|  |  |  |
| --- | --- | --- |
|  | | |
| **Závěrečná studijní práce**  **dokumentace** | | |
| **Automatické blinkry na motocykl** | | |
| David Dumbrovský | | |
|  | | |
|  | |  |
| **Obor:** | 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  se zaměřením na počítačové sítě a programování | |
| **Třída:**  **Školní rok:** | IT4  2020/2021 | |

#### Poděkování

*Děkuji panu učiteli Ing. Petru Grussmannovi za cenné rady a pomoc při vybírání součástek.*

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité   
informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým účelům na Střední průmyslové   
a umělecké škole v Opavě, Praskova 399/8.

V Opavě 31. 12. 2020

*podpis autora práce*

**ANOTACE**

Projekt se zabývá tvorbou automatických blinkrů na motocykl. Skládá se z hardwarové a softwarové části. Základ hardwarové části tvoří čip MPU-6050 a vývojová deska Arduino Nano. Po stisknutí tlačítka se zapne program, který co půl vteřiny mění stav LE diody a zároveň kontroluje aktuální polohu pomocí tříosého gyroskopu, kterou dočasně zapisuje do pomocné proměnné, kterou program současně porovnává s minulou hodnotou. Je-li rozdíl těchto hodnot větší než možná odchylka, program se vypne. Program lze vypnout i opětovným stisknutím tlačítka. Programová část automatických blinkrů je řešena v jazyce C++ s pomocí knihoven Arduino.h a Wire.h.

**Klíčová slova**

Arduino, MPU-6050, gyroskop, automatizace

OBSAH

[Úvod 5](#_Toc61215819)

[1 VÝROBA automatických blinkrů 6](#_Toc61215820)

[2 Využité technologie 7](#_Toc61215821)

[2.1 Hardware 7](#_Toc61215822)

[2.1.1 Seznam součástek 7](#_Toc61215823)

[2.1.2 Arduino Nano V3.0 8](#_Toc61215824)

[2.1.3 Tříosý gyroskop a akcelerometr MPU-6050 8](#_Toc61215825)

[2.1.4 LED (*Light-Emitting Diode*) 9](#_Toc61215827)

[2.1.5 Tlačítko 9](#_Toc61215828)

[2.2 Software 10](#_Toc61215829)

[2.2.1 PlatformIO v2.2.1 10](#_Toc61215830)

[2.2.2 VS Code v1.52.1 10](#_Toc61215831)

[2.2.3 Knihovna Arduino 10](#_Toc61215832)

[3 Způsoby řešení a použité postupy 11](#_Toc61215833)

[3.1 Zjišťování odchylky gyroskopu 11](#_Toc61215834)

[3.2 Popis fungování automatických blinkrů 12](#_Toc61215835)

[4 Výsledky řešení 13](#_Toc61215836)

[Závěr 14](#_Toc61215837)

[Seznam použitýCH INFORMAČNÍCH ZDROJů 15](#_Toc61215838)

Úvod

Rozhodl jsem se vytvořit automatické blinkry na motocykl, protože motorky jsou můj koníček a nejednou se mi stalo, že jsem jezdil s blikajícím směrovým světlem. Neřekl bych, že se to stává jen mi, jelikož jsem už i na pár jedinců motorkářů narazil, kteří blikali, ale neodbočovali. Je to vskutku nebezpečné, jelikož za vámi může jet nějaký pirát a srazit vás v křižovatce.

Cílem této práce bylo vytvořit funkční automatické blinkry, které po návratu motocyklu do původní polohy zhasnou. Dalšími prioritami byla nízká cena, jednoduchost a co největší spolehlivost.

Ve své dokumentaci zmiňuji použité technologie a blíže popisuji princip fungování automatických blinkrů. Zmiňuji se o problémech, se kterými jsem se setkal při vývoji, popisem technologií nezbytných k jeho výrobě i k jeho funkčnosti.

# VÝROBA automatických blinkrů

První, co bylo třeba udělat, bylo vybrat správnou vývojovou desku. Protože již dřív jsem trochu experimentoval s deskou Arduino Nano založenou na mikrokontroléru ATmega328 od firmy Atmel a jedna mi ležela doma, byla to pro mě jasná volba.

Jelikož jsem s většinou součástek již pracoval, tak jsem moc nedbal na bezpečnost a trochu se mi to vymstilo. Po prvním nahrání programu se mi z Arduina začalo kouřit a trochu smrdět, tak jsem věděl, že je něco špatně. Proto jsem si objednal dvě další zároveň s čipem MPU-6050. Jakmile balíček přišel, čekalo mě další překvapení. Žádné ze zařízení nemělo připájené piny, takže jsem si je musel dopájet sám se svými nulovými zkušenostmi s pájením. Po dopájení jsem zařízení zapojil a zkopíroval z internetu nějaký program na odzkoušení a fungoval.

S dalším zkoušením součástek jsem byl už opatrnější. Ze začátku jsem si to rozdělil na dvě části. Jedná část, kde jsem si udělal obyčejné blinkry, jako jsou na motorce, to znamená dvě tlačítka a dvě LE diody. Druhá část byl zapojený čip MPU-6050, na kterém jsem se učil s gyroskopem. Po seznámení se s knihovnou Wire.h jsem zjistil, že každý čip má nějakou svou odchylku, na kterou jsem musel přijít, po chvíli hledání jsem konečně našel kód, který fungoval i pro mé zařízení. Následně jsem tyto dvě části spojil dohromady a začal programovat.

# Využité technologie

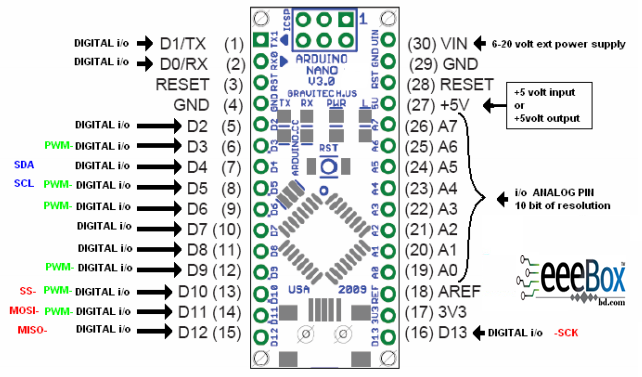
## Hardware

### Seznam součástek

* Vývojová deska Arduino V3.0,
* Tříosý gyroskop a akcelerometr MPU-6050,
* 2x žlutá LED,
* 2x tlačítko,
* 2x rezistor 330 Ω,
* 2x rezistor 6k8 Ω,
* Nepájivé pole

### Arduino Nano V3.0

Základ projektu tvoří vývojová deska Arduina Nano V3.0 (obrázek č. 1). Funkčně je shodná s verzí Arduino UNO, je ale přízpůsobená pro zasunutí do nepájivého kontakního pole. Je založena na mikrokontroléru AVR ATmega 328. Arduino Nano V3.0 obsahuje 14 GPIO digitálních pinů a 8 analogových pinů (obrázek č. 2). Pro program je k dispozici paměť o velikosti 32 kB.

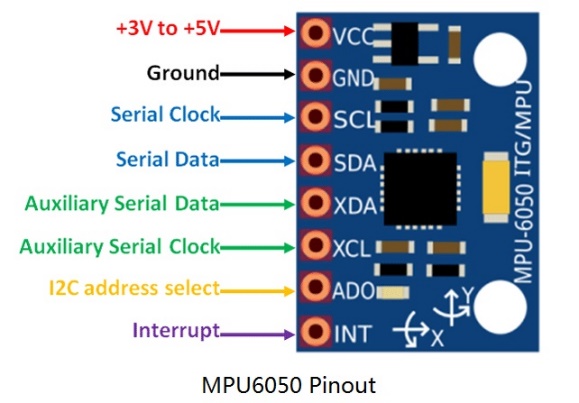


*Obrázek č. 2 Arduino Nano V3.0 pinout*

*Obrázek č. 1 Arduino Nano V3*

### Tříosý gyroskop a akcelerometr MPU-6050

Modul obsahuje 3osý akcelerometr a také 3osý gyroskop (MEMS). Je možné jej připojit ke všem vývojovým kitům Arduino/Genuino a také spoustě dalších (např. Raspberry PI) – podmínkou je možnost komunikace přes rozhraní I**²**C. Modul obsahuje DMP (digital motion processor). Ten v reálném čase přepočítává naměřené hodnoty akcelerometru a gyroskopu na jednotlivé složky rotačního pohybu v ose X, Y a Z.



### 

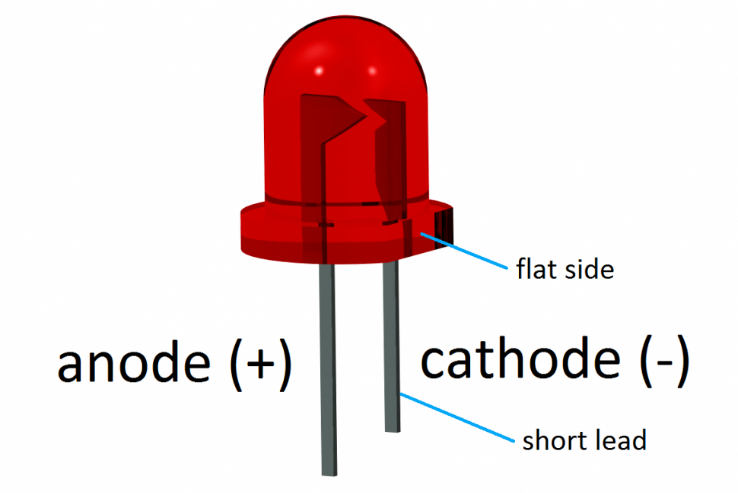
### 

*Obrázek č. 4 MPU-6050 pinout*

*Obrázek č. 3 MPU-6050*

### LED (*Light-Emitting Diode*)

LED je v elektrotechnice označení pro diodu, která emituje světlo, případně infračervené nebo ultrafialové záření, čímž se liší od standardní diody. LED vyzařuje z obnaženého PN přechodu a vede stejnosměrný proud pouze jedním směrem (obrázek č. 5). Na rozdíl od žárovky dosahuje vysoké účinnosti, je mechanicky odolná, levná na výrobu, a proto je čím dál více využívána.



*Obrázek č. 5 LE dioda*

### Tlačítko

Tlačítko je jednoduchý spínač, který slouží k ručnímu ovládání elektrického zařízení. Hmatník tlačítka je obvykle vyroben z tvrdého materiálu a tvar je přizpůsoben tak, aby bylo možné tlačítko obsluhovat tlakem prstu nebo rukou. Hlavní odlišnost tlačítka od ostatních spínačů je v tom, že nemá aretaci. Po oddálení prstu nebo ruky se vrací do původní polohy.

*Obrázek č. 6 Tlačítko*



## Software

### PlatformIO v2.2.1

PlatformIO je postavené na Visual Studio Code od Miscrosoftu a je dostupné na všech platformách. Nabízí všechny základní funkce z Arduino IDE, ale navíc povětšinou zachází ještě dále. Kromě běžných funkcí pro psaní kódu, nahrávání do Arduina a terminálu nabízí PlatformIO mnoho funkcí navíc.

* Podporu 689 mikrokontroléru na 31 platformách (vč. Arduina)
* Doplňování kódu (IntelliSense)
* Lepší správu knihoven
* Podporu GITu
* Více vzhledů vývojového prostředí
* Podporu pluginů
* Z mé zkušenosti i rychlejší build nahrávání programů do Arduina

### VS Code v1.52.1

Visual Studio Code je editor zdrojového kódu vyvíjený společností Microsoft pro operační systémy Windows, Linux a macOS. Obsahuje podporu Git, zvýraznění syntaxe, kontextový našeptávač a podporu pro ladění a refaktorizaci. Zdrojový kód je svobodný software pod licencí MIT. Sestavené binárky nabízené přímo Microsoftem jsou freewarem obsujícím telemetrii, ale existuje i komunitně sestavovaná varianta VSCodium. Editor je naprogramovaný v JavaScriptu a TypeScriptu.

Tento editor jsem si vybral, jelikož s ním pracuji již od prvního ročníku, a přijde mi velmi přehledný, intuitivní a jde do něho stáhnout dalších mnoho rozšíření. Také je to jediný editor, který podporuje rozšíření PlatformIO.

### Knihovna Arduino

Knihovna sloužící k ovládání hardwarových součástek v jazyce C++.

# Způsoby řešení a použité postupy

## Zjišťování odchylky gyroskopu

Nejdříve, co jsem musel udělat, bylo zjistit odchylku. Na to jsem si na internetu našel funkci, kterou jsem použil. Odchylka byla měřena z 200 hodnot (Obrázek č. 7).

**void** calculate\_IMU\_error() {

  while (c < 200) {

    Wire.beginTransmission(MPU);

    Wire.write(0x3B);

    Wire.endTransmission(false);

    Wire.requestFrom(MPU, 6, true);

    AccX = (Wire.read() << 8 | Wire.read()) / 16384.0 ;

    AccY = (Wire.read() << 8 | Wire.read()) / 16384.0 ;

    AccZ = (Wire.read() << 8 | Wire.read()) / 16384.0 ;

    AccErrorX = AccErrorX + ((atan((AccY) / sqrt(pow((AccX), 2) + pow((AccZ), 2))) \* 180 / PI));

    AccErrorY = AccErrorY + ((atan(-1 \* (AccX) / sqrt(pow((AccY), 2) + pow((AccZ), 2))) \* 180 / PI));

    c++;

  }

  AccErrorX = AccErrorX / 200;

  AccErrorY = AccErrorY / 200;

  c = 0;

  while (c < 200) {

    Wire.beginTransmission(MPU);

    Wire.write(0x43);

    Wire.endTransmission(false);

    Wire.requestFrom(MPU, 6, true);

    GyroX = Wire.read() << 8 | Wire.read();

    GyroY = Wire.read() << 8 | Wire.read();

    GyroZ = Wire.read() << 8 | Wire.read();

    GyroErrorX = GyroErrorX + (GyroX / 131.0);

    GyroErrorY = GyroErrorY + (GyroY / 131.0);

    GyroErrorZ = GyroErrorZ + (GyroZ / 131.0);

    c++;

  }

  GyroErrorX = GyroErrorX / 200;

  GyroErrorY = GyroErrorY / 200;

  GyroErrorZ = GyroErrorZ / 200;

  Serial.print("AccErrorX: ");

  Serial.println(AccErrorX);

  Serial.print("AccErrorY: ");

  Serial.println(AccErrorY);

  Serial.print("GyroErrorX: ");

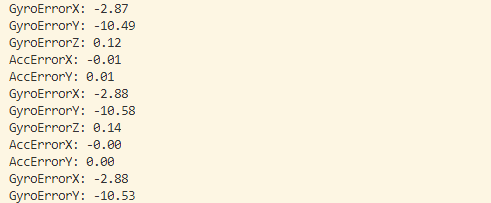
  Serial.println(GyroErrorX);

  Serial.print("GyroErrorY: ");

  Serial.println(GyroErrorY);

  Serial.print("GyroErrorZ: ");

  Serial.println(GyroErrorZ);

}

*Obrázek č. 7 Výsledné hodnoty*

## Popis fungování automatických blinkrů

Po stisknutí tlačítka se zapne program, který každé půl vteřiny zjišťuje stav gyroskopu a porovnává s minulou hodnotou. Jestli se nahne, nastaví se první stav na true, jestli se nahne zpátky na druhou stranu, nastaví se druhý stav také na true. Jakmile je druhý stav nastavený na true a zároveň rozdíl mezi hodnotami se blíží nule, program se vypne.

if(pS){

      digitalWrite(LEDP, !digitalRead(LEDP));

      if(roll > rollTmp + 1 && !stateP){

        stateP2 = false;

        stateP = true;

      }

      else if (roll < rollTmp -1 && stateP){

        stateP2 = true;

        stateP = false;

      }

      else if ((roll < rollTmp +0.1 && roll>rollTmp - 0.1)&&stateP2){

        stateP2=false;

        pS=false;

        zhasni();

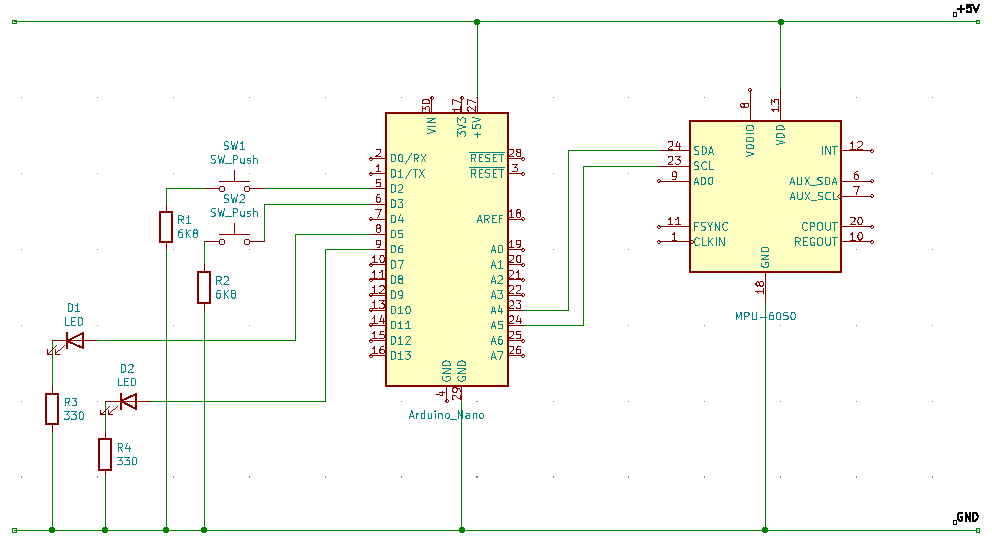
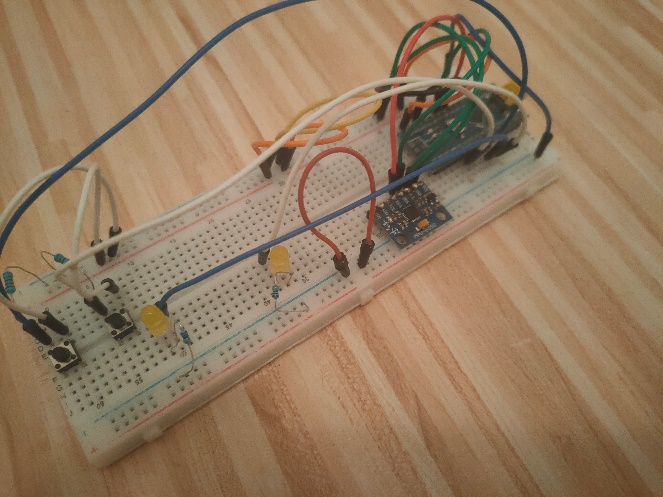
      }

      rollTmp = roll;

    }

# Výsledky řešení

Automatické blinkry jsou v současné době plně funkční, po několika pokusech testování splňuje mé počáteční očekávání. Do budoucna bych to chtěl zkusit připojit na motocykl a zkoušet již na něm. Bylo by vhodné vytvořit pevný plošný spoj na pájivém poli.



*Obrázek č. 8 Zapojení na nepájivém poli*

*Obrázek č. 9 Schéma*

# **Závěr**

Cílem projektu bylo vytvořit funkční model automatických blinkrů.

Tento cíl byl splněn, zařízení splňuje všechny moje požadavky. To znamená, že po stisknutí se zapne program, který mění stav LE diody a zároveň kontroluje stav gyroskopu a ukládá do proměnné, kterou porovnává s minulou hodnotou. Je-li rozdíl větší než možná odchylka, program se vypne. Program lze také vypnout pomocí opětovným stisknutím tlačítka.

Do budoucna bych si chtěl pořídit nějakou starší motorku, na které bych toto zařízení mohl vyzkoušet, popřípadě nechat ho tam.

Zdrojový kód je k dispozici na mém Githubu https://github.com/dumbricek/final\_project

Seznam použitýCH INFORMAČNÍCH ZDROJů

[] *Návod k čipu MPU-6050* [online]. [cit. 2021-01-10]. Dostupné z: https://navody.dratek.cz/navody-k-produktum/gyroskop-akcelerometr-a-magnetometr-mpu-9250.html

[2] *Návod k čipu MPU-6050* [online]. 2017 [cit. 2021-01-10]. Dostupné z: https://howtomechatronics.com/tutorials/arduino/arduino-and-mpu6050-accelerometer-and-gyroscope-tutorial/

[3] *Návod k čipu MPU-6050* [online]. 2017 [cit. 2021-01-10]. Dostupné z: https://www.youtube.com/watch?v=UxABxSADZ6U&ab\_channel=HowToMechatronics

[4] *Seznámení se s Arduinem* [online]. 2016 [cit. 2021-01-10]. Dostupné z: https://navody.dratek.cz/zaciname-s-arduinem/