**IIoT test bed using Maixduino**

The main idea of this project is to develop a test bed for Industrial Internet of Things to monitor parameters like temperature, humidity, light level, DC motor RPM and its power consumption. Sipeed Maixduino board act as the processing unit to measure, process and display the parameters in the LCD. Further the data is transferred to a host PC to feed the data to Adafruit IO platform.

**Project Description:**

This prototype could be used as a IIoT test bed for data collection, processing and to upload data to cloud platform. DC motor RPM and its power consumption measurement is vital in predicting faults and analyzing the motor life in industrial applications. In this prototype 30 slot rotating disc mechanism and holder for DC motor is developed using 3D printing. Used any cubic Kobra2 neo printer and PETG material for making the mechanism. The mechanism holds the motor as well as the opto interrupter as shown in Fig 1 below.

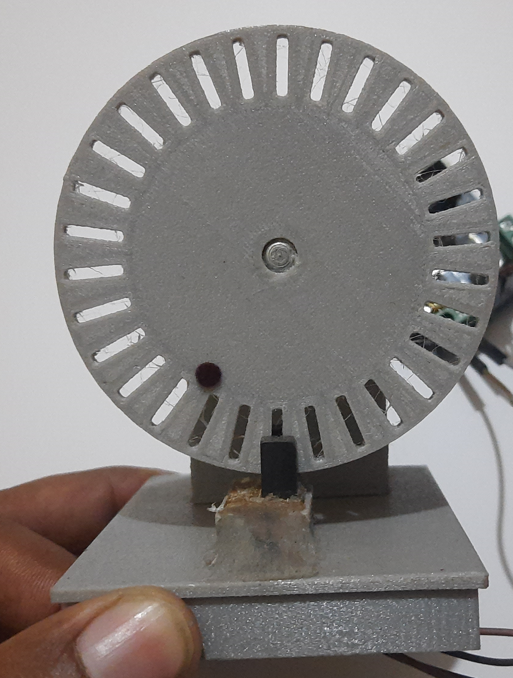
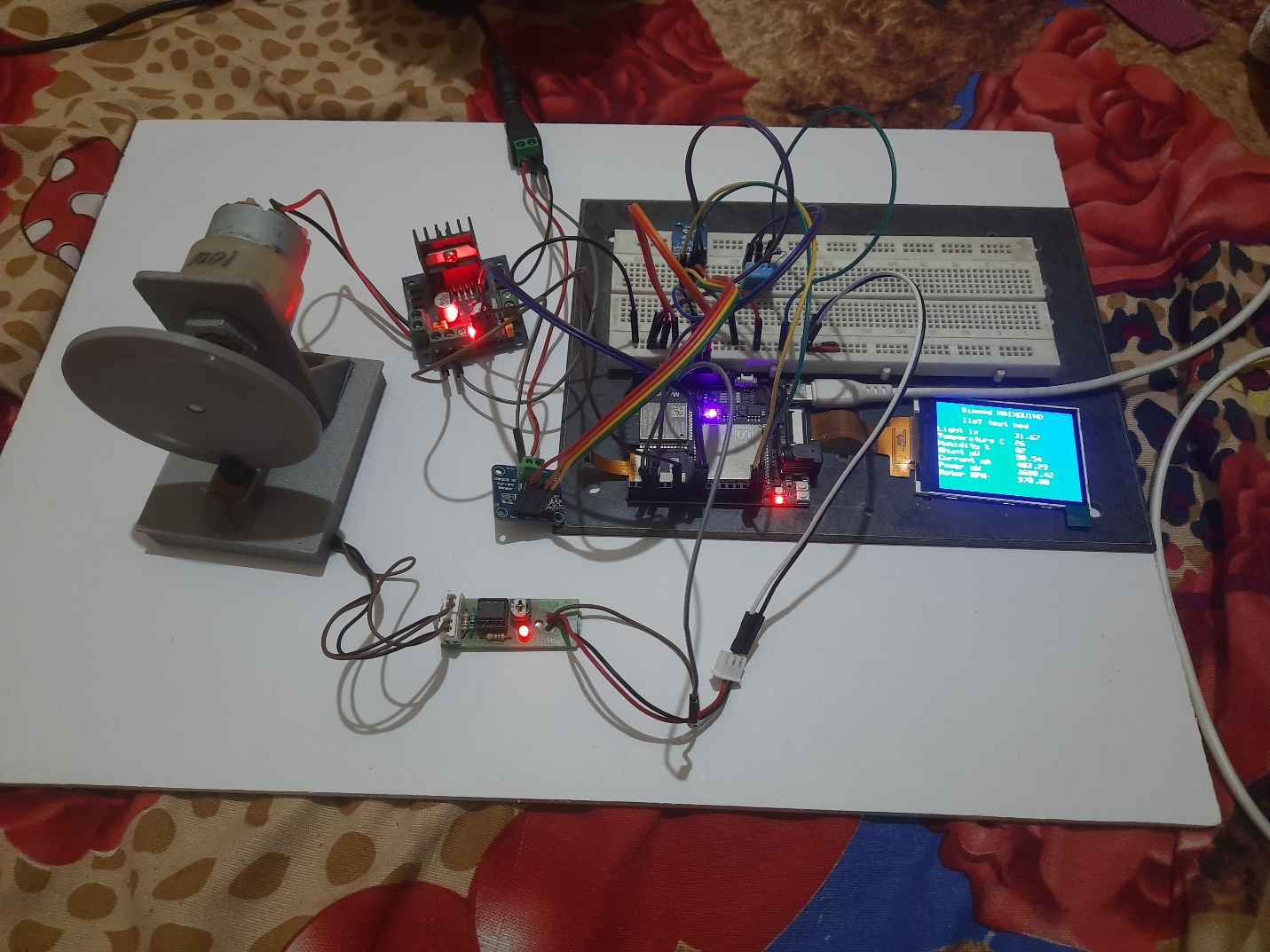


Fig 1: 3D printed 30 slot encoder disc and motor holding mechanism.

Whenever the DC motor is enabled with defined duty cycle, 30 slot encoder disc mechanism mounter with the DC motor starts rotating, further the opto-interrupter starts to detect the number of ticks per slot detected and increments the counter value up to 30 counts in this setup since the encoder disc has 30 slots. One complete rotation takes 30 count value. Based on the slot counted for defined interval motor speed is computed. Along with this current and power consumption of the motor also measured using the INA 219 current sensor module. DHT11 sensor is used to measure the Ambient temperature and relative humidity of the test environment. Light sensor BH1750 is used to measure the light intensity. Opto interrupter sensor is used to detect the disc rotation to compute the motor speed. Below Fig2 shows the working setup of the developed system.



**BH1750**

**DHT11**

**Maixduino**

**INA219**

**1000 RPM DC motor**

**L298N motor driver**

**3D Printed 30 slot rotating mechanism with motor holder.**

**LCD Display**

Fig 2: Developed system working setup.

**Components Used:**

1. Sipeed Maixduino Kit for RISC-V AI + IoT
2. DHT 11 temperature and humidity sensor
3. BH1750 Light sensor
4. 1000 RPM DC motor
5. L298N dual H-Bridge motor driver
6. Opto-interrupter sensor or IR LED sensor module
7. INA219 current sensor module
8. 30 slot rotary encoder disc mechanism and arrangement to hold the motor.
9. 12 V 5 A DC power adopter for DC motor

**Circuit Description:**

BH1750 and INA219 sensor modules are I2C compatible and their SDA and SCL lines are connected to the Maixduino SDA and SCL pins respectively for data communication. DHT11 sensor supports onewire interface and it’s connected to PIN3 of Maixduino. Opto interrupter sensor output is connected to the PIN2 of Maixduino. L298N H-Bridge motor driver is used to control the speed of 1000 RPM DC motor. ENA pin of motor driver is connected to PIN6, Motor input IN1 and IN2 are connected to PIN8 and PIN9 of Maixduino. DC motor connected with OUT1 terminals of motor driver module. To measure the current and power consumption of DC motor, motor drive module is power with a 12V 5A DC power adopter through the INA219 module. Circuit diagram of the developed system is shown in Fig 3. Maixduino detects and computes the motor speed, current and power consumed, temperature and relative humidity and light level. Further the computed data are transferred to the host PC via serial communication and onboard LCD display of Maixduino.

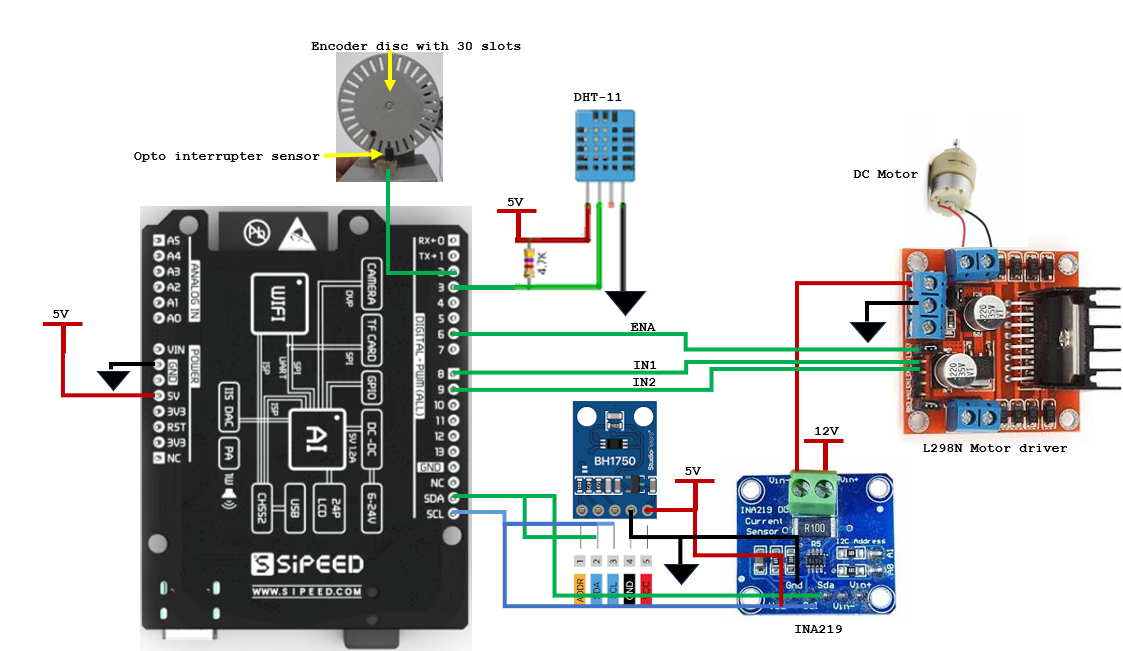


Fig 3: circuit diagram of the developed system

**Software description:**

Arduino IDE is used for the code development. The code utilizes wire.h, for I2C communication and BH1750.h for reading data from BH1750 sensor, INA219.h for reading data from current sensor module. DHT11.h is used for reading temperature and humidity data. Sipeed\_ST7789.h is used for the display utilization.

Below are the few functions created:

1. title() – This function, initializes the display and print project title on the display
2. motorinit() – This function, initializes the motor settings (ENA to control speed, IN1 & IN2 for direction and motor control.
3. motoroff() – This function turns off the motor
4. motoronforward() – This function, moves the motor in forward direction
5. motoronbackward() – This function, moves the motor in backward direction
6. RPM\_speed(int a) – This function, get the duty cycle value for the motor speed
7. readlight() – This function, reads the light value and update the value display and send it to serial port
8. readrpm() – This function computes the RPM and update the value in display and send it to serial port
9. readINA() – This function reads the current and power consumption data and update the value in display and send it to serial port
10. temphumi() – This function reads the temperature and humidity values and update the values in display and send it to serial port

the code initializes required settings in the void setup() and void loop() has the function calls to read the sensor data and update the display and transfer the data over serial communication to host PC.

**Results:**

Below Fig 4 & Fig 5 shows the display of measurement parameters during Motor OFF and Motor ON state. Fig 6 shows the measurement data collected on the serial port of the host PC.



Fig 4: Measurement data on display during Motor OFF



Fig 5: Measurement data on display during Motor ON

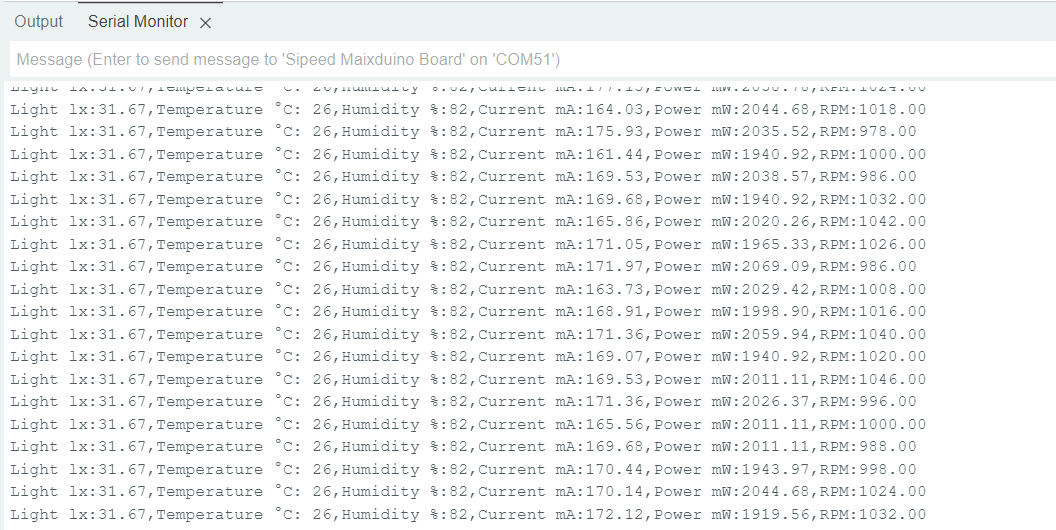


Fig 6A: Measurement data collected on the serial port of the host PC

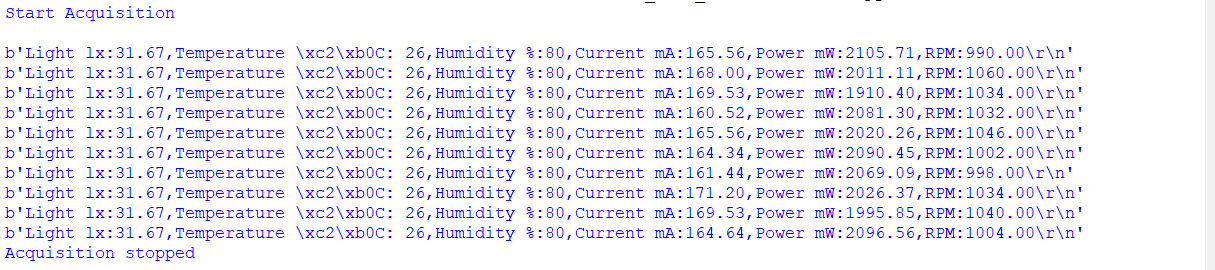


Fig 6B: measurement of data with python script to read the serial data on the host PC

**Future enhancement:**

At its present status, the developed system is not having IIoT enablement. In future, host PC can forward data or from the Maixduino through Wi-Fi communication can send data to Adafruit IO or similar cloud platforms to have the IoT connectivity, through the IoT platform further motor speed could be controlled and monitored for motor fault analysis and prediction of motor life which enables the complete IIoT system. In connection to this work, dashboard is created with Adafruit IO for the IIoT enablement as shown in Fig 7.

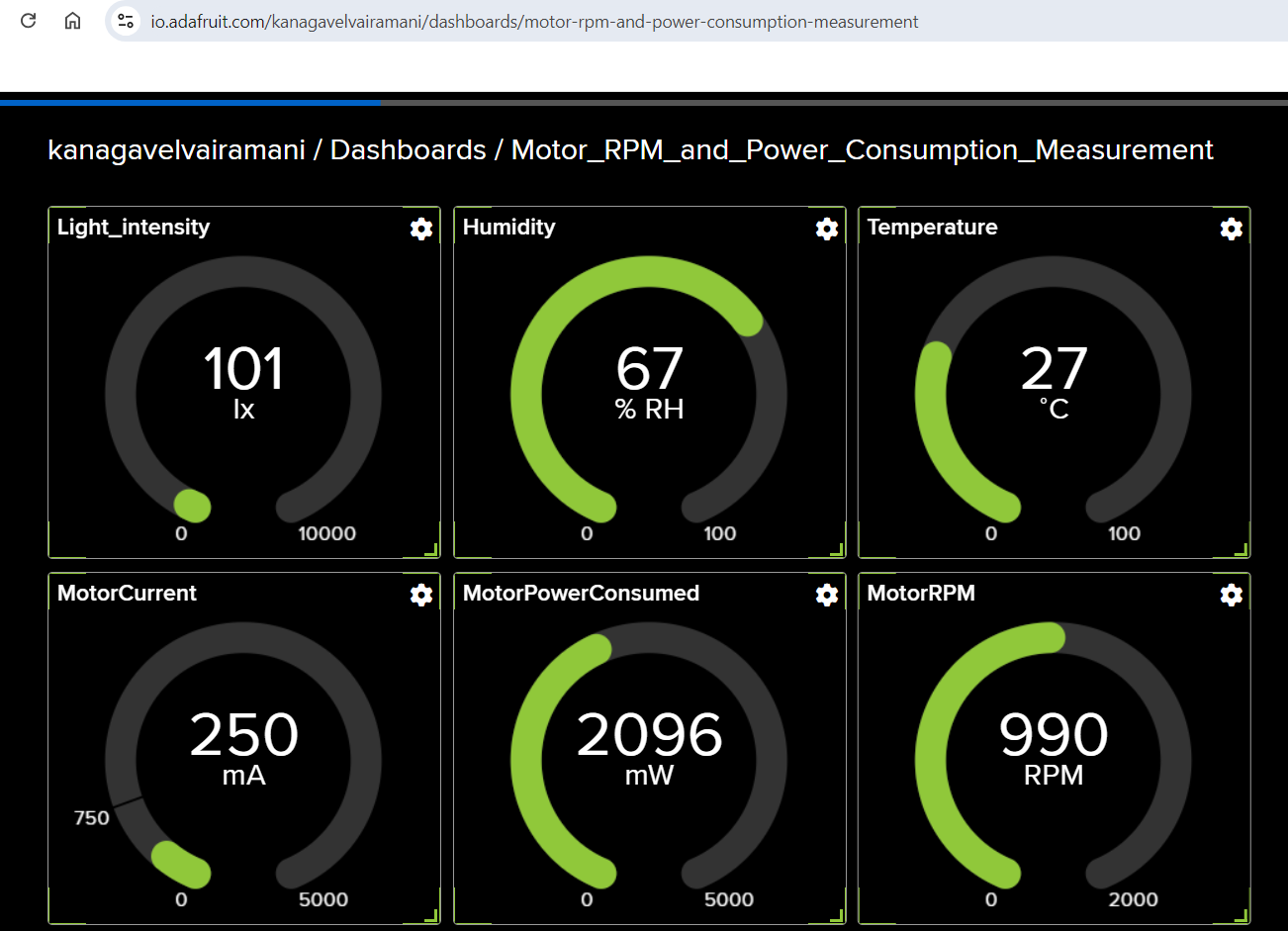


Fig 7: Adafruit IO dashboard view for the developed system

**Code link:**

<https://github.com/kvairamani/Maixduino_Kit/blob/main/Sipeed_Light_RPM_Current_LCD.ino>

**Adafruit IO dash board link:**

<https://io.adafruit.com/kanagavelvairamani/dashboards/motor-rpm-and-power-consumption-measurement>

**Video link:**

<https://youtu.be/nZbs_3EOoJU?si=JXn-GHbo1dQ8fONA>