TC4_diag User Guide

Rev. 0.02

TC4_diag is a simple diagnostic program to program the calibration EEPROM of a TC4 shield and do some basic tests to verify the major chips are working or addressable.

TC4_Diag User Interface

1. Generic Stuff

   Enter/send a blank line in the serial monitor to repaint the menu.
   Menu items that take an argument must have a single space between the menu item command and the argument, like “m 10” to read the thermal couples 10 times.
   A short to ground on the I2C lines will cause the program to hang on power-up.
   Don’t include the single quotes when sending a command

2. Menu Item ‘a’, display cal block

   Displays Calblock contents read from the EEPROM.

3. Menu Item ‘b’, display cal fill info

   Displays calibration fill block contents used to program the EEPROM calblock. The fill memory is loaded with defaults on power up, but can be modified using field specific menu items d, e, f, g, h and j.

4. Menu Item ‘c’, write fill block to eeprom

   Moves the contents of the fill memory to the EEPROM. It also rereads the content back from the EEPROM to update the calibration information used by the sketch

5. Menu Item ‘C’, Copy cal block from eeprom to fill block

   Moves the content read from the EEPROM at power up into the fill memory. This is used if you TC4 that different from the default V4.00 and plan to modify some of the fields.

6. Menu Item ‘d’, change fill PCB

   Used to create the PCB type string, must start with TC4 or the read calibration function will not consider the content valid. The string can be up to 40 characters long including the null terminator.

7. Menu Item ‘e ‘, change fill Version

   Used to set the version of the board. It is a freeform string, but should reflect the schematic version, like V4.00. The string can be up to 16 characters long including the null terminator.

8. Menu Item ‘f’, change fill Cal Gain

   Used to compensate for gain errors in the ADC gain stage. It is a floating point value, like 1.0025.

9. Menu Item ‘g’, change fill Cal offset

   Offset value to compensate for errors in the ADC output. It is normally a small integer value.

10. Menu Item ‘h’, change fill T offset

    Offset to compensate for errors in temperature measured using a T type thermal couple. It is a floating point value.

11. Menu Item ‘j’, change fill K offset

    Offset to compensate for errors in temperature measured using a K type thermal couple. It is a floating point value.

12. Menu Item ‘k’, Set Number of TC channels to display or selects TC

    Used to limit the numbers of channels displayed when the TCs are read using menu item ’m’. It also can be used to select which thermal couple is displayed for menu items ‘M’ and ‘s’.

13. Menu Item ‘m’, read TC(s) up to 1000 times

    Used to display the ambient and temperature of the TC, it take a single argument which determines how many time the TC(s) are read.

14. Menu Item ‘M’, read TC microvolt

    Used to display the ADC output as a voltage. It is used to find the cal_gain. It also requires the use of a precise voltage of 50uV being applied to the TC input. The TC being displayed is determined by the value set with menu item k.

15. Menu Item ‘n’,   test adc

    Reads the MCP3224 ADC. It checks that various bits can be set and read back with the same value. It also checks to see if the ADC can do a conversion in the expected amount of time.

16. Menu Item ‘N’, read Raw adc data

    Does a few conversions and dumps the raw data being read from the ADC. Inspecting the raw data may give hints to possible malfunctions of the ADC.

17. Menu Item ‘q’, test MCP9800

    Reads the MCP9800 Ambient temperature sensor. It checks that all of the registers can be accessed and checks that bits can be set and read back with the same value.

18. Menu Item ‘Q’, read Raw MCP9800 data

    Does a few conversions and dumps the raw data being read from the MCP9800. Inspecting the raw data may give hints to possible malfunctions of the MCP9800.

19. Menu Item ‘r’, eeprom dump

    Dumps a 128 byte block/page of data from the EEPROM. Accepts one argument with values from 0 to 511. Entering ‘r’ without an argument will display the first block/page which holds the calibration block.

20. Menu Item ‘s’, Calibration reference temp

    Used to set reference temperature when calibrating a thermal couple.

21. Menu Item ‘S’, toggle calculate Cal diff

    Control to alter the read thermal couple output ‘m’. When enabled the read thermal couple display will show the temperature read from a thermal as well as the difference between the thermal couple temp and the reference temp set with ‘s’. Menu item ‘k’ can be used to select the thermal that is being read.

22. Menu Item ‘T’, pin number to toggle(arduino numbers 2-17 or T enter to reset)

    Selects a pin to toggle when the ‘t’ menu item is entered. This item takes one argument between 2-17, pins 0, 1, 18, 19 are the serial pins and I2C pins and can’t be enabled for toggle.

23. Menu Item ‘t’, toggle pin

    Toggles the pin selected by the ‘T’ menu item. This item takes one argument between 2-17, pins 0, 1, 18, 19 are the serial pins and I2C pins and can’t be read.

    Entering ‘T’ without an argument turns off toggling. If the pin defaults to an output the last state of the pin is left on the pin. If the pin defaults to an input the pin is reset back to an input when toggling is turned off.

24. Menu Item ‘U’, pin number to read

    Selects the pin to be read by the ‘u’ menu item. This item takes one argument between 2-17, pins 0, 1, 18, 19 are the serial pins and I2C pins and can’t be read. When a pin is selected to be read it is set to an input with a pull up to ensure the input is not floating.

    Entering ‘U’ without an argument turns off toggling. If the pin defaults to an output the pin is turned back to an output.

25. Menu Item ‘u’, read pin

    Read the pin selected by the ‘U’ menu item.

26. Menu Item ‘v’ = toggle verbose debug mode

    When set to 1 additional debug information is displayed for some menu items.

27. Menu Item ‘V’ = show program variables

    Show the state of program variables.

28. Menu Item ‘1’ = scan I2C bus

    Scan the I2C bus for devices. Used to ensure the major ICs on the shield are responding to the appropriate addresses. This can be used to find open pins on address lines which would cause a device to not respond normally to the sketches. If a device has opens or shorts on the address lines a device may show up in the scan in the wrong address.

Initial testing of a TC4 shield

Assumptions:

Basic knowledge of Arduino IDE

Basic understanding of Digital Logic terms

1) Download TC4_diag.ino from Github https://github.com/RusticRoaster/TC4_diag
2) Perform a continuity test of the shield before install on the Arduino
3) Load the TC4_diag sketch into your Arduino with the TC4 installed
4) From inside the Arduino IDE Open the serial monitor
5) Verify menu is printed and no error messages
   A blank EEPROM will be identified by this printout
   "# Failed to read EEPROM. Using default calibration data."
   If a pin is identified as not being High, use a multi-meter to verify the voltage on the I/O pin of the TC4 shield (Table 1 lists the pin names and default voltages).

6) Run a I2C scan of the shield(menu item 1),
   Only 3 devices should be found at addresses 0x48, 0x50, 0x68
   0x48 = MCP9800, Ambient Temperature Sensor
   0x50= 24LC512, EEPROM
   0x68=MCP3424, ADC

7) Program the EEPROM with the default calibration data (menu item c)
   Close Serial Monitor and reopen the Serial Monitor verify sketch identities valid calibration
   Use menu item A to display contents to verify all fields programmed correctly
   Note - Menu items d, e, f, g, h, j can be used to create custom data, before using c to write EEPROM. The PCB string must start with TC4 or the calibration block will be identified as not valid.

8) Verify communications to the ADC (menu item n)
9) Verify communications with the MCP9800 Ambient Temp Sensor(menu item q)
10) Verify ambient temp sensor results (menu item m)
    Placing your finger on the MCP9800 for a few seconds and rerun 'm' should show a increase in temp
11) Connect a TC to TC0 and verify results (menu item m)
    Place the probe into a glass of ice water slurry should show a temp close to freezing

Table 1

Pin Number	Name	Default Voltage	Input/Output
0	DIO0/SERIAL-RXD	5V	Output
1	DIO1/SERIAL-TXD	5V	Input
2	DIO2	5V	Input
3	DIO3	5V	Input
4	DIO4	5V	Input
5	DIO5	5V	Input
6	DIO6	5V	Input
7	DIO7	5V	Input
8	DIO8	5V	Input
9	DIO9/OT1	5V	Input
10	DIO10/OT2	0V	Output
11	DIO11	0V	Output
12	DIO12	5V	Input
13	DIO13/Arduino LED	0V	Output
14	AIN0	5V	Input
15	AIN1	5V	Input
16	AIN2	5V	Input
17	AIN3	5V	Input
18	AIN4/I2C-DAT	5V	I/O
19	AIN5/I2C-CLK	5V	I/O

Pin Output mode testing

1) Connect the Negative lead of the Multi-meter to ground
2) Connect the positive lead of the Multi-meter to the pin to be tested
3) Enter ‘T’ # into the serial monitor to select the pin to test (# is pin number to test)
4) Voltage should be approximately 0V
5) Enter ‘t’ into the serial monitor to toggle the output voltage
6) Voltage should be approximately 5V
7) Enter ‘t’ to toggle the pin back to 0V
8) Enter ‘T’ to turn off toggle function on that pin
   If pin is by default an output the voltage will stay low
   If pin is by default an input the voltage will change to 5V

Pin Input mode testing

1) Enter ‘U’ # into the serial monitor to select pin to read(# is pin number to test)
2) Enter ‘u’ and verify pin is reported being High
3) Enter ‘T’ 13 into the serial monitor to turn on output toggling on pin 13
4) Enter ‘t’ to toggle the output pin until it reports being Low
5) Connect a short jumper wire from pin 13 to the input pin to test
   Run Pin output mode test first to verify output works correctly
6) Enter ‘u’ and verify pin is reported being Low
7) Enter ‘t’ to toggle the output pin
8) Enter ‘u’ and verify pin is reported being High

Continuity Tests

1) Confirm high impedance between Ground and 5V
2) Confirm high impedance between Ground and 3.3V
3) Confirm high impedance between Ground and Vin
4) Confirm high impedance between 5V and 3.3V
5) Confirm high impedance between 5V and Vin
6) Confirm high impedance between 3.3V and Vin
7) Confirm 4.7K between Pin 18 and 5V
8) Confirm high impedance between Pin 18 and Ground
9) Confirm 4.7K between Pin 19 and 5V
10) Confirm high impedance between Pin 18 and Ground
11) Verify high impedance between TC connector leads
12) Verify high impedance between TC connector leads and ground

    Troubleshooting tips:
    If this is a newly built unit, inspect unit for solder splashes and bridges using planet of light and some sort of magnification, look at the IC and bypass caps closely from multiple angles. If this was a working unit inspect unit for foreign objects as well as solder bridges and splashes. The ADC can be easily damaged if a TC is connected to an external voltage source, inspect it for cracks or burn marks.

OT1 Test

1) Enter 'U' 2
2) Enter 'u'
3) Verify pin 2 reported as being high
4) Enter 'T' 9
5) Connect OT1 – to Pin 2
6) Enter 'u'
7) Verify pin 2 reported as being high
8) Enter 't'
9) Enter 'u'
10) Verify pin 2 reported as being low
11) Connect voltmeter negative to ground
12) Connect a voltmeter positive to OT1+
13) Verify voltage approx. Vin(5V for V5.xx)
14) Enter 't'
15) Enter 'T'
16) Enter 'U'

OT2 Test

1) Enter ‘U’ 2
2) Enter ‘u’
3) Verify pin 2 reported as being high
4) Enter ‘T’ 10
5) Connect OT2 – to Pin 2
6) Enter ‘u’
7) Verify pin 2 reported as being high
8) Enter ‘t’
9) Enter ‘u’
10) Verify pin 2 reported as being low
11) Connect voltmeter negative to ground
12) Connect a voltmeter positive to OT2+
13) Verify voltage approx. Vin(5V for V5.xx)
14) Enter ‘t’
15) Enter ‘T’
16) Enter ‘U’