ABSTRACT

Microcontrollers can be found everywhere. They are becoming more numerous and more powerful. Programming them is well within the reach of the average computer scientist. This tutorial will introduce computer scientists to microcontrollers, their capabilities, applications, development environments, programming styles, and uses in the computer science curriculum.

1. INTRODUCTION

Microcontrollers have become part of our everyday lives. Billions of them are sold every year, and they control our appliances, vehicles, and even our toys. They are cheaper, more powerful, and easier to utilize than ever before.

In the past, microcontrollers have mostly been left to electrical and electronics engineers, “the hardware guys.” There’s no reason for that to remain the case. Microcontrollers and, perhaps more importantly, information about microcontrollers, have become more accessible to the point where they are usable even by non-technical hobbyists.

Microcontrollers consist of a CPU, memory, clock, IO ports, and other peripherals all combined on a small integrated circuit. These days, a typical microcontroller might have a 20MHz 8-bit RISC processor, 16kB of flash storage, 1kB of RAM, and several types of input and output, including multi-channel digital-to-analog and analog-to-digital converters, all for around four dollars; less in quantity. Of course, some are less powerful and others are much more powerful.

Some microcontrollers can run off a battery for months or even years, making for very interesting applications. They can also be used in extreme environments where one would never dream of risking a full-fledged computer. They can interface with an incredible array of sensors and control a wide range of devices via their IO ports.

In the past, programming microcontrollers meant using assembly language, but these days they can be programmed in C and even higher-level languages.

This hour and a half long tutorial will introduce participants to microcontrollers, their capabilities, applications, development environments, programming styles, and uses in the computer science curriculum.

2. PURPOSE AND OBJECTIVES

The tutorial will discuss the differences between the various types and brands of microcontrollers, including the very popular Arduino platform. At the conclusion of the tutorial, participants should be able to choose a microcontroller for a project.

Applications for microcontrollers will be discussed. The presenter will show microcontrollers in various form factors. Participants will be able to identify situations in which microcontrollers could be utilized.

The presenter will demonstrate the development environments for popular microcontrollers, including simple programming examples. Using hardware programmers to load software into microcontrollers will be demonstrated. Various resources for learning about microcontrollers will be shared. Participants should leave the tutorial with the confidence to start learning about the microcontroller of their choice.

The tutorial will also cover how programming for microcontrollers differs from programming for a typical computer, even when using the same programming language.

Usage of microcontrollers in a computer science curriculum will also be discussed. In addition to the obvious class on programming for microcontrollers, they could be used in an architecture class to compare and contrast to the typical microprocessor architectures (x86, MIPS, etc.). Also, while students sometimes resist learning x86 assembly because they believe they will never use it, students can immediately see the utility of using assembly in a resource-limited environment such as a microcontroller. The logistics of using microcontrollers in the classroom will also be discussed.

Beyond acquiring a useful and marketable skill, there are other reasons for students to learn to program microcontrollers. While getting started with microcontrollers and embedded programming is incredibly simple, it is inevitable that students will encounter new ideas and challenges if they move beyond the basics. They will steadily acquire valuable knowledge regarding sensors and electronics, and become more well-rounded computer scientists and engineers. Both beginners and advanced students can find microcontroller projects that will hold their interest and force them to learn new concepts.
3. INTENDED AUDIENCE

The tutorial will be appropriate for anyone with even minimal programming experience. Knowledge of the C programming language is useful, but certainly not required.

The tutorial should be of interest to instructors, students, and anyone interested in seeing what microcontrollers are capable of.