Министерство образования Республики Беларусь
Учреждение образования «Витебский государственный университет имени П.М. Машерова»
Кафедра инженерной физики

Е.А. Краснобаев, И.Ф. Мехов

ЛАБОРАТОРНЫЕ РАБОТЫ ПО КУРСУ «ТЕОРЕТИЧЕСКИЕ ОСНОВЫ РОБОТОТЕХНИКИ»

Методические рекомендации

Витебск
ВГУ имени П.М. Машерова
2013

УДК 004.896(075.8)
ББК 32.816.2я73
К78

Печатается по решению научно-методического совета учреждения образования «Витебский государственный университет имени П.М. Машерова». Протокол № 9 от 20.06.2013 г.

Авторы: заведующий кафедрой инженерной физики ВГУ имени П.М. Машерова **Е.А. Краснобаев;** студент 5-го курса физического факультета ВГУ имени П.М. Машерова **И.Ф. Мехов**

Р е ц е н з е н т :
заведующий кафедрой теоретической физики ВГУ
имени П.М. Машерова, доктор физико-математических наук,
профессор *Ю.В. Трубников*

Краснобаев, Е.А.

К78 Лабораторные работы по курсу «Теоретические основы робототехники» : методические рекомендации / Е.А. Краснобаев, И.Ф. Мехов. – Витебск : ВГУ имени П.М. Машерова, 2013. – 22 с.

Курс лабораторных работ по дисциплине специализации «Теоретические основы робототехники» предназначен для студентов специальности 1-31 04 01-02 «Физика» (производственная деятельность) специализации 1-31 04 01-02-16 «Компьютерное моделирование физических процессов» и составлен в соответствии с учебной программой указанного курса.

УДК 004.896(075.8)
ББК 32.816.2я73

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Робототехника – область науки и техники, ориентированная на создание роботов и робототехнических систем, предназначенных для автоматизации сложных технологических процессов и операций, в том числе, выполняемых в недетерминированных условиях, для замены человека при выполнении тяжелых, утомительных и опасных работ.

В настоящее время робототехника вышла за рамки промышленного применения и сейчас представляет собой значительно более обширную область науки, чем можно было себе представить ранее.

Робототехнические комплексы стали популярны и в области образования как современные высокотехнологичные исследовательские инструменты в области теории автоматического управления и мехатроники. Их использование в различных учебных заведениях среднего и высшего профессионального образования позволяет реализовывать концепцию проектного обучения, что позволяет развивать исследовательские умения и системное мышление у студентов.

Применение возможностей робототехнических комплексов в инженерном образовании даёт возможность одновременной отработки профессиональных навыков сразу по нескольким смежным дисциплинам: механика, теория управления, схемотехника, программирование, теория информации, искусственный интеллект.

Учебный план специальности 1-31 04 01-02 Физика (производственная деятельность) на физическом факультете Витебского государственного университета имени П.М. Машерова включает дисциплину специализации «Теоретические основы робототехники». Целью преподавания дисциплины является обучение основным теоретическим положениям, понятиям, методам и практической работе с робототехническими системами.

В данных методических рекомендациях изложен курс лабораторных работ по указанной дисциплине, на базе учебного мобильного робота DAGU ASURO Robot kit.

1. ОПИСАНИЕ РОБОТА ASURO ROBOT KIT

DAGU ASURO Robot kit - мобильный робот, разработанный в образовательных целях для школ и университетов.

Робот, представляет собой набор из процессора Atmel AVR RISC, двух моторов, шести детекторов касания, фотодатчиков, светодиодов, индикаторов и инфракрасного канала.

Робот поставляется с прошитой во флэш-память программой AVR-Bootloader. После подачи питания загрузчик пытается принять с компьютера данные, а если инфракрасный канал сигнал не обнаружен, через две секунды робот начинает исполнять текущую программу. Два фототранзистора позволяют отслеживать цвет поверхности, по которой перемещается робот. Две фотоячейки считают обороты колёс. В комплект входит CD-диск с компилятором Си, а также среда разработки. Внешний вид робота изображен на Рис. 1.1.

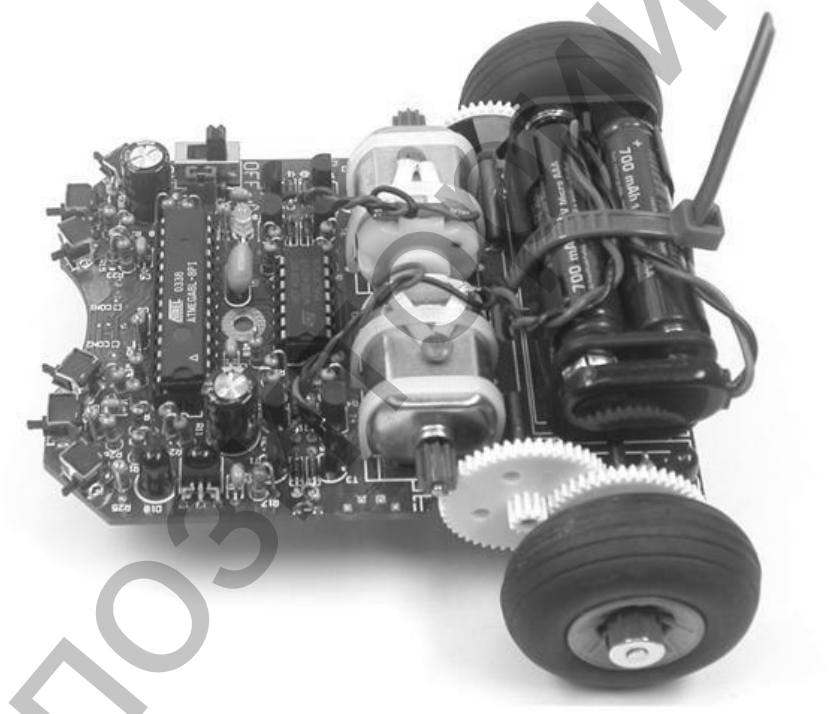

Рисунок 1.1. Готовый к работе ASURO ROBOT KIT
(вид сверху)

Электрическая схема робота изображена на Рис. 1.2. Описание основных элементов:

IC1: микроконтроллер Atmega8L-8PI.

IC3: микроконтроллер CD4081.

K1, K2, K3, K4, K5, K6: датчики столкновений.

D12: двухцветный светодиод, диаметр 3 мм, три ножки.

IC2: (SFH5110-36) инфракрасный приемник.

D10: (SFH 415-U) IR-LED 5мм, инфракрасный диод.

T11, T12: (LPT80A) фототранзистор, бесцветный.
D13, D14: (IRL80A) IR-LED, ИК-светодиод розового цвета.
D15, D16: светодиод 5 мм, красные.
S1: переключатель вкл/выкл.
T9, T10: (SFH300) фототранзистор 5 мм.
D11: светодиод 5 мм красный.

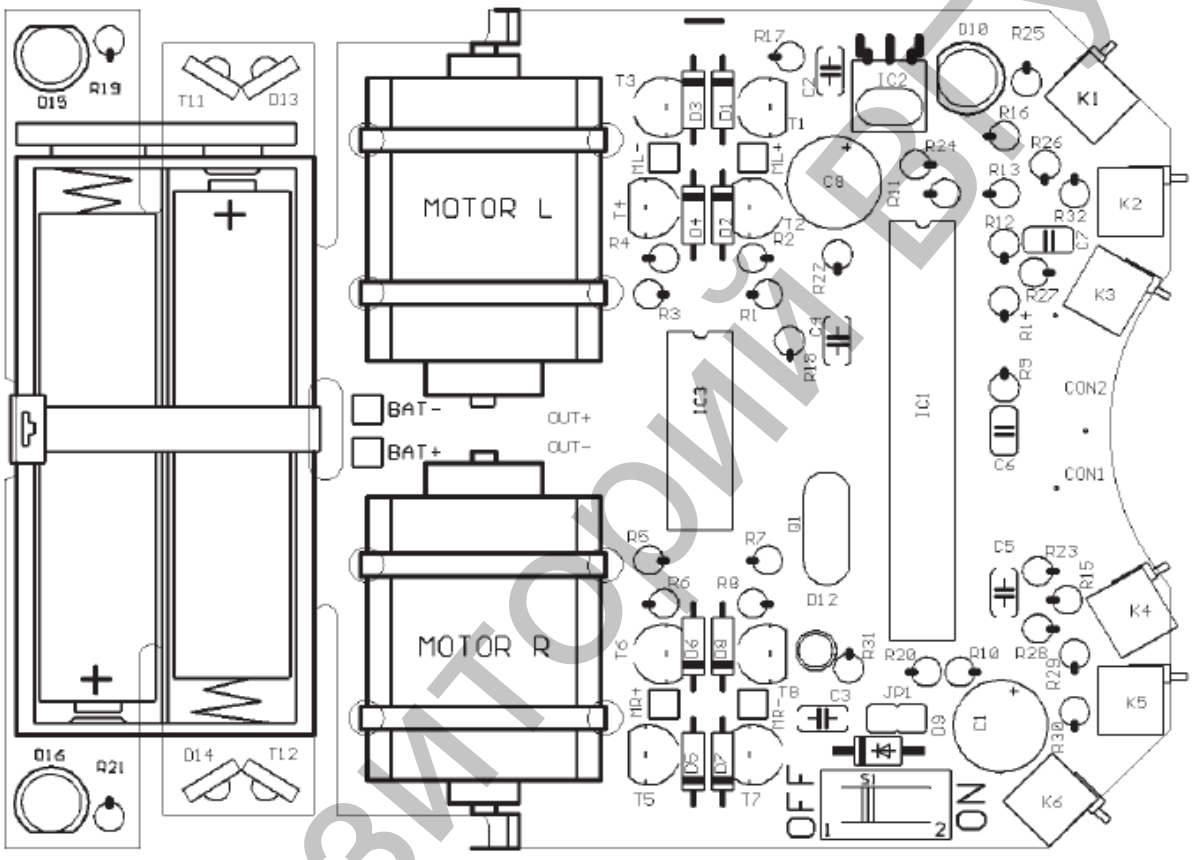

Рисунок 1.2. Электрическая схема ASURO

2. СОЗДАНИЕ, ЗАГРУЗКА И ВЫПОЛНЕНИЕ ПРОГРАММ

Для создания программ управления роботом ASURO, применяется следующее программное обеспечение:

1. Flash-Tool, программа для загрузки программы во флеш-память робота ASURO;
2. редактор программ Programmers Notepad и компилятор WinAVR;
3. драйвер для ИК-трансивера.

Программное обеспечение разработано и устанавливается на операционные системы Windows XP-8.

Программа Programmers Notepad представляет собой среду разработки, предназначенную для создания программ на языке Си. Главное окно программы изображено на Рис. 2.1.

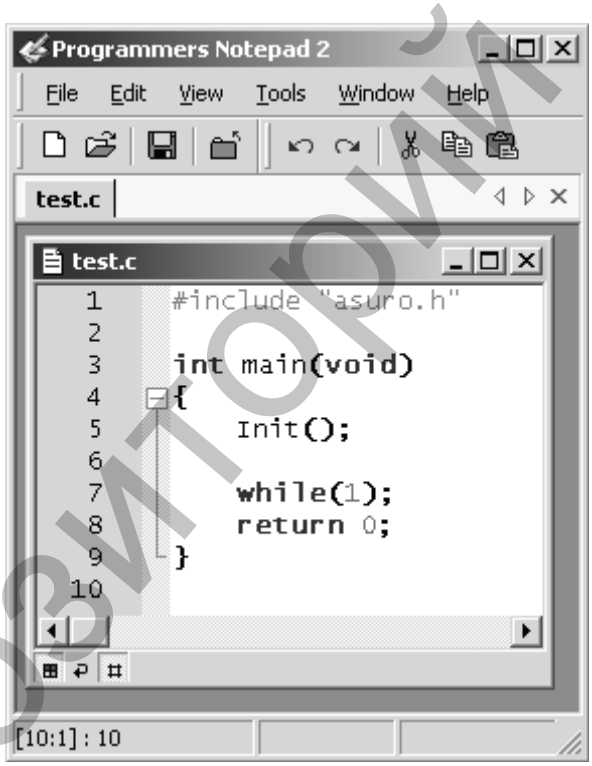

Рисунок 2.1. Главное окно программы Programmers Notepad

После того, как в текстовом редакторе составлена программа, необходимо выполнить ее компиляцию. Для этого, в главном меню Tools нужно выбрать команду **make**. При это компилятор выполнит отладку программы и укажет строки программы в которых присутствует ошибка. Если ошибок нет, то из файла исходного кода test.c создастся бинарный файл test.hex, который и будет загружен в память робота через ИК-трансивер с помощью программы Flash-tool.

В ходе компиляции генерируются также некоторые дополнительные файлы. Они необходимы лишь во время преобразования, после чего они будут бесполезны. Эти файлы могут быть удалены с помо-

щью команды **clean**, которую необходимо выполнять после каждой компиляции.

На следующем этапе необходимо загрузить полученный hex-файл в память робота через ИК-трансивер с помощью программы Flash-Tool (Рис.2.2). При этом драйвер ИК-трансивера должен быть предварительно установлен на компьютере.

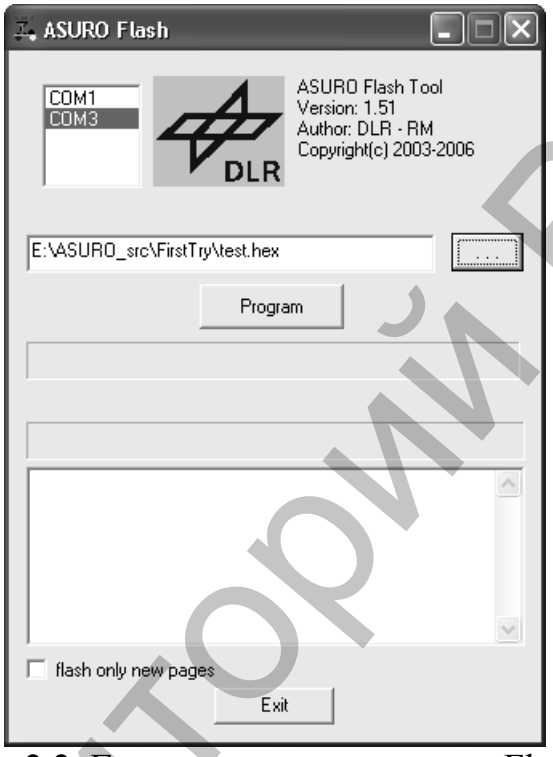

Рисунок 2.2. Главное окно программы Flash-Tool

После запуска программы Flash-Tool необходимо выполнить следующие указания, соблюдая строгую последовательность:

1. Выбрать необходимый порт;
2. Указать путь к hex-файлу на диске;
3. Нажать кнопку Program.
4. В течение 10 секунд включить робота и направить ИК-трансивер на инфракрасный диод D10. При включении робота в первую очередь будет выполняться соединения с компьютером. Если такое подключение отсутствует по каким-либо причинам, загрузка программы не произойдет.

После того как выполнится загрузка программы во флеш-память, необходимо выключить робота. При последующем его включении, загруженная программа начнет выполняться.

Важно! При загрузке следующей программы, даже если путь к файлу не изменился, выполнить пункт 2.

3. ЗАПУСК ПРОГРАММЫ САМОТЕСТИРОВАНИЯ

При первом, после сборки, включении робота, а также при загрузке программы SelfTest робот начнет выполнять команды самодиагностики. Если в ходе самодиагностики происходит ошибка, необходимо выключить робота и устранить ошибку.

1. Индикаторы

При запуске программы в первую очередь будет запущена самодиагностика индикаторов с интервалом в 3 секунды в следующем порядке:

- Status-LED (D12) зеленый (диод состояния)
- Status-LED (D12) красный
- Front-LED (D11) на нижней стороне платы (диод одометра)
- Back- LED (D15) (левый задний светодиод)
- Back- LED (D16) (правый задний светодиод)

2. Фототранзисторы (T9, T10)

После окончания проверки индикаторов, Status-LED (D12) должен загореться зеленым цветом. Фототранзисторы на обратной стороне ASURO ответственные за отслеживание линии, во время самодиагностики проверяются в течение, примерно, 10 секунд.

Если во время проверки фототранзисторы (T9, T10) будут освещены, светодиоды (D15, D16) будут гореть тускло. Если закрыть фототранзисторы от света, светодиоды (D15, D16) засветятся ярче.

3. Датчики касания

Далее, в течении 15 секунд будут проверены датчики касания. Необходимо поочередно нажать на каждый датчик и проверить отклик. Результат должен быть следующим:

K1 → Status-LED (D12) переключается на зеленый;
K2 → Status-LED (D12) переключается на красный;
K3 → загорится Front-LED (D11);
K4 → загорится Back-LED слева (D15);
K5 → загорится Back-LED справа (D16);
K6 → работает левый мотор.

4. Проверка одометра

По окончании предыдущего теста, светящийся индикатор Front-LED (D11) указывает на начало следующего теста, в котором будут проверены оба одометра, состоящих из светодиода и фототранзистора. Для выполнения теста необходимо вставить белый лист бумаги перед датчиками. Вначале проверяется левый датчик (T11), в случае успеха Status-LED (D12) загорится зеленым цветом. Затем, удерживая лист бумаги в передней части правого датчика (T12), необходимо проверить Status-LED (D12), который должен загореться красным. Удаление листа приведет к выключению Status-LED.

5. Проверка двигателей

Самодиагностика двигателей начинается с включения Back-LED (D15, D16). Левый двигатель запускается в прямом направлении с нулевой скорости до максимальной и обратно до нуля. Идёт смена направления и тест повторяется. Тот же тест проводится и на правом двигателе. После проверки одного двигателя будет проверяться работа двух двигателей одновременно по тому же сценарию.

6. Проверка ИК-порта

На этом завершающем этапе проверяется ИК-порт в результате которого последний передает и получает данные с компьютера. Тест начинается с периодического мигания Status-LED. Для того чтобы получить эти данные, ИК-трансивер должен быть подключен к компьютеру с запущенной программой «HyperTerminal». При получении символов, робот ответит символом, следующим в алфавите. Любой отправленный символ переключит Status-LED на противоположный цвет.

4. ОПИСАНИЕ ФУНКЦИЙ БИБЛИОТЕКИ ASURO

Для управления роботом разработчиками создана библиотека asuro.h, которая подключается с помощью директивы препроцессора include.

void Init(void)

Эта функция сброса микропроцессора в исходное состояние. Всегда должна быть выполнена в начале программы.

Пример:

```
#include “asuro.h”
int main(void) {
Init();
while(1); // бесконечный цикл
return 0; // никогда не будет выполнен
}
```

Обычно основная функция возвращает 0, отмечая конец программы. Однако в ASURO некоторые части старых программ могут по-прежнему остаться в памяти и выполняться снова. В результате чего могут быть получены странные эффекты. Для исключения таких ошибок каждая программа в конце должна быть зациклена.

void StatusLED (unsigned char color)

При вызове данной функции индикатор Status-LED (D12) будет отключен или включен. Допустимые значения параметра: OFF, зеленый, красный или желтый.

Пример, в результате выполнения которого Status-LED будет гореть желтым:

```
#include “asuro.h“
int main(void) {
Init();
StatusLED (YELLOW);
while(1);
return 0;
}
```

void FrontLED (unsigned char status)

При вызове данной функции индикатор Front-LED будет включаться или выключаться. Допустимые параметры ON и OFF соответственно.

Пример вызова:

```
FrontLED(ON);
```

void BackLED (unsigned char left, unsigned char right)

При вызове данной функции Back-LED (D15 и D16) будут включаться или выключаться. Первый параметр описывает состояние левого (D15), а второй параметр описывает состояние правого (D16) индикатора. Допустимые параметры: ON и OFF соответственно.

Пример вызова:

BackLED(OFF,ON);

void Sleep (unsigned char time)

Эта функция даёт команду процессору ожидать некоторое время. Период ожидания может быть определен через количество циклов таймера с частотой 72кГц.

Пример:

Процессор должен «спать» в течение примерно 3мс

$$\frac{0{,}003\text{с}}{\frac{1}{72\text{КГц}}} = 216$$

Отсюда следует вызов:

Sleep (216) ;

void MotorDir (unsigned char left_dir, unsigned char right_dir)

Эта функция управляет направлением обоих двигателей и должна быть вызвана перед вызовом функции включения скорости движения. Параметры FWD (вперед), RWD (назад), BREAK (тормоз) и FREE (свободное вращение).

Пример:

MotorDir(FWD,BREAK);

Левый двигатель должен двигаться вперед, в то время как правый двигатель остановлен.

void MotorSpeed (unsigned char left_speed,unsigned char right_speed)

Эта функция управляет скоростью двигателей. Максимальная скорость составляет 255. Двигатель начнет вращаться на значении около 60, в зависимости от механических условий. Значение параметра фактически контролирует мощность двигателя, и скорость вращения зависит от факторов, таких как трение или наклон.

Пример вызова:

MotorSpeed (255,0);

Левый мотор работает на максимальной скорости, а правый неподвижен.

void SerWrite (unsigned char *data, unsigned char length)

Эта функция выводит данные с робота через последовательный ИК-интерфейс. Первый параметр содержит ссылку на данные, подлежащие передаче, второй параметр описывает количество символов, которые будут отправлены.

Пример вызова:

SerWrite (“Hello how are you?”,18);

Строка «Привет, как дела?» будет передана по ИК-интерфейсу.

void SerRead(unsigned char *data, unsigned char length, unsigned int timeout)

Эта функция считывать данные через последовательный ИК-интерфейс. Первый параметр представляет собой указатель на адрес памяти, где необходимо сохранить сообщение. Второй параметр описывает количество символов, третий параметр описывает период тайм-аута. Тайм-аут используется для предотвращения бесконечного ожидания, если данных окажется меньше, чем ожидалось.

Пример. Чтение строки «GoAhead».

```
#include “asuro.h”
int main(void) {
char data[8]; // Выделение памяти
Init();
SerRead (data,8,0); // Чтение данных
MotorDir(FWD,FWD);
MotorSpeed(120,120);
while(1); // бесконечный цикл
return 0;
}
```

void LineData(unsigned int *data)

Эта функция считывает интенсивность света, падающего на фототранзисторы, установленные на нижней части робота. Функция передает значения, измеренные обоими фототранзисторами. Первое целое значение представляет собой интенсивность света левого фототранзистора (T9), второе – значение правого (T10) фототранзистора. Максимальная интенсивность имеет значение 1023, минимальная - 0.

Пример. Чтение интенсивности света с фототранзисторов (T9, T10).

```
unsigned int data[2]; //выделение памяти
…
LineData(data);
data[0] // содержит значение с левого фототранзистора (T9)
data[1] // содержит значение с правого фототранзистора (T10)
```

Пример полной программы:

```
#include “asuro.h”
int main(void) {
unsigned int data[2]; // выделение памяти
Init();
FrontLED(ON); // включение переднего светодиода
MotorDir(FWD,FWD); // направление движения
while(1){ // бесконечный цикл
LineData(data); // чтение яркости данных с фототранзисторов
if (data[0]>data[1]) //  если слева ярче, чем справа
{MotorSpeed(200,150);} // поворот налево
Else // иначе
{MotorSpeed(150,200);} // поворот направо
}
return 0;
```

void OdometrieData(unsigned int *data)

Принцип работы одометра основывается на сканировании отраженного света испускаемого светодиодом и измерении его яркости сенсором. Функция возвращает значение яркости, измеренное фототранзисторами (T11, T12). Как и в функции Linedata() параметр состоит из двух целочисленных значений, которые будут заполнены функцией. Первое целое значение - яркость от левого (T11) фототранзистора, второй целое значение - яркость от правого (T12) фототранзистора. Уровень яркости изменяется в пределах от 0 до 1023.

Пример.

```
unsigned int data[2]; //выделяем память
…
OdometrieData(data);
data[0] данные с левого фототранзистора (T11)
data[1] данные с правого фототранзистора (T12)
```

OdometrieData() не считает число оборотов, но фактически освещенность датчиков зависит от скорости вращения шестерёнок, на которых есть разбиение на светлые и тёмные области. В данном случае задача подсчёта переходов «свет-темнота» и количество оборотов остаётся на программисте.

unsigned char PollSwitch (void)

Эта функция сканирует положение датчиков (К1-К6), и возвращает один байт, содержащий информацию о датчиках, которые были активированы. Каждый переключатель устанавливает свой бит возвращаемого значения:

Bit0 (1) → K6
Bit1 (2) → K5
Bit2 (4) → K4
Bit3 (8) → K3
Bit4 (16) → K2
Bit5 (32) → K1
Активация датчиков 1, 3 и 5 вернет значение 42 (32 +8 +2 = 42).
Пример:

```
unsigned char switch;
…
switch = PollSwitch();
if (switch>0) {MotorSpeed(0,0);}
```

5. КУРС ЛАБОРАТОРНЫХ РАБОТ

Лабораторная работа № 1
Первое знакомство с ASURO ROBOT KIT

Цель работы:

Ознакомится с устройством основных узлов и систем ASURO, запустить первую программу.

План работы:

1. Прочитать руководство по ASURO.
2. Ознакомится со списком деталей из которых он состоит.
3. Изучить электрическую систему робота и установленные на нем системы.
4. Изучить алгоритм программирования робота и, если программное обеспечение не установлено, установить его и настроить.
5. Используя руководство, ознакомится с принципом работы программного обеспечения.
6. Закачать программу самотестирования.
7. Выполнить самостоятельное задание.

Порядок выполнения работы:

Ознакомится с руководством и изучить основные системы. Для загрузки программ использовать программу Flash-Tool 151. Запустить программу, выбрать нужный hex файл и нажать на кнопку Programm. Программа автоматически начнет выполнять соединение с роботом. Стоит заметить, что для успешной закачки программы нужно сначала нажать кнопку Programm, а после включить робот, затем ожидать подключения в течение 10 с. В противном случае ASURO начнёт выполнять программу, закаченную ранее.

Задание для самостоятельной работы:

Включить визуальные системы индикации робота, используя функции:

StatusLED(ON\OFF)
FrontLED(ON\OFF)
BackLED(ON,ON\OFF,OFF).

Лабораторная работа № 2
Движение по простым траекториям

Цель работы:

Ознакомится с работой двигателей робота.

План работы:

1. Ознакомится с функциями, необходимыми для движения робота по простым траекториям.
2. Написать программу движения робота по простым траекториям.

Порядок выполнения работы:

Следует изучить функции отвечающие за движение робота и написать программу для выполнения роботом простейших движений. Следует использовать следующие команды:

MotorDir(FWD,FWD\RWD,RWD\BREAK,BREAK); // направление движения робота

MotorSpeed(159,150); // скорость робота.

Стоит учитывать, что минимальная скорость, при которой робот начнёт двигаться должна быть не менее 60.

Задание для самостоятельной работы:

Запрограммировать робота для движения:

1. вперёд,
2. назад,
3. задать движение по окружности,
4. движение с поворотом на заданный угол.

Лабораторная работа № 3
Управление датчиками касания ASURO

Цель работы:

Изучить и проверить работу датчиков касания.

План работы:

1. Ознакомится с особенностями программирования датчиков из руководства.
2. Написать и запрограммировать программу-пример.
3. Выполнить самостоятельное задание

Порядок выполнения работы:

Для работы к датчикам касания следует использовать функцию PollSwitch(). Пример программы для работы с датчиками:

```
#include "asuro.h"
void main(void)
{
Init();
unsigned char sw;
while(1)
{ sw = PollSwitch();// считывание в переменную информации с датчиков
if (sw == 1) // если нажат датчик K6
MotorSpeed(200,0); // включить левый двигатель
else// иначе
MotorSpeed(0,200); // включить правый двигатель
}}
```

Задание для самостоятельной работы:

1. Написать программу, использующую все 6 датчиков касания с различными вариациями откликов (визуальные или последствием работы двигателей, рассмотренные в предыдущих работах).
2. Написать программу: движение робота вперёд и остановка при столкновении с препятствием.

Лабораторная работа №4
Управление системой одометра робота

Цель работы:

Ознакомится с работой и принципами программирования системы одометра робота.

План работы:

1. Ознакомится с особенностями программирования одометра из руководства.
2. Написать и запрограммировать программу-пример.
3. Выполнить самостоятельное задание.

Порядок выполнения работы:

Пример программы:

```
#include "asuro.h"
void main(void)
{
Init();
BackLED(OFF,OFF);// задние диоды выключены
FrontLED(OFF); // передний диод выключен
StatusLED(OFF); // диод-статус выключен
unsigned int data[2]; // создание массива куда будут заносится результаты
MotorDir(FWD, FWD); // установка направления движения
MotorSpeed(150,150); // включение двигателей

while(1){
odometrieData(data); // считывание информации с одометра
if (data[0]>757) // если значение с левого датчика достигает определенной величины
BackLED(ON,ON); // включение индикации
if (data[1]>757) // если значение с левого датчика достигает определенной величины
FrontLED(ON); // включение индикации
}}
```

Задание для самостоятельной работы:

1. Используя пример, протабулировать скорость двигателей и соответствующий им отклик датчиков.
2. Написать программу остановки робота при достижении им определенной скорости при разгоне.

Лабораторная работа № 5
Управление фотодатчиками ASURO

Цель работы:

Ознакомится с фотодатчиками робота на примере задачи удержания заданной траектории.

План работы:

1. Ознакомится с особенностями программирования системы удержания траектории.
2. Написать и запрограммировать программу-пример.
3. Выполнить самостоятельное задание.

Порядок выполнения работы:

Программа, демонстрирующая движение по линии.

```
#include “asuro.h”
int main(void){
unsigned int data[2];
Init();
FrontLED(ON); // включение переднего светодиода
MotorDir(FWD,FWD); // установка направления движения
while(1){
LineData(data); // получение данных с фотодатчиков
if (data[0]>data[1]) // если значение яркости левого датчика больше другого
{MotorSpeed(200,150);} // поворот направо
else
{MotorSpeed(150,200);} // поворот налево
}
return 0;
}
```

Задание для самостоятельной работы:

Используя пример, написать программу позволяющую роботу удерживать траекторию (воспользоваться белым листом бумаги, с нарисованной замкнутой линией черного цвета).

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. Юревич, Е.И. Основы робототехники / Е.И. Юревич. – СПб.: БХВ-Петербург, 2005. – 416 с.
2. Фу, К. Робототехника: пер. с англ. / К. Фу, Р. Гонсалес, К. Ли. – М.: Мир, 1989. – 624 с.
3. Корендясев, А.И. Теоретические основы робототехники / А.И. Корендясев. – М.: Наука, 2006. – 376 с.
4. Кудрявцев, С.А. Основы робототехники / С.А. Кудрявцев, А.А. Иванов. – Нижний Новгород: НГТУ, 2010. – 203 с.
5. Амирханов, Д.Р. Робототехнические системы: конспект лекций / Д.Р. Амирханов, А.Э. Буевич. – Витебск: УО «ВГТУ», 2010. – 100 с.

Учебное издание

КРАСНОБАЕВ Евгений Алексеевич

МЕХОВ Илья Федорович

ЛАБОРАТОРНЫЕ РАБОТЫ ПО КУРСУ «ТЕОРЕТИЧЕСКИЕ ОСНОВЫ РОБОТОТЕХНИКИ»

Методические рекомендации

Технический редактор *Г.В. Разбоева*

Компьютерный дизайн *Л.Р. Жигунова*

Подписано в печать . .2013. Формат 60x84 $^{1}/_{16}$. Бумага офсетная.

Усл. печ. л. 1,28. Уч.-изд. л. 0,57. Тираж экз. Заказ .

Издатель и полиграфическое исполнение – учреждение образования «Витебский государственный университет имени П.М. Машерова».

ЛИ № 02330/110 от 30.01.2013.

Отпечатано на ризографе учреждения образования «Витебский государственный университет имени П.М. Машерова».

210038, г. Витебск, Московский проспект, 33.