```
================================================================
LAB 0 PRELAB!
================================================================
PART 1
================================================================
Keep soldering iron around 350C (320-400) for lead.
Cold weld requires no heat and can be done with pressure between
     two of the same material.
Hot weld involves melting of at least one material.
If meant cold and hot soldering, cold is less clean and rough while
     hot is shiny and volcano like.
The term "black" refers to weld smut accumulating through the
     evaporation of aluminum and magnesium in the weld when
     temperatures are lower than welding level.
     In soldering, black is when soldering tip is oxidized and
     the solder won't stick.


================================================================
PART 4
================================================================
print serial port and intro;

flash 5 times
print instructions

while(1){

     if LeftRearBumper{
          flash 3 times
          turn left motor to 100, 80, 60, 40, 0 at set intervals
          have the left fwd led turn on until 0
          set leds initially to all on and decrease as motor slows eventually all
off
          the same for reverse.
          print left motor at 0
          print test complete
          flash 3 times
     }
     if RightRearBumper{
          flash 4 times
          similar to leftrearbumper but right side
          print right motor at 0
          print test complete
          flash 4 times

     }
     if LeftFrontBumper{
```

```
          flash once
          if(unplugged && switch on){
               depending on battery level, match by percentage on LED bar
          }
          print battery level
          print test complete
          flash once
     }
     if RightFrontBumper{
          flash twice
          read lightlevel
          match lightlevel to LED bar by percentage
          print light level;
          print test complete
          flash twice
     }

}

void flashLED(int howmany){
     for(i=0 i<howmany i++){
          LEDSET(ALL)
          wait a bit
          LEDCLR(ALL)
          wait a bit
     }
}
```

For testing, have keyboard inputs simulate bumper triggering

==========

PART 5

==========

Events to check:

MyEvents added to ES_Configure.h:

INTO_LIGHT

INTO_DARK

BATTERY_CONNECTED

BATTERY_DISCONNECTED

BUMP_ON

BUMP_OFF

---

have states for current and last battery level

create a macro for voltage threshold

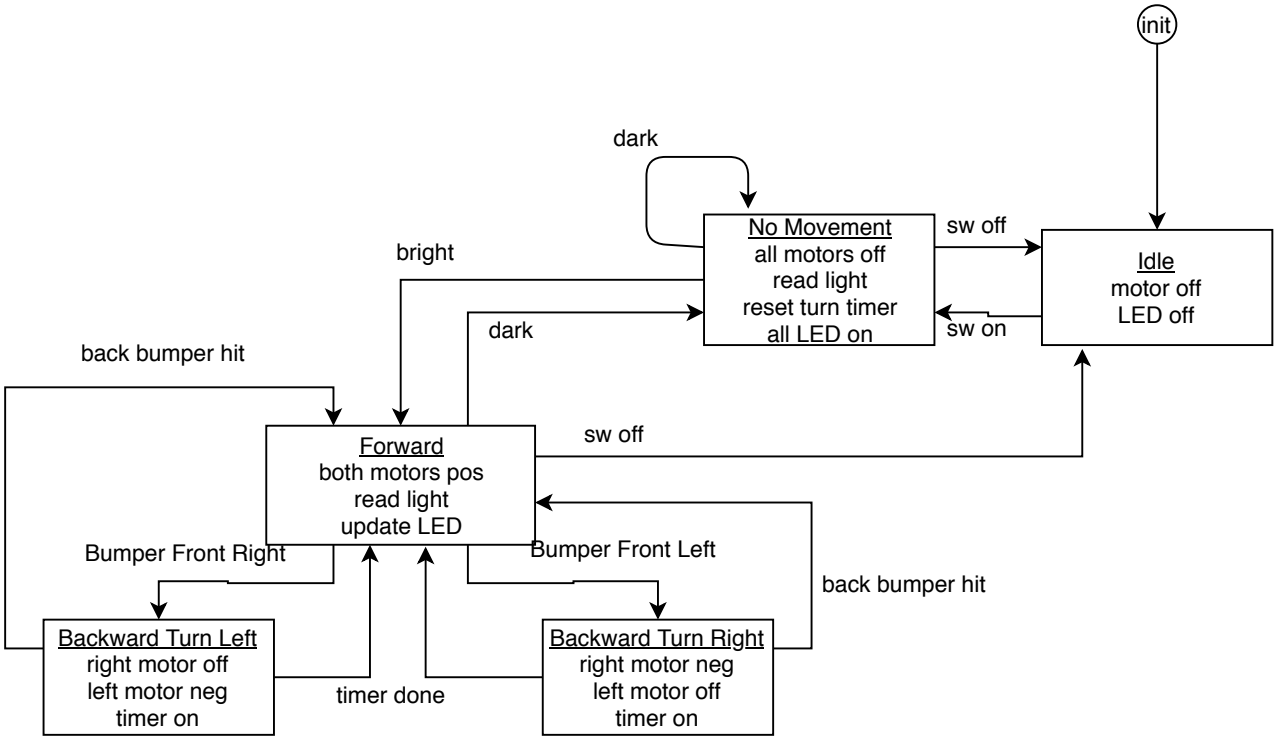
```
uint8_t TemplateCheckBattery(void){
     read voltage
```

```
     if(above threshold){
          curr is connected
     }
     else{
          curr is disconnected
     }
     if(curr != last){
          update event
          last = curr
          return true
     }
     return false;
}

//test : unplug and plug.

-----------------------------------------------------------------
create macros for dark and light threshold
create variables for the current and last light state

uint8_t CheckLight(){
     read lightlevel
     if(lightLevel > light threshold){
          curr = light
     }
     else if(lightlevel < dark threshold){
          curr = dark
     }
     else {
          stayed the same
     }
     if(curr != last){
          update event
          last = curr
          return true
     }
     return false
}

//test : input values from keyboard to simulate different light level

-----------------------------------------------------------------
static char lastBumperState = 0
uint8_t CheckCollision(){
     char currBumperState;

     call Roach_ReadBumpers() to see if any bumper are triggered.
     check if any of the states have changed
```

```
     if yes{
          update event
          set last = curr
          return true
     }
     return false
}

//test: physically press the bumpers on the roach to trigger event.

=================================================================
PART 6
=================================================================
//use the ES_Timeout from framework setting to 5ms (200hz)
//is basically an interrupt that will do the following

create static last state for all bumpers init to low or maybe all at once.
uint8_t CheckCollision(){
     create current state for all bumpers or maybe all at once
     //do for all bumpers the following if statements
     if(curr on and last off){
          update event
          last = curr
          return true
     }
     else if(curr off and last on){
          update event
          last = curr
          return true
     }
     return false
}

ES_EventTyp_t debounceBumper(bumper, last state){
     will handle debouncing of the bumpers.
}
//test: physically press the bumpers on the roach to trigger event
//      after 5 bumps exactly, leds turn on.

-----------------------------------------------------------------
For the light one, I already did the hysteresis in part 5
For the battery, don't think anything needs to be improved.

=================================================================
Part 7
=================================================================
States: idle, No Movement, Forward, Turn Right, Turn Left
```

```
Helper functions:
void RoachTurn(int time, int speed);    //time is how long to turn
                                        //pos speed is forward.
                                        //most likely drop speed and have constant
speed
void RoachPivot(direction dir)?

=====================================================================
Part 8
=====================================================================
States:
     idle
     No Movement
     Move
     Collision:
          Turn Left
          Turn Right

Will be using the same RoachTurn function.
Collision will have its own state machine.
```

INIT
init motors to off
init led?
init sensors?
set turn timer



Backwards
both motors neg
read light

Forward Turn Left
right motor pos
left motor off
timer on

Forward Turn Right
right motor off
left motor pos
timer on

INIT
init motors to off
init led?
init sensors?
set turn timer