**IOT based Sensor Virtualisation and Analytics using Deep learning**

Nikita Patil, Divya Mokal, Satyamurti M, Gayatri Shinde, Prof. Ruchira Patole

***Abstract:* Internet of Things (IoT) and Network Softwarization are fast becoming core technologies of information systems and network management for the next generation Internet. The concept of IoT means more than just devices connecting to the global Internet. IoT is closely coupled with sensor technology, because 2in most of the cases sensors & actuators are part of a larger IoT network. These IoT sensors & actuators may produce large volumes of data. Hence, the need for installing new network access & core devices will increase. To manage the network devices efficiently, the network hardware resources need to be virtualized.Virtualization is the logical abstraction of the underlying hardware devices within a network, through software implementation. Here we use deep learning, a computer model which performs classification tasks directly from images, text, or sound. Virtual sensor is a pure software sensor which autonomously produces signals by combining and aggregating signals that it receives (synchronously or asynchronously) from physical or other virtual sensors. Virtual sensors serve to overcome a number of weaknesses of purely physical sensors.**

**1. INTRODUCTION**

Virtualization is the process or a technique which can create a virtual version of IT datacenter components such as Compute, storage and network etc. Hypervisor is one of the commonly used virtualization technology to create virtualized IT infrastructures. Generally, sensors are used in the architecture of IOT devices. Sensors are used for sensing things and devices etc. A device that provides a usable output in response to a specified measurement. The sensor attains a physical parameter and converts it into a signal suitable for processing the characteristics of any device or material to detect the presence of a particular physical quantity. The output of the sensor is a signal which is converted to a human-readable form like changes in characteristics, changes in resistance, capacitance, impedance etc.

**2. LITERATURE REVIEW**

Dominic Martin, Niklas Kuhl, Gerhard Satzger: "Virtual Sensors" Information systems increasingly link to the physical world. Technological advancements and declining unit costs of sensor technology combined with increased connectivity drive the spread and complexity of the Internet of Things (IoT) or so-called cyber-physical systems.

Iqbal Alam, Kashif Sharif, Member, IEEE, Fan Li, Member, IEEE"IoT Virtualization: A Survey of Software Definition & Function Virtualization Techniques for Internet of Things". Internet of Things (IoT) and Network Softwarization are fast becoming core technologies of information systems and network management for the next generation Internet.

Hillol Debnath, Narain Gehani, Xiaoning Ding, Reza Curtmola and Cristian Borcea "Sentio: Distributed Sensor Virtualization for Mobile Apps". This paper presents Sentio, a distributed middleware designed to provide mobile apps with seamless connectivity to remote sensors when the sensing code and the sensors are not physically on the same device, e.g., when the sensing code is offloaded to the cloud.

JeongGil Ko, Byung-Bog Lee, Kyesun Lee, Sang Gi Hong, Naesoo Kim, Jeongyeup Paek "Sensor Virtualization Module: Virtualizing IoT Devices on Mobile Smartphones for Effective Sensor Data Management" Low-power embedded sensor networking platforms willsoon be deployed for various application purposes in our everyday environments.

**3. METHODOLOGY**

The system overview is presented in this Section. Network virtualization is the mechanism of combining both software & hardware resources and network functionality into a logically configured single software-based administrative entity.

# SVM Architecture

*SVM* simplifies application development by providing an abstraction of the IoT devices that are present in the local field. *SVM* makes it look as if the external devices are on-board sensor components such as accelerometer, gyro, GPS, or camera. When applications interconnect with the *SVM* layer, a set of open APIs are provided to the applications for them to interact with the *SVM* layer and to easily access these resources. For example,launchSensorDiscovery() API searches for all the external physical sensors that can be connected to the smart device via any of the network access interfaces (e.g., Bluetooth, WiFi, and ZigBee) on the smart device and provides a list of those sensors.

Although this process allows easy access to external IoT devices via mobile platforms from various applications, the increase in the number of interacting applications can result in conflicts among resource requests. Since such application level algorithms are not enforced by any standardization body, we cannot assure that these conflicts will be properly processed at the end-device level. As a result,besides managing the connectivity, another major role of the SVM engine is to resolve such conflicts caused by multiple requests from different applications.

# 

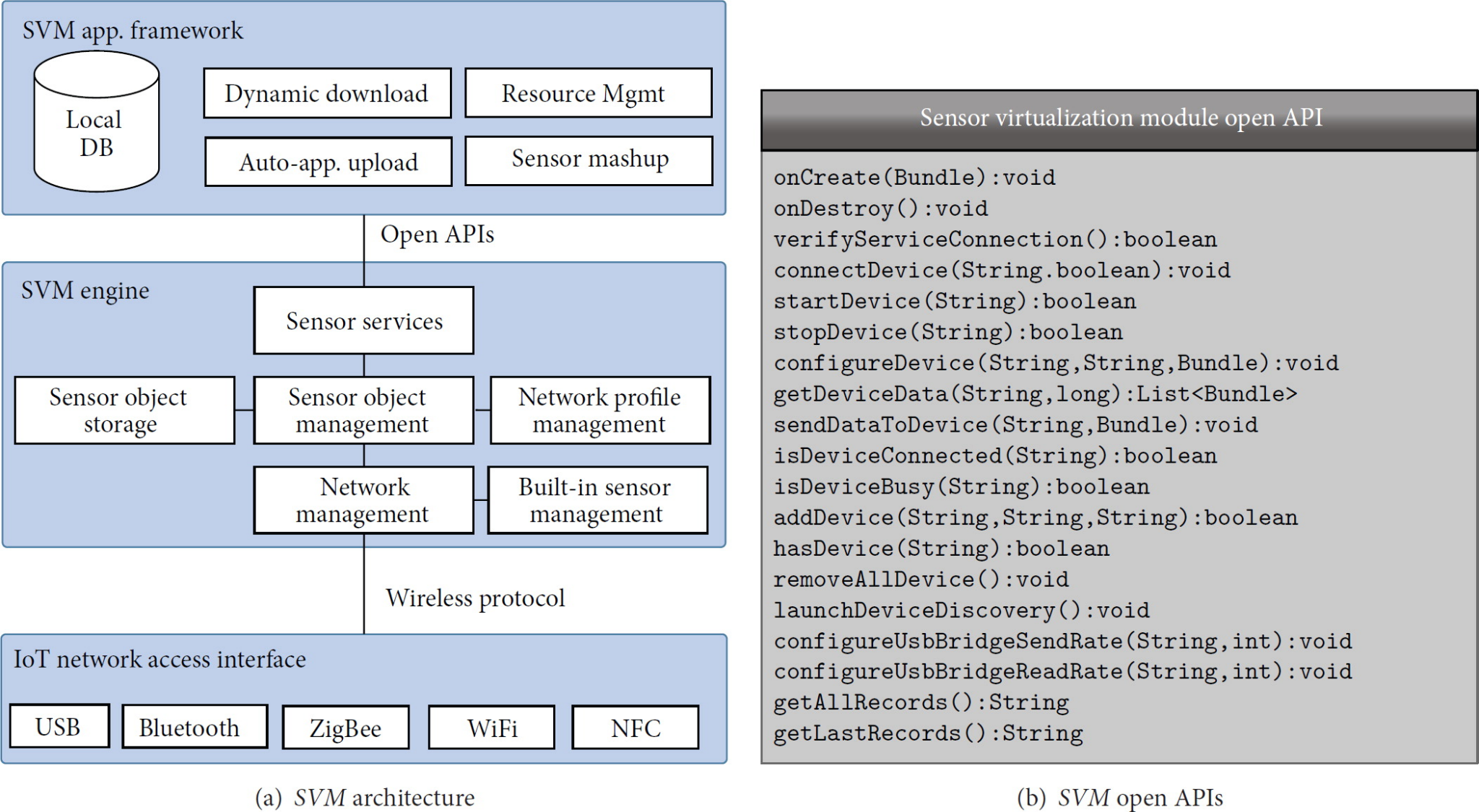
****

Fig .1.1: SVM architecture

# Proposed System Architecture

In this system we are using two raspberry pi 4 with five sensors of single type and connecting them in wireless mode . By this system architecture we will be able to perform virtualization wirelessly . So that we can check the system whenever and wherever we want .

Virtual sensors will provide indirect measurements of data by combining data from different heterogeneous physical sensors in order to provide services to the user . The sensor virtualization technique will allow the user to obtain preferred and precise information in a more efficient manner from a limited number of sensors. Furthermore, this helps in reducing the energy consumption and cost of the overall network.

**Train the model:** Virtualization is the creation of a virtual version rather than an actual version of something, such as an operating system (OS), a server, a storage device or network resources.

For example, here we create a system which uses one raspberry pi 4 with five smoke sensors which detect LPG gas connected to each other. So we connect the network with another raspberry pi 4 in wireless mode.By this, the outcome of this network will be shown as the same as that which we have checked by actually working with the sensors. So this will show that we have created a virtual version of a physical network which gives the same result as the physical version.

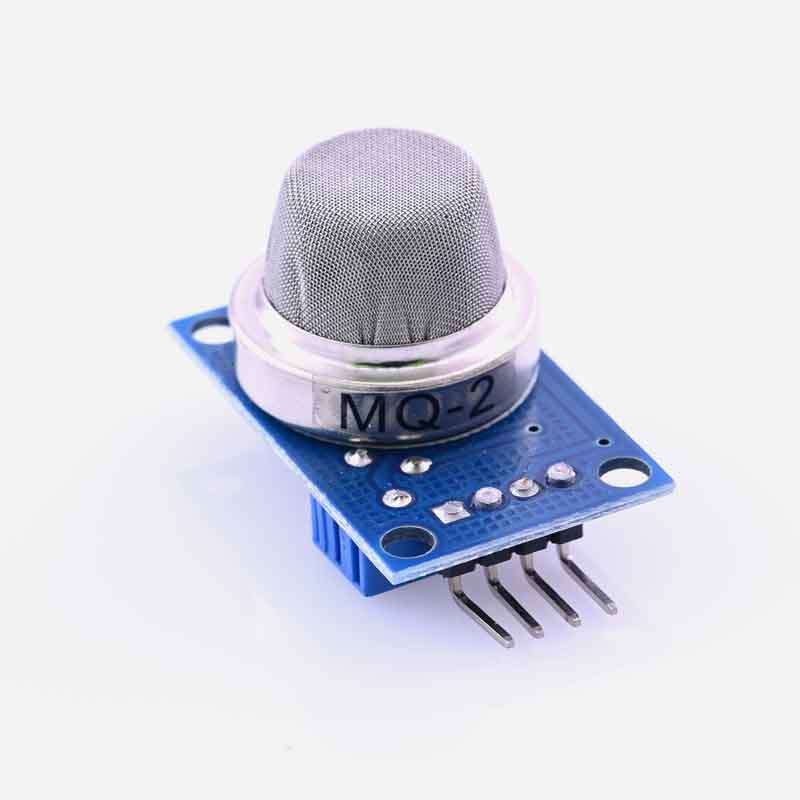


Fig -1.2:MQ2 Sensor

The following shows the diagram for network we had created:

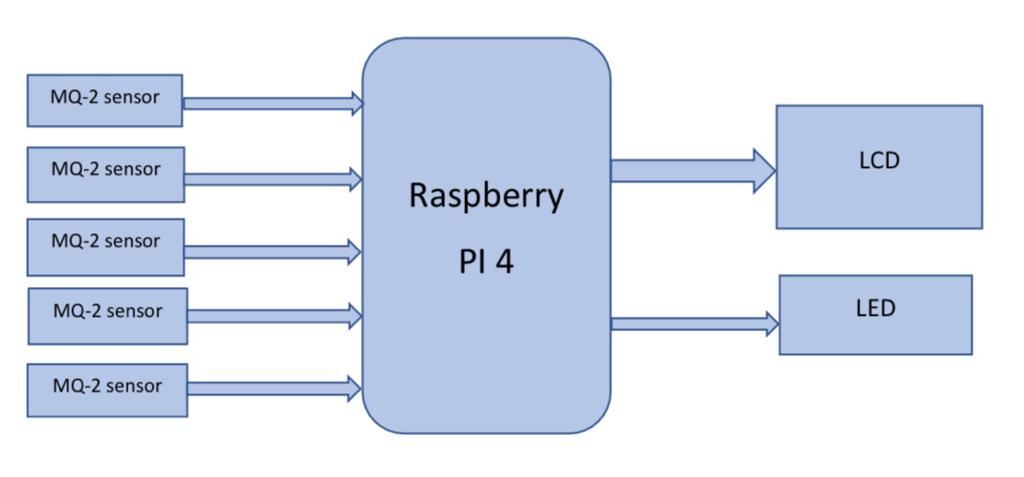


Fig -1.3 Basic Block Diagram

The above diagram shows :

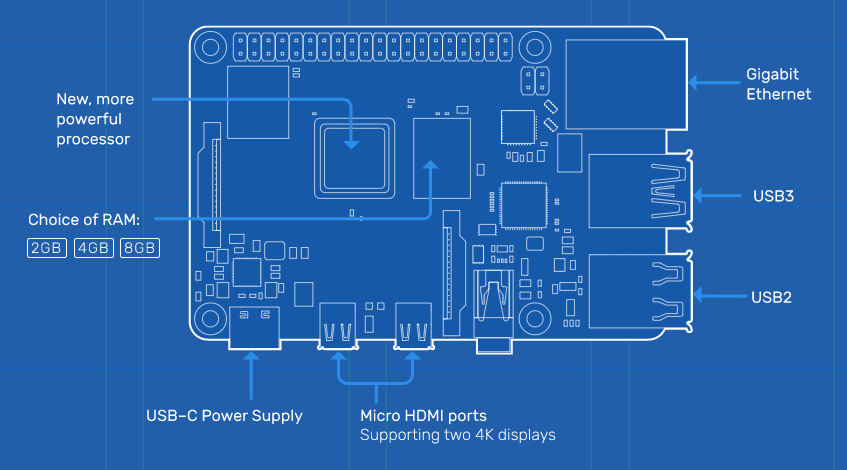
1] Five MQ-2 gas sensors are connected to raspberry pi 4 which can detect LPG gas . The MQ2 gas sensor operates on 5V DC and consumes approximately 800mW. It can detect LPG, Smoke, Alcohol, Propane, Hydrogen, Methane and Carbon Monoxide concentrations ranging from 200 to 10000 ppm.2] One LCD connected to raspberry pi 4 which will display the result. It is a 16x4 which means it will take upto 16 characters and 4 rows.

3] And one LED which glows when detected by sensor 5.

**Hardware and Software Specifications**

The experiment setup is carried out on a computer system which has the different hardware and software specifications as given in Table 2.1 and Table 2.2 respectively.

.

 Table 2.1:Hardware details (Raspberry Pi 4)

Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz.1GB, 2GB, 4GB or 8GB LPDDR4-3200 SDRAM (depending on model). 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE.Gigabit Ethernet. 2 USB 3.0 ports; 2 USB 2.0 ports. Raspberry Pi standard 40 pin GPIO header (fully backwards compatible with previous boards). 2 × micro-HDMI ports (up to 4kp60 supported). 2-lane MIPI DSI display port. 2-lane MIPI CSI camera port. 4-pole stereo audio and composite video port. H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode). OpenGL ES3.1, Vulkan 1.0. Micro-SD card slot for loading operating system.

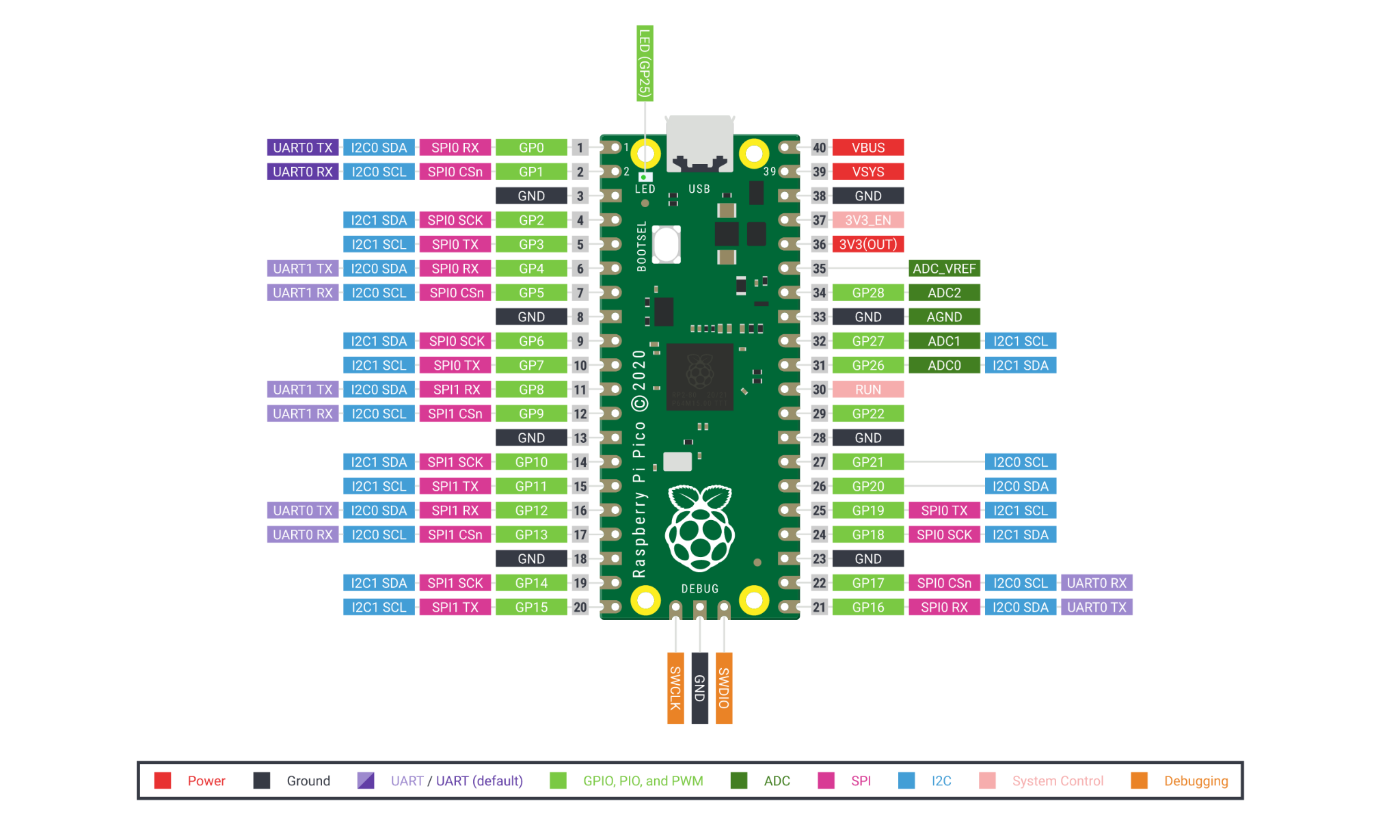


Table 2.1:Hardware details (Raspberrypi pico)

RP2040 microcontroller with 2MB Flash. Micro-USB B port for power and data (and for reprogramming the Flash). 40 pin 21×51 'DIP' style 1 mm thick PCB with 0.1" through-hole pins also with edge castellations. 3-pin ARM Serial Wire Debug (SWD) port. Simple yet highly flexible power supply architecture. Dual-core cortex M0+ at up to 133MHz. 264kB multi-bank high performance SRAM. External Quad-SPI Flash with eXecute In Place (XIP) and 16kB on-chip cache. High performance full-crossbar bus fabric.

**4. RESULTS**

Raspberry pico detection results are shown below which shows the distance measured by ultrasonic sensors and the distance is plotted on the graph:

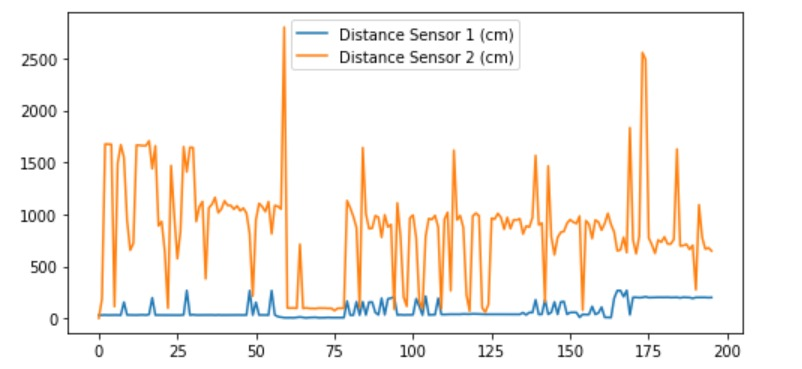


Fig. -1.4: Graph showing distance

**5. CONCLUSION**

IOT technology has come a long way since it was conceptualized two decades ago. The work presented in this project was directed towards pushing IOT technology to the next level. The principle of Operation of IOT based gas leakage and monitoring system was shown by operating the Raspberry pi 4 model attached to an embedded system with required input and output gas level with the help of gas sensors.This results in a more efficient operation because it is connected to a LCD display.Using a real time gas leakage monitoring and sensing the output levels of gas has been clearly observed.

**6. REFERENCES**

[1] Dominic Martin, Niklas Kuhl, Gerhard Satzger: "Virtual Sensors"

[2] Iqbal Alam, Kashif Sharif, Member, IEEE, Fan Li, Member, IEEE"IoT Virtualization: A Survey of Software Definition & Function Virtualization Techniques for Internet of Things"

[3] Hillol Debnath, Narain Gehani, Xiaoning Ding,

Reza Curtmola and Cristian Borcea "Sentio: Distributed Sensor Virtualization for Mobile Apps"

[4] JeongGil Ko, Byung-Bog Lee, Kyesun Lee, Sang Gi Hong, Naesoo Kim, Jeongyeup Paek "Sensor Virtualization Module: Virtualizing IoT Devices on Mobile Smartphones for Effective Sensor Data Management"

[5] Kumar Keshamoni and Sabbani Hemanth. "Smart Gas Level Monitoring, Booking & Gas Leakage Detector over IoT " International Advance Computing Conference IEEE, 2017.

[6] S. El Bouanani, M. A. E. Kiram, and O. Achbarou, “Introduction to the internet of things security: Standardization and research challenges,” in Proc. of Int. Conf. on Information Assurance and Security, December 2015, pp. 32–37

[7] Saleem, T. J., & Chishti, M. A. (2019). Deep Learning for Internet of Things Data Analytics. Procedia Computer Science, 163, 381–390. doi:10.1016/j.procs.2019.12.120