XDEF hexkeypad

XREF rows

XREF lookup, shutsound, Alternator

XREF PTU, pushpress, port\_p, MAX

XREF TRACKER, LEDroutine, setLEDTRACKER, switchchange

XREF delaytimer, delayon, power\_output, pot\_value, read\_pot, TON, real\_power\_output

XREF gen1off, gen2off, gen3off, value, sum, real\_value, delaytimers, delayons

XREF shutdown, shutoff, potflg, priorvalue, startup, HOMEflg

XREF sendhome, delayonm, delaytimerm, letknow, display\_string, disp

;This is the keypad

;This is almost pulled straight from the lab we did it

;The major change is that it will leave to update the home screen everyonce and a while

; It also has a few different flags in case things like the emergency shutoff or a switch has changed

; There is a second keypad in a nother file that is similar but does not check as many things

hexkeypad:

LDAA HOMEflg ; check if program is home

CMPA #1

BNE nothome

MOVB #1, delayons ; want to initiate a short timer if home to adjust

; home screen when necessary

MOVB #0, delaytimers

nothome:MOVB #1, delayonm ; set a longer timer to leave menu if idle for too long

MOVB #0, delaytimerm

PSHX

PSHY

loop: LDX #rows ; check rows of keypad

loop1: CPX #rows+4

BEQ loop

LDAA shutoff ; check if a shutoff has commenced and leave keypad

CMPA #1

BEQ shutdown1jmp

BRA pastshutdown

shutdown1jmp: JMP shutdown1

pastshutdown:

LDAA letknow ; check if auto shutoff has occured and leave keypad

CMPA #1

BEQ autoshutstuffjmp

BRA pushing

autoshutstuffjmp: JSR autoshutstuff

pushing:

PSHD

PSHX

PSHY

JSR read\_pot ; check pot value

LDD pot\_value

STD power\_output

CMPB MAX ; set a new max value if pot surpasses previous max value

BLS nonew

STAB MAX

nonew: LDX #3 ; want to set PWM counter so divide pot value by 3 generators

IDIV

TFR x, b

STAB real\_value ; transfer quotient to accumulator B and store value to memory

LDAA gen1off ; check if generator 1 is on

CMPA #$1

BEQ skip1

LDAA sum ; if generator 1 is on, add third of pot value to sum

ADDA real\_value

STAA sum

skip1:

LDAA gen2off ; see generator 1 steps for generators 2 and 3

CMPA #$1

BEQ skip2

LDAA sum

ADDA real\_value

STAA sum

skip2:

LDAA gen3off

CMPA #$1

BEQ skip3

LDAA sum

ADDA real\_value

STAA sum

skip3:

LDAA sum

STAA TON ; store total sum into PWM value

dismiss:

PULD

PULX

PULY

MOVB #0, sum ; reset sum value

LDAA port\_p ; if stepper motor is operating, leave keypad

ANDA #$20

CMPA #0

BEQ gohomepp

LDAA switchchange ; if a switch was changed, leave keypad

CMPA #1

BNE checkLED

BRA gohome

gohomepp: MOVB #1, pushpress

BRA gohome

shutdown1:

MOVB #0, shutoff

JSR shutdown

BRA gohome

checkLED: LDAA HOMEflg ; check if program is at home screen

CMPA #1

BNE skipthis

LDAA delaytimers ; if 25ms goes by and no button was pressed,

; exit keypad and check leds

CMPA #25

BEQ gohomey

skipthis: LDAA delaytimerm ; leave keypad if 10 seconds goes by with no user

; input

CMPA #10

BEQ gohomey

LDAA LEDroutine ; check if in generator menu

CMPA #1

BNE checktrack

LDAA setLEDTRACKER ; check if generator led capacity needs to be changed

CMPA #1

BNE checktrack

MOVB #0, setLEDTRACKER

BRA gohome

checktrack: LDAA TRACKER

CMPA #0

BNE gohome

LDAA 1, x+ ; checking if user has pressed a button

STAA PTU

JSR debounce

ANDA #$0F ; check if a button was pressed

CMPA #$0F

BEQ loop1jmp

LDAA potflg

MOVB real\_power\_output, priorvalue

CMPA #1

BEQ gohome

BRA letgo

loop1jmp: jmp loop1

letgo: LDAA PTU

ANDA #$0F ; dont progress until user lets go of keypad button

CMPA #$0F

BNE letgo

LDAA #0 ; translate coded value from keypad to the number/letter we expect in

; next 8 lines

LDY #lookup

loop2: CMPB 1, y+

BEQ gohome

inca

CPY #lookup + 16

BNE loop2

BRA loopjmp

loopjmp: jmp loop

gohome: MOVB #0, delayon

MOVB #0, delaytimer ; turn off any timers

PULY

PULX

RTS

gohomey:

MOVB #1, sendhome ; let program know that it needs to go to home

; screen

MOVB #0, delayonm

MOVB #0, delaytimerm ; reset timers

MOVB #0, delayons

MOVB #0, delaytimers

PULY

PULX

RTS

debounce:

JSR delay

LDAA PTU ; accumulator a has button pressed

TAB

RTS

delay:

LDY #1000

loop4: DEY

BNE loop4

RTS

autoshutstuff:

movb #'A',disp ; wasnt to display that an autoshutoff completed

movb #'u',disp+1

movb #'t',disp+2

movb #'o',disp+3

movb #'-',disp+4

movb #'S',disp+5

movb #'h',disp+6

movb #'u',disp+7

movb #'t',disp+8

movb #'o',disp+9

movb #'f',disp+10

movb #'f',disp+11

movb #' ',disp+12

movb #' ',disp+13

movb #' ',disp+14

movb #' ',disp+15

movb #'C',disp+16

movb #'o',disp+17

movb #'m',disp+18

movb #'p',disp+19

movb #'l',disp+20

movb #'e',disp+21

movb #'t',disp+22

movb #'e',disp+23

MOVB #0, Alternator

MOVB #1, shutsound ; want autoshutoff sound to go off

LDD #disp

JSR display\_string ; display on LCD

MOVB #1, delayon ; next 5 lines keep this on lcd for specified time

MOVB #0, delaytimer

stud: LDAA delaytimer

CMPA #18

BNE stud

MOVB #0, delayon

MOVB #0, delaytimer

movb #' ',disp

movb #' ',disp+1

movb #' ',disp+2

movb #' ',disp+3

movb #' ',disp+4

movb #' ',disp+5

movb #' ',disp+6

movb #' ',disp+7

movb #' ',disp+8

movb #' ',disp+9

movb #' ',disp+10

movb #' ',disp+11

movb #' ',disp+12

movb #' ',disp+13

movb #' ',disp+14

movb #' ',disp+15

movb #' ',disp+16

movb #' ',disp+17

movb #' ',disp+18

movb #' ',disp+19

movb #' ',disp+20

movb #' ',disp+21

movb #' ',disp+22

movb #' ',disp+23

MOVB #0, letknow

RTS