Source: “stepper.c”

```
        // update steps to speed and steps to stop calculations
        // only calculate when the motor is idle and commanded to move
        time = (float) (DesiredTargetSpeed - StartingSpeed) / AccelRate;
        time_squared = time * time;
        // calculate steps to target
        Steps2Target = ((AccelRate >> 1) * (time_squared))
                + (StartingSpeed * time);
        // calculate Steps2Stop if:
        //  StepsToStop in GUI is >1 and
        //  StoppingSpeed is not equal Starting Speed
        //
        if ((StepsToStop == 2 && use_one_edge == 1) || StepsToStop == 1)
        {
            if (StartingSpeed != StoppingSpeed)
            {   // recalculate time if necessary
                time = (float) (DesiredTargetSpeed - StoppingSpeed) / AccelRate;
                time_squared = time * time;
            }
            // calculate steps required to stop
            Steps2Stop = ((AccelRate >> 1) * (time_squared))
                    + (StartingSpeed * time);
        }
        else    // defined number of steps to stop
        {
            Steps2Stop = StepsToStop;
        }
        // calculate stepping rate update (integer value divide by 32)
        if (AccelRate > 31 || AccelRate == 0) // if >=32 or 0
        {
            SteppingRateUpdate = AccelRate >> 5; // divide by 32 by right shifting 5
positions
        }
        else
        {
            SteppingRateUpdate = 1; // set it to 1 to get correct value
        }

        if (StartingSpeed >= DesiredTargetSpeed)
        // Special case -- override prior settings
        // allow motor to attempt to run only at starting speed
        {
            SteppingRateUpdate = 0;
            Steps2Target = 0;
            Steps2Stop = 0;
        }
        // transition from ACCEL to DECEL when count equals(StepsToMove - Steps2Stop) <=0
        if ((Steps2Stop > (StepsToMove >> 1))
                && (StartingSpeed == StoppingSpeed))
            Steps2Decel = StepsToMove >> 1; // decel at midpoint
        else
            Steps2Decel = StepsToMove - Steps2Stop;
```