UPPSALA
UNIVERSITET

TVE-E 17 006 juni
Examensarbete 15 hp
Juni 2017

Independent project in electrical engineering

Magnetic hand timepiece

Erik Larsson
Niklas Kron

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Magnetic hand timepiece

Erik Larsson, Niklas Kron

Time is of utmost importance in today's society and this project's goal has been to build a clock. This project was carried out at Uppsala University in the course Independent Projocet in Electrical Engineering from January to May 2017. With the use of a microcontroller a motor was programmed to guide a time hand in a clock-wise manner. Controlled with a smartphone through bluetooth the user had the option to choose whether the time hand would show seconds, minutes or hours. Along with the traditional clock face a display also showed the current time, as well as the temperature in the room. In the end the timepiece showed the right time both on the display and with the time hand. Though due to different complications during the assembly the final product did not, as the motor did not work as intended after installation.

Handledare: Dana Salar
Ämnesgranskare: Ladislav Bardos
Examinator: Hana Barankova
ISSN: 1654-7616, TVE-E 17 006 juni

Abbreviations

RTC ..... Real-time clock

LCD ..... Liquid crystal display

BT ..... Bluetooth

MCU ..... Microcontroller unit

BLDC ..... Brushless direct current motor

DC ..... Direct current

PWM ..... Pulse width modulation

SoC ..... System on chip

WiFi ..... Wireless fidelity

IDE ..... Integrated development environment

IC ..... Integrated circuit

Contents

1 Introduction

1.1 Background

An invention of great importance that will be significant for all future is the ability to keep track of time. One of the first methods that was used to keep track of time is the sundial and archaeological record can track them back until around 1500 BC.[1] Henceforth the methods of keeping time has evolved and in today's society, time is used everywhere and at all times, meanwhile being extremely accurate. Nowadays many timepieces use electricity and displays the time mechanically with hour/minute hands or on a digital/analog display. A mechanical clock is usually driven by some kind of motor or clockwork while a digital clock is controlled by a microcontroller (MCU). In this project an Arduino microcontroller board is used to fulfill both operations, as well as measuring and displaying the current temperature. The microcontroller uses a real-time clock to keep track of time and a digital temperature sensor to measure the temperature.

A clock can be constructed in countless ways, the options are limitless. The magnetic timepiece offers the user to both see the time on a mechanical time hand as well as on a display. With the Arduino it is possible to choose whether the time hand should show the current hour, minute or second with a smartphone, connected via bluetooth.

The magnetic hand timepiece uses a bipolar stepper motor to operate the time hand, which then controls a magnetic sphere moving along a track on the clock surfaces. Due to the nature of the stepper motor a hall effect sensor is used to calibrate the clock and to let the microcontroller keep track of the position of the time hand when starting up or switching operation modes. A liquid crystal display is used to display the current time and temperature digitally.

1.2 Objective

The main target of the project is to create a clock that displays the time with a display and with a time hand. Further, it should also be possible to control if the time hand should show the current second, minute or hour with a smartphone via bluetooth. While no instructions are given to the clock it should continue to operate according to previous instructions. Additionally, temperature is also measured and displayed on the lcd. In conclusion:

- Show time on a display and with a time hand.
- Show current temperature.
- Optional to choose between seconds, minutes or hours.

1.3 Limitations

As a prototype the priority is to create a system that functions according to the requirements. The aesthetic aspect is given a lower amount of priority due to time limitation and budget.

2 Theory

In figure 1 one can see the block diagram of the communication lines between each component and the different Arduinos. Each component and communication line will be explained in section 2.1 through 2.7.

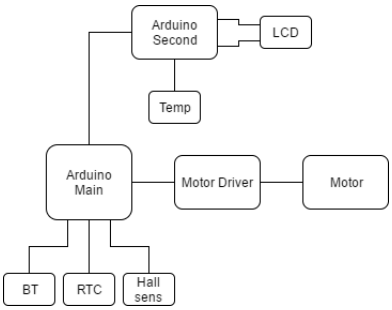


Figure 1: Block diagram

2.1 Magnetic motors

A revolutionizing tool for technology and industry is the mechanical engine. One of the most important is the brushless direct current electric motor (BLDC motor), which is a synchronous motor controlled by magnetic field. Typically the motor is constructed with a permanent magnet which rotates around a fixed shaft; the rotor. Therefore, there is no need to transfer any magnetizing current to the rotor.

To create rotation it requires a rotating magnetic field. This is created by switching the current in the coils surrounding the motor. Commonly, this is done with a microcontroller and a motor driver.[2]

For this project a stepper motor is used to control the clock arm. A stepper motor is recognized by its operation mode, the motor divides a revolution into a number of steps. If the motor is labelled with 200 steps per revolution it means that a revolution of 360 degrees is divided into 200 equally spaced steps 1.8 degrees between. It can either be operated in step by step or in a more continuous motion where you link multiple steps together.[3]

2.2 Bipolar Stepper motor

A simplified model of the bipolar stepper motor can be explained by two coils positioned 90 degrees apart. These coils produces the electromagnetic field to turn and move the rotor. It can be seen in figure 2.

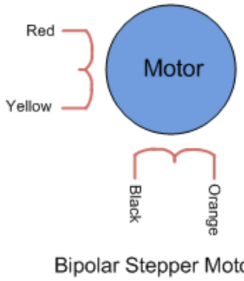
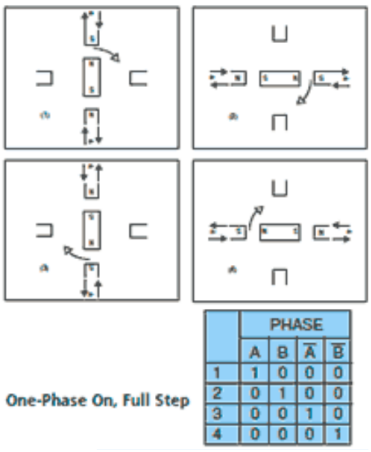


Figure 2: Schematic figure of Bipolar stepper motor windings [4]

By varying the direction of the current through the coils the direction of the magnetic field can be reversed. If the current passes through the coil from red to yellow the coil will represent a magnetic north/south pole and by reversing the direction of the current it will represent the other magnetic pole. Hence, one can create a rotating magnetic field by using a microcontroller that controls the current through the coils.[4]

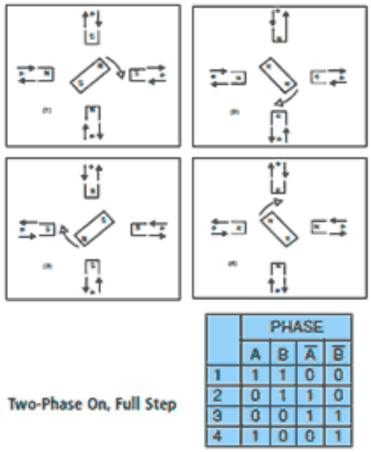
2.3 Step options

Depending on the demands of the motor it can be operated with different excitation modes. This will result in different step sizes and the three most common is full step, half step and micro step. Where full step utilizes the number of steps specified by the motor. In this mode one is also able to choose if the motor should be stepped with one or two phases simultaneously. One phase on lets the motor operate at the least amount of power, while with two phases on, the motor generates more torque and speed but requires twice the amount of power. Thus, it is possible to adjust the driver mode according to the operational demands. One phase- and two phase mode can be seen in figure 3 and 4.[5] [6]



	PHASE			
	A	B	$\overline{A}$	$\overline{B}$
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

Figure 3: One phase on [5]



	PHASE			
	A	B	$\overline{A}$	$\overline{B}$
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1

Figure 4: Two phase on [5]

To realize the rotating magnetic field needed to accomplish this type of rotation one need to excite the coils in the right order as well as in the right direction. As seen in figure 2, the bipolar stepper motor only uses two coils. Therefore it follows a certain pattern for full stepping with one phase on as opposed to two phases on. To create a virtual magnetic north pole on the opposite side of the black-orange coil, the black-orange coil acts as a magnetic south pole. This results in a certain direction of the current through the coil. For instance, to turn the rotor into a two o'clock position both the black-orange and red-yellow coil need to act as magnetic south poles. With these methods it is possible to generate a rotating magnetic field.

2.3.1 Half step

The second mode of interest is the half step mode. It is created by combining the two different options while operating with full step. By alternating these two methods one can divide the motor step into half steps. Half step excitation gives the opportunity to increase the resolution of the stepper motor significantly by reducing the step angle to half its original value. Other benefits include a smoother operation of the motor and it will be able to create more speed and torque. Unfortunately this is done at the expense of increased power losses. [5] [6]

2.3.2 Micro step

Finally it is possible to microstep the motor. This will generate the highest amount of resolution and the operation of the motor will seem to be continuous. In some stepper motors it is possible to divide each of the motor's steps in up to 256 microsteps.[7] However the disadvantages are that the motor produces less torque and the exact positioning of the rotor becomes less accurate.

In theory microstepping is commonly done by controlling the amplitude of the current through the coils. This is usually done by two ideal sine waves 90 degrees apart, called the sine-cosine-method, and will let the motor to operate without any noticeable stepping due to the two waves cooperating to make the stepping of the motor extremely smooth while transitioning from one pole to another.

Unfortunately microcontrollers are not able to produce ideal sine waves and therefore the result will not be the same as in theory. The microcontroller uses digital signals and thus only discrete values are possible and the output current will not be continuous, which can be seen in figure 5.[5] [6] [8]

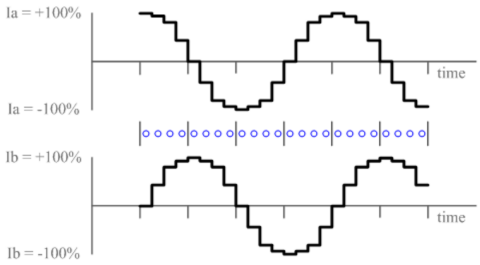


Figure 5: Sine cosine method [6]

Thus the number of microsteps is given by the quota between I_a and I_b according to

$$\theta = \tan^{-1}(\frac{I_a}{I_b}) \tag{1}$$

Any angle between 0 and 90 degrees represents a possible microstep. If one were to divide each full step into 4 microsteps, the quota of the current values need to generate the following theta values, 0 degrees, 22.5 degrees, 45 degrees, 77.5 degrees, 90 degrees. The phasors of these microsteps are illustrated in figure 6. Assuming that the phasors 0, 90, 180 and 270 degrees represent full steps and 45, 135, 225 and 320 represent half steps.[6]

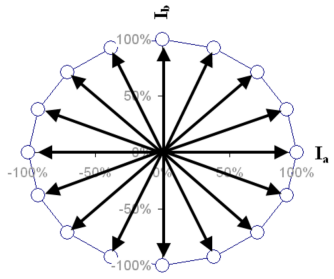


Figure 6: Phase diagram sine cosine method [6]

This mode of operation would let the motor to operate with 600 steps instead of the standard 200 steps.

2.4 Motor driver

The motor driver is an important electrical piece as it controls the excitation and current to the motor without having a direct connection between the motor and, in this project, the microcontroller. In larger applications the ATmega microcontroller is not powerful enough to drive the motor since the current is limited at 0.5mA. Therefore it is useful to control a motor driver or H-bridge, with an external power source, with the microcontroller.

In this project the L298N dual H-bridge driver is used. The motor uses four digital pins to communicate with the microcontroller. Since the motor used in this project is a bipolar stepper motor the operation of the motor driver becomes very simple. As can be seen in section 2.2 the windings only have to be turned on/off and reversed in polarity. However, if microstepping is used where needed it would be necessary to use a PWM-signal (pulse width modulation) to control the driver since the output current from the driver would have to be regulated according to different percentages. These can be seen in section 2.3.2.[10]

2.5 Real-time clock

Many electronic devices and applications are in need of keeping time since a lot of operations need to be executed in a certain order and might be dependent on what time it is. For example, mobile devices that need to be charged can detect when the device is usually charged for longer periods and slow down the charging during these hours in the future to prevent damage on the battery. A system that has been idle for longer periods can be put in sleep mode and digital speedometers need to know how fast the wheels spin by comparing how long it takes for them to do a full turn. Time logging is thus very important for many systems.

A real-time clock (RTC) is a circuit that makes this possible. Many built in clocks are used to log the time of the system and resets every time it is shut down. Many real-time clocks come as integrated circuits and utilize an oscillating crystal for reference. The accuracy for the clock is dependent on the frequency of this crystal. A stand alone RTC has some advantages over using the integrated clock that is present in microcontrollers and other circuits. For example, the external RTC is immune against lock up and can reset the microcontroller if it is stuck in a loop, it is often times more accurate and has a lower power consumption. With a battery the RTC can keep counting even after shutdown and let the system resume its operation when powered back on.[11]

2.6 Arduino

Arduino is a microcontroller development board with an open-source software based on the programming languages C and C++. It started as a project at the Ivrea Interaction Design Institute in Italy to help students without any background in electronics and programming learn more about these subjects. Arduino gained a wide community and adapted to the users' needs and desires. It was made with the purpose to make it cheaper and easier to develop electronic projects for professionals and non-professionals alike. The company offers several boards for different applications such as the Arduino UNO, Nano, Mega and more. They vary in memory size, clock frequency, number of inputs and outputs and some offer special features such as WiFi and ethernet connection.[12]

The Arduino Uno and Nano both utilize the Atmel ATmega328 chip. An 8-bit microcontroller unit with 32KB flash memory[13]. An MCU is a system on a chip (SoC), which is a small computer on an integrated circuit and they are widely used in embedded systems due to their size, price and simplicity. With programmable input/output peripherals and program memory users can customize the MCU to control devices as they see fit. There are different kinds of microcontrollers, such as various bit and memory size models with different processor speeds. Various microcontrollers are also used for different applications such as signal processing, automatization or security systems, and there are many more examples of applications for microcontrollers. Almost every electronic device has an MCU in it.

2.7 Temperature with moving average

Together with the magnetic sphere to display time the clock also displays the current hour, minute and second on an LCD, as opposed to the time hand only showing one of the units at a time. The screen is a 16x2 liquid crystal display with an adjustable contrast through the use of a potentiometer and is powered by the Hitachi HD44780 driver. Since it uses six digital pins on the Arduino board a second Arduino had to be implemented to the circuit to control the LCD. The time data is collected on the main Arduino from the RTC and then sent over to the secondary Arduino through serial communication.

On the screen the current temperature in the room is also displayed. The temperature is measured with the DS18B20 sensor from Maxim Integrated and a running average is calculated to smooth out and give a more accurate temperature reading. A running average (also called rolling average or moving mean) is a type of finite impulse response filter that takes a fixed number of previous data sets and calculates the mean value of these.[14] There are different methods for calculating the moving mean for different purposes such as simple, cumulative and weighted. They are often used to reduce impact from fluctuations and spikes in longer datasets and instead emphasise how the trend looks.

Simple moving average is the one used for this project. It takes the mean value of a fixed number of previous values and calculates the mean. Thus creating an unweighted average. It is called running mean because it shifts the subset forward by dropping the first acquired value and inserts a new one and calculates a more recent average. For this project the rolling mean is calculated over a subset of 60 values which means that the average temperature is obtained for one minute periods since a new value is stored every second. This means it takes a minute for the temperature reading to stabilize when the clock is reset.

3 Method and materials

3.1 Materials

Listed in section 3.1.1 through 3.1.8 are the most important components in the project setup. Other components that are not listed could be resistors, diodes, magnets and the 3D-printed body.

3.1.1 Arduino

Two Arduino nano are used. Arduino nano is a 16Mhz microcontroller board programmable in Arduino's open-source IDE, based on C and C++. The microcontroller has 8 analog pins and 12 digital pins. Analog pin A4 and A5 supports I^2C communication and all digital pins supports TX/RX communication. Its operating input voltage level is 5-12V and has a maximum output current of 40mA.[9]

3.1.2 Motor driver

The motor driver L298N H-bridge IC powers the stepper motor and uses four digital pins, pin 8 to pin 11, to communicate with the Arduino. The logical voltage level is between 5-35V and the drive current is between 0-2A.[10]

3.1.3 Motor

The motor used in the project is MY7001 bipolar stepper motor. Its rated current is 400mA and it has two phases with a resistance of 22Ω and inductance of 10mH each. It uses four wires connected to the motor driver.[15]

3.1.4 Temperature sensor

The digital temperature sensor is a DS18B2 which uses one digital pin, Digital pin 6. Its power supply range is 3.3-5.5V.[16]

3.1.5 Hall effect sensor

The hall effect sensor is a A1301 which has a supply voltage 4.5-6V and uses one digital pin, namely pin 7.[17]

3.1.6 Real-time clock

To keep the time an RTC that utilizes the ds3231 chip is used. It operates on 3.3-5V, communicates via I^2C and has a maximum clock frequency of 100kHz in standard mode and 400kHz in fast mode.[18] A 2032 coin cell battery is used to keep the RTC counting when it is not powered. The clock is connected to digital pins 4 and 5 on the main Arduino and it is calibrated against a computer's clock through a program in Arduino.

3.1.7 Bluetooth module

The bluetooth module used is a Keyes HM-10 model with bluetooth 4.0. On the module is a CC2541 chip from Texas Instruments and operates on 3.3-5V in the 2.4GHz band. It supports 250-kbps, 500-kbps, 1-Mbps and 2-Mbps data rates and communicates over I^2C.[19]

3.1.8 Liquid crystal display

The liquid crystal display is a LCM1602C which uses six digital pins on the second Arduino nano, digital pin 2-5 and digital pin 11-12. It operates at 3.3-5V supply voltage and uses 3.3-5V logic for the digital connections.[20]

3.2 Method

In the block diagram in figure 1 the setup and communication lines for the project can be seen. Simplified, the main Arduino operates the system while the second Arduino controls the LCD display and receives the time information from the main Arduino and calculates the temperature with a moving average. The main Arduino controls the motor driver and calibration as well as communication with the Bluetooth module and RTC. According to the signals received from the bluetooth module it chooses the operation mode for the time hand.

3.2.1 Stepper motor

To accomplish the stepping methods described in section, 3.3, 3.3.1, 3.3.2, one can either use a preprogrammed Arduino library called stepper, which has the fundamental stepping function for the stepper motor. It allows one to take a chosen amount of full steps. However, it was decided a library that allowed more stepping functions was to be programmed. In the created library there are options to take single steps as well as multiple steps. Also, the motor is operated in half step mode which means that one revolution is divided into 400 steps instead of 200 steps.

The stepping is done with a specific excitation pattern for the bipolar stepper motor, which can be seen in table 1, and is applied to the configuration in figure 2.

Wires	1	2	3	4	5	6	7	8
Red	Low	Low	Low	Low	Low	High	High	High
Yellow	Low	High	High	High	Low	Low	Low	Low
Black	Low	Low	Low	High	High	High	Low	Low
Orange	High	High	Low	Low	Low	Low	Low	High

Table 1: Excitation scheme for half step sequence

In the main code it is optional to choose the mode of operation for the time hand. The time hand has three different operation modes; it can show either seconds, minutes or hours. For each mode simple calculations were made to adjust the steps of the motor depending on seconds, minutes or hours.

On a regular clock the minute-hand completes a revolution every hour. To adapt this into the 400 steps of the motor the arm has to move one step in a given amount of seconds in between. A revolution consists of 60 minutes and each minute consists of 60 seconds. Thus the 3600 seconds are divided equally between the 400 steps. Hence the motor needs to step once every $3600/400 = 9$ seconds. For the hour-hand it completes a revolution every twelfth hour. This equals 43200 seconds and thus the motor needs to step once every $43200/400 = 108$ seconds.

Unfortunately, the second-hand makes 60 steps in one revolution which is not dividable by 400, nor is a multiple of 60 dividable by 400. Therefore, a certain pattern has been chosen to keep the clock arm in sync with the display clock, this pattern lets the clock arm complete a full revolution in 60 seconds but with a different amount of half steps included in each step.

3.2.2 Calibration and setup of the time hand

Since the motor used in the project is a stepper motor it is necessary to calibrate the position of the time hand. Upon start the Arduino has no information or reference for the position of the time hand and therefore calibration is needed. The calibration is done with a hall effect sensor and a magnet. The hall effect sensor is fixed onto the wall of the clock body and connected to the main Arduino while the magnet is attached to the top of the time hand. Each revolution the magnet passes by the hall effect sensor once.

When the Arduino is restarted it steps once and checks the input of the hall effect sensor. This is repeated until the time hand is in a vertical position at 12 o'clock. Due to the sensitivity of the hall effect sensor one step is added after the position is calibrated. This gives a better approximation of the vertical position.

Further, each time the mode of operation is changed the position of the time hand is recalibrated. Thenceforth the current time is acquired from the RTC and a hypothetical position of the time hand is calculated and moved accordingly. For instance, if the clock is supposed to operate as a minute hand and the time is 25 minutes past, the time hand will move 166 steps. Thereafter it follows the chosen stepping pattern. The stepping scheme is explained further in section 2.3.

3.2.3 Bluetooth module and cases

To control the mode of operation the main Arduino needs input from a smartphone connected via the bluetooth module. While using the phone the operator has the option to choose between hours, minutes and seconds from an application on the phone. It is possible to use both Android and Ios.

In the main Arduino the implemented code is divided into three different cases; one for each mode of operation. The difference is which stepping pattern that should be used; hours, minutes or seconds. If there is no available information to be read from the BT-module the system does nothing and continues to check for new instructions from the BT-module each loop iteration.

When there are instructions to read the information is matched against the different cases and if it matches, that operation mode is chosen. This mode continues until another instruction is given.

3.2.4 Real-time clock

The real-time clock keeps track of time and updates every second and it is connected to the main Arduino which continuously compares the stored second value with the current value. When the values are not equal the stored second value is updated to the current second value. Every second the main Arduino sends an array with the current time information to the second Arduino, which updates the time on the display. At the same time the second Arduino updates the temperature value. Finally it moves the time hand according to the mode in operation.

3.2.5 LCD

The second Arduino is mainly used to control the display since a sufficient amount of digital output pins on the first Arduino was not available. Hence the time from the RTC had to be shared between two units. This was done by letting the first Arduino read the time and send it over through serial communication to the second Arduino which displays it on the LCD. Current temperature in the room is also presented on the screen. The screen's contrast is controlled by a voltage divider consisting of a 3.3kΩ and 10kΩ resistor while the backlight is limited with a 220Ω resistor between pin 15 and 5V.

3.2.6 Assembly

Two dual in-line sockets were soldered on to a prototype circuit board to hold the two Arduinos. On the same circuit board the RTC, bluetooth module and temperature sensor were also soldered. Four pins were soldered to the inputs for the motor pins for easier mounting and removal of the motor driver with the use of female-female contacts.

The clock body and time hand were modelled in a CAD design program called Solidworks and 3D-printed from these sketches. Since the available 3D-printer only could print 22x22x22cm the clock body had to be split in four different bodies due to its size. Therefore joints were created on the clock body in order to strengthen the rebuilt construction. A steel plate was also screwed on the backside of the clock body to

further increase its strength and stability. The final sketch can be seen in figure 7. The time hand was printed in one session and can be seen in figure 8.

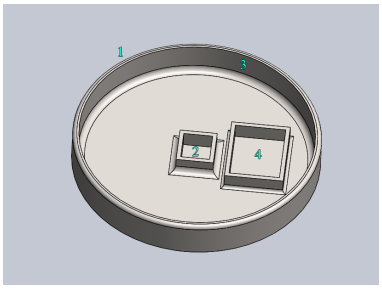


Figure 7: Clock body

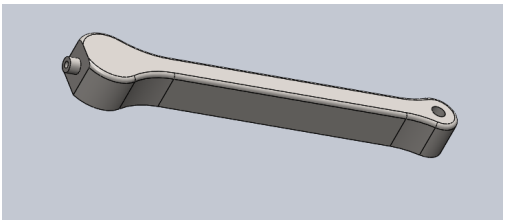


Figure 8: Time hand

1. The diameter of the clock body was chosen to be 300mm due to aesthetic reasons. A common size for wall mounted clocks.
2. This is the frame that holds the motor in place and is centred around the origin of the clock body. The walls were adjusted in relation to the height of the motor so that the axle could rotate freely when the time hand was mounted on the rotor. The motor dimensions were acquired from the datasheet, which can be seen in figure 9.

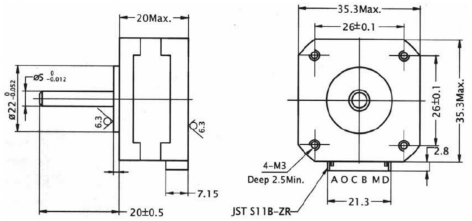


Figure 9: Stepper motor sketch.[15]

3. The clock walls were adjusted in relation to the height of the clock axle which is approximately 4cm. Half a centimetre was added to make space for the magnetic disk and also to make room for some error when 3D-printing.
4. Socket created to mount the experimental circuit board.
5. In figure 8 the time hand can be seen. The length is dimensioned to be approximately half a centimetre from the inner clock wall; 29cm in diameter. On the top of the time hand a mount was created for the calibration magnet.

The clock lid was also designed is Solidworks. To save time the sketch was simplified so that the part could be milled instead of 3D-printed. Its thickness was set to circa half a centimetre and the bottom side of the lid was given a shorter diameter so that it could fit inside the walls of the clock body. On the top side of the lid a track for the sphere was milled to allow it to move smoother along the surface. The thickness in the track was also decreased significantly so that the magnet on top of the time hand did not get too far away from the magnetic sphere. Finally the track was sanded to decrease the amount of friction between it and the magnetic sphere. The final lid is shown in figure 10 and 11.

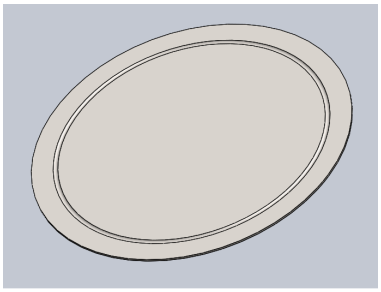


Figure 10: Clock lid frontside with magnetic sphere track.

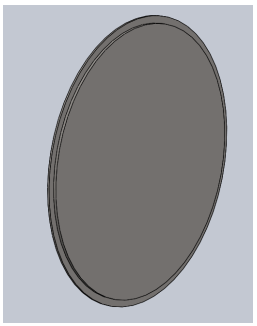


Figure 11: Clock lid backside.

Final constructions included:

- Creating a hole for supply voltage from a wall socket.
- Mounting and adjusting the hall effect sensor on the clock wall to register the magnet that is attach to the time hand.
- Glue the experimental circuit board into its socket, attach the motor driver to the clock body and connect it to the motor.

4 Results

In figure 12 to 15 the clock prototype can be seen inside the lid and on the outside. The described functions and requirements were to some degree fulfilled. Everything was powered sufficiently and the devices operated accordingly, though due to other interfering factors the motor was not able to move in a clock-wise motion with the originally intended equipment.

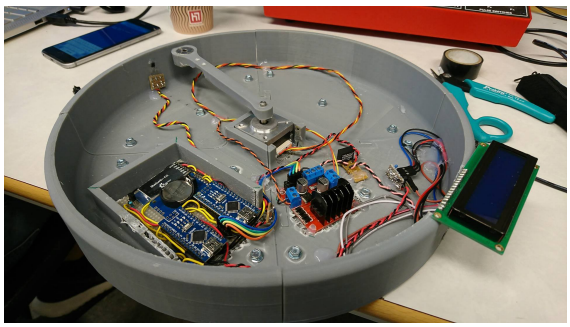


Figure 12: Inside of the clock with everything put in place.

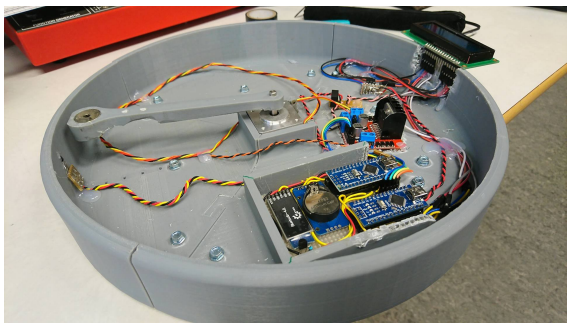


Figure 13: Inside of the clock with everything put in place.

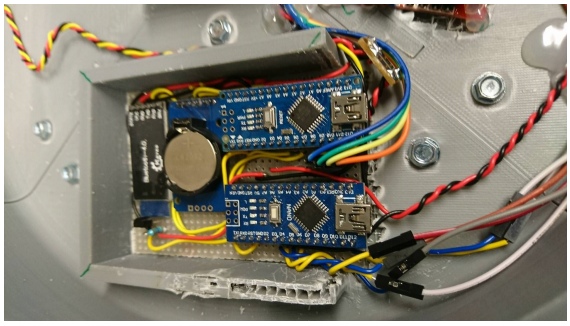


Figure 14: The two arduinos with the RTC, bluetooth module and temperature sensor.

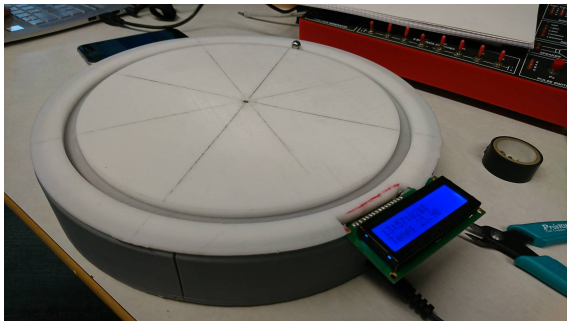


Figure 15: The clock from the outside, powered from the grid.

During the final testings of the clock some errors occurred. A defection in the motor was noticed when operated for a longer period of time. As the heat increased the motor's axle had a tendency to get stuck at a specific place. This affected the motion of the time hand and the accuracy of the position while moving into the theoretical position after calibration.

Further, the attached metallic backside and cables connected to the LCD affected the magnet and attracted it, which disrupted the motion of the time hand. With the lid mounted and the magnetic sphere put in its track, guided by the time hand, the motor did not suffice to guide the metal sphere in a convincing manner. Without the heavy magnet mounted on the time hand, which also alters its balance, the motor was stable enough.

5 Discussion

The magnetic hand timepiece prototype works quite well and satisfies the demanded requirements with sufficient accuracy and precision, though not with the intentional way of presenting the time. While running, the mode of operation can be chosen and the system executes accordingly. There are some timing issues when the clock is operated for a longer period of time and the real position of the time hand then differs from the theoretical position.

During the building process we learned that there are many aspects to consider that one originally might not have thought of. One example is that when printing parts in a 3D printer tolerance in the designs has to be taken to account, or else they might not fit properly when assembled. It would not necessarily have been a problem if the whole body would have been printed at once, but since we printed it in parts, due to size limitations in the printer, the different pieces did not fit perfectly in to each other.

When all the parts were assembled the clock bulged and created difficulties for the rest of the assembly, not least for the lid. To stabilize the clock body it was screwed down on a metal plate. This solution, however, created another problem. As mentioned in the results section the magnet on the time hand was affected by the back plate as it attracted it down towards the clock floor. The milled clock lid also got too heavy and the friction in the track too high. With additional time these are some things we would have liked to improve upon.

Other noticed disadvantages is that compared to a regular clock it is quite heavy, which makes it problematic when mounting the clock on a wall. Some walls might even be unsuitable, due to the potential damage the weight of the clock can do, for example plaster walls. Another detected problem is that it is energy inefficient and therefore it was chosen to power the clock from a wall socket rather than from batteries. This will further limit the positions the clock can be mounted at. From a business perspective, this makes the product less attractive to manufacture and sell, due to the potential loss of customers.

5.1 Stepper motor and calibration

The stepper motor was our choice since it is easy to use in practise and operates similar to a regular clock. It provides good accuracy and especially with the different step options that can be implemented. Unfortunately it loses torque when half or microstepping is used. However this project is not torque dependent. Furthermore the stepper motor is pretty cost and power efficient and there is no need for additional equipment, except for calibration.

One disadvantage with the stepper motor is that it does not have any references system compared to for example a servo motor. Which has a built-in microchip with a regulator that keeps track of the position using a potentiometer. Unfortunately it is not suitable for multiple revolutions. A DC-motor could have been used, though it would be needing additional equipment to operate as a regular clock. The BLDC motor would also have been a good choice, however it is harder than the stepper motor to operate.

Due to the uncertainty while stepping it is necessary to recalibrate the motor regularly. This is done every time the Arduino is started and every time the mode of operation is changed. To improve the accuracy further one could have checked the status of the hall sensor for each revolution of the time hand and compare it to the time of the RTC clock. If there is a difference, the time hand could either be stalled or sped up to stay in phase with the current time.

5.2 RTC and second Arduino

When the second Arduino was implemented in the circuit we had to find a way to send the time data from the RTC to that board as well. In a first attempt to solve the problem we set up the two Arduino boards to take turns reading the external clock. This did however not work as intended and instead of the current time we received random numbers. Since the RTC communicates over I^2C we believe we could have set up a master/slave configuration since the protocol only allows serial communications. With such a set up the slave (second Arduino in this case) is only allowed to read the data if the master (main Arduino) is not collecting data from it.

In another attempt we let only the main Arduino read the RTC and send the data through the Tx pin to the second unit's Rx pin. This method works well, but one has to be observant of the Rx pin when uploading new code to the board since it is not possible to do as long as the Rx pin is connected.

5.3 Improvements

To make the clock more energy efficient and lower the power consumption we programmed the stepper library to only power the motor when it had to move. Previously it was powered at all times and thus consuming more power. However, since we are using a stepper motor this is not possible if one would like to have the clock in any position other than lying flat. When the motor's electromagnets are powered the shaft is locked in place and released when the voltage is turned off. Thus our method to lower the power consumption results in the time hand falling down when going past twelve o'clock and not being able to climb up again.

In a commercial perspective this clock might not compete very well on the market. Since the timepiece only utilizes one time hand an improvement would be to include another motor and make it possible to show both hours and minutes simultaneously, as well as keeping the option to use only one hand. It would be easy to remove one hand as the indicator is just a magnet on the clock surface. However another motor might have to replace the one in this prototype as to not raise the price too much. An alternative to changing motor, is implementing a conventional clockwork with cogs would allow more hands. Though there are some uncertainties to this method as we are not sure how microstepping and the changing of modes would work.

6 Conclusion

This project's operation has been going fairly smooth overall with a few setbacks from time to time. Mostly minor (but also time consuming in the sense of debugging) things in the code or in the circuit, like a bad connection or wrongly implemented algorithm. The resulting clock was a heavy and power-inefficient timepiece with a big price tag. Even though the power consumption was reduced it was at the cost of the clock's performance and hence the method for lowering the power was not implemented.

Luckily the main focus has not been to produce an aesthetic appealing wall clock, but instead get the motor to move in a convincing manner in relation to the flow of time. Because of this the prototype is not very pretty, and while some adjustments in the design could have been made and several versions produced, the first design was considered adequate since it took almost three working days to 3D-print everything and it was good enough.

Final adjustments to make the product better would be to decrease the weight further, as mentioned previously this can be done by changes to the body or a lighter motor. Increasing accuracy and precision for the time hand also is a necessary improvement. Lastly, powering the system from batteries would make the product more desirable for potential customers. All these modifications would improve the final result.

6.1 Acknowledgement

We would like to thank our supervisor Dana Salar for all help during this project. We also want to thank the division of electricity at Uppsala university for the financial support.

7 References

[1] Vodolazhskaya (2013), Sundial
L.N. Reconstruction of ancient Egyptian sundials. Archaeoastronomy and Ancient Technologies, Accessed 2017 may 10.

[2] All about circuit (2013), *Brushless DC motor*,
https://electronics.stackexchange.com/questions/43105/control-differences-between-ac-induction-motor

[3] Bill Earl (2015), *What is a stepper motor*,
https://learn.adafruit.com/all-about-stepper-motors/what-is-a-stepper-motor

[4] George (2012), *Unipolar stepper motor vs bipolar stepper motors*, [image], Accessed 2017 may 10,
https://www.circuitspecialists.com/blog/unipolar-stepper-motor-vs-bipolar-stepper-motors

[5] Unknown author (2014), *Full-, half- and microstepping*, [image], Accessed 2017 may 15,
http://www.nmbtc.com/step-motors/engineering/full-half-and-microstepping/

[6] Zaber wiki (2014), *Microstepping*, [image], Accessed 2017 may 5,
https://www.zaber.com/wiki/Tutorials/Microstepping

[7] Micromo (2015), *Microstepping. Myths and realities*,
http://www.micromo.com/technical-library/stepper-motor-tutorials/microstepping-myths-and-realities

[8] Coeleveld (2015), *Arduino stepper*,
https://coeleveld.com/arduino-stepper/

[9] Arduino, *Arduino nano*,
https://www.arduino.cc/en/Main/ArduinoBoardNano

[10] Reichenstein7 (2015), *Arduino Modules - L298N Dual H-Bridge Motor Controller*,
http://www.instructables.com/id/Arduino-Modules-L298N-Dual-H-Bridge-Motor-Controll/

[11] Digi-Key Electronics (2011), *Enabling Timekeeping Function and Prolonging Battery Life in Low Power Systems*
https://www.digikey.com/en/articles/techzone/2011/dec/enabling-timekeeping-function-and-prolonging-b

[12] Arduino (2017), *What is Arduino?*,
https://www.arduino.cc/en/Guide/Introduction

[13] Atmel (2016), *8-bit AVR Microcontroller ATmega328/P Datasheet Summary*,
http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P Summary.pdf

[14] Investopedia (2005), *Moving average*,
http://www.investopedia.com/terms/t/tsi.asp

[15] Farnell (2016), *ASTROSYN MY7001 Stegmotor*,
http://se.farnell.com/astrosyn/my7001/stepper-motor-1-8deg-mini-hybrid/dp/8425914

[16] Sparkfun (2009), *One Wire Digital Temperature Sensor - DS18B20*,
https://www.sparkfun.com/products/245

[17] Allegro MicroSystems, Inc. (2010), *Continuous-Time Ratiometric Linear Hall Effect Sensor ICs*,
http://pdf1.alldatasheet.com/datasheet-pdf/view/120794/ALLEGRO/A1301.html

[18] Maxim Integrated (2015), *Extremely Accurate I2C-Integrated RTC/TCXO/Crystal*, https://datasheets.maximintegrated.com/en/ds/DS3231.pdf

[19] Texas Instruments (2012-2013), *2.4-GHz Bluetooth^{TM} low energy and Proprietary System-on-Chip*, http://www.ti.com/lit/ds/symlink/cc2541.pdf

[20] Liquid crystal display, *LCM1602C Datasheet PDF – 16x2 LCD display*, http://www.datasheetcafe.com/lcm1602c-datasheet-pdf