**ADS1115**

The key features of this ADC are:

* 16 bit resolution
* Between 8 to 860 samples per second.
* I2C communication interface

**we can connect up to four ADS1115 on the same I2C bus.**

The device has a number of programmable operating modes:

**Samples per second (sps):**  this parameter actually sets the amount of time each sample takes, not the sampling frequency. If we set the value of the sps to say 250, then each conversion will last for 1/250 seconds. This means that at lower sample rates we are performing the sample over a longer time period and the result of the conversion is the average of the input to the ADC over this period.

**Input Channel:**

 The device has four input pins and A0-A3. The device is capable of taking both single ended and differential readings.

* In the differential readings mode the device reads the voltage difference between two of the input pins.
* In single ended readings mode the device reads the voltage difference between a single input and ground.

**PGA-Programmable gain amplifier**

Input voltages are first passed through amplifier ,gain of this amplifier can be change hence allow us to measure small voltage with great resolution.

for example:-

Default is +/-6.144v

i.e 6.144/32728 gives 0.1875mv per bit.

If we set full scale to 2.3v then 2.3/32728=0.078mv per bit better resolution

**Continuous Conversion**

Two conversion mode:-

* conversion is initiated externally the result is stored within a register in the ADS1115.
* Conversion is initiated automatically continuous conversion depends on the sps value.Overwrites the perivious value stored in the resistor.However the rate at which microcontroller is reading resistor and rate at which ads1115 is sampling should be same.

Configuring the ADS1115

The config resistor

Bit[15] :-used to start conversion by setting it to 1 a conversion is initiated.

Bits [14:12] These bits set which pin to use as input to the ADC.Each configuration has two inputs AIN~p~ and AIN~n~. By setting AIN~n~ to GND we obtain a single ended input with AIN~p~ as the input.

Bits [11:9] These bits set which setting of the programmable gain amplifier to use

Bit [8] Continuous conversion / No Continuous conversion

Bits [7:5] Set the samples per second (sps) value

Bit [4:2] Comparator setup, we will not use the comparator so these bits are irrelevant

Bit [1:0] Comparator mode, set to 11 to disable the comparator

**For arduino coding we can use the adafruit library**

#include <Wire.h>

#include <Adafruit\_ADS1015.h>

Adafruit\_ADS1115 ads(0x48);//since address is 0x48 we have to connect address pin to ground

float Voltage = 0.0;

void setup(void)

{

  Serial.begin(9600);

  ads.begin();//now the adafruit lib uses wire.h that’s why we have included it

}

void loop()

{

  int16\_t adc0;  // we read from the ADC, we have a sixteen bit integer as a result

  adc0 = ads.readADC\_SingleEnded(0);// 0 stands for the AIN0 we are taking input from pin zero

  Voltage = (adc0 \* 0.1875)/1000;//dividing by 0.1875 bcoz default full scale is 6.144 and dividing it with 32728 we get 0.1875

  Serial.print("AIN0: ");

  Serial.print(adc0);//this will output the digital value

  Serial.print("\tVoltage: ");

  Serial.println(Voltage, 7); //this will output the input voltage that we give as input upto 7 places

  Serial.println();

  delay(1000);

}

I have also tried to understand registers involved so that I can write code for other microcontrollers as well(RPi)

First we transmit four bytes:

1. First is the address of slave
2. Set the pointer register to the config register
3. Then transmit the two bytes of config register

for example 1001000 00000001 11000011 00000011

Then again transmit the slave address and read 2 bytes of config register

Then change the pointer register to conversion register by first transmitting the slave address and then 000000000 to change pointer register.

Then transmit slave address and read two bytes of conversion registration.