EVALUATION OF ALTERNATIVE MICROCONTROLLERS FOR MECHATRONICS EDUCATION

Burford Furman and Eric Moen
Mechanical and Aerospace Engineering
San José State University
San José, CA 95192-0087 USA

Abstract

We conducted a broad search for an alternative microcontroller that could be programmed in the C language, be affordable for students to purchase their own, and be capable enough to be used in introductory as well as more advanced courses in Mechatronics at San José State University or other universities with similar courses.

Our method consisted of identifying criteria for evaluation, investigating which microcontrollers were being used in other institutions with mechatronics courses, compiling a manageable list of microcontroller candidates, and evaluating the top candidates.

We came up with a spreadsheet and rating scheme along with a list of top candidates and their key features. The Rabbit RCM3410 by Z-World and the Atmega128 by Atmel scored most favorably.

Introduction and Overview

Over the past decade or so, many universities have developed courses, programs, and even degrees in mechatronics (see for example,[3],[6]), where mechatronics is often defined along the lines of, “the synergistic integration of mechanical engineering with electronics and intelligent computer control in the design and manufacture of products and processes.” [8]. A common thread in most courses in mechatronics is the focused use of microcontrollers in some kind of hands-on laboratory environment[3]. By and large, the eight-bit microcontroller has been the preferred choice for mechatronic laboratory experiments and student projects, however some courses are making use of digital signal processors (DSP’s) and personal computers (PC’s), for example, [17],[18], and [16]. We will describe here our recent experience with several microcontrollers and a systematic evaluation of alternative microcontrollers for mechatronics education.

Mechatronics at San José State University

At San José State University, we began development of a curriculum stem in mechatronics in 1993, perhaps the first of its kind in the United States. Under a grant from the National Science Foundation Division of Undergraduate Education and substantial support from local industry, we developed a series of five courses and several laboratories to support undergraduate instruction in mechatronics. Our mechatronics program has been described in several papers[7],[5].

ME 106 Fundamentals of Mechatronics is the first course in our mechatronics curriculum where students are exposed to microcontrollers. The laboratory portion of ME 106 is conducted in the SJSU Mechatronic Engineering Laboratory, which consists of ten identical workbenches each having a power supply, function generator, digital multimeter, oscilloscope, and PC. Furman[5] describes the development of the laboratory and the early experiments.

Chronology of Microcontrollers Used in ME 106

In 1996, we originally began using the Motorola 68HC11 microcontroller packaged in the NMIX-0029-OEM single board computer from New Micros, Inc. (www.newmicros.com). We chose the NMIX-0029-OEM because of its versatility and expandability. A variety of add-on boards with functions such as solid-state relays, mechanical relays, servo motor controllers, etc. are available and easily plug into sockets on the main board. We also chose the ICC11 C compiler from Imagecraft (www.imagecraft.com), a widely used cross-compiler for the 68HC11 microcontroller. The per-station cost for microcontroller and software was approximately $200.

The NMIX-0029-OEM and ICC11 compiler arrangement served us well for several years, but we observed after a while that students who had taken ME 106 were subsequently gravitating, on their own accord, to the Basic Stamp microcontroller (www.parallax.com) for other class projects. The Basic Stamp is a relatively low-cost microcontroller that is especially suited for educational uses. The attraction for our students was its low-cost, approximately $50, and ease of use. The Basic Stamp uses PBASIC, an interpreted language that is available as a free download from Parallax.

In 2000, we decided to switch to the Basic Stamp in ME 106 to leverage the benefits that arose from students being able to buy their own microcontroller and software and have the freedom to develop applications "anytime, anywhere" rather than being bound by the fixed hours and the fixed location of the Mechatronic Engineering Laboratory. This arrangement worked well for several years, but always presented some fundamental limitations. Foremost was that students were not programming in C, which most learn several semesters earlier in an introductory programming class. This made it harder for the students and the instructor of the follow-on mechatronic capstone class, ME 190 Mechatronics System Design, which was still making use of the 68HC11. Also, knowledge of and experience with C is arguably far more useful for the students in their professional career than PBASIC, so using the Basic Stamp in some ways was doing a disservice to the students. A second issue is that the Basic Stamp is quite limited in terms of speed, functionality, and program capacity. While students were able to quickly learn how to use the Basic Stamp, they often bumped into its limitations for the term project in ME 106: no onboard A/D, 32 bytes of variable space, 2 Kbytes of program memory, integer math only, no interrupts, relatively slow program execution speed, etc.

In the fall of 2003, we switched to the OOPic microcontroller (www.oopic.com) packaged in the OOBOT 40-II board by Oricom Technologies (www.oricomtech. com). The OOPic is also a relatively inexpensive microcontroller (about $50), and its software is free for download. It has substantially more capability than the Basic Stamp, can be programmed in Basic, C, or Java syntax.

Also in 2003 we began to look for an alternate microcontroller that would provide a better balance between cost and functionality, be programmable in C, and that could be used in all of our mechatronics courses that have laboratory components. The following section describes our approach to the evaluation and its findings. Additional details pertaining to the evaluation can be found in Moen,[10].

Search Method for an Alternative Microcontroller

Our method consisted of identifying criteria for evaluation, investigating which microcontrollers were being used in other institutions with mechatronics courses, compiling a manageable list of microcontroller candidates, and evaluating the top candidates.

The criteria for evaluation were broadly grouped into three areas: hardware features, "user friendliness", and cost. An additional constraint was that a C compiler must be available for the microcontroller. As mentioned above, for the sake of cohesion in the curriculum and professional development of the students, we decided to only consider microcontrollers that could be programmed in C.

With regard to hardware features, we considered the configuration of digital and analog inputs and outputs, communications ports, timers, memory, and processor speed. In general, the more I/O, the better (especially at the same price); however, it was important that the controller be "well rounded." For example, a controller with one hundred digital lines should not be rated as high as a controller that has only twenty digital I/O but also has two analog inputs, and a 16-bit timer.

With regard to cost, we included not only the cost of the processor, but also the cost of the compiler, interface board, and/or any other expense related to extras that might be required (i.e. programming cable, DAC or A/D chips, LCD, etc.)

With regard to "user friendliness", we considered a variety of factors such as, good documentation, clear example programs, ergonomics of the user interface for the programming environment, layout, type, and robustness of hardware and I/O connections, etc.

Microcontrollers Used by Other Universities

We began the search by first considering what microcontrollers were being used by other universities with courses in mechatronics. Table 1 lists the seven universities we investigated, the microcontroller(s) used, and the associated course or project. Interestingly, the Motorola 68HC11/12 is being or has been used at five of the seven schools.

Formation of Candidate List

The second step in compiling the list of candidates was to go to the websites of microcontroller manufacturers and try to sort through the myriad of choices they provide. Our selection of a short-list of candidates was guided by considering cost, availability of a C-language cross-compiler, amount and type of I/O, memory, and processor speed. Candidates with more memory and higher speed were given higher preference. Table 2 presents the short-list we came up with. The Basic Stamp and OOPic microcontrollers were included on the list for baseline comparisons.

With this more manageable list of microcontrollers, a rating scheme was created based on memory, speed, I/O, and cost. Each category was normalized to a percentage of the "Best-in-Class" controller for that category, and then a total score was determined for each controller by averaging the scores in each category. Table 3 displays the comparison chart with each controller's options and rating values. The difference in total scores is shown in the table.

Several columns in Table 3 ere highlighted to indicate either the highest-ranking microcontrollers or microcontrollers of "special interest." A "special interest" microcontroller is one that had an outstanding reason that made it interesting to evaluate regardless of the total score it received. For example, the OOPIC controller received a relatively low score; however, due to its unique programming scheme, it was included as a possible candidate for evaluation for further study.

Evaluation of Candidate Microcontrollers

The highlighted microcontrollers from Table 3: Z-world RCM3410, Tern 586-Engine, Atmel Atmega 128, and the Technological Arts Adapt812DX were then singled out for further evaluation. The following subsections expand upon the information shown in Table 3.

Z-world RCM3410

The Z-world RCM3410 microcontroller (www.zworld.com) received the highest total score as compared to the other candidates (64.8%). The RCM3410 has a relatively fast processor, plenty of I/O, and lots of memory for a relatively low cost. It also has Ethernet capability, and its programming environment, Dynamic C, uses a unique feature called "Co-statements." Co-statements are a means by

Table 1 – Microcontroller used at various universities. This list gave us a subset of microcontrollers to consider.

School	Microcontroller	Course/Project
UC Berkeley	C167	EECS 43 Calbot Project
	HC11	EE 192
	Hitachi SH-2	EE 290A
	Xilinx Spartan-XL FPGA	ME 235
Stanford	HC11	ME 118/218
Purdue	Atmel Atmega 16	ECET 209
Georgia Tech	HC11	ME 3056
MIT	PIC16F876	SP.710: REC Spring 2001 Intensive Seminar #1 "Introduction to Programmable Microcontrollers"
	HC11	
	HC12	
San Jose State	Basic Stamp	ME 106
	Handyboard (HC11)	ME 285
UC Davis	Tern 586 Engine	The SAE Walking Machine Challenge

Table 2 – List of microcontrollers for evaluation. This table is a short-list of candidates for subsequent evaluation. The Basic Stamp and OOPic microcontrollers were included for baseline comparisons.

Technological Arts (Motorola)	Microchip	OOPIC	Ubicon / Scenix	Tern	Basic Stamp	Atmel	Z-world
Adapt11C24DX-60K (HC11)	PIC17C766	OOPIC2 S	SX52BD/PQ	586-Engine	OEMBS2	Atmega 128	BL1810
Adapt812DX128 (HC12)	PIC18F8720	OOPIC R	SX28AC/DP				OEM2500 or OEM 2510
							LP3510
							RCM2020
							RCM3010
							RCM3410

which a programmer can accomplish pseudo-multitasking of the processor. As a program enters a co-statement loop, it may exit that loop prematurely to wait on an event. The next time the program enters the same co-statement loop, it resumes at the place where it previously exited. This allows the processor to execute other commands in the mean time.

The Z-world microcontroller comes with an outstanding documentation package and a user-friendly programming environment. Z-world's website has a good technical support sections that include bulletin boards, FAQs, technical notes and white papers, and a variety of resources for someone who needs help. In addition, the Dynamic C software has good help

Table 3 (and continued on the next page) Microcontroller comparison. The table lists the items we considered important to evaluate, and rates each as a percentage relative to the best-in-class score for that item. The total rating for each microcontroller is an average of the percentages for each item category.

		Zworld						Technological Arts	
		Single Board Computers			Core Modules			HC11 Family	HC12 Family
		BL1810	OEM2500 or OEM2510	LP3510	RCM2020	RCM3010	RCM3410	Adapt11C24DX- 60K	Adapt812DX128
Processor	Type	Rabbit 2000	Rabbit 3000	Rabbit 3000	Rabbit 2000	Rabbit 3000	Rabbit 3000	68HC11E0FN	68HC812A4CPV8
Memory	Flash	128000	256000	256000	256000	256000	256000		128000
	+ EEPROM							32000	4000
	+ SRAM	128000	128000	128000	128000	128000	256000	512	
	+ External RAM							32000	128000
	= Total	256000	384000	384000	384000	384000	512000	64512	260000
	Rating (vs max)	**50.00**	**75.00**	**75.00**	**75.00**	**75.00**	**100.00**	**12.60**	**50.78**
Speed	MHz	14.732	29.4	7.4	18.432	29.4	29.4	8	16
	Rating (vs max)	**11.08**	**22.11**	**5.56**	**13.86**	**22.11**	**22.11**	**6.02**	**12.03**
I/O	Digital Inputs	6	8	16	8	4	3		
	+ Digital Outputs	8	8	10	6	4	3		
	+1.1* Configurable	14	0	0	26	44	47	38	60
	= Total "Score"	29.4	16	26	42.6	56.4	57.7	41.8	66
	Rating (vs max)	**39.30**	**21.39**	**34.76**	**56.95**	**75.40**	**77.14**	**55.88**	**88.24**
	Analog Inputs	1	1	0	0	0	8	8	8
	* (bits)	8	10	0	0	0	11	8	8
	+ Analog Outputs	2	2	3	0	0	0	0	0
	* (bits)	9	9	9	0	0	0	0	0
	= Total "Score"	26	28	27	0	0	88	64	64
	Rating (vs max)	**16.25**	**17.50**	**16.88**	**0.00**	**0.00**	**55.00**	**40**	**40**
	Timers	5	0	0	5 1	10 1	10 1	0	8
	* (bits)	8	0	0	8 10	8 10	8 10	0	16
	= Total "Score"	40	0	0	50	90	90	0	128
	Rating	**31.25**	**0.00**	**0.00**	**39.06**	**70.31**	**70.31**	**0.00**	**100.00**
	Serial	4	5	6	4	6	5	2	2
	+10* Ethernet	NO	YES	NO	NO	Yes	YES	NO	NO
	= Total "Score"	4	15	6	4	16	15	2	2
	Rating (vs max)	**25.00**	**93.75**	**37.5**	**25**	**100**	**93.75**	**12.5**	**12.5**
	Motor Control	NO	NO	NO	NO	No	No	NO	YES
	+ LCD	NO	NO	YES	NO	No	No	NO	NO
	+ Keypad	NO	NO	YES	NO	No	No	NO	NO
	+ Quad Encoder	NO	NO	NO	NO	Yes	Yes	NO	NO
	+ Compatator	NO	NO	NO	NO	No	No	NO	NO
	= Total "Score"	0	0	20	0	10	10	0	10
	Rating (vs max)	**0.00**	**0.00**	**100.00**	**0.00**	**50.00**	**50.00**	**0.00**	**50.00**
							69.24		58.15
Cost	Processor	$69	$99	$149	$39	$55	$39	$93	$129
	+ Compiler	$28	$28	$28	$28	$28	$28	$0	$0
	+ Extra Hardware	$0	$0	$0	$50	$50	$50	$0	$0
	= Total	$97	$127	$177	$117	$133	$117	$93	$129
	Rating (vs min)	**60.82**	**46.46**	**33.33**	**50.43**	**44.36**	**50.43**	**63.44**	**45.74**
	Total Rating (avg)	**29.2**	**34.5**	**37.9**	**32.5**	**54.6**	**64.8**	**23.8**	**49.9**

Table 3, cont. Microcontroller comparison. The table lists the items we considered important to evaluate, and rates each as a percentage relative to the best-in-class score for that item. The total rating for each microcontroller is an average of the percentages for each item category.

		Ubicon / Scenix		Microchip		OOPIC		Tern	Basic Stamp	Atmel Procyon
		SX52BD/PQ	SX28AC/DP	PIC17C766	PIC18F8720	OOPIC2 S style	OOPIC R	586-Engine	OEM BS2	Atmega128
Processor	Type	SX52BD/PQ	SX28AC/DP	PIC17C766	PIC18F8720	OOPIC2	OOPIC2	AMD Elan SC520	Basic Stamp 2	Atmega128
Memory	Flash	49000	24000					256000		128000
	+ EEPROM				1000	32000	32000		2000	4000
	+ SRAM	262	136	902	3840	4000	4000	64000		4000
	+ External RAM								32	64000
	= Total	49262	24136	902	4840	36000	36000	320000		200000
	Rating (vs max)	**9.62**	**4.71**	**0.18**	**0.95**	**7.03**	**7.03**	**62.50**	**0.00**	**39.06**
Speed	MHz	50	50	33	25	0.002	0.002	133	0.004	8
	Rating (vs max)	**37.59**	**37.59**	**24.81**	**18.80**	**0.002**	**0.002**	**100.00**	**0.003**	**6.015**
I/O	Digital Inputs									
	+ Digital Outputs									
	+1.1* Configurable	40	20	66	68	31	16	32	16	53
	= Total "Score"	44	22	72.6	74.8	34.1	17.6	35.2	17.6	58.3
	Rating (vs max)	**58.82**	**29.41**	**97.06**	**100.00**	**45.59**	**23.53**	**47.06**	**23.53**	**77.94**
	Analog Inputs	0	0	16	16	0	4	11	0	8
	* (bits)	0	0	10	10	0	8	12	0	10
	+ Analog Outputs	0	0	0	0	0	0	2	0	
	* (bits)	0	0	0	0	0	0	12	0	
	= Total "Score"	0	0	160	160	0	32	156	0	80
	Rating (vs max)	**0**	**0**	**100**	**100**	**0**	**20**	**97.5**	**0**	**50**
	Timers	2	0	2 2	2 3	0	0	7	0	2 2
	* (bits)	16	0	8 16	8 16	0	0	16	0	8 16
	= Total "Score"	32	0	48	64	0	0	112	0	48
	Rating	**25.00**	**0.00**	**37.50**	**50.00**	**0.00**	**0.00**	**87.50**	**0.00**	**37.50**
	Serial	0	0	3	3	1	1	3	1	2
	+10* Ethernet	NO	NO	NO	NO	NO	NO	NO	NO	NO
	= Total "Score"	0	0	3	3	1	1	3	1	2
	Rating (vs max)	**0**	**0**	**18.75**	**18.75**	**6.25**	**6.25**	**18.75**	**6.25**	**12.5**
	Motor Control	NO	NO	NO	NO	NO	YES	NO	NO	NO
	+ LCD	NO	NO	NO	NO	NO	NO	NO	NO	NO
	+ Keypad	NO	NO	NO	NO	NO	NO	NO	NO	NO
	+ Quad Encoder	NO	NO	NO	NO	NO	NO	NO	NO	NO
	+ Compatator	yes	NO	NO	NO	NO	NO	NO	NO	YES
	= Total "Score"	10	0	0	0	0	10	0	0	10
	Rating (vs max)	**50.00**	**0.00**	**0.00**	**0.00**	**0.00**	**50.00**	**0.00**	**0.00**	**50.00**
Cost	Processor	$9	$4	$13	$10	$59	$79	$289	$59	$0
	+ Compiler	$199	$199	$495	$495	$0	$0	$0	$0	$0
	+ Extra Hardware	$50	$50	$50	$50	$0	$0	$0	$0	$70
	= Total	$258	$253	$558	$555	$59	$79	$289	$59	$70
	Rating (vs min)	**22.84**	**23.29**	**10.57**	**10.63**	**100.00**	**74.68**	**20.42**	**100.00**	**84.29**
	Total Rating (avg)	**25.5**	**11.9**	**36.1**	**37.4**	**19.9**	**22.7**	**54.2**	**16.2**	**45.7**

files, many example programs, and a useful function lookup feature that helps a user understand the arguments and actions of a particular function included in the Z-world libraries.

Dynamic C also has a useful debugging mode where a user can set breakpoints and watch variables, stepping through sections of code to help with debugging. The only problem we encountered with the Z-world microcontroller happened when trying to program it from a laptop computer. The Z-world microcontroller uses a special programming cable that has a built-in circuit to convert 3-volt RS232 signals coming from the processor to 5-volt RS232 signals going to the serial port of a computer. The programming cable uses power from the computer's serial port to power the circuit, and we surmise that because some laptops use low-power serial ports to help conserve battery life, this arrangement sometimes may cause intermittent communication errors between the host computer and the Z-world processor. A solution for this problem (aside from using a desktop computer without power limitations) would be to supply an external power source to the built-in circuit. The RCM3410 has other attractive features such as quadrature encoder inputs, PWM outputs, pulse capture and measurement capabilities, a real-time clock with battery-backup, and a pre-assigned MAC I.D. for Ethernet connectivity.

Tern 586-Engine

The Tern 586-Engine microcontroller (www.tern.com) had the second highest total score as compared to the other candidates (54.2%). It has the fastest processor (AMD586) and plenty of I/O and memory. It is also the most expensive controller, and is well out of the price range of the average student despite the fact that the compiler is free. The Tern microcontroller comes with a good documentation package and a user-friendly programming environment. Tern's website has a technical support section that includes bulletin boards, and FAQs; however, they are not comprehensive. The online technical support, however, does include a real-time chat mode with an expert. This can prove to be extremely useful when immediate assistance is needed, as long as your problems occur between the hours of 8 am and 5 pm Pacific Standard Time! The 586-Engine comes with excellent example programs, and useful function libraries. The programming environment, Paradigm C++ Tern Edition, has several nice features including project building, color-coded text editing, a jump-to-function feature, etc. Paradigm is free of charge, is relatively easy to use, and has a useful debugging mode similar to Dynamic C.

OOPic

The OOPic microcontroller (www.oopic.com) had one of the lowest total scores as compared to the other candidates (22.7%). The OOPic is based on the PIC 16F877 microcontroller from Microchip. The OOPic runs an interpreter program in firmware (rather than using a compiler) that allows the user to write programs using BASIC, C, or Java syntax. It has relatively limited I/O and memory, but it does, have analog inputs and a nice arrangement of hardware for mechatronics experiments. There are several power and ground connections, and some of the inputs and outputs are arranged so that they can easily interface with standard hobby-style servomotors. Also, there is a small speaker that can play generated frequencies, and three LEDs that are connected to corresponding input switches.

The OOPic microcontroller comes with good documentation and an extremely user-friendly programming environment that is free and available for downloading on the OOPic website. The OOPic website has sections that include documentation updates, and FAQs. It comes with excellent example programs mostly in BASIC (some in C). The software is object-oriented, and it includes a host of easy-to-use objects for controlling I/O, motors, servos, sensors, etc. The help files are quite good and can be updated readily via the Internet. Setting up the connections to the OOPic is automatic, and simply pressing the Run button takes care of interpreting the program, downloading it to an EEPROM, and running it.

One of the major disadvantages of the OOPic is its slow speed in executing a program, because it runs an interpreter instead of using compiled machine code. It does have the capability to link properties of objects (linked objects are called “virtual circuits”), which results in much faster execution and something akin to multitasking. We found that even though the OOPic is slow for code that does not use object linking, it can produce a PWM signal with a frequency up to 2.5 MHz [10]. The OOPic has a 5 MHz crystal and a PWM object that can generate a variety of waveforms.

Basic Stamp

The Basic Stamp OEMBS2 microcontroller (www.parallax.com) had the lowest total score as compared to the other candidates (16.2%). This low score is offset slightly by the fact that it least expensive microcontroller. However, the Basic Stamp, based on the Microchip PIC16C57 microcontroller, has limited I/O (16 pins of digital I/O only), and it would really need an interface board to incorporate A/D input or timing features that the other controllers already have.

The Basic Stamp also uses an interpreter for programming in PBASIC rather than a compiler and has limited I/O and memory. It is only slightly faster than the OOPic in terms of instruction time, however, it does not have analog inputs or outputs and does not have the object-oriented nature of the OOPic. The Basic Stamp has no extra hardware on board, such as switches, LEDs, pullup/pulldown resistors, etc. This means that the students must provide any external circuitry to interface with the digital I/O, and run a higher risk of burning out the controller or some of its pins. The OEMBS2 controller comes with good documentation and a user-friendly, easy to learn programming environment. The Parallax website has several sections that include documentation, technical support, controller knowledge bases, and FAQs. While the Basic Stamp hardware and software are easy to learn and use, one can quickly exhaust their capability beyond relatively modest projects.

Adapt812DX

Technological Arts makes a board based on the Motorola HC12 microcontroller called the Adapt812DX (www.technologicalarts.com). This microcontroller scored fairly high as compared to the other candidates (49.9%).

As seen in Table 3, the Adapt812DX has plenty of configurable I/O and memory and reasonable speed. The cost to use this controller is affordable for students especially since a freeware C compiler (GNU-C) is available. The HC12 processor itself has thorough documentation, but we found it difficult to figure out how to set up the Adapt812DX controller from Technological Arts’ documentation. They have a website with a section for technical support that includes links to different bulletin boards, some technical articles, and FAQs for their HC12 products, however, there seemed to be little help for users who are just starting out with the HC12 microcontroller.

There are a few freeware-programming environments that use the GNU-C compiler, such as EmbeddedGNU, Engler[4]. EmbeddedGNU was found to be a good choice after trying several different alternative interfaces. It is a useful interface for the GNU-C compiler, allows for building projects that include several files, and uses color-coded text editing for the C language. In addition, it has a terminal program that can be used to download programs to the HC12, and will create makefiles and memory.x files for the compilation process using the GNU-C compiler. Unfortunately, due to the lack of documentation regarding GNU-C for the HC12, we found it difficult to successfully compile, download, and run a C-language program on the Adapt812DX controller.

Atmega 128

The Atmel Atmega 128 microcontroller (www.atmel.com) placed fourth in total scores as compared to the other candidates (45.7%). An evaluation platform for the Atmega 128 is available from Digikey (www.digikey.com). The platform consists of two boards: the STK500 and STK501, which sell for about $80 ($79 each less a 50% educational discount) at the time of this writing. These boards come with user switches and LEDs, easy access to all I/O ports, and 9-pin DB connectors for the serial ports. Procyon Engineering (www.procyonengineering.com) also sells a nicely implemented, low-cost (about $70) board that is highly compatible with the STK 500. The Procyon board was used in the comparison.

The Atmega 128 has plenty of I/O that is configurable as either inputs or outputs and a reasonable amount of memory. In addition, it has several timers, an analog comparator, A/D converters, and two serial ports. The Atmega 128 controller comes with an outstanding documentation package and a user-friendly programming environment called AVR Studio. Atmel's website (www.atmel.com) has satisfactory technical support sections that allow a user to email for help however, a much better resource for technical help with the Atmel AVR series of processors can be found at the AVRfreaks website (www.avrfreaks.com). This website includes bulletin boards, FAQs, technical notes and white papers, and a variety of resources for help. There are also several freeware compilers and utilities available for the Atmel AVR series (including the Atmega 128) and some nice C-language libraries and examples available as well (see for example, [22] and [19]).

The programming environment of AVR Studio is a familiar Windows application with the standard features such as Cut, Copy, and Paste. Several windows can be opened at once, and compiling and downloading to the target device can be done with ease. There is a small learning curve to become proficient at building, compiling, and downloading a project from scratch, but the environment does allow for flexibility. The Atmega 128 comes with an internal, 8MHz crystal, however, an external crystal up to 16MHz can drive it. Despite having a slower internal clock speed than some of the other controllers, the Atmega 128 outperforms even the Tern 586 Engine when using simple bit writing functions to change the state of a digital output as fast as possible (2.8 MHz compared to about 0.85 MHz respectively [10]). This indicates that the compiler and/or the instruction set for the processor is more efficient, typically using only single clock cycles to execute its instruction set. The Tern can still out-clock the Atmega 128 when using PWM output compare modes with its timer outputs, however, the Atmega 128 is impressive nonetheless.

Final Ranking of Microcontrollers

After doing the more in-depth evaluation of the aforementioned microcontrollers, we came up with a final ranking based on all factors including a "user-friendliness" measure. The user-friendliness measure was developed by giving each microcontroller a score out of 10, based on their ease of use. This included the programming environment, whether an interface board was provided or needed to be developed separately, the physical layout of such boards, technical support, documentation, and, more generally, how much effort was needed to get a simple program up and running. After the microcontrollers were ranked, the user-friendliness score was determined by converting it to a percentage value. This score was then combined with those, which rated hardware and cost into the Total Rating, which is shown at the bottom of Table 3. Table 4 shows the resulting Total Rating considering several weighting schemes applied to the cost and user-friendliness scores. Other weighting arrangements could be explored, but we considered cost and ease of use as two of the most important factors in our search for an alternative microcontroller that students could affordably purchase and easily use on their own. With equal weights applied to all rating scores, the Z-world RCM3410 and the Tern 586-Engine have the highest total ratings respectively. With additional weight given to cost and user-friendliness, the Atmel Atmega128 displaces the Tern. The Z-world RCM3410 is consistently in the top two, even when more emphasis (up to 4x) is placed on cost and ease of use.

Table 5 shows the final rating values for the top candidate microcontrollers with user-friendliness included and 2x weighting applied to the cost and user-friendliness scores. The Z-World RCM3410 and Atmel Atmega 128 controllers received the highest total rankings (64% and 57% respectively). The OOPic controller came in fourth on the scale (41%), since it was the easiest controller to use.

CONCLUSIONS

There are many varieties of 8-bit microcontrollers to choose from for use in Mechatronics education. We found that the Rabbit RCM3410 from Z-World and the Atmega128 from Atmel rated highest in our scheme of evaluating cost, availability of a C-language cross-compiler, amount and type of I/O, memory, processor speed, and user-friendliness.

We hope that our investigation will be helpful to other educators who may be considering alternatives to the microcontrollers they are presently using. The spreadsheets we used are available upon request.

References

1. Atmel Corporation, http://www.atmel.com/.
2. AvrFreaks, http://www.avrfreaks.com/.
3. Carryer, J. E. (organizer) (1994). Proceedings of the Workshop on Mechatronics Education, Stanford University, July 21-22, 1994.
4. Engler, E. (2004). Embedded Tools web page, [online: http://www.geocities.com/englere_geo/]
5. Furman, B. J., Hsu, T. R., Wang, J. C., Barez, F., Tesfaye, A., Hsu, P., Reischl, P.(1996). "Laboratory Development for Mechatronics Education," Session 1626, Proceedings of the ASEE Annual Conference, Washington, D. C., June 23-26, 1996.
6. Golnaraghi, F., Gorbet, R. (organizers) (2001). Proceedings of the Forum on Mechatronics in Canada: Past Experience and Future Directions, May 18, 2001, University of Waterloo, Waterloo, Ontario N2L3G1 [on-line: http://mecforum.uwaterloo.ca/proceedings.pdf].

Table 4 – Total rating results considering various weightings for cost and user-friendliness scores. The Z-world RCM3410 is consistently in the top two. The Atmel Atmega 128 joins the top two when more emphasis is given to cost and user-friendliness.

	Total Rating Results			
	Cost and User-Friendliness Weighting			
Microcontroller	1x	2x	3x	4x
Z-world RCM3410	65	64	64	63
Adapt812DX128	48	46	45	44
OOPic R	31	41	49	54
Tern 586-Engine	54	50	48	46
Basic Stamp	22	34	42	47
Atmel Atmega 128	50	57	61	65

Table 5 - Final Ranking of Microcontrollers. The ranking scheme includes a rating for "user-friendliness" and 2x factors applied to the cost and user-friendliness scores in Total Rating score.

		Zworld Core Modules RCM3410	Technological Arts HC12 Family Adapt812DX128	OOPIC OOPIC R	Tern 586-Engine	Basic Stamp OEM BS2	Atmel Procyon Atmega128
Processor	Type	Rabbit 3000	68HC812A4CPV8	OOPIC2	AMD Elan SC520	Basic Stamp 2	Atmega128
Memory	Flash	256000	128000		256000		128000
	+ EEPROM		4000	32000		2000	4000
	+ SRAM	256000		4000	64000		4000
	+ External RAM		128000			32	64000
	= Total	512000	260000	36000	320000		200000
	Rating (vs max)	**100.00**	**50.78**	**7.03**	**62.50**	**0.00**	**39.06**
Speed	MHz	29.4	16	0.002	133	0.004	8
	Rating (vs max)	**22.11**	**12.03**	**0.002**	**100.00**	**0.003**	**6.015**
I/O	Digital Inputs	3					
	+ Digital Outputs	3					
	+1.1* Configurable	47	60	16	32	16	53
	= Total "Score"	57.7	66	17.6	35.2	17.6	58.3
	Rating (vs max)	**77.14**	**88.24**	**23.53**	**47.06**	**23.53**	**77.94**
	Analog Inputs	8	8	4	11	0	8
	* (bits)	11	8	8	12	0	10
	+ Analog Outputs	0	0	0	2	0	
	* (bits)	0	0	0	12	0	
	= Total "Score"	88	64	32	156	0	80
	Rating (vs max)	**55.00**	**40**	**20**	**97.5**	**0**	**50**
	Timers	10 1	8	0	7	0	2 2
	* (bits)	8 10	16	0	16	0	8 16
	= Total "Score"	90	128	0	112	0	48
	Rating	**70.31**	**100.00**	**0.00**	**87.50**	**0.00**	**37.50**
	Serial	5	2	1	3	1	2
	+10* Ethernet	YES	NO	NO	NO	NO	NO
	= Total "Score"	15	2	1	3	1	2
	Rating (vs max)	**93.75**	**12.5**	**6.25**	**18.75**	**6.25**	**12.5**
	Motor Control	No	YES	YES	NO	NO	NO
	+ LCD	No	NO	NO	NO	NO	NO
	+ Keypad	No	NO	NO	NO	NO	NO
	+ Quad Encoder	Yes	NO	NO	NO	NO	NO
	+ Comparator	No	NO	NO	NO	NO	YES
	= Total "Score"	10	10	10	0	0	10
	Rating (vs max)	**50.00**	**50.00**	**50.00**	**0.00**	**0.00**	**50.00**
Cost	Processor	$39	$129	$79	$289	$59	$0
	+ Compiler	$28	$0	$0	$0	$0	$0
	+ Extra Hardware	$50	$0	$0	$0	$0	$70
	= Total	$117	$129	$79	$289	$59	$70
	Rating (vs min*)	**50.43**	**45.74**	**74.68**	**20.42**	**100.00**	**84.29**
User Friendly	Rank (out of 10)	7	3	10	5	7	9
	Rating	**70.00**	**30.00**	**100.00**	**50.00**	**70.00**	**90.00**
Total Rating (weighted avg)**		**64**	**46**	**41**	**49**	**34**	**57**

** weighted average gives extra consideration to cost and user friendly ratings by applying a 2x weighting factor in the average.

7. Hsu, T.R. (1995). "Mechatronics for Undergraduate Mechanical Engineering Education", 1995 ASEE Annual Conference Proceedings, Vol. 1, pp. 1312-1324.

8. Kaynak, O. (1996). Special Issue of Robotics and Autonomous Systems, vol. 19, no. 2, Dec.1996, pp. 113-116 [on-line: http://mecha.ee.boun.edu.tr/cv/publications/g1.html].

9. Microchip, http://www.microchip.com/.

10. Moen, E. (2003). Microcontrollers for Mechatronics Education: An Evalutation of Alternatives and Adaptation for Use at San José State University, SJSU Department of Mechanical and Aerospace Engineering Masters of Science Project Report.

11. Motorola, Inc., http://e-www.motorola.com/

12. OOPic, http://www.oopic.com/.

13. Oricom Technologies, http://www.oricomtech.com/.

14. Parallax, http://www.parallax.com/.

15. Procyon Engineering, http://www.procyonengineering.com/.

16. Shetty, D., Kolk, R. A., (1997). Mechatronics System Design, PWS Publishing Company, Boston.

17. Shoureshi, R. (1994). "Foundations for Mechatronics Education," Proceedings of the Workshop on Mechatronics Education, Stanford University, July 21-22, 1994, pp.103-106.

18. Simões, M. G., Szafir, S. (2003). "A DSP Computer Platform for Mechatronics Teaching and Research," Computers in Education Journal, vol. 13, no. 4, October-December 2003, pp. 15-25.

19. Stang, P. "Procyon AVRLIB, C-Language Function Library for Atmel AVR Processors," [on-line: http://hubbard.engr.scu.edu/embedded/avr/avrlib/].

20. Technological Arts, http://www.technologicalarts.com/.

21. Tern, Inc., http://www.tern.com/.

22. Weddington, E. (2003) WinAVR software [on-line: http://sourceforge.net/projects/winavr/].

23. Wikander, J., Törngren, M., Hanson, M. (1999). "Mechatronics Engineering - Science and Education," Proceedings of the 1999 IEEE International Conference on Robotics and Automation, May 10-15, 1999, Detroit, MI [on-line: http://www.damek.kth.se/RTC/Papers/Mecha-ICRA99.pdf].

24. Z-world, Inc.,http://www.zworld.com/.

Biographical Information

Burford Furman is currently an Associate Professor in the Department of Mechanical and Aerospace Engineering at San José State University, where he has been since 1994. He received a Bachelor's and Masters degree in Mechanical Engineering from UC Davis in 1980 and 1982 respectively, and a PhD in Mechanical Engineering from Stanford University in 1991. From 1982 to 1986 and following his doctoral study, he worked at IBM in the development of disk drive products. His areas of interest for research and teaching are mechatronics, engineering measurements, and precision machine design.

Eric Moen is currently a design engineer in the semiconductor equipment industry. He received a Bachelor's degree in Mechanical Engineering from UC Davis in 1996, and a Master's degree in Mechanical Engineering from San Jose State University in 2003. Since achieving his undergraduate degree in 1996, he has worked as a research and development process engineer and equipment design engineer for the disk drive and semiconductor equipment industries. His professional areas of interest include mechatronics, system integration, and precision machine design.