



# Středoškolská technika 2017

Setkání a prezentace prací středoškolských studentů na ČVUT

## Mirrorbot

Roman Ludvík, Petra Melicharová

Gymnázium Jindřicha Šimona Baara  
Domažlice

# Problém

Naším cílem bylo sestavit robota, který by kopíroval pohyby člověka. Takový robot by mohl zastávat přesnou práci lidských rukou na pro člověka nebezpečných místech. Takový robot by se dal využít například pro stavbu vesmírných stanic.

Chtěli jsme k tomu použít pouze cenově dostupnou elektroniku. Pro konstrukci jsme se snažili volit levné a snadno dostupné materiály a součástky. Ačkoliv máme s projektem velké plány do budoucna, cílem stavby tohoto prototypu bylo pouze navrhnout algoritmus a vyzkoušet jeho fungování na našem robotovi.

# Hypotéza

Nejdříve jsme se museli rozhodnout, jakou elektroniku použijeme pro řízení našeho robota. Vzhledem k tomu, že jsem minulý rok spolupracoval na stavbě 3D tiskárny řízené Arduinem, rozhodl jsem se využít svých zkušeností a postavit Mirrorbota na podobném principu.

Mirrorbot je proto řízen Arduinem Mega se Shieldem RAMPS 1.4, které ovládá sestavu krokových motorů.

Pro získávání dat o pohybech operátora jsme rozhodli využít gyroskopů MPU, taktéž řízených Arduinem.

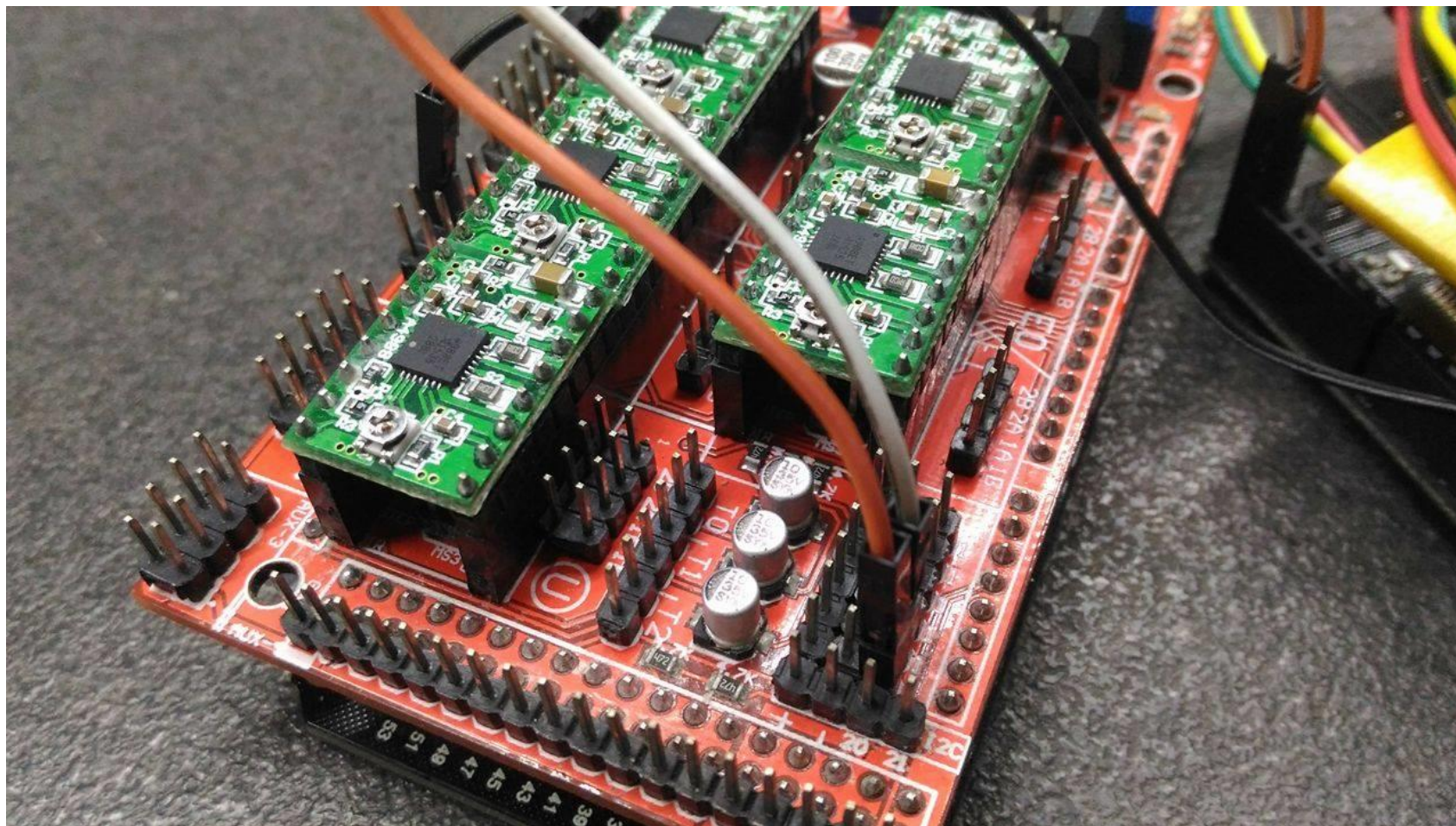
# Materiály

Řízení a mechanická část byly inspirovány mým předchozím projektem 3D tiskárna, na kterém jsem spolupracoval s Jaroslavem Ihou a Danielem Hromádkem. Princip snímání pohybů byl inspirován prací Ergonomie práce na počítači, autora Lukáše Forsta.

# Mirrorbot - model 1

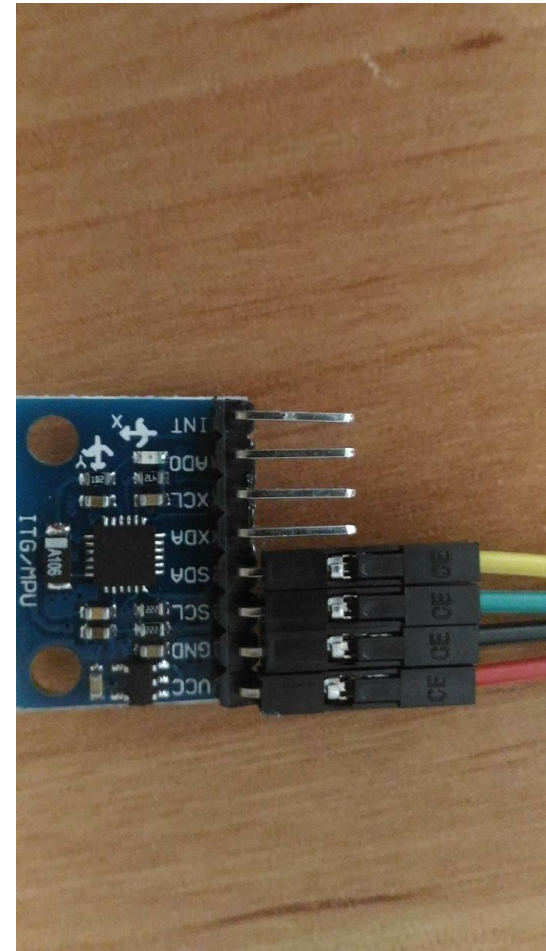


# Arduino Mega se Shieldem RAMPS

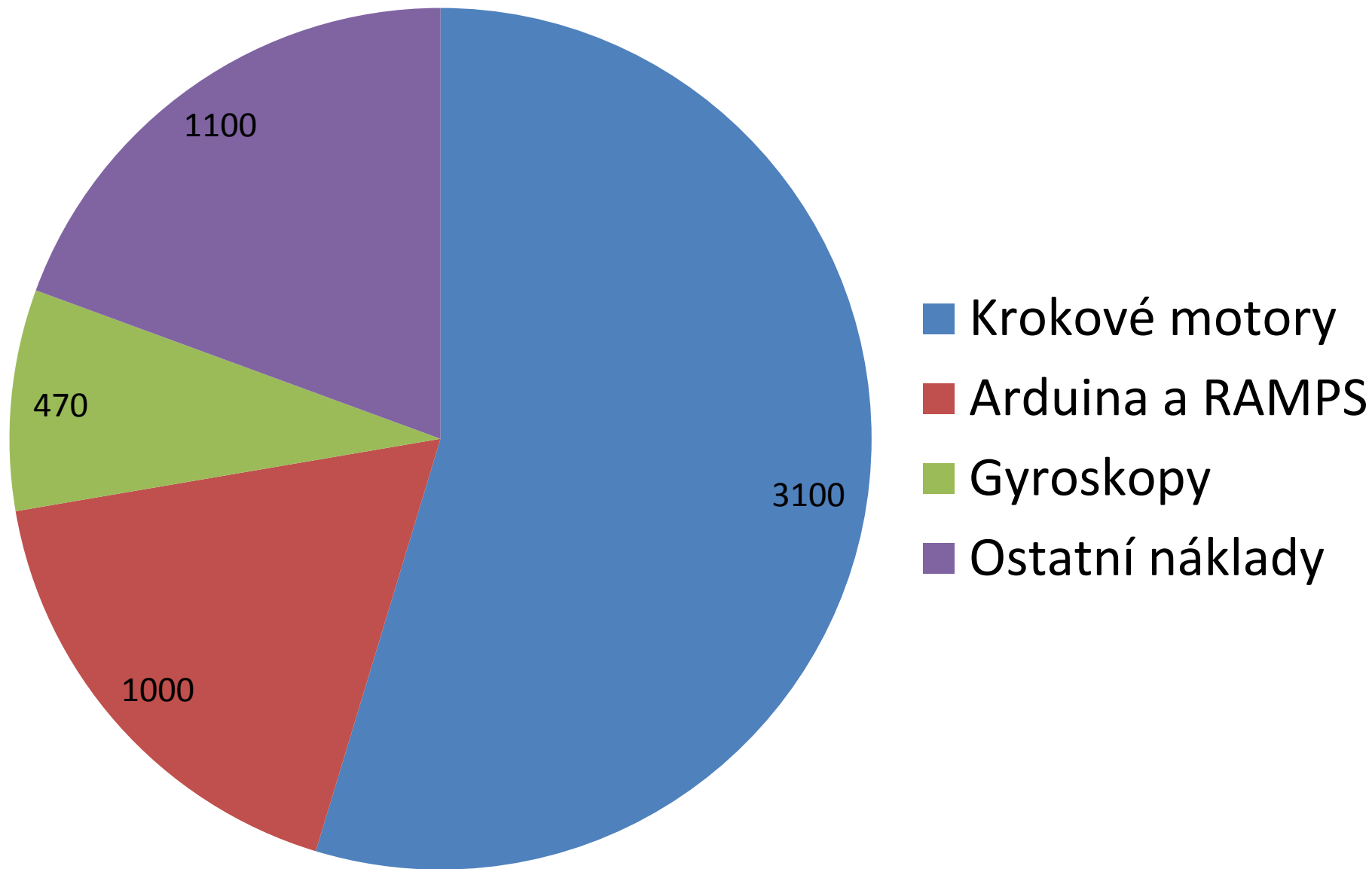




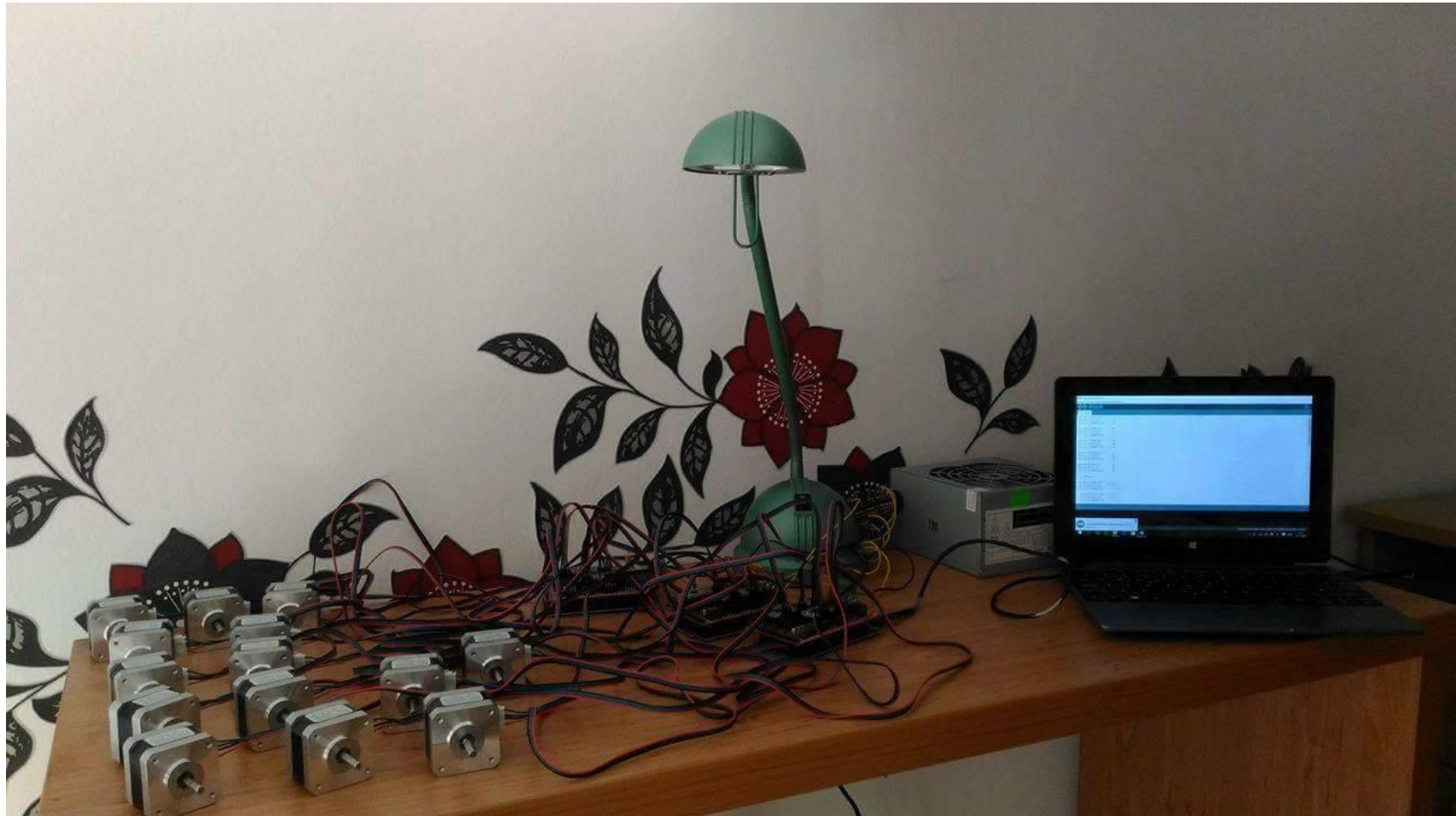
# Krokový motor a gyroskop



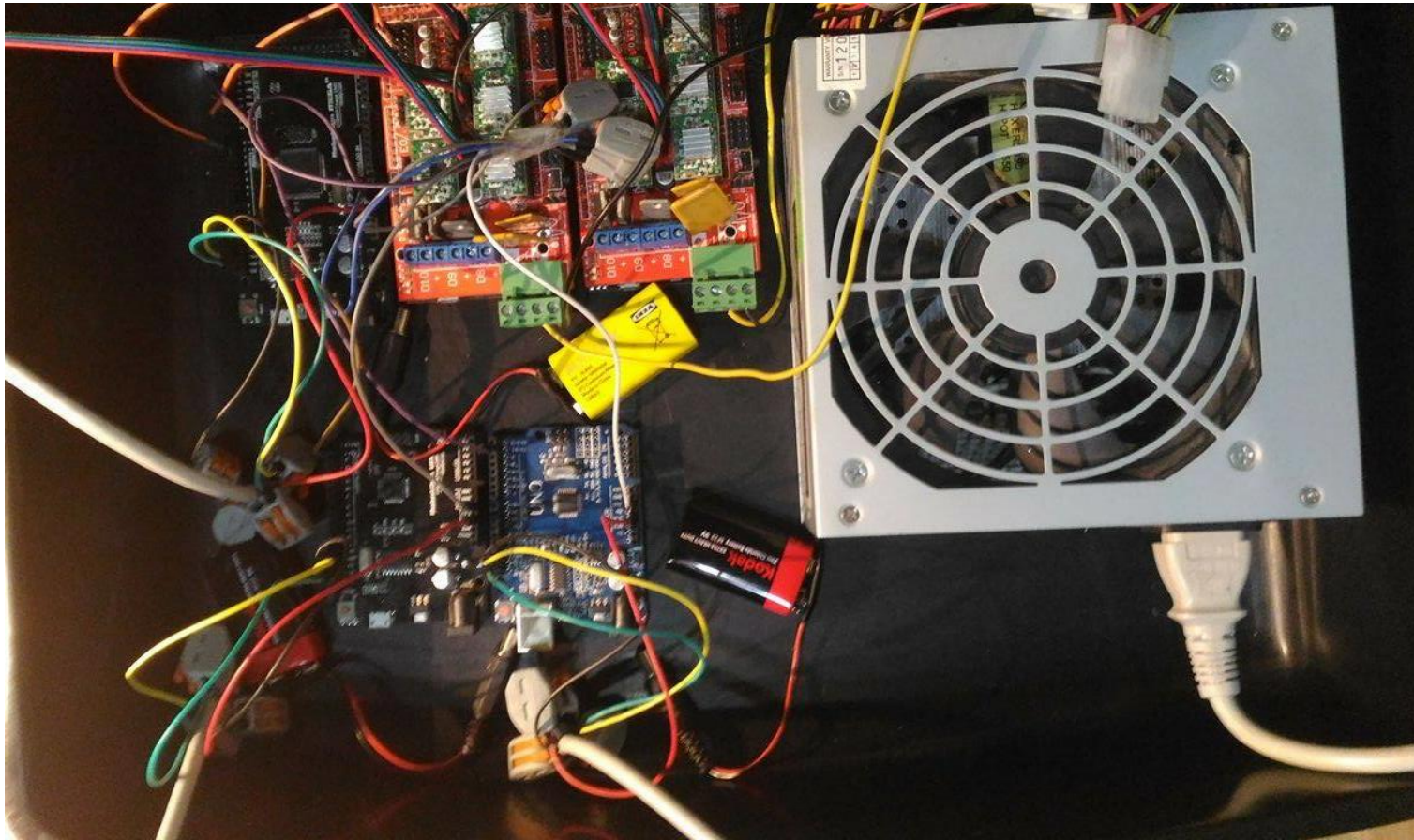
# Náklady



# Testování řízení krokových motorů pomocí Arduina



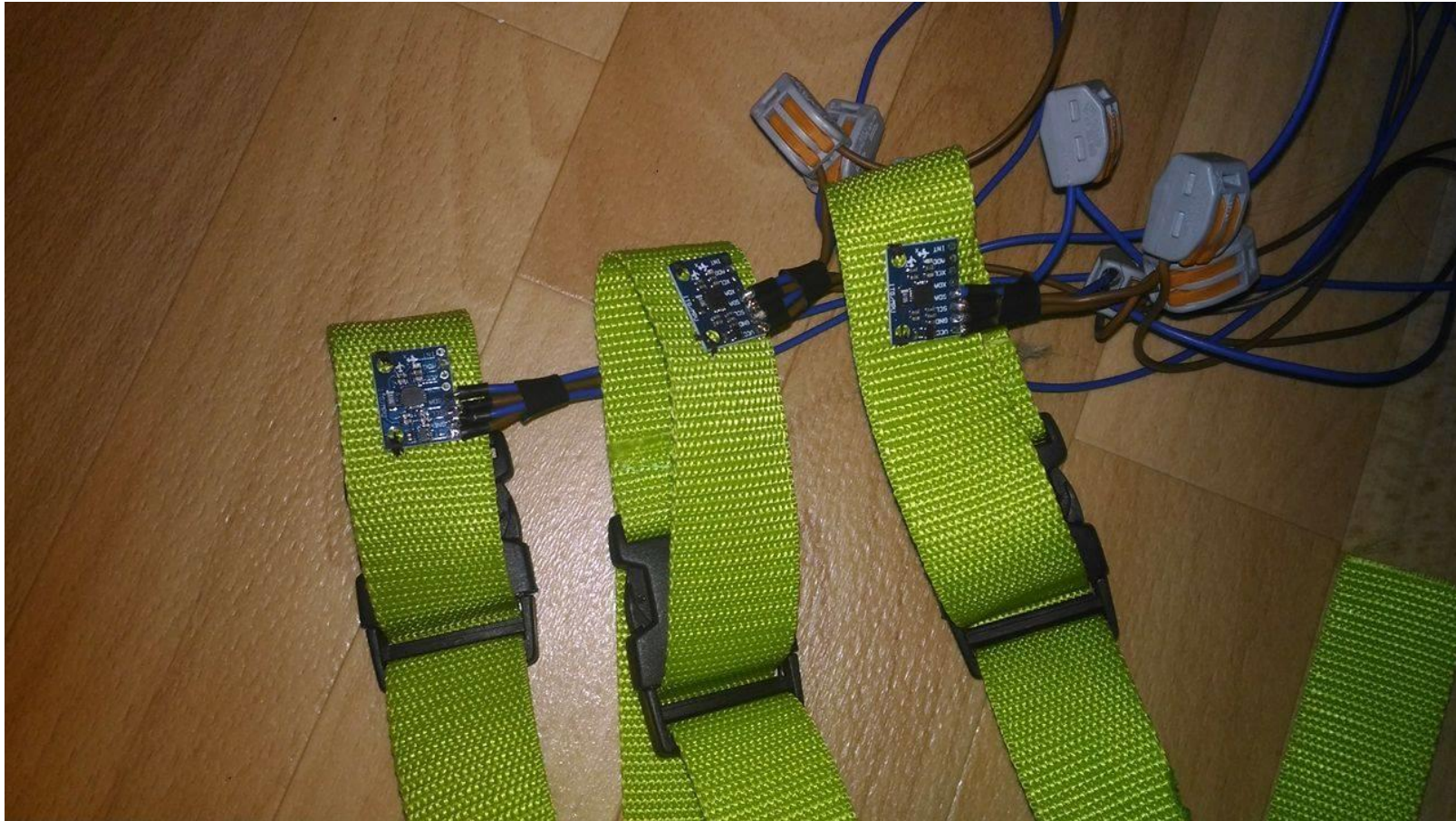
# Zapojení sériových linek mezi Arduiny



# Kladky a krokové motory pro pohyb paží Mirrorbota



# Návlek s gyroskopy



# Závěr

## Plány do budoucna a možná využití

V blízké době se pokusíme docílit asynchronního pohybu rukou. Dále bychom našemu robotovi rádi přidali ještě nohy či kola a také otáčení kamery podle otočení hlavy uživatele. Pro lepší pohled z první osoby by také bylo možné přenášet obraz z kamery do brýlí pro virtuální realitu.

Klíčové však bude hlavně vytvořit celého robota z pevnějších materiálů a se silnějším pohonem. Například ocelovou konstrukci poháněnou pneumatickými písty. V takovém případě by Mirrorbot mohl zasahovat a pracovat na nebezpečných místech jako jsou bojiště, vesmír nebo mořské dno.



## Postup práce

Na začátku jsme potřebovali rozpohybovat motory v závislosti na náklonech gyroskopů. Prvním krokem bylo otestování řízení motorů a získávání dat z gyroskopů. První problém nastal, když jsme zkoušeli otáčet motory na základě těchto dat. Docházelo k velmi intenzivnímu vzájemnému rušení motorů a gyroskopů.

Kvůli tomu bylo nutné připojit gyroskopy na oddělené Arduino, a pak po sériové lince odesílat pouze jednoznakové příkazy.

Dalším krokem bylo navržení hlavního algoritmu, který kombinuje získávání hodnot gyroskopů, sériovou komunikaci a následné ovládání motorů. Algoritmus byl zapsán do několika samostatných programů a nahrán na jednotlivé desky Arduino.

Když jsme měli připravený celý algoritmus, začali jsme s navrhováním a samotnou konstrukcí robota. Zvolili jsme konstrukci ze dřeva a plastu. Motory zvedají plastové paže pomocí kladek. Senzory jsme umístili na speciální návlek. Když byla celá konstrukce sestavená, bylo třeba doladit hodnoty, aby byly pohyby přesné. Toho jsme dosáhli postupným upravováním proměnných násobičů a následným testováním přesnosti pohybu.

# Program - gyroskop

```
void loop() {  
  
    while (fifoCount < packetSize) fifoCount = mpu.getFIFOCount();  
  
    mpu.getFIFOBytes(fifoBuffer, packetSize);  
    fifoCount -= packetSize;  
  
    mpu.dmpGetQuaternion(&q, fifoBuffer);  
    mpu.dmpGetGravity(&gravity, &q);  
    mpu.dmpGetYawPitchRoll(x, &q, &gravity);  
  
    new_x=(x[1] * 180/M_PI);  
    final_x = (new_x-old_x);
```

```
    if(final_x>=3)  
    {  
        while(final_x>=3)  
        {  
            Serial.println('A');  
            final_x=(final_x-3);  
        }  
        Serial.println('O');  
        old_x=new_x;  
    }  
    else if (final_x<=(-3))  
    {  
        while(final_x<=(-3))  
        {  
            Serial.println('B');  
            final_x=(final_x+3);  
        }  
        Serial.println('O');  
        old_x=new_x;  
    }  
}
```

Zjištění aktuálního  
náklonu gyroskopu

Porovnání s předchozím  
náklonem

Pokud je rozdíl dostatečně  
velký je odeslán příkaz za  
každý interval rozdílu

# Program - motor

```
if (Serial1.available(>0)
{
  x=Serial1.read();
  Serial.println(x);
  index_x=32;
}
```

Kontrola příchozích příkazů a  
nastavení násobiče počtu kroků

```
if ((x=='A')||(x=='B'))
{
  digitalWrite(X_ENABLE_PIN , LOW);
  digitalWrite(Y_ENABLE_PIN , HIGH);
  digitalWrite(Z_ENABLE_PIN , HIGH);
  if (x=='A')
  {
    digitalWrite(X_DIR_PIN , HIGH);
  }
  else
  {
    digitalWrite(X_DIR_PIN , LOW);
  }
  while(index_x>0)
  {
    digitalWrite(X_STEP_PIN , LOW);
    delay(1);
    digitalWrite(X_STEP_PIN , HIGH);
    index_x--;
  }
}
```

Zapnutí příslušného motoru

Nastavení správného směru

Provedení počtu kroků na  
základě násobiče