



## **Středoškolská technika 2022**

**Setkání a prezentace prací středoškolských studentů na ČVUT**

# **HODINY - ORLOJ**

**Tomáš Janoušek**

VOŠ, SPŠ a JŠ Kutná Hora  
Masarykova 197 - Kutná Hora



VYŠŠÍ ODBORNÁ ŠKOLA, STŘEDNÍ PRŮMYSLOVÁ ŠKOLA  
A JAZYKOVÁ ŠKOLA S PRÁVEM STÁTNÍ JAZYKOVÉ ZKOUŠKY

KUTNÁ HORA, MASARYKOVA 197

# Maturitní práce

**Rok 2022**

**Tomáš Janoušek**



VYŠŠÍ ODBORNÁ ŠKOLA, STŘEDNÍ PRŮMYSLOVÁ ŠKOLA  
A JAZYKOVÁ ŠKOLA S PŘÁVEM STÁTNÍ JAZYKOVÉ ZKOUŠKY

## ZADÁNÍ MATURITNÍ PRÁCE S OBHAJOBOU PŘED ZKUŠEBNÍ KOMISÍ

studijní obor: (26-41-M/01) ELEKTROTECHNIKA

Student: Tomáš Janoušek

Třída: E4A

Školní rok: 2021 / 2022

Téma: **22012** Elektronické hodiny s mechanickým výstupem

Konkretní ukoly řešene v maturitní práci:

Navrh a konstrukce digitalních hodin s ručičkovým ciferníkem

Zadání schváleno dne: 30. 9. 2021

**Termín dokončení práce a předání výstupů vedoucímu práce: 31. 3. 2022**

Vedoucí práce:

Ing. Stanislav Moravec

Ředitel školy:

Ing. Josef Tremel

## PROHLÁŠENÍ

Prohlašuji, že jsem svou maturitní práci vypracoval/a samostatně a použil/a jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů.

Prohlašuji, že tištěná a elektronická verze maturitní práce jsou shodné.

Nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

Současně souhlasím se zpracováním uvedených osobních údajů dle nařízení GDPR (nařízení Evropského parlamentu a Rady EU č. 2016/679 ze dne 27. dubna 2016 o ochraně fyzických osob v souvislosti se zpracováním osobních údajů a o volném pohybu těchto údajů) po celou dobu archivace mé maturitní práce. Po převedení dokumentu do archivu školy mohu kdykoliv využít právo písemnou formou svůj souhlas se zpracováním osobních dat odvolat.

V Kutné Hoře dne 13. dubna 2022

.....  
(vlastnoruční podpis)

## **PODĚKOVÁNÍ**

Tímto drobným však důležitým textem bych rád poděkoval předem učitelům, kteří mi věnovali čas při odlaďování detailů na mé práci. Velké poděkování patří mému hlavnímu vedoucímu, panu Stanislavu Moravci. Musím zde zmínit taktéž učitele, který díky mne neměl ráno čas na klidné probuzení, pana učitele Václava Brože. Za vlídnou a ochotnou pomoc při programování musím zmínit mého třídního učitele Pavla Stejskala. Za dobrou podporu v programování s ESP a radami zde zmiňuji svého spolužáka Vítka Růžičku.

## **ABSTRAKT**

Hodiny jenž se automaticky nastaví pomocí zařízení ESP, které bude mít za úkol hlídat čas podle internetu. Tato data se budou ukládat do „paměti“ RTC, kde bude program brát údaje, pro případ výpadku elektřiny je zde baterka a zařízení tak nezapomene hodnotu hodin. Tedy mělo by se jednat o plně automatické hodiny, které budou z velké části tisknutelné. Je tu jen jedna mechanická součástka a to krokový motor NEM-17, s mnou vytvořeným převodem z 3D tisku.

## **Abstract**

A clock that is automatically set by the ESP device, which will monitor the time according to the Internet. This data will be stored in the "memory" of the RTC, where the program will take the data, in case of power failure there is a flashlight and the device will not forget the value of the hour. So it should be a fully automatic clock that will be largely printable. There is only one mechanical component, the NEM-17 stepper motor, with a 3D print conversion created by me.

## **Klíčová slova**

Hodiny, ESP modul, Automatické řízení, 3D model, CAD, Programování ESP/arduino, NEM-17, Fusion 360,

# OBSAH

## Obsah

Prohlášení .....	3
Poděkování .....	4
Abstrakt .....	5
Klíčová slova .....	5
OBSAH .....	6
1 Úvod .....	7
1.1 Využité komponenty .....	8
1.1.1 ESP 32– WROOM .....	8
1.1.2 RTC modul DS1302 .....	8
1.1.3 Motorový kontrolér A4988 .....	9
1.1.4 Krokový motor NEMA-17 .....	9
1.1.5 Napájecí zdroj .....	9
1.2 Mechanická část .....	10
1.2.1 První možnost .....	10
1.2.2 Druhá možnost .....	10
1.2.3 Výpočet zubů a postup .....	10
1.2.4 Finální verze provedení sestavy .....	11
1.2.5 Synchronizace a převodovka .....	12
1.2.6 Finální verze převodovky .....	13
1.2.7 Hřídele .....	13
1.2.8 Spodní a horní část konstrukce .....	14
1.3 Programování úvod .....	16
1.3.1 Home pozice .....	16
1.3.2 Přepočítání na čas / kroky .....	16
1.3.3 Čtení RTC .....	17
1.3.4 Zápis RTC .....	17
1.3.5 ESP „připojování ke světu“ .....	18
1.4 Samotná sestava .....	19
1.4.1 Tisk .....	19
1.5 Závěr .....	20

# 1 ÚVOD

Cílem mé práce bylo sestavit hodiny s analogovým výstupem. Musel jsem navrhnout mechanické provedení, elektronickou část, řídicí program a následně sestavit hodiny. Musel jsem zvládnout programování v ESP, naučit se pracovat s krokovým motorem NEMA-17. Mechanickou část jsem navrhl v CAD programu Fusion 360. Spočítal jsem potřebné převody a odladil jsem je v Microsoft Visual Studio, kde jsem vytvořil program kvůli hledání „teoretické“ chybovosti při daných poměrech.



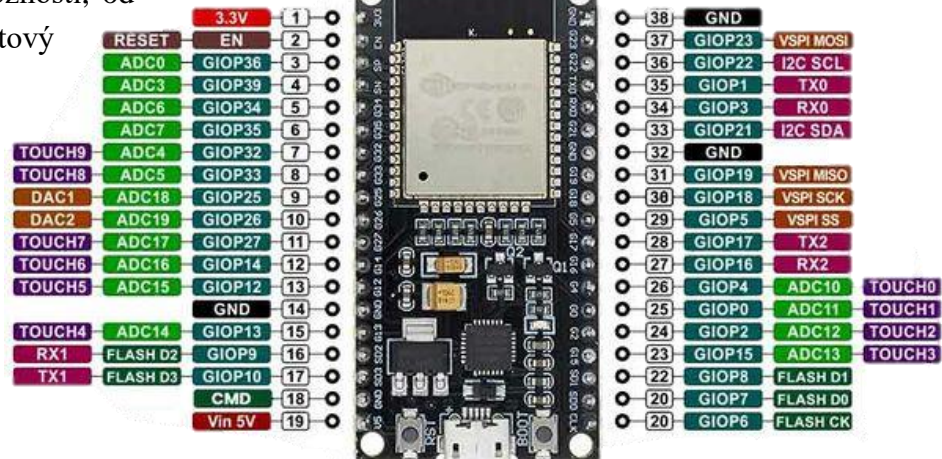
## 1.1 Využité komponenty

Z počátku jsem uvažoval o použití mikropočítače na bázi Arduina. Po delším uvážení jsem zjistil, že lepší volbou bude použit modul ESP 32. Obr.1. EPS32 - WROOM

### 1.1.1 ESP 32– WROOM

Jedná se o rychlou, levnou a multifunkční programovací desku (mikropočítač), která má řadu možností, od funkce wifi až po rychlejší 32 bitový procesor.

**Parametry :** 34 GPIO pinů (záleží na typu), Wifi (800 b/g/n), Bluetooth, 10 kapacitních senzorů / pinů, možnosti PWM pinů, několik GPIO jsou pouze vstupní. Osobně jsem kupoval za 250,- Kč.



„rodina ESP32“ : velkou výhodou je, že většina typů esp32 může mít stejné programy, tedy nemají skoro žádné rozdíly, převážně se můžeme setkat pouze s rozdíly v zapojení nebo frekvenci, na které daný mikroprocesor pracuje. Většinou pracují v řadách okolo 200+ MHz. Výhodou je jejich možnosti programování, lze je programovat i jinde, bez větších problémů, než jen na Arduino IDE. Pro tyto případy lze programovat v Pythonu, MicroPython, Visual Studio Code, NodeMCU a spousty dalších platform.

### 1.1.2 RTC modul DS1302

Od vedoucího mé práce jsem dostal modul, který je starší řady a bylo tedy obtížné sehnat nějaký materiál pro jeho ovládání (knihovny). Použití tohoto modulu je nezbytné pro případ výpadku internetového signálu. RTC modul v tomto případě udrží aktuální stav časové hodnoty.

**Připojení modulu :** prostřednictvím pěti pinů, tři datové a dva napájecí. Mezi datové spadá „CLK“, „DAT“ a „RST“. Cenová relace je cca 20,- Kč.

**Programování:** Komplikované bylo naučit se jej obsluhovat. Jsou zde novější verze s jednoduchými a velice snadno dohledatelnými knihovnami. Využil jsem knihovnu „**ErriezDS1302**“ (přiložena v dokumentech), ona samotná ještě vyžaduje (tedy v mém případě) ještě knihovnu „**Wire**“.

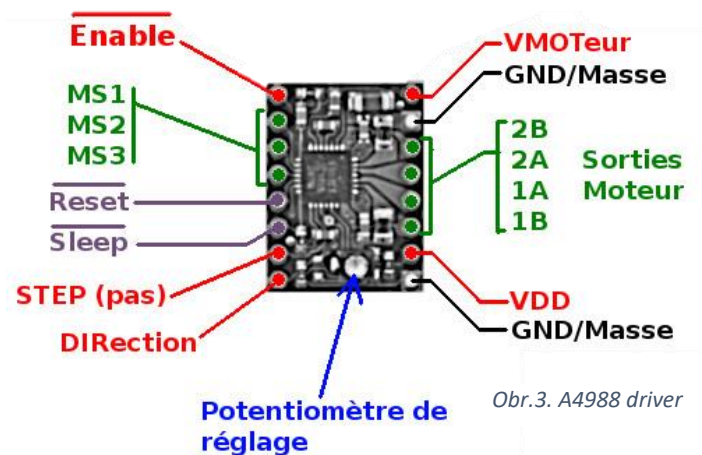


Obr.2. RTC DS1302

### 1.1.3 Motorový kontrolér A4988

Zařízení, které se využívá pro ovládání některých typů krokových motorů. Spotřeba motoru je tak velká, že samotný výstup z ESP by nebyl schopný snést proudové špičky, které vznikají při rozběhu motoru.

Ovládá se opět několika piny z ESP, které umožňují řízení rychlosti pomocí PWM. Napájecí napětí pro motor může být 12 – 20 V, při proudu v řádech stovek mA. Také je levný, cenová relace je okolo 50,- Kč.



**Programování :** Za pomoci knihovny „AccelStepper“, dokumentace na stránkách (od.1. AccelStepper dok.)

### 1.1.4 Krokový motor NEMA-17

Jedná se o krokový motor, který má velkou přesnost a solidní robustnost. Zvolil jsem jej také díky jeho specifickému kroku, který je 1.8°.

To znamená 200 kroků na jednu otáčku s možností mikrokrokování 0.9 °.

Rozměry 43x43 mm. Proud 1,5 – 1,8 A na fázi. Napětí od 10 - 20 V. Síla otáčení 44 Ncm. Má pouze jednu hřídel a je bez převodovky. Cenová dostupnost okolo 600,- Kč.



**Programování:** co se ovládání týče vše je řešeno přes driver A4988 (viz zmíněný). Prostřednictvím knihovny „MultiStepper“ a „AccelStepper“ dle dokumentace na stránkách (od.1. AccelStepper dok.).

### 1.1.5 Napájecí zdroj

V projektu jsem potřeboval dvě různá napájecí napětí, 5 a 12 V. Zvolil jsem napájecí adaptér 12V a napětí 5V jsem získal z tohoto zdroje příslušným modulem.

## 1.2 Mechanická část

Vytvořit návrh, který by se správně vytiskl a zároveň aby splňoval složité požadavky typu, výdrž, velikost a jednoduchost provedení nebylo zdaleka jednoduché.

### 1.2.1 První možnost

Začínal jsem konstrukcí samotných hodin jakožto převodového mechanismu. Tato cesta byla první ale ne zdaleka finální. Jednalo se o možnost 2 hřídelí, které by byly samostatné, uprostřed by byla převodovka s vnořenou další převodovkou. Tento návrh měl spoustu nedostatků a byl velice složitý na výrobu.

### 1.2.2 Druhá možnost

V této fázi jsem se ujistil, že nehodlám dělat dvojitou převodovku a nic tak složitého, chtěl jsem něco jednoduššího. Začal jsem proto shánět informace o tom jak takové hodiny vypadají a jak se dají vypočítat různá provedení takových převodů ( 1:60 ). Našel jsem „planetární převodovku“. Jedná se o takový převod, který dnes můžeme najít v autech s automatickou převodovkou. V převodu se nachází kola „sun“, „planet“ a „ring“. Taková ozubení mají dané poměry mezi sebou, jejich následný výpočet je taktéž jednoduchý.

### 1.2.3 Výpočet zubů a postup

$i$  = poměr kol ;

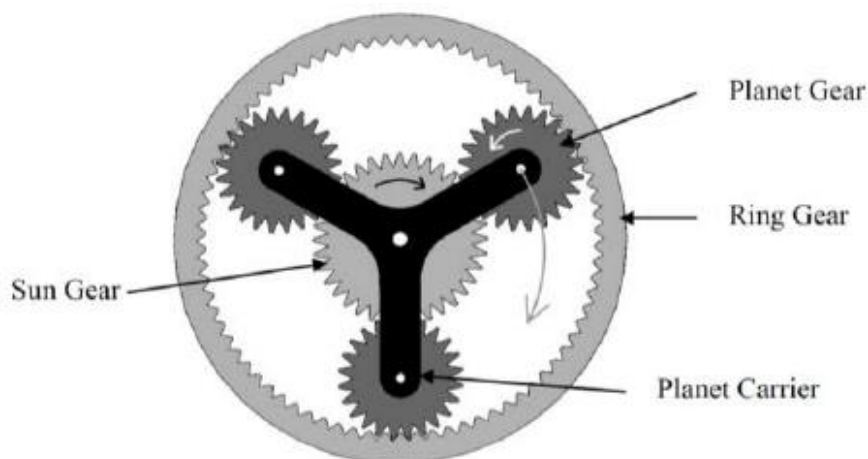
$S$  = počet zubů na „sun“;

$R$  = počet zubů na „ring“

$P$  = počet zubů na „planet“

základní vzorec : 
$$i = \frac{S}{R} + S$$

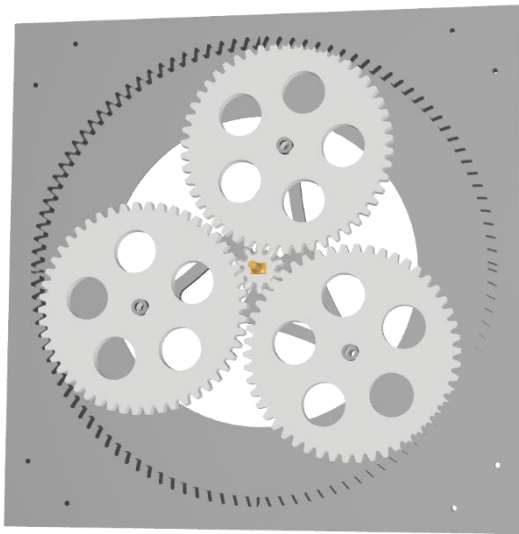
pokračující vzorec pro počet zubů : 
$$P = \frac{(R-S)}{2}$$



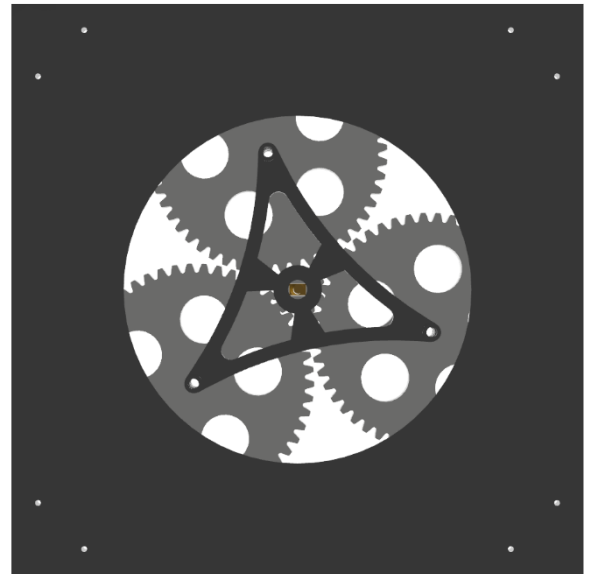
Obr.5.0. Planetární převodovka

Na konec jsem skončil s často využívaným poměrem a to je  $S = 10$ ,  $R = 110$  a  $P = 50$ . Jedná se tedy o větší počet zubů, podle obrázku (Obr.5.) jsem postupoval při navrhování .

Celé zařízení má fungovat tak, že se bude otáčet se sluncem, které bude točit planetami a úhel, který bude vznikat od počátečního bodu od začátku otáčení je počet hodin. Minuty jsou vytažené přímo ze slunce, viz (Obr.5.1. a Obr.5.2.).



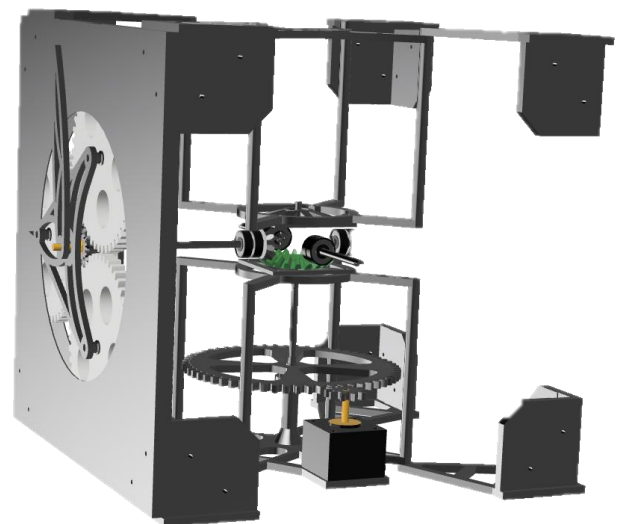
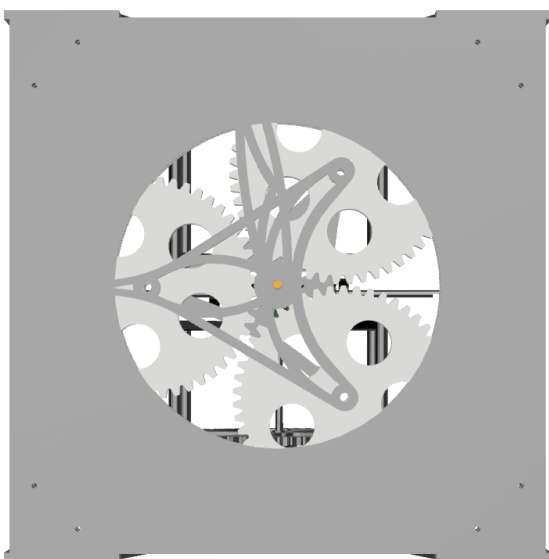
Obr.5.1.



Obr.5.2.

#### 1.2.4 Finální verze provedení sestavy

Nakonec jsem zkombinoval jednoduchost samotného převodu, tak rozvod od motoru do všech čtyřech směrů. Motor se nachází v spodní části celého projektu, elektronika je umístěna nahoře. Objekt bude uzavřen ze všech stran až na podstavu. Jedná se o krychli o rozměrech 296x296x296 mm. Na obrázku (Obr.6.) můžete vidět základní myšlenku a princip provedení, jedná se o celkovou sestavu všech dílů (uvedena v příloze). K ilustraci jsem zneviditelnil prvky typu „kryt na převodovku“ a ostatní ciferníky, aby byla vidět funkce a větší detail.

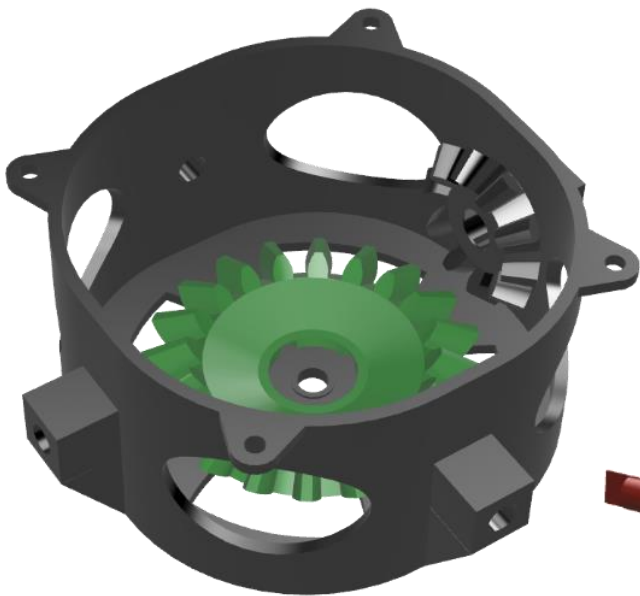


Obr.6.

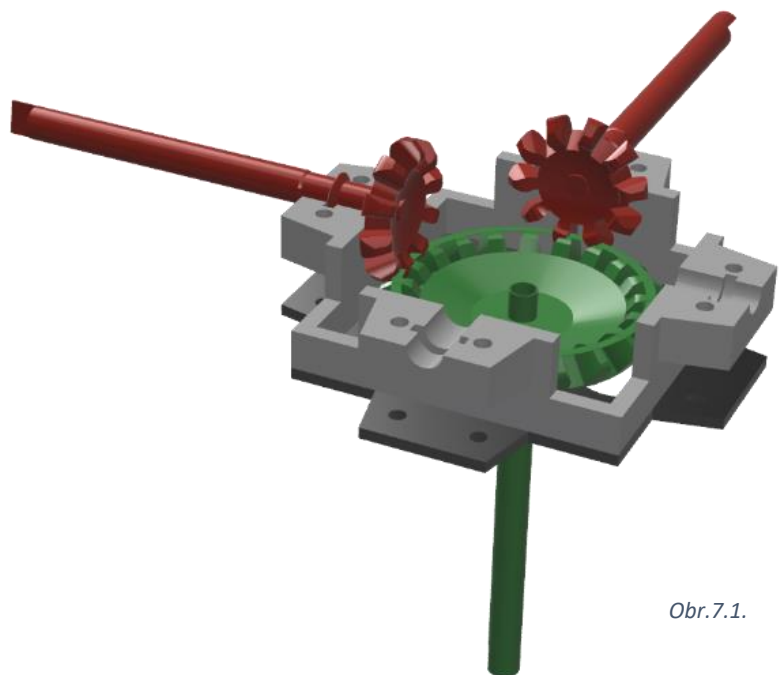
## 1.2.5 Synchronizace a převodovka

Jedním z nejobtížnějších prvků na tomto návrhu patří samotná převodovka, která má jeden vstup (zespoda) a další 4 výstupy s daným poměrem 1/2. Tyto hřídele dále pokračují na slunce u všech ciferníků. Tímto mechanismem jsem si zjednodušil ovládání a „ladění“ všech ciferníků současně. Nemůže nastat chvíle kdy by jeden ciferník ukazoval jinou hodnotu než druhý. Mají propojené hřídele, takže zde nemusím řešit synchronizaci.

Převodovku jsem tvořil v několika krocích. Nejprve jsem si vyzkoušel možnost využití 3D tisku pro 100% celého projektu. Po odladění jsem dospěl k verzi č.2. kde jsem měl podobný princip jen vše šlo lehčeji a fungovalo v rámci možností dobře. Potom jsem využil již vyzkoušeného nápadu s „půlenou“ převodovkou. Tedy objekt, který má dvě části, obě symetrické. Je to navrženo tak aby se daly všechny hřídele vyndat bez větších potíží. Ovšem přesto zde byly nedostatky. Původně první nápad s vylepšením (Obr.7.0.), rozdělovací převodovka (Obr.7.1.).



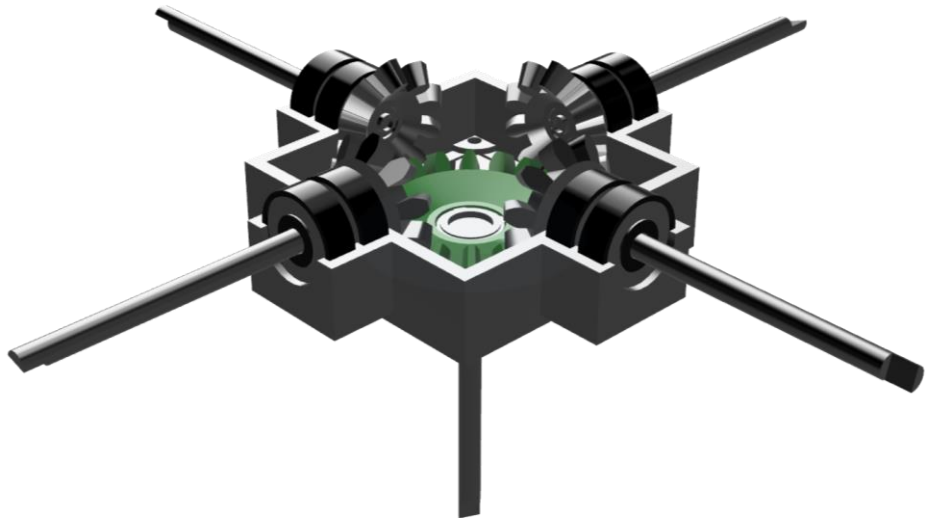
Obr.7.0.



Obr.7.1.

## 1.2.6 Finální verze převodovky

Následným vývojem a velkou řadou testů jsem došel k verzi kde využívám ložiska (model 635, 5mm vnitřní průměr). Všechny mé nápady měly průměr hřídele okolo 6 až 5.5 mm. Tento poslední model musel být užší aby se dalo využít tak malého ložiska, které jsem zakoupil. Je zde řada ložisek, jsou po dvou a mají tvořit rovinu a zároveň rozložení páky, kterou by mohla tvořit jakákoliv nerovnost zapříčiněna tiskem nebo možnou vůlí pro pohyb koleček. Kombinací dvou ložisek řeším spousty nedostatků a hlavně možnosti páky, které se projevovaly u předešlých, jednodušších verzích. Následný obrázek (Obr.7.2.) , opět jsem zneviditelnal pár prvků, které by překážely pro viditelnost. I zde jsem využil možnost použít vícero částí pro jednodušší zacházení s hřídelemi. Tato možnost je obohacena možnou výměnou nebo vyjmutím samotných ložisek.



Obr.7.2.

## 1.2.7 Hřídele

Všechny mé nápady obsahovaly stejný způsob, jak přenášet pohyb. Jedná se o hřídele, které jsou vytisknuté a mají na svých koncích pouze polovinu, takže dva konce do sebe zapadnou. Následnou součástí „svorkou“ je uzavřu do sebe. Zajistím tak větší odolnost proti zlomení a zlepším tak i tuhost. Kryt vytisknu na FDM tiskárně (Prusa MK3S). Hřídele jsou tvořeny na SLA tiskárně. (Obr.8.0. hřídel pro malé ozubené kolo) a (Obr.8.1. hřídel pro velké ozubené kolo). Bylo zapotřebí vytvořit zámek, pro bezpečné ukotvení ozubeného kola. Každé kolo má svůj hřídel (10x ozubené kolo (Obr.8.0) a 20x ozubené kolo (Obr.8.1.)). Jsou specifické svojí délkou a zámkem, který je navržen tak aby sedl co nejtěsněji do ozubeného kola.



*Obr.8.0. (10x ozubené kolo)*



*Obr.8.1.*

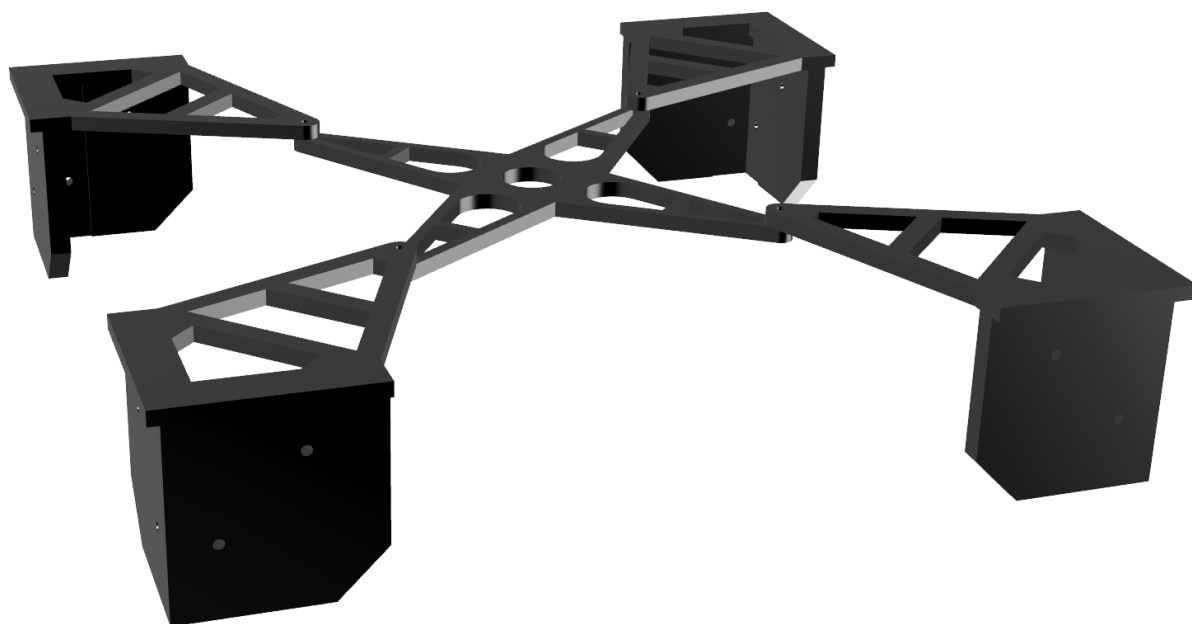
*(vrchní část, pouze pro symetrii)*



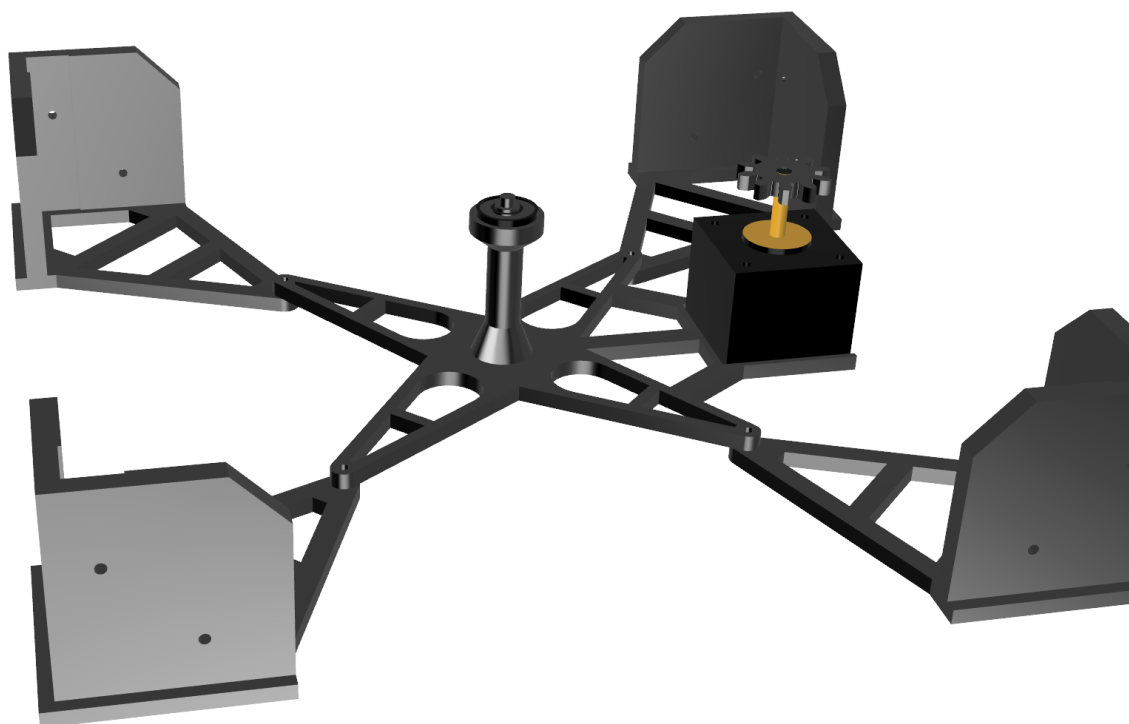
*Obr.8.1. (20x ozubené kolo)*

## **1.2.8 Spodní a horní část konstrukce**

Bylo zapotřebí navrhnout celistvou konstrukci. Nejlépe aby byla pevná a jednoduchá na sestavení. Jako první jsem začal s ciferníkem (viz. Obr.5.1. a Obr.5.2.). Po dokončení stran jsem potřeboval uchytit krokový motor a zajistit elektroniku. Tuto část úlohy řeší komponenty na spodní a na vrchní části konstrukce (Obr.9.1. a Obr.9.2.). Celá sestava se ještě skládá z „držák ciferník“. Tyto trojúhelníkovité díly mají za úkol spojit ciferníky a držet je dohromady v jednom kuse.



Obr.9.1. (vrchní část)



Obr.9.2. (spodní část)



## 1.3 Programování úvod

Začínal jsem tím, že jsem neměl kde a co vyzkoušet. Celý postup byl tedy na základě „prostě si to zkusit“. Po nasbírání informací o novém mikropočítači (ESP 32 WROOM). Následném naučením a pochopením, jak by měly takové hodiny fungovat. Co by se mělo upřednostňovat před jinou akcí atd...

### 1.3.1 Home pozice

Po obdržení krokového motoru jsem svou nešikovností zničil 3 drivery (špatné + a – na napájení). Po koupi dalších jsem již vše zapojil správně a motor začal fungovat přesně tak jak bylo požadováno. Aktuální verze má za úkol, zjistit jaký je čas v RTC, porovnat s ESP a popřípadě přepsat. Další částí programu je posun do dané polohy, jako první je ale třeba znát aktuální pozici. Což není reálné, protože NEM-17 nemá encoder pro určování absolutní / relativního umístění. Bylo zapotřebí určovat tak zvanou „home pozici“. Tento díl úlohy vyřešil spínač, který hlídá až se do polohy 12 hodin dostane hodinová ručička. Tedy má home pozice se nachází ve 12 hodinách. Pokaždé když vypadne napájení tak se hodiny po obnovení snaží dosáhnout tohoto stavu. Následně znovu odpočítat počet potřebných kroků k „nové“ pozici.

### 1.3.2 Přepočítání na čas / kroky

Tedy pokud má program svou home pozici tak přepočítá hodnotu čteného času z RTC na počet kroků pomocí vzorce (Obr.10.1. „prepcet24“). Do proměnné „SekundyK“ se počítá požadovaný počet kroků, který je následně nahrán do krokového motoru.

Číslo „6“ na konci řádku u počítání „SekundyK“. Jedná se o parametr, který jsem vypočítal pro optimální rotaci. Je odvozen z mnoha testů a přepočtů. Taktéž je tu jako zobrazení 6 sekund na krok.

```
void prepcet24()
{
  if (HodinyRTC > 12) HodinyRTC = HodinyRTC - 12;
  SekundyK = (( 60 * HodinyRTC + MinutyRTC) * 60 + SekundyRTC) / 6;
  Text(2, " HodinyRTC : ", HodinyRTC, " celkové sekundyK : ", SekundyK);
}
```

Obr.10.1. (prepcet24)

### 1.3.3 Čtení RTC

Pro čtení z ESP a zápis do RTC jsem vytvořil speciální funkční bloky (Obr.10.2.) . Využívají funkcí z knihovny („ErriezDS1302“ a „WiFiUDP“).

```
void cteniRTC(bool trida)
{
  byte hour; byte min; byte sec;
  rtc.getTime(&hour, &min, &sec);
  HodinyRTC = hour; SekundyRTC = sec; MinutyRTC = min;
  if(trida){Serial.print(hour);Serial.print(" : ");Serial.print(min);Serial.print(" : "); Serial.println(sec);}
}
```

Obr.10.2.

Poznámka: Proměnná „trida“, ve vstupu je zde pro možný zápis do sériového monitoru, nemá zde funkční podstatu ale vizualizační.

### 1.3.4 Zápis RTC

Ukázka programu (Obr.10.3.), který se stará o zápis hodnot z ESP, které je připojené k wifi. V ukázce můžeme vidět „ConnectionToWorld“, tato hodnota se mění v závislosti je-li zařízení připojené. Pokud nastane moment, kdy ztratí připojení každých 6 sekund je zde interval (asi 5 vteřin) pro snahu se připojit. Bez jakéhokoli přerušení mikropočítače. Vše je sestaveno na principu Millis().

Nastane-li chvíle kdy se připojí, změní se hodnota „ConnectionToWorld“ na logickou jedničku. Až v tento moment lze zapisovat do RTC. Do té doby se z něj mohlo pouze a jenom číst, to vše jen aby nedošlo k chybnému přepisu (čas by se mohl rovnat 00:00:00). Je zde další „ochranný“ prvek, který zamezuje zbytečným úpravám. Jedná se o část, kde program porovnává, zda nastala změna hodin nebo minut oproti ESP. Pokud ano, opět může nastat chvíle na úpravu dat dle internetu.

```
void zapisRTC() // funkce pro zápis času z ESP
{
  if (ConnectionToWorld == 1) // funkce pro zjištění zda je ESP připojeno k netu
  {
    TimeC.update();
    if(TimeC.getHours() != HodinyRTC )
    {
      rtc.setTime(TimeC.getHours(), TimeC.getMinutes(), TimeC.getSeconds());
      Serial.println(" _____zapis casu do RTC podle ESP _____");
    }
  }else {Serial.println("nelze zapsat to RTC, neni wifi, pouziji pamet");}
}
```

Obr.10.3.

### 1.3.5 ESP „připojování ke světu“

Podstatným dílem je funkce, která zařizuje samotné připojování (Obr.10.4.). Ve vrchním díle využívám možnosti na automatické připojení. Mám tu ochranu v podobě „IF“, která se snaží co nejméně vyrušovat ESP pokud je již připojené „if ( ConnectionToWorld == 0 )“. Následující „WHILE“, zde vytváří smyčku. V této časové době, která se počítá pomocí Millis() se ESP snaží připojit na WIFI. Za případu neúspěchu vytvoří chybovou hlášku a upraví proměnnou „ConnectionToWorld“ na 0. Při nadcházejících příležitostech se opět snaží připojit.

Spodní částí program vyhodnocuje, v jakém je ESP stavu. Zda je či není připojeno zapisuje do globální proměnné „ConnectionToWorld“.

```
void pripojovaniESP()
{
    WiFi.setAutoReconnect(true);
    WiFi.persistent(true);
    TimeC.setTimeOffset(posun*3600);

    if (ConnectionToWorld == 0)
    {
        while (WiFi.status() != WL_CONNECTED )
        {
            currentMil = millis();
            if(currentMil - startMil < doba1 - 100){Serial.print("");}
            else {break;}
        }
    }
    if(WiFi.status() == WL_CONNECTED )
    {
        Serial.println("Connected to the WiFi network");
        ConnectionToWorld = 1;
    }else {
        Serial.println(" bez netu !!!");
        ConnectionToWorld = 0;
    }
}
}
```

Obr.10.4.

## 1.4 Samotná sestava

K úvodu do této kapitoly bych rád zmínil, že většinu projektu jsem vytvářel dlouho dopředu. Jedná se o prvky konstrukce, programu a mechanických dílů. Celá sestava mohla být uskutečněna pouze za podmínky, že všechny díly jsou na 100% připravené k usazení bez velkých změn.

### 1.4.1 Tisk

Z počátku jsem začal s tiskem prvního ciferníku, který mě měl ujistit, že vše, co je zatím navrženo tak bude fungovat. Po této jistotě jsem začal tisknout ozubená kola. Zároveň jsem ve škole tiskl rám pro celý objekt, byly to dlouhé hodiny tisku, musím zde podotknout, že nenastala situace, kdy by se tisk nepovedl a materiál by byl vyhozen, chybou tiskárny.

Postupně jsem tiskl ostatní díly. Dostal jsem se do fáze, kde je třeba nastavovat rozteče u hřídelí, které rozvádějí pohybovou sílu od převodovky na ciferníky. V následujícím kroku budu přidávat senzor pro určování pozice („jazýčkové relé“).

Při tisku jsem v CAD programu vytvořil rezervu pro ozubená kola v ciferníku. Tato úprava byla nakonec změněna. První verze měla výšku stejnou jako je výška konstrukce. Druhá již měla zmenšenou výšku a k tomu přidanou rezervu, aby se zmenšila třecí plocha. Výsledkem bylo zbytečné vyvracování v ozubení, což způsobovalo nechtěné záseky a možnost zničit hřídele. Vrátil jsem se tedy zpět k původnímu návrhu, kde jsou kola větší, než drážka pro ně určena.

## 1.5 Závěr

Jsem rád že jsem práci mohl vytvořit. Mechanické provedení se, podle mého, vydařilo do jisté míry vychytat k funkčnímu stavu. Dalším bodem jsem dokázal využít nový mikro počítač a naučit se s ním. V budoucnu budu velmi pravděpodobně využívat těchto těžce získaných dovedností, a to jak ve škole, tak v osobním čase. Konstrukce byla jednou z největších CAD prací v mém životě, modeloval jsem již pár větších projektů, ale tento mi dal nejvíce. Od řešení mechaniky a fyziky až po navrhování přesných dílů na určitá místa. Určitě mě tato práce posunula v daném oboru vpřed. Jak k automatizaci a uvědomění si základních myšlenek tak i praktickou částí.

Zařízení je schopné vyrovnat se s výpadkem proudu, pomocí programu, jež jsem sestavil. Hodiny jsou z velké části tisknutelné. Práce na takovémto rozmanitém a vcelku komplikovaném projektu mě těšila a doufám, že budu mít ještě příležitost vytvářet obdobné projekty, které vyžadují moji 100% soustředěnost.

## **Obrázkové odkazy :**

Obr.1 ESP32 WROOM -

<https://www.b4x.com/android/forum/proxy.php?image=https%3A%2F%2Fimg.olx.com.br%2Fimages%2F18%2F186902002392514.jpg&hash=41e2c44881d2873c2a4dc2c4121439ea>

Obr.2 RTC DS1302 - <https://www.hadex.cz/img/zbozi/m481b.jpg>

Obr.3. A4988 driver - <https://wiki.mchobby.be/images/2/22/A4988-PinOut.jpg>

Obr.4. krokový motor NEM-17 - <http://www.eu.diigit.com/image/cache/catalog/pololu/A00612-1000x750.jpg>

Obr.5.0. Planetární převodovka - <https://www.researchgate.net/profile/Wade-Smith/publication/260290993/figure/fig1/AS:297150190768139@1447857535008/Basic-layout-of-planetary-gearbox-5.png>

Obr.5.1. fotodokumentace – vlastní zdroj (Fusion 360)

Obr.5.2. fotodokumentace – vlastní zdroj (Fusion 360)

Obr.6. fotodokumentace – vlastní zdroj (Fusion 360)

Obr.7. fotodokumentace – vlastní zdroj (Fusion 360)

Obr.8. fotodokumentace – vlastní zdroj (Fusion 360)

Obr.9. fotodokumentace – vlastní zdroj (Fusion 360)

Obr.10. fotodokumentace – zdroj kód (Arduino)

## **Textové odkazy :**

Od.1. AccelStepper dok. - <https://www.airspayce.com/mikem/arduino/AccelStepper/classAccelStepper.html>

## **Využitá „literatura“ :**

ESP připojení :[https://randomnerdtutorials.com/solved-reconnect-esp8266-nodemcu-to-wifi/?fbclid=IwAR11DC3\\_oGvOjn0OYdiGoZIEe-LSqVYCLRM8pJ9915B-ppg11LhZaLOrb0w](https://randomnerdtutorials.com/solved-reconnect-esp8266-nodemcu-to-wifi/?fbclid=IwAR11DC3_oGvOjn0OYdiGoZIEe-LSqVYCLRM8pJ9915B-ppg11LhZaLOrb0w)

ESP vstupy/výstupy : <https://microcontrollerslab.com/esp32-pinout-use-gpio-pins/>

**Návrh konstrukce a programu : Janoušek Tomáš**