

Работа на тему: «Позиционирование объекта в закрытом пространстве»

Работу выполнил:

Савин Денис Васильевич,

Ученик 9 класса Харьковской гимназии №46

им. М. В. Ломоносова

Харьковского городского совета Харьковской области

Научный руководитель: Савенкова Ольга Сергеевна,
учитель информатики Харьковской гимназии №46,
"специалист Высшей категории"

Актуальность

- Использование систем позиционирования людей и материальных объектов - одно из актуальных направлений совершенствования технологических и бизнес процессов в самых различных областях деятельности.



Цель

- Исследовать существующие системы позиционирования объектов.
- Построить устройство для позиционирования объектов в закрытом пространстве.
- Написать программное обеспечение для функционирования системы позиционирования объектов в закрытом пространстве.

Научная новизна

Архитектура программно-аппаратного комплекса для получения координат объекта в замкнутом пространстве.

Практическая ценность

- В будущем это изобретение можно использовать для роботов-миноискателей в помещениях с большим количеством людей. например:
- Железнодорожные, аэро и автовокзалы, развлекательные центры, кинотеатры, торговые центры, больницы, что угодно, что может стать объектом террористического акта.

Существующие способы позиционирования

- радиочастотные технологии (точность 1 - 4 метра)
- спутниковые технологии навигации (GPS, ГЛОНАСС точность 10-15 метров, а также используется вне помещений)
- технологии локального позиционирования (инфракрасные и ультразвуковые) (точность 3-10 сантиметров)
- радиочастотные метки - RFID (1- 3 метра)

Аналог

- Ультразвуковой дальномер HC-SR04



- Данный дальномер имеет большую точность, то есть 3-10 сантиметров.

Изобретение

Мое изобретение состоит из микроконтроллеров, ультразвуковых датчиков, радиопередатчик и разработки схемы их расположения, а главное - программы, которая рассчитывает координаты объекта в зависимости от данных, поступающих от датчиков.

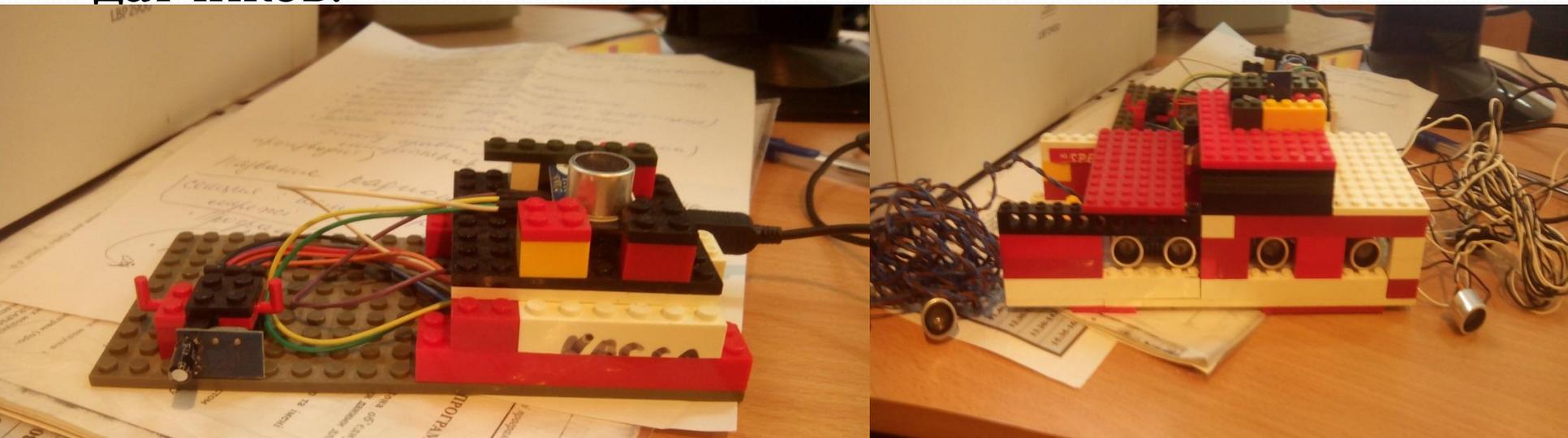
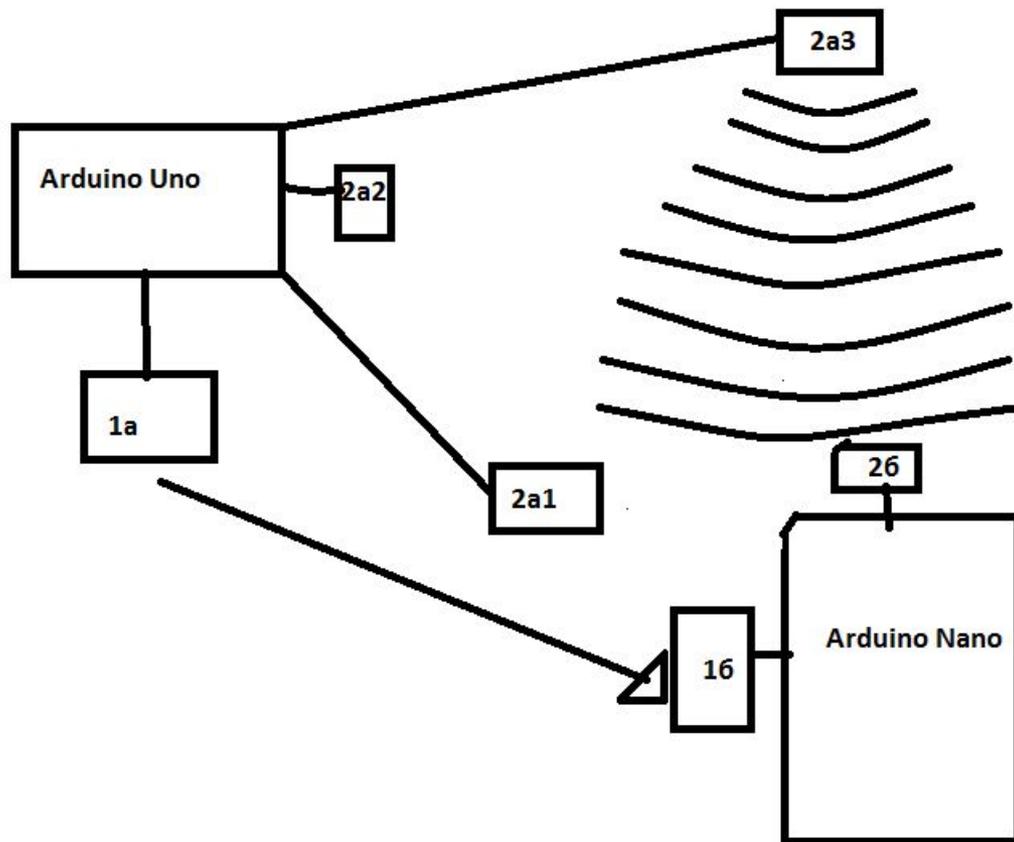


Схема работы прибора



1а - радио передатчик

1б - радиоприемник

2А1, 2А2, 2а3 - ультразвуковые датчики стационарного прибора

2б - ультразвуковые датчики пререносного прибора

Сравнение микроконтроллеров

	Arduino	Raspberry PI
Размер	7 x	6 см
Тактовая частота	16МГц	700 МГц
Питание	7 – 12 В	5 В
Многозадачность	-	+
Легкость программирования	+	-
Цена	4-9\$	35\$

Компоненты и их цена

В своей работе я использовал следующие
компоненты:

Arduino Nano - 4 \$

Arduino Uno - 6 \$

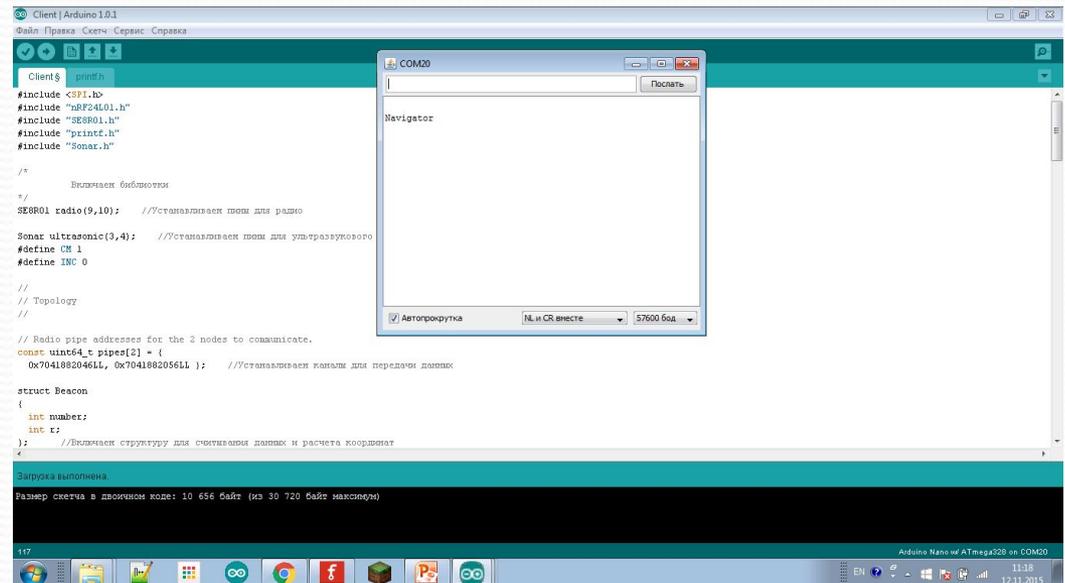
HC-SR04 (4 шт) - 6 \$

Nrf24l01 (2шт) - 2 \$

Всего: 18 \$

Оболочка и язык программирования

- Язык программирования Arduino является стандартным C++ (используется компилятор AVR-GCC) с некоторыми особенностями, облегчающие написание программы.



```
Client.ino
#include <SPI.h>
#include "nRF24L01.h"
#include "SE8R01.h"
#include "printf.h"
#include "Sonar.h"

/*
   Включает библиотеки
*/
SE8R01 radio(9,10); //Устанавливаем пины для радио
Sonar ultrasonic(3,4); //Устанавливаем пины для ультразвукового
#define CH 1
#define INC 0

//
// Topology
//

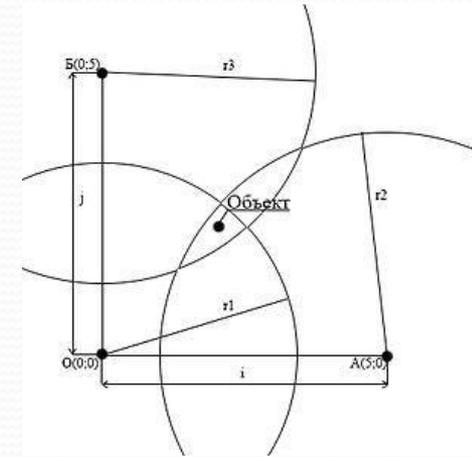
// Radio pipe addresses for the 2 nodes to communicate.
const uint64_t pipes[2] = {
  0x7041862046LL, 0x7041862056LL }; //Устанавливаем каналы для передачи данных

struct Beacon
{
  int number;
  int sz;
}; //Включает структуру для считывания данных и расчета координат
};

Загрузка выполнена
Размер скетча в двоичном коде: 10 656 байт (из 30 720 байт максимум)
```

Алгоритм нахождения координат

- Для определения положения объекта используется алгоритм трилатерации.



$$x = \frac{r_1^2 - r_2^2 + i^2}{2i} \quad y = \frac{r_1^2 - r_3^2 + j^2}{2j} \quad z = \pm \sqrt{r_1^2 - x^2 - y^2}$$

Модули программы переносного устройства

- Инициализация переменных
- Настройка радиоприемника
- Прием данных от радио передатчика
- Расчет координат и вывод их на экран

Модули программы стационарного прибора

- Инициализация переменных
- Настройка радио передатчика
- Передача данных к переносному прибору

Программа

```
#include <SPI.h>
#include "nRF24Lo1.h"
#include "SE8Ro1.h"
#include "printf.h"
#include "Sonar.h"
SE8Ro1 radio(9,10);
Sonar ultrasonic(3,4);
#define CM 1
#define INC 0
const uint64_t pipes[2] = {
    0x7041882046LL, 0x7041882056LL };
struct Beacon
{
    int number;
    int r;
};
Beacon beaono;
Beacon beacon1;
Beacon beacon2;
long timeOffset;
long got_time;
float x;
float y;
float z;
int i = 67;
int j = 86;
void setup(void)
{
    Serial.begin(57600);
    printf_begin();
    printf("\n\rNavigator\n\r");
    radio.begin();
    radio.openWritingPipe(pipes[1]);
    radio.openReadingPipe(1,pipes[0]);
    radio.setRetries(15, 15);
    radio.setChannel(120);
    radio.setPALevel(RF24_PA_MAX);
    radio.setDataRate( RF24_2MBPS );
    radio.setPayloadSize(sizeof(Beacon));
    radio.startListening();
}
```

```
void loop(void)
{
    measuring();
}

void measuring(void)
{
    struct Beacon *p;
    p = &beaono;
    if ( radio.available() )
    {
        bool done = false;
        while (!done)
        {done = radio.read( p , sizeof(Beacon) );
        if (beaono.number == 0){
            long dist_cm = ultrasonic.Ranging(CM);
            beaono.r = dist_cm;
        }
        else if (beaono.number == 1){
            long dist_cm = ultrasonic.Ranging(CM);
            beacon1.r = dist_cm;
        }
        else if (beaono.number == 2){
            long dist_cm = ultrasonic.Ranging(CM);
            beacon2.r = dist_cm;
        }
    }

    x = (sq(float(beaono.r)) - sq(float(beacon1.r)) + sq(float(i)))/(2*i);
    y = (sq(float(beaono.r)) - sq(float(beacon2.r)) + sq(float(j)))/(2*j);
    z = sqrt(sq(float(beaono.r))-sq(x)-sq(y));
    /**/int x=float (x);
    int y=float (y);
    int z=float (z);
    printf("X: %d Y: %d Z: %d ",x,y,z );
}
}
```