

Impact Factor:

ISRA (India) = 1.344	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	ПИИИ (Russia) = 0.234	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 3.860	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 2.031	

SOI: [1.1/TAS](#) DOI: [10.15863/TAS](#)

International Scientific Journal Theoretical & Applied Science

p-ISSN: 2308-4944 (print) e-ISSN: 2409-0085 (online)

Year: 2017 Issue: 05 Volume: 49

Published: 5.05.2017 <http://T-Science.org>

Ular Abenov

Master student

Taraz State University after M.Kh.Dulaty

ular.abenov@mail.ru

Alexandr Shevtsov

candidate of technical sciences, member of PILA (USA),

member of European Academy of Natural History (UK),

member of Federation of Robotics Kazakhstan,

Department of «Mathematics», Deputy Director on

Science of faculty of information technologies,

automation and telecommunications,

Taraz state University named after M.Kh. Dulaty

Shev_AlexXXXX@mail.ru

SECTION 2. Applied mathematics. Mathematical modeling.

THE PRACTICAL ASPECT OF USING GREEDY ALGORITHM

Abstract: Initial data for algorithm development are well-known algorithms for the problem of kommivoyazhera, methods of calculation way for Greedy algorithm, known as source data for the task. The justification for the research work is based on the need of commercialization of the algorithm and the creation of methods of calculation capable of implementing the algorithm with the possibility of implementation.

Key words: introduction, greedy algorithm, Delphi.

Language: Russian

Citation: Abenov U, Shevtsov A (2017) THE PRACTICAL ASPECT OF USING GREEDY ALGORITHM. ISJ Theoretical & Applied Science, 05 (49): 120-127.

Soi: <http://s-o-i.org/1.1/TAS-05-49-18> **Doi:**  <https://dx.doi.org/10.15863/TAS.2017.05.49.18>

ПРАКТИЧЕСКИЙ АСПЕКТ ИСПОЛЬЗОВАНИЯ ЖАДНОГО АЛГОРИТМА

Аннотация: Исходными данными для разработки алгоритмов являются известные алгоритмы задачи коммивояжера, методы расчета пути для «Жадного алгоритма», известные исходные данные для задачи. Обоснование необходимости проведения научно-исследовательской работы опирается на необходимость коммерциализации алгоритма и создание методов расчета, способных реализовать алгоритм с возможностью внедрения.

Ключевые слова: внедрение, жадный алгоритм, Delphi.

Introduction

Актуальность проблемы.

При решении различных технологических задач требуется найти оптимальный путь решения обусловленный малым временем выполнения задачи. Поэтому задача расчета оптимального пути является актуальной для промышленности.

Сведения о планируемом научно-техническом уровне разработки.

Теоретические исследования и математические модели позволяют прогнозировать оптимальный путь с минимальным временем. Результаты исследования могут быть направлены на автоматизацию процессов и сокращение сроков и средств затрачиваемых на их выполнение.

Целью работы является разработка алгоритмов и программ для Delphi и Maple для поиска оптимального пути и возможности их коммерциализации.

Для достижения этой цели в работе решались **следующие задачи:**

- разработка программы для Delphi и ее апробация для различных начальных данных;
- изучение возможности коммерциализации данного алгоритма в производстве.

Научная новизна исследований

- разработаны программы на Delphi с возможностью их коммерциализации.

Практическая ценность

- разработанные алгоритмы и программное обеспечение позволяет находить оптимальный путь. Результаты имеют возможность коммерциализации.

Обоснованность и достоверность полученных результатов -подтверждается тем, что они были получены на основе известных теоретических и апробированных методов дискретной математики.



Impact Factor:

ISRA (India) = 1.344
 ISI (Dubai, UAE) = 0.829
 GIF (Australia) = 0.564
 JIF = 1.500

SIS (USA) = 0.912
 ПИИЦ (Russia) = 0.234
 ESJI (KZ) = 3.860
 SJIF (Morocco) = 2.031

ICV (Poland) = 6.630
 PIF (India) = 1.940
 IBI (India) = 4.260

Materials and Methods Практический аспект использования

выбрать второй пункт, загрузить карту, и ввести все координаты всех городов.

Апробируем данный алгоритм на 16 городах Казахстана. Для этого необходимо

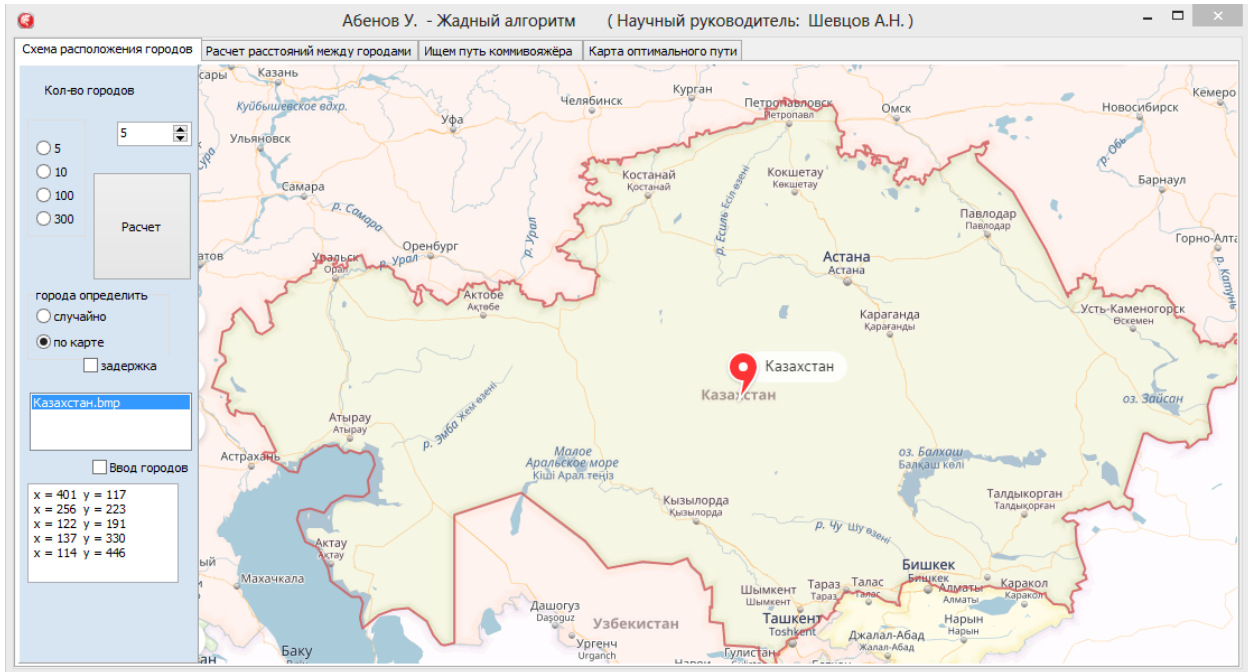


Рисунок 1 – Ввод карты и координат городов.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	-1	59,66	131,86	255,98	260	288,1	274,01	294,59	252,49	95,77	154,65	194,33	324,21	457,01	462,15	528,41
2	59,66	-1	123,74	215,11	200,41	230,82	231,65	257,76	240,43	154,77	211,62	249,06	361,44	497,51	488,18	543,17
3	131,86	123,74	-1	151,47	251,91	303,01	340,93	372,09	364,16	176,08	204,43	305,13	453,43	583,23	593,98	658,74
4	255,98	215,11	151,47	-1	194,85	257,76	347,5	386,35	424,72	323,31	355,86	446,59	576,1	711,3	701,59	748,62
5	260	200,41	251,91	194,85	-1	62,96	172,15	212,32	290,06	355,06	410,07	439,71	511,7	648,62	602,43	622,48
6	288,1	230,82	303,01	257,76	62,96	-1	125,67	163,95	260,27	383,73	442,24	452,68	502,3	636,76	577,9	582,06
7	274,01	231,65	340,93	347,5	172,15	125,67	-1	40,24	144,08	360,57	422,73	393,1	403,62	532,35	460,48	462,57
8	294,59	257,76	372,09	386,35	212,32	163,95	40,24	-1	122,91	376,06	438,16	395,24	387,75	512,19	432,5	422,91
9	252,49	240,43	364,16	424,72	290,06	260,27	144,08	122,91	-1	310,72	369,44	295,29	265,17	389,68	317,97	332,06
10	95,77	154,77	176,08	323,31	355,06	383,73	360,57	376,06	310,72	-1	62,2	132,13	298,78	418,71	452,52	536,45
11	154,65	211,62	204,43	355,86	410,07	442,24	422,73	438,16	369,44	62,2	-1	144,05	322,89	430,11	481,49	572,06
12	194,33	249,06	305,13	446,59	439,71	452,68	393,1	395,24	295,29	132,13	144,05	-1	179,61	288,64	339,21	436,57
13	324,21	361,44	453,43	576,1	511,7	502,3	403,62	387,75	265,17	298,78	322,89	179,61	-1	137,76	160,03	264,37
14	457,01	497,51	583,23	711,3	648,62	636,76	532,35	512,19	389,68	418,71	430,11	288,64	137,76	-1	139,8	255,12
15	462,15	488,18	593,98	701,59	602,43	577,9	460,48	432,5	317,97	452,52	481,49	339,21	160,03	139,8	-1	118,25
16	528,41	543,17	658,74	748,64	622,48	587,08	462,57	427,96	333,06	536,45	573,58	436,58	264,37	255,12	118,25	-1

Рисунок 2 – Расчет расстояний между городами.

Impact Factor:

ISRA (India) = 1.344	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	ПИИЦ (Russia) = 0.234	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 3.860	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 2.031	

Как видим применение задачи коммивояжера на практике очень обширно. Ее можно использовать для поиска кратчайшего маршрута при гастролях эстрадной группы по городам, нахождения последовательности технологических операций обеспечивающей наименьшее время выполнения всего производственного цикла и пр.:

- **Задача о путешественнике.** Коммивояжер должен объездить N городов. Для того чтобы уменьшить расходы, он хочет построить такой маршрут, чтобы объездить все города только по одному разу и вернуться в исходный с минимумом затрат. Затраты на перемещение между городами заданы матрицей.

- **Задача о производстве красок.** Имеется цех или производственная линия для производства n красок разного цвета; обозначим эти краски номерами $1, 2, \dots, n$. Всю производственную линию можно считать одним станком. Будем также считать, что одновременно станок производит только одну краску, поэтому краски нужно производить в некотором порядке. Поскольку производство циклическое, то краски надо производить в циклическом порядке. После окончания производства краски i и перед началом производства краски j надо отмыть оборудование от краски i . Для этого требуется время $C[i, j]$. Очевидно, что $C[i, j]$ зависит как от i , так и от j , и что, вообще говоря, $C[i, j] \neq C[j, i]$. При некотором выбранном порядке придется на цикл производства красок потратить время:

$$f = \sum_{(i,j) \in \pi} C_{i,j} + \sum_{k=1}^n t_k$$

где t_k - время производства k -ой краски (не считая очистки). Вторая сумма в правой части постоянна, поэтому полное время на цикл

производства минимизируется вместе с общим временем на очистку. Таким образом, задача коммивояжера и задача о минимизации времени очистки – это одна задача, только варианты ее описаны разными словами.

- **Задача о дыропробивном прессе.** Дыропробивной пресс производит большое число одинаковых металлических листов, в которых последовательно по одному пробиваются отверстия разной формы и величины. Схематически пресс можно представить в виде стола,двигающегося независимо по координатам x, y , и вращающегося над столом диска, по периметру которого расположены дыропробивные инструменты разной формы и величины. Каждый инструмент присутствует в одном экземпляре. Диск может вращаться одинаково в двух направлениях (по координате вращения z). Имеется собственно пресс, который надавливает на подвешенный под него инструмент тогда, когда под инструмент подведена нужная точка листа. Теперь, как и в предыдущем случае, задача составления оптимальной программы для дыропробивного пресса сводится к задаче коммивояжера.

В результате проведенной апробации и реализации двух программ на Maple и Delphi можно сделать вывод, что полученные данные свидетельствуют о корректной работе Жадного алгоритма при различных начальных условиях.

Апробация разработанных алгоритмов

Для апробации разработанных алгоритмов дискретной математики разработаем устройство состоящее из двух линейных приводов, двух драйверов и платы Ардуино уно.

Подключение к компьютеру будем осуществлять по USB кабелю.

Impact Factor:

ISRA (India)	= 1.344	SIS (USA)	= 0.912	ICV (Poland)	= 6.630
ISI (Dubai, UAE)	= 0.829	ПИИИ (Russia)	= 0.234	PIF (India)	= 1.940
GIF (Australia)	= 0.564	ESJI (KZ)	= 3.860	IBI (India)	= 4.260
JIF	= 1.500	SJIF (Morocco)	= 2.031		

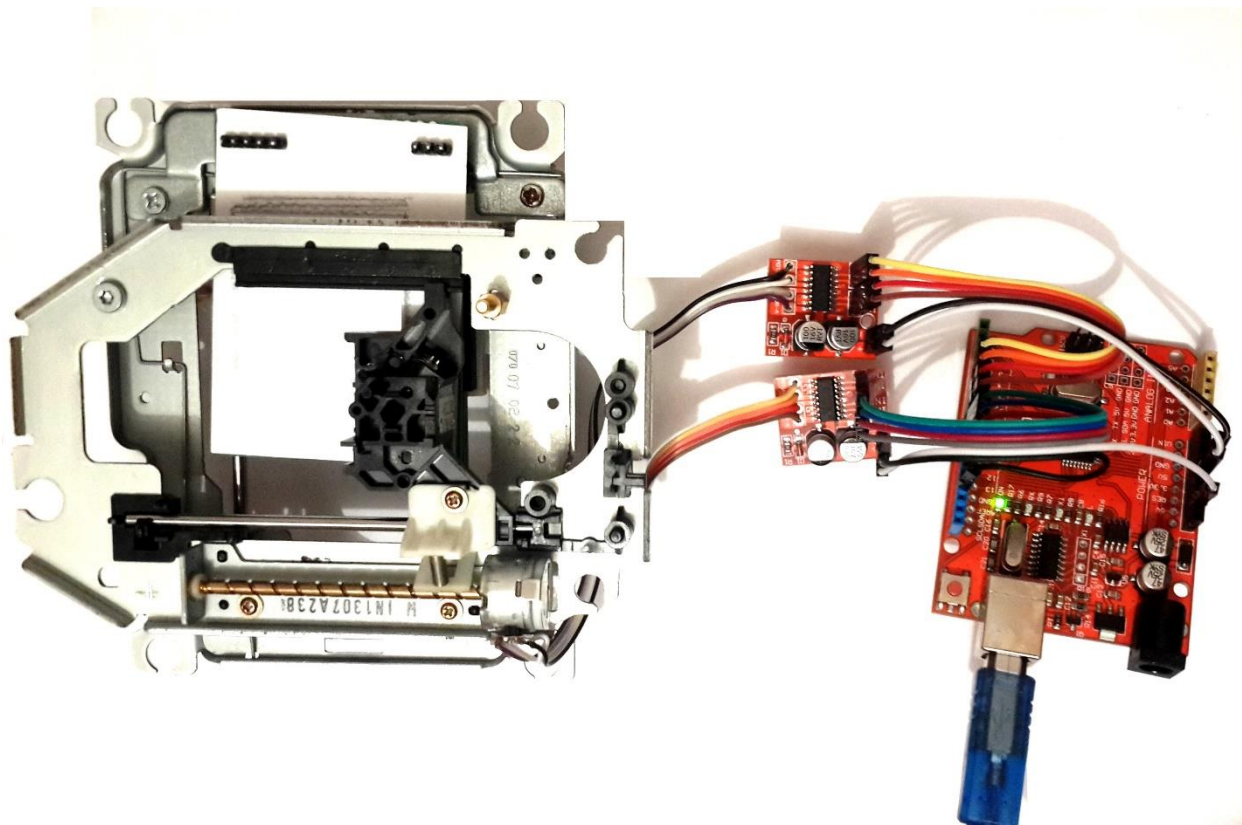


Рисунок 5 – Микроконтроллер и два линейных привода.

Разработаем программу для микроконтроллера:

```
#include <Stepper.h>
// change this to the number of steps on your motor
#define STEPS 100

Stepper s1(STEPS, 11, 10, 9, 8);
Stepper s2(STEPS, 7, 6, 5, 4);

int s1max = 220;
int s2max = 160;

int x,y,x0,y0 =0;
char incomingByte; // входящие данные

void setup() {
  Serial.begin(115200);
  // set the speed of the motor to 30 RPMs
  s1.setSpeed(80);
  s2.setSpeed(60);
  s1.step(-s1max);
  s2.step(-s2max);
}

void loop() {

  if (Serial.available() > 0) { //если пришли данные
    incomingByte = Serial.read(); // считываем байт
```

Impact Factor:

ISRA (India) = 1.344	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	ПИИЦ (Russia) = 0.234	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 3.860	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 2.031	

```
x:=x0+incomingByte;
incomingByte = Serial.read(); // считываем байт
y:=y0+incomingByte;

s1.step(x);
s2.step(y);
x0:=x;
y0:=y;
// delay(5000);
}
```

Дополним программу на Delphi возможностью подключения к микроконтроллеру:

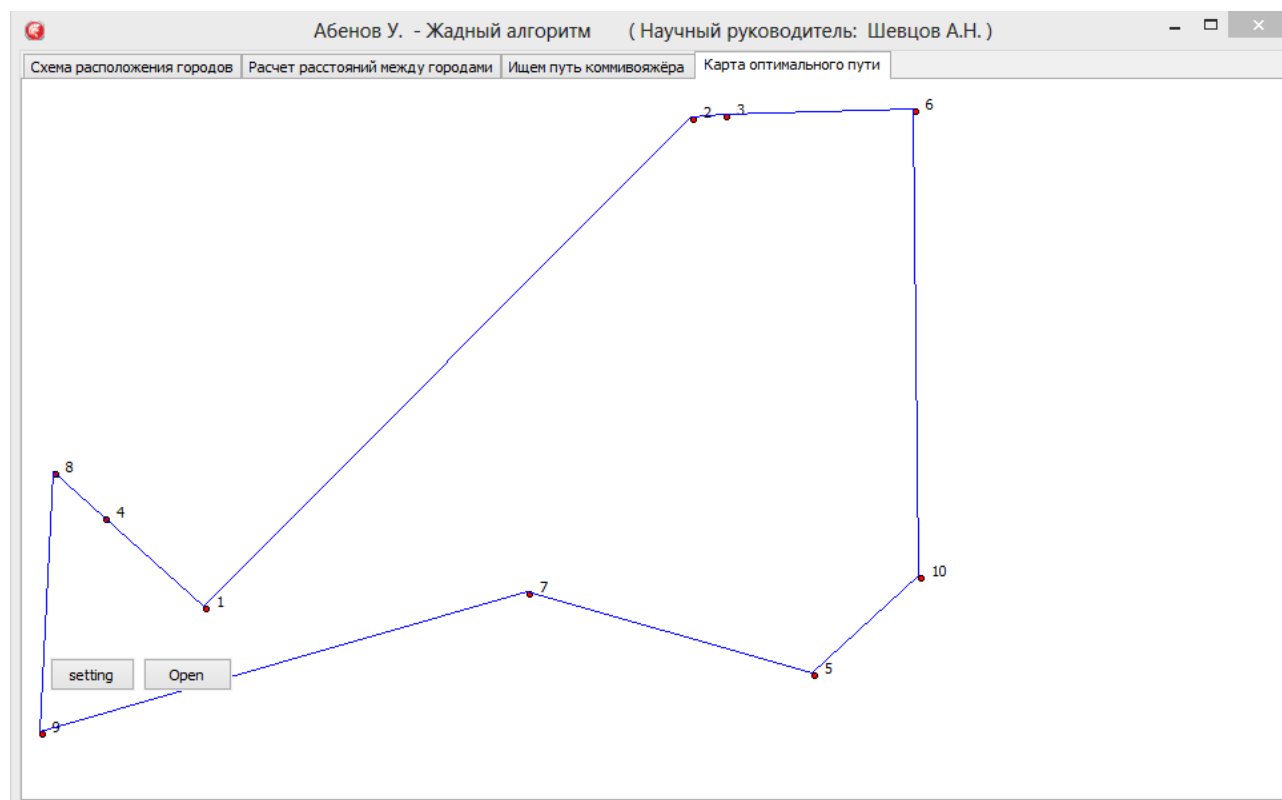


Рисунок 6 – Программа на Delphi с подключением к микроконтроллеру.

Impact Factor:

ISRA (India)	= 1.344	SIS (USA)	= 0.912	ICV (Poland)	= 6.630
ISI (Dubai, UAE)	= 0.829	ПИИЦ (Russia)	= 0.234	PIF (India)	= 1.940
GIF (Australia)	= 0.564	ESJI (KZ)	= 3.860	IBI (India)	= 4.260
JIF	= 1.500	SJIF (Morocco)	= 2.031		

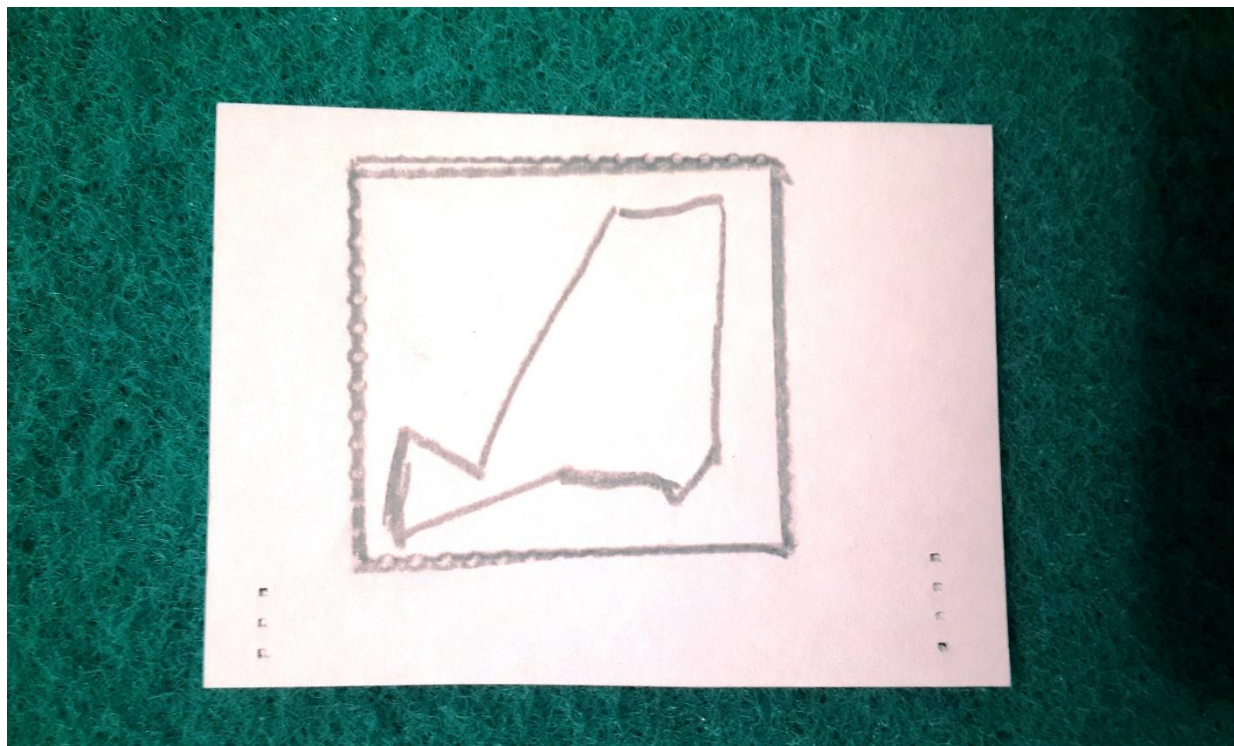


Рисунок 7 – Результат работы устройства.

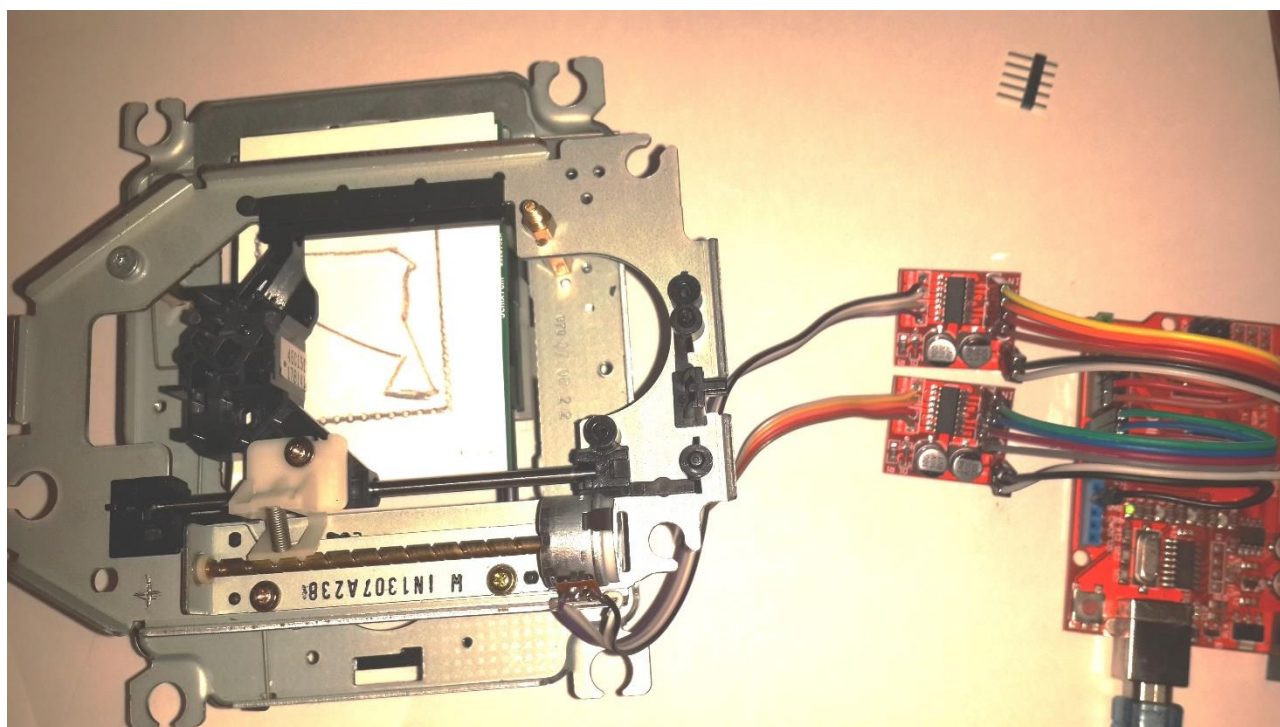


Рисунок 8 – Программа в действии.

Impact Factor:

ISRA (India) = 1.344	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	ПИИЦ (Russia) = 0.234	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 3.860	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 2.031	

Conclusion

В результате проведенного исследования были получены следующие результаты:

Программа на Delphi адаптирована к практическому использованию с картами и может использоваться для решения практических задач,

Имеется возможность коммерциализации результатов работы.

Разработано устройство на микроконтроллере реализующее жадный алгоритм.

References:

1. Tomas H. Cormen, Charl'z I. Lejzerson, Ronal'd L. Rivest, Klifford Shtajn (2006) *Algoritmy: postroenie i analiz = Introduction to Algorithms*. — 2-e izd. — M.: Vil'jams, 2006. — 1296 p. — ISBN 0-07-013151-1.
2. Sjelomon D (2004) *Szhatie dannyh, izobrazhenija i zvuka*. — M.: Tehnosfera, 2004. — 368 p. 3000 jekz. — ISBN 5-94836-027-X.
3. Levitin AV (2006) Glava 9. Zhadnye metody: Algoritm Haffmana // *Algoritmy. Vvedenie v razrabotku i analiz* — M.: Vil'jams, 2006. — p. 392–398. — 576 p. — ISBN 978-5-8459-0987-9
4. Markov AA (1982) *Vvedenie v teoriju kodirovanija*. — M.: Nauka, 1982. — 192 p.
5. Joseph B. Kruskal (1956) On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. // *Proc. AMS*. 1956. Vol 7, No. 1. p. 48–50
6. Jarník V (1930) O jistém problému minimálním [About a certain minimal problem], *Práce Moravské Přírodovědecké Společnosti*, 6, 1930, p. 57–63. (cheshsk.)
7. Prim RC (1957) Shortest connection networks and some generalizations. In: *Bell System Technical Journal*, 36, p. 1389–1401 (angl.)
8. Cheriton D, Tarjan RE (1976) Finding minimum spanning trees. In: *SIAM Journal on Computing*, 5 (Dec. 1976), p. 724–741 (angl.)
9. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein (2009) *Introduction to Algorithms, Third Edition*. MIT Press. ISBN 0-262-03384-4. Section 23.2: The algorithms of Kruskal and Prim, p. 631–638.
10. Rado R (1942) A theorem on independence relations. *Quart. J. Math.*, 13:83–89.
11. Edmonds J (1971) Matroids and the Greedy Algorithms // *Math Programming*. 1971. - p. 127-136. doi:10.1007/BF01584082
12. Novikov FA (2000) "Diskretnaja matematika dlja programmistov", - p. 74-77
13. Alekseev VE, Talanov VA (2006) *Grafy i algoritmy // Intuit*, - ISBN 5-9556-0066-3.
14. (2009) *Matroidy, Kurs "Diskretnaja matematika"*, Novosibirskij gosudarstvennyj universitet.
15. Cormen T, Lejzerson C, Rivest R, Shtajn K (2017) Glava 16. Zhadnye algoritmy
16. Cormen T, et al. (2000) *Algoritmy: postroenie i analiz*. — M.: MCNMO.
17. Alfjorova ZV (1973) *Teorija algoritmov*. — M.: Statistika.

