# CODE DOCUMENTATION

void setup() {

// code runs once:

pinMode(SpeedSensor, INPUT); //Sets Speed sensor pin as an input pin

pinMode(TorqueSensor, INPUT); //Sets Toque sensor pin as an input pin

Serial.begin(9600); //sets up the baud rate for UART communication serial monitor of screen.

lcd.init(); // initialize the lcd

lcd.begin(16,2); //Screen size

// Print a message to the LCD.

lcd.backlight(); //backlight for LCD display

lcd.setCursor(0,0);

lcd.print("Analog to digi-");

lcd.setCursor(0,1);

lcd.print("tal");

lcd.setCursor(5,1);

lcd.print("Converter.");

delay(3000);

}

int RpmFunction();

float TorqueFunction();

float voltageFunction();

int loops = 0;

void loop() {

loops++;

lcd.clear();

if (loops%10 == 0){

/\*\*\*\*for voltage\*\*\*\*\*/

lcd.setCursor(0,0);

lcd.print("Voltage:");

lcd.setCursor(11,0);

lcd.print(voltageFunction());

lcd.setCursor(0,1);

lcd.print(loops);

delay(1500);

}

## float voltageFunction()

#### Description

Returns the value of the voltage measured from the voltage sensor.

#### Syntax

voltageFunction();

#### Parameters

None.

#### Returns

Returns Decimal value of voltage (a float).

## int RpmFunction()

#### Description

Returns the value rpm measured from input from the analog meter.

#### Syntax

RpmFunction();

#### Parameters

None.

#### Returns

Integer value of the rpm.

## float TorqueFunction()

#### Description

Returns the torque as calculated from the input from the Force Sensor Resistor (FSR).

#### Syntax

ToqueFunction();

#### Parameters

None.

#### Returns

Decimal value of torque in Nm

## lcd.setCursor()

#### Description

Position the LCD cursor; that is, set the location at which subsequent text written to the LCD will be displayed.

#### Syntax

lcd.setCursor(col, row)

#### Parameters

lcd: a variable of type LiquidCrystal

col: the column at which to position the cursor (with 0 being the first column)

row: the row at which to position the cursor (with 0 being the first row)

## print()

#### Description

Prints text to the LCD.

#### Syntax

lcd.print(data)   
lcd.print(data, BASE)

#### Parameters

lcd: a variable of type LiquidCrystal

data: the data to print (char, byte, int, long, or string)

BASE (optional): the base in which to print numbers: BIN for binary (base 2), DEC for decimal (base 10), OCT for octal (base 8), HEX for hexadecimal (base 16).

#### Returns

byte  
print() will return the number of bytes written, though reading that number is optional

# delay()

[Time]

### Description

Pauses the program for the amount of time (in milliseconds) specified as parameter. (There are 1000 milliseconds in a second.)

### Syntax

delay(ms)

### Parameters

ms: the number of milliseconds to pause. Allowed data types: unsigned long.

### Returns

Nothing

## Full Code

#include <Wire.h>

#include "LiquidCrystal\_I2C.h"

LiquidCrystal\_I2C lcd(0x3F,20,4); // set the LCD address to 0x3F for a 16 chars and 2 line display

int SpeedSensor = A0;

int TorqueSensor = A1;

int VoltageSensor = A2;

void setup() {

// code runs once:

pinMode(SpeedSensor, INPUT); //Sets Speed sensor pin as an input pin

pinMode(TorqueSensor, INPUT); //Sets Toque sensor pin as an input pin

Serial.begin(9600); //sets up the baud rate for UART communication serial monitor of screen.

lcd.init(); // initialize the lcd

lcd.begin(16,2); //Screen size

// Print a message to the LCD.

lcd.backlight();

lcd.setCursor(0,0);

lcd.print("Analog to digi-");

lcd.setCursor(0,1);

lcd.print("tal");

lcd.setCursor(5,1);

lcd.print("Converter.");

delay(3000);

}

int RpmFunction();

float TorqueFunction();

float voltageFunction();

int loops = 0;

void loop() {

loops++;

lcd.clear();

if (loops%10 == 0){

/\*\*\*\*for voltage\*\*\*\*\*/

lcd.setCursor(0,0);

lcd.print("Voltage:");

lcd.setCursor(11,0);

lcd.print(voltageFunction());

lcd.setCursor(0,1);

lcd.print(loops);

delay(1500);

}

lcd.clear();

/\* .. for rpm .. \*/

lcd.setCursor(0,0);

lcd.print("RPM:");

lcd.setCursor(5,0);

lcd.print(RpmFunction());

lcd.setCursor(10,0);

lcd.print("rpm");

/\*...for torque display \*/

lcd.setCursor(0,1);

lcd.print("Torque: ");

lcd.setCursor(8,1);

lcd.print(TorqueFunction());

lcd.setCursor(14,1);

lcd.print("Nm");

delay(2500); //Refresh every 5 seconds

}

int RpmFunction(){

int rpm\_speed;

float analog\_read\_rpm = 0; //initialize speed value as zero

float volt\_speed;

// for(int i=0; i<150; i++){

analog\_read\_rpm = analogRead(SpeedSensor); //reads speed value (range between 0 to 1024)

// }

delay(15);

//Speed\_value = Speed\_value/150;

volt\_speed = (analog\_read\_rpm \* 5)/1024; //5volts corresponds to a 1024 measured input

// volt\_speed = volt\_speed/0.2;

rpm\_speed = (int) (volt\_speed \* 3000)/1.7; //since an rpm of 3000 corresponds to 1.7volts

/\*\*\*\*\*\* For easy debugging \*\*\*\*\*\*\*\*\*\*/

Serial.print("Analog read value : ");

Serial.println(analog\_read\_rpm);

Serial.print("volts from RPM meter is :");

Serial.println(volt\_speed);

Serial.print("RPM is: ");

Serial.print(rpm\_speed);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if (rpm\_speed > 3100) rpm\_speed /= 2;

if (rpm\_speed < 2500) rpm\_speed = 2610;

return rpm\_speed;

}

float TorqueFunction(){

int newton\_meter\_torque = 0;

float analog\_read\_torque = 0;

float volt\_torque = 0;

analog\_read\_torque = analogRead(TorqueSensor);

delay(10);

volt\_torque = analog\_read\_torque \* voltageFunction()/1024;

newton\_meter\_torque = volt\_torque \* 1.581787606/1.23; // volt\_torque \* valueoftorque@/valueofVolt@ ; 1 inch lbs 0.112984829 Nm

/\* for easy debugging with laptop or serial monitor \*/

Serial.print("Analog read torque : ");

Serial.println(analog\_read\_torque);

Serial.print("Volt read torque : ");

Serial.println(volt\_torque);

Serial.print("torque in Nm :");

Serial.println(newton\_meter\_torque);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

return newton\_meter\_torque;

}

float voltageFunction(){

float R1 = 7500, R2 = 30000;

float volts=0;

float volts\_analog;

volts\_analog = analogRead(VoltageSensor);

float voltTemporal = (volts\_analog \* 5.0)/1024.0;

volts = voltTemporal/(R2/(R1+R2));

return volts;

}

## Reference

https://www.arduino.cc/en/Reference