```
a000400             myval ds.b 1
a000401             Pval  ds.b 1
a000402             tens  ds.b 1
a000403             ones  ds.b 1

        0000 C350   const1 equ $C350 ;TCNT = ((intTime/(1/BusCLK)) / Pre) = ((100ms/(1/4Mhz)) / 8) = 50,000 = $C350

        0000 0020   Spoint equ 32
        0000 0014   Gain   equ $14
        0000 0010   Adjust equ $10   ;Offset

        0000 0070   PRE       EQU $70 ;PWM Prescaler
        0000 00FA   PERIOD    EQU $FA ;PWM Period
        0000 0000   PWMCLOCK  EQU $00 ;0 when scaler is not used, $08 when scaler is used

        0000 0064   BILL      EQU 100 ;Used for calculating around decimals

                    ;code section
                                ORG   ROMStart

                    ;*****************************************************
                    ;*                                                   *
                    ;*                                                   *
                    ;*    ECNS-414                                       *
                    ;*    29 Oct 2020                                    *
                    ;*    Lab 7                                          *
                    ;*                                                   *
                    ;*    The purpose of this program is to construct    *
                    ;*    and demonstrate a closed loop control system   *
                    ;*    using pulse width modulation.                  *
                    ;*                                                   *
                    ;*    Created by: Brandon Empie                      *
                    ;*                                                   *
                    ;*****************************************************


                    ;**************************************************************
                    ;*                 Initilizations                             *
                    ;**************************************************************
                    Entry:
                    _Startup:
a004000 CF20 00                 LDS  #RAMEnd+1        ;initialize the stack pointer

a004003 8600                    LDAA #PWMCLOCK        ;turns on scaler B for channel 3
a004005 5AA2                    STAA PWMCLK

a004007 8670                    LDAA #PRE             ;Set PWM PRE
a004009 5AA3                    STAA PWMPRCLK


a00400B 86FA                    LDAA #PERIOD          ;Set PWM Period for channel 3
a00400D 5AB7                    STAA PWMPER3

a00400F 8608                    LDAA #08              ;Enable PWM channel 3
a004011 5AA0                    STAA PWME
a004013 5AA1                    STAA PWMPOL           ;Set polarity (the bulb will now be off when at 0% DTYCYCLE)

a004015 7900 BF                 CLR PWMDTY3           ;PWMDTY3 = 0

a004018 8603                    LDAA #$03
a00401A 5A4D                    STAA TSCR2            ;Set prescaler to 8

a00401C C601                    LDAB #$01
a00401E 5B40                    STAB TIOS             ;sets channel 0 to act as output compare
a004020 5B4C                    STAB TIE              ;enable interrupt for compare 0


a004022 CCC3 50                 LDD #const1           ;D = Const1
a004025 5C50                    STD TC0               ;TC0 = D

a004027 8680                    LDAA #$80
a004029 5A46                    STAA TSCR1            ;Turn on Timer

a00402B 86FF                    LDAA #$FF
a00402D 5A02                    STAA DDRA             ;Sets portA as output

a00402F 10EF                    CLI                   ;enable interrupts

a004031 87                      CLRA
a004032 8680                    LDAA #$80             ;A=$80
```

```
a004034 7A01 22                 STAA ATD1CTL2         ;Turn on A/D
a004037 8608                    LDAA #$08             ;A=$08
a004039 7A01 23                 STAA ATD1CTL3         ;once conversion/cycle


                    ;**************************************************************
                    ;*                        Main                                *
                    ;**************************************************************

a00403C 8661        Start:      LDAA #$61
a00403E 7A01 24                 STAA ATD1CTL4         ;2mHz CLK 10 bit mode

a004041 8682                    LDAA #$82             ;A=$82
a004043 7A01 25                 STAA ATD1CTL5         ;Start A/D channel 2
a004046 1F01 2682   Here:       BRCLR ATD1STAT0,$82,Here
 00404A FB
a00404B B601 30                 LDAA ATD1DR0H          ;Get 10-bit result  (high)
a00404E F601 31                 LDAB ATD1DR0L          ;Get 10-bit result  (low)

a004051 CD01 E9                 LDY #489               ;Y = 489
a004054 13                      EMUL                   ;Y:D = D * Y
a004055 CE03 E8                 LDX #1000              ;X = 1000
a004058 1814                    EDIVS                  ;Y = Y:D/X

a00405A B7C6                    XGDY                   ;Y -> D, D -> Y
a00405C 8301 11                 SUBD #273              ;D = D - 273
a00405F 7B04 01                 STAB Pval              ;Pval = B (celcius)
a004062 7B04 00                 STAB myval             ;myval = B

a004065 1640 7B                 JSR SEPERATE           ;call "SEPERATE"

a004068 1640 9F                 JSR SHOW               ;call "SHOW"

a00406B 1640 C6                 JSR DELAY              ;call DELAY


a00406E 0640 3C                 JMP Start


a004071 5F06 3B2F   display:    fcb $5f,$06,$3b,$2f,$66,$6d,$7d,$07,$7F,$67
 004075 666D 7D07
 004079 7F67

                    ;****************************************************
                    ;*                                                  *
                    ;*     SUBROUTINE:SEPERATE                          *
                    ;*     IN:Nothing                                   *
                    ;*     OUT:Nothing                                  *
                    ;*                                                  *
                    ;*    The purpose of this subroutine is to          *
                    ;*    seperate a decimal number into two            *
                    ;*    variables so they can be displayed one at     *
                    ;*    a time on a 7-segment display.                *
                    ;*                                                  *
                    ;*                                                  *
                    ;*    Created by Brandon Empie                      *
                    ;*                                                  *
                    ;****************************************************

a00407B 36          SEPERATE: PSHA        ;Saves A

a00407C 7904 02               CLR tens    ;tens = 0
a00407F 7904 03               CLR ones    ;ones = 0

a004082 860A                  LDAA #10    ;A = 10

a004084 F704 00     TEST:     TST myval   ;myval = 0?
a004087 2714                  BEQ DONE    ;if true branch to done

a004089 7304 00               DEC myval    ;myval = myval - 1
a00408C 7204 03               INC ones    ;ones = ones + 1

a00408F B104 03               CMPA ones   ;ones = 10?
a004092 26F0                  BNE TEST    ;branch to test if not equal

a004094 7204 02               INC tens    ;tens = tens + 1

a004097 7904 03               CLR ones    ;ones = 0
a00409A 0640 84               JMP TEST    ;go to Test

```

```
a00409D 32          DONE:     PULA        ;Restore A
a00409E 3D                    RTS         ;Return

                    ;***************************************************
                    ;*                                                 *
                    ;*     SUBROUTINE: SHOW                            *
                    ;*     IN:Nothing                                  *
                    ;*     OUT:Nothing                                 *
                    ;*                                                 *
                    ;*    The purpose of this subroutine is to show    *
                    ;*    the tempeture in celcius on the 7-segment    *
                    ;*    display                                      *
                    ;*                                                 *
                    ;*    Created by Brandon Empie                     *
                    ;*                                                 *
                    ;***************************************************

a00409F 34          SHOW: PSHX                  ;Saves X
a0040A0 37                PSHB                  ;Saves B
a0040A1 36                PSHA                  ;Saves A

a0040A2 CE40 71           LDX #display          ;X points to first element of array

a0040A5 F604 02           LDAB tens             ;B = tens
a0040A8 A6E5              LDAA B,x              ;A = (display(x) + B)
a0040AA 5A00              STAA PORTA            ;PORTA = A

a0040AC 1640 C6           JSR DELAY             ;call 'DELAY'


a0040AF 87                CLRA                  ;A = 0
a0040B0 5A00              STAA PORTA            ;PORTA = 0
a0040B2 1640 C6           JSR DELAY             ;call 'DELAY'

a0040B5 F604 03           LDAB ones             ;B = ones
a0040B8 A6E5              LDAA B,x              ;A = display(x) + B)
a0040BA 5A00              STAA PORTA            ;PORTA = A

a0040BC 1640 C6           JSR DELAY             ;call 'DELAY'


a0040BF 87                CLRA                  ;A = 0
a0040C0 5A00              STAA PORTA            ;PORTA = 0

a0040C2 32                PULA                  ;Saves A
a0040C3 33                PULB                  ;Saves B
a0040C4 30                PULX                  ;Restore X
a0040C5 3D                RTS                   ;Return

                    ;***************************************************
                    ;*                                                 *
                    ;*     SUBROUTINE: DELAY                           *
                    ;*     IN:Nothing                                  *
                    ;*     OUT:Nothing                                 *
                    ;*                                                 *
                    ;*    The purpose of this subroutine is to create  *
                    ;*    A delay.                                     *
                    ;*                                                 *
                    ;*    Created by Brandon Empie                     *
                    ;*                                                 *
                    ;***************************************************

a0040C6 34          DELAY:    PSHX                    ;Saves X
a0040C7 35                    PSHY                    ;Saves Y
a0040C8 CD00 05               LDY   #$05              ;Y = $07
a0040CB CEFF FF               LDX   #$FFFF            ;X = $FFFF
a0040CE A7          WAIT:     NOP                     ;do nothing
a0040CF A7                    NOP                     ;do nothing
a0040D0 A7                    NOP                     ;do nothing

a0040D1 09                    DEX                     ;X = X - 1
a0040D2 26FA                  BNE WAIT                ;Branch to WAIT if X isn't 0

a0040D4 03                    DEY                     ;Y = Y - 1
a0040D5 26F7                  BNE WAIT                ;Branch to WAIT if Y isn't 0

a0040D7 31                    PULY                    ;Restore Y
a0040D8 30                    PULX                    ;Restore X
a0040D9 3D                    RTS                     ;Return

```

```
                    ;**********************************************************
                    ;*                                                        *
                    ;*     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                            *
                    ;*                                                        *
                    ;**********************************************************

a0040DA F700 01     TIMER:     TST PORTB              ;Does PORTB = 0?
a0040DD 2706                   BEQ Control            ;Branch to Control if it does

a0040DF 1641 08                JSR Manual             ;Call Manual

a0040E2 0640 FF                JMP GO                 ;Jump to GO


a0040E5 8620        Control:   LDAA #Spoint           ;A = Spoint

a0040E7 B004 01                SUBA Pval              ;A = A - Pval

a0040EA 97                     TSTA                   ;A - 0, result not stored, CCR updated
a0040EB 2F10                   BLE  COOL              ;Branch if less than or equal to zero

a0040ED C614                   LDAB #Gain             ;B = Gain
a0040EF 12                     MUL                    ;D = A * B
a0040F0 C300 10                ADDD #Adjust           ;D = D + Adjust (offset)

a0040F3 8C00 FF                CPD #$00FF
a0040F6 2F07                   BLE GO                 ;Is D < or = FF? branch to go if it is

a0040F8 C6FF                   LDAB #$FF              ;B = $FF

a0040FA 0640 FF                JMP GO                 ;Jump to go


a0040FD D610        COOL:      LDAB Adjust            ;B = Adjust


a0040FF 5BBF        GO:        STAB PWMDTY3           ;PWMDTY = B


a004101 7900 44                CLR TCNT               ;TCNT = $0000
a004104 4C4E 01                BSET TFLG1,$01         ;clear TFLG1 for next timer interrupt


a004107 0B                     RTI                    ;Return

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

a004108 36          Manual:     PSHA                  ;Save A
a004109 34                      PSHX                  ;Save X

a00410A 86E1                    LDAA #$E1
a00410C 7A01 24                 STAA ATD1CTL4         ;Set A/D for 8 bit mode


a00410F 8680                    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                      ;*                 Interrupt Vectors                           *
8899  367                      ;***************************************************************
8900  368                                  ORG   $FFFE
8901  369  a00FFFE 4000                    DC.W  Entry           ;Reset Vector
8902  370
8903  371                                  ORG   $FFEE           ;Timer channel 0 Vector
8904  372  a00FFEE 40DA                    DC.W  TIMER
```