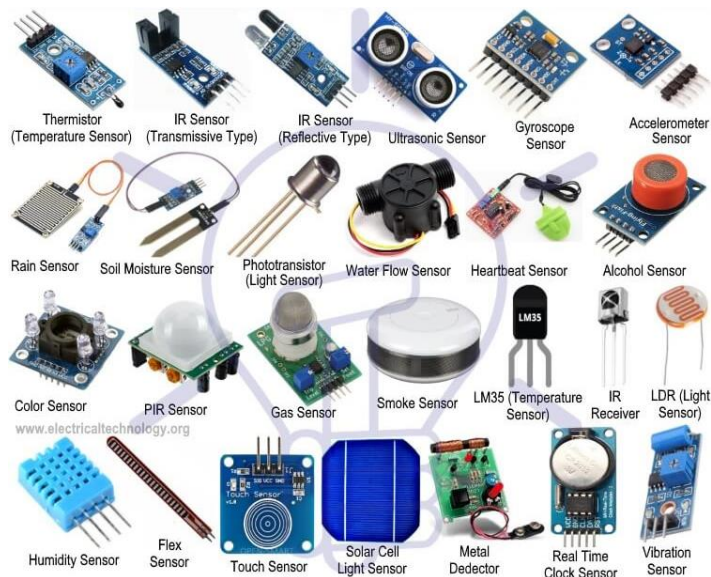
Giới thiệu

- Ví dụ: máy ấp trứng tự động
- Các tính năng
 - Đo nhiệt độ, độ ẩm
 - Bật/tắt đèn sưởi, quạt, phun sương
 - Tự động duy trì nhiệt độ, độ ẩm
 - Tự đảo trứng
- Nguyên tắc làm việc?



Cảm biến

- Là thiết bị phát hiện sự thay đổi của đại lượng vật lý, rồi gửi thay đổi đó về máy tính dưới dạng tín hiệu đọc được
- Thiết bị biến đổi đại lượng vật lý cần đo thành tín hiệu điện, cho phép máy tính đo lường đại lượng đó



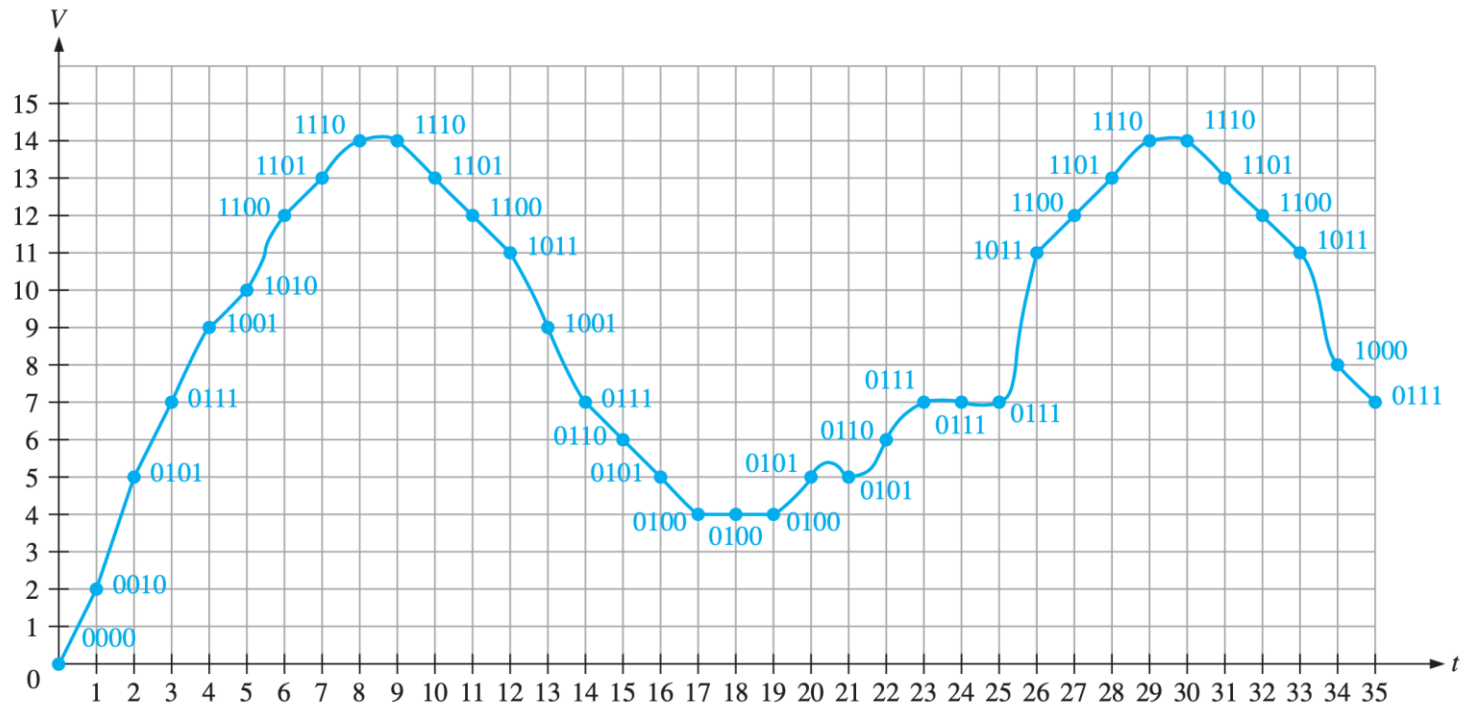
Cơ cấu chấp hành

- Là thiết bị nhận lệnh điều khiển và năng lượng đầu vào để biến thành chuyển động phù hợp của một hệ thống điện-cơ.
- Mở rộng: là các thiết bị nhận điều khiển từ hệ trung tâm để tác động đến môi trường vật lý.
- Thường hoạt động với công suất lớn.
- Ví dụ:
 - Động cơ
 - Van
 - ...

Nội dung

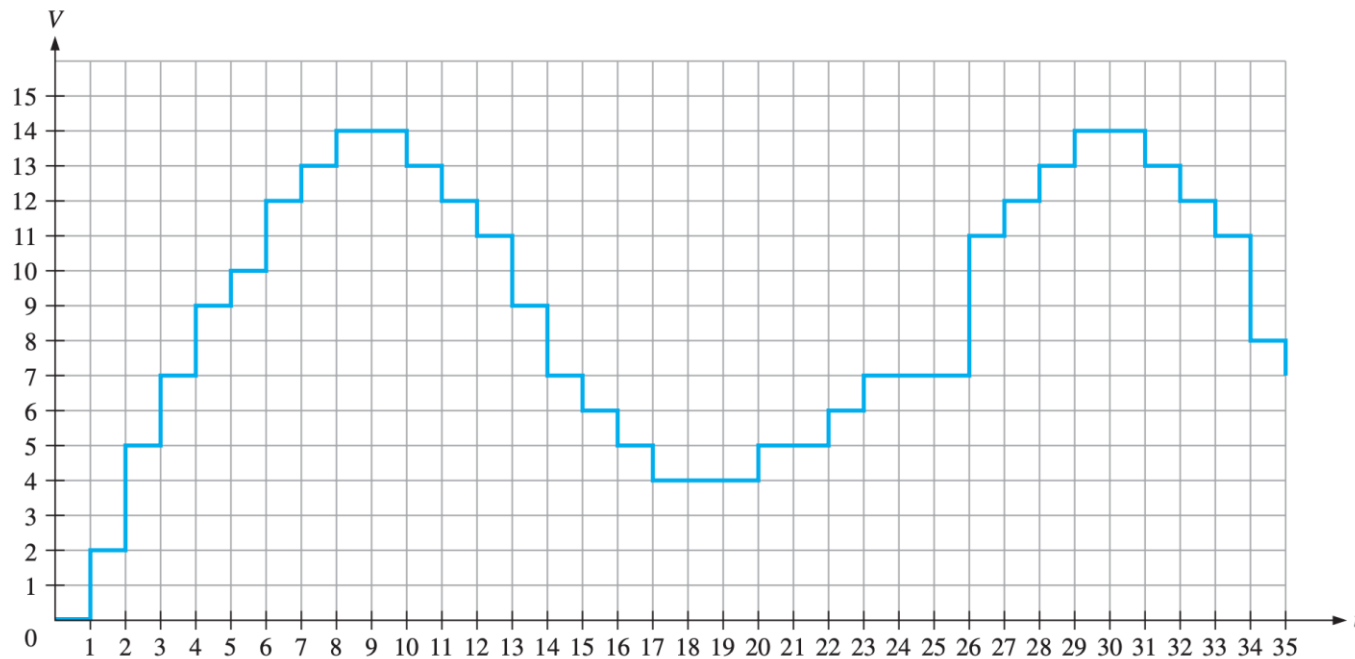
- Giới thiệu
- **ADC/DAC**
- Ghép nối ADC
- Ghép nối thiết bị công suất cao

Analog-to-Digital



- Các điểm rời rạc trên tín hiệu tương tự

Digital-to-Analog

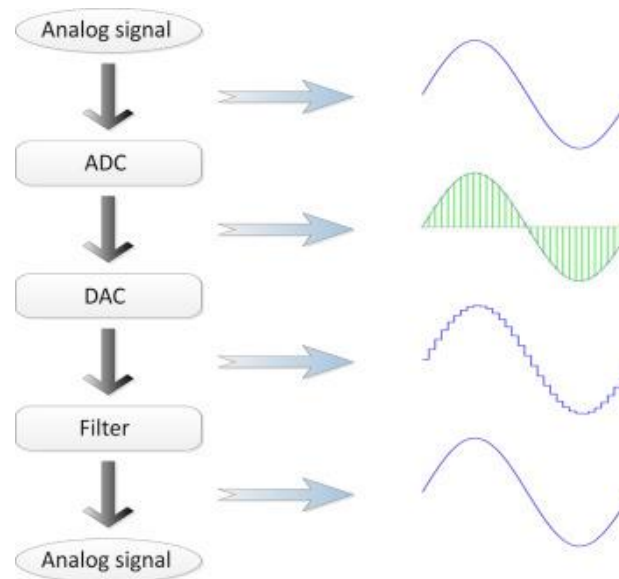


- Tái tạo tín hiệu tương tự từ xấp xỉ tín hiệu số

ADC/DAC

- ADC: Analog to Digital Converter
 - Là thiết bị chuyển đổi tín hiệu tương tự (điện áp) thành chuỗi giá trị số tương ứng theo thời gian (tín hiệu số).
- DAC: Digital to Analog Converter
 - Thiết bị chuyển đổi tín hiệu số thành tín hiệu tương tự (điện áp).

$$U(t) \leftrightarrow X(k)$$



ADC/DAC

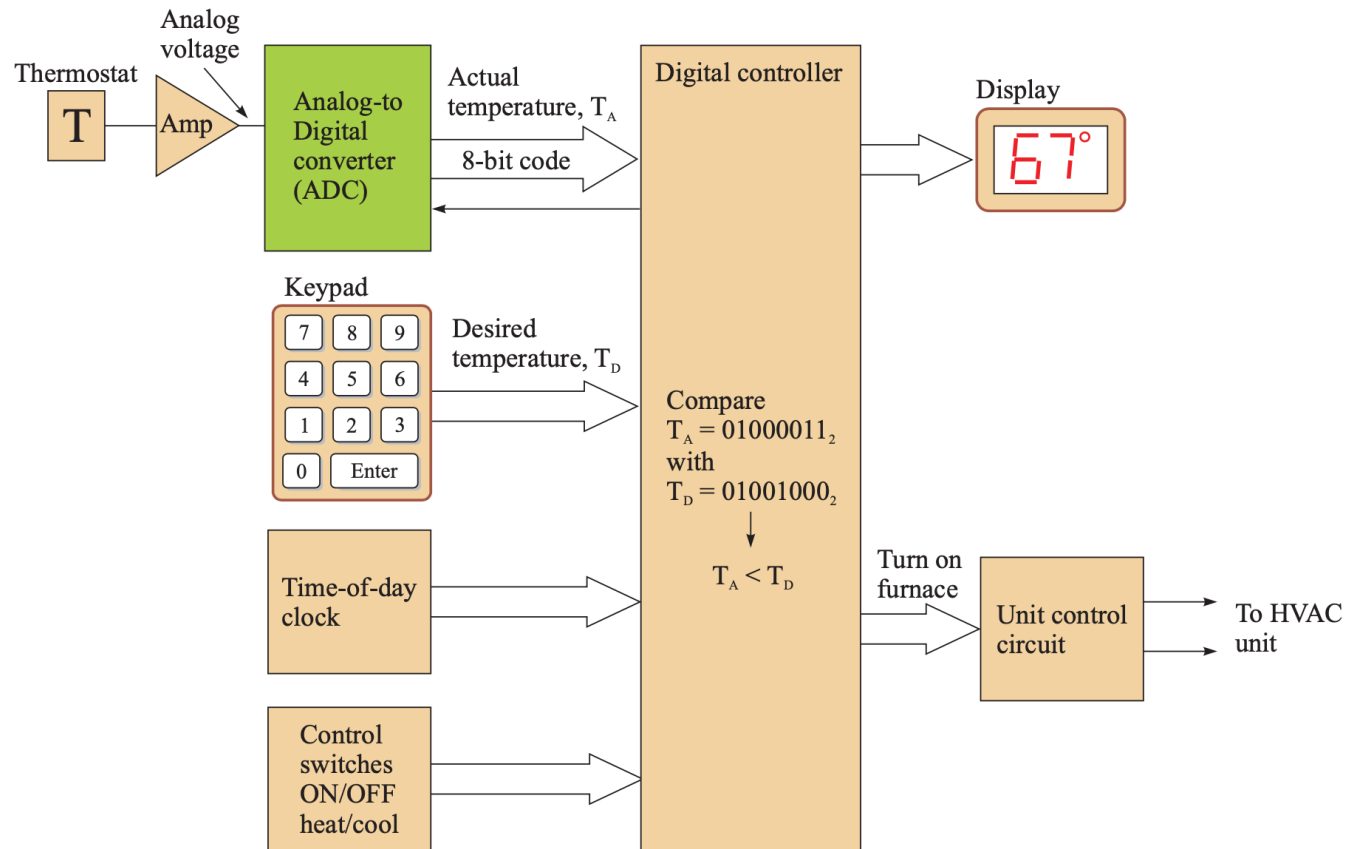
- Công thức chuyển đổi:

$$U = U_{ref} \times \frac{x_{n-1}x_{n-2} \dots x_1x_0}{2^n}$$

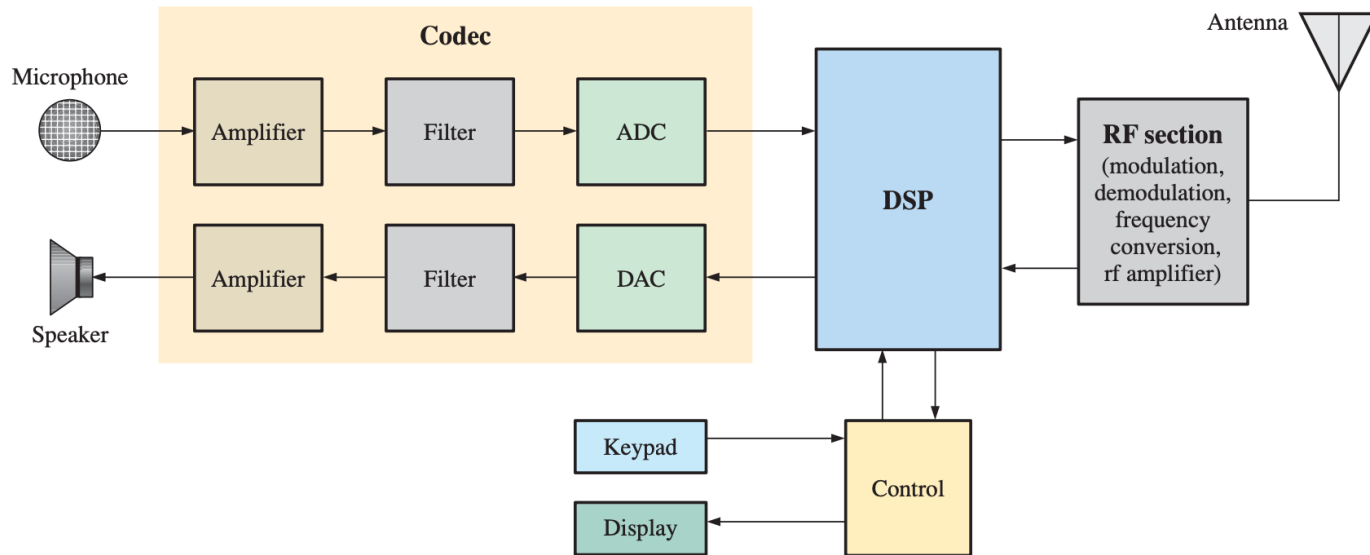
U_{ref} : điện áp tham chiếu, là hằng số của mạch ADC/DAC

- ADC: có U_{in} , cần tìm X tương ứng.
- DAC: có X, cần tạo ra điện áp U_{out} tương ứng.

Ứng dụng ADC - Nhiệt kế điện tử

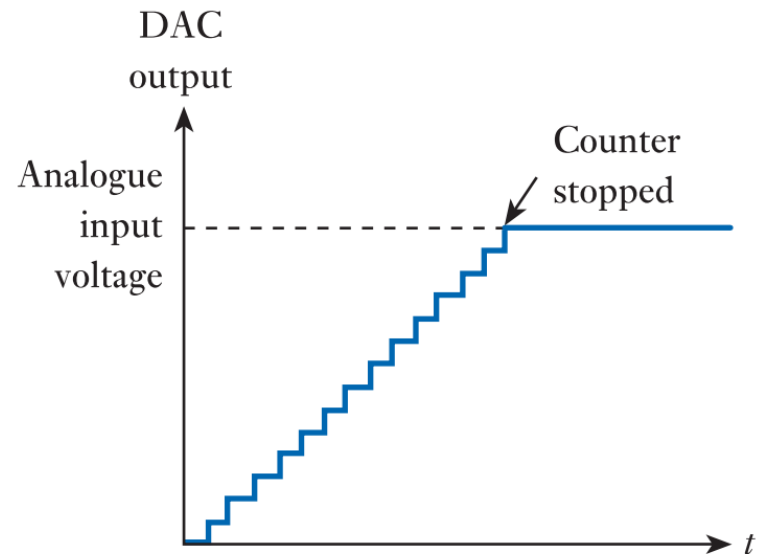
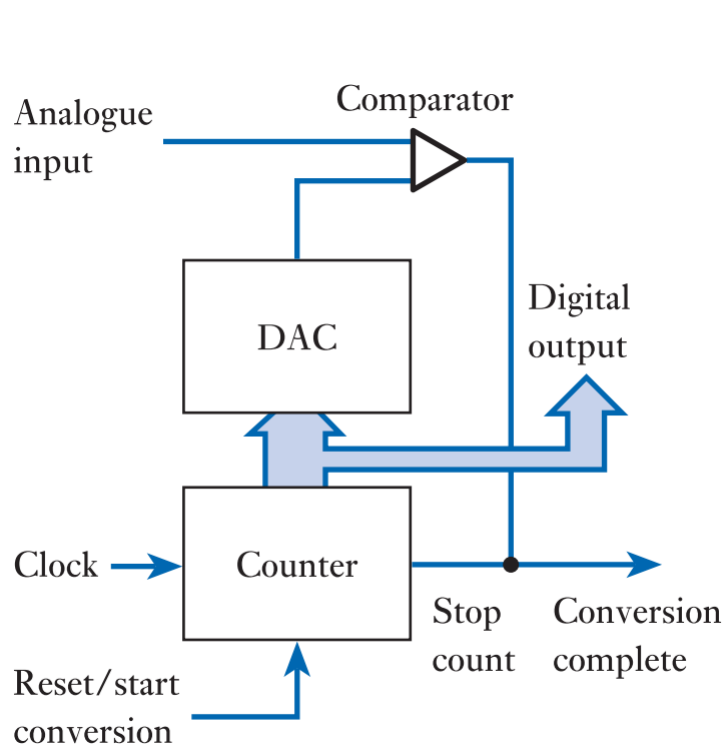


Ứng dụng ADC/DAC - Điện thoại



Counter ADC

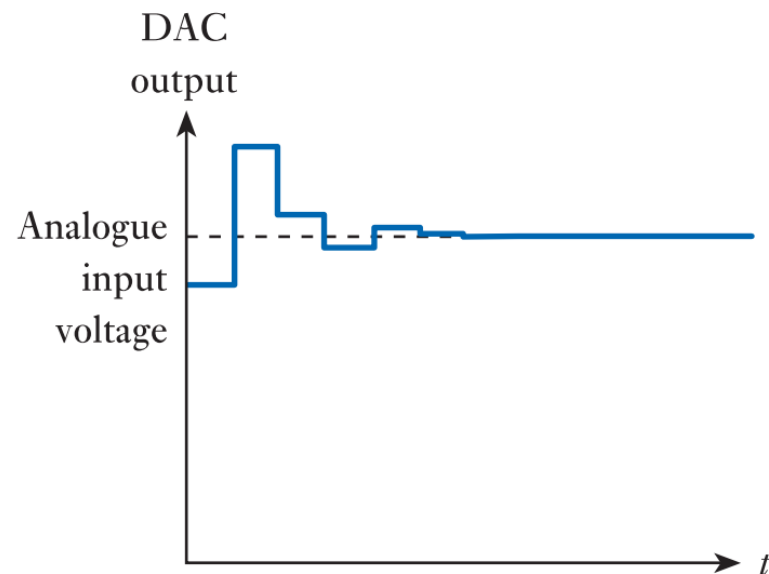
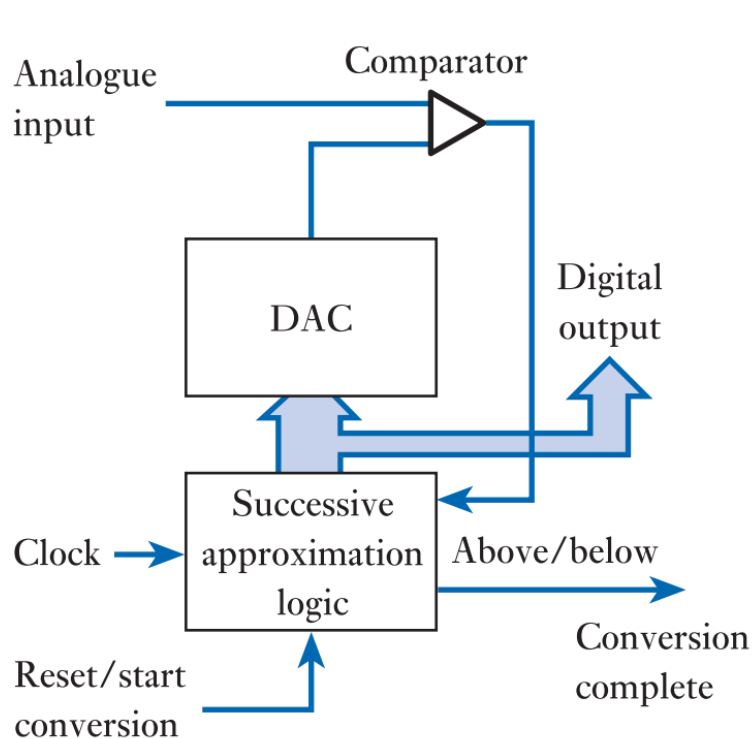
- Initially, the counter is zeroed and then starts to count up.
- Stopped when the DAC output (voltage) becomes equal to the analog input signal.



- Pros and cons:
 - One of the simplest forms of ADC
 - Relatively slow in operation

Successive Approximation ADC

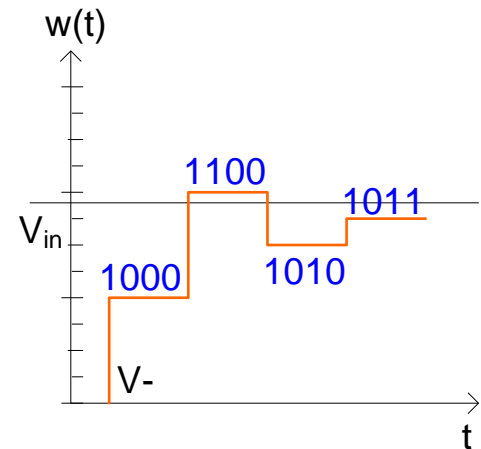
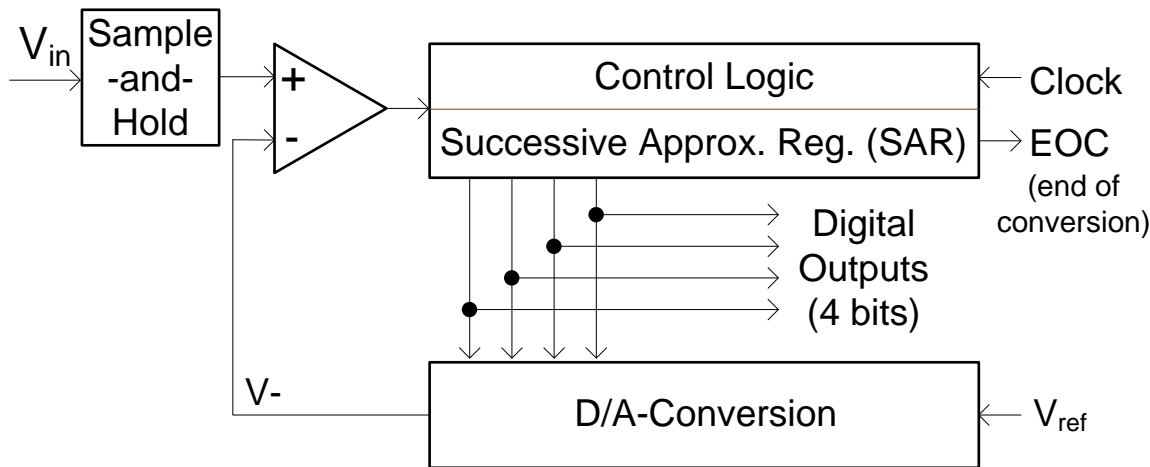
- The up counter in the counter ADC is replaced by logic circuitry - Successive Approximation Logic.



- Pros and cons:
 - Hardware efficiency
 - Relatively fast, compared to Counter ADC

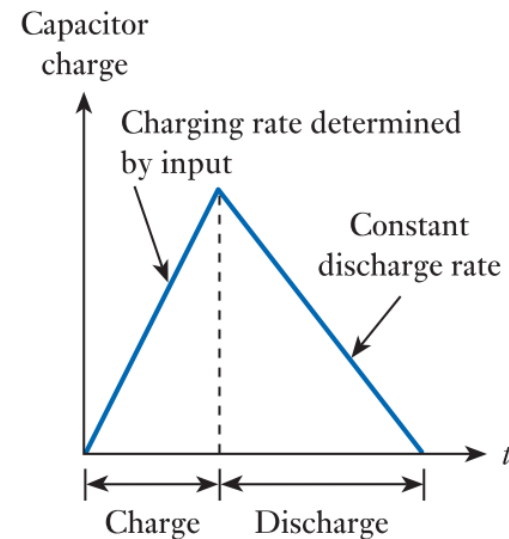
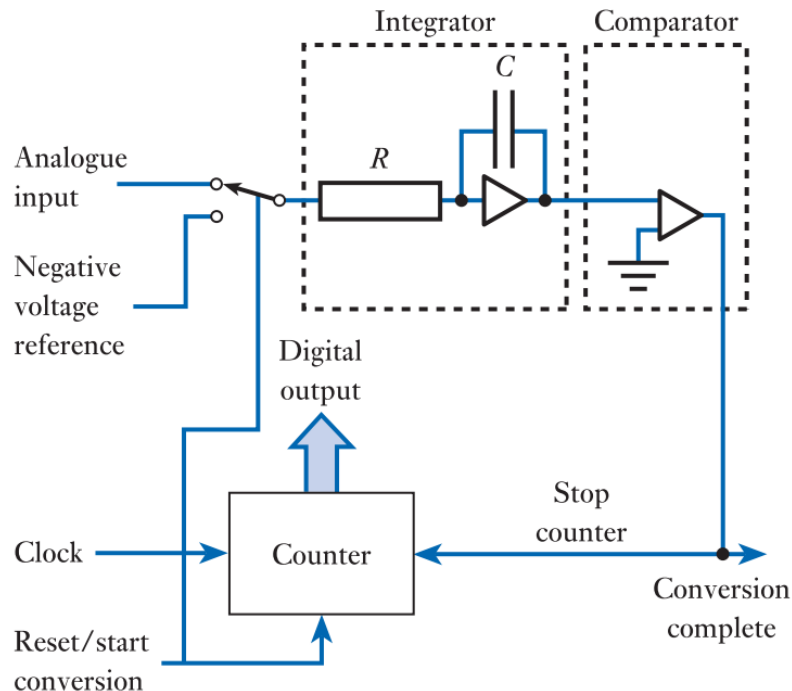
Successive Approximation ADC

- Initially, SAR = 100..0 (i.e., MSB = 1) $\rightarrow V_- = \frac{1}{2} V_{ref}$
- If $\frac{1}{2} V_{ref} > V_{in}$, MSB = 0. Otherwise, MSB = 1 (unchanged)
- Then, the next bit in the SAR = 1, and the same test is done.
- Continue this **binary search** until every bit in the SAR has been tested.



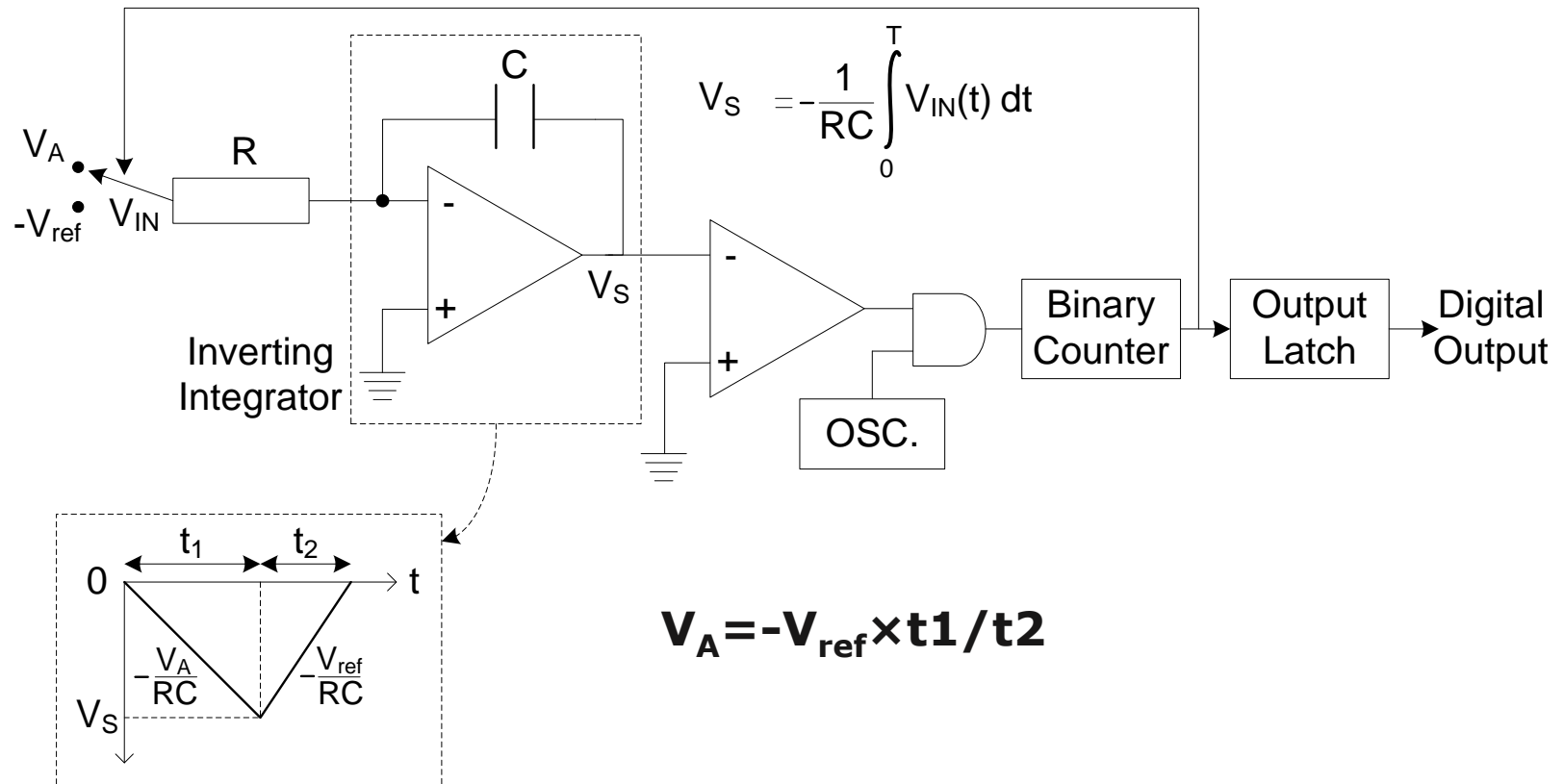
Dual Slope Integrating ADC

- An op-amp is used to integrate the input signal for a fixed period \rightarrow to produce a charge on the integrator's capacitor that is proportional to the input voltage.



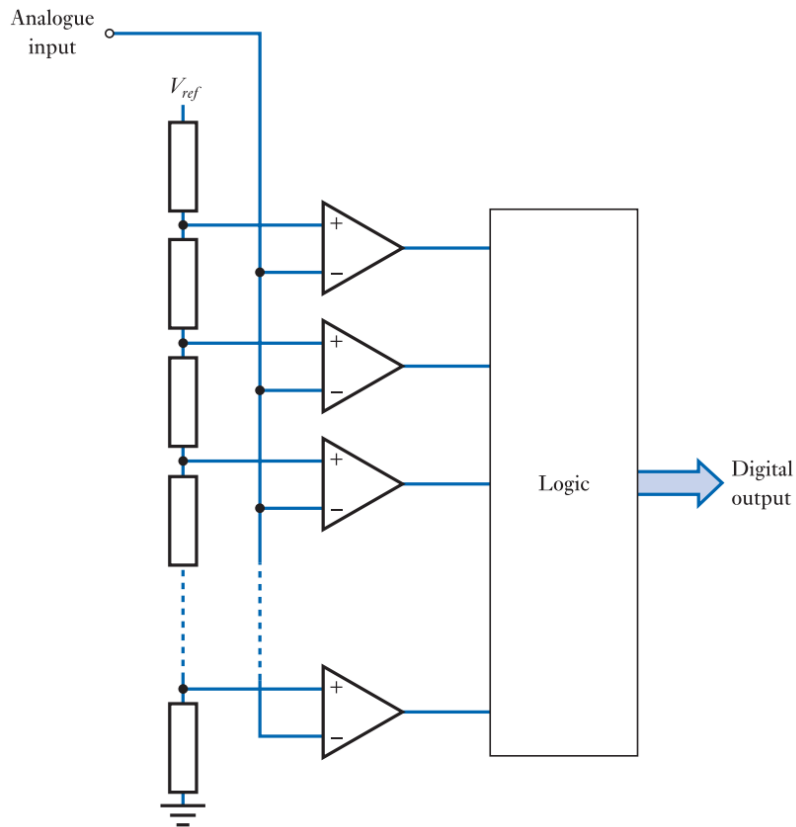
- Pros and cons:
 - The most accurate ADC among all
 - The slowest ADC among all

Dual Slope Integrating ADC



Flash ADC

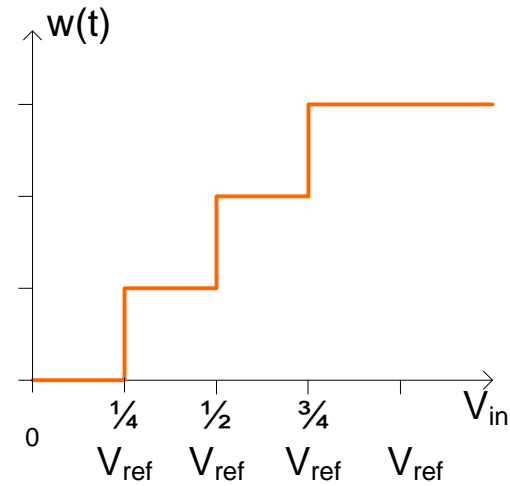
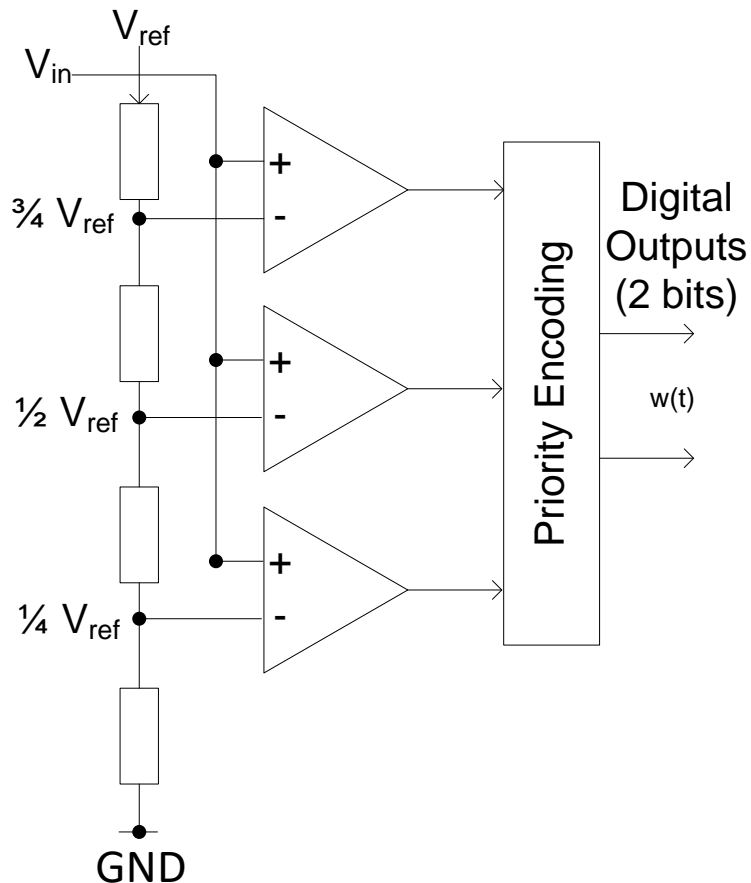
- Each **separate comparator** compares the input voltage with **every discernible voltage step** within the converter's range.



- Pros and cons:
 - The fastest among various forms of ADCs
 - Hardware inefficiency

Flash ADC

- Example: 2-bit flash ADC

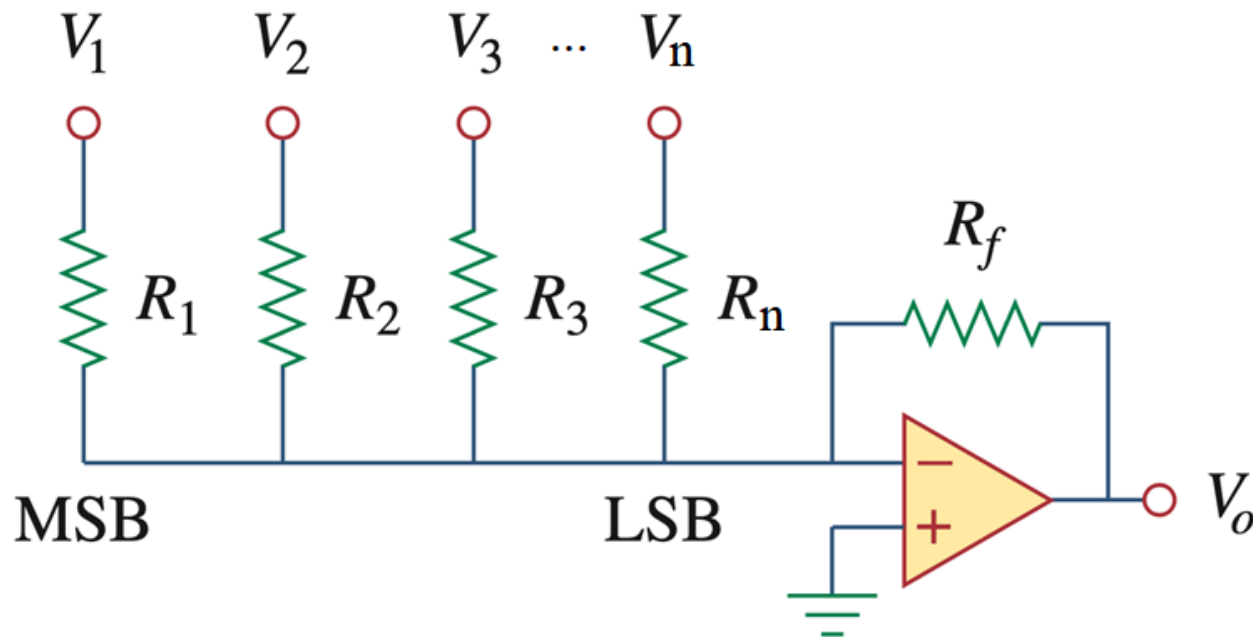


4 to 2 Priority Encoder

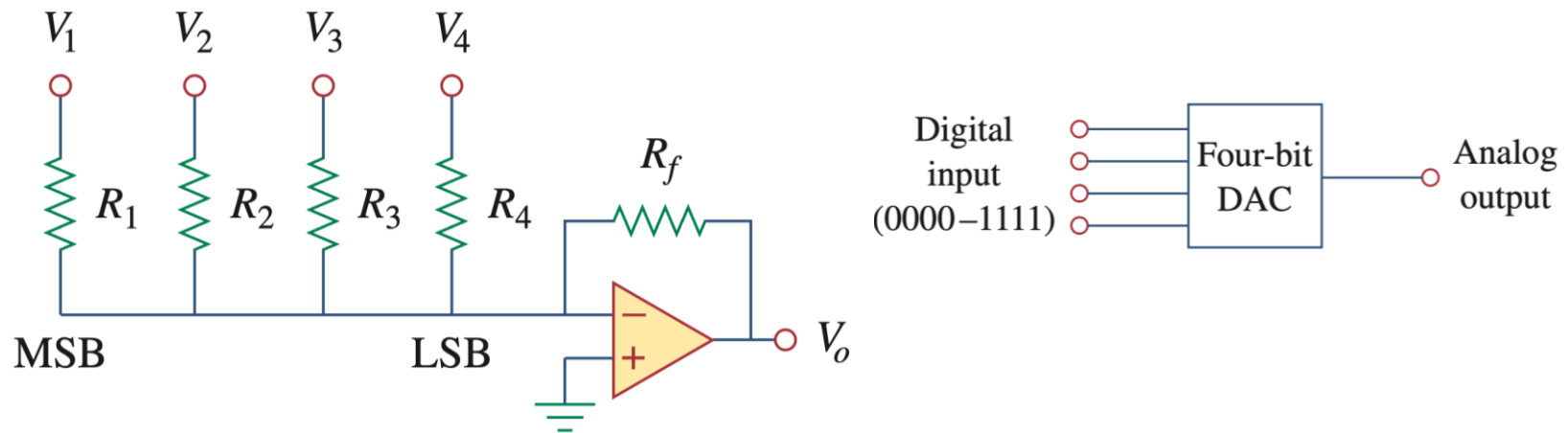
I_3	I_2	I_1	I_0	O_1	O_0	V
0	0	0	0	x	x	0
0	0	0	1	0	0	1
0	0	1	x	0	1	1
0	1	x	x	1	0	1
1	x	x	x	1	1	1

Binary Weighted Resistor DAC

- Aka Binary Weighted Ladder
- Input voltages (V_1, \dots, V_n): Logic 1 = +5V (V_{ref}) or Logic 0 = 0V
- $R_1 = R$; $R_2 = 2R$; $R_3 = 4R$, ..., $R_n = 2^{n-1}R$



4-bit Binary Weighted Resistor DAC

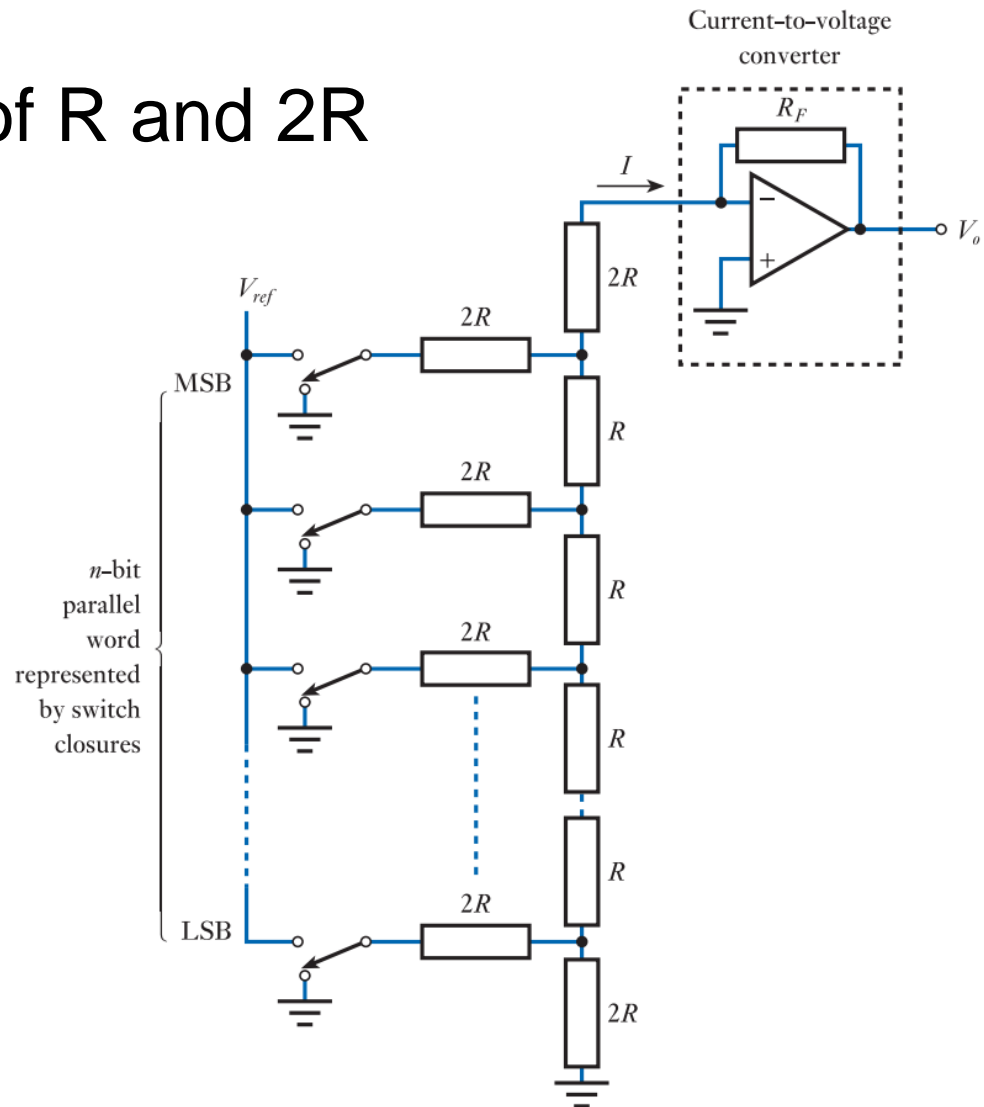


- Output voltage:

$$-V_o = \frac{R_f}{R_1}V_1 + \frac{R_f}{R_2}V_2 + \frac{R_f}{R_3}V_3 + \frac{R_f}{R_4}V_4$$

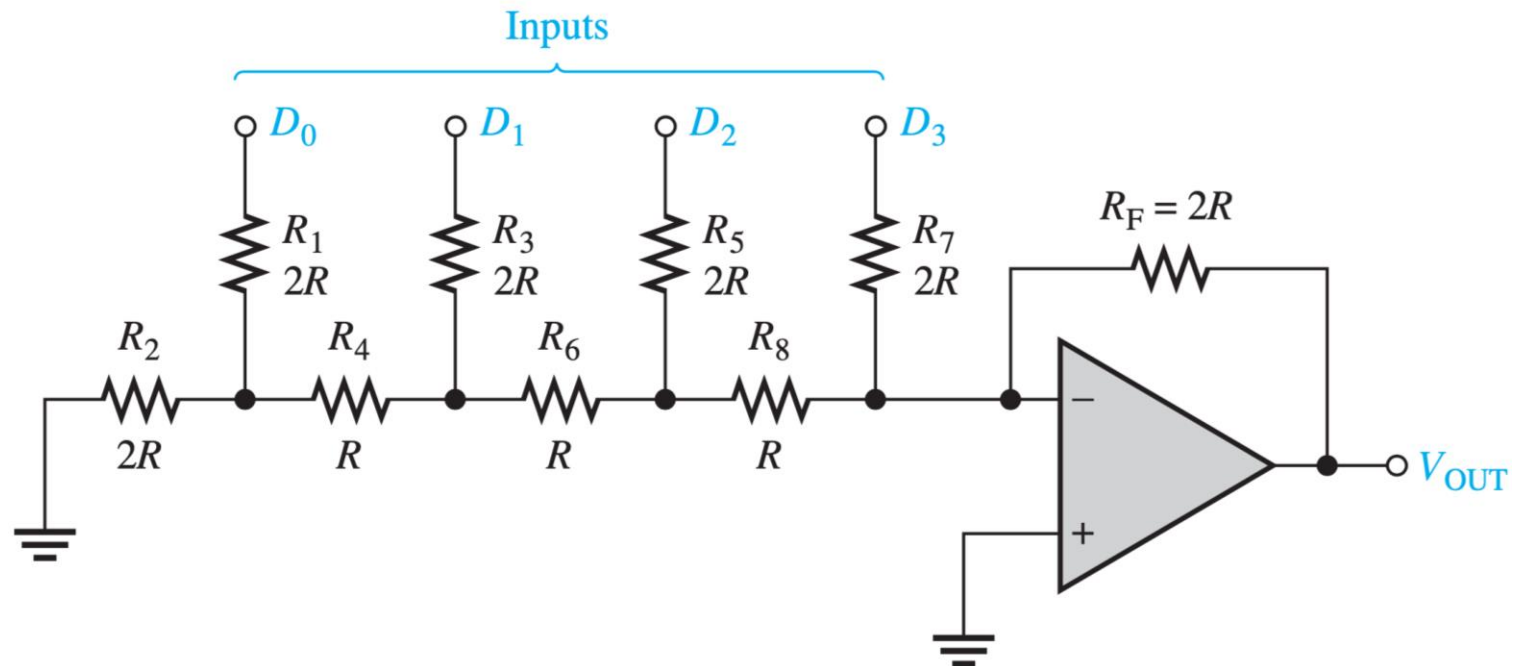
R-2R Ladder DAC

- Only resistors of R and $2R$ are required.



4-bit R-2R Ladder DAC

- Logic 1 = +5V, Logic 0 = 0V

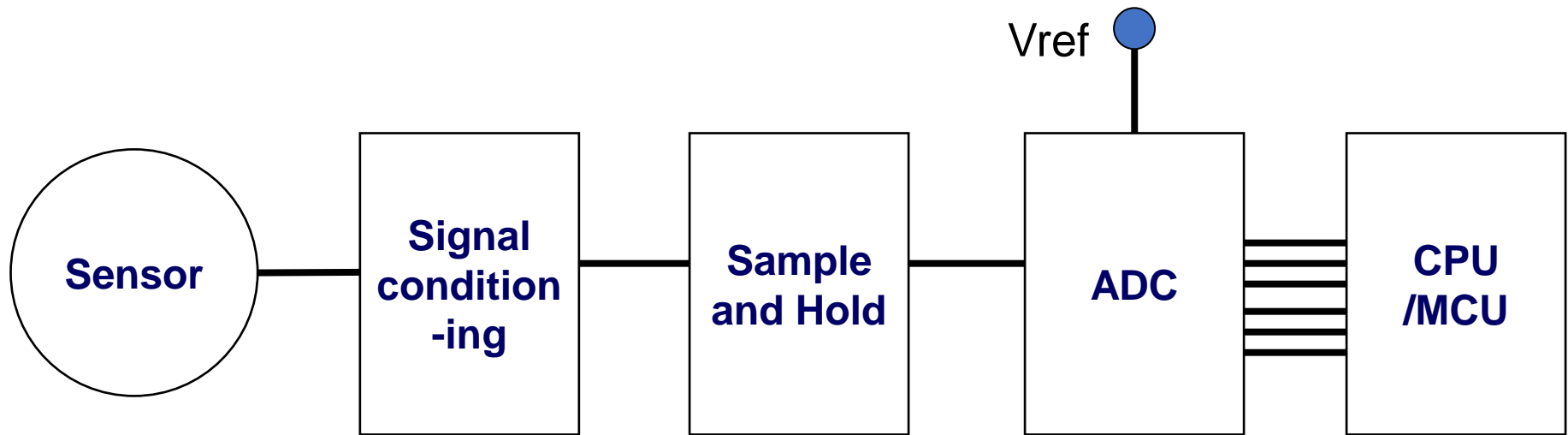


Nội dung

- Giới thiệu
- ADC/DAC
- Ghép nối ADC
- Ghép nối thiết bị công suất cao

Ghép nối ADC với Bộ xử lý

- Sơ đồ khối ghép nối:

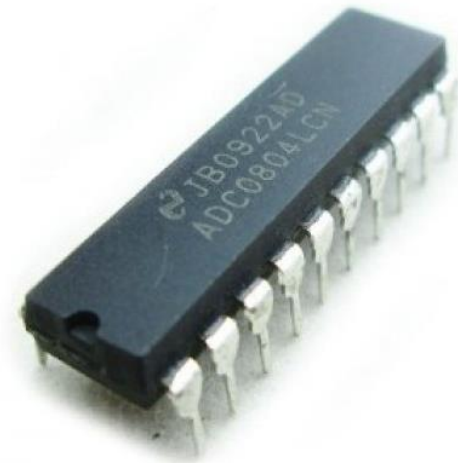


Ví dụ ghép nối ADC với MCU

- Cảm biến nhiệt độ: LM35
- ADC0804
- Vi điều khiển: 8051

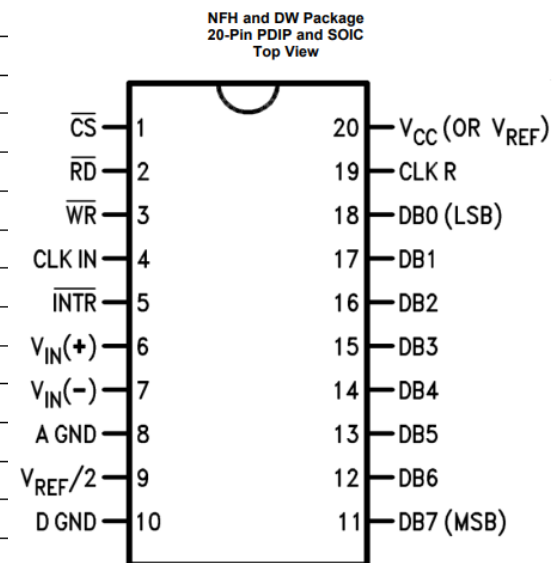
ADC080X

- 8-bit successive approximation converters (ADC)
- Key Specifications:
 - Resolution: 8 bits (digital out range is from 0 – 255)
 - Input voltage range: 2.5 – 6.5V
 - Conversion time: 100 μ s
- $V_{ref} = 5V \rightarrow$ Every 19.53mV increase in analogous value causes an increase in one bit on the digital side.
- Total Error: $\pm 1/4$ LSB, $\pm 1/2$ LSB and ± 1 LSB



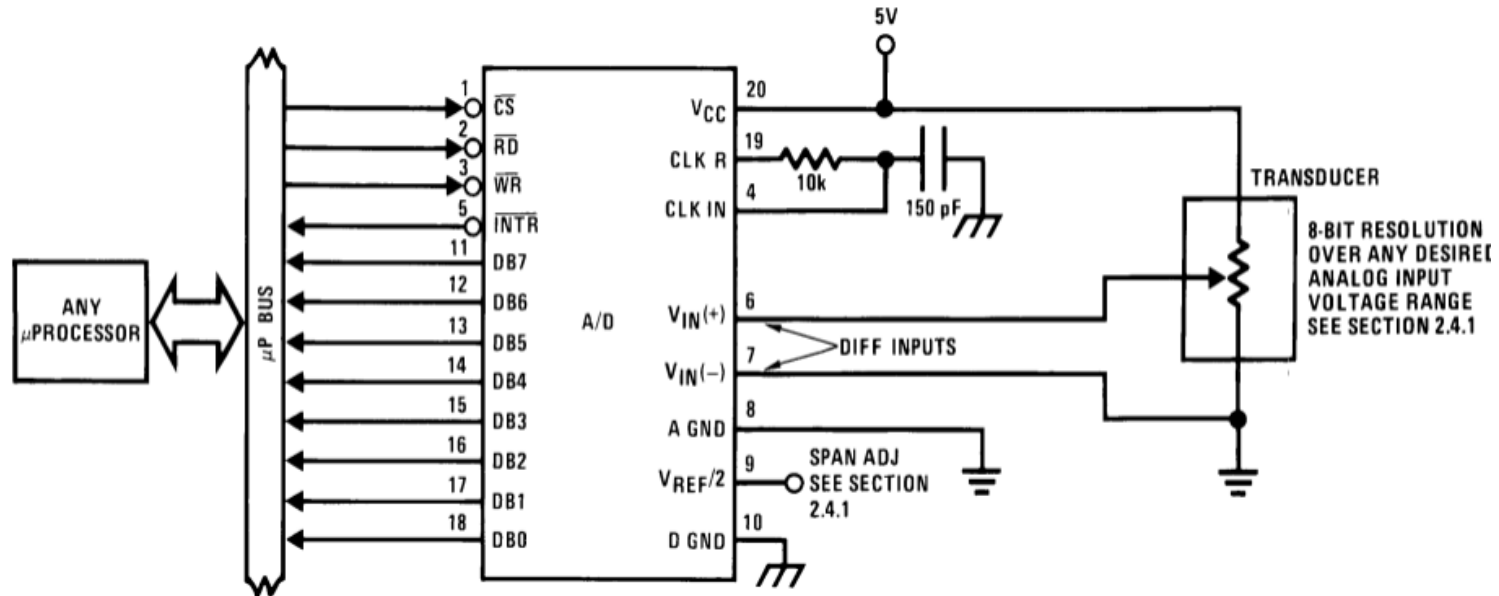
ADC0804

PIN		I/O	DESCRIPTION
NO.	NAME		
1	CS	I	Chip Select
2	RD	I	Read
3	WR	I	Write
4	CLK IN	I	External Clock input or use internal clock gen with external RC elements
5	INTR	O	Interrupt request
6	$V_{IN}(+)$	I	Differential analog input+
7	$V_{IN}(-)$	I	Differential analog input-
8	A GND	I	Analog ground pin
9	$V_{REF}/2$	I	Reference voltage input for adjustment to correct full scale reading
10	D GND	I	Digital ground pin
11	DB7	O	Data bit 7
12	DB6	O	Data bit 6
13	DB5	O	Data bit 5
14	DB4	O	Data bit 4
15	DB3	O	Data bit 3
16	DB2	O	Data bit 2
17	DB1	O	Data bit 1
18	DB0 (LSB)	O	Data bit 0
19	CLK R	I	RC timing resistor input pin for internal clock gen
20	V_{CC} (or V_{REF})	I	+5V supply voltage, also upper reference input to the ladder



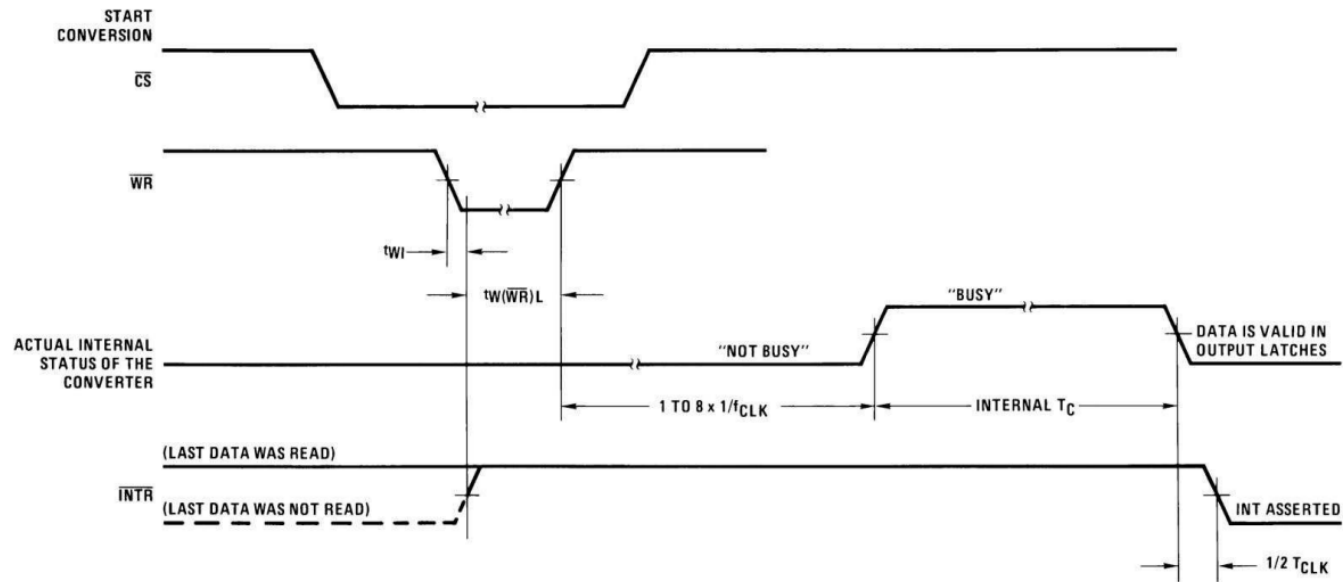
Typical Application Schematic

- Operates With Any 8-Bit μ P Processors or as a Stand-Alone Device
- Interface to Temp Sensors, Voltage Sources, and Transducers

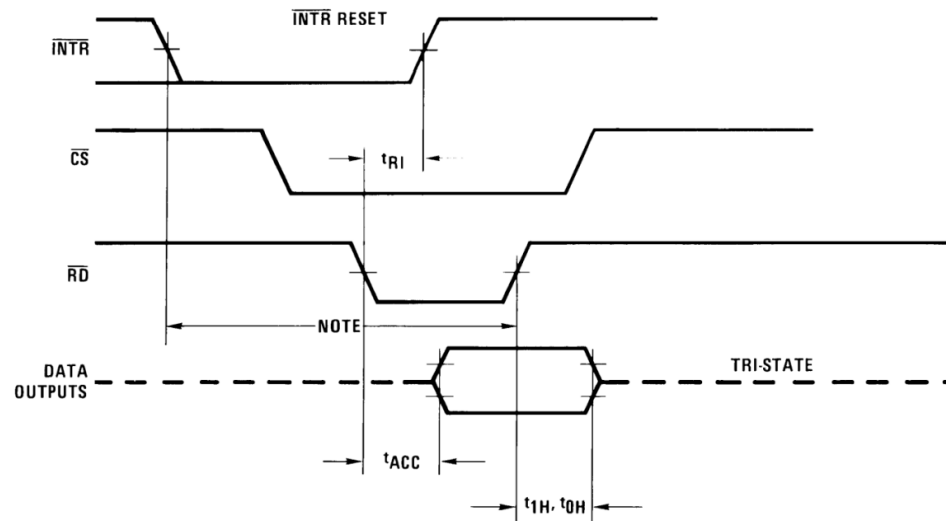


ADC0804

Start Conversion



Output Enable and Reset With \overline{INTR}



Sample Code

```
#define ADC0804_PORT      P1
#define ADC0804_CS        P3_5
#define ADC0804_WR        P3_6
#define ADC0804_RD        P3_7
#define MOTOR_IN1         P3_0
#define MOTOR_IN2         P3_1
#define MOTOR_EN1         P3_2

unsigned char temperature;
unsigned long int adc_value;
unsigned long int V_LM35_out;

/*Read digital dat from adc0804*/
unsigned char read_adc0804(void);
```


Sample Code

```
void main()
{
    ADC0804_CS = 0;
    ADC0804_RD = 0;
    ADC0804_WR = 0;
    while(1)
    {
        adc_value = read_adc0804();
        V_LM35_out = (adc_value * 5000)/256;
        temperature = V_LM35_out/10;
        delay_ms(500);

        if(temperature > 25){
            MOTOR_IN1 = 1;
            MOTOR_IN2 = 0;
            MOTOR_EN1 = 1;
        }
        else{
            MOTOR_EN1 = 0;
        }
    }
}
```

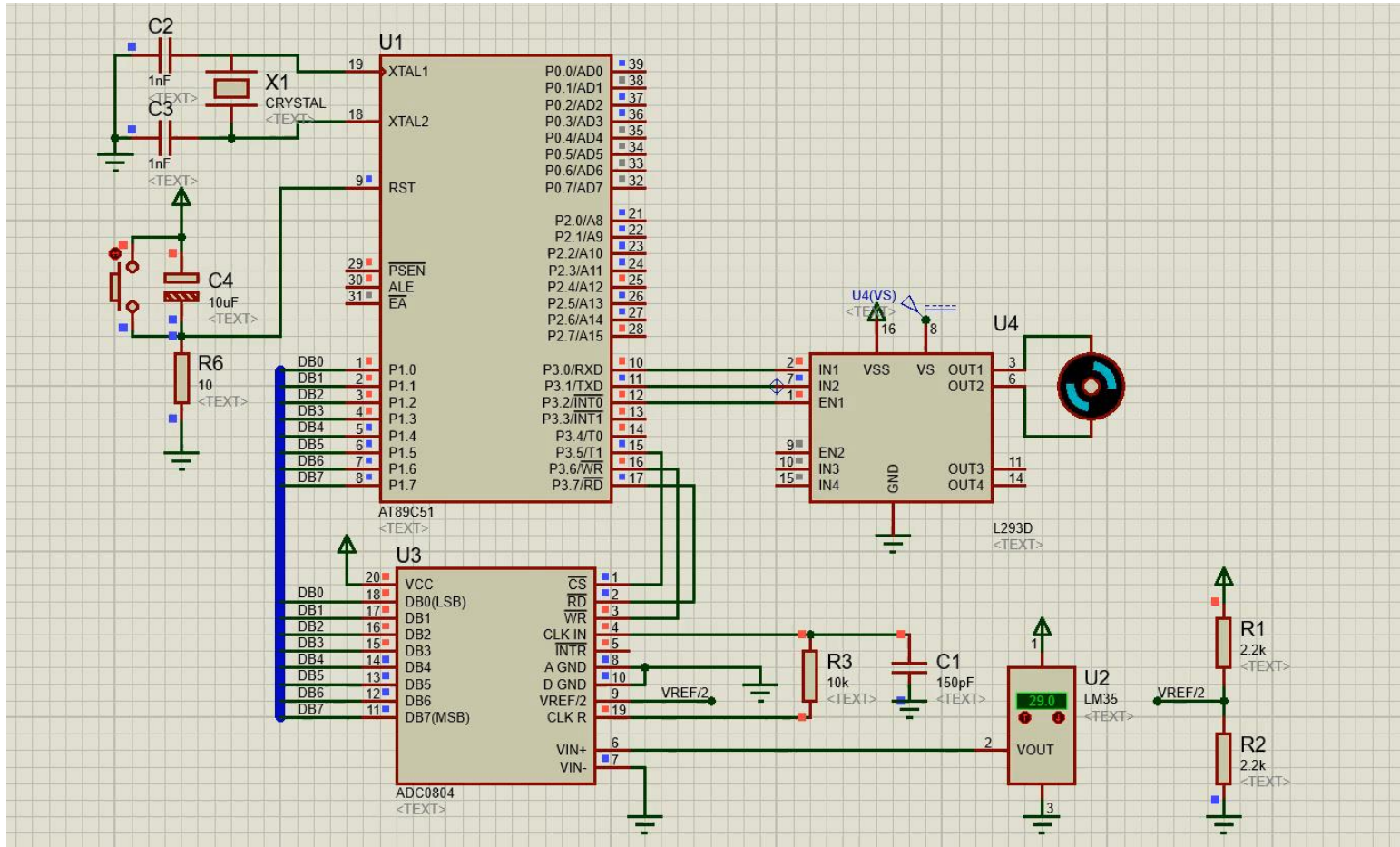
Sample Code

```
unsigned char read_adc0804(void)
{
    unsigned char temp_value;

    ADC0804_WR = 0;
    ADC0804_WR = 1;
    temp_value = ADC0804_PORT;

    return temp_value;
}
```

Demo



Lập trình ghép nối ADC với Arduino

BOARD	OPERATING VOLTAGE	USABLE PINS	MAX RESOLUTION
Uno	5 Volts	A0 to A5	10 bits
Mini, Nano	5 Volts	A0 to A7	10 bits
Mega, Mega2560, MegaADK	5 Volts	A0 to A14	10 bits
Micro	5 Volts	A0 to A11*	10 bits
Leonardo	5 Volts	A0 to A11*	10 bits
Zero	3.3 Volts	A0 to A5	12 bits**
Due	3.3 Volts	A0 to A11	12 bits**
MKR Family boards	3.3 Volts	A0 to A6	12 bits**

**: require `analogReadResolution()` to change it to 12 bits.

Lập trình ghép nối ADC với Arduino

- `analogRead(pin)`
- `analogWrite()`
- `analogReference()`
 - **DEFAULT**: the default analog reference (5V/3.3V)
 - **INTERNAL**: built-in reference
 - ✓ 1.1 volts on the ATmega168 or ATmega328P
 - ✓ 2.56 volts on the ATmega32U4 and ATmega8
 - **INTERNAL1V1**: 1.1V reference (Arduino Mega)
 - **INTERNAL2V56**: 2.56V reference (Arduino Mega)
 - **EXTERNAL**: voltage of the AREF pin (0 to 5V only)

Lập trình ghép nối ADC với Arduino

- `analogWrite(pin, value)`
- `analogWriteRange(new_range);`
- `analogWriteFreq(new_frequency);`
- `dacWrite(pin, value)`
- ESP32:
 - 18 chân ADC (DEVKIT V1 chỉ có 15 chân). Mặc định độ phân giải 12 bit.
 - 2 chân DAC, độ phân giải 8 bit.

Lập trình ghép nối ADC với Arduino

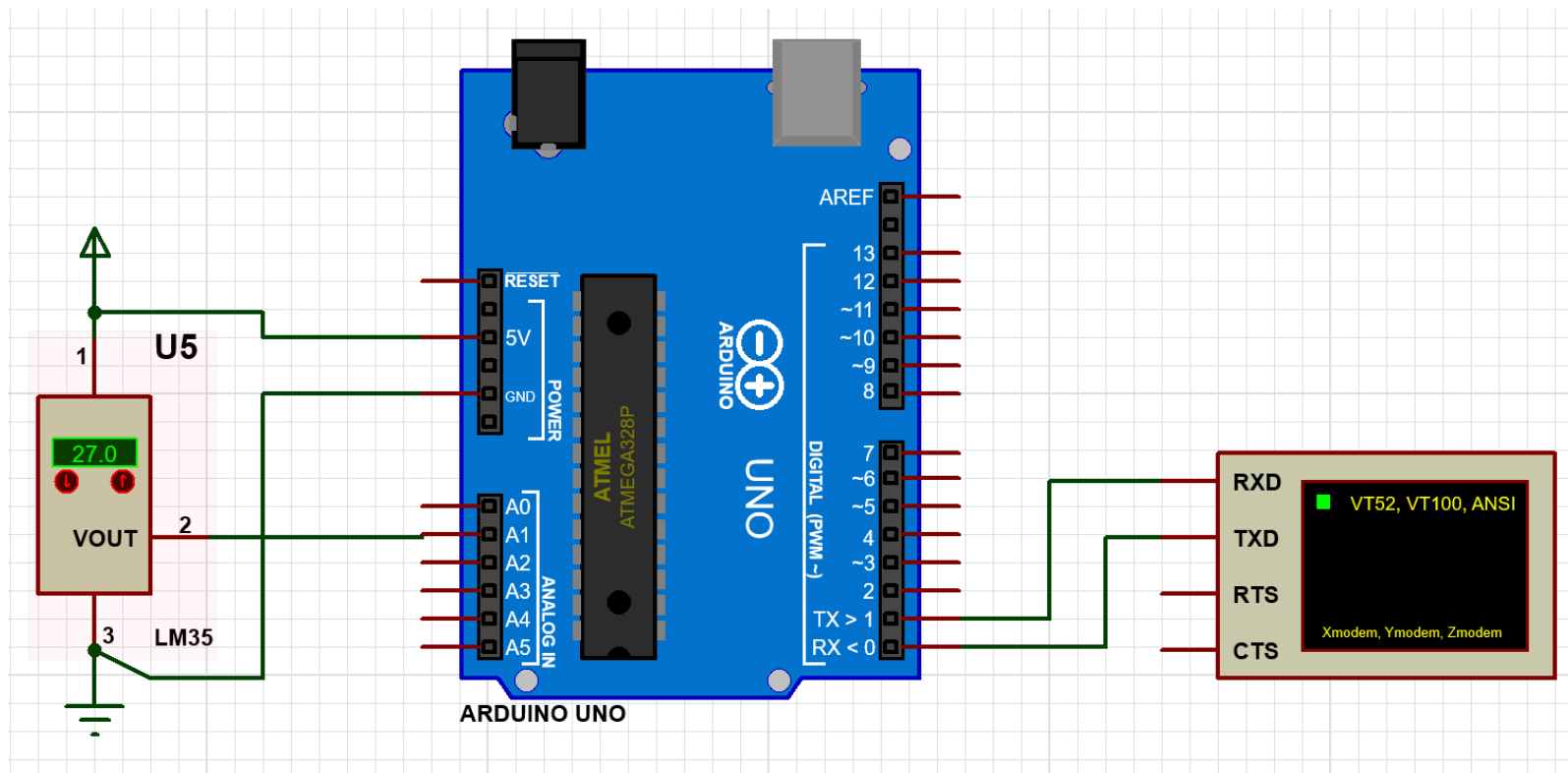
- Ví dụ:

```
int analogPin = A3;
int val = 0;

void setup() {
    Serial.begin(9600);
}

void loop() {
    val = analogRead(analogPin);
    Serial.println(val);
}
```

Đo nhiệt độ



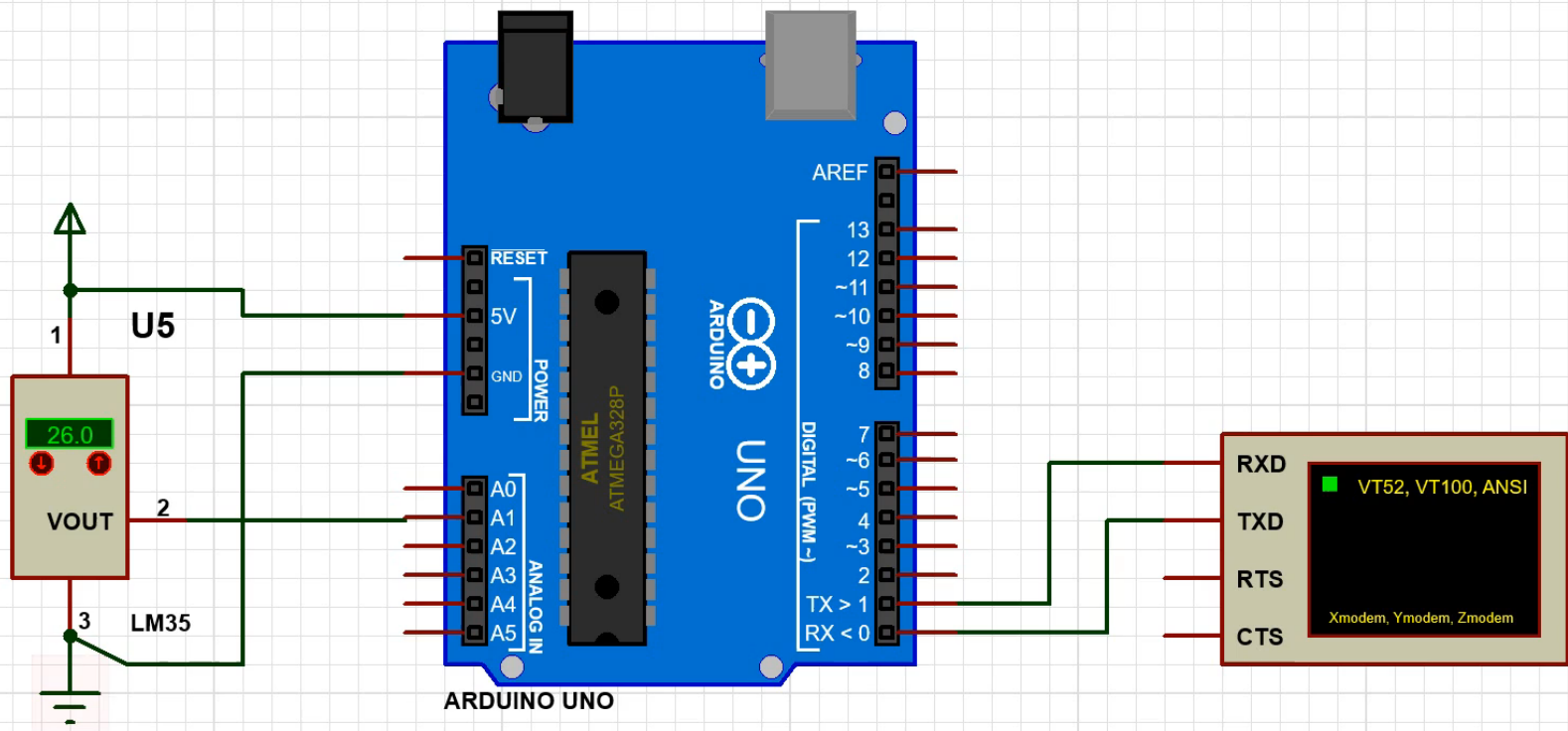
Đo nhiệt độ

```
#define tempPin A1
int val;

void setup()
{
  Serial.begin(9600);
}

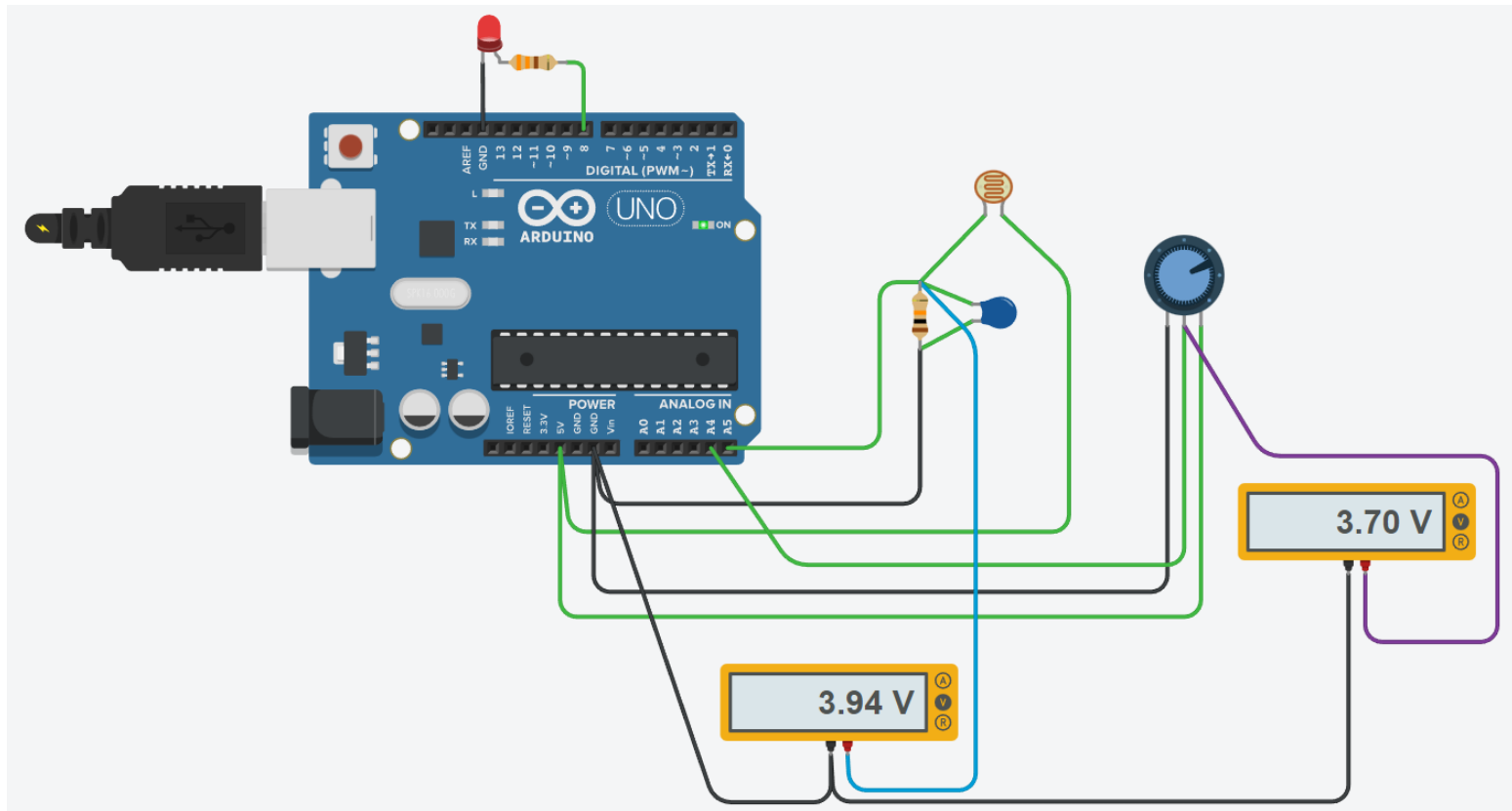
void loop()
{
  val = analogRead(tempPin);
  float mv = (val/1024.0)*5000;
  float cel = mv/10;
  Serial.print("Temp = ");
  Serial.print(cel);
  Serial.println("*C");
  delay(1000);
}
```

Demo



Đo cường độ sáng

- Đo cường độ sáng → Bật đèn khi tối, tắt đèn khi sáng



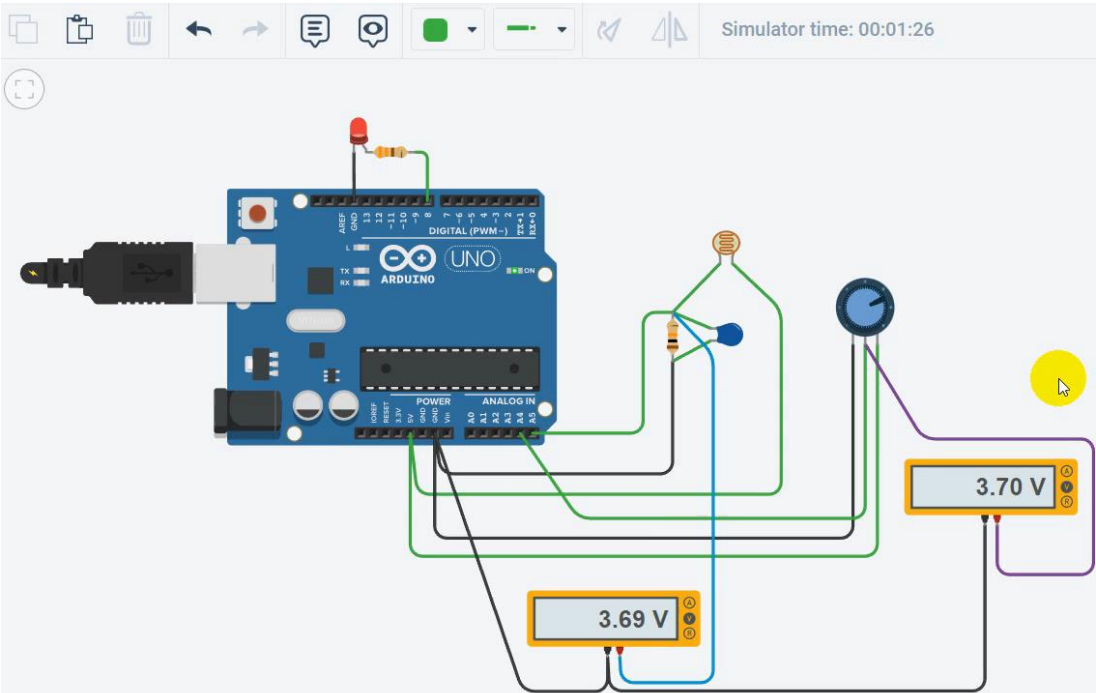
Đo cường độ sáng

```
void setup()
{
  Serial.begin(9600);
  pinMode(8, OUTPUT);
}

void loop()
{
  int value_photoresistor = analogRead(A5);
  int value_potentiometer = analogRead(A4);

  Serial.print("Value of photoresistor = ");
  Serial.println(value_photoresistor);
  Serial.print("Value of potentiometer = ");
  Serial.println(value_potentiometer);
  if (value_photoresistor < value_potentiometer){
    digitalWrite(8, HIGH);
  } else{
    digitalWrite(8, LOW);
  }
  delay(1000);
}
```

Demo



The screenshot shows an Arduino Uno R3 board in a simulation environment. The board is connected to a potentiometer and a photoresistor. Two voltage meters are connected to the circuit: one showing 3.69 V and another showing 3.70 V. The potentiometer is connected to the 5V and GND pins of the Arduino. The photoresistor is connected to the A5 pin. The voltage meters are connected to the potentiometer's wiper and the photoresistor's terminals.

Simulator time: 00:01:26

Code

```
1 void setup()
2 {
3   Serial.begin(9600);
4   pinMode(8, OUTPUT);
5 }
6
7 void loop()
8 {
9   int value_photoresistor = analogRead(A5);
10  int value_potentiometer = analogRead(A4);
11
12  Serial.print("Value of photoresistor = ");
13  Serial.println(value_photoresistor);
14  Serial.print("Value of potentiometer = ");
15  Serial.println(value_potentiometer);
16  if (value_photoresistor < value_potentiometer){
17    digitalWrite(8, HIGH);
18  } else{
19    digitalWrite(8, LOW);
20  }
21  delay(1000);
22 }
```

1 (Arduino Uno R3)

Serial Monitor

```
Value of potentiometer = 757
Value of photoresistor = 755
Value of potentiometer = 757
Value of photoresistor = 755
Value of potentiometer = 757
Value of photoresistor = 755
Value of potentiometer = 757
Value of photoresistor = 755
Value of potentiometer = 757
Value of photoresistor = 755
```

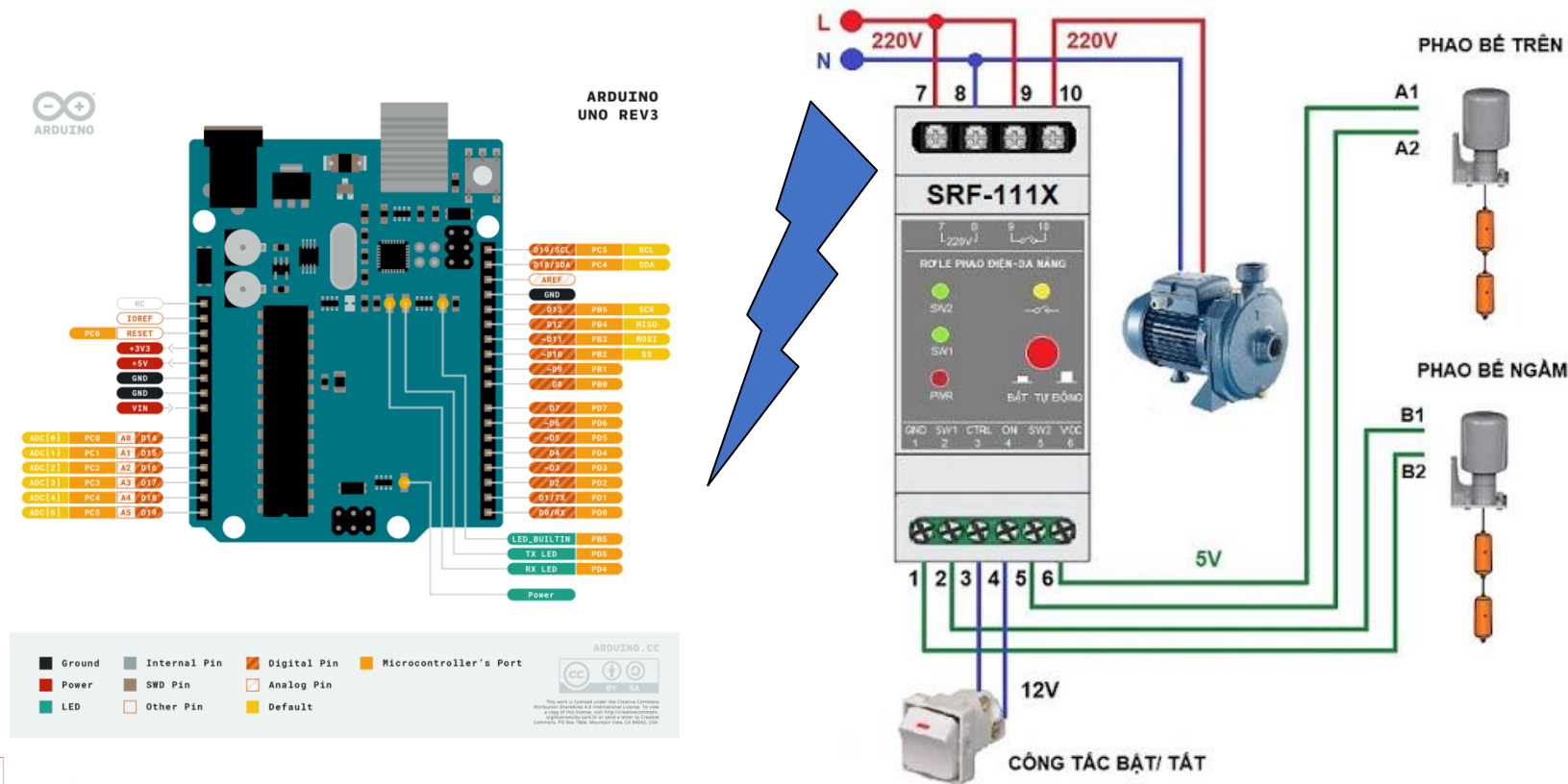
Send Clear

Nội dung

- Giới thiệu
- ADC/DAC
- Ghép nối ADC
- Ghép nối thiết bị công suất cao

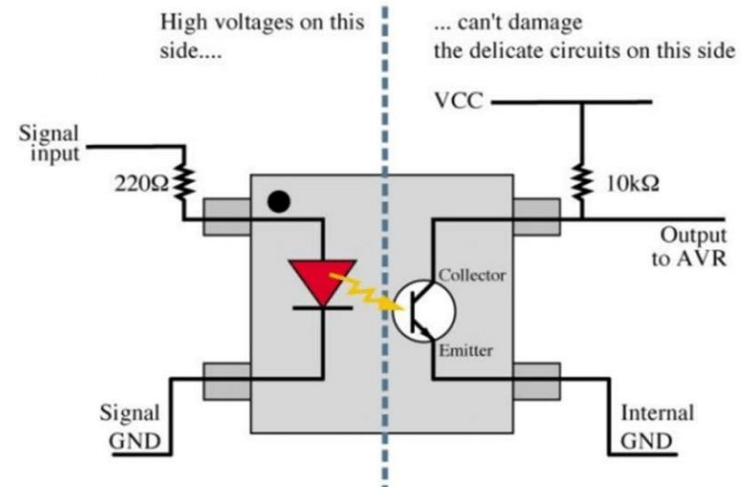
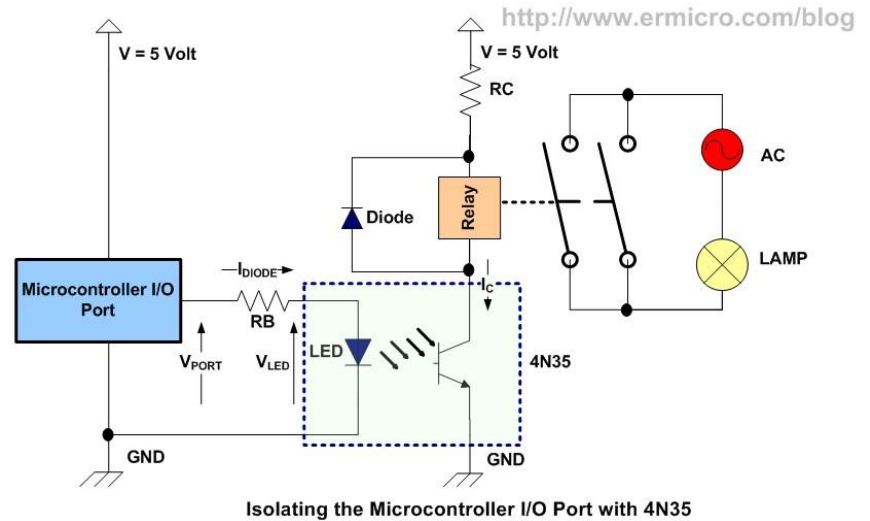
Ghép nối thiết bị công suất cao

- Nhiều nguồn từ tải có thể làm MCU treo.
- Khi có sự cố điện áp cao từ tải có thể “xông” sang mạch điều khiển → Nguy hiểm!!!

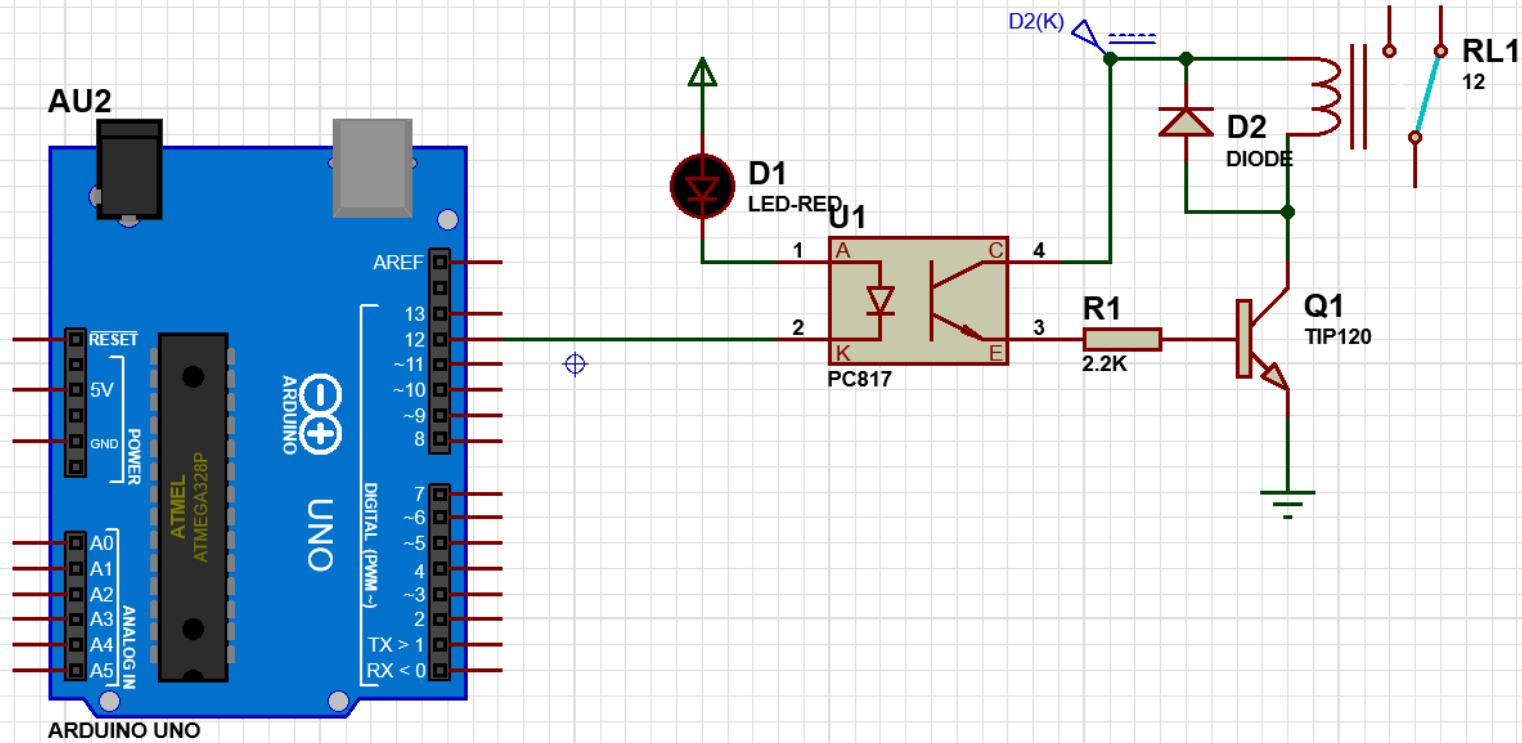


Ghép nối thiết bị công suất cao

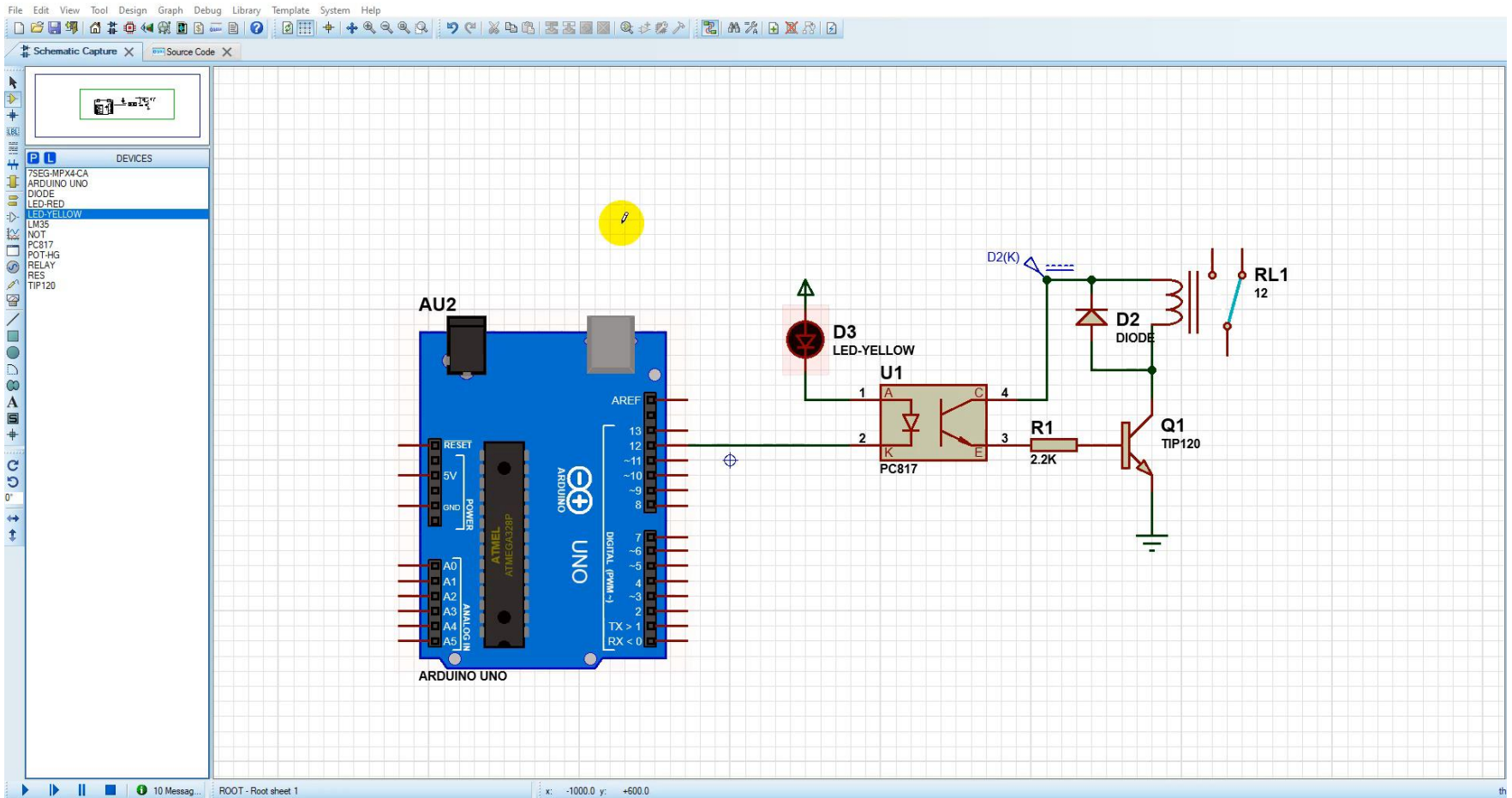
- Làm sao để cách ly?
- Relay + Optocoupler
 - Dùng relay để đóng cắt mạch điện áp cao từ mạch điều khiển điện áp thấp.
 - Dùng optocoupler để cách ly quang chống nhiễu cho mạch điều khiển. Có thể dùng cả input và output



Ví dụ

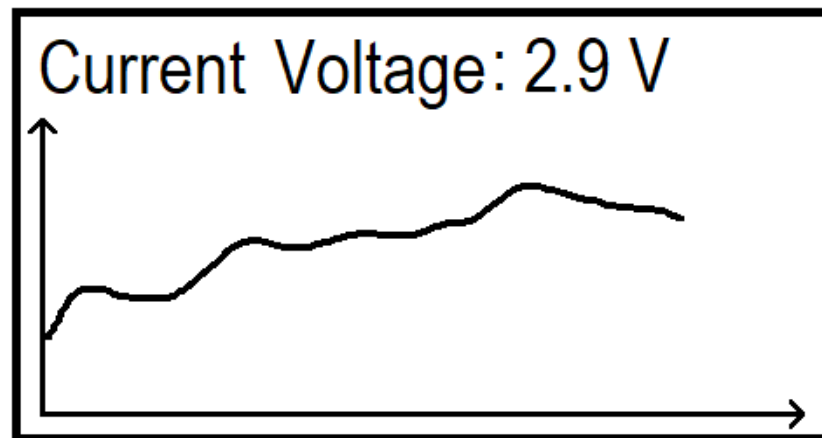


Demo



Bài tập 1

- Xây dựng mạch + chương trình trên ESP32:
 - Nối một chân GPIO (tùy chọn) với một biến trở cho phép điều chỉnh điện áp rơi trên chân đó.
 - Liên tục đo điện áp trên chân GPIO đó.
 - Hiện thị giá trị điện áp tức thì và vẽ đồ thị điện áp lên màn hình OLED SSD1306.
 - Minh họa hiển thị trên OLED SSD1306:



Bài tập 2

- Tạo tín hiệu hình sin với tần số xác định (ví dụ: 1 Hz) và gửi tín hiệu ra 1 chân DAC (tùy chọn) của ESP32.
- Hiển thị tín hiệu hình sin trên màn hình OLED SSD1306. Lưu ý, cần chọn tỷ lệ hiển thị phù hợp.
- Lắp 1 LED vào chân DAC tương ứng và quan sát độ sáng của LED.
- Thay đổi tần số của hình sin (và thay đổi tương ứng hiển thị trên màn hình OLED SSD1306) và quan sát độ sáng của LED.