

**Program: ESE 4009\_2**

**INSTRUCTOR:** Prof**.** Mike Aleshams

# Group# 1

|  |  |  |
| --- | --- | --- |
| Student Name | Student ID | Signature\* |
| Ratna Jahnavi Rebbapragada | C0762196 | Jahnavi R |
| Ajay Kumar Vattikonda | C0763572 | Ajay Kumar |
| Prabhnoor Kaur | C0760731 | Prabhnoor |

*\*By signing above you attest that you have contributed to this submission and confirm that all work you have contributed to this submission is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a “0” on the work, an “F” in the course, or possibly more severe penalties.*

**Project Proposal**

**Project Title: IoT Based Weather Station**

**Description of the latest similar system: Arduino Weather Station Using DHT11**

In this universe the are many factors which are the main reason for human survivals. One of them is Climatic/Weather conditions. It includes, rain air pressure, moisture, sun, temperature etc. All the factors affect the environment and small changes can bring a large variation in day to weather report. At that moment when data is not analyzed the predictions may go wrong because there is continuous change in the atmosphere. To avoid such problems a simple “ARDUINO WEATHER STATION USING DHT11” is invented. The main objective of this project is to have an own device which can predict the temperature and humidity of the small surrounding which we are living with simple devices which is portable and keep us on safe side by estimating atmospheric conditions.

**Hardware Components and Description:**

For a simple project we have very less components which are cheap and compact in size. They are:

* DHT11 Temperature and Humidity Sensor:

It is a sensor which is used for sensing temperature and humidity at very low cost. The sensor can consume maximum of only 2.8mA with 3 to 5V power supply. For every second it can take 1Hz sampling rate and shows 0-80% good humidity reading with 5% accuracy and 0-50C temperature with +- 2C accuracy. Small and portable are its special features.

* 16\*2 LCD Keypad Shield:

It is a display device which can show both alphabets and numbers with each character 5\*8-pixel box. The operating voltage is always 4.7v to 5.3v with current consumption of 1mA. There are 16 pins in it where each one has its own specifications as: PIN 1: - VSS Ground, PIN 2: - VDD(+5V), PIN 3:- VE( contrast) , PIN 4:- Register select , PIN 5:- Read/Write, PIN 6:- Enable, PIN ( 7-14) :- Data pins , PIN 15:- LED +Ve , PIN 16:- LED -Ve.

* Arduino Mega:

It is an open-source electronic platform and a microcontroller with 54 Digital I/O pins where 16 are analog i/p’s and 4 are serial ports. Along with them it contains 16MHz crystal oscillators, a USB connection, a power Pack, an ICSP header and a Reset button. The cross-platform feature in Arduino gives access to many OS to work and run operations. Simple programming language ARDUINO IDE is understandable.

* Jumper wires:

Connecting the components to bread board and pins of sensor to Arduino.

* Bread board:

To hold the components and connection purpose.

* Power supply:

To supply power to Arduino and LCD display to work and produce O/P.

* Resistors:

Here in this project the resistors are used to control the current flow and adjust the signal levels are the power consumption is more by Arduino and LCD display which may damage the system.

**Software Requirements and Description:**

* Arduino IDE

**Block diagram:**

A

R

D

U

I

N

0

MEGA

Power Supply

DHT11 Temperature and humidity sensor

Resistor

16\*2 LCD Keypad Display

**Figure: Block diagram of Arduino Weather Station Using DHT11**

**Working:**

Firstly, take an Arduino Mega which has 54 digital I/O pins and it is easy to fix the 16\*2 LCD Display in it. Now connect the power supply and see the display is working. As there is no information or sensor connected, we can see a blank space. Now take the DHT11 temperature and humidity sensor which has 3 pins which are GND, VCC, SIGNAL O/P. To get a perfect output just connect the VCC pin of the sensor to 5V of Arduino, GND of the sensor to GND of Arduino and finally SIGNAL O/P of the DHT11 sensor to any digital pin (pin 22). As the power supply is on, we can see that the DHT11 sensor senses the temperature senses the temperature and humidity of very short area near you and display the reading on the LCD where the information is processed by the Arduino.

**Limitations of the latest similar system:**

* It is limited to smaller area around you and cannot cover large areas like a house or city etc.
* It doesn’t have any other sensors which can calculate all the features of atmospheric changes like rainfall, oxygen, and moisture etc.
* The system doesn’t have any storage device for recording and analyzing/reading data.
* Alerting about weather is not there and you won’t get any information in emergency. There is no IoT based module to transfer message.
* Arduino uses single programming language and the code cannot be written any other language
* No accuracy in the calculation of temperature and humidity its only approx. Value.
* 2sec delay in every reading and cannot show decimal temperatures and humidity
* Power consumption is 0.44 watts, current is 90Ma which is high for low projects.
* No web applications and GPS/GSM modules.

**References:**

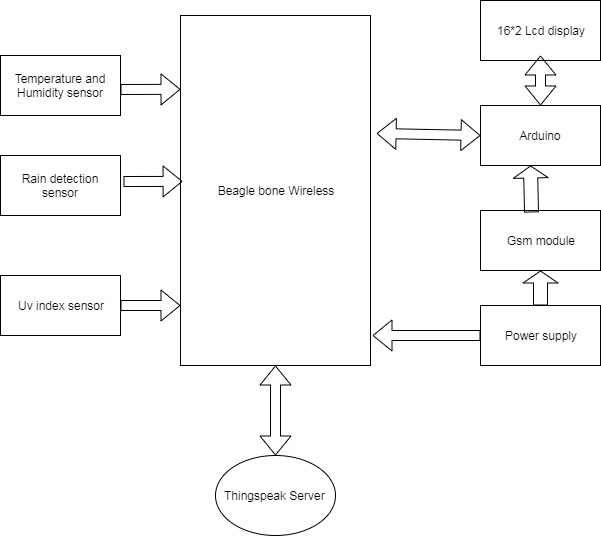
1. [**https://www.electronics-lab.com/project/arduino-weather-station-using-dht11/**](https://www.electronics-lab.com/project/arduino-weather-station-using-dht11/)
2. [**https://shop.evilmadscientist.com/productsmenu/716#:~:text=The%20DHT11%20is%20a%20basic,careful%20timing%20to%20grab%20data**](https://shop.evilmadscientist.com/productsmenu/716#:~:text=The DHT11 is a basic,careful timing to grab data)**.**
3. [**https://store.arduino.cc/usa/mega-2560-r3**](https://store.arduino.cc/usa/mega-2560-r3)
4. [**https://components101.com/16x2-lcd-pinout-datasheet**](https://components101.com/16x2-lcd-pinout-datasheet)

**FINAL SOULTION:**

**INTRODUCTION:**

Climatic change and environmental monitoring have received much attention recently. Man wants to stay updated about the latest weather conditions of any place like a college campus or any other particular building. Since the world is changing so fast so there should be the weather stations. Here we present a weather station that is very helpful for any places. This weather station is based on IOT (internet of things). It is equipped with environmental sensors used for measurements at any particular place and report them in real time on cloud. To accomplish this we used Arduino Uno, Beagle bone wireless and different environmental sensors like Temperature and Humidity sensor, UV sensor and rain drop sensor. IoT means Internet of Things. It provides internetworking of physical devices, buildings, vehicles and other components like sensors and actuators. By giving network connectivity to systems embedded with electronics, software, sensors and actuators; these objects are able to collect and exchange data. By using IoT objects to be sensed or controlled remotely through existing network. It gives opportunity to connect physical world with computer-based systems. IoT improves efficiency, accuracy, economic benefits along with reduced manpower.

**BLOCK DIAGRAM:**



**Use of various peripherals such as touch screens, cameras, microphones and speakers, GPIOs, timers, GPS modules, Bluetooth, WIFI, and ADC/DACs?**

* Temperature and humidity sensor are used for finding the temperature and humidity in the atmosphere, the Uv sensor which will be used for finding the uv index in atmosphere
* GPIO (General purpose input and output pins): These are used for communicating the sensor input with the beagle bone.
* Use of the Beagle bone wireless as the master device because it has the inbuilt WIFI module for communicating with cloud
* Use of Arduino for interfacing the LCD display with the beagle bone black.
* Use of Gsm module for communicating with the user if any extreme weather conditions

**Use of I2C, SPI, RS232/RS-485, IrDA infrared, JTAG, USB, Bluetooth, IEEE 802.11 WIFI, IEEE 802.3 Ethernet, CAN and GPS protocols and systems?**

* The Wi-Fi protocol used is IEEE802.11 for sending the data to the cloud.
* Digital communication: This is used for communicating all the devices connected to the beagle bone
* The communication to master device from the host machine is through SSH which uses the TCP(Transmission communication protocol)
* UART Protocol is used to interface Arduino Uno with Beagle Bone.
* UART protocol is used to interface the Gsm module with the Arduino
* **Use of preemptive versus cooperative scheduler operation; tick rate and time slicing; critical code; fixed, dynamic and hybrid task priority allocation; application-specific considerations; power management tactics; semaphores, mutexes and queues; debugging strategies; performance estimation?**
* Use of pre-emptive scheduler operation concept is used in this system where as each task assigned to sensors and other peripheral devices used is assigned for their functioning.
* The sensor data is processed in real time and is sent to the cloud.
* The data stored in the cloud is accessible from anywhere.
* Considering the extreme weather conditions in environment an interrupt is used for temperature sensor so that the system sends the SMS through the gsm module upon detection of extreme weather
* Since the project is stationary and connected to the wall adapter power management tactics are not discussed.

**Hardware and software requirements.**

**Hard ware components**

* Beagle Bone wireless
* Temperature and Humidity sensor
* UV sensor
* Rain level sensor
* Arduino uno
* 16\*2 lcd display
* Gsm module
* Jumper wires and breadboard wires
* Breadboard

**Hardware Tools:**

**1.Breadboard:**

A breadboard is a solder less construction base used for developing an electronic circuit Itcomes indifferent types; full-sized, Full+, half-sized, Half+, and mini.

**2.Digital Multimeter:**

DMM is a test instrument used to measure electrical values including voltage, current and resistance and for testing, diagnosing, and troubleshooting electrical circuits, components, and devices.

**3.Jumper wires:**

These are simple wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed.

**4. Laptop:**

Laptop is necessary for our entire project and must have all the software tools.

**5. Soldering tools:**

Soldering iron is used to connect components on the PCB board.

Solder is substance that melts and forms the bond between the two soldered components.

Wire strippers are used to remove the insulation from electric wires in order to make contact.

Wire cutter commonly used to cut copper, brass, iron, aluminium, and steel wire .Tweezers are used to keep components in place and avoid burning your fingers while soldering. Are used to keep components in place and avoid burning your fingers while soldering.

**6. Beaglebone wireless:**

It is a low-cost, credit-card sized development platform with good support from a fast-growing community. It comes with built-in wireless networking capability Like Beagle Bone Black, Wireless retains HDMI output, serial debug port, PC USB interface, USB 2.0 host, microSD card slot, reset and power buttons, and adds two more status LEDs for WIFI and Bluetooth. They have also changed the USB programming port from a mini to a micro, which should allow better compatibility with common cables and power supplies. Also retained is the 4GB of onboard eMMC memory with Debian Linux pre-installed, allowing Beagle Bone Black Wireless to boot in around 10 seconds. Beagle Bone Black Wireless is fully software, hardware and mechanically compatible with Beagle Bone Black, and is even compatible with all existing Beagle Bone.

Software Compatibility

* Debian
* Android
* Ubuntu

**7.Arduino Uno.**

It is an open-source platform, means the boards and software are readily available and anyone can modify and optimize the boards for better functionality. The controller appears not to be expensive and uses low electrical power, 5.5 volts. C and C++ were employed for this development. Arduino can connect to a computer via the Universal Serial Bus (USB) If you are planning to create a project relating to digital electronics, embedded system, robotics, or IoT, then using Arduino Uno would be the best, easy and most economical option. It can be programmed using C and C++ language.

**8. DHT11**

The DHT11 is a basic, low-cost digital temperature and humidity 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.In this Temperature will display in Celsius. Humidity (H) will display in %.

Specifications:

Low cost

3 to 5V power and I/O

Good for 0-100% humidity readings with 2-5% accuracy

Good for -40 to 80°C temperature readings ±0.5°C accuracy

Body size 15.1mm x 25mm x 7.7mm

4 pins with 0.1" spacing

**9. UV Sensor**

UV sensors measure the power or intensity of incident ultraviolet (UV) radiation. UV sensors are used for determining exposure to ultraviolet radiation in laboratory or environmental settings.

**10.Rain Drop Detector**

This is great for detecting the presence of rainfall or even small water droplets. The sensors give both an analog and digital output which can be feed to your data acquisition device. The detection of rain can be used to take actions for other applications. Raindrop Sensor is a tool used for sensing rain. It consists of two modules, a rain board that detects the rain and a control module, which compares the analog value, and converts it to a digital value.

**11. 16×2 LCD module**

LCD modules are very commonly used in most embedded projects, the reason being its cheap price, availability and programmer friendly. It has 16 Columns and 2 Rows. The 16x1 display unit will have 16 characters and are in one line. The 16x2 LCD will have 32 characters in total 16in 1st line and another 16 in 2nd line. AnInterface IC like HD44780is used, which is mounted on the backside of the LCD Module itself. The function of this IC is to get the Commands and Data from the MCU and process them to display meaningful information onto our LCD Screen.

Features of 16×2 LCD module

•Operating Voltage is 4.7V to 5.3V

•Current consumption is 1mA.

•Every character can be built with a 5×8 pixel box

•Alphanumeric LCD display module, meaning can display alphabets and numbers

•Consists of two rows and each row can print 16 characters.

**GSM module**

* Global system for mobile communications(GSM).
* A GSM module or a GPRS module is a chip or circuit that will be used to establish communication between a mobile device or a computing machine and a GSM or GPRS system.
* They can feature all the functionalities of a mobile phone through computer like making and receiving calls, SMS, MMS etc. These are mainly employed for computer based SMS and MMS services.

**Software Requirements**

* Debian latest image of beagle bone wireless
* Linux os
* EasyEda/fritzing
* Arduino Ide
* Eclipse IDE
* C/C++ programming
* GCC compiler
* Things speak

**Software Tools:**

**1.Text editor:** It is a program used for editing text files. The text editors are ideal tools for anyone who needs to write quickly and simply, read source code, or create text files.

**2. IDE:** IDE or Integrated Development Environment is a tool or software that allows you to write, edit, modify, run, execute, debug and deploy your code.

• **Eclipse IDE:** Eclipse is one of the best IDEs to develop programs in C, C++.Eclipse is a very powerful and is used for other programming languages as well.

• **Arduino IDE:** Arduino IDE is an integrated development environment. Can be used over different operating systems or platforms i.e. Windows, macOS, Linux. IDE is used for writing and uploading the code of the hardware for programming the microcontroller accordingly.

Features:

Both Analog and digital signal can be read by the IDE.

Arduino IDE supports different languages for writing the code -C, C++, JAVA.

**3.** **Compiler:**

The most important software-development tool in Linux is GCC — the GNU C and C++ compiler. GCC stands for GNU Compiler Collections which is used to compile mainly C and C++ language.

**Simulation Software:**

It is the process of creating a model of an existing or proposed system to predict the future behaviour of the system.eg Easy EDA, fritxing, proteus are simulation and design software tools, used for schematic, simulation and PCB designing.

**5.ThingSpeak**

Thing Speak is an application which is open source used for the purpose of mail of Internet of things (IOT) API used for storing the information gathered from the internet of things system. Data can also be retrieved using different protocols i.e. HTTP protocol or LAN. Thing Speak make us able to create the sensor logging applications, as well as location tracking applications, and a status updates with social network of things”. Thing Speak has major support from MATLAB which is a software of numerical computing from MathWorks which allow Thing Speak users to analyse and retrieve uploaded data using MATLAB without wasting money on its licenses.

**6.** **Debugger:**

GDB, the GNU Project debugger, allows you to see what is going on `inside' another program while it executes -- or what another program was doing at the moment it crashed.

**Working of the Flowchart:**

The Whole system consists of a beagle bone wireless as the main unit of processing for the entire system. Firstly, when the power supply in ON the sensors are operated through the beagle bone which are connected to any of the GPIO (General purpose input and output) to retrieve the data from the sensors and do the analysis with the sensor data and updates data to the internet using inbuilt Wi-Fi module. There is an in-built memory to beagle bone wireless (4GB on board eMMC memory) and also a SD card slot is used to store the operating system as well as all programs and files needed for this project. The Arduino Uno which is connected to GPIO pin of beagle bone wireless will collect all the sensor data from the SD card or in-built memory and then send to the Thing Speak website while the Arduino process the information and display it on LCD. The data collected will be analysed to configure the actual condition and the current condition after the values of the temperature, humidity, rain level and the UV spectrum light readings are store in the cloud server. Considering the extreme weather condition sometimes in winter or summer season when the temperature drops too low or too high we are suing the **GSM** module to give the user a notification regarding this .The result of this data analysis then will be made and tell the user about weather condition is it good or bad. We can also use IOT platforms such as Thing Speak IoT to collect data into the cloud for analysis.

**Flow Chart:**

START

INITIALIZE THE SENSORS AND ALSO BEAGLEBONE BLACK WIRELESS WHICH HAS IN BUILT WIFI MODULE IN IT.

LCD DISPLAY

COLLECT THE DATA FROM THE SENSORS AND STORERD IN THE SD CARD OR IN-BUILT MEMORY WHICH IS CONNECTED TO THE ARDUINO.

if

THE TEMPERATURE IS HIGH THAN THE THERSHOLD VALUE

SEND THE MESSAGE TO THE USER USING GSM MODULE

if not

SEND SENSORS DATA TO THE THINK SPEAK SERVER.

CLOUD SERVE DATA BASE

DATA VISUALIZATION AND ANALYSIS

STOP

**References:**

* [**https://www.researchgate.net/publication/333698819\_IOT\_Based\_Weather\_Monitoring\_and\_Reporting\_System\_Project/link/5cffb14b299bf13a384cab8a/download**](https://www.researchgate.net/publication/333698819_IOT_Based_Weather_Monitoring_and_Reporting_System_Project/link/5cffb14b299bf13a384cab8a/download)
* [**https://www.irjet.net/archives/V6/i1/IRJET-V6I1220.pdf**](https://www.irjet.net/archives/V6/i1/IRJET-V6I1220.pdf)
* <https://blogs.mathworks.com/iot/2017/01/20/use-mqtt-to-send-iot-data-to-thingspeak/>
* [https://www.electronics-lab.com/interesting-sensors-to-add-to-your-weather-station-project/#:~:text=Light%20Intensity%20Sensor-,Light%20Intensity%20Sensor,be%20confused%20with%20UV%20sensor](https://www.electronics-lab.com/interesting-sensors-to-add-to-your-weather-station-project/#:~:text=Light Intensity Sensor-,Light Intensity Sensor,be confused with UV sensor).
* [**https://www.geeksforgeeks.org/gcc-command-in-linux-with-examples/**](https://www.geeksforgeeks.org/gcc-command-in-linux-with-examples/)
* [**https://www.google.com/search?q=flowchart+for+weather+station+using+sensors&tbm=isch&ved=2ahUKEwjZ9YK49cvuAhUlA50JHSdOAZYQ2-cCegQIABAA&oq=flowchart+for+weather+station+using+sensors&gs\_lcp=CgNpbWcQAzoECCMQJ1C7qilYw9gpYNzbKWgDcAB4AIAB7wKIAbYekgEINi4xNC4yLjSYAQCgAQGqAQtnd3Mtd2l6LWltZ8ABAQ&sclient=img&ei=l6UZYJnONaWG9PwPp5yFsAk#imgrc=8V3RRmvMjc7HGM**](https://www.google.com/search?q=flowchart+for+weather+station+using+sensors&tbm=isch&ved=2ahUKEwjZ9YK49cvuAhUlA50JHSdOAZYQ2-cCegQIABAA&oq=flowchart+for+weather+station+using+sensors&gs_lcp=CgNpbWcQAzoECCMQJ1C7qilYw9gpYNzbKWgDcAB4AIAB7wKIAbYekgEINi4xNC4yLjSYAQCgAQGqAQtnd3Mtd2l6LWltZ8ABAQ&sclient=img&ei=l6UZYJnONaWG9PwPp5yFsAk#imgrc=8V3RRmvMjc7HGM)

**MILESTONES And DELIVERABLES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task Name** | **Start Date** | **End date** | **Person In charge** |
| Project proposal | Jan 18,2021 | Feb 12,2021 |  |
| Finalizing/ordering Hardware components | Feb 13,2021 | Feb 20,2021 | Ajay |
| Testing Hardware components | Feb 21,2021 | Feb 23,2021 | Ajay |
| Circuit design | Feb 24,2021 | Feb 28,2021 | Prabhnoor |
| Interfacing the Temperature and humidity DHT11 sensor with BB-wireless | Mar 01,2021 | March 03,2021 | Ratna |
| Interfacing the UV sensor with BB-wireless | March 04,2021 | March 07,2021 | Prabhnoor |
| Interfacing the rain detection sensor | March 08,2021 | March 10,2021 | Ajay |
| Interfacing the LCD display with Arduino | March 11,2021 | March 15,2021 | Ratna |
| Interfacing the Arduino with the Beagle bone | March 16,2021 | March 19,2021 | Prabhnoor |
| Interfacing Gsm module with Arduino | March 20,2021 | March 30,2021 | Ajay |
| Interfacing the cloud storage with BB-wireless (Thing speak) | March 31,2021 | April 05,2021 | Ratna |
| Pcb design | April 6,2021 | April 10,2021 | Prabhnoor |
| Zero Pcb Implementation | April 11,2021 | April 19,2021 | Ratna |
| Final report presentation | April 20,2021 | April 25,2021 |  |
| Final presentation | April 26,2021 | April 28,2021 |  |

**Coding standard for final solution:**

**C coding standard:** MISRA C (Motor Industry Software Reliability Association)is a coding standard which contains a collections of coding rules, guidelines. Usually, it is most used in embedded systems which are programmed in ISO C/C90/C99. These coding standards are important to make sure the code is safe, reliable, testable, maintainable, and portable. The two coding standards used for C programming are: MISRA and CERT. Helix QAC is used to find out and report any violation of MISRA rules and directive in C and C++.

The three standards released by MISRA C are:

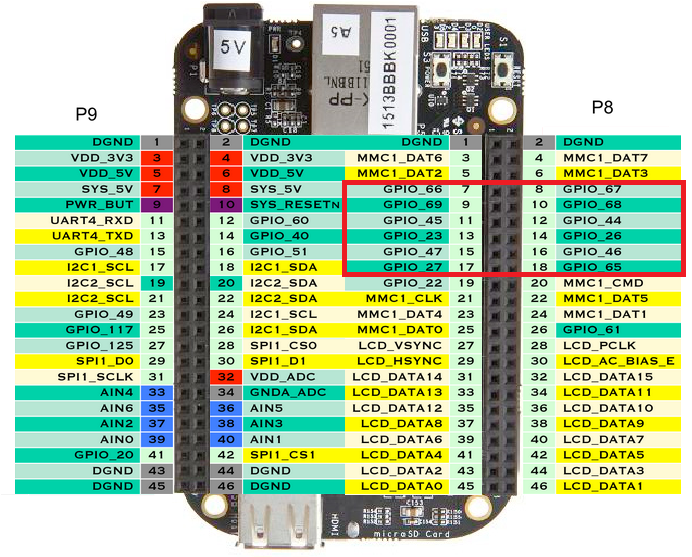
* MISRA C: 1998: - It was written for C90. They are 127 coding rules.
* MISRA C:2004: - It was written for C90. They are 142 coding rules.
* MISRA C:2012: - It was written for C99. They are 143 coding rules.
* MISRA C: 2012 Amendment 1: - It has 156 rules and 17 directives total of 173 guidelines.
* MISRA C:2012 Amendment 2: - It has 158 rules and 17 directives total of 175 guidelines.
* MISRA C:2012 Addenda: - This is designed for the functional safety and also security.

**Python Coding Standards:**

Python Enhancement Proposal (PEP-8) is a documentation providing information to the python community. It is a programing style guide used to make the code maximize its readability. In this feature and a rationale are provided to the coding sector. They are three types of PEP-8. They are:

* **STANDARD TRACK:** It describes a new features or implementation for python.
* **INFORMATIONAL:** Python design issues/ general guidelines or information to python community.
* **PROCESS:** Describes surrounding python issues or process to change an event in it which are not pythons code base.

Beagle bone pin diagram



**Connecting LCD to Arduino:**

The Arduino community has a library called LiquidCrystal that makes programming the LCD module less difficult.

// include the library code:

#include <LiquidCrystal.h>

Next we have to create an LiquidCrystal object. This object uses 6 parameters and specifies which Arduino pins are connected to the LCD’s  pins.

// Creates an LCD object. Parameters: (rs, enable, d4, d5, d6, d7)

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

lcd.begin():This function sets the dimensions of the LCD. The number of rows and columns are specified as lcd.begin(16, 2).

Four data pins (D4-D7) from the LCD will be connected to Arduino’s digital pins from #4-7.

The Enable pin on LCD will be connected to Arduino #2 and the RS pin on LCD will be connected to Arduino #1.

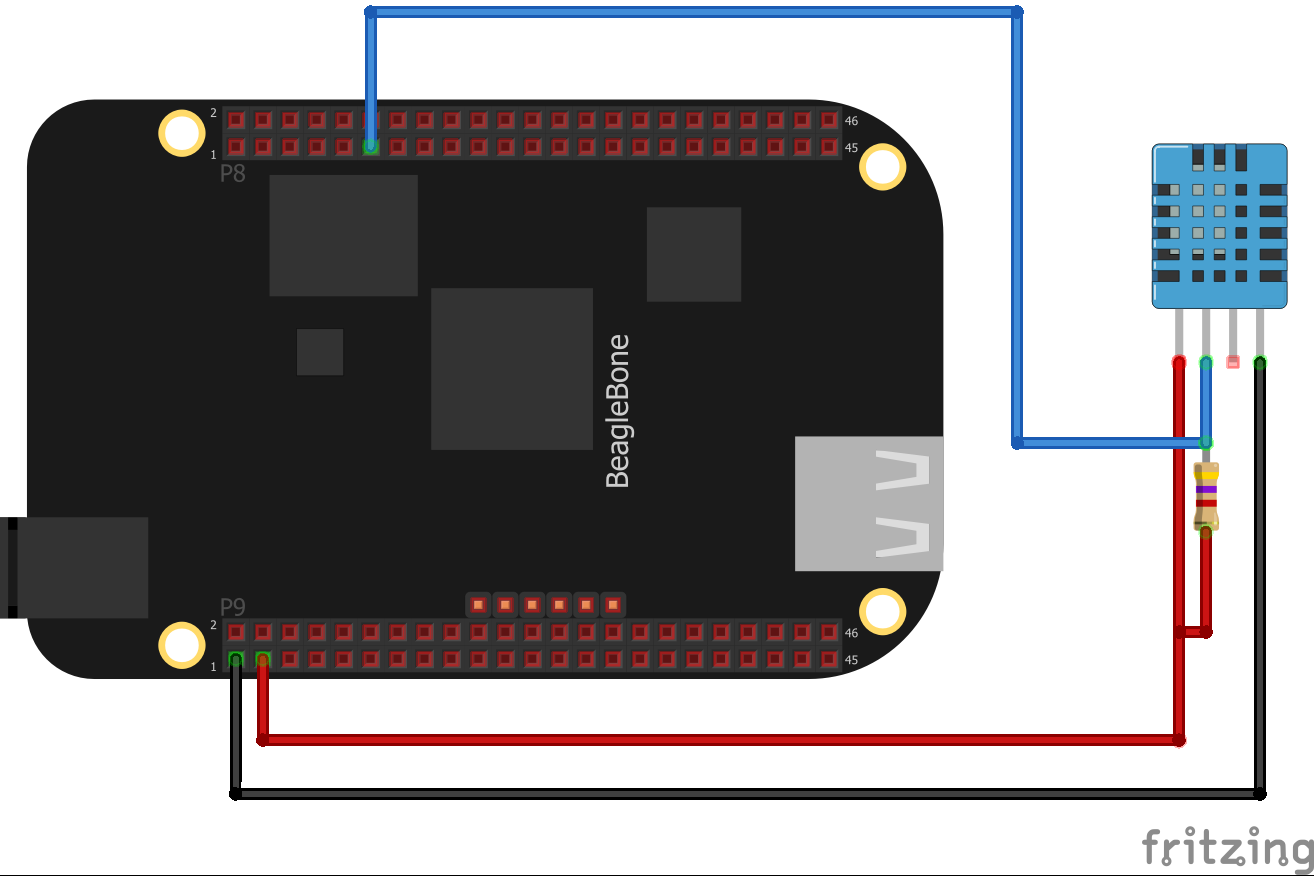
GND should be connected to the ground of Arduino.

VCC is the power supply for the LCD which we connect the 5 volts pin on the Arduino.

LCD RS pin to digital pin 12

* LCD Enable pin to digital pin 11
* LCD D4 pin to digital pin 5
* LCD D5 pin to digital pin 4
* LCD D6 pin to digital pin 3
* LCD D7 pin to digital pin 2

**DHT11**



The above picture shows the connection between beagle bone and DHT11 sensor. Beagle bone pin P9.6 is used to power the sensor and pin P9.2 is connected to the ground of the sensor. There are almost 69 I/O pins available on the beagle bone. The GPIO Pin P9.15 of beagle bone is connected to the data pin of the sensor. We can use Adafruit Blinka (circuit python), we can install circuit python libraries straight to our small Linux board. In this case we are going to install the CircuitPython\_DHT library. This library works with the DHT22 and DHT11 sensors.

Commands to install the CircuitPython DHT library:

Pi3 install adafruit-crircuitpython-dht

sudo apt-get install libgpiod2

**If you're using a DHT11 sensor**, you can change the sensor type by**renaming the DHT22 class to DHT11**:

dhtDevice = adafruit\_dht.**DHT11**(**board.D18**)

**UV sensor**

In Beagle Bone analog signal can be converted directly to the Digital Signal. It has a total of 7 A/D channels, All ADC channels and power pins are in expansion header P9. The UV Sensor has 4pins 3V3, GND, OUT, EN.

****

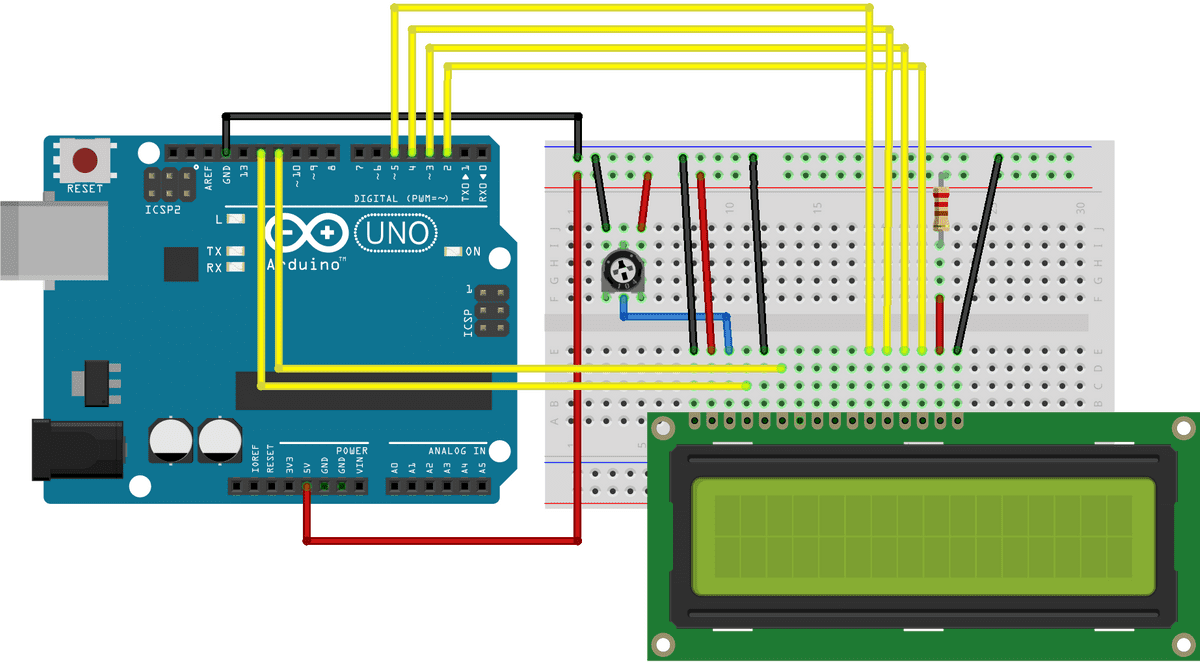
Connect the EN pin on the to 3.3V to enable the device. Also connect the 3.3V pin to beagle bone pin P9.32.

Connect ground of sensor to pin P9.34 and connect output pin to pin P9.36 of the beagle bone.

**Rain drop sensor**

****

This module has 4 pins:

* VCC: Module power supply – 5 V
* GND: Ground
* D0: Digital Output
*  A0: Analog output

Connect Vcc to pin P9.6 of beagle bone and gnd to pin P9.2

Connect pin D0 to pin P9.15 of beagle bone.

**GSM module (SIM 900)**

#### Connecting GSM Module to Arduino

The operating voltage of SIM900 chip is from 3.4V to 4.4V. You can add an external power supply to the shield with the 5.5mm DC jack, to which you can connect any 5V-9V DC wall adapter you have. Next to the DC jack, is a Slide Switch to select the power source labeled EXTERN. To use external power source, move the slider as shown above.

UART communication is used to communicate with Arduino. While programming Arduino uses serial ports to load program from the Arduino IDE. If these pins are used in wiring, the program will not be loaded successfully to Arduino. So, you have to disconnect wiring in Rx and Tx each time you burn the program to Arduino. Once the program is loaded successfully, you can reconnect these pins and have the system working!

To avoid this difficulty, there is a alternate method in which two digital pins of Arduino are used for serial communication.

Software Serial is a library of Arduino which enables serial data communication through other digital pins of Arduino.

### Code explanation for gsm module interfacing with Arduino:

AT commands are used to configure the module in different modes and to perform various functions like calling, posting data to a site, etc.

#include <SoftwareSerial.h>

**We need to select two PWM enabled pins i.e., pins 7and 8** **of Arduino. First, we need to create a constructor of Software Serial with name my Serial and pass the digital pin numbers as parameters.**

**Software Serial my Serial (7, 8);**

We initialize a serial communication link between Arduino, Arduino IDE and SIM900 shield. The **first task is to set baud rates of Software Serial library to communicate with GSM module.**

**mySerial.begin**

To set the baud rate of Arduino IDE’s Serial Monitor

Serial. Begin **function**

Both should be set at the same baud rate **.**

Serial.available() – **checks for any data coming through serial port of arduino.**

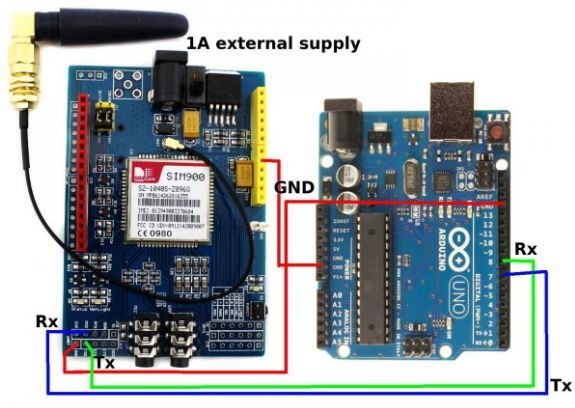
mySerial.available() – **checks for any data coming from GSM module**

Serial.write()**– Prints data to serial monitor of Arduino. So the function**Serial. Write(mySerial.read())**– prints the data collected from software serial port to serial monitor of Arduino.**

Send Message ()  -**is the function we created to send an SMS.** **AT Command “AT+CMGF=1”** is used to **set our GSM module to Text mode** using **mySerial.println() function.** **AT command** ("AT+CMGS=\"+639xxxxxxxxxx\"\r") is used to input the number to which you want to send the sms with country code .

RecieveMessage() – **is the function to receive an SMS (a live SMS). The AT command to receive a live SMS is “AT+CNMI=2, 2, 0, 0, 0” – we just need to send this command to GSM module.**

If you want to read all SMS’s stored in your SIM card, send the following AT Command to gsm module – “AT+CMGL=\”ALL\”\r”

****

**Interfacing beagle bone with thing speak:**

Enabling WIFI

debian@beaglebone: ~$ sudo connmanctl

**[sudo] password for Debian: temppwd**

connmanctl> tether WIFI disable

connmanctl> enable WIFI

connmanctl> scan WIFI

connmanctl> services List off visible WIFI networks

connmanctl> agent on to handle secure WIFI networks

Agent registered

connmanctl> **connect *wifi\_506583e2cd76\_416c6f66745f4775657374\_managed\_psk***

*Replace with your WIFI network and* paste after *‘connect’*.

*...Provide password…*

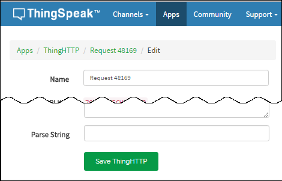
connmanctl> quit

We are using thing Speak as a web server. It is an open source “Internet of Things” application and API to store and retrieve data from things using HTTP.

Http protocol id TCP/IP protocol to deliver data on world wide web. It is standard way of computers to communicate. It is a response/request protocol based on client/server.

thing Speak allows you to aggregate, visualize, and analyze live data streams in the cloud.  Iolib is data i/o library and provides a unified interface for accessing different types of data streams.

The primary element of thing Speak activity is the channel. You put data into a thing Speak Channel by using HTTP POST. You create an HTTP object using the Thing HTTP App and then control the object with simple API commands.



To Create a new Thing HTTP request. Click Apps > Thing HTTP, and then click New Thing HTTP.

In a free trial version cloud service gets updated every 15 seconds.

We have to follow following steps:

a) Sign In to Things peak™ using either your Math Works® Account or thing Speak account, or create a new Thing speak account. b) Click Channels > My Channels. c) On the Channels page, click New Channel. d) Check the boxes next to Fields 1–3. Enter these channel setting values: Name: Smart Healthcare System, Measurement Field 1: Temperature (F), Field 2: Heart Beat, Field 3: ECG e) Click Save Channel at the bottom of the settings.

You now see these tabs:

Private View: This tab displays information about your channel that only you can see.

Public View: If you choose to make your channel publicly available, use this tab to display selected fields and channel visualizations.

Field – One of eight specific locations for data inside of a channel, identified by a number between 1 to 8 – A field can store numeric data from sensors.

Channel Settings: This tab shows all the channel options you set at creation. You can edit, clear, or delete the channel from this tab. Every channel has a unique Channel ID. The Channel ID number is used to identify the channel when your application reads data from the channel.

API Keys: This tab displays your channel API keys.

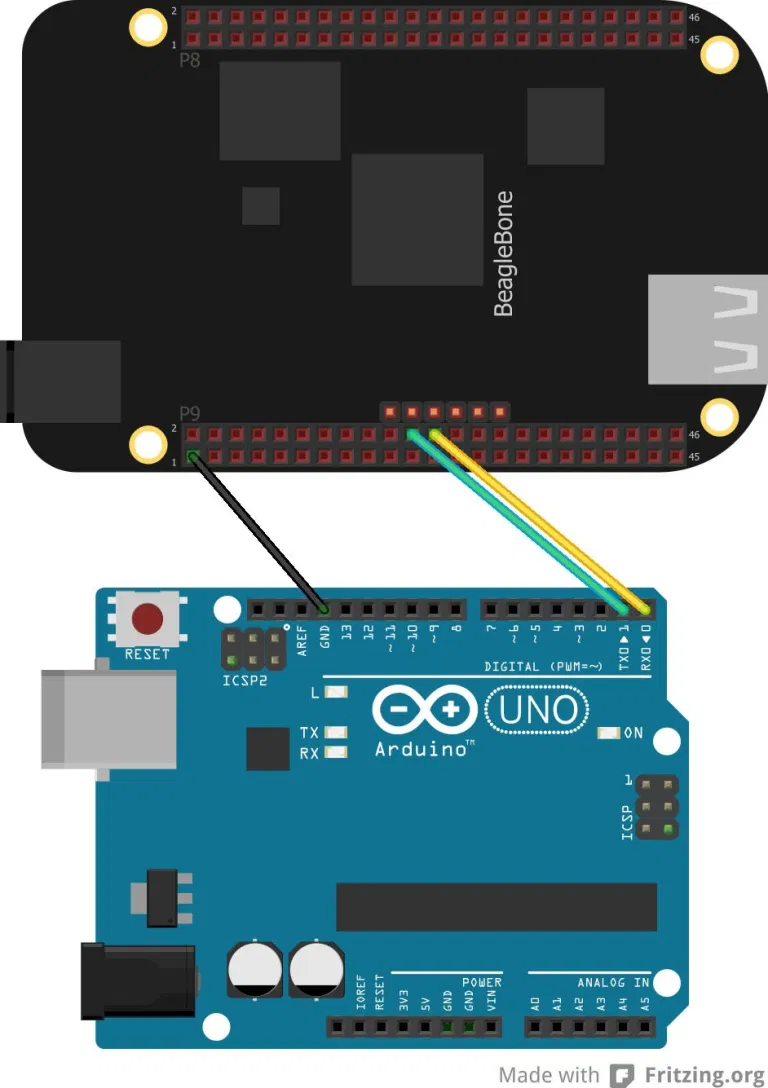
Write API Key – A 16-digit code that allows an application to write data to a channel.

Read API Key – A 16-digit code that allows an application to read the data stored in a channel.

To Write API Key: Select Channels, Select the Channel to update, Select Manage API Keys

Data Import/Export: This tab enables you to import and export channel data.

**Interfacing beagle bone and Arduino uno:**



The picture above specifies P9\_24 and P9\_26 as the transmitter and receiver pins for the serial port. you need the device tree overlays BB-UART1 and BB-UART2 for the serial ports.

#### Creating Device Tree Binary Object

DT overlays allow optional external hardware to be described and configured, and they also support parameters for more control. Next you need to compile it to create the binary object. dtbo. Here's how to do that: Alright, the next thing we need is an app that will allow us to communicate with the serial port. Minicom is that app. Minicom is a serial port communications program. Everything we type on the minicom screen also appears on the Arduino IDE's serial monitor.

Install and start minicom (apt-get install minicom) and launch it in setup mode (minicom-s). Once minicom is installed, fire it up using: minicom-b 9600 -D/dev/tty01

-b is the command line option for specifying the baud rate. We can see what we are typing on the minicom terminal. To do that, just hit CTRL A then E. To exit minicom, hit CTRL A again then X.

**References:**

[**https://www.circuitbasics.com/how-to-set-up-an-lcd-display-on-an-arduino/**](https://www.circuitbasics.com/how-to-set-up-an-lcd-display-on-an-arduino/)

[**https://www.arduino.cc/en/Tutorial/LibraryExamples/LiquidCrystalDisplay**](https://www.arduino.cc/en/Tutorial/LibraryExamples/LiquidCrystalDisplay)

[**https://www.circuitstoday.com/interface-gsm-module-with-arduino**](https://www.circuitstoday.com/interface-gsm-module-with-arduino)

[**https://lastminuteengineers.com/sim900-gsm-shield-arduino-tutorial/#:~:text=The%20SIM900%20GSM%2FGPRS%20shield%20uses%20UART%20protocol%20to%20communicate,115200bps%20with%20Auto%2DBaud%20detection**](https://lastminuteengineers.com/sim900-gsm-shield-arduino-tutorial/#:~:text=The SIM900 GSM%2FGPRS shield uses UART protocol to communicate,115200bps with Auto-Baud detection)**.**

[**https://microcontrollerslab.com/gsm-module-interfacing-arduino-send-receive-sms/**](https://microcontrollerslab.com/gsm-module-interfacing-arduino-send-receive-sms/)

<http://mrityunjai.in/beaglebone-black-with-thinkspeak/>

<http://beagleboard.org/static/librobotcontrol/networking_wifi.html>

<https://github.com/beagleboard/beaglebone-black-wireless/wiki/Wifi>

<http://www.ijarem.org/papers/v3-i4/version-3/24.IJAREM-B131.pdf>

<https://learn.adafruit.com/introduction-to-the-beaglebone-black-device-tree/device-tree-overlays>https://learn.adafruit.com/dht-humidity-sensing-on-raspberry-pi-with-gdocs-logging/python-setup

<https://help.ubuntu.com/community/Minicom>

<https://www.armhf.com/beaglebone-black-serial-uart-device-tree-overlays-for-ubuntu-and-debian-wheezy-tty01-tty02-tty04-tty05-dtbo-files/>

<https://www.python.org/dev/peps/pep-0008/>

<https://www.perforce.com/resources/qac/misra-c-cpp> <https://www.teachmemicro.com/beaglebone-black-serial-arduino/>

**Environmental, Legal and Ethical Ramifications of final solution:**

**Environment Ramifications:**

* The smart devices used in every house have increased luxurious which leads global warming and UV radiation with ozone layer depletion.
* We also should think from where these does the energy is getting renewable always for these devices. Knowingly or Unknowingly all human beings are draining the ENERGY RESOURCES.
* Disposal of older equipment (e waste) is increased which cause in landfills. According to survey 56.3 billon waste is produced in 2020.
* For the production of new products raw materials are used heavily.

**Legal Ramifications:**

* There will be a copyright or stealing the idea issue which is already existing. The patent rights are a legal issue.
* If the products supplied doesn’t survive till the warrantee period and damage to the kit before delivering.
* Cyber security as there are many hackers who hacks the information through a single link and while using IoT is easier.

**Ethical Ramifications:**

* As all the IoT system or devices are high of cost and no under privileged people can use them and our devices doesn’t work if there is no internet in rural areas.
* Companies will collect information kept in the IoT system or internet and make revenue out of it. There is no guarantee of confidentiality and privacy.
* People who are relied on weather reporting channels and news readers will lose their jobs and leads to unemployment.
* Trust issues arise between people. As these smart systems makes human beings to rely on machines than persons them. They take advantage of weakness and attack them personally.

**References:**

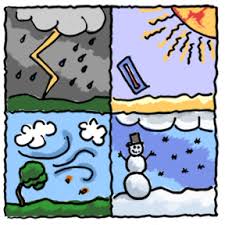
<https://sites.google.com/a/cortland.edu/the-internet-of-things/ethics>

<https://www.researchgate.net/publication/322875867_IoT_ethics_challenges_and_legal_issues>

<https://mse238blog.stanford.edu/2018/07/spak7/how-the-internet-of-things-affects-the-environment/>

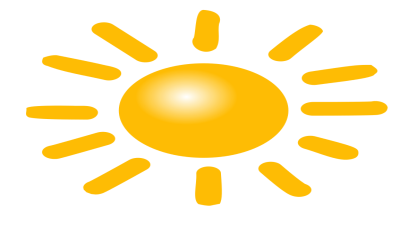
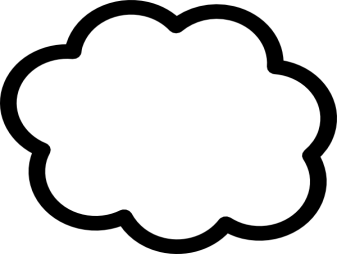
**BOM Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item number #** | **Name of component** | **Part Number** | **How it helps to the project** | **Link to get** |
| **1** | Beagle bone wireless | BBBWL-SC-562-ND | It is the main processing unit used in this project where all the sensors are connected and other devices like arduino is also connected and also for communicating with thingspeak | [**https://www.digikey.ca/en/products/detail/BBBWL-SC-562/BBBWL-SC-562-ND/6211000?itemSeq=354362643**](https://www.digikey.ca/en/products/detail/BBBWL-SC-562/BBBWL-SC-562-ND/6211000?itemSeq=354362643) |
| 2 | Arduino Uno | Arduino Uno - R3 | It is a microcontroller board which has 14 pins and in our project it is used to connect the 16\*2 lcd display and also for gsm module interafcing | sparkfun.com/products/11021 |
| 3 | 16\*2 LCD display | SH-D1602 | It is used to display the weather conditions like temperature and humidity ,rain detection,uv index | <https://www.amazon.ca/DSD-TECH-SH-D1602-Interface-> |
| 4 | Gsm module | Walfrontqb2noh75ka | It is used for sending the text message to the user when there are extreme weather conditions | Amazon, Ali express |
| 5 | DHT11 sensor | Product id :386 | This sensor is used for detecting the temperature and humidity around us | <https://www.adafruit.com/product/386> |
| 6 | Rain drop sensor module | RBsuf-35 | This sensor module helps in detecting the rain drops | [**https://www.robotshop.com/ca/en/raindrop-sensor-module.html?gclid=Cj0KCQiAyJOBBhDCARIsAJG2h5eyWnyntCb95eJO-df6BVz4XyBi8gu2eJYDhX0P99OOoIgfL0KbEMUaAqz-EALw\_wcB**](https://www.robotshop.com/ca/en/raindrop-sensor-module.html?gclid=Cj0KCQiAyJOBBhDCARIsAJG2h5eyWnyntCb95eJO-df6BVz4XyBi8gu2eJYDhX0P99OOoIgfL0KbEMUaAqz-EALw_wcB) |
| 7 | UV index sensor | 1777 | This sensor helps in detecting the uv index in the atmosphere | [**https://www.adafruit.com/product/1777#**](https://www.adafruit.com/product/1777) |
| 8 | Bread board | WYTP-EC01 | This breadboard is used for prototyping before transferring it to pcb and also for checking connections | [**https://www.amazon.ca/WayinTop-Electronics-Electronic-Breadboard-Resistance**](https://www.amazon.ca/WayinTop-Electronics-Electronic-Breadboard-Resistance) |
| 9 | Power supply adaptor | 237-1385-ND | This is used for supplying the power to the beaglebone wireless | [**https://www.digikey.ca/en/products/detail/WSU050-2000/237-1385-ND/3094911?itemSeq=354362444**](https://www.digikey.ca/en/products/detail/WSU050-2000/237-1385-ND/3094911?itemSeq=354362444) |
| 10 | Digital multimeter | BTMETER BT-39C | This is used for checking the current flow and volatage | <https://www.amazon.ca/BTMETER-BMT-BT-39C-Multimeter-> |
| 11 | Jumper wires | WYTP-EC01 | The jumper wires are used for connecting the sensors and also forother hardware device connections like arduino and lcd | [**https://www.amazon.ca/WayinTop-Electronics-Electronic-Breadboard-Resistance**](https://www.amazon.ca/WayinTop-Electronics-Electronic-Breadboard-Resistance) |
| 12 | Sd card | SanDisk | Sd card is used for downloading the latest image of debian and then to transfer it to the beaglebone wireless | <https://www.walmart.ca/en/ip/2-Pack-16GB-Sandisk-Ultra-Plus-Class-10> |
| 13 | Soldering kit | KLARYTYMA | The soldering kit is used for soldering the hardware components on the pcb board after successfully prototyping the project | <https://www.amazon.ca/Soldering-Electronics-Welding-Switch-Tweezers> |
| 14 | Zero PCB | Eiechip® | Once the prototyping is done in the breadboard then we can transfer the design to the pcb board and solder all the components | <https://www.amazon.ca/Eiechip%C2%AE-Prototype-Universal-Electronics-Protoboard> |

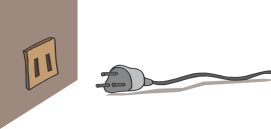
**** 



**Engineering drawing of iot based weather station**



 **C:\Program Files (x86)\Microsoft Office\MEDIA\CAGCAT10\j0293828.wmf** 

**Edit-Test-Debug Cycle implementation**

All embedded systems need some software for their functioning. For developing this software, a

Number of different tools is needed. These tools include editor, compiler, assembler and

Debugger. There are five approaches:

1. Using a Target System 2. Using Emulator for Target System 3. Using Target Processor and

ICE 4. Using a Simulator for Hardware 5. Using IDE or Prototyping Tool

The host machine is the machine on which you write and compile programs. All of the

Development tools are installed on this machine. Most of the time this is your PC or workstation.

Your compiler is built to run on this machine.

A **target machine** may be another general-purpose computer, a special-purpose device

Employing a single-board computer or any other intelligent device. Usually the target machine is

Not able to host all the development tools. That is why a host computer is used to complete

Development work.

**Editor**

The first tool you need for Embedded Systems Software Development Tools is text editor. This is where you write the code for your embedded system. The code is written in some programming language. Most commonly used language is C or C++.The code written in editor is also referred to source code.Nano is an easy to use text editor, especially for both new and advanced Linux users.

**Emulator**

In-circuit emulators In-circuit emulators are devices that are used to emulate a CPU. Using an incircuit emulator, you can start and stop execution of a program as you wish. An in-circuit

emulator provides you access to real hardware for testing purpose.This is in contrast to a

software emulator, where the CPU emulator is a program running on your machine, isolated

from the circuit board on which you want to test your software.

Using JTAG or BDM, you don’t need an in-circuit emulator and a debugger running on your

host machine can connect to a target board.Beaglebone is equipped with a built-in JTAG

emulator providing connectivity over the same USB cable that powers the board, eliminating the

need to buy expensive JTAG debugging hardware.The designer can enable a host debug

computer to take complete control of the target system – stopping it, examining and changing

registers and memory locations,resetting, and restarting it – without affecting normal operation in

any way.But we are not going to use these for this project.

**Simultor:**

It is a program or machine that simulates real –life situations and creates a virtual version of it.

But we are not going to use it in our project.

**IDE**

Software developers use it to find the bugs, analyze the bugs and enhance the quality and

Performance of the software. The standard debugger tool or the debug mode of the Integral

Development Environment (IDE) helps determine the code’s logging and error messages.

An Integrated Development Environment is software that contains all the necessary tools

required for embedded software developments it is very helpful to have software that can

provide all of the necessary tools from writing to testing of your code, in one package.An IDE

normally consists of a code editor, compiler and a debugger.

Other examples of IDE that are common are:Eclipse,Arduino software.

Eclipse

Eclipse IDE is cmmonly used for the projects. Eclipse is a modern IDE supporting many types of

languages and setups .Initially, the Eclipse integrated development environment was created for

Java applications. Nevertheless, Eclipse can work with other programming languages (Ada,

ABAP, C, C++, C#, Python, PHP, etc.) via plug-ins.

Arduino IDE

The open-source IDE Arduino software helps you to create program providing all the necessary

embedded software tools.

It provides a range of features and libraries that make the life of embedded programmers easier.

The main advantages are:

•Ready-to-use boards with all needed components

•Libraries with examples of codes

•Open-source and extensible hardware and software

•Cross-platform support for Windows, Mac OS and Linux

•Access to a large community

•Easy to learn and use

GDB Tool: It will be used in the project for debbuging.GNU debugger, more commonly known

as GDB, is a common debugger which is pre-installed in all Linux systems if not, it is necessary

to download the GCC compiler package. It is one of the most popular debuggers out there.Many

other debuggers are also built on this debugger. The GNU debugger is started using the gdb

command.You can download the latest version from ftp://ftp.gnu.org/gnu/gdb directory.

A compiler is used when you are done with the editing part and made a source code.Source code

is written in a high-level programming language.It is used for transforming the code into a lowlevel

machine language code — the one that a machine can understand. GNU ompile will be

used in the project for compiling.

GNU Cross-Platform Development Toolchain:A toolchain is a set of software tools needed to

build computer software.

The C library

The standard C library most often used with current day Linux systems is thein GNU C library,

often abbreviated as glibc. glibc is a portable, high-performance C library.These libraries provide

critical APIs and these APIs include such foundational facilities

as open, read, write, malloc, printf, getaddrinfo, login, exit and more.It aims to follow all relevant

standards including ISO C11, POSIX.1-2008, and IEEE 754-2008.The GNU C Library releases

every 6 months. The current stable version of glibc is 2.33, released on February 1st, 2021.You

can download the latest version of GCC from ftp://ftp.gnu.org/gnu/gcc/.Many programming

languages use the GNU C Library indirectly including C#, Java, Perl, Python, and Ruby

(interpreters, VMs, or compiled code for these langauges use glibc directly).

**References:**

<https://www.elprocus.com/what-is-debugging-types-techniques-in-embedded-systems/>

<https://www.sam-solutions.com/blog/top-ten-embedded-software-development-tools>

<https://www.theengineeringprojects.com/2016/11/top-10-embedded-systems-softwaredevelopment-tools.html>

<https://www.gnu.org/software/libc/>

<https://ptgmedia.pearsoncmg.com/imprint_downloads/informit/perens/0130091154.pdf>

<https://learning.oreilly.com/library/view/building-embedded-linux/9780596529680/ch04.html>

<https://beagleboard.org/static/Docs/ccs-jtag-simple.htm>

<https://www.ics.com/blog/exploring-simulators-and-emulators-embedded-development>

<https://www.dictionary.com/browse/simulator>