



COMILLAS
UNIVERSIDAD PONTIFICIA

ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS
INDUSTRIALES

TRABAJO FIN DE GITI:

**Automation Tool for the calibration
of a bimetallic component of a circuit
breaker**

Autor: MIGUEL GALLEGO VARA

Director: WILLIAM HAUSER

Co-Director: FERNANDO DE CUADRA GARCÍA

Madrid
JULIO de 2019

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESIS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D.

Miguel Gallego Vara

DECLARA ser el titular de los derechos de propiedad intelectual de la obra: Automation Tool for the calibration of a bimetallic component of a circuit breaker, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

- a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.
- b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
- c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
- d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
- e) Asignar por defecto a estos trabajos una licencia Creative Commons.
- f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

- a) Que la Universidad identifique claramente su nombre como autor de la misma
- b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
- c) Solicitar la retirada de la obra del repositorio por causa justificada.
- d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

- a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
- b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
- c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.
- d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción

de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a2..... deJulio..... de ...2019.....

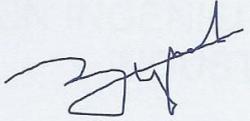
ACEPTA



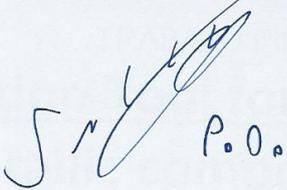
Fdo.....

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el títuloAutomation Tool for the calibration of a bimetallic component of a circuit breaker..... en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico2018-2019..... es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.:  Fecha: 2 / 7 / 19

Autorizada la entrega del proyecto
EL DIRECTOR DEL PROYECTO

Fdo.:  Fecha: 3 / 7 / 19

V.B.




COMILLAS
UNIVERSIDAD PONTIFICIA

ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS
INDUSTRIALES

TRABAJO FIN DE GITI:

**Automation Tool for the calibration of
a bimetallic component of a circuit
breaker**

Autor: MIGUEL GALLEGO VARA

Director: WILLIAM HAUSER

Co-Director: FERNANDO DE CUADRA GARCÍA

Madrid
JULIO de 2019

AUTOMATION TOOL FOR THE CALIBRATION OF A BIMETALLIC COMPONENT OF A CIRCUIT BREAKER

Autor: Gallego Vara, Miguel

Director: Hauser, William

Entidad Colaboradora: Boston University

RESUMEN DEL PROYECTO

El objetivo de este proyecto es el diseño y fabricación de un sistema automático capaz de ajustar el ángulo y la posición de un componente bimetálico. Estos componentes bimetálicos son una de las piezas más importantes de los disyuntores que se utilizan como elemento protector en los circuitos eléctricos. Su función es la de saltar en el caso de que la corriente sea demasiado alta, puesto que podría provocar múltiples daños en el sistema. Este proyecto se realiza para “Sensata Technologies”, una empresa de ingeniería ubicada en Attleboro, MA, proveedor de sistemas de control, protección eléctrica y administración de potencia. Este proyecto se realizó de forma grupal, puesto que así lo requería Boston University.

El proceso de calibración que se realiza actualmente en Sensata Technologies es muy importante para la seguridad y protección de la industria aeroespacial. Se trata de un proceso manual que requiere de la noción previa por parte del profesional. Sin embargo, los trabajadores que realizan este proceso están cerca de la jubilación, de ahí que Sensata Technologies haya solicitado la automatización del proceso. Aunque tenían un proceso totalmente establecido, no se dieron muchas especificaciones en como se debía diseñar y fabricar. Durante los primeros pasos del proyecto, se pensaron en diferentes posibilidades de cómo afrontar el problema, sin embargo, para facilitar el nuevo sistema y el acoplamiento a los trabajadores de Sensata, se decidió optar por un sistema parecido al que ya tenían. Después de esta decisión, se dividió el sistema en diferentes partes: modulo de medición, modulo de carga y modulo de calibración. El diseño de todas las partes se realizó con el programa de CAD SolidWorks y estas piezas se descargaron en GibbsCAM para su futura fabricación en las máquinas de CNC disponibles en la universidad. El proyecto también tenía una gran parte electrónica, en la que se utilizó un microprocesador como cerebro de las operaciones y para comunicarse con una interfaz visual, que se incluyó para facilitar la interacción entre el sistema y el usuario.

El diseño del modulo de carga y medición fue muy similar al que Sensata estaba utilizando: un sistema de dos muelles unido a un deslizador y un elemento fijo que sostenía un medidor digital. Todas estas piezas se fabricaron en las máquinas de CNC con los dibujos proporcionados por el cliente.

La mayor parte del diseño mecánico se produjo en el modulo de calibración. Una pequeña pieza, guiada por un motor, se anclaba en el componente bimetálico y le aplicaba un par de torsión mediante un engranaje. Este mecanismo se utilizaba para curvar el componente bimetálico y poder calibrarlo. El engranaje era muy importante porque incrementaba considerablemente el par de torsión aplicado a la pieza. Uno de los grandes retos del diseño del mecanismo era como tener en cuenta la elasticidad de

la pieza. En el proceso manual, el profesional no tenía que tenerlo en cuenta puesto que no existía un momento de retención. Sin embargo, al utilizar un motor de paso como sustituto, tenía un momento de retención muy grande que no dejaba a la pieza volver a su posición original. Se necesitaron dos iteraciones para fabricar un mecanismo completamente funcional.

Como ya se ha mencionado antes, el proyecto tenía una gran parte electrónica, puesto que tenía que ser un proceso automatizado. Se utilizó un Arduino UNO como microprocesador del sistema e interconectaba el motor, la lectura digital del medidor y una interfaz visual. Uno de los grandes retos fue la lectura del medidor. Después de analizar la salida, el microprocesador fue capaz de interpretar y decodificar la señal, pudiendo entonces saber cual era la posición exacta del componente bimetálico. Después de hacer un análisis de la elasticidad de la pieza a calibrar, se pudo establecer una relación entre la distancia objetivo y la distancia a la que había que curvar la pieza antes de que volviese a su posición estable. Se hizo una regresión y se obtuvo una curva que relacionaba estas dos variables.

La siguiente parte del proyecto fue el diseño de una interfaz visual, que permitiese al usuario interactuar con el sistema automático. Sensata Technologies nos proporcionó una guía de los objetivos o funciones que quería implementar en la interfaz. Se decidió utilizar una pantalla táctil y se programó para que controlase todo el proceso.

Después de que todos los módulos se interconectasen con el microprocesador, se vio que el proceso todavía era algo lento por lo que se realizaron varios experimentos para optimizar el proceso. Finalmente, después de la optimización, el sistema era un 90% eficiente con un tiempo medio de calibración razonable. Sin embargo, es cierto que con algo más de tiempo no se consiguió optimizar el sistema totalmente y más ensayos hubiesen tenido un impacto positivo en los resultados.

El sistema fue entregado al cliente y lo utilizaron como base como diseño de su futuro sistema automático.

Palabras clave: disyuntor, componente bimetálico, elasticidad, calibración, motor de paso, medición.

AUTOMATION TOOL FOR THE CALIBRATION OF A BIMETALLIC COMPONENT OF A CIRCUIT BREAKER

The purpose of this project is to design and manufacture an automated calibration tool capable of adjusting the angle and position tabs on a bimetallic thermal compensator. Bimetallic compensators are key components used in aircraft circuit breakers to trigger the circuit when currents are too high that could damage the different components in the circuitry.

The present process Sensata Technologies is using is manual and highly dependent on the instructor's knowledge, adding to the fact that the current workers are retiring soon. Consequently, they have asked a group of students to create and develop an automated calibration tool.

The calibration of bimetallic components in Sensata Technologies is a very important process for the security and protection of aircraft circuitry. Although they already had a complete process established, they did not give us any important specifications for the design and manufacture of the automated calibration tool. At the beginning stages of this project, we thought of different possible solutions, however, we finally decided to use the same basic design as Sensata was currently exploiting as this would make it easier for the user to get accustomed of the process. After this decision was taken, we divided the tool into different modules: the measuring module, the loading module and the calibration module. The design of all the parts was done with the CAD program SolidWorks and these pieces were downloaded into GibbsCAM to manufacture them in the CNC milling machines available in EPIC (Engineering Product Innovation Center). The other important section of the project was the electronic configuration. We used a microprocessor to control the different components needed and to communicate with an included graphical user interface to make the interaction between the user and the calibration tool easier.

The design of the loading module and measuring were very similar to the one Sensata was using: a two-spring system attached to a slider and a fixture that holds a drop gauge. All of these pieces were manufactured in CNC mills using the drawings provided by the customer.

Most of the new design came from the calibration module. Driven by a motor, a tool grabbed the bimetallic compensator and applied torque through a set of gears to bend it. These set of gears were very important as they significantly increased the torque applied by the motor. One of the main challenges of the design of this process was to account for the spring back of the bimetallic piece. In the current process, the user did not have to account for this because there was not any holding torque preventing the piece to spring back. However, as we decided to use a motor to substitute the user's hand, it had a big holding torque that would stop the bimetallic component to return to its original position. Two iterations were needed to design a complete functional mechanism that would account on this problem.

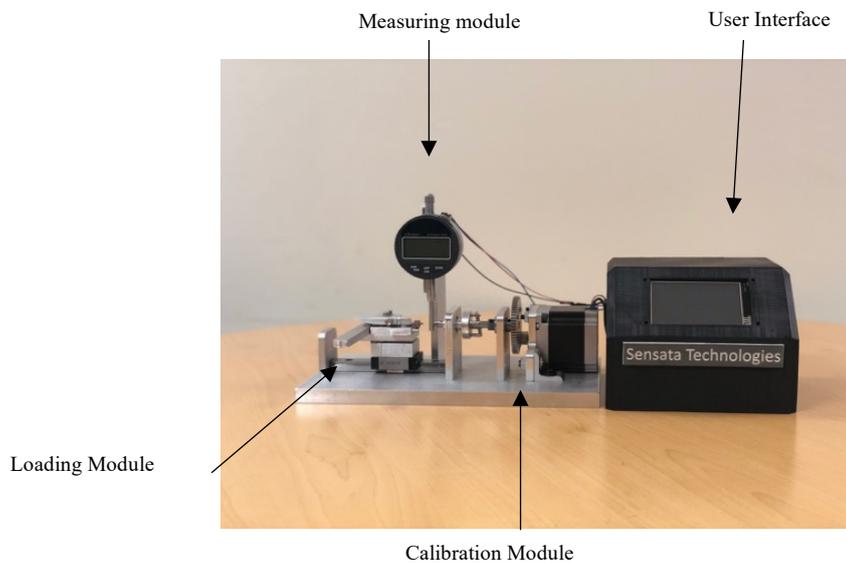
Finally, all of these modules had to be interconnected between them to ensure an automated process. Using a microprocessor as the brain of the whole system was the solution we opted for. There were two main challenges in the electronic configuration: reading the output of the digital drop gauge and completing the calibration tool with an

easy and user-friendly interface. After some analysis of the type of output signal the drop gauge sent, we finally were able to decode it and the microprocessor was finally capable of reading the numbers shown in the drop gauge's screen. The next step of the project was to know how much we did have bend the bimetallic component in order to reach the objective distance the customer needed. However, reaching the objective distance was not the solution due to the mentioned spring back, the bimetallic compensator had to be overbent. After some testing we obtained a relationship between how much the piece was bent and the final distance after the spring back. We did a regression and obtained an equation that related these two variables and incorporated it in the microprocessor.

The last part of the project consisted on the design of the graphical user interface. The specifications Sensata gave us were to design an easy to read interface that enabled the user to interact with the calibration tool. We decided to use a touchscreen and programmed it to control the process.

After all of the pieces were interconnected between them, we tested the results of the process and saw possible solutions. At the beginning the process was slow but after some days of optimization we were to decrease the time significantly. Due to lack of time we were not able to completely optimize the tool and more testing would uncover necessary design changes and new updates.

The final prototype was delivered to the customer and they would use it as a base for their future designs and automated processes.



Keywords: circuit breaker, bimetallic component, calibration, stepper motor, measuring module, spring back.

Table of Contents

List of Figures	11
List of Tables.....	12
Project Description: Introduction and Background	13
Customer: Sensata Technologies	13
Circuit Breakers	13
Bimetallic Component	14
Calibration Process	15
Project Requirements.....	19
Engineering Specifications and Relevant Basic Physics	20
Design Decisions	21
Design Description	23
Loading Module	23
Measuring Module	24
Calibration Module.....	27
Driving Function.....	27
Bending Function	31
Final Prototype.....	32
Graphical User Interface	33
Design Evaluations	37
Spring Back Factor	37
Results	39
Cost Analysis.....	42
Future Improvements	43
Conclusion and Recommendations.....	44
References.....	45
Appendix	46
Working Drawings	46
Electronics.....	52
Arduino Code	54

List of Figures

Figure 1: Circuit Breaker A	14
Figure 2: Circuit Breaker B	14
Figure 3: Bimetallic Compensator in Circuit Breaker	14
Figure 4: Size Comparison of Bimetallic Compensator	14
Figure 5: Bottom part of Bimetallic Compensator	15
Figure 6: Bimetallic compensator side view	16
Figure 7: Mitutoyo Digital Dial Indicator	16
Figure 8: Sensata Technologies 173621 calibration fixture (pre heat treatment). CAD	17
Figure 9: Sensata Technologies 173621 calibration fixture (pre heat treatment).....	18
Figure 10: Final Prototype Modules	23
Figure 11: Loading Module Components.....	24
Figure 12: U-tool attached to Bimetallic Compensator	24
Figure 13: Measuring Module	24
Figure 14: Output of DDI seen in oscilloscope	25
Figure 15: Level Shifter Circuitry	26
Figure 16: Calibration Module Components	27
Figure 17: Bipolar Stepper Motor	28
Figure 18: Full Step Drive	29
Figure 19: Microstepping	29
Figure 20: Gear ratio analysis.....	30
Figure 21: Driving Module.....	30
Figure 22: Final Prototype Design.....	32
Figure 23: Final Prototype Process.....	32
Figure 24: User Interface Steps	34
Figure 25: GUI-Add New Gap Set Option.....	35
Figure 26: GUI - Motor Movement Capability	35
Figure 27: Nextion Editor.....	36
Figure 28: Springback Factor Analysis	37
Figure 29: Graph - Springback Factor	38
Figure 30: Graph - Final Value VS Reach Value	38
Figure 31: Graph - Results of Calibration Process	40
Figure 32: Level Shifter Circuitry	52
Figure 33: Stepper Motor Circuitry	53
Figure 34: GUI Circuitry	54

List of Tables

Table 1: Bimetallic Compensators Provided	18
Table 2: Engineering Specifications	20
Table 3: Design Decisions	22
Table 4: Binary Numbers Analysis (DDI).....	26
Table 5: Cost Analysis.....	42

Project Description: Introduction and Background

Customer: Sensata Technologies

The project has been proposed by Sensata Technologies, a large engineering company settled in Attleboro, MA, focused on control systems for cleaner and more efficient products. Although it has a long history, Sensata Technologies was reborn as a private company in 2006. Nowadays, it is one of the “world’s leading suppliers of sensing, electrical protection, control and power management”.¹ The aerospace industry is one of the biggest clients of this company, where it provides safety and affordable devices. Consequently, as a protection supplier for this industry, Sensata has asked Boston University (BU) to find a group of students that can solve a problem they currently have.

Problem Statement: *Design and manufacture an automated calibration tool that will be capable of adjusting the curvature of a bimetallic compensator of a circuit breaker.*

Circuit Breakers

In the engineering world, there are an infinite number of protection devices, and when we talk of electric circuits, the most common one is the fuse. A fuse is a “switch” composed by a thin wire that is part of the circuit but disintegrates when the current is too high, due to the extreme temperatures.² However, as the wire disintegrates, the fuse can only be used once and it has to be replaced each time.

Consequently, a new device that could be used over and over again was needed, and the circuit breaker was developed. These devices are used all around the modern world. The National Electrical Code (NEMA) define the circuit breaker as: “A device designed to open and close a circuit by non-automatic means and to open the circuit automatically on a predetermined overcurrent without damage to itself when properly applied within its rating”.

Sensata’s aircraft circuit breakers are widely used on military and commercial aircrafts to keep the safety of the passengers and pilots on board. The aircrafts often need to endure through extreme temperature conditions during flight, which poses high safety requirements on the circuit design to prevent current overflowing. The critical component of protection would be the circuit breaker.

The inside of the circuit breaker is mostly mechanical and is composed of very different pieces. The circuit breakers Sensata Technologies work for are illustrated in *Figures 1 and 2*.

¹ <https://www.sensata.com/about>

² <https://electronics.howstuffworks.com/circuit-breaker2.htm>



Figure 1: Circuit Breaker A



Figure 2: Circuit Breaker B

Bimetallic Component

Inside the circuit breaker there is a bimetallic component used as the trigger when large currents pass through it. As the name indicates, this part is composed of two metals with two different thermal coefficients. Due to this difference, when high currents go through them, one metal stretches more than the other one, and as they are attached, the component will bend. This movement is what causes the trigger to jump and the circuit to break.



Figure 3: Bimetallic Compensator in Circuit Breaker

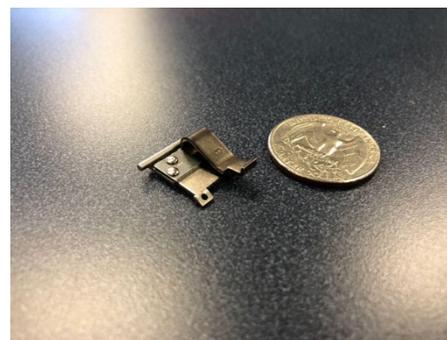


Figure 4: Size Comparison of Bimetallic Compensator

In industry, each bimetallic component is different, and it is the most important piece of the circuit breaker. Sensata Technologies is one of the few international companies that work with these pieces and calibrate them. Although Sensata Technologies work with hundreds of different bimetallic components, the project focused on the five most used pieces. These components are shown in *Figures 3 and 4*. A SolidWorks CAD representation is also illustrated in *Figure 5*.

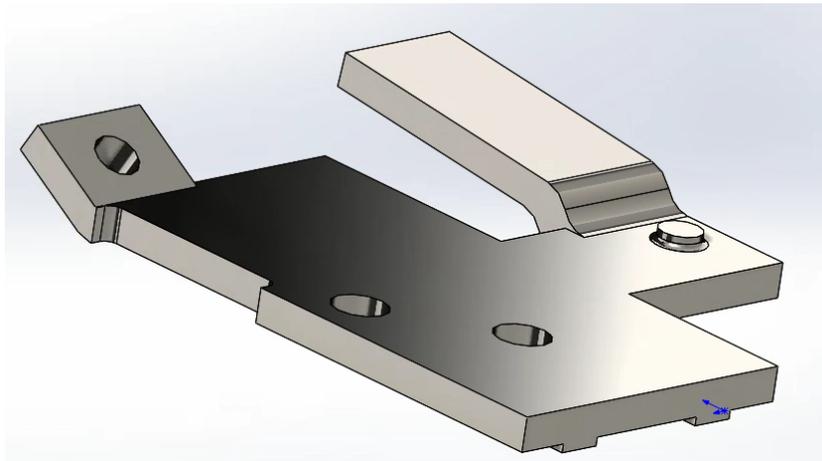


Figure 5: Bottom part of Bimetallic Compensator

It is important to highlight some of the properties of this compensator. The most important one is its high spring back factor. In order to bend a certain amount of the component, much more force and bending motion was needed. This was an issue throughout the project as it made much more difficult its consecution. Furthermore, they are composed of Nickel, Copper along with small amounts of other alloys.

Calibration Process

These bimetallic components are small and need to be precisely manufactured. However, they do not come perfectly calibrated, therefore, Sensata Technologies has this job to do. Currently, the calibration process is manually done, which is highly dependent on the operator's knowledge and experience. There are two or three operators that have been working in the calibration process for over 20 years and since some of them are retiring soon, Sensata urgently needs a replacement process. This is the reason Sensata has asked for help and desires a tool capable of calibrating the piece automatically. This automated calibration project will not only increase Sensata's capabilities to produce compensators correctly and efficiently, but also improve the company's lean manufacturing capabilities.

Two critical dimensions are shown in *Figure 6* below. The **gap set** is the distance between two specific features of the thermal compensator. The gap set controls how close the compensator is to the circuit breaker trip arm (not shown). The second critical dimension, the **stop gap**, controls how much the compensator will shift with temperature variation for thermally-compensated devices. On non-compensated devices, the stop gap is zero. A specific **gap set** and **stop gap** distance is specified for each circuit breaker product.

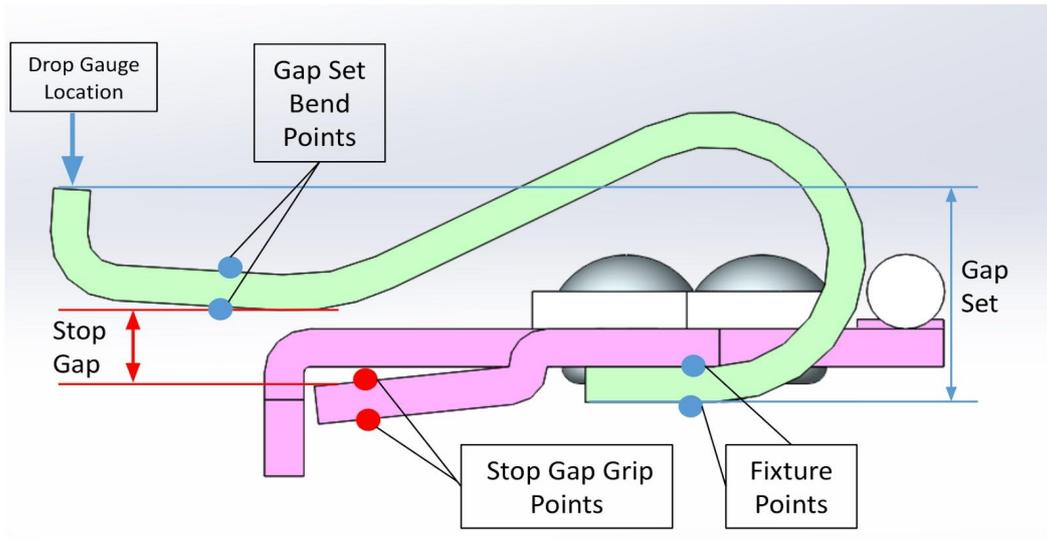


Figure 6: Bimetallic compensator side view

To measure the Gap Set, Sensata Technologies used a Digital Dial Indicator (DDI). This apparatus is basically a drop gauge used to measure very small distances. The one Sensata Technologies used was a "Mitutoyo" with a precision of 0.01 mm/0.0005 in.



Figure 7: Mitutoyo Digital Dial Indicator

Currently, the compensator is calibrated by skilled operators in four steps:

1. They place the bimetallic component into a fixture and the gap set is measured by a digital dial indicator. The person in charge would start applying torque with a U-tool attached to the bimetallic component until the dial indicator reached the objective value. To check if the DDI reading was accurate, the person remeasured the component several times. If all of the distances were within tolerance, the gap set was adjusted.
2. The compensator is heat treated in Sensata's lab, where they put the parts in an oven at several hundred degrees.
3. The stop gap is set using a similar U-Shaped tool for bending a tab on the compensator. This step is done without a fixture and the gap is checked using go/no-go pin gauges.
4. The gap set is readjusted by bending the bottom portion of the compensator in a setup similar to the first step, until the height is within a specified tolerance range.

As an indication, Sensata showed the way they calibrated the tool manually. A CAD of the complete tool is showed in Figure 7.

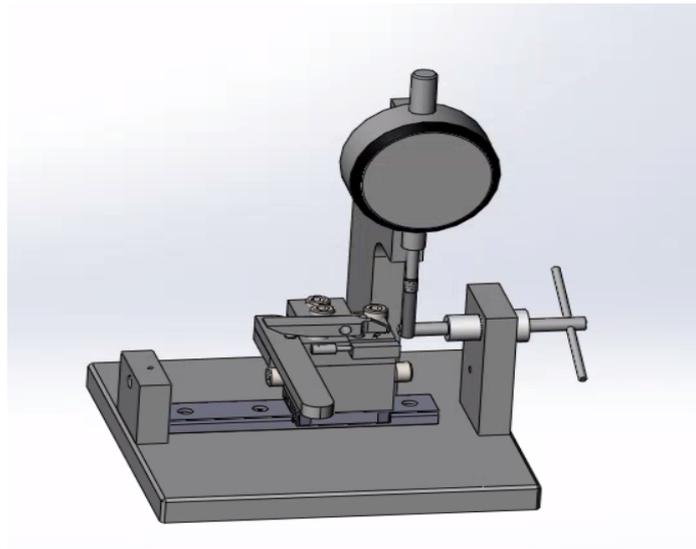


Figure 8: Sensata Technologies 173621 calibration fixture (pre heat treatment). CAD



Figure 9: Sensata Technologies 173621 calibration fixture (pre heat treatment).

Figure 7 shows the calibration fixture used for the adjustment of the gap set. This fixture consists of two modules: the loading module and the measuring module. The operator would fix the bimetallic component in the loading module that consisted of two spring systems and then by pushing it with the slider, the bimetallic component would attach to the U-tool. As mentioned before, the digital dial indicator would measure the gap set and the operator would overbend the piece until the desired distance is achieved.

Although Sensata works with more than 30 different pieces, we are only going to use the different components that are represented in the next table:

Table 1: Bimetallic Compensators Provided

Series	311-50	311-50	311-50	100-50	12-50
	A	B	C	S	S
Number ID	12652-4	12658-4	12670-4	12028-8	12028-5
Gap Set (in)	0.155	0.155	0.155	0.140	0.135
Number In Piece	1	1	1	4	8

Project Requirements

After the initial customer interview with Sensata, our team categorized the objectives and requirements of this project into three levels. Mandatory is the most important and basic requirement that our team needs to achieve in order to complete the project, desirable is the optional, advanced requirement that will further improve the efficiency of the calibration system, and stretch is the hardest, most advanced goals that is generally expected if our team is given more time and budget.

Mandatory

- Prototype is able to perform at least one of the two specified calibration processes in under 15 seconds within the specified tolerance.

Desirable

- Creation of a functional graphical user interface with real-time feedback to operate and set parameters given calibration requirements.
- Perform both calibration operations (pre and post heat treatment).

Stretch

- Create a fully automated system with calibration and adjustment capability. The operator only has to feed in uncalibrated parts and remove finished parts after the calibration.

Engineering Specifications and Relevant Basic Physics

Table 2: Engineering Specifications

ENGINEERING SPECIFICATIONS		
<i>Specification</i>	<i>Design Methods</i>	<i>Units and Dimensions</i>
Estimated Gap Set Size	(From Sensata 6752 ACCB model # 311-50, #100-50)	~ 0.155, 0.140, 0.135 in.
Gap Set Tolerance	From Sensata's Requirements	±0.002 in.
Estimated Prototype Size	Calculated from Device dimension	10.0 x 6.0 x 8.0 in. (Main body) 6.0 x 5.5 x 5.5 in. (Interface)
Motor	Nema 23 CNC Stepper Motor	2.24" x 2.24"x 2.2" 4-wire 200 steps/rev Rated Current: 2.8 A Resistance: 0.9 Ohms
Motor Torque	Nema 23 CNC Stepper Motor	1.26 Nm
Microcontroller	Arduino UNO	

Design Decisions

The calibration tool had different critical functions and it was very important to decide what was the best option for each of one. The five main functions were:

1. Driving Motion: This function would be the substitute of the user's hand. We decided to use a bipolar stepper motor because it has high torque and it's easy to be controlled by a microprocessor. Furthermore, as the gap sets of the different components were very small and precise, a stepper motor would ensure minuscule movements.
2. Bending Motion: For the bending mechanism, it was decided to use the same U-tool as Sensata Technologies was already using because we already had the design and it was relatively easy to manufacture
3. Measurement: For the measuring function we had several different possibilities.
 - a. The first one was to use a digital dial indicator, extracting the information into a micro processing unit and using the step motor with some gears to provide the necessary torque. As mentioned before the digital dial indicator had a precision of half of a thousand of an inch (+/- 0.0005 in).
 - b. The second option was to use a laser system. A laser beam would be directed into the bimetallic component. A fixture connecting both the laser emitter and a receiver would move up and down using a stepper motor. When the receiver stopped detecting the beam, it meant the laser was being obstructed by the component and by counting the number of steps turned by the motor, we would be able to calculate the distance.
 - c. The third idea was to use some type of imaging technique, like photography or shadow imaging to capture the distances to be set and readjust them with a stepper motor.

Using photographic imaging as well as shadow imaging to measure distances or even using laser movement were too complex and not as precise as using a Digital Dial Indicator. Consequently, we finally decided to go for the drop gauge.

4. Graphical User Interface: We wanted an easy and user-friendly interface for Sensatas workers, therefore, we rapidly discarded using a laptop as it would make the process more tedious. Although it was the most expensive one, we decided to use a touchscreen as using an LCD display would limit the possibilities.
5. Microprocessor: Due to our familiarity with the Arduino world we finally went for the Arduino UNO processor.

Table 3: Design Decisions

Function	Concepts			
Driving	Motor	Actuator	Robot arm	
Bending	Mechanical Claw	U-tool		
Measuring	Laser	Digital Dial Indicator	Shadow Imaging	Camera
GUI	No GUI	LCD Display	HMI Touchscreen	MATLAB GUI
Processor	Raspberry PI	Arduino UNO	Laptop	

Design Description

The final prototype consisted of four modules that together completed the calibration of the bimetallic components: the loading Module, the measurement Module, the calibration Module and the user interface. The loading, measurement and calibration work together to physically bend the component, while the interface allows the user to choose the calibration settings as well as adding new settings.

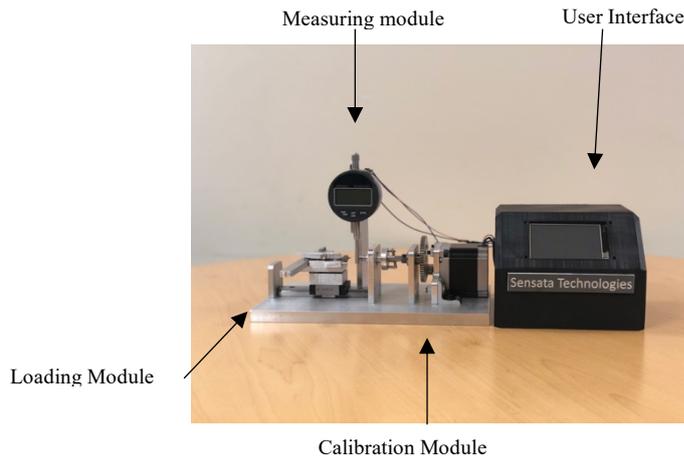


Figure 10: Final Prototype Modules

There are going to be several steps for the user to follow:

1. Load the bimetallic compensator in the loading module
2. Push the loading module through the slider into the calibration module
3. Select the component in the graphical user interface and start the process
4. Stepper motor will automatically adjust the bimetallic piece
5. The results are going to be shown in the user interface

Loading Module

The Loading Module allows the operator to easily and securely fixate the bimetallic components for the bending, and zero sets the digital dial indicator (DDI) in the measurement module to ensure the accuracy of the reading. The Loading Module mainly consists of the lever mechanism for fixation and a linear guide for sliding movement during loading and unloading.

Since the current fixture used at Sensata is proven to be reliable and effective, we adopted the loading module design into our final prototype to ensure the reliability and decrease the learning curve of operators who will use our device. The lever and the clamp are spring loaded. When loading the bimetallic component, the user pulls the lever to the left to open the clamps and places the component until it contacts the guiding pins. Then the user pulls the lever to the right until the clamp bites into the component tightly.

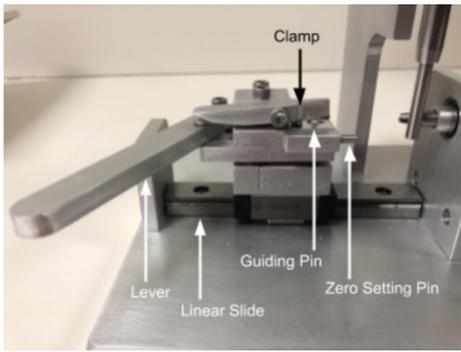


Figure 11: Loading Module Components



Figure 12: U-tool attached to Bimetallic Compensator

The zero-set function is achieved by the stainless-steel pin that is located on the side of the loading component. This function is very important as it zero-set the Digital Dial Indicator, so it measures the correct distance. The top surface of the pin is coincident with the loading surface. When the user turns on the DDI, he/she will lower the measuring tip until it hits the pin, and zero set the DDI so that it measures the gat set accurately.

The Linear guide allows the sliding movement of the loading module, which makes the loading and unloading process easier for the user. The Aluminum walls at the both ends of the slide set the boundaries of the linear movement and prevent the loading module from sliding out.

Measuring Module

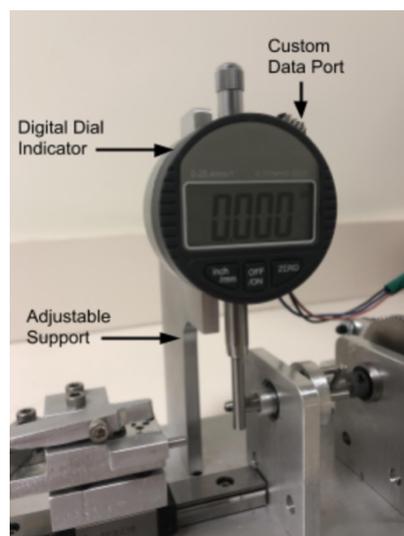


Figure 13: Measuring Module

For the measuring module, a digital dial indicator was used which basically enables the user to measure very small differences. It is used in infinite different engineering applications. As Sensata Technologies measured the gap set with this type of instrument we decided to continue with it as it had a very high precision, of half a thousand of an inch.

To automate the process, we had to take the measurement from the digital dial indicator electronically. Consequently, we had to decode the digital dial indicator to send this information to the microcontroller.

Firstly, the Digital Dial Indicator (DDI) had four output pins.

1. 1.5 Volts
2. Clock Pin: this pin is used to determine the data transfer rate
3. Data Pin: this pin is used to determine the number to read
4. Ground (GND)

In order to access the data, we 3D printed a DIY port and connected some cables to the measurement module.

The output of the DDI was a “RS232” (Recommended Standard 232) type. These signals are represented by voltage levels with respect to a common ground.

In the case of the digital dial indicators as well as digital dial calipers, the different numbers that appeared on screen were transferred in series of multiple voltage pulses. The different voltage pulses represented a bit, HIGH or LOW, and the combination of these bits would finally determine a binary number.

An example of a number would be:

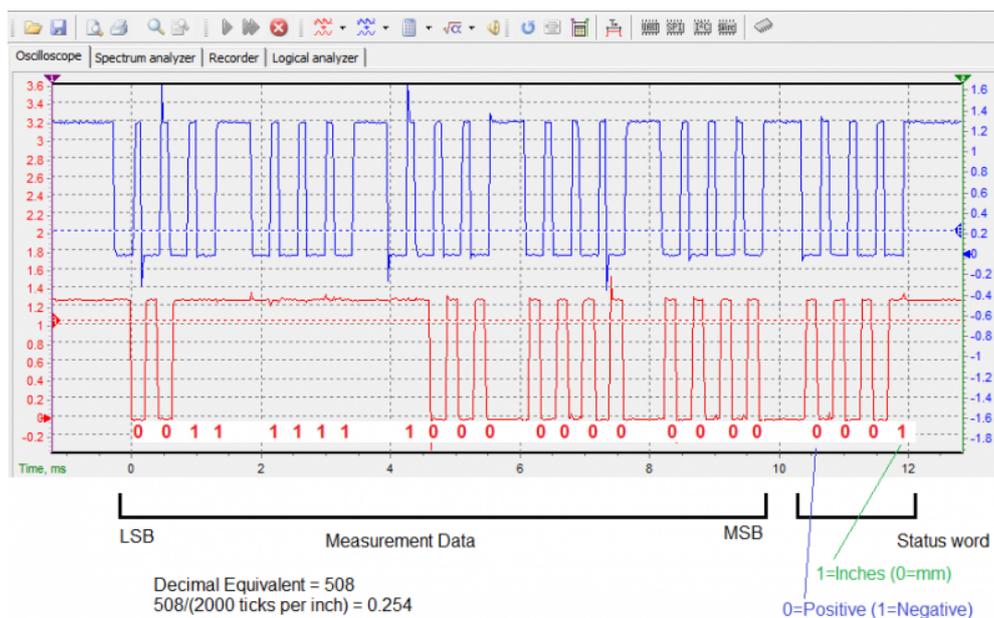


Figure 14: Output of DDI seen in oscilloscope, <https://hackaday.io/project/511-digital-dial-indicator-cnc-surface-probe/log/814-the-digital-dial-indicator-and-how-to-read-from-it>

In the figure above we can see the clock pin and the data pin are going from 1.5volts to 0 volts, or common ground. We had to determine which were the significant bits and if any of the bits determined the sign and the units of the number.

Table 4: Binary Numbers Analysis (DDI)

Reading	Unit	Sign	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Binary Number	Number	Divisor
10,67	mm	Negative	1	1	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1067	10,67	100
0,42	in	Negative	0	0	0	1	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	1	0	0	1	840	0,42	2000
27,9	mm	Positive	0	1	1	0	0	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	2790	27,9	100
1,1005	in	Negative	1	0	0	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	2202	1,101	2000
0,7345	in	Negative	1	0	1	1	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	1469	0,7345	2000
0,3455	in	Negative	1	1	0	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	691	0,3455	2000
9,74	mm	Positive	0	1	1	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	974	9,74	100

After analyzing the data, we reached several conclusions:

1. Each package was composed of 24 voltage pulses that represented 24 bits
2. We had to read the data pin when the clock pin was rising (from LOW to HIGH)
3. Bit number 21 determined the sign (negative if HIGH and positive when LOW)
4. Bit number 24 determined the unit (inches when HIGH and mm when LOW)
5. Only bits from 1 to 16 were significant

Another important aspect we had to take into account was the fact that the Digital Dial Indicator worked with 1.5 volts, and Arduino works with 5 volts. Therefore, if we directly hooked it up with the microprocessor, it would not recognize the 1.5 volts as HIGH. Consequently, we had to create a circuit that would level shift the signal.

The next circuit composed of two BJT transistors was used.

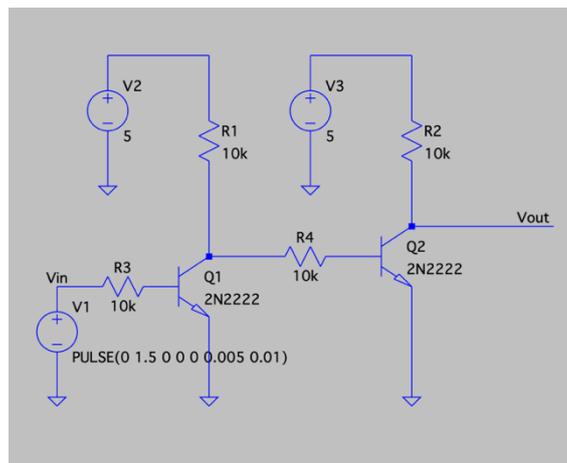


Figure 15: Level Shifter Circuitry (LTspice)

This circuit can be seen as two inverters put in series. When the pulse is HIGH or 1.5 Volts the output will be 5 V and when the input is LOW or 0 V, the output would be close to zero.

Using this method, the Arduino UNO was able to read the numbers applying the code described in the Appendix.

Calibration Module

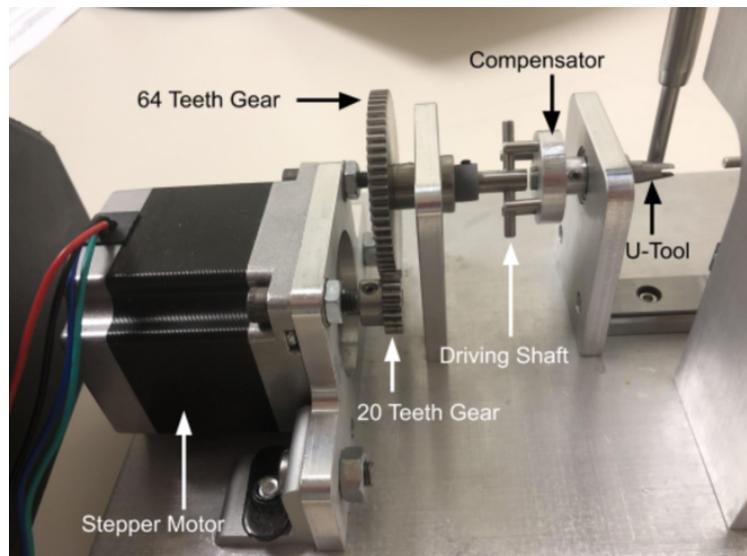


Figure 16: Calibration Module Components

The main function of the calibration module is to execute the command given by the Arduino, and physically calibrate the gap set of the bimetallic components by bending. The calibration module consisted on two different parts: the driving motion and the bending motion. Both of them had to be combined and assembled together to achieve the maximum amount of torque in order to bend the bimetallic compensator. The driving module consisted on a motor and a set of gears and the bending function consisted on the compensator, a mechanism that accounted for the spring back factor of the bimetallic compensator, and the tool that would grab the piece. For this module we did two different iterations/prototypes.

Driving Function

The driving module in the calibration tool had to be stable and provide high torque to bend the bimetallic compensator. To control the driving module with a microprocessor we decided to use a motor, in particular, a stepper motor.

Stepper motors are widely used in the engineering industry, from CNC machines to robotic arms. They have the ability to make precise movements and can be easily electronically controlled. Stepper motors work with direct current but have some differences with DC motors. As every motor, a stepper motor consists on a stator and a rotor, however, they do not have brushes which revert the electrical current and keep the rotor turning in the same direction.³ Consequently, they are known as “brushless

³ <https://www.explainthatstuff.com/how-stepper-motors-work.html>

motors”. Stepper motors consist of a rotor that generally is a magnet and a stator that consists of different coils as we can see in *Figure 18*.

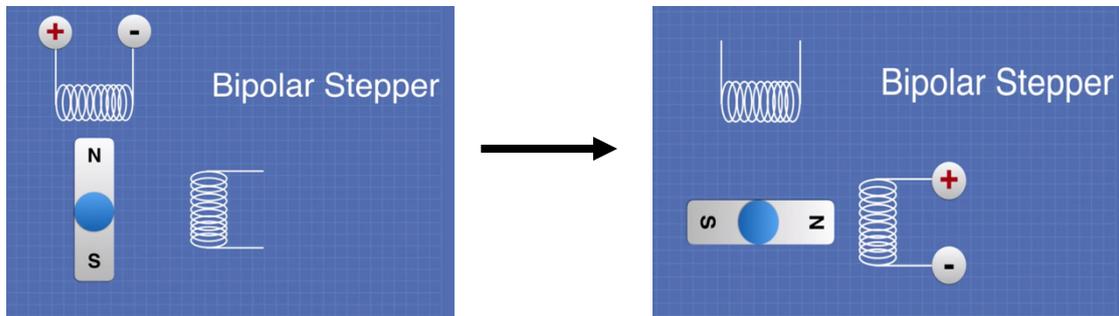


Figure 17: Bipolar Stepper Motor, <https://www.youtube.com/watch?v=0qwrnUeSpYQ>

and therefore moving the center shaft. There are two main types of stepper motors: unipolar and bipolar.

- Unipolar: the windings of this type of stepper motor have a center tap which is always attached to the positive voltage. These motors are used when less torque is needed but higher speeds.
- Bipolar: They only have one winding per phase as shown in *Figure 17*. They are used when higher torque is needed but lower speeds.

The main specifications of stepper motors are:

- Step angle: how many degrees does the shaft advance for each step
- Voltage rating: maximum voltage the coils can tolerate
- Current rating: current achieved to meet specifications ($I = V/R$)
- Holding Torque: it is the amount of torque created when the motor is energized.

When the stepper motor is controlled by a microprocessor, there has to be a driver that connects both of them. This is due to the fact that the motors can't be connected directly to the microprocessor due to their high currents, that could possibly harm the Arduino. Furthermore, stepper motor drivers are devices that control how the stepper motor operates. They basically send the current through the different phases/coils of the stepper motor to make the rotor advance. They are very important devices in electronics and make the control of stepper motors much easier.

There are different possible ways to drive the stepper motor. The most basic one would be to activate one coil at a time, which doesn't use the full possible torque. To increase this torque, it is possible to activate two coils at a time, providing much higher torque. This method is known as "Full Step Drive".

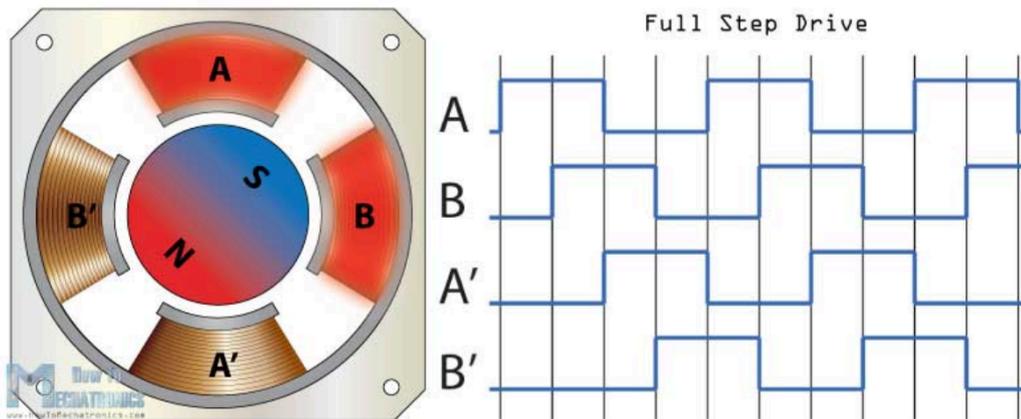


Figure 18: Full Step Drive, <https://howtomechatronics.com/how-it-works/electrical-engineering/stepper-motor/>

However, it is very common nowadays to use micro stepping. This method is achieved by providing the current of the coils in the form of a sine wave as the next figure illustrates. With this method, you are able to achieve much more steps per revolution and therefore, increase the precision.

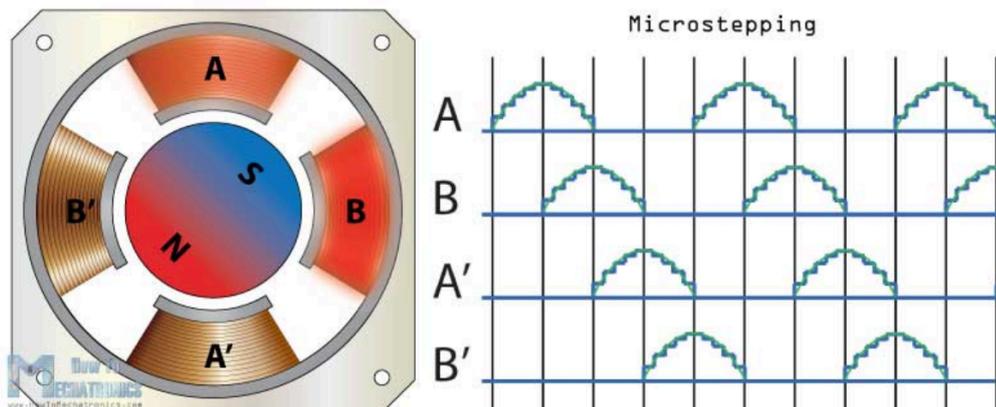


Figure 19: Microstepping, <https://howtomechatronics.com/how-it-works/electrical-engineering/stepper-motor/>

In this project, the bimetallic compensator had to be overbent in order to change the distances. Consequently, high torque was needed. A NEMA 23 stepper motor with 1.26 Nm of holding torque was going to be used. To increase the amount of torque reaching the bimetallic compensator, a set of gears were used. The amount of torque increased is directly related to the pitch diameter and number of teeth the gears have:

$$\text{Stepper Motor: } 200 \frac{\text{steps}}{\text{revolution}}, \text{Torque} = 1.26 \text{ Nm}$$

$$\text{Gear 1 : Pitch Diameter} = 2 \text{ in, Number of Teeth} = 64$$

$$\text{Gear 2 : Pitch} = 0.69 \text{ in, Number of Teeth} = 20$$

$$\text{Gear Ratio} = \frac{64}{20} = 3.2$$

→ For 1 turn of the Big Gear, the small one will turn 3.2 times

$$\text{The big gear will be able to make: } 200 * 3.2 = 640 \frac{\text{steps}}{\text{revolution}} = 0.5625 \frac{\circ}{\text{step}}$$

The torque applied to the bimetallic component would be:

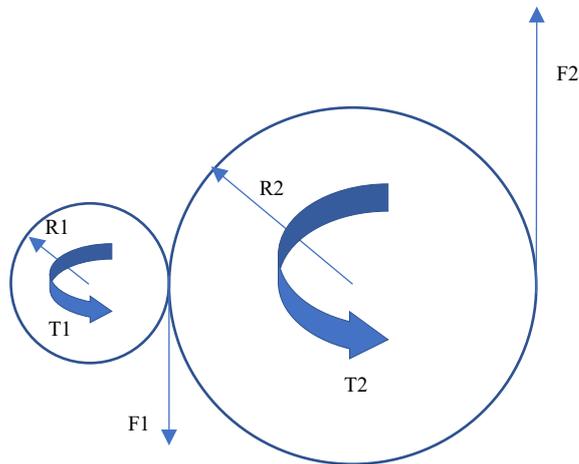


Figure 20: Gear ratio analysis

$$T1 = 1.26 \text{ Nm}$$

$$F1 = \frac{T1}{R1} = \frac{T1}{\frac{PD}{2}} = F2$$

$$T2 = F1 * R2 = F1 * \frac{PD2}{2}$$

$$T2 = T1 * \frac{R2}{R1} = T1 * \frac{64}{20} = 4.032 \text{ Nm}$$

We chose the gears to be made from stainless steel, as they were an important feature of the fixture, being extremely necessary their precision and reliability.

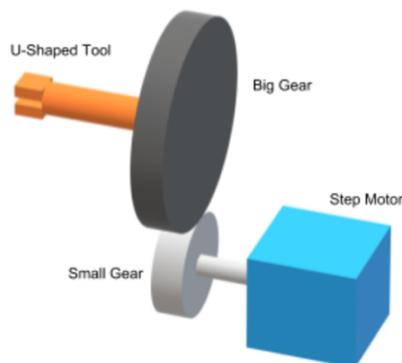


Figure 21: Driving Module

Bending Function

In the current calibration process Sensata is using, the u-tool shown in *Figure 12* is attached into the bimetallic compensator and it applies the necessary torque to overbend it. We decided to use the same component. However, we had to take into account a very important issue when automating the process: the spring back factor of the bimetallic compensator.

The spring back happens when a material tries to return to its original position after being bent. In the current manual process, the user did not have to worry about this factor because there wasn't any holding torque after it was being applied. However, as we were going to use a stepper motor with a very high holding torque, we had to create a mechanism that enabled the component to spring back.

The U-tool mechanism applies the torque from the motor directly to the bimetallic components, while compensating the spring back of the components. The mechanism has several components; the driving shaft, the compensator, the U-tool and Bearing shaft supports.

The driving shaft is a T-shaped shaft directly connected to the 64 teeth metal gear. It transfers the torque from the motor to the mechanism. The two pins serve as the arms of the driving shaft and turns the compensator when the motor turns. The compensator is mainly consisted of four steel pins on a blank gear, forming a two by two array with a 0.25-inch gap (compensation gap). The U-tool is directly connected the compensator.

At home position, the arms of driving shaft are positioned at the middle of the compensation gap, so that the compensator can be turned freely within the gap distance, to be more tolerant for the different initial gap set of the uncalibrated components. When the motor turns, the driving shaft rotates clockwise or counter-clockwise until it contacts one of the pins on the blank gear, which then will transfer the torque to the U-tool and finally to the bimetallic component for the bending.

After "overbending" the bimetallic component, the motor will turn back the driving shaft to its initial position; the driving shaft and the compensator will release during the process. Due to the springback, the bimetallic component will partially revert towards its original position and the compensator gap allows its to turn freely without restraint. Finally, the Digital dial indicator can recheck the post-calibration gap set measurement and decide whether the component needs to be calibrated again.

Final Prototype

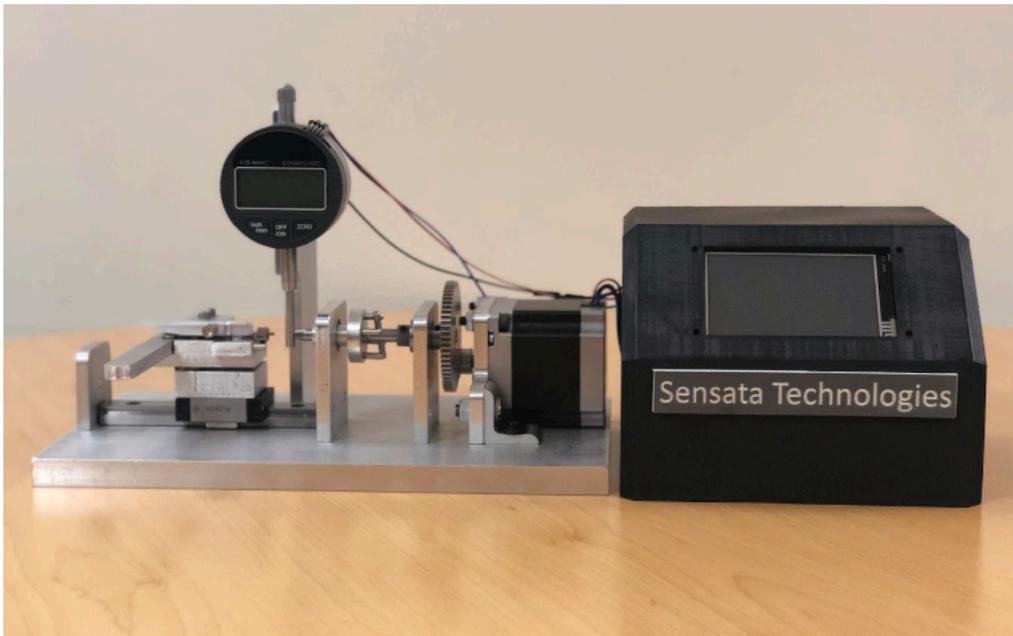


Figure 22: Final Prototype Design

After completing the first prototype and analyzing all the possible improvements we designed and manufactured our second and final prototype. This prototype combines both mechanical and electronic components to ensure a stable and efficient process. We decided to combine both the motor and measuring module together. Furthermore, we changed the loading module to the one Sensata Technologies was using as it enabled a firm grip and ease to use. The steps this calibration tool follows to bend the bimetallic component are represented in the next figure.

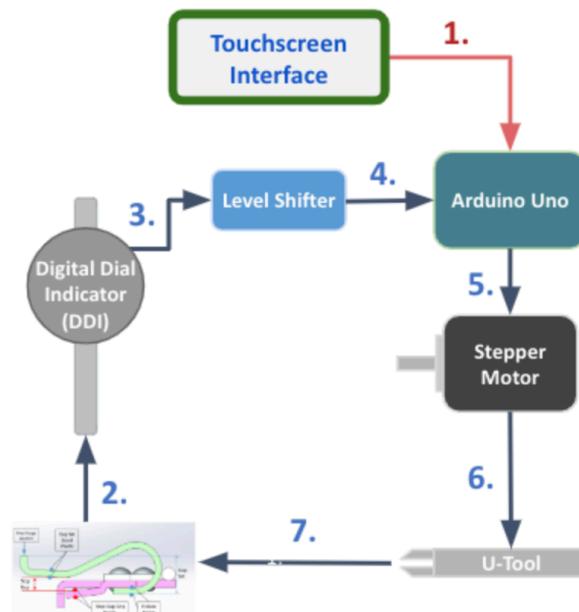


Figure 23: Final Prototype Process

1. The user selects between the different components stored in the memory of the microprocessor and starts the process.
2. The digital dial indicator (DDI) reads the current gap set.
3. The drop gauge outputs the signal through the DIY port.
4. The output is sent through a level shifter to the Arduino, that reads the number and stores it.
5. Through several operations, Arduino decides how much does the motor bend in order to reach the Objective Gap Set value
6. The Stepper motor drives the U-tool through a set of gears that increase the overall applied torque
7. The U-tool grabs the bimetallic component and bends it
8. The loop starts again with the reading of the digital dial indicator (2)

Graphical User Interface

The last part of the calibration tool was the design of a graphical user interface (GUI). This was a very important part of the project because it enabled the customer to use and control the tool easily. As mentioned before, we discussed different possibilities such as an LCD display, using a laptop or a touchscreen. We finally decided to use a 3.2" touchscreen because it was the most user-friendly device and the customer seemed to agree with it.

In the first place, we questioned ourselves what the possible functions for the interface were. After placing the bimetallic compensator in the loading module and attaching it to the U-tool, the user would select in the interface what was the piece number. Consequently, the user interface should have some type of number keypad that enabled the user to choose the number ID. After selecting the component, some information should be displayed, like the Objective Gap Set or the Current Gap Set and after this, the he/she should be able to automatically start the process. We also introduced a stop button in case there was any problem with the calibration process.

A screenshot of the process of the calibration tool:

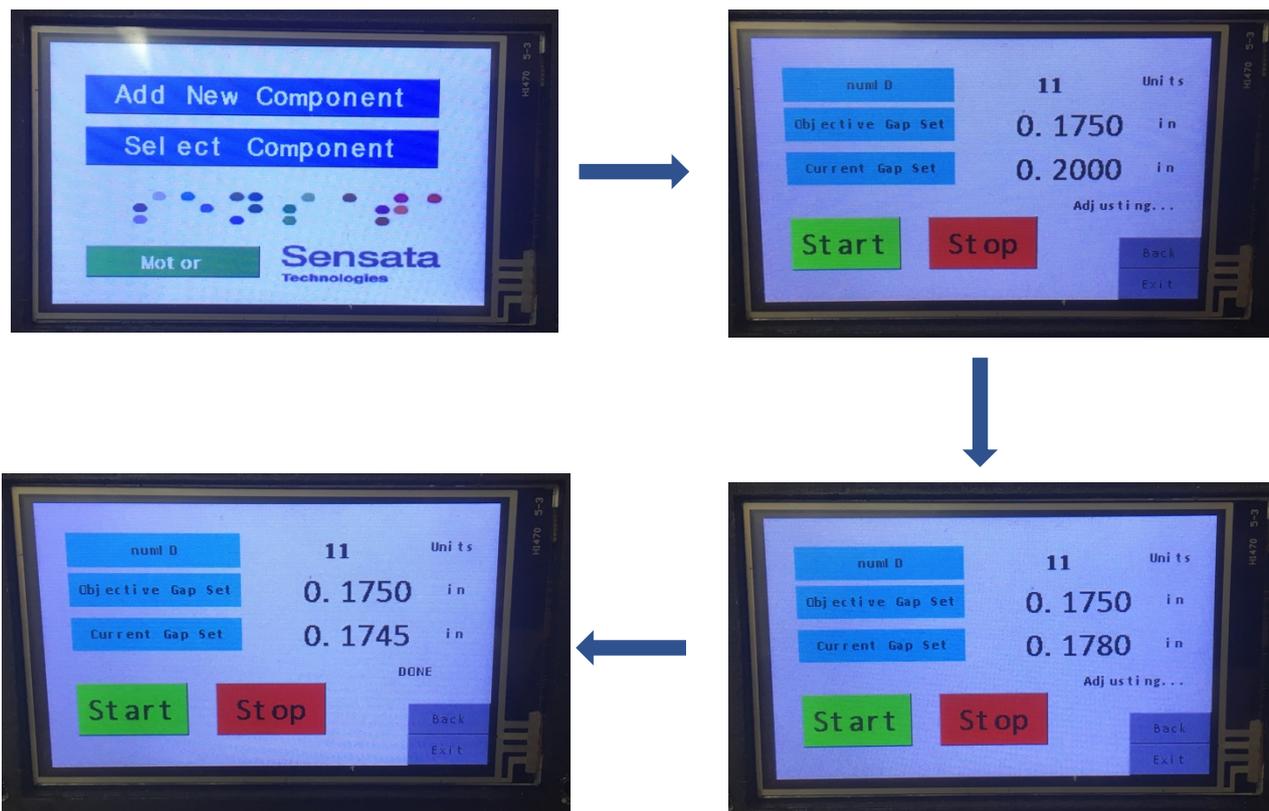


Figure 24: User Interface Steps

The next function to be introduced in the user interface was the possibility of adding new components to the memory of the microprocessor. To do this, we would ask the user what the new number ID and new objective gap set distance for the piece was. This distance was then stored in the memory of the Arduino. To do this we had to include the EEPROM (Electrically Erasable Programmable Read-Only Memory) library in the Arduino code, so we would be able to store and get certain values for it. This memory is non-volatile, which means it would not be lost even if the power is turned off of the Arduino. The Arduino UNO has 1024 bytes of EEPROM memory but as we were storing float variables, that each of them use 4 bytes, the storing capabilities would be reduced to 256 different components.

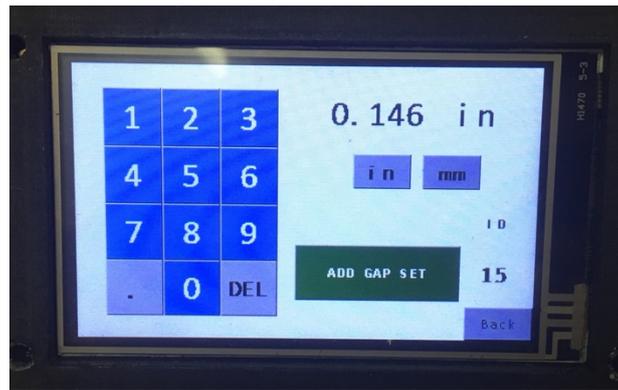


Figure 25: GUI-Add New Gap Set Option

Finally, the last function introduced in the display was the capability of the user to move the motor when necessary. Sometimes, the motor would not finish in the desired position and it would not be able to grab the bimetallic component correctly, therefore, we added this function: being able to move the motor clockwise and counter-clockwise a certain number of steps.

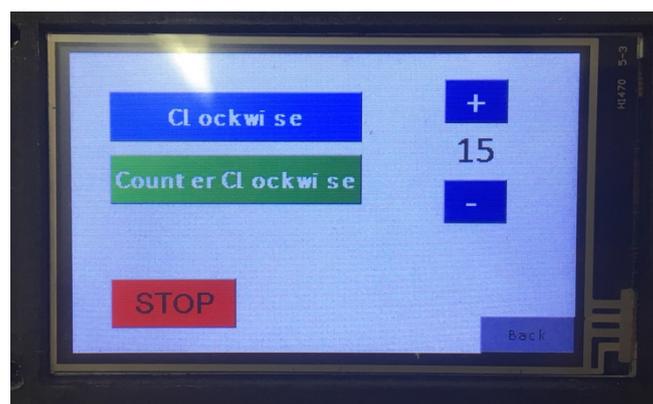


Figure 26: GUI - Motor Movement Capability

The touchscreen we bought was partially programmed in a Windows software known as the *Nextion Editor*. It is a Human Machine Interface (HMI) that combines common TFT touchscreens with an onboard processor and memory⁴. We chose these types of touchscreens due to its ease of customization and the different possibilities in size. Consequently, all of the functions mentioned earlier had to be programmed both in the Arduino and in the touchscreen itself.

⁴ <https://nextion.itead.cc/>

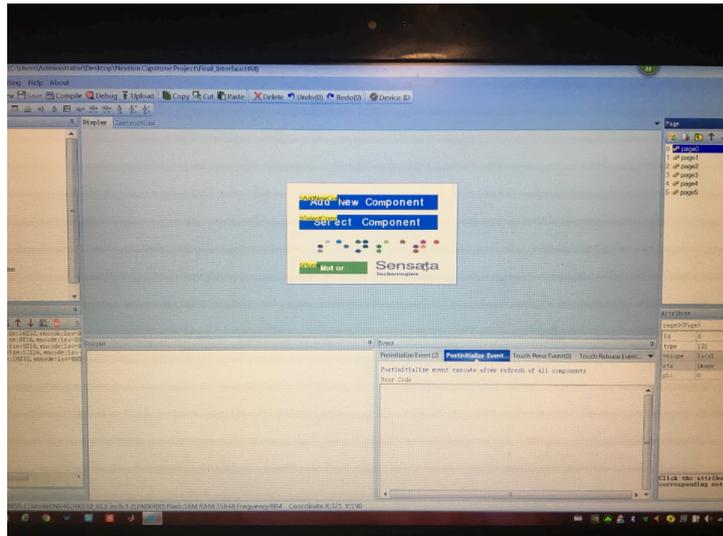


Figure 27: Nextion Editor

As an example, the number keypad was programmed within the Nextion Editor as well as the Exit and Back buttons but the motor movement functions and adding new components were all programmed in the microprocessor. To be able to start a communication between the touchscreen and the microprocessor we had to include the Nextion library. This library contained all the possible Nextion functions that could be used. All of the applications we used for the touchscreen are within the Arduino UNO's code attached in the Appendix.

Design Evaluations

Spring Back Factor

The bimetallic compensator, as mentioned before, had a high and variable spring back factor and this value was unknown. Consequently, after finishing the design and manufacture of the first prototype, we decided to test what was this value for the different types of compensators.

To conduct this experiment, we had to take some factors into consideration. In the first place, the pieces were made of metals, therefore, we could not test the same piece several times due to work hardening. By continuously plastically deforming a metal piece, the strength could increase significantly, so its properties would change, and the analysis would not be reliable. Furthermore, during the experiment we were very careful to place the components in the same position and totally introducing the U-tool in the bimetallic component.

To be able to calculate the spring back factor, we had to bend the piece sufficiently, so it entered the plastic region and did not come back to its original position. We recorded how much it bent for a certain value and repeated this procedure throughout the different pieces Sensata Technologies provided us with.

The next figure shows how we did the experiment:

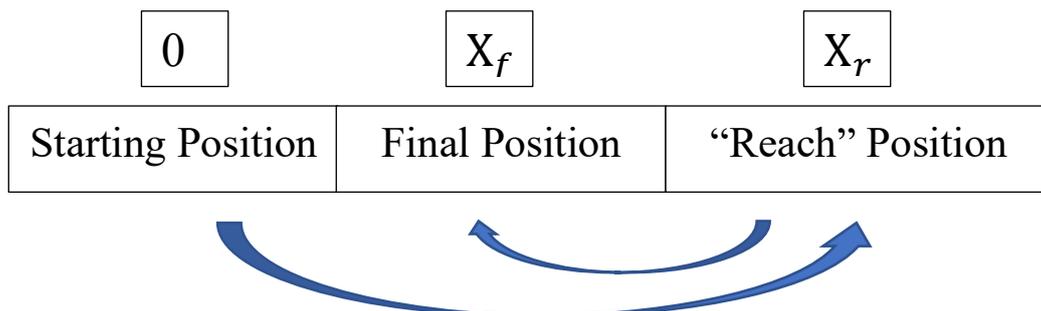


Figure 28: Springback Factor Analysis

The result show two different values:

- Reach value: it is the position reached when the component has been overbent, before the spring back has happened
- Final value: the value after the compensator springs back

The data shown below represents the spring back factor of one component.

Springback Effect

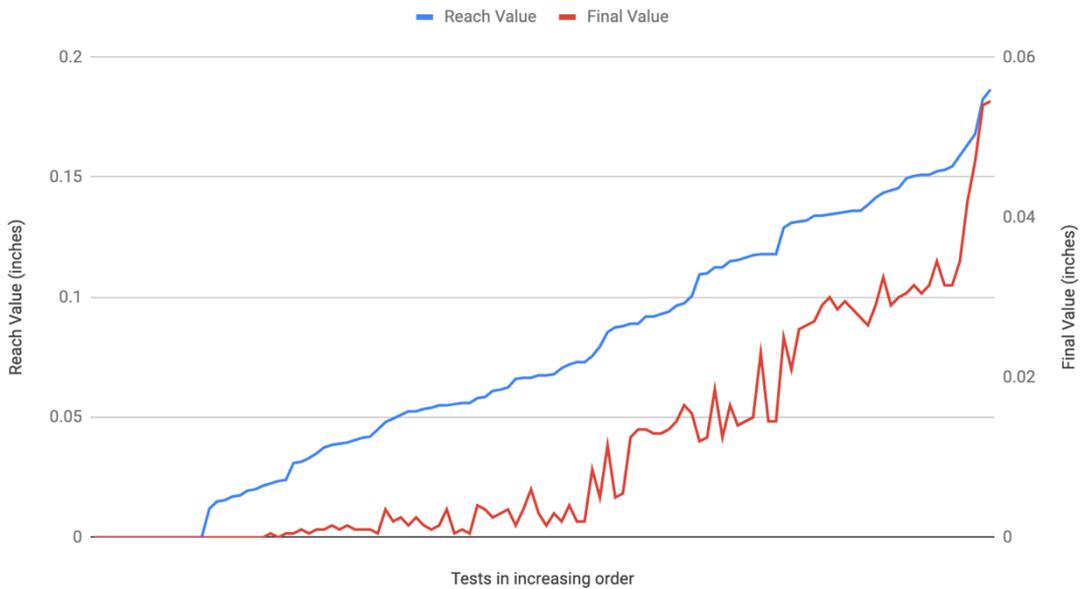


Figure 29: Graph - Springback Factor

As we can see in the graph above, by reaching a higher value when overbending, the plastic deformation of the piece increased, which is logical. Analyzing the data above, it can be clearly when does the piece enter the plastic region as there is a part of the graph where it does not bend (elastic region). To better understand the relation between these two values we created the next graph:

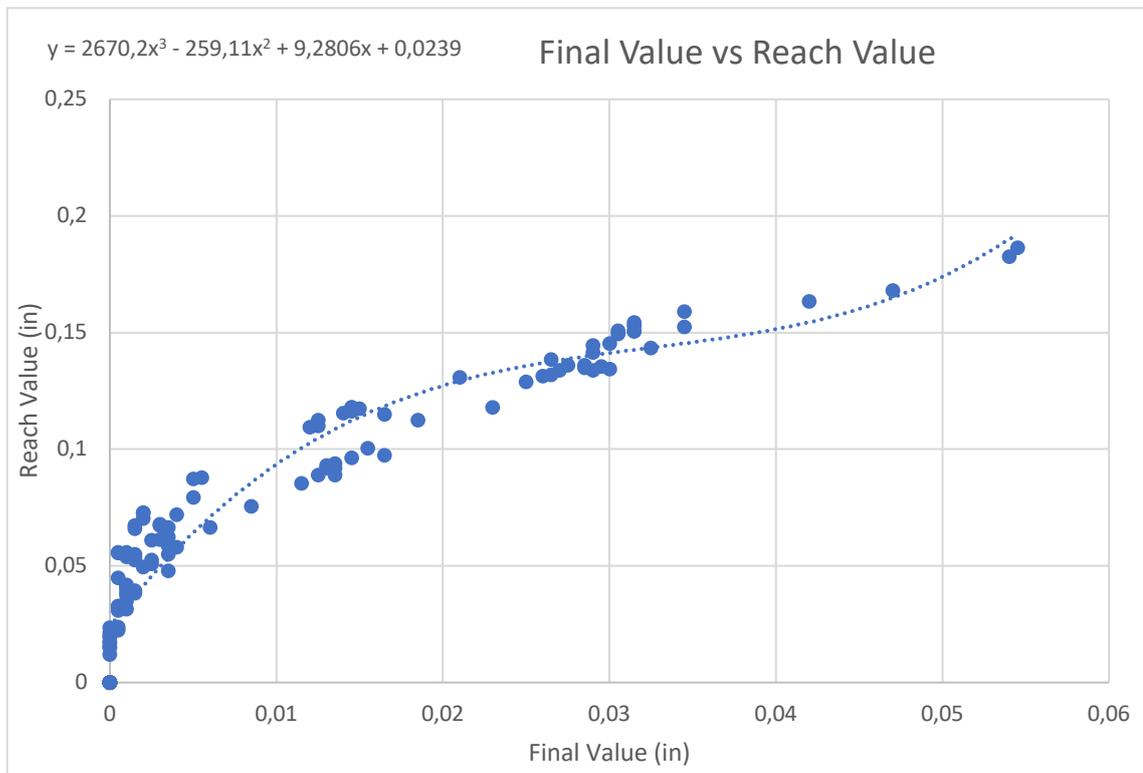


Figure 30: Graph - Final Value VS Reach Value

The spring back factor would be:

$$\text{Spring Back } \left(\frac{\text{in}}{\text{in}}\right) = \frac{\text{Reach Value}}{\text{Final Value}}$$

Looking at the graph above, we can see that the spring back reduces significantly over time. For the real calibration process, we would have to calculate the Reach value to obtain a certain final value. To obtain this value we did a regression and obtained the next equation:

$$\text{Reach Value (RV)} = 2670.2 * FV^3 - 259.11 * FV^2 + 9.2806 * FV + 0.0239$$

We would use this equation in the Arduino code. However, in the code, the FV was the difference between the objective gap set and the current gap set and the real reach value would have been RV + current gap set. An example of the procedure would have been:

$$\text{Objective Gap Set} = 0.173 \text{ in} \quad \text{Current Gap Set} = 0.155 \text{ in}$$

$$\text{Difference} = 0.173 - 0.155 = 0.018 \text{ in}$$

$$\text{Reach Value} = 2670.2 * 0.018^3 - 259.11 * 0.018^2 + 9.2806 * 0.018 + 0.0239 = 0.123 \text{ in}$$

$$\text{Real Reach Value} = \text{Reach Value} + \text{Current Gap Set} = 0.123 + 0.155 = 0.277 \text{ in}$$

However, we saw that these values were too high so decided to multiply by a factor of less than 1.

Results

After designing and manufacturing the final prototype, we headed to test the calibration tool and see what the time results were. As mentioned earlier, we decided that the equation that related the reach value and the final value had to be multiplied by a factor less than 1.

The time results that are shown in the next figure combine three different components as we did not see any great difference between them.

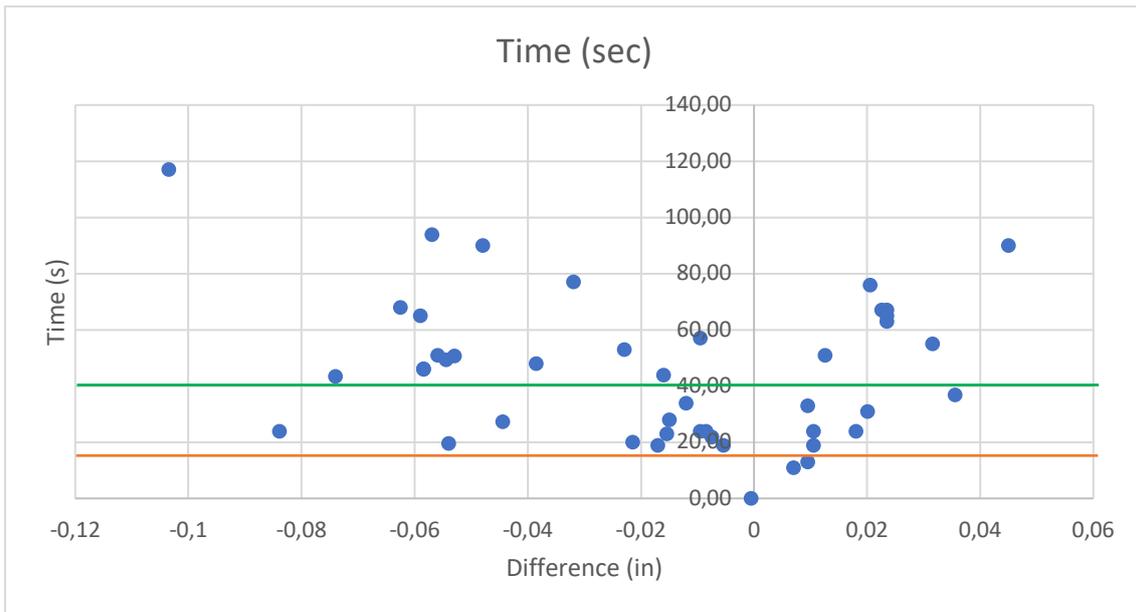


Figure 31: Graph - Results of Calibration Process

Although the requirement of having an average calibration time under 15 seconds was not achieved, most of the components that lie within a certain distance can be calibrated below 40 seconds. It is interesting to see that the negative differences calibration times are much more scattered than the positive differences. The possible reason for this is that the factor that multiplies negative distances is lower than the factor used for positive distances.

Furthermore, we had an efficiency of over 90% for the calibration process and the fastest calibration time was of 11 seconds.

The variation of the calibration time can be contributed to the initial, uncalibrated gap set of the bimetallic components, where some can be further away from the desired gap set. In addition, we realized that the mathematical model we used to calculate the spring back factor varies in effectiveness, and that the model is more accurate in predicting the spring back of the bimetallic component when the current gap set is relatively close to objective gap set. When the current gap set is far away from the objective gap set, the calibration module often needs to repeat the calibration process twice until the desired value is reached.

The calibration quality of the final prototype is satisfactory. After each calibration process, the Digital dial indicator rechecks the new gap set value and compare it with the objective value, and Arduino will repeat the calibration until the result falls within the tolerance of ± 0.002 inch. The tolerance is smaller than the manual calibration tolerance of ± 0.003 inch at Sensata, which indicates the higher accuracy and precision of the final prototype.

The User Interface allows the operator to easily use the device to calibrate bimetallic components without high skill dependency, and the user is able to select existing component gap set settings and add new settings, which significantly improves the user experience.

Our customers at Sensata were satisfied with our final prototype. Roberto Rivera, one of the design engineers at Sensata, showed his excitement for the end result after watching our team's live demonstration and final presentation. Roberto especially

liked the spring back empirical model and found the final prototype useful for future production at Sensata. Overall, this project met the mandatory objective of automatic gap set calibration and achieved desirable objective of implementing a easy to use interface, and it is approved by our customer.

It is important to highlight that if we had had more time and budget, the calibration tool process could have been optimized much more, which will be commented in the future improvements section.

Cost Analysis

The mechanical engineering department of Boston University gave us a budget of \$400 to complete this project. Throughout its evolution, we had to account for every part we bought and finally calculate the approximate cost of the final prototype. The next figure shows the costs tracking during the semester.

Table 5: Cost Analysis

Description	Funds In	Funds Out	Remaining Balance
Initial team allocation	\$ 400.00		\$ 400.00
Digital Dial Indicator		\$ 29.99	\$ 370.01
Metal Gear - 64 Teeth		\$ 34.23	\$ 335.78
Metal Gear - 20 Teeth		\$ 19.60	\$ 316.18
Shaft 0.25"		\$ 6.79	\$ 309.39
Stepper Motor		\$ 26.00	\$ 283.39
A4988 Driver for Stepper Motor		\$ 8.49	\$ 274.90
TB6600 Stepper motor driver		\$ 13.79	\$ 261.11
Nextion Enhanced HMI Touchscreen		\$ 33.99	\$ 227.12
AC/DC Converter		\$ 18.99	\$ 208.13
MicroSD 8GB card		\$ 3.99	\$ 204.14
USB SD Card reader		\$ 6.99	\$ 197.15
Linear Guide		\$ 42.15	\$ 155.00
Flange Bearing and 0.5" Aluminum		\$ 68.84	\$ 86.16
	Total Spending	\$ 313.84	

All pieces that were designed to be cut using a CNC mill were designated as Aluminum. Aluminum has a high strength, low weight, high sturdiness, easy machinability, and relatively cheap cost allowed us to utilize aluminum for different pieces with different functions. Almost all the pieces were made from a 0.5" or 0.25" aluminum stock.

The team discussed different options for the interface such as a LCD screen with a keypad shield but the team ended up deciding on a Nextion Enhanced HMI Touchscreen so that we would be able to program an interface that would be more intuitive and easy to use than the other options. The U-tool piece was fixtured using bearings, but we replaced those with flanged bearings to prevent vibrations.

A stepper motor was used due to its superior accuracy in comparison to other motors. The motor driver allowed for small current control signals to be converted into a high current signal that was able to drive the motor. The gears were used to increase the torque and the number of steps of the motor. With a larger number of steps in the larger gear, the team could take more accurate measurements of how many steps it took to bend the bimetallic component a certain amount.

The linear slide was by far the most expensive component. The motor, electronics, and gears all accounted for large portions of the budget as well.

Future Improvements

After finishing the final prototype and analyzing the different results obtained, we are proud of the project resolution. Although we have not achieved a calibration time below 15 seconds, the calibration process is efficient and works for every possible gap set. It is true however, that there are certain areas of improvement:

1. Reduction of the total calibration time: As mentioned earlier, we did not accomplish our set time of under 15 seconds. Nevertheless, with more time, we would have been able to optimize the code and improve significantly the process.
2. Reduction of noise from gears: Throughout the whole process we noticed that the use of gears was inducing a big amount of noise during the calibration process. This is due to the fact that the torque applied is high and the gear's teeth collide with each other. This could be reduced by limiting the current of the stepper motor. However, this would reduce the torque applied and the U-tool wouldn't be capable of adjusting the gap set.
3. Stability improvements: It is important to highlight that the efficiency of the process does not reach 100% due to some problems with the Digital Dial Indicator. Every now and then, the reading of the drop gauge suddenly increased or decreased by 0.1 inches, inducing in errors and stopping the process. We were not able to find an explanation for this problem; it is possible that the DDI was not as reliable as we wanted.
4. In the future, Sensata would like to export and analyze the data of the calibration time and spring back factors for the different components. These exporting capabilities would need a computer to communication via the serial port with the Arduino UNO and export the data to an excel file.
5. Finally, an improvement that was thought after the presentation of the project was of memory allocation: instead of storing float numbers, integers could have been stored and then dividing by the corresponding number. Consequently, as integers are stored in one byte, the capacity would increase to 1024 different pieces.

Conclusion and Recommendations

At the conclusion of this project, the team was able to successfully design and manufacture a prototype capable of adjusting the calibration of bimetallic components. This prototype should not be viewed as a final prototype but as a framework that Sensata will use to implement their own design into the current process. After presenting the device to Sensata, they expressed their satisfaction in the design. They told us that they were especially impressed in how we accounted for the spring back with the special U-tool mechanism coupled with how we determined whether to bend CW or CCW and by how much the motor needed to spin to bend the component.

When Sensata incorporates our design into their process, they should look into using more powerful circuitry and hardware so that they may be able to compute more complex solutions more efficiently to determine the necessary bend. They may also improve on the hardware design, for example, utilize magnets to achieve easier loading of the bimetallic pieces, and adding a shell over the motor gear component for noise reduction and safety.

References

[1]. Arduino Reference

[2]. Sensata Technologies, <https://www.sensata.com/>

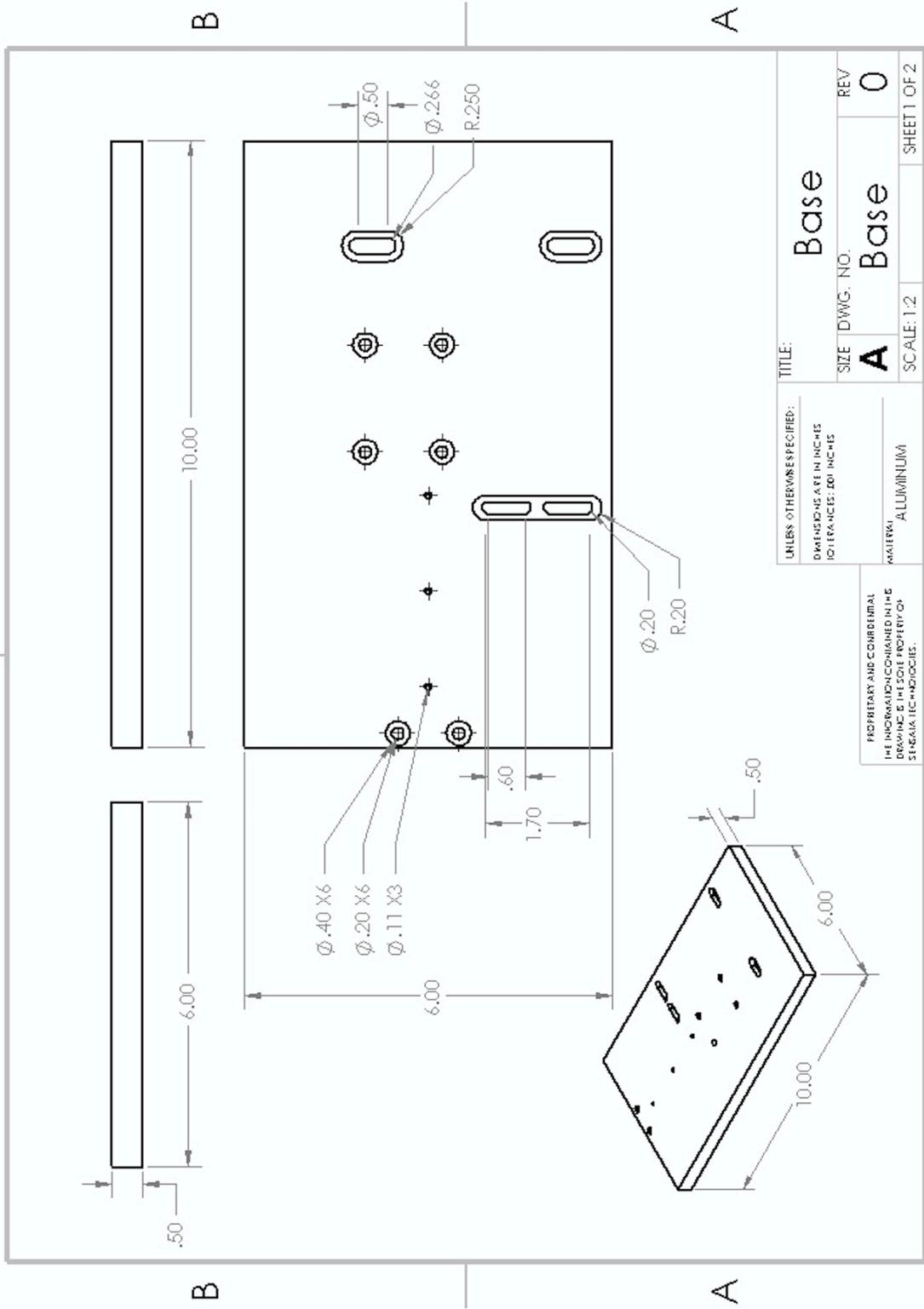
[3]. "The Digital Dial Indicator and How to read from it", Justin R. 2014.,
<https://hackaday.io/project/511-digital-dial-indicator-cnc-surface-probe/log/814-the-digital-dial-indicator-and-how-to-read-from-it>

[4]. "Using Digital Caliper for Digital Read Out (DRO) Applications, 2012.,
http://wei48221.blogspot.com/2016/01/using-digital-caliper-for-digital-read_21.html

[5]. Nextion Editor – Nextion, <https://nextion.itead.cc/resources/download/nextion-editor/>

1

2

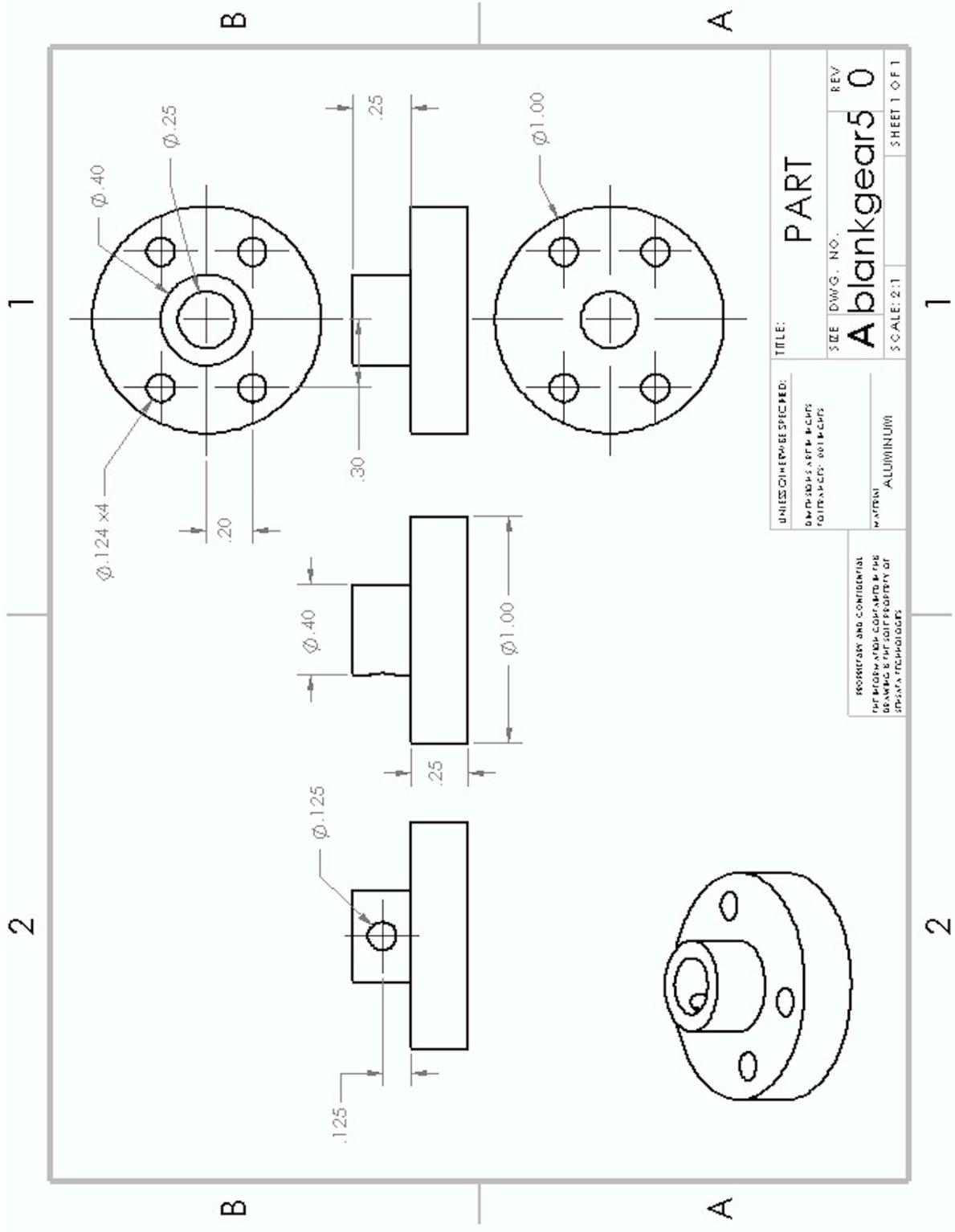


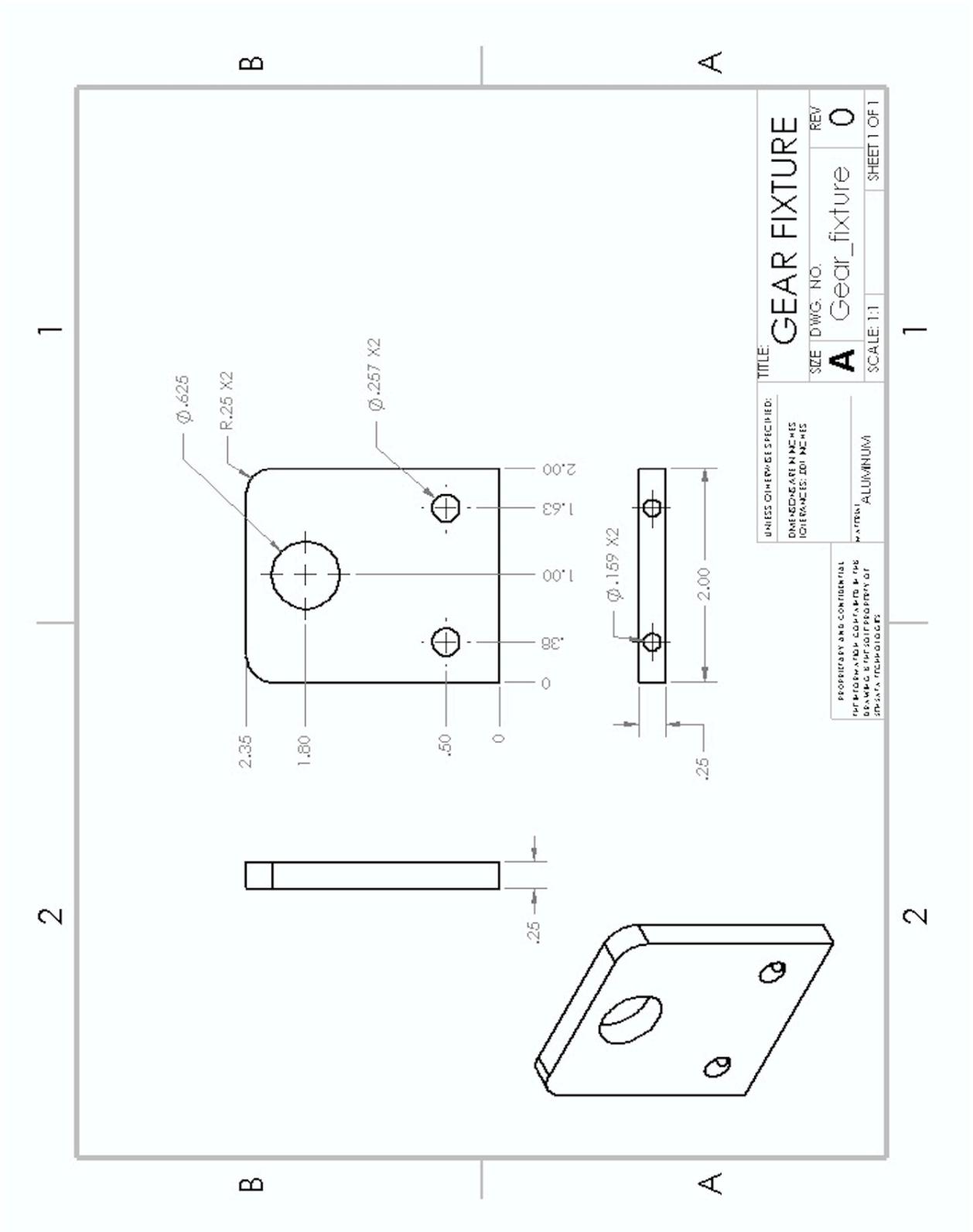
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN INCHES TOLERANCES: .001 INCHES		TITLE: Base		REV
SIZE	DWG. NO.	A	Base	0
MATERIAL: ALUMINUM		SCALE: 1:2		

PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS
DRAWING IS THE SOLE PROPERTY OF
SENEXA TECHNOLOGIES.

1

2



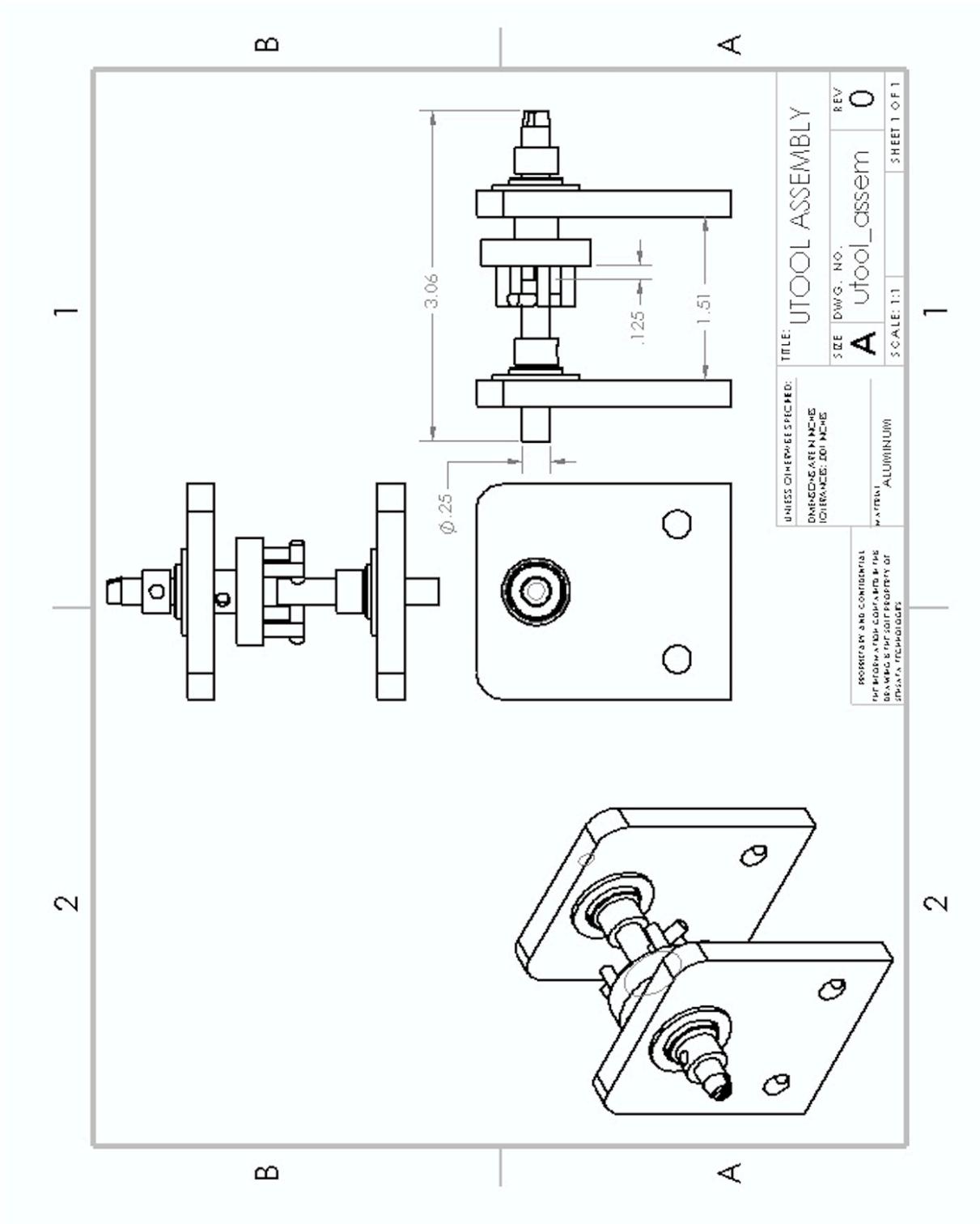


TITLE: GEAR FIXTURE		REV
SIZE	DWG. NO.	0
A	Gear_fixture	0
SCALE: 1:1	SHEET 1 OF 1	

UNLESS OTHERWISE SPECIFIED:
 DIMENSIONS ARE IN INCHES
 TOLERANCES: .001 INCHES

MATERIAL: ALUMINUM

PROPRIETARY AND CONFIDENTIAL
 INFORMATION CONTAINED IN THIS
 DRAWING IS THE PROPERTY OF
 SPARTA TECHNOLOGIES



Stepper Motor with Arduino

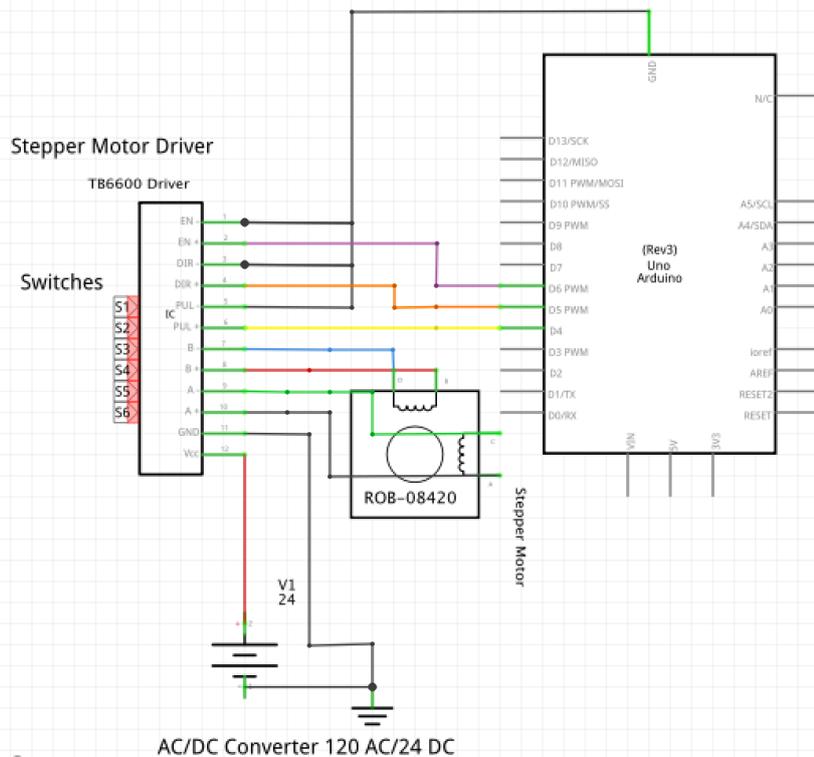


Figure 33: Stepper Motor Circuitry

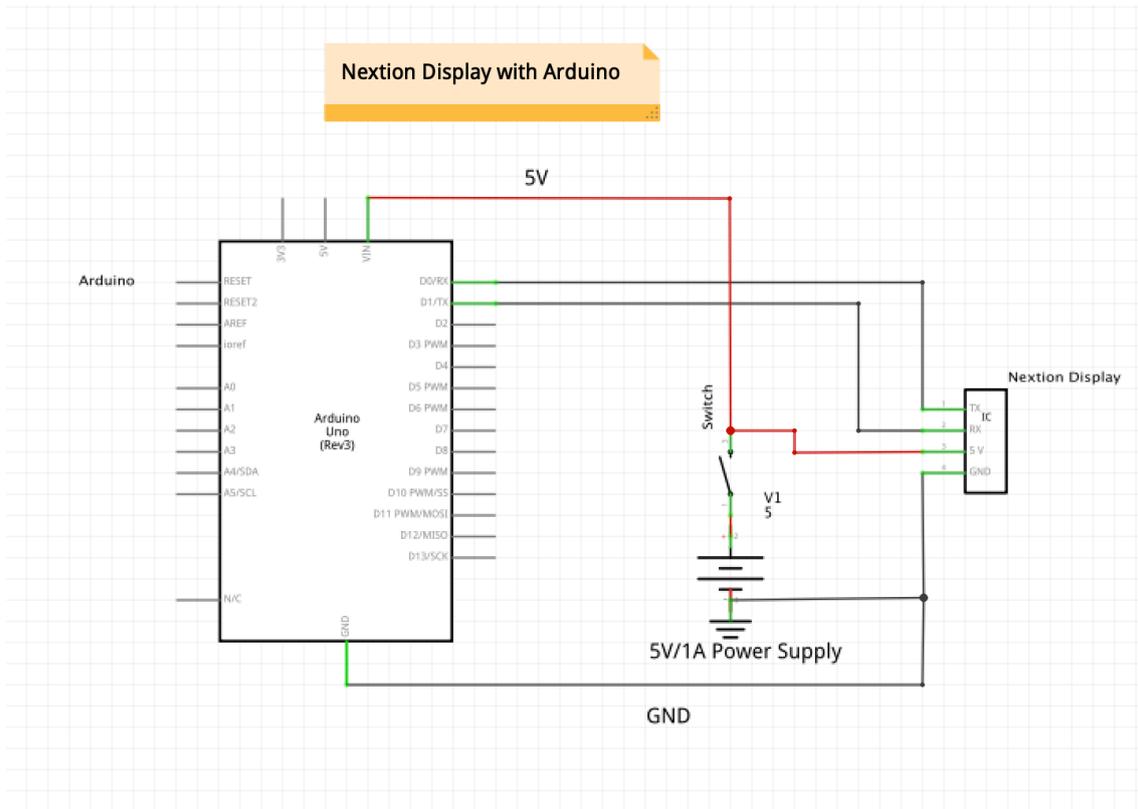


Figure 34: GUI Circuitry

Arduino Code

This is the code for the extraction of data for the Digital Dial Indicator, it is included in the appendix as it could be useful for other types of apparatus such as a Digital Calipers.

```
void Reading_DDI (){

    // This is the reading function
    // micros() is a function in the arduino that counts how much time it has been ON
    // so we set the variable counter to be equal to micros()
    counter = micros(); //important to take into account micros() can only count up to
    70 minutes
    while(digitalRead(clockPin)== HIGH){} //we don't do anything when the clockPin is
    HIGH
    while(digitalRead(clockPin)== LOW){} //we don't do anything when the clockPin is
    LOW
    /*
    * This means we have waited one whole pulse, after this pulse
    * we check much time has passed, if its bigger than the transit time,
    * we will be at the beggining of a 24 bit cluster, if not that means
    * we are in the middle of a number so we don't call the function
    * that extracts the binary number
    */
}
```

```

    //It is important to highlight that we need to read the data pin when the clockpin
    goes
    //from LOW to HIGH

    // Now we check how much time has passed between micros() and counter
    // if its bigger than the transit time this means we are at the beggining
    // of a new number so we would enter the loop
    if((micros()-counter) > transitTime){

        // descifrar() is the function that determines the DDI's number
        finalvalue = descifrar();

        // this is not important, just checks if the previous value is the same as before
        if(newvalue == 1){
            if(finalvalue != previousvalue){

                previousvalue = finalvalue;

            } else {

                newvalue = 0;

            }
        }

    }

}

float descifrar(){

    unsigned int value = 0;
    signed int value_sign;
    float good_value; //this is the variable returned
    float previous_valor;

    //this is an array of integers that will store the different binary digits
    int bin_value[24]; //this vector has a space of 24 numbers
    int sign;

    //We need to separate the first bit from the rest because we are already
    // in the first rising of the clockPin, so we directly read the data PIN
    for(int x = 0; x<1;x++){
        bin_value[x]=digitalRead(dataPin);
    }
}

```

```

}

// However, for the rest of the bits, we need to wait until the clockPin is rising again
// therefore, we don't do anything when its HIGH or LOW, after the second while
loop has finished
// we are at the rising of the clockPin and we read the data PIN
for(int x = 1; x <=23; x++){

    while(digitalRead(clockPin)==HIGH){}
    while(digitalRead(clockPin)==LOW){}
    bin_value[x]=digitalRead(dataPin);
}

//The sign and the units bit were analysed beforehand
//if bit number 20 was HIGH it was a negative number, if its was LOW it was a positive
number
//The sign bit is bit number 20, positive or negative

if(bin_value[20]== 1){
    sign = -1;

}else if (bin_value[20]== 0){
    sign = 1;

}

//if bit number 23 was HIGH the units were inches, if its was LOW it was "mm"
//The unit bit is bit number 23, inches or mm
if(bin_value[23]== 1){
    divider = 2000.00;//to get the real number that the DDI was showing you had to get
the binary number read and divided it by 2000
}else if (bin_value[23]== 0){
    divider = 100.00;
}

//only the first 16 bits represented the number so we just had to read these.
for(int i=0; i<16; i++){ //we only get the values that give us information, that is, the
first 16 binary numbers
    //in the binary number read the Least Significant figure was to the left, opposite to
real binary numbers
    //However, just by using the function bitWrite, the number is inverted.
    bitWrite(value, i , bin_value[i]);
}

//We multiplied the value by its sign and divided by its divisor
value_sign = value*sign;

```

```
good_value = value_sign/divider;

//this variable is HIGH as we have a newvalue
newvalue = 1;

//we only return if the value is in inches.
if(divider == 2000.00){
    return good_value;
}
}
```