Remote Industry Appliance Management System (RIAMS)

Submitted to:

Nahyan Al Mahmud
Assistant Professor, Dept. of EEE
Ahsanullah University of Science and Technology

Date of Submission:
16 October, 2019

Submitted by:

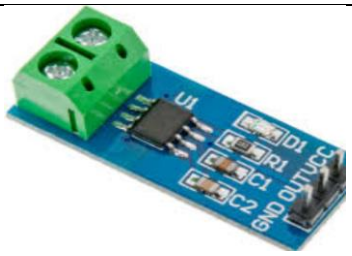
Objectives: To make a system to control industry appliances like motor, generator using a web tool that will help the authority to control the industry remotely. Besides, the system is able to fetch data from the environment and report to the authority.

Apparatus: The tools are divided into two parts. Software and Hardware. They are listed below:

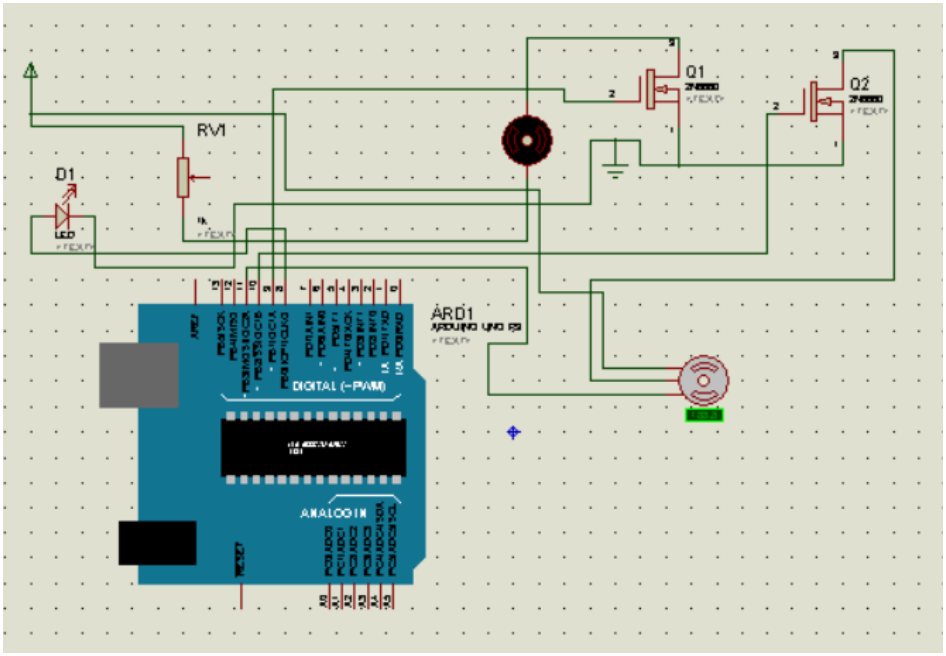
1. **Software tools:**
    a. Adafruit IO
    b. Python

2. **Hardware tools:**
    a. DC motor
    b. Servo motor
    c. Hall sensor
    d. Thermistor
    e. MOSFET (n type)
    f. Registor
    g. LED
    h. Arduino
    i. Raspberry Pi

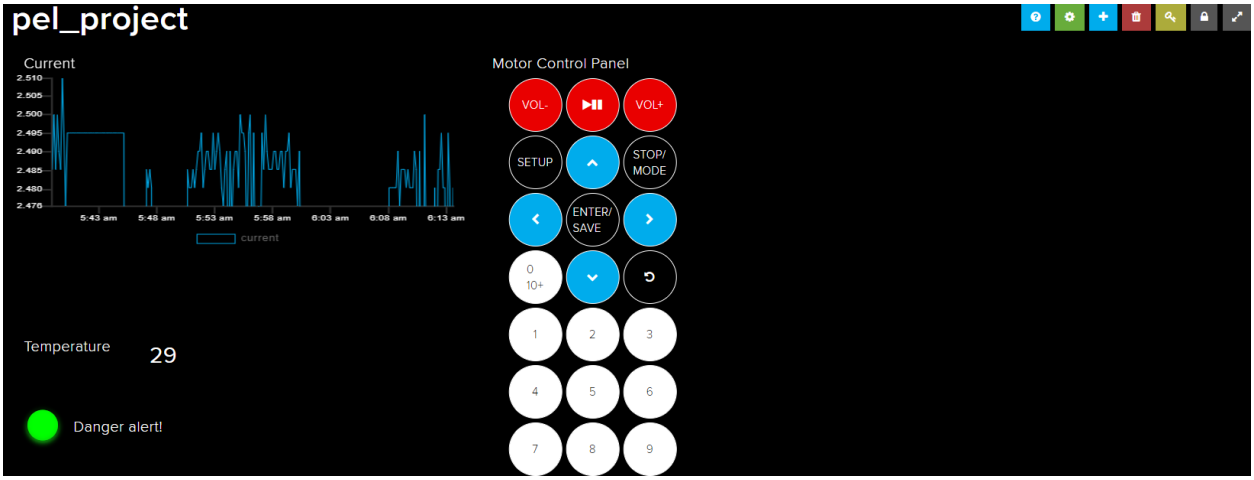
Major Tools:

Raspberry Pi: It is a development board especially made for IoT application. ARM Cortex A72 processor is used in raspberry pi. A built in WiFi module is installed to that it can be controlled using internet wirelessly. 4 USB port, Eathernet port, 3.5mm sound output port makes it a computer and it is totally mobile. 40 IO pins are also given to be used in controlling application.	
Arduino: Arduino Uno is a development board based on Atmega 328. It has several IO pins to generate and receive signals. ADC is a great feature of Arduino. It is used to control various devices and take data from sensors.	
Hall Sensor: It is used to measure current flowing through a circuit. Based on theory of Hall Voltage, it generates voltage in case of increment of current. In this project we have used ACS712	
Servo Motor: The degree of rotation of servo motor can be controlled using PWM signal. Each motor has a specific rated voltage and torque.	
Adafruid IO: Adafruit IO is a platform to control raspberry pi online without any read IP. Using login id, we can access pi from anywhere of the world using internet. A web interface is provided by them to make controlling easier	

Schematic Design:



Interface:

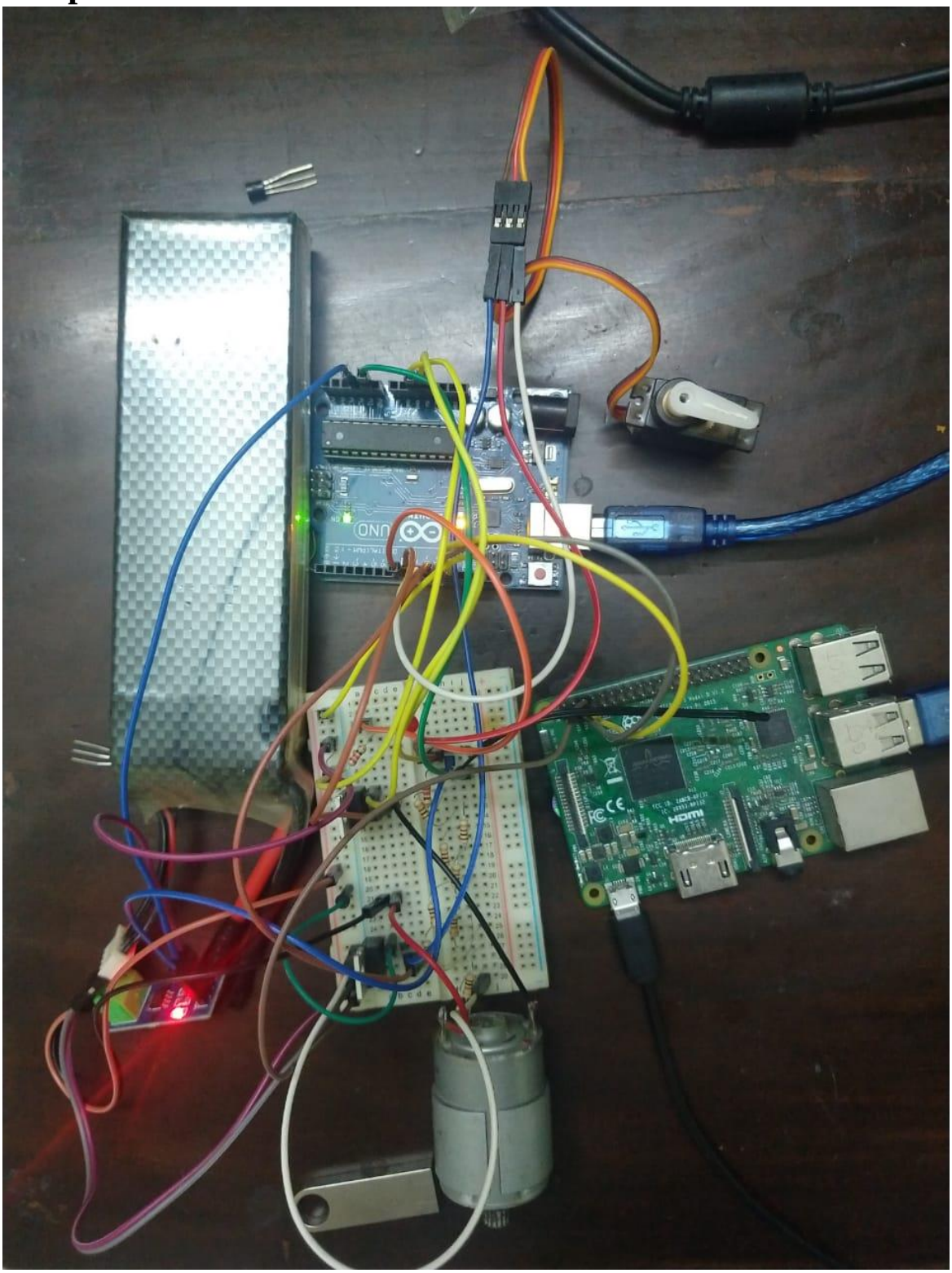


Working Principle:

There are two parts of this project. First we discuss the hardware part. In this part DC motor and a servo motor is connected to Arduino as per schematic. There is a MOSFET series with DC motor and Servo motor which controls the motor's status. If we want to stop the motor we'd put 0 in the gate of the MOSFET. A current sensor is used to measure the power consumption of the circuit. A thermistor is connected with Arduino to measure the temperature of the motor. If there is a rise such that the motor can burn, the MOSFET will turn off and the circuit will be cooled down. Same for the servo motor.

In the software part, we have used adafruit IO to control raspberry pi using internet. Normally it requires real ip to access pi from internet but using arafruit.io we can access using a hash code without real IP. Raspberry pi sends control signal to Arduino. And for reading the data from Arduino, we have used a json file which contains temperature and current. In the web interface, we'll see the information in graph. If there is an increase of temperature above threshold, the system will turn off and it will take time to make itself cool.

Setup:



Shortcomings:

1. The current sensor has 30A highest rating and the current in our circuit setup is too much lower than that. So the fluctuations may not be visible always.
2. We have used the free version of Adafruit IO. So limited IO pins could be used.
3. We have used the thermistor using resistance which is a power hungry process. Using digital temperature sensor like DTH11 could save the power.
4. Raspberry pi is costly. We could use ESP8266 in this purpose to save costs.

Further Development:

1. An android app can be made to control the system from android
2. More plausible temperature sensor can be used to make it more practical and efficient.
3. Using FPGA rather than Arduino could provide parallel operation rather than serial operation. But it would cost more.
4. Without using adafruit IO, the interface could be designed using php. It would be cost effective and more flexible.

Conclusion:

Though our setup has some shortcomings, it works perfectly as a demo of a management system. A person can not only control the devices but also can fetch real time sensor values in this setup. And as this system is accessible by internet, one can easily control stuff from very far. Also it can be used in localhost but it would limit it s usability. For using in high voltage, it would be needed powerful relay, and other power electronic devices to run securely.