

# JAK SI SESTAVIT VLASTNÍHO ROBOTA

TOMÁŠ VODEHNAL

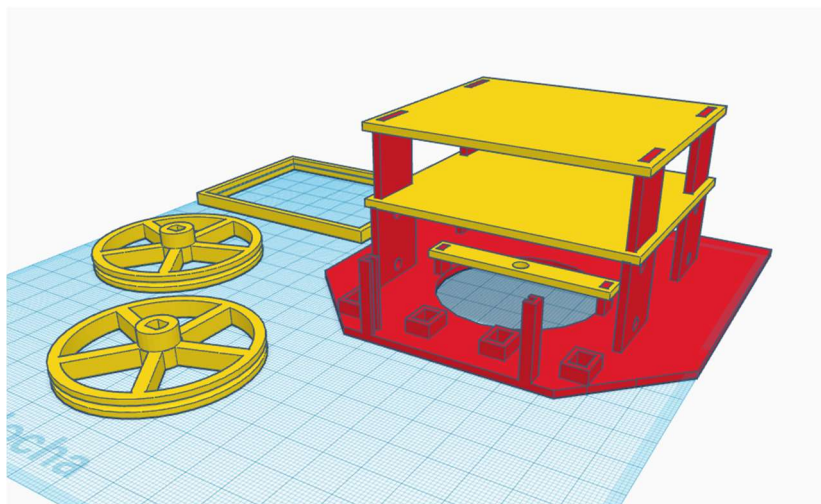








# MODELOVÁNÍ PODVOZKU





# ZÁKLADNÍ OVLÁDÁNÍ

## PRÁCE S MYŠÍ

Úkol:

Napiš, co se děje když:

Držím levé tlačítko myši a pohybuji s ní: \_\_\_\_\_

\_\_\_\_\_

Držím pravé tlačítko myši a pohybuji s ní: \_\_\_\_\_

\_\_\_\_\_

Držím kolečko myši a pohybuji s ní: \_\_\_\_\_

\_\_\_\_\_

## NOVÝ TVAR

Úkol:

Vlož do pracovní plochy kvádr a zjisti jaké má rozměry.

Šířka: \_\_\_\_\_

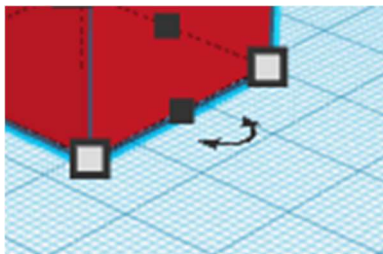
Výška: \_\_\_\_\_

Hloubka: \_\_\_\_\_

## ZMĚNA VELIKOSTI

Úkol:

Napiš, co se stane, pokud začneš myší tahat za některý z těchto bílých čtverců.



\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

## ZÁKLADNÍ OVLÁDÁNÍ – ŘEŠENÍ

---

### PRÁCE S MYŠÍ

Držím levé tlačítko myši a pohybuji s ní: **Objevuje se červený obdélník.**  
Tímto obdélníkem je možné vybírat objekty v pracovní ploše.

Držím pravé tlačítko myši a pohybuji s ní: **Pohybuje se úhel kamery.**

Držím kolečko myši a pohybuji s ní: **Posouvá se pozice kamery.**

---

### NOVÝ TVAR

Šířka: 20mm

Výška: 20mm

Hloubka: 20mm

---

### ZMĚNA VELIKOSTI

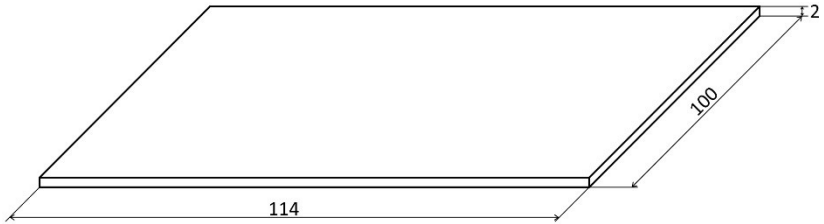
Tvar začne měnit své rozměry.

## DESKA PODVOZKU

### PRVNÍ KVÁDR

Úkol:

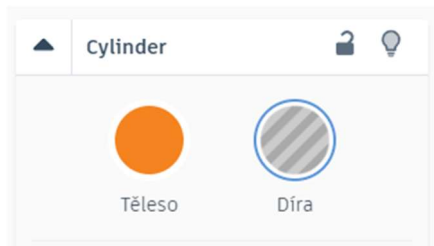
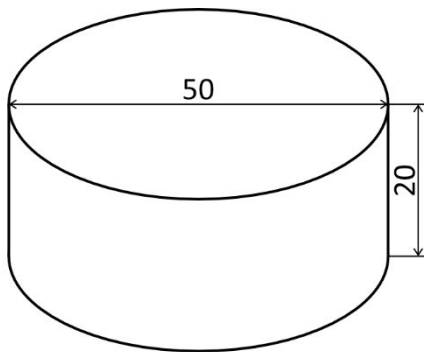
Vytvoř kvádr podle nákresu. Dodržuj přesné rozměry. Kóty jsou v jednotkách mm.



### DÍRA PRO KOLO

Úkol:

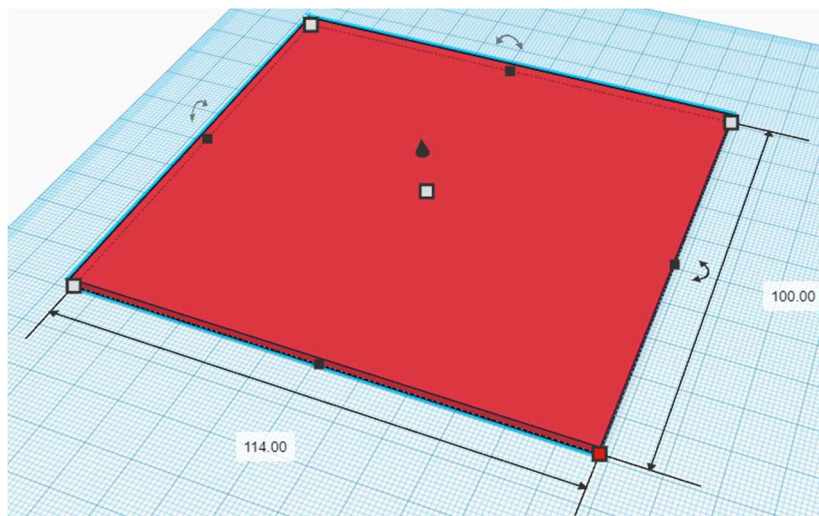
Vlož do pracovní plochy válec. Nastav mu rozměry podle obrázku a zapni mu režim díra.



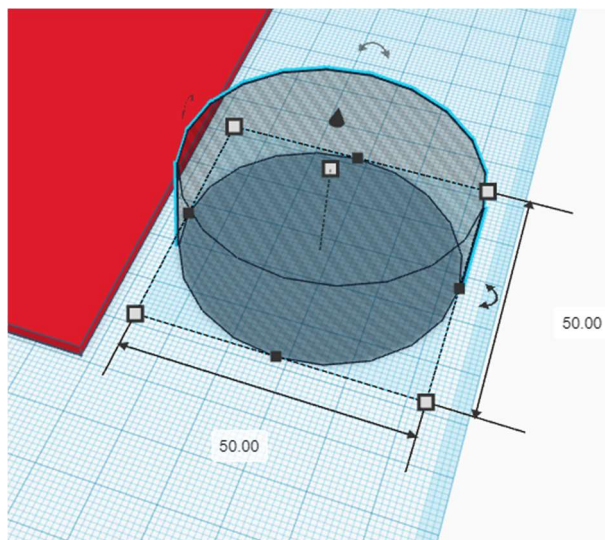


## DESKA PODVOZKU – ŘEŠENÍ

### PRVNÍ KVÁDR



### DÍRA PRO KOLO

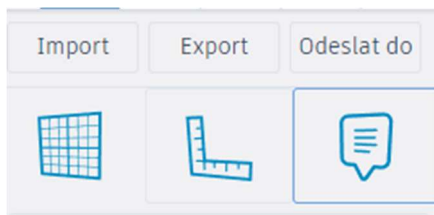


## PRAVÍTKO

### POČÁTEK SOUŘADNIC

Úkol:

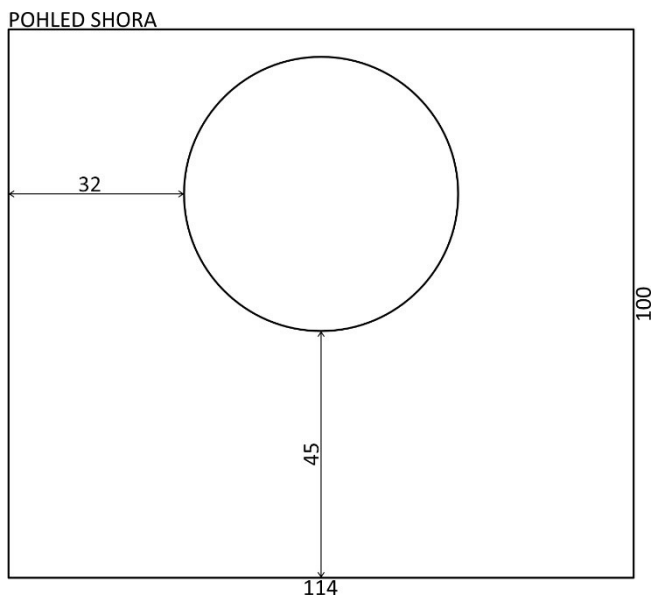
Najdi v TinkerCadu nástroj pravítko a umísti ho přesně do levého dolního rohu základní desky.



### VÁLEC NA DESKU

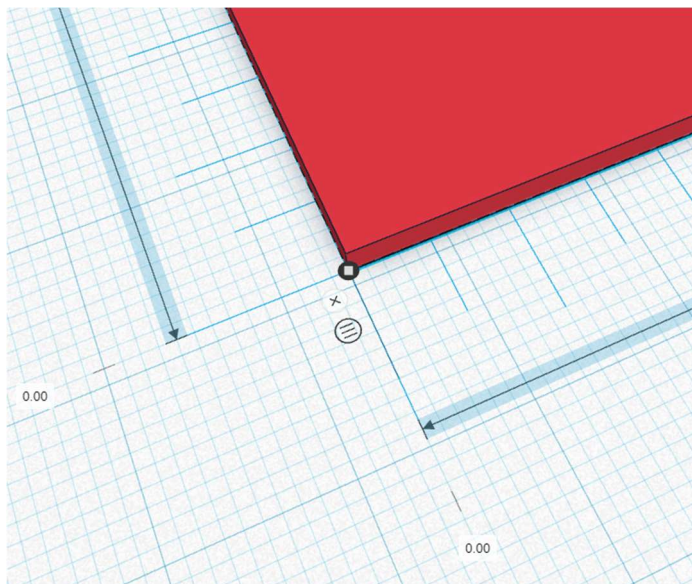
Úkol:

Umísti válec přesně na desku podle nákresu. K umístění ti pomůže pravítko.

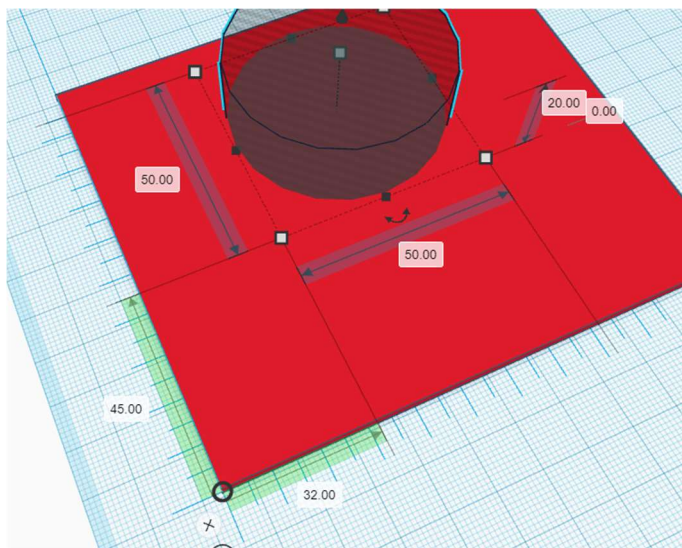


# PRAVÍTKO – ŘEŠENÍ

## POČÁTEK SOUŘADNIC




## VÁLEC NA DESKU



## SLOUPKY PRO NOSNÍK I

### SLOUČENÍ OBJEKTŮ

Úkol:

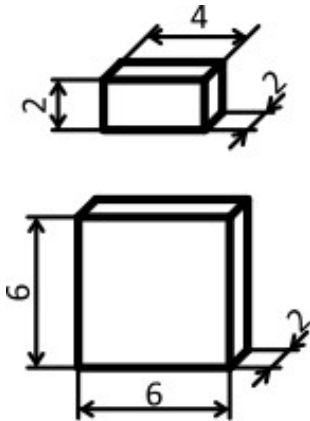
Vyber základní desku a správně umístěný válec pomocí klávesy SHIFT + kliknutí. Oba objekty sluč . Co se stalo?

- Uvnitř desky se vytvořila díra
- Nestalo se vůbec nic
- Na desce trčí do výšky válec

### DVA KVÁDRY

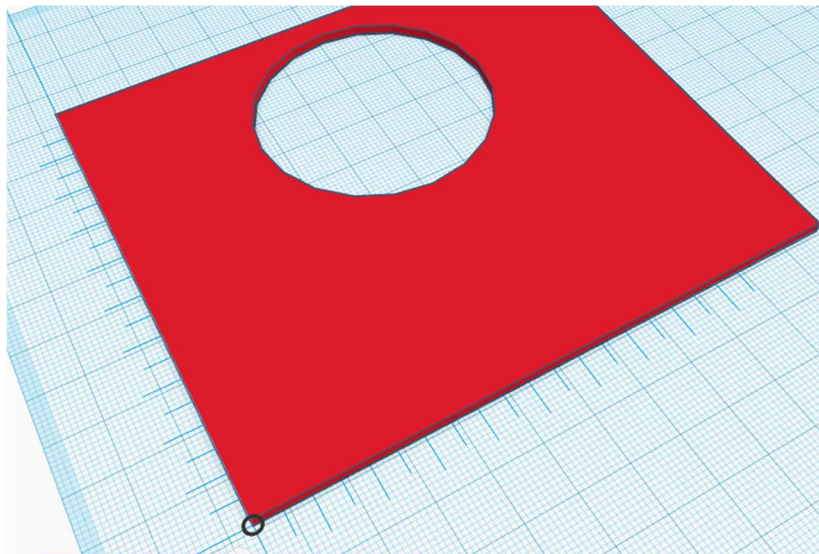
Úkol:

Vymodeluj dva kvádry podle obrázku.



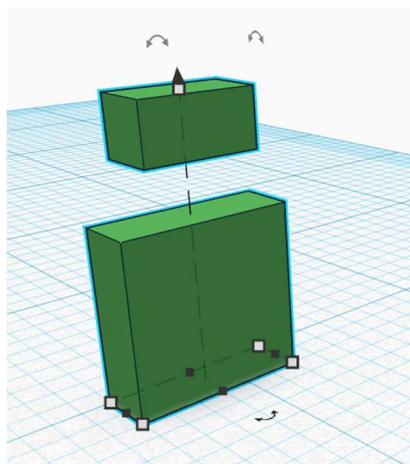
## SLOUPKY PRO NOSNÍK I – ŘEŠENÍ

### SLOUČENÍ OBJEKTŮ



- Uvnitř desky se vytvořila díra
- Nestalo se vůbec nic
- Na desce trčí do výšky válec

### DVA KVÁDRY





## SLOUPKY PRO NOSNÍK II

### SLOUČIT A ZKOPÍROVAT



Úkol:

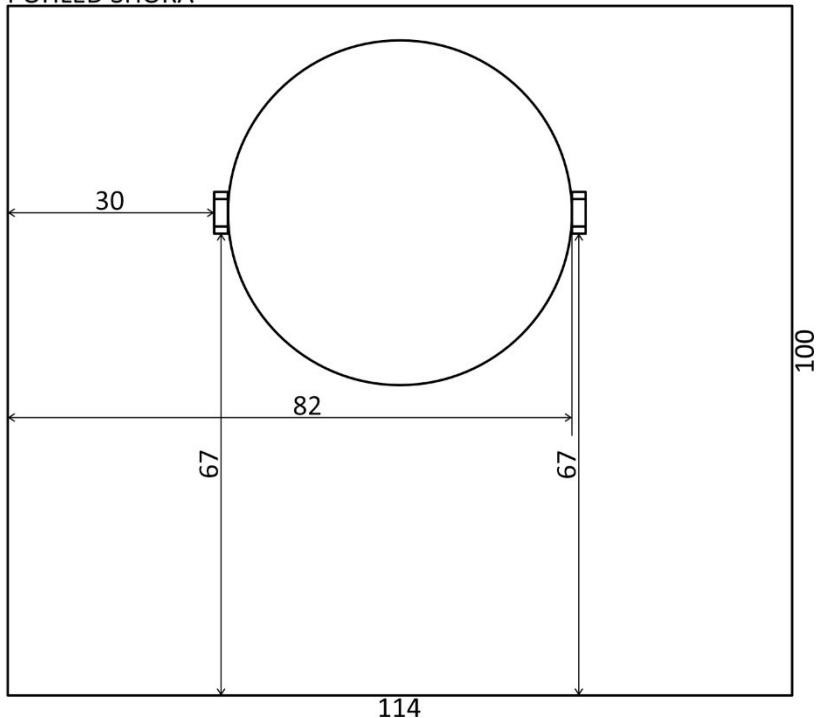
Umístí menší hranol na střed větší hranolu a spoj objekty. Následně celý sloupek zkopíruj.

### UMÍSTĚNÍ SLOUPKŮ

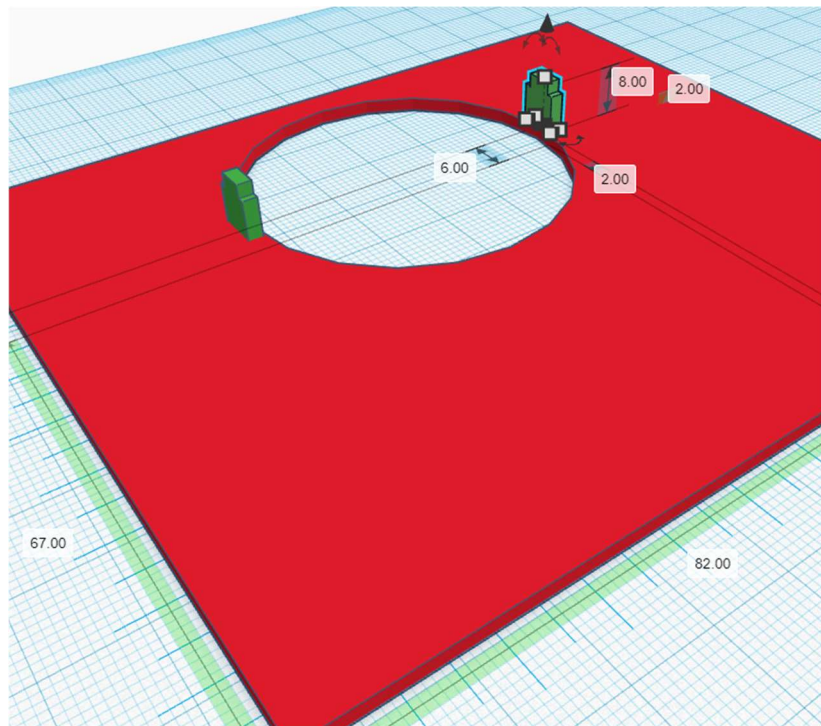
Úkol:

Umístí sloupky na hlavní desku podle nákresu. Pozor sloupky musíš o 2mm zvednout do výšky.

POHLED SHORA




## SLOUPKY PRO NOSNÍK II – ŘEŠENÍ



## TŘETÍ BOD

### NÁKRES NOSNÍKU

Úkol:

Vytvoř v TinkerCadu nosník pro třetí kolo. Díra o velikosti 5mm bude přesně uprostřed nosníku, k tomu použij toto tlačítko . Tloušťka bude opět 2mm.



### VĚTŠÍ DÍRY

Zamysli se:

Hranaté díry pomocí, kterých umístíme nosník na sloupky jsou v každém rozměru o 0,4mm větší než samotný sloupek. Proč si myslíš že to tak je?

---

---

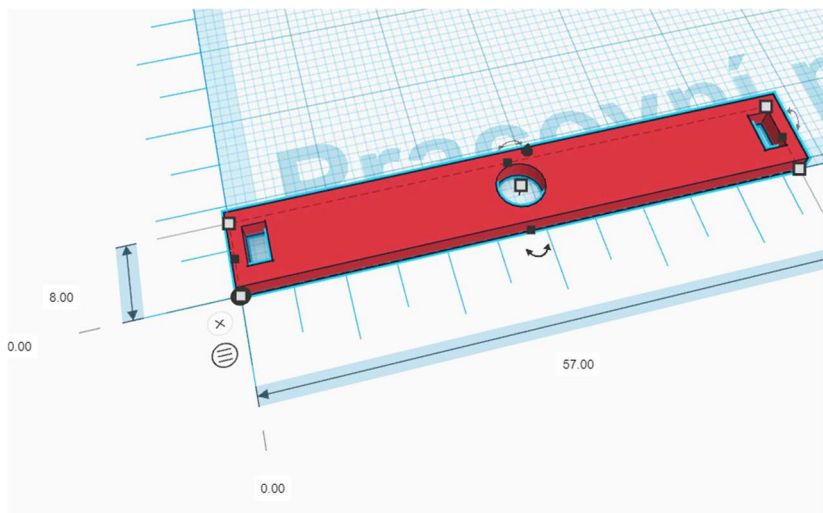
---

---

---

## TŘETÍ BOD – ŘEŠENÍ

### NÁKRES NOSNÍKU

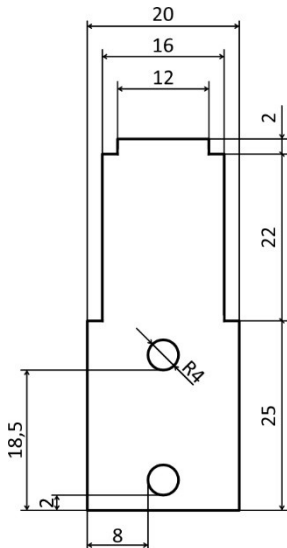


### VĚTŠÍ DÍRY

Důvod větších děr je tu proto, že tiskárna při 3D tisku nanáší o trochu vrstvu materiálu než je předepsaný rozměr, proto se na každé hraně přidávají 0,2mm.

## SLOUPKY PRO DESKY

### NÁKRES



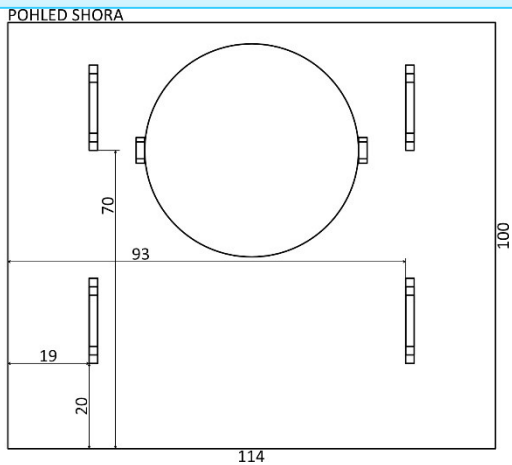
Úkol:

Vytvoř v TinkerCadu sloupek pro desku. Tento sloupek budeme potřebovat 4x. Až to budeš mít tak ho 4x zkopíruj. Tloušťka 2mm.

### UMÍSTĚNÍ

Úkol:

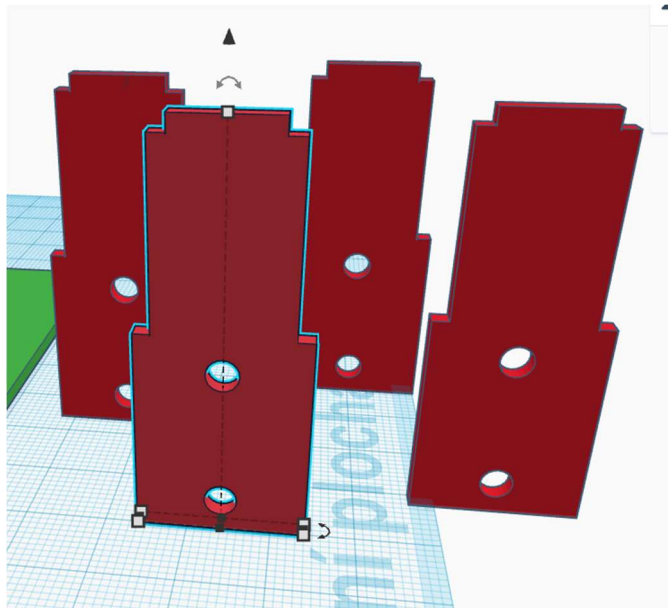
Čtyři takto vyrobené sloupky umísti na základní desku podle nákresu.



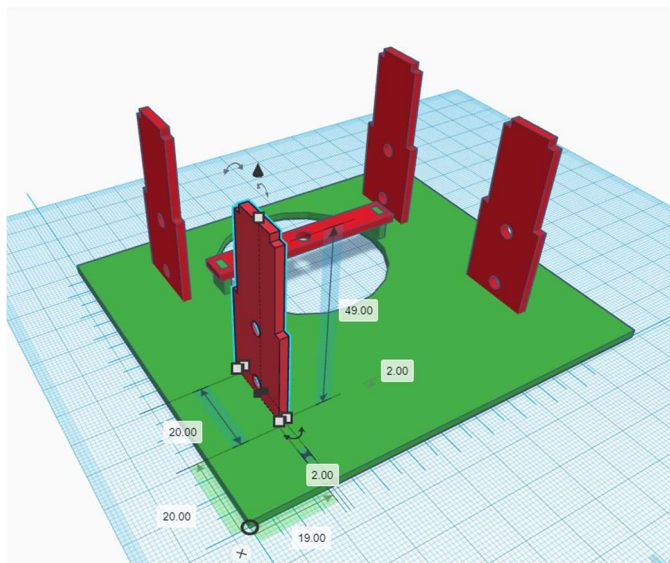


## SLOUPKY PRO DESKY - ŘEŠENÍ

### NÁKRES



### UMÍSTĚNÍ

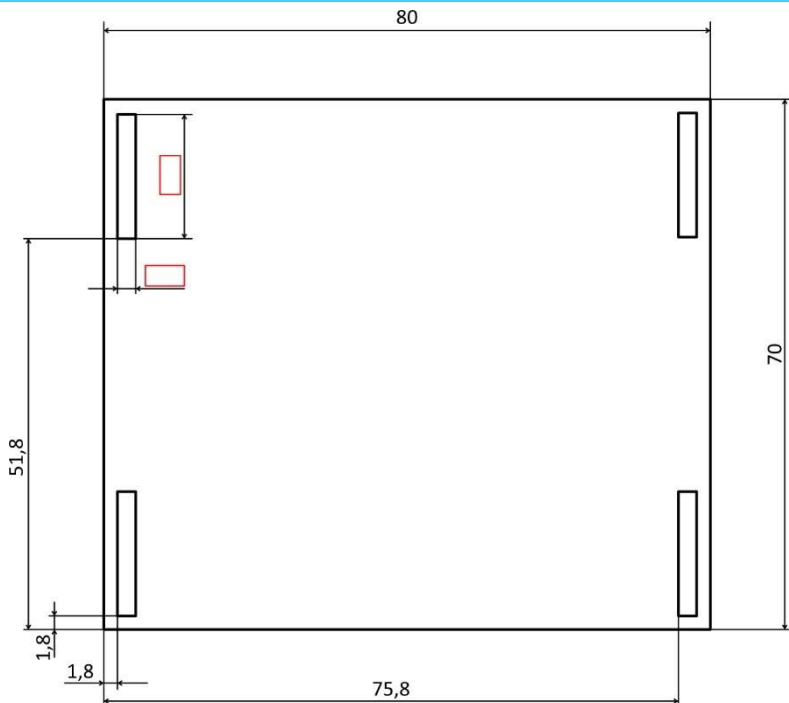


## SPODNÍ DESKA

### VELIKOST DĚR

Úkol:

Doplň do nákresu velikosti děr pro obě desky, které budou nasazeny na právě vymodelované sloupky. Nezapomeň, že díra musí být o 0,2mm větší na každé hraně.

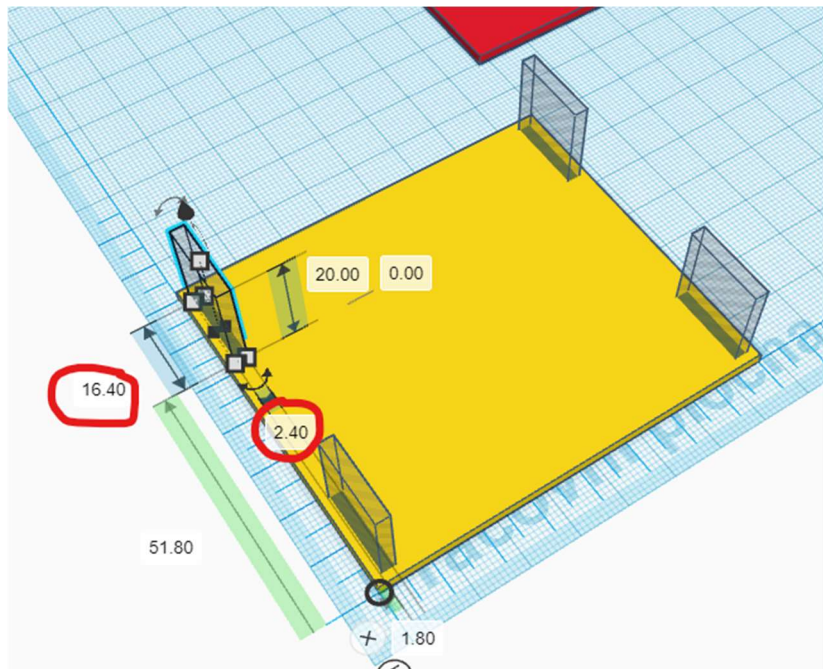


### VYMODELUJ TO

Úkol:

Když budeš mít doplněnou velikost děr, vymodeluj desku v TinkerCadu.

## SPODNÍ DESKA – ŘEŠENÍ

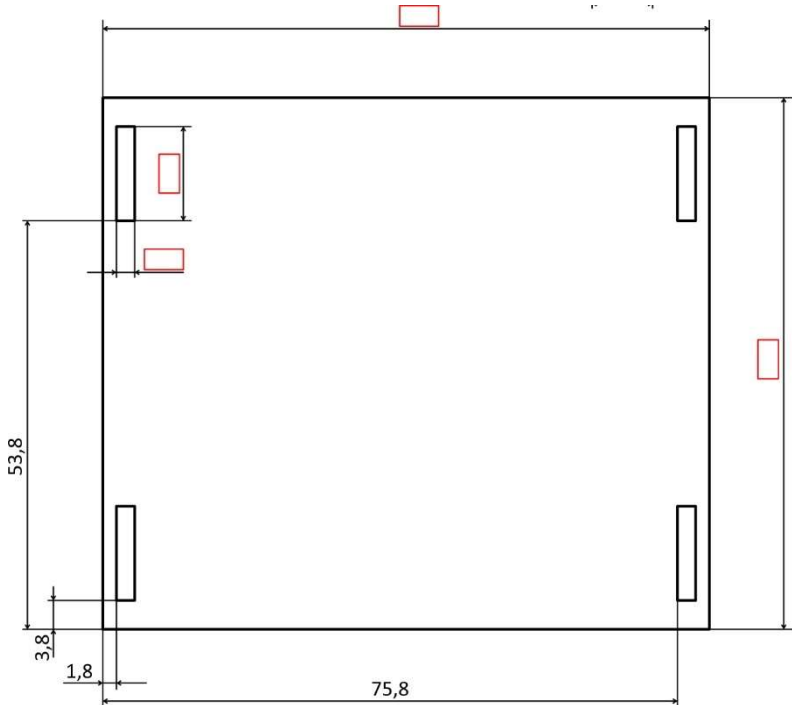


## HORNÍ DESKA

### VELIKOST DĚR

Úkol:

Horní deska bude mít stejné rozměry jako spodní deska. Jak velké budou její díry a rozměry? Doplň do obrázku.

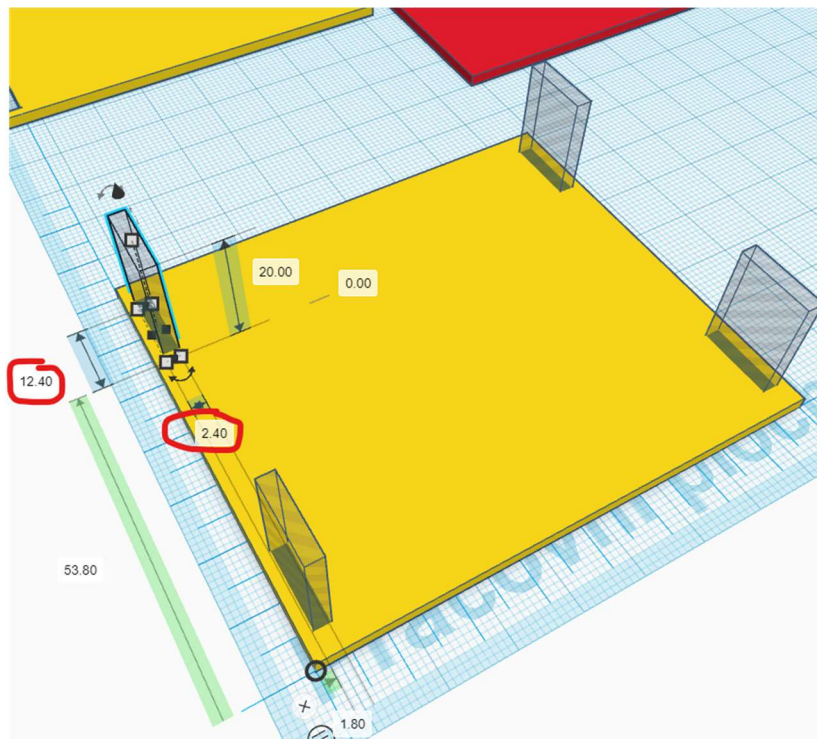


### VYMODELUJ TO

Úkol:

Když budeš mít doplněnou velikost děr, vymodeluj desku v TinkerCadu.

## HORNÍ DESKA – ŘEŠENÍ



Velikost desky je 70x80mm.

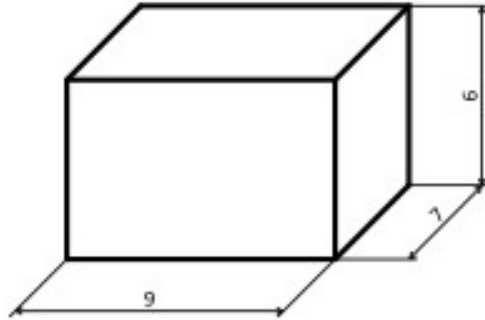


## POUZDRO PRO ČIDLA I

### KOSTKA

Úkol:

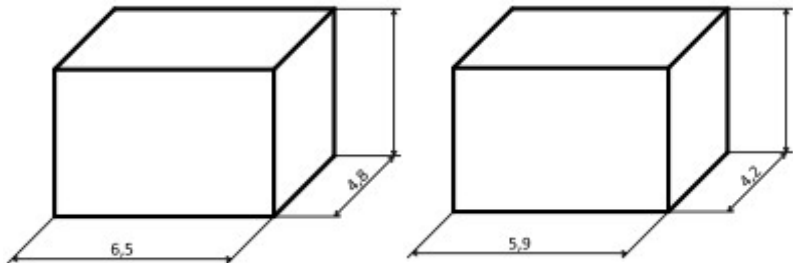
Vymodeluj v TinkerCadu kostku, do které budeme vkládat čidlo.



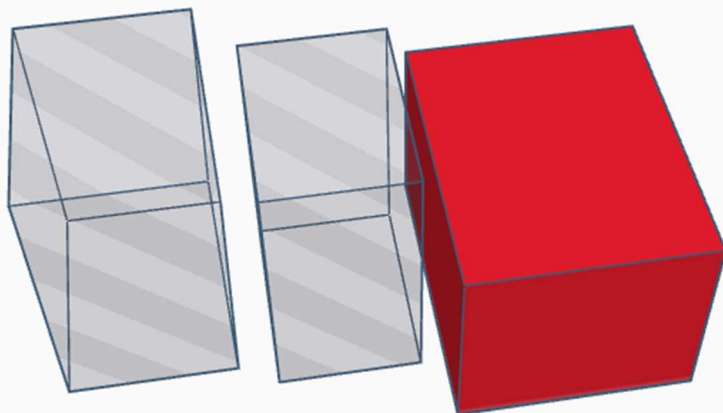
### HORNÍ A DOLNÍ DÍRA

Úkol:

Vymodeluj další dva kvádry, pomocí kterých budeme dělat otvor do hlavního kvádrů. Nastav jim možnost díra. Výška těchto kvádrů není podstatná, klidně ji můžeš nechat nastavenou na výchozích 20.



## POUZDRO PRO ČIDLA I – ŘEŠENÍ

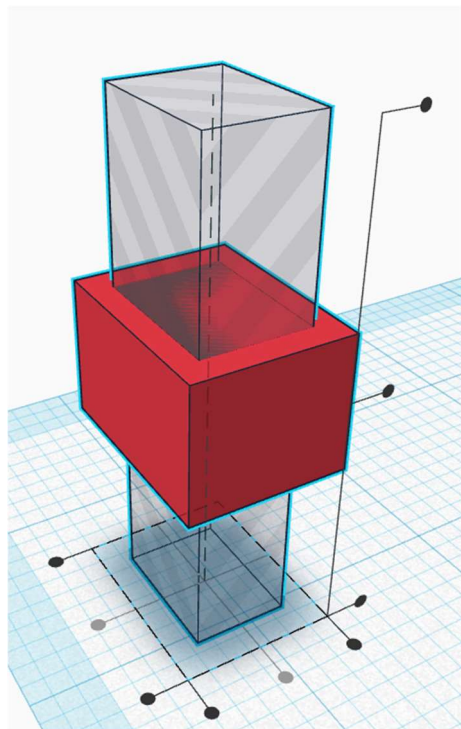


Pozor:

Výška průhledných kvádrů může být u každého jiná. Důležité pouze bude, jak hlubokou díru necháme vytvořit v dalším kroku.

## POUZDRO PRO ČIDLA II

### JENGA



#### Úkol:

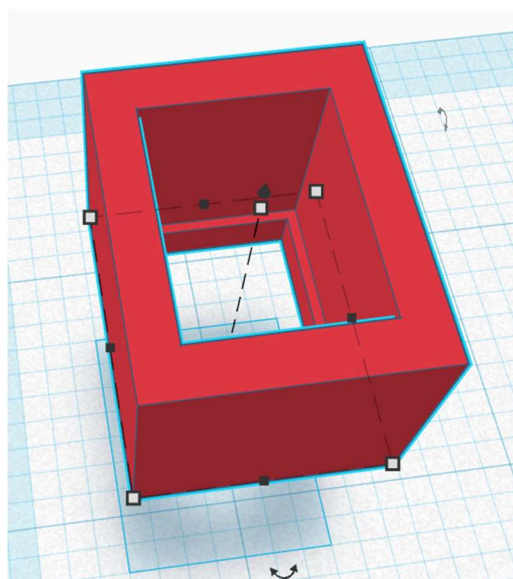
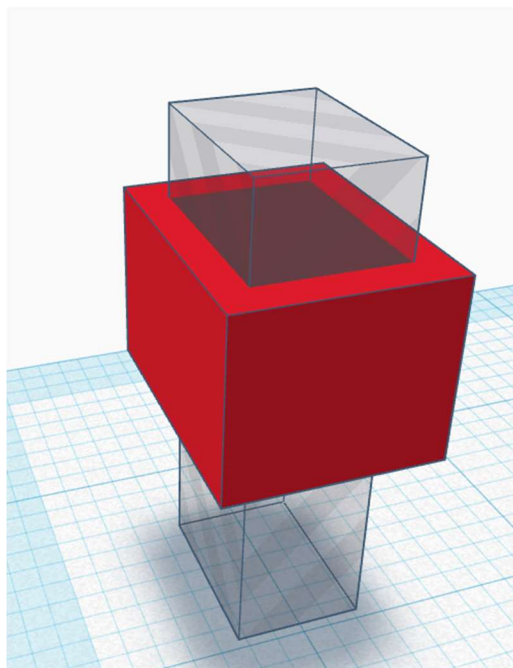
Naskládej 3 kvádry na sebe tak, aby dole byl nejmenší kvádr, uprostřed plný a nahoře velký průhledný kvádr. Umísti kvádry přesně na střed na sebe.

### ZASUNOUT DOVNITŘ

#### Úkol:

Horní kvádr zasuň do plného kvádru o 5mm a spodní o 1mm. Následně sluč všechny objekty dohromady.

## POUZDRO PRO ČIDLA II – ŘEŠENÍ

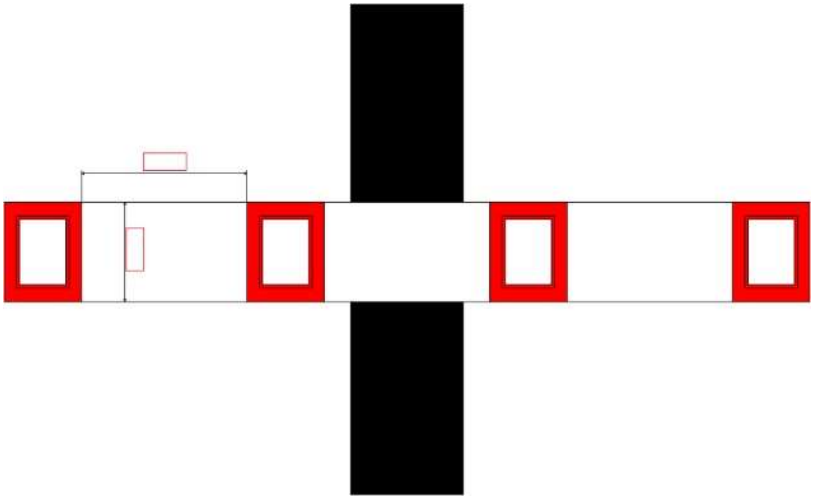


# SLEDOVAČ ČÁRY

## ROZTEČ

Úkol:

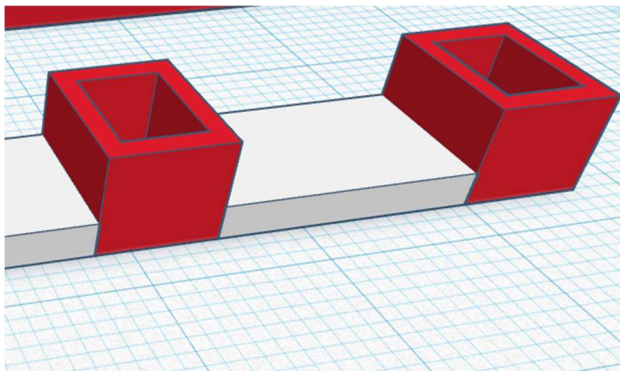
Čidla budou mít mezi sebou mezeru 15mm. Jakou velikost musí mít bloky, které umístíme mezi čidla a kolik jich musí být?



## VYMODELUJ TO

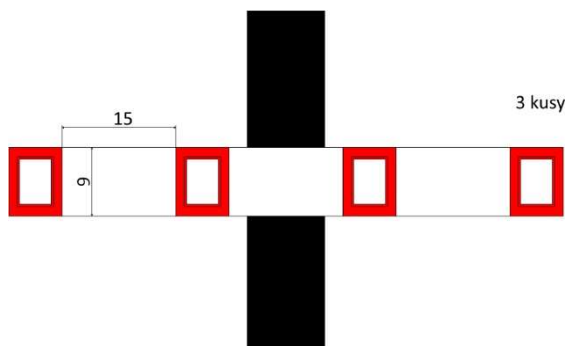
Úkol:

Vymodeluj držák čidel tak jak vidíš na horním obrázku. Tloušťka spojek bude 2mm. Na obrázku vidíš nápovědu.

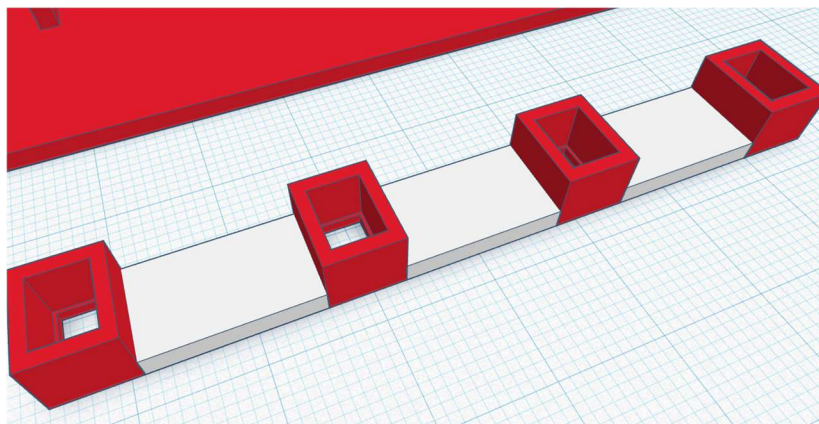


## SLEDOVAČ ČÁRY – ŘEŠENÍ

ROZTEČ



VYMODELUJ TO

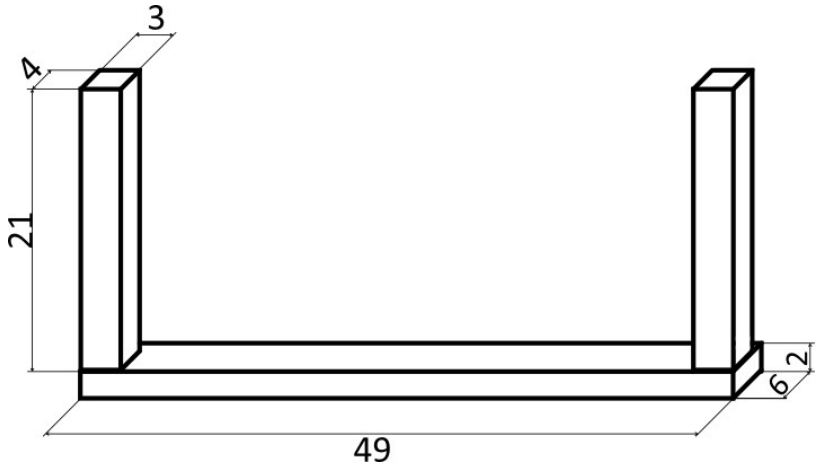


## DRÁŽKA PRO ULTRAZVUK

### 3 ZÁKLADNÍ KVÁDRY

Úkol:

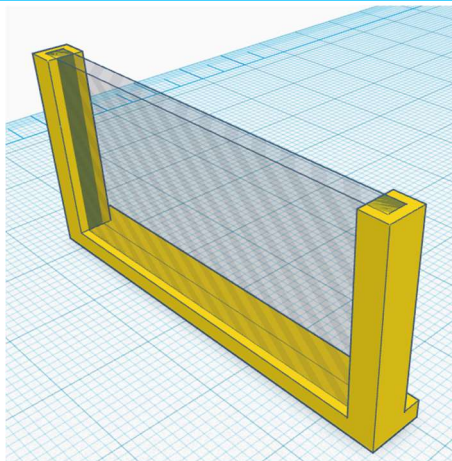
Vymodeluj 3 základní kvádry podle nákresu.



### DRÁŽKA

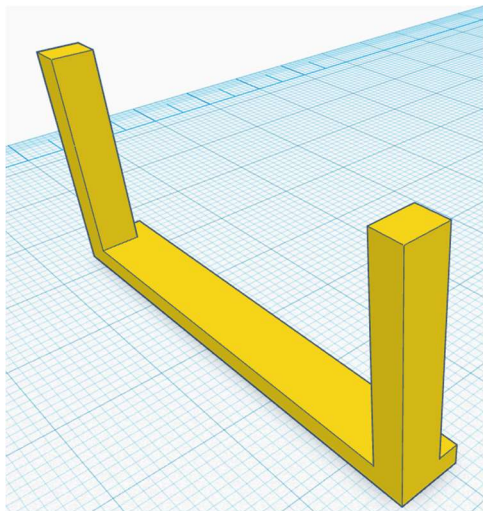
Úkol:

Vytvoř si desku o rozměrech 47x21x2mm a vytvoř pomocí ní drážky do sloupků.

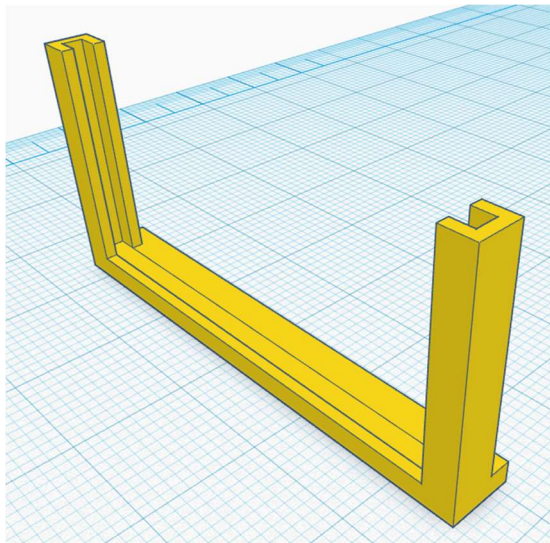


## DRÁŽKA PRO ULTRAZVUK – ŘEŠENÍ

### 3 ZÁKLADNÍ KVÁDRY



### DRÁŽKA



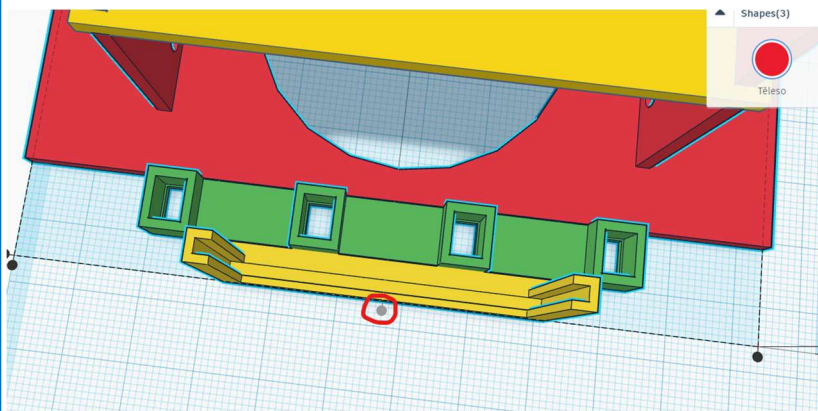


## HOTOVO

### SLOŽIT DOHROMADY

Úkol:

Obě vymodelované části připoj k přední části robota, jak vidíš na obrázku. Nezapomeň vše vycentrovat a objekty nakonec sloučit.



### KOLA

Dárek:

Možná si říkáš a kde jsou kola pro robota, tak to mám pro tebe dobrou zprávu. Kola ode mě dostáváš jako dárek, protože jsou hodně složitá na vymodelování.

Kdybys ale chtěl(a) můžeš si zkusit kola vymodelovat. Dělá se to pomocí importu svg souboru.

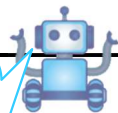


# ELEKTRONIKA V TINKERCADU





## KONTAKTNÍ POLE



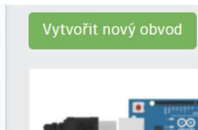
### NOVÝ OBVOD

Vyhledávání návrhů...

Vytvořit nový obvod

3D návrhy

Obvody

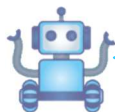


Pozor! Možná by ses chtěl(a) hned pustit do zapojování všech těch drátů, motorů součástek. Ale při troše nepozornosti by se také mohlo stát, že nějakou tu součástku „odpálíš“ a přestane fungovat. Proto si vše nejdříve zapojíme virtuálně.

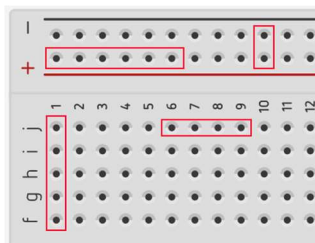
Úkol:

Zapni si TinkerCad, klikni na tlačítko obvody a vytvoř nový obvod.

### PROPOJENÍ KONTAKTNÍHO POLE



Hmmm??? Je tu hodně dírek, ale které jsou spolu propojené?

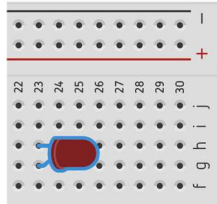
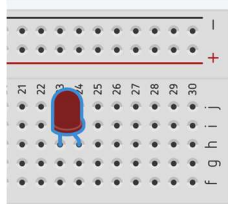


Úkol:

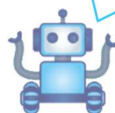
V TinkerCadu si přetáhni do plochy kontaktní pole.

Podívej se na obrázek a rozhodni, které obdélníky špatně označují dírky, které jsou spolu propojeny.

### PŘIDÁNÍ DIODY



Když teda chceš zapojit do pole součástku. Jak ji mám otočit?

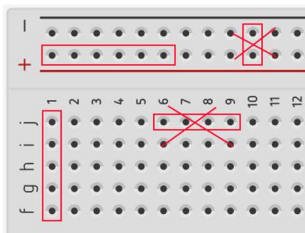


Úkol:

Co myslíš, který obrázek je správně zapojený?

## KONTAKTNÍ POLE – ŘEŠENÍ

### PROPOJENÍ KONTAKTNÍHO POLE



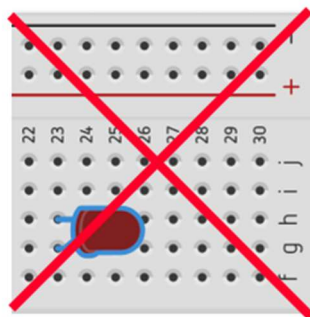
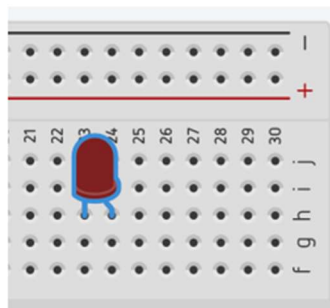
V kontaktním poli je vždy propojeno 5 svislých dírek a vždy celá řada pro napájení označená červenou a černou čarou a značkou + a –

Zapamatuj si:

V kontaktním poli jsou vždy spolu propojeny sloupečky 5 dírek nad sebou. A dva řádky označené + a – kam se zapojuje napájení (baterky).

### PŘIDÁNÍ DIODY

Špatně zapojená dioda je ta, co je otočená svisle dolů. Protože oba konce diody, jsou zapojené do stejného sloupečku a tím pádem jsou spolu propojeny a nemohli bychom jeden konec připojit do + a druhý do -.



Zapamatuj si:

Do kontaktního pole zapojujeme součástky vždy tak, aby každý její vývod byl zapojen v jiném sloupečku.

## ROZSVÍTÍME DIODU

### PLUS A MINUS

Úkol:

Zjisti v TinkerCadu jak se jmenují vývody LED diody a k jakému pólu se co připojuje?

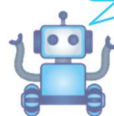
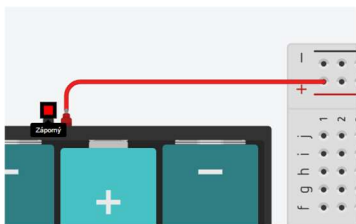
A \_\_\_\_\_

ZÁPORNÝ

K \_\_\_\_\_

KLADNÝ

### BEZ ENERGIE TO NEPŮJDE

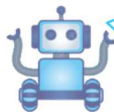


Auta jezdí na benzín nebo na naftu, ale na co pojedou já?

Úkol:

Najdi v TinkerCadu baterie a nastav je tak, aby byly 4. Připoj napájení do kontaktního pole. Co myslíš, kam připojíš kladný a záporný pól?

### SVÍTÍ?



Když mám plus a minus na diodě a plus a minus na baterkách, tak to jednoduše propojím a bude to svítit ne?

Úkol:

Připoj diodu ke zdroji napájení a spusť simulaci tlačítkem

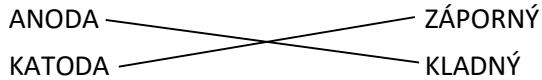
▶ Spustit simulaci

a zakroužkuj co se stalo.

dioda se rozsvítila    dioda se zničila    dioda se nerozsvítila

## ROZSVÍTÍME DIODU – ŘEŠENÍ

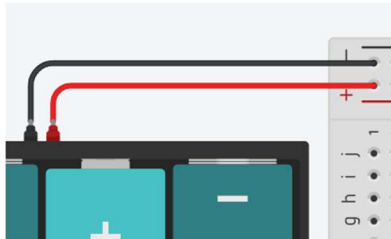
### PLUS A MINUS



Zapamatuj si:

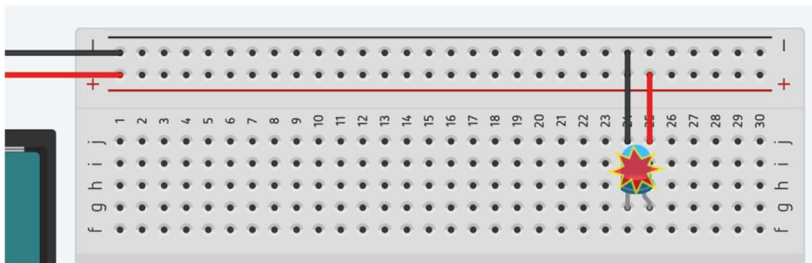
Vývody LED diody se jmenují ANODA a KATODA. Anoda se zapojuje na + a katoda na -. Skutečná dioda má anodu o trochu delší než je katoda.

### BEZ ENERGIE TO NEPŮJDE



### SVÍTÍ?

Pokud jsi vše zapojil správně, měl bys dostat odpověď, že se dioda zničila. V TinkerCadu to vypadá nějak takhle:

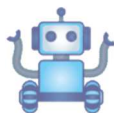


Pokud se ti dioda nerozsvítila, pak jsi pravděpodobně prohodil plus a minus.



## PŘEDŘADNÝ ODPOR

### HOŘÍ!



Uf... tak to bylo těsný, ta elektřina umí být fakt nebezpečná. Musíme ji trochu zkrotit.

### Úkol:

Zjisti, jak se jmenuje součástka, která přeměňuje elektrickou energii na teplo.

KONDENZÁTOR

REZISTOR

TRANZISTOR

### VÝPOČET

### Úkol:

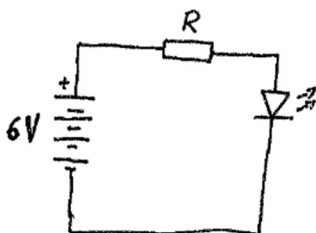
Spočítej předřadný odpor před diodu. Pro jednoduchost počítej s tím, že dioda potřebuje 2V napětí a 20mA proud. Napětí zdroje (baterií) je 6V.

$$R = \frac{U}{I}$$

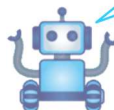
### SVÍTÍ, TEĎ UŽ DOOPRAVDY 😊

### Úkol:

Připoj předřadný odpor před diodu a nastav mu hodnotu, kterou jsi vypočítal(a) v předchozím cvičení. Spusť simulaci a vše by mělo v pořádku svítit.



Teď už to snad bude fungovat 😊



## PŘEDŘADNÝ ODPOR – ŘEŠENÍ

### HOŘÍ!

Zapamatuj si:

Součástka se jmenuje **rezistor** a její hlavní veličinou je elektrický **odpor**. Rezistor přeměňuje elektrickou energii na energii tepelnou.

### VÝPOČET

Nejdřív si převedeme proud na základní jednotky:

$$I = 20mA \rightarrow 0,02A$$

Zjistíme napětí na rezistoru.

$$U = \text{napětí zdroje} - \text{napětí diody}$$

$$U = 6 - 2 = 4V$$

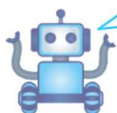
Potom můžeme dosadit do vzorce ohmova zákona:

$$R = \frac{U}{I}$$

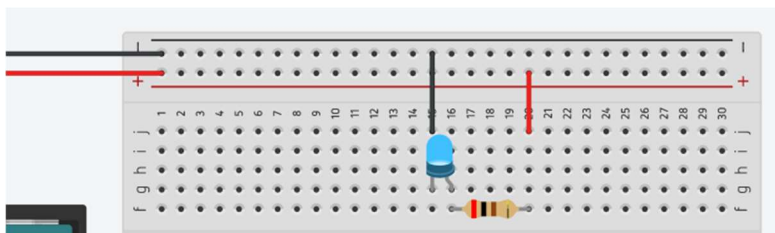
$$R = \frac{4}{0,02}$$

$$R = 200\Omega$$

### SVÍTÍ TEĎ UŽ DOOPRAVDY 😊



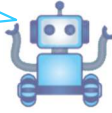
Hurá! Můj první sestavený obvod!!!



# ARDUINO

## MOZEK ROBOTA

Světýlko svítí, ale jak mám udělat, abych mohl diodu ovládat?



Arduino Uno R3

Úkol:

Najdi součástku Arduino Uno R3 a přidej ji do svého obvodu. Zjisti, čemu se u součástek říká pin?

Pin = \_\_\_\_\_

## NAPÁJENÍ Z ARDUINA

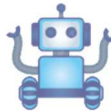
Od teď budeme vše zapojovat pomocí Arduina.

Úkol:

Smaž baterky, ty už nebudeme potřebovat. Najdi na Arduinu piny označené +5V a GND a připoj je ke kontaktnímu poli místo baterií.

## DIGITÁLNÍ VÝSTUPY

Digitální to jako znamená ty jedničky a nuly ne?



Úkol:

Podívej se na Arduino a zjisti kolik má procesor digitálních a kolik analogových pinů.

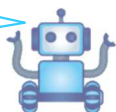
DIGITÁLNÍ: \_\_\_\_\_

ANALOGOVÉ: \_\_\_\_\_

# ARDUINO – ŘEŠENÍ

## MOZEK ROBOTA

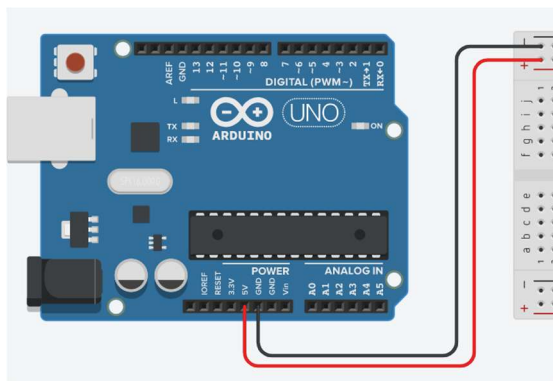
Takže můj mozek funguje digitálně? Analogově? Anebo má obří POWER???



Zapamatuj si:

Pin je označení pro vývod integrovaného obvodu, tedy i čipu Arduina. Arduino má mnoho různých pinů. Některé jsou napájecí (Power), některé analogové (Analog) a některé digitální (Digital).

## NAPÁJENÍ Z ARDUINA



## DIGITÁLNÍ VÝSTUPY

DIGITÁLNÍ: 14

ANALOGOVÉ: 6

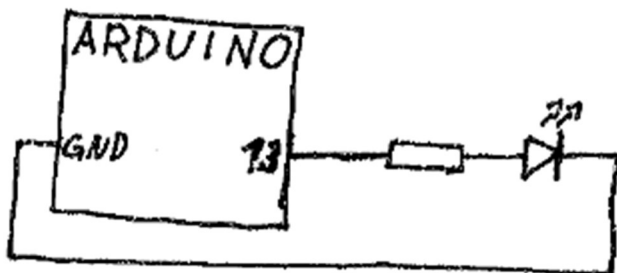
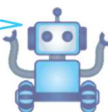
Zapamatuj si:

Digitální znamená, že procesor může svůj výstup buď zapnout (logická jednička) nebo vypnout (logická nula). Nic mezi tím. Má pouze dva stavy.

# HELLO WORLD!

## LEDKU PŘIPOJÍME K ČIPU

HELLO WORLD! V překladu Ahoj světe je nejjednodušší program, kterým každý programátor začíná.




Úkol:

Místo k plusu na kontaktním poli připoj rezistor s diodou k digitálnímu pinu číslo 13 na Arduinu a spusť simulaci. Pokud je všechno, jak má, dioda by měla blikat.

## PROGRAM

Úkol:

Najdi tlačítko  Kód na obrazovce a uvidíš program, který je automaticky nahraný v Arduinu. Změň program tak, aby dioda blikala každých 250ms (tj. ¼ sekundy).

## ZMĚNA PINU

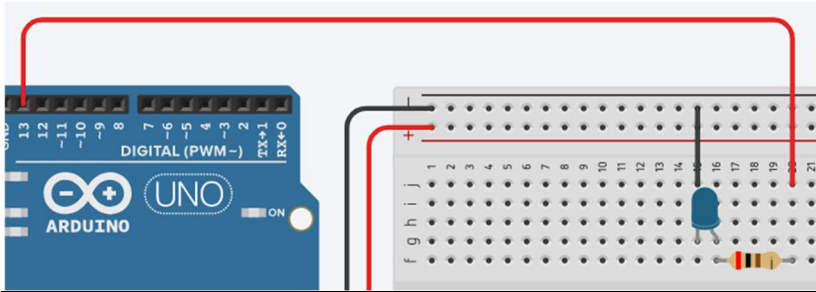
Úkol:

Připoj ledku k jinému digitálnímu pinu a uprav program Arduina tak, aby znovu ledka blikala.

## HELLO WORLD! – ŘEŠENÍ

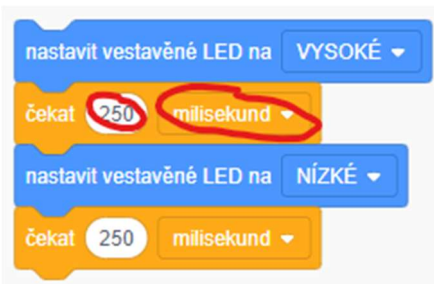
### LEDKU PŘIPOJÍME K ČIPU

Drátek, který vede na plus smažeš a místo toho připojíš drát k pinu číslo 13.



### PROGRAM

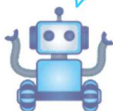
Stačí změnit parametry oranžových funkcí na 250 a přepnout místo sekund na milisekund. Dioda potom bliká rychleji.



### ZMĚNA PINU

Ledku připoj třeba na pin 6. Musíš vyhodit z programu modré bloky a dát místo nich nové. Nastavit pin, ke kterému je ledka připojena a nastavit, aby střídavě nastavil vysokou hodnotu (jednička) a nízkou

Ach ta práce s bloky, je trochu kostrbatá, ale snad to nějak půjde.

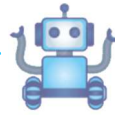


# TLAČÍTKO

## PŘIDEJ TLAČÍTKO



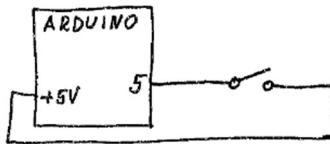
Dioda si bliká sama jak chce,  
ale jak ji můžu ovládat?



Úkol:

Najdi součástku „tlačítko“ a zapoj ho do kontaktního pole tak, aby každý vývod měl vlastní sloupeček.

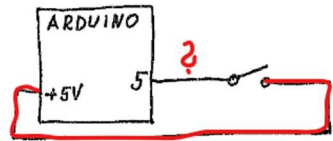
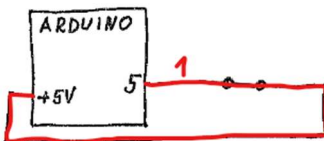
## DIGITÁLNÍ VSTUP



Úkol:

Připoj tlačítko podle nákresu. Když zmáčkneš tlačítko, na vstup Arduina se dostane logická jednička +5V.

## JEDNIČKA NEBO NULA?



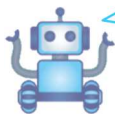
Zamysli se:

Když bude tlačítko zmáčknuté, co bude připojené k pinu 5? Co když tlačítko zmáčknuté nebude bude k pinu 5 něco připojeno?

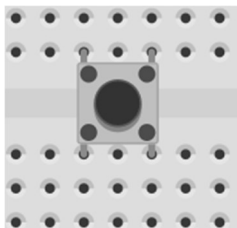
Tlačítko zmáčknuté:	JEDNIČKA	NULA	NIC
Tlačítko rozepnuté:	JEDNIČKA	NULA	NIC

## TLAČÍTKO – ŘEŠENÍ

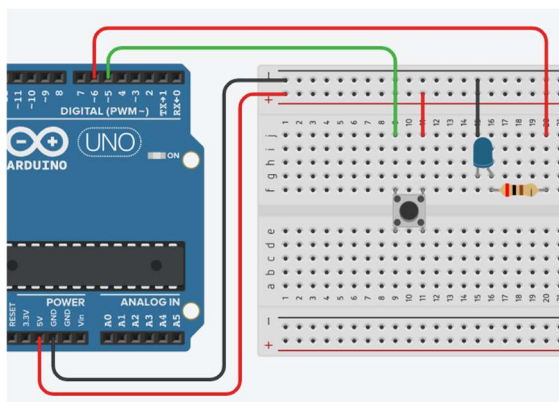
### PŘIDEJ TLAČÍTKO



Pokud si dobře pamatují, tak součástky musím zapojit tak, aby každý vývod byl ve vlastním sloupečku.



### DIGITÁLNÍ VSTUP



### JEDNIČKA NEBO NULA?

Tlačítko zmáčknuté: **JEDNIČKA**                      NULA                      NIC

Tlačítko rozeprnuté: **JEDNIČKA**                      NULA                      **NIC**

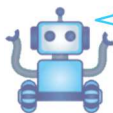
Zapamatuj si:

Pokud nějaký vstup není jednoznačně připojen k jedničce, ani k nule, pak je na něm tzv. **nedefinovaný stav**.

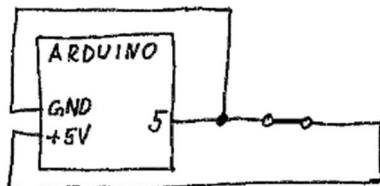


## ZNIČ NEDEFINOVANÝ STAV

### ZKRAT



Tak když to tlačítko není nikam připojené, tak ho prostě připojím na nulu ne?

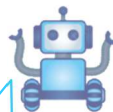
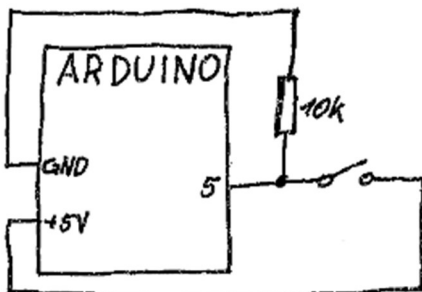


### Úkol:

Podívej se na zapojení tlačítka. Označ, v obrázku, jakou nejkratší cestou bude elektrický proud procházet od 5V k GND. Pak se zamysli a vzpomeň si co se stalo s LEDkou když jsme ji zapojili bez rezistoru.

Co se stalo s LEDkou? \_\_\_\_\_

### PULL-DOWN RESISTOR



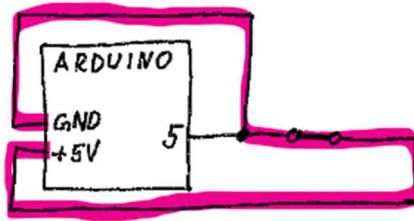
Zkrat, to je tedy pořádná síla. Tak to proti němu musíme vyvinout pořádný odpor.

### Úkol:

Zapoj tzv. pull-down rezistor podle nákresu. Rezistor by měl být hodně velký, proto mu nastav hodnotu 10k $\Omega$ .

## ZNIČ NEDEFINOVANÝ STAV – ŘEŠENÍ

### ZKRAT

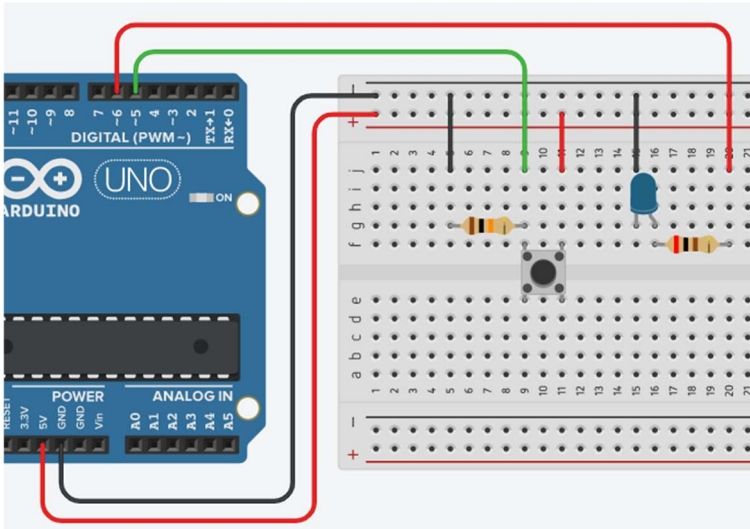


Co se stalo s LEDkou? Ledka se zničila, protože přes ní šel tak obrovský proud, že jeho energii nevydržela.

Zapamatuj si:

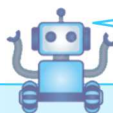
Zkrat je stav obvodu, kdy bez jediného odporu je spojen kladný pól napájení se záporným pólem.

### PULL-DOWN RESISTOR



## JE TLAČÍTKO ZMÁČKNUTÉ NEBO NE?

### VSTUP DO PROGRAMU

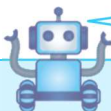


Jak že to je? Tlačítko je digitální vstup?

Úkol:

Podívej v sekci vstup jsou samé oválné fialové bloky. Jeden z nich je: **čist digitální kolík #**. Přetáhni si ho do plochy a nastav ho tak, aby nám četl kolík, ke kterému je připojeno tlačítko.

### ZAPAMATUJ SI



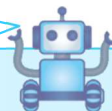
Já??? Jo nee, ten program, já už si nechci nic pamatovat.

Úkol:

Teď musíme program nechat, aby přečtenou jedničku nebo nulu si někde zapamatoval. K tomu slouží proměnná. Vytvoř proměnnou s názvem „tlačitko“ a nastav ji tak, aby si pamatovala to, co si přečetla na digitálním kolíku.

### POKUD ZMÁČKNU TLAČÍTKO, TAK...

Nápověda: jsou žluté 😊



Úkol:

Prohledej bloky v kódu a zkus najít nějaké, které by nám pomohli říct programu, aby něco udělal, pokud zmáčkeme tlačítko. Jaký tvar bloku se může umístit do žluté podmínky pokud?

ŠPIČATÝ

OVÁLNÝ

OBDELNÍKOVÝ

### OPERÁTOR

Úkol:

Opět prohledej bloky a najdi nějaký, který by se mohl do žlutého bloku umístit a zároveň nám dokáže porovnat dvě hodnoty.

Nápověda: je zelený 😊



## JE TLAČÍTKO ZMÁČKNUTÉ NEBO NE? – ŘEŠENÍ

### VSTUP DO PROGRAMU



číst digitální kolík 5 ▾

### ZAPAMATUJ SI



nastavit tlačítko ▾ na číst digitální kolík 5 ▾

Zapamatuj si:

Proměnná je kousek paměti programu, do které můžeme ukládat různé informace. Například o tom, jestli je na vstupu logická jednička nebo nula.

### POKUD ZMÁČKNU TLAČÍTKO TAK...

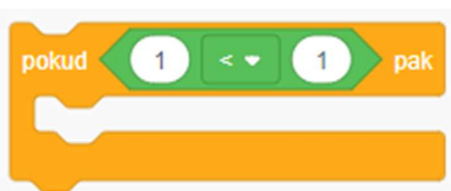


ŠPIČATÝ

OVÁLNÝ

OBDELNÍKOVÝ

### OPERÁTOR

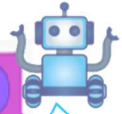
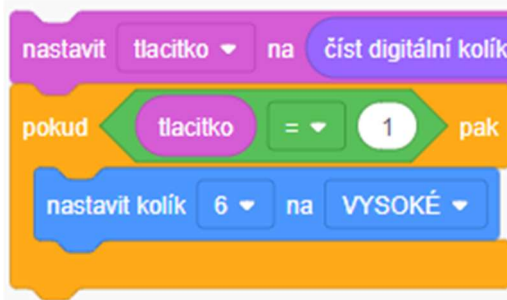


Zapamatuj si:

Operátor porovnání nám zjistí, jestli jsou nějaké dvě hodnoty větší, menší nebo se rovnají.

## OVLÁDEJ DIODU

### POKUD MAČKÁM, TY SVÍTÍŠ



Toto není celý program,  
je to jenom nápověda,  
ještě kousek chybí.

Úkol: Doplň text

Teď už je to jasné. Když zmáčknu tlačítko, na pinu je \_\_\_\_\_ a program rozsvítí LEDku. Když tlačítko nezmáčknu na pinu je \_\_\_\_\_ a program zhasne LEDku.

Naprogramuj Arduino.

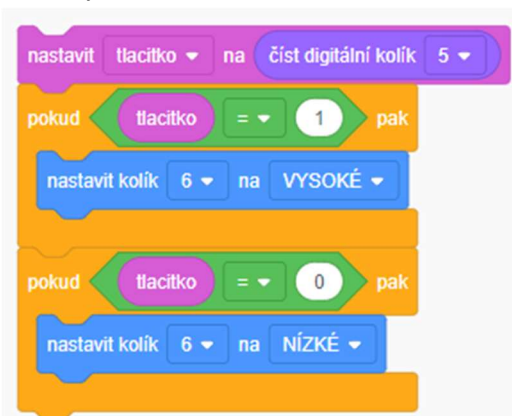
### POKUD NEMAČKÁM TY SVÍTÍŠ

Úkol:

Teď to zkus naprogramovat opačně. Když je tlačítko zmáčknuté, LEDka je zhaslá. Pokud ho pustím, LEDka bude svítit.

## OVLÁDEJ DIODU – ŘEŠENÍ

### POKUD MAČKÁM, TY SVÍTÍŠ



Teď už je to jasné. Když zmáčknu tlačítko, na pinu je **jednička** a program rozsvítí LEDku. Když tlačítko nezmáčknu na pinu je **nula** a program zhasne LEDku.

### POKUD NEMAČKÁM, TY SVÍTÍŠ



Všimni si:

Jediná změna programu je v samotné podmínce. Kde porovnáváme, jestli se tlačítko rovná nule nebo jedné

# SVĚTLO

## FOTOREZISTOR

Úkol:

Najdi součástku fotorezistor a připoj ho na volné místo do kontaktního pole.

Vyber, co označuje slovo „foto“.

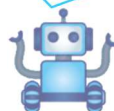
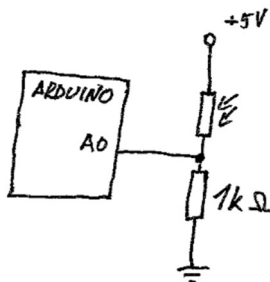
SVĚTLO

FOTOGRAFIE

NAPĚTÍ

## ZNOVU PULL-DOWN

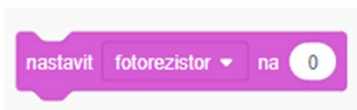
Takže fotorezistor je takové tlačítko, které reaguje na světlo?



Úkol:

Připoj fotorezistor k jednomu z analogových pinů Arduina stejným způsobem jako jsme zapojovali tlačítko.

## ČÍST ANALOGOVÝ KOLÍK



Úkol:

Vytvoř si novou proměnnou s názvem fotorezistor a nastav ji na hodnotu analogového pinu, ke které jsi připojil(a) fotorezistor.

# SVĚTLO - ŘEŠENÍ

## FOTOREZISTOR



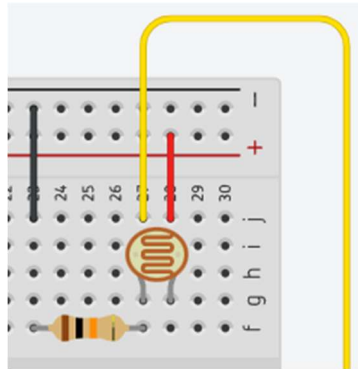
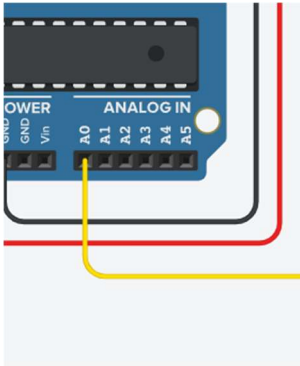
Slovo foto znamená:

**SVĚTLO**

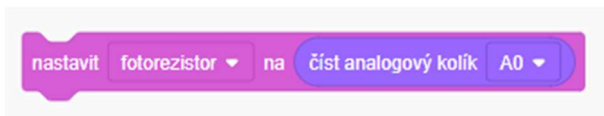
FOTOGRAFIE

NAPĚTÍ

## ZNOVU PULL-DOWN



## ČÍST ANALGOVÝ KOLÍK





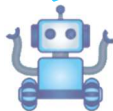
## KOLIK JE TU SVĚTLA?

### SÉRIOVÝ MONITOR

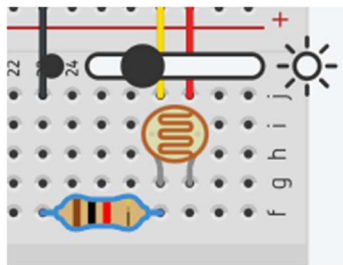
Úkol:

Najdi v kódu blok „vytisknout na sériovém monitoru a vlož do něj proměnnou fotorezistor.“

Jak zjistím, co se uvnitř Arduina děje?



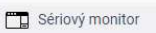
### 0 AŽ 1024



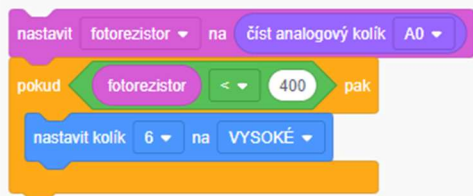
TMA: \_\_\_\_\_

SVĚTLO: \_\_\_\_\_

Úkol:

Spusť simulaci a během ní klikni na fotorezistor. Objeví se ti posuvník, kterým simuluješ světlo nebo tmu. Napiš, jakou hodnotu vidíš na , když je tma a jakou když je světlo.

### KDYŽ JE TMA TAK ROZSVÍŤ



To je jenom část kódu, ještě chybí druhá polovina.

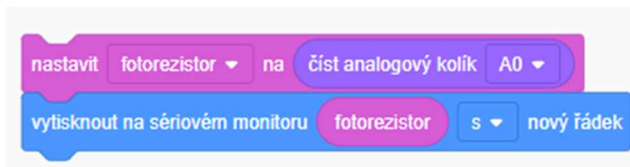


Úkol:

Naprogramuj program, který rozsvítí diodu, když bude tma a zhasne ji, když bude světlo.

## KOLIK JE TU SVĚTLA – ŘEŠENÍ

### SÉRIOVÝ MONITOR



### 0 – 1024

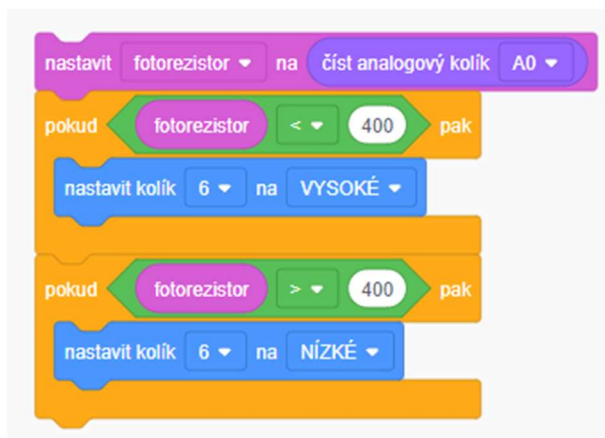
TMA: 6

SVĚTLO: 679

Pozor:

Čísla se mohou u každého lišit. Důležité je najít jakou hodnotu má fotorezistor zhruba v polovině posuvníku a tu poté použít v dalším úkolu do podmínek.

### KDYŽ JE TMA, TAK ROZSVÍŤ



## STMÍVÁ SE

### PWM VÝSTUPY



Úkol:

Podívej se na Arduino a napiš, jaké digitální piny mají tzv. funkci PWM. Tyto piny jsou označeny vlnovkou.

PINY S FUNKCÍ PWM: \_\_\_\_\_

### PŘIPOJ DO PWM DIODU

Úkol:

Zkontroluj si, jestli máš připojenou diodu do jednoho PWM výstupů. Pokud ne, přepoj ji třeba na pin 6.

### 0 AŽ 256

Úkol:

Nech v programu jenom první růžový blok a pod něj vlož blok, kde můžeš nastavit pinu 6 i jiné hodnoty než jenom vysoké a nízké.

### O KOLIK MUSÍME HODNOTU ZMENŠIT?

Úkol:

Vyzkoušej na kalkulačce. Fotorezistor dává hodnoty od 0 do 1024, ale pwm mohou nastavit od 0 do 256. Jakým číslem musíme hodnotu fotorezistoru vydělit, aby se nám vešla do pwm?

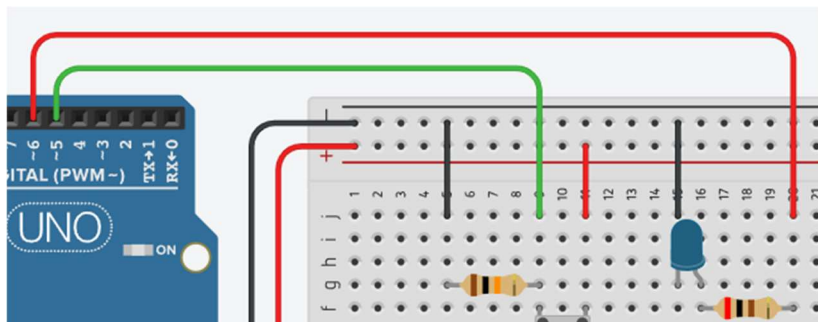
$$\frac{1024}{???} = 256$$

## STMÍVÁ SE – ŘEŠENÍ

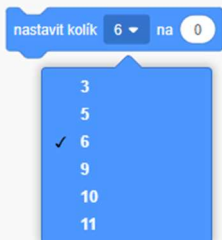
### PWM VÝSTUPY

PINY S FUNKCÍ PWM: 3, 5, 6, 9, 10 a 11.

### PŘIPOJ DO PWM DIODU



### 0 AŽ 255



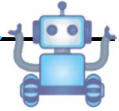
Všimni si:

Že nastavit libovolnou hodnotu jde pouze na těch pinech, které mají funkci pwm.

### O KOLIK MUSÍME HODNOTU ZMENŠIT?

$$\frac{1024}{4} = 256$$

# INTENZITA SVĚTLA



## MATEMATIKA

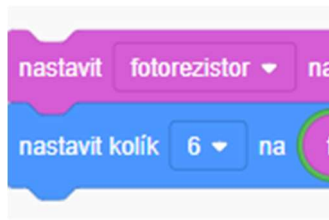


Aha, takže programování je trochu i o matematice.

Úkol:

Najdi matematickou funkci v kódu TinkerCadu, která nám dokáže vydělit hodnotu fotorezistoru čtyřmi.

## HOĎ TO NA VÝSTUP



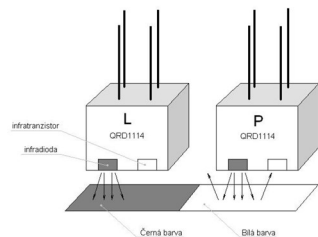
Úkol:

Fajn jestli máš dělení čtyřmi za sebou, pak už nám stačí tuto hodnotu hodit na výstup. Spusť si pak simulaci. Při změně světla by měla jinak svítit i dioda.

## JAK TO POUŽÍT U ROBOTA?

Zamysli se:

Jak by se tohle všechno dalo použít pro navigaci robota po čáře? Jestli máš nějaký nápad, poznamenej si ho.



---

---

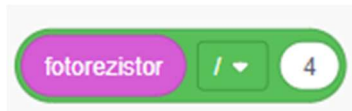
---

---

---

## INTENZITA SVĚTLA – ŘEŠENÍ

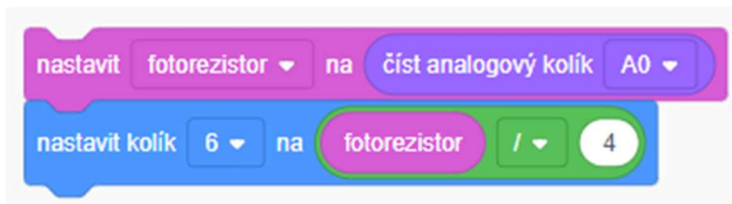
### MATEMATIKA



Všimni si:

Všechny bloky mají určitý tvar, který jde umístit pouze do míst se stejným tvarem. Všechny bloky, které pro nás reprezentují nějakou hodnotu (třeba kolik je světla na fotorezistoru děleno čtyřma) mají oválný tvar.

### HOĎ TO NA VÝSTUP



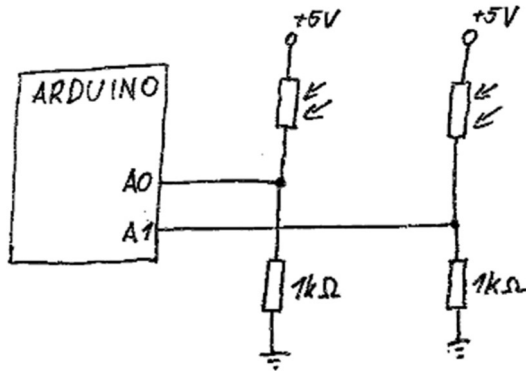
### JAK TO POUŽÍT U ROBOTA?

Pro zvědavé:

Čára je černá a okolí čáry je bílé. Bílá barva odráží světlo lépe než černá barva. Pro čidlo to pak vypadá jako kdyby bylo ve tmě. Když je tedy na čidle tma, pak to znamená, že je na černé čáře.

## VIRTUÁLNÍ ROBOT (HARDWARE)

### DVĚ ČIDLA

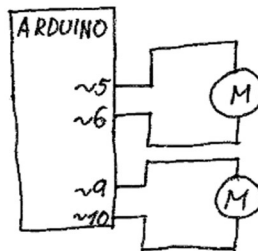
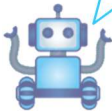


Úkol:

Připoj k procesoru dva fotorezistory podle nákresu. Jeden bude simulovat levý senzor a druhý pravý senzor.

### DVA MOTORY

Hej a to můžu jen tak připojit motory k čipu?

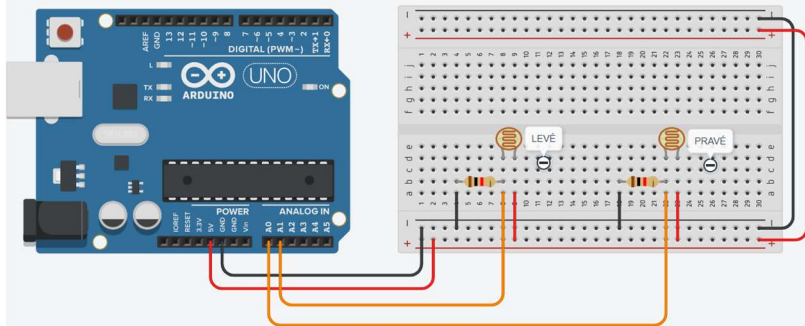


Úkol:

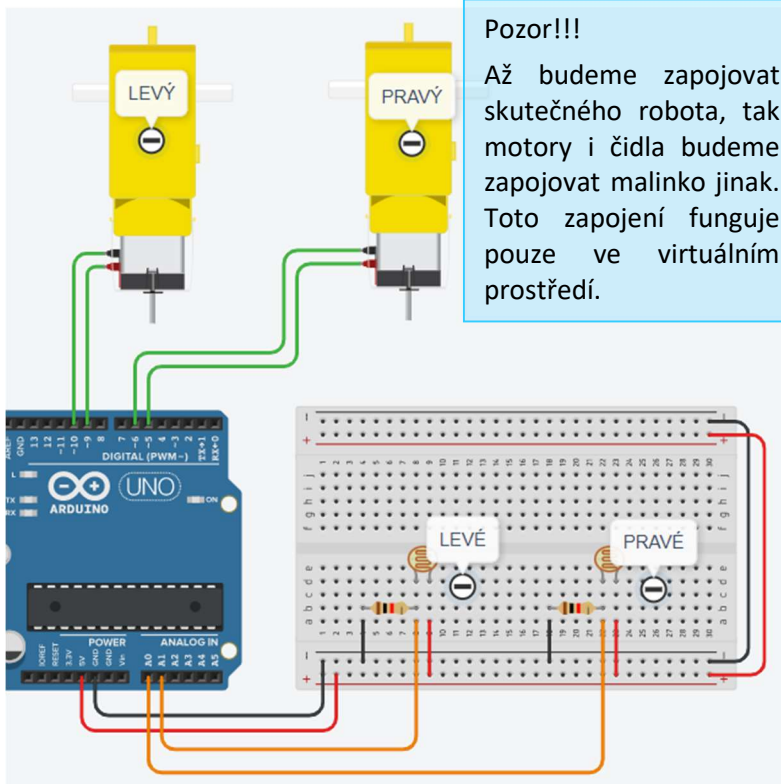
Připoj k procesoru dva motory podle nákresu. Jeden bude simulovat levé kolo a druhý pravé kolo.

# VIRTUÁLNÍ ROBOT (HARDWARE) – ŘEŠENÍ

## DVĚ ČIDLA



## DVA MOTORY



Pozor!!!

Až budeme zapojovat skutečného robota, tak motory i čidla budeme zapojovat malinko jinak. Toto zapojení funguje pouze ve virtuálním prostředí.

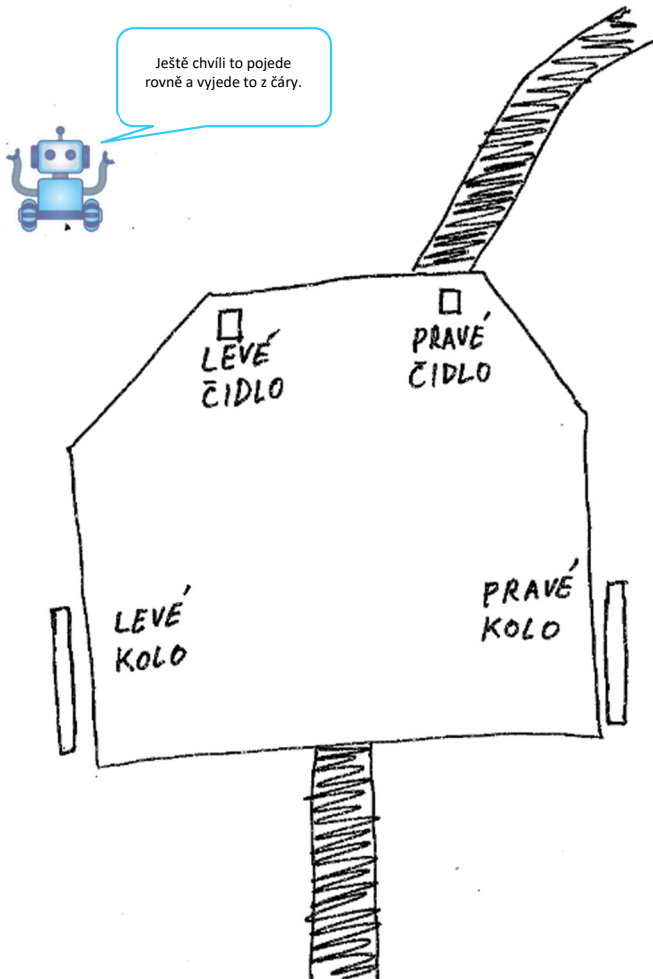


## VIRTUÁLNÍ ROBOT (ŘÍZENÍ)

### VLEVO NEBO VPRAVO?

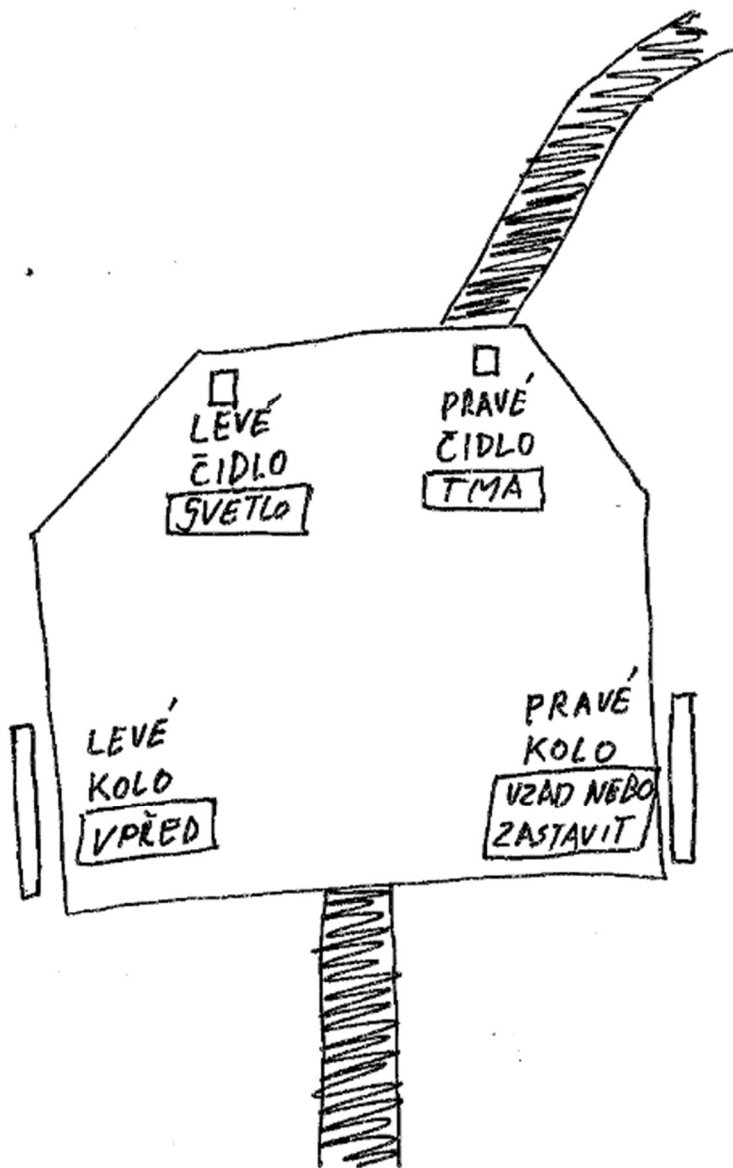
Zamysli se:

Podívej se na obrázek. Zapiš do něj, které čidlo bude snímat světlo, a které tmu. Dále do něj zapiš, kterým kolem budeme otáčet, aby se robot udržel na čáře.



# VIRTUÁLNÍ ROBOT (ŘÍZENÍ) – ŘEŠENÍ

VLEVO NEBO VPRAVO?

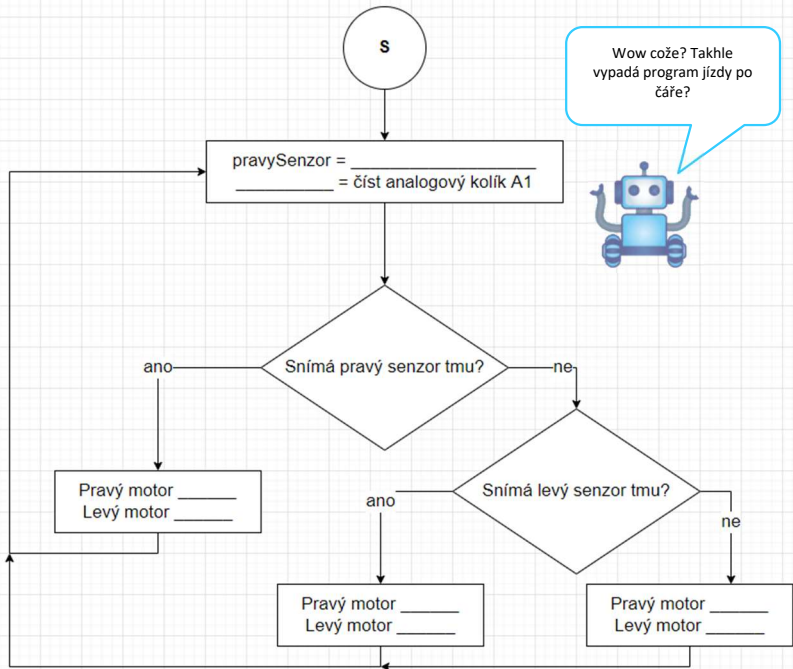


# VIRTUÁLNÍ ROBOT (DIAGRAM)

## VÝVOJOVÝ DIAGRAM

Úkol:

Doplň vývojový diagram programu.

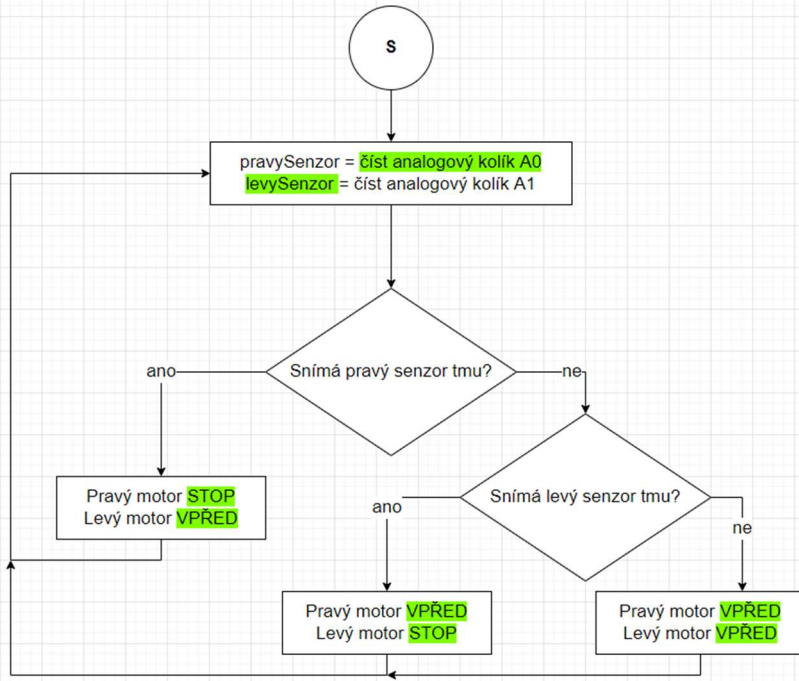


Nápověda:

Slova, která chybí v diagramu: VPŘED 4x, STOP 2x, levýSenzor, číst analogový kolík A0

# VIRTUÁLNÍ ROBOT (DIAGRAM) – ŘEŠENÍ

## VÝVOJOVÝ DIAGRAM

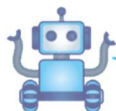
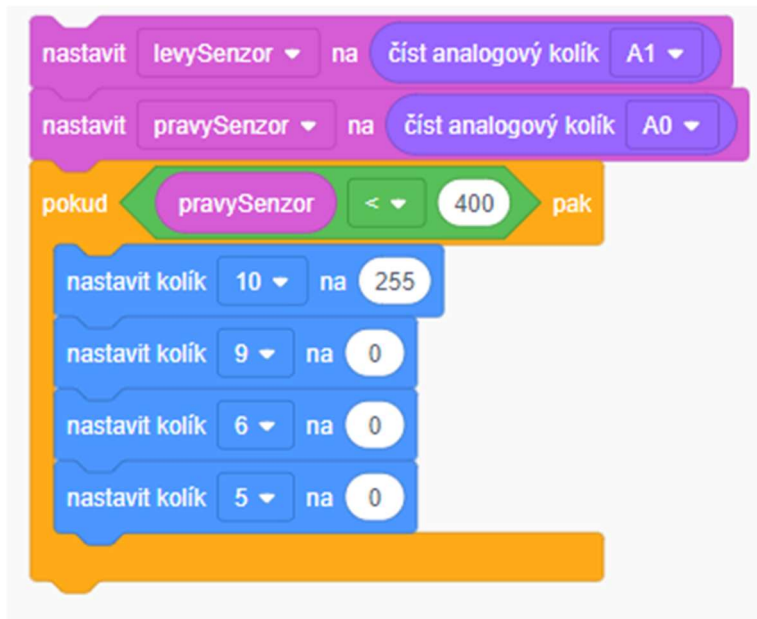


## VIRTUÁLNÍ ROBOT (SOFTWARE)

### PROGRAM

Úkol:

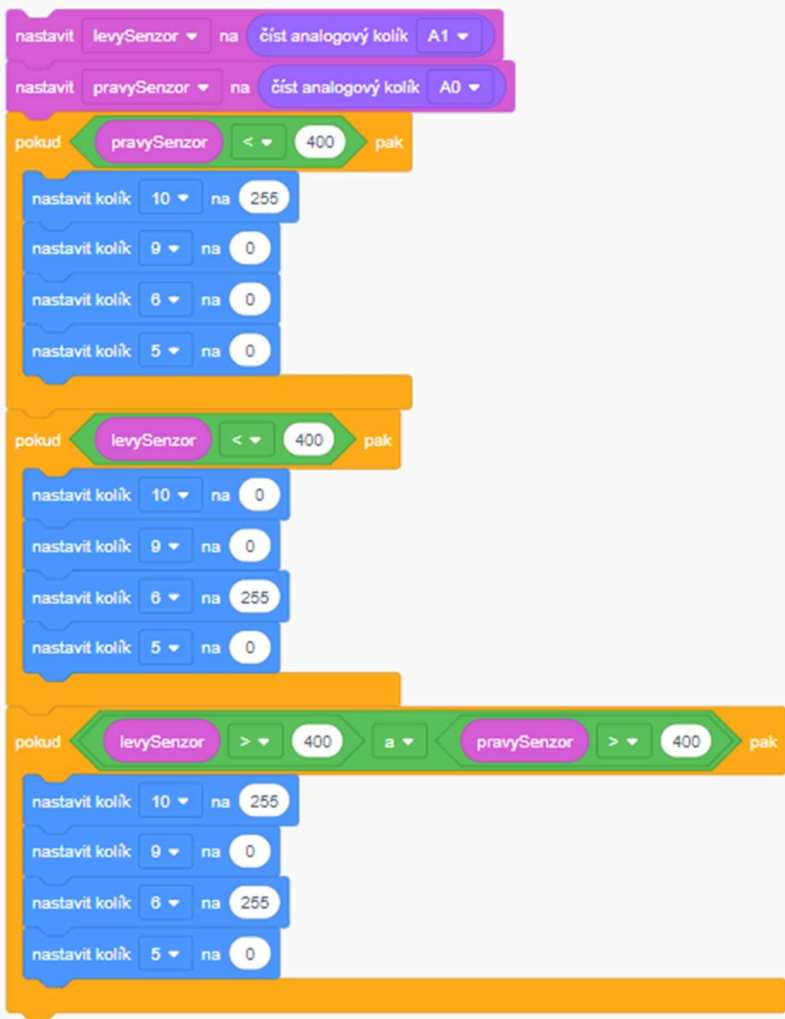
Vývojový diagram z předchozího úkolu naprogramuj pomocí bloků v TinkerCad. Na obrázku je nápověda jak na to.



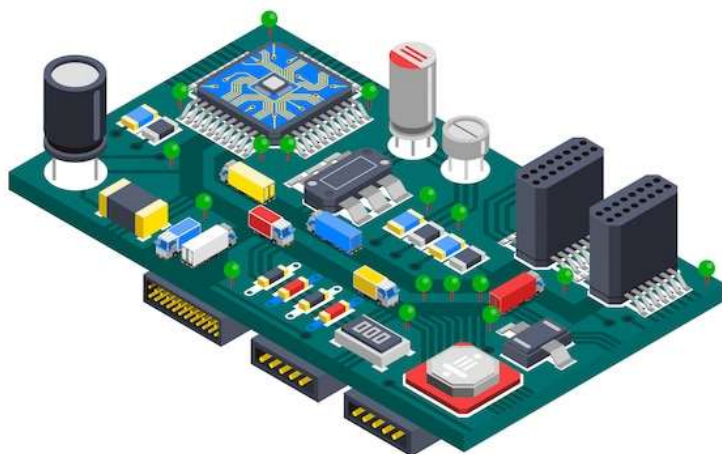
Pokud pravý senzor je na tmavé čáře, tak musím zatočit doprava.

# VIRTUÁLNÍ ROBOT (SOFTWARE) – ŘEŠENÍ

## PROGRAM



## SKUTEČNÉ ZAPOJENÍ

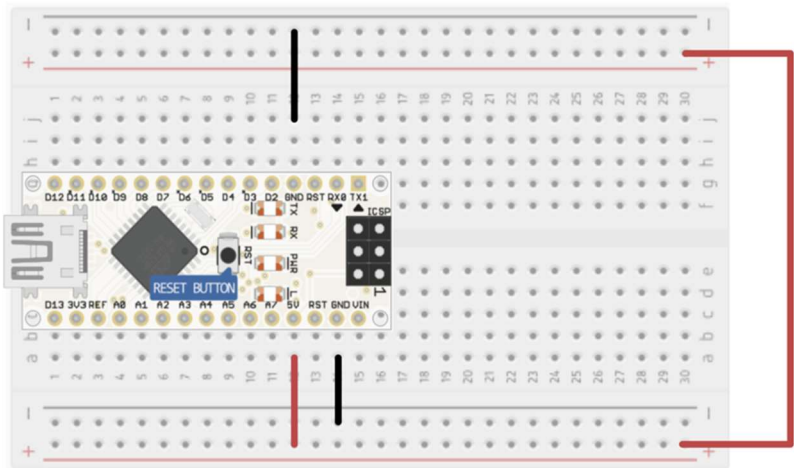






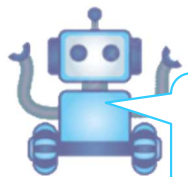
# ZÁKLADY

## NAPÁJENÍ A ARDUINO



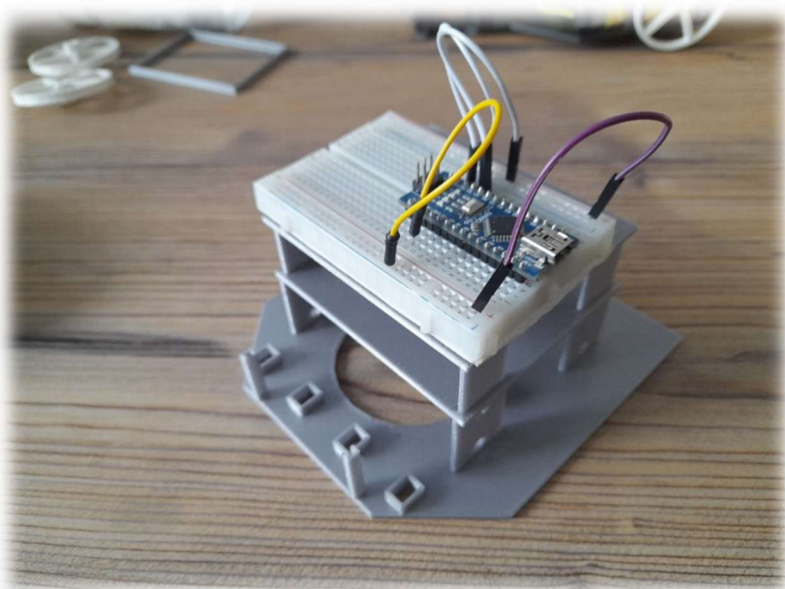
Úkol:

Umístí na podvozek kontaktní pole. Do pole usad' Arduino a propoj jeho napájení tak jak je na obrázku.



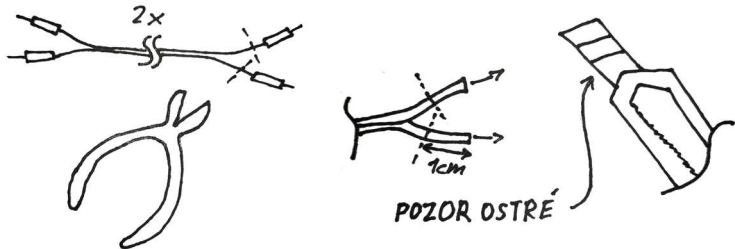
A hele to Arduino vypadá jinak než v TinkerCadu.

## ZÁKLADY – ŘEŠENÍ



## MOTORY

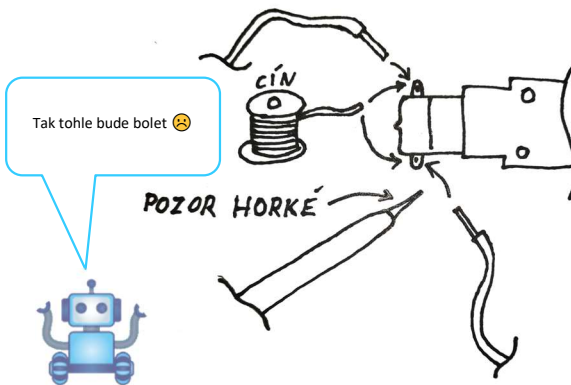
### PŘÍPRAVA VODIČŮ



Úkol:

Připrav si vodiče, kterými připojíme motory. Nejdříve jim odstříhni na jedné straně konce a poté odizoluj gumové bužírky.

### PÁJENÍ VODIČŮ

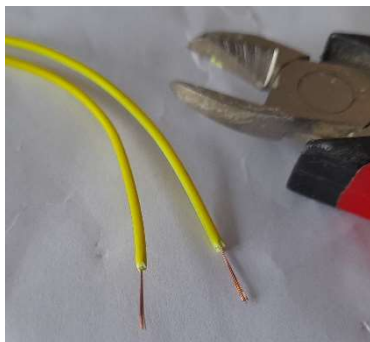


Úkol:

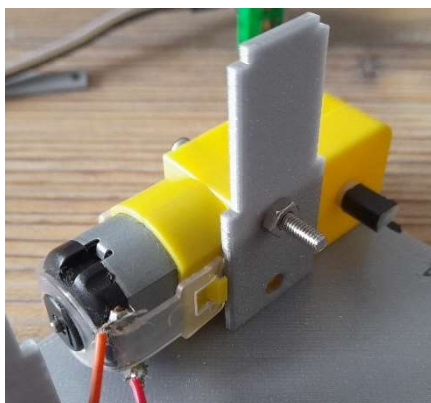
Konce připravených vodičů připájej na dva plíšky u hlavy motoru. Výsledný celý motor usaď do podvozku a zajisti šroubem a matkou.

## MOTORY - ŘEŠENÍ

### PŘÍPRAVA VODIČŮ

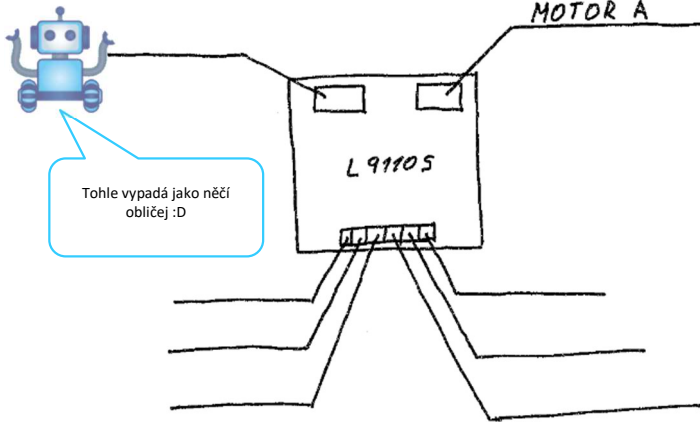


### PÁJENÍ VODIČŮ



# H-MŮSTEK

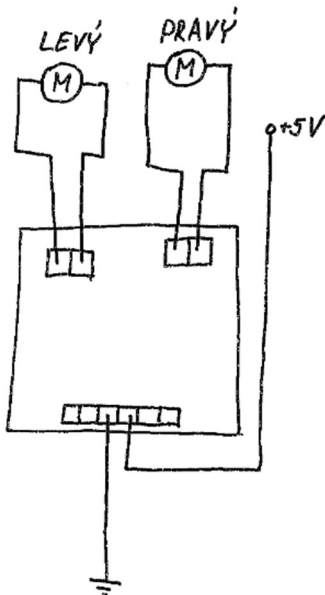
## VÝVODY



Úkol:

Podívej se na obvod L9110S a pojmenuj jednotlivé vývody.

## PŘIPOJ NAPÁJENÍ A MOTORY

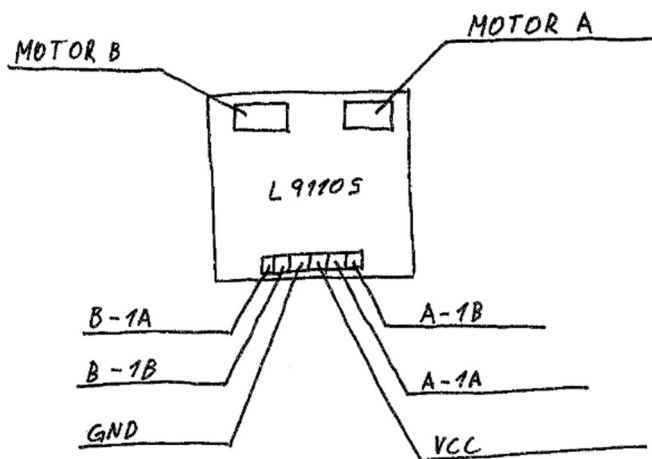


Úkol:

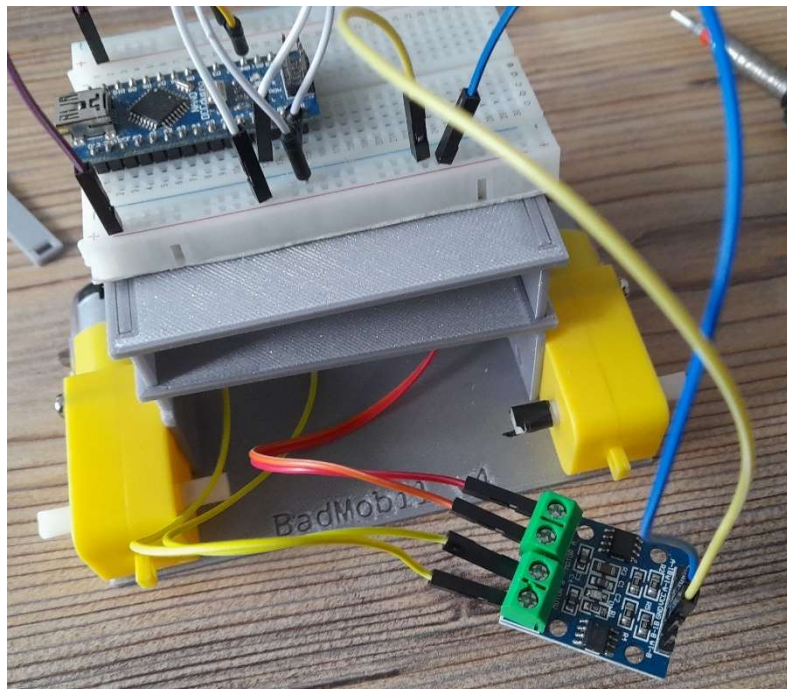
Připoj k odpovídajícím vývodům plus a minus a do velkých svorek připoj jeden a druhý motor.

## H-MŮSTEK – ŘEŠENÍ

### VÝVODY

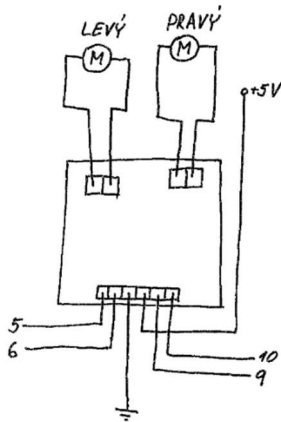


### PŘIPOJ NAPÁJENÍ A MOTORY



## PŘIPOJENÍ A TEST

### ZAPOJ ŘÍZENÍ



Úkol:

Připoj ovládací piny H-můstku k pinům Arduino. Jsou to ty stejné piny, které jsme používali v TinkerCadu.

### TEST MOTORŮ

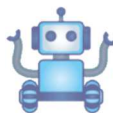
Úkol:

Nahrej do Arduino program `test_motory.ino`. Oba motory by se měly otáčet nejdříve 5 vteřin vpřed a potom 5 vteřin vzad.

### OPRAVA OTÁČENÍ

Úkol:

Pokud se ti motory otáčely jinak, než by měly, stačí prohodit piny na Arduino. Tzn. Prohodit 5 a 6 nebo 9 a 10, podle toho, který motor se točil špatně.



Takže, když přehodím dráty, tak se motor začne točit obráceně?

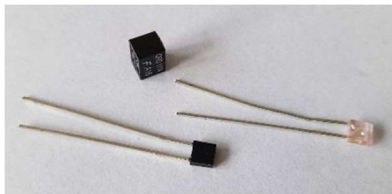






## SENZOR ČÁRY I. – ŘEŠENÍ

### ROZLOŽ TO NA ČÁSTI



1. Rezistor a pouzdro
2. Laser a kondenzátor
3. LED dioda a pouzdro

### CO KAM ZAPOJIT

Jmenuje se: **anoda** katoda

Zapojuje se k: ~~minus~~ **plus**

### PŘEDŘADNÝ ODPOR DIODY

Zapamatuj si:

Odpory se nevyrábí všech hodnot, ale pouze v tzv. hodnotových řadách. Vždy proto volíme nejbližší větší hodnotu odporu. V tomto případě je to 330Ω.

$$R = \frac{3}{0,01}$$
$$R = 300\Omega$$

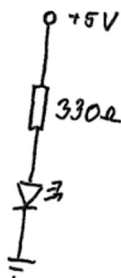
## SENZOR ČÁRY II.

### UMÍSTI SENZORY

Úkol:

Vezmi dva senzory čáry a umísti je do vnějších pozic v podvozku robota.

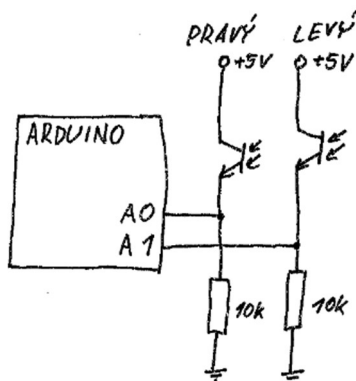
### ZAPOJENÍ LED DIODY



Úkol:

Připoj LED diodu (průhledná součástka) k napájení. Nezapomeň mezi plus a diodu zapojit předřadný rezistor, který jsme vypočítali.

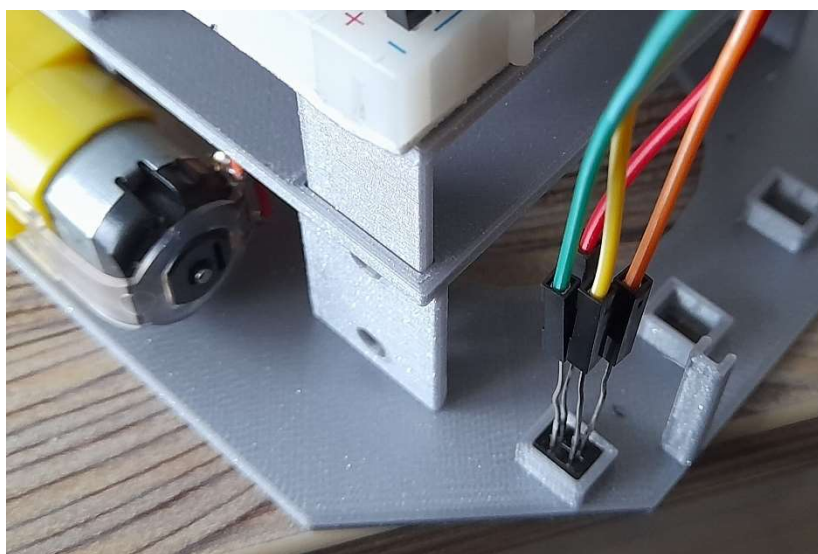
### ZAPOJENÍ FOTOTRANZISTORU



Úkol:

Připoj fototranzistor (černá součástka) podle obrázku. Delší nožičku k rezistoru a plus, kratší nožičku k minus. Podobně jsme zapojovali v TinkerCadu fotorezistor.

## SENZOR ČÁRY II. - ŘEŠENÍ



## TEST SENZORŮ

### NEVIDITELNÉ SVĚTLO

Úkol:

Připoj Arduino ke zdroji napájení. Zapni si na telefonu foťák a podívej se přes kameru foťáku ze spodu na čidla. Obě by měla svítit fialovobílou barvou.



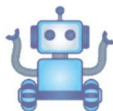
### TEST

Úkol:

Nahrej do Arduina program `test_cidla.ino`. Zakryj rukou nejdříve levé a potom pravé čidlo. Popiš, co se děje.

Když zakryji levé čidlo točí se \_\_\_\_\_ motor.

Když zakryji pravé čidlo točí se \_\_\_\_\_ motor.



Hustý, světlo co není vidět, ale přitom je vidět přes telefon...

## TEST SENZORŮ – ŘEŠENÍ

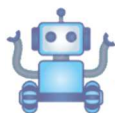
### TEST

Když zakryji levé čidlo točí se **levý** motor.

Když zakryji pravé čidlo točí se **pravý** motor.

#### Úkol:

Pokud se motory točí obráceně. To znamená při levém čidle se točí pravý motor atd... Tak stačí přehodit vstupy čidel do Arduina. Prohod' tedy Vodiče vedoucí do A0 a A1 a zkus test znovu.



Někdy mám pocit, že se všechny robotí problémy řeší jenom přehazováním drátů.

## TŘETÍ BOD

Úkol:

Namontuj třetí opěrný bod tak jak je na obrázku. Destičku opatrně zakápní vteřinovým lepidlem a počkej, dokud nezatuhne.







## PRAHOVÁ HODNOTA

### MĚŘENÍ SVĚTLA

Úkol:

Nahrej do Arduina program prah.ino. Otevři si sériový monitor a střídatě pokládej robota čidly na černou a bílou plochu. Napiš si jaká je minimální a maximální hodnota pro levé a pravé čidlo.

	Minimální hodnota	Maximální hodnota
Levé čidlo		
Pravé čidlo		

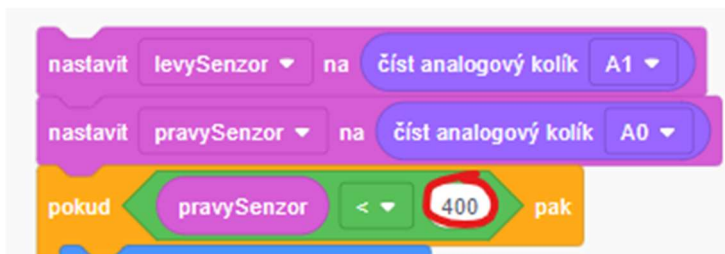
### PRAHOVÁ HODNOTA

Úkol:

Podívej se na zapsané hodnoty. Urči hodnotu, která je přibližně uprostřed mezi minimální a maximální hodnotou.

Prahová hodnota: \_\_\_\_\_

### ÚPRAVA PROGRAMU



Úkol:

Nově zjištěnou prahovou hodnotu zadej do podmínek programu, který jsi tvoří(a) v TinkerCadu.

## PRAHOVÁ HODNOTA – ŘEŠENÍ

### MĚŘENÍ SVĚTLA

Pozor:

Každému mohou vyjít trochu jiná čísla, ale vždy by minimální hodnota měla být menší než 100 a maximální větší než 500.

	Minimální hodnota	Maximální hodnota
Levé čidlo	87	970
Pravé čidlo	83	930

### PRAHOVÁ HODNOTA

Prahová hodnota: 600

Zapamatuj si:

Při této hodnotě se program rozhodne, zda čidlo kouká na tmavý, nebo světlý podklad.

### ÚPRAVA PROGRAMU

Nezapomeň:

Nezapomeň prahovou hodnotu změnit ve všech podmínkách. Dohromady změníš prahovou hodnotu 4x.

## PROGRAM KAČENKA

### STÁHNOUT KÓD

Úkol:

Pokud jsi vše zapojil(a) tak, jak je napsáno v tomto sešitě, tak stačí nahrát do Arduina program Kacenka, který jsme programovali v TinkerCadu. Otevři si svůj projekt a v sekci kód stiskni tlačítko



### NAHRÁT

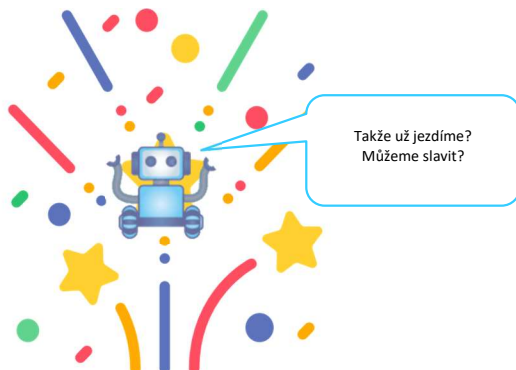
Úkol:

Webový prohlížeč buď kód přímo stáhne, pak ho najdeš ve stažených souborech nebo se tě zeptá, kam ho chceš uložit, pak ho uloží třeba na plochu. Tento kód si otevři v programu Arduino IDE a nahraj jej do robota.

### POLOŽ ROBOTA NA ČÁRU A SPUŠŤ

Úkol:

Polož robota na čáru a připoj ho k PowerBance. Měl by být schopen čáru sledovat. Pokud se občas zamotá, nic se neděje, občas se to stává.



## ROBOT NEJEDE? 😞

Rada:

Pokud robot nejede, musíš najít důvod proč tomu tak je. Zkus zkontrolovat následující věci.

### KONTROLA ZAPOJENÍ

Znovu si překontroluj zapojení. Zvláště u čidel čáry drátky často vypadávají.

### ENERGIE V POWERBANCE

Koukni, jestli máš dostatečně nabitou powerbanku.

### VADNÁ SOUČÁSTKA

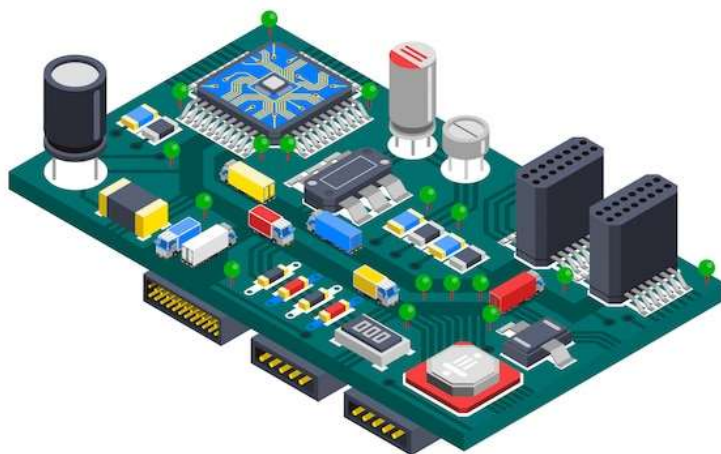
Pokud máš podezření, že nějaká součástka nefunguje, zkus ji vyměnit za jinou.

### NOVÉ TESTOVÁNÍ

Vyzkoušej nahrání všech předchozích testů, jestli fungují tak jak mají.

# SKUTEČNÉ ZAPOJENÍ

## - SENZOR PŘEKÁŽEK -





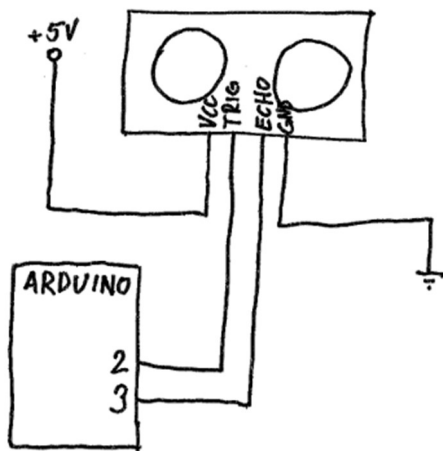
## ULTRAZVUKOVÝ SENZOR

### VÝVODY

Úkol:

Podívej se na součástku ultrazvukového senzoru. Napiš, jaké má vývody.

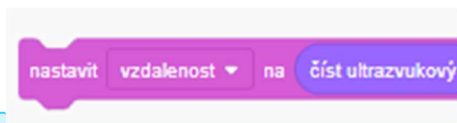
### ZAPOJENÍ



Úkol:

Zapoj ultrazvukový senzor podle obrázku.

### NOVÁ PROMĚNNÁ



Úkol:

V programu si vytvoř novou proměnnou s názvem vzdalenost. Najdi ve  Vstup blok pro čtení ultrazvukového senzoru. Nastav jí novou proměnnou.

## ULTRAZVUKOVÝ SENZOR – ŘEŠENÍ

### VÝVODY

VCC TRIG ECHO GND

Zapamatuj si:

VCC a GND jsou napájecí vývody. TRIG spouští měření a vysílá ultrazvukový signál. Na pinu ECHO se objevuje signál tak dlouho, dokud zvuk nedorazí zpátky od překážky.

### ZAPOJENÍ



### NOVÁ PROMĚNNÁ

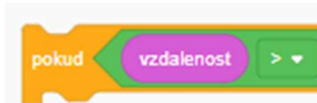
nastavit vzdalenost ▼ na čísl ultrazvukový snímač vzdálenosti na spouštěcím kolíku 2 ▼

echo kolíku 3 ▼ v jednotkách cm ▼



## POZOR PŘEKÁŽKA

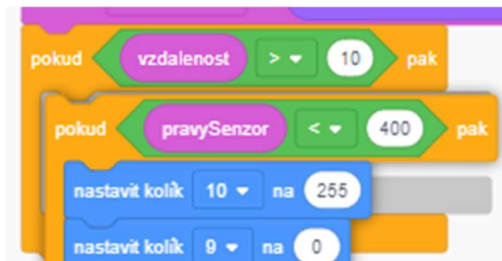
### POKUD NEVIDÍŠ PŘEKÁŽKU



Úkol:

Vytvoř podmínku pokud je překážka dál než 10 cm.

### TAK JEĎ NORMÁLNĚ



Úkol:

Do této podmínky vlož všechny dosud vytvořené podmínky.

### POKUD VIDÍŠ PŘEKÁŽKU, ZASTAV

Úkol:

Připrav si podmínku pokud je vzdalenost menší než 10, pak robot zastaví.

# POZOR PŘEKÁŽKA – ŘEŠENÍ

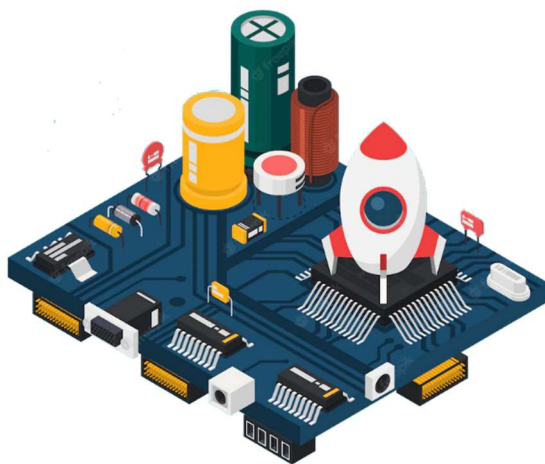
## KOMPLETNÍ

The script is composed of several initialization and conditional blocks:

- Initialization:**
  - nastavit levýSenzor na čísl analogový kolík A1
  - nastavit pravýSenzor na čísl analogový kolík A0
  - nastavit vzdálenost na čísl ultrazvukový snímač vzdálenosti na spouštěcím kolíku 2 echo kolíku 3 v jednotkách cm
- Distance > 10:** If the distance is greater than 10, the script proceeds to the next step.
- Right Sensor < 400:** If the right sensor value is less than 400, it sets pins 10, 9, 6, and 5 to 255, 0, 0, and 0 respectively.
- Left Sensor < 400:** If the left sensor value is less than 400, it sets pins 10, 9, 6, and 5 to 0, 0, 255, and 0 respectively.
- Both Sensors > 400:** If both sensor values are greater than 400, it sets pins 10, 9, 6, and 5 to 255, 0, 255, and 0 respectively.
- Distance < 10:** If the distance is less than 10, it sets pins 10, 9, 6, and 5 to 0, 0, 0, and 0 respectively.

# SKUTEČNÉ ZAPOJENÍ

## - 4 ČIDLA -



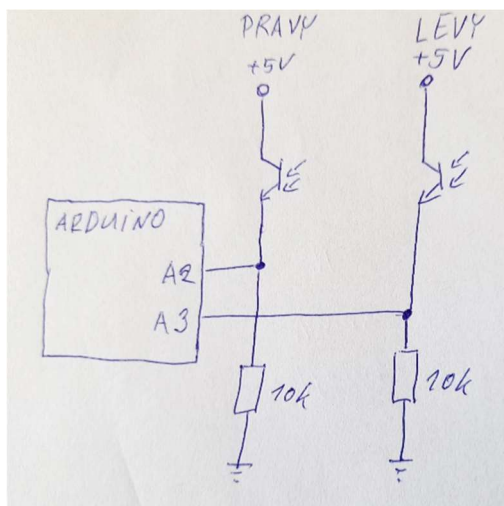


## ZAPOJENÍ DALŠÍCH ČIDEL

### PŘIPOJENÍ ČIDEL

Úkol:

Vezmi dva senzory čáry a umísti je do vnitřních pozic v podvozku. Připoj je k Arduino podle nákresu. Nezapomeň připojit LED diodu.

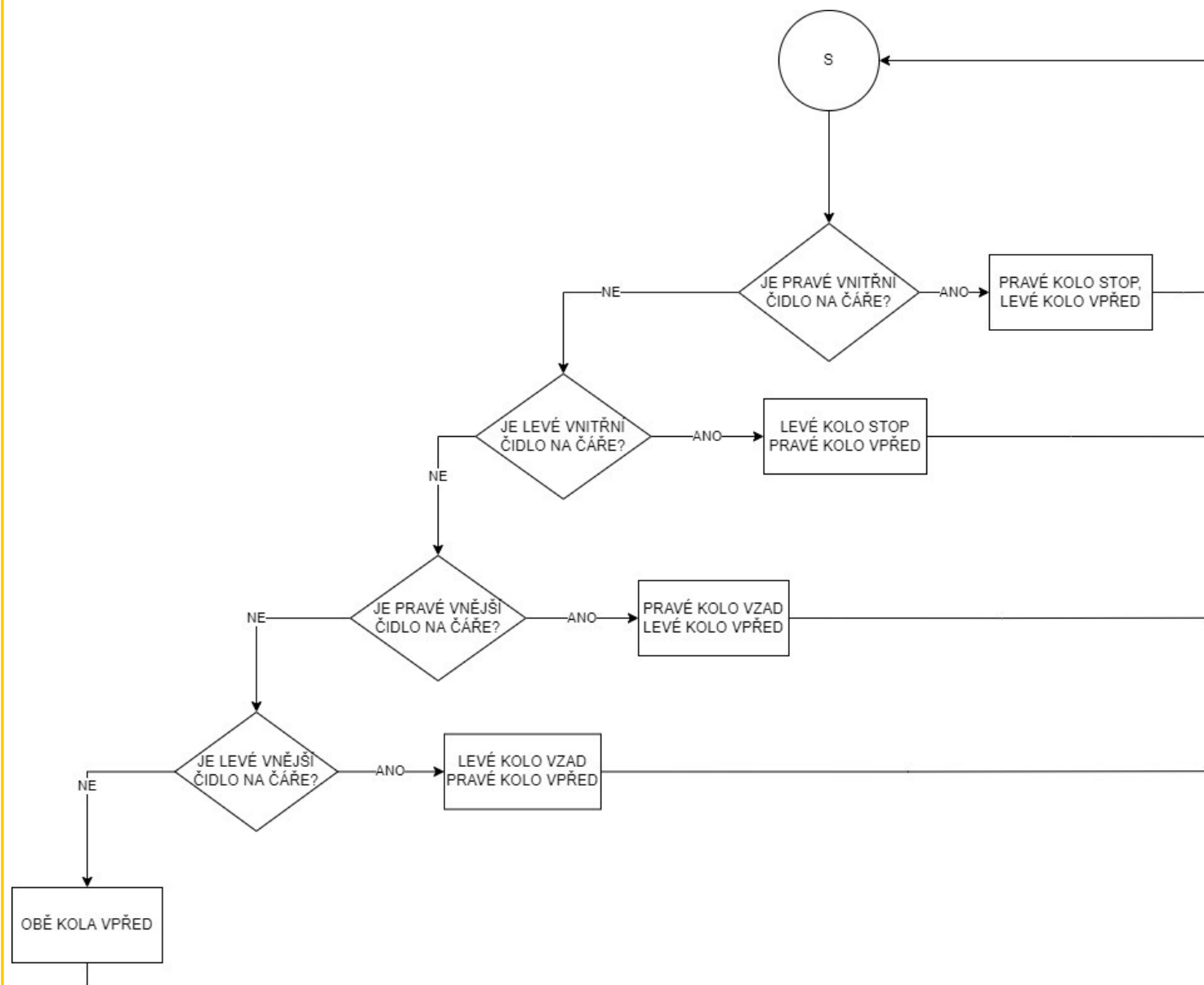


### JAK NAPROGRAMOVAT 4 ČIDLA?

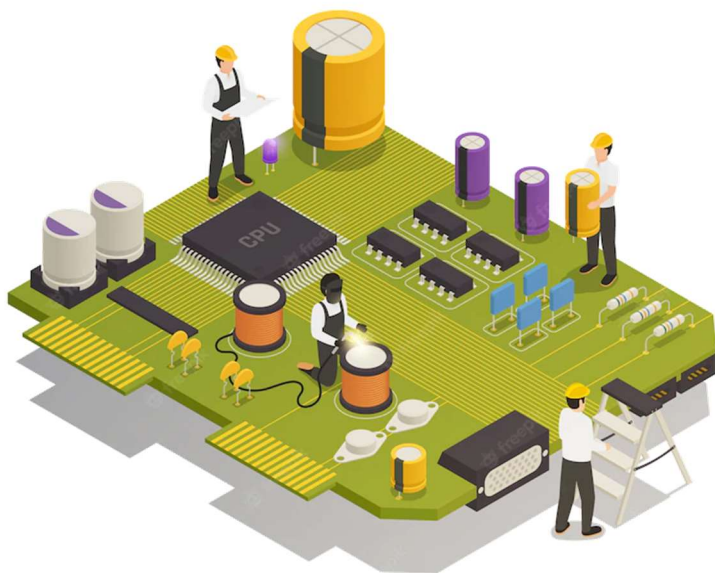
Je tu několik možností, co s tím.

1. Vylepšená kačenka – diagram na další stránce
2. P-regulace – složitý, ale skvělý systém řízení, není součástí tohoto sešitu.
3. Jakýkoli jiný algoritmus, který si vymyslíš – experimentuj je to zábava.

## VYLEPŠENÁ KAČENKA - DIAGRAM



# SKUTEČNÉ ZAPOJENÍ - ENKODÉR -







## DALŠÍ 4 ČIDLA

### ZAPOJENÍ

Úkol:

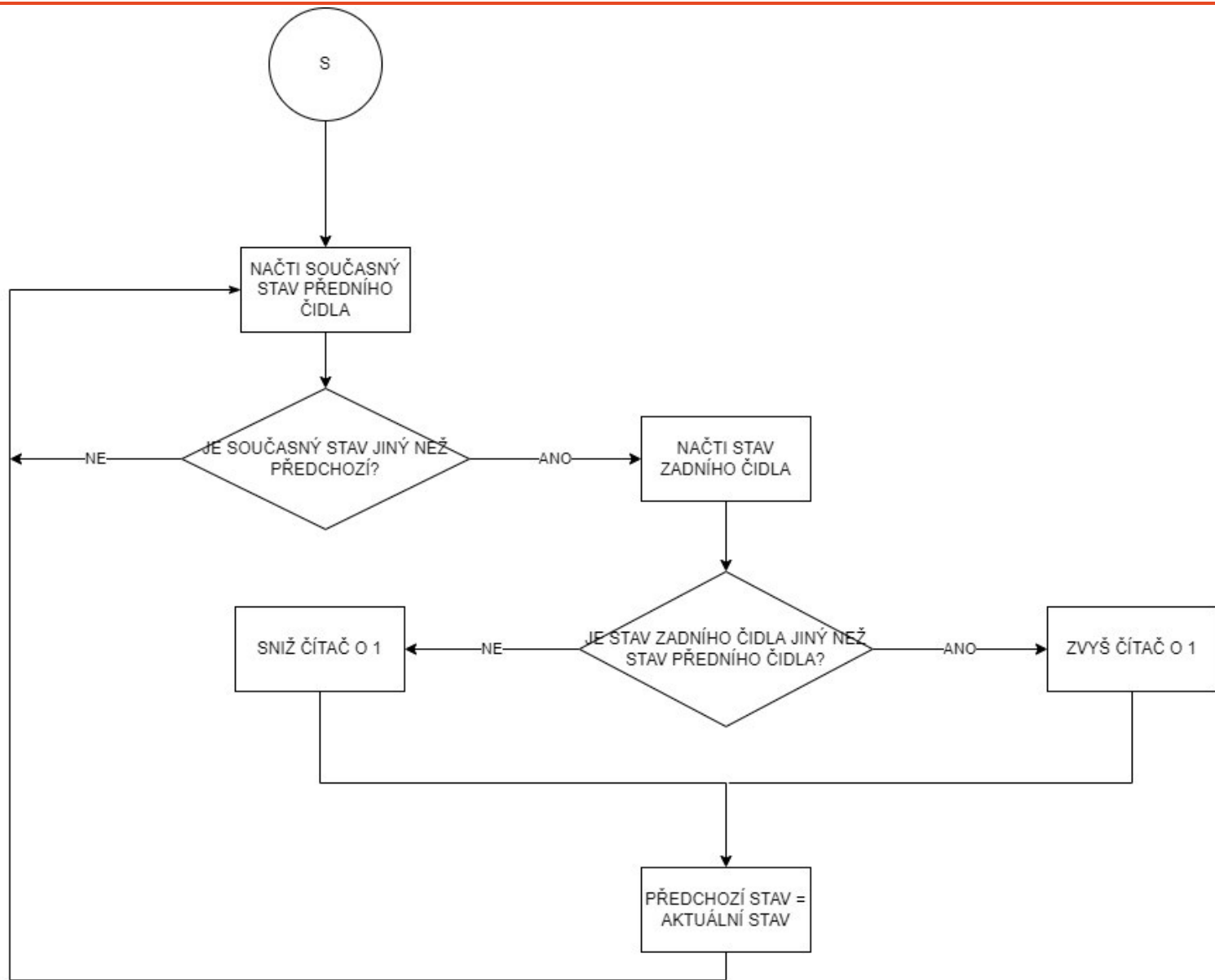
Připoj další 4 čidla vždy dvě spolu, tak, aby směřovaly směrem k loukotím kola. Připoj je k analogovým vstupům.



### JAK JE NAPROGRAMOVAT

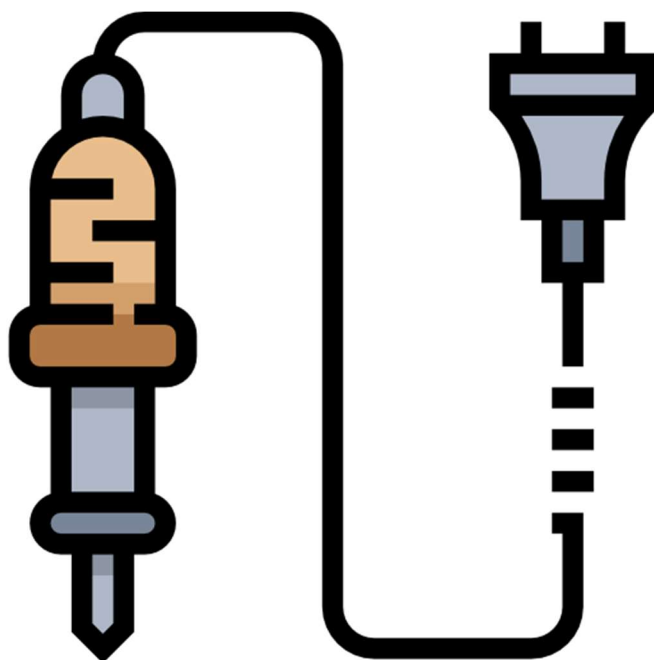
Enkodér určuje směr otáčení a úhel natočení kola. Vývojový diagram na další stránce.







# SKUTEČNÉ ZAPOJENÍ - UNIVERZÁLNÍ PÁJIVÉ POLE





## VYPADLÉ DRÁTKY

Otázka:

Už tě štvě, že většina problémů, která s robotem byla, byla způsobena vypadlými drátky z nepájivého pole nebo z čidel? Vyzkoušej tedy univerzální pájivé pole, kde všechny propojky budou pevně držet na svém místě pomocí cínu. Při práci se ale nespál.

