



CZ NÁVOD K OBSLUZE

Programovatelný robot Yeti YT-3000



Obj. č. 19 11 07

Vážený zákazníku,

děkujeme Vám za Vaši důvěru a za nákup programovatelného robota.

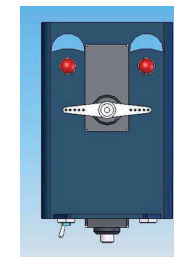
Tento návod k obsluze je nedílnou součástí tohoto výrobku. Obsahuje důležité pokyny k uvedení výrobku do provozu a k jeho obsluze. Jestliže výrobek předáte jiným osobám, dbejte na to, abyste jim odevzdali i tento návod k obsluze.

Ponechejte si tento návod, abyste si jej mohli znovu kdykoliv přečíst.



Popis funkce robota Yeti

Yeti je chodící robot pojmenovaný po sněžném muži, který podle legendy žil v Himalájích. Stejně tak jako legendární obr, umí i tento robot chodit na dvou velkých nohách. Robot je vybaven mozkiem (mikroprocesorem) a počítačovým programem, který ovládá všechny servomotory, jež pohánějí jeho chodidla.



Technické údaje

Pohon	2 servomotory (napájení 5 V)
Procesor	ATmega8L
Programovací jazyk	C
Zdroj napájení	4 akumulátory formátu AA (4,8 – 6 V)
Spotřeba proudu	min. 10 mA, max. 600 mA
Komunikace	IR a komunikační sběrnice 12C
Rozšíření	možnost připojení 2 modulů (plochý kabel)
Výška	278 mm
Šířka	155 mm
Hloubka	100 mm

Robot během chůze provádí realistické kroky stejné jako člověk. Ovládání pohybu zajišťují celkem 2 servomotory řízené mikroprocesorem. Motory mají vlastní převodovku, která zajišťuje velmi vysoký výkon motorů. Rychlost a směr otáčení servomotorů podporuje elektronický obvod s PWM (Pulse Width Modulation). Tato technologie umožňuje motorům provádět velmi přesné kroky. Robot má přední a spodní servo. Přední servo zajišťuje pohyb chodidel směrem vzhůru. Pohyb vpřed a vzad provádí spodní neboli „leg“ servo.

Možnosti robota

Do systému robota můžete načíst testovací program a stejně tak i vaše vlastní programy. Robot navíc umožňuje připojení dalších přídavných modulů, díky kterým se dokáže vyhybat překážkám po detekci a měření celkové vzdálenosti od těchto objektů. K systému robota však můžete připojit i vlastní elektronické moduly. Robot může bezdrátově komunikovat s počítačem prostřednictvím bezdrátové IR technologie. Ovládat jej pak můžete nejen z vašeho počítače, ale například i pomocí dálkového ovladače. Robot umí přehrávat různé melodie a generovat systémové zvuky. Po zapnutí systému se rozsvítí LED v jeho očích. Vlastním experimentováním můžete změnit hardwarovou výbavu robota, jeho tvář a přidat mu například ústa nebo LCD displej.

Robota můžete aktivovat ve 3. jednoduchých krocích. Jako první přitom musíte sestavit mechanické a elektronické komponenty robota, připojit jej ke zdroji napájení a zapnout robota hlavním vypínačem, který je umístěn v jeho spodní části. Pokud budete používat k napájení robota akumulátory, zajistěte nejprve jejich nabití na 100% kapacitu. Po úspěšném sestavení a zapnutí si robot jako první nejprve protáhne nohy a poté předvede své možnosti a pohyby v rámci výchozího programu, který je součástí paměti procesoru. Při dalších úpravách a při programování robota zažijete skutečně spoustu zábavy!

Načtení testovacího programu do interní paměti robota

Pro načtení programu do paměti robota bude použito neviditelné, infračervené technologie. Proto však musíte připojit dodávaný COM-port adaptér, který je vybaven infračerveným RS232 transceiverem, do COM portu vašeho počítače. Robot má vestavěný IR transceiver, který je umístěn za malými otvory v jeho zádech. Namísto COM-port adaptéru můžete použít i kompatibilní USB modul. Robot ASURO, který patří mezi další programovatelné roboty z naší nabídky, používá stejný IR-transceiver systém.

Načtením programu do paměti robota dojde k přepsání stávajícího programu v interní paměti robota. To znamená, že původní testovací program tak bude nevratně vymazán. Testovací program však můžete kdykoliv znovu stáhnout a zavést do interní paměti robota prostřednictvím vašeho počítače.

Postup pro načtení programu do paměti robota

- Připojte COM-port adaptér (nebo USB adaptér) do vašeho počítače.
- Spusťte flash program v počítači.
- V počítači a flash programu konfiguruje COM port pro COM-port adaptér.
- Vyberte program YETI ve Flash programu.
- Umístěte robota tak, aby jeho otvory v zádech směřovaly na horní část COM-port adaptéru.
- Vypněte robota (Off) pomocí hlavního vypínače.
- V aplikaci vstupte do nabídky „Programming“.
- Během dalších 10. sekund robota zapněte (On).
- Systém zahájí automatický přenos programu do procesorové paměti robota.
- Vyčkejte do úplného načtení programu.
- Restartujte robota. Vypněte jej a poté znovu zapněte.
- Vyčkejte poté další 3 sekundy.

V případě, že načtení programu proběhlo úspěšně, robot automaticky spustí nově načtený program.

Robota můžete velmi snadno vybavit různými doplňky, které jeho schopnosti dále rozšiřují. Pokud robota vybavíte například ultrazvukovým transceiverem, dokáže následně rozeznávat různé překážky a umí se jim pak i vyhnout. Robota můžete doplnit i o displej, na kterém robot může zobrazovat různé zprávy a data. Přídavné moduly jsou tvořeny obvody s tištěnými spoji (PCB) a různými elektronickými komponenty. Tyto moduly jsou konstruovány pro připevnění k hlavě robota do oblastí jeho očí. Sami si pak můžete některý elektronický PCB modul jednoduše vyrobit. Tyto moduly se pak připojují do horní části hlavy a tvoří tak jakousi „střechu“ hlavy. Moduly se k robotovi připojují pomocí plochého kabelu do základní desky a mikroprocesorové 12C sběrnice v hlavě robota.

Komunikace mezi PC a robotem

Po stisku tlačítka „Programming“ ve flash programu dojde k aktivaci a přenosu dat během dalších 10 sekund. Pokud dojde k odezvě systému robota, spustí se samotný přenos programu. Robot a jeho systém zajistí odezvu během 3. sekund pouze v případě, že dojde k jeho restartu (vypnutí a zapnutí). Pokud však nedojde k navázání komunikace robota s počítačem v průběhu 3. sekund, spustí jeho systém automaticky program, který je aktuálně uložený v jeho interní paměti. Pokud však nebude systém robota nijak reagovat, vygeneruje flash program chybové hlášení po uplynutí intervalu 10. sekund. Aby nedošlo k potížím při přenosu programu, postupujte vždy podle následujících pokynů:

- Zajistěte optimální viditelnost mezi IR vysílačem a přijímačem. Odstraňte proto různé překážky nebo umístěte robota tak, aby jeho transceiver měl dostatečný vizuální kontakt s USB-port adaptérem.
- Použijte nejnovější verzi flash software.
- Nabijte akumulátory na plnou kapacitu.
- Odstraňte systém z blízkosti zdrojů viditelného světla (například úsporných žárovek).
- Použijte USB adaptér (port RS-232 je napájen nízkou hodnotou napětí).

YETI je robot, který stojí a dokáže chodit na obou nohou všemi směry. Každý jeho krok je prováděn vyvážením druhé nohy, která právě stojí na zemi. Veškerý pohyb během chůze zajišťují pouze 2 servomotory. Servomotor je speciální druh motoru, který je vybaven vlastní převodovkou. Převodovka zajišťuje regulaci rychlosti a točivého momentu. Tyto motory jsou navíc vybaveny elektronickými komponenty s řízením impulzu, což jim umožňuje použití velmi přesné pozice a úhlu rotace motoru resp. částí robota. Servo umístěné v přední části robota zajišťuje pohyb jeho chodidel („feet-servo“). Druhé servo („legs-servo“) ve spodní části dává pohyb nohám (včetně chodidel) v postupném pořadí.

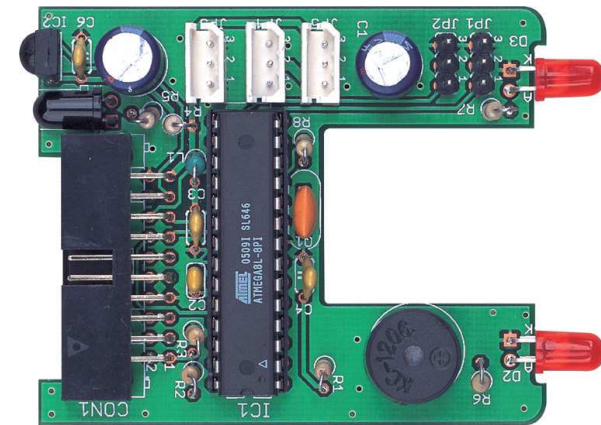
Základ chůze

Robot dokáže velmi dobře udržovat svou stabilitu ve chvíli, kdy stojí na jedné noze a druhá noha provádí pohyb. Při chůzi vyvíjí tlak na vnější část „vyvažujícího“ chodidla směrem nahoru a současně přitom vyvíjí tlak na vnější část „pohyb provádějícího“ chodidla směrem dolů. Díky tomuto pohybu dochází k naklánění těla robota nad „vyvažující“ chodidlo, které v té chvíli nese většinu robotovy hmotnosti. V další fázi robot přesune jeho „pohyb provádějící“ chodidlo zpět a dokončí tak krok. Obě nohy si pak jednoduše vymění úlohy a tento proces se obdobným způsobem opakuje neustále dokola. Během chůze se pak tyto procesy neustále opakují v nekonečné smyčce.

Hardware

Základní deska robota

Robot má základní desku osazenou mikroprocesorem Atmega8L, který je připojen k velkému generátoru zvuku a dvěma červeným LED v přední části. Většina vodičů vedoucích z připojovacích pinů procesoru (například sběrnice 12C) je vedena do 20-pinového konektoru v zadní části. Tento konektor zároveň slouží jako port pro připojení ostatního hardware pomocí plochého kabelu. Oba černé konektory (JP1, JP2), které jsou v blízkosti červených LED zajišťují připojení k oběma servomotorům. Bílé konektory (celkem 3) slouží pro připojení hlavního vypínače a pro připojení obou napájecích článků (aku-packů).



Připojení pomocí plochých kabelů

Konstruktéři tohoto robota ponechali možnost pro připojení dalších přídavných modulů. Pin 19 je určen pro VCC, pin 7, 8 a PIN 20 pro GND. Všechny zbývající konektory jsou připojeny do příslušných pinů mikroprocesoru. Číselné označení pinů je v dalším seznamu uvedené s odpovídajícími funkcemi pinů mikroprocesoru. Piny mikroprocesoru mohou být použity v různých režimech a to v závislosti na programovacím režimu. Více informací o tomto tématu získáte v datovém listu tohoto mikroprocesoru nebo jeho návodu.

Osazení konektoru pro připojení plochého kabelu

Pin 1	SCL
Pin 2	SDA
Pin 3	PC3(ADC3)
Pin 4	PC2(ADC2)
Pin 5	PC1(ADC1)
Pin 6	PC0(ADC0)
Pin 7	GND
Pin 8	GND
Pin 9	AVCC
Pin 10	PC6(RESET)
Pin 11	PB5(SCK)
Pin 12	PB4(MISO)
Pin 13	PB3(MOSI/OC2)
Pin 14	PD3(INT1)
Pin 15	PD6(AIN0)
Pin 16	D7(AIN1)
Pin 17	PD0(RXD)
Pin 18	PD1(TXD)
Pin 19	VCC
Pin 20	GND



Pokyny pro montáž elektronických komponentů

Pájení elektronických komponentů na desku

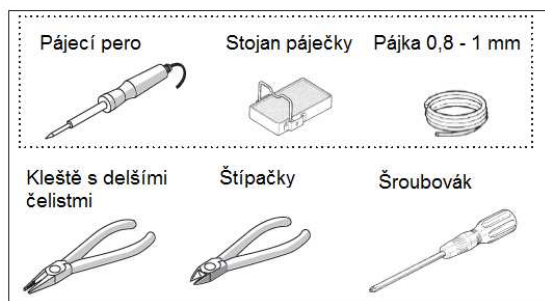
Uspořádání desky plošných spojů (PCB) velmi názorně ukazuje, kam je možné připojit jednotlivé elektronické komponenty. Při sestavování vždy dbejte přesné pozice a polaritu každého komponentu. Využít proto můžete dostupné schémata, obrázky a zvláštní pokyny. Při sestavování elektronických komponentů je vhodné začít vždy s nejmenšími pasivními součástkami. Zpravidla proto jako první je vhodné k desce připájet rezistory a následně kondenzátory. Po dokončení pájení všech komponentů zkrátíte přečnívající kusy vývodů / elektrod na straně spojů / pájení. Vždy při samotném pájení navíc musíte ponechat dostatečný prostor pro ostatní komponenty. Před pájením některých aktivních komponentů (tranzistory, diody, IC) doporučujeme vložit do desky všechny vývody / elektrody a teprve poté celý komponent naráz připájet. Přitom bude zapotřebí přizpůsobit vývody / elektrody součástky do určitého úhlu a ohybu. Pájení integrovaných obvodů (IC) ponechejte ideálně až na samotný konec.

Důležité! Zejména při pájení elektrolytických kondenzátorů a IC musíte zachovat přesnou polohu v obvodu a zejména polaritu. Pájení součástek nikdy nezačínáte s IC. Vždy, když to uspořádání součástek umožňuje použijte vhodnou patici (socket) pro IC. Vyhnete se tím případným chybám v zapojení jednotlivých pinů integrovaného obvodu. V případě, že špatně připájíte samotný IC do desky, je pak většinou velmi složité jej z desky znovu demontovat, aniž by nedošlo k jeho poškození. Jednotlivé piny IC můžete velmi snadno ohnout na hladkém a rovném povrchu pracovní desky.

Důležitá bezpečnostní opatření

Před uvedením tohoto výrobku do provozu si pečlivě přečtěte celý tento návod k obsluze. Děti mladší 14. let smějí tento výrobek sestavovat a používat pouze za neustálého dohledu dospělých! Při sestavování dbejte zvýšené opatrnosti, aby nedošlo k úrazu například v důsledku použití ostrých nástrojů (modelářských noží, kleští nebo nůžek) a při manipulaci s ostrými předměty a součástkami. Veškeré malé součástky odstraňte z dosahu malých dětí. Mohou se pro ně stát nebezpečnou hračkou. Nikdy k provozu robota nepoužívejte zároveň staré a nové baterie (akumulátory). V případě, že nebudete robota delší dobu používat, vyjměte z něj baterie.

Potřebné nářadí a příslušenství



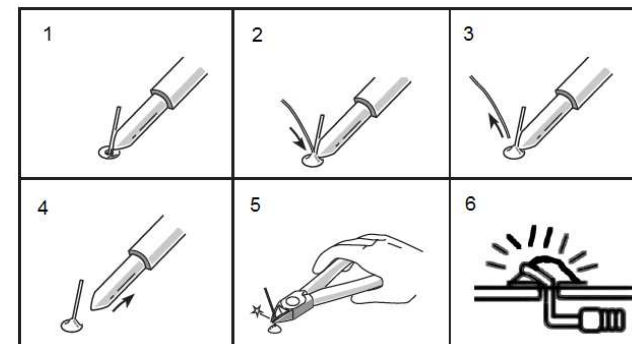
Technika pájení

Zachovejte správnou polohu (úhel) páječky při pájení jednotlivých součástek na desku.

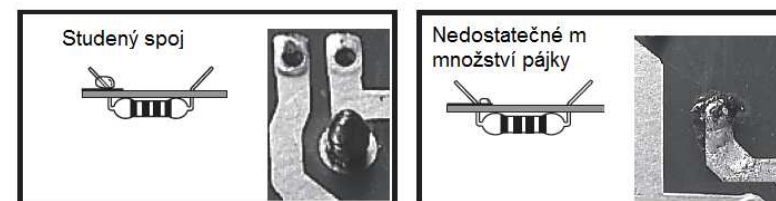
Pájecí pero přitom držte stejně jako běžné pero nebo propisku.

Postupně pak přidávejte potřebné množství pájky.

- Jako první dostatečně ohřejte vývod komponentu, který budete do desky pájet. Použijte proto hrot pájecího pera. Zabraňte extrémně dlouhému ohřevu!
- Přidejte určité množství pájky. Nepřidávejte však příliš velké množství pájky, neboť v takovém případě může dojít k nežádoucím spojům a zkratování jiných částí obvodu.
- Z pájeného místa odstraňte pájku.
- Teprve poté z místa odstraňte i hrot pájecího pera.
- Zkrátte přečnívající konce vývodů elektronické součástky.
- Kvalitně provedený pájený spoj má výrazně lesklý povrch pájky.



Nejčastější chyby při pájení



U studeného spoje pájka zůstane zpravidla jen na vývodu pájené součástky, ne však ve spoji i na desce. Při použití příliš malého množství pájky naopak nedojde k jejímu potřebnému vniknutí do všech míst ve spoji.

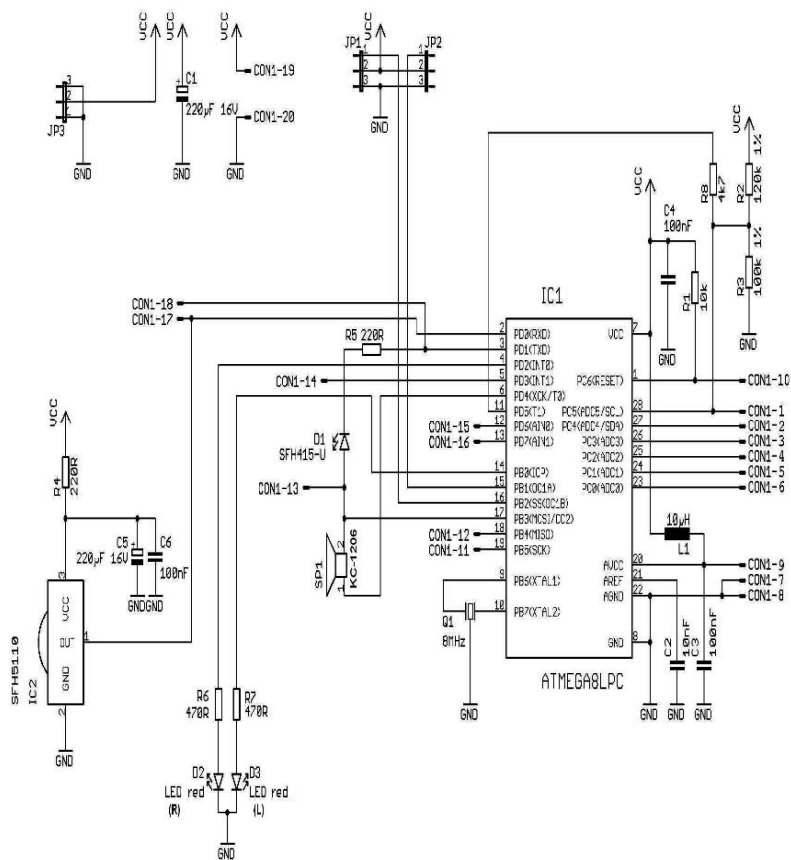


Při přemostění pájky (propojení) s jinými částmi obvodu může dojít k jeho zkratování a zároveň i nevratnému zničení některých elektronických komponentů (například IC). Dokonalý spoj má výrazně lesklý povrch pájky.

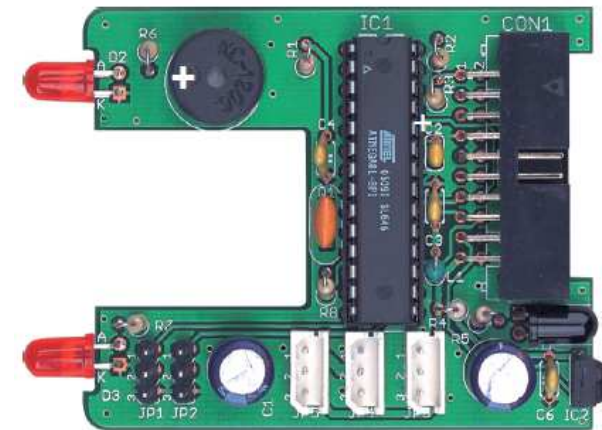
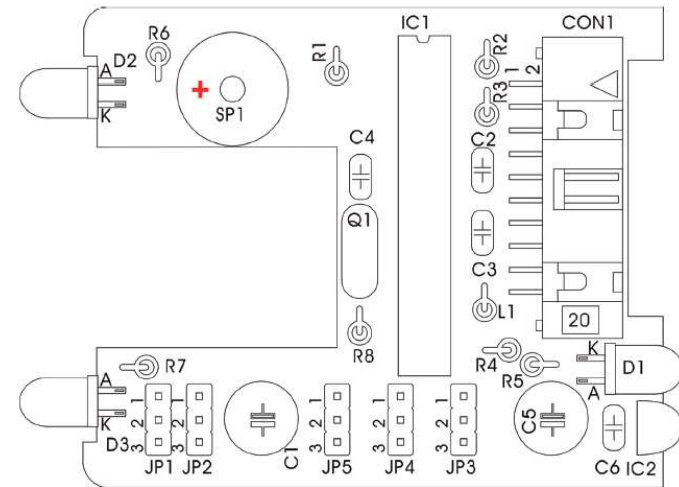
Seznam dodávaných komponentů

		Počet kusů
PCB1	Deska spojů PCB	1
IC1	ATmega8-I (při pájení dodržujte polaritu!)	1
IC2	SFH511 0 IR-receiver-IC (dodržujte polaritu!)	1
R1	1 0K / 0.25W / 5% (barvy proužků hnědý, černý nebo zlatý)	1
R2	120 K / 0.25W / 1% (hnědý, červený, černý nebo hnědý)	1
R3	100 K / 0.25W / 1% (hnědý, černý, černý nebo hnědý)	1
R4	220 R / 0.25W / 5% (červený, červený, hnědý, zlatý)	1
R5	220 R / 0.25W / 5% (červený, červený, hnědý, zlatý)	1
R6	470 R / 0.25W / 5% (žlutý, fialový, hnědý, zlatý)	1
R7	470 R / 0.25W / 5% (žlutý, fialový, hnědý, zlatý)	1
R8	4K7 / 0.25W / 5% (žlutý, fialový, červený, zlatý)	1
L1	1 0uH (hnědý, černý, černý, stříbrný)	1
C1	22 0uF/16V (při pájení dodržujte polaritu!)	1

C2	10 nF (označení na pouzdře 103)	1
C3	100 nF (104)	1
C4	100 nF (104)	1
C5	220 uF/16V (dodržujte polaritu!)	1
C6	100 nF (104)	1
D1	SFH415-U IR-LED (dodržujte polaritu!)	1
D2	LED červená, 5MM (dodržujte polaritu!)	1
D3	LED červená, 5MM (dodržujte polaritu!)	1
Q1	Krystal / Quarz, 8Mhz / 3 PIN	1
SP1	Beeper (reproduktor), 5V (KC1206) (dodržujte polaritu!)	1
IC socket	28 PIN, konektor pro IC (dodržujte polaritu!)	1
JP1	3 PIN, PCB typ, černý	1
JP2	3 PIN, PCB typ, černý	1
JP3	3 PIN, PCB typ, bílý	1
JP4	3 PIN, PCB typ, bílý	1
JP5	3 PIN, PCB typ, bílý	1
CON1_PCB	konektor, male (samec), 20 pinů, pro připojení plochého kabelu / úhel 90 stupňů	1



Rozložení komponentů na desce spojů

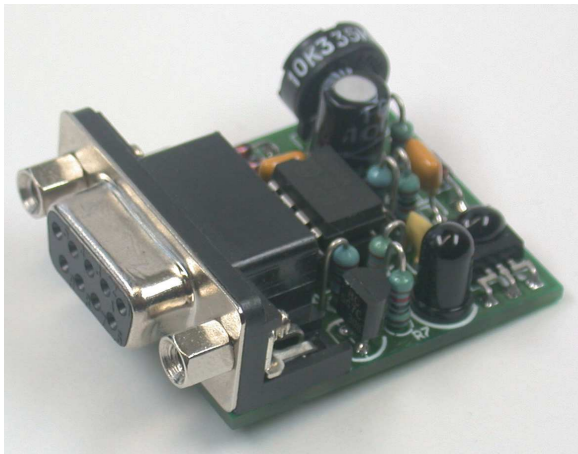
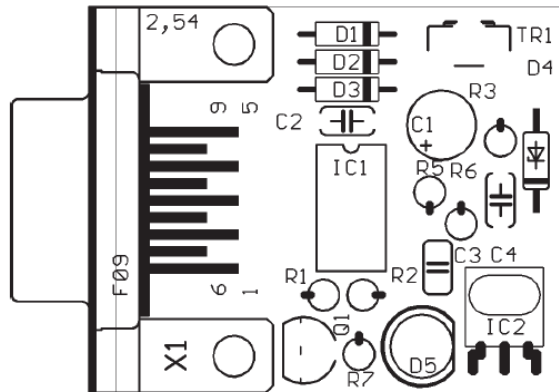


Sestavení IR tranciveru RS232

- IC1: instalujte 8-pólový konektor. Symbol polarity (nepatrně asymetrický) musí odpovídat s označením na PC.
- Diody D1, D2, D3: 1N4148 dodržujte správnou polaritu! Všimněte si proto označení na součástkách a nezaměňujte je s ZPD5.1 nebo BZX55-C5V1!
- D4: ZPD5.1 nebo BZX55-C5V1, dodržujte správnou polaritu při pájení těchto součástek! Všimněte si proto označení na této součástce a nezaměňte je s diodou 1N4148!
- D5: SFH-415-U IR LED (černá LED). Při pájení dodržujte správnou polaritu této součástky! Zajistěte její ohyb směre k desce spojů.
- C1: 100 μ F / 16 V – zachovejte správnou polaritu!
- C2, C4: 100 nF, keramický kondenzátor (označení na pouzdře 104).
- C3: 680 pF keramický kondenzátor (označení na pouzdře 681).

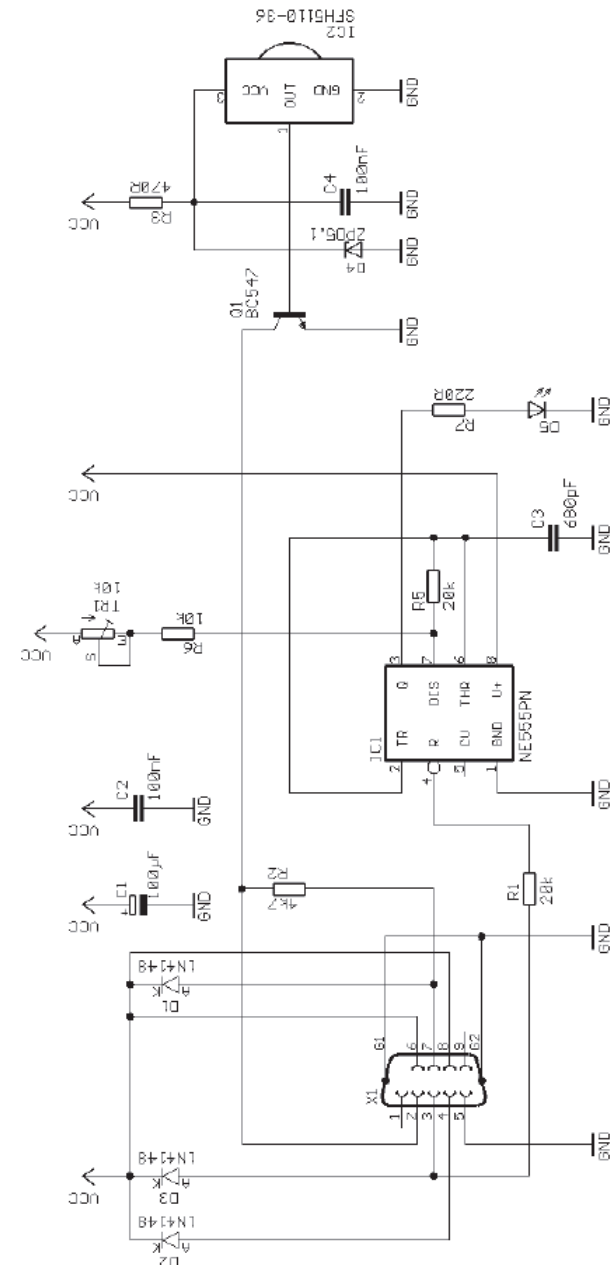
- Q1: BC 547 (A, B nebo C) nebo BC 548 (A, B nebo C).
- R1, R5: 20 k Ω (barvy proužků červený, černý, oranžový, zlatý)
- R2: 4,7 k Ω (žlutý, fialový, červený, zlatý)
- R3: 470 Ω (žlutý, fialový, černý, zlatý)
- R4: není použito
- R6: 10 k Ω (hnědý, černý, oranžový, zlatý)
- R7: 220 Ω (černý, červený, hnědý, zlatý)
- TR1: 10 K Ω tranzistor
- IC2: SFH5110-36 IR receiver IC, zajistěte potřebný ohyb jeho vývodů.
Při pájení dodržujte polaritu! Elektrostatický výboj (ESD) nebo extrémní ohřev při pájení může způsobit nevratné zničení této elektronické součástky.
- X1: 9-pólový, SUB-D konektor, kryt musí být umístěn do blízkosti desky spojů.
Zajistěte i připájení připojovacích proužků!
- IC1: při vkládání NE555P dbejte zachování správné polarity!

Osazení komponentů RS-232 IR Tranceiveru na desce spojů

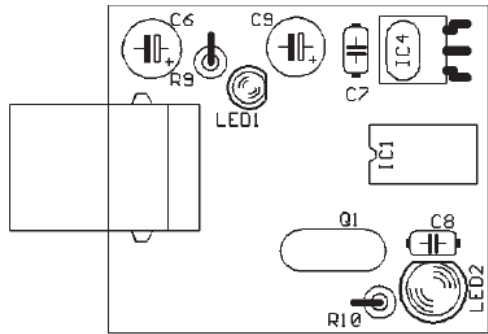
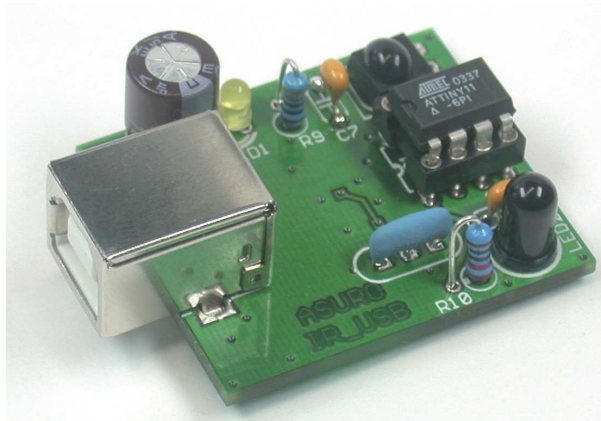


Na závěr pájení ověřte správnost umístění všech komponentů a zejména správné polohy (polarity).

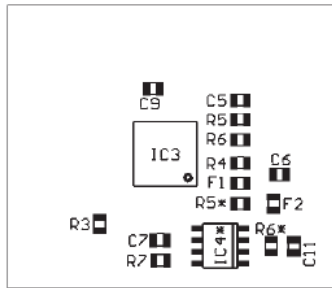
Schématický náčrt RS-232 IR Tranceiveru na desce spojů

























Použít stejně tak můžete již sestavený USB IR-Tranceiver.

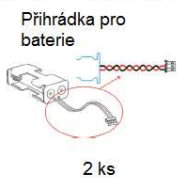
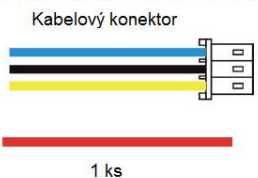
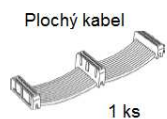
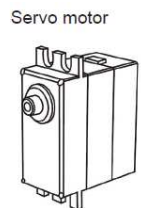
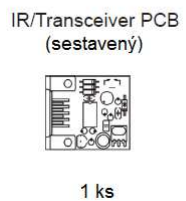
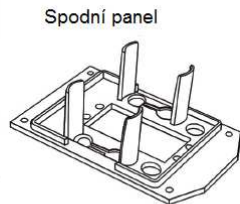
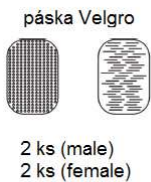
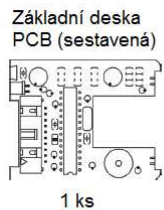
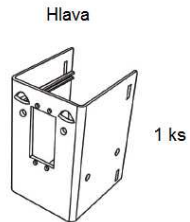
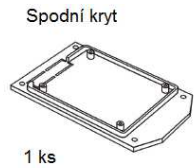
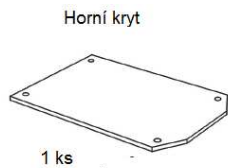


USB-Infrared-Tranceiver



Seznam dodávaného spojovacího materiálu

Maticе M3	Šroub M3 x 8mm	Objímka se závitem	Závit	Maticе M2
				
4 ks	8 ks	4ks	4 ks	10 ks
Maticе M2	Šroub M2	Objímka se závitem	Závit	Maticе M3
				
4 ks	4 ks	2 ks	2 ks	2 ks
Šroubky s oválnou hlavou	Nýt	Šroubovací konektor	Servo šroub	
				
4 ks	4 ks	4 ks	1 ks	
Dlouhá tyč se závity na koncích	Krátká tyč se závity na koncích			
				
2 ks	4 ks			
Chodidlo	Tyč 5 x 80 mm	Spoj v chodidle		
				
2 ks	2 ks	2 ks		
Část nohy (přední)	Část nohy (zadní)	Tyč pohonu servo ramene		
				
2 ks	2 ks	2 ks		

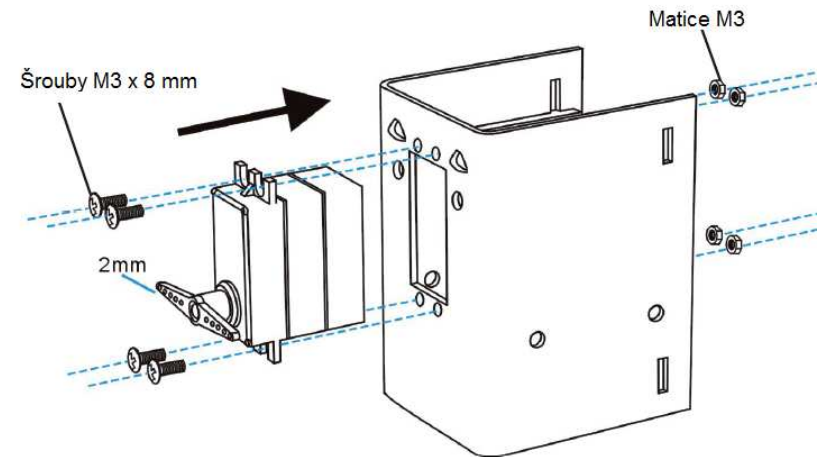
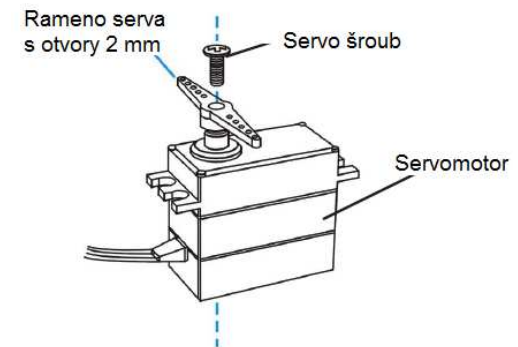


Postup pro sestavení mechanických částí

Montáž servomotoru do hlavy robota

Potřebné části a spojovací materiál:

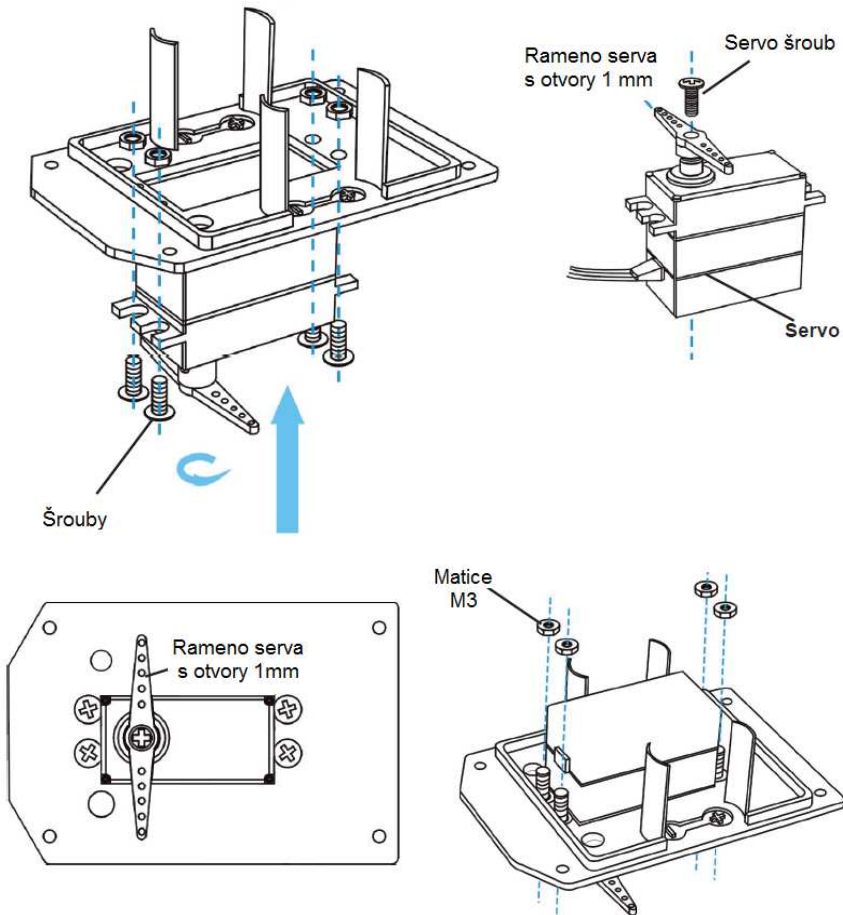
- Přední panel hlavy robota
- 1 servomotor
- 4 ks šroubů
- 4 matice M3
- 1 ks servo ramene (s otvory 2 mm)
- 1 ks servo šroubu



Sestavte servo a jeho rameno podle výše uvedeného obrázku.

Instalace spodního servomotoru

Potřebné části a spojovací materiál: Spodní kryt (panel)
1 servomotor
4 ks šroubů
4 matice M3
1 rameno (s otvory 1 mm)
1 ks servo šroubu

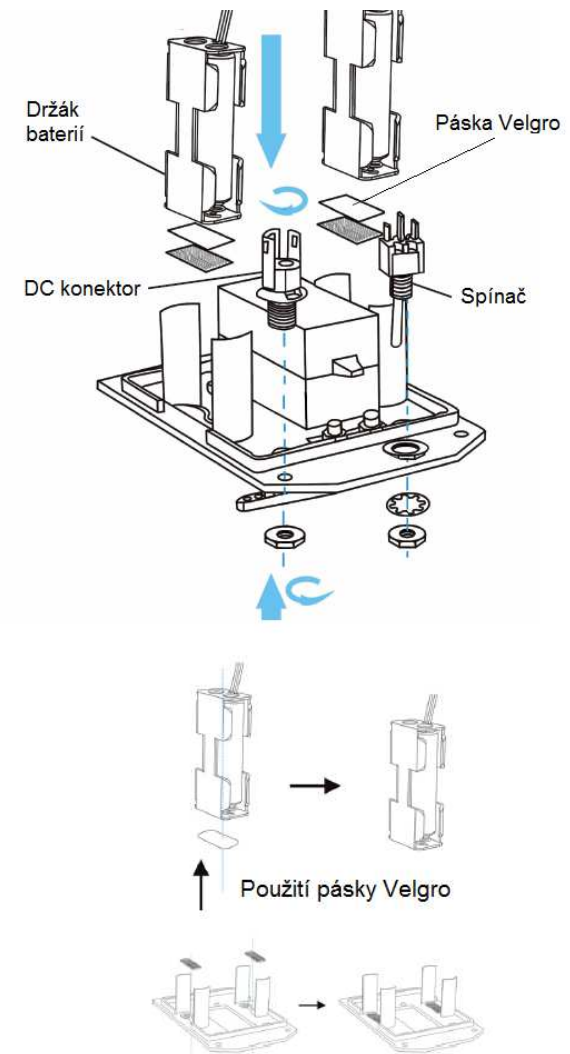
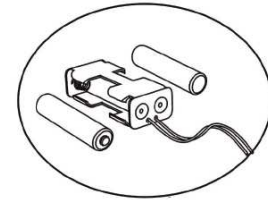


Sestavení spodního panelu

Potřebné části a spojovací materiál: Sestavený spodní panel
1 spínač
1 DC konektor
2 ks držáku baterií
4 ks baterií (akumulátory) typ AA
2 ks pásky Velgro (male)
2 ks pásky Velgro (female)

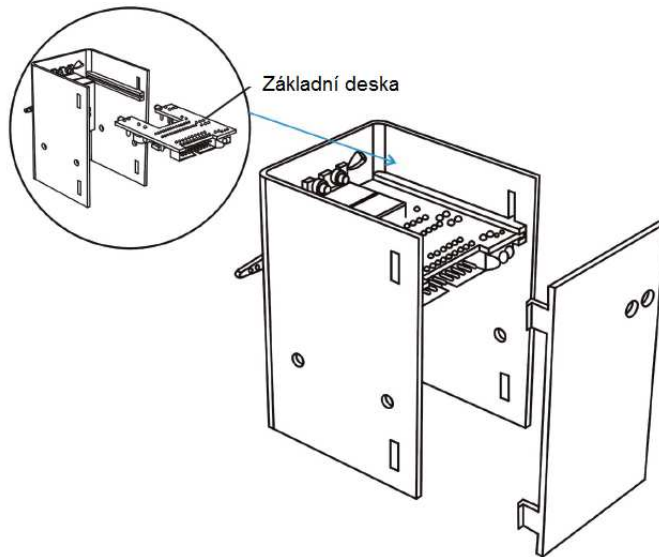
Držák baterií

Baterie do držáku vložte předtím, než uzavřete zadní panel robota!

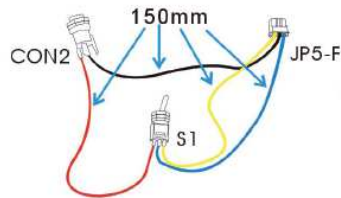


Hlava robota

Potřebné části a spojovací materiál: Sestavený spodní panel, sestavený panel hlavy, základní deska a zadní část (panel).

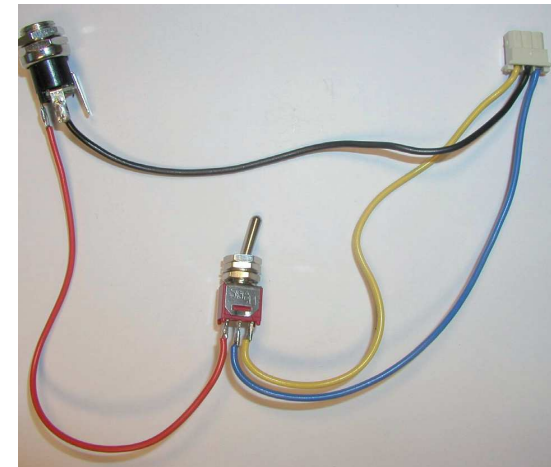
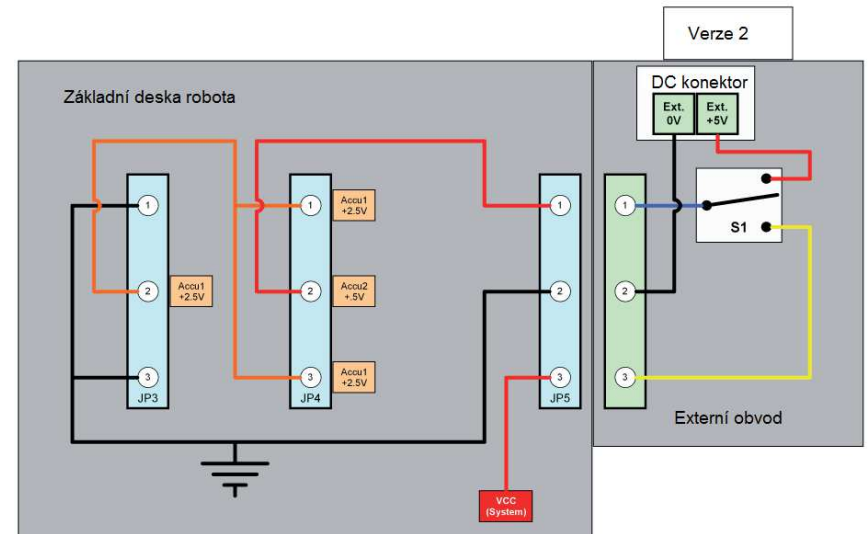


Důležité upozornění! Předtím, než hlavu robota sestavíte a uzavřete, proveďte všechna potřebná kabelová zapojení!

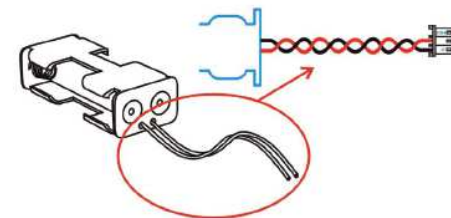


Kabelové zapojení

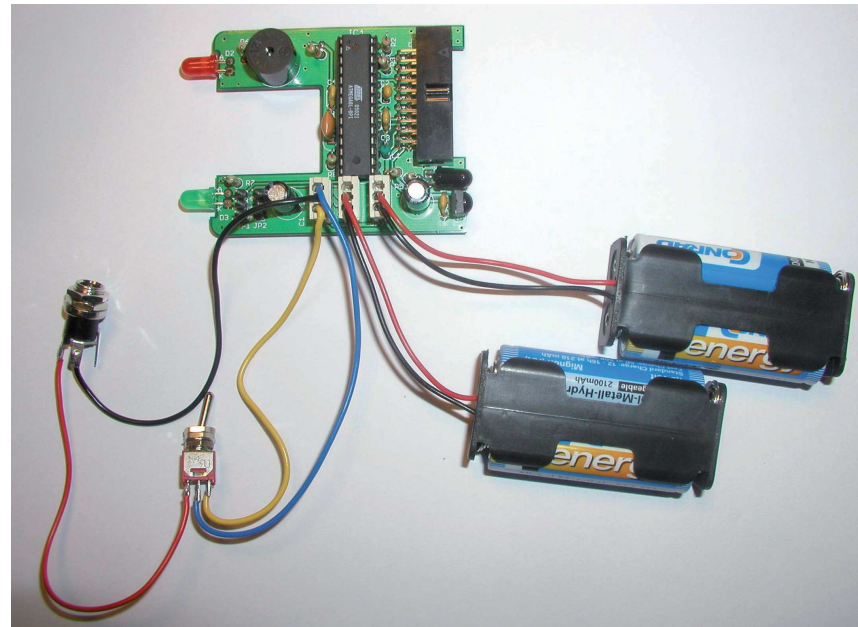
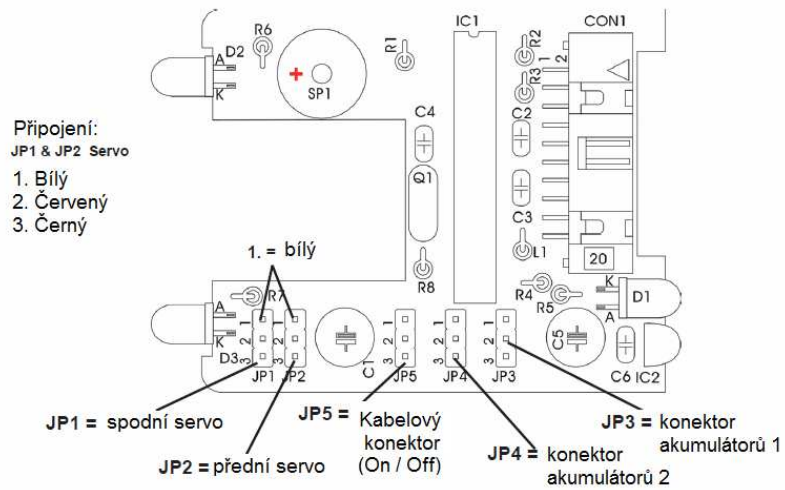
V tomto kroku bude zapotřebí komponentů sestavených v předchozích částech (spodní část, hlava robota, základní deska PCB). Navíc použijete kabelové zapojení, které tvoří černý, modrý, žlutý a červený vodič (na obrázku výše). Na následujícím schématu odpovídá barevné označení vodičů skutečnosti. Vodiče proto instalujte přesně podle tohoto schématu.



Sestavení držáku baterií

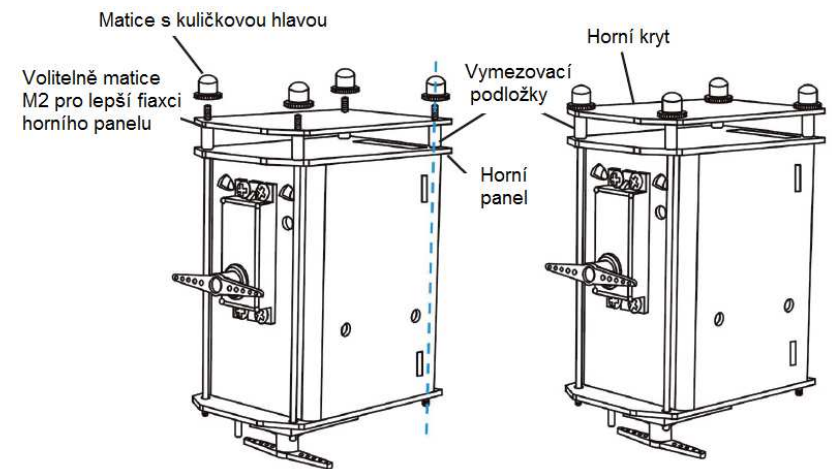
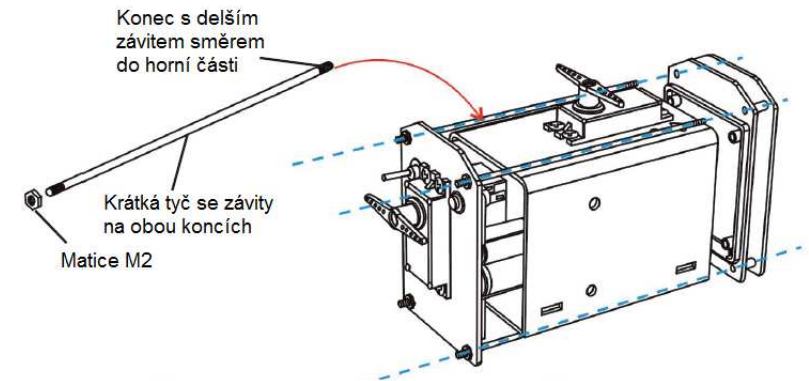


Kabelové připojení u základní desky robota



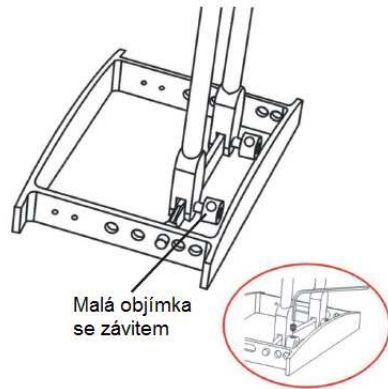
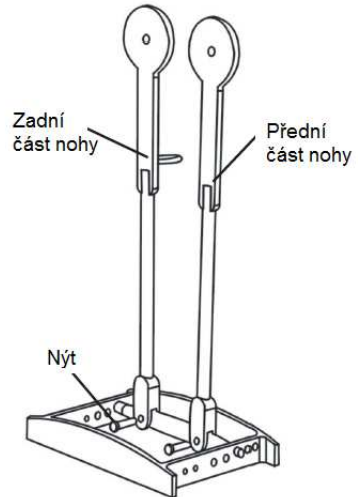
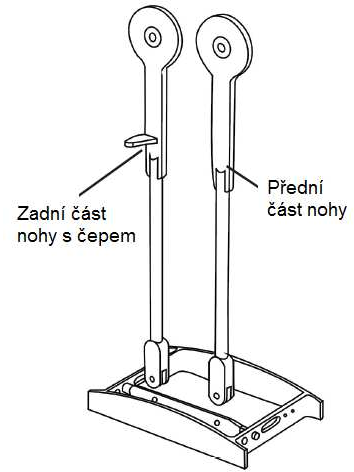
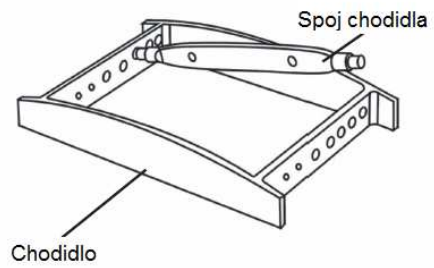
Připojení mechanických částí k hlavě robota

V této části bude zapotřebí sestavené hlavy robota, horní panel, horní kryt, 4 ks kratší tyče se závity na obou koncích, 4 ks matic M2, vymezovací podložky a 4 ks matic s kuličkovou hlavou.



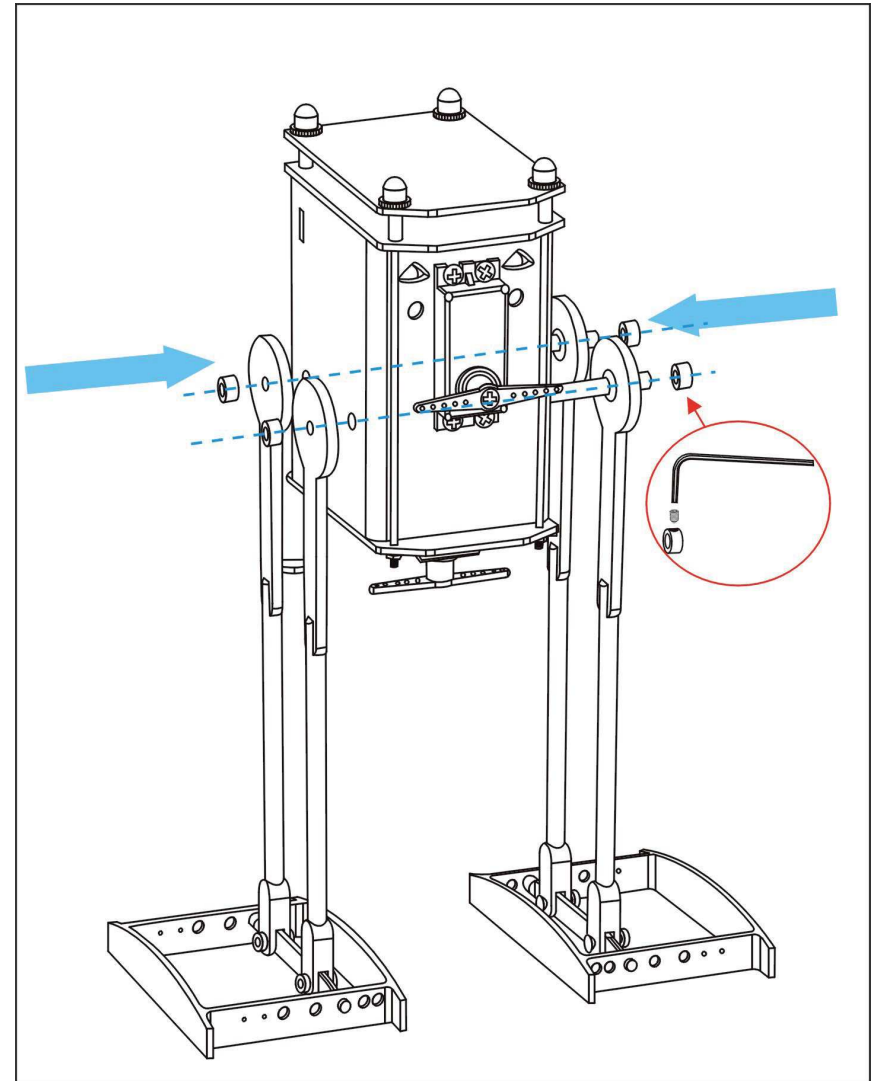
Sestavení nohou a chodidel robota

Potřebné komponenty: 2 chodidla, 2 ks přední části nohy, 2 ks zadní části nohy, 2x spoj chodidla, 2 nůty a 2 objímky se závitem.



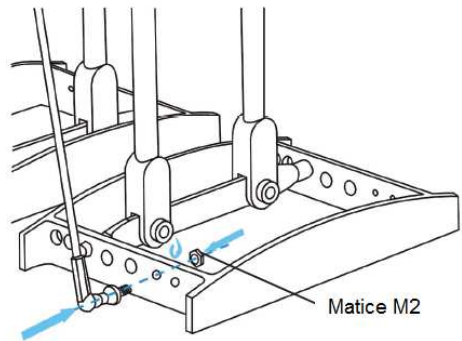
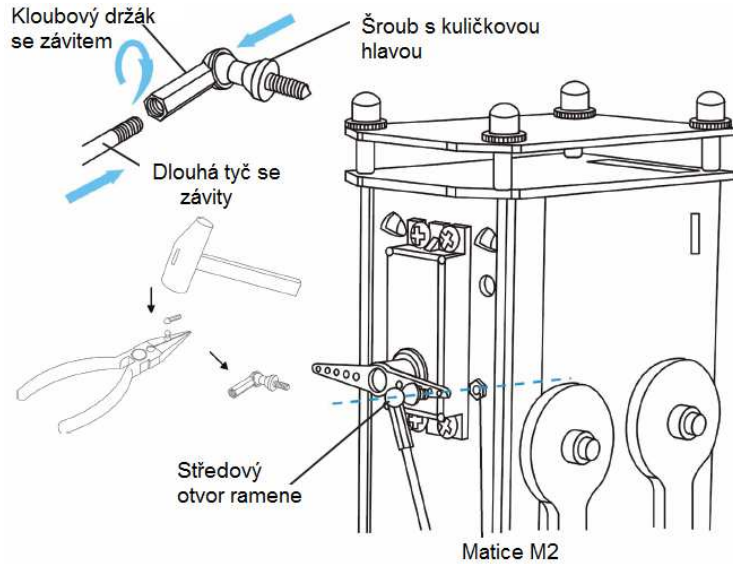
Přípevnění nohou k hlavní části robota

V tomto kroku bude zapotřebí následujících dílů: Hlavní část (hlava) robota, sestavené obě nohy, šroub 5 x 80 mm a 4 velké úchyty se závitem.



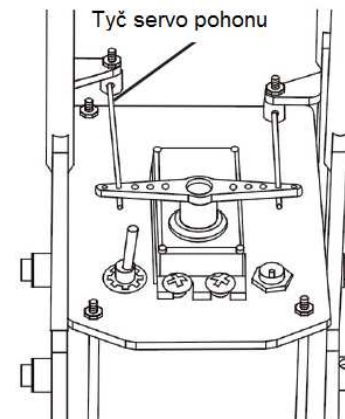
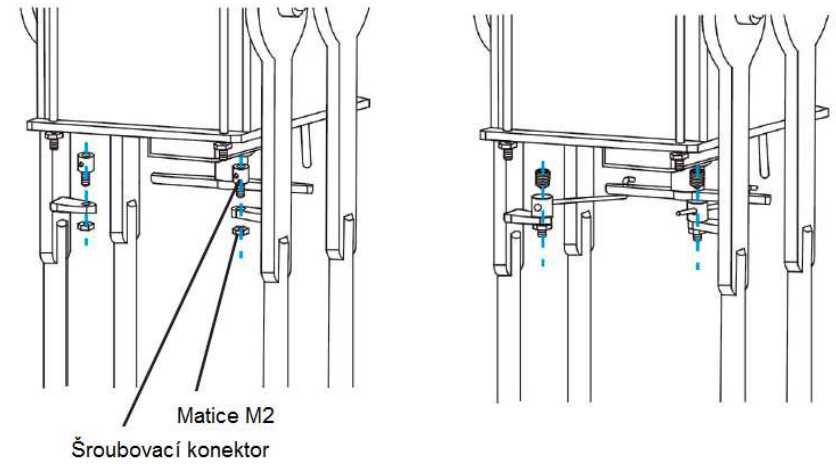
Připojení nohou k servo pohonu

V tomto kroku budete potřebovat sestavenou hlavní část robota včetně nohou, 4 ks šroubků k kuličkové hlavě, 4 ks kloubového držáku se závitem, 2 tyče (dlouhé) se závitem na obou koncích a 4 matice M2.

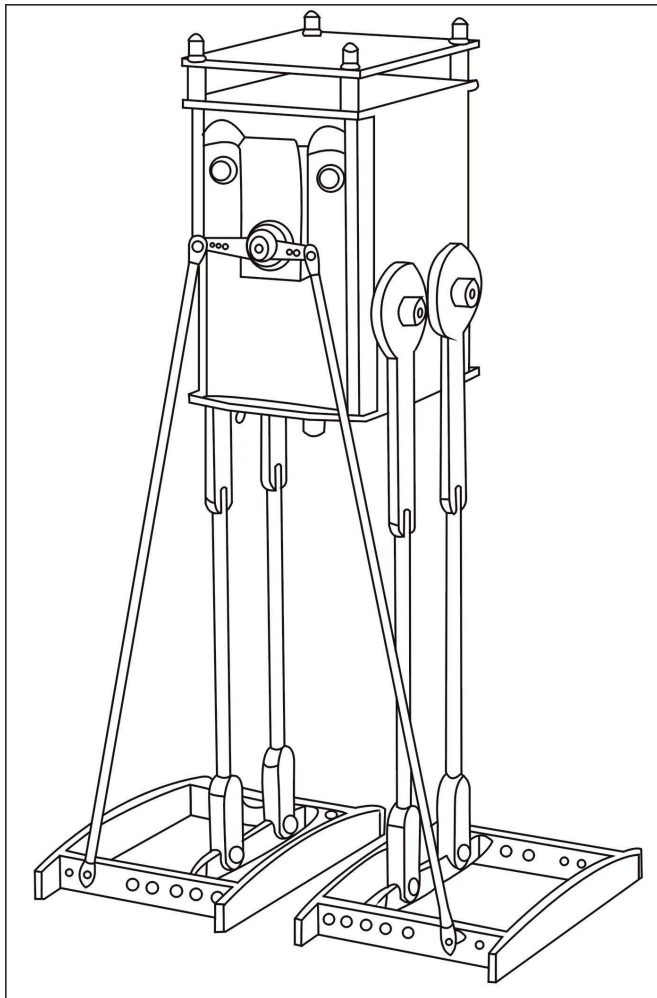


Finální montáž nohou

Předchozí sestavenou část, 2 matice M2, 2 tyče pro servo pohon, 2 šroubovací konektory.

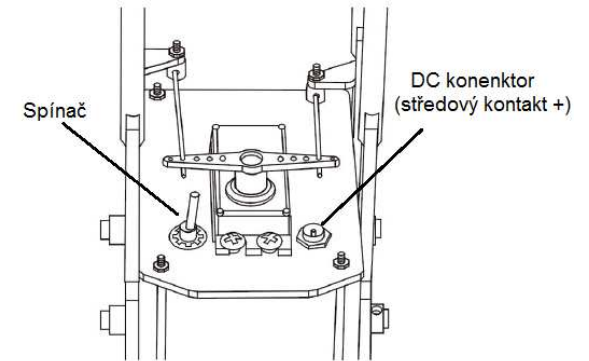


Tím je sestavení mechanických částí robota u konce!



Postup pro nabíjení akumulátorů

1. Jako první připojte kulatý konektor nabíječky do DC konektoru robota.
2. Hlavní spínač Power přepněte do polohy Off (vypnuto).



Software

Návrh a tvorba vlastního programu

V této části získáte základní a stejně tak i podrobné informace ohledně programování. Dozvíte se také více o definicích a slovech používaných v programovacím jazyku. Vlastní program (například „test.c“) můžete vytvořit v jazyce zvaném „C“. Tento program pak bude zapotřebí převést do hex souboru (například „test.hex“), který pak budete moci načíst do paměti robota.

1 krok – tvorba „C“ programu

Pro tvorbu programu byl zvolen programovací jazyk „C“. Program pro robota pak můžete psát ve speciálním textovém procesoru (editoru). Námi doporučovaný editor je pro tyto účely „Programmers Notepad 2“ (PN2). Program však můžete napsat i v běžném poznámkovém bloku „Notepad“ nebo standardním textovém editoru jako je MS Word. Přesto však nedoporučujeme používat běžný textový procesor, vzhledem k tomu, že v tomto prostředí je proces editace vždy o něco složitější. Program PN2 disponuje funkcemi pro editaci a odstraňování různých chyb s podporou několika různých programovacích jazyků. Jednoduše si vyberte programovací jazyk, například „C“, „Visual Basic“ nebo „HTML“ v konfiguraci programu PN2. Tato aplikace výrazně usnadňuje programování označením všech speciálních příkazů, komentářů, funkcí a proměnných v několika barvách. Všechny tyto funkce vám výrazně usnadní první kroky při programování.

Krok 2 – sestavení programu

Vlastní program se nazývá „source“ (zdroj) nebo „source program“ (zdrojový program). Jde o jednoduchý textový dokument s názvem například „test.c“. Spuštění tzv. „compileru“, který je GCC.exe v našem systému, zajistí transformaci zdrojového kódu „test.c“ do object file (objektový soubor) s názvem „test.o“. Tento soubor je čitelný textový soubor, který obsahuje sadu vývojářských instrukcí. Robot bude těmito instrukcím rozumět, ale předtím musejí být tyto instrukce převedeny do jiného kódu. „O“ (objektový soubor) obsahuje váš „C“-zdroj a zpravidla i seznam instrukcí k vykonání. V posledním kroku provede tzv. „linker“ transformaci těchto instrukcí do procesorem vykonatelných instrukcí (v hexadecimálním kódu), který bude uložen v hex-souboru, například „test.hex“. V základu je každá programová modifikace provázena compilerem a linkerem, nicméně pro celkové zjednodušení se používá automatická dávka pro zpracování a používá proto tzv. „makefile“ proces.

Dávkový proces umožňuje spojení několika různých funkcí a procesů. Makefile navíc zajišťuje jak a který soubor bude sestaven „compiled“ a propojený „linked“. Tyto procesy se dají aktivovat jednoduše po výběru příslušného menu v aplikaci PN2 nebo použitím speciální klávesové zkratky. Do paměti robota bude následně načteno velké množství různých procesů a akcí v hex-souboru.

PN2 editor je univerzální programovací systém, vytvořený pro velké množství různých compiler a linker systémů, jež tvoří komplex rozhraní mezi PN2 programem a compiler / linker programy. Pro základní programování „C“ však není zapotřebí znát komplex všech metod. Rozhraní a jeho konfigurace bude v další části návodu vysvětlena krok za krokem.

Nabíjení akumulátorů robota

Napájení robota 4,8 V zajišťují celkem 4 akumulátory (Ni-CD). Akumulátory nejsou jistě žádnou pojistkou nebo sériovým rezistorem! Pro nabíjení akumulátorů používejte výhradně kvalitní nabíječku, která podporuje nejmodernější nabíjecí technologie. Doporučujeme použití nabíječky vybavené mikroprocesorem, případně můžete použít síťový adaptér se stabilizovaným napětím a velmi malým nabíjecím proudem, například 5 V / 300 mA.

GCC compiler

Hlavní výhodou „gnu“ je veřejná licence a vysoká úroveň kvality GCC compiler systému, který je vydáván a spravován „gnu“ organizací. Jedná se o bezplatný a open source software. Více informací můžete získat na <http://www.gnu.org/>. Mezi takové programy patří i WINAVR, který je speciálně vyvinutý pro ATmega procesory, který používá i náš robot. Balíček WINAVR obsahuje compiler program AVR-GCC, linker, PN2 a obrovské množství mikroprocesorových souborů. WINAVR software je dodáván jako single installation file, který naleznete také na dodávaném instalačním CD. Program si můžete aktualizovat na webu: <http://www.sourceforge.net/>.

Krok 3 – Přenos programu do robota

DLR (Německá centrála pro letecký a vesmírné technologie) vydalo nástroj zvaný Flash, který zajišťuje přenos programu do systému robota prostřednictvím příslušného COM portu. Flash program dokáže zajistit velké množství akcí, které jsou zapotřebí pro stažení vybraného YETI HEX programu, například „test.hex“ z počítače do mikroprocesoru robota. Robot je vybaven vestavěným komunikačním nástrojem tzv. „bootloader“, který nemůže být odstraněn nebo do něj nelze získat přístup běžnými procesy prováděnými uživatelem. Po zapnutí robota se jako první spustí bootloader. Tento komunikační nástroj nejprve vyčkává po dobu 3 sekund, aby poté vyhledával IR signál z Flash programu počítače. Po přijetí signálu systém robota zajistí zapisování programu do interní paměti. Pakliže robot nezaznamená žádný signál, spustí zcela automaticky naposledy načtený program ve ze paměti. Robot je vybaven speciálním typem paměti pro uchovávání dat i po jeho vypnutí. Tento druh paměti se nazývá „Flash“ paměť. Flash je obecně používaným názvem pro načítací nástroj u tohoto typu paměti.

Instalace software

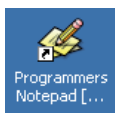
Vložte instalační disk do CD/DVD mechaniky počítače. Instalátor automaticky spustí instalační proces. Jako první zvolte jazykové nastavení pro prostředí aplikace. Nezbytné aplikace pro chod robota naleznete v hlavní nabídce instalátoru. Instalaci software proveďte na pevný disk počítače. K tomu však bude zapotřebí administrátorského oprávnění. Během instalace nástroje Flash, který zajišťuje přenos programu do robota, proces dále zajistí instalaci programového editoru (Programmers Notepad 2, PN2) a Compiler (WinAVR). V rámci instalace získáte i několik testovacích programů. V editoru (PN2) dojde k vytvoření menu pro správu souborů „Make and Clean files“.

Windows Flash-Tool

Flash program přeneste na pevný disk vašeho počítače (například do C:\programs\flash). Flash nástroj však můžete spustit i přímo z CD. V obou případech však doporučujeme vytvořit programového zástupce na pracovní plochu počítače. Pro instalaci compileru je zapotřebí administrátorských oprávnění (během instalace dochází k úpravě systémového registru). Instalaci spusťte výběrem „Install“.

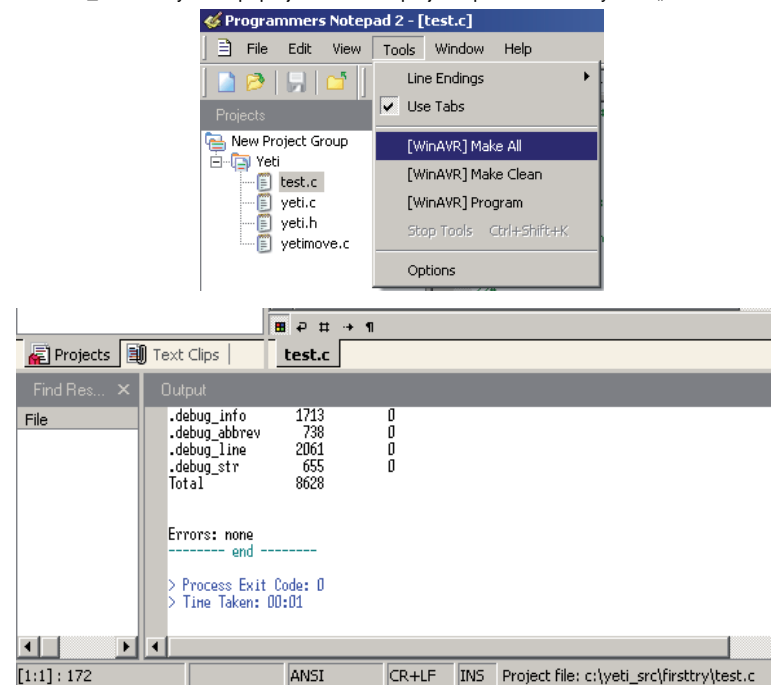
COMPILER WinAVR (20060421)

Windows je obchodní značkou společnosti Microsoft Corporation. V nabídce „Please select a language“ vyberte požadovaný jazyk a potvrďte OK. Pro pokračování v instalaci zvolte „Next“. Pro instalaci programu musíte potvrdit a odsouhlasit licenční ujednání výběrem „I agree“. V dalším kroku vyberte cílové umístění pro instalaci programu, například C:\WinAVR a potvrďte výběrem „Next“. Zaškrtněte výběr všech instalačních komponentů: Install Files, Add Directories to PATH (Recommended), Add Shortcuts to Desktop a Install Programmers Notepad a klikněte na „Install“. Na obrazovce následně uvidíte spuštění instalačního procesu souborů. To může trvat i několik minut. Po úspěšném dokončení instalace programu vyberte „Finish“. Instalátor poté automaticky spustí prohlížeč s uživatelským návodem k této aplikaci. Zároveň přitom dojde k vytvoření programového zástupce na ploše počítače.



Vytvoření „C“ souboru

Na začátek spusťte soubor 'C:\YETI_src\FirstTry\test.c'. Otevřete aplikaci Programmers Notepad 2 a přejděte do nabídky pro správu souborů „File“ a poté vyberte „Open“. V adresáři FirstTry vyhledejte data: C:\YETI_src\FirstTry\YETI.pnproj. V seznamu projektů pak následně vyberte „test.c“.



Zobrazí se: ...Errors: „none“ and „Process Exit Code: 0“. V této fázi jsou kompilovaná data připravena k aplikaci a zároveň tím došlo k vygenerování souboru „test.hex“. V případě, že program nebude obsahovat žádné chyby, ve spodní části se zobrazí informace Errors: none.

Tímto krokem došlo k vygenerování nového souboru test.hex ze souboru test.c. (a YETI.c). Tento soubor obsahuje konvertovaný program pro robota, který se přeneso do jeho paměti. Tento program však zatím nemá žádnou funkci. Později dojde k načtení jeho obsahu do interní paměti robota prostřednictvím nástroje Flash.

Jak vše funguje?

Soubory „input files“ v hlavní nabídce zajistí volání (call) dávkového souboru test-all.bat (tento dávkový soubor obsahuje seznam příkazových řádků, které jsou spuštěny jeden po druhém). V souboru Test-all.bat bude spuštěn příkaz „make all“, který vytvoří soubor „make file“ v oblasti paměti (při programování robota) ve stejném souboru jako Test-all.bat. Make file je textový soubor, který definuje tvorbu jednoho nebo více programů. Během programování, v případě jediného souboru, máte neustálý přehled o obsahu programu. V pozdější fázi, kdy dojde k zápisu komplexního systému a programovací data obsahují více souborů, které musejí být všechny konvertovány ve správném pořadí krok po kroku a jsou propojené (linked) s ostatními, budou i soubory make file velmi komplexní.

Všechna „all“ volání (call) vstupních souborů v make file znamená, že dojde ke konverzi komplexního projektu a ne pouze jednotlivých vstupů. Make files v tomto testovacím programu jsou zapsány tak, že soubor s názvem test.c bude kompilován s YETI.c (který zahrnuje některé předdefinované funkce) a vytvoří nový hex soubor. Tento soubor může být načten do paměti robota.

Upozornění! Dokud neprovedete úpravu souboru make file a pouze jej kopírujete, můžete zadat název svému vlastnímu programu test.c.

Více o tématu „make files“ naleznete například na internetovém portále <http://www.gnu.org>.

Poté, co budete kompilovat program, dojde k vygenerování extra dat. Tato data jsou však zapotřebí pouze během procesu konverze, po ukončení tohoto procesu již nejsou zapotřebí. Tato data poté můžete odstranit pomocí nástroje „Clean“. V souborové nabídce „File“ a „Open“, kde můžete spatřit vygenerovaná data. Další informace k aplikaci WinAVR naleznete například po přechodu na následující odkazy: <http://www.avrfreaks.net>, <http://winavr.sourceforge.net> a <http://www.kreatives-chaos.com>.

LINUX

Pro instalaci software nebude zapotřebí administrátorských oprávnění (root rights).

Flash-Tool

Spusťte program z nabídky software CD a přepokopujte nástroje „yetiflash“ a „yeticon“ z adresáře „linux/tools“ do adresáře „usr/local/bin“. Na závěr musíte zajistit spuštění programu pomocí „chmod a+usr/local/bin yeticon yetiflash“.

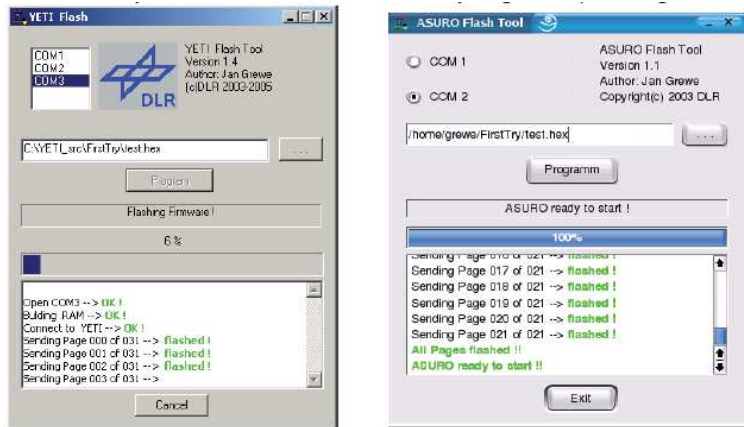
Compiler

Pro instalaci Gnu-compileru u AVR procesorů, vložte instalační CD YETI a v adresáři „Linux/Compiler“ vyberte následující:

1. **avr-binutils-... .rpm**
2. **avr-gcc-... .rpm**
3. **avr-libc-... .rpm**

Celá instalace je skutečně velmi jednoduchá. Zadejte proto pouze příkaz: rpm -i <paket>.rpm do kořenového adresáře (root directory). Pro programovou editaci můžete použít aplikace Exmacs, Kate nebo Kedit. Demo programy naleznete v adresáři „YETI_src/FirstTry“.

Otevřete rozhraní, změňte adresář a zadejte „make“.



Flash – programovací nástroj pro robota Yeti

Pro tento proces bude zapotřebí Flash programu. Spusťte program a zvolte rozhraní do kterého jste připojili IR Tranceiver. Vyberte hex. soubory v adresáři C:\Own files\YETI_src\FirstTry. Umístěte kompletně sestaveného a otestovaného robota do blízkosti IR Tranceiveru (do max. vzdálenosti 50 cm). IR komponenty obou modulů musí mít vzájemnou viditelnost. Ve Flash nástroji přepněte S1 do polohy ON. K tomu musí dojít dříve, než status indikátor dosáhne pravé koncové polohy svého rozsahu. Pokud se vám nepodaří tuto akci provést včas nebo došlo k selhání vzájemné komunikace, vypněte robota, spusťte program a znovu přesuňte spínač S1 do polohy ON.

Po úspěšném navázání komunikace se zobrazí informace o přenosu souboru test.hex do robota. Programové soubory tím budou uloženy do flash paměti robota uvnitř procesoru, kde program zůstává dostupný i po jeho odpojení od zdroje napájení. Po načení programu musíte robota vypnout (OFF) a znovu zapnout (ON). Tím dojde k automatickému spuštění načteného programu. Zároveň přitom se rozsvítí zelená LED kontrolka.

Po spuštění Flash programu se počítač pokusí o navázání komunikace s robotem. Po zapnutí robota se načte jeho systém. Tento proces je indikován krátkým probliknutím LED kontrolky po dobu 1 sekundy. Systém robota poté spustí vyhledávání nového software. Pokud dojde k nalezení nového programu, systém jej automaticky načte. Program se pak spustí jednoduše po restartu robota (zapnutí a vypnutí).

Chybové stavy Flash procesu

„c“ = CHECKSUM ERROR. Robot přijal jiný signál. Tento signál může být generován jinými optickými zdroji (například fluorescenční zářivky) nebo došlo k přerušení signálu pohybem různých předmětů. „t“ = TIMEOUT. Přerušení signálu v důsledku vypršení časového limitu pro zahájení přenosu. „v“ = VERIFY ERROR. Došlo k zápisu nesprávných dat do flash paměti. V tomto případě se jedná o stav, při kterém je indikováno, že došlo k dosažení konce provozních cyklů non-volatile programovací paměti. Tato paměť má provozní omezení na přibližně 10000 programovacích cyklů.

V případě, že dochází k častému zobrazení chybového hlášení „c“ se ujistěte, že v okolí nesvítí žádný zdroj světla, zejména v okolí přenosu IR signálu. Vždy stiskněte tlačítko „Program“ před samotným zapnutím robota. V opačném případě nemůže dojít ke spuštění procesu přenosu dat.

Testování a provoz robota

Poté, co kompletně sestavíte robota a načtete program do jeho systému, robot automaticky spustí tento program a bude se pohybovat. Hlavním úkolem je však schopnost nalézt a odstranit konstrukční a stejně tak i programové chyby, aniž by přitom došlo k jakémukoliv poškození samotného robota.

Infračervený tranceiver RS232

Připojte tranceiver do volného sériového rozhraní vašeho počítače pomocí dodávaného 9-pólového sériového kabelu a spusťte program „Hyperterminal“ (nebo „Municom“ v systému Linux).

Tento program naleznete na instalačním disku. Po spuštění této aplikace vás systém vyzve pro zadání názvu připojení (můžete použít například Yeti). V dalším kroku „connect by“ vyberte rozhraní COM do kterého je tranceiver připojený. Zadejte následující parametry:

- Bits pro Second: 2400
- Databits: 8
- Parity: none
- Stopbits: 1
- Flowcontrol: none

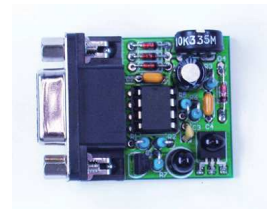
Ve vzdálenosti 10 cm od tranceiveru přidrže list běžného kancelářského papíru.

Stiskněte některá tlačítka v programovém terminálu. Na obrazovce se budou zobrazovat vybrané znaky. IR tranceiver vyšle tyto znaky prostřednictvím IR diody (D5). Díky listu papíru dojde k odrazu signálu a jeho zpětnému přijetí IC receiverem (IC2), odkud se znovu přenesou do počítače. Pakliže se zobrazí zadané znaky, můžete jen velmi nepatrně otáčet trimrem mezi jeho levou a pravou koncovou polohou. Použijte proto miniaturní šroubovák a upravte polohu trimru, dokud se nebudou zobrazovat správné a čitelné znaky.

USB IR Tranceiver

Tato část je věnována použití USB IR tranceiveru. USB tranceiver nemá žádné pouzdro a je proto vystaven většímu riziku poškození v důsledku elektrostatického výboje. Předtím, než se jej budete dotýkat, uzemněte se dotykem kovového krytu počítače nebo jiného uzemněného předmětu.

Tento modul můžete opatřit plastovým a transparentním krytem. Zajistíte tím jeho lepší ochranu.



Windows

Jako první ověřte správnou funkci IR transceiveru.

Připojte proto transceiver do USB portu počítače prostřednictvím příslušného kabelu. Systém počítače následně zobrazí informaci o připojení nového hardware: Arexx Asuro/Yeti USB-IR-Transceiver. Z dodávaného CD nainstalujte potřebné ovladače (Windows/USB Driver). Pro tento proces bude zapotřebí použití administrátorského oprávnění k provedení změn v počítači. Po úspěšné instalaci ovladačů, bude transceiver fungovat jako běžné USB zařízení.



Po spuštění programu Hyperterminal vás systém vyzve ke konfiguraci názvu připojení.

Použít můžete například YETIUSB nebo jiný název. V dalším kroku „connect by“ zvolte rozhraní COM, ke kterému byl transceiver připojen v předchozí fázi. Zadejte následující parametry transceiveru:

- Bits pro Second: 2400
- Databits: 8
- Parity: none
- Stopbits: 1
- Flowcontrol: none

Obdobným způsobem otestujte funkci přenosu dat. Použijte proto znovu list papíru a zadejte několik znaků. Pomocí šroubováku upravte polohu trimru, dokud na monitoru počítače nevidíte čitelné znaky.

Linux

Pro použití transceiveru v prostředí Linux postupujte podle následujících pokynů.

Připojte transceiver do volného USB portu vašeho počítače. Použijte proto příslušný kabel.

Po jeho připojení systém vydá krátký tón, kterým signalizuje, že došlo k detekci nového hardware.

Systém navíc zobrazí informaci

```
foo@bar:~>cat /proc/tty/driver/usb-serial
```

```
usbserinfo:1.0 driver:v1.4
```

```
0: module:ftdi_sio name:"FTDI 8U232AM Compatible" vendor:0403 product:6001
```

```
num_ports:1 port:1 path:usb-00:11.2-1
```

(„0“ v tomto případě může být „1“ nebo „2“)

Pro konfiguraci použijte následující parametry:

- Bits pro Second: 2400
- Databits: 8
- Parity: none
- Stopbits: 1
- Flowcontrol: none

Před použitím hardware bude zřejmě nezbytné seznámit se a na závěr potvrdit licenční ujednání.

Použit proto můžete `chmod u+rw /dev/ttyUSB0` (nebo 1, 2...) nebo `chmod g+rw /dev/ttyUSB0`.

Stejně jako v předchozím případě otestujte správnost funkce přenosu dat. Použijte proto znovu list papíru a zadejte několik znaků. Pomocí malého šroubováku upravte polohu trimru, dokud na monitoru počítače nevidíte dobře čitelné znaky.

Kalibrace

Abych robot byl stabilní, bude nezbytné spustit kalibrační proces. Tento proces je však nutné provést pouze jednou. Pro kalibraci musíte použít vhodný software, kterým dojde k seřízení polohy servomotorů do středové (výchozí) polohy. Kalibraci je nutné spustit po montáži nohou a chodidel.

Kalibrační software

Pro kalibraci použijte program Hyperterminal (například verze 6.3). Spusťte aplikaci Hyperterminal.

Nastavte parametry COM (X), baud-rate 2400, 8-bit, bez parity (no parity) a hardwarového řízení.

Připojte RS232 nebo USB adaptér do vašeho počítače. Zajistěte vhodnou polohu robota

pro infračervený přenos dat mezi USB adaptérem a zády robota. Zapněte robota.

Po zapnutí uslyšíte sérii akustické signalizace. Během 1 sekundy stiskněte na PC terminálu libovolnou klávesu. V tabulce 1 je seznam dostupných tlačítek pro kalibraci robota.

Kalibrační proces spusťte tlačítkem [-]. Oba servomotory se přesunou do své výchozí polohy (0).

Uvedte rameno serva do kolmé polohy. Použijte přesnou polohu, aniž by přitom bylo nutné jakkoliv manuálně upravovat polohu hřídele. Tuto polohu upravte pomocí tlačítek [7] a [9].

Stiskněte tlačítko [1] pro kalibraci této polohy do systému. Při testovacím procesu upravte polohu serva použitím tlačítek [7] a [9]. Stiskem tlačítka [8] pak dojde k přesunutí serva do výchozí polohy [0].

Více k tomuto tématu naleznete v příloze „J“. Zopakujte tento kalibrační proces i pro spodní servo.

Použijte však proto tlačítka [4], [5] a [6]. Tlačítko [+] se v tomto případě použije pro současnou kombinaci tlačítek [8] a [5]. Stiskem tlačítka [ENTER] režim kalibrace ukončíte.

Po restartu pak vždy zaujme robot výchozí polohu.

Příloha J – Tlačítka pro kalibrační proces

[7] Přední servomotor doleva	[8] Přední servomotor - kalibrovaná poloha	[9] Přední servomotor doprava	[-] [+] Návrat do výchozí (0) polohy (obdoba [8] a [5])
[4] Spodní servomotor doleva [1]	[5] Spodní servomotor – kalibrovaná poloha [2]	[6] Spodní servomotor doprava	[ENTER] Ukončení kalibračního procesu

Hardwarová kalibrace

Po úspěšné kalibraci servomotorů provedte mechanickou kalibraci nohou robota.

Uvolněte šrouby, které fixují „svaly“ robota. Zajistěte, aby se oba svaly mohly volně pohybovat.

Položte robota na záda a umístěte pod jeho záda obal CD. Spodní část chodidel přitom bude

v úrovni nad deskou stolu (výška obalu CD). Pro zachování této polohy použijte lepicí pásku, kterou

zařijte nohy robota. Spusťte aplikaci Hyperterminal. Otočte robota čelem k desce pracovního stolu

a zajistěte viditelný kontakt IR komunikace mezi USB adaptérem a zády robota.

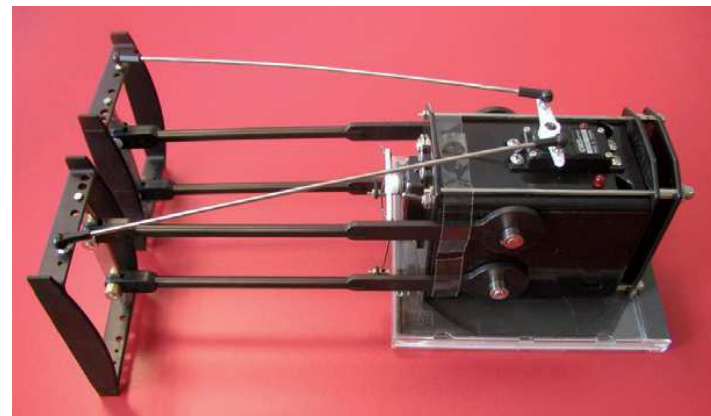
Zapněte robota a vyčkejte, dokud neuslyšíte sérii krátkých tónů. Robot se následně přesune

do výchozí polohy. Během dalších 3. sekund stiskněte libovolné tlačítko na PC terminálu.

Robot vydá krátký tón a přejde do kalibračního režimu, ve kterém budou oba servomotory aktivovány.

Použijte imbusový klíč pro dotažení šroubů ve svalech robota.

Vypněte robota a poté odstraňte lepicí pásku. Tím došlo k dostatečné kalibraci robota a jeho chůze.



Programování robota

Mozek robota

Na desce s tištěnými spoji má robot miniaturní počítač, který se nazývá mikrokontrolér. Tento mikrokontrolér je integrovaný obvod (IC). Jedná se o komponent s 28 piny, které se připojují do celého systému robota jako jsou LED kontrolky v očích, do reproduktoru, do IR komunikačního systému a servomotorů, kterými se ovládají všechny pohyby robota.

Servomotory

Tyto speciální motory ukryté ve speciálním pouzdře mají svou vlastní převodovku a různé elektronické komponenty. Požadovaný úhel mechanických částí se pak provádí pomocí jednotlivých koleček převodovky. Ve skutečnosti převodovka zpomaluje počet otáček v závislosti na dosažení požadovaného úhlu. Tento motor dokáže zajistit otáčení mechanických částí až do úhlu 220°. Nižší rychlost přitom vyvíjí značkou sílu nebo točivý moment na výstupní pozici a požadovaný úhel určité mechanické části.

Mikrokontrolér

Tento mikrokontrolér dokáže přijímat sadu 120 základních instrukcí a velké množství kombinací těchto instrukcí. Série instrukcí se nazývá program. Pro spuštění programu je nezbytné, aby počítač načel program do své interní paměti. Mikrokontrolér poté čte zadané instrukce z paměti a vykonává příslušné operace. Po splnění jedné instrukce přejde systém k plnění další. Tento proces se pak opakuje v trvalé smyčce až do ukončení programu.

Paměť Flash

V každém počítači musíte nejprve „odstartovat“ program, například hru. Po spuštění programu pak dochází ke kopírování programu z pevného disku do operační paměti. Po vypnutí počítače dojde k odstranění programu z paměti. Mikrokontrolér je vybaven zvláštním druhem paměti zvaným „Flash“, ve které je program uložen trvale. K odstranění programu z flash paměti proto nedochází ani po vypnutí počítače. Mikrokontroléry používají flash paměti coby permanentní operační paměť pro ukládání programů do doby, než samotný program neodstraníte.

Načtení programu

Pro načtení programu do flash paměti mikrokontroléru je zpravidla zapotřebí speciálního programovacího zařízení, které je součástí integrovaného obvodu. Programovací zařízení umožňuje přenos programu přímo do flash paměti. Toto programovací zařízení se nazývá „programátor“. Pro zjednodušení tohoto procesu byl vyvinutý software „Bootloader“, který zajišťuje vyhledávání programu ve flash paměti ihned po připojení zdroje napájení. Pakliže se ve flash paměti nachází program, bootloader zajistí jeho okamžité spuštění.

Tvorba programu a kompilace

Program se píše v textovém editoru za použití speciální „jazyka“. Po zapsání textu programu je pak zapotřebí data „přeložit“ do dalších instrukcí, kterým mikrokontrolér bude rozumět. Tyto instrukce jsou pak uloženy do hex-souborů. Pro přenos hex-souborů do robota se pak používá „flash“ program. Pro zápis programu se používá speciálního textového editoru například Programmers Notepad2 (PN2). Použít však můžete i jiný textový editor (například MS Word), avšak přesto doporučujeme použití standardních programovacích editorů, které jsou vybaveny velkými množstvím speciálních funkcí. Vzhledem k tomu, že mikrokontrolér nerozumí programovacímu jazyku „C“, je zapotřebí provést jeho přeložení (kompilace) do jemu srozumitelného jazyka. Pro naprogramování robota proto bude nezbytné použití programu „C“ pro kompilaci a přeložení programovacího jazyka. Pro tyto účely může dobře sloužit program například Gcc.exe.

Načtení programu (Programm upload)

Poté, co kompilační program dokončil překlad „C“ programu „test.c“ do srozumitelných instrukcí pro mikrokontrolér (soubor „test.hex“), můžete přenést a načíst „hex“ soubor do paměti robota.

Základní struktura „C“ programu

Základní a minimální požadavky na strukturu „C“ programu:

```
int main(void){
return 0;
}
'int'           je typ pro hlavní funkci
'main'         je název pro hlavní funkci
'void'         indikace 'no entry'
'return 0'     je hodnota pro návrat funkce „main“
```

Jako základní pravidlo pro každou „C“ instrukci platí, že musí být ukončeno středníkem, vyjma zakončování závorkou. V editoru Programmers Notepad2 se automaticky zobrazují zdrojové kódy v předdefinovaných barvách v závislosti na zvláštních kategoriích. Syntaxe nebo specifická „C“ klíčová slova (keyword) budou podbarvena zeleně, číslice červeně a komentáře modře.

Můžete vložit virtuální komentář na libovolné místo, následovaný „/“ nebo mezi „/*“ a „*/“.

```
int main/*střední část textu*/(bez textu){ /*jakýkoliv text*/
//jeden řádek textu
/* řádky textu
*/
return 0; //libovolný text
/*
Další textové řádky
*/
}
```

V následujícím zdrojovém programu dojde k vygenerování chybového hlášení „compiler error“, pakliže napíšeme „Main“ s velkým písmenem na začátku slova. Jazyk „C“ nepodporuje psaní klíčových slov (například „main“) s velkými písmeny.

```
int Main(void){
return 0;
}
```

Na druhou stranu compiler umožňuje vytvoření řetězce pro všechny instrukce do jedné řádky, přestože tím nedochází k výraznému zlepšení čitelnosti programu:

```
int main(void){ //main function entry
return 0; //terminating the main function and
returning a value 0
} //end of the main function
```

#Comments can also be inserted above the program lines:

```
// main function entry
int main(void){
// terminating the main function and returning a value 0
return 0;
// end of the main function
}
```

V tomto návodu jste získali pouze základní náhled do programování a vysvětlení různých funkcí u programovacího jazyka „C“. Přesto však obsažení všech funkcí a principů programování je nad rámec tohoto návodu.

Důležité upozornění! Písmeno „v“ na začátku názvu funkce slouží jako doporučení při programování a označení funkce, která nezajistí návrat hodnoty!

Program pro rozsvícení levého oka robota

```
#This program will swich on YETI's right 'eye'
#include "YETI.h" //load definitions and functions
int main(void){ //main function entry
vlnitYeti(); //initialize all microprocessor modules in
//the YETI
vFrontLEDs(RIGHT); //swich on YETI's right 'eye'-LED
return 0; //terminating the main function and
//returning a value 0
} //end of the main function
```

#include "YETI.h"

Do tohoto místa vložíte kompilátor soubor 'YETI.h', který obsahuje všechny definice pro funkce, například funkce 'vYetiInit()'. Kódování řádků pro funkce najdete v souboru 'YETI.c'.

vlnitYeti();

Jedná se o speciální funkci, která aktivuje všechny mikrokontroléry a moduly robota. Funkci můžete volat příkazem **vlnitYeti();** na začátku programu robota kódování naleznete v souboru 'YETI.c'.

vFrontLEDs(LEFT);

Tato funkce aktivuje levé oko robota.

Ekvivalentní funkce jsou:

vFrontLEDs(RIGHT);

Aktivace pravého oka robota.

vFrontLEDs(OFF);

Zhasnutí obou očí.

vFrontLEDs(BOTH);

Rozsvícení obou očí.

5x probliknutí LED v pravém oku robota

Tento program zajistí, aby pravé oko robota probliklo celkem 5x s prodlevou 1 sekundy mezi jednotlivým rozsvícením.

```
#include "YETI.h" //load definitions and
//functions
int main(void){ //main function entry
int i; //define variable 'i' as an integer
vlnitYeti(); //initialize all microprocessor
//modules in the YETI
for(i=0 ;i<5 ;i++){ //repeat for-loop (5 times)
vFrontLEDs(RIGHT); //activate YETI's right eye LED
vWaitMilliseconds(500); //wait half a second
vFrontLEDs(OFF); //deactivate YETI's right eye LED
vWaitMilliseconds(500); //wait half a second
} //terminate for-loop (5 times)
return 0; //terminating the main function
} //end of the main function
for(...){
}
```

Programový modul vložený do závorek { a } se nazývá „loop“.

Int i; je definice proměnné s označení „i“.

vWaitMilliseconds(500);

Tuto funkci můžete nalézt v souboru 'YETI.c', který můžete otevřít v aplikaci PN2.

Tato funkce využívá interní mikrokontrolér a předdefinovaný počet milisekund a návrat na volání programu. Tato funkce akceptuje celou řadu milisekund (1/1000 s). Výše uvedená sekvence pro volání bude generovat cyklus prodlevy ½ sekundy.

Generování akustický signálů

```
#include "YETI.h" //load definitions and functions
int main(void){ //main function entry
vlnitYeti(); //initialize all microprocessor
//modules in the YETI
vBeep(400,80); //make YETI generate a beep sound
vBeep(520,120); //make YETI generate a beep sound
vBeep(360,80); //make YETI generate a beep sound
vBeep(580,160); //make YETI generate a beep sound
return 0; //terminating the main function
} //end of the main function
```

vBeep(400,80);

volání funkce 'vBeep' s intenzitou 400 a průběhem 8 x 10 = 80 milisekund.

Pohyb levé nohy robota

```
#include "YETI.h" //load definitions and functions
#include "yetimove.c" //insert YETI's servo functions
int main(void){ //main function entry
int i; //define variable 'i' as an integer
vlnitYeti(); //initialize all microprocessor modules in
//the YETI
for(i=0 ;i<3 ;i++){ //do for-loop 3 times
vMoveBody(16,10); //move body to the left
vMoveBody(-16,10); //move body to the right
} //terminate for-loop (3 times)
vMoveBody(0,10); //move body to the centre position
return 0; //terminating the main function
}
```

#include "yetimove.c"

Vložení funkcí pro servomotor.

vMoveBody(16,10);

Pohyb nohy robota (servomotoru) o 16 kroků doleva, v časovém úseku 10 milisekund pro každý krok.

vMoveBody(-16,10);

Pohyb nohy robota (servomotoru) o 16 kroků doprava, v časovém úseku 10 milisekund pro každý krok.

Pro krok motoru můžete volit pozice v rozsahu -58 až +58. Prodleva bude vždy zaokrouhlena směrem nahoru na vyšší hodnotu milisekund, což ve výsledku může představovat 10 milisekund na 1 krok motoru pro každou hodnotu parametru mezi 1 -10 milisekund a prodleva 20 milisekund na 1 krok motoru pro každou hodnotu parametru mezi 11 – 20 milisekund.

vMoveBody(0,10);

Reset „feet“ servomotoru robota a přechod do výchozí pozice (vzpřímená poloha).

Obdobná funkce 'vMoveLegs()' aktivuje pohyb nohou vpřed nebo dozadu.

Chůze robota

Tento program zajistí, aby robot provedl první 3 kroky směrem dopředu.

```
#include "YETI.h"           //load definitions and functions
#include "yetimove.c"       //insert YETI's servo functions
int main(void){           //main function entry
  vInitYeti();             //initialize all microprocessor modules in
                           //the YETI
  vMoveForwardXSteps(3);   //YETI will march forward 3 steps
  vStandUpright();        //makes YETI stand upright
  return 0;               //terminating the main function
}
```

vMoveForwardXSteps(3);

Robot vykoná 3 kroky směrem vpřed, přestože je ve výchozí (vzpřímené) poloze. Chůze začíná vždy nakročením jeho pravé nohy.

vStandUpright();

Příkaz pro uvedení robota do výchozí (vzpřímené) polohy.

Set pro rozšíření robota

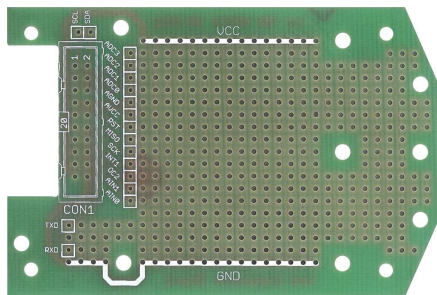
Veškeré přídatné moduly a příslušenství se připojuje do základní desky robota pomocí 20-pinového plochého kabelu. Kabel zároveň zajišťuje připojení přídatného modulu ke zdroji napájení a datovému přenosu sběrnice 12C ze stávajícího modulu.

YETI experimental set YT-EXP1

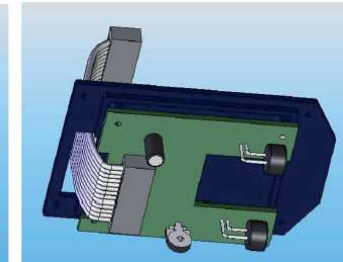
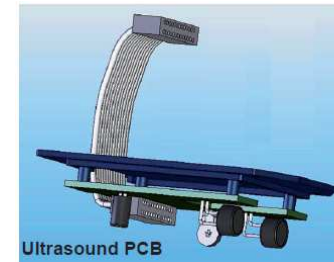
Tento set slouží pro konfiguraci vašich vlastních elektronických součástek a pro připojení experimentálních modulů do mikrokontroléru. Tato experimentální sada zahrnuje následující součástky:

PCB-DSP YETI
CON1-PCB
CON1-FC (2 ks)
F1

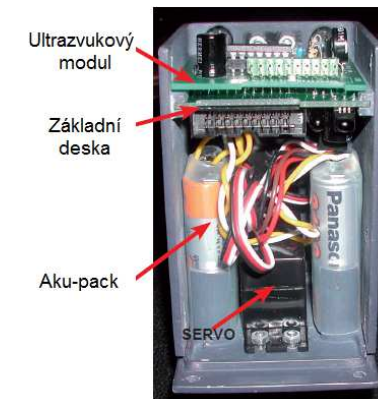
experimentální deska
PCB konektor, male (samec), 20-pinový pro plochý kabel
konektor pro připojení plochého kabelu, 20 pinový
plochý kabel, 20-vodičový, délka 10 cm



Instalace rozšiřujícího modulu



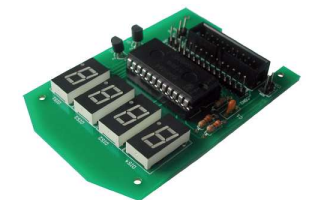
Instalace experimentálního modulu s LCD



YETI Display kit YT-DSP2

Tento displej disponuje zobrazením celkem 8 segmentů. Displej je zároveň vybaven 24-pinovým 12C ovladačem pro zobrazování symbolů displeje. 12C je standardní komunikační protokol pro datový přenos mezi elektronickými komponenty. Používá pouze 2 vodiče pro přenos signálu SCL (sériový čas) a SDA (sériová data). Na displeji se mohou zobrazovat číslice nebo zprávy.

Intenzitu displeje můžete nastavit v celkem 8 úrovních. Pro zobrazení je pokaždé zapotřebí 8 řádků, 1 řádek pro každý segment. Z datových řádků slouží k zobrazení 7 segmentů a jeden řádek pak desetinnou čárku. 7 segmentů a desetinná čárka obsahují LED. Pro každé zobrazení slouží jeden společný pin, který přivádí napájení od zdroje. V případě řízení signálu pro 4 jednotky displeje bude zapotřebí $8 + 2 = 10$ řádků pro každý symbol, což vyžaduje použití IC se 40 piny.



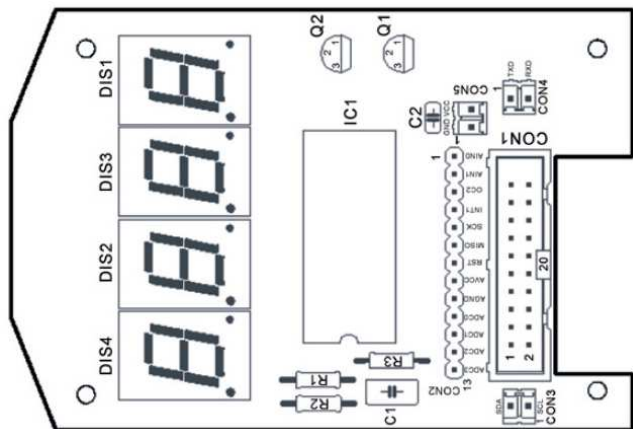
Použijeme proto malého triku a multiplexního systému, který provádí 2 sady 8-segmentových pinů: P1 – P8 a P9 – P16. Při použití P1 – P8 se stejnou bitovou kombinací pro zobrazení speciálního symbolu, například „X“, jednotka 1 a 2 zobrazí tento speciální symbol „X“. Nyní je nezbytné aktivovat displej 1 a deaktivovat displej 2 pomocí tranzistoru Q1 a Q2, po aktivaci displeje 1.

Posloupnost displeje: Displej - Displej 2 – Displej 3 – Displej 1

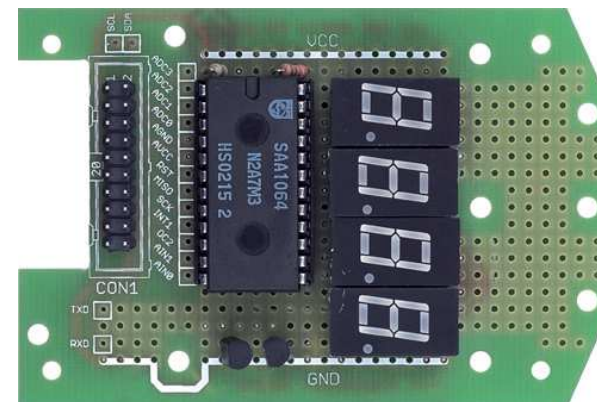
Seznam komponentů LCD modulu Y-DSP2

PCB-DSP	Deska spojů YETI DISPLAY PCB
R1	rezistor 330 Ω (oranžová, oranžová, hnědá, zlatá)
R2	rezistor 330 Ω (oranžová, oranžová, hnědá, zlatá)
R3	18 KΩ (hnědá, šedá, oranžová, zlatá)
C1	kondenzátor 2,7nF (označení na pouzdře 272)
C2	100 nF (104)
Q1	tranzistor BC 547B/C (při zapojování dbejte správné polarity)
Q2	tranzistor BC 547B/C (při zapojování dbejte správné polarity)
D1	8-segmentový displej, společná anoda
D2	8-segmentový displej, společná anoda
D3	8-segmentový displej, společná anoda
D4	8-segmentový displej, společná anoda
IC1	SAA 1064 (při zapojování dbejte správné polarity)
S1	patice pro IC, 24-pinů, 600mil
CON2	Pin header 13 -pinů, montáž na desku
CON, 3, 4 and 5	(3 ks) Pin header 2 -piny, montáž na desku
CON1-PCB	konektor, male, 20-pinový, pro plochý kabel
CON1-FC	(2 ks) konektor pro plochý kabel, 20-pinový, female
F1	plochý kabel, 20-vodičový, 10 cm

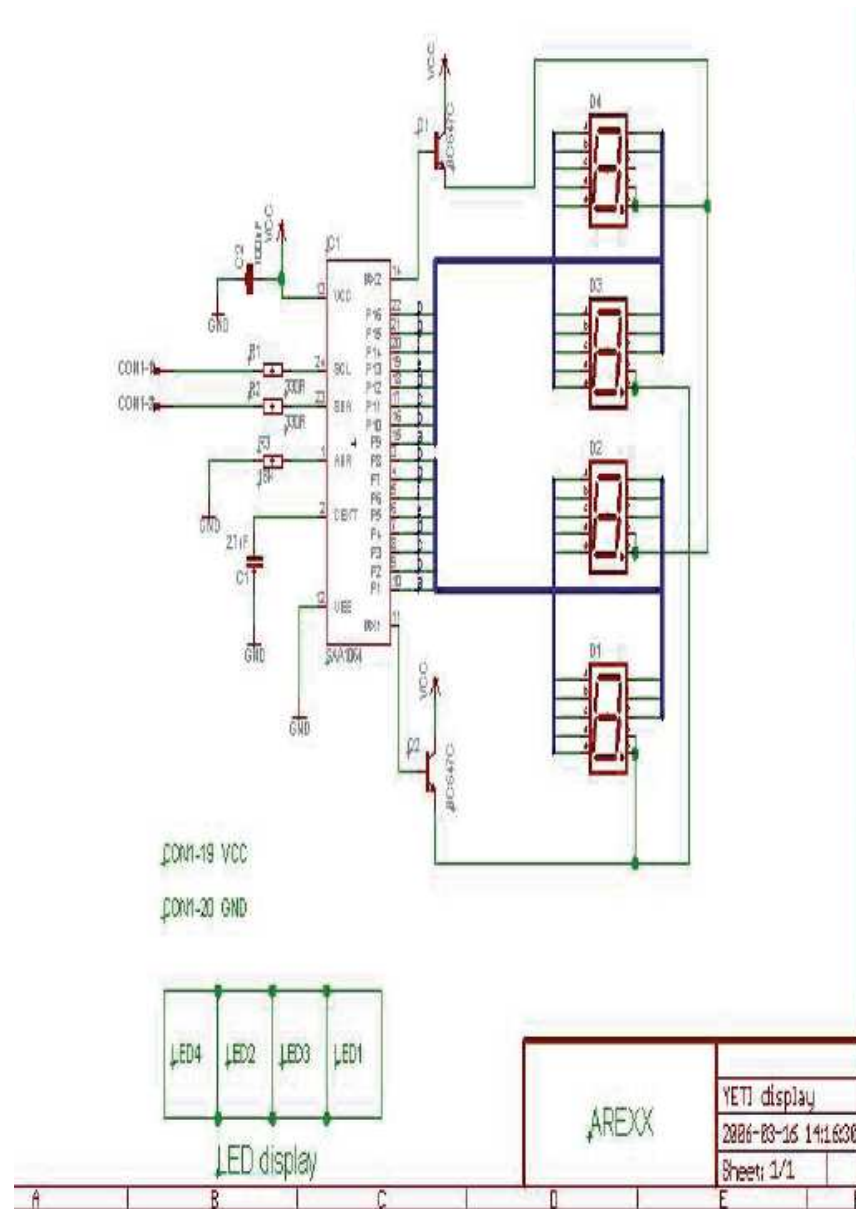
Displej instalovaný na originální desce PCB YT-DSP2 KIT



Displej instalovaný na experimentální desce



Schématické zapojení komponentů na desce YT-DSP



Ultrazvukový modul YT-ULT3

Tato experimentální sada obsahuje ultrazvukový vysílač a přijímač. Ultrazvukové vlny jsou zvukové vlny s relativně vysokou frekvencí, které však již nedokáže lidský sluch nijak vnímat. V přírodě používají netopyři ultrazvuk pro prostorovou orientaci během letu i za naprosté tmy. Tato metoda orientace se nazývá echolokace. Veškeré překážky a předměty odrážejí vyslané ultrazvukové vlny a z jejich odrazu si netopýr udělá velmi přesný obraz o okolním prostředí.

Tento robot používá pro vlnovou orientaci mikrofon. Vysílač emituje akustické impulzy (vlnové dávky) s frekvencí okolo 40000 Hz. Přijímač detekuje signály, odražené od okolních předmětů a vyhodnocuje prodlevu mezi odesláním a přijetím signálu. Prodleva umožňuje vypočítat vzdálenosti mezi přijímačem / vysílačem a okolními předměty. Ultrazvukový modul dokáže převést časovou prodlevu na elektrické napětí s určitou úrovní a plochý kabel připojený k analogovo-digitálnímu převodníku v procesoru robota pak tuto úroveň monitoruje. Softwarový modul pak monitoruje napětí a díky tomu dokáže řídit pohybu robota.

Seznam jednotlivých komponentů ultrazvukového modulu

1. Vysílač
2. Přijímač
3. Zesilovač přijímače
4. Fixní hodnota referenčního napětí
5. Proměnná hodnota napětí

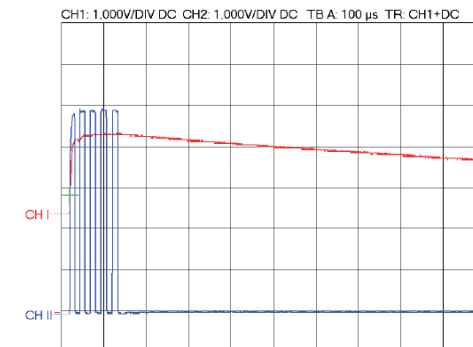
Mikrokontrolér generuje ultrazvukové vlny, které jsou vyslané prostřednictvím reproduktoru vysílače (TX). Mikrofon přijímače (RX) přijímá odražené zvukové vlny, které se však musejí zesílit v zesilovači přijímače. Rezistor R10 umožňuje manuální ovládání faktoru zesílení. Hodnota referenčního napětí, která je přizpůsobena na hodnotu 50 % napájecího napětí, se používá pro vysílač a pro generování proměnné hodnoty napětí. Generovaný ultrazvukový signál vstupuje do modulu na pinu CON1 – 13. Odražený signál je monitorován mikrofonem a vrácen do mikrokontroléru na pinu CON1 – 6.

Generovaný ultrazvukový signál opouští mikrokontrolér a vstupuje do modulu vysílače na pinu CON1 – 13 a rezistoru R3. Vysílač obsahuje 2 zvláštní zesilovače v čipu IC1 z celkového počtu 4. zesilovačů. Jedná se o tzv. operační zesilovač (Opamp). Operační zesilovače generují maximální energii do reproduktoru a přijímají slabý a zkreslený mikrofonní signál. Ultrazvukový mikrofon detekuje odražený signál a transformuje jej do elektrického signálu, který může být zesilovačem odfiltrován v operačním zesilovači / přijímači Opamp IC1B. Nastavitelný rezistor pak umožňuje ovládání faktoru zesílení celého systému.

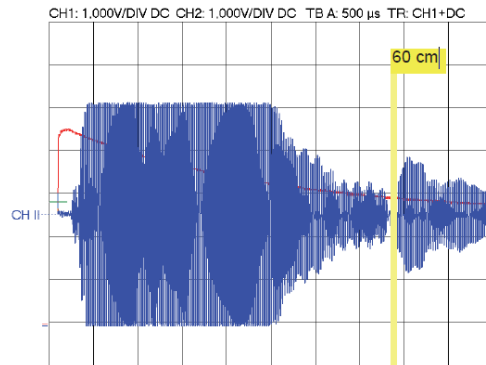
Ultrazvukový systém má extrémně vysokou citlivost pro detekci signálu odraženého od okolních předmětů, zejména signál předmětů v blízkosti vysílače. Aby se zabránilo odrazům signálu od hlavy robota doporučujeme proto kompletně vyplnit hlavu robota například vatou.

Sestavený ultrazvukový modul se instaluje přímo do hlavy robota. Přesto má umístění tohoto modulu výrazný vliv na správnou funkci tohoto systému, vzhledem k nežádoucím odrazům ultrazvuku uvnitř hlavy robota a nežádoucím odrazům v okolí robota.

Charakteristiky ultrazvukových vln



Připojením ultrazvukového signálu do osciloskopu jsou patrné 2 signály, měřené na pinu CON1 – 13 a CON1 – 15. První signál (modrý) pinu CON1 – 13 zobrazuje 5 pulzů s hodnotou přibližně 4,5 V. Tento signál přichází z mikroprocesoru přímo a je dále předáván do dalších komponentů modulu. Druhý signál (červený) na pinu CON1 – 15, je referenční signál přicházející z modulu, který se pak vrací do procesoru. Pakliže odraz zaznamená napětí nad hodnotou referenčního signálu, dojde k jeho vyhodnocení a měření.

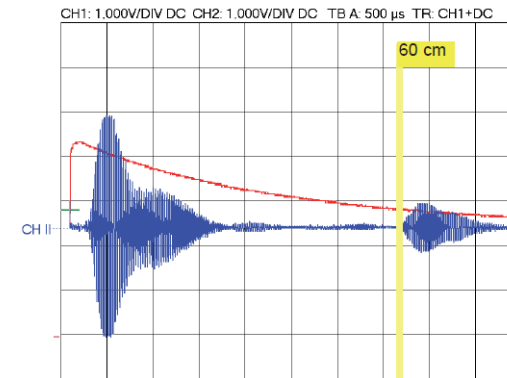


Na obrázku výše jsou patrné 2 druhy signálu. Červený referenční signál CON1 – 13 a přijímaný signál (modrý) CON1 – 6, signál přijatý a odražený ultrazvukovým modulem, který odchází ke zpracování do mikroprocesoru na pinu CON1 – 6. Na grafu je označený signál ve vzdálenosti 60 cm. Všechny silně odražené signály jsou umístěny nad referenční linkou coby výsledek nežádoucích odrazů uvnitř hlavy robota. Takové signály pak způsobují výrazné zkreslení signálu a pořízení nesprávných výsledků měření.

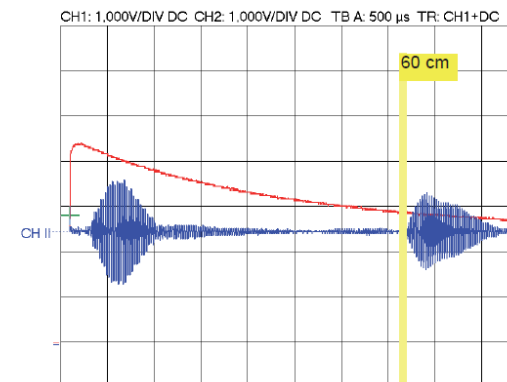
Hlava robota vyplněná vatou



Na následující obrázku je zaznamenána nová situace. Nežádoucí signál je na něm daleko lépe patrný. Odrazy, které jsou silnější, než je referenční signál, nejsou zaznamenány jako platná hodnota pro měření a jedná se tak o chybové měření.



Po vyplnění vnitřku hlavy robota vatou dojde k odfiltrování vnitřních odrazů ultrazvuku.



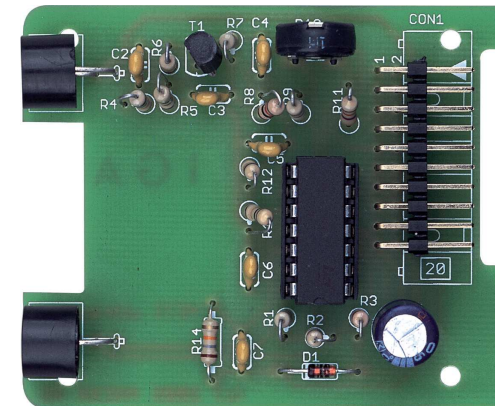
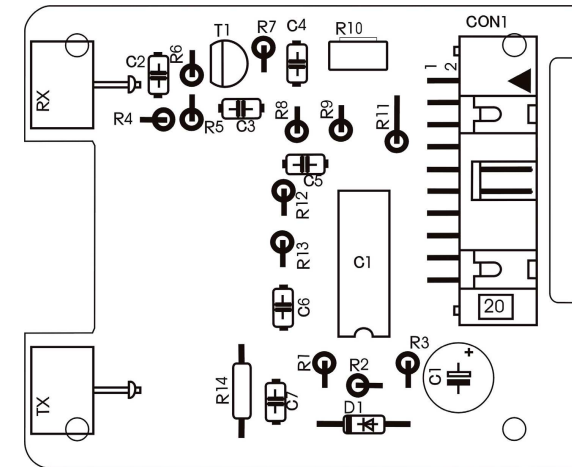
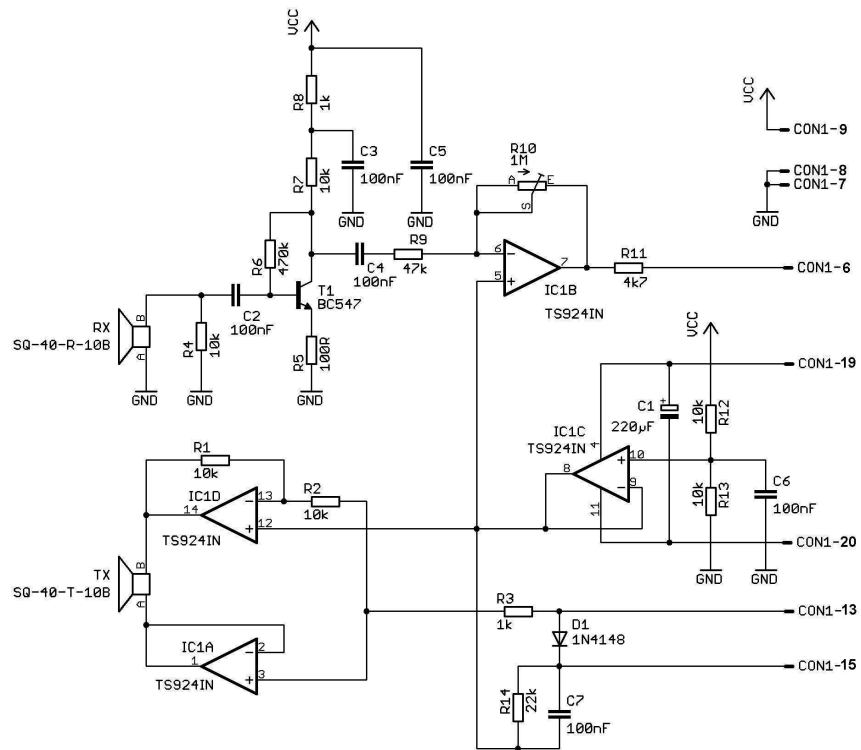
Za tohoto stavu jsou všechny odražené signály pod referenční linkou. Díky tomu může dojít k velmi přesnému měření vzdálenosti ultrazvukem. Doba mezi 5. vyslanými impulzy a odraženým signálem je vypočítán mikroprocesorem a převeden do aktuální vzdálenosti.

Seznam komponentů ultrazvukového modulu

R1	10 KΩ (hnědý, černý, oranžový, zlatý)
R2	10 KΩ (hnědý, černý, oranžový, zlatý)
R3	1 KΩ (hnědý, černý, červený, zlatý)
R4	10 KΩ (hnědý, černý, oranžový, zlatý)
R5	100 RΩ (hnědý, černý, hnědý, zlatý)
R6	470 KΩ (žlutý, fialový, žlutý, zlatý)
R7	10 KΩ (hnědý, černý, oranžový, zlatý)
R8	1 KΩ (hnědý, černý, červený, zlatý)
R9	47 KΩ (žlutý, fialový, oranžový, zlatý)
R10	TRIMMER 1 MΩ potenciometr
R11	4K7 (žlutý, fialový, červený, zlatý)
R12	10 KΩ (hnědý, černý, oranžový, zlatý)
R13	10 KΩ (hnědý, černý, oranžový, zlatý)
R14	22 KΩ (červený, červený, oranžový, žlutý)

C1	kondenzátor 220 uF (dbejte zachování správné polarity)
C2	kondenzátor 100 nF (označení 104)
C3	kondenzátor 100 nF (označení 104)
C4	kondenzátor 100 nF (označení 104)
C5	kondenzátor 100 nF (označení 104)
C6	kondenzátor 100 nF (označení 104)
C7	kondenzátor 100 nF (označení 104)
IC1	TS924IN (Quad Opamp) (dbejte zachování správné polarity)
S1	IC-patice, 14-pinů (dbejte zachování správné polarity)
TX	400ST100 (Ultrasonic transmitter) vysílač, malý
RX	400SR100 (Ultrasonic receiver) přijímač, small
T1	tranzistory BC547B/C nebo BC547B/C (dbejte zachování správné polarity)
D1	dioda 1N4148 (dbejte zachování správné polarity)
CON1-PCB	PCB konektor, male, 20-pinový, pro plochý kabel
CON1-FC (2 ks)	Konektor pro plochý kabel, 20-pinový, female
F1	plochý kabel, 20 vodičů, délka 10 cm

Schématický náčrt ultrazvukového modulu



Přehled funkcí robota Yeti

Základní funkce

V následující části budou popsány základní funkce robota, které naleznete v souboru „Yeti.c“.

vInitYeti()

Inicializace všech modulů robota.

vFrontLEDs(x)

x = ON, LEFT, RIGHT, OFF

Aktivace očí robota.

Příklad: vFrontLEDs(LEFT);

vServo1ToPosition(x)

x = 0-65635

Seřízení předního „body“ servomotoru do pozice x.

Upozornění! Tato funkce může seřadit servomotor do libovolné pozice, včetně libovolné mechanické pozice, která však není k dispozici pro systém robota. Použitím těchto příkazů může dojít k nevratnému poškození robota. Doporučujeme proto použít bezpečnější funkci `vMoveBody()`.

Příklad: vServo1ToPosition(35);

vServo2ToPosition(x)

x = 0-65635

Seřízení předního (legs) servomotoru do pozice x. **Upozornění!** Tato funkce může seřadit servomotor do libovolné pozice, včetně mechanické pozice, které však není k dispozici pro systém robota.

Použitím těchto příkazů může dojít k poškození robota. Doporučujeme proto použít bezpečnější funkci `vMoveLegs()`.

Příklad: vServo1ToPosition(35);

vRs232Write(x,y)

x = text

y = text length, 0-255

Příklad:

void vRs232Write("Hello World",11) ;

vRs232Read(x,y,z)

x = pointer to an array to store a text for reception

y = expected text length

z = timeout

Příklad:

char RxData[10] ;

vRs232Read(&RxData[0],4,0);

Po přijetí 4. symbolů se funkce ukončí.

vWaitMilliseconds(x)

x = waiting time in milliseconds, 0 - 65635

Tato funkce zaokrouhlí interval v milisekundách, například čas 23 na 30.

Příklad:

Prodleva 1 s.

vWaitMilliseconds(1000);

vBeep(x,y)

x = pitch number, 1-3906

y = duration in milliseconds.

Příklad:

vBeep(200,100);

Kombinované funkce

Následující funkce jsou kombinacemi několika funkcí, jež zahrnují použití jedné nebo více základních funkcí. Tyto funkce naleznete v souboru „yetimove.c“.

vStandUpright()

Tento příkaz zajistí uvedení robota do vzpřímené pozice. Oba servomotory „body“ a „legs“ budou seřizené do výchozí, nulové pozice.

vMoveBody(x,y)

x = tělo robota se naklání vpravo nebo vlevo, z polohy -58 (korespondující s krajní pravou polohou) do polohy -58 (korespondující s krajní levou polohou).

y = execution speed in milliseconds / step, 0 – 65635.

Tato funkce zaokrouhlí časovou prodlevu na desetiny milisekund například 17 na 20.

Příklad:

vMoveBody(-25,20);

Ze vzpřímené pozice bude vyžadovat příkaz vMoveBody(-25,20) dobu 25*20=500 ms= 0,5 sekundy k dosažení náklonu na pravou stranu a polohy -25.

vMoveLegs(x,y)

x = pravá noha robota se přesune vpřed nebo vzad (a opačná noha se bude přesouvat do opačné polohy a směru) v rozsahu -58 (pravá noha vpřed) a +58 (levá noha vpřed) rychlost na vykonání pohybu v milisekundách / krok, 0 – 65635. Tato funkce zajistí zaokrouhlení časového průběhu na desetiny milisekund, například 3 na 10.

Příklad:

vMoveLegs(-25,20);

Ze vzpřímené polohy pomocí příkazu vMoveLegs(-25,20) bude zapotřebí 25 x 20=500 ms= 0,5 sekundy na pohyb pravé nohy robota do přední pozice -25 (zatímco levá noha se bude pohybovat dozadu).

vMoveForwardXSteps(x)

x = počet kroků vpřed, 0-255

Příklad:

vMoveForwardXSteps(3);

Robot vykoná 3 kroky směrem dopředu. Robot se ze vzpřímené pozice rozejde vykročením pravou nohou.

vMoveBackwardXSteps(x)

x = počet kroků dozadu, 0-255

Příklad:

vMoveBackwardXSteps(4);

Robot vykoná 3 kroky směrem vzad. Ze vzpřímené pozice se rozejde vykročením pravou nohou.

vTurnLeftXSteps(x,y)

x = počet kroků, 0-255

y = 'true' respectivilly 'false', true = pohyb vpřed, false = pohyb vpřed.

Příklad:

vTurnLeftXSteps(2,false);

Robot se bude otáčet dozadu, přitom vykoná 2 kroky během svého otáčení vlevo.

vTurnRightXSteps(x,y)

x = počet kroků, 0-255

y = 'true' respectivilly 'false', true = pohyb vpřed, false = pohyb vpřed.

Příklad:

vTurnLeftXSteps(3,true);

Robot se bude otáčet dozadu, přitom vykoná 3 kroky během svého otáčení vpravo.

vCalibrateServos()

V kombinaci s komunikací prostřednictvím software Hyperterminal bude tato funkce provádět rekalicaci servo systému pokaždé, když upravíte mechanické pozice servomotorů nebo končetin robota. Stejně tak může být tato funkce použita pro dálkové ovládání robota a jeho servomotorů z klávesnice prostředí Hyperterminal pomocí tlačítek W, A, S a D.

Poznámka: V případě provedení kalibrace servomotorů můžete tuto funkci ze souboru odstranit. Tím dojde k nepatrnému zmenšení objemu programového souboru.

Schéma ultrazvukového modulu

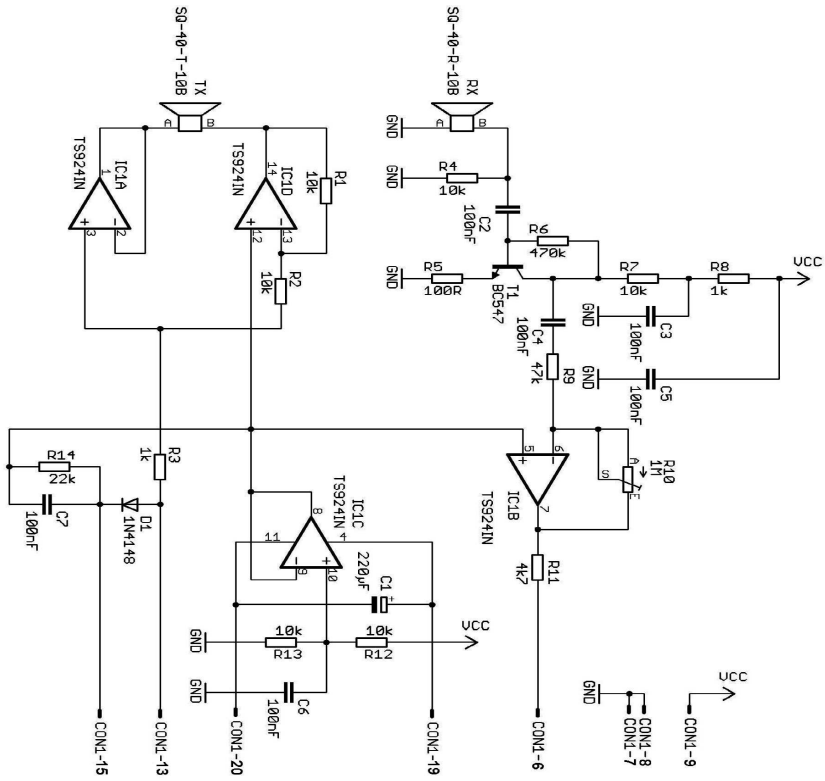


Schéma IR Transceiveru RS-232

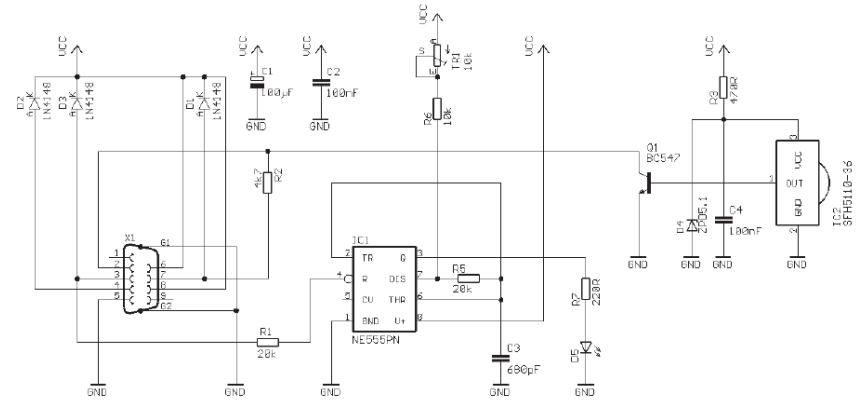
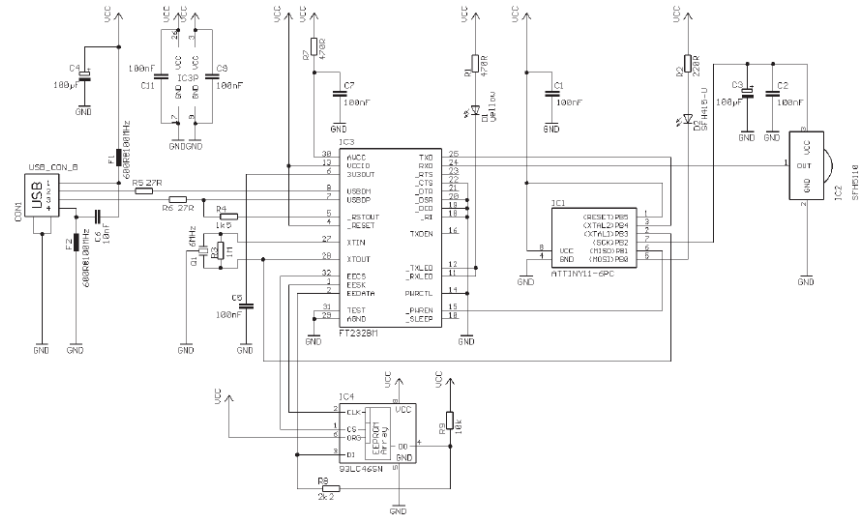


Schéma USB modulu IR Transceiveru RS-232



Osazení pinů plochého kabelu

Pin 1 SCL	Serial Clock (pro I2C komunikaci)
Pin 2 SDA	Serial Data (pro I2C komunikaci)
Pin 3 PC3 (ADC3)	Digital input/output nebo analog monitor input
Pin 4 PC2 (ADC2)	Digital input/output nebo analog monitor input
Pin 5 PC1 (ADC1)	Digital input/output nebo analog monitor input
Pin 6 PC0 (ADC0)	Digital input/output nebo analog monitor input
Pin 7 GND	GND (konektory pro eliminaci signal noise)
Pin 8 GND	GND (konektory pro eliminaci signal noise)
Pin 9 AVCC	Analog reference-voltage pro AD-converters
Pin 10 PC6 (RESET)	Microcontroller reset pin
Pin 11 PB5 (SCK)	Digital input/output
Pin 12 PB4 (MISO)	Digital input/output nebo I2C function pin
Pin 13 PB3 (MOSI/OC2)	Digital input/output nebo I2C function pin nebo Timer2 pin
Pin 14 PD3 (INT1)	Digital input/output nebo external interrupt
Pin 15 PD6 (AIN0)	Digital input/output nebo analog testinput
Pin 16 D7 (AIN1)	Digital input/output nebo analog testinput
Pin 17 PD0 (RXD)	Digital input/output nebo RS232 input
Pin 18 PD1 (TXD)	Digital input/output nebo RS232 input
Pin 19 VCC	VCC
Pin 20 GND	GND (konektory pro eliminaci signal noise)

Předcházení chybovému stavu

Před uvedením robota do provozu ověřte veškeré zapojení a použití správných parametrů. Ujistěte se zejména o správné poloze / polaritě všech komponentů osazovaných do modulu. Ověřte, zda se v obvodu nenacházejí žádné nežádoucí spoje v důsledku použití většího množství pájky. Pakliže jsou všechna zapojení a komponenty ověřeny a nevykazují žádné závady, můžete pomocí osciloskopu vyhledávat například zničený komponent. Použijte proto připojených schémat.

Chybový stav IR Transceiveru RS232

Aktivovaný znak a zobrazovaný symbol nekorespondují: Kalibrujte trimr TR1 dokud nedojde k správnému zobrazení aktivovaného znaku.

Terminál programu nezobrazuje žádný znak: Ujistěte se o správné poloze (montáži) časovače IC1. Použijte libovolný dálkový ovladač nebo multimediální zařízení HiFi / video (videorekordér nebo TV) a nasměřujte na IR transceiver a stiskněte přitom několik tlačítek. Pakliže terminál zobrazuje neznámé a nečitelné symboly znamená to, že receiver (IC2, R3, C4, D4, T1) funguje správně a musíte prověřit všechny zbývající komponenty. Nejčastěji se však stává, že dojde k poškození některého z následujících komponentů: IC1, IC2, Q1 a D4.

USB IR Transceiver nefunguje

Windows: Ujistěte se o tom, že jsou nainstalovány všechny potřebné ovladače a je použitý správný komunikační (COM) port.

Linux: Odpojte USB transceiver a před jeho opětovným připojením vyčkejte několik minut. Tento proces by měl veškeré problémy vyřešit. Další možností je instalovat nové jádro.

IR rozhraní / Robot nedodesílá žádné symboly

Ujistěte se o polaritě IR diody D10, ověřte rezistor R16 220 Ω (červený, červený, hnědý, zlatý).

Robot nepřijímá žádné symboly

Zajistěte přímou viditelnost mezi IR transceiverem a robotem (max. vzdálenost do 50 cm) a IR transceiver musí být plně funkční. Ověřte polaritu C2. Ujistěte se o funkci rezistoru R17 470 Ω (červený, červený, hnědý, zlatý) a kondenzátoru C2 100 nF (označení na použít 104).

Pakliže nemůžete nalézt příčinu potíží, ujistěte se o tom, že nedošlo k poškození integrovaného obvodu IC2. Při pájení může dojít k jeho tepelnému přetížení a tím i jeho nevratnému zničení. V takovém případě jej vyměňte za nový IC (SFH 5110-36). Pokud přenos dat mezi počítačem a robotem i nadále nefunguje, přizpůsobte nastavení trimru TR1 u transceiveru. Ověřte správnou polaritu kondenzátoru C8 220 μF / 10 V a pokud to bude znovu nezbytné i nastavení trimru TR1.

Kalibrace a testovací software

Robot je z výroby dodáván s otestovaným systémem a kalibračním programem „test.hex“. Tento program obsahuje kalibrační režim a režim chůze.

Proces auto-test robota

Jako první robota zapněte. Bootloader zajistí spuštění programu. Zároveň přitom se rozsvítí obě LED. Oba servomotory se neopatrně pootočí ve směru hodinových ručiček. Po uplynutí dalších 3. sekund zajistí Bootloader ukončení programu.

Spuštění programu „Test.hex“

Jako první dojde k vypnutí obou LED. Obě serva se přesunou do středové polohy. Zazní série krátkých akustických tónů. Program poté vyčkává po dobu 3. sekund na stisk libovolného tlačítka rozhraní Hyperterminal. Po stisku libovolného tlačítka přejde robot do režimu kalibrace a informuje o tom, že robot přijal infračervený signál. To znamená, že IR transceiver funguje správně. Stiskem tlačítka ENTER dojde k ukončení režimu kalibrace. Robot poté přejde do režimu chůze. Pokud však nedojde ke stisku žádného tlačítka po dobu 3. sekund, režim kalibrace se automaticky ukončí a robot přejde do režimu chůze.

Spuštění programu chůze

Na začátku robot vygeneruje krátký tón a zároveň přitom se rozsvítí levá LED. Robot se nakloní vlevo a vykročí pravou nohou a poté vykoná celkem 3 kroky. Následně se robot otočí vlevo a ujde 4 kroky. Následně robot vykoná 3 kroky zpět, otočí se vpravo a provede další 4 kroky. Robot zatřese tělem (vpravo a vlevo) celkem 3x. Dále bude systém robota opakovat všechny tyto akce. Na konci jednotlivé sekvence robot vygeneruje krátký tón a zajistí kombinaci stavu LED (vypnutí nebo zapnutí). Od spuštění bude robot vykonávat tuto akci v nekonečné smyčce (loop).

Důležité! Ujistěte se o tom, že robot při zahájení chůze vždy vykročí pravou nohou. Pokud tomu tak není, došlo zřejmě k nesprávnému (opačnému) zapojení servomotorů.

Manipulace s bateriemi a akumulátory



Nenechávejte baterie (akumulátory) volně ležet. Hrozí nebezpečí, že by je mohly spolknout děti nebo domácí zvířata! V případě spolknutí baterií vyhledejte okamžitě lékaře! Baterie (akumulátory) nepatří do rukou malých dětí! Vytéklé nebo jinak poškozené baterie mohou způsobit poleptání pokožky. V takovém případě použijte vhodné ochranné rukavice! Dejte pozor nato, že baterie nesmějí být zkratovány, odhazovány do ohně nebo nabíjeny! V takovýchto případech hrozí nebezpečí exploze! Nabíjet můžete pouze akumulátory.



Vybité baterie (již nepoužitelné akumulátory) jsou zvláštním odpadem a nepatří do domovního odpadu a musí být s nimi zacházeno tak, aby nedocházelo k poškození životního prostředí!



K těmto účelům (k jejich likvidaci) slouží speciální sběrné nádoby v prodejnách s elektrospotřebiči nebo ve sběrných surovinách!

Šetřete životní prostředí!

Recyklace



Elektronické a elektrické produkty nesmějí být vhažovány do domovních odpadů. Likviduje odpad na konci doby životnosti výrobku přiměřeně podle platných zákonných ustanovení.

Šetřete životní prostředí! Přispějte k jeho ochraně!



Pokud si nebudete vědět rady, jak tohoto programovatelného robota správně a bezpečně používat a v tomto návodu k obsluze nenaleznete všechny potřebné informace, obraťte se na naši technickou podporu, nebo požádejte o radu kvalifikovaného odborníka.