

Mode: All Lines

Left file: C:\xxd\myGit\waterbath\waterbath\waterbath.ino

Right file: C:\xxd\myGit\waterbath\Sous_Viduino\Sous_Viduino.ino

| | | | | |
|----|--|----|----|--|
| 1 | //----- » ----- 2 | // | 1 | //----- » ----- 2 |
| 3 | // Water Bath Controller | <> | | |
| 4 | // | | | |
| 5 | // Based oSous Vide Controller | | 3 | // Sous Vide Controller |
| 6 | // Bill Earl - for Adafruit Industries | = | 4 | // Bill Earl - for Adafruit Industries |
| 7 | // | | 5 | // |
| 8 | // Based on the Arduino PID and PID Auto » Tune Libraries | | 6 | // Based on the Arduino PID and PID Auto » Tune Libraries |
| 9 | // by Brett Beauregarda | <> | 7 | // by Brett Beauregard |
| 10 | //----- » ----- 11 | = | 8 | //----- » ----- 9 |
| 12 | // PID Library | | 10 | // PID Library |
| 13 | #include <PID_v1.h> | | 11 | #include <PID_v1.h> |
| 14 | #include <PID_AutoTune_v0.h> | | 12 | #include <PID_AutoTune_v0.h> |
| 15 | | | 13 | |
| 16 | // Libraries for the Adafruit RGB/LCD Sh » ield | | 14 | // Libraries for the Adafruit RGB/LCD Sh » ield |
| 17 | #include <Wire.h> | | 15 | #include <Wire.h> |
| 18 | #include <LiquidCrystal.h> | <> | 16 | #include <Adafruit_MCP23017.h> |
| 19 | | | 17 | #include <Adafruit_RGBLCDShield.h> |
| 20 | // Libraries for the DS18B20 Temperature » Sensor | = | 18 | // Libraries for the DS18B20 Temperature » Sensor |
| 21 | #include <OneWire.h> | | 19 | #include <OneWire.h> |
| 22 | #include <DallasTemperature.h> | | 20 | #include <DallasTemperature.h> |
| 23 | | | 21 | #include <DallasTemperature.h> |
| 24 | // So we can save and retrieve settings | | 22 | |
| 25 | #include <EEPROM.h> | | 23 | // So we can save and retrieve settings |
| 26 | | | 24 | #include <EEPROM.h> |
| 27 | // ***** » ***** | | 25 | |
| 28 | // Pin definitions | | 26 | // ***** » ***** |
| 29 | // ***** » ***** | | 27 | // Pin definitions |
| 30 | | | 28 | // ***** » ***** |
| 31 | // Output Relay | | 29 | |
| 32 | #define RelayPin 13 | <> | 30 | // Output Relay |
| 33 | | | 31 | #define RelayPin 7 |
| 34 | // One-Wire Temperature Sensor | = | 32 | |
| | | | 33 | // One-Wire Temperature Sensor |
| | | + | 34 | // (Use GPIO pins for power/ground to si » mplify the wiring) |
| 35 | #define ONE_WIRE_BUS 2 | = | 35 | #define ONE_WIRE_BUS 2 |
| 36 | | <> | 36 | #define ONE_WIRE_PWR 3 |
| | | | 37 | #define ONE_WIRE_GND 4 |
| 37 | | = | 38 | |
| 38 | // ***** » ***** | | 39 | // ***** » ***** |
| 39 | // PID Variables and constants | | 40 | // PID Variables and constants |
| 40 | // ***** » ***** | | 41 | // ***** » ***** |
| 41 | | | 42 | |
| 42 | //Define Variables we'll be connecting t » o | | 43 | //Define Variables we'll be connecting t » o |

Left file: C:\xxd\myGit\waterbath\waterbath\waterbath.ino

Right file: C:\xxd\myGit\waterbath\Sous_Viduino\Sous_Viduino.ino

(continued)

```

43 double Setpoint;
44 double Input;
45 double Output;
46
47 volatile long onTime = 0;
48
49 // pid tuning parameters
50 double Kp;
51 double Ki;
52 double Kd;
53
54 // EEPROM addresses for persisted data
55 const int SpAddress = 0;
56 const int KpAddress = 8;
57 const int KiAddress = 16;
58 const int KdAddress = 24;
59
60 //Specify the links and initial tuning p
  » arameters
61 PID myPID(&Input, &Output, &Setpoint, Kp
  » , Ki, Kd, DIRECT);
62
63 // 10 second Time Proportional Output wi
  » ndow
64 int WindowSize = 10000;
65 unsigned long windowStartTime;
66
67 // *****
  » *****
68 // Auto Tune Variables and constants
69 // *****
  » *****
70 byte ATuneModeRemember=2;
71
72 double aTuneStep=500;
73 double aTuneNoise=1;
74 unsigned int aTuneLookBack=20;
75
76 boolean tuning = false;
77
78 PID_ATune aTune(&Input, &Output);
79
80 // *****
  » *****
81 // DiSplay Variables and constants
82 // *****
  » *****
83

```

```

84 //Adafruit_RGBLCDShield lcd = Adafruit_R
  » GBLCDShield();
85 // select the pins used on the LCD panel
86 LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
87
88 // define some values used by the panel
  » and buttons
89 int lcd_key    = 0;
90 int adc_key_in = 0;

```

```

44 double Setpoint;
45 double Input;
46 double Output;
47
48 volatile long onTime = 0;
49
50 // pid tuning parameters
51 double Kp;
52 double Ki;
53 double Kd;
54
55 // EEPROM addresses for persisted data
56 const int SpAddress = 0;
57 const int KpAddress = 8;
58 const int KiAddress = 16;
59 const int KdAddress = 24;
60
61 //Specify the links and initial tuning p
  » arameters
62 PID myPID(&Input, &Output, &Setpoint, Kp
  » , Ki, Kd, DIRECT);
63
64 // 10 second Time Proportional Output wi
  » ndow
65 int WindowSize = 10000;
66 unsigned long windowStartTime;
67
68 // *****
  » *****
69 // Auto Tune Variables and constants
70 // *****
  » *****
71 byte ATuneModeRemember=2;
72
73 double aTuneStep=500;
74 double aTuneNoise=1;
75 unsigned int aTuneLookBack=20;
76
77 boolean tuning = false;
78
79 PID_ATune aTune(&Input, &Output);
80
81 // *****
  » *****
82 // DiSplay Variables and constants
83 // *****
  » *****
84

```

```

85 Adafruit_RGBLCDShield lcd = Adafruit_RGB
  » LCDShield();

```

Left file: C:\xxd\myGit\waterbath\waterbath\waterbath.ino

Right file: C:\xxd\myGit\waterbath\Sous_Viduino\Sous_Viduino.ino

(continued)

| | | | | |
|-----|--|----|-----|--|
| 91 | | | | |
| 92 | #define btnNONE 0 | | | |
| 93 | #define BUTTON_UP 1 | | | |
| 94 | #define BUTTON_DOWN 2 | | | |
| 95 | #define BUTTON_LEFT 3 | | | |
| 96 | #define BUTTON_SHIFT 4 | | | |
| 97 | #define BUTTON_RIGHT 5 | | | |
| 98 | | | | |
| 99 | | | | |
| 100 | // These #defines make it easy to set th | = | 86 | // These #defines make it easy to set th |
| | » e backlight color | | | » e backlight color |
| 101 | #define RED 0x1 | | 87 | #define RED 0x1 |
| 102 | #define YELLOW 0x3 | | 88 | #define YELLOW 0x3 |
| 103 | #define GREEN 0x2 | | 89 | #define GREEN 0x2 |
| 104 | #define TEAL 0x6 | | 90 | #define TEAL 0x6 |
| 105 | #define BLUE 0x4 | | 91 | #define BLUE 0x4 |
| 106 | #define VIOLET 0x5 | | 92 | #define VIOLET 0x5 |
| 107 | #define WHITE 0x7 | | 93 | #define WHITE 0x7 |
| | | | | |
| | | <> | 94 | |
| | | | 95 | #define BUTTON_SHIFT BUTTON_SELECT |
| 108 | | = | 96 | |
| 109 | unsigned long lastInput = 0; // last but | | 97 | unsigned long lastInput = 0; // last but |
| | » ton press | | | » ton press |
| 110 | | | 98 | |
| 111 | byte degree[8] = // define the degree sy | | 99 | byte degree[8] = // define the degree sy |
| | » mbol | | | » mbol |
| 112 | { | | 100 | { |
| 113 | B00110, | | 101 | B00110, |
| 114 | B01001, | | 102 | B01001, |
| 115 | B01001, | | 103 | B01001, |
| 116 | B00110, | | 104 | B00110, |
| 117 | B00000, | | 105 | B00000, |
| 118 | B00000, | | 106 | B00000, |
| 119 | B00000, | | 107 | B00000, |
| 120 | B00000 | | 108 | B00000 |
| 121 | }; | | 109 | }; |
| 122 | | | 110 | |
| 123 | const int logInterval = 10000; // log ev | | 111 | const int logInterval = 10000; // log ev |
| | » ery 10 seconds | | | » ery 10 seconds |
| 124 | long lastLogTime = 0; | | 112 | long lastLogTime = 0; |
| 125 | | | 113 | |
| 126 | // ***** | | 114 | // ***** |
| | » ***** | | | » ***** |
| 127 | // States for state machine | | 115 | // States for state machine |
| 128 | // ***** | | 116 | // ***** |
| | » ***** | | | » ***** |
| 129 | enum operatingState { OFF = 0, SETP, RUN | | 117 | enum operatingState { OFF = 0, SETP, RUN |
| | » , TUNE_P, TUNE_I, TUNE_D, AUTO}; | | | » , TUNE_P, TUNE_I, TUNE_D, AUTO}; |
| 130 | operatingState opState = OFF; | | 118 | operatingState opState = OFF; |
| 131 | | | 119 | |
| 132 | // ***** | | 120 | // ***** |
| | » ***** | | | » ***** |
| 133 | // Sensor Variables and constants | | 121 | // Sensor Variables and constants |
| 134 | // Data wire is plugged into port 2 on t | | 122 | // Data wire is plugged into port 2 on t |
| | » he Arduino | | | » he Arduino |
| 135 | | | 123 | |
| 136 | // Setup a oneWire instance to communica | | 124 | // Setup a oneWire instance to communica |

Left file: C:\xxd\myGit\waterbath\waterbath\waterbath.ino

Right file: C:\xxd\myGit\waterbath\Sous_Viduino\Sous_Viduino.ino

(continued)

| | | | | | |
|-----|--|----|-----|--|--|
| 137 | » te with any OneWire devices (not just » Maxim/Dallas temperature ICs) | | 125 | » te with any OneWire devices (not just » Maxim/Dallas temperature ICs) | |
| 138 | OneWire oneWire(ONE_WIRE_BUS); | | 126 | OneWire oneWire(ONE_WIRE_BUS); | |
| 139 | // Pass our oneWire reference to Dallas » Temperature. | | 127 | // Pass our oneWire reference to Dallas » Temperature. | |
| 140 | DallasTemperature sensors(&oneWire); | | 128 | DallasTemperature sensors(&oneWire); | |
| 141 | | | 129 | | |
| 142 | // arrays to hold device address | | 130 | // arrays to hold device address | |
| 143 | DeviceAddress tempSensor; | | 131 | DeviceAddress tempSensor; | |
| 144 | | | 132 | | |
| 145 | // ***** | | 133 | // ***** | |
| 146 | » ***** | | 134 | » ***** | |
| 147 | // Setup and diSplay initial screen | | 135 | // Setup and diSplay initial screen | |
| 148 | // ***** | | 136 | // ***** | |
| 149 | » ***** | | 137 | » ***** | |
| 150 | void setup() | | 138 | void setup() | |
| 151 | { | | 139 | { | |
| 152 | Serial.begin(9600); | | 140 | Serial.begin(9600); | |
| 153 | | +- | 141 | | |
| 154 | | = | 142 | | |
| 155 | // Initialize Relay Control: | | 143 | // Initialize Relay Control: | |
| 156 | pinMode(RelayPin, OUTPUT); // Outp | | 144 | pinMode(RelayPin, OUTPUT); // Outp | |
| 157 | » ut mode to drive relay | | 145 | » ut mode to drive relay | |
| | digitalWrite(RelayPin, LOW); // make | | 146 | digitalWrite(RelayPin, LOW); // make | |
| | » sure it is off to start | | 147 | » sure it is off to start | |
| | | | 148 | | |
| | | <> | 149 | | |
| | | | 150 | // Set up Ground & Power for the sens | |
| | | | 151 | » or from GPIO pins | |
| | | | 152 | | |
| | | | 153 | pinMode(ONE_WIRE_GND, OUTPUT); | |
| | | | 154 | digitalWrite(ONE_WIRE_GND, LOW); | |
| | | | 155 | | |
| | | | 156 | pinMode(ONE_WIRE_PWR, OUTPUT); | |
| | | | 157 | digitalWrite(ONE_WIRE_PWR, HIGH); | |
| 158 | // Initialize LCD DiSplay | = | 158 | // Initialize LCD DiSplay | |
| 159 | lcd.begin(16, 2); | | 159 | lcd.begin(16, 2); | |
| 160 | lcd.createChar(1, degree); // create | | 160 | lcd.createChar(1, degree); // create | |
| 161 | » degree symbol from the binary | | 161 | » degree symbol from the binary | |
| 162 | | | 162 | | |
| 163 | lcd.clear(); | <> | 163 | lcd.setBacklight(VIOLET); | |
| 164 | //lcd.setBacklight(VIOLET); | | 164 | lcd.print(F(" Adafruit")); | |
| 165 | lcd.setCursor(0, 0); | | 165 | lcd.setCursor(0, 1); | |
| 166 | lcd.print(F(" Water Bath")); | | 166 | lcd.print(F(" Sous Vide!")); | |
| 167 | lcd.setCursor(0, 1); | = | 167 | | |
| 168 | lcd.print(F(" by Jill Xu")); | <> | 168 | | |
| 169 | | = | 169 | | |
| 170 | | <> | 170 | | |
| 171 | // Start up the DS18B20 One Wire Temp | = | 171 | // Start up the DS18B20 One Wire Temp | |
| 172 | » erature Sensor | | 172 | » erature Sensor | |
| 173 | sensors.begin(); | | 173 | sensors.begin(); | |
| 174 | if (!sensors.getAddress(tempSensor, 0 | | 174 | if (!sensors.getAddress(tempSensor, 0 | |
| | »)) | | | »)) | |

Left file: C:\xxd\myGit\waterbath\waterbath\waterbath.ino

Right file: C:\xxd\myGit\waterbath\Sous_Viduino\Sous_Viduino.ino

(continued)

| | | | | |
|-----|--|----|-----|---------------------------------------|
| 175 | { | | 167 | { |
| 176 | lcd.setCursor(0, 1); | | 168 | lcd.setCursor(0, 1); |
| 177 | LcdClearLine(1); | +- | | |
| 178 | lcd.print(F("Sensor Error")); | = | 169 | lcd.print(F("Sensor Error")); |
| 179 | } | | 170 | } |
| 180 | sensors.setResolution(tempSensor, 12) | | 171 | sensors.setResolution(tempSensor, 12) |
| 181 | » ; | | 172 | » ; |
| 182 | sensors.waitForConversion(false); | | 173 | sensors.waitForConversion(false); |
| 183 | delay(3000); // Splash screen | | 174 | delay(3000); // Splash screen |
| 184 | | | 175 | |
| 185 | | <> | | |
| 186 | | | | |
| 187 | // Initialize the PID and related var | = | 176 | // Initialize the PID and related var |
| 188 | » iables | | 177 | » iables |
| 189 | LoadParameters(); | | 178 | LoadParameters(); |
| 190 | myPID.SetTunings(Kp,Ki,Kd); | | 179 | myPID.SetTunings(Kp,Ki,Kd); |
| 191 | myPID.SetSampleTime(1000); | | 180 | myPID.SetSampleTime(1000); |
| 192 | myPID.SetOutputLimits(0, WindowSize); | | 181 | myPID.SetOutputLimits(0, WindowSize); |
| 193 | | <> | 182 | |
| 194 | | | | |
| 195 | // Run timer1 interrupt every 15 ms (r | | 183 | // Run timer2 interrupt every 15 ms |
| 196 | » oughly 67Hz) | | | |
| 197 | noInterrupts(); // disable a | | | |
| 198 | » ll interrupts | | | |
| 199 | TCCR1A = 0; | | 184 | TCCR2A = 0; |
| 200 | TCCR1B = 0; | | 185 | TCCR2B = 1<<CS22 1<<CS21 1<<CS20; |
| 201 | TCNT1 = 0; | | | |
| 202 | | = | 186 | |
| 203 | // check the following link how to calcu | <> | 187 | //Timer2 Overflow Interrupt Enable |
| 204 | » late OCR1A | | | |
| 205 | // http://www.instructables.com/id/Ardui | | | |
| 206 | » no-Timer-Interrupts/step1/Prescalers-a | | | |
| 207 | » nd-the-Compare-Match-Register/ | | | |
| 208 | | | | |
| 209 | OCR1A = 932; // compare mat | | | |
| 210 | » ch register 16MHz/256/67Hz | | | |
| 211 | TCCR1B = (1 << WGM12); // CTC mode | | | |
| 212 | TCCR1B = (1 << CS12); // 256 presc | | | |
| | » aler | | | |
| | TIMSK1 = (1 << OCIE1A); // enable ti | | | |
| | » mer compare interrupt | | | |
| | | | 188 | TIMSK2 = 1<<TOIE2; |
| | interrupts(); // enable al | | | |
| | » l interrupts | | | |
| 211 | } | = | 189 | } |
| 212 | | | 190 | |
| 213 | | <> | 191 | // ***** |
| 214 | ISR(TIMER1_COMPA_vect) // timer | | 192 | » ***** |
| 215 | » compare interrupt service routine | | 193 | // Timer Interrupt Handler |
| | | | 194 | // ***** |
| | | | | » ***** |
| | | | | SIGNAL(TIMER2_OVF_vect) |
| 215 | { | = | 195 | { |
| 216 | //Serial.print("millis in timer1 int: | +- | | |

Left file: C:\xxd\myGit\waterbath\waterbath\waterbath.ino

Right file: C:\xxd\myGit\waterbath\Sous_Viduino\Sous_Viduino.ino

(continued)

| | | | | |
|-----|--|----|-----|--|
| 217 | » "); | | | |
| 218 | //Serial.println(millis()); | | | |
| 219 | if (opState == OFF) | = | 196 | if (opState == OFF) |
| 220 | { | | 197 | { |
| 221 | digitalWrite(RelayPin, LOW); // mak | | 198 | digitalWrite(RelayPin, LOW); // mak |
| 222 | » e sure relay is off | | 199 | » e sure relay is off |
| 223 | } | | 200 | } |
| 224 | else | | 201 | else |
| 225 | { | | 202 | { |
| 226 | DriveOutput(); | | 203 | DriveOutput(); |
| 227 | } | | 204 | } |
| 228 | } | | 205 | } |
| 229 | void printStatus() | <> | | |
| 230 | { | | | |
| 231 | char buf[32]; // needs to be at least | | | |
| 232 | » large enough to fit the formatted text | | | |
| 233 | dtostrf(Setpoint, 2, 2, buf); | | | |
| 234 | String parameters = String("Setpoint=" | | | |
| 235 | »)+String(buf) + " "; | | | |
| 236 | dtostrf(Kp, 2, 2, buf); | | | |
| 237 | parameters += String("Kp=")+String(buf | | | |
| 238 | ») + " "; | | | |
| 239 | dtostrf(Kd, 2, 2, buf); | | | |
| 240 | parameters += String("Kd=")+String(buf | | | |
| 241 | ») + " "; | | | |
| 242 | dtostrf(Ki, 2, 2, buf); | | | |
| 243 | parameters += String("Ki=")+String(buf | | | |
| 244 | »); | | | |
| 245 | Serial.println(parameters); | | | |
| 246 | | | | |
| 247 | sensors.requestTemperatures(); | | | |
| 248 | | | | |
| 249 | Serial.print("The temperature from sen | | | |
| 250 | » sor is: "); | | | |
| 251 | Serial.println(sensors.getTempCByIndex | | | |
| 252 | » (0)); | | | |
| 253 | Serial.print("Input and Output: "); | | | |
| 254 | Serial.print(Input); | | | |
| 255 | Serial.print(", "); | | | |
| 256 | Serial.println(Output); | | | |
| 257 | } | | | |
| 258 | | | | |
| 259 | // ***** | = | 206 | // ***** |
| 260 | » ***** | | 207 | » ***** |
| 261 | // Main Control Loop | | 208 | // Main Control Loop |
| 262 | // | | 209 | // |
| 263 | // All state changes pass through here | | 210 | // All state changes pass through here |
| | // ***** | | | // ***** |

Left file: C:\xxd\myGit\waterbath\waterbath\waterbath.ino

Right file: C:\xxd\myGit\waterbath\Sous_Viduino\Sous_Viduino.ino

(continued)

| | | | | |
|-----|---|----|-----|---------------------------------------|
| 264 | void loop() | | 211 | void loop() |
| 265 | { | | 212 | { |
| 266 | //printStatus(); | +- | | |
| 267 | | | | |
| 268 | | | | |
| 269 | | | | |
| 270 | // wait for button release before cha | = | 213 | // wait for button release before cha |
| 271 | » nging state | | 214 | » nging state |
| 272 | while(ReadButtons() != 0) {} | | 215 | while(ReadButtons() != 0) {} |
| 273 | lcd.clear(); | | 216 | lcd.clear(); |
| 274 | | | 217 | |
| 275 | switch (opState) | | 218 | switch (opState) |
| 276 | { | | 219 | { |
| 277 | case OFF: | | 220 | case OFF: |
| 278 | Off(); | | 221 | Off(); |
| 279 | break; | | 222 | break; |
| 280 | case SETP: | | 223 | case SETP: |
| 281 | Tune_Sp(); | | 224 | Tune_Sp(); |
| 282 | break; | | 225 | break; |
| 283 | case RUN: | | 226 | case RUN: |
| 284 | Run(); | | 227 | Run(); |
| 285 | break; | | 228 | break; |
| 286 | case TUNE_P: | | 229 | case TUNE_P: |
| 287 | TuneP(); | | 230 | TuneP(); |
| 288 | break; | | 231 | break; |
| 289 | case TUNE_I: | | 232 | case TUNE_I: |
| 290 | TuneI(); | | 233 | TuneI(); |
| 291 | break; | | 234 | break; |
| 292 | case TUNE_D: | | 235 | case TUNE_D: |
| 293 | TuneD(); | | 236 | TuneD(); |
| 294 | break; | | 237 | break; |
| 295 | } | | 238 | } |
| 296 | | +- | | |
| 297 | | | | |
| 298 | } | = | 239 | } |
| 299 | | | 240 | |
| 300 | // read the buttons | <> | | |
| 301 | int read_LCD_buttons() | | | |
| 302 | { | | | |
| 303 | adc_key_in = analogRead(0); // rea | | | |
| 304 | » d the value from the sensor | | | |
| 305 | // my buttons when read are centered at | | | |
| 306 | » these valies: 0, 144, 329, 504, 741 | | | |
| 307 | // we add approx 50 to those values and | | | |
| 308 | » check to see if we are close | | | |
| 309 | if (adc_key_in > 1000) return btnNONE; | | | |
| 310 | » // We make this the 1st option for spe | | | |
| | » ed reasons since it will be the most l | | | |
| | » ikely result | | | |
| | // For V1.1 us this threshold | | | |
| | if (adc_key_in < 50) return BUTTON_RI | | | |
| | » GHT; | | | |
| | if (adc_key_in < 250) return BUTTON_UP | | | |
| | » ; | | | |
| | if (adc_key_in < 450) return BUTTON_DO | | | |

Left file: C:\xxd\myGit\waterbath\waterbath\waterbath.ino

Right file: C:\xxd\myGit\waterbath\Sous_Viduino\Sous_Viduino.ino

(continued)

| | | | | | |
|-----|---|----|-----|--|--|
| 311 | » WN; if (adc_key_in < 650) return BUTTON_LE | | | | |
| 312 | » FT; if (adc_key_in < 850) return BUTTON_SH | | | | |
| 313 | » IFT; | | | | |
| 314 | return btnNONE; // when all others fai | | | | |
| 315 | » 1, return this... | | | | |
| 316 | } | | | | |
| 317 | | | | | |
| 318 | // ***** » ***** | = | 241 | // ***** » ***** | |
| 319 | // Initial State - press RIGHT to enter | | 242 | // Initial State - press RIGHT to enter | |
| 320 | » setpoint | | 243 | » setpoint | |
| 321 | // ***** | | 244 | // ***** | |
| 322 | » ***** | | 245 | » ***** | |
| 323 | void Off() | | 246 | void Off() | |
| 324 | { | | 247 | { | |
| 325 | myPID.SetMode(MANUAL); | | 248 | myPID.SetMode(MANUAL); | |
| 326 | //lcd.setBacklight(0); | <> | 249 | lcd.setBacklight(0); | |
| 327 | digitalWrite(RelayPin, LOW); // make | = | 250 | digitalWrite(RelayPin, LOW); // make | |
| 328 | » sure it is off | | 251 | » sure it is off | |
| 329 | lcd.print(F(" Water Bath")); | <> | 252 | lcd.print(F(" Adafruit")); | |
| 330 | lcd.setCursor(0, 1); | = | 253 | lcd.setCursor(0, 1); | |
| 331 | lcd.print(F(" off state")); | <> | 254 | lcd.print(F(" Sous Vide!")); | |
| 332 | uint8_t buttons = 0; | = | 255 | uint8_t buttons = 0; | |
| 333 | | <> | 256 | | |
| 334 | while(buttons != BUTTON_RIGHT) | | 257 | while(!(buttons & (BUTTON_RIGHT))) | |
| 335 | { | = | 258 | { | |
| 336 | buttons = ReadButtons(); | | 259 | buttons = ReadButtons(); | |
| 337 | } | | 260 | } | |
| 338 | | | 261 | | |
| 339 | Serial.println("Right button is press | +- | 262 | Serial.println("Right button is press | |
| 340 | » ed"); | | 263 | » ed"); | |
| 341 | printStats(); | | 264 | printStats(); | |
| 342 | | | 265 | | |
| 343 | | | 266 | | |
| 344 | // Prepare to transition to the RUN s | = | 267 | // Prepare to transition to the RUN s | |
| 345 | » tate | | 268 | » tate | |
| 346 | sensors.requestTemperatures(); // Sta | | 269 | sensors.requestTemperatures(); // Sta | |
| 347 | » rt an asynchronous temperature reading | | 270 | » rt an asynchronous temperature reading | |
| 348 | | | 271 | | |
| 349 | //turn the PID on | | 272 | //turn the PID on | |
| 350 | myPID.SetMode(AUTOMATIC); | | 273 | myPID.SetMode(AUTOMATIC); | |
| 351 | windowStartTime = millis(); | | 274 | windowStartTime = millis(); | |
| 352 | opState = RUN; // start control | | 275 | opState = RUN; // start control | |
| 353 | } | | 276 | } | |
| 354 | | | 277 | | |
| 355 | // ***** | | 278 | // ***** | |
| 356 | » ***** | | 279 | » ***** | |
| 357 | // Setpoint Entry State | | 280 | // Setpoint Entry State | |
| 358 | // UP/DOWN to change setpoint | | 281 | // UP/DOWN to change setpoint | |
| 359 | // RIGHT for tuning parameters | | 282 | // RIGHT for tuning parameters | |
| 360 | // LEFT for OFF | | 283 | // LEFT for OFF | |
| 361 | // SHIFT for 10x tuning | | 284 | // SHIFT for 10x tuning | |
| 362 | // ***** | | 285 | // ***** | |
| 363 | » ***** | | 286 | » ***** | |

Left file: C:\xxd\myGit\waterbath\waterbath\waterbath.ino

Right file: C:\xxd\myGit\waterbath\Sous_Viduino\Sous_Viduino.ino

(continued)

| | | | | |
|-----|---|----|-----|---|
| 355 | void Tune_Sp() | | 274 | void Tune_Sp() |
| 356 | { | | 275 | { |
| 357 | //lcd.setBacklight(TEAL); | <> | 276 | lcd.setBacklight(TEAL); |
| 358 | lcd.print(F("Set Temperature:")); | = | 277 | lcd.print(F("Set Temperature:")); |
| 359 | Serial.println("Entering Tune Set poi | +- | | |
| | >> nt"); | | | |
| 360 | uint8_t buttons = 0; | = | 278 | uint8_t buttons = 0; |
| 361 | | <> | | |
| 362 | while(true) | = | 279 | while(true) |
| 363 | { | | 280 | { |
| 364 | buttons = ReadButtons(); | | 281 | buttons = ReadButtons(); |
| 365 | | <> | 282 | |
| 366 | float increment = 0.1; | | 283 | float increment = 0.1; |
| 367 | /* | | | |
| 368 | Serial.print("buttons= "); | | | |
| 369 | Serial.println(buttons); | | | |
| 370 | */ | | | |
| 371 | if (buttons == BUTTON_SHIFT) | | 284 | if (buttons & BUTTON_SHIFT) |
| 372 | { | = | 285 | { |
| 373 | Serial.println("button shift is | +- | | |
| | >> pressed"); | | | |
| 374 | increment *= 10; | = | 286 | increment *= 10; |
| 375 | Serial.print("increment= "); | +- | | |
| 376 | Serial.println(increment); | | | |
| 377 | } | = | 287 | } |
| 378 | if (buttons == BUTTON_LEFT) | <> | 288 | if (buttons & BUTTON_LEFT) |
| 379 | { | = | 289 | { |
| 380 | opState = RUN; | | 290 | opState = RUN; |
| 381 | return; | | 291 | return; |
| 382 | } | | 292 | } |
| 383 | if (buttons == BUTTON_RIGHT) | <> | 293 | if (buttons & BUTTON_RIGHT) |
| 384 | { | = | 294 | { |
| 385 | opState = TUNE_P; | | 295 | opState = TUNE_P; |
| 386 | return; | | 296 | return; |
| 387 | } | | 297 | } |
| 388 | if (buttons == BUTTON_UP) | <> | 298 | if (buttons & BUTTON_UP) |
| 389 | { | = | 299 | { |
| 390 | Setpoint += increment; | | 300 | Setpoint += increment; |
| 391 | delay(200); | | 301 | delay(200); |
| 392 | } | | 302 | } |
| 393 | if (buttons == BUTTON_DOWN) | <> | 303 | if (buttons & BUTTON_DOWN) |
| 394 | { | = | 304 | { |
| 395 | Serial.println("reduce setpoint | +- | | |
| | >> "); | | | |
| 396 | Setpoint -= increment; | = | 305 | Setpoint -= increment; |
| 397 | Serial.print("increment= "); | +- | | |
| 398 | Serial.println(increment); | | | |
| 399 | delay(200); | = | 306 | delay(200); |
| 400 | } | | 307 | } |
| 401 | | | 308 | |
| 402 | if ((millis() - lastInput) > 3000) | | 309 | if ((millis() - lastInput) > 3000) |
| | >> // return to RUN after 3 seconds idl | | | >> // return to RUN after 3 seconds idl |
| | >> e | | | >> e |
| 403 | { | | 310 | { |
| 404 | opState = RUN; | | 311 | opState = RUN; |
| 405 | return; | | 312 | return; |
| 406 | } | | 313 | } |

Left file: C:\xxd\myGit\waterbath\waterbath\waterbath.ino

Right file: C:\xxd\myGit\waterbath\Sous_Viduino\Sous_Viduino.ino

(continued)

| | | | | |
|-----|--|----|-----|--|
| 407 | lcd.setCursor(0,1); | | 314 | lcd.setCursor(0,1); |
| 408 | lcd.print(Setpoint); | | 315 | lcd.print(Setpoint); |
| 409 | lcd.print(" "); | | 316 | lcd.print(" "); |
| 410 | DoControl(); | | 317 | DoControl(); |
| 411 | } | | 318 | } |
| 412 | } | | 319 | } |
| 413 | | | 320 | |
| 414 | // ***** | | 321 | // ***** |
| | » ***** | | | » ***** |
| 415 | // Proportional Tuning State | | 322 | // Proportional Tuning State |
| 416 | // UP/DOWN to change Kp | | 323 | // UP/DOWN to change Kp |
| 417 | // RIGHT for Ki | | 324 | // RIGHT for Ki |
| 418 | // LEFT for setpoint | | 325 | // LEFT for setpoint |
| 419 | // SHIFT for 10x tuning | | 326 | // SHIFT for 10x tuning |
| 420 | // ***** | | 327 | // ***** |
| | » ***** | | | » ***** |
| 421 | void TuneP() | | 328 | void TuneP() |
| 422 | { | | 329 | { |
| 423 | //lcd.setBacklight(TEAL); | <> | 330 | lcd.setBacklight(TEAL); |
| 424 | lcd.print(F("Set Kp")); | = | 331 | lcd.print(F("Set Kp")); |
| 425 | | | 332 | |
| 426 | uint8_t buttons = 0; | | 333 | uint8_t buttons = 0; |
| 427 | while(true) | | 334 | while(true) |
| 428 | { | | 335 | { |
| 429 | buttons = ReadButtons(); | | 336 | buttons = ReadButtons(); |
| 430 | | | 337 | |
| 431 | float increment = 1.0; | | 338 | float increment = 1.0; |
| 432 | if (buttons == BUTTON_SHIFT) | <> | 339 | if (buttons & BUTTON_SHIFT) |
| 433 | { | = | 340 | { |
| 434 | increment *= 10; | | 341 | increment *= 10; |
| 435 | } | | 342 | } |
| 436 | if (buttons == BUTTON_LEFT) | <> | 343 | if (buttons & BUTTON_LEFT) |
| 437 | { | = | 344 | { |
| 438 | opState = SETP; | | 345 | opState = SETP; |
| 439 | return; | | 346 | return; |
| 440 | } | | 347 | } |
| 441 | if (buttons == BUTTON_RIGHT) | <> | 348 | if (buttons & BUTTON_RIGHT) |
| 442 | { | = | 349 | { |
| 443 | opState = TUNE_I; | | 350 | opState = TUNE_I; |
| 444 | return; | | 351 | return; |
| 445 | } | | 352 | } |
| 446 | if (buttons == BUTTON_UP) | <> | 353 | if (buttons & BUTTON_UP) |
| 447 | { | = | 354 | { |
| 448 | Kp += increment; | | 355 | Kp += increment; |
| 449 | delay(200); | | 356 | delay(200); |
| 450 | } | | 357 | } |
| 451 | if (buttons == BUTTON_DOWN) | <> | 358 | if (buttons & BUTTON_DOWN) |
| 452 | { | = | 359 | { |
| 453 | Kp -= increment; | | 360 | Kp -= increment; |
| 454 | delay(200); | | 361 | delay(200); |
| 455 | } | | 362 | } |
| 456 | if ((millis() - lastInput) > 3000) | | 363 | if ((millis() - lastInput) > 3000) |
| | » // return to RUN after 3 seconds idl | | | » // return to RUN after 3 seconds idl |
| | » e | | | » e |
| 457 | { | | 364 | { |
| 458 | opState = RUN; | | 365 | opState = RUN; |
| 459 | return; | | 366 | return; |

Left file: C:\xxd\myGit\waterbath\waterbath\waterbath.ino

Right file: C:\xxd\myGit\waterbath\Sous_Viduino\Sous_Viduino.ino

(continued)

| | | | | |
|-----|--|----|-----|--|
| 460 | } | | 367 | } |
| 461 | lcd.setCursor(0,1); | | 368 | lcd.setCursor(0,1); |
| 462 | lcd.print(Kp); | | 369 | lcd.print(Kp); |
| 463 | lcd.print(" "); | | 370 | lcd.print(" "); |
| 464 | DoControl(); | | 371 | DoControl(); |
| 465 | } | | 372 | } |
| 466 | } | | 373 | } |
| 467 | | | 374 | |
| 468 | // ***** | | 375 | // ***** |
| | » ***** | | | » ***** |
| 469 | // Integral Tuning State | | 376 | // Integral Tuning State |
| 470 | // UP/DOWN to change Ki | | 377 | // UP/DOWN to change Ki |
| 471 | // RIGHT for Kd | | 378 | // RIGHT for Kd |
| 472 | // LEFT for Kp | | 379 | // LEFT for Kp |
| 473 | // SHIFT for 10x tuning | | 380 | // SHIFT for 10x tuning |
| 474 | // ***** | | 381 | // ***** |
| | » ***** | | | » ***** |
| 475 | void TuneI() | | 382 | void TuneI() |
| 476 | { | | 383 | { |
| 477 | //lcd.setBacklight(TEAL); | <> | 384 | lcd.setBacklight(TEAL); |
| 478 | lcd.print(F("Set Ki")); | = | 385 | lcd.print(F("Set Ki")); |
| 479 | | | 386 | |
| 480 | uint8_t buttons = 0; | | 387 | uint8_t buttons = 0; |
| 481 | while(true) | | 388 | while(true) |
| 482 | { | | 389 | { |
| 483 | buttons = ReadButtons(); | | 390 | buttons = ReadButtons(); |
| 484 | | | 391 | |
| 485 | float increment = 0.01; | | 392 | float increment = 0.01; |
| 486 | if (buttons == BUTTON_SHIFT) | <> | 393 | if (buttons & BUTTON_SHIFT) |
| 487 | { | = | 394 | { |
| 488 | increment *= 10; | | 395 | increment *= 10; |
| 489 | } | | 396 | } |
| 490 | if (buttons == BUTTON_LEFT) | <> | 397 | if (buttons & BUTTON_LEFT) |
| 491 | { | = | 398 | { |
| 492 | opState = TUNE_P; | | 399 | opState = TUNE_P; |
| 493 | return; | | 400 | return; |
| 494 | } | | 401 | } |
| 495 | if (buttons == BUTTON_RIGHT) | <> | 402 | if (buttons & BUTTON_RIGHT) |
| 496 | { | = | 403 | { |
| 497 | opState = TUNE_D; | | 404 | opState = TUNE_D; |
| 498 | return; | | 405 | return; |
| 499 | } | | 406 | } |
| 500 | if (buttons == BUTTON_UP) | <> | 407 | if (buttons & BUTTON_UP) |
| 501 | { | = | 408 | { |
| 502 | Ki += increment; | | 409 | Ki += increment; |
| 503 | delay(200); | | 410 | delay(200); |
| 504 | } | | 411 | } |
| 505 | if (buttons == BUTTON_DOWN) | <> | 412 | if (buttons & BUTTON_DOWN) |
| 506 | { | = | 413 | { |
| 507 | Ki -= increment; | | 414 | Ki -= increment; |
| 508 | delay(200); | | 415 | delay(200); |
| 509 | } | | 416 | } |
| 510 | if ((millis() - lastInput) > 3000) | | 417 | if ((millis() - lastInput) > 3000) |
| | » // return to RUN after 3 seconds idl | | | » // return to RUN after 3 seconds idl |
| | » e | | | » e |
| 511 | { | | 418 | { |
| 512 | opState = RUN; | | 419 | opState = RUN; |

Left file: C:\xxd\myGit\waterbath\waterbath\waterbath.ino

Right file: C:\xxd\myGit\waterbath\Sous_Viduino\Sous_Viduino.ino

(continued)

| | | | | | |
|-----|--|----|-----|--|--|
| 513 | return; | | 420 | return; | |
| 514 | } | | 421 | } | |
| 515 | lcd.setCursor(0,1); | | 422 | lcd.setCursor(0,1); | |
| 516 | lcd.print(Ki); | | 423 | lcd.print(Ki); | |
| 517 | lcd.print(" "); | | 424 | lcd.print(" "); | |
| 518 | DoControl(); | | 425 | DoControl(); | |
| 519 | } | | 426 | } | |
| 520 | } | | 427 | } | |
| 521 | | | 428 | | |
| 522 | // ***** | | 429 | // ***** | |
| | » ***** | | | » ***** | |
| 523 | // Derivative Tuning State | | 430 | // Derivative Tuning State | |
| 524 | // UP/DOWN to change Kd | | 431 | // UP/DOWN to change Kd | |
| 525 | // RIGHT for setpoint | | 432 | // RIGHT for setpoint | |
| 526 | // LEFT for Ki | | 433 | // LEFT for Ki | |
| 527 | // SHIFT for 10x tuning | | 434 | // SHIFT for 10x tuning | |
| 528 | // ***** | | 435 | // ***** | |
| | » ***** | | | » ***** | |
| 529 | void TuneD() | | 436 | void TuneD() | |
| 530 | { | | 437 | { | |
| 531 | //lcd.setBacklight(TEAL); | <> | 438 | lcd.setBacklight(TEAL); | |
| 532 | lcd.print(F("Set Kd")); | = | 439 | lcd.print(F("Set Kd")); | |
| 533 | | | 440 | | |
| 534 | uint8_t buttons = 0; | | 441 | uint8_t buttons = 0; | |
| 535 | while(true) | | 442 | while(true) | |
| 536 | { | | 443 | { | |
| 537 | buttons = ReadButtons(); | | 444 | buttons = ReadButtons(); | |
| 538 | float increment = 0.01; | | 445 | float increment = 0.01; | |
| 539 | if (buttons == BUTTON_SHIFT) | <> | 446 | if (buttons & BUTTON_SHIFT) | |
| 540 | { | = | 447 | { | |
| 541 | increment *= 10; | | 448 | increment *= 10; | |
| 542 | } | | 449 | } | |
| 543 | if (buttons == BUTTON_LEFT) | <> | 450 | if (buttons & BUTTON_LEFT) | |
| 544 | { | = | 451 | { | |
| 545 | opState = TUNE_I; | | 452 | opState = TUNE_I; | |
| 546 | return; | | 453 | return; | |
| 547 | } | | 454 | } | |
| 548 | if (buttons == BUTTON_RIGHT) | <> | 455 | if (buttons & BUTTON_RIGHT) | |
| 549 | { | = | 456 | { | |
| 550 | opState = RUN; | | 457 | opState = RUN; | |
| 551 | return; | | 458 | return; | |
| 552 | } | | 459 | } | |
| 553 | if (buttons == BUTTON_UP) | <> | 460 | if (buttons & BUTTON_UP) | |
| 554 | { | = | 461 | { | |
| 555 | Kd += increment; | | 462 | Kd += increment; | |
| 556 | delay(200); | | 463 | delay(200); | |
| 557 | } | | 464 | } | |
| 558 | if (buttons == BUTTON_DOWN) | <> | 465 | if (buttons & BUTTON_DOWN) | |
| 559 | { | = | 466 | { | |
| 560 | Kd -= increment; | | 467 | Kd -= increment; | |
| 561 | delay(200); | | 468 | delay(200); | |
| 562 | } | | 469 | } | |
| 563 | if ((millis() - lastInput) > 3000) | | 470 | if ((millis() - lastInput) > 3000) | |
| | » // return to RUN after 3 seconds idl | | | » // return to RUN after 3 seconds idl | |
| | » e | | | » e | |
| 564 | { | | 471 | { | |
| 565 | opState = RUN; | | 472 | opState = RUN; | |

Left file: C:\xxd\myGit\waterbath\waterbath\waterbath.ino

Right file: C:\xxd\myGit\waterbath\Sous_Viduino\Sous_Viduino.ino

(continued)

| | | | | | |
|-----|---------------------------------------|----|-----|---------------------------------------|--|
| 566 | return; | | 473 | return; | |
| 567 | } | | 474 | } | |
| 568 | lcd.setCursor(0,1); | | 475 | lcd.setCursor(0,1); | |
| 569 | lcd.print(Kd); | | 476 | lcd.print(Kd); | |
| 570 | lcd.print(" "); | | 477 | lcd.print(" "); | |
| 571 | DoControl(); | | 478 | DoControl(); | |
| 572 | } | | 479 | } | |
| 573 | } | | 480 | } | |
| 574 | | | 481 | | |
| 575 | // ***** | | 482 | // ***** | |
| | » ***** | | | » ***** | |
| 576 | // PID Control State | | 483 | // PID Control State | |
| 577 | // SHIFT and RIGHT for autotune | | 484 | // SHIFT and RIGHT for autotune | |
| 578 | // RIGHT - Setpoint | | 485 | // RIGHT - Setpoint | |
| 579 | // LEFT - OFF | | 486 | // LEFT - OFF | |
| 580 | // ***** | | 487 | // ***** | |
| | » ***** | | | » ***** | |
| 581 | void Run() | | 488 | void Run() | |
| 582 | { | | 489 | { | |
| 583 | Serial.println("Entering Run state"); | +- | | | |
| 584 | // set up the LCD's number of rows an | = | 490 | // set up the LCD's number of rows an | |
| | » d columns: | | | » d columns: | |
| 585 | lcd.print(F("Sp: ")); | | 491 | lcd.print(F("Sp: ")); | |
| 586 | lcd.print(Setpoint); | | 492 | lcd.print(Setpoint); | |
| 587 | lcd.write(1); | | 493 | lcd.write(1); | |
| 588 | lcd.print(F("C : ")); | | 494 | lcd.print(F("C : ")); | |
| 589 | | | 495 | | |
| 590 | SaveParameters(); | | 496 | SaveParameters(); | |
| 591 | myPID.SetTunings(Kp,Ki,Kd); | | 497 | myPID.SetTunings(Kp,Ki,Kd); | |
| 592 | | | 498 | | |
| 593 | uint8_t buttons = 0; | | 499 | uint8_t buttons = 0; | |
| 594 | while(true) | | 500 | while(true) | |
| 595 | { | | 501 | { | |
| 596 | setBacklight(); // set backlight | | 502 | setBacklight(); // set backlight | |
| | » based on state | | | » based on state | |
| 597 | | | 503 | | |
| 598 | buttons = ReadButtons(); | | 504 | buttons = ReadButtons(); | |
| 599 | if ((buttons == BUTTON_SHIFT) | <> | 505 | if ((buttons & BUTTON_SHIFT) | |
| 600 | //&& (buttons == BUTTON_RIGHT) | | 506 | && (buttons & BUTTON_RIGHT) | |
| 601 | && (abs(Input - Setpoint) < 0.5 | = | 507 | && (abs(Input - Setpoint) < 0.5 | |
| | »)) // Should be at steady-state | | | »)) // Should be at steady-state | |
| 602 | { | | 508 | { | |
| 603 | Serial.println("Entering auto t | +- | | | |
| | » une"); | | | | |
| 604 | StartAutoTune(); | = | 509 | StartAutoTune(); | |
| 605 | } | | 510 | } | |
| 606 | else if (buttons == BUTTON_RIGHT) | <> | 511 | else if (buttons & BUTTON_RIGHT) | |
| 607 | { | = | 512 | { | |
| 608 | opState = SETP; | | 513 | opState = SETP; | |
| 609 | return; | | 514 | return; | |
| 610 | } | | 515 | } | |
| 611 | else if (buttons == BUTTON_LEFT) | <> | 516 | else if (buttons & BUTTON_LEFT) | |
| 612 | { | = | 517 | { | |
| 613 | opState = OFF; | | 518 | opState = OFF; | |
| 614 | return; | | 519 | return; | |
| 615 | } | | 520 | } | |
| 616 | | | 521 | | |

Left file: C:\xxd\myGit\waterbath\waterbath\waterbath.ino

Right file: C:\xxd\myGit\waterbath\Sous_Viduino\Sous_Viduino.ino

(continued)

| | | | | |
|------------------|---------------------------------------|----|------------------|---------------------------------------|
| 617 | DoControl(); | | 522 | DoControl(); |
| 618 | | | 523 | |
| 619 | lcd.setCursor(0,1); | | 524 | lcd.setCursor(0,1); |
| 620 | lcd.print(Input); | | 525 | lcd.print(Input); |
| 621 | lcd.write(1); | | 526 | lcd.write(1); |
| 622 | lcd.print(F("C : ")); | | 527 | lcd.print(F("C : ")); |
| 623 | | | 528 | |
| 624 | float pct = map(Output, 0, Windows | | 529 | float pct = map(Output, 0, Windows |
| » ize, 0, 1000); | | | » ize, 0, 1000); | |
| 625 | lcd.setCursor(10,1); | | 530 | lcd.setCursor(10,1); |
| 626 | //lcd.print(F(" ")); | <> | 531 | lcd.print(F(" ")); |
| 627 | lcd.setCursor(9,1); | | 532 | lcd.setCursor(10,1); |
| 628 | lcd.print(pct/10); | = | 533 | lcd.print(pct/10); |
| 629 | //lcd.print(Output); | | 534 | //lcd.print(Output); |
| 630 | lcd.print("%"); | | 535 | lcd.print("%"); |
| 631 | | | 536 | |
| 632 | lcd.setCursor(15,0); | | 537 | lcd.setCursor(15,0); |
| 633 | if (tuning) | | 538 | if (tuning) |
| 634 | { | | 539 | { |
| 635 | lcd.print("T"); | | 540 | lcd.print("T"); |
| 636 | } | | 541 | } |
| 637 | else | | 542 | else |
| 638 | { | | 543 | { |
| 639 | lcd.print(" "); | | 544 | lcd.print(" "); |
| 640 | } | | 545 | } |
| 641 | | | 546 | |
| 642 | // periodically log to serial port | | 547 | // periodically log to serial port |
| » in csv format | | | » in csv format | |
| 643 | if (millis() - lastLogTime > logIn | | 548 | if (millis() - lastLogTime > logIn |
| » terval) | | | » terval) | |
| 644 | { | | 549 | { |
| 645 | lastLogTime = millis(); | +- | | |
| 646 | | | | |
| 647 | printStats(); | | | |
| 648 | /** | | | |
| 649 | Serial.print(Input); | = | 550 | Serial.print(Input); |
| 650 | Serial.print(","); | | 551 | Serial.print(","); |
| 651 | Serial.println(Output); | | 552 | Serial.println(Output); |
| 652 | **/ | +- | | |
| 653 | } | = | 553 | } |
| 654 | | | 554 | |
| 655 | delay(100); | | 555 | delay(100); |
| 656 | } | | 556 | } |
| 657 | } | | 557 | } |
| 658 | | | 558 | |
| 659 | // ***** | | 559 | // ***** |
| » ***** | | | » ***** | |
| 660 | // Execute the control loop | | 560 | // Execute the control loop |
| 661 | // ***** | | 561 | // ***** |
| » ***** | | | » ***** | |
| 662 | void DoControl() | | 562 | void DoControl() |
| 663 | { | | 563 | { |
| 664 | // Read the input: | | 564 | // Read the input: |
| 665 | if (sensors.isConversionAvailable(0)) | | 565 | if (sensors.isConversionAvailable(0)) |
| 666 | { | | 566 | { |
| 667 | Input = sensors.getTempC(tempSensor) | | 567 | Input = sensors.getTempC(tempSensor) |
| » ; | | | » ; | |

Left file: C:\xxd\myGit\waterbath\waterbath\waterbath.ino

Right file: C:\xxd\myGit\waterbath\Sous_Viduino\Sous_Viduino.ino

(continued)

| | | | | | |
|-----|--|----|-----|--|--|
| 668 | sensors.requestTemperatures(); // pr » ime the pump for the next one - but do » n't wait | | 568 | sensors.requestTemperatures(); // pr » ime the pump for the next one - but do » n't wait | |
| 669 | } | | 569 | } | |
| 670 | | | 570 | | |
| 671 | if (tuning) // run the auto-tuner | | 571 | if (tuning) // run the auto-tuner | |
| 672 | { | | 572 | { | |
| 673 | if (aTune.Runtime()) // returns 'tr » ue' when done | | 573 | if (aTune.Runtime()) // returns 'tr » ue' when done | |
| 674 | { | | 574 | { | |
| 675 | Serial.println("finish auto tune » ."); | +- | | | |
| 676 | FinishAutoTune(); | = | 575 | FinishAutoTune(); | |
| 677 | } | | 576 | } | |
| 678 | } | | 577 | } | |
| 679 | else // Execute control algorithm | | 578 | else // Execute control algorithm | |
| 680 | { | | 579 | { | |
| 681 | myPID.Compute(); | | 580 | myPID.Compute(); | |
| 682 | } | | 581 | } | |
| 683 | | | 582 | | |
| 684 | // Time Proportional relay state is up » dated regularly via timer interrupt. | | 583 | // Time Proportional relay state is up » dated regularly via timer interrupt. | |
| 685 | onTime = Output; | | 584 | onTime = Output; | |
| 686 | } | | 585 | } | |
| 687 | | | 586 | | |
| 688 | // ***** | | 587 | // ***** | |
| 689 | // Called by ISR every 15ms to drive the » output | | 588 | // Called by ISR every 15ms to drive the » output | |
| 690 | // ***** | | 589 | // ***** | |
| 691 | void DriveOutput() | | 590 | void DriveOutput() | |
| 692 | { | | 591 | { | |
| 693 | long now = millis(); | | 592 | long now = millis(); | |
| 694 | // Set the output | | 593 | // Set the output | |
| 695 | // "on time" is proportional to the PI » D output | | 594 | // "on time" is proportional to the PI » D output | |
| 696 | if(now - windowStartTime>WindowSize) | | 595 | if(now - windowStartTime>WindowSize) | |
| 697 | { //time to shift the Relay Window | | 596 | { //time to shift the Relay Window | |
| 698 | windowStartTime += WindowSize; | | 597 | windowStartTime += WindowSize; | |
| 699 | } | | 598 | } | |
| 700 | if((onTime > 100) && (onTime > (now - » windowStartTime))) | | 599 | if((onTime > 100) && (onTime > (now - » windowStartTime))) | |
| 701 | { | | 600 | { | |
| 702 | //Serial.println("Turn on the heate » r"); | +- | | | |
| 703 | digitalWrite(RelayPin,HIGH); | = | 601 | digitalWrite(RelayPin,HIGH); | |
| 704 | } | | 602 | } | |
| 705 | else | | 603 | else | |
| 706 | { | | 604 | { | |
| 707 | //Serial.println("Shut off the heat » er"); | +- | | | |
| 708 | digitalWrite(RelayPin,LOW); | = | 605 | digitalWrite(RelayPin,LOW); | |
| 709 | } | | 606 | } | |
| 710 | } | | 607 | } | |
| 711 | | | 608 | | |
| 712 | // ***** | | 609 | // ***** | |

Left file: C:\xxd\myGit\waterbath\waterbath\waterbath.ino

Right file: C:\xxd\myGit\waterbath\Sous_Viduino\Sous_Viduino.ino

(continued)

| | | | | | |
|-----|---|----|-----|--|--|
| 713 | » ***** // Set Backlight based on the state of c » ontrol | | 610 | » ***** // Set Backlight based on the state of c » ontrol | |
| 714 | // ***** » ***** | | 611 | // ***** » ***** | |
| 715 | void setBacklight() | | 612 | void setBacklight() | |
| 716 | { | | 613 | { | |
| 717 | if (tuning) | | 614 | if (tuning) | |
| 718 | { | | 615 | { | |
| 719 | //lcd.setBacklight(VIOLET); // Tun » ing Mode | <> | 616 | lcd.setBacklight(VIOLET); // Tunin » g Mode | |
| 720 | } | = | 617 | } | |
| 721 | else if (abs(Input - Setpoint) > 1.0) | | 618 | else if (abs(Input - Setpoint) > 1.0) | |
| 722 | » | | 619 | » | |
| 723 | { | | 620 | { | |
| 724 | //lcd.setBacklight(RED); // High » Alarm - off by more than 1 degree | <> | 621 | lcd.setBacklight(RED); // High Al » arm - off by more than 1 degree | |
| 725 | } | = | 622 | } | |
| 726 | else if (abs(Input - Setpoint) > 0.2) | | 623 | else if (abs(Input - Setpoint) > 0.2) | |
| 727 | » | | 624 | » | |
| 728 | { | | 625 | { | |
| 729 | //lcd.setBacklight(YELLOW); // Lo » w Alarm - off by more than 0.2 degrees | <> | 626 | lcd.setBacklight(YELLOW); // Low » Alarm - off by more than 0.2 degrees | |
| 730 | } | = | 627 | } | |
| 731 | //lcd.setBacklight(WHITE); // We' » re on target! | <> | 628 | lcd.setBacklight(WHITE); // We're » on target! | |
| 732 | } | = | 629 | } | |
| 733 | } | | 630 | } | |
| 734 | | | 631 | | |
| 735 | // ***** » ***** | | 632 | // ***** » ***** | |
| 736 | // Start the Auto-Tuning cycle | | 633 | // Start the Auto-Tuning cycle | |
| 737 | // ***** » ***** | | 634 | // ***** » ***** | |
| 738 | | | 635 | | |
| 739 | void StartAutoTune() | | 636 | void StartAutoTune() | |
| 740 | { | | 637 | { | |
| 741 | // REmember the mode we were in | | 638 | // REmember the mode we were in | |
| 742 | ATuneModeRemember = myPID.GetMode(); | | 639 | ATuneModeRemember = myPID.GetMode(); | |
| 743 | | | 640 | | |
| 744 | // set up the auto-tune parameters | | 641 | // set up the auto-tune parameters | |
| 745 | aTune.SetNoiseBand(aTuneNoise); | | 642 | aTune.SetNoiseBand(aTuneNoise); | |
| 746 | aTune.SetOutputStep(aTuneStep); | | 643 | aTune.SetOutputStep(aTuneStep); | |
| 747 | aTune.SetLookbackSec((int)aTuneLookBa » ck); | | 644 | aTune.SetLookbackSec((int)aTuneLookBa » ck); | |
| 748 | tuning = true; | | 645 | tuning = true; | |
| 749 | } | | 646 | } | |
| 750 | | | 647 | | |
| 751 | // ***** » ***** | | 648 | // ***** » ***** | |
| 752 | // Return to normal control | | 649 | // Return to normal control | |
| 753 | // ***** » ***** | | 650 | // ***** » ***** | |
| 754 | void FinishAutoTune() | | 651 | void FinishAutoTune() | |
| 755 | { | | 652 | { | |

Left file: C:\xxd\myGit\waterbath\waterbath\waterbath.ino

Right file: C:\xxd\myGit\waterbath\Sous_Viduino\Sous_Viduino.ino

(continued)

| | | | | |
|-----|--|----|-----|--|
| 756 | tuning = false; | | 653 | tuning = false; |
| 757 | | | 654 | |
| 758 | // Extract the auto-tune calculated p | | 655 | // Extract the auto-tune calculated p |
| » | arameters | | » | arameters |
| 759 | Kp = aTune.GetKp(); | | 656 | Kp = aTune.GetKp(); |
| 760 | Ki = aTune.GetKi(); | | 657 | Ki = aTune.GetKi(); |
| 761 | Kd = aTune.GetKd(); | | 658 | Kd = aTune.GetKd(); |
| 762 | | | 659 | |
| 763 | // Re-tune the PID and revert to norm | | 660 | // Re-tune the PID and revert to norm |
| » | al control mode | | » | al control mode |
| 764 | myPID.SetTunings(Kp,Ki,Kd); | | 661 | myPID.SetTunings(Kp,Ki,Kd); |
| 765 | myPID.SetMode(ATuneModeRemember); | | 662 | myPID.SetMode(ATuneModeRemember); |
| 766 | | | 663 | |
| 767 | // Persist any changed parameters to | | 664 | // Persist any changed parameters to |
| » | EEPROM | | » | EEPROM |
| 768 | SaveParameters(); | | 665 | SaveParameters(); |
| 769 | } | | 666 | } |
| 770 | | | 667 | |
| 771 | // ***** | | 668 | // ***** |
| » | ***** | | » | ***** |
| 772 | // Check buttons and time-stamp the last | | 669 | // Check buttons and time-stamp the last |
| » | press | | » | press |
| 773 | // ***** | | 670 | // ***** |
| » | ***** | | » | ***** |
| 774 | uint8_t ReadButtons() | | 671 | uint8_t ReadButtons() |
| 775 | { | | 672 | { |
| 776 | //uint8_t buttons = lcd.readButtons(); | <> | 673 | uint8_t buttons = lcd.readButtons(); |
| 777 | uint8_t buttons = read_LCD_buttons(); | | | |
| 778 | if (buttons != 0) | = | 674 | if (buttons != 0) |
| 779 | { | | 675 | { |
| 780 | lastInput = millis(); | | 676 | lastInput = millis(); |
| 781 | } | | 677 | } |
| 782 | return buttons; | | 678 | return buttons; |
| 783 | } | | 679 | } |
| 784 | | | 680 | |
| 785 | // ***** | | 681 | // ***** |
| » | ***** | | » | ***** |
| 786 | // Save any parameter changes to EEPROM | | 682 | // Save any parameter changes to EEPROM |
| 787 | // ***** | | 683 | // ***** |
| » | ***** | | » | ***** |
| 788 | void SaveParameters() | | 684 | void SaveParameters() |
| 789 | { | | 685 | { |
| 790 | if (Setpoint != EEPROM_readDouble(SpA | | 686 | if (Setpoint != EEPROM_readDouble(SpA |
| » | ddress)) | | » | ddress)) |
| 791 | { | | 687 | { |
| 792 | EEPROM_writeDouble(SpAddress, Setp | | 688 | EEPROM_writeDouble(SpAddress, Setp |
| » | oint); | | » | oint); |
| 793 | } | | 689 | } |
| 794 | if (Kp != EEPROM_readDouble(KpAddress | | 690 | if (Kp != EEPROM_readDouble(KpAddress |
| » |)) | | » |)) |
| 795 | { | | 691 | { |
| 796 | EEPROM_writeDouble(KpAddress, Kp); | | 692 | EEPROM_writeDouble(KpAddress, Kp); |
| 797 | } | | 693 | } |
| 798 | if (Ki != EEPROM_readDouble(KiAddress | | 694 | if (Ki != EEPROM_readDouble(KiAddress |
| » |)) | | » |)) |
| 799 | { | | 695 | { |
| 800 | EEPROM_writeDouble(KiAddress, Ki); | | 696 | EEPROM_writeDouble(KiAddress, Ki); |

Left file: C:\xxd\myGit\waterbath\waterbath\waterbath.ino

Right file: C:\xxd\myGit\waterbath\Sous_Viduino\Sous_Viduino.ino

(continued)

| | | | | |
|-----|--|----|-----|--|
| 801 | } | | 697 | } |
| 802 | if (Kd != EEPROM_readDouble(KdAddress | | 698 | if (Kd != EEPROM_readDouble(KdAddress |
| | »)) | | | »)) |
| 803 | { | | 699 | { |
| 804 | EEPROM_writeDouble(KdAddress, Kd); | | 700 | EEPROM_writeDouble(KdAddress, Kd); |
| 805 | } | | 701 | } |
| 806 | } | | 702 | } |
| 807 | | | 703 | |
| 808 | // ***** | | 704 | // ***** |
| | » ***** | | | » ***** |
| 809 | // Load parameters from EEPROM | | 705 | // Load parameters from EEPROM |
| 810 | // ***** | | 706 | // ***** |
| | » ***** | | | » ***** |
| 811 | void LoadParameters() | | 707 | void LoadParameters() |
| 812 | { | | 708 | { |
| 813 | // Load from EEPROM | | 709 | // Load from EEPROM |
| 814 | Setpoint = EEPROM_readDouble(SpAddress | | 710 | Setpoint = EEPROM_readDouble(SpAddress |
| | » s); | | | » s); |
| 815 | Kp = EEPROM_readDouble(KpAddress); | | 711 | Kp = EEPROM_readDouble(KpAddress); |
| 816 | Ki = EEPROM_readDouble(KiAddress); | | 712 | Ki = EEPROM_readDouble(KiAddress); |
| 817 | Kd = EEPROM_readDouble(KdAddress); | | 713 | Kd = EEPROM_readDouble(KdAddress); |
| 818 | | | 714 | |
| 819 | // Use defaults if EEPROM values are | | 715 | // Use defaults if EEPROM values are |
| | » invalid | | | » invalid |
| 820 | if (isnan(Setpoint) Setpoint > 40 | <> | 716 | if (isnan(Setpoint)) |
| | » Setpoint < 0) | | | |
| 821 | { | = | 717 | { |
| 822 | Setpoint = 20; | <> | 718 | Setpoint = 60; |
| 823 | } | = | 719 | } |
| 824 | if (isnan(Kp)) | | 720 | if (isnan(Kp)) |
| 825 | { | | 721 | { |
| 826 | Kp = 850; | | 722 | Kp = 850; |
| 827 | } | | 723 | } |
| 828 | if (isnan(Ki)) | | 724 | if (isnan(Ki)) |
| 829 | { | | 725 | { |
| 830 | Ki = 0.5; | | 726 | Ki = 0.5; |
| 831 | } | | 727 | } |
| 832 | if (isnan(Kd)) | | 728 | if (isnan(Kd)) |
| 833 | { | | 729 | { |
| 834 | Kd = 0.1; | | 730 | Kd = 0.1; |
| 835 | } | | 731 | } |
| 836 | } | | 732 | } |
| 837 | | | 733 | |
| 838 | | | 734 | |
| 839 | // ***** | | 735 | // ***** |
| | » ***** | | | » ***** |
| 840 | // Write floating point values to EEPROM | | 736 | // Write floating point values to EEPROM |
| 841 | // ***** | | 737 | // ***** |
| | » ***** | | | » ***** |
| 842 | void EEPROM_writeDouble(int address, dou | | 738 | void EEPROM_writeDouble(int address, dou |
| | » ble value) | | | » ble value) |
| 843 | { | | 739 | { |
| 844 | byte* p = (byte*)(void*)&value; | | 740 | byte* p = (byte*)(void*)&value; |
| 845 | for (int i = 0; i < sizeof(value); i+ | | 741 | for (int i = 0; i < sizeof(value); i+ |
| | » +) | | | » +) |
| 846 | { | | 742 | { |
| 847 | EEPROM.write(address++, *p++); | | 743 | EEPROM.write(address++, *p++); |

Left file: C:\xxd\myGit\waterbath\waterbath\waterbath.ino

Right file: C:\xxd\myGit\waterbath\Sous_Viduino\Sous_Viduino.ino

(continued)

| | | | | |
|-----|---|----|-----|---|
| 848 | } | | 744 | } |
| 849 | } | | 745 | } |
| 850 | | | 746 | |
| 851 | // ***** | | 747 | // ***** |
| | » ***** | | | » ***** |
| 852 | // Read floating point values from EEPROM | | 748 | // Read floating point values from EEPROM |
| | » M | | | » M |
| 853 | // ***** | | 749 | // ***** |
| | » ***** | | | » ***** |
| 854 | double EEPROM_readDouble(int address) | | 750 | double EEPROM_readDouble(int address) |
| 855 | { | | 751 | { |
| 856 | double value = 0.0; | | 752 | double value = 0.0; |
| 857 | byte* p = (byte*)(void*)&value; | | 753 | byte* p = (byte*)(void*)&value; |
| 858 | for (int i = 0; i < sizeof(value); i+ | | 754 | for (int i = 0; i < sizeof(value); i+ |
| | » +) | | | » +) |
| 859 | { | | 755 | { |
| 860 | *p++ = EEPROM.read(address++); | | 756 | *p++ = EEPROM.read(address++); |
| 861 | } | | 757 | } |
| 862 | return value; | | 758 | return value; |
| 863 | } | | 759 | } |
| 864 | | <> | | |
| 865 | void LcdClearLine(int r) | | | |
| 866 | { | | | |
| 867 | lcd.setCursor(0,r); | | | |
| 868 | lcd.print(" "); | | | |
| 869 | lcd.setCursor(0,r); | | | |
| 870 | } | | | |