TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

SYSTÉM PRE VYTVÁRANIA MAPY PROSTREDIA

Príloha C: Systémová príručka

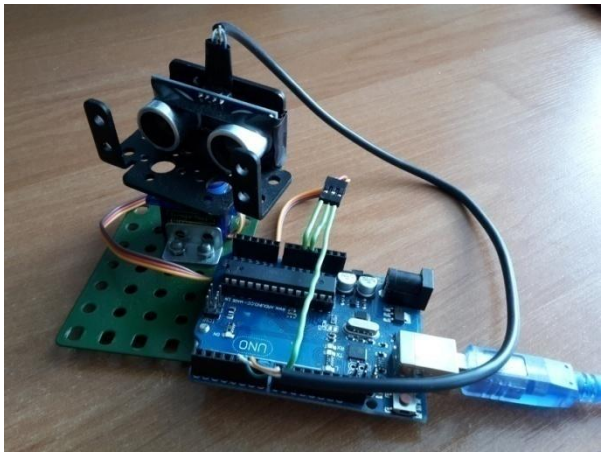
2015 **Matej Kurinec**

Obsah

Zoznam obrázkov

1. Funkcia programu

Systém pre vytvárania mapy prostredia je systém pozostávajúci zo softvérových programov a hardvérových komponentov, slúžiaci na snímanie parametrov prostredia a následne vytvorenie jeho mapy. Pozostáva z dvoch hlavných častí, a to aplikácie, ktorá slúži ako komunikačný prostriedok medzi používateľom a samotného snímača (Obr. 1), ktorý je pripojený k počítaču cez univerzálny sériový port. Program nie je obmedzený iba na snímanie prostredia, ale je možné vytvárať nové projekty, ukladať ich pod jedinečnými názvami, prípadne vykresliť mapy prostredí uložených projektov.



Obr. 1 Snímač prostredia

Uchytenie všetkých komponentov k plechovej doske je zrealizované pomocou skrutiek, čo uľahčuje demontáž tohto zariadenia. Po jeho následnom skladaní je potrebné dodržať správne pripojenie konektora servo motorčeka a snímača vzdialeností k mikrokontroléru arduino.

Farebné označenie vodičov servo motorčeka a ich pripojenie k mikrokontroléru arduino:

- Hnedý - pin GND.
- Červený - pin 5V.
- Oranžový - pin 9

Farebné označenie vodičov snímača vzdialeností a ich pripojenie k mikrokontroléru arduino:

- Biely - pin 4.
- Zelený - pin 5.
- Oranžový - pin 6.
- Modrý - pin 7.

2. Popis programu

Celkové riešenie je zhotovené pomocou dvoch rôznych programov. Používateľská aplikácia je vytvorená v objektovo orientovanom jazyku Java a predstavuje rozhranie medzi systémom a používateľom. V tejto aplikácie sa vykresľuje výsledná mapa a používateľ má možnosť prispôsobovať vykreslenie tejto mapy svojím požiadavkám, ako aj ukladať a načítavať uložené projekty. Druhá časť slúži na ovládanie mikrokontroléra Arduino a je vytvorená v jazyku C. Po preložení zdrojových kódov spolu s potrebnými knižnicami je program spustiteľný na všetkých verziách operačného systému Windows. V prípade, že používateľ nechce skenovať nové prostredie, ale chce iba vykresliť prostredie snímané v minulosti, ktorého parametre má uložené, nie je potrebné mať snímač pripojený k počítaču.

2.1. Popis riešenia

Grafická stránka používateľskej aplikácie je zrealizovaná pomocou nástrojov *Swing a AWT*, ktoré sú obsiahnuté v Jave samotnej a nie je potrebné ich nijak vkladať do projektu. Riadenie mikrokontroléra Arduino a jeho komponentov je zabezpečené pomocou knižničných funkcií určených pre mikroprocesor Atmel ATMEGA, ktoré sú súčasťou balíka spolu s aplikáciou od spoločnosti Arduino, určenej pre rýchly preklad zdrojových textov a ukladanie preloženého programu do pamäte tohto mikrokontroléra. Komunikácie medzi arduinom a počítačom je zabezpečená cez sériový port. K tomu bolo potrebné použiť knižnicu RXTX pre vytvorenie komunikácie v Java aplikácii. Celá komunikácia je synchronizovaná, aby sa nevyskytli prípady čítania ešte nezapísaných dát alebo zapisovanie skôr, ako boli dáta prečítané. Synchronizácia je riešená posielaním odpoveďových rámcov.

2.2. Proces ukladania informácií do mriežky

Celý proces je rozdelený do štyroch častí príslušných pre každý kvadrant. V tomto riešení sú ale použité iba dve, pretože sa snímač dokáže otáčať len o 180°. Na začiatku sa inicializujú premenné potrebné pre výpočty. Cyklus prechádza mriežku po x-ovej osi, čiže je potrebné vypočítať šírku danej úsečky - rozdiel x-ových súradníc koncových bodov úsečky. Začiatok každej úsečky sa nachádza v počiatku súradnicového systému, čiže šírka úsečky je totožná s krajnou x-ovou súradnicou. Následne sa v cykle prechádza mriežka po x-ovej osi od začiatočnej x-ovej súradnice po koncovú. Pre každú hodnotu x-ovej osi sa zistí počet a pozícia bodov, ležiacich na y-ovej osi a tieto sa označia ako "voľné". Po dosiahnutí konca úsečky sa daný bod označí hodnotou "prekážka". Tento proces sa opakuje pre každú úsečku, ktorej dĺžka sa získala zo snímača vzdialenosti.

Popis ukladania informácií do mriežky je znázornený nasledujúcim pseudokódom:

x1, y1 - súradnice začiatočného bodu

x2, y2 - súradnice koncového bodu

```
BEGIN
 Δy := (y2 - y1) / x2
 yd := 0
 yh := 0 + Δy
 x := 0
 WHILE x < x2 DO
  BEGIN
   WHILE yd <= yh DO
    BEGIN
     pole[yd][x] := VOĽNÉ
     yd := yd + 1
    END
   yh := yh + Δy
   yd := yh - Δy
   x := x + 1
   END
  pole[yh][x] := PREKÁŽKA
 END
END
```

Kde Δy je počet bodov ležiacich na y-ovej osi prislúchajúcich jednému bodu na x-ovej osi, yd je aktuálny spodný bod na y-ovej osi a yh je aktuálny horný bod na y-ovej osi.

2.3. Popis tried a údajových štruktúr balíka arduino

V tejto kapitole sú popísané všetky triedy a údajové štruktúry balíka arduino.

2.3.1. Trieda Arduino

Všetky implementované rozhrania:

java.lang.Runnable

public class Arduino extends java.lang.Thread

popis triedy: Slúži na riadenie komunikácie s Arduino

Sumarizácia metód:

public synchronized void writeData(java.lang.String data)

public int byteArrayToInt(byte[] b)

public void readData()

Premenné:

private InputStream input

- smerník na objekt typu InputStream

private OutputStream output

- smerník na objekt typu OutputStream

private final String fileName

- názov súboru, ktorý obsahuje informácie o parametroch prostredia

private final JFrame myjframe

- smerník na objekt hlavného okna aplikácie

Detail konštruktora:

public Arduino (java.lang.String fileName, javax.swing.JFrame jframe)

popis: konštruktor pre vytvorenie objektu Arduino

parametre:

fileName - názov súboru, ktorý obsahuje informácie o parametroch prostredia

jframe - smerník na objekt hlavného okna aplikácie

Detail metód:

public void writeData(java.lang.String data)

popis: metóda na zapisovanie údajov do sériového portu

parametre:

data - reťazec znakov na zapísanie

public int byteArrayToInt(byte[] b)

popis: metóda na prevedenie jednotlivých bytov čísla na celé číslo (BCD -> int)

parametre:

b - BCD kód čísla

návrat: číslo vo formáte integer

public void readData()

popis: metóda na načítavanie dát zo sériového portu

2.3.2. Trieda Map

Všetky implementované rozhrania:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants

public class Map extends javax.swing.JFrame
popis triedy: Trieda na vytvorenie mapy zo súboru, ktorý obsahuje parametre prostredia

Sumarizácia metód:

public Tile[][] getMapField()

public int getHeight()

public int getWidth()

public void processLine(int length, int angle)

public int createMapFromFile()

Premenné:

private final Tile[][] pole

- pravdepodobnostná mriežka obsahujúca informácie o vytvorenej mape

private int width

- šírka mriežky

private int height

- výška mriežky

private final int centerX

- X pozícia robota

private final String fileLocation

- cesta k súboru, v ktorom sa nachádzajú informácie o parametroch prostredia

Detail konštruktora:

public Map(java.lang.String fileLocation)
popis: konštruktor pre vytvorenie objektu Map
parametre:
fileLocation - cesta k súboru, v ktorom sa nachádzajú informácie o parametroch prostredia

Detail metód:

public Tile[][] getMapField()

popis: získanie mriežky

návrat: dvojrozmerné pole charakterizujúce pravdepodobnostnú mriežku

public int getHeight()

popis: získanie výšky mriežky

návrat: výška mriežky

public int getWidth()

popis: získanie šírky mriežky

návrat: šírka mriežky

public void processLine(int length, int angle)

popis: spracovanie jedného údaju zo súboru. Na základe modifikovaného Bresenhamovho algoritmu sa do mriežky zapíšu informácie o nasnímanej prekážke

public int createMapFromFile()

popis: spracovanie obsahu súboru s parametrami prostredia

návrat: 0 ak sa podarilo spracovať dáta, ináč informácia o chybe

2.3.3. Trieda SerialClass

Všetky implementované rozhrania:

gnu.io.SerialPortEventListener, java.util.EventListener

public class SerialClass extends java.lang.Object implements gnu.io.SerialPortEventListener
popis triedy: Vytvorenie a riadenie komunikácie cez sériový port

Sumarizácia metód:

public InputStream getInputStream()

public OutputStream getOutputStream()

public void initialize()

public synchronized void close()

public synchronized void writeData(java.lang.String data)

Premenné:

private final String PORT_NAMES[]

- možnosti názvov portov pre rôzne operačné systémy

private InputStream input

- objekt vstupného prúdu

private OutputStream output

- objekt výstupného prúdu

Detail metód:

public InputStream getInputStream()

popis: získanie vstupného prúdu

návrat: referencia na vstupný prúd

public OutputStream getOutputStream()

popis: získanie výstupného prúdu

návrat: referencia na výstupný prúd

public void initialize()

popis: inicializácia komunikácie

public synchronized void close()

popis: ukončenie komunikácie; uzatvorenie komunikačného kanála

public synchronized void writeData(java.lang.String data)

popis: zápis dát do sériovej linky

parametre:

data - reťazec údajov, ktoré sa majú zapísať

2.3.4. Enumeračný typ Tile

public enum Tile

popis triedy: enumeračný typ slúžiaci na charakterizovanie obsahu mriežky

Enumeračné konštanty a ich popis:

VOLNE

- voľné miesto v mriežke

PREKAZKA

- miesto prekážky

NEPRESKUMANE

- nepreskúmaná časť priestoru

ROBOTPOS

- pozícia robota

2.4. Popis tried a údajových štruktúr balíka frames

V tejto kapitole sú popísané všetky triedy a údajové štruktúry balíka frames.

2.4.1. Enumeračný typ ColorEnum

public enum ColorEnum

popis triedy: enumeračný typ obsahujúci prvky slúžiace na uchovávanie aktuálnych farieb vykresľovania

Enumeračné konštanty a ich popis:

BARRIERS(Color.RED)

- prvok uchovávajúci farbu vykresľovania prekážok. Predvolená farba - červená

FREE(Color.WHITE)

- prvok uchovávajúci farbu vykresľovania voľných častí. Predvolená farba - biela

UNEXPLORED(Color.WHITE)

- prvok uchovávajúci farbu vykresľovania nepreskúmaných časti. Predvolená farba - biela

SENSOR(Color.BLACK)

- prvok uchovávajúci farbu vykresľovania pozície snímača. Predvolená farba - biela

2.4.2. Trieda ColorFrame

public class ColorFrame extends javax.swing.JFrame
popis triedy: Okno výberu farby

Premenné:

private JPanel jpanel

- referencia na hlavné okno

Detail konštruktora:

ColorFrame(javax.swing.JPanel jpanel)

popis: vytvorenie nového okna pre možnosť výberu farby

parametre:

jpanel - referencia na objekt hlavného okna

2.4.3. Trieda MainFrame

Všetky implementované rozhrania:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants

public class MainFrame extends javax.swing.JFrame
popis triedy: hlavné okno programu

Sumarizácia metód:

public void createNewMap()

public boolean getBariestOn()

public boolean getFreeOn()

public boolean getUnexploredOn()

public static void main(String args[])

Premenné:

private String openLocation = ""

- cesta k otvorenému súboru. Predvolená hodnota - žiadna cesta

private final String defaultFile = "file.txt"

- názov súboru po novom naskenovaní priestoru

private boolean bariestOn = true

- majú/nemajú sa vykresľovať prekážky

private boolean freeOn = false

- majú/nemajú sa vykresľovať voľné časti

private boolean unexploredOn = false

- majú/nemajú sa vykresľovať nepreskúmané časti

Detail konštruktora:

public MainFrame()

popis: vytvorenie základného okna a všetkých jeho komponentov

Detail metód:

public void createNewMap()

popis: vytvorenie a vykreslenie novej mapy

public boolean getBariestOn()

popis: zistenie, či sa majú/nemajú vykresľovať prekážky

návrat: true, ak sa majú, false ak sa nemajú

public boolean getFreeOn()

popis: zistenie, či sa majú/nemajú vykresľovať voľné miesta

návrat: true, ak sa majú, false ak sa nemajú

public boolean getUnexploredOn()

popis: zistenie, či sa majú/nemajú vykresľovať nepreskúmané miesta

návrat: true, ak sa majú, false ak sa nemajú

public static void main(String args[])

popis: hlavná metóda celého programu, volaná po spustení programu

parametre: parametre príkazového riadku

2.4.4. Trieda MapFrame

Všetky implementované rozhrania:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants

public final class MapFrame extends javax.swing.JFrame
popis triedy: trieda na vykreslenie mapy v celoobrazovkovom režime

Sumarizácia metód:

public void createImage()

Premenné:

private final Tile[][] pole

- pole reprezentujúce mriežku mapy

private final int width

- šírka mriežky

private final int height

- výška mriežky

private final Map map

- referencia na spracované údaje zo súboru uložených parametrov prostredia. Vytvorená mapa v mriežke.

Detail konštruktora:

public MapFrame(java.lang.String fileLocation)

popis: vytvorenie nového obrazu mapy

parametre:

fileLocation - reťazec charakterizujúci cestu k súboru, v ktorom sú uložené parametre prostredia, ktoré sa má vykresliť

Detail metód:

public void createImage()

popis: vytvorenie nového okna na vykresľovanie mapy

2.4.5. Trieda MessageBox

Všetky implementované rozhrania:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants

public class MessageBox extends javax.swing.JFrame
popis triedy: trieda slúžiaca na vytvorenie okna obsahujúceho správy pre používateľa

Detail konštruktora:

public MessageBox(java.lang.String text)

popis: vytvorenie nového message box-u

parametre:

text - správa, ktorá sa má v okne vypísať

2.4.6. Trieda MyJPanel

Všetky implementované rozhrania:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.accessibility.Accessible

public class MyJPanel extends javax.swing.JPanel
popis triedy: trieda na vytvorenie objektu plochy, v ktorej sa bude vykresľovať mapa v hlavnom okne programu

Sumarizácia metód:

public void render()

Premenné:

private final Tile[][] pole

- pole reprezentujúce mriežku mapy

private final int width

- šírka mriežky

private final int height

- výška mriežky

private final Map map

- referencia na spracované údaje zo súboru parametrov prostredia. Vytvorená mapa v mriežke.

private MainFrame JF

- referencia na hlavné okno aplikácie

Detail konštruktora:

public MyJPanel(java.lang.String fileLocation, MainFrame JF, javax.swing.JScrollPane jsp)
popis: konštruktor pre vytvorenie plochy pre vykresľovanie
parametre:
fileLocation - cesta k súboru s parametrami prostredia
JF - referencia na hlavné okno aplikácie
jsp - referencia na scroll pane, v ktorom je tento objekt umiestnený

Detail metód:

public void render()
popis: prekreslenie aktuálneho stavu okna

2.4.7. Trieda ProgressBar

Všetky implementované rozhrania:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants

public class ProgressBar extends javax.swing.JFrame
popis triedy: vytvorenie okna, v ktorom sa zobrazuje stav procesu skenovania

Sumarizácia metód:

public void setPercentage(int number)

Detail konštruktora:

public ProgressBar()

popis: vytvorenie nového okna

Detail metód:

public void setPercentage(int number)

popis: nastavenie zobrazených percent v stavovom riadku
parametre:
number - číslo percenta, ktoré sa má zobraziť

2.5. Popis programu mikrokontroléra arduino

Popis programu:

Program je napísaný v jazyku C++ a slúži na ovládanie hardvérových komponentov systému. Pre ovládanie servo motorčeka je použitá knižnica Servo.h. Proces začína nastavením servo motorčeka na začiatočné natočenie (0°) a po každom meraní sa pootočí o 1°. Každé meranie sa skladá z 10 meraní, z ktorých sa výsledná hodnota získa aritmetickým priemerom týchto meraní. Dané meranie sa spustí, ak je na vstupe prijatá hodnota 0.

Sumarizácia funkcií:

void setup()

void loop()

Premenné:

Servo myservo

- referencia na objekt servo motorčeka

int vzdialenost

- odmeraná vzdialenosť prekážky

int recv = 0

- prijatá hodnota znaku zo vstupu

int pos = 0

- pozícia natočenia servo motorčeka

Detail funkcií:

void setup()

popis: funkcia, v ktorej na nastavia všetky potrebné piny arduina na vstup/výstup

void loop()

popis: funkcia, v ktorej sa vykonáva hlavná slučka programu