Amicrosystems

DATASHEET

AMI-AD1224

HIGH PRECISION CURRENT-TO-DIGITAL CONVERSION MODULE

FEATURES

Excellent long term bias stability 5ppm

Extremely low nonlinearity 5ppm

No latency, each conversion is accurate

6.9Hz to 3.5kHz output rate with selectable speed/resolution

200nV$_{rms}$ noise at 6.6Hz output rate with simultaneous 50/60Hz rejection

Temperature scale factor drift 5 ppm/°C

Larger full scale range easily modified

High output drive interface to DSP/MCU with buffered 3.3V LVC logic

Input resolution / threshold < 10 nA

Low power consumption, small size and low cost

Channel crosstalk guaranteed to -120dB

Specified for operation over the –40°C to +85°C industrial temperature

PRODUCT DESCRIPTION

AMI-AD1224 is a three-channel current-to-digital converter module. The module operates as an interface to current-output sensors to implement high precision analog-to-digital conversion. This high precision module is designed specifically for requiring bias and scale factor stability and excellent non-linearity and extremely low temperature drift. The main applications are in the fields of strapdown inertial navigation, industrial grade sensing, instrumentation and measurements.

The converter operates in the principle of incremental sigma-delta ADC. Fully-differential circuit architecture and optimized anti-aliasing filters are optimized to achieve an excellent common mode rejection ratio (CMRR). The channel crosstalk can be guaranteed to -120dB. The converter is reset periodically; both the analog and digital integrators are reset after each conversion to eliminate any memory effects between successive conversions. High sample-by-sample conversion accuracy is thus guaranteed, and where offset and gain errors are greatly minimized. An analog current signal from sensors is converted to 24-bit digital signal and interfaced through SPI protocol to DSP/MCU.

ABSOLUTE MAXIMUM RATINGS

Parameter	Ratings
Positive Supply VDDA to GND	+36VDC
Negative Supply VSSA to GND	-33VDC
Input Current Limts	±100mA
Shock Resistance	15ms half-sine, positive/negative single shock, in one direction x, y, z
Vibration	20 g_{rms}, 20-2000 Hz, random noise, in each direction x, y, z
Operating Temperature Range	- 40°C to + 85°C
Storage Temperature Range	- 55°C to + 125°C

Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the module.

RECOMMENDED POWER SUPPLY OPERATING RANGE

Parameter	Min	Typ	Max	Unit
Analogue power supply VDDA	+12.5	+15	+18	V
Analogue power supply VSSA	-12.5	-15	-18	V

ELECTRICAL CHARACTERISTICS SPECIFICATIONS

Typical @ TA = 25°C, VDDA = +15 V, VSSA=-15V, unless otherwise noted. Measurements done using the Arduino Due platform, which is a microcontroller board based on the Atmel 32-bit SAM3X8E ARM Cortex-M3 CPU.

Parameter	Min	Typ	Max	Unit
Resolution (No Missing Codes)		24		bits
Full scale range	-12		+12	mA
Bias stability**		1	5	ppm*
Output stability with load***		5	10	ppm*
Bias without calibration		5	25	ppm*
Switch on/off repeatability		0.3	3	ppm*
Bias temperature drift coefficient	-10	5	10	ppm* / °C
Full-scale error		10	50	ppm*
Scale factor drift coefficient		1	5	ppm* / °C
Resolution / Threshold		5	10	nA
Integral nonlinearity		5	15	ppm*
Bandwidth	0	6.9	3500	Hz
VDDA supply current	7	8	9	mA
VSSA supply current	3	4	5	mA

*ppm - parts per million of VREF, VREF=5V.

** input open, output rate 6.9Hz, standard deviation measured in one hour.

***input current ±1 mA, output rate 6.9Hz, standard deviation measured in one hour.

PCB DIMENSIONS

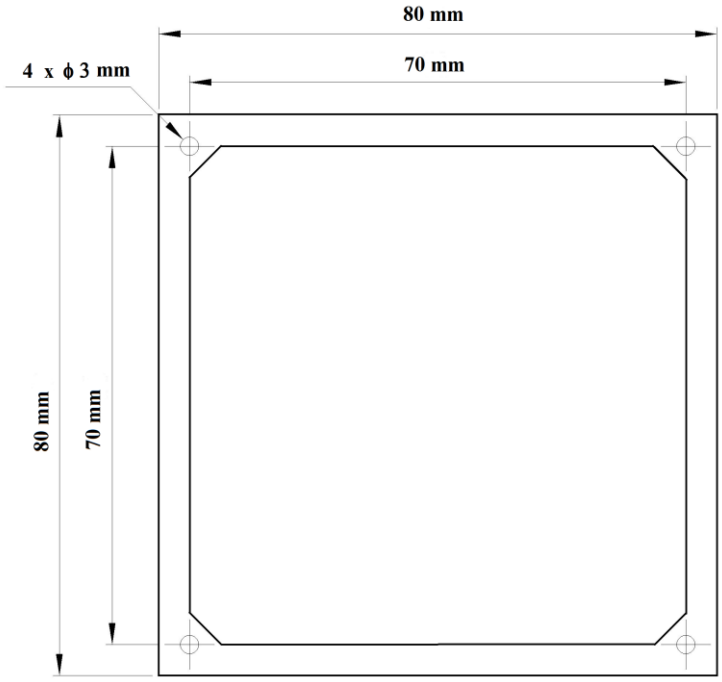

FIGURE 1. PCB dimensions and four mounting holes Ø 3 mm.

CONNECTOR DIMENSIONS (mm)

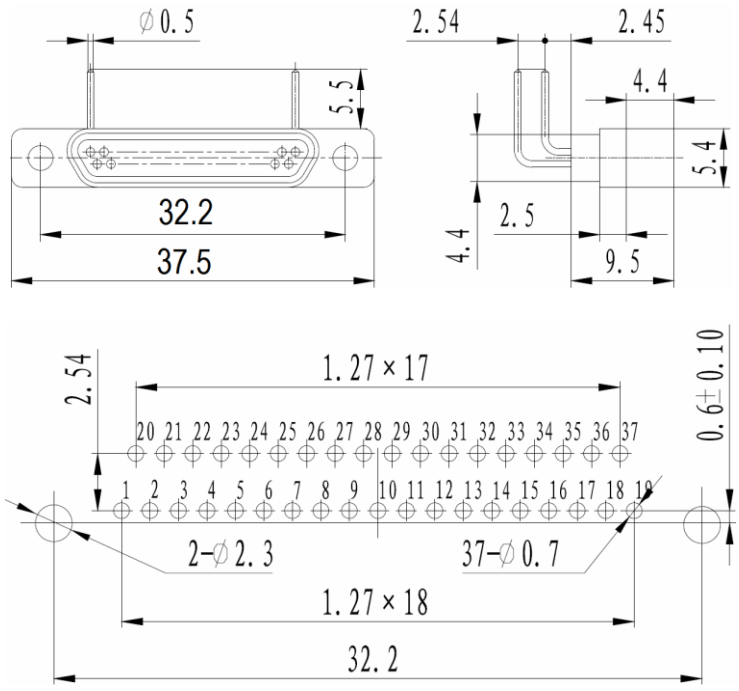

FIGURE 2. J30-9-37ZKW-J plug having 1.27 x 2.54 pitch connector dimension

ELECTRICAL PIN CONFIGURATIONS

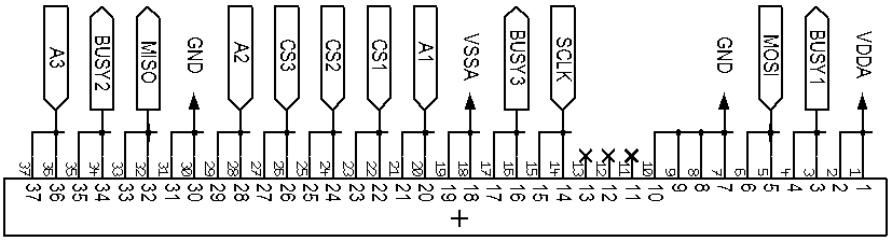

TABLE 1. PIN CONFIGURATIONS

PIN	SYMBO	SIGNAL	PIN	SYMBO	SIGNAL
1	VDDA	+15V positive power supply	20	A1	channel 1 input
2	VDDA	+15V positive power supply	21	A1	channel 1 input
3	BUSY1	channel 1 conversion busy	22	CS1	SPI - Slave Select channel 1
4	BUSY1	channel 1 conversion busy	23	CS1	SPI - Slave Select channel 1
5	MOSI	SPI - Master Output Slave Input	24	CS2	SPI - Slave Select channel 2
6	MOSI	SPI - Master Output Slave Input	25	CS2	SPI - Slave Select channel 2
7	GND	GROUND	26	CS3	SPI - Slave Select channel 3
8	GND	GROUND	27	CS3	SPI - Slave Select channel 3
9	GND	GROUND	28	A2	Channel 2 input
10	GND	GROUND	29	A2	Channel 2 input
11		reserved	30	GND	GROUND
12		reserved	31	GND	GROUND
13		reserved	32	MISO	SPI - Master Input, Slave Output
14	SCLK	SPI - Serial Clock	33	MISO	SPI - Master Input, Slave Output
15	SCLK	SPI - Serial Clock	34	BUSY2	channel 2 conversion busy
16	BUSY3	channel 3 conversion busy	35	BUSY2	channel 2 conversion busy
17	BUSY3	channel 3 conversion busy	36	A3	channel 3 input
18	VSSA	-15V negative power supply	37	A3	channel 3 input
19	VSSA	-15V negative power supply			

*__BUSY__ pins: Conversion in Progress Indicator. This pin is HIGH while the conversion is in progress and goes LOW indicating the conversion is complete and data is ready. It remains low during the sleep and data output states. At the conclusion of the data output state, it goes HIGH indicating a new conversion has begun.

ESD CAUTION

ESD (electrostatic discharge) sensitive components. Proper ESD precautions are recommended to avoid performance degradation or loss of functionality subjected to high energy electrostatic discharges.

PROGRAMMING GUIDANCE

EXAMPLE: Code for interfacing the module to ARDUINO DUE platform.

TABLE 2. MOSI Speed/Resolution Programming

OSR4	OSR3	OSR2	OSR1	OSR0	CONVERSION RATE	RMS NOISE	ENOB
X	0	0	0	1	3.52kHz	23µV	17
X	0	0	1	0	1.76kHz	3.5µV	20
X	0	0	1	1	880Hz	2µV	21.3
X	0	1	0	0	440Hz	1.4µV	21.8
X	0	1	0	1	220Hz	1µV	22.4
X	0	1	1	0	110Hz	750nV	22.9
X	0	1	1	1	55Hz	510nV	23.4
X	1	0	0	0	27.5Hz	375nV	24
X	1	0	0	1	13.75Hz	250nV	24.4
X	1	1	1	1	6.875Hz	200nV	24.6

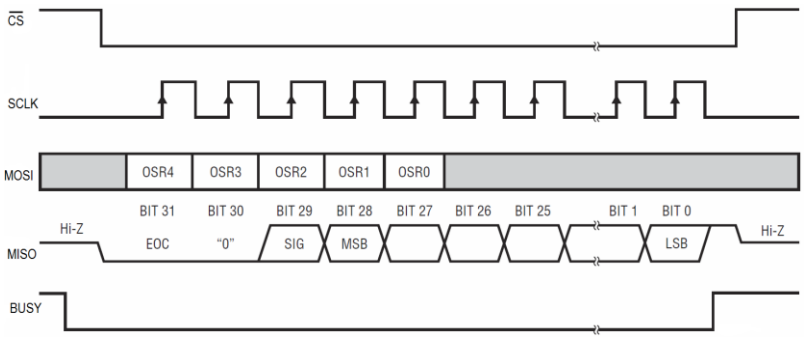

FIGURE 3. MOSI Speed/Resolution Programming

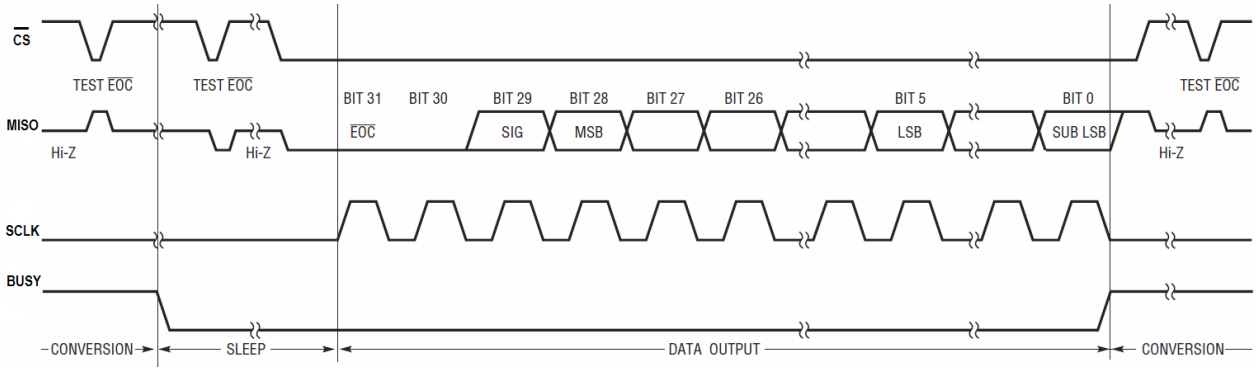

FIGURE 4. Single cycle operation timing mode

// **Code for the SPI interface to ARDUINO DUE**

```
// http://arduino.cc/en/Reference/SPI
// On the Arduino Due, the SAM3X has advanced SPI capabilities. The extended API can use pins 4, 10, and 52 for CS.
// for DUE    MOSI - ICSP PIN4
//            MISO - ICSP PIN1
//            SCK  - ICSP PIN3
//            CS1  -      PIN4
//            CS2  -      PIN10
//            CS3  -      PIN52
#include <SPI.h>      // include spi library
float volts1;
float volts2;
float volts3;
const int CS1 = 4;
const int CS2 = 10;
const int CS3 = 52;
const int BUSY1 = 7;
const int BUSY2 = 6;
const int BUSY3 = 5;

void setup()
{ Serial.begin(115200);              //set serial rate to 115200
  pinMode (CS1, OUTPUT);
  pinMode (CS2, OUTPUT);
  pinMode (CS3, OUTPUT);
  pinMode (BUSY1, INPUT);
  pinMode (BUSY2, INPUT);
  pinMode (BUSY3, INPUT);

  digitalWrite( CS1, HIGH);
  digitalWrite( CS2, HIGH);
  digitalWrite( CS3, HIGH);

  digitalWrite(CS1,LOW);
  delayMicroseconds(1);
  digitalWrite(CS1,HIGH);
```

```
 digitalWrite(CS2,LOW);
 delayMicroseconds(1);
 digitalWrite(CS2,HIGH);

 digitalWrite(CS3,LOW);
 delayMicroseconds(1);
 digitalWrite(CS3,HIGH);

 SPI.begin();              //  initialize SPI
 SPI.setBitOrder(MSBFIRST);
 SPI.setDataMode(SPI_MODE0);
 SPI.setClockDivider(84);   // set clock sync to 1MHz div = 84,
}
void loop()
{uint32_t startTime = millis();
 float mins;
 float volts1=0;
 float volts2=0;
 float volts3=0;
 volts1 = SpiRead_X();
 volts2 = SpiRead_Y();
 volts3 = SpiRead_Z();
 mins = (float) millis () /1000;   // elapsed time in minutes
 Serial.println("time(s),Xo(uV),Yo(uV),Zo(uV)");
 Serial.print(mins,2);
 Serial.print(", ");
 Serial.print(volts1,2);
 Serial.print(", ");
 Serial.print(volts2,2);
 Serial.print(", ");
 Serial.print(volts3,2);
 Serial.println();
 Serial.println();
 }
// ================================================================
// SpiRead() -- read 4 bytes from ADCs via SPI, return uVolts
```

```
float SpiRead_X(void) {
long result1 = 0;
long OFFSET1 = 0;
byte sig1 = 0;   // sign bit
byte b1;
byte REG1 = 0xff;            //MOSI speed/resolution selection table 2, 0xff - CONVERSION RATE = 6.9 Hz; RMS NOISE = 200nV.
float v1;

digitalWrite(CS1, LOW);
digitalWrite(CS1, HIGH);
delay(198);                  // time required for conversion, depended on MOSI speed/resolution selection table 2., here 6.9 Hz example.
digitalWrite(CS1,LOW);    // take the CS1 pin low to select CHANNEL-1
delayMicroseconds(1);
b1 = SPI.transfer(REG1);  // B3
if ((b1 & 0x20) ==0) sig1=1;  // is input negative ?
b1 &=0x1f;              // discard bits 25..31
result1 = b1;
result1 <<= 8;
b1 = SPI.transfer(0xff);  // B2
result1 |= b1;
result1 = result1<<8;
b1 = SPI.transfer(0xff);  // B1
result1 |= b1;
result1 = result1<<8;
b1 = SPI.transfer(0xff);  // B0
result1 |= b1;
digitalWrite(CS1,HIGH);    // take the CS1 pin high to bring MISO to hi-Z and start new conversion
if (sig1) result1 |= 0xe0000000;    // if input is negative, insert sign bit (0xf0.. or 0xe0... ?)
v1 = result1;
v1 = v1 / 32;                // scale result down , last 5 bits are "sub-LSBs"
v1 = v1 * 5e6 / (2 * 16777216)-OFFSET1;    //  max scale (2^24 = 16777216), OFFSET1-the input offset of channel-1.
return(v1);
}
float SpiRead_Y(void) {
long result2 = 0;
long OFFSET2 = 0;
```

```
 byte sig2 = 0;    // sign bit
 byte b2;
 byte REG2 = 0xff;              //MOSI speed/resolution selection table 2., 0xff - CONVERSION RATE = 6.9 Hz; RMS NOISE = 200nV.
 float v2;

 digitalWrite(CS2, LOW);
 digitalWrite(CS2, HIGH);
 delay(198);                    // time required for conversion, depended on MOSI speed/resolution selection table 2., here 6.9 Hz example.
 digitalWrite(CS2,LOW);     // take the CS2 pin low to select CHANNEL-2
 delayMicroseconds(1);

 b2 = SPI.transfer(REG2);   // B3
 if ((b2 & 0x20) ==0) sig2=1;  // is input negative ?
 b2 &=0x1f;    // discard bits 25..31
 result2 = b2;
 result2 <<= 8;
 b2 = SPI.transfer(0xff);   // B2
 result2 |= b2;
 result2 = result2<<8;
 b2 = SPI.transfer(0xff);   // B1
 result2 |= b2;
 result2 = result2<<8;
 b2 = SPI.transfer(0xff);   // B0
 result2 |= b2;
 digitalWrite(CS2,HIGH);              // take the CS2 pin high to bring MISO to hi-Z and start new conversion
 if (sig2) result2 |= 0xe0000000;       // if input is negative, insert sign bit (0xf0.. or 0xe0... ?)
 v2 = result2;
 v2 = v2 / 32;                        // scale result down , last 5 bits are "sub-LSBs"
 v2 = v2 * 5e6 / (2 * 16777216)-OFFSET2;  // max scale (2^24 = 16777216), OFFSET2-the input offset of channel-2.
 return(v2);
}

 float SpiRead_Z(void) {
 long result3 = 0;
 long OFFSET3 = 0;
 byte sig3 = 0;    // sign bit
```

```
 byte b3;
 byte REG3 = 0xff;            //MOSI speed/resolution selection table 2., 0xff - CONVERSION RATE = 6.875Hz; RMS NOISE = 200nV.
 float v3;

 digitalWrite(CS3, LOW);
 digitalWrite(CS3, HIGH);
 delay(198);                  // time required for conversion, depended on MOSI speed/resolution selection table 2., here 6.9 Hz example.
 digitalWrite(CS3,LOW);    // take the CS3 pin low to select CHANNEL-3
 delayMicroseconds(1);

 b3 = SPI.transfer(REG3);   // B3
 if ((b3 & 0x20) ==0) sig3=1;  // is input negative ?
 b3 &=0x1f;    // discard bits 25..31
 result3 = b3;
 result3 <<= 8;
 b3 = SPI.transfer(0xff);   // B2
 result3 |= b3;
 result3 = result3<<8;
 b3 = SPI.transfer(0xff);   // B1
 result3 |= b3;
 result3 = result3<<8;
 b3 = SPI.transfer(0xff);   // B0
 result3 |= b3;
 digitalWrite(CS3,HIGH);                  // take the CS3 pin high to bring MISO to hi-Z and start new conversion
  if (sig3) result3 |= 0xe0000000;          // if input is negative, insert sign bit (0xf0.. or 0xe0... ?)
 v3 = result3;
 v3 = v3 / 32;                                  // scale result down , last 5 bits are "sub-LSBs"
 v3 = v3 * 5e6 / (2 * 16777216)-OFFSET3;  // max scale (2^24 = 16777216), OFFSET3-the input offset of channel-3.
 return(v3);
}
```