**Exploring the Potential of IoT with Sensor Networks**

**1**

### AN INTERNSHIP REPORT

***Submitted in partial fulfillment for the award of the degree of***

## BACHELOR OF TECHNOLOGY

**in**

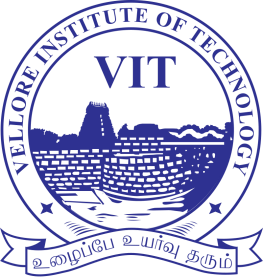
### ELECTRONICS AND COMMUNICATION ENGINEERING

***by***

## ASWATH S

### (Reg. No.: 21BEC2188)

**SCHOOL OF ELECTRONICS ENGINEERING (SENSE)**



VIT

## U N I V E R S I T Y

(Estd. u/s 3 of UGC Act, 1956)

## SEPTEMBER 2023

## ACKNOWLEDGEMENT

This report wouldn’t have been possible without the constant help and support I have received from various places. I take this opportunity to express our sincere thanks and deep gratitude to all those people who extended their whole-hearted cooperation and helped me to complete this project.

First, I would like to thank Mr. Prakash R the Educational Head of Enthu Technology and Mrs. Geetha for giving me the opportunity to do an internship with the organization.

I also would like to thank all the people that worked along with me with their patience and openness they created an enjoyable working environment. It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of these individuals.

I would like to thank my Head of the Department Dr. Noor Mohammed (SENSE) at VIT Vellore for giving me this opportunity for carrying out this internship.

**1**

# ABSTRACT

The Internet of Things (IoT) has emerged as a transformative force, integrating the physical world with digital networks through the deployment of interconnected sensors. These sensors, ranging from simple temperature gauges to complex environmental monitors, form the backbone of IoT systems, enabling real-time data collection, analysis, and action. This abstract explores the significance of IoT with sensor networks, highlighting its impact across various domains.

In the realm of smart cities, IoT sensors facilitate efficient resource management, traffic optimization, and environmental monitoring. Through interconnected networks, these sensors enable authorities to make data-driven decisions, enhancing urban livability and sustainability.

In industrial settings, IoT sensor deployments revolutionize operations by providing predictive maintenance insights, optimizing supply chains, and ensuring workplace safety. Real-time monitoring of machinery health, inventory levels, and environmental conditions not only improves efficiency but also minimizes downtime and reduces costs.

In healthcare, IoT sensors enable remote patient monitoring, personalized treatment plans, and early disease detection. Wearable devices equipped with biometric sensors collect vital health data, empowering individuals to take proactive measures for their well-being while allowing healthcare providers to deliver targeted interventions.

Moreover, in agriculture, IoT sensors revolutionize farming practices by monitoring soil moisture levels, crop health, and weather conditions. This data-driven approach enhances agricultural productivity, conserves resources, and mitigates the impact of climate change on food security.

Despite its transformative potential, the widespread adoption of IoT with sensor networks faces challenges related to data privacy, security, and interoperability. Addressing these concerns requires collaborative efforts from stakeholders across industries to establish robust standards and protocols.

In conclusion, the convergence of IoT with sensor networks heralds a new era of connectivity and intelligence, revolutionizing how we interact with the world around us. By harnessing the power of real-time data analytics and machine learning, IoT systems equipped with sensors have the potential to drive innovation, improve quality of life, and create sustainable ecosystems.

### ESP32

### ESP32-WROOM-32E and ESP32-WROOM-32UE are two powerful, generic Wi-Fi + Bluetooth + Bluetooth LE MCU modules that target a wide variety of applications, ranging from low-power sensor networks to the most demanding tasks, such as voice encoding, music streaming and MP3 decoding. ESP32-WROOM-32E comes with a PCB antenna, and ESP32-WROOM-32UE with a connector for an external antenna. The information in this datasheet is applicable to both modules. N-channel MOSFET

At the core of the module is the ESP32-D0WD-V3 chip or ESP32-D0WDR2-V3 chip\*. The chip embedded is designed to be scalable and adaptive. There are two CPU cores that can be individually controlled, and the CPU clock frequency is adjustable from 80 MHz to 240 MHz. The chip also has a low-power coprocessor that can be used instead of the CPU to save power while performing tasks that do not require much computing power, such as monitoring of peripherals. ESP32 integrates a rich set of peripherals, ranging from capacitive touch sensors, SD card interface, Ethernet, high-speed SPI, UART, I2S, and I2C.

### STM-32

The MCU integrates multiple series of software- and pin-compatible Arm®-based 32-bit cores that share a common set of Renesas peripherals to facilitate design scalability and efficient platform-based product development. The MCU in this series incorporates a high-performance Arm Cortex-M4 core running up to 120 MHz with the following features:384-KB SRAM , Capacitive Touch Sensing Unit (CTSU), Ethernet MAC Controller (ETHERC), USBFS, SD/MMC Host Interface , Quad Serial Peripheral Interface (QSPI),Security and safety features, Analog peripherals,12-bitConverter(ADC12)|(DAC12).

The Memory Mirror Function (MMF) can be configured to mirror the target application image load address in code flash memory to the application image link address in the 23-bit unused memory space (memory mirror space addresses). Your application code is developed and linked to run from this MMF destination address. The application code does not need to know the load location where it is stored in code flash memory. On-chip high-speed SRAM with either parity-bit or Error Correction Code (ECC).

### STM32WLE5xx (LORA)

The STM32WLE5/E4xx long-range wireless and ultra-low-power devices embed a powerful and ultra-low-power LPWAN-compliant radio solution, enabling the following modulations: LoRa, (G)FSK, (G)MSK, and BPSK. The LoRa® modulation is available in STM32WLx5xx only. These devices are designed to be extremely low-power and are based on the high-performance Arm Cortex®-M4 32-bit RISC core operating at a frequency of up to 48 MHz. This core implements a full set of DSP instructions and an independent memory protection unit (MPU) that enhances the application security.

The devices embed high-speed memories (flash memory up to 256 Kbytes, SRAM up to 64 Kbytes), and an extensive range of enhanced I/Os and peripherals. The devices also embed several protection mechanisms for embedded flash memory and SRAM: readout protection, write protection and proprietary code readout protection. These devices offer a 12-bit ADC, a 12-bit DAC low-power sample-and-hold, two ultra-low-power comparators associated with a high-accuracy reference voltage generator. The devices embed a low-power RTC with a 32-bit sub-second wakeup counter, one 16-bit single-channel timer, two 16-bit four-channel timers (supporting motor control), one 32-bit four-channel timer and three

**INTERNSHIP CERTIFICATE:**



**1**

# TABLE OF CONTENTS

**CHAPTER TITLE PAGE NO**

1. INTRODUCTION 6
   1. [Objective](#_TOC_250005) **6**
   2. Organization Information **6**
2. Overview of the Industry 7
3. INFERENCE 1
   1. Sensors interfacing ESP32 **1**
   2. Sensors interfacing STM32 **1**
   3. Sensor Interfacing LORAWAN **4**
4. **Detailed Description 6**
5. **Software Learnt 7**
6. **Summary 8**
7. Reference 10

# INTRODUCTION

### Objective

During my internship at Enthu Technology Solutions India Pvt Ltd, my objective was to study various aspects related to IOT such as Sensor interfacing with ESP32,Sensor interfacing with STM32 [and Sensor interfacing with LoRaWAN and](https://www.ti.com/lit/ml/slua618a/slua618a.pdf) LoRaWAN technology, PCB making and Other different product of the company.

### Organisation Information

Enthu Technology Solutions India Pvt. Ltd. is an ISO-Certified Organization founded by a group of young Entrepreneurs in 2007 and incorporated as a Private Limited Company in 2015. We offer best-in-class end-to-end Product Design, Development, and Manufacturing services for Embedded and IoT products. As a product development R&D company, we have in-house Hardware Design and Prototyping, Software Development, EMS Assembly, Enclosure Design & Manufacturing, IoT Cloud based Web & Mobile App Development, product testing, and Deployment. As a Solution Provider, we offer complete digital transformation for all your problem statements under one roof.

We have a tremendous track record of bringing up new start-up entrepreneurs by establishing their product manufacturing facility with PCBMATE & SMTMATE Products. We provide best-in-class OEM & ODM Services to our clients based on their demands.

On behalf of the Industry Institute Collaboration, we are operating Enthu Academic Solutions to fill the gap between the Industry and Institute by offering laboratory establishment, real-time exposure to research activities, and other Industrial training and developments that would benefit the institutes and the industry.

VIT has signed a MoU with Enthu technology. VIT is client of Enthu technologies. Enthu Academic Solutions (EAS) is an academic division of Enthu Technology Solutions has taken part in implementation of project based learning. Enthu tech established a Modern Industrial grade Laboratories with smart campus Infrastructure at PRP 108 in VIT.

**OVERVIEW OF THE INDUSTRY:**

The IoT (Internet of Things) industry is a rapidly growing sector that encompasses a wide range of technologies, applications, and market segments. Here's an overview:

* Definition and Scope: IoT refers to the network of interconnected devices, sensors, and software that enables the exchange of data and communication over the internet. These devices can range from everyday objects like household appliances and wearable devices to industrial machinery and infrastructure components.
* Market Size and Growth: The IoT industry has experienced significant growth in recent years and is projected to continue expanding rapidly. Market research forecasts indicate that the global IoT market size will reach several trillion dollars by the end of the decade, driven by increasing adoption across industries.
* Key Technologies: IoT solutions rely on various technologies such as sensors, actuators, connectivity protocols (e.g., Wi-Fi, Bluetooth, Zigbee, LoRaWAN), cloud computing, edge computing, artificial intelligence (AI), and data analytics. These technologies enable devices to collect, process, and analyze data, as well as communicate with each other and with centralized systems.
* Vertical Markets: The IoT industry spans multiple vertical markets, including:
* Smart Home: IoT devices for home automation, security, energy management, and entertainment.
* Smart Cities: IoT deployments for urban infrastructure, transportation, public safety, environmental monitoring, and waste management.
* Industrial IoT (IIoT): IoT solutions in manufacturing, logistics, supply chain management, predictive maintenance, and asset tracking.
* Healthcare: IoT applications for remote patient monitoring, telemedicine, medical device connectivity, and personalized healthcare.
* Agriculture: IoT technologies for precision farming, crop monitoring, livestock management, and agricultural automation.
* Retail: IoT-enabled inventory management, supply chain optimization, customer engagement, and personalized shopping experiences.
* Challenges and Opportunities: Despite its growth potential, the IoT industry faces several challenges, including data privacy and security concerns, interoperability issues, regulatory compliance, and the complexity of integrating diverse technologies. However, these challenges also present opportunities for innovation and market differentiation, driving the development of new solutions and business models.

**INFERENCE**

**Equipment’s used and observed:**

### ESP32

**I had done various experiments using the ESP 32 using different types of sensors, such as temperature sensors, smoke sensors, LDR sensors, gas sensors, and ultrasonic sensors, Soil Moisture Sensor, Rain Sensor, and Vibrating Sensor using Arduino IDE software**

**Here are some explanation about all sensor used with ESP32 we use such as**

* **Temperature sensor: Temperature sensors used with ESP32 Arduino setups provide accurate temperature readings for monitoring and control applications. By interfacing the sensor with the ESP32 microcontroller, users can easily integrate temperature sensing capabilities into their projects. These sensors are commonly employed in IoT devices, environmental monitoring systems, and smart home automation projects.**
* **Soil Moisture:** Soil moisture sensors paired with ESP32 and Arduino IDE enable real-time monitoring of soil moisture levels for optimized irrigation systems. By interfacing with the ESP32, these sensors provide data for agricultural applications, allowing users to automate watering schedules based on soil moisture content. This integration is crucial for enhancing crop yields, conserving water, and promoting sustainable farming practices.
* **Rain drop sensor: Raindrop sensors integrated with ESP32 and Arduino IDE facilitate the detection of rainfall for weather monitoring and smart irrigation systems. Interfacing the sensor with ESP32 enables real-time data collection, allowing users to automate actions based on rainfall intensity. This setup enhances efficiency in managing outdoor activities, such as adjusting sprinkler systems or alerting users to potential flooding.**
* **LDR sensor: LDR (Light Dependent Resistor) sensors measure ambient light levels and change their resistance accordingly, making them essential components in light-sensitive applications. They are commonly used in streetlights, cameras, and automatic brightness adjustment systems in electronic devices. LDR sensors offer a cost-effective solution for detecting light variations, enabling responsive and energy-efficient operation in various environments.**
* **GAS sensor: The MQ-135 gas sensor, detects a variety of gases such as ammonia, nitrogen oxides, and benzene, crucial for air quality monitoring in indoor and outdoor environments. By interfacing with ESP32, it enables real-time gas detection and analysis, supporting applications like smart home air purifiers and pollution monitoring systems. Integrating the MQ-135 sensor with ESP32 facilitates the development of IoT solutions aimed at promoting healthier and safer living spaces.**
* **Ultra sonic sensor: The ultrasonic sensor, enables accurate distance measurement using sound waves, vital for robotics, obstacle avoidance, and smart parking systems. By interfacing with ESP32, it provides real-time data on object proximity, allowing for responsive and precise control in IoT applications. Integrating the ultrasonic sensor with ESP32 enhances the versatility and functionality of projects, contributing to advancements in automation and sensing technology.**
* **Vibration sensor: Vibration sensors are crucial for industrial machinery monitoring, structure health monitoring, and security alarm systems since they detect mechanical vibrations and impacts when connected with ESP32. It can detect vibrations in real time and send out notifications or take action in response to changes in the surrounding environment by integrating with ESP32. The capabilities of IoT devices are boosted by integrating the vibration sensor with ESP32, which helps with improved maintenance and safety in a variety of applications.**

**Temperature sensor using ESP 32:**

**Temperature Monitoring: You can keep an eye on the temperature in a certain area or setting by connecting a temperature sensor to an ESP32 microcontroller through the Arduino IDE. Furthermore, it is possible to compute real-time data. The ESP32 continuously receives temperature data from the sensor, processes it in real-time, and outputs the most recent temperature conditions.**

**Here the brief explanation of Sensing temperature using temperature Sensor**

* **Hardware Setup:** The required parts are as follows: an ESP32 development board (such the Node MCU ESP-32S), jumper wires, a temperature sensor (like the DHT11 or DHT22), and a breadboard (if needed). Join the ESP32 and the temperature sensor. Connect the DHT11 or DHT22, for instance, to the ESP32's VCC pin, GND to GND, and data pin to a GPIO pin,
* I**nstalling**: Installing the Arduino IDE launching it and installing support for ESP32 band. Here is how to set up things for Arduino IDE:   
  Navigate to "File" > "Preferences" in the Arduino IDE. Add the following URL to the "Additional Board Manager URLs" field: https://dl.espressif.com/dl/package\_esp32\_index.json   
  To exit the Preferences window, click "OK". Navigate to "Board" > "Tools" > "Boards Manager..."
* **Select Board and Port**: Go to "Tools" > "Board" and select your ESP32 board from the list (e.g., "ESP32 Dev Module"). Choose the appropriate Port under "Tools" > "Port".
* **Install DHT Sensor Library:** In the Arduino IDE, go to "Sketch" > "Include Library" > "Manage Libraries...". Search for "DHT" and install the "DHT sensor library" by Adafruit.
* **Upload and Monitor**: After doing the above steps upload code in a\Arduino ide and I have connected the ESP 32 to my computer via USB. And clicked the "Upload" button in the Arduino IDE top load the code to the ESP32. After uploading, click "Tools" > "Serial Monitor" to open the serial monitor and view the temperature readings.

# Inferencing with protocol based sensor:

Protocol-based sensors such as I2C communication protocol (only two allowed)., OLED with ESP32. Temperature Sensor – SHT31(inbuilt in esp32) Light Sensor (BH1750) – I2C facilitating seamless integration with microcontrollers for data exchange and control. These sensors offer versatility and interoperability, allowing for standardized communication across various embedded systems and applications. Here some of protocol based sensor we used and explanation about the sensor

* **I2C communication protocol (only two allowed):** I2C (Inter-Integrated Circuit) protocol facilitates bidirectional serial communication between multiple devices over two-wire connections, comprising a data line (SDA) and a clock line (SCL). It allows for efficient data transfer between sensors, actuators, and microcontrollers, enabling robust and scalable communication in embedded systems.
* **Interfacing OLED with ESP32:** Interfacing an OLED display with ESP32 involves connecting the display to the ESP32's GPIO pins and utilizing libraries like Adafruit SSD1306 or u8g2 for easy integration. By sending commands and data over SPI or I2C communication protocols, the ESP32 can display text, graphics, and images on the OLED screen, enabling versatile user interfaces in IoT projects and wearable devices.
* **Temperature Sensor – SHT31:** The SHT31 temperature sensor is a senor which is in build in the esp32 in the company which we have done the internship and this esp32 with Temperature sensor (SHT31) is their own product, The SHT31 provides precise environmental data for various applications, including weather stations, HVAC systems, and indoor climate monitoring. With its low power consumption and compact design, the SHT31 enhances the ESP32's capabilities for IoT deployments requiring accurate environmental sensing.
* **Light Sensor (BH1750)**: The BH1750 light sensor communicates over the I2C protocol, simplifying integration into various microcontroller-based projects. With its high accuracy and wide dynamic range, it provides precise measurements of ambient light intensity for applications like automatic brightness adjustment and energy-efficient lighting systems. Interface enables seamless connectivity with microcontrollers like Arduino and ESP32, facilitating quick and efficient implementation in IoT and sensor-based projects. He ESP32 is used.
* **Accelerometer:** Accelerometers measure acceleration forces in multiple directions, crucial for detecting motion, tilt, and vibration in devices like smartphones, wearables, and vehicle navigation systems, enhancing user experience and enabling safety features.
* **Gyroscope:** Gyroscopes measure angular velocity, providing information about rotation and orientation changes, essential for stabilizing devices like drones, gaming controllers, and virtual reality headsets, enhancing precision in motion control and navigation applications.

### STM-32

We have done some of the basic programs using STM 32 and some of experiments using STM 32 with some sensor such as IR sensor, PIR sensor, LED sensor etc. Here some explanation about the sensor used and what we have done using that.

**Printing the Word “HELLO” using STM 32 and turning one the LED:**

To print "Hello World" using an STM32 microcontroller, you'd typically use a serial communication interface like UART or USB to transmit the message to a connected terminal or display.

For turning on and off an LED, you'd typically configure a GPIO pin as an output, set it high to turn the LED on, and low to turn it off, using the microcontroller's GPIO peripheral and corresponding registers.

By compiling and flashing this program onto an STM32 microcontroller, you can observe "Hello World" being printed to the terminal while controlling an LED's state, providing a starting point for further exploration and development.

**Toggling of led with STM 32 interfacing multiple sensors with STM 32 Tilt Sensor, PIR Sensor:**

**What is TILT SENSOR**:

One kind of sensor that may identify tilt or orientation changes is a tilt sensor. It can detect whether an object is level or inclined by measuring the angle of tilt with respect to a reference plane.   
There are several different types of tilt sensors, such as MEMS (Micro-Electro-Mechanical Systems) accelerometers, ball switches, and mercury-based switches. They are frequently utilized in applications including electronic game controllers, industrial equipment monitoring, and vehicle stability control.

**What is PIR SENSOR**:

PIR (Passive Infrared) sensors detect changes in infrared radiation emitted by objects within their detection range, making them ideal for motion detection applications.

These sensors are commonly used in security systems, automatic lighting, and smart home devices to detect human or animal movement.

PIR sensors are passive devices, meaning they do not emit any energy themselves but instead detect changes in the infrared radiation emitted by objects in their surroundings

Now we come to the point of Toggling of LED With the two sensor, both the sensor performed using same procedure and here are some points about using these sensor in toggling of LED

* Interfacing multiple sensors like a Tilt Sensor and PIR Sensor with an STM32 microcontroller involves connecting each sensor's output to separate GPIO pins.
* By configuring the GPIO pins as inputs, the STM32 can read the digital signals from the sensors, indicating their respective states, such as tilt or motion detection.
* Using conditional logic in the microcontroller's firmware, you can toggle an LED connected to another GPIO pin based on the sensor readings.
* For example, when the Tilt Sensor detects a tilt or the PIR Sensor detects motion, the STM32 can toggle the LED on or off accordingly.
* This setup enables the STM32 to respond dynamically to changes detected by multiple sensors, providing flexible control over the LED based on environmental conditions.

# Program on Interrupts using STM 32:

Some of key points about the program on Interrupts using STM32:

1. **Interrupt Configuration**: Developers configure interrupt handlers to respond to specific events, such as external signals from sensors or internal events like timer overflows, by setting up the interrupt vector table and enabling interrupt sources in the microcontroller's registers.

2. **Interrupt Service Routines (ISRs):** Each interrupt has an associated ISR, which is a function that executes in response to the interrupt. ISRs handle the interrupt's associated event, such as reading sensor data or updating system state, and are designed to execute quickly to minimize disruption to the main program flow.

3. I**nterrupt Priorities:** STM32 microcontrollers support interrupt priorities, allowing developers to assign different priority levels to different interrupts. This ensures that critical interrupts receive prompt attention, while lower-priority interrupts may be temporarily delayed.

4. **Interrupt Nesting:** The STM32's Nested Vector Interrupt Controller (NVIC) supports interrupt nesting, enabling ISRs to preempt lower-priority interrupt handlers if necessary. This feature ensures that high-priority interrupts can interrupt lower-priority interrupts without delay.

5. **Interrupt-driven Programming:** By utilizing interrupts, developers can implement event-driven software designs that efficiently respond to external events. Instead of continuously polling sensors or waiting for specific conditions in a loop, the microcontroller can remain in a low-power state until an interrupt occurs, conserving energy and processing resources.

**Interfaced I2C communication protocol based sensor BH1750 (LDR) Sensor with STM 32:**

1. **I2C Communication Protocol:** I2C (Inter-Integrated Circuit) is a serial communication protocol commonly used for connecting peripherals to microcontrollers. It uses two wires (SDA and SCL) for communication and supports multiple devices on the same bus.

2**. BH1750 Sensor:** The BH1750 is a digital light sensor that communicates over the I2C protocol. It measures ambient light intensity and provides digital readings that can be read by the STM32 microcontroller via the I2C bus.

3. **Interfacing with STM32:** To interface the BH1750 sensor with an STM32 microcontroller, you would connect the sensor's SDA and SCL pins to the corresponding I2C pins on the microcontroller. Then, you would use the STM32's I2C peripheral and appropriate libraries to communicate with the sensor and read its data.

4. **Reading Sensor Data:** With the BH1750 sensor connected and the I2C communication set up, you can read light intensity data from the sensor using the STM32 microcontroller. This data can then be processed and used in various applications, such as automatic lighting control or environmental monitoring.

5. **Software Development:** When interfacing the BH1750 sensor with the STM32 microcontroller, you would typically write firmware code in a programming language such as C or C++ using an Integrated Development Environment (IDE) like STM32CubeIDE. You would use I2C library functions provided by the STM32 HAL (Hardware Abstraction Layer) to communicate with the sensor and read its data.

By following these steps, you can successfully interface an I2C communication protocol-based sensor like the BH1750 with an STM32 microcontroller, enabling accurate light intensity measurements in your embedded systems projects.

**Introduction to communication Protocols like I2C, SPI. SPI Serial Peripheral Interface I2C Inter Integrated Circuit:**

Here some of the basic points and what is I2c and SPI and difference between them application of this protocol are given

1. **I2C (Inter-Integrated Circuit):**

* I2C is a synchronous serial communication protocol developed by Philips (now NXP)
* It uses two bidirectional lines: SDA (data line) and SCL (clock line), allowing multiple devices to be connected to the same bus.
* I2C supports multi-master communication, meaning multiple devices can initiate data transfer on the bus.

2. **SPI (Serial Peripheral Interface):**

* SPI is a synchronous serial communication protocol commonly used for communication between microcontrollers and peripheral devices.
* -It uses four signals: MOSI (Master Out Slave In), MISO (Master In Slave Out), SCK (Serial Clock), and SS (Slave Select).

3. **Differences:**

* I2C uses a shared bus architecture, while SPI typically uses point-to-point communication between a master and one or more slave devices.
* I2C operates at lower speeds compared to SPI, but it requires fewer pins and supports more devices on the same bus.
* SPI offers higher data rates and more flexibility in terms of communication modes and device configurations.

4. **Applications:**

* I2C is commonly used for connecting sensors, EEPROMs, and other low-speed peripheral devices in embedded systems
* SPI is often used for high-speed communication with devices such as ADCs, DACs, flash memory, and display controllers in embedded systems and IoT devices.
* Understanding these communication protocols is essential for designing and implementing efficient and reliable communication between microcontrollers and peripheral devices in embedded systems and IoT applications.

## Interfaced I2C communication protocol based sensor BH1750 (LDR) Sensor with STM 32:

Interfacing I2C communication protocol-based sensors like the BH1750 (light sensor) and LDR (Light Dependent Resistor) with an STM32 microcontroller allows for accurate light sensing and control in embedded systems. This integration enables the STM32 to read data from the BH1750 sensor, providing precise light intensity measurements, while also capturing analog readings from the LDR sensor for comparison or redundancy. By leveraging the STM32's capabilities to interface with multiple sensors via I2C and ADC, developers can create sophisticated applications such as automatic lighting systems, environmental monitoring, or smart energy management, enhancing efficiency and functionality in various IoT and embedded projects.

Procedure of detecting up this BH1750 senor includes Certainly! Here's a three-point procedure for interfacing I2C communication protocol-based sensors like the BH1750 (light sensor) and LDR (Light Dependent Resistor) with an STM32 microcontroller:

1. . Hardware Setup.
2. Software Configuration.
3. Data Acquisition and Processing.
4. Process the data obtained from both sensors as per the application requirements. This may involve calculating light intensity values from the BH1750 sensor and interpreting LDR readings.

By following this procedure, you can successfully interface I2C communication protocol-based sensors like the BH1750 and LDR with an STM32 microcontroller, enabling accurate light sensing and control in your embedded systems projects.

### Sensor interfacing LoRaWAN

* LoRaWAN (Long Range Wide Area Network) is a wireless communication protocol designed for long-range communication with low-power consumption, making it ideal for IoT (Internet of Things) and M2M (Machine to Machine) applications. It operates on unlicensed radio frequencies and allows for secure bi-directional communication between IoT devices and a centralized network server.
* Key features of LoRaWAN include:

1. Long Range: LoRaWAN can transmit data over several kilometers in urban environments and up to tens of kilometers in rural areas, enabling communication over vast distances.

2. Low Power: LoRaWAN devices can operate on battery power for years, thanks to their low power consumption. This makes them suitable for remote and energy-efficient applications.

3. Scalability: LoRaWAN networks can support thousands to millions of devices, making them highly scalable for large-scale IoT deployments.

4. Security: LoRaWAN incorporates various security mechanisms, including AES encryption, to ensure the confidentiality and integrity of data transmitted over the network.

5. Bi-directional Communication: LoRaWAN supports bi-directional communication, allowing IoT devices to both send data to the network server and receive commands or configurations from the server.

Overall, LoRaWAN provides a cost-effective and efficient solution for deploying IoT networks in diverse applications such as smart cities, industrial monitoring, agriculture, asset tracking, and environmental sensing.

**Sensor Interfacing**:  
Interfacing sensors with LoRaWAN using STM32 microcontrollers facilitates the creation of robust and scalable IoT solutions. By leveraging LoRaWAN's long-range, low-power communication capabilities and STM32's versatility and processing power, developers can deploy sensor networks for various applications with ease. This integration enables efficient transmission of sensor data over long distances, making it ideal for remote monitoring, environmental sensing, asset tracking, and smart city initiatives. Additionally, the combination of LoRaWAN and STM32 provides cost-effective and energy-efficient solutions suitable for deployment in diverse environments and industries.

### Introduction to SPI communication Protocol Interfacing RFID using SPI protocol with STM 32 WL development Board:

We have learned some basic information about RFID SPI and protocol interfacing RFID using SPI protocol with the STM 32 development board, some basic things about the STM32 Wl development kit with LORA WAN, about the LORA Wan board pins, and the basic program of relay turn on and turn off that is built into that kit. That LORAwan developed kit is a product of Enthu Technologies; they have created their own developed board with some LORAWan technologies.

And I had done some basic programming using This developed LORAWAN kit with STM32, such as LED turn-on and off with the STM WL development board Oled and LED is an I2C obeyed protocol device.

**What is SPI and RFID**:

***SPI (Serial Peripheral Interface)*** is a synchronous serial communication protocol commonly used for communication between microcontrollers and peripheral devices. It involves a master-slave architecture where one device acts as the master, controlling the communication, while one or more devices act as slaves.

**RFID (Radio-Frequency Identification**) is a technology that uses electromagnetic fields to automatically identify and track tags attached to objects. These tags contain electronically stored information that can be read wirelessly by RFID readers.

Interfacing RFID with an STM32 WL development board using SPI protocol enables the STM32 microcontroller to communicate with RFID readers and read RFID tags wirelessly.:

By interfacing RFID with an STM32 WL development board using SPI protocol, developers can create versatile and efficient RFID-based applications for a wide range of industries, enabling seamless identification and tracking of objects in real-time.

**Detailed Description:**

|  |  |
| --- | --- |
| **Date** | **Activity** |
| 14-Aug-2023 | Brief Introduction on :  1.Introduction to IoT  2.Explained about company features, achievements and their services.  3.PCBMATE 4.WDM5.ESP 32 with LORAWAN attached |
| 15-Aug-2023 | Independence Day - Government Holiday |
| 16-Aug-2023 | Interfacing Sensors with ESP 32 1. Smoke Sensor 2. Temperature sensor – LM 35 |
| 17-Aug-2023 | Interfacing Sensors with ESP 32  1. LDR Sensor.  2. Gas Sensor - MQ135.  3.Ultrasonic Sensor. |
| 18-Aug-2023 | Interfacing Sensors with ESP 32  1. Soil Moisture Sensor  2. Rain Sensor  3. Vibration Sensor |
| 19-Aug-2023 | Saturday – Non-Working Day |
| 20-Aug-2023 | Sunday Holiday |
| 21-Aug-2023 | Worked With Protocols Based Sensors  1. I2C communication protocol(only two allowed).  2. Interfacing OLED with ESP32.  3. Temperature Sensor – SHT31(inbuilt in esp32).  4. Light Sensor (BH1750 ) – I2C.  5. Accelerometer and gyroscopic Sensor. |
| 22-Aug-2023 | 1. Presentation on Smart street using Different Sensor  2. Communication Protocols  3. SPI(synchronous communication protocol)  4. UART,USART(Universal Asynchronous Receiver Transmitter)  5. I2C(Inter-Integrated Circuit)  6. WIFI using ThingSpeak(ESP 32) both WIFI and BLE |
| 23-Aug-2023 | 1. IoT basic on LORAWAN Technology.  2. Saw machines such as SMBMATE and PCBMATE.  3. Explaination on how PCB is made.. |
| 24-Aug-2023 | WiFi ThingSpeak ultrasonic sensor and inbuilt hall sensor  Transferring the sensors data to the Iot Cloud platform like ThingSpeak. |
| 25-Aug-2023 | Controlling led using ThingSpeak WIFI  Bluetooth connection with phone  Controlling light using Bluetooth |
| 26-Aug-2023 | Controlling sensor using Bluetooth and checking the reading in the serial monitor as well as in SmartPhone. |
| 27-Aug-2023 | Sunday - Holiday |
| 28-Aug-2023 | LORA Technology explanation and login into the ThingsMate software.  we are using ABP activation in which each and every user is given with APPkey and device key.  Using the ThingsMate software we are creating a network layer in which data is sent from gateway. |
| 29-Aug-2023 | Using Iot Cloud Services like Things Mate and application layer software Ubi dots.  basic program on LORAWAN transferring of simple bits to the think mate server |
| 30-Aug-2023 | interfacing a ultrasonic sensor with ESP 32 and transferring the data to the network server(things mate )using LORAWAN  The data can be viewed in the application layer via website called ubi dots. |
| 31-Aug-2023 | Installation of STM CUBE IDE, STM Programmer, Serial Port Utility.  Selection of Board in STM Cube Ide and Explanation on PIN Diagram of STM32F103C8T6 |
| 1-Sep-2023 | Started with the introduction of STM-32 blue pill chip (STM32F103C8T6) |
| 2-Sep-2023 | Saturday Holiday |
| 3-Sep-2023 | Sunday Holiday |
| 4-Sep-2023 | Basic program of printing Hello World using STM 32 using STM 32 to turn on and off the led |
| 5-Sep-2023 | Toggling of led with STM 32  interfacing multiple sensors with STM 32  Tilt Sensor, PIR Sensor |
| 6-Sep-2023 | Program on Interrupts using STM 32 |
| 7-Sep-2023 | Analog Sensor interfacing with STM 32 using ADC concept Interfaced TILT Sensor with STM 32 |
| 8-Sep-2023 | Introduction to Communication Protocols like I2C, SPI.  SPI Serial Peripheral Interface  I2C Inter Integrated Circuit |
| 9-Sep-2023 | Saturday Hoilday |
| 10-Sep-2023 | Sunday Holiday |
| 11-Sep-2023 | Interfaced I2C communication protocol based sensor BH1750 (LDR) Sensor with STM32 |
| 12-Sep-2023 | Introduction to STM 32 WL development kit with LoRa Equipped Board  Basic program of relay turn on and turn off, which in inbuild in the development kit. |
| 13-Sep-2023 | led turn on and off with STM WL development Board  Oled interfacing with STM 32 development kit, OLed is a I2C obeyed Protocol device. |
| 14-Sep-2023 | Introduction to SPI communication Protocol  Interfacing RFID using SPI protocol with STM 32 WL development Board. |
| 15-Sep-2023 | Introduction to LoraWan using STM 32 WL development Board.  Programs on Interfacing Sensors With STM 32 WL board and Transferring the Sensor data to the Iot Cloud Service (Things Mate) using LoraWan. |

**Software Learnt:**

**STM32 IDE:**

1.**Integrated Development Environment (IDE):** STM32 IDE is an integrated development environment tailored for STM32 microcontrollers developed by STMicroelectronics.

2.**Based on Eclipse:** It is built on the Eclipse platform, providing a familiar environment for developers.

3.**Cross-Platform:** Supports Windows, macOS, and Linux operating systems, ensuring compatibility across various platforms.

4.**STM32CubeMX Integration:** Integrated with STM32CubeMX, a graphical tool that allows users to generate initialization code and configure peripherals graphically.

5.**Code Generation:** Generates initialization code for peripherals, middleware, and application code, speeding up development.

6.**Code Editing and Debugging:** Offers features for code editing, syntax highlighting, code completion, and debugging functionalities, essential for efficient development and debugging processes.

7.**Peripheral Configuration:** Provides tools for configuring and fine-tuning STM32 microcontroller peripherals, such as GPIOs, timers, UARTs, SPIs, etc.

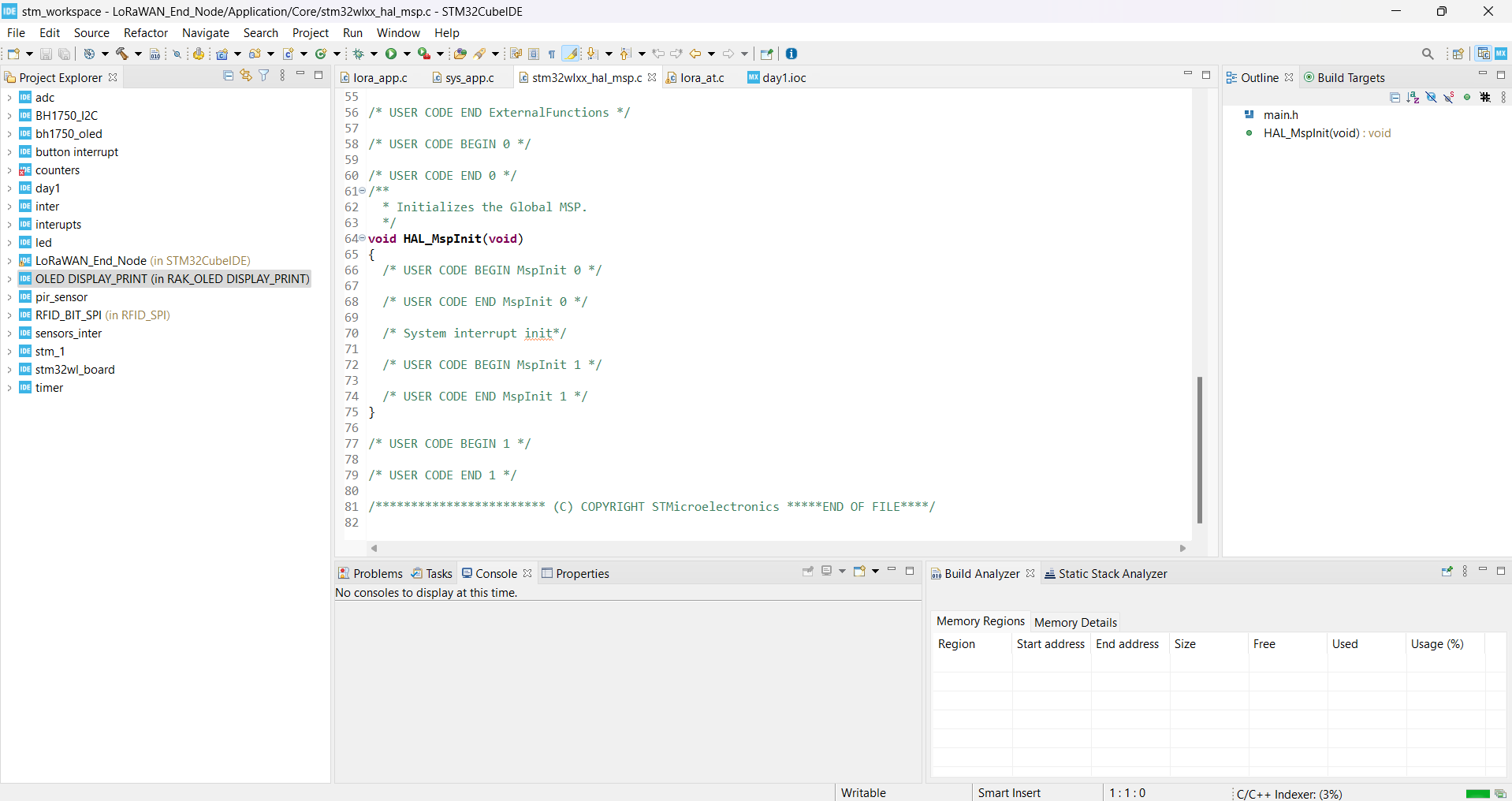
8.**HAL (Hardware Abstraction Layer) Library:** Supports HAL library, a set of APIs abstracting low-level hardware access, facilitating portability across different STM32 microcontroller families.

9.**Real-Time Operating System (RTOS) Support:** Integrates support for FreeRTOS, enabling developers to create applications with real-time capabilities.

10.**Version Control Integration:** Supports integration with version control systems like Git, allowing collaborative development and version tracking.

11.**Community and Support:** Access to a large community of developers, forums, and documentation, providing assistance and resources for troubleshooting and learning.

12.**Updates and Maintenance:** Regular updates and maintenance releases to incorporate bug fixes, improvements, and support for new STM32 microcontroller series and features.



**Summary:**

This internship provides a comprehensive hands-on experience in developing applications using ESP32 and STM32 microcontrollers with a focus on LoRaWAN technology. Interns will gain practical skills in IoT device development, wireless communication protocols, and embedded systems programming.

**Key Learning Objectives:**

**ESP32 Development:**

* Learn to program ESP32 microcontrollers using the Arduino IDE and ESP-IDF framework.
* Understand ESP32 hardware features and capabilities, including Wi-Fi and Bluetooth connectivity.
* Develop IoT applications leveraging the power and versatility of the ESP32 platform.

**STM32 Development:**

* Gain proficiency in programming STM32 microcontrollers using STM32CubeIDE and HAL library.
* Explore STM32 hardware peripherals and functionalities for sensor interfacing, motor control, and communication protocols.
* Develop embedded applications targeting STM32 microcontrollers for diverse IoT use cases.

**LoRaWAN Technology:**

* Learn the principles and architecture of LoRaWAN for long-range wireless communication in IoT.
* Understand LoRa modulation techniques and LoRaWAN protocol stack layers.
* Implement LoRaWAN end-device applications on ESP32 and STM32 platforms for transmitting sensor data over long distances.

**Internship Activities:**

**Hands-on Projects:**

Engage in practical projects to design and implement IoT solutions using ESP32 and STM32 microcontrollers.

Develop sensor nodes, gateways, and edge devices for collecting, processing, and transmitting data.

Integrate LoRaWAN communication for long-range connectivity in IoT deployments.

**Code Development:**

* Write and debug firmware code in C/C++ for ESP32 and STM32 microcontrollers.
* Utilize hardware peripherals and libraries to interface with sensors, actuators, and communication modules.
* Optimize code efficiency and resource utilization for embedded systems applications.

**Testing and Validation:**

* Conduct testing and validation of IoT devices to ensure reliability, performance, and compliance with specifications.
* Use simulation tools, hardware-in-the-loop testing, and real-world deployments for validation purposes.
* Troubleshoot issues and refine solutions through iterative development cycles.

**Benefits and Outcomes:**

* Gain practical experience in embedded systems development, IoT, and wireless communication technologies.
* Acquire proficiency in programming ESP32 and STM32 microcontrollers for diverse applications.
* Develop a deep understanding of LoRaWAN technology and its role in IoT connectivity.
* Collaborate with experienced engineers and mentors to tackle real-world challenges in IoT development.
* Receive valuable feedback, guidance, and networking opportunities for future career advancement in the field of embedded systems and IoT.

**Conclusion:**

This internship offers a unique opportunity for hands-on learning and skill development in ESP32, STM32, and LoRaWAN technologies. Interns will engage in practical projects, code development, testing, and documentation activities, gaining valuable experience and expertise in the rapidly evolving field of IoT and embedded systems.

# REFERENCES

1. "Building Arduino Projects for the Internet of Things: Experiments with Real-World Applications" by Adeel Javed, June 2016
2. "Internet of Things (IoT) Technologies for Healthcare: Third Edition" edited by Chinmay Chakraborty, Valentina E. Balas, and Sudip Misra, December 2017
3. "Internet of Things: Principles and Paradigms" by Rajkumar Buyya, Amir Vahid Dastjerdi, and Satish Narayana Srirama (2016)
4. "The Internet of Things (The MIT Press Essential Knowledge series)" by Samuel Greengard (2015)
5. "IoT Inc: How Your Company Can Use the Internet of Things to Win in the Outcome Economy" by Bruce Sinclair (2016)
6. "Internet of Things: Architectures, Protocols, and Standards" edited by Houbing Song, Ravi Srinivasan, Tamim Sookoor, and Sabina Jeschke (2017)
7. "Getting Started with the Internet of Things: Connecting Sensors and Microcontrollers to the Cloud" by Cuno Pfister (2011)