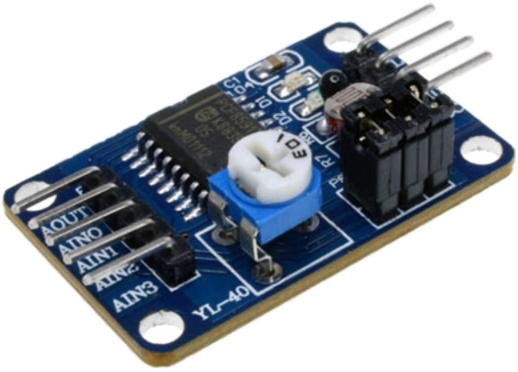
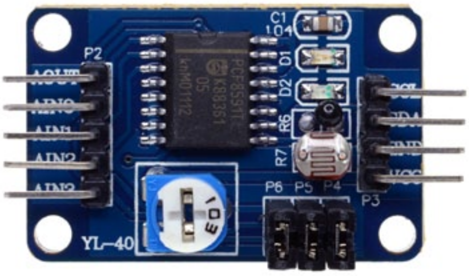
ARD 2 **Arduino Compatibles**
Controllers, Shields, Modules & Sensors

Digital to Analog Converter Module ARD2-2120

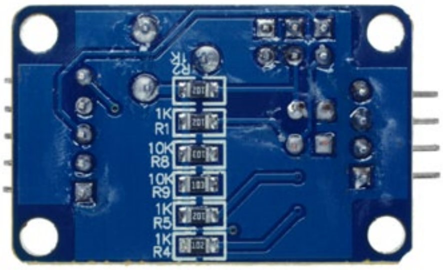


- ***Convert analog signal to digital & vice-versa***
- ***4 analog inputs programmable as single-ended or differential inputs***
- ***Automatic incremental channel selection***
- ***Low standby current***
- ***Integrated photoresistor***
- ***Power LED & DA Output LED (higher voltage = brighter)***
- ***Analog voltage range from VSS to VDD***
- ***Potentiometer to adjust input voltage***

Description



This module is designed to convert analog signals to digital & vice versa. The PCF8591 is a single-chip, single-supply low-power 8-bit CMOS data acquisition device with four analog inputs, one analog output and a serial I2C-bus interface. Three address pins A0, A1 and A2 are used for programming the hardware address, allowing the use of up to eight devices connected to the I2C-bus without additional hardware. Address, control and data to and from the device are transferred serially via the two-line bidirectional I2C-bus.



The jumpers control whether analog input channels of the IC are connected to the analog sources:

Jumper P4 for AIN1: The temperature sensed by the R6 thermister is provided to the ADC.
Jumper P5 to AIN0: The R7 photocell voltage (resistance drop) is provided to the DAC.
Jumper P6 to AIN3: The single turn 10K ohm trimpot voltage (resistance drop – brighter light, lower resistance).

Removing a jumper allows an input channel to be fed from one of the external pins, labelled accordingly.

Specifications

Operating Voltage	2.5–6.0V
Dimensions	36mm x 23mm
Weight	20g
Material	PCB
Analog Voltage	VSS to VDD
Input/Output	I^{2}C

Test Code (Digital to Analog)

```
/**PCF8591 Module Digital To analog test program.
Essentially, this tests the I2C communications to the chip.
The chip address is 0x90.
*/
#include <Wire.h>
#define PCF8591 (0x90 >> 1)     // Device address = 0
#define PCF8591_DAC_ENABLE 0x40
#define PCF8591_ADC_CH0 0x40
#define PCF8591_ADC_CH1 0x41
#define PCF8591_ADC_CH2 0x42
#define PCF8591_ADC_CH3 0x43
byte dac_value=0;
void putDAC(byte dac_value)
{
   Wire.beginTransmission(PCF8591);    //Calls the 8591 to
attention.
   Wire.write(PCF8591_DAC_ENABLE);    //Send a DAC enable word.
   Wire.write(dac_value);             //Send the desired DAC value
(0-255)
   Wire.endTransmission();
 }
void setup()
 {
   Serial.begin(19200);    //Be sure to check the computer port's
speed for this to work.
   Wire.begin();
 }
void loop()
 {            //The IC is stepped by one notch on each loop. The
green LED should slowly increase in brightness
   putDAC(dac_value);
   delay(10);
   Serial.println(dac_value);        //This goes to a terminal
program
   dac_value++;
   delay(200);
}
```

Test Code (Analog to Digital)

```
/**PCF8591 Module Analog to Digital test program.
Essentially, this tests the I2C communications to the chip.
The chip address is 0x90.
*/

#include <Wire.h>
#define PCF8591 (0x90 >> 1)      // Device address = 0
#define PCF8591_DAC_ENABLE 0x40
#define PCF8591_ADC_CH0 0x40
#define PCF8591_ADC_CH1 0x41
#define PCF8591_ADC_CH2 0x42
#define PCF8591_ADC_CH3 0x43

byte adc_value;
byte getADC(byte config)
{
   Wire.beginTransmission(PCF8591);
   Wire.write(config);
   Wire.endTransmission();
   Wire.requestFrom((int) PCF8591,2);
   while (Wire.available())
   {
     adc_value = Wire.read(); //This needs two reads to get the
value.
     adc_value = Wire.read();
   }
   return adc_value;
}
void setup()
{
   Serial.begin(115200);
   Wire.begin();
   Serial.println(“ADC Test”);
}
void loop()
{
   adc_value = getADC(PCF8591_ADC_CH3); //Channel 3 is the pot
   Serial.print(adc_value);
   delay(200);
}
```