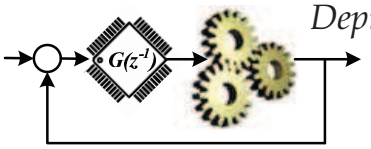
Chapter 5
ATmega328 Analog Interfacing

Assoc. Professor Dr. Rosbi bin Mamat

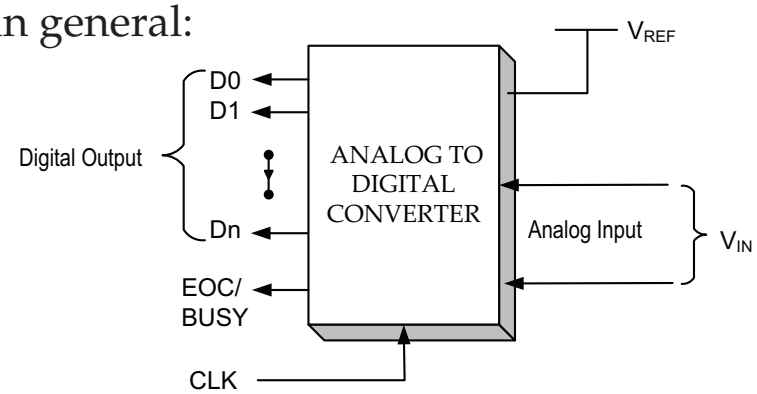
rosbi@fke.utm.my



Dept. of Control & Mechatronics Engineering
Faculty of Electrical Engineering
Universiti Teknologi Malaysia

5.1 Analog-to-Digital Converter (ADC)

- ADC in general:



  - `D0-Dn` ≡ n-bit digital output which represent the analog input value.
  - For ATmega328 ADC, $n = 9 \Rightarrow$ **10-bit ADC**.
  - `EOC/BUSY` ≡ signals the μc when ADC has finished converting analog signal to digital output.

5.1 Analog-to-Digital Converter (ADC)

- ADC in general:
  - V_{REF} ≡ **reference voltage** for ADC. Reference voltage must be stable to give accurate conversion result.
  - V_{IN} ≡ input voltage to be converted to digital output. The possible range for V_{IN} is $0 \rightarrow V_{REF}$ – called **Full-scale Voltage**, V_{FS}.
  - **Conversion time** (τ_c) – time required by ADC to convert an analog voltage to digital code.
  - τ_c is determined by clock freq used & the type of ADC tech. For Arduino Uno, default $\tau_c \approx 100\mu$seconds.

5.1 Analog-to-Digital Converter (ADC)

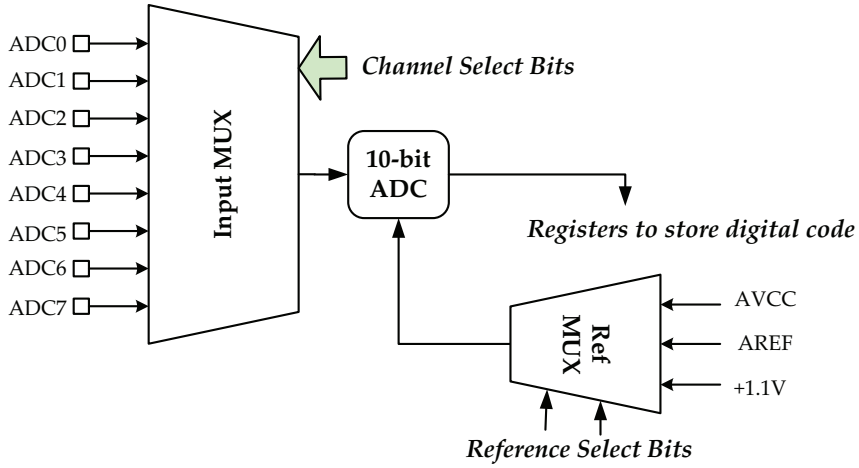
- **Resolution** is the smallest changed in analog voltage to change the digital code by 1 bit.

$$Resolution\ (R) = \frac{V_{FS}}{2^{n+1}}$$

  - **Example:** *Calculate the resolution of ATmega328 ADC if V_{REF} connects to +5V. Compare with an 8-bit ADC.*
  - $V_{FS} = V_{REF} = +5V$
    For 8-bit ADC, $Resolution\ (R) = \frac{5V}{2^8} = 19.5mV$
    For ATmega328 ADC, $Resolution\ (R) = \frac{5V}{2^{10}} = 4.88mV$
  - The ATmega328 ADC can measure smallest change of **4.88mV** while the 8-bit ADC can measure smallest change of **19.5mV**.
  - The ATmega328 ADC has higher resolution. In general, the bigger is n, the higher is the resolution & the more precise voltage change can be measured.

5.2 ATmega328 Internal ADC

- **Main blocks** of internal ADC



  - A 10-bit ADC is shared by 8 channels (`ADC0-ADC7`). The analog Mux choose one of the channels.
  - Port C becomes **analog inputs**. On Arduino Uno, only 6 channels (`A0-A5`) are available.

5.2 ATmega328 Internal ADC

- Digital code output (D) from ATmega328 ADC is given by:

$$D = \frac{V_{IN}}{V_{REF}} \times (2^{10} - 1) = \frac{V_{IN}}{V_{REF}} \times 1023$$

- **Exercise:** *If V_{REF} connects to +5V, calculate the digital codes output from ATmega328 ADC when the analog voltage is: a) +2.5V b) +4.0V What is the analog voltage when the digital code is* `0x250`*?*

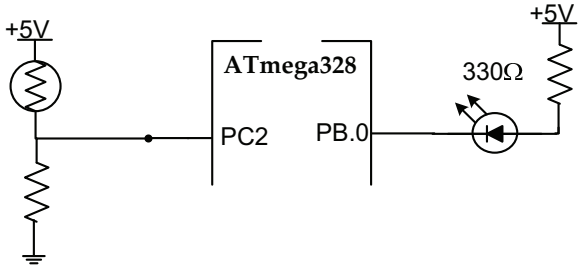
5.3 Arduino ADC Functions

- **Analog Interface** to the outside world with Arduino Uno is provided by 6 pins marked with `ANALOG IN`.

- Arduino libraries for ADC:
  - `analogReference(type)` – configure V_{REF} for ADC. `type` are: `DEFAULT` (5V), `INTERNAL` (1.1V) or `EXTERNAL` (voltage 0-5V connects to V_{REF}).
  - `analogRead(pin)` – convert analog voltage at `pin` (0–5) & returns digital value (0–1023).

5.3 Arduino ADC Functions

- Interfacing **Light Sensor** – an example of interfacing to **analog input device**.
- The light sensor is a **photoresistor** or **light dependent resistor (LDR)** or **photocell** whose resistance decreases with increasing incident light intensity.
- **Exercise:** *For the following circuit, write a program to turn on the LED when it is dark & turn off LED when there is light.*



5.3 Arduino ADC Functions

- **Exercise: interfacing light sensor** – the LDR is connected to PC2/A2 & LED is at PB0/D8. The circuit produce output voltage proportional to light intensity.

```
#define THRESHOLD  50          // below this value means dark

const int ldrPin = 2;          // LDR connects to A2
int sensorValue = 0;           // value read from the LDR

void setup() {
  pinMode(8, OUTPUT);          // PB0 as output
  digitalWrite(8, HIGH);       // Turn off led
  // No need to setup ADC if DEFAULT Vref is used
}

void loop() {
  sensorValue = analogRead(ldrPin);  // read ldr
  if (sensorValue < THRESHOLD)  // Check intensity for dark
    digitalWrite(8, LOW);       // On led
  else
    digitalWrite(8, HIGH);      // Off led
}
```