Flowchart Software for a Modular Robot

M. A. Joordens
Deakin University, Waurn Ponds, Australia
majoor@deakin.edu.au

***Abstract**: Flowcharting is a common method of setting out the requirements for a piece of code. It is simple with few rules to follow. Rarely however, is it used as the code itself. This paper describes the outline of a software package that uses the flowchart as the code for a small, autonomous, modular robot, designed for use in High Schools and Universities at an introductory level.*
By using flowcharting the student is introduced to the concept of structured programming. A flowchart is often the first step in programming. Here it is the only step, easing the student into the art of coding, and simplifying the teachers job.

***Keywords**: Robotics, Educational software, Flowchart*

Introduction

During a project the author had at a high school that involved enabling a small group of students to build a robot that could move over a piece of paper, drawing as it went, a discussion developed between the author and the teachers. The point of the discussion was that the teachers were looking for a robot system that could take the students beyond the Lego "Mindstorms" robot. The "Mindstorms" robot was too limiting with only 3 inputs and 3 outputs. The teachers found that some of the students mastered this robot to easily are required a more flexible robot to continue their learning progression. The result was a small, autonomous, modular robot that the students could plug together in any configuration.
One hurdle in this robot system was the programming of the robot. The teachers did not want a programming language that would take too long for the teachers to learn and so it was decided to use flowcharting, which the teachers had experience with, as the language.

This software package was written to allow high school students to program a small modular robot with little or no previous experience. The robot itself was designed to be studied after the students had had some experience with the Lego "Mindstorms" robot system.

This meant that the students should have had some ability with programming with icon-based, flowchart-like systems. This also had a bearing on the decision to use flowcharting as the method of programming.

Other Flowcharting software

As the Flowcharts were to be a stage that followed the Lego Mindstorms icon based system, the author looked in this area and found the "Chart Programming Language".(Gilder 2001) This is a flowchart based system. Unfortunately on a more detailed examination it was found

that the system allowed the student to write a flowchart then to write the code from the flowchart. As one requirement was too just use the flowchart itself as the code, this option was discarded.

Dr. Tia Watts from the Sonoma State University is working on a flowchart editor.(Watts 2002) This is just a GUI flowchart editor, but could be turned into a compiler with a little more work. It is very comprehensive but was a little to detailed for the job at hand.
Matrix Multimedia has a very promising package called "Flowcode".(Matrix Multimedia Limited 2002) This package allows students to design complex electronic systems straight from the Flowchart. Its output is used to program a PIC microprocessor. It is however, very detailed, found to take a while to learn and a different processor had already been chosen for the modular robot.

Inhouse Software

In the end, it was decided that the author would write this software. This gave the author full control over the code and no licensing issues. As the author also had to write the code that resided in the robot, it was easy to ensure compatibility and to make the job easier.

The Robot

To write the software, a little needs to be known about the robot it is interfacing to. The Robot has 16 addressable 1 byte inputs and 16 addressable 1 byte outputs. Inputs and Outputs, known as Ports, are for the sensors that the students attach to the robot. The robot also has 4 motors and a timer system.

Figure 1: The modular robot

The Robot's code

To simplify the programming job for the students, it was decided to let the robot's code do as much as possible. The robot used the 68HC08 microprocessor with 1K of RAM, 512 Bytes of EEPROM and 32K of Flash memory. The download and Flash programming code resides in the EEPROM as no code is allowed to run in Flash while it is being programmed. The download code accepts S19 files for programming.
The beginning of the Flash was reserved for the student program whilst the remaining code sat in the last bit of Flash. This code had a standard loop that would continuously run the students code. The main part of the code is in an interrupt routine that occurs every 20ms. The routine updates the Pulse Width Modulation signal for each of the motors from the data

in a series of memory locations. It updates the 16 output ports from more data in memory and reads the 16 inputs ports and save the data in memory. It also increments the timer system.

The Software

With the robot's code handling all of the robot's system from data in memory, all that the student's code needed to do was access and manipulate the memory. The various memory registers can be seen in the following dialog box.

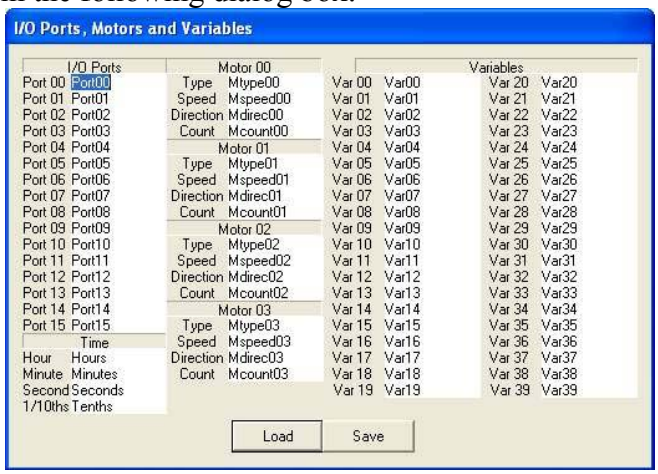

Figure 2: The available memory locations

There are the 16 input/output ports, the time registers, the 4 motor registers and 40 variable registers. In this dialog box we see that each address is given a default name that can be changed by the student to make more sense. For instance, If port00 is a Light Dependant Resistor (LDR) on the left side of the robot, its name could be changed to LDR_Left.
This greatly simplified the job of the complier in the software and it simplified the Flowchart required. The Flow chart has two basic blocks, a process block and a decision block.

The Process Block

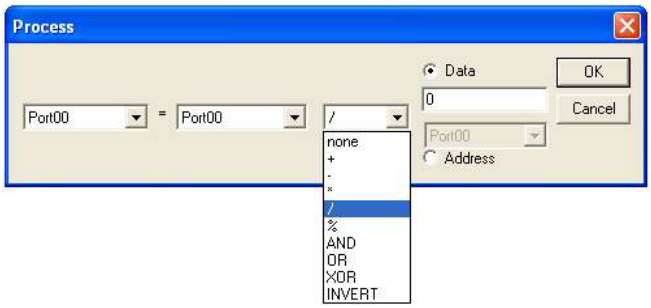

Figure 3: Process dialog box

The above dialog box shows the basis of the process block. It allows the student to set any of the memory locations to any arithmetic or logical combinations of other memory locations or data.

The Decision Block

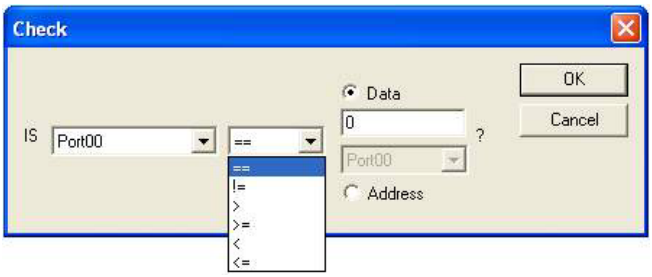

Figure 4: Decision dialog block

As can be seen here, any memory location can be compared to any other location or data and a decision can be made on the outcome.

A typical Flowchart would look like this:

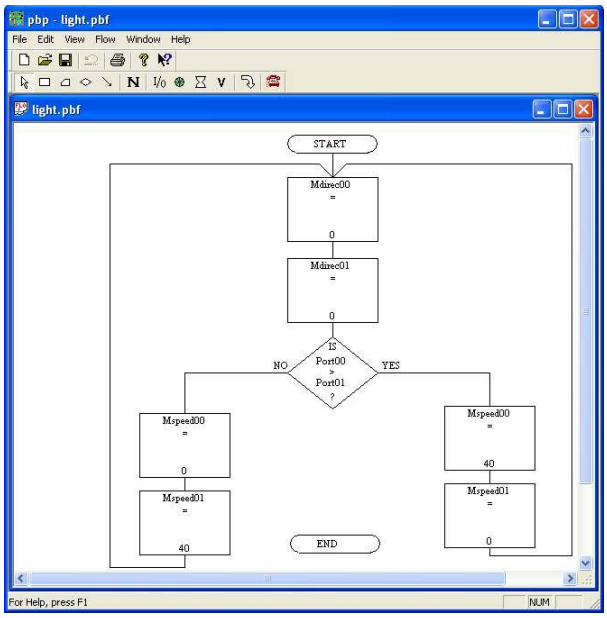

Figure 5: The software with a flowchart example

This Flowchart sets the direction of the motors and then compares input ports 0 and 1, turning the robot one way or the other by setting the motor speeds. If ports 0 and 1 were LDR sensors, then the robot would seek light. Note that the End block is not reached. Each flowchart requires a Start and End block. This flowchart never ends.

The Run Block

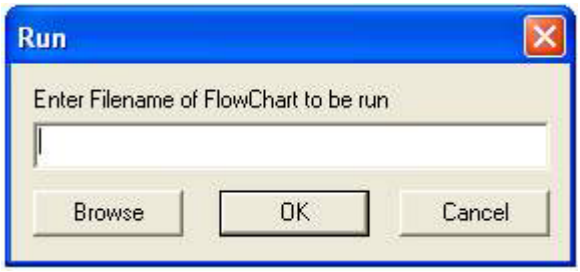

Figure 6: Run dialog box

While this Flowchart is fairly simple, more complex Flowcharts are possible. To this end, a Run Block was added. The run block simply calls a flowchart that has been saved as seen in the following example.

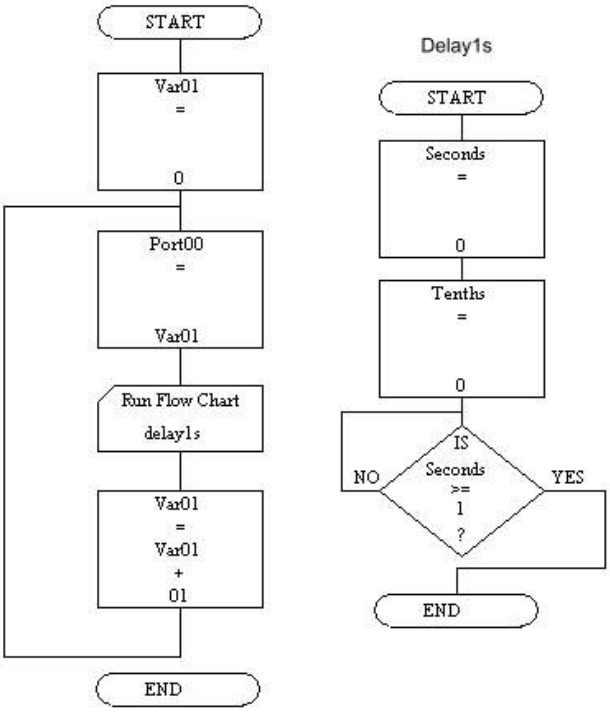

Figure 7: Example of flowchart calling another flowchart

The flowchart on the left increments a variable in a continuous loop. In that loop a Run Block calls the Flowchart on the right, which creates a one second delay. Thus the variable is incremented once per second.

Compiling and Downloading code

Once the Flowchart is drawn the student selects download. As it was decided that the flowchart was to be the program as far as the student was concerned, all processing from this point is hidden from the student. First the software compiles the flowchart in a two pass process. Because of the limited number and variation of blocks, each possibility is precompiled. Thus the compiler just selects the required code, inserts the memory location or data and ends the block with a jump to the next block or blocks. The second pass inserts the jump addresses.

This done, the software creates the S19 record and then uses the MSComm routines to download the code, via a serial port, to the robot.

The serial port was chosen over USB as all high schools will have serial port on their PC's while only the most up to date schools will have PCs with USB. In the worst case, a school with only USB can get USB to serial converters very cost effectively.

While serial communications is more complex under Windows 95 and later, it also has a greater capability.(Mirho 1996) Or does it?

Communications

The biggest problem the author had in writing this software was with the Serial port communications. The software was written using Microsoft's Visual C++ as the author was most familiar with this language. The creators of the MSComm routines wrote these routines for Visual Basic and Microsoft converted them for Visual C++. This means that there is very

little support and documentation for them and the author found that a lot of trial and error was required to get these routines to work correctly. For example, the comm. Port likes to shut itself down and so the software must regularly check if the port is closed and reopen it.

The Student

The challenge for the student is two fold. As the robot is modular, the student must be able to assemble the robot in the correct configuration for the chosen task. (This aspect is not covered in the paper.) That done, the student must use the limited flowchart blocks available to create the correct flowchart to enable the robot to complete the task. As each input and output module has an address which the student chooses, the student must learn about address busses. This in turn can be used to teach data and control busses.
Once the student knows how to address the input and output modules, the student is able to begin flowcharting. As the flowchart block deal only with simple arithmetic and logic functions, the student must learn to use these fundamental operations to design functional flowcharts that can perform a specific task, such as controlling a motor and creating a time delay. This is similar to how a student learning to program in assembly language must learn, thus the student is being prepared for further programming studies at a later date.
The functional flowcharts can now be pieced together to create an operational flowchart.
For advanced students, including secondary and tertiary students, the flowchart language can be bypass and any language, such as assembly or C, can be used as the robot itself does not accept the flowchart but S19 files which are an industry standard for the programming of microprocessors. Thus any complier that can create S19 files for the robots Microcontroller, the M68HC05, can be used.

Current Field Trails

There are currently six prototype robots at Brauer College in Warrnambool. The students there have taken to the robots very readily. Preliminary verbal feedback is that the students have accepted the robots and enjoy working with them. They prefer them to the "Mindstorms" robots because they are able to do more with them.
Although the science teachers use the robots in the robotics subject, the computer teachers now want their students to use the robots as they find the robot flowcharting system as a good way to introduce their students to structured programming and because the robots also provide an introduction to computer architecture.

Conclusion

The author found this project to be a challenging but thoroughly enjoyable exercise. It pointed out to him, again, that writing code that talks to external devices, especially one off devices such as this projects robot, is one of the harder types of code to work on.
The robot designed gives the teacher a structured environment in which to teacher the fundamentals of programming in a manner that can capture the students attention and thus allows both the teacher and students to obtain more out of the learning process.
The robot affords an introduction to robotic principles, structured programming, and computer architecture in a fun and an easily approachable way. The robot can also be used as a robotic base for secondary and tertiary students. Its modular approach and industry standard programming system, allow it to be used as both an introductory robot and a robot for advanced learning.

References

Book:

Charles A. Mirho, Andre Terrisse, (1996) "*Communication Programming for Windows 95*", Microsoft Press, Washington

Online source:

Jason Gilder (2001). "*The CHART Programming Language for the Lego Mindstorm*", Wright State University. Retrieved from http://brig.cs.wright.edu/legoasm.html

Matrix Multimedia Limited (2002), "*Flowcode*". Retrieved from http://www.matrixmultimedia.co.uk

Tia Watts (2002), "*SFC - A Structured FlowChart Editor*", Sonoma State University. Retrieved from http://www.cs.sonoma.edu/~tiawatts