Latin-American alliance for capacity building in advanced physics

LA-CoNGA physics

Módulo de Instrumentación

Introducción a los Sistemas de Medida

Dennis Cazar Ramírez

Universidad San Francisco de Quito USFQ

15 - febrero - 2021

Latin American alliance for Capacity buildiNG in Advanced physics

LA-CoNGA physics

Cofinanciado por el programa Erasmus+ de la Unión Europea

Contents

Introduction to Measuring systems

- Interfacing with and ADC
- Temperature measure

Interfacing with an ADC

- When we interface sensors to the a μC, μP or FPGA, the output of the sensor many of the times is analog in nature. But μC, μP or FPGA processes digital signals.
- Hence, we use ADC in between sensor and μC. It converts an analog signal into digital and gives it to the μC.

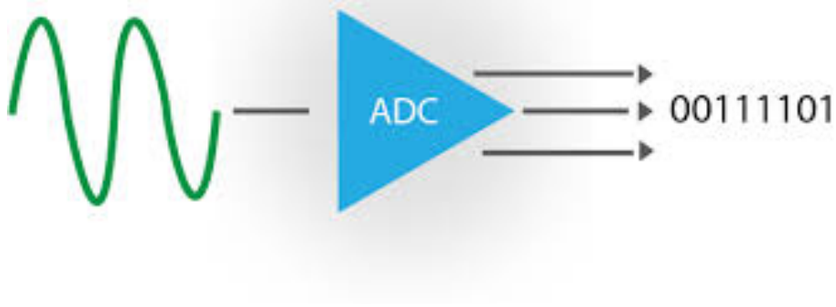


Figure 1: ADC basic function

- We need to know the characteristics of the ADC to design a conditioning circuit for the sensor signal

Interfacing with an ADC

Arduino ADC

- Arduino Uno has **6 0n-board ADC channels** which can be used to read analog signal in the **range 0-5V**.
- It has **10-bit** ADC means it will give digital value in the range of 0 – 1023 (2^10)
- We say it has a **resolution of 10 bits**
- Resolution indicates the number of discrete values it can produce over the range of analog values.

Digital Output Calculation

- ADC Resolution in mV

$$Res(mV) = \frac{V_{ref}}{2^{N-1} - 1} \tag{1}$$

- Digital Output

$$D_{out} = \frac{V_{in}}{Res} \tag{2}$$

- Where V_{ref} The reference voltage is the maximum value that the ADC can convert ($5V$)

Functions for Arduino ADC

analogRead (pin)

- This function is used to read analog value from specified analog pin.
- **pin** - number of analog pin which we want to read
- **returns** - digital value 0 – 1023
- e.g. analogRead(A0) //read analog value at A0 channel

analogReference (type)

- This function is used for configuring the reference voltage used for analog input.
  - **DEFAULT:** the default analog reference of 5 volts (on 5V Arduino boards) or 3.3 volts (on 3.3V Arduino boards)
  - **INTERNAL:** a built-in reference, equal to 1.1 volts on the ATmega168 or ATmega328P and 2.56 volts on the ATmega32U4 and ATmega8 (not available on the Arduino Mega)
  - **INTERNAL1V1:** a built-in 1.1V reference (Arduino Mega only)
  - **INTERNAL2V56:** a built-in 2.56V reference (Arduino Mega only)
  - **EXTERNAL:** the voltage applied to the AREF pin (0 to 5V only) is used as the reference.

Read Analog value using Arduino

- Let's write a program to read varying analog value generated using potentiometer which is connected to $A0$ analog channel.
- Display the digital value on Serial monitor which we got from the Arduino ADC.

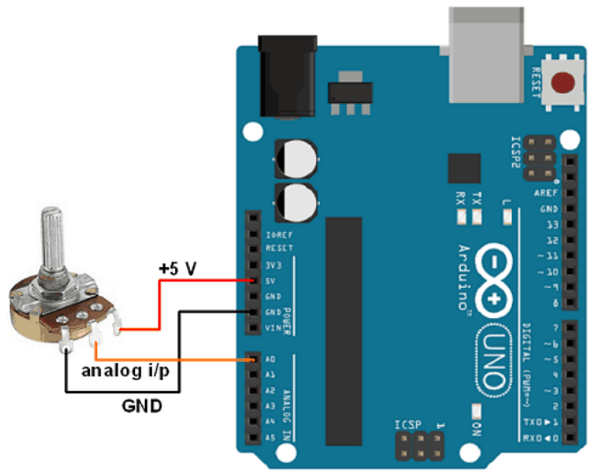


Figure 2: Potentiometer connected Arduino ADC Channel

- Create the circuit in TinkerCad

Sketch for reading analog value

Arduino code

```
int sensorPin = A0; // input pin for the potentiometer
int digitalValue = 0; // variable to store the value coming from the sensor
void setup ()
{
Serial.begin(9600);
}
void loop ()
{
digitalValue = analogRead(sensorPin); // read the value from the analog channel
Serial.print("digital value = ");
Serial.println(digitalValue); //print digital value on serial monitor
delay(1000);
}
```

Example

Example

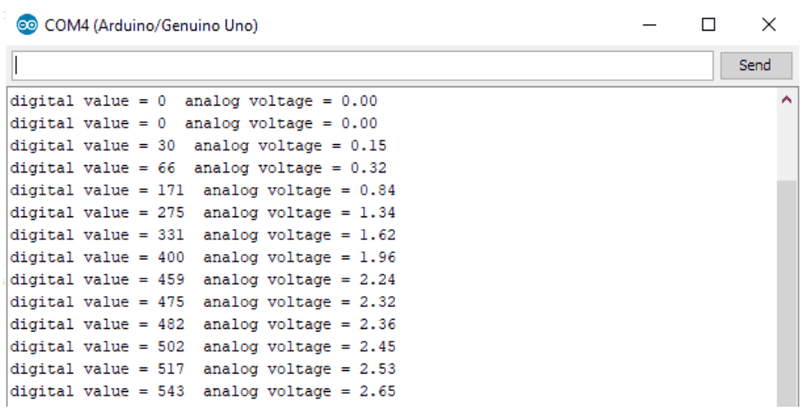
Example

- Modify the code to display the analog value corresponding to digitalValue

Analog output Calculation

$$A_{out} = D_{out} \times \frac{V_{ref}}{2^{N-1}} \tag{3}$$

Output on Serial Window



Exercise: temperature sensor

- Use TinkerCad to do the following
  1. Connect a TMP36 temperature sensor to the Arduino
  2. Read the Analog voltage range of sensor with Arduino
  3. Design a conditioning circuit to adapt senor output range with Arduino ADC input range
  4. Shows temperature value in serial monitor
- EXTRA: Connect an external display to show the temperature value