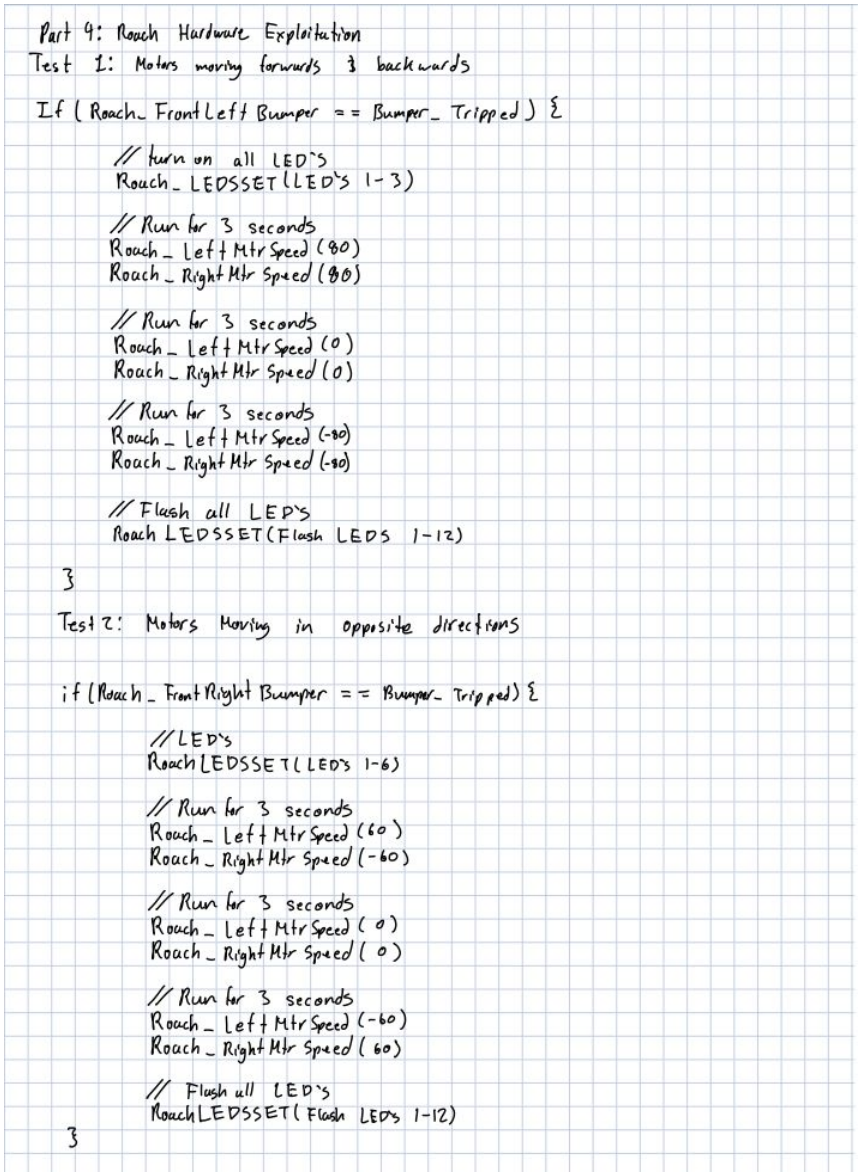
Manuel Lira
ECE 118/L
PreLab 0
9/29/2019

Part 1: PCB Assembly and Soldering

The iron should be set to a medium temperature ranging from 325 - 375 degrees celsius. A hot weld forms a volcano like form on the board whereas a cold weld will leave a blob like form on the desired board and tip. Black in the weld refers to the tip of the soldering iron that blackens over time due to oxidation and will not want to accept solder. One method to prevent this is by using a sponge to wipe off the build-up of solder on the tip.

Part 4: Roach Hardware Exploration



```
Part 4: Roach Hardware Exploitation
Test 1: Motors moving forwards & backwards

If ( Roach_FrontLeftBumper == Bumper_Tripped ) {

        // turn on all LED's
        Roach_LEDSSET(LED's 1-3)

        // Run for 3 seconds
        Roach_LeftMtrSpeed(80)
        Roach_RightMtrSpeed(80)

        // Run for 3 seconds
        Roach_LeftMtrSpeed(0)
        Roach_RightMtrSpeed(0)

        // Run for 3 seconds
        Roach_LeftMtrSpeed(-80)
        Roach_RightMtrSpeed(-80)

        // Flash all LED's
        RoachLEDSSET(Flash LEDs 1-12)
  }
  Test 2: Motors Moving in opposite directions

  if (Roach_FrontRightBumper == Bumper_Tripped) {

        //LED's
        RoachLEDSSET(LED's 1-6)

        // Run for 3 seconds
        Roach_LeftMtrSpeed(60)
        Roach_RightMtrSpeed(-60)

        // Run for 3 seconds
        Roach_LeftMtrSpeed(0)
        Roach_RightMtrSpeed(0)

        // Run for 3 seconds
        Roach_LeftMtrSpeed(-60)
        Roach_RightMtrSpeed(60)

        // Flash all LED's
        RoachLEDSSET(Flash LEDs 1-12)
  }
```

Part 4 continued...

Test 3: Battery Voltage

```
if(Roach_RearLeftBumper == Bumper_Triggered){

    //LED's
    RoachLEDSSET(LED's 1-9)

    currentButteryVoltage = Roach_ButteryVoltage();

    display("Buttery voltage is " + currentButteryVoltage)

    //flash all LEDS for 3 seconds
    RoachLEDSSET(Flash LEDS 1-9)

}
```

Test 4: Reading Light Levels

```
if(Roach_RearRightBumper == Bumper_Triggered){

    //LED's
    RoachLEDSSET(LED's 1-12)

    //display light levels reading @ a lower rate
    //display to user

    display("current Light Level: " + Roach_LightLevel)
    //... wait 3 seconds

    display("current Light Level: " + Roach_LightLevel)
    //... wait 3 seconds

    display("current Light Level: " + Roach_LightLevel)
    //... wait 3 seconds

    //flash all LEDS
    RoachLEDSSET(Flash LEDS 1-12)
}
```

Part 5: Event Detection

```
Part 5 : Event Detection

// Light Level Detection Constants
# define Dark_Threshold '88'
# define Light_Threshold '234'

type Event LightLevel (void) {

Ex.
'uint8_t'

currentLightValue = LightLevel(); // checks light level & stores it.

if (currentLightValue > Dark_Threshold) {

    currentLightState = Dark
    Roach LEDSSET (Turn on first LED)

}
if (currentLightValue < Light_Threshold) {
    currentLightState = LIGHT
    Roach LEDSSET (Turn on all LED'S)
}
if (currentLightState != lastLightState) { // event detected

    // Perform Event
    Roach LEDSSET (Flash All LEDS)

    return True
}
lastLightState = currentLightState
return False
}
```

Part 5 continued...

```
// Detect Bumper Event

type Event Bumper (void) {

    currentRoachBumper = Roach_ReadBumpers()

    if(currentRoachBumper & 0x01) { // Front Left Bumper Hit
        currentBumperState = HIT_FRONTLEFTBUMPER
        RoachLEDsSet(FlashLED1) // a test
    }
    if(currentRoachBumper & 0x02) {
        currentBumperState = HIT_FRONTRIGHTBUMPER
        RoachLEDsSet(FlashLED2) // a test
    }
    if(currentRoachBumper & 0x03) {
        currentBumperState = HIT_REARLEFTBUMPER
        RoachLEDsSet(FlashLED3) // a test
    }
    if(currentRoachBumper & 0x04) {
        currentBumperState = HIT_REARRIGHTBUMPER
        RoachLEDsSet(FlashLED4) // a test
    }

    if(currentRoachBumper != lastRoachBumper) { //event detected

        RoachLEDsSet(FlashallLEDs)
        return True
    }

    lastBumperState = currentBumperState
    return False

}
```

The modifications to the ES_Configure.h file include: adding user defined events, adding the name of the event checking function header file, adding the event checking functions, defining the timer, and initializing the service.

Part 6: Better Event Detection

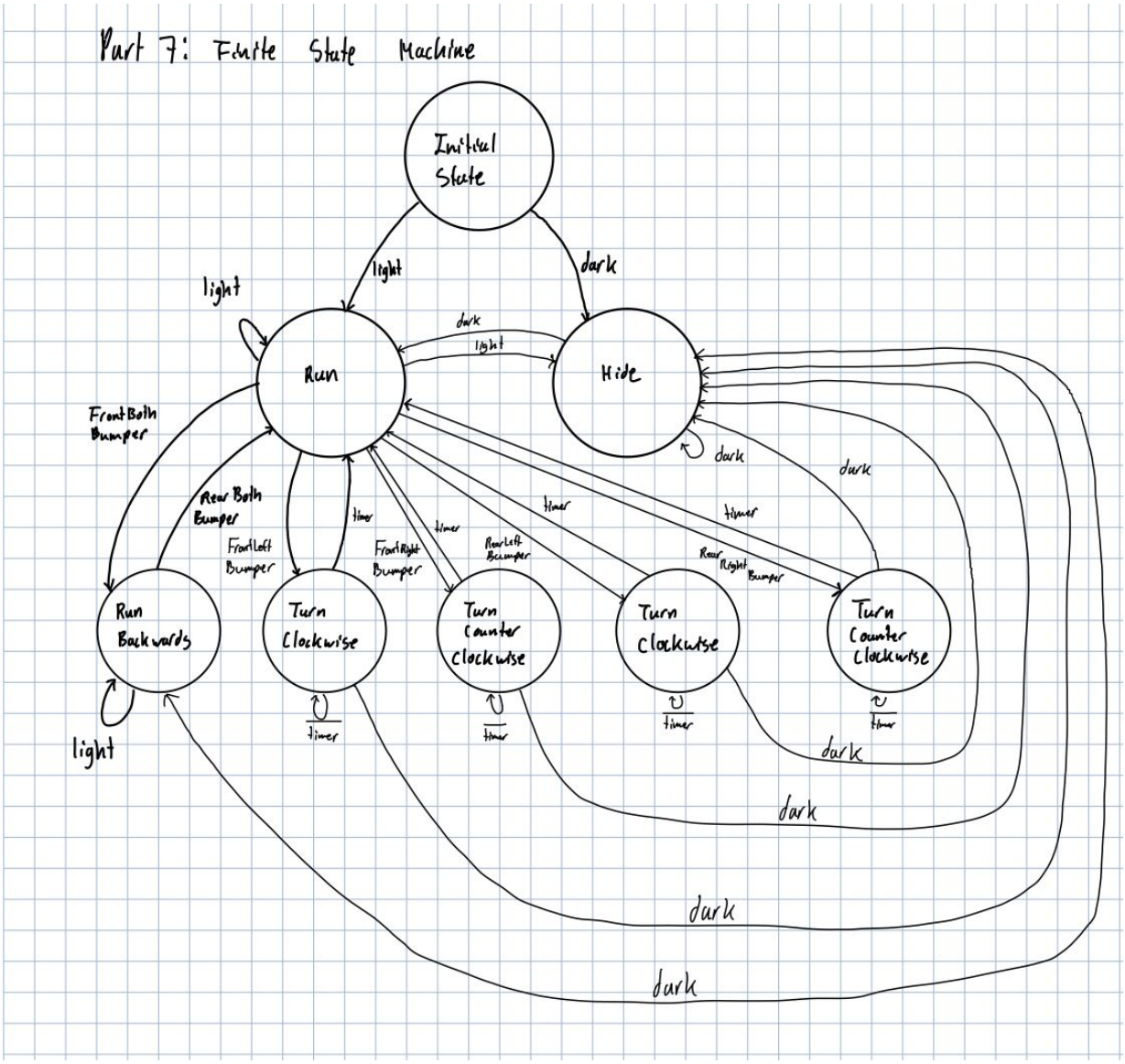
```
Part 6: Better Event Detection

Editing Part 5....

type Event Bumper (void) {

    currentRoachBumper = Roach_ReadBumpers()

    if((currentRoachBumper & 0x01) { // Front Left Bumper Hit
        currentBumperState = HIT_FRONTLEFTBUMPER
        RoachLEDSSET(Flash LED1) // a test
        bumper_bounce1 = currentBumperState // new line
    }
    if((currentRoachBumper & 0x02) {
        currentBumperState = HIT_FRONTRIGHTBUMPER
        RoachLEDSSET(Flash LED2) // a test
        bumper_bounce1 = currentBumperState // new line
    }
    if((currentRoachBumper & 0x03) {
        currentBumperState = HIT_REARLEFTBUMPER
        RoachLEDSSET(Flash LED3) // a test
        bumper_bounce1 = currentBumperState // new line
    }
    if((currentRoachBumper & 0x04) {
        currentBumperState = HIT_REARRIGHTBUMPER
        RoachLEDSSET(Flash LED4) // a test
        bumper_bounce1 = currentBumperState // new line
    }

    if(currentRoachBumper != lastRoachBumper) { //event detected

        RoachLEDSSET(Flash all LEDS)
        return True
    }

    lastBumperState = currentBumperState
    return False

    // here we continue for debouncing

    if( bumper_bounce1 == bumper_bounce2)

        if(bumper_bounce2 == bumper_bounce3)

            if(bumper_bounce3 == bumper_bounce4)
                return bumperbounce1 // an event


            else if(bumper_bounce3 != bumper_bounce4)
                bumper_bounce1 = none
                return bumperbounce1

        else if(bumper_bounce2 != bumper_bounce3)
                bumberbounce1 = none


    else if(bumperbounce1 != bumperbounce2)
        bumper_bounce1 = none
        return bumperbounce4

    bumperbounce4 = bumperbounce3
    bumperbounce3 = bumperbounce2
    bumperbounce2 = bumperbounce1

}
```

```
// Hysterisis

enum State1 { none, dark, light};
enum State2 { none, dark, light};

    State1 = LightLevel()
    State2 = none
    userLight = Light level()

    if (userLight < Light Threshold)

        State 2 = Light

    if (userLight > Dark Threshold)

        State 2 = Dark


    if (user Light > Light Threshold && user Light < Dark Threshold)
        return State 1


    else if( State 1 == State 2)

        return  State 2

    else if ( State 1 != State 2)

        return State 2

        State 1 = State 2
```

```
// Initialize  timer & reset each time

    Initialize   ES_TIMEOUT to 0.

    if( ES_TIMEOUT == 5ms) {

        call  Event Light Level  function
        call  Event Bumper  function

        Reset  ES_TIMEOUT  to  0.

        Service function calls itself

    }
```

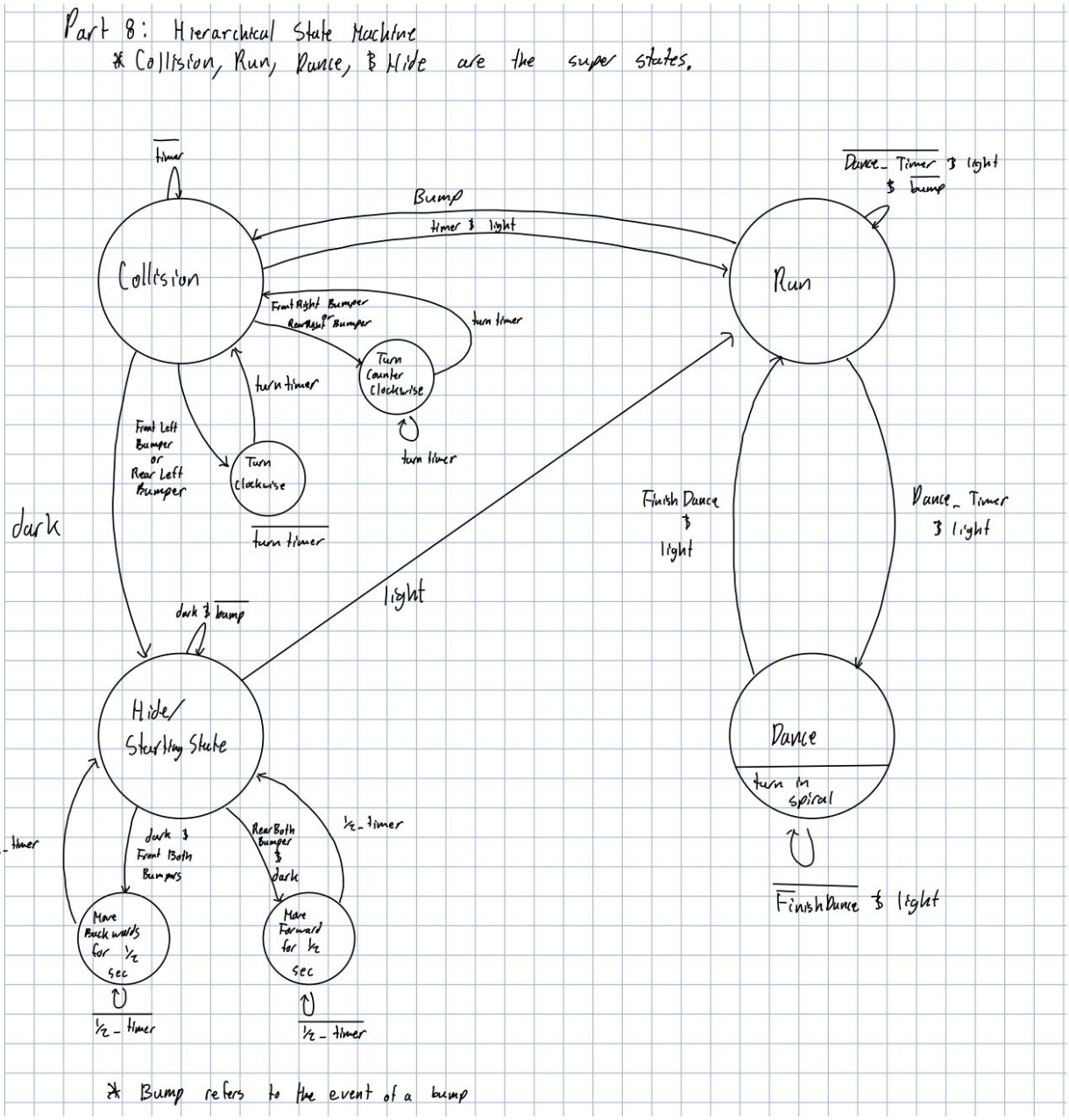
The modifications to the ES_Configure.h file will remain the same as in part 5.

Part 7: Finite State Machine



We will create various helper functions for the different timers in the state machine. Furthermore, using functions for the counterclockwise and clockwise turning will also make our life easier.

Part 8: Hierarchical State Machine



CHECKOFF AND TIME TRACKING

Student Name: Manuel Lira CruzID mlira2 @ucsc.edu

Time Spent out of Lab	Time Spent in Lab	Lab Part - Description
		Part 1 – PCB Assembly and Soldering
		Part 2 – "Hello World!" on a Roach
		Part 3 – Running the Roach Test Harness
		Part 4 – Roach Hardware Exploration
		Part 5 – Event Detection
		Part 6 – Better Event Detection
		Part 7 – Finite State Machine (FSM)
		Part 8 – Hierarchical State Machine (HSM)

Checkoff: TA/Tutor Initials	Lab Part - Description
Tony Li	PreLab – Preparation for the Roach Lab
	Part 1 – PCB Assembly and Soldering
	Part 4 – Roach Hardware Exploration
	Part 5 – Event Detection
	Part 6 – Better Event Detection
	Part 7 – Finite State Machine (FSM)
	Part 8 – Hierarchical State Machine (HSM)