PROJECT: BLACK AND WHITE

- Requirements and analysis of the problem
  1. Let us assume that the AGV moves on a perfect straight line throughout the barcode.
  2. The AGV moves in a perpendicular direction to the barcode lines.
  3. The color sensor returns 1 when it detects white and 0 when it detects black.
  4. The speed at which the AVG moves is also a factor that affects the quality of information acquired. The clock frequency of the ATMega-328p is 16Mhz which should be a sufficient sampling rate for an AVG because the usual speed of such vehicles are intended to be very low, maybe around 5Kmph to 10Kmph, say.
  5. The software will repeatedly check for the sensor's value and count the number of blacks and white.
  6. Let us consider that each bar in the barcode has a width.
  7. Let us assume that there are no distortions in the color pattern.
  8. Our goal is to find the number of black and white bars in the barcode.
- Formulating solution
  1. Let there be two integer variables each that holds the number of black and white colors detected, respectively. Variables: nWhite, nBlack of type int.
  2. Let the AVG be initially placed at one end of the barcode such that its direction of motion will be perpendicular to the color bars of the barcode.
  3. Let us consider the termination condition (i.e., when the AVG should stop) to be when the color sensor detects neither of the color. Let's add a new feature to the sensor which is such that the sensor returns -1 if neither black nor white is detected.
  4. From the previous point we can also consider that the AVG starts counting colors when it detects either black or white. It must anyway consider an end of the barcode if it detects some other color anywhere.

5. Since each bar has a certain width, the software should be prevented to read the same bar multiple times during sampling. The motive is – one bar one sample. To prevent this issue, we make use of two Boolean variables ‘inWhite’ and ‘inBlack’ which indicates whether the color sensor is currently above the white or black bar region, respectively.
6. Let us consider that the color sensor provides an analog value through its output pin. So the Arduino has to read the sensor’s value in the range of 0 – 1023 from its analog pin. We separately define the algorithm for this function also.

- Algorithm

```
Vars: nWhite, nBlack of type Integer.

BlackNWhite(): Returns two integers
      nWhite = nBlack = 0
      inWhite = inBlack = false
      while True:
            sensorVal = getColor()
            if inBlack == false AND sensorVal == 0:
                  nBlack = nBlack + 1
                  inBlack = true;
                  inWhite = false;
            else if inWhite == false AND sensorVal == 1:
                  nWhite = nWhite + 1
                  inWhite = true;
                  inBlack = false;
            else if sensorVal == -1:
                  break
      return nWhite, nBlack

getColor(): Returns an integer
      readVal = analogRead(sensor_pin)
      if readVal == 0
            return 0
      else if readVal == 1023
```

- ## Implementation Details

1. **Development Tools Used**
    - Arduino IDE
2. **Hardware Used for testing the code**
    - Arduino pro mini
3. **Operating System**
    - Windows 10 x64

- ## Coding Approach

The code has been designed to work in such a way that if the barcode operation is done the program will not terminate and bring the CPU to halt state. Rather it will continue to look for another barcode sequence whenever detected. Whenever a white or a black color is detected the black-and-white algorithm starts and terminates upon detecting color other than black or white.