3.2.2 Pseudo-Code

```
Control “LeftLDR” by the microcontroller                          ▷These statements basically mean that while designing the actual code, all the
Control “RightLDR” by the microcontroller                           pins connected and controlled by the microcontroller would be defined and
Control “Solenoid” by the microcontroller                           configured
Control “Enable” pin of the motor driver by the microcontroller
Control “step” pin of the motor driver through microcontroller
Control “dir” pin of the motor driver by the microcontroller
int LeftSensorValue ← 0                                           ▷ADC value stored from the left sensor
int RightSensorValue ← 0                                          ▷ADC value stored from the right sensor
boolean MotorMode ← true                                          ▷Basically orders the motor to power up or shut down
int ShadePosition ← 0                                             ▷Shade set vertically initially as a reference position (in degrees)
const int minLimit ← *                                            ▷Lower boundary limit for shade position. The value is in degrees
const int maxLimit ← *                                            ▷Upper boundary limit for shade position. The value is in degrees
long initialTime ← Read time from computer                        ▷They are meant for time tracking so that the rest mode can be defined
long presentTime ← 0                                                appropriately by reading time from the computer
const int tolerance ← *                                           ▷Basically defines the level of accuracy under which the sensors work
const int MotorRotationalSpeed ← *                                ▷Defines the speed of rotation of the stepper motor
const int Step ← *                                                ▷Defines the no. of steps the motor should rotate in every single revolution of the motor
Solenoid ← Set the solenoid output low                            ▷Initially solenoid locks the system for security; to keep shade in position
int dir ← *                                                       ▷Defines the direction of rotation of the motor according to the control structure evaluation
```

▷At the device start-up, the solenoid is set to low as well as the motor is turned on in order to be completely sure that the shade remains fixed & locked in its position with the desired amount of holding torque needed from the motor as well as keeping it still mechanically.

```
while true do                                                  ▷Forever Loop; runs as soon as system switched on & powered
LeftSensorValue ← LeftLDR                                      ▷Microcontroller reads & stores ADC value from left sensor
RightSensorValue ← RightLDR                                    ▷ Microcontroller reads & stores ADC value from right sensor
MotorMode ← false
Enable ← MotorMode                                             ▷Motor disabled/shutdown to save power
presentTime ← Read time from computer every iteration          ▷Reading current time to check further for rest mode
    if (presentTime – initialTime < 10 minutes) then           ▷Check for rest mode
        Solenoid ← Set the solenoid output low                 ▷Locks system with solenoid; shade fixed in position
    else
        MotorMode ← true
        Enable ← MotorMode                                     ▷Motor enabled/powered up
Label: A                                                       ▷A label to redirect the code for reiteration according to the need
LeftSensorValue ← LeftLDR                                      ▷Microcontroller reads & stores ADC value from left sensor
RightSensorValue ← RightLDR                                    ▷Microcontroller reads & stores ADC value from right sensor
            if (ShadePosition > minLimit && ShadePosition < maxLimit)   then     ▷Condition to check for physical shade position
                if (LeftSensorValue > (RightSensorValue + tolerance))  then      ▷Condition;Check for difference b/w the photodiode values
                    Solenoid ← Set the solenoid output high                      ▷Unlocking system via solenoid
                    Rotate motor clockwise with defined speed and steps          ▷Motor rotation for shade compensation
                    Solenoid ← Set the solenoid output low                       ▷Locking the system via solenoid
                    Shade position calculated* and updated to the “ShadePosition” variable
                    Go to Label: A                                               ▷To check until the shade has been compensated
                else if (RightSensorValue > (LeftSensorValue + tolerance)) then
                    Solenoid ← Set the solenoid output high                      ▷Unlocking system via solenoid
                    Rotate motor counter clockwise with defined speed and steps  ▷Motor rotation for shade compensation
                    Solenoid ← Set the solenoid output low                       ▷Locking the system via solenoid
                    Shade position calculated* and updated to the “ShadePosition” variable
                    Go to Label: A                                               ▷To check until the shade has been compensated
                else if ((LeftSensorValue – RightSensorValue) <= tolerance or equal to zero) then
                    MotorMode ← false
```

```
48:                 Enable ← MotorMode                            ▷Motor disabled/shutdown
49:                 Solenoid ← Set the solenoid output low        ▷Locking the system via solenoid
50:             else                                              ▷For line no. 34
51:             end if                                            ▷For line no. 34
52:         else                                                  ▷For line no. 33
53:             Solenoid ← Set the solenoid output low            ▷Locking the system via solenoid
54:             MotorMode ← false
55:             Enable ← MotorMode                                ▷Motor disabled/shutdown
56:         end if                                                ▷For line no. 33
57: initialTime ← Read time from the computer every iteration of the loop   ▷Initial time updated for the next execution of the loop
58:     end if                                                    ▷For line no. 25
59: end while                                                     ▷For line no. 19
```