

THE NAVIGATION CONTROL OF A ROBOTIC STRUCTURE

Laurean BOGDAN¹, Gheorghe DANCIU², Flaviu STANCIULEA³

¹University LUCIAN BLAGA of Sibiu, ²Tera Impex SRL, ³Tera Impex SRL

e-mail: laurean.bogdan@ulbsibiu.ro, danciu@teraimpex.ro

Keywords: mechatronics, kinematics, sensors, PBASIC Software, Light Sensitive Navigation

ABSTRACT: The goal of this paper is to build a robotic structure together, program it for basic maneuvers and then proceed to adding sensors and other detection devices, followed by developing programs which enable the robot to react to its surroundings and perform autonomous tasks.

To aid the movement of the robot, two servo motors were used and two optical sensors called photoresistors. The main device of this paper, the robot, includes a carrier board in its structure which is mounted on a wheeled chassis.

This paper also enables the understanding of the operating parameters of a microcontroller and also permits to transfer the graphical results of the detected signals, to a computer.

1. INTRODUCTION

Building and programming a robot is a combination of mechanics, electronics and computer programming, synthesized in one word, mechatronics.

Robots are used in the auto, medical, and manufacturing industries, in all manner of exploration vehicles, and, of course, in many science fiction films.

The general topics covered by this paper are: basic robot navigation under program control, navigation using a variety of sensor inputs, navigation using feedback and various control techniques, and navigation using programmed artificial intelligence. The motion is based on DC motors. Dialogue between motors and microcontroller is made by a special driver.

After a brief introduction of the brain of the robot represented by the microcontroller, we will pursue with the construction, testing and calibration of the robotic structure.

A microcontroller is the kind of miniature computer that you can find in all kinds of gizmos. Here are some examples: if your clock radio goes off, and you hit the snooze button a few times in the morning, the first thing you do in your day is interact with a microcontroller. Heating up some food in the microwave oven or making a call on a phone also involves operating microcontrollers. That's just the beginning. Here are a few more examples: turning on the television with a handheld remote, playing a handheld game, using a calculator, and checking your digital wristwatch. All those devices have microcontrollers inside them that interact with you.

The microchip used to control the motion of the robot that we present in this paper is a PIC16C57. This microcontroller is mounted on a module which also includes other devices like a voltage regulator, a resonator and a EEPROM memory.

A picture of such a module is posted in the figure below:

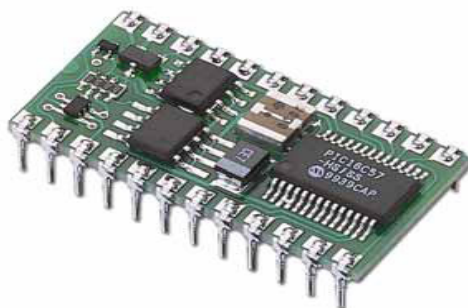


Fig.1 The module which enables the communication with the robotic structure

Robots have been designed to do everything from helping students learn more about microcontrollers, to mowing the lawn, to solving complex mechanical problems. On each of these robots, microcontrollers are used to read sensors, control motors, and communicate with other computers.

For this paper we will use a combination of hardware equipment and microcontroller programming which will determine the navigation of our robot.

2. LIGHT DETECTING SENSORIC SYSTEMS

Light sensors have many different functions, and they come in different shapes, with different price tags. Some sensors are designed to sense a particular color of light, such as blue, green, red, or infrared. Some sensors don't care what color the light is because they react to how bright the light is. Other sensors look for only special kinds of light given off by certain chemical reactions. Light sensors also have a variety of ways to tell a microcontroller what they see. Some sensors send a voltage, some send a sequence of binary values, and others react to different kinds of light or light levels by changing resistance.

Of the light sensors that react to light by changing their resistance, the photoresistor shown in Figure 2 is probably the most common, least expensive and easiest to use. Its active ingredient is a chemical compound called cadmium sulfide (CdS). This compound changes resistance depending on how bright the light is that shines on its collecting surface. Bright light causes low resistance values between the two leads while dim light causes higher resistance values.



Fig. 2. Photoresistor Schematic

Light has many applications in robotics and industrial control. Some examples include sensing the edge of a roll of fabric in the textile industry, determining when to activate streetlights at different times of the year, when to take a picture, or when to deliver water to a crop of plants. The light sensor used in this paper is designed to detect visible light, and it can be used to make the robot detect variations in light level.

With this ability our robotic structure can be programmed to recognize areas with light or dark perimeters, report the overall brightness and darkness level it sees, and seek out light sources such as flashlight beams and doorways letting light into dark rooms.

Common resistors have fixed values, the photoresistor on the other hand is a light dependent resistor. This means that its resistance value depends on the brightness, or illuminance, of the light falling on its light detecting surface.

Illuminance is a scientific name for the measurement of incident light. One way to understand incident light is to think about shining a flashlight at a wall. The focused beam that you see shining on the wall is incident light. The unit of measurement of illuminance is commonly the "foot-candle" in the English system or the "lux" in the metric system. While using the photoresistors we won't be concerned about lux levels, just whether illuminance is higher or lower in certain directions.

The robot in this paper will be programmed to use the relative light intensity information to make navigation decisions.

3 Mechanical Assembly Process

In order to proceed to the programming of the robot, first it has to be mounted. The first step consists of placing the servomotors and the battery pack in the chassis of the robot like illustrated in the figure below:

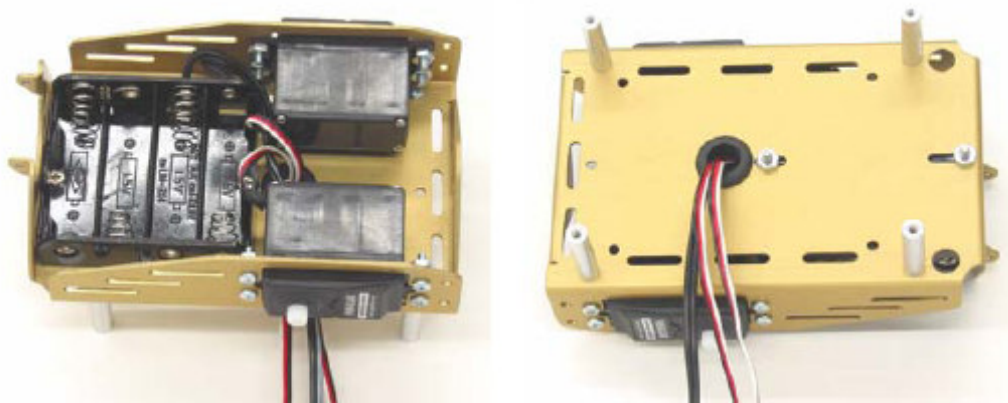


Fig.3 Assembly of Battery Pack and Servos

Next the wheel hardware has to be attached to the horns of the servos.

As a final step the development platform which includes the microcontroller, has to be mounted on the chassis and then to complete the assembly we have to make all the electrical wire connections.

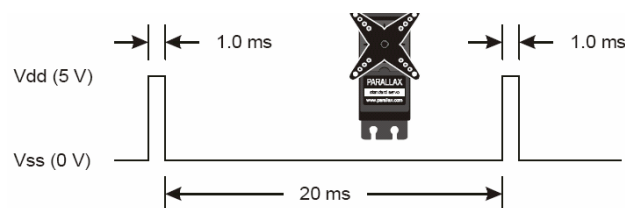


Fig.4 Timing Diagram for servomotors

3.1 Hardware Setup

The most important device on the development board is the before mentioned microcontroller. This chip has the task to send the codes received from the computer to the robot. Below there's an enhanced view of this device.

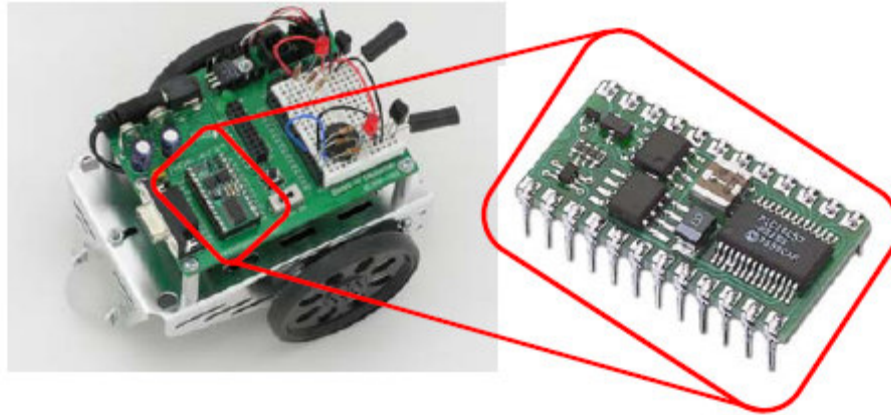


Fig.5 Microcontroller module on development board

In order to proceed to the programming part, we have built the light level sensor circuit. This circuit will be able to detect the difference between shade and no shade. The parts list needed for this experiment will be: (2) Photoresistors – CDS; (2) Capacitors – $0.01 \mu\text{F}$; (2) Resistors - 220Ω ; (2) Jumper wires; (1) piezoelectric speaker.

All these devices have to be mounted on the development board's prototyping area. The wiring diagram that appears in the next figure has to be fully respected for a proper movement of the robot.

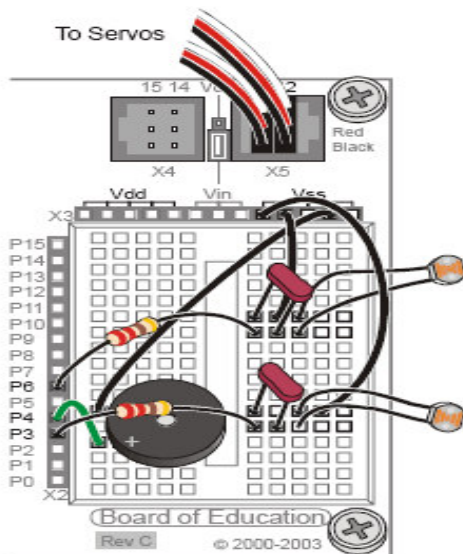


Fig.6 Wiring Diagram for photoresistor circuit

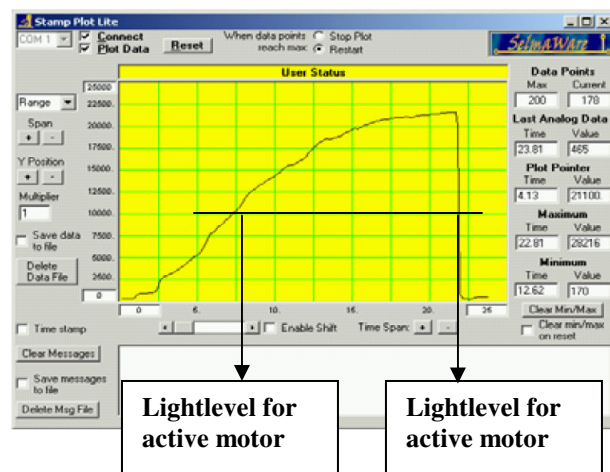


Fig. 7 Light Diagram for servomotors

4. Software development

The program that we developed can be used to guide the robot through exiting a fairly dark room toward a doorway that's letting in brighter light. It also allows for much better control over the Robot's roaming by casting shadows over the photoresistors. Hence the program was named 'ROAMING TOWARD THE LIGHT'. The key to roaming toward brighter light sources is going straight ahead when the differences between the photoresistor measurements are small, and turning toward the smaller photoresistor measurement when there is a large difference between the two measurements. In effect, this means the robot will turn toward bright light.

4.1 Programming Instructions

In order to enable the robotic structure's movement the following program was used:

```
' Boe-Bot roams, and turns away from dark areas in favor of
brighter areas.
' {$STAMP BS2} ' Stamp directive.
' {$PBASIC 2.5} ' PBASIC directive.
DEBUG "Program Running!"
' Declare variables for storing measured RC times of the
' left & right photoresistors.
timeLeft VAR Word
timeRight VAR Word
average VAR Word
difference VAR Word
FREQOUT 4, 2000, 3000
DO
GOSUB Test_Photoresistors
GOSUB Average_And_Difference
GOSUB Navigate
LOOP
Test_Photoresistors:
HIGH 6 ' Left RC time measurement.
PAUSE 3
RCTIME 6,1,timeLeft
HIGH 3 ' Right RC time measurement.
PAUSE 3
RCTIME 3,1,timeRight
RETURN
Average_And_Difference:
average = timeRight + timeLeft / 2
difference = average / 6
RETURN
Navigate:
' Shadow significantly stronger on left detector, turn
right.
IF (timeLeft > timeRight + difference) THEN
PULSOUT 13, 850
PULSOUT 12, 850
```

```

' Shadow significantly stronger on right detector, turn
left.
ELSEIF (timeRight > timeLeft + difference) THEN
PULSOUT 13, 650
PULSOUT 12, 650
' Shadows in same neighborhood of intensity on both
detectors.
ELSE
PULSOUT 13, 850
PULSOUT 12, 650
ENDIF
PAUSE 10
RETURN

```

5. Extension possibilities

The flexibility of the software allows us to make numerous applications with this robotic structure by just changing a few lines and parameters in the main program. The hardware setup must not be modified at all which simplifies the work of the user. But by adding further parts on the prototyping area like new types of sensors, the range of experiments which can be performed with this robot will increase. Below we listed an example of a different program:

```

' Boe-Bot avoids black pieces of paper.
' {$STAMP BS2} ' Stamp directive
' {$PBASIC 2.5} ' PBASIC directive.
LeftWhite CON 16
RightWhite CON 33
LeftBlack CON 26
RightBlack CON 45
LeftAvg CON LeftWhite + LeftBlack / 2
RightAvg CON RightWhite + RightBlack / 2
timeLeft VAR Word ' Left photoresistor
reading
timeRight VAR Word ' Right photoresistor
reading
FREQOUT 4, 2000, 3000
DO
GOSUB Test_Photoresistors
GOSUB Navigate
LOOP
Test_Photoresistors:
HIGH 6 ' Left RC time
Measurement.
PAUSE 3
RCTIME 6,1,timeLeft
HIGH 3 ' Right RC time
measurement.
PAUSE 3
RCTIME 3,1,timeRight

```



```

RETURN
Navigate:
' Both detect black paper, back up and make a noise
IF (timeLeft > LeftAvg) AND (timeRight > RightAvg) THEN
PULSOUT 13, 650
PULSOUT 12, 850
FREQOUT 4, 20, 4400 ' Beep instead of pause
' Left detects black paper, turn away to right, make a noise
ELSEIF (timeLeft > LeftAvg) THEN
PULSOUT 13, 850
PULSOUT 12, 850
FREQOUT 4, 20, 2200
' Right detects black paper, turn away to left, make a noise
ELSEIF (timeRight > RightAvg) THEN
PULSOUT 13, 650
PULSOUT 12, 650
FREQOUT 4, 20, 3300
' Neither detects black paper, go forward one pulse.
ELSE
PULSOUT 13,850
PULSOUT 12,650
PAUSE 20
ENDIF
RETURN
This program for instance will keep the robot confined in a
black bordered space.

```

6. Conclusions

This paper focused on measuring differences in light intensity and programming the robot to act on these differences. A pair of cadmium sulfide (CdS) photoresistors were used to measure differences in visible light. The CdS photoresistors were first connected to resistors to form voltage dividers, and the microcontroller monitored the voltage at the connection between the photoresistor and the fixed resistor. When this voltage dropped below or raise above 1.4 V the input register for the I/O pin connected to the circuit stored either a 0 or 1. The robot was programmed to make decisions using these binary values.

The photoresistor divider technique works so long as the right resistors are chosen and The lighting doesn't change. However, a much more versatile way of detecting light Levels with the microcontroller are to use the CdS photoresistor in an RC circuit, charge the capacitor, and then measure the decay time. RC stands for resistor capacitor, and the capacitor was introduced in this paper along with a circuit that makes it possible for the microchip to measure RC decay time. This is easily done, using the RCTIME command, which is specifically designed for measuring RC decay and growth times.

Constants were introduced as a way to substitute meaningful names for numbers that are used in a PBASIC program. Scaling and averaging were also introduced.

Scaling was used to determine the average value of the light levels in an area based on the two photoresistor RC time measurements. This was used to create a threshold that Automatically self-adjusted to the overall lighting conditions, eliminating the need to Change resistors when the light levels change.

7. References

- [1] Bogdan, L., Dorin, A. *Acționarea și comanda electrică a mașinilor unelte și roboților industriali*, 1998, Editura Bren, București, 1998.
- [2] Bogdan, L., Breaz, E.R. *Study regarding the influence of information conversion upon the positioning systems accuracy*, ICMAS, Bucuresti, 2004.
- [3] Bogdan, L., Cenuse, L. *SCADA – Tehnics and Dedicated Methodes for Reducing of Material and Energetical Losses*. Electrotehnica aplicata in Eco-reconstructia industrială, ECOEIC`05, Sibiu-Romania, 23-24 septembrie 2005.
- [4] Doug Pientak , *Understanding Signals* version 1.0 2004
- [5] Andy Lindsay, *What's a microcontroller* Student Guide Version 2.1
- [6] Parallax Inc. *Robotics!* Version 1.5 2000
- [7] Parallax Inc. *Applied Sensors* Version 2.0 2003