```
//*****************************************************************************************************
//*****************************************************************************************************

int packVoltage(){
    int Voltage = analogRead(PwrIn);
    delay(ADC_DELAY);
    if(Voltage < minPackVoltage) {
       while(1){
         //Infinite Loop Halting all other processes other than indicating low voltage (LED Blinks)
         digitalWrite(LEDPwr, HIGH);
         delay(500);
         digitalWrite(LEDPwr, LOW);
         delay(500);
       }
    }
}

//*****************************************************************************************************
//*****************************************************************************************************

void differential(int Steering, int throttle){
  //UNFINISHED, NEED TO WORK OUT EQUATIONS FOR Ri AND THEN WORK OUT GOOD CODE FOR DIFFERENTIAL
  //THESE EQUATIONS ARE EXTREMELY RUDIMENTARY AND NEED TUNING BUT THEY SHOULD "WORK"...
    double IV, OV, Ri;
    double idealDiff;
    double steerDifference = Steering - steeringMidpt;
    double controlApplied;
    double rpmDifference;

    if(((steerDifference >= 0) && (steerLast <= 0)) || ((steerDifference <= 0) && (steerLast >= 0))){
       control = 0;  //This is the "passed through zero" check and resets control to accomodate that
    }

    if(steerDifference < leftSteerBuffer){
     Ri = minTurnRadiusLeft*((steeringMidpt - steeringMin)/(steeringMidpt - Steering));
     if((RPM_1 == 0)){
       rpmDifference = 0;
     } else {
       rpmDifference = ((RPM_1) - (RPM_0))/(RPM_1);  //Outer wheel minus inner wheel
     }
    } else if(steerDifference > rightSteerBuffer){
     Ri = minTurnRadiusRight*((steeringMax - steeringMidpt)/(steeringMax - Steering));
     rpmDifference = ((RPM_0) - (RPM_1))/(RPM_0);  //Outer wheel minus inner wheel
     if((RPM_0 == 0)){
       rpmDifference = 0;
     } else {
       rpmDifference = ((RPM_0) - (RPM_1))/(RPM_0);  //Outer wheel minus inner wheel
     }
    } else {
     Ri = 35000; // This value is "infinity" because the radius of curvature for a straight line is infinity
    }

    #if DEBUG
        Serial.print("rpmDifference = ");
        Serial.println(rpmDifference);
        Serial.print("Ri = ");
        Serial.println(Ri);
    #endif
```