<u>Microcontrollers</u>

A First User Guide to the Technology

Contents

Chapter 1: What are Microcontrollers?

Basically a microcontroller is a computing device, and is a single integrated circuit (“silicon chip” or IC) used to form part of a product that incorporates some software program control. As a microcontroller is basically part of a computing system it can be used in applications requiring control, operator and user display generation, simple sequencing and many other mundane tasks.

In the past the implementation of the control functions made available by microcontrollers would have involved the use of sizeable computers, but the continued trend towards minimization of size and low cost has meant that micrcontrollers have achieved very low unit costs and low implementation costs. As such the microcontroller is a very flexible electronic ‘tool’ to be applied in a very wide range of product and process applications.

Microcontrollers offer a low cost computing solution. Alternative computing solutions come in many forms, including personal computers and programmable logic controllers. Personal computers (PCs) are used to process high levels of data at high speed, and incorporate microprocessor devices. These devices can be considered a ‘cousin’ of the microcontroller device, but are optimised to manipulate high volumes of data and to provide the facilities for several tasks or windows to operate at any one time. Microprocessor devices are relatively high cost items when compared to microcontrollers because of this high performance capability. In comparison, microcontrollers, as their name suggests, are in general optimised for control applications and not data manipulation. However, the principles and jargon often encountered in PCs are often replicated when considering microcontroller units.

Microcontrollers are found in many common application areas, including domestic appliances such as microwaves, televisions and television remote control units, stereo units, and increasingly in automobiles for engine control, passenger heater unit control, display instrumentation and many other tasks. The widespread availability of microcontrollers is a testament to their flexibility and low unit cost.

A consideration of these application areas also gives some clues as to the potential for their use in any product or process. It is evident that the control of a television or microwave unit does not require the use of the massive processing power found in your PC. A review of the simple control facilities enabled by the use of low cost microcontroller devices provides guidance for your assessment of the use of this technology in your company.

Defining a microcontroller device is not simple, but in general a microcontroller unit may be considered as a computing device offering internal memory and a high level of input and output (I/O) device options. Ideally the use of a microcontroller device minimises the number of external devices used in the system, and integrates as much of the external interfacing to switches, motors or other input / output devices as is practically possible.

This high level of integration means that the unit has low cost, small size and flexibility. These desirable characteristics are available to assist first user companies in improving their products and processes in often dramatic fashion. A consideration of the FUSE world wide web site (http:\www.fuse-network.com) will demonstrate the high returns on investment yielded by the adoption of this technology in a wide range of applications and industrial activities. These examples may encourage you to review your business for opportunities offered by this low cost enabling technology.

Chapter 2: Where can they be used?

As indicated previously, microcontrollers can be used where low cost, flexible and high integration electronic systems are required. Microcontroller solutions have been demonstrated in the FUSE programme to have applications in almost all industrial sectors, and have included application experiments in the textile, food processing, chemical and metal fabrication sectors, amongst many others.

In general microcontroller device options can be used in products where the following needs have to be provided:

- Logging functions (AE2155).
- Control functions (AE 2212))
- Timing functions (AE 25272).
- Display functions (AE2174)
- Sensor information processing (AE 22842)

In addition, micrcontrollers have been demonstrated in certain manufacturing process and machine design applications to provide an alternative design solution to the use of Programmable Logic Controllers (PLCs). PLCs incorporate microcontroller devices but are distributed as complete units requiring no hardware design. However, the fact that at the component level the interconnectivity of the device is already defined limits its applications where control of the hardware interface is required. For example, AE-23779 demonstrates the application of systems design at the component level to replace PLCs with increased system reliability, achieved at lower costs.

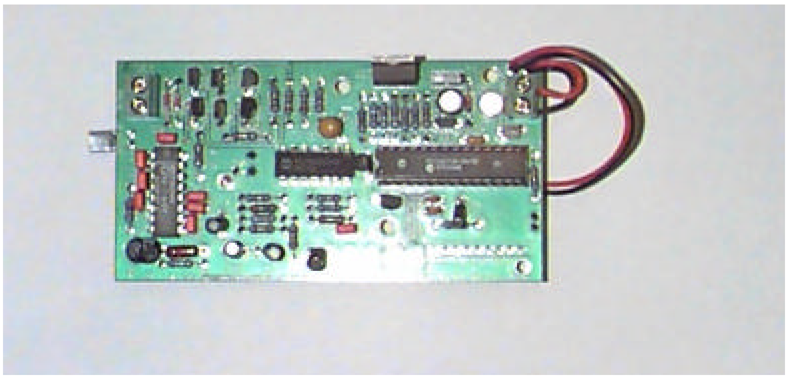
Application experiment AE 26224 also demonstrates the application of the computing facilities offered by microprocessor and microcontroller devices in the design of a vision Assisted Spray machine for agricultural use. This application experiment demonstrates that microprocessor and microcontroller device technology can be applied in machinery subjected to the harshest levels of environmental exposure, and systems subject to high levels of shock, vibration and pollution.

These application examples indicate that the processing capabilities of the microprocessor and microcontroller devices are not limited to applications in normal office type environments. The scope for adopting the technology is therefore very large.

Typical benefits of the adoption of microcontroller technology, as reported in AE2037, would therefore include:

- Reduced Component count.
- Lower production costs.
- Reduced development time for next generation products because of the inherent flexibility in the embedded programs.

- Enhanced operational self test functions.
- Improved production test, by means of internal test algorithms.
- Enhanced functionality.



A Typical Microcontroller Circuit Board

Chapter 3: When should I use them?

Not all electronic implementations require the application of microcontrollers despite this widespread range of applications. Like any other engineering decision, the technology selected is a compromise between costs, performance and volume. Three case studies from the FUSE portfolio illustrate their potential applications and limitations; these case studies are outlined below:

Case Study 1: AE 22643

Problem: This company manufactures thermostat controls for automotive applications, especilally in air conditioning equipment. An improved system was required that allowed the switching temperature to be pre-programmed and signaled the temperature to the automotive control systems in the vehicle. The volumes used for the product would be very high (hundreds of thousands per annum), but problems with customer specific requirements were anticipated.

Technology Solution: Several different component technologies were considered for this application experiment. The criteria used for the identification of the preferred component technology included the following issues:-

- Low cost
- Small size
- Ease of design
- Ease of design change
- Proven technology
- Speed of development
- Ability to change parameters at short notice
- Inexpensive development equipment
- Ability to upgrade/add features, with minor disruption to project.
- Comprehensive product support

Four technologies were considered to be possible candidates for this application experiment; their merits are discussed below.

1. **Application Specific Integrated Circuits** (ASICs) offer a low cost solution if large quantities are envisaged, but were ruled out as they can not easily accommodate change, without incurring significant re-masking charges once production from a foundry has commenced. They also require high non- recurring engineering (NRE) costs and high annual production volumes to be economic, either as semi-custom or full custom.

   ASIC devices however, offer cost advantages when high volumes are anticipated, and the company is aware that such scenarios are likely when the company supplies the device for full production runs in automotive applications. However, the specification of devices for such applications is generally bespoke to that specific application, and differs in combination of load switching levels, temperature setiings, hysteresis levels, and communications protocols. The risk in attmpting to anticipate customer requirements by moving directly to ASIC implementations were therefore too high. The provision of a more flexible technology, utilising microcontroller devices, enables the company to supply aerly prototypes to potential customers for evaluation, and thereafter, to consider investing in ASIC device fabrication. ASIC device technology did not offer a low risk technology option given these commercial circumstances and was therefore avoided for this application experiment.

2. **Discrete solutions** are feasible**,** but the size and cost constraints rule out this approach.

3. **Field Programmable Gate Arrays** and **Programmable Logic Devices** were also ruled out on the grounds of higher device costs than that obtained by the use of microcontrollers. The design methodology compared to the minor amendment of microcontroller code was considered to be relatively inflexible, and therefore this device technology was considered inappropriate for a universal product range. FPGA devices also tend to have high pin out numbers which produced difficulties in delivering a compact packaging solution.

4. **Microcontrollers** offer a low cost solution, a wide selection of suppliers and 4bit, 8bit, & 16bit technology options. The embedded program can be changed reasonably simply, even during production. Volume pricing is attractive with devices costing less than 1 ECU at volumes of 20-50k units per annum. Clearly the choice for this type of project is an embedded Micro controller as it meets all of the criteria mentioned previously, and was selected as the preferred technology.

The improved product employs a widely available 8-bit microprocessor as it’s active element. The selection of the family of microcontroller devices was influenced by the competitive cost of the device, and the potential for transferring the design to a lower cost, pin reduced version in the future without major design amendments. The possibility of using a smaller, more cost-effective unit for future production was crucial. The application experiment uses the currently series of devices as it was judged as being the most suitable unit as it is widely supported, documented and has a proven track record.

Features of the selected Microcontroller that are of particular interest are as follows:

- 12 programmable I/O pins.
- 8 bit real time clock/counter with 8 bit prescaler.
- Wide voltage operating range 2.5v to 6.25v
- One time programmable parts available (OTP’s)

The program development route utilised two main approaches:

1. The use of a low cost prototyping board freely commercially available to prototype peripheral circuitry, and to develop some low complexity software routines in the initial devlopment phase,
2. The use of a full in circuit emulator equipment to develop the final software, and to integrate this software into the final circuit board.

The detailed development route is defined in the work plan section of this demonstrator document.

The program was written in assembly language using the assmbler and simulator tools supplied by the device manufacturer. Assembly language programming was selected because the complexity of the program was considered low, and to ensure that the minimum memory usage was achieved.

Case Study 2 :   AE 24550

Problem: A company producing data logger systems (AE 24550) required the capability to capture not only analogue signals from transducers but to capture digital signals (for example pulse trains from shaft encoders etc.). The company had microcontroller design experience and required at least 16 channels of high speed digital pulse tracking at the product's input.

Technology solution: This solution requires high speed, parallel signal processing.

The technologies that were considered to add frequency, interval, counts and were as follows:

Technology	Speed	Versatility	Cost	Parts Count	Software i/f	Hardware i/f	power	First User
Microprocessor h/w	mod	poor	mod	4-8+	good	poor unless dual port	mod	no
Microprocessor s/w	poor	good	good	1	good	n/a	n/a	no
Microprocessor Peripherals	good	poor	mod	5-6	good	good	mod	no
Single Risc e.g. PICs)	slow	mod	good	1	good	poor	good	no
Multiple Risc e.g. PICs)	mod	mod	poor	16	good	poor	good	no
Many Discrete ICs	good	good	poor	150	good	good	good	no
Fewer Discrete ICs	good	poor	mod	50	good	good	good	no
Analogue Solutions	good	bad	bad	32	n/a	n/a	mod	no
DSP	mod	mod	mod	1-4	good	mod	poor	yes
Digital ASIC	good	good	prohibitive	1	good	good	good	yes
PLDs	good	good	good	4-8	mod	good	mod	yes
FPGAs	good	good	good	1	good	good	mod	yes

<u>Technologies Considered for Component</u>

Discrete devices - allow the parallel processing of all of the input channels to the logger equipment, but offers a complex and expensive solution in terms of assembly, testing and repair times. The design of an 8 channel frequency and time interval measurement circuit for the equipment has been estimated to require 24 integrated circuits (ICs), including some 28 pin crosspoint switches. The design is also limited in flexibility by measuring frequency as counts over a fixed interval (nominally one second). The expansion of this discrete design solution to provide 16 and 32 channel solutions is clearly impractical, and the use of discrete logic devices has been rejected because of its inflexibility, large board space, power requirements, and high cost.

Microcontroller and Microprocessor Technology - The use of a microprocessor peripheral device, such as the HD63B40, has been previously implemented as a method of achieving the frequency measurement requirement. This requires 3 ICs for implementing 3 channels of frequency measurement, and modifications to the Chartlogger enclosure and operational software. These modifications are costly. This solution is also an impractical means for extension to meet demands for 16 or 32 channel measurement requirements, and does not provide time interval capabilities.

The use of a stand alone Microcontroller, such as the Mitsubishi's M37732, to provide frequency measurement on 16 channels was also evaluated. This solution however limits the resolution of the measurement to 8 bits because of the configuration of the internal timers. Greater resolution requires more Microcontrollers. Unrestricted access by the logger's central processor is also not possible, and the latency introduced by the need to request data transfer is unacceptable. Additionally, the resetting of the channels frequency or time interval counter after reading would have to be accomplished by the Microcontroller. The sequencing of channel reset intervals will not enable customer requirements for simultaneous time interval measurements of all channels to be achieved.

DSPs (Digital Signal Processing)- offer little advantage over RISC processors like PICs and seem an over complex and costly approach to the task at hand. They are essentially sequential processors and have difficulty processing fast parallel streams of data. Multiple devices with EPROMs and support logic would probably have been needed.
The use of Microcontroller and microprocessor devices has therefore been rejected because it does not offer a practical method of implementing the desired features.

PLD Technology - Small and medium-size PLDs can be used to implement the proposed improvements. However, such an implementation will result in a solution with similar disadvantages to discrete device implementation. A large number of PLDs will be required, thus resulting in a larger PCB and the need to modify the enclosure of the equipment. Furthermore, this solution will unacceptably increase the cost of the components, assembly and testing.

Complex PLDs (CPLDs) from different manufacturers were investigated as possible routes for implementing the system. These manufacturers included Altera, Xilinx, Lattice and AMD. The implementation of the proposed system requires a large number of sequential elements (flip-flops) to realise the counters and registers. Unfortunately, it was not possible to find a single CPLD that will accommodate the high number of flip-flops required for the system. In addition, implementing this type of system with CPLDs is expensive due to wasted resources in the combinational sections of these devices. The use of multiple CPLDs was rejected due to the cost of large CPLDs, their power consumption and PCB area required to accommodate these devices.

Digital ASIC Technology -The annual sales volume of equipment has been in the range of 50 to 100 units per annum since the launch of the equipment. These relatively low volumes, even with the increased sales resulting from the improvements, do not justify the investment in the design costs and foundry costs associated with the production of an application specific device. In addition, the ASIC unit cost for such volume can exceed 200 ECUs. The technology increment for Pi Logic also represents a substantial risk. The use of ASIC devices was therefore rejected on cost and risk grounds.

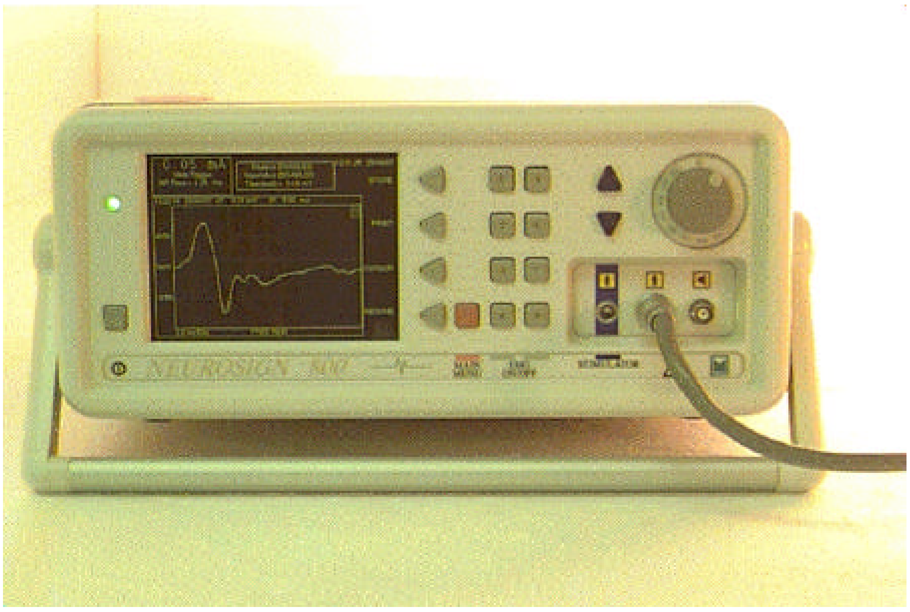
FPGA Technology - FPGAs are highly suited for the implementation of the proposed system because of their register-rich architectures. In addition, most FPGA architectures have sufficient combinational elements to cover the requirements of the proposed design.

In-circuit-programmability pointed towards the use of SRAM-based FPGA technologies,.

After the FU attended a number of seminars by FPGA manufacturers, listened and talked Device and software evolution is very fast in the FPGA area with increasing device size, falling costs and device geometries. The software and devices targeted at the start of the AE have both changed throughout the course of the AE. This was partly due to a break in AE work due to other work pressures.

Case Study 3: AE 2174

Problem: A medical equipment company wished to introduce multiple channel monitoring capabilities and improved display capabilities to its nerve function monitoring equipment. The display requirements are indicated in the following photograph of the completed product.



Photograph of Improved Equipment Display Functions (Neurosign®)

Technical Solution: The requirement to display the waveforms could have been met by two different methods. The first of these is the use of a software driven device, such as a microprocessor or microcontroller, in conjunction with a bitmapped display device such as a raster scanned CRT, or one of the flat panel display technologies (LCD, Plasma, Electroluminescent). The alternative method would utilise hardware alone, and is similar to the techniques used in many storage oscilloscopes. The data is digitised, and stored in a memory buffer. The display is generated on a vector CRT, by clocking the data out of the memory buffer, via a digital to analogue coverter, to drive the CRT's "Y" deflection amplifier whilst the electron beam is scanned across the screen in the "X" direction.

The former method was chosen for the improved product, for the following reasons:

1) The use of a processor offers the greatest flexibility for the presentation of the data. For example it will be relatively straightforward to compare the current waveform with a reference waveform stored earlier.
2) Text and cursors are relatively easy to incorporate on the screen, allowing the measurement of amplitude and latency.
3) The cost of implementing the processor based solution is much lower than that for the hardware solution.
4) The use of a processor brings with it easy, random access to memory for the storage of waveform data.
5) The system will allow for future enhancements, such as the incorporation of digital signal processing to automatically detect amplitude and latency.

The choice of display technology was influenced, primarily by “viewability”, but also by flexibility, cost and size. CRTs were discounted, because of the relatively high volume that they require, and because small displays (150cm, diagonal) are relatively expensive. LCDs were discounted because they are not suitable for viewing from a variety of angles, or from any distance. Plasma displays are expensive, so this left electroluminescent panel displays as the most cost effective choice.

The choice of a suitable processor was more difficult as there is a vast array of devices available, ranging from simple, more or less self contained microcontrollers, 8, 16 and 32 bit microprocessors and digital signal processors.

Microcontrollers were considered, particularly as they come equipped with useful built in peripherals, such as serial ports, timers etc. However, they were ultimately discounted because the system requires a fairly wide address space in order to interface the data storage RAM. True microcontrollers are really intended to use their own ‘on chip’ memory. Many can be configured to operate in microprocessor mode, using external memory, but when this is done access to other on chip peripherals is compromised as the I/O pins used are shared between the two functions.

Digital signal processors were also rejected because they are highly optimised for the processing of digital signal data. Whilst this would have been extremely convenient for handling the large amount of EMG data, it would have made interfacing to a display quite cumbersome and awkward. However, the primary purpose of incorporating a processor into the Neurosign® was to add a display, so a more general purpose device was considered to be a better choice.

16 bit sampling is used to digitise the EMG signals, in order to maintain adequate resolution over the dynamic range (approximately 200:1). If an 8 bit processor were to be used, two read operations would be required for every sample digitised, whereas a 16 bit device would require just one. If the data rate was relatively low, this would not present a problem, but faithful reproduction of the EMG signal requires a bandwidth of at least 5kHz (the Neurosign 100 has a bandwidth of 10kHz). A data sampling rate of 10kHz gives a maximum bandwidth of 5kHz, by Nyquist, which is a reasonable compromise. There are 8 channels of data, so a 16 bit microprocessor would have to perform 80,000 read operations per second, which would increase to 160,000 for an 8 bit device. This poses a significant overhead, so a 16 bit microprocessor seemed the most appropriate choice, as the overhead is halved. Furthermore, 16 bit devices are designed to handle data a word at a time, rather than a byte at a time, making any further processing of the data easier in a 16 bit part.

The selection of 16 bit microprocessors currently available is still bewildering. The ideal device would allow simple interfacing with external memory, yet would incorporate integrated peripherals (DMA, serial I/O, timers, etc), to minimise the complexity and cost of the final design. The final choice was Hitachi’s H8/300 which incorporates memory management and addressing on chip, as well as DMA, asynchronous and synchronous serial ports, parallel I/O ports and programmable timers.

The program was written in the ‘C’ language. Assembly language was also used for a few sections of the program where speed of execution was critical. The choice of a sufficiently powerful microprocessor meant that this was only necessary in relatively few instances. The code was developed on a PC using Hitachi’s development tools (‘C’ cross-compiler) linked to a ROM emulator, which allowed the program to be downloaded to the prototype for testing. All of the software (including the specifications) was controlled under the MKS version control system.

Chapter 4: What does it cost?

A major barrier to company's considering microcontroller technology development for the first time is the financial barrier. This is in fact not present as low cost development solutions are easily available. As an example, consider the lesson learned by one first user company (AE 22643):

"A major concern for most companies in undertaking any new technology development is the initial start up costs. The initial feasibility study provided reassurance in this regard, and the eventual start up costs for the development of a microcontroller support CAD system were much lower than initially feared. A basic development environment was established for approximately 2 KECU; a breakdown of approximate costs is shown below:

Equipment	Cost
120Mhz Pentium PC	700 ECU
In-circuit Emulator	1,050 ECU
Development System*	free of charge
Microcontroller Demonstration board	95 ECU
Stag U-V Device Eraser	130 ECU
Kanda Training Aid	140 ECU

- Including a software editor, assembler, compiler, and simulator downloaded from the supplier's Web site."

These costs (less than 3 KECU) is not a major barrier to any first user. Similarly, the first user in AE 2212 invested a relatively modest1,200 ECU for an in-circuit emulator and C compiler for a reliable and powerful development facility.

The other cost to be considered is that of unit costs of the microcontroller devices themselves. These again are related to performance, and can be obtained easily enough by contacting distributors directly. Simple, low cost microcontrollers (512 Bytes memory) in order quantities of 1,000 units+ will typically cost less than 2 ECU each. More complicated microcontrollers (generally with a high number of external I/O options and typically 64KBytes of memory) may coat in the region of 50 ECU in these quantities. The high level of integration offered by the microcontrollers mean that total circuit board component costs do not typically exceed the microcontroller cost by more than a factor of 5x. This indicates the circuit board based on microcontrollers can be a very low cost item.

Production cost estimation accuracy can be extremely accurate for microcontroller device applications even at the feasibility stage of the application experiment. AE 2212 demonstrated a 2% cost variance from 117 ECU predicted to 120 ECU product costs. In this specific instance, the accuracy of the final production costs has been highly significant in that existing customers of the company which were sceptical about the market exploitation of the technology on the grounds of commercial viability based on previous overruns in cost estimates from other suppliers. The company management and customers are now enthusiasts of the potential of this new technology.

Chapter 5: Selecting a Microcontroller Device

Because a microcontroller device is a computing device the selection of the optimum microcontroller device is often as confusing as considering the range of PCs offered by companies in your high street computer sales room. The key to selecting the optimum device is understanding the jargon, and establishing priorities. The most important characteristics of selecting PCs is the memory available, the speed of the processor, and its interface features. This is no different for microcontrollers. Memory capacity is segregated into program space and scratchpad, temporary space; these are termed ROM (read only memory) and RAM (random access memory) respectively.

Technical and support requirements define the selection of microcontrollers.

The most important technical selection criteria is ensuring your selected device has enough memory. Without adequate memory (especially ROM memory) your program control solutions will not function at all!

Program memories can be internal to the device or external to the device. Obviously internal memory offers the most compact design solution, but limits memory size. External memory allows more memory to be used but requires additional inputs and outputs on the microcontroller to physically connect itself to these devices.

Internal program memory (ROM memory) can vary in size between the limits of 512 Bytes to 64KBytes, typically. To scale these values into meaningful values, the use of a 512 byte program device would be adequate for the simplest control systems (for example, sensing, processing and controlling one interface), whereas the 64KByte memory device would enable multiple, complex control to be provided with a reasonably simple LCD (liquid crystal display) interface. In general, memory size is related to product cost. Therefore the minimum size microcontroller in term of memory capacity should be uses. However, sizing the program size is not an easy process, and requires some experience. The use of an experienced subcontractor can miminise the uncertainty at this stage; the use of the FUSE demonstrator documents also provides guidelines for selection.

Other technical selection criteria include speed and I/O capability.

Processing speed is not to be measured in the number of MHHZ quoted by the supplier as the ‘clock speed’. This is a guide, but other factors, such as the number of instructions required to perform a time delay, addition or other common instruction may be important. RISC (reduced instruction set complexity) based microcontrollers may often perform simple tasks more effectively than other microcontrollers event though they may be operated at slightly lower clock speeds.

I / O capability is easier to assess. A wide range of I/O options are available, including:

- Analogue inputs (typically 8 bit or 10 bit (<1% accuracy) resolutions).

- Analogue outputs.
- Parallel digital outputs, sometimes capable of directly driving LED (light emitting diodes) with < 15mA output capabilities.
- Parallel digital inputs.
- Timer inputs, for use in event or interval counting.
- Timer outputs, often used in PWM (pulse width motor) control.
- Serial communications, for example to interface via RS232 interfaces to PCs.
- Display driver facilities. Although less frequent some microcontroller units integrate LCD drivers into the device.

Analysing the I/O requirements of a design application is often simpler than estimating program size. A simple list based on the number of physical devices connected to the controller often allows an accurate assessment of microcontroller I/O to be made. For example, itemise the number of sensors (1 analogue input per sensor?), discrete switches (one digital input per switch) etc.

If the total number of inputs and outputs exceeds the available I/O availability of the device then *multiplexed* solutions may be used (this means using the microcontroller to ‘address’ or select each I/O independently of its neighbours), but this requires additional hardware. This compromise experience to evaluate, and guidance and training from a subcontractor at this stage may be useful.

Having conducted this evaluation process, one further complexity arises; the selection of device options. Options include:

- ***One time programmable (OTP) devices:*** devices programmed by the user in the production environment on a once only basis. This device option was used for AE 2212 because the company had an annual production level of 1,000 units per annum.
- ***Re-programmable devices:*** These were generally based on windowed UV or electrically erasable devices, and were very convenient for prototyping and trialling designs.
- ***Flash devices:*** These are a variation of re-programmable devices, now increasingly becoming popular. They are often in-circuit programmable.
- ***Masked (ROM) devices***: these are devices with the users code physically incorporated into the device by the device manufacturer. This offers lower costs in high volumes.

It is generally advised to ensure a re-programmable option to support the development of prototype units before conversion to OTP devices. The availability of these devices should be investigated.

Finally, most devices work from a 5 volt supply. Alternative supplies (for example, 3V) will considerably narrow the search.

Equally important to these technical evaluations is the selection of a device with a wide range of development support options. You should consider asking the following questions:

- Can I obtain help when I have problems? Easily?
- What development tools are there and how much do they cost?
- What documentary support is available? (manuals, application notes, books etc.)

A review of a range of electronic component distributor catalogues can provide a good indication as to the answer to this question. Reviewing these sources will provide the answer as to the most popular device families in your area.

Development Support Facilities

The range of development support tools is equally wide. This section briefly overviews these.

i. ***Language support tools****:* The most general division is between assembly and high level language programming. Assembly programming involves operation at the lowest programming level possible, and requires detailed knowledge of the device. Assembly language is often used where code minimisation or speed maximisation is critical. Assemblers are often provided free of charge by the device manufacturer. High level language programming (often C) requires the use of a compiler to translate the more readable and structured program constructs produced by the engineer into microcontroller readable form. C programs allow reusability, and advantages in debugging your system, but the compilers are often produced by third party suppliers and as such can cost between 250 and 5,000 ECU depending on the features provided.

An increasingly popular route is the use of interpreters to program the device. These interpreters allow programs produced in BASIC, Forth or other common languages to be executed by the microcontroller. These interpreters are programs that reside in the microcontroller, and allow the programmer to easily trial the programs produced.

ii. ***Development Tools****:* A wide variety of options exist. These include:

Simulators: A program that runs your code on a PC, and predicts the outcome of the program. The outputs are displayed as memory, input, output and internal memory values on a window based program. The simulation allows the logic of the program to be checked, but cannot simulate hardware interface and timing factors. Usually, a simulator is built into the assembler or compiler, and is often included in the cost of those programs.

ROM Emulators: This is a form of a simulator, but is a hardware unit that plugs into the circuit board to be developed, often, only if, an external program memory device location is available. It therefore allows both the logic and the physical interface to be tested, but not at full speed. Typically, an emulator will cost in the range of 50 ECU to 500ECU.

Development Boards: A range of development boards with an integral microcontroller and ‘breadboarding’, prototyping area are generally available. These boards often include resident debuggers, which allow the prototyping board to communicate information for debugging purposes to a host PC. These boards are relatively inexpensive, and may range from 50 ECU upwards.

Emulators: These units are the top end debugging tools, and allow complex real time evaluations to be conducted. Because of their high performance they cost more than other development tools (typically 3,000 ECU or more), but may be essential if intractable problems are encountered. Hire options could also be investigated in this case.

The selection of a suitable development tool will be related to the available budget. Low complexity systems can be developed using low cost development boards and assemblers; more complex tasks require emulator of some form.

Finally, production programming devices should be identified for the selected microcontroller. If high volumes are to be manufactured then ‘gang’ programmer devices capable of programming multiple devices a tone time should be considered.

Putting it all together

Summarising all of this technical information is fraught with the danger of generalisations, but if:

1. The complexity of the product is very low (for example, a simple timer) then a low range microcontroller (512 Bytes ROM) assembly programming is adequate, and a low cost development facility should be adequate. For example refer to AE22843.
2. The complexity of the product is low (for example, a microwave unit controller) a low range microcontroller with integral memory (2-4 Kbytes ROM) may be adequate. In this case development using either assemble of C compilers may be equally valid, and mid range cost development tools may be needed (for example, a good ROM emulator). For example refer to AE 23565.
3. The system has reasonable complexity, defined by complex operator interfaces of many hardware interfaces. This will require a higher specification microcontroller with say 8K to 64 K Bytes of ROM memory, and would probably require a good ROM emulator of full Emulator system for development. For example, refer to AE 2174.

The first use of microcontroller technology obviously introduces many concerns. These can be overcome using a subcontractor to provide advice and guidance. The selections of the subcontract support requirements are defined in the next chapter.

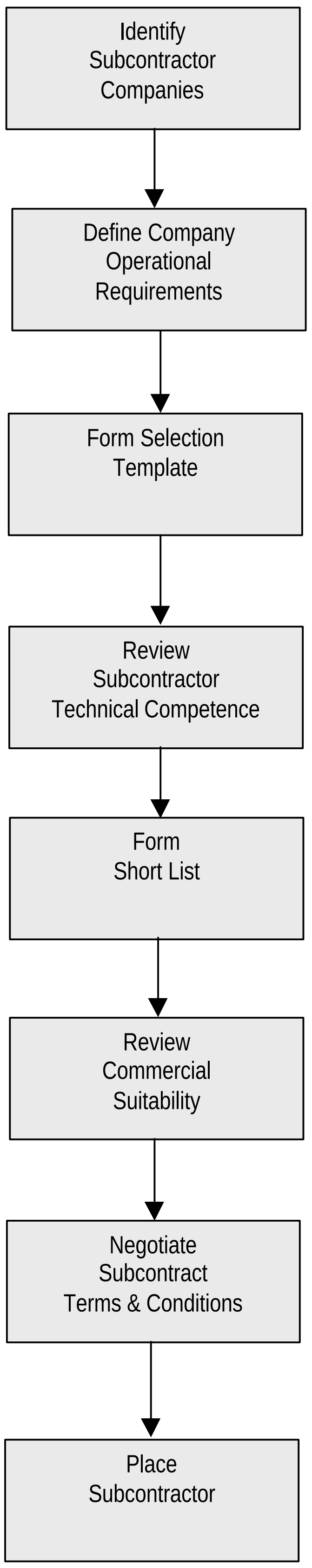
Chapter 6: Selecting and Working with Subcontractors

One of the most important decisions to be undertaken before any development project is completed concerns the identification, selection and negotiations with subcontractors. There is no one simple template to apply in identifying a 'suitable' subcontractor; rather the template used to match candidates for the subcontractor role must be developed by the company based on an analysis of its own requirements.

The process of identifying potential design subcontractors is relatively simple in comparison to the evaluation of their suitability. The identification process can be undertaken in the following manner:

- Use of trade directories.
- Use of local authority company databases.
- Use of chartered engineer institutions.
- Use of local Universities and research bodies.
- Use of local business clubs and collaborative organisations.
- Personal references.

Developing the specification or template used for selecting the best subcontractor from those identified will be partially based on the company's analysis of the extent of the knowledge development process it requires, and the rate at which this knowledge transfer is to occur. The extent of knowledge transfer to be achieved and the time required for the project to be accomplished over are directly related. In addition, more in depth knowledge requires additional training effort and company personnel time, both of which increases the project costs.


Identify Subcontractor Companies
Define Company Operational Requirements
Form Selection Template
Review Subcontractor Technical Competence
Form Short List
Review Commercial Suitability
Negotiate Subcontract Terms & Conditions
Place Subcontractor

Subcontractor Selection Process

Specific examples of how companies have attempted to manage the knowledge development process and subcontractor selection include:

AE 22843

This company required that the company engineer acquires a complete range of microcontroller development skills, but recognised that time to assimilate this knowledge was required. Subcontractors were therefore required for initial training at the beginning of the project, and at specified intervals throughout the application experiment to work on site with the company engineer. In the intervening intervals the company engineer continued the project development, experimenting with the technology to develop through unaided practice the skills taught by the subcontractor. In these intervening periods help line support to the subcontractor was required.

Given this scenario, the subcontractor selection was not influenced by geographical proximity or detailed knowledge of the company's product range or markets. However, major influences were flexibility in terms of scheduling consultant attendance at defined, and possibly moveable, periods of the project, remote access via telephone, fax and e-mail, and the commonality of development tools at both parties.

AE 2155

This small one person company decided to undertake the complete project, with remote assistance from a subcontractor to provide advice and guidance only. As the company was very small, and reliant on servicing key customer orders in a very responsive manner, it was considered unlikely that the company could identify accurately slots in its calendar for the development task. This made traditional operations with subcontractor companies difficult to schedule. The subcontract requirements in this case were therefore, determined by continuation of support over an elongated development period, and assessments of subcontractor interest in supporting the company for many very short intervals over a long project. This company identified a local University's electronic design consultants as suitable candidates to supply the required technical support.

AE 2174

This company manufactures medical equipment often used in surgical procedures. As such its development programme was highly influenced by the need to consider risk, medical liability issues, regulatory requirements and related documentary needs. The company concluded that it was not possible to undertake such a development project without the use of a subcontractor who was aware of the medical product development requirements, and who could attend the company's premises over the complete development period so as to ensure an integrated development team structure. This latter requirement clearly demanded a local consultant to act as a subcontractor.

These examples usefully identify characteristics of the subcontractor, for example location, commercial outlook and product knowledge, that are specifically related to how a company plans to allocate its personnel to the project. Companies considering adopting microcontroller technology for the first time should therefore define answers to the following questions in devising its concept of the ideal subcontractor:

- Can an engineer be allocated full time to the development activity ? If not, what proportion of the time can be made available for this engineer to work full time on the project (for example, 2 days per week)?
- Does the subcontractor need to work in the company, for example, to liaise with other staff on a regular basis, to use internal test resources, or to ensure legal requirements are met?
- Does the subcontractor need to know about the company's product application areas, and are there specific product related regulations the subcontractor must possess?
- Does the company wish to acquire all of the microcontroller development skills or can some related skills, for example circuit board layout, be considered services that will always be outsourced?

Once the 'modus operandii' of the development project is determined the company must thereafter consider the technical and commercial suitability of its subcontractors it identifies.

Technical evaluation of a subcontractor's competence is difficult for a first user company, especially if that company has no prior microelectronic experience. Reliance on verbal assurances and assessments should be avoided; evidence should be made available by the subcontractor of:

1. Previous experience with the selected microcontroller family of devices.
2. Previous experience of applying the selected device technology in similar types of applications to that proposed (for example, control processes, high-speed data processing etc.).
3. Previous experience of operating with companies of a similar size, and client lists.

In addition the company should review the company facilities and equipment base to ensure that the subcontractor is capable of undertaking all of the required work.

Best practice indications are that the pursuit of references from the client list is worthwhile. This avenue allows the calibration of the company's assessments against other companies practical experiences, and therefore adds value to the assessment process.

The commercial suitability of the two partners in the subcontract process is crucial to the smooth operation of the development programme. The company should not rely on legal contractual clauses alone. Whilst these are important, the enforcement of the contract through the courts can cause significant delays and costs. The commercial suitability of the company and subcontractor is often determined by the following factors:

- Similar size (refer to AE 24667 Pentwyn Splicers)
- Clear understandings of the staged payment profiles, and the deliverables required to achieve this. This is essential to avoid the potential for disputes (refer to AE 23565 C&G Systems).
- The unambiguous identification of personnel assigned to the programme of work by the subcontractor; this is especially important for the person responsible for subcontract management at the subcontractor.
- Personal rapport and confidence developed during the contract negotiation process. This process is critical in developing an understanding of the viewpoints of the subcontractor. These viewpoints can be assessed against the company’s expectations and used to form an assessment of the compatibility of commercial approaches.
- Discussions on follow on options after the development project is concluded. It is useful to decide if the subcontractor

Chapter 7: Subcontract Preparation

The importance of the subcontract in clarifying the relative responsibilities of both parties is not to be underestimated. However, be aware that the existence of a subcontract in itself is no guarantee of delivery, and the enforcement of contract terms and conditions are costly and time consuming. The importance of developing a commercial understanding is crucial, and the establishment of an open personal relationship between the organisations is a critical factor in maintaining schedules.

The content of a subcontract is obviously the subject of a legal assessment by the company's solicitors, ***and this document does not offer legal advice.*** However, the issues to be considered in structuring the technical content of the subcontract should include (but may not be limited to) those detailed below:

1. Scope

This section document defines in general terms the scope of work to be conducted by the subcontractor. It should define the two parties involved, an overview of the services to be undertaken, and the general outputs to be produced.

1.1 Related Documents

The technical requirements for this subcontract will be normally be defined in other documents, as the requirements of these documents would normally be too detailed to include in an all-embracing contract. For convenience these documents are normally referenced in the contract by Title, Number, Issue and Date information.

These documents may include the following documents:

- The Product Requirements Specification.
- The Acceptance Test Requirement Specification.
- The System Test Requirements Specification.

These documents will be considered contractual obligations. As such they should not change unless agreed by both parties. The contract may therefore make reference toan updating process; for example new official issue number and date, and countersignature by both parties.

2. Services to be Provided

This section should define in detail the services to be provided , and may include the following services:

i. Training in the selected microcontroller; such training to include hardware and software design and development guidance and advice. Training support, consisting of advice, guidance and technical review services. The number of days required for this task may be stated.

ii. Technical advice on system design issues.

iii. Design and development activities.

iv. Prototype manufacture, assembly and evaluation.

v. The delivery of all supporting information required to manufacture the circuit board.

vi. Attendance at progress review regular meetings.

3 Company Responsibilities

This optional section may b included to limit the company's responsibilities or to clarify the interface between the company and the subcontractor.

Examples might include the provision of the following:

- The Product Requirements Specification.
- The System Test Requirements Specification.
- Representative test jigs.
- Drawings and related documentation defining the sizes and positions of the electronic modules.
- Access to such information required by the subcontractor to discharge their responsibilities.
- Subsequent evaluations and document preparation as required to obtain CE accreditation for the product.

4. Deliverables

This section of the subcontract should define the deliverables required from the subcontractor in detail. Examples of deliverables might include:

a) A specified number of prototype units.

b) Circuit schematics, layout data, components lists, and all such other information as is required to enable third party manufacture of the circuit board(s).

c) Software code listings, design documentation, test software, source files, and all such other information as is required to allow the software to be maintained by a third party.

The deliverable format should be specified by the company, and would normally include both paper hard copies and / or computer disk files.

4.1 Delivery Schedule

The supply of the defined deliverables will be conducted to a time schedule defined by the company and agreed with the subcontractor in this section.

Amendments to the time schedule could be controlled by both parties agreeing to an update mechanism; for example, the signature of both parties to a dated and issue controlled schedule document .

5. Acceptance Criteria

This section defines how the company accepts the subcontractor's responsibilities under this contract to have been completed. It will define criteria such as :

1. Successful system performance demonstrated by compliance with an agreed System Test Requirements Specification.
2. Delivery of all deliverables and documentation.

6. Payment

A payment schedule against deliverables should be defined. The value of the deliverables are subject to negotiation, but the company should retain a sizeable retention (for example, 25% until all acceptance criteria have been met).

Fixed price contracts are preferable for first users as the financial obligations are known.

Ensure useful deliverables are provided at each stage. For example, if software programs are demonstrated ensure the documented code is made available on disk, or layout and component data is available when a circuit board is demonstrated. This allows subcontractors to be changed in the event of subsequent non-performance.

7. Ownwership and Intellectual Property Rights

The ownership of all design, copyright and intellectual property rights embodied in all the designs, including (but not limited to) the circuit board designs, and software code and designs must be defined. The company would need ownership of these in most cases. Note as an example, if ownership is not established the subcontractor could charge a licence fee in future for the use of their software! This situation is to be avoided.

8. Confidentiality

The company must define the obligations of confidentiality on the subcontractor. If this is not done competitors might gain information on product or commercial details through this source.

The period over which this confidentiality obligation is to exist should also be defined (e.g. 5 years).

9. Termination

This section should define each party’s obligations in the event of termination. For example, the methods of termination, notice of termination, the discharge of financial obligations entered into, etc.

Chapter 8: Project Planning, Management and Operation.

Microcontroller development projects operated under the FUSE programme have typically taken between 6 months and 12 months to complete. Situational factors, such as staff reallocations, may influence the completion of the project within these time frames, and these project duration assume full time staff allocation.

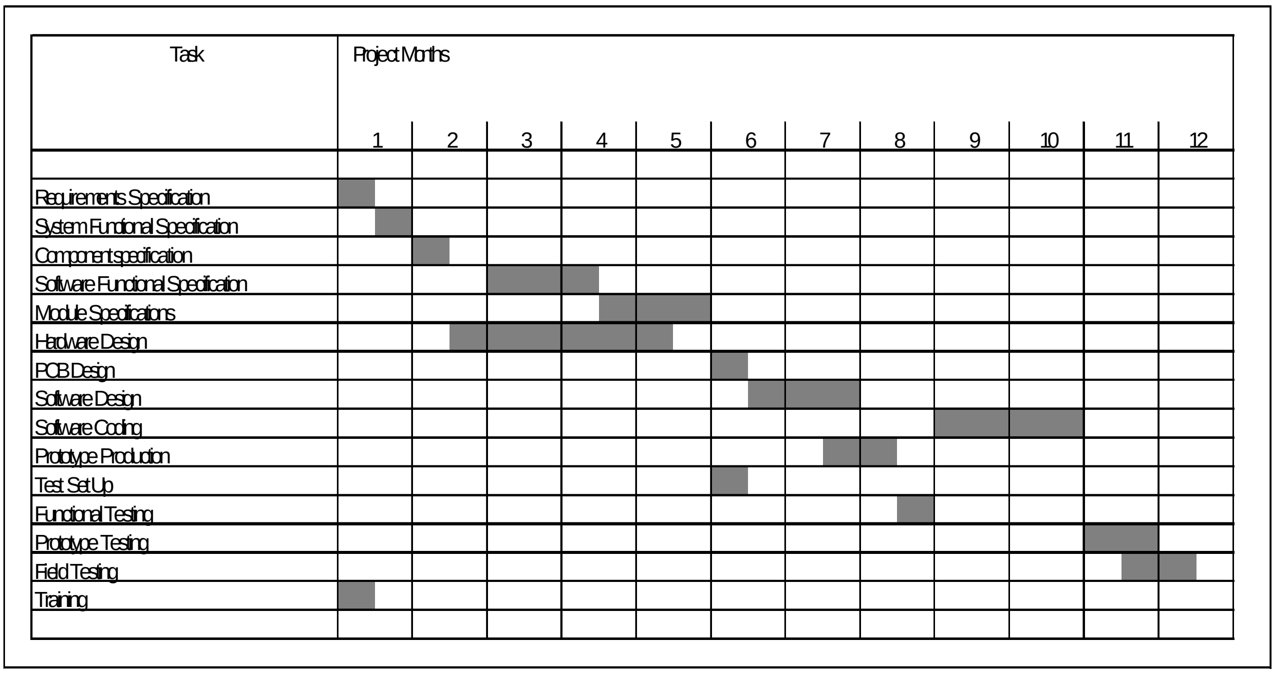
In determining whether a new microcontroller development programme is to be at the shortest end or the longest end of this scale, the first user should consider as the single most important factor the complexity of the task to be undertaken. Complexity can be judged in the number of different tasks required for the improved product; for example, indicators of complexity might include:

1. Display complexity – a display with menu options and responses taking much more time to develop than a simple LED display.
2. Control complexity – is the control function related to the measurement of several inputs or only one.
3. Interface Complexity – the number of interfaces controlled by the microcontroller device influences the complexity of the hardware design and the software design.

As indicators of the impact of complexity on the timescales of a microcontroller product development consider the programmes adopted for AE 2174 and AE 23565 shown on the following page. The most significant differences in these two AEs is the complexity of the products, viz:

- AE 2174 proposes a full computer type display, operator menus and complex hardware .
- AE 23565 proposes only an LED indicator, one interface via an Infra Red link, and relatively simple hardware.

The complexity issue resulted in an increase in company resources from 113 days (AE 23565) to 239 person days (AE 2174).



Actual Workplan Adopted for the AE 2174

Task	**Company Actual Effort (person days)**	**Cost of subcontractor (KECU)**
Technical Management	24	1.14
Training	4	1.0
Design	59	3.41
Verification	98	6.82
Test & Final Report	54	1.82
Total	239	14.19

Resources Used in the Prototype Unit Development

<u>AE 2174 Work Plan Information</u>

	ELAPSED WEEKS																					
DESCRIPTION	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
	▨																					
Training	■																					
		▨	▨	▨																		
Design Specification		■	■																			
					▨	▨	▨	▨														
Hardware Design				■	■	■	■	■														
									▨	▨	▨	▨										
Software Design									■	■	■	■	■									
													▨	▨	▨	▨						
Hardware Build & Test														■	■	■						
																	▨	▨				
Hardware / Software Integration																	■	■				
																			▨	▨	▨	▨
Functional Test																			■	■	■	■

Plan ▨
Actual ■

Planned and Actual Workplan AE 23565

Task	**Company Effort (days)**	**Subcontractor Costs (KECU)**	**Planned Company Effort (days)**
Management	15		10
Training	6	1.0	5
Specification	22	1.1	20
Hardware & Software Design	28	6.1	38
Testing & Design refinements	42	2.4	34
TOTAL	113	12.6	107

Resources Used in the Prototype Unit Development

<u>AE 23565 Work Plan Information</u>

Work Plan Elements

Microcontroller work plan elements will vary from project to project in several ways; however, the general structure of these work plans remains essentially the same. This generic workplan consists of the following steps (reference AE 23600):

Engineer Training: This includes attendance at training courses designed to enable the first user to gain experience of designing with a microcontroller, and also programming and debugging the microcontroller software.

System Specification: This task confirms the features required by marketing, and formally defines the requirements of the microcontroller-based product <u>in detail.</u> Operational sequences, displays and interfaces are all defined unambiguously in this document.

Functional Test Specification)*:* This task involves formally defining the tests required to demonstrate that the product complies with the requirements specification.

Systems Design: This task involves the high level definition of the system implementation, including the allocation of functionality to the microcontroller program or hardware solutions, and defines in detail the major components to be used, including the selection of the microcontroller.

Circuit Board Design*:* This task involved producing a hardware design based on the selected microcontroller. The task involves the generation of schematic diagrams and the production of circuit board layout data for the procurement of the circuit board. This circuit board is the basis for the hardware development to proceed.

Hardware Build and Test*:* The assembly of the circuit board, and the initial testing of the circuit board. This testing usually requires some simple microcontroller software routines to be written to exercise the hardware.

Hardware Modifications: This task involved making any hardware changes found necessary after the initial hardware tests, to produce a revised circuit board schematic diagram and layout

Software Specification: This task involves developing the Software Functional Specification, defining the hardware/software interface and defining the memory allocations for various data in the microcontroller.

Software Design*:* This task involves the first stage of designing the software for the microcontroller, including establishing the software structure and identifying the content of software algorithms required.

Software Coding*:* This task involves code development (writing the program), and verifying the program in accordance with the Software Functional Specification.

First Stage Integration*:* This task results in the integration of the developed software into the hardware board, and usually involves some design support tools to identify and eliminate interface problems, gross code errors or microcontroller hardware configuration errors. This task can be the most complex stage of the process, requiring a good understanding of the hardware design, software design and CAD support tools to achieve a satisfactory outcome.

Functional Test: This task involves the testing of the final product against the Functional Test Specification.

Software Modification*:* This task involves software changes to amend the software design as a result of the functional or integration testing.

.

Project Management and Operation

Projects involving software development are notoriously difficult to manage, and often over run because the invisible nature of the software product means that errors are not easily detected at the early stage of development unless a rigorous approach is adopted.

The project management approach adopted by Magstim (AE 2174) is advised as a best practice model for the management of the project, and provides a clear structure for the operation of the project. Extracts from that demonstrator are reproduced below:

“The rationale behind the plan was to focus on planning, specification and system design, so that the functional requirements, implementation details, interactions and interdependencies of each of the hardware and software components were defined. Once this had been achieved, the implementation could be expected to be a straightforward “handle turning” exercise, and would avoid the unforeseen interactions that are the most common cause of software bugs. Furthermore, the chances of having to make compromises on the system performance late in the project due to underestimating the system requirements, would be minimised.

Project Management was an ongoing task throughout the application experiment. Frequent review meetings were held to assess the progress of the project and to ensure that it remained on track. Subcontractor input to the technical management of the programme was substantial, and involved the definition of the philosophy to be adopted for the application experiment. The presence of the subcontractor also meant that the approach of thorough specification before designing anything was maintained, even when progress appeared ‘slow’.

The key lesson learned has been the value of rigorous and detailed specification and system design phases in the project.

Traditionally, a prototype would have been designed and constructed as early as possible during the project. Subsequently it would have been modified and “tweaked” in order to deliver the required performance. The primary reason for this approach is that it makes both management and the development team feel a sense of security, as there is something tangible to show that the project is making progress. Whilst this may work for relatively simple designs, as the system becomes more complex the chances that this approach will deliver anything close to an optimal design in the shortest possible time become increasingly remote.

During this project, a considerable amount of effort was expended in developing detailed system specifications and subsequently the high level system design. This meant that when the detailed design started, all of the interactions between the different components on the system (both hardware sub-systems and software modules) had been designed. Consequently the detailed design and software coding progressed rapidly, and when the hardware and software were integrated, the system functioned as intended the first time (barring a few minor mistakes that were quickly corrected).

The biggest problem was resisting the pressure from management, whose perception was that no progress was being made during the early phases. The company has now learnt that the effort expended working out how the system will work before lifting a soldering iron or writing a single line of code, pays for itself in the long run.

The second important lesson learned was the value of timely project reviews, throughout the development process. Early recognition that something has changed, or is not going according to plan enables corrective action to be taken, before the consequences become disastrous. For example, once the hardware design had been completed, it became obvious that the amount of time budgeted to design the printed circuit board was completely inadequate. If the company’s engineer had had to undertake this design, (as was originally planned), a delay of 4 to 6 weeks would have been incurred. In order to avoid this, the company purchased an auto-router to enhance its existing printed circuit CAD tools and asked the subcontractor to implement the design.

The success of the design reviews is highly dependant on the way in which they are conducted, and the atmosphere which prevails. It is vital that the meetings remain objective, forward looking and focus on problem solving. It is difficult enough for any member of the team to reveal a problem in front of his or her peers, but if they are subsequently made to feel like a scapegoat and the meetings degenerate into a witch-hunt, then they will not do so. At this point, the meetings become worthless. The strength of the technique lies in identifying problems as early as possible, before they become disasters, and taking corrective action early enough. This can only happen with the cooperation of every member of the project team.”