

Automated material testing

Lorenzo Bosio

A thesis submitted in partial fulfilment of the requirements for
the degree of
Master of Science in Materials Engineering for Industry 4.0

Politecnico di Torino - Turin, Italy
Chalmers University of technology - Göteborg, Sweden
June 19, 2025

Abstract

The integration of robotic systems into materials testing laboratories is increasingly recognized for its potential to enhance throughput and achieve high-precision experimental measurements. At the Volvo Cars facility, an automated 3-point bending test cell was developed to enable high-throughput material testing in accordance with the VDA 238-100 standard, ensuring consistent and accurate test execution. This project addresses key challenges encountered during system development, including automation integration, sample handling, and data management. Various design strategies and technical solutions are explored to optimize testing efficiency and measurement accuracy. Additionally, the potential for extending the system to accommodate other test types, such as tensile testing, is examined. The outcomes, challenges, and proposed enhancements provide valuable insights into advancing automated testing systems for industrial applications.

Acknowledgments

The successful completion of this project would not have been possible without the invaluable contributions of several individuals and organizations.

Special thanks go to Kai Kallio, whose original vision and unwavering support were instrumental to the realization of this work. His extensive technical and practical expertise greatly influenced the project's development and success. Gratitude is extended to Arian Kamal for his significant assistance with PLC programming and software integration, to Lucas Scheuer for sharing his in-depth knowledge of material testing and for facilitating key interactions with component suppliers, to Eric Bojestig, for being my reference point about anything laboratory related and Nils Andersson for all the help with robot usage. I would like to express my sincere gratitude to Patrik Träff for his warm welcome and the dedication he showed in integrating me into the team. I am also deeply thankful to the entire Materials Engineering Center for their invaluable support and continued interest throughout my time at Volvo Cars.

Special thanks go to Professors Martin Fabian and Matteo Pavese for their guidance and for helping maintain an academic perspective during the thesis work.

Several companies provided essential support and equipment throughout the project: Piab Vacuum Technology, for supplying suction cups and vacuum grippers; Robot System Products, for providing the manual tool changer; Zwick Roell and MTS, for their responsive and effective technical support; ABB, for their highly reliable systems and the expertise of their technical staff.

Lastly, this work would not have been possible without the immeasurable support and encouragement of my family Alessandro, Rosetta, and Francesca. I am also deeply grateful to the friends I met along the way Alberto, Leonardo, Giovanni, Renato, Luca, Marina, Antoine, Luigi, Matteo, Federico, Francesco and many others whose presence and motivation made this journey truly special.

Contents

1	Introduction	8
1.1	Contextualization of the thesis work at Volvo Cars	8
1.2	Aim of the project and evaluation parameters	9
1.3	Analysis of already existing systems	9
2	Automated bending system	11
2.1	System requirements	11
2.2	Test setup	12
2.3	Layout	13
2.3.1	Coordinate systems	14
2.4	Robot setup	15
2.4.1	Robot specification	15
2.4.2	Safety configuration	16
2.4.3	Programming	17
2.4.4	Tool	18
2.5	Pick up station	19
2.6	QR code station	20
2.7	Measurement station	20
2.8	Testing station	22
2.9	Disposal station	23
2.10	Signal handling	24
2.10.1	PLC	27
2.11	Data handling	27
2.12	Error and safety	28
2.13	Results	28
2.13.1	Imperfections	30
3	Automated tensile system	32
3.1	Requirements	32
3.2	Test Setup	33
3.3	Robot Setup	33

3.3.1	Tool head	34
3.4	Robot-Machine communication	34
3.5	Disposal	35
4	Discussion	36
4.1	Bending test system	36
4.1.1	Improvements	37
4.2	Tensile test system	38
4.2.1	Improvements	38
5	Conclusion	39
A	Graphs and Images	46
A.1	Data for Section 2.13	46
A.2	Test samples	47
A.3	Robot platform and bending machine	47
B	Calibration	48
B.1	Calibration for laser sensor	48
B.2	Calibration of force and crosshead displacement	48
C	Full code	49
C.1	PLC code	49
C.2	Python Code	49
C.3	Robot code	51

List of Figures

2.1	Geometry used according to VDA 238-100	12
2.2	Layout plan showing all stations and their relative position. . .	14
2.3	Robot schematic [1]	16
2.4	Modelled station implemented in RobotStudio.	17
2.5	Virtual model (above) and physical tool (below).	19
2.6	Measuring station	22
2.7	Disposal station showing 12 processed samples.	24
2.8	Signal map	25
2.9	CT scan of the samples from 1 to 12.	31
2.10	Data for bending angle (left) and max force (right).	31
3.1	Tool used to handle tensile test samples.	34
A.1	Bending angle and peak force raw data, plotted with thickness	46
A.2	Aluminium test sample for bending (left) and tensile(right). .	47
A.3	Robot arm mounted on movable platform (left) and bending machine (right).	47
C.1	FBD code for PLC	49

List of Tables

2.1	Test parameters	13
2.2	Signal connections	26
2.3	Comparison of mean and standard deviation	29

Abbreviations and Acronyms

PLC	Programmable Logic Controller
DI	Digital Input
DO	Digital Output
AI	Analog Input
AO	Analog Output
TCP	Tool Center Point
FBD	Function Block Diagram
DIC	Digital Image Correlation
CT Scan	Computed Tomography Scanning

Chapter 1

Introduction

Manual testing of raw material properties is a slow, repetitive, and error-prone process. Operators must measure each sample individually, configure the testing setup, and evaluate the results. As batch sizes increase to hundreds of samples, manual testing becomes increasingly impractical due to time constraints and the heightened risk of human error, which introduces variability into the data. To address these challenges, automated testing stations can significantly improve precision, consistency, and throughput. This report presents the development and implementation of an automated three-point bending test cell. It outlines the key design considerations, setup procedures, and system integration steps, and further explores the potential for automating other mechanical test types, such as tensile testing.

1.1 Contextualization of the thesis work at Volvo Cars

At the Volvo Cars facility, in Göteborg, high-volume data acquisition is required on a daily basis to support ongoing development activities. The material analysis team is responsible for processing all testing requests originating from other departments. These material properties are critical both for ensuring vehicle safety and for calibrating predictive simulations under development.

The long-term objective is to establish a fully autonomous laboratory capable of processing various types of samples and performing tests rapidly with reduced human error. This project serves as a foundation for significantly improving laboratory efficiency by automating as many procedural steps as possible. Given the diversity of test types performed, an evaluation of the system's expansion capabilities will also be conducted.

1.2 Aim of the project and evaluation parameters

The objective of the project is to identify and evaluate the challenges associated with establishing an autonomous laboratory environment. Each automated step enables operators to focus on higher-value tasks, such as project coordination or data analysis. The steps targeted for automation include sample measurement, the testing phase, and data collection. However, with the technology present today in the Volvo Cars material testing laboratory, certain steps will continue to require manual intervention, such as obtaining samples, applying QR code labels, and analysing the collected data.

The project is structured into two phases. First, a complete three-point bending test cell will be developed. Three-point bending was selected due to the relatively simple handling of samples, particularly after testing. This phase aims to identify the principal technical challenges associated with automation with the aim of obtaining high volume of data for statistical analysis [2]. System performance will be evaluated by collecting real-world test data and comparing it to results obtained through manual testing procedures.

In the second phase, modifications necessary to extend the system for tensile testing will be explored. The objective is to adapt the automated system to accommodate multiple test types, thereby increasing its versatility.

1.3 Analysis of already existing systems

Materials can be characterized using a variety of testing methods, depending on the type of data required for the target application. Mechanical properties such as Young's modulus and yield strength are typically determined through destructive testing techniques [3],[4].

Automated testing becomes particularly relevant when dealing with a large number of samples or when high levels of precision and repeatability are required. Given the specialized nature of such systems, only a limited number are currently available and comprehensively documented in the literature [5, 6].

Several automated material testing setups have been developed for both bending and tensile testing [7],[8],[9]. Compact systems generally prioritize ease of use and standardization over speed and throughput, often being designed to process only a small number of samples. These systems aim to replicate the performance of a skilled operator rather than to maximize processing efficiency.

Conversely, larger-scale tensile testing systems are available and typically

designed for industrial environments [10],[11],[12],[13]. However, these often lack the flexibility needed to meet specific, customized requirements. Limitations include the inability to preserve fracture surfaces, measure all relevant sample dimensions, apply unique sample labelling and tracking methods, or perform multiple tests on the same specimen.

Chapter 2

Automated bending system

The following chapter details the development process of the 3-point bending test cell, highlighting the key considerations necessary to create a fully functional and reliable testing system. This includes an overview of the design process, from selecting appropriate components to ensuring the mechanical and electronic systems are integrated effectively. Critical factors such as the calibration of force and distance sensors, precision of the loading mechanism, and alignment of the sample holders will be discussed. Furthermore, the chapter will examine the importance of selecting the correct test parameters, including the load rate, displacement limits, and the environmental conditions required for optimal performance. Practical considerations, such as ease of operation, safety protocols, and the automation of data collection will also be integrated to enhance the efficiency and repeatability of testing. Finally, potential challenges in system integration, troubleshooting, and optimization of throughput will be addressed, setting the stage for a successful implementation of the bending test cell [14].

2.1 System requirements

The bending test cell must meet several key requirements. First, it is essential to track each sample throughout the process, eliminating the need for written labels and enabling the reconstruction of each sample's history, from cutting the base material to the final test data.

Preserving the sample after testing is also important, as it allows for visual inspection of the fractured surface in case the results deviate from expectations. The precision required for execution is moderate; however, the robot performance is within acceptable limits and can be further improved by utilizing the full capabilities of the hardware. Therefore, the focus will be on

maximizing productivity while maintaining test quality. The primary data obtained from the three-point bending test is the bending angle. Additional useful data include the maximum applied force and the thickness of each sample.

2.2 Test setup

The VDA 238-100 standard for three point bending of sheet materials is widely adopted by automotive suppliers and material manufacturers to assess local formability [15],[16]. The standard specifies geometric constraints, including the size and shape of the fixture, as well as the punch dimensions. A diagram illustrating the geometry is provided in Figure 2.1. In addition, the standard outlines test execution parameters, such as preload, stopping conditions, and speed settings to be applied at each step of the process.

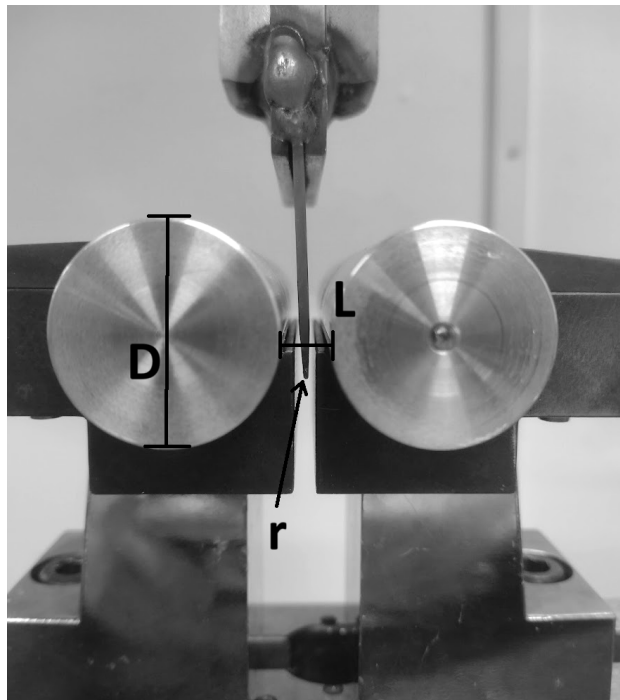


Figure 2.1: Geometry used according to VDA 238-100

The test is carried out using 60 mm×60 mm square samples. Sample thickness is a critical parameter in bending tests and will be measured with greater precision for each sample during the testing cycle. The distance between the rollers L is a function of the sample thickness t_m . The fixture used in the

test does not permit automated adjustments, so it must be fixed in position. Given that all samples have a nominal thickness of approximately 3 mm, the roller distance is set at 6.09 mm. This decision was made to maintain consistency with the reference data used in the results (see Section 2.13). All other test parameters are in accordance with the standard, and precise values are provided in the table below.

Symbol	Description	Value [mm]
r	Punch radius	0.4 ± 0.02
D	Roller diameter	30 ± 0.01
L	Roller distance	6.09 ± 0.1
b	Sample width	60 ± 2
l	Sample length	60 ± 2
t_m	Sample thickness	Measured for each sample

Table 2.1: Test parameters

2.3 Layout

To facilitate the organization and design of the system, a layout sketch is essential. The layout, shown in Figure 2.2, includes several key areas, each fulfilling a distinct role within the automated cycle. Starting from the pick-up station, it serves as the initial point where samples are stored and retrieved by the robot. Next, at the QR code station, each specimen is identified via a QR code sticker to ensure accurate traceability throughout the process. The measurement station is dedicated to determining the thickness of each sample, a parameter of particular importance in the bending procedure. The testing station is where the three-point bending test is performed, following precise alignment and placement on the fixture. Finally, the disposal station is used to offload samples after testing is complete, allowing for either removal or post-test inspection.

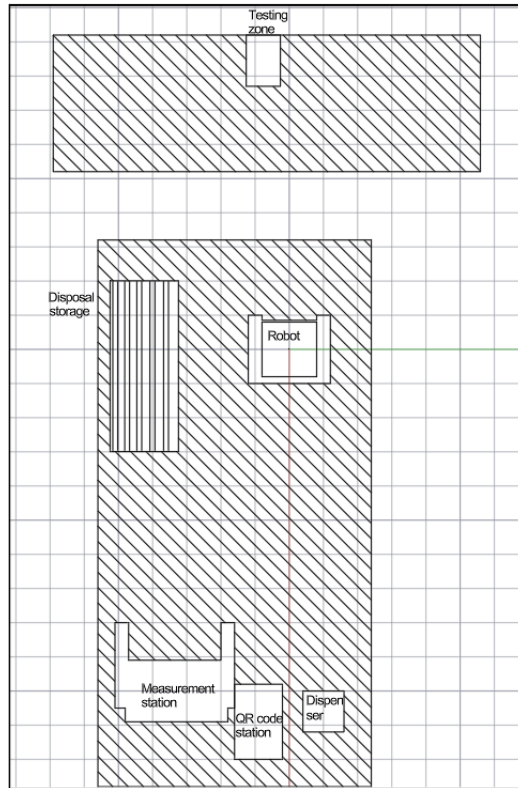


Figure 2.2: Layout plan showing all stations and their relative position.

The layout is in scale, to have a clear idea of the available space for manoeuvrability and the overall appearance of the cell.

2.3.1 Coordinate systems

Multiple coordinate systems are used within the robotic setup, each serving a distinct purpose depending on whether they are fixed or dynamic. The primary reference frame is the world coordinate system, which is static and is located in the centre of the robot base. It is used to define the position of all elements in the layout within the $x - y$ plane, as illustrated in Figure 2.2. This choice facilitates straightforward access to the spatial coordinates of specific station components relative to the robot base, and the same reference is maintained in the virtual RobotStudio environment for consistency between the physical and simulated setups. Another key reference frame is the TCP coordinate system, which is dynamic and rigidly attached to the tool. It moves with the tool and is used to define the position and orientation of the end effector when reaching programmed targets. Each time a new

tool is defined, a corresponding TCP coordinate system is established at its designated location and orientation.

Lastly, the work object coordinate system is a local frame typically employed when operations must remain consistent relative to the work object and independent of external elements. In the current implementation, this coordinate system was not required, as all necessary positioning could be handled using the world and TCP references.

2.4 Robot setup

The robotic arm plays a crucial role in transferring the sample between stations and ensuring smooth operation throughout the testing process. It is controlled by the robot controller, which executes the programmed instructions, manages signal communication, and handles error responses.

A tool is mounted at the end of the robotic arm, functioning as an interface between the robot and the objects it manipulates—in this context, the test samples. The tool is custom-made for the required application, and must be defined within the robot’s programming environment. It is discussed in detail in Section 2.4.4.

2.4.1 Robot specification

The robot used is an ABB cobot (collaborative robot) CRB 15000 [17]. It has an arm reach of 0.95 m and 6 degrees of freedom. The absolute position accuracy of 0.1 mm ensures the required precision is reached. The maximum payload of 5 kg is not a limiting factor, nor is the maximum speed of 2.2 m/s of the TCP (Tool Center Point) [18]. A visual representation of the robot arm is shown in Figure 2.3.

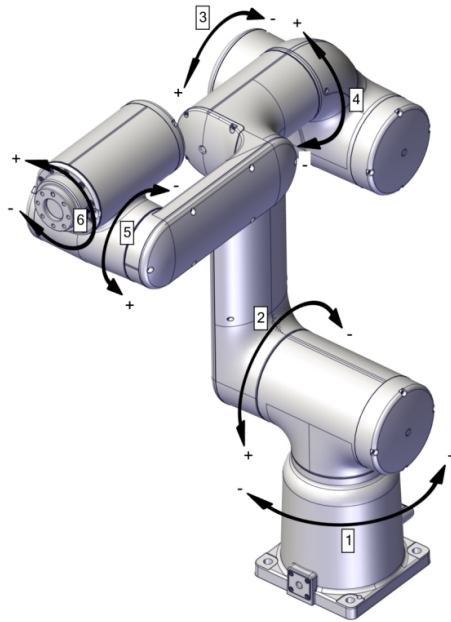


Figure 2.3: Robot schematic [1]

Being a cobot, it has force and torque sensors to stop movement in case of contact with a human or an unexpected object. This feature will also be used for force dependent operations.

The available signals are 16 DO, 16 DI, but no analog input or output is available directly in the robot controller. The robot includes a FlexPendant [19] as interface directly with the robot controller. It is used to visualize uploaded programs, monitor execution and perform jogging operations.

2.4.2 Safety configuration

The robot requires a properly configured safety system to ensure safe operation. These configurations are implemented through safety zones, which establish spatial boundaries where specific constraints of speed and force are applied. Accurate calibration of these zones is essential for maintaining a smooth and efficient workflow while ensuring operator safety.

Beyond zone configuration, comprehensive safety involves performing a risk assessment in compliance with industrial safety standards such as ISO 10218 [20],[21] for industrial robots and ISO/TS 15066 [22],[23],[24] for collaborative applications [25]. Properly implementing these measures ensures that the robot operates within a safe and controlled environment.

2.4.3 Programming

The robot program is written in RAPID [26], a high-level programming language developed by ABB specifically for robotic systems. RAPID can be learned through a variety of online resources, including instructional videos and documentation. The corresponding software for programming and simulation is RobotStudio, also developed by ABB, which allows users to modify code, model the station in a 3D environment, and simulate the current setup [27, 28]. Figure 2.4 shows the 3D modelled station inside RobotStudio.

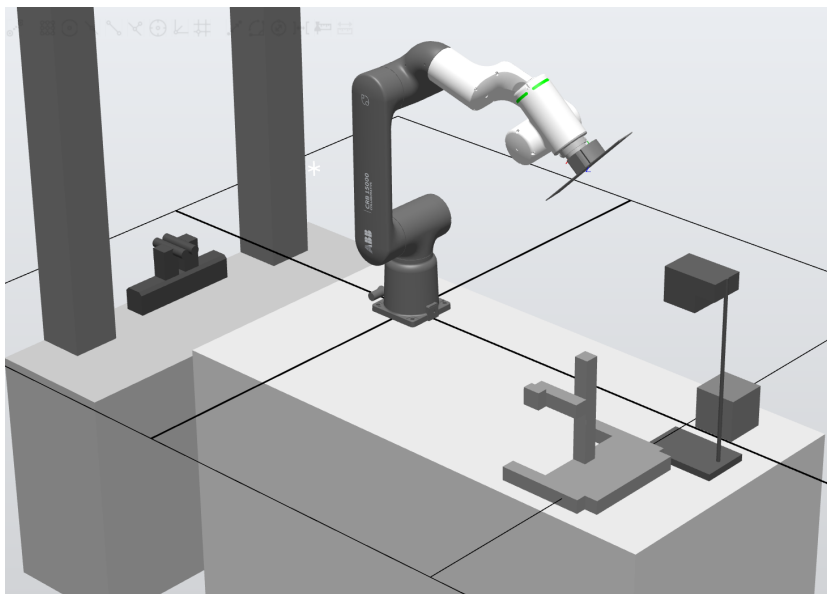


Figure 2.4: Modelled station implemented in RobotStudio.

At the beginning of the programming process, key points of interest, as well as all necessary tool data and variables, must be defined. The robot targets can be specified either through jogging, using the “lead-through” functionality of the robot, or manually. It is crucial that each target not only defines the robot’s position in space but also the orientation of the tool at that position.

The program is also responsible for overseeing and altering all signals states in the controller and determines the pace of the whole cell by monitoring the measurement process and test execution.

To facilitate debugging and improve code understanding, the use of functions is essential. By invoking functions only when necessary, it is possible to reuse, adapt, and organize code sections more effectively. The program is executed cyclically an indefinite number of times unless otherwise specified.

See Appendix C.3 for the full program.

2.4.4 Tool

The tool, or end effector, is an essential component of the robotic setup, acting as the interface between the robot and the test samples. It consists of two main components: the tool exchanger and the tool head.

The tool exchanger is a small cylindrical part itself also composed of two sections. The tool attachment is fixed and connects directly to the robot arm, transmitting all electrical signals from the controller and air connections. The tool exchanger, in turn, serves to transfer these signals and air connections to the tool head and is designed to be detachable.

The specific parts used in this setup are the Coboshift tool attachment MTA18 and Coboshift tool changer MTC18, provided by Robot System Products [29]. These components support up to six electrical connections and four air ducts. While the tool exchanger used here is manual, automatic versions are also available.

The tool head was designed using 3D CAD software to securely attach to the tool exchanger with four screws. The design consists of two distinct sections (see Figure 2.5). The first section houses two suction cups, which are used to handle the test samples before they are subjected to the bending test. The second section is a vacuum gripper, responsible for managing the bent samples after the test is completed. The suction cups, vacuum gripper, and vacuum controller were all provided by Piab.

Modelling the tool head is crucial, especially if 3D printing is required for fabrication. Even in other cases, having an accurate 3D model is beneficial, as it ensures proper dimensioning, enables easy import into RobotStudio, and allows for accurate adaptation of the safety configuration. Figure 2.5 below shows both the 3D CAD model and the corresponding physical object.

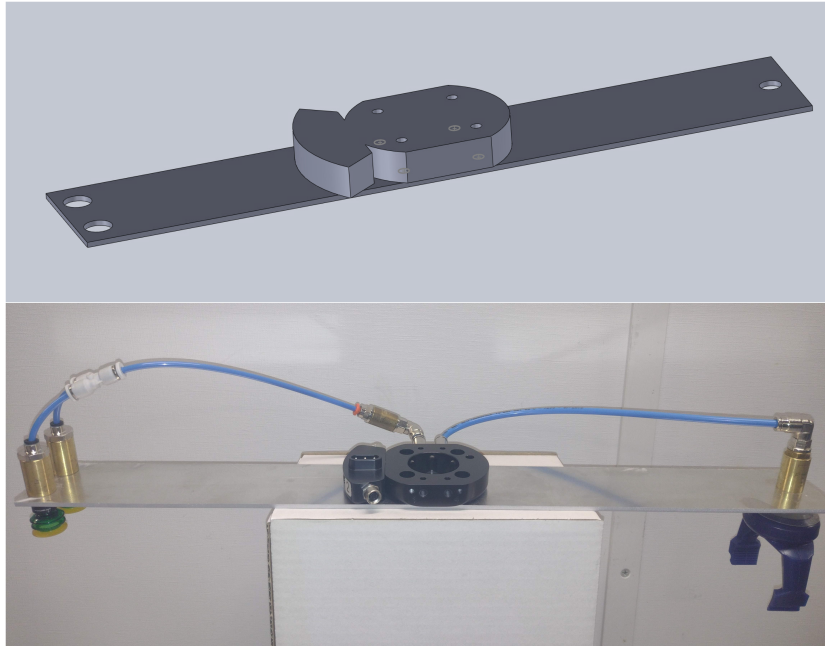


Figure 2.5: Virtual model (above) and physical tool (below).

2.5 Pick up station

Before initiating the system, the operator must manually prepare the samples by cutting them to size, labelling them with QR code stickers, and placing them in the designated storage area. Various storage solutions were considered; however, a simple stacking approach proved to be the most effective. In this configuration, samples are stacked vertically in a predefined location. Once the system is activated, the robot is programmed to retrieve and process each sample in sequence without further input. Only the x and y position of the sample stack needs to be defined, as the robot relies on a searching routine to detect the correct vertical distance. Once the appropriate proximity is reached, the sample is lifted off by the negative pressure in the suction cups, even before coming in full contact, and is securely attached to the tool. Although this approach results in some loss of positional precision, it is acceptable at this stage, as high accuracy is not critical during the initial pickup. However, adjustments must be made to eliminate this upwards motion before placing the sample onto the test fixture, where precise positioning becomes essential for reliable test execution.

If the height of a single sample stack exceeds the robot's reach or the system's constraints, a multi-stack configuration can be implemented. Addi-

tional stacks (e.g., stack 2, stack 3, etc.) follow the same operating principles and are processed in sequence. At program startup, the user must input the number of samples in each stack. This allows the system to track the number of completed cycles and ensures that all samples are processed accordingly.

2.6 QR code station

The QR code station consists of a fixed camera and a designated area where the robot presents each sample for identification. The camera is connected to a computer running a continuously active Python script. This script analyses the camera feed frame by frame to detect and decode any visible QR codes through the OpenCV library [30]. Each QR code encodes information related to the sample's processing history. For simplicity, the current implementation encodes only the sample number, but this can be extended to include additional metadata if required.

The most critical to adjust, to ensure smooth operation, is the focus point of the camera, as QR-codes not in focus might not be recognised by the camera. Although this is a rare occurrence, placing the camera at 15 to 20 cm from the ground surface can help minimize errors.

Upon detecting a new QR code, the script automatically generates a new Excel spreadsheet. This file includes predefined column headers and populates the first row with the decoded sample number. As the process continues, each cell is filled with the corresponding test data, already converted to the appropriate measurement units.

2.7 Measurement station

The measurement station, shown in Figure 2.6, is dedicated to evaluating the sample thickness, a critical parameter in the bending test that must be determined with high precision. Other geometric parameters, while relevant, are not as essential for this specific test and will therefore not be measured individually for each sample; instead, representative estimations will be provided.

The measurement procedure involves determining the difference in distance readings between the presence and absence of a sample, thereby allowing the extrapolation of thickness values. This is achieved using a downward-facing laser sensor, which determines distance based on the triangulation of the reflected beam. The sensor used for this task is the CP35MHT80 [31], a high-performance device capable of achieving a resolution of 0.05 mm and a

working range from 5 to 35 cm. The sensor outputs an analog voltage signal ranging from 0 to 10 V.

Calibration of the device was carried out using reference objects with known thicknesses. These measurements were used to construct a calibration line (see Appendix B.1), enabling accurate interpretation of sensor readings [32, 33].

The measurement station is intentionally angled to allow each sample to be dropped and guided into a predefined position using gravity. This design minimizes variation in sample placement, thereby enhancing precision during the subsequent pick-up and testing phases. Retrieving the sample after measurement requires delicate handling due to the need for high positional accuracy. To address this, the robot employs force control functionality, which enables the programming of a specific trajectory along with a target force in the z direction. Once the specified force is reached, the program proceeds to the next step. This functionality is enabled by the force sensors integrated into the robot arm. The applied force is set to 5 N, ensuring consistent contact and minimizing potential inaccuracies during the lifting operation.

The method chosen is not optimal in terms of accuracy and cannot take into account curved samples. Solving this issue involves more advanced equipment to take measurements from both sides [34]. This laser measurement solution was selected for convenience and the possibility to be combined with the alignment.

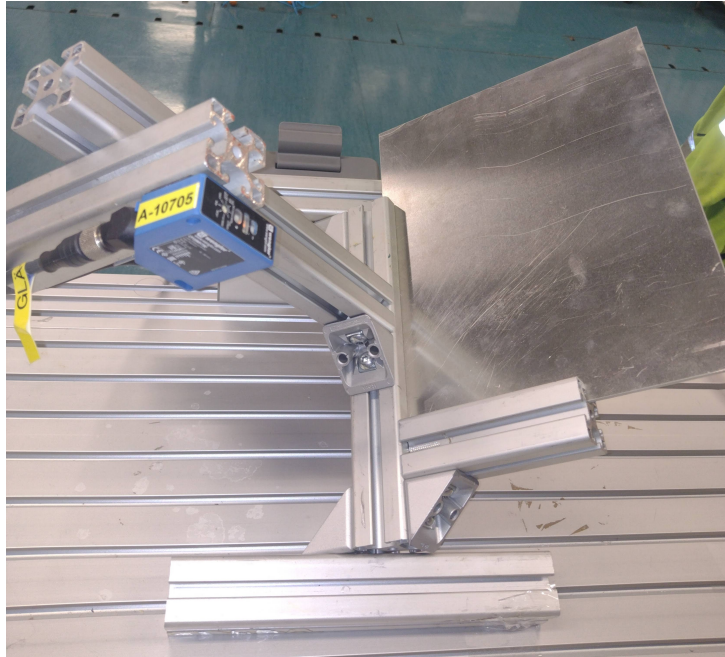


Figure 2.6: Measuring station

2.8 Testing station

The testing procedure consists of multiple steps and safety checks to ensure proper operation. The process begins with the precise placement of the sample onto the fixture. Position verification is carried out using an upward-facing laser sensor, similar to the one used for thickness measurement, to confirm the presence of a specimen.

Testing is conducted using the MST Alliance RF/100 machine [35] (see Appendix A.3), which is capable of performing both tensile and compressive tests. The crosshead, the machine's active component, consists of a knife that applies bending force to the sample and a load cell rated for measurements up to 100 kN.

The test procedure starts with a preload phase at lowering the crosshead at 10 mm/min, followed by a main loading phase at 30 mm/min. To accurately determine the bending angle, it is essential to isolate the plastic deformation by removing the influence of elastic recovery. This is achieved by introducing an additional unloading phase following the previously described loading sequence. During this phase, the crosshead is moved upwards at a rate of 10 mm/min until the applied load approaches 0 N. At this point, the system performs a 5 second hold to allow for stabilization and to record the crosshead

displacement, before returning to its initial position, thereby completing the test cycle. This additional step ensures that the displacement value used for calculating the bending angle reflects only the permanent (plastic) deformation, excluding any contribution from elastic rebound.

Several mathematical models exist to determine the bending angle α_c in degrees from the crosshead displacement. The standard VDA 238-100 provides a set of equations to use:

$$\alpha_c = 2 \left\{ \cos^{-1} \left[\frac{W \cdot p - c \cdot (S - c)}{p^2 + (S - c)^2} \right] \cdot \left(\frac{180^\circ}{\pi} \right) \right\}, \quad (2.1)$$

where:

$$W = \sqrt{p^2 + (S - c)^2 - c^2}, \quad (2.2)$$

and:

$$c = \frac{D}{2} + t_m + r. \quad (2.3)$$

Here, S is the crosshead displacement after preload, while the other parameters are described in Table 2.1 (see Appendix C.2, Lines 124-131 for the implementation into the program). Although mathematical models are based on analytical formulas and experimental errors are unavoidable, independent assessments and literature reviews found the angle prediction to be reasonably accurate, typically underestimating by up to 5° [36, 16, 37].

The testing sequence is triggered by a single digital input, while data acquisition is handled through two analog outputs, one for crosshead position and one for applied load. Both outputs are calibrated within the 0–10 V range and are interpreted by the PLC. During the test, the robot tool rotates into position, and once the crosshead returns to its initial state, the vacuum gripper is activated to safely remove the tested sample from the fixture.

2.9 Disposal station

The purpose of the disposal storage, shown in Figure 2.7, is to collect and organize all tested samples in sequential order, while preventing damage to their surfaces. In line with the design philosophy of maintaining simplicity, the solution involved attaching strips of plastic to a base, creating compartments where samples can be vertically inserted from above. Each sample is placed upright into an individual slot within a grid, ensuring separation and traceability. The robot determines the correct placement based on the completed cycle count, thereby preserving the sample order. Once this step is finalized, the system resets and is ready to initiate the next testing cycle.

The number of storage slots in the disposal station is also the limiting factor for the maximum number of samples that can be processed without supervision. Increasing the capacity is a straightforward modification that involves the physical expansion and the corresponding adaptation in the code.

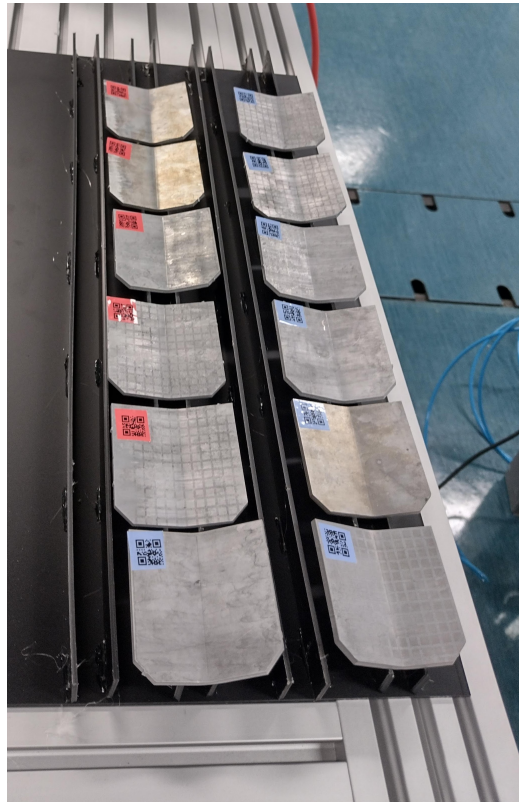


Figure 2.7: Disposal station showing 12 processed samples.

2.10 Signal handling

In order to communicate between the robot controller, the PLC and the other components in the system, several signals were used. Digital inputs and outputs are 24 V switches on and off. Analog signals instead can carry any voltage between 0 and 10 V. Figure 2.8 provides an overview of all signals used within the system, including the corresponding input ports where applicable. The primary components involved are the robot controller, the PLC, and a PC, which exchange signals not only with each other but also with peripheral devices such as the end effector, MTS testing machine, laser sensors, and QR code camera.

Table 2.2 presents the functionality of each signal, along with the corresponding source and destination components involved in the communication.

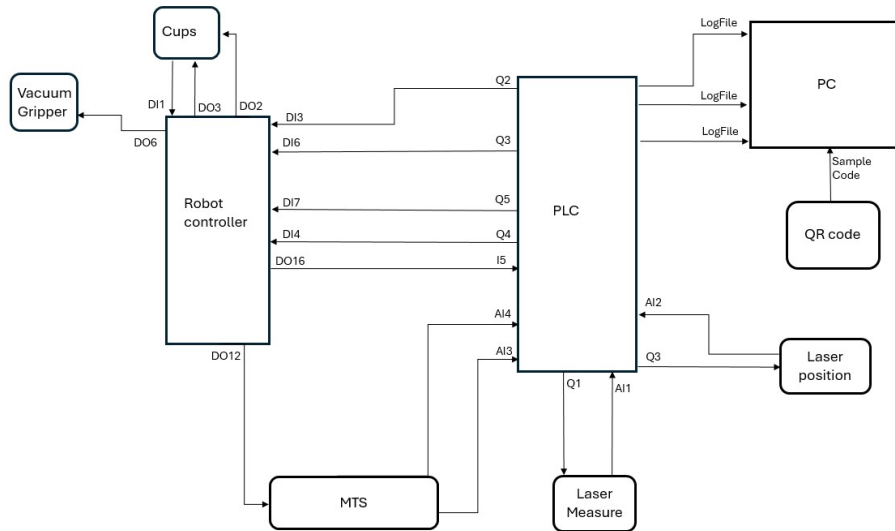


Figure 2.8: Signal map

Name	Type	From	To
Activate vacuum	Digital	Robot controller DO2	Suction cups
Activate blow-off	Digital	Robot controller DO3	Suction cups
Is vacuum on	Digital	Robot controller DI1	Suction cups
Activate gripper	Digital	Robot controller DO6	Vacuum gripper
Measurement is OK	Digital	PLC Q2	Robot controller DI3
Test finished	Digital	PLC Q5	Robot controller DI7
Sample position OK	Digital	PLC Q4	Robot controller DI4
Test type	Digital	Robot controller DO16	PLC I5
Execute test	Digital	Robot controller DO12	MTS machine
Crosshead position	Analog	MTS machine	PLC AI3
Load applied	Analog	MTS machine	PLC AI4
Execute measurement	Digital	PLC Q1	Laser measure
Thickness	Analog	Laser measure	PLC AI1
Do position check	Digital	PLC Q3	Laser position
Position	Analog	Laser position	PLC AI2
Thickness	Ethernet	PLC Ethernet LAN2	PC LogFile
Crosshead position	Ethernet	PLC Ethernet LAN2	PC LogFile
Load applied	Ethernet	PLC Ethernet LAN2	PC LogFile
QR code data	USB	Camera	PC USB1

Table 2.2: Signal connections

2.10.1 PLC

The use of a Programmable Logic Controller (PLC) is necessitated by the fact that the robot controller does not support analog signal ports. Additionally, the PLC enables real-time communication with a computer, allowing data acquisition and subsequent analysis using Python. For this purpose, the Siemens LOGO! 8.3 PLC [38] was selected. The control logic was implemented using Function Block Diagram (FBD) programming (see Appendix C.1) and uploaded to the PLC's onboard memory.

However, several limitations of this PLC must be addressed when high-precision operations are required. First, by default, analog signals are internally scaled to a range of 0 to 1000. This resolution is insufficient for applications demanding finer accuracy such as thickness measurement or applied load data. Second, the communication protocol only permits one-way data transmission from the PLC to the PC, by reading the memory at 0.5 second intervals. Bidirectional communication is not supported in this configuration, and implementing more advanced systems would be necessary to overcome this limitation.

2.11 Data handling

During system operation, a Python script (see Appendix C.2) runs continuously in the background. Its primary function is to process data received from both the PLC and the camera, and to generate an Excel spreadsheet containing all relevant test information. Initially, it activates the camera designated for QR code detection and then selects the correct section based on the test type. When a new QR code is detected, a row is created in the Excel file, populated with the sample identifier.

Simultaneously, a CSV file named LogFile is generated. This file is updated every 0.5 seconds and logs the current values of all PLC memory cells, including both digital states (high or low) and analog signal values. By processing this recorded data in real time, it is possible to extract and log key test parameters such as sample thickness, bending angle, maximum applied force and the slope of the force-displacement curve in the elastic region.

All obtained results are to be stored directly on the company database to be accessed from other departments.

2.12 Error and safety

To ensure smooth operation of the entire system, several safety checks are implemented. In the event of errors, the system is designed to reset itself automatically when possible, thereby avoiding complete interruption of the testing process.

Most issues encountered during operation are related to the robot's movements or speed limitations. Safety-critical errors—such as contact with a human—immediately halt the system, and in such cases, operation cannot resume automatically. All other errors within the robot program are handled through a dedicated error handler function (see Appendix C.3, Line 91). This function is responsible for either resuming the operation or resetting the system to initiate a new cycle. It can be triggered automatically upon detecting an error or manually by raising a custom exception.

To verify that each sample is within acceptable specifications, a check is performed following the thickness measurement. The digital input signal DI3 (measurement OK, Table 2.2) is activated only if the measured value falls within predefined limits, ensuring that no out-of-spec samples proceed to testing. In case of inappropriate dimensions, the specimen is retrieved and directly disposed.

Before initiating the bending test, a position verification step is performed using a laser sensor. The signal DI4 (position is OK, Table 2.2) is enabled only if a sample is correctly positioned. Although the current setup verifies the presence of a specimen, orientation checks have not been implemented.

During test execution, force limits are monitored to prevent equipment damage. Even though the load cell is rated for 100 kN, if the applied load in compression exceeds 10 kN, the system halts immediately.

Because the Python script runs continuously in the background, any runtime errors, such as camera initialization failures, file access issues, or calculation exceptions, are printed to the terminal. However, these do not interrupt the script's execution, allowing operation to continue when non-critical issues arise.

2.13 Results

The developed testing cell was evaluated by comparing two datasets. The first dataset was generated using the automated system, while the second was obtained through manual testing of samples taken from the same cast aluminium component. By testing identical material, the aim was to isolate and assess the performance of the test execution.

The system functioned as intended, successfully collecting thickness measurements, bending angles, and peak force values for all 12 samples. Table 2.3 presents the bending angle and peak force values of the two testing methodologies.

Method	Mean Bending Angle [°]	Std Dev Bending Angle [°]	Mean Max Force [N]	Std Dev Max Force [N]
Automatic	23.3	3.9	5577	686
Manual	24.9	3.7	5491	649

Table 2.3: Comparison of mean and standard deviation

The automated system demonstrated a reasonable level of accuracy, as indicated by the minimal variation observed between mean of bending angle and max force. The standard deviation serves as an indicator of both the consistency of the testing procedure and any inherent variation in the material properties [39, 40]. To isolate the precision of the testing setup itself, the same casted aluminium was chosen, thereby minimizing material-based variability.

Despite this, the standard deviation for both bending angle and maximum force was found to be higher in the automated system compared to manual testing. Several factors beyond test repeatability may have contributed to this outcome. Notably, 24 samples were tested manually, while only 12 were processed by the automated system. Additionally, the material used in the automated tests was of comparatively lower quality and could have imperfections in the structure and pores in the inside.

These external variables may have contributed to the increased standard deviation, despite potentially comparable or superior consistency in the automated test execution.

Cycle time was also calculated while operating the automated bending test cell. Slight variation occurred in the pick up, testing and disposal procedure, but the average cycle time was found to be 176 seconds.

The thickness of the 12 automatically processed samples was evaluated both manually and by the system. Notable discrepancies were observed between the two methods, with a mean difference of 0.19 mm. Given the nominal thickness of 3 mm, this corresponds to a 6% variation, which is considered unacceptable, as thickness significantly influences result interpretation and the calculation of flexural strength.

This error is entirely due to the LOGO! PLC, which converts incoming analog signals 0 to 10 V into a numerical range from 0 to 1000. The laser sen-

sensor has a range of 30 cm, meaning that the maximum resolution obtainable in the PLC memory is 0.3 mm. While being sufficient for many applications, the current use case demands higher granularity to improve measurement precision. The same limitation affects other analog signals, such as force and crosshead position. A practical workaround involves calibrating the machine's output around the relevant measurement ranges to achieve finer resolution where it is most needed (see Appendix B.2). However, this adjustment cannot be configured for the laser sensor.

2.13.1 Imperfections

Aluminium components produced through casting are susceptible to various types of internal defects, which can significantly impair their mechanical properties. These defects are typically classified based on their origin, including shrinkage, gas entrapment, die-filling issues, undesired phases, thermal contraction, and metal-die interactions [41]. The most common manifestation of these defects is internal porosity, primarily caused by material shrinkage and trapped gases.

Detecting these internal defects in cast metal components typically requires expensive equipment and time-consuming procedure. The most common methods are ultrasonic testing [42], X-ray Radiography [43], and Computed Tomography (CT) Scanning [43]. Therefore, identifying a correlation between porosity and another measurable quantity could help determine which samples warrant further analysis and which can be excluded from additional testing.

Dispersed microporosity is known to significantly affect mechanical properties, particularly by reducing the ductility and strength of materials [44, 45, 46, 47, 48]. While the influence of porosity on tensile test results is well-documented in the literature, its effect on bending test outcomes remains relatively unexplored.

In this study, the only data obtained from the automated system were bending test results, including bending angle and maximum force. To investigate potential correlations with porosity, the 12 automatically processed samples were CT scanned and are shown in Figure 2.9. The results plotted in Figure 2.10, were analysed to assess any relationship between porosity and the mechanical test data.

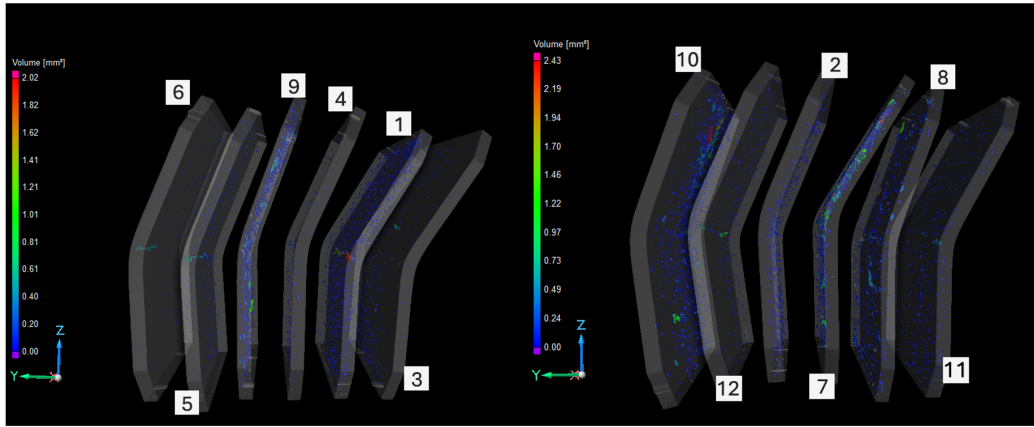


Figure 2.9: CT scan of the samples from 1 to 12.

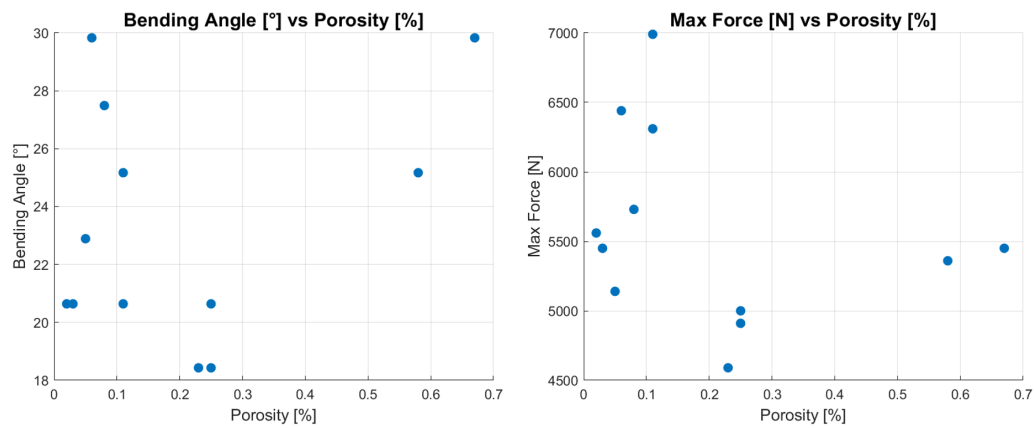


Figure 2.10: Data for bending angle (left) and max force (right).

The correlation coefficient (r value) for bending angle and porosity is 0.3, while for max force and porosity is -0.29. This indicates a vague trend for both cases, but not strong enough to make reliable predictions. In order to determine any relationship, further evaluation is required with more data points [49, 50, 51].

Chapter 3

Automated tensile system

Several potential expansions of the system can be considered. One approach is to adapt the system to accommodate a wider range of sample geometries, enabling broader application across different use cases. Another possibility is to increase the number of automated steps performed by the robot, thereby reducing operator involvement. For example, implementing automatic sample labelling could streamline the workflow. A third option is to enhance the system's functionality by enabling additional types of mechanical tests through hardware and software modifications.

Regardless of the chosen expansion path, it is essential that the current system remains fully operational and available for use at all times.

Even if the main focus of the project was dedicated to the development of the automated three point bending test cell, other test types were also examined for automation. Since the tensile test provides some of the most useful properties, such as Young's modulus and yield strength [52], it has the highest priority to be automated as well.

Due to the limited time-frame of the project, this second section is less developed compared to the bending test cell. Nevertheless, the progress achieved and the insights gained are presented in the following discussion.

3.1 Requirements

Several key requirements from the bending test remain applicable to the tensile test. For instance, preserving samples for post-test fracture surface analysis and maintaining traceability through QR-code tracking are still essential. However, notable differences arise in the execution of tensile tests, particularly in the monitoring of the strain field. This process involves applying a dotted pattern to the sample surface and capturing its deformation

using Digital Image Correlation (DIC) software. For accurate results, the surface must exhibit suitable properties such as elasticity and adhesion [53]. To ensure optimal conditions, the test should be conducted between 5 and 10 minutes after applying the paint, and no later than one hour afterward. Consequently, the sample surface becomes highly sensitive and must be handled with particular care throughout the process [54, 55].

3.2 Test Setup

The tensile testing machine consists of two grippers, which can regulate the clamping pressure during specimen insertion, and a camera system for strain field measurement using DIC.

A specific sequence must be followed to ensure accurate test execution. First, the specimen is inserted into the lower gripper and securely clamped. Next, the testing procedure is initiated, during which the upper gripper is lowered into position via a predefined approach path. While the upper gripper is closing, the crosshead position must be carefully adjusted to avoid introducing compressive stresses before the test begins. Once all conditions are met, the machine proceeds with the test, applying load until the specimen fractures and the two separated sections of the specimen remain attached to the respective grippers.

After fracture, the crosshead moves to a preset position to facilitate safe handling by the robotic system. The test procedure itself is customizable and can be configured in the software according to the selected testing standard.

3.3 Robot Setup

The same robot as described in Section 2.4.1 is used for this setup. Mounted on a movable platform, it can be relocated to predefined positions to perform different types of tests. This configuration ensures that the bending test cell remains available for use.

To accommodate different testing procedures, the robot's code was modified. Upon system startup, the user is prompted to select the desired test type, currently either bending or tensile. Based on this selection, the corresponding section of the control code is activated to invoke the appropriate workflow.

The preferred test type is then communicated to the Python script via the PLC, ensuring that the appropriate part of the code is executed.

3.3.1 Tool head

Accommodating the new sample geometry (see Appendix A.2) required the development of a specialized end effector. This component was designed to securely grip tensile test specimens without damaging the painted surface used for strain measurement. The padded jaws from Piab (specifically the GRZ 10-10 NPC-P [56]) was mounted at the tip of the end effector, and a distance laser sensor was integrated to monitor the separation between the gripper jaws in real time.

The final version of the end effector used in this setup is shown in Figure 3.1.



Figure 3.1: Tool used to handle tensile test samples.

3.4 Robot-Machine communication

The robot must be capable of controlling the machine's grippers during the opening and closing sequence, as well as initiating the test at the appropriate moment. This is achieved by activating the corresponding digital signal for a defined duration. In this setup, a signal duration of three seconds is sufficient to ensure secure gripping of the samples.

Communication between the robot and the testing machine is managed through an I/O interface, which is used to initiate and monitor the test execution process.

3.5 Disposal

The final step in the testing cycle involves removing both fragments of the broken specimen from the machine's grippers. This task can be particularly challenging due to the unpredictable location of the fracture along the sample's length. To address this issue, the distance laser sensor is used. The robot follows a predefined trajectory toward the lower portion of the sample until the sensor detects the correct distance. Once the part is securely gripped, the lower machine gripper is released, and the fragment is placed in a predefined disposal location.

The same procedure is then applied to remove the upper fragment of the sample.

Chapter 4

Discussion

In the pursuit of increased efficiency, consistency, and data quality in mechanical testing, robotic systems have been developed to automate both bending and tensile test procedures. These automated cells are designed to minimize manual intervention, reduce testing time, and enhance repeatability while maintaining accuracy comparable to traditional methods.

Separate discussions will be provided for the bending and tensile test setups, along with an evaluation of potential improvements that could be implemented in both systems.

4.1 Bending test system

The robotic station for automatic three-point bending tests has been successfully implemented and is now suitable for routine operation. The operator is responsible for pre-test tasks, including cutting samples from the components to be analysed, labelling each sample and stacking them in the designated storage location. All subsequent operations are carried out autonomously by the robotic cell, with data output automatically organized and visualized in the corresponding Excel file, or loaded directly into the material database. The system is optimized for processing standard-sized samples with thicknesses between 1.5 and 4 mm, and is not intended for use with custom-tailored or non-standard samples.

Robot movements are smooth and consistent, with minimal errors encountered during normal operation. The integration of functionalities such as force control and external sensors enhances system reliability by enabling adaptive responses to specific situations. The thickness measurement approach is valid, and the laser sensor provides sufficient precision. However, limitations arise in the interpretation of analog signals by the PLC, leading

to a reduction in resolution.

The laser measurement provides a sufficient thickness estimation while also aligning the sample in the correct position and orientation. As a consequence, only flat specimens can be processed and the other specimen dimensions neglected.

The use of a PLC is fundamental to the system's operation, serving as a critical interface for reading analog signals, enabling communication, and exchanging data with the PC. Its flexibility and programmability make it an essential component in automation. Combined with the Python program, the data handling process becomes straightforward and reliable, ensuring efficient organization and analysis.

The accuracy of the results has been validated, showing performance comparable to that of manual testing. However, the primary benefit of automation lies in improved efficiency and throughput. While a human operator typically requires 5 to 10 minutes to measure a sample, configure the testing machine, execute the procedure, and record the results, the automated system completes the same process in under 3 minutes. Under continuous 24-hour operation, the system is capable of processing approximately 400 to 500 tests per day. Additionally, it enables operators to focus on higher-value tasks by minimizing manual involvement.

4.1.1 Improvements

Although the cell is functional and operating, resources and time would allow more precision, faster execution and more processing capabilities. Possible improvements in order of importance are discussed here.

The LOGO PLC used is a limiting factor when it comes to measurement precision and number of signals available. Another downside of the LOGO PLC, is the unavailability of two-way communication with the computer. By using a more advanced controller such as the S7-300, more functionalities are available, and signals could be manipulated through the Python code directly.

The quality of the machine itself is also a factor in determining quality of results. For this reason, modern equipment is required to acquire accurate data.

To cut down on cycle time, a possibility is to perform operations with the robot when it is idle. During testing, a new sample can be processed by overlapping two cycles. Although an easy implementation, it could face issues during data logging and the current methodology is not suitable.

Occasionally, three-point bending tests must be conducted on small or irregularly shaped components. Currently, such tests are either not feasible or

pose significant risks when performed on non-standard samples. Enhancing the system's adaptability to accommodate a wider variety of sample geometries would significantly broaden its range of potential applications.

Adding functionalities is an ongoing process. Introducing diverse test types to be executed in line would allow to gather more data on the single specimen, such as hardness evaluation. Moreover, incorporating automated sample labelling could help reduce manual tasks and improve overall workflow efficiency.

4.2 Tensile test system

The robotic setup was relocated near the tensile testing machine and adapted to support multiple test types. Equipped with a newly designed end effector, the system can perform precise operations by continuously measuring the distance between the gripper and surrounding surfaces in real time.

The automated testing cycle includes loading the specimen into the machine's grippers, initiating the test procedure, and removing the fractured parts upon completion.

4.2.1 Improvements

Although the basic operating cycle is functional, the system is still under development and requires further refinement before it can be considered fully operational. First, a dedicated storage and pick-up station must be implemented to enable continuous operation. This station could also incorporate a mechanism for applying the speckle pattern to the sample surface, further reducing the need for manual intervention.

Additionally, a measurement station should be integrated to capture not only thickness but also other relevant sample dimensions.

Another critical aspect is data logging. Each test generates a substantial amount of data, and managing this information efficiently remains a challenge, particularly in automating the setup and calibration of the DIC camera for each new test.

At this stage, no safety features have been implemented, making continuous monitoring by an operator essential.

Chapter 5

Conclusion

Automation of mechanical testing was explored with the aim of finding and resolving issues arising from the use of a robot in sample and data handling. To do so, a system capable of processing 3-point bending test samples has been developed and the reliability of the system verified. Given the repetitive nature of the task and the precision required in testing, the use of a robot arm is justified and suitable for the job. The attempt at automating tensile test required first adaptation of the system to be able to run multiple test types. Consequently, new problems arise from dealing with broken samples and equipment limitations, which lead to creative solution and a deeper understanding of the instrument.

Bibliography

- [1] ABB. Product specification: Crb15000. Technical Report 3HAC077390-001, ABB, February 2025.
- [2] Dennis M. Dimiduk, Elizabeth A. Holm, and Stephen R. Niezgoda. Perspectives on the impact of machine learning, deep learning, and artificial intelligence on materials, processes, and structures engineering. *Integrating materials and manufacturing innovation*, 7(3):157–172, August 2018.
- [3] Howard Kuhn and Dana Medlin. *Mechanical testing and evaluation*. January 2000.
- [4] Kyriakos Komvopoulos. *Mechanical testing of engineering materials*. Cognella Academic Publishing, January 2017.
- [5] ML Renauld, JA Scott, LH Favrow, MA McGaw, MD Marotta, and DM Nissley. An automated facility for advanced testing of materials. *APPLICATIONS OF AUTOMATION TECHNOLOGY IN FATIGUE AND FRACTURE TESTING AND ANALYSIS, VOL 4*, 1411:16–26, 2002.
- [6] ZwickRoell GmbH Co. KG. Robotic Materials Testing: Optimum quality assurance at reduced cost. *ZwickRoell GmbH Co. KG*, June 2010.
- [7] W. Lim and H.-K Kim. Design and development of a miniaturised tensile testing machine. 15:48–53, January 2013.
- [8] Miguel Hernández-Del-Valle, Christina Schenk, Lucía Echevarría-Pastrana, Burcu Ozdemir, Enrique Dios-Lázaro, Jorge Ilarraza-Zuazo, De-Yi Wang, and Maciej Haranczyk. Robotically automated 3D printing and testing of thermoplastic material specimens. *Digital Discovery*, 2(6):1969–1979, January 2023.

- [9] Julia Głowacka, Łukasz Derpeński, Miłosz Frydrych, Bogna Sztorch, Błażej Bartoszewicz, and Robert E. Przekop. Robotization of Three-Point bending mechanical tests using PLA/TPU blends as an example in the 0–100 *Materials*, 16(21):6927, October 2023.
- [10] Instron. At6 6-axis robotic automated testing system. <https://www.instron.com/en/products/testing-systems/automated-testing-systems/at6-6-axis-robotic-automated-testing-system>, n.d.
- [11] ZwickRoell. robotest r – robotic testing system. <https://www.zwickroell.com/products/automated-testing-systems/robotest-r-robotic-testing-system/>, n.d.
- [12] ZwickRoell GmbH Co. KG. Robotic materials testing: Optimum quality assurance at reduced cost. <https://www.azom.com/article.aspx?ArticleID=5283>, November 2022.
- [13] Zwick GmbH & Co KG. Robotic testing of composites. <https://www.iem.eu/article/robotic-testing-of-composites/>, December 2009.
- [14] Cezary Zieliński. *General Robotic System Software Design Methodology*. January 2019.
- [15] Ralf Neuhaus and Michael Borsutzki. Plättchen-Biegeversuch nach VDA 238-100. *Materials Testing*, 55(9):654–659, October 2013.
- [16] K. Cheong, K. Omer, C. Butcher, R. George, and J. Dykeman. Evaluation of the vda 238-100 tight radius bending test using digital image correlation strain measurement. *Journal of Physics: Conference Series*, 896(1):012075, 2017.
- [17] ABB. Product manual – crb15000 (3hac077389-001). <https://search.abb.com/library/Download.aspx?DocumentID=3HAC077389-001>, 2021.
- [18] ABB robotics. Gofa™ CRB 15000 technical data. 2023.
- [19] ABB. Flexpendant product manual – 3hac093167-001 (rev. a). <https://library.abb.com/d/3HAC093167-001>, 2024.
- [20] Weidong Li, Yudie Hu, Yong Zhou, and D. Pham. Safe human–robot collaboration for industrial settings: a survey. *Journal of Intelligent Manufacturing*, 35, June 2023.

- [21] International Organization for Standardization (ISO). ISO 10218-1:2025 – Robotics— Safety requirements— Part1: Industrial robots. <https://www.iso.org/standard/73933.html>, 2025.
- [22] International Organization for Standardization (ISO). ISO/TS 15066:2016 – Robots and robotic devices— Collaborative robots. <https://www.iso.org/standard/62996.html>, 2016.
- [23] Peter Chemweno, Liliane Pintelon, and Wilm Decre. Orienting safety assurance with outcomes of hazard analysis and risk assessment: A review of the iso 15066 standard for collaborative robot systems. *Safety Science*, 129:104832, 2020.
- [24] Robin Jeanne Kirschner, Nico Mansfeld, Saeed Abdolshah, and Sami Haddadin. Iso/ts 15066: How different interpretations affect risk assessment. <https://arxiv.org/abs/2203.02706>, 2022.
- [25] Seungbin Moon and Gurvinder Virk. Survey on iso standards for industrial and service robots. *ICCAS-SICE*, pages 1878 – 1881, September 2009.
- [26] ABB. Technical reference manual – rapid overview (3hac050947-001, rev.u). <https://library.abb.com/d/3HAC050947-001>, 2025.
- [27] David Marshall. Abb’s robotstudio software enables cost savings in robot-programming. *Industrial Robot*, 29(6), 2002.
- [28] C. Connolly. Technology and applications of abb robotstudio. *Industrial Robot*, 36(6):540–545, 2009.
- [29] Robot System Products AB. Technical description – tool changer atc18/mtc18 (m0121-1). https://downloads.robotssystemproducts.com/manuals/M01211_Technical_Description_Tool_changer_ATC18_MTC18.pdf, 2025.
- [30] V. Mathumitha, Krishna Kumar, M. Ganesh Reddy, K. L. Sekhar, and Kuruba Pramod. Enhanced detection and decoding of qr code and barcode using machine learning. *International Journal of Research in Engineering, Science and Management*, 7(4):22–27, April 2024.
- [31] wenglor sensoric GmbH. Cp35mht80 laser distance sensor (triangulation). <https://www.wenglor.com/en/Sensors/Photoelectronic-Sensors/Laser-Sensors-for-distance-measurement/>

Laser-Distance-Sensors-Triangulation/
Laser-Distance-Sensor-Triangulation/p/CP35MHT80, 2025.

- [32] Jorge Santolaria, David Guillomía, Carlos Cajal, José A. Albajez, and Juan J. Aguilar. Modelling and calibration technique of laser triangulation sensors for integration in robot arms and articulated arm coordinate measuring machines. *Sensors*, September(9):7374–7396, 2009.
- [33] Gerardo Antonio Idrobo-Pizo, José Maurício S. T. Motta, and Renato Coral Sampaio. A calibration method for a laser triangulation scanner mounted on a robot arm for surface mapping. *Sensors*, 19(8), 2019.
- [34] AZoSensors. Thickness measurement with integrated width determination. <https://www.azosensors.com/article.aspx?ArticleID=843>, July 2017.
- [35] MTS Systems Corporation. *Insight Material Testing Systems – Technical Description (Manual Part No. 100-146-666E)*. MTS Systems Corporation, May 2007.
- [36] Patrick Larour, Benjamin Hackl, and Franz Leomann. Sensitivity analysis on the calculated bending angle in the instrumented bending test. June 2013.
- [37] Patrick Larour, Benjamin Hackl, Franz Leomann, and Katarzyna Benedykt. Bending angle calculation in the instrumented three-point bending test. November 2012.
- [38] Siemens AG. LOGO!24CE Logic Module Data Sheet (6ED1052-1CC08-0BA0). <https://docs.rs-online.com/eb10/0900766b8164240a.pdf>, March 2021.
- [39] Diwaker Gupta and Amy Peterson. Statistical methods for materials testing. Technical Report MN/RC 2009-41, Industrial & Systems Engineering Program University of Minnesota 111 Church Street SE Minneapolis, MN 55455, December 2009.
- [40] P.J. Blau. Lessons learned from the test-to-test variability of different types of wear data. *Wear*, 376-377:1830–1840, 2017. 21st International Conference on Wear of Materials.

- [41] Elena Fiorese, Franco Bonollo, Giulio Timelli, Lars Arnberg, and Elisabetta Gariboldi. New classification of defects and imperfections for aluminum alloy castings. *International Journal of Metalcasting*, 9(1):55–66, January 2015.
- [42] Hélder Puga. Casting and forming of advanced aluminum alloys. *Metals*, 10(4):494, April 2020.
- [43] A. García Pérez, M. J. Gómez Silva, and A. De La Escalera Hueso. Automated defect recognition of castings defects using neural networks. *Journal of Nondestructive Evaluation*, 41(1), December 2021.
- [44] Choong Do Lee. Effects of microporosity on tensile properties of A356 aluminum alloy. *Materials Science and Engineering A*, 464(1-2):249–254, February 2007.
- [45] Jianli Zhao, Yilin Wang, Xiaowei Wang, and Yisheng Zhang. An experimental investigation of the material properties of the A356 aluminum alloy power fittings in the vacuum Die-Casting process. *Materials*, 17(6):1242, March 2024.
- [46] Ewan Lordan, Jaime Lazaro-Nebreda, Yijie Zhang, Kun Dou, Paul Blake, and Zhongyun Fan. On the relationship between internal porosity and the tensile ductility of aluminium alloy die-castings. *Materials Science and Engineering A*, 778:139107, February 2020.
- [47] D. G. Eskin. *Structure and casting defects of aluminum billets produced by Direct-Chill Casting*. January 2015.
- [48] Filip Nikolić, Ivan Štajduhar, and Marko Čanađija. Casting Defects Detection in Aluminum Alloys Using Deep Learning: a Classification Approach. *International Journal of Metalcasting*, 17(1):386–398, March 2022. <https://doi.org/10.1007/s40962-022-00777x>.
- [49] Roemer J Janse, Tiny Hoekstra, Kitty J Jager, Carmine Zoccali, Giovanni Tripepi, Friedo W Dekker, and Merel Van Diepen. Conducting correlation analysis: important limitations and pitfalls. *Clinical Kidney Journal*, 14(11):2332–2337, May 2021.
- [50] Avijit Hazra and Nithya Gogtay. Biostatistics series module 6: Correlation and linear regression. *Indian Journal of Dermatology*, 61(6):593, January 2016.

- [51] Marie-Therese Puth, Markus Neuhäuser, and Graeme D. Ruxton. Effective use of Pearson's product-moment correlation coefficient. *Animal Behaviour*, 93:183–189, June 2014.
- [52] J.R. Davis. *Tensile Testing*. ASM International, December 2004.
- [53] Paweł Bogusz, Wiesław Krasoń, and Kamil Pazur. Application of digital image correlation for strain mapping of structural elements and materials. *Materials*, 17(11):2577, May 2024.
- [54] Michał Marcin Bakalarz and Paweł Piotr Tworzewski. Application of digital image correlation to evaluate strain, stiffness and ductility of full-scale lvl beams strengthened by cfrp. *Materials*, 16(3), 2023.
- [55] Akshat Agha. Effectiveness of 2D digital image correlation in capturing the fracture behavior of sheet metal alloys. *SAE International Journal of Materials and Manufacturing*, December 2022.
- [56] Piab Group. Gripdon för gjutdelar – sprue/part grippers (produktnummer0213202). <https://www.piab.com/sv-se/komponenterforrobotverktygeoat/formsprutningavplast/gripdonforgjutdelar/0213202>, 2025.

Appendix A

Graphs and Images

A.1 Data for Section 2.13

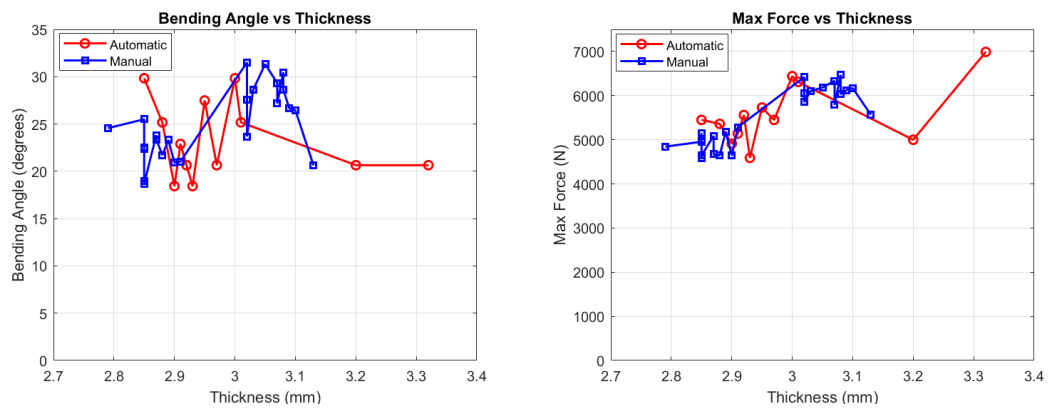


Figure A.1: Bending angle and peak force raw data, plotted with thickness

A.2 Test samples



Figure A.2: Aluminium test sample for bending (left) and tensile(right).

A.3 Robot platform and bending machine

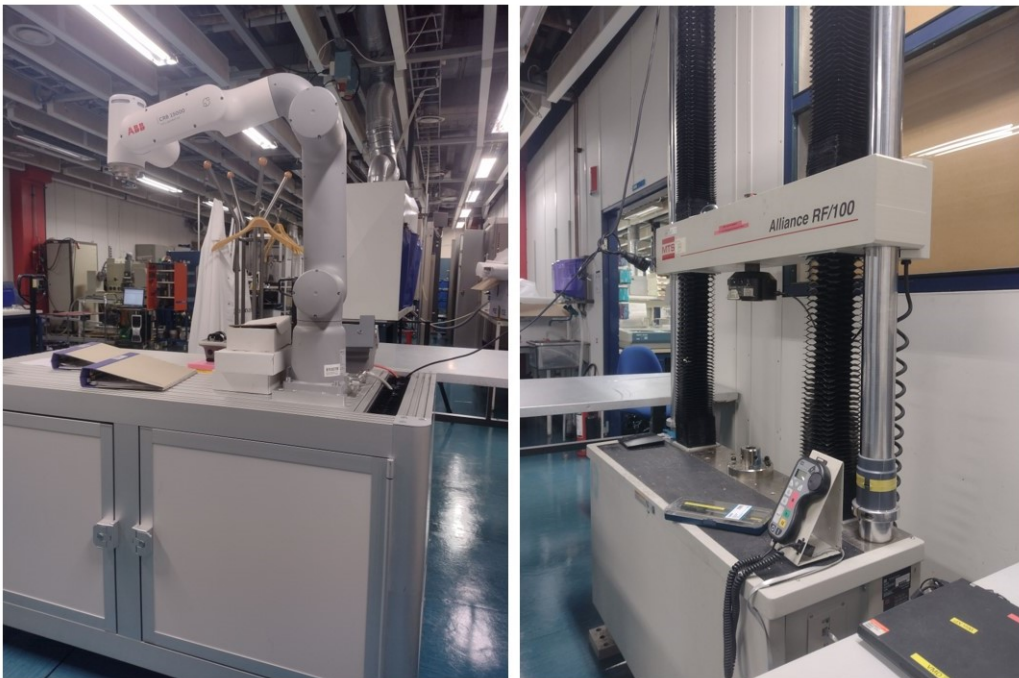


Figure A.3: Robot arm mounted on movable platform (left) and bending machine (right).

Appendix B

Calibration

B.1 Calibration for laser sensor

The calibration line is used to convert the value V , ranging from 0 to 1000 in the PLC memory, to a thickness measurement t_m in mm. Since only a difference in values is representative of a thickness, the line $y = mx + q$ is simplified to $y = mx$. The final relationship used is:

$$t_m = V/3.51 \tag{B.1}$$

(see Appendix C.2, Line 92).

B.2 Calibration of force and crosshead displacement

The calibration for force applied and crosshead displacement is done inside the bending machine software TestWorks 4 and can be customized.

$$F_{max} = 10 \cdot V_F, \tag{B.2}$$

$$S = V_S/3.33 \tag{B.3}$$

F_{max} is the peak force, S the crosshead displacement after preload, and V_F and V_S the corresponding values from 0 to 1000 in the PLC memory (see Appendix C.2, Lines 102 and 123).

Appendix C

Full code

C.1 PLC code

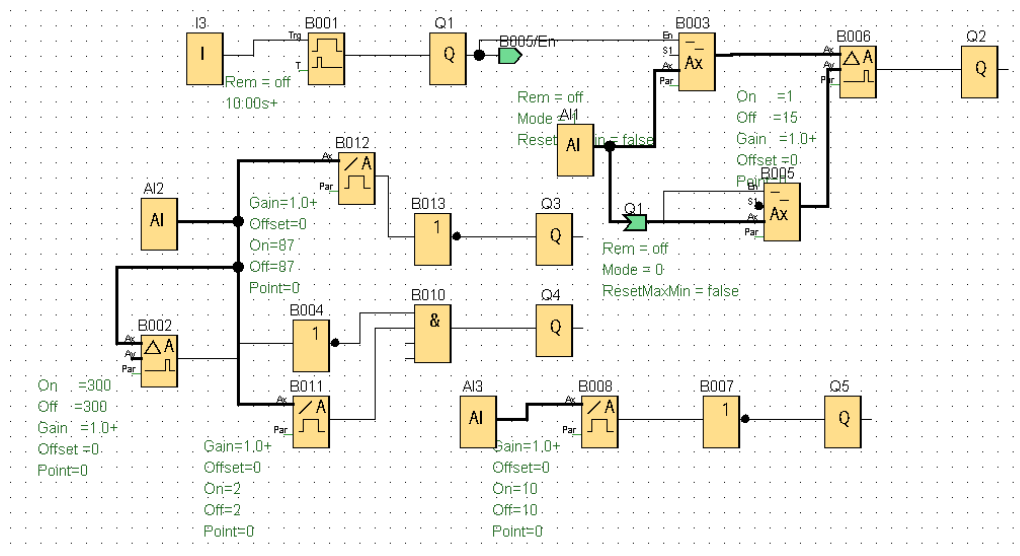


Figure C.1: FBD code for PLC

C.2 Python Code

```
1 import cv2
2 import time
3 import openpyxl
4 import os
5 import pandas as pd
6 import math
7 #import snap7
8 #from snap7.util import set_bool
9 startTime=time.time()
10 count=0
11 # Get the path to the Desktop (works for both Windows and macOS/Linux)
```

```

12 desktop_path = os.path.join(os.path.expanduser("~"), "Desktop", "qr_codes.xlsx")
13 # Check if the Excel file already exists
14 if os.path.exists(desktop_path):
15     # Open the existing Excel file
16     wb = openpyxl.load_workbook(desktop_path)
17     ws = wb.active # Get the active sheet
18 else:
19     # Create a new workbook if the file doesn't exist
20     wb = openpyxl.Workbook()
21     ws = wb.active
22     ws.title = "QR Codes"
23     PLC_IP= '192.168.0.1'
24     DB_Number= 1
25     Byte_index= 0
26     Bit_Index=0
27     #plc= snap7.client.Client()
28     #plc.connect(PLC_IP, 0, 1)
29     bending=True
30     istheFirst=0
31 # Open the camera (0 is usually the default webcam)
32 cap = cv2.VideoCapture(0)
33 # Create a QRCodeDetector object
34 qr_detector = cv2.QRCodeDetector()
35 preloaddisplacement= 591
36 while True:
37     #this part is for adding values in the excel
38     file_path = '\u202aC:\\Users\\S90\\Desktop\\LogFile.csv'
39     cleaned_path = file_path.replace('\u202a', '')
40     df = pd.read_csv(cleaned_path)
41     # Capture frame-by-frame
42     ret, frame = cap.read()
43     if not ret:
44         print("Failed to capture image")
45         break
46     # Detect and decode the QR code
47     try:
48         data, bbox, _ = qr_detector.detectAndDecode(frame)
49         # If a QR code is detected and bbox is valid
50         if data and bbox is not None:
51             # Put the decoded data on the frame
52             cv2.putText(frame, f'Data: {data}', (50, 50), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
53             # Add a 10-second delay after detecting the QR code
54             print("QR code detected! Creating new line...")
55             istheFirst=istheFirst +1
56             #data1=bytearray(plc.db_read(DB_Number, Byte_index, 1))
57             #set_bool(data1, 0, Bit_Index, True)
58             #plc.db_write(DB_Number, Byte_index, data1) #changes the value to 1 and activates Q7
59             time.sleep(10) # Delay for 10 seconds
60             last_row = df.iloc[-1] # -1 gives the last row
61             if istheFirst==1:
62                 if last_row['I4']==1: #if the test type is confirmed to be BENDING
63                     ws.append(["New batch - test data - 3 point bending"])
64                     ws.append(["Sample number", "thickness ", "bending angle", "max force ", "slope"])
65                 elif last_row['I4']==0: #if testtype is confirmed to be tensile
66                     ws.append(["New batch - test data - tensile"])
67                     ws.append(["Sample number", "thickness [mm]", "max force [N]"])
68                 #set_bool(data1, 0, Bit_Index, False)
69             #plc.db_write(DB_Number, Byte_index, data1)
70             # Append the QR code data to the Excel file
71             ws.append([data]) # Add the data in a new row of the excel
72             # Save the workbook to the desktop
73             wb.save(desktop_path)
74             print(f"QR code data saved: {data}")
75         except Exception as e:
76             print(f"Error detecting or decoding QR code: {e}")
77             # Display the frame with QR code data
78             cv2.imshow("QR Code Scanner", frame)
79             last_row = df.iloc[-1] # -1 gives the last row
80 #execute this only if it's a bending test
81 if last_row['I4']==1: #if the test type is confirmed to be BENDING
82     #Thickness measurement
83     #this are read from the csv file
84     last_row = df.iloc[-1] # -1 gives the last row
85     last_last_row = df.iloc[-2] #gives the row before the last one
86     last_last_last_row= df.iloc[-3] #third row from the bottom
87     diff= last_last_row['C4']-last_row['C4'] #this are the correct values for AI1, laser measurement
88     #print(diff)
89     if diff>4 and diff<10: #condition to log the difference
90         #write the diff in the excel
91         print("Log new thickness measurement", diff)
92         thickness=diff/3.51 #from the calibration, now is in mm
93         last_row_excel = ws.max_row #gets the last row of the excel file
94         ws.cell(row=last_row_excel, column=2, value=thickness)

```

```

95     wb.save(desktop_path)
96     time.sleep(5)
97 #logs the max force in excel
98     diff4= last_row['AI3']-last_last_row['AI3']#this are the correct values for AI4, force applied
99     diff5= last_last_row['AI3']- last_last_last_row['AI3']
100     if diff4<0 and diff5>0:
101         print("log new max force", last_last_row['AI3'])
102         maxforce=last_last_row['AI3']*10 #converts the value in N
103         last_row_excel = ws.max_row #gets the last row of the excel file
104         ws.cell(row=last_row_excel , column=4, value=maxforce)
105         wb.save(desktop_path)
106     diff3= last_row['AI3']-last_last_row['AI3']#this are the correct values for AI4, force applied
107     if diff3>0 and last_last_row['AI3']==0:
108         preloaddisplacement=last_last_row['AI2'] #crossehad value (in 0-1000)at preload
109         print("new displacement", preloaddisplacement)
110     if (last_row['AI2']-last_last_row['AI2']) !=0:
111         m=((last_row['AI3']-last_last_row['AI3'])*10)/((last_row['AI2']-last_last_row['AI2'])/3.333)
112         if diff3>0 and last_last_row['AI3']<10 and m>0:
113             print("new slope value:", m)
114             last_row_excel = ws.max_row #gets the last row of the excel file
115             ws.cell(row=last_row_excel , column=5, value=m)
116             wb.save(desktop_path)
117 #Bending angle measurement
118 #this are read from the csv file
119     diff2= last_row['AI3']-last_last_row['AI3']#this are the correct values for AI4, force applied
120     if diff2<0 and last_row['AI3']<2: #by making the 2 smaller the measurement is more precise
121         print("log new crosshead position", last_row['AI2'] )
122         disp= last_row['AI2'] -preloaddisplacement #crosshead displacement form preload
123         disp=disp/3.333 #now in mm
124         r=0.4 #mm, punch radius
125         D=30 #mm, roller diameter
126         L=6.09 #mm, roller distance
127         tm=thickness #sample thickness
128         c=(D/2)+tm+r
129         p= (D/2)+(L/2)
130         W= math.sqrt(p**2+(disp-c)**2-c**2)
131         angle= math.degrees(2*math.acos((W*p-c*(disp-c))/(p**2+(disp-c)**2)))
132         last_row_excel = ws.max_row #gets the last row of the excel file
133         ws.cell(row=last_row_excel , column=3, value=angle)
134         wb.save(desktop_path)
135 #execute only if it's a tensile test
136     elif last_row['I4']==0: #if testtype is confirmed to be tensile
137         #this are read from the csv file
138         last_row = df.iloc[-1] # -1 gives the last row
139         last_last_row = df.iloc[-2] #gives the row before the last one
140         last_last_last_row= df.iloc[-3] #third row from the bottom
141         diff5= last_last_row['C4']-last_row['C4']
142         if diff5<0 and last_last_row['C4']==0:
143             startmeasure=time.time()
144         if diff5>0 and last_row['C4']==0:
145             endmeasure=time.time()
146         thickness_=endmeasure-startmeasure #this is the time, needs to be converted to mm
147         print("Logging new thickness value", thickness_)
148         last_row_excel = ws.max_row #gets the last row of the excel file
149         ws.cell(row=last_row_excel , column=3, value=thickness_)
150         wb.save(desktop_path)
151 # Break the loop when 'q' is pressed
152     if cv2.waitKey(1) & 0xFF == ord('q'):
153         break
154 #plc.disconnect()
155 # Release the camera and close all OpenCV windows
156 cap.release()
157 cv2.destroyAllWindows()

```

C.3 Robot code

```

1 MODULE MainModule
2 CONST robtargt pickUp := [[993.12,93.50,136.26], [0.00184305,0.699293,0.714779,-0.00874605], [0,-1,1,0], [9E+09,9E+09,9E
+09,9E+09,9E+09,9E+09]]; !cups
3 CONST robtargt qrCode := [[949.20,-83.98,47.41], [0.00163261,0.699483,0.714591,-0.00892689], [-1,-1,0,0], [9E+09,9E+09,9E
+09,9E+09,9E+09,9E+09]]; !cups
4 CONST robtargt before_qrCode := [[782.50,-83.94,47.35], [0.00158366,0.699486,0.714589,-0.0089546], [-1,-1,0,0], [9E+09,9
E+09,9E+09,9E+09,9E+09,9E+09]]; !cups
5 CONST robtargt before_pickUp := [[969.36,93.38,162.60], [0.00181894,0.699326,0.714746,-0.00878559], [0,-1,1,0], [9E+09,9
E+09,9E+09,9E+09,9E+09,9E+09]]; !cups
6 CONST robtargt after_pickUp := [[841.00,93.28,162.49], [0.00175594,0.699364,0.714709,-0.00884688], [0,-1,1,0], [9E+09,9E
+09,9E+09,9E+09,9E+09,9E+09]]; !cups
7 VAR robtargt currentPos;
8 CONST robtargt before_measurement:=
[[729.36,-267.85,298.45], [0.00161964,0.699505,0.714571,-0.00890977], [-1,-1,0,0], [9E+09,9E+09,9E+09,9E+09,9E+09,9

```

```

E+09]; !cups
9  CONST robtargt measurement:= [[748.22,-368.08,203.29],[0.113901,-0.670866,-0.668719,0.299633],[-1,-1,0],[9E+09,9E
+09,9E+09,9E+09,9E+09,9E+09]]; !cups
10 CONST robtargt after_measurement := [[807.31,-314.79,110.04],[0.128322,-0.689958,-0.626499,0.339101],[-1,-1,0],[9E
+09,9E+09,9E+09,9E+09,9E+09,9E+09]]; !cups
11 CONST robtargt after_testing:= [[-814.21,-103.97,81.26],[0.00509138,-0.711578,-0.701649,-0.0363391],[-2,-1,-1,0],[9E
+09,9E+09,9E+09,9E+09,9E+09,9E+09]]; !gripper
12 CONST robtargt disposal_pickUp :=
[[-1052.97,-113.96,17.09],[0.00502324,-0.706542,-0.70764,-0.00431698],[-2,-1,-1,0],[9E+09,9E+09,9E+09,9E+09,9E
+09,9E+09]]; !gripper
13 CONST robtargt disposal:= [[-86.18,-499.14,18.58],[0.0190478,-0.715392,0.698397,0.00969954],[-2,-1,1,0],[9E+09,9E
+09,9E+09,9E+09,9E+09,9E+09]]; !gripper
14 PERS tooldata cups := [TRUE,[[0,250,70.5],[1,0,0,0]],[0.3,[0,0,45],[1,0,0,0],0.000076,0.00569,0.00577]];
15 PERS tooldata gripper := [TRUE,[[0,-242,115.8],[1,0,0,0]],[0.3,[0,0,45],[1,0,0,0],0.000076,0.00569,0.00577]];
16 CONST robtargt before_testing := [[-824.05,-107.80,126.28],[0.0363917,-0.703476,0.709768,0.00503275],[-2,-1,1,0],[9E
+09,9E+09,9E+09,9E+09,9E+09,9E+09]]; !cups
17 CONST robtargt testing := [[-1026.92,-111.82,8.26],[0.00132636,0.779113,-0.626878,0.00233852],[-2,0,0],[9E+09,9E+09,9E
+09,9E+09,9E+09,9E+09]]; !cups
18 CONST robtargt midwaytesting1 := [[633.51,-617.37,426.98],[0.00765945,0.850706,0.519315,-0.0809479],[-1,-1,0],[9E
+09,9E+09,9E+09,9E+09,9E+09,9E+09]]; !cups
19 CONST robtargt midwaytesting2 := [[-487.59,-738.07,426.94],[0.0457086,-0.986518,0.142026,0.067248],[-2,-1,0],[9E+09,9E
+09,9E+09,9E+09,9E+09,9E+09]]; !cups
20 CONST robtargt return1 := [[-329.57,-509.56,545.56],[0.0190071,-0.715366,0.698424,0.00970655],[-2,-1,1,0],[9E+09,9E
+09,9E+09,9E+09,9E+09,9E+09]]; !tool0
21 CONST robtargt return2 := [[300.53,-527.21,545.53],[0.0112175,-0.974053,0.225311,0.0181802],[-1,-1,1,0],[9E+09,9E+09,9E
+09,9E+09,9E+09,9E+09]]; !tool0
22 CONST robtargt return3 := [[365.46,-394.59,545.53],[0.021308,-0.739627,-0.672679,0.00112997],[-1,-1,0],[9E+09,9E+09,9E
+09,9E+09,9E+09,9E+09]]; !tool0
23 CONST robtargt return4 := [[446.03,79.01,587.76],[0.0213177,-0.739603,-0.672705,0.00115511],[0,0,1,0],[9E+09,9E+09,9E
+09,9E+09,9E+09,9E+09]]; !tool0
24 PERS loaddata my_load:= [0.001,[0,0,0.001],[1,0,0,0],0,0,0];
25 CONST robtargt after_qrCode := [[729.24,-91.73,228.01],[0.0017099,0.699398,0.714675,-0.00887841],[-1,-1,0],[9E+09,9E
+09,9E+09,9E+09,9E+09,9E+09]];
26 VAR num completedcycleCount_bending := 0;
27 VAR num completedcycleCout_tensile := 0;
28 VAR num n_first_stack:=12; !change to 0 if the user inputs it
29 CONST robtargt pickUp2 := [[987.17,183.74,127.66],[0.00180553,0.699512,0.714563,-0.00891453],[0,-1,1,0],[9E+09,9E
+09,9E+09,9E+09,9E+09,9E+09]];
30 VAR robtargt disposal_error;
31 CONST robtargt beforedisposal1 := [[-610.29,-125.48,154.84],[0.0221359,-0.975715,-0.216719,-0.0228719],[-2,-1,0],[9E
+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
32 VAR num testtype:=0;
33 CONST robtargt beforedisposal2 := [[-86.29,-474.07,149.95],[0.0189831,-0.715332,0.69846,0.0096874],[-2,-1,1,0],[9E
+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
34 VAR robtargt pOffsetTarget;
35 CONST robtargt tens_before_pickUp := [[861.46,26.96,126.66],[0.00420267,0.0314857,0.99943,-0.011459],[0,-1,0],[9E
+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
36 CONST robtargt tens_pickUp := [[870.14,88.35,42.13],[0.00960392,-0.0165421,0.999541,-0.0235006],[0,-1,0],[9E+09,9E
+09,9E+09,9E+09,9E+09,9E+09]];
37 CONST robtargt tens_after_pickUp := [[390.28,129.12,400.92],[0.444508,-0.52413,-0.510271,0.517033],[0,0,1,0],[9E+09,9E
+09,9E+09,9E+09,9E+09,9E+09]];
38 PERS tooldata softGripper := [TRUE,[[0,0,300],[1,0,0,0]],[0.1,[0,0,45],[1,0,0,0],0.000076,0.00569,0.00577]];
39 CONST robtargt tens_before_qrCode := [[842.16,-76.65,79.42],[0.0109854,0.717299,0.696658,-0.00529772],[-1,-1,0],[9E
+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
40 CONST robtargt tens_after_qrCode := [[662.54,-75.24,104.81],[0.421883,-0.512219,0.569136,-0.485521],[-1,-2,0],[9E
+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
41 CONST robtargt tens_qrCode := [[897.79,-76.68,47.80],[0.0109815,0.717302,0.696656,-0.00530085],[-1,-1,0],[9E+09,9E
+09,9E+09,9E+09,9E+09,9E+09]];
42 CONST robtargt tensbeforemeasurement := [[902.84,-171.58,166.97],[0.32982,0.169322,0.477976,-0.796296],[0,-2,0],[9E
+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
43 CONST robtargt tensaftermeasurement := [[873.84,-171.67,166.91],[0.329766,0.169317,0.477997,-0.796307],[0,-2,1,0],[9E
+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
44 CONST robtargt tenshome := [[505.76,-76.38,572.84],[0.0241064,0.72101,0.690225,-0.0561474],[-1,-1,0],[9E+09,9E+09,9E
+09,9E+09,9E+09,9E+09]];
45 CONST robtargt tens_midwaytesting1 := [[61.28,-693.78,528.06],[0.0958796,-0.0601222,0.720197,-0.684477],[-1,1,1,0],[9E
+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
46 CONST robtargt tens_midwaytesting2 := [[-636.49,-282.80,528.07],[0.319913,-0.369766,-0.620939,0.612669],[-2,1,1,0],[9E
+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
47 CONST robtargt tens_testing := [[-1089.38,-189.56,96.20],[0.330586,-0.37914,-0.600713,0.621378],[-2,1,1,0],[9E+09,9E
+09,9E+09,9E+09,9E+09,9E+09]];
48 CONST robtargt tens_beforeTesting := [[-1017.51,-315.88,96.18],[0.330597,-0.379166,-0.600709,0.62136],[2,-1,1,0],[9E
+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
49 CONST robtargt tens_duringTesting := [[-943.35,-118.66,96.13],[0.389514,-0.447151,-0.560054,0.57851],[-2,1,1,0],[9E
+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
50 CONST robtargt tens_disposal_upper :=
[[-1067.91,-183.78,139.27],[0.368677,-0.354257,-0.621869,0.593176],[-2,1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E
+09]];
51 CONST robtargt tens_disposal_lower := [[-1061.65,-173.36,86.28],[0.33724,-0.411471,-0.598387,0.599077],[-2,1,1,0],[9E
+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
52 CONST robtargt tens_disposal_general :=
[[-269.14,-129.66,52.68],[0.0175953,0.0412698,0.998137,-0.0413612],[-2,0,2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E
+09]];
53 CONST robtargt tens_disposal_upper_afterpickup :=

```

```

[[[-719.33,142.16,80.80],[0.364359,-0.365372,-0.621605,0.589366],[1,1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]]];
54 CONST robtarget tens_disposal_lower_afterpickup :=
[[[-719.33,142.16,80.80],[0.364359,-0.365372,-0.621605,0.589366],[1,1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]]];
55 !CONST robtarget tens_midwaydisposal :=
[[[-417.31,-178.46,356.09],[0.0685055,-0.211178,0.755584,-0.616283],[1,0,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]]];
56 CONST robtarget tens_midwaydisposal := [[-718.61,-247.19,322.68],[0.337264,-0.411542,-0.598363,0.599039],[-2,1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]]];
57 CONST robtarget tens_midwaydisposal2 := [[-805.92,-208.46,159.85],[0.341957,-0.36695,-0.643607,0.578086],[-2,1,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]]];
58 PROC main()
59 reset_all_scalableD0;
60 RESET ABB_Scalable_IO_0_D016; !has to reset testType from the previous cycle
61 IF completedcycleCount_bending=0 AND completedcycleCout_tensile=0 THEN
62 TPErase;
63 TPReadFK testtype, "What type of test will be executed? ", "Bending", "Tensile", "trial", stEmpty, stEmpty;
64 ENDF
65 IF testtype = 1 THEN
66 SET ABB_Scalable_IO_0_D016; !this is the verification for bending test
67 pickUp_procedure_bending;
68 qrCode_procedure_bending;
69 measurement_procedure_bending;
70 testing_procedure_bending;
71 disposal_procedure_bending;
72 return_procedure_bending;
73 ELSEIF testtype = 2 THEN
74 pickUp_procedure_tensile;
75 qrCode_procedure_tensile;
76 measurement_procedure_tensile;
77 testing_procedure_tensile;
78 disposal_procedure_tensile;
79 ELSEIF testtype= 3 THEN
80 MoveJ tens_before_pickUp, v50, z0, softGripper; !goes to home position
81 WaitTime 1;
82 currentPos := CRobT(); ! Get current position
83 WHILE ABB_Scalable_IO_0_DI6 = 0 DO !until laser detect correct position on the z
84 MoveL Offs(currentPos, -5, -5, 0), v50, z0, softGripper; ! Move in steps of 5 mm
85 currentPos := CRobT(); ! Update current position
86 ENDWHILE
87 WaitTime 1;
88 ELSE
89 ErrRaise "Custom error", 90001, "error", "error","error", "error", "error";
90 ENDF
91 ERROR !deals with errors in the entire code
92 TPWrite "An error occurred, trying reset...";
93 IF ABB_Scalable_IO_0_DI1 =1 THEN !detects if the vacuum is on(there's a sample attached)
94 MoveJ disposal_error, v50, z0, cups;
95 RESET ABB_Scalable_IO_0_D02; !deactivates vacuum
96 SET ABB_Scalable_IO_0_D03; !activates and deactivates blow off
97 WaitTime 1;
98 RESET ABB_Scalable_IO_0_D03;
99 ENDF
100 MoveJ wi_homePosition, v50, z0, tool0; !goes to home position
101 StartMoveRetry; !restats the program
102 ENDPROC
103 PROC pickUp_procedure_bending()
104 MoveL before_pickUp, v50, z0, cups;
105 IF completedcycleCount_bending=0 THEN
106 !TPReadNum n_first_stack, "How many samples are there in the first stack?";
107 ENDF
108 IF completedcycleCount_bending <n_first_stack THEN !picks up only the correct number of samples from the first
stack
109 MoveL pickUp, v20, z0, cups;
110 ELSE
111 MoveL pickUp2, v20, z0, cups;
112 ENDF
113 SET ABB_Scalable_IO_0_D02; !activates vacuum
114 currentPos := CRobT(); ! Get current position
115 WHILE ABB_Scalable_IO_0_DI1 = 0 DO !until vacuum is detected, goes down
116 MoveL Offs(currentPos, 0, 0, -10), v20, z0, cups; ! Move down in steps of 10 mm
117 currentPos := CRobT(); ! Update current position
118 ENDWHILE
119 IF completedcycleCount_bending <n_first_stack THEN
120 MoveL pickUp, v20,z0,cups; !goes back up straight, to not move the samples below
121 ELSE
122 MoveL pickUp2, v20, z0, cups;
123 ENDF
124 MoveL after_pickUp,v100,z0, cups;
125 ENDPROC
126 PROC qrCode_procedure_bending()
127 MoveL before_qrCode, v200, z0, cups;

```

```

128     MoveL qrCode, v200, z0, cups;
129     WaitTime 7; !reads QR code
130     MoveL before_qrCode, v200, z0, cups;
131     MoveL after_qrCode, v200, z0, cups;
132 ENDPROC
133 PROC measurement_procedure_bending()
134     MoveL before_measurement, v200, z0, cups;
135     MoveL measurement, v100, z0, cups;
136     SET ABB_Scalable_IO_0_D05; ! this is the output doMeasurement seen in the PLC as doMeasure, II
137     WaitTime 5;
138     RESET ABB_Scalable_IO_0_D02; !deactivates vacuum
139     SET ABB_Scalable_IO_0_D03; !activates and deactivates blow off
140     WaitTime 1;
141     RESET ABB_Scalable_IO_0_D03;
142     WaitTime 5; !waits for the measurement to take place until the end
143     !WaitDI ABB_Scalable_IO_0_DI3, 1, \MaxTime:=30; !waits the input continueTest_measurement from the PLC, maxtime 1 min,
        after calls the error handler !deactivated
144     MoveL after_measurement, v50, z0, cups; ! goes to pick up the sample after measurement
145     FCRefForce \Fz:=-5; ! Setup the force reference with 5N in Z-direction of the world frame
146     FCAct cups; ! Activate Force Control
147     FCRefStart; ! Start moving the robot to achieve the specified force
148     WaitTime 10;
149     SET ABB_Scalable_IO_0_D02; !activates vacuum
150     FCRefStop; ! Stop the reference values
151     FCDeact; ! Deactivate force control
152     MoveL before_measurement, v100, z0, cups;
153     RESET ABB_Scalable_IO_0_D05; !resets output doMeasurement to be ready for next cycle
154     RESET ABB_Scalable_IO_0_D05; !deactivates the plc circuit to do the measurement
155 ENDPROC
156 PROC testing_procedure_bending()
157     MoveJ midwaytesting1, v1000, z0, cups;
158     MoveJ midwaytesting2, v1000, z0, cups;
159     MoveJ before_testing, v1000, z0, cups; !movest into the testing area
160     MoveL testing, v50, z0, cups;
161     WaitTime 3;
162     RESET ABB_Scalable_IO_0_D02; !deactivates vacuum
163     SET ABB_Scalable_IO_0_D03; !activates and deactivates blow off
164     WaitTime 1;
165     RESET ABB_Scalable_IO_0_D03;
166     MoveL testing, v50, z0, cups;
167     MoveL before_testing, v50, z0, cups; !gets out of the way
168     WaitDI ABB_Scalable_IO_0_DI4, 1, \MaxTime:=120; !positon check
169     TPWrite "Position check passed, test starts in 5 seconds";
170     WaitTime 5;
171     SET ABB_Scalable_IO_0_D012; !this is start test
172     MoveJ after_testing, v100, z0, gripper; !turns the tool around
173     RESET ABB_Scalable_IO_0_D012; !this needed to be reset
174 ENDPROC
175 PROC disposal_procedure_bending()
176     WaitDI ABB_Scalable_IO_0_DI7, 1, \MaxTime:=300; !waits for the test to be finished
177     MoveL disposal_pickUp, v100, z0, gripper;
178     WaitTime 2;
179     SET ABB_Scalable_IO_0_D06; !closes the gripper
180     MoveL after_testing, v100, z0, gripper;
181     MoveL beforedisposal1, v100, z0, gripper;
182     MoveL beforedisposal2, v100, z0, gripper;
183     IF completedcycleCount_bending<6 THEN
184         pOffsetTarget := disposal; ! Copy original position
185         pOffsetTarget.trans := disposal.trans + [80*completedcycleCount_bending, 0, 0]; ! Add 60mm in X per completed cycle
186         MoveL pOffsetTarget, v50, z0, gripper;
187         RESET ABB_Scalable_IO_0_D06; !drops tested sample
188     ELSEIF completedcycleCount_bending>5 AND completedcycleCount_bending<12 THEN
189         pOffsetTarget := disposal; ! Copy original position
190         pOffsetTarget.trans := disposal.trans + [80*(completedcycleCount_bending-6), 40, 0]; ! Add 60mm in X per completed
            cycle
191         MoveL pOffsetTarget, v50, z0, gripper;
192         RESET ABB_Scalable_IO_0_D06; !drops tested sample
193     ELSE
194         pOffsetTarget := disposal; ! Copy original position
195         pOffsetTarget.trans := disposal.trans + [80*(completedcycleCount_bending-12), 80, 0]; ! Add 60mm in X per completed
            cycle
196         MoveL pOffsetTarget, v50, z0, gripper;
197         RESET ABB_Scalable_IO_0_D06; !drops tested sample
198     ENDIF
199 ENDPROC
200 PROC return_procedure_bending()
201     MoveL return1, v300, z0, tool0;
202     MoveJ return2, v300, z0, tool0;
203     MoveJ return3, v300, z0, tool0;
204     MoveJ return4, v300, z0, tool0;
205     completedcycleCount_bending := completedcycleCount_bending + 1; !counts the number of cycles
206 ENDPROC
207 PROC reset_all_scalableD0()

```

```

208 RESET ABB_Scalable_IO_0_D01;
209 RESET ABB_Scalable_IO_0_D02;
210 RESET ABB_Scalable_IO_0_D03;
211 RESET ABB_Scalable_IO_0_D04;
212 RESET ABB_Scalable_IO_0_D05;
213 RESET ABB_Scalable_IO_0_D06;
214 RESET ABB_Scalable_IO_0_D07;
215 RESET ABB_Scalable_IO_0_D08;
216 RESET ABB_Scalable_IO_0_D09;
217 RESET ABB_Scalable_IO_0_D010;
218 RESET ABB_Scalable_IO_0_D011;
219 RESET ABB_Scalable_IO_0_D012;
220 RESET ABB_Scalable_IO_0_D013;
221 RESET ABB_Scalable_IO_0_D014;
222 RESET ABB_Scalable_IO_0_D015;
223 RESET ABB_Scalable_IO_0_D016;
224 ENDPROC
225 PROC pickUp_procedure_tensile()
226 MoveL tens_before_pickUp, v100, z0, softGripper;
227 WaitTime 1;
228 currentPos := CRobT(); ! Get current position
229 WHILE ABB_Scalable_IO_0_DI6 = 0 DO !until laser detect correct position, goes down
230 MoveL Offs(currentPos, 0, 0, -5), v10, z0, softGripper; ! Move down in steps of 5 mm
231 currentPos := CRobT(); ! Update current position
232 ENDWHILE
233 WaitTime 1;
234 SET ABB_Scalable_IO_0_D015; !closes the soft gripper
235 MoveL tens_before_pickUp, v100, z0, softGripper;
236 ENDPROC
237 PROC qrCode_procedure_tensile()
238 MoveJ tens_before_qrCode, v100, z0, softGripper;
239 MoveL tens_qrCode, v100, z0, softGripper;
240 WaitTime 7; !wait for qr code detection
241 MoveL tens_before_qrCode, v100, z0, softGripper;
242 ENDPROC
243 PROC measurement_procedure_tensile()
244 MoveJ tensbeforemeasurement, v20, z0, softGripper;
245 MoveL tensaftermeasurement, v5, z0, softGripper; !does the measurement, recod the time
246 MoveJ tenshome, v100, z0, softGripper;
247 ENDPROC
248 PROC testing_procedure_tensile()
249 MoveJ tens_midwaytesting1, v300, z0, softGripper;
250 MoveJ tens_midwaytesting2, v300, z0, softGripper;
251 MoveJ tens_beforeTesting, v300, z0, softGripper; !right above the machine lower gripper
252 MoveL tens_testing, v30, z0, softGripper;
253 SET ABB_Scalable_IO_0_D09; !this is close the lower gripper on the machine
254 WaitTime 3; !closes for 3 seconds
255 RESET ABB_Scalable_IO_0_D09; !the gripper finished the closing procedure
256 WaitTime 2;
257 RESET ABB_Scalable_IO_0_D015; !opens the soft gripper
258 MoveL tens_duringTesting, v50, z0, softGripper;
259 WaitTime 1;
260 SET ABB_Scalable_IO_0_D010; !this is start test on the machine (with the finger)
261 WaitTime 3;
262 RESET ABB_Scalable_IO_0_D010;
263 WaitTime 15; !waits
264 SET ABB_Scalable_IO_0_D013; !close the upper gripper
265 WaitTime 3;
266 RESET ABB_Scalable_IO_0_D013;
267 TPWrite "Test starting...";
268 !execute test
269 ENDPROC
270 PROC disposal_procedure_tensile()
271 WaitDI ABB_Scalable_IO_0_DI9, 1; !test finished and grippers in position
272 MoveL tens_disposal_lower, v50, z0, softGripper;
273 WaitTime 1;
274 currentPos := CRobT(); ! Update current position
275 WHILE ABB_Scalable_IO_0_DI6 = 0 DO !until laser detect correct position
276 MoveL Offs(currentPos, -10, -5, 0), v10, z0, softGripper; ! Move in steps of 5 mm
277 currentPos := CRobT(); ! Update current position
278 ENDWHILE
279 WaitTime 1;
280 SET ABB_Scalable_IO_0_D015; !closes the soft gripper
281 WaitTime 1;
282 SET ABB_Scalable_IO_0_D011; !opens the lower gripper
283 WaitTime 3;
284 RESET ABB_Scalable_IO_0_D011;
285 MoveL tens_beforeTesting, v20, z0, softGripper; !this point is on the side
286 MoveJ tens_midwaydisposal, v100, z0, softGripper;
287 MoveJ tens_disposal_general, v100, z0, softGripper;
288 WaitTime 1;
289 RESET ABB_Scalable_IO_0_D015; !opens the soft gripper
290 WaitTime 1;

```

```

291 MoveL tens_midwaydisposal, v100, z0, softGripper;
292 MoveJ tens_midwaydisposal2, v100, z0, softGripper;
293 MoveL tens_disposal_upper, v50, z0, softGripper;
294 WaitTime 1;
295 currentPos := CRobT(); ! Get current position
296 WHILE ABB_Scalable_IO_0_DI6 = 0 DO !until laser detect correct position
297     MoveL Offs(currentPos, -10, -5, 0), v10, z0, softGripper; ! Move in steps of 5 mm
298     currentPos := CRobT(); ! Update current position
299 ENDWHILE
300 WaitTime 1;
301 SET ABB_Scalable_IO_0_D015; !closes the soft gripper
302 WaitTime 1;
303 SET ABB_Scalable_IO_0_D014; !opens the upper gripper
304 WaitTime 3;
305 RESET ABB_Scalable_IO_0_D014; !finish opening the upper gripper
306 MoveL tens_beforeTesting, v20, z0, softGripper; !this point is on the side
307 MoveL tens_midwaydisposal, v100, z0, softGripper;
308 MoveJ tens_disposal_general, v100, z0, softGripper;
309 WaitTime 1;
310 RESET ABB_Scalable_IO_0_D015; !opens the soft gripper, drop sample
311 WaitTime 1;
312 MoveJ tens_midwaytesting2, v300, z0, softGripper;
313 MoveJ tens_midwaytesting1, v300, z0, softGripper;
314 MoveJ tenshome, v300, z0, tool0;
315 completedcycleCout_tensile:=completedcycleCout_tensile+1; !counts the number of cycles of tensile
316 ENDPROC
317 ENDMODULE

```