

```

8558 26 a000400 myval ds.b 1
8559 27 a000401 Pval ds.b 1
8560 28 a000402 tens ds.b 1
8561 29 a000403 ones ds.b 1
8562 30
8563 31 0000 C350 const1 equ $C350 ;TCNT = ((intTime/(1/BusCLK)) / Pre) = ((100ms/(1/4Mhz)) / 8) = 50,000 = $C350
8564 32
8565 33 0000 0020 Spoint equ 32
8566 34 0000 0014 Gain equ $14
8567 35 0000 0010 Adjust equ $10 ;Offset
8568 36
8569 37 0000 0070 PRE EQU $70 ;PWM Prescaler
8570 38 0000 00FA PERIOD EQU $FA ;PWM Period
8571 39 0000 0000 PWMCLK EQU $00 ;0 when scaler is not used, $08 when scaler is used
8572 40
8573 41 0000 0064 BILL EQU 100 ;Used for calculating around decimals
8574 42
8575 43 ;code section
8576 44 ORG ROMStart
8577 45
8578 46 ;*****
8579 47 ;* *
8580 48 ;* *
8581 49 ;* ECNS-414 *
8582 50 ;* 29 Oct 2020 *
8583 51 ;* Lab 7 *
8584 52 ;* *
8585 53 ;* The purpose of this program is to construct *
8586 54 ;* and demonstrate a closed loop control system *
8587 55 ;* using pulse width modulation. *
8588 56 ;* *
8589 57 ;* Created by: Brandon Empie *
8590 58 ;* *
8591 59 ;*****
8592 60
8593 61
8594 62 ;*****
8595 63 ;* Initilizations *
8596 64 ;*****
8597 65 Entry:
8598 66 _Startup:
8599 67 a004000 CF20 00 LDS #RAMEnd+1 ;initialize the stack pointer
8600 68
8601 69 a004003 8600 LDAA #PWMCLK ;turns on scaler B for channel 3
8602 70 a004005 5AA2 STAA PWMCLK
8603 71
8604 72 a004007 8670 LDAA #PRE ;Set PWM PRE
8605 73 a004009 5AA3 STAA PWMPRCLK
8606 74
8607 75
8608 76 a00400B 86FA LDAA #PERIOD ;Set PWM Period for channel 3
8609 77 a00400D 5AB7 STAA PWMPER3
8610 78
8611 79 a00400F 8608 LDAA #08 ;Enable PWM channel 3
8612 80 a004011 5AA0 STAA PWME
8613 81 a004013 5AA1 STAA PWMPOL ;Set polarity (the bulb will now be off when at 0% DT CYCLE)
8614 82
8615 83 a004015 7900 BF CLR PWMDTY3 ;PWMDTY3 = 0
8616 84
8617 85 a004018 8603 LDAA #$03
8618 86 a00401A 5A4D STAA TSCR2 ;Set prescaler to 8
8619 87
8620 88 a00401C C601 LDAB #$01
8621 89 a00401E 5B40 STAB TIOS ;sets channel 0 to act as output compare
8622 90 a004020 5B4C STAB TIE ;enable interrupt for compare 0
8623 91
8624 92
8625 93 a004022 CCC3 50 LDD #const1 ;D = Const1
8626 94 a004025 5C50 STD TC0 ;TC0 = D
8627 95
8628 96 a004027 8680 LDAA #$80
8629 97 a004029 5A46 STAA TSCR1 ;Turn on Timer
8630 98
8631 99 a00402B 86FF LDAA #$FF
8632 100 a00402D 5A02 STAA DDRA ;Sets portA as output
8633 101
8634 102 a00402F 10EF CLI ;enable interrupts
8635 103
8636 104 a004031 87 CLRA
8637 105 a004032 8680 LDAA #$80 ;A=$80

```

```

8638 106 a004034 7A01 22          STAA ATD1CTL2          ;Turn on A/D
8639 107 a004037 8608          LDAA #$08          ;A=$08
8640 108 a004039 7A01 23          STAA ATD1CTL3          ;once conversion/cycle
8641 109
8642 110
8643 111          ;*****
8644 112          ;*                      Main                      *
8645 113          ;*****
8646 114
8647 115 a00403C 8661      Start:    LDAA #$61
8648 116 a00403E 7A01 24          STAA ATD1CTL4          ;2mHz CLK 10 bit mode
8649 117
8650 118 a004041 8682          LDAA #$82          ;A=$82
8651 119 a004043 7A01 25          STAA ATD1CTL5          ;Start A/D channel 2
8652 120 a004046 1F01 2682 Here:    BRCLR ATD1STAT0,$82,Here
      00404A FB
8653 121 a00404B B601 30          LDAA ATD1DR0H          ;Get 10-bit result (high)
8654 122 a00404E F601 31          LDAB ATD1DR0L          ;Get 10-bit result (low)
8655 123
8656 124 a004051 CD01 E9          LDY #489          ;Y = 489
8657 125 a004054 13          EMUL          ;Y:D = D * Y
8658 126 a004055 CE03 E8          LDX #1000          ;X = 1000
8659 127 a004058 1814          EDIVS          ;Y = Y:D/X
8660 128
8661 129 a00405A B7C6          XGDY          ;Y -> D, D -> Y
8662 130 a00405C 8301 11          SUBD #273          ;D = D - 273
8663 131 a00405F 7B04 01          STAB Pval          ;Pval = B (celcius)
8664 132 a004062 7B04 00          STAB myval          ;myval = B
8665 133
8666 134 a004065 1640 7B          JSR SEPERATE          ;call "SEPERATE"
8667 135
8668 136 a004068 1640 9F          JSR SHOW          ;call "SHOW"
8669 137
8670 138 a00406B 1640 C6          JSR DELAY          ;call DELAY
8671 139
8672 140
8673 141 a00406E 0640 3C          JMP Start
8674 142
8675 143
8676 144 a004071 5F06 3B2F display: fcb $5f,$06,$3b,$2f,$66,$6d,$7d,$07,$7F,$67
      004075 666D 7D07
      004079 7F67
8677 145
8678 146          ;*****
8679 147          ;*                      *
8680 148          ;*      SUBROUTINE:SEPERATE      *
8681 149          ;*      IN:Nothing      *
8682 150          ;*      OUT:Nothing      *
8683 151          ;*                      *
8684 152          ;*      The purpose of this subroutine is to      *
8685 153          ;*      seperate a decimal number into two      *
8686 154          ;*      variables so they can be displayed one at      *
8687 155          ;*      a time on a 7-segment display.      *
8688 156          ;*                      *
8689 157          ;*                      *
8690 158          ;*      Created by Brandon Empie      *
8691 159          ;*                      *
8692 160          ;*****
8693 161
8694 162 a00407B 36      SEPERATE: PSHA          ;Saves A
8695 163
8696 164 a00407C 7904 02          CLR tens          ;tens = 0
8697 165 a00407F 7904 03          CLR ones          ;ones = 0
8698 166
8699 167 a004082 860A          LDAA #10          ;A = 10
8700 168
8701 169 a004084 F704 00      TEST:    TST myval          ;myval = 0?
8702 170 a004087 2714          BEQ DONE          ;if true branch to done
8703 171
8704 172 a004089 7304 00          DEC myval          ;myval = myval - 1
8705 173 a00408C 7204 03          INC ones          ;ones = ones + 1
8706 174
8707 175 a00408F B104 03          CMPA ones          ;ones = 10?
8708 176 a004092 26F0          BNE TEST          ;branch to test if not equal
8709 177
8710 178 a004094 7204 02          INC tens          ;tens = tens + 1
8711 179
8712 180 a004097 7904 03          CLR ones          ;ones = 0
8713 181 a00409A 0640 84          JMP TEST          ;go to Test
8714 182

```

```

8715 183 a00409D 32      DONE:      PULA      ;Restore A
8716 184 a00409E 3D      RTS          ;Return
8717 185
8718 186                ;*****
8719 187                ;*
8720 188                ;*      SUBROUTINE: SHOW
8721 189                ;*      IN:Nothing
8722 190                ;*      OUT:Nothing
8723 191                ;*
8724 192                ;*      The purpose of this subroutine is to show
8725 193                ;*      the tempeture in celcius on the 7-segment
8726 194                ;*      display
8727 195                ;*
8728 196                ;*      Created by Brandon Empie
8729 197                ;*
8730 198                ;*****
8731 199
8732 200 a00409F 34      SHOW: PSHX          ;Saves X
8733 201 a0040A0 37      PSHB          ;Saves B
8734 202 a0040A1 36      PSHA          ;Saves A
8735 203
8736 204 a0040A2 CE40 71      LDX #display      ;X points to first element of array
8737 205
8738 206 a0040A5 F604 02      LDAB tens          ;B = tens
8739 207 a0040A8 A6E5      LDAA B,x          ;A = (display(x) + B)
8740 208 a0040AA 5A00      STAA PORTA        ;PORTA = A
8741 209
8742 210 a0040AC 1640 C6      JSR DELAY        ;call 'DELAY'
8743 211
8744 212
8745 213 a0040AF 87          CLRA          ;A = 0
8746 214 a0040B0 5A00      STAA PORTA        ;PORTA = 0
8747 215 a0040B2 1640 C6      JSR DELAY        ;call 'DELAY'
8748 216
8749 217 a0040B5 F604 03      LDAB ones          ;B = ones
8750 218 a0040B8 A6E5      LDAA B,x          ;A = display(x) + B)
8751 219 a0040BA 5A00      STAA PORTA        ;PORTA = A
8752 220
8753 221 a0040BC 1640 C6      JSR DELAY        ;call 'DELAY'
8754 222
8755 223
8756 224 a0040BF 87          CLRA          ;A = 0
8757 225 a0040C0 5A00      STAA PORTA        ;PORTA = 0
8758 226
8759 227 a0040C2 32          PULA          ;Saves A
8760 228 a0040C3 33          PULB          ;Saves B
8761 229 a0040C4 30          PULX          ;Restore X
8762 230 a0040C5 3D          RTS          ;Return
8763 231
8764 232                ;*****
8765 233                ;*
8766 234                ;*      SUBROUTINE: DELAY
8767 235                ;*      IN:Nothing
8768 236                ;*      OUT:Nothing
8769 237                ;*
8770 238                ;*      The purpose of this subroutine is to create
8771 239                ;*      A delay.
8772 240                ;*
8773 241                ;*      Created by Brandon Empie
8774 242                ;*
8775 243                ;*****
8776 244
8777 245 a0040C6 34      DELAY: PSHX          ;Saves X
8778 246 a0040C7 35      PSHY          ;Saves Y
8779 247 a0040C8 CD00 05      LDY #$05          ;Y = $07
8780 248 a0040CB CEFF FF      LDY #$FFFF        ;X = $FFFF
8781 249 a0040CE A7      WAIT:  NOP          ;do nothing
8782 250 a0040CF A7      NOP          ;do nothing
8783 251 a0040D0 A7      NOP          ;do nothing
8784 252
8785 253 a0040D1 09          DEX          ;X = X - 1
8786 254 a0040D2 26FA      BNE WAIT        ;Branch to WAIT if X isn't 0
8787 255
8788 256 a0040D4 03          DEY          ;Y = Y - 1
8789 257 a0040D5 26F7      BNE WAIT        ;Branch to WAIT if Y isn't 0
8790 258
8791 259 a0040D7 31          PULY          ;Restore Y
8792 260 a0040D8 30          PULX          ;Restore X
8793 261 a0040D9 3D          RTS          ;Return
8794 262

```

```

8795 263
8796 264
8797 265
8798 266
8799 267
8800 268
8801 269
8802 270
8803 271
8804 272
8805 273
8806 274
8807 275
8808 276
8809 277
8810 278
8811 279
8812 280
8813 281 a0040DA F700 01
8814 282 a0040DD 2706
8815 283
8816 284 a0040DF 1641 08
8817 285
8818 286 a0040E2 0640 FF
8819 287
8820 288
8821 289 a0040E5 8620
8822 290
8823 291 a0040E7 B004 01
8824 292
8825 293 a0040EA 97
8826 294 a0040EB 2F10
8827 295
8828 296 a0040ED C614
8829 297 a0040EF 12
8830 298 a0040F0 C300 10
8831 299
8832 300 a0040F3 8C00 FF
8833 301 a0040F6 2F07
8834 302
8835 303 a0040F8 C6FF
8836 304
8837 305 a0040FA 0640 FF
8838 306
8839 307
8840 308 a0040FD D610
8841 309
8842 310
8843 311 a0040FF 5BBF
8844 312
8845 313
8846 314 a004101 7900 44
8847 315 a004104 4C4E 01
8848 316
8849 317
8850 318 a004107 0B
8851 319
8852 320
8853 321
8854 322
8855 323
8856 324
8857 325
8858 326
8859 327
8860 328
8861 329
8862 330
8863 331
8864 332
8865 333
8866 334
8867 335 a004108 36
8868 336 a004109 34
8869 337
8870 338 a00410A 86E1
8871 339 a00410C 7A01 24
8872 340
8873 341
8874 342 a00410F 8680

;*****
;*
;*
;*   ISR: TIMER
;*   IN:Nothing
;*   OUT:Nothing
;*
;*
;*   The purpose of this Interrupt is to
;*   update lamp brightness (heat) in manual mode or
;*   based on proportioncontrol algorithm:
;*   Output = (setpoint - present value)*gain + offset
;*   depending on the state of PORTB
;*
;*
;*   Created by Brandon Empie
;*
;*****

TIMER:   TST PORTB           ;Does PORTB = 0?
          BEQ Control        ;Branch to Control if it does
          JSR Manual          ;Call Manual
          JMP GO              ;Jump to GO

Control: LDAA #Spoint        ;A = Spoint
          SUBA Pval           ;A = A - Pval
          TSTA                ;A - 0, result not stored, CCR updated
          BLE COOL            ;Branch if less than or equal to zero
          LDAB #Gain          ;B = Gain
          MUL                 ;D = A * B
          ADDD #Adjust        ;D = D + Adjust (offset)
          CPD #$00FF
          BLE GO              ;Is D < or = FF? branch to go if it is
          LDAB #$FF           ;B = $FF
          JMP GO              ;Jump to go

COOL:    LDAB Adjust          ;B = Adjust

GO:       STAB PWMDTY3        ;PWMDTY = B
          CLR TCNT            ;TCNT = $0000
          BSET TFLG1,$01      ;clear TFLG1 for next timer interrupt

          RTI                 ;Return

;*****
;*
;*
;*   SUBROUTINE: Manual
;*   IN:Nothing
;*   OUT:B
;*
;*
;*   The purpose of this subroutine is to place the
;*   lamp into manual mode based on PORTBs evaluation
;*   in the Timer ISR. B is being returned to update
;*   the lamps brightness to reflect POT1s state.
;*
;*
;*   Created by Brandon Empie
;*
;*****

Manual:   PSHA                ;Save A
          PSHX                ;Save X
          LDAA #$E1
          STAA ATD1CTL4       ;Set A/D for 8 bit mode
          LDAA #$80            ;A=$80

```

8875	343	a004111 7A01 25		STAA ATD1CTL5	;Start A/D channel 0
8876	344	a004114 1F01 2680	Check:	BRCLR ATD1STAT0,\$80,Check	
		004118 FB			
8877	345	a004119 F601 31		LDAB ATD1DR0L	;Get 8-bit result
8878	346				
8879	347	a00411C 8664		LDAA #BILL	;A = BILL
8880	348	a00411E 12		MUL	;D = A * B
8881	349	a00411F CE00 FF		LDX #255	;X = 255
8882	350	a004122 1810		IDIV	;X = D/X r -> D
8883	351	a004124 B7C5		XGDX	;X -> D, D -> X
8884	352				
8885	353	a004126 86FA		LDAA #PERIOD	;A = PERIOD
8886	354	a004128 12		MUL	;D = A * B
8887	355				
8888	356	a004129 CE00 64		LDX #BILL	;X = BILL
8889	357	a00412C 1810		IDIV	;X = D/X r -> D
8890	358	a00412E B7C5		XGDX	;X -> D, D -> X
8891	359				
8892	360	a004130 30		PULX	;Restore X
8893	361	a004131 32		PULA	;Restore A
8894	362				
8895	363	a004132 3D		RTS	;Return (to Timer ISR)
8896	364				
8897	365				
8898	366				
8899	367				
8900	368				
8901	369	a00FFFE 4000		ORG \$FFFE	
8902	370			DC.W Entry	;Reset Vector
8903	371				
8904	372	a00FFEE 40DA		ORG \$FFEE	;Timer channel 0 Vector
				DC.W TIMER	