XDEF initialize

XREF welcomehold

XREF counter, HOMEflg

XREF gen1cap

XREF gen2cap, delayont, delaytimert

XREF gen3cap, elevator

XREF interval25, g1a, g2a, g3a

XREF seconds, switchflg, gen2flg, gen3flg

XREF gen1timer, gen2timer, gen3timer, TRACKER, noTRACKER

XREF timercorrection, LEDroutine, setLEDTRACKER, switchstatus

XREF switchchange, prevswitchstatus, skipswitchcheck

XREF holdold, gen1au, gen2au, gen3au

XREF PWMCOUNTER, times8, locknoise, unlocknoise

XREF TON, sendhome, wait1, shutsound

XREF TOFF, pushpress, delaytimerm, delayonm

XREF delay1, timeswrong, instepper

XREF delayon, real\_power\_output, mu

XREF delaytimer, gen1off, gen2off, gen3off, ledswitches, switchchecker

XREF wait, waithold, power\_output, value, sum, real\_value

XREF gen2stat, gen3stat, gen1stat, SoundCounter, Alternator

XREF shutoff, potflg, priorvalue, startup, val1, val2, val3,

XREF time3off, time2off,time1off, letknow, clock1, autoshut1, autoshut2, autoshut3

; Initialize file requires referencing most important variables and files as it sets the original state ;of the power plant. This file is run twice during the start of the program and then is never used ;again.

initialize:

MOVB #$08, $242 ; Switch four is an output

MOVB #$FF, $24A ; initialize LEDs as output

MOVB #$FF, $25A ; Initialize Stepper motor as ouput

MOVB #$F0, $26A ; next three lines initialize hex keypad

MOVB #$F0, $26D

MOVB #$0F, $26C

MOVB #0, welcomehold ; initialize welcome hold variable

MOVB #0, counter ; counter starts at zero

MOVB #0, $403 ; counter is two bytes

MOVB #0, $405

MOVB #0, $406

MOVB #$17, $410

MOVB #$FF, gen1cap ; Generators start with max capacity indicated by hex value FF

MOVB #$FF, gen2cap ; Generators start with max capacity indicated by hex value FF

MOVB #$FF, gen3cap ; Generators start with max capacity indicated by hex value FF

MOVB #$0, autoshut1 ;auto shutoff timer initialized at zero until an auto shut off

;sequence begins

MOVB #$0, autoshut2

MOVB #$0, autoshut3

MOVB #$0, clock1

MOVB #$0, interval25

MOVB #$0, seconds ; variable keeps track of how many seconds have passed

MOVB #$8, switchflg

MOVB #$8, gen2flg

MOVB #$8, gen3flg

MOVB #0, gen1timer

MOVB #0, gen2timer

MOVB #0, gen3timer

MOVB #0, TRACKER

MOVB #1, noTRACKER

MOVB #42, timercorrection

MOVB #0, setLEDTRACKER

MOVB #0, LEDroutine

MOVB #$7, switchstatus ; indicates status of switches at any point in the program

MOVB #0, switchchange ; flag that lets program know a switch was changed

MOVB #$7, prevswitchstatus ; important to keep track of previous state of switches

MOVB #0, skipswitchcheck

MOVB #0, holdold

MOVB #0, PWMCOUNTER

MOVB #0, TON

MOVB #0, TOFF

MOVB #0, delay1

MOVB #0, delayon

MOVB #0, delaytimer

MOVB #0, gen1off

MOVB #0, gen2off

MOVB #0, gen3off

MOVB #0, ledswitches

MOVB #0, switchchecker

MOVB #0, wait

MOVB #0, waithold

MOVB #$FF, power\_output

MOVB #$FF, real\_power\_output

MOVB #0, value

MOVB #0, sum

MOVB #0, real\_value

MOVB #0, shutoff ;flag that indicates a shutdown has been attempted

MOVB #0, gen2stat

MOVB #0, gen1stat

MOVB #0, gen3stat

MOVB #0, potflg

MOVB #$FF, priorvalue

MOVB #0, startup

MOVB #0, val1 ;need this value to be 0 so higher nibble of register x is zero

MOVB #0, val2

MOVB #0, val3

MOVB #0, pushpress ;indicated push button has been pressed

MOVB #0, timeswrong

MOVB #$1E, $25A ;initialize stepper motor

MOVB #0, sendhome

MOVB #0, delaytimerm

MOVB #0, delayonm

MOVB #0, wait1

MOVB #0, instepper

MOVB #0, times8

MOVB #0, time3off

MOVB #0, time1off

MOVB #0, time2off

MOVB #0, g1a ;

MOVB #0, g2a ;

MOVB #0, g3a ;

MOVB #0, letknow1 ;

MOVB #0, gen1au ;

MOVB #0, gen2au ;

MOVB #0, gen3au ;

BSET $242, $20 ;

MOVB #0, SoundCounter ;

MOVB #0, Alternator ;

MOVB #0, locknoise ;

MOVB #0, unlocknoise ;

MOVB #0, elevator ;

MOVB #0, delaytimert ;

MOVB #0, delayont ;

MOVB #0, shutsound ;

MOVB #0, mu ;

MOVB #0, $48E ;

MOVB #0, HOMEflg ;

RTS ; return to main program