K-Thermocouple Amp - MAX6675 MDU33

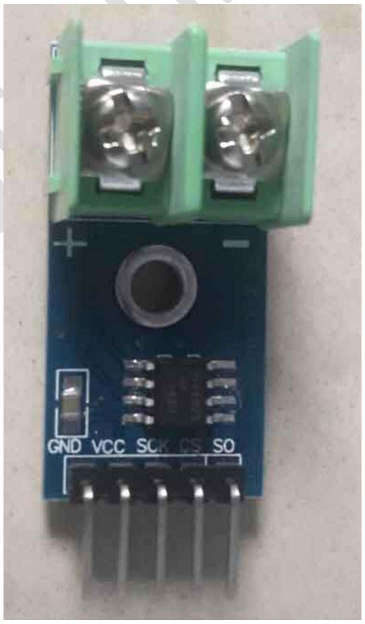
Digital Output K-Type, Measure temperature from 0 to +1024 deg Celcus, SPI interface

Description

Measures temperature from 0 to +1024 deg Celsius with 0.25 deg C resolution. Output is simple serial SPI interface to be used with any microcontroller.

Thermocouples are very sensitive, requiring a good amplifier with a cold-compensation reference. This is an excellent digital-output amplifier which is very simple to use, and if your microcontroller has SPI interface capability.

The board performs cold-junction compensation and digitizes the signal from a type-K thermocouple. The data is output in a 12-bit resolution, SPI-compatible, read-only format. This converter resolves temperatures to 0.25°C, allows readings as high as +1024°C.



Features

- Covers almost entire range from 0 to +1024 deg C in resolution of 0.25 deg C
- Highly accurate 12-Bit results having 0.25°C Resolution
- Direct Digital Conversion of Type –K Thermocouple Output
- Cold-Junction Compensation
- Simple SPI-Compatible Serial Interface
- Open Thermocouple Detection

Applications

- Heat Exchanges & Furnaces
- Industrial Monitoring and Control
- Appliances Control
- HVAC Environment Monitoring
- Automotive Applications like ECU, Exhaust Gas temperature measurement

Specification

Parameter	Min	Typical	Max	Unit
Operating Voltage	3	5	5.5	V DC
Current Consumption	0.7		10	mA
Conversion Time		0.17	0.22	Seconds
Resolution		0.25		Deg C
Serial Clock Frequency			4	Mhz Max

Pin Details

These pins are marked on PCB as follows

#	Symbol	Name	Details
1	GND	Ground	Power Supply Ground
2	+5V	Positive	Power Supply +5V DC regulated
3	SCK	Serial Clock	Serial Clock Input
4	CS	Chip Select	Active low Chip Select Input, Set CS low to enable serial interface
5	SO	Serial Out	Serial Data Out

Connecting Thermocouple

Green connector is screw terminal for connection of thermocouple

Pin	Details
TC-	Alumel Lead of Type-K Thermocouple.
TC+	Chromel Lead of Type- K Thermocouple

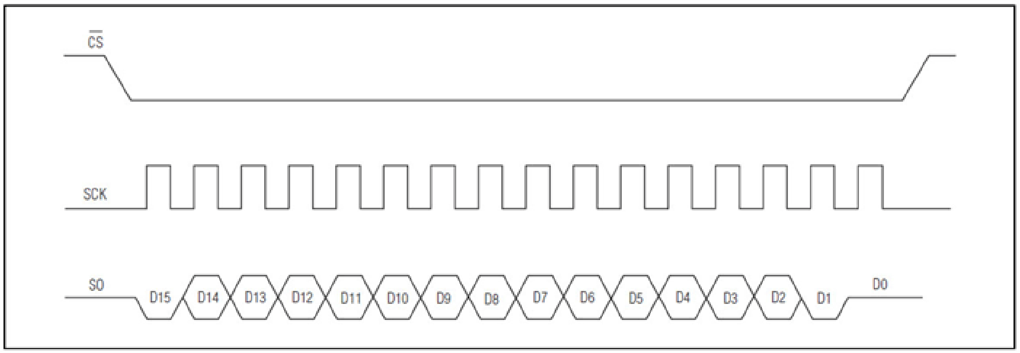
Information

The board is a sophisticated thermocouple-to-digital converter with a built-in 12-bit analog-to-digital converter (ADC). The board also contains cold-junction compensation sensing and correction, a digital controller, an SPI-compatible interface, and associated control logic. The board is designed to work in conjunction with an external microcontroller (µC) or other intelligence in thermostatic, process-control, or monitoring applications.

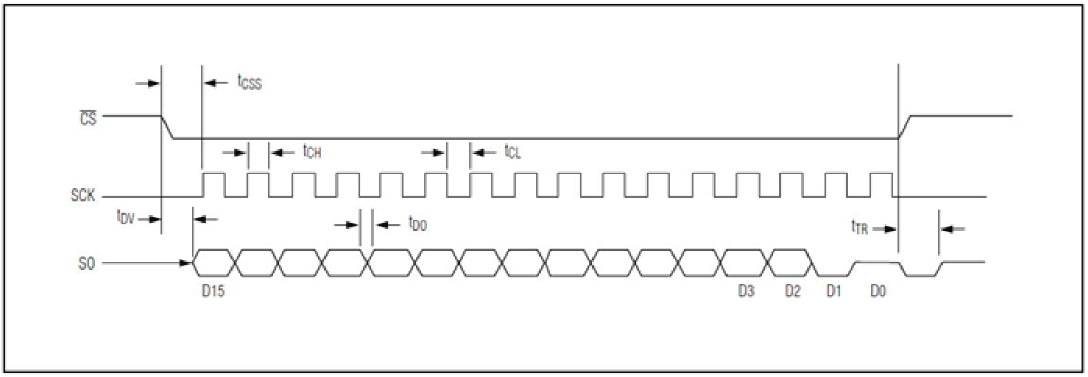
Digitization

The ADC adds the cold-junction diode measurement with the amplified thermocouple voltage and reads out the 12-bit result onto the SO pin. A sequence of all zeros means the thermocouple reading is 0°C. A sequence of all ones means the thermocouple reading is +1023.75°C.

Serial interface protocol



Serial interface timing



SO output

BIT	DUMMY SIGN BIT	12-BIT TEMPERATURE READING												THERMOCOUPLE INPUT	DEVICE ID	STATE
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	MSB											LSB		0	Three-state

Open Thermocouple

Bit D2 is normally low and goes high if the thermocouple input is open.

Sample code for implementing in C language of any microcontroller

```
#include <REGX51.H>
#include <stdio.h>

sbit CS = P2^0;
sbit SO = P2^1;
sbit SCK = P2^2;

//global error flag to tell you if a thermocouple is connected or not
unsigned char tc_error;

void init_serial( void )
{
        //9600 bps @ 11.059 MHz
        SCON = 0x50;          /* Setup serial port control register */
                                                  /* Mode 1: 8-bit uart var.
baud rate */
                                                  /* REN: enable receiver */

        PCON &= 0x7F;         /* Clear SMOD bit in power ctrl reg */
```

```
                                                /* This bit doubles the baud
rate */

        TMOD &= 0xCF;           /* Setup timer/counter mode register */
                                                /* Clear M1 and M0 for timer 1
*/
        TMOD |= 0x20;          /* Set M1 for 8-bit autoreload timer */

        TH1 = 0xFD;            /* Set autoreload value for timer 1 */
                                                /* 9600 baud with 11.0592 MHz
xtal */

        TR1 = 1;               /* Start timer 1 */

        TI = 1;                /* Set TI to indicate ready to xmit */

}
// -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
// -=-=-=-=-  Delay x ms       -=-=-=-=-=-=-=
// -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
void delayms(int x)     // delays x msec (at fosc=11.0592MHz)
{
        int j=0;
        while(x>=0)
        {
                for (j=0; j<100; j++);
                x--;
        }
}

void init_sensor()
{
        CS = 1; // disable chip select
        SO = 0;
        SCK = 0;
}

float read_sensor()
{
   int i;
   unsigned int AdcResult, k;  // 16 bit

        CS=0;   // Active chip select

        SCK = 1;       //Cycle the clock for dummy bit 15
        SCK = 0;

   //--- read ADC result 12 bit --------
   AdcResult=0;
   for(i=11;i>=0;i--) {
      SCK=1;
          k = 0;
          if(SO==1)
                k = 1 << i;
          AdcResult|=k;
      SCK=0;                  // falling clock
   }
   /* Read the TC Input inp to check for TC Errors */
    SCK = 1;
        if(SO==1)
        tc_error = 1;
        else
                tc_error = 0;
    SCK = 0;

   CS=1;

   //adjusts data to floating point format, and accounts for the decimal
point
   return(float)(AdcResult/4.0);
}

void main()
{
        init_serial();
        init_sensor();
        printf("Welcome\n");
        while(1)
        {
          if(tc_error)
                printf("Thermocouple Open");
          else
                printf("%01.2f%cC",read_sensor(),0xB0);
          putchar(13);
          delayms(1000);
        }
}
```

Note: Board **does NOT** include K type thermocouple; you can order it separately if required.