TC4 Shield
Version 7.00

MCP9800
Temperature Sensor

MCP3424
ADC

Ground points

IO2

IO3

I2C

OT2

OT1

TC1

TC2

TC3

TC4

Arduino Uno

Reset
Button

512K
EEPROM

ANLG1

ANLG2

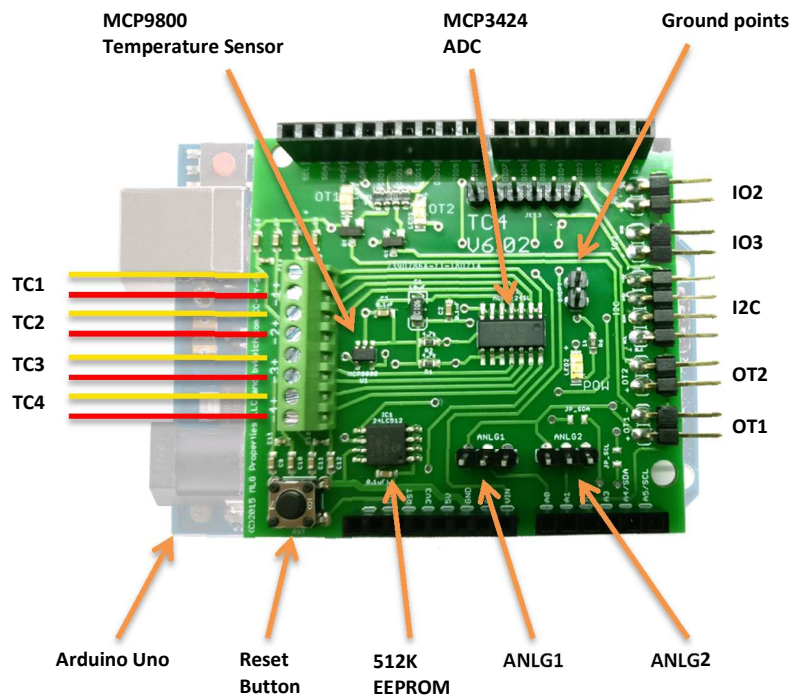**Arduino Uno:** The TC4 Shield was intended to be comnnected to an Arduino Uno.

TC1 – TC4 ports: These screw terminals are for connecting thermocouples. Up to four thermocouples can be connected.

OT1 port: This port is typically used to control a heating element via a SSR (zero voltage turn-on). It is an open collector output switched by a small transistor.

OT2 port: This port is typically used to control an AC fan/blower via a SSR (instantaneous turn-on) when using phase angle control (CONFIG_PAC2 and CONFIG_PAC3 modes). It is an open collector output switched by a small transistor.

IO2 port: This port is typacally used to connect a Zero Cross Detector (CONFIG_PAC2 mode)

IO3 port: This port can be configured as either in input or output. It can be used as an input for a Zero Cross Detector (CONFIG_PAC3 mode) or as an output to control either a DC fan/blower (CONFIG_PWM mode) or a proportional gas valve (CONFIG_PAC2_IO3HTR mode)

ANLG1 and ANLG2 ports: 10K potentiometers can be connected to these ports to allow manual control of the heater and fan/blower outputs. ANLG1 controls the heater and ANLG2 controls the fan/blower.

I2C port: The I2C port is typically used to connect to a LCD screen so roasting temperatures and other information can be shown.

Reset Button: The reset button is connected to the Arduino reset pins.

Ground points: The ground pins provide a convenient place to attach grounding wires if required. Sometimes grounding the TC4 helps get clean temperature readings.

MCP3424 ADC: This is a 4 channel analogue to digital convertor that reads the thermocouple inputs.
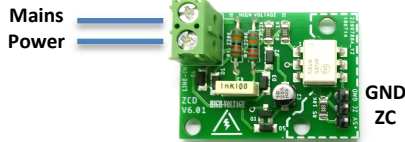
MCP9800 Temperature Sensor: This onboard temperature sensor provides an ambient temperature reading and is also used to do the cold junction compensation for the thermocouple readings.

512K EEPROM: The EEPROM memory can be used to store some temocouple calibration data as well as roast profiles for use in stand alone mode.
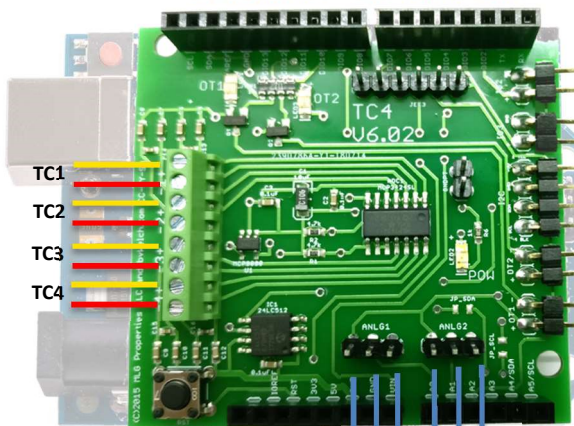
CONFIG_PAC2 Mode
Connection Example 1

Zero Cross Detector

Mains
Power

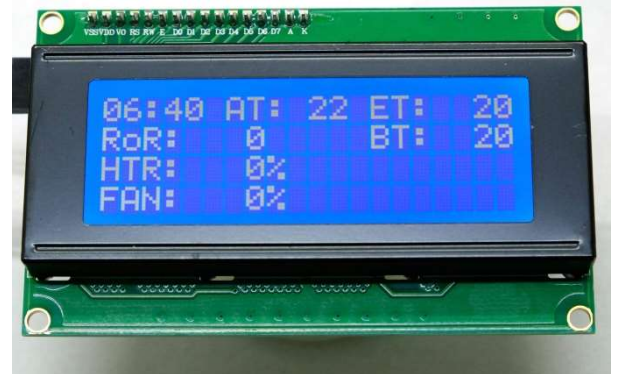GND
ZC

TC4 Shield + Arduino Uno

TC1
TC2
TC3
TC4

IO2
IO3

GND
ZC

I2C
+5V
GND
SDA
SCL

GND
+5V
SDA
SCL

OT2
OT1

+5V
+5V

ANLG1
ANLG2

I2C LCD - Optional

```
06:40 AT: 22 ET:  20
RoR:      0      BT:  20
HTR:     0%
FAN:     0%
```

POT 2 - Optional

POT 1 - Optional

Instantaneous Turn-on SSR

CRYDOM
SOLID-STATE RELAY
D2425-10

Zero Voltage Turn-on SSR

crydom
SOLID-STATE RELAY
D2425

CONFIG_PAC2 Mode
Connection Example 2

RobotDyn AC Dimmer
(Upgrade heatsink and add forced airflow for controlling >2.5A)

AC Light Dimmer Module — RobotDyn robotdyn.com

AC-IN — Mains Power
GND
LOAD — AC Fan

Zero-cross detector

TC4 Shield + Arduino Uno

DIO10
DIO09

IO2
IO3

GND
ZC

TC4 V6.02

TC1
TC2
TC3
TC4

I2C

GND

+5V   GND
GND   +5V
SDA   SDA
SCL   SCL

OT2
+5V

OT1
+5V

ANLG1
ANLG2

I2C LCD - Optional

```
06:40 AT: 22 ET:    20
RoR:      0      BT:  20
HTR:      0%
FAN:      0%
```

POT 2 - Optional

POT 1 - Optional

AC Light Dimmer Module — RobotDyn robotdyn.com

AC-IN — Mains Power
LOAD — AC Heater
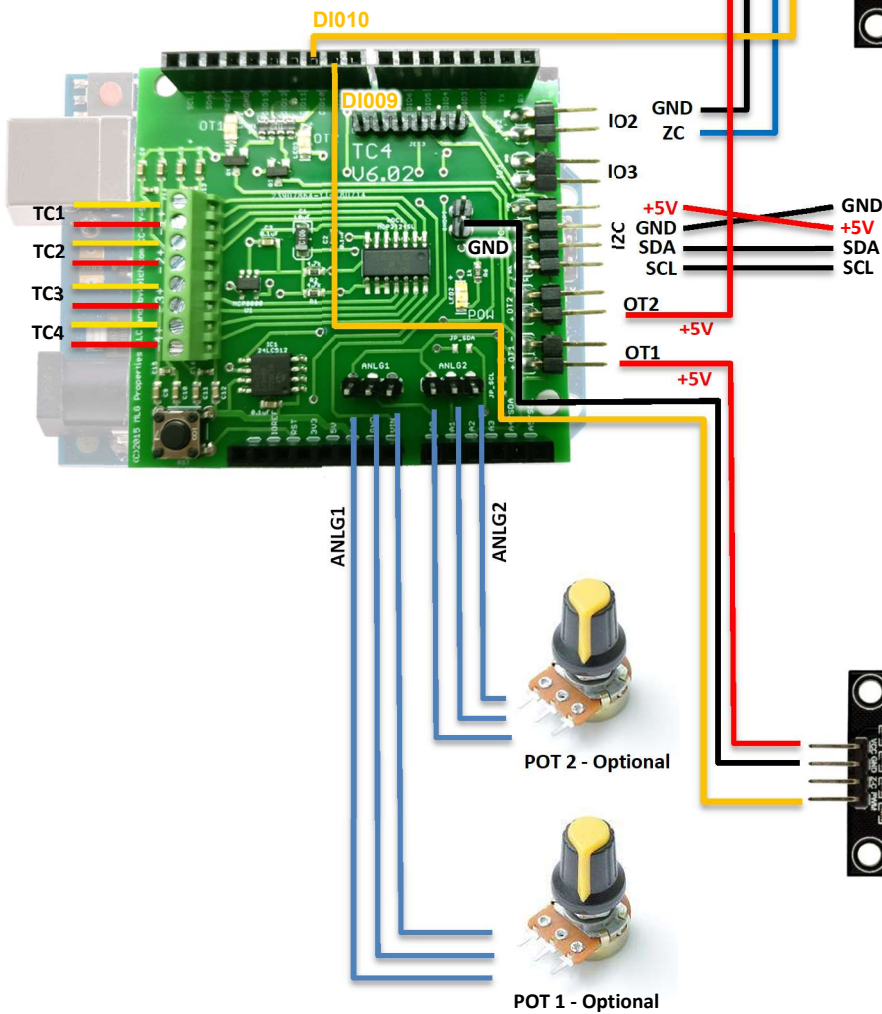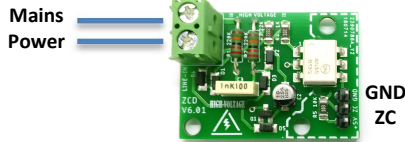
Zero-cross detector
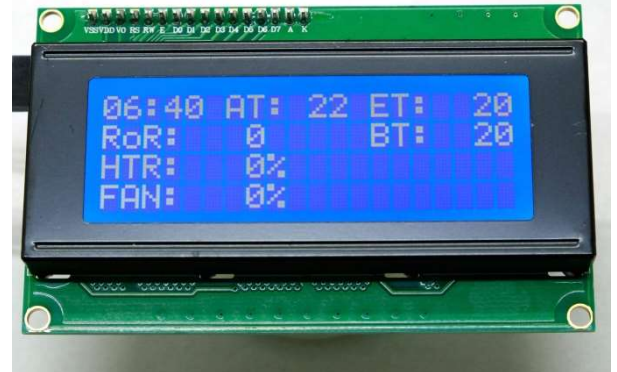
RobotDyn AC Dimmer
(Upgrade heatsink and add forced airflow for controlling >2.5A)

CONFIG_PAC3 Mode
Connection Example

Zero Cross Detector
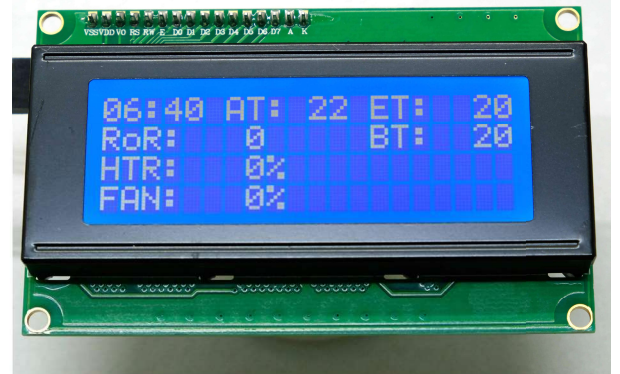
Mains Power

GND
ZC

TC4 Shield + Arduino Uno

TC1
TC2
TC3
TC4

IO2
IO3

GND
ZC

+5V
GND
SDA
SCL

I2C

OT2
OT1

+5V
+5V

ANLG1
ANLG2

POT 2 - Optional

POT 1 - Optional

I2C LCD - Optional

GND
+5V
SDA
SCL

```
06:40 AT: 22 ET:    20
RoR:     0      BT:  20
HTR:     0%
FAN:     0%
```

Instantaneous Turn-on SSR

CRYDOM
SOLID-STATE RELAY
D2425-10

Zero Voltage Turn-on SSR

crydom
SOLID-STATE RELAY
D2425

CONFIG_PAC2_IO3HTR Mode
Connection Example

Zero Cross Detector

Mains Power

GND
ZC

TC4 Shield + Arduino Uno

TC1
TC2
TC3
TC4

ANLG1
ANLG2
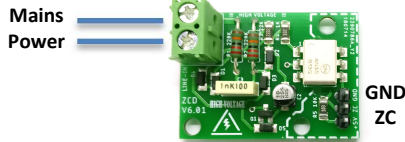
Proportional Gas Valve and Driver

IO2 — GND / ZC
IO3 — GND / +5V
I2C — +5V / GND / SDA / SCL
— GND / +5V / SDA / SCL
OT2 — +5V
OT1

I2C LCD - Optional

```
06:40 AT: 22 ET:   20
RoR:      0      BT: 20
HTR:      0%
FAN:      0%
```
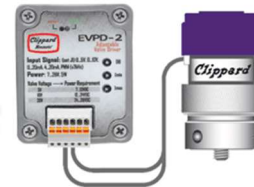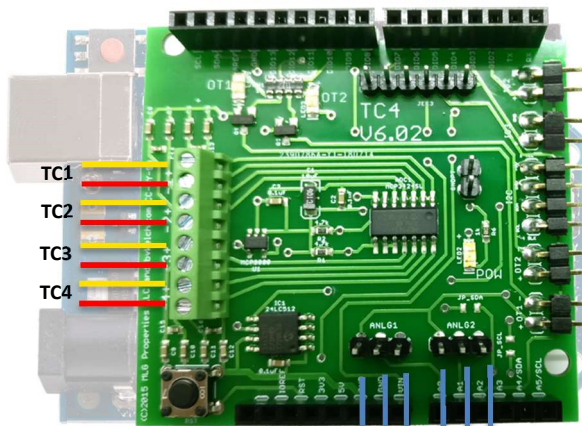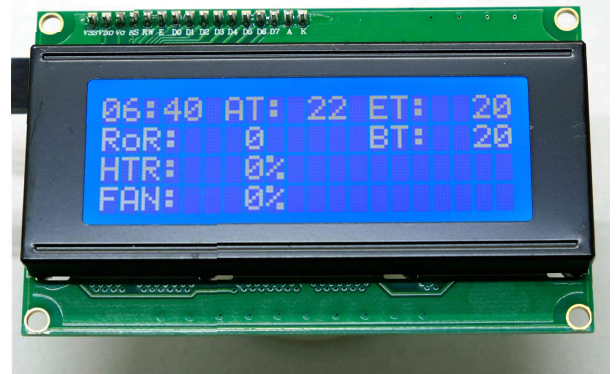
Instantaneous Turn-on SSR

POT 2 - Optional

POT 1 - Optional

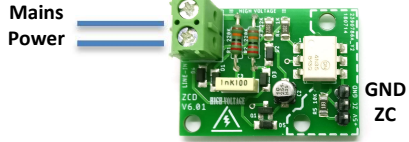# CONFIG_PAC2_IO3FAN Mode
Connection Example
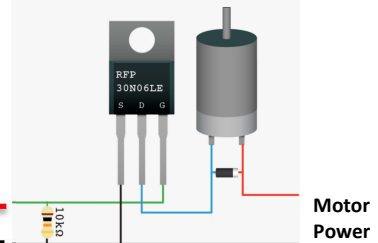
Zero Cross Detector

Mains
Power

GND
ZC

Logic Level MOSFET Motor Driver Circuit

RFP 30N06LE

10kΩ

Motor Power

TC4 Shield + Arduino Uno

TC1
TC2
TC3
TC4

IO2 — GND / ZC
IO3 — GND / +5V
I2C — +5V / GND / SDA / SCL
OT2
OT1 — +5V

GND
+5V
SDA
SCL

I2C LCD - Optional

```
06:40 AT: 22 ET:  20
RoR:     0    BT:  20
HTR:    0%
FAN:    0%
```

ANLG1
ANLG2

POT 2 - Optional

POT 1 - Optional

Zero Voltage Turn-on SSR

crydom
SOLID-STATE RELAY
D2425
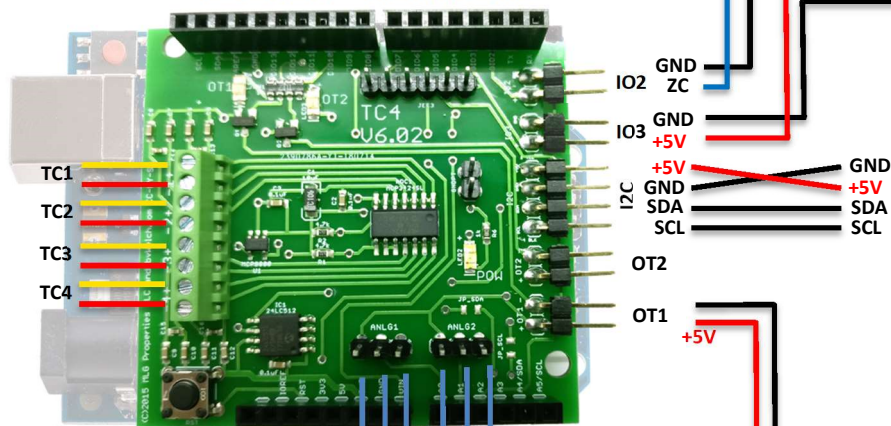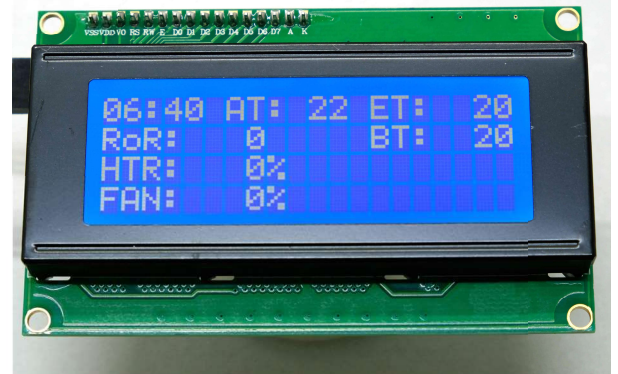
CONFIG_PWM Mode
Connection Example
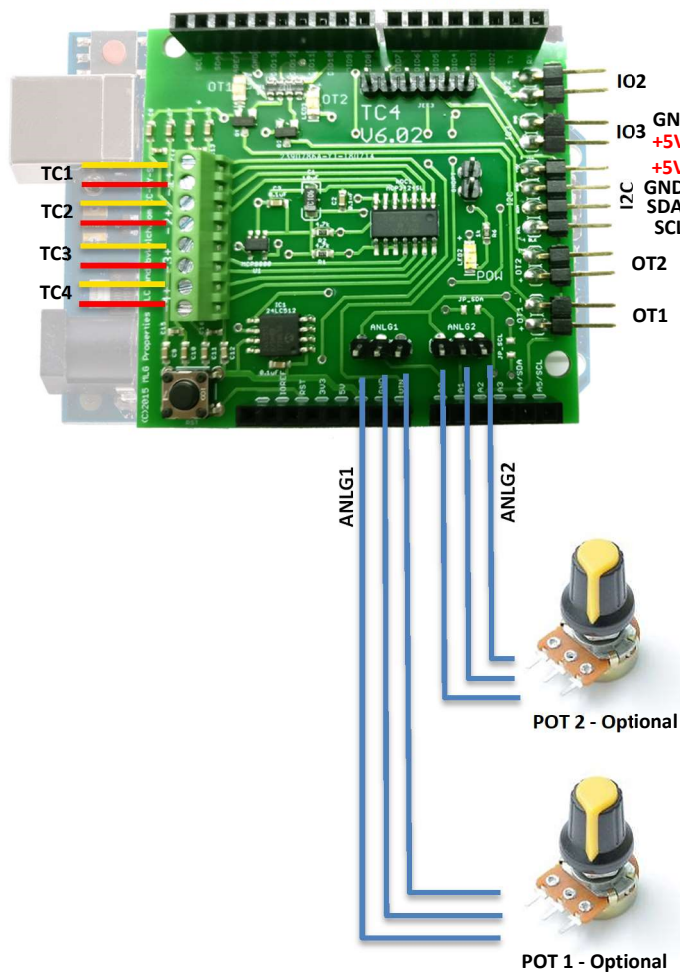
Logic Level MOSFET Motor Driver Circuit

RFP 30N06LE

S D G

10kΩ

Motor Power

TC4 Shield + Arduino Uno

TC4 V6.02

TC1
TC2
TC3
TC4

IO2
IO3
GND
+5V
I2C
+5V
GND
SDA
SCL
OT2
OT1
+5V

GND
+5V
SDA
SCL

I2C LCD - Optional

```
06:40 AT: 22 ET:   20
RoR:     0       BT: 20
HTR:    0%
FAN:    0%
```

ANLG1
ANLG2

POT 2 - Optional

POT 1 - Optional

Zero Voltage Turn-on SSR

crydom SOLID-STATE RELAY
D2425
240V~ OUTPUT 25A
INPUT 3-32V
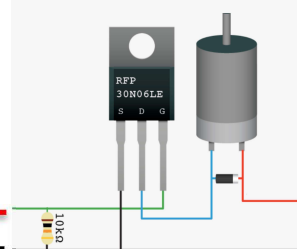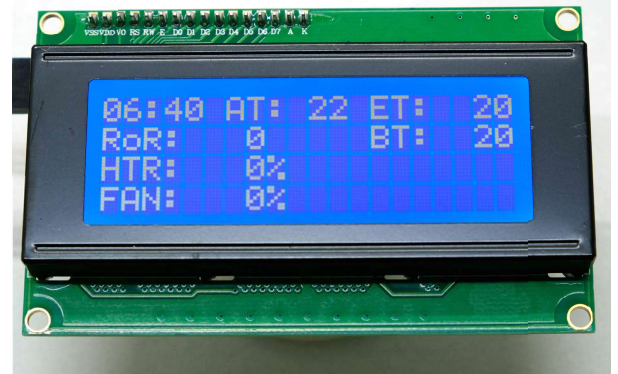
Setup Information for aArtisanQ_PID.ino (v6_8) for TC4-Shield

user.h Options

Introduction

The aArtisanQ_PID.ino sketch for the TC4-Shield has many configuration options and can be customized to suit various roasting hardware and logging software. The parameters and options within the user.h file should be reviewed and modified to suit your preferences and hardware setup.

This guide assumes you have a basic knowledge of the Arduino software and can add new libraries and load sketches onto an Arduino. There are many resources online if you need to acquire these skills.

Load the aArtisanQ_PID.ino sketch in the Arduino IDE and select the user.h tab. After reviewing and editing the user.h parameters and options, compile the sketch and upload it to the Arduino.

Most sections in the user.h files have multiple options to choose from. The inactive options are preceded by a double backslash //. The // disables that line of code by turning it into a comment. To change an option, add a // to the start of the current option and remove the // from the desired option.

The following information should help you choose the most appropriate parameters and options.

Roasting Software

This section sets the logging software being used if any. Activate the appropriate option or disable all options to use the TC4 in stand-alone mode. This option determines how the TC4 communicates with and responds to the software.

The aArtisanQ_PID.ino sketch was initially written to support the Artisan Roaster Scope software, however, the RoastLogger software is also supported. This software typically connects to an Arduino/TC4 via a USB cable and allows real-time logging and control for your roaster.

The ANDROID option is a variation of the ARTISAN option and was intended to support the TC4 Roasting app available for Android devices on the Google Play store. This requires a Bluetooth module connected to your Arduino serial pins.

Stand-alone mode (all options disabled) allows the use of input buttons connected to the Arduino input pins or on an LCDapter board. In standalone mode, an LCD screen connected to the TC4 is recommended.

Base Configurations

The type of roaster you are using will determine what option is chosen in this important section. The main consideration is the type of output required to control your roaster (PWM or phase angle control etc). If you are logging data only, this option is irrelevant.

The Configuration Options and Connection Requirements tables provide additional advice and examples to help you choose the correct configuration option.

Temperature Unit

This option changes the temperature units for data sent to the logging software or displayed on a connected LCD. Leave it disabled for Fahrenheit or enabled for Celsius.

LCD Options

Enable the option for the type of LCD screen you are using. If not using an LCD screen, disable all three types.

- The LCDapter is and I2C adapter board that includes support for four input buttons and 4 LEDs. An LCD display must be purchased separately and connected to the LCDapter board
- The LCD_I2C option is for LCD screens with an included I2C module and can be found cheaply at online stores.
- The LCD_PARALLEL option is for an LCD screen connected directly to the Arduino output pins. This option is not recommended.

The sketch supports both 16x2 and 20x4 LCDs. Disable the LCD_4x20 option if using a 16x2 display. A 20x4 LCD is recommended as it allows the TC4 to display a more useful set of data.

If using the LCD_I2C option, you must correctly set the LCD_I2C_ADDRESS value. Many of these screens use the 0x27 address, so try that first.

Input Button Options

If using the TC4 in stand-alone mode (no logging software) and not using a LCDapter board, you can assign unused Arduino input pins to certain tasks. Input buttons can be used to:

- Reset the roast timer / next profile
- Toggle PID on/off / previous profile
- Switch LCD display mode
- Enter button

AC Power Options

If controlling AC powered electric roasters using Phase Angle Control and Integral Cycle Control, you must specify the mains power frequency your country/state uses. Enable either the 50Hz or 60Hz option.

Thermocouple Input Options

In this section you can specify the type of thermocouple being used for each of the input channels. Most commonly used thermocouples for coffee roasting are typeK, however you can specify typeT or typeJ depending of the thermocouple you are using.

BAUD Rate

If you are using roast logging or control software you should review the baud rate setting. The rate specified here must be used in the logging software. The default baud rate of 115200 should be fine in most cases.

Analogue inputs

The TC4 shield has pin headers for two analogue inputs. Potentiometers can be connected to these headers to control heater and blower power.

Activate the option corresponding to any analogue inputs you are using. If not using one or both of the analogue inputs, leave them disabled, otherwise the inputs will float causing random output power changes.

Duty Cycle Adjustment Increment

This setting can be adjusted to change the increment value for output power changes when using the analogue inputs and power UP/DOWN commands.

Physical input channel for RoR display on LCD

The TC4 can display the Rate of Rise value for an input channel on an LCD if used. You can adjust the number here to choose which channel's RoR gets displayed.

PID Control Options

This section allows you to enable or disable the PID control feature and associated channels and parameters. Having the PID feature enabled does not mean it will be permanently active. With the feature enabled, the PID can still be turned on and off as required using a serial command or input button.

When enabled and active, the PID will control the output associated with the heater.

Disable the PID_CONTROL line if you wan to permanently disable this feature.

The PID_CHAN setting allows you to specify what thermocouple channel is used for the PID input. For example, if you want the PID to follow a bean temp profile, enter the channel number corresponding to your bean temp thermocouple.

Adjust the PID cycle time (CT) if required however, 1 second (1000ms) should be fine.

The PID algorithm relies on the three parameters PRO, INT and DER. Different values will change how well the PID can track a roast profile. Change these three values to suit your roaster.

There are two variations of the PID algorithm to choose from; Proportional on Error or Proportional on Measurement. Enable the POM line to activate the Proportional on Measurement mode. You will have to adjust the PID parameters when changing between Proportional on Error and Proportional on Measurement.

In stand-alone mode (no logging software) the PID can follow roast profiles stored on the TC4s on-board EEPROM. These profiles are loaded onto the EEPROM using the Profile_Loader.ino sketch. Adjust the NUM_PROFILES setting to let the aArtisanQ_PID sketch know how many profiles are stored in the EEPROM.

Heater and Fan Limits/Options

This section of the settings allows you to limit the output range for each output.

The HTR_CUTOFF_FAN_VAL setting allows you to automatically switch off power to your heater if the blower/fan output level goes below this specified value. This is primarily intended to protect the electric element from burning out in an air roaster if the blower output gets set too low. Leave this set at zero if this feature is not being used.

When using the PID;STOP serial command, the blower/fan output will automatically be set to the value specified here under the FAN_AUTO_COOL setting.

Command Echo

Enabling this option will result in all serial commands received from the roasting software to be displayed on the LCD for a short time. Useful for debugging.

Temperature Reading Filters

Filtering is applied to the raw temperature readings and the amount of filtering is specified in this section. The default values should be fine but can be increased or decreased depending on how noisy the raw temperature readings are. The same applies to the filtering of the RoR values.

Thermocouple inputs

This option is to specify how many physical channels exist on your TC4. You should not need to adjust this.

Calibration Values

Thermocouple calibration data can be stored on the TC4 EEPROM. The default vales here will be used if no data is in the EEPROM. Adjusting these is optional but they may help you get more accurate temperature readings.

Time Base for slow PWM on OT1, OT2

When using the CONFIG_PWM mode, the default output frequency for the OT1 and OT2 outputs is 1Hz. For example, at 50% power, the output will be on for half a second and then off for half a second and so on. You can change the frequency

between 1/4 Hz and 8Hz by activating the desired line. Frequencies faster than 8Hz may not work properly without other code changes.

Debugging Options

No change required from default.

Output Pin Setup

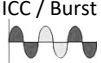The settings in this section should not need changing. They map the TC4 output ports to the Arduino pins.
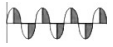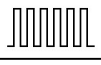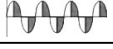
Set up parameters for the base configurations

The settings in this section should not need changing. These are set up depending on the configuration options specified earlier.

Phase Angle Control Options

The settings in this section should not need changing unless using the outputs in a different manner than usual.

Configuration Options and Connection Requirements

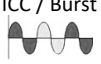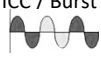| Configuration Mode	Heating				Airflow				Zero Cross Detector		Comments
	Typical Power Source	Output Type	TC4 / TC4C Port	Interface Suggestion	Typical Power Source	Output Type	TC4 / TC4C Port	Interface Suggestion	Required?	TC4 / TC4C Port	
CONFIG_PWM	AC Mains	Slow PWM	OT1	SSR Zero Voltage Turn-on type	DC	PWM	IO3	MOSFET MOSFET Driver	No	N/A	- Slow PWM on OT1 - Default 1Hz but adjustable between 1/4Hz to 8Hz in user.h (up to 3.9kHz with other code changes) - Fast PWM on IO3 - Default 3.9kHz but adjustable between 30.64Hz and 62.5kHz - Also allows slow PWM output on OT2 (same frequency as OT1)
CONFIG_PAC2	AC Mains	ICC / Burst	OT1	SSR Zero Voltage Turn-on type	AC Mains	PAC	OT2	SSR Instantaneous Turn-on type	Yes	IO2	Also allows fast PWM output on IO3 - Default 3.9kHz but adjustable between 30.64Hz and 62.5kHz
CONFIG_PAC2_IO3HTR	Gas	PWM	IO3	Solenoid interface circuit	AC Mains	PAC	OT2	SSR Instantaneous Turn-on type	Yes	IO2	Fast PWM on IO3 - Default 3.9kHz but adjustable between 30.64Hz and 62.5kHz
CONFIG_PAC2_IO3FAN	AC Mains	ICC / Burst	OT1	SSR Zero Voltage Turn-on type	DC	PWM	IO3	MOSFET MOSFET Driver	Yes	IO2	Similar to CONFIG_PWM but with ICC heater control
CONFIG_PAC3	AC Mains	ICC / Burst	OT1	SSR Zero Voltage Turn-on type	AC Mains	PAC	OT2	SSR Instantaneous Turn-on type	Yes	IO3	- Same as CONFIG_PAC2 but with ZCD connected to IO3 and no fast PWM output

PWM = Pulse Width Modulation
PAC = Phase Angle Control
ICC = Integral Cycle Control
SSR = Solid State Relay

Configuration Option Usage Examples

Configuration Mode	Typical Roaster Types Used	
CONFIG_PWM	**Small Popcorn Machine Roaster** - AC heating element connected to OT1 via a standard Zero Voltage Turn-on SSR - DC fan connected to IO3 port via an appropriate driver circuit	**Gas Powered Roaster (for advanced users only - requires changes to TC4 library files)** - Gas heating with gas valve connected to OT1 via an appropriate valve driver circuit - OT1 PWM frequency adjusted to 2.2kHz (requires change to PWM16 library) - DC fan connected to IO3 port via an appropriate driver circuit
CONFIG_PAC2	**Electric Air Roaster** - AC heating element connected to OT1 via standard Zero Voltage Turn-on SSR - AC Fan/Blower connected to OT2 via a Instantaneous Turn-on SSR - Zero Voltage Turn-on detector connected to IO2	**Electric Drum Roaster** - AC heating element connected to OT1 via standard Zero Voltage Turn-on SSR - AC Fan/Blower connected to OT2 via a Instantaneous Turn-on SSR - Zero Voltage Turn-on detector connected to IO2 - Optional DC drum rotation motor connected to IO3 via an appropriate driver circuit
CONFIG_PAC2_IO3HTR	**Gas Powered Roaster** - Gas heating with gas valve connected to IO3 via valve driver circuit - AC Fan/Blower connected to OT2 via a Instantaneous Turn-on SSR - Zero cross detector connected to IO2	
CONFIG_PAC2_IO3FAN	**Small Popcorn Machine Roaster** - AC heating element connected to OT1 via a standard Zero Voltage Turn-on SSR - DC fan connected to IO3 port via an appropriate driver circuit	
CONFIG_PAC3	**Electric Air Roaster** - AC heating element connected to OT1 via standard Zero Voltage Turn-on SSR - AC Fan/Blower connected to OT2 via a Instantaneous Turn-on SSR - Zero Voltage Turn-on detector connected to IO3	**Electric Drum Roaster** - AC heating element connected to OT1 via standard Zero Voltage Turn-on SSR - AC Fan/Blower connected to OT2 via a Instantaneous Turn-on SSR - Zero Voltage Turn-on detector connected to IO3

Serial Commands for aArtisanQ_PID

Notes:

1. Terminate all command strings with newline, i.e. /n
2. Delimiters between parameters may be comma, space, semicolon, or equals sign
3. First five characters (max) of command name are significant
4. A command may have up to 5 parameters, including the keyword

CHAN,ijkl

where i,j,k,l = a decimal value 0 to 4 representing a physical port

Establishes the active logical channels (chan1, chan2, chan3, chan4) and maps them to physical ports (TC1, TC2, TC3, TC4).

The Artisan Roaster Scope software expects environment temp (ET) on logical chan1, and bean temp (BT) on logical chan2.

A value of zero for the physical port inactivates the channel.

CHAN,ijkl examples:

chan,1200/n TC1 -> chan1, TC2 -> chan2, chan3 and chan4 inactive
chan,3210/n TC3 -> chan1, TC2 -> chan2, TC1 -> chan3, chan4 inactive

Response from TC4 device is "# Active channels set to ijkl"

FILT,lev1,lev2,lev3,lev4

Sets new digital filtering level (0-100 percent) on logical channels 1, 2, 3, and 4.

lev1 = new filtering level for logical channel 1 (etc)

PID,T,ppp,iii,ddd and PID,T_POM,ppp,iii,ddd

Sets new PID parameters ("T" is for "tuning"):

ppp = proportional coefficient Kp

iii = integral coefficient Ki

ddd = derivative coefficient Kd

T = Proportional on Error PID mode

T_POM = Proportional on Measurement PID mode

PID,LIMIT,min,max

Sets PID output limits

min = lower limit (0-100 and < max)

max = upper limit (0-100 and > min)

PID,SV,vvv

Establishes new PID setpoint (vvv = set value, or SV)

PID,ON and PID,OFF

Turns the PID on or off. PID always operates on OT1 (or IO3 in CONFIG_PAC2_IO3HTR mode).

PID,CT,mmmm

Sets new PID cycle time, in ms, to mmmm. Default value is 1000 (1 second).

PID,CHAN,i

Sets the input for the PID controller.
i = a decimal value 1 to 4 representing a physical port (TC1, TC2, TC3, TC4).

Example:

CHAN,2100/n
PID,CHAN,2/n

This results in the thermocouple connected to physical port TC2 being used by the PID. And in this example, TC2 was mapped to logical channel chan1, which corresponds to ET when using Artisan Roasting Scope

PID,START

Resets TC4 timer and activates the PID.

Can be used when starting a roast. Response from TC4 device is "# PID Roast Start"

PID,STOP

Turns off the PID and heater. Sets fan to auto cool setting as defined in user.h.

Can be used at end of roast. Response from TC4 device is "# PID Roast Stop"

PID,Px

Sets the roast profile being used by the PID to val.

x = 0 used when receiving set variable values with the PID,SV command. Default condition.
x > 0 to access corresponding roast profile stored on TC4 EEPROM

Response from TC4 device is "# Profile 'val' selected"

DCFAN,duty

where duty = 0 to 100 (percent output)

Changes the PWM duty cycle on FAN_PORT to duty. By default, FAN_PORT is DI03 on Arduino (same as IO3 port).

Limits the increase in duty to 25 points per second to address fan inrush current on Hottop (and possibly other) roasters.

No response from TC4 device.

UNITS,u

where u = C or F.

Sets active temperature scale. No response from TC4 device.

READ

Requests current temperature readings on all active channels.

Response from TC4 device depends on the roaster software being used. For Artisan Roaster Scope, this will be the ambient temperature followed by a comma separated list of temperatures in current active units in logical channel order, then followed by the heater, fan and PID set value:

Eg: ambient,chan1,chan2,heater,fan,SV

OT1,duty

where duty = 0 to 100 (percent output)

Changes the output duty cycle on OT1 to duty.

No response from TC4 device.

OT1,up and OT1,down

Adjusts OT1 duty cycle up or down by DUTY_STEP increment as defined in user.h.

No response from TC4 device.

OT2,duty

where duty = 0 to 100 (percent output)

Changes the output duty cycle on OT2 to duty.

No response from TC4 device.

OT2,up and OT2,down

Adjusts OT2 duty cycle up/down by DUTY_STEP increment as defined in user.h.

No response from TC4 device.

IO3,duty (not available in CONFIG_PAC3 mode)

where duty = 0 to 100 (percent output)

Changes the PWM duty cycle on IO3 to duty.

Unlike the DCFAN command, the duty cycle is increased immediately. Sudden increases in fan speed is known to cause problems on some hottop roasters, so users are advised to use the dcfan command instead.

No response from TC4 device.

IO3,up and IO3,down (not available in CONFIG_PAC3 mode)

Adjusts IO3 duty cycle up or down by DUTY_STEP increment as defined in user.h.

No response from TC4 device.

DWRITE,pin,val

where pin = Arduino pin number (D0 to D13 or A0 to A5)
val = HIGH or LOW

Puts port associated with pin number in output mode and sets it HIGH or LOW.

No response from the TC4 device.

AWRITE,pin,level

where pin = Arduino pin number (D0 to D13)
val = 0 to 255

Puts port associated with pin number in output mode and sets output level 0 to 255 (i.e. duty cycle 0 to 100%).

No response from the TC4 device.

Reset

Resets TC4 timer

Response from TC4 device is "# Reset"

LOAD (available in Roastlogger mode)

Resets TC4 timer

No response from the TC4 device.

POWER=duty (available in Roastlogger mode)

where duty = 0 to 100 (percent output)

Changes the heater duty cycle on OT1 or IO3 to duty.

No response from TC4 device.

FAN=duty (available in Roastlogger mode)

where duty = 0 to 100 (percent output)

Changes the duty cycle on OT2 or IO3 to duty.

No response from TC4 device.