

DATA MONITORING ON DATA DASHBOARD USING LIFA

N. Anju Latha¹, B. Rama Murthy²

^{1,2} Department of Instrumentation, Sri Krishnadevaraya University, Anantapur, A.P., (India)

ABSTRACT

The evolution of the web and mobile devices has changed how we interact with our surroundings. Data monitoring using LabVIEW interface for arduino systems from a mobile platform is an increasingly important aspect to a flexible and efficient system. NI Data Dashboard for LabVIEW is a mobile application that enables the user to interface and that it can monitor LabVIEW measured parameter remotely. A data dashboard is a tool for visualizing and communication imported data from arduino. The input and output data that can be used with Data Dashboard is shared variables. They can both read and write data to the host system. in present work we are interfacing DHT22 Digital-output relative humidity & temperature sensor with the arduino. The measured Temperature and humidity are transferred to dash board using Lab view.

Keywords: *Sensor DHT22, Arduino ATmega328,LIFA,Shared Variable, Dash Board*

I INTRODUCTION

Dashboards often provide at-a-glance views of KPIs (key performance indicators) relevant to a particular objective. In real-world terms, "dashboard" is another name for "progress report" or "report." The "dashboard" is displayed on a web page that is linked to a database which allows the report to be constantly updated. For example, a manufacturing dashboard may show numbers related to productivity such as number of parts manufactured, or number of failed quality inspections per hour

Data Dashboard allows to create custom, portable views of National Instruments LabVIEW applications [1]. The type of I/O that can be used with Data Dashboard is shared variables. They can both read and write data to the host system. Shared variables interact with the host application through the LabVIEW Shared Variable Engine (SVE) [2] and update in Data Dashboard when they receive updated information from the server. Shared Variables supported data types are Numeric Doubles, Booleans and Strings.

The basic function of the DHT22 is to measure Humidity but in order to do that accurately it also needs to measure temperature. The Temperature and Humidity Monitoring Project has been made to check the accuracy of temperature at a data center. The control and monitoring of the room temperature is done using a DHT22 sensor that

is based on Arduino Uno. Thus, designed to make it easier to regulate and monitor the temperature and humidity of the data center.

II TEMPERATURE AND HUMIDITY SENSOR- DHT22

The DHT-22 is a digital-output relative humidity and temperature sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin. The DHT22 sensor use a serial communication interface, operating over long distances over 100m. A DHT22 sensor[3] is used to record the temperature and the humidity of the environment. The temperature is measured using an NTC temperature measurement component, while the humidity is measured using a resistive-type humidity measurement component. This sensor senses humidity in the range of 20-90% RH (relative humidity) with accuracy of $\pm 5\%$ RH and the temperature in the range of 0-50°C with $\pm 2^\circ\text{C}$ accuracy. The DHT22 sensor has 4 pins: V_{CC}, GND, NC (No Connection) and Data. The digital data pin sends a 40 bit signal data to the controller unit, where the first set of 8 bits (from MSB) constitute of integral part of the humidity value, second set of 8 bits compose the decimal part of the humidity. Similarly, the third 8 bit set and the fourth 8 bit set of the 40 bit data represent the integral and decimal values of temperature respectively. The final 8 bits represent the check sum. The check sum has to be the sum of 8 bit integral part of humidity, 8 bit decimal part of humidity, 8 bit integral part of temperature and 8 bit integral part of temperature, if the data transmission is performed accurately. The overall communication process requires about 4ms. In the work conducted, DHT22 is used to sense the temperature and the humidity of the environment.

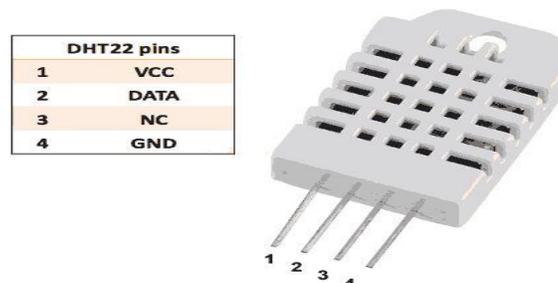


Fig1 :DHT22 Temperature and Humidity sensor

III ARDUINO-ATMEGA 328 MICROCONTROLLER

The Arduino Uno is an open-source microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs) and 6 analog input. It contains everything needed to support the microcontroller, and it can be simply connected to a computer with a Universal Serial Bus (USB) cable to get started.

The Arduino Uno can be programmed with the Arduino Integrated Development Environment (IDE). The C-based simple program code for the Arduino is referred to as a sketch. Collection of sketches for specific functionalities is referred to as libraries. The Arduino can be programmed upto 32 KB memory. Arduino can function autonomously without being connected to a computer, or alternatively programmed to respond mainly to commands sent from the computer via various software interfaces or to the data acquired from the input channels. The Arduino UNO based on ATmega-328 Microcontroller [4] ,[5]

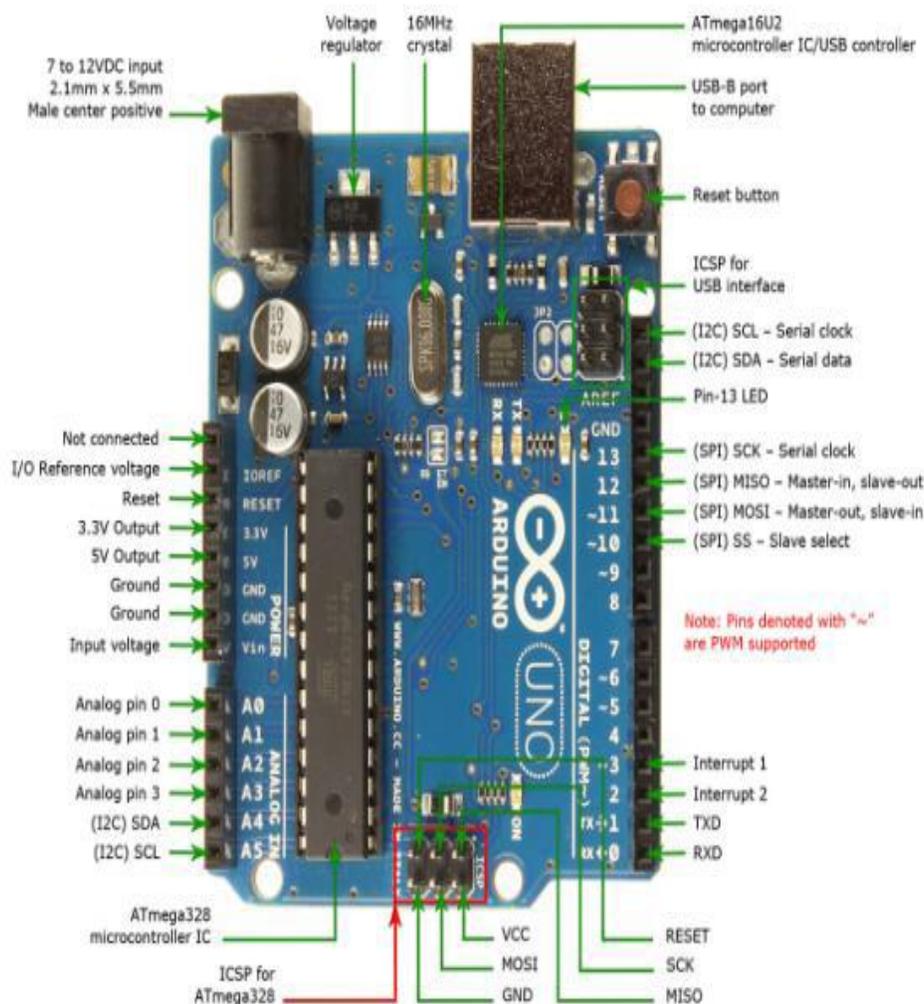


Figure2 : Arduino UNO Based on ATmega-328 Microcontroller

Personal Computer with LabVIEW

LabVIEW (short for Laboratory Virtual Instrument Engineering Workbench) is a system-design platform and development environment for a visual programming language from National Instruments. The graphical language is

named "G". Originally released for the Apple Macintosh in 1986, LabVIEW [6] is commonly used for data acquisition, instrument control, and industrial automation on a variety of platforms including Microsoft Windows, various versions of UNIX,

Linux, and Mac OS X. LabVIEW programs are called virtual instruments, or VIs. LabVIEW contains a comprehensive set of tools for acquiring, analyzing, displaying, and storing data, as well as tools to help in troubleshoot. The proposed system has a VI which is capable of processing the data coming from the arduino through sensor module and shows it graphically through meters, and graphs. To configure and install the application has been made very easy to understand.

LabVIEW Interface for Arduino (LIFA)

The NI LIFA toolkit helps us to interface LabVIEW software with Arduino microcontroller. With LIFA we can acquire or control data from the Arduino microcontroller. Once the information is in LabVIEW, analyze it using the hundreds of built-in LabVIEW libraries; develop algorithms to control the Arduino hardware. A sketch for the Arduino microcontroller acts as an I/O engine that interfaces with LabVIEW VI's through a serial connection. This helps to transmit information from Arduino pins to LabVIEW[7]. Using the common Open, Read/Write, and Close convention in LabVIEW, we can access the digital, analog, pulse-width modulated, I2C, and SPI signals of the Arduino microcontroller. Some of the advantageous of LIFA are

- Interact with your system through a graphical user interface.
- Streamline your design process with intuitive graphical programming.
- Improve your debugging experience with interactive tools.
- Leverage built in resources/functions for implementing simple to complex tasks.
- Open API (Application Programming Interface) allows for complete customization.

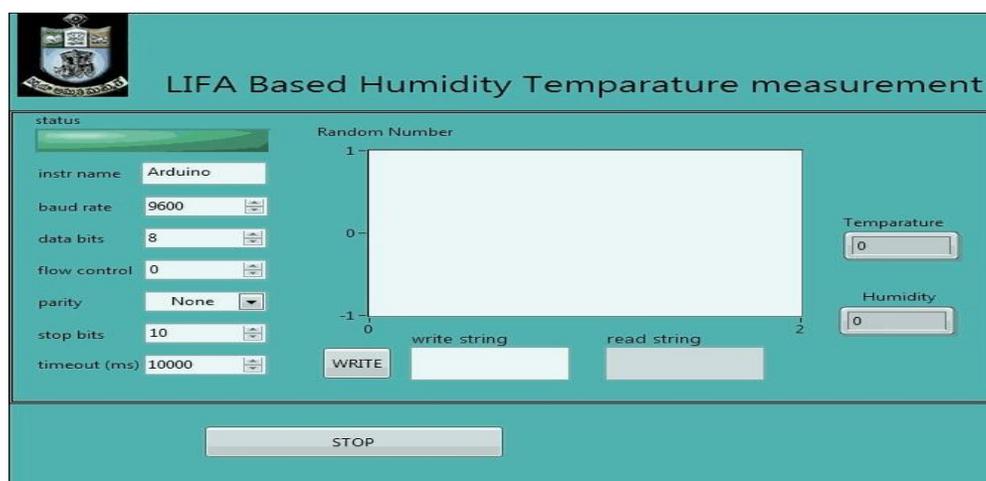


Figure 3: Front Panel of LIFA Based Humidity Temperature measurement

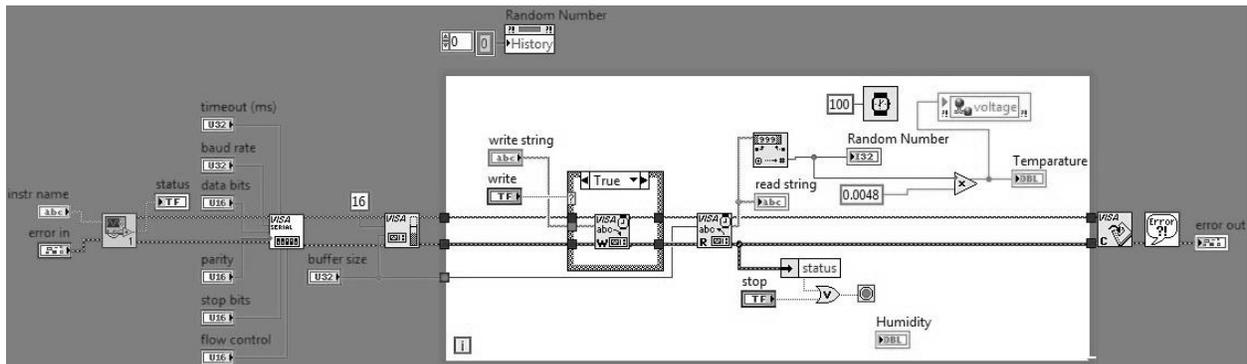


Figure 4: Block Diagram of LIFA Based Humidity Temperature measurement

Select **Shared Variables** in the NI Data Dash Board. Then, enter the Shared Variable server DNS or IP address as shown in the following image. In present case, i have entered the my computer's IP address. Tap **Connect** to connect to the Share Variable Engine.



Fig 5: Server connection

Tap the name of the library that contains the deployed shared variable. Then, select your shared variable, as shown below. The data type of each shared variable is listed below the variable name.

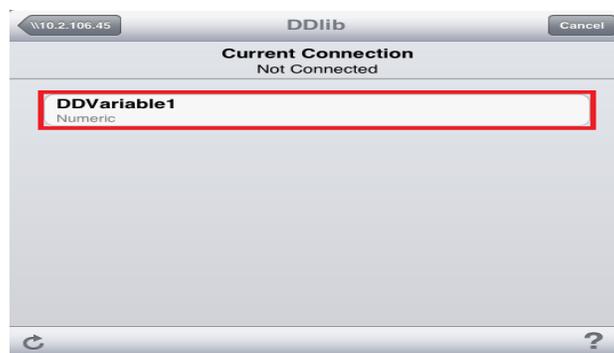


Fig 6: Variable name selection

Run the dashboard by selecting **Play** in the upper right hand corner. The running dashboard should appear similar to the following image.

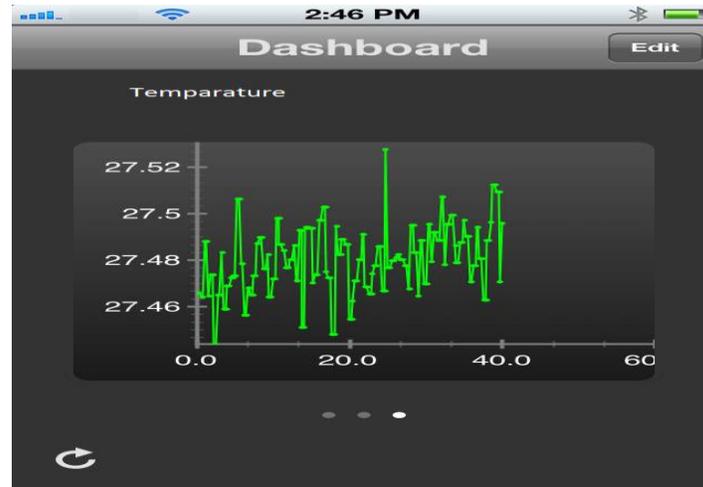


Fig 7: Temperature Display on Dash Board

REFERENCE

1. [https://en.wikipedia.org/wiki/Dashboard_\(business\)](https://en.wikipedia.org/wiki/Dashboard_(business))
2. <http://www.ni.com/white-paper/14689/en/>
3. Temperature and humidity sensor, DHT11. Datasheet, available online: www.micro4you.com/files/sensor/DHT11.pdf
4. Temperature and humidity monitoring and alert Management system, Sophiya Kunjumon Kenneth Pinto Jude Saldanha International Journal of Engineering Research and General Science Volume 4, Issue 4, July-August, 2016, ISSN 2091-2730, 349 www.ijergs.org
5. Distance Sensing with Ultrasonic Sensor and Arduino ,N. Anju Latha1, B. Rama Murthy, K. Bharat Kumar, *International Journal of Advance Research, Ideas and Innovations in Technology. IJARIT* ISSN: 2454-132X ,Volume2, Issue5
6. Naveen Kumar R, et. al., "Low Cost Data Acquisition and Control using Arduino Prototyping platform and LabVIEW", *IJSR, India Online*, Volume 2, Issue 2, February 2013, ISSN: 2319-7064.
7. Gupta Sanjay, Joseph John "Virtual instrumentation using Lab-VIEW", *Electrical Engineering Series*, Tata McGraw Hill pub. India, 2006.