Low Cost Ambient Monitoring using ESP8266

Ravi Kishore Kodali and Kopulwar Shishir Mahesh
Department of Electronics and Communication Engineering
National Institute of Technology, Warangal
WARANGAL 506004 INDIA
Email: ravikkodali@gmail.com

***Abstract*—A different language is different vision of life. Micropython is the open source scripting language and an interpreter that expresses the concepts in fewer lines of codes. This high level language possesses clear and expressive codes which is designed to respond to the actions. It is a lean and efficient implementation of python 3 programming language that comes with the quick language feedback and is optimized for interactive input to run on microcontroller or embedded systems. This functional and dynamic programming language is ported to ESP8266 board. The DHT11 sensor is interfaced with ESP8266 board to sense the ambient conditions like temperature and humidity. The dht module which is confined to DHT sensor is being imported to perform its programming task. The OLED display used here have dimensions of 128x64, it has an I2C interface and can be driven by SSD1306 driver. Firstly, the library for the SSD1306 driver of the chip needs to be imported. The ESP8266 board have two connections, one with the DHT11 sensor to sense the temperature and humidity, and the second with OLED display to show the collected information of temperature and humidity from DHT11 sensor on to the screen. These all are implemented using lean and efficient Micropython language and thus it is easy to control hardware and connected devices using Micropython.**

I. Introduction

The temperature and humidity of ambient surrounding have variety of changes everyday, it might look familiar for the human beings to predict it, but it never repeats exactly. The seasonal change are known for the largest change in an environment so the temperature and humidity fluctuates with the variety of weather conditions[10]. The variations in an environment can be disruptive and an expensive for the human beings if they do not expect them and monitor them properly so as to plan for such variations in the future[10]. So an automated system is needed which can measure such weather parameters like temperature and humidity without human interventions using sensor taking cost and accuracy into consideration.

Fig.1 shows the simple architecture of the proposed system. The proposed system designed to be cost effective, more accurate and simpler to implement. Thus, the system can be deployed in the wide range of applications like the farmers can use it to plan for their crops and harvests around their seasonal cycle, home automation, weather station, can also be operated by an individual or even in business and so.

As of the 21st Century, the world is growing rapidly with the technology. The people demands higher quality of technology with both the hardware and the software aspects. The world

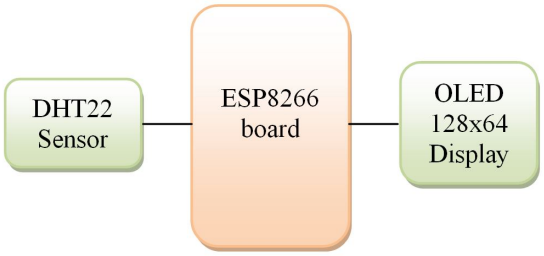

Fig. 1. Block Diagram of Ambient Monitoring System on MicroPython enabled ESP8266 Board

is moving towards Internet of Things (IoT) generation and by 2020 nearly 50 billion devices will be connected to internet [1] and in order to fulfill the requirements the technology should be an optimized, robust, spreadable, and must be used for the good cause. As we move towards the new world of IoT platforms, the one thing should remain consistent is programming language.

As far as computer programming is concern, it became a necessary tool for academic and professional development. The traditional programming language like C/C++ are slower with respect to writing and debugging code. The debugging of computer programming which locates and fix the bugs in the program code may costs around 20%-25% of a large software project[2]. Therefore, it is important to choose a proper programming taking costs of the project into consideration. The micropython gives live development environment and when comparing its writing and debugging speed, it results 5-10 times faster than the traditional programming languages[3].

The speed of the computer programming defines by its relative response when its input is provided. The variables used in programming language associates with its name and the type of data which will be vary. Type checking of such variables can be done either during the compile time as opposed to run time or during the run time as opposed to the compile time[4]. The programming languages like C, C++, Java are said to be static typing as their type checking is done during the compiled time as opposed to the run time[4]. On the other hand, the programming languages like Lua, JavaScript, Micropython are said to be dynamic typing as their type checking is done during the run time as opposed to the compile time, this will result to the quicker response.

978-1-5090-5256-1/16/$31.00

II. CIRCUIT AND WORKING

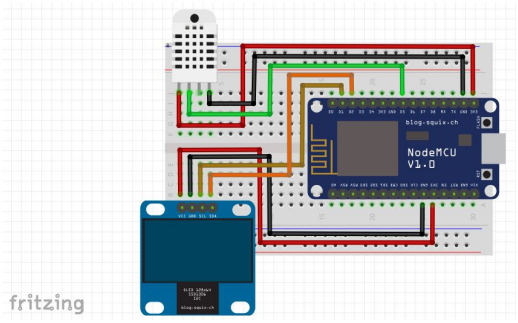

Fig. 2. Circuit Connection of System

The ESP8266 nodemcu board is usb powered, meanwhile DHT22 sensor and 128x64 OLED display are interfaced with the ESP8266 board. The DATA pin of DHT22 sensor is programmed with GPIO14(D5) pin of the board and VCC, GND of the sensor is connected to 3.3V, GND of the ESP8266 board respectively as shown in Fig.2. With the capacitive humidity and temperature sensing, the DHT22 sensor gives the calibrated digital signal output of humidity and temperature. The DHT22 sensor uses 1-wire communication protocol wherein the single bus for data line exchanges the data between the sensor and ESP8266 board to complete the communication.

The OLED display is interfaced with the ESP8266 board with the application of standard I2C protocol designed for low speed devices. The I2C is 2-wire serial protocol comes with Serial Data Line(SDA) and Serial Clock Line(SCL) allowing the serial transmission between the OLED display and the ESP8266 board so a bit is sent on each clock pulse. As in Fig.2, GPIO5(D1) and GPIO4(D2) of the ESP8266 board is configured with the SCL and SDA line of OLED display respectively.

```
# import ssd1306 driver chip for OLED display
import ssd1306
# import I2C interface and Pin configuration from machine module
from machine import I2C,Pin
# import dht module for DHTXX Sensor
import dht
# import machine and time modules
import machine
import time

# Initialize i2c object
i2c = I2C(sda=Pin(4),scl=Pin(5))
# Creating disply object with 128 width, 64 height & i2c connection
display = ssd1306.SSD1306_I2C(128, 64, i2c)
# Configure GPIO14(D5) for DHT22 Sensor
d = dht.DHT22(machine.Pin(14))
```

Fig. 3. Initial setup for Ambient Monitoring System in Micropython

The ambient conditions like temperature and humidity are continuously sensed by DHT22 sensor and the same parameters information is retrieved to display on OLED screen. As far as programming in Micropython for the desired system, ssd1306 module along with I2C interface, Pin configuration from machine module for OLED display and the dht module for DHT22 sensor are necessary to import as shown in Fig.3. After this initialization, the subsequent logic is made and it is programmed in such a way that it updates temperature and humidity information continuously in every one second of time and the same being displayed on OLED screen.

III. HARDWARE ASPECT

A. ESP8266

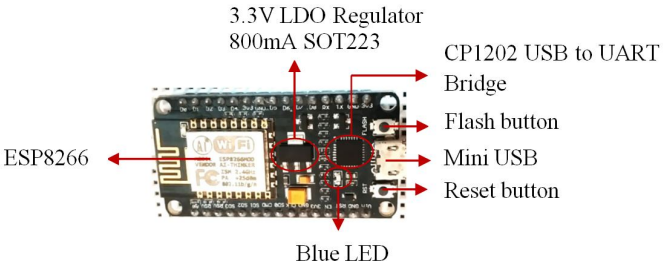

Fig. 4. ESP8266 ESP-12E Development board

ESP8266 is low cost Wifi Module and has ultra low power technology. It is integrated with 32 bit TenSilica L 106 microcontroller which features extra low power consumption. The power saving architecture of the ESP8266 operates in 3 modes: active mode, sleep mode, deep sleep mode. ESP8266 consumes about 60uA in deep sleep mode with RTC clock still running[7]. The algorithms for ESP8266 power saving operation is explained in [7]. The integration of application specific devices or the sensors with ESP8266 board is easy through its GPIO pins which creates a way to connect ESP8266 board to the external world.

TABLE I
ESP8266 ESP-12E SPECIFICATION

Specification Parameters	Values
MCU	32bit TenSilica L 106
Clock Speed	80MHz/160MHz
RAM	<36Kb
Operating Voltage	3.0V ~ 3.6V
Operating Current	80mA (Average)
Available GPIO pins	10

B. DHT22

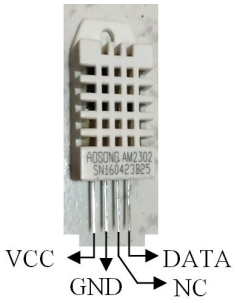

Fig. 5. DHT22 Sensor

DHT22 is low cost, more accurate, highly precision temperature and humidity sensor with 1-wire digital interface.

It has digital signal collecting technique with the humidity sensing technology which assures reliability and stability. It consumes less power, comes with the long term stability and gives calibrated digital signal response. The comparison of DHT22, DHT11, SHT71 humidity and temperature sensors with respect to their cost and specifications is shown in Table II. Thus, it shows DHT22 sensor is more accurate and cost effective for the given ambient monitoring system.

TABLE II
ESP8266 ESP-12E SPECIFICATION

Parameter	DHT22	DHT11	SHT71
Humidity Range	0 - 100%	20 - 80%	0 - 100%
Temperature Range	-40°C∼ 80°C	0°C∼ 50°C	-40°C∼ 123.8°C
Accuracy	±2%(Humd) ±0.5°C(Temp)	±5%(Humd) ±2°C(Temp)	±3%(Humd) ±2°C∼±3°C(Temp)
Repeatability	±0.3% (Humd) ±0.2°C(Temp)	±1%(Humd) ±1°C(Temp)	±0.1%(Humd) ±0.1°C(Temp)
Typical Price	$ 4 - 10	$ 1 - 5	$ 30 - 50

C. OLED 128x64

Fig. 6. 128x64 OLED Display

The given 128 x 64 dot matrix panel consists of 128 segments and 64 commons and has 256 brightness step control. The Data/Commands are sent from Microcontroller through peripheral interface of either I2C interface or SPI interface. The given OLED runs by SSD1306 single chip CMOS driver which embeds with contrast control, display RAM and oscillator which in turn reduces the number of external connections and power consumption. The SSD1306 driver IC is designed for common cathode type OLED panel. The OLED panel has continuous scrolling function in both horizontal and vertical direction which enables to save the screen. It is only having 4 pins which makes the connection with the microcontroller quite simpler.

IV. SOFTWARE ASPECT

A. MicroPython

Micropython is an open source lean and efficient implementation of Python 3 programming language[6]. It is an interpreter that has been optimized to run on small embedded development boards. It has clean and simple syntax to control the hardware with the 93% of code coverage[6]. The REPL stands for Read Evaluate Print Loop is the Micropython interactive prompt that allows to connect to board for experimenting with hardware without any need of compiling and uploading process[8]. The codes are expressive and robust so that the logical concepts are implemented with the fewer lines of codes. With Micropython language ported to ESP8266 it has peripheral support of GPIO, ADC, PWM, I2C/SPI and also the WiFi connections are well supported[9]. The table III gives the comparison between MicroPython and Arduino C/CPP as programming language. The comparison is not exhaustive but it features the necessary advantageous parameter of MicroPython over the Arduino C/CPP.

TABLE III
MICROPYTHON AND ARDUINO C/CPP COMPARISON

FEATURES	MICROPYTHON	Arduino C/CPP
Language Type	Scripting Language Interpreter	Compiling Language
Language Complexity	Simpler	Complex
Implementation	No separate compiling or uploading Step	Compiling and uploading is needed
Memory Management	Automatic, allocation of memory for new variable is not needed	It uses raw pointers to manage and access memory
Syntax	Cleaner and Simpler codes without use of braces and semicolons	Braces and semicolons are mandatory
Static/Dynamic Typing	Dynamic	Static
Writing and Debugging Speed	5 - 10 times faster	Slower
Readability	More	Less

V. EXPERIMENTAL RESULTS

As shown in Fig.7, the DHT22 sensor and 128x64 OLED display are interfaced with usb powered ESP8266 board. Once everything is configured properly with respect to hardware connections and programmatically, the calibrated accurate digital output of temperature and humidity from DHT22 sensor are being displayed on dot matrix OLED panel as shown in Fig.7.

VI. CONCLUSION

The use of ESP8266 board for the desired system makes the system cost effective with the ultra low power consumption capability. The highly precision DHT22 sensor gives out accurate results for temperature and humidity measurements assuring its reliability and stability. The OLED display makes it possible to reduce number of external connections and power consumption for the system. On the other hand, Micropython

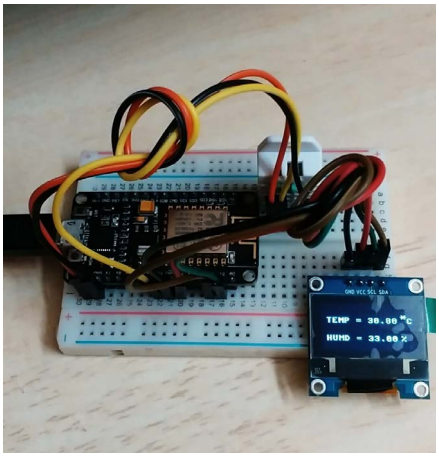

Fig. 7. Sensed temperature and humidity from DHT22 sensor being displayed on 128x64 OLED panel

as programming language is quite easy to learn and explore. Thus overall ambient monitoring system can be optimized with the cost, power and with ease of implementation.

REFERENCES

[1] Dave Evans, "The Internet of Things, How the Next Evolution of the Internet Is Changing Everything", Cisco Internet Business Solutions Group (IBSG), April 2011

[2] Du Chuntao, Empirical Study on College Students' Debugging Abilities in Computer Programming, The 1st International Conference on Information Science and Engineering (ICISE2009)

[3] https://www.pycom.io/support/faqs/what-is-micropython/

[4] Pierce, Benjamin (2002). Types and Programming Languages. MIT Press. ISBN 0-262-16209-1

[5] Kazunori Sakamoto, Hironori Washizaki, Yoshiaki Fukazawa, Open Code Coverage Framework: A Consistent and Flexible Framework for Measuring Test Coverage Supporting Multiple Programming Languages, 10th International Conference on Quality Software, 2010

[6] http://www.Micropython.org/

[7] Tejas Thaker, ESP8266 based Implementation of Wireless Sensor Network with Linux Based Web-Server, 2016 Symposium on Colossal Data Analysis and Networking (CDAN)

[8] https://learn.adafruit.com/micropython-basics-what-is-micropython/overview

[9] https://www.kickstarter.com/projects/214379695/micropython-on-the-esp8266-beautifully-easy-iot

[10] Kevin E. Trenberth, Kathleen Miller, Linda Mearns and teven Rhodes, Effects of Changing climate On weather And human activities, National Center for Atmospheric Research Boulder, Colorado