DEGREE PROJECT IN MECHANICAL ENGINEERING,
FIRST CYCLE, 15 CREDITS
STOCKHOLM, SWEDEN 2021

Anti-lock braking system for bicycles

KRISTIAN JANDRIC

LUCAS ANDERSSON

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF INDUSTRIAL ENGINEERING AND MANAGEMENT

Anti-lock braking system for bicycles

KRISTIAN JANDRIC
LUCAS ANDERSSON

Bachelor Thesis at ITM
Supervisor: Nihad Subasic
Examiner: Nihad Subasic

TRITA-ITM-EX 2021:25

Abstract

An attempt was made to construct an ABS system that would both lock the wheel and release the brakes. The system would be mounted on a bicycle with v-brakes. It would then be tested if it could decrease the braking distance and if the system would respond fast enough. A literature study was made to learn what was needed for such a project. After many attempts of using re-purposed components an ABS system would eventually be built with a new stepper motor, and it was strong enough to lock the back wheels. Unfortunately the system could not be as thoroughly tested as expected, where only the reaction time of the system could be tested and not the braking distance due to a motor driver failure prior to the tests taking place. Due to shipping times a our budget and time constraints, further testing could not be done.

Keywords: Mechatronics, Bicycle, ABS, Arduino, Stepper motor, Brakes

Referat

Antiblockeringssystem för cyklar

Ett försök att bygga ett ABS system som både låser hjulet samt släpper på bromsen gjordes. Systemet skulle kunna monteras på en cykel med fälgbromsar. Systemet skulle testas genom att mäta skillnaden i bromssträcka samt om reaktionstiden var snabb nog. En litteraturstudie gjordes för att få tillräcklig kunskap om vad som krävdes för ett sådant projekt. Efter många försök med att använda olika återanvända komponenter kunde ett ABS system till slut konstrueras med hjälp av en ny stegmotor, som var stark nog för att låsa bakhjulet. Tyvärr kunde bara systemets reaktionstid testas och inte bromssträckans förändring. Detta berodde på en motordrivare slutade fungera. På grund av frakttider och en fast budget samt en tidsbegränsning, kunde inte ytterligare tester genomföras.

Nyckelord: Mekatronik, Cykel, ABS, Arduino, Stegmotor, Broms

Acknowledgements

We would like to thank our teaching assistant Amir Avdic for helping out with finding re-purposed components and other tips and tricks. Martin Olanders and Erik Anderberg made for good company during the time spent tinkering and helped with some photo backgrounds and shared the site where we created the wiring diagram. Finally we would like to thank our examiner Nihad Subasic for spending the time grading this project.

Contents

List of Figures

List of Tables

List of Abbreviations

ABS - Anti-lock braking system

MCU - Microcontroller unit

PCB - Printed circuit board

LED - Light emitting diode

IDE - Integrated Development Environment

CPU - Central Processing Unit

I/O - Input/Output

DC - Direct Current

PLA - Polylactic Acid

IR - Infra Red

US - Ultra Sonic / Ultra Sound

3D - Three Dimensional

MEMS - Microelectromechanical systems

Chapter 1

Introduction

1.1 Background

In almost all of today's modern road fairing vehicles ABS-systems are used. The purpose of an ABS-system is to keep control of the vehicle while braking and to prevent the vehicle from skidding, and help keeping traction between the wheels and the road. The method by which this is achieved is by repeatedly pressing and releasing the brakes and keeping the wheels rotating [1]. If a wheel were to stop rotating by braking too hard the wheels are said to have ”locked up”. If the wheels are locked up but the vehicle still has forward momentum, the vehicle will slide until it has slowed down enough to stop or sometimes if the brakes have been released, letting the wheels turn freely again. A sliding vehicle has very limited steering and is difficult to control, and to let go of the brakes might seem counter-intuitive but could help to regain traction and control of the vehicle. The majority of people's first reaction is to do something intuitive rather than something counter-intuitive, which in this case could lead to a bigger accident. ABS is designed to help the driver by releasing and reapplying the brakes automatically if the driver is braking as hard as possible while sliding to hopefully regain control of the sliding vehicle.

As more and more cities build for a more sustainable and green future, the bicycle might play a bigger role in the city transportation. Compared to other vehicles within the city limits, the bicycle driver is more susceptible to injury. Further safety features on a bicycle might be needed for people to feel safe while riding them. An ABS-system might let the cyclist have more control and decrease the braking distance, which makes for a safer ride.

1.2 Purpose

Many people get injured in bike accidents every year, and having brakes that make for shorter braking distances and more stability might lead to fewer accidents[2]. Our research questions are as follows: Will an ABS on a bicycle

- Decrease the braking distance? And if so, by how much?

- Be able to respond fast enough to a wheel locking up?

1.3 Scope

This project will be limited to a time frame of about four months. This in turn will only allow for prototypes to be built, and not fully fledged designs. Parts, such as micro-controllers and cables, will be supplied by the school institution. It will also be possible to use different parts & components used in previous projects that can be found in the institutions storage. Additionally a budget of 1000 SEK can be used to buy new components. This project will only focus on applying ABS to a bicycle with v-brakes.

1.4 Method

A literature study was conducted to gain the necessary knowledge and understanding to create a prototype system. By designing parts in CAD to be 3D printed and by soldering wires and other components to blank PCB:s to later be attached to a bicycle, an ABS system could theoretically be created. By attaching a motor and a brake button to the handlebars, a disc with slits on the back wheel with a ultra sonic (US) sensor to keep track of the wheel rotation and by attaching and connecting everything with zip ties and 3D printed parts to an microcontroller unit (MCU), the system could drive the motor with a stepper motor driver to create a working ABS system.

The response time of the ABS system was measured by wiring a Piezo speaker and a red LED to the MCU, which would output a tone and light if the system recognised that the back wheel was spinning. If the wheel was not spinning there would be no tone or light emitted. The response time of the system was then determined to be the time between when the wheel locked and the light and speaker stopped emitting. The measurements where made by having a cyclist accelerate the bicycle up to a significant speed and press a button to activate the light and speaker. Once the cyclist had achieved a significant speed he pressed down the brake handle as hard as possible, locking the rear wheel. A smartphone camera was used to film the speaker, LED and rear wheel. The footage was then analysed on a computer by slowing it down to 0.13X speed. From the film a response time was determined using the method described above.

To measure the braking distance the idea was to accelerate the bicycle up to a specific velocity when passing a marked point on a track. After passing this point the driver used the rear wheel brakes to decelerate the bicycle to full stop, with and without the ABS activated. The braking distance was then determined by measuring the distance between the stopping point and the marked point. Unfortunately the braking distance test could not be conducted as the only functioning motor available did not have high enough torque to lock either of the wheels on the bicycle.

Chapter 2

Theory

In this chapter the theory needed to understand the components used for this project are described.

2.1 ABS system

An ABS, or Anti-lock Braking System, is used to stop the wheels on a vehicle from locking up during braking. The theory behind this system is that while the wheels are rolling, the vehicle has to go in a direction parallel to the wheels. Going in another direction other than parallel, for instance perpendicular to the direction of the wheels, can only be achieved by losing traction and sliding. ABS systems work by letting go of the brakes if the system detects a lock up, letting the wheel spin again and regaining the lost traction from sliding and later let the brakes engage again to try and slow the vehicle down. If the brakes lock up again, the system kicks in and releases the brakes until the wheels start to roll again. Waiting for the wheel to lock up completely before releasing the brakes will most likely work, but detecting that the wheel is close to locking up before it does so will make the system even safer[3].

2.2 V-brakes

The V-brake is a common type of bicycle brake, often mounted on the front wheel of the bicycle. The V-brake used in this project is made up of two rubber plates attached to two arms, with one of the arms connected to a cable-housing. To activate the brake one must pull a cable, which runs through the cable-housing and connects to the other arm. The pull will bring the arms close together and thereby pressing the rubber plates against the wheel rim, causing friction to slow down the wheel rotation [7].

Figure 2.1. Back wheel V-brake. *Photo taken at KTH mechatronics lab.*

2.3 Braking and releasing implementation

The motor has to be strong enough to brake the bicycle[4], so a rough estimation of the force needed to brake was done.

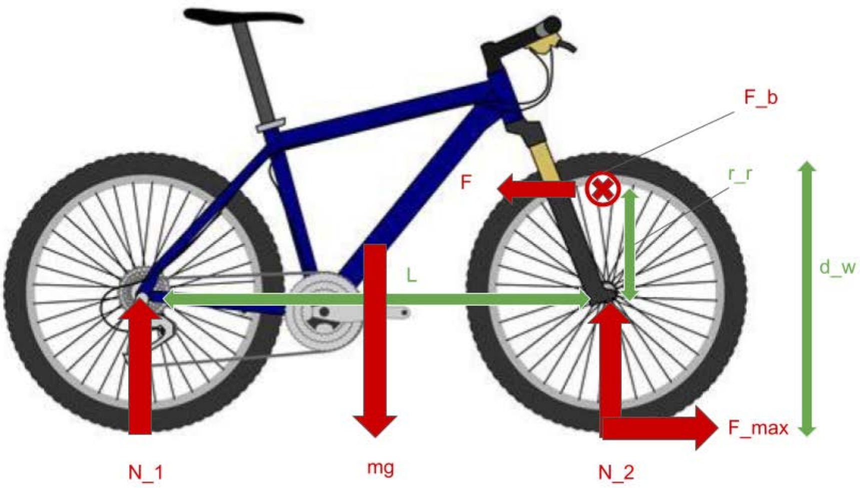


Figure 2.2. Free body diagram of the bicycle.
Illustration from *www.clipartpanda.com*. Forces added in *Google Slides*.

Bicycle parameter	Symbol	Value	Unit
Wheelbase	L	100	cm
Wheel diameter	d_w	55	cm
Rim radius	r_r	25	cm
Weight distribution	Δ	50	%
Brake pad length	l_b	5,5	cm
Brake pad height	l_h	1,2	cm
Bicycle mass	m	15	kg
Bicycle and driver mass	m_{tot}	80	kg
Friction coefficient between wheel and ground	μ_{ground}	0.8	-
Friction coefficient between rim and brake pad	μ_{brake}	0.6	-

Table 2.1. Bicycle parameters

From figure 2.2 we can see that the maximum braking force that can be applied on a wheel is:

$$F_{max} = m_{tot} \cdot g \cdot \mu_{ground} \cdot \Delta \tag{2.1}$$

Where m_{tot} is the weight of the bicycle and cyclist, g is the gravitational acceleration, μ is the friction coefficient between the wheels and the ground and Δ is the weight distributed on the front wheel. We roughly approximated the weight distribution to be 50% on each wheel and the friction coefficient was approximated as it is hard to get a correct value [5].

The braking force from a force applied on the brake handle can be approximated as

$$F = \frac{2 \cdot F_{handle} \cdot r_r \cdot \mu_{brake}}{\frac{d_w}{2}} \tag{2.2}$$

Where F_{handle} is the force applied on the brake handle, r_r is the radius from the center of the wheel to the rims where the v-brakes are engaged, μ_{ground} is the friction coefficient between the brake discs and the rim and d_w is the wheel diameter. Everything is multiplied by 2 as there are 2 brake pads in the v-brake. The friction coefficient μ_{brake} is approximated to be 0.6 as it was found in a lookup table[6]. The point at which the bicycle will lock up is if the braking force applied from brakes, F, is equal to or greater than the maximum braking force F_{max} (or $F \geq F_{max}$). If the leverage from the v-brakes and a leverage arm extending the length of the brake handle are taken into account as the coefficient α, the force needed to lock the back wheel, F_{handle}, can now be rewritten as:

$$F_{handle} = \frac{F_{max} \cdot d_w}{4 \cdot r_r \cdot \mu_{brake} \cdot \alpha} \tag{2.3}$$

The coefficient α, as previously said, is composed of two different leverages where the first one comes from the leverage of the v-brakes, and the second one from the brake handle lever. As can be seen in figure 2.1 in the section on V-brakes, the leverage arms from the rotation point are approximately 2.5 cm to the braking pads and 11 cm to the point where the braking wires are attached. This gives a leverage of approximately 4.4 times more than initially. The distance the extended brake handle travels compared to the braking wire when braking as much as possible is 8.2 cm and 1.3 cm respectively. This gives a leverage of approximately 6.3. Altogether the α leverage will give around 28 times more force. Inserting α and all other parameters from the table 2.1 into equation 2.3 the total braking force needed is around 10N.

2.4 Components

2.4.1 Stepper motor

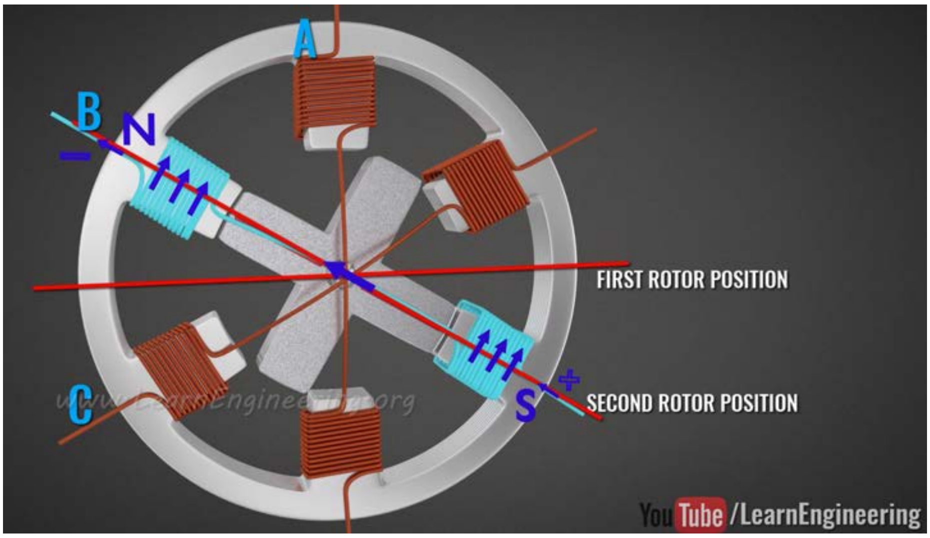


Figure 2.3. *'How does a Stepper Motor work ?' by LearnEngineering.org*

A stepper motor is a type of DC motor that rotates a polarised rod by feeding a current through coils, which surrounds the rod. The magnetic field created by the coils will allow the polarised rod to align it self with the magnetic field. By running a current through different sets of coils in a sequence, the rod can turn a fixed amount of degrees, or ”steps”. This is where the motor gets its name from, as it turns with the same step [8]. Figure 2.3 above shows a simple stepper motor. Stepper motors are often used for their precision and high torque.

2.4.2 Stepper motor driver

Because a stepper motor only can turn enough to align with the powered coils, these coils have to be powered in a specific sequence to be able to get the motor to rotate. A stepper motor driver is able to provide the motor with the necessary inputs to power the coils in this sequence[9].

2.4.3 Microcontroller

A Microcontroller unit (MCU) is a small single circuit computer typically comprised of a CPU, memory and I/O peripherals. Microcontrollers are designed to be embedded within a larger system, controlling a single or a few features of the system by running a single program[10]. They are therefore often designed to be run with relatively low power consumption. There are many popular microcontrollers that are suitable for hobby & student projects. One of these is the Arduino, which was used for this project. An Arduino is a small MCU that is easy to program and has multiple inputs and outputs making it great for prototyping.

2.4.4 Accelerometer

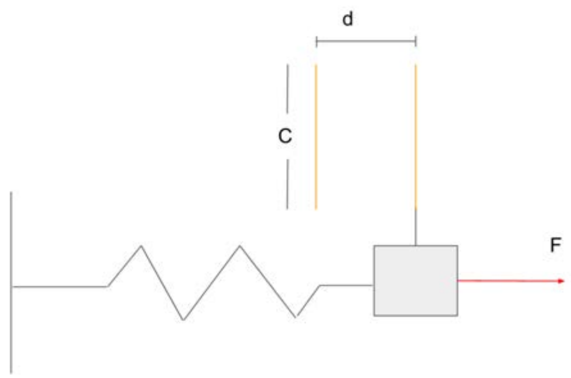


Figure 2.4. Spring mass system with a capacitor made in *Google slides*.

An accelerometer is a sensor used to measure acceleration. To demonstrate how a certain type of accelerometers work, a one degree spring mass system can be used. For a one degree spring mass system, as seen in figure 2.4, a relationship can be determined between the spring extension and the acceleration using Newtons second law:

$$ma = kx \tag{2.4}$$

Where m is the mass, a is the acceleration, k is a spring coefficient and x is the extension of the spring. An accelerometer may use different techniques for calculating the acceleration. The one used in this project uses capacitors. For a two sided capacitor the capacitance can be calculated with:

$$C = \epsilon \frac{A}{d} \tag{2.5}$$

Where ϵ is the permittivity, A is the area of the capacitors plates and d is the distance between the plates [10]. As seen in the formula the capacitance can changed by moving the capacitors plates closer or further from one another. Now imagine one of the capacitors plates mounted on a spring as in figure 2.4, if the plate is affected by a force it will accelerate and move the distance x. This will make it possible to calculate the acceleration by measuring the change of the capacitance. A typical accelerometer sensor uses MEMS to measure the acceleration. MEMS stands for Microelectromechanical systems [11]. For accelerometers it applies the spring-mass-capacitor theory described above on a micro scale, illustrated by figure 2.5 below.

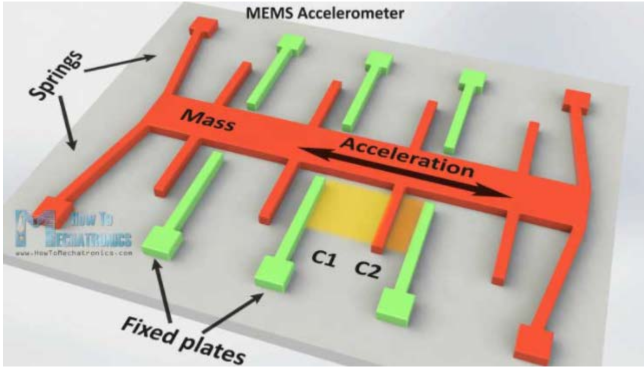


Figure 2.5. MEMS by *www.howtomechatronics.com*

2.4.5 Ultra sonic sensor

Sound waves which have a frequency of 20 kHz and above are called ultrasonic sound waves and are outside of the human hearing frequency spectrum. Ultra sonic sensors are often used to determine distances and are commonly found in many of today's road vehicles. They function by sending out a high frequency sound wave and measuring the time it takes for the wave to come back using the following equation[12].

$$s = \frac{t \cdot v}{2} \tag{2.6}$$

Where s is the distance from the sensor to the obstacle, t is the time it takes for the wave to bounce and come back to the sensor again, and finally v is the sound wave's velocity. We divide by two to get the one way distance between the obstacle and sensor. By using a pair of sensors, one transmitter and one receiver, the sensors can continuously calculate the distance to an obstacle.

Figure 2.6. Ultra sonic sensor HC-SR04 illustration by *www.howtomechatronics.com*

Chapter 3

Design

In this chapter the system and how it was constructed is described.

In order to convert the bicycle back to it's original state after the project was finished, a motor mounted directly to the handlebar will engage and release the brakes by connecting the motors axle to the handbrake. A button mounted on the other side of the handlebar will be the used as the braking button, engaging the brakes with as much power needed to lock the back wheel. Sensors will then be used to determine if the wheel is locked, and disengage the brakes using the same motor. If the button is still pressed when the wheel starts turning again, the cycle will repeat and the brakes will engage. This implementation of an ABS prototype would not be recommended for use on public roads, but will make testing and research significantly easier.

The reasoning behind these decisions were based on safety and consistency. Installing an ABS on the front wheel would be more beneficial because bikes have a tendency to throw the bike riders over the handlebar if the front wheel is locked, and minimizing the possibility of this would be ideal but also dangerous to test. To consciously try to lock the front wheel multiple times while trying this prototype could lead to injury and was therefore decided against.

Having a button to brake the bike instead of the rider using their hand to brake will make it easier to mount and operate the motor without the risks of hands getting injured, but also make for more controlled tests where a specific amount of force can be applied more consistently.

3.1 Hardware

3.1.1 Arduino / MCU

The microcontroller used for this project was an *Arduino UNO R3*. This type of microcontroller is open source and are manufactured by a multitude of different companies. In this case, the Arduino used was manufactured by the Swedish company *Kjell och Company.* The Arduino MCU is used to gather information from an ultrasonic distance sensor, accelerometer and a push button, and depending on if certain criteria is fulfilled operates the motor accordingly.

The final controller module is a compact layered design with the motor driver and other components soldered to a blank PCB board mounted with spacers above the MCU using the mounting holes in the MCU board. The connections between the PCB and the MCU are wires soldered to connector pins on smaller blank PCB boards. Connections to components mounted to different parts of the bicycle further away from the MCU module, such as the Ultrasonic distance sensor, are connected using white pins in top of the PCB. The final controller module can be seen in Figure 3.1 below.

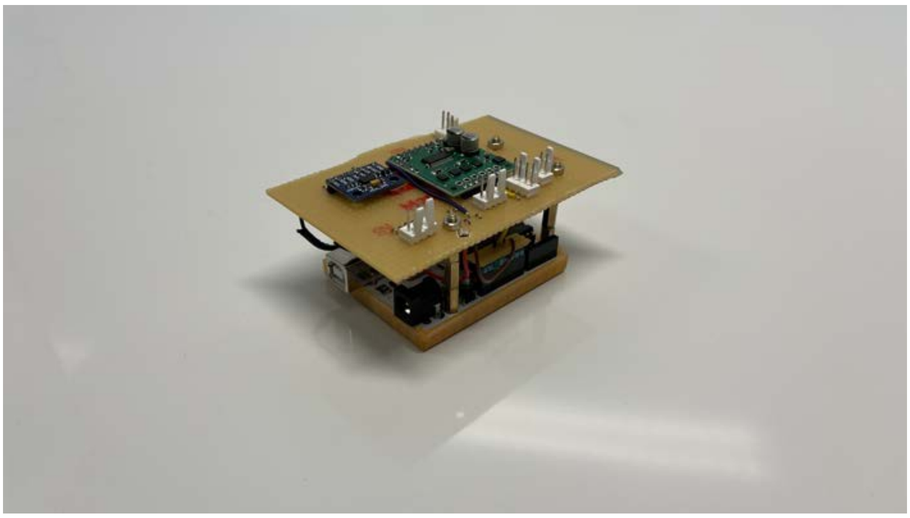


Figure 3.1. Layered controller module. *Photo taken at KTH mechatronics lab.*

3.1.2 Sensor system

Figure 3.2. Back wheel sensor system. *Photo taken at KTH mechatronics lab.*

To determine if the back wheel is locked or not, a US sensor of type *HY-SRF05* and a 3D printed disc with holes cut out was used. The disc was mounted on the wheel using zip-ties. The US sensor was mounted on a 3D printed block which was fitted to the frame of the bicycle. The block was positioned so that the sensors transmitted sound would be periodically blocked and let through the holes of the disc as the bicycle wheel spins. A photo of the rear wheel can be seen above in figure 3.2. As the US sensor is constantly emitting and receiving sound waves, the distance calculated will ideally continuously change between two distances as the wheel spins. By saving these distances on the memory of the MCU, it will be able to tell if the wheel is locked by comparing the recently saved values and see if they are constant or not. To make sure that the MCU knows if the bicycle is standing still or not, a accelerometer was soldered on to a electrical component board and wired to the MCU. In this project a *ADXl345* accelerometer was used.

3.1.3 Brake system

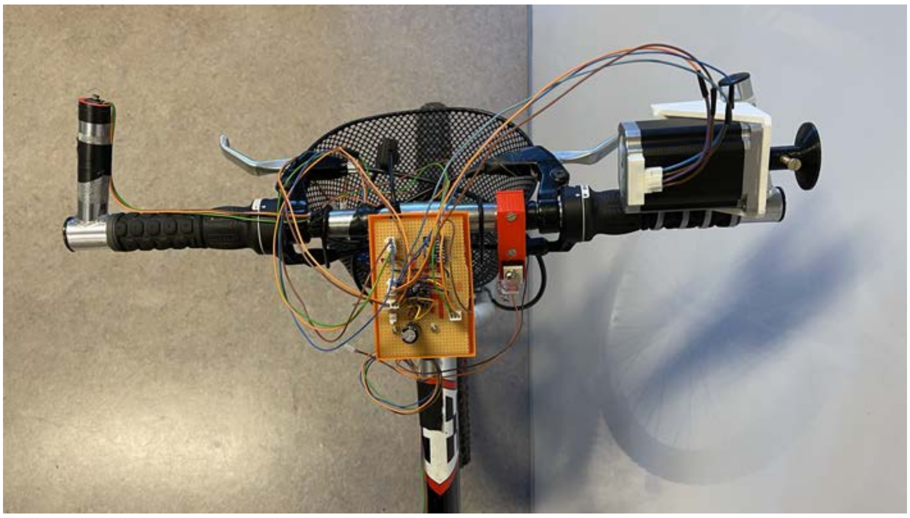


Figure 3.3. Drivers view of the bicycle. *Photo taken at KTH mechatronics lab.*

The *Sanyo Denki 103H7823-1740* stepper motor (see Appendix A) was chosen to be used in this project. Being the strongest motor within the budget and in the list of permitted stores/suppliers with a holding torque of 2.7Nm, it could be possible to add more leverage to achieve a braking force of around 10N as previously calculated from equation 2.3. The motor is mounted on a 3D printed plate (see figure .5 in appendix C), which is zip-tied to the right side of the handlebar (see Figure 3.3). On the motor shaft a coupling is fitted with a small screw using one of the couplings two holes. In the other hole of the coupling there is a screw which purpose is to transfer the motors torque with a string connected to the front wheel brake handle. To brake the bike a button is pressed on the left side of the handlebar. The button send a signal to a MCU which in turn makes the motor rotate clockwise. This will tighten the string and pull the brake handle. When the front wheel locks the MCU will make the motor rotate counter-clockwise and thereby releasing the brake. A photograph of the motor can be seen in figure 3.4.

Figure 3.4. Sanyo Denki 103H7823-1740 Stepper motor with 3D-printed axle attachment. *Photo taken at KTH mechatronics lab.*

Stepper motor driver

Stepper motors require specific electrical inputs to functions correctly, and an arduino does not meet the requirements for powering the *Sanyo Denki 103H7823-1740* stepper motor, which requires 4A per phase to be able to utilize the motors full strength. The *Pololu High-Power Stepper Motor Driver 36v4* is capable of delivering the necessary current.

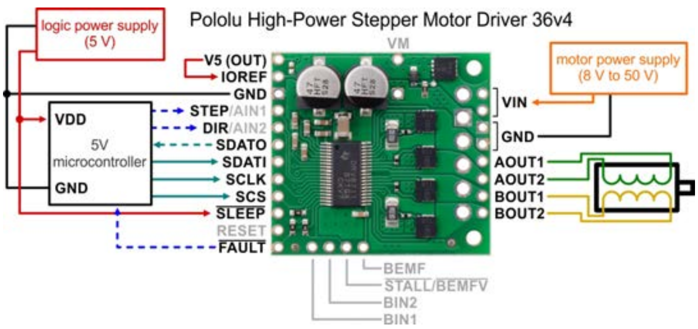


Figure 3.5. A Pololu High-Power Stepper Motor Driver 36v4 with a typical wiring diagram. *Image from Pololu's website https://www.pololu.com/product/3730*

As can be seen in Figure 3.5, the driver requires a 5V power supply, which can be provided from the arduino 5V connection, and a 8-50V power supply which will have to come from an external power source to power the stepper motor.

3.1.4 Power supply

To be able to operate the ABS, all components have to be supplied with power. The power has to come from mobile sources as to not limit the distance the bike can travel. Batteries are relatively cheap and easily accessible and were used as the power sources.

The *Sanyo Denki 103H7823-1740* stepper motor requires a voltage of 24VDC. In order to achieve this voltage, 16 1.5V AA batteries were connected together in series with two battery holders, each holding 8 batteries, which adds up to 24V. The MCU also has to be powered in order to perform the necessary calculations from the sensors and to control the motor. A 9V battery was used for this purpose.

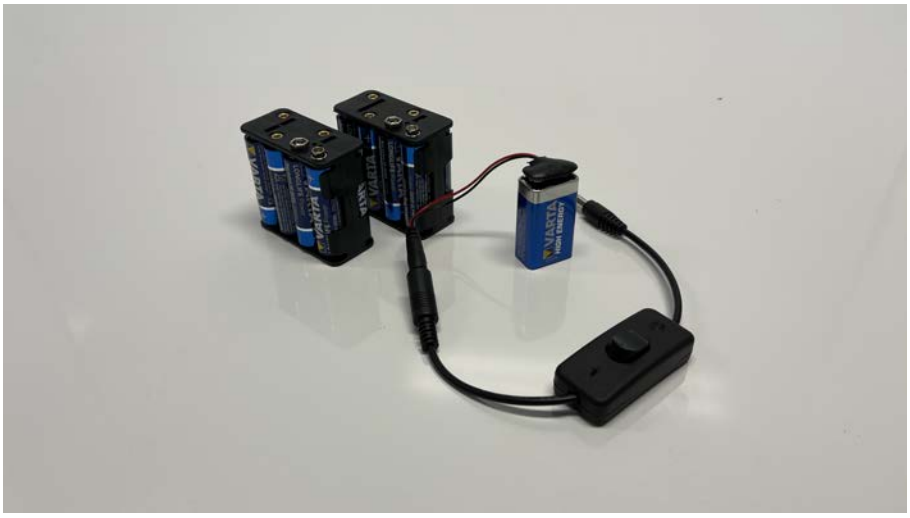


Figure 3.6. 16 AA batteries and a 9V battery with a switch. *Photo taken at KTH mechatronics lab.*

As can be seen in figure 3.6, a switch was connected to the 9V battery as a way to conserve the energy stored in the battery while it is not in use. A way to disconnect the 24V supply using a key switch was also implemented for the same reasons. This can be seen below in Figure 3.7.

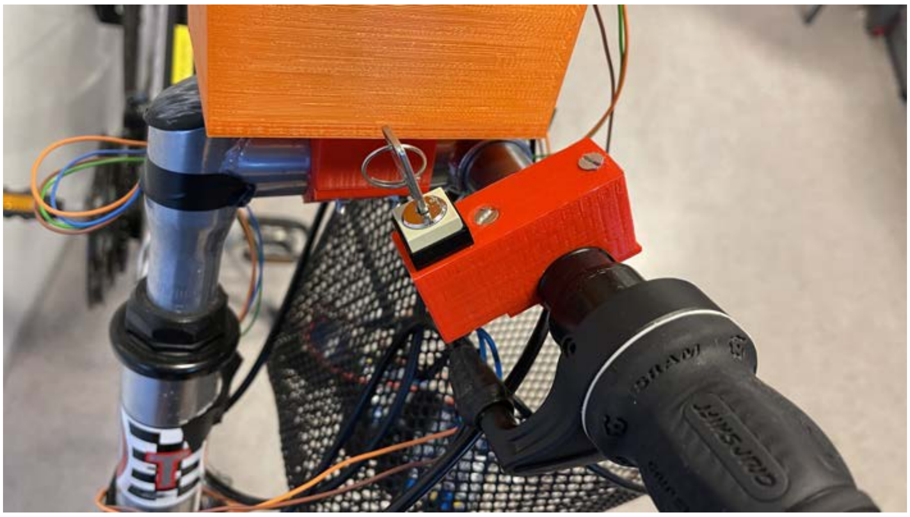


Figure 3.7. A key switch used to connect and disconnect the motors 24V power supply. *Photo taken at KTH mechatronics lab.*

3.1.5 Response time indicator

To measure the system response time a LED and speaker was used to indicate of the system detected a lockup. A photo of the indicator can be seen in Figure 3.8 below.

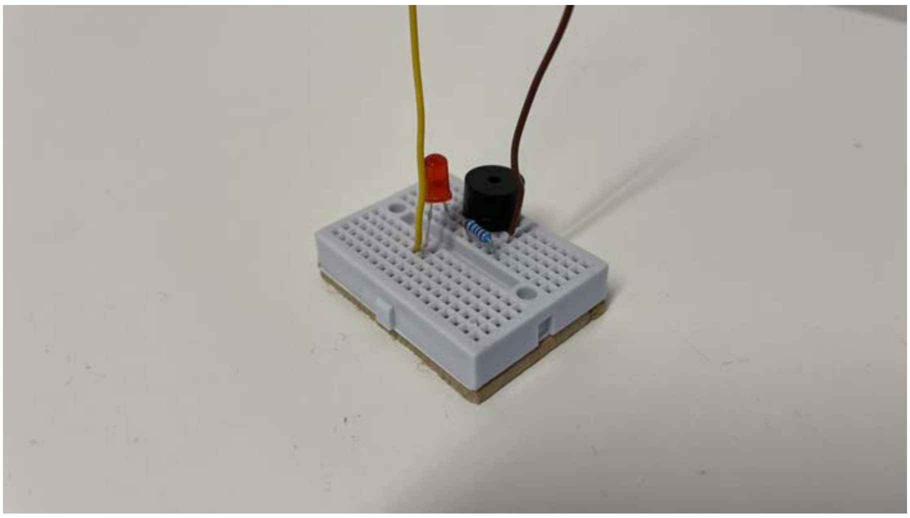


Figure 3.8. The LED and speaker used as an indicator. *Photo taken at KTH mechatronics lab.*

3.1.6 Component communication

To have a functioning system, all components have to be able to communicate with each other. Wires were soldered between all components on blank PCB boards using tin rods infused with flux. The connection diagram can be seen in figure 3.9.

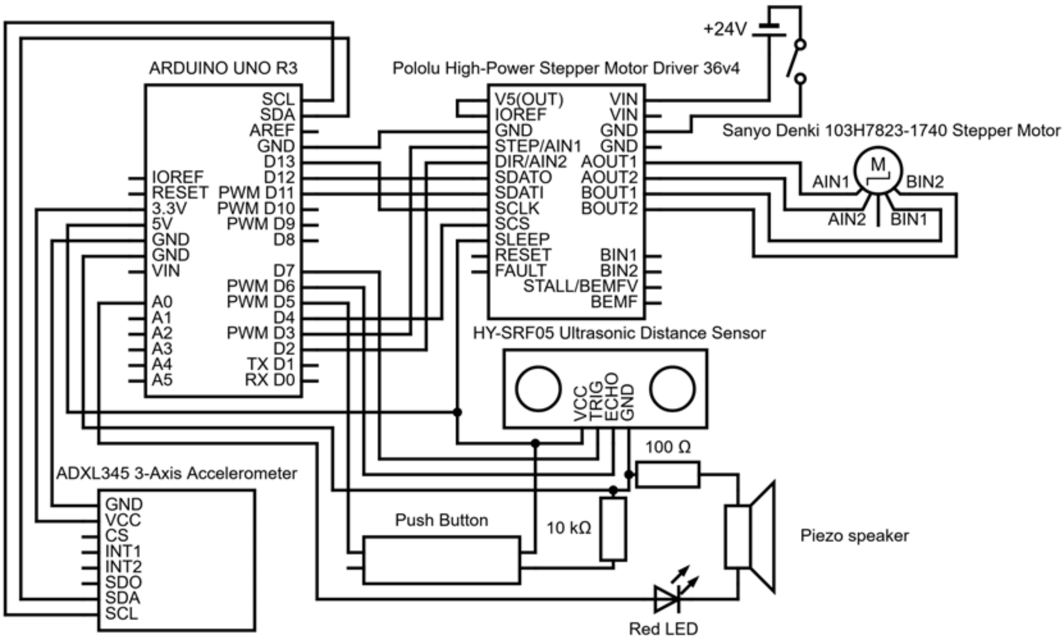


Figure 3.9. Connection diagram between all components and MCU. *Created with circuit-diagram.org.*

3.1.7 3D printed parts

In this project several different types of parts were 3D printed using a Ultimaker 3D printer. The material used to print was PLA filament. In the final result nine different parts were utilized. For the MCU and blank PCB a box and mount for the handlebar were printed. A mount for the key switch and a cylinder for the brake button were also printed and can be found on the handlebar. On the motor axle a 3D printed coupling with an end cap was designed to better hold the string when braking. To increase the motors braking torque, a steel rod was used to extend the braking lever to get a longer leverage arm. The steel rod was mounted on a square which was screwed on to the braking lever, this mount was 3D printed. By the rear wheel a mount for the US sensor was printed and a cylinder which was fitted to the transmitter were made. The disc mounted on the spokes of the rear wheel was also 3D printed. All of the CAD models of the printed parts can be found in Appendix C.

3.2 Software

The Arduino MCU can be programmed using the Arduino IDE in a version of the C++ programming language. The program can be uploaded to the MCU and will be stored until a new program has been uploaded and will restart every time the MCU is restarted, either by clicking the reset button on the board or plugged into power.

The program running on the MCU that controls the ABS logic can be found in appendix B and a flowchart of the logic can be seen in figure 3.10.

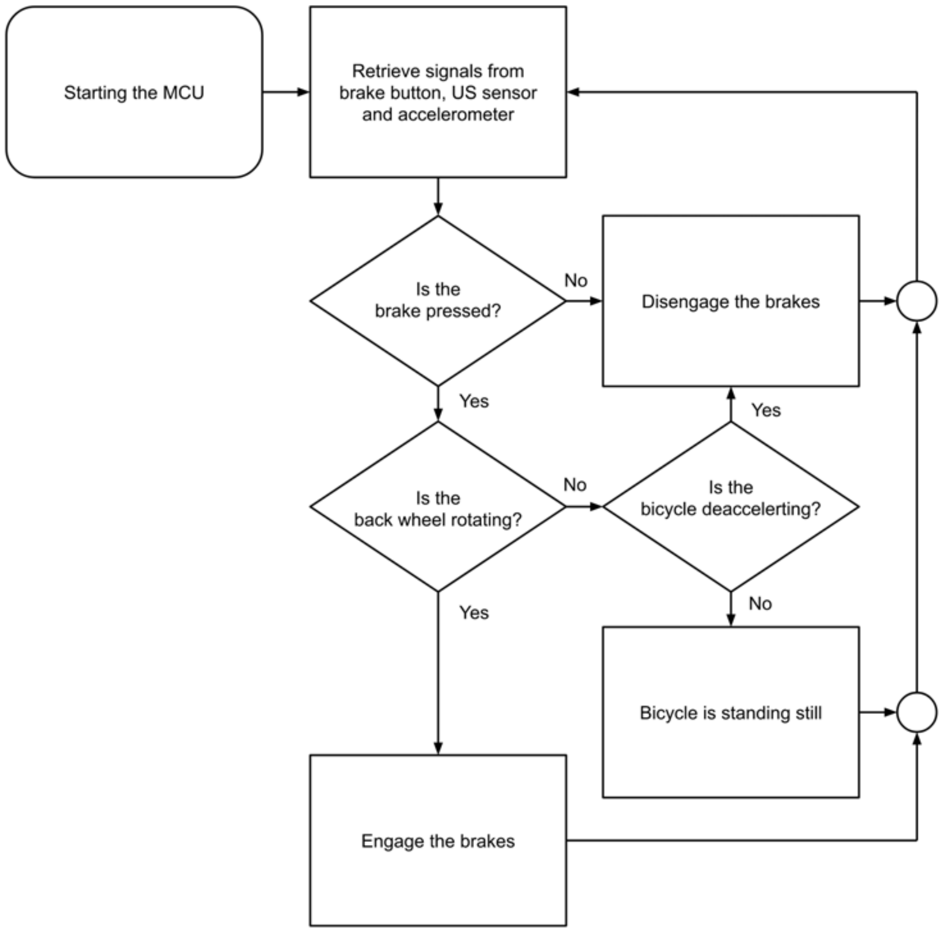


Figure 3.10. Flowchart of ABS logic. *Created using Google Slides.*

Chapter 4

Result

4.1 Braking distance

Unfortunately the functioning motor with highest torque that could be used for this project was not strong enough to lock either wheels. Thereby no braking distances can be compared and no data was collected for this test.

4.2 Response time

The results of the response time measurements can be seen in table 4.1 and Figure 4.1 below. The measurements were made using the method described in the method section of chapter 1.

Brake test#	Response time (s)
1	0,245
2	0,102
3	0,094
4	0,089
5	0,140
6	0,377
7	0,117
8	0,103
9	0,069
10	0,255
11	0,061
12	0,311
13	Did not activate
14	Did not activate
15	Did not activate
16	0,084
17	0,946
18	0,176
19	0,06
20	0,076
21	1,562
22	0,132
23	0,15
24	0,096
25	0,19
Average:	**0,247**

Table 4.1. Measured Response time

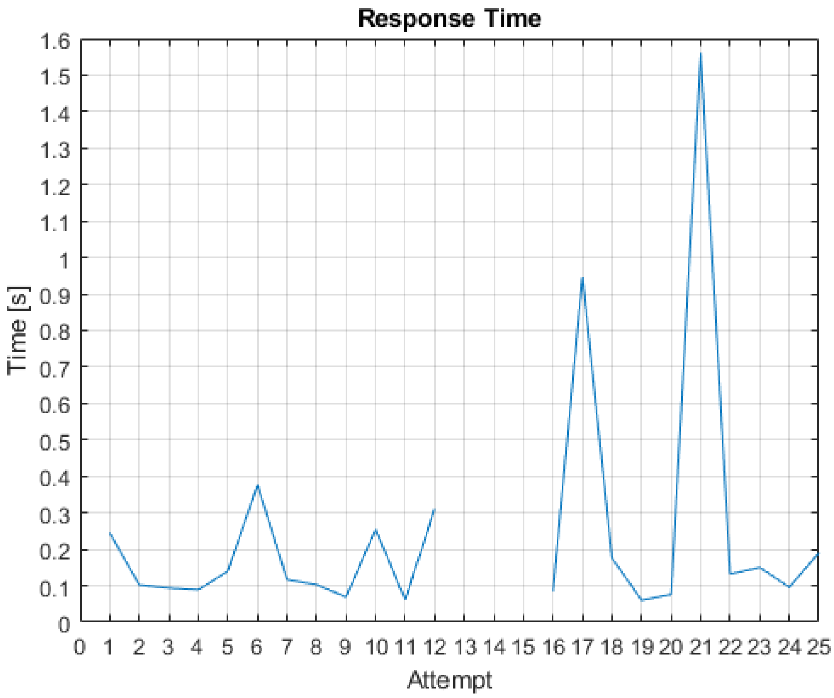


Figure 4.1. Graph showing the response time of the ABS brake release. *Created in Matlab.*

Chapter 5

Discussion

As seen in figure 4.1 the system does not have a steady response time and some times it did not activate at all. This can be because a number of reasons. One main reason might be that the update frequency of the disc is too low. There are only four holes in it. More holes would lead to a better resolution. There was suspicion that the US sensors sound wave would spread too wide and therefor give us false distances if more holes on the disc was used. The cylinder fitted on the US transmitter was a way to safeguard from wider signals. Another reason might be that the wheel locked up in an angle so that the disc would only just be in the sound waves path and as the bicycle decelerated, vibrations would make it seem as the disc was still spinning. There was uncertainty about how vibrations from the braking of the bicycle would affect the US sensor. But the result of an average response time of 247 ms is considered to be acceptable and tells us that the use of the US sensor was not a bad choice. However the fact that the system did not respond until around 1000 ms or not at all at times, is unacceptable and needs to be investigated further, as it could lead to a serious accident if the system would be used.

An infrared sensor with a low response time was first used in the project but was replaced by the US sensor at a later stage. This was because there were problems with interference and false signals to the IR receiver. This was thought to be because of the poor quality of the sensor. The whole budget was spent on the new stepper motor so they could not be replaced by other components other than those that could be found in the institutions storage.

Another option could possibly be a hall effect sensor and magnets embedded in a disc where each magnets polarity would be the opposite of its neighbors. The hall effect sensor would measure the changes in the magnetic field and store the changes similarly to the US or IR sensors. This type of sensor would not be prone to outside interference and could possibly have a better resolution than the other sensors depending on the amount of magnets.

Unfortunately the braking distance testing using the motor could not be performed. While the motor worked perfectly with the motor driver for more than a weeks time while the sensors were tested, the motor did not engage the day before testing. After spending a days worth of troubleshooting every connection and re-soldering all components, it was concluded that the driver did not work. The reason for the driver failing could not be determined. Another motor that could be operated with the much smaller *Pololu A4988* stepper driver was found in the institution storage, and tested it was as expected not even close to the strength of the motor that was bought as the new driver and motor only operated on 1A per phase. While not being as strong as the new motor, it was very close to working and many weeks were spent trying to make it work. This motor was actually used from the beginning of the project together with other sensors found at the institution, but as they were used in other projects prior to this one, they could be damaged or labels would be missing making it hard to find data sheets. Much time was spent trying to find data to the weaker motor, but only a sheet with 'sister' motors were found, one last try was done after the new motor and driver stopped working, but to no avail. Due to shipping times and that new purchases, as for example a new motor driver, had to be made out of pocket, attempting to measure any sort of braking distance was regrettably abandoned. Too much time was spent on trying to increase the braking force of the first motor.

Given that the global COVID-19 pandemic was still widespread in Sweden during the time of this project didn't help with ordering components or working in shared environments with tools, such as soldering irons, that weren't common household tools.

Chapter 6

Conclusion

The research question regarding the braking distance could not be answered, however, the reaction time could be tested and the results indicated that the prototype ABS system could react fast enough to a wheel locking up. while it was not possible to test a fully functioning ABS system, all the components except a motor driver worked together as intended. But for now the prototype system can only brake and release the bicycles brakes in theory. The project was also a success in the way as we learned what was needed to build an ABS system, and what was needed to make it respond fast enough.

Future Work

Redundancy is very important, and spending a bit more to get duplicate components will lead to less sitting around waiting for new ones if one were to break. As mentioned in chapter 5, other sensors than a US sensor should be tested to see if it could lead to a better response time. With a better sensor system, a more robust lockup detection system could be installed using control theory which could detect a lockup before it even happens and further lower the reaction time of the system[13].

Bibliography

[1] Volkswagen. (May-2021). innovation & technology word list - ABS. *https://www.volkswagen.se/sv/innovation-och-teknik/ordlista/a.html*

[2] Schyllander,J. Ekman,R. (2013). *Skadade cyklister - en studie av skadeutveckligen över tid.* MSB. *https://rib.msb.se/dok.aspx?Tab=2&dokid=27022*

[3] Bhasin,K.(2019) *A Review Paper on Anti-Lock Braking System (ABS) and its Future Scope* . HMR Institute of Technology Management, Hamidpur, New Delhi. Retrieved from *https://www.researchgate.net/publication/336667949_A_Review_Paper_on_Anti-Lock_Braking_System_ABS_and_its_Future_Scope*

[4] ENISZ,K. SZALAY,I. FODOR,D. NAGY,K. JAKAB,R. (2012) *Bicycle Anti-Lock Braking System Prototype Development.* Department of Automotive Mechatronics, University of Pannonia, Veszprém, Hungary. *http://www.acta.sapientia.ro/acta-emeng/C4/emeng4-3.pdf*

[5] Pettersson, O., Eriksson, A., Johansson, S., Norén, O. (2002). Bromsning av traktortåg: eftermontering av bromsar. Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:ri:diva-1986

[6] Sunström,N (2021-04-20). Friktionskoefficienter. *http://nilsun.se/articles/Friktionskoefficienter.pdf*

[7] Sheldon,B.(February-2021) *"Articles by Sheldon Brown and Others".* *https://www.sheldonbrown.com/canti-direct.html*

[8] Earl.B.(February-2021) *All about stepper motors* `https://cdn-learn.adafruit.com/downloads/pdf/all-about-stepper-motors.pdf`

[9] Pololu Robotics & Electronics. (2021-04-22). Pololu High-Power Stepper Motor Driver 36v4 product page *https://www.pololu.com/product/3730*

[10] Johansson, H. *Elektroteknik.* 2013 year edition. Stockholm, KTH Insti-tutionen för Maskinkonstruktion Mekatronik

[11] Grahn, E. (2017). Evaluation of MEMS accelerometer and gyroscope for orientation tracking nutrunner functionality (Dissertation). Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-215163

[12] Faltpihl, P. (2012). Ultrasonic sensing design and implementation for detecting and interacting with human beings in an AI system (Dissertation). Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-78429

[13] Glad,T. Ljung,L. (2006). *Reglerteknik, Grundläggande teori.* Studentlitteratur AB. Lund.

[14] Linder, A., Silvano, A. P., Sörensen, G., Pettersson, T. (2018). Stabilitet på cykel med och utan ABS: en pilotstudie. Retrieved from Statens väg- och transportforskningsinstitut website: http://urn.kb.se/resolve?urn=urn:nbn:se:vti:diva-14107

[15] Adler, G. A. (2017). Säkrare på två hjul, en studie för cykelsäkerhet (Dissertation). Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-210789

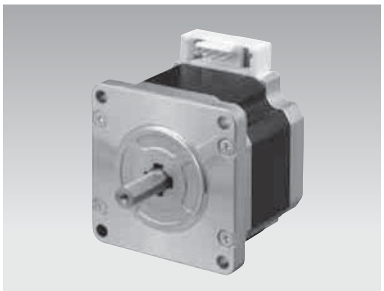
[16] Ming-chin Wu. Ming-chang Shih.(2013).*Simulated and experimental study of hydraulic anti-lock braking system using sliding-mode PWM control* Mechatronics,Volume 13, Issue 4,2003,Pages 331-351,ISSN 0957-4158. Retrieved from https://www.sciencedirect.com/science/article/pii/S0957415801000496

[17] Moberg, D., Munoz, M. (2017). Mekanisk ABS for cyklar (Dissertation). Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-210897

[18] Corno,M. D'Avico,L. Savaresi,S. (2018). *An Anti-lock Braking System for bicycles.* Retrieved from *https://www.researchgate.net/publication/328990313_An_Anti-Lock_Braking_System_for_Bicycles*

Appendix A

Stepper Motor Data sheet



60 mm sq. (2.36 inch sq.)

1.8° /step RoHS

Bipolar winding, Connector type
Bipolar winding, Lead wire type
Dimensions for attaching NEMA23 are interchangeable (47.14 mm-pitch)
Unipolar winding, Connector type ▶ p. 74
Unipolar winding, Lead wire type
Dimensions for attaching NEMA23 are interchangeable (47.14 mm-pitch) ▶ p. 74

Customizing

Hollow | Shaft modification | Decelerator | Encoder | Brake

Varies depending on the model number and quantity. Contact us for details.

Bipolar winding, Connector type

Model number		Holding torque at 2-phase energization	Rated current	Wiring resistance	Winding inductance	Rotor inertia	Mass (Weight)	Motor length (L)
Single shaft	Dual shaft	[N·m (oz·in) min.]	A/phase	Ω /phase	mH/phase	[×10⁻⁴kg·m² (oz·in²)]	[kg (lbs)]	mm (in)
103H7821-5740	**103H7821-5710**	0.88 (124.6)	2	1.27	3.3	0.275 (1.50)	0.6 (1.32)	44.8 (1.76)
103H7821-1740	**103H7821-1710**	0.88 (124.6)	4	0.35	0.8	0.275 (1.50)	0.6 (1.32)	44.8 (1.76)
103H7822-5740	**103H7822-5710**	1.37 (194.0)	2	1.55	5.5	0.4 (2.19)	0.77 (1.70)	53.8 (2.12)
103H7822-1740	**103H7822-1710**	1.37 (194.0)	4	0.43	1.38	0.4 (2.19)	0.77 (1.70)	53.8 (2.12)
103H7823-5740	**103H7823-5710**	2.7 (382.3)	2	2.4	9.5	0.84 (4.59)	1.34 (2.95)	85.8 (3.38)
103H7823-1740	**103H7823-1710**	2.7 (382.3)	4	0.65	2.4	0.84 (4.59)	1.34 (2.95)	85.8 (3.38)

Motor cable: Model No. 4837961-1

Bipolar winding, Lead wire type Dimensions for attaching NEMA23 are interchangeable (47.14 mm-pitch)

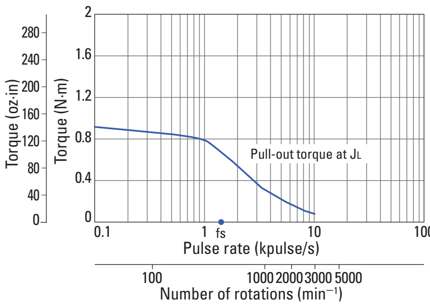
Model number		Holding torque at 2-phase energization	Rated current	Wiring resistance	Winding inductance	Rotor inertia	Mass (Weight)	Motor length (L)
Single shaft	Dual shaft	[N·m (oz·in) min.]	A/phase	Ω /phase	mH/phase	[×10⁻⁴kg·m² (oz·in²)]	[kg (lbs)]	mm (in)
103H7821-5760	**103H7821-5730**	0.88 (124.6)	2	1.27	3.3	0.275 (1.50)	0.6 (1.32)	43.5 (1.71)
103H7821-1760	**103H7821-1730**	0.88 (124.6)	4	0.35	0.8	0.275 (1.50)	0.6 (1.32)	43.5 (1.71)
103H7822-5760	**103H7822-5730**	1.37 (194.0)	2	1.55	5.5	0.4 (2.19)	0.77 (1.70)	52.5 (2.07)
103H7822-1760	**103H7822-1730**	1.37 (194.0)	4	0.43	1.38	0.4 (2.19)	0.77 (1.70)	52.5 (2.07)
103H7823-5760	**103H7823-5730**	2.7 (382.3)	2	2.4	9.5	0.84 (4.59)	1.34 (2.95)	84.5 (3.33)
103H7823-1760	**103H7823-1730**	2.7 (382.3)	4	0.65	2.4	0.84 (4.59)	1.34 (2.95)	84.5 (3.33)

Characteristics diagram

103H7821-5740
103H7821-5710

103H7821-5760
103H7821-5730

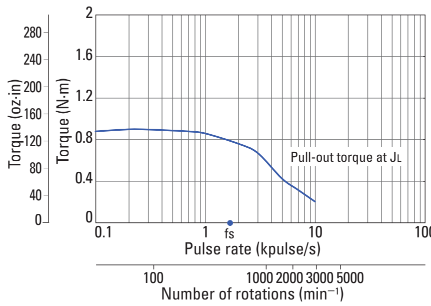
Constant current circuit
Source voltage: 24 VDC
Operating current:
2 A/phase, 2-phase energization (full-step)
J_L=[2.6 × 10⁻⁴kg·m² (14.22 oz·in²) use the rubber coupling]
fs: Maximum self-start frequency when not loaded



103H7821-1740
103H7821-1710

103H7821-1760
103H7821-1730

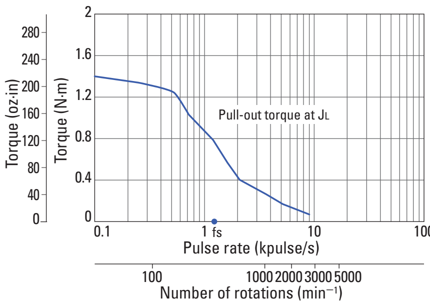
Constant current circuit
Source voltage: 24 VDC
Operating current:
4 A/phase, 2-phase energization (full-step)
J_L=[2.6 × 10⁻⁴kg·m² (14.22 oz·in²) use the rubber coupling]
fs: Maximum self-start frequency when not loaded



103H7822-5740
103H7822-5710

103H7822-5760
103H7822-5730

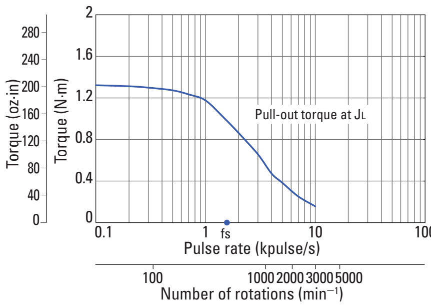
Constant current circuit
Source voltage: 24 VDC
Operating current:
2 A/phase, 2-phase energization (full-step)
J_L=[2.6 × 10⁻⁴kg·m² (14.22 oz·in²) use the rubber coupling]
fs: Maximum self-start frequency when not loaded



103H7822-1740
103H7822-1710

103H7822-1760
103H7822-1730

Constant current circuit
Source voltage: 24 VDC
Operating current:
4 A/phase, 2-phase energization (full-step)
J_L=[2.6 × 10⁻⁴kg·m² (14.22 oz·in²) use the rubber coupling]
fs: Maximum self-start frequency when not loaded



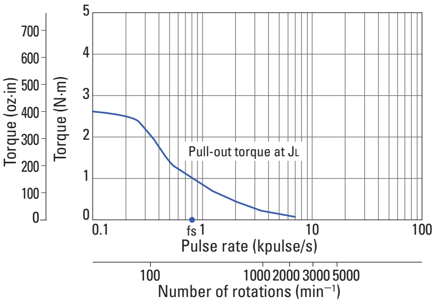
Data is measured under the trial conditions of SANYO DENKI. Driving torque may vary according to actual machine precision.

Characteristics diagram

103H7823-5740
103H7823-5710

103H7823-5760
103H7823-5730

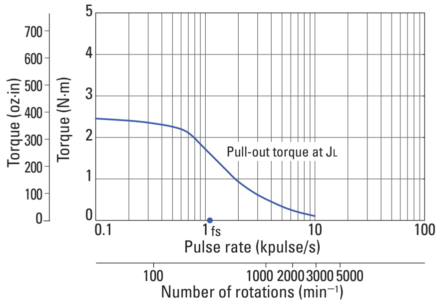
Constant current circuit
Source voltage: 24 VDC
Operating current:
2 A/phase, 2-phase energization (full-step)
J_L=[7.4 × 10^{-4}kg·m² (40.46 oz·in²) use the rubber coupling]
fs: Maximum self-start frequency when not loaded



103H7823-1740
103H7823-1710

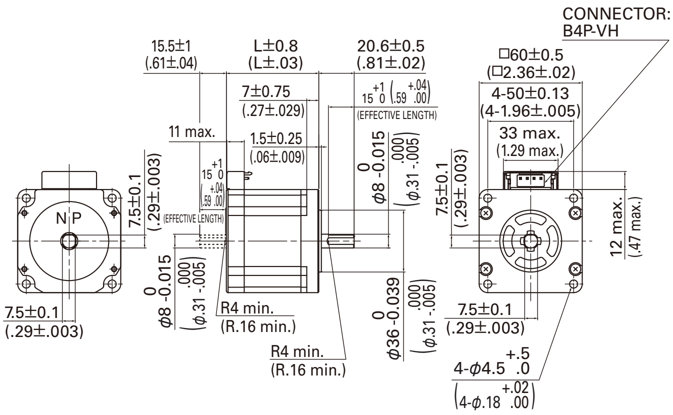
103H7823-1760
103H7823-1730

Constant current circuit
Source voltage: 24 VDC
Operating current:
4 A/phase, 2-phase energization (full-step)
J_L=[7.4 × 10^{-4}kg·m² (40.46 oz·in²) use the rubber coupling]
fs: Maximum self-start frequency when not loaded

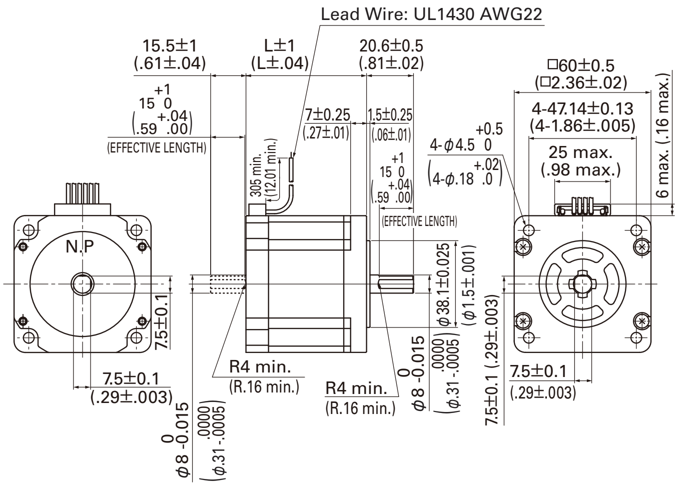


Dimensions [Unit: mm (inch)]

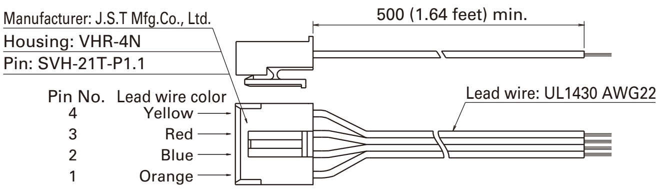
Connector type



Lead wire type

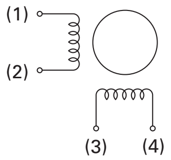


Motor cable Bipolar Model number: 4837961-1

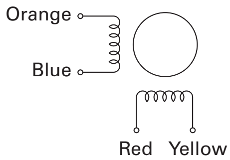


Internal wiring

Connector type
( ) connector pin number, terminal block number



Lead wire type



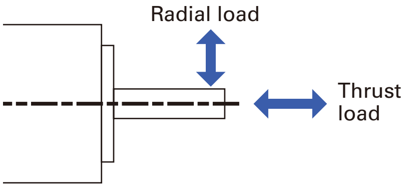
Compatible drivers

- For motor model number 103H782 □ -17 □ 0 (4 A/phase)
  Driver is not included.
  If you require assistance finding a driver, contact us for details.
- For motors not listed above (2 A/phase)
  Model number: BS1D200P10 (DC input)
  Operating current select switch setting: 0

Allowable Load, Rotation Direction ▶p. 87 General Specifications ▶p. 88

Data is measured under the trial conditions of SANYO DENKI. Driving torque may vary according to actual machine precision.

Allowable Radial/Thrust Load



Flange size	Model number	Distance from end of shaft : mm (in) 0	5	10	15	Thrust load N (lbs)
		Radial load : N (lbs)				
14 mm sq. (0.55 in sq.)	SH2141	10 (2.25)	11 (2.47)	13 (2.92)	-	0.7 (0.16)
28 mm sq. (1.10 in sq.)	SH228☐	42 (9)	48 (10)	56 (12)	66 (14)	3 (0.67)
35 mm sq. (1.38 in sq.)	SH353☐	40 (8)	50 (11)	67 (15)	98 (22)	10 (2.25)
42 mm sq. (1.65 in sq.)	103H52☐☐ SH142☐	22 (4)	26 (5)	33 (7)	46 (10)	10 (2.25)
50 mm sq. (1.97 in sq.)	103H670☐	71 (15)	87 (19)	115 (25)	167 (37)	15 (3.37)
56 mm sq. (2.20 in sq.)	103H712☐	52 (11)	65 (14)	85 (19)	123 (27)	15 (3.37)
	103H7128	85 (19)	105 (23)	138 (31)	200 (44)	15 (3.37)
60 mm sq. (2.36 in sq.)	103H782☐ SH160☐	70 (15)	87 (19)	114 (25)	165 (37)	20 (4.50) 15 (3.37)
86 mm sq. (3.39 in sq.)	SM286☐ SH286☐	167 (37)	193 (43)	229 (51)	280 (62)	60 (13.488)
86 mm sq. (3.39 in sq.)	103H822☐	191 (43)	234 (53)	301 (68)	421 (95)	60 (13.488)
φ106 mm (φ4.17 in)	103H8922☐	321 (72)	356 (79)	401 (90)	457 (101)	100 (22.48)

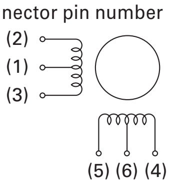
Internal Wiring and Rotation Direction

Unipolar winding

Connector type Model number: 103H52☐☐

Internal wire connection

( ) connector pin number



Direction of motor rotation

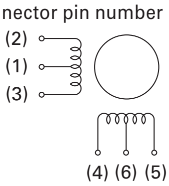
When excited by a direct current in the order shown below, the direction of rotation is clockwise as viewed from the output shaft side.

		Connector pin number (1.6)	(5)	(3)	(4)	(2)
Exciting order	1	+	−	−		
	2	+		−	−	
	3	+			−	−
	4	+	−			−

Connector type Model number: 103H782☐☐

Internal wire connection

( ) connector pin number



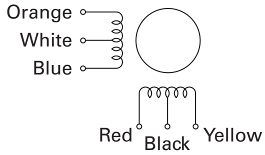
Direction of motor rotation

When excited by a direct current in the order shown below, the direction of rotation is clockwise as viewed from the output shaft side.

		Connector pin number (1.6)	(4)	(3)	(5)	(2)
Exciting order	1	+	−	−		
	2	+		−	−	
	3	+			−	−
	4	+	−			−

Lead wire type

Internal wire connection



Direction of motor rotation

When excited by a direct current in the order shown below, the direction of rotation is clockwise as viewed from the output shaft side.

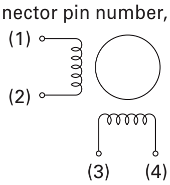
		Lead wire color White & black	Red	Blue	Yellow	Orange
Exciting order	1	+	−	−		
	2	+		−	−	
	3	+			−	−
	4	+	−			−

Bipolar winding

Connector type

Internal wire connection

( ) connector pin number, terminal block number



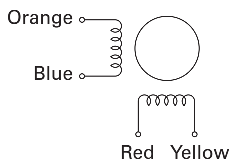
Direction of motor rotation

When excited by a direct current in the order shown below, the direction of rotation is clockwise as viewed from the output shaft side.

		Connector pin number, terminal block number (3)	(2)	(4)	(1)
Exciting order	1	−	−	+	+
	2	+	−	−	+
	3	+	+	−	−
	4	−	+	+	−

Lead wire type

Internal wire connection



Direction of motor rotation

When excited by a direct current in the order shown below, the direction of rotation is clockwise as viewed from the output shaft side.

		Lead wire color Red	Blue	Yellow	Orange
Exciting order	1	−	−	+	+
	2	+	−	−	+
	3	+	+	−	−
	4	−	+	+	−

General Specifications

Motor model number	SH2141	SH228□	SH353□	SS242□	SH142□	103H52□□	SS250□	103H67□□	103H712□
Type	—								
Operating ambient temperature	−10℃ to +50℃								
Conversation temperature	−20℃ to +65℃								
Operating ambient humidity	20 to 90% RH (no condensation)								
Conversation humidity	5 to 95% RH (no condensation)								
Operation altitude	1000 m (3281 feet) max. above sea level								
Vibration resistance	Vibration frequency 10 to 500 Hz, total amplitude 1.52 mm (10 to 70 Hz), vibration acceleration 150 m/s² (70 to 500 Hz), sweep time 15 min/cycle, 12 sweeps in each X, Y and Z direction.								
Impact resistance	500 m/s² of acceleration for 11 ms with half-sine wave applying three times for X, Y, and Z axes each, 18 times in total.								
Insulation class	Class B (+130℃)								
Withstandable voltage	At normal temperature and humidity, no failure with 500 VAC @50/60 Hz applied for one minute between motor winding and frame.							At normal temperature and humidity, no failure with 1000 VAC @50/60 Hz applied for one minute between motor winding and frame.	
Insulation resistance	At normal temperature and humidity, not less than 100 MΩ between winding and frame by 500 VDC megger.								
Protection grade	IP40								
Winding temperature rise	80 K max. (Based on Sanyo Denki standard)								
Static angle error	±0.09°				±0.054°	±0.09°			
Thrust play *1	0.075 mm (0.003 in) max. (load: 0.35 N (0.08 lbs))	0.075 mm (0.003 in) max. (load: 1.5 N (0.34 lbs))	0.075 mm (0.003 in) max. (load: 5 N (1.12 lbs))	0.075 mm (0.003 in) max. (load: 4 N (0.9 lbs))	0.075 mm (0.003 in) max. (load: 5 N (1.12 lbs))	0.075 mm (0.003 in) (load: 5 N (1.12 lbs))	0.075 mm (0.003 in) max. (load: 4 N (0.9 lbs))	0.075 mm (0.003 in) (load: 10 N (2.25 lbs))	0.075 mm (0.003 in) (load: 10 N (2.25 lbs))
Radial play *2	0.025 mm (0.001 in) max. (load: 5 N (1.12 lbs))								
Shaft runout	0.025 mm (0.001 in)								
Concentricity of mounting pilot relative to shaft	ϕ 0.05 mm (ϕ 0.002 in)	ϕ 0.05 mm (ϕ 0.002 in)	ϕ 0.075 mm (ϕ 0.003 in)	ϕ 0.075 mm (ϕ 0.003 in)	ϕ 0.05 mm (ϕ 0.002 in)	ϕ 0.05 mm (ϕ 0.002 in)	ϕ 0.075 mm (ϕ 0.003 in)	ϕ 0.075 mm (ϕ 0.003 in)	ϕ 0.075 mm (ϕ 0.003 in)
Squareness of mounting surface relative to shaft	0.1 mm (0.004 in)	0.1 mm (0.004 in)	0.1 mm (0.004 in)	0.1 mm (0.004 in)	0.1 mm (0.004 in)	0.1 mm (0.004 in)	0.1 mm (0.004 in)	0.075 mm (0.003 in)	0.075 mm (0.003 in)
Direction of motor mounting	Can be freely mounted vertically or horizontally								

Motor model number	SH160□	103H78□□	SH286□	103H8922□	SM286□	103H712□-6□□0 CE Model	103H822□-6□□0 CE Model	103H8922□-63□1 CE Model
Type	—				S1 (continuous operation)			
Operating ambient temperature	−10℃ to +50℃				−10℃ to +40℃			
Conversation temperature	−20℃ to +65℃				−20℃ to +60℃			
Operating ambient humidity	20 to 90% RH (no condensation)				95% max.: 40℃ max., 57% max.: 50℃ max., 35% max.: 60℃ max. (no condensation)			
Conversation humidity	5 to 95% RH (no condensation)							
Operation altitude	1000 m (3280 feet) max. above sea level							
Vibration resistance	Vibration frequency 10 to 500 Hz, total amplitude 1.52 mm (10 to 70 Hz), vibration acceleration 150 m/s² (70 to 500 Hz), sweep time 15 min/cycle, 12 sweeps in each X, Y and Z direction.							
Impact resistance	500 m/s² of acceleration for 11 ms with half-sine wave applying three times for X, Y and Z axes each, 18 times in total.							
Insulation class	Class B (+130℃)				Class F (+155℃)	Class B (+130℃)		
Withstandable voltage	At normal temperature and humidity, no failure with 1000 VAC @50/60 Hz applied for one minute between motor winding and frame.			At normal temperature and humidity, no failure with 1500 VAC @50/60 Hz applied for one minute between motor winding and frame.				
Insulation resistance	At normal temperature and humidity, not less then 100 MΩ between winding and frame by 500 VDC megger.							
Protection grade	IP40				IP43			
Winding temperature rise	80 K max. (Based on Sanyo Denki standard)							
Static angle error	±0.054°	±0.09°						
Thrust play *1	0.075 mm (0.003 in) max. (load: 10 N (2.25 lbs))							
Radial play *2	0.025 mm (0.001 in) (load: 5 N (1.12 lbs))	0.025 mm (0.001 in) (load: 5 N (1.12 lbs))	0.025 mm (0.001 in) (load: 5 N (1.12 lbs))	0.025 mm (0.001 in) (load: 10 N (2.25 lbs))	0.025 mm (0.001 in) (load: 5 N (1.12 lbs))	0.025 mm (0.001 in) (load: 5 N (1.12 lbs))	0.025 mm (0.001 in) (load: 5 N (1.12 lbs))	0.025 mm (0.001 in) (load: 10 N (2.25 lbs))
Shaft runout	0.025 mm (0.001 in)							
Concentricity of mounting pilot relative to shaft	ϕ 0.075 mm (ϕ 0.003 in)							
Squareness of mounting surface relative to shaft	0.1 mm (0.004 in)	0.075 mm (0.003 in)	0.15 mm (0.006 in)	0.1 mm (0.004 in)	0.15 mm (0.006 in)	0.075 mm (0.003 in)	0.1 mm (0.004 in)	0.1 mm (0.004 in)
Direction of motor mounting	Can be freely mounted vertically or horizontally							

*1 Thrust play: Shaft displacement under axial load.
*2 Radial play: Shaft displacement under radial load applied 1/3rd of the length from the end of the shaft.

Safety standards

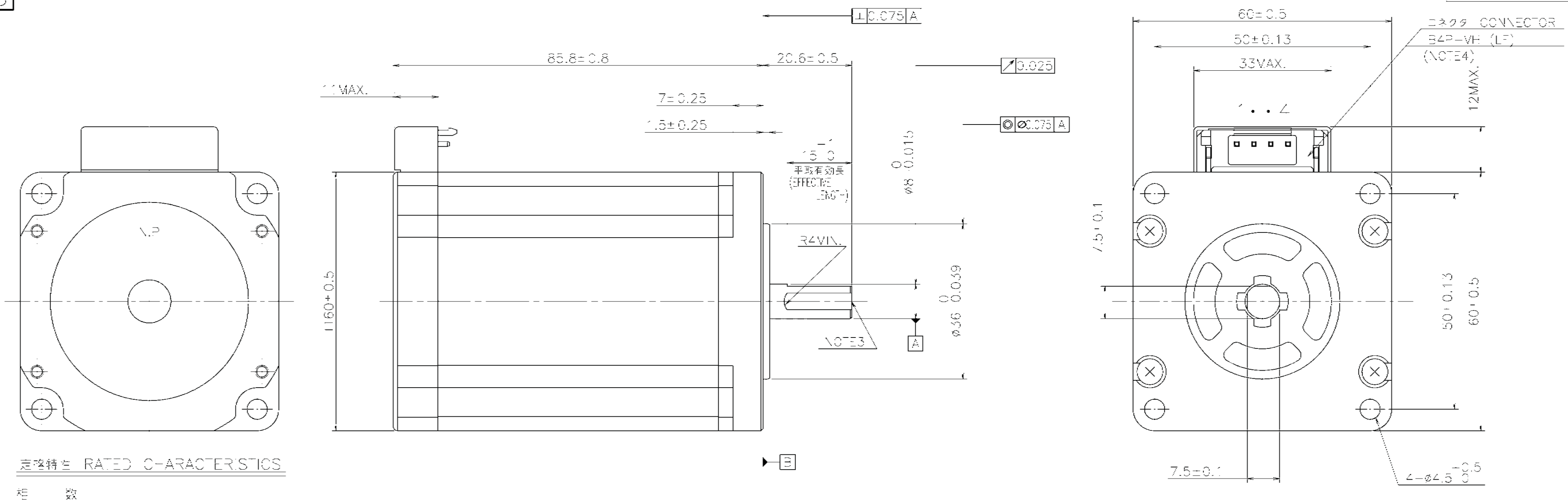
Model Number: SM286□ CE/UL marked models

CE (TÜV)	Standard category		Applicable standard
	Low-voltage directives		EN60034-1, EN60034-5
UL	Acquired standards	Applicable standard	File No.
	UL	UL1004-1, UL1004-6	E179832
	UL for Canada	CSA C22.2 No.100	

Model Number: 103H712□-6□□0, 103H822□-6□□0, 103H8922□-63□1 CE marked model

CE (TÜV)	Standard category	Applicable standard
	Low-voltage directives	EN60034-1, EN60034-5

RoHS ③



定格特性 RATED CHARACTERISTICS

	相数 PHASES	2	
	基本ステップ角 FUNDAMENTAL STEP ANGLE	1.8	°
	定格電圧 VOLTS	2.6	V (DC)
	定格電流 AMPS	4	A/phase
	巻線抵抗 WINDING RESISTANCE	0.65	Ω±10% at 25℃
	巻線インダクタンス WINDING INDUCTANCE	2.4	mH±20% at 1 kHz, 1 V(r.m.s)
	ホールディングトルク HOLDING TORQUE	2.7	N·m MIN. at I=4 A/phase 2EX.
注1. NOTE 1.	脱出トルク PULL OUT TORQUE	2	N·m MIN. at 200 pulse/s 負荷イナーシャ INERTIAL LOAD 7.4×10^-4 kg·m^2 (ラバーカップリングイナーシャを含む) (INERTIA OF RUBBER COUPLING IS INCLUDED)
注1. NOTE 1.	最大自起動周波数 MAX. STARTING RATE	1100	pulse/s MIN. at NO LOAD
注1. NOTE 1.	最大連続応答周波数 MAX. SLEWING RATE	3700	pulse/s MIN. at NO LOAD
	静止角度誤差 POSITIONAL ACCURACY	±0.054	°(0.108° SPREAD MAX.) 2EX.
注2. NOTE 2.	温度上昇値 COIL TEMPERATURE RISE	80	K MAX.
	ロータイナーシャ ROTOR INERTIA	0.84×10^-4	kg·m^2 NOMINAL
	絶縁階級 INSULATION CLASS	B	
③	許容スラスト荷重 ALLOWABLE THRUST LOAD	20	N
③	許容ラジアル荷重 ALLOWABLE RADIAL LOAD	71	N 軸先端荷重 LOAD TO SHAFT END.

注1.ドライバー：PMM-CS-803 E=100VAC, I=4 A/相, 2相励磁。
NOTE) DRIVER: PMM-CS-803 E=100VAC, I=4 A/PHASE, 2EX.

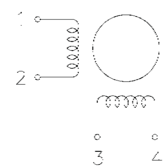
2.160×160×6t アルミ放熱板に取付け、2相励磁 I=4 A/相を連続通電し、抵抗法にて測定した時の値。
MOUNTED A MOTOR ON 160X160X6t ALUMINUM HEAT SINK AND CONTINUOUSLY ENERGIZED THE COIL AT 2 PHASE I=4 A/PHASE.
MEASURED BY THE CHANGE OF RESISTANCE METHOD.

3.シャフトセンター穴の有無及び形状は、製造上の都合により任意とする。
CENTER HOLE ON THE SHAFT END IS NOT ALWAYS MADE.

4.適合ハウジング及びコンタクト（例）：VHR-4N, SVH-21T-P1.1（日本圧着端子）
MATING HOUSING AND CONTACT (e.g.): VHR-4N, SVH-21T-P1.1 (JST)

5.適合ハウジング及びコンタクトはユーザー様でご用意してください。
PLEASE SUPPLY MATING HOUSING AND CONTACTS BY THE USER-SELF.

内部結線・CONNECTION
（ピン番号）(PIN NO.)



回転方向・DIRECTION OF ROTATION

下記の順に通電励磁した場合、回転方向は面B側より見て時計方向に回転のこと。

WHEN A MOTOR IS SEQUENCED AS SHOWN IN THE TABLE BELOW, THE SHAFT ROTATION MUST BE CLOCKWISE WHEN YOU SEE FROM SURFACE "B" SIDE.

励磁順序 ENERGIZE ORDER	コネクタピン番号 CONNECTOR PIN NO. 3	2	4	1
1	⊖	⊖	⊖	⊖
2	⊖	⊖	⊖	⊖
3	⊖	⊕	⊖	⊖
4	⊖	⊕	⊖	⊖

③和英併記化・Change to Japanese-English drawing.

C	E0085439	07-07-02	承認 APPROVED BY 07-07-06
B	E0082010	07-01-06	検査 CHECKED BY 07-07-06
A	NEW DESIGN	02-09-17	設計 DESIGNED BY KUCHIYAMA 07-07-02
記号	記事 DESCRIPTION	年月日 DATE	

名称 TITLE: STEPPING MOTOR ステッピングモータ

山洋電気株式会社 SANYO DENKI CO.,LTD.

図番 DWG NO.: 103-7823-1740 C

Appendix B

Arduino Code

```
// --------------------------------------------------------------------------
// Program:
// Anti-lock braking system for bicycles
//
// Description:
// This program operates a motor to either brake or release
// the brakes depending on if the back wheel is locked or not,
// which is determined by the processing signals
// gathered from the connected sensors. The code was built
// around each components example code from each respective
// library used.
//
// Names:
// - Kristian Jandric
// - Lucas Andersson
//
// Date:
// 2021-05-09
//
// Hardware:
// - Arduino UNO R3
// - Sanyo Denki 103H7823-174 Stepper Motor
// - Pololu High-Power Stepper Motor Driver 36v
// - ADXL345 Accelerometer
// - HY-SRF05 Ultrasonic Distance Sensor
// - Push Button
// - Piezo Speaker
// - Red LED
// --------------------------------------------------------------------------

#include <SPI.h>
#include <HighPowerStepperDriver.h>
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_ADXL345_U.h>

Adafruit_ADXL345_Unified accel = Adafruit_ADXL345_Unified();
HighPowerStepperDriver sd;
```

```
// Pins
const uint8_t dirPin = 2;  // Direction
const uint8_t stepPin = 3; // Step
const uint8_t CSPin = 4;   //CS

const int TRIG_PIN = 6; // Output US sensor
const int ECHO_PIN = 7; // Input US sensor
const int button = 5;

// Other constants
const int usBufferAmount = 50;
const int accBufferAmount = 10;
const int STEPS_PER_REV = 200;   // Default 200
const int DELAY_PER_STEP = 1200; // Default 2000
const uint16_t StepPeriodUs = 200;

// Variables
int freq = 700;
bool braking = false;
bool motorDir = true;
long usThreshold = 10;
double accDiff = 2;
bool usBuffer[usBufferAmount];
double accBuffer[accBufferAmount];

// Returns a boolean value depending on if the wheel is
// locked up or not.
bool lockedUp()
{

    bool isLocked = true;
    // checking ultrasonic distance sensor
    for (int i = 1; i < usBufferAmount; i++)
    {
        if (usBuffer[i] != usBuffer[0])
        {

            isLocked = false;
            Serial.println("US ");
            break;
        }
    }

    // Checking accelerometer
    if (isLocked)
    {
        freq = 2000;
        for (int i = 0; i < accBufferAmount-1; i++)
        {
            double diff = accBuffer[i] - accBuffer[i+1];
            if (abs(diff) < accDiff)
            {
                isLocked = false;
```

```
                Serial.println("acc");
                break;
            }
        }
    }
    else
    {
        freq = 700;
    }

    if (isLocked)
    {
        digitalWrite(LED_BUILTIN, HIGH);
    }
    else
    {
        digitalWrite(LED_BUILTIN, LOW);
    }
    // Serial.println(isLocked);
    return isLocked;
}

// Retrieves and stores the signals from the sensors
void retrieveSignals()
{

    double acc = getAcceleration();
    long usDist = getUltraDistance();

    // Serial.println(acc);
    // Serial.println(usDist);

    for (int i = usBufferAmount - 1; i > 0; i--)
    {
        usBuffer[i] = usBuffer[i - 1];
    }
    for (int i = accBufferAmount - 1; i > 0; i--)
    {
        accBuffer[i] = accBuffer[i - 1];
    }
    usBuffer[0] = usDist < usThreshold ? true : false;
    accBuffer[0] = acc;
}

// Prints a speciefied buffer where the signal data is stored
// Used as a debug function
void printBuffer(int buf)
{
    if (buf == 1) {

        Serial.print("us");
        for (int i = 0; i < usBufferAmount; i++)
        {

```

```
            Serial.print(usBuffer[i]);
        }
    }
    else if (buf == 0) {

        Serial.print("acc");
        for (int i = 0; i < accBufferAmount; i++)
        {
            Serial.print(accBuffer[i]);
            Serial.print(" ");
        }
    }
    else if (buf == 2)
    {
         for (int i = 0; i < accBufferAmount-1; i++)
        {
            double diff = accBuffer[i] - accBuffer[i+1];
            if (abs(diff) < accDiff )
            {
                Serial.print("0");
                Serial.print(" ");
            }
            else
            {
                Serial.print("1");
                Serial.print(" ");
            }
        }
    }
    else if (buf == 3)
    {
        Serial.print(lockedUp());

    }
    Serial.println("");
}

// Returns if the brake button is pressed or not
bool brakePressed()
{

    if (digitalRead(button) == LOW)
    {

        //digitalWrite(LED_BUILTIN, LOW);
        return false;
    }
    else
    {

        //digitalWrite(LED_BUILTIN, HIGH);
        return true;
    }
}
```

```
// Engages the brakes by powering the motor
void brake()
{

    braking = true;
    sd.setDirection(motorDir);

    for (int x = 0; x < 2 * STEPS_PER_REV; x++)
    {
        if (brakePressed() && !lockedUp())
        {
            step();
            delayMicroseconds(StepPeriodUs);
        }
        else
        {
            break;
        }
    }
}

// Disengages the brakes by powering the motor
void release()
{

    braking = false;
    sd.setDirection(!motorDir);

    for (int x = 0; x < STEPS_PER_REV; x++) {
        if (brakePressed() && lockedUp())
        {
            step();
            delayMicroseconds(StepPeriodUs);
        }
        else
        {
            break;
        }
    }
}

// Help function to get the motor to step
void step()
{
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(3);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(3);
}

// Retrieves the signals sent from the Ultrasonic
// Distance sensor
long getUltraDistance()
```

```
{
    long duration, distanceCm;

    digitalWrite(TRIG_PIN, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIG_PIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG_PIN, LOW);
    duration = pulseIn(ECHO_PIN, HIGH);

    // convert the time into a distance
    distanceCm = duration / 29.1 / 2;

    return distanceCm;
}

// Retrieves the signals sent from the Accelerometer
double getAcceleration()
{
    sensors_event_t event;
    accel.getEvent(&event);
    /*Serial.print("X: ");
    Serial.print(event.acceleration.x);
    Serial.print("  ");
    */
    return event.acceleration.x;
}

void setup()
{
    if(!accel.begin())
   {
      //Serial.println("No ADXL345 sensor detected.");
      while(1);
   }

    pinMode(TRIG_PIN, OUTPUT);
    pinMode(ECHO_PIN, INPUT);
    pinMode(button, INPUT);
    pinMode(LED_BUILTIN, OUTPUT);
    pinMode(A0, OUTPUT);

    Serial.begin(9600);


    SPI.begin();
    sd.setChipSelectPin(CSPin);

    // Drive the STEP and DIR pins low initially.
    pinMode(stepPin, OUTPUT);
    digitalWrite(stepPin, LOW);
    pinMode(dirPin, OUTPUT);
    digitalWrite(dirPin, LOW);

```

```
    // Give the driver some time to power up.
    delay(5);

    // Reset the driver to its default settings and clear latched status
    // conditions.
    sd.resetSettings();
    sd.clearStatus();

    // Select auto mixed decay.  TI's DRV8711 documentation recommends this mode
    // for most applications, and we find that it usually works well.
    sd.setDecayMode(HPSDDecayMode::Slow);

    // Set the current limit. You should change the number here to an appropriate
    // value for your particular system.
    sd.setCurrentMilliamps36v4(4000); //  VERSTIG  INTE 4000 mA

    // Set the number of microsteps that correspond to one full step.
    sd.setStepMode(HPSDStepMode::MicroStep8);

    // Enable the motor outputs.
    sd.enableDriver();
}

// Main logic
void loop()
{

    retrieveSignals();
    if (brakePressed())
    {

        if (lockedUp() && braking)
        {

            release();
        }
        else if (!braking)
        {
            brake();
        }
    }
    else if (braking)
    {

        release();
    }


    // Enables and disables the response test module
    if(!lockedUp() && brakePressed())
    {
```

```
            tone(A0, freq);
        }
        else
        {
            noTone(A0);
        }
        // printBuffer(1);
}

}
```

Acumen Code

```
// Bicycle simulation
// Created by: Kristian Jandric & Lucas Andersson
// Date: 2021-03-27
// Basic simulation of bicycle braking coded with Accumen.


model Main(simulator) =
initially

//Create frame of bicycle
bike = create frame((0,2,0),red),
//wheel_back = create wheel((0,0,0),red),
//wheel_front = create wheel((0,0.45*cos(pi/4) + 0.6 + 0.155,0),blue),


x1=0, x1'=10, x1''=0,

z = 0
//always
  //x1 = 0,
  //x2 = 0,
  //x3 = 0.45*cos(pi/4) + 0.6 + 0.155,
   //base.pos = (0,x1,0),
   //wheel_back.pos = (0,x2,0),
   //wheel_front.pos = (0,x3,0),

 //z = x1-x2


 always
 if x1<10
  then x1'' = -(x1')
  else if x1'>0
         then x1'' = -1
         else x1'' = 0,
 bike.pos = (0,x1,0),
 z = x1



// Model the frame and wheels of the bicycle
model frame (pos,col) =
initially
```

```
_3D = (), _Plot=()
always
_3D = (
        //Forward wheel
        Cylinder    center = pos+(0,0.45*cos(pi/4) + 0.6 + 0.155,0)
//L1
                    color = 0.4*col+ 0.6*white
                    length= 0.01
                    radius= 0.2
                    rotation=(0,0,pi/2)
            //Rear wheel
        Cylinder    center = pos+(0,0,0)    //L1
                    color = 0.4*col+ 0.6*white
                    length= 0.01
                    radius= 0.2
                    rotation=(0,0,pi/2)

         // Frame
        Cylinder    center = pos+(0,0.45*cos(pi/4)/2,0.45*sin(pi/4)/2)
//L1
                    color = 0.4*col+ 0.6*white
                    length= 0.45
                    radius= 0.01
                    rotation=(pi/4,0,0)

        Cylinder    center = pos+(0,0.45*cos(pi/4)/2,0)    //L2
                    color = 0.4*col+ 0.6*white
                    length= 0.45*cos(pi/4)
                    radius= 0.01
                    rotation=(0,0,0)

        Cylinder    center = pos+(0,0.45*cos(pi/4)  ,0.45*sin(pi/4)/2 + 0.1/2)
//L3
                    color = 0.4*col+ 0.6*white
                    length= 0.45*sin(pi/4) + 0.1
                    radius= 0.01
                    rotation=(-pi/2,0,0)

        Cylinder    center = pos+(0,0.45*cos(pi/4)+0.6/2,0.45*sin(pi/4))
//L4
                    color = 0.4*col+ 0.6*white
                    length= 0.6
                    radius= 0.01
                    rotation=(0,0,0)
```

```
        Cylinder    center = pos+(0,0.45*cos(pi/4)+0.6/2,0.45*sin(pi/4)/2)
//L5
                    color = 0.4*col+ 0.6*white
                    length= sqrt((0.45*sin(pi/4))^2+0.6^2)
                    radius= 0.01
                    rotation=(pi/6,0,0)

        Cylinder    center = pos+(0,0.45*cos(pi/4)+0.6 + 0.08,0.45*sin(pi/4)/2
//L6
                    color = 0.4*col+ 0.6*white
                    length= 0.45*sin(pi/4) / cos(pi/6)
                    radius= 0.01
                    rotation=(-pi/3,0,0)

        Cylinder    center = pos+(0,0.45*cos(pi/4)+0.6 -0.015,0.45*sin(pi/4) +
//"styre"
                    color = 0.4*col+ 0.6*white
                    length= 0.1
                    radius= 0.01
                    rotation=(pi/2,0,0)





)

//model wheel (pos,col) =
//initially
//_3D = (),_Plot=()
//always
//_3D = (

  //        Cylinder    center = pos+(0,0,0)    //L1
   //                   color = 0.4*col+ 0.6*white
    //                  length= 0.01
     //                 radius= 0.2
      //                rotation=(0,0,pi/2)

      // )
```

Appendix C

Cad Models

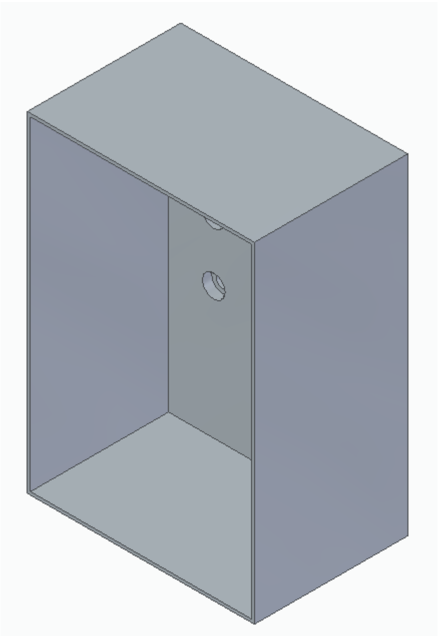


Figure .1. Box for holding MCU *created with Solid Edge.*

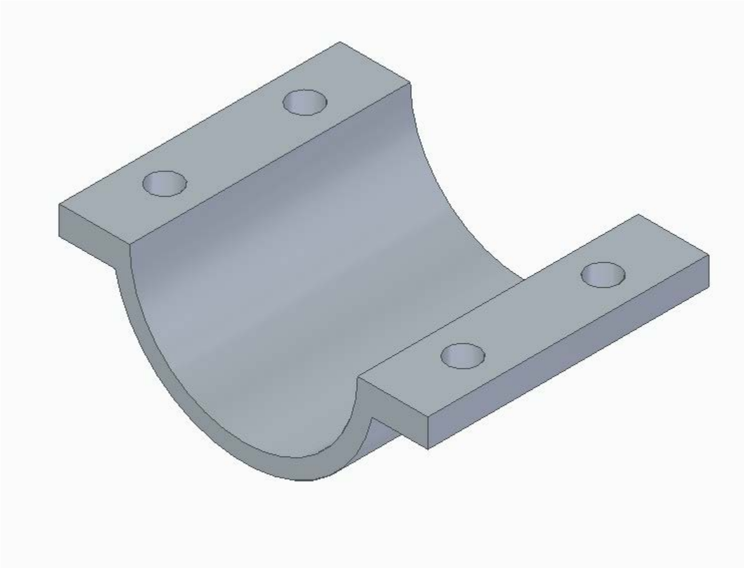


Figure .2. Mount for MCU box *created with Solid Edge.*

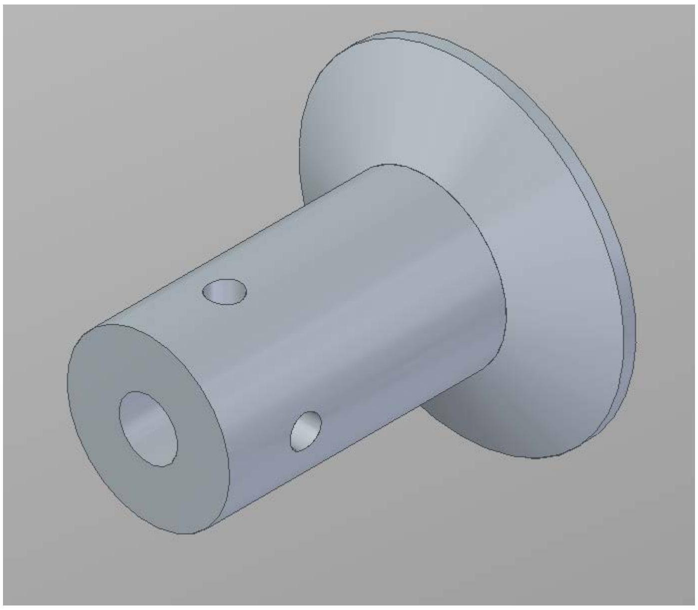


Figure .3. Axle coupling with end kap *created with Solid Edge.*

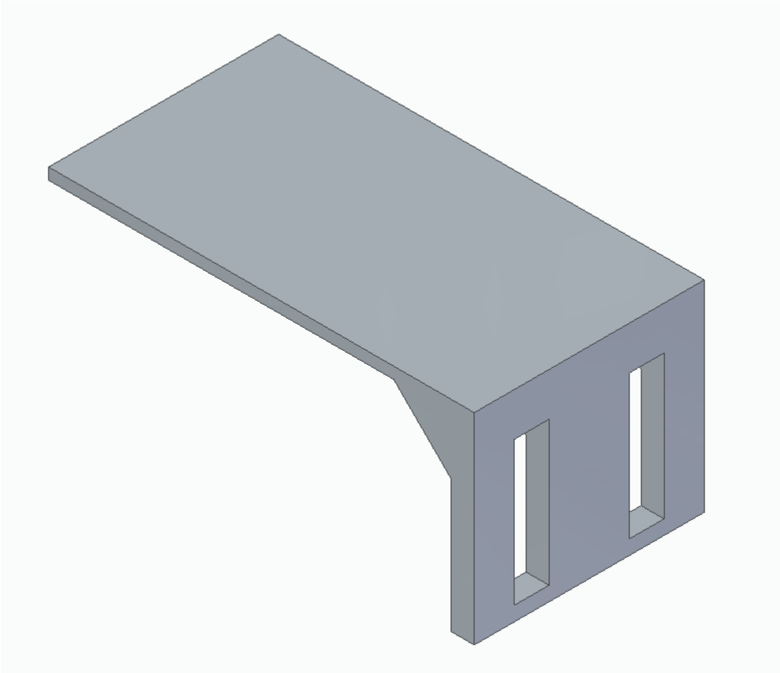


Figure .4. Rear wheel sensor mount *created with Solid Edge.*

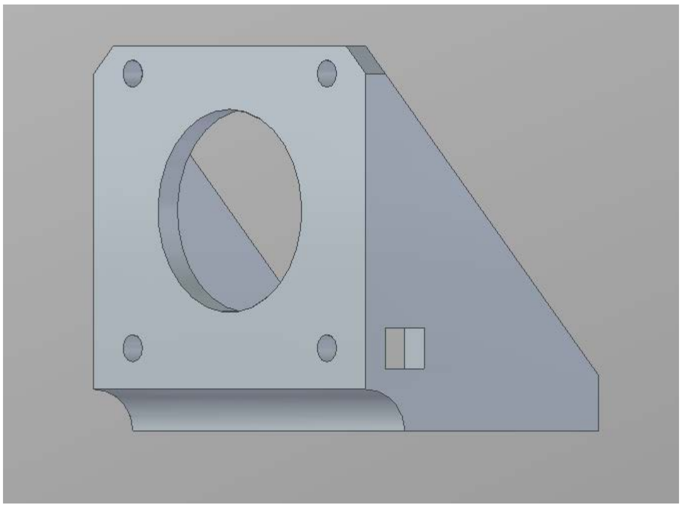


Figure .5. Motor holder *created with Solid Edge.*

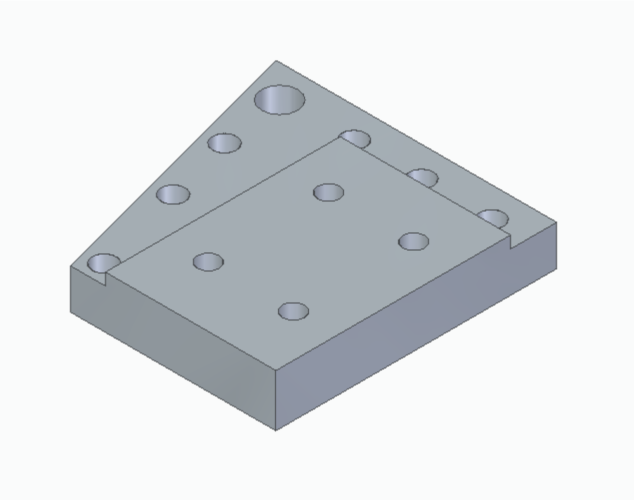


Figure .6. Motor mount *created with Solid Edge.*

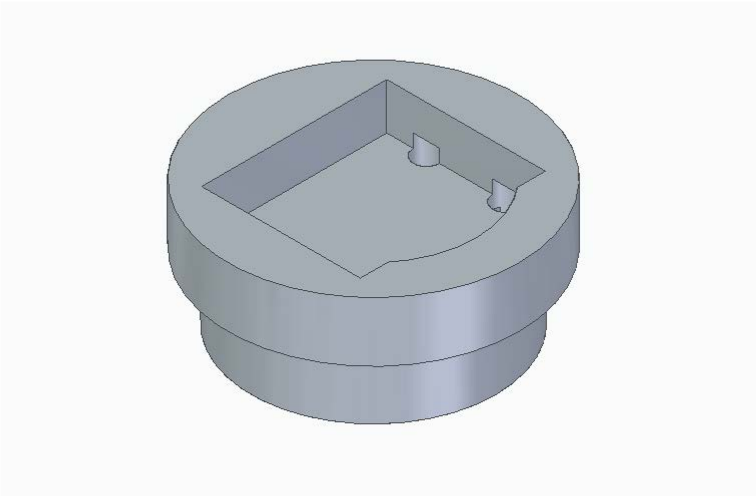


Figure .7. Endkap *created with Solid Edge.*

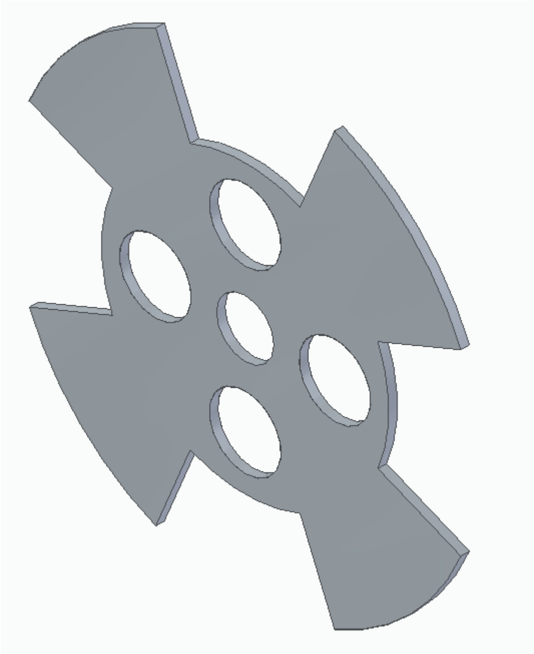


Figure .8. Rear wheel disc *created with Solid Edge.*

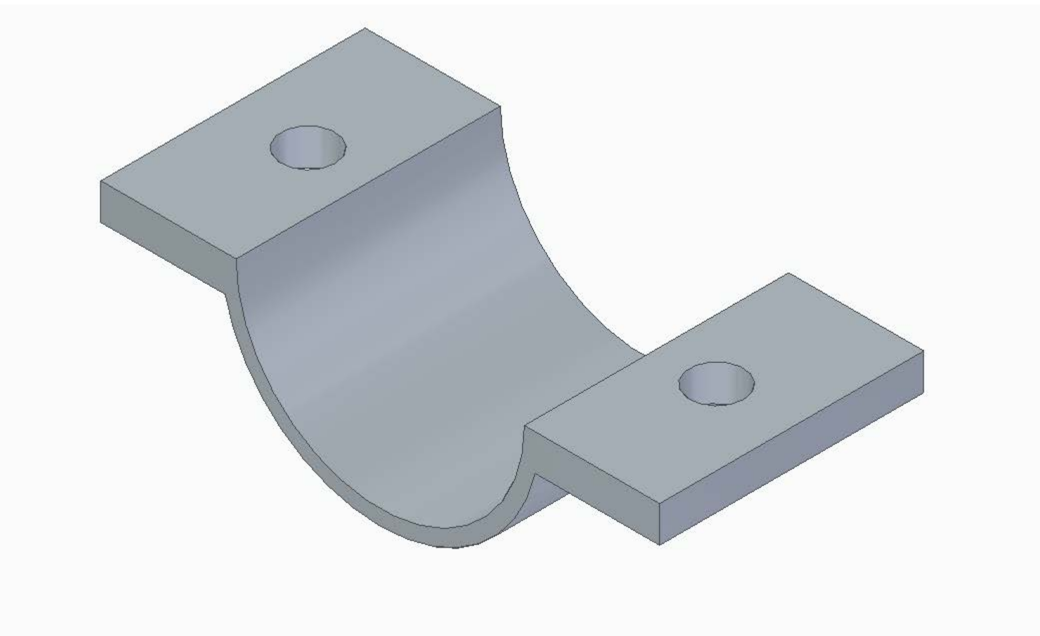


Figure .9. Bottom key switch mount *created with Solid Edge.*

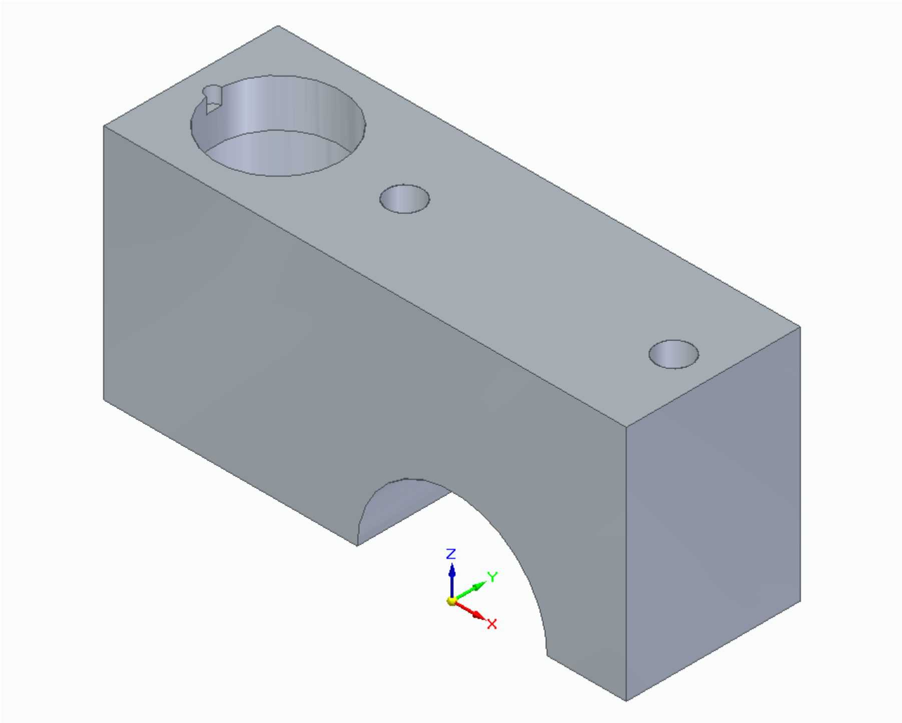


Figure .10. Top key switch mount *created with Solid Edge.*

Figure .11. Cylinder for US sensor *created with Solid Edge.*

TRITA-ITM-EX 2021:25