"REAL TIME DATA ACQUISITION IN AN INTERNET OF THINGS BASED CONTROL UNIT"

THESIS REPORT

***Submitted in view of the** fulfillment for the requirements of the award of degree*

of

BACHELOR OF ENGINEERING

in

ELECTRONICS & COMMUNICATION ENGINEERING

Submitted By

PRANAV JAYARAM: 1DA15EC105

DR. AMBEDKAR INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

2017-18

ABSTRACT

This project aims to design and implement a novel real-time data acquisition and batch-processing technique on an Internet of Things (IoT) based customizable control unit. The scope of the project is to incorporate the use of interconnectivity between devices, which generate data and to process the generated data onboard with the use of a real-time batch data processing algorithm. The processed data is then logged to the control unit, which has a continuous feedback to an IOT based platform. This platform includes online and offline nodes, which can monitor and control the devices present in the control unit. The project incorporates the design of a hardware circuitry, which includes the control panel and the processing unit.

The online and offline nodes are the interconnected devices, which use an instance of the same control unit application that can be set up on mobile and wireless devices. The project also aims to design the native control application, which can be deployed on mobile devices.

ACKNOWLEDGMENT

To my eternal friend, late Naveen Kumar N: because I owe it to you. Many Thanks!

To my parents: My perpetual well-wishers: I shall always be heedful to your guidance.

To my forever interested, encouraging and always enthusiastic grandparents: They are always keen to know what I am doing and how I am proceeding, although it is likely that they cannot comprehend what it is all about!. I will always cherish your support and endurance in taking care of me through my thick and thin.

To my late grandfather, H V Nagaraj: I shall always admire your will-power, resolve and do-it-yourself attitude which has inspired me significantly.

I am grateful to my uncle, Vinaya Skanda: who has provided me with not just abundant technical and industry knowledge in life, but also the determination and drive to excel. I am also grateful to my other family members and friends who have supported me along the way.

A very special gratitude goes out to all at the Dept. of Electronics Systems Engineering, Indian Institute of Science and also Dr. Ambedkar Institute of Technology for helping and providing me a plethora of domains to work and excel in. It was fantastic to have the opportunity to do the majority of research in these facilities.

I am thankful to my project guide and advisor, Muralidhara K: Your ever-lasting thirst for knowledge and passion for engineering has rubbed off on me significantly. Your sagacity and discernment has truly influenced my work.

I am also grateful to the following university staff: Late Dr. K M Rajanna, Dr. Jayaramaih G V, Hemalatha K N and Prof. N S Dinesh for your unfailing support and assistance.

And finally, last but by no means least; also to everyone in the batch of 2019, ECE… it was great sharing the laboratories with all of you for the last four years.

Thanks for all your encouragement!

PRANAV JAYARAM

TABLE OF CONTENTS

TABLE OF FIGURES

1 CHAPTER 1: INTRODUCTION

1.1 AIM OF THE PROJECT

The aim of this project is to design and implement a real-time data acquisition and batch-processing algorithm on an Internet of Things based control unit. The fundamental concept of this project is to show the typical use-case applications of Internet of Things (IoT) and how it can help solve day-to-day issues in homes, industries and in many businesses. The project incorporates the use of an embedded system which comprises of a central processor, ancillary controllers and peripheral devices. The project is demonstrates the application of IoT in the field of Home-automation, Industrial IoT, and Smart Monitoring.

The project runs a real time batch processing system on the processor which samples 15 bits of data at about 2000 samples of data per minute. This data is dynamic and customizable to the application in use. The use of DHT's (Digital Humidity and Temperature) sensors in the project is to demonstrate the use of temperature and humidity values to trigger ancillary controllers and to switch peripherals.

The main concept of (IoT) implemented in this project is the interconnection between devices and the control of these interconnected devices through online and offline nodes via the network and through a mobile application. This report elaborates the complete design and implementation of the project.

1.2 PROBLEM STATEMENT

> "One of the main issues in today's world is the requirement of multiple devices to perform multiple tasks, namely switching, monitoring, displaying of data and so on. The unification of all these tasks into a single control unit helps to build efficient systems which consume less power, run more efficient data processing algorithms and are retrofit into many applications such as smart monitoring and surveillance, smart home and industrial automation and so on"

This is an important problem in today's fast-growing (IoT) sector, which requires a robust, efficient and less power consuming system and all to be integrated into one unit that runs a real-time algorithm to schedule multiple such tasks.

1.3 PROJECT SCOPE AND OBJECTIVES

The scope of this project is to develop customizable data acquisition hardware and dedicated standalone controllers which perform unique tasks such as sensor data gathering, switching and inverting, display and multimedia and wireless control. The project also involves building some of the use-case applications using the hardware that can be deployed in industries and smart homes.

The data gathered after a subsequent amount of time can be used as training data for a future Machine learning based application that can also be incorporated into the same.

The full extent of batch processing helps a manufacturing based growth industry that can regularly monitor and control a complete production cycle/ product build.

This project has a product-oriented design that can be tapered to specific industry needs in the future.

The main objective of the project is to design and develop a generic hardware that supports up to 3 IoT based applications such as Industrial IoT, ADAS (Advanced driver assistance system) and Home Automation using a set of 2 standalone controllers with a processor.

Another important objective is to design and develop compatible onboard and mobile applications which run on a wide range of platforms such as Linux, Windows, and Android (Mobile).

2 CHAPTER 2: BLOCK DIAGRAMS AND ILLUSTRATIONS

2.1 OVERVIEW OF THE EMBEDDED SYSTEM

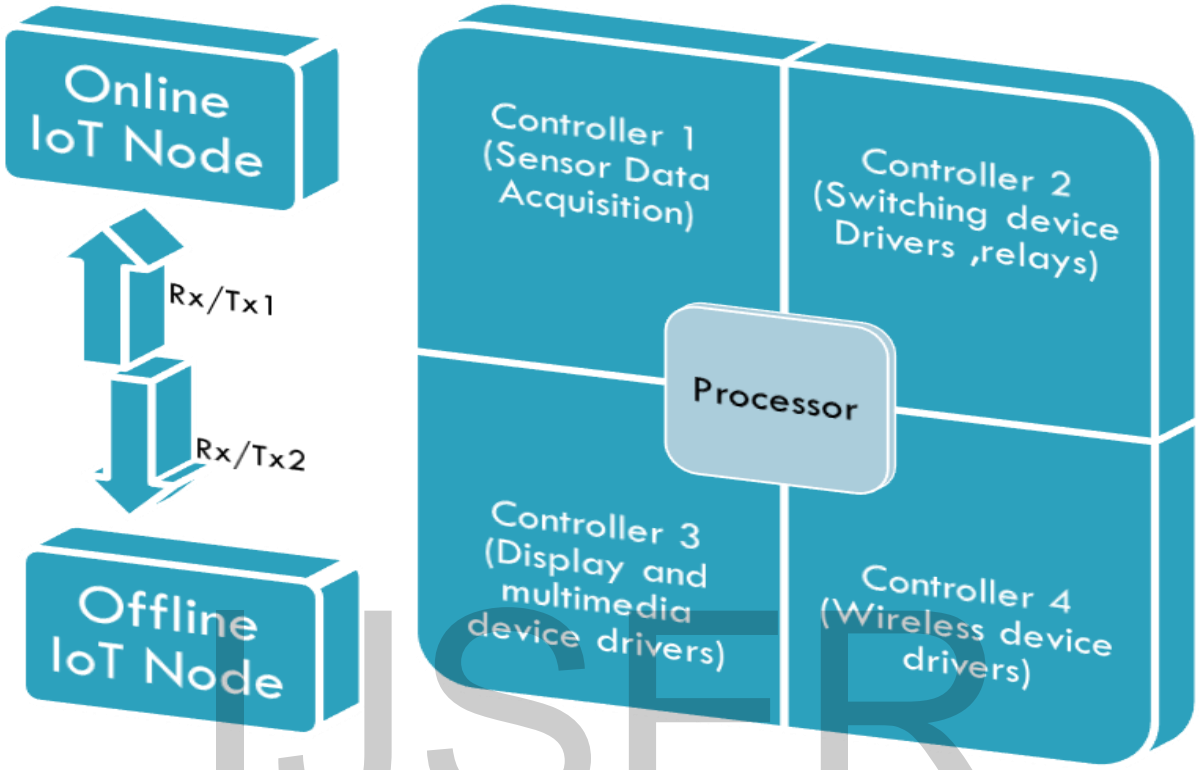


Figure 1-Overview of the Embedded System

1. The Block diagram shows the embedded system block diagram of the real-time systems that comprises the processor and standalone ancillary controller units present.
2. It shows the two Nodes that are present in the control of the IoT unit namely, the online node and the offline node. The Online node is any device that is connected to the same network as that of the embedded system, the offline node is a wireless node that isn't necessarily connected to the same network but, is interfaced with the embedded system via Bluetooth. BLE <4.0.
3. The processor runs a real-time data acquisition algorithm which acquires and stores data locally and can also be accessed from a remote node.
4. The controllers run independent firmware's pertain to the real-time working on tasks such as sensor data acquiring, Switching (relay control) and Display and multimedia operations.
5. The mobile offline node has a control application developed for the native Android OS.

2.2 FUNCTIONAL BLOCK DIAGRAM

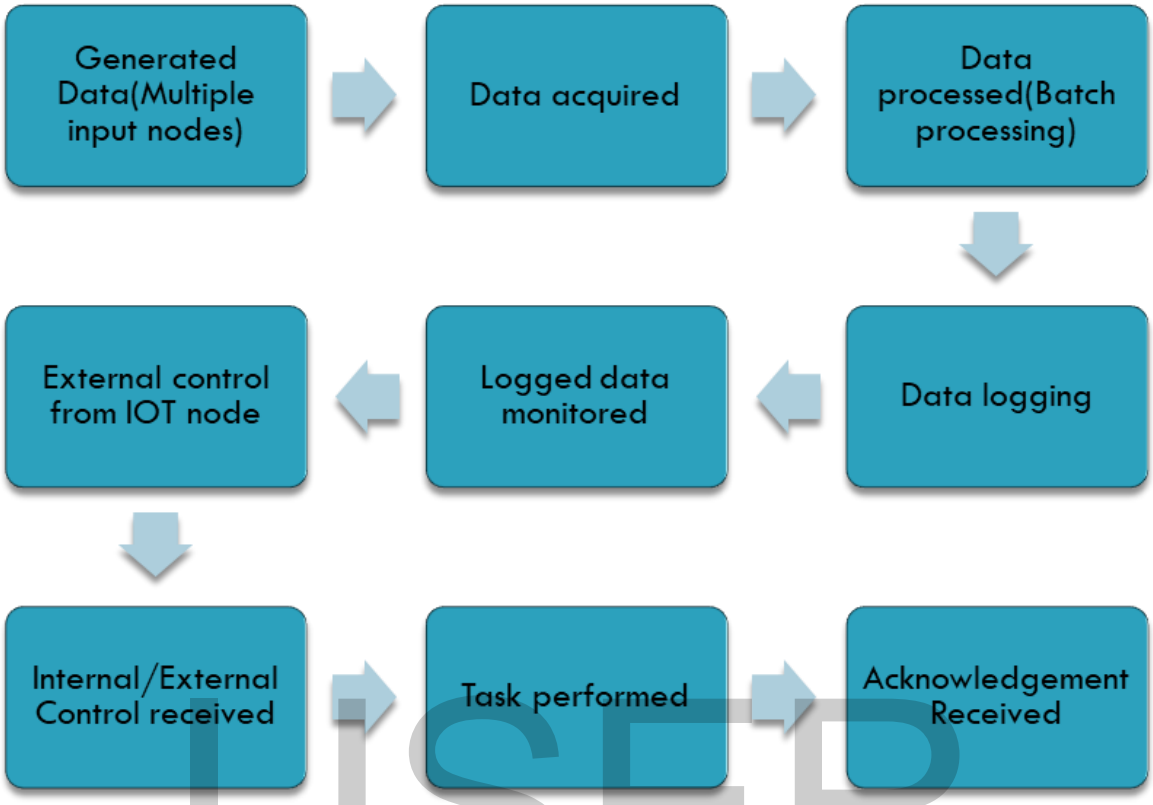


Figure 2-Functional Block diagram

1. The project involves the use of the adjacent functional blocks where an IoT node (Online/Offline) is used as a master device which targets controlling the main processor running the batch processing algorithm.
2. The concept of IoT is typically used by the processor, which interconnects with multiple subsystems and the master node.
3. The Data generated from the standalone controllers is acquired by the processor via the batch processing algorithm and available for future reference.
4. The same processing system acknowledges an event trigger from an offline or an online node. On receiving the event trigger, the processor immediately understands the request and performs the desired task and sends an acknowledgment of the same.
5. This is the real-time system that is always running in the background of the Real-time operating Kernel of the processor.
6. These functions performed by the processor are also logged. The standalone controllers are also equipped with an onboard EEPROM which stores the switching tasks performed.

2.3 COMPLETE USE-CASE SCHEMATIC

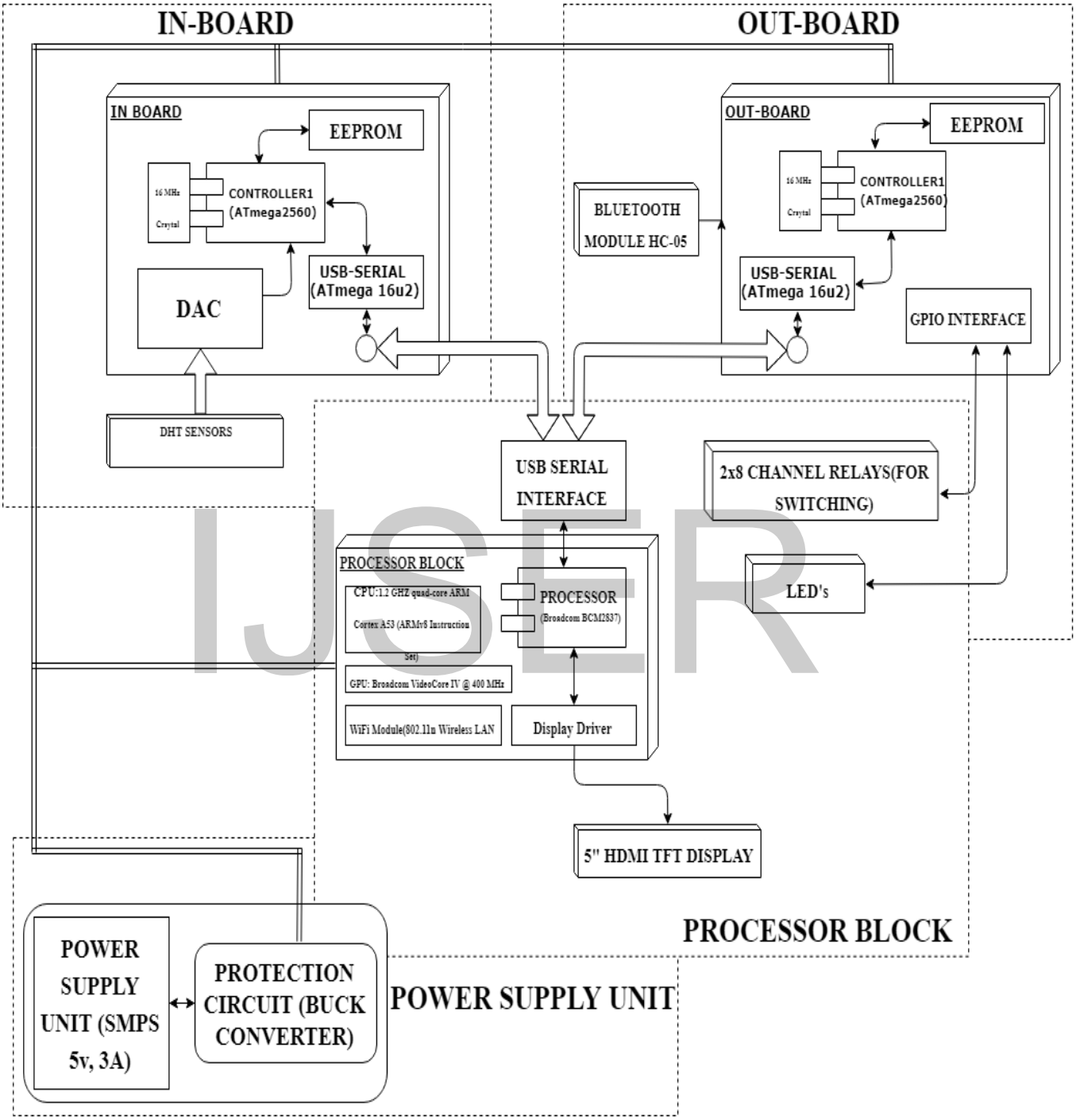


Figure 3-Complete Circuit schemeatic

The figure depicts the complete use-case schematic of the project which is a combination of the Power supply unit, the Processor block, In-board and the Out-board.

3 CHAPTER 3: HARDWARE AND SOFTWARE DESIGN AND DEVELOPMENT

3.1 HARDWARE BLOCK

The Hardware block consists of the following blocks which are the different components present in the embedded system.

Sl. No	FUNCTIONAL BLOCK
3.11	PROCESSOR BLOCK
3.12	IN-BOARD
3.13	OUT-BOARD
3.14	POWER SUPPLY AND PROTECTION CIRCUIT
3.15	PERIPHERAL COMPONENTS

3.1.1 PROCESSOR BLOCK

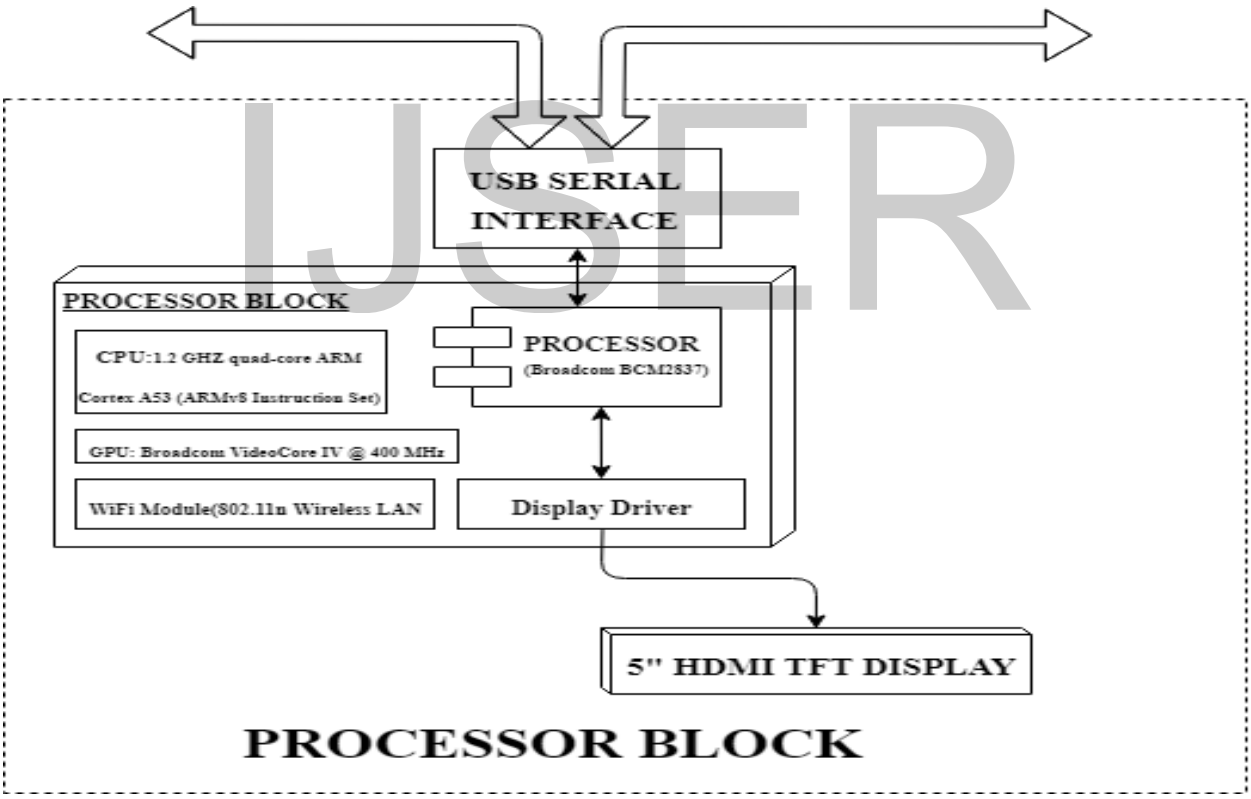


Figure 4-Processor Block

PROCESSOR BLOCK SPECIFICATIONS:

System on Chip: Broadcom BCM2837

CPU: 4× ARM Cortex-A53, 1.2GHz

GPU: Broadcom Video Core IV

RAM: 1GB LPDDR2 (900 MHz)

Networking: 10/100 Ethernet, 2.4GHz 802.11n wireless

Bluetooth: Bluetooth 4.1 Classic, Bluetooth Low Energy

Storage: micro SD

GPIO: 40-pin header, populated

Ports: HDMI, 3.5mm analog audio-video jack, 4× USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)

Interfaces Used: 2 USB-Serial ports are used to connect to the IN-BOARD and OUT-BOARD. 5" TFT display connected via HDMI port

3.1.2 IN-BOARD

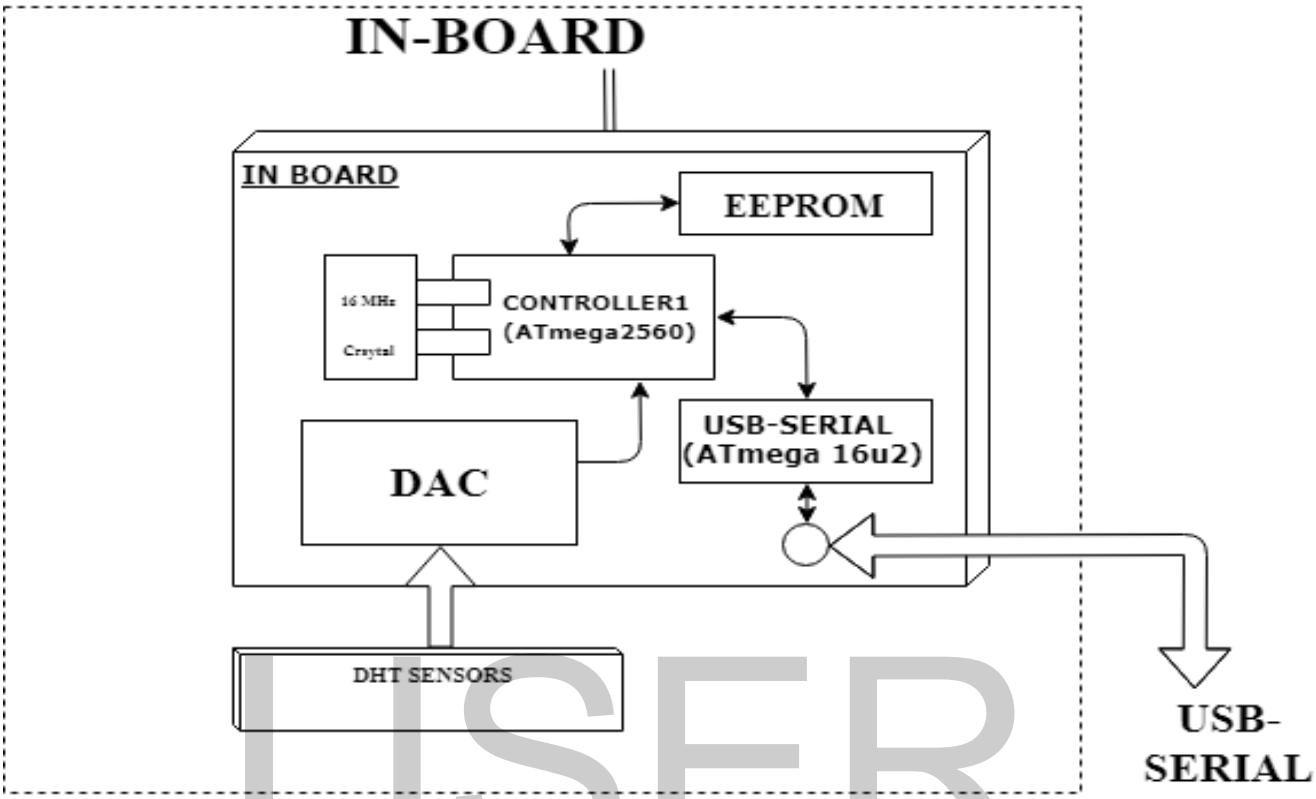


Figure 5-In-Board circuit

1. The IN-BOARD has an ATmega2560 microcontroller.
2. It has 54 digital input/output pins (of which 15 can be used as PWM outputs)
3. 16 analog input pins
4. 4 UARTs (hardware serial ports)
5. A 16 MHz crystal oscillator
6. A USB connection
7. A power jack
8. An ICSP header
9. 1 reset button.
10. 5 DHT (Digital Humidity and Temperature) Sensors connected to the digital PWM pins as sensor data Input port. The sensor data sampled from each DHT is 2 bits. So, 10 bits of data is sampled at 2000 samples/minute, resulting in 20000 bits of samples/minute.

3.1.3 OUT –BOARD

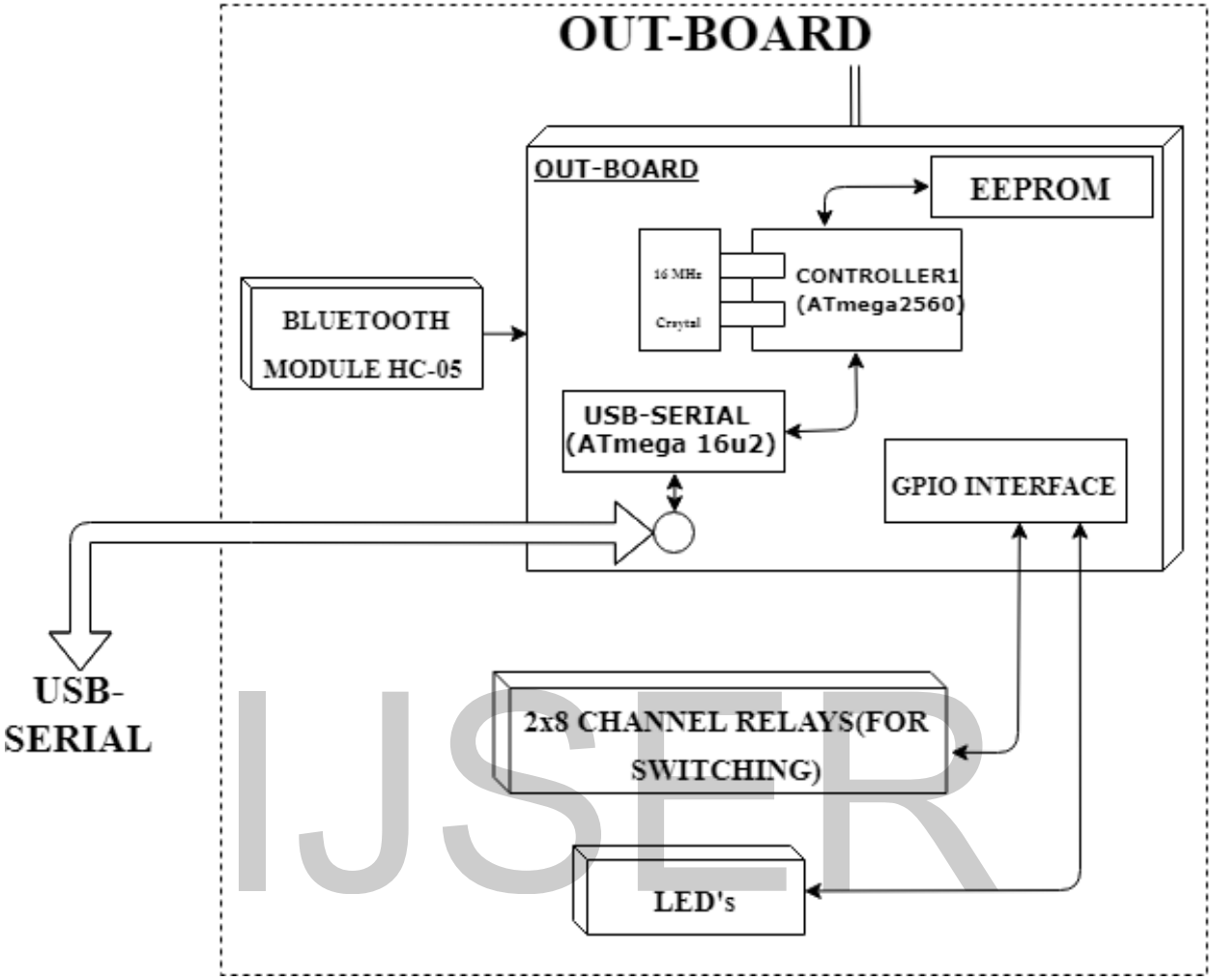


Figure 6-Out-Board circuit

The OUT-BOARD has the same specifications as that of the IN-BOARD, The difference is that it is used to perform different tasks.

1. The OUT-BOARD is connected to the peripheral devices that must be accessed using the processor and the online and offline IoT nodes.
2. The Bluetooth module HC-05 is also connected to the OUT-BOARD for wireless communication from mobile devices.
3. The peripherals connected to the OUT-BOARD are, LED’s, 2x8 Channel relay module, and the USB-Serial port.

3.1.4 POWER SUPPLY AND PROTECTION CIRCUIT

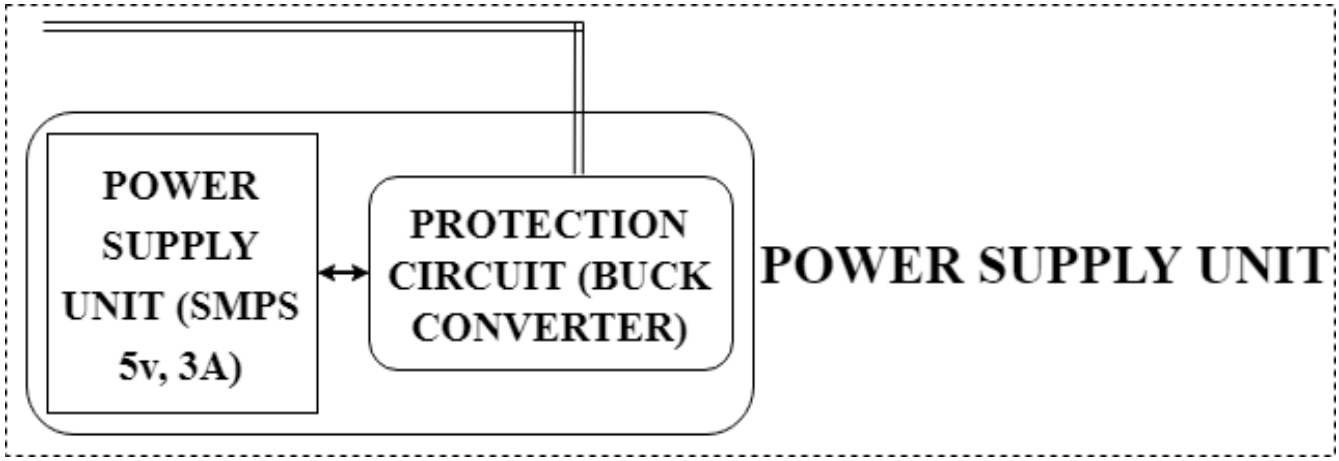
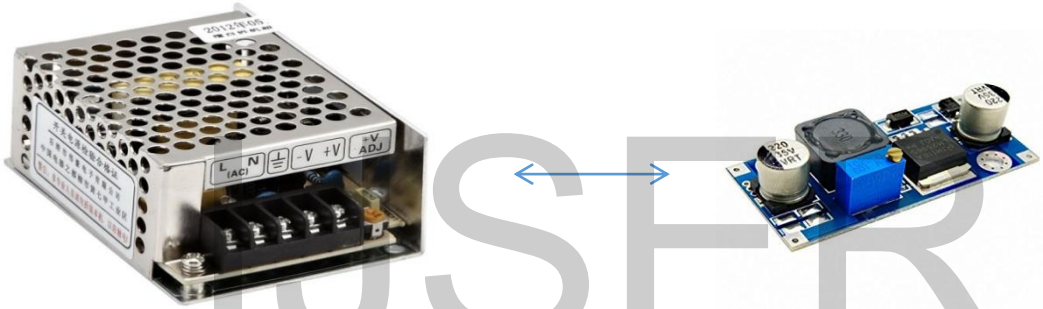


Figure 7-Power supply circuit



POWER SUPPLY SPECIFICATIONS:

DC VOLTAGE	**5V**
RATED CURRENT	**3A**
CURRENT RANGE	**0-3A**
RATED POWER	**15W**
RIPPLE AND NOISE(max)	**80mVp-p**
VOLTAGE ADJUSTMENT RANGE	**4.75-5.5V**
LINE REGULATION	**+/- 0.5%**
LOAD REGULATION	**+/- 1.5%**

3.1.5 PERIPHERAL COMPONENTS

Sl. No	COMPONENT
3.1.5.1	HC-05 Bluetooth Module (Wireless offline communication)
3.1.5.2	LED's
3.1.5.3	8 Channel Relay module 1
3.1.5.4	8 Channel Relay module 2

INTERFACING ATMEGA2560 WITH HC-05 BLUETOOTH MODULE, RELAY AND LED:

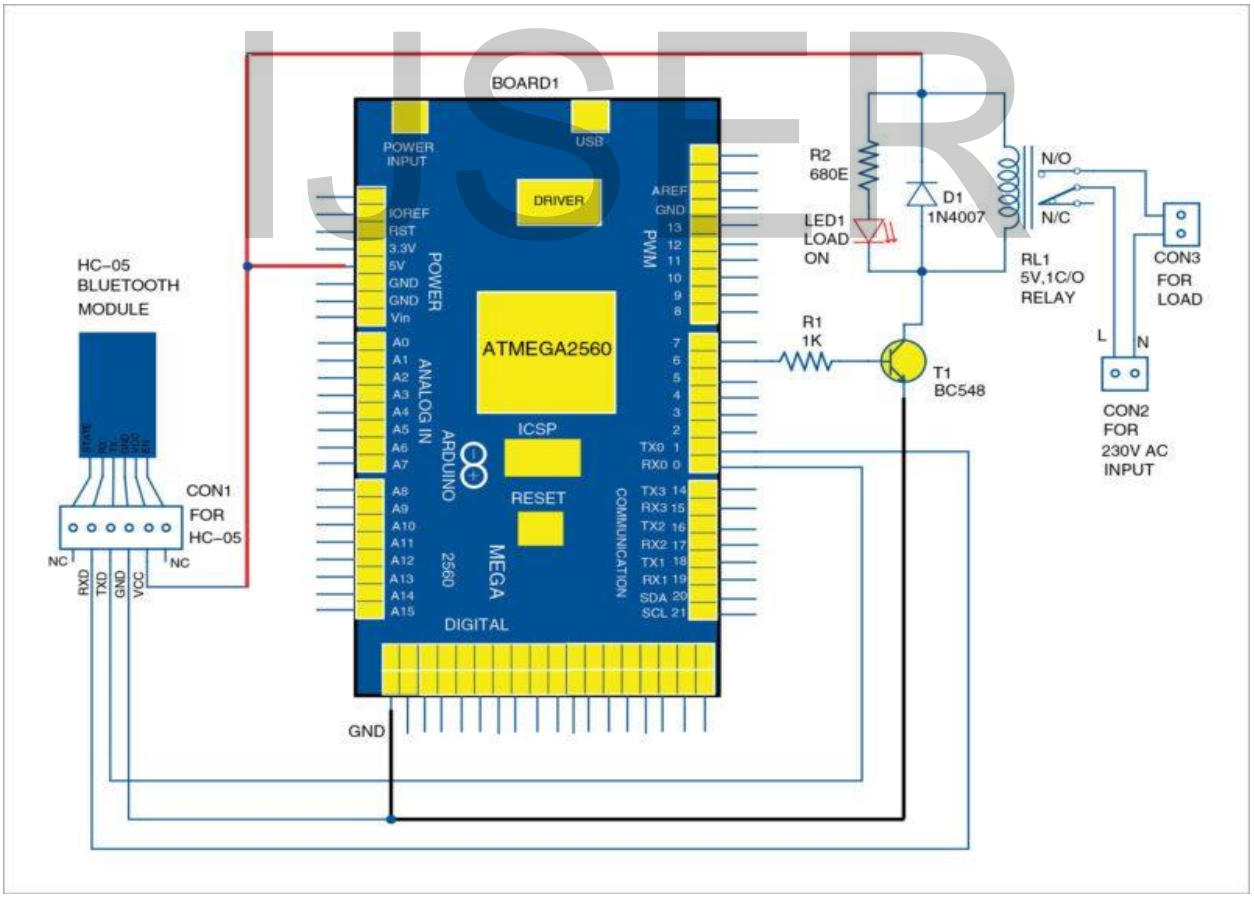


Figure 8-Atmega2560 interfacing

3.2 COMPONENT SPECIFICATIONS

The datasheet and specifications of various components used in the embedded system are mentioned below.

Sl. No	COMPONENT
3.2.1	Broadcom BCM2873 processor
3.2.2	ATMega 2560 controller
3.2.3	5” TFT HDMI display
3.2.4	DHT (Digital Humidity and Temperature) Sensor
3.2.5	Bluetooth module (HC-05)

3.2.1 Broadcom BCM2873 Processor (Raspberry pi 3)

Figure 9-BCM2873 Block diagram

SPECIFICATIONS:

CPU: Quad-core Cortex-A57 64-bit ARMv8 up to 1.8GHz – Two-stage address translation for virtualization with I/O memory management unit (MMU) – 48 KB I-cache and 32 KB D-cache per core – 2 MB shared L2 cache.

Memory: 1MB of platform SRAM – 72-bit wide DDR3/3L/4 memory controller with error correction coding (ECC) (64b data, 8b ECC) up to DDR4-2400.

Flash interface: 4x UART, two BSC, two SPI, MDIO, GPIO (32), Local Bus, TDM.

3.2.2 ATMega2560 Controller

Figure 10-ATMega2560 Pin Layout

SPECIFICATIONS:

1. Advanced RISC Architecture
2. Up to 16 MIPS Throughput at 16 MHz
3. 256K Bytes of In-System Self-Programmable Flash
4. 8K Bytes RAM
5. 4K Byte Internal SRAM
6. 86 Programmable I/O Lines arranged in nine 8 bit ports
7. In-System Programming by On-chip Boot Program
8. JTAG
9. 16-channel, 10-bit ADC
10. Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
11. Four 16-bit Timer/Counter with Separate Prescalers, Compare Mode, and Capture
12. Real Time Counter with Separate Oscillator
13. Twelve 16 bit and Four 8 bit PWM Channels,
14. Four Programmable Serial USART
15. Eight external interrupts
16. Master/Slave SPI Serial Interface
17. Byte-oriented Two-wire Serial Interface
18. Programmable Watchdog Timer with Separate On-chip Oscillator
19. 100 pin TQFP package

3.2.3 5” TFT HDMI DISPLAY

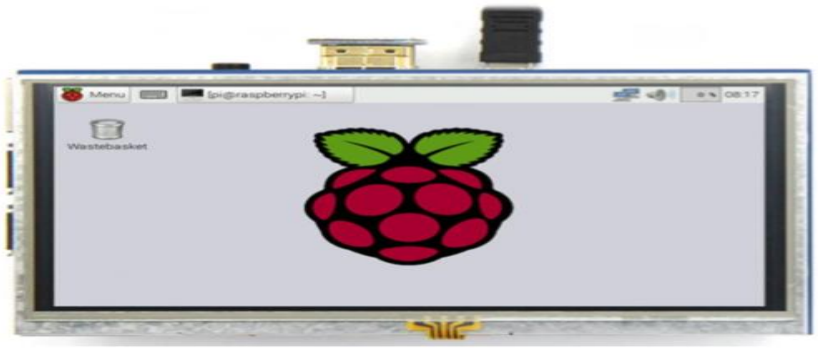
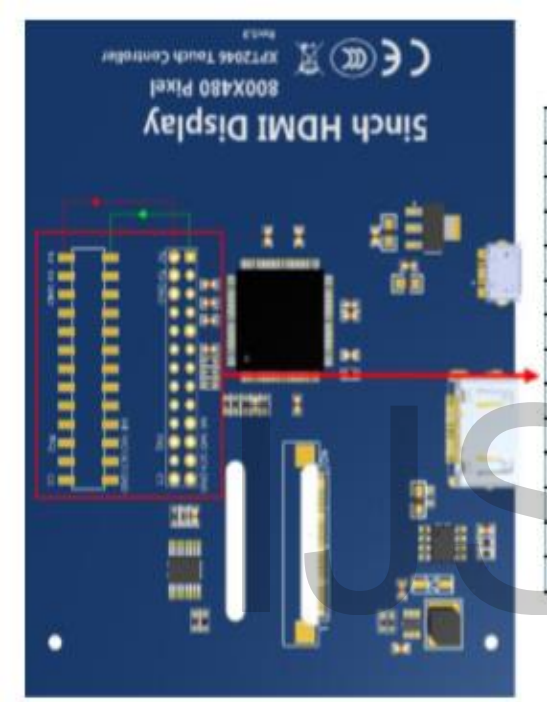


Figure 11-5" TFT display panel

The HDMI display is a resistive touchscreen panel that serves as both display and touch interface. It is a cheap and inexpensive device available in any standard electronics market for a price of 3000 Rs. It can be connected as a display interface to any Raspberry Pi or a development board with HDMI interface.

The following image shows the display along with the pin layout:



Description	Pin	NO.	NO.	Pin	Description
Power input(5V)	5V	2	1	3.3V	NC
Power input(5V)	5V	4	3	SDA	NC
Power GND	GND	6	5	SCL	NC
NC	TX	8	7	P7	NC
NC	RX	10	9	GND	Power GND
NC	P1	12	11	P0	NC
Power GND	GND	14	13	P2	NC
NC	P4	16	15	P3	NC
NC	P5	18	17	3.3V	NC
Power GND	GND	20	19	MI	TP SPI Bus input(MOSI)
TP Interrupt	IRQ	22	21	MO	TP SPI Bus output(MISO)
NC	CE0	24	23	SCK	TP SPI Bus Clock(SCLK)
TP Chip Sellect	TCS	26	25	GND	Power GND

Figure 12-5" Display Pin layout

SPECIFICATIONS:

1. 5'' standard display, 800 × 480 resolution
2. With a resistive touch screen, support touch control
3. Support backlight control alone, the backlight can be turned off to save power
4. Supports standard HDMI interface input, compatible with and can be directly inserted with Raspberry Pi (3rd, 2nd, and 1st generation)
5. Can be used as general-purpose-use HDMI monitor, for example: connect with a computer HDMI as the sub-display (resolution need to be able to force output for 800 x480)

6. Used as a raspberry pie display that supports Raspbian, Ubuntu, Kodi, win10 IOT(resistive touch)
7. Work as a PC monitor, support XP,win7, win8, win10 system(do not support touch)
8. CE, RoHS certification.

INTERFACING RASPBERRY PI AND DISPLAY:

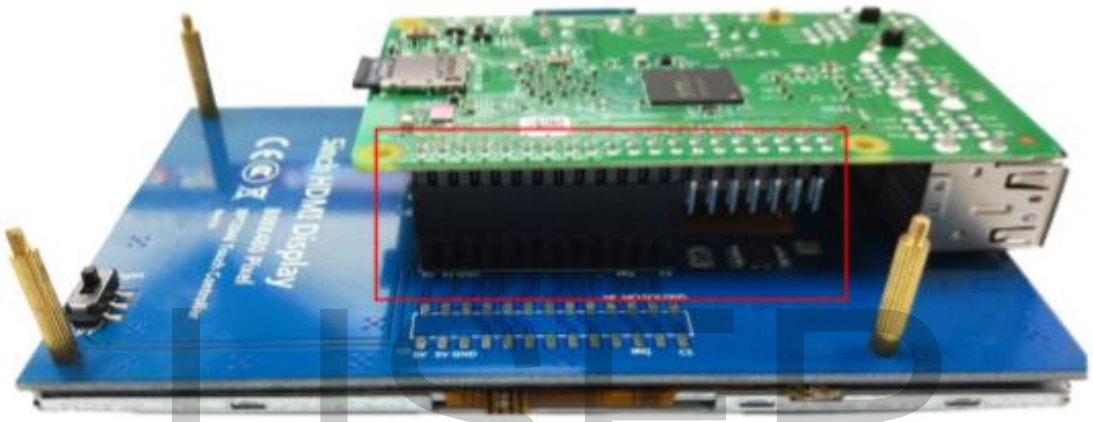


Figure 13-Interfacing Display and Raspberry Pi

1. The 5" HDMI display connects to 13 pins of the Raspberry Pi GPIO pins and also connects to the HDMI port of the Raspberry Pi. It also has mounting holes which can be used to fix the Display to the Raspberry Pi using standard 2mm mounting screws.
2. USB Interface: Gets 5V Power from USB, If -13*2 Pin Socket has been connected, that this USB interface can be No Connect.
3. HDMI Interface : for HDMI transmission. Backlight Power switch Controls the backlight turned on and off to save power.
4. 13*2 Pin Socket : Gets 5V Power from raspberry Pi to LCD, at the same time transfers touch signal back to Raspberry Pi.
5. Extended interface : Extended 13*2 Pin Socket signal Pin-to-Pin.

3.2.4 DHT (DIGITAL HUMIDITY AND TEMPERATURE) SENSOR

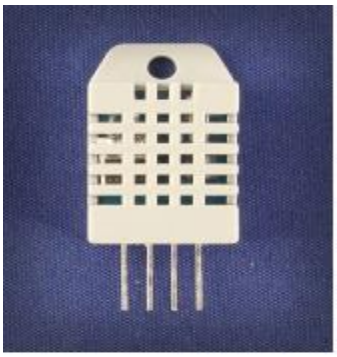


Figure 14-DHT Sensor

The DHT sensor is a cheap and inexpensive Digital humidity and temperature sensor which works on the principle of using a capacitive based dielectric for relative humidity and temperature measurement. The working of the sensor corresponds to the time period of sampling in which the sensor is calibrated to sense the relative humidity and temperature.

SENSOR WORKING DIAGRAM:

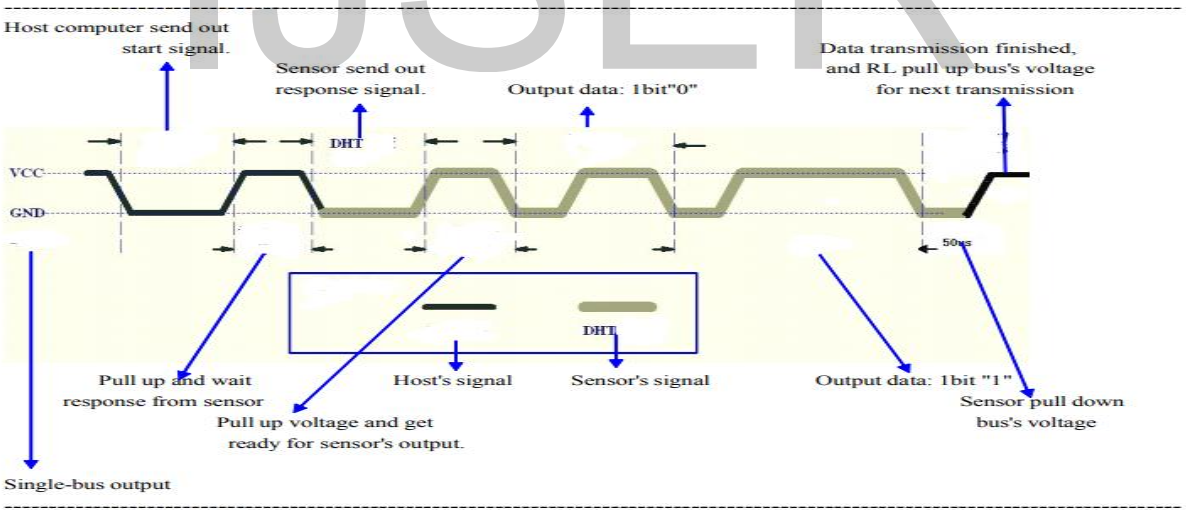


Figure 15-Sensor working diagram

As seen above, the DHT sensor sends a signal and awaits a response from the capacitive material, the time duration taken for a response from the sensor is the digital equivalent to the relative humidity and temperature.

TECHNICAL SPECIFICATIONS:

Every sensor of this model is temperature compensated and calibrated in accurate calibration chamber and the calibration-coefficient is saved in type of program code in OTP memory when the sensor is detecting, it will cite coefficient from memory.

Small size, low consumption and long transmission distance (20m) enable the DHT11 to be suited to all kinds of harsh application occasions. Single-row packaged with four pins, making the connection very convenient.

Model	**DHT11 / DHT22**
Power Supply	**3.3-6V DC**
Output Signal	**Digital Signal via a single bus**
Sensing element	**Polymer humidity capacitor & DS18B20 for detecting temperature.**
Accuracy	**Humidity +-2%RH(Max +-5% RH); Temperature +-0.2Celsius**
Resolution	**Humidity 0.1%RH; Temperature 0.1Celsius**
Repeatability	**Humidity +-1%RH; temperature +-0.2Celsius**
Humidity hysteresis	**+-0.3%RH**
Long-term Stability	**+-0.5%RH/year**
Sensing period	**Average: 2s**
Interchangeability	**Fully interchangeable**

This sensor is connected to the IN-board and the digital bits are samples and also stored in the EEPROM of the controller. The DHT sensors work as inputs in a typical Industrial IoT setup where Real-time data must be sampled and specific parameters must be monitored.

It can be customized and changed according to the required specifications and different sensors can also be used in place of the DHT sensor.

The main advantage of using the DHT sensor is the acquisition of 2 bits of information in one clock cycle of the microcontroller. So, more data can be sampled for the same number of clock cycles as that of other sensors.

3.2.5 Bluetooth module HC-05

HC-05 is an inexpensive Bluetooth module which can easily be interfaced with microcontrollers such as ATMega2560 and is easily available. The module runs a low power wireless BLE RF Transceiver which makes it an energy efficient solution for the embedded system design.

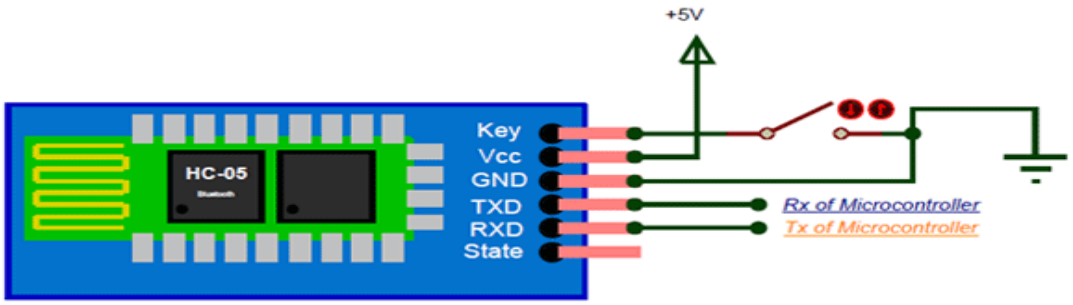


Figure 16-Bleutooth module HC-05

PIN NUMBER	PIN NAME	DESCRIPTION
1	Enable / Key	This pin is used to toggle between Data Mode (set low) and AT command mode (set high). By default, it is in Data mode
2	Vcc	Powers of the module. Connect to +5V Supply voltage
3	Ground	Ground pin of the module, connect to system ground.
4	TX – Transmitter	Transmits Serial Data. Everything received via Bluetooth will be given out by this pin as serial data.

5	RX – Receiver	Receive Serial Data. Every serial data given to this pin will be broadcasted via Bluetooth
6	State	The state pin is connected to onboard LED, it can be used as a feedback to check if Bluetooth is working properly.
7	LED	Indicates the status of Module • Blink once in 2 sec: Module has entered Command Mode • Repeated Blinking: Waiting for connection in Data Mode • Blink twice in 1 sec: Connection successful in Data Mode
8	Button	Used to control the Key/Enable pin to toggle between Data and command Mode

3.3 HARDWARE DESIGN USING EAGLE

The In-Board and the Out-Board are designed in such a way that both the logical firmware's can be implemented on the same PCB. This PCB is designed using the EAGLE software. To design the PCB layout, the schematic is first made which is corresponding to the controller and peripheral component pinout. To design the schematic, first a working model must be made on a breadboard and the concept must be verified.

The use of designing and fabricating PCB's is so that the project can also be made commercial after customizing to the specific application such as home automation, where switching and display standalone controllers are required. So, only the Out-Board and the processor unit can be used. For an Industrial IoT application, the data acquisition, switching, and display controllers are required. So, the In-Board, Out-board and the processor boards can be used.

This brings about the customizability in the application at hand. The same firmware can be dumped onto the respective controller boards by altering the pinouts and the same application software's can be used by increasing or decreasing the parameters of switching and monitoring.

On creating the schematic circuit of the logic, the same schematic can be converted into a board file and here, the component layout and placement must be made. For doing so, we must first have an idea of the actual component dimensions that need to be taken into account. The Arduino Mega Board is used as a substitute for the ATMega2560 and a Raspberry Pi 3 is used as the main processor unit onto which the display unit fits perfectly as it is a Raspberry Pi specific module. The Arduino Mega pinouts are taken as a footprint file in the design of the controller card. A footprint is basically a vector representation of the number of pins, its placement and pitch distances (Pitch: Least distance between the pins of the controller).

After finalizing all the peripheral footprints, such as the HC-05 module footprint which is a 6 pin through-hole standard 2.54 mils pitch. For conversion, 1mils is equivalent to one-thousandth of an inch which is 0.00254cm. It must be taken care so as to choose the perfect pitch from the component datasheet if not, the component cannot be mounted on the board post-fabrication.

3.3.1 BOARD LAYOUT AND FABRICATION

SCHEMATIC:

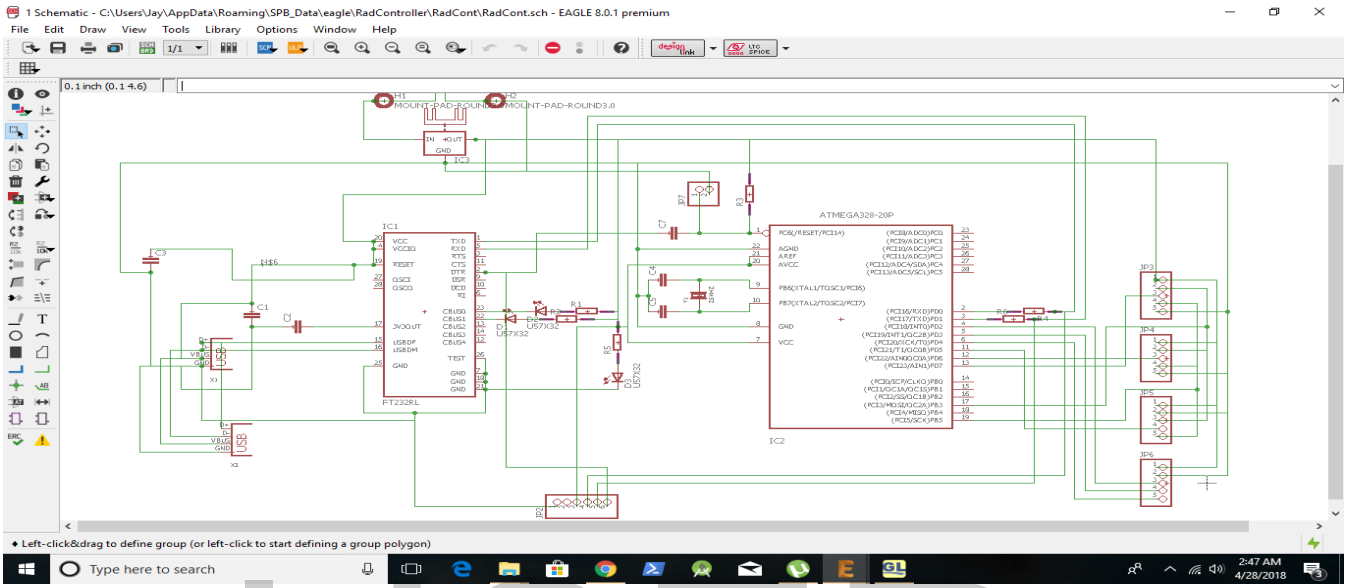


Figure 17-Board Schematic

COMPONENT PLACEMENT:

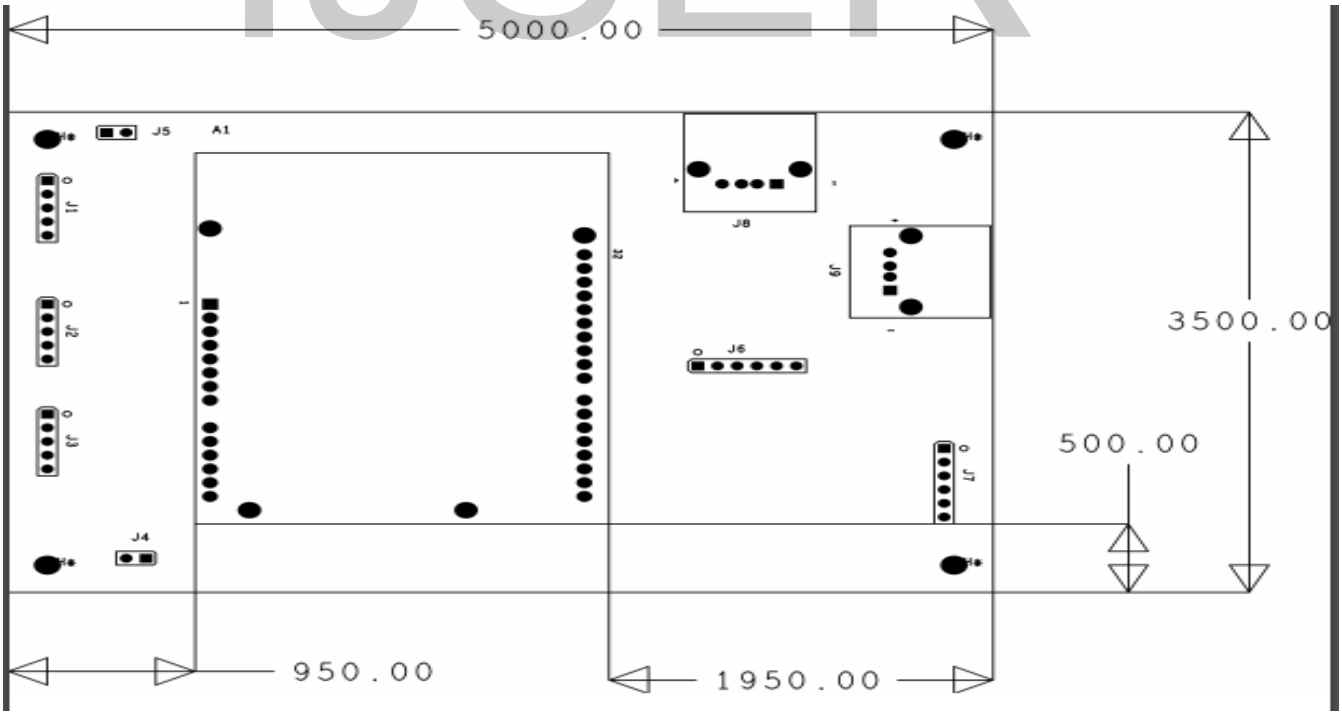


Figure 18-Component Placement

BOARD FILE FOR FABRICATION:

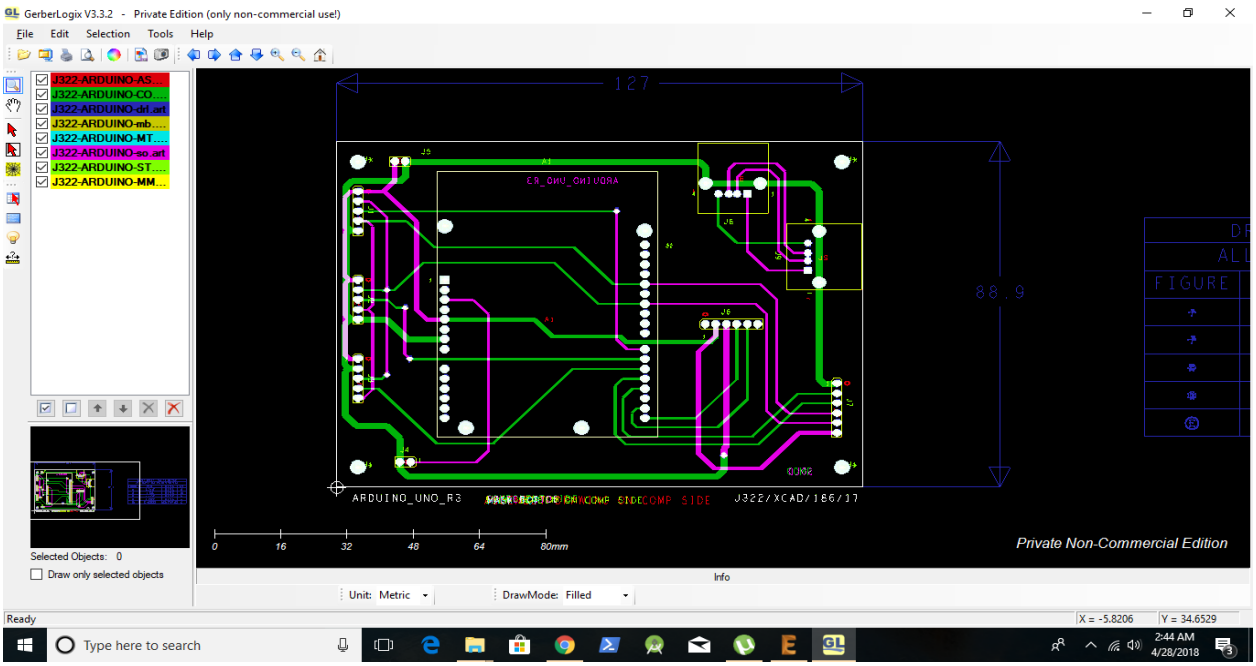


Figure 19-Board file

TOP LAYER:

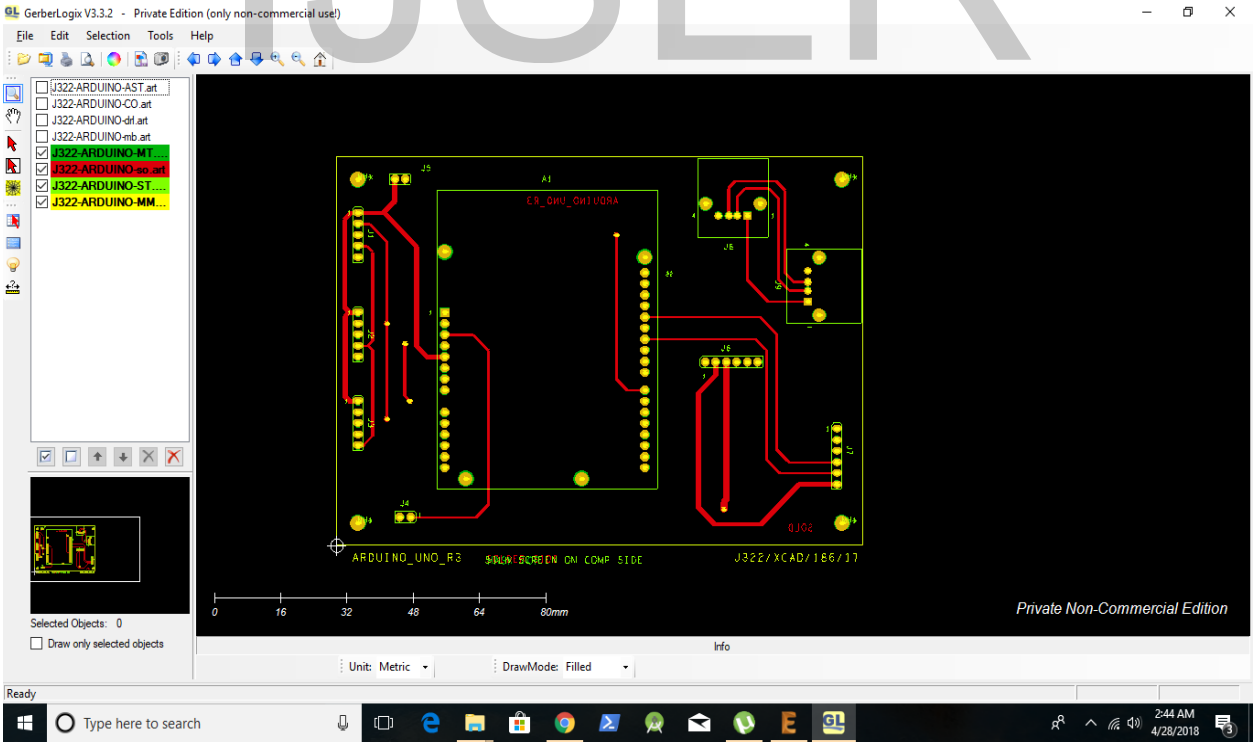


Figure 20-Top Layer

DIMENSIONS IN MILLIMETERS FOR FABRICATION:

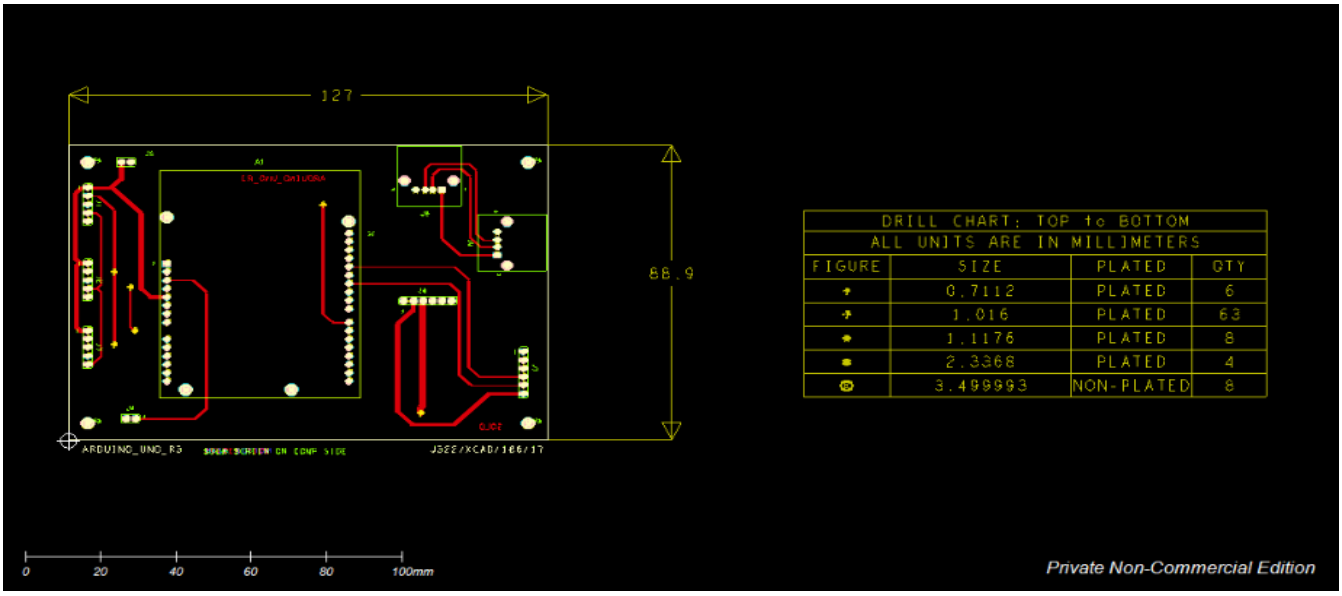


Figure 21-Dimensions

3.3.2 POST FABRICATION

The board file designed in EAGLE was converted to Gerber files which contain 8 layers required for fabrication, Namely Top layer, a bottom layer, Silkscreen, Pads and tracks, holes, and drills, etc.

These files are also called as the CAM files (Computer-aided manufacturing) files, which are required for the CNC machine to create holes, pads, tracks and other precise etching on the perforated copper board. The PCB after processing through the CNC is then dipped into a hot molten lead solution which gets accumulated on top of the board where the etching and CuS (Copper Sulphide) solution is applied to. The PCB is now sent to the green masking table, where the PCB is coated with a layer of green solution which forms a protective layer for the PCB. It prevents shorts, reduces EMI and EMF from external sources and also gives a recognizable color to the PCB. After green masking is done, the PCB is sent to the silkscreen print stage. Here, the PCB is printed with the required lines, dots and component locations. Any required important information can also be printed onto the PCB such as “Arduino UNO R3” which is printed on the In and Out boards.

This is the PCB after Fabrication, onto which the basic components such as bergsticks and headers are mounted.

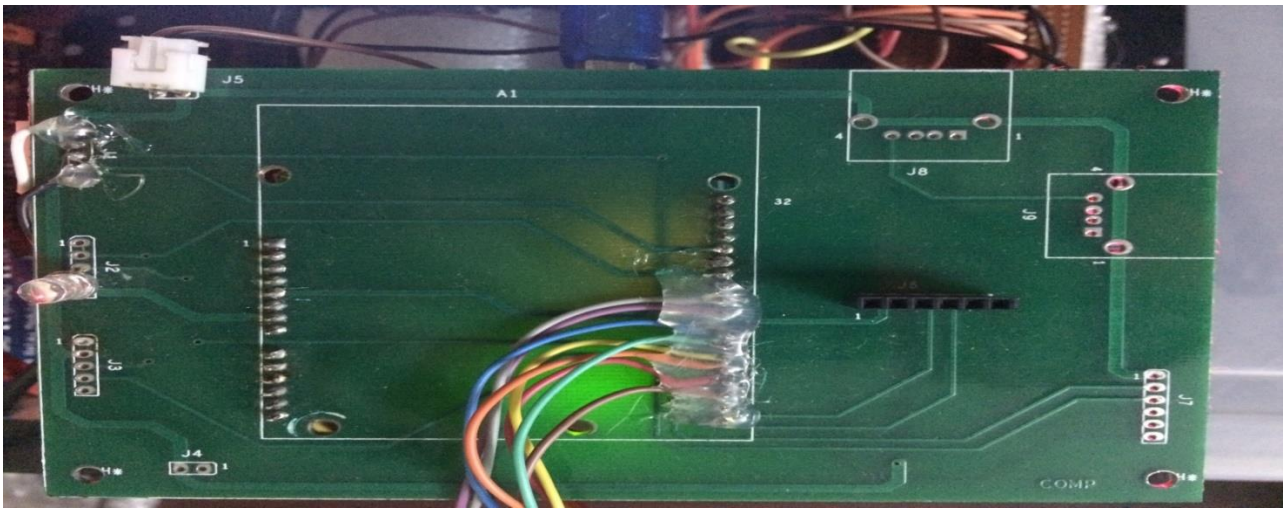


Figure 22-Post fabrication

The PCB is now ready for assembly and soldering, the Arduino card is mounted to the PCB as seen below. The LED, and relays and also connected to the same board, A 6 pin female bergstick is placed for attaching the Bluetooth module (HC-05). The holes on the PCB are mounted onto the Arduino using male bergsticks.

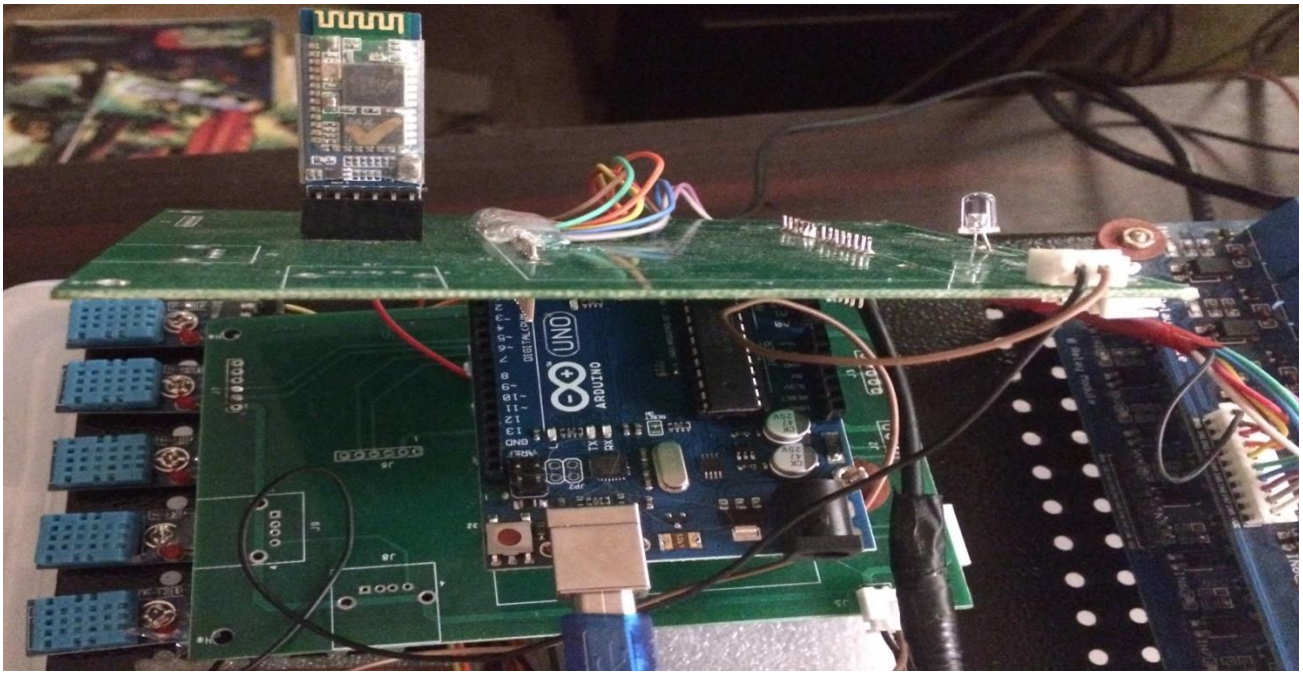


Figure 23-Arduino placement

3.4 ASSEMBLING HARDWARE AND COMPONENT PLACEMENT

1. The DHT11 sensors are connected to the IN-BOARD and the USB-Serial cable is connected to the Raspberry Pi using a USB A type to USB B type cable.
2. The Relays are connected to the OUT-BOARD using a 10 pin RMC (Relay mate connectors) cable. The OUT-BOARD is connected to the Raspberry Pi using a USB A-USB B type serial cable.
3. The display is mounted on the Raspberry Pi and connected to the HDMI port via the 2 way male to male HDMI jack.
4. All the subsystems are mounted on a perforated metal base using metal studs and nuts.
5. The individual firmware's are now dumped onto the In and Out boards and the Processor is loaded with the Real-time batch processing application that initializes on commands from the terminal of the Linux based Debian OS, (a.k.a) Raspbian Jessie.

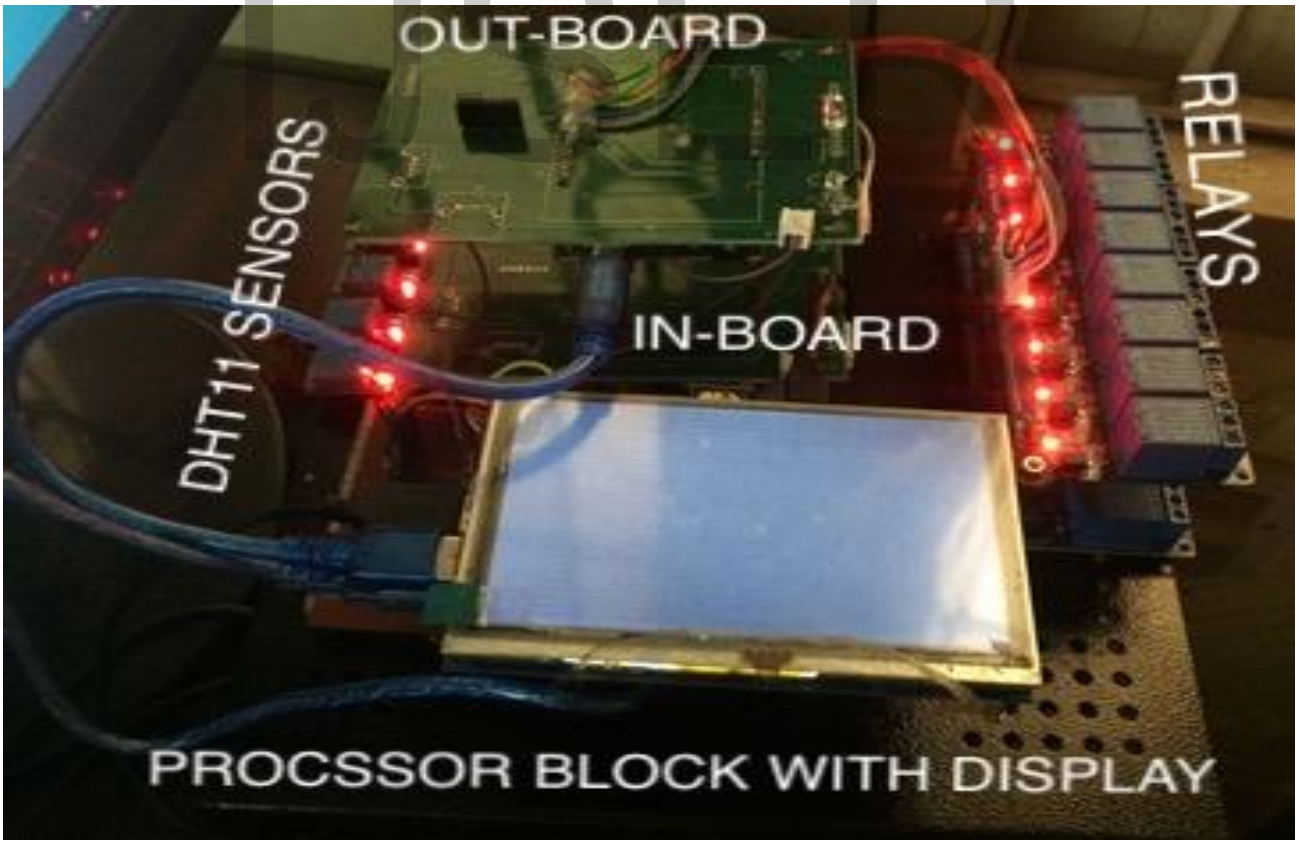


Figure 24-Assembling components

The final peripheral components are assembled in place and the Bluetooth module (HC-05) is also mounted. The Embedded system is now complete and is awaiting user commands for processing and monitoring tasks.

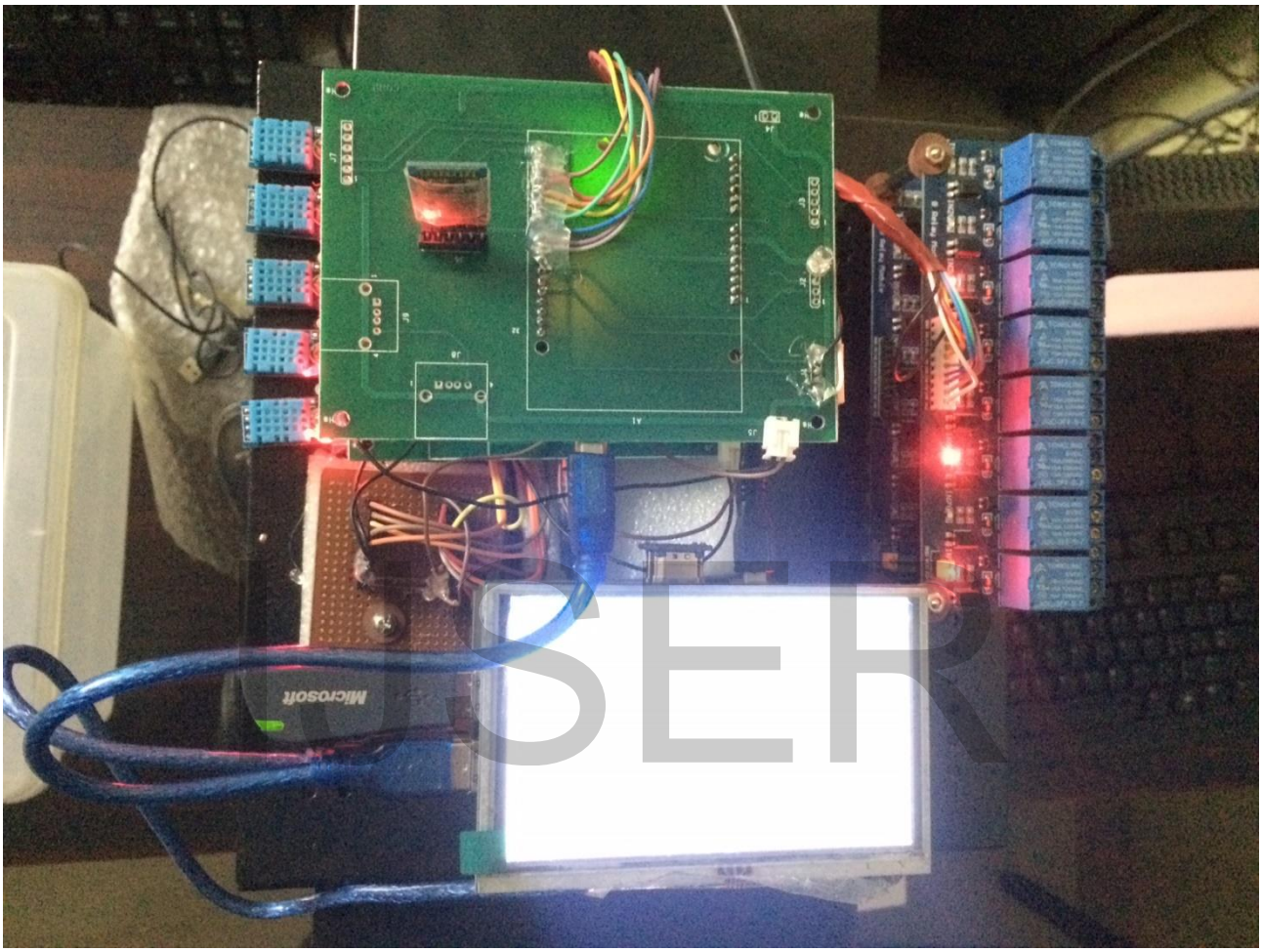


Figure 25-Final Assembly

3.5 SOFTWARE BLOCK

The Software block consists of the different types of software, code and application sources required to build the real-time embedded system.

The different software's are namely, the specific controller firmware's, the processor application kernel and the mobile application development source code (Android Studios).

Different software tools are employed in writing and compiling the code of the software.

3.5.1 TYPICAL CONTROL FLOW

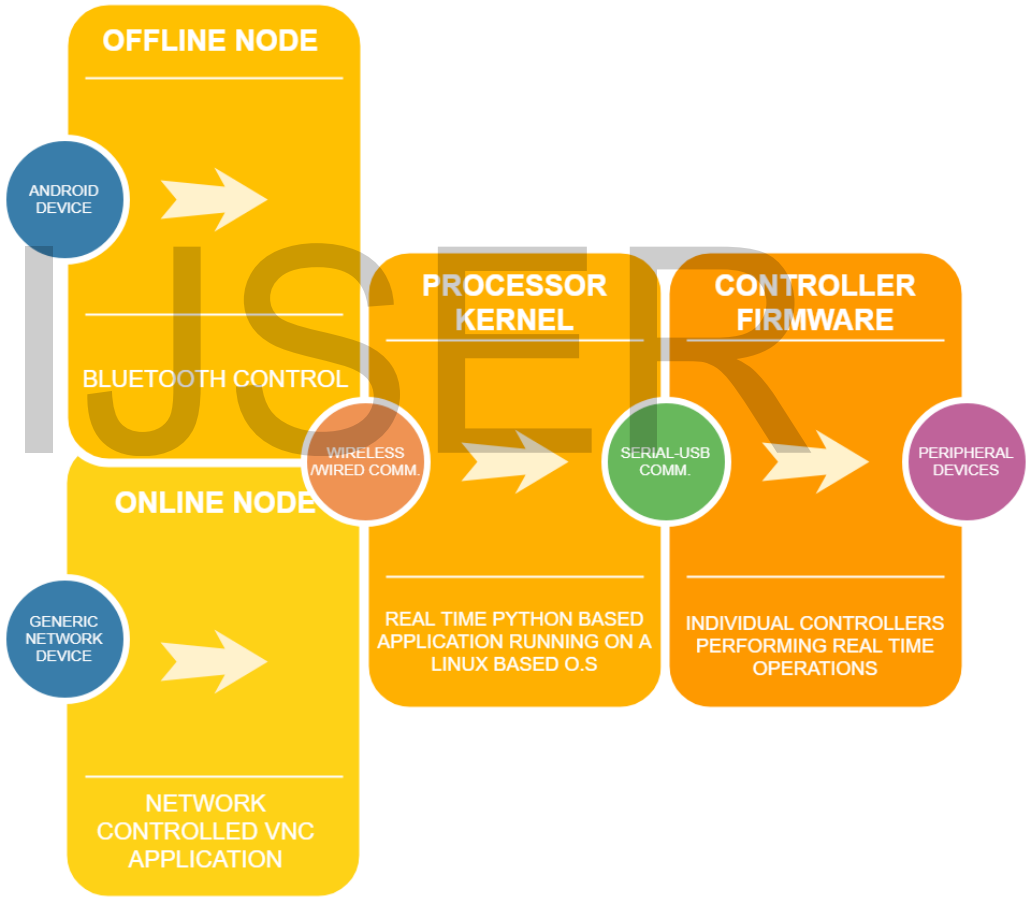


Figure 26-Control flow diagram

1. As shown in the above control flow diagram, the first event trigger can originate from either the offline node or the online node, the offline node being an android device which runs a mobile application that is an instance of the same features of the processor application. The code of the offline node is compiled and emulated in Java using the

Android Studios IDE. A signed apk is generated which can be installed on to any android mobile above the version 4.1 i.e. (Jelly Bean).

2. The online node is any device connected to the same network as that of the processor. By using VNC (Virtual Network Computing), which is a graphical desktop sharing system that uses the Remote Frame Buffer protocol (RFB) to remotely control another computer, we can access the processor application from a remote location and monitor the data logging and switching of relays and LED's.
3. The processor application kernel is a simple Python-based code that runs the real-time data acquisition algorithm which acquires data from the IN-BOARD and it also runs a switching code that can be used to switch Relays and LED's. Both the tasks are accomplished by connecting the Raspberry Pi to the controllers using the USB-Serial connector cables. This application code is written in Python that is interpreted using PiP3 (Pip installs python) using which, the code can be deployed in any platform that supports PiP3.
4. The Controllers have specific firmware's to accept data from the serial port process it and perform the necessary task or to use the serial port and send acquired data. This is a C++ code written on Arduino IDE, an open source platform for compiling and dumping code onto the controllers.

3.5.2 SOFTWARE TOOLS USED

CODE	SOFTWARE USED
Controller Firmware	Arduino IDE
Processor Application	PiP3 (Pip Installs Python) Version 3
Online Application	Processor application running on VNC
Offline Application	Android Studios

3.5.3 CONTROLLER FIRMWARE

IN-BOARD FIRMWARE:

This code must only perform the task on continuously acquiring data from the 5 DHT (Digital Humidity and Temperature) sensors and sending it to the Processor via the USB-Serial interface.

The code must also be written keeping the baud rate (data transfer rate) in mind, so to obtain a baud rate of 2000 samples/ minute, there must be absolutely no delay in acquiring data and send it via the USB-COM port. The only delay that is allowed is the clocking delay of taking data from each sensor one after the other.

The different libraries used in the IN-BOARD are:

1. HCMAX7219.h: 7 segment LED display library
2. SPI.h: Serial Port Interface library for displaying to the 7 segment LED's serially
3. DHT.h: The Digital Humidity and Temperature library which contains the memory addresses of the capacitive data coefficients of the DHT sensors in order to acquire and sample data.

The data acquired is sampled in a way that can be understood during data transfer. The Bits must be sent in an array of Strings. The array arrSend[] is declared as the data transfer array in the code. The array has a dimension of 15(0-14), wherein the first bit represents the ON/OFF bit i.e. 1/0 of the device, the next 5 bits are the Temperature bits from the 5 DHT sensors, the next 5 bits are the Relative Humidity bits from the 5 DHT sensors, the next bit represents the Tavg bit, which is the average temperature of the 5 bits, the next bit is the Havg bit, which represents the average relative humidity of the 5 humidity bits. The next two bits are also On/OFF bits used to signify that both the Temperature and Humidity values are samples from all 5 sensors.

CODE SNIPPET (DATA ACQUISITION):

```
  for(i=0;i<5;i++)
{
    h[i] = dht[i].readHumidity();
    t[i] = dht[i].readTemperature();
    arrSend[i+1]=t[i];
```

```
        arrSend[i+6]=h[i];
        if (isnan(t[i]) || isnan(h[i]))
         {
          t[i]=h[i]=0;
           Serial.print("Failed to read from DHT at Sensor =");
           Serial.print(i+1);
         }
    }
```

OUT-BOARD FIRMWARE:

This firmware must perform the task of receiving data from the processor for switching of relays and LED's and should be configurable for future applications. The firmware must also be equipped with ports for connecting the 7 segment LED panels and data transfer to other peripheral devices using SPI protocol.

CODE SNIPPET (Serial Port request handling):

```
void loop()
{
  Serial.flush();
if (Serial.available() > 0)//If serial port is open
  {string = "";}//input string=0
  while(Serial.available() > 0)
  {
    command = ((byte)Serial.read());//real a chunk of bytes from serial port
    if(command == ':')//If serial read() gives a ":" symbol, then stop reading
    {
      break;
    }
    else
    {
      string += command;//Keep accumulating the String "string" with next bits.
    }
    delay(1);//Delay reading by 1 millisecond.
  }
```

This code shows how data is read and stored from the serial port.

The respective tasks to be performed, such as LED On/OFF, Relay ON/OFF is done in this way, by using a chunk of bytes sent via Serial port.

3.5.4 PROCESSOR APPLICATION AND KERNEL

The processor must run the real-time data acquisition and batch processing system what can communicate with both the IN-BOARD and OUT-BOARD and also connect to the Online Node of the IoT Gateway.

APPLICATION FRONT END:

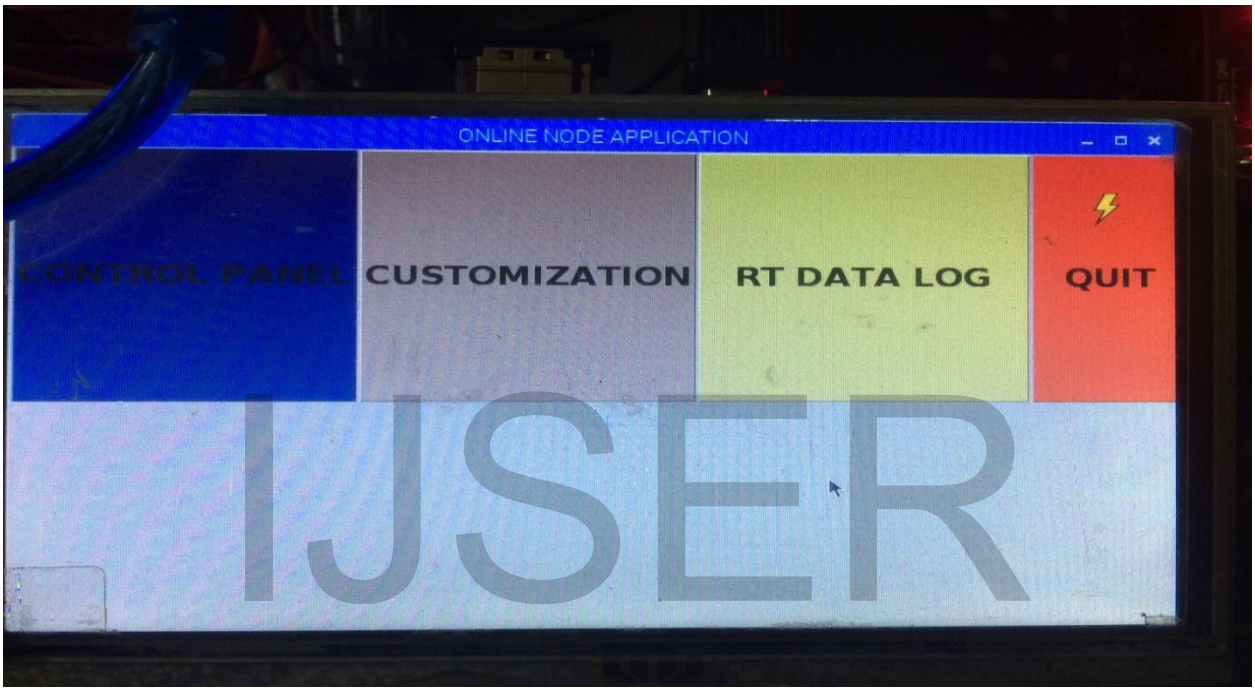


Figure 27-Online application front end

1. The Control Panel is the window which can be used in switching ON/OFF all devices such as LED's and Relays.
2. The Customization window can be used to customize parameters such as input/output checking, for example, if the temperature of T1 bit is >35 deg Celcius, switch off Relay 1 and 2. This kind of customizability is used for Industrial IoT applications.
3. The RT data log window acquires data from the IN-BOARD, logs it in the Micro SD card present in the device, and displays it to the user for monitoring based applications. A futuristic dashboard can also be made using the same real-time log.
4. The quit option quits the Online Node application kernel.

3.5.5 ONLINE APPLICATION

The VNC client is set up in any device connected to the same network as that of the processor and the IP address and port number of the processor is obtained by typing the instruction "ifconfig" in the terminal of the processor Operating System. These details are taken and fed into the VNC client of any system and made to connect automatically.

Figure 28-VCN Terminal application implementation

As seen above, the same processor application is accessible from any device connected to the same network. If the device is a remote node, then a global IP address is required for accessing the application. As seen above the VNC client recognizes the application in the local IP address of 192.168.43.247 and in Port 0.

3.5.6 OFFLINE MOBILE APPLICATION

The Offline Application is a mobile application which runs on any generic android device with version >4.1 (Jellybean). This application can be deployed to any android device and can communicate with the Bluetooth module of the OUT board and directly switch the LED's and the Relays. As seen below, the opening page of the application is designed and emulated.

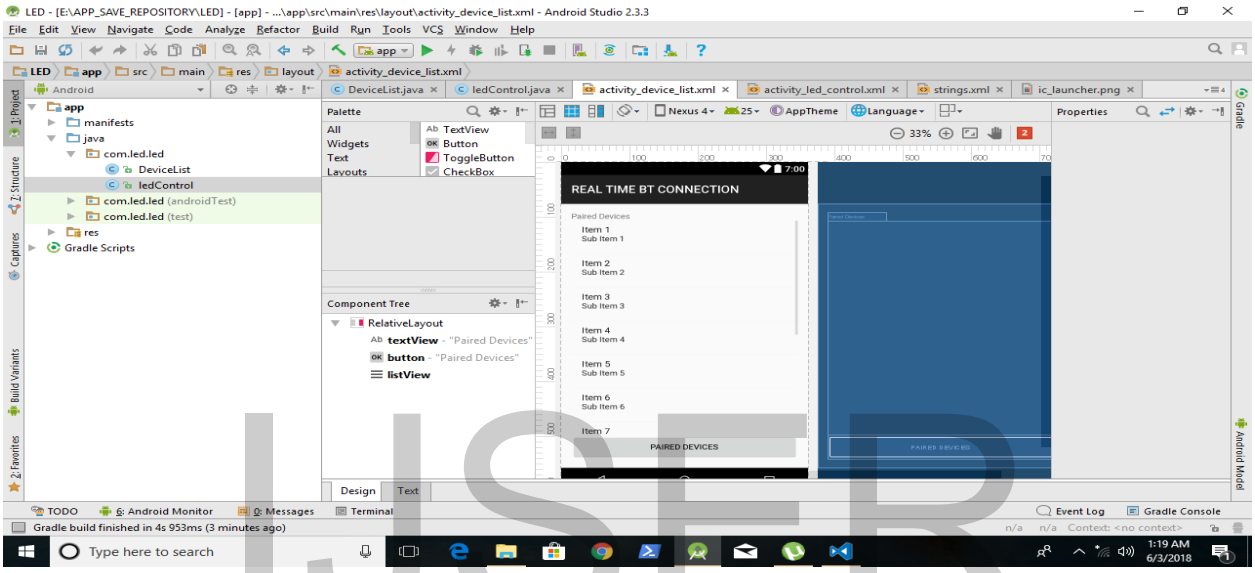


Figure 29-XML Emulated design

The Main page of the application is designed with Text boxes and Buttons.

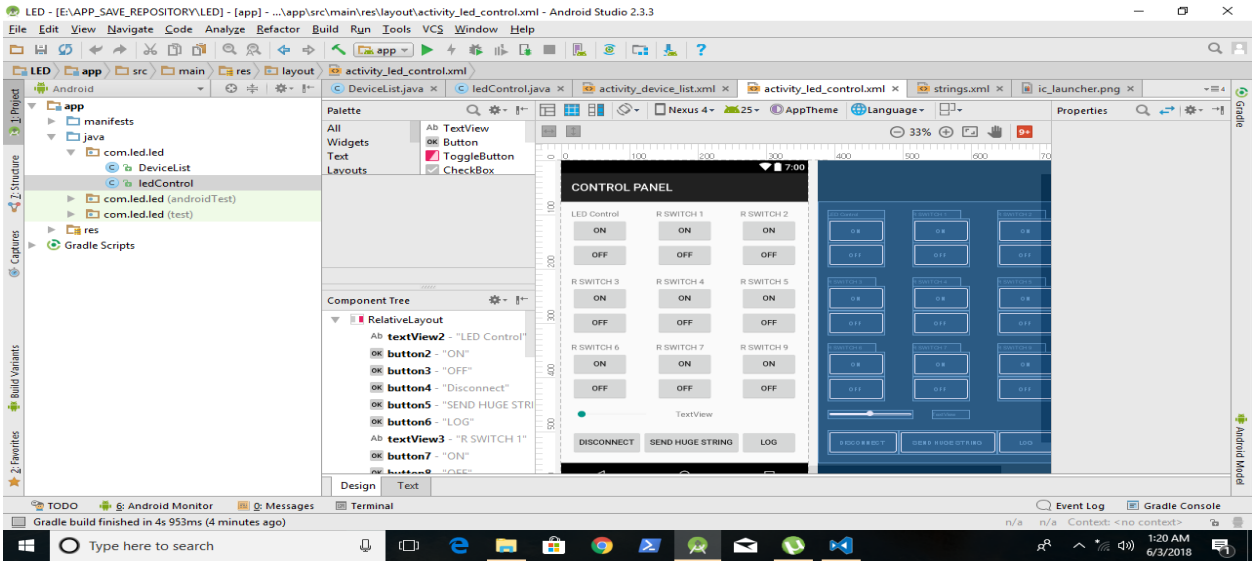


Figure 30-Main page emulation

4 CHAPTER 4: RESULTS AND OUTPUTS

4.1 IN-BOARD OUTPUTS

The data obtained in the controller on the IN-BOARD is the real-time log of sensor data which is monitored on the processor also. An array of strings called as “arrSend[]” is used for data transmission. The array has a dimension of 15(0-14), wherein the first bit represents the ON/OFF bit i.e. 1/0 of the device, the next 5 bits are the Temperature bits from the 5 DHT sensors, the next 5 bits are the Relative Humidity bits from the 5 DHT sensors, the next bit represents the Tavg bit, which is the average temperature of the 5 bits, the next bit is the Havg bit, which represents the average relative humidity of the 5 humidity bits. The next two bits are also On/OFF bits used to signify that both the Temperature and Humidity values are samples from all 5 sensors.

IN-BOARD COM PORT OUTPUT (ARDUINO SERIAL MONITOR):

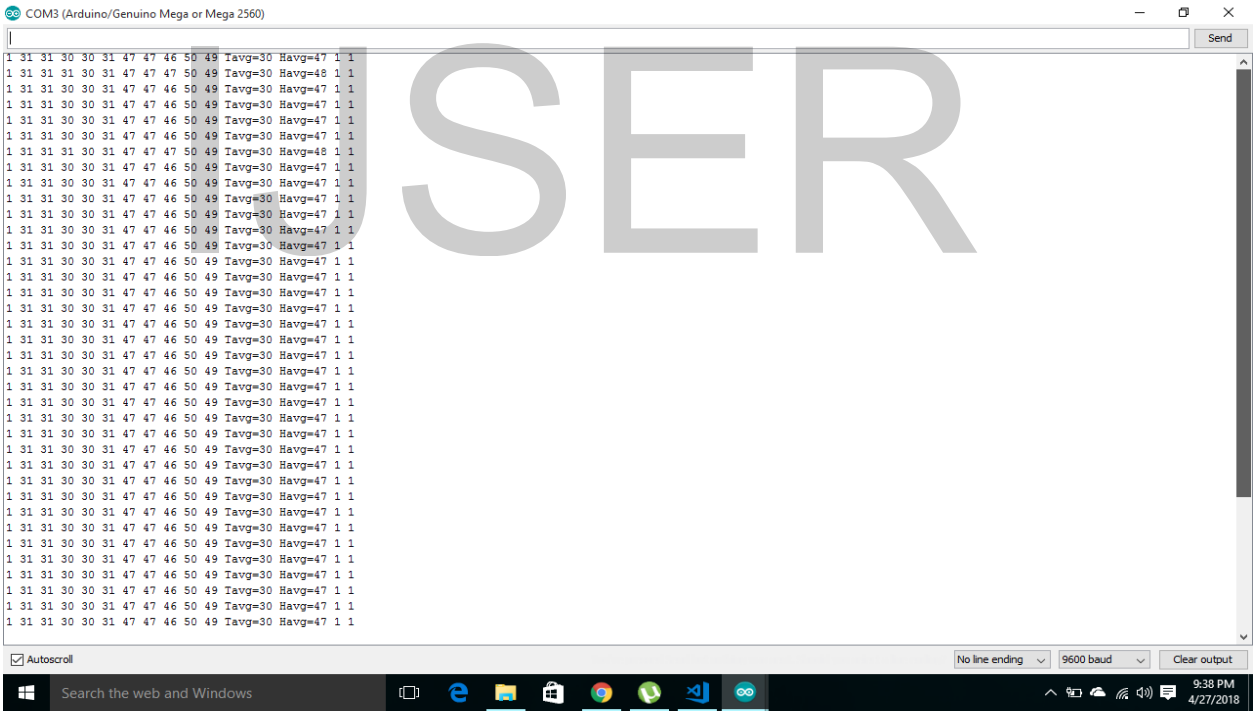


Figure 31-COM PORT Output of Serial Monitor (Arduino IDE)

As seen in the above picture, 15 bits of data are being sent in the array that is being sampled at a baud rate of 9600bps and totally a sampling rate of 20000 samples/minute.

PROCESSOR RT DATA LOG:

The dame array of data is acquired in the processor log also, which is then stored in the SD card or the processor.

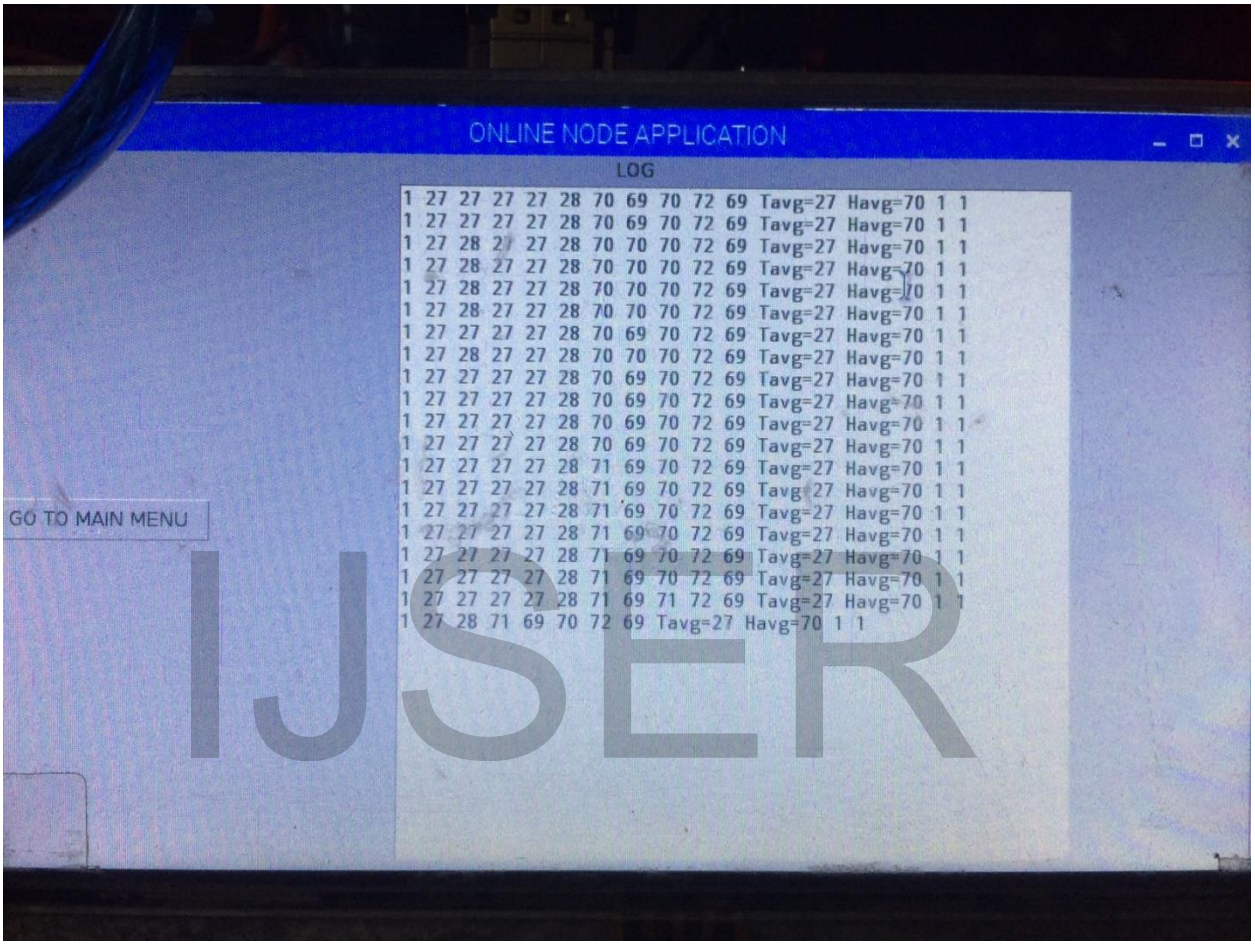


Figure 32-Processor Data log

So, the processor log can also be obtained in the form of a text file which stores the information onboard. This is very useful in applications for smart health monitoring, vehicle OBD2 (On board diagnostics) data acquisition and so on.

4.2 OUT-BOARD RESULTS

The OUT-BOARD is connected to an LED and 8 channel relays. So, the output from these peripherals can be viewed as a result of event triggers from online and offline IoT nodes.

The trigger parameters are predefined in the code as follows:

LED ON/LED OFF	“TO”/”TOFF”
RELAY1 ON/OFF	“RSON1”/”RSOFF1”
RELAY2 ON/OFF	“RSON2”/”RSOFF2”
RELAY3 ON/OFF	“RSON3”/”RSOFF3”
RELAY4 ON/OFF	“RSON4”/”RSOFF4”
RELAY5 ON/OFF	“RSON5”/”RSOFF5”
RELAY6 ON/OFF	“RSON6”/”RSOFF6”
RELAY7 ON/OFF	“RSON7”/”RSOFF7”
RELAY8 ON/OFF	“RSON8”/”RSOFF8”

4.21 RELAY AND LED OUTPUTS

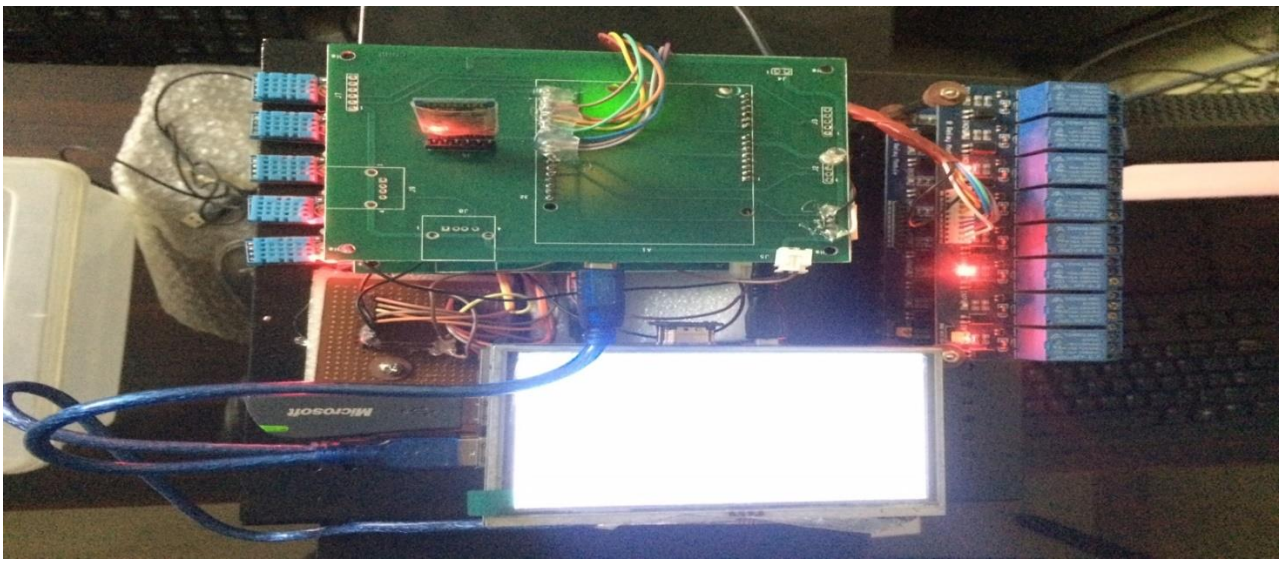


Figure 33-Relay 1,2,6-OFF , LED -OFF

Here, relays 1,2 and 6 are made to be OFF and LED is also OFF

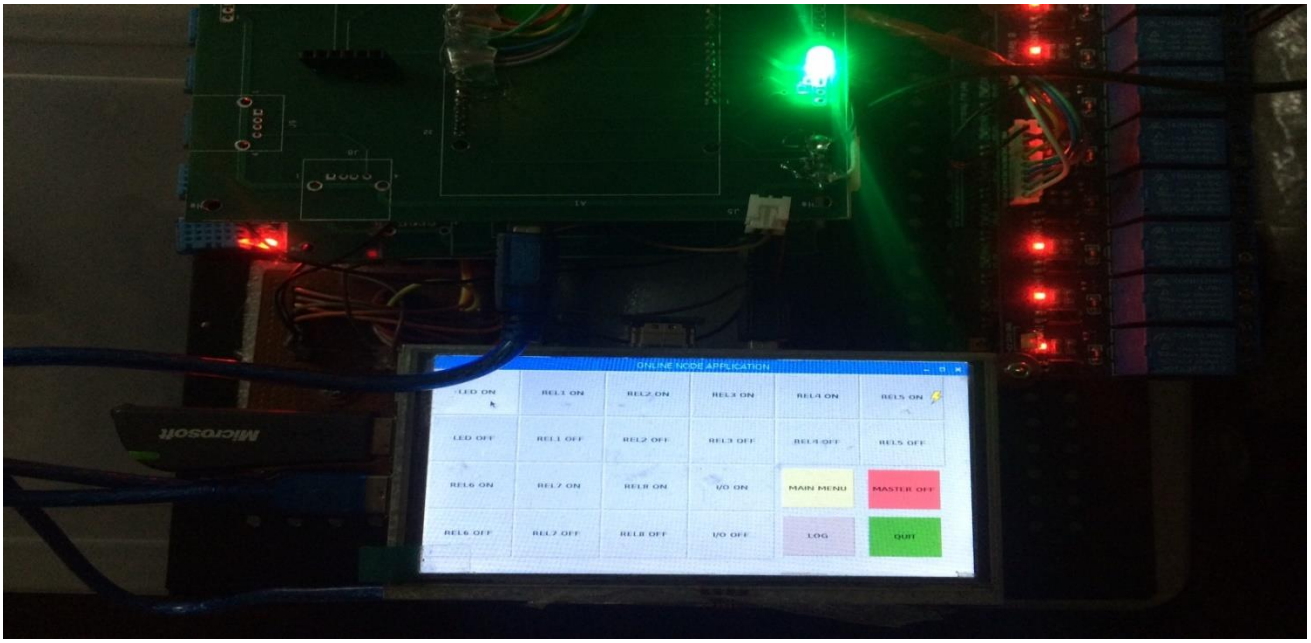


Figure 34-Relay1-8 ON,LED-ON

Here, all the relays from 1-8 are made to be on and LED is also ON. This is an event trigger from the processor.

4.22 EVENT TRIGGERS FROM PROCESSOR (ONLINE) AND OFFLINE NODES

As seen in the results, the Relays and LED must be controlled either by the processor (Online) or a mobile device (Offline) node.

PROCESSOR (ONLINE) TRIGGER:

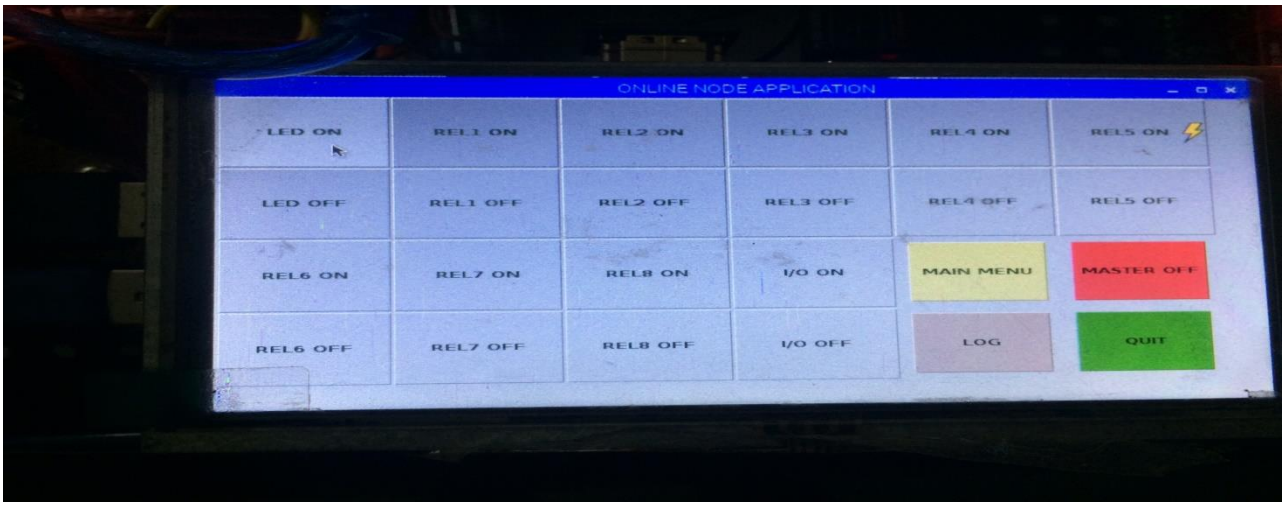


Figure 35-Online trigger

As seen in the picture, the LED ON button is clicked and the processor sends a signal via, the Serial port and the LED on the OUT-BOARD turns on.

OFFLINE TRIGGER:

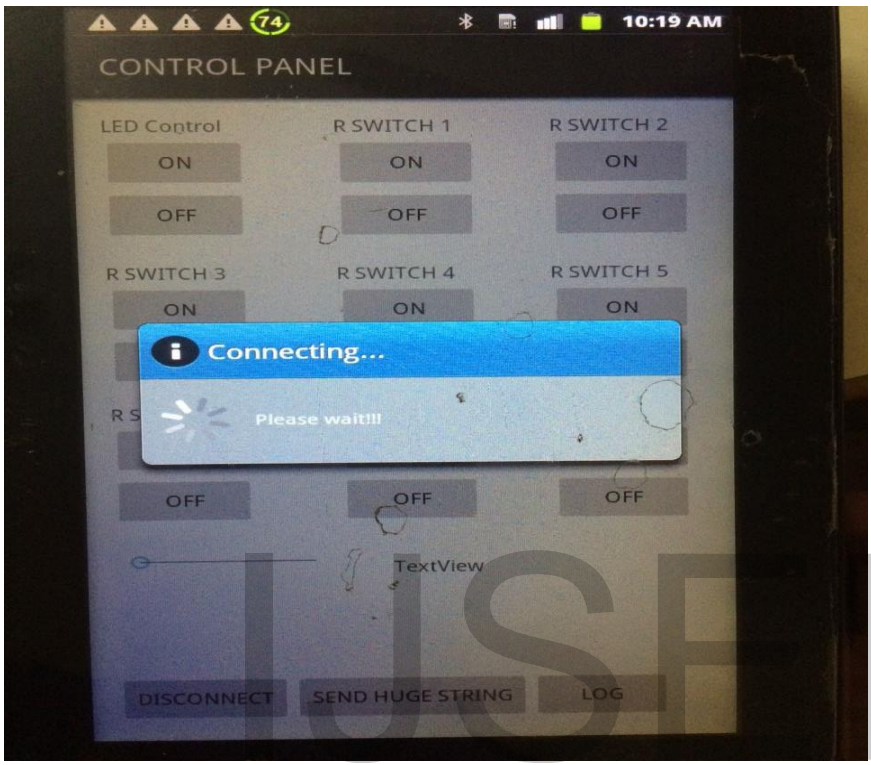
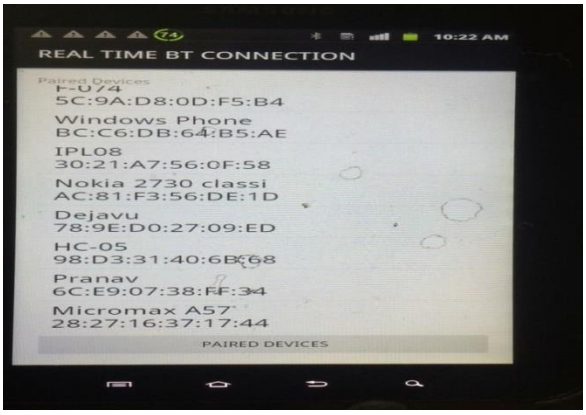


Figure 36-Offline trigger

As seen above, the mobile application is used for sending the signals via Bluetooth to the embedded system. The OUT-BOARD Bluetooth module connects directly to the Mobile device once it is initialized.



As seen in the picture, the device HC-05 is connected to the application running on the Android device.

Once, it is connected to the device, it connects automatically the next time the Bluetooth module is plugged into the OUT-BOARD.

Figure 37-Android app

5 CHAPTER 5: USE-CASE APPLICATIONS

1. INDUSTRIAL IoT

A Real-time industrial application of the project would be in an industry which requires a batch processing system that acquires data from multiple locations, processes it in the form of a batch and performs the required tasks. Example: A manufacturing industry worker needs to turn off 3 devices such as a hydraulic press, CNC drill, and a grinder once the core temperatures of the devices reach 32°C, 45°C and 80°C respectively.

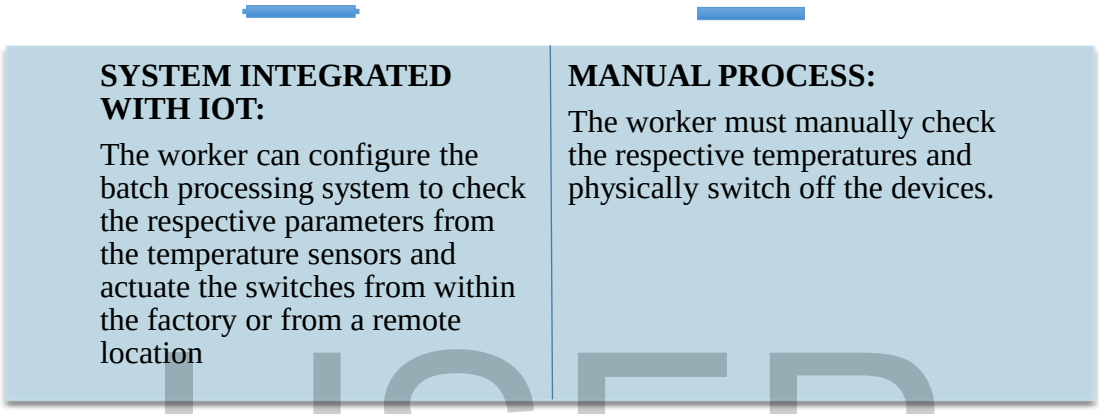


Figure 38-Industrial IoT vs Manual process

2. HOME AUTOMATION

Smart homes are a growing trend in the present day and age. A home can be completely retrofit with the embedded hardware present in the project. A home can be made completely automated and also, the user must be able to monitor and control his/her home from a remote location.

A few of the main applications in the home automation sector is automatic switching, automatic dimming, and dipping of lighting and smart metering. Also, to bring about energy efficient power consumption solutions by using lights with proximity sensors at home. So, only when there is a person in a particular room, the lights and electrical devices will turn on, else they will turn off automatically.

3. SMART FARMING

Smart irrigation systems based on sensor data input. DHT (Digital Humidity and temperature) sensors can be used to predict and determine the water content present in the soil.

4. WEARABLES

Real-time monitoring of body parameters can be done using the same hardware. The parameters such as heart rate, body temperature, and step count can be measured on a wearable device implemented using the same system.

GPS connectivity for an on-board wearable device greatly benefits the visually challenged.

5. ADAS

(Advance driver-assistance systems)

Very commonly used technology in vehicles in today's world to assist the driver on road and provide multiple safety features.

6 CHAPTER 6: CONCLUSION AND SCOPE OF THE PROJECT

CONCLUSION:

The project is designed and implemented as described and, all test cases for IN-BOARD and OUT-BOARD are checked and verified for applications such as Industrial IoT, Home Automation, and Smart monitoring applications.

The project has also resulted in the design and fabrication of generic PCB's which can be used for different applications. A generic application is deployed on the processor which can be tailored for various use-case applications of the project.

A mobile application is also implemented which has a signed apk, that can be installed on any generic mobile device >Android 4.1 (Jelly Bean). This can be used to generate event triggers and communicate with the controller boards independently.

FUTURE SCOPE:

The data acquired from the controller can be sent to a cloud-based platform for storage and for creating a real-time dashboard. Using the real-time data on the cloud, a Machine learning algorithm can be developed and implemented on the cloud for performing even more intuitive tasks such as predictive modeling, speech-text conversion, and Image recognition and so on.

The processor application can be implemented with a machine learning algorithm that is specific to each application and can be tailored for a particular user who is not connected to the Internet. The Advanced driver assistance system can be modeled using this perspective.

The IoT devices can be connected to various display devices such as 7 segment displays and dot matrix display, for which the libraries and hardware is pre-designed in the embedded system.

7 REFERENCES

[1] Probabilistic Recovery of Incomplete Sensed Data in IoT

Berihun Fekade ; Taras Maksymyuk ; Maryan Kyryk ; Minho Jo

Publication Year: 2018, Page(s):2282 - 2292
Cited by: Papers (3)

[2] Narrowband Internet of Things: Simulation and Modeling

Yiming Miao ; Wei Li ; Daxin Tian ; M. Shamim Hossain ;Mohammed F. Alhamid

Publication Year: 2018, Page(s):2304 – 2314

[3] The Future Internet of Things: Secure, Efficient, and Model-Based

Joshua E. Siegel ; Sumeet Kumar ; Sanjay E. Sarma

Publication Year: 2018, Page(s):2386 – 2398

[4] Guest Editorial Special Issue on Emerging Social Internet of Things: Recent Advances and Applications

Giancarlo Fortino ; Mohammad Mehedi Hassan ; Mengchu Zhou ;Andrzej M. Goscinski ; Md Zakirul Alam Bhuiyan ; Jianqiang Li ;Sourav Bhattacharya

Publication Year: 2018, Page(s):2478 – 2482

[5] Evaluating Critical Security Issues of the IoT World: Present and Future Challenges

Mario Frustaci ; Pasquale Pace ; Gianluca Aloi ; Giancarlo Fortino

Publication Year: 2018, Page(s):2483 – 2495

[6] Edge Computing and Social Internet of Things for Large-Scale Smart Environments Development

Franco Cicirelli ; Antonio Guerrieri ; Giandomenico Spezzano ;Andrea Vinci ; Orazio Briante ; Antonio Iera ; Giuseppe Ruggeri

Publication Year: 2018, Page(s):2557 - 2571

[7] Distributed Wireless Power Transfer System for Internet of Things Devices

Kae Won Choi ; Arif Abdul Aziz ; Dedi Setiawan ; Nguyen Minh Tran ; Lorenz Ginting ; Dong In Kim

Publication Year: 2018, Page(s):2657 – 2671

[8] Multiband Ambient RF Energy Harvesting Circuit Design for Enabling Batteryless Sensors and IoT

Ufuk Muncuk ; Kubra Alemdar ; Jayesh D. Sarode ; Kaushik Roy Chowdhury

Publication Year: 2018, Page(s):2700 - 2714

[9] A New Relay Policy in RF Energy Harvesting for IoT Networks—A Cooperative Network Approach

Z. Behdad ; M. Mahdavi ; N. Razmi

Publication Year: 2018, Page(s):2715 – 2728

[10] RF Energy Transfer Channel Models for Sustainable IoT

Sidharth Kumar ; Swades De ; Deepak Mishra

Publication Year: 2018, Page(s):2817 – 2828

[11] Simultaneous Wireless Information and Power Transfer for Internet of Things Sensor Networks

Sung Ho Chae ; Cheol Jeong ; Sung Hoon Lim

Publication Year: 2018, Page(s):2829 – 2843

[12] A Privacy-Aware Architecture at the Edge for Autonomous Real-Time Identity Reidentification in Crowds

Seyed Ali Miraftabzadeh ; Paul Rad ; Kim-Kwang Raymond Choo ;Mo Jamshidi

Publication Year: 2018, Page(s):2936 - 2946

[13] Toward a Smart Society Through Semantic Virtual-Object Enabled Real-Time Management Framework in the Social Internet of Things

Zia Ush Shamszaman ; Muhammad Intizar Ali

Publication Year: 2018, Page(s):2572 – 2579

[14] Accumulate Then Transmit: Multiuser Scheduling in Full-Duplex Wireless-Powered IoT Systems

Di Zhai ; He Chen ; Zihuai Lin ; Yonghui Li ; Branka Vucetic

Publication Year: 2018, Page(s):2753 - 2767

[15] https://www.elprocus.com/building-the-internet-of-things-using-raspberry-pi/

[16] https://www.silvan.co.in/projects-products.php

[17] https://iot-analytics.com/10-internet-of-things-applications/

[18] http://www.engpaper.com/iot-2016.htm (IEEE Journal)

[19] http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?reload=true&punumber=6488907

[21] https://learn.sparkfun.com/tutorials/using-eagle-schematic

[22] https://www.youtube.com/watch?v=1AXwjZoyNno

[23] https://developer.android.com/training/basics/firstapp/ (Android app development tutorials and guide to designing a basic first application).

[24] https://developer.android.com/guide/topics/connectivity/bluetooth (Android app development guide for Bluetooth connectivity and related features).