RACECAR DASHBOARD

GitHub: https://github.com/vishnukumar10/HallEffectRPM

YouTube: https://youtu.be/9_FRK3s-XQQ

Budget: Procured from Ajanta Electronics, Ritchie St.,

Hall Effect Sensor and Magnet – Rs. 110.

ESP8266 – Rs. 340

Jumper Cables and Bread Board – Rs. 200

Total – Rs. 650.

Steps and Connection Specification:

The Hall Effect sensor uses the magnitude of a magnetic field to measure proximity sensing, positioning, speed detection, and current sensing applications. It is placed at a stationary point in the wheel of the car and the magnet is placed on one of the spokes of the wheel on the inner side which rotates.

Make the required connections. Every time the magnetic sensor comes near the sensor, the sensor takes in the input by measuring the magnitude and sends it to ESP8266 board.

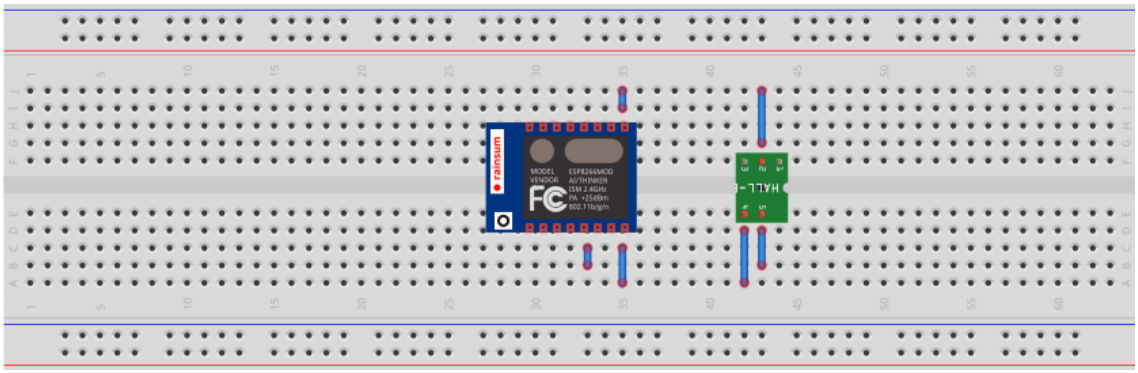
The board which is connected to the Wi-Fi with the help of the User name and AIO key sends the data to the specific feed that is mentioned in the code. That is, it publishes the data onto the dashboard and the feed.

This data can be viewed real time on the dashboard and the feed.

[Pin 1] The GND connects to the GND on the board.

[Pin 2] The Source pin (S) connects to the Voltage supply in the board.

[Pin 3] And the V.out is connected to the D2 on board.



Code:

```
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"

volatile byte counter;

unsigned int rpm;
unsigned int v =0;

unsigned long passedtime;


#define WLAN_SSID       "B8wz-VkhWSw"
#define WLAN_PASS       "vhvk_1999"

/************************* Adafruit.io Setup *********************************/

#define AIO_SERVER      "io.adafruit.com"
#define AIO_SERVERPORT  1883                   // use 8883 for SSL
#define AIO_USERNAME    "vhvk_99"
#define AIO_KEY         "548586a16cc646329a637a9e7c382e6b"
WiFiClient client;
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT,
AIO_USERNAME, AIO_KEY);

Adafruit_MQTT_Publish Rpm = Adafruit_MQTT_Publish(&mqtt,
AIO_USERNAME "/feeds/rpm");
Adafruit_MQTT_Publish Velocity = Adafruit_MQTT_Publish(&mqtt,
AIO_USERNAME "/feeds/velocity");
// Setup feeds subscribing to changes.

/*************************** Sketch Code ************************************/

// Bug workaround for Arduino 1.6.6, it seems to need a function declaration
// for some reason (only affects ESP8266, likely an arduino-builder bug).
void MQTT_connect();

void isr()

 {

   //Each rotation, this interrupt function is run twice, so take that into consideration
for
   //calculating RPM
   //Update count

      counter++;
```

```
}
void setup()

{
 Serial.begin(9600);
 Serial.println(); Serial.println();
 Serial.print("Connecting to ");
 Serial.println(WLAN_SSID);

 WiFi.begin(WLAN_SSID, WLAN_PASS);
 while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
 }
 Serial.println();

 Serial.println("WiFi connected");
 Serial.println("IP address: "); Serial.println(WiFi.localIP());

  //Intiates Serial communications

  attachInterrupt(0, isr, RISING); //Interrupts are called on Rise of Input


  counter = 0;

  rpm = 0;

  passedtime = 0; //Initialise the values

}
uint32_t x=0;

void loop()
{
   MQTT_connect();
    Serial.print(F("\nSending RPM and Velocity Values "));


  delay(5000);//Update RPM every second

  detachInterrupt(0); //Interrupts are disabled

  rpm = 30*1000/(millis() - passedtime)*counter;

  passedtime = millis();

  counter = 0;

  /* Velocity = RPM * 3.14 / Diameter of the Wheel
```

```
 Assuming the Diameter of the wheel is 0.75m and conversion for

 m/s to km/h */

v = rpm * 3.14 * 3.6 / 75;

 Serial.print("RPM=");

 Serial.println(rpm);
  Serial.print("Velocity=");

Serial.println(v); //Print out result to monitor
  if (! Rpm.publish(rpm)) {
  Serial.println(F("Failed"));
} else {
  Serial.println(F("OK!"));
}
if (! Velocity.publish(v)) {
  Serial.println(F("Failed"));
} else {
  Serial.println(F("OK!"));
}//Print out result to monitor
 attachInterrupt(0, isr, RISING);   //Restart the interrupt processing

}

void MQTT_connect() {
int8_t ret;

// Stop if already connected.
if (mqtt.connected()) {
  return;
}

Serial.print("Connecting to MQTT... ");

uint8_t retries = 3;
while ((ret = mqtt.connect()) != 0) { // connect will return 0 for connected
     Serial.println(mqtt.connectErrorString(ret));
     Serial.println("Retrying MQTT connection in 5 seconds...");
     mqtt.disconnect();
     delay(5000);  // wait 5 seconds
     retries--;
     if (retries == 0) {
       // basically die and wait for WDT to reset me
       while (1);
     }
}
Serial.println("MQTT Connected!");
}
```