XDEF menu

XREF hexkeypad

XREF display\_string

XREF gen1cap

XREF gen2cap

XREF gen3cap

XREF autoshut1

XREF autoshut2

XREF autoshut3

XREF disp2

XREF disp5

XREF disp6, disp3

XREF disp7, HOMEflg

XREF disp8, mu

XREF disp9

XREF dispa, disph

XREF dispb

XREF port\_s

XREF port\_t

XREF disp1

XREF gen1flg, sendhome

XREF gen2flg, stepper

XREF gen3flg, pushpress

XREF noTRACKER, LEDroutine, Alternator,

XREF switch\_file, switchchange, waithold, gen1off, gen2off, gen3off

; This is were all of our menues are stored.

; They ranged from selecting generators to changing your password.

; All of the menues work the same.

; 1)Update the LCD to show what you are on

; 2)Keypad waits for A, B, or F

; 3)If A or B are pressed it sends them to the next option then go back to 1

; 4)If F is pressed then it sends them to where they want

; Every menu is set on a loop using braches

; If F is pressed it sends them to the picked loction.

; The date and time, ID, and password all send them to a similar code to when they set it

; There is also a timer in the keypard that when it hits 10 seconds it will boot you home

menu: MOVB #0, HOMEflg ; let program know not at home screen

MOVB #0, noTRACKER ; don’t prematurely leave keypad

MOVB #$17, $410

LDX #16

LDY #10

passcheck: CPX #20 ; verify correct password before progressing

BEQ control

PSHX

LDD #disp5

JSR display\_string

PULX

for1: JSR hexkeypad

CMPA #$FF

BEQ passcheck

LDAB sendhome ; might have received send home form keypad file

CMPB #1

BEQ HOME

LDAB switchchange

CMPB #1 ; check for switch changes

BEQ jmptoswitchfile

CMPA #9

BLE number3

ADDA #$7

number3: STAA $411 ;Address 411 contains button pressed for password

LDAB disp2, x

SUBB #$30

CMPB $411 ; if wrong password, send home

BNE HOME

LDAA #'\*' ; show progress on LCD

STAA disp5,y

INX

INY

BRA passcheck

jmptoswitchfile: JSR switch\_file

MOVB #0, port\_s

RTS

HOME: movb #'X',disp5+10

movb #'X',disp5+11 ;reset password string to default and send home

movb #'X',disp5+12

movb #'X',disp5+13

MOVB #1, noTRACKER

MOVB #0, sendhome

RTS

control:

movb #'X',disp5+10

movb #'X',disp5+11

movb #'X',disp5+12 ; reset password string to default

movb #'X',disp5+13

genclick1:

LDAA #'-' ;set arrow to be next to generator

STAA disp7

LDAA #'>'

STAA disp7+1

LDAA #' '

STAA disp7+17 ; erase arrow from other areas

STAA disp7+16

genclick2: LDD #disp7

JSR display\_string

JSR hexkeypad ;user input

CMPA #$FF

BEQ genclick2

LDAB sendhome

CMPB #1

BEQ HOME

LDAB switchchange

CMPB #1

BEQ jmptoswitchfile2

CMPA #$B ;check and see if B was pressed to move arrow or if F was

; pressed to select button

BEQ passclick1

CMPA #$A ; pressing A goes “back” on menu

BEQ MUTEjmp

CMPA #$F

BEQ GENJUMP

BRA genclick2

MUTEjmp: JMP MUTE

jmptoswitchfile2:

JSR switch\_file

LDD #disp7 ; return to menu after a switch routine

JSR display\_string

MOVB #0, waithold

MOVB #0, port\_s

BRA genclick2

HOMEjmp: JMP HOME

GENJUMP: JMP GenSel ;need jump as GenSel is too far for a normal branch

genclick1jmp: JMP genclick1

passclick1: LDAA #' ' ;erase arrow from other areas

STAA disp7

STAA disp7+1

LDAA #'-'

STAA disp7+16 ; set arrow next to password change

LDAA #'>'

STAA disp7+17

passclick2: LDD #disp7

JSR display\_string

JSR hexkeypad

LDAB sendhome

CMPB #1

BEQ HOMEjmp

LDAB switchchange

CMPB #1

BEQ jmptoswitchfile3

CMPA #$A ; check if A, B, or F was pressed. A will move arrow back to

; generator, B will move arrow to DT, F will select

BEQ genclick1jmp

CMPA #$B

BEQ DT1

CMPA #$F

BEQ JUMPSEL

BRA passclick2

jmptoswitchfile3: JSR switch\_file

LDD #disp7 ; return to menu after switch routine

JSR display\_string

MOVB #0, waithold

MOVB #0, port\_s

BRA passclick2

passclick1jmp: JMP passclick1

JUMPSEL: JMP passSel

nextSel:

DT1: LDAA #' '

STAA disp8+12 ; erase arrow from other areas

STAA disp8+13

STAA disp8+16

STAA disp8+17

LDAA #'-'

STAA disp8

LDAA #'>' ; move arrow next to date time set

STAA disp8+1

DT2: LDD #disp8

JSR display\_string

JSR hexkeypad ; user input

LDAB sendhome

CMPB #1

BEQ HOMEjmper

LDAB switchchange

CMPB #1

BEQ jmptoswitchfile4

CMPA #$A ; check if a or f has been pressed. F will select date and time, A move arrow to password

BEQ passclick1jmp

CMPA #$B

BEQ menu1

CMPA #$F

BEQ DTJUMP

BRA DT2

jmptoswitchfile4: JSR switch\_file

LDD #disp8

JSR display\_string

MOVB #0, waithold

MOVB #0, port\_s

BRA DT2

DTJUMP: JMP DTSel

menu1: LDAA #' '

STAA disp8+1 ; erase arrow from other areas

STAA disp8

LDAA #'-'

STAA disp8+16

LDAA #'>' ; move arrow next to date time set

STAA disp8+17

menu3: LDD #disp8

JSR display\_string

JSR hexkeypad

LDAB sendhome

CMPB #1

BEQ HOMEjmper

LDAB switchchange

CMPB #1

BEQ jmptoswitchfile5

CMPA #$A ; check if a or f has been pressed. F will select date and time, A

; move arrow to password

BEQ DT1JMP

CMPA #$B

BEQ ID

CMPA #$F

BEQ HOMEjmper

BRA DT2

HOMEjmper: JMP HOME

jmptoswitchfile5: JSR switch\_file

LDD #disp8

JSR display\_string ; return to menu after switch routine

MOVB #0, waithold

MOVB #0, port\_s

BRA menu3

DT1JMP: JMP DT1

ID: LDAA #'>'

STAA disph+1

LDAA #'-' ; erase arrow from other areas

STAA disph

LDAA #' '

STAA disph+16

LDAA #' ' ; move arrow next to date time set

STAA disph+17

ID2: LDD #disph

JSR display\_string

JSR hexkeypad

LDAB sendhome

CMPB #1

BEQ HOMEjmper

LDAB switchchange

CMPB #1

BEQ jmptoswitchfilea

CMPA #$A ; check if a or f has been pressed. F will select date and time, A

; move arrow to password

BEQ Menu1jmp

CMPA #$B

BEQ MUTE

CMPA #$F

BEQ ChangeIDJMP

BRA ID

jmptoswitchfilea: JSR switch\_file

LDD #disp8

JSR display\_string ; return to menu after switch routine

MOVB #0, waithold

MOVB #0, port\_s

BRA ID

ChangeIDJMP: JMP ChangeID

Menu1jmp: JMP menu1

MUTE: LDAA #' '

STAA disph+1

LDAA #' ' ; erase arrow from other areas

STAA disph

LDAA #'-'

STAA disph+16

LDAA #'>' ; move arrow next to date time set

STAA disph+17

MUTE2: LDD #disph

JSR display\_string

JSR hexkeypad

LDAB sendhome

CMPB #1

BEQ HOMEjmperJMP

LDAB switchchange

CMPB #1

BEQ jmptoswitchfileb

CMPA #$A ; check if a or f has been pressed. F will select date and time, A

; move arrow to password

BEQ IDJMP

CMPA #$B

BEQ GENJUMPB

CMPA #$F

BEQ Update

BRA MUTE2

IDJMP JMP ID

HOMEjmperJMP: JMP HOMEjmper

GENJUMPB: JMP genclick1

Update: LDAA mu

CMPA #1

BEQ OFF ; update mute on LCD menu

MOVB #1, mu

LDAA #'O'

STAA disph+27

LDAA #'n'

STAA disph+28

LDAA #' '

STAA disph+29

BRA MUTE2

jmptoswitchfileb: JSR switch\_file

LDD #disp8

JSR display\_string

MOVB #0, waithold

MOVB #0, port\_s

BRA MUTE

OFF: MOVB #0,mu

LDAA #'O'

STAA disph+27

LDAA #'f'

STAA disph+28

LDAA #'f'

STAA disph+29

BRA MUTE2

ChangeID: ; same as setting ID in main file

LDD #disp3

JSR display\_string

LDX #9

STX $408 ;408 is a placeholder for x

CHID: LDAA disp3, x

JSR hexkeypad

CMPA #9

BLE numberID

ADDA #$37

STAA disp3,x

BRA skipnumberID

numberID: ADDA #$30

STAA disp3, x

skipnumberID: inx

iny

STX $408 ;408 is a placeholder for x

LDD #disp3

JSR display\_string

LDX $408

CPX #13

BEQ HOMEJUMP

BRA CHID

HOMEJUMP: JMP HOME

;CLICKJUMP: JMP genclick1

DTSel: ; same as setting date and time in main file

LDX #16

STX $408

setdate: ;Setting date and time

PSHX

LDD #disp1

JSR display\_string

PULX

LDAA disp1, x

CMPA #' '

BEQ increx1 ;Checking for /,:, and spaces

CMPA #'/'

BEQ increx1

CMPA #':'

BEQ increx1

for3: JSR hexkeypad

CMPA #$A

BHS for3

CMPA #$FF

BEQ setdate

LDAB sendhome

CMPB #1

BEQ HOMEJUMP

LDAB switchchange

CMPB #1

BEQ jmptoswitchfile6

CMPA #9

BLE number4

ADDA #$37

STAA disp1, x

STAA disp6, x

BRA increx1

number4: ADDA #$30

STAA disp1, x

STAA disp6, x

increx1: inx

STX $408 ;408 is a placeholder for x

LDD #disp1

JSR display\_string

LDX $408

CPX #32

BEQ home1

BRA setdate

jmptoswitchfile6: JSR switch\_file ; return to menu after switch routine

LDD #disp1

JSR display\_string

MOVB #0, waithold

BRA for3

passSel: ; same as password set in main file

LDX #16

STX $408 ;408 is a placeholder for x

setpass: LDAA disp2, x

PSHX

LDD #disp2

JSR display\_string

PULX

for4: JSR hexkeypad

LDAB sendhome

CMPB #1

BEQ home1

CMPA #$FF

BEQ setpass

LDAB switchchange

CMPB #1

BEQ jmptoswitchfile7

CMPA #9

BLE number5

ADDA #$37

STAA disp2, x

BRA skipnumber5

number5: ADDA #$30

STAA disp2, x

skipnumber5: inx

STX $408

LDX $408

CPX #20

BEQ home1

BRA setpass

jmptoswitchfile7: JSR switch\_file

LDD #disp2

JSR display\_string ; display menu after switch routine

MOVB #0, waithold

MOVB #0, port\_s

BRA for4

home1: JMP HOME

GenSel:

gen1: LDAA #$1 ;store current generator into address

STAA $450

LDAA #'-'

STAA disp9

LDAA #'>' ;move arrow to gen 1

STAA disp9+1

LDAA #' '

STAA disp9+17

STAA disp9+16 ; clear arrow from other areas

gen1a: LDD #disp9

JSR display\_string

JSR hexkeypad

LDAB sendhome

CMPB #1

BEQ home1

LDAB switchchange

CMPB #1

BEQ jmptoswitchfile8

CMPA #$B

BEQ gen2

CMPA #$A

BEQ homeJMP ; if F is pressed select gen 1, if b was pressed shift arrow to gen 2

CMPA #$F

BEQ JUMPPICK

BRA gen1a

homeJMP: JMP home

jmptoswitchfile8: JSR switch\_file

LDD #disp9

JSR display\_string

MOVB #0, waithold

MOVB #0, port\_s

BRA gen1a

JUMPPICK: JMP GenPick ; GenPick is too far, needed a jump

gen2: LDAA #$2 ;store current generator into address

STAA $450

LDAA #'-'

STAA disp9+16 ; move arrow next to gen 2

LDAA #'>'

STAA disp9+17

LDAA #' '

STAA disp9 ; clear arrow from other areas

STAA disp9+1

gen2a: LDD #disp9

JSR display\_string

JSR hexkeypad

LDAB sendhome

CMPB #1

BEQ homeJMP

LDAB switchchange

CMPB #1

BEQ jmptoswitchfile9

CMPA #$B

BEQ gen3

CMPA #$A ; if b is pressed, move arrow to gen 3, if f is pressed select gen 2, if a is pressed shift arrow to gen 1

BEQ gen1jmp

CMPA #$F

BEQ JUMPPICK

BRA gen2a

gen1jmp: JMP gen1

jmptoswitchfile9: JSR switch\_file

LDD #disp9

JSR display\_string

MOVB #0, waithold

MOVB #0, port\_s

BRA gen2a

gen3: LDAA #$3 ; store current generator into address

STAA $450

LDAA #'-' ; move arrow to gen3

STAA dispa

LDAA #'>'

STAA dispa+1

LDAA #' ' ; erase arrow from other areas

STAA dispa+16

STAA dispa+17

gen3a: LDD #dispa

JSR display\_string

JSR hexkeypad

LDAB sendhome

CMPB #1

BEQ JUMPHOME

LDAB switchchange

CMPB #1

BEQ jmptoswitchfile10

CMPA #$B

BEQ home

CMPA #$A ; if b is pressed, move arrow to home, if a is pressed move arrow to gen 2, if f is pressed, select gen 3

BEQ gen2jmp

CMPA #$F

BEQ GenPick

BRA gen3a

gen2jmp: JMP gen2

jmptoswitchfile10: JSR switch\_file

LDD #dispa

JSR display\_string

MOVB #0, waithold

MOVB #0, port\_s

BRA gen3a

home: LDAA #'-' ; shift arrow next to home

STAA dispa+16

LDAA #'>'

STAA dispa+17

LDAA #' '

STAA dispa ; erase arrow from other areas

STAA dispa+1

home2: LDD #dispa

JSR display\_string

JSR hexkeypad

LDAB sendhome

CMPB #1

BEQ JUMPHOME

LDAB switchchange

CMPB #1

BEQ jmptoswitchfile11

CMPA #$A

BEQ gen3jmp

CMPA #$B

BEQ JUMPGEN1

CMPA #$F ; if a is pressed, move arrow to gen 3, if f is pressed return home

BEQ JUMPHOME

BRA home2

JUMPGEN1: JMP gen1

JUMPHOME: JMP HOME

gen3jmp: JMP gen3

jmptoswitchfile11: JSR switch\_file

LDD #dispa

JSR display\_string

MOVB #0, waithold

MOVB #0, port\_s

BRA home2

GenPick: LDAA $450 ; display generator selected on LCD

ADDA #$30

STAA dispb+10

LDAA $450

CMPA #1 ; branch to generator selected

BEQ one

CMPA #2

BEQ two

LDAA gen3off

CMPA #0

BEQ GENON ; check if generator 3 is on or off

BRA GENOFF

two: LDAA gen2off

CMPA #0 ; check if generator 2 is on or off

BEQ GENON

BRA GENOFF

one: LDAA gen1off ; check if generator 1 is on or off

CMPA #0

BEQ GENON

BRA GENOFF

GENON: JMP genon

GENOFF: JMP genoff ; genon and genoff are too far and need a jump

autocheck: LDAA $450

CMPA #1

BEQ auto1 ; check which generator was selected

CMPB #2

BEQ auto2

CMPA #3

BEQ auto3

auto1: LDAA autoshut1

CMPA #0

BEQ autooffjmp ; check if generator 1 autoshutoff is on

BRA autoonjmp

autoonjmp: JMP autoon

autooffjmp: JMP autooff

auto2: LDAA autoshut2

CMPA #0 ; check if generator 2 autoshutoff is on

BEQ autooffjmp

BRA autoonjmp

auto3: LDAA autoshut3

CMPA #0

BEQ autooffjmp ; check if generator 3 autoshutoff is on

BRA autoonjmp

LEDstatus: LDAA $450

CMPA #1

BEQ LEDs1 ; check which generator selected

CMPA #2

BEQ LEDs2

CMPA #3

BEQ LEDs3

LEDs1: LDAA gen1cap

STAA port\_s ; load generator 1 capacity to LEDs

BRA waitforhome1

LEDs2: LDAA gen2cap

STAA port\_s ; load generator 2 capacity to LEDs

BRA waitforhome1

LEDs3: LDAA gen3cap

STAA port\_s

BRA waitforhome1 ; load generator 2 capacity to LEDs

waitforhome1: MOVB #0, pushpress

LDD #dispb ; display generator status on LCD

JSR display\_string

MOVB #1, LEDroutine

waitforhome: JSR hexkeypad

LDAB sendhome

CMPB #1

BEQ HOMEJMPER

LDAB pushpress ; check if push button pressed

CMPB #1

BEQ stepper1

anyways: LDAB switchchange

CMPB #1

BEQ jmptoswitchfile12

CMPA #$F ; branch to Generator selection when F is selected

BEQ JUMPHOME1

LDAA $450

CMPA #1

BEQ LEDs11 ; check which generator selected

CMPA #2

BEQ LEDs21

CMPA #3

BEQ LEDs31

HOMEJMPER: JMP HOME

stepper1: PSHX ; if push button pressed

PSHY

PSHD

JSR stepper

PULD

PULY

PULX

LDAA #$FE

MOVB #0, pushpress

LDAB #0

MOVB #0, switchchange

BRA anyways

LEDs11: LDAA gen1cap

STAA port\_s ; load generator 1 capacity to LEDs

BRA genpickjmp

LEDs21: LDAA gen2cap

STAA port\_s ; load generator 2 capacity to LEDs

BRA genpickjmp

LEDs31: LDAA gen3cap

STAA port\_s

BRA genpickjmp

jmptoswitchfile12: JSR switch\_file

LDD #dispb

JSR display\_string

MOVB #0, waithold

MOVB #0, port\_s

genpickjmp: JMP GenPick

JUMPHOME1: MOVB #0, port\_s

JMP GenSel

genoff: LDAA #'O'

STAA dispb+13

LDAA #'f'

STAA dispb+14 ;make LCD display off

LDAA #'f'

STAA dispb+15

JMP autocheck ;branch to check auto on/off

genon: LDAA #'O'

STAA dispb+14

LDAA #'n'

STAA dispb+15 ; make LCD display on

ldaa #' '

STAA dispb+13

JMP autocheck ; branch to check auto on/off

autoon LDAA #'O'

STAA dispb+30 ; make LCD display on

LDAA #'n' ; branch to LED status

STAA dispb+31

LDAA #' '

STAA dispb+29

BRA LEDstatusjmp

LEDstatusjmp: JMP LEDstatus

autooff: LDAA #'O'

STAA dispb+29

LDAA #'f'

STAA dispb+30 ;make LCD display on

LDAA #'f'

STAA dispb+31

JMP LEDstatus ; branch to led status

JMP HOME