Measuring Altitude using BME680

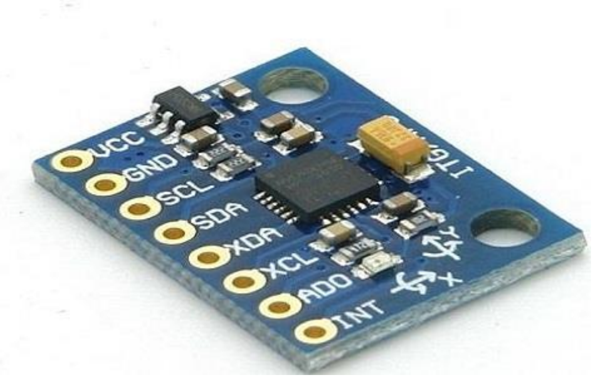
Asser Soliman

Agenda

- Introduction To BME680
- Pressure Sensor Working principle
- Getting Altitude From Pressure
- Communication Protocols Supported by BME680
- BME680 with Microcontroller

What is Sensor ?

- Sensor is a device that detect events or changes in environment and send this information
- There is a lot of sensors that are used in today's world
- Sensors are used in autonomous cars , smart homes and cities



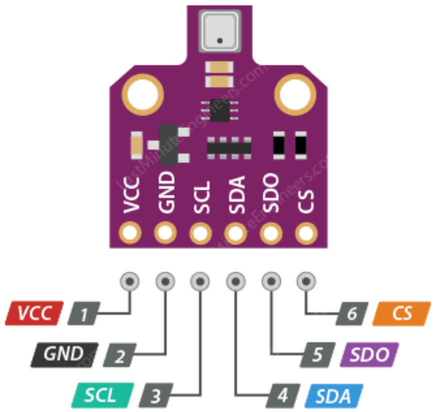
Gyroscope Sensor

Motion Detection Sensor

Introduction to BME680

Introduction to BME680

- BME680 is a 4 in 1 environmental sensor that was developed by Bosch Sensortec.
- It can Measure temperature , humidity , air quality and pressure.
- It has a low power consumption and high accuracy which makes it ideal for various applications.
- Some of it is applications are home automation and control and GPS enhancement.



Introduction to BME680

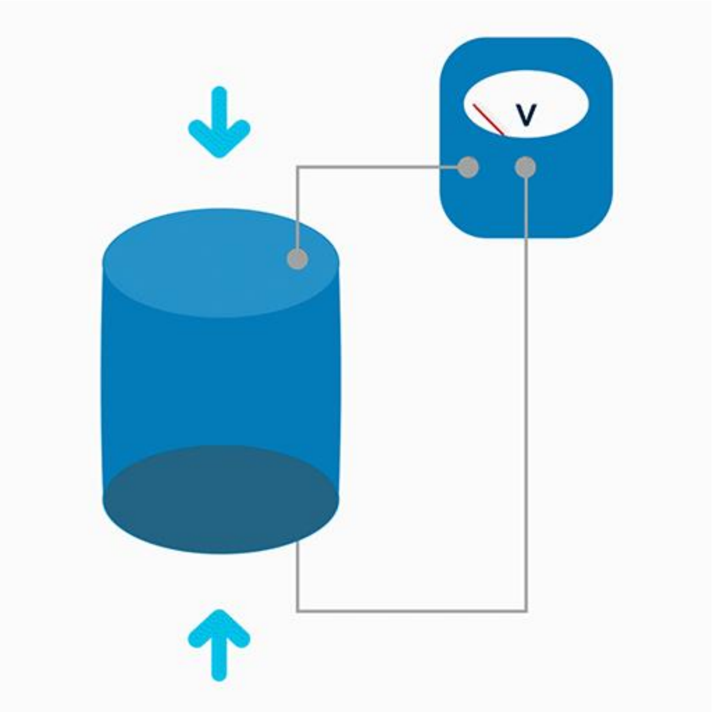
Pressure Sensor Working principle

Pressure Sensor Working principle

- The pressure sensor is a piezoresistive sensor, which changes resistance based on the pressure exerted on it.

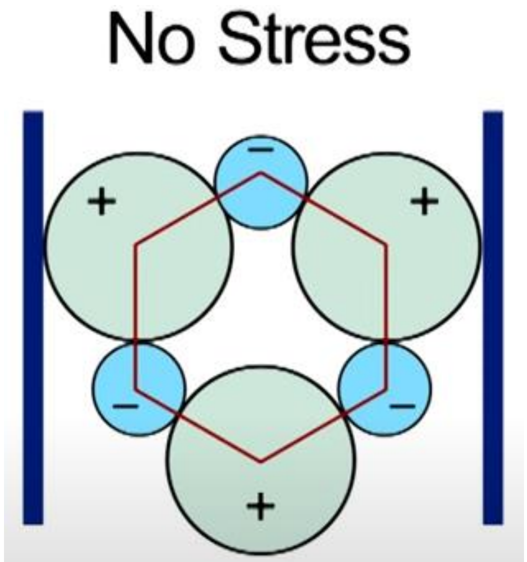
Pressure Sensor Working principle

- The pressure sensor is a piezoresistive sensor, which changes resistance based on the pressure exerted on it.

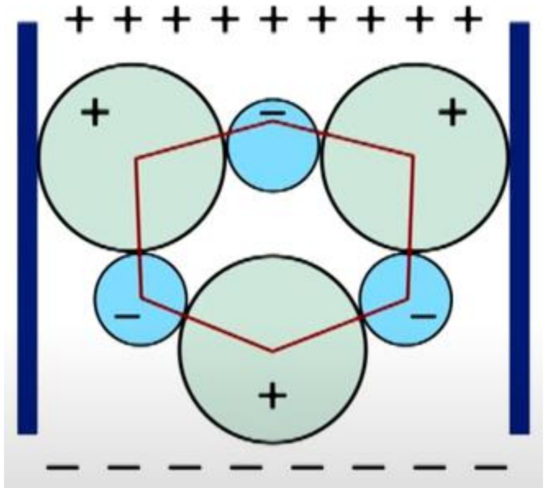


Pressure Sensor Working principle

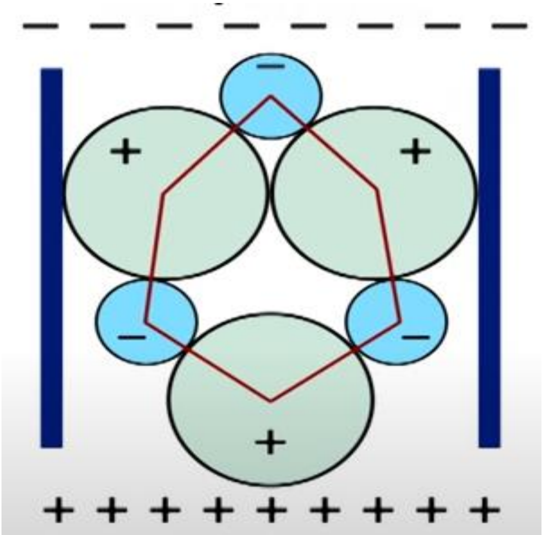
- Piezoresistive sensors are doped with silicon or germanium.



Stretched

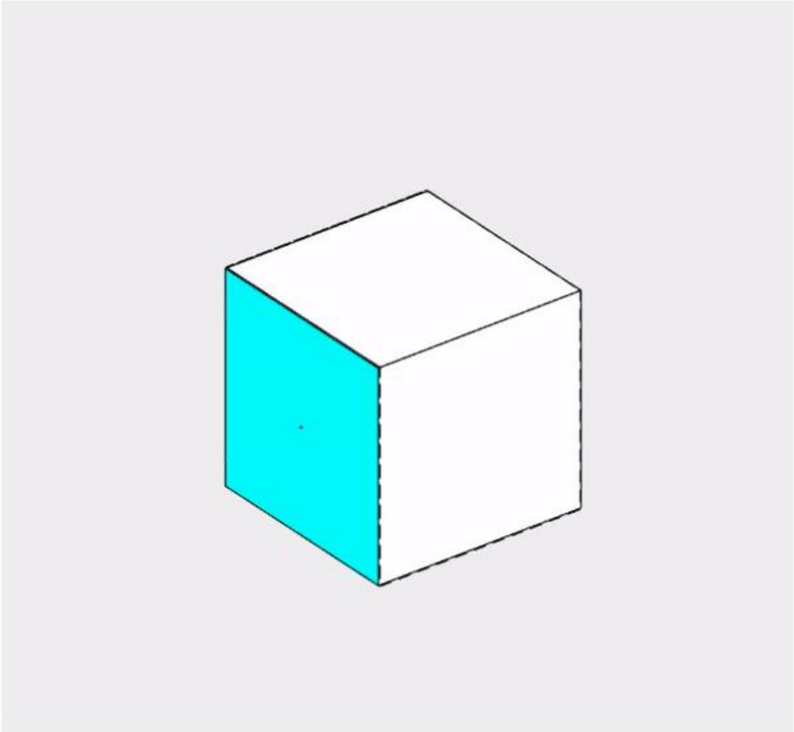


Compressed



Pressure Sensor Working principle

- Also, when pressure is exerted on the material its dimensions changes



Pressure Sensor Working principle

- Stress is how much a material resists being deformed when a force is applied to it

$$\sigma = \frac{F}{A}$$

where

- σ stress [Pa]
- F applied force [N]
- A cross-sectional area [m^2]

$$\sigma = E.\varepsilon$$

where

- σ is stress [Pa]
- ε is strain $= \frac{\Delta L}{L_0}$
- E is the modulus of elasticity [Pa]

Pressure Sensor Working principle

- Mechanical strain refers to the amount of deformation experienced by the body .
- Poisson's ratio is the negative ratio of lateral strain to longitudinal strain

$$\varepsilon = \frac{\Delta L}{L_0}$$

where

- ε strain
- ΔL total elongation [m]
- L_0 original length [m]

$$v = -\frac{\varepsilon_{lat}}{\varepsilon_{long}}$$

Pressure Sensor Working principle

Diameter $D = 0.1 m$, Resistivity $\rho = 1.68 \times 10^{-6}\ \Omega.m$, stiffness $E = 110\ GPa$

$$A = \pi r^2 L = \pi \times 0.05^2 \times 1 = 7.85 \times 10^{-3}\ m^2$$ Area

$$R = \rho \frac{L}{A} = 1.68 \times 10^{-6} \times \frac{1}{7.85 \times 10^{-3}} = 2.140127389 \times 10^{-4}$$ Resistance

Pressure Sensor Working principle

Diameter, Resistivity, stiffness, Area

$D = 0.1\,m$, $\rho = 1.68 \times 10^{-6}\,\Omega.m$, $E = 110\ GPa$, $A = 7.85 \times 10^{-3}\,m^2$

$$\sigma = \frac{F}{A} = \frac{1.571}{7.85 \times 10^{-3}} = 200.13\ Pa$$ stress

$$\sigma = E\underbrace{\epsilon}_{\text{strain}} , \quad \epsilon = \frac{\sigma}{E} = \frac{200.13}{110 \times 10^{6}} = 1.82 \times 10^{-6}$$ strain

$$\epsilon = \frac{\Delta L}{L} , \quad \Delta L = L\epsilon = 1 \times 1.82 \times 10^{-6} = 1.82 \times 10^{-6}$$ diff in length

$$R = \rho\frac{(L+\Delta L)}{A} = 1.68 \times 10^{-6} \cdot \frac{1+1.82 \times 10^{-6}}{7.85 \times 10^{-3}} = 2.14013128 \times 10^{-4}\,\Omega$$ Resistance

Pressure Sensor Working principle

Diameter, Resistivity, stiffness, Area, Poisson ratio

$$D = 0.1\,m,\quad \rho = 1.68\times10^{-6}\,\Omega.m,\quad E = 110\,GPa,\quad A = 7.85\times10^{-3}\,m^2,\quad \gamma = 0.364$$

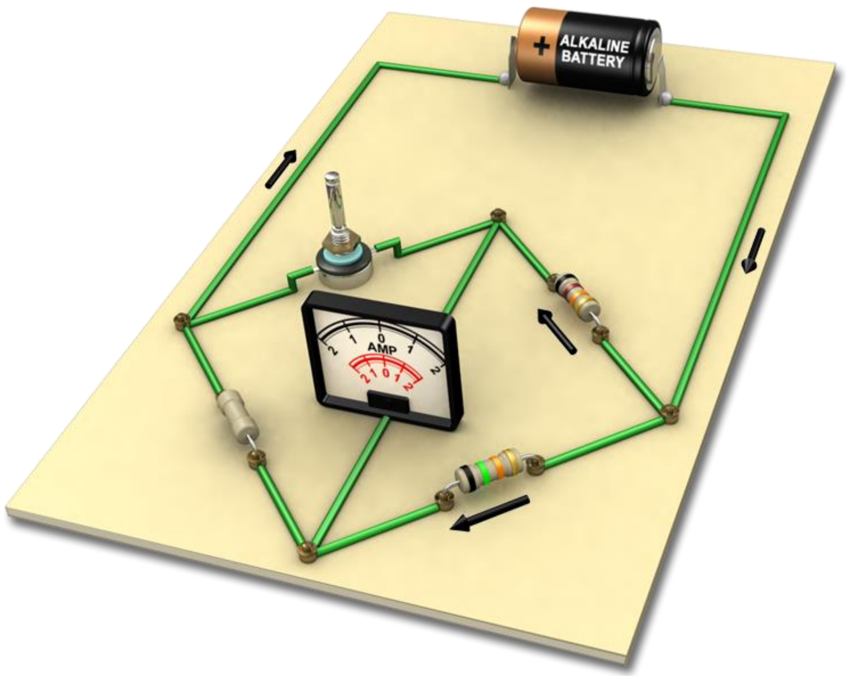
$$\Delta L = 1.82\times10^{-6}$$

$$\gamma = \frac{\text{lateral strain}}{\text{longtinudal strain}} = 0.364,\quad \frac{\text{lateral}}{1.82\times10^{-6}} = 0.364,\quad \text{lateral} = 6.6248\times10^{-7}$$

$$R = \rho\,\frac{L+\Delta L}{A-\Delta A} = 1.68\times10^{-6}\cdot\frac{1+1.82\times10^{-6}}{7.85\times10^{-3} - 6.6248\times10^{-7}} = 2.14031191\times10^{-4}\,\Omega$$

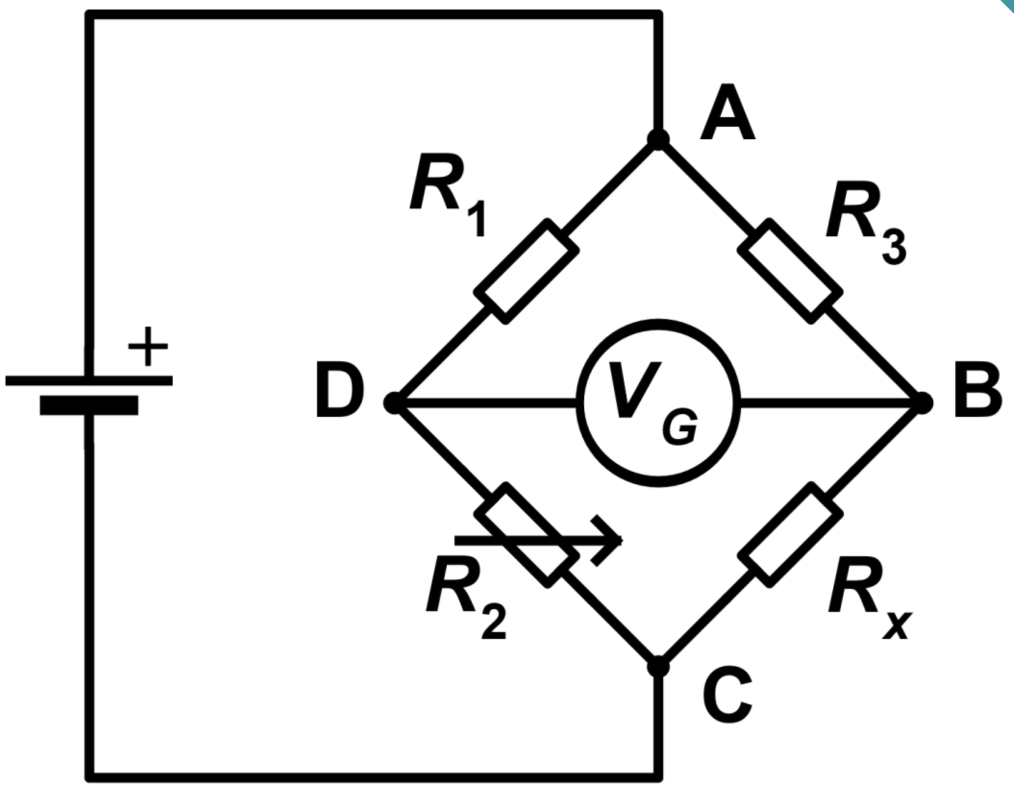
Pressure Sensor Working principle

- The Wheatstone bridge is used to sense the changes in pressure sensor
- The Wheatstone Bridge is an electrical circuit used to measure an unknown electrical resistance



Pressure Sensor Working principle

We have the values of R1 R2 R3
We want to get the value of Rx

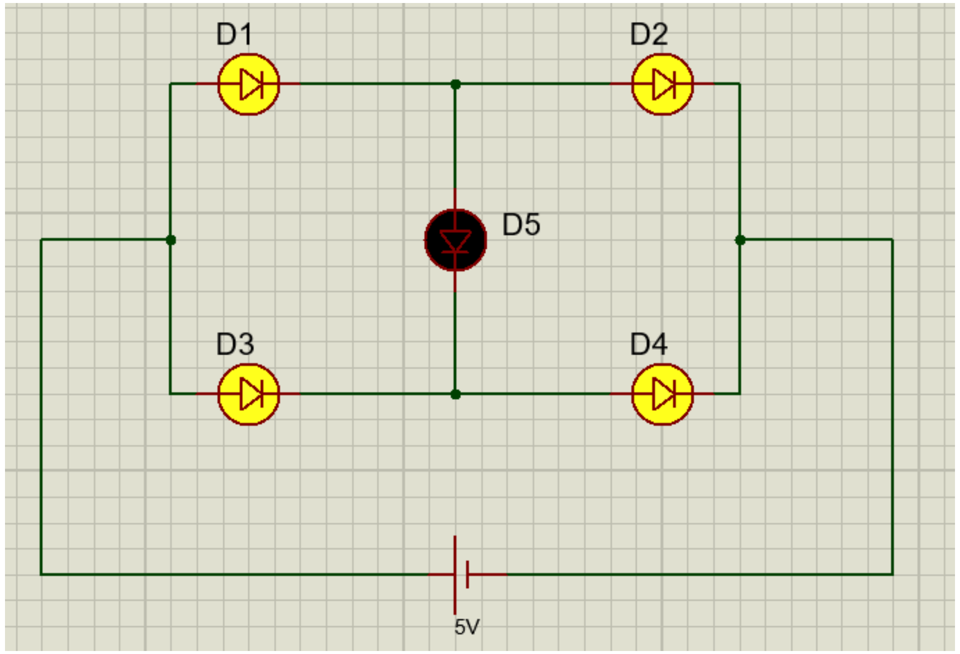


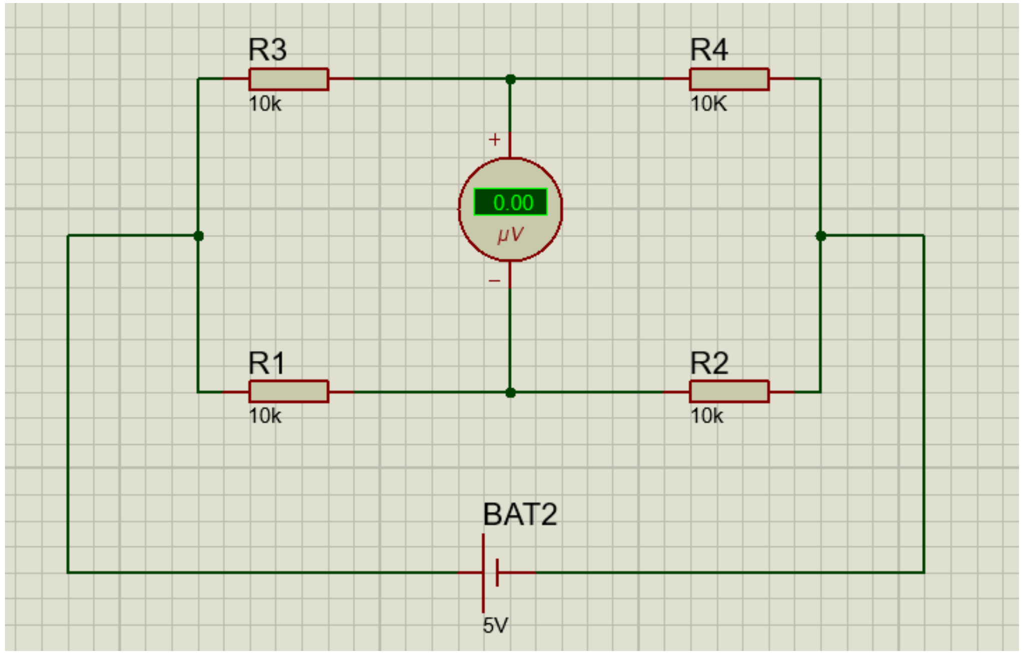
Pressure Sensor Working principle

The Wheatstone bridge is used because

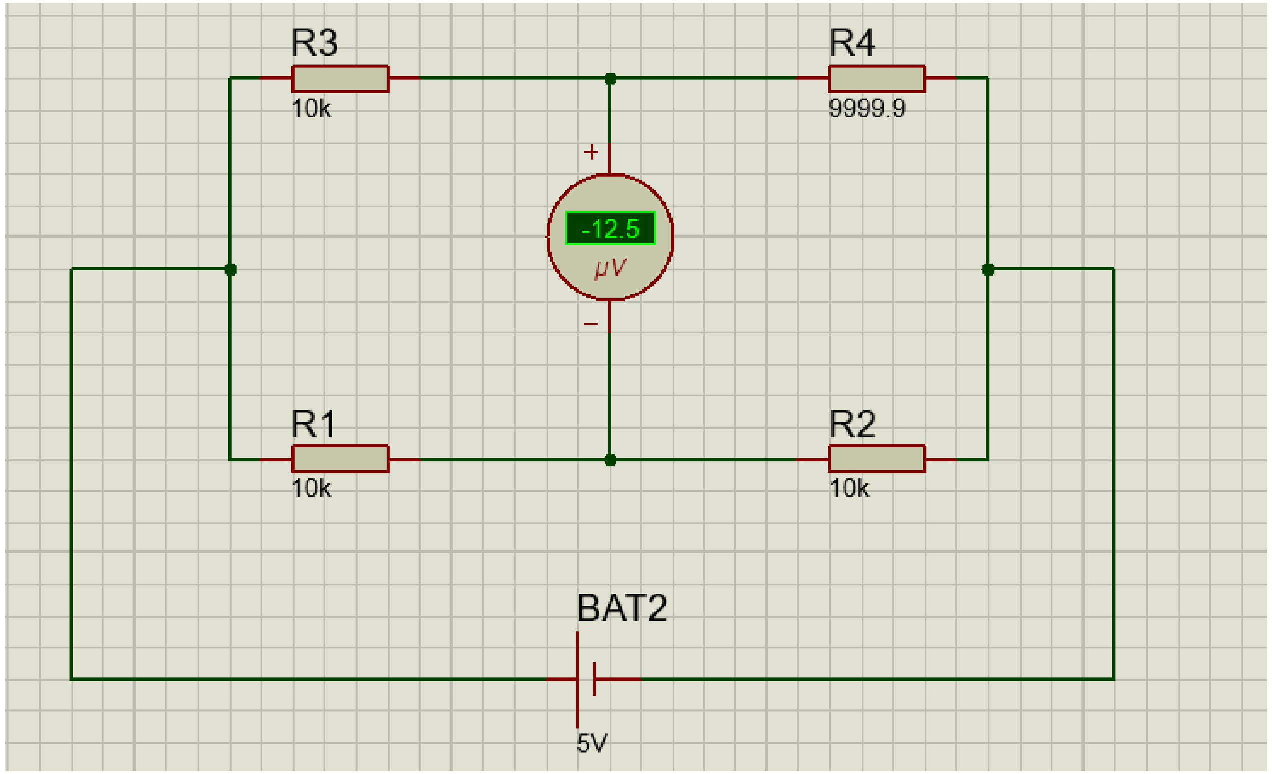
- It is ideal for detecting small changes in resistors.
- Temperature compensation
- It also convert small changes in resistance into voltage.

Pressure Sensor Working principle



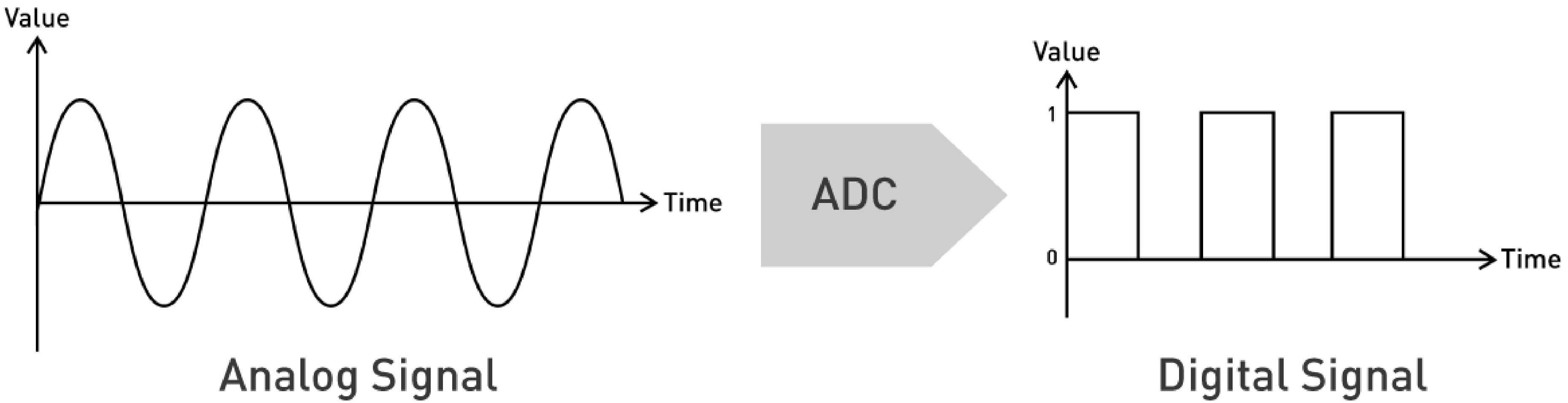


Pressure Sensor Working principle



Pressure Sensor Working principle

- Since the output of the Wheatstone bridge is analog, we have to change it to digital so that our computer can understand it .
- For this reason, Digital to Analog converter is used

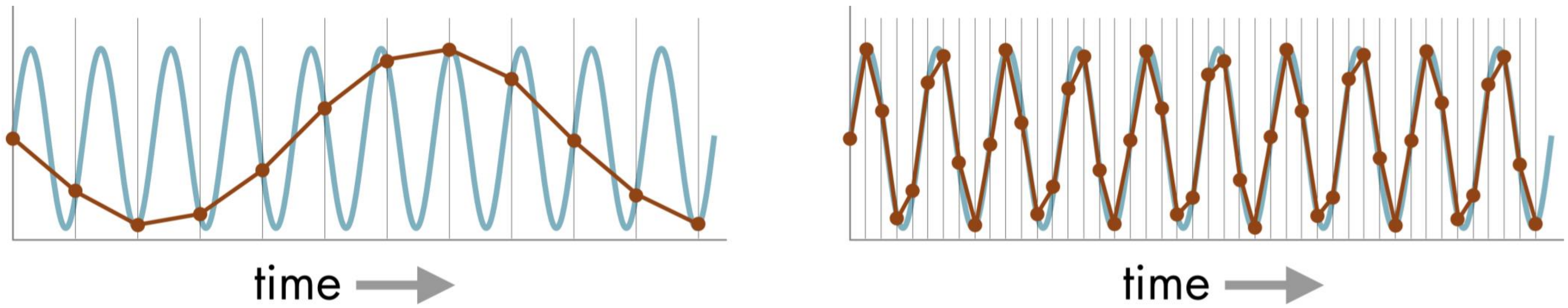


Pressure Sensor Working principle

- The process of converting the signal from analog to signal consists of two major step:
  1. Sampling
  2. Quantization

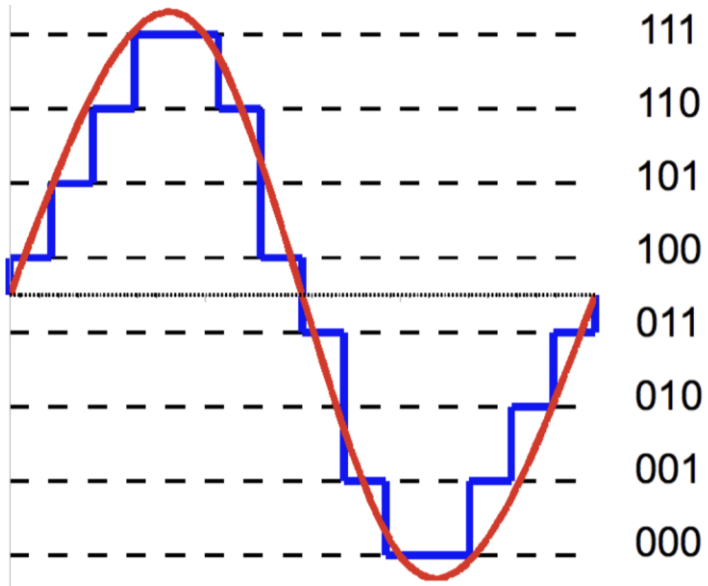
Pressure Sensor Working principle

- Sampling : is the process of transforming the signal from continuous time to discrete time by taking samples at specific rate.
- The determination of the rate must follow Nyquist theorem which states that the sampling frequency must be at least twice the highest frequency.



Pressure Sensor Working principle

- Quantization : is the process of transforming the signal from continuous values to discrete values
- Each sampled value is approximated to the nearest value within a set range of discrete levels.
- The number of these is determined by the resolution of ADC

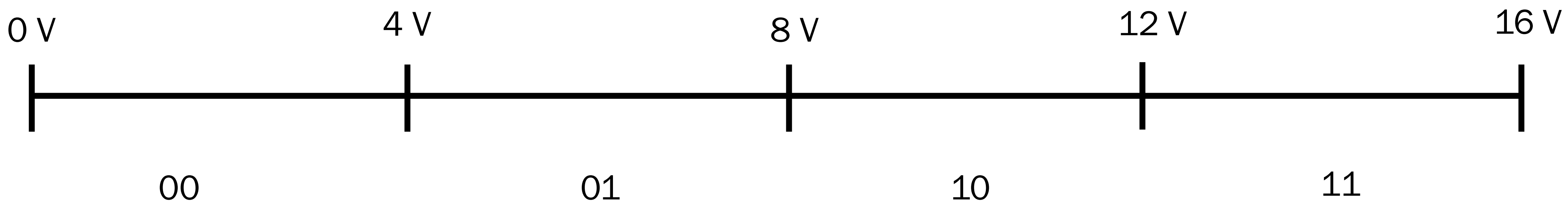


Pressure Sensor Working principle

- Let's say we have an ADC with 2 bits resolution , Vmax = 16 , Vmin = 0.

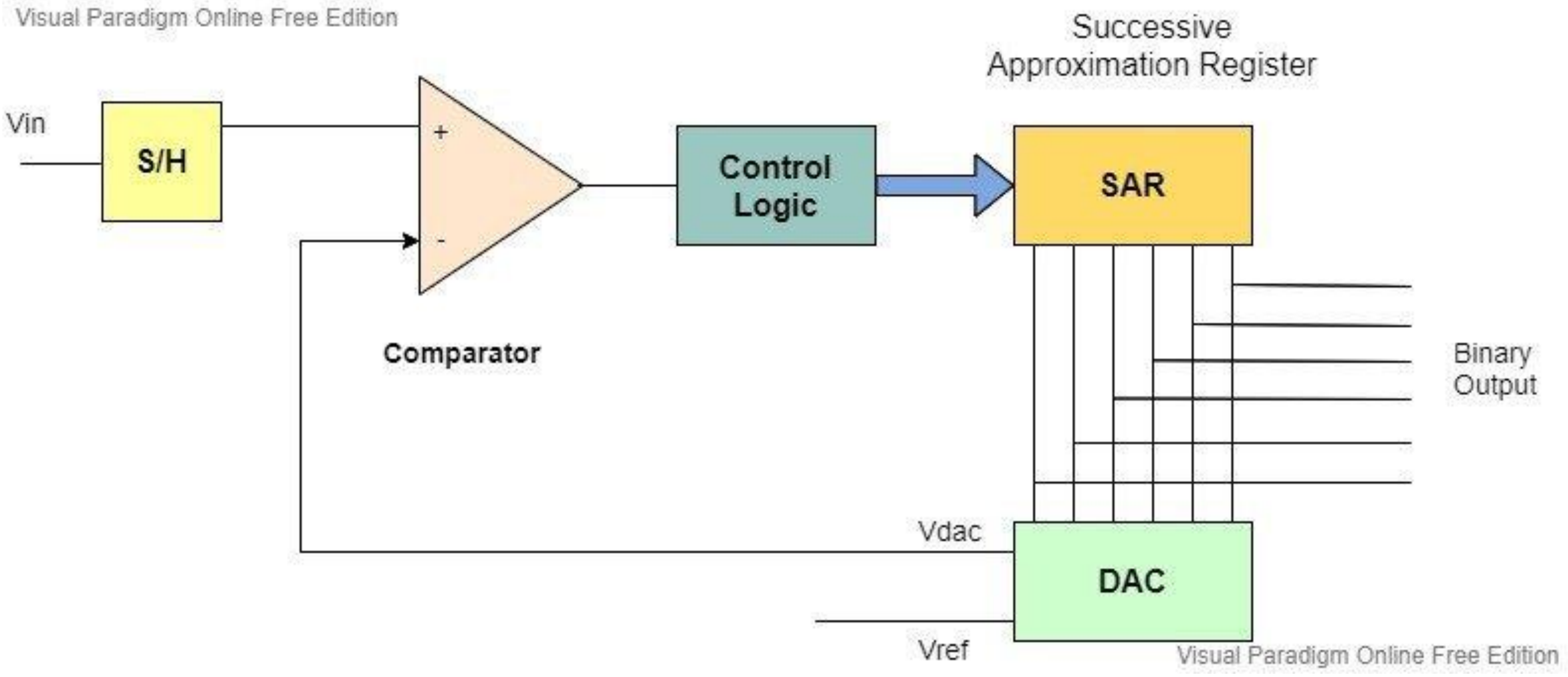
We can deduce that we will have 2^2 = 4 intervals.

Each interval will be of size $\frac{16-0}{4}$ = 4 volt



Pressure Sensor Working principle

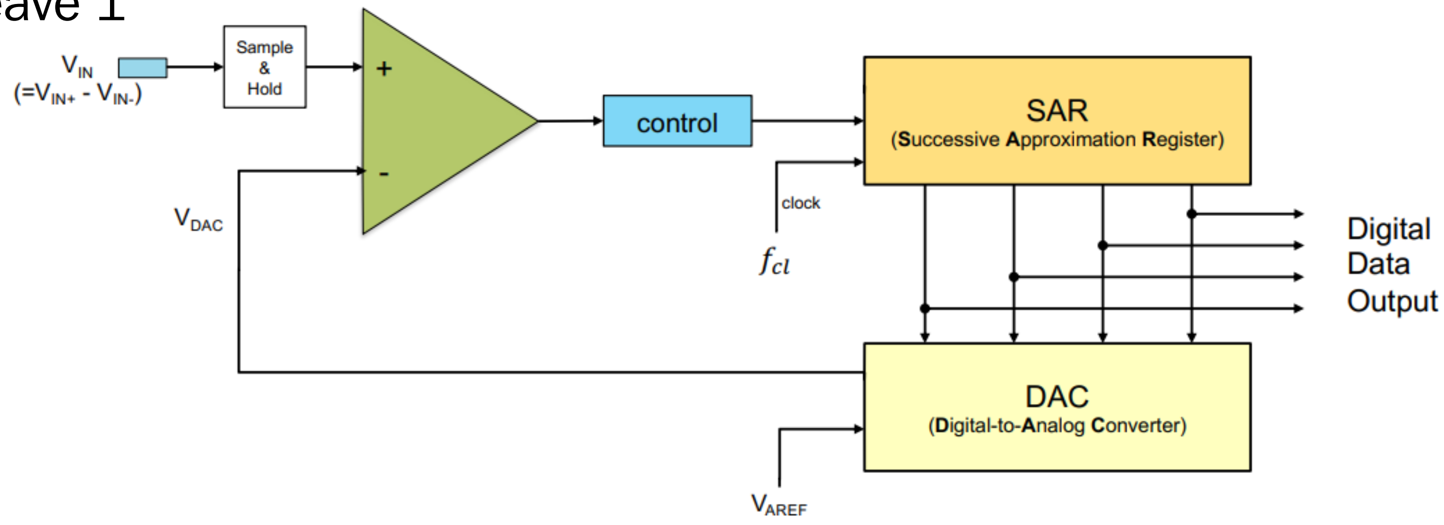
- The most used technique in ADC is Successive Approximation Method



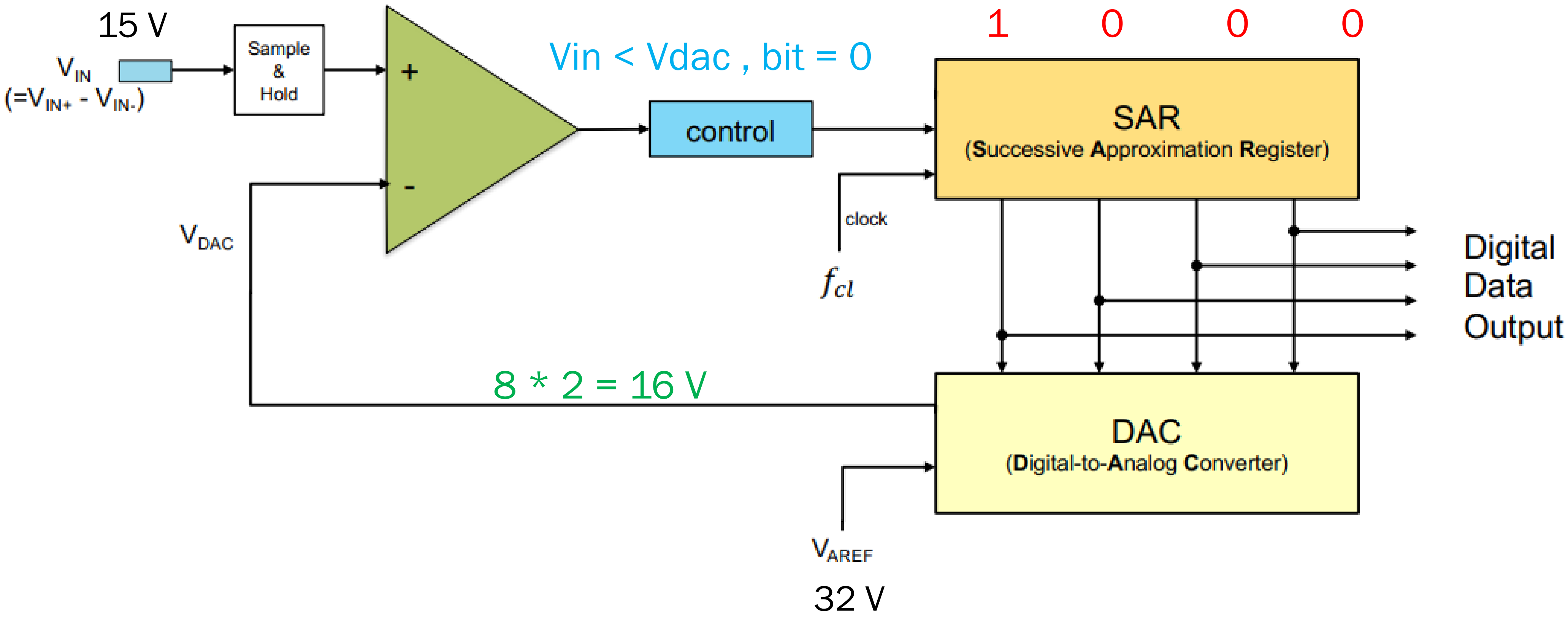
Pressure Sensor Working principle

How does Successive Approximation method work

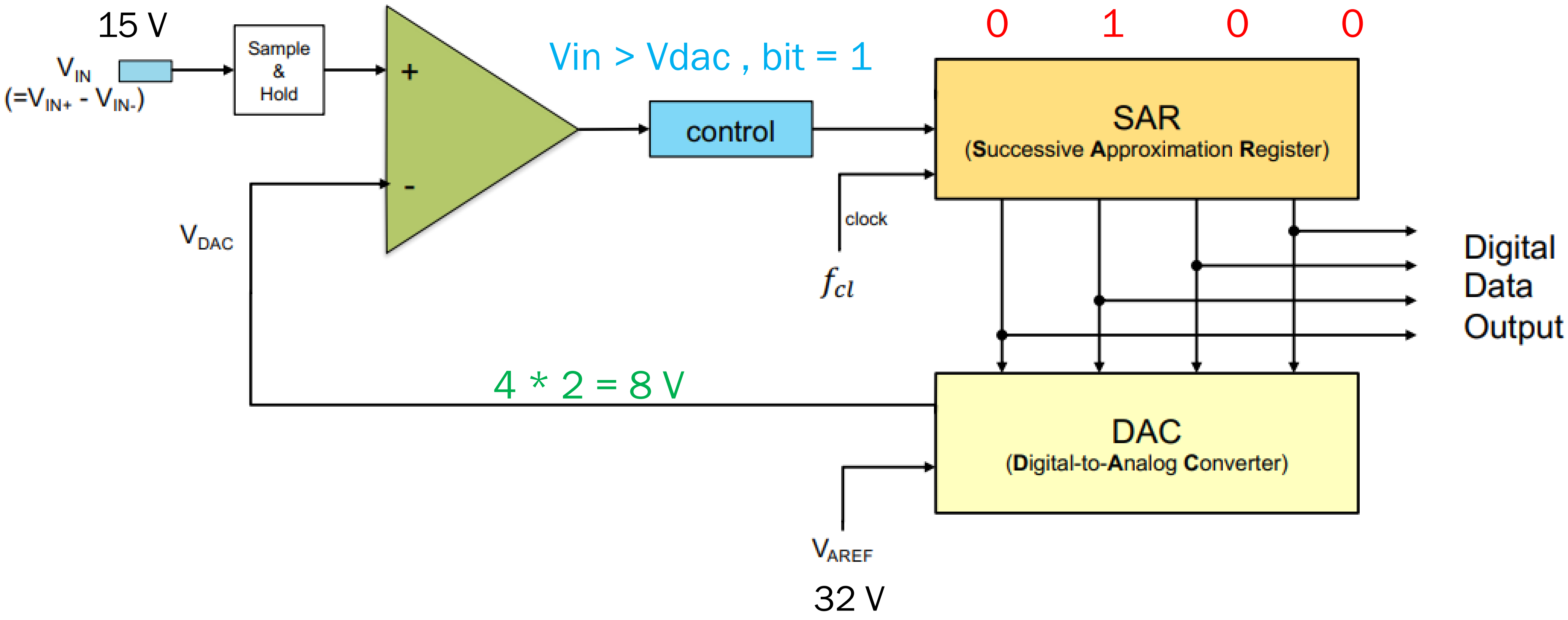
1. Set the most significant bit
2. Compare generated voltage with sample voltage
3. If Vin < Vgen reset bit otherwise leave 1
4. Move to next significant bit
5. repeat



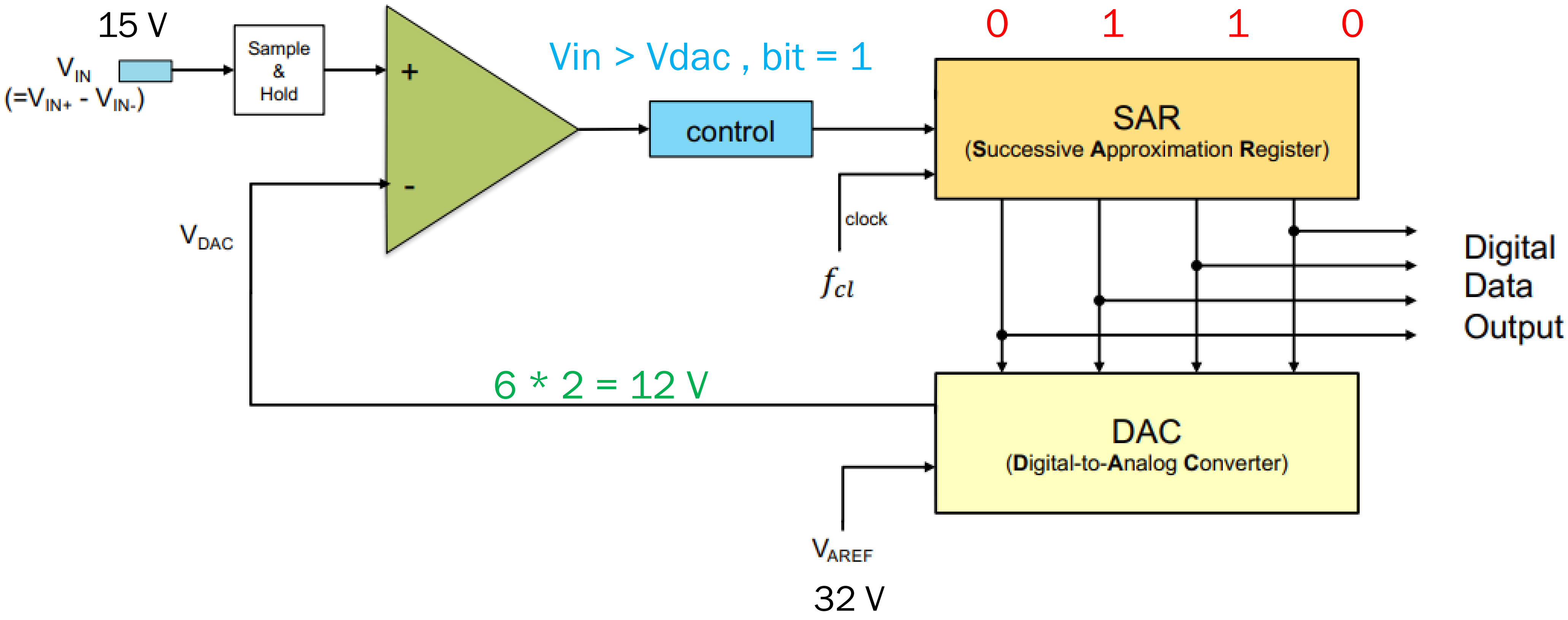
Pressure Sensor Working principle



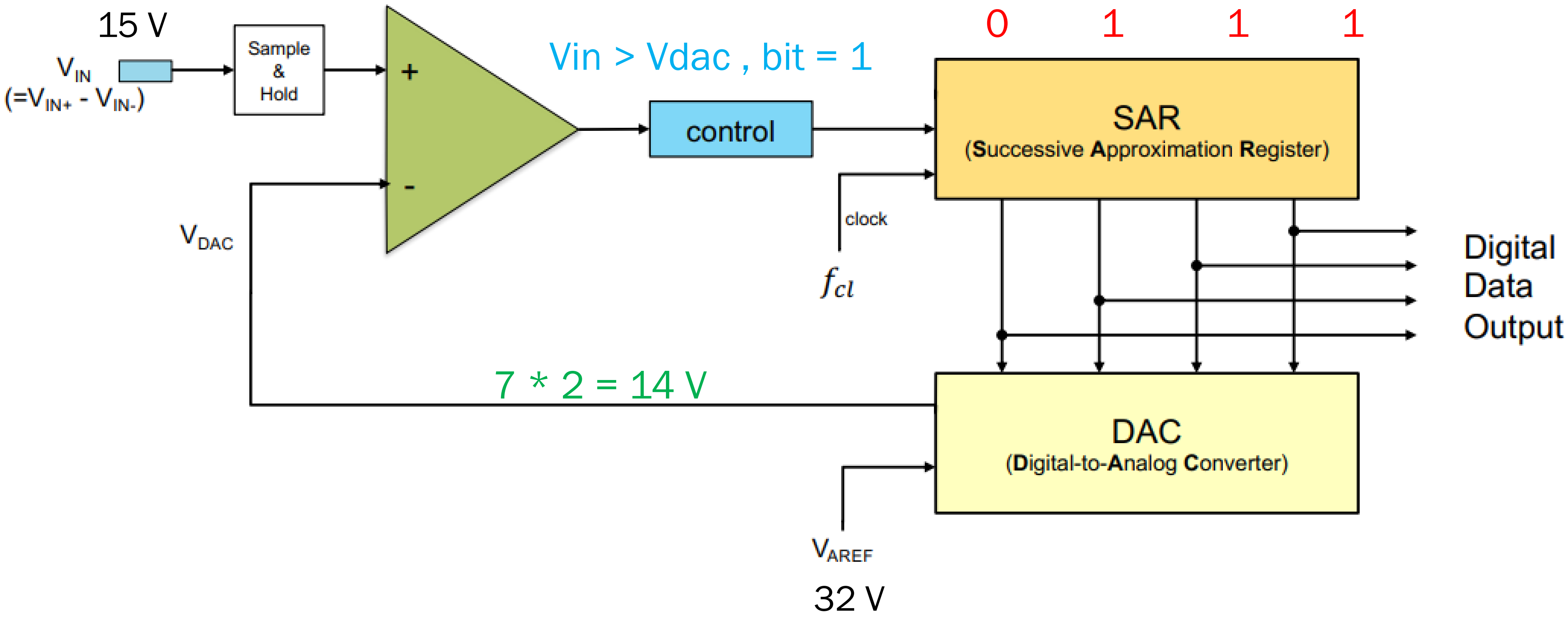
Pressure Sensor Working principle



Pressure Sensor Working principle



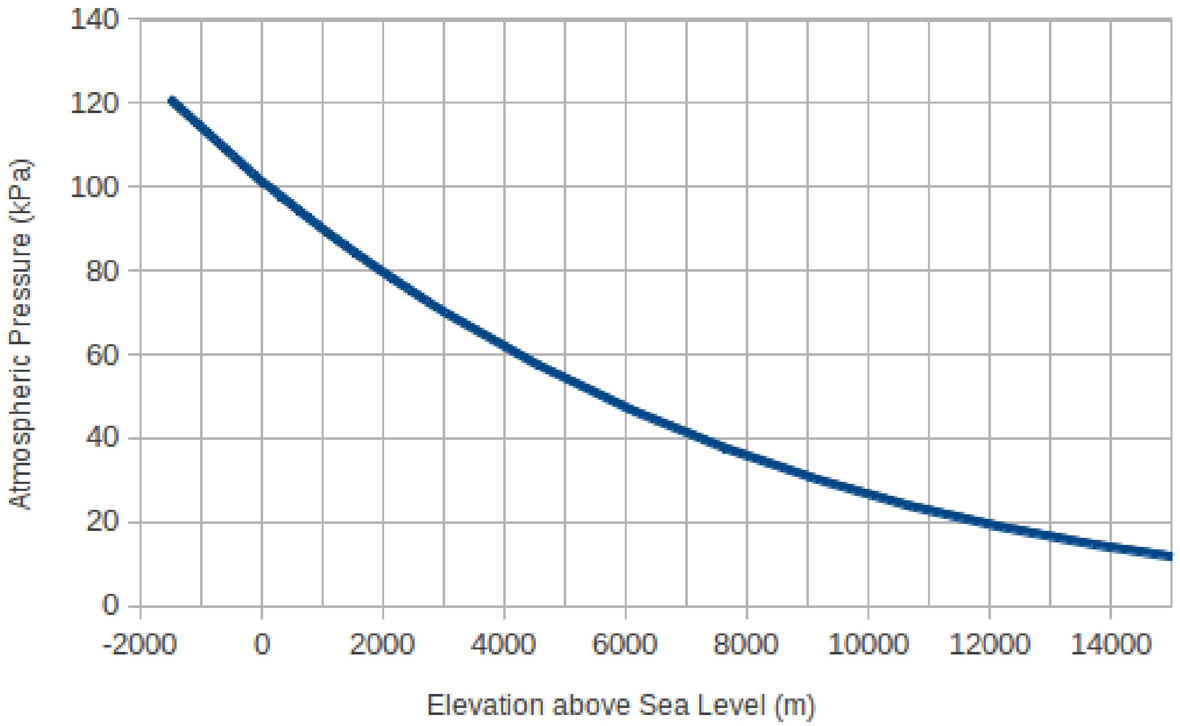
Pressure Sensor Working principle



Getting Altitude From Pressure

Getting Altitude From Pressure

- Altitude refers to the measurement of height or elevation of point with relation to sea level.



Pressure is inversely proportional with Altitude

Getting Altitude From Pressure

- This is the equation that connects altitude with Pressure

$$\mathrm{H} = \left(\frac{T0}{L}\right) * \left(1 - \frac{P}{P0}^{\frac{R*L}{g*M}}\right)$$

Getting Altitude From Pressure

- H : altitude

$$\mathrm{H} = \left(\frac{T0}{L}\right) * \left(1 - \frac{P}{P0}^{\frac{R*L}{g*M}}\right)$$

Getting Altitude From Pressure

- H : altitude
- T0 : standard temperature at sea level (288.15 K)

$$\mathrm{H} = \left(\frac{T0}{L}\right) * (1 - \frac{P}{P0}^{\frac{R*L}{g*M}})$$

Getting Altitude From Pressure

- H : altitude
- T0 : standard temperature at sea level (288.15 K)
- L : standard temperature lapse (0.0065 K/m)

$$\mathrm{H} = \left(\frac{T0}{L}\right) * \left(1 - \frac{P}{P0}^{\frac{R*L}{g*M}}\right)$$

Getting Altitude From Pressure

- H : altitude
- T0 : standard temperature at sea level (288.15 K)
- L : standard temperature lapse (0.0065 K/m)
- P : Atmospheric pressure at the altitude h

$$\mathrm{H} = \left(\frac{T0}{L}\right) * \left(1 - \frac{P}{P0}^{\frac{R*L}{g*M}}\right)$$

Getting Altitude From Pressure

- H : altitude
- T0 : standard temperature at sea level (288.15 K)
- L : standard temperature lapse (0.0065 K/m)
- P : Atmospheric pressure at the altitude h
- P0 : Atmospheric pressure at the sea level (101325 Pa)

$$\mathrm{H} = \left(\frac{T0}{L}\right) * \left(1 - \frac{P}{P0}^{\frac{R*L}{g*M}}\right)$$

Getting Altitude From Pressure

- H : altitude
- T0 : standard temperature at sea level (288.15 K)
- L : standard temperature lapse (0.0065 K/m)
- P : Atmospheric pressure at the altitude h
- P0 : Atmospheric pressure at the sea level (101325 Pa)
- R : ideal gas constant (8.31447)

$$\mathrm{H} = \left(\frac{T0}{L}\right) * \left(1 - \frac{P}{P0}^{\frac{R * L}{g * M}}\right)$$

Getting Altitude From Pressure

- H : altitude
- T0 : standard temperature at sea level (288.15 K)
- L : standard temperature lapse (0.0065 K/m)
- P : Atmospheric pressure at the altitude h
- P0 : Atmospheric pressure at the sea level (101325 Pa)
- R : ideal gas constant (8.31447)
- g : gravitational acceleration (9.8 m/s^2)

$$\mathrm{H} = \left(\frac{T0}{L}\right) * \left(1 - \frac{P}{P0}^{\frac{R * L}{g * M}}\right)$$

Getting Altitude From Pressure

$$H = \left(\frac{T0}{L}\right) * \left(1 - \frac{P}{P0}^{\frac{R*L}{g*M}}\right)$$

- H : altitude
- T0 : standard temperature at sea level (288.15 K)
- L : standard temperature lapse (0.0065 K/m)
- P : Atmospheric pressure at the altitude h
- P0 : Atmospheric pressure at the sea level (101325 Pa)
- R : ideal gas constant (8.31447)
- g : gravitational acceleration (9.8 m/s^2)
- M : molar mass (0.0289644 kg/mol)

Getting Altitude From Pressure

- H : altitude
- T0 : standard temperature at sea level (288.15 K)
- L : standard temperature lapse (0.0065 K/m)
- P : Atmospheric pressure at the altitude h
- P0 : Atmospheric pressure at the sea level (101325 Pa)
- R : ideal gas constant (8.31447)
- g : gravitational acceleration (9.8 m/s^2)
- M : molar mass (0.0289644 kg/mol)

$$\mathrm{H} = 44330 \;*\; (1 - \frac{P}{101325}^{1/5.255})$$

Getting Altitude From Pressure

This equation is very effective in calculating altitude, but it has some limitations

$$\mathrm{H} = \left(\frac{T0}{L}\right) * \left(1 - \frac{P}{P0}^{\frac{R*L}{g*M}}\right)$$

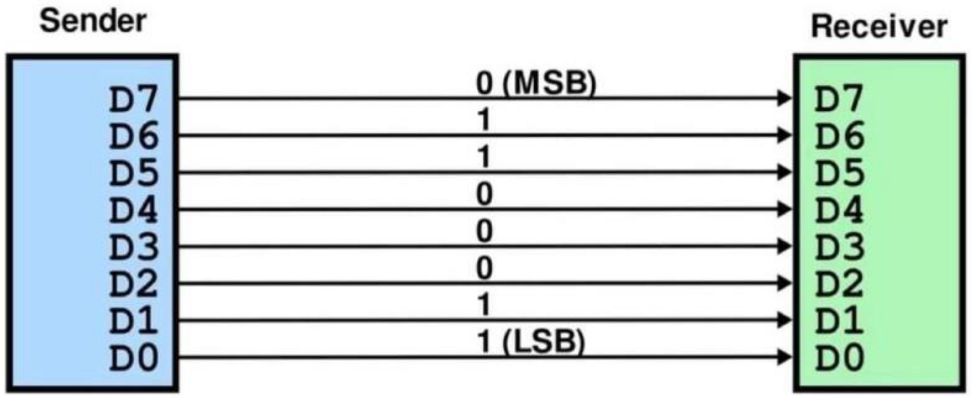
Communication Protocols Supported by BME680

Communication Protocols Supported by BME680

- Communications protocols are set of rules that allow two or more devices to transmit information.

Communication Protocols Supported by BME680

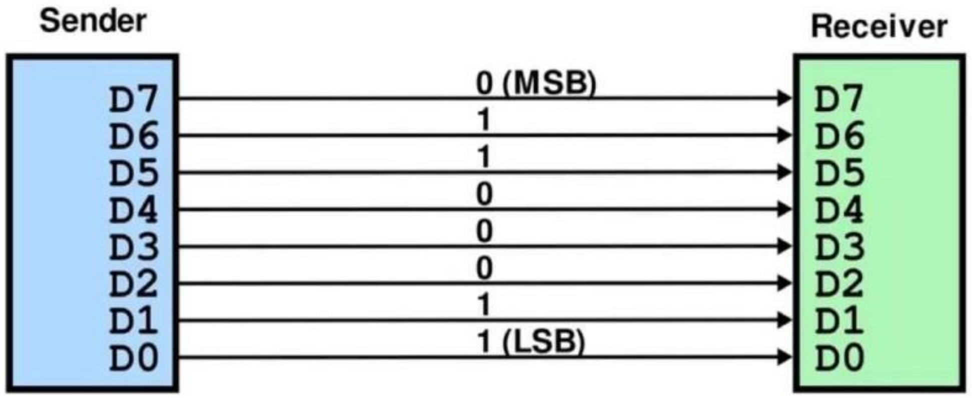
- Communications protocols are set of rules that allow devices to transmit information.



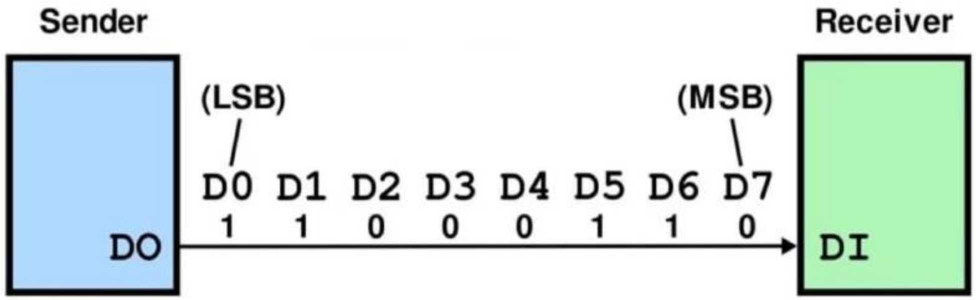
Parallel Communication

Communication Protocols Supported by BME680

- Communications protocols are set of rules that allow devices to transmit information.



Parallel Communication



Serial Communication

Communication Protocols Supported by BME680

- There are three communication modes

Simplex

Communication Protocols Supported by BME680

- There are three communication modes

Simplex

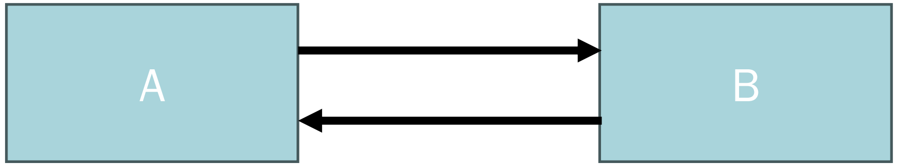
Half Duplex

Communication Protocols Supported by BME680

- There are three communication modes

Simplex

Half Duplex



Full Duplex

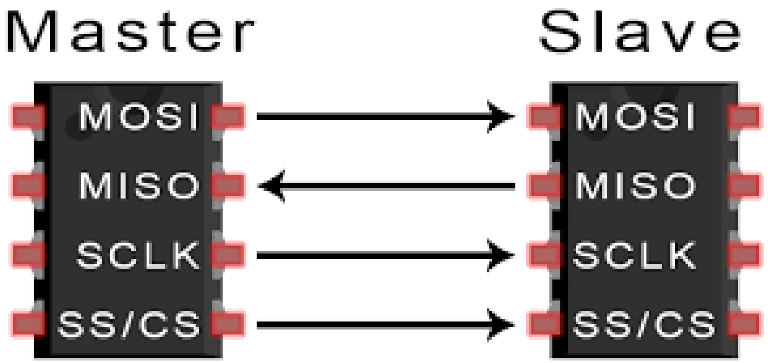
Communication Protocols Supported by BME680

- The SPI was developed by Motorola in 1980s.
- It is a synchronous serial protocol.
- It is used for short distance communication.
- Master Slave topology
- It can be used as Full duplex or Half duplex
- It has low power consumption and high transmission speed
- SPI has simpler hardware design than I2C and UART

Communication Protocols Supported by BME680

Full Duplex 4 pins (MOSI - MISO – CLK –SS)

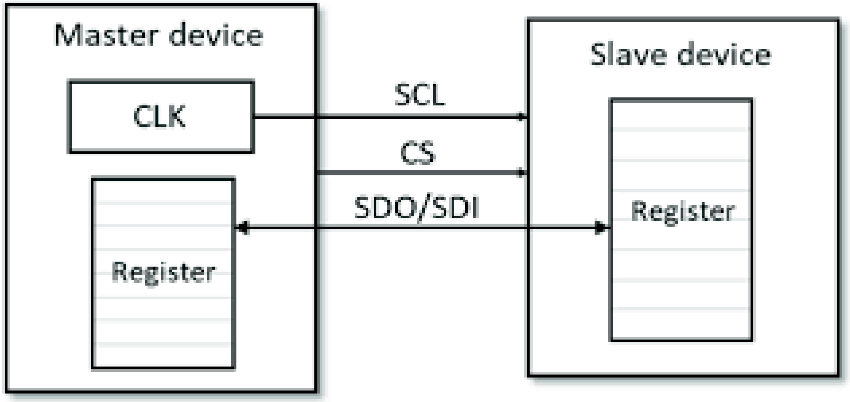
- **MOSI** : Master Output Slave Input
- **MISO**: Master Input Slave Output
- **CLK**: Serial Clock (Controlled by master)
- **SS**: Slave Selector (Requires more pins)



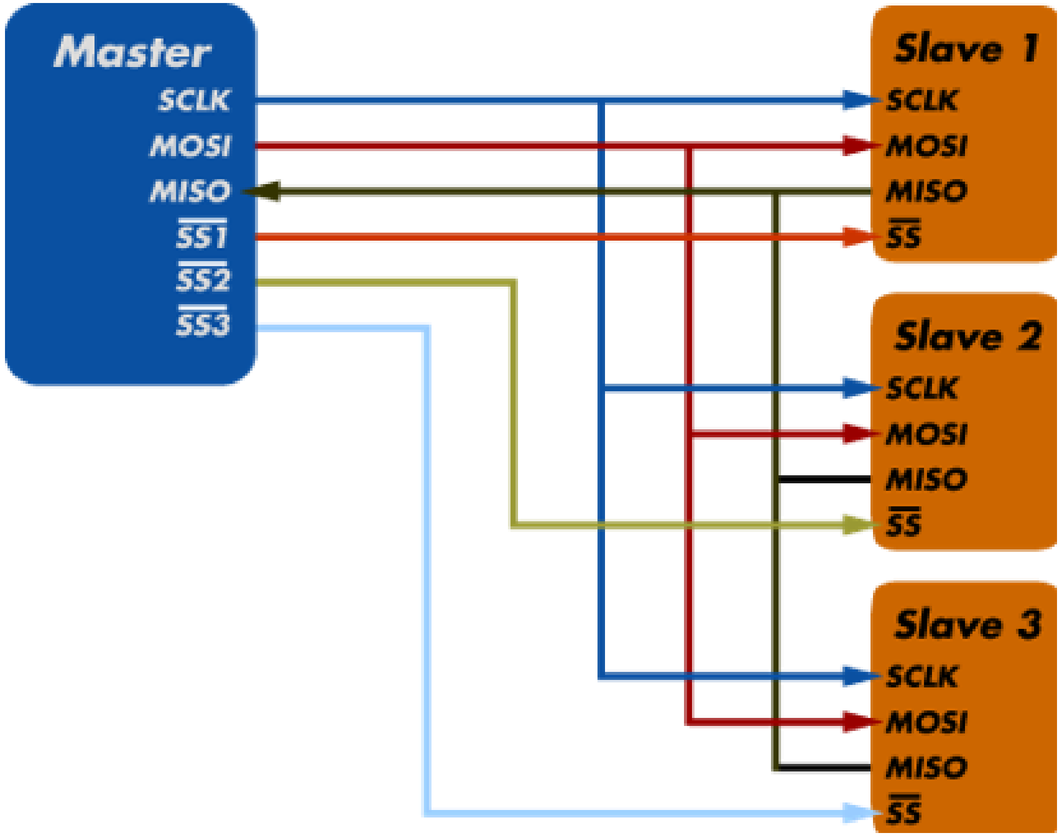
Communication Protocols Supported by BME680

Half Duplex 3 pins (Data I/O – CLK –SS)

- **Data I/O**: used for transmitting and receiving data
- **CLK**: Serial Clock (Controlled by master)
- **SS**: Slave Selector (Requires more pins)



Communication Protocols Supported by BME680



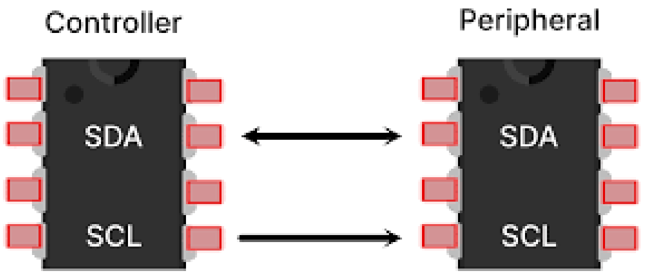
Communication Protocols Supported by BME680

0	1	0	0	0	0	0	1
A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0

0	1	0	0	0	1	1	1
B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0

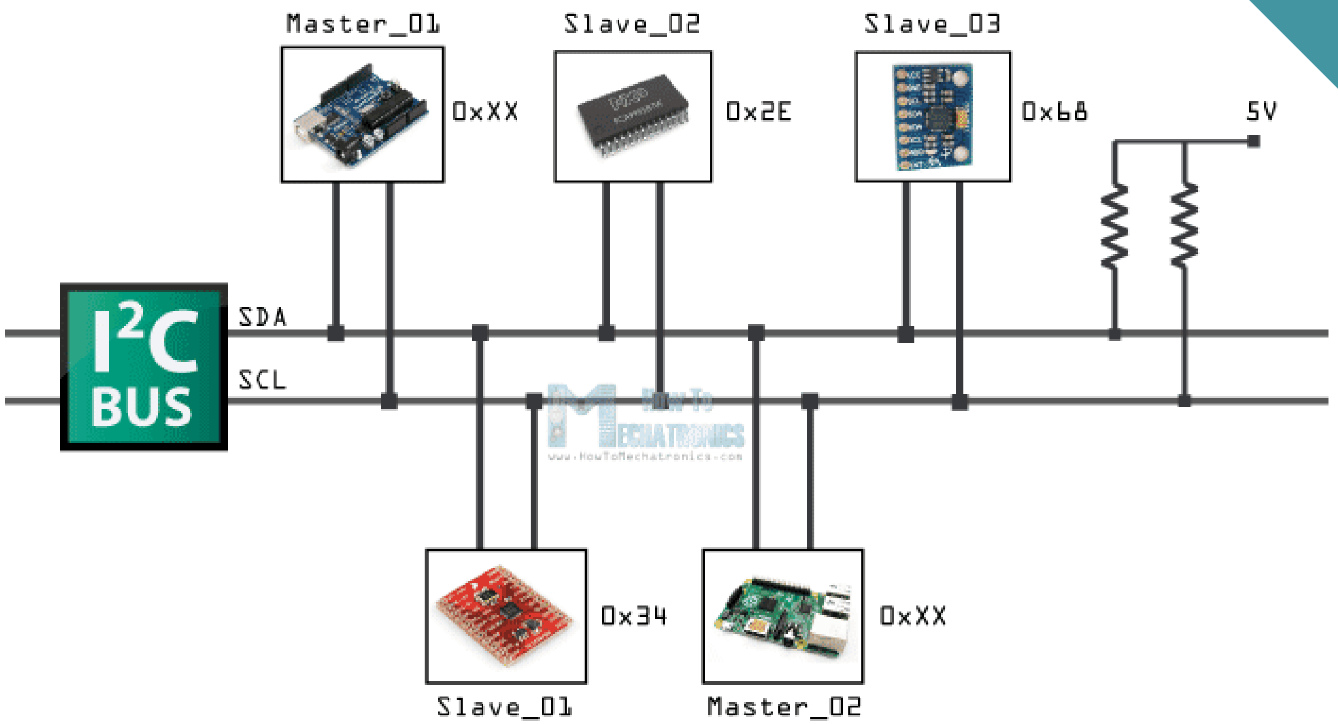
Communication Protocols Supported by BME680

- The I2C was developed by Philips semiconductors (now NXP Semiconductors) in 1982.
- It is a synchronous serial protocol.
- It is used for short distance communication.
- Master Slave topology
- Half Duplex and used only 2 pins (SDA – SCL)
- Each device is software addressable
- Support Multimaster with collision detection and bus arbitration



Communication Protocols Supported by BME680

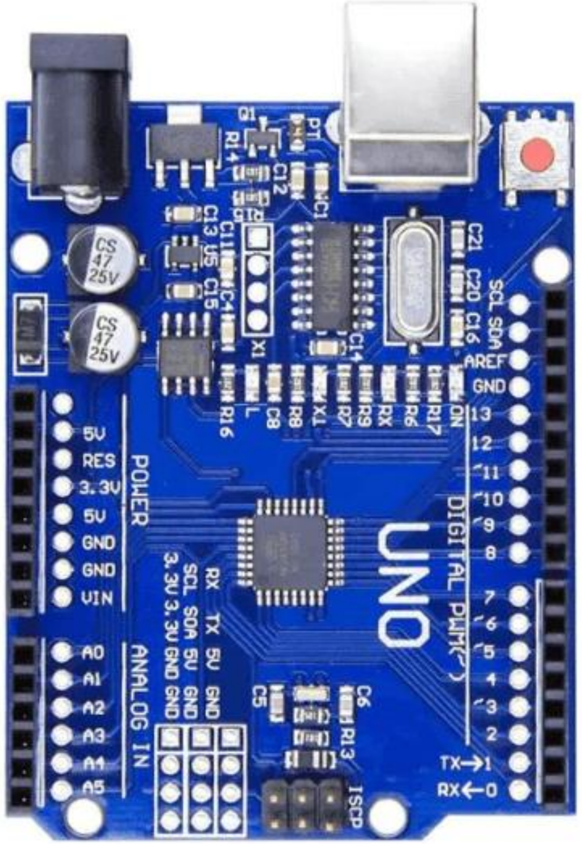
- Supports multiple speeds
  - Standard mode 100 Kbit/Sec
  - Fast mode 400 Kbit/Sec
  - Fast mode plus 1 Mbit/Sec
  - High speed mode 3.4 Mbit/Sec
  - Ultra Fast speed 5 Mbit/Sec



BME680 with Microcontroller

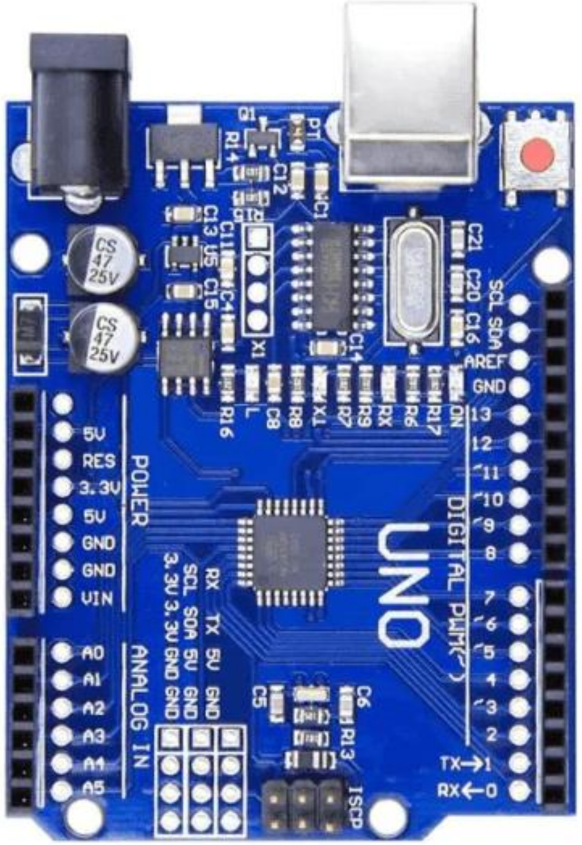
BME680 with Microcontroller

- Arduino UNO R3 is used.



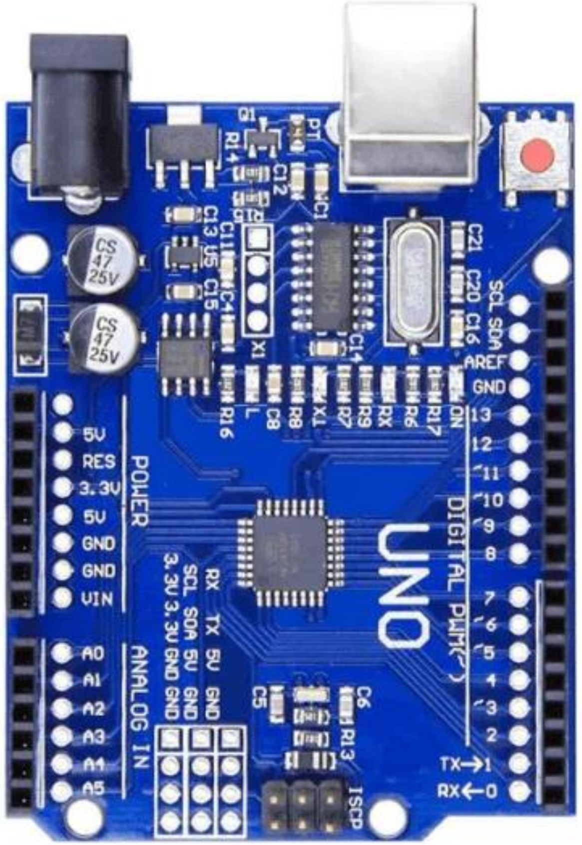
BME680 with Microcontroller

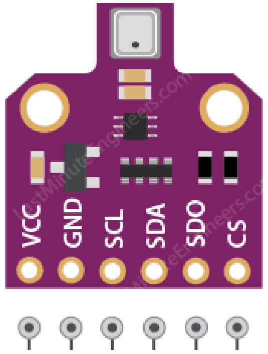
- Arduino UNO R3 is used.



- **Peripherals**
  - 2x 8-bit Timer/Counter with a dedicated period register and compare channels
  - 1x 16-bit Timer/Counter with a dedicated period register, input capture and compare channels
  - 1x USART with fractional baud rate generator and start-of-frame detection
  - 1x controller/peripheral Serial Peripheral Interface (SPI)
  - 1x Dual mode controller/peripheral I2C
  - 1x Analog Comparator (AC) with a scalable reference input
  - Watchdog Timer with separate on-chip oscillator
  - Six PWM channels
  - Interrupt and wake-up on pin change

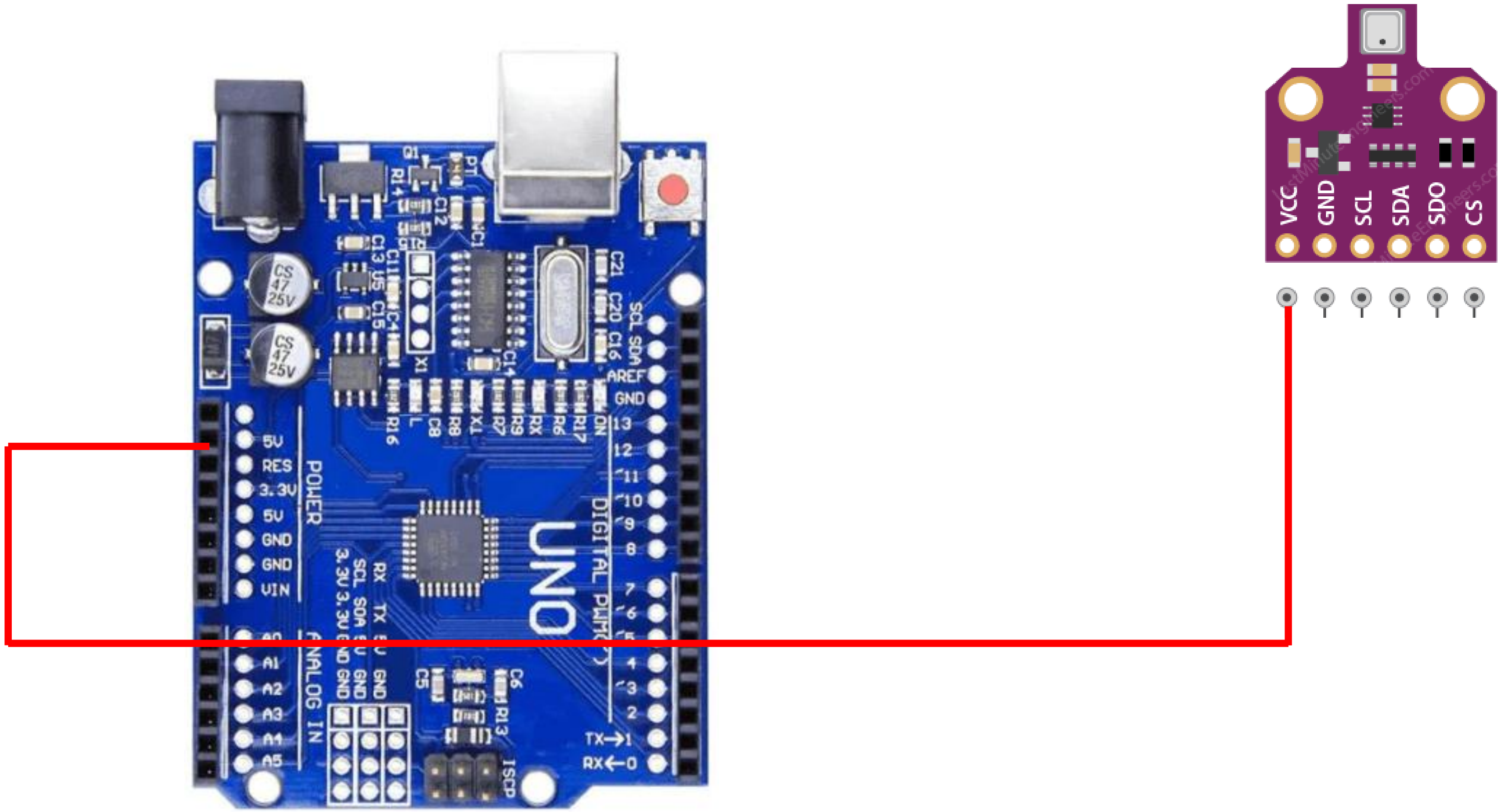
BME680 with Microcontroller





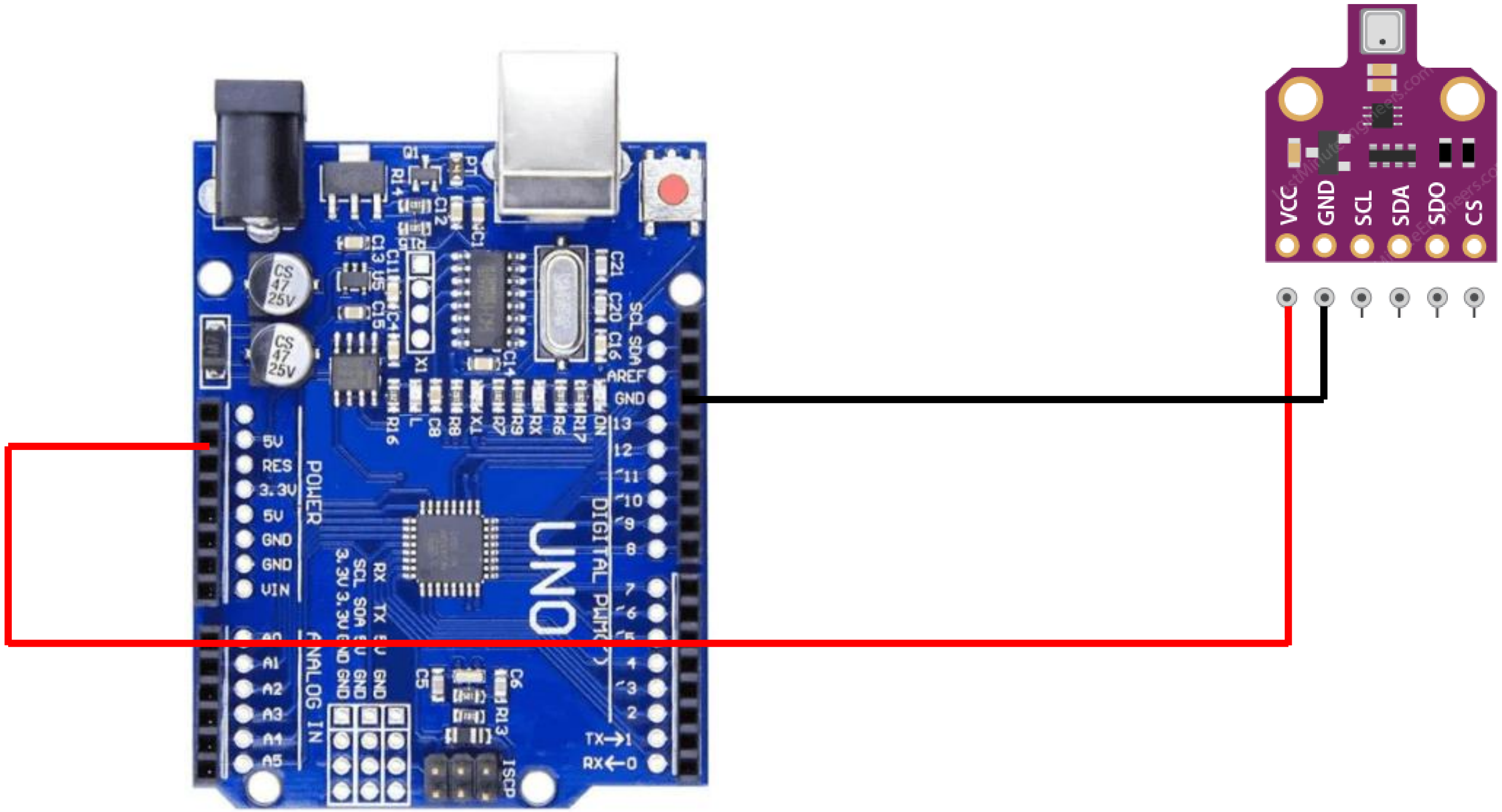
I2C

BME680 with Microcontroller



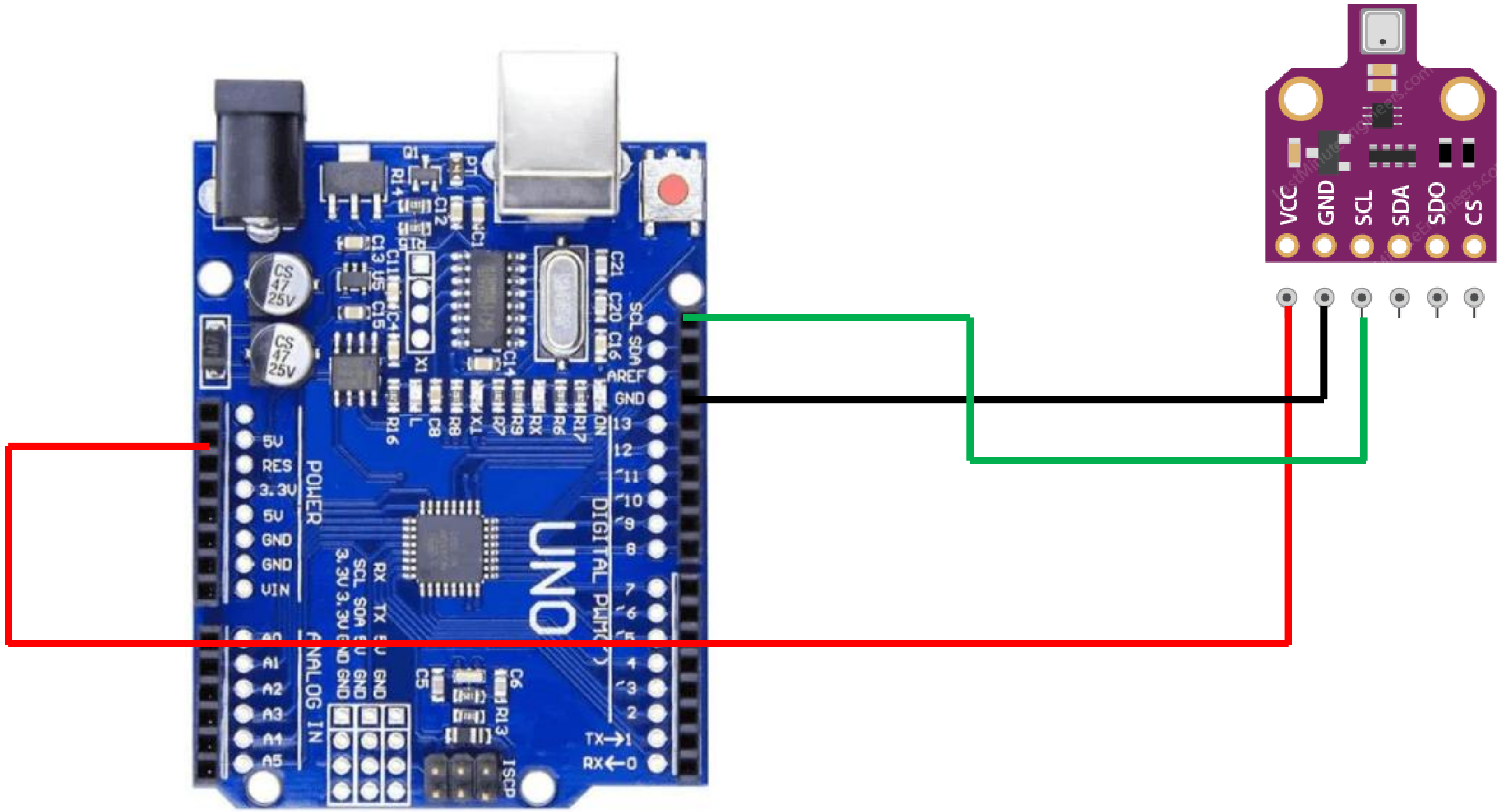
I2C

BME680 with Microcontroller



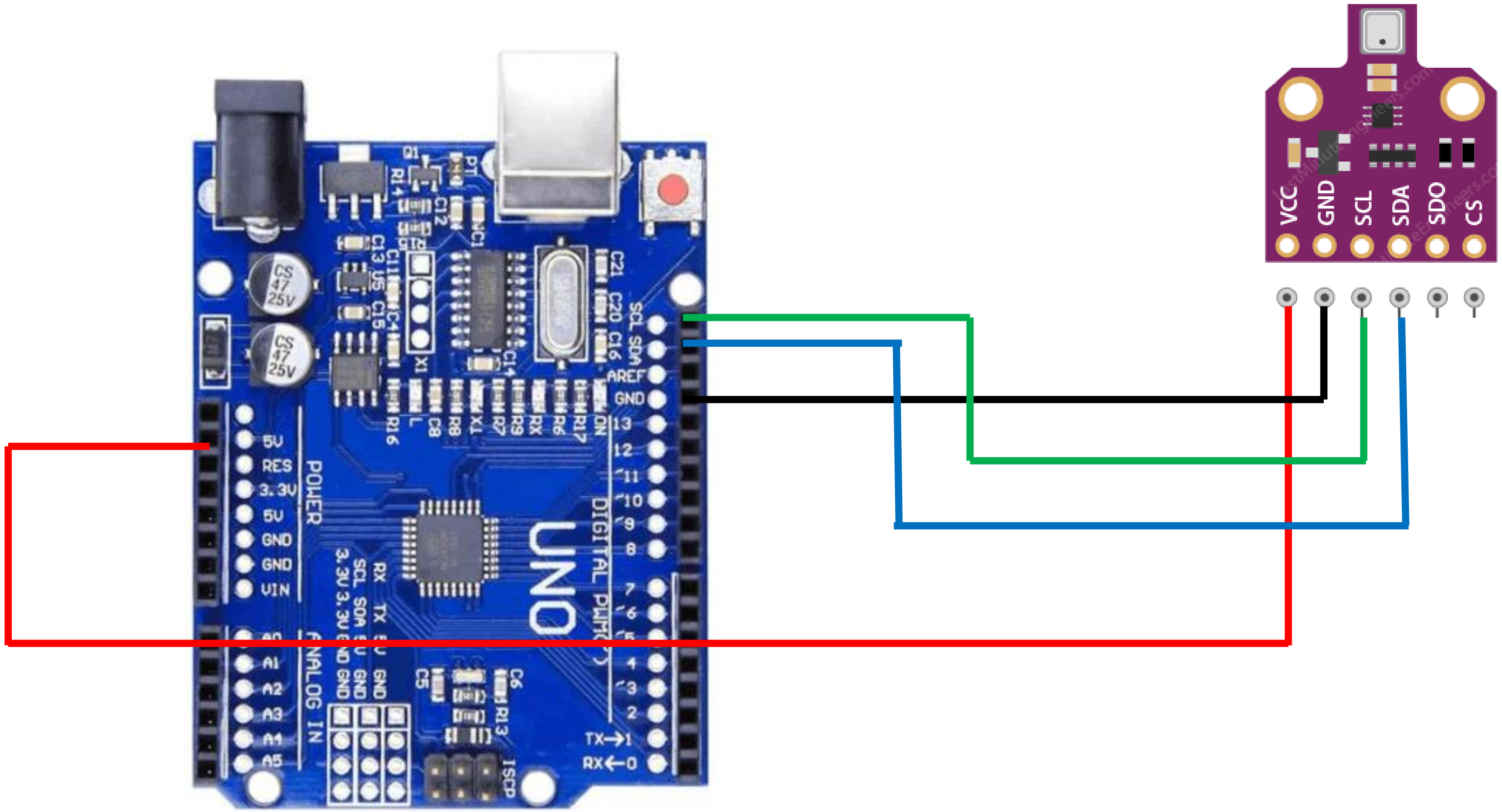
I2C

BME680 with Microcontroller



I2C

BME680 with Microcontroller



I2C

BME680 with Microcontroller

```
void setup()
{
  Serial.begin(9600);
  while(!Serial); // Waiting till serial communication is ready

  lcd.begin(16, 2);

  status = bme.begin(0x76);  // Communicating with sensor

  while (!status) {
    Serial.println("Could not find a valid BME680 sensor, check wiring!");
    delay(1000);
    status = bme.begin(0x76);
  }
}
```

I2C

BME680 with Microcontroller

```
Serial.print("Pressure = ");
Serial.print(bme.readPressure() / 100.0F);
Serial.println(" hPa");

Serial.print("Approx. Altitude = ");
Serial.print(bme.readAltitude(SEALEVELPRESSURE_HPA));
Serial.println(" m");
```

Reading pressure and calculating altitude

I2C

Thank you