

DFRobot_ADS1115. Guía rápida de usuario

El DFRobot_ADS1115 es un dispositivo HW/SW para usar con plataformas de programación entre otras Arduino o RaspberryPi en cualquiera de sus modelos. El enlace a las características completas del producto se puede encontrar en:

<https://www.dfrobot.com/product-1730.html>

Hardware.

- ✓ Utiliza el CI ADS1115, un ADC de alta precisión con 16 bits de resolución. 4 entradas analógicas que puede operar en un rango comprendido entre **GND - 0,3 V** y **V_{cc} + 0,3 V** para cada una de las 4.
- ✓ Permite programar la ganancia de la señal de entrada al ADC lo que le permite trabajar con 6 resoluciones distintas. La mayor resolución de 0,0078125 mV/bit, opera en el rango [0 : +0,256] V, con una ganancia de x16 para la señal de entrada analógica. La menor resolución de 0,1875 mV/bit, opera en el rango [0 : +6,144] V, con una ganancia de x2/3 para la señal de entrada analógica.
- ✓ La salida en serie del ADC sigue un protocolo I²C.

Especificaciones:

- Voltaje de alimentación (V_{cc}): 3.3~5.0V
 - Rango de detección de señales analógicas: [0 : V_{cc}] en la práctica [- 0,3 V < V_{AINX} < V_{cc}+0,3 V]. **Cuidado** con sobrepasar estos límites.
 - Número de canales analógicos: 4
 - Tamaño en bits del ADC: 16 bit
- Nota:** No olvide que el rango máximo de la señal es V_{cc}+0,3. Es decir si se alimenta el CI con 5 V, la señal máxima en cualquiera de las entradas sería 5,3 V. El valor referido de 6,144 V solo tiene sentido de cara al cálculo de la resolución.
- Corriente de operación: [2 : 3] mA
 - Tipo de interfaz: Serie con protocolo I²C
 - Nivel de la señal de salida: Alto: 3.3V, Bajo 0V
 - Dimensiones Módulo: 32mm * 32mm

Descripción de pines

Num	Etiqueta	Descripción
+	V _{cc}	Alimentación: V _{cc} [3.3 : 5.0V] V
-	GND	Alimentación: GND (0 V)
C	SCL	I ² C Línea de Reloj
D	SDA	I ² C Línea de Datos
A	Analog In	Canales Analógicos de Entrada: A0, A1, A2, A3
ADDR	I ² C Address	Interruptor de selección de la dirección I ² C: Direcciones: 0x48, 0x49



Software. Se puede acceder a una librería completa con ejemplos de uso en el enlace:

https://github.com/DFRobot/DFRobot_ADS1115

Resumen de las funcionalidades del dispositivo

El ADS1115 tiene dos direcciones físicas para el canal I²C: 0x48 y 0x49, su selección se realiza de forma manual con el switch de la placa, lo que permite con dos dispositivos poder leer al mismo tiempo (secuencialmente) hasta 8 entradas simples o 4 diferenciales.

/**

* @fn DFRobot_ADS1115

* @brief Sets the IIC address

* @param pWire The pointer of TwoWire class object.

*/

DFRobot_ADS1115(TwoWire *pWire);

```

/**
 * @fn setAddr_ADS1115
 * @brief Sets the IIC address
 * @param i2cAddress: 7 bits I2C address, the range is 1~127.
 */

```

```

    void setAddr_ADS1115(uint8_t i2cAddress);

```

```

/**
 * @fn init
 * @brief init ADS1115 device
 */

```

```

    void init(void);

```

```

/**
 * @fn checkADS1115
 * @brief Checks the connection status of ads1115
 * @return result:
 * @retval true or false
 */

```

```

    bool checkADS1115();

```

```

/**
 * @fn setCompQue
 * @brief Sets the Comparator queue and disable .
 * @n This perform two functions.
 * @n It can disable the comparator function and put the ALERT/RDY pin into a high state.
 * @n It also can control the number of successive conversions exceeding the upper
 * @n or lower thresholds required before asserting the ALERT/RDY pin.
 * @param value The enum variable struct of eADSCompLat_t
 * @n eCOMPQUE_ONE
 * @n eCOMPQUE_TWO
 * @n eCOMPQUE_FOUR
 * @n eCOMPQUE_NONE
 */

```

```

    void setCompQue(eADSCompQue_t value);

```

```

/**
 * @fn setCompLat
 * @brief Sets the Latching comparator .
 * @n This controls whether the ALERT/RDY pin latches once asserted
 * @n or clears when conversions are within the margin of the upper
 * @n and lower threshold values.
 * @param value The enum variable struct of eADSCompLat_t
 * @n eCOMPLAT_NONLAT
 * @n eCOMPLAT_LATCH
 */

```

```

    void setCompLat(eADSCompLat_t value);

```

```

/**
 * @fn setCompPol
 * @brief Sets the Comparator polarity. This controls the polarity of the ALERT/RDY pin
 * @param value The enum variable struct of eADSCompPol_t
 * @n eCOMPPOL_LOW
 * @n eCOMPPOL_HIGH
 */

```

```

    void setCompPol(eADSCompPol_t value);

```

```

/**
 * @fn setCompMode
 * @brief Sets the Comparator mode. This controls the comparator mode of operation.
 * @param value The enum variable struct of eADSCompMode_t
 * @n eCOMP_MODE_TRAD
 * @n eCOMP_MODE_WINDOW
 */
void setCompMode(eADSCompMode_t value);

/**
 * @fn setRate
 * @brief Sets the Data Rate. This controls the data rate setting.
 * @param value The enum variable struct of eADSRate_t
 * @n eRATE_8 8 SPS
 * @n eRATE_16 16 SPS
 * @n eRATE_32 32 SPS
 * @n eRATE_64 64 SPS
 * @n eRATE_128 128 SPS (default)
 * @n eRATE_250 250 SPS
 * @n eRATE_475 475 SPS
 * @n eRATE_860 860 SPS
 */
void setRate(eADSRate_t value);

/**
 * @fn setMode
 * @brief Sets the Device operating mode. This controls the current operational mode of the ADS1115
 * @param value The enum variable struct of eADSMode_t
 * @n eMODE_CONTIN
 * @n eMODE_SINGLE
 */
void setMode(eADSMode_t value);

/**
 * @fn setGain
 * @brief Sets the gain and input voltage range. This configures the programmable gain amplifier
 * @param value The enum variable struct of eADSGain_t
 * @n eGAIN_TWOTHIRDS ADC Range: 0~6.144V (1 bit = 0.1875mV)
 * @n eGAIN_ONE ADC Range: 0~4.096V (1 bit = 0.125mV)
 * @n eGAIN_TWO ADC Range: 0~2.048V (1 bit = 0.0625mV)
 * @n eGAIN_FOUR ADC Range: 0~1.024V (1 bit = 0.03125mV)
 * @n eGAIN_EIGHT ADC Range: 0~0.512V (1 bit = 0.015625mV)
 * @n eGAIN_SIXTEEN ADC Range: 0~0.256V (1 bit = 0.0078125mV)
 */
void setGain(eADSGain_t value);

/**
 * @fn setMux
 * @brief Sets multiplexer
 * @param value The enum variable struct of eADSMux_t
 * @n eADSMUX_1 multiplexer 1
 * @n eADSMUX_2 multiplexer 2
 * @n eADSMUX_3 multiplexer 3
 * @n eADSMUX_4 multiplexer 4
 * @n eADSMUX_5 multiplexer 5
 * @n eADSMUX_6 multiplexer 6
 * @n eADSMUX_7 multiplexer 7
 * @n eADSMUX_8 multiplexer 8
 */
void setMux(eADSMux_t value);

/**

```

```

* @fn setOSMode
* @brief Sets the Operational status/single-shot conversion to start.
* @n This determines the operational status of the device.
* @param value The enum variable struct of eADSOSMode_t
* @n eOSMODE_SINGLE
* @n eOSMODE_BUSY
* @n eOSMODE_NOTBUSY
*/
void setOSMode(eADSOSMode_t value);

/**
* @fn readVoltage
* @brief Reads the voltage of the specified channel
* @param channel The analog channel of MCU
* @return Voltage
*/
uint16_t readVoltage(uint8_t channel);

/**
* @fn comparatorVoltage
* @brief Sets up the comparator causing the ALERT/RDY pin to assert
* @n (go from high to low) when the ADC value exceeds the specified upper or lower threshold
* @n ADC is Differential input channel
* @param channel The analog channel of MCU
* @n channel = 01, // AIN0 = P, AIN1 = N
* @n channel = 03, // AIN0 = P, AIN3 = N
* @n channel = 13, // AIN1 = P, AIN3 = N
* @n channel = 23, // AIN2 = P, AIN3 = N
* @return Voltage
*/
int16_t comparatorVoltage(uint8_t channel);

/**
* @fn setLowThreshold
* @brief Sets the low threshold value
* @param threshold The low threshold value
*/
void setLowThreshold(int16_t threshold);

/**
* @fn getLowThreshold
* @brief Get the low threshold value
* @return The low threshold value
*/
int16_t getLowThreshold();

/**
* @fn setHighThreshold
* @brief Sets the high threshold value
* @param threshold The high threshold value
*/
void setHighThreshold(int16_t threshold);

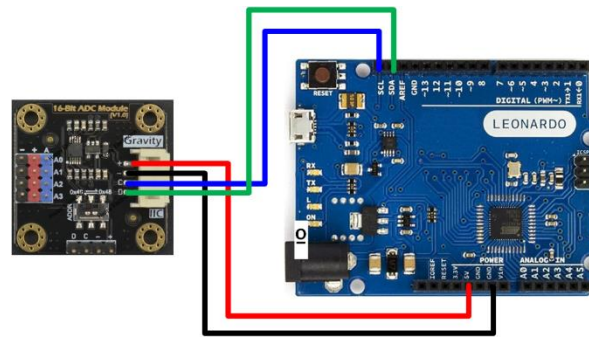
/**
* @fn getHighThreshold
* @brief Get the high threshold value
* @return The high threshold value
*/
int16_t getHighThreshold();

```

Ejemplo de Medida del Módulo DFRobot con Arduino

Conexionado

En la figura adjunta se muestra es conexionado del Módulo DFRobot-DS1115 con Arduino



Programa ejemplo

```
#include <Wire.h>
#include <DFRobot_ADS1115.h>

// Declaracion de variables globales
int16_t adc0, adc2; // Variables para los canales de lectura del DFRobot

DFRobot_ADS1115 ads(&Wire); // Crea un objeto que asocia a ambas librerías cargadas

void setup(void)
{
  // Configura el puerto serie a 9600 bps
  Serial.begin(9600);
  // Configura DFRobot-ADS1115
  ads.setAddr_ADS1115(ADS1115_IIC_ADDRESS0);
  ads.setGain(eGAIN_TWOTHIRDS);
  ads.setMode(eMODE_SINGLE);
  ads.init();
}

void loop(void)
{
  if (ads.checkADS1115()) // Chequea si la conexión es correcta
  {
    adc0 = ads.readVoltage(0); // lee el valor analógico entre la señal de la entrada A0 y tierra
    // con la ganancia programada.

    Serial.print("A0: ");
    Serial.print(adc0);
    Serial.print(" mV, \n ");
    adc2 = ads.comparatorVoltage(23); // lee la diferencia entre los valores analógicos de las entradas A2 y A3
    // con la ganancia programada.

    Serial.print("A2 - A3: ");
    Serial.print(adc2);
    Serial.print(" mV, \n ");
  }
  else
  {
    Serial.println("ADS1115 Desconectado!!");
  }
  delay(1000);
}
```