APPLICATION 1

Moving straight and detecting obstacles

Table of contents

1 Introduction **3**
2 Application description **3**
2.1 Moving the robot straight 3
2.2 Using the IR sensors to detect an obstacle 3
3 Hardware resources used **4**
4 Program description **4**
5 The example C code **6**

K-Team
MOBILE ROBOTICS

VERSION 1.0
DECEMBER 2010

Documentation Authors

K-Team S.A.
Rue Galilee 9, Y-Park
1400 Yverdon-les-Bains
Switzerland

Email: info@k-team.com
Url: www.k-team.com

In collaboration with:

Assistant Professor Sanda Paturca
Associate Professor Constantin Ilas
Department of Electrical Engineering
Politehnica University of Bucharest
Romania

Trademark Acknowledgements:

IBM PC: International Business Machines Corp.
Macintosh: Apple Corp.
SUN Sparc-Station: SUN Microsystems Corp.
LabVIEW: National Instruments Corp.
Matlab: MathWorks Corp.
Webots: Cyberbotics
Khepera: K-Team and LAMI
Pic: Microchip Corp.

LEGAL NOTICE:

- The contents of this manual are subject to change without notice
- All efforts have been made to ensure the accuracy of the content of this manual. However, should any error be detected, please inform K-Team.
- The above notwithstanding, K-Team can assume no responsibility for any error in this manual.

1 Introduction

In this application you will learn how to program K-Junior to use, in a simple way, its two main capabilities: sensing and moving.
If this is your first application, we suggest you read first the Getting Started document, learn how to open the K-Junior operating system project, add your code, build it and upload the resulting hex file on the robot.
This document shows you how to use K-Junior IR sensors and how to give simple movement commands to the robot motors.

2 Application description

Our goal is to make the robot move straight and stop when it detects an obstacle. The obstacle will be detected by using one of the robot IR sensors.

2.1 Moving the robot straight

The robot motors can be controlled using the following function:

```
KJunior_set_speed(char LeftMotorSpeed,char RightMotorSpeed)
```

In order to move in a straight line, the two speed values have to be equal. (If the two wheels have different values, the robot will move on a circular trajectory. This property is used when we want to make the robot change its trajectory). To stop the robot, the two speed values in the above function must be set to zero.

2.2 Using the IR sensors to detect an obstacle

In this application we will use the IR frontal sensor to detect the obstacle and trigger the stop command.
Infra-red (IR) sensors are very common, and can be configured in numerous ways. The typical setup has the sensor and emitter facing the same direction. The IR beam produced from the emitter bounces off objects and its reflection is detected by the sensor. For more details about K-Junior IR sensors, please read section 3.2.6 of **the K-JuniorUserManual.**

First, we have to understand the IR sensor digital values for different conditions. For this, we can simply use the sensors indicator in the K-Junior Navigator application, as seen in figure 1. This application can be downloaded from there: http://ftp.k-team.com/K-Junior/software/Applications/Navigator/

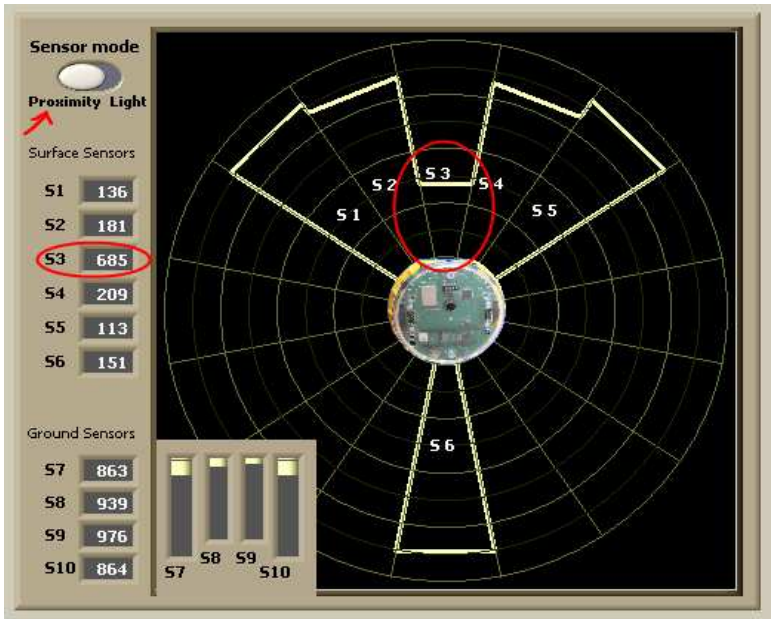


Figure 1

We select the proximity mode, which is related to the distance to the object. The sensor value varies between 0 and 1023, and note that the value increases as the distance decreases.

Now we can manually face an obstacle to the front sensor (S3) at a convenient distance that will be considered the safety distance, to avoid collision. Let's name the corresponding sensor value the "threshold value". So, when the sensor value is greater than the threshold, the robot must stop. In our application we considered a threshold value of 800.

3 Hardware resources used

- Sensors: Front IR sensor
- Motors

4 Program description

The principal task of our program is to continuously read the front sensor, deciding whether to move or stop according to the "front obstacle detected" condition.

Now we are going to edit the program in the "KJOs.c" file. First we make a copy of the entire folder "KJOS source code" and rename it, say "KJOS Application1".

The program layout that will be present from now on in our applications comprises two sections:

- **Initialization**. Here we will always call the standard robot initialization function *KJunior_init()* and do other necessary initializations.

  In this example, no other initialization is needed.

- **Infinite loop**. This section includes repetitive tasks.

In this case, the repetitive task does the following:

- Read the Front sensor with the function

  KJunior_get_proximity(char *Sensor_Name***)**

  where: *Sensor_Name* is the name of the IR sensor as defined in the KJOSManual.

- Compare the sensor value with the threshold value chosen in section 2.1 above.

  We will use a boolean variable named `front_obstacle_detected` to store the compare result

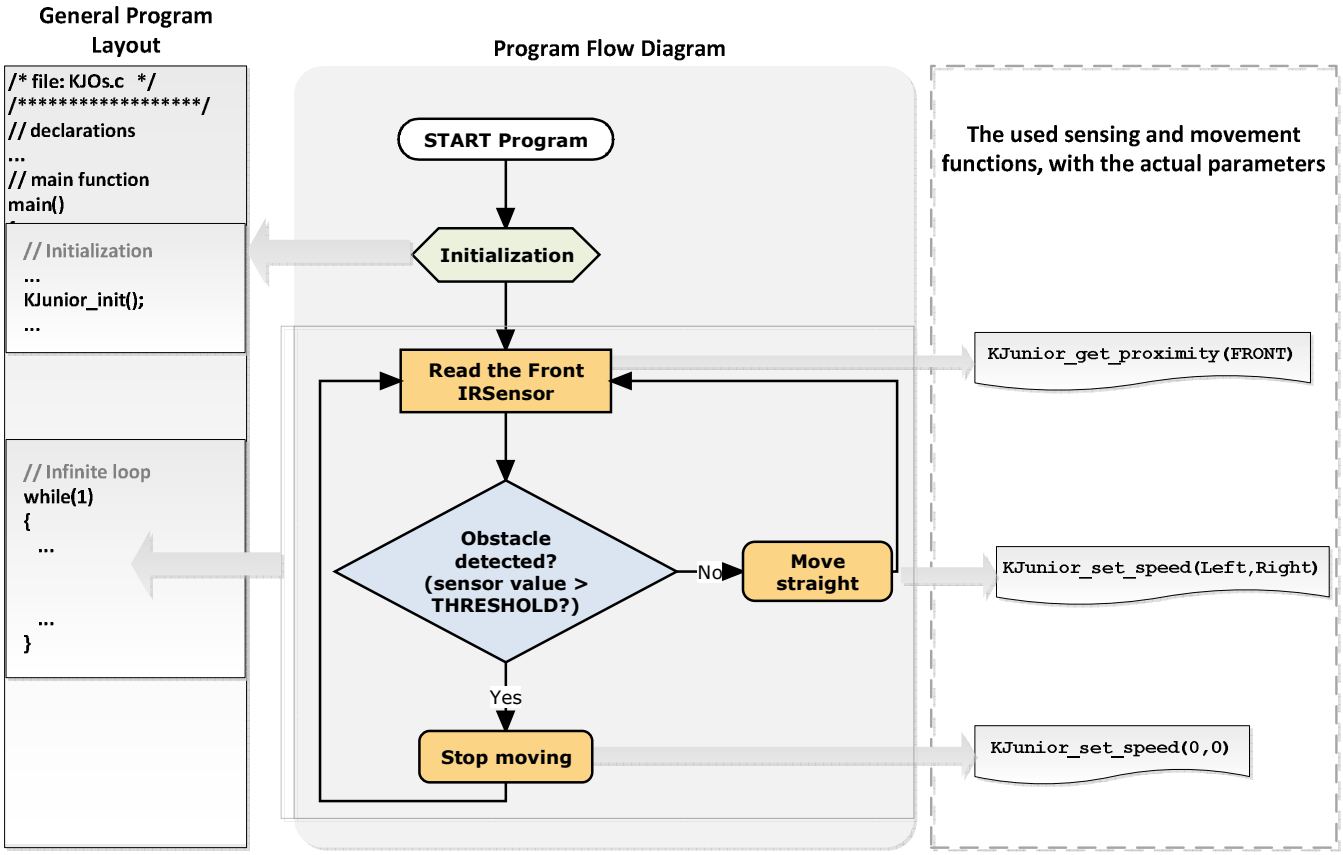
- Move or stop the robot depending on the above compare result, by calling the function

  KJunior_set_speed(char *LeftMotorSpeed*,char *RightMotorSpeed***)**

  The motor speed parameters can have any integer value between -20 and 20. Use negative values for reverse direction and 0 for stop. Use absolute values more than 4 to make the motors rotating.

  In this case the robot must only move straight, so we will use the same value for both motors.

The program flow diagram, along with the functions used for sensing and moving and the program layout, is illustrated in the figure below.



After editing the program, build your project and upload the hex file following the instructions in the "**Getting Started**" document.

Have fun!

5 The C code

```
/*****************************************************************************************/
// This is a very simple program that makes the robot moving forward and stop when it detects an obstacle in front of it.
/*****************************************************************************************/
#include "KJunior.h"

// define constants
#define THRESHOLD 800                  //  the chosen threshold value for the margin of the safety distance

// define variables
char  front_obstacle_detected;         //  boolean variable that signals the obstacle detection

//-------------------------------------------------------------------------------//
//- Main program                                                                -//
//-------------------------------------------------------------------------------//
#separate

void main(void)
{
   //  Standard robot initialization, using function defined in KJunior.c
    KJunior_init();

    //*****************************Main loop*************************************//
    while (1)          //  Infinite loop
    {
      if((SerialCommandOK == 1) && (Enable_RS232_Control == 1))
      SerialCommandHandler();

        //  Read the value of the proximity "Front" IR sensor and compares it with the threshold
        //  Save this value in the variable "front_obstacle_detect"
       front_obstacle_detect = (KJunior_get_proximity(FRONT) > THRESHOLD);

       if (front_obstacle_detect)
              //  Case of front obstacle detected - Turn the motors off
              KJunior_set_speed(0,0);     //  sets the same speed to each motors (0 to stop motors)
       else
              //  Case of space free of obstacle - The robot moves straight
              KJunior_set_speed(5,5);      //  sets the same speed to each motors, a number between 5 and 20
   }
}
/*****************************************************************************************/
```