2-in-1 IoT Development Kit Manual

ADVANCED ELECTRONIC SYSTEMS
#143, 9th Main road, 3rd Phase,
Near Laggere Cross, Peenya,
Bengaluru – 560 058
E-mail: sales@alsindia.net
WEB: alsindia.net

CONTENTS PAGE No.

1. INTRODUCTION

The 2in 1 IoT Development Kit is designed to be a comprehensive tool to accelerate learning IoT Fundamentals and apply the concepts to get a Hands-On experience by performing different experiments. It can also be used by students and faculty alike to implement projects based on IoT concepts. Flexibility is at the heart of the Kit, and accordingly the students and faculty can not only use the hardware on the board in imaginative ways, but also innovatively write programs to try their own use cases.

This user manual provides the descriptions of the features and also examples of use cases, to guide the students to use it effectively.

The 2 in 1 IoT Development Kit can have an Arduino UNO R3 or a Raspberry Pi 3 B+ controllers, based on the order. In cases, where it was ordered with both, it can be used with one controller at a time. The kit has both open source micro-controllers. It helps the students to perform IoT related experiments based on either Arduino or Raspberry pi 3 B+. The kit has on board sensors which help the students to develop number of applications on IoT. As the kit has on board ESP8266 Wi-Fi module the students can send the different sensors information to the cloud. Using Raspberry Pi 3 B+ one can login to the kit from a remote location using SSH tools. One can flash code into the Arduino using Arduino IDE. The Raspberry Pi 3 B+ works on Raspbian OS which is installed on to the SD Card. Using Arduino IDE one can write the program in Embedded C Language. For Raspberry Pi 3 B+ one can write the programs in different Languages like Python, Embedded C, Node JS, Java, etc.

2. IoT DEVELOPMENT KIT SPECIFICATIONS

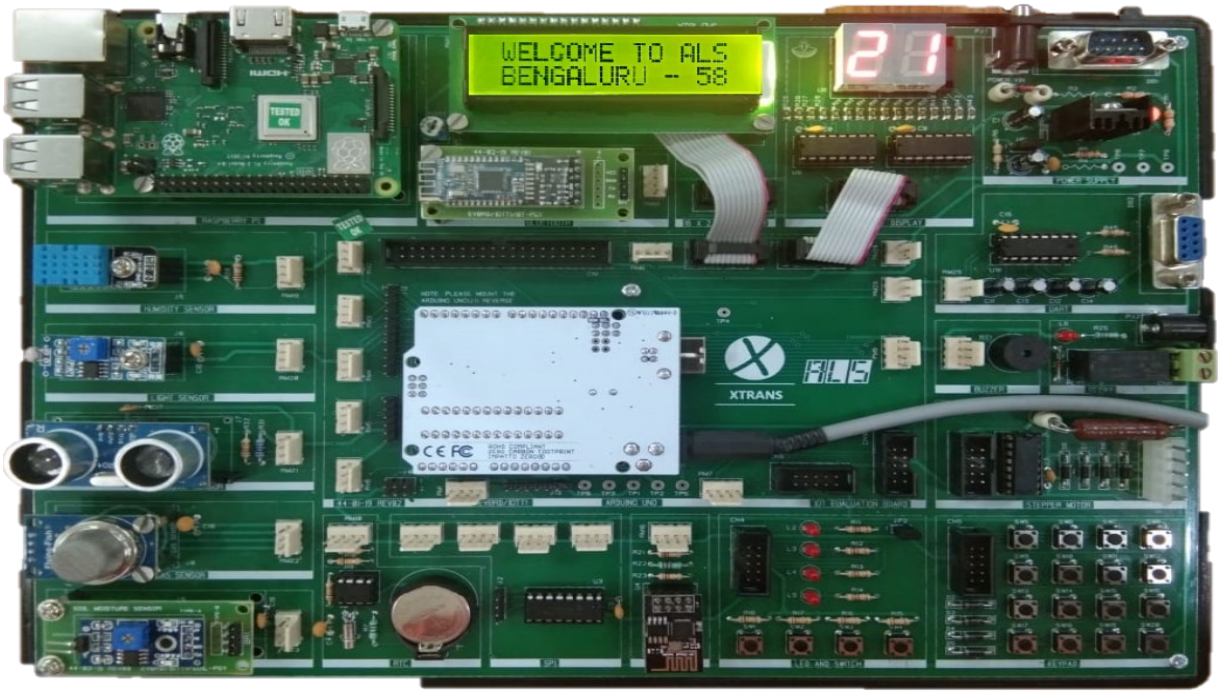


Fig. 2: 1-in-1 Iot Development Kit.

Hardware Description:

1. 16×2 LCD Display
2. Humidity sensor
3. Light sensor
4. Ultrasonic sensor
5. Gas sensor
6. Soil Moisture sensor
7. Bluetooth
8. Raspberry Pi 3 B+
9. 7 – Segment display
10. Power Supply
11. UART
12. Relay
13. Buzzer
14. Stepper Motor Interface
15. 4*4 Matrix Keypad
16. LED and Switch
17. WiFi ESP8266
18. ADC MCP3008
19. RTC
20. Arduino Uno

Specifications:

- Board is driven using either Raspberry Pi 3 B+ or Arduino UNO R3 controllers.
- Facilitates MQTT or HTTP application protocol connection to various IoT cloud platforms such as ThingSpeak, AWS IoT, Microsoft Azure, IBM Bluemix, etc.
- It can be connected to a Private Cloud including X trans Cloud.
- On board voltage regulator for generating 3.3V from +5V input through power supply /adapter.
- On board Wi-Fi connectivity through ESP8266 for Arduino and internal Wi-Fi connectivity for Raspberry Pi 3 B+.
- On board digital and analog sensors like Temperature & Humidity, Gas, Light, Ultrasonic distance sensor and Soil moisture sensor.
- On board ADC MCP3008 for Raspberry Pi 3 B+ to connect analog sensors and internal ADC for Arduino.
- On board 16×2 character LCD.
- On board 4×4 keypad matrixes.
- On board 32.768 kHz quartz crystal RTC DS1307 powered by coin cell battery of 3V.
- 4 LED's and 4 Switches.
- 2-digit 7 segment display.
- On board stepper motor driver to drive 5V stepper motor.
- One Buzzer.
- Connector slot for external Bluetooth module for Arduino.
- Provided extra connectors for connecting external analog and digital sensors.
- I2C, SPI communication protocols.
- On board UART connector.
- UART Software serial communication.
- On board Relay to drive external device.
- Several of the outputs to the on-board controller can be accomplished through events from cloud such as turning on LED’s or sending a test message to the display is possible.
- Outputs can be programmed to be event or threshold driven.
- Performance of IoT experiments and design / implementation of student projects possible including provision to write their own programs.

3. HARDWARE DETAILS

3.1. Arduino UNO R3

Arduino is an open-source hardware, software and content platform with a global community.

Specifications:

- Micro-controller: ATmega328.
- Operating Voltage: 5V.
- Input Voltage (recommended): 7-12V.
- Input Voltage (limits): 6-20V.
- Digital I/O Pins: 14 (of which 6 provide PWM output).
- Analog Input Pins: 6.
- DC Current per I/O Pin: 40 mA.
- DC Current for 3.3V Pin: 50 mA.
- Flash Memory: 32 KB of which 0.5 KB used by boot-loader.
- SRAM: 2 KB (ATmega328).
- EEPROM: 1 KB (ATmega328).
- Clock Speed: 16 MHz

Fig. 3.1: Arduino UNO R3 (reference part 21 of fig 2).

3.2. Raspberry Pi 3 B+

Specifications:

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU.
- 1GB RAM.
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board.
- 100 mbps Base Ethernet.
- 40-pin extended GPIO.
- 4 USB 2.0 ports.
- 4 Pole stereo output and composite video port.
- Full size HDMI.
- CSI camera port for connecting a Raspberry Pi 3 B+ camera.
- DSI display port for connecting a Raspberry Pi 3 B+ touchscreen display.
- Micro SD port for loading your operating system and storing data.
- Upgraded switched Micro USB power source up to 2.5A.

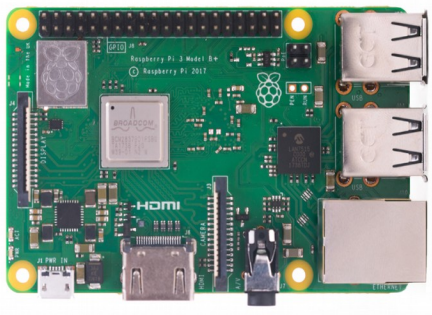


Fig. 3.2: Raspberry Pi 3 B+.

3.3. Connection Details

3.3.1. CN1 Connector: 40 pin 20X2 Header is connected to the Raspberry pi 3 B+.

Pin	Description	Pin	Description
1	NC	2	NC
3	SDA	4	NC
5	SCL	6	GND
7	P7/GPIO 7 GPCLK0	8	P8/GPIO 15 TXD (UART)
9	NC	10	P14/GPIO 16 RxD (UART)
11	P11/GPIO 0	12	P12/GPIO 1 CM_CLK/PWMO
13	P13/GPIO 2	14	NC
15	P15/GPIO 3	16	NC
17	NC	18	NC
19	P19/GPIO 12 MOSI (SPI)	20	NC
21	P21/GPIO 13 MISO (SPI)	22	NC
23	P23/GPIO 14 SCLK (SPI)	24	P24/GPIO 10 CEO (SPI)
25	NC	26	NC
27	NC	28	NC
29	P29/GPIO 21 GPCLK1	30	NC
31	P31/GPIO 22 GPCLK2	32	NC
33	P33/GPIO 23 PWM1	34	NC
35	P35/GPIO 24 PCM_FS/PWM1	36	P36/GPIO 27
37	P37/GPIO 28	38	P38/GPIO 28 PCM_DIN
39	NC	40	NC

3.3.2. CN3 Connector: 10 Pin interface connector.

Pin	Description	Pin	Description
1	SCL	6	VCC
2	SDA	7	P24
3	P12	8	P19
4	P15	9	P21
5	GND	10	P23

3.3.3. CN2 Connector: 10 pin header for LCD interface.

Pin	Description	Pin	Description
1	P38	6	VCC
2	P36	7	P29
3	P37	8	P31
4	P35	9	P7
5	GND	10	P33

3.3.4. CN4 Connector: Switches and LED's connector.

Pin	Description	Pin	Description
1	LED-4	6	VCC
2	LED-3	7	SWITCH-1
3	LED-2	8	SWITCH-2
4	LED-1	9	SWITCH-3
5	GND	10	SWITCH-4

3.4. Relimate Details

3.4.1. RM1: 4 pin relimate for RTC Interface.

Pin	Description	Pin	Description
1	SDA	3	GND
2	SCL	4	VCC

3.4.2. RM2: 4 pin relimate for DHT11 Interface.

Pin	Description	Pin	Description
1	P7	3	GND
2	P31	4	VCC

3.4.3. RM3: 4 pin relimate for LDR Interface.

Pin	Description	Pin	Description
1	P33	3	GND
2	P7	4	VCC

3.4.4. RM4: 4 pin relimate for Ultrasonic Interface.

Pin	Description	Pin	Description
1	P37	3	GND
2	P35	4	VCC

3.4.5. RM5: 4 pin relimate for Gas Sensor Interface.

Pin	Description	Pin	Description
1	P11	3	GND
2	P36	4	VCC

3.4.6. RM6: 4 pin relimate for Soil Moisture Sensor Interface.

Pin	Description	Pin	Description
1	P13	3	GND
2	P38	4	VCC

3.4.7. RM7: 4 pin relimate for ESP8266 Wi-Fi Interface.

Pin	Description	Pin	Description
1	P31	3	GND
2	P29	4	VCC

3.4.8. RM8: 4 pin relimate for Bluetooth Interface.

Pin	Description	Pin	Description
1	P19	3	GND
2	P24	4	VCC

3.4.9. RM9: 4 pin relimate for Buzzer Interface.

Pin	Description	Pin	Description
1	P36	3	GND
2	P38	4	VCC

3.4.10. RM10: 4 pin relimate for RTC Connector.

Pin	Description	Pin	Description
1	SDA (A4)	3	GND
2	SCL (A5)	4	VCC

3.4.11. RM12: 4 pin relimate for MCP3008 CH0.

Pin	Description	Pin	Description
1	A00	3	GND
2	NC	4	VCC

3.4.12. RM13: 4 pin relimate for MCP3008 CH1.

Pin	Description	Pin	Description
1	A11	3	GND
2	NC	4	VCC

3.4.13. RM14: 4 pin relimate for MCP3008 CH2.

Pin	Description	Pin	Description
1	A22	3	GND
2	NC	4	VCC

3.4.14. RM15: 4 pin relimate for MCP3008 CH3.

Pin	Description	Pin	Description
1	A33	3	GND
2	NC	4	VCC

3.4.15. RM16: 4 pin relimate for ESP8266 Wi-Fi Module Connector.

Pin	Description	Pin	Description
1	TX	3	GND
2	RX	4	VCC

3.4.16. RM17: 4 pin relimate for Relay and Buzzer Interface. P_36 is O/p of ULN2003 IC used for Relay and P_38 is O/p of ULN2003 IC used for Buzzer.

Pin	Description	Pin	Description
1	P_36	3	GND
2	P_38	4	VCC

3.4.17. RM18: 4 pin relimate for Bluetooth Connector.

Pin	Description	Pin	Description
1	TX (D11)	3	GND
2	RX (D10)	4	VCC

3.4.18. RM19: 4 pin relimate for DHT11 Connector.

Pin	Description	Pin	Description
1	DATA	3	GND
2	NC	4	VCC

3.4.19. RM20: 4 pin relimate for Light Sensor Connector.

Pin	Description	Pin	Description
1	D0	3	GND
2	NC	4	VCC

3.4.20. RM21: 4 pin relimate for Ultrasonic Sensor Connector.

Pin	Description	Pin	Description
1	TRIG	3	GND
2	ECHO	4	VCC

3.4.21. RM22: 4 pin relimate for Gas Sensor Connector.

Pin	Description	Pin	Description
1	A0	3	GND
2	D0	4	VCC

3.4.22. RM23: 4 pin relimate for Soil Moisture Connector.

Pin	Description	Pin	Description
1	A0	3	GND
2	D0	4	VCC

3.4.23. RM24: 3 pin relimate UART Interface for Arduino-Uno.

Pin	Description	Pin	Description
1	P19	3	GND
2	P24		

3.4.24. RM25: 3 pin relimate for UART.

Pin	Description	Pin	Description
1	RX	3	GND
2	TX		

3.4.25. RM26: 3 pin relimate UART Interface for Raspberry.

Pin	Description	Pin	Description
1	P14	3	GND
2	P8	4	VCC

3.4.26. J1: 32 pin Male Burg for Arduino Connector.

Pin	Description	Pin	Description
1	NC	17	P29/D2
2	IOREF	18	P31/D3
3	RESET	19	P7/D4
4	3V3_1	20	P33/D5
5	VCC_1	21	P35/D6
6	NC	22	P37/D7
7	GND_1	23	P36/D8

8	VIN	24	P38/D9
9	P11/A0	25	P24/D10
10	P13/A1	26	P19/D11
11	P15/A2	27	P21/D12
12	P12/A3	28	P23/D13
13	SDA/A4	29	NC
14	SCL/A5	30	AREF
15	P14/D0	31	SDA
16	P8/D1	32	SCL

3.4.27. J2: SPI ADC LINES single row berg pins

Pin	Description	Pin	Description
1	A44	2	A55
3	A66	4	A77

3.4.28. BR1: 4 pin Female Burg for Bluetooth Connector slot.

Pin	Description	Pin	Description
1	TXD	3	GND
2	RXD	4	VCC

3.4.29. J5: 3 pin Female Burg for DHT11 Connector slot.

Pin	Description	Pin	Description
1	VCC	3	GND
2	DATA		

3.4.30. J6: 3 pin Female Burg for Light Sensor Connector slot.

Pin	Description	Pin	Description
1	VCC	3	D0
2	GND		

3.4.31. J7: 4 pin Female Burg for Ultrasonic Connector slot.

Pin	Description	Pin	Description
1	GND	3	TRIG
2	ECHO	4	VCC

3.4.32. J8: 4 pin Female Burg for Gas Sensor Connector slot.

Pin	Description	Pin	Description

1	VCC	3	D0
2	GND	4	A0

3.4.33. RM23A: 4 pin Female Burg for Soil Moisture.

Pin	Description	Pin	Description
1	VCC	3	D0
2	GND	4	A0

3.4.34. J10: 6 pin Male Burg for SPI Connector slot (3×2).

Pin	Description	Pin	Description
1	MISO	2	VCC
3	SCK	4	MOSI
5	RESET	6	GND

3.4.35. J11: 6 pin Male Burg for SPI Connector slot (1×6).

Pin	Description	Pin	Description
1	D13	4	D10
2	D12	5	GND
3	D11	6	VCC

3.4.36. J12: 12 pin Male Burg for Extra Arduino Digital Pins (From D2 to D13).

Pin	Description	Pin	Description
1	D2	7	D8
2	D3	8	D9
3	D4	9	D10
4	D5	10	D11
5	D6	11	D12
6	D7	12	D13

3.4.37. J13: 6 pin Male Burg for Extra Arduino Analog Pins.

Pin	Description	Pin	Description
1	A0	6	A3
2	A1	7	A4
3	A2	8	A5

4. INSTALLATION PROCEDURE

4.1. Install the Arduino Software (IDE) on Windows PCs

Get the latest version from https://www.arduino.cc/en/Main/Software. You can choose between the Installer (.exe) and the Zip packages.
When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

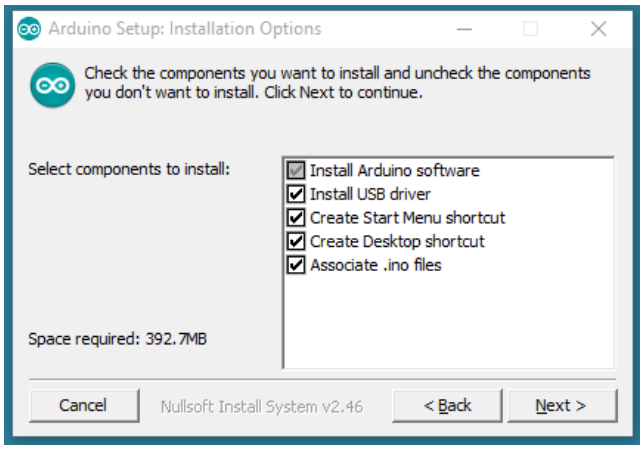


Fig. 4.1: Arduino initial setup.

Choose the components to install.

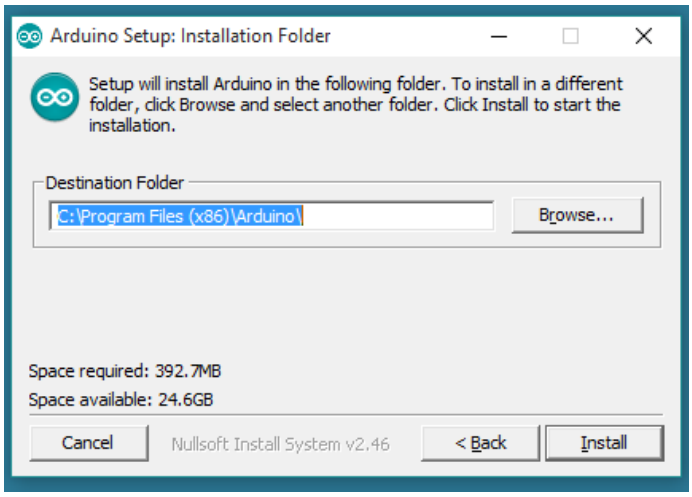


Fig. 4.2: Select folder Arduino installation.

Choose the installation directory (we suggest to keep the default one)

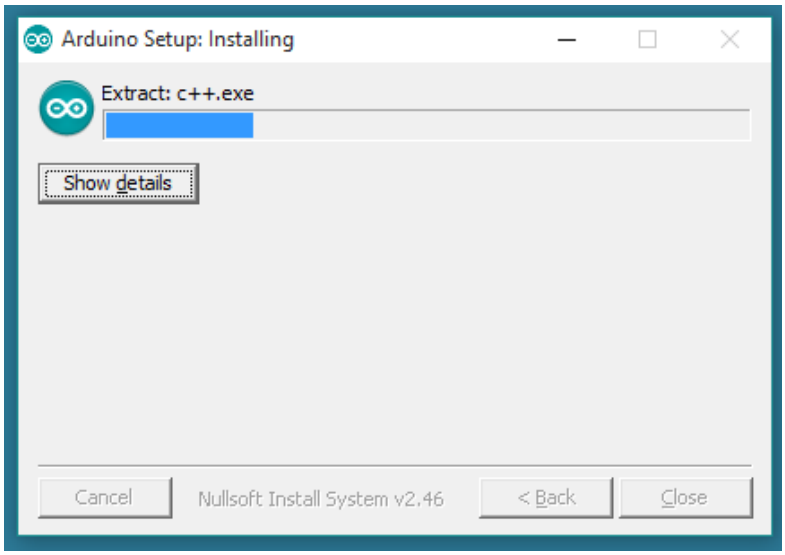


Fig. 4.3: Arduino IDE installing

The process will extract and install all the required files to execute properly the Arduino Software (IDE)

4.1.1. Using Arduino Uno with Arduino Desktop IDE

Connect Uno board with an A B USB cable; sometimes this cable is called a USB printer cable.

Fig. 4.4: A B USB Cable.

4.1.2. Install the Board Drivers

If you used the Installer, Windows – from XP up to 10 – will install drivers automatically as soon as you connect your board.

Troubleshooting:

If you downloaded and expanded the Zip package or, for some reason, the board wasn't properly recognized, please follow the procedure below.

- Click on the Start Menu, and open up the Control Panel.
- While in the Control Panel, navigate to System and Security. Next, click on System. Once the System window is up, open the Device Manager.
- Look under Ports (COM & LPT). You should see an open port named "Arduino UNO (COMxx)". If there is no COM & LPT section, look under "Other Devices" for "Unknown Device".

- Right click on the "Arduino UNO (COMxx)" port and choose the "Update Driver Software" option.
- Next, choose the "Browse my computer for Driver software" option.
- Finally, navigate to and select the driver file named "arduino.inf", located in the "Drivers" folder of the Arduino Software download (not the "FTDI USB Drivers" sub-directory). If you are using an old version of the IDE (1.0.3 or older), choose the Uno driver file named "Arduino UNO.inf"
- Windows will finish up the driver installation from there.

4.1.3. Getting Started with Sketch

Open the LED blink example sketch: File > Examples >01.Basics > Blink.

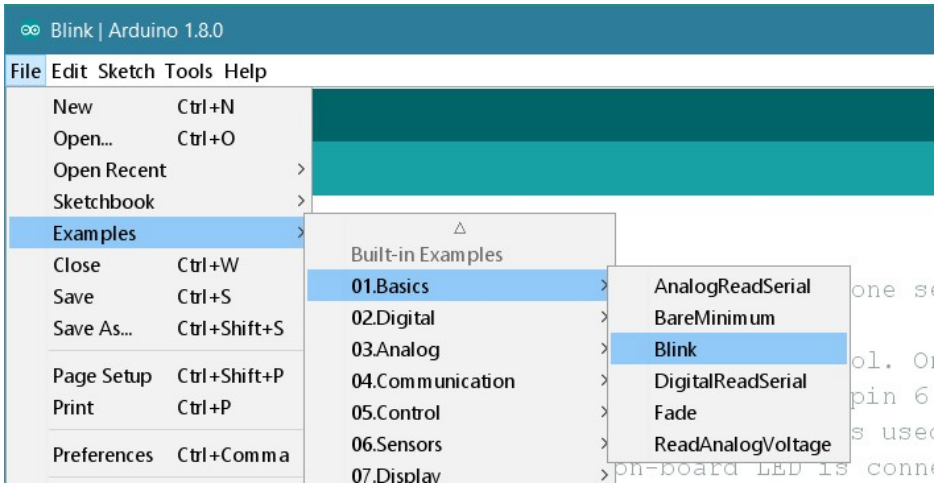


Fig. 4.5: Arduino IDE Examples

Select the Board Type and Port

You'll need to select the entry in the Tools > Board menu that corresponds to your Arduino or Genuino board.

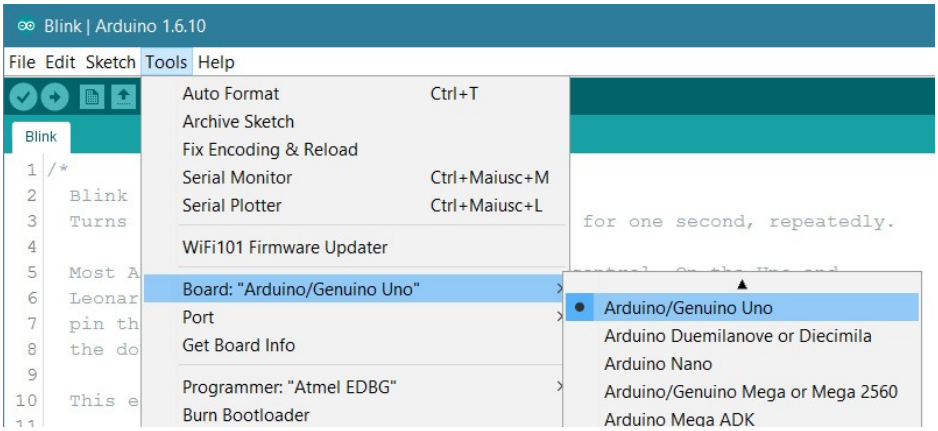


Fig. 4.6: Arduino IDE Select Board

Select the serial device of the board from the Tools | Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find

out, you can disconnect your board and re-open the menu; the entry that disappears should be the Arduino or Genuino board. Reconnect the board and select that serial port.

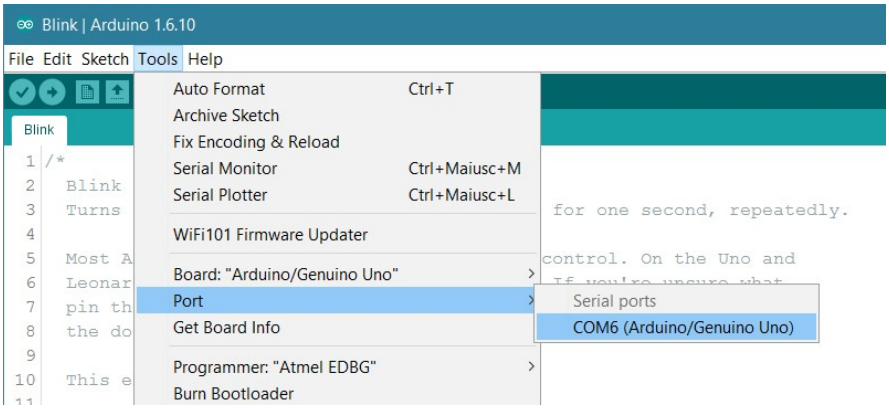


Fig. 4.7: Arduino IDE Select Port

Upload the Program

Now, simply click the "Upload" button in the environment. Wait for few seconds - you should see the RX and TX LEDs on the board flashing. If the upload is successful, the message "Done uploading." will appear in the status bar.

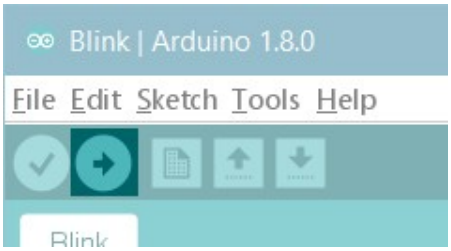


Fig: 4.8 Arduino IDE Upload Project

A few seconds after the upload finishes, you should see the pin 13 LED on the board starts blinking (in orange). If it does, congratulations!

4.2. Installing Raspbian OS for Raspberry Pi 3 B+

4.2.1. Using Raspbian Stretch OS Image Downloaded in. zip format

1. Download the image file in the form Zip file from the following link
   https://www.raspberrypi.org/downloads/raspbian/

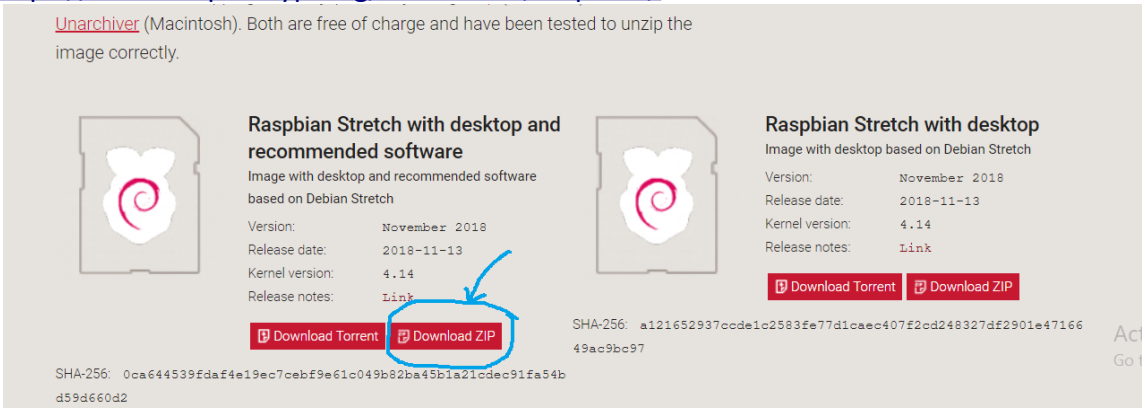


Fig. 4.12: Download Raspbian Stretch

2. Download the win32 Disk Imager Software to write the image file to SD Card from the following link. https://sourceforge.net/projects/win32diskimager/
3. Now insert the SD Card in SD card slot and connect it to PC/Laptop.
4. Open Win32 Disk Imager, choose the .img or image file you want to write as Image File and choose the SD drive as Device and press Write.
5. The write may take a while. Once it is done, remove the SD card and insert it into the device you want to use with.

Initial Setup for Raspberry Pi 3 B+

1. Once Raspberry Pi 3 B+ is booted up to GUI, Connect the keyboard and Mouse to USB slots.
2. Change the following
   a. language: English
   b. Country and Time Zone: India and Kolkata
3. From menu-->Preferences-->Raspberry Pi 3 B+ Configuration.
   Then Go To Interfaces
   Enable Camera, SSH, VNC, SPI, I2C, Serial Port. Don't Enable Remaining Things.
4. Connect to WiFi.
5. Check the date and Time.
6. Open the Terminal.
   Execute following default commands.
   a. ***sudo apt-get update***
   b. ***sudo apt-get upgrade***
7. Now you can write your programs at Thonny Python IDE.
   Menu → Programming → Thonny Python IDE.
   File → New ….. You can write your Program here and save.
8. Execute the program at terminal only by following command
   sudo python <program-name>.py

5. ON BOARD INTERFACES

5.1. LCD Display

A Liquid Crystal Display is an electronic display module. It displays 16 characters per line in 2 such lines. Each character in LCD displays 5×7 pixel matrix. Here is the LCD is interfaced using 4 – bit mode.

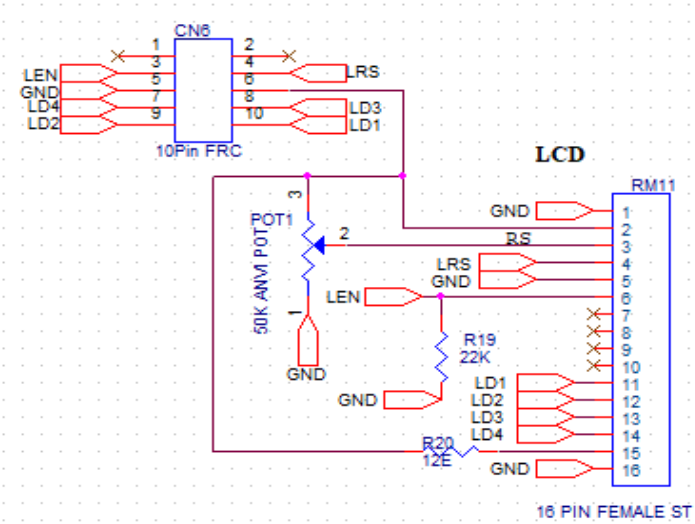


Fig. 5.1: Schematic pin diagram of LCD Interface.

Pin	Function	Name
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	Vcc
3	Contrast adjustment; the best way is to use a variable resistor.	Vo / VEE
4	Selects command register when low, and data register when high	RS (Register Select)
5	Low to write to the register; High to read from the register	Read/write
6	Sends data to data pins when a high to low pulse is given;	Enable
7-14	8-bit data pins	DB0-DB7
15	Back-light VCC (5V)	LED+
16	Back-light Ground (0V)	LED-

5.2. 7 – Segment Display

A 7-segment display is used for displaying decimal numerals. The seven-segment display consists of 7 LED's, so called 7-segment display. 7-segment display is manufactured in a rectangular fashion.

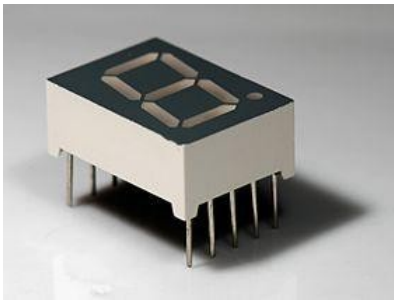


Fig. 5.2: 7-Segment Display(reference part)

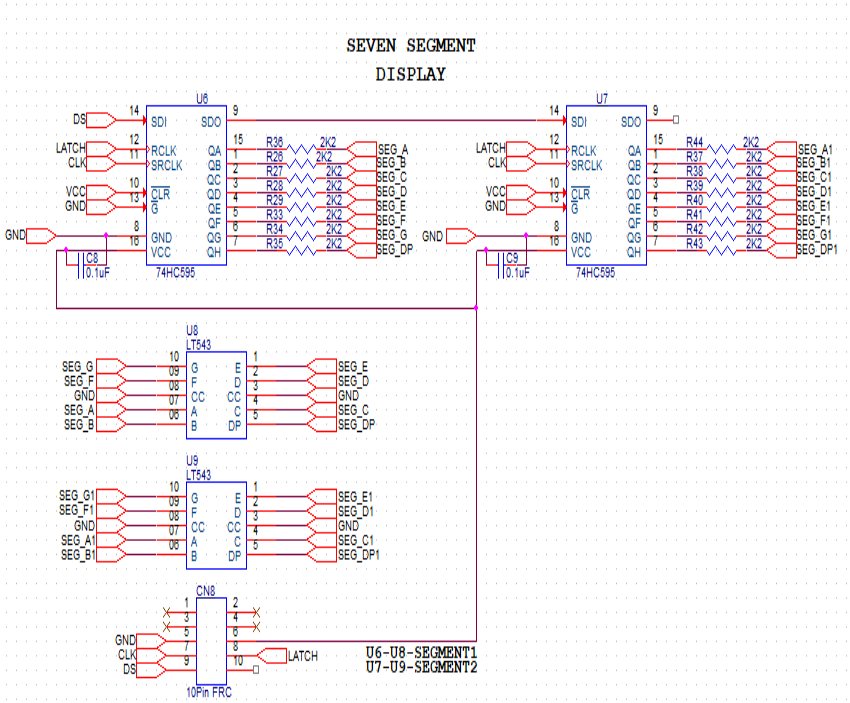


Fig. 5.3: Schematic pin diagram of 7-Segment Display.

74HC595 shift register

For saving the pin number for controlling a 7-segment display, a shift register is used as a serial-to-parallel converter to send signals to the display. That is, we serially send 8 bits of data, which represents the way we want to turn on the display, by one signal pin into the shift register and the register can output the corresponding data pattern to its 8 output pins at once (parallel). Signal pins we need to take care about when using a 74HC595.

Input pins

- 1 input data pin (DS)
- 1 clock pin for input data (SHCP)
- 1 clock pin for output data (STCP)

Output pins

- 8 data pins (Q0 ~ Q7)

Control pins

- 1 output enable pin (OE)
- 1 data reset pin (MR)

5.3. RTC

The DS1307 real time clock (RTC) IC is an 8-pin IC. It is a low-power clock/calendar with 56 bytes of battery backup SRAM. The clock provides the qualified data of year, months, date, day, hours, minutes and seconds. The end date of the month will be automatically adjusted.

They are available as ICs and supervise timing like a clock. Also it will operate date like a calendar. The clock/calendar running even if there is power failure, because the RTC have an arrangement of battery backup. This is the advantage of RTC. An exceptionally little current is required for keeping the RTC animated. We can find these RTCs in many applications like embedded systems and computer motherboards, etc.

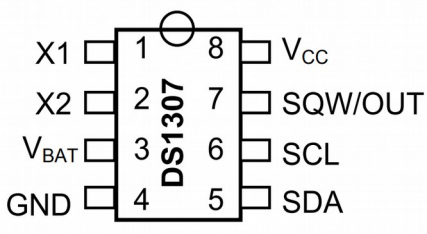


Fig. 5.4: RTC (reference part 13)

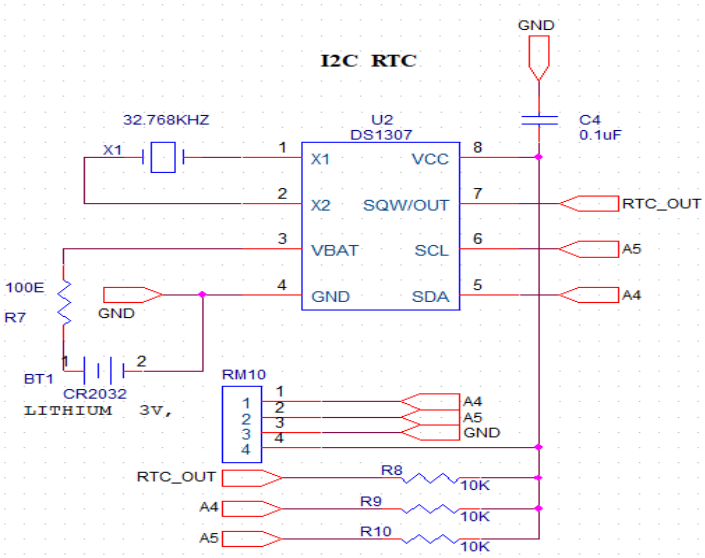


Fig. 5.5: Schematic Pin Diagram of RTC.

In the above schematic circuit, the two inputs X1 and X2 are connected to a 32.768 kHz crystal oscillator as the source for the chip. VBAT is connected to positive culture of a 3V battery chip. Vcc power to the I2C interface is 5V. If the power supply VCC is not granted read and writes are inhibited.

5.4. ESP8266-01

The Wi-Fi module ESP8266 ESP-01 that allows micro-controllers access to a Wi-Fi network. ESP8266 ESP-01 module is a self-contained SOC (System on a Chip) that doesn't required a micro-controller to manipulate inputs and outputs. You can normally do with an Arduino. The ESP8266 ESP-01 module has three operation modes:

1. **Access Point (AP)**
2. **Station (STA)**
3. **Both**

In **Access Point** the Wi-Fi module acts as a Wi-Fi network, or access point, allowing other devices to connect to it. Access Point simply establishes a two-way communication between the ESP8266 and the device that is connected to it through Wi-Fi.

In **Station (STA)** mode, the ESP-01 can connect to an Access Point such as the WiFi network from your house. This allows any device connected to that network to communicate with the module.

The third mode of operation permits the module to act as both an **Access Point** and a **Station (STA)**.

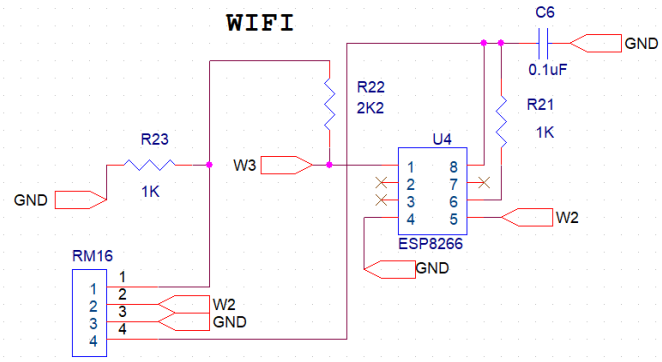


Fig. 5.6: Schematic Pin Diagram of ESP8266.

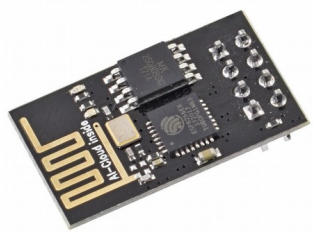


Fig. 5.7: ESP8266(reference part).

5.5. Switches and LED's

Switch and LED's are combination of 4 LED's and 4 switches which are programmed to operate LED's using the switches.

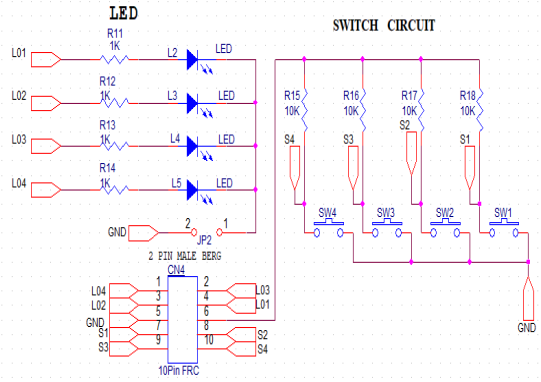


Fig. 5.8: Schematic Pin Diagram of Switches and LED's.

5.6. 4×4 Keypad Matrix

The Purpose of having the 4*4 Keypad Matrix is to reduce the number of input pins. The order of matrix increases with the decreases of the number of inputs.

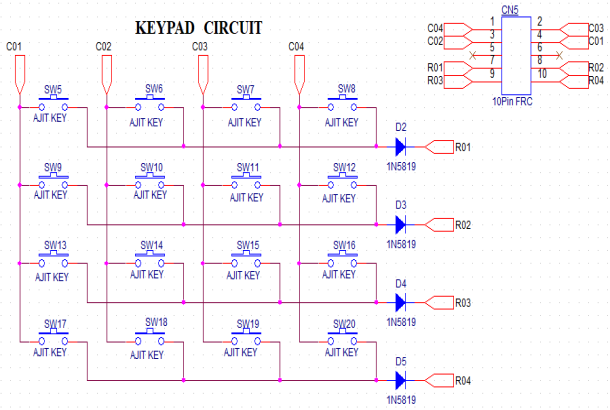


Fig. 5.9: Schematic Pin Diagram of 4*4 Keypad Matrix.

5.7. Stepper Motor

Stepping motor is a brush-less synchronous electric motor that converts digital pulses into mechanical shaft rotation. The motor's position can then be commanded to move and hold at one of these steps without any position sensor for feedback (an open-loop controller), as long as the motor is carefully sized to the application in respect to torque and speed.

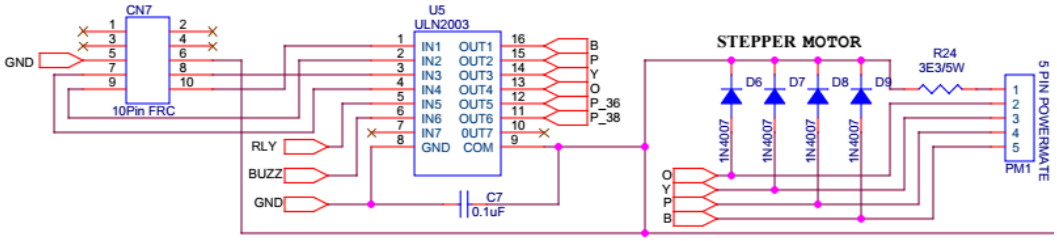


Fig. 5.11: Schematic Pin Diagram of Stepper Motor Interface.

5.8. Buzzer

Buzzer is an output module which can be used to show the result of any program as a sound. Providing VCC to BUZZ pin will blow the Buzzer.

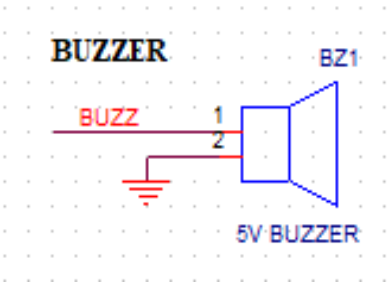


Fig. 5.12: Schematic Pin Diagram of Buzzer.

5.9. Bluetooth

Bluetooth is used for standard for exchanging data over short distances using short wavelength UHF radio waves in the ISM band from 2.400 to 2.485 GHz from fixed and mobile devices, and building personal area networks (PANs). It is wireless technology. It was originally conceived as a wireless alternative to RS-232 data cables.

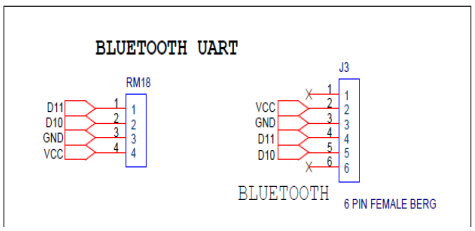


Fig. 5.13: Schematic Pin Diagram of Bluetooth.

5.10. Relay

A relay works to isolate or change the state of an electric circuit from one state to another. Relays use an electromagnet to mechanically operate a switch, but other operating principles are also used, such as solid-state relays. Relay are used to to control a circuit by a separate low-power signal.

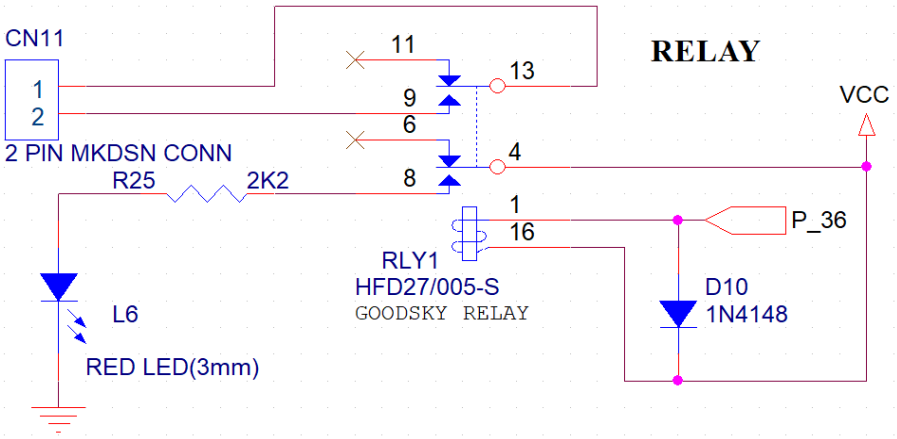


Fig. 5.14: Schematic Pin Diagram of Relay.

5.11. UART (On Board RS232- UART Using MAX3232)

A universal asynchronous receiver-transmitter is a computer hardware device for asynchronous serial communication in which the data format and transmission speeds are configurable. A UART is usually an individual integrated circuit used for serial communications over a computer or peripheral device serial port. The System Serial Port is having Voltage Level is 12V. But Micro-controllers like Arduino/Raspberry Pi 3 B+ works with 5V / 3.3V Voltage Level.

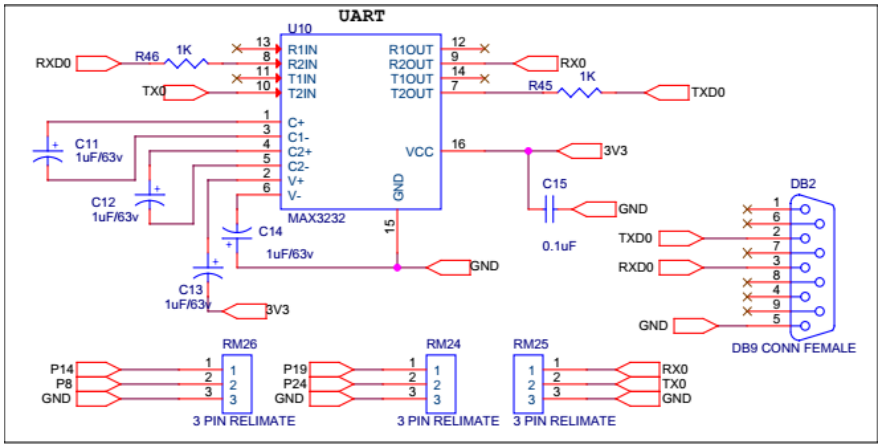


Fig. 5.15: Schematic Pin Diagram of UART.

5.12. Sensors

Sensors are the devices that are used to detect and respond to electrical or optical signals. Sensors converts the physical parameter into a signal which can be measured electrically. Let's explain the example of temperature. The mercury in the glass thermometer expands and contracts the liquid to convert the measured temperature which can be read by a viewer on the calibrated glass tube.

5.12.1. GAS Sensor

Gas Sensors (MQ2) are useful to detect the gas leakage (home and industry). It is suitable for detecting the Smoke, CO, H2, LPG, CH4, Alcohol. Due to fast response time and its high sensitivity, measurement can be taken as soon as possible. Potentiometer is used to adjust the sensitivity of the sensor.

Fig. 5.16: Schematic Pin Diagram of GAS Sensor.

Fig. 5.17: GAS Sensor

Specifications:

- Operating Voltage is +5V.
- Can be used to Measure or detect LPG, Alcohol, Propane, Hydrogen, CO and even methane.
- Analog output voltage: 0V to 5V.
- Digital Output Voltage: 0V or 5V (TTL Logic).
- Preheat duration 20 seconds.
- Can be used as a Digital or analog sensor.
- The Sensitivity of Digital pin can be varied using the potentiometer.

5.12.2. Soil Moisture Sensor

The dielectric constant of soil increases as the water content of the soil increases. The sensor is designed to estimate soil volumetric water content based on the dielectric constant of the soil. The dielectric constant can be thought of as the soil's ability to transmit electricity. The dielectric constant of soil increases as the water content of the soil increases. This response is due to the fact that the dielectric constant of water is much larger than the other soil components, including air. Thus, measurement of the dielectric constant gives a predictable estimation of water content.

Fig. 5.18: Schematic Pin Diagram of Soil Moisture Sensor.

Fig. 5.19: Ultrasonic Sensor.

Specifications:

Input Voltage 3.3 – 5V.
Output Voltage 0 – 4.2V.
Input Current 35mA.
Output Signal Both Analog and Digital.

5.12.3. Light Sensor

A Light Sensor generates an output signal indicating the intensity of light by measuring the radiant energy that exists in a very narrow range of frequencies basically called “light”, and which ranges in frequency from “Infrared” to “Visible” up to “Ultraviolet” light spectrum.

The light sensor converts this “light energy” whether visible or in the infrared parts of the spectrum into an electrical signal output. from “Infrared” to “Visible” up to “Ultraviolet” light spectrum. Sensitivity can be adjusted through potentiometer.

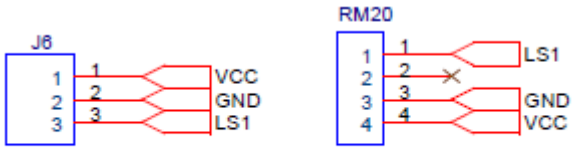


Fig. 5.20: Schematic Pin Diagram of Light Sensor.

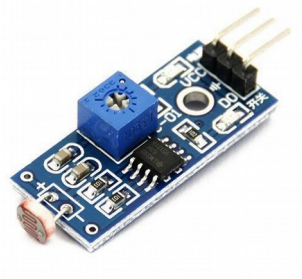


Fig. 5.21: Light Sensor.

Specifications:

- Using photosensitive resistance sensor sensitive type.

- Working Voltage:3.3V-5V.
- Output form: Digital Switch show (0 for Brightness and 1 for darkness).
- Fixed Bolt Hole, easy installation.

5.12.4. Ultrasonic Distance Sensor

An ultrasonic sensor is to measures the distance of an object using ultrasonic sound waves. An ultrasonic sensor uses a transducer to send and receive ultrasonic pulses that relay back information about an object's proximity.
High-frequency sound waves reflect from boundaries to produce distinct echo patterns.

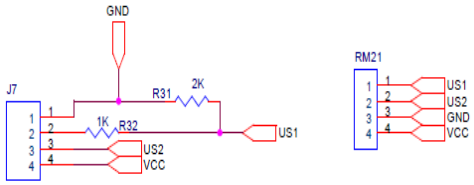


Fig. 5.22: Schematic Pin Diagram of Ultrasonic Distance Sensor.

Fig. 5.23: Ultrasonic Distance Sensor .

Specifications:

- Working Voltage: DC 5V.
- Working Current: 15mA.
- Working Frequency: 40Hz.
- Max Range: 4m.
- Min Range: 2cm.
- Measuring Angle: 15 degree.
- Trigger Input Signal: 10µS TTL pulse.
- Echo Output Signal Input TTL level signal and the range in proportion.

5.12.5. Temperature and Humidity Sensor

The DHT11 is a digital temperature and humidity sensor. It uses a humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin. It's fairly simple to use, but requires careful timing to grab data. You can only get new data from it once every 2 seconds, so when using our library, sensor readings can be up to 2 seconds old.

Specifications:

Operating Voltage: 3.5V to 5.5V.
Operating current: 0.3mA (measuring) 60uA (standby)

Output: Serial data.
Temperature Range: 0°C to 50°C.
Humidity Range: 20% to 90%
Resolution: Temperature and Humidity both are 16-bit.
Accuracy: ±1°C and ±1%

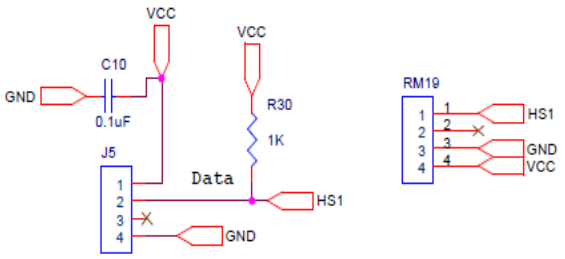


Fig. 5.24: Schematic Pin Diagram of Temperature and Humidity Sensor

Fig. 5.25: Temperature and Humidity Sensor.

6. DEMO PROGRAMS

General Guidelines:

1. Do not power on Raspberry Pi 3 B+ and Arduino Boards at the same time. Connect Power to either Raspberry Pi 3 B+ or Arduino board.
2. Do not connect CN2 and CN9 at a time (or) CN3 and CN10 at a time as CN9 is duplication of CN2 and CN10 is duplication of CN3.
3. Make sure that all the specified connectors are connected before executing any program.
4. For all the programs please refer **Xtrans IoT Cloud**.

6.1. For Raspberry Pi 3 B+

Do's

1. Power on the Kit using 5V 2.5A Adapter provided along with the Kit.
2. Connect the Raspberry Pi with Keyboard, Mouse to USB Ports and Monitor to HDMI Port before powering Raspberry Pi 3 B+.
3. Now, Power on Raspberry pi 3B+ with USB to Mini USB Cable provided with kit.
4. As soon as Raspberry Pi is Booted up check the proper internet connection through connecting Wi-Fi or Ethernet cable.
5. Remove the Arduino completely out of the kit before executing programs on Raspberry Pi 3 B+.

Don't s

Connect Only specified connectors while executing particular programs and remove all other connections.

1. Don't power off the Raspberry Pi directly. Shutdown in proper way.

NOTE: ***All the below explained programs will run in a terminal window***

6.1.1. To Test LCD

Connection: Connect 10 pin FRC cable between CN3 to CN6

Connected Pin details: RS – P15, EN -P12, D4 – P23, D5 – 21, D6 – P19, D7 – P24

To run this test type the command as below:

$ sudo python lcd.py

Result: It will display 4 different messages

"Raspberry Pi"	first message
"16x2 LCD Test"	
"1234567890123456"	second message
"abcdefghijklmnop"	
"RaspberryPi-spy"	third message
".co.uk"	
"Follow me on"	fourth message
"Twitter @RpiSpy"	

if you terminate the program by entering **^C. "Goodbye!"** will be displayed.

6.1.2. To Test 7 Segment Display

Connection: Connect 10 pin FRC cable between CN2-CN8

Connected Pin details: Data – P7, Latch – P31, Clock – P29

To run this test type the command as below:

$ sudo python seven-segment.py

Result: It will display 0-99

you can terminate the program by entering **"^C".**

6.1.3. To Test UART-RELAY

Connection: Connect 4 pin relimate cable between RM25 – RM26, RM17 – RM9.

Connected Pin details: RxD – P14, TxD – P8, P36.

Open COM Port Through one of the App like Hyper Terminal / Hercules. Set the baud rate to 9600 and Force on Local Echo.

$ sudo python uart-relay.py

Result: It can communicate (Send/Receive) from laptop/ PC through RS232 to UART Converter to Raspberry Pi Serial Monitor. Send 1 or 0 from the serial port and the relay will switch On or Off and the LED will light On when Relay is ON and turn Off when relay is turned Off. The status is also displayed on the console monitor. You can terminate the program by entering **"^C".**

6.1.4. To Test UART-Buzzer

Connection: Connect 4 pin relimate cable between RM25 – RM26, RM17 – RM9.

Connected Pin details: +ve – Vcc, -ve – GND and P38.

To run this test type the command as below:

$ sudo python uart-bz.py

Result: It can communicate (Send/Receive) from laptop/ PC through RS232 to UART Converter to Raspberry Pi Serial Monitor. Send 1 or 0 from the serial port and the Buzzer will blow when high and turn of when low is provided to the pin. The status is also displayed on the console monitor. You can terminate the program by entering **"^C".**

6.1.5. To Test Stepper Motor

Connection: Connect 10 pin FRC cable between CN10-CN7.

Connected Pin details: IN1 – P33, IN2 – P7, IN3 – P31, IN4 – P29.

To run this test type the command as below:

$ sudo python stepper_motor.py

Result: Stepper Motor rotates in clockwise direction.

you can terminate the program by entering **"^C".**

6.1.6. To Test 4X4 Keypad Matrix

Commands to be executed: ***sudo pip install pad4pi***

Connection: Connect 10 pin FRC cable between CN9-CN5.

Connected Pin details: C4 -P38, C3 – P36, C2 – P37, C1 – P35, R4 – P33, R3 – P7, R2 – P31, R1 – P29.

To run this test type the command as below:

$ sudo python keypad.py

Result: Pressed key will be displayed in console.

you can terminate the program by entering **"^C".**

6.1.7. To Test Switches and LED's

Connection: Connect 10 pin FRC cable between CN9-CN4.

Connected Pin details: LED1 – P15, LED2 – P12, LED3 – P3, LED4 – P5, SW1 – P24, SW2 – P19, SW3 – P21, SW4 – P23.

Note: short the JP2 jumper pins before executing.

To run this test type the command as below:

$ sudo python led-switch.py

Result: Corresponding LED will glow with respect to the pressed key.

you can terminate the program by entering **"^C".**

6.1.8. To Test RTC

Connection: Connect 4 pin relimate cable between RM1-RM10.

Connected Pin details: SDA – P3, SCL – P5.

Initial setup:

Step 1: Run ***sudo nano /boot/config.txt*** to edit the pi configuration and add the RTC chip. Add below statement last line of the file.

dtoverlay =i2c-rtc,ds1307

Press CTL+X, Press Y, Press Enter

Step2: Save it and run ***sudo reboot*** to start again.

Log in and run ***sudo i2cdetect -y 1*** and observe the **UU** in the location 68

Step 3: Disable the "fake hwclock" which interferes with the 'real' hwclock

sudo apt-get -y remove fake-hwclock

sudo update-rc.d -f fake-hwclock remove

sudo systemctl disable fake-hwclock

Now with the fake-hwclock off, you can start the original 'hardware clock' script**.**

Step 4: Run ***sudo nano /lib/udev/hwclock-set*** and comment out these three lines:

#if [-e /run/systemd/system] ; then

 # exit 0

#fi

Press CTL+X, Press Y, Press Enter

sudo reboot

Step 5: When you first plug in the RTC module, it's going to have the wrong time because it has to be set once.
You can always read the time directly from the RTC with
sudo hwclock -D -r

Step 6: To set the time in Raspberry pi using RTC Module execute following Command
sudo hwclock –w

Step 7: To read time from RTC Module execute following Command
sudo hwclock -r

To run this test type the command as below:
$ sudo python rtclcd.py
Result: Current time will be synced with controller and displayed on the console.

6.1.9. To Test GAS Sensor

Click on application menu then go to Preferences and select Raspberry Pi Configuration, a window will pop up, in that select interfaces. Now enable SPI. After this one has to reboot the Raspberry Pi.

Commands to be executed:
$ sudo apt-get update
$ sudo apt-get install build-essential python-dev python-smbus git
$ cd ~
$ git clone https://github.com/adafruit/Adafruit_Python_MCP3008.git
$ cd Adafruit_Python_MCP3008
$ sudo python setup.py install

Connection: Connect 4 pin relimate cable between RM22-RM12.
Connected Pin details: A0 – MCP-3008 Channel 0.
To run this test type the command as below:
$ sudo python gas.py
Result: Random gas value can be seen in console. Use any lighter which contains gas, bring the lighter near the sensor and observe the readings on the console.

6.1.10. To Test Soil Moisture Sensor

Connection: Connect 4 pin relimate cable between RM23 – RM13.
Connected Pin details: A0 – MCP3008 Channel 1
To run this test type the command as below:

$ sudo python soil_moisture.py

Result: It will show the moisture value in the console. Initially it will display 1023 (assuming that the soil moisture sensor is left open) on console. Just insert the sensor in the soil and it can measure moisture or water level content in it.

6.1.11. To Test Ultrasonic Distance Sensor

Connection: Connect 4 pin relimate cable between RM21 to RM4.

Connected Pin details: Trig – P37, Echo – P35.

To run this test type the command as below:

$ sudo python ultrasonic.py

Result: In console values in centimeter can be seen depending on the distance between object and ultrasonic sensor.

6.1.12. To Test Light Sensor

Connection: Connect 4 pin relimate cable between RM20 and RM3.

Connected Pin details: D0 – P33.

To run this test type the command as below:

$ sudo python ldr.py

Result: Displays according to light intensity. **"Light Detected"** or **"Light Not Detected".**

6.1.13. To Test Temperature and Humidity Sensor

Commands to be executed: ***sudo pip install Adafruit_DHT***

Connection: Connect 4 pin relimate cable between RM2-RM19.

Connected Pin details: data – P7.

To run this test type the command as below:

$ sudo python dht11.py

Result: Current Temperature and Humidity can be seen in console.

6.2. For Arduino

Do's

1. Do not forget to power off the Raspberry Pi 3.
2. Do not forget to remove 40 Pin Connector from the Raspberry Pi 3.
3. Connect external cross connector cable to power up Arduino UNO R3.
4. Connect Arduino serial cable to flash the code.
5. Connect only Specified Connectors and remove all others while executing particular program.
6. Open the Arduino IDE then Write and Save the Program.
7. Before Compiling the program select the Board as Arduino/Genuino UNO from Tools → Board.
8. Select the Arduino Port from Tools→ Port.
9. If you are not able see the port install the Arduino drivers again. (Refer Arduino Installation procedure section)

Don't s

1. Do not connect the 40 pin FRC cable to Raspberry Pi 3 while Arduino is powered on.
2. Do not power on Raspberry Pi 3 without powered off Arduino.

Procedure to add the Libraries

* Sketch → Include Library→ add ZIP library

6.2.1. To Test 7 – Segment Display

Connection: Connect 10 pin FRC cable between CN2-CN8.
Connected Pin details: Data – D12, Latch – D11, Clock – D10.
Library to be add: Shift Register library.
Result: It will Display Numbers from 00-99.

6.2.2. To Test GAS Sensor

Connection: Connect 4 pin relimate cable between RM22-RM5.
Connected Pin details: A0-A0
Result: Gas values between 0-1023 can be seen in console.

6.2.3. To Test Temperature and Humidity Sensor

Connection: Connect 4 pin relimate cable between RM2 – RM19.
Connected Pin details: Data – D4
Library to be add: DHT library.
Result: Current Temperature and Humidity can be seen in console.

6.2.4. To Test LCD

Connection: Connect 10 pin FRC cable between CN3-CN6.
Connected Pin details: RS – D6, EN – D7, D4 – D5, D5 – D4, D6 – D3, D7 – D2.
Result: It will display **WELCOME TO ALS** in Line1 and **BENGALURU – 58** in Line 2.

6.2.5. To Test Ultrasonic Distance Sensor

Connection: Connect 4 pin relimate cable between RM4 – RM21.
Connected Pin details: Trig – D6, Echo – D7.
Result: In console values can be seen depending on the distance between object and ultrasonic sensor.

6.2.6. To Test 4X4 Keypad Matrix

Connection: Connect 10 pin FRC cable between CN10-CN5.
Connected Pin details: C4–D9, C3–D8, C2–D7, C1–D6, R4–D5, R3–D4, R2–D3, R1–D2.
Library to be add: keypad register and shift register library.
Result: Pressed key number will be displayed on 7segment display.

6.2.7. To Test Light Sensor

Connection: Connect 4 pin relimate cable between RM20 – RM3.
Connected Pin details: D0 – D5.
Result: Boolean Value according to light intensity.

6.2.8. To Test Switches and LED's

Connection: Connect 10 pin FRC cable between CN9-CN4.
Connected Pin details: LED1 – A2, LED2 – A3, LED3 – A4, LED4 – A5, SW1 – D10, SW2 – D11, SW3 – D12, SW4 – D13.
Jumper: Short JP2/1&2.
Result: All the LED's will be glowing when the program is downloaded, when respective key is pressed the corresponding led will turn off.

6.2.9. To Test RTC

Connection: Connect 4 pin relimate cable between RM1 – RM10, CN3 to CN6.
Connected Pin details: SDA – A4, SCL – A5.
Library to be add: time_master library.
Result: Time entered in the program will be synced with controller and even after power off RTC will be running background. Time will be displayed on console and also LCD.

6.2.10. To Test Soil Moisture Sensor

Connection: Connect 4 pin relimate cable between RM6 – RM23.
Connected Pin details: A0 – A1.
Result: It will show the moisture value in the console.

6.2.11. To Test Buzzer

Connection: Connect 4 pin relimate cable between RM17 – RM9.
Connected Pin details: -VE - GND, +VE – D9.
Result: It'll blow when high/low provided to the pin.

6.2.12. To Test Stepper Motor

Connection: Connect 10 pin FRC cable between CN9-CN7.
Connected Pin details: IN1 – D5, IN2 – D4, IN3 – D3, IN4 – D2.
Library to be add: Accel stepper library.
Result: Stepper Motor rotates in clockwise and anti-clockwise direction.

6.2.13. To Test ESP8266 Wi-Fi Module

Connection: Connect 4 pin relimate cable between RM16 – RM7.
Connected Pin details: RX – D3, TX – D2.
After flashing the ESP8266 test program in the serial monitor set ESP8266 baud rate to 9600 using "AT+UART_DEF=9600,8,1,0,0" command.
Library to be add: ESP8266 wifi_master, Arduinojson_master, Thingspeak, Dht.lib library.
Result: We can Test All AT commands in Serial Monitor.

6.2.14. To Test Bluetooth Module

Connection: Connect 4 pin relimate cable between RM8 – RM18.
Connected Pin details: RxD – D10, TxD – D11.
Result: It can communicate from any mobile phone through **Serial Bluetooth Terminal App** to Arduino Serial Monitor.

6.2.15. To Test UART Module

Connection: Connect 4 pin relimate cable between RM25 – RM24.
Connected Pin details: RX – D10, TX – D11
Open COM Port Through one of the applications like Hyper Terminal (Windows XP), Hercules (Windows 10). Set the baud rate to 9600 and Force on Local Echo.
Result: It can communicate (Send/Receive) from laptop or desk top PC through RS232 to UART Converter to Arduino Serial Monitor.

6.2.16. To Test Relay Module

Connection: Connect 4 pin relimate cable between RM9-RM17
Connected Pin details: IN1 – D8.
Result: It controls an LED connected to Relay, LED will Blink.

7. COMBINATION OF COMPONENTS

DHT11	Light	Ultrasonic	Gas	Soil Moisture	Buzzer	RTC	ESP8266(only Arduino)	LCD
	Light	Ultrasonic	Gas	Soil Moisture	Buzzer	RTC	ESP8266(only Arduino)	7- Segment Display
	LED's	Buzzer	Stepper Motor					
Light	DHT11	Ultrasonic	Gas	Soil Moisture	Buzzer	RTC	ESP8266(only Arduino)	LCD
	DHT11	Ultrasonic	Gas	Soil Moisture	Buzzer	RTC	ESP8266(only Arduino)	7- Segment Display
	LED's	Buzzer	Stepper Motor					
Ultrasonic	Light	DHT11	Gas	Soil Moisture	Buzzer	RTC	ESP8266(only Arduino)	LCD
	Light	DHT11	Gas	Soil Moisture	Buzzer	RTC	ESP8266(only Arduino)	7-Segment Display
	LED's	Buzzer	Stepper Motor					
Gas	Light	Ultrasonic	DHT11	Soil Moisture	Buzzer	RTC	ESP8266(only Arduino)	LCD
	Light	Ultrasonic	DHT11	Soil Moisture	Buzzer	RTC	ESP8266(only Arduino)	7-Segment Display
	LED's	Buzzer	Stepper Motor					
Soil Moisture	Light	Ultrasonic	Gas	DHT11	Buzzer	RTC	ESP8266(only Arduino)	LCD
	Light	Ultrasonic	Gas	DHT11	Buzzer	RTC	ESP8266(only Arduino)	7- Segment Display
	LED's	Buzzer	Stepper Motor					
4*4 Matrix Keypad	LCD							
	7- Segment Display							
RTC	Buzzer							
	LCD							
ESP8266	DHT11	Ultrasonic	Light	Soil Moisture	Gas	LCD		
	Buzzer	Stepper Motor	LED's					

8. USE CASES

- Smart street light prototype using Ultrasonic and Light sensor.
- Live weather broadcasting using DHT11 and Xtrans Cloud/ThingSpeak cloud.
- Smart irrigation prototype using Soil Moisture and DHT11.
- Smart gas leakage email alerts using Gas sensor and Xtrans Cloud Events or with ThingSpeak alerts.
- Smart environment monitoring prototype using DHT11, Light and Gas sensor.
- Weather display system using DHT11 and LCD display.
- Automatic gate opening and closing system using Mobile app and stepper motor.
- Alarm Setting System using RTC, Switches and Buzzer.
- Read the sensor data when specified key is pressed.
- Automatic Fan/AC controlling system using DHT11 and Relay.
- Automatic Alert Through On Board LED or Buzzer by retrieving the values from Cloud.
- Object Distance Display System using 7- Segment Display and Ultrasonic.

9. EXTERNAL SENSORS SUPPORT

External sensor can be connected directly to the available Relimate socket or Male Berg pins. The below sensors will support both Arduino UNO R3 and Raspberry pi 3B+ modules.
Listed below are some sensors that can be connected externally to the IoT Development Kit. These are given as examples only. More sensors available in the market can also be connected.

1. IR distance sensor
2. Thermal detection sensor
1. SW-420 Motion Sensor/ Vibration switch
2. Humidity and Rain Detection Sensor
3. Speed Sensor
4. IR Flame Detection Sensor
5. SEN-0052 Magnetic Reed Sensor
6. Touch sensor
7. R305 Optical Fingerprint Sensor
8. Sound detector
9. Magnetic switch
10. Non-invasive AC current sensor
11. 3-Axis Accelerometer Sensor
12. Optical Dust Sensor
13. RFID Reader
14. Voltage / current meter
15. Hydrogen Gas Sensor
16. pH Sensor
17. Muscle Sensor
18. Pulse Sensor
19. Single Lead Heart Rate Monitor
20. Gyroscope
21. Sunlight Sensor
22. Water Temperature-DS18B20
23. Body Temperature

10. XTRANS IoT CLOUD

10.1 Introduction to Xtrans IoT Cloud

Welcome to our Xtrans IoT Cloud, a Private cloud designed and implemented by Xtrans Solutions, specially tuned for the IoT Development Kit. We will help you to connect your

devices and applications to the Xtrans IoT Cloud. Using http://alpha.iot.xtranssolutions.com this link you can connect to the Xtrans IoT Cloud.

Here are the few screenshots of Xtrans IoT Cloud.

Dashboard:

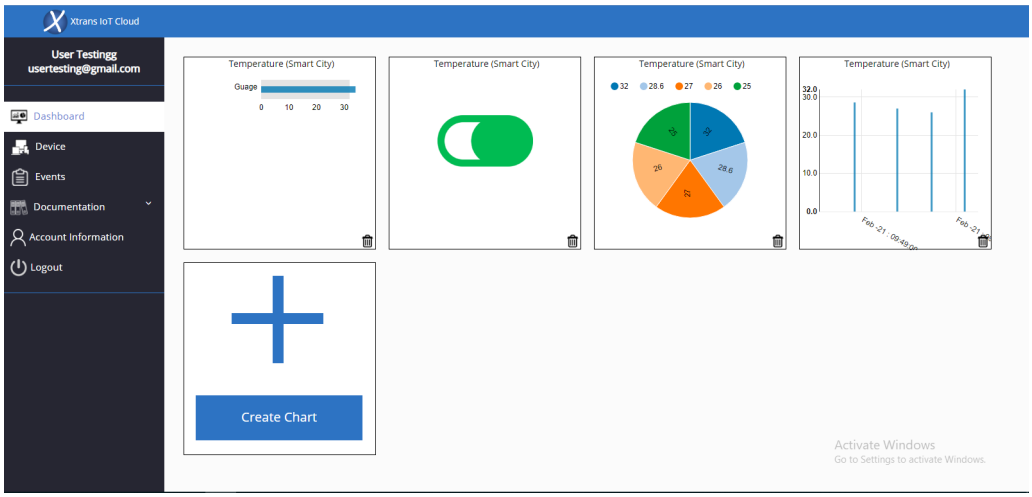


Fig. 10.1: Dashboard Page

Device:

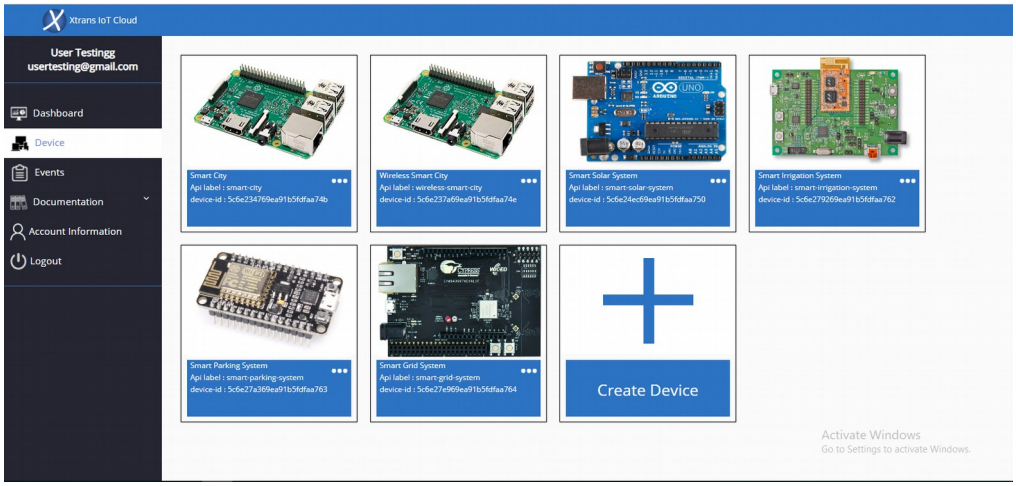


Fig. 10.2: Device Page

Sensor:

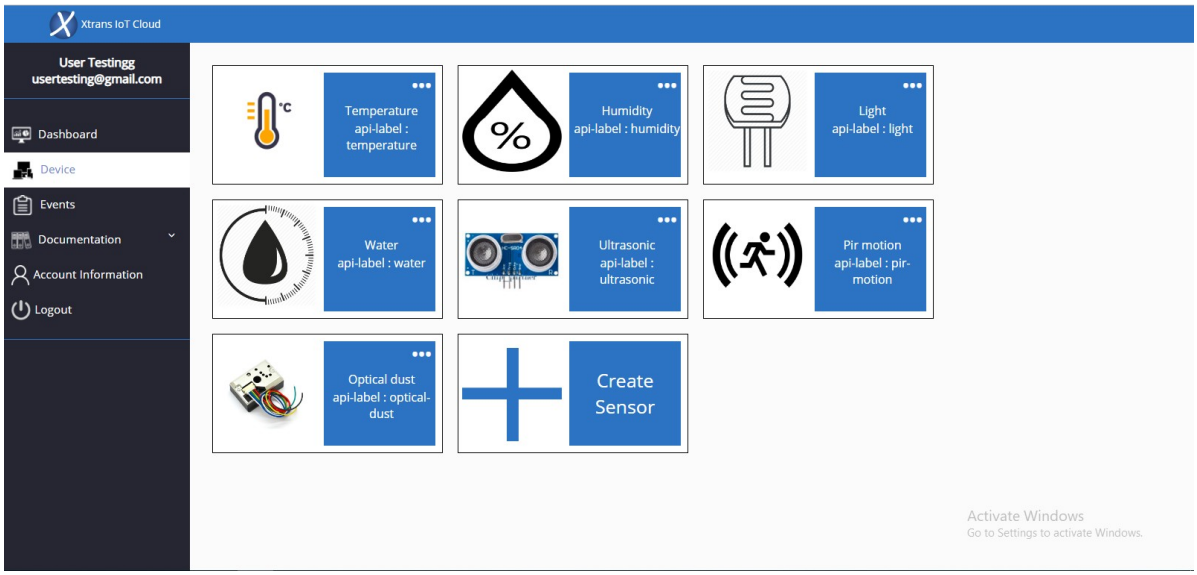


Fig. 10.4: Sensors Page

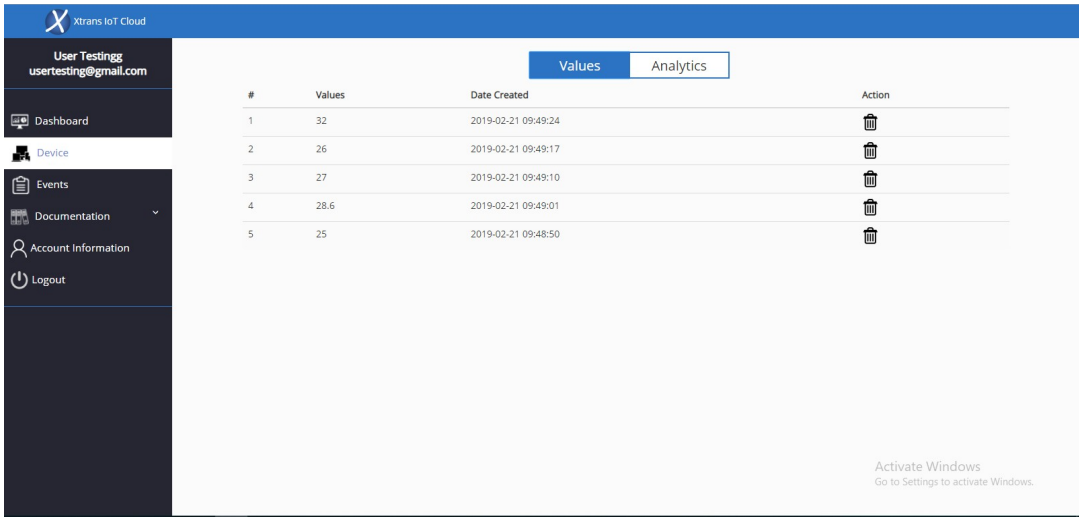


Fig. 10.5: Sensor Values Page

Welcome to Xtrans IoT Cloud REST API documentation. API allows for programmatic access to your accounts resources, including devices, variables, values and others.

10.2 Connecting the Device to Xtrans IoT Cloud

There are two ways to connect your device to Xtrans IoT Cloud. They are

1. Using HTTP
2. Using MQTT

10.2.1 Connecting the Device to Xtrans IoT Cloud Through HTTP

API access can be made over HTTP. Use the following endpoints for the Xtrans IoT Cloud.

Sending Data from Your IoT Device to Xtrans Cloud

Domain: "alpha.iot.xtranssolutions.com"
End-point: "/device/(device-label)?id=(device-id)”
Port: "3000"
Body: "[{"sensor": (sensor-label), "value": (sensor value)}]"

Where:

device-name: label of the device.
device-id: ID of the device.

10.2.2 Connecting the Device to Xtrans IoT Cloud Through MQTT

API access can be made over MQTT. Use the following endpoints for the Xtrans IoT Cloud.

Sending Data From Your IoT Device to Xtrans Cloud

Domain: "alpha.iot.xtranssolutions.com"
Topic: "/device/(device-id)"
Port: "1883"
Body: "[{"sensor": (sensor-label), "value": (sensor value)}]"

Where:

device-label: label of the device.
device-id: ID of the device.

11. THINGSPEAK CLOUD

11.1 Introduction

ThingSpeak is a cloud platform providing various services exclusively targeted for building IoT applications. It offers the capabilities of real-time data collection, visualizing the collected data in the form of charts, ability to create plugins and apps for collaborating with web services, social network and other APIs. We will consider each of these features in detail below.

The core element of ThingSpeak is a 'ThingSpeak Channel'. A channel stores the data that we send to ThingSpeak and comprises of the below elements:

- 8 fields for storing data of any type – These can be used to store the data from a sensor or from an embedded device.
- 3 location fields – Can be used to store the latitude, longitude and the elevation. These are very useful for tracking a moving device.
- 1 status field – A short message to describe the data stored in the channel.

To use ThingSpeak, we need to sign up and create a channel. Once we have a channel, we can send the data, allow ThingSpeak to process it and also retrieve the same. Let us start exploring ThingSpeak by signing up and setting up a channel.

Using https://thingspeak.com/ the link user can connect to the ThingSpeak Cloud.

Here are the few screenshots of ThingSpeak Cloud.

Create Channel: Once the user login to ThingSpeak account. Tap on the channel tab in the menu. follow these steps.

New channel button > Fill the details > Save Channel.

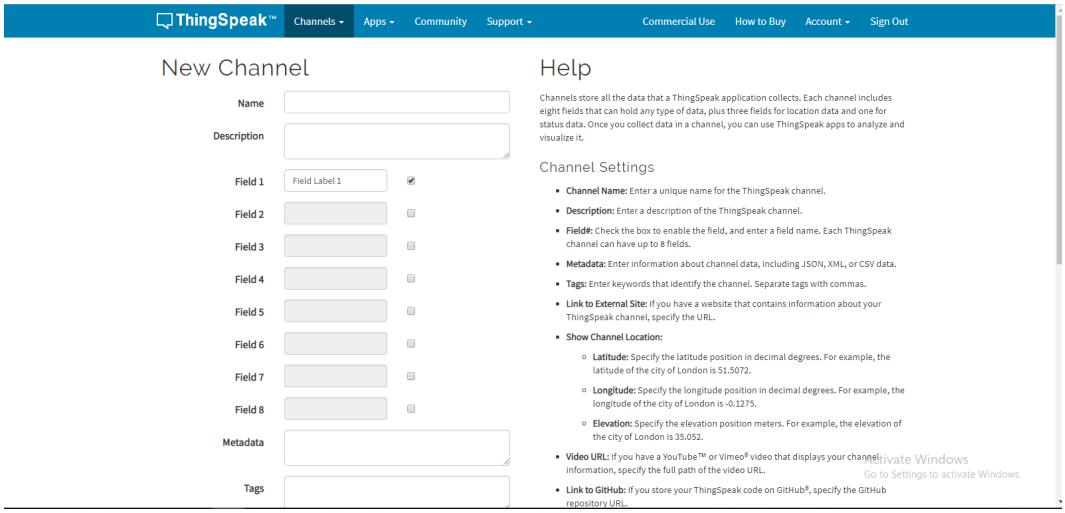


Fig. 11.1: Create New Channel

Channels

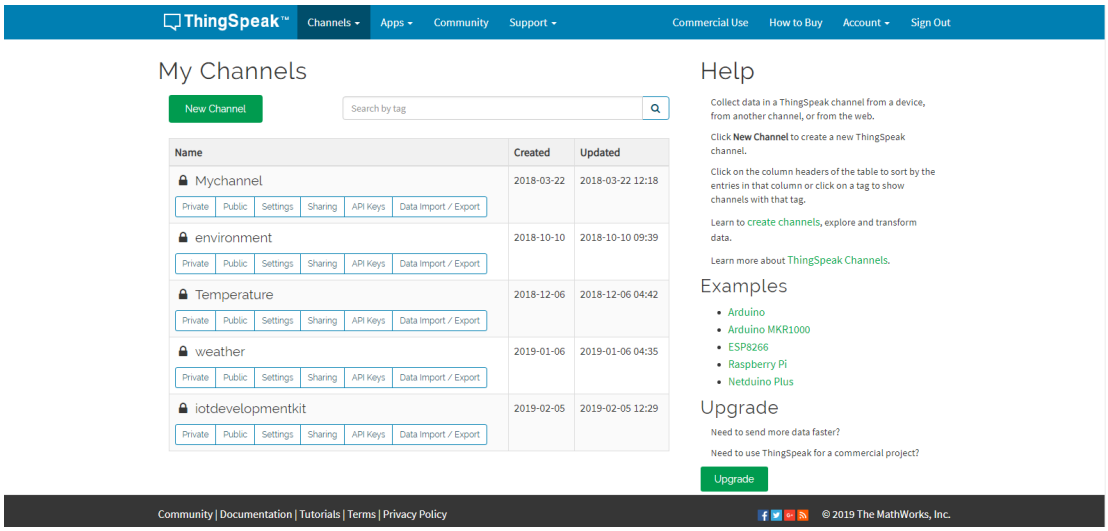
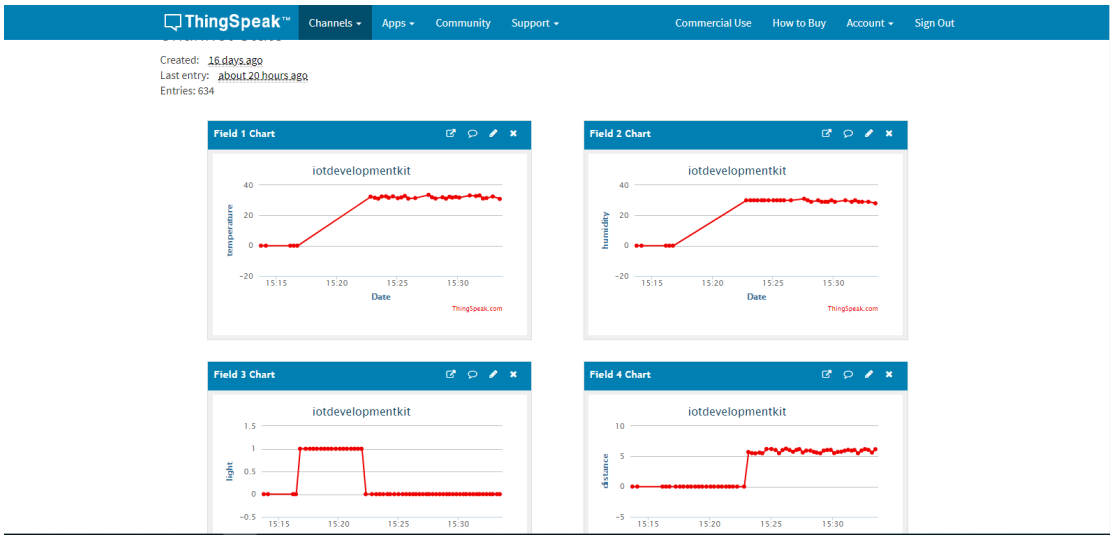


Fig. 11.2: All Channels

Fields Data



11.2 API Reference

ThingSpeak is an IoT platform that uses channels to store data sent from apps or devices. With the settings described in Channel Configurations, you create a channel, and then send and retrieve data to and from the channel. You can make your channels public to share data. Using the REST API calls such as GET, POST, PUT, and DELETE, you can create a channel and update its feed, update an existing channel, clear a channel feed, and delete a channel. You can also use the MQTT Publish method to update a channel feed and MQTT Subscribe to receive messages when a channel updates. Learn more about when to Choose between REST API and MQTT API while updating a channel.

MATLAB analysis and visualization apps enable you to explore and view your channel data. ThingSpeak enables you to interact with social media, web services, and devices.

REST API: Use REST API calls to Create and update ThingSpeak Channels and Charts.
MQTT API: Use the MQTT API to update ThingSpeak channels.

11.2.1 REST API:

Representational state transfer (REST) is an architectural style designed as a request-response model that communicates over HTTP. ThingSpeak, an IoT platform, uses the REST API calls GET, POST, PUT, and DELETE to create and delete channels, read and write channel data, and clear the data in a channel. A web browser or client sends a request to the server, which responds with data in the requested format. Web browsers use this interface to retrieve web pages or to send data to remote servers.

Sending Data to ThingSpeak Cloud

GET /update?api_key="+ <write-api> +"&field1="+<field1-value>

<write-api>: Write API Key of particular channel.

<field1-value>: sensor data.

Retrieving Data from ThingSpeak Cloud

GET /channels/<channel-id>/fields/<field-number>/last.json?api_key=<read-api-key>

Channel-id: ID of the particular Channel.

read-api-key: Read API key of Particular channel.

For more information about API, please refer the below link
https://thingspeak.com/channels/456119/api_keys

11.2.2 MQTT API:

The ThingSpeak IoT platform enables clients to update and receive updates from channel feeds via the ThingSpeak MQTT broker. MQTT is a publish/subscribe communication protocol that uses TCP/IP sockets or WebSockets. MQTT over WebSockets can be secured with SSL. A client device connects to the MQTT broker and can publish to a channel or subscribe to updates from that channel.

Publish:

Publish to Channel Feed	Publish message to update multiple channel field simultaneously.
Publish to Channel field Feed	Publish message to update single channel field

Subscribe:

Subscribe to Channel Feed	Subscribe to updates from a channel feed with MQTT
Subscribe to Channel Field Feed	Subscribe to channel updates from a particular field of a channel with MQTT

User can refer the below link for the complete information of ThingSpeak cloud document.
https://in.mathworks.com/help/thingspeak/

12 TROUBLESHOOTING:

Here are some guidelines to troubleshoot some typical problems. Problems in field, however, can vary, depending on field conditions.

1. **ESP8266 is not responding.**
   - Make sure that you have connected relimate cable from RM16 to RM7.
   - Make sure that you have set the baud rate to 9600. (Refer ESP8266 Demo program and flash the program and set the baud rate to 9600).
   - Make sure that your internet is working.
   - Make sure that you have given correct spelling of WiFi Username and Password.
2. **DHT11 is giving -999 or garbage values**

- Make sure that you have connected external power cable for Arduino.
- Make sure that proper libraries are installed.
- Make sure that you have given proper delay before reading the data from sensor. Minimum you have to provide 2 seconds delay.

3. **Ultrasonic sensor is producing garbage values**

- Make sure that the object is not less than 2 cm or not more than 400 cm.
- Make sure that you have given proper delay to read the sensor. Minimum 1 second delay required to read the distance.

4. **Gas Sensor is producing garbage values**

- Make sure that you have given preheating delay to the sensor to heat up.
- Adjust the potentiometer to increase the sensitivity of the sensor.

5. **RTC is not working Properly**

- Make sure that you have connected the Relimate cable from RM1 to RM10.
- Make sure that you have installed proper Libraries/ Executed Proper Commands.
- Still you find problem reset/reboot the device and flash the program again.

6. **MCP3008 ADC is giving Continuously 0 value.**
   - Make sure that you have installed proper libraries.
   - Make sure that you have cleanup GPIO pins in earlier program (In Raspberry Pi you have stop every program execution using ctrl+c keyword only. Otherwise GPIO pins are not cleaned up from memory which shows the effect on next program.)
   - Still if you have the problem reboot the device.
7. **Soil Moisture sensor is not working properly**

- Make sure that the conductor in soil moisture sensor is in good condition.
- If you are using the conductor for long time you need to replace the conductor.
- You need to understand that for wet soil it will give different values depending on air and water content in the soil. It changes the values for different soils and different water we have used.

REFERENCES

1. https://www.arduino.cc/en/main/software
2. https://www.raspberrypi.org/documentation/installation/noobs.md
3. https://www.raspberrypi.org/downloads/raspbian/
4. https://sourceforge.net/projects/win32diskimager/
5. https://thingspeak.com/
6. https://www.sdcard.org/downloads/formatter_4/
7. https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html