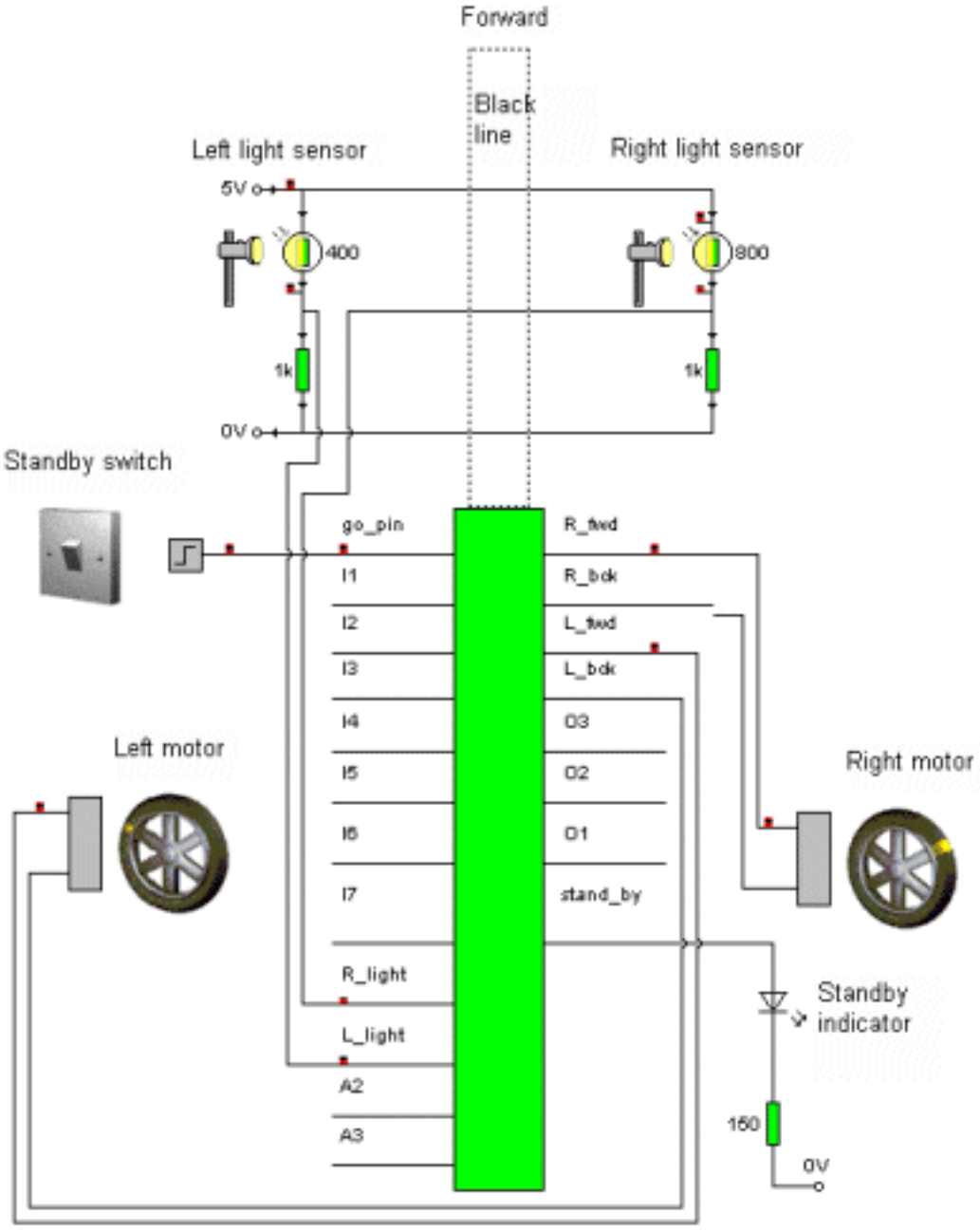
Black Line Follower

Introduction

The final objective when learning about microcontrollers is to program a device that does something in the real world. In this activity you will study a flowchart that simulates how a buggy could be controlled by a microcontroller such that it automatically follows a black line drawn on white surface.

You will then convert the flowchart into the equivalent PBasic code. Then type this code into the Program editor software and download the code onto a buggy and watch it follow the black line.

Task 1



Black line following buggy

1. Open Crocodile Technology file Model 1.

2. Make sure the standby switch is off. Start the flowchart running. Describe what happens to the outputs.

3. Do not adjust the light levels yet. Switch on the standby switch. Describe what happens to the outputs.

4. Imagine that the buggy is moving on a large piece of white paper. Drawn on the paper is a thick black line. Initially the buggy is placed such that the light sensors are on either side of this black line.

   Light is reflected off the paper. More light is reflected off the white surface than off the black surface (i.e. the line).

   Try adjusting both the left and the right lamps shining onto the LDRs. Look at the "Monitoring flowchart1" box. Describe what happens to the variables shown when you adjust the light levels.

5. Adjust both lamps to the top (bright) position. This simulates both sensors being above the white paper.

   If the buggy is moving forwards then at some point one of the light sensors will move over the black line. Simulate this by reducing the light levels on one sensor. Describe what happens. How would this affect the position of the buggy over the black line.

6. The sensors provide analog inputs to the microcontroller. These analog voltage signals are converted within the microcontroller by an ADC (analog to digital converter). The output from the ADC is a number that represents the value of the analog voltage input from the sensors.

   Study the flowchart. At what level of input signals from the light sensors do the motors stop.

7. You will have to obtain a buggy which is configured the same way as the simulator.

   Once all the hardware is set up download the flowchart onto the microcontroller on the buggy by clicking on the red circle. Place the buggy on your big piece of white paper with the thick black line drawn round it. Describe what happens.

8. In all likelihood your buggy did not follow the black line straight away. With the help of your teacher you will have to adjust the time delays in the subroutines and the values used checking whether or not your sensors are over the black line. Having done all this describe what happens.

9. Now convert the flowchart into the equivalent PBasic code. Remember to include the subroutines at the end of the main flowchart loop.

   Load the Program editor software and type the code into it. Download your program onto the buggy. Describe what happens.

10. Again a good bit of adjustments will be necessary to get the buggy working exactly as required. Based on how your buggy performs keep adjusting and downloading your program until the buggy follows the black line. Describe your final results.

Programming microcontroller's in real devices can be fun and very rewarding. Many different possible set ups is possible with the hardware involved in this activity. You will have to spend some time adjusting parts of your system to get it to do exactly what you want.

Having completed this exercise you will then be ready to program PICAXE microcontroller's in a wide range of applications - limited only by your own imagination.

Remember however to always design your system on paper and in simulation before moving onto the final hardware. That way you can fix many of the problems before

they become expensive mistakes.

-- END OF ACTIVITY --