Industrial workers Safety and Automation

An Internship Report

submitted by

KARTHIKEYAN (EDM19B001)

in partial fulfilment of requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

Department of Electronics and Communication Engineering
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, DESIGN AND MANUFACTURING KANCHEEPURAM

OCTOBER 2022

DECLARATION OF ORIGINALITY

I, **Karthikeyan**, with Roll No: **EDM19B001** hereby declare that the material presented in the Project Report titled **Industrial workers Safety and Automation** represents original work carried out by me in the **Department of Electronics and Communication Engineering** at the Indian Institute of Information Technology, Design and Manufacturing, Kancheepuram.

With my signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property. I have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I have understood that any false claim will result in severe disciplinary action.
- I have understood that the work may be screened for any form of academic misconduct.

Karthikeyan

Place: Chennai

Date: 26.October.2022

CERTIFICATE

This is to certify that the report titled **Industrial workers Safety and Automation**, submitted by **Karthikeyan (EDM19B001)**, to the Indian Institute of Information Technology, Design and Manufacturing Kancheepuram, for the award of the degree of **BACHELOR OF TECHNOLOGY** is a bonafide record of the work done by him/her under my supervision. The contents of this report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Rohini P and Dr. Karthik C
Internal Guide
Assistant Professor
Department of ECE
IIITDM Kancheepuram, 600 127

Vikram Rastogi
External Guide
Founder, CEO
Hacklab Solutions
Bangalore

Place: Chennai

Date: 26.October.2022

ACKNOWLEDGEMENTS

I would like to thank CEO of Hacklab solutions Mr. Vikram Rastogi and COO Mr. Bharath Bhushan for providing me with this internship opportunity and resources required to carry out the project. I am would like to extend my gratitude towards and thank my mentor faculty from IIITDM Kancheepuram Dr. Rohini P and Dr. Karthik C for their valuable inputs and guidance.

ABSTRACT

Nowadays lot of accidents occur in the industrial workspace due to crash accidents where the worker comes in contact with a vehicle or vehicles come in contact with each other. These vehicles are generally forklifts, which are used to transport heavy goods inside factories, industries and warehouses. The field of vision of forklift is limited as it carries heavy loads and objects. So it becomes difficult for the driver to have clear understanding of surroundings and this leads to accidents. Therefore Embedded devices will play an important role in this field, it provides cost efficient, precise and robust solution for industrial safety. We should also note that these embedded devices comsume a lot of power, hence we should reduce the power consumption of these devices so that they are power efficient.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

IoT	Internet of Things
ACK	Acknowledgement
TWR	Two Way Ranging
CAN	Controller Area Network
MCU	Microcontroller
ADC	Analog to Digital Converter
UART	Universal Asynchronous Receiver Transmitter
SPI	Serial Peripheral Interface
UWB	Ultra Wide Band
IC	Integrated Circuit
SPIFFS	SPI Flash File System
Tx and Rx	Transmitter and Receiver

CHAPTER 1

Introduction

All the major indsutries like electronics, food and beverages, chemical etc, depend on storage and transporation of products and goods. Storage and transporation forms an important aspect of production and export cycle of these industries. Therefore it plays an important role in economic and business flow of these industries. It involves lot of workers and heavy machineries used to transport heavy goods within the storage facility. The workers will be checking the packages and if the goods are properly sorted and arranged. The organization, transportation, packaging, status and dispatch of the goods are done by the workers, thus the manpower in the storage facility is tremendous. Huge amount of cost is spent on labours for the proper functioning of work. Manufacturing industry is a major contributor to injury, morbidity and mortality in India. After road traffic collisions, workplace accidents are the next major cause of injuries in India. The International Labour Organization (ILO) has estimated that there were 47,000 fatal and 44.1 million non-fatal work-related accidents in India in 2003. Industrial accidents are important contributors to workplace injuries and preventable death and disability.

Injuries put an enormous financial burden on the poor industrial workers. Aside from hospitalization and treatment expenses, injuries also lead to a decline in earnings and rehabilitation cost. Moderate and severe grade injuries may lead to poor quality of life and have long-term psychosocial impacts. People working in factories and construction sites are at significantly greater risk due to heavy objects, loads, tools, forklift trucks and production lines which run throughout the day. Therefore site safety of workers is a crucial aspect of storage and transportation of any industry.

India is moving on a fast trajectory of growth, development, and economic prosperity. This is likely to push the injury morbidity and mortality rates and burden upward. Successful implementation of injury prevention policies and programs will lead to multiple benefits including reduction in fatal and non-fatal accidents, reduction in the number

and severity of disabilities caused by injuries, an increase in the number of productive working years, a decrease in the costs associated with treatment and rehabilitation of trauma victims. There is a need to wake up to the rising trend of injuries as serious contributors to morbidity and mortality.

1.1 Problem Statement

From small scale to large scale industries, storage and transportation is an important aspect of the industry. Everything from small goods to heavy goods are stored in warehouses and other storage facilities. Therefore these resources have to transported to appropriate location in order to use them efficiently, therefore heavy machines and workers are involved in organizing and transportation of resources. Forklifts play a major role in transportation inside the warehouse. Since these forklifts are loaded with large and heavy goods, their feild of vision is highly limited. Therefore these vehicles are prone to accidents where it comes in contact with any worker or another vehicle in the storage facility. The field of vision is almost completely distrupted during a sharp turning or corner. In this case chance of accidents are very high. Furthermore depending on the speed of the vehicle or the forklift incoming the accident can be lethal. Therefore it is absolutely crucial to implement safety mechanism to safeguard and protect workers from any accidents. Thus Iot enabled Embedded Devices are used to implement safety mechanism to prevent and avoid accidents.

These Embedded devices in turn consume lot of power, depleting the power source thereby reducing the active operating time of the device. These devices operate only for certain time intervals during the complete active time. Therefore sleep is implemented in these devices to reduce the power consumption from these devices which in turn increases the active time and battery life of the device.

CHAPTER 2

Problem Analysis and Solution

2.1 Embedded Devices and IoT

Embedded Devices could play an important in providing safety and protection to industrial workers. They have specially designed embedded systems circuits which are application oriented and are of specific use case. So we would be implementing one embedded device in workers helmet and one in the vehicle. Then a connection mechanism must be established between these two devices, so that they could exchange messages. They need to connect with each other so that they could understand or notify the current status. The range between the devices is a crcucial parameter to compute and take decisions based on. The range gives us the distance between the worker and the vehicle. After computing the range we will be able to make decisions like whether to give alert signal, buzzer or to apply brakes.

2.1.1 Embedded System

Two different Embedded Systems will be implemented for helmet and vehicle. The reason behind this is that vehicle requires different functioning and safety features as compared to helmet. Features of system implemented in the helmet are given below:

- This embedded system interacts with the one in the vehicle with the help of an IoT enabled IC.
- It has a mechanism to reset the system, in case if the system goes into loop.
- It functions with the help of battery as it will not be directly connected to power supply.
- There is an indicator LED to show battery status and interaction status if any.
- It has an charging IC and connector with the help of which the battery can be charged when required.

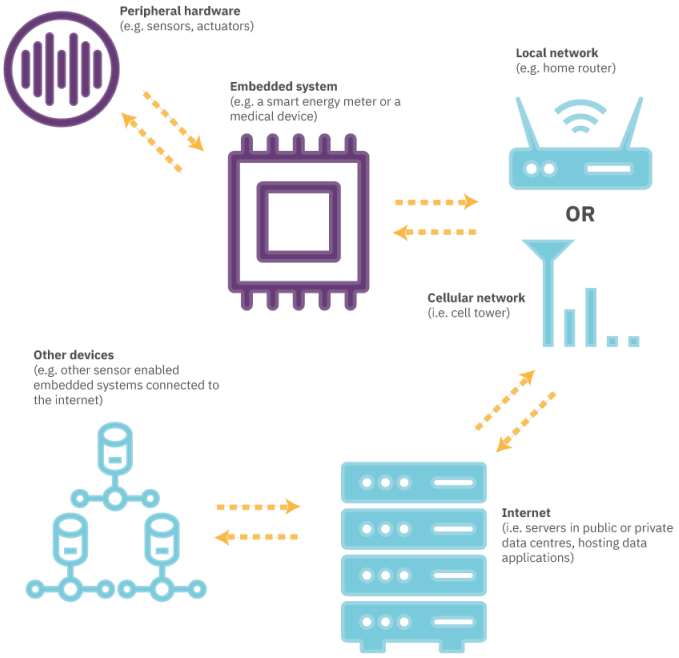

Figure 2.1: Embedded Iot Devices [source]

- Atmega is the main Microcontroller in this system, therefore it is an atmega system.

Features of system implemented in the vehicle are mentioned below:

- This embedded system interacts with the one in the helmet with the help of an IoT enabled IC.
- It also has a resetting mechanism, so that loops and connection losses can be avoided.
- It has a DC-DC converter as it will be directly connected to power supply (DC). The converter converts the incoming DC voltage to 5V DC.
- There are indicator LEDs to display the status.
- There is another embedded system implemented in vechile to perform functions like applying brakes when closeby.
- The system sends messages via CAN protocol through the CAN IC, which is interpreted by the other system in the vehicle. When worker is closeby it will send approproate CAN message, after interpretation of which brakes will be applied.
- ESP32 is the main Microcontroller in this system, therefore it is an esp system.
- There are four of these esp embedded systems mounted to the four corners of vehicle to complete the ranging region

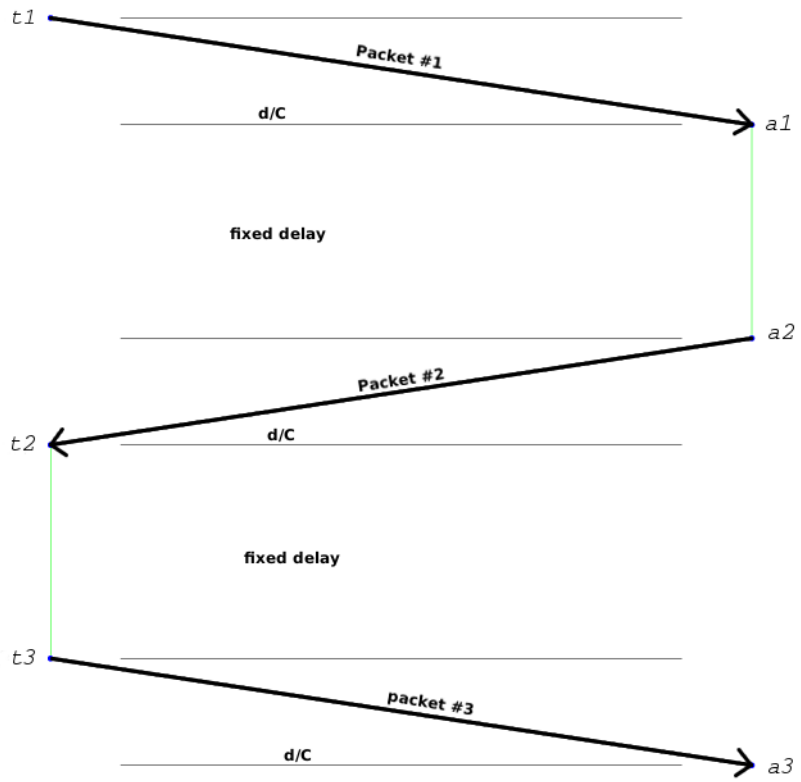

Figure 2.2: Two Way Ranging Mechanism [source]

2.1.2 IoT and Communication

The two embedded systems communicate with each other to find the range and take imporant decisions. The two systems sends series of messages to each other after which the range is computed. It uses two way ranging mechanism to compute the range. Which is a mechanism by which, with the help of series of messages between two devices, the range between them is calculated by using the delay in receiving the messages.

The atmega system will initiate the two way ranging, ACKs for which are transmitted by the esp system. The first message is to identify if there are any esp system in the surrounding range. If there is an esp system, then it will transmit ACK, after which the two way ranging mechanism will start between atmega and esp system. It starts with a POLL message from atmega, followed by ack given by the esp system. Then the RANGE message is sent by the atmega system, after receiving which a RANGE_REPORT packet is sent by the esp system. The range packet contains the

computed range between the two system depending on the delays created during the transmission and receiving of the messages. Once computed the range is obtained in both the sytems through the packets, depending on the value of range the CAN messages are sent by the esp system.

Iot is used to upload the log data to cloud if required. During the working of the system to check if all the parts are functioning properly we log the data sent by each device. Therefore these logs can be used to understand if the system is working properly. If any bug or failiure occurs, traceback can be done using the logs. These logs are uploaded to the cloud with the help of Iot.

CHAPTER 3

Working of Hardware and Software

3.1 Hardware

The hardware for the atmega and the esp system are different as they are powered through different type of sources and they perform different functions. .The atmega system is powered through Lithium ion battery of 3.8V. Since the logic level operations of atmega is at 3.3V, the voltage is reduced by voltage divider. The output of the divider is at 3.3V, which is then given to load switch through which it is given to the entire system.

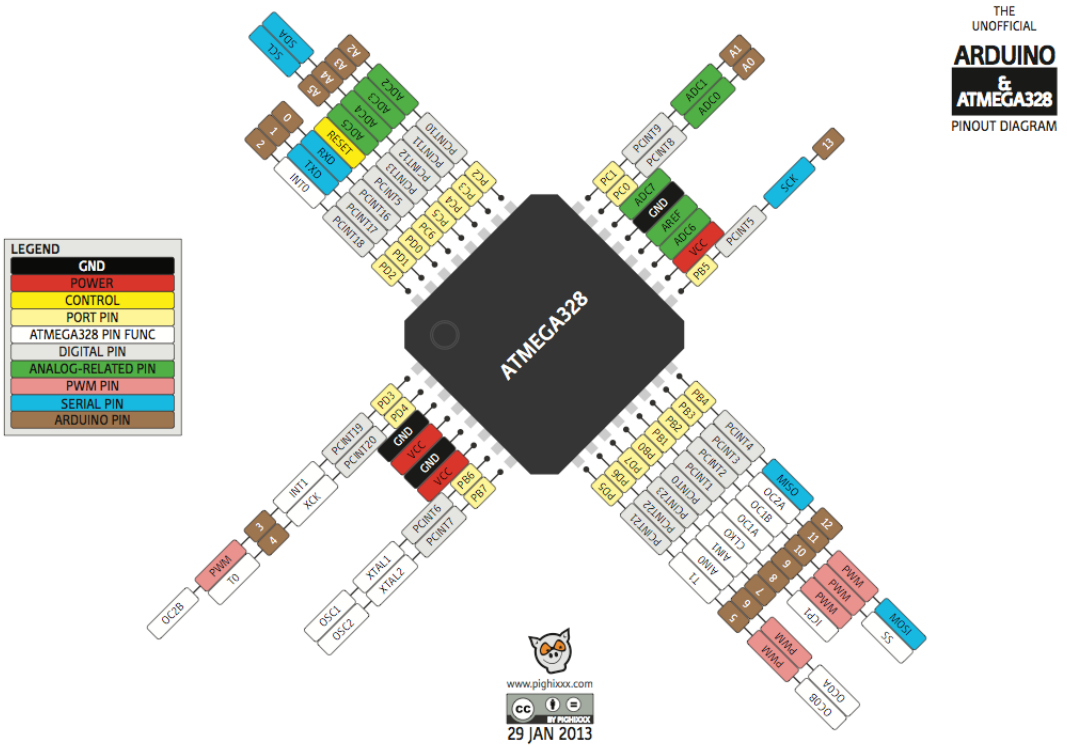

Figure 3.1: Atmega Pinout [source]

The input voltage of 3.3V powers up the atmega and all other ICs in the system. The atmega is connected to an Ultra wide band communication for controlling transmission

and receiving of messages through SPI protocol. SPI (serial peripheral interface) protocol is a high speed data exchange protocol between master and slave. It has four pins - MOSI, MISO, SCK and SS.

The atmega is also connected to an ATtiny which is used to reset the system when there is no response from UWB IC. The ATtiny is in turn connected to load switch. The atmega gives continous pulses to the ATtiny, which in turn listens to the pulses. There is a timeout timer for the ATtiny, within which if it doesnt receive pulse from atmega, then it will give high pulse to enable pin of load switch, which then will reset the power line of the entire system. Then reset happens and the same is indicated through the LED.

A seperate IC is used for resetting so that when the main Microcontroller goes into idle state or is not responding, then it can be identified by other IC and necessary decisions can be taken from there. If only the main MCU is used as watchdog timer, then if it is inside a loop and not functioning properly then a hardware reset would be required to put the system on line again. Which may not be a good option as the device needs to work continously for long time in industrial conditions. It would affect the workers protection and safety if it is looping through UWB IC packets and not performing the main function. So in that case the second IC would identify and reset the system thus putting all the parts in their proper functioning modes.

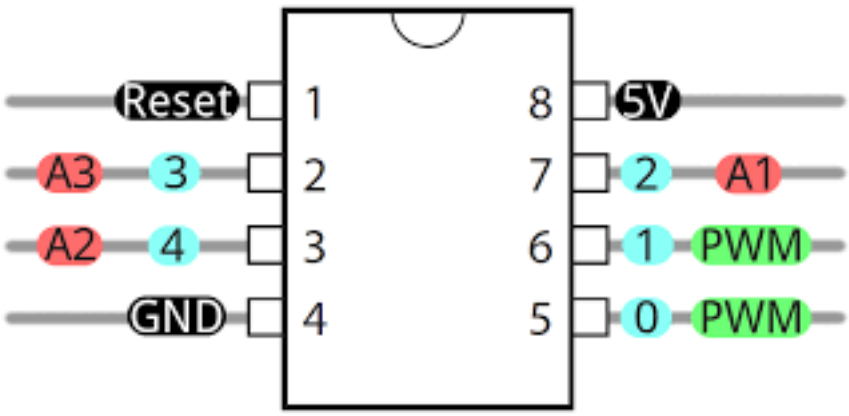

Figure 3.2: ATtiny Pinout [source]

The system is powered up by 3 way battery. The output of the battery is connected to the PCB. From where it goes to voltage regulator then the components. For charging the battery, there is a connector and charging IC in the PCB. The micro USB connector is used to provide input power to the PCB, the power coming from the connector

goes to the charging IC. The charging IC in turn charges the battery with the incoming power. The same is indicated in the LED once charging is initiated. Once charging starts the atmega detects it. Voltage in the battery is read using ADC (Analog to Digital Converter) in atmega.

The ADC pins of atmega are given in the pinout, from that we can use only the pins labelled as ADC for getting the input from the battery. The arduino ADC is of 10 bits resolution and gives the voltage level of battery to that precision. During charging the atmega goes into charging loop, where it writes the computed battery voltage through ADC to UART. The Rx and Tx UART pins of the arduino is connected to the connector. Therefore serial prints can be read from the connector by using any computer. The serial prints from the device can be used to debug or understand the status of the device.

Therefore with the help of a PC we can see the amount of voltage the battery has and how much it has increased in the past few hours or minutes of charging. This can also be used to understand the battery charge holding ability, discharge time etc. The device doesnt communicate with the esp system while under charge. The system attains full charge at around 4.3V and the same is indicated in the led to show that the device is completely charged. The sytem can operate untill the voltage in the battery drops below 3.1V, after which the atmega wont be able to perform logic level operations, which usually requires 3.3V.

UART (Universal Asynchronous Receiver Transmitter) is a Asynchronous communication protocol. Where the rate at which data is transmitted and received is set by the baud rate. Both the devices communicating with UART should have the same baudrate so that the receiver can interpret the messages properly.

The UART frame format consist of start bit, stop bit followed by 8 bits of word data containing the information to be transeferred. The UART frame format is given below:

Therefore while reading the serial prints from the device, the computer's software baud rate also has to be set to the same value. Once charging is removed from the device, the atmega comes out of the charging loop and starts to execute the normal code. It prints battery voltage at a time interval just to know the status of input voltage. As charging is removed the power going inside the battery stops and charging IC also

goes into idle state, the same is shown in the led as status of the deivce, so that we can understand that device is removed from charge.

The atmega system tries to communicate with esp system once operating amount of voltage is available in the battery. It sends the initial signal and then proceeds with the two way ranging as we have discussed in the previous chapter. It starts with the beacon message and checks whether it got acks from the esp system. If it doesnt get any acks, after a particular period of time it sends the beacon message again. If it gets any acks then it proceeds with the two way ranging signals for that particular esp system.

Now let us discuss about the hardware of esp system, the esp system is powered via dc power. The power input is sent to DC-DC converter which converts 7 to 27V to 5V. Therefore we can give a wide range of input voltage, but the amount of input power must be considerate or else it may damage the system. The power then goes through a load switch then to the entire system. The general load switch circuit is given below:

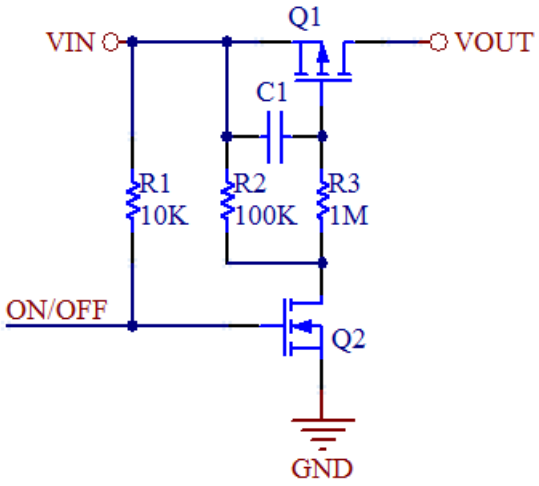

Figure 3.3: Load Switch [source]

There is an ATtiny as second Microcontroller for this device also and the resetting mechanism is the same as the atmega system. The 5V is then converted to 3.3V with voltage regulator, as the operating voltage of ESP32 Microcontroller is 3.3V. Therefore in order to perform logic level operations and other peripheral activities like ADC, it has to be supplied with 3.3V. The ESP32 pinout diagram is given below.

The ESP is the main MCU of the device, it is connected to the UWB IC and CAN IC for communication. It receives messages from the atmega system through the UWB IC,

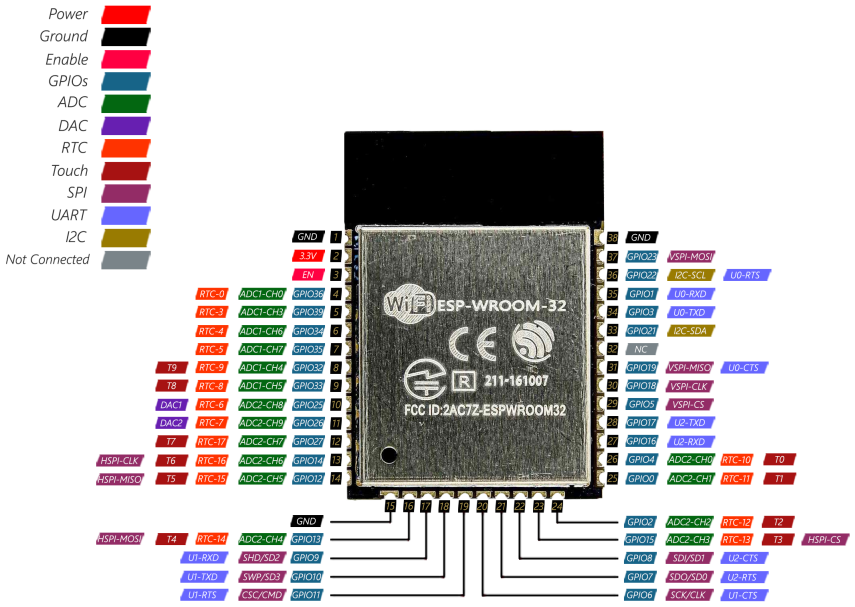

Figure 3.4: ESP32 Pinout [source]

then the esp is interfaced with UWB IC through SPI (Serial Peripheral Interface). Once it receives the packets from atmega system, it interprets them and sends appropriate CAN messages through the CAN IC. The CAN is controller area network protocol and it is used to send message over a single bus where several CAN nodes are connected. The CAN is used to send message without any attenuation and noise in the signal and over long ranges. This is due to the way in which CAN protocol is built, it sends two signals CAN-H and CAN-L, any error in the transmission will be generated in both lines, therefore can be identified and rectified in the receiver. The standard CAN message format is given below.

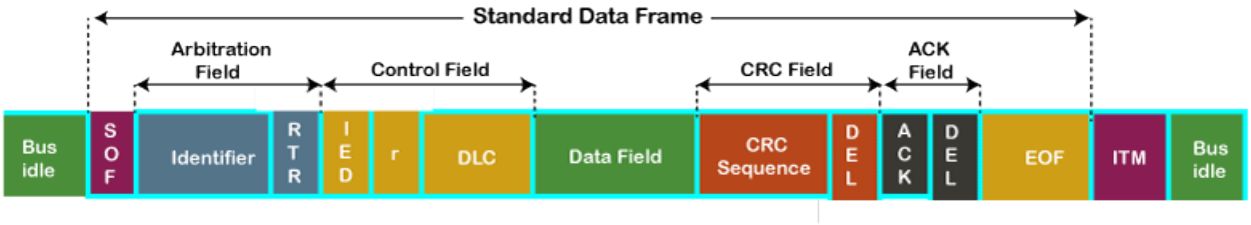

Figure 3.5: CAN Message Format [source]

Hence once CAN message is sent the other systems in the line will be able to interpret it and understand the reason due to which a particular CAN message was received. After which the corresponding action will be taken by the receiving node. For exam-

ple in case if the worker helmet is nearby, then the esp system will send a emergency CAN message, upon receiving which the vehicle will be stopped by the main system connceted to the brakes of the vehicle. This will be the general working of esp and the atmega system. The CAN protocol implementation contains terminal resistance in both the ends of the CAN bus. This is because to eliminate the bounce back of the signals reaching the ends of the bus. If bounce back happens then the signal will superimpose with the existing message signal and create lot of noise. Generally 120 ohms terminal resistance is used at both the ends of the CAN bus.

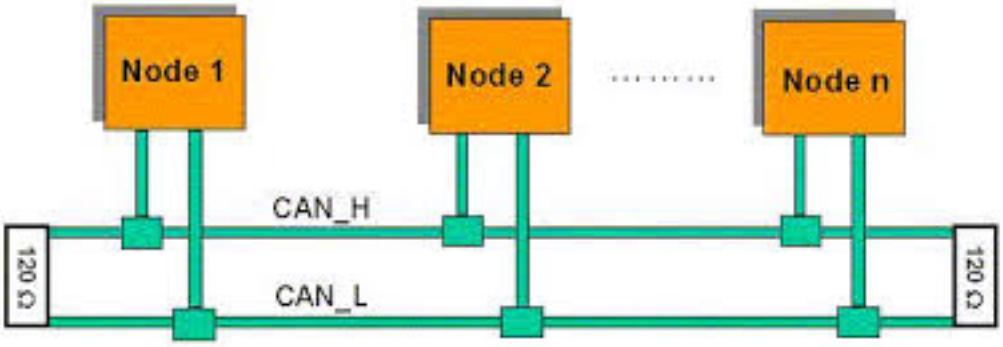

Figure 3.6: CAN Architecture [source]

3.2 Software

In the firmware part of the systems, the functioning of the devices are programmed into the flash memories of the Microcontrollers. In atmega system, the program is loaded into atmega and in esp system, the program is loaded into esp. The program is first coded in a seperate machine and an executable binary or hexadecimal file is created after compiling the high level C/C++ code. This binary file is then flashed into the microcontroller. In the atmega system the program is loaded into flash through SPI protocol. So SPI pins of atmega will be active during this action. Therefore if required power should be provided so that the pins of atmega function properly. The program is loaded from the PC through SPI programmer into atmega.

In esp system UART protocol is used to upload the program. Hence an uart programmer is used to upload the code into ESP32. These programmers convert USB to the required protocol - SPI or UART.

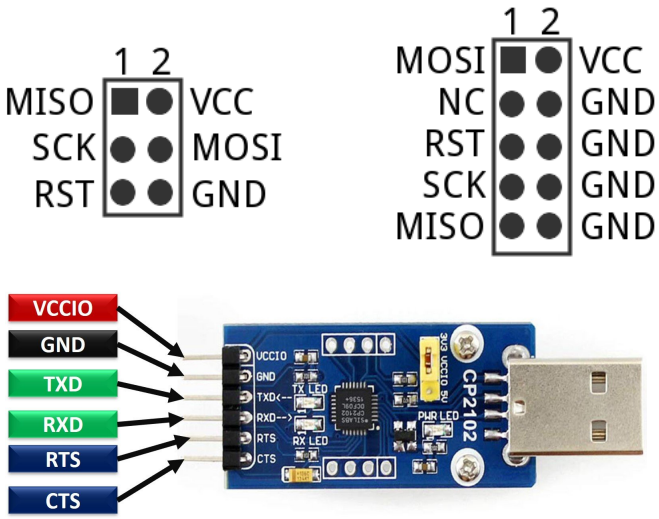

Figure 3.7: SPI and UART programmer pinout [source]

The software used for firmware is arduino ide. Initially the setup function gets executed, therefore initializations are done insdie setup function. The GPIOs (general purpose input outputs) initializations, Serial initialization and UWB IC initialization are done inside the setup function. The corresponding serial prints are printed as the initializations gets completed.

The loop function contains the code that runs repeatedly. The setup function only executes once after hardware reset of the system. The loop function contains seperate functions for checking battery voltage, checking power input from connector, handling the sent and received data (from UWB IC) and function to send beacon signal at regular intervals.

The send beacon function takes care of the option of receiving and not receiving an ACK from esp system, in which case it starts two way ranging and sends beacon again after a time interval respectively. The type of message and the payload (data) to be transmitted can be specified in the transmit function of the firwmare. Upon which the corresponding UWB IC function will be called and will transmit the required message.

Upon receiving a message, the handle recevied packet function calls the functions which filter and call specified functions according to the received packet. Then according to the packet the ranging is performed and computed, which is then transmitted in the Range packet. The range report packet is the acknowledgement for the range packet

from the esp system, which marks the end of the two way ranging process.

Now let us discuss the firmware of esp system, the setup function contains initializations for CAN IC, UWB IC, Serial Port and SPIFFS (SPI Flash File System). The SPIFFS is a storage used for storing files, checking file system consistency and more.

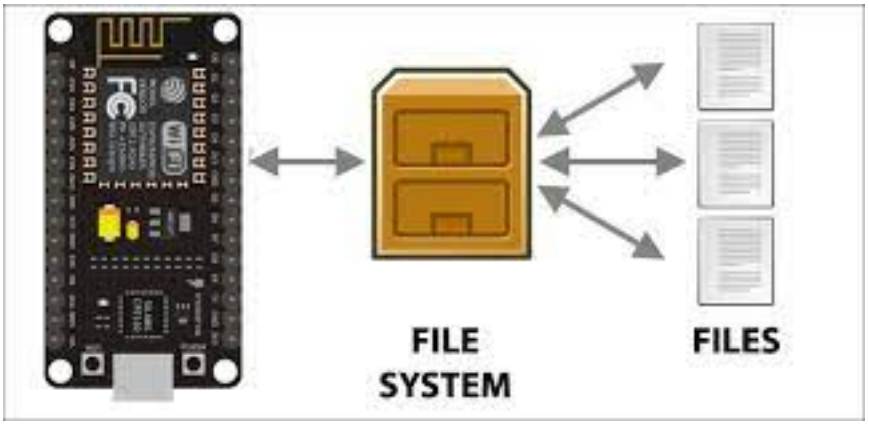

Figure 3.8: SPI Flash File System [source]

The Sent and the receive handlers are intialized for the UWB IC, that is in case any message is sent or received, then the corresponding call back function is executed. The call back function sets the flag, upon detecting which the backend functions are called. The backend functions service the incoming packet by calling the corresponding UWB IC functions. There is a seperate library for the UWB IC, which is used in the firmware.

The configuration for the alert and the emergency distance are done in the firmware for the esp system. These configurations have the value of distance in meters within which the alert and emergency zones are defined. Let us understand what are alert and emergency zones. The distance between the worker and the vehicle (atmega and esp system respectively) in which alert signal is sent as CAN message is called alert zone, it is a circular zone of configurable radius within which the alert signal is generated. Accordingly the distance within which emergency signal is given and brakes are applied is called emergency zone. These are hardcoded into the firware of the esp system and can be changed by making necessary changes to firmware and flashing it again.

The esp systems interact with each other also in order to avoid vehicle-vehicle contact and crashing. Sicne four esp systems are mounted in each vehicle we have to take care that the esp systems in one vehicle dont interact with each other. This is done by

implementing ignore id in firmware, where the IDs of the other three esp systems are hardcoded as ignore IDs. The esp system's own ID is hardcoded as Device ID. Thus if it receives packets from system within the same vehicle, it will look into the source of the packet and compare it with ignore ID. If Ignore ID is present then it ignores and discards the packet.

3.3 Power Consumption Reduction

The atmega system consumes lot of power during its operating time. Since it draws it power from 3.8V battery it will difficult to operate for long time without charging. The average life time of atmega system comes around 7 hours. The power consumption from atmega system was measured with a digital multimeter. It consumes a peak power of 0.475 W. Which is 3.3V at 144mA current. The current consumption is mainly due to the amount of components in the system. The main power consumption comes from Atmega, ATtiny and UWB IC. If the amount of current drawn by these ICs can be reduced then it will increase the life time of the atmega system.

The atmega system consumes peak power when UWB IC is transmitting and receiving messages. The two way ranging process happens for only a short duration of time. Thus if we are able to find the nearby devices and complete the ranging process, we can send the ICs to sleep mode for the remaining calculated amount of time which in turn will reduce the overall power consumption.

The ICs should go to sleep mode at a particular time so that it does affect the normal functioning of the system. The time at which sleep mode can be implemented are:

- During the initial beacon signal timeout - broadcast timeout sleep
- After completing the ranging process - Ranging complete sleep

Broadcast Timeout Sleep: The atmega system sends the initial beacon message and waits for a short period of time, if there is no ACK from esp system nearby, then before sending the second beacon it goes into sleep for sometime. The time interval for this sleep is 65ms. The Atmega and UWB IC are put to sleep for this duration.

The LowPower.powerDown function puts the Atmega to sleep and turns off ADC and BOD (brown out detector). The DW1000.deepSleep() function puts UWB IC to sleep. The sleep duration is given as argument to powerDown function as seen above. After the period is over atmega wakes up automatically and starts executing code from the next line. It executes DW1000.spiWakeup function which wakes up UWB IC, thus bringing the system back into normal operating mode. After waking up the initialization function is executed to setup and initialize the UWB IC.

Ranging Complete Sleep: The atmega systems after completing the ranging process with esp system goes to sleep for certain duration. This is because all the emergency and alert signals would have been delivered if distance is under threshold. So since the decisions will be taken based on the interaction, the system can go to sleep for a certain amount of time. The time interval for this sleep is 310ms. The Atmega and UWB IC are put to sleep for this duration.

The broadcast timeout sleep and the ranging complete sleep together puts Atmega and UWB IC to sleep in an efficient manner and reduces the amount of power consumed by the system. The ATtiny, which is used for resetting the system, has its own implementation of sleep mode. The ATtiny contributes significantly towards current drawn by the atmega system. Since the role played by ATtiny is specific, the firmware is implemented to maximise the amount of sleep and reduce the power consumption.

The ATtiny is programmed such that it remains in sleep with a backround timer running. When the timer runs to a specific value, it wakes up, resets the system and goes to sleep. The system reset also resets the timer so it starts from beginning again. If ATtiny receives pulse from atmega then it knows that the system is functioning proeprly, hence it resets only the timer and goes to sleep. As long as it receives pulses from atmega it keeps resetting the timer and acts as a watchdog over the system. Hence the pulse pin from atmega is set as interrupt pin as it needs from wake from sleep each time. Therefore by this mechanism the ATtiny remains in sleep almost all the time except for small amount of time when it functions. Therefore this firmware implementation reduces the power consumption significantly.

CHAPTER 4

Results and Discussion

4.1 Collision Avoidance

The collision between vehicle-vehicle and worker-vehicle is avoided with the help of TWR. Near and emergency zones are defined and are hardcoded in the firmware of the device. When the worker enters the near zone, the alert signal is given and if the worker enters the emergency zone, the emergency CAN message is given and the brakes in the vehicle are applied.

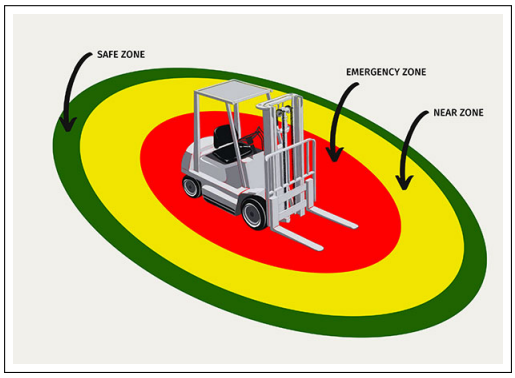

Figure 4.1: Collision Avoidance Zones [source]

When the atmega system (worker) and esp system (vehicle) interact and the range obtained is greater than near zone. Then the worker is in safe zone and there is no hazard. In this range the systems will be communicating but there will be no actions taken as the worker is in safe zone. The TWR mechanism happens in a short span of time. Each ranging mechanism between an atmega and esp systems takes 50 ms maximum. The time to initiate the ranging, compute the range and taking the effective decisions takes less than 500 ms. Only a short period of time should be taken for the working of the systems as the breaks should be applied as early as possible. There should not be any delay, which will lead to accident.

Figure 4.2: Worker-Vehicle Collision Avoidance [source]

In the case of two vehicles the collision is avoided through the interaction of two esp systems. The esp system in both the vehicles interact with each other and perform the two way ranging mechanism, through which they will be able to obtain the range between them. Once the range is obtained the decision is made based on the value of the range. The emergency and near zones are previously defined in the firmware of the systems. So according to the data stored the alert (near) and the emergency signals are given. When the emergency signal is given, the brakes of the vehicles are applied automatically.

Figure 4.3: Vehicle-Vehicle Collision Avoidance [source]

4.2 Power Consumption Reduction

After the implementation of power consumption reduction the average amount of current drawn by the device has reduced. Since the system goes into sleep during the computed time intervals, the functioning of the system is also not affected and the power

consumption is also reduced. The device draws more power only during the functioning part of the cycle. The value of current drawn as a function of time can be seen below, the reading was taken using digital multimeter, it was used in the ammeter mode. The current drawn was directly measured from the battery to the device.

ADC refers to DC current and DCI refers to the mode of the multimeter. The multimeter has recorded in the time samples of 15ms.

S. NO	MODE	VALUE (mA)		Time (s)
1	DCI	19.2	ADC	0.015
2	DCI	19.99	ADC	0.03
3	DCI	20.01	ADC	0.045
4	DCI	19.96	ADC	0.06
5	DCI	19.86	ADC	0.075
6	DCI	10.23	ADC	0.09
7	DCI	0.38	ADC	0.105
8	DCI	0.41	ADC	0.12
9	DCI	0.36	ADC	0.135
10	DCI	0.37	ADC	0.15
11	DCI	0.41	ADC	0.165
12	DCI	0.41	ADC	0.18
13	DCI	0.38	ADC	0.195
14	DCI	0.34	ADC	0.21
15	DCI	0.39	ADC	0.225
16	DCI	0.36	ADC	0.24
17	DCI	0.38	ADC	0.255
18	DCI	0.37	ADC	0.27
19	DCI	0.38	ADC	0.285
20	DCI	0.43	ADC	0.3
21	DCI	0.39	ADC	0.315
22	DCI	0.58	ADC	0.33
23	DCI	18.43	ADC	0.345
24	DCI	18.92	ADC	0.36
25	DCI	17.75	ADC	0.375
26	DCI	20.07	ADC	0.39
27	DCI	19.88	ADC	0.405
28	DCI	20.06	ADC	0.42
29	DCI	141.28	ADC	0.435
30	DCI	141.32	ADC	0.45
31	DCI	141.76	ADC	0.465
32	DCI	141.71	ADC	0.48
33	DCI	144.58	ADC	0.495
34	DCI	144.94	ADC	0.51
35	DCI	144.58	ADC	0.525
36	DCI	142.99	ADC	0.54
37	DCI	71.06	ADC	0.555
38	DCI	0.35	ADC	0.57
39	DCI	0.36	ADC	0.585
40	DCI	0.4	ADC	0.6
41	DCI	0.36	ADC	0.615
42	DCI	0.35	ADC	0.63
43	DCI	0.36	ADC	0.645
44	DCI	0.37	ADC	0.66
45	DCI	0.38	ADC	0.675

Figure 4.4: Current Consumption Readings

The peak current consumption is only for a specific time interval. We can also see that very less current consumption has taken place during the sleep phase of the device. The increase in current draw can be observed during the wakeup phase after the sleep. During this phase initialization of the system takes place. Then as interactions increase, the power consumption increases simultaneously and reaches the peak value.

This peak value is observed during the two way ranging process between the esp and atmega system. Then as the ranging process completes, the system goes to sleep again.

The current consumption during the operating time, is plotted to observe the different regions. The plot is shown below.

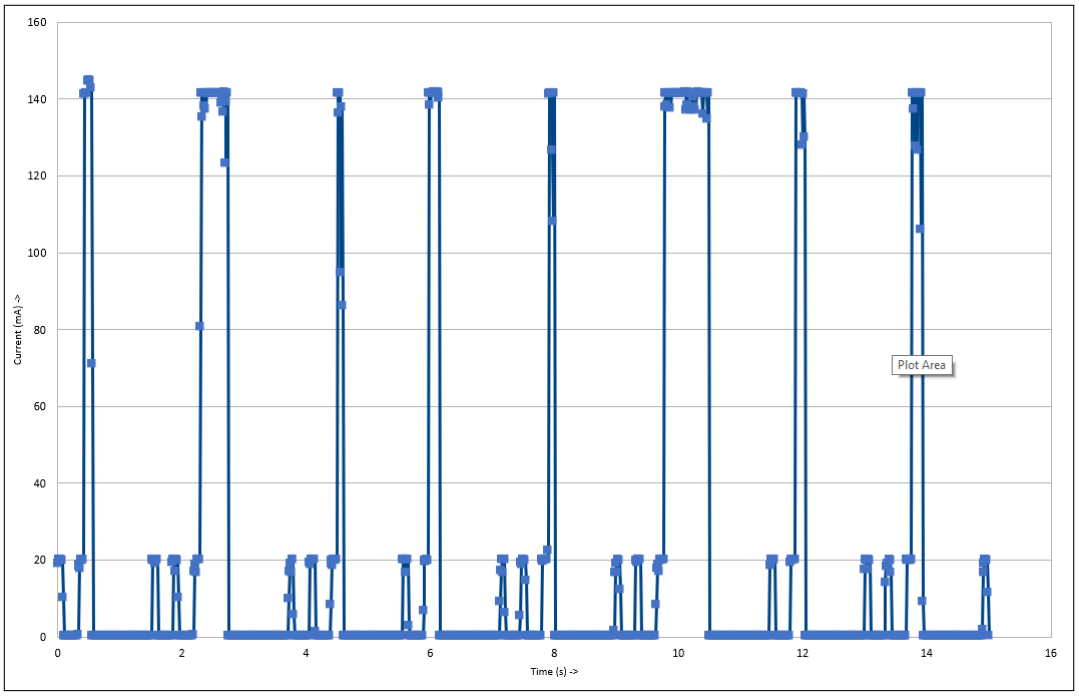

Figure 4.5: Sleep Mode Power Consumption Graph

The plot reaches peak and then falls down to micro amperes. Then it slowly increases and reaches peak once again. The cycle repeats throughout the operating time. It is to note that the peak current consumption only occurs if there is an esp system to interact with to perform the two way ranging. In case of absence of esp system, the atmega system consumes a maximum of 40 mA, instead of the peak value of 144 mA.

The power consumption is different when the atmega system reacts with different number of esp system during one cycle. The peak value is retained for more time when the system reacts with 4 esp systems during one cycle. The plots of the current consumption with varying esp system is shown below.

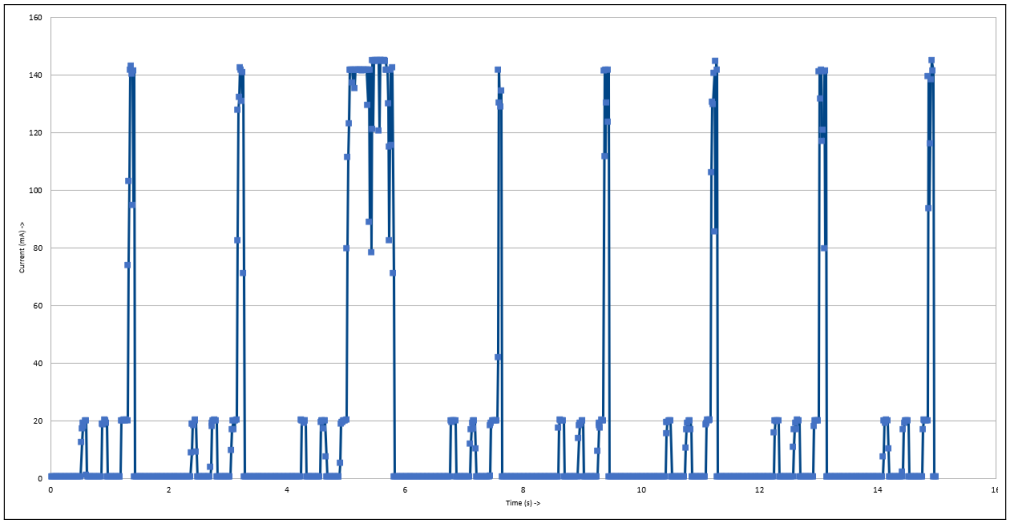

Figure 4.6: Sleep Mode Power Consumption Graph - 4 ESP systems

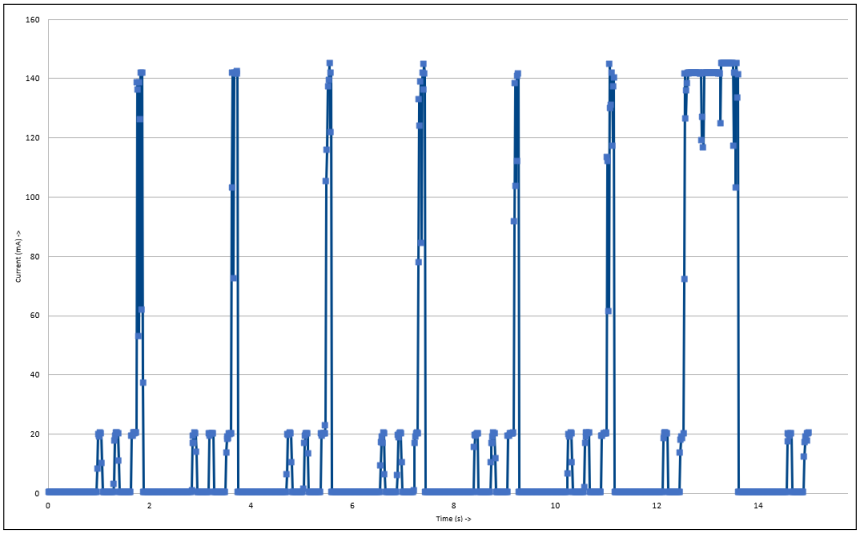

Figure 4.7: Sleep Mode Power Consumption Graph - 3 ESP systems

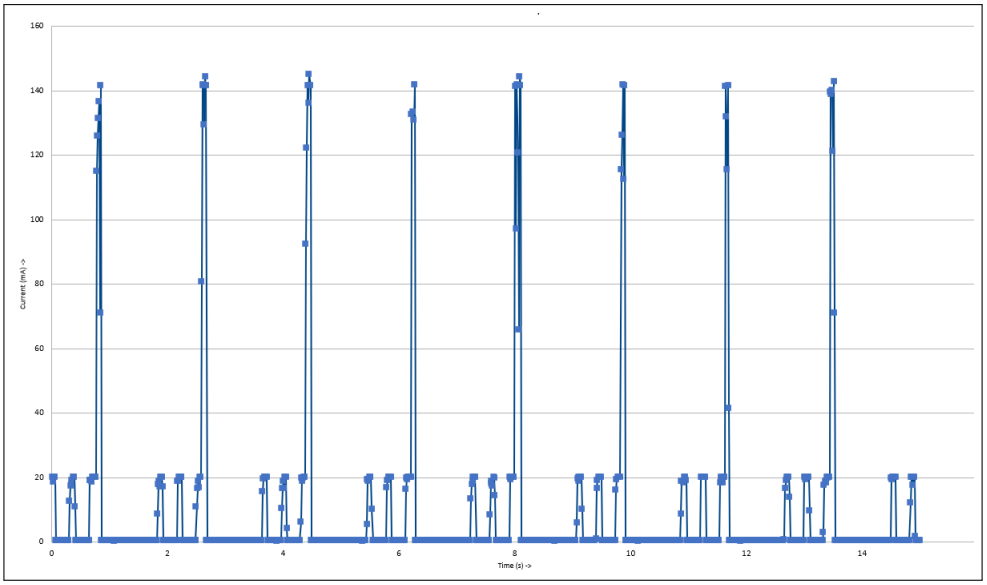

Figure 4.8: Sleep Mode Power Consumption Graph - 2 ESP systems

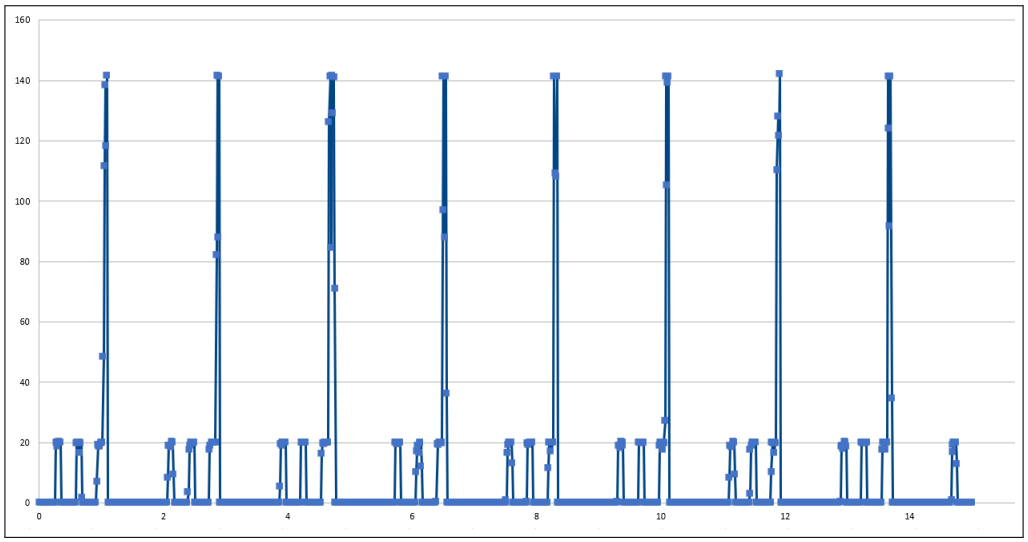

Figure 4.9: Sleep Mode Power Consumption Graph - 1 ESP system

The peak value of current consumption is retained for longer time when communicate with more number of systems. This is because the atmega system has to complete the TWR process with each esp system. In this process the UWB IC (antenna) is used continously to send and receive packets. This consumes lot of power and hence draws more current. As the TWR process with consecutive esp system takes time, it keeps drawing current untill the process is completed. After which it goes to sleep which can be observed in the drastic fall in the current consumption.

The device was tested for the sleep mode functioning using test script written using

python. In which we have logged the serial prints from the device to understand the status of the device. The script generates the log file and the log file is written so that the ranging process can be understood easily. The snippets from the log file is shown below.

The script generates extra text in the log file to show the starting and ending of ranging process. The script works on the serial prints from the device as input, it then makes decisions based on it and generates the log file accordingly. The USB port of the device (serial data input) must be mentioned in the script.

The device starts the ranging process, then gets acks from esp system. After which it completes the ranging process and goes to process complete sleep. Then the cycle repeats. It also prints battery voltage at regular time intervals. In case of no esp system in surrounding, the device goes reaches beacon timeout.

Thus the script runs continously and logs the data. In case of bug or unusual behaviour of the device, the log file can be looked into. The issue can be traced as the date and time are also logged.

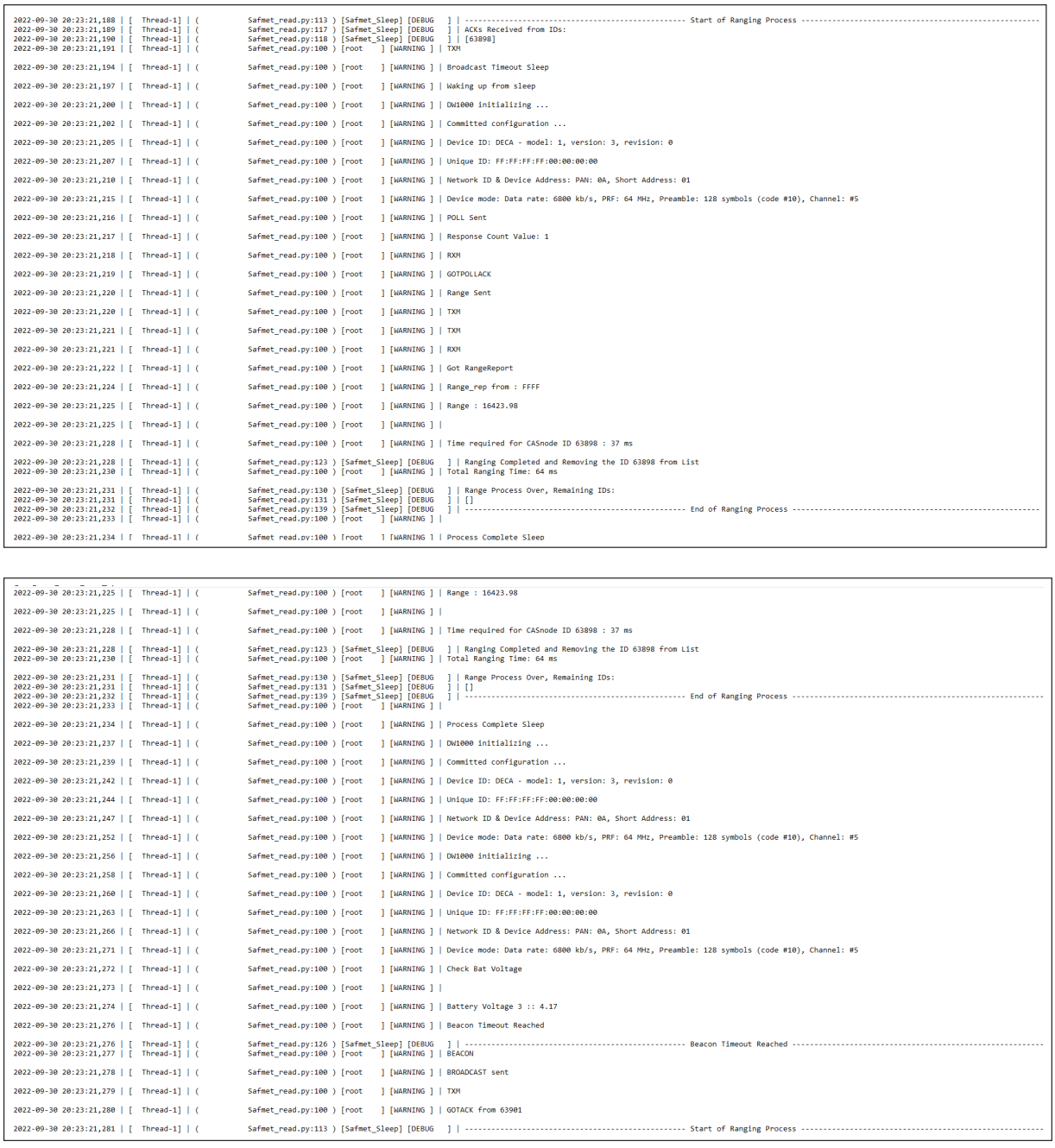

Figure 4.10: Sleep Mode Log Snippets

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

- The zones are maintained and once a worker comes inside near zone alert is given and as worker comes inside emergency zone breaks are applied.
- The sleep mode in the ICs reduced the power consumption. The average power consumption around the peak (0.475W, 144mA at 3.3V) was reduced.
- By this mechanism the ATtiny remains in sleep almost all the time except for small amount of time when it functions. Thus we can see the current consumption goes to 345 uA when the complete system is in sleep, after which initializations take place so current drawn increases to 20 mA. The peak value time is also greatly reduced, thus it draws maximum power for a very less time.

5.2 Future Scope

- Firmware changes can be made to improve the ranging process accuracy and response time of the system.
- The power consumption by the system can be reduced further by implementing improved sleep functions and precise sleep timings.

REFERENCES

[1] Sachin Kumar Yadav, Dr K K Sharma. Industrial Automation: Overview of the Internet of Things (IoT) (2017)

[2] Farouq Halawa, Husam Dauod, In Gyu Lee, Yinglei Li, Sang Won Yoon, Sung Hoon Chung. Introduction of a real time location system to enhance the warehouse safety and operational efficiency (2020)

[3] Thangavel Murugan, Abhijith V. S. and Sudersan. S. Industrial Automation Using Mobile Cyber Physical Systems (2022)

[4] Bello R S. Workplace Hazards Risks and Control (2012)

[5] Nienke Hofstra, Boyana Petkova, Wout Dullaert, Genserik Reniers, Sander De Leeuw. Assessing and facilitating warehouse safety (2018)

THE END