Int. Conf. on Signal, Image Processing Communication & Automation, ICSIPCA

Reduction of Power Consumption and Data Transmission on to a Webpage with a Comparative Study on Arduino Uno and NodeMCU

Srinidhi.D [1], R.Sowmya[2], Sahana S.M[3], Samyukta Sudhindra[4] and Mrs. Shubha B[5]
[1-4]B.E Student, Dept. of ECE, JSS Academy of Technical Education, Bangalore, Karnataka, India
Email: nidhi.thula@gmail.com, Sowmims95@gmail.com, sahanasm16@gmail.com, samyuktastar007@gmail.com
[5]Asst.Professor, Dept. of ECE, JSS Academy of Technical Education, Bangalore, Karnataka, India
Email: sriniketana@yahoo.co.in

***Abstract*— This paper aims at reducing the power consumed by the sensors, to ensure that the battery life comes for a longer duration which is an advantage for smart city applications.Apart from this we also aim to bring in a comparative study between Arduino Uno and esp8266 12e Node MCU. We transmit the data for the sampled duration and make the microprocessor in Arduino Uno go to sleep for every 8 seconds. The same is carried out in Node MCU as Arduino Uno, but here since Node MCU already has an inbuilt sleep mode and transmits data only when active, the execution of sleep mode for every 8seconds is not required. Through Node MCU, the data obtained from the sensors are sent to the web server. This comparative study is drawn out in order to determine which gives a more holistic approach for power reduction and can be used for IOT applications.**

***Index Terms*— sensors, Arduino Uno, Node MCU, wireless node, system flow.**

I. INTRODUCTION

Today, internet of things (IOT) is becoming very popular where each and every application involves electronic devices getting connected to each other, which is very important for the IOT system to be realized. The smart sensors always pose energy reduction challenge because in a wireless environment we expect the sensors to send the real time data continuously, but these sensors work on batteries, hence battery changing remains a difficult task especially if the sensors are placed in remote areas. Thus, power reduction of the batteries must be taken care so that the shelf life of the battery increases. In a wireless IOT space each sensor node collects information by sensing its surrounding and then transfers the information to a sink via wireless transmission. Different from other battery-powered apparatus, recharging a sensors battery is generally an impossible task. Although solar and wind energy can be used, but such energy supplies are not reliable. Therefore, network lifetime is a key issue in wireless sensor nodes. Depletion of these finite energy batteries can result in a change in network topology or end of network life itself. Hence, prolonging the life of wireless sensor networks via nodes of these network is important. In order to make these networks a reality, the node hardware and implementation should be optimized for three characteristics. 1.Low cost: The utility of the networks depends on high density of nodes. Inorder to make large scale deployments economically feasible, nodes must be of very low cost.

2. Small size: For the same reasons, the size of modules must be of small size so that the network is

Grenze ID: 02.ICSIPCA.2017.1.62

unobtrusive.
3. Low power: For large networks with many nodes, battery replacement is very difficult, expensive or even impossible. Nodes must have efficient energy so that it can function for long periods without running out of power.
Hence in this paper we are trying to establish methods to achieve low power consumption and bring in real time experience through an evolving domain, IOT which can give the users a better way for data acquisition and transmission on a real time basis.

II. SYSTEM SPECIFICATION

Hardware

- Arduino Uno
- esp8266 Node MCU
- simple labs sound sensor
- DHT 11 – Temperature and humidity sensor
- DS1307-RTC,for the display of date and time
- LED
- PC or battery can be used for power

A. Arduino Uno

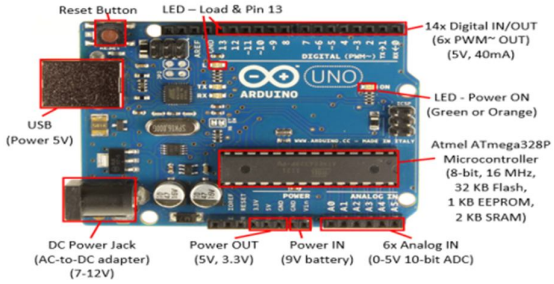

FIG.1. Pin Description Of Arduino Uno

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically [2]. The Arduino Uno is a microcontroller board based on the ATmega328, which is shown in FIG.1. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by power jack. Leads from a battery can be inserted in the Ground andVin pin headers of the POWER connector. The various pins of Arduino can be seen in the FIG.1. The features are shown in Table 1.

TABLE 1: FEATURES OF ARDUINO UNO

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12 V
Digital i/o	Pins 14(of which 6 provide PWM output)
Analog input pins	6
SRAM	2KB(ATmega328)
EEPROM	1KB (ATmega328)

ESP8266 Node MCU

The features of Node MCU is listed below:
1)32 bit RISC CPU, 80MHz
2)64 kb of RAM instruction
3)96 kb of data RAM
4)4MB Flash, CP2102 USB
5)IEEE 802.11 b/g/n Wi-Fi

6)16 GPIO PINS which are not 5V tolerant, 1 10 bit ADC
7) UART on dedicated pins, plus a transmit-only UART can be enabled on GPIO2
8) Operating voltage : 3.3V

FIG.2

Sensors used. Here we use two sensors which are DHT11 and sound sensor. DHT11senses the humidity and the temperature and sound sensor gives the output voltage and the nature of sound in the surrounding environment. Any sensors can be used according to the requirement.
DS1307-Real Time Clock. As the name suggests, RTC is a real-time clock that keeps track of the current time. Here this sensor is used as an additional feature to note the date and time of data transmission.

III. GENERAL SYSTEM OVERVIEW

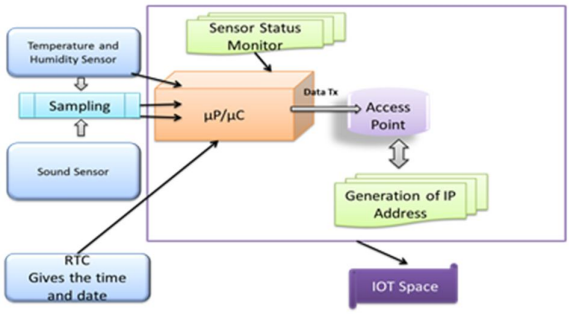

Fig.3. In Fig µp: Microprocessor, µc: Microcontroller

This system aims at gathering data from the sensors through the microprocessor or microcontroller which is subjected to sampling for power consumption reduction. The data obtained is then transmitted over to an access point which plays a pivotal role in getting the required IP address. The IP address is generated as the microcontroller or microprocessor has TCP/IP stack. This would enable the data obtained from the microprocessor to be transmitted over to a web server.

IV. SYSTEM FLOW

The typical sensor operation can be seen from Figure [2] and the details of the flow diagram is as mentioned below :
Initialization:
Once the microprocessor or microcontroller is turned on, it authenticates and associates itself with a predetermined access point (AP) and acquires an IP address.
Keep-alive messages:
Depending on the implementation, an access point (AP) may remove a device

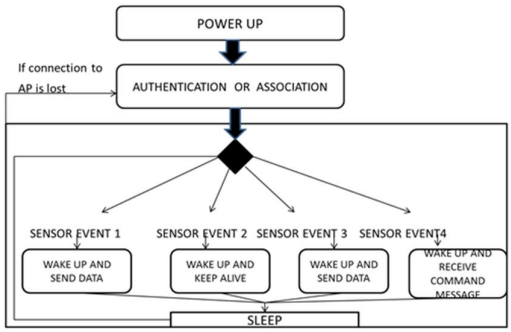

Fig 4. System Flow

from its associated client list if it does not hear from the device for a certain interval. To maintain its association, the device needs to communicate with the access point (AP) periodically.

Periodic data transmission:

Most of the sensing application involves nodes to wake up periodically, read the data, transmit the data and go back to sleep.

Event-triggered data transmission:

The sensor device monitors the environment and if certain events are detected, a message is generated and transmitted.

Command messages:

Another common use case is transmission of messages to the sensor or actuator device. Examples of this can be a query, configuration or command/action messages to the actuator device. The sensor nodes should periodically wakeup and check if any command messages are in wait state.

V. DATA ACQUISITION AND REDUCTION IN POWER USING ARDUINOUNO

Here we try to configure Arduino Uno in such a way that it paves the way for reduction in power consumption of the entire system. The DHT11 (temperature and humidity sensor) and the sound sensor are connected to the appropriate pins on the Arduino Uno board. Once this is done, then to make sure these sensors transmit data, appropriate programming of the Arduino Uno is done. But, here the major aim is to reduce power. Thus, the data from the sensors are sampled by giving them appropriate delays and the processor is made to go to sleep by using watchdog timer of Arduino Uno for 4/8 seconds during each cycle of transmission of data. There are several sleep modes in Arduino Uno which can be used to reduce the power consumption.

The various sleep modes are:

SLEEP_MODE_IDLE: *15mA*

SLEEP_MODE_ADC*: 6.5mA*

SLEEP_MODE_PWR_SAVE*: 1.62mA*

SLEEP_MODE_EXT_STANDBY: *1.62mA*

SLEEP_MODE_STANDBY*: 0.84mA*

SLEEP_MODE_PWR_DOWN : *0.36mA*

Here we have used the power down mode as it draws very less current and it reduces the power sufficiently.

From our observation when power reduction is considered, Arduino Uno uses 20mA from a 9V battery during its active state while it uses only 5mA during its sleep mode, thereby increasing the battery life from 5 days to 18 days as shown in Table 2.

TABLE 2: OBSERVATIONS MADE FROM ARDUINO UNO

Current used from 9v battery	current used in active state by Arduino Uno	Shelf life of battery before power reduction	Current used in sleep mode by Arduino Uno	Shelf life of battery after power reduction
50mA	20mA	5 days	5mA	18 days

VI. DATA ACQUISITION AND REDUCTION OF POWER USING NODE MCU

The esp8266 Node MCU can be used both as a network layer and as a processor. Thus, we can exploit this attribute of esp8266 Node MCU to achieve the required results. The mechanism involved here is the connection of DHT11 (temperature and humidity sensor) and the sound sensor to appropriate ESP pins and programming them using ArduinoIDE to obtain the data. From this we have used Node MCU as a mere processor, now in order to access it as a network layer and to communicate the data transmitted onto a web server, we can make use of certain link tools such as http. Here, when the Node MCU is made to communicate with a nearby Wi-Fi it has an inbuilt function to generate a TCP protocol in order to establish the web server, and it also generates an IP address which helps to access this web server. Once the sensors' data are obtained, the web server is established with the client being available and the IP address generated. Then if the data has to be accessed, it becomes as simple as entering the IP address given by Node MCU onto to the web browser tab and we can immediately see our web server being activated and creating a page which would display our desired results conveniently. Here, the reduction in power consumption is automatically taken care of, as there exists an inbuilt functionality in esp8266 Node MCU to go to
sleep once the required data is transmitted, and it wakes up after the reset button is pressed. Else, if specifically a time limit has to be given for the sleep mode of ESP Node MCU, then it can be achieved through a simple command.
From our observation when power reduction is considered, Node MCU uses 75mA from a 9V battery during its active state while it uses only 8mA during its sleep mode, thereby increasing the battery life from 3 days to 27 days as shown in Table 3.

TABLE 3: OBSERVATIONS MADE FROM NODE MCU

Current used from 9v battery	current used in active state by Arduino Uno	Shelf life of battery before power reduction	Current used in sleep mode by Arduino Uno	Shelf life of battery after power reduction
80mA	75mA	3 days	7mA	27 days

VII. COMPARATIVE STUDY OF ARDUINO UNO AND ESP NODE MCU BASED ON OBSERVATIONS MADE.

- Arduino Uno can work on 5V, hence it has greater compatibility with a wide range of sensors. On the other hand, Node MCU provides only 3.3V supply which poses some amount of restriction.
- Arduino Uno does not support a TCP/IP stack, hence, it has to be interfaced with devices which has one, if the data is to be transmitted onto a web server. While Node MCU does not face this problem, as it has both microcontroller and TCP/IP stack (Network Layer).
- Arduino Uno can be put to sleep through a series of command messages and libraries, while esp Node MCU has an inbuilt capability to go into sleep mode, else if the time of sleep has to be specified, then it can be done using a simple command message.
- As observed, it is seen that there is considerable amount of reduction in power in esp Node MCU when compared to Arduino Uno.
- Arduino Uno can only use Arduino IDE, while esp Node MCU can use Arduino IDE, Python, lua programming software.

VIII. APPLICATION

The application we implemented here was an Environmental Temperature and Humidity monitoring system as we used DHT11 as one of the sensors. This system in general can be used for smart city applications.

IX. CONCLUSION

Here, we are reducing power consumption in both Arduino Uno and esp Node MCU modules and also sending data on to a web server. This in turn can be used for several IOT applications.

X. SCOPE

- Wireless Networks has to be made available every where.

- RTC can be used to make both Arduino Uno and Node MCU to sleep for a longer duration.
- Arduino Uno can be interfaced with the devices having TCP/IP stack to access the web server.
- This system has various smart city applications apart from environmental temperature and humidity monitoring system, such as sound alert systems, soil testing systems etc.

XI. ACKNOWLEDGMENT

We would like to thank our HOD H.S.Aravind of ECE department. We thank Mrs.Shubha B for guiding us and proof reading the paper. We would also like to thank Mr.SachitRao of IIITB for his extensive support.

REFERENCES

[1] ESP8266EX datasheet version 4.3 Espressif systems IOT Team,https://bbs.espressif.com, copyright 2015
[2] https://www.farnell.com/datasheets/1682209.pdf
[3] Low power internet connectivity over Wi-Fi by Texas Instruments.
[4] Neetu Kumari , Nikita Patel , Satyajit Anand , Partha Pratim Bhattacharya,"Designing Low Power Wireless Sensor Networks: A Brief Survey" , vol.2, Issue-9 of IJAREEIE, ISSN (print) :2320-3765,September 2013.
[5] https://www.arduino.cc/en/guide/environment