

BÀI 4: GIAO TIẾP NỐI TIẾP (tiếp theo)

1. MỤC TIÊU BÀI TN

Trong bài TN này SV sẽ thực hành những phần sau:

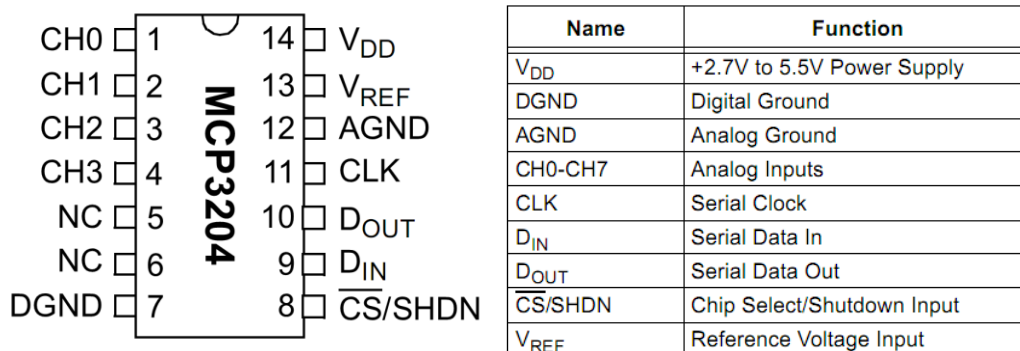
- Viết chương trình điều khiển các modul giao tiếp nối tiếp SPI, I2C, UART
- Kết nối vật lý raspberry với các ic ngoại vi như: Arduino, ADC MCP3204, modul thời gian thực DS3231

2. GIAO TIẾP SPI

2.3 Bài tập thực hành: Kết nối Raspberry Pi với bộ chuyển đổi tương tự số (ADC MCP3204) sử dụng giao tiếp SPI

Trong ví dụ này, chúng ta sẽ sử dụng Raspberry Pi để lấy dữ liệu hiệu điện thế trên biến trở. Vì raspberry pi không thể xử lý đầu vào tương tự, chúng tôi cần một bộ chuyển đổi A/D để chuyển đổi điện áp tương tự thành tín hiệu kỹ thuật số. Bạn có thể chọn mô-đun MCP3008, MCP3204 hoặc MCP3208 làm bộ chuyển đổi A / D. Trong bài thực hành này, chúng ta sẽ dùng MCP3204

<https://download.mikroe.com/documents/datasheets/MCP3204.pdf>

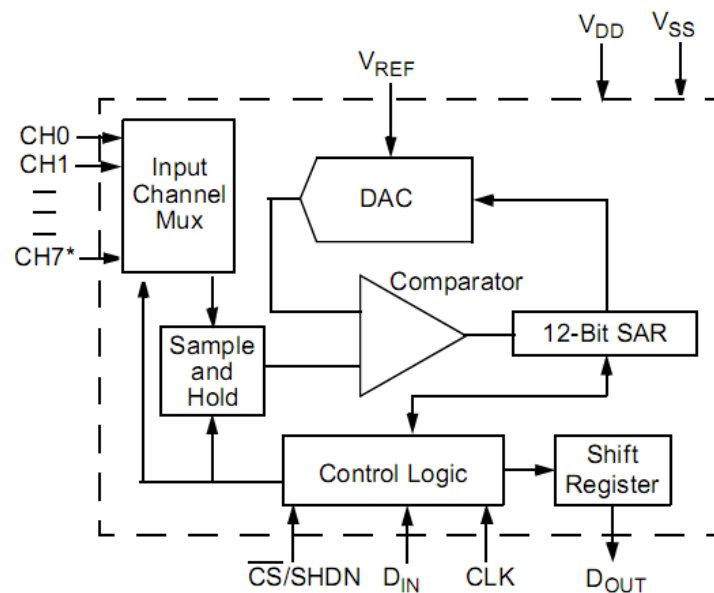


Hình 1: Pin map của MCP3204

Một vài thông số của MCP3204

12-bit resolution

- ± 1 LSB max DNL
- ± 1 LSB max INL (MCP3204/3208-B)
- ± 2 LSB max INL (MCP3204/3208-C)
- 4 (MCP3204) or 8 (MCP3208) input channels
- Analog inputs programmable as single-ended or pseudo-differential pairs
- On-chip sample and hold
- SPI serial interface (modes 0,0 and 1,1)
- Single supply operation: 2.7V - 5.5V
- 100 ksp/s max. sampling rate at $V_{DD} = 5V$
- 50 ksp/s max. sampling rate at $V_{DD} = 2.7V$
- Low power CMOS technology:
 - 500 nA typical standby current, 2 μA max.
 - 400 μA max. active current at 5V
- Industrial temp range: $-40^{\circ}C$ to $+85^{\circ}C$
- Available in PDIP, SOIC and TSSOP packages



* **Note:** Channels 5-7 available on MCP3208 Only

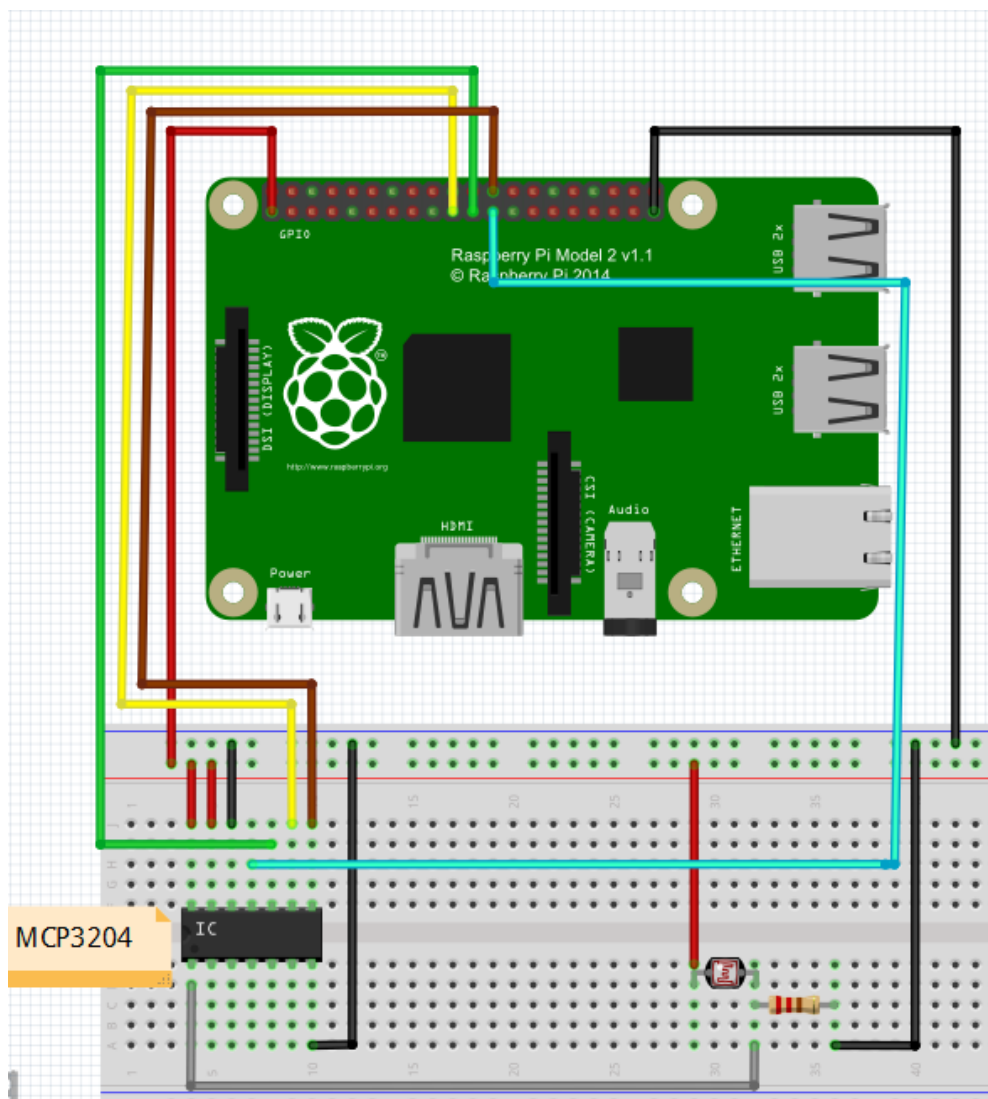
Hình 2: Sơ đồ khối của MCP3204

2.3.1 Thiết lập phần cứng

Nối các chân MISO với MISO và MOSI với MOSI, **không phải MISO với MOSI**. Khi viết code chúng ta sẽ xử lý các chân MISO và MOSI tùy thuộc vào thiết bị được đặt làm Master hay Slave.

Lưu ý rằng đối với SPI, ta thường có một dây tín hiệu được kết nối với CS (Chip Select) hoặc SS (Slave Select). Điều này rất hữu ích để chọn Slave đang cần truyền/nhận. Ở đây, vì chỉ có một Slave là Arduino, nên không cần dây này, giao tiếp vẫn hoạt động.

Ngoài ra, cần chú ý: **Raspberry Pi hoạt động ở 3.3V, trong khi cấp nguồn cho MCP3204 dùng chân 3.3V**. Đối với ví dụ này, chúng ta kết nối trực tiếp như hình 3 **không cần sử dụng bộ chuyển đổi mức logic (3.3V - 5V)**.



Hình 3: Ghép nối MCP3204

2.3.2 Thiết lập phần mềm

Hàm đọc giá trị ADC

```
int read_mcp3208_adc(unsigned char adcChannel)
{
    unsigned char buff[3];
    int adcValue = 0;

    buff[0] = 0x06 | ((adcChannel & 0x07) >> 7);
    buff[1] = ((adcChannel & 0x07) << 6);
    buff[2] = 0x00;

    digitalWrite(CS_MCP3208, 0); // Low : CS Active

    wiringPiSPIDataRW(SPI_CHANNEL, buff, 3);

    buff[1] = 0x0F & buff[1];
    adcValue = ( buff[1] << 8) | buff[2];

    digitalWrite(CS_MCP3208, 1); // High : CS Inactive

    return adcValue;
}
```

Hàm main

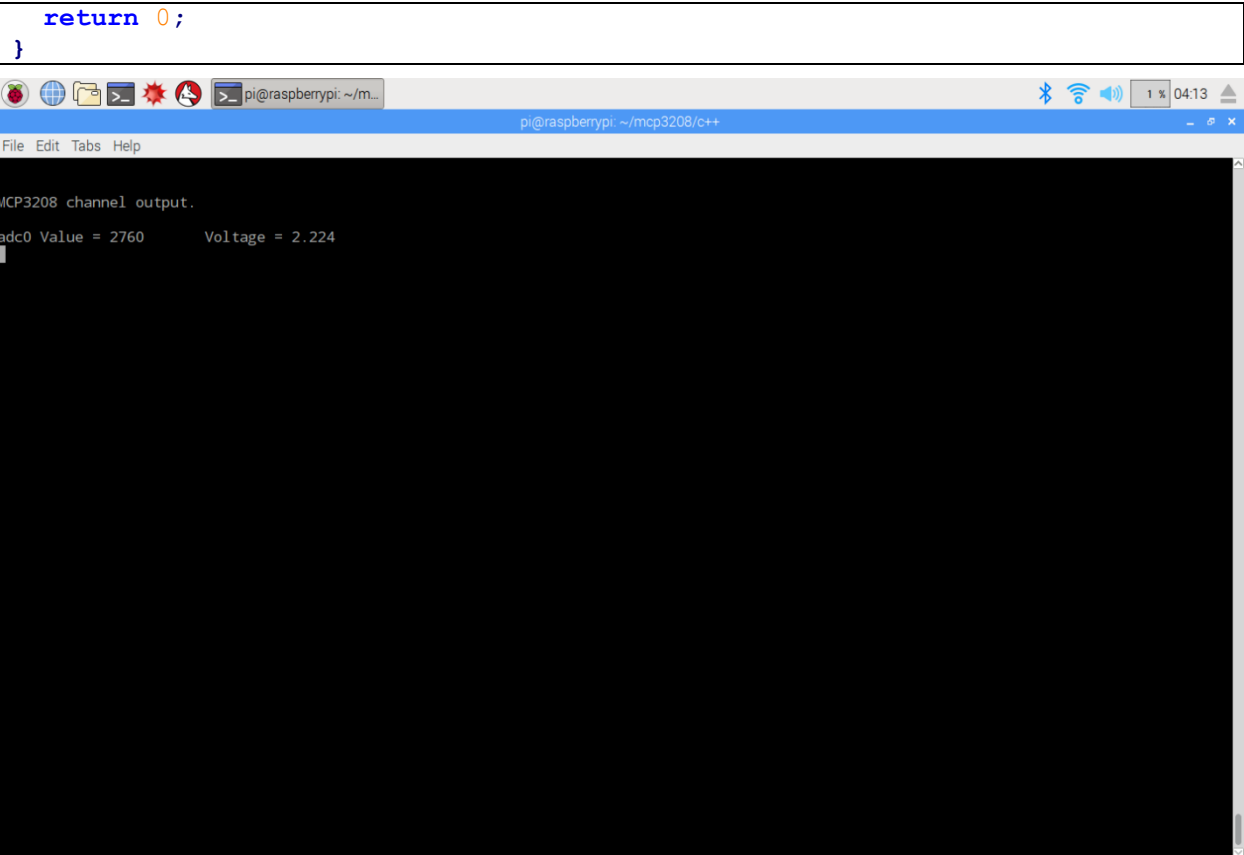
```
int main (void)
{
    int adc1Channel = 0;
    int adc1Value = 0;

    if(wiringPiSetup() == -1)
    {
        fprintf (stdout, "Unable to start wiringPi: %s\n", strerror(errno));
        return 1 ;
    }
    if(wiringPiSPISetup(SPI_CHANNEL, SPI_SPEED) == -1)
    {
        fprintf (stdout, "wiringPiSPISetup Failed: %s\n", strerror(errno));
        return 1 ;
    }

    pinMode(CS_MCP3208, OUTPUT);

    while(1)
    {
        system("clear");
        printf("\n\nMCP3208 channel output.\n\n");
        adc1Value = read_mcp3208_adc(adc1Channel);
        printf("adc0 Value = %04u", adc1Value);
        printf("\tVoltage = %.3f\n", ((3.3/4096) * adc1Value));
        usleep(1000000);
    }
}
```

```
return 0;
}
```



The image shows a terminal window on a Raspberry Pi. The window title is "pi@raspberrypi: ~/mcp3208/c++". The terminal output displays "MCP3208 channel output." followed by "adc0 Value = 2760 Voltage = 2.224". The terminal has a black background and white text. The window's top bar includes standard Linux window controls and system status icons like Bluetooth, Wi-Fi, and the time 04:13.

pi@raspberrypi: ~/mcp3208/c++

MCP3208 channel output.

adc0 Value = 2760 Voltage = 2.224