# DFT Inspection Module Electronic System Architecture Plan (ESAP)

| Rev | Author | Date | Modification |
| --- | --- | --- | --- |
| 1 | Mohamed Abdelmoneim | 04/07/2023 | Initial draft |

Appendix

DFT Inspection Module Electronic System Architecture Plan (ESAP) [1](#_heading=h.gjdgxs)

[Appendix 1](#_heading=h.qsh70q)

[1- Introduction: 2](#_heading=h.30j0zll)

[- 1.1 Purpose 2](#_heading=h.1fob9te)

[- 1.2 Overview 2](#_heading=h.3znysh7)

[- 1.3 Definitions 2](#_heading=h.2et92p0)

[2- Electronics required. 2](#_heading=h.tyjcwt)

[- 2.1 Raspberry Pi 2](#_heading=h.3dy6vkm)

[- 2.2 Elcometer scale 3 Ferrous probe 2](#_heading=h.1t3h5sf)

[- 2.3 Elcometer 456 Ferrous gage – Model S 3](#_heading=h.4d34og8)

[- 2.4 AIDA 1080p camera 3](#_heading=h.2s8eyo1)

[- 2.5 Stepper motor 3](#_heading=h.17dp8vu)

[- 2.6 Stepper motor driver 3](#_heading=h.3rdcrjn)

[- 2.7 Ethernet switch 3](#_heading=h.26in1rg)

[- 2.8 Solenoid: 3](#_heading=h.lnxbz9)

[- 2.9 USB to Ethernet adapter 3](#_heading=h.35nkun2)

[3- Communication protocol: 3](#_heading=h.1ksv4uv)

[- 3.1 Overview 3](#_heading=h.44sinio)

[- 3.2 Communication bytes 3](#_heading=h.2jxsxqh)

[- 3.3 Checksum 4](#_heading=h.z337ya)

[4- Firmware Design: 4](#_heading=h.3j2qqm3)

[- 4.1 Firmware overall design 4](#_heading=h.1y810tw)

[- 4.2 Control box 4](#_heading=h.4i7ojhp)

[- 4.3 Raspberry Pi 4](#_heading=h.2xcytpi)

[5- Software Design: 5](#_heading=h.1ci93xb)

[- 5.1 User interface 5](#_heading=h.3whwml4)

[- 5.2 Data storage 5](#_heading=h.2bn6wsx)

# Introduction:

# 1.1 Purpose

* The purpose of this document is to describe the behavior of the In-pipe inspection crawler, list all the needed electronics, describe the purpose of each component, and provide a deeper explanation of the integration of the control box and the inspection module.

# 1.2 Overview

* The overall functionality of this robot is to ensure that the coatings on the pipe are within the required thickness, record the thickness measurement, the distance in the pipe at which the measurement was taken, the angle at which the measurement was taken, along with time stamps, live feed, and a photo for each measurement.

# 1.3 Definitions

* IAS: A robot responsible for painting the pipe with the ferrous coating.
* ISAAC: A newer version of the IAS that has the same functionality as the IAS but has more efficiency and durability.
* Control box: A computer that has the user interface for the system and gives the user control over the whole coating and measurement system.
* DFT inspection module: A robot that is responsible for measuring the coating thickness after being painted and providing feedback on the measurement to the control box.
* Time stamp: Data associated with each measurement that contains the month, day, year, hour, and minute at which the measurement was taken, and it will be in the following format: MM-DD-YYYY-HH\_MINMIN.

# Electronics required.

# 2.1 Raspberry Pi

* The Raspberry Pi will be controlling the inspection module mechanism and providing feedback to the control box.

# 2.2 Elcometer scale 3 Ferrous probe

* The probe will be responsible for measuring the thickness of the coating through the Eddy Current Sensing measurement technique and provide feedback to the gauge.

# 2.3 Elcometer 456 Ferrous gage – Model S

* The gauge will take the measurements from the probe and send the measurements to the Raspberry Pi.

# 2.4 AIDA 1080p camera

* There will be 2 cameras in this system, responsible for providing the control box with live feedback, along with pictures for each measurement.

# 2.5 Stepper motor

* The stepper motor will be responsible for tilting the cameras, and the probe.

# 2.6 Stepper motor driver

* The stepper motor driver is responsible for providing the Raspberry Pi with control over the stepper motor.

# 2.7 Ethernet switch

* The Ethernet switch connects the Raspberry Pi and the 2 cameras with the control box.

# 2.8 Solenoid:

* Responsible for extending the probe to touch the coating, to take a measurement.

# 2.9 USB to ethernet adapter

* Responsible for connecting the Raspberry Pi to the ethernet switch with an Ethernet cable.

# Communication protocol:

# 3.1 Overview

* The Raspberry Pi's responsibility will be to send the values of the measurements and the visual feedback to the control box in the form of Ethernet packets.
* The Control box’s responsibility will be to store these packets in a form that the user can understand.
* To ensure that the connection is properly established there will be signals sent between the Raspberry Pi and the control box.

# 3.2 Communication bytes

* Request to connect: A byte in the form of 0xAA (10101010 in binary)
* Take a measurement command: A byte in the form of 0x55 (01010101 in binary)
* Ack for receiving the photo: A byte in the form of 0xD5 (11010101 in binary)
* Ack for getting the current position from the Roboclaw encoder: A byte in the form of 0x2A (00101010 in binary)

# 3.3 Checksum

* To make sure that the values sent by the Raspberry Pi were not corrupted a 16-bit checksum that’s built in the TCP header will be used.
* The checksum will be used when the Raspberry Pi sends back the measurements, the angle, and the distance in the Raspberry Pi.

# Firmware Design:

# 4.1 Firmware overall design

* The Firmware is responsible for opening a connection with the control box, taking measurements, providing live feed for the control box, and controlling the inspection module mechanism.

# 4.2 Control box

* The control box will represent the user, and it will be responsible for three main things:

1. Send commands to the Raspberry Pi based on the user’s needs, and Receive data (Measurements, Live feed, position during measurement, and angle) from the Raspberry Pi.
2. Send the current position of the robot to the Raspberry Pi to associate it with the measurement.
3. Store the data in a form that the user can understand.

# 4.3 Raspberry Pi

* Raspberry Pi is responsible for controlling the mechanism of the robot based on the user’s requests coming from the control box.
* Raspberry Pi will take the measurement data in the Elcometer gauge through USB and transfer them to the control box.
* Raspberry Pi will also send to the control box the following data:

1. The distance at which the measurement was taken.
2. The angle at which the measurement was taken.
3. The timestamp for the measurement.
4. A photo associated with the measurement.

* Raspberry Pi will control the movement of the stepper motor to put the position at the right angle that the user asked for.
* Once in position the Raspberry Pi will drive the Solenoid Pin HIGH to extend the probe.
* Once a measurement is taken and recorded in the gauge’s USB Drive, the Raspberry Pi will record it, and associate it with the distance, timestamp, and angle at which the measurement was taken.

# Software Design:

# 5.1 User interface

* There already exists a user interface in the control box in the form of an .exe file that’s written in Python.
* An additional user interface feature must be added to the existing Python program to enable users to transfer to the Live Feed view provided by the AIDA cameras.

# 5.2 Data storage

* The control box needs to store the data sent by the inspection module in an Excel file in a readable way for the user.
* The control box will also store the photo associated with each measurement and name it according to the time stamp format.
* The control box will mark the failure in the Excel file, and name the failed picture as a failed picture with its time stamp.