Review of College Instrumentation

Seventh College on Microprocessor-Based Real-Time Systems in Physics

Trieste, 28 October - 22 November, 2002

A.J. Wetherilt
Artesis A.S
Tuzla
Istanbul
Turkey

email:anthony.james.wetherilt@arcelik.com

Contents

Chapter 1

Review of College Instrumentation

by Anthony James Wetherilt

Abstract

Hardware developed for the Colleges on Microprocessor—based Real—time systems in Physics is reviewed. An embedded system based around an **MC 6809** microprocessor is introduced together with a real–time, multi–threading kernel developed to run on the board. The kernel is designed to implement a small memory manager, a task scheduler,software system calls and installable device drivers. On top of the system, several layers of software are implemented, that provide full high level language library support including a version of the Posix 1003.1c (PThreads) standard. Examples are provided that illustrate the use of these libraries together with methods for compilation and debugging of C code.

1.1 Introduction

Since the start of the series of Colleges on Real–Time Systems in Physics, several pieces of small but useful hardware items have been developed by members of the instruction staff with the aim of furthering the effectiveness of the material presented in the course lectures. Several such items are discussed in these notes from both their hardware, and where appropriate, their software aspects. It is important to realise that although developed primarily for teaching the principles of real time systems using personal computers and embedded systems, several pieces of the hardware can be used for a much wider class of applications than found in the teaching laboratory. Cards similar in design to the MC6809 board described here have been used by the author for many data acquisition applications such as temperature control, transient digitisers and intelligent signal averagers. When equipped with the IEEE 488 instrumentation interface, the *de facto* standard for small laboratories, such instrumentation can perform significantly better than many commercially obtainable pieces of equipment and at prices at least an order of magnitude lower.

1.2 Hardware

1.2.1 The Colombo board

The Colombo board is actually a complete system with provisions made for a 6809 microprocessor, a 6821 programmable interface adapter, and RAM and ROM situated on one half of the board (see Figure 1.2). The remainder of the board comprises a 4 digit, 8 segment LED display together with various switches and devices that can be interfaced via a 26 pin connector to either the on-board microprocessor or an external host machine. It is this latter feature that has been used predominantly during the various colleges.

The 26 pin connector definitions are shown in Figure 1.2 and are to be considered the standard connections for College instrumentation. Two sets of data lines can be seen which reflect the characteristics of the 6821 PIA around which the board was designed. The set of A lines (PA0 - PA7) are connected to the latch/drivers of the 4 LED displays and data latched in the following manner (Figure 1.3): The hexadecimal digit to be written on a given LED is placed on lines PA4 - PA7. The data is latched by first setting the E pin of the specified digit low and then resetting it back high again. As each digit has a separate line attached to it, digits that share the same data can be latched individually. A clock that produces pulses at a selectable rate can be attached to line CA1. When connected to a suitable input on the host device an interrupt can be raised by these pulses. On some cards a buzzer is connected to line CA2, on others the buzzer has been replaced by a LED. In either case, setting CA2 high causes the attached device to function.

Connections to the B side are entirely inputs (Figure 1.3): On lines PB4-PB7, a 16 position rotary switch is attached; on PB3 and PB2, are two toggle switches; and on PB0 and PB1 are two push button switches, connected via a 74279 for debouncing. These push buttons can also be jumpered to line CB2 which, when connected as an input on the host machine, can raise interrupts. Finally, a voltage to frequency converter is attached to line CB1. This device converts an analogue voltage signal into a sequence of pulses at a frequency determined by the magnitude of the signal. If the number of pulses arriving per unit time is counted, the magnitude of the signal can be determined.

1.2.2 The LCD display board

Designed, by C.S. Ang, as a more up to date and modern replacement for the Colombo board described previously, this card features a 16 digit ASCII LCD display panel in addition to two push button switches, an 8 way DIP switch and an 8 LED strip (Figure 1.4). A number of different connectors allows several possibilities for the host machine. The first of these, is a 40 pin strip connector for direct interfacing to the 6811 card described next. The standard ICTP 26 pin connector is also found on the card allowing connections to be made to either the GPI card or the 6809 card described later. Since the latter connector has fewer pins than the former, the card functionality is also somewhat reduced when this

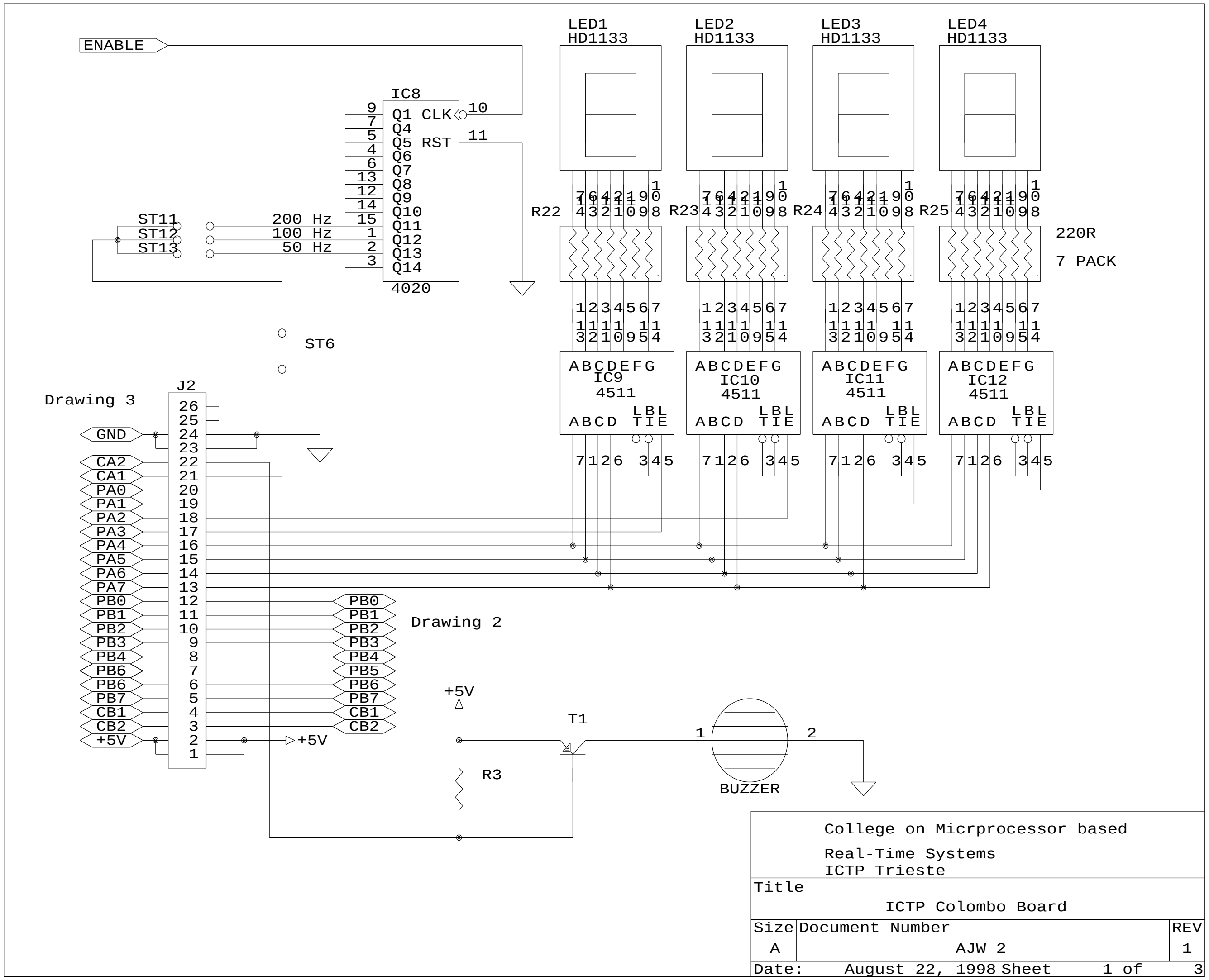


Figure 1.1: Schematic Drawing of the Colombo Board

+5V	1	2	+5V
CB2	3	4	CB1
PB7	5	6	PB6
PB5	7	8	PB4
PB3	9	10	PB2
PB1	11	12	PB0
PA7	13	14	PA6
PA5	15	16	PA4
PA3	17	18	PA2
PA1	19	20	PA0
CA1	21	22	CA2
Ground	23	24	Ground
Timer 3 gate	25	26	Timer 3 output

Figure 1.2: Pin Definition of the Standard ICTP

connector is used. However, it still allows a significantly better display capability than the Colombo board. As the details of the card with reference to the 6811 interface are more than adequately covered in the notes of C.S.Ang, only those aspects relevant to the standard ICTP interface will be discussed here.

The LCD display is an Agena AA16102 module capable of displaying up to 16, 5x7 pixel alphanumeric characters. With CA2 held high, data are placed on lines PB0-PB7. The line PA1 is set low for a write operation and PA2 is set according to whether the operation is a write data (high) or a write instruction (low). The line PA0 is then strobed from low to high to low for the data to be latched. For full details of all possible operations, please refer to the manufacturer's data sheet.

If line CA2 is set low, writing to the LCD is disabled and the DIP switch and LED strip become accessible by reading from and writing to, the lines PB0-7 and PA0-7 respectively. In both modes a push button is connected to CA1 to allow interrupt capability. Neither the second push button, nor the analogue voltage are connected when using the ICTP connector.

1.2.3 The M68HC11 board

This board was also designed by C.S.Ang for this College and provides a complete but minimal system with but three IC components (Figure 1.5). The board's designer discusses its details in his notes elsewhere and the reader is referred to these for further information

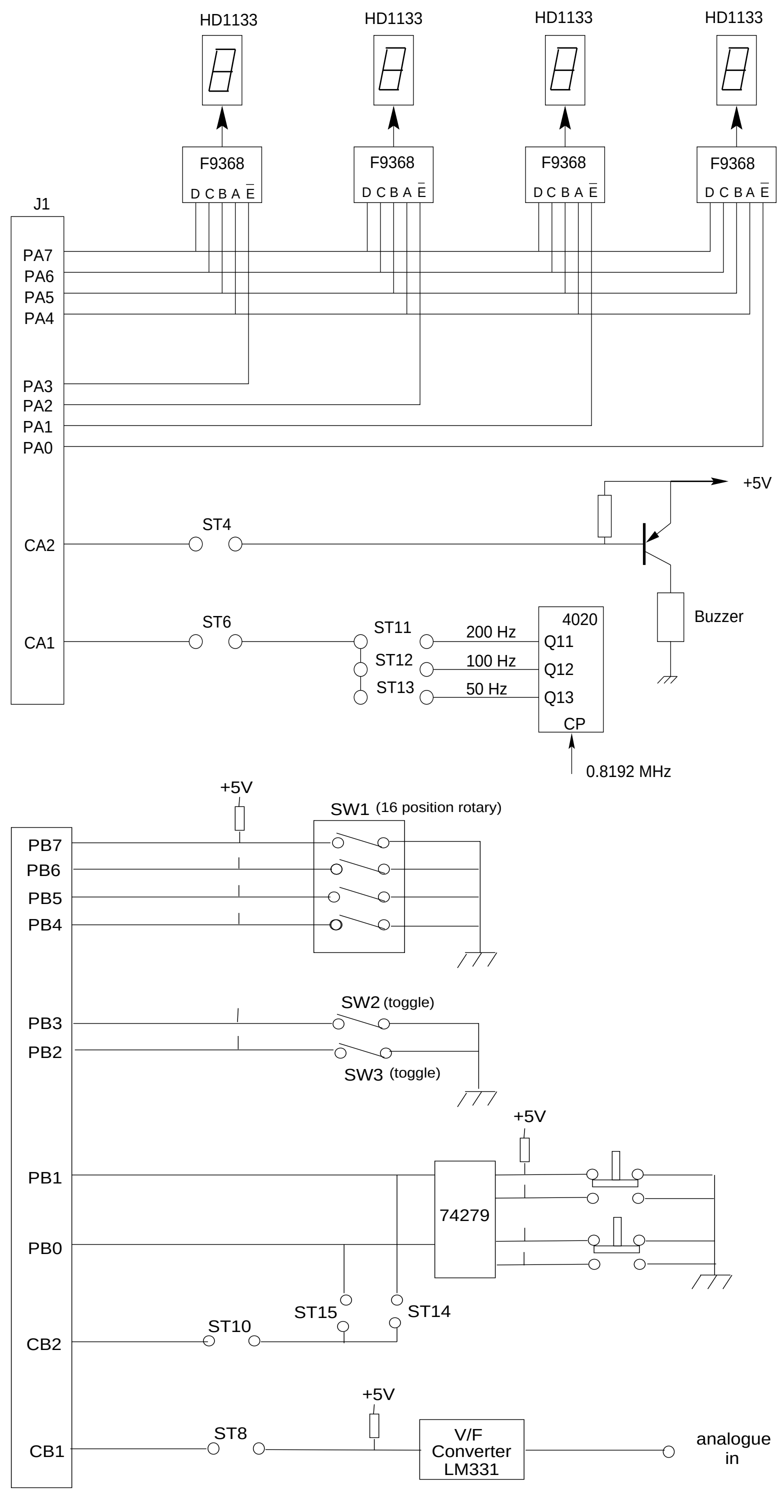


Figure 1.3: Simplified details of the Colombo Board Showing the A and B Sides

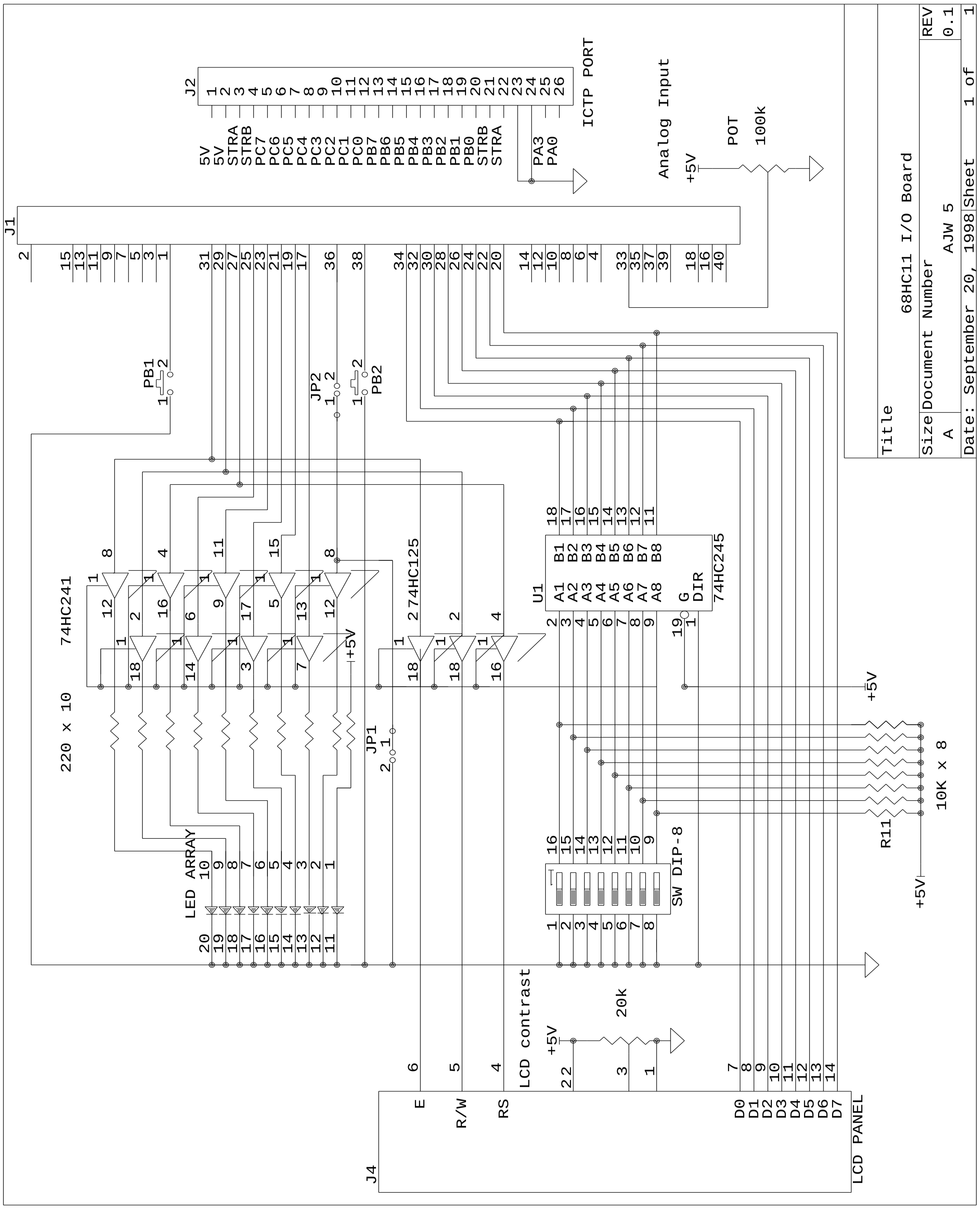


Figure 1.4: Schematic Drawing of the LCD Display Board

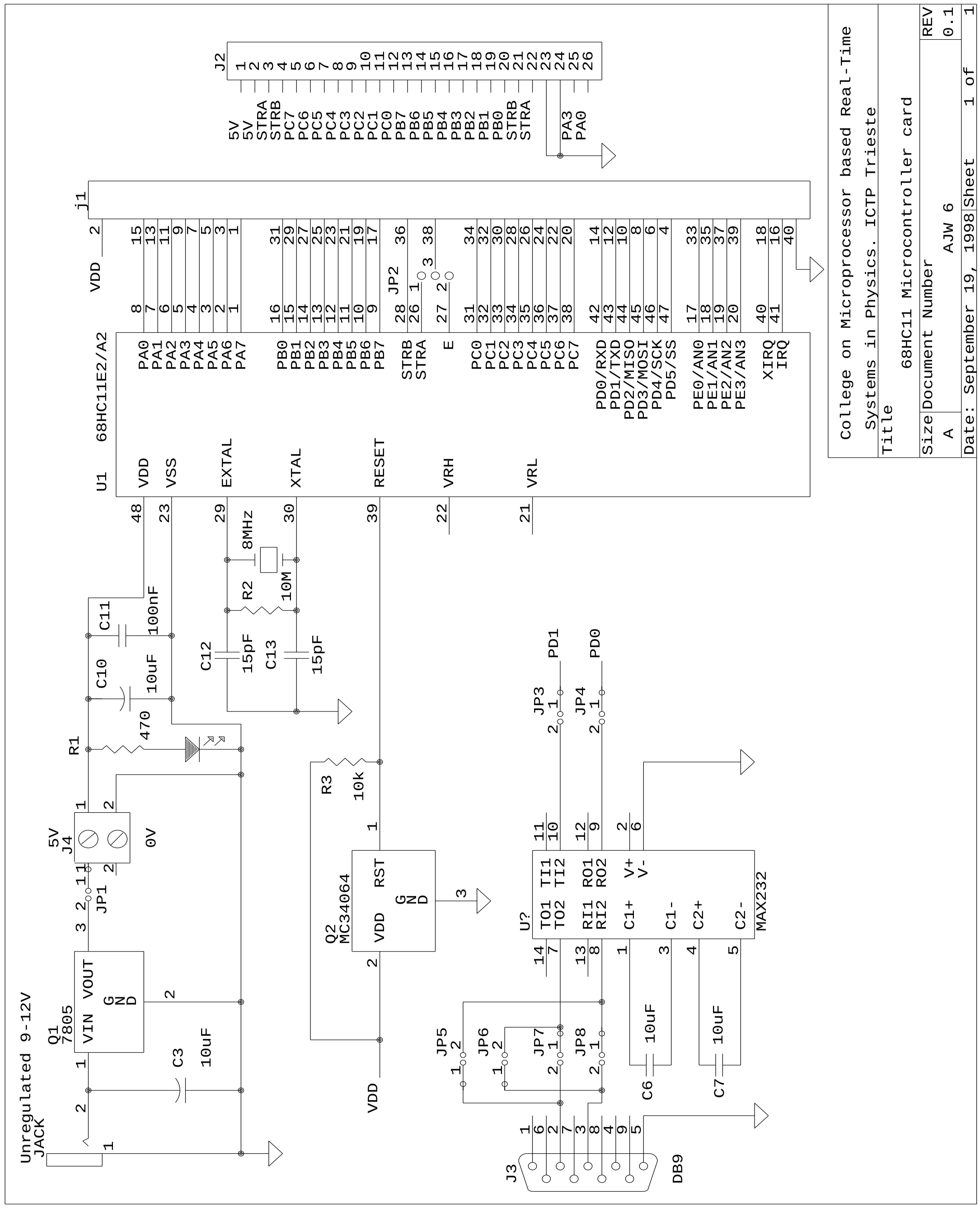


Figure 1.5: Schematic Drawing of the M6811 board

1.2.4 The ICTP09 board

Introduction

This board was also designed by the author for the present College to provide better coverage of embedded systems programming. In contrast to the 6811 board, the 6809 board implements a large number of functions at the price of complexity: Some 24 integrated circuits are used in its construction (see Figure 1.6). It comprises:

- 2 serial communications ports
- 1 parallel port
- 3 timer channels
- 2 channels of 12 bit ADC input
- 2 channels of 12 bit DAC output
- 16k EPROM
- 8k base RAM
- 128k RAM arranged in 4 pages, each of 32k

The memory map of the system is shown in Figure 1.7.

In order to facilitate programming, a command interpreter shell is provided at address 0xf000. The shell allows full debugging and downloading facilities including the setting of breakpoints and tracing through instructions. A real time kernel, RInOS (or Real time INtegrated Operating System) has been written that provides the basic functions of a real time multitasking environment such as context switching, semaphores, message passing between tasks, priority changing, hardware interrupt handling and device drivers etc. As these services are provided in the form of system calls, the user is unaware of the details of the kernel and has only to supply a means of using the system calls from the programming language of his choice. For C running under LINUX, the GCC compiler modified to produce absolute code for the 6809 can make use of libraries encapsulating the assembler commands. Simple memory management is provided so that a process can be allocated memory as and when it is needed and return the memory to the heap when finished. Processes can be loaded and relocated by the memory manager. Alternatively, absolute code can be used as long as certain well defined steps are followed.

Hardware description

The board is based around a MC6809 processor running at a clock speed of 1 MHz. Although the 6809 is now an old microprocessor, its use in a piece of hardware intended mainly for teaching purposes can be justified on the grounds of its superior instruction set and clarity of use. The 6809 arguably, still has the best instruction set of any 8 bit microprocessor or micro controller and is ideally suited for the current purpose. Development tools are widely and freely available at many sites on the Internet which is a great advantage for any device.

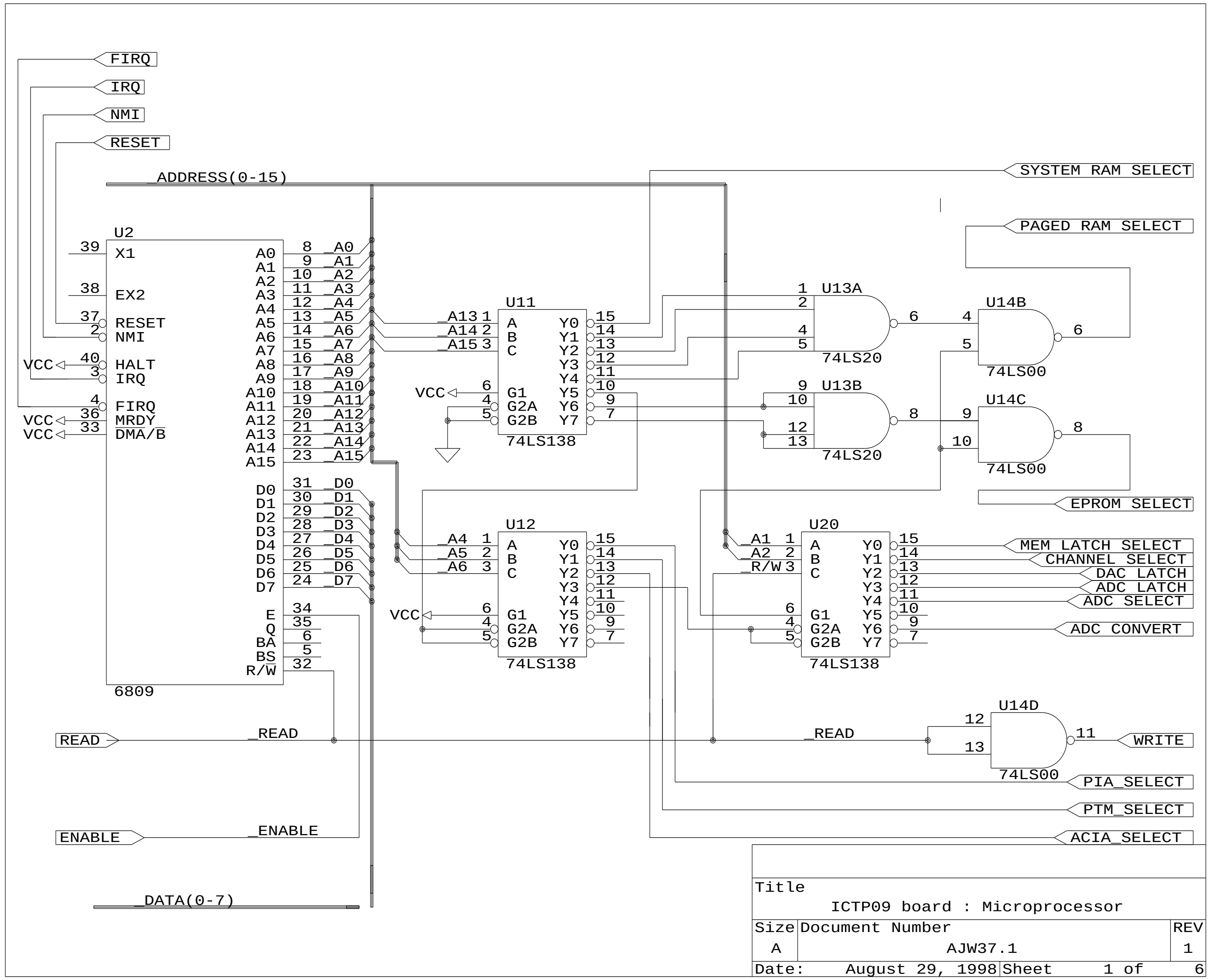


Figure 1.6: Schematic Drawing of the M6809 board

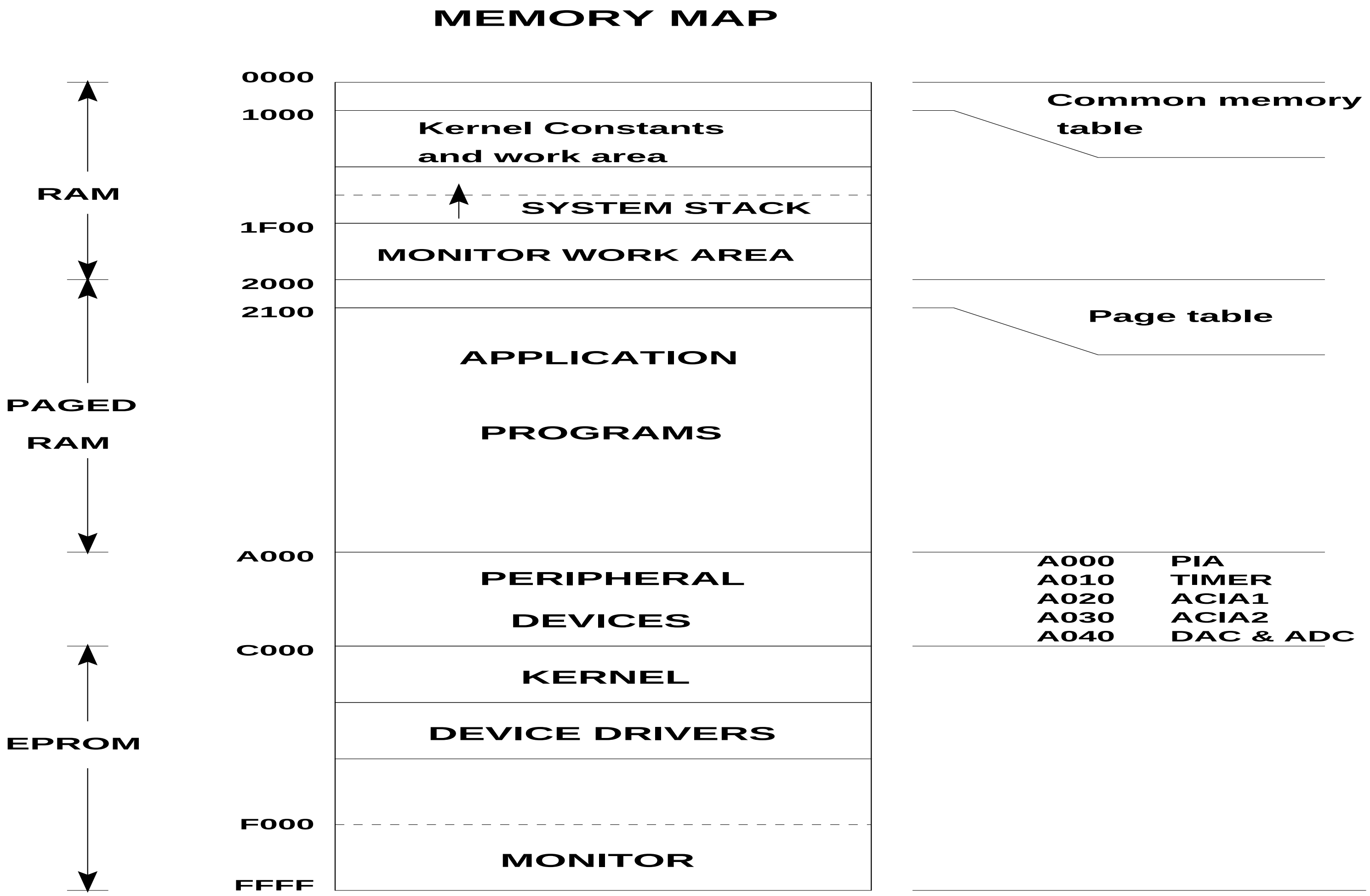
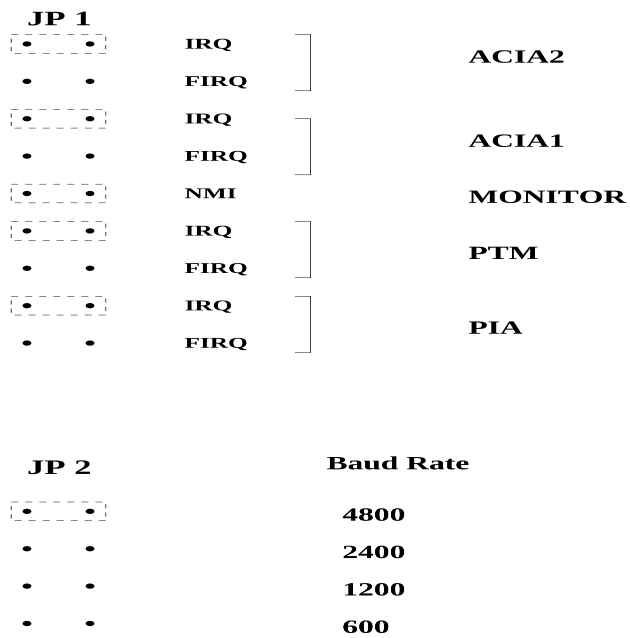


Figure 1.7: Memory Map of the M6809 under RInOS



Dashed line indicates default jumper settings.

Figure 1.8: Jumpers Setting for the ICTP09 Board

Throughout the design stage, stress has always been laid on those areas that will allow the various aspects of microprocessor teaching to be emphasised. For this reason two identical serial communications ports have been provided. These allow communications drivers to be debugged easily using one port connected via the monitor to the host machine and the second to the hardware application. For both ports, the baud rate can be set by changing jumper JP2. If faster rates are required, the ACIAs at 0xA020 and 0xA030 (Figure 1.7) must be configured so that the clock is divided by 1 rather than 16 and the jumpers adjusted accordingly. Communication uses only the TxD, RxD and ground return lines of the RS 232 9 pin ports. For interconnection between the board and a host PC, null modem cables must be used.

The 6840 PTM provides 3 timer channels. The first is attached to the NMI line and is used by the monitor for tracing through code, and the second is used for the system clock by the kernel. It issues a clock interrupt on the IRQ line at 10 ms intervals. The third clock is available to a user and has both gate and output on the onboard standard ICTP 26 pin strip connector. To ensure these and other interrupt signals are processed, the jumpers must be set correctly on jumper JP1. Under RInOS, all interrupts except the MON signal from timer channel 1 which is jumpered to the NMI line, must be jumpered to the IRQ line. Jumpering to the FIRQ line without special provision will cause unpredictable results and generally will hang the system. Refer to Figure 1.8 for a description of the jumper settings.

Random access memory is used to provide (i) a common area for system and application programme use and (ii) an area in which large processes can be loaded. These are supplied by a 2764 equivalent 8k RAM at 0x0000 - 0x1FFF and a 581000 128k RAM at 0x2000-0x9FFF. Since the entire address space of the 6809 is only 64k, the 128k of the 581000 is divided into 4 pages each of 32k in size by decoding the upper two address lines of the 581000 with an address latch. Writing the values 0-3 to the latch will cause the appropriate page to be set. It is advised that application processes do not interfere with this register when the kernel is running.

Two channels each of ADC and DAC are provided. No interrupt capability is provided for the ADC channels as at a clock rate of one MHz, conversion takes less than approximately 25 μs, which is only barely more than the time required to handle a straight forward interrupt request. For times longer than this, timer channel 3 can be used.