

Programming documentation

Martin Červeň & Rastislav Galváneš

Dokumentácia k dance robotovi

Praha 2019

Contents

1	Choreografia a ovládanie robota	2
1.1	Uloženie EEPROM	2
1.2	Defaultná choreografia	2
1.3	Custom choreografia	2
1.4	Výber choreografie	2
1.5	Zahájenie vykonávania choreografie	2
2	Načítanie a spracovanie Custom choreografie	4
2.1	Špecifikácia custom choreografie	4
3	Pohyb robota	5
3.1	Pohnutie sa o políčko dopredu	5
3.2	Otočenie sa doprava/dolava	5
3.3	Pohnutie robota na požadovanú pozíciu	5

1. Choreografia a ovládanie robota

Podľa zadania robot môže použiť buď defaultnú, alebo custom choreografiu nahranú z EEPROM.

1.1 Uloženie EEPROM

Choreografia je na EEPROM uložená nasledovne: prvý bajt určuje štartovaciu orientáciu (N,W,E,S). Potom nasledujú jednotlivé koordináty, skladajúce sa z pozície na ktorú má ísť ako prvú, pozície, na ktorú má ísť ako druhú a času, kedy môže túto pozíciu opustiť ($2 \times \text{char} + 1 \times \text{unsigned int} = 6$ bajtov). Koniec každej choreografie je označený špeciálnym koorinátom q,q,0.

1.2 Defaultná choreografia

Defaultná choreografia začína na prvom bajte bajte v EEPROM, kde sa nahráva v setupe a môže byť dlhá 510 bajtov (čo znamená 82 inštrukcií pre robota, kde ísť, v prípade, že by sme nepoužívali custom choreografiu, je možné nahráť ďalších 85 inštrukcií).

1.3 Custom choreografia

Custom choreografiu je možné po resetovaní robota nahráť za pomoci serial portu. V prípade, že nahrávaná choreografia nie je validná, treba robota resetovať, aby mohla byť nahraná nová platná custom choreografia - inak bude použitá defaultná choreografia. Custom choreografia sa nahráva od 512 bajtu, teda môže byť dlhá 83 inštrukcií (nepočítajúc počiatočnú pozíciu).

1.4 Výber choreografie

Výber choreografie je určený nultým bajtom v EEPROM. V prípade, že je nultý bajt nastavený na 0 robot sa riadi defaultnou choreografiou. V prípade, že je v bajte nastavená 1, robot vykonáva custom choreografiu. Bajt je nastavený na nulu, ak sa po resete robota nepokúsime zadať žiadnu custom choreografiu, alebo ak je zadávaná custom choreografia nevalidná. Ak sme serial portom zadali validnú custom choreografiu, vykonávať sa bude tá.

1.5 Zahájenie vykonávania choreografie

Po stlačení tlačidla na robotovi začne okamžite vykonávať choreografiu. Po jej skončení robot ostane čakať na koncovom políčku. Po opätovnom stlačení tlačidla sa vráti späť do štartovacej pozície podľa toho, ako bola zadaná (1A, znamená, že najprv pôjde na prvý riadok, A1 by zas znamenalo, že najprv pôjde na stĺpec A

). Po stlačení tlačidla v štartovacej pozícii robot znova začne vykonávať vybratú choreografiu.

2. Načítanie a spracovanie Custom choreografie

2.1 Špecifikácia custom choreografie

Custom choreografia sa skladá z postupnosti tokenov. Token je postupnosť znakov na vstupe ktoré nie sú biele znaky. Medzi každým tokenom môžu byť ľubovoľné biele znaky. Pri načítaní je použitý stavový automat s štyrmi stavmi:

```
reading_starting_coordinate_state,  
reading_coordinate_state,  
reading_time_state,  
malformed_input_state
```

Najprv je program v stave

`reading_starting_coordinate_state`

Ako prvá musí byť na vstupe štartovacia pozícia. Tá sa skladá z koordinátu (riadok, stĺpec, smer) alebo (stĺpec, riadok, smer) kde smer sú znaky N, E, S, W značiaci sever, východ, juh a západ. koordináty sú znaky v rozmedzí A-I, 1-9.

Po načítaní platnej štartovacej pozície prejde program do stavu

`reading_coordinate_state`

A program očakáva platný koordinát, ktorý má tvar (stĺpec, riadok) a obsahuje znaky v rozmedzí A-I, 1-9. Potom prejde program do stavu:

`reading_time_state`

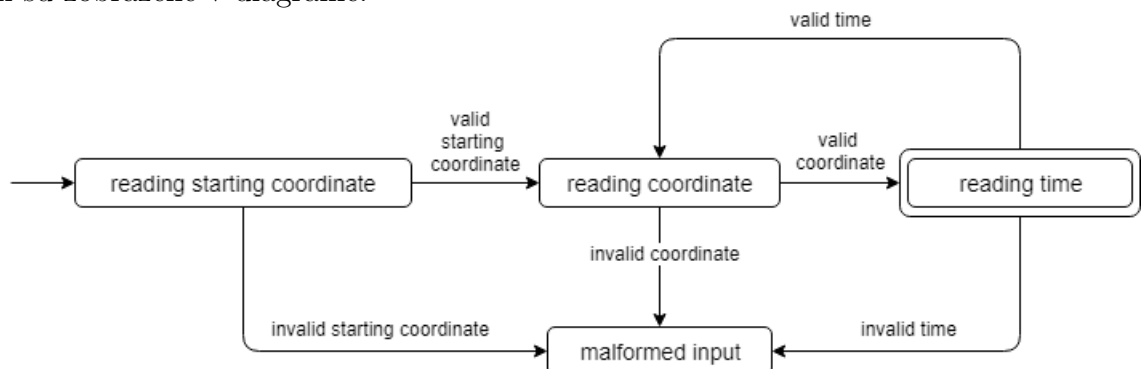
kde musí nasledovať token času ktorý sa skladá z písmena T a až 5 čísel 0-9. To znamená, že maximálny čas pre danú súradnicu je 9999,9 sekundy.

Potom program prijíma ďalšie dvojice koordinát, čas.

Ak niektoré tokeny nie sú validné, program vojde do stavu

`malformed_input_state`

a bude ignorovať ďalší vstup až kým sa robot neresetuje. Stavy a prechody medzi nimi sú zobrazené v diagrame.



3. Pohyb robota

Robot si v každom kroku pamätá svoju štartovaciu pozíciu a orientáciu a tiež aktuálnu poíciu a orientáciu, ktorú po každom otočení alebo prejdení jedného "políčka" aktualizuje. Robot sa stále udržiava tak, aby bol osou otáčania v križovatke.

3.1 Pohnutie sa o políčko dopredu

Tri prostredné senzory slúžia na to, aby robot sledoval čiaru a dva krajné senzory sledujú, či už prešiel jedno políčko. Sledovanie čiary je riešené tak, že ak má čiaru pod stredným alebo medzi stredným a krajným sensorom ide dopredu, ak je čiaru pod krajným sensorom zatača na správnu stranu, aby sa vyrovnal. Sledovanie prejdenia políčka je vyriešené tak, že ak je robot oboma najkrajnejšími senzormi na bielom podklade (nenachádza sa pod nimi čiaru) a následne aspoň jeden senzor zachytí čiaru, vieme, že robot sa dostal na ďalšiu križovatku. Následne robot ešte pokračuje nejaký experimentálne určený čas dopredu, aby zastavil osou otáčania nad križovatkou a bol pripravený na otočenie.

3.2 Otočenie sa doprava/doľava

Otáčanie je vyriešené jednoducho tak, že robot sa otáča doprava alebo doľava na mieste, kým sa stredný senzor nesníma čiernu čiaru. Snímať však začne až po určitom čase (300ms) aby sa zabránilo zastaveniu na čiare (smeru), z ktorej vychádzal. Po zosnímaní čiary stredným sensorom sa ešte robot pár milisekúnd otáča, aby sa dostal do stredu.

3.3 Pohnutie robota na požadovanú pozíciu

Pohnutie robota na požadovanú pozíciu je následne už len opakované poižitie dvoch hore popísaných krokov. V prípade že je robot na 1 riadku respektíve v stĺpci A sa otáča tak, aby s aneotáčal smerom, kde na kraji chýba čiaru a nemal by s apodľa čoho riadiť. Ak je na pozícií inej, otáča sa vždy na opačný smer.