A Homeostatic Approach to Adaptive Ambient Control in Smart Factories

BY

CHONG SHAO YANG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONOURS) BUSINESS INFORMATION SYSTEMS

Faculty of Information and Communication Technology

(Kampar Campus)

JUNE 2024

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: A Homeostatic Approach to Adaptive Ambient Control in Smart Factories

____

____

Academic Session: ___ 202406 ___

I ____CHONG SHAO YANG______________________________________

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in

Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

_________________________ _________________________

(Author's signature) (Supervisor's signature)

Address:

__23, Lebuh Rasi Jaya 2______

__Taman Rasi Jaya, 31450____

__Menglembu, Perak_________

Aun Yichiet

Supervisor's name

Date: __12/09/2024_______

Date: 12/09/2024

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: __12/09/2024_____

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that Chong Shao Yang (ID No: 2005213) has completed this final year project entitled "A Homeostatic Approach to Adaptive Ambient Control in Smart Factories" under the supervision of Dr. Aun Yichiet (Supervisor) from the Department of Digital Economy Technology, Faculty of Information and Communication Technology from the Department of Digital Economy Technology, Faculty of Information and Communication Technology.

I understand that University will upload softcopy of my final year project in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

DECLARATION OF ORIGINALITY

I declare that this report entitled "**A Homeostatic Approach to Adaptive Ambient Control in Smart Factories**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ______________________________

Name : Chong Shao Yang

Date : 12/09/2024

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratefulness and appreciation to my supervisor, Dr Aun Yichiet, who presented a great opportunity to fully utilise my skills in designing and develop a full fledge IoT and AI system. It builds a solid foundation for my future career in the fields of IoT integration development. In addition, the knowledge shared from Dr Aun also allow me to improve not only in the technical scope, but also as an individual.

The second person I wanted to thanks is Corrine Heng, who supported me unconditionally during the hardship. It has been my main pillar of supports throughout the development of the project and my whole university period. Thank you for the support and love in all aspects.

Lastly, coming from a single parented family, I would like to thanks my mum, who is an independent woman. Thank you for supporting me whenever I faced problems. Throughout my studies, she has given me life advices which I would not receive anywhere but from her.

ABSTRACT

This project aims to develop a system using Artificial Intelligence (AI) and Internet of Things (IoT) to regulate and adjust environmental conditions in smart factories, focusing on the production of feed pellets, particularly during the drying stage. This paper addresses this gap by exploring existing technologies within the framework of Industrial Revolution 4.0. We've examined various factors affecting feed pellet quality by examining moisture level and climate setting using near-infrared (NIR), microwave, and capacitance sensors to detect ambient conditions. These data are visualized on a Grafana Dashboard for real-time data monitoring, a predictive model for dryer setup, and a computer vision system for quality control assessment. Testing will be conducted in a simulated environment to achieve a minimal working product (MVP). Lastly, the project has successfully implemented both a LSTM predictive modelling and a Siamese Network with CNN as base model for computer vision task. The LSTM predictive model is used to identify the optimal ambient setting to produce the highest quality pellet based on parametric optimisation. It is then integrated with the machine to manipulate and scale the environment conditions to best fit the pellet requirements. With the implemented Siamese Network with CNN has also successfully classify the pellets quality based on the appearance and colour of the given pellet.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

4IR	Industrial Revolution 4.0
ADC	Analog-to-Digital Converter
AI	Artificial Intelligence
CNN	Convolutional Neural Network
CV	Computer Vision
IDE	Integrated Development Environment
IoT	Internet of Things
LSTM	Long Short-Term Memory
ML	Machine Learning
RNN	Recurrent Neural Network

Chapter 1

Introduction

As technology continue to evolve, technologies that were once in science fiction are realised. Artificial intelligence (AI), Big Data, Internet of Things (IoT), robotics, and blockchain signalled a profound shift in the industry. With the emergence of technology and data, the industry sees this as an opportunity to further grow and enhance business through utilising both factors. Thus, introducing Industrial Revolution 4.0 (4IR) by incorporating the technologies. 4IR enabled smart production and manufacturing processes in Malaysia. Through government and private sectors effort [1], [2], industries are slowly transitioning to 4IR incorporating the available technologies and fostering sustainable growth in all aspect.

In animal feed production, 4IR came in as a game changer for the industry. Through the adoption of 4IR, feed production could be further enhanced with the use of IoT to ensure the production meets the topmost standard as an outcome. Based on research, a high-accuracy and precision procedure is applied at every stage in the process [3], [4]. Thus, to produce quality feed pellets, IoT can be introduced to accurately capture the real-time data from each machine, monitoring the process and outcome. There has been works done in this particular sector, as the industries realised the problem and started to adopt and bring in IR4, in transitioning to the smart factory [5]. Through the integration of IoT, data collected by sensors within the IoT network can be further utilised by implementing AI / ML model to create a predictive analysis model. Predictive models provide advantages that drive businesses to grow prosperously not only in terms of increasing efficiencies and effectiveness but also more predictable.

Figure 1.1 shows the illustration of feed pellet manufacturing process. IR4 can be applied in each stage to ensure the quality and enhance efficiency and effectiveness. Starting with the first stage in processing the raw materials, from batching to the end of the production line can be installed with sensors and integrate with AI / ML model in predicting and controlling the quality of the feed pellet with each serve a different purpose to achieve the quality to be assured at the very end.

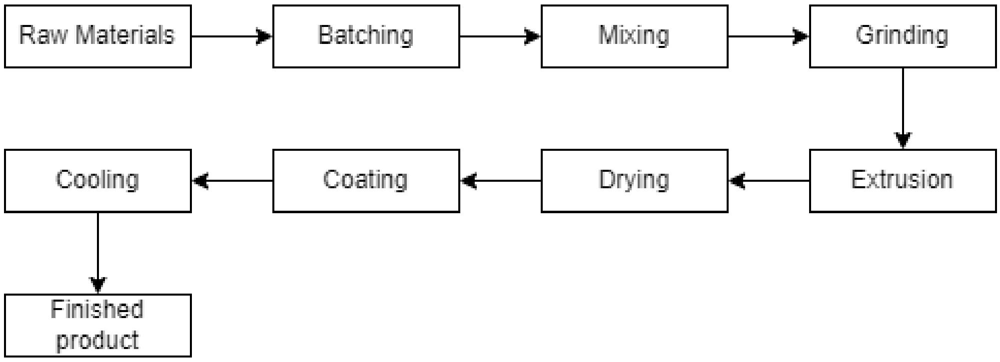


Figure 1.1 Flowchart of feed production process.

1.1 Problem Statement and Motivation

Despite the production and manufacturing process of animal feed pellets being handled carefully even in manual processes, **the dependencies on calculation and analysis with manpower in adjusting the dryer settings would not necessarily achieve the final product specifications**. Corns are the main source used in animal feed pellet production, there are various parameters that need to be taken care of, such as the moisture within the pellet after extrusion, the pellet size, surrounding temperatures, temperature of the dryer, and the corn kernel hardness [3], [4], [6]. In addition to the current methods of manually performing calculations and analysis, **human errors are inevitable. Hence causing over-drying or under-drying when adjusting the dryer setting.** During the evaluation, the particle size and the pellet moisture will be the significant factors and are prone to cause **overlooking in configuring the dryer settings**. Moreover, the data collection takes a lot of time, in return, the extruded pellets are set aside during the process awaiting the completion of analysis. As a matter of fact, the condition of the pellet could change during the waiting time. Thus, causing an unsatisfaction outcome. Nonetheless, **performing evaluation and configuration in the manual way is not feasible in the context of 4IR.** To further elaborate, the time consumption in each process waiting for manual sample testing, and the time consumed to analyse and configure the dryer are wasted. The efficiency and effectiveness are not predictable in turns, it is affecting the quality of the outcome. Each factor that affects the **quality of the pellet must be ensured** [4], otherwise the business would **bear the cost of low-quality feed pellets**.

REFERENCES

1.2 Objectives

This project proposes the implementation of computer vision technology for the meticulous monitoring of feed quality attributes, including particle size, colour consistency, and moisture levels. Concurrently, the utilization of IoT sensors aims to capture ambient data, facilitating real-time adjustments of machine parameters. The overarching objective is to optimize the feed production process systematically, ensuring a consistent and high-quality output; specifically,

(1) To integrate IoT sensors like moisture, humidity and temperature for ambient control and presented in dashboard with Grafana [7].

(2) .

   a. Prepare a prediction model for the dryer
   b. A front-end interface for controlling and predict the output of the input data

(3) To develop a computer vision (CV) model for Quality Control (QC) and monitoring based on colour and appearance.

During the extrusion process, the extruder forces the mixtures of different materials into a single substance with two mechanisms which are pressure and screw barrel rotation [8]. Based on Figure 1.1, some of the extruded feed pellets will then be taken as a sample set to be evaluated. As mentioned, the traditional way of manual value measuring, and configuration does not necessarily produce the optimum final product. Thus, the first objective is to create a streamlined process for the workers through the use of IoT sensors to capture moisture content of the extruded product, and the ambient humidity and temperature. These values would then be presented in a visually understandable way through Grafana.

Given that all the data has been collected into the database, as there would be no point leaving all the data to not be further processed into more valuable information. This project aimed to create a predictive model to further streamline the process of drying extruded pellets through minimising possible human error by automating the configuration of the dryer.

Continuing on streamlining the process, the end results from the dryer will then be constantly monitored with a CV model for QC control. The dried pellet should always be within a threshold of a certain predefined criteria. Based on the appearance and the colour, the pellets obtained could tell differences between a normal and a defect pellet. Taking in example such as pellets expanded from the inside causing a crack during drying process due to high internal pressure. Some other defects such as discolouration, inclusions of foreign object, and deformation would also be monitored as when the dried pellets are retrieved from the dryer.

1.3 Project Scope and Direction

In this project, research and development will only be based on the drying process of animal feed pellet production process based on Figure 1.1. As in the current stage of study, external factors will also be considered such as parameters available from the outcome of the extrusion, and the environmental factor which may or may not be affecting the feed pellet quality. In addition to the predictive model mentioned in the sub-section *Project Objective*, relying 100% of the efforts in configuring the process does not necessarily accurate and precise. Thus, the project should at least provide a suggested configuration setting for the dryer in terms of temperature supposed to be the optimum configuration.

At the end of the project, a combination of hardware and software would be delivered. Hardware includes the sensors used in capturing the outcome of the extrusion of the animal pellets, and a server that used to configure and streamline the process of the captured data. Software as in the predictive model and CV model, as well as the Grafana dashboard to present the data in visually pleasing way. All and all it would be a prototype in streamlining the production of animal feed pellets.

1.4 Project Impact, Significance, and Contributions

The research in the project confirms the feasibility in achieving smart factories aligned with the context of 4IR. Firstly, the sensors available in the current market is mature and powerful to gather information that will be needed in predicting the required outcome based on the quality of animal feed pellet. Secondly, by utilising the data collected from the sensors, a reliable predictive model can be produced to overcome issues such as over-drying or under-drying in the drying process based on Figure 1.1. Thirdly, the predicted outcome could also be further utilised in automating the process from extrusion to drying, eliminating human errors and shorten the time consumed in configurating the machine.

1.5 Report Organisation

In this project, we will be mainly focusing on the drying process of the feed pellets. We will also look into the sensors that are currently used in the industry for related works in Chapter 2. In Chapter 3, we will discuss on the system methodology. In Chapter 4, we will be discussing on the system design. Then in Chapter 5 will discussing on the system implementation of the project. In Chapter 6 will be the evaluation and testing on the implemented system. Lastly at Chapter 7 will be the concluding the project as a whole.

Chapter 2

Literature Review

2.1 Microwave Moisture Sensor

In this section, we will be reviewing the Microwave Moisture Sensor. Based on the patented design [9], the working principle of the sensor is through the microwave transmission and reflection when subject is passing through the sensor. When the microwave is emitted from the transmitter, a receiver will be setup in the other end to detect the return microwave as shown in Figure 2.1.

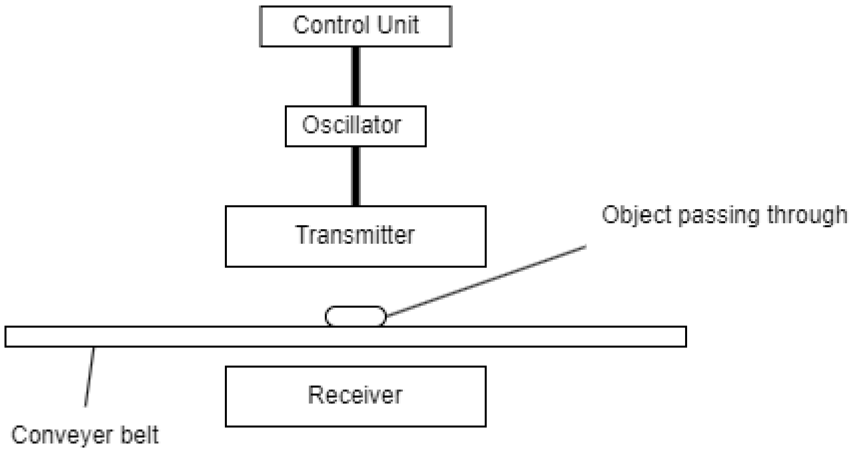


Figure 2.1 Extracted microwave moisture sensor architecture.

The moisture content will then be calculated based on the shared microwave oscillator to calculate the moisture content based on dielectric constant [9]-[11] the general formula is as shown as Equation 2.1.

$$\mathrm{MC} = \left(\frac{\varepsilon_r - \varepsilon_{\{r,d\}}}{\varepsilon_{\{r,s\}} - \varepsilon_{\{r,d\}}}\right) \times 100$$

Where, MC is the moisture content, ε_r is the measured dielectric constant of the material, $\varepsilon_{r,d}$ is the dielectric constant of the material in the dry state, and $\varepsilon_{r,s}$ is te dielectric constant of the material in saturated state.

With technological advancement, much smaller and accurate microwave moisture sensor has been developed for industrial purpose such as Hydro-Probe XT [12]. In the sector of pellet production, pellet moisture is the primary factor in achieving the targeted quality [3], [4]. The drying process need to adopt the variability of pellet moisture to achieve the optimum result [4, 13]. The sensor too has its advantages based on the review [14], microwave sensors are fast as the frequencies of the waves can be configured as per the application's needs. Thus, through the continuous emission of microwave allow the computation to be at least precise enough to automate the dryer to adopt and change it drying temperature.

2.2 Near Infrared (NIR) Moisture Sensor

In this section, we will be reviewing NIR moisture sensor and its application on the current pellet production process. In the spectrum of detecting moisture content, NIR came in handy to resolve the issue [15]. Figure 2.2 shows the system diagram of current NIR moisture sensors in the market based on Sensortech Systems' NIR-6000 Series On-Line Moisture Sensors design [16].

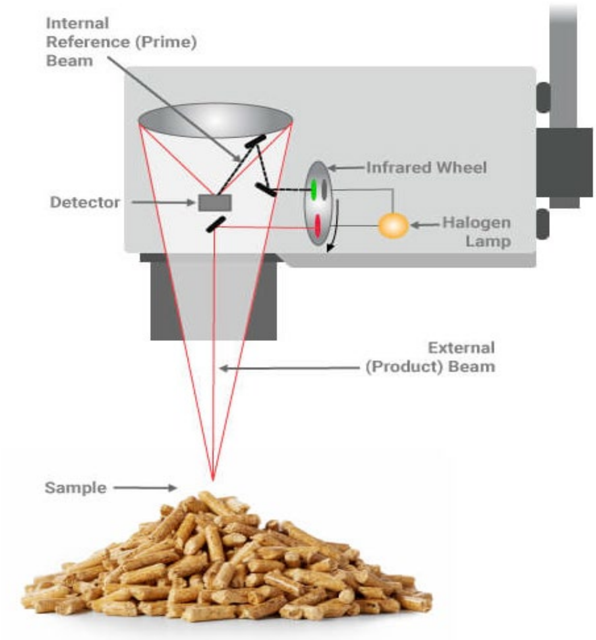


Figure 2.2 System design based on NIR-6000 Series On-Line Moisture Sensor [16]

Based on Figure 2.2, the NIR rays is being emitted from the halogen lamp to the sample. When the rays hit onto the sample, it will then interact with the water molecules in the sample. Some of the NIR rays will be refracted back to the machine itself into the detector. The moisture content will then be calculated based on the internal reference beam to obtain the moisture content percentage. The result obtained is based on the Beer Lambert expression [15] as shown in Equation 2.1.

$$log \frac{l_o}{l_t} = k\_t$$

Where l_o is the incident energy generated, I_t is the transmitted energy or the intensity of the radiation emerged, and k is the absorption coefficient [15].

Through the adoption of NIR statoscope in the industry and the advancement in the technology itself, it has gained its position at one of the creditable sensors to be integrated due to its high accuracy and it also can be used in rapid detection of flowing samples in conveyer belts or through portable NIR statoscope device [15, 17].

2.3 Capacitance Moisture Sensor

In this section, we will be reviewing on the capacitance moisture sensor which has been originally invented for capturing soil moisture content. Based on the patent [18], the working principle of the capacitance moisture sensor revolve in measuring the capacitance and the resistance to calculate the dielectric constant, and it can then be further calculated to obtain moisture content. As mentioned before, capacitance moisture sensor main purpose is to measure soil moisture content. In the context of agriculture, the sensor detects the water molecules in the soil. As for the detection, the sensor will release an electrical field to its surrounding. The receiver on the sensor will detect the returning field and measures the moisture content based on the reduction in capacitance due to the resistance within water molecules

Chapter 3

System Methodology

3.1 System Architecture Diagram

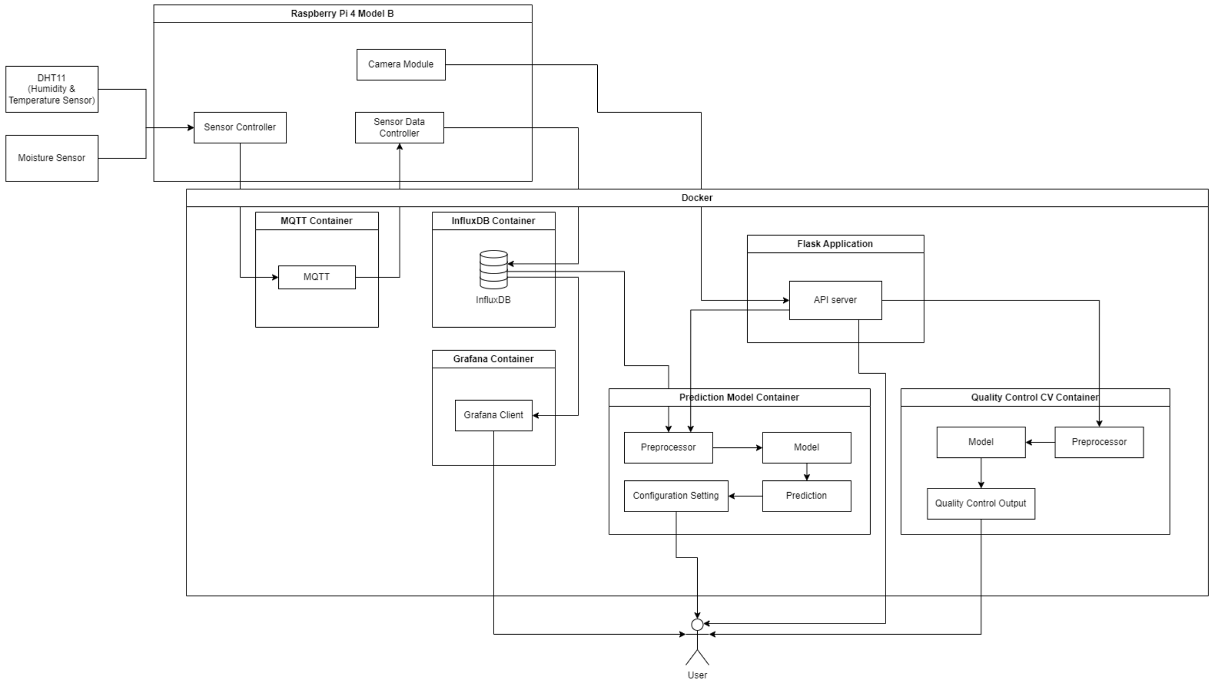


Figure 3.1 System Architecture Diagram

The figure above defines the overall architecture of the proposed system. As mentioned in earlier section, the application will be hosted in docker to standardize the environment. Each of the major components are separated into different container that hosted as an application. In the current proposed system architecture, there will be five major components hosted in the docker, which are MQTT broker, InfluxDB, Grafana, Flask API server, Predictive Model application, and the Quality Control CV application. Each container will perform only their specified task without interfering others application hosted within the same docker application. The Raspberry Pi in the other hand performs three different applications, which are the sensors controller, sensors data controller, and the operation of the camera module The Flask API server hosted in the primary machine is to accept the images received from the Raspberry Pi's camera module. At the end, the user will receive three different application which the first it the configured dashboard to show the data in visual graph dashboard, second is the dryer configuration based on the prediction model, the third is the quality control of the dryer output, and a demo application for visualizing the output from the model.

3.2 Machine Learning Lifecycle

To ensure the successful development of the machine learning model, a standard procedure has to be in place. Modelling a machine learning model can be complex and repetitive, without a clear and structured procedure, the process can be become cumbersome. In addition, without a clear framework, it would also be easy to overlook critical steps that can greatly affect the performance of the model. One of the most widely accepted machine learning development methodology is Cross Industry Standard Process for Data Mining (CRISP-DM), originally designed for data mining task; but due to it robust and flexible nature can be further adopted into machine learning modelling workflow.

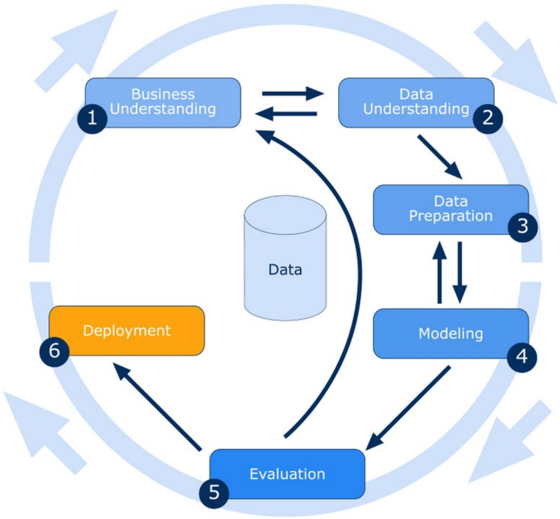


Figure 3.2 Illustration of Cross Industry Standard Process for Data Mining

Figure above illustrate the lifecycle process of CRISP-DM, the process starts from business understanding, data understanding, data preparation, modelling, evaluation, to deployment. In business understanding, the core process and objective of the task has to be determined before starting the process as a whole. With the business understanding in hand, the objective can be further defined and elaborated. This ensures the machine learning model aligns with the business objectives and act as a guidance to the subsequent phases of the CRISP-DM process.

Proceeding to data understanding, by understanding the data implies to the knowledge and information on the data itself. Aside from knowing the data type of the given dataset, its not sufficient to provide a sophisticated and comprehensive image. It has to be achieved in a way that the data's structure, distribution, and relationship has to be uncovered to reveal the meaningful pattern and potential issues that may arise during modelling the machine learning algorithm. In this phase, descriptive statistics, data visualisation, and identifying outliers and missing value has to be done to thoroughly understand the data.

Then, in data preparation stage, the focus shift to transforming and preparing the dataset to fit into the model in modelling stage. This stage involves handling the uncovered missing values, outliers, correcting inconsistency, normalise dataset, and feature selection. This stage is important as it will directly affect the models' performance, as poorly prepared data such as imbalance data and high number of outliers, might lead to high inaccuracy and bias.

In modelling stage, it will be focus on the machine learning modelling and. The usual practice in this stage is to try out on various models and algorithm to find the most suitable model for the given use case. Different algorithms and model may yield a similar or vastly different results in term of model's accuracy especially in supervised learning. Though this process defines the success criteria of the final model, but it needs to work tandem with data preparation as an iterative process to find out the most suitable model and algorithm by masking or transposing to their required input. This process ensures the selected model is provided with the most suitable data in goal to deliver the most optimal performance.

With the tested models, model evaluation has to be performed to evaluate the performance of each model. This process aimed to find the most suitable model for the given use case. In this process, with all the model, the usual practice is to select two that may seem to have the highest accuracy and align with the project objective. Then, hyperparameter tuning may be performed on each model to further optimize the selected model, so that the accuracy of the model can be slightly higher to ensure more reliable prediction.

At the end, with the completed model, deployment will be made to fulfil the business needs. It will be integrated into the production environment, and provide real-time or batch prediction depending on the business requirement.

3.3 Algorithm of ADC Convertion

Before going into the algorithm itself, it is important to go back to the basic of the capacitors, and dielectric by using the formula:

$$C = QV$$

Where *C* is the capacitance, *Q* is the charge, and *V* is the voltage. The charge refers to the amount of electrical energy (measured in Coulombs) stored within a capacitor, while the voltage refers to the electrical potential energy, or the energy needed to push the electron out of it place. Thus, the formula is to measure the amount of charge to store in capacitor per voltage.

Given that the capacitance has an inversely proportional with voltage, when the capacitance of the moisture sensor increases, the voltage would also be decreasing. With this in mind, the calibration of the sensors has been done through measuring the maximum voltage by holding the sensor in air, and the minimum voltage by submerging the sensor in the water. The maximum operating voltage returned from the reading is 2.5V while the minimum operating voltage is 1.0V. To calculate the percentage of the object moisture, the mathematical equation shows the steps of computation through utilising linear interpolation between the maximum and minimum allowed moisture level based on the voltage in the circuit:

Let,

MaxV as maximum operating voltage = 2.5

MinV as minimum operating voltage = 1.0

MaxM as maximum moisture = 100

MinM as minimum moisture = 0

V as the input voltage from V_{DD}

$$Hypothetical\ Moisture\ Content = MaxM - \frac{(V - MinV)}{MaxV - MinV} \times (MaxM - MinM)$$

3.4 Project Timeline

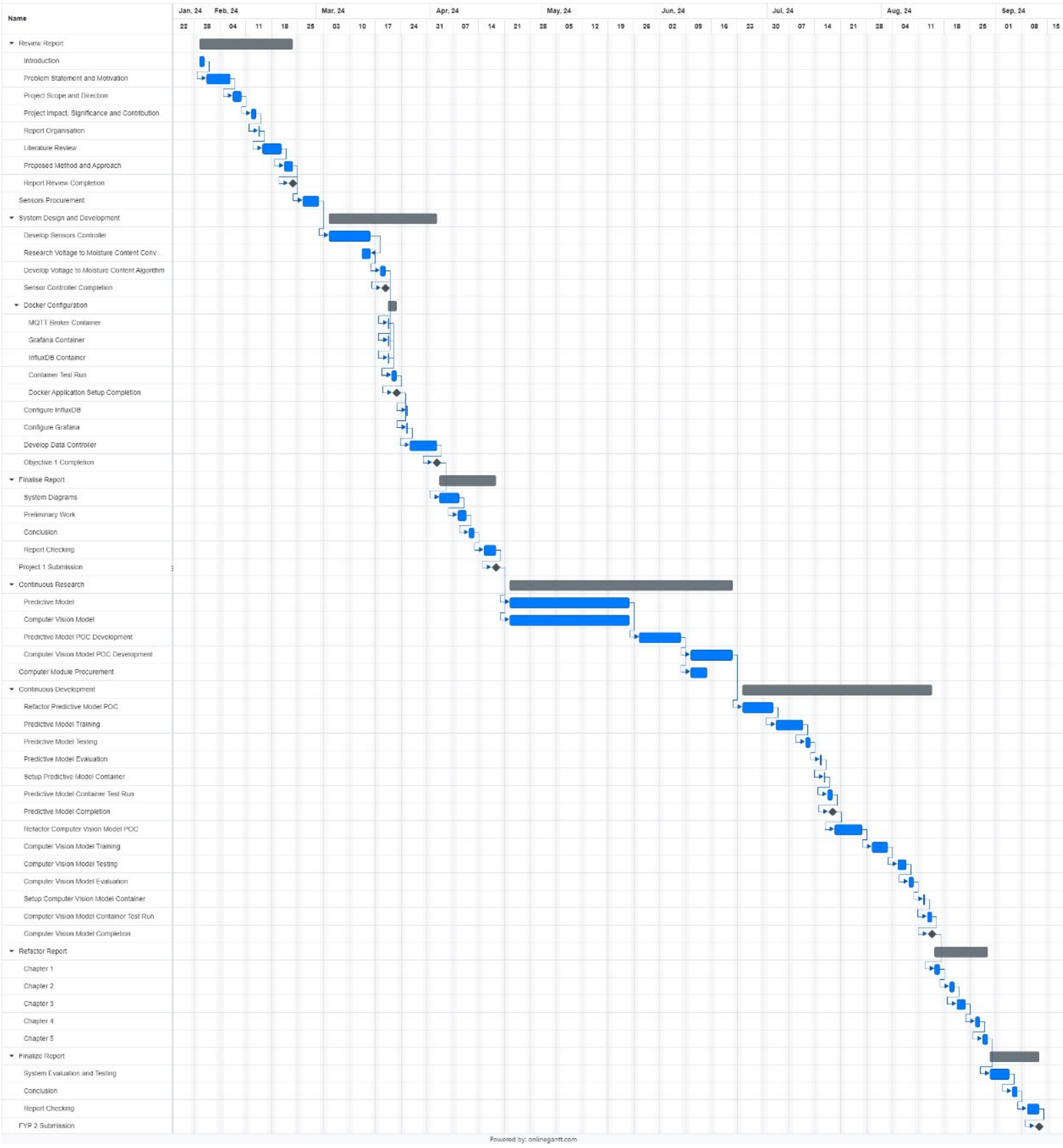


Figure 3.3 Gantt Chart of the project timeline

For clearer image, please refer to Appendix A

Chapter 4

System Design

4.1 Circuit Diagram

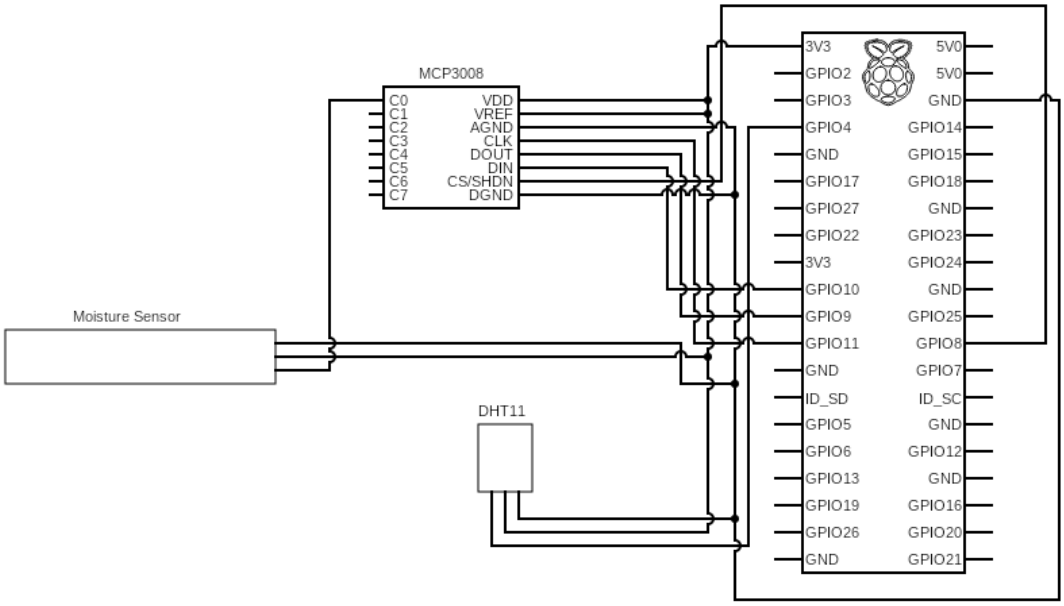


Figure 4.1 Circuit diagram of the sensors and Raspberry Pi

Figure above shows the connectivity between the sensors and the Raspberry Pi. A parallel circuit connecting between the capacitive moisture sensor, the temperature and humidity sensors (DHT11), and the Analog-to-Digital Converter (ADC) microcontroller MCP3008. The sensors and microcontroller are powered by a single 3V3 pin from the Raspberry Pi.

Based on the Raspberry Pi Pinout documentation, although in the 3V3 output supports up to 500mA on paper, but it would still be suggested to follow the 50mA limitation [19]. Thus, the components' operating current in the circuit has to be at most 50mA. To be more optimistic, the proposed circuit left some room to buffer for any unexpected condition.

Table 4.1 Maximum operating current of each component

Component	Max. Operating Current (mA)
Capacitive Moisture Sensor	5
MCP3008	0.002
DHT11	0.3
Total	5.302

From Table 4.1, the maximum operating current of each component are retrieved from the Appendix B, C, and D. The total maximum operating current is 5.302mA which is lower than the limitation from the 3V3 pin. Thus, using one pin is sufficient to power up whole circuit without any issue.

4.2 Hardwares Used in The Project

The hardware that will be involved in this project is a laptop for development purpose, Raspberry Pi. The laptop will be the main device to configure the Grafana interface, and host major component, such as MQTT broker, Grafana, InfluxDB, ML application, and CV application. While the Raspberry Pi will be the medium to connect the sensors, perform MQTT connection between itself and the MQTT broker hosted in docker, and also a MQTT client to receive and stores them into database.

Table 4.2 Specifications of development laptop

Description	Specifications
Model	Lenovo Legion Slim 5
Processor	AMD Ryzen 7 7840HS
Operating System	Windows 11 with Ubuntu Subsystem
Graphic	NVIDIA GeForce RTX 4060
Memory	16GB DDR5 5600MHz
Storage	512GB NVME SSD + 1TB SDD

Table 4.3 Specification of Raspberry Pi

Description	Specification
Model	Raspberry Pi 4 Model B
Processor	Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.8GHz
RAM	2GB
Wireless LAN and Bluetooth	2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
Ethernet	Gigabit Ethernet
GPIO Pins	40-pin extended GPIO
USB Ports	2 USB 3.0 ports; 2 USB 2.0 ports.
Audio Output	4 Pole stereo output and composite video port
Video Output	2 × micro-HDMI® ports (up to 4kp60 supported)
Camera Port	2-lane MIPI CSI camera port
Display Port	2-lane MIPI DSI display port
Storage	Micro SD port for loading the operating system and storing data
Power Source	Micro USB power source up to 2.5A

4.3 Software Used in The Project

Software such as Visual Studio Code, PyCharm, Postman, Eclipse Mosquitto, InfluxDB, and Docker will be chosen to achieve the proposed solution. Below will list out all the usage and benefits of each proposed software to be used.

Visual Studio Code [20]:

- A popular code editor from Microsoft.
- Benefits:
  - It has extensive language support, which allows developer to code in whatever language that the requirement needed.
  - It also has built in GitHub [21] source control to enable easier and faster version control within the editor.
  - It has extensive collection of extensions to support developers in completing tasks.

PyCharm [22]:

- A popular integrated development environment (IDE) for Python [23].
- Benefits:
  - It has built in debugging and profiling tools which enhance developers' experience in debugging.
  - It has built in version control such as Git, Mercurial and Subversion.
  - It allows developers to set up a virtual environment to isolate the dependencies of each project.

Postman [24]:

- A popular API development and testing tools
- Benefits:
  - It supports various methods in building HTTP request.
  - It has various pre-built integration that support third-party services such as GraphQL [25].

Eclipse Mosquitto [26]:

- A MQ Telemetry Transport (MQTT) protocol broker
- Benefits:
    - It is lightweight and efficient, suitable for environment with limited resource.
    - It also utilises binary encoding for serializing messages which mitigate the need for sensors data to be decoded into utf-8.

InfluxDB [27]:

- A popular open-source time-series database
- Benefits:
    - It provides time-based storage which will be suitable for IoT application.
    - It is highly scalable suitable for large volumes of data.

Docker [28]

- A tool to separate developed software from current infrastructure.
- Benefits:
    - It creates an isolated environment for the application to run independently.
    - It standardises the developed application such that scenario where "It runs on my machine" does not happen.

Chapter 5

System Implementation

In this chapter, the system implementation will be discussed starting from the implementation of the sensors with the Raspberry Pi, connecting it to the Docker container running on the main frame machine, and the inside working of the Docker container application that holds the prediction model, and computer vision model.

5.1 Hardware Setup

Figure 5.1 Hardware setup

As illustrated in the figure above, the Raspberry Pi is connected to two sensors which are the capacitance moisture sensors, and temperature and humidity temperature, as well as a Raspberry Pi camera module expansion. Due to the limited numbers of GPIO pins available on the device, but with its ability to steadily provide voltage output of 3V and 5.5V, the circuit is extended using a breadboard, connecting both aforementioned sensors and the MCP3008 analogue to digital converter chipset. Aside from the circuits connecting to the sensors, the camera modules extension is connected to the Raspberry Pi through the dedicated Camera Serial Interface (CSI) port.

5.2 Software Setup

5.2.1 Sensors Manager and Controller

Starting with Sensor Controller, as the name suggest, it handles the operation of the DHT11, and MCP3008. Both sensors have a Python library provided by Adafruit [29], [33], [31], while the moisture sensor did not have one publicly available. Thus, using the MCP3008 Python library provided by Adafruit, a conversion from the analogue to digital has been programmed to fit the supported input of Raspberry Pi. Using Object-Oriented Programming Practice (OOPP), both sensors are abstracted by a class each exposing the default behaviour and with extra functionalities scoped to the need of the project. The aim is to make the sensor class as generic as possible to ease the future development if something needed to be changed.

In addition, due to the processing power constraint, a thorough memory and processing management had been done to make the program runs more efficient. Since the both of the sensors needed to be run in an infinite loop at the same time, the sensors are programmed to be run in concurrency utilising Linux's concurrency support. Both sensors are managed by a manager which then initialises the concurrency control to create two software threads managing the instantiated sensor with same memory address.

In Sensor Data Controller, the program is also restricted by the limited resource. Thus, considering the controller needs to constantly listening to the MQTT broker for message, while writing the response value to InfluxDB, multiprocessing is used in handling the situation. Since the Python version used in the current project is 3.9.2, according to the documentation, the multiprocessing library uses fork by default in Unix environment [32]. By using the default behaviour, Python interpreter spawn a child process which utilising the same shared memory which is the instance of subscription to MQTT broker.

5.2.2 Docker Application Container Configuration

The Docker application was configured and deployed on the mainframe computer to act as a service server. Based on the figure 3.1 in Chapter 3, the backend of the system is hosted in a Docker application container, which allows for efficient management and scalability of each service. In within the Docker application itself, a MQTT server, InfluxDB, the animal feed pellet prediction model, the animal feed pellet quality control computer vision model, and a Flask server are hosted.

Each of the individual application has its own container isolating itself from the others, ensuring the integrity of each application. Thus, if some application in the container failed, it would not affect the others. Additionally with the provided configuration scripts, each container can be configured to automatically restart in case of failure, as part of the automation process, ensuring seamless operation and minimizing downtime of the service.

The composed application container was also configured with its own internal network to further secure the internal communication between each application in the container. It also allows the internal application to effectively communicates with each other through the internal network without any interference from the outside.

5.3 Predictive Model Implementation

Based on the implementation of machine learning development lifecycle mentioned earlier, CRISP-DM, in this section, process from business understanding to modelling. Each stage will be explored in details to provide the rationale behind the implementation and its alignment to the aforementioned project objectives.

5.3.1 Business Understanding

The initial business understanding phase is crucial as it ensures the machine learning solution aligns with the project objectives. In reference to the project objective outlined in Chapter 1.2, the goal is to develop a predictive model has to be developed based on batches of the produced pellets and a computer vision model for quality control and monitoring, specifically focusing on the colour and appearance of the pellets.

First off, it is crucial that to understand the underlying task for the project. There will be two models to be implemented, a time series predictive model, and a computer vision model. Both have to be implemented with the end goals in mind. The predictive model has to be effectively and accurate in predicting the possible outcome from the given input parameter, as it will be used to predict the possible output when certain configuration is in place. The second mode, computer vision will be used to determine if the output from the machinery meets the expectation and quality of the company in terms of their quality standards.

In practice, it affects the choice of algorithm, structure of data pipeline, and the deployment of the application. Take the instance of the predictive model, it has to be trained with a comprehensive and sophisticated dataset to meet the required output. Data such as the batch parameters and production environment conditions have to be carefully identified.

The tools that will be using for the machine learning implementation will be Python with Jupyter Notebook, and TensorFlow with Keras as our primary library in data exploratory to modelling.

5.3.2 Data Understanding

With data obtained through third-party supplier, the predictive model starts off without any further postpone. The dataset includes two product SKUs where each SKU has two set of data representing input and output from both different machineries. Take an instance from the dataset, there are two product SKU where the first is SKU-A, and the second is SKU-B. Thus, each of the product corresponds to two datasets where the first is Extruder vs Dryer data, and the second is the Cooler data.

```
import pandas as pd
from pathlib import Path

pd.set_option("display.max_columns", None)

SKU_A_Extruder_vs_Dryer = "SKU-A Extruder vs Dryer"
Cooler_SKU_A = "Cooler SKU-A"
SKU_B_Extruder_vs_Dryer = "SKU-B Extruder vs Dryer"
Cooler_SKU_B = "Cooler SKU-B"

excel_file = Path("data/Extruder vs Dryer 20240827.xlsx")

SKU_A_Extruder_vs_Dryer_data = pd.read_excel(excel_file, sheet_name=SKU_A_Extruder_vs_Dryer, index_col=0)
✓ 2.6s
```

Figure 5.2 Importing the dataset into Jupyter Notebook

First, data is imported into the Notebook with Pandas library into a Data Frame to be further process. By utilizing pandas, data can be further processed and understanded through various different data visualisation methods. These techniques help in identifying underlying patterns, trends, and potential anomalies in the data, which will be critical for continuous analysis and modelling.

```
SKU_A_Extruder_vs_Dryer_data.head()
```
✓ 0.0s Python

time	SKU	A_FeedRate	Feeder_Hz	SC_Water%	SC_Steam%	MainMotor_Hz	MainMotor_Amp	CutterSpeed	sc_WaterFlow	sc_SteamFlow	sc_Temperature	ddc_Temperature	barrel_inl
2024-08-15 11:07:37	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2024-08-15 11:08:07	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2024-08-15 11:08:37	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2024-08-15 11:09:07	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2024-08-15 11:09:37	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

Figure 5.3 Quick look into the first five rows of SKU-A extruder vs dryer of the dataset

```
SKU_A_Extruder_vs_Dryer_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 845 entries, 2024-08-15 11:07:37 to 2024-08-15 18:10:38
Data columns (total 42 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   SKU                       828 non-null    object 
 1   A_FeedRate                797 non-null    float64
 2   Feeder_Hz                 797 non-null    float64
 3   SC_Water%                 797 non-null    float64
 4   SC_Steam%                 797 non-null    float64
 5   MainMotor_Hz              797 non-null    float64
 6   MainMotor_Amp             797 non-null    float64
 7   CutterSpeed               797 non-null    float64
 8   sc_WaterFlow              797 non-null    float64
 9   sc_SteamFlow              797 non-null    float64
 10  sc_Temperature            797 non-null    float64
 11  ddc_Temperature           797 non-null    float64
 12  barrel_inletTemp          797 non-null    float64
 13  barrel_Zone2Temp          797 non-null    float64
 14  barrel_Zone3Temp          797 non-null    float64
 15  FinalHead_Temp            797 non-null    float64
 16  totalTonnage              797 non-null    float64
 17  total_Runtime             797 non-null    float64
 18  liveBin_Weight            797 non-null    float64
 19  DryerZone1_ActualTemp     841 non-null    float64
...
 40  Dryer_BottomRetention_Hz  841 non-null    float64
 41  Dryer_TotalRetention_min  841 non-null    float64
dtypes: float64(41), object(1)
memory usage: 283.9+ KB
```
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

Figure 5.4 Information on each column and their corresponding data type (SKU-A extruder vs dryer)

```python
SKU_A_Extruder_vs_Dryer_data.describe()
```

Python

	A_FeedRate	Feeder_Hz	SC_Water%	SC_Steam%	MainMotor_Hz	MainMotor_Amp	CutterSpeed	sc_WaterFlow	sc_SteamFlow	sc_Temperature	ddc_Temperature	barrel_inletTemp
count	797.000000	797.000000	797.000000	797.000000	797.000000	797.000000	797.00000	797.000000	797.000000	797.000000	797.000000	797.000000
mean	4609.503137	52.458469	19.107277	8.543287	44.203262	277.084065	25.93601	880.379009	394.160540	95.862422	99.310715	107.946048
std	592.022920	6.726548	0.348323	0.394301	5.156967	37.039015	2.75228	114.964596	49.974359	2.704252	7.607995	1.268422
min	0.000000	1.000000	19.000000	8.500000	10.000000	0.000000	1.00000	0.000000	0.000000	45.140000	33.550000	98.000000
25%	4699.000000	52.680000	19.000000	8.500000	45.000000	280.000000	26.00000	889.870000	399.110000	95.640000	100.120000	108.000000
50%	4700.000000	53.810000	19.000000	8.500000	45.000000	283.000000	26.00000	894.030000	399.490000	96.060000	100.350000	108.000000
75%	4701.000000	54.090000	19.000000	8.500000	45.000000	286.000000	26.00000	896.940000	399.860000	96.530000	100.720000	108.000000
max	4723.000000	55.290000	21.000000	13.000000	45.000000	302.000000	30.00000	948.440000	513.550000	99.590000	101.120000	110.000000

Figure 5.5 Statistical description of each column (SKU-A extruder vs dryer)

From figure 5.3 to 5.5 shows the information of dataset for product SKU-A. In figure 5.3 and 5.4 there exists many null values especially from the start of the dataset as seen in figure 5.3. Figure 5.4 also mentioned that there are a total of 845 entries where most of the column does not fill out the data. The same processes were made to dataset for product SKU-B as shown in figures below.

```python
SKU_B_Extruder_vs_Dryer_data.head()
```

✓ 0.0s Python

	SKU	A_FeedRate	Feeder_Hz	SC_Water%	SC_Steam%	MainMotor_Hz	MainMotor_Amp	CutterSpeed	sc_WaterFlow	sc_SteamFlow	sc_Temperature	ddc_Temperature	barrel_in
time													
2024-08-12 11:13:04	SKU-B	3110.0	31.37	19.0	13.0	10.0	0.0	33.0	0.00	442.57	32.01	31.53	
2024-08-12 11:13:34	SKU-B	3431.0	32.88	19.0	13.0	10.0	0.0	33.0	0.00	489.66	38.76	31.81	
2024-08-12 11:14:04	SKU-B	3562.0	34.21	19.0	13.0	10.0	0.0	33.0	0.00	506.77	57.60	32.14	
2024-08-12 11:14:34	SKU-B	3506.0	35.63	19.0	13.0	10.0	0.0	33.0	762.18	514.95	75.61	33.91	
2024-08-12 11:15:04	SKU-B	3726.0	36.67	19.0	13.0	10.0	0.0	33.0	759.43	517.44	85.83	38.87	

Figure 5.6 Quick look into the first five rows of SKU-B extruder vs dryer dataset

```
SKU_B_Extruder_vs_Dryer_data.info()
✓ 0.0s

<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1222 entries, 2024-08-12 11:13:04 to 2024-08-12 21:23:35
Data columns (total 42 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   SKU                      1222 non-null   object
 1   A_FeedRate               1220 non-null   float64
 2   Feeder_Hz                1220 non-null   float64
 3   SC_Water%                1220 non-null   float64
 4   SC_Steam%                1220 non-null   float64
 5   MainMotor_Hz             1220 non-null   float64
 6   MainMotor_Amp            1220 non-null   float64
 7   CutterSpeed              1220 non-null   float64
 8   sc_WaterFlow             1220 non-null   float64
 9   sc_SteamFlow             1220 non-null   float64
 10  sc_Temperature           1220 non-null   float64
 11  ddc_Temperature          1220 non-null   float64
 12  barrel_inletTemp         1220 non-null   float64
 13  barrel_Zone2Temp         1220 non-null   float64
 14  barrel_Zone3Temp         1220 non-null   float64
 15  FinalHead_Temp           1220 non-null   float64
 16  totalTonnage             1220 non-null   float64
 17  total_Runtime            1220 non-null   float64
 18  liveBin_Weight           1220 non-null   float64
 19  DryerZone1_ActualTemp    1214 non-null   float64
...
 40  Dryer_BottomRetention_Hz    1214 non-null   float64
 41  Dryer_TotalRetention_min    1214 non-null   float64
dtypes: float64(41), object(1)
memory usage: 410.5+ KB
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Figure 5.7 Information on each column and their corresponding data type (SKU-B extruder vs dryer)

```
SKU_B_Extruder_vs_Dryer_data.describe()
✓ 0.0s                                                     Python
```

	A_FeedRate	Feeder_Hz	SC_Water%	SC_Steam%	MainMotor_Hz	MainMotor_Amp	CutterSpeed	sc_WaterFlow	sc_SteamFlow	sc_Temperature	ddc_Temperature	barrel_inletTem
count	1220.000000	1220.000000	1220.000000	1220.000000	1220.000000	1220.000000	1220.000000	1220.000000	1220.000000	1220.000000	1220.000000	1220.00000
mean	4739.313115	53.492910	19.139836	9.624098	44.688525	290.728689	35.883607	906.110574	456.386049	100.007795	99.860967	107.87950
std	298.444205	3.798976	0.205391	0.279886	3.187038	28.254167	1.001013	71.554101	27.343111	3.032352	4.657721	4.26763
min	0.000000	1.000000	19.000000	9.000000	10.000000	0.000000	20.000000	0.000000	0.000000	32.010000	31.530000	96.00000
25%	4701.000000	52.580000	19.000000	9.500000	45.000000	288.000000	36.000000	899.087500	455.630000	100.040000	100.120000	109.00000
50%	4796.000000	53.690000	19.000000	9.700000	45.000000	292.000000	36.000000	910.170000	456.350000	100.230000	100.260000	109.00000
75%	4801.000000	54.960000	19.300000	9.700000	45.000000	299.000000	36.000000	917.012500	460.510000	100.380000	100.380000	110.00000
max	4900.000000	57.170000	19.500000	13.000000	45.000000	316.000000	36.000000	964.080000	521.330000	100.800000	100.940000	123.00000

Figure 5.8 Statistical description of each column (SKU-B extruder vs dryer)

As mentioned, figure 5.6 to 5.8 shows the information on product SKU-B in extruder vs dryer dataset. In figure 5.6 and 5.7, it also shows that the same situation as the product SKU-A there exists null values in the dataset, though it is not as much as the previous dataset.

```
SKU_B_Extruder_vs_Dryer_data.head(n=10)
```
✓ 0.0s

p	FinalHead_Temp	totalTonnage	total_Runtime	liveBin_Weight	DryerZone1_ActualTemp	DryerZone2_ActualTemp	DryerZone3_ActualTemp	DryerZone1_Setpoint	Dryer
.0	92.0	23.0	0.0	725.8	NaN	NaN	NaN	NaN	
.0	91.0	49.0	0.0	699.6	NaN	NaN	NaN	NaN	
.0	90.0	76.4	0.0	672.6	NaN	NaN	NaN	NaN	
.0	90.0	102.8	0.0	645.8	NaN	NaN	NaN	NaN	
.0	89.0	131.4	0.0	617.4	NaN	NaN	NaN	NaN	
.0	89.0	159.8	0.0	589.0	NaN	NaN	NaN	NaN	
.0	88.0	187.4	0.0	736.4	104.87	113.43	77.87	105.0	
.0	88.0	216.8	0.0	707.0	NaN	NaN	NaN	NaN	
.0	87.0	246.4	0.0	677.4	NaN	NaN	NaN	NaN	
.0	87.0	276.8	0.0	647.0	104.77	110.46	77.33	105.0	

Figure 5.9 The first 10 rows of SKU-B extruder vs dryer dataset

```
SKU_A_Extruder_vs_Dryer_data.head(n=10)
```
✓ 0.0s

time	SKU	A_FeedRate	Feeder_Hz	SC_Water%	SC_Steam%	MainMotor_Hz	MainMotor_Amp	CutterSpeed	sc_W
2024-08-15 11:07:37	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2024-08-15 11:08:07	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2024-08-15 11:08:37	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2024-08-15 11:09:07	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2024-08-15 11:09:37	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2024-08-15 11:10:07	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2024-08-15 11:10:37	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2024-08-15 11:11:07	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2024-08-15 11:11:37	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2024-08-15 11:12:07	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

Figure 5.10 The first 10 rows of SKU-A extruder vs dryer dataset

Taking a deeper look into the dataset itself, we can see that the there are a slight difference in between both data collected. In dataset of SKU-A have the null values starts from the extruder, while in SKU-B starts from the dryer. In the case of SKU-A it shows that the null values at the starts of the dataset indicates that the machine is at its wind-up time whereby when we look at the dryer, the temperature of the dryer started increasing slowly as the time passes. Dataset of SKU-B on the other hand shows otherwise, where based on the dryer zone actual temperature and the temperature setpoint indicates has only a slight difference, which indicates the there may be loss of data in the dataset.

```
SKU_A_Cooler_data.head()
✓ 0.0s
```

SKU	Sampling Date	Time	Adjusted Moisture	Length	Diameter	Density (g/L)	Hardness (kg/cm)	Buoyancy (%) (Sink/Float)
SKU-A	2024/08/15	12.20PM	0.0889	6.94	7.16	0.358	1.5	1
SKU-A	2024/08/15	12.40PM	0.0859	7.32	7.23	0.355	1.5	1
SKU-A	2024/08/15	1.00PM	0.0844	7.25	7.39	0.355	1.5	1
SKU-A	2024/08/15	1.20PM	0.0897	6.99	7.32	0.350	1.5	1
SKU-A	2024/08/15	1.40PM	0.0875	7.55	7.50	0.348	1.5	1

Figure 5.11 The first 5 rows of SKU-A cooler data

```
SKU_A_Cooler_data.info()
✓ 0.0s

<class 'pandas.core.frame.DataFrame'>
Index: 15 entries, SKU-A to SKU-A
Data columns (total 8 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Sampling Date              15 non-null     object
 1   Time                       15 non-null     object
 2   Adjusted Moisture          15 non-null     float64
 3   Length                     15 non-null     float64
 4   Diameter                   15 non-null     float64
 5   Density (g/L)              15 non-null     float64
 6   Hardness (kg/cm)           15 non-null     float64
 7   Buoyancy (%) (Sink/Float)  15 non-null     int64
dtypes: float64(5), int64(1), object(2)
memory usage: 1.1+ KB
```

Figure 5.12 Information on each column and their corresponding data type (SKU-A cooler data)

```
SKU_A_Cooler_data.describe()
✓ 0.0s
```

	Adjusted Moisture	Length	Diameter	Density (g/L)	Hardness (kg/cm)	Buoyancy (%) (Sink/Float)
count	15.000000	15.000000	15.000000	15.000000	15.000000	15.0
mean	0.088187	7.259333	7.382000	0.354400	1.480000	1.0
std	0.004176	0.220793	0.149867	0.006367	0.041404	0.0
min	0.081200	6.880000	7.160000	0.348000	1.400000	1.0
25%	0.085150	7.150000	7.275000	0.349000	1.500000	1.0
50%	0.088800	7.250000	7.350000	0.353000	1.500000	1.0
75%	0.091650	7.420000	7.525000	0.359000	1.500000	1.0
max	0.094700	7.580000	7.650000	0.365000	1.500000	1.0

Figure 5.13 Statistical description of the dataset (SKU-A cooler data)

From figure 5.11 to 5.13, it shows the output of the product at the cooler. Figure 5.# shows no null values across all 15 entries of the data. Though there is no null values, but there is another concern that has to be properly handled later on in data preparation stage. Based on the "Sampling Date" and "Time" columns, it shows that the data are collected periodically in a fixed interval in minutes. While in the extruder vs dryer dataset, the data are collected in fixed 30 seconds interval. Thus, making the cooler data set way too less to be appended into the main data set as it would be a problem later on.

```
SKU_B_Cooler_data.head()
✓ 0.0s
```

SKU	Sampling Date	Time	Adjusted Moisture	Length	Diameter	Density (g/L)	Hardness (kg/cm)	Buoyancy (%) (Sink/Float)
SKU-B	2024/08/12	12.40PM	0.0579	4.28	13.33	0.373	NaN	1
SKU-B	2024/08/12	1.00PM	0.0544	4.08	13.33	0.388	NaN	1
SKU-B	2024/08/12	1.20PM	0.0561	4.41	12.85	0.383	NaN	1
SKU-B	2024/08/12	1.40PM	0.0553	4.15	13.50	0.385	NaN	1
SKU-B	2024/08/12	2.00PM	0.0607	4.30	13.48	0.370	NaN	1

Figure 5.14 First five rows of SKU-B cooler data

```
SKU_B_Cooler_data.info()
✓ 0.0s
<class 'pandas.core.frame.DataFrame'>
Index: 23 entries, SKU-B to SKU-B
Data columns (total 8 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Sampling Date              23 non-null     object
 1   Time                       23 non-null     object
 2   Adjusted Moisture          23 non-null     float64
 3   Length                     23 non-null     float64
 4   Diameter                   23 non-null     float64
 5   Density (g/L)              23 non-null     float64
 6   Hardness (kg/cm)           0 non-null      float64
 7   Buoyancy (%) (Sink/Float)  23 non-null     int64
dtypes: float64(5), int64(1), object(2)
memory usage: 1.6+ KB
```

Figure 5.15 Information on each column and their corresponding data type (SKU-B cooler data)

```
SKU_B_Cooler_data.describe()
```
✓ 0.0s

	Adjusted Moisture	Length	Diameter	Density (g/L)	Hardness (kg/cm)	Buoyancy (%) (Sink/Float)
count	23.000000	23.000000	23.000000	23.000000	0.0	23.0
mean	0.065748	4.430000	13.296087	0.377609	NaN	1.0
std	0.006726	0.158745	0.219767	0.008675	NaN	0.0
min	0.054400	4.080000	12.850000	0.360000	NaN	1.0
25%	0.060000	4.305000	13.130000	0.373000	NaN	1.0
50%	0.065700	4.430000	13.250000	0.379000	NaN	1.0
75%	0.072200	4.550000	13.500000	0.384000	NaN	1.0
max	0.075900	4.650000	13.670000	0.388000	NaN	1.0

Figure 5.16 Statistical description of the dataset (SKU-B cooler data)

Figure 5.14 to 5.16, shows the output of the product at the cooler for SKU-B. Based on the output, there exists null values across the "Hardness (kg/cm)" column where it is all null values across all rows.

Due to the missing labels in the labelled data, it is difficult for the time series prediction model to perform well as such model are intended for supervised learning as oppose to semi-supervised learning. Thus, the remaining will be covering only for product SKU-A.

Before digging deeper into the data, a preprocess has to be done on the labelled data which is the cooler dataset where the sampling data and time has to be compiled as one and turns into the index of the dataset. To rationale the method used in this situation were based on the extruder vs dryer for product SKU-A is a timeseries dataset and the project aimed to produce a time series predictive model

```
import re

SKU_A_Cooler_data["time"] = SKU_A_Cooler_data["Sampling Date"] + " " + SKU_A_Cooler_data["Time"].apply(lambda x: re.sub(r"(\d)\.(\d\d)(AM|PM)", r"\1:\2\3", x))
SKU_A_Cooler_data["time"] = pd.to_datetime(SKU_A_Cooler_data["time"])

SKU_A_Cooler_data.reset_index()
SKU_A_Cooler_data.set_index("time", inplace=True)

SKU_A_Cooler_data.drop(columns=["Sampling Date", "Time", "SKU"], inplace=True)
```
✓ 0.0s

Figure 5.17 Joining the "Sampling Date" and "Time"

Figure 5.17 shows the feature combination of both "Sampling Date" and "Time" into "time" column as a timestamp and reindexed as a timeseries dataset. Then with the reindexed SKU-A cooler dataset, it was then concatenate with the extruder vs dryer for product SKU-A with backward fill method as shown in figure below.

```
reindexed_SKU_A_Cooler_data = SKU_A_Cooler_data.reindex(SKU_A_Extruder_vs_Dryer_data.index, method="bfill")
labelled_SKU_A_Extruder_vs_Dryer_data = pd.concat([SKU_A_Extruder_vs_Dryer_data, reindexed_SKU_A_Cooler_data], axis=1)
labelled_SKU_A_Extruder_vs_Dryer_data.head()
```

✓ 0.0s

	SKU	A_FeedRate	Feeder_Hz	SC_Water%	SC_Steam%	MainMotor_Hz	MainMotor_Amp	CutterSpeed	sc_WaterFlow	sc_SteamFlow	sc_Temperature	ddc_Temper
time												
2024-08-15 11:07:37	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2024-08-15 11:08:07	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2024-08-15 11:08:37	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2024-08-15 11:09:07	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2024-08-15 11:09:37	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

Figure 5.18 Reindex and joining both cooler dataset and extruder vs dryer dataset

```
from sklearn.model_selection import train_test_split

test_size = 0.2
val_size = 0.1

train_set_A, test_set_A = train_test_split(labelled_SKU_A_Extruder_vs_Dryer_data, test_size=test_size, random_state=42, shuffle=False)
test_set_A, val_set_A = train_test_split(test_set_A, test_size=0.1, random_state=42, shuffle=False)

train_set_A.info()
```

✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 676 entries, 2024-08-15 11:07:37 to 2024-08-15 16:45:08
Data columns (total 48 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   SKU                     659 non-null    object
 1   A_FeedRate              659 non-null    float64
 2   Feeder_Hz               659 non-null    float64
 3   SC_Water%               659 non-null    float64
 4   SC_Steam%               659 non-null    float64
 5   MainMotor_Hz            659 non-null    float64
 6   MainMotor_Amp           659 non-null    float64
 7   CutterSpeed             659 non-null    float64
 8   sc_WaterFlow            659 non-null    float64
 9   sc_SteamFlow            659 non-null    float64
 10  sc_Temperature          659 non-null    float64
 11  ddc_Temperature         659 non-null    float64
 12  barrel_inletTemp        659 non-null    float64
 13  barrel_Zone2Temp        659 non-null    float64
 14  barrel_Zone3Temp        659 non-null    float64
 15  FinalHead_Temp          659 non-null    float64
 16  totalTonnage            659 non-null    float64
 17  total_Runtime           659 non-null    float64
 18  liveBin_Weight          659 non-null    float64
 19  DryerZone1_ActualTemp   676 non-null    float64
...
 46  Hardness (kg/cm)        676 non-null    float64
 47  Buoyancy (%) (Sink/Float)  676 non-null  int64
dtypes: float64(46), int64(1), object(1)
memory usage: 258.8+ KB
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

Figure 5.19 Train test split on SKU-A extruder vs dryer dataset

Only then the data was split into test set, training set, and validation set with ratio of *16:2:1*. as shown in figure 5.#. Since the data is a continuous data, Pearson correlation matrix is plotted to visualize the correlation between the features and the label data.

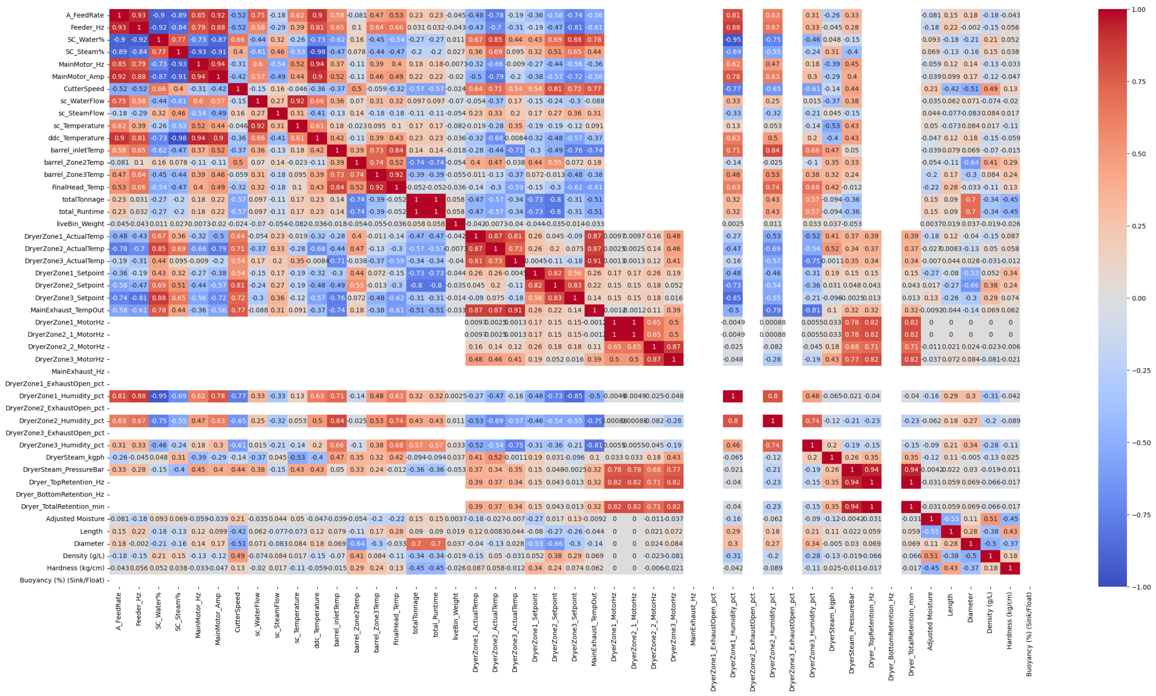


Figure 5.20 Pearson correlation matrix of joined SKU-A dataset.

As illustrated in figure 5.20, the matrix shows there are some columns without correlation between the independent variables and the dependent variable such as “MainExhaust_Hz”, “DryerZone1_ExhaustOpen_pct”, “DryerZone2_ExhaustOpen_pct”, “DryerZone3_ExhaustOpen_pct”, and “Dryer_BottomRetention_Hz”. These features have to be removed from the dataset as it implies that the feature has nothing to do with the dataset.

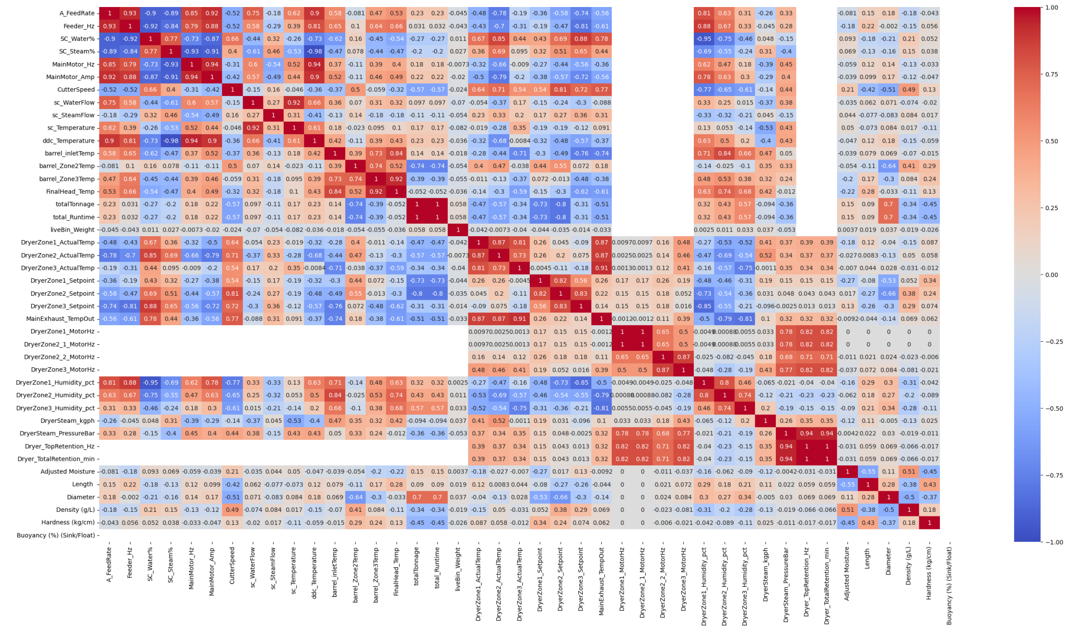


Figure 5.21 Pearson correlation matrix after performing first feature selection

With the removed features, another correlation matrix was plotted to continue investigate the correlation between the independent feature and the dependent feature. Features such as "DryerZone1_MotorHz", "DryerZone2_1_MotorHz", "DryerZone2_2_MotorHz", "DryerZone3_MotorHz", "Dryer_TopRetention_Hz", and "Dryer_TotalRetention_min" are uncorrelated with the first half of the data and only appears to shows some correlation between other variables in the second half. Thus, this triggers further studies on these columns by checking on the value count on the column itself.

```
features_to_investigates = ["DryerZone1_MotorHz", "DryerZone2_1_MotorHz", "DryerZone2_2_Mo

for col in features_to_investigates:
    print(train_set_A[[col]].value_counts())
    print()
✓ 0.0s

DryerZone1_MotorHz
42.0                 670
44.0                   4
38.0                   2
Name: count, dtype: int64

DryerZone2_1_MotorHz
40.0                   670
42.0                     4
36.0                     2
Name: count, dtype: int64

DryerZone2_2_MotorHz
40.0                   669
42.0                     4
36.0                     3
Name: count, dtype: int64

DryerZone3_MotorHz
38.0                 673
34.0                   3
Name: count, dtype: int64

Dryer_TopRetention_Hz
45.0                    674
...
23.61                      674
0.00                         2
Name: count, dtype: int64
```

Figure 5.22 Value count of the investigated columns.

Based on the result, the dryer zone motor Hz seems to be running in a preset frequency as oppose to a continuous value. Looking at the dryer retention too shows the value are also fixed at a constant. Thus, decision was made to remove the investigated columns and the uncorrelated all together from the dataset, as the feature would not be helpful in aiding the model to make accurate prediction. Then, a final correlation matrix is computed to as seen in figure below.

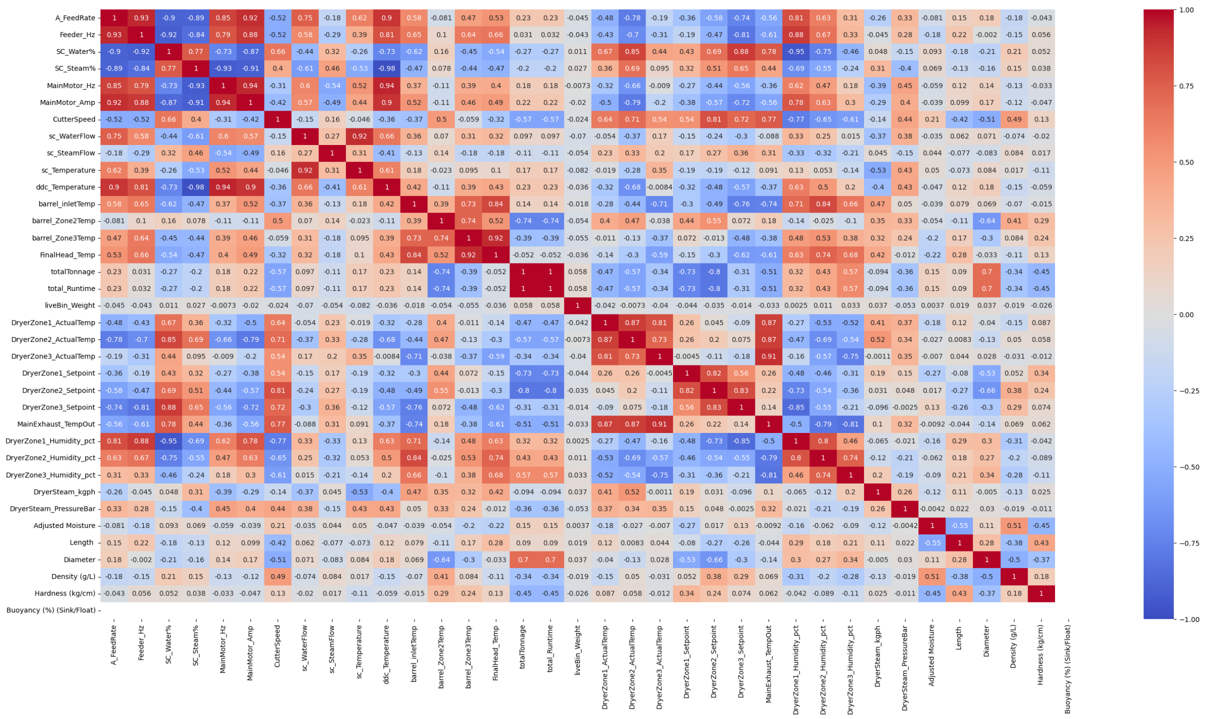


Figure 5.23 Final correlation matrix of the dataset

With the final correlation matrix, the independent variables do not show much correlation between the dependent variable. There are few possible issues, first is that there are not enough data in terms of quantity to show the full image of the whole dataset; second is that there are other factors that may require feature extraction and aggregation. Since feature extraction and aggregation could not be performed due to the limitation of the machineries, the process will be continuing with the dataset.

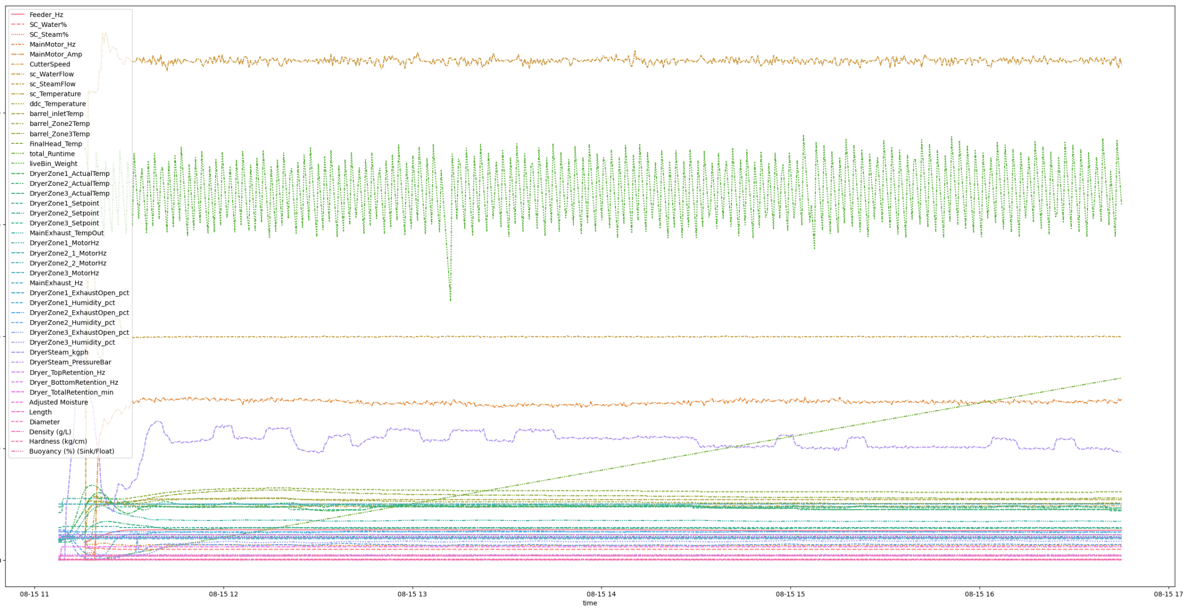


Figure 5.24 Line plot of features to time

As illustrated in figure 5.24 the line plot shows a time series interpretation of the data. From a glance, there is no high fluctuation except at the start of the plot, which indicates the machinery wind up configuration. Other than that, the configuration across the graph seems to be stable and does not show any sign of trends.

5.3.3 Data Preparation

With the data thoroughly investigated, before getting into the modelling stage, the available input data has to be pre-processed to fit into the models. By utilizing scikit-learn library, a preprocessing pipeline is prepared for input preprocessing.

```
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
from sklearn.impute import SimpleImputer

features = [feature for feature in included_feature if feature not in labels and feature not in unnecessary_feature]

# Define the preprocessor pipeline
preprocessor = Pipeline(
    steps=[
        ("imputer", SimpleImputer(strategy="mean")),
        ("scaler", StandardScaler()),
    ]
)

preprocessing = ColumnTransformer([
    ("preprocessing", preprocessor, features)
])

preprocessing
```

✓ 0.0s

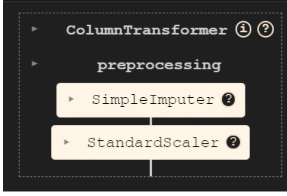


Figure 5.25 Scikit-learn preprocessing pipeline

In figure 5.25 shows the preprocessing pipeline that consist of a simple imputer, and a standard scaler. Simple imputer is used to fill the missing value with mean of their respective feature based on the training data. This ensures that the incomplete data points do not negatively affect the model performance.

Since the data consist of features with vastly different scales, the values have to be scaled to normalized the values across the rows. The process transforms the data to have a mean of 0 and standard deviation of 1, ensuring the features with large numerical range do not disproportionately influence the model.

5.3.4 Modelling

From the figure 5.24, the line plot does not show any visible fluctuation and trends except for the machine wind up time and the increase in "liveBin_Weight" which indicates the storage compartment in the machinery. Although the data has no visible trends that is apparent, but since it is a sequential data that exist across different time steps. Long Short-Term Memory (LSTM) and simple Recurrent Neural Network (RNN) was considered to be used at the modelling stage. Thus, the pre-processed data are then padded into an input sequences of a consistent length, and reshaped to 3D input format which is in *(sample, time steps, features)* as seen in figure below.

```
import numpy as np

# Reshaping function
def reshape_data(df, timesteps):
    data = df.to_numpy()
    num_samples = data.shape[0] // timesteps
    reshaped_data = data[:num_samples * timesteps].reshape((num_samples, timesteps, data.shape[1]))
    return reshaped_data

# Define the max timesteps
max_timesteps = 40

# Convert and reshape the scaled datasets
X_train_reshaped = reshape_data(pd.DataFrame(X_train_scaled), max_timesteps)
X_validation_reshaped = reshape_data(pd.DataFrame(X_val_scaled), max_timesteps)
X_test_reshaped = reshape_data(pd.DataFrame(X_test_scaled), max_timesteps)

# Check shapes
print("X_train_reshaped shape:", X_train_reshaped.shape)
print("X_validation_reshaped shape:", X_validation_reshaped.shape)
print("X_test_reshaped shape:", X_test_reshaped.shape)
✓ 0.0s

X_train_reshaped shape: (16, 40, 30)
X_validation_reshaped shape: (1, 40, 30)
X_test_reshaped shape: (2, 40, 30)
```

Figure 5.26 Reshaped input data

Figure above illustrates the process of reshaping the input data from 2D to 3D. A max time steps of 40 data per timesteps is defined and the data are shaped to capture the temporal form of the data.

```
import tensorflow as tf
import keras
from keras import layers, preprocessing

X_train_padded = preprocessing.sequence.pad_sequences(X_train_reshaped, maxlen=max_timesteps, padding='post', truncating='post')
X_val_padded = preprocessing.sequence.pad_sequences(X_validation_reshaped, maxlen=max_timesteps, padding='post', truncating='post')
X_test_padded = preprocessing.sequence.pad_sequences(X_test_reshaped, maxlen=max_timesteps, padding='post', truncating='post')
✓ 0.0s
```

Figure 5.27 Padded input data

Figure above illustrate the sequence padding process; the input data are padded to fill the null values with 0 as an indicator for the neural network. Then the input data are trained with both different neural network, first with simple RNN, and the second with LSTM.

Layer (type)	Output Shape	Param #
masking_1 (Masking)	(None, 40, 30)	0
simple_rnn_2 (SimpleRNN)	(None, 40, 50)	4,050
dropout_14 (Dropout)	(None, 40, 50)	0
simple_rnn_3 (SimpleRNN)	(None, 50)	5,050
dropout_15 (Dropout)	(None, 50)	0
dense_5 (Dense)	(None, 6)	306

```
Total params: 9,406 (36.74 KB)

Trainable params: 9,406 (36.74 KB)

Non-trainable params: 0 (0.00 B)
```

Figure 5.28 Simple RNN neural network

The first model built is a simple RNN neural network. The architecture starts with a Masking layer which mask any missing values in the input data by treating value 0 as a placeholder. This ensures the padded value do not interfere with the learning process. Then, a simple RNN layer with 50 units and RuLU activation is included to process the sequential data, and learn patterns and dependencies over time. After the first simple RNN, a Dropout layer is introduced with 20% dropout rate to prevent overfitting by randomly ignoring subsets of unit during each epoch. Then, a second simple RNN layer with 50 units and ReLU activation is included to further refine the sequential learning process. Similarly, a Dropout layer with 20% dropout rate is applied for additional regularization. Finally, the model end with a Dense output layer, which predict based on the number of unique labels in the prediction task.

Layer (type)	Output Shape	Param #
lstm_10 (LSTM)	(None, 40, 50)	16,200
dropout_12 (Dropout)	(None, 40, 50)	0
lstm_11 (LSTM)	(None, 50)	20,200
dropout_13 (Dropout)	(None, 50)	0
dense_4 (Dense)	(None, 6)	306

Total params: 36,706 (143.38 KB)

Trainable params: 36,706 (143.38 KB)

Non-trainable params: 0 (0.00 B)

Figure 5.29 LSRM neural network

The second model built as an advancement from the first model, a LSTM neural network. It starts with a LSTM layer containing 50 units with tanh activation. This layer aimed to capture the long-term dependencies in the sequential data, especially in the given dataset with variable timesteps. Similarly, a Dropout layer is added with 20% dropout rate to reduce overfitting by randomly dropping connection during training. Then, a second LSTM layer with 50 units and tanh activation is included to further improve the temporal feature extraction. Then, another Dropout layer is applied for regularization with 20% dropout rate. The final layer will be a Dense output layer, which produce prediction based on the number of unique labels in the prediction task.

5.4 Computer Vision Model Implementation

5.4.1 Business Understanding

Following up to section 5.3.1, the implementation has been focussed on the image-based quality control which will needs to further investigated with the images and identify the classes to be assigned to different image. The technical aspect such as resolution, lightning condition, and the acceptable colour variations across different image encodings have to be considered when modelling the computer vision model.

The computer vision model has to be aligned to the objective in terms of its ability to streamline the quality control process and reduce the need of inspection efforts, thus increasing the production efficiency. Through monitoring the visual quality of the pellets, the system can support in maintaining the product standards, while also potentially reducing waste.

5.4.2 Data Understanding

Unlike predictive model, this project does not receive nor found any images through third party providers. Instead, all the images are manually downloaded from various sources through Google Image. In addition, there are not much sources that fit the minimum requirement of the predetermined image size, which are 250x250 *(width x height)*. The total number of images downloaded are only at 169 images.

The images are then manually sorted to two different classes, “good quality” and “bad quality”. The terms “bad quality” is based on the pellet appearance and colours, those appears to looks discoloured or deformed, or those that have broken down into a powdery form, as shown in figure 5.#. The “good quality” pellets are based on the pellet appears to have a regular pellet shape without any sort of decolourisation as shown in figure 5.30

Figure 5.30 Sample picture of a “bad quality” pellet

Figure 5.31 Sample picture of a “good quality: pellet

5.4.3 Data Preparation

In the previous stage, the images are separated into two different classes. With the organised images, the images have to be standardised in terms of the image height and weight. For the project, the image height and width are set to constant of 250 pixel each. The batch size for processing is also set to a constant of 32 pictures per batch.

```
# Create a dataset
def preprocess_image(image, label):
    image = tf.image.resize(image, [img_height, img_width])
    image = tf.cast(image, tf.float32) / 255.0  # Normalize to [0, 1]
    return image, label


def load_data_from_directory(directory, class_names, batch_size, validation_split=0.2):
    dataset = tf.keras.utils.image_dataset_from_directory(
        directory,
        image_size=(img_height, img_width),
        batch_size=batch_size,
        label_mode='binary',
        class_names=class_names,
        subset='training',
        validation_split=validation_split,
        seed=123
    )
    val_dataset = tf.keras.utils.image_dataset_from_directory(
        directory,
        image_size=(img_height, img_width),
        batch_size=batch_size,
        label_mode='binary',
        class_names=class_names,
        subset='validation',
        validation_split=validation_split,
        seed=123
    )

    # Apply preprocessing
    dataset = dataset.map(preprocess_image)
    val_dataset = val_dataset.map(preprocess_image)

    return dataset, val_dataset

train_data, val_data = load_data_from_directory('data/images/clusters_test', ("cluster_0", "cluster_1"), batch_size)
✓ 0.1s

Found 169 files belonging to 2 classes.
Using 136 files for training.
Found 169 files belonging to 2 classes.
Using 33 files for validation.
```

Figure 5.32 Image preprocessing to resize and create dataset

With the predetermined constant, the images are then loaded with Keras to create a dataset to prepare for image classification modelling as illustrated in figure 5.32. Set of training data, testing data, and validation data are created through the *load_data_from_dictionay()* method as the testing data is embedded within the process.

5.4.4 Modelling

Continuing from previous stage, a binary classification model will be used such as Convolutional Neural Network (CNN) to process the images then

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 248, 248, 32)	896
max_pooling2d (MaxPooling2D)	(None, 124, 124, 32)	0
conv2d_1 (Conv2D)	(None, 122, 122, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 61, 61, 64)	0
conv2d_2 (Conv2D)	(None, 59, 59, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 29, 29, 128)	0
conv2d_3 (Conv2D)	(None, 27, 27, 256)	295,168
max_pooling2d_3 (MaxPooling2D)	(None, 13, 13, 256)	0
conv2d_4 (Conv2D)	(None, 11, 11, 512)	1,180,160
max_pooling2d_4 (MaxPooling2D)	(None, 5, 5, 512)	0
flatten (Flatten)	(None, 12800)	0
dense (Dense)	(None, 512)	6,554,112
dense_1 (Dense)	(None, 1)	513

Total params: 24,369,605 (92.96 MB)

Trainable params: 8,123,201 (30.99 MB)

Non-trainable params: 0 (0.00 B)

Optimizer params: 16,246,404 (61.98 MB)

Figure 5.33 CNN model for image classification

Based on figure 5.33, a Convolutional 2D layer is introduced with 32 filters and a 3x3 kernel size, using ReLU activation function. The first layer is responsible for detecting basic features in the input images, searching for edges and textures. Then the second layer is a MaxPooling 2D layer with a 2x2 pool size to reduce the spatial dimensions of the feature map, which helps to increase the computation efficiency by down-sampling the image. Starting from the second layers are being repeated through increasing the filter size from 32 with 64, 128, 256, and 512 filters, to increase the capability of the model to learn mode complex and higher-level features at each stage. Once the image go pass every Convolutional layer, it will be flattened into a 1D vector which will then pass through two Dense layers, with 512 units and ReLu activation, and another with 1 unit with sigmoid activation function which output the probability score between 0 and 1 for binary classification.

```
val_loss, val_acc = model.evaluate(val_data)
print(f"Validation Accuracy: {val_acc}")
✓ 0.1s

2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - accuracy: 0.5511 - loss: 0.6893
Validation Accuracy: 0.5454545617103577
```

Figure 5.34 Trained CNN model average accuracy score

Due to the model does not perform well mainly due to low sample size, the trained CNN model only achieves an average score of ~0.5455. Thus, the model is further optimized through introducing Few-Shot Learning with Siamese Network.

Utilising Few-Shot Learning allow the model to utilize small amount of labelled example to make accurate prediction, further addressing the data scarcity issue. Through Siamese Network a pair of identical subnetworks, sharing the same weight will process two input images and determine whether if both of the image belongs to the same class. But it introduced a tweak in the phase of determining the output, rather than classifying the image, the model measures the similarity between two images, hence suitable with model trained on scarce dataset.

```
val_loss, val_acc = siamese_model.evaluate([val_pairs[:, 0], val_pairs[:, 1]], val_labels)
print(f"Validation Accuracy: {val_acc}")
✓ 0.1s

1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 93ms/step - accuracy: 0.8000 - loss: 0.7660
Validation Accuracy: 0.800000011920929
```

Figure 5.35 Validation accuracy of Siamese CNN Model Using Few-Shot Learning

As illustrated in figure 5.35, the validation accuracy of the Siamese CNN model is 0.8000 demonstrates the model’s ability to generalize well, even with the limited dataset.

5.5 System Operation

5.5.1 Python Program in Raspberry Pi

There are total of 3 programs hosted and running in the Raspberry Pi. Based, the running programs are Sensor Controller, Sensor Data Controller, Camera Module.

Starting with Sensor Controller, as mentioned, it handles the operation of the DHT11, and MCP3008. Both sensors have a Python library provided by Adafruit [30], [31], [32], while the moisture sensor did not have one publicly available. Thus, using the MCP3008 Python library provided by Adafruit, a conversion from the analogue to digital has been programmed to fit the supported input of Raspberry Pi. Using Object-Oriented Programming Practice (OOPP), both sensors are abstracted by a class each exposing the default behaviour and with extra functionalities scoped to the need of the project. The aim is to make the sensor class as generic as possible to ease the future development if something needed to be changed.

In addition, due to the processing power constraint, a thorough memory and processing management had been done to make the program runs more efficient. Since the both of the sensors needed to be run in an infinite loop at the same time, the sensors are programmed to be run in concurrency utilising Linux's concurrency support. Both sensors are managed by a manager which then initialises the concurrency control to create two software threads managing the instantiated sensor with same memory address.

In Sensor Data Controller, the program is also restricted by the limited resource. Thus, considering the controller needs to constantly listening to the MQTT broker for message, while writing the response value to InfluxDB, multiprocessing is used in handling the situation. Since the Python version used in the current project is 3.9.2, according to the documentation, the multiprocessing library uses fork by default in Unix environment [33]. By using the default behaviour, Python interpreter spawn a child process which utilising the same shared memory which is the instance of subscription to MQTT broker.

5.5.2 Sensors Manager

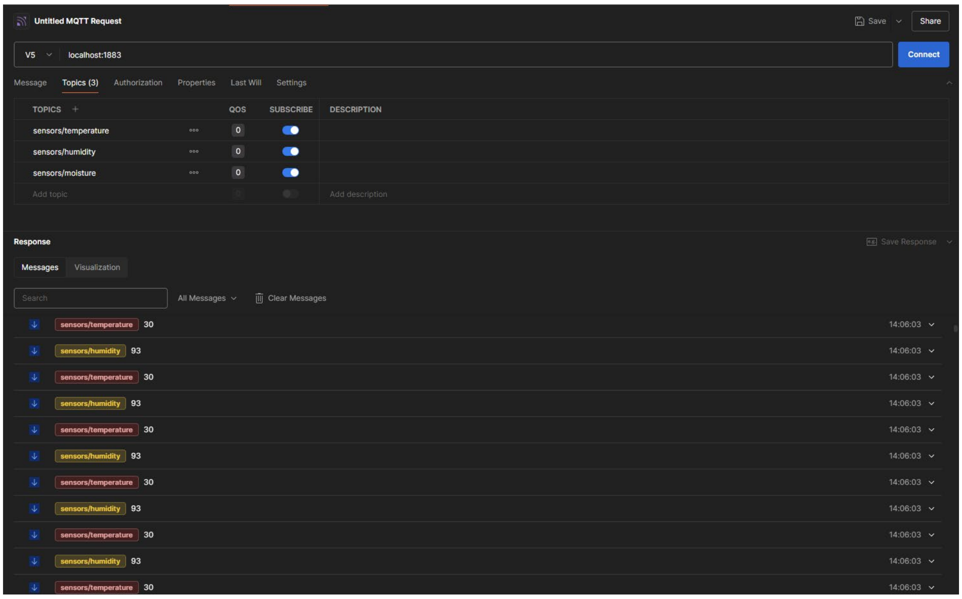


Figure 5.36 Postman interface visualising MQTT packets

Figure above illustrate the signals being sent through MQTT protocol. In the figure itself, subscription to sensors/temperature, sensors/humidity, and sensors/moisture can be seen as these are the available sensors that installed onto the Raspberry Pi and controlled through the Sensor Controller. Due to the DHT11 sensors being too responsive, the interval between the signal received is short and able to capture a lot of information in one second.

5.5.3 Grafana Dashboard

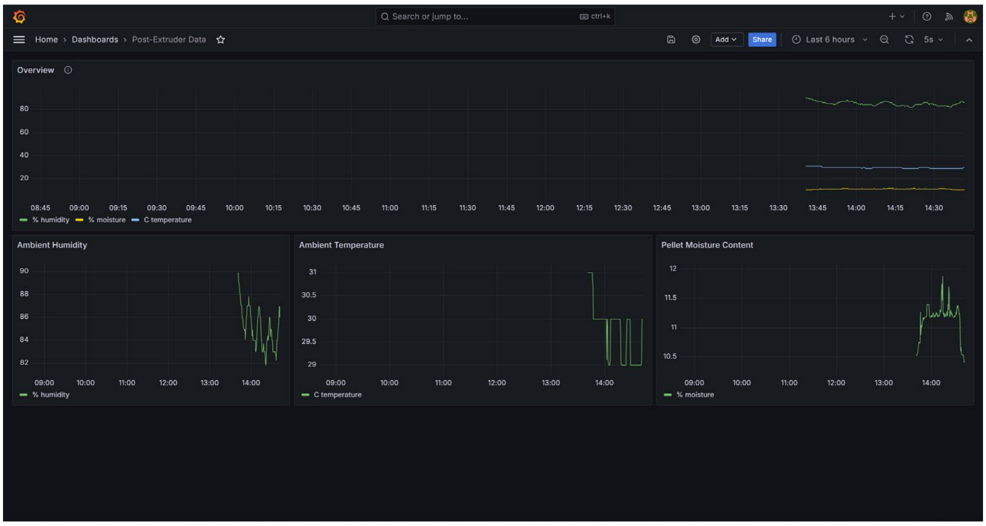


Figure 5.37 Grafana dashboard configured

To visualised the collected data, Grafana dashboard will be used for monitoring the real-time data collected by the sensors. This is to add another layer of precaution in the case of any error produced by the prediction model. The main purpose is solely for monitoring and controlling.

5.5.4 Predictive Model

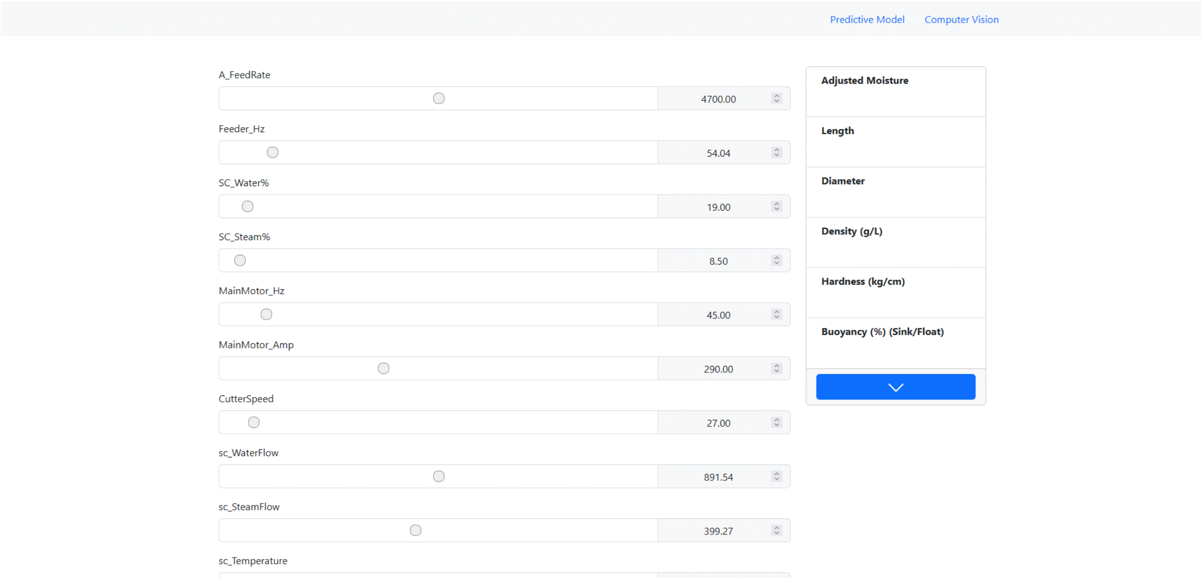


Figure 5.38 Web application for controlling the prediction model

A Python Flask application was developed to showcase the predictive model in action for demonstration purpose. The web interface features groups of input at the left, and the output is displayed at the right. Upon inserting the input either through number input or the slider, the inputs are fitted into the predictive model to generate predicted output and display at the right container.

When the webpage is first loaded, set of default values are assigned to each input fields, allowing the user to quickly observe a prediction without making any adjustment. User can also fine-tune the input parameters through the sliders and submit changes to update the prediction output.

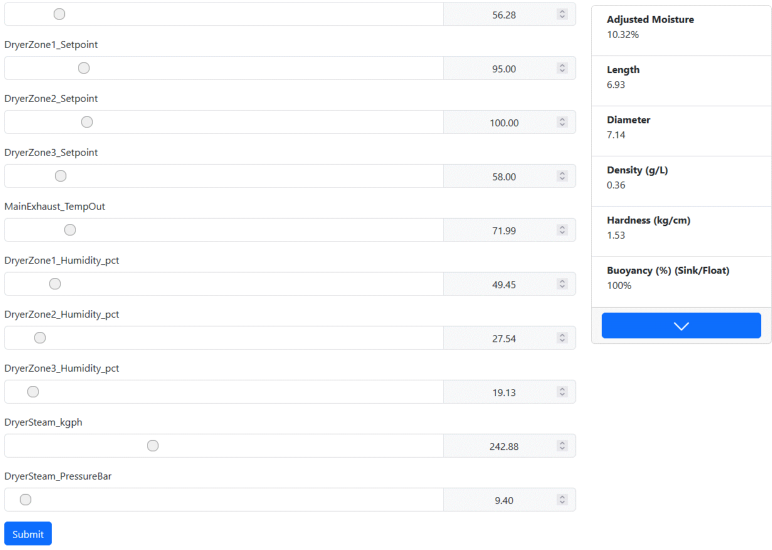


Figure 5.39 Output value from the predictive model

With the default value being sent to the Flask server, it will be parsed and padded with sequence of 40 timesteps to ensure it fits the required input format. In the demo application, the 40 timesteps will be padded with additional values, generated from a normal distribution. This is crucial as it stimulate a complete sequence for the model to process. This approach also showcased the model's ability in handling sequential data in a real-world scenario when full sequences are provided.

5.5.5 Computer Vision Model

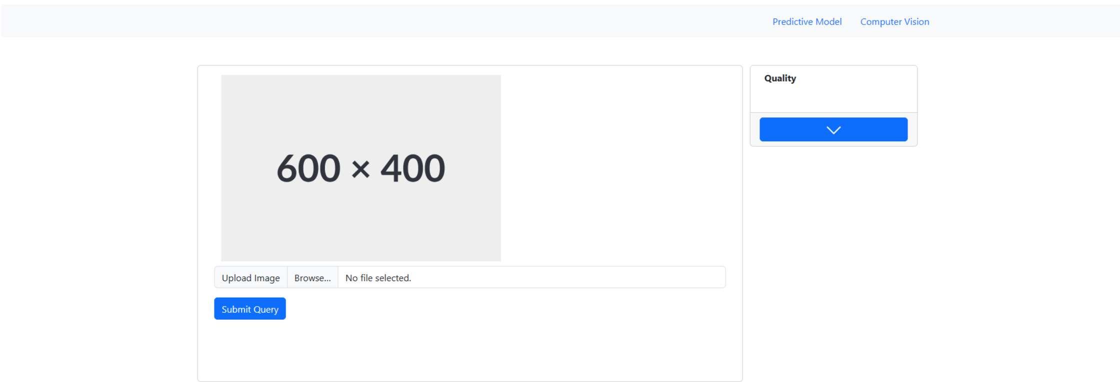


Figure 5.40 Web application for computer vision demo

Another page from the Python Flask application has also been developed to demonstrate the use of the computer vision classifier. The interface uses the same design approach as the previous predictive model, allowing user to use the demo application quickly knowingly what to click at first.

When the webpage is loaded, the application shows a placeholder image and a file input, for the user to upload image of pellets onto the application. Once the image sent to the server, the server will preprocess it and fit it into the computer vision model trained earlier to get the classification prediction outcome.

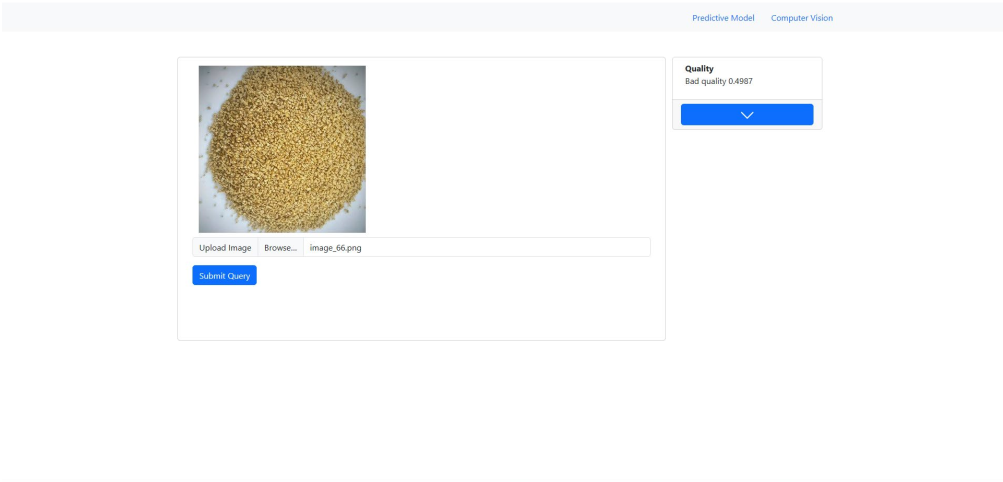


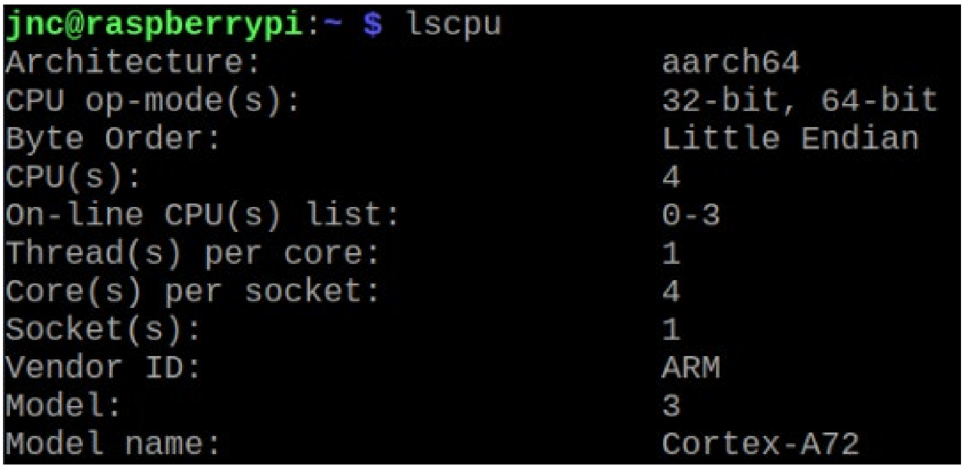
Figure 5.41 Output value from the computer vision model

By sending an image to the server, the server preprocess and fit it into the model hence showing the quality outcome. If the quality outcome is larger than 0.5 then it will be good quality pellets, if the outcome is less than 0.5, it will not be a quality pellet. The image provided in figure 5.41 is labelled as non-quality pellets as it is in a powdery form rather than pellets form.

5.6 Implementation Issues and Challenges

5.6.1 Restricted Resource at Raspberry Pi

The first issue that arise when developing the system is the hardware constraint of the Raspberry Pi. Based on the technical hardware specification shown in 4.3, the Raspberry Pi has only 2GB of RAM / Memory space. Which is not sufficient to run any piece of applications that are too large. In addition to the insufficient memory space, although the processor has four processing cores, but each core only operates with one hardware thread as shown in figure below.



```
jnc@raspberrypi:~ $ lscpu
Architecture:                    aarch64
CPU op-mode(s):                  32-bit, 64-bit
Byte Order:                      Little Endian
CPU(s):                          4
On-line CPU(s) list:             0-3
Thread(s) per core:              1
Core(s) per socket:              4
Socket(s):                       1
Vendor ID:                       ARM
Model:                           3
Model name:                      Cortex-A72
```

Figure 5.42 CPU specification listed through Linux command

With the processing power limitation, the application programmed has to be constantly improved into a more efficient piece and utilise the features provided by the operating system, in this case Raspberry Pi OS powered by Debian Linux.

5.6.2 Moisture Sensor

The second issue is the capacitive moisture sensor only has one analogue output. An ADC microcontroller is needed to convert the analogue signal to digital output as Raspberry Pi only supports digital input and output. Thus, the retrieved data from the sensors are only ADC value and voltage of the sensor. It does not provide any data on the moisture content. Hence, to calculate the moisture content, calibration and formulation of algorithm needed to be performed at the environment to be as accurate as possible.

In addition to the capacitive moisture sensor analogue output, the sensor itself has exposed circuit as seen in Figure 5.43. It cannot be exposed to water as it will cause short circuit. Thus, solution such as using recyclable material or 3D printing to make a casing encapsulating the circuit component leaving the detection part open.

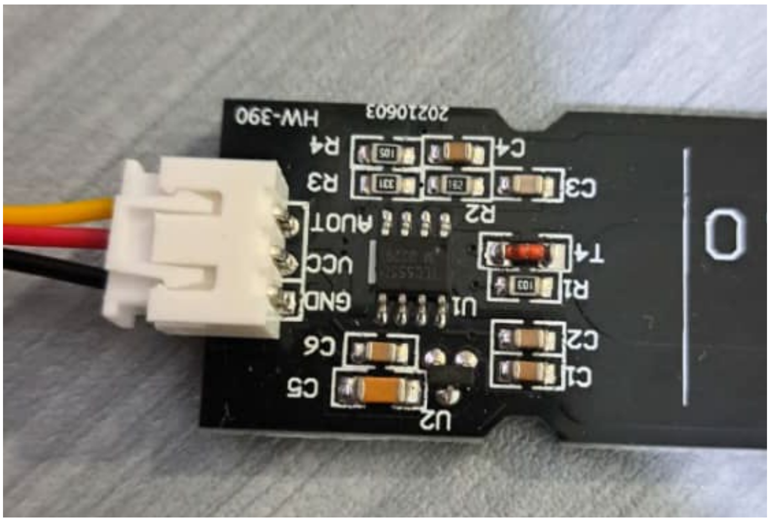


Figure 5.43 Exposed circuit on the moisture sensor

5.6.3 Insufficient Data for Model Training

As mentioned before, the data received from the third-party provider does not sufficient enough to train the neural network model. Although there are a pair of datasets provided by the provider for two different product SKUs, one of the product datasets could not be used due to missing label in the dataset, while the other has too less data for the model to be consumed. The dataset given for the SKU-A has only 845 entries where many of them are null values. Since it is a timeseries dataset, the data received only consist of data for a day. This bring an issue to the implemented LSTM model, where the model is being fed with too insufficient data, causing it hard to generalised the underlying pattern.

Not only the LSTM model, the computer vision on the other hand has no data hence require manual data selection. This method brings up disadvantage where the image downloaded from other sources may tends to be imbalance at the end. “Good quality” pellets may outnumber the “Bad quality” pellets, or vice versa. This imbalance will bias the model final prediction when it comes to predicting the given input.

In image classification task, a balance dataset is important to ensure the model does not favour one class than the others. It helps the model to generalise and recognise the underlying pattern hence able to distinguish the classes effectively.

Moreover, the image quality is also another crucial factor when training computer vision task. At the stage of sourcing images from external sources, many of the image does not fit into the predetermined minimum requirement. Thus, limiting the number of images that can be selected also introduced another issue where the lighting and the image quality may hinder the outcome of the computer vision model accuracy.

5.7 Conclusion Remark

In conclusion the system implementation is a success, although the flow between each project objective may not seems to linked up, but project objective one is solely for demonstrating the flow on how the system will be interacting in the real world. The main gist is in the implementation of project objective two and three, where it provides the backbone of the whole system through machine learning model.

With that being said, the project too faced several challenges, especially with the scarce in data quality and quantity. It had directly impacted the model performance, thus limiting the model underlying ability to achieve higher prediction accuracy.

Chapter 6

System Evaluation

6.1 System Testing and Performance Metrics

6.1.1 Machine Learning Evaluation Metrics

For evaluating the performance of the time series predictive model, metrics such as mean absolute error, mean squared error, and validation loss at the latest stage.

Starting with the mean absolute error, it measures the average absolutes error in prediction. It provides a clear and straight forward interpretation on the model's prediction deviate from the actual value. For instance, a lower value in MAE indicates the prediction are closer to the actual value.

Then, with mean squared error, it shows the variance in the prediction error while penalizing the larger errors. This metrics is valuable in a production environment, as it tries to eliminate large deviation from the model. Since the model will be placed in a continuous production environment, and large deviations can be costly. Thus, MSE helps in minimising the impact of the errors, mainly in improving the reliability of the model

With the validation loss, it helps in measures how well the model generalized to the unseen data. By monitoring the validation loss, the model can be assessed to figure out whether it is overfitting or underfitting.

To evaluate the performance of the computer vision model in the other hand, metrics such as accuracy, precision and recall, F1 score, ROC Curve and AUC (Area Under the Curve), and confusion matrix. With accuracy, it shows the proportion of the correctly classified data out of the total input data, it acts as a general measurement on showing how many times the model successfully classifies the correct output.

Precision and recall can be helpful as there will be a trade-off in terms of the correctness of the prediction. Take precision as an example, precision measures the correctness of the positive prediction is actually correct. By focusing on precision, the model will become more conservative. As oppose to recall, it prioritises in ensuring the model captures as many actual positive as possible.

F1 score provide the harmonic mean of both precision and recall. It provides a single metric that balances out both precision and recall, and it shows it usefulness in imbalance classes.

ROC Curve and AUC shows the recall against the false positive rate, it helps to visualize the performance if of the model in different threshold.

At last, confusion matrix is also chosen as it shows the count of prediction of true positives, true negatives, false positive, and false negatives. It provides a detailed view on how the model's prediction and compared to the actual label of their corresponding data.

6.1.2 Sensors Data Evaluation Metrics

Since there are no viable way of testing the sensors accuracy as it will be based on the sensor itself, this project simulates environment with high fluctuation data input. For instance, the moisture sensor will be dipped in and out of the container of water, while the moisture and temperature sensor will be blown by a hair dryer to by unfixed amount of interval to check the responsiveness of the data being sent to the server and stored in the database.

6.2 Testing Setup and Result

6.2.1 Machine Learning Evaluation Result

Continuing from Section 5.3.4, two models are prepared where the first is the simple RNN model, while the second is LSTM model. The evaluation of both models is performed prior deploying onto the demo application. Both models use validation data and use the trained model to predict to calculate for the evaluation metrics.

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Compute metrics
mae = mean_absolute_error(y_val[:X_val_padded.shape[0]], y_pred)
mse = mean_squared_error(y_val[:X_val_padded.shape[0]], y_pred)
rmse = np.sqrt(mse)

print(f'Mean Absolute Error: {mae}')
print(f'Mean Squared Error: {mse}')
print(f'Root Mean Squared Error: {rmse}')
✓ 0.0s

Mean Absolute Error: 0.6763942942033211
Mean Squared Error: 1.0206170923277351
Root Mean Squared Error: 1.0102559538689861
```

Figure 6.1 Evaluation metrics on simple RNN

Starting with simple RNN, the mean absolute error is 0.6764, mean squared error is 1.0206, and root mean squared error is 1.010. With these calculated metrics it is safe to say that the model does not suitable for the application and cannot be used as the primary model to predict the output of based on the input features. The values indicate the model's prediction is largely deviate from the actual value.

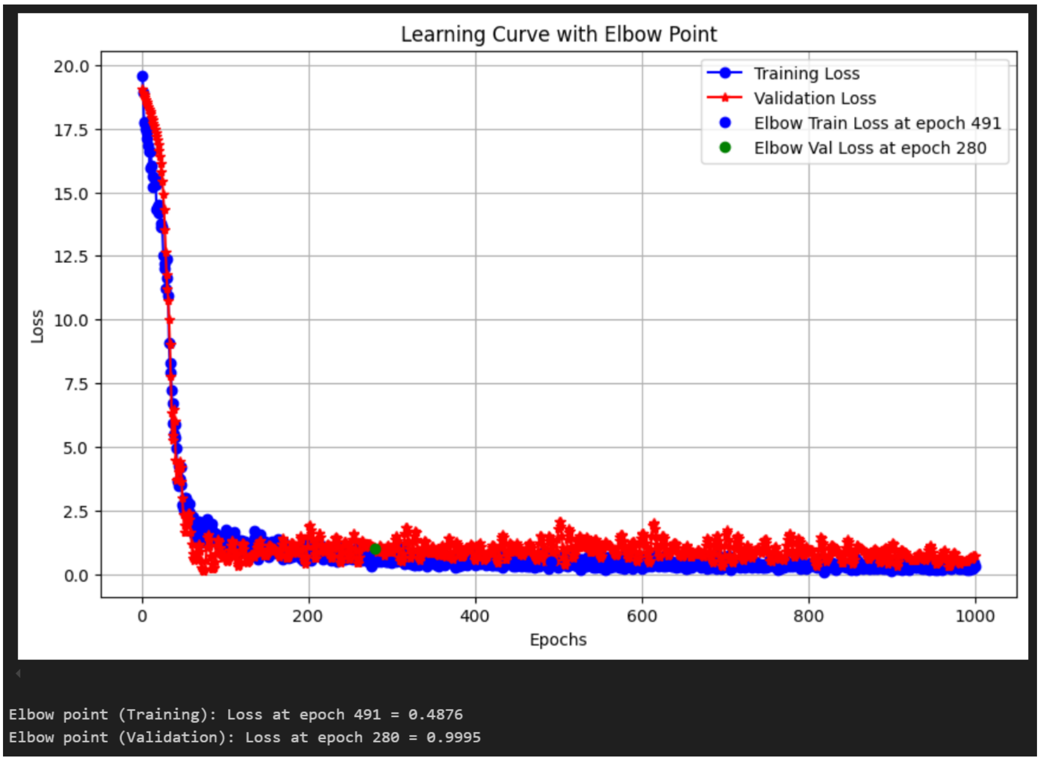


Figure 6.2 Evaluation of Simple RNN Model

Based on the figure, the lowest validation loss at the model, it also has a high value of 0.9995 almost to 1.0. The model too unable to generalise well to the unseen data, and further implies that the model is not suitable for the prediction task.

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Compute metrics
mae = mean_absolute_error(y_val[:X_val_padded.shape[0]], y_pred)
mse = mean_squared_error(y_val[:X_val_padded.shape[0]], y_pred)
rmse = np.sqrt(mse)

print(f'Mean Absolute Error: {mae}')
print(f'Mean Squared Error: {mse}')
print(f'Root Mean Squared Error: {rmse}')

✓ 0.0s

Mean Absolute Error: 0.1798933411876361
Mean Squared Error: 0.11482568019178378
Root Mean Squared Error: 0.3388593811476728
```

Figure 6.3 Evaluation metrics for LSTM Model

Based on figure 6.3, the model achieved a mean absolute error of 0.1799, mean squared error of 0.1148, and root mean squared error of 0.3389. The low mean absolute error indicates that the difference between the predicted value and the actual value is mild, the mean squared error also further captures the reliability of the model by penalizing larger error more heavily. The root mean squared error, provides more interpretable measurement of the models, affirming that the overall error of the model remains low.

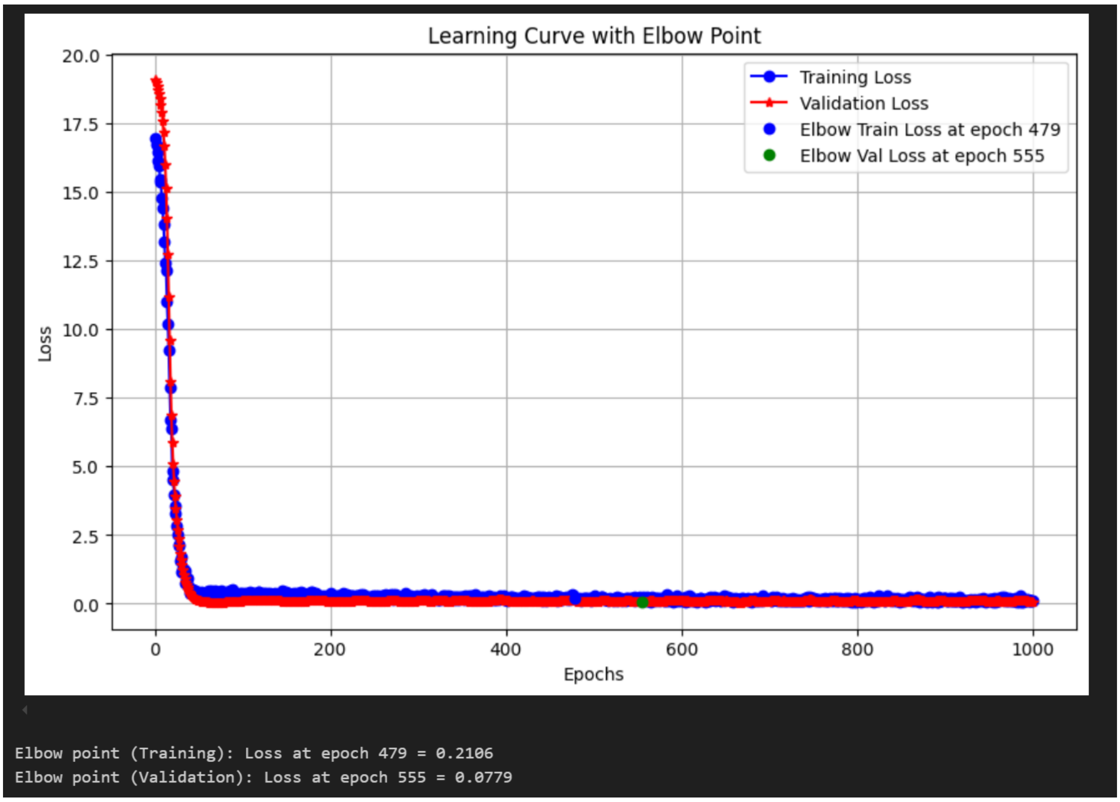


Figure 6.4 Evaluation of LSTM model

Based on figure 6.4 the lowest validation loss at the model is 0.0779. It indicates that the model is able to generalise well to the unseen data.

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score, confusion_matrix

y_true = val_labels

# Calculate evaluation metrics
accuracy = accuracy_score(y_true, binary_predictions)
precision = precision_score(y_true, binary_predictions)
recall = recall_score(y_true, binary_predictions)
f1 = f1_score(y_true, binary_predictions)
roc_auc = roc_auc_score(y_true, predictions)
conf_matrix = confusion_matrix(y_true, binary_predictions)

print(f"Accuracy: {accuracy}")
print(f"Precision: {precision}")
print(f"Recall: {recall}")
print(f"F1 Score: {f1}")
print(f"ROC AUC: {roc_auc}")
print(f"Confusion Matrix:\n{conf_matrix}")

✓ 0.0s

Accuracy: 0.8
Precision: 0.5
Recall: 1.0
F1 Score: 0.6666666666666666
ROC AUC: 0.125
Confusion Matrix:
[[6 2]
 [0 2]]
```

Figure 6.5 Evaluation metrics of computer vision model

Moving on to the computer vision model, with the Siamese Network, the accuracy for the prediction with validation data is high at 0.8, precision scores only 0.5 indicating the model only made half of the prediction correct, while recall scores 1.0 stating that it has correctly predicted all the positive case correctly, F1 Scores 0.6667 indicating the models able to maintain a balance at precision and recall, and ROC AUC scores 0.125 which suggest that the model struggles to distinguished between true positives and false positives.

For the computer vision model, the moderate accuracy is due to the lack of data, as with the current trained accuracy, it has been a successful attempt as the sample data has only 169 images in total. The accuracy of 0.8 also suggested the Siamese Network has managed to generalized well. But the lack of data is much likely the main contributors to the low precision and struggle with distinguishing between true and false positive as reflected by ROC AUC scores.

6.2.2 Sensor Data Evaluation Testing

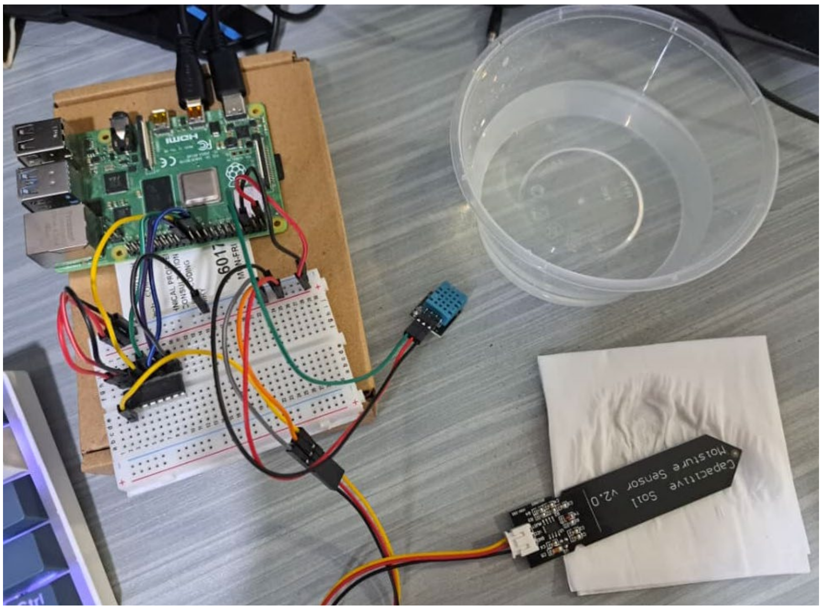


Figure 6.6 Testing environment setup

To satisfy the proof of concept, the current testing environment is setup to retrieve data based on indoor environment. The retrieved temperature and humidity are based on the condition of either switching on the air conditioner (A/C) or in room temperature. Moisture sensor in the other hand, the container of water indicates the absolute moist condition, and the tissue is to simulate the gradual decrease of moisture content.

Figure 6.7 Fluctuation of the sensors

In Figure 6.7 shows the fluctuation of sensor through stimulating the DHT11 by blowing hot air with a hair dryer. Hair dryer is blown from 00:17:00 to 00:18:00, then blown once again from around 00:19:31 to 00:21:55. The ambient temperature increase significantly when the air is directly blown to it, and the ambient humidity drops accordingly to the rise in temperature.

6.3 Project Challenges

One of the biggest challenges in this project is that the data required for training the neural network is not sufficient. When training both of the neural network, both the predictive model and especially the computer vision model took large amount of time just for training and finding solution to increase the overall accuracy of the model. Given that the dataset is known to be not sufficient, yet the features available are way too many; thus, increasing the model complexity and a big dimension of the data. Not only that, it also introduces the risk of overfitting where with the small dataset, the model will tend to memorise the training data and failed to generalised the pattern of the dataset. This alone lengthen the time to train the model, as well as optimising it for better performance.

Fast forward to computer vision model, also without sufficient images, the model unable to learn and generalised the pattern of the dataset. Causing low prediction accuracy and thus requires the need for utilising few-shot learning with Siamese Network. Although with Siamese Network, it is still not accurate enough as seen in

Figure 6.5. Although the accuracy is high but still unable to score well at ROC AUC score.

In addition, with the fast paced of technology evolution, many libraries that used in previous work has deprecated. Hence, a complete rework on the application needed to be performed such that the compatibility across all application can be achieved and to make sure the sensors could detect and process data correctly.

6.4 Objective Evaluation

Based on current development, the aforementioned project objectives have been successfully implemented and completed. However, further development is still necessary to enhance the system performance and scalability. As mentioned earlier, with the fast paced of technology evolution, continuous development is a must to keep up with the changes.

One critical area that require the most attention is data acquisition. As previously discussed in section 5.6.3, the implementation faces the main challenges due to the limited amount of data available for training, which hinders the model performance. Through obtaining more data will be crucial in improving the model effectiveness and accuracy.

Although that is the case, at the current stage of implementation, both models able to perform reasonably accurate prediction of the outcome with the input data. Susception would still be made available on both models as this little data would not be enough for the application to be deployed in the real world, at least not at this stage of training. It still lacks the required robustness and generalisation towards real world problem.

6.5 Conclusion Remark

As a summary for the chapter, it highlighted the key challenges faced in the implementation through evaluation and testing. Mainly due to the lack of data for training both predictive and computer vision model, resulting in limitation of model performance and accuracy. Although that is the case, both models still perform reasonably well in terms of predicting the outcome based on the implemented evaluation metrics. However, further development needs to be done in order to continuously improve the system performance.

Chapter 7

Conclusion

7.1 Conclusion

In conclusion, the project is an overall success. With the implementation of both LSTM model and Siamese Network with CNN as the base model for computer vision has been successfully realised.

With project objective one implementation of Raspberry Pi with humidity and temperature sensors, moisture sensors, and camera modules enabled the demonstration of the data collection in the production flow. With all the data collected, it will be sent to the Docker application container which act as a server to save all the input information and further processes the information.

Moving on to the second project objective where the predictive model comes in hand to predict the outcome based on the configuration settings from the user. With a front-end application which enables the user to tweak the input of each feature and find the most optimal configuration that will be needed to configure the machine especially the dryer to produce the optimal outcome.

With the third project objective, the computer vision model will be placed at the end of the production line to capture the output images, and send it to the Docker application container to perform image classification. With the results, the user will be notified that the produced outcome is good or bad.

7.2 Recommendation

While the implemented system works reasonably well in cure current state, further development has to be done to improve the system effectiveness and efficiency. First off, data acquisition has to be done to continuously train the model to achieve better generalisation on the given task. Given that the current dataset is way too less for the model to uncover the underlying pattern, more data needs to be acquired to change this. Such data can be acquired from third party provider or through process of data generation. Retrieving data from third party provider can be easy but at the same time needs time and resource to do so. With the current trends of generative artificial intelligence, data can too be generated through utilising these models to get more data.

Other than acquiring more data for training, continuous integration and continuous development (CI/CD) needs to be performed on the model as well. For instance, saving the input data and compile with the previous and automate the process to train the model continuously in a predetermined time interval. This approach will help to improve the performance of the model time by time and increase the robustness of the model.

REFERENCES

[1] Economic Planning Unit, Prime Minister's Department (2019). National Fourth Industrial Revolution (4IR) Policy. [Online]. Available: https://www.ekonomi.gov.my/sites/default/files/2021-07/National-4IR-Policy.pdf

[2] MyGovernment. "Industry4wrd National Policy On Industry 4.0". MyGovernment. https://www.malaysia.gov.my/portal/content/31224 (accessed Nov. 30, 2023)

[3] Dr. KC Behnke. "Factors influencing pellet quality," in Feed Tech, vol. 5, no. 4, pp. 19-22, Jan. 2002. [Online]. Available: https://www.academia.edu/download/38331153/6055-Factors_influencing_pellet_quality.pdf

[4] K. Muramatsu, A. Massuquetto, F. Dahlke, and A. Maiorka. "Factors that Affect Pellet Quality: A Review," in Journal of Agricultural Science and Technology, vol 5, pp. 717-722, Jan. 2015. doi: 10.17265/2161-6256/2015.09.002

[5] N. Kiktev, T. Lendiel, and V. Osypenko. "Application of the Internet of Things Technology in the Automation of the Production of Compound Feed and Premixes," in IT&I, (Kyiv, Ukraine), 2020, pp. 124-133. [Online]. Available: https://ceur-ws.org/Vol-2833/Paper_12.pdf

[6] J. A. Cabañas-Ojeda, N. J.J. Mejia-Abaunza, P. A. Lozano-Cruz, V. Aragão-Netto, S. Brown, A. Rubio, A. Fahrenholz, and E. O. Oviedo-Rondón. "Corn kernel hardness and drying temperature affect particle size post-hammer-milling and pellet quality in broiler and swine diets," in Animal Feed Science and Technology, vol. 304, Aug. 2023. doi: 10.1016/j.anifeedsci.2023.115744

[7] *Grafana*. (2024). Grafana Labs. Available: https://grafana.com/get/?tab=self-managed

[8] Single screw extruder, by D. Grunschloss. (2002, Sept. 19). US20020131322A1 [Online]. Available: https://image-ppubs.uspto.gov/dirsearch-public/print/downloadPdf/6705752

[9] Microwave Moisture Sensor, by S. Kiyobe and T. Hirano. (1987, Jun. 23). US4674325A [Online]. Available: https://image-ppubs.uspto.gov/dirsearch-public/print/downloadPdf/4674325

[10] Dr. A. Bekal, G. Balsubramaniam, V. Awaghade, and S. Ghute. “Application of Microwave Moisture Sensor for DOC and Animal Feed,” in IEEE Sensors Journal, vol. 20, no. 24, pp. 14809-14816. doi: 10.1109/jsen.2020.3010574

[11] M. T. Sebastian. “Chapter Two – Measurement of Microwave Dielectric Properties and Factors Affecting Them,” in *Dielectric Materials for Wireless Communication*. Elsevier, 2008, ch. 2, pp. 11-47. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780080453309000029

[12] Hydronix, “Hydro-Probe XT”, Hydro-Probe XT - Moisture Sensor for Grain, Animal Feed, Organic, Biomass and Agricultural Materials. https://www.hydronix.com/products/hydroprobe_xt.php (accessed Dec. 1, 2023)

[13] C. Lambert, J. Cartailler, S. Rouchouse, G. Almeida, and F. Courtois. “Characterization and Modeling of Cooling and Drying of Pellets for Animal Feed,” in Drying Technology, vol. 36, no. 3, pp. 255-266. doi: 10.1080/07373937.2017.1323760

[14] E. Nyfors, “Industrial Microwave Sensors – A Review,” in Subsurface Sensing Technologies and Applications, vol. 1, no. 1, Jan. 2000. doi: 10.1023/a:1010118609079

[15] I. B. Benson, “Industrial applications of near infrared reflectance for the measurement of moisture,” in Measurement and Control, vol. 22, no. 2, pp. 45-49, Mar. 1989. doi: 10.1177/002029408902200203

[16] KPM Analytics, “NIR-6000 Series On-Line Moisture Sensors”, NIR-6000 Series On-Line Moisture Sensors | KPM Analytics. https://www.kpmanalytics.com/products/moisture-compositional/nir-6000-series#how_it_works (accessed Dec. 1, 2023)

[17] L. Lin, Y. He, Z. Xiao, K. Zhao, T. Dong, and P. Nie, “Rapid-Detection Sensor for Rice Grain Moisture Based on NIR Spectroscopy,” in Applied Sciences, vol. 9, no. 8, Apr. 2019. doi: 10.3390/app9081654

[18] Soil moisture sensor, by J. E. Walsh. (1985, Sept. 18). US4540936A [Online]. Available: https://patentimages.storage.googleapis.com/bd/7f/59/cc711526930bd4/US4540936.pdf

[19] "3v3 Power". Raspberry Pi Pinout. https://pinout.xyz/pinout/3v3_power (accessed Apr. 7, 2024)

[20] *Visual Studio Code.* (1.85). Microsoft. Available: https://code.visualstudio.com/download

[21] *GitHub.* (2023). Github, Inc. Available: https://github.com/

[22] *PyCharm.* (2023.3). JetBrains. Available: https://www.jetbrains.com/pycharm/

[23] *Python.* (3.12.0). Python Software Foundation. Available: https://www.python.org/

[24] *Postman.* (2023). Postman, Inc. Available: https://www.postman.com/downloads/

[25] *GraphQL.* (2023). The GraphQL Foundation. Available: https://graphql.org/

[26] *Mosquitto.* (2.0.18). Eclipse Foundation. Available: https://mosquitto.org/download/

[27] *InfluxDB.* (2024). InfluxData Inc. Available: https://www.influxdata.com/

[28] *Docker* (2024). Docker Inc. Available: https://www.docker.com/products/docker-desktop/

[29] Adafruit Industries. "About Us". adafruit. https://www.adafruit.com/about

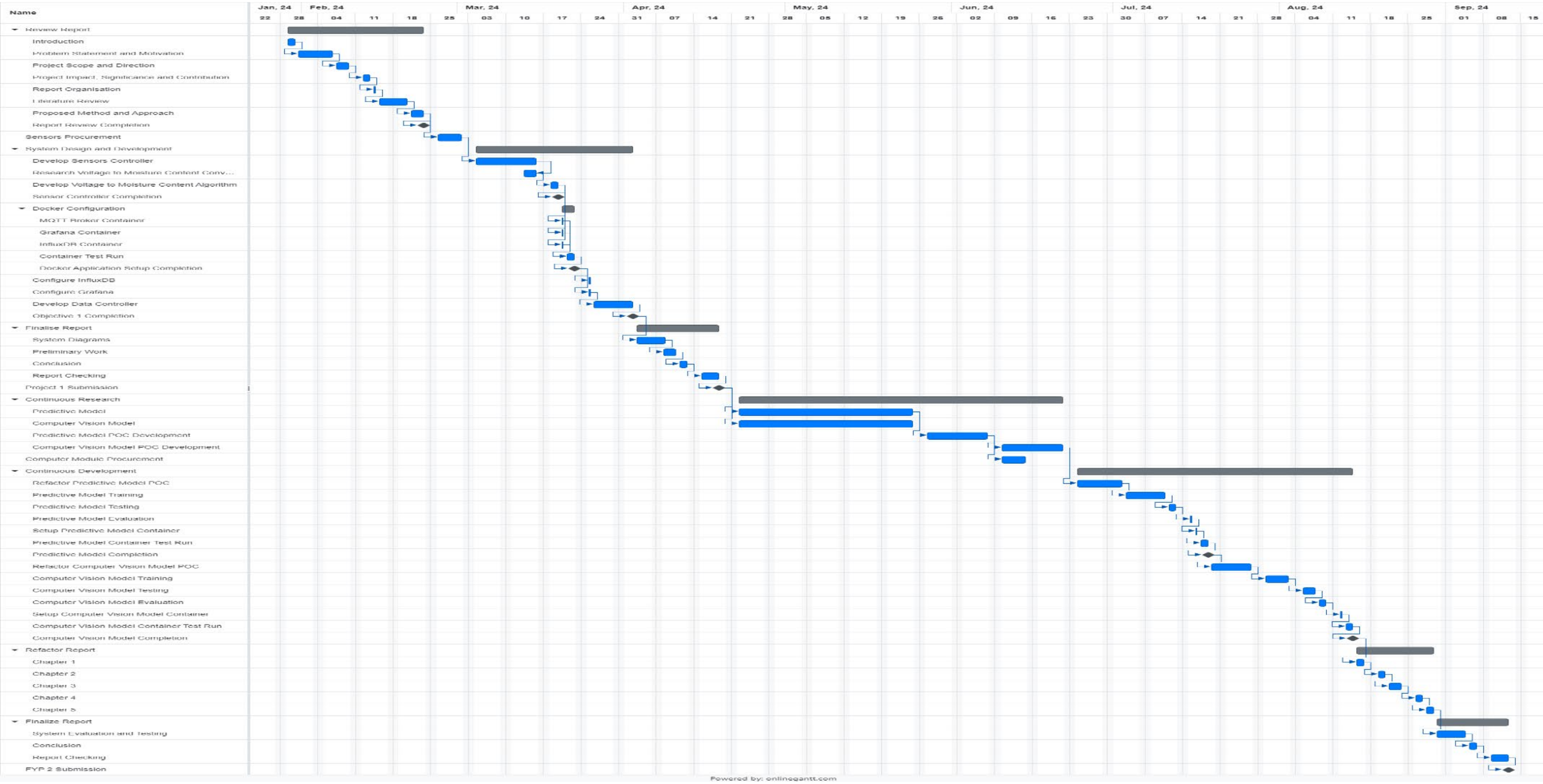
[30] adafruit. *Adafruit_CircuitPython_DHT*. (4.04). GitHub, Inc. Available: https://github.com/adafruit/Adafruit_CircuitPython_DHT

[31] adafruit. *Adafruit_CircuitPython_MCP3xxx*. (1.4.17). GitHub, Inc. Available: https://github.com/adafruit/Adafruit_CircuitPython_MCP3xxx

[32] Python Software Foundation. *multiprocessing – Process-based parallelism.* (2024). Accessed: Apr. 8, 2024. [Online]. Available: https://docs.python.org/3.9/library/multiprocessing.html

APPENDIX

Appendix A

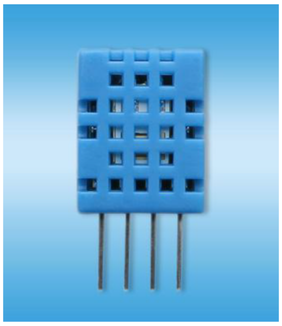


Appendix B

AOSONG

Temperature and Humidity Module

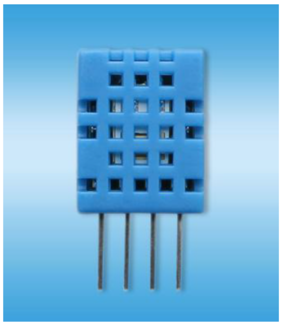
DHT11 Product Manual



For more information, please visit：www.aosong.com

1. Product Overview

DHT11 digital temperature and humidity sensor is a calibrated digital signal output of the temperature and humidity combined sensor. It uses a dedicated digital modules capture technology and the temperature and humidity sensor technology to ensure that products with high reliability and excellent long-term stability. Sensor includes a resistive element and a sense of wet NTC temperature measurement devices, and with a high-performance 8-bit microcontroller connected .



2. Applications

HVAC, dehumidifiers, testing and inspection equipment, consumer goods, automotive, automation, data loggers, weather stations, home appliances, humidity regulator, medical and other relevant humidity measurement and control.

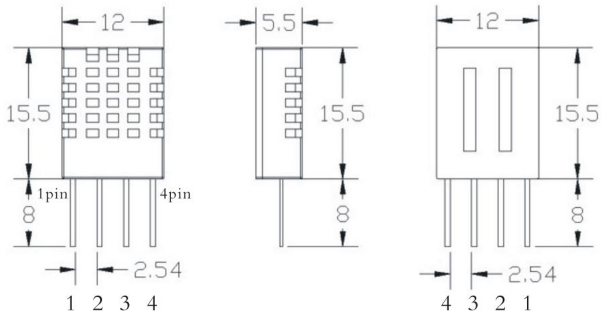
3. Product Highlights

Low-cost, long-term stability, relative humidity and temperature measurement, excellent quality, fast response, anti-interference ability, long distance signal transmission, the digital signal output, precise calibration.

4. Dimensions（Unit：mm）

AOSONG
Aosong Electronics

Temp、Humidity & Dew point measurement experts



5. Parameters

Relative Humidity

Resolution：16Bit

Repeatability：±1%RH

Accuracy：25℃ ±5%RH

Interchangeability：Fully interchangeable

Response time：1/e (63%)25℃ 6s

1m/s Air 6s

Hysteresis：<±0.3%RH

Long-term stability：<±0.5%RH/yr

Temperature

Resolution：16Bit

Repeatability：±1℃

Accuracy：25℃ ±2℃

Response time：1/e (63%) 10S

Electrical Characteristics

Power supply：DC 3.3 ~ 5.5V

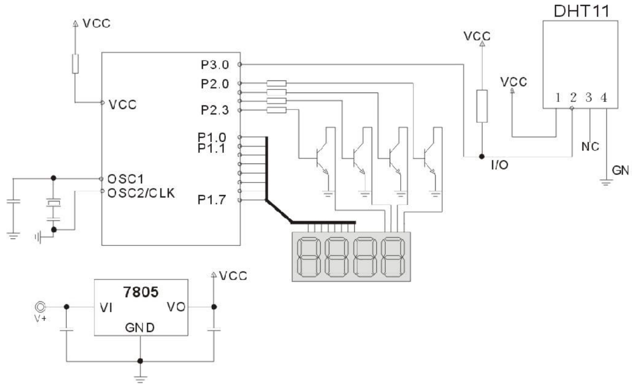
Supply current：Measure 0.3mA Standby 60μA

Sampling period：Secondary Greater than 2 seconds

Pin Description

1. VDD supply 3.3 ~ 5.5V DC
2. DATA serial data, single-bus
3. NC NC
4. GND grounding, power negative

6. Typical circuit



Connecting the typical application circuit shown above the microprocessor and DHT11, DATA pull-up and microprocessor I/O port.

1. A typical application circuit recommended cable length shorter than 20 meters with a 5.1K pull-up resistor when greater than 20 meters when the pull-up resistor to reduce the actual situation.

2. When using a 3.3V voltage supply cable length must not be greater than 100cm. Otherwise it will lead to lack of line drop sensor supply, causing measurement bias.

3. Temperature and humidity values are read out every last measurement result, want to get real-time data, to be read twice in a row, but not recommended repeatedly read sensors, each sensor reading interval of more than 5 seconds to obtain accurate data.

7. Serial Communications Description (single-wire bidirectional)

◎Single Bus Description

DHT11 device uses a simplified single-bus communication. Single bus that only one data line, the data exchange system, are controlled by a single bus is complete. Device (master or slave) through an open-drain or tri-state port is connected to the data line to allow the device to send data when not able to release the bus, and let other devices use the bus; single bus usually requires an external approximately 5.1kΩ pull up resistor, so that when the bus is idle, the state is high. Because they are master-slave structure, only the host calls a slave, a slave to answer, so the host access devices must strictly follow the sequence of a single bus, if there is a sequence of confusion, the device will not respond to the host.

◎Single bus transfer data bit definition

DATA is used for communication between the microprocessor and DHT11 and synchronization, single-bus data format, a 40-bit data transfer, high first-out.
Data formats:
8bit humidity integer data + 8bit decimal data +8 bit temperature and humidity data + 8bit temperature decimal integer data +8 bit parity bit.
Note: The fractional portion wherein the temperature and humidity of 0.

◎Parity bit data definition

"8bit humidity decimal integer data + 8bit humidity temperature data +8 bit decimal integer data + 8bit temperature data" 8bit parity bit is equal to the result of the end of eight.

Example One：40 receives the data to:

0011 0101	0000 0000	0001 1000	0000 0000	0100 1101
High humidity 8	Low humidity 8	High temperature 8	Low temperature 8	Parity bit

Calculated as follows:
0011 0101+0000 0000+0001 1000+0000 0000= 0100 1101
Receive data is correct:
Humidity: 0011 0101=35H=53%RH
Temperature:0001 1000=18H=24℃
Example Two: The received data is 40:

0011 0101	0000 0000	0001 1000	0000 0000	0100 1001
High humidity 8	High humidity 8	High temperature 8	High temperature 8	Parity bit

Calculated as follows:

0011 0101+0000 0000+0001 1000+0000 0000 = 0100 1101

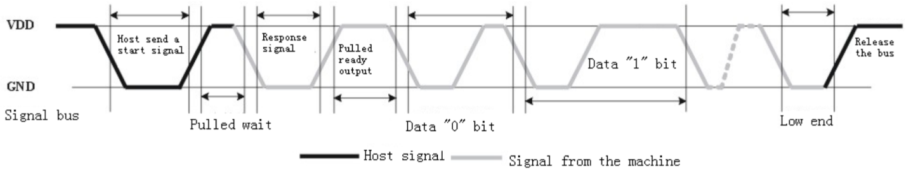
01001001 is not equal to 01001101
The received data is not correct, give up, again receiving data.

◎Data Timing Diagram

Hosts (MCU) after sending a start signal, DHT11 transition from a low-power mode to high-speed mode, the host until after the end of the start signal, DHT11 send a response signal, send 40bit data acquisition and trigger a letter. Signal transmission shown in fig.

AOSONG
Aosong Electronics

Temp、Humidity & Dew point measurement experts



Data Timing Diagram

Note: The host reads temperature and humidity data from DHT11 always previous measurements, such as the two measured time interval is long, please read twice in a row is the second time in real time temperature and humidity values.

◎Peripheral reading step

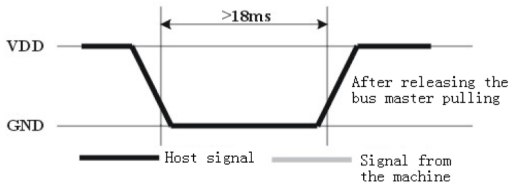
Communication between master and slave can be completed by the following steps (peripherals (such as a microprocessor) to read step DHT11 data).

Step one:

DHT11 after power (power after DHT11 1S to wait to cross the unstable state during this period can't send any commands), test environment temperature and humidity data, and record data while the data lines DATA DHT11 pulled by a pull-up resistor remains high; DHT11 this time the DATA pin is the input state, always detect external signals.

Step two:

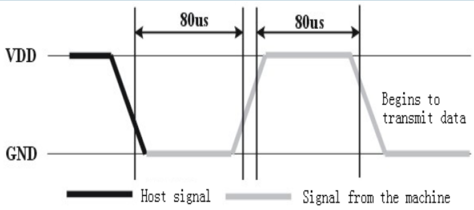
Microprocessor I / O output while the output is set to low, and low retention time can't be less than 18ms, then the microprocessor I / O is set to enter the state, due to the pull-up resistor, the microprocessor I / O that the data lines DHT11 also will go high, waiting to answer DHT11 signals transmitted signal as shown:



The host sends a start signal

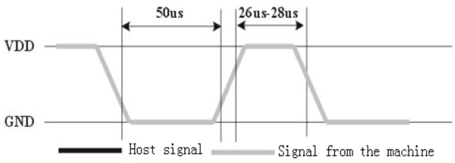
Step three:

DHT11 the DATA pin when external signals detected low, waiting for the external signal low end, after a delay DHT11 the DATA pin is an output, the output low as 80 microseconds response signal, followed by the output of 80 micro-notify the second high peripheral is ready to receive data, the microprocessor I / O at this time in the input state detecting I / O with low (DHT11 echo signal) to the wait for 80 microseconds high data receiving and sending signals as shown:

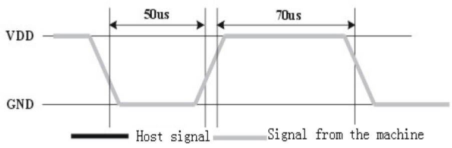


Step four:

The 40 bit data output by the DHT11 DATA pin, the microprocessor according to the change of I/O level receive 40 bits of data, a data format of "0": high level and low level of 50 microseconds and 26-28 microsecond, format data "1": low level 50 microsecond plus 70 microsecond high. Bit data "0", "1" format signal as shown in fig:



Bit data "0" format



Bit data "1" format

End signal:

DHT11 the DATA pin output 40-bit data, the continued output low 50 microseconds after the entry into the state, due to the pull-up resistor attendant goes high. But DHT11 temperature and humidity inside the test-retest data, and record the data, awaiting the arrival of an external signal.

8. Application Information

1. Working and storage conditions

The proposed scope of work may result in up to 3% RH temporary drift of the signal. Return to normal working conditions, the sensor calibration status will slowly recover. To speed up the recovery process can be found in "recovery process." The use of the product will accelerate the aging process for a long time under abnormal operating conditions.

Avoid placing components on a long-term condensation and dry conditions and the following environments.

A. smoke

B. Acid or oxidizing gases such as sulfur dioxide, hydrochloric acid

Recommended Storage Environment

Temperature：10~40℃ Humidity：60% RH or less

2. Effects of exposure to chemical substances

Sensing resistive humidity sensor will be disturbed chemical vapor layer, the diffusion layer in the induction of chemicals may cause drift and measurement sensitivity. In a clean environment, slowly release contaminants out. The recovery process described below to accelerate the process.

High concentrations of chemical pollution can cause damage to the sensor sensing layer completely.

3. Temperature Effect

Relative humidity of the gas is largely dependent on temperature. Therefore, when measuring the humidity should be possible to ensure that the humidity sensor works at the same temperature. If you share a printed circuit board with electronic components heat released in the sensor should be installed as far as possible away from the electronic components, and installed at the bottom of the heat source, while maintaining a well-ventilated enclosure. To reduce the thermal conductivity sensor and a copper plating layer of the printed circuit board should be as minimal other portions, and leaving a gap between them.

4. Light effects

Prolonged exposure to sunlight or strong ultraviolet radiation, will reduce performance.

5. Recovery process

Placed under extreme operating conditions or chemical vapor sensors, through the following process, you can return it to the state calibration. <2 hours (drying) under 10% RH humidity conditions; then at 20-30 ℃ and> 45 ℃ and humidity under 70% RH conditions were maintained for more than 5 hours.

6. Wiring Precautions

DATA signal wire quality will affect the communication distance and communication quality, we recommend using a high-quality shielded cable.

7. Soldering Information

Manual welding, at a temperature of 300 ℃ maximum contact time must be less than 10 seconds.

8. Product upgrades

For details, please consult our technical department。

9. License Agreement

Without the prior written permission of the copyright holder, shall not in any form or by any means, whether electronic or mechanical (including photocopying), for any part of this manual maybe reproduced, nor may its contents be communicated to a third party. The contents are subject to change without notice.

Aosong Electronics Co., Ltd. and third-party ownership of the software, users only be used in the signing of a contract or license the software.

10. Warnings and personal injury

Do not use this product as safety or emergency stop devices, as well as due to the failure of the product could result in personal injuries to any of the other applications. The product shall not apply unless there is a particular purpose or use authorization. Before installation, handling, use or maintenance of the product to the reference product data sheets and application notes. Failure to comply with this recommendation could result in death or serious injury. The resulting company will not be liable for all damages in personal injury and death, and thus exempt from any claims against the company managers and employees and affiliated agents, distributors, etc. that may arise, including: a variety of costs, damages costs, attorney fees, and so on.

11. Quality Assurance

The company offers a three-month quality assurance (calculated from the date of shipment from) their direct purchasers of the product. Data sheet of the company published the technical specifications of the product shall prevail. If the warranty period, the product is proved to be defective quality, the company will provide free repair or replacement. Users must satisfy the following conditions:

① Product is found defective within 14 days written notice to the Company;

② Product should be returned to the purchaser to pay the company;

③ Shelf life of the product should be.

The company only to those used in compliance with the technical condition and defects of products is responsible for. Company for its products used in those special applications without any warranty, guarantee or written statement. The company applied to the product or its products reliability of the circuit does not make any promises.

Appendix C

MICROCHIP

MCP3004/3008

2.7V 4-Channel/8-Channel 10-Bit A/D Converters with SPI Serial Interface

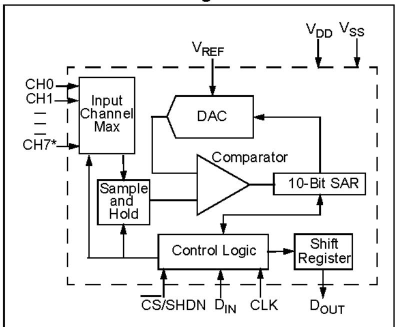
Features

- 10-bit resolution
- ± 1 LSB max DNL
- ± 1 LSB max INL
- 4 (MCP3004) or 8 (MCP3008) input channels
- Analog inputs programmable as single-ended or pseudo-differential pairs
- On-chip sample and hold
- SPI serial interface (modes 0,0 and 1,1)
- Single supply operation: 2.7V - 5.5V
- 200 ksps max. sampling rate at V_{DD} = 5V
- 75 ksps max. sampling rate at V_{DD} = 2.7V
- Low power CMOS technology
- 5 nA typical standby current, 2 µA max.
- 500 µA max. active current at 5V
- Industrial temp range: -40°C to +85°C
- Available in PDIP, SOIC and TSSOP packages

Applications

- Sensor Interface
- Process Control
- Data Acquisition
- Battery Operated Systems

Functional Block Diagram

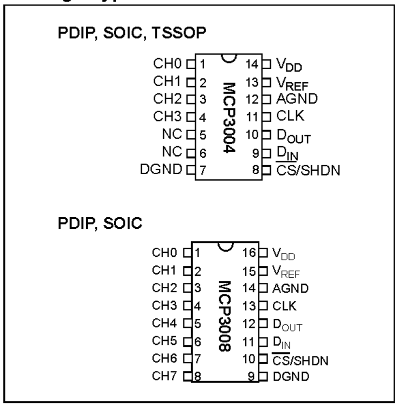


* **Note:** Channels 4-7 are available on MCP3008 Only

Description

The Microchip Technology Inc. MCP3004/3008 devices are successive approximation 10-bit Analog-to-Digital (A/D) converters with on-board sample and hold circuitry. The MCP3004 is programmable to provide two pseudo-differential input pairs or four single-ended inputs. The MCP3008 is programmable to provide four pseudo-differential input pairs or eight single-ended inputs. Differential Nonlinearity (DNL) and Integral Nonlinearity (INL) are specified at ±1 LSB. Communication with the devices is accomplished using a simple serial interface compatible with the SPI protocol. The devices are capable of conversion rates of up to 200 ksps. The MCP3004/3008 devices operate over a broad voltage range (2.7V - 5.5V). Low-current design permits operation with typical standby currents of only 5 nA and typical active currents of 320 µA. The MCP3004 is offered in 14-pin PDIP, 150 mil SOIC and TSSOP packages, while the MCP3008 is offered in 16-pin PDIP and SOIC packages.

Package Types



NOTES:

1.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings †

V_{DD} 7.0V
All Inputs and Outputs w.r.t. V_{SS} ....– 0.6V to V_{DD} + 0.6V
Storage Temperature ....–65°C to +150°C
Ambient temperature with power applied.......–65°C to +150°C
Soldering temperature of leads (10 seconds) ....+300°C
ESD Protection On All Pins (HBM)..................≥ 4 kV

† Notice: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operational listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL SPECIFICATIONS

Electrical Characteristics: Unless otherwise noted, all parameters apply at V_{DD} = 5V, V_{REF} = 5V, T_A = -40°C to +85°C, f_{SAMPLE} = 200 ksps and f_{CLK} = 18*f_{SAMPLE}. Unless otherwise noted, typical values apply for V_{DD} = 5V, T_A = +25°C.

Parameter	Sym	Min	Typ	Max	Units	Conditions
Conversion Rate						
Conversion Time	t_{CONV}	—	—	10	clock cycles	
Analog Input Sample Time	t_{SAMPLE}		1.5		clock cycles	
Throughput Rate	f_{SAMPLE}	—	—	200 75	ksps ksps	V_{DD} = V_{REF} = 5V V_{DD} = V_{REF} = 2.7V
DC Accuracy						
Resolution			10		bits	
Integral Nonlinearity	INL	—	±0.5	±1	LSB	
Differential Nonlinearity	DNL	—	±0.25	±1	LSB	No missing codes over temperature
Offset Error		—	—	±1.5	LSB	
Gain Error		—	—	±1.0	LSB	
Dynamic Performance						
Total Harmonic Distortion		—	-76		dB	V_{IN} = 0.1V to 4.9V@1 kHz
Signal-to-Noise and Distortion (SINAD)		—	61		dB	V_{IN} = 0.1V to 4.9V@1 kHz
Spurious Free Dynamic Range		—	78		dB	V_{IN} = 0.1V to 4.9V@1 kHz
Reference Input						
Voltage Range		0.25	—	V_{DD}	V	Note 2
Current Drain		—	100 0.001	150 3	µA µA	 $\overline{CS}$ = V_{DD} = 5V
Analog Inputs						
Input Voltage Range for CH0 or CH1 in Single-Ended Mode		V_{SS}	—	V_{REF}	V	
Input Voltage Range for IN+ in pseudo-differential mode		IN-	—	V_{REF}+IN-		
Input Voltage Range for IN- in pseudo-differential mode		V_{SS}-100	—	V_{SS}+100	mV	

Note 1: This parameter is established by characterization and not 100% tested.
2: See graphs that relate linearity performance to V_{REF} levels.
3: Because the sample cap will eventually lose charge, effective clock rates below 10 kHz can affect linearity performance, especially at elevated temperatures. See **Section 6.2 “Maintaining Minimum Clock Speed”**, “Maintaining Minimum Clock Speed”, for more information.

ELECTRICAL SPECIFICATIONS (CONTINUED)

Parameter	Sym	Min	Typ	Max	Units	Conditions
Electrical Characteristics: Unless otherwise noted, all parameters apply at V_{DD} = 5V, V_{REF} = 5V, T_A = -40°C to +85°C, f_{SAMPLE} = 200 ksps and f_{CLK} = 18*f_{SAMPLE}. Unless otherwise noted, typical values apply for V_{DD} = 5V, T_A = +25°C.						
Leakage Current		—	0.001	±1	µA	
Switch Resistance		—	1000	—	Ω	See Figure 4-1
Sample Capacitor		—	20	—	pF	See Figure 4-1
Digital Input/Output						
Data Coding Format		Straight Binary				
High Level Input Voltage	V_{IH}	0.7 V_{DD}	—	—	V	
Low Level Input Voltage	V_{IL}		—	0.3 V_{DD}	V	
High Level Output Voltage	V_{OH}	4.1	—	—	V	I_{OH} = -1 mA, V_{DD} = 4.5V
Low Level Output Voltage	V_{OL}	—	—	0.4	V	I_{OL} = 1 mA, V_{DD} = 4.5V
Input Leakage Current	I_{LI}	-10	—	10	µA	V_{IN} = V_{SS} or V_{DD}
Output Leakage Current	I_{LO}	-10	—	10	µA	V_{OUT} = V_{SS} or V_{DD}
Pin Capacitance (All Inputs/Outputs)	C_{IN}, C_{OUT}	—	—	10	pF	V_{DD} = 5.0V (**Note 1**) T_A = 25°C, f = 1 MHz
Timing Parameters						
Clock Frequency	f_{CLK}	—	—	3.6 1.35	MHz MHz	V_{DD} = 5V (**Note 3**) V_{DD} = 2.7V (**Note 3**)
Clock High Time	t_{HI}	125	—	—	ns	
Clock Low Time	t_{LO}	125	—	—	ns	
$\overline{CS}$ Fall To First Rising CLK Edge	t_{SUCS}	100	—	—	ns	
$\overline{CS}$ Fall To Falling CLK Edge	t_{CSD}	—	—	0	ns	
Data Input Setup Time	t_{SU}	50	—	—	ns	
Data Input Hold Time	t_{HD}	50	—	—	ns	
CLK Fall To Output Data Valid	t_{DO}	—	—	125 200	ns ns	V_{DD} = 5V, See Figure 1-2 V_{DD} = 2.7V, See Figure 1-2
CLK Fall To Output Enable	t_{EN}	—	—	125 200	ns ns	V_{DD} = 5V, See Figure 1-2 V_{DD} = 2.7V, See Figure 1-2
$\overline{CS}$ Rise To Output Disable	t_{DIS}	—	—	100	ns	See Test Circuits, Figure 1-2
$\overline{CS}$ Disable Time	t_{CSH}	270	—	—	ns	
D_{OUT} Rise Time	t_R	—	—	100	ns	See Test Circuits, Figure 1-2 (**Note 1**)
D_{OUT} Fall Time	t_F	—	—	100	ns	See Test Circuits, Figure 1-2 (**Note 1**)

Note 1: This parameter is established by characterization and not 100% tested.
2: See graphs that relate linearity performance to V_{REF} levels.
3: Because the sample cap will eventually lose charge, effective clock rates below 10 kHz can affect linearity performance, especially at elevated temperatures. See **Section 6.2 "Maintaining Minimum Clock Speed"**, "Maintaining Minimum Clock Speed", for more information.

ELECTRICAL SPECIFICATIONS (CONTINUED)

Electrical Characteristics: Unless otherwise noted, all parameters apply at V_{DD} = 5V, V_{REF} = 5V, T_A = -40°C to +85°C, f_{SAMPLE} = 200 ksps and f_{CLK} = 18*f_{SAMPLE}. Unless otherwise noted, typical values apply for V_{DD} = 5V, T_A = +25°C.

Parameter	Sym	Min	Typ	Max	Units	Conditions
Power Requirements						
Operating Voltage	V_{DD}	2.7	—	5.5	V	
Operating Current	I_{DD}	—	425 225	550	µA	V_{DD} = V_{REF} = 5V, D_{OUT} unloaded V_{DD} = V_{REF} = 2.7V, D_{OUT} unloaded
Standby Current	I_{DDS}	—	0.005	2	µA	$\overline{CS}$ = V_{DD} = 5.0V

Note 1: This parameter is established by characterization and not 100% tested.
2: See graphs that relate linearity performance to V_{REF} levels.
3: Because the sample cap will eventually lose charge, effective clock rates below 10 kHz can affect linearity performance, especially at elevated temperatures. See **Section 6.2 "Maintaining Minimum Clock Speed"**, "Maintaining Minimum Clock Speed", for more information.

TEMPERATURE CHARACTERISTICS

Electrical Specifications: Unless otherwise indicated, V_{DD} = +2.7V to +5.5V, V_{SS} = GND.

Parameters	Sym	Min	Typ	Max	Units	Conditions
Temperature Ranges						
Specified Temperature Range	T_A	-40	—	+85	°C	
Operating Temperature Range	T_A	-40	—	+85	°C	
Storage Temperature Range	T_A	-65	—	+150	°C	
Thermal Package Resistances						
Thermal Resistance, 14L-PDIP	θ_{JA}	—	70	—	°C/W	
Thermal Resistance, 14L-SOIC	θ_{JA}	—	108	—	°C/W	
Thermal Resistance, 14L-TSSOP	θ_{JA}	—	100	—	°C/W	
Thermal Resistance, 16L-PDIP	θ_{JA}	—	70	—	°C/W	
Thermal Resistance, 16L-SOIC	θ_{JA}	—	90	—	°C/W	

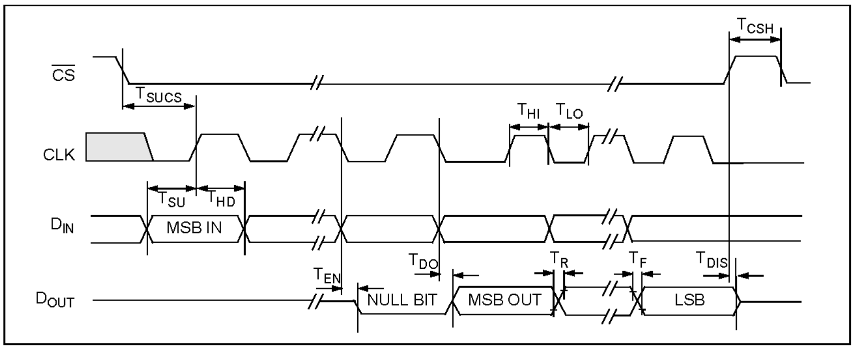


FIGURE 1-1: *Serial Interface Timing.*

MCP3004/3008

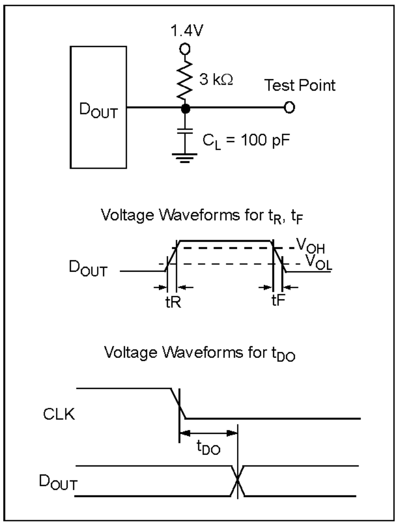


FIGURE 1-2: *Load Circuit for t_R, t_F, t_{DO}.*

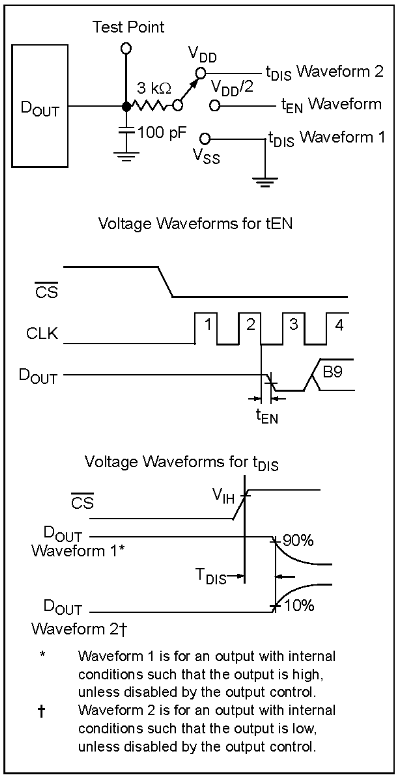


FIGURE 1-3: *Load circuit for t_{DIS} and t_{EN}.*

2.0 TYPICAL PERFORMANCE CHARACTERISTICS

Note:	The graphs and tables provided following this note are a statistical summary based on a limited number of samples and are provided for informational purposes only. The performance characteristics listed herein are not tested or guaranteed. In some graphs or tables, the data presented may be outside the specified operating range (e.g., outside specified power supply range) and therefore outside the warranted range.

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $f_{CLK} = 18^* f_{SAMPLE}$, $T_A = +25°C$.

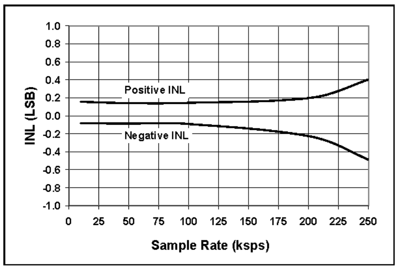


FIGURE 2-1: *Integral Nonlinearity (INL) vs. Sample Rate.*

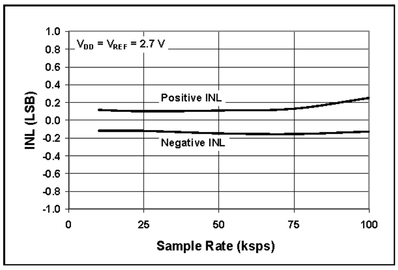


FIGURE 2-4: *Integral Nonlinearity (INL) vs. Sample Rate (V_{DD} = 2.7V).*

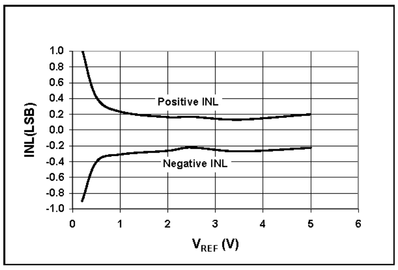


FIGURE 2-2: *Integral Nonlinearity (INL) vs. V_{REF}*

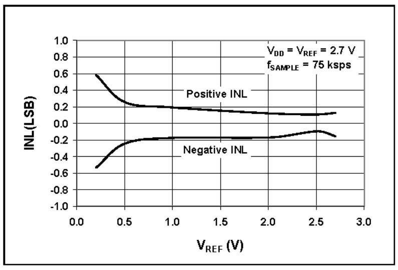


FIGURE 2-5: *Integral Nonlinearity (INL) vs. V_{REF} (V_{DD} = 2.7V).*

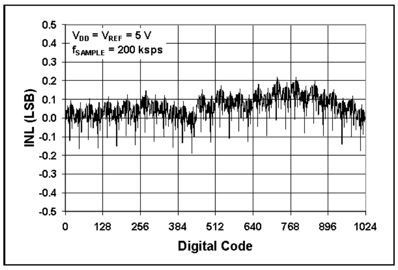


FIGURE 2-3: *Integral Nonlinearity (INL) vs. Code (Representative Part).*

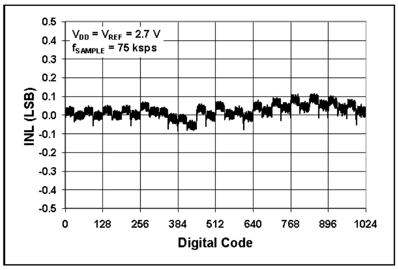


FIGURE 2-6: *Integral Nonlinearity (INL) vs. Code (Representative Part, V_{DD} = 2.7V).*

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $f_{CLK} = 18^* f_{SAMPLE}$, $T_A = +25°C$.

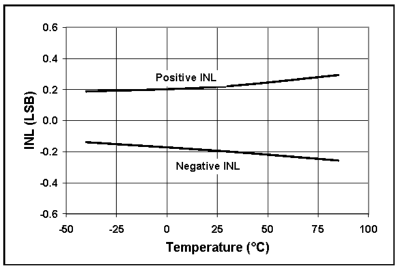


FIGURE 2-7: *Integral Nonlinearity (INL) vs. Temperature.*

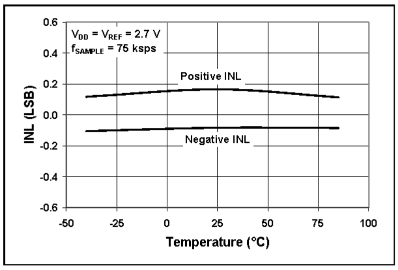


FIGURE 2-10: *Integral Nonlinearity (INL) vs. Temperature (V_{DD} = 2.7V).*

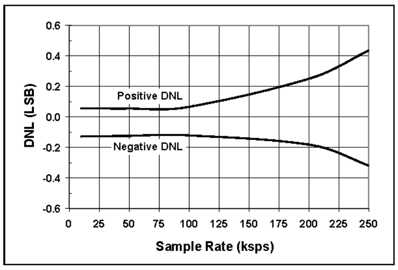


FIGURE 2-8: *Differential Nonlinearity (DNL) vs. Sample Rate.*

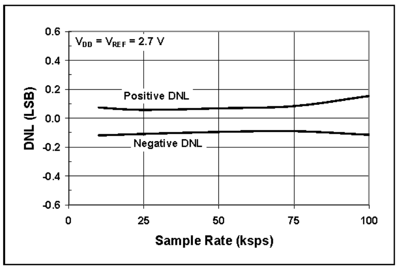


FIGURE 2-11: *Differential Nonlinearity (DNL) vs. Sample Rate (V_{DD} = 2.7V).*

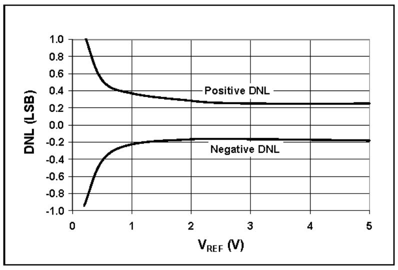


FIGURE 2-9: *Differential Nonlinearity (DNL) vs. V_{REF}*

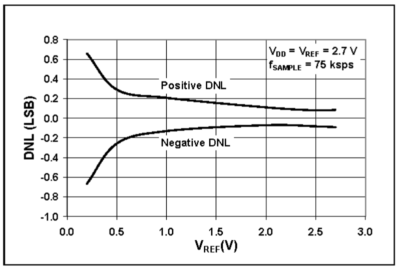


FIGURE 2-12: *Differential Nonlinearity (DNL) vs. V_{REF} (V_{DD} = 2.7V).*

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $f_{CLK} = 18 * f_{SAMPLE}$, $T_A = +25°C$.

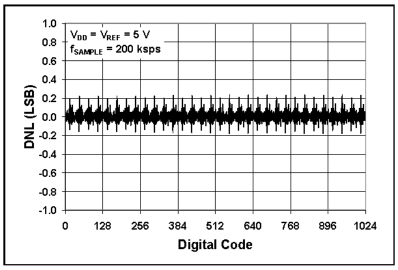


FIGURE 2-13: *Differential Nonlinearity (DNL) vs. Code (Representative Part).*

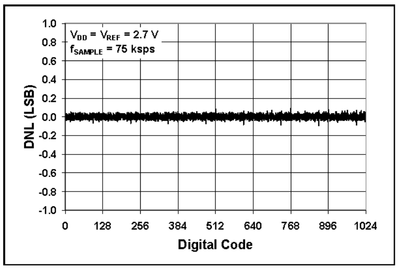


FIGURE 2-16: *Differential Nonlinearity (DNL) vs. Code (Representative Part, V_{DD} =2.7V).*

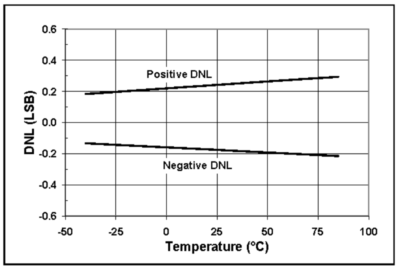


FIGURE 2-14: *Differential Nonlinearity (DNL) vs. Temperature.*

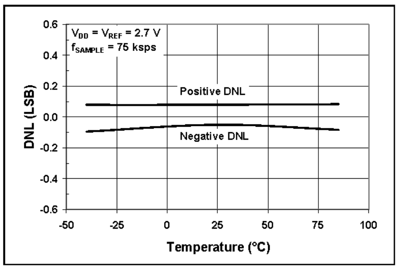


FIGURE 2-17: *Differential Nonlinearity (DNL) vs. Temperature (V_{DD} = 2.7V).*

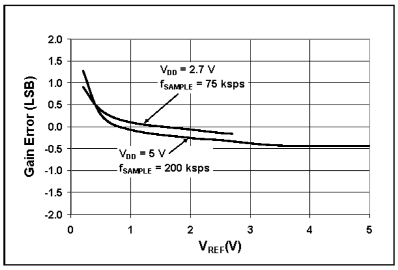


FIGURE 2-15: *Gain Error vs. V_{REF}*

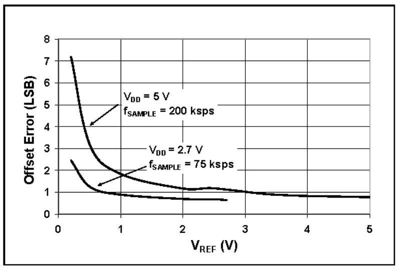


FIGURE 2-18: *Offset Error vs. V_{REF}*

MCP3004/3008

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $f_{CLK} = 18^* f_{SAMPLE}$, $T_A = +25°C$.

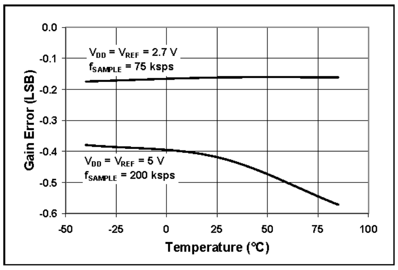


FIGURE 2-19: *Gain Error vs. Temperature.*

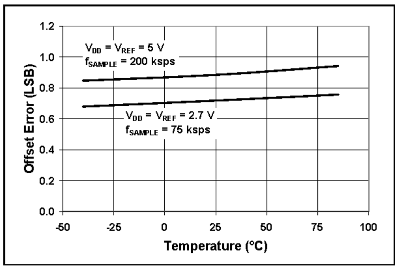


FIGURE 2-22: *Offset Error vs. Temperature.*

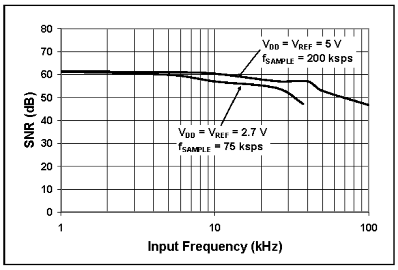


FIGURE 2-20: *Signal-to-Noise (SNR) vs. Input Frequency.*

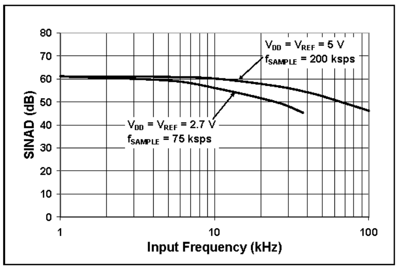


FIGURE 2-23: *Signal-to-Noise and Distortion (SINAD) vs. Input Frequency.*

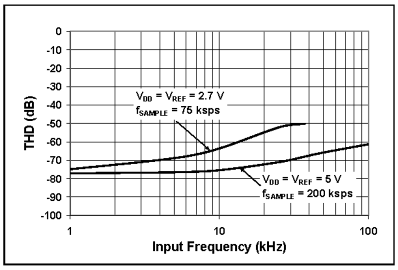


FIGURE 2-21: *Total Harmonic Distortion (THD) vs. Input Frequency.*

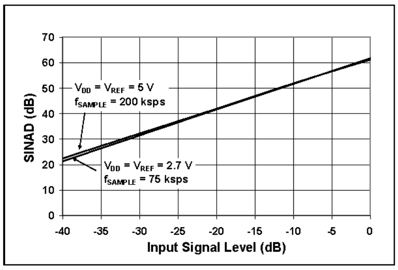


FIGURE 2-24: *Signal-to-Noise and Distortion (SINAD) vs. Input Signal Level.*

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $f_{CLK} = 18* f_{SAMPLE}$, $T_A = +25°C$.

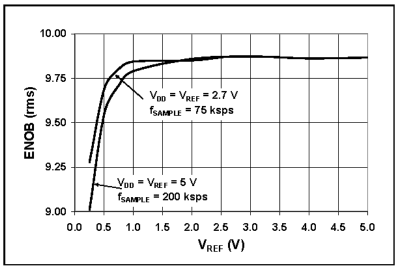


FIGURE 2-25: *Effective Number of Bits (ENOB) vs. V_{REF}*

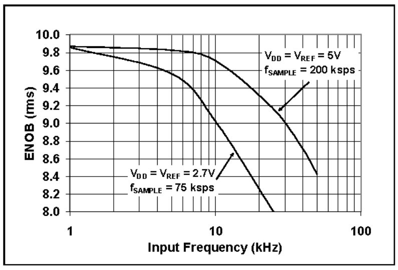


FIGURE 2-28: *Effective Number of Bits (ENOB) vs. Input Frequency.*

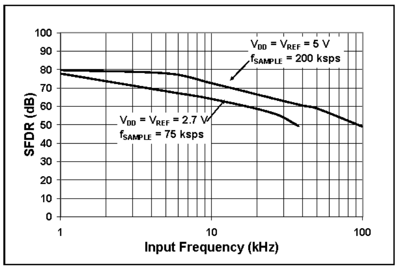


FIGURE 2-26: *Spurious Free Dynamic Range (SFDR) vs. Input Frequency.*

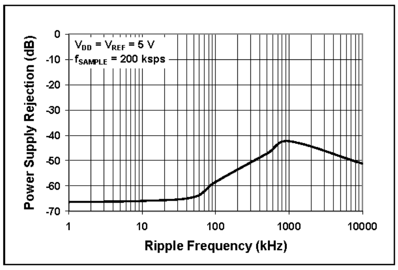


FIGURE 2-29: *Power Supply Rejection (PSR) vs. Ripple Frequency.*

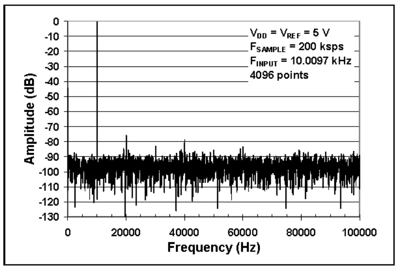


FIGURE 2-27: *Frequency Spectrum of 10 kHz Input (Representative Part).*

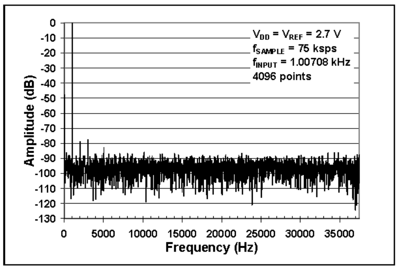


FIGURE 2-30: *Frequency Spectrum of 1 kHz Input (Representative Part, V_{DD} = 2.7V).*

MCP3004/3008

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $f_{CLK} = 18^* f_{SAMPLE}$, $T_A = +25°C$.

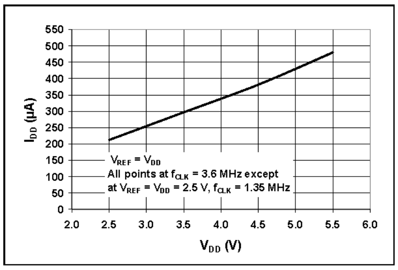


FIGURE 2-31: *I_{DD} vs. V_{DD}.*

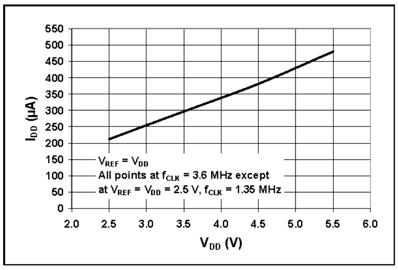


FIGURE 2-34: *I_{REF} vs. V_{DD}.*

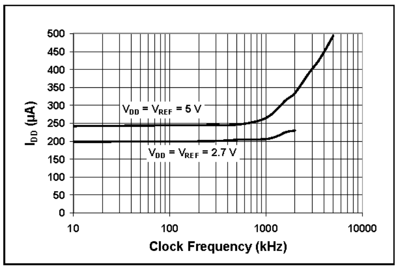


FIGURE 2-32: *I_{DD} vs. Clock Frequency.*

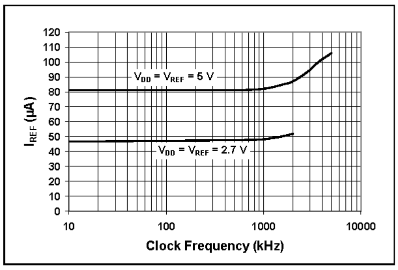


FIGURE 2-35: *I_{REF} vs. Clock Frequency.*

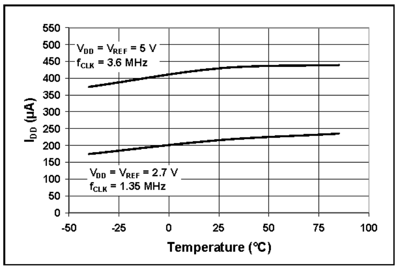


FIGURE 2-33: *I_{DD} vs. Temperature.*

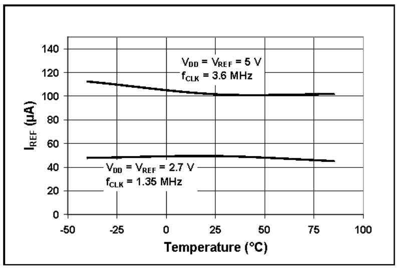


FIGURE 2-36: *I_{REF} vs. Temperature.*

Note: Unless otherwise indicated, $V_{DD} = V_{REF} = 5V$, $f_{CLK} = 18^* f_{SAMPLE}$, $T_A = +25°C$.

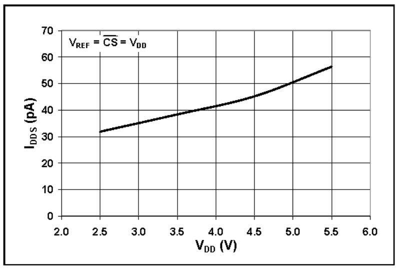


FIGURE 2-37: *I_{DDS} vs. V_{DD}.*

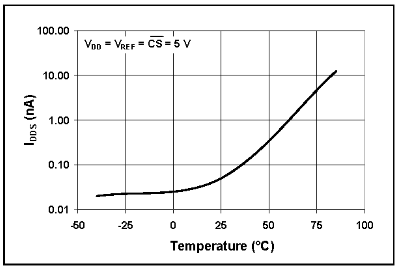


FIGURE 2-38: *I_{DDS} vs. Temperature.*

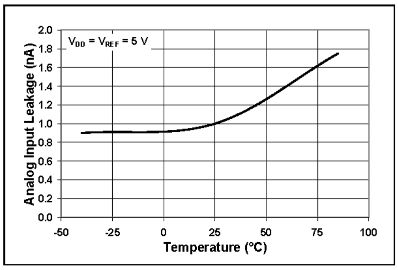


FIGURE 2-39: *Analog Input Leakage Current vs. Temperature.*

NOTES:

3.0 PIN DESCRIPTIONS

The descriptions of the pins are listed in Table 3-1. Additional descriptions of the device pins follows.

TABLE 3-1: PIN FUNCTION TABLE

MCP3004	MCP3008	Symbol	Description
PDIP, SOIC, TSSOP	PDIP, SOIC		
1	1	CH0	Analog Input
2	2	CH1	Analog Input
3	3	CH2	Analog Input
4	4	CH3	Analog Input
–	5	CH4	Analog Input
–	6	CH5	Analog Input
–	7	CH6	Analog Input
–	8	CH7	Analog Input
7	9	DGND	Digital Ground
8	10	$\overline{CS}$/SHDN	Chip Select/Shutdown Input
9	11	D_{IN}	Serial Data In
10	12	D_{OUT}	Serial Data Out
11	13	CLK	Serial Clock
12	14	AGND	Analog Ground
13	15	V_{REF}	Reference Voltage Input
14	16	V_{DD}	+2.7V to 5.5V Power Supply
5,6	–	NC	No Connection

3.1 Digital Ground (DGND)

Digital ground connection to internal digital circuitry.

3.2 Analog Ground (AGND)

Analog ground connection to internal analog circuitry.

3.3 Analog inputs (CH0 - CH7)

Analog inputs for channels 0 - 7, respectively, for the multiplexed inputs. Each pair of channels can be programmed to be used as two independent channels in single-ended mode or as a single pseudo-differential input where one channel is IN+ and one channel is IN. See **Section 4.1 “Analog Inputs”**, “Analog Inputs”, and **Section 5.0 “Serial Communication”**, “Serial Communication”, for information on programming the channel configuration.

3.4 Serial Clock (CLK)

The SPI clock pin is used to initiate a conversion and clock out each bit of the conversion as it takes place. See **Section 6.2 “Maintaining Minimum Clock Speed”**, “Maintaining Minimum Clock Speed”, for constraints on clock speed.

3.5 Serial Data Input (D_{IN})

The SPI port serial data input pin is used to load channel configuration data into the device.

3.6 Serial Data Output (D_{OUT})

The SPI serial data output pin is used to shift out the results of the A/D conversion. Data will always change on the falling edge of each clock as the conversion takes place.

3.7 Chip Select/Shutdown ($\overline{CS}$/SHDN)

The $\overline{CS}$/SHDN pin is used to initiate communication with the device when pulled low. When pulled high, it will end a conversion and put the device in low-power standby. The $\overline{CS}$/SHDN pin must be pulled high between conversions.

MCP3004/3008

NOTES:

4.0 DEVICE OPERATION

The MCP3004/3008 A/D converters employ a conventional SAR architecture. With this architecture, a sample is acquired on an internal sample/hold capacitor for 1.5 clock cycles starting on the first rising edge of the serial clock once $\overline{CS}$ has been pulled low. Following this sample time, the device uses the collected charge on the internal sample and hold capacitor to produce a serial 10-bit digital output code. Conversion rates of 100 ksps are possible on the MCP3004/3008. See **Section 6.2 "Maintaining Minimum Clock Speed"**, "Maintaining Minimum Clock Speed", for information on minimum clock rates. Communication with the device is accomplished using a 4-wire SPI-compatible interface.

4.1 Analog Inputs

The MCP3004/3008 devices offer the choice of using the analog input channels configured as single-ended inputs or pseudo-differential pairs. The MCP3004 can be configured to provide two pseudo-differential input pairs or four single-ended inputs. The MCP3008 can be configured to provide four pseudo-differential input pairs or eight single-ended inputs. Configuration is done as part of the serial command before each conversion begins. When used in the pseudo-differential mode, each channel pair (i.e., CH0 and CH1, CH2 and CH3 etc.) are programmed as the IN+ and IN- inputs as part of the command string transmitted to the device. The IN+ input can range from IN- to (V_{REF} + IN-). The IN- input is limited to ±100 mV from the V_{SS} rail. The IN- input can be used to cancel small signal common-mode noise, which is present on both the IN+ and IN- inputs.

When operating in the pseudo-differential mode, if the voltage level of IN+ is equal to or less than IN-, the resultant code will be `000h`. If the voltage at IN+ is equal to or greater than {[V_{REF} + (IN-)] - 1 LSB}, then the output code will be `3FFh`. If the voltage level at IN- is more than 1 LSB below V_{SS}, the voltage level at the IN+ input will have to go below V_{SS} to see the `000h` output code. Conversely, if IN- is more than 1 LSB above V_{SS}, the `3FFh` code will not be seen unless the IN+ input level goes above V_{REF} level.

For the A/D converter to meet specification, the charge holding capacitor (C_{SAMPLE}) must be given enough time to acquire a 10-bit accurate voltage level during the 1.5 clock cycle sampling period. The analog input model is shown in Figure 4-1.

This diagram illustrates that the source impedance (R_S) adds to the internal sampling switch (R_{SS}) impedance, directly affecting the time that is required to charge the capacitor (C_{SAMPLE}). Consequently, larger source impedances increase the offset, gain and integral linearity errors of the conversion (see Figure 4-2).

4.2 Reference Input

For each device in the family, the reference input (V_{REF}) determines the analog input voltage range. As the reference input is reduced, the LSB size is reduced accordingly.

EQUATION 4-1: LSB SIZE CALCULATION

$$LSB\ Size = \frac{V_{REF}}{1024}$$

The theoretical digital output code produced by the A/D converter is a function of the analog input signal and the reference input, as shown below.

EQUATION 4-2: DIGITAL OUTPUT CODE CALCULATION

$$Digital\ Output\ Code = \frac{1024 \times V_{IN}}{V_{REF}}$$

Where:

V_{IN} = analog input voltage

V_{REF} = analog input voltage

When using an external voltage reference device, the system designer should always refer to the manufacturer's recommendations for circuit layout. Any instability in the operation of the reference device will have a direct effect on the operation of the A/D converter.

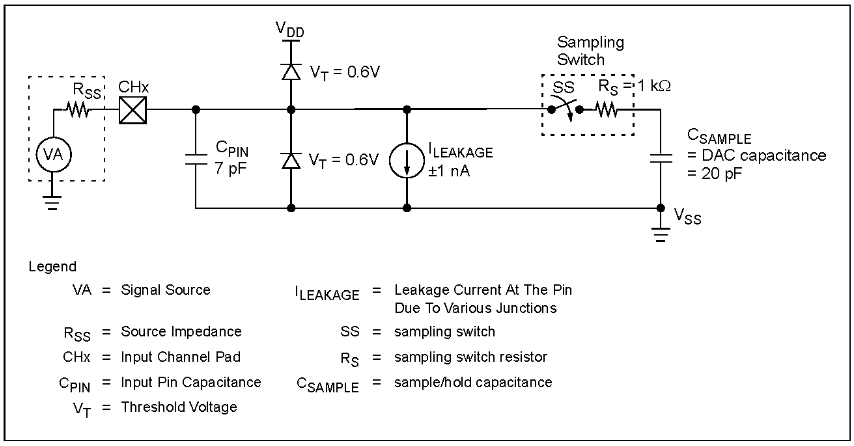


FIGURE 4-1: *Analog Input Model.*

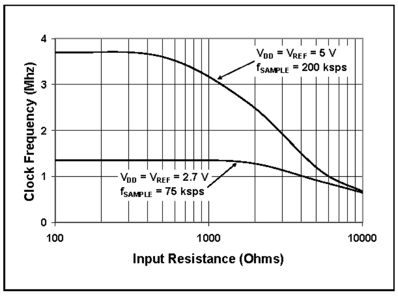


FIGURE 4-2: *Maximum Clock Frequency vs. Input resistance (R_S) to maintain less than a 0.1 LSB deviation in INL from nominal conditions.*

5.0 SERIAL COMMUNICATION

Communication with the MCP3004/3008 devices is accomplished using a standard SPI-compatible serial interface. Initiating communication with either device is done by bringing the $\overline{CS}$ line low (see Figure 5-1). If the device was powered up with the $\overline{CS}$ pin low, it must be brought high and back low to initiate communication. The first clock received with $\overline{CS}$ low and D_{IN} high will constitute a start bit. The SGL/$\overline{DIFF}$ bit follows the start bit and will determine if the conversion will be done using single-ended or differential input mode. The next three bits (D0, D1 and D2) are used to select the input channel configuration. Table 5-1 and Table 5-2 show the configuration bits for the MCP3004 and MCP3008, respectively. The device will begin to sample the analog input on the fourth rising edge of the clock after the start bit has been received. The sample period will end on the falling edge of the fifth clock following the start bit.

Once the D0 bit is input, one more clock is required to complete the sample and hold period (D_{IN} is a "don't care" for this clock). On the falling edge of the next clock, the device will output a low null bit. The next 10 clocks will output the result of the conversion with MSB first, as shown in Figure 5-1. Data is always output from the device on the falling edge of the clock. If all 10 data bits have been transmitted and the device continues to receive clocks while the $\overline{CS}$ is held low, the device will output the conversion result LSB first, as is shown in Figure 5-2. If more clocks are provided to the device while $\overline{CS}$ is still low (after the LSB first data has been transmitted), the device will clock out zeros indefinitely.

If necessary, it is possible to bring $\overline{CS}$ low and clock in leading zeros on the D_{IN} line before the start bit. This is often done when dealing with microcontroller-based SPI ports that must send 8 bits at a time. Refer to **Section 6.1 "Using the MCP3004/3008 with Microcontroller (MCU) SPI Ports"**, "Using the MCP3004/3008 with Microcontroller (MCU) SPI Ports", for more details on using the MCP3004/3008 devices with hardware SPI ports.

TABLE 5-1: CONFIGURE BITS FOR THE MCP3004

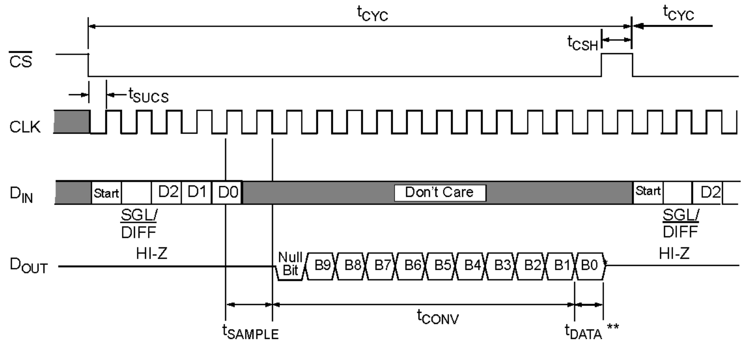
Control Bit Selections				Input Configuration	Channel Selection
Single/Diff	D2*	D1	D0		
1	X	0	0	single-ended	CH0
1	X	0	1	single-ended	CH1
1	X	1	0	single-ended	CH2
1	X	1	1	single-ended	CH3
0	X	0	0	differential	CH0 = IN+ CH1 = IN-
0	X	0	1	differential	CH0 = IN- CH1 = IN+
0	X	1	0	differential	CH2 = IN+ CH3 = IN-
0	X	1	1	differential	CH2 = IN- CH3 = IN+

* D2 is "don't care" for MCP3004

TABLE 5-2: CONFIGURE BITS FOR THE MCP3008

Control Bit Selections				Input Configuration	Channel Selection
Single/Diff	D2	D1	D0		
1	0	0	0	single-ended	CH0
1	0	0	1	single-ended	CH1
1	0	1	0	single-ended	CH2
1	0	1	1	single-ended	CH3
1	1	0	0	single-ended	CH4
1	1	0	1	single-ended	CH5
1	1	1	0	single-ended	CH6
1	1	1	1	single-ended	CH7
0	0	0	0	differential	CH0 = IN+ CH1 = IN-
0	0	0	1	differential	CH0 = IN- CH1 = IN+
0	0	1	0	differential	CH2 = IN+ CH3 = IN-
0	0	1	1	differential	CH2 = IN- CH3 = IN+
0	1	0	0	differential	CH4 = IN+ CH5 = IN-
0	1	0	1	differential	CH4 = IN- CH5 = IN+
0	1	1	0	differential	CH6 = IN+ CH7 = IN-
0	1	1	1	differential	CH6 = IN- CH7 = IN+

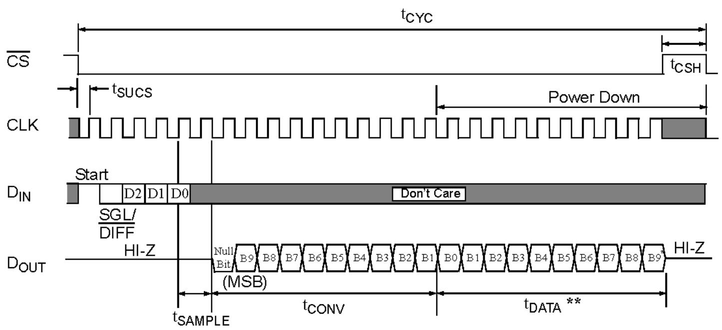
MCP3004/3008



* After completing the data transfer, if further clocks are applied with $\overline{CS}$ low, the A/D converter will output LSB first data, then followed with zeros indefinitely. See Figure 5-2 below.

** t_{DATA}: during this time, the bias current and the comparator powers down while the reference input becomes a high-impedance node.

FIGURE 5-1: *Communication with the MCP3004 or MCP3008.*



* After completing the data transfer, if further clocks are applied with $\overline{CS}$ low, the A/D converter will output zeros indefinitely.

** t_{DATA}: During this time, the bias circuit and the comparator powers down while the reference input becomes a high-impedance node, leaving the CLK running to clock out LSB first data or zeroes.

FIGURE 5-2: *Communication with MCP3004 or MCP3008 in LSB First Format.*

6.0 APPLICATIONS INFORMATION

6.1 Using the MCP3004/3008 with Microcontroller (MCU) SPI Ports

With most microcontroller SPI ports, it is required to send groups of eight bits. It is also required that the microcontroller SPI port be configured to clock out data on the falling edge of clock and latch data in on the rising edge. Because communication with the MCP3004/3008 devices may not need multiples of eight clocks, it will be necessary to provide more clocks than are required. This is usually done by sending 'leading zeros' before the start bit. As an example, Figure 6-1 and Figure 6-2 shows how the MCP3004/3008 can be interfaced to a MCU with a hardware SPI port. Figure 6-1 depicts the operation shown in SPI Mode 0,0, which requires that the SCLK from the MCU idles in the 'low' state, while Figure 6-2 shows the similar case of SPI Mode 1,1, where the clock idles in the 'high' state.

As is shown in Figure 6-1, the first byte transmitted to the A/D converter contains seven leading zeros before the start bit. Arranging the leading zeros this way induces the 10 data bits to fall in positions easily manipulated by the MCU. The MSB is clocked out of the A/D converter on the falling edge of clock number 14. Once the second eight clocks have been sent to the device, the MCU receive buffer will contain five unknown bits (the output is at high-impedance for the first two clocks), the null bit and the highest order 2 bits of the conversion. Once the third byte has been sent to the device, the receive register will contain the lowest order eight bits of the conversion results. Employing this method ensures simpler manipulation of the converted data.

Figure 6-2 shows the same thing in SPI Mode 1,1, which requires that the clock idles in the high state. As with mode 0,0, the A/D converter outputs data on the falling edge of the clock and the MCU latches data from the A/D converter in on the rising edge of the clock.

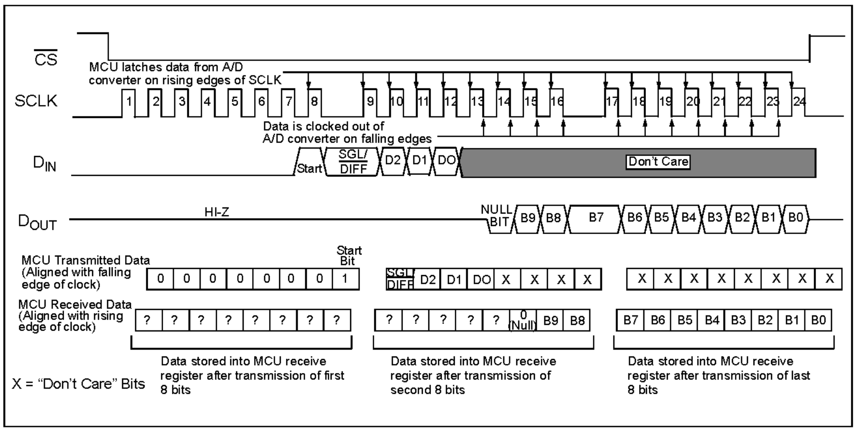


FIGURE 6-1: *SPI Communication with the MCP3004/3008 using 8-bit segments (Mode 0,0: SCLK idles low).*

MCP3004/3008

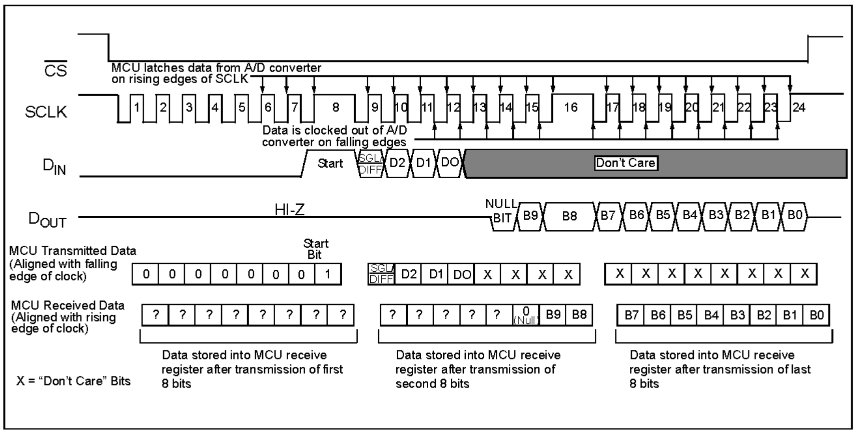


FIGURE 6-2: *SPI Communication with the MCP3004/3008 using 8-bit segments (Mode 1,1: SCLK idles high).*

6.2 Maintaining Minimum Clock Speed

When the MCP3004/3008 initiates the sample period, charge is stored on the sample capacitor. When the sample period is complete, the device converts one bit for each clock that is received. It is important for the user to note that a slow clock rate will allow charge to bleed off the sample capacitor while the conversion is taking place. At 85°C (worst case condition), the part will maintain proper charge on the sample capacitor for at least 1.2 ms after the sample period has ended. This means that the time between the end of the sample period and the time that all 10 data bits have been clocked out must not exceed 1.2 ms (effective clock frequency of 10 kHz). Failure to meet this criterion may introduce linearity errors into the conversion outside the rated specifications. It should be noted that during the entire conversion cycle, the A/D converter does not require a constant clock speed or duty cycle, as long as all timing specifications are met.

6.3 Buffering/Filtering the Analog Inputs

If the signal source for the A/D converter is not a low-impedance source, it will have to be buffered or inaccurate conversion results may occur (see Figure 4-2). It is also recommended that a filter be used to eliminate any signals that may be aliased back in to the conversion results, as is illustrated in Figure 6-3, where an op amp is used to drive, filter and gain the analog input of the MCP3004/3008. This amplifier provides a low-impedance source for the converter input, plus a low-pass filter, which eliminates unwanted high-frequency noise.

Low-pass (anti-aliasing) filters can be designed using Microchip's free interactive FilterLab® software. FilterLab will calculate capacitor and resistors values, as well as determine the number of poles that are required for the application. For more information on filtering signals, see AN699, *"Anti-Aliasing Analog Filters for Data Acquisition Systems"*.

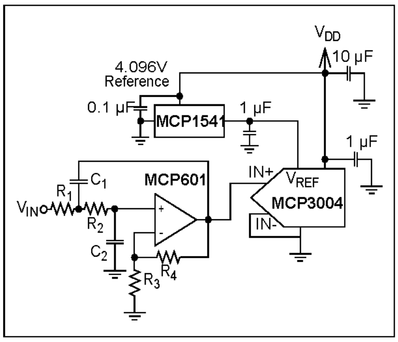


FIGURE 6-3: *The MCP601 Operational Amplifier is used to implement a second order anti-aliasing filter for the signal being converted by the MCP3004.*

6.4 Layout Considerations

When laying out a printed circuit board for use with analog components, care should be taken to reduce noise wherever possible. A bypass capacitor should always be used with this device and should be placed as close as possible to the device pin. A bypass capacitor value of 1 µF is recommended.

Digital and analog traces should be separated as much as possible on the board, with no traces running underneath the device or bypass capacitor. Extra precautions should be taken to keep traces with high-frequency signals (such as clock lines) as far as possible from analog traces.

Use of an analog ground plane is recommended in order to keep the ground potential the same for all devices on the board. Providing V_{DD} connections to devices in a "star" configuration can also reduce noise by eliminating return current paths and associated errors (see Figure 6-4). For more information on layout tips when using A/D converters, refer to AN688, *"Layout Tips for 12-Bit A/D Converter Applications"*.

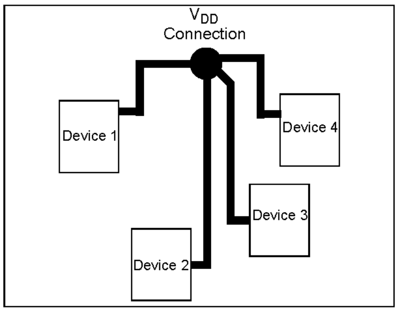


FIGURE 6-4: *V_{DD} traces arranged in a 'Star' configuration in order to reduce errors caused by current return paths.*

6.5 Utilizing the Digital and Analog Ground Pins

The MCP3004/3008 devices provide both digital and analog ground connections to provide additional means of noise reduction. As is shown in Figure 6-5, the analog and digital circuitry is separated internal to the device. This reduces noise from the digital portion of the device being coupled into the analog portion of the device. The two grounds are connected internally through the substrate which has a resistance of 5 -10Ω.

If no ground plane is utilized, both grounds must be connected to V_{SS} on the board. If a ground plane is available, both digital and analog ground pins should be connected to the analog ground plane. If both an analog and a digital ground plane are available, both the digital and the analog ground pins should be connected to the analog ground plane. Following these steps will reduce the amount of digital noise from the rest of the board being coupled into the A/D converter.

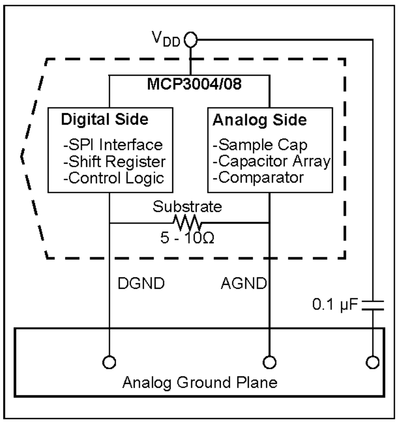


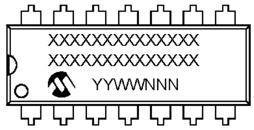
FIGURE 6-5: *Separation of Analog and Digital Ground Pins.*

NOTES:

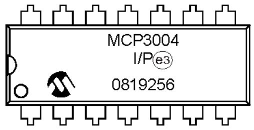
7.0 PACKAGING INFORMATION

7.1 Package Marking Information

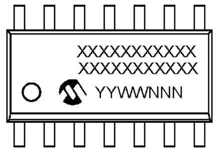
14-Lead PDIP (300 mil)



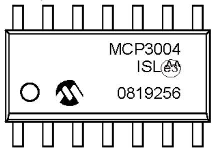
Example:



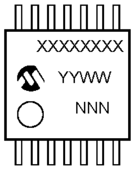
14-Lead SOIC (150 mil)



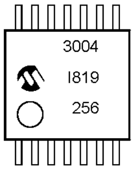
Example:



14-Lead TSSOP (4.4mm) *



Example:



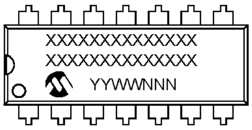
Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package.

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

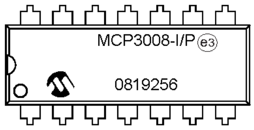
MCP3004/3008

Package Marking Information (Continued)

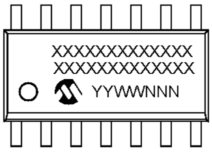
16-Lead PDIP (300 mil) (**MCP3008**)



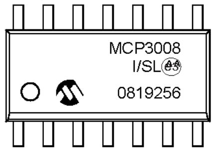
Example:



16-Lead SOIC (150 mil) (**MCP3008**)

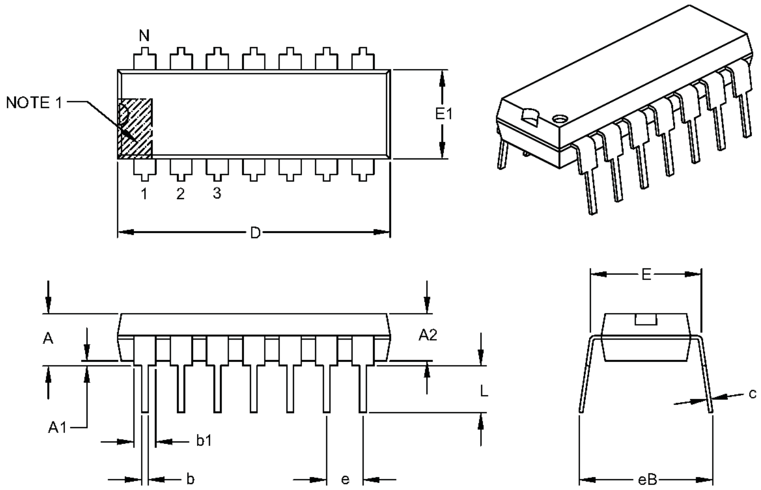


Example:



14-Lead Plastic Dual In-Line (P) – 300 mil Body [PDIP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		INCHES		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		14	
Pitch	e		.100 BSC	
Top to Seating Plane	A	–	–	.210
Molded Package Thickness	A2	.115	.130	.195
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.290	.310	.325
Molded Package Width	E1	.240	.250	.280
Overall Length	D	.735	.750	.775
Tip to Seating Plane	L	.115	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.045	.060	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

Notes:

1. Pin 1 visual index feature may vary, but must be located with the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

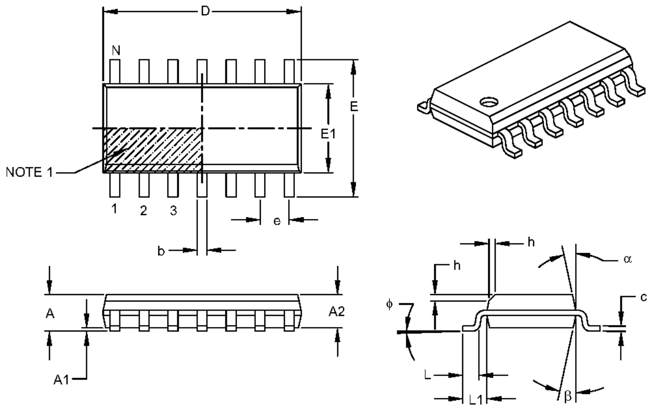
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-005B

MCP3004/3008

14-Lead Plastic Small Outline (SL) – Narrow, 3.90 mm Body [SOIC]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



		Units	MILLIMETERS	
	Dimension Limits	MIN	NOM	MAX
Number of Pins	N		14	
Pitch	e		1.27 BSC	
Overall Height	A	–	–	1.75
Molded Package Thickness	A2	1.25	–	–
Standoff §	A1	0.10	–	0.25
Overall Width	E		6.00 BSC	
Molded Package Width	E1		3.90 BSC	
Overall Length	D		8.65 BSC	
Chamfer (optional)	h	0.25	–	0.50
Foot Length	L	0.40	–	1.27
Footprint	L1		1.04 REF	
Foot Angle	φ	0°	–	8°
Lead Thickness	c	0.17	–	0.25
Lead Width	b	0.31	–	0.51
Mold Draft Angle Top	α	5°	–	15°
Mold Draft Angle Bottom	β	5°	–	15°

Notes:

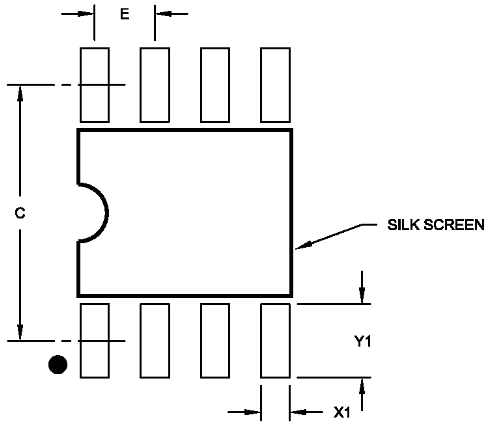
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-065B

8-Lead Plastic Small Outline (SN) – Narrow, 3.90 mm Body [SOIC]

Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		1.27 BSC	
Contact Pad Spacing	C		5.40	
Contact Pad Width (X8)	X1			0.60
Contact Pad Length (X8)	Y1			1.55

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

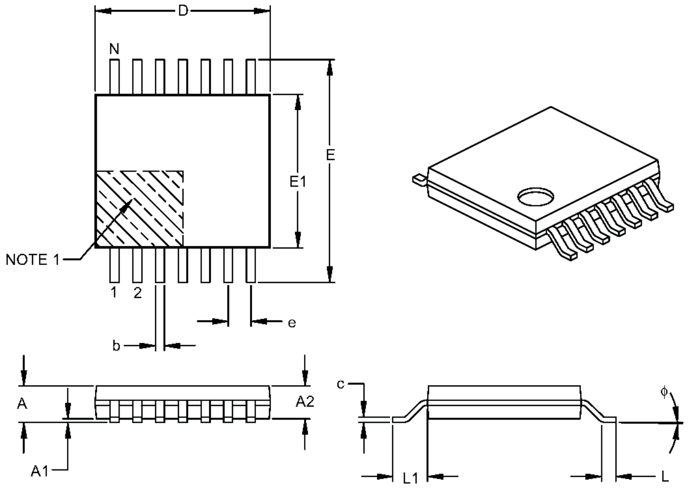
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2057A

MCP3004/3008

14-Lead Plastic Thin Shrink Small Outline (ST) – 4.4 mm Body [TSSOP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



	Units	MILLIMETERS		
	Dimension Limits	MIN	NOM	MAX
Number of Pins	N		14	
Pitch	e		0.65 BSC	
Overall Height	A	–	–	1.20
Molded Package Thickness	A2	0.80	1.00	1.05
Standoff	A1	0.05	–	0.15
Overall Width	E		6.40 BSC	
Molded Package Width	E1	4.30	4.40	4.50
Molded Package Length	D	4.90	5.00	5.10
Foot Length	L	0.45	0.60	0.75
Footprint	L1		1.00 REF	
Foot Angle	φ	0°	–	8°
Lead Thickness	c	0.09	–	0.20
Lead Width	b	0.19	–	0.30

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15 mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M.

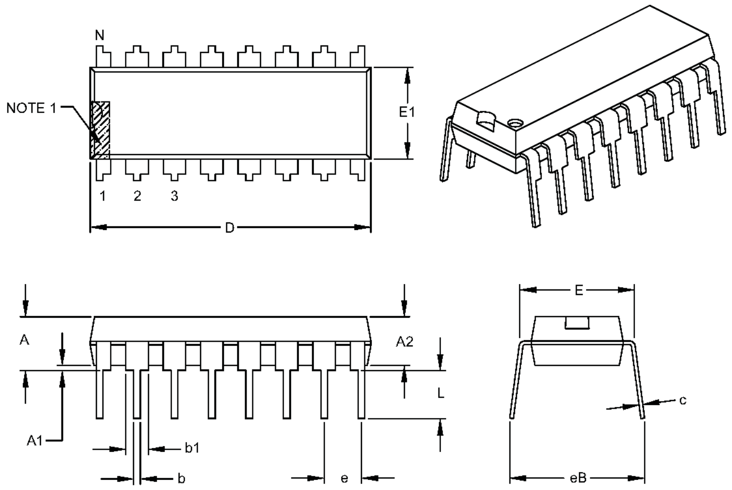
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-087B

16-Lead Plastic Dual In-Line (P) – 300 mil Body [PDIP]

Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



		Units	INCHES		
		Dimension Limits	MIN	NOM	MAX
Number of Pins	N			16	
Pitch	e			.100 BSC	
Top to Seating Plane	A		–	–	.210
Molded Package Thickness	A2		.115	.130	.195
Base to Seating Plane	A1		.015	–	–
Shoulder to Shoulder Width	E		.290	.310	.325
Molded Package Width	E1		.240	.250	.280
Overall Length	D		.735	.755	.775
Tip to Seating Plane	L		.115	.130	.150
Lead Thickness	c		.008	.010	.015
Upper Lead Width	b1		.045	.060	.070
Lower Lead Width	b		.014	.018	.022
Overall Row Spacing §	eB		–	–	.430

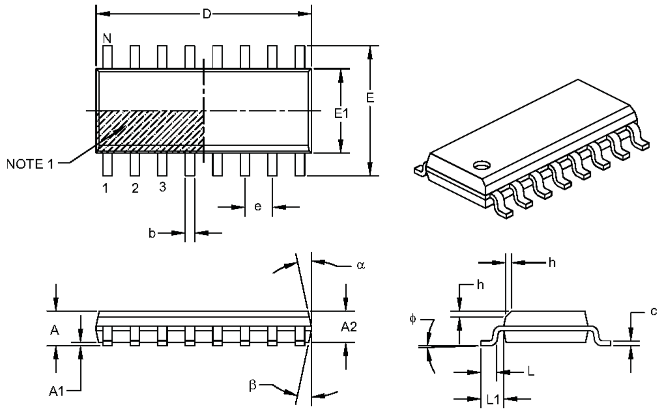
Notes:
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-017B

MCP3004/3008

16-Lead Plastic Small Outline (SL) – Narrow, 3.90 mm Body [SOIC]

Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



	Units	MILLIMETERS		
	Dimension Limits	MIN	NOM	MAX
Number of Pins	N		16	
Pitch	e		1.27 BSC	
Overall Height	A	–	–	1.75
Molded Package Thickness	A2	1.25	–	–
Standoff §	A1	0.10	–	0.25
Overall Width	E		6.00 BSC	
Molded Package Width	E1		3.90 BSC	
Overall Length	D		9.90 BSC	
Chamfer (optional)	h	0.25	–	0.50
Foot Length	L	0.40	–	1.27
Footprint	L1		1.04 REF	
Foot Angle	φ	0°	–	8°
Lead Thickness	c	0.17	–	0.25
Lead Width	b	0.31	–	0.51
Mold Draft Angle Top	α	5°	–	15°
Mold Draft Angle Bottom	β	5°	–	15°

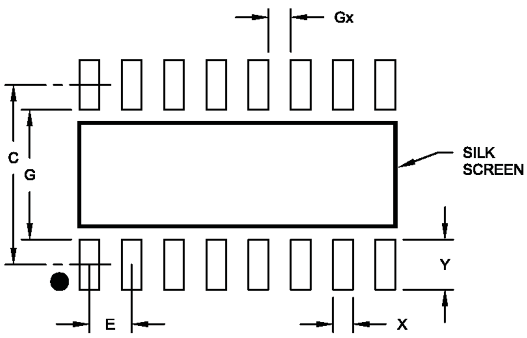
Notes:
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-108B

16-Lead Plastic Small Outline (SL) - Narrow, 3.90 mm Body [SOIC]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		1.27 BSC	
Contact Pad Spacing	C		5.40	
Contact Pad Width	X			0.60
Contact Pad Length	Y			1.50
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	3.90		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2108A

NOTES:

APPENDIX A: REVISION HISTORY

Revision D (December 2008)

The following is the list of modifications:

1. Updates to **Section 7.0 "Packaging Information"**.

Revision C (January 2007)

The following is the list of modifications:

1. Updates to the packaging diagrams.

Revision B (May 2002)

The following is the list of modifications:

1. Undocumented changes.

Revision A (February 2000)

- Initial release of this document.

NOTES:

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	X	/XX
Device	Temperature Range	Package

Device	MCP3004:	4-Channel 10-Bit Serial A/D Converter
	MCP3004T:	4-Channel 10-Bit Serial A/D Converter (Tape and Reel)
	MCP3008:	8-Channel 10-Bit Serial A/D Converter
	MCP3008T:	8-Channel 10-Bit Serial A/D Converter (Tape and Reel)
Temperature Range	I =	-40°C to +85°C (Industrial)
Package	P =	Plastic DIP (300 mil Body), 14-lead, 16-lead
	SL =	Plastic SOIC (150 mil Body), 14-lead, 16-lead
	ST =	Plastic TSSOP (4.4mm), 14-lead

Examples:

a) MCP3004-I/P: Industrial Temperature, PDIP package.
b) MCP3004-I/SL: Industrial Temperature, SOIC package.
c) MCP3004-I/ST: Industrial Temperature, TSSOP package.
d) MCP3004T-I/ST: Industrial Temperature, TSSOP package, Tape and Reel.

a) MCP3008-I/P: Industrial Temperature, PDIP package.
b) MCP3008-I/SL: Industrial Temperature, SOIC package.

NOTES:

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949:2002 ==

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

MICROCHIP

WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://support.microchip.com
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario, Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-4182-8400
Fax: 91-80-4182-8422

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or 82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-572-9526
Fax: 886-3-572-6459

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

01/02/08

Appendix D

Capacitive Soil Moisture Sensor SKU:SEN0193

Contents

(https://www.dfrobot.com/product-1385.html)
Capacitive Soil Moisture Sensor
(https://www.dfrobot.com/product-1385.html)

Introduction

Our **soil moisture sensor** (https://www.dfrobot.com/product-1385.html) measures soil mositure levels by capacitive sensing rather than resistive sensing like other sensors on the market. It is made of corrosion resistant material which gives it an excellent service life.
Insert it in to the soil around your plants and impress your friends with real-time soil moisture data!
This module includes an on-board voltage regulator which gives it an operating voltage range of 3.3 ~ 5.5V. It is perfect for low-voltage MCUs, both 3.3V and 5V. For compatibility with a Raspberry Pi it will need an ADC converter.
This soil moisture sensor is compatible with our 3-pin "Gravity" interface, which can be directly connected to the Gravity I/O expansion shield.

Specification

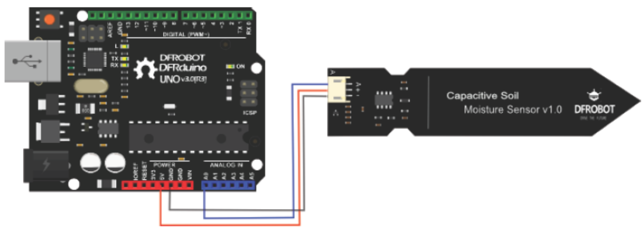
- Operating Voltage: 3.3 ~ 5.5 VDC
- Output Voltage: 0 ~ 3.0VDC
- Operating Current: 5mA
- Interface: PH2.0-3P
- Dimensions: 3.86 x 0.905 inches (L x W)
- Weight: 15g

Tutorial

Requirements

- **Hardware**
  - DFRduino UNO x1
  - Capacitive Soil Moisture Sensor x1
  - Jumper Cable x3
- **Software**
  - Arduino IDE V1.6.5 Click to Download Arduino IDE (https://www.arduino.cc/en/Main/Software)

Connection Diagram



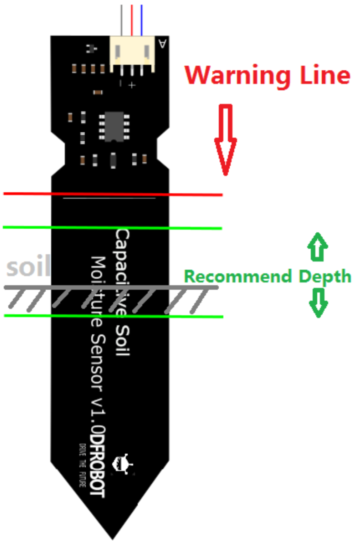
(/wiki/index.php/File:SEN0193-line.png)

Calibration Code

```
void setup() {
  Serial.begin(9600); // open serial port, set the baud rate as 9600 bps
}
void loop() {
  int val;
  val = analogRead(0); //connect sensor to Analog 0
  Serial.print(val); //print the value to serial port
  delay(100);
}
```

Calibration

Calibration Range



(/wiki/index.php/File:SEN0193-test.png)

1. Open the serial port monitor and set the baud rate to 9600
2. Record the sensor value when the probe is exposed to the air as "Value 1". This is the boundary value of dry soil "Humidity: 0%RH"
3. Take a cup of water and insert the probe into it no further than the red line in the diagram
4. Record the sensor value when the probe is exposed to the water as "Value 2". This is the boundary value of moist soil "Humidity: 100%RH"

(/wiki/index.php/File:Warning_yellow.png)

- The components on this board are NOT waterproof, do not expose to moisture further than the red line. (If you want to protect components from the elements, try using a length of wide heat shrink tubing around the upper-section of the board.)
- There is an inverse ratio between the sensor output value and soil moisture.

Section Settings

The final output value is affected by probe insertion depth and how tight the soil packed around it is. We regard "value_1" as dry soil and "value_2" as soaked soil. This is the sensor detection range.
For example: Value_1 = 520; Value_2 = 260.
The range will be divided into three sections: dry, wet, water. Their related values are:

- Dry: (520 430]
- Wet: (430 350]
- Water: (350 260]

Test Code

```
/***************************************************
 This example reads Capacitive Soil Moisture Sensor.

 Created 2015-10-21
 By berinie Chen <bernie.chen@dfrobot.com>

 GNU Lesser General Public License.
 See <http://www.gnu.org/licenses/> for details.
 All above must be included in any redistribution
 ****************************************************/

/***********Notice and Trouble shooting***************
 1.Connection and Diagram can be found here: https://www.dfrobot.com/wiki/index.php?title=Capacitive_Soil_Moisture_Sensor_SKU:SEN0193
 2.This code is tested on Arduino Uno.
 3.Sensor is connect to Analog 0 port.
 ****************************************************/

const int AirValue = 520;   //you need to replace this value with Value_1
const int WaterValue = 260;  //you need to replace this value with Value_2
int intervals = (AirValue - WaterValue)/3;
int soilMoistureValue = 0;
void setup() {
  Serial.begin(9600); // open serial port, set the baud rate to 9600 bps
}
void loop() {
soilMoistureValue = analogRead(A0);  //put Sensor insert into soil
if(soilMoistureValue > WaterValue && soilMoistureValue < (WaterValue + intervals))
{
  Serial.println("Very Wet");
}
else if(soilMoistureValue > (WaterValue + intervals) && soilMoistureValue < (AirValue - intervals))
{
  Serial.println("Wet");
}
else if(soilMoistureValue < AirValue && soilMoistureValue > (AirValue - intervals))
{
  Serial.println("Dry");
}
delay(100);
}
```

FAQ

There are no questions about this product yet.

For any questions/advice/cool ideas to share with us, please visit **DFRobot Forum** (http://www.dfrobot.com/forum/).

More

- **Documents**
  - Schematic (https://github.com/Arduinolibrary/DFRobot_Capacitive_Soil_Moisture_Sensor/raw/master/SEN0193%20%20Capacitive%20Soil%20Moisture%20SensorV1.0.PDF)
  - Product SVGs (https://github.com/Arduinolibrary/DFRobot_Capacitive_Soil_Moisture_Sensor/raw/master/SEN0193.zip)
- **Share**
  - Relative humidity to absolute humidity calculator (http://planetcalc.com/2167/)

(http://www.dfrobot.com/index.php?route=product/product&product_id=1385&search=sen0193&description=true#.VnJZsvmqqAw) Get it from Gravity: Analog Capacitive Soil Moisture Sensor- Corrosion Resistant (https://www.dfrobot.com/product-1385.html) or **DFRobot Distributor** (http://www.dfrobot.com/index.php?route=information/distributorslogo).

Category: DFRobot (https://www.dfrobot.com/) > Sensors &Modules (https://www.dfrobot.com/category-156.html) >Sensors (https://www.dfrobot.com/category-36.html) > Temperature & Humidity Sensor (https://www.dfrobot.com/category-64.html)

APPENDIX

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3Y3	**Study week no.:1**
Student Name & ID: Chong Shao Yang 20ACB05213	
Supervisor: Dr. Aun Yichiet	
Project Title: A Homeostatic Approach to Adaptive Ambient Control in Smart Factories	

1. WORK DONE Research on various types of predictive machines learning modelling techniques available.
2. WORK TO BE DONE Research on computer visions machines learning modelling techniques available.
3. PROBLEMS ENCOUNTERED No problem encountered
4. SELF EVALUATION OF THE PROGRESS Need to put more effort in learning as much model as possible, current progress so far so good.

Supervisor's signature

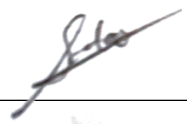
Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3Y3	**Study week no.:2**
Student Name & ID: Chong Shao Yang 20ACB05213	
Supervisor: Dr. Aun Yichiet	
Project Title: A Homeostatic Approach to Adaptive Ambient Control in Smart Factories	

1. WORK DONE
Research on various types of predictive machines learning modelling techniques available
2. WORK TO BE DONE
Determine and acquire data that needed for the project
3. PROBLEMS ENCOUNTERED
Not sure on the features that required as the input for the model
4. SELF EVALUATION OF THE PROGRESS
Slow progress but need to keep it up. Need to concerned of the task and time management



Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3Y3	**Study week no.:3**
Student Name & ID: Chong Shao Yang 20ACB05213	
Supervisor: Dr. Aun Yichiet	
Project Title: A Homeostatic Approach to Adaptive Ambient Control in Smart Factories	

1. WORK DONE
Reached out to third party providers to acquire data
2. WORK TO BE DONE
Continue study machine learning model that could be useful for the application
3. PROBLEMS ENCOUNTERED
No problem so far
4. SELF EVALUATION OF THE PROGRESS
Good progress, need to keep it up

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3Y3	**Study week no.:4**
Student Name & ID: Chong Shao Yang 20ACB05213	
Supervisor: Dr. Aun Yichiet	
Project Title: A Homeostatic Approach to Adaptive Ambient Control in Smart Factories	

1. WORK DONE
Continued studying the machine learning models for computer vision while waiting for response.
2. WORK TO BE DONE
Need to get the data for working on the model
3. PROBLEMS ENCOUNTERED
No problem so far
4. SELF EVALUATION OF THE PROGRESS
Good progress, still need to wait for the data to continue

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3Y3	**Study week no.:5**
Student Name & ID: Chong Shao Yang 20ACB05213	
Supervisor: Dr. Aun Yichiet	
Project Title: A Homeostatic Approach to Adaptive Ambient Control in Smart Factories	

1. WORK DONE Worked on the sensor controller in Raspberry Pi since some library is deprecated
2. WORK TO BE DONE Need to complete the python program before next week
3. PROBLEMS ENCOUNTERED The new library is unstable for third party DHT sensor, but the old one has been archived and no support and further development nor it can be used anymore
4. SELF EVALUATION OF THE PROGRESS Need to keep up the work, otherwise it will be in a rush

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3Y3	**Study week no.:6**
Student Name & ID: Chong Shao Yang 20ACB05213	
Supervisor: Dr. Aun Yichiet	
Project Title: A Homeostatic Approach to Adaptive Ambient Control in Smart Factories	

1. WORK DONE Completed rewrote the sensor controller
2. WORK TO BE DONE Purchase a camera module and write a camera controller
3. PROBLEMS ENCOUNTERED No problem so far
4. SELF EVALUATION OF THE PROGRESS Need to keep up the work.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3Y3	**Study week no.:7**
Student Name & ID: Chong Shao Yang 20ACB05213	
Supervisor: Dr. Aun Yichiet	
Project Title: A Homeostatic Approach to Adaptive Ambient Control in Smart Factories	

1. WORK DONE Nothing has done this week since its midterm week and still waiting for data and camera
2. WORK TO BE DONE Need to complete the camera setup and file server setup next week
3. PROBLEMS ENCOUNTERED None
4. SELF EVALUATION OF THE PROGRESS Next week need to be working faster

_________________________ _________________________

Supervisor's signature Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3Y3	**Study week no.:8**
Student Name & ID: Chong Shao Yang 20ACB05213	
Supervisor: Dr. Aun Yichiet	
Project Title: A Homeostatic Approach to Adaptive Ambient Control in Smart Factories	

1. WORK DONE
Camera is acquired and camera controller is half way implemented to the Raspberry Pi due to some technical issue on the camera itself
2. WORK TO BE DONE
Need to resolve the issue
3. PROBLEMS ENCOUNTERED
The camera does not get detected by the Raspberry Pi
4. SELF EVALUATION OF THE PROGRESS
Need to hurry up and figure out the issue, it might take a lot of time for modelling the machine learning model.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3Y3	**Study week no.:9**
Student Name & ID: Chong Shao Yang 20ACB05213	
Supervisor: Dr. Aun Yichiet	
Project Title: A Homeostatic Approach to Adaptive Ambient Control in Smart Factories	

1. WORK DONE
Data has been acquired and started with the data exploration
2. WORK TO BE DONE
Camera at Raspberry Pi still need to be fixed, and need to start the modelling the model soon.
3. PROBLEMS ENCOUNTERED
The camera is still not working.
4. SELF EVALUATION OF THE PROGRESS
So far so good still need to be quick with the work.

Supervisor's signature

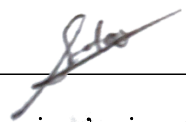
Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3Y3	Study week no.:10
Student Name & ID: Chong Shao Yang 20ACB05213	
Supervisor: Dr. Aun Yichiet	
Project Title: A Homeostatic Approach to Adaptive Ambient Control in Smart Factories	

1. WORK DONE Completed preparing and started with modelling the predictive model.
2. WORK TO BE DONE Camera at Raspberry Pi still need to find a solution to fix it.
3. PROBLEMS ENCOUNTERED Camera is still not working but tested with voltmeter shown the camera is working fine.
4. SELF EVALUATION OF THE PROGRESS Need to hurry up with solving the camera issue



Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3Y3	**Study week no.:11**
Student Name & ID: Chong Shao Yang 20ACB05213	
Supervisor: Dr. Aun Yichiet	
Project Title: A Homeostatic Approach to Adaptive Ambient Control in Smart Factories	

1. WORK DONE Completed with predictive modelling and halfway to the computer vision model
2. WORK TO BE DONE Need to get more data for training and find a way to increase the accuracy of the model
3. PROBLEMS ENCOUNTERED The accuracy of the computer vision model is too low due to insufficient data
4. SELF EVALUATION OF THE PROGRESS Need to keep up the work.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3Y3	**Study week no.:12**
Student Name & ID: Chong Shao Yang 20ACB05213	
Supervisor: Dr. Aun Yichiet	
Project Title: A Homeostatic Approach to Adaptive Ambient Control in Smart Factories	

1. WORK DONE
Completed with computer vision model
2. WORK TO BE DONE
Need to try find a way to enhance the model while writing report
3. PROBLEMS ENCOUNTERED
The model accuracy is still not achieve the targeted accuracy
4. SELF EVALUATION OF THE PROGRESS
So far so good.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3Y3	**Study week no.:13**
Student Name & ID: Chong Shao Yang 20ACB05213	
Supervisor: Dr. Aun Yichiet	
Project Title: A Homeostatic Approach to Adaptive Ambient Control in Smart Factories	

1. WORK DONE
Completed the report and improved the performance of the computer vision model
2. WORK TO BE DONE
Need to continue working on the python program in raspberry pi
3. PROBLEMS ENCOUNTERED
None
4. SELF EVALUATION OF THE PROGRESS
Great, everything went well.

Supervisor's signature

Student's signature

APPENDIX

POSTER

A HOMEOSTATIC APPROACH TO ADAPTIVE AMBIENT CONTROL IN SMART FACTORIES

01. INTRODUCTION

- The integration of 4IR technologies, particularly IoT and AI/ML, presents a transformative opportunity in animal feed production.
- By implementing real-time data collection, process monitoring, and predictive analysis, manufacturers can achieve significant improvements in feed quality, production efficiency, and overall business sustainability.

02. OBJECTIVES

To develop a predictive model to predict the outcome based on the input configuration

To integrate IoT sensors like moisture, humidity and temperature for ambient control and presented in dashboard with Grafana

To develop a computer vision (CV) model for Quality Control (QC) and monitoring based on colour and appearance

03. SOLUTION

With Raspberry Pi and sensors setup as a demo of the factory workflow

Push every data to Docker application container with configured apps

interface for demo ML model

04. CONCLUSION

The system is built in the motivation of

- Minimizes human error.
- Automates dryer settings.
- Ensures consistent feed quality through real-time monitoring and predictive models.

Additionally, the project presents a novel algorithm for converting sensor data into moisture content

APPENDIX

PLAGIARISM CHECK RESULT

plagiarism check.docx

ORIGINALITY REPORT

7%	7%	5%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	eprints.utar.edu.my Internet Source	2%
2	aaloktech.in Internet Source	1%
3	www.bartleby.com Internet Source	<1%
4	www.cyberciti.biz Internet Source	<1%
5	kanazawa-u.repo.nii.ac.jp Internet Source	<1%
6	sigmatechbd.com Internet Source	<1%
7	erimt2015.ic.ufmt.br Internet Source	<1%
8	link.springer.com Internet Source	<1%
9	www.hackster.io Internet Source	<1%

10	timesofindia.indiatimes.com Internet Source	<1 %
11	Maria Pia Del Rosso, Alessandro Sebastianelli, Dario Spiller, Pierre Philippe Mathieu, Silvia Liberata Ullo. "On-Board Volcanic Eruption Detection through CNNs and Satellite Multispectral Imagery", Remote Sensing, 2021 Publication	<1 %
12	sin.put.poznan.pl Internet Source	<1 %
13	S. Omid Fatemieh. "A Review of: “Discovering Knowledge in Data: An Introduction to Data Mining”", Journal of Biopharmaceutical Statistics, 2006 Publication	<1 %
14	www.livemint.com Internet Source	<1 %
15	Lewis, Micah, Samir Trabelsi, and Stuart O. Nelson. "Circuits and signal conditioning for a peanut-drying monitoring system", SoutheastCon 2015, 2015. Publication	<1 %
16	Pawan Singh Mehra, Dhirendra Kumar Shukla. "Artificial Intelligence, Blockchain, Computing and Security - Volume 2", CRC Press, 2023	<1 %

Publication

17	fastercapital.com Internet Source	<1 %

Exclude quotes	On	Exclude matches	< 8 words
Exclude bibliography	On		

APPENDIX

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Chong Shao Yang
ID Number(s)	20ACB05213
Programme / Course	IB
Title of Final Year Project	A Homeostatic Approach to Adaptive Ambient Control in Smart Factories

Similarity	**Supervisor's Comments** (Compulsory if parameters of originality exceeds the
Overall similarity index: 7 **%** **Similarity by source** Internet Sources: 7 % Publications: 5 % Student Papers: 0 %	
Number of individual sources listed of more than 3% similarity: 0	
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words**	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

APPENDIX

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

____
Signature of Supervisor

Name: Aun Yichiet

Date: 12/09/2024

____
Signature of Co-Supervisor

Name: ______________________________

Date: ______________________________

APPENDIX

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY

(KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	2005213
Student Name	Chong Shao Yang
Supervisor Name	Dr. Aun Yichiet

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content

APPENDIX

√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
✓	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

____

(Signature of Student)

Date: 10/09/2024